ICT cloud-based platform and mobility services available, universal and safe for all users

D3.2.1 MoveUs cloud-based platform: specification and architecture

Deliverable Id :	**D3.2.1**
Deliverable Name :	**MoveUs cloud-based platform: specification and architecture**
Status :	**Final**
Dissemination Level :	**PU**
Due date of deliverable :	**M12**
Actual submission date :	**M13**
Work Package :	**WP3**
Organization name of lead contractor for this deliverable :	**TECNALIA**
Author(s):	**Sergio Campos, Eider Iturbe, Begoña Molinete**
Partner(s) contributing :	**ALL**

Abstract: This document describes the current outcome of Task 3.2, which addresses the MoveUs cloud-based platform architecture, detailed specifications and high level interfaces between the different components. At this stage, the architecture definition process is technologically neutral, remaining technology choices as part of WP5.

HISTORY

Version	Date	Modification reason	Modified by
0.1	29/05/2014	TOC	Sergio Campos, Eider Iturbe
0.2	27/06/2014	Inputs from all pilots	All
0.3	31/07/2014	Mapping DFDs WP2 High-level functions	TECNALIA
0.4	25/09/2014	Detailed module descriptions	SICE, SOF, ATOS, TUT, TECNALIA
0.5	29/09/2014	Final revision and improvements	SICE, SOF, TECNALIA
0.6	10/10/2014	Minor adjustments after crosscheck with WP3 deliverables	TECNALIA
0.9	22/10/2014	Final review. Version ready for Quality Check	TECNALIA
1.0	23/10/2014	Final Version & Quality Check	ATOS

Contents

List of Figures

List of Tables

List of Abbreviations

<Abbreviation>	<Explanation>
API	Application Programming Interface
App	Application
ATOS	ATOS SPAIN
BT	Bluetooth
BTFix	Fix Bluetooth Reader
CAI	Common Agreed Interface
CF	Carbon Footprint
D	Deliverable
DFD	Data Flow Diagram
DoW	Document of Work
EC	Energy Consumption
ESB	Enterprise Service Bus
FCD	Floating Car Data
IAM	Identity & Access Management
ID	Identification
IT	Information Technology
ITS	Intelligent Transport System
KPI	Key Performance Indicator
LDSS	Local Data & Service Servers
MAP	Map Data
MSG	Message
PI Hub	Pseudo ID Hub
POI	Point of Interest
PT	Public Transport
RSU	Road Side Unit
RT	Real Time

SICE	Sociedad Ibérica de Construcciones Eléctricas, S.A.
SIRI	Service Interface for Real Time Information
SOF	Softeco Sismat Srl
SSM	Signal Status Message
TECNALIA	Tecnalia Research and Innovation
TRE	Tampereen Kaupunki
TUT	Tampere University of Technology
UC	Use-case
UISP	Upstream Information Service Provider
URL	Uniform resource locator (internet)
WP	Work Package
WS	Web Service

Executive Summary

The objective of deliverable D3.2 is to define a comprehensive architecture for the MoveUs cloud-based platform, the specifications of its core facilities and the high level interfaces between the different platform components.

The methodology adopted to define the specifications of this platform integrates insights and approaches from different engineering fields: traffic engineering, data analytics, software architecting and cloud-based computation deployment. Each view focuses or emphasizes specific aspects: functionality, performance and openness allowing third parties access to exploit the information and extend the platform. As key challenges, the demanding real-time requirements, the vast amount of data to be handled, the scalability and availability needs and finally, the compliance of security/privacy normative are worth mentioning.

The concept of ITS reference architecture has been presented during this work by analyzing existing solutions and by adopting and adapting selected best-practices and knowledge. In conclusion, we found that our main challenges and added-value functionalities, i.e. incentive management, energy efficiency and service customization go beyond the current ITS reference architectures, meaning that they have not been explicitly addressed before and consequently, remain uncovered by these architectures. Therefore, the project extends this basis with the introduction of a service view, a view on the mobility-domain in terms of services linked to the functions those services perform.

As mentioned in other related deliverables, the Work Package 3 schedule determines an iterative approach. This deliverable sets the foundations for the platform development. At this stage, the architecture definition process is technologically neutral, remaining technology choices as part of WP5. An update and completion of this document is expected by M23 to reflect the architecture "as-built" once the design details, the technologies deployed and the underlying business model are clearly defined. This point is particularly noteworthy when we deal with the cloud paradigm.

1 Scope of the document

This Deliverable D3.2 is the first outcome expected from task **T3.2 – Specification and architecture design of the MoveUs cloud-based platform.**

Task T3.2 is part of Work Package 3, whose **objectives** are the following:

- To define the data models relevant for MoveUs operation.
- To define the high-level architecture for the MoveUs Cloud-based platform and its functional specifications in detail.
- To provide detailed specifications and design for the set of services to be provided in MoveUs pilots.
- To identify the data security and privacy issues to be taken into account in the MoveUs architecture and include them in the definition of the platform and services.
- Develop innovative business models determining the users' willingness to pay for the uptake of MoveUs services.

Specifically, Task 3.2 is aimed at defining an **architecture for the MoveUs cloud-based platform**, able to support the use-cases previously elicited in WP2 and the advanced city services defined in Task 3.3.

This deliverable D3.2 deals with the specification and architecture design of the MoveUs cloud-based platform, providing full detail from both technical and functional perspectives. Together with all remaining deliverables of Work Package 3 it can be seen as part of the technical specification of the entire MoveUs system.

Specifically, with respect to the platform, the deliverable D3.4 - Data Security & Privacy in the MoveUs system architecture addresses horizontal and system-wide security analysis and design aspects. For this reason, as far as the MoveUs platform is concerned, both deliverables should be regarded as complementary, jointly conforming a single and unique specification that will drive the implementation phase, with D3.4 covering the security aspects not specified in D3.2 but having a clear impact on the platform development.

Considering this, some decisions will be made on the implementation of certain security features and recommendations at the different city pilots as identified in the technical specification. These will be tackled during the 2nd project year, mostly focused on system and service implementation.

2 Introduction to MoveUs Cloud-based Platform

According the Description of Work, the targeted MoveUs Platform is a cloud-based mobility management platform which will collect input data from distributed heterogeneous sources and process these data to infer valuable information of the transport status and users' mobility patterns, ensuring data privacy and security all along the handling process. Specific functionalities envisaged such as multi-modal trip planning or traffic management add new challenges such as the highly demanding real-time operation requirements, ITS equipment control and continuous user tracking, making the architecture concept evolve towards a desegregated approach.

These functional requirements demand architecture and infrastructure specific features such as: massive data handling, real-time processing and actuation, scalability and high-availability, qualities in which cloud-based solutions provide their differential value. On the other hand, the objective of providing developers and third parties with access to these data and services, enabling the extension of the project results, points out to the concept of Platform as a Service (PaaS).

The following figure presents the MoveUs operational scenario, showing the main actors and roles, architecture elements and information flows.

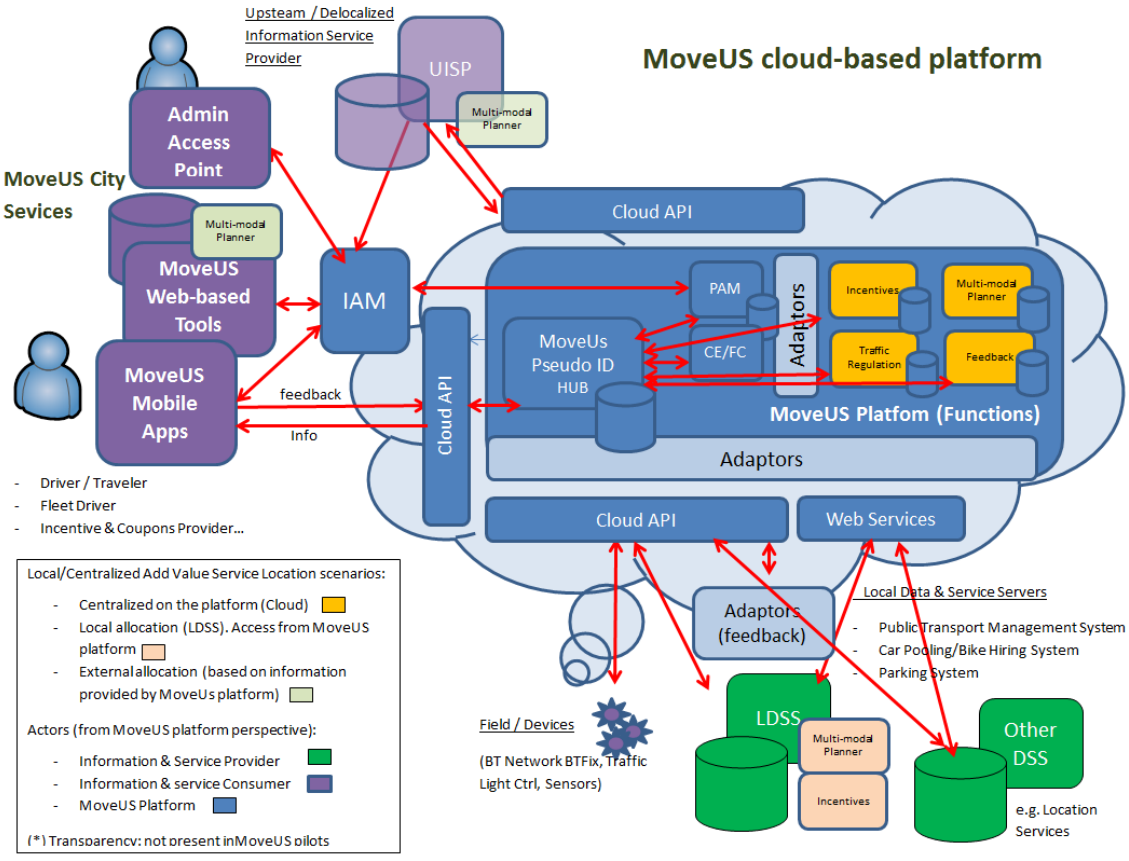


Figure 1 MoveUs operational scenario

Conceptually, MoveUs follows the approach of previous reference projects as In-Time or Co-Cities, where the interaction with the platform data and service consumers is supported by the so-called Common Agreed Standardized Interface (CAI), which determines exactly the information exchange and syntaxes of the application interface, providing a standardised access. Consumers include MoveUs mobile apps and specific city services (defined in D3.3 [8]), but also upstream and delocalized Information Service Providers (UISP).

Downstream, the communication and data capture from different Data & Service Servers (LDSS) (local or external) or field equipment, is realized by means of data adaptors that transform heterogeneous input formats into the MoveUs internal model (whose data model has been defined in D3.1 [7]). The schema is extended with services for feedback loop data, providing additional information captured in site or a quality of service evaluation.

Given the importance of security/privacy aspects and their impact on the architecture, also the security components and flows are depicted.

- The independent Identity & Access Management (IAM) component is responsible for establishing users' identity during registration and then authenticating the user.
- The Pseudo ID Hub (PI Hub) manages the correlation between registered UserIds and Pseudo Ids used for storing their information.
- Cloud APIs. Application Programming Interfaces that enable applications to request data or computing services to the cloud infrastructure.
- Web Services (WS). Direct Web application components.
- MoveUs Functional Components. Components providing MoveUs functionalities: consumption and carbon footprint estimation and measures, incentive management, multi-modal planning, feedback management and specific traffic management capabilities (priority request and smart crossing) among others.

Taking deliverable D2.2 on use-cases [6] and city existing infrastructure and servers as a reference, the MoveUs platform architecture has been conceived searching for flexibility on the allocation of functionalities, aiming to cover the most possible implementation scenarios and use-cases not only in the pilots but also in other adopting cities. These functionalities may be local (provided by existing LDSS and used internally by the platform), fully supported by the MoveUs platform (cloud located) and/or external (UISP or specific MoveUs City Services). As a consequence, the component/service specifications will be abstract enough to allow their deployment in different locations and technical implementations (e.g. communication protocols).

The adopted approach covers the current reality and scope of MoveUs, but also opens the potential exploitation of the results beyond the project lifetime.

3 Methodology

The driving aspects of the technical specification for the MoveUs platform are:

- The general project indications from the DoW
- The Use Case definition for each city and the high level Functional blocks achieved in Task 2.5
- The incentive-based model developed in Task 2.4
- The energy-efficiency model developed in WP4
- The privacy and security recommendations and indications developed in T3.4

3.1 ITS Reference Architecture

The advantage of using a standard or de-facto reference architecture as a basis relies on interoperability, avoidance of vendor lock-in situations and in general, on the consistency of information delivered to end users while manufacturers and designers focus on added value (design optimization) aspects.

Basically, a reference architecture defines the structure, behavior and integration of a given system in its operational context by defining functions, system partitioning into logical or functional entities and definition of data stores and information/data flows between the different systems, subsystem and nodes.

In the ITS domain, some experiences are identified:

- **US National ITS Architecture** [4], funded by the United States Department of Transportation (USDOT) and supported by a large ITS community. The National ITS Architecture provides a common framework for planning, defining, and integrating intelligent transportation systems.

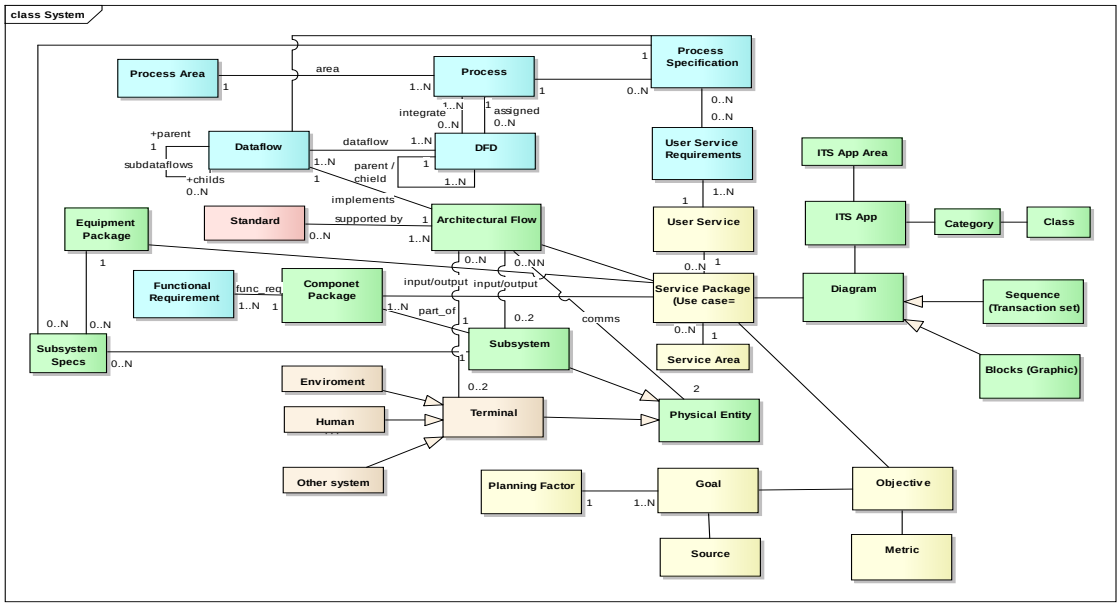


Figure 2 US ITS Reference Architecture Metamodel

- **European ITS Framework Architecture** (**FRAME** Architecture) [5], covers the following ITS areas: In-Vehicle Systems, Traveler Assistance – Pre-Journey and On-Trip Planning, Travel Information, Support for Law Enforcement ,Freight and Fleet Management and partially, support for Cooperative Systems. It is supported by computer-based tools.

  The system structure contains a number of viewpoints: organizational viewpoint, functional viewpoint and links with the ITS implementation (physical viewpoint) integrating communications and information elements.

  Current deployments include CHARM-PCP, interoperability sample between the Highways Agency (UK) and Rijkswaterstaat (the Netherlands) and EASYWAY, used to create ITS Architecture subsets for Member States and their regions, as well as for RTD projects.

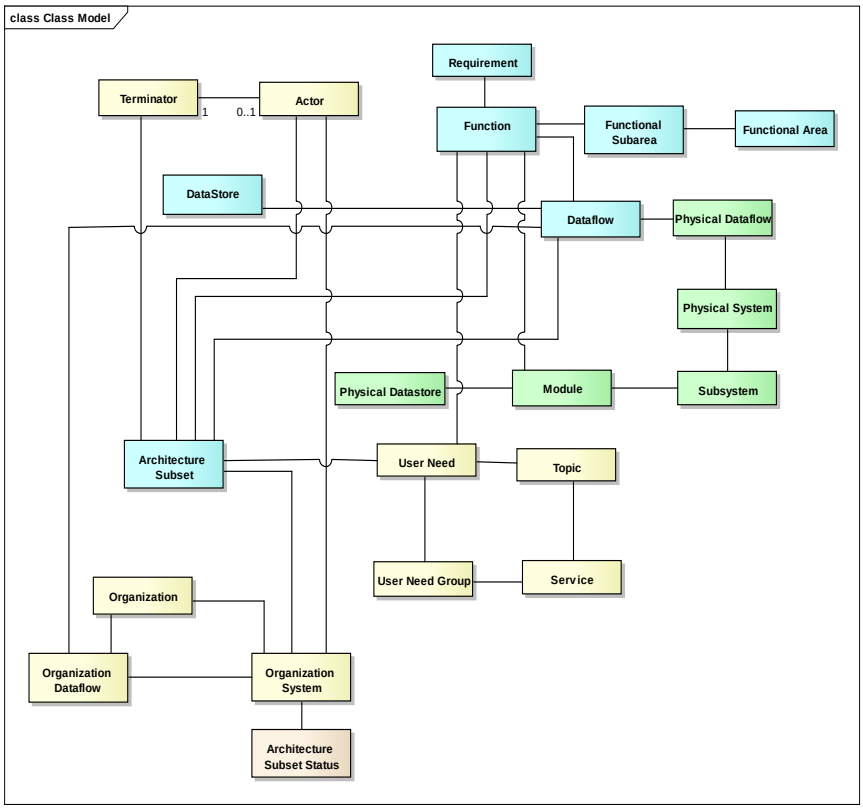


Figure 3 FRAME European Reference Architecture Metamodel

- **Other solutions** are the Canadian ITS Architecture and the Japanese national ITS architecture, adopting an object-oriented method. Existing harmonization Initiatives (EU-US-Japan) are currently supported by ERTICO.

