

Usando FIWARE FIWARE ZONE_

<http://www.fiware.org>

<http://lab.fiware.org>

Follow @FIWARE on Twitter!

0. Índice

1. El origen de los datos IoT
 - Things: sensores, actuadores, things...
 - Comunicaciones
2. Arquitectura FIWARE
 - Orion Context Broker
 - IDAS Backend Devices Gateway: IoT Agents
 - Procesando datos FIWARE

01



El origen de los datos

1.0 El origen de los datos IoT

Aplicaciones
de servicio y
dashboard

Industria 4.0

e-Health

Utilities

Otros
servicios

Gestión y
análisis de
datos

Fiware (open source)

↕ M2M

↕ LPWA

↕ HTTP

↕

Equipos
terminales

Actuadores

Sensores

IoT Things

Sensores



Sistemas



1.1. Things: sensores, actuadores...things

Sensores: convierten magnitudes físicas en datos que pueden ser tratados según la aplicación donde vayan a ser desplegados.

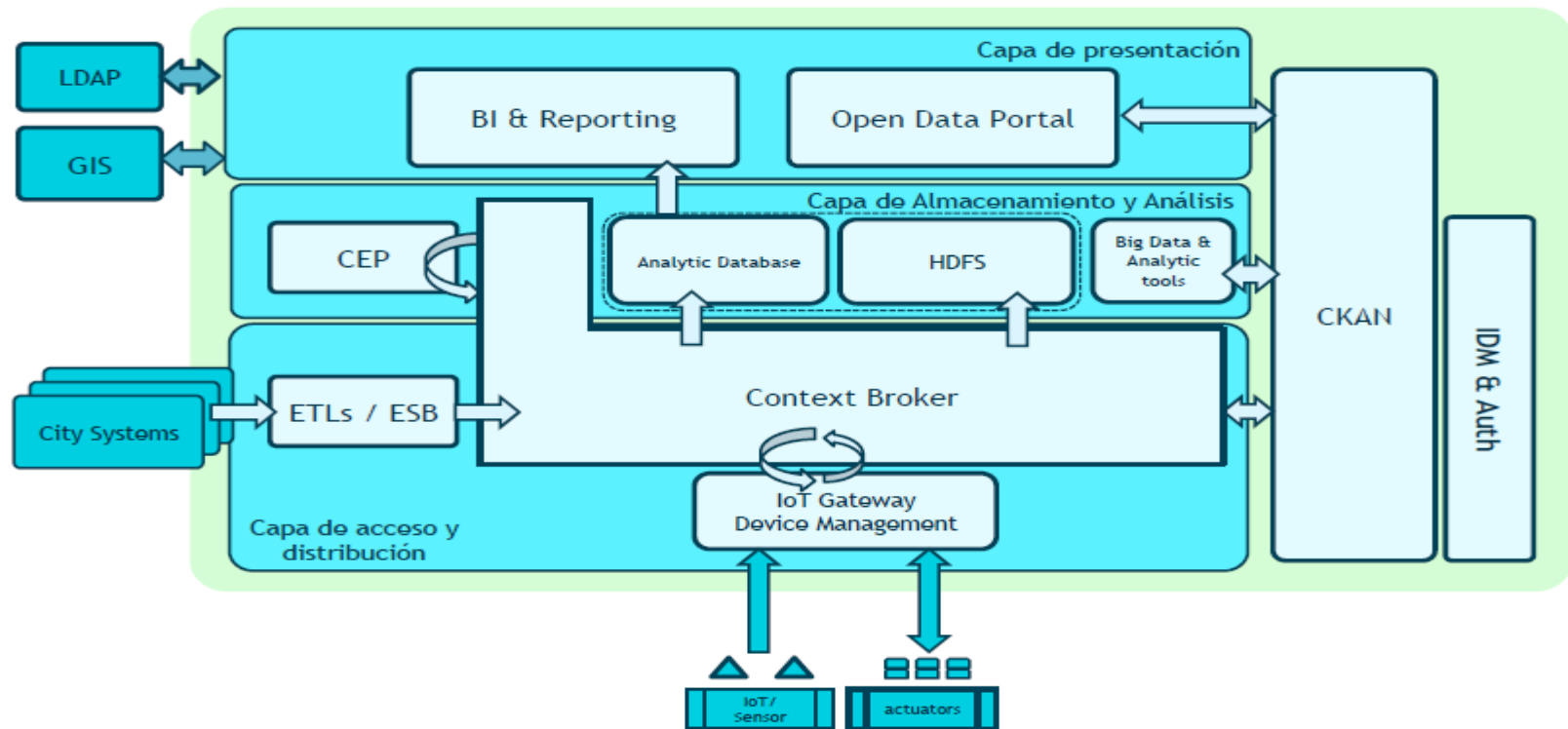
Actuadores: permite la interacción con el mundo físico.



