Instant Mobility

Multimodality for people and goods in urban areas

FP7 . CP 284906

WP5.2 – Domain specific enablers

March 2013

Editor(s): Jean-Marie Dautelle / Thales
Marco BOTTERO / Swarco Mizar S.p.A.
Samson TSEGAY / Swarco Mizar S.p.A.

Project funded by the European Commission under the 7th European Framework Programme for RTD - ICT theme of the Cooperation Programme.

Instant Mobility WP5
Instant Mobility prototype

WP5	Deliverable title
Authors	Jean-Marie DAUTELLE - THALES Marco BOTTERO – Swarco Mizar S.p.A. Samson TSEGAY – Swarco Mizar S.p.A. Marcus LARSSON - VOLVO Michael BOC - CEA Gérard Scemama - IFSTTAR
Short Description	Based on requirements and feedback from the Core Platform project, this document reports on the development of the relevant domain specific enablers to fulfil Instant Mobility prototype functionalities. This deliverable targets also specifically the conditions for reuse of the components in later phases.
Dissemination level (select)	PP - Restricted to other program participants (including Commission Services)
Date	28th of March 2013
Status	Deliverable
Contributions by:	Jakob Fryk – VOLVO Dirk BECKMANN – DLR Michael BOC – CEA Andrea Bragagnini - TLI
Internal review by	Patrick Constant – PERTIMM Merja Pettinen – VTT
Internally accepted by	WP5 leader
Date of acceptance	25.04.2013

Document history

Version	Date	Author / Reviewer	Description
0.1	03.12.2012	M. Bottero, S. Tsegay	TOC
0.2	01.02.2013	M. Bottero, S. Tsegay	Scenario 3 section added
0.3	08.03.2013	M. Larsson, J. Fryk	Scenario 2 section added
0.4	11.03.2013	S. Tsegay	Testing section redefined
0.5	12.03.2013	M. Bottero	New consolidated version
0.6	19.03.2013	J.M. Dautelle	Scenario 1 section added
0.7	27.03.2013	M. Larsson, J.M. Dautelle	Added section on reusability
0.8	28.03.2013	J.M. Dautelle	Added Chapter 7 (FI-Ware GE Validation)
0.9	07.04.2013	G. Scemama	Scenario 1 updated
0.10	25.04.2013	T. Nagellen	Scenario Smart City Logistics with FI-

			Ware GEs

Deliverable Abstract

This document reports on the development of the relevant domain specific enablers to fulfil Instant Mobility prototype functionalities.

This document is divided into 6 sections. The first section is the introduction, setting the context of the deliverable. Sections 2-3-4 are related to the three Conceptual Prototypes developed in each one of the three scenarios of Instant Mobility and describe their design, development, deployment and FI-WARE integration. Section 5 reports some conclusions coming from the work done in developing the three Instant Mobility prototypes. Finally, Section 6 holds the result of the FI-Ware GEs Validation and the lessons learned from the integration of the GEs to the relevant demonstrators.

Table of Content

Table of Figures

1. Introduction

The “Instant Mobility” project is developing and exploring a concept for transforming the mobility of persons and goods in the future through application of advanced Internet technologies. The project has created a concept for a virtual "Transport and Mobility Internet", a platform for information and services able to support radically new types of connected applications for scenarios centred on the stakeholder groups: multimodal travellers, drivers, passengers, transport operators, goods vehicle operators, road operators and traffic managers.

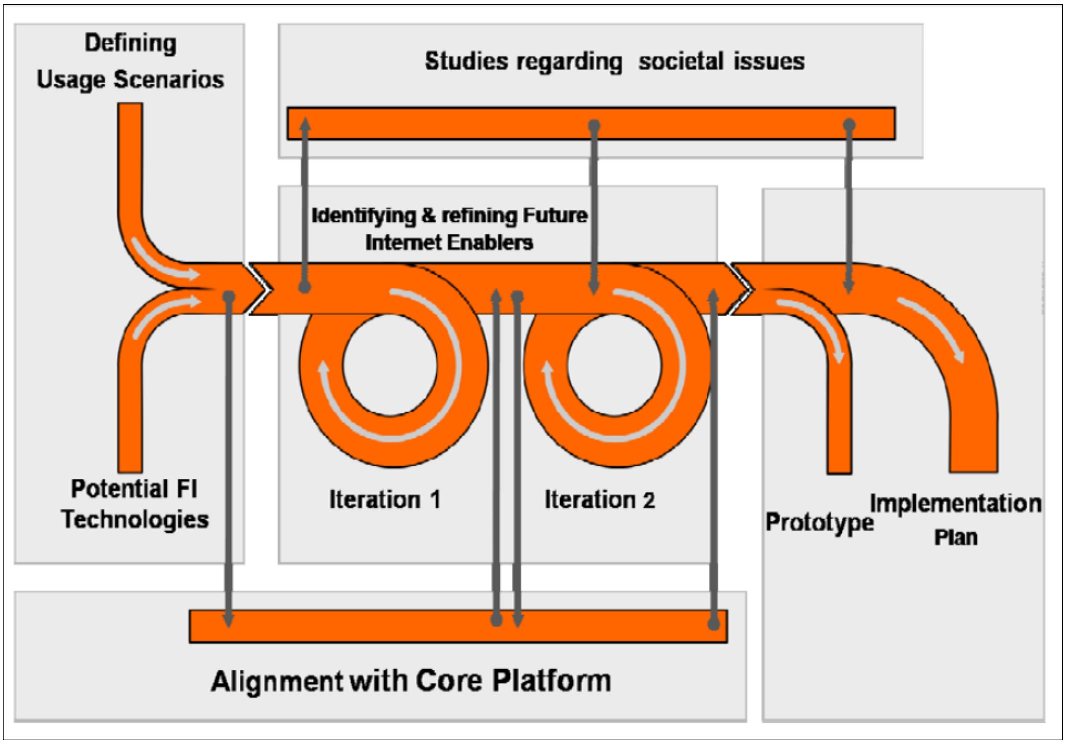


Figure 1. Project overall process & main results

Work Package 5 receives the consolidated full technical specifications that are derived in WP4 and implements a specific subset of the specifications to fulfil three main purposes:

- Define and implement a conceptual prototype showing the feasibility both of the underlying Future Internet technologies and the applicability of those technologies in the domain of transport and mobility;
- Implement domain specific enablers: These are generic and re-usable software components that will be used in the Instant Mobility prototype or serve as a starting point in Phase 2;
- Prepare for phase 2 a comprehensive implementation plan. This will serve as a conceptual link to the next phase and will especially consider the domains specific enabler development, the evaluation of concept feasibility study and the assessment and selection of potential test sites for a small and large trials in phase 2 and 3 respectively.

The prototypes presented in this document also facilitate key FI technologies to demonstrate their feasibility in this domain and to proof the Instant Mobility technological approach. The development outcome of this work package is twofold:

- A prototype that is used to verify the specified concepts and technologies within this work package and as well provide a demonstrator for WP7;

- This work package is rounding up the specifications of this project and is providing a bridge to the full realization of the Instant Mobility concepts in Phase 2 by supplying full specifications that have been (defined in WP4) and are then verified by the three conceptual prototypes.

Starting from these considerations, this document describes the realisation of the three prototypes defined in Task 5.1 with the developed domain specific enablers (task 5.2) and the components supplied by the FI-WARE core platform.

1.1 Abbreviations

A-GPS	Assisted GPS
API	Application Programming Interface
CP	Core Platform
DDS	Data Distribution Service
GE	Generic Enabler
GPS	Global Positioning System
GUI	Graphical User Interface
IaaS	Infrastructure as a Service
IoS	Internet of Services
IoT	Internet of Things
IM	Instant Mobility
PaaS	Platform as a Service
PT	Public Transport
REST	REpresentational State Transfer
RSU	Road Side Unit
SAML	Security Assertion Markup Language
SOAP	Simple Object Access Protocol
TLC	Traffic Light Controller
TMC	Traffic Management Centre
VM	Virtual Machine
V-RSU	Virtual road-side unit
XML	Extensible Markup Language

2. Scenario 1: Personal Travel Companion

2.1 Design

2.1.1 Prototype definition

Instant Mobility Scenario 1 prototype simulates in a realistic way, the urban and suburban trips in a city, based on the population's movement statistics. The city map displays the live positions of buses, travellers and cars and the journey details of each traveller could be obtained in real time. A statistics panel provides key figures of the simulation. The demonstration shows also the dialog between a traveller and a car driver involved in a common-ride.

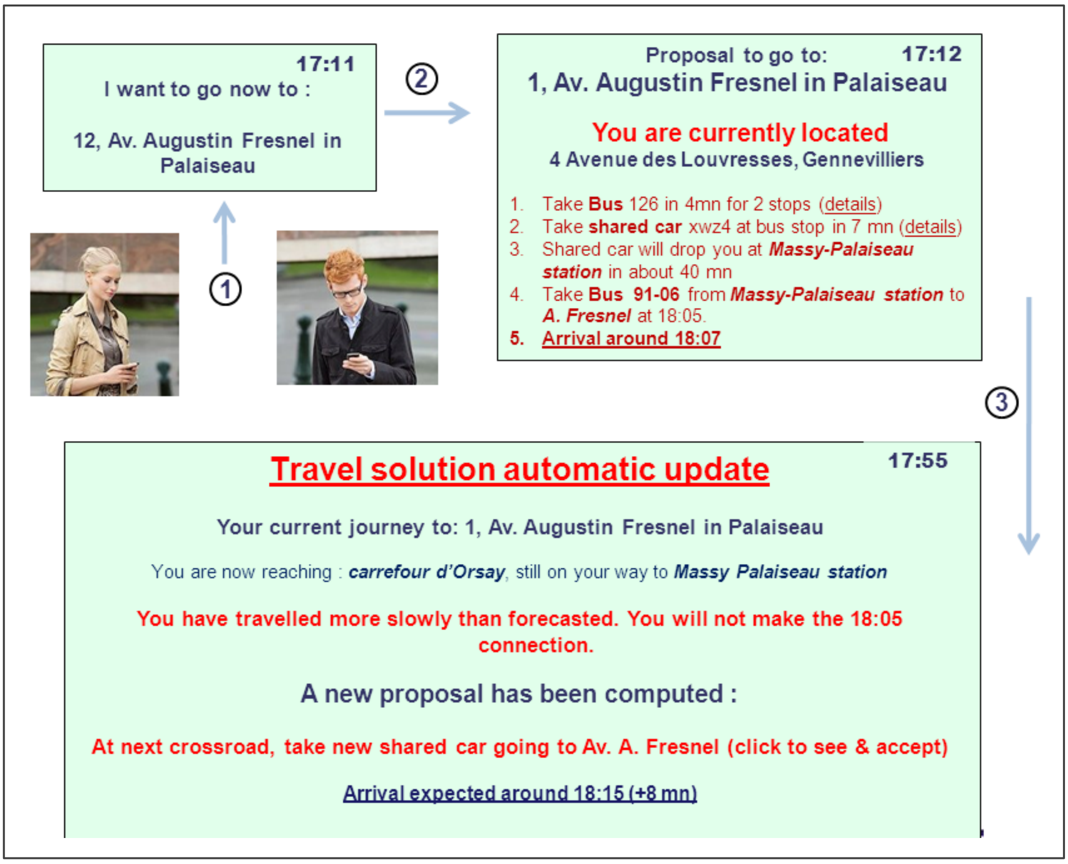


Figure 2. Scenario 1 - Ride sharing prototype layout

2.1.2 Architecture

Here is the logical view of the multi-modal system main prototype.

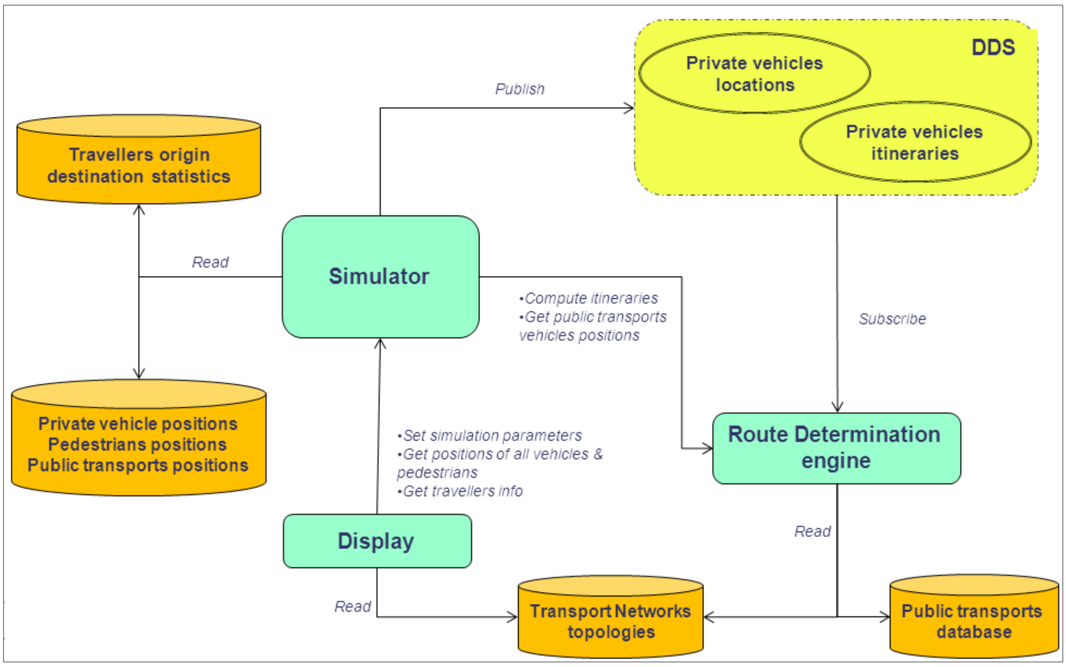


Figure 3. Scenario 1 – Prototype logical architecture

As shown above, there are three domain specific enablers:

- The Route Determination Engine.
- The Situation Display
- The Simulator

The system uses an OMG Data Distribution Service middleware to achieve linear scalability (application can scale by adding more processing units). There can be as many instances of the domain specific enablers as required. For example, our demo uses four instances of the "Route Determination Engine" all of them could be running on separate machines interconnected through a local network.
All potential points of contentions have been identified and removed through data distribution, aggregation and filtering (e.g. situation displays). One of the key benefits is that all domain specific enablers are entirely decoupled. The application is fully distributed and most of the modules are pluggable.

2.1.3 Development of domain specific enabler

Development is done in Java, Scala, C++, JavaScript for the mobile interface, and Repast Simphony simulation platform.

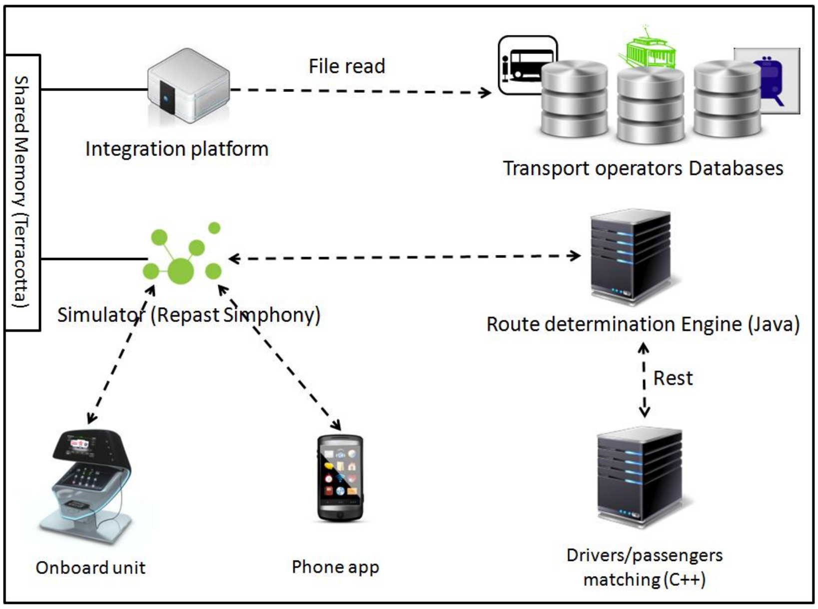


Figure 4. Scenario 1 prototype components and nodes (version 1)

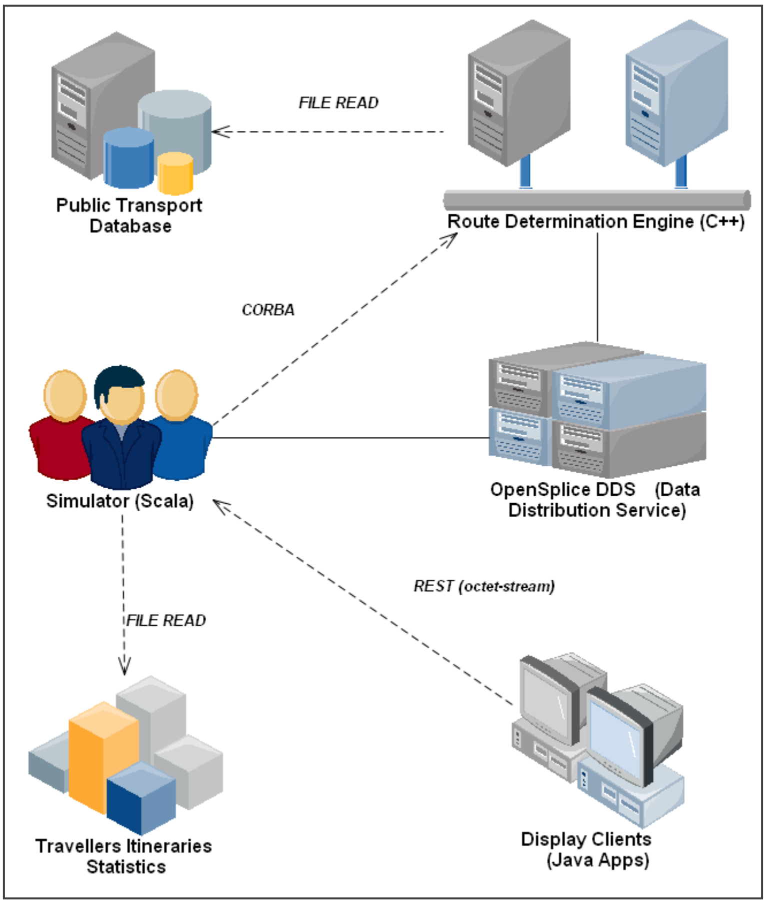


Figure 5. Scenario 1 prototype components and nodes (final version)

2.1.3.1 The Route Determination Engine (C++ and Java)

This domain specific enabler provides itineraries for travellers and drivers alike (multi-modal solutions for travellers and the fastest routes for drivers). Multi-modality requires to efficiently mixing multiple means of transportation in a single consistent and optimal journey proposal. This enabler takes into account all the means of transportation (private cars, buses, metro) in a consistent manner making it easy to add new transportation mode. Each transportation mode has its own requirements (sorted by increasing complexity)

- Pedestrian only
- Pedestrian + Public transports (+ pedestrian)
- Pedestrian + car (+ pedestrian)
- Pedestrian + Public transports + car (+ pedestrian)
- Pedestrian + car + Public transports (+ pedestrian)
- ...

The complexity comes from the increasing combination of pick-up and drop points for acceptable solutions in multi-modal schemes. The final version of the engine is developed in C++.

A first version of the Route determination Engine has been developed (cf. Figure 4), and was composed of two different modules. The first module, the ridesharing engine developed in C++, is responsible for the determination of ridesharing possibilities, provided a set of drivers who are willing to share their rides. The set of drivers is continuously updated by the main route determination engine, developed in Java. This second module is responsible for the calculation of multimodal itineraries (pedestrian and public transport) and combines them with the ridesharing parts of the itinerary.

2.1.3.2 The Simulator (Scala and Repast Simphony)

The simulator has multiple purposes. First, the simulation of the travellers/drivers by performing itinerary requests to the route determination engines. These requests are based on the actual statistical data of source-destination needs of the users. Second, the simulation moves the cars, buses and other mean of transport taking into account historical flow speed at the time of travel. The simulator is also responsible of the temporal coherence of the movements and of the matchmaking between travellers and means of transport. Third, the simulator provides an interface (REST) to the mobiles to show itineraries of selected travellers (selection being done by the situation display). The mobile applications (which include the on-board unit) are pure javascripts calling the simulator REST services.

2.1.3.3 The Situation Display (Java)

The situation display shows the position of travellers and mean of transportation. It calculates and shows the mobility metrics (e.g. percentage of multi-modal itineraries). It allows for the selection of the traveller/driver pair whose itinerary (and initial hand-shake dialog) can be shown on the mobiles. The display interface authorizes new events to be injected to the simulation (e.g. car broken) leading to automatic rerouting of the potential travellers when required.

2.1.1 Reusability

The “Route Determination Engine” enabler can be reused standalone (CORBA interface). The two others (simulator and situation display) are not (they require each other’s). To be reused the “Route Determination Engine” requires a DDS Pub/Sub instance running with information about all the means of transportations published.

2.1.2 FI-WARE integration

The prototype integrates with the **Data Handling GE**[1] to allow/disallow diffusion of travellers/drivers pictures based on users predefined policies. If the Data Handling GE is disabled or if the policy does not allow for picture diffusion, generic pictures (male or female) are shown on the traveller mobiles and on-board units.

In addition the prototype integration with Identity Management GE that provides user management capabilities has been tested (registration, login, etc.). A Java server that interacts with the IDM GE has been developed, and now it’s ready to be integrated into the existing Personal Traveler Companion server.

[1] http://forge.fi-ware.eu/plugins/mediawiki/wiki/fiware/index.php/Internet_of_Things_(IoT)_Services_Enablement#IoT_Data_handling

2.2 Deployment

2.2.1 Hardware environment

The prototype does not require specific hardware environment.

2.2.2 Implementation

All communications to the simulator are REST based: JSON for the mobiles (javascript) and octet-stream for the main situation display (uses the default Java serialization mechanism for the encoding/decoding).

We are using space based architecture for achieving linear scalability (application can scale up with the number of users/travellers by adding more processing units). This type of scalability is supported by the OpenSplice DDS middleware (a key enabler for us). All the potential points of contentions are identified and removed through data distribution, aggregation and filtering (e.g. situation displays). Finally, the algorithms we use may not provide the optimum solution 100 % of the time (problem NP-complete), but they do work fast (O(n)) and provide excellent average solutions.

2.3 Testing

2.3.1 Testing configuration

The prototype is based the Bordeaux urban area. Overall, the purpose of the prototype is to be a fair simulation of the local reality.

Some figures about the metropolitan area of Bordeaux:

- Urban area: 96 municipalities;
- Daily urban trips: 3, 224 millions.

