

**Private Public Partnership Project (PPP)**

Large-scale Integrated Project (IP)



#### **D.11.5.1: Live demos**

**Project acronym:** FI-Core

**Project full title:** Future Internet - Core

**Contract No.:** 632893

**Strategic Objective:** FI.ICT-2011.1.7 Technology foundation: Future Internet Core Platform

**Project Document Number:** ICT-2013-FI-632893-WP1-D.11.5.1

**Project Document Date:** 2015-04-14

**Deliverable Type and Security:** PU

**Author:** ilknur Chulani (ATOS), Santiago Martinez Garcia (TID), Davide Zerbetto (Eng), Javier Soriano (UPM), Pierangelo Garino (TI), Angel Hernandez (ET), José Jaime Ariza (ET), Silvia Castellvi Catala (ATOS), Jesus Benedicto Cirujeda (ATOS)

**Contributors:** Jose Manuel Cantera Fonseca (TID), Toni Alatalo (Admino), Monica Franceschini (Eng), Javier Lopez (Naevatec), Luis Lopez (Naevatec), Christian Salerno (Consoft), Sami Jylkka (Cyber), Philipp Slusallek (DFKI)

# Table of Contents

<b>1</b>	<b>INTRODUCTION .....</b>	<b>4</b>
<b>2</b>	<b>OBJECTIVES AND SCOPE .....</b>	<b>4</b>
<b>3</b>	<b>APPROACH .....</b>	<b>5</b>
<b>4</b>	<b>DEMOS .....</b>	<b>5</b>
4.1	SMART CITY DEMO .....	6
4.1.1	<i>Demo idea</i> .....	6
4.1.2	<i>Demo contacts</i> .....	6
4.1.3	<i>Demo scenario</i> .....	6
4.1.4	<i>Demo design</i> .....	6
4.1.5	<i>GEs used</i> .....	7
4.1.6	<i>Time plan</i> .....	8
4.2	TRANSPORT DEMO .....	8
4.2.1	<i>Demo idea</i> .....	8
4.2.2	<i>Demo contacts</i> .....	9
4.2.3	<i>Demo scenario</i> .....	10
4.2.4	<i>Demo design</i> .....	15
4.2.5	<i>GEs used</i> .....	17
4.2.6	<i>Time Plan</i> .....	18
4.3	BUSINESS DEMO .....	18
4.3.1	<i>Demo idea</i> .....	18
4.3.2	<i>Demo contacts</i> .....	18
4.3.3	<i>Demo scenario</i> .....	19
4.3.4	<i>Demo design</i> .....	20
4.3.5	<i>GEs used</i> .....	21
4.3.6	<i>Time plan</i> .....	21
4.4	ROBOTICS DEMO: FROM INTERACTIVE TO AUTONOMOUS ROBOT MANAGEMENT IN FIWARE.....	22
4.4.1	<i>Demo idea</i> .....	22
4.4.2	<i>Demo contacts</i> .....	23
4.4.3	<i>Demo scenario</i> .....	23
4.4.3.1	Step1: Managing interactively robots .....	23
4.4.3.2	Step 2: Navigation to a target Location.....	24
4.4.3.3	Step 3: Achieve a specified goal – Firefighting Robot .....	25
4.4.4	<i>Demo design</i> .....	25
4.4.4.1	Design of step 1: Teleoperation as a Service (TaaS).....	25
4.4.4.2	Design of step 2: Navigation to a target location .....	28
4.4.4.3	Design of step 3: Achieve a specified goal – Firefighting Robot.....	29
4.4.5	<i>Time Plan</i> .....	31
4.5	MANUFACTURING DEMO .....	32
4.5.1	<i>Demo idea</i> .....	32
4.5.2	<i>Demo contacts</i> .....	32
4.5.3	<i>Demo scenario</i> .....	32
4.5.4	<i>Demo design</i> .....	33
4.5.5	<i>GEs used</i> .....	37
4.5.6	<i>Time Plan</i> .....	38

<b>5</b>	<b>DELIVERY MECHANISM.....</b>	<b>38</b>
<b>6</b>	<b>INFORMATION ON OTHER DEMOS IN THE PROJECT .....</b>	<b>38</b>
	<b>ANNEX A. DETAILED CONTACT INFORMATION .....</b>	<b>39</b>
	<b>ANNEX B. CONTEXT ELEMENTS FOR TRANSPORT DEMO.....</b>	<b>40</b>

# 1 Introduction

This document is devoted to provide specifications for several prototypes that showcase the usage of FIWARE Generic Enablers (GEs) and their reference implementations (GERis) in real life scenarios.

The prototypes are intended to build reference examples for the FIWARE Developer Community and to support training and dissemination activities.

In the following sections, two and three, the purpose and scope of these demos is explained in more detail, together with the approach taken. In section four, each demo is described, providing details on the scenario, the design, the generic enablers used and time plan.

## 2 Objectives and scope

*The creation of Live demos in FIWARE* aims to build prototypes in different domains, namely: *Smart Cities, Transport, Robotics, Business (Smart Energy) and Manufacturing*, in order to demonstrate the capabilities of the FIWARE platform. These prototypes' objectives are threefold:

- Support FIWARE dissemination and training activities with real content to be demonstrated. The code associated with the demos should be available as open source so that developers in bootcamp/training events can inspect them and take it as a basis for development of their first apps using FIWARE.
- Provide reference examples of real use case applications (i.e., solving real life problems) that make use of a wide set of FIWARE GEs from different chapters, thus as complete as possible.
- Give FIWARE GE developers direct feedback on the usability and applicability of their own components for specific purposes, and this way improve them for the benefit of the whole community.

Our main assumptions are the following:

- All demos will be running on the FIWARE Lab environment. Additionally, documentation will be provided describing the architecture and how to install and run the software. All this information, together with the software of the demo applications will be available as a Github project for each demo.
- Demos will focus on the Generic Enablers. Specific enablers or incubated generic enablers will only be used if absolutely necessary, as demonstrating these is not the main goal. The architecture of the demos will be heavily based on the FIWARE core platform.
- Demos will focus on scenarios which use GEs across multiple chapters to demonstrate the potential of the whole platform, with as many GEs as possible.

- Demos will be developed with the intention of showing code. The FIWARE Developer Community should be able to use them as reference examples, and easily create similar applications using the provided demo code, the tutorials, the documentation, etc.

### 3 Approach

A collaborative and iterative approach will be followed for developing the demos.

*Collaboration among partners:*

Five dedicated workgroups have been set up for the development of the demos, based on the expertise of partners in the GEs and specific domains involved:

Demo domain	Work group
Smart Cities	<b>ATOS</b> , TID, Naevatec, UPM, Admino, Cyber
Transport	<b>TID</b> , Cyber, OKFN
Business (Smart Energy)	<b>Engineering</b> , UPM
Robotics	<b>Etxe-Tar</b> , TI, Consoft, Naevatec, UPM
Manufacturing	<b>ATOS</b> , ET

*Table3.1. Demo work groups*

Partners with most effort in the task will be leading these work groups, as indicated in **bold**.

We will also be collaborating with the chapter leaders, chapter architects and GE owners in the rest of the project, in addition to the partners listed in these workgroups, in order to get support for the GEs demonstrated.

*Iterative development:*

All demos will produce a public release every six months (from *M13* to *M25*), collecting feedback from project partners and from external people (especially developers and commercially oriented people). Every upgraded version will incorporate the received feedback, and build on the previous version to demonstrate additional features.

### 4 Demos

Demos will be targeting five different domains: *Smart Cities*, *Transport*, *Business (Smart Energy)*, *Robotics* and *Manufacturing*. In the following sub-sections, the details will be provided on the idea, contacts, scenario, design and time plan for each planned demo.

## 4.1 Smart city demo

### 4.1.1 Demo idea

This demo aims to showcase the usage of FIWARE *Generic Enablers* in the Smart City domain. It will be showcasing various electronic services that will be available to the citizens of *Santander (Spain)*. Such services will make use of the open data available in the FIWARE *Lab* for smart cities.

### 4.1.2 Demo contacts

Demo lead: **ATOS**

Contributing partners: TID, UPM, Naevatec, Adminotech, Cyber, DFKI

More detailed contact information can be found in *Annex A*.

### 4.1.3 Demo scenario

In this demo, several scenarios intended to offer citizens value added services will be considered, exploiting data coming from different sensors deployed around the city of Santander. Sensor data, including traffic data, buses location, temperature, light levels, noise levels, humidity, pollution, etc. will be retrieved through open data made available by the city on the FIWARE Lab.

In year one, the scenario will focus on the public transportation system. The demo application will suggest different bus routes to reach a particular point of interest based on current geolocation of the user and the destination. Incrementally more features will be added, such as displaying to the passengers on a bus information on other points of interest around the next bus stop they are approaching. Even some 3D user interface models might be provided to help users visualise how to walk to a point of interest after getting off a bus. Another feature considered will be to add information on live events happening that day around town, and again suggesting the best way to get there.

In year two, other smart city scenarios will be explored based on the exact types of open data available at that time. For instance, an application that suggests which outdoor attractions (parks, beaches, etc.) are best to go to at a given moment in time can be considered based on live measurements of temperature humidity, air quality, and traffic conditions on the way there.

More scenarios can be incrementally added in year two as more open data becomes available from the city.

### 4.1.4 Demo design

The Smart City demo will consist of several layers:

- *Frontend* - in the form of a rich user interface (making use of *Advanced Web UI* chapter GEs) available through mobile devices and desktop browsers.
- *Composition and Data visualization layer* - making use of *Apps and Services* chapter GEs

- *Data access layer* -consisting of the *Data/Context Management* chapter GEs.
- *Data layer* - consisting of open data available from the Santander city on the FIWARE Lab. The dynamic data available from Santander at the time of writing include: traffic data, location of buses, temperature, light levels, noise levels, humidity and pollution. Some static data such as data on points of interest, bus routes, maps, etc. will be used as well.

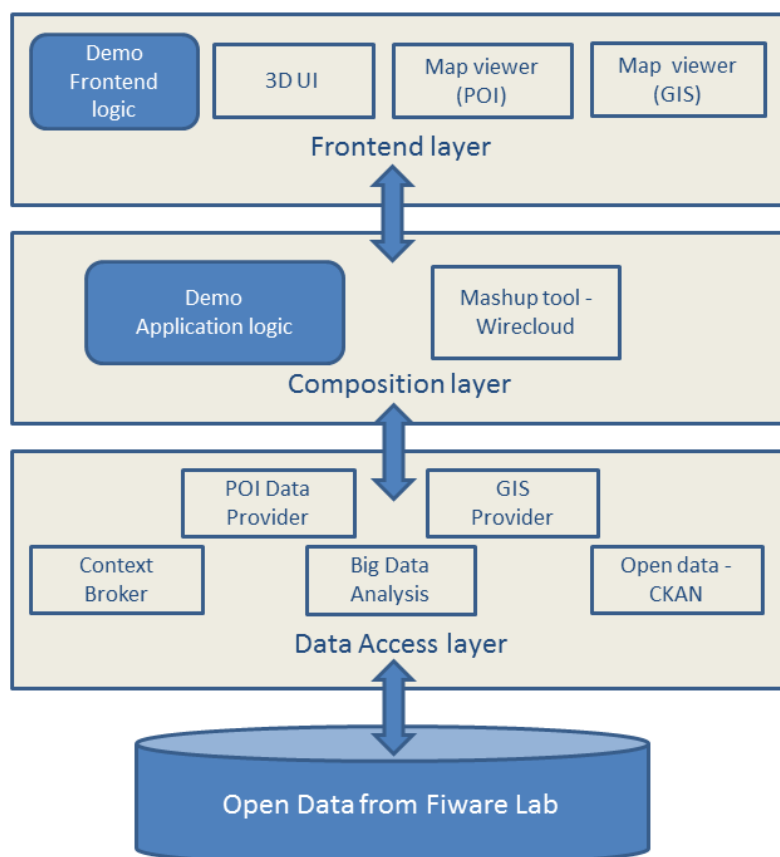


Figure1.1. Architecture diagram for Smart City demo

#### 4.1.5 GEs used

The following *Generic Enablers* will be used by the demo:

- [3D UI-WebTundra](http://catalogue.fiware.org/enablers/3dui-webtundra)<sup>1</sup> : to provide a 3D model on a map to visualise the path to walk between the bus stop and the point of interest.
- [POI data provider](http://catalogue.fiware.org/enablers/poi-data-provider)<sup>2</sup> : to retrieve info on the points of interest that the bus is approaching to.
- [GIS data provider - Geoserver/3D](http://catalogue.fiware.org/enablers/gis-data-provider-geoserver3d)<sup>3</sup> : to determine the current coordinates of the user, to determine the starting point for the bus journey.
- [Mashup tool - Wirecloud](http://catalogue.fiware.org/enablers/mashup-tool-wirecloud)<sup>4</sup> : to compose the demo application from different FIWARE services available, for instance for connecting the visual elements to POI data provider and GIS data provider through the Map Viewer widget.

<sup>1</sup> <http://catalogue.fiware.org/enablers/3dui-webtundra>

<sup>2</sup> <http://catalogue.fiware.org/enablers/poi-data-provider>

<sup>3</sup> <http://catalogue.fiware.org/enablers/gis-data-provider-geoserver3d>

- [Context broker - Orion](#)<sup>5</sup> : to get the context data such as bus location data, the geo location of the user, etc., and to register for notification on events such as approaching bus stop.
- [Big data analysis - Cosmos](#)<sup>6</sup> : to analyse the sensor data available on FIWARE Lab from the city of Santander.
- [CKAN](#)<sup>7</sup> : to get access to open data on FIWARE Lab.

#### 4.1.6 Time plan

##### *Phase I:*

In the first year, focus will be put on the public transportation scenario described in Section 4.1.3. The public release on *M13* will deliver the initial version of a system suggesting bus routes to reach a particular point of interest based on the current geolocation of user. A 3D user interface will be built in order to help passengers visualise how to walk to a point of interest after getting off the bus at the planned bus stop.