02

Arquitectura FIWARE

2.0 Arquitectura Fiware



2.1 Generic Enablers (GEs)

FIWARE está basado en un conjunto de herramientas y librerías, conocidas como Generic Enablers con especificaciones de código abierto y de interfaces.

- Son componentes de software genéricos, reutilizables y adaptables, que sirven como bloques de construcción para crear, de forma rápida, aplicaciones y servicios específicos basados en Internet.
- Los GEs ofrecen funciones para múltiples áreas de uso que son comunes a varios sectores.

2.1 Generic Enablers (GEs)



Architecture of Applications / Services Ecosystem and Delivery Framework

Co-create, publish, cross-sell and consume applications/services, addressing all business aspects.



Cloud Hosting

Provides computation, storage and network resources to manage services.



Data/Context Management

Easing access, gathering, processing, publication and analysis of context information at large scale.



Advanced middleware and interfaces to Network and Devices (I2ND)

Build communication-efficient distributed applications, exploit advanced network capabilities and easily manage robotic devices.



Internet of Things (IoT) Services Enablement

Make connected things available, searchable, accessible, and usable.



Security

Make delivery and usage of services trustworthy by meeting security and privacy requirements.



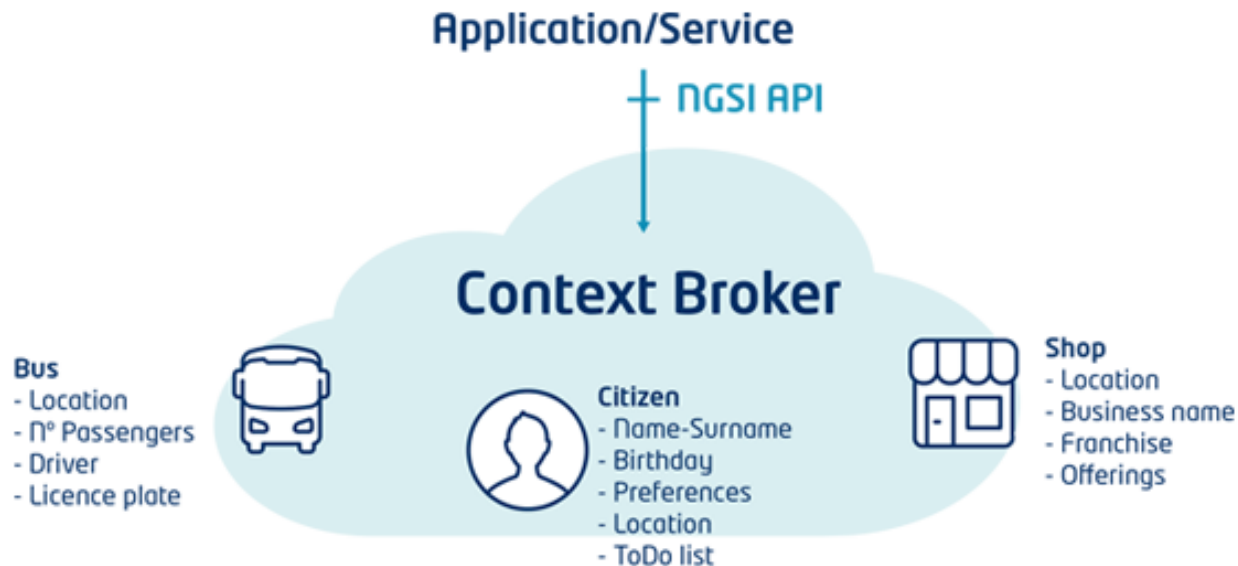
Advanced Web-based User Interface

3D & AR capabilities for web-based UI.

(<http://catalogue.fiware.org/>)

2.2 Orion Context Broker

La **información de contexto** se refiere a los valores que toman los atributos de las entidades relevantes para cada uno de los sistemas.

