

Tema 4: Sistemas de adquisición y gestión de datos

Arquitectura y uso de FIWARE

Curso 2023 - 2024

Profesor:

Juan Antonio Martínez (juanantonio@um.es)



ÍNDICE



- 1. Introducción**
- 2. Arquitectura FIWARE**
- 3. Publish/Subscribe Context Broker – Orion Context Broker**
- 4. BigData Analysis – Cosmos**
- 5. Application Mashup – Wirecloud**
- 6. Portal FI-LAB**

1. INTRODUCCIÓN

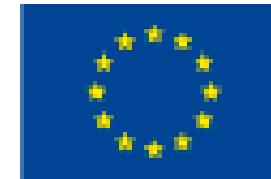


FIWARE es una plataforma, impulsada por la Unión Europea, para el desarrollo y despliegue global de aplicaciones de Internet del Futuro. FIWARE intenta proveer de una arquitectura totalmente abierta, pública y libre así como de un conjunto de especificaciones que permita a los desarrolladores, proveedores de servicios, empresas y otras organizaciones desarrollar productos que satisfagan sus necesidades

- La **Comisión Europea** (CE) y las principales **empresas TIC europeas** emprendieron en 2011 un programa de Colaboración Público-Privada (*Public Private Partnership – PPP*), con el objetivo de definir una plataforma que represente una opción abierta para el desarrollo y despliegue global de aplicaciones en la Internet del Futuro.
- **Telefónica** lidera el desarrollo de la plataforma **FIWARE**.
- Las especificaciones de las APIs (*Application Programming Interfaces*) ofrecidas por los componentes de esta plataforma son abiertas y libres de royalties.

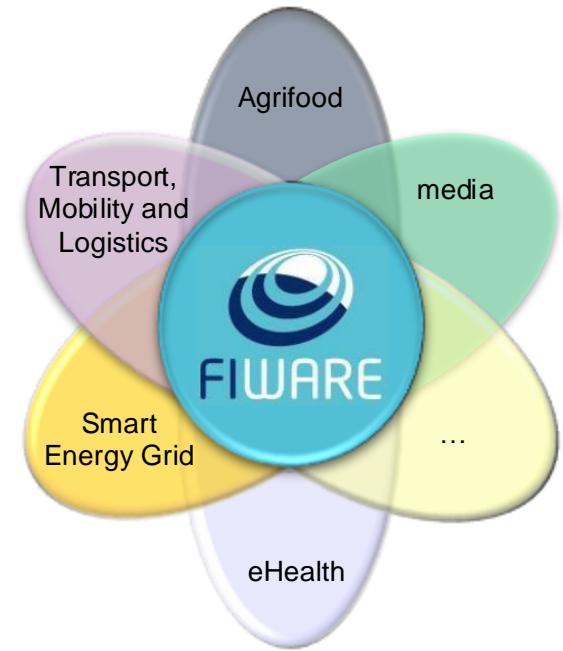


- Plataforma Europea
- Totalmente gratuita
- Gran potencial
- Fomenta emprendimiento



1. INTRODUCCIÓN

- **Objetivo:** Aprovechar las oportunidades derivadas de la era de la digitalización de la vida y los negocios.
- **Estrategia:** Construir un ecosistema que funcione como catalizador para la captura de las oportunidades. Hacia una estandarización en Smart Cities e IoT.
- **Pilares:**
 -  **FIWARE** : Ofrece un conjunto amplio de **APIs** (Interfaces de aplicaciones) que facilitan el **desarrollo de aplicaciones** en varios sectores.
 -  **FIWARE Lab** : entorno donde se pueden probar todas las aplicaciones basadas en tecnologías FIWARE. Es un **entorno de prueba** y no comercial.
 -  **FIWARE Accelerate** : un **programa de apoyo** a desarrolladores y emprendedores, con especial atención a las PYMEs y empresas de nueva creación
 -  **FIWARE Ops** : herramientas que facilita el despliegue, configuración y operación de instancias FIWARE por los proveedores de la plataforma.
 -  **FIWARE Mundus** : programa creado para extender la tecnología FIWARE más allá de Europa incluyendo América Latina, África y Asia.



1. INTRODUCCIÓN

**FI-WARE = advanced OpenStack-based Cloud +
rich library of Generic Enablers +
developers tools**



1. INTRODUCCIÓN

SMART Water



SMART Agrifood



SMART City



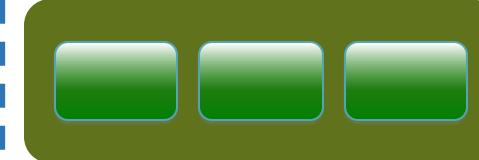
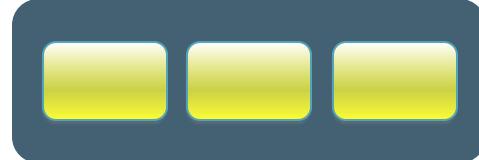
SMART Energy



SMART Industry



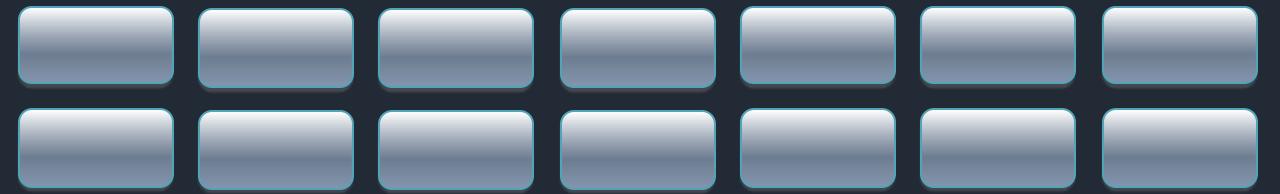
Domain specific enablers



FIWARE

GEs

{



1. INTRODUCCIÓN

Se denominan **Generic Enablers (GEs)** a los diferentes servicios disponibles en su catálogo. Posee un entorno web de Cloud donde es posible enlazar unas aplicaciones o widget con otras, hasta conseguir otras nuevas de mayor complejidad sin tener que crearlas desde cero.



- ✓ Proporciona herramientas para **desarrollar aplicaciones**
- ✓ Formada por módulos independientes. **Generic Enablers (GEs)**
- ✓ Todas las APIs y muchos elementos se basan en soluciones Open Software

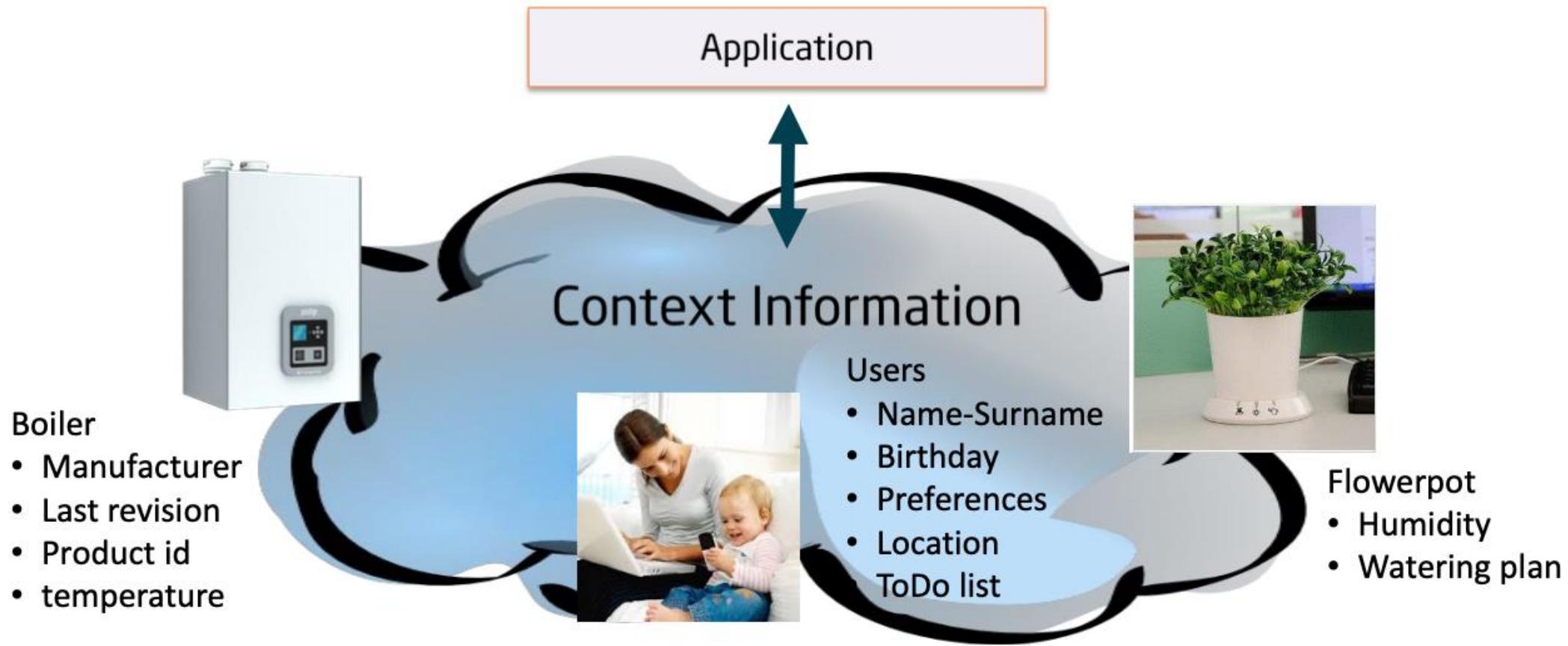
1. INTRODUCCIÓN

Los **Generic Enablers** son divididos en las siguientes categorías:

- Core Context Management
 - The core and mandatory component of any “Powered by FIWARE” platform or solution. It enables to manage context information in a highly decentralized and large-scale manner.
- Interface with IoT, Robots and Third-party Systems
 - Making it easier to interface with the Internet of Things, Robots and Third-party systems for the purpose of gathering valuable context information or trigger actuations in response to context updates:
- Context Processing, Analysis and Visualisation
 - Making it easier to process, analyse or visualize context information for the purpose of implementing the “smart behaviour” expected in any application
- Context Data/API Management, Publication and Monetization
 - Implement secured access to components in the architecture of any “Powered by FIWARE” solution using Generic Enablers of the Security chapter
- Deployment Tools
- External Third-party Components

1. INTRODUCCIÓN

Ser “inteligente” requiere ser primero “consciente” → Información Contextual



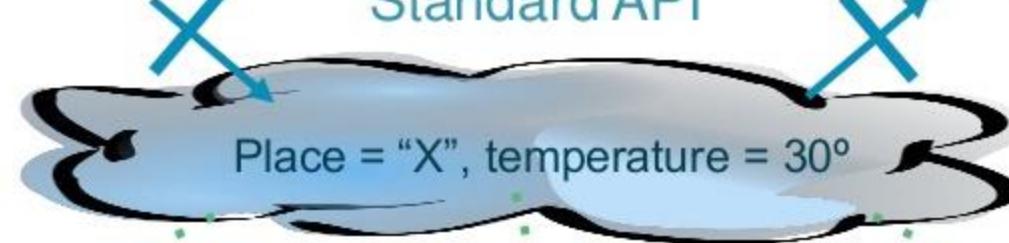
1. INTRODUCCIÓN

What's the current temperature in place "X"?



A sensor in a pedestrian street

Standard API



It's too hot!



A person from his smartphone

Notify me the changes of temperature in place "X"

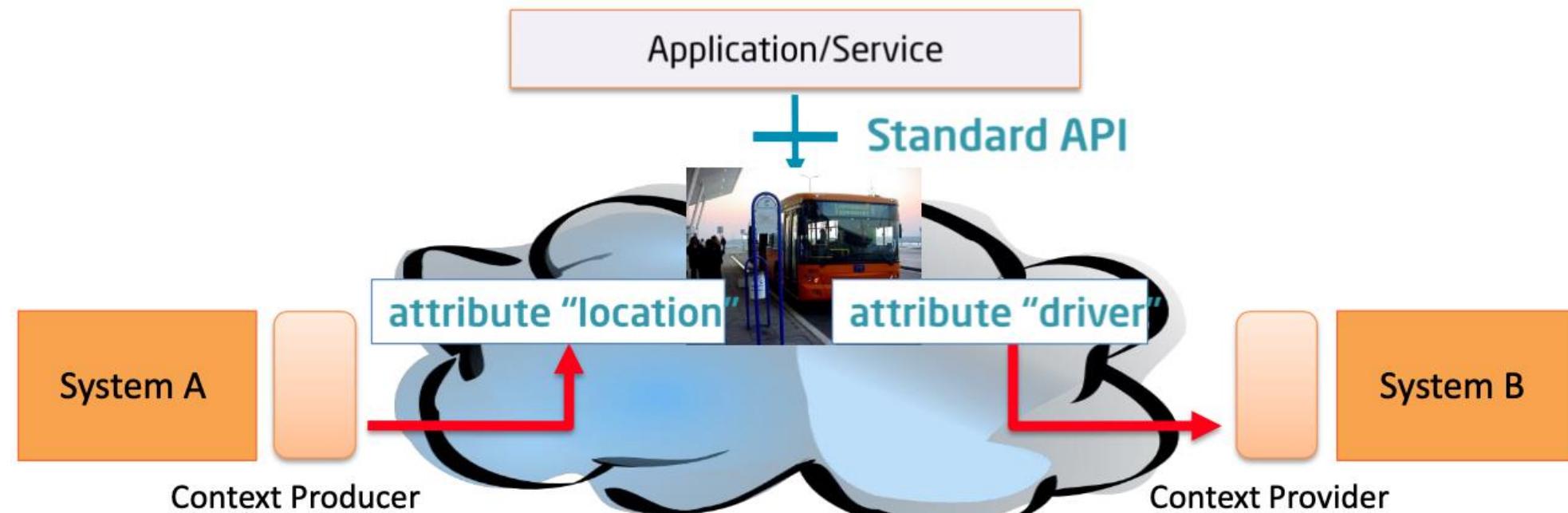


The Public Bus Transport Management system

1. INTRODUCCIÓN

Es necesario un enfoque no intrusivo

- Integrador
- Información puede venir de cualquier sistema
- Único modelo

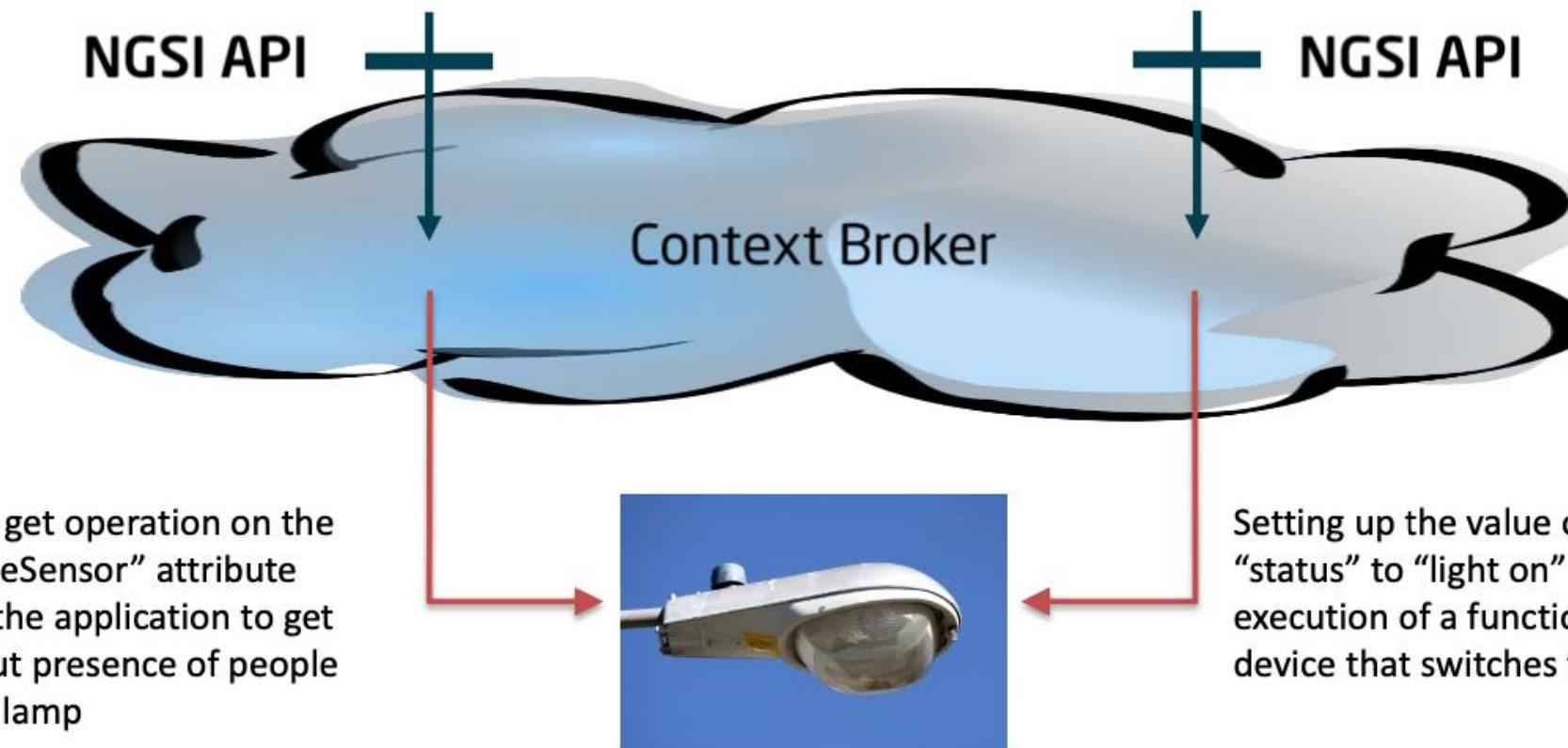


1. INTRODUCCIÓN

Lectura y escritura en dispositivos IoT tan simple como leer y escribir atributos

