



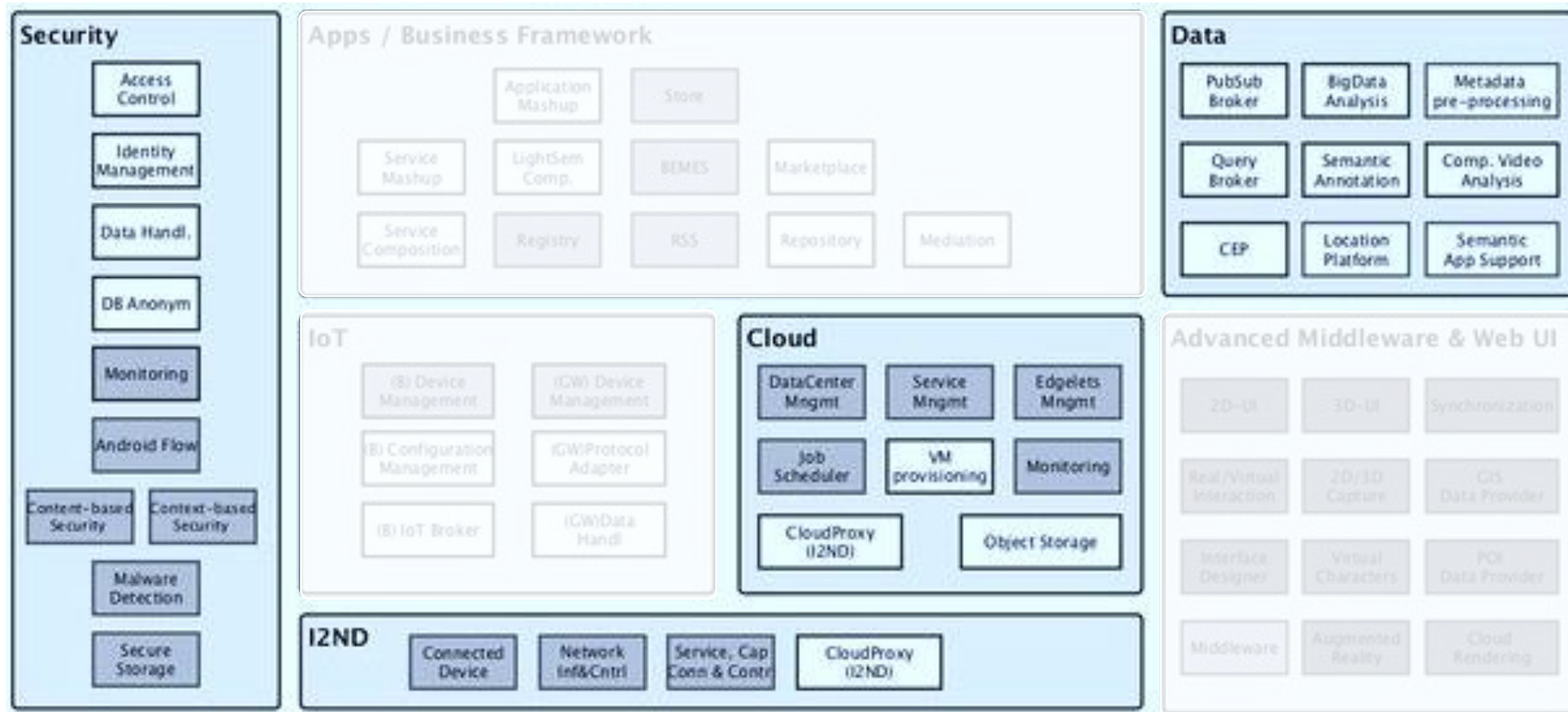
*CONTEXT / DATA MANAGEMENT,
CLOUD HOSTING, SECURITY & I2ND*

Gradiant

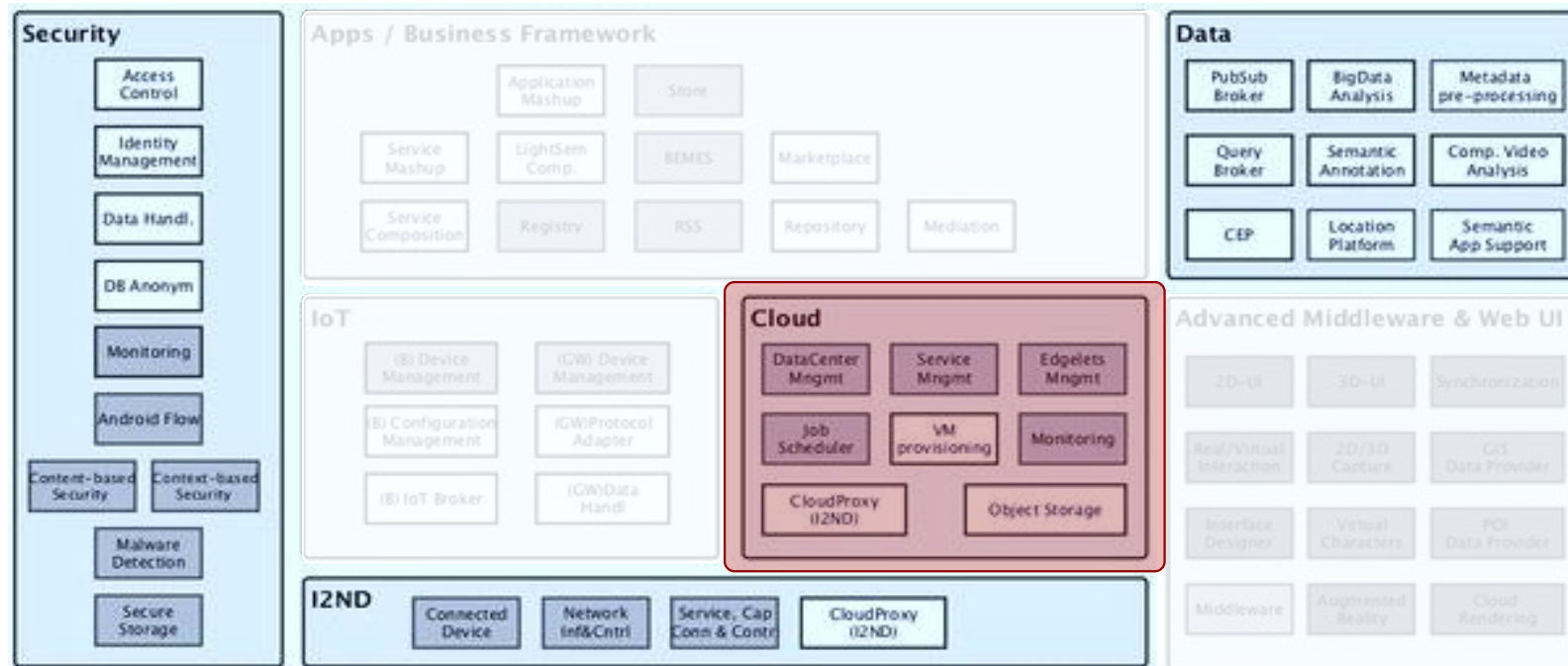
Galician research and development center
in advanced telecommunications

2015

Architecture Overview



CLOUD HOSTING



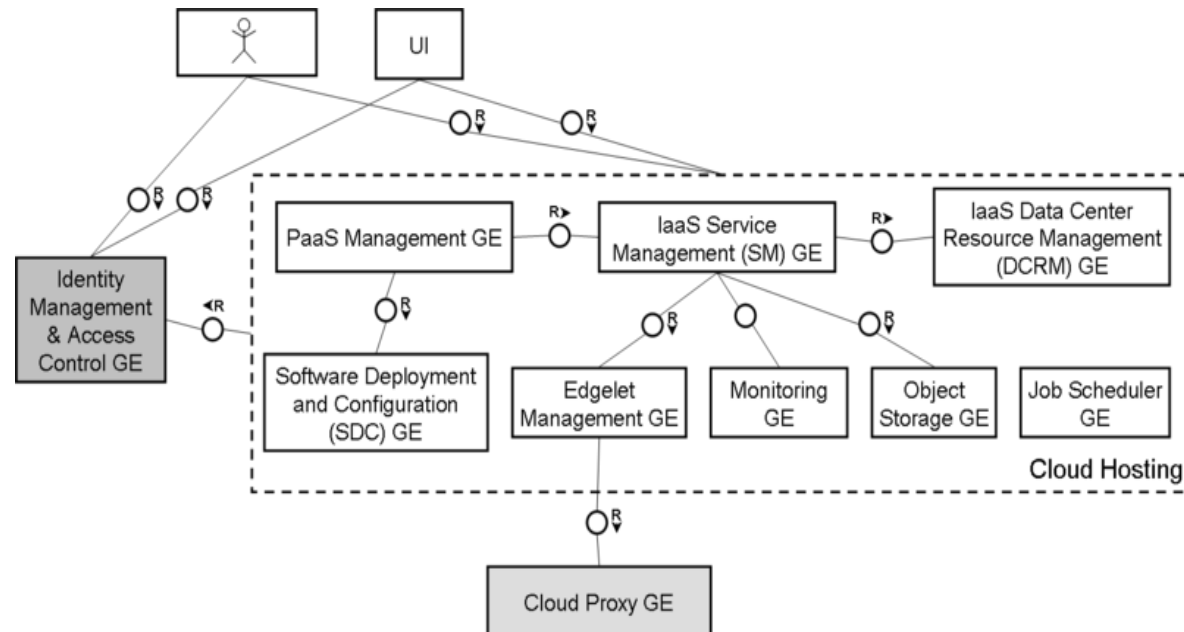
Architecture Overview

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.

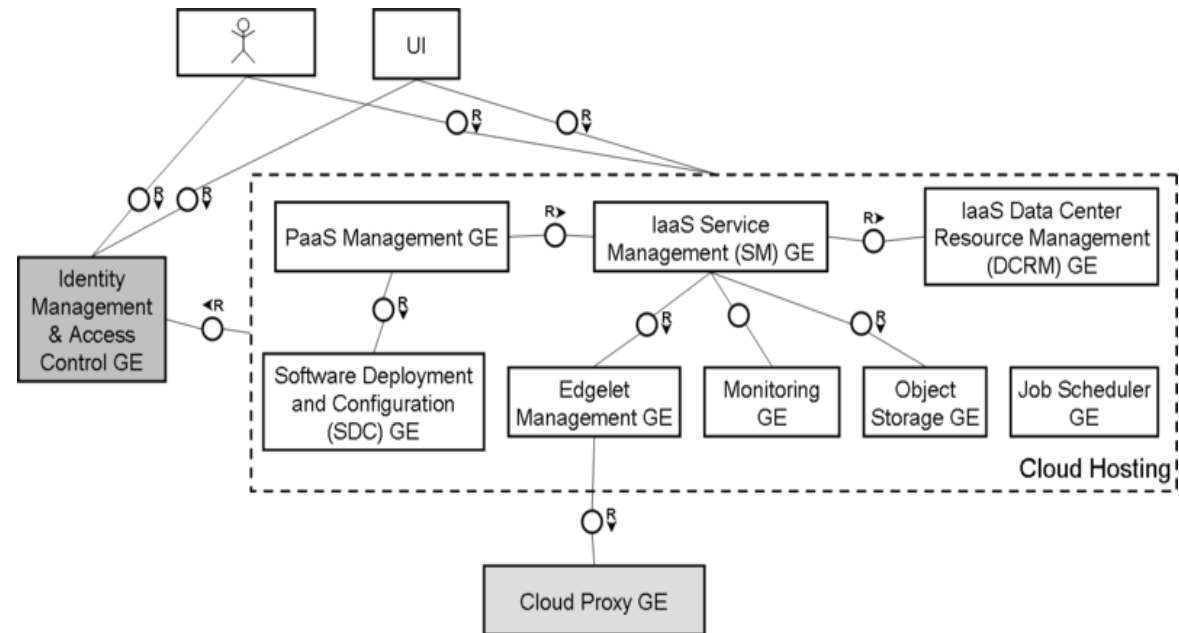
(continue ...)



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

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.