##### *Phase II:*

The public release on *M18* will add more features such as displaying (on the bus) information about other points of interest around the next stop bus is approaching, or live events happening around town, and how to get there. More advanced user interface features will be provided as well.

##### *Phase III:*

In the third phase, other smart city scenarios will be explored in order to extend the demo, possibly with the outdoor attractions scenario mentioned in Section 4.1.3. Exact scenario to be implemented for the *M25* public release will be decided based on the type of open data available from the city at that time.

## 4.2 Transport demo

### 4.2.1 Demo idea

This demonstration illustrates how the FIWARE Platform is being successfully used in the technological modernization of Seville's (Spain) port which manages goods and freight transport made by trains and river ships. This initiative is called "**Tecnoport**" and will allow, thanks to the capabilities offered by FIWARE, to convert Seville's port in an integrated, intermodal logistics node. To this aim, a new information infrastructure is going to be provided consisting in an **Integrated Service Platform** based on FIWARE. The FIWARE components to be used are, among others, the IoT

---

<sup>4</sup> <http://catalogue.fiware.org/enablers/application-mashup-wirecloud>

<sup>5</sup> <http://catalogue.fiware.org/enablers/publishsubscribe-context-broker-orion-context-broker>

<sup>6</sup> <http://catalogue.fiware.org/enablers/bigdata-analysis-cosmos>

<sup>7</sup> A new Generic Enabler for Open Data Access Management. It is not yet integrated in the FIWARE catalogue as it is still under development.



Backend Device Management GE and the NGSI Context Broker implementation (Orion), together with the necessary communications network.

Seville's port manages goods and freight transport that arrives and departs from the port through different transport channel modes, mainly train, railways and river.

Tecnoport is composed of three verticals (and systems) addressing the main areas in port logistics:

**FPS** (*Ferro Port System*)

- Provides a system able to manage automatically rail traffic at the port premises.
- Improves the knowledge about train delays and unexpected rail traffic events.
- Stores information about trains traffic, estimated arrivals, and rail issues.

**eRIO**

- Provides a help system for navigating through the "Guadalquivir" river
- Improves the information about the navigation conditions for the ships.
- Stores weather and tide conditions on the platform gathered from sensors aboard.

**CUTS** (*Cooperative Unitized Tracking System*)

- Provides a container tracking system, connected even during navigation on High Seas from Seville to Canary Islands.
- Improves the security and provides information about the status of the goods in the container.
- Stores location, temperature and door opening events on the platform.

The data provided by these three systems will be used as part of the live demo, which ultimately has the objective of showing how the FIWARE based Integrated Service Platform can manage, in an efficient and optimized manner, the data provided by all of these systems, allowing also the interaction with each of them by using the standard FIWARE NGSI interface.

The visualization part will consist on a 3D representation of this data, which will be made available as Open Data as well. This will be based on the *GIS GE* and *CKAN GE* from FIWARE, respectively.

As a result, it will be demonstrated how FIWARE technologies can be applied to turn Tecnoport into a multimodal logistics node.

#### 4.2.2 Demo contacts

Demo lead: **TID**

Contributing partners: Cyber, OKFN

More detailed contact information can be found in *Annex A*.

### 4.2.3 Demo scenario

The demo scenario consists mainly on a FIWARE based **Integrated Service Platform** that will interact with the previously described FPS, CUTS and eRIO systems, following the FIWARE NGSI protocol. The operations of ships, trains and freight containers will be optimized, and the platform will take as input the data gathered from sensor networks and communication elements specifically deployed for each of these systems.

Many insights will be provided to this platform regarding logistics area. For example, the freight container ownership may change during its journey and it will be necessary to have a dynamic entity ownership managed on the Integrated Service Platform. In order to fully cover these insights, the necessary Context Entities will be defined so that the context information that each of these systems is able to provide to the Integrated Service Platform can be described.

As a result of this, a multimodal logistics port node will be created. Tracking of the location of a specific container at any time, e.g. when it arrives by train at the port, and leaves the port by ship, is just an example of the advantages a multimodal logistics port node can provide.

In particular, the live demo will deal with a specific use case for each of the previously described port systems (FUPS, CUTS and eRIO), managing the visualization and interaction with these systems in an integrated and uniform way through the FIWARE based Integrated Service Platform:

- *FPS (Ferro Port System)*

This scenario will involve the management of the freight transportation trains and railway operations in the port of Seville, including the automatic freight load/unload destination point selection for trains arriving in Seville's port terminal, based on the characteristics of the train (e.g.: train's length) and the automatic actuation on the railway elements to enable the train reaching the selected destination point in the terminal for loading/unloading the freight.

FPS and a specifically deployed sensor network will be used in one of the railways which arrive at the port of Seville during train freight transport. Once the train arrives at the port terminal facilities, the registered sequence of events will be the one described on the figure below:

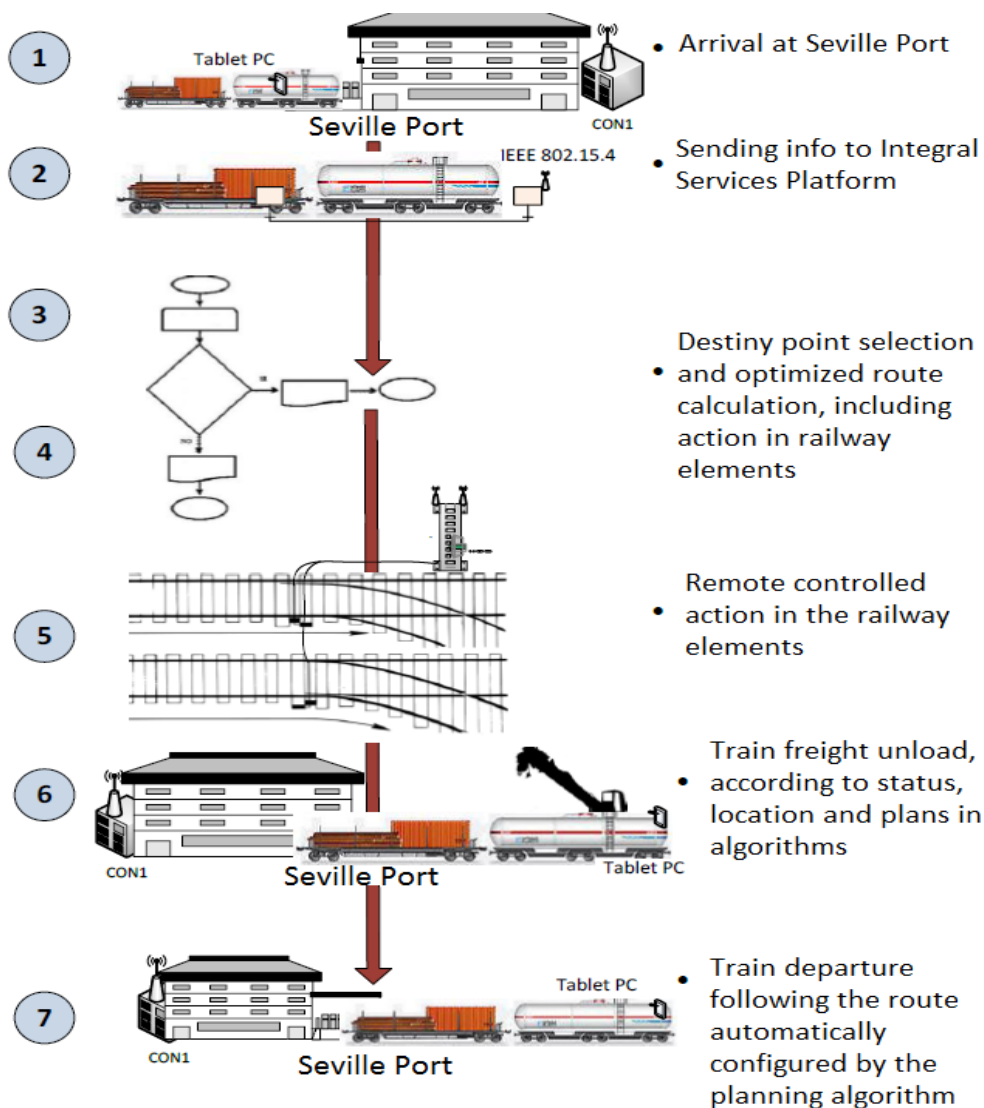


Figure2.1. Automatic Train Operations Management Demo scenario

The explanation of the sequence shown before is the following:

1. The train arrives at Seville's port terminal. The wireless communication terminal located in the train (Tablet PC) registers in the FPS system through the concentrator (CON1) from the communications network in Seville's port terminal.
2. FPS determines the length of the train by using the sensor network deployed, and transmits this information to the Integrated Service Platform.
3. The most appropriate terminal destination point for the train is selected, based on the information received, and is communicated to the FPS system.
4. Also, the necessary commands for actuating on the railway elements are sent to the FPS system, in order to enable the train to reach the selected destination point in the terminal.

5. FPS controls that the railway elements are set correctly for allowing the train's route to the selected destination point in the terminal.
6. Train unloads the freight when appropriate according to its current position transmitted by the Tablet PC to the FPS system, and the destination point indicated by the FPS system.
7. Once finished, FPS indicates it to the Integrated Service Platform, and the train's exit route from the port terminal is selected, actuating on the necessary railway elements to allow this.

- *eRIO (River Navigation Aid System)*

This scenario involves the management of a navigation aid system for the "Guadalquivir" river in Spain, which is the route that ships transporting goods and freight use for arriving at Seville's port.

In order to gather relevant context information about the "Guadalquivir" river, an intelligent buoy sensor based network will be deployed on specific areas of the "Guadalquivir" river route. This sensor network will provide context information about water conditions and quality, weather conditions (temperature, relative humidity, wind speed or visibility), sea status and ship presence in the river.

Ships registered on the eRIO system will be provided with the best route possible to reach the Seville's port, based on the context information obtained.

The information gathered from this sensor based network will be transmitted to the Integrated Service Platform for route selection, monitoring and visualization purposes.

The geographic extension from the "Guadalquivir" river that will be covered for this demo scenario is 12km (from "Coria" to "Darsena del Batan" regions), which is considered to be an important enough extension for validating the solution.

The figure below shows the different location points where specific sensors will be deployed in the context of this demo scenario:

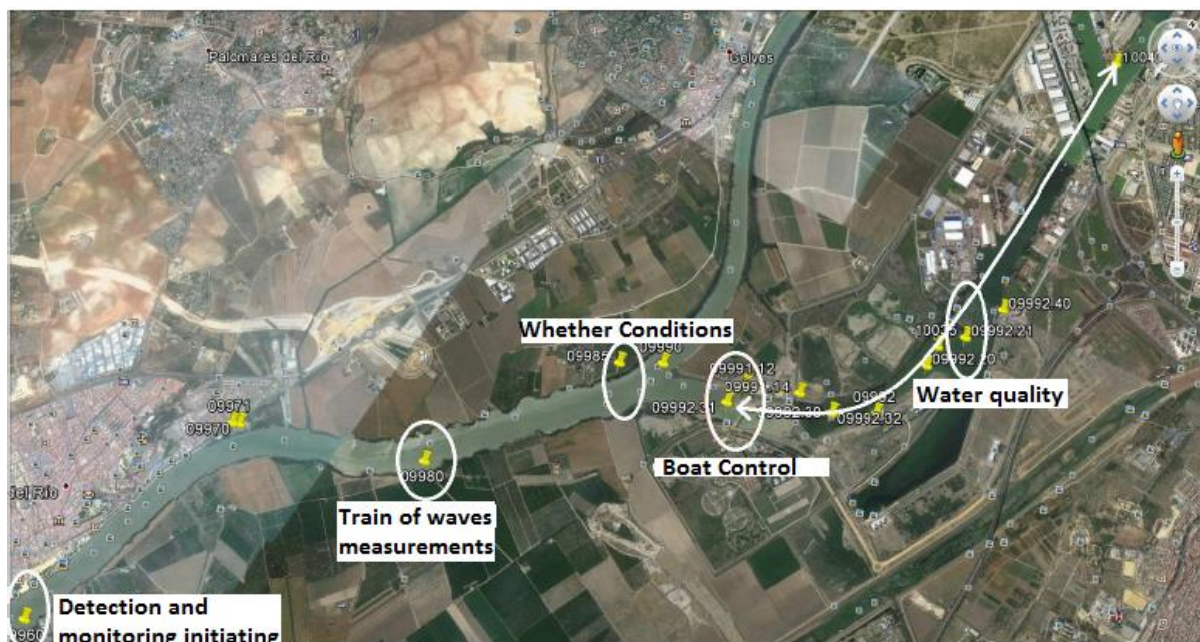


Figure2.2. “Guadalquivir” River Navigation Aid System Demo scenario

- *CUTS (Cooperative Unitized Tracking System)*

This demo scenario consists on tracking in real time the location of the freight containers, involving different means for their transportation.

As an example, a Madrid-Seville-Canary Islands transportation route, using train and boat for reaching the destination point will be used for this demo. The purpose is to fully cover a typical freight/goods transportation process, going from the distributor premises to the final destination point and involving as many actors from the logistics transport chain as possible. This is the reason why the transport route monitored for the demo involves both train transport and boat transport means.

Freight containers will include a *User Terminal (TU)* that will allow them to be registered in the CUTS system in order to provide context information about their location and status to the Integrated Service Platform. This platform will take care of managing in real time the dynamic context information about the freight container entities (e.g.: ownership, current transport mode, location etc.) received from the CUTS system.