GET <Oauth token>
 /v2/entities/lamp1/attrs/presenceSensor

PUT <Oauth token>
 /v2/entities/lamp1/attrs/status/value
 "light on"

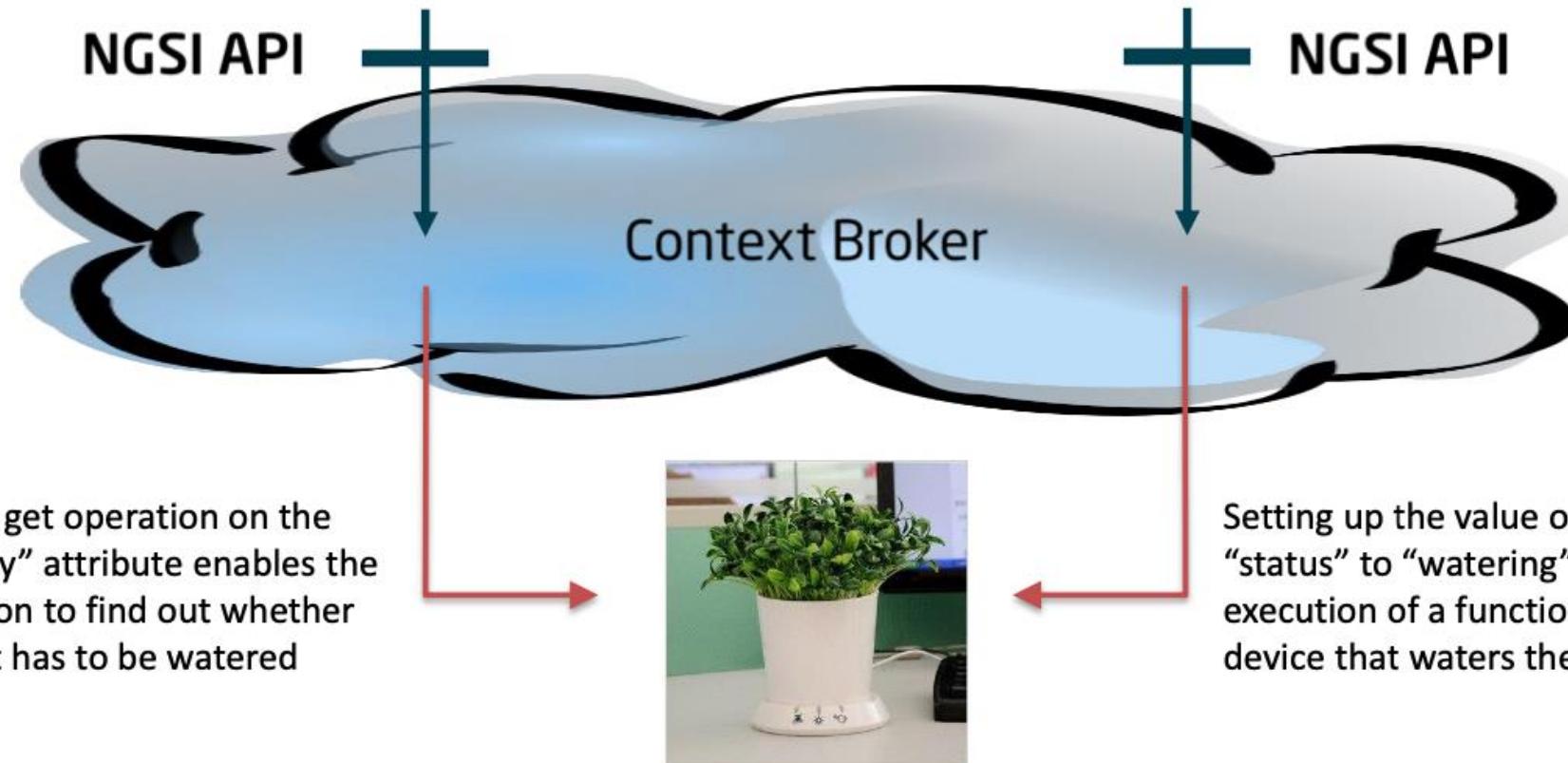


1. INTRODUCCIÓN

Lectura y escritura en dispositivos IoT tan simple como leer y escribir atributos

GET <Oauth token>
 /v2/entities/lamp1/attrs/humidity

PUT <Oauth token>
 /v2/entities/lamp1/attrs/status/value
 "watering"



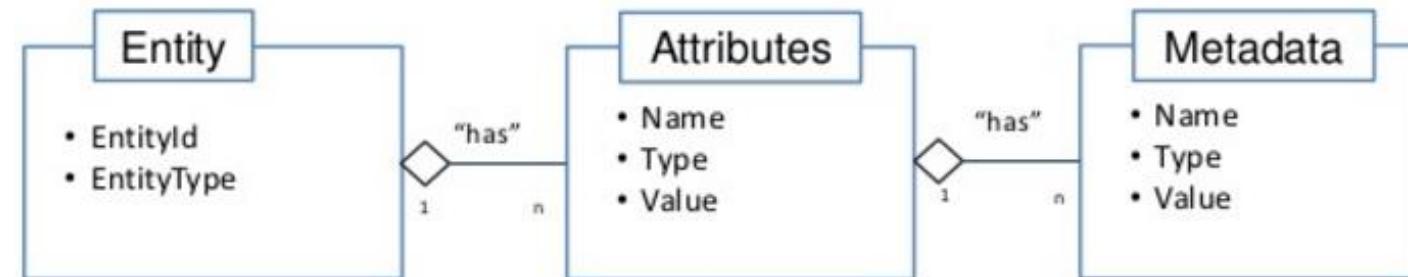
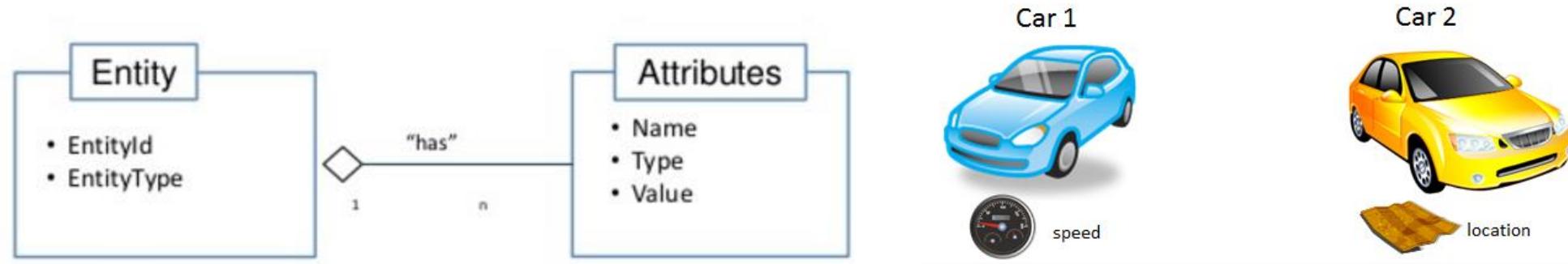
FIWARE

- Las aplicaciones de IoT necesitan recoger y gestionar información que describa el “estado” actual de todo lo que está sucediendo a su alrededor (contexto), como es la última información reportada por los sensores o aquella capturada por cualquier otro sistema externo.

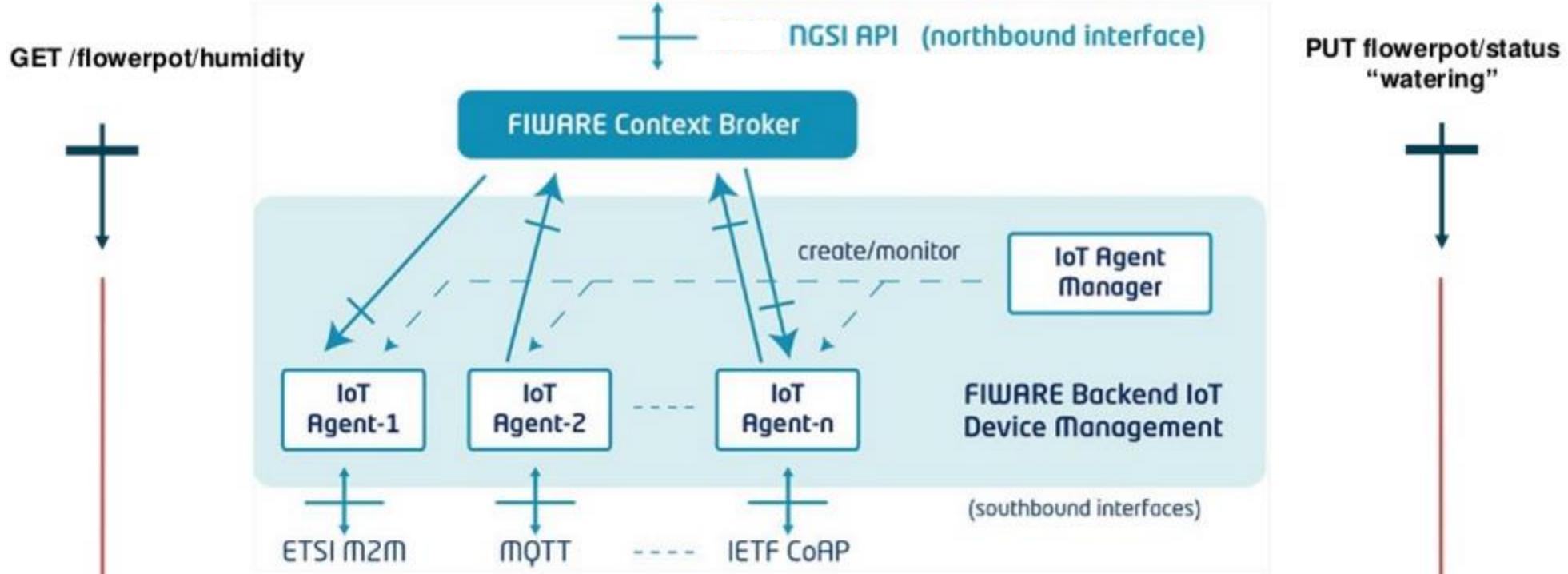


FIWARE

- FI-WARE estandariza el **modelo de datos (entidades, atributos y metadatos)**
- API **FIWARE NGSI (Next Generation Services Interface)**



FIWARE



La lectura del atributo
“humidity” permite
conocer si la planta
necesita ser regada



Modificando el atributo
“status” se consigue
actuar sobre el
dispositivo para que
riegue la planta

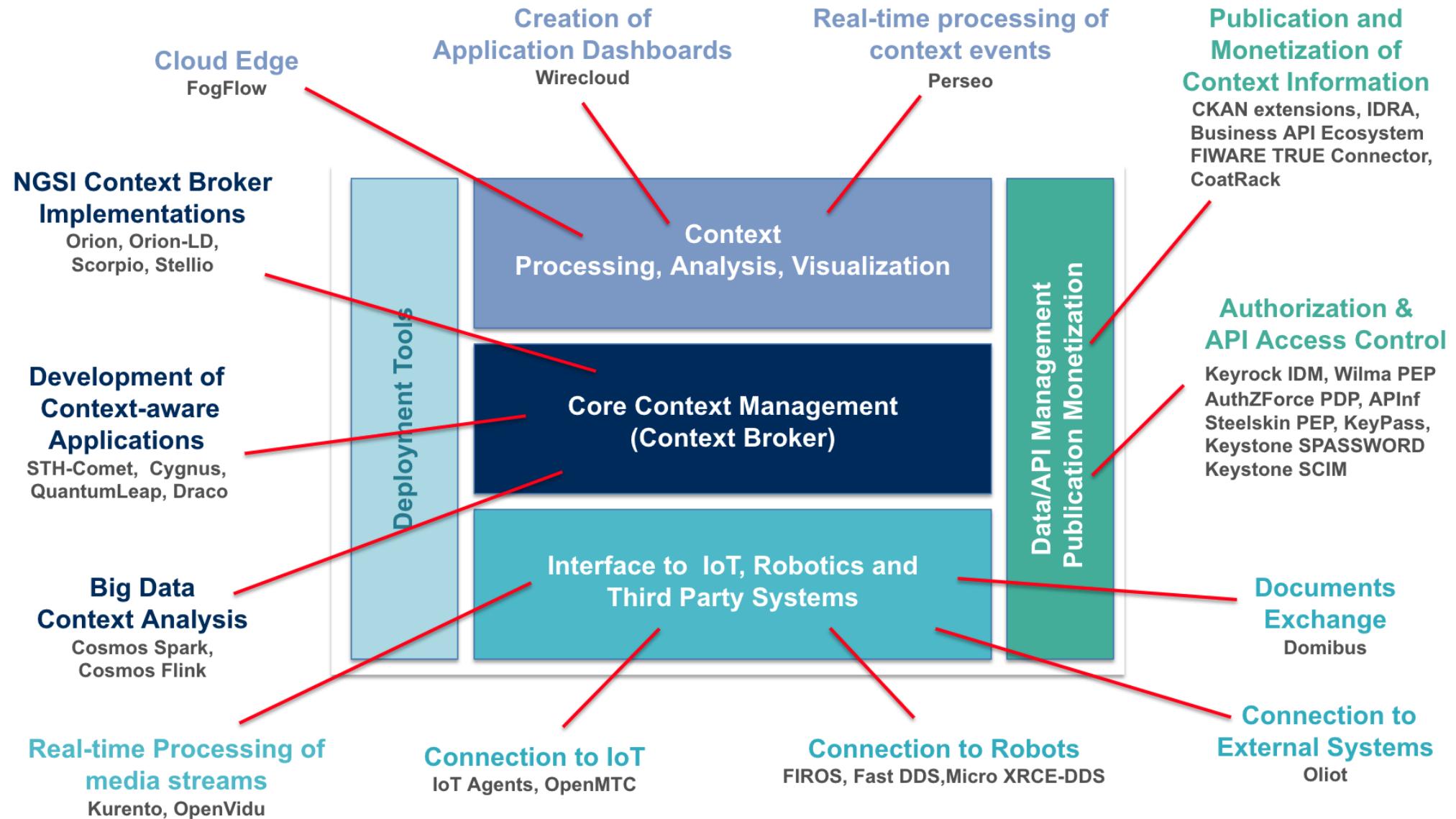
*Recoger datos o actuar sobre dispositivos IoT es tan fácil como
suscribirse/leer/cambiar los valores de los atributos asociados al
contexto*