Mode	PT	Metro	Bike	Motorbike	Private car	Other	Total
Volume	302 072	675 758	108 118	44 012	2 045 042	49278	3 224 280

Figure 6. Bordeaux modal split

Figure 7. Map of Bordeaux urban area

- The prototype use data from effective origin-destination travels from home to work and home to school;
- The simulation currently use 450 K travel per days (departures only), i.e. 10% of effective journeys in the urban area;
- The density of cars and buses reflect the reality;
- The journey density along the day is preserved.

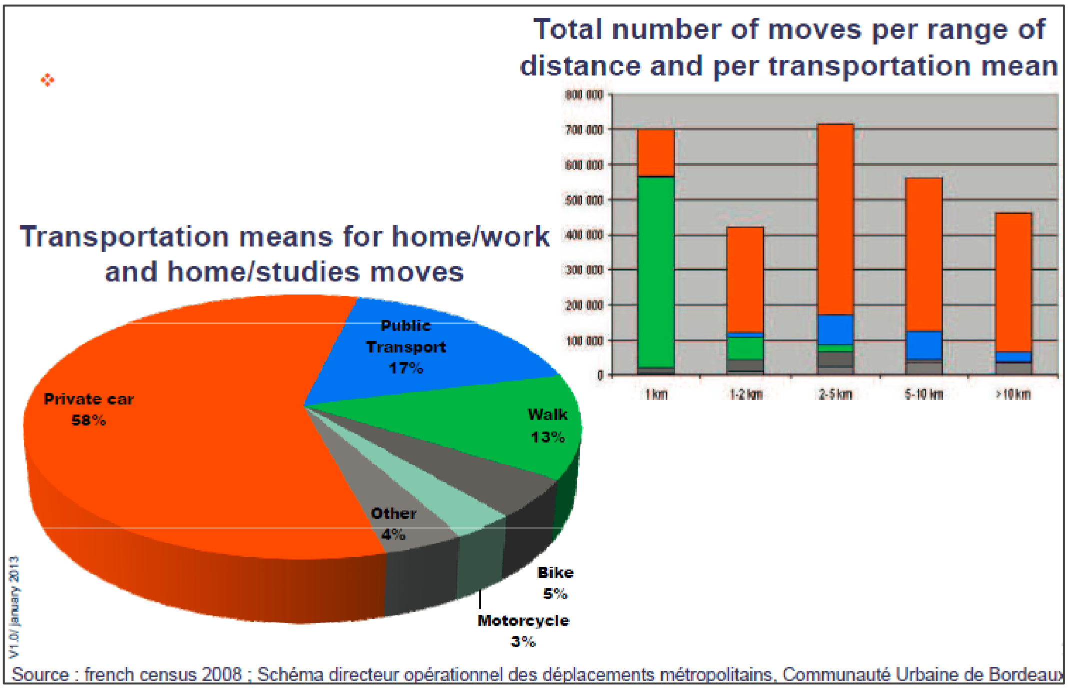


Figure 8. Bordeaux modal split survey results

2.3.2 Integration with a Transport Management System

The purpose of this test is to integrate an actual public transport traffic management system with the Multimodal journey optimizer prototype, in order to take into account various real-time modifications in the transport offer (delays, trip cancellation, etc.) in the route determination and journey monitoring algorithms. This is done by simulating Bordeaux's Tramway B line with Thales industrial-grade Automatic Train Supervision (ATS) product, and by integrating this product with the Multimodal journey optimizer prototype on the Communauté Urbaine de Bordeaux demonstrated at Barcelona's MWC 2013. ATS is configured to simulate the Tramway B line and modified to export real-time information on the operation of this line to the Multimodal journey optimizer prototype using the GTFS-Realtime (General Transit Feed Specification-Realtime) format. The Multimodal journey optimizer prototype is modified to handle the reception of these updates and integrate them in both the route planning and journey monitoring. The journey simulation will be modified to share a common simulation clock with ATS. At least one GTFS-RT event type (trip cancellation) shall be handled on both ATS and Multimodal journey optimizer prototype for Instant Mobility final review.

2.3.3 Testing results

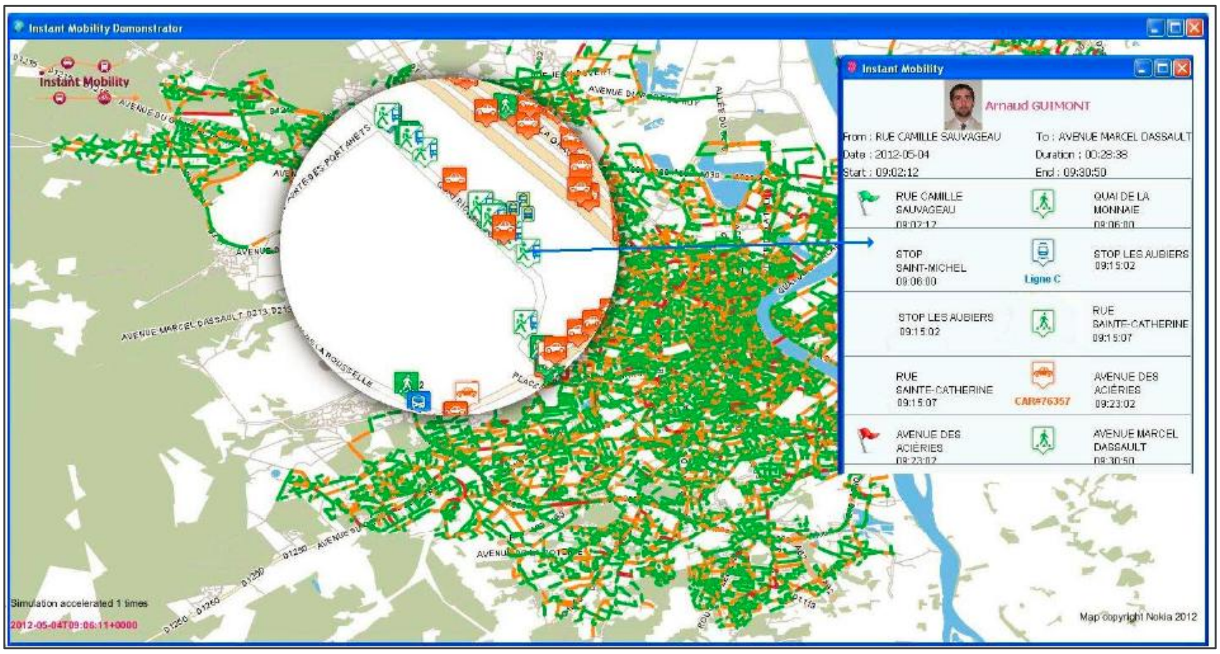
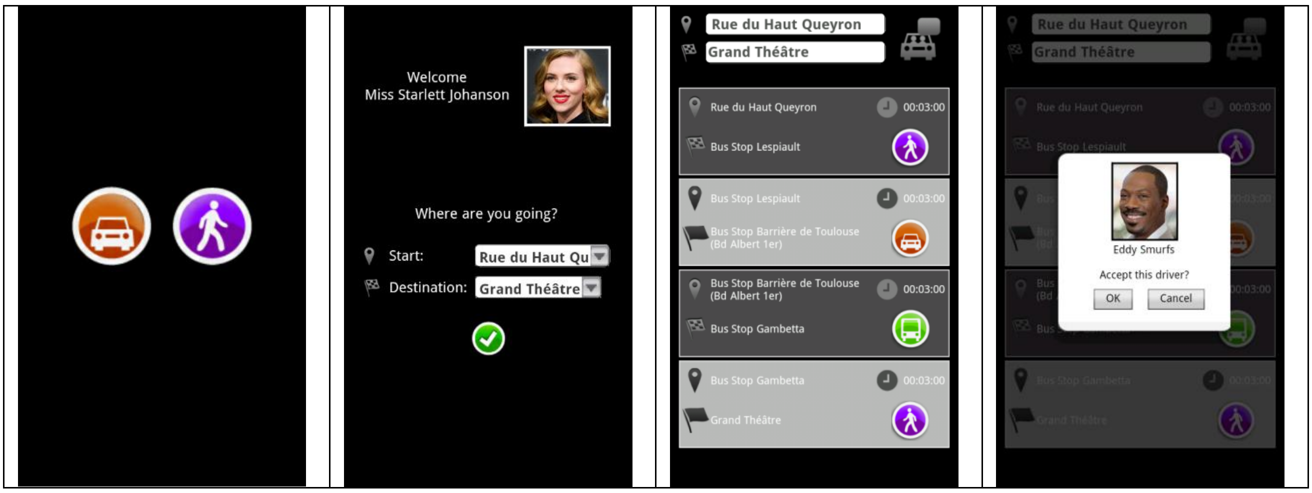


Figure 9. Scenario 1 prototype – Operator GUI

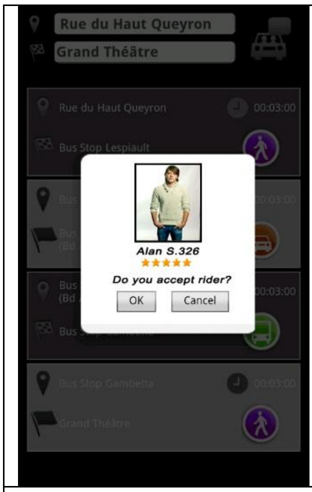
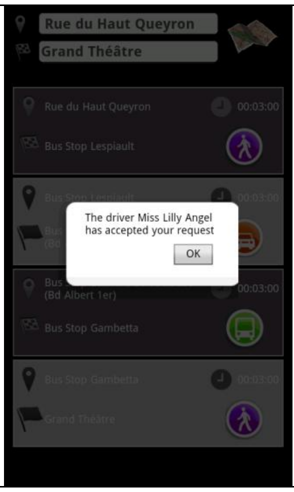
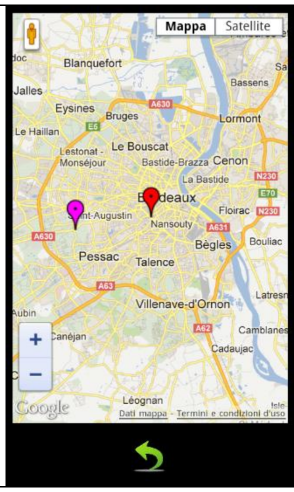
One of the main discoveries when looking at the statistical data generated when running the simulation is that by allowing common-ride we improve the travel time by at least 5 min for at least 1/3 of the travellers (it is assumed that only 10 % of the drivers are willing to use the system and pick-up travellers).

2.3.1 Ride Sharing User Experience

The Ride Sharing mobile application has been developed in in order to show the possible user experience. It was developed in HTML/Css/javascript and can thus be accessed by any standard web browser. There are two users involved: a pedestrian (rider) looking for a ride and a driver that potentially is able and willing to offer a ride There is a single application which is thus twofold; the first splash screen has the task to determine the user's role.



Initial Splash Screen that allows to choose between driver and traveller interface	Here the user can choose the start and the destination (both driver and rider case)	The system provides an itinerary to the rider. The itinerary has a ride sharing segment, indicated by the multimodal icon in the top-right corner of the screen.	After clicking on multimodal icon the application asks the rider if he wants to accept the proposed driver

<table>
<tr>
<td>
</td>
<td>
</td>
<td>
</td>
<td></td>
</tr>
<tr>
<td>If rider accepts, also the driver is asked to accept the proposed rider.</td>
<td>When the driver accepts the rider, the system notifies it to the driver and vice versa</td>
<td>Here the two users can see their respective positions.</td>
<td></td>
</tr>
</table>

3. Scenario 2: Smart City Logistics

3.1 Design

3.1.1 Prototype definition

The purpose of the “smart city logistics scenario” is to increase the sustainability of city logistics by optimizing the vehicle utilization, in order not to use more vehicles than necessary and also to optimize the way the vehicles are driven. A purpose specific to the “dynamic drop point” application, on which this prototype is focused, is to increase the convenience for the consignee, by offering flexibility in choosing a suitable time and location for dropping off goods, in order to avoid unnecessary transports, both by the transporter, who can avoid trying to deliver goods to someone who is not at home, but also by the consignee, who can avoid make an additional trip in order to pick up goods.

The pre-condition of the smart city logistics prototype is that a transport need has been detected and transport resources has been registered in the system; that is, a set of transport bookings has been submitted to the transport exchange portal via the web service interface. Based on available transport resources and submitted transport bookings; itineraries are planned by the system.
By using the Consignee App, the goods receiver can select calendar events to upload, which are then used by the system as input when planning itineraries. Using the vehicle simulation web application, itineraries can be downloaded to a simulated vehicle, which executes them on a map and report back to the transport exchange portal.

3.1.2 Architecture

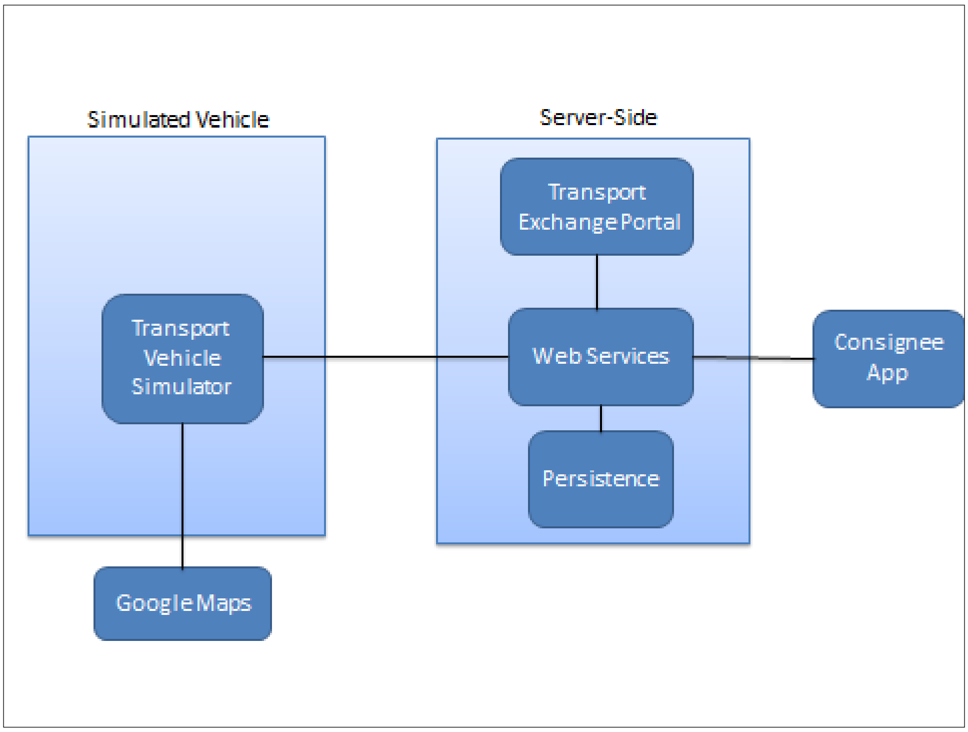


Figure 10: Smart City Logistics prototype architecture

The Smart City Logistics prototype consists of three main nodes; the vehicle, the server-side and the consignee's mobile device. The main components of the server side are the web services, which implement the vehicle transport management interface, the transport resource manager and the consignee API. These web services are domain specific enablers which are described in detail in D4.13. There are several clients to these web services; the itinerary processor develops itineraries based on the transport need and available resources, the transport exchange portal provides an overview of the transport bookings and itineraries, the consignee app provides the dynamic drop point service to the end user and the simulated vehicle executes the itineraries.
The simulated vehicle is a client to the vehicle transport management interface. It is implemented as a Google Web Toolkit application running inside a web browser, which lets the user select a vehicle itinerary and follow the execution of the itinerary on a map.

3.1.3 Development of domain specific enablers

The development activities performed to create this prototype consisted of both development of off-board (server-side) and on-board (mobile) enablers, as well as creating a simulation of a freight vehicle, performing the cargo distribution. This section lists the enablers developed. The same terminology as in Instant Mobility deliverable D4.13, where the details of the enablers are documented, is used.

3.1.3.1 On-board enablers

- ***Consignee App***

The Consignee App is a smartphone app, developed for Android 4.0.x, which is used by the consignee to share data on his/her whereabouts in order to agree with the transport planner about a convenient time and location to receive goods. The consignee app first retrieves transport booking information for transport bookings of which the consignee expects a delivery. The consignee then selects the calendar events for which he/she is able to receive the goods. These calendar events are then uploaded to the transport exchange portal. When the calendar events have been processed a suggestion for a drop point is presented to the consignee, who is then able to accept or reject it. In Section 3.3 a typical flow of interactions is presented together with screenshots.

- ***Remote Internet tethering***

The remote Internet tethering enabler is composed by developments on an on-board unit (having at least the function of IP router) and developments on smartphones. The objective of this enabler as detailed in D4.11 in the context of logistic and/or goods transportation is to offer a wider range of cellular connectivity to the on-board unit (OBU) by using opportunistically the cellular network interfaces of smartphones (3G/4G) connected to the OBU.
Within a vehicle an IP router, or "OBU", interconnects communicating devices in the vehicle (sensors, goods, etc.) to the infrastructure, i.e., a management platform in the Internet.
In some scenarios, the truck - or by using a more generic term, the vehicle -, will use a generic OBU that do not offer any cellular network interface by itself. Such network interface, requiring a monthly subscription and then more expensive, could be deployed in tier-one vehicle however. The remote Internet tethering enabler, allows the OBU to take the control of neighboring smartphones, wirelessly, to transfer data information such as sensors data, CAN bus data, goods monitoring data to the management server in the Internet.
The developed functionalities on the smartphones (Android 4.1.2) give the capability to the user to configure and authorize the use of the cellular interface by the OBU through a GUI.
The developed functionalities on the OBU (x86 platform) gives the capability to use the cellular network interfaces of neighboring smartphones, as well as dispatching data stream to one or several smartphones following a load balancing algorithm.

3.1.3.2 Off-board enablers

This section lists the server side enablers that were developed for the smart city logistics prototype. This set of enablers consists of web applications and web services, which are developed for a Java Servlet platform. The web services are based on the JAX-RS standard, which is the Java API for RESTful web services. The data transferred via the web services is formatted on the JSON format, which has several advantages compared to the XML format as it is less verbose and also works well with the database management system. MongoDB is used as the database management system; it is a document-oriented database that uses the BSON format, which is a binary JSON format. Since the MongoDB utilizes BSON, a JSON mapper, in this case Jackson, can be used to automate the mapping of the data format used in the web service interfaces to the data format used in the database. This contributed to a shorter development time than what is normally expected on applications with similar complexity.

The Transport Exchange Portal web application was developed using the Vaadin framework, which is similar to Google web toolkit (GWT) in that it enables the developer to write a web browser based user interface in the higher level Java language, without having to bother about details about HTML and JavaScript, but different in that the presentation logic executes on the server side. The main advantages are that less data needs to be downloaded by the browser on application start up and shorter development time due to a higher abstraction level than coding directly in HTML/JavaScript and less time spent on dealing with web browser interoperability issues.

The simulated vehicle is a web application, developed with GWT. Here GWT was chosen instead of Vaadin, since the functionalities needed from the Google Maps API was not exposed through the Vaadin API. This implies that the logic of the vehicle simulator is run inside the web browser including the client to the RESTful web services.

For a complete documentation of the enablers, including interface specifications, see the corresponding topics in D4.13.

- ***Transport Exchange Portal***

The Transport Exchange Portal functionalities developed for the prototype consist of a web portal for monitoring incoming transport bookings and transport missions, which are updated in real time as they are being processed by the transport resources. The transport exchange portal also consists of a number of other enablers, which implement the web service interface that the vehicle- and consignee applications use to interact with the system.

- ***Vehicle Transport Management Interface***

This is a RESTful web service interface, implemented as a JAX-RS web service. It contains functions for retrieving transport itineraries, reporting status on a transport mission, reporting current position and verifying the driver.

- ***Transport Resource Manager***

This is a RESTful web service interface, implemented as a JAX-RS web service. It contains functions for creating, retrieving and deleting transport resources and for authenticating the transport manager.

- ***Consignee API***

This is a RESTful web service interface, implemented as a JAX-RS web service. It contains functions for authenticating the consignee, retrieving, creating and deleting calendar events, retrieving drop points and accepting drop point suggestions.

- ***Itinerary Processor***

For the prototype a subset of itinerary processor functionalities was developed to cover the functionality needed by the demonstration scenario. It was originally foreseen that this enabler was supposed to build upon a generic scheduler enabler, which was proposed to FI-WARE. FI-WARE however, never expressed any interest in developing this functionality.

- ***Transport Resource Persistence***

The transport resource persistence is implemented not as a separate enabler, but as a part of the transport resource manager enabler, using the MongoDB Jackson mapper, which provides seamless mapping between the JSON format used by the web services, and the BSON (binary JSON) format used internally by the MongoDB database.

- ***Transport Booking Persistence***

The transport resource persistence is implemented not as a separate enabler, but as a part of the itinerary processor enabler, using the MongoDB Jackson mapper, which provides seamless mapping between the JSON format used by the web services, and the BSON (binary JSON) format used internally by the MongoDB database.

3.1.4 Reusability

This section discusses reusability from a software architecture point-of-view. The sections on deployment, "**Erreur ! Source du renvoi introuvable.** Hardware environment" and "3.2.2 Software environment for server-side" discuss the environment needed to set up these enablers thus also addressing the conditions for reuse at the platform level.

3.1.4.1 On board enablers

The Consignee App exchanges information with the server side using a RESTful web service interface. If different business logic is needed on the server side, the implementation of the web service interfaces can be replaced, while the Consignee App can be reused without modifications.

3.1.4.2 Off board enablers

The Transport Exchange Portal provides a user interface which is loosely coupled to the business logic via RESTful web services. Since the business logic and presentation layer is clearly separated, it is possible to reuse the existing Transport Exchange Portal functionality developed for the demonstrator, while, for example, further refining the functionalities of the underlying business logic, or the underlying business logic could be reused from a completely new Transport Exchange Portal, perhaps developed on a different platform.

The business logic enablers; vehicle transport manager interface, transport resource manager, consignee API, itinerary processor and the persistence components are bundled together because they operate on common data. The external interface is RESTful web services, which means that implementations can be replaced with ones exposing the same external interface.

3.1.5 FI-WARE integration

As part of WP4 the FI-WARE these generic enablers were analyzed in order to see how they could provide value to the solution. The result of this analysis was that gaps was detected that must be addressed by FI-WARE before the generic enablers can be incorporated in a meaningful way. The Identity Management only supports SAML for API based authentication/authorization. SAML doesn't however work with RESTful web services, since it is dependent on a SOAP header for transferring the SAML messages. The FI-WARE Big Data Enabler[2] today consists of a map/reduce platform, which is a specialized tool for tasks such as extracting information from large data streams, such as log files generated by telecom systems. If the Big Data generic enablers are further developed to also include tools for supporting other big data challenges, such as, for example, document oriented storage, it will be relevant for a larger set of use cases, including the ones within smart city logistics.