The event sequence for the Madrid-Seville-Canary Islands freight transport demo scenario is shown in the figure below:

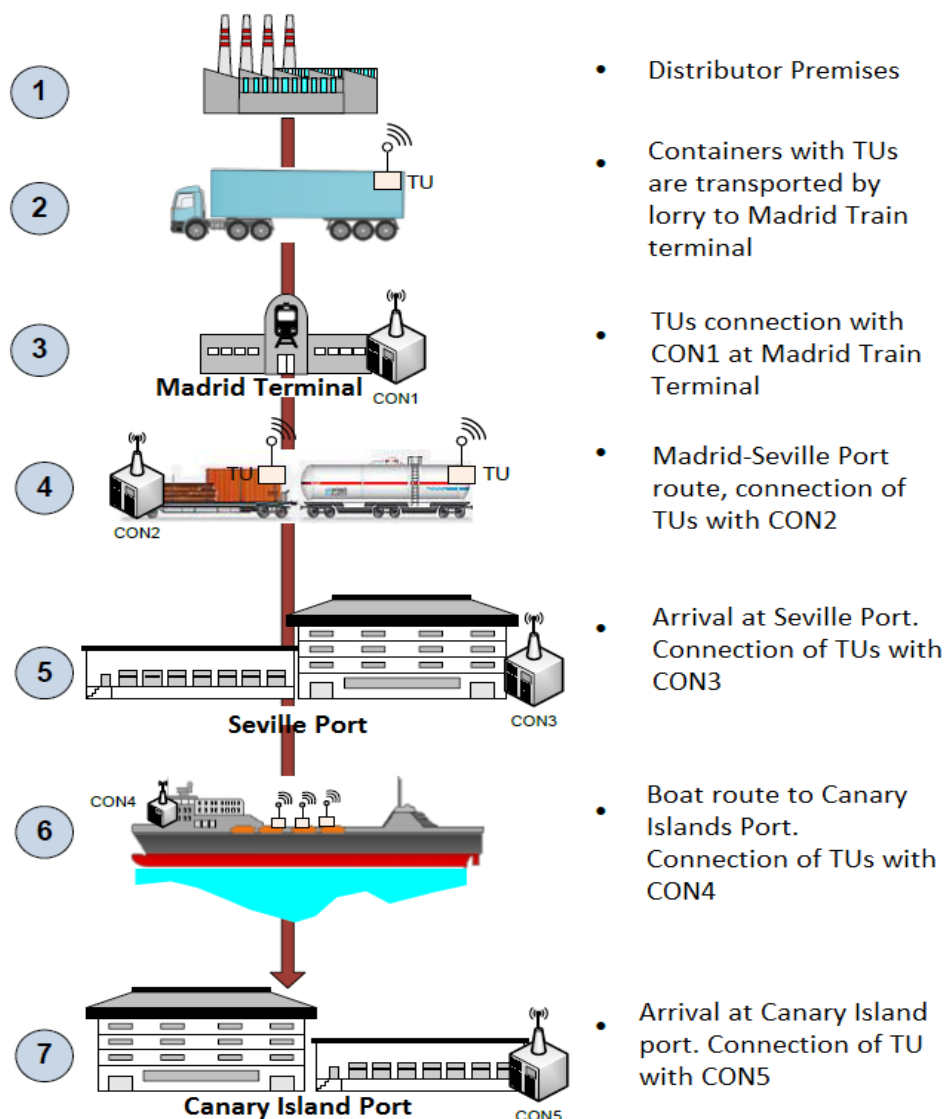


Figure2.3. Container Tracking Demo scenario

The explanation of the sequence shown above is the following:

1. The freight is shipped on a container owned by the distributor which includes an active User Terminal (TU) that the distributor has previously registered on the CUTS system.
2. The containers are transported to the Madrid train terminal.
3. While accessing the train terminal, the TU is detected by the Concentrator (CON1) from the communications network in the train terminal area. Location and status information of the freight container is transmitted periodically to the Integrated Services Platform.
4. When the freight container is placed on the train, its TU is detected by the Concentrator (CON2) on the train, and its location and status information is transmitted periodically to the Integrated Services Platform, during the route from Madrid Train Terminal to Seville's port.

5. When the train arrives at the Seville's port terminal, the container's TU is detected by the communications Concentrator (CON3) at Seville's port.
6. Finally the freight containers are placed on a ship that transports them to the final destination port in the Canary Islands. The *Concentrator* (CON4) present in the ship's communication network detects the User Terminal from the freight container and starts transmitting periodically its associated location and status information to the Integrated Services Platform.
7. Upon arrival at the Canary Islands port, the *Concentrator* (CON5) from the communications network in the port detected the container's TU and transmits the location and status information to the Integrated Services Platform.

In order to cover the scenarios described above, several context elements relevant for each of the Tecnoport systems (FPS, eRIO, CUTS) are being defined. These can be found in *Annex B. Context Elements for Transport Demo*.

#### 4.2.4 Demo design

The core element for Tecnoport is given by the FIWARE based Integrated Service Platform, which is in charge of integrating the context information and services provided by the FPS, CUTS and eRIO systems, as has been explained in the previous sections.

It is envisaged that in the future, not only these three, but also further logistics systems will need to be integrated in the platform as well, each of these providing relevant context information for the Tecnoport multimodal logistics node. It is of utmost importance for Tecnoport that the data generated and the interaction with these individual systems is managed in an integrated way, and also that the creation of new services for the multimodal logistics node using the data coming from these and future systems is enabled through an open, standard and well defined API, that internal and even external developers can make use of. These are the main reasons why the Tecnoport's multimodal logistics node design is based on an Integrated Service Platform containing instances of the following FIWARE GEs (see high level logic architecture figure below, for more details):

- *Orion Context Broker*: it will be in charge of managing the context information generated by the FPS, CUTS, and e-RIO systems described before, using the standard FIWARE NGSI operations. In turn, it will also allow the frontend's data visualization components to consume this information using the FIWARE NGSI interface, as well.
- *IoT Backend Device Management*: it will be in charge of the direct interaction, when necessary, with the sensor networks deployed in the different demo scenarios. Interactions with the sensors (southbound interface) will be performed by the corresponding *IoT agents*, which in turn, will also provide the adaptation to NGSI (northbound interface), so that the *Context Broker* can receive the context information generated from the sensors.
- *Big Data Analysis*: it will be in charge of processing the context information and obtaining valuable insights based on the data provided by the *Context Broker*, such as the routing optimization processing algorithms.

- *CKAN Open Data portal*: dynamic data received from the *Context Broker* will be stored on a short term historic repository and published as Open Data using *CKAN*, including also the relevant static datasets.
- *GIS -Geographic Information System*: this will be used for obtaining a 3D visualization of the geolocated data available from the demo scenarios described before, based on the relevant context information available for each of them.
- *Identity management & Access control*: these will be used to guarantee security when accessing the different data and platform resources used for Tecnoport.

The figure below shows the high level logic architecture for the Tecnoport multimodal logistics node, including the interactions between the different GEs and the rest of the systems that form Tecnoport. FIWARE GEs are shown as blue boxes (see section 4.2.5 for an additional description for these GEs, and reference links to the FIWARE catalogue for each of them):

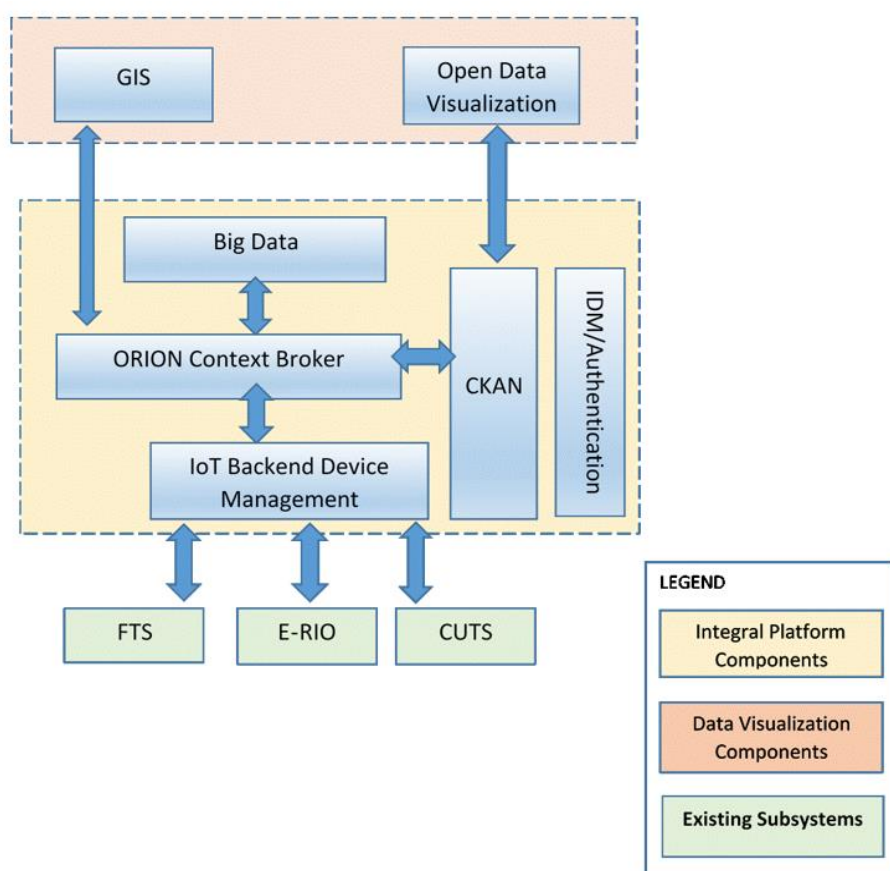


Figure 2.4. Tecnoport High Level Logical Architecture

Additionally, some of the FIWARE Cloud chapter GEs will be used for managing the Cloud based infrastructure dedicated for Tecnoport, however it will not be in the scope of this demo to show how these are used.



#### 4.2.5 GEs used

- [IoT Backend Device Management](#)<sup>8</sup>

In the context of this demo, it enables the Integrated Service Platform interaction with the various sensors and actuators deployed in the trains and railways (FPS demo scenario), boats (eRio demo scenario) and freight containers (CUTS demo scenario), adapting the various technologies, and protocols used by these devices to FIWARE NGSI.

The interaction will mainly consist on the *IoT Backend Device Management GE* receiving measurements data (context information) from these sensors/actuators and sending configuration commands that FPS, eRIO and CUTS systems can use in order to actuate on them.

- [Publish Subscribe Context Broker \(Orion\)](#)

*Orion Context Broker* will be used for gathering the NGSI based context information received from the *IoT Backend Device Management GE* and making it available to the rest of the FIWARE GEs in the platform, including the data visualization/frontend layer, through the standard *Context Broker* NGSI interface.

After the Integrated Service Platform processes the data, the *Context Broker* will make available the updated context information relevant to the FPS, eRIO and CUTS systems (e.g.: the selected or recommended transport routes) back through the NGSI interface.

- [Big Data Analysis](#)

The *Big Data Analysis GE* will be in charge of processing the context information received from the *Orion Context Broker* and generating the necessary insights for the FPS, eRIO and CUTS systems, in the form of updated context information.

- *Short Term Historic Repository CKAN*

*Publish Subscribe Context Broker GE* will interact with the *Short Term Historic Repository CKAN GE* for storing historic context information, and *CKAN* views will be used to visualize both dynamic and static available datasets from the FPS, eRIO and CUTS systems.

Different access permissions could be defined in *CKAN* for the available datasets, however for this demo purposes it is envisaged they will be available as Open Data.

- [GIS Data Provider/Geoserver 3D](#)

For this demo, the *Publish Subscribe Context Broker* will interact with the *GIS server* and *GIS backend database* in order to store geolocated context information that will be shown on a map based 3D visualization client.

---

<sup>8</sup> <http://catalogue.fiware.org/enablers/backend-device-management-idas>

## 4.2.6 Time Plan

### *Phase I: M13*

This phase will focus on integrating FIWARE with the FPS Tecnoport system. Particularly, data visualization/publication through CKAN, including dynamic and static historic context information, and context data access and acquisition through FIWARE NGSI APIs will form the main focus of integration.

The demo might include simulated data for replicating the demo behaviour at any time.

### *Phase II: M18*

This phase will extend the demo scope, including at this point the integration with the eRIO Tecnoport system as well, and visualizing the geolocated data based on a 3D map representation.

Similarly to the previous phase, simulated data might be used at any time.

### *Phase III: M25*

The final phase of the live demo will enrich further its functionality by integrating the FIWARE based platform with the CUTS system as well. The route selection and recommendation functionalities based on the context information provided by the FPS, eRIO and CUTS systems will be validated.

## 4.3 Business demo

### 4.3.1 Demo idea

The objective of this demo is to show how Apps Chapter GEs can be used to create a fully-fledged data market with extended data publication, discovery, visualization, analysis and monetization capabilities.

### 4.3.2 Demo contacts

Demo lead: **Engineering**

Contributing partners: UPM

More detailed contact information can be found in *Annex A*.

### 4.3.3 Demo scenario

The Business demo base architecture will be inspired by the experience gained in the FINESCE<sup>9</sup> project, in particular in the successful WP4 Energy Marketplace trial site located in Terni - Italy<sup>10</sup> led by Engineering.

Considering the FINESCE trial as a reference architecture, we'll implement a fully-fledged data market with extended data publication, discovery, visualization, analysis and monetization capabilities using the Apps Chapter GEs. These GEs are already integrated with a number of GEs from other chapters that will also be used in this demo, including CKAN, Context Broker, and IdM.

In particular, this demo will demonstrate:

- The use of the business framework GEs (*WStore*, *WMarket*, *Repository RI*, *Revenue Sharing and Settlement System*) for easily enhancing the current architecture of the FINESCE's e-marketplace with a fully-fledged data market providing the following features:
  - Publication of datasets created from the data being collected in the *Context Broker*, with improved support for smart searches and discovery.
  - Support for complex offerings that include both data and dashboards (based on *Wirecloud*) created specifically for visualizing those data.
  - Data access restrictions (this is very important since in some situations they have confidential information that cannot be spread publicly), including:
    - Enforcing access control policies
    - Forcing the users to accept some terms and conditions before using your data (e.g. prohibiting the use for commercial purposes).
    - Charging the users for accessing your "premium" data using different payment models such as single payment, subscription or pay per use.
- The use of *SpagoBI* for providing data analytics, including the generation of enhanced "premium" datasets from the output of these analytics. The data could come from:
  - Context information provided by the *Context Broker (Orion)*
  - Datasets stored in the *Data Portal (CKAN)*
  - *Big Data analysis (Cosmos)*
- The use of *SpagoBI* for performing data analytics on the use of the data being offered (e.g. geo-distribution of consumers, usage patterns, peaks detection, etc.).