FIWARE IaaS Resource Management

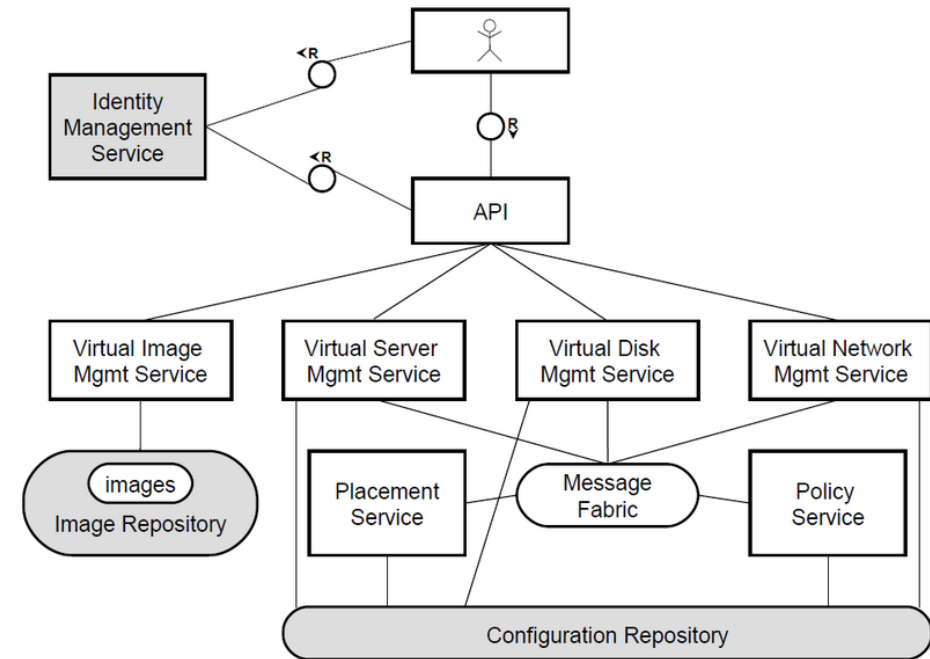
The main capabilities of this GE are:

- 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

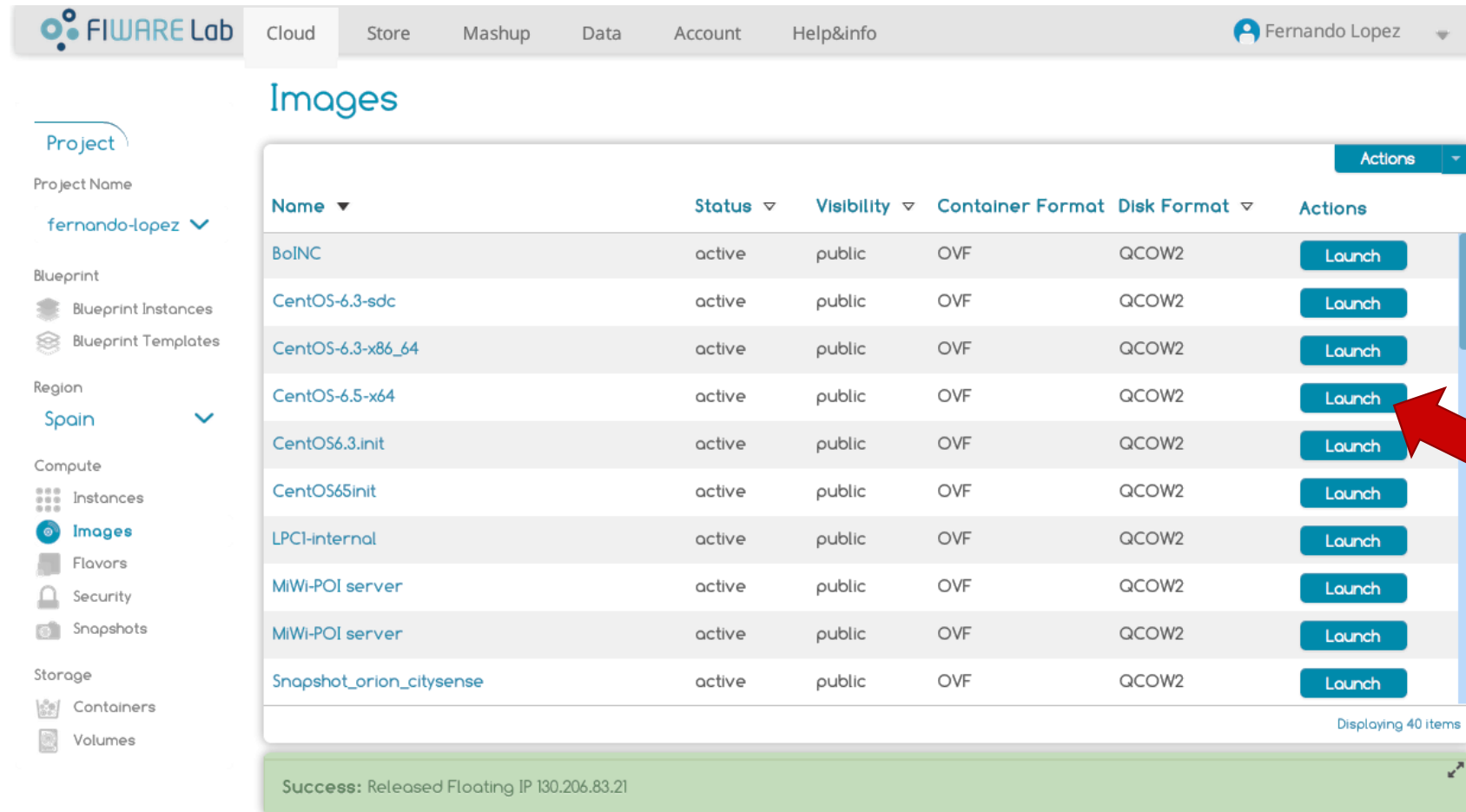
IaaS Resource Management GE – DCRM Provisioning

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



The screenshot shows the FIWARE Lab Cloud interface. The top navigation bar includes links for Cloud, Store, Mashup, Data, Account, and Help&info. The user is logged in as Fernando Lopez. The left sidebar shows the Project Name 'fernando-lopez' and various navigation options under Compute, including Images, which is currently selected. The main content area displays a table of available images.

Name	Status	Visibility	Container Format	Disk Format	Actions
BoINC	active	public	OVF	QCOW2	Launch
CentOS-6.3-sdc	active	public	OVF	QCOW2	Launch
CentOS-6.3-x86_64	active	public	OVF	QCOW2	Launch
CentOS-6.5-x64	active	public	OVF	QCOW2	Launch
CentOS6.3-init	active	public	OVF	QCOW2	Launch
CentOS65init	active	public	OVF	QCOW2	Launch
LPCI-internal	active	public	OVF	QCOW2	Launch
MIWi-POI server	active	public	OVF	QCOW2	Launch
MIWi-POI server	active	public	OVF	QCOW2	Launch
Snapshot_orion_citysense	active	public	OVF	QCOW2	Launch

Displaying 40 items

Success: Released Floating IP 130.206.83.21

2014 © FIWARE. The use of FIWARE Lab services is subject to the acceptance of the [Terms and Conditions](#), [Personal Data Protection Policy](#) and [Cookies Policy](#)

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.

Launch Instances

1. Details 2. Access & Security 3. Post-Creation 4. Summary

Instance Name *

Flavor

m1.tiny

Instance Count *

1

Description

Specify the details for launching an instance. The chart below shows the resources used by this project in relation to the project's quotas.

Flavor Details

Name	m1.tiny
VCPUs	1
Root Disk	0 GB
Ephemeral Disk	0 GB
Total Disk	0 GB
RAM	512 MB

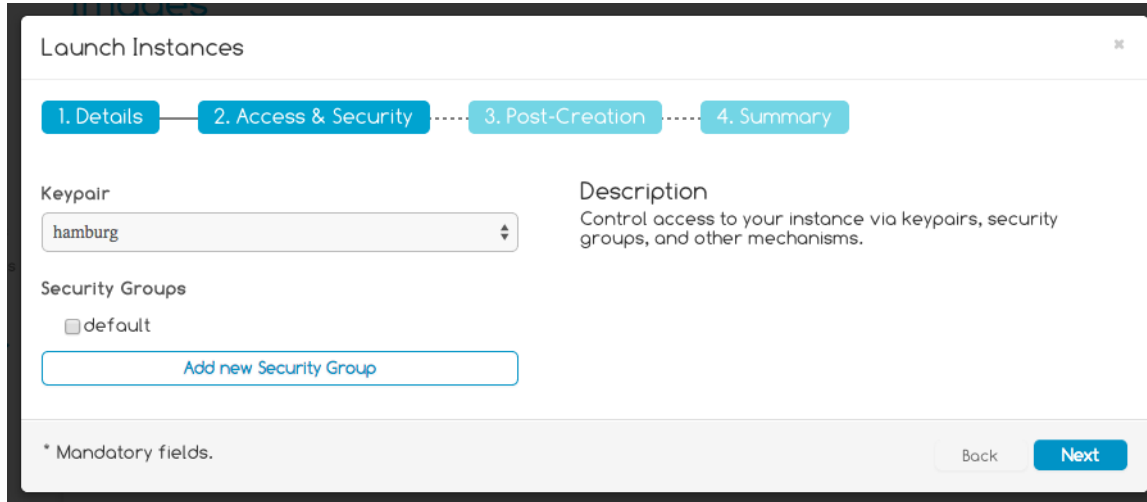
Project Quotas

Instance Count (3)	0 Available
VCPUs (3)	3 Available
Disk (20 GB)	980 GB Available
Memory (4608 MB)	20392 MB Available

* Mandatory fields.

Cancel Next

IaaS GE – Launch instance III



Launch Instances

1. Details — 2. Access & Security — 3. Post-Creation — 4. Summary

Keypair
hamburg

Security Groups
☐ default
Add new Security Group

Description
Control access to your instance via keypairs, security groups, and other mechanisms.

* Mandatory fields.

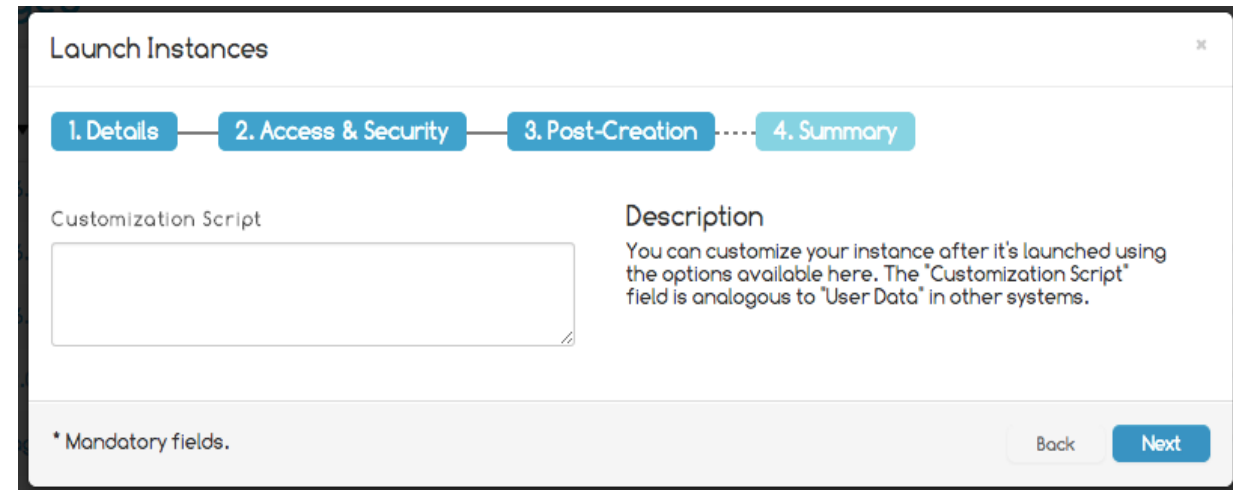
Back Next

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*”



Launch Instances

1. Details — 2. Access & Security — 3. Post-Creation — 4. Summary

Customization Script

Description
You can customize your instance after it's launched using the options available here. The "Customization Script" field is analogous to "User Data" in other systems.

* Mandatory fields.

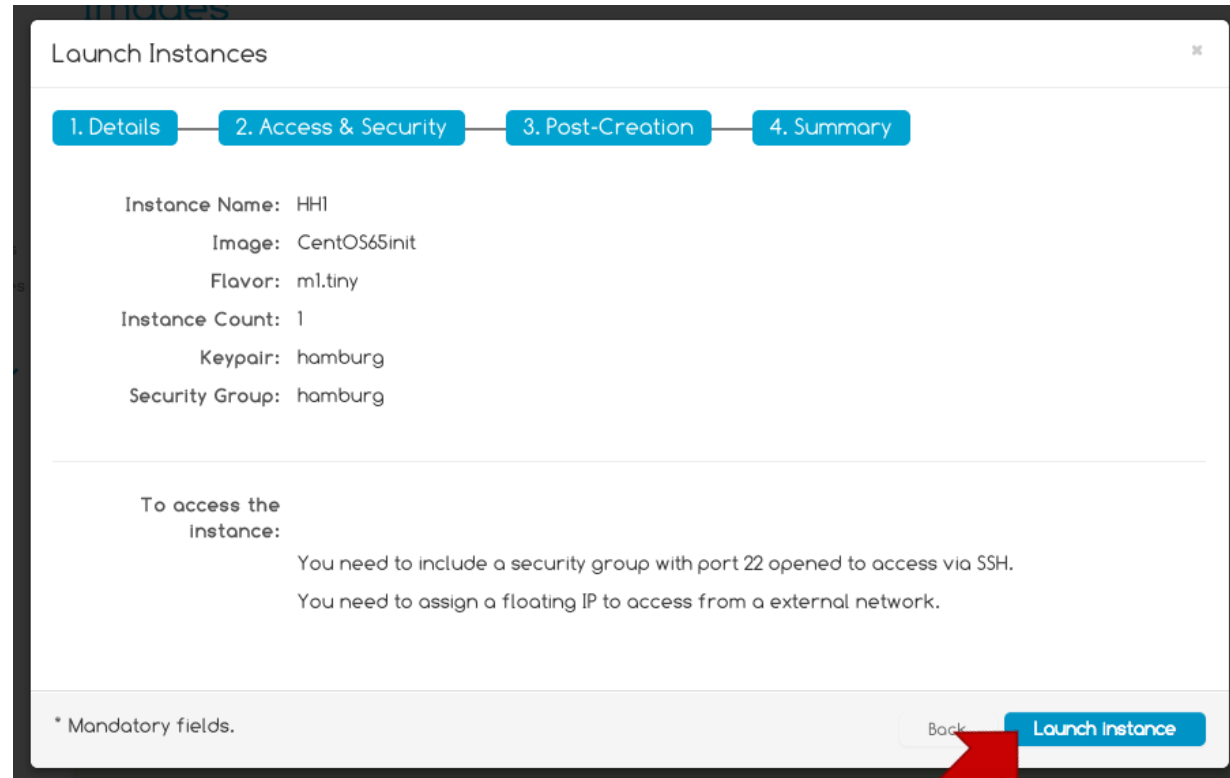
Back Next

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”



Launch Instances

1. Details 2. Access & Security 3. Post-Creation 4. Summary

Instance Name: HH1
Image: CentOS65init
Flavor: m1.tiny
Instance Count: 1
Keypair: hamburg
Security Group: hamburg

To access the instance:
You need to include a security group with port 22 opened to access via SSH.
You need to assign a floating IP to access from a external network.

* Mandatory fields.