FRAME is chosen as ITS Reference Architecture in the context of MoveUs.

3.2 General Methodology

FRAME defines a methodology which starts from a well-defined set of functions, users, datasets and ends in a set of data stores and dataflows ensuring consistency links. Nevertheless, the divergence between the common needs and requirements considered by FRAME and the peculiarities offered in MoveUs, and the lack of a

user-friendly tool able to update the resulting Dataflows Diagrams (DFDs), forced us to adapt the workflow.

Hence, the finally adopted methodology covers the following steps:

1. Analysis of relevant projects in the domain (ITS reference architectures. In-Time, Co-Cities, etc.) [5],[4],[2],[3]
2. Analysis of MoveUs requirements and use cases (and Sequence Diagrams, Forms)
3. MoveUs Functional Architecture definition (Data Flow Diagram)
    a. Elicitation of User Needs per pilot (according to FRAME methodology)
    b. Elicitation of Terminals (Actors and external systems) per pilot (according to FRAME methodology)
    c. Selection of candidate Functionalities (automatically, supported by FRAME tools)
    d. Decision on what can be re-used, alignment and identification of remaining gaps. Functions, data stores, data flows and terminators

    At this stage, in order to ensure a common functional view of the whole system between WP2, D3.2 and D.3, the resulting functionalities are mapped on MoveUs High-level functions (final result from T2.5).

4. Definition of the MoveUs platform service Viewpoint
5. Technical/Application (sub-systems and modules) architecture definition
    a. Sub-Systems. Containing element types of the functional architecture (functions, data stores) and communicating with either another Sub-System or terminators
    b. Modules. Aggregation of related functions.

6. Mapping functionalities/services vs. modules
7. Module detailed functional description
8. Physical/Deployment viewpoint definition
9. Cloud-based platform definition.

The Work Package 3 schedule has determined an iterative approach, which has led to refine the Task 3.1 outcomes as more complete information was received on Use-Cases and City Services, as origin of MoveUs platform requirements in terms of functionalities and also interfaces. Although the present specification has been conceived as technologically neutral, once the technology blocks (e.g. middleware, persistence systems) have been identified, a brief analysis of existing products has been conducted. This information is mandatory for the definition of a Global Technical Architecture (6.4) and the subsequent cloud deployment choice analysis (6.5.6). Final decisions on specific products, communication protocols and cloud deployment will be made in the scope of WP5, ensuring full alignment with Task 3.5.

4 User Needs

This section collects the results obtained from checking the user needs defined in FRAME versus the requirements identified from the use-cases and elicited from each city.

4.1 Madrid Pilot

4.1.1 Preliminary User Needs

					MADRID			
					UC1	UC2a	UC2b	UC3
Id	Level		Title	Description				
2.1	2		Transport Planning Support			1		1
2.1.1	3		Information Management			1		1
2.1.1.1	4			The system shall be able to produce information for travelers on the traffic and travel conditions of all transport modes relevant to the geographical area covered.		1		1
2.1.1.3	4			The system shall be able to collect traffic data for road network use analysis and prediction calculations.		1		1
6.1	2		Pre-trip Information			1		1
6.1.0	3		Objectives			1		1
6.1.0.1	4			The system shall provide emergency, or urgent, information to all road users free of charge.				1
6.1.0.3	4			The system shall be able to provide accurate, credible, timely, and easy to understand traffic and travel information where it may be of benefit to the user.		1		1
6.1.0.4	4			The system shall be able to provide information on alternative routes where they are quicker, cheaper, shorter, scenic, etc.		1		1
6.1.0.5	4			The system shall enable travelers to plan their trip using their own travel criteria (modes of transport, time of departure/arrival, road selection criteria, etc.).		1		1
6.1.1	3		Modal Choice			1		1
6.1.1.2	4			The system shall be able to provide trip information on other modes of transport, e.g. for demand-spreading when major events occur, or when weather conditions, strikes, cultural or sports events etc. cause problems for one mode.		1		1
6.1.1.3	4			The system shall be able to provide current and forecast traffic and travel information for all modes at local, regional, national and international		1		1

					MADRID			
					UC1	UC2a	UC2b	UC3
Id	Level		Title	Description				
				levels.				
6.1.1.4	4			The system shall be able to provide extensive multi-modal trip information, e.g. prices, fares, routes, forecast & current traffic situations, traffic control, demand management measures, local warnings, special events, weather conditions, hotels etc.		1		1
6.1.2	3		Information Handling		1	1	1	1
6.1.2.1	4			The system shall inform the User when changes occur to the criteria upon which the pre-trip information had been given.				1
6.1.2.3	4			The system shall be able to provide information to all drivers including route restrictions, travel times, etc.		1		1
6.1.2.4	4			The system shall be able to support a database of events with links between events that occur concurrently and at the same or adjacent locations.		1		1
6.1.2.5	4			The system shall be able to analyse, process and retrieve data from different combinations of sources (including floating car).	1	1		1
6.1.2.6	4			The system shall be able to provide road and traffic information adapted to different classes of users, e.g. travelers, radio broadcasters, service operators.		1		
6.1.2.7	4			The system shall provide information using graphical representation or text. Graphical form shall include the use of maps as well as text.		1	1	1
6.1.2.8	4			The system shall provide information in the native language at the output location, and/or from a user selected choice of other appropriate foreign languages.		1	1	1
6.1.2.11	4			The system shall be able to provide information about "Points of Interest", e.g. location, opening times, price of service, nearest transport service points.		1		
6.1.2.12	4			The system shall be able to receive information about a point of interest from the providers/owners/managers of that "Point of Interest".		1		
6.1.3	3		Traveler Interaction			1		1
6.1.3.4	4			The system shall be able to provide access to reservations and pre-payment services.				
6.1.3.8	4			The system shall be able to provide customized pre-trip information to hand-held and in-vehicle devices.				1
6.2	2		On-trip Information		1	1	1	1
6.2.0	3		Objectives		1	1		1
6.2.0.1	4			The system shall provide emergency, or urgent, information to all users free of charge.				1
6.2.0.3	4			The system shall be able to be activated automatically by another system, e.g.	1			

					MADRID			
					UC1	UC2a	UC2b	UC3
Id	Level		Title	Description				
				traffic management.				
6.2.0.4	4			The system shall provide traffic information to the traveler during his/her trip in a timely manner, and include travel conditions, accidents, special events, car park status, etc.		1		1
6.2.0.5	4			The system shall be able to provide urban and inter-urban traffic and travel information to drivers about the domain they are not currently in.				
6.2.0.6	4			The system shall inform the User when changes occur to the criteria upon which the pre-trip information had been given.		1		1
6.2.0.7	4			The system shall be able to know where it is in the transport network, and hence provide the position of vehicle or person carrying it.		1		1
6.2.1	3		Mode Change			1		1
6.2.1.1	4			The system shall be able to provide alternative routes or mode-switch recommendations when it detects, or is informed, that problems have occurred on a mode		1		1
6.2.1.2	4			The system shall be able to display alternative routes or modes at inter-modal interchanges, or at places where tourism information is available.		1		1
6.2.1.3	4			The system shall be able to provide information about other transport modes: e.g. location of P+R areas, PT timetable, etc.		1		1
6.2.2	3		Information Handling		1	1	1	1
6.2.2.1	4			The system shall be able to inform travelers on the current average travel time between fixed points.		1		1
6.2.2.10	4			The system shall be able to collect data from a variety of different sources, e.g. road/traffic management, police, weather services, floating car etc.	1	1	1	1
6.2.2.14	4			The system shall be able to modify a travel plan if the traveler does not follow it.		1		1
6.2.2.2	4			The system shall be able to provide real-time P&R and PT information to vehicle drivers.		1		1
6.2.2.4	4			The system shall provide road and traffic safety advice based on current weather and traffic conditions.				
6.2.2.5	4			The system shall be able to provide information to all drivers including route restrictions, travel times, etc.		1		1
6.2.2.9	4			The system shall be able to adapt the information to different classes of users, e.g. travelers, radio broadcasters, service operators.	1			
6.2.3	3		Traveler Interaction		1	1	1	1
6.2.3.1	4			The system within the vehicle, or in the centre, shall support various types of presentation to the user.				1
6.2.3.2	4			The system shall normally provide messages from a finite set of well-defined messages.		1	1	1

					MADRID			
					UC1	UC2a	UC2b	UC3
Id	Level		Title	Description				
6.2.3.3	4			The system shall provide information in the native language at the output location, and/or from a user selected choice of other appropriate foreign languages, when applicable.		1	1	1
6.2.3.4	4			The system shall provide information using "open" standard communication protocols.	1	1	1	1
6.2.3.5	4			The system shall be able to provide customised on-trip information to hand-held and in-vehicle devices.		1		1
6.4	2		Route Guidance and Navigation		1	1	1	1
6.4.0	3		Objectives		1	1	1	1
6.4.0.1	4			The system shall provide travelers with recommended routes to specified destinations.		1		1
6.4.0.3	4			The system shall know where it is within the road network.	1	1	1	1
6.4.0.4	4			The system shall be able to modify its navigation instructions if an incorrect turn is made.		1		1
6.4.0.5	4			The system shall be able to provide a driver with a suitable alternative route, when the original planned route becomes unavailable.				1
6.4.1	3		Information Handling			1		1
6.4.1.2	4			The system shall be able to use real-time information to compute the recommended route.		1		1
6.4.1.3	4			The system shall be able to compute the total predicted journey time over the route selected.		1		1
6.4.1.4	4			The system shall be able to provide customised navigation information to the destination using a variety of selection criteria.		1		1
6.4.1.5	4			The system shall be able to provide guidance to "Points of Interest".		1		
6.4.1.6	4			The system shall provide information which is consistent with any other information being presented about the road.		1		1
6.4.2	3		Traveler Interaction			1		1
6.4.2.2	4			The system shall contain menus which are structured in a logical manner and oriented towards the requirements of the driver (e.g. the most frequently used function shall be the easiest to select).		1		1
6.4.2.4	4			The system shall enable the use of portable equipment to provide route guidance.		1		1
7.1	2		Traffic Control				1	1
7.1.0	3		Objectives		1		1	
7.1.0.10	4			The system shall be able to manage traffic in all or part of the road network using a methodology that is appropriate	1		1	

					MADRID			
					UC1	UC2a	UC2b	UC3
Id	Level		Title	Description				
				for urban roads, e.g. using traffic lights at junctions and with the possibility of incorporating facilities for pedestrians to cross the road in a controlled manner.				
7.1.1	3		Monitoring				1	1
7.1.1.1	4			The system shall be able to monitor sections of the road network to provide the current traffic conditions (e.g. flows, occupancies, speed and travel times etc.) as real time data.				1
7.1.1.2	4			The system shall monitor traffic in all or part of the road network that uses an "urban" style of road management, e.g. using traffic lights at junctions and with the possibility of incorporating facilities for pedestrians to cross the road in a controlled manner.			1	
7.1.1.4	4			The system shall be able to monitor traffic flow at, and the operation of, the road intersections of the network over which it has the control.				1
7.1.1.6	4			The system shall be able to monitor and record weather conditions (wind, fog, rain level, ice, etc.).				no
7.1.12	3		Vulnerable Road Users				1	
7.1.12.1	4			The system shall be able to control pedestrian and cycle crossings.			1	
7.1.12.2	4			The system shall be able to monitor and control pedestrian and cycle crossings in order to optimize their use.			1	
7.1.3	3		Traffic Control Centres		1		1	
7.1.3.4	4			The system shall be able to activate control devices (e.g. traffic lights, VMS), either individually or in groups.	1		1	
7.1.9	3		Adaptive Traffic Control		1			
7.1.9.2	4			The system shall be able to minimize delays of all vehicles using adaptive signal control				
7.1.9.3	4			The system shall be able to grant priority to selected vehicles (e.g. PT, emergency vehicles) at an intersection controlled by some form of traffic signals.	1			
7.2	2		Traffic Control		1		1	1
7.2.0	3		Objectives		1		1	
7.5	2		Cooperative Systems - Traffic Efficiency		1		1	1
7.5.1	3		Traffic Flow Optimization		1		1	
7.5.1.5	4			The system shall be able to manage the traffic in an area using a number of local semi-autonomous traffic management units, whose rules can be modified when required.	1		1	
7.5.2	3		Advanced Adaptive		1			

					MADRID			
					UC1	UC2a	UC2b	UC3
Id	Level		Title	Description				
			Traffic Signals					
7.5.2.2	4			The system shall enable the driver of a host vehicle to request a series of green phases from traffic signals (i.e. a green wave) for the route that is about to be taken.	1			
7.5.2.3	4			The system shall enable a traffic signal controller to receive a request for a green phase from an approaching vehicle; in the event that more than one conflicting request is received at the same time they shall be prioritised (e.g. emergency vehicles before private vehicles), possibly by the TCC operator.	1			
7.5.2.5	4			The system shall enable the traffic signal controller to determine the expected arrival time of a vehicle at the junction using data received from that vehicle (e.g. current location and speed profile, estimated time of arrival).	1			
7.5.2.8	4			The system shall enable a traffic signal controller that has received a green phase request to inform downstream controllers that a green wave vehicle is approaching.	1			
8.12	2		Vehicle Priority Management		1			
8.12.0	3		Basic Services		1			
8.12.0.2	4			The system shall enable the green wave to apply to a single controlled junction, a single segment of the road network, or multiple combinations of these.	1			
10.1	2		Public Transport Management		1			
10.1.0	3		Objectives		1			
10.1.0.1	4			The system shall provide effective and attractive PT.	1			

Table 1 Madrid preliminary user needs

4.2 Genoa Pilot

4.2.1 Preliminary User Needs

					GENOA	
					UC1	UC2
Id	Level		Title	Description		
6.1.3	3		Traveler Interaction		1	
6.1.3.8	4			The system shall be able to provide customised pre-trip information to hand-held and in-vehicle devices.	1	
6.2	2		On-trip Information			

					GENOA	
					UC1	UC2
Id	Level		Title	Description		
6.2.0	3		Objectives		1	1
6.2.0.4	4			The system shall provide traffic information to the traveler during his/her trip in a timely manner, and include travel conditions, accidents, special events, car park status, etc.	1	
6.2.0.5	4			The system shall be able to provide urban and inter-urban traffic and travel information to drivers about the domain they are not currently in.	1	
6.2.0.6	4			The system shall inform the User when changes occur to the criteria upon which the pre-trip information had been given.	1	
6.2.0.7	4			The system shall be able to know where it is in the transport network, and hence provide the position of vehicle or person carrying it.	1	1
6.2.1	3		Mode Change		1	
6.2.1.1	4			The system shall be able to provide alternative routes or mode-switch recommendations when it detects, or is informed, that problems have occurred on a mode	1	
6.2.1.2	4			The system shall be able to display alternative routes or modes at inter-modal interchanges, or at places where tourism information is available.	1	
6.2.1.3	4			The system shall be able to provide information about other transport modes: e.g. location of P+R areas, PT timetable, etc.	1	
6.2.2	3		Information Handling		1	1
6.2.2.1	4			The system shall be able to inform travelers on the current average travel time between fixed points.	1	
6.2.2.10	4			The system shall be able to collect data from a variety of different sources, e.g. road/traffic management, police, weather services, floating car etc.	1	1
6.2.2.11	4			The system shall be able to provide operators with an overall view of all active events in an area.	1	
6.2.2.12	4			The system shall provide Information Management tools for the operator.	1	1
6.2.2.13	4			The system shall be able to provide information to vehicle drivers in case of medical emergency, e.g. location of rest areas, medical assistance, etc.		
6.2.2.14	4			The system shall be able to modify a travel plan if the traveler does not follow it.	1	
6.2.2.2	4			The system shall be able to provide real-time P&R and PT information to vehicle drivers.	1	
6.2.2.3	4			The system shall be able to provide cyclists and pedestrians with information about suitable routes.	1	
6.2.2.4	4			The system shall provide road and traffic safety advice based on current weather and traffic conditions.	1	
6.2.2.5	4			The system shall be able to provide information to all drivers including route restrictions, travel times, etc.	1	
6.2.2.6	4			The system shall be able to provide routing information for Commercial traffic to/from a (cargo) modal interchange.		
6.2.2.7	4			The system shall be able to support a database of events with links between events that occur concurrently and at the same or adjacent locations.	1	

					GENOA	
					UC1	UC2
Id	Level		Title	Description		
6.2.2.8	4			The system shall be able to provide road information according to different geographic scales, e.g. local, regional, national, international.	1	
6.2.2.9	4			The system shall be able to adapt the information to different classes of users, e.g. travelers, radio broadcasters, service operators.	1	
6.2.3	3		Traveler Interaction		1	
6.2.3.1	4			The system within the vehicle, or in the centre, shall support various types of presentation to the user.	1	
6.2.3.2	4			The system shall normally provide messages from a finite set of well-defined messages.	1	
6.2.3.3	4			The system shall provide information in the native language at the output location, and/or from a user selected choice of other appropriate foreign languages, when applicable.	1	
6.2.3.4	4			The system shall provide information using "open" standard communication protocols.	1	
6.2.3.5	4			The system shall be able to provide customised on-trip information to hand-held and in-vehicle devices.	1	
6.4	2		Route Guidance and Navigation			
6.4.0	3		Objectives		1	1
6.4.0.1	4			The system shall provide travelers with recommended routes to specified destinations.	1	
6.4.0.2	4			The system shall not base its decisions on a restricted sub-set of the road network, e.g. motorways only.	1	
6.4.0.3	4			The system shall know where it is within the road network.	1	1
6.4.0.4	4			The system shall be able to modify its navigation instructions if an incorrect turn is made.		
6.4.0.5	4			The system shall be able to provide a driver with a suitable alternative route, when the original planned route becomes unavailable.	1	
6.4.1	3		Information Handling		1	1
6.4.1.1	4			The system shall be able to provide guidance to Car Parks (with parking spaces).	1	
6.4.1.2	4			The system shall be able to use real-time information to compute the recommended route.	1	
6.4.1.3	4			The system shall be able to compute the total predicted journey time over the route selected.	1	
6.4.1.4	4			The system shall be able provide customised navigation information to the destination using a variety of selection criteria.	1	
6.4.1.5	4			The system shall be able to provide guidance to "Points of Interest".	1	
6.4.1.6	4			The system shall provide information which is consistent with any other information being presented about the road.	1	
6.4.1.7	4			The system shall be able to provide reports on the effectiveness of the navigation instructions that have been provided.		1
6.4.2	3		Traveler Interaction		1	
6.4.2.2	4			The system shall contain menus which are structured in a logical manner and oriented towards the requirements of the driver (e.g. the most frequently used function shall be the easiest to select).	1	

					GENOA	
					UC1	UC2
Id	Level		Title	Description		
6.4.2.4	4			The system shall enable the use of portable equipment to provide route guidance.	1	

Table 2 Genoa preliminary user needs

4.2.2 Refined User Needs. New and updated user needs

The following table describes the changes derived from the updated user needs. Only the affected (updated) rows from the previous user needs table are included here below.

