

ICT-2011.8
GET Service Project
2013-318275

Deliverable D8.3.2

Deployed Prototype Platform

September 30, 2015
Public Document



GET
SERVICE



The GET Service project (<http://www.getservice-project.eu>) has received funding from the European Commission under the 7th Framework Programme (FP7) for Research and Technological Development under grant agreement n°2012-318275.

Project acronym:	GET Service
Project full title:	Service Platform for Green European Transportation
Work package:	8
Document number:	D8.3.2
Document title:	Deployed Prototype Platform
Version:	v1.0
Delivery date:	September 30, 2015
Actual publication date:	September 24, 2015
Dissemination level:	Public
Nature:	Prototype
Editor / lead beneficiary:	Remco Dijkman (TU/e)
Authors:	Remco Dijkman (TU/e)
Reviewers:	Paul Grefen (TU/e) Mathias Weske (HPI)

Contents

Executive summary.....	4
1 Introduction.....	5
1.1 Project Goal	5
1.2 Work package goal	5
1.3 Deliverable goal	5
1.4 Deliverable structure	5
2 Public Prototype Demo Scenario.....	6
3 Conclusion.....	10

Executive summary

This report briefly describes the publicly available prototype of the GET Service Platform. The prototype is made available on the web and has a subset of the functionality of the complete GET Service Platform, which is described in Deliverable D2.4.2. The prototype serves as a demonstrator and aims to give the general audience an overview of the potential of a platform for Green European Transportation Planning and Monitoring.

This document accompanies the actual deliverable, the public prototype, and provides a scenario that the reader can follow to help understand the prototype. The prototype can easily be used in a web-browser at <http://is.ieis.tue.nl/research/getservice>.

1 Introduction

This deliverable presents the public prototype that was delivered by the GET Service project. This section provides the background to this deliverable, by presenting the goal of the project as a whole, the goal of the work package of which the deliverable is a part, and the goal of the deliverable itself. Finally, it presents the structure of the remainder of the deliverable.

1.1 Project Goal

The GET Service platform provides transportation planners with the means to plan transportation routes more efficiently and to respond quickly to unexpected events during transportation. To this end, it connects to existing transportation management systems and improves on their performance by enabling sharing of selected information between transportation partners, logistics service providers and authorities. In particular, the GET Service platform consists of components that: (i) enable aggregation of information from the raw data that is shared between partners and transportation information providers; (ii) facilitate planning and re-planning of transportation based on that real-time information; and (iii) facilitate real-time monitoring and control of transportation, as it is being carried out by own resources and partner resources.

By providing this functionality, the GET Service platform aims to reduce the number of empty miles that are driven, improve the modal split, and reduce transportation times and slack, as well as response times to unexpected events during transportation. Thus it reduces CO2 emissions and improves efficiency.

1.2 Work package goal

This deliverable is part of work package 8, which has as its goal to disseminate and exploit the results of the GET Service project. To achieve that goal, work package 8 develops a dissemination plan, an exploitation plan and various dissemination and exploitation results.

1.3 Deliverable goal

This deliverable forms one of the dissemination results of work package 8: the public prototype of the GET Service Platform that is made available to a general audience. The public prototype of the GET Service Platform provides a subset of the functionality of the overall GET Service Platform, which is described in deliverable D2.4.2. By making this functionality publicly available, the public prototype aims to help a general audience understand the benefits of a platform for collaborative transportation planning and monitoring.

1.4 Deliverable structure

This document accompanies the actual deliverable, the public prototype, which is available at: <http://is.ieis.tue.nl/research/getservice>. In the following chapter, this document describes a suggested scenario that the reader can follow through the public prototype, to help understand it. To understand the GET Service Platform as a whole, the reader is referred to deliverable D2.4.2.