---

<sup>9</sup> Future INtErnet Smart Utility ServiCEs, <http://www.finesce.eu/>

<sup>10</sup> [http://www.finesce.eu/Trial\\_Site\\_Terni.html](http://www.finesce.eu/Trial_Site_Terni.html)

### 4.3.4 Demo design

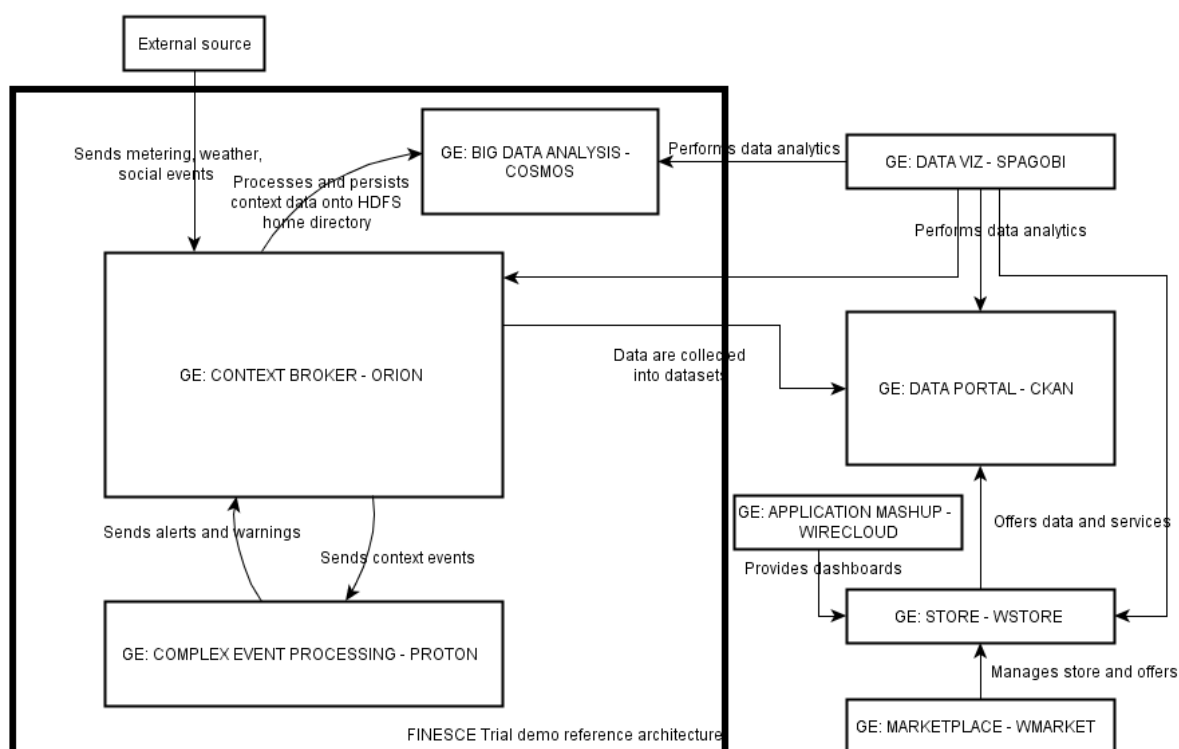


Figure3.1. Architecture of the Business demo

In the above schema, the base reference architecture in the FINESCE trial, on the left (within the box) and the business framework GEs enhancement on the right, can be distinguished.

Data will be collected on Data Portal GE (CKAN) within datasets; those datasets will be offered by the Store GE (WStore) and Marketplace GE (WMarket) in combination with dashboards based on Application Mashup GE (Wirecloud). Access control policies are applied at this stage.

On the other hand, Data visualization GE (SpagoBI) will provide data analytics on different data sources:

- the *Data Portal GE (CKAN)* and the *Big Data Analysis GE (COSMOS)* for core data analyses;
- the *Context Broker GE (ORION)* for context analyses;
- the *Store GE (WStore)* for analyses on usage of the data offered by the *Store* itself.

#### 4.3.5 GEs used

- [Context Broker - Orion](#): this GE will collect all the data coming from the external sources
- [Complex event processing - Proton](#)<sup>11</sup>: this GE will process the data to get alerts and warnings
- [Data Visualization - SpagoBI](#)<sup>12</sup>: this GE will provide data analytics, including analysis of existing data on sales, data API usage, etc. provided by the store / data market, and creation of “Premium” datasets
- [Store - WStore](#)<sup>13</sup>: this GE will be used in order to offer premium datasets and dashboards. It will allow the selling of these digital assets to consumers as well as to application developers, and will be responsible for managing offerings and sales.
- [Marketplace - WMarket](#)<sup>14</sup>: this GE will be used for presenting the consumer with a fully-fledged data market and allow consumers to search for, discover and compare offerings published on different stores
- [Data Portal - CKAN](#): this GE will be used for datasets publication
- [Application Mashup - Wirecloud](#): this GE will provide end users with a tool to easily build visualization dashboards for both the data and the results of the analysis performed on those data. These dashboards will then be available for publication in any web sites and web portals and even be sold through the Store jointly with the data to which the user has gained access.
- [Big Data Analysis - Cosmos](#): this GE will be used as data storage

#### 4.3.6 Time plan

##### *Phase I: M13*

- a. The live demo will show *Data Viz GE (SpagoBI)* analytical capabilities as a stand-alone GE (i.e. without any connection with the other GEs), plus some usage scenarios involving other GEs, i.e. analysis built on top of the *data market -Store GE (WStore) + Data Portal GE (CKAN)*. This way we will show “business” capabilities such as the need to have the rights required to access the data the user wants to analyze (been registered, having accepted the terms and conditions, having acquired the data, etc.).
- b. The demo will also show how *SpagoBI* can create enriched datasets from the output of the performed analysis and make them available through the Store, thus enriching the data market with “premium” data offerings.

##### *Phase II: M18*

- a. The demo will show the key data market capabilities provided by the integration of the Store GE, the RSS GE, the Marketplace GE and CKAN. Specifically, how this data market will allow

---

<sup>11</sup> <http://catalogue.fiware.org/enablers/complex-event-processing-cep-proactive-technology-online>

<sup>12</sup> A new Generic Enabler for data visualization. It is not yet integrated in the FIWARE catalogue as it is still under development.

<sup>13</sup> <http://catalogue.fiware.org/enablers/store-wstore>

<sup>14</sup> <http://catalogue.fiware.org/enablers/marketplace-wmarket>

data providers to control access to their data, how basic management of offerings and sales (accounting, revenue sharing, etc.) are performed. In other words, the demo will demonstrate the process of selling these data, possibly packaged with customized visualization dashboards and analysis tools that confer the offering the character of “premium”, to consumers as well as to application developers.

- c. The demo will show how the *Data Viz GE* (SpagoBI) has been enriched to be able to perform analyses on the *Big Data Analysis GE* (COSMOS) and the *Context Broker GE* (ORION).
- d. The demo will show the integration of SpagoBI, WStore and WMarket for performing data analytics on existing data on sales, data API usage, etc. provided by the store / data market. The demo will also show the way the existing data and the results from the analysis performed on them can be visualized by sellers for decision taking.

#### Phase III: M25

- a. The demo will be progressively enriched with new features (basic functionalities shown in previous stages are to be enriched, etc.)

## 4.4 Robotics demo: From interactive to autonomous robot management in FIWARE

### 4.4.1 Demo idea

This demo is meant to demonstrate how robots can be operated by FIWARE GEs, and particularly the Robotics GE.

The demo will build incrementally from an interactive robot control scenario up to robots autonomously managing a number of tasks (always under the supervision of a FIWARE application). This will be done in conjunction with several other FIWARE GEs managing IoT, video, security, etc., by providing a very easy and visual understanding of the different components being involved in the system.

To make the setup appealing from a demonstration perspective, and to make it suitable for hands-on workshops, the use case will be centered on one/many robots, and multiple sensors, instructed to extinguish a fire in a room.

The Robotics GE exploits the ROS framework (Robotics Operating System): there is a wide community of developers supporting ROS, thus ensuring that a broad range of devices can be managed by the GE.

As said, the demo will build up incrementally, through a number of steps of increasing complexity. This is particularly important for both being able to provide material worth for hands-on workshops, as well as developing demonstrations of the Robotics GE gradually, which could be helpful to

promote the use of the GE even before the complete demo is created. This is valuable in order to promote early adoption of the GE by Accelerator Projects developers, as soon as basic functionalities are released by the design team.

The incremental design of the demo will rely on a number of steps, which are shortly described below, and then detailed in the demo scenario description:

- The first step implies the creation of a system to manage interactively one or more robots by means of the *Robotics GE*. This consists of building a platform that supports the concept of “Teleoperation as a Service” that exploits FIWARE GEs.
- The second step has to do with the capability of one or more robots to autonomously navigate to reach a given location, through e.g. areas with obstacles to be avoided.
- The third step includes the capability to instruct the robots to achieve a specified goal by navigating through areas with obstacles (from step two) and doing programmed actions, i.e. extinguishing a fire.

Throughout the demo steps, an increasing number of other FIWARE GEs will be used, to show how the integration between them and the Robotics GE is achieved, and to show the use of increasingly more complex features of this latter GE. It is assumed to implement further demo steps in future updates of the deliverable, covering a time period up to M25. The update will further highlight the enabler functionalities and how they can be adopted in different service scenarios.

#### 4.4.2 Demo contacts

Demo lead: **Exte-Tar**

Contributing partners: TI, Consoft, Naevatec, UPM

More detailed contact information can be found in *Annex A*.

#### 4.4.3 Demo scenario

In this section, the incremental steps of the demo are explained.

##### **4.4.3.1 Step1: Managing interactively robots**

In this first scenario set up, it is explained how to create an environment for remote control and teleoperation capabilities of a set of robots, equipped with (at least) a camera and with wireless connectivity (either WiFi or mobile e.g. LTE). The user will be able to manage the robots via a FIWARE web application (implemented e.g. by exploiting the Wirecloud GE).

Once the user has connected to the FIWARE app, he will get a page with the list of the available robots. After selecting one of them, the robot will disappear from the list in the first page and the user will get a second page with information about the robot status, the translational controls and a video stream of the camera boarded in the robot. The user will be able to explore the demo environment by employing the translation control and the video stream, managed through the Kurento Stream Oriented GE.

A sketch of those two screens is shown in the picture below.

Once the user stops the exploration activity, the robot will stop as well, and it can be found again in the list of available robots of the first page, ready to be controlled by a new user.

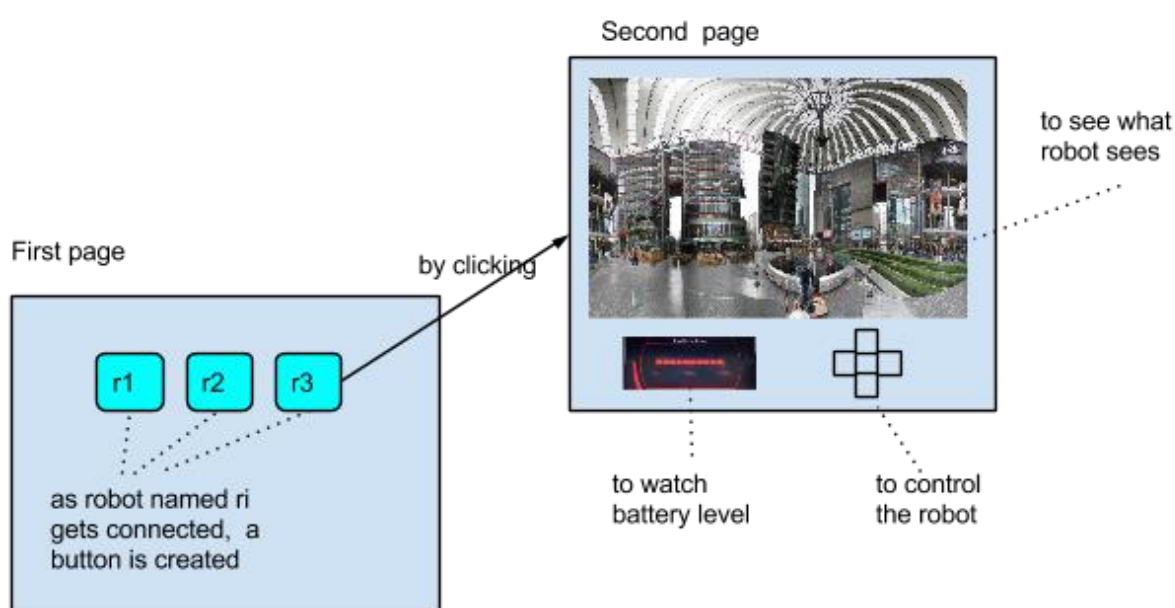


Figure4.1. Example screenshots for Step 1 of robotics demo.

#### 4.4.3.2 Step 2: Navigation to a target Location

In this step one or more robots will autonomously navigate to reach a given location, through e.g. areas with obstacles to be avoided.

One robot is instructed to navigate to a given location. A set of algorithms running on the robot will help the robot not running into any obstacles and will take care of planning both the global and local path. The setting implies using specific functionalities made available by the Robotics GE, i.e. path planning.

During the navigation process, the robot will be updating its own location on the Orion Context Broker and a FIWARE based application will represent the live location on an interactive 2D map.



The path planning algorithms need a map of the environment, where robot has to navigate, hence the very first step is “building a map” of the environment by using classical SLAM (Simultaneous Localization And Mapping) algorithm, e.g. gmapping ROS node. Once the map is available, this can be used by:

- path planning algorithms to perform localization (e.g. robot pose estimation) and plan both global (the minimum shortest path to reach the point) and local (tracking the global trajectory and avoiding “non-mapped” obstacles as needed)
- UI to visualize map and choose goals.

#### **4.4.3.3 Step 3: Achieve a specified goal – Firefighting Robot**

In the third demo step, robots will be instructed to achieve a specified goal by e.g. navigating through areas with obstacles and by doing programmed actions, in this case extinguishing a fire.