					GENOA		Update	Reason for update
					UC1	UC2		
Id	Level		Title	Description			Status	
6.2	2		On-trip Information					
6.2.0	3		Objectives		1	1		
6.2.0.7	4			The system shall be able to know where it is in the transport network, and hence provide the position of vehicle or person carrying it.	1	1	The user need is also applicable to UC2	The user feedback (Use Case 2) shall include the details on user position. This has to be automatically retrieved
6.2.2	3		Information Handling		1	1		
6.2.2.10	4			The system shall be able to collect data from a variety of different sources, e.g. road/traffic management, police, weather services, floating car etc.	1	1	The user need is also applicable to UC2	
6.2.2.12	4			The system shall provide Information Management tools for the operator.	1	1	The user need is also applicable to UC2	
6.4	2		Route Guidance and Navigation					
6.4.0	3		Objectives		1	1		
6.4.0.3	4			The system shall know where it is within the road network.	1	1	added for both S1 and S2	Please see also the note on the previous table about this need. The user position (with the meaning and approximation mentioned in the previous note) should be known for

					GENOA		Update	Reason for update
					UC1	UC2		
Id	Level		Title	Description			Status	
								both S1 and S2
6.4.0.4	4			The system shall be able to modify its navigation instructions if an incorrect turn is made.			deleted	Not foreseen
6.4.1	3		Information Handling		1	1		
6.4.1.7	4			The system shall be able to provide reports on the effectiveness of the navigation instructions that have been provided.		1	added to S2	It has been added supposing that the meaning of the need is "there shall be a way to formulate a feedback on the quality of the journey planning output" which is exactly the objective of S2.

Table 3 Genoa refined user needs

4.3 Tampere Pilot

4.3.1 Preliminary User Needs

					TAMPERE			
					UC1	UC2	UC3	UC4
Id	Level		Title	Description				
2.1	2		Transport Planning Support		1			
2.1.2	3		Planning		1	1		
2.1.2.2	4			The system shall be able to develop and implement traffic environmental management strategies based on current and predicted traffic conditions.		1		
2.1.2.3	4			The system shall be able to assist in the planning of (inter-modal) routes.	1			
2.1.4	3		Reporting		1			
2.1.4.1	4			The system shall collect and report data as required by legally appointed authorities.	1			
2.2.2	3		Monitoring		1			1
2.2.2.1	4			The system shall be able to receive infrastructure equipment status data remotely.	1	1		1
6.1	2		Pre-trip information		1	1		1
6.1.0	3		Objectives		1			1

					TAMPERE			
					UC1	UC2	UC3	UC4
Id	Level		Title	Description				
6.1.0.1	4			The system shall provide emergency, or urgent, information to all road users free of charge.	1			1
6.1.0.3	4			The system shall be able to provide accurate, credible, timely, and easy to comprehend traffic and travel information where it may be of benefit to the user.	1			1
6.1.0.4	4			The system shall be able to provide information on alternative routes where they are quicker, cheaper, shorter, scenic, etc.	1			1
6.1.0.5	4			The system shall enable travelers to plan their trip using their own travel criteria (modes of transport, time of departure/arrival, road selection criteria, etc.).	1			1
6.1.1	3		Modal Choice		1			1
6.1.1.4	4			The system shall be able to provide extensive multi-modal trip information, e.g. prices, fares, routes, forecast & current traffic situations, traffic control, demand management measures, local warnings, special events, weather conditions, hotels etc.	1			1
6.1.2	3		Information Handling		1	1		1
6.1.2.3	4			The system shall be able to provide information to all drivers including route restrictions, travel times, etc.	1			1
6.1.2.7	4			The system shall provide information using graphical representation or text. Graphical form shall include the use of maps as well as text.	1			1
6.1.2.8	4			The system shall provide information in the native language at the output location, and/or from a user selected choice of other appropriate foreign languages.	1			1
6.1.2.10	4			The system shall be able to provide access information for those travelers with special needs (e.g. physical access, lifts, escalators, parking & toilets, nappy changing rooms, access for (guide) dogs, etc.) at relevant areas, e.g. transit areas.	1			1
6.1.2.13	4			The system shall be able to provide information to travelers so as to influence their choice of destination and/or mode of travel, e.g. to protect the environment of a "Point of Interest", or geographic area.		1		
6.2	2		On-trip information		1			1
6.2.0	3		Objectives		1		1	1
6.2.0.1	4			The system shall provide emergency, or urgent, information to all users free of charge.	1		1	1

					TAMPERE			
					UC1	UC2	UC3	UC4
Id	Level		Title	Description				
6.2.0.4	4			The system shall provide traffic information to the traveler during his/her trip in a timely manner, and include travel conditions, accidents, special events, car park status, etc.	1		1	1
6.2.1	3		Mode Change		1			1
6.2.1.1	4			The system shall be able to provide alternative routes or mode-switch recommendations when it detects, or is informed, that problems have occurred on a mode	1			1
6.2.1.2	4			The system shall be able to display alternative routes or modes at inter-modal interchanges, or at places where tourism information is available.	1			
6.2.1.3	4			The system shall be able to provide information about other transport modes: e.g. location of P+R areas, PT timetable, etc.	1			
6.2.2	3		Information Handling		1		1	1
6.2.2.10	4			The system shall be able to collect data from a variety of different sources, e.g. road/traffic management, police, weather services, floating car etc.	1		1	1
6.2.2.3	4			The system shall be able to provide cyclists and pedestrians with information about suitable routes.	1			
6.2.2.4	4			The system shall provide road and traffic safety advice based on current weather and traffic conditions.	1			
6.2.2.5	4			The system shall be able to provide information to all drivers including route restrictions, travel times, etc.	1			1
6.2.2.7	4			The system shall be able to support a database of events with links between events that occur concurrently and at the same or adjacent locations.			1	
6.2.2.8	4			The system shall be able to provide road information according to different geographic scales, e.g. local, regional, national, and international.	1			1
6.2.3	3		Traveler Interaction		1		1	1
6.2.3.1	4			The system within the vehicle, or in the centre, shall support various types of presentation to the user.			1	
6.2.3.2	4			The system shall normally provide messages from a finite set of well-defined messages.			1	
6.2.3.3	4			The system shall provide information in the native language at the output location, and/or from a user selected choice of other appropriate foreign languages, when applicable.			1	1
6.2.3.4	4			The system shall provide information using "open" standard communication protocols.	1		1	1
6.2.3.5	4			The system shall be able to provide	1		1	1

					TAMPERE			
					UC1	UC2	UC3	UC4
Id	Level		Title	Description				
				customised on-trip information to hand-held and in-vehicle devices.				
6.4	2		Route-guidance and navigation		1			1
6.4.0	3		Objectives		1			1
6.4.0.1	4			The system shall provide travelers with recommended routes to specified destinations.	1			1
6.4.0.2	4			The system shall not base its decisions on a restricted sub-set of the road network, e.g. motorways only.	1			1
6.4.0.3	4			The system shall know where it is within the road network.	1			1
6.4.0.5	4			The system shall be able to provide a driver with a suitable alternative route, when the original planned route becomes unavailable.	1			1
6.4.1	3		Information Handling		1			1
6.4.1.1	4			The system shall be able to provide guidance to Car Parks (with parking spaces).	1			1
6.4.1.3	4			The system shall be able to compute the total predicted journey time over the route selected.	1			1
6.4.1.4	4			The system shall be able provide customised navigation information to the destination using a variety of selection criteria.	1			1
6.4.1.6	4			The system shall provide information which is consistent with any other information being presented about the road.	1			1
6.4.2	3		Traveler Interaction		1			
6.4.2.2	4			The system shall contain menus which are structured in a logical manner and oriented towards the requirements of the driver (e.g. the most frequently used function shall be the easiest to select).	1			
7.1	2		Traffic Control		1			1
7.1.1	3		Monitoring		1			1
7.1.1.6	4			The system shall be able to monitor and record weather conditions (wind, fog, rain level, ice, etc.).	1			1
7.1.11	3		Parking Management					1
7.1.11.1	4			The system shall be able to monitor the current usage of the parking facilities.				1
7.1.2	3		Planning		1		1	1
7.1.2.1	4			The system shall be able to use consistent historical data to complement real-time data, when necessary.	1		1	1
7.1.2.8	4			The system shall be able to produce new traffic management strategies from one or more of	1			1

					TAMPERE			
					UC1	UC2	UC3	UC4
Id	Level		Title	Description				
				historic, current, or predicted road traffic data.				
7.3	2		Demand Management		1			1
7.3.0	3		Objectives		1			1
7.3.0.2	4			The system shall receive up-to-date information on those factors that will influence the demand management strategy, e.g. traffic levels, car park usage, other modes usage, fares, tolls, etc.	1			1
10.1	2		Public Transport Management		1			
10.1.4	3		Information Handling		1			
10.1.4.1	4			The system shall be able to inform travelers about PT operations for a mode of transport, e.g. travel times, delays, fares etc.	1			
10.1.4.2	4			The system shall be able to provide information about a PT service to the travelers before and during the journey.	1			
10.1.4.3	4			The system shall be able to provide an update of arrival/departure information in real-time and present it to travellers of that mode before and during the journey.	1			
10.2	2		PT Planning					
10.2.1	3		Information Handling		1			
10.2.1.9	4			The system shall enable the traveler to specify any special needs that he or she may have, e.g. disability, young children, etc.	1			
10.4			On-trip PT information		1			
10.4.0	3		Objectives		1			
10.4.0.1	4			The system shall be able to inform travelers about all PT operations (including bus, rail, metro, air, taxi, car pooling etc.).	1			
10.4.1	3		Information Handling		1			
10.4.1.1	4			The system shall be able to provide in-vehicle general (dynamic) PT information, as well as the arrival time at, and name of, the next stop for this vehicle.	1			
10.4.1.2	4			The system shall be able to provide general (dynamic) PT information, personal safety information, as well as the arrival times of next vehicles, delays, etc. at mode interchanges, e.g. bus stops, in metro, railway or bus stations.	1			
10.4.2	3		Traveler Interaction		1			
10.4.2.1	4			The system shall provide information which is legible, understandable and capable of	1			

					TAMPERE			
					UC1	UC2	UC3	UC4
Id	Level		Title	Description				
				being assimilated very quickly by all travelers.				
10.4.2.2	4			The system shall provide information in the native language at the output location, and/or from a user selected choice of other appropriate foreign languages, when applicable.	1			

Table 4 Tampere preliminary user needs

5 Functional Viewpoint

5.1 Consolidated Functional Viewpoint

Following the FRAME methodology and after an analysis of the obtained results, the next functional areas have been identified as relevant:

- F1. User Management and Application Tailoring (formerly Provide Electronic Payment Facilities)
- F3. Manage Traffic
- F4. Manage PT Operations
- F6. Provide Traveler Journey Assistance
- F10. EC/CF Estimation (new)

The following table shows the cross-check between the identified functionalities and MoveUs high-level functions identified in D2.2.

Function ID	Name	MADRID	GENOA	TAMPERE	MoveUs	MoveUs High-level Function (D2.2)	Group	Sub-group
1	User Management and Application Tailoring							
1.2	Manage Personal Accounts					PER		
1.3	Manage User Identities (Perform Electronic Payment Transaction)							
1.3.1	Manage User identity	1	1	1	1	REG		
1.3.2	Check User Identity (Authentication)	1	1	1	1	LOG		
1.5	Access Control							
1.5.1	Check User's Rights (Authorization)	1	1	1	1	REG		
1.6	Manage Tariffs and Access Rights							
1.6.2	Manage Access Rights					LOG		
1.7	Manage Register Data					REG		
1.8	Manage Electronic Wallet		1	1	1	PAY		
3	Manage Traffic							
3.1.1	Provide Urban Traffic Management	1			1			
3.1.1.5	Provide Urban Traffic Management Facilities	1			1	CTL		
3.1.1.5.22	Output Stop&go Commands to Urban Roads	1			1	CTL		
3.1.1.5.24	Implement Urban Traffic Strategies			1	1	X		
3.1.1.6	Manage Urban Static Traffic Data	1			1	TOP		
3.1.1.8	Collect Urban Data from Vehicles	1			1	SEN	Urban Traffic	
3.1.1.9	Output Urban Traffic Data	1			1	GET	Urban Traffic	

Function ID	Name	MADRID	GENOA	TAMPERE	MoveUs	MoveUs High-level Function (D2.2)	Group	Sub-group
3.1.1.10	Collect Urban Traffic Data	1			1	SEN	Urban Traffic	
3.1.1.11	Provide Updated Urban data for Digital Maps	1			1	GET	Urban Traffic	
3.1.1.13	Predict Short & Medium Term Urban Conditions	1			1	SEN	Urban Traffic	Prediction
3.1.1.14	Manage Urban Traffic Data	1	1	1	1	TOP/SEN		
3.1.2.16	Manage Inter-urban Traffic Data		1	1	1	SEN	Inter-urban Traffic	
3.1.2.9	Output Inter-urban Traffic Data		1		1	GET	Inter-urban Traffic	
3.1.4	Provide Management of Car Parks				1	GET	Parking	
3.1.4.1	Monitor numbers of vehicles in Car Parks			1	1	X		
3.1.4.2	Detect the occupancy of Car Park spaces			1	1	X		
3.1.4.4	Calculate Car Park Occupancy and Status			1	1	X		
3.1.4.8	Manage Urban Parking Data Store			1	1	SEN	Parking	
3.1.4.9	Output Car Park Information to Drivers		1	1	1	GET	Parking	
3.1.6	Provide Traffic Predictions				1	SEN	Traffic	Prediction
3.1.6.3	Create Traffic Predictions with Simulation Methods			1	1	SEN	Traffic	Prediction
3.1.6.4	Manage Traffic Prediction Data Store			1	1	SEN	Traffic	Prediction
3.1.6.5	Provide Traffic Predictions Operator Interface			1	1	SEN	Traffic	Prediction
3.1.6.6	Process Traffic Prediction Results			1	1	SEN	Traffic	Prediction
3.2.10	Manage Store of Incident Data		1		1	SEN	Incidences	
3.2.11	Provide Operator Interface for Incident Management		1		1	X		
3.4.2	Monitor Atmospheric Pollution				1	SEN	Enviromental	
3.4.3	Monitor Noise Pollution				1	SEN	Enviromental	
3.4.10	Output Environmental Information		1	1	1	GET	Enviromental	
3.4.11	Analyse Environmental Data and Implement Actions			1	1	SEN	Enviromental	
3.4.4	Predict Environmental Conditions				1	SEN	Enviromental	Prediction
3.4.8	Manage Environmental Conditions Data Store			1	1	SEN	Enviromental	
4	Manage PT Operations							
4.1.12	Output Service Information to Travelers			1	1	GET	PT Transport	Services
4.1.5	Collect PT Vehicle data	1			1	SEN	PT Transport	
4.1.6	Predict Vehicle Timings	1		1	1	SEN	PT Transport	