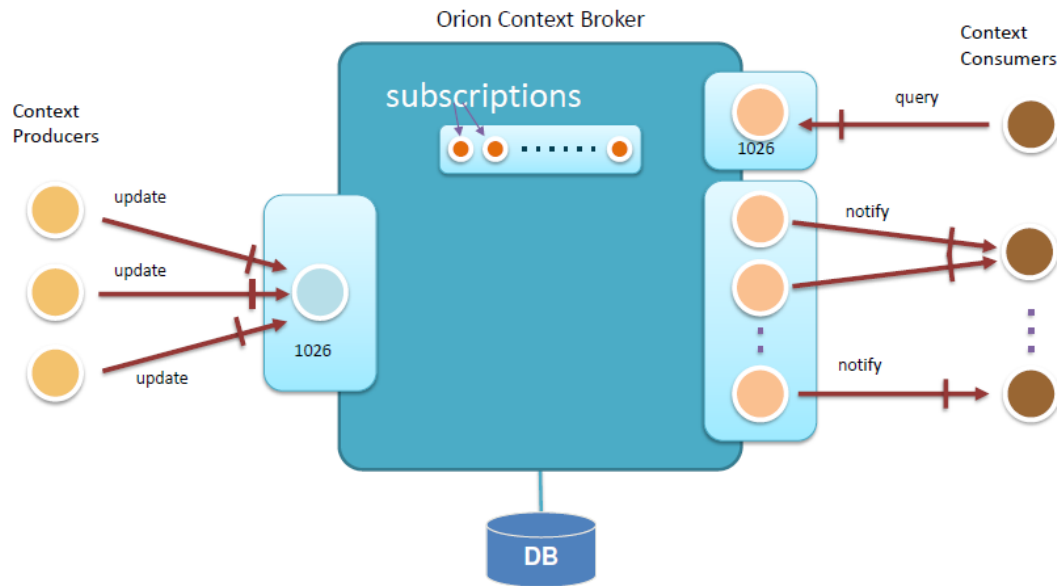


Un sistema requiere de la recopilación y la gestión de la información de contexto a gran escala, transformando un sistema normal en un sistema inteligente.

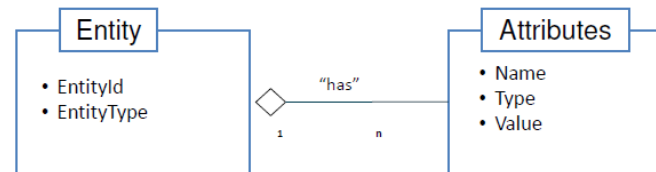
The Context Broker GE

2.3 Envío de datos al CB

Orion Context Broker implementa OMA NGSI 9/10 API: un simple pero potente standard API para gestionar información de contexto.



Modelo de datos NGSI basado en entidad-atributos



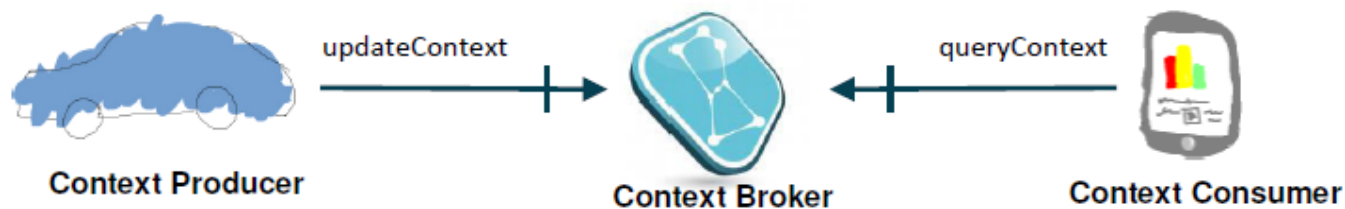
2.3 Operaciones con el CB

Context Producer

publican datos y elementos de contexto a través de la operación **updateContext**

Context Consumers

recuperan datos y elementos de contexto a través de la operación **queryContext**



CB – Operaciones con Tipos

- GET /v2/types
 - Retrieve a list of all entity types currently in Orion, including their corresponding attributes and entities count
- GET /v2/types?options=values
 - Retrieves just a list of all entity types without any extra info
- GET /v2/types/{typeID}
 - Retrieve attributes and entities count associated to an entity type

CB – Operaciones con Entidades

- 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

CB – Operaciones con atributos

- 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

CB - Subscripciones

- GET /v2/subscriptions
 - Retrieve all subscriptions
- POST /v2/subscriptions
 - Creates a subscription
- GET /v2/subscriptions/{subID}
 - Retrieves a subscription
- PATCH /v2/subscriptions/{subID}
 - Updates a subscription
- DELETE /v2/subscriptions/{subID}
 - Deletes a subscription

2.3 Envío de datos al CB

Creando entidad “car”

```
POST <cb_host>:1026/v2/entities
Content-Type: application/json
```

...

```
{
  "id": "Car1",
  "type": "Car",
  "speed": {
    "type": "Float",
    "value": 98
  }
}
```