Back Launch Instance

IaaS GE – Allocate Floating IP

The screenshot shows the FIWARE Lab interface. The top navigation bar includes 'Cloud', 'Store', 'Mashup', 'Data', 'Account', and 'Help&info'. The user 'Fernando Lopez' is logged in. The left sidebar shows a 'Project' section with 'fernando-lopez' selected, and a 'Security' section with 'Instances' selected. The main content area is titled 'Security' and has tabs for 'Floating IPs', 'Security Groups', and 'Keypairs'. The 'Floating IPs' tab is active, showing a table with columns 'IP Address', 'Instance', and 'Floating IP Pool'. A row is visible with IP '130.206.83.21' and pool 'net8300'. An 'Actions' button is next to the row, and a red arrow points to it. A dropdown menu is open from the 'Actions' button, showing options: 'Associate IP', 'Dissasociate Floating IP', and 'Release Floating IPs'. A modal dialog titled 'Associate Floating IP' is open in the foreground. It contains the following fields: 'Associate Floating IP:' with the value '130.206.83.21', 'to instance:' with a dropdown showing 'HH1', and 'and to IP Address:' with a dropdown showing 'Select IP to associate with'. The modal also has a 'Description' section with the text 'Associate a floating ip with an instance.' and buttons for 'Cancel' and 'Associate IP'.

Success: Success

2014 © FIWARE. The use of FIWARE Lab

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

Cloud Store Mashup Data Account Help&info Fernando Lopez

Instances

Overview Log Connection Monitoring

Info

Name: HH1
ID: 125cd18e-fa14-4f5a-8d4e-14a524b5d4fe
Status: ACTIVE

Security Groups

default

Installed Software

[Edit](#)

Success: Successfully allocated floating IP

the use of FIWARE Lab services is subject to the acceptance of the terms and conditions of the FIWARE Lab services.

Specs

RAM: 512MB
VCPU: 1

IP Addresses

private: 10.0.4.209

Edit Software in Instance

Select the software you want to install in this instance. The software will be installed in a moment when added to the left-hand table.

You can also uninstall the software by dragging it back to the right-hand table (Software in Catalogue).

Software in Instance		Software in Catalogue
python 2.7.5	INSTALLED	augmentedreality 3.3.3
wirecloud 0.6.4	INSTALLED	django 1.5.5
		git 1.7
		mediawiki 1.17.0
		mongodbconfig 2.2.3
		mongodb 2.2.3
		mongodb 2.2.3

[Done](#)

An instance may require 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

The screenshot shows the FIWARE Lab web interface. The top navigation bar includes links for Cloud, Store, Mashup, Data, Account, and Help&info, along with a user profile for Fernando Lopez. The left sidebar lists various project and resource categories, with 'Blueprint Templates' highlighted and indicated by a red arrow. The main content area, titled 'Blueprint Templates', features a table with columns for Name, Description, and Tiers. A modal window titled 'Create Blueprint' is open, allowing users to define a new blueprint. The modal includes input fields for 'Name' (containing 'hh-template') and 'Description' (containing 'description'), and buttons for 'Cancel' and 'Create Blueprint'. A success message 'Success: Containers' is visible at the bottom left of the main area.

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.

Blueprint Templates

Open Catalog Create New Template Actions

Name	Description	Tiers
hh-template	description	0

Create Tier

1. Details 2. Install Software

4 0

?

Name *:

Region: Spain

Flavor *: m1_small (1VCPU / 10GB Disk / 2048ME)

Image *: ubuntu-13.10-sdc

Icon: View

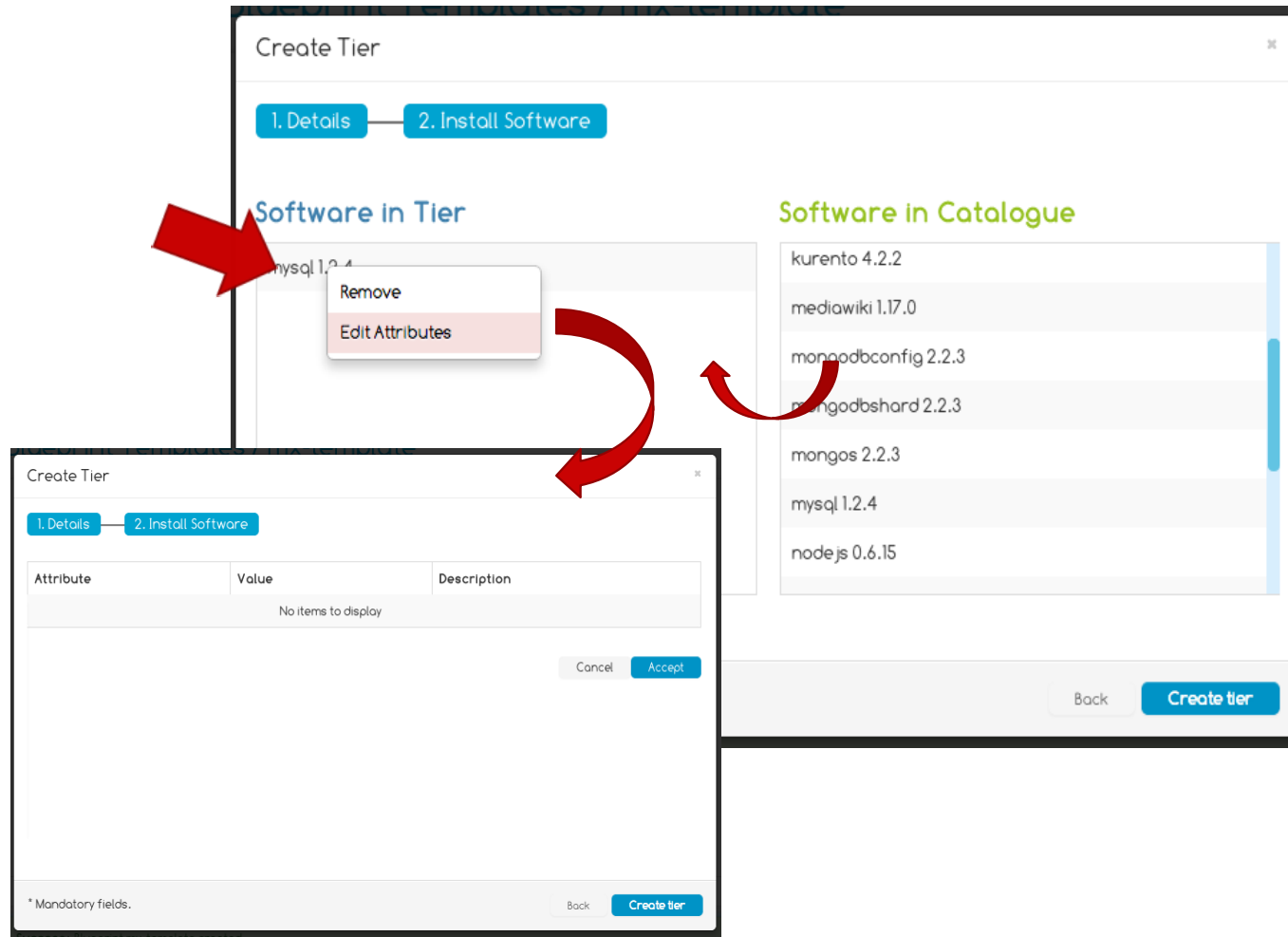
Keypair: mexico

Public IP: ☐

* Mandatory fields.

Cancel Next

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.

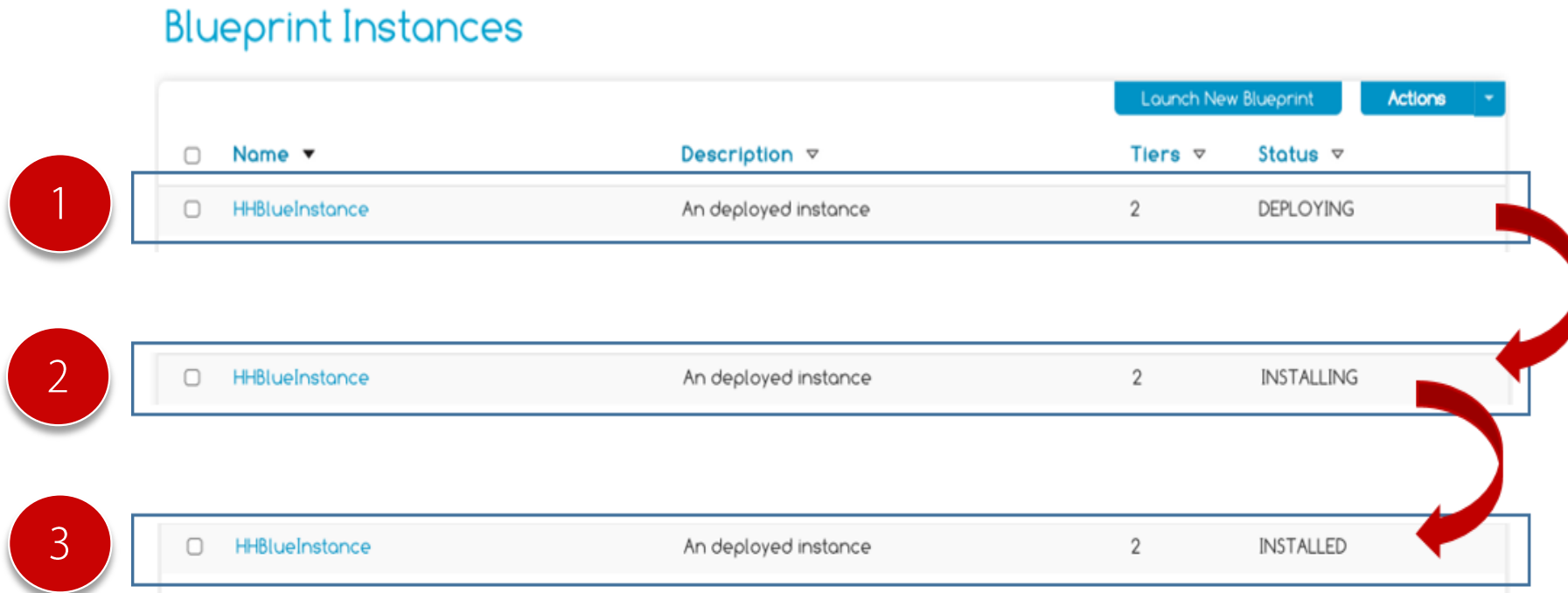
PaaS GE – From Template to Instance

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.

The screenshot shows the FIWARE Lab interface. The top navigation bar includes 'Cloud', 'Store', 'Mashup', 'Data', 'Account', and 'Help&info'. The user 'Fernando Lopez' is logged in. The left sidebar shows a project structure with 'Project Name' (fernando-lopez), 'Blueprint' (Blueprint Instances, Blueprint Templates), 'Region' (Spain), 'Compute' (Instances, Images, Flavors, Security, Snapshots), and 'Storage' (Containers, Volumes). The main area displays 'Blueprint Templates' with a table containing one entry: 'hh-template' with description 'description'. The 'Actions' dropdown menu is open, showing 'Launch Template', 'Clone Template', and 'Delete Template'. A modal window titled 'Launch Blueprint Instance' is open, showing fields for 'Name' (HHBlueInstance) and 'Description' (An deployed instance). The modal also has a 'Launch Blueprint Instance' button and a 'Cancel' button. A red arrow points to the 'Launch Template' action in the 'Actions' dropdown menu.

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.

The screenshot displays the FIWARE Lab web interface. The top navigation bar includes links for Cloud, Store, Mashup, Data, Account, and Help&info. The main heading is "Blueprint Instances / HHBlueInstance". On the left, a sidebar menu shows the navigation structure: Project (fernando-lopez), Blueprint (Blueprint Instances, Blueprint Templates), Region (Spain), Compute (Instances, Images, Flavors, Security, Snapshots), and Storage (Containers, Volumes). A red arrow points to the "Blueprint Instances" link. The main content area shows two instances, each with a circular progress indicator (4/0) and a question mark icon. To the right of each instance, details are listed: Name (hh-tier1, hh-tier2), Flavor (ml.small), Image (CentOS6.3.init), and Keypair (hamburg). Further right, two "Software in Tier" sections are shown, listing "tomcat 6" and "mysql 1.2.4" respectively.

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

The screenshot displays the OpenStack Volumes dashboard. On the left sidebar, the 'Volumes' link is highlighted with a red arrow. The main panel shows a table with columns: Name, Description, Size (GB), Status, and Attachments. A red arrow points to the 'Create Volume' button in the top right of the table. A modal window titled 'Create Volume' is open, containing the following fields:

- Volume Name ***: A text input field containing 'volume1'.
- Description**: A text area containing 'A volume'. To the right of this field, a description states: 'Volumes are block devices that can be attached to instances.'
- Size (GB) ***: A numeric input field containing '1'.

At the bottom of the modal, there is a note '* Mandatory fields.' and two buttons: 'Cancel' and 'Create Volume'.