ÍNDICE

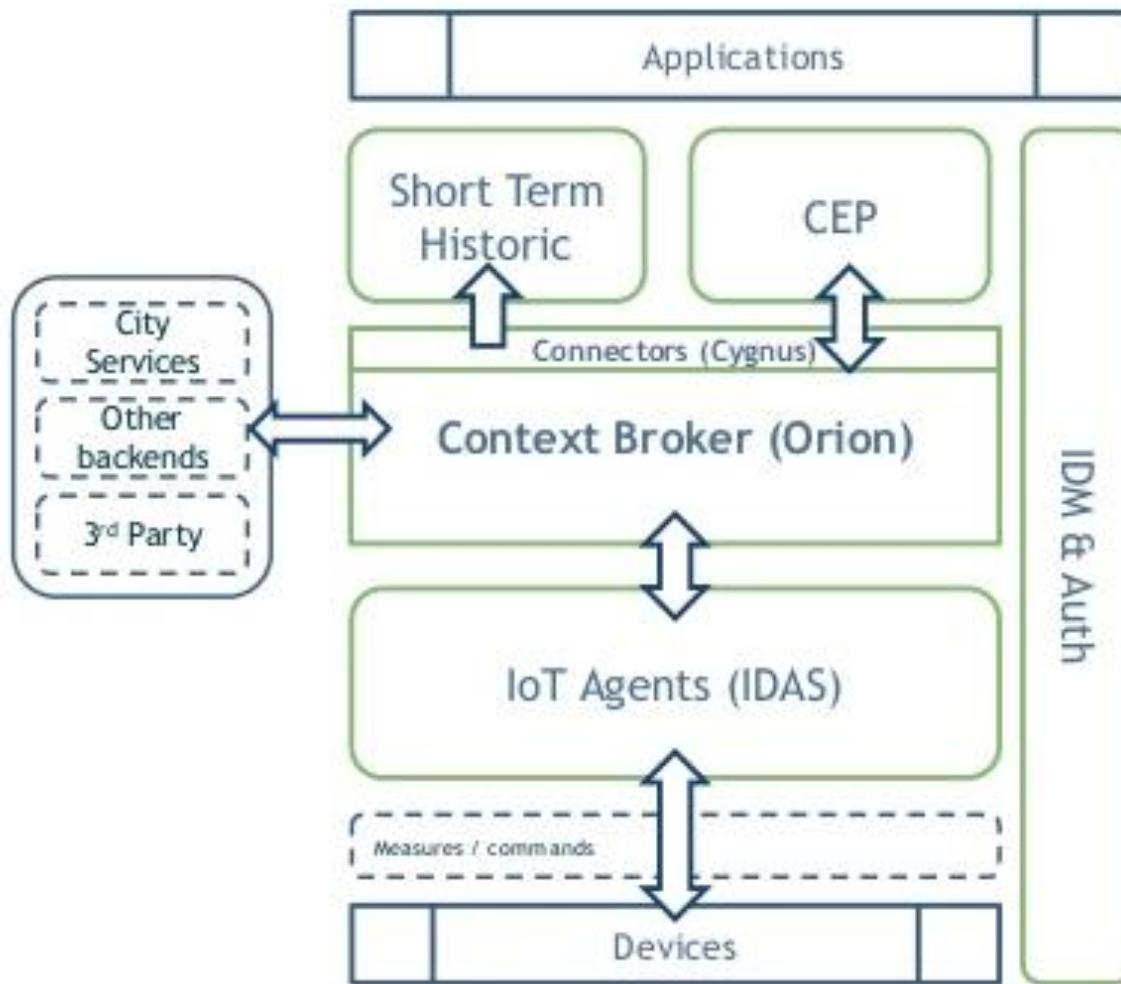
- 1. Introducción**
- 2. Arquitectura FIWARE**
- 3. Publish/Subscribe Context Broker – Orion Context Broker**
- 4. BigData Analysis – Cosmos**
- 5. Application Mashup – Wirecloud**
- 6. Portal FI-LAB**



Arquitectura básica de FI-WARE para IoT

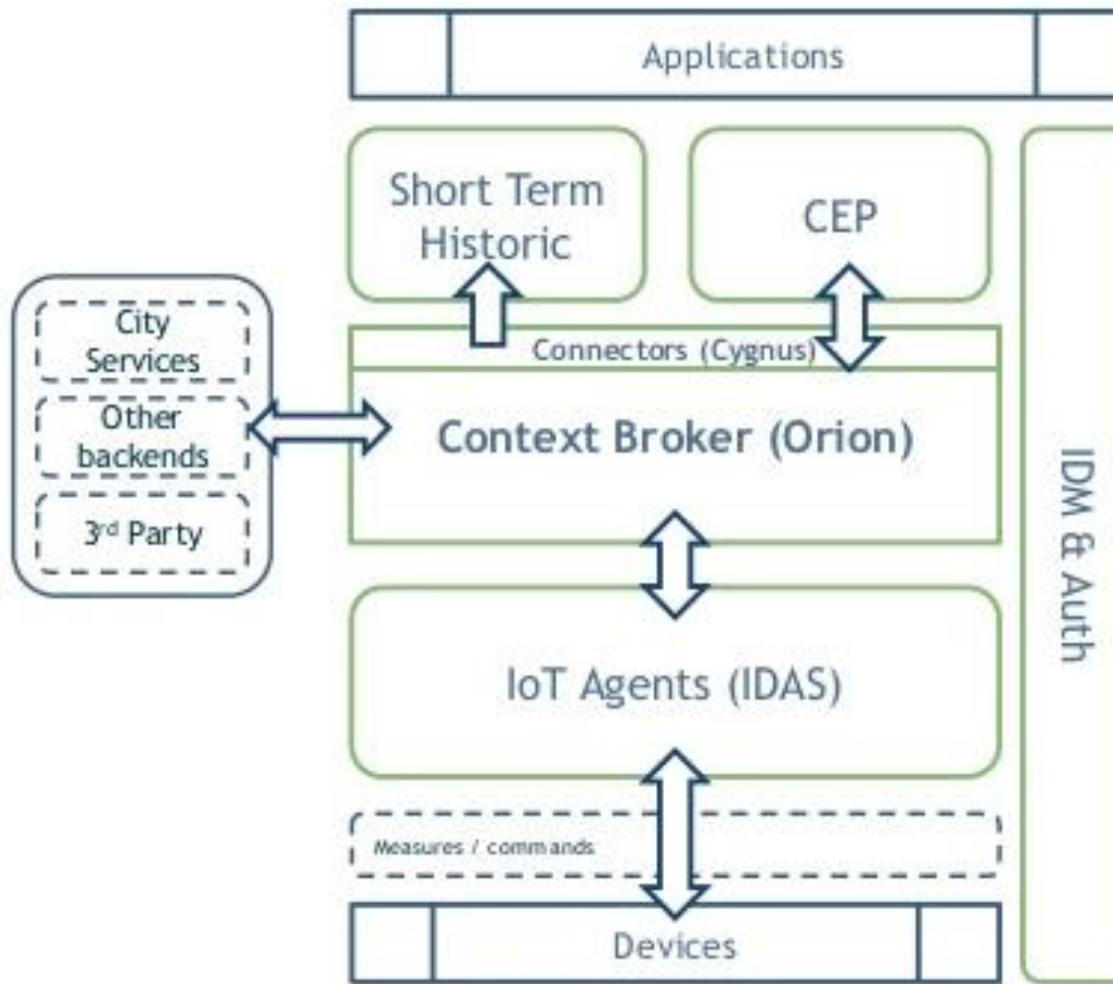


Arquitectura básica de FI-WARE para IoT



1. Los **dispositivos físicos** actúan como fuentes de datos en la arquitectura.
2. **IoT Agents** es la puerta de entrada para que los dispositivos puedan comunicarse con el resto de los componentes de la arquitectura. Soporta diversos protocolos específicos (CoAP, MQTT, etc.) para recibir datos de sensores y se conecta al Context Broker el cual los almacena y los gestiona.
3. El **Context Broker** mantiene representaciones virtuales de los dispositivos físicos y mantiene el último dato entregado por los dispositivos. Se interconecta con otras plataformas.
4. Los componentes de gestión de **identidad y autorización** bridan los mecanismos para garantizar la confiabilidad, seguridad y privacidad en la entrega y uso de servicios.

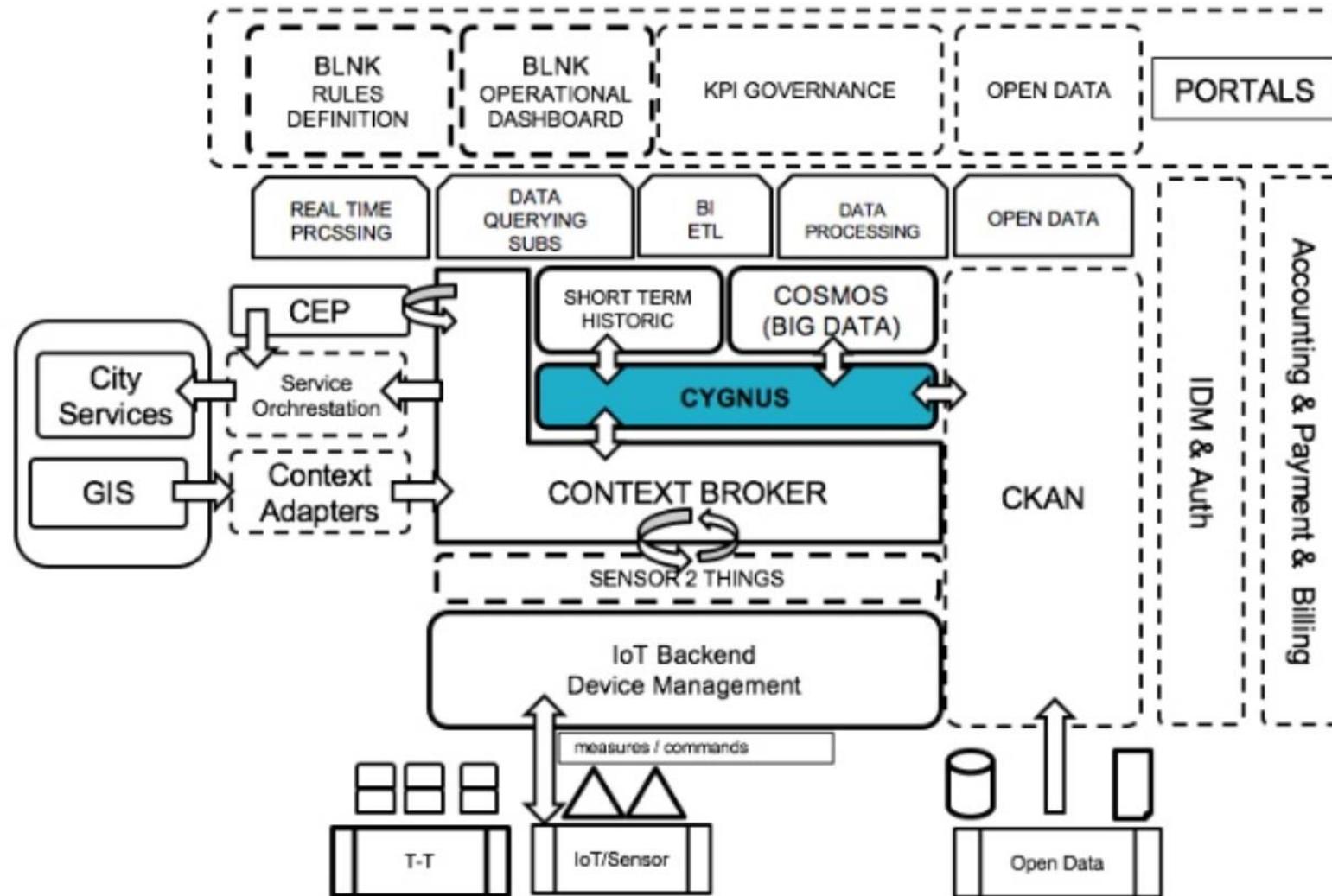
Arquitectura básica de FI-WARE para IoT



5. El componente **Short Term Historic** se encarga de proporcionar información agregada de series de tiempo sobre la evolución temporal de los valores de los atributos de una entidad, los cuales se registraron utilizando el CB. Procesado simple.
 6. El componente **Complex Event Processing** analiza datos de eventos en tiempo real con técnicas de big data. Procesado complejo.
 7. **Aplicaciones** que hacen uso de los datos generados por la plataforma.

FIWARE ARCHITECTURE

FIWARE Architecture



Context Management

- [Orion Context Broker](#) el componente central y obligatorio de cualquier plataforma o solución "Powered by FIWARE". Permite gestionar la información de contexto de manera altamente descentralizada y a gran escala.
 - Proporciona [FIWARE NGSIV2 API](#) que es una API Restful simple pero potente que permite realizar actualizaciones, consultas o suscribirse a cambios en la información de contexto.
 - Obtiene información sobre el contexto actual. Sin embargo, la información de contexto evoluciona con el tiempo, creando un historial de contexto.
- [STH Comet](#) ofrece los medios para almacenar un historial a corto plazo de datos de contexto (generalmente meses) en MongoDB
- [Cygnus](#) ofrece los medios para administrar el historial de contexto que se crea como una secuencia de datos que se puede injectar en múltiples sumideros de datos, incluidas algunas bases de datos populares como PostgreSQL, MySQL, MongoDB o AWS DynamoDB, así como plataformas BigData como Hadoop , Tormenta, Chispa o Flink.

Context Processing, Analysis and Visualisation

- [Cosmos](#) permite un análisis de Big Data más fácil sobre el contexto integrado con las plataformas de Big Data más populares.
- [Perseo](#) introduces Complex Event Processing (CEP) defined using a rules-based system, enabling you to fire events which send HTTP requests, emails, tweets, SMS messages etc.
- [Wirecloud](#) ofrece plataforma web mashup que hace que sea más fácil desarrollar paneles operativos que sean altamente personalizables por los usuarios finales
- [Knowage](#) ofrece plataforma de Business Intelligence que permite realizar análisis de negocios sobre fuentes tradicionales y sistemas de big data.
- [Kurento](#) permite el procesamiento en tiempo real de transmisiones en streamings que admiten la transformación de cámaras de video en sensores, así como la incorporación de funciones de aplicación avanzadas (comunicaciones audiovisuales integradas, realidad aumentada, reproducción y grabación de medios flexibles, etc.)
- [FogFlow](#) es un framework distribuido para admitir flujos de procesamiento dinámico sobre el Cloud y el Edge.