One robot capable of navigating autonomously and extinguishing a fire will be still on its dock-in station while several fire detectors are in constant connection with a remote context broker on FIWARE Lab. A web application deployed on the FIWARE Mashup Wirecloud, allows users to assign the robot the “fire extinguishing” task by configuring some parameters of the given platform as well as seeing the live video from the robot via the Kurento Stream Oriented GE. Once the robot has been properly configured by the user, it will be able to react autonomously to the “FIRE” alarm raised by any sensor on the context broker via the Publish/Subscribe method. After extinguishing the fire, the robot gets a confirmation from the application and returns back to its docking station. Users will be able to check the firefighter operations from the Mashup application where they will also be able to manage some parameters such as robot configurations, allowed areas, linked sensors, etc.

The incremental functionality exemplified by this step is the capability to associate tasks to a goal: once the robot reaches the goal where the fire has to be extinguished, associated tasks can be performed by accessing specific ROS nodes, i.e. the fan mounted on robot activated by a ROS service.

#### **4.4.4 Demo design**

In accordance to the definition of the three steps demo, the complexity of the design is gradually increasing, involving more and more FIWARE GEs.

##### **4.4.4.1 Design of step 1: Teleoperation as a Service (TaaS)**

In this step the demo is composed of three main elements:

- **Remote video system**, will be a webRTC-based embedded component in a Chrome browser installed in each robotic platform. This will connect to *Kurento GE* to manage video streaming.

- **Robot(s)** running ROS, can be any kind of robot equipped with a computer, a screen, a wireless connection and a video camera. The computer should have a GUI and should be able to run the elements required by the Robotics GE and all the elements related with the video system. A connectivity module is to be mounted on the robot, to ensure the connectivity to the platform and the FIWARE App; connectivity can be either WiFi or mobile (e.g. LTE to provide low latency).
- **FIWARE app** that can be implemented by using the *Wirecloud GE*.

### **GEs used**

- *Robotics*<sup>15</sup>: This GE will provide interface with robots.
- [\*Orion Context Broker\*](#): This GE provides context information for robots connected, maps and other information relevant for the environment detection
- [\*Stream-oriented - Kurento\*](#)<sup>16</sup>: All images coming from the robot will be streamed by using this GE, to display to the user of the application what the robot sees.
- [\*Application Mashup - Wirecloud\*](#): The Mashup will contain the management interface for the robots. This interface will allow visualization of the app, of the video, and management of the backend for allowed users.

The setup includes the following operations:

- Deploy the elements required by the *Robotics GE* in each robot.
- Install the camera in each robot.
- Deploy the Chrome browser and the required video component in the computer of each robot.

Once the setup is complete, every time a robot connects to the platform a clone is created by the *Robotics GE*. The setup will project the robot clone to the *Orion Context Broker* (via its FIROS component) by creating a proper entity (named as the clone) and by updating its attributes. The FIWARE app will be notified by the Context Broker about the connected robot. A high level chronogram is depicted below.

---

<sup>15</sup> A new Generic Enabler for Robotics. It is not yet integrated in the FIWARE catalogue as it is still under development.

<sup>16</sup> <http://catalogue.fiware.org/enablers/stream-oriented-kurento>

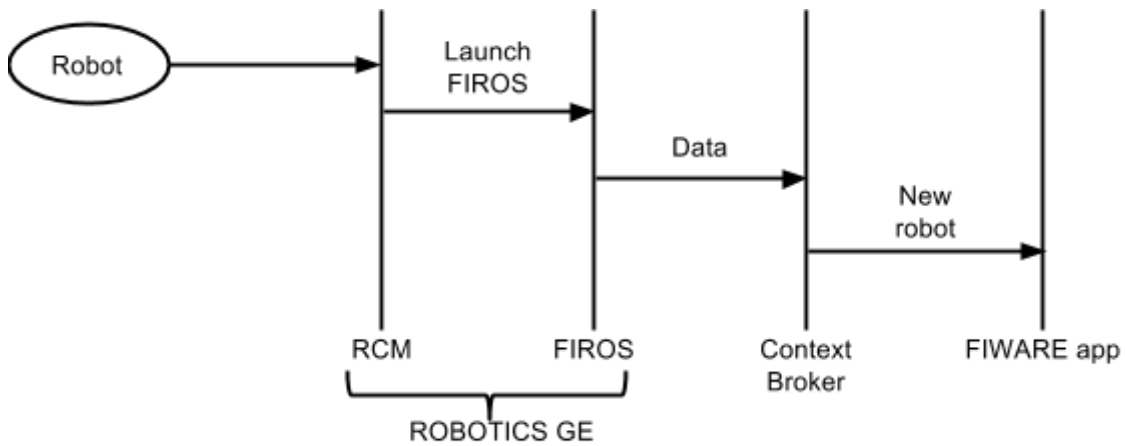


Figure4.2. Simplified transaction diagram for Step 1 of robotics demo.

Once a user has selected a robot, the FIWARE app will send commands to the robot and will receive its status (including its position) via the Context Broker communication (queries and subscriptions). The video from the robot’s camera will be sent to the FIWARE app via the *Kurento Stream Oriented GE*. A simplified data flow is depicted below.

- Information and commands
- Video streaming

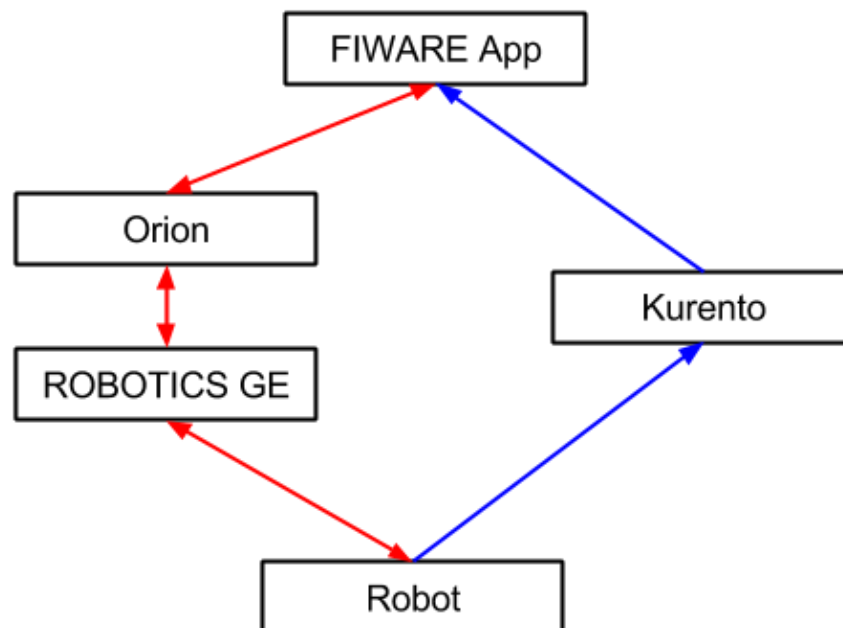


Figure4.3. Simplified data flow for Step 1 of robotics demo.

#### 4.4.4.2 Design of step 2: Navigation to a target location

These are the minimum requirements in order to run the demo step concerning the navigation to a target location:

- A robot capable of running ROS, i.e. with computing capabilities. It should be able to run the elements required by the Robotics GE and all the elements concerning with the video system.
- 30sqm to 200sqm room plenty of obstacles: The room should have a proper floor for the robot being used. It should also have several obstacles in order to help the robot mapping the terrain and making the demo more visually appealing. In order to help the robots visual perception systems, walls must not be transparent and no direct sunlight should affect the demo area.
- A controlled WiFi connection: In many places, WiFi connections are limited or require special logins. For those demo cases, it would be recommended to bring a own WiFi AP and build the whole system locally, including the Context Broker and the Web app.

#### GEs used

- *Robotics*: The Robotics GE provides connectivity between ROS and FIWARE components.
- [\*Application Mashup - Wirecloud\*](#) : The Mashup will contain the management interface for the robots. This interface will allow visualization of the app and management of the backend for allowed users.
- [\*Orion Context Broker\*](#) : This GE provides context information for the robots connected, maps and other information relevant for the environment detection. It also provides asynchronous events happening on the context broker in order to make the robot react to the events and configuration.
- [\*Identity Management - KeyRock\*](#)<sup>17</sup>: All robots and users must use the Identity manager in order to make the application safe. Each allowed user will be able to manage some settings on the management website and available robots must be also registered.
- [\*Stream-oriented - Kurento\*](#): All images coming from the robot will be streamed by using this GE. *Kurento* will also take care of some non-crucial image processing on the cloud. This will not include 3D video required to navigate or any other critical data.

A simplified transaction diagram is shown in the following picture.

---

<sup>17</sup> <http://catalogue.fiware.org/enablers/identity-management-keyrock>

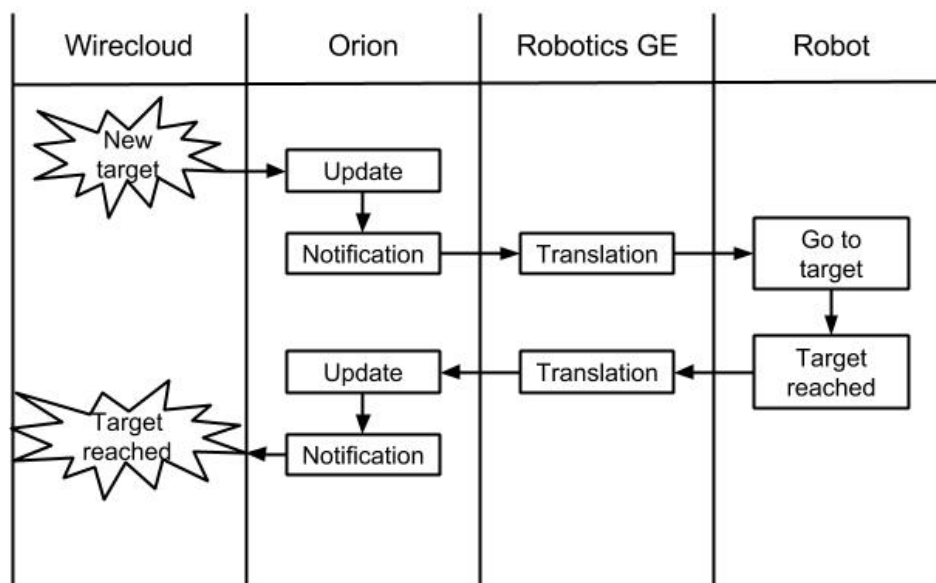


Figure4.4. Simplified transaction diagram for Step 2 of robotics demo.

#### 4.4.4.3 Design of step 3: Achieve a specified goal – Firefighting Robot

This third step of the robotics demo may consist of a technical hands-on workshop preceding the actual demo, where sensors and robots can be built and programmed, in order to make a comprehensive FIWARE activity centered on teaching the capabilities of the technology involved.

These are the minimum requirements in order to run this demo step:

- A robot capable of running *ROS* and putting out a fire:  
A simple [turtlebot](http://turtlebot.com/)<sup>18</sup> with a custom actuator made out of an Arduino and a fan will be enough. The arduino source code, as well as the ROS node for controlling it will be provided.
- At least one FIWARE compatible fire detector:  
If no fire detectors are available to connect to FIWARE, a custom sensor can be made as part of any workshop preceding the actual demo. These sensors must be able to connect to the Internet or local area network. Source code as well as electrical schemas will also be provided in order to be used as part of any workshop.
- 30m<sup>2</sup> to 200m<sup>2</sup> room plenty of obstacles:  
The room should have a proper floor for the robot being used. It should also have several obstacles in order to help the robot mapping the terrain and making the demo more visually appealing. In order to help the robots visual perception systems, walls must not be transparent and no direct sunlight should affect the demo area.
- A controlled WiFi connection:  
In many places, WiFi connections are limited or require special logins. For those demo cases, it would be recommended to bring a own WiFi AP and build the whole system locally, including the *Context Broker* and the *Web app*.

<sup>18</sup> <http://turtlebot.com/>

High level transaction diagrams can be found in the following two figures.

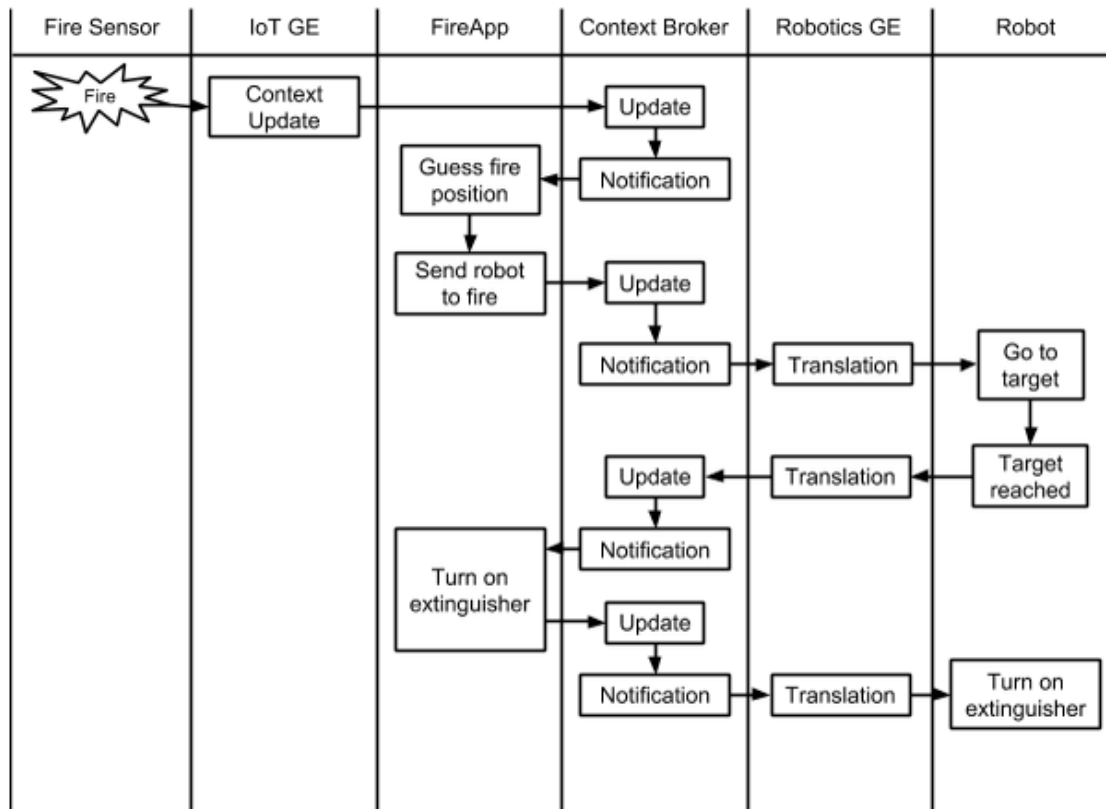


Figure4.5. Transaction diagram for Step 3 of robotics demo: Activating robot and performing target action.