2 Public Prototype Demo Scenario

A typical scenario for executing a transportation order is:

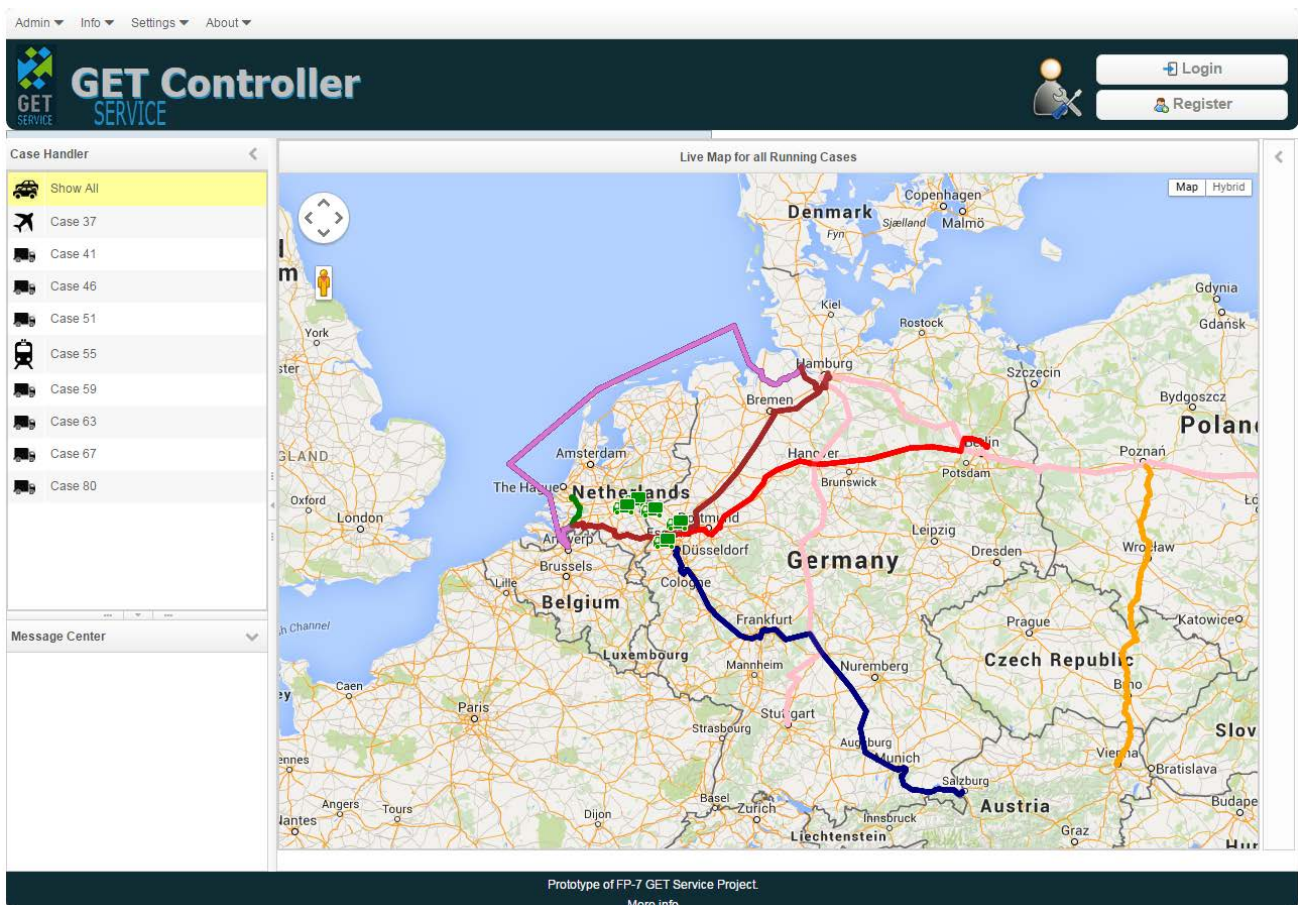
1. Enter a transportation order into the system.
2. Plan routes for the transportation orders.
3. Plan assets that execute the transportation orders.
4. Monitor and control the transportation orders as they are being executed.

We will show how these typical tasks can be executed in the public prototype of the GET Service Platform.