3.2 Deployment

3.2.1 Hardware environment

The hardware environment of the demonstrator is fairly straightforward. The server side runs on commodity x86 hardware. Since the datasets used in the demonstrator are fairly small, the hardware requirement is moderate; a mid-end laptop will do. The simulated vehicle runs inside a recent version of a mainstream web browser, also on x86 hardware. The Consignee App in the demonstrator runs on a Samsung Galaxy S3 smartphone. Any smartphone capable of running Android 4.0.x would have sufficed.

Although not presented as part of the demonstrator in Barcelona MWC, the developed domain specific enabler "remote Internet tethering" relied on an x86 Intel In-Vehicle Infotainment (IVI) on-board unit. This low-cost hardware is composed by a small electronic board that features rear-seat entertainment (audio and video through a small screen), Ethernet connectivity, CAN bus connectivity and, two USB ports. The enabler makes use of one USB port to connect a WiFi USB dongle.
By using such specific hardware, one assesses the possibility of integrating the in-vehicule part of the prototype (CAR API enabler, Mobile router enabler, remote Internet tethering enabler and, in-vehicle smartphone application enabler) in a real vehicle.

3.2.2 Software environment for server-side

The server-side on the system is deployed as web applications in a Tomcat servlet container (version 7.0.29). HTTP Basic Authentication combined with HTTPS is used to protect it, thus the Tomcat HTTPS connector needs to be configured, using the supplied key store. For authentication a role with role name "im" and a user assigned to the role needs to be configured.

A standard installation of the MongoDB database is used (version 2.0.7). No additional configuration is required. An environment variable, IM_MONGO_URI, containing the database connection string, should be defined on the machine running the Tomcat server.

3.2.3 Implementation

The demonstrator in Barcelona MWC used a single computer, which executed both the server side of the system; the database, the web services and the transport exchange portal. This computer

[2] http://forge.fi-ware.eu/plugins/mediawiki/wiki/fiware/index.php/Data/Context_Management#Big_Data_Analysis

was connected to Internet via its Ethernet interface, for access to Google Maps and to a WLAN access point, in order to be accessible by WLAN from the smartphone on a static IP address. This implementation was chosen to avoid being dependant on wireless Internet connections available at the congress, which may not have been stable enough to perform the demonstrations.

Another setup, more close to a real deployment, was also used. This setup used a cloud computing infrastructure provided by DLR, together with a 3G connected smartphone.

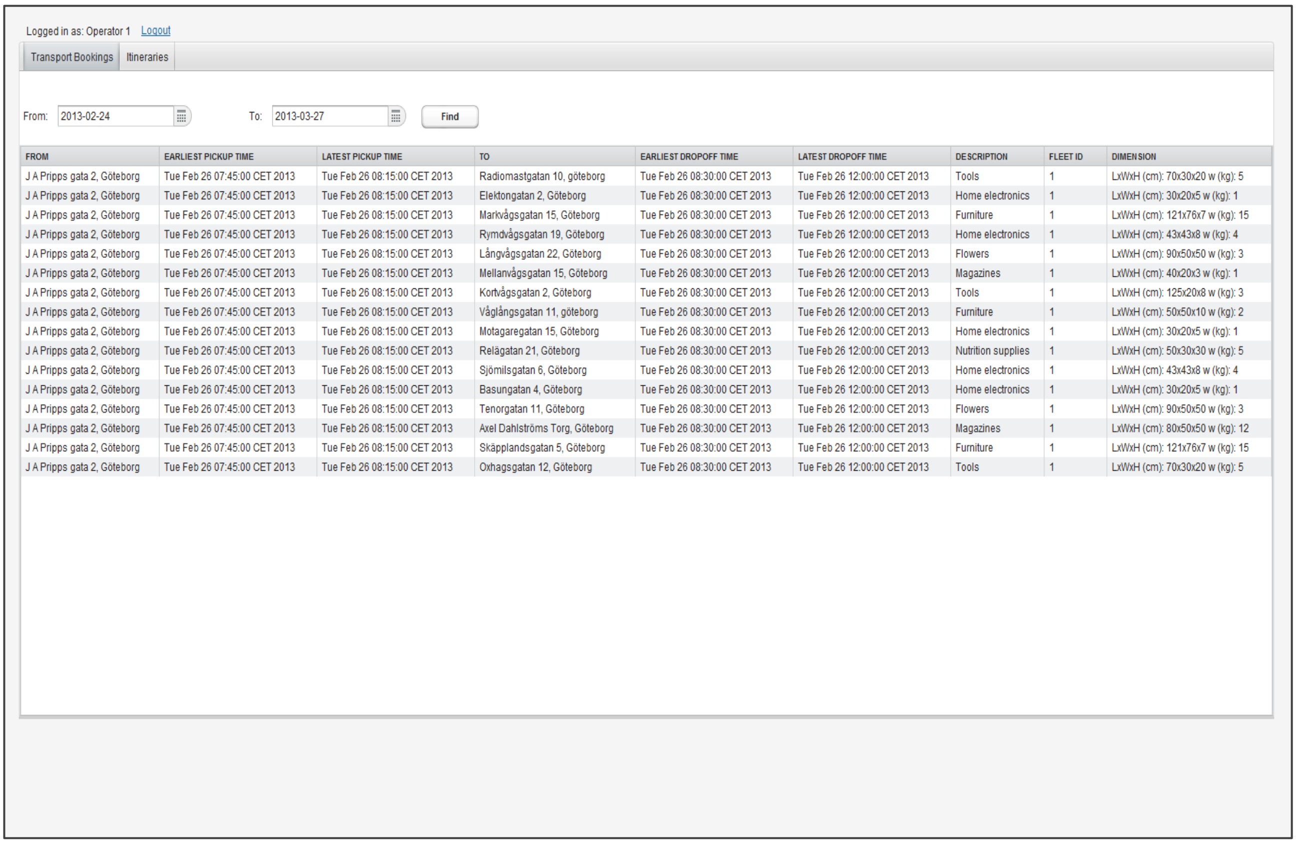
3.3 Testing

3.3.1 Testing configuration

Testing was performed on both the configuration used at Barcelona MWC, with the server-side running on the local computer and on the cloud computing setup. This was necessary in order to verify that certain functionality, such as Google Maps license keys, would work both on a localhost server and on a real server setup.

3.3.2 Testing results

A use case scenario for testing the developed software is carried out in a series of steps, thoroughly explained in this section. In this particular scenario, we assume that the Transport Booker requests 16 different goods deliveries, to 16 different consignees, from 16 different consignors. The same Transport Planner is used to provide resources for the three different deliveries, and the Transport Booker requests that the goods are to be delivered according to the drop-off times described in Figure 11.



FROM	EARLIEST PICKUP TIME	LATEST PICKUP TIME	TO	EARLIEST DROPOFF TIME	LATEST DROPOFF TIME	DESCRIPTION	FLEET ID	DIMENSION
J A Pripps gata 2, Göteborg	Tue Feb 26 07:45:00 CET 2013	Tue Feb 26 08:15:00 CET 2013	Radiomastgatan 10, göteborg	Tue Feb 26 08:30:00 CET 2013	Tue Feb 26 12:00:00 CET 2013	Tools	1	LxWxH (cm): 70x30x20 w (kg): 5
J A Pripps gata 2, Göteborg	Tue Feb 26 07:45:00 CET 2013	Tue Feb 26 08:15:00 CET 2013	Elektongatan 2, Göteborg	Tue Feb 26 08:30:00 CET 2013	Tue Feb 26 12:00:00 CET 2013	Home electronics	1	LxWxH (cm): 30x20x5 w (kg): 1
J A Pripps gata 2, Göteborg	Tue Feb 26 07:45:00 CET 2013	Tue Feb 26 08:15:00 CET 2013	Markvågsgatan 15, Göteborg	Tue Feb 26 08:30:00 CET 2013	Tue Feb 26 12:00:00 CET 2013	Furniture	1	LxWxH (cm): 121x76x7 w (kg): 15
J A Pripps gata 2, Göteborg	Tue Feb 26 07:45:00 CET 2013	Tue Feb 26 08:15:00 CET 2013	Rymdvågsgatan 19, Göteborg	Tue Feb 26 08:30:00 CET 2013	Tue Feb 26 12:00:00 CET 2013	Home electronics	1	LxWxH (cm): 43x43x8 w (kg): 4
J A Pripps gata 2, Göteborg	Tue Feb 26 07:45:00 CET 2013	Tue Feb 26 08:15:00 CET 2013	Långvågsgatan 22, Göteborg	Tue Feb 26 08:30:00 CET 2013	Tue Feb 26 12:00:00 CET 2013	Flowers	1	LxWxH (cm): 90x50x50 w (kg): 3
J A Pripps gata 2, Göteborg	Tue Feb 26 07:45:00 CET 2013	Tue Feb 26 08:15:00 CET 2013	Mellanvågsgatan 15, Göteborg	Tue Feb 26 08:30:00 CET 2013	Tue Feb 26 12:00:00 CET 2013	Magazines	1	LxWxH (cm): 40x20x3 w (kg): 1
J A Pripps gata 2, Göteborg	Tue Feb 26 07:45:00 CET 2013	Tue Feb 26 08:15:00 CET 2013	Kortvågsgatan 2, Göteborg	Tue Feb 26 08:30:00 CET 2013	Tue Feb 26 12:00:00 CET 2013	Tools	1	LxWxH (cm): 125x20x8 w (kg): 3
J A Pripps gata 2, Göteborg	Tue Feb 26 07:45:00 CET 2013	Tue Feb 26 08:15:00 CET 2013	Våglångsgatan 11, göteborg	Tue Feb 26 08:30:00 CET 2013	Tue Feb 26 12:00:00 CET 2013	Furniture	1	LxWxH (cm): 50x50x10 w (kg): 2
J A Pripps gata 2, Göteborg	Tue Feb 26 07:45:00 CET 2013	Tue Feb 26 08:15:00 CET 2013	Motagaregatan 15, Göteborg	Tue Feb 26 08:30:00 CET 2013	Tue Feb 26 12:00:00 CET 2013	Home electronics	1	LxWxH (cm): 30x20x5 w (kg): 1
J A Pripps gata 2, Göteborg	Tue Feb 26 07:45:00 CET 2013	Tue Feb 26 08:15:00 CET 2013	Relägatan 21, Göteborg	Tue Feb 26 08:30:00 CET 2013	Tue Feb 26 12:00:00 CET 2013	Nutrition supplies	1	LxWxH (cm): 50x30x30 w (kg): 5
J A Pripps gata 2, Göteborg	Tue Feb 26 07:45:00 CET 2013	Tue Feb 26 08:15:00 CET 2013	Sjömilsgatan 6, Göteborg	Tue Feb 26 08:30:00 CET 2013	Tue Feb 26 12:00:00 CET 2013	Home electronics	1	LxWxH (cm): 43x43x8 w (kg): 4
J A Pripps gata 2, Göteborg	Tue Feb 26 07:45:00 CET 2013	Tue Feb 26 08:15:00 CET 2013	Basungatan 4, Göteborg	Tue Feb 26 08:30:00 CET 2013	Tue Feb 26 12:00:00 CET 2013	Home electronics	1	LxWxH (cm): 30x20x5 w (kg): 1
J A Pripps gata 2, Göteborg	Tue Feb 26 07:45:00 CET 2013	Tue Feb 26 08:15:00 CET 2013	Tenorgatan 11, Göteborg	Tue Feb 26 08:30:00 CET 2013	Tue Feb 26 12:00:00 CET 2013	Flowers	1	LxWxH (cm): 90x50x50 w (kg): 3
J A Pripps gata 2, Göteborg	Tue Feb 26 07:45:00 CET 2013	Tue Feb 26 08:15:00 CET 2013	Axel Dahlströms Torg, Göteborg	Tue Feb 26 08:30:00 CET 2013	Tue Feb 26 12:00:00 CET 2013	Magazines	1	LxWxH (cm): 80x50x50 w (kg): 12
J A Pripps gata 2, Göteborg	Tue Feb 26 07:45:00 CET 2013	Tue Feb 26 08:15:00 CET 2013	Skäpplandsgatan 5, Göteborg	Tue Feb 26 08:30:00 CET 2013	Tue Feb 26 12:00:00 CET 2013	Furniture	1	LxWxH (cm): 121x76x7 w (kg): 15
J A Pripps gata 2, Göteborg	Tue Feb 26 07:45:00 CET 2013	Tue Feb 26 08:15:00 CET 2013	Oxhagsgatan 12, Göteborg	Tue Feb 26 08:30:00 CET 2013	Tue Feb 26 12:00:00 CET 2013	Tools	1	LxWxH (cm): 70x30x20 w (kg): 5

Figure 11: Transport Booking Manager HMI

Figure 11 shows a screenshot of the Transport Booking Manager HMI for this test example. When the Transport Booker has provided the requests described in Figure 11, the Itinerary Processor will try to find match between these requests and the available resources. In this use case, the Transport Planner has provided three different resources to the Transport Exchange Portal System.

The Itinerary Processor will, from the delivery requests given in Figure 11, suggest an Itinerary; the outcome is shown in Figure 13. The itinerary suggestion is accepted automatically in this use case, without any interaction with the Transport Resource Management HMI. At the same time, when the transport bookings are registered in the Transport Exchange Portal System - the consignee API, which is in the form of an Android application, will download the transport bookings and display them to the consignee. Figure 12 shows how the transport booking is displayed to the consignee – the presented view is for the consignee that is awaiting the “Home electronics” delivery.

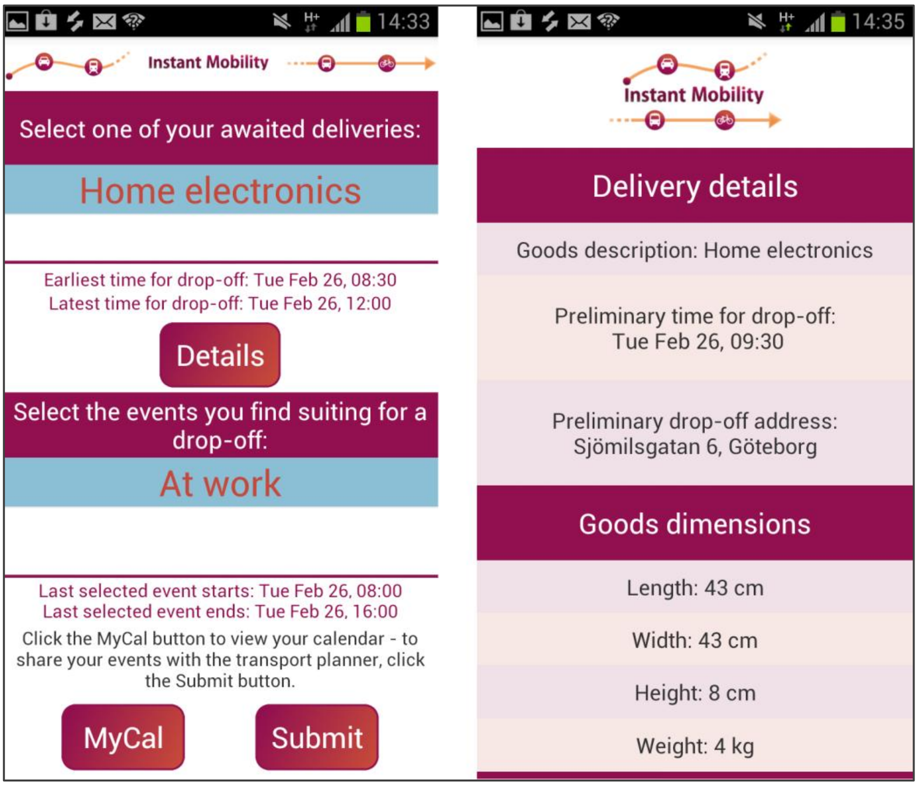
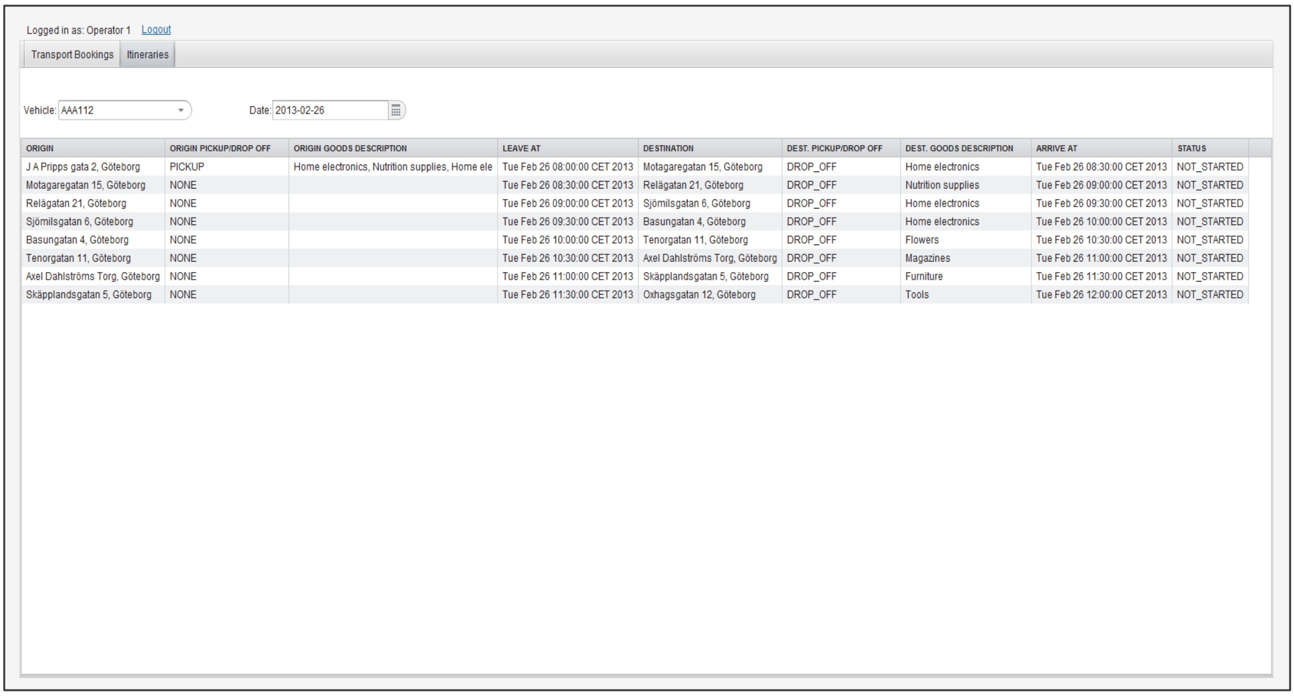


Figure 12: Consignee API

The left hand side of Figure 12 shows how the general view of the consignees deliveries is shown, whilst the right hand side of the figure shows the details view, which is displayed when clicking the Details button. In this case the preliminary drop-off address is set to his/her home address.



Logged in as: Operator 1 Logout

Transport Bookings | Itineraries

Vehicle: AAA112 Date: 2013-02-26

ORIGIN	ORIGIN PICKUP/DROP OFF	ORIGIN GOODS DESCRIPTION	LEAVE AT	DESTINATION	DEST. PICKUP/DROP OFF	DEST. GOODS DESCRIPTION	ARRIVE AT	STATUS
J A Pripps gata 2, Göteborg	PICKUP	Home electronics, Nutrition supplies, Home ele	Tue Feb 26 08:00:00 CET 2013	Motagaregatan 15, Göteborg	DROP_OFF	Home electronics	Tue Feb 26 08:30:00 CET 2013	NOT_STARTED
Motagaregatan 15, Göteborg	NONE		Tue Feb 26 08:30:00 CET 2013	Relägatan 21, Göteborg	DROP_OFF	Nutrition supplies	Tue Feb 26 09:00:00 CET 2013	NOT_STARTED
Relägatan 21, Göteborg	NONE		Tue Feb 26 09:00:00 CET 2013	Sjömilsgatan 6, Göteborg	DROP_OFF	Home electronics	Tue Feb 26 09:30:00 CET 2013	NOT_STARTED
Sjömilsgatan 6, Göteborg	NONE		Tue Feb 26 09:30:00 CET 2013	Basungatan 4, Göteborg	DROP_OFF	Home electronics	Tue Feb 26 10:00:00 CET 2013	NOT_STARTED
Basungatan 4, Göteborg	NONE		Tue Feb 26 10:00:00 CET 2013	Tenorgatan 11, Göteborg	DROP_OFF	Flowers	Tue Feb 26 10:30:00 CET 2013	NOT_STARTED
Tenorgatan 11, Göteborg	NONE		Tue Feb 26 10:30:00 CET 2013	Axel Dahlströms Torg, Göteborg	DROP_OFF	Magazines	Tue Feb 26 11:00:00 CET 2013	NOT_STARTED
Axel Dahlströms Torg, Göteborg	NONE		Tue Feb 26 11:00:00 CET 2013	Skäpplandsgatan 5, Göteborg	DROP_OFF	Furniture	Tue Feb 26 11:30:00 CET 2013	NOT_STARTED
Skäpplandsgatan 5, Göteborg	NONE		Tue Feb 26 11:30:00 CET 2013	Oxhagsgatan 12, Göteborg	DROP_OFF	Tools	Tue Feb 26 12:00:00 CET 2013	NOT_STARTED

Figure 13: Transport Resource Management HMI.

The consignee is asked to select events that he/she finds suiting for a drop-off, in this example the only event that overlaps with the delivery time is a work session. If the consignee would find it more suiting to receive the delivery just after the work session, he/she can click the MyCal button which takes him/her to the in-built calendar on the Android device. He/she can then add an event that explains his/her presence just after the work session has ended.

The left hand side of Figure 14 depicts how he/she submits the recently added event to the Exchange Portal System by clicking on the Submit button.