Arquitectura FIWARE



Generic Enablers, GE



Destacamos los siguientes GEs

Publish/Subscribe Context Broker - Orion

- NGSI Rest API
 - Create
 - Update
 - Query
 - Subscribe

BigData Analysis - Cosmos

- Cygnus
- Cluster Infinity
- Sistemas
 - HDFS
 - SQL

Application Mashup – Wirecloud

- Espacios de trabajo
- Wiring entre widgets
- Código abierto

ÍNDICE



- 1. Introducción**
- 2. Arquitectura FIWARE**
- 3. Publish/Subscribe Context Broker – Orion Context Broker**
- 4. BigData Analysis – Cosmos**
- 5. Application Mashup – Wirecloud**
- 6. Portal FI-LAB**

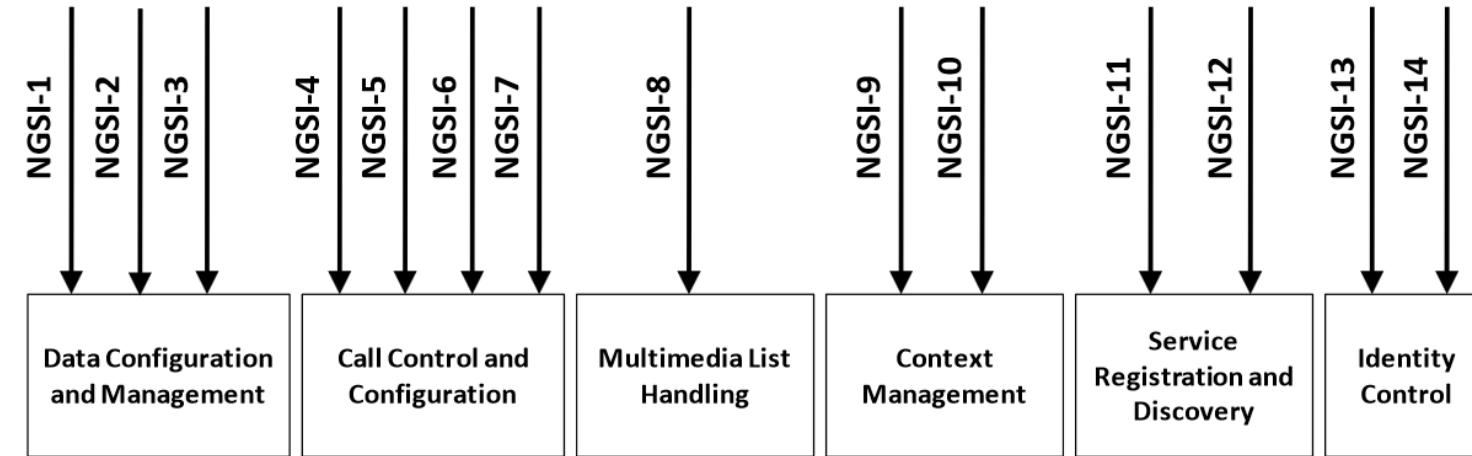
Orion Context Broker es una implementación del GE Public/Subscribe Context Broker, mediante las interfaces NGSI-9 y NGSI-10. El propósito del CB es el intercambio de información de contexto



- O • **Registrar** aplicaciones productoras de información de contexto, p. ej. un sensor de temperatura en una habitación.
- P • **Actualizar** el valor de contexto, p. ej. actualizar el valor del sensor de temperatura.
- E • **Recibir notificaciones** cuando se produce un cambio en la información de contexto o cada cierto tiempo, p. ej. recibir el valor de temperatura cada minuto.
- R • **Consultar** la información de contexto mediante *querys*.
- A
- C
- I
- O
- N
- E
- S Instalación Docker <https://hub.docker.com/r/fiware/orion>
Ejemplos API <https://documenter.getpostman.com/view/513743/fiware-getting-started/RVu5kp1c>

NGSI - Next Generation Services Interface

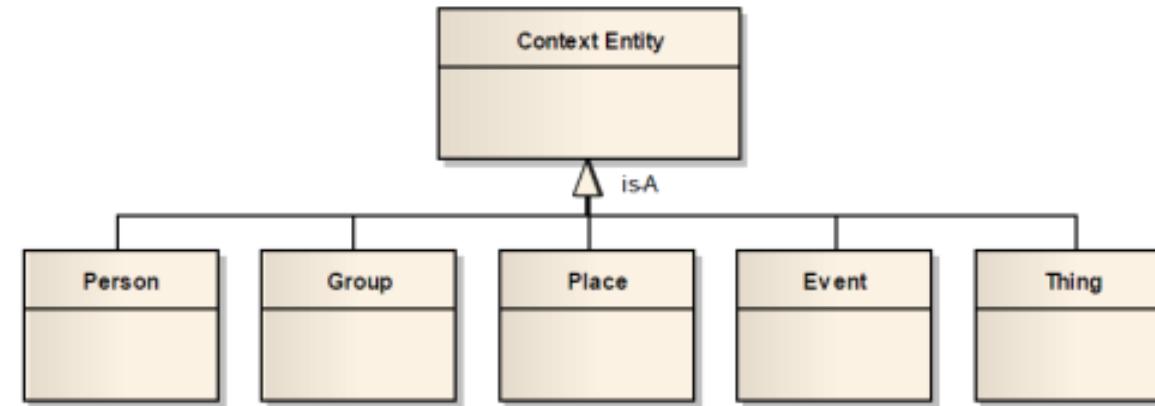
- Open Mobile Alliance (OMA) standard
- Architecture with 6 com



- The Context Management component provides the NGSI-9 and NGSI-10 interfaces for managing context entity information
 - *NGSI-9: Context Entity Discovery Interface*
 - *NGSI-10: Context Information Interface*

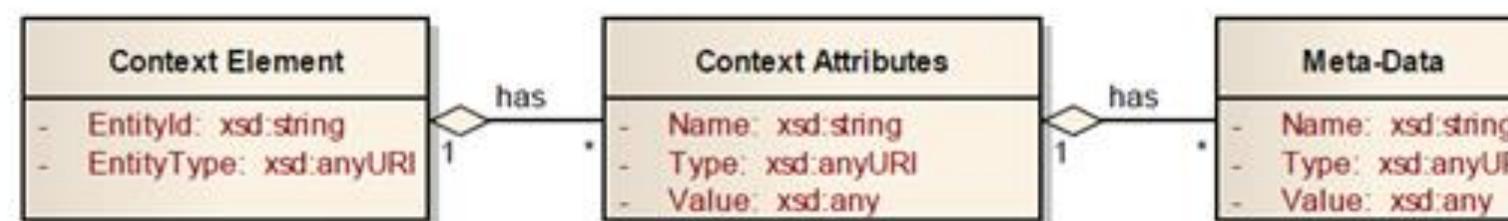
NGSI - Next Generation Services Interface

- Example context entities



- Context Information model

- The context information model details how context information is structured and associated with context entities in order to describe their situation.



Two “flavors” of NGSI API

- NGSIv1
 - Original NGSI RESTful binding of OMA-NGSI
 - Implemented in 2013
 - Uses the **/v1** prefix in resource URL
- NGSIv2
 - A revamped, simplified binding of OMA-NGSI
 - Simple things must be easy
 - Complex things should be possible
 - Agile, implementation-driven approach
 - Make it as developer-friendly as possible (RESTful, JSON, ...)
 - Enhanced functionality compared with NGSIv1 (eg. filtering)
 - Stable, ready for production, version already available
 - Current NGSIv2 version is **Release Candidate 2016.10** <http://telefonicaid.github.io/fiware-orion/api/v2/stable>
 - New features coming (<http://telefonicaid.github.io/fiware-orion/api/v2/stable>)
 - Uses the **/v2** prefix in resource URL
- Introduction to NGSIv2
 - https://docs.google.com/presentation/d/1_fv9dB5joCsOCHlb4Ld6A-QmeIYhDzHgFHUWreGmvKU/edit#slide=id.g53c31d7074fd7bc7_0

Publish/Subscribe Context Broker – Orion Context Broker

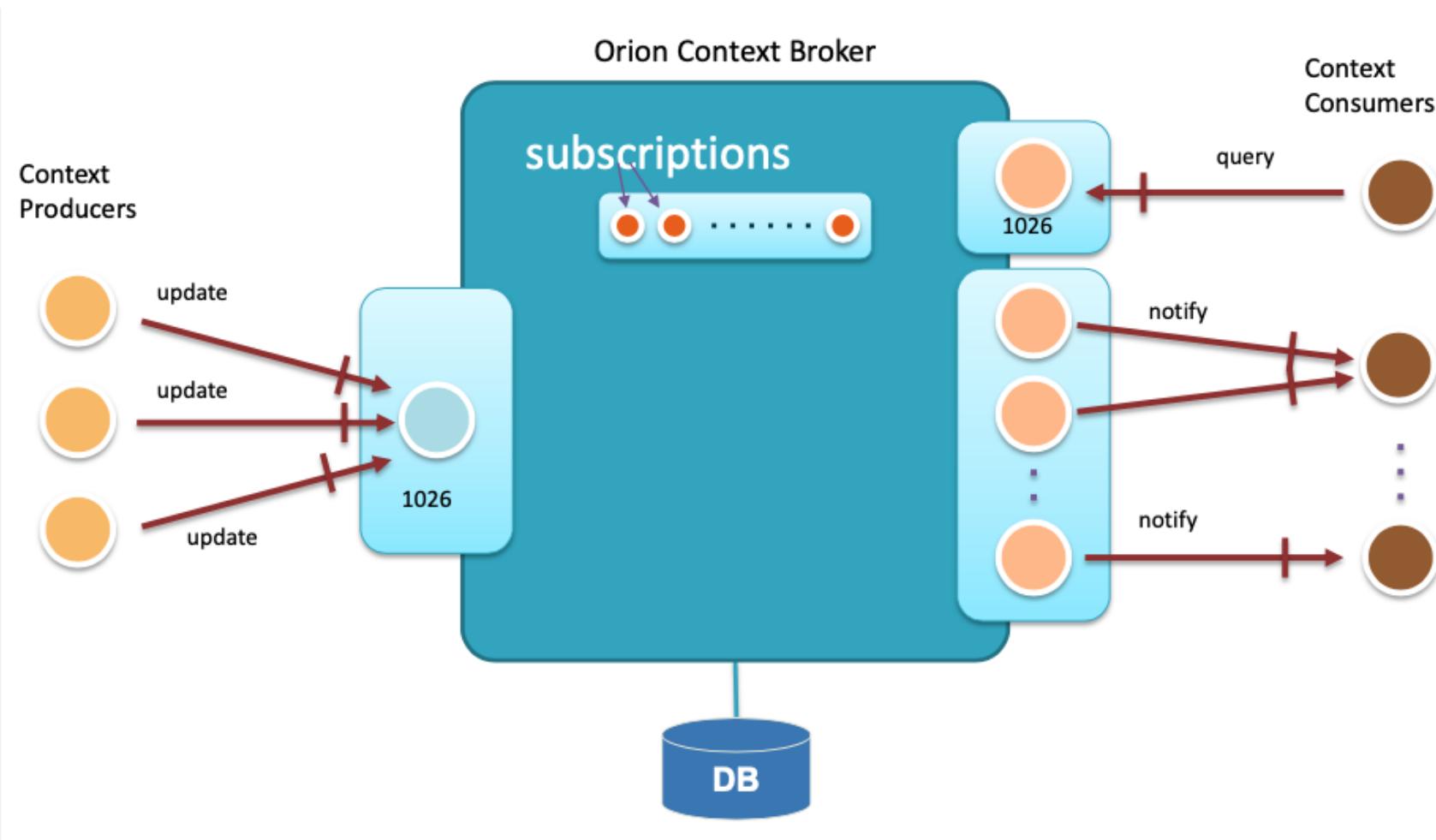
Interacción básica entre actores del CB



Suscripciones



Orion Context Broker in a nutshell



Basic Operations - Entities

- GET /v2/entities
 - Retrieve all entities
- POST /v2/entities
 - Creates an entity
- GET /v2/entities/{entityID}
 - Retrieves an entity
- [PUT|PATCH|POST] /v2/entities/{entityID}
 - Updates an entity (different “flavors”)
- DELETE /v2/entities/{entityID}
 - Deletes an entity

Basic Operations - Attributes

- GET /v2/entities/{entityID}/attrs/{attrName}
 - Retrieves an attribute's data
- PUT /v2/entities/{entityID}/attrs/{attrName}
 - Updates an attribute's data
- DELETE /v2/entities/{entityID}/attrs/{attrName}
 - Deletes an attribute
- GET /v2/entities/{entityID}/attrs/{attrName}/value
 - Retrieves an attribute's value
- PUT /v2/entities/{entityID}/attrs/{attrName}/value
 - Updates an attribute's value



Quick Usage Example:

Room Create

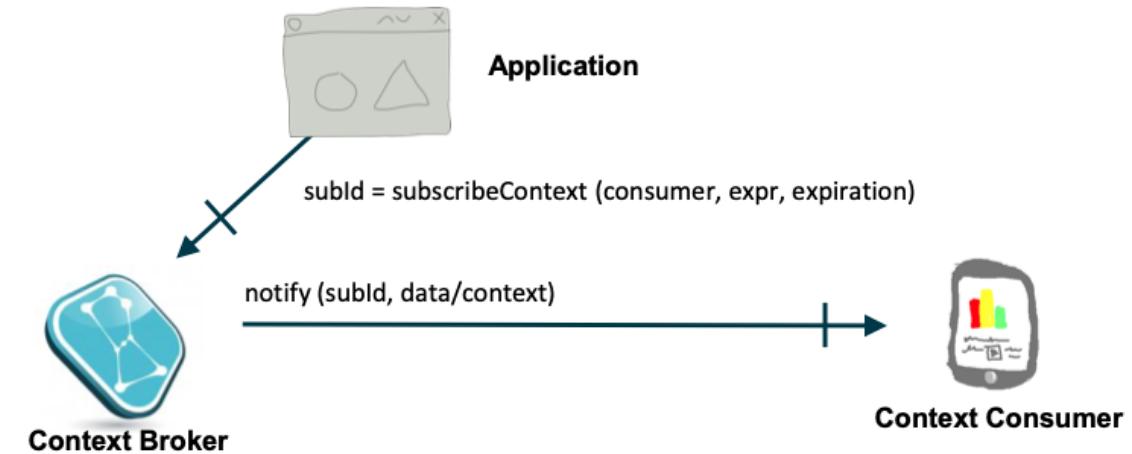
```
POST <cb_host>:1026/v2/entities  
Content-Type: application/json  
...  
  
{  
  "id": "Room1",  
  "type": "Room",  
  "temperature": {  
    "type": "Float",  
    "value": 24  
  },  
  "pressure": {  
    "type": "Integer",  
    "value": 718  
  }  
}
```

Room Update

```
PATCH <cb_host>:1026/v2/entities/Room1/attrs  
Content-Type: application/json  
...  
  
{  
  "temperature": {  
    "type": "Float",  
    "value": 25  
  },  
  "pressure": {  
    "type": "Integer",  
    "value": 720  
  }  
}
```

Context Broker operations: push data

- **Context Consumers** can subscribe to receive context information that satisfy certain conditions using the **subscribe** operation. Such subscriptions may have an expiration time.
- The Context Broker notifies updates on context information to subscribed Context Consumers by invoking the **notify** operation they export



Quick Usage Example: Subscription

```
POST <cb_host>:1026/v2/subscriptions
Content-Type: application/json
...
{
  "subject": {
    "entities": [
      {
        "id": "Room1",
        "type": "Room"
      }
    ],
    "condition": {
      "attrs": [ "temperature" ]
    },
    "notification": {
      "http": {
        "url": "http://<host>:<port>/publish"
      },
      "attrs": [ "temperature" ]
    },
    "expires": "2026-04-05T14:00:00.00Z"
  }
}
```

```
201 Created
Location: /v2/subscriptions/51c0ac9ed714fb3b37d7d5a8
...
```



Quick Usage Example: Notification

```
POST /publish HTTP/1.1
Content-type: application/json; charset=utf-8
Ngsiv2-AttrsFormat: normalized

...
{
  "subscriptionId": "574d720dbef22abb860534a",
  "data": [
    {
      "id": "Room1",
      "type": "Room",
      "temperature": {
        "type": "Float",
        "value": 19,
        "metadata": {}
      }
    }
  ]
}
```



Query filters

- For the **GET /v2/entities** operation
- By **entity type** GET <cb_host>:1026/v2/entities?type=Room
- By **entity id list** GET <cb_host>:1026/v2/entities?id=Room1,Room2
- By **entity id pattern** (regex) GET <cb_host>:1026/v2/entities?idPattern=^Room[2-5]
- By **entity type pattern** (regex) GET <cb_host>:1026/v2/entities?typePattern=T[ABC]
- By **geographical location**
 GET
 <cb_host>/v2/entities?georel=near;maxDistance:1000&geometry=point
 &coords=40.4168,-3.7038
- Filters can be used simultaneously (i.e. like **AND** condition)

Query filters

- By **attribute value** (q)
- By **metadata value** (mq)
- Filters can be also used in subscriptions
 - id
 - type
 - Id pattern
 - type pattern
 - attribute values
 - metadata value
 - geographical location

```
POST <cb_host>:1026/v2/subscriptions
...
{
  "subject": {
    "entities": [
      {
        "id": "Car5", "type": "Car"
      },
      {
        "idPattern": "^Room[2-5]", "type": "Room"
      },
      {
        "id": "D37", "typePattern": "Type[ABC]"
      }
    ],
    "condition": {
      "attrs": [ "temperature" ],
      "expression": {
        "q": "temperature>40",
        "mq": "humidity.avg==80..90",
        "georel": "near;maxDistance:100000", "geometry": "point",
        "coords": "40.418889,-3.691944"
      }
    }
  },
  ...
}
```



ÍNDICE



- 1. Introducción**
- 2. Arquitectura FIWARE**
- 3. Publish/Subscribe Context Broker – Orion Context Broker**
- 4. BigData Analysis – Cosmos**
- 5. Application Mashup – Wirecloud**
- 6. Portal FI-LAB**

BigData Analysis – Cosmos

Cosmos es una implementación del GE Big Data de Telefónica que permite el despliegue de clusters de computación privada basados en el ecosistema *Hadoop*

- **Orion** únicamente actualiza y guarda el **último valor** leído.
- **Cosmos** es un sistema de procesado complejo de datos (Big Data)
- **Cygnus** es un inyector de datos de contexto procedentes de Orion CB.
- Para conseguir la **persistencia de datos** en Cosmos es necesario utilizar el inyector **Cygnus**.
- Éste se encarga de enviar la información recibida a la máquina virtual donde reside **Cosmos** y a hacer que éstos persistan allí, es decir, inyecta la información en el histórico.
- Esa información puede almacenarse en diferentes **formatos: HDFS, CKAN y MySQL**.
- La forma de hacer que Cygnus reciba información es suscribirlo a la instancia que contiene dicha información, generalmente con una **suscripción** de tipo **ONCHANGE**.