Function ID	Name	MADRID	GENOA	TAMPERE	MoveUs	MoveUs High-level Function (D2.2)	Group	Sub-group
4.2.5	Manage Fare Schemes				1	SEN	PT Transport	Fares
4.2.6	Manage Fare Schemes data store				1	SEN	PT Transport	Fares
4.2.7	Manage PT Route Data Store			1	1	SEN	PT Transport	Routes
4.6.1	Provide Car Pooler Interface	1			1	POL		
4.6.3	Manage Bike Hiring Information	1			1	POL		
6	Provide Traveler Journey Assistance							
6.3.10	Implement Trip Plan and Track Traveler		1	1	1	ONT		
6.3.11	Monitor Trip Plan Implementation for Traveler		1	1	1	ONT		
6.3.12	Manage revised trip plan creation to traveler				1	ONT/POS		
6.3.13	Provide Traveler Trip Interface		1	1	1	GET	Journey Plan	
6.5.10	Provide Traveler Trip Planning Interface		1	1	1	PREI		
6.5.3.11	Provide Green Wave Routes		1	1	1	PRET		
6.5.3.13	Provide Data & Routes to Fleet Operators & Drivers		1	1	1	GET	Journey Plan	
6.5.3.3	Collect PT Data	1	1	1	1	SEN	PT Transport	
6.5.3.7	Enable Operator Access to Trip Planning Data		1		1	GET	Journey Plan	
6.5.3.8	Collect Data About Road Traffic	1	1	1	1	SEN	Traffic	
6.5.3.9	Plan Trip Details	1	1	1	1	PRET		
6.6.1	Provide Traveler Information Interface	1	1	1	1	GET	Travel Information	
6.6.2	Produce Travel Information	1	1	1	1	PRET		
6.6.3	Output Travel Information	1	1	1	1	GET	Travel Information	
6.6.4	Manage travel information data store				1	PRET		
6.6.5	Provide Travel information operator interface				1	GET	Travel Information	
6.7.1	Define Traveler's General Trip Preferences	1	1	1	1	PER		
6.7.2	Evaluate Trip After Completion	1	1	1	1	POS		
6.7.3	Enable operator access to trip information				1	GET	Travel Information	
6.7.4	Manage General Trip Preference				1	PER		
6.8.1	Manage Store of trip plan data				1	ONT, POS		
10	EC/CF Estimation							
10.1	Energy Consumption Calculation			1	1	ECC		
10.2	Carbon Footprint			1	1	ECC		

Function ID	Name	MADR ID	GEN OA	TAMPERE	MoveUs	MoveUs High-level Function (D2.2)	Group	Sub-group
	Calculation							

Table 5 Mapping FRAME function and MoveUs HL Functions

Below, for each Functional Area, a DFD (of level 3) is shown, identifying external actors, functions (in green those belonging to other areas), datastores and dataflows. For User Management and Application Tailoring, a description of the provided facilities for each function is given; for the rest of the functions, this information will be integrated and aligned with High-level functions.

Higher-size pictures of the DFDs shown below can be found in Annex 1.

- **F1. User Management and Application Tailoring** (formerly Provide Electronic Payment Facilities)

  This High-level Function shall provide the facilities needed to manage the users and application register and tailoring according specific city deployment needs.

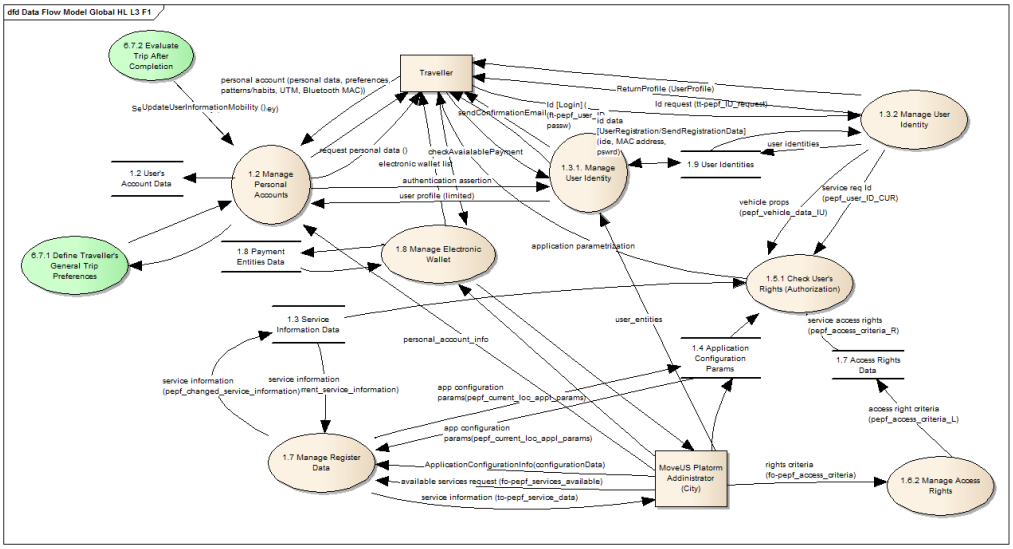


Figure 4 User Management and Application Tailoring DFD

1.2 Manage Personal Account
(1) The ability to apply different operations to user personal information: Add, Modify, Delete and Consult, being applicable to different content, from personal data, preferences, patterns/habits, energy consumption, and mobile identification.

1.7 Manage Register Data
(1) The ability to manage the content of the store of Service Information Data and Application Configuration Parameters.
(2) The management of the store shall enable a MoveUs platform administrator to review the contents of the store, to add, delete, or amend the contents.

1.8 Manage Electronic Wallet (to coordinate with Incentive Management)

(1) The ability to keep alive a register of banking or payment institutions available for any specific user.
(2) The management of the information shall enable a MoveUs platform administrator to review the contents of the store, to add, delete, or amend the contents.

1.3.1 Manage User Identity
(1) The ability to manage the content of the store of User identities, being available for future authentication processes.

1.3.2 Check User Identity (User Identify)
(1) The ability to access to Use Identities file and authenticate the access.
1.5.1 Check Users Rights (Authorization)
(1) The ability to determine the access rights of users to the service(s) that they have requested based on their profile and
(2) The ability to use the ID of the user, or the characteristics of the Vehicle they are using, and traffic conditions in comparison with the access criteria obtained from the store of Access Rights Data to determine the users' access rights. In particular, dealing with "eco-friendly cars" with access to green lines.
(3) The ability to guide user and/or the check user's rights functionality, depending on the origin of the request.
(4) The ability to select and send the configuration parameters and constraints to MoveUs Application, according its situation.

1.6.2 Manage Access Rights
(1) The ability to manage the store of "access rights" Data based on user profile, application deployment (localization) and available services.

- **F3. Manage Traffic**

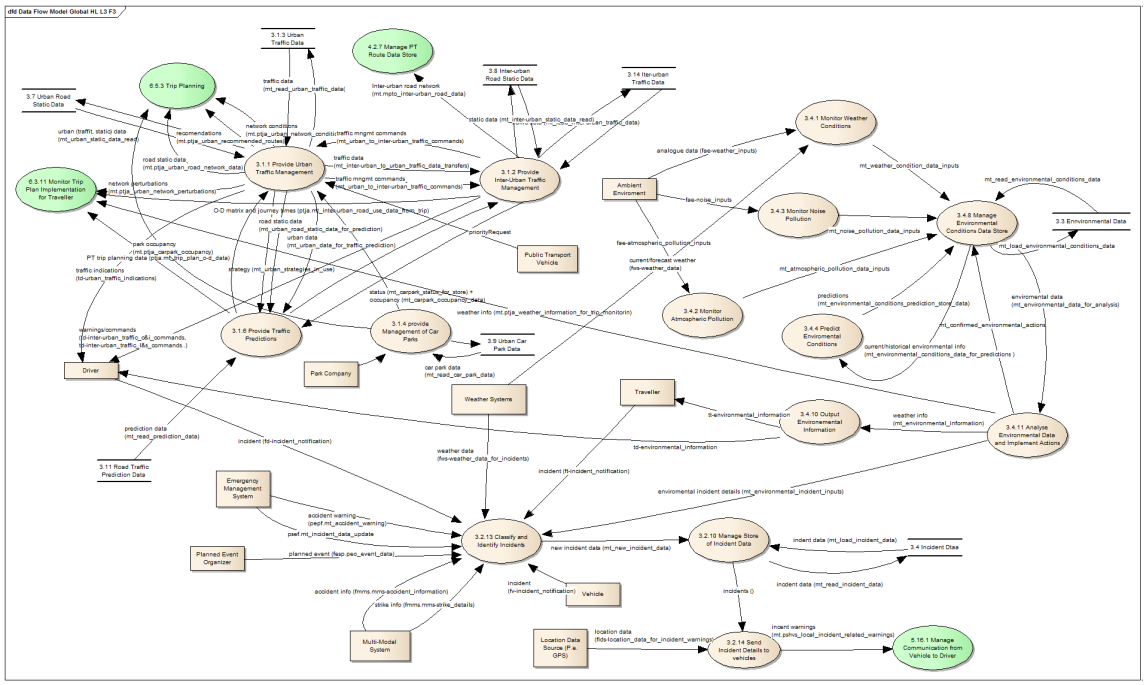


Figure 5 Traffic Management DFD

- **F4. Manage PT Operations**

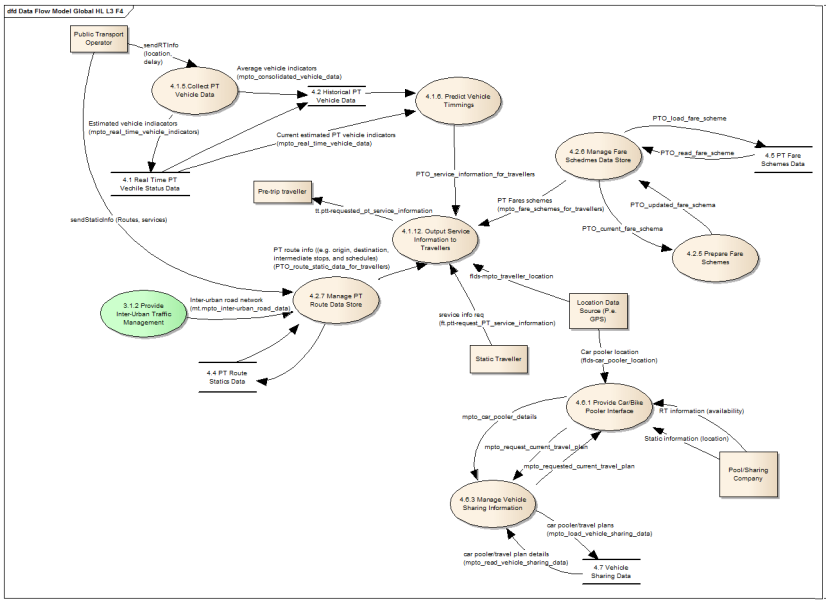


Figure 6 PT Operation Management DFD

- **F6. Provide Traveler Journey Assistance**

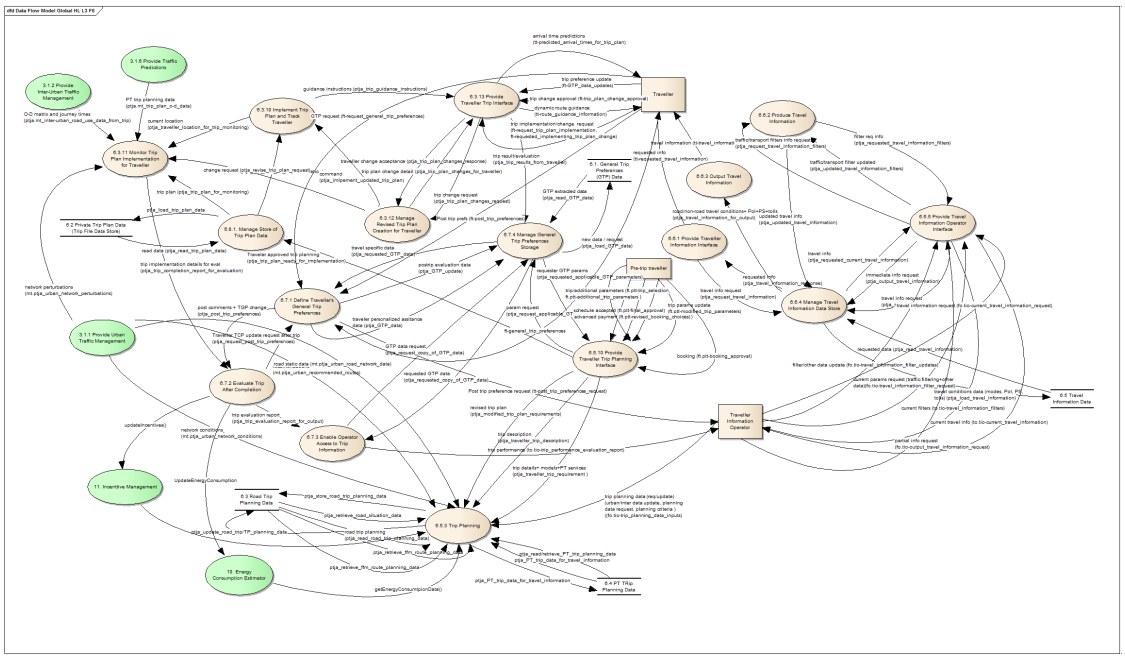


Figure 7 Traveler Journey Assistance DFD

A preliminary mapping between High-level Functional Components (WP2) to FRAME Functional Diagrams (DFDs of level 2/3) has been performed. The following table shows this preliminary mapping providing an idea of which DFDs are relevant for the different use-cases. In addition, relations and exchanges between high-level functions are identified.

				MADRID				GENOA		TAMPERE			
			FRAME DFDs	UC 1	UC 2a	UC 2b	UC 3	UC 1	UC 2	UC 1	UC 2	UC 3	UC 4
REG	Registration		1.7, 1.3.1	X	X	X	X	X	X	X	X	X	X
LOG	Login (Authentication and Authorization)		1.3.2, 1.6.2, 1.5.1,	X	X	X	X	X	X	X	X	X	X
PER	Personal Account Management		1.2, 6.7.1, 6.7.4		X	X	X	X	X	X	X	X	X
PREI	Pre-Trip. Request Trip Planning (Routing)	Trip Information & Request	6.5.10		X		X	X		X	X	X	X
PRET		Trip Planning (Computation)	6.5.3 (11,9), 6.6.2, 6.6.4		X		X	X		X			
GET	Get Information		3.1.1 (9,11), 3.1.2 (9), 3.1.4 (9), 3.2.14, 3.4.10, 4.1.12, 6.3.13, 6.5.3 (7,13), 6.6.1, 6.6.3, 6.6.5, 6.7.3		X		X	X					
POL	Car/Bike Pooling/ Sharing	Trip Execution	4.6.1, 4.6.3		X			X					
ONT	On-Trip		6.3.10, 6.3.11, 6.3.12, 6.8.1		X		X						
PAY	Electronic Wallet Registry	Post-Trip	1.8					X					
POS	Post-Trip	Rules management	6.7.2, 6.8.1	X	X	X	X	X	X	X	X		
INCM	Incentive Management	Incentive management	11.1, 11.2, 11.3, 11.4, 11.5, 11.6		X			X	X			X	
SEN	Traffic / Mobility Capture		3.1.1 (8,10,13), 3.1.2 (16), 3.1.4 (8), 3.1.6 (3,4,5,6), 3.2.10, 3.2.13, 3.4.1, 3.4.2, 3.4.3, 3.4.11, 3.4.4, 3.4.8, 4.1.5, 4.1.6, 4.2.5, 4.2.6,	X	X		X	X	X	X			

				MADRID				GENOA		TAMPERE			
			FRAME DFDs	**UC 1**	**UC 2a**	**UC 2b**	**UC 3**	**UC 1**	**UC 2**	**UC 1**	**UC 2**	**UC 3**	**UC 4**
			4.2.7, 6.5.3 (3,8), 3.4.2, 3.4.3, 3.4.1, 3.4.10, 3.4.11, 3.4.12, 3.4.13, 3.4.4, 3.4.8										
CTR	Traffic Control		3.1.1.5 (22)	X		X			X				
TOP	Device/ Topology Configuration		3.1.1 (6,14)	X		X	X						
ECC	CF/EC Calculation		10.1, 10.2		X	X		X			X	X	