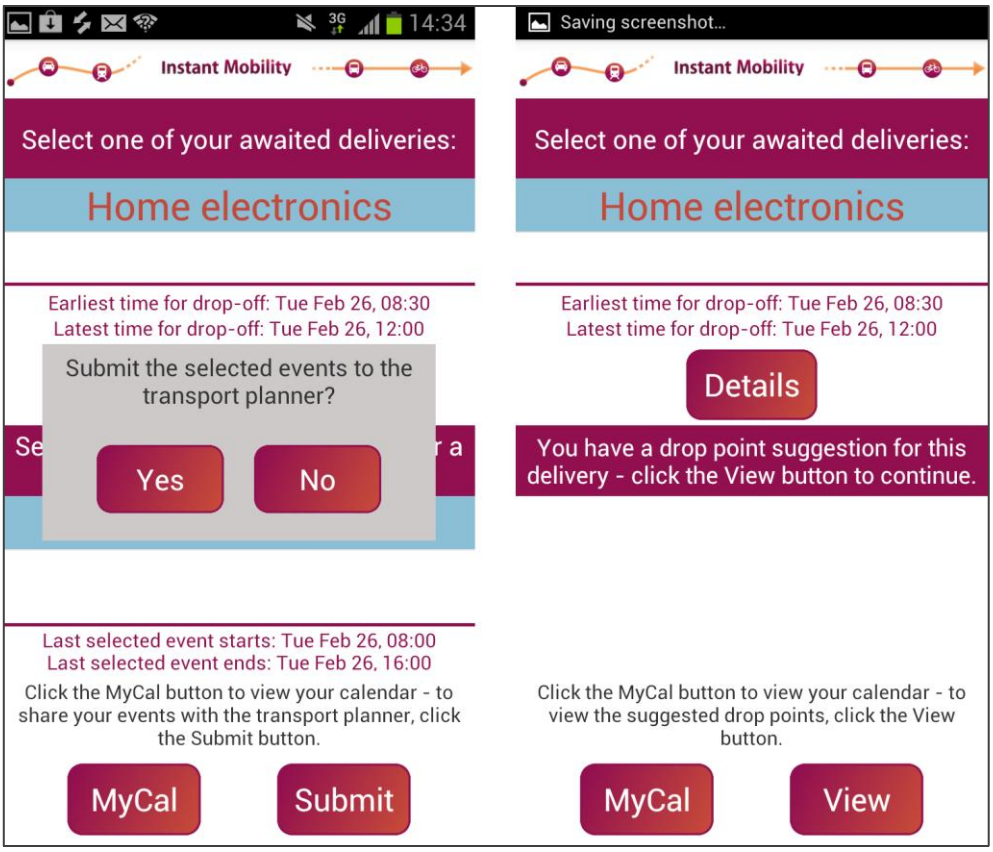


Figure 14: Consignee API

When the selected event is uploaded, the Itinerary Processor will provide a drop point suggestion for that delivery. When the updated drop point data is downloaded to the consignee API, the view in the right hand side of Figure 14 is displayed to the consignee. When the consignee clicks on the View button, the drop point suggestions are displayed as depicted in Figure 15. When the consignee accepts the drop point, the Itinerary processor tries to reschedule the Itinerary to fit the consignee's request.

Upon accepting the drop point, it will automatically be added to the in-built calendar in the consignee's Android device. Figure 16 shows the updated Transport Resource Management HMI, where the itinerary is matched to fit the request of the consignee.

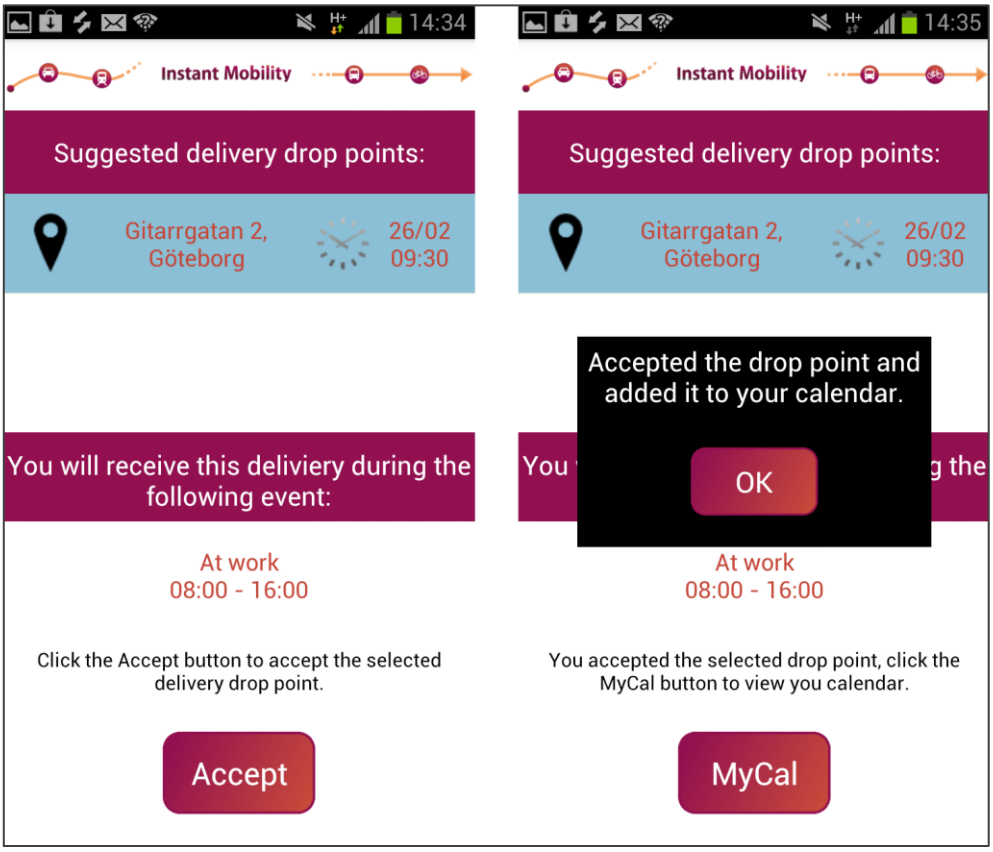
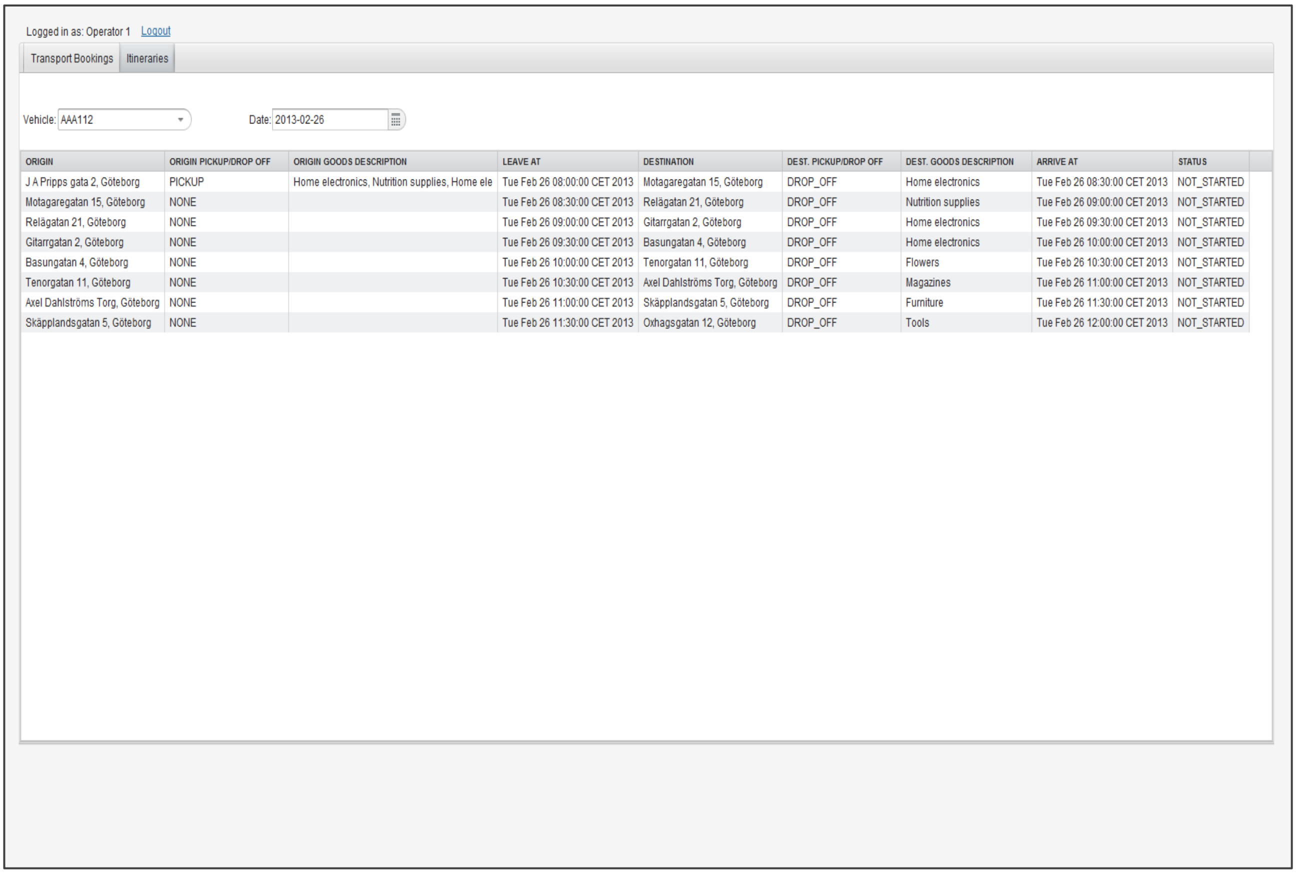


Figure 15: Consignee API


Logged in as: Operator 1 Logout

Transport Bookings | Itineraries

Vehicle: AAA112 Date: 2013-02-26

ORIGIN	ORIGIN PICKUP/DROP OFF	ORIGIN GOODS DESCRIPTION	LEAVE AT	DESTINATION	DEST. PICKUP/DROP OFF	DEST. GOODS DESCRIPTION	ARRIVE AT	STATUS
J A Pripps gata 2, Göteborg	PICKUP	Home electronics, Nutrition supplies, Home ele	Tue Feb 26 08:00:00 CET 2013	Motagaregatan 15, Göteborg	DROP_OFF	Home electronics	Tue Feb 26 08:30:00 CET 2013	NOT_STARTED
Motagaregatan 15, Göteborg	NONE		Tue Feb 26 08:30:00 CET 2013	Relägatan 21, Göteborg	DROP_OFF	Nutrition supplies	Tue Feb 26 09:00:00 CET 2013	NOT_STARTED
Relägatan 21, Göteborg	NONE		Tue Feb 26 09:00:00 CET 2013	Gitarrgatan 2, Göteborg	DROP_OFF	Home electronics	Tue Feb 26 09:30:00 CET 2013	NOT_STARTED
Gitarrgatan 2, Göteborg	NONE		Tue Feb 26 09:30:00 CET 2013	Basungatan 4, Göteborg	DROP_OFF	Home electronics	Tue Feb 26 10:00:00 CET 2013	NOT_STARTED
Basungatan 4, Göteborg	NONE		Tue Feb 26 10:00:00 CET 2013	Tenorgatan 11, Göteborg	DROP_OFF	Flowers	Tue Feb 26 10:30:00 CET 2013	NOT_STARTED
Tenorgatan 11, Göteborg	NONE		Tue Feb 26 10:30:00 CET 2013	Axel Dahlströms Torg, Göteborg	DROP_OFF	Magazines	Tue Feb 26 11:00:00 CET 2013	NOT_STARTED
Axel Dahlströms Torg, Göteborg	NONE		Tue Feb 26 11:00:00 CET 2013	Skäpplandsgatan 5, Göteborg	DROP_OFF	Furniture	Tue Feb 26 11:30:00 CET 2013	NOT_STARTED
Skäpplandsgatan 5, Göteborg	NONE		Tue Feb 26 11:30:00 CET 2013	Oxhagsgatan 12, Göteborg	DROP_OFF	Tools	Tue Feb 26 12:00:00 CET 2013	NOT_STARTED

Figure 16: Transport Resource Management HMI with updated itinerary

4. Scenario 3: Transport Infrastructure as a Service

4.1 Design

4.1.1 Prototype definition

Traffic control in-the-cloud prototype is related to Instant Mobility Scenario 3 *Transport infrastructure as a service*, which allows the rapid deployment of a new generation of traffic management systems by exploiting, among others, Future Internet technologies such as cloud data storage, cloud computing and virtualisation. The main objective of Scenario 3 is to carry out a study of the conditions needed for dynamic traffic & urban space management, on how to use Future Internet technologies such as cloud data storage, cloud computing virtualization or services-in-the-cloud. By exploiting the enablers provided by the FI-PPP programme, this scenario will allow the rapid deployment of a new generation of traffic management services. These will result in an improvement in the levels of mobility on the roads by acting as B2B services, for instance by providing accurate RTTI for mobility services such as routing information, personalized route guidance, eco-driving support. The traffic management centres will be able to create the most suitable strategic action plans only by using available high quality data. The actuation services will run optimal policies which take into account the specific requirements of each zone, region or city.

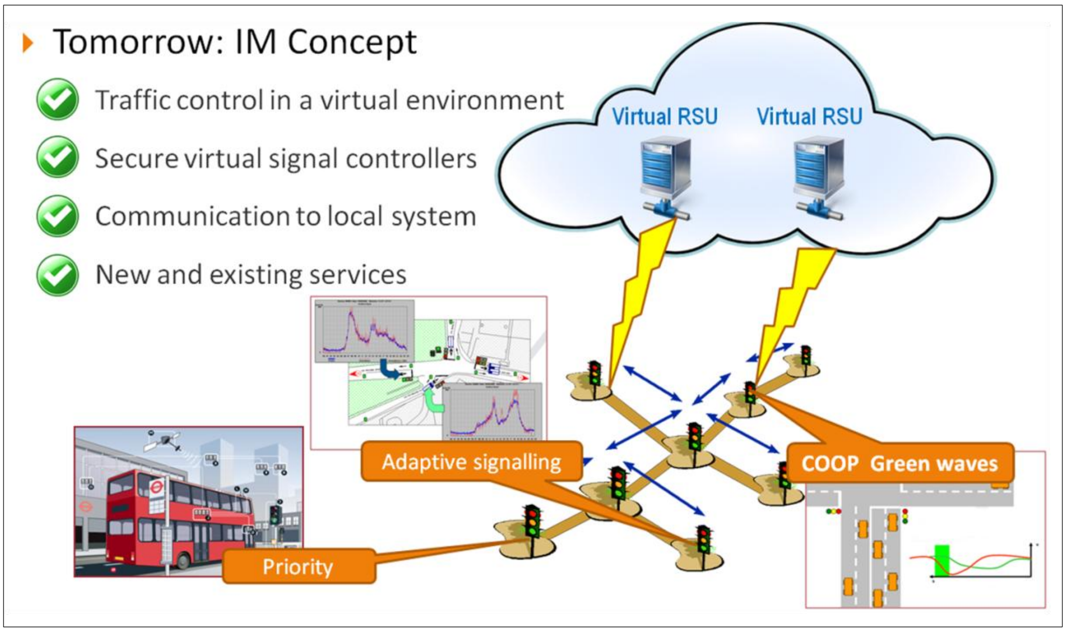


Figure 17. Traffic control in-the-cloud concept

In order to achieve efficient urban traffic management, cities need to deploy not only the technologies for traffic monitoring, but also dedicated traffic management platforms, which can integrate all the data coming from the different monitoring technologies so as to calculate and provide meaningful real time information and strategies either for their own purpose as operators or for end users. All this has an enormous cost for cities. Looking at it from a intersection control engineering, *Traffic control in-the-cloud* is different from the existing approaches as traffic control operations will be hosted in the Internet, leaving local systems the task of providing safety control and communications. At the intersections absolute or selective prioritization policies can be applied, e.g. to give priority to special vehicles such as buses, trams, emergency vehicles, commercial vehicles (HGV), etc. Furthermore, the system is able to automatically detect critical traffic conditions and to activate dynamic green waves, to always keep the network under equilibrium conditions.

4.1.2 Architecture

The architecture of Instant Mobility Scenario 3 and related Enablers are described in detail in WP4 Deliverable D4.14 “Traffic management enablers specification – iteration 2”. In this context the architecture represents to show how the conceptual prototype of scenario 3 represented by different V-RSU is configured and interfaced with TSS Aimsun3, a traffic microscopic simulator.
According to the *Traffic control in-the-cloud* concept, virtual road side units (V-RSUs) are hosted in the cloud, in comfortable server farms. These can apply different control strategies to control different systems and/or different sub-areas of the same system. Different control strategies can be applied according to local needs, availability of traffic measures and design options. Mainly, the system is fully adaptive on traffic and applies dynamic optimisation concepts. The control strategies applied to the network are the result of an optimisation problem aiming to minimise the overall time spent in the network by private traffic and public transport. Real time optimisation and monitoring ensure an immediate reaction on traffic events. The overall network optimisation is decomposed into co-ordinated junction problems solved by the virtual intersection units. As will be described later, the units are hosted in-the-cloud by means of FI-WARE PaaS Management GE, so they can easily and efficiently interact continuously to evaluate the effects of the planned control actions on the downstream intersections. Each unit looks-ahead in the optimisation horizon in order to compute its own control strategy in accordance with the upstream intersections.

The prototype is based on 10 V-RSUs (one for each intersection) installed in corresponding cloud VMs. Since it is not possible to run tests on real roads, these have been substituted by a traffic simulator, based on a microscopic approach to accurately represent the traffic conditions in the network. The used microscopic simulator – implemented in TSS Aimsun[3] – is a combined discrete continuous simulator. This means that there are some elements of the system (vehicles, detectors) whose states change continuously over simulated time, which is split into short fixed time intervals called simulation cycles or steps. There are other elements (traffic signals, entrance points) whose states change discretely at specific points in simulation time. In the simulation model the vehicles are generated and shipped through the network. This kind of approach can be distinguished from other types of computer modelling in that it looks at the interaction of individual "units" such as people or vehicles. Each vehicle is treated as an autonomous entity and the interaction of the vehicles is allowed vary depending on stochastic (randomized) parameters. These parameters are intended to represent individual preferences and tendencies. The area of the city involved in the prototype will be comparable with a larger quarter of the city, and 10 intersections are included in the simulator.

Each physical detector required by the V-RSUs has been modelled as one virtual detector. When a vehicle is passing a detector message is generated towards the adaptive signalling which is treating the incoming traffic information to set-up the best signal setting according to the present traffic situation. The chosen strategy is shifted back towards the simulation model which is changing its signal according to the adaptive strategy. It is important to underline that the abovementioned detectors have been used as they represent the most consolidated way to realise communications between traffic simulators and external controllers. According to the vision

[3] http://www.aimsun.com/wp/

proposed by other "Transport Infrastructure as a Service" Scenario applications, these detectors can be replaced from many other kinds of data coming either from the infrastructure or from mobile devices, cooperative vehicles, external providers and so on.

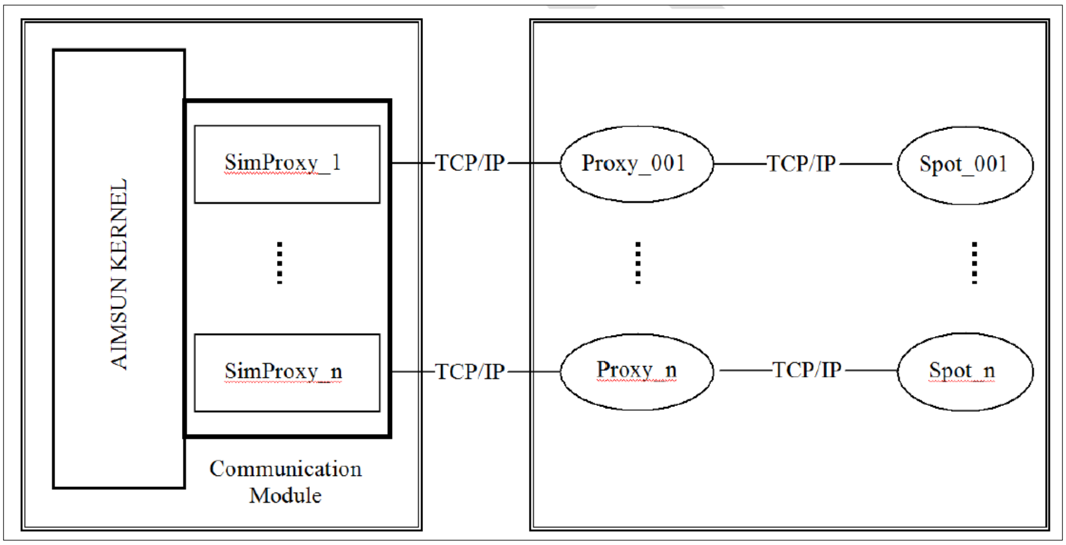


Figure 18. Prototype communication architecture

The communication between the simulator and the cloud V-RSUs is made using sockets (TCP/IP). The simulator module in charge of this is the SimProxy, a module that deals the communication between V-RSUs and the simulator models (implemented in the microsimulator kernel). The architecture implemented, as Figure 18 shows, creates one instance of SimProxy per each V-RSU Unit (intersection control unit) defined in the cloud traffic control centre, and the communication is between each SimProxy unit and their symmetric Proxy unit in V-RSU. Each SimProxy unit acts as a controller, and each connection is identified by a communication port number.

4.1.3 Development

Development activities have been focused on the traffic control functionalities defined in WP4, taking into account the opportunities provided by cloud hosting. Particularly the Enabler named "Intersection virtual controller" that is defined in detail in WP4[4]. This Enabler is in charge of creating the traffic signalling plans by computing the aggregated traffic information that receives from the abovementioned components and systems. This enabler will enable the system to implement V-RSUs, hosted in the cloud, in comfortable server farms. As will be described later in the "Fi-WARE integration" section, the virtualization in this case is achieved by integrating the PaaS Management GE[5].