Open the public prototype of the GET Service Platform at:

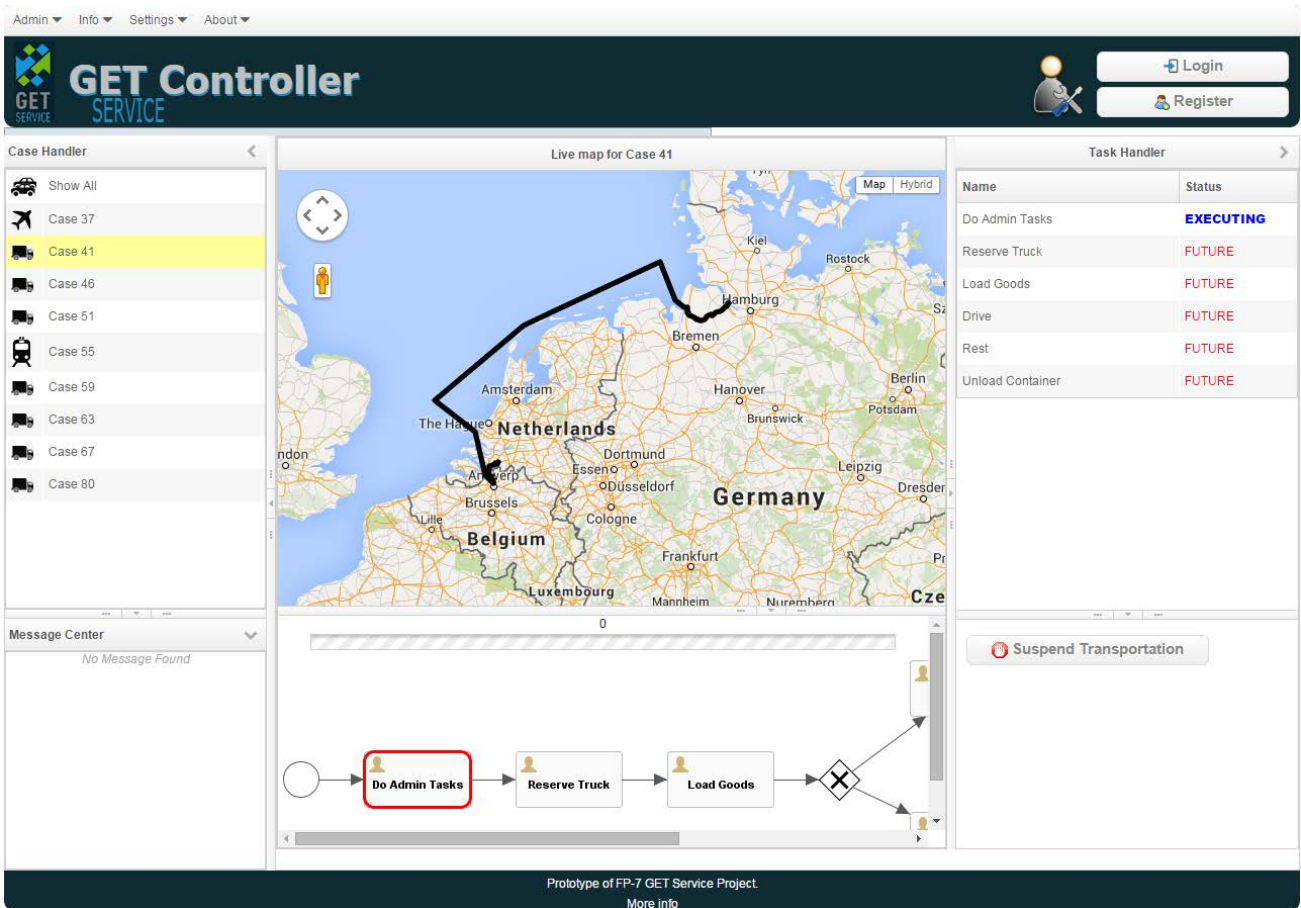
<http://is.ieis.tue.nl/research/getservice>.

