

Connecting WiseTown to FIWARE IoT

1. What is FIWARE IoT

FIWARE is an open software ecosystem provided by the FIWARE Community (<http://www.fiware.org>).

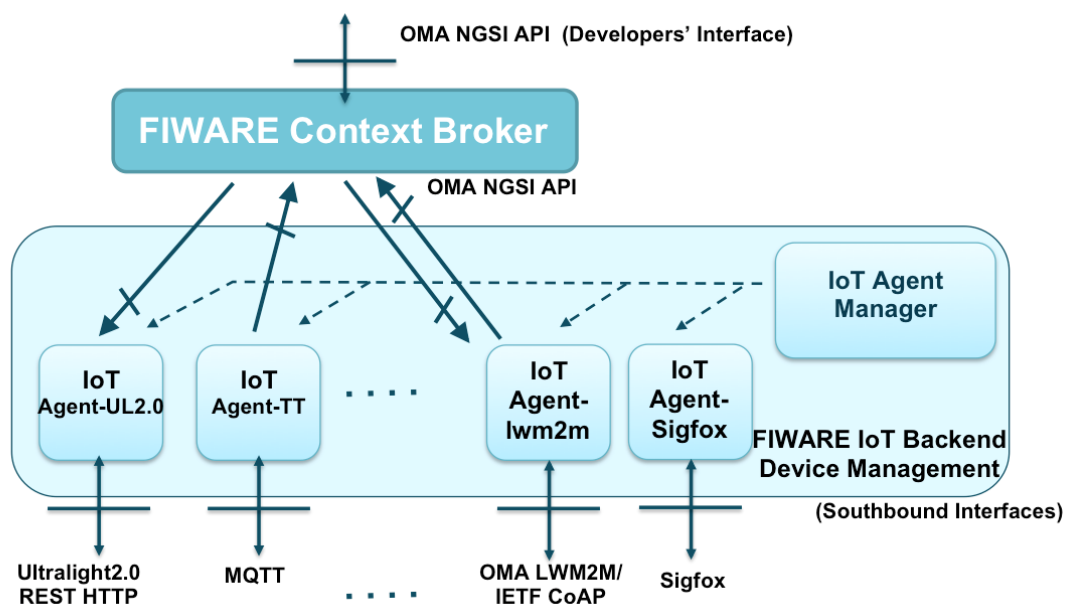
FIWARE exposes to developers Data Context elements or entities (JSON objects) with attributes and metadata with a uniform REST API (NGSI9/10). NGSI is now being adopted by more than 60 smartcities worldwide:

- <http://connectedsmartcities.eu/open-agile-smart-cities>

Orion Context Broker is available at: <http://bit.ly/github-ContextBroker>

FIWARE IoT is an opensource software stack aiming to bring Data-level interoperability to the complex salad of standards and protocols in the world of IoT today.

FIWARE IoT is able to expose -by means of the Orion Context Broker component- all IoT devices information and commands using the Data Context API (NGSI).



Any IoT standard or proprietary protocol can be connected to FIWARE via the IoT-Agent components. Currently FIWARE IoT is providing IoT-Agents for:

- Ultralight2.0/HTTP: http://bit.ly/fiware_iot-ul20
- MQTT/TCP: http://bit.ly/fiware_iot-ul20
- LWM2M/CoAP: http://bit.ly/fiware_iot-lwm2m-coap
- SIGFox Cloud: <https://github.com/telefonicaid/sigfox-iotagent>

Additionally, any new standard or proprietary protocol might be implemented using this node.js IoT-Agent library:

- <https://github.com/telefonicaid/iotagent-node-lib>

or this C++ library:

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



2. How to use WiseTown with FIWARE

WiseTown can be integrated with the above-described FIWARE IoT ecosystems using the following IoT-Agent(s):

- UltraLight 2.0

3 Configuring WiseTown with UltraLight 2.0 IoT-Agent

3.1 Intro & Requirements

WiseTown platform is capable of acting as a virtual device provider for weather data read by different kinds of weather stations. All the different devices are homogenized under just one type of entity in the Context Broker: WiseTownWeatherStation.

Apart access to the UltraLight 2.0 IoT Agent, there are no special needs to use WiseTown with the FIWARE IoT stack.

At the moment WiseTown is not integrated with Keystone IDM so access to the IoT Agent or Context Broker has to be open.

3.2 Configuring **WiseTown** to send data to FIWARE IoT

In the web.config are defined the following parameters:

- PROVISIONING_ENDPOINTS: the url that will be used to access the provisioning API.
- UL_ENDPOINTS: the url that will be used to send the devices observations with the UltraLight 2.0 payloads.
- FIWARE_SERVICE: the name of the IoT service under which data will be accessible.
- FIWARE_SUBSERVICE: the name of the subservice under which data will be accessible.

3.3 Accessing **WiseTown** data at FIWARE Context Broker

```
curl -X POST -H "Content-Type: application/json" -H "Accept: application/json" -H
"fiware-service: wisetownweather" -H "fiware-servicePath: /stations" -d '{
  "entities": [
    {
      "type": "WiseTownWeatherStation",
      "isPattern": "false",
      "id": "Perugia Santa Giuliana"
    }
  ]
}' "http://<ContextBrokerNameOrIP:port>/v1/queryContext"
```

3.4 Preparing FIWARE IoT to send commands to **WiseTown**

Our device implementation does not provide commands to be executed.

3.5 Sending commands to **WiseTown** from FIWARE IoT

Our device implementation does not provide commands to be executed.