Actualizando atributo “speed”

```
PUT <cb_host>:1026/v2/entities/Car1/attrs/speed
Content-Type: application/json
```

...

```
{
  "type": "Float",
  "value": 110
}
```

201 Created



2.4 Consulta de datos del CB

Consultando atributo speed

GET <cb_host>:1026/v2/entities/Car1/attrs/speed



200 OK
Content-Type: application/json
...
{
 "type": "Float",
 "value": **110**,
 "metadata": {}
}

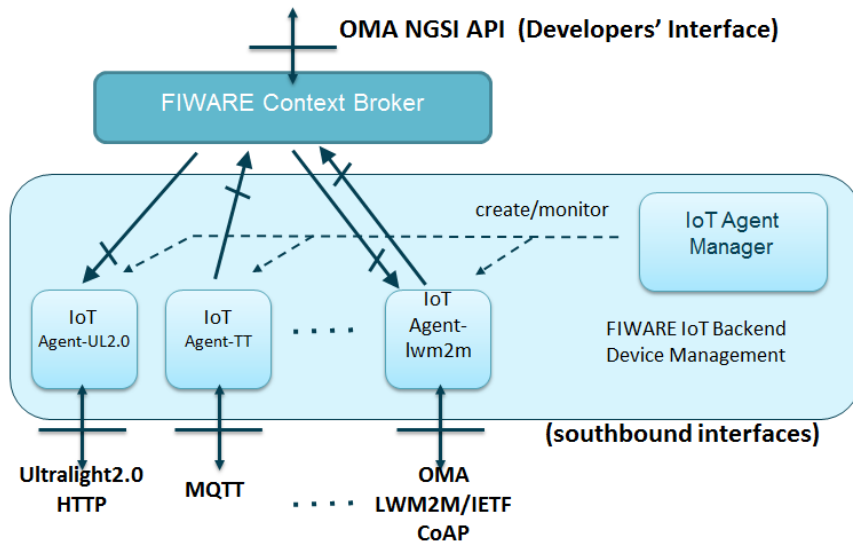


You can get all the attributes of the entity using the entity URL:
GET/v2/entities/Car1/attrs

3.1 IDAS BE Management

IDAS es una implementación de FIWARE IoT Backend Device Management GE para

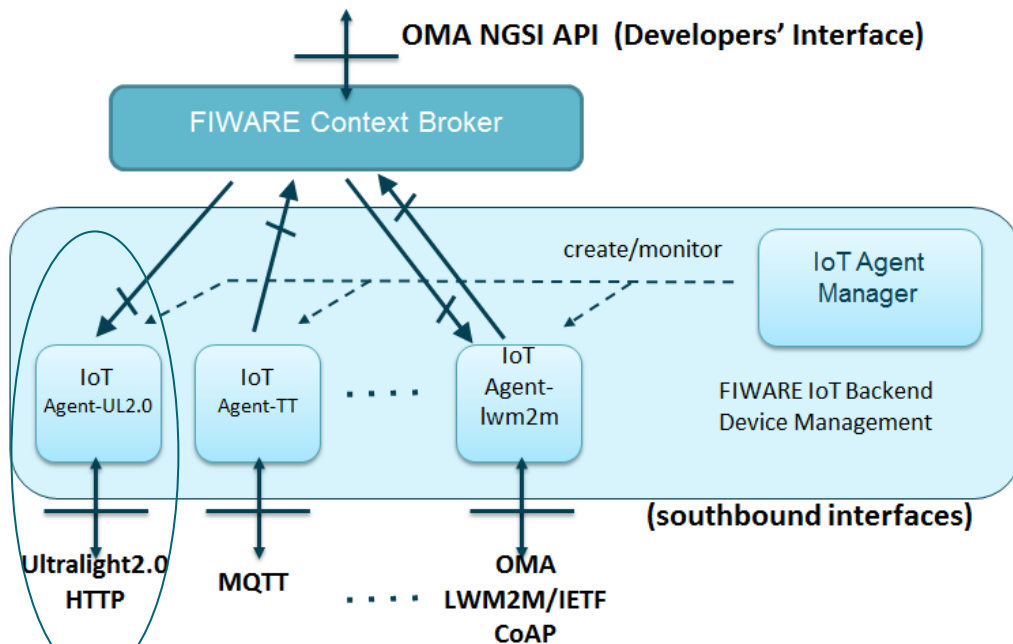
- Conectar al Orion Context Broker
- Transformar lo observado/sensado por los “things” en eventos NGSI.