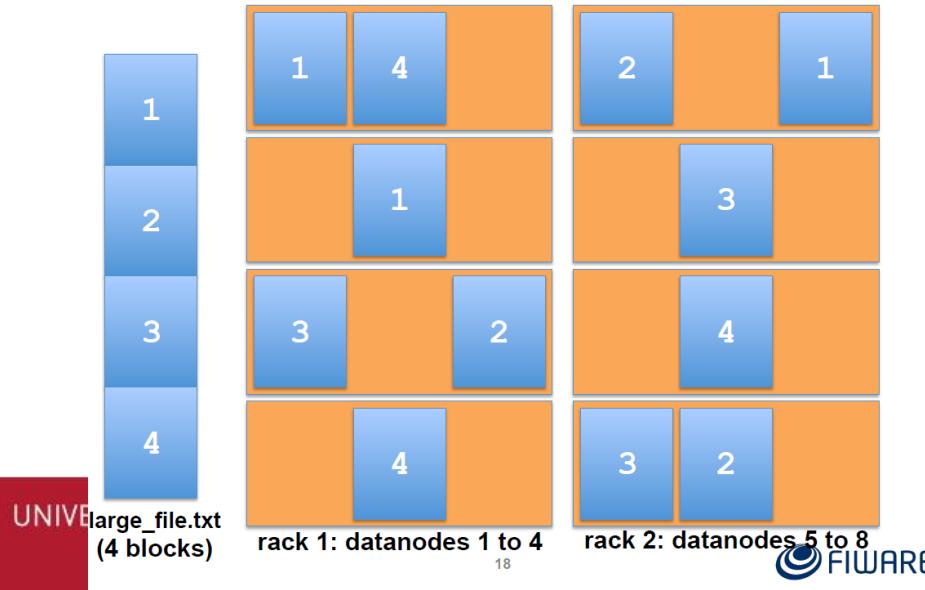


Hadoop

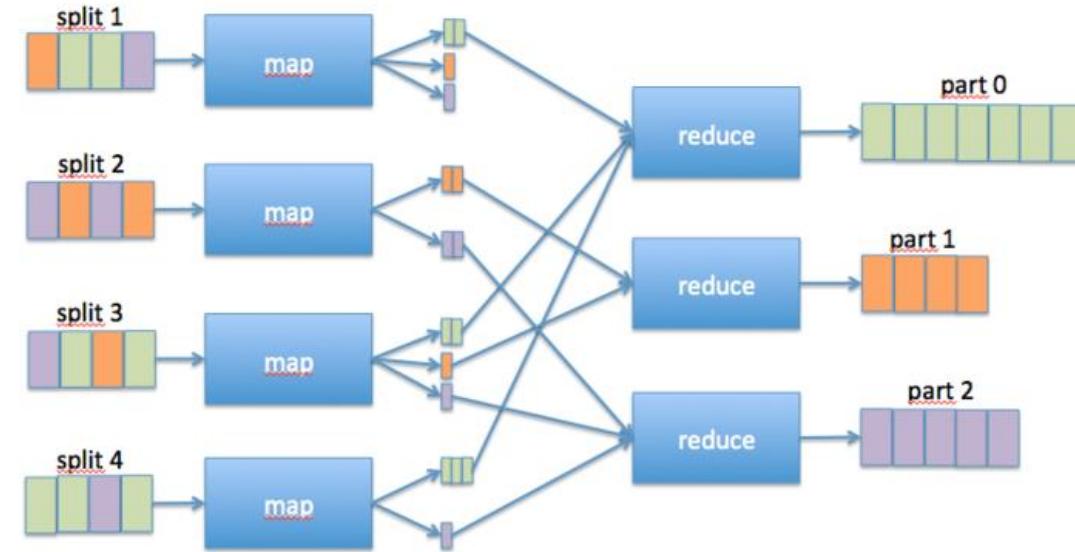
Apache Hadoop es un framework scalable que permite el procesamiento distribuido de data-set grandes en diferentes clusters.

Hadoop Distributed File System (HDFS)

- Based on Google file system
- Large files are stored across multiple machines (**datanodes**) by splitting them into blocks that are distributed
- Metadata is managed by the **namenode**
- Scalable by simply adding more datanodes
- Fault-tolerant since HDFS replicates each block (default to 3)



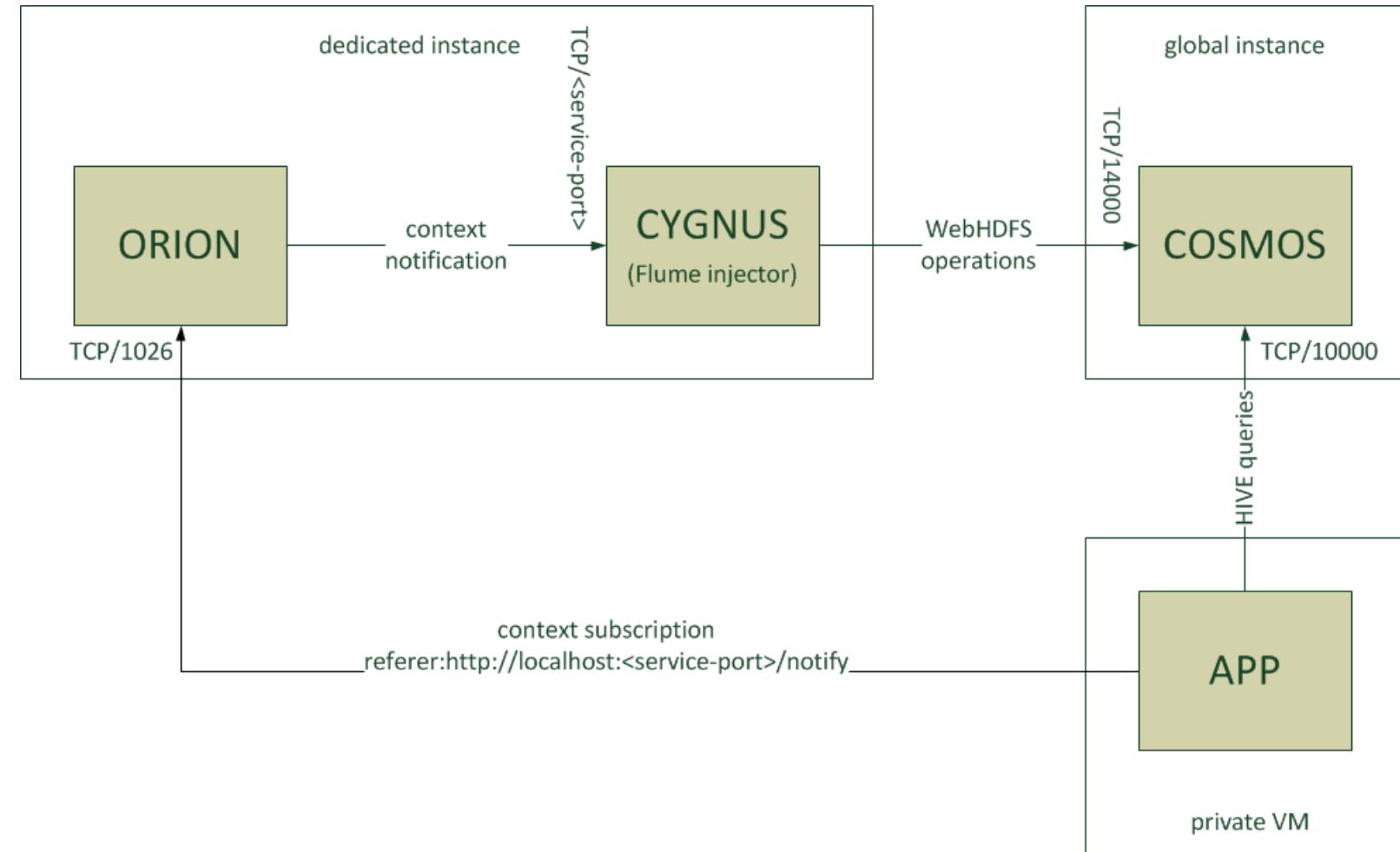
Distributed batch processing MapReduce



Path	Replicas	Block IDs
/user/user1/data/ large_file.txt	3	1 → {dn1,dn2,dn5} 2 → {dn3,dn5,dn8} 3 → {dn3,dn6,dn8} 4 → {dn1,dn4,dn7}
/user/user1/data/ other_file.txt	2	5 → {...} 6 → {...} 7 → {...}
...

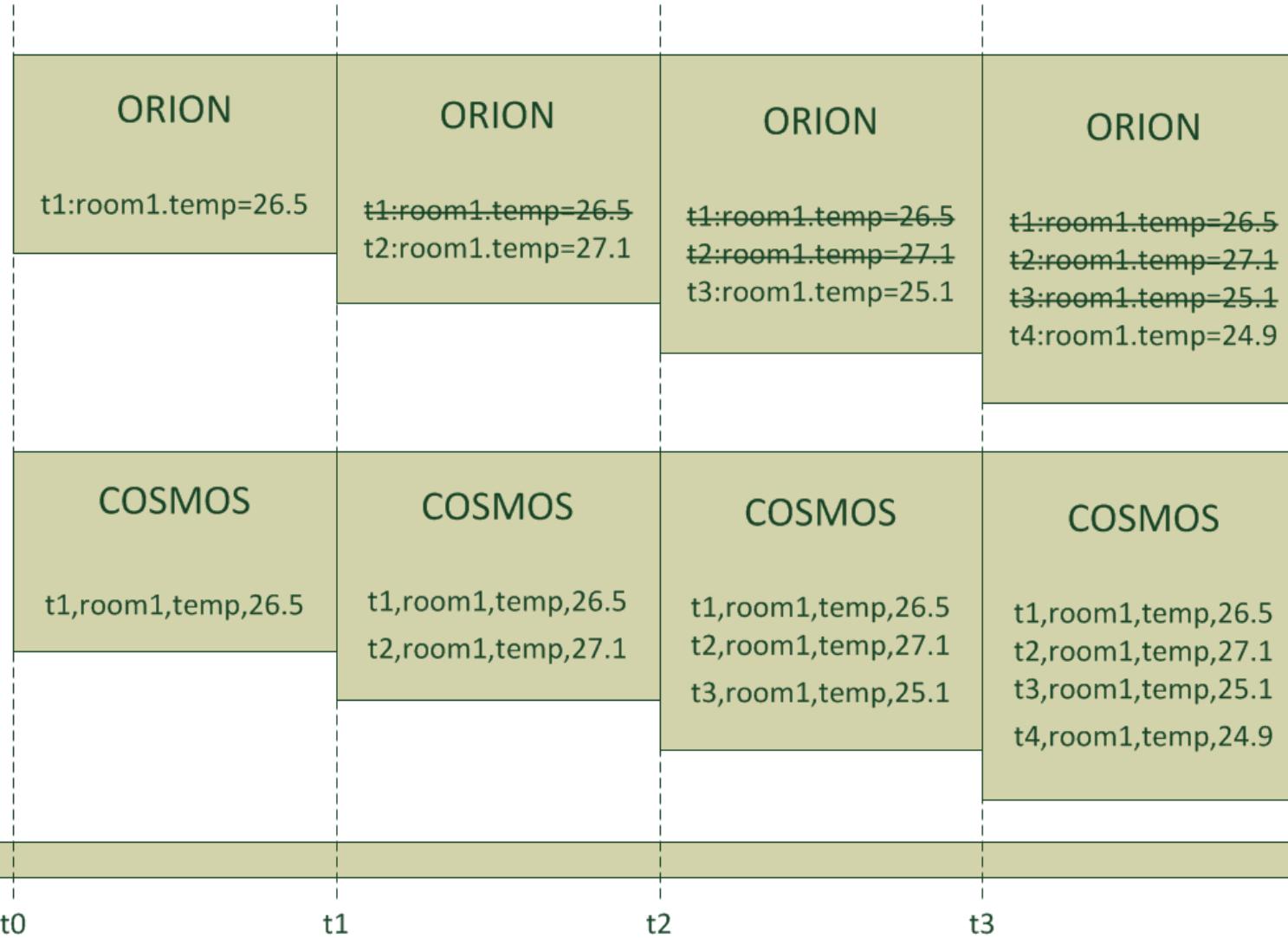
BigData Analysis – Cosmos

Real time context data persistence (architecture)



BigData Analysis – Cosmos

Real time context data persistence



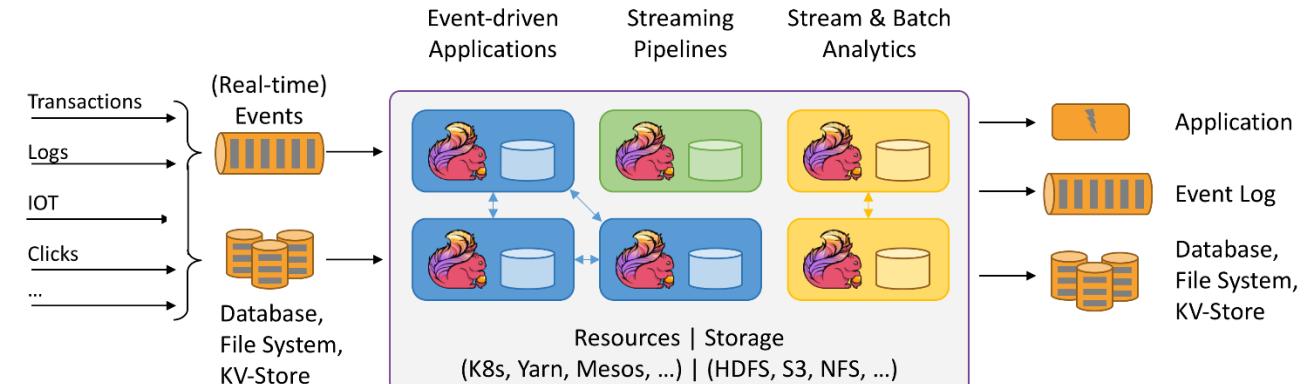
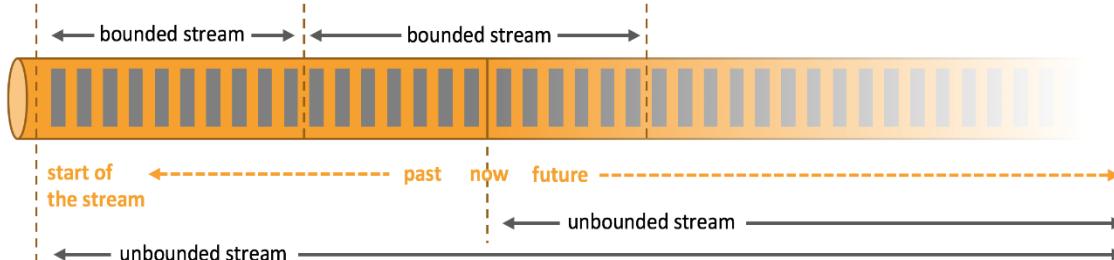
BigData Analysis – Cosmos

Cosmos está compuesto por una serie de herramientas para tareas de procesamiento de context-data tanto para Streaming y Batch:

- Orion-Flink Connector (Source and Sink)
- Apache Flink Processing Engine
- Apache Spark Processing Engine (work in progress)
- Streaming processing examples using Orion Context Broker

Apache Flink es un framework y un motor de procesamiento distribuido para cálculos stateful sobre data **streams** de tipo *unbounded and bounded*.