[4] Deliverable D4.14 "Traffic management enablers specification – iteration 2"
[5] http://forge.fi-ware.eu/plugins/mediawiki/wiki/fiware/index.php/Cloud_Hosting#PaaS_Management

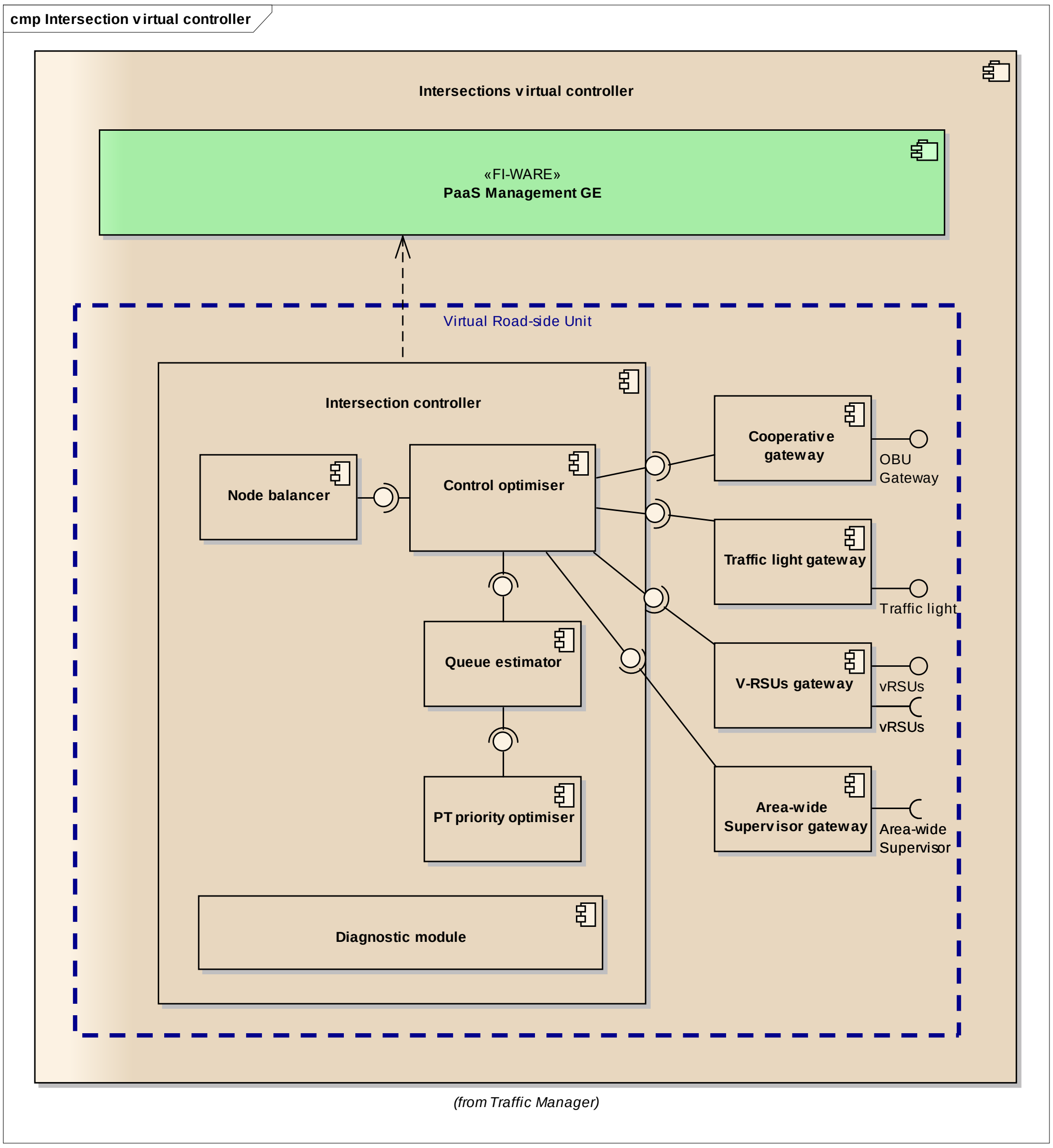


Figure 19. Intersections virtual controller components

This module can apply different control strategies to control different systems and/or different sub-areas of the same system. Different control strategies can be applied according to local needs, availability of traffic measures and design options. As already introduced, the control strategies applied to the network are the result of an optimisation problem aiming to minimise the overall time spent in the network by private traffic and public transport. The optimisation is based on the continuous monitoring of the controlled network. Traffic flows are measured every second and the intersection status is updated every three seconds. Real time optimisation and monitoring ensure for an immediate reaction on traffic demand and events (incidents, priority requests).

Into other words, this component is in charge to real-time calculation and adaptation of traffic lights timing and phases. The detailed status of the proximity network is defined by the status of all the traffic related data of the controlled network that are essential for determining the effectiveness of the control actions and to ensure fast reactions when anomalies arise. The traffic light cycle is updated depending on traffic data such as Traffic Volume, Measurement Sections Occupancy/Speed, Signal Plan Cycle, Automatic Vehicle Classification, Traffic Movement queues and clearance capacity, Turning proportions, Driving Duty Cycle. This Enabler also hosts the mathematical models and the algorithms that are the basis for the control action.

4.1.4 FI-WARE integration

The relevant GE for this prototype are related to the Cloud Hosting chapter[6]. Cloud computing is nowadays a reality. Cloud hosting companies, which can be considered as a particular type of FI-WARE Instance Providers, are already delivering on the promise of the Cloud paradigm. They own and manage large IT infrastructures and offer their use as a service on a pay-as-you-go model. For the success of the Internet-based service economy it is crucial that cloud hosting does not become a market limited to a few strong players, and that future cloud hosting is based on open standards and support interoperability and portability. The FI-WARE project focuses a great part of its efforts on making sure that these standards materialise and in facilitating their adoption by providing open specifications and reference implementations. This standards-based and open approach will cover the fundamental technologies on which the cloud paradigm is based, such a virtualization, as well as new emerging technologies that will differentiate FI-WARE also in terms of the functionality offered.

Virtualisation technologies, such as hypervisors or OS containers, enable partitioning of a physical resource into virtual resources that are functionally equivalent to the physical resource. Moreover, virtualisation creates a very flexible environment in which logical functions are separated from the physical resources. IaaS /PaaS Cloud hosting providers can leverage this capability to further enhance their business. For example live-migration of virtual resources, i.e., the capability of moving the virtual resource from one physical resource to another while the virtual resource remains functional; enable the cloud hosting providers to optimize the resource utilization. However, running different workloads on a shared infrastructure, hosted by a 3rd party, introduces new challenges related to security and trust. FI-WARE will address these challenges by leveraging generic enablers defined in the FI-WARE Security chapter.

PaaS managment[7] is the GE defined in the architecture (WP4). The PaaS Management GE will provide to the users the facility to manage their applications without worrying about the underlying infrastructure of virtual resources (VMs, virtual networks and virtual storage) required for the execution of the application components. This means that the user will only describe the structure of the application, the links among Application Components (ACs) and the requirements for each Application Component. The users will also define the Elasticity Rules based on KPIs that are relevant from an application perspective (i.e. maximum number of DB connection, maximum number of concurrent access to an Web Server, and so on). It is the PaaS Management GE that knows the way in which these application-oriented elasticity rules will be mapped into elasticity rules handled by the IaaS Service Manager GE.

[6] http://forge.fi-ware.eu/plugins/mediawiki/wiki/fiware/index.php/Cloud_Hosting
[7] http://forge.fi-ware.eu/plugins/mediawiki/wiki/fiware/index.php/Main_Page

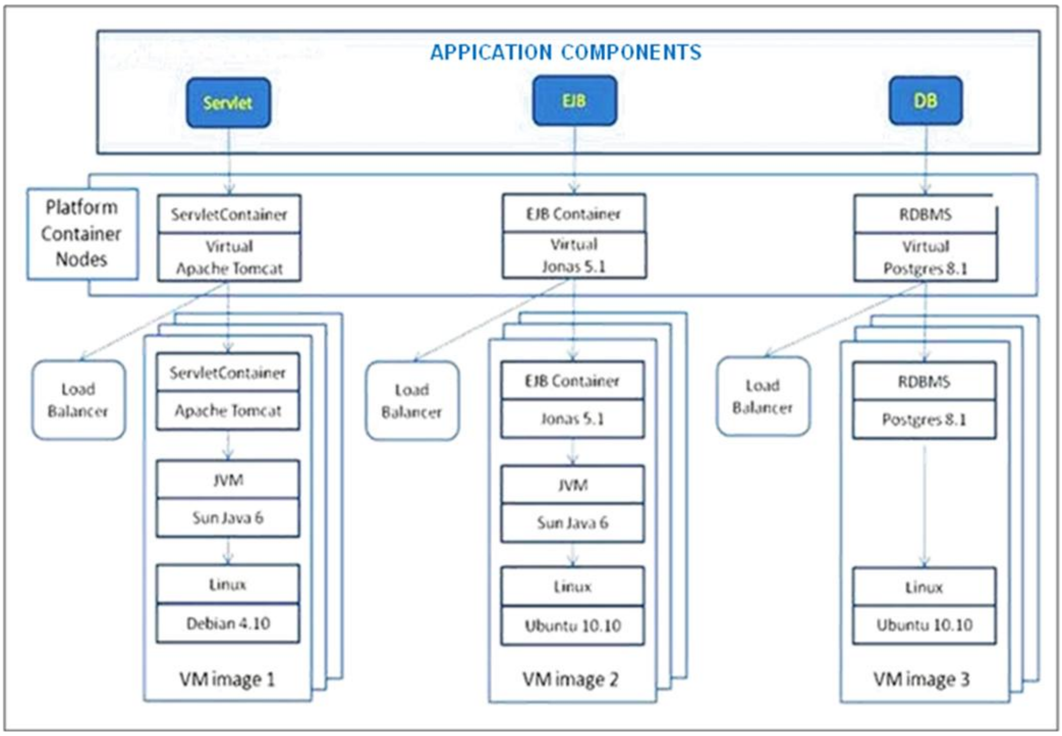


Figure 20. PaaS Management GE

The PaaS Management GE will allow applications to adapt during execution to the changing demands of users or resource shortages. The scalability capability is closely related with the monitoring system to collect and process the different KPIs that could affect the scaling of the application. The scalability can affect application components or platform elements (products implementing the software stack linked to Platform Container Nodes) or complementary FI-WAGE GE services integrated as Cloud Services. This layer should know about the characteristics of the different elements to know how they scale and their limitations.
The PaaS GE is not available in the FI-WARE catalogue for developers who want to test it[8]. The first release of FI-WARE GEs (July/September 2012) does not include any feature related to PaaS GE. The first features of PaaS GE will be released in the next GEs release (Release 2.x), forecasted in the third quarter of 2013. IaaS[9] are the GE developed by FI-WARE. The main difference between IaaS and PaaS is that IaaS manages the structure and lifecycle of the virtual infrastructure required for applications to run. PaaS on the other hand, manages the structure and lifecycle of the applications and platform containers. The user describes the application without considering how such description will map into a final IaaS Service manifest or changes into existing IaaS deployments. PaaS Management GE gives to the user the possibility to define the Elasticity Rules based on KPIs that are relevant from an application perspective. In the case of the V-RSUs, the KPIs will be base on latency, availability and VMs performance, resulting in a robust, scalable and elastic system.

[8] http://forge.fi-ware.eu/plugins/mediawiki/wiki/fiware/index.php/Roadmap_of_Cloud_Hosting
[9] http://catalogue.fi-ware.eu/

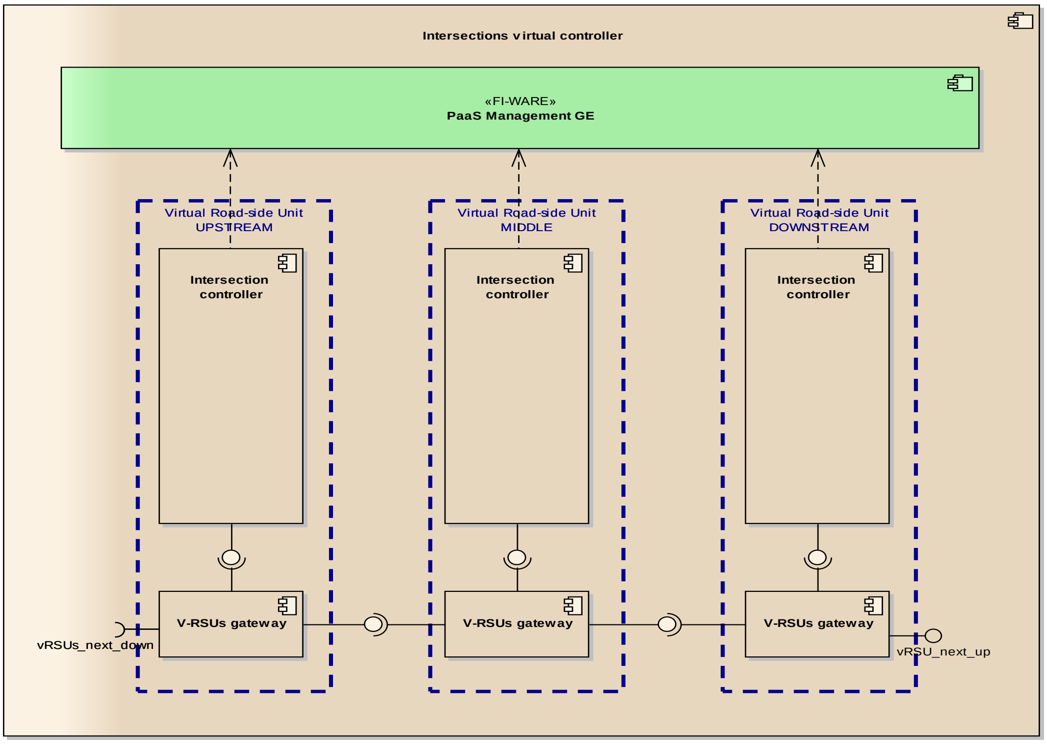


Figure 21. V-RSUs usage of PaaS Management

Further, this GE will allow V-RSUs to adapt during execution to the changing demands of resource shortages. Not only the scale-up (in the same physical host) or scale-out (to multiple VMs) will be supported, also the shrinking of resources will be performed when the environment of the application allows this. There will be an interaction between the scalability components, the provisioning and deployment layer to create, stop, destroy, and reconfigure VMs, infrastructure and/or network resources. This will be used to adapt the resources given to each v-RSU according to the optimisation needed (e.g. growing congestion -> high optimisation; weak vehicular flow -> low optimisation).

4.2 Deployment

The simulator is virtually located in the city of Trondheim (Norway) and is based on a micro approach. Virtual road side units control all the intersections of the model, giving the services listed in previous sections and using real on-field data coming from traditional road sensors and innovative sources and potentially data providers. The area of the city that has been selected for the demonstration is the main corridor crossing the southern part of the city from the motorway to the city centre. This corridor experiences some congestion during the morning that at present are managed with a traditional road-side system. This will allow to evaluate the cloud-based system proposed by Instant Mobility and to compare its performances with the ones of the traditional road-side system.

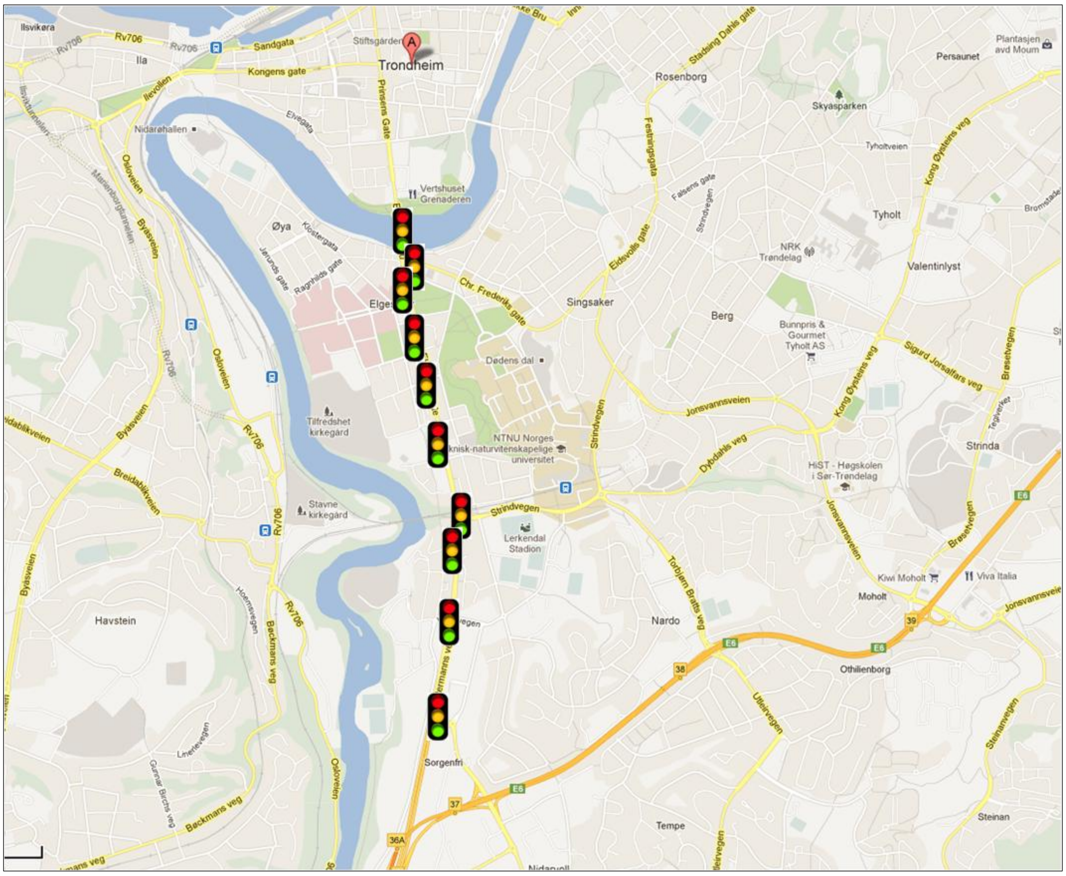


Figure 22. The area of the demonstration in Trondheim

4.2.1 Hardware environment

Since PaaS Management GE was not available in FI-WARE testbeds at the time of the IM scenario 3 prototype development activities, another solution has been developed to deploy V-RSU in the cloud and therefore demonstrate the concept proposed here by Instant Mobility.

This solution (see next Figure) is based on virtual machines (VMs) hosted in-the-cloud, delivered by Swarco Mizar and used here to deploy V-RSUs hosintg platforms. In first approximation, a VM is needed for each of the traffic control zones the city is divided. In the case of this demonstration, only one zone is involved (zone 6). Another VM has been created for the traffic simulator and virtually represents the road of the city of Trondheim. All these VMs are based on MS Windows Server 2008[10]. For the purpose of the demonstration, all the VMs have been accessed by remote desktop. Web-based GUI is available as well.

[10] http://www.microsoft.com/it-it/server-cloud/windows-server/default.aspx

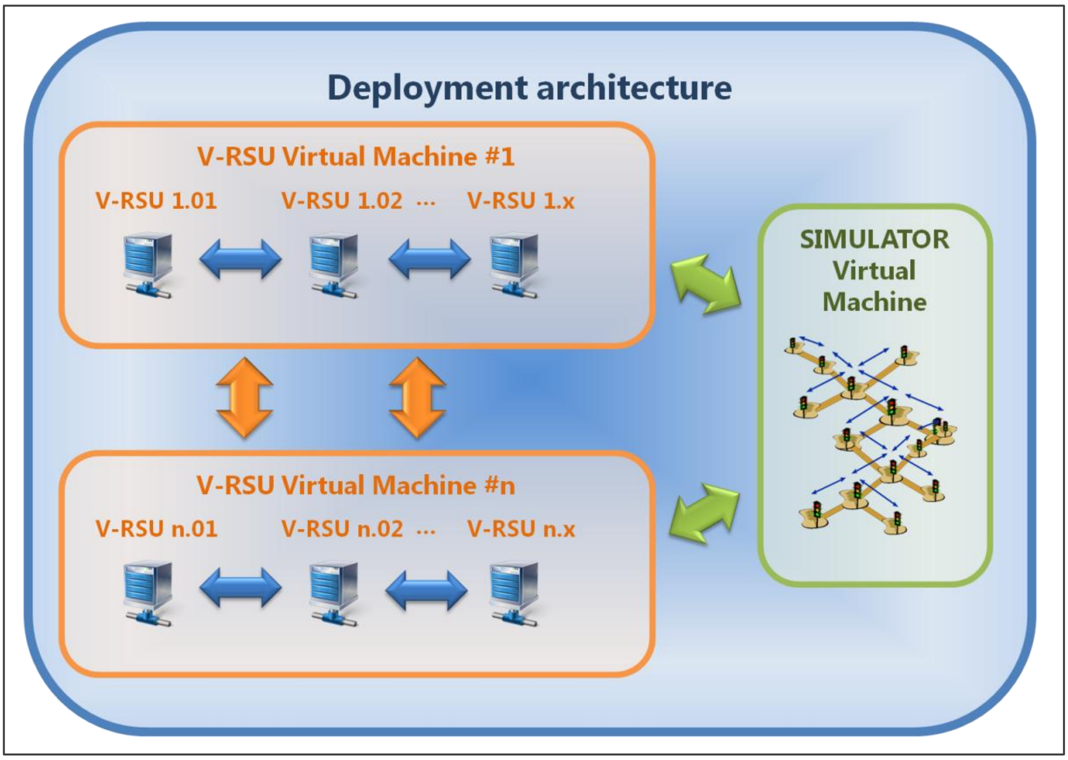


Figure 23. "Traffic control in-the-cloud" prototype deployment architecture

4.2.2 Implementation

After testing the cloud-based architecture (see section 4.3), to be demonstrated in congresses, this prototype has been implemented in a different way, making use of physical machines but keeping the architecture unmodified. This has been necessary because in most of the events where the Scenario 3 demonstrators has been shown, many of the requirements of the prototype were not met (e.g. quite often Public IP address was not available or it was not possible to connect it in a time fashion).

In other words, a local implementation of the prototype has been installed, keeping communications at a local level in order to satisfy all the requirements needed and to speed up the configuration of the demonstrator (essential when many demonstrators are involved – e.g. in the MWC 2013 FI-PPP booth in Barcelona), but without changing the basic concept of *traffic control in-the-cloud* defined in the previous sections.

Figure 24. FI-PPP Event in Barcelona (March 2013) - “Traffic control in-the-cloud” demonstrator

4.3 Testing

In this project the main objective of scenario 3 is to move the RSU from its current state which is physically located in the street to a virtualized environment. How this can be done is described in the above chapters in detail. However, in this project we went a bit further also in testing the virtualized RSU, if it gives similar results like those RSU which are physically located in the streets (traditional constellation). In here as it described above, since this is not the main objective of scenario 3, we are not going to go in detail to explain all the methodologies needed to make the simulation activity used for testing purpose. Instead a brief description will be given in the following sub chapter.

4.3.1 Testing configuration

As already introduced in the city corridor of Trondheim 10 cloud intersection controllers (called SPOT) as seen in the figure below has been selected and placed in a virtual environment (see Section 4.2.1) as a trial demo for this purpose. A microscopic simulation model is used together with the virtual adaptive signalling model to simulate the traffic flow and signal control that will occur when combining both a distributed and a virtually decentralised model by using cloud computing capabilities. All the V-RSUs have been configured according to the parameters (signal calls, groups weight, movements conflicts management, …) used in the real on-field installation, in order to properly evaluate the performances of the cloud based system. The V-RSUs have been configured to work in centralised mode. This means that each intersection is optimised taking into account also the status and the traffic demand at the upstream and downstream intersections. In this way the whole network is then optimised and the capacity of the traffic corridor can be significantly increased.