Cada “IoT Agent” gestiona la interacción con un conjunto específico de dispositivos (protocolos).

3.2 Envío de datos al CB por IDAS

ENVÍO DE DATOS AL ORION CONTEXT BROKER UTILIZANDO IDAS 4.0 CON PROTOCOLO IOT Agent UL2.0



Ultralight2.0:

<https://github.com/telefonicaid/fiware-IoTAgent-Cplusplus>

- MQTT

<https://github.com/telefonicaid/fiware-IoTAgent-Cplusplus>

- LWM2M/CoAP:

<https://github.com/telefonicaid/lightweightm2m-iotagent>

- SigFox:

<https://github.com/telefonicaid/sigfox-iotagent>

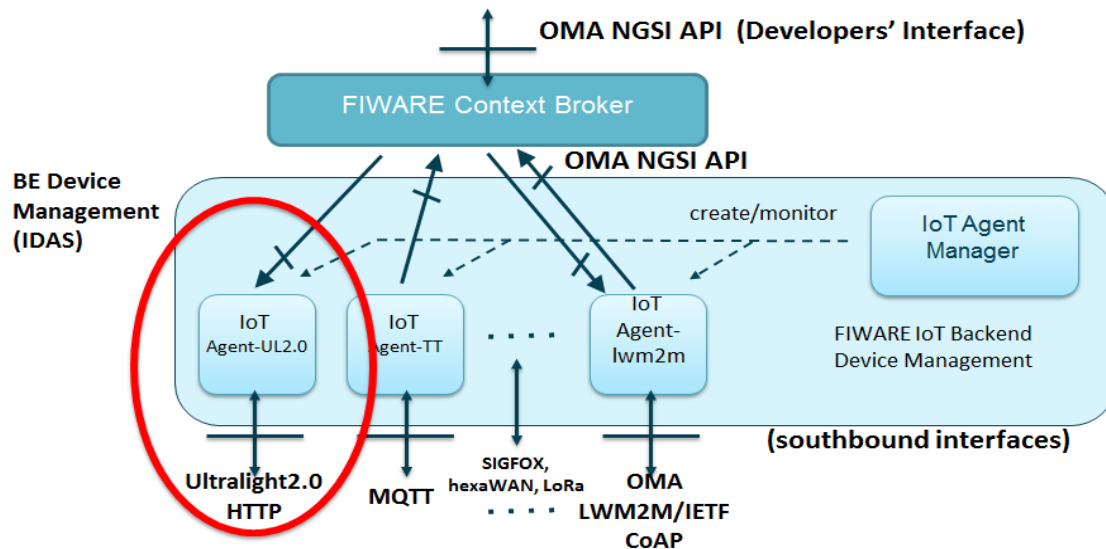
[iotagent](#)

3.3 IDAS IoT Agente UL2.0

Paso 1.Crear un servicio

Paso 2.Crear un dispositivo

Paso 3.Envíar una medida



3.5 Creación de dispositivo

HTTP POST:

<http://{{IP Servidor}}:4061/iot/services>

Headers: {'content-type': 'application/json'; "Fiware-Service: **[myservice]**"; "Fiware-ServicePath: /"}

Payload:

```
{
  "devices": [
    {
      "device_id": "OG_CI1",
      "entity_name": "OG_Centro_Innovacion_terminal1",
      "entity_type": "device",
      "timezone": "Europe/Madrid",
      "attributes": [
        {
          "object_id": "t",
          "name": "temperature",
          "type": "number"
        },
        {
          "object_id": "hr",
          "name": "humidity_relative",
          "type": "number"
        }
      ]
    }
  ]
}
```

3.6 Enviar una medida con UL2.0

HTTP POST:

http://{{IP Servidor}}:7896/iot/d?k={{key}}&i=OG_CI1

Headers: {'content-type': 'application/json'; "Fiware-Service: **[myservice]**"; "Fiware-ServicePath: /"}

Payload:

2016-11-15T15:15Z|t|25#hr|80

Payload

<medida1>#<medida2>#(...)

Donde <medida1> debe seguir el formato <fecha__medida>|<id_alias>|<valor> , en el que <fecha__medida> es la fecha de la lectura en formato ISO 8601 este campo es opcional, <id_alias> es la magnitud medida y <valor> es el valor interpretado. Se envían varios valores se separan por #

3.7 Envío de datos al CB por IDAS

ENVÍO DE DATOS DE SENSOR SIMULADO A ORION CONTEXT BROKER UTILIZANDO IDAS 4.0 CON PROTOCOLO IOT Agent UL.2.0

Envío de datos simulados utilizando POSTMAN