Table 6 Mapping between FRAME functions and MoveUs HL functions per pilot

- Relationships between High-level Functionalities

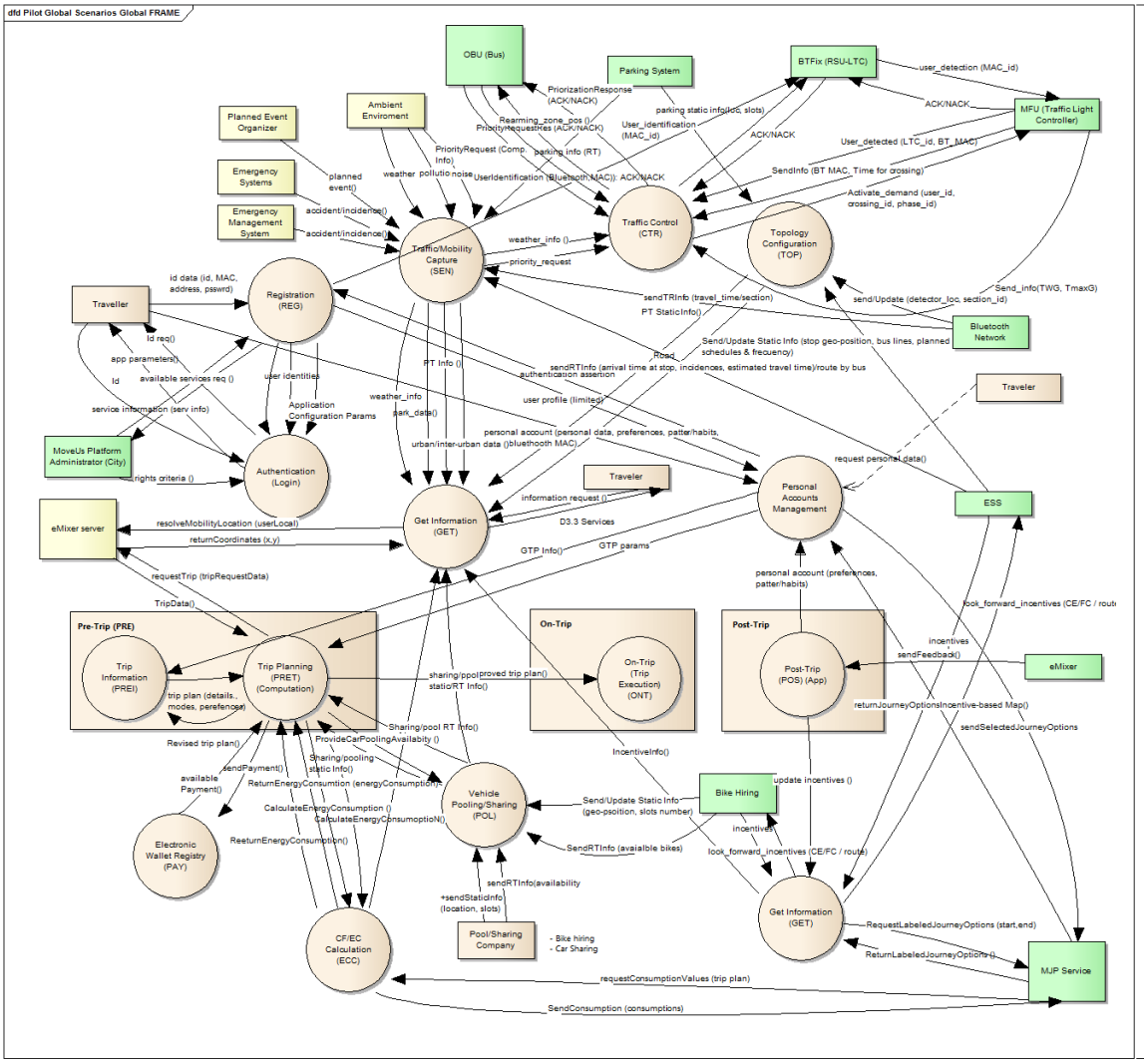


Figure 8 MoveUs High-level functionalities dependences

5.1.1 Pilot Actors and interfaces

In this section, actors and communications for the different project pilots are depicted, as displayed in the following pictures.

5.1.1.1 Madrid Pilot

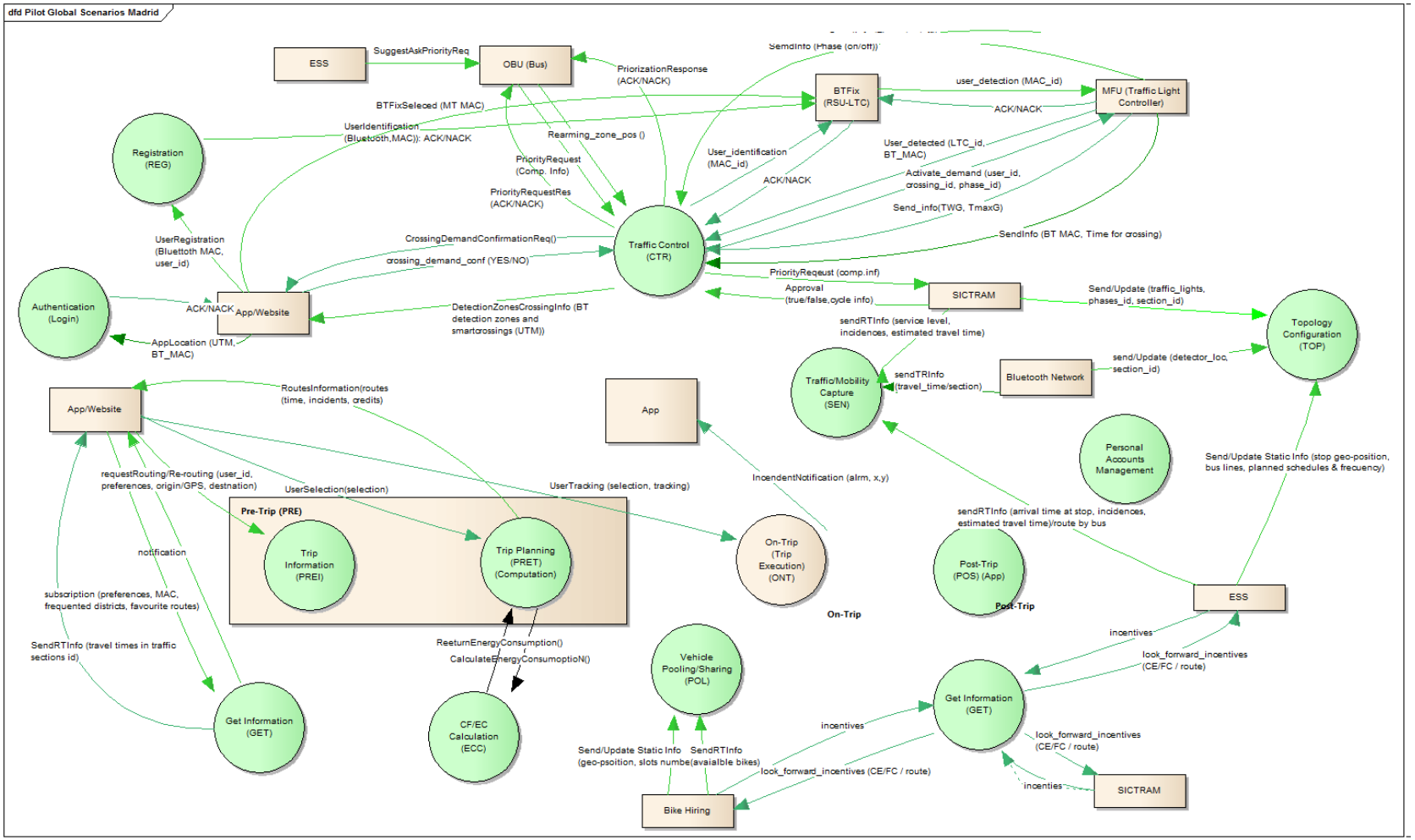


Figure 9 Madrid Pilot actors and functionalities

5.1.1.2 Genoa Pilot

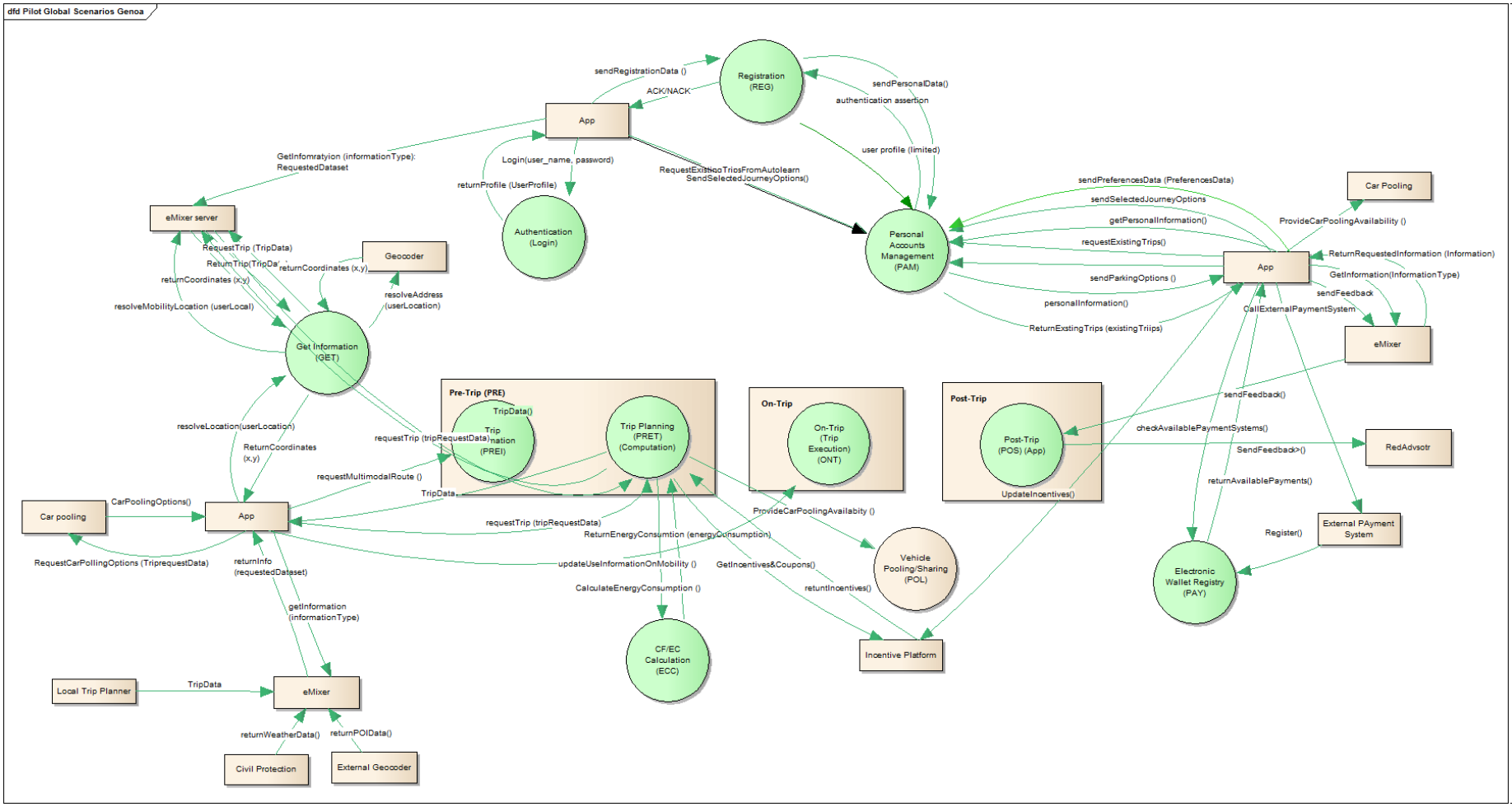


Figure 10 Genoa Pilot actors and functionalities

5.1.1.3 Tampere Pilot

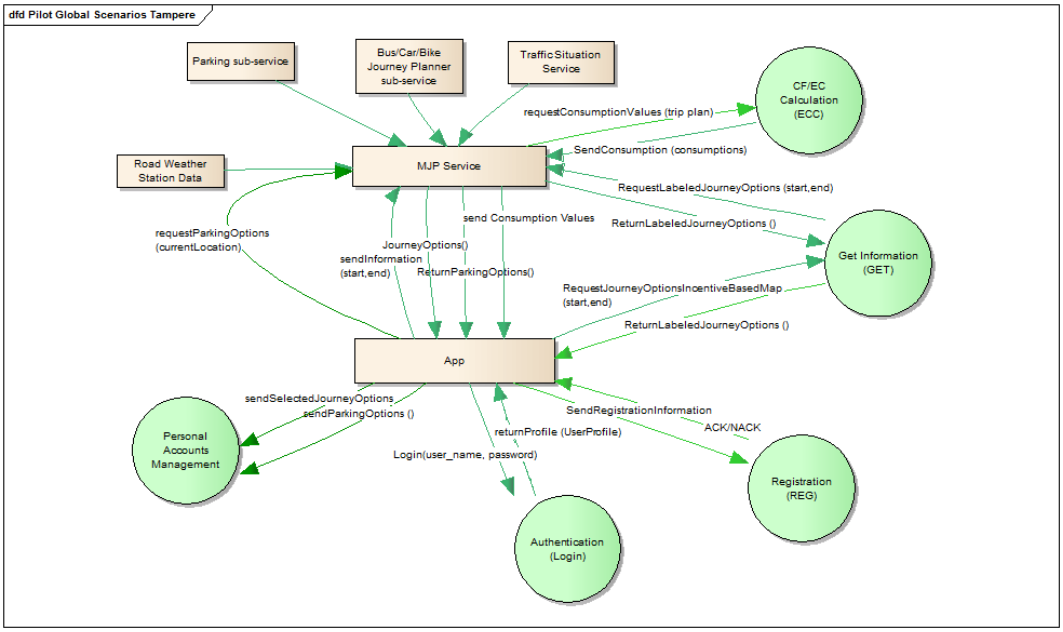


Figure 11 Tampere Pilot actors and functionalities

6 Technical / Application Architecture

6.1 MoveUs reference Service Model

In parallel to the work done around the functional definition of MoveUs platform (Section 5.1), and aligned with the common approach to deal with Service-Oriented Architectures (SOA), a definition of external services is presented below.

MoveUs service catalogue is based on previous experiences as In-Time, Co-cities and in origin, eMotion. Specifically, two main categories of services are defined:

1. Services aimed at providing access to mobility data and services available at the input sources (LDSSs and other DSSs).
2. Services of general interest and related to the management and operation of the platform and deployed applications (here city services): as (meta-) services such as configuration, register and application tailoring, subscriptions, service catalogues, etc.

In addition, specifically for MoveUs platform, we identified a third category:

3. Functional, whose representative services not only serve as data gateways between providers and consumers, but also implement underlying complex operations such as: multi-modal planning, incentive management, traffic regulation, predictions or data-fusion processes in order to aggregate heterogeneous data sources for traffic estimation.

Inspired on the service taxonomy defined by eMotion [1], Figure 12 below shows the existing and functionally equivalent service types highlighted with a red box for In-Time project, a blue box for Co-Cities project and a green box for MoveUs specific services. All the services deployed in the context of MoveUs platform appear shaded in grey, elicited from the Use cases defined in D2.2 [6] and the project Description of Work.

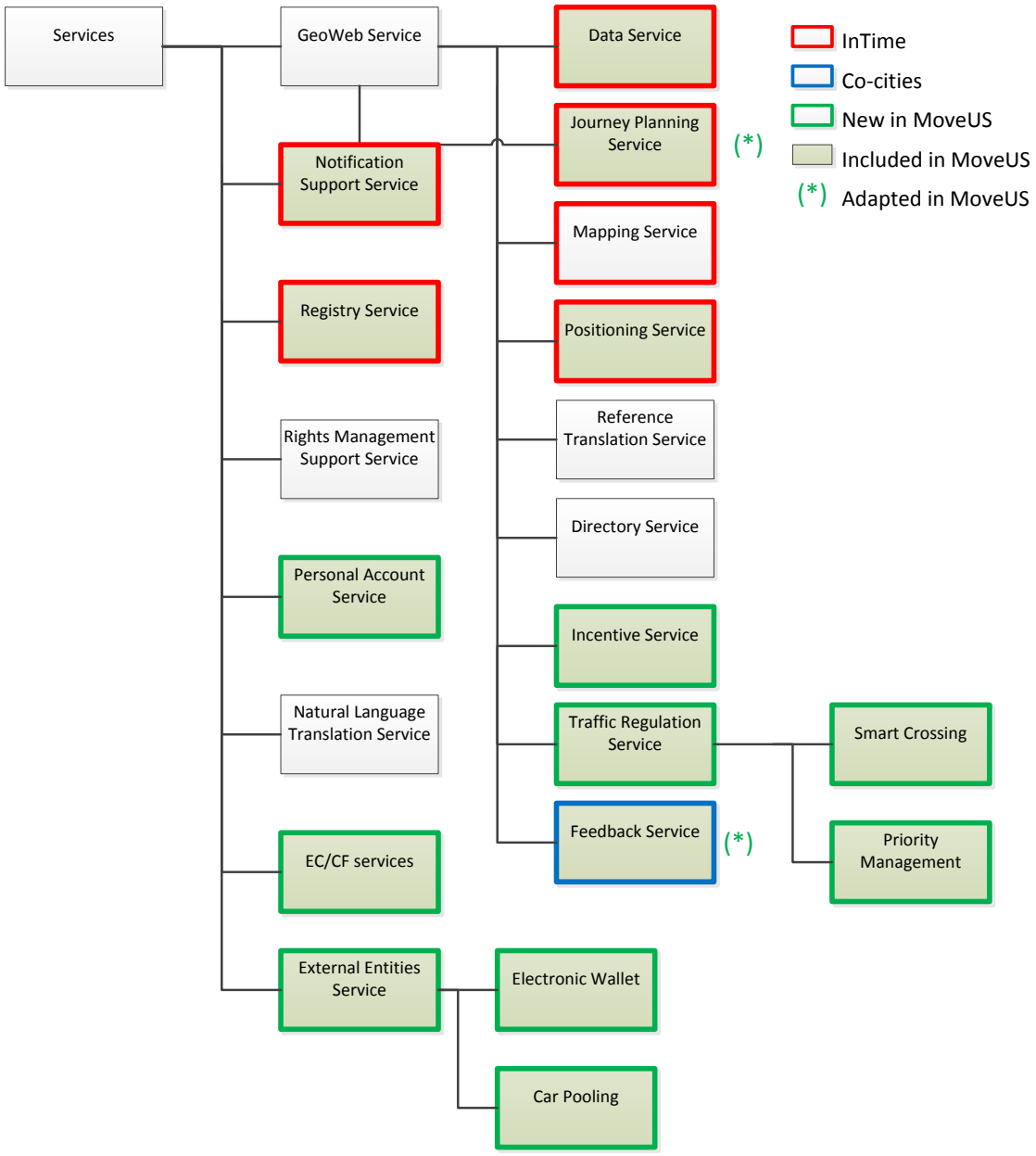


Figure 12 Services taxonomy and In-Time/Co-Cities/MoveUs service categories (red/blue/green boxes respectively)