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

Volumes

					Create Volume	Actions
<input type="checkbox"/>	Name ▾	Description ▾	Size (GB) ▾	Status ▾	Attachments ▾	
<input checked="" type="checkbox"/>	volume1	A volume	1	In-use	1	

Manage Volume Attachments

Attachments

Detach Volumes

	Instance	Device	Actions
Displaying 0 items			

Attach To Instance

Attach to Instance *

HH1

Device Name *

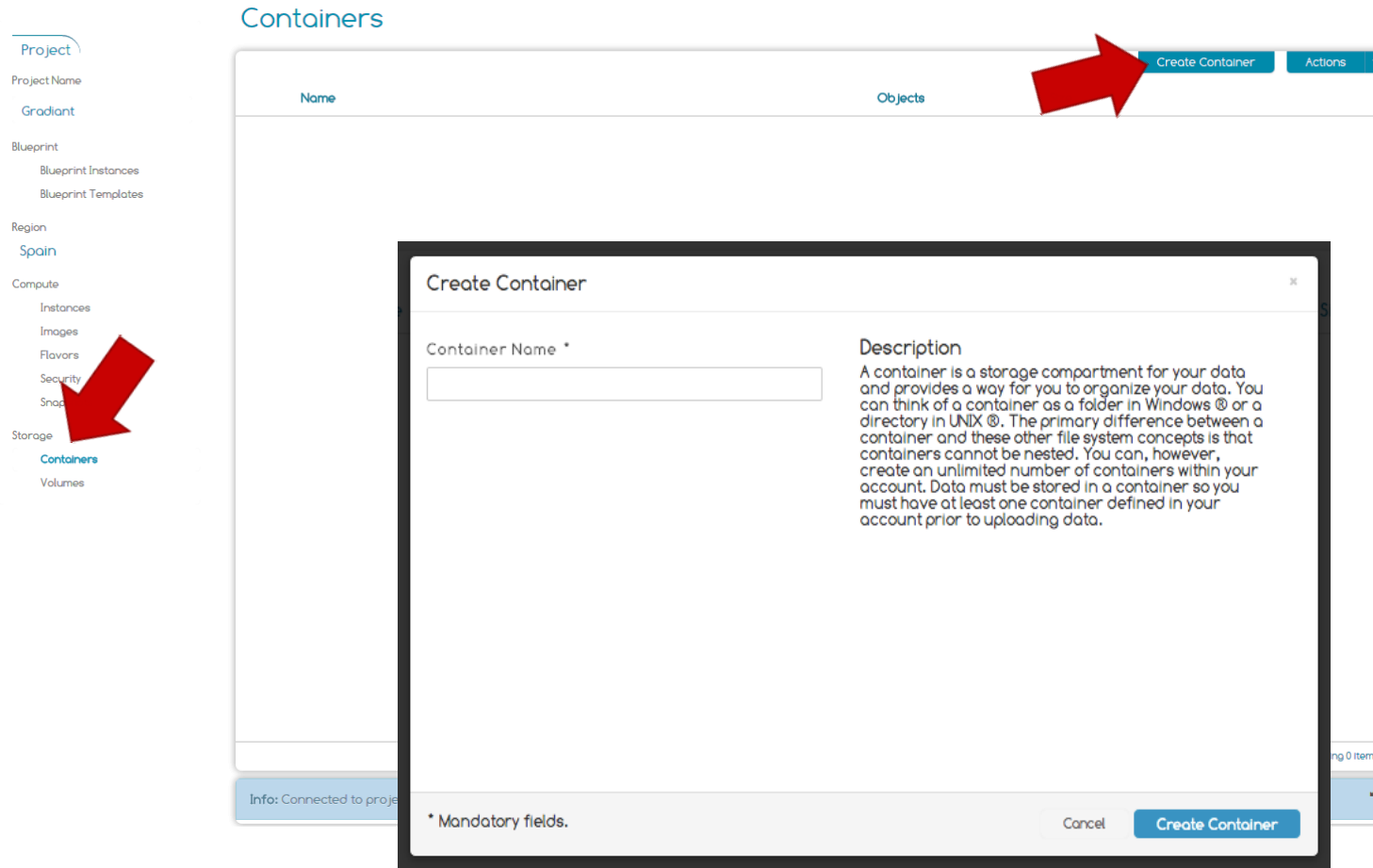
/dev/vdc

* Mandatory fields.

Cancel

Attach Volume

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.

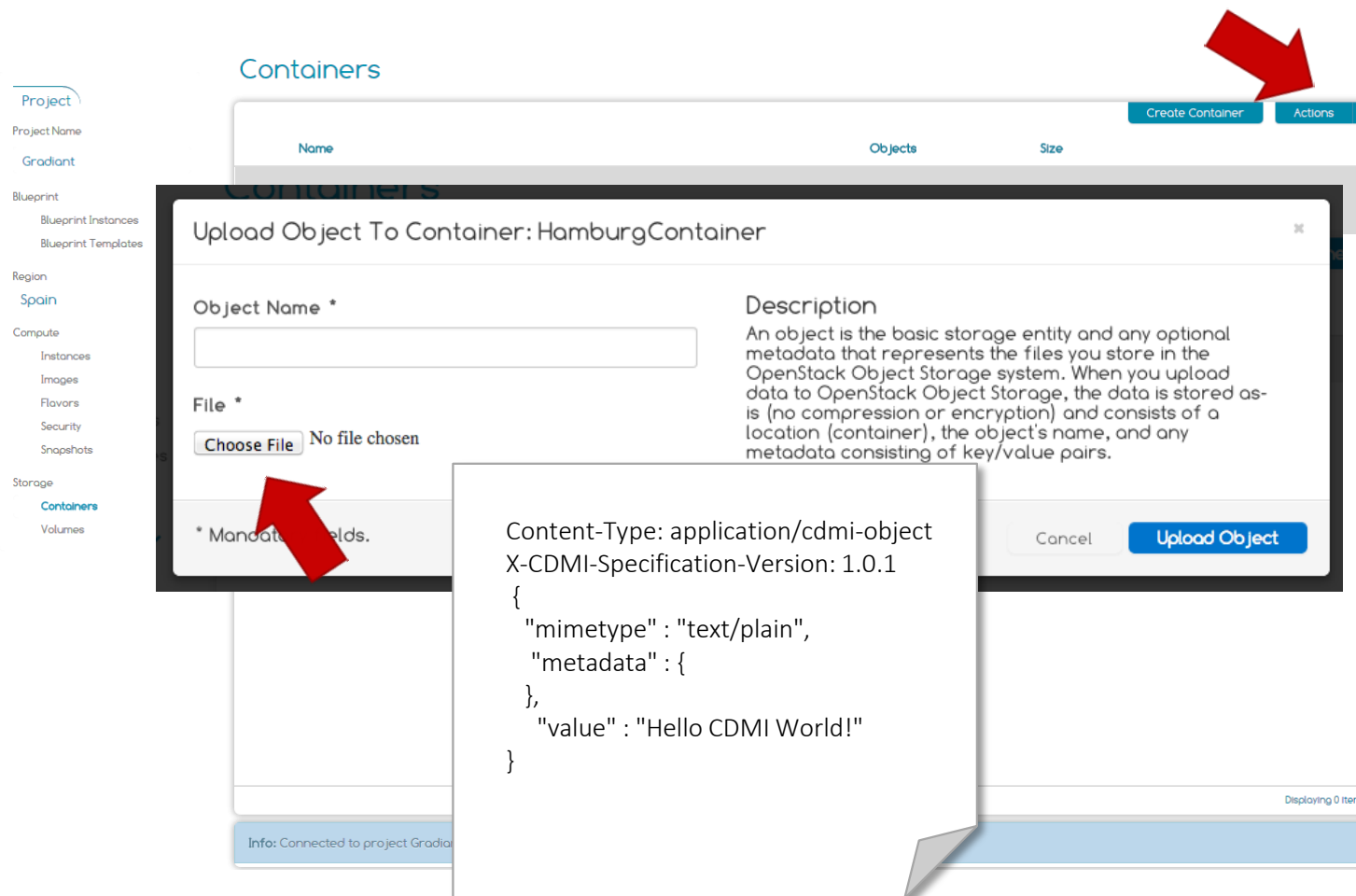
These containers can be created and removed on demand.

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

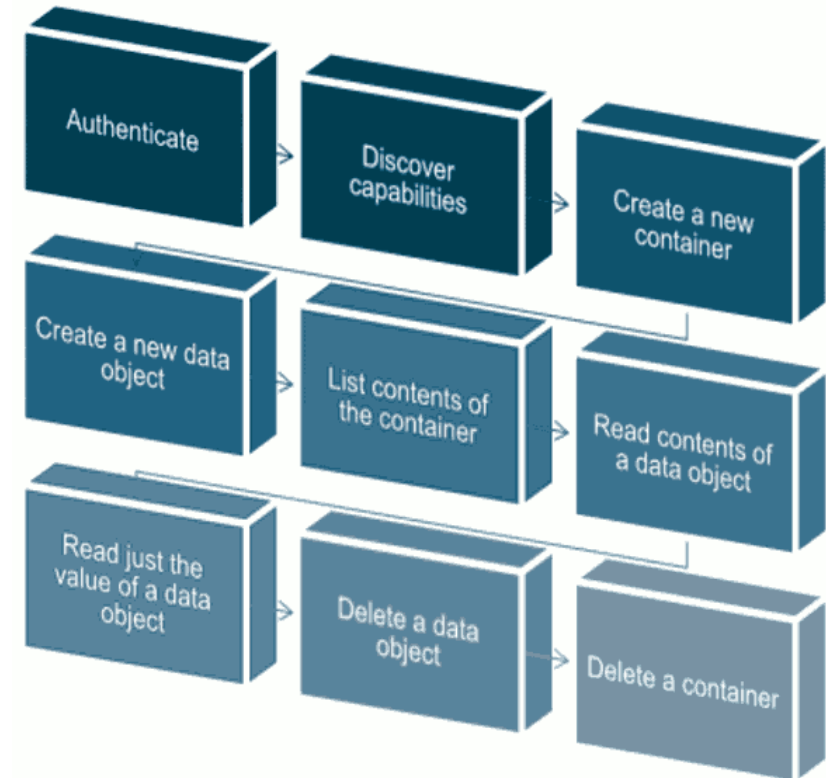
```
$ curl -v -X GET -H 'X-Auth-Token: b518d061ce2844ac80d1e116c9d857ed' -H 'Accept: application/cdm-capability' -H 'X-CDMI-SpecificationVersion: 1.0.1' http://130.206.80.102:8080/cdm_capabilities/AUTH_d418851c6d294381bbe6e082849686d6/
```

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/cdm/AUTH_d418851c6d294381bbe6e082849686d6/foo/
```

Create object

```
$ curl -v -X PUT -d '{"mimetype": "text/plain", "metadata": {}, "value": "bar"}' -H 'X-Auth-Token: 8197f75188304431bd3181ca79b0b884' -H 'Accept: application/cdm-object' -H 'Content-Type: application/cdmobject' -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)

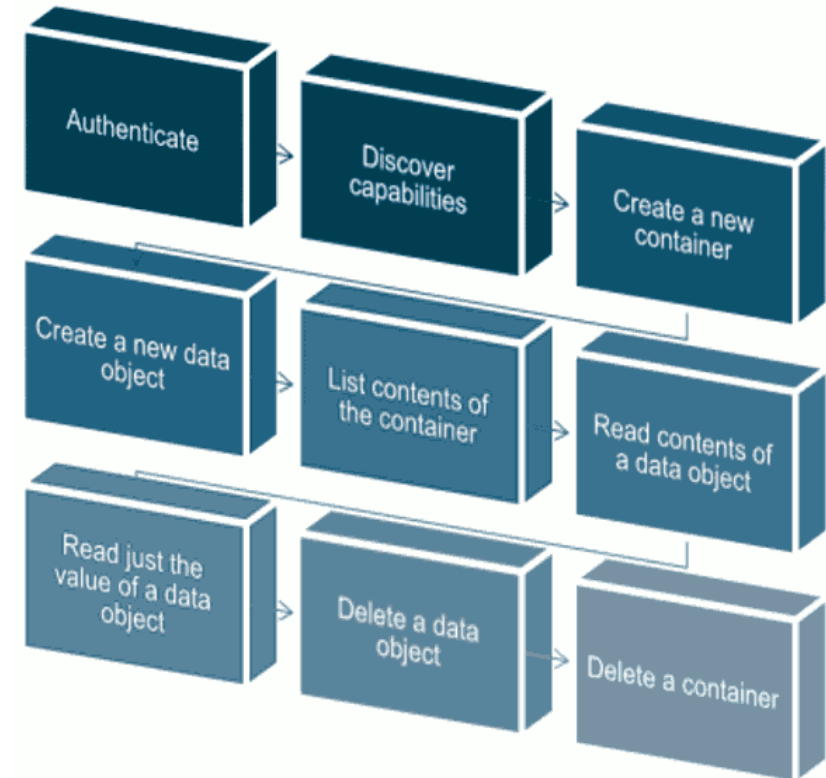
```
$ curl -v -X GET -H 'X-Auth-Token: 8197f75188304431bd3181ca79b0b884' -H 'Accept: text/plain' http://130.206.80.102:8080/cdmi/AUTH_d418851c6d294381bbe6e082849686d6/foo/text_doc
```

Delete object

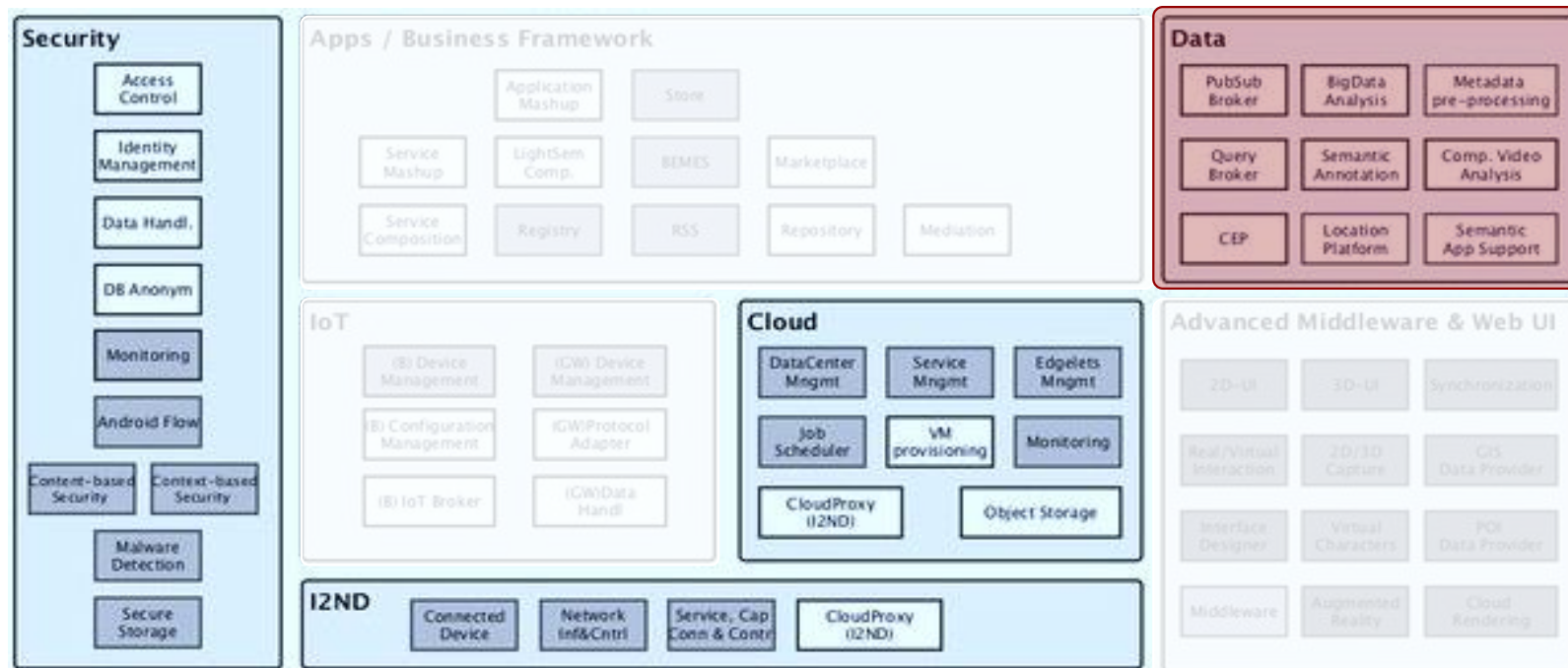
```
$ curl -v -X DELETE -H 'X-Auth-Token: 8197f75188304431bd3181ca79b0b884' http://130.206.80.102:8080/cdmi/AUTH_d418851c6d294381bbe6e082849686d6/foo/text_doc
```