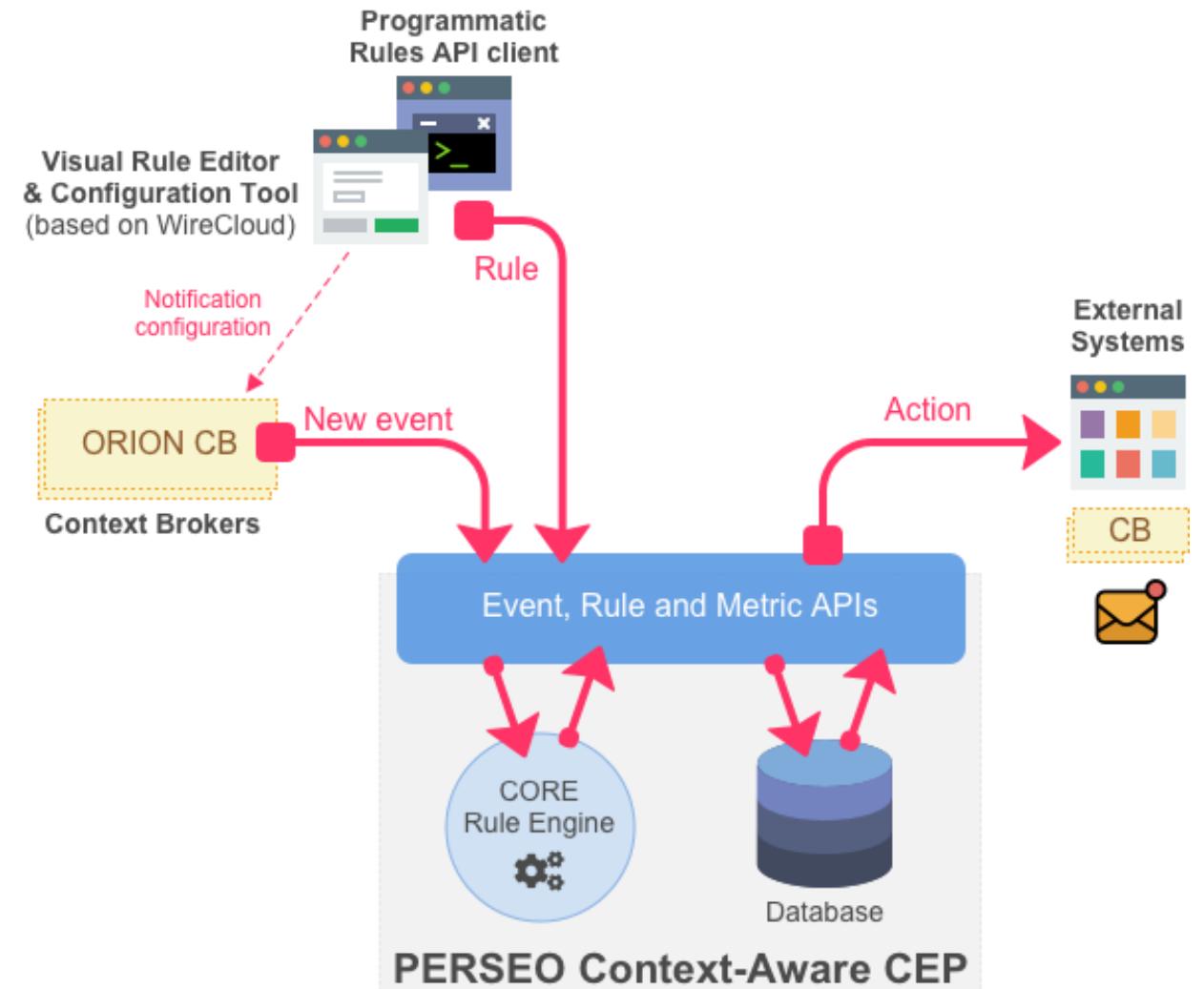
Flink ha sido diseñado para ejecutarse en todos los entornos de clúster comunes, realizar cálculos a la velocidad de la memoria y en cualquier escala.



Source:
[Flink: https://flink.apache.org/flink-architecture.html](https://flink.apache.org/flink-architecture.html)

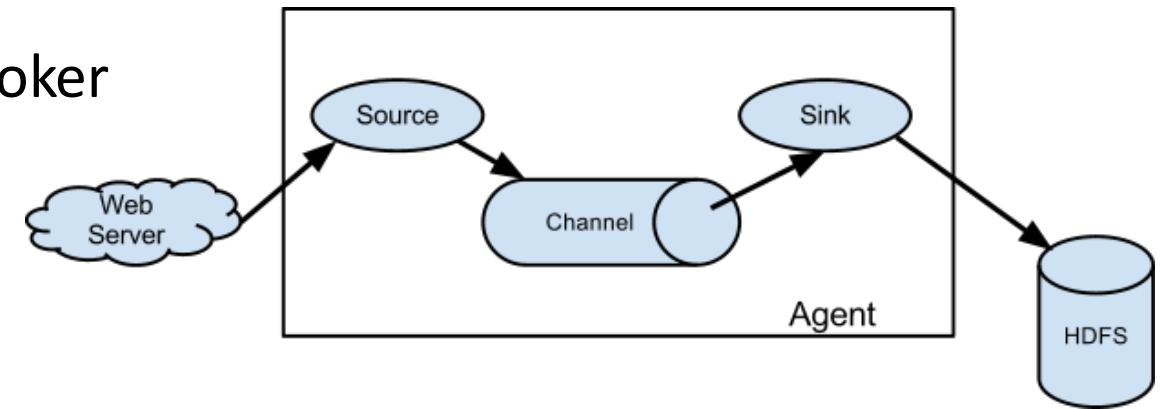
PERSEO - CEP

- Perseo es un procesador Complex Event Processing (CEP) basado en [Esper](#) y diseñado para ser *NGSIv2-compliant*
- Perseo escucha eventos que provienen del Context Broker
- Analiza e identifica patrones descritos por reglas
 - Correlación de eventos
 - Agregación, filtrado, unión, ordenación de eventos...
- Rule-engine reacciona a patrones y aplica acciones.

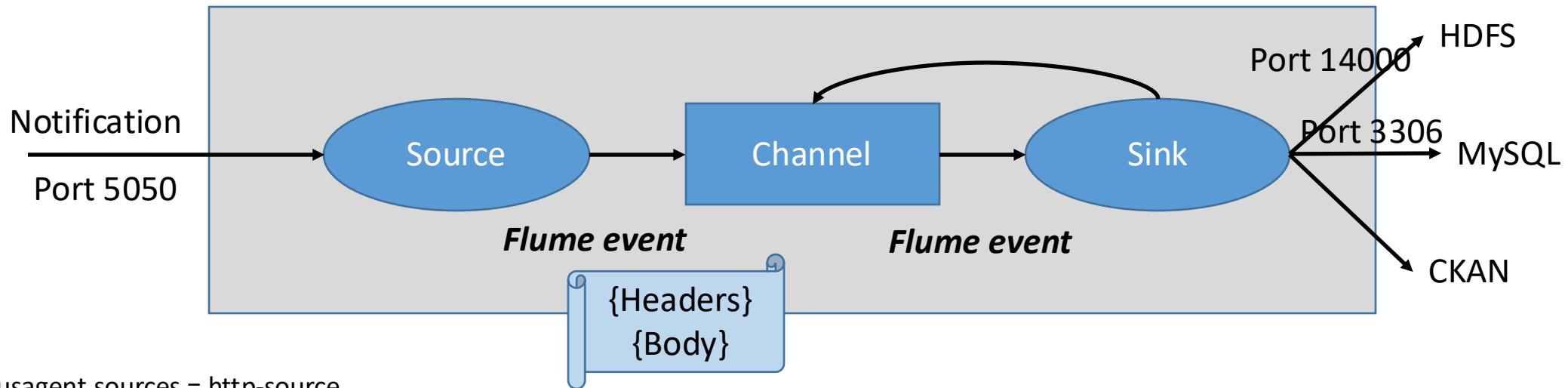


CYGNUS CONNECTOR

- Basado en Apache Flume
- Agente compuesto por un **listener** que recibe los eventos, un **canal** donde se mandan los datos y un **sink** que manda los datos en una entidad externa, e.j. HDFS.
- Permite mandar datos NGSI del Contex Broker source para datos de aplicaciones como
 - [HDFS](#), the [Hadoop](#) distributed file system.
 - [MySQL](#), the well-known relational database manager.
 - [CKAN](#), an Open Data platform.
 - [MongoDB](#), the NoSQL document-oriented database.
 - [STH Comet](#), a Short-Term Historic database built on top of MongoDB.
 - [Kafka](#), the publish-subscribe messaging broker.
 - [DynamoDB](#), a cloud-based NoSQL database by [Amazon Web Services](#).
 - [PostgreSQL](#), the well-known relational database manager.
 - [Elasticsearch](#), the distributed full-text search engine with JSON documents.



BigData Analysis - configuración Cygnus



```
cygnusagent.sources = http-source
cygnusagent.sinks = hdfs-sink
cygnusagent.channels = hdfs-channel
```

```
cygnusagent.sources.http-source.channels = hdfs-channel
cygnusagent.sources.http-source.port = 5050
...
cygnusagent.channels.hdfs-channel.type = memory
cygnusagent.channels.hdfs-channel.capacity = 1000
cygnusagent.channels.hdfs-channel.transactionCapacity = 100
```

```
cygnusagent.sinks.hdfs-sink.channel = hdfs-channel
cygnusagent.sinks.hdfs-sink.cosmos_host = cosmos.lab.fi-ware.org
cygnusagent.sinks.hdfs-sink.cosmos_port = 14000
cygnusagent.sinks.hdfs-sink.cosmos_default_username = nombreusuario
cygnusagent.sinks.hdfs-sink.cosmos_default_password = *****
...
cygnusagent.sinks.hdfs-sink.hive_host = cosmos.lab.fi-ware.org
cygnusagent.sinks.hdfs-sink.hive_port = 10000
```

ÍNDICE



- 1. Introducción**
- 2. Arquitectura FIWARE**
- 3. Publish/Subscribe Context Broker – Orion Context Broker**
- 4. BigData Analysis – Cosmos**
- 5. Application Mashup – Wirecloud**
- 6. Portal FI-LAB**

Application Mashup – Wirecloud



¿Qué es?

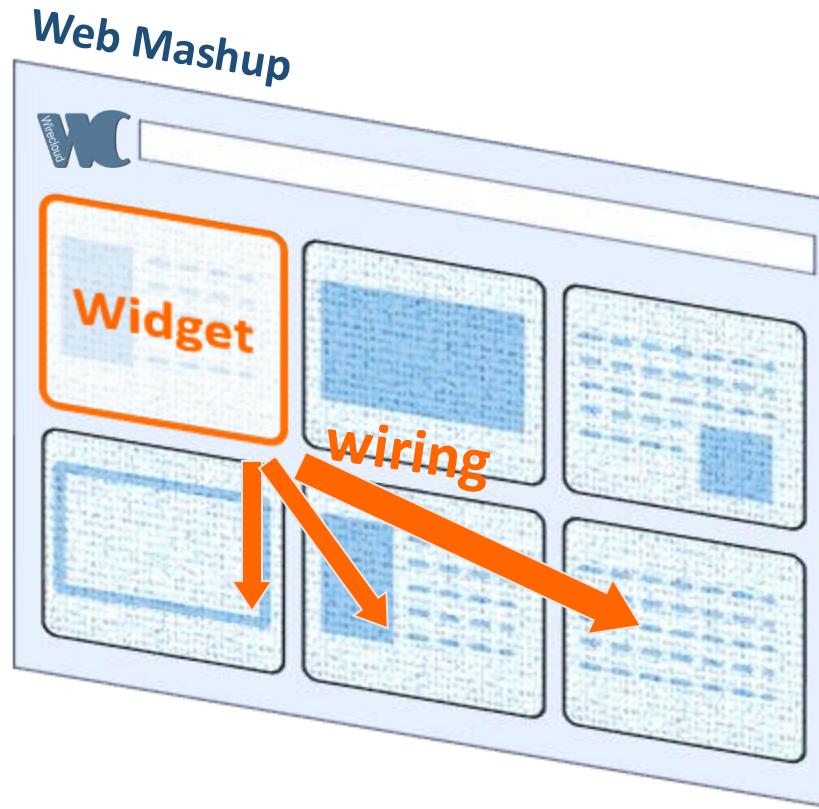
- Plataforma de construcción de *mashups* de explotación de datos
- Foco en la interfaz de usuario

¿Cómo funciona?

- Catálogo de *widgets* y *operadores*
- Panel de visualización
- Entorno *wiring*. Conexión de *widgets* mediante eventos

- Los **Mashups** o aplicaciones web integran masas heterogéneas de datos, lógica mediante aplicaciones y componentes de interfaz de usuario (UI), como son los *Widgets/Gadgets*.
- Permite que las **aplicaciones** puedan ser **enlazadas** para crear composiciones más complejas.
- Se promueve que todas las creaciones sean **compartidas entre usuarios** y estén disponibles para todos:
 - Mercado de aplicaciones, **Marketplace**.
 - **Store** para la publicación de servicios Web.

Application Mashup - Wirecloud

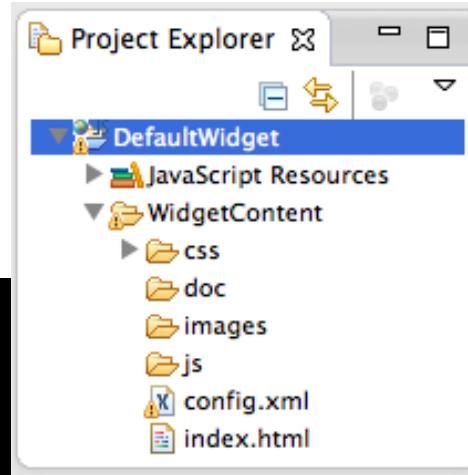


- **Widget.** Aplicación web de alcance limitado que facilita el acceso a funciones y provee de una interfaz de usuario.
- **Mashup.** Combinación en una página web o app de datos y funcionalidad de múltiples fuentes heterogéneas para crear una experiencia de usuario Enriquecida y unificada, sin desarrollos y con *time-to-market* mínimo.
- El **catálogo** permite que un usuario constructor seleccione widgets o mashups de widgets disponibles
- Los **desarrolladores** pueden enriquecer el catálogo extendiendo o con nuevos widgets/mashups.

Application Mashup – Wirecloud

Estructura de un Widget

```
.
├── README.md
├── config.xml
├── css
│   └── style.css
├── doc
│   └── developer-guide.md
├── images
│   └── chat_logo.png
├── index.html
└── js
    └── main.js
```



Index.html. Diseño del widget y los elementos que lo conformarán.