Such configuration is therefore detected by the traffic simulator (see next Figure) that waits for information on how to control intersections from the V-RSU connected to the simulator itself. In case of communication problems, the traffic simulator has been configured to automatically switch to a fixed reference traffic control plan and continue the simulation, coherently to what happens in real world when physical road-side units lose connection with the traffic management centre.

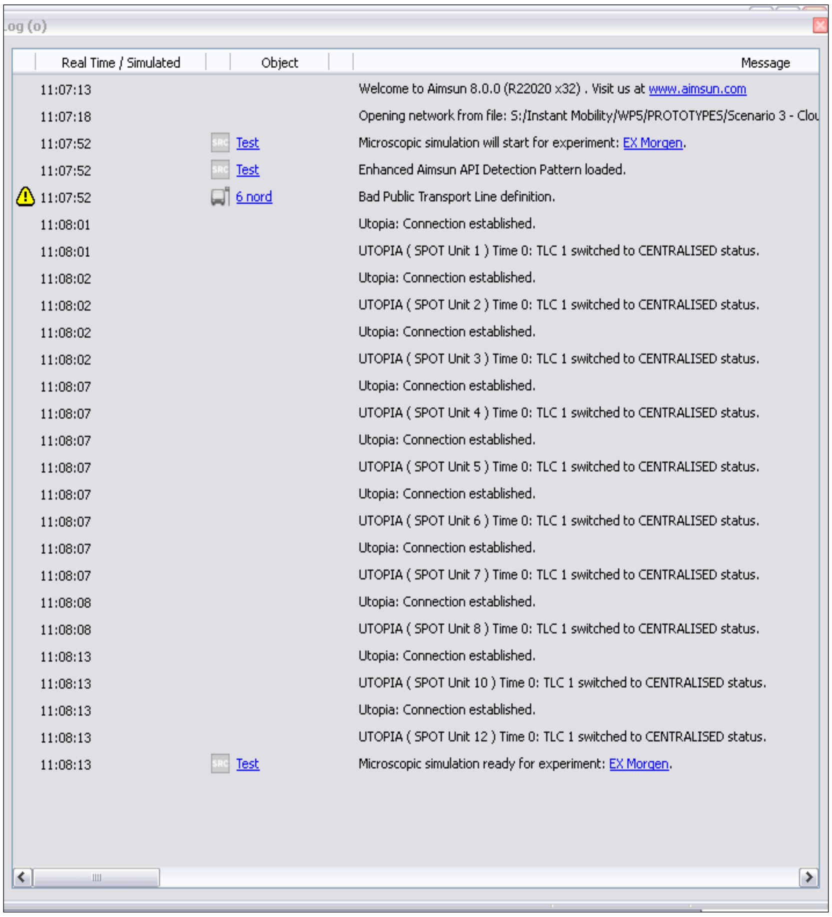


Figure 25. Log file that show the connection of V-RSU to the traffic simulator

To complete the configuration it has been necessary to set area parameters in order to have corridor-wide services like green waves and PT priority. In the following figure it is possible to see three V-RSU (607, 612 and 610 – not visible in the figure) operating in coordinated way in the northern part of the model and giving green wave to travelling vehicles.

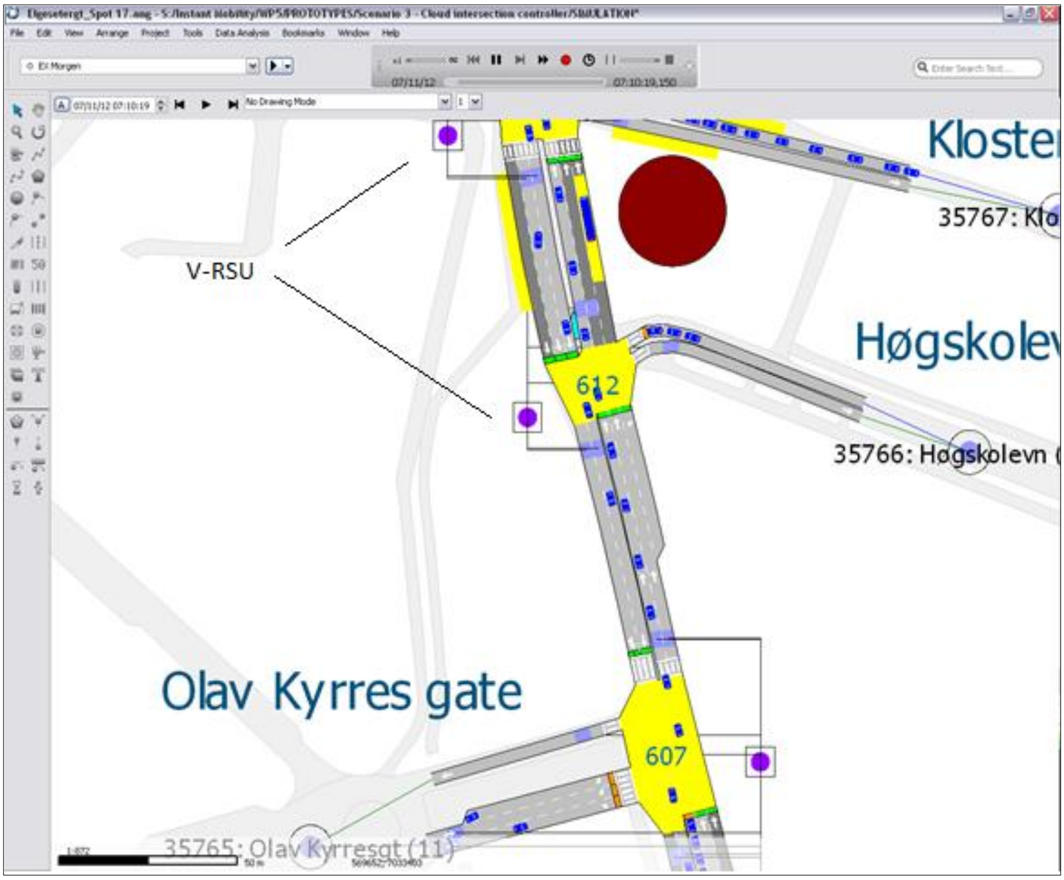


Figure 26. Micro-simulation model – Virtual controller and detectors

Last but not least, the V-RSUs have been configured to report to the operator through a simple GUI the status of each intersection. This kind of interface with the current road-side technology is not available for operators working in the centre, and has to be accessed locally. In the next figure it is possible to see the control panel of the virtual traffic management centre (background window) showing the connection to the 10 V-RSU during the simulation. As a proof of concept of the operator GUI three (607, 606, 604) V-RSUs interfaces have been reported in the screen and the table about stages and timing is shown. In the table the active signal group with its timing is reported, in addition to special calls stages (e.g. for on-demand turning or PT priority – for instance V-RSU 604 is going to give priority on a demand signal group to a PT vehicle turning in signal groups 5 and 6).

Since signal phases and timing are available at the centre, further services based on vehicle-infrastructure cooperation could be enabled, such as cooperative signalling, speed advice and vehicle cooperative start&stop for vehicles queuing at the controlled intersections. The definition of these additional services is out of the scope of Instant Mobility, but mentioning is important in order to underline the potentialities given by *traffic control in-the-cloud* approach demonstrated with this prototype.

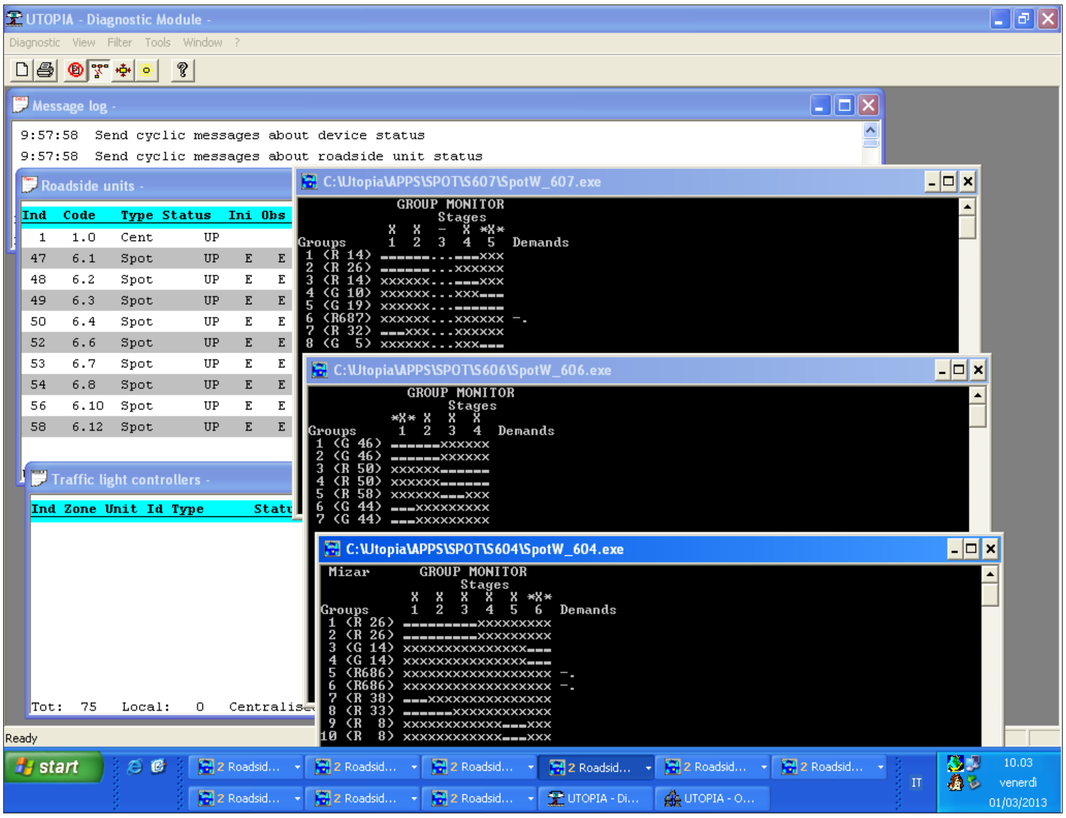


Figure 27. V-RSU interfaces showing control stages and signal groups

4.3.2 Testing results

The aim of the simulation is to test and evaluate if SPOT/UTOPIA located in a virtual environment "Traffic control in the cloud" is able to give enhanced traffic-adaptive control services like fully adaptive signalling, green waves management, and priority to public transport. This prototype represents the first step towards a more efficient utilisation of adaptive signalling looking at cost efficient solutions for installing the UTC in new areas within the cities.
For the testing purpose 10 signalized junctions and pedestrian crossing with adaptive traffic signal control in the centre part of the city was considered. The effects are primarily calculated based on quantitative analysis with simulations in TSS Aimsun. An important prerequisite for comparing different control strategies is to have a controlled traffic. By using micro-simulation, evaluation carried out much more quickly and at significantly lower costs. Mobility groups involved in the evaluation are private cars and public transport vehicles, while the evaluation is done for the peak morning traffic conditions (from 7.30am to 8.30am), which represent the most difficult (and at the same time interesting) scenario for traffic control issues. In large-scale installations, traditional road-side SPOT controllers give significant benefits to urban traffic flows in terms of travel time reduction, up to 15% for private cars and up to 28% for PT vehicles with activated PT priority functionalities. The activation of this service does not affect travel time of private vehicles, which increases of less than 1%. Such reduction in travel time originates relevant side effect as regards the reduction of fuel consumptions and greenhouse gases emissions.

The results of the simulated prototype in Trondheim confirmed the equivalent performances of the virtual intersection controller, when compared with SPOT (intersection intelligence) located locally as the today's state-of-the art (traditional road-side unit). Even if this kind of evaluation is out of the scope of the activities described in this document, the results of several simulations have been analysed and averaged in order to estimate the benefits related to the implementation

of the above mentioned virtual RSU in a traffic corridor in Trondheim. This kind of evaluation (see also Figure below) shows that, even if the virtual RSU are deployed only in a relatively small part of the city, improvements in traffic conditions are significant.
To support this statement, the time series of mean queue length and average spatial density in the considered corridor are reported in the next chart. In order to give to the virtual RSU the required time to self-learn and adapt traffic control, a warm up period of 90 minutes has been used (left part of the chart - of course in real life this activity is needed only at system installation). Then, feeding the simulator with real traffic data, two scenarios have been identified and tested for the morning peak time window (extended from 6am to 10am):

1. **ITS Scenario**: implementation of the 10 virtual RSU along the corridor with coordinated adaptive signalling functionalities enabled;
2. **Baseline Scenario**: a reference fixed plan (normally used as a backup solution in case of system failure in the real city of Trondheim) is applied for all the duration of the simulation.

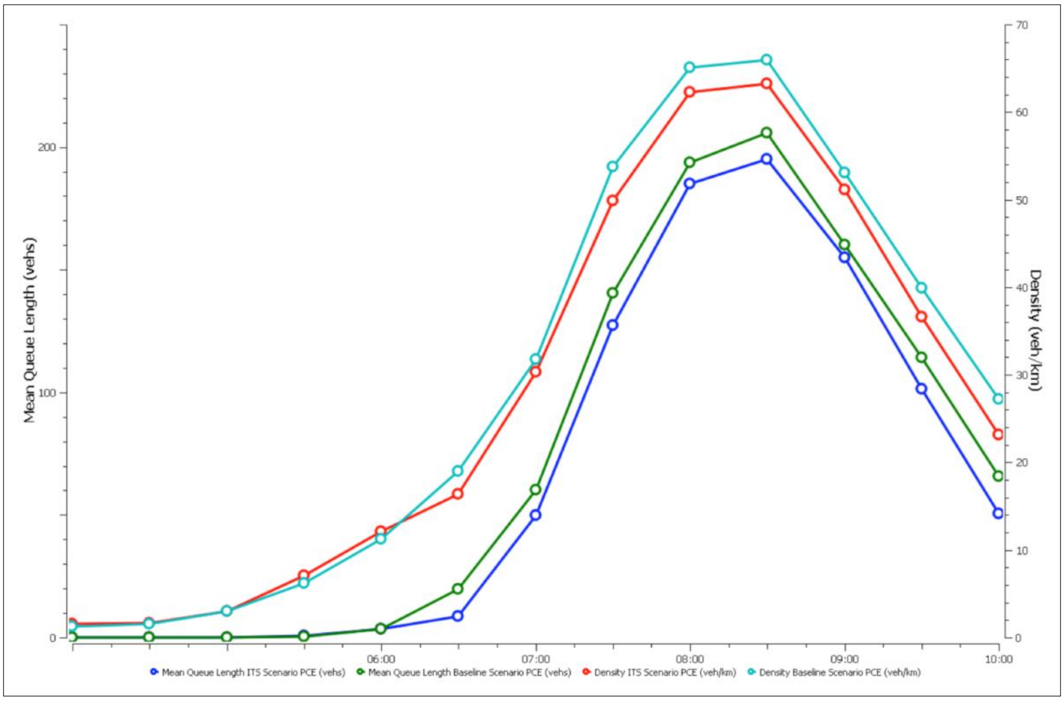


Figure 28. V-RSU assessment – Average queue length and vehicle density

Looking at the four time series, it is possible to see the improvements (comprised between 5% and 10%) both for the mean length of the queue, which is one of the main parameters used to measure drivers' discomfort while travelling, and for the average density, which is one of the most direct measurement of the quality of traffic flow and of, in first approximation, fuel consumptions and CO_2 emissions. Such improvements are more evident in the peak of the traffic demand, around 8am, and in the discharging part of the graph, where the adaptive signalling operated by the virtual RSU is much more efficient in draining vehicles.

5. FI-Ware Generic Enablers in Smart City Logistics

This specific development was planned after specific recommendation to include more Generic Enablers into Instant Mobility prototypes.

This prototype aims at implementing the functionality of the use cases of “Smart City Logistics” as it was described in deliverable “D5.1 Prototype Description” with a demonstration that soem Generic Enablers could support the whole goods transport chain including the end-customer, starting with load sensing and pick up planning until dynamic drop point.

Example of use-case to be aligned as much as possible with the Smart City Logistics scenario including HMI:

- New goods delivery (from end customer order to delivery)

The transportation company receives a pick up request in the back office system and accepts the assignment.

- Message for the van driver to be aware of the new location to deliver parcel and choose where to pick up the goods (smartphone van driver)
- Based on pick-up and delivery places, new journey and load sharing, and external events which are not directly managed by the transport company, choose vehicle
- Based on the current fleet load and location, dynamic allocation of which vehicle is optimal for picking up and delivering the goods.

To be relevant, it was assumed that the whole process would be managed along a unique day (order in the morning, delivery in the same day) and because Generic Enablers do not include any specific tools for HMI, to provide a simple interface showing the whole scenario instead of focusing on how interactions would happen between humans involved in this scenario (end-user customer, consignee, van driver) and some relevant devices.

5.1 Scenario hypothesys

Hyp 1 : Smartphones as communication devices

As it was described in many Instant Mobility scenarios, floating data are of high value to create innovative services. To collect all the relevant floating data in this scenario we assume that smartphones would be the main smart devices all actors could use:

- This hypothesys is fully compliant with the Open Mobile Framework (Mirrorlink) demonstrated by Valeo, using communication facilities of a smartphone instead of using another gateway in a vehicle to communicate;
- This hypothesys is also fully compliant with the interfaces developed in the other Smart City Logistics scenario with HMI on tablets for van drivers.

Hyp 2 : People, vehicles, parcels are smart things

The major issue is to be able to track and trace a specific parcel, ordered in the morning by the end-user consumer and to monitor it in real-time when it is included in the delivery process. While it could be interesting to include internal process inside warehouses, we focus mainly on urban delivery monitoring, from a central warehouse to customer home and customer office as new delivery address.

How smart people are? We consider that a typical apps provided by the webstore when you finalize your order, or by the logistics company when the consumer receives his parcel track ID would be install on smartphone providing smart features linked to geolocation. A A-GPS smartphone si the best device to transform a person in a connected thing in our scenario.

How smart the vehicle is? Based on the demonstration realized in ITS Vienna with a mirrorlink system, the delivery van could be the smarter thing especially coupling GPS features with communication facilities of the van driver smartphone.

How smart parcels are? A parcel could be referenced by many kinds of tag (barcode, 2D, RFID) but currently the parcels with these tags is not smart but could be track in an asynchronous mode, point by point when a tag reader is used. When moving outside any warehouse (van, parcel, tag) is a perfect triplet to have a smart parcel, fleet management of the delivery van should provide real-time location of this smart parcel.

5.2 Identification of relevant Generic Enablers from FI-Ware project.

Based on FI-Ware release 1 available on the testbed facilities, the following Generic Enablers should be relevant.

Internet of Things chapter:

- Gateway Protocol Adapter
  It could be relevant if a Protocol Adapter related to traceability standards as EPC Global have been available, connected any RFID reader to the other Generic Enabler. As this specific instance is not available, EPC Global events have been simulated

- Gateway Device Management
  This enabler has a specific role if you have many things to manage behind a gateway. It should be included typically in the van driver smartphone or tablet to manage all connected things we could found in the van.

- Gateway Data Handling
  This enabler would collect events at the gateway level based on the relevant protocols adapters which are sued by the gateway. In our scenario, two Protocol Adpaters should be required, EPC Global (parcel information) and CAN Bus (integrated GPS for fleet management). Using a smartphone, we consider that the RFID reader as well as the A-GPS are managed by the smartphone OS or apps and that hese events could be collected. Rules can be applied on a collection of events to decide some actions.

- Backend Things Management
  This Generic enabler integrates two different components, first managing a specific flow of information using the OMA-NGSI 9&10 standard (component IoT Broker), second managing the object which would provide relevant data (Configuration Management component. We understood that this Configuration Management component will be able to manage the triplet (van, parcel, tag) as the object which would provide events.

Data and Context Management chapter

- PubSub Broker
  This Generic Enabler could manage the collection of events, collecting them if they happen, or requesting update of data if the application level needs them. This PubSub broker should be able to manage the message coming from the end-user consumer, the van or the warehouse/consignee in a dynamic way depending of their situation along the scenario:
  Consumer at home, consumer moving to his ofiice, consumer at his office, parcel in the warehouse, in a van, in a new drop point, in another van, as well as external events from the Smart city (traffic jam, pollution alerts…)

- CEP
  This enabler should be useful to improve the whole monitoring of the scenario, especially using statistical data on previous days (level of van shipment, potential traffic jam on the same timeslots in a day…). Because we have no historical or statistical data, we do not plan to integrate this Generic Enabler in our scenario.

Based on the new release which is planned by FI-Ware, the two components of Backend Things Management GE would become Generic Enablers: Backend IoT Broker GE and Configuration Management GE.

Two different instances of the PubSub broker GE are available on FI-Ware Catalogue. We have used the PubSub broker instance from Telefonica to develop our prototype (PubSub Broker SAMSON).
On the following picture, in red circles the Generic Enablers used by the prototype, in dashed-line red circles, potential other GE.