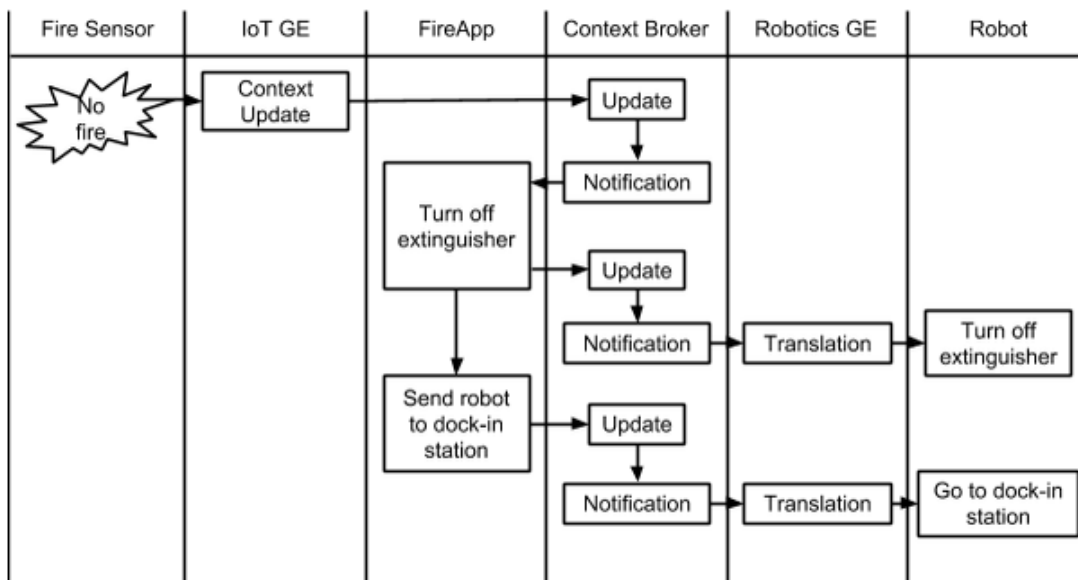


Figure4.6. Transaction diagram for Step 3 of robotics demo: Return to idle state after target action performed.

### GEs used

Same Generic Enablers as used for Step 2.

An overall high level block diagram of the Robotics live demo main components is shown in the following picture.

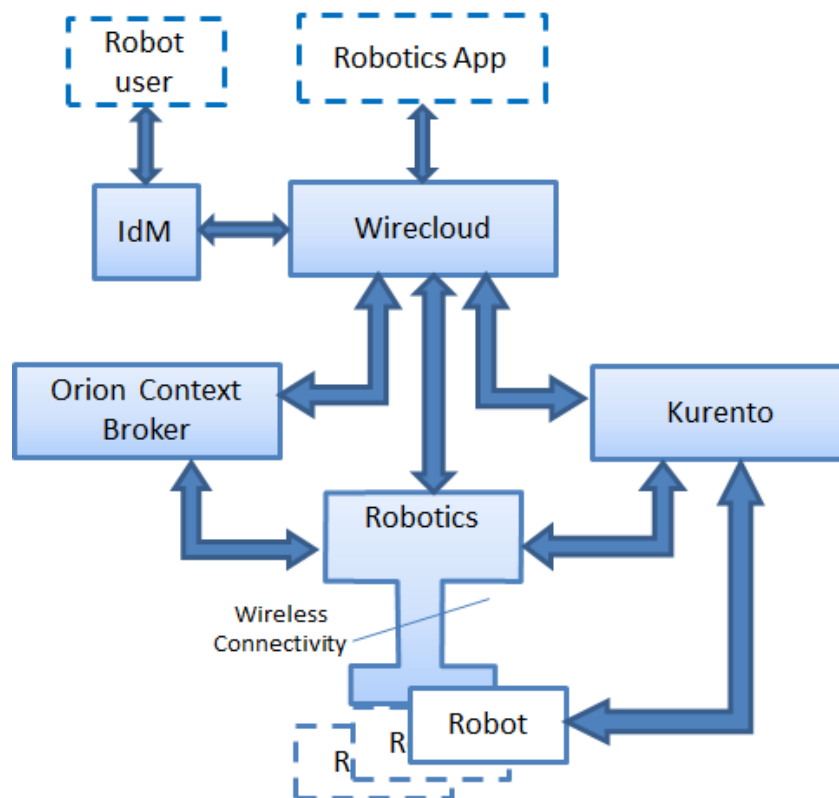


Figure4.7. High Level architecture of Robotics live demo..

### 4.4.5 Time Plan

Public releases of Robotics live demo will be available according to the plan centered around the three steps defined so far.

- *Phase I:* First of all, Step 1 described in Section 4.4.3.1 will be released at *M13*. This will go along with the development of the first publicly available release of Robotics GE.
- *Phase II:* the demo scenario described as Step 2 in Section 4.4.3.2 will be released at *M15*.
- *Phase III:* the demo scenario described as Step 3 in Section 4.4.3.3 will be released at *M18*.

An update of the robotics live demo description will be done after *M13*, to align with the first official release of the GE. It is assumed that further functionalities will be added, along with an update of the demo time plan up to *M25*. The update will possibly consist of the introduction of additional steps in the robotics live demo, to further highlight the enabler functionalities and how they can be adopted in different service scenarios.

## 4.5 Manufacturing Demo

### 4.5.1 Demo idea

The purpose of this demo in the Manufacturing domain is to enhance the interoperability between machines and robots in a factory shop floor. Additionally it pursues monitoring the global state of a factory enabling new types of services related to the data analysis, simulations and visualization techniques for machines.

Based on different machine data, such as power and temperature collected in real-time, the demonstrator will be mainly focused on energy issues in several areas. These areas are: energy usage monitoring and energy performance indicators visualisation in order to be able to act motivated by energy aware behaviours. Energy measurements are usually related to the health of the machine and, therefore, the final quality of the product being manufactured, that is why the information provided by these indicators will help creating the right context to better understand and predict faults on the machines and defective parts. For instance, when the machine works with a low power, indicates that the laser is off or not working properly, so the part will be defective. Additionally, when this situation happens, robots must be notified in order to remove the defective part.

In this aspect, this demo is meant to demonstrate the communication between machines and robots, monitor the machines and the interaction between machines, robots and several FIWARE components (such as IoT, video, security, etc) in an industrial environment by providing a very easy to understand demo.

Therefore, we will monitor a cell where there are a machine and a robot, the machine will communicate with the robot different status and if it is necessary to pick faulty parts.

### 4.5.2 Demo contacts

Leading partner: **Atos**

Contributors: Etxe-Tar

More detailed contact information can be found in *Annex A*.

### 4.5.3 Demo scenario

These are the minimum requirements in order to run this manufacturing demo:

- A machine able to provide power data (consumption) and temperature.
- A robot capable of running ROS and transporting parts.
- A cell big enough where to deploy the manufacturing machines, dock-in area, FIWARE enabled doors and delivery point. A manufacturing plant would also be suitable.
- A controlled wireless connection.



#### 4.5.4 Demo design

In the context of the shop-floor, sensorized machines part of a manufacturing line are able to generate and upload energy and temperature related data obtained from their internal systems through various embedded sensors connected to these machines. Such data will be uploaded to a dedicated Context Broker which will fire events related to the quality assurance of the manufactured parts.

A robot capable of navigating autonomously in a manufacturing area and able to carry parts that will be placed in its dock-in station will be involved. In the same area there will be a set of machining-tools manufacturing parts. Some of these parts may be discarded as faulty or as test samples due to problems with the power of the machine, for example when the laser is not working properly, energy consumption decreases significantly, also problems with the temperature levels, etc.; every time a part is discarded, the Context Broker is notified through an event generated by the DataHandling GE, which supervises that the machine is working within the established thresholds (power and temperature). These indicators are collected from machines in real-time.

In parallel, events with power and temperature are also sent to specific widgets deployed on Wirecloud, in order to provide a real time decision support strategy. Risks for unnecessary energy use or a malfunction of the machine due to energy are identified by CEP algorithms. This data shown graphically will be visualized by operators in several widgets, particularly there will be one devoted to alarm visualization.

The following sequence diagram illustrates a high-level operation focused on part of the demo responsible for monitoring the status of the machine and displaying this data, as well as triggering the application of Robots responsible for initiating the necessary process to discard the defective part (detailed in subsequent process diagrams)

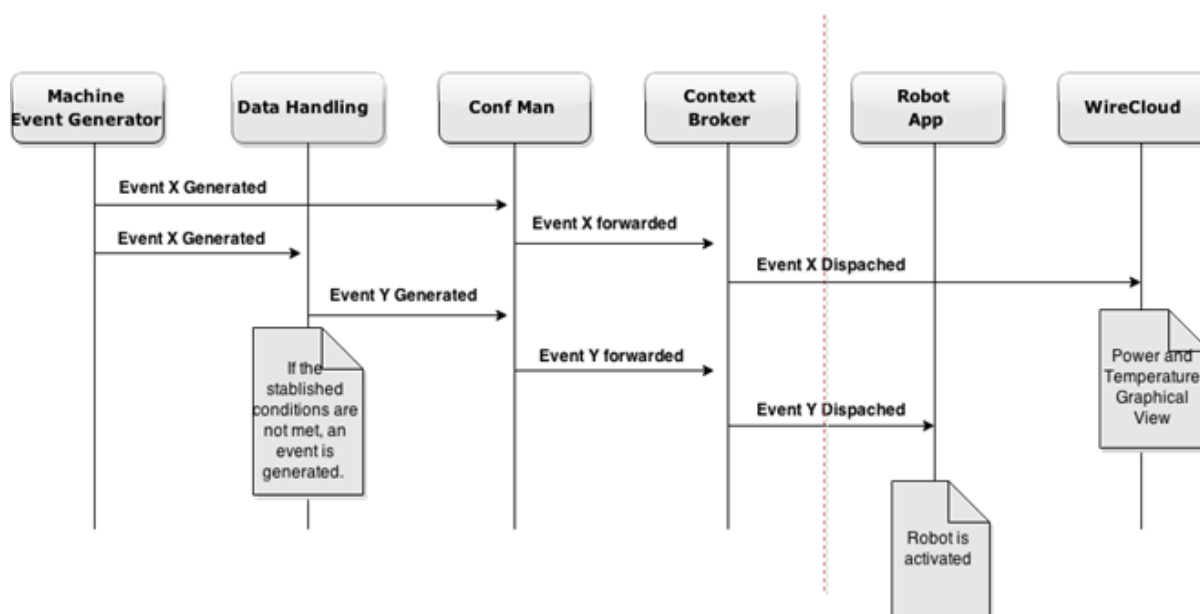


Figure 5.1. Operational diagram for monitoring and visualization in the Manufacturing demo

After the *Context Broker* has received the notification, an application will dispatch the robot to collect the part. Once the robot has the part, the application will be notified and will send the robot to the delivery point.

The delivery area will be protected by FIWARE enabled doors, which are mechanical doors with the required electronics to be used as a remote actuator so they can be opened by sending a command through the FIWARE platform. The application will operate the doors based on the position and heading of the robot.

Once the robot has reached the delivery point, the part will be placed on the “quality check station”. Afterwards the application is notified and the robot is sent back to its dock-in station, crossing the above mentioned doors.

All the process could be monitored and commanded via a FIWARE-based web application.

The critical points (both sides of the robots, transferences points and dock-in area) could be monitored via cameras connected to *Kurento GE*. This image could be processed in order to help the operators focusing their attention where it is required.

An approximated process diagram is depicted below.