Config.xml. Archivo XML donde se detallan los siguientes aspectos:

- Datos del creador tales como nombre, dirección, e-mail, etc.
- Preferencias de la plataforma.
- Imagen del widget en las distintas plataformas.
- Entradas y salidas (I/O) para poder crear mashups mediante wiring.
- Tamaño del widget.

Javascript. Es un directorio donde se guardarán los documentos .js. En ellos es donde se configura la conexión con FI-WARE y la funcionalidad del widget.

CSS. Directorio donde se guardan las hojas de estilos del aspecto del widget.

Images. Directorio donde se pueden guardar las imágenes que son usadas como recurso en el diseño.

Para poder utilizar el widget en la sección Mashup de FI-WARE hay que comprimir los archivos en formato .zip y cambiar la extensión de éste a .wgt.

Application Mashup - Wirecloud

Developing a Widget: Config.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<Template xmlns="http://wirecloud.conwet.fi.upm.es/ns/template#">

<Catalog.ResourceDescription>
    <Vendor>CoNWeT</Vendor>
    <Name>weather-example</Name>
    <Version>3.0.0</Version>
    <DisplayName>Weather Widget Example</DisplayName>
    <Author>Name</Author>
    <Mail>email@email.com</Mail>
    <Description>Example of widget for the Wirecloud Platform.</Description>
    <ImageURI>images/catalogue.png</ImageURI>
    <iPhoneImageURI>images/catalogue_iphone.png</iPhoneImageURI>
    <WikiURI>doc/index.html</WikiURI>
</Catalog.ResourceDescription>
```

Application Mashup – Wirecloud

Developing a Widget: wiring

- Declaration on config.xml

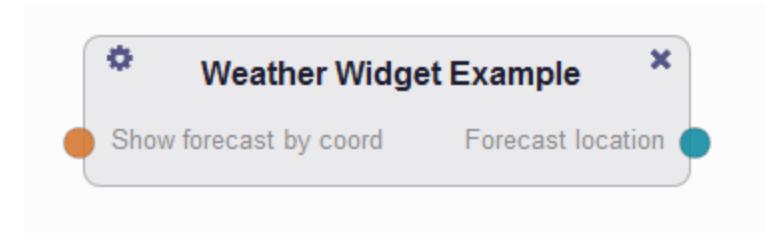
```
<Platform.Wiring>
    <OutputEndpoint label="Forecast location" description=""
type="text"
        name="location_coord" friendcode="location"/>
    <InputEndpoint label="Show forecast by coord" description=""
type="text"
        name="coord" friendcode="location"/>
</Platform.Wiring>
```

- Input endpoints

```
MashupPlatform.wiring.registerCallback('coord', searchByCoordListener);
```

- Output endpoints