Overall, the MoveUs service architecture includes the following main services types:

- Data Service/Journey Planning Services

  Static Road Traffic Information, Dynamic Road Traffic Information, Static Parking Information, Static Public Transport Information, Walking Information, Dynamic Road Traffic Routing Information, Dynamic Public Transport Information, Dynamic Public Transport Journey Routing, Dynamic Parking Information, Dynamic Walking Planning, Dynamic Cycling Planning, Multi Modal Journey Planning

- Positioning Services

  Resolve Mobility Location, Resolve Address

- Notification Support Services

  Subscribe database element, Subscribe Habitual Routes, Subscribe Frequent PoI

- Registry Services

  Request for available services, Set application configuration options, Request for application parameters, Request route preference criteria

- Personal Account Services

  Mobility personal data, Personal Data, General Preferences, User preferences, personal energy consumption data, Preferred/Habitual Trips Learning, Personal favourite PoI, User Mobility Information

- Incentive Services

  Rules Management, Currency Management, Incentive Management, Incentive Payment Type Management, Awards Management, Coupon Management, Advertisement Management, Incentive Information Access, Request of Awards, Request of Coupons, Access Voucher, Administration

- Traffic Regulation Services

  Smart Crossing and Priority Management Services

- Feedback Services

  Traffic Quality Feedback, Journey Planning Quality Feedback, Traffic Feedback

- Energy Consumption /& Carbon Footprint Estimation

  Route Energy Consumption Estimation, Current Energy Consumption, Historical Energy Consumption, period Historical Energy Consumption, Energy Incentives, Energy Efficiency Tips (currently under definition in WP4)

- External Entities Services

  Request for available external payment systems and car pooling services

The detailed description of the underlying function operation is described in section 6.3. Also, a more detailed description of the currently defined services can be found in Annex 2: MoveUs Platform Services, identifying inputs, outputs and syntaxes. At this stage, the detailed services corresponds to those used by the MoveUs City Services (defined in D3.3 [8]); for those identified in use-cases (defined in D2.2 [6]), a deeper analysis is needed and it will be subject of Task 5.1. Something similar happens with some parameters as UserId, that whether or not explicit depending on the selected communication (service) technology (e.g. WS, WEB-API, WEB-Sockets). Finally, energy efficiency services are still under definition (WP4), being available for the next version of this document.

Some relevant discussions on certain issues led to the following statements:

- MoveUs platform is primarily a SOA- Service-Oriented Architecture, so we adopt or at least consider at this stage most of the relevant standards as e.g. XML, WSDL, SOAP. However, given the implicit overhead of SOAP in terms of performance and traffic communication bandwidth, particularly striking in a cloud-based scenario, other options such as REST-full technologies or Web Services must be carefully evaluated.

- About Geographic Information standards, it is commonly accepted the compliance with ISO/TC 211 and OGC (Open Geospatial Consortium) standards. However, on the other hand, another important reference is the OSM- Open Street Map model (http://www.openstreetmap.es/). This option is determined by its capabilities as a gateway (import / export) between many other map formats (e.g. GML, Google Earth (Keyhole) Markup Language (KML), GPX, SVG) and available supporting tools in most cases.

6.2 System/Subsystem/Module/Component View

The definition of the technical architecture has tried to keep in mind the common FRAME High-level Functions, but adapted and extended according to MoveUs specific requirements.

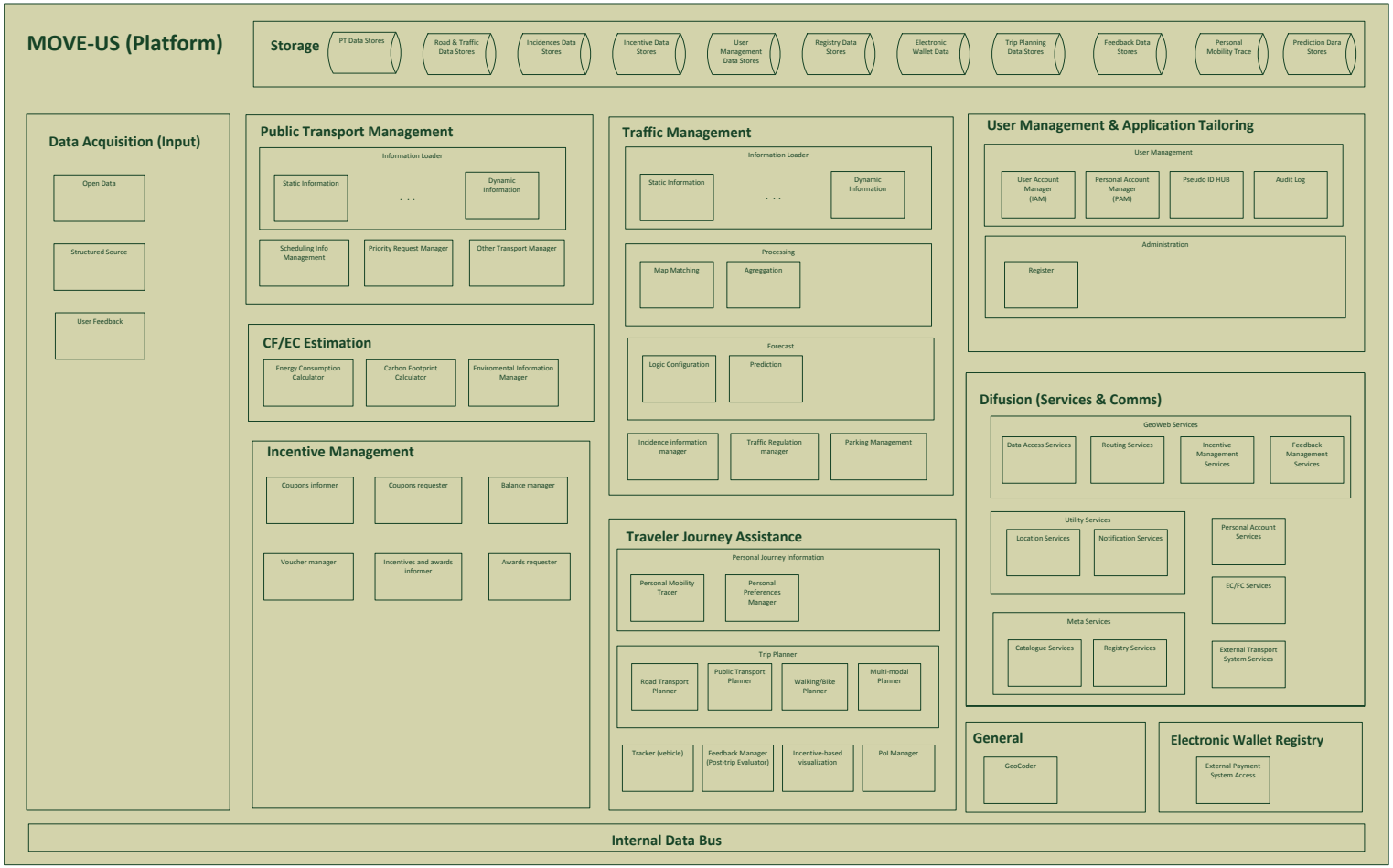


Figure 13 MoveUs Technical Architecture

6.2.1 Functional mapping

In the following table, functions and data stores are mapped into architectural modules.

System/ Subsystem	Module	Comp	HL func	FRAME Function	FRAME sub-function	FRAME Datastore / MoveUs datamodel package (D3.1)	Madrid (Platf/Loc)	Genoa (Platf/Loc)	Tampere (Platf/Loc)
Personal Account Management	Identify Access Manager (IAM)		LOG	1.3 Perform Electronic Payment Transactions	1.3.2 Check User Identity	User Identities / credentials	X	X	X
			REG		1.3.1 Manage User Identity	User Identities / credentials	X	X	X
			LOG	1.5 Access and Credit Control	1.5.1 Check User's rights	Access Rights data / credentials	X	X	X
			LOG	1.6 Manage Tariffs and Access Rights	1.6.2 Manage Access rights	Access Rights data / credentials	X	X	X
	Logger		LOG	1.5 Access and Credit Control	1.5.1 Check User's rights	Access Rights data / credentials	X	X	X
	Personal Account Manager (PAM)		PAM	1.2 Manage Personal Accounts		User Management / User Account Data	X	X	X
	Electric Wallet Register		PAY	1.8 Manage Electronic Wallet	1.8.1 Electronic Wallet List	Payment Entity Data /		X	
			PAY		1.8.2. External Payment	Payment Entity Data / Incentives.eWallet Service Registry		X	
Administration	Register		REG	1.7 Manage Register Data		Service Information, Application Configuration Parameters (Registry)	X	X	X
Data Acquisition	Open Data		Platform			Specific Data Adaptors			
	Structured Information		Platform			Specific Data Adaptors	X	X	X
	User Feedback		Platform			Specific Data Adaptors	X	X	

System/ Subsystem	Module	Comp	HL func	FRAME Function	FRAME sub-function	FRAME Datastore / MoveUs datamodel package (D3.1)	Madrid (Platf/Loc)	Genoa (Platf/Loc)	Tampere (Platf/Loc)
Public Transport Management	Information Loader	Static Information	TOP	6.5.3. Trip Planning	6.5.3.3. Collect PT Data	PT Static Data (PT ServiceModel, PT Line ManagementModel)	X		
		Dynamic Information	SEN (PT)	6.5.3. Trip Planning	6.5.3.3. Collect PT Data	PT Dynamic Data (PT ServiceModel, PT Line ManagementModel)	X		
			SEN (PT)	4.2 Plan PT Service	4.2.6 Manage fare schemes data store	PT Fare Schema (PT ServiceModel, PT Line ManagementModel)			
			SEN (PT)	4.1 Monitor PT Fleet	4.1.6 Predict vehicle timings	PT Dynamic Data (PT ServiceModel, PT Line ManagementModel)	X		
	Scheduling Manager		SEN (PT)		4.2.7 Manage PT route data store	PT Static Data /PT Trip PlanningData / Priority Request Register (PT ServiceModel, PT Line ManagementModel)	X		
	Priority Request Manager		SEN (PT)	6.5.3 Trip Planning	6.5.3.11. Provide Green Wave Routes	Urban Road Static data (I/O) (Urban Road Static Data (RoadDataModel, TrafficRegulation.StaticModel.BT and J235)) Urban Road Dynamic Data (I)(O)(MeasuredDataPublication)	X		
	Other transport manager (sharing/renting)		SEN (PT)	4.6 Provide facilities for vehicle sharing	4.6.1 Provide car pooler interface	Vehicle Sharing Data (I/O) (CarPoolingService)	X	X	
					4.6.3 Manage vehicle sharing information	Vehicle Sharing Data (I/O) (CarPoolingService)		X	
CF/EC Estimation	Energy Consumption Calculator		ECC		10.1 Energy Consumption Calculus	Energy Efficiency. KPIs (I) Energy Efficiency.Energy labels (I) Energy Efficiency.Energy Affecting parameters (I)	X	X	X
	Environmental Information		SEN (ENV)	3.4 Provide Environmental Information	3.4.1 Monitor Weather Conditions	Incidence.Weather Data (I)(O) Event.RoadWeatherAndEnvirome			

System/ Subsystem	Module	Comp	HL func	FRAME Function	FRAME sub-function	FRAME Datastore / MoveUs datamodel package (D3.1)	Madrid (Platf/Loc)	Genoa (Platf/Loc)	Tampere (Platf/Loc)
	Manager					ntEvent			
			SEN (ENV)		3.4.2 Monitor Atmospheric Pollution	Incidence. Weather Data (I)(O) Event.RoadWeatherAndEnvironmentEvent			
			SEN (ENV)		3.4.8 Manage Environmental data store	Incidence. Weather Data (I)(O) Event.RoadWeatherAndEnvironmentEvent			
			GET		3.4.10 Output Environmental information	Incidence. Weather Data (O) Event.RoadWeatherAndEnvironmentEvent			
Incentive Management	Coupons informer		INC	11. Incentive Management	11.1 Information of Coupons/Deals	Incentive.Coupons/Deals (I)	X	X	X
	Coupons requester		INC	11. Incentive Management	11.2 Request of Coupons/Deals	Incentive.Coupons/Deals (I)(O)	X	X	X
	Balance manager		INC	11. Incentive Management	11.3 Balance Management	Incentives. Incentive Balance (O)	X	X	X
	Voucher manager		INC	11. Incentive Management	11.4 Voucher Management	Incentives .Vouchers (I)(O)	X	X	X
	Incentives and awards informer		INC	11. Incentive Management	11.5 Information on Incentives & Awards	Incentives .Incentives (I) Incentives .Awards catalogue (I)	X	X	X
	Awards requester		INC	11. Incentive Management	11.6 Request of Awards		X	X	X
Traffic Management	Information Loader	Information Loader- Static Information	TOP	3.1.1 Provide urban Traffic Management	3.1.1.6 Manage Urban Static Traffic Data	Urban Road Static data (I/O) (Urban Road Static Data (RoadDataModel, TrafficRegulation.StaticModel.BT and J235))	X		
			SEN (TRAF)		3.1.1.14 Manage urban traffic data	Urban Road Static data (I/O) (Urban Road Static Data (RoadDataModel, TrafficRegulation.StaticModel.BT and J235))	X		

System/ Subsystem	Module	Comp	HL func	FRAME Function	FRAME sub-function	FRAME Datastore / MoveUs datamodel package (D3.1)	Madrid (Platf/Loc)	Genoa (Platf/Loc)	Tampere (Platf/Loc)
		Information Loader-Dynamic Information	SEN (TRAF)	3.1.1 Provide urban Traffic Management	3.1.1.10 Collect urban traffic data	Urban Road Dynamic Data (I)(O)(MeasuredDataPublication)	X		
			SEN (TRAF)		3.1.1.6 Collect urban traffic data from vehicles	Urban Road Static data (I/O) (Urban Road Static Data (RoadDataModel)	X		
			SEN (TRAF)		3.1.1.14 Manage urban traffic data	Urban Road Static data (I/O) (Urban Road Static Data (RoadDataModel, TrafficRegulation.StaticModel.BT and J235))	X		
		Information Loader-Dynamic Information	SEN (TRAF)	3.1.2 Provide inter-urban Traffic Management	3.1.2.10 Collect inter-urban traffic data	Inter-Urban Road Dynamic Data (I)(O) (MeasuredDataPublication)	X		
			SEN (TRAF)		3.1.2.6 Collect inter-urban traffic data from vehicles	Inter-Urban Road Dynamic Data (I)(O) (MeasuredDataPublication)	X		
			SEN (TRAF)		3.1.2.16 Manage interurban traffic data	Inter-Urban Road Dynamic Data (I)(O) (MeasuredDataPublication)	X		
	Processing	Map Matching	SEN (TRAF)		3.1.2.16 Manage interurban traffic data		X		
		Aggregation	SEN (TRAF)		3.1.2.16 Manage interurban traffic data	Urban/Inter-urban Traffic Data (I/O) Aggregated Road Data (O)	X		
	Forecast	Logic Configuration	SEN (TRAF)	3.1.6. Provide Traffic Predictions	3.1.6.5 Provide Traffic Predictions Operator Interface		X		
			SEN (TRAF)		3.1.6.6. Process Traffic Prediction Results		X		
		Prediction	SEN (TRAF)	3.1.6. Provide Traffic Predictions	3.1.6.1 Process Road Network Static Data	Urban Road Static data (I/O) (Urban Road Static Data (RoadDataModel)		X	
			SEN (TRAF)		3.1.6.3. Create Traffic Predictions	Aggregated Road History Data (I)(O), Aggregated Road Data (I) ,		X	