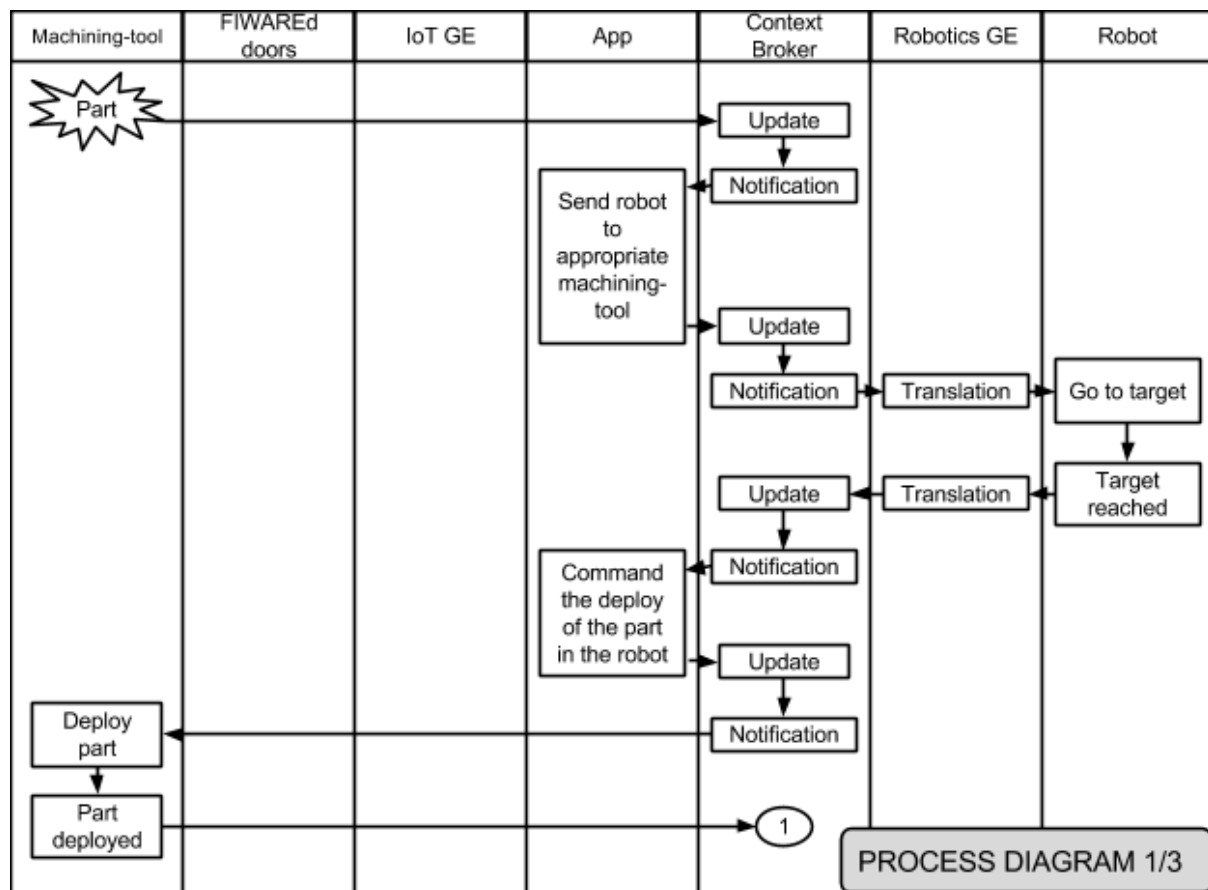


Figure5.2. Process diagram for Manufacturing demo (1 of 3)

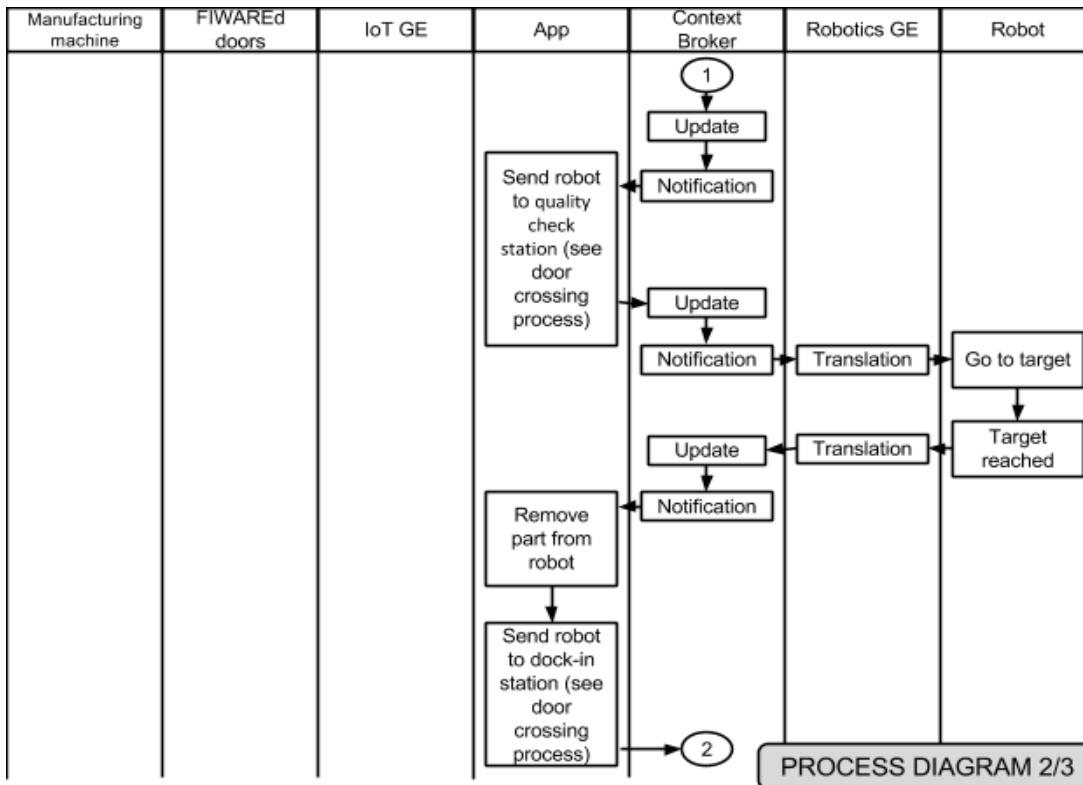


Figure5.3.

Process diagram for Manufacturing demo (2/3 of 3)

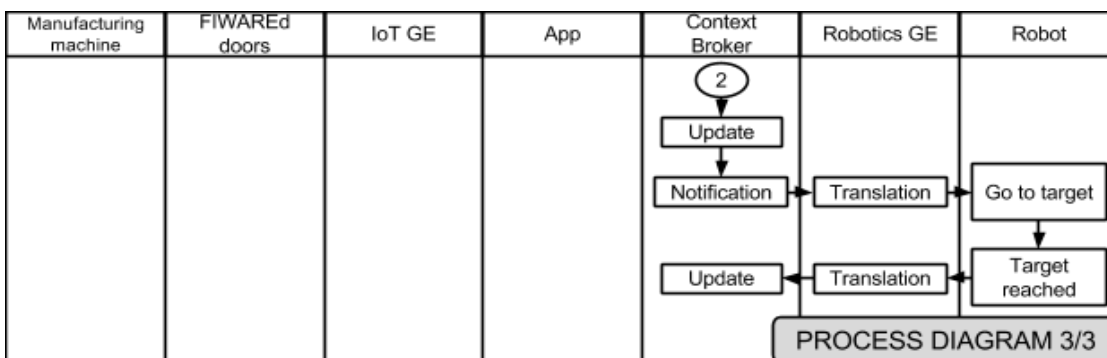


Figure5.4. Process diagram for Manufacturing demo (3/3 of 3)

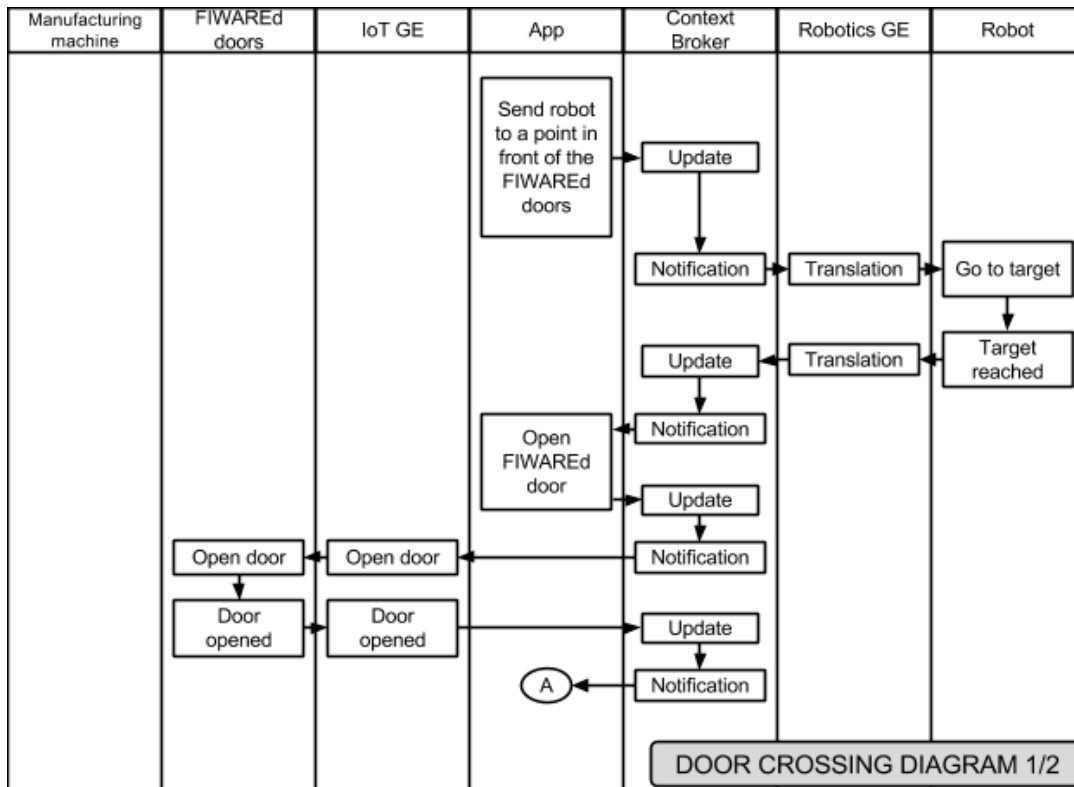


Figure5.5. Door Crossing Process diagram for Manufacturing demo (1/2 of 2)

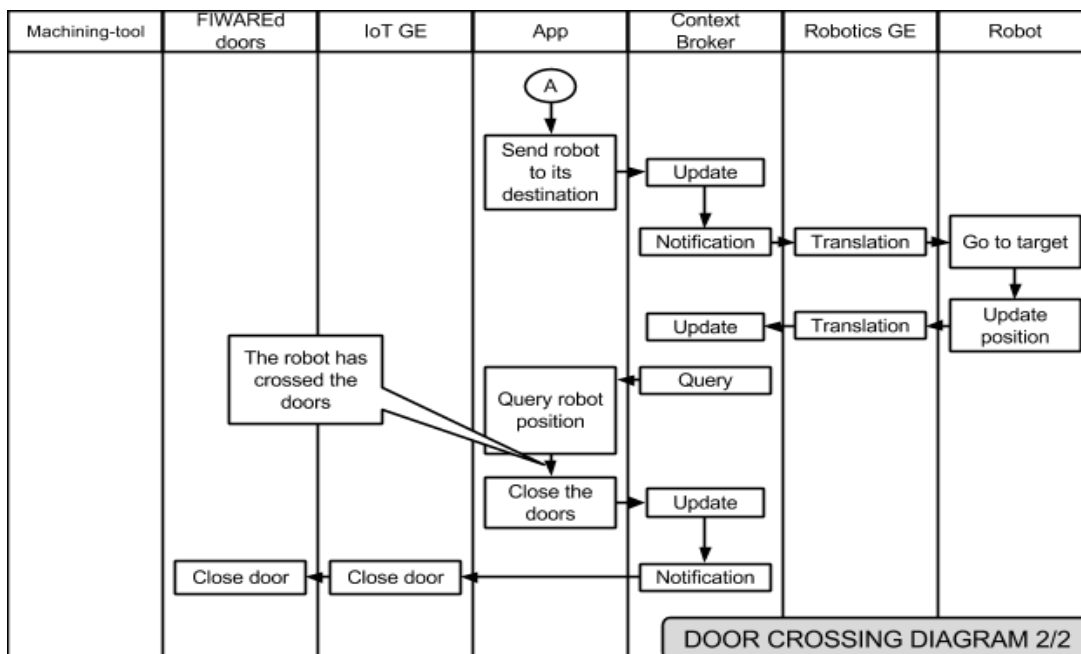


Figure5.6. Door Crossing Process diagram for Manufacturing demo (2/2 of 2)

#### 4.5.5 GEs used

- **Robotics:** The *Robotics GE* provides connectivity between ROS and FIWARE components.
- **Application Mashup - Wirecloud:** A Mashup will contain the management interface for the robots and machines. This interface will allow visualization of the app and management of the backend for allowed users.
- **Configuration Manager - IoT Discovery<sup>19</sup>:** This GE provides the discovery process for the FIWARE enabled sensors and actuators.
- **Gateway Data Handling GE - EspR4FastData<sup>20</sup>:** Executes the custom event processing logic, evaluates conditions based on event payload, and generates new events when conditions are met, in this case conditions related to Power and Temperature. This is the one of the core GEs in the demonstrator.
- **Orion Context Broker:** This GE provides context information for the environment, manufacturing machines, robots connected, maps and other information relevant for the surroundings detection.
- **Stream-oriented - Kurento:** All images coming from the cameras installed in the aforementioned critical points will be streamed by using this GE. *Kurento* will also take care of some non-crucial image processing on the cloud. This will not include 3D video required to navigate or any other critical data.

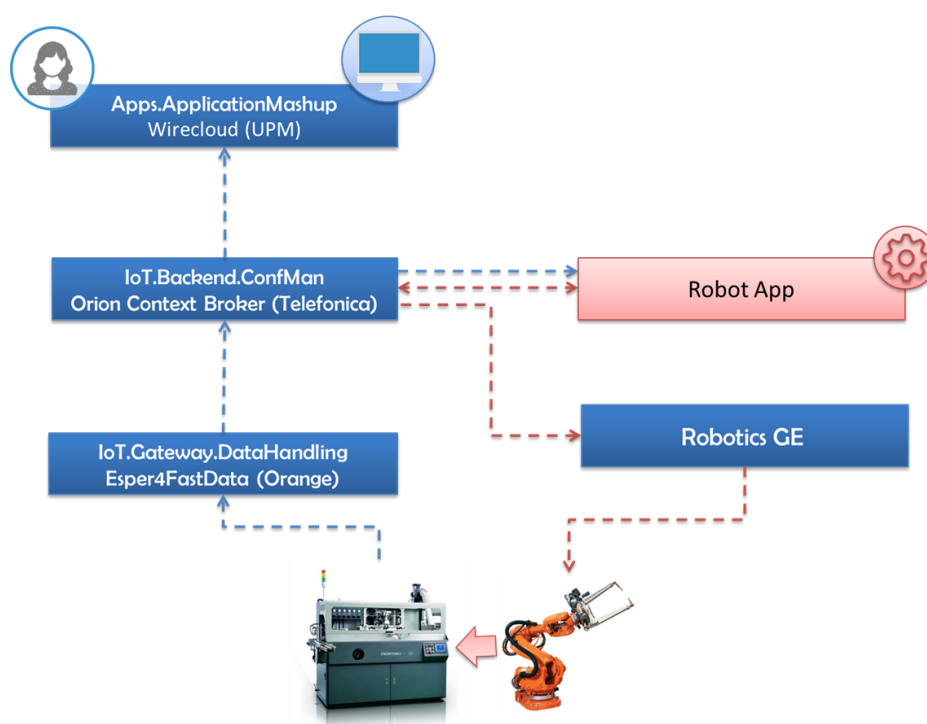


Figure5.7. High level architecture for Manufacturing demo.