Delete container

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



DATA/CONTEXT MANAGEMENT



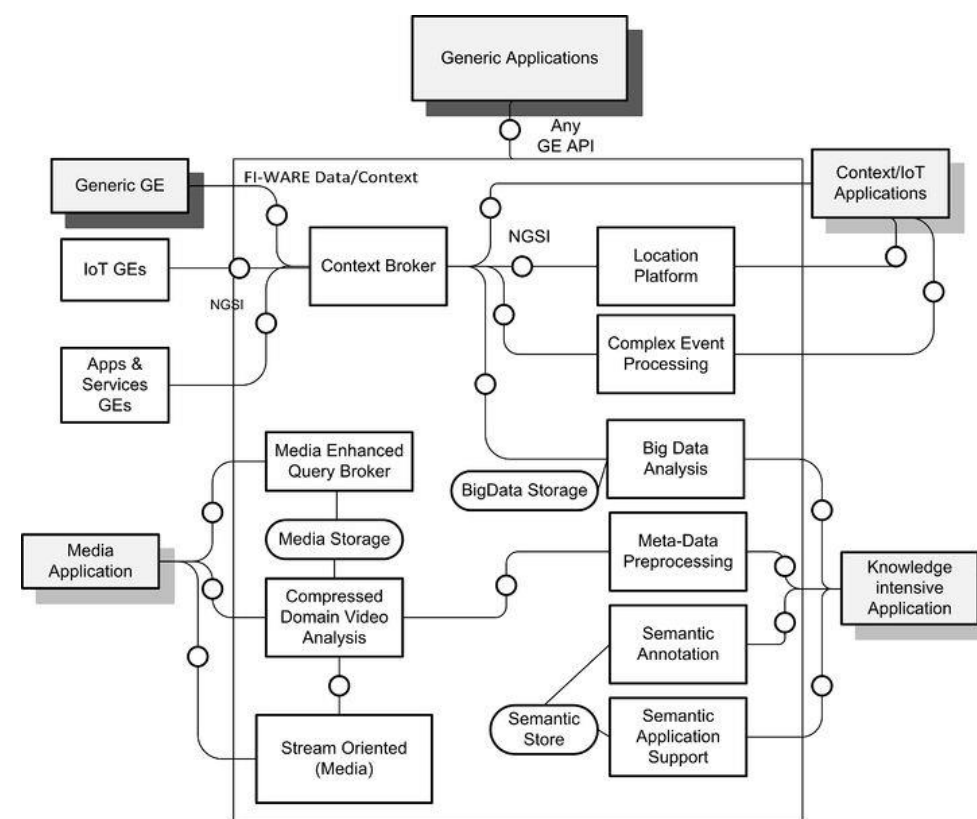
Architecture Overview

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.



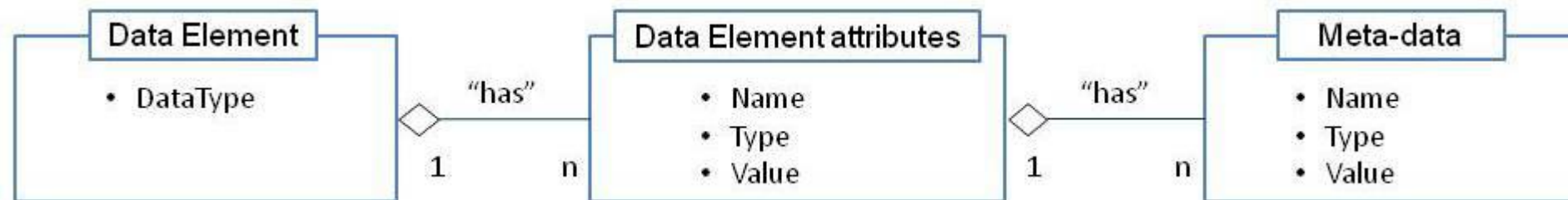
Data Elements

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 its concrete sequence of attributes.

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



Context Elements and Events

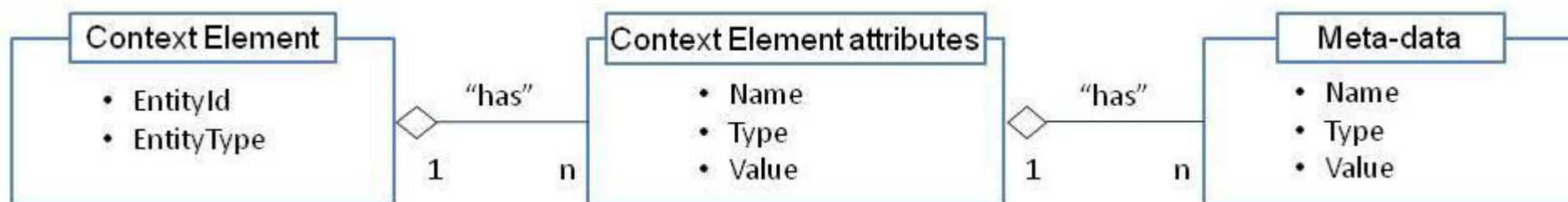
A **Context Element** extends the concept of data element by associating an **EntityId** and **EntityType** to it, uniquely identifying the entity in the FI-WARE system.

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**

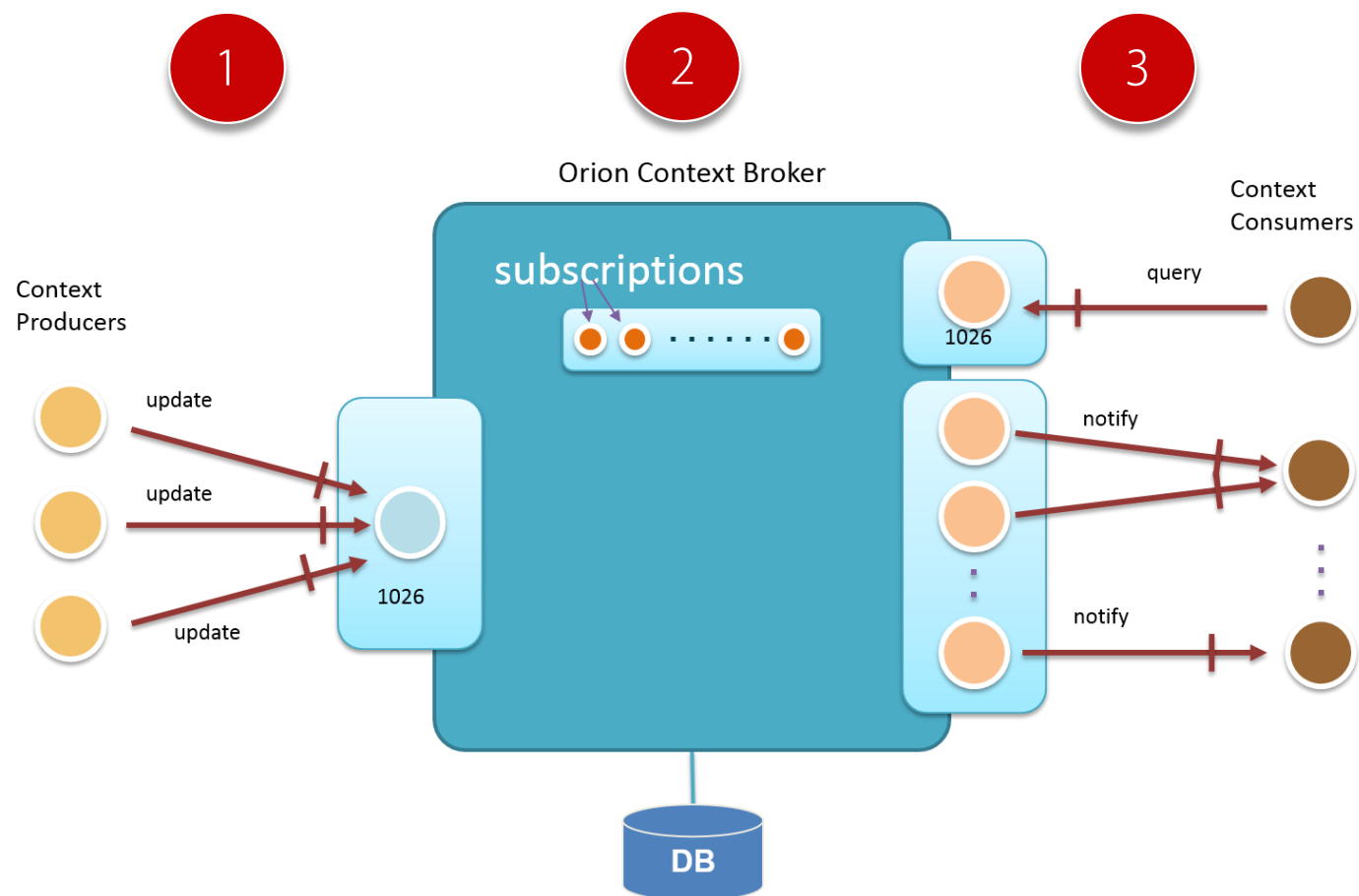
The **Orion Context Broker** is an implementation of the Publish/Subscribe Context Broker GE, providing the NGSi9 and NGSi10 interfaces.

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"
}
```

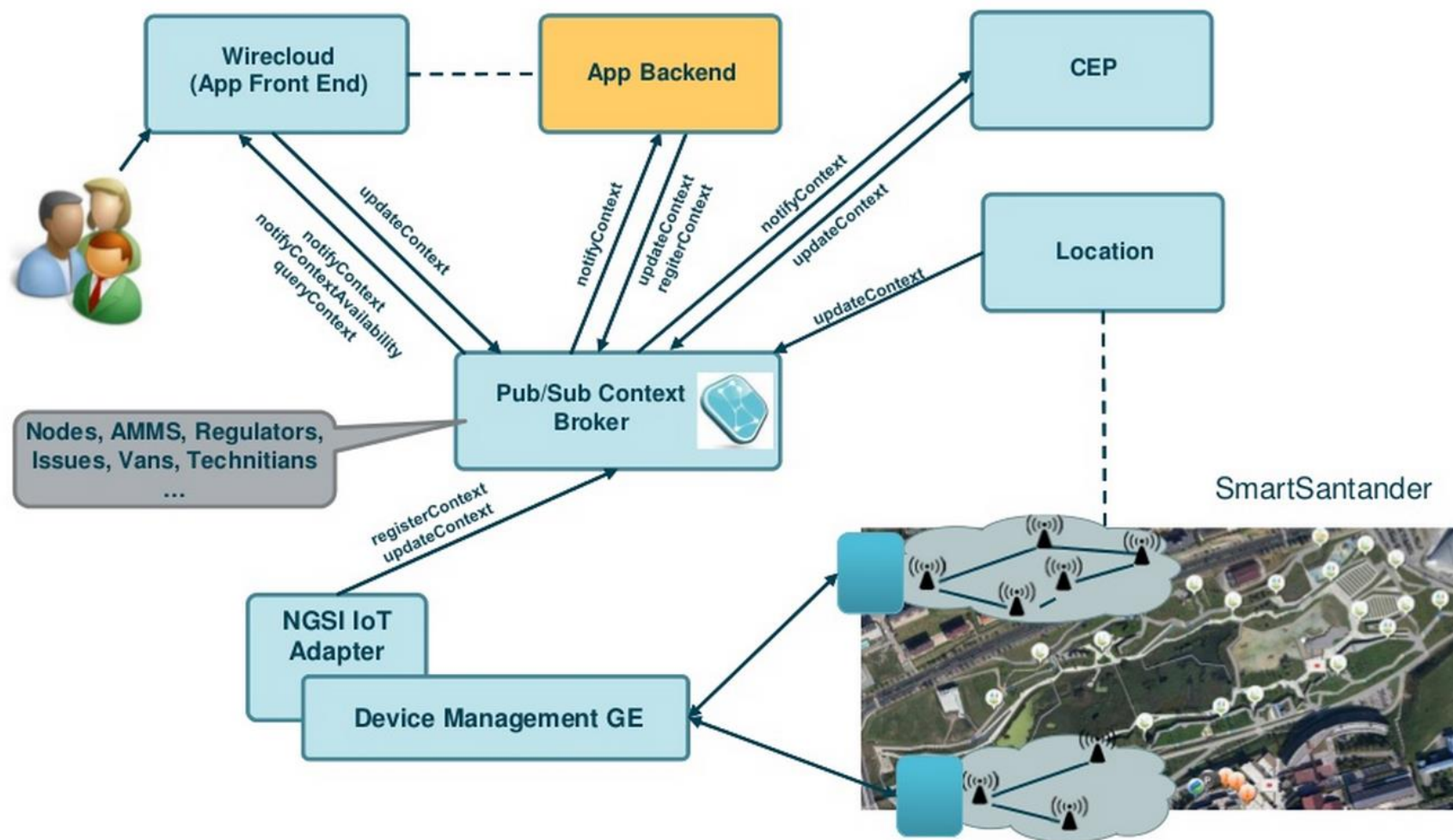
Metadata