System/ Subsystem	Module	Comp	HL func	FRAME Function	FRAME sub-function	FRAME Datastore / MoveUs datamodel package (D3.1)	Madrid (Platf/Loc)	Genoa (Platf/Loc)	Tampere (Platf/Loc)
						Road Traffic Prediction Data (O) (MeasuredDataPublication)			
			SEN (TRAF)		3.1.1.13 Predict short & medium term urban conditions	Road Traffic Prediction Data (O) (MeasuredDataPublication)		X	
			SEN (TRAF)		3.1.6.4. Manage Traffic Prediction Data Store	Road Traffic Prediction Data (I/O) (MeasuredDataPublication)		X	
	Parking Information Manager		SEN (PARK)	3.1.4 Provide Management of Car Parks	3.1.4.8 Manage Urban Parking Data Store	Urban Car Park Data (I/O), PoI Data (I)(O) (CarParking, EMotionFeature.PointOfInterest))			X
	Traffic Regulation Manager (Smart Crossing)		CTR	3.1.1.5 Provide urban Traffic management facilities	3.1.1.5.22. Output stop&go commands to urban roads	Smart Crossing Register (O), Urban Road Static data (I/O) (Urban Road Static Data (RoadDataModel, TrafficRegulation.StaticModel.BT and J235))	X		
	Incidence Information Manager		SEN (INC)	3.2 Manage Incidences	3.2.10 Manage store of Incidence Data	Incidence Data (I/O) (EMotionFeature.Incidence)	X	X	
				3.3.1 Receive Information on Travel Factors	3.1.1.11 Provide Updated traffic data for digital maps	Urban/Inter-urban Traffic Data (I/O) Aggregated Road Data (O)	X		
Traveler Journey Assistance	Personal Journey Information	Personal Mobility Tracer	ONT	6.7.3 Enables operator Access To Trip Information		Personal Mobility Trace Data (I/O) (JP_LegTracks, Personal Mobility Trace)			
		Personal Preferences Manager	PAM	6.7 Manage General Trip Preferences	6.7.1. Define travelers' General trip preferences	General Trip Preferences (GTP) data (I/O) (GeneralTripPreferences)	X	X	X
					6.7.4. Manage General Trip Preferences Storage	General Trip Preferences (GTP) data (I/O) (GeneralTripPreferences)	X	X	X

System/ Subsystem	Module	Comp	HL func	FRAME Function	FRAME sub-function	FRAME Datastore / MoveUs datamodel package (D3.1)	Madrid (Platf/Loc)	Genoa (Platf/Loc)	Tampere (Platf/Loc)
	Trip Planner	Road Transport Planner	PREI/P RET	6.5. Prepare Trip Plan		Urban/Inter-urban Traffic Data (I/O) Aggregated Road Data (O)	X		
		Public Transport Planner	PREI/P RET			PT Static Data / PT Trip Planning Data / PT Fare Schema (PT ServiceModel, PT Line ManagementModel)	X		
		Walking/Bike Planner	PREI/P RET			Urban/Inter-urban Traffic Data (I/O) Aggregated Road Data (O)	X		
		Multi-modal Planner	PREI/P RET	6.5.3 Trip Planning	6.5.3.9 Plan Trip Details	Trip Plan Data (O) (Traveller Journey Assistance JP_Journey)	X		
			PREI/P RET		6.5.3.13 Provide Data & Routes to Fleet Operators & Drivers	Road Trip Planning Data (O)	X		
			PREI/P RET	6.6 Provide Traveler Information	6.6.1 Provide Traveler Information Interface	Trip Plan Data (O) (Traveller Journey Assistance JP_Journey)	X		
			PREI/P RET		6.6.2 Produce Traveler Information	Trip Plan Data (O) (Traveller Journey Assistance JP_Journey)	X		
			GET		6.6.3 Output Travel Information	Trip Plan Data (O) (Traveller Journey Assistance JP_Journey)	X		
			PREI/P RET		6.6.4. Manage Travel Information Data Store	Trip Plan Data (O) (Traveller Journey Assistance JP_Journey)	X		
			PREI/P RET		6.6.5 Provide Travel Information Operator Interface	Trip Plan Data (O) (Traveller Journey Assistance JP_Journey)	X		
			PREI/P RET	6.8. Manage Trip Plans	6.8.1 Manage Trip Store of Trip Plan Data	Trip Plan Data (Traveler JP_Journey)	X		
	Vehicle Tracker		ONT	6.3. Support Trip	6.3.10. Implement Trip Plan and Track Traveler	Trip Plan Data (Traveller Journey Assistance JP_Journey)	X		
			ONT		6.3.11. Monitor Trip Plan Implementation for Traveler	Trip Plan Data (Traveller Journey Assistance .JP_Journey)	X		
			ONT		6.3.12. Manage Revised Trip Plan Creation for Travelers	Trip Plan Data (Traveller Journey Assistance .JP_Journey)	X		

System/ Subsystem	Module	Comp	HL func	FRAME Function	FRAME sub-function	FRAME Datastore / MoveUs datamodel package (D3.1)	Madrid (Platf/Loc)	Genoa (Platf/Loc)	Tampere (Platf/Loc)
	Post-trip Evaluation Manager		POS	6.7.2. Evaluate Trip After Completion		Feedback Data (O) (Feedback)	X	X	X
	PoI Manager		GET	6.6 Provide Traveler Information		PoI Data (I)(O) (PointOfInterest (EMotionFeature.PointOfInterest), BikeHiringPoint)	X	X	X
Diffusion (Services & Comms)	Data Services	Data Access Services	GET						
		Routing Services	GET				X		
		Trip Planning Interface	GET				X		
		Incentive Management Services	GET				X	X	X
		Feedback Services	POS				X	X	
	Utility Services	Location Services	GET					X	
	Meta Services	Catalogue Services	GET						
		Registry Services	GET				X	X	X
General	GeoCoder		GET					X	

Table 7 Mapping between functions and data stores to architectural modules

6.3 Module Detailed Description

This section provides a detailed description of the main modules and components, specifically targeting the following aspects for each module:

- Define exactly the users that interact with the module
- Enumerate and briefly describe the data stores or schemas that interact with the module
- Introduce the components that integrate the module
- Depict a functional view of the interactions between actors and modules
- Provide description of supported functions, operations and/or services

A deep description of the data capture, diffusion and subscription/notification functionalities will be included in the next version of the document.

6.3.1 MoveUs User Management & Application Tailoring

Definitions

- **Tailoring.** Customization of application (here, city services) according local and user needs.
- **Authentication.** Verification of the user identity.
- **Authorization.** Check Users Rights.

Users

- **Type 4 (UT4)**: Final users
  - **UT4_Anonymous**: User that wants to keep their identity anonymous
  - **UT4_Identified**: Users that provide their information to the MoveUs platform
- **Type 8 (UT8)**: Field devices and External Traffic/Service Systems
  - **UT8_TLC**: Traffic Light Controllers
- **Type 9 (UT9)**: MoveUs Platform Administrator / City Administrator
- **Type X (UTX)**: Any registered user

Functional View

The functional view of the User Management & Application Tailoring Module is depicted in the following figure.

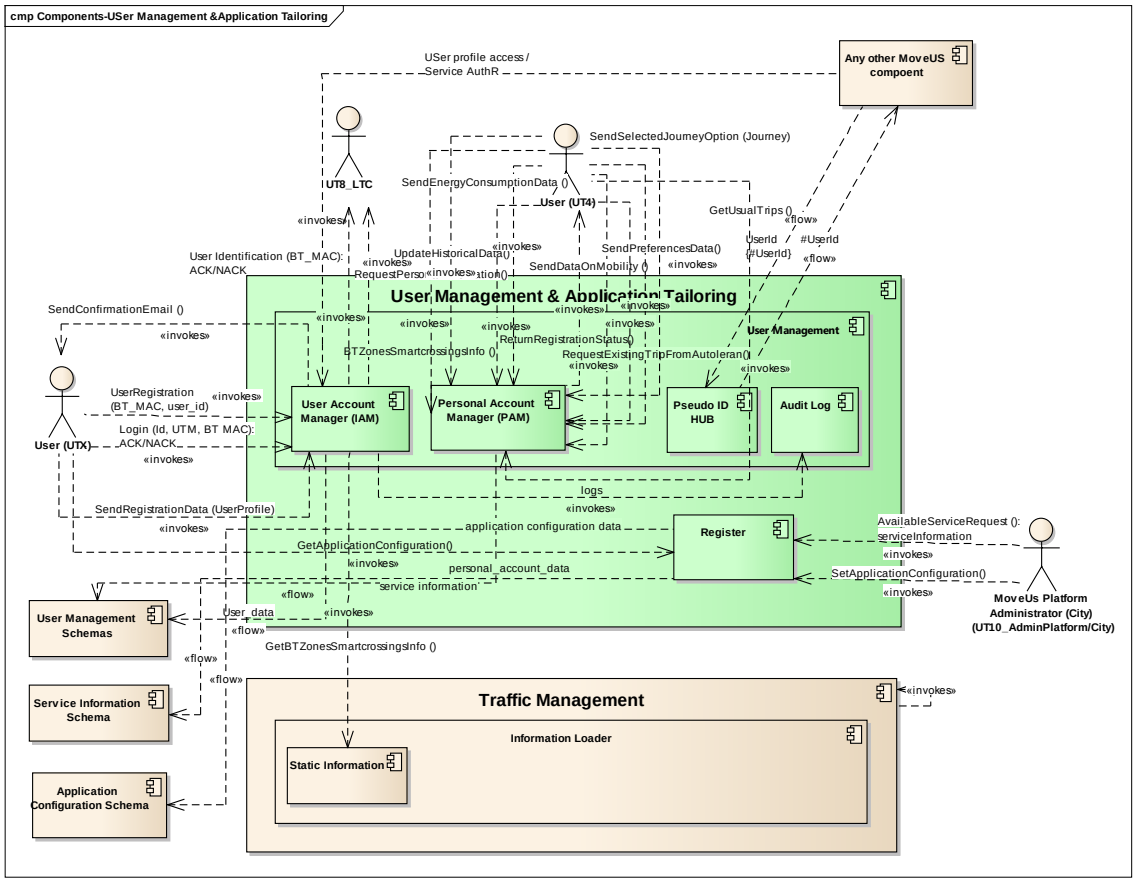


Figure 14 User Management & Application Tailoring functional view (in green)

MoveUs Database - high level definition

Credentials	Registry of basic registration data: • UserId, • DeviceId, • Password
Users	Registry of users described by: • ID • User Type • Name • Personal Data • Personal Preferences Data • Mobility Data
Service Information	Information of the available platform services, as a whole, locally or personalized according user profile.
Application Configuration	Application parameterization to direct the dynamic tailoring and customization of deployed city services.

Table 8 User Management datastores

MoveUs Modules

User Management	
User Account Manager (IAM)	Manages the user identity from the registration and then authenticating the user.
Pseudo Id Hub	Manages the association between the individual identification and the user data.
Audit Log	Manages a registry of the access to personal data.
Personal Account Manager (PAM)	Manages the user's mobility personal profile: personal data, preferences and habits
Administration	
Register	Manages the information for activating the proper services according to the city from which the user is accessing the system

Table 9 User Management modules

Functions/Operations/services

1. **User Management**

   a. **User Account Manager (IAM)**

   The independent Identity & Access Management (IAM) component will be responsible for establishing users' identity during registration and then authenticating the user.

   i. User Registration

   A user can be registered by providing his/her identity, Device Id, and password. The steps are the following:

   1. The user starts the MoveUs App & obtains the MCC from mobile SIM to connect to the IAM or the user´s home country IAM (in case of a distributed solution).
   2. Authentication to IAM with User Id, Device Id and password, being UserId native or email address.
   3. The IAM verifies the registration data, checking if user already exists and information format.
   4. If successful, the user data is stored in the credentials structure and the IAM returns an authentication assertion, used later to services/component access by means of the local PI Hub. An additional eMail is sent to the new user. In case of failure, it returns an error.

   - UserRegistration (Id, Device Id, Passwd)

   ii. Login

   1. Each time a user accesses to MoveUs services logs introducing the login information (user, password).
   2. The IAM will perform all authorizations based on Policy Agent checks so that if a user does not have incentives then he/she will not be allowed to access this area or if an administrator does not have access permissions to the stored location data store then he/she will not be granted access.
   3. The component checks the user and password; if successful it returns ok, otherwise, it returns an error.

   - Login (user_name, Passwd)

   iii. Password Change

   A user can change his/her own password at any time.

   1. The user requests the change of password, providing the old and the new one.
   2. The system checks if current password matches, verifies the correctness of the new password and updates the user credentials structure to the new one.
   3. In case of failure, it returns an error.

- ChangeUserPassword (UserId, OldPasswd, Passwd)

iv. Password Recovery

A user can request the recovery of his/her account, depending of specific implementation; this request will be reported via eMail or specific message.

1. The user requests the password recovery
2. The notification arrives to MoveUs platform
3. If userId is not conformed and only eMail is available, search on user register (MV_User_Basics.eMail)
4. The component resets automatically (or manually by the platform administrator) the password, generates a new random one and notifies the user to upgrade.

- PasswordRecovery (UserId) (message) or
- eMail (with specific subject as "Password Recovery")

b. Pseudo Id Hub

User's sensitive information should be stored with pseudo identifiers, avoiding, or at least doing harder, the bi-univocal association between the data and the individual identification. This component only accepts requests from authenticated servers (digital certificate).

i. Generation and link of user IDs hashes

The Pseudo Identity (PI) Hub manages the correlation between registered UserId and Pseudo IDs used for storing its information. It is recommended that the PI Hub generates different #User Ids per Data Store in the cloud e.g. #Loc_User-Id, #Inc_User-Id thus making it more difficult to correlate and link personal data in different data stores that could be potentially hacked in the cloud.

ii. Updating Index Algorithms

Operator could add in a process to update a hub's indexing algorithms so that the hash function could be upgraded or have an evaluation of change policy once a year as better hash functions become available.

iii. Pseudo Identities (PI) management

Two different behaviours can be identified:

- If a user wants to access to own information (incentives, mobility trace, and personal data) he/she logs in and then is connected to the desired component. The functional component would request the PI HUB to translate the user´s User ID to the Component #Comp_User-Id to access his/her Incentives account.

- If a user needs to refer to other user (e.g. to give his incentives to a friend or a Type 2 or Type 3 entity wants to add or remove an incentive to/from a user´s account) he/she would need to use the real User ID of the individual and the component would hash this to the actual user´s Component pseudo ID (#Comp_user-Id).

c. Audit Log

A user mobility record and personal data event log is kept so that only the owner can see all the actions (by whom and when) that have taken place on their account.

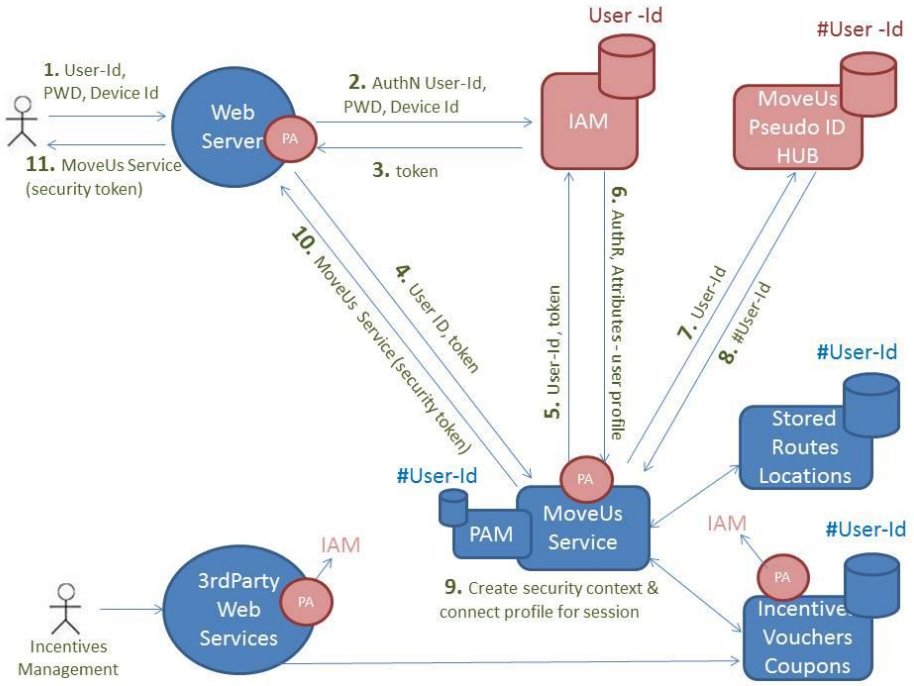


Figure 15 Security Management diagram (from D3.4 [10])

d. Personal Account Manager (PAM)

It manages the user's mobility personal profile: personal data, preferences and habits enabling personalized and customized mobility information service.

i. Manage mobility personal data

It Recovers/Updates the mobility information, in terms of preferences and habits of the logged user:

- Usual Trips. Including origin, destination, time and calendar (day of the year, month or week) (User.MV_TravellerProfile.Usual_Trips)
- Mobility Preferences: preferred transport mode (car, PT, bike, walk, sharing). This information is stored as pairs of type (transport, rating), where rating is the rating of preference. This information could be filled in by the user or in an automatic way taking the history of trip plans

(User.MV_TravellerProfile.MV_TravelProfile. PreferedTransport).

- Car pooling Preferences: preferred role (driver, passenger), gender, smoker…) for pooling service (MV_CarPoolingProfile)
  - SendDataOnMobility (UserId, MobilityData): returnRegistrationStatus (G) (I)
  - RequestDataOnMobility (UserId): MobilityData
  - SendUsualTrips (UserId, [UsualTrip]*) (M)(I)
  - RequestUsualTrips (UserId): [UsualTrip]*
  - SendCarPoolingPreferences (UserId, CarPoolingProfile)
  - RequestCarPoolingPreferences (UserId): CarPoolingProfile
  - SendMobilityPreferences (UserId, MobilityPreferences)
  - RequestMobilityPreferences (UserId): MobilityPreferences

Where MobilityData = UsualTrips () + Mobility Preferences + Car Pooling Preferences and Use.

ii. Manage personal Data

It Recovers/Updates the personal and vehicle data of the logged user:

- PersonalData. Includes: name, surname gender, address, email, telephone 1/2, fax, accessibility constraints and POI preferences (for any of the predefined PoI (e.g. parkings, restaurants or merely locations). Access to the User Management Schemes Datastore user (MV_PersonalProfile data).
- Vehicle Data. The information of the user registered vehicles (until 2): engine type, power, fuel consumption and other environmental-related data (to be defined according the WP4 algorithm for energy consumption/efficiency calculus). Access to the User Management Schemes Datastore user (MV_VehicleProfile data).