```
MashupPlatform.wiring.pushEvent('location_coord', longitude + ',' + 
latitude);
```



Application Mashup - Wirecloud

¿Cómo subir un widget en Mashup?



1

The screenshot shows the Wirecloud interface with a blue circular callout labeled '1' in the top-left corner. The main area displays a message: "Hey! Welcome to Wirecloud! This is an empty workspace. To create really impressive mashup applications, the first step to take is always to add widgets in this area. To do so, please surf the Marketplace the place where resources are all in there, by clicking on the proper button up in the right corner! If you prefer, you can follow some of these tutorials: • Basic concepts". Below this message is a dashed-line box containing the same text.

2

The screenshot shows the Wirecloud Marketplace interface with a blue circular callout labeled '2' in the top-left corner. The main area displays a message: "marketplace / local". Below this, there is a list of items: "local" and "FI-LAB". A modal window is open over the list, containing buttons for "Add to workspace", "Upload", and "Refresh marketplace list". A large blue arrow points from the bottom of this screenshot towards the third screenshot.

- Una vez creado el widget, es posible agregarlo a la plataforma haciendo uso de la interfaz web.
- Comprimir el directorio del widget en zip, renombrarlo para que tenga extensión .wgt.
- Por último, subir el archivo .wgt a la plataforma.

3

The screenshot shows the Wirecloud Marketplace interface with a blue circular callout labeled '3' in the top-left corner. The main area displays a message: "/ marketplace / local". Below this, there is a list of widgets: "Weather Widget Exam" and "Web Map Service". Both widgets have a "widget" badge, a "CoNWeT" badge, and an "Add to workspace" button. A large blue arrow points from the bottom of the second screenshot towards this one.

Application Mashup – Wirecloud

¿Cómo subir un widget en Mashup?



4

The screenshot shows the FI-WARE Marketplace interface. A blue circle with the number '4' is in the top-left corner. The main area displays a list of widgets under the heading '/ marketplace / local'. One widget, 'Input Box', is highlighted with a tooltip: 'local FI-LAB' and 'Add to workspace'. Other options like 'Upload', 'Refresh marketplace list', 'Add new marketplace', and 'Delete marketplace' are also visible.

The screenshot shows the FI-WARE Marketplace interface. A blue circle with the number '4' is in the top-left corner. The main area displays a list of widgets under the heading '/ marketplace / FI-LAB'. One specific widget, 'Web Map Service', is circled in red with the word 'Install' highlighted. Other entries include 'Web Map Service' from 'WStore FI-LAB' and 'WMS Map widget'.

5

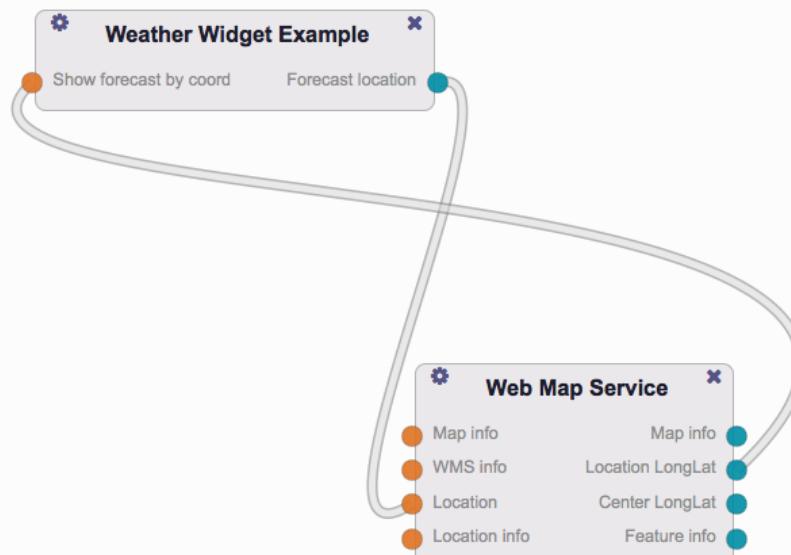
The screenshot shows the FI-WARE Cloud Incubator interface. A blue circle with the number '5' is in the top-left corner. The main area shows a 'Wiring' tab selected. On the left, a sidebar lists 'Widgets' with 'Weather Widget Example' and 'Web Map Service' circled in red. In the center, there's a 'Weather Widget Example' card with fields 'Show forecast by coord' and 'Forecast location'. On the right, a 'Web Map Service' card is shown with various connection points like 'Map info', 'WMS info', 'Location', 'Location info', 'Location LongLat', 'Center LongLat', and 'Feature info'.

- Una vez que está subido al market, lo buscamos para añadirlo al espacio de trabajo local y lo instalamos.
- Repetir para cada widget.
- Acceder a Wiring y arrastrar los dos widgets al centro

Application Mashup – Wirecloud

: Cómo subir un widget en Mashup?

6 / cloud-incubator-hub / Ejemplo conectividad / wiring



7 / cloud-incubator-hub / Ejemplo conectividad▼

The screenshot displays the results of the mashup. On the left, the 'Weather Widget Example' shows the current weather in Murcia, Spain, with a temperature of 29 °C and a map showing the location. On the right, the 'Web Map Service' shows a map of Southern Europe with a red marker indicating the location of Murcia. Below the map, there is a small table for the next four days:

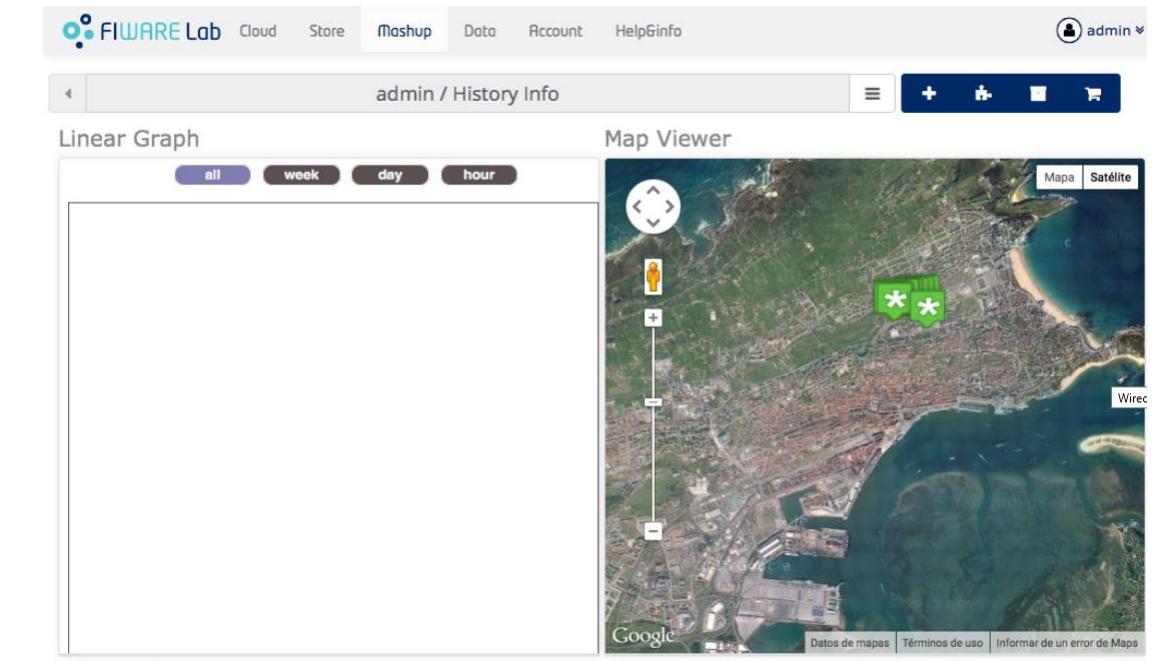
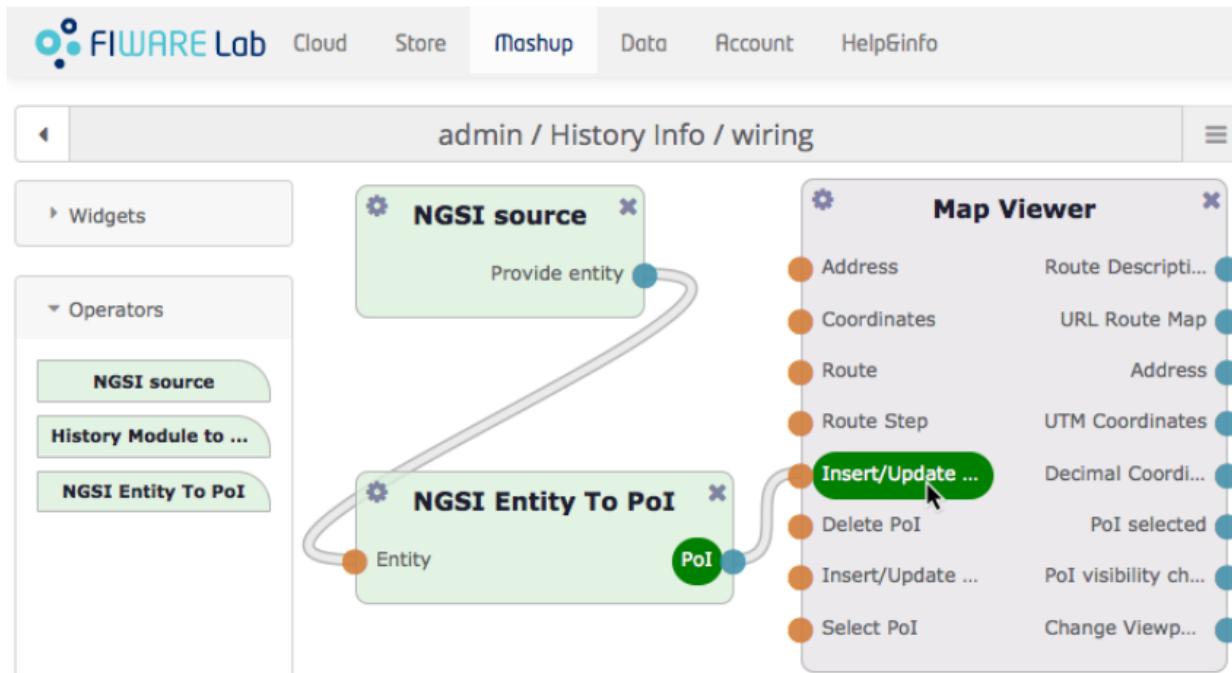
Day	Icon	Temperature	Description
Thursday	Cloudy	31 °C	Partly Cloudy
Friday	Cloudy	30 °C	Partly Cloudy
Saturday	Cloudy	30 °C	Mostly Cloudy
Sunday	Cloudy	29 °C	Partly Cloudy

- El usuario realiza la conexión de eventos entre *widgets* (*wiring*).
- En este ejemplo se puede ver el Interfaz de usuario y el *wiring* entre widgets.
- Se enlaza un widget de entrada con el mapa del tiempo de la ciudad donde se encuentra el sensor.

Application Mashup – Wirecloud

Enlazar con entidades NGSI

Ejemplo: widget map que se actualiza automáticamente con los Puntos de interest (PoI) obtenidos por un *NGSI Source operator*.



ÍNDICE



- 1. Introducción**
- 2. Arquitectura FIWARE**
- 3. Publish/Subscribe Context Broker – Orion Context Broker**
- 4. BigData Analysis – Cosmos**
- 5. Application Mashup – Wirecloud**
- 6. Portal FI-LAB**

Portal FI-LAB



FI-LAB, una instancia de FIWARE accesible en Internet para experimentar con las tecnologías FI-WARE

Cloud

- Blueprint
- Blueprint Instances
- Blueprint Templates

- Compute
- Instances
- Images
- Flavors
- Security
- Snapshots

Store

- Services
- Data
- Widgets/Mashups

MapViewer
alvaro-arranz-garcia **Free**

Place things on map! Explore the world, trace routes... Don't

Mashup

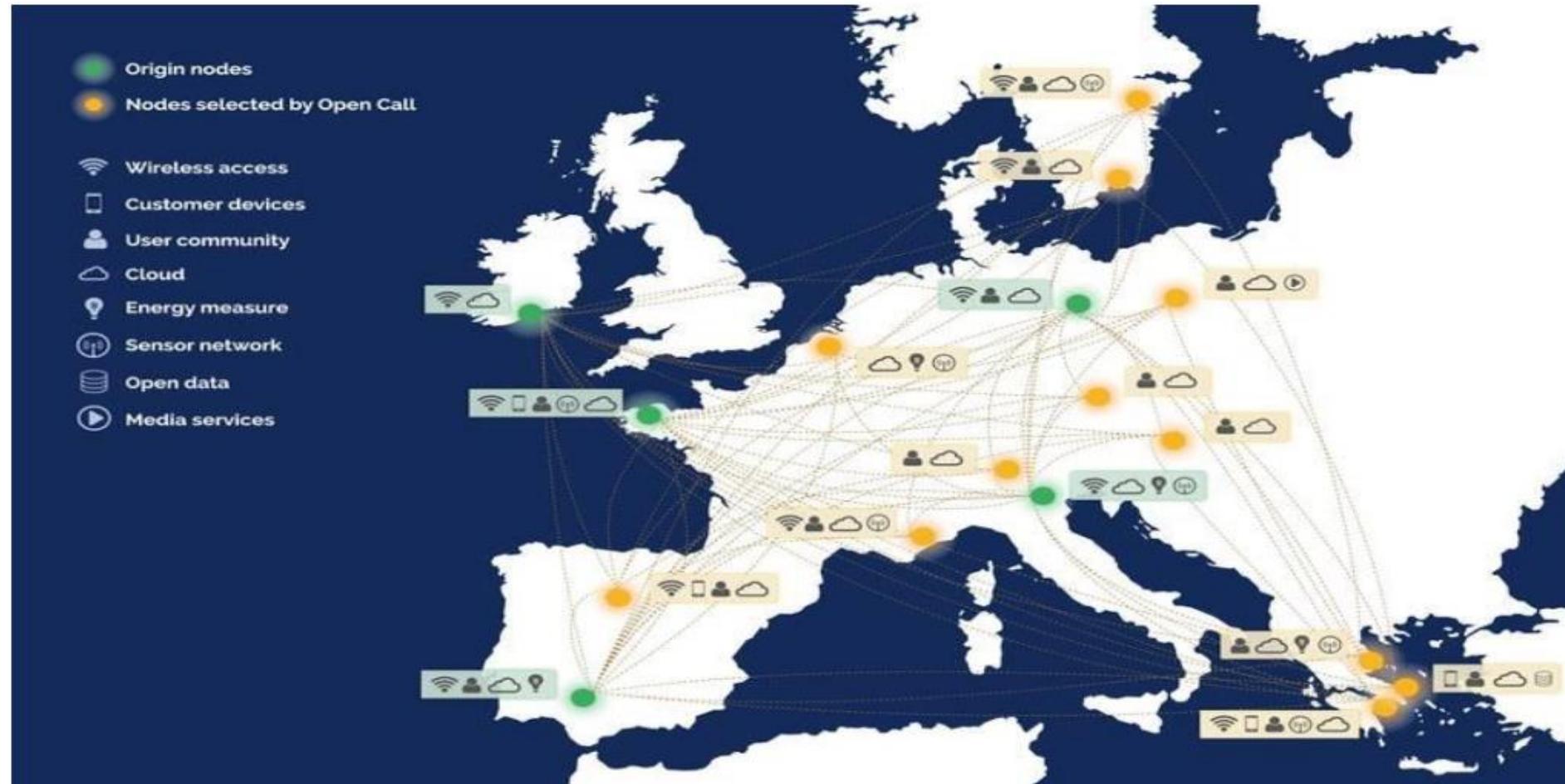
Data

Search Your Data

FIWARE Lab Data Portal statistics

2.7k	22	19
datasets	organizations	groups

Portal FI-LAB - Multi-región



Portal FI-LAB



FIWARE Lab

Cloud Store Mashup Data

Account Help&Info

FIWARE Lab

FIWARE Lab is a working instance of FIWARE available for experimentation.

You will be able to setup the basic virtual infrastructure needed to run applications that make use of the APIs provided by FIWARE Generic Enablers deployed as a Service either globally or by you (as private instance).

[Request Community Account upgrade](#) [Sign up](#)

Need Help?
 Ask a question.

Our GEs
 See our Catalogue.

FIWARE nodes
 Learn about FIWARE Ops.

FIWARE Academy
 Train yourself.

Log In

Email

Password

(eye)

remember me

Sign In

[Sign up](#) | [Forgot password](#) | [Didn't receive confirmation instructions?](#)

Portal FI-LAB: Cloud



Cloud

It provides management of all your Cloud resources, including images, instances, blueprints and software.

Basic functionalities

- Create your account in lab.FI-WARE.org
- Enter in the Cloud Portal
- Create your keypair (private key)
- Deploy your instance
- Add a public IP
- Open ports to the VM

Networks functionalities

- Create your own network
- Create your subnet associate to the previous network
- Create a router
- Set gateway
- Assign subnet
- Deploy your instance
- Assign public IP to your instance
- Check the new instance.

Portal FI-LAB: Cloud - *Create your account*

Community Account Request



Fill in the data to request a Community Account, you will receive a confirmation email at each step of the process. If not please check your SPAM inbox or contact the helpdesk: fiware-lab-help@lists.fiware.org.

User Full Name*

Your full name as by your ID

User Account Email*

Insert the email associated to the main representative in your project

Are you already*
registered in
FIWARE Lab?

Yes
 No
Confirm that you created a main account for your project in [FIWARE Lab](#). You should be able to register a "basic" account without any issue. In case of problems, the Help Desk will support you in the creation of the account.

Company*



Portal FI-LAB: Cloud Portal

Una de las principales características de la **creación de las VM** es que sobre ellas se puede instalar software de dos formas distintas:

- 1) Creación de máquinas usando **Blueprint Templates**. Es decir, se utilizarán plantillas de software que mediante unos sencillos pasos instalarán y configurarán la VM de forma automática.
- 2) Existen **imágenes pre-configuradas** que permiten una instalación más rápida y sencilla. Este tipo de instalación se emplea para casos en los que no se requiera un software específico en la máquina virtual. Existe, por ejemplo, una configuración rápida de un Context Broker mediante el lanzamiento de la imagen orion-psb-image

Name	Status	Visibility	Container Format	Disk Format	Actions
BolINC	active	public	OVF	QCOW2	<button>Launch</button>
CentOS-6.3-sdc	active	public	OVF	QCOW2	<button>Launch</button>
CentOS-6.3-x86_64	active	public	OVF	QCOW2	<button>Launch</button>
CentOS-6.5-x64	active	public	OVF	QCOW2	<button>Launch</button>
CentOS6.3.int	active	public	OVF	QCOW2	<button>Launch</button>
CentOS6init	active	public	OVF	QCOW2	<button>Launch</button>
LPCI-internal	active	public	OVF	QCOW2	<button>Launch</button>
MIWi-POI server	active	public	OVF	QCOW2	<button>Launch</button>
MIWi-POI server	active	public	OVF	QCOW2	<button>Launch</button>
Snapshot_orion_citiesense	active	public	OVF	QCOW2	<button>Launch</button>

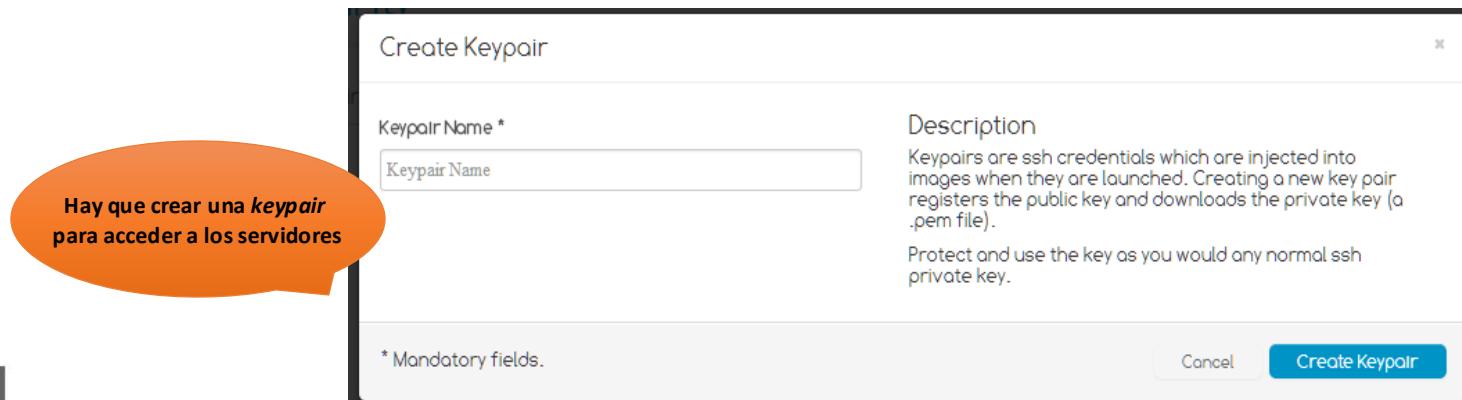
Success: Released Floating IP 130.206.83.21

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

Portal FI-LAB: Cloud - *keypair*

SEGURIDAD

- Existe un apartado donde se crean las IPs públicas, a través de las cuales se puede acceder a una máquina virtual y otro, donde se crean los llamados Grupos de Seguridad (*Security group*), en los cuales se permiten administrar los puertos abiertos a una máquina.
- Para el acceso a las máquinas a través de la IP pública se generan claves públicas llamadas *keypair*. Todo ello es assignable a una máquina virtual de forma intuitiva y en el caso de los *keypair* y *security group*, a varias a la vez.



Portal FI-LAB: Cloud - Deploy your instance

FIWARE Lab Cloud Store Mashup Data Account Help&info Fernando Lopez

Images

Name	Status	Visibility	Container Format	Disk Format	Actions
BoINC	active	public	OVF	QCOW2	<button>Launch</button>
CentOS-6.3-sdc	active	public	OVF	QCOW2	<button>Launch</button>
CentOS-6.3-x86_64	active	public	OVF	QCOW2	<button>Launch</button>
CentOS-6.5-x64	active	public	OVF	QCOW2	<button>Launch</button>
CentOS6.3.init	active	public	OVF	QCOW2	<button>Launch</button>
CentOS65init	active	public	OVF	QCOW2	<button>Launch</button>
LPCI-internal	active	public	OVF	QCOW2	<button>Launch</button>
MiWi-POI server	active	public	OVF	QCOW2	<button>Launch</button>
MiWi-POI server	active	public	OVF	QCOW2	<button>Launch</button>
Snapshot_orion_citiesense	active	public	OVF	QCOW2	<button>Launch</button>

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](#)



Launch Instances

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

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
m1.tiny

Instance Count *
1

Flavor Details

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

Project Quotas

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

Launch Instances

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

Keypair
hamburg

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

Security Groups
 default

Add new Security Group

* Mandatory fields.

Back Next

Portal FI-LAB: Cloud - IP

Asignar nueva IP

Fernando Lopez

Project Name: fernando-lopez

Blueprint: Blueprint Instances, Blueprint Templates

Region: Spain

Compute: Instances, Images, Flavors, Security (selected), Snapshots

Storage: Containers, Volumes

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.

UNIVERSIDAD DE MURCIA

Fernando Lopez

Cloud Store Mashup Data Account Help&info

Security

Floating IPs Security Groups Keypairs

Allocate IP to Project Actions

IP Address Instance Floating IP Pool

Project Name: fernando-lopez

Blueprint: Blueprint Instances, Blueprint Templates

Success: Released Floating IP 130.206.83.21

Associate Floating IP

Associate Floating IP: 130.206.83.21

Description: Associate a floating ip with an instance.

to instance: HH1

and to IP Address: Select IP to associate with

Cancel Associate IP

Associate IP Dissociate Floating IP Release Floating IPs

Displaying 1 item

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.

Portal FI-LAB: Cloud - Open ports to the VM. Security Group rules

FIWARE Lab Cloud Store Mashup Data Account Help&info Fernando Lopez ▾

Security

Project
Project Name fernando-lopez ▾
Blueprint Blueprint Instances Blueprint Templates
Region Spain ▾
Compute Instances Images Flavors Security Snapshots
Storage Containers Volumes

Floating IPs **Security Groups** Keypairs

Create a Security Group

Create Security Group

Name * hamburg Description From here you can create a new security group
Description * This is a new security group for Hamburg
* Mandatory fields. Cancel Create Security Group

Success: Keypair mex deleted.

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

Edit Security Group Rules

Security Group Rules

IP Protocol	From Port	To Port	Source	Action
TCP	22	22	0.0.0.0/0 (CIDR)	Delete Rule
TCP	3306	3306	0.0.0.0/0 (CIDR)	Delete Rule

Displaying 2 items

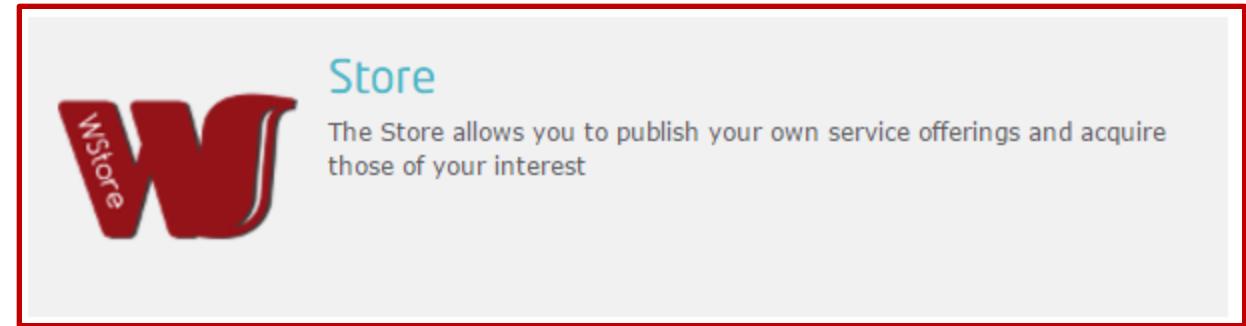
Add Rule

IP Protocol	From Port *	To Port *	Source Group	CIDR
TCP	Required field.	Required field.	CIDR	0.0.0.0/0

* Mandatory fields.

Cancel Add Rule

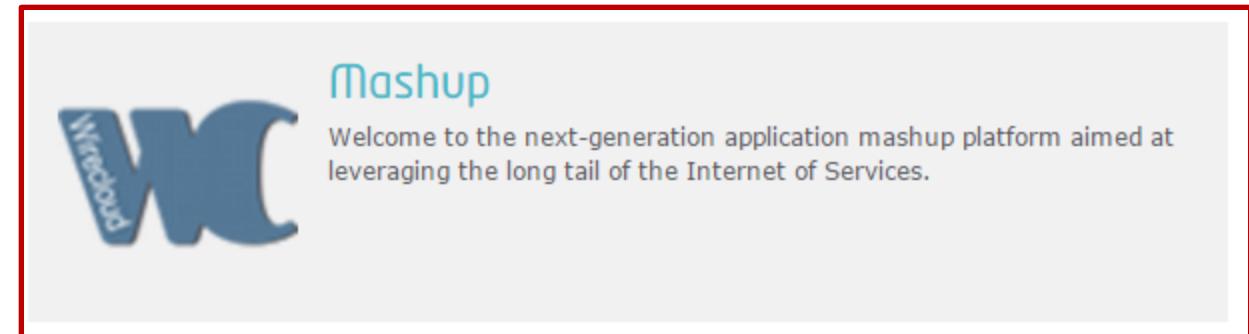
Portal FI-LAB: Store



- En este apartado del portal FI-Lab se puede encontrar una **tienda** en la que se publican las **aplicaciones** que los desarrolladores han publicado. Éstas pueden ser **gratuitos o de pago**.
- En la mayoría de los casos es posible descargar el código y observar su composición.

Top rated					
	MapViewer alvaro-arranz-garcia				MultimediaPack CoNWeT
	LinearGraph CoNWeT				Minesweeper CoNWeT
	MyLiveDemo1 fi-ware-developer				InputBoxWidget CoNWeT
	LinearGraph CoNWeT				Minesweeper CoNWeT
	MyLiveDemo1 fi-ware-developer				InputBoxWidget CoNWeT
	LinearGraph CoNWeT				Minesweeper CoNWeT

Portal FI-LAB: Mashup



La sección **Mashup** es aquella donde se suben los **widgets** o se descargan otros existentes, enlazarlos mediante **wiring** y finalmente, hacerlos funcionar.

Esta sección a su vez se divide en tres:

- En **Marketplace** se pueden ver las aplicaciones disponibles, subir las propias e instalar y desinstalar del espacio de trabajo del usuario.
- **Wiring**, espacio para situar los widgets que se desean enlazar y crear las convenientes conexiones. Todo ello se realiza de forma intuitiva y gráfica. Además, se podrán configurar los widgets mediante el menú destinado a ello que cada uno posee.
- **Editor**, es el espacio donde ejecutar las aplicaciones y ver su funcionamiento final. Se subdivide en espacios de trabajo o *workspaces* con el fin de manejar diferentes widgets y aplicaciones en escenarios separados.

Tema 4: Sistemas de adquisición y gestión de datos

Arquitectura y uso de FI-WARE

Curso 2023 - 2024

Profesor:

Juan Antonio Martínez (juanantonio@um.es)