```
...
"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 expressed 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%'");

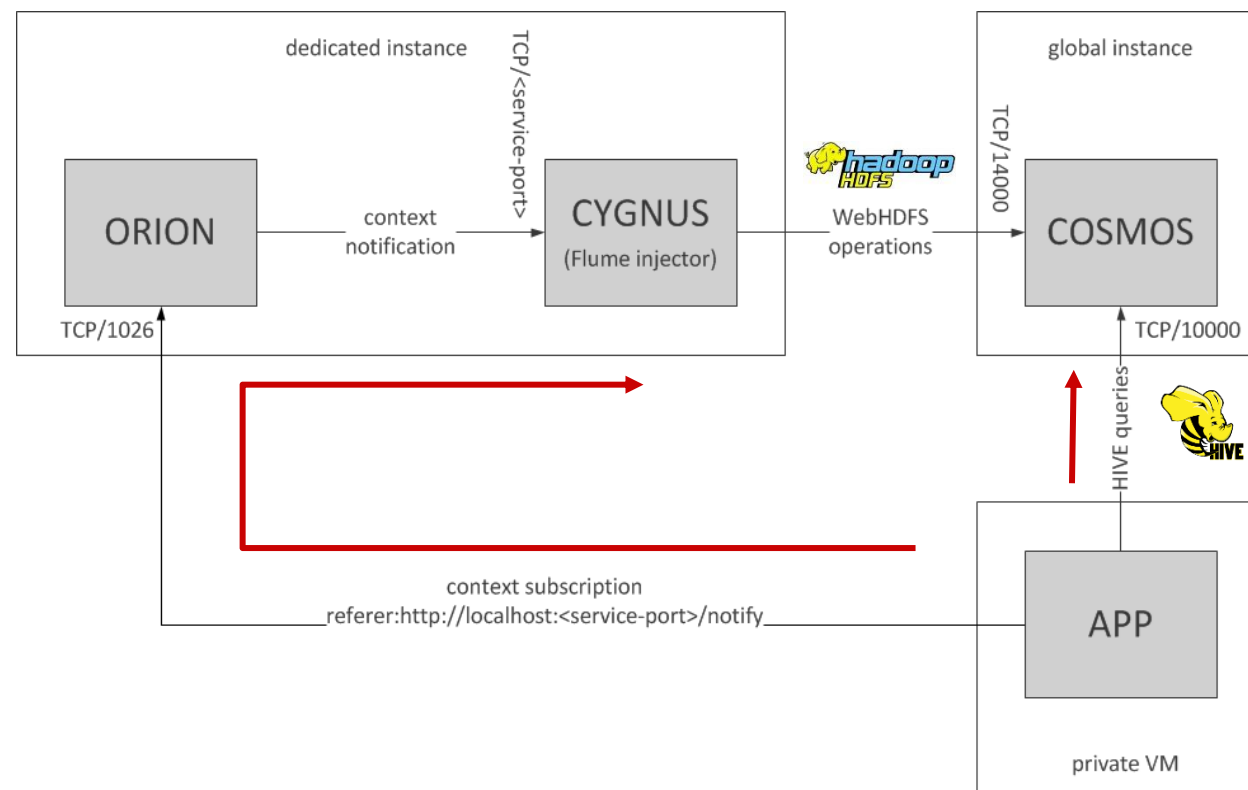
        // iterate on the result
        while (res.next()) {
            String column1 = res.getString(1);
            Integer column2 = res.getInteger(2);
            // whatever you want to do with this row, here
        } // while
        // close everything
        res.close(); stmt.close(); con.close();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



COSMOS – Cygnus

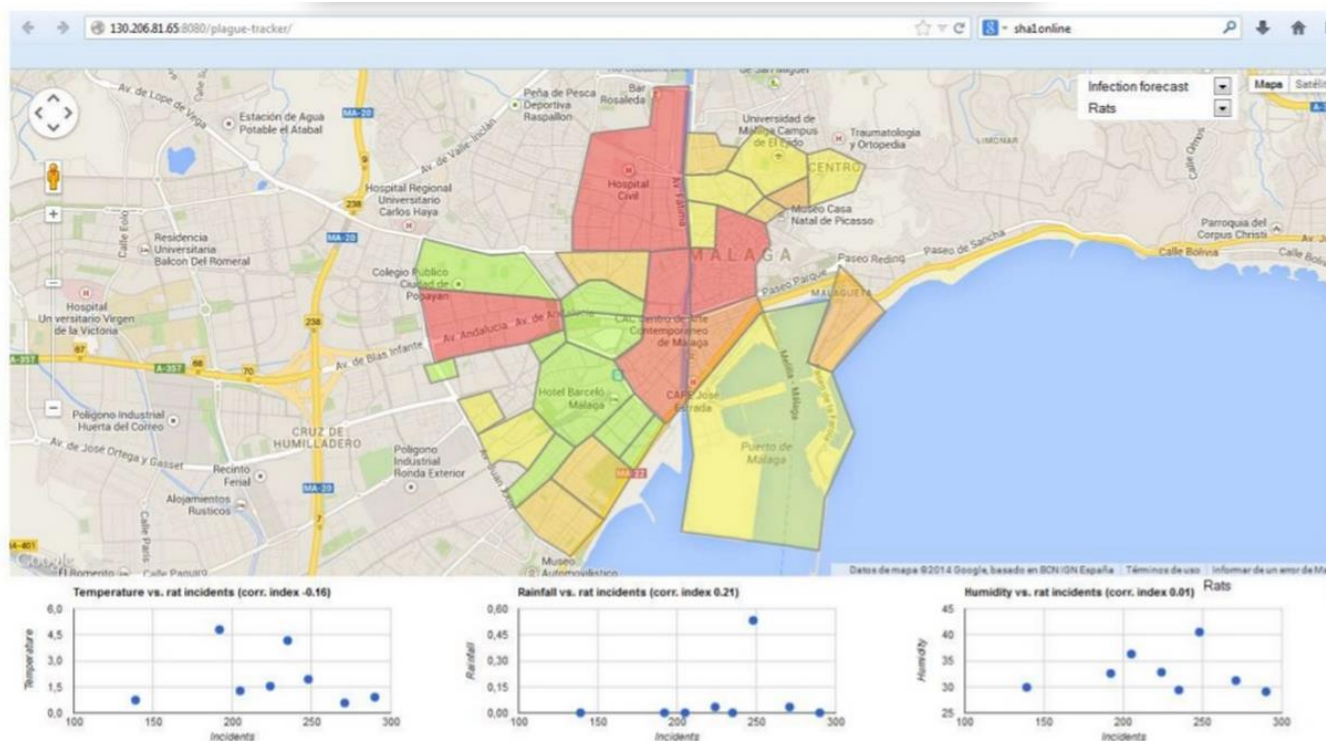
Cygnus offers integration between Context Broker and Cosmos Big Data. Context data is managed by Orion in last-value-storage, thus no historical data is stored. That historical storage could be possible if Orion writes all the updates in Cosmos.

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



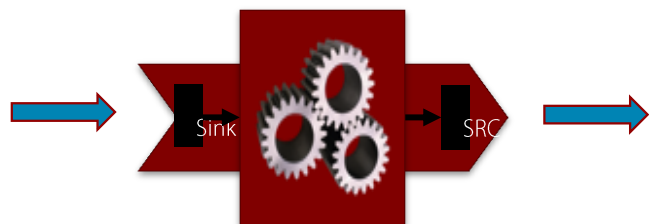
KURENTO
Stream oriented

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.

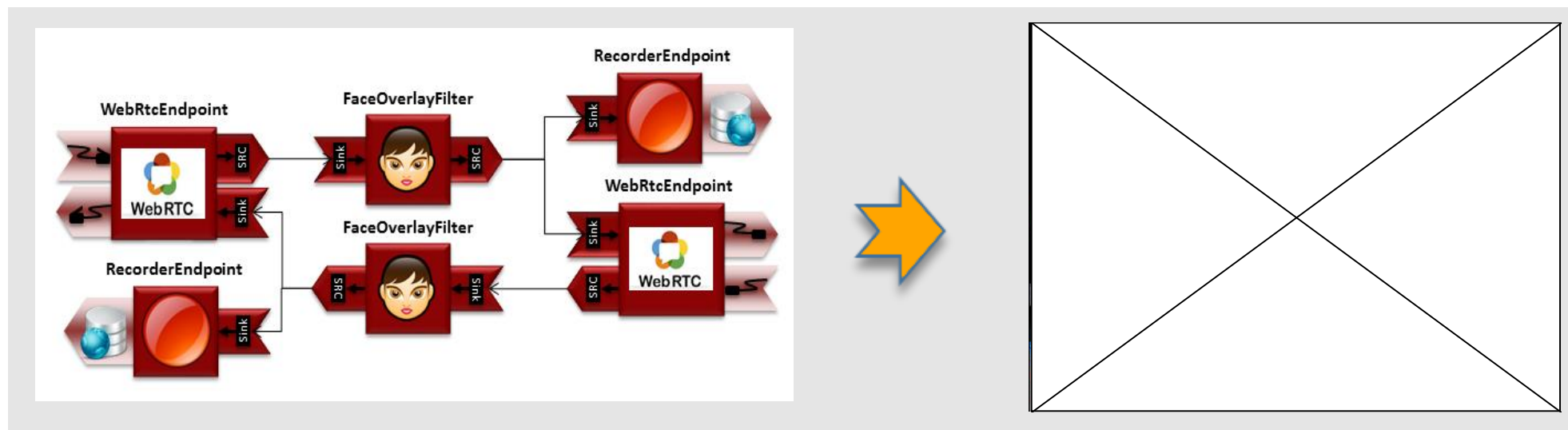
KURENTO – Example



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

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

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.

The screenshot shows the FIWARE Lab interface with the 'Data' tab selected. A red arrow labeled '1' points to the 'Data' tab, and another red arrow labeled '2' points to the 'Datasets' sub-tab. The main content area displays search results for 'Vigo', showing 10 datasets found. The left sidebar contains filters for Organizations, Groups, Tags, and Formats. A callout box labeled 'Datasets by tag' points to the 'Tags' section, and another callout box labeled 'Datasets by format' points to the 'Formats' section. A third callout box labeled 'Filter' points to the search input field where 'Vigo' is entered.

1 Data

2 Datasets

Search datasets...

/ Datasets

Organizations

Vigo (10)

Show More Organizations

Groups

There are no Groups that match this search

Tags

Vigo (8)

Tráfico (1)

Puerto (1)

Playas (1)

POI (1)

Gasolineras (1)

Farmacias (1)

Avisos (1)

Agenda (1)

Show More Tags

Formats

CSV (5)

rss (4)

rdf+xml (1)

Add Dataset

Vigo

10 datasets found for "Vigo"

Order by: Relevancia

Vigo: Avisos

Avisos al ciudadano del ayuntamiento de Vigo

rss

Vigo: Gasolineras

Inventario de gasolineras

CSV

Vigo: Farmacias

Inventario de las farmacias geolocalizadas del área de Vigo. También están disponibles las guardias para el año 2014.

CSV

Vigo: Playas

Información sobre el inventario de playas del área de Vigo y la calidad de sus aguas de baño.

CSV

Vigo: Avisos de tráfico

Avisos de tráfico proporcionados por el ayuntamiento de Vigo

Filter

CKAN – Dataset creation

The image illustrates the CKAN dataset creation workflow. It features three main components:

- 1. Create Dataset Form:** A multi-step form with three tabs: 'What are datasets?', 'Create dataset', and 'Additional info'. The 'Create dataset' tab is active, showing fields for Title, Description, Tags, License, Organization, Visibility, Searchable, and Allowed Users. A 'Next: Add Data' button is at the bottom.
- 2. Search Results:** A search results page for 'Vigo' showing 10 datasets. A red arrow points to the 'Add Dataset' button. The results list includes 'Vigo: Avisos' and 'Vigo: Gasolineras'.
- Create Resource Form:** A form for creating a resource, with tabs for 'What's a resource?', 'Create dataset', 'Add data', and 'Additional info'. The 'Add data' tab is active, showing fields for File (Upload/Link), Name, Description, and Format. A 'Next: Additional Info' button is at the bottom.

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.