<sup>19</sup> <http://catalogue.fiware.org/enablers/configuration-manager-iot-discovery>

<sup>20</sup> <http://catalogue.fiware.org/enablers/gateway-data-handling-ge-espr4fastdata>

#### 4.5.6 Time Plan

This live demo includes the use of actual manufacturing machinery that will have to be manipulated in order to allow it to connect to FIWARE components. Therefore, it will have to be divided into three phases:

- *Phase 1 (M13)*: Demo based on simulated data. This data will be provided by Etxe-Tar and extracted from actual machines that are either being built or already working on a manufacturing plant.
- *Phase 2 (M18)*: Machine connectivity. Some hardware will have to be developed and attached to the machine in order to allow it to upload data to FIWARE components. After this stage is completed, actual data coming from the machine will be available at manufacturing time.
- *Phase 3 (M25)*: In *M18* and *M25* releases, we will explore additional functionalities to extend the demo, possibly related with operational monitoring, predictive maintenance or decision support.

## 5 Delivery mechanism

The following artefacts will be provided for each demo:

- Running version on a FIWARE infrastructure - Actual demo development may be done on different FIWARE environments, but the goal is to make an effort to deploy at least the final versions for the public releases on the FIWARE Lab.
- Open data used in the demo on FIWARE Lab (either as real data or simulated data, depending on the demo)
- A video on Youtube
- Demo source code on Github
- Documentation on Github and FIWARE wiki along with the code, on how to run, how to install, the architecture, the GEs used, open data used, etc.
- A blog post for each demo published on the FIWARE Developers Website, intended to disseminate the lessons learnt while building the prototypes
- A tutorial on FIWARE Academy on how to re-create the same demo, for self-learning
- Simplified lab exercises for use in training events

## 6 Information on other demos in the project

Information on some of the other demos previously developed in the project can be found at: <http://www.fiware.org/fi-ware-developers-portal>.

## Annex A. Detailed Contact Information

Detailed contact information for the demos can be found in the following tables. Leading partner for each demo is marked in **bold**.

Partner	Name	Details
<b>ATOS</b>	ilknur Chulani	ilknur.chulani@atos.net
	Omer Faruk Ozdemir	omer.ozdemir@atos.net
TID	Santiago Martinez Garcia	santiago.martinezgarcia@telefonica.com
UPM	Javier Soriano	jsoriano@fi.upm.es
Naevatec	Javier Lopez Fernandez	fjlopez@naevatec.com
Adminotech	Toni Alatalo	toni@playsign.net
Cyber	Sami Jylkkä	sami.jylkka@cyberlightning.com
DFKI	Philipp Slusallek	philipp.slusallek@dfki.de

*Table A.1 Contact information for Smart city demo*

Partner	Name	Details
<b>TID</b>	Santiago Martinez Garcia	santiago.martinezgarcia@telefonica.com
	Jose Rodriguez	jose.rodriguezperez@telefonica.com
	Sergio García	sergio.garciagomez@telefonica.com
Cyber	Sami Jylkkä	sami.jylkka@cyberlightning.com
OKFN	Tryggvi Björgvinsson	tryggvi.bjorgvinsson@okfn.org

*Table A.2 Contact information for Transport demo*

Partner	Name	Details
<b>Engineering</b>	Stefano De Panfilis	stefano.depanfilis@eng.it
	Monica Franceschini	monica.franceschini@eng.it
	Davide Zerbetto	davide.zerbetto@eng.it
UPM	Javier Soriano	jsoriano@fi.upm.es

*Table A.3 Contact information for Business (Smart Energy) demo*

Partner	Name	Details
Etxe-Tar	Angel Hernandez	ahernandez@ikergune.com
TI	Pierangelo Garino	pierangelo.garino@telecomitalia.it
Consoft	Fabio Di Benedetto	fabio.dibenedetto@consoft.it
UPM	Javier Soriano	jsoriano@fi.upm.es
Naevatec	Javier Lopez Fernandez	fjlopez@naevatec.com

Table A.4 Contact information for Robotics demo

Partner	Name	Details
ATOS	Jesus Benedicto	jesus.benedicto@atos.net
	Silvia Castellví	silvia.castellvi@atos.net
Etxe-Tar	Angel Hernandez	ahernandez@ikergune.com
	Jose Jaime Ariza	jjjaime@ikergune.com

Table A.5 Contact information for Manufacturing demo

## Annex B. Context Elements for Transport Demo

In order to cover the Transport demo scenarios described in Section 4.2, several context elements relevant for each of the Tecnoport systems (FPS, eRIO, CUTS) are being defined.

The following context elements have already been identified as necessary for the demo scenarios described:

- FPS

contextElement TRAIN			
id	xxx		
type	train		
attributes	train registration number	name	registration number
		type	string



		value	xxxxx			
		metadata	Timestamp of the update	name	timestamp	
				type	timestamp	
				value	ISO8601	
	Train Owner	name	owner			
		type	string			
		value	xxxxxx			
		metadata	Timestamp of the update	name	timestamp	
	type			timestamp		
	value			ISO8601		
	GPS position	name	position			
		type	coords			
		value	xx.xxxxxx, yy.yyyyyy			
		metadata	Timestamp of the update	name	timestamp	
	type			timestamp		
value	ISO8601					
Train longitude	name	longitude				
	type	float				
	value	xx.xx				

		metadata	Timestamp of the update	name	timestamp
				type	timestamp
				value	ISO8601
	Train Origin	name	origin		
		type	string		
		value	xxxxxxx		
		metadata	Timestamp of the update	name	timestamp
				type	timestamp
				value	ISO8601
	Train Destination	name	destination		
		type	string		
		value	xxxxxxx		
		metadata	Timestamp of the update	name	timestamp
				type	timestamp
				value	ISO8601
	Expected Arrival time in ADIF-port frontier	name	Expected Arrival Time (ADIF-port)		
	type	timestamp			
	value	ISO8601			
	metadata	Timestamp	name	timestamp	

			of the update	type	timestamp	
				value	ISO8601	
	Expected Arrival Time (port)	name	Expected Arrival Time (port)			
		type	timestamp			
		value	ISO8601			
		metadata	Timestamp of the update	name	timestamp	
	type			timestamp		
	value			ISO8601		
	Entrance Route	name	Entrance Route			
		type	integer			
		value	xx			
		metadata	Timestamp of the update	name	timestamp	
	type			timestamp		
	value			ISO8601		
	Timeslot assigned in Quay	name	Timeslot assigned in Quay			
		type	timestamp			
value		ISO8601				
metadata		Timestamp of the update	name	timestamp		
	type		timestamp			

				value	ISO8601	
	Departure Time from Quay	name	Departure Time from Quay			
		type	timestamp			
		value	ISO8601			
		metadata	Timestamp of the update	name	timestamp	
				type	timestamp	
				value	ISO8601	
		Departure Route	name	Departure Route		
type	integer					
value	xx					
metadata	Timestamp of the update		name	timestamp		
			type	timestamp		
			value	ISO8601		
Departure Time (ADIF-port frontier)	name		Departure time (ADIF-port frontier)			
	type	timestamp				
	value	ISO8601				
	metadata	Timestamp of the update	name	timestamp		
			type	timestamp		
			value	ISO8601		

	Priority	name	priority	
		type	integer	
		value	xx	
		metadata	Timestamp of the update	name
	type	timestamp		
	value	ISO8601		
	Load Type	name	load	
		type	string	
		value	xxxxxxxxxxxx	
		metadata		name
	type	timestamp		
	value	ISO8601		

contextElement Railway resource			
id	rfX		
type	railway resource		
attributes	Current route	name	Current route
		type	integer
		value	x

		metadata	Timestamp of the update	name	timestamp
				type	timestamp
				value	ISO8601

contextElement Railroad Crossing						
Id	pnX					
type	Railroad Crossing					
attributes	GPS Position	name	position			
		type	coords			
		value	xx.xxxxxx, yy.yyyyyy			
		metadata	Timestamp of the update	name	timestamp	
				type	timestamp	
				value	ISO8601	
		km point of the railway where the railroad crossing is located	name	km point		
			type	integer		
			value	xx		
	metadata		Timestamp of the update	name	timestamp	
				type	timestamp	
		value		ISO8601		

	Train railway	name	Train railway	
		type	integer	
		value	xx	
		metadata	Timestamp of the update	name
		type		timestamp
		value		ISO8601
	Occupancy or not in the nearest side to Km0	name	statusA	
		type	bool	
		value	0->free, 1->occupied	
		metadata	Timestamp of the update	name
		type		timestamp
		value		ISO8601
	Occupancy or not in the most far away side to Km0	name	statusB	
type		bool		
value		0->free, 1->occupied		
metadata		Timestamp of the update	name	timestamp
	type		timestamp	
	value		ISO8601	
Direction of the	name	direction		

	train route	type	string		
		value	{arrival, departure}		
		metadata	Timestamp of the update	name	timestamp
				type	timestamp
				value	ISO8601

contextElement Switch Junction					
id	caX				
type	switch junction				
attributes	name	position			
	type	coords			
	value	xx.xxxxxx, yy.yyyyyy			
	metadata	Timestamp of the update	name	timestamp	
			type	timestamp	
			value	ISO8601	
	name	km point			
	type	integer			
	value	xx			



	metadata	Timestamp of the update	name	timestamp
			type	timestamp
			value	ISO8601
	name	via		
	type	integer		
	value	xx		
	metadata	Timestamp of the update	name	timestamp
			type	timestamp
			value	ISO8601
	name	status		
	type	bool		
	value	0->free, 1->occupied		
	metadata	Timestamp of the update	name	timestamp
			type	timestamp
			value	ISO8601

contextElement Length Measurement System	
id	sIX
type	Length system

attributes	GPS position	name		position	
		type		coords	
		value		xx.xxxxxx, yy.yyyyyy	
		metadata	Timestamp of the update	name	timestamp
	type			timestamp	
	value			ISO8601	
	km point in the railway where length measurement system is located	name		km point	
		type		integer	
		value		xx	
		metadata	Timestamp of the update	name	timestamp
	type			timestamp	
	value			ISO8601	
	Length of the train measured at the port entrance	name		length	
		type		float	
		value		xx.xx	
metadata		Timestamp of the update	name	timestamp	
			type	timestamp	
	value		ISO8601		
	Estimated error	name	Estimated error		

				type	float
				value	xx
	Train speed at port entrance	name	speed		
		type	integer		
		value	xx		
		metadata	Time stamp of the update	name	timestamp
		type		timestamp	
		value		ISO8601	
	Direction of the train	name	direction		
		type	string		
		value	{arrival, departure}		
		metadata	Timestamp of the update	name	timestamp
	type	timestamp			
	value	ISO8601			

contextElement Railway resource	
id	rfX
type	railway resource

attributes	Current route	name		Current route	
		type		integer	
		value		x	
		metadata	Timestamp of the update	name	timestamp
type	timestamp				
value	ISO8601				

- eRIO

contextElement Boat					
id		xxx			
type		boat			
attributes	IMO	name		IMO	
		type		string	
		value		xxxxxxxx	
		metadata	Timestamp of update	name	timestamp
	type			timestamp	
	value			ISO8601	
	MMSI	name		MMSI	
		type		string	

		value	xxxxxxxx			
		metadata	Timestamp of update	name	timestamp	
				type	timestamp	
				value	ISO8601	
	Owner of the boat	name	owner			
		type	string			
		value	xxxxxx			
		metadata	Timestamp of update	name	timestamp	
	type			timestamp		
	value			ISO8601		
	GPS position of the boat	name	position			
		type	coords			
		value	xx.xxxxxx, yy.yyyyyy			
		metadata	Timestamp of update	name	timestamp	
	type			timestamp		
value	ISO8601					
Minimum speed of the boat	name	Minimum speed				
	type	float				
	value	xx.xx				

		metadata	Timestamp of the update	name	timestamp		
				type	timestamp		
				value	ISO8601		
	Maximum speed of the boat	name	Maximum speed				
			type	float			
				value	xx.xx		
	metadata	Timestamp of the update	name	timestamp			
			type	timestamp			
			value	ISO8601			
	Boat dimensions	name	dimensions				
			type	string			
				value	[xx, yy, ..]		
	metadata	Timestamp of the update	name	timestamp			
type			timestamp				
value			ISO8601				

- CUTS

contextElement User Terminal				
id	xxx			
type	user terminal			
	Hub ID	name	Hub ID	
		type	string	
		value	xxxxxxxx	
		metadata	Timestamp of the update	name
		type		timestamp
		value		ISO8601
	GPS position of the hub	name	position	
		type	coords	
		value	xx.xxxxxx, yy.yyyyyy	
		metadata	Timestamp of the update	name
		type		timestamp
		value		ISO8601
	Sensor temperature measurement	name	temperature	
		type	float	
		value	xx.xx	

		metadata	Timestamp of the update	name	timestamp
				type	timestamp
				value	ISO8601
			Minimum allowed temperature	name	minimum
				type	float
				value	xx.xx
			Maximum allowed temperature	name	maximum
				type	float
				value	xx.xx
	Humidity measurement sensor	Name	humidity		
		Type	float		
		Value	xx.xx		
		metadata	Timestamp of the update	name	timestamp
				type	timestamp
value				ISO8601	
Minimum allowed humidity		name	minimum		
		type	float		
		value	xx.xx		
Maximum	name	maximum			



			allowed humidity	type	float	
			value	xx.xx		
	luminosity measurement sensor	Name	luminosity			
		Type	float			
		value	xx.xx			
		metadata	Timestamp of the update	name	timestamp	
				type	timestamp	
				value	ISO8601	