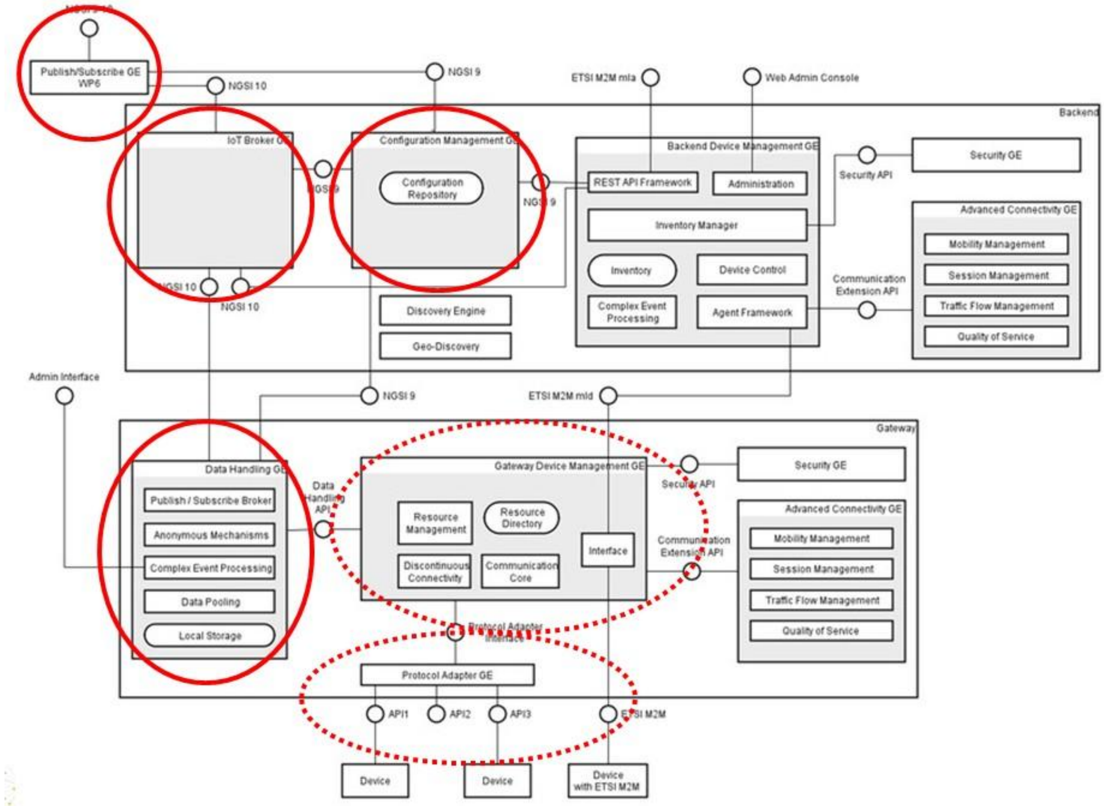


Figure 29: useful Generic Enablers from FI-Ware for Smart City Logistics scenario

5.3 Parcel delivery process using FI-Ware Generic Enablers

To implement the full scenario taking into account that several smart things are involved, the initial view is the following:

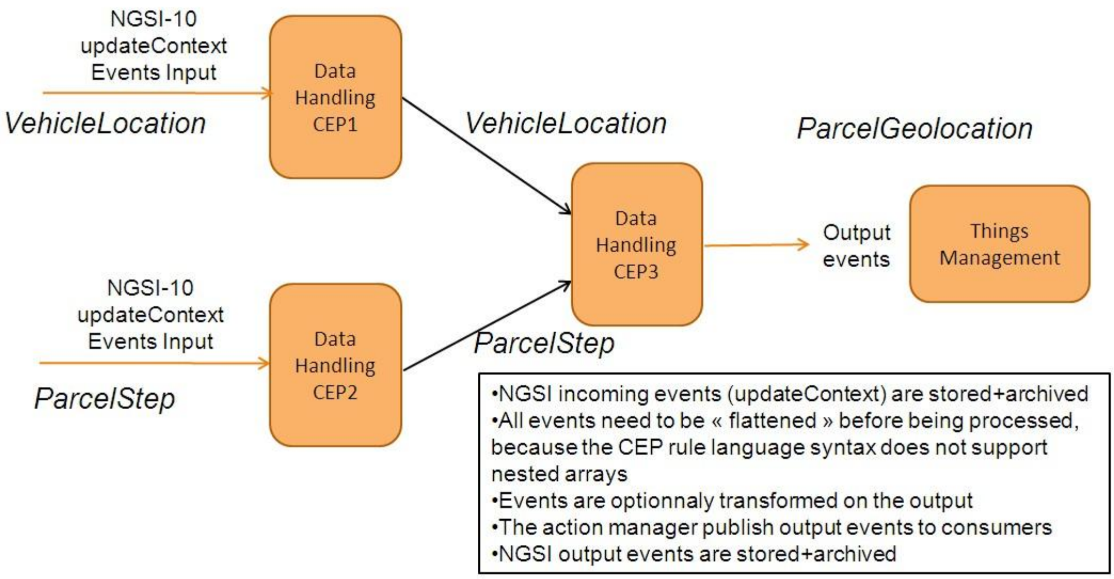


Figure 30: Parcel delivery OMA-NGSI process

Two main query processing strategies are supported:

1. Local processing: when registered and participating retrieval systems are able to process the whole query locally.
   Each system can provide their local metadata format and local/autonomous data-set.
   A query transmitted to such systems is understood as a whole and the items of the result set are the outcome of an execution of the query.
   In case of differing metadata formats in the back ends a transformation of the metadata format may be needed before the (sub)query is transmitted. In addition, depending on the degree of overlap among the data sets, the individual result sets may contain duplicates. However, a result aggregation process only needs to perform an overall ranking of the result items of the involved retrieval systems. Here, duplication elimination algorithms may be applied as well.
2. Distributed processing: registered and participating retrieval systems that allow distributed processing on the basis of a global data set.
   In this context, it behaves like a federated Database Management System.
   The involved heterogeneous systems may depend on different data representation (e.g., ontology based semantic annotations and XML-based feature values) and query interfaces (e.g., SPARQL and XQuery) but describe a common (linked) global data set.
   Each query needs to be evaluated and optimized, which results into a specific query execution plan.
   In series, segments of the query are forwarded to the respective engines and executed.
   Now, the result aggregation has to deal with a correct consolidation and (if required) format conversion of the partial result sets.

Because we cannot use a specific gateway for the parcel itself (RFID tags do not have the computational and storage resources for it), we implemented an external context partially aligned with the eco-driving use-case of the other prototype: pollution management in the smart city.

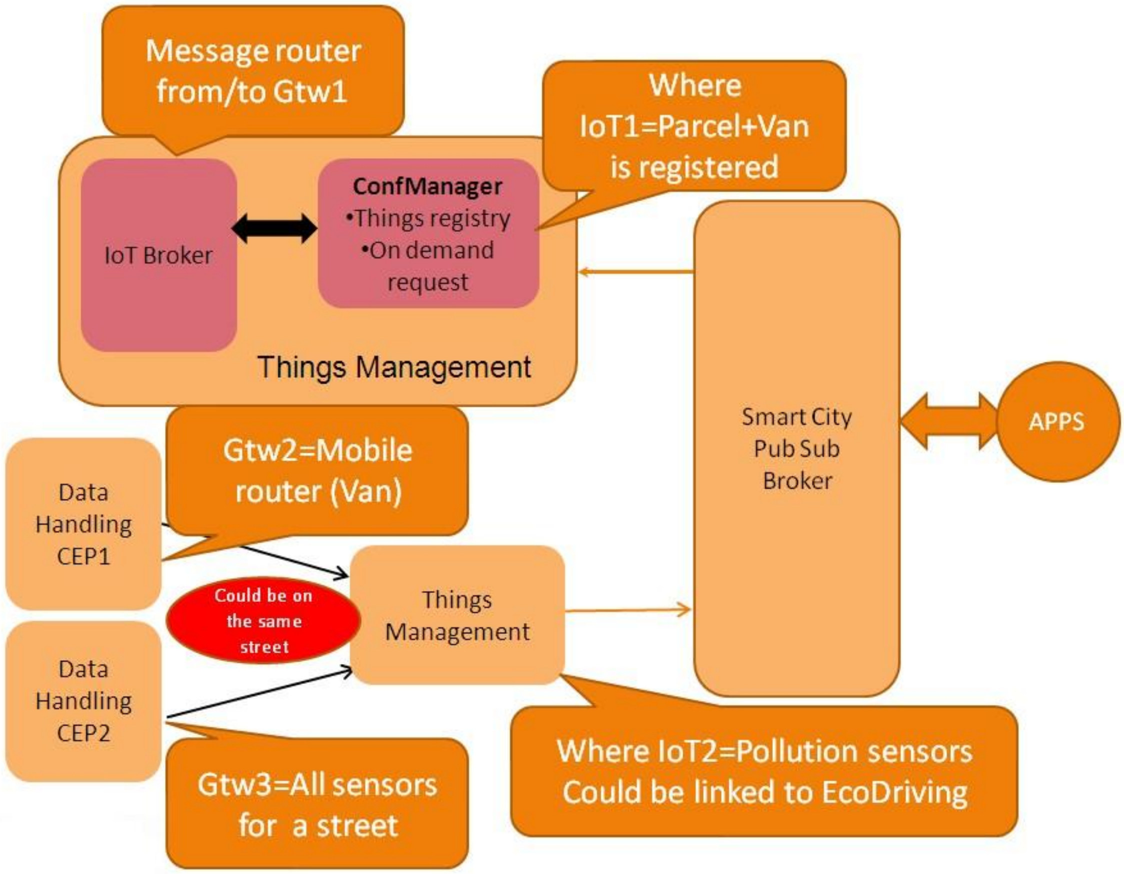


Figure 31: concrete implementation with FI-Ware GEs

To manage this implementation, two different instances of the Gateway Data Handling Generic Enablers were used: CEP Mobile Manager (Android compliant) to be hosted in a smartphone or a tablet, and Esper4FastData Servlet instance, running in a computer with Java.

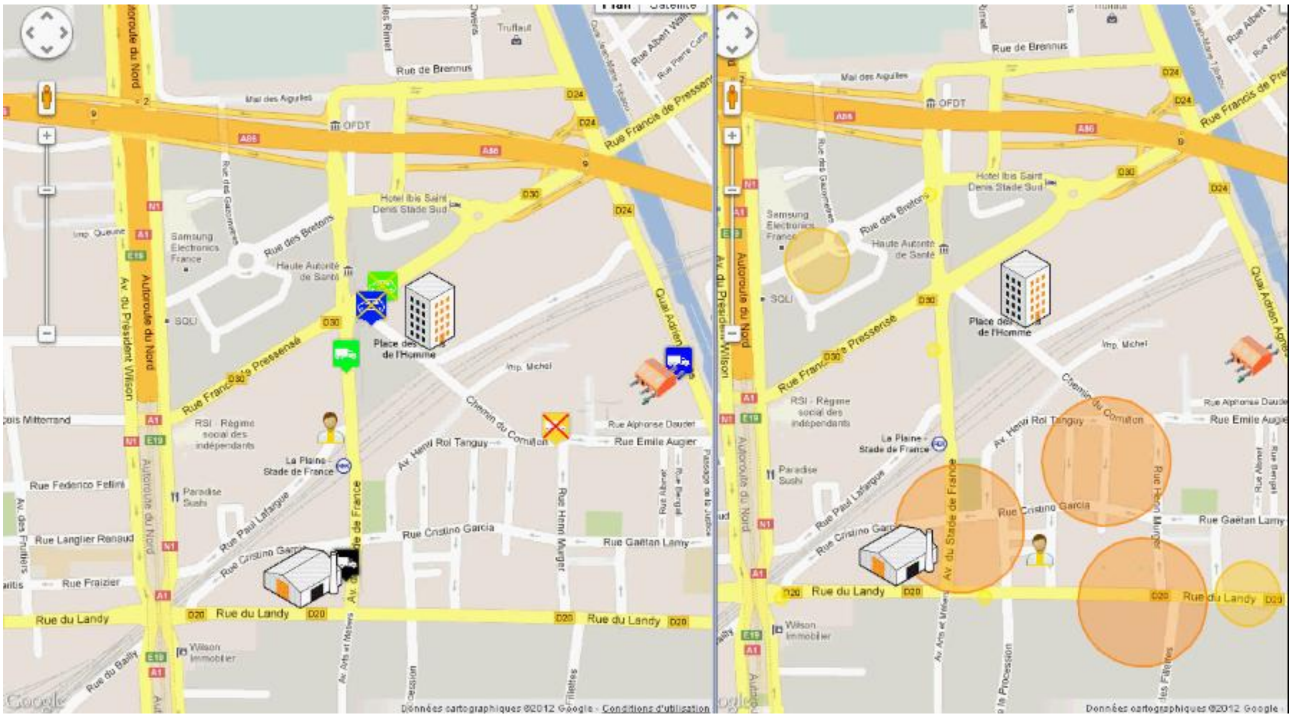
To provide a better understanding of the prototype, a specific interface was used (SIAFU), this simulator is easy to use to show the result of the scenario and takes as input events from Gateway Data Handling (no OMA NGSI compliance).

Logistics console

To show van loading parcel in the warehouse
Consumer going from home to office
Incoming alert pollution: only electric vehicles are authorized to deliver parcels
Vans go to the drop point to transfer parcels to electric vehicles

Smart City Console

Monitoring of pollution level in different places
At one time in the day, alert pollution happens and is sent to all actors and especially logistics companies
- ⇨ Electric vehicles only can circulate



5.4 Parcel scenario and OMA-NGSI implementation

The parcel arrives by Plane at the airport, is loaded in a first van, then is transferred in a second van after some transport, and is then delivered to the customer by this second van.

The parcel is associated in real-time to the current mean of transport (Plane or van). A mean of transport has an identifier which can be: Plane, Van1, Van2.
All these vehicles send their own geolocation events.
As explain before, the combination of van events and parcel events allows parcel geolocation. This combination is achieved by setting a combination CEP rule in the Gateway Data Handling GE.

Three Gateway Data Handling GE instances are involved. The first one provides vans location, the second one provides parcel state, and the third one combines parcel state with vans location to allow parcel geolocation.

It is possible to call the OMA-NGSI queryContext method of any Gateway Data Handling GE instance, at any time.

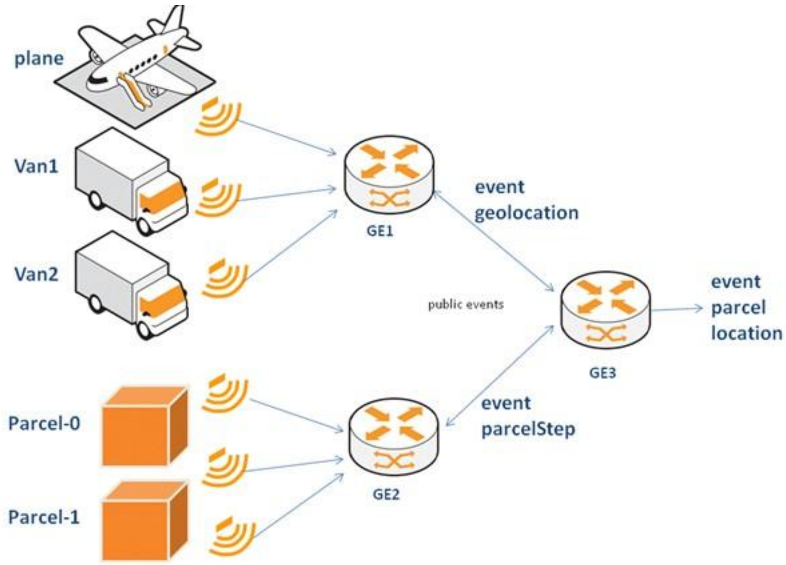


Figure 32: scenario overview with OMA-NGSI events

5.5 Description of the OMA-NGSI events used for parcels and vehicles :

ParcelStep events - Events description

Attributes:

- entity id= parcel-0
- vehicle id= Plane,Van1, Van2
- deliveryStep= Ordered , DeliveryInProgress, Delivered
- timeStamp

Event file	vehicle	Delivery Step	timeStamp	Date
contextElementResponseListParcelStep0.xml	Plane	Ordered	1352191090	06/11/2012 09:38:10
contextElementResponseListParcelStep1.xml	Plane	DeliveryInProgress	1352199104	06/11/2012 11:51:44
contextElementResponseListParcelStep2.xml	Van1	DeliveryInProgress	1352207169	06/11/2012 14:06:09
contextElementResponseListParcelStep3.xml	Van2	DeliveryInProgress	1352215801	06/11/2012 16:30:01
contextElementResponseListParcelStep4.xml	Van2	Delivered	1352226632	06/11/2012 19:30:32

Vehicles location events - Events description

Attributes:

- entity id= Plane,Van1, Van2
- latitude
- longitude
- timeStamp

Event File	Vehicle	latitude	longitude	timeStamp	⇔Date
contextElementResponseListVehicleLocation0.xml	Plane	43.660266	7.205032	1352199104	06/11/2012 11:51:44
contextElementResponseListVehicleLocation1.xml	Van1	43.660266	7.205032	1352200525	06/11/2012 12:15:25
contextElementResponseListVehicleLocation2.xml	Van1	43.665512	7.20353	1352207169	06/11/2012 14:06:09
contextElementResponseListVehicleLocation3.xml	Van1	43.669812	7.203487	1352211315	06/11/2012 15:15:15
contextElementResponseListVehicleLocation4.xml	Van2	43.669812	7.203487	1352215801	06/11/2012 16:30:01
contextElementResponseListVehicleLocation5.xml	Van2	43.69841	7.188091	1352218255	06/11/2012 17:10:55
contextElementResponseListVehicleLocation6.xml	Van2	43.718016	7.186718	1352222660	06/11/2012 18:24:20
contextElementResponseListVehicleLocation7.xml	Van2	43.734391	7.184658	1352226632	06/11/2012 19:30:32

Example of quetyContext requests :

queryContext request for Van1 entity

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<queryContextRequest>
  <entityIdList>
  <entityId>
    <name>Van1</name>
    <type>http://www.ict-sensei.org/EntityOfInterest#vehicle</type>
    <isPattern>false</isPattern>
    </entityId>
  </entityIdList>
</queryContextRequest>
```

queryContext request for parcel entity

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<queryContextRequest>
  <entityIdList>
  <entityId>
    <name>parcel-0</name>
    <type>http://www.ict-sensei.org/EntityOfInterest#parcel</type>
    <isPattern>false</isPattern>
     </entityId>
  </entityIdList>
</queryContextRequest>
```

queryContext response samples for parcel-0 entity

contextElementResponseListParcelStep0.xml


```xml
<?xml version="1.0" encoding="iso-8859-1" standalone="no"?>
<contextElementResponseList>
  <contextElementResponse>
      <contextElement>
        <attributeDomainName>parcelStep</attributeDomainName>
        <entityId>
          <name>parcel-0</name>
        </entityId>
        <contextAttributeList>
          <contextAttribute>
              <name>vehicleId</name>
              <type>string</type>
              <contextValue>Plane</contextValue>
          </contextAttribute>
          <contextAttribute>
              <name>deliveryStep</name>
              <type>string</type>
              <contextValue>Ordered</contextValue>
          </contextAttribute>
        </contextAttributeList>
        <domainMetadata>
          <contextMetadata>
              <name>timestamp</name>
              <type>xsd:dataTime</type>
              <value>1352191090</value>
          </contextMetadata>
        </domainMetadata>
      </contextElement>
    <statusCode>                <code>0</code>
<reasonPhrase>OK</reasonPhrase>
      </statusCode>
  </contextElementResponse>
</contextElementResponseList>
```


contextElementResponseListParcelStep1.xml


```xml
<?xml version="1.0" encoding="iso-8859-1" standalone="no"?>
<contextElementResponseList>
  <contextElementResponse>
      <contextElement>
        <attributeDomainName>parcelStep</attributeDomainName>
        <entityId>
<name>parcel-0</name>
        </entityId>
        <contextAttributeList>
          <contextAttribute>
              <name>vehicleId</name>
              <type>string</type>
              <contextValue>Plane</contextValue>
          </contextAttribute>
          <contextAttribute>
              <name>deliveryStep</name>
              <type>string</type>
              <contextValue>DeliveryInProgress</contextValue>
```

```
</contextAttribute>
</contextAttributeList>
<domainMetadata>
<contextMetadata>
<name>timestamp</name>
<type>xsd:dataTime</type>
<value>1352199104</value>
</contextMetadata>
</domainMetadata>
</contextElement>
<statusCode>                    <code>0</code>
<reasonPhrase>OK</reasonPhrase>
</statusCode>
</contextElementResponse>
</contextElementResponseList>
```


queryContext response samples for vehicle entities

geolocation for Van1


```
<?xml version="1.0" encoding="iso-8859-1" standalone="no"?>
<contextElementResponseList>
  <contextElementResponse>
    <contextElement>
      <attributeDomainName>geolocation</attributeDomainName>
      <entityId>
        <name>Van1</name>
      </entityId>
      <contextAttributeList>
        <contextAttribute>
          <name>latitude</name>
          <type>double</type>
          <contextValue>43.660266</contextValue>
        </contextAttribute>
        <contextAttribute>
          <name>longitude</name>
          <type>double</type>
          <contextValue>7.205032</contextValue>
        </contextAttribute>
      </contextAttributeList>
      <domainMetadata>
        <contextMetadata>
          <name>timestamp</name>
          <type>xsd:dataTime</type>
          <value>1352200525</value>
        </contextMetadata>
      </domainMetadata>
    </contextElement>
    <statusCode>                    <code>0</code>
<reasonPhrase>OK</reasonPhrase>
    </statusCode>
  </contextElementResponse>
</contextElementResponseList>
```

geolocation for Van2


```
<?xml version="1.0" encoding="iso-8859-1" standalone="no"?>
<contextElementResponseList>
  <contextElementResponse>
     <contextElement>
       <attributeDomainName>geolocation</attributeDomainName>
       <entityId>
         <name>Van2</name>
       </entityId>
       <contextAttributeList>
          <contextAttribute>
            <name>latitude</name>
            <type>double</type>
            <contextValue>43.734391</contextValue>
         </contextAttribute>
         <contextAttribute>
            <name>longitude</name>
            <type>double</type>
            <contextValue>7.184658</contextValue>
         </contextAttribute>
       </contextAttributeList>
       <domainMetadata>
         <contextMetadata>
            <name>timestamp</name>
            <type>xsd:dataTime</type>
            <value>1352226632</value>
         </contextMetadata>
       </domainMetadata>
     </contextElement>
    <statusCode>                    <code>0</code>
<reasonPhrase>OK</reasonPhrase>
     </statusCode>
   </contextElementResponse>
</contextElementResponseList>
```


Parcel geolocation combined events - Events description

Attributes:

- entity id= parcel-0
- vehicle id= Plane,Van1, Van2
- latitude
- longitude
- deliveryStep= Ordered , DeliveryInProgress, Delivered
- timestamp

Event File	Entity Id	Delivery step	Corresponding ParcelStep Event	Corresponding VehicleLocation Event
contextElementResponseListParcel Geolocation0.xml	parcel-0	Ordered	ParcelStep0	Geoloc0
contextElementResponseListParcel Geolocation1.xml	parcel-0	Delivery InProgress	ParcelStep1	Geoloc0
contextElementResponseListParcel Geolocation2.xml	parcel-0	Delivery InProgress	ParcelStep2	Geoloc2
contextElementResponseListParcel Geolocation3.xml	parcel-0	Delivery InProgress	ParcelStep3	Geoloc4
contextElementResponseListParcel Geolocation4.xml	parcel-0	Delivered	ParcelStep4	Geoloc7

Event File	Vehicle	latitude	longitude	timeStamp	Date
contextElementResponseListParcel Geolocation0.xml	Plane	43.660266	7.205032	1352191090	06/11/2012 09:38:10
contextElementResponseListParcel Geolocation1.xml	Plane	43.660266	7.205032	1352199104	06/11/2012 11:51:44
contextElementResponseListParcel Geolocation2.xml	Van1	43.665512	7.20353	1352207169	06/11/2012 14:06:09
contextElementResponseListParcel Geolocation3.xml	Van2	43.669812	7.203487	1352215801	06/11/2012 16:30:01
contextElementResponseListParcel Geolocation4.xml	Van2	43.734391	7.184658	1352226632	06/11/2012 19:30:32

queryContext response samples for geolocation parcel-0 entity

geolocation for parcel-0: step1


```xml
<?xml version="1.0" encoding="iso-8859-1" standalone="no"?>
<contextElementResponseList>
  <contextElementResponse>
      <contextElement>
        <attributeDomainName>parcelGeo</attributeDomainName>
        <entityId>
          <name>parcel-0</name>
        </entityId>
        <contextAttributeList>
          <contextAttribute>
              <name>vehicleId</name>
              <type>string</type>
              <contextValue>Plane</contextValue>
          </contextAttribute>
          <contextAttribute>
              <name>latitude</name>
              <type>double</type>
              <contextValue>43.660266</contextValue>
          </contextAttribute>
          <contextAttribute>
              <name>longitude</name>
              <type>double</type>
              <contextValue>7.205032</contextValue>
          </contextAttribute>
          <contextAttribute>
              <name>deliveryStep</name>
              <type>string</type>
              <contextValue>Ordered</contextValue>
```

```
                </contextAttribute>
              </contextAttributeList>
              <domainMetadata>
                <contextMetadata>
                    <name>timestamp</name>
                    <type>xsd:dataTime</type>
                    <value>1352191090</value>
                </contextMetadata>
              </domainMetadata>
            </contextElement>
              <statusCode>          <code>0</code>
<reasonPhrase>OK</reasonPhrase>
              </statusCode>
            </contextElementResponse>
        </contextElementResponseList>
```


geolocation for parcel-0 : step5


```
<?xml version="1.0" encoding="iso-8859-1" standalone="no"?>
<contextElementResponseList>
  <contextElementResponse>
    <contextElement>
      <attributeDomainName>parcelGeo</attributeDomainName>
      <entityId>
        <name>parcel-O</name>
      </entityId>
      <contextAttributeList>
        <contextAttribute>
            <name>vehicleId</name>
            <type>string</type>
            <contextValue>Van2</contextValue>
        </contextAttribute>
        <contextAttribute>
            <name>latitude</name>
            <type>double</type>
            <contextValue>43.734391</contextValue>
        </contextAttribute>
        <contextAttribute>
            <name>longitude</name>
            <type>double</type>
            <contextValue>7.184658</contextValue>
        </contextAttribute>
        <contextAttribute>
            <name>deliveryStep</name>
            <type>string</type>
            <contextValue>Delivered</contextValue>
        </contextAttribute>
      </contextAttributeList>
      <domainMetadata>
        <contextMetadata>
            <name>timestamp</name>
            <type>xsd:dataTime</type>
            <value>1352226632</value>
        </contextMetadata>
      </domainMetadata>
    </contextElement>
      <statusCode>          <code>0</code>
<reasonPhrase>OK</reasonPhrase>
```

</statusCode>
</contextElementResponse>
</contextElementResponseList>

6. Conclusions

On the technical level, Instant Mobility Scenario 1 has demonstrated the scalability and dynamicity (elasticity) of our proposed architecture. Our demonstrator has shown very low CPU usage (about 10% for each of the eight cores) despites heavy load, accelerated time (6x) and “journey monitoring” turned on!