URL: <https://data.lab.fi-ware.org/dataset/98593337-cb7f-409b-8ff9-78dcd011eb2e/resource/b3061e34-7a83-4c67-8864-f3918acd7714/doi...>

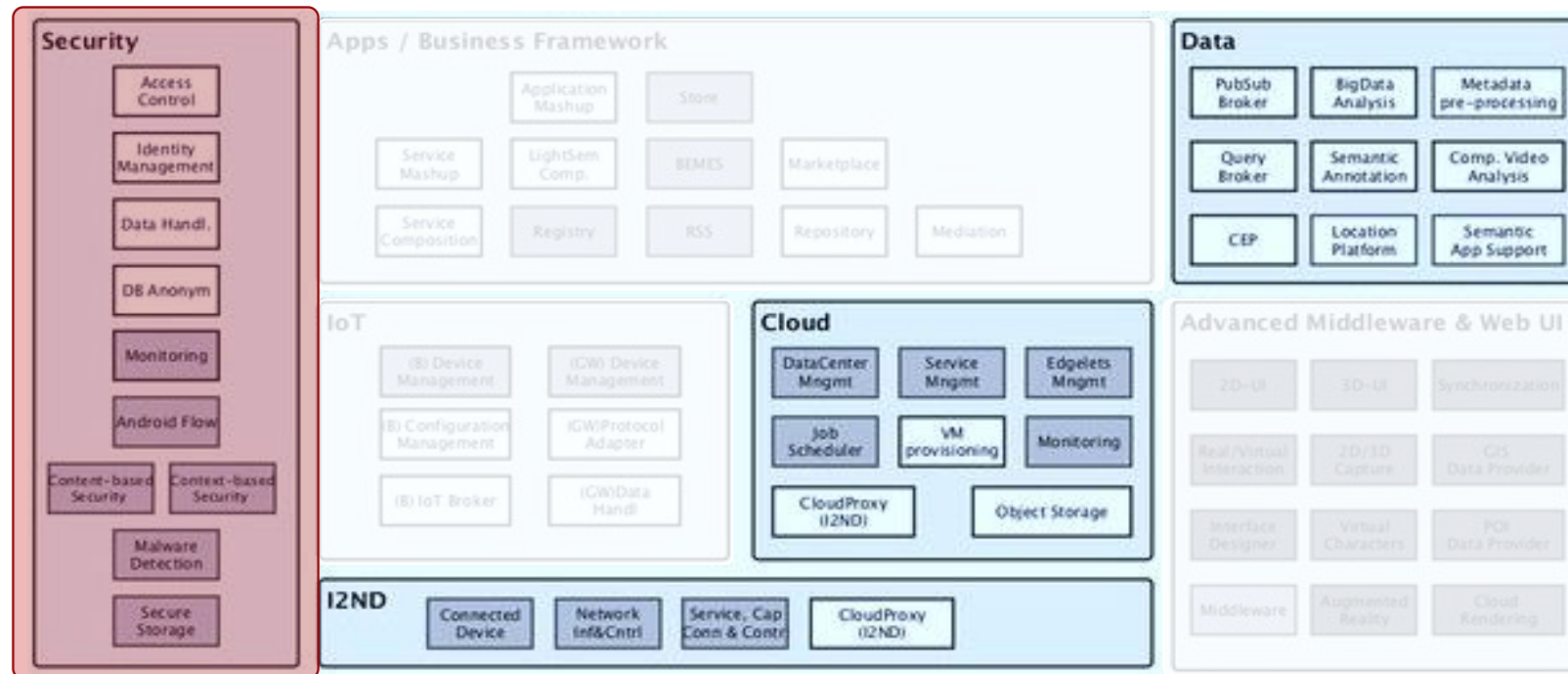
Inventario de playas del área de Vigo con su nombre, geolocalización, enlace a detalles proporcionados por el ayuntamiento y relación con los identificadores de calidad de agua (idZona) y previsiones meteorológicas (idMeteogalicia)

_id	id	nombre	latitud	longitud	url	idZona	idMeteog...
1	1	Playa de ...	42.264019	-8.67874	http://hox...	NULL	NULL
2	2	Playa de ...	42.26339...	-8.681344...	http://hox...	396	2085
3	3	Playa de ...	42.25725	-8.69259	http://hox...	NULL	2084
4	4	Playa de ...	42.257754	-8.694526	http://hox...	NULL	NULL
5	5	Playa de ...	42.26004...	-8.697502...	http://hox...	388	2082
6	6	Calo del ...	42.258956	-8.70324	http://hox...	NULL	NULL
7	7	Playa de ...	42.255452	-8.705439	http://hox...	NULL	NULL
8	8	Playa de ...					
9	9	Playa de ...					
10	10	Playa de ...					
11	11	Playa de ...					
12	12	Playa de ...					
13	13	Playa de ...					
14	14	Playa de ...					
15	15	Playa de ...					
16	16	Playa de ...					
17	17	Playa O ...					
18	18	Playa de ...					
19	19	Playa de ...					
20	20	Playa de ...					
21	21	Playa de ...					
22	22	Playa de ...					
23	23	Playa de ...					

Additional Information

Field	Value
Last updated	July 16, 2014
Created	July 16, 2014
Format	CSV
License	License Not Specified
can be previewed	1
created	4 months ago
datastore active	1
format	CSV
hash	ff87bdc96ecc72980a071c5867b8b1c
id	b3061e34-7a83-4c67-8864-f3918acd7714
on same domain	1
resource group id	cf431ac3-f7be-41f7-abe5-da996fbad33b
revision id	e75bf06d-2b41-4de7-81a8-e77f6d1cbce4
state	active
uri type	upload
webstore last updated	4 months ago
webstore uri	active

SECURITY

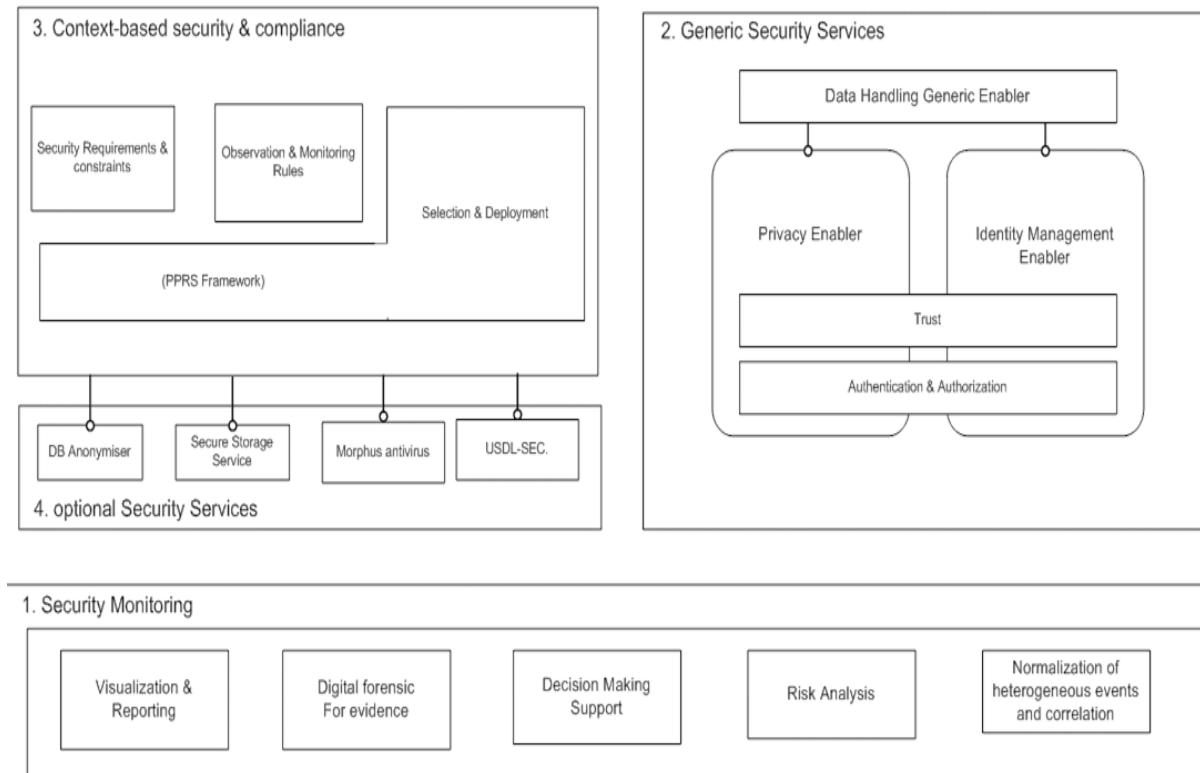


Architecture Overview

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



KeyRock Identity Management

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

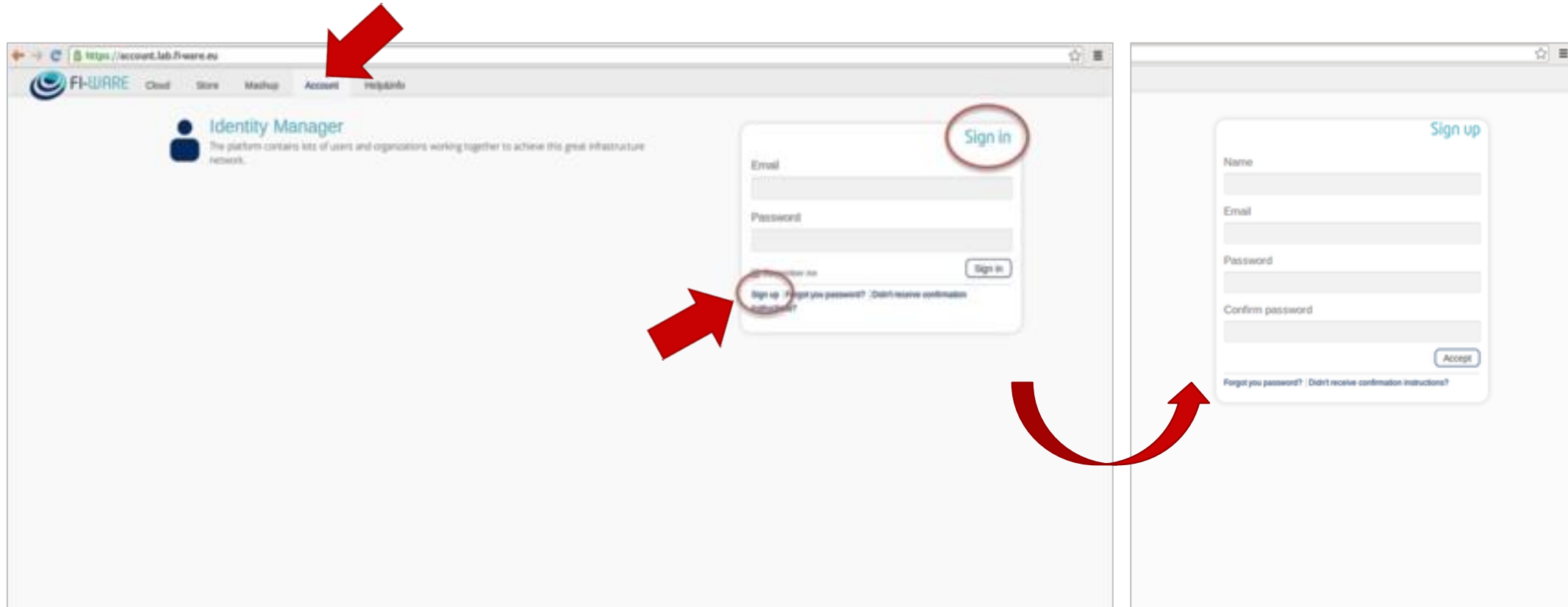
IdM include many aspects about users' access to networks, services or applications:

- 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

The screenshot shows the FIWARE Identity Manager interface. The top navigation bar includes links for Cloud, Store, Mashup, Account, and Help&info. The user 'Irena' is logged in. The left sidebar shows 'Home', 'Organizations', and 'My Applications'. The main content area is titled 'Organizations' and features a 'Create Organization' button. Below this button, there are tabs for 'Owned' and 'Others'. Under the 'Owned' tab, a list of organizations is shown, including 'GING' (Grupo de Internet de Nueva Generación, Universidad Politécnica de Madrid). A red arrow points from the 'Create Organization' button to the 'Organizations' link in the sidebar. Another red arrow points from the 'Create Organization' button to the 'Create an organization' form. A third red arrow points from the 'Create Organization' button to the 'Create Organization' button at the bottom of the form. The form itself has fields for 'Name' (MyOrganization), 'Owners' (Irena, Antonio Tapiador), and 'Description' (Test Organization).

Identity Manager

Cloud Store Mashup Account Help&info

Irena

Organizations

Create Organization

Owned Others

GING

Grupo de Internet de Nueva Generación, Universidad Politécnica de Madrid

Create an organization

Name

MyOrganization

Owners

Irena Antonio Tapiador

Description

Test Organization

Create Organization

IdM- Application Registering

The image displays two screenshots of the FI-WARE Identity Manager web interface, illustrating the application registration process.

Top Screenshot: The URL is <https://account.lab.fi-ware.eu/home>. The navigation bar includes links for Cloud, Store, Mashup, Account, and Help&Info. The main content area shows the 'Applications' section with a message: "You don't have any application. Would you like create your first application? Register Application". A red arrow points to the 'Register' button. The 'Organizations' section is also visible, showing a list of organizations.

Bottom Screenshot: The URL is <https://account.lab.fi-ware.eu/applications/new>. The navigation bar is the same. The main content area shows the 'Register Application' form. A dashed line with three numbered circles (1, 2, 3) indicates the registration steps. The form fields are:

- Name:** TestApplication
- Description:** test app
- URL:** <https://www.testapp.com>
- Callback URL:** <https://www.testapp.com>

A 'Next' button is located at the bottom right of the form. A 'Signed in successfully.' message is visible in the bottom left corner.

IdM- Application Roles and Permissions

Identity Manager

Cloud Store Mashup Account Help&info

Irena

1 2 3

Manage roles and permissions

Roles

- Provider
- Purchaser
- Client**

+ New role

Permissions in Client role

- ☒ Manage the application
- ☒ Manage roles
- ☐ Manage authorizations

+ New permission

Finish

Available roles

Create new role

Create new permission

Select permissions for each role

New permission definition

New permission

Name

Manage rules

Description

Permission to manage rules

HTTP verb

GET

Path

http://permissionpath.com

Create permission

IdM– Adding members and Managing roles

The screenshot shows the FIWARE Identity Manager interface. The sidebar on the left has a red arrow pointing to 'My Applications'. The main content area shows the 'TestApplication' configuration page, with the title 'TestApplication' circled in red. Below the application details, there is a section for 'Authorized' members, showing three users: Irena (2 roles), Antonio Tapiador (1 role), and Demo User (1 role). A red arrow points to the 'Add' button in the top right corner. A red arrow also points to the 'Add members' dialog box, which is open and shows a list of users and groups. A red arrow points to the 'Demo User' entry in the list, with a callout box saying 'Select member'. Another callout box says 'Add members to the application' pointing to the 'Add' button. A third callout box says 'Add role to a member' pointing to the 'Authorized' section. A fourth callout box says 'Authorized members' pointing to the list of users.