The web-browser shows a typical user interface for a transportation management system, like in the screenshot below. On the left, it shows the transportation orders that are currently in the system. On the right, it shows the routes that are taken by the transportation orders, the resources that are used to execute these transportation orders (in black) and resources that are available to execute other transportation orders (in green).



Select one of the transportation orders.

The screen shows only the routes and the resources that are associated with the selected transportation order. On the right of the screen, a new view opens that shows the tasks that are associated with the selected transportation order, such as in the screenshot below. Examples of tasks are ‘drive’, ‘load goods’, and ‘fill out forms’. The view also shows the status of the tasks, by showing whether they are already completed, are currently being executed, or may have to be executed in the future.



Create a new transportation order, by selecting ‘admin’ and then ‘add transportation order’ in the menu.

A window appears in which the details of the transportation order have to be provided. While arbitrary information can be entered here, we suggest that the reader selects one of the transportation orders that were created for demonstration purposes. The reason for this is that the information for planning the transportation order is obtained from the PTV xServer and this system only provides a restricted subset of information for demonstration purposes.

Select ‘Export Case Scenario (1)’ from the drop-down list at the bottom of the dialog.

The transportation order is pre-filled with demo information.

Click ‘Find Addresses’

The prototype now immediately continues with route selection, as shown in the screenshot below. You can click routes to look at their details and you can select a route by right-clicking it. For example, you could select the route with the lowest CO2 emission.

The screenshot shows the 'GET Controller SERVICE' interface. At the top, there is a navigation bar with 'Admin', 'Info', 'Settings', and 'About' menus. Below this is the 'GET Controller SERVICE' logo and a 'Back To Main Page' button. On the right, there are 'Login' and 'Register' buttons. The main content area is split into two parts: a left sidebar and a right map area. The sidebar, titled 'Alternative routes', lists 'All Routes', 'R1_Road', 'R2_Rail' (highlighted in yellow), and 'R3_Barge'. The map area shows a map of Europe with a yellow line indicating a route through the Netherlands and Germany. Below the map is a table with the following data:

Name	Total Cost	TotalTravelTime	TotalDistance	Total Transfer	Start	Total Emission
R1_Road	257	171	202	0	Tue Sep 08 16:34:22 CEST 2015	177.1479220152341
R2_Rail	125	1525	210	1	Tue Sep 08 16:34:22 CEST 2015	44.06628960006674
R3_Barge	146	7285	278	1	Tue Sep 08 16:34:22 CEST 2015	159.12065624918435

Right click on a route, then click 'Select Route'.

The prototype now continues with composition of the transportation process that is based on the transportation plan and with asset selection. The transportation process determines the list of tasks that will be executed for the transportation order.

Admin Info Settings About

GET Controller
SERVICE

Back To Main Page Login Register

Map

Available Assets

Process Properties

of Instances Process Model

1

Run Transportation Process

Last update: 16:37:42.531

Source ID	Distance (KM)
57011	40.5
42555	22.68
12543	47.19
79122	36.22
10005	43.56

Select Asset

Prototype of FP-7 GET Service Platform.
More info

Select one of the transportation processes that is listed and click 'Run Transportation Process'.

Select one of the assets that is listed and click 'Select Asset'.

Your order is now planned and you return to the initial screen, where you can monitor all your orders (including the one that you just added). Your order should be the one with the highest order number.

You can decide to execute tasks for your order, or to suspend your order. For some of the tasks that you execute, you may need to enter data. You can also decide to suspend an order, should you observe that the transportation plan is disrupted. After you suspend an order, you can replan it, by clicking 'admin' and then 'update transport plan' in the menu.

3 Conclusion

This report described a scenario that the reader can follow through the public prototype of the GET Service Platform. It accompanies the publicly available version of the platform, which can be accessed at <http://is.ieis.tue.nl/research/getservice>.

The usage scenario demonstrates various functionality classed of the GET Service platform. In particular, it demonstrates the two main functions of transportation planning and monitoring.

The full GET Service platform, as it is described in deliverable D2.4.2, provides more functionality. In particular, it provides more advanced functionality for handling real-time data and using that data for planning. We refer the reader to deliverable D2.4.2 to read more about this functionality.