On the business level, by running extremely realistic simulations, our prototype has demonstrated the benefit of mixing common-ride and public transport (1/3 of the journeys significantly improved). We do believe that by adding on-demand transport (e.g. taxies in the vicinity) we should be able to demonstrate even greater advantages. There is no doubt that multi-modal travel is going to be mainstream in the next 10 years, hopefully we will be part of it.

Instant Mobility Scenario 2 demonstrated how a “Dynamic time/place drop point” combined with the concept of a “Transport Exchange Portal” can contribute to decrease congestion in city centres in order to address issues with air- (NOx, SOx, CO2, particles, etc.) and noise pollution, while saving and adding convenience to the individual, in this case the consignee, or the person who is expecting delivery of goods. This is achieved by increasing the utilization of cargo vehicles and eliminates the unnecessary transports which occur when a delivery fails due to unavailability of the consignee and the subsequent extra journey in order to pick up the goods at a central pickup location. The next step after addressing these issues with low vehicle utilization and unnecessary transports, is to also consider how the vehicles can be driven in a fuel efficient way. This can be achieved by integrating with the traffic infrastructure in order to enable services like green light optimized speed advisory and prioritization of heavy vehicles.

Instant Mobility Scenario 3 demonstrated that The new paradigm “Traffic control in the cloud” is cost effective if compared to the current/state-of the art implementation in that:

- Seamless installation & configuration capabilities;
- Reduction of maintenance costs;
- easy repair intervention;
- High scalability.

In the near future our vision is to extend such constellation the “traffic control in the cloud” to the un-monitored area of a city as seen in the figure below.

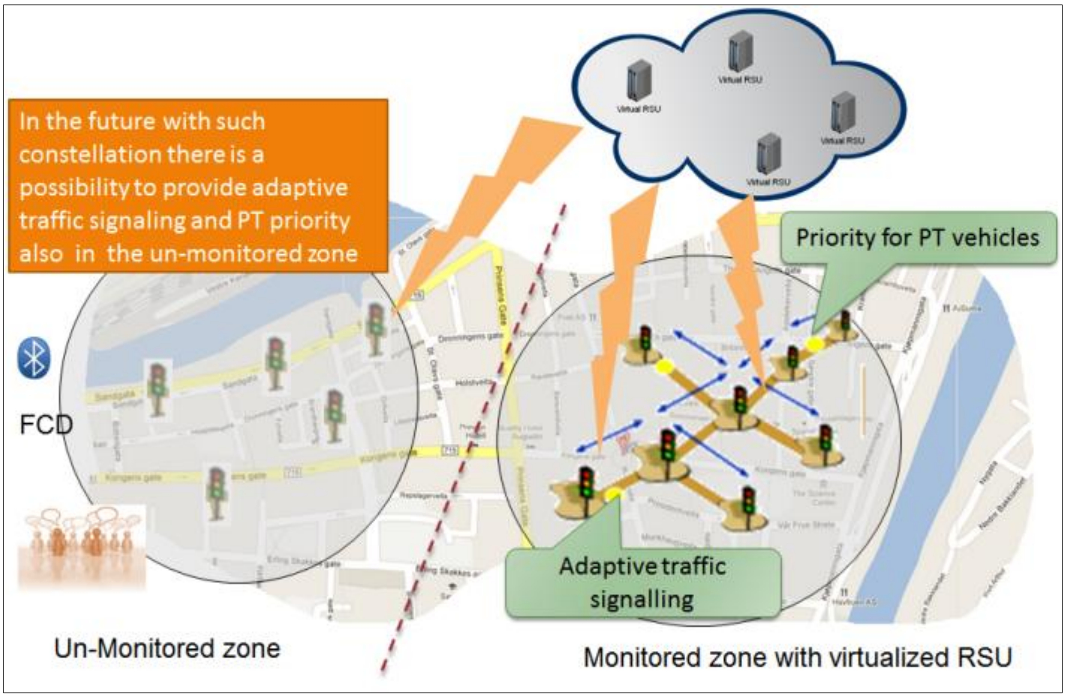


Figure 33. "Traffic control in-the-cloud" possible future extension

This will benefit the cities not only in terms of cost effectiveness and network capacity but also it will be possible to have other value added potential features. Normally with traditional constellation services such as priority, adaptive traffic control etc are provided only in the monitored area of the city. However, the new paradigm traffic control in the cloud will enable the public authorities to give services (priority, green wave, and other cooperative system services etc) also in the un-monitored part of the city through the integration of 3rd party information such as FCD, social network info, street light as sensor, Bluetooth, GPS data and other sensors info.

7. FI-Ware GE Validation

7.1 SELECTED GE (IM)

Chapter	Generic Enabler	Fi-Ware Chapter	Validators
3.1	Location Server - LOCS	DCM	Thales
3.2	Identity Management - DT GCP	Security	Volvo
3.2	Identity Management - NSN One-IDM	Security	Telecom Italia; Volvo; DLR
3.3	PubSub broker-Samson	DCM	France Telecom; Thales
3.3	Pub/Sub Context Broker - TI Context Awareness Platform	DCM	France Telecom; Thales
3.4	Semantic Application Support	DCM	Thales
3.5	Object Storage GE - FI-WARE Implementation	Cloud	Thales
3.6	Data Handling - PPL	Security	Thales
3.7	Service Description Repository	ASE	Telecom Italia
3.8	Complex Event Processing - PROTON	DCM	France Telecom
3.9	BigData Analysis - SAMSON BigData Platform	DCM	Thales, Volvo

7.2 Location GE

7.2.1 Use Cases (IM)

This GE facilitates indoors navigation and improves navigation in poor signal conditions (A-GPS)

7.2.2 Webinar

Availability:

- All phones (except possibly iPhone) support SUPL V1.0
- SUPL V2.0 required for WiFi/3G triangulation (required reporting several access point / Cell)
- Most phones in 2013 will have SUPL V2.0. For phone without SUPL V2.0 support, native applications can be used.
- Beware: The SUPL V2.0 native application will not support AGPS since it does not have access to the GPS low-level sensor API.

Interfaces provided by FI-Ware:

- NetAPI Terminal Location REST API
- MLP Interface

Testbed:

- Use a simulated handset (one single position) for which the position may be requested using an accuracy parameter (for high accuracy the GPS receiver will be activated)

It is possible to send a scenario (with moving position) for IM prototype demonstration (Barcelona) Security:

- Retrieval of his device location can be setup to required user authorization.

Capabilities:

- SUPL V2.0 provides AGPS for indoor location, integrates central reporting or users' locations and provides geo-fencing capabilities.

7.2.3 Issues

The location GE with SUPL V2.0 enabled devices could be quite useful and a key component for Instant Mobility. Not only it could provide centralized position of all travellers (Location Platform) but also provides AGPS capabilities for indoors/city environment (including WiFi triangulation positioning).
Unfortunately, the current testbed instantiation is source of concern to us. As we understand it, it supports only one simulated handheld location returning a static position.
What we really need to validate the GE, is the possibility to use the testbed with real handset devices (not simulated ones) and multiple of them (returning multiple locations).

7.2.4 Discussion

We discussed the possibility of integrating live terminals in our test bed in order to perform end to end tests. However, in order to support all services, SUPLv2 handsets are required. As presented in our webinar, SUPLv2 handsets are not yet available in the EU market today. The only possibility would be to use SUPLv1 terminals, but reducing the number of features to the single location retrieval via AGPS and CID. I confirm that SUPLv1 would not support WiFi.

We also discussed the possibility to use SUPLv2 Androïd application (developed in-house) on current available smartphones. This is something we will investigate internally and get back to you, but please consider that this would not be available in the short term.

A much quicker alternative that we discussed during the webinar is to use enhanced simulation tools in order to simulate a fleet of handsets with their own simulated path (or static position), which we are willing to investigate for Fi-Ware. I don't see in your request below anything related to this, is it no more an option? If so, we would need from you a detailed scenario description in order for us to assess the technical feasibility. Please use the Fi-ware general tracker to post your needs for the Location GE at https://forge.fi-ware.eu/tracker/index.php?func=browse&group_id=7&atid=162&set=&start=50.

7.2.5 Conclusion:

The location GE with SUPL V2.0 enabled devices could be quite useful and a key component for Instant Mobility. Not only it could provide centralized position of all travellers (Location Platform) but also provides AGPS capabilities for indoors/city environment (including WiFi triangulation positioning).
Unfortunately, due to the lack of SUPLv2 handsets during phase 2 life experiments, a SUPL v2 client will have to be installed on the handset.

7.3 Identity Management GE

7.3.1 Use Cases (IM)

This GE manages access to mobility services (token based)

7.3.2 Webinar

Here is a summary of our findings relative to the Identity Management enabler, or enablers as there are two of them. They are independent developments of essentially similar functionality.
The offering from Deutsche Telekom is called the Global Customer Platform (GCP) and is more focused on user management. A RESTful API is provided to manage the user accounts. To be able to use the GCP, all users need to have a valid e-mail address.We have not investigated this particular functionality closer, since it is not of major interest to us.
Our needs are instead in API-based authentication/authorization.
We have explained this during a conference call with both representatives from the NSN and Deutsch Telekom Identity GEs.
They proposed a solution based on SAML, which they could both support. Unfortunately SAML does not work very well with RESTful web services, since it uses the SOAP header to transfer the SAML messages and there is no standard way to use it with REST. They asked if it would be an option for us to use SOAP instead of REST. I'm a bit surprised that a project referring to itself as the "Future Internet" are suggesting a solution based on, what many would consider to be, legacy technologies.
The optimal solution would have been to use the upcoming OpenID Connect (it combines both the federated authentication approach of OpenID and the authorisation delegation mechanism of OAuth, see D4.16 section 4.6.1 for details) standard, which the IdM guys agreed would be a solution to our problem. That would however require a substantial development effort from FI-WARE and also taking into account that the OpenID Connect standard is not yet completely finalized, this is nothing that will happen during the phase 1 projects. In early 2013 a decision was actually made by FI-WARE to start the implementation of this functionality.

7.3.3 Issues

There is nothing in the roadmap relative to the OpenID Connect standard implementation. But there is something in the second release about token-based authentication. Not having this capability validated would significantly impact the risk associated with using this GEs in phase 2.
As of today (November 19) it seems the GE testbed instance is only available on a non-standard port (8443) which prevents access to it from behind a proxy.
There were no real developer friendly documentation provided by these enablers. A developer might be interested in how to integrate the enabler in, for example, a Java web project. The documentation provided, mostly consists of links to the standards upon which the enablers build. The owner of the enablers agreed to provide such documentation and develop examples of how to use the enablers from commonly used web application platforms, before the end of 2012.

7.3.4 Conclusion

The current testbed functionality of this GE shows all the possible authentication systems. In our opinion this GE should be suitable for Instant Mobility project, in particular in the use case diagrams of the scenarios considered in "D4.10 Driver and Traveler enablers". In the mobile application development this GE would be suitable for the login functionality. Unfortunately, the current testbed functionality of this GE is insufficient to address the risk associated to token-based authentication support. What are FI-Ware plans regarding this important issue?

7.4 Pub Sub Broker GE

7.4.1 Use Cases (IM)

7.4.2 Webinar

Here is a summary of our findings relative to the Pub Sub Broker enabler, or enablers as there are two of them. They are independent developments of essentially similar functionality.
Context Information is delivered in the form of Context Element which is a data structure. A given entity has a given number of attributes.
Notification of updates for values is possible but we will receive only the attributes where the values changed.
OMA NGSI does not include any definition of a type for entity ID.
Context provider could be the phone handset. The phone will not send its location all the time. It is up ot IM to decide if the device is passive or active. In this case the device could manage the update context at the rhythm we want. Hybrid solution is possible between push mode and pull mode to satisfy update context and notify context.
Complex rules should be managed through the CEP GE
Integration with the security GE because there are sensitive data? In second release it is expected that a token could be use for some operations. But we have to define when a more specific control should be applied on some data or for some particular entities. It is a critical issue for...
Regarding OMA NGSI XML file: where are the units for the different values? It is very important to avoid any confusion which could fail to adopt such format.

7.4.3 Issues

Difficulties encountered with network connections, switch from one port to another (port 1026 to 1033 available) to perform all tests.

7.4.4 Conclusion

We had planned to use this GE to monitor travelers locations in order to get alerted in case of unexpected delays during the journey and of the proximity of point of interest. Unfortunately, it is not possible to plug-in your own listener handler to this pub/sub. The only kind of automatic notification we can build are based on interval values of latitude/longitude which is useless (we are interested to know if the traveller is on a specific road or at walking distance of a point of interest, not if its latitude/longitude are in specific ranges). Unless the pub/sub allows custom code (in our case geocoding) to be plugged for notification purpose, the interest of this GE is very limited. Once again, we had identified a GE "Location Monitoring" (see corresponding EPICS) which was supposed to address specifically these issues.

7.5 Semantic Support GE

7.5.1 Use Cases (IM)

This GE supports the fusion of mobility data (e.g. public transport timetables) from various sources (public transport operators).

7.5.2 Webinar

- What functionalities (if any) FI-Ware adds to the NeOn Toolkit?

➢ We plan to provided a Neon Toolkit's plugin to interact with the Semantic Application Support GE in order provide the GE functionalities inside the Neon Toolkit

- Could you tell us what function this plugin will provide and when it will be available?

- When will there be a final solution for the access API to the RDF repository? We are using currently raw Sesame but providing a GE API for that was in the agenda.

➢ We will be having a Proof-Of-Concept demo around a week, and we will use the raw Sesame API. For another PoC we have in February we will try to migrate to the 'official' API and send you feedback.

7.5.3 Issues

N/A (not yet validated)

7.5.4 Conclusion

It is not clear at this time, what are the advantages of using this FI-Ware GE versus using the NeOn Toolkit directly.

7.6 Object Storage GE

7.6.1 Use Cases (IM)

This GE would support the storage of historical data (e.g. itineraries) into the cloud for later analysis

7.6.2 Issues

We tried to integrate this GE into our final prototype; but due to difficulties in accessing the instance from a mobile station we were able to demonstrate its usage during the FI-PPP event in Barcelona.

7.6.3 Conclusion

Despite the backdoor provided to us for accessing this GE, we have not been able to make it work for us (runtime exceptions).

7.7 Data Handling GE

7.7.1 Use Cases (IM)

This GE would support the diffusion of resources such as user pictures based on usage policy specified by the user.

7.7.2 Issues

The testbed instance being not easily accessible, SAP provided us with a local instance to run on our web server.

7.7.3 Conclusion

This GE has been integrated into our final prototype to manage the diffusion policy of the travelers/drivers pictures (shown for identification purpose during the common ride scenario). It is the only enabler we successfully integrated in our prototypes showed in Barcelona.

7.8 Service Description Repository GE

7.8.1 Use Cases (IM)

This GE could be used to publish the “Personal Travel Companion” services of Instant Mobility.

7.8.2 Issues

The IP-based firewall that protects the instances is not suitable for work with dynamic IP addresses.
In “Repository - User and Programmer Guide” there isn’t an API that answers with a list of all available services.

7.8.3 Conclusion

At the moment the service maybe improved with a better documentation and with the addition of the API that provides the list of all available services.

7.9 Complex Event Processing GE

7.9.1 Use Cases (IM)

This GE is used to detect traffic jam (e.g. accidents) in real-time (vehicles on their itinerary whose context location is not changing much for a specified duration).

7.9.2 Webinar

- Would the CEP be appropriate to detect entries within an area using context location events?

➢ No. For events uncorrelated between each other's (e.g. a location within a predefined area) the pub sub context broker is more appropriate.

7.9.3 Issues

N/A (not yet validated)

7.9.4 Conclusion

Some overlap with the Context Pub/Sub. We should use the CEP only for complex events involving temporal patterns (e.g. time window) or for correlations between events.

7.10 Big Data Analysis GE

7.10.1 Use Cases (IM)

This GE would have been key to IM in order to achieve its scalability requirement (up to 10 millions user in the Istanbul city). .

7.10.2 Issues

The big data enabler is the Telefonica Samson MapReduce platform. It is not really a general purpose data store, but rather a specialized tool for tasks like extracting information on huge data streams, like log files generated by telecom systems.
GE does not bring added value to existing Open Source Solutions.

7.10.3 Conclusion

For Instant Mobility we opted for data distribution (OMG DDS) to perform big data analysis. More flexible since it does not require processing do be independent (finding an itinerary solution modify the vehicle capacity and may invalidate concurrent solutions). It should be noted that we had identified the "Data Distribution GE" as a key enabler for IM to no avail (see corresponding EPICS). For our final demo in Barcelona we integrated DDS (OpenSplice Implementation) to our prototype in order to achieve our scalability requirement (up to 10 millions users). We have demonstrated that using DDS, our system is infinitely scalable; doubling the number of users requires only doubling the number of servers (in Barcelona, four multi-modal itinerary calculation engines were used).

7.11 Connected Device Interface

7.11.1 Use Cases

This GE is involved in Instant Mobility Scenarios 1 and 2 that are Personal Travel Companion and Smart City logistics. Both those scenarios require mobile applications running on user terminals; location, personal data, sensors are considered fundamental enablers for those scenarios.

7.11.2 Issues

CDI GE was not available when Instant Mobility prototypes were developed, so it was not possible to integrate CDI GE in prototypes.
CDI does not include access to NFC radio interface. We believe NFC must be included in CDI because it enables secure and easy interaction between travellers and transport infrastructure (e.g. for access control or ticketing functions). The number of NFC-enabled mobile terminals is rapidly increasing.
CDI has no vocal tool to facilitate development of transport/mobility related applications (particularly important for automotive use cases like ride-sharing, in order to avoid driver distraction).

7.11.3 Conclusion

Despite the issues listed above, we can say that overall CDI GE contains a useful list of enablers that will facilitate the development of transport related mobile applications.
FI-WARE was also recommended to provide the capability to authenticate mobile users by means of the information stored in the mobile terminals' UICC (Epic #349) . This requirement has recently been adopted by FI-WARE; for sake of precision the enabler is not in I2ND.CDI but in I2ND.S3C chapter, with a liason to Identity Management GE (see http://forge.fi-ware.eu/plugins/mediawiki/wiki/fiware/index.php/FIWARE.OpenSpecification.I2ND.S3C).
The table below completes our analysis rating the importance of each particular CDI enabler in the perspective of transport and mobility related services.

CDI GE family	GE Title	Comment
On Device Interface	Quality of Experience	Low relevance for IM.
	Device Features	Medium to high relevance for IM, to optimize screen usage, performance and battery autonomy.
	Media Services	Low relevance for IM.
	Phone	Medium to high relevance for IM. In some use cases for Multimodal Transport travellers may be automatically put in contact with assisting personnel.
	Messaging	Medium relevance for IM.
	Device Connectivity	Medium relevance for IM.
Remote Management Interface	Device Configuration	Medium to high relevance for IM.
	Application Management	Medium to high relevance for IM.
	Firmware Update	Medium to high relevance for IM.

CDI GE family	GE Title	Comment
Mobility Manager Interface	S3C Interface	Medium to high relevance for IM.
	Interface to Applications	Medium to high relevance for IM.
	Interface to QoE Compoment	Medium to high relevance for IM.

8. References

- FI-WARE wiki <http://forge.fi-ware.eu/plugins/mediawiki/wiki/fiware/index.php/Main_Page>
- FI-WARE product vision <http://forge.fi-ware.eu/plugins/mediawiki/wiki/fiware/index.php/FI-ARE_Product_Vision>
- FI-WARE catalogue <http://catalogue.fi-ware.eu/>
- FI-WARE technical roadmap <http://forge.fi-ware.eu/plugins/mediawiki/wiki/fiware/index.php/FI-WARE_Technical_Roadmap>