Authorized members

Add members to the application

Select member

Add role to a member

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: [{
    id: 1,
    name: "Manager"
  },],
  organizations: [
    {
      id: 1,
      actorId: 2,
      displayName: "Universidad Politecnica de
Madrid",
      roles: [
        {
          id: 14,
          name: "Admin"
        }
      ]
    }
  ]
} ... (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



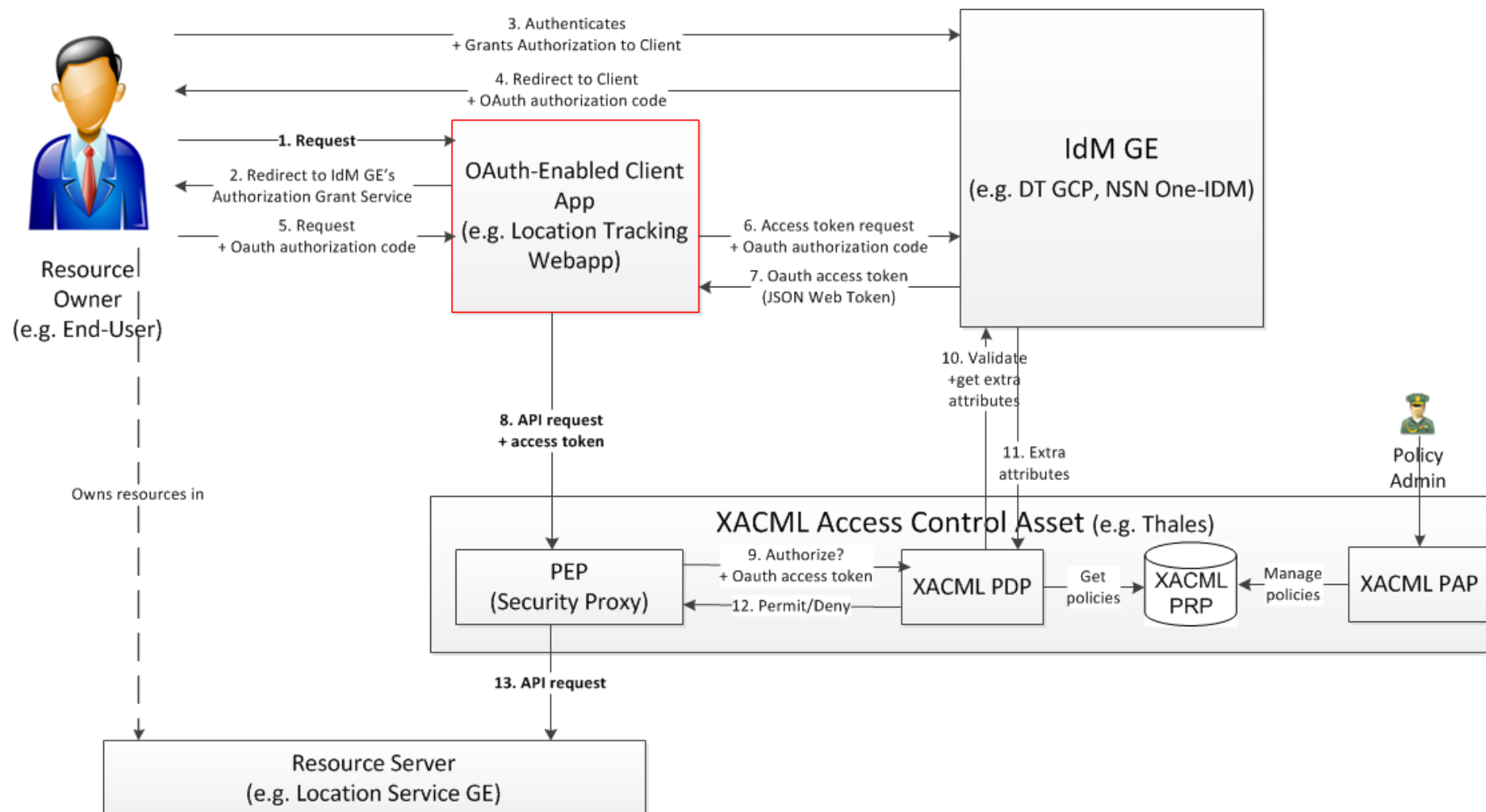
Access Control GE THALES Implementation

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.
```

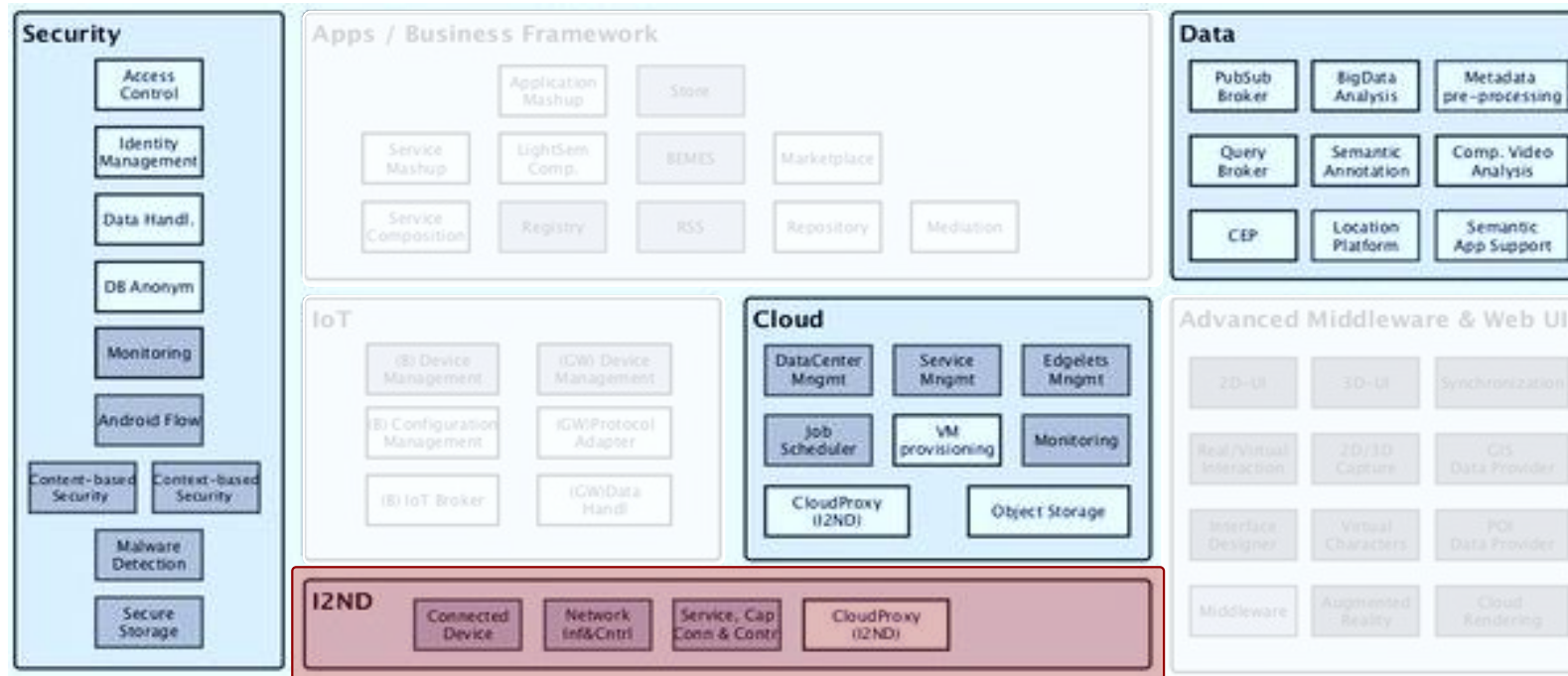
Policy Decision Point (PDP)

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

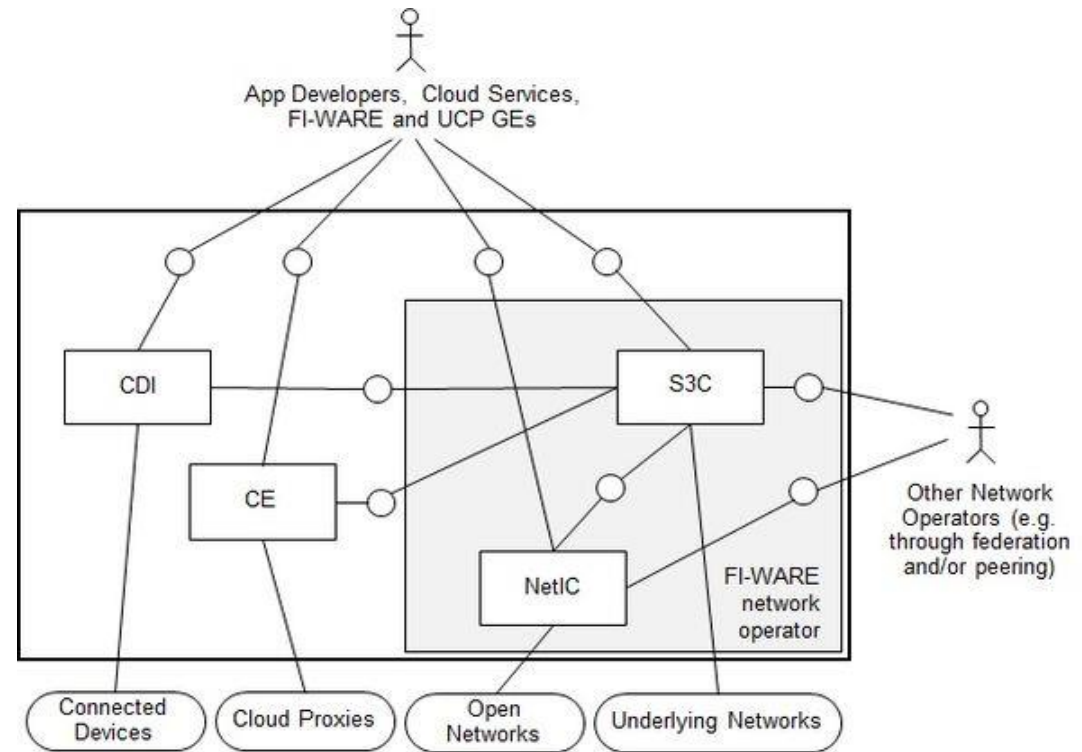


Architecture Overview

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

The screenshot displays the OFNIC web interface, which is used for network information and control. The interface is divided into several sections:

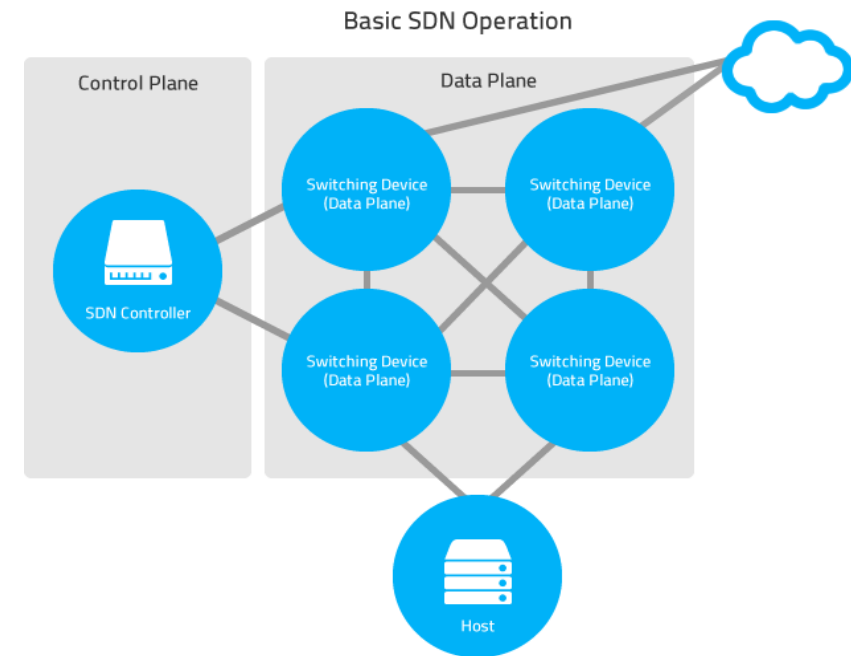
- Top Navigation Bar:** Contains tabs for Synchronization, Network Statistics, and Routing. The 'Network Statistics' tab is currently selected.
- Server Path:** A text input field containing 'https://localhost' and a 'submit' button.
- Topology Information:** A section with a 'Select a Node from the graph or from the menu below' prompt and a 'Select Node' button. Below this is a table with columns for Node ID, Buffers, Actions, and Tables.
- Node Information:** A section titled 'Information about node 2' showing the following data:
 - Num Buffers: 256
 - Num Tables: 255
 - Actions: 4095
- Ports of node 2:** A section with buttons for 'eth3', 'eth2', 'br0', and 'eth1'. The 'eth1' button is currently selected.
- Statistics about port eth1:** A section showing the following data:
 - Tx_bytes: 27
 - Rx_bytes: 20
- Network Graph:** A visual representation of the network topology. It shows three nodes (1, 2, and 3) connected in a triangle. Node 1 is connected to Node 2 and Node 3. Node 2 is connected to Node 1 and Node 3. Node 3 is connected to Node 1 and Node 2. The nodes are labeled 'host1', 'host2', and 'host3' respectively.

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.fiware.org/plugins/mediawiki/wiki/fiware/index.php/Security_Architecture

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

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

https://forge.fiware.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/flopezaguiar/setting-up-your-virtual-infrastructure-using-fi-lab-cloud-32388357?related=2>

<http://www.slideshare.net/flopezaguiar/idm-and-ac?related=3>

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

I2ND

[https://forge.fiware.org/plugins/mediawiki/wiki/fiware/index.php/Interface_to_Networks_and_Devices_\(I2ND\)_Architecture](https://forge.fiware.org/plugins/mediawiki/wiki/fiware/index.php/Interface_to_Networks_and_Devices_(I2ND)_Architecture)

<https://forge.fiware.org/plugins/mediawiki/wiki/fiware/index.php/FIWARE.ArchitectureDescription.I2ND.NetIC>

<https://forge.fiware.org/plugins/mediawiki/wiki/fiware/index.php/FIWARE.ArchitectureDescription.I2ND.CDI>

<https://forge.fiware.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

Gradiant

Galician research and development center
in advanced telecommunications

2015