Both data structures can be received at the same time (PersonalData) or separately (UserPersonalData/VehicleData)

- SendPersonalData (UserId, PersonalData)/ SendPersonalData (UserId, UserPersonalData) / SendPersonalData (UserId, VehicleData): returnRegistrationStatus (G) (I)
- RequestPersonalData (UserId): PersonalData
- RequestUserPersonalData (UserId): UserPersonalData)
- RequestPersonalData (UserId): VehicleData

Where PersonalData = MV_PersonalProfile (User Personal data) + MV_VehicleProfile (Vehilce Data)

iii. Manage general preferences

It Recovers/Updates user general preferences related to its desired application behavior. Specifically, it allows setting the following aspects: enable/disable push notifications, enable/disable real-time tracking, incentives, coupons and advertisement (MV_GeneralUserPreferences).

- SendGeneralPreferencesData (UserId, GeneralPreferencesData); returnRegistrationStatus (G) (I)
- RequestGeneralPreferencesData (): GeneralPreferencesData

iv. Manage user preferences.

The system saves and recovers the user preferences full data, integrating as PreferencesData: MV_GeneralUserPreferences (general preferences), MV_CarPoolingProfile (Car pooling Preferences) and MV_TravelerProfile (mobility preferences), all previously defined.

- SendPreferencesData (PreferencesData); returnRegistrationStatus (G) (I)
- RequestPreferencesData (UserId): PreferencesData

v. Manage personal energy consumption data

The end user enters the personal information on environmental footprint including Maximum amount of carbon footprint (monthly, yearly) and Maximum amount of energy consumption (monthly, yearly).

- SendEnergyConsumptionData (ConsumptionData):returnRegistrationStatus(G)(I)

vi. Preferred/Habitual Trips Learning

As inputs from the user are the origin and/or destination, the date and time of the query.

Different approaches are candidates, depending on the storage capabilities and response times:

- [L-V],S,D. Constant along the year, defining a day a working, Saturday or Sunday.
- [L-V],S,D and bank days. Current along the year, with the exception of these days when the common behavior differs from habitual.

- Personalized trip pattern. Automatically, the system processes previous trips and extracts the user mobility pattern, defining accordantly day type.

Internally, the GTP Data, stores intermediate information, easing and speeding up the identification of habitual trips from previous experience. As parameters we have the time slice.

Procedure:

The component returns the most frequent trips, according the user day and temporal slots.

A simplified version, where values are introduced directly by the user could be deployed by calling RequestUsualTrips (origin, destination, time and calendar (day of the year, month or week).

- RequestExistingTripsFromAutoLearn (User Id, origin, destination):ExistingTrips (G) (I)

-

vii. Manage personal favourite PoI

As input the component receives the PoI location or PoI identification and its type.

Three operations are observed:

- SendAddPoIPreference. To add/update a new user favourite PoI.
  - Searches the PoI into the PoI Catalog (PoI Data) and recovers its identifier. If not found, inserts the new PoI on the catalog with the available information.
  - Adds a new entrance in User.MV_PoIpreferences and linked to the PoI catalog.
- SendDeletePoIPreference. To remove an existing user favourity PoI. Searches and removes the PoI in the MV_PoIpreferences list.
- RequestUserPoIpreferences. Returns the full list

  - sendSelectedParkingOption (selectedParkinkOption) (T) (I)

viii. Update user mobility information.

The system receives the trip information as input and updates accordingly the user mobility preferences.

Insert the trip into the Travel Information Data store (Personal Mobility Trace Data)

- Re-calculates the traveler profile, specifically transport mode preferences (User.MV_travelerProfile.MV_transportRating)

- Updates incentive balance. Call to the method.UpdateIncentiveStatus (trip information).
- Updates energy consumption information. Increasing the value of User.MV_EnergyConsumption.Total and User.MV_EnergyConsumption.Transport_EC for those transport modes used along the trip. The current measures are taken from the trip description JP_Journey.JP_Leg.ECValue.

2. Administration

a. Register

i. Request for available services

Any user can request for services available in a specific location or city. The component returns the registry information (MV_LocalRegistry.Services), in coherence with the services deployed on the platform. Specifically, users can select their own services.

- GetAvailableServices (version: string): MV_Service*
- GetAvailableServices (User_id: string, Location: string): MV_Service*
- GetAvailableServices (User_id: string, X,Y: float (GPS position)): MV_Service*

ii. Set of application configuration options

The platform Administrator (or local administrator) sets the different specific tailoring and customization configurations according each pilot and user type peculiarities and needs.

- SetApplicationConfiguration ([MV_LocalRegistry]*)

iii. Request for application configuration parameters

Any user (except those of type UT-7) requests for specific application configuration. The Register component will select the configuration according the user input: user id and location (country and city) or his/her GPS location.

1. In case of specific location, the function searches on Register Data structure and returns the registry associated (MV_SpatialExtent.LocationName) for the user type or role (MV_FeatureRole.RoleId/UserType).

2. Otherwise, if current GPS location is populated, the function will return the nearest registry associated to the user type (MV_SpatialExtent.Center).

   - GetApplicationConfiguration (User_id: string, Location: string): MV_LocalRegistry.

- GetApplicationConfiguration (User_id: string, X,Y: float (GPS position)): MV_LocalRegistry.

iv. RequestRoutePreferenceCriteria ()

The component recovers the full list of preference criteria from the application configuration data (MV_Registry) (e.g. cheapest, easiest, fastest, minimum transfers, scenic, eco or incentive). But it is also possible for n user to select from its mobility preference profile (User.Mobilitypreferences.PreferenceCriteria from the MN_RoutePreference).

6.3.2 MoveUs Incentive Management Module

The section MoveUs City Service Web-based tools for incentive management (deliverable 3.3 [8], section 4.2) describes deeply the application specified to manage incentives in the context of MoveUs project. For a detailed description of the different operations supported, refer to that document. Its main functionality will be deployed also on the MoveUs platform, providing appropriate interfaces. Below, we introduce again and describe the platform elements: users and interfaces, supporting the incentive functionality. The operation diagram remains, with the exception of and additional service: RequestCalculeIncentive, allowing the calculation of incentives associated to a certain route or feedback activity.

Definitions

- **Electronic Wallet Registry**: set of payments systems;
- **Coupon**: a digital code that allows you to get discounts;
- **Voucher**: a digital code that corresponds to a purchase prepaid;
- **Incentives**: set of material and virtual objects that help modifying the mobility behavior to obtain a reduction of driving and/or a use of alternative modes (i.e. from private vehicle to public transportation, or to a higher Euro class). Incentives can be also defined as the generic 'money' that can be spent to get benefits or deals. Concretely this 'money' will be defined more specifically (see afterwards)

Users

- **Type 1 (UT1)**: Entity defining RULES associated to the incentives
- **Type 2 (UT2)**: Entity providing INCENTIVES
- **Type 3 (UT3)**: Entity where INCENTIVES can be spent, entity providing awards (benefits that can be obtained with a certain amount of incentives)
- **Type 4 (UT4)**: Final users
- **Type 5 (UT5)**: Entity providing COUPONS. They can be:
  - **UT5_MoveUs:** MoveUs Internal module usable by other entities to provide COUPONS/DEALS
  - **UT5_EXT:** External entities providing directly COUPONS via a MoveUs interface available for this purpose
- **Type 6 (UT6):** Entity providing ADVERTISEMENTS
  - **UT6_MoveUs:** users that uses a MoveUs input tool to insert ADVERTISEMENTS items
  - **UT6_EXT:** users that provides ADVERTISEMENTS items automatically via a MoveUs interface specifically available for this purpose

Types of Incentives

Type	Unit of measure / Currency
Incentives that are calculated from Energy Efficient Behavior	CREDIT
Incentives that can be spent in general at more UT3s associated to MoveUs	M_COIN (MoveUs Coins)
Incentives that can be spent only at one specific UT3	B_COIN (Bonded Coin)

Table 10 Incentive types

Functional View

The functional view of the I/C/D/V is depicted in the following figure.

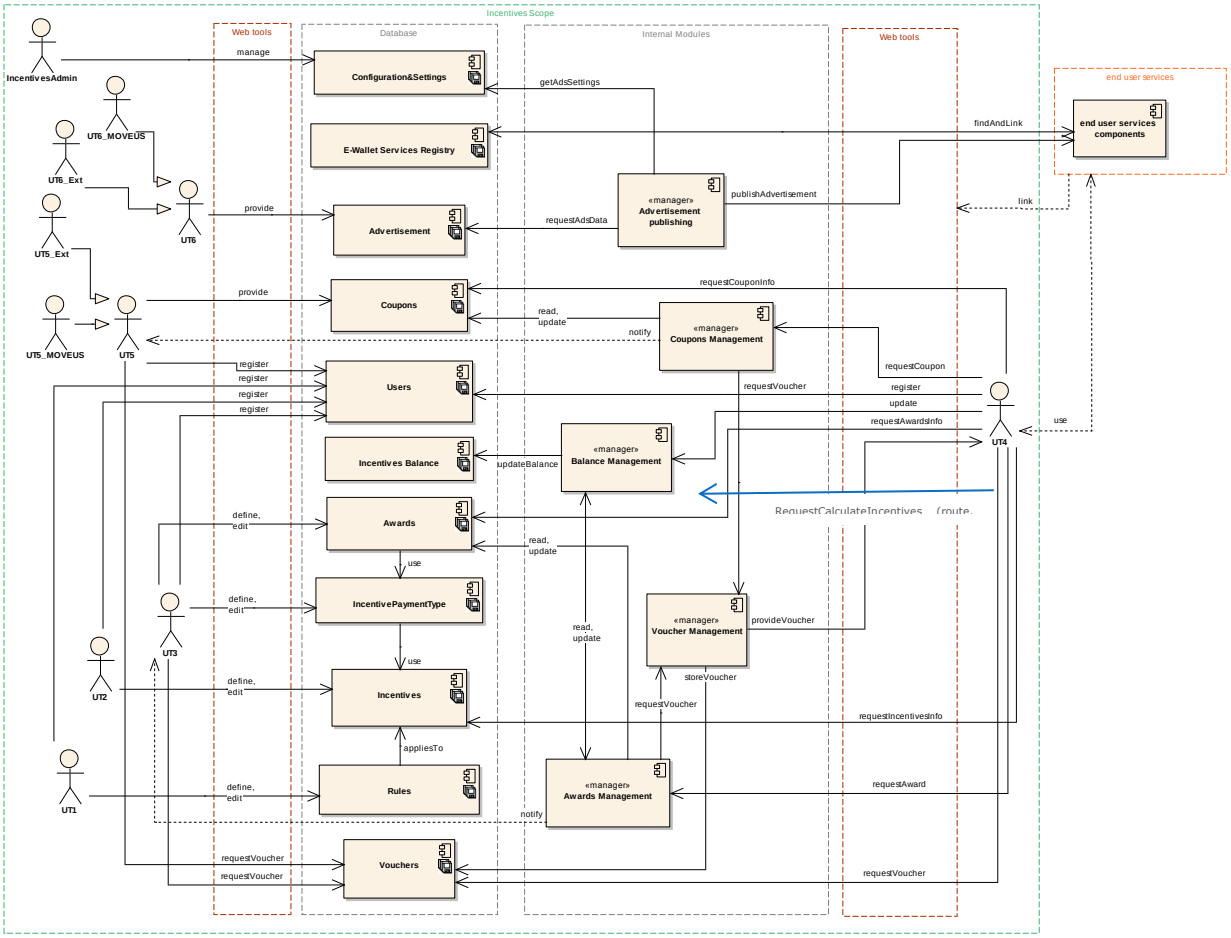


Figure 16 Incentive Management functional view

Interfaces (to be refined in WP5):

Rules Management

- RequestOwnedRules (UserId): [MV_I_Rule_Basic]*
- RemoveRule (UserId, RuleId): ACK/NACK
- UpdateRule(UserId, RuleDesc): ACK/NACK

Currency Management

- RequestCurrencies (UserId): [Currencies]*: [MV_IncentiveCurrency]*
- RemoveCurrency (UserId, CurrencyId (ID:int)): ACK/NACK
- UpdateCurrency (UserId, CurrencyDesc): ACK/NACK
- CreateCurrency (UserId, CurrencyDesc (MV_IncentiveCurrency): ACK/NACK

Incentive Management

- RequestIncentiveList (UserId): [MV_Incentive]*
- RemoveIncentive (UserId, IncentiveId (ID:int)): ACK/NACK
- UpdateIncentive (UserId, IncentiveDesc): ACK/NACK
- CreateIncentive (UserId, Currency (Id: int), Rules: [Id: int]*): ACK/NACK

Incentive Payment Type Management

- RemoveIncentivePaymentType (UserId, IncentivePaymentTypeId (ID:int))
- RemoveIncentivePaymentType (UserId, IncentivePaymentTypeId (ID:int))
- UpdateIncentivePaymentType (UserId, IncentivePaymentDesc)
- CreateIncentivePaymentType (UserId, IncentivePaymentDesc)

Awards Management

- CreateAwards (UserId, Award)
- RequestAwardList (UserId): [MV_Awards]*
- RemoveAwards (UserId, AwardId (ID:int))
- UpdateAwards (UserId, IncentiveDesc)

Coupon Management

- RequestCouponList (UserId, Award): [CouponDesc]*
- RemoveCoupons (UserId,CouponId (ID:int))
- UpdateCoupons (UserId, CouponDesc)
- CreateCoupons (UserId, CouponDesc)
- SetCouponActivation (UserId, CouponId (ID:int), activated: True/false)
- EmitVoucher (UserId, CouponId)

Advertisement Management

- RequestAdvertisementItem (UserId): [Advertisement]*
- RemoveAdvertisementItem (UserId,AdvertisementId (title:string))
- UpdateAdvertisementItem (UserId, AdvertisementDesc)
- CreateAdvertisementItem (UserId)
- NotifyAdvertisementData (UserId, AdvertisementInfo) (canalized via subscription/notify module

Incentive Information Access

- RequestIncentiveList (UserId): [MV_Incentive]*
- RequestAwardList (UserId): [MV_Awards]*
- RequestCouponList (UserId, Award): [CouponDesc]*

Request of Awards

- RequestAwardList (): [AwardDesc]*
- RequestAward (UserId,AwardsId): ACK/NACK
- NotifyRequestAward (UserId,AcquirerUserId, AwardsId): ACK/NACK
- EmitVoucher (UserId, CouponId)
- NotifyVoucherIssue (UserId,AcquirerUserId, AwardsId): ACK/NACK

Request of Coupons

- RequestCouponDeal(UserId,CouponId): ACK/NACK
- NotifyRequestCoupon(UserId,AcquirerUserId, CouponId): ACK/NACK
- EmitVoucher (UserId, CouponId)
- NotifyVoucherIssue (UserId,AcquirerUserId, AwardsId): ACK/NACK

Access Voucher

- VoucherRetrieval (UserId)
- RequestAVoucher (UserId, VoucherId, eMail: boolean)

Administration

- SetAdvertisementCondition ([Interest of user||Geographical criteria], TRUE/FALSE)
- SetUserAdvertisement (UserId, true/false)
- UpdateCoupons (UserId, CouponDesc)

6.3.3 MoveUs CF/EC Estimation Module

The assessment methodology, methods and underlying data needs are currently under definition in WP4 (refer to document Methodology benchmarking for energy efficiency and carbon footprint assessment.docx led by TUT).

Definitions

- **Key Performance Indicator**: energy related indicators relevant in the transportation sector.
- **Affecting parameters**: parameters that affect energy consumption and carbon footprint values.
- **User parameters translations**: translation of energy efficiency and carbon footprint values for users.
- **Incentives:** set of material and virtual objects that help modifying the mobility behavior to obtain a reduction of driving and/or a use of alternative modes (i.e. from private vehicle to public transportation, or to a higher Euro class). Incentives can be also defined as the generic 'money' that can be spent to get benefits or deals. Concretely this 'money' will be defined more specifically (see afterwards).

Users

- **Type 1***: Final users
  - **Anonymous**: User that wants to keep their identity anonymous.
  - **Identified**: Users that provide their information to the MoveUs platform to get incentives.
- **Type 2***: Entity that receives the incentives equivalences – green points values per journey.

- **Type 3*:** Entity in charge of updating the equivalences of energy efficiency and carbon footprint to incentives (credits/green points) based on the characteristics of different cities.

*: *specific user type naming currently under definition in WP4.*

Types of Equivalences

Type	Unit of measure / Currency
Incentives that are calculated from Energy Efficient Behavior	Green points/CREDIT

Table 11 Incentive/EC equivalences

In order to have more flexibility, the following behavior is foreseen in the current design of the Energy Efficiency Database:

- One single Type_2*_user can accept one type of equivalences – green points
- One single Type_3*_user can define/change the rules for the energy and carbon footprint equivalences

Additional constraints could be added at applicative level.

Functional View

The functional view of the EC/ CF is depicted in the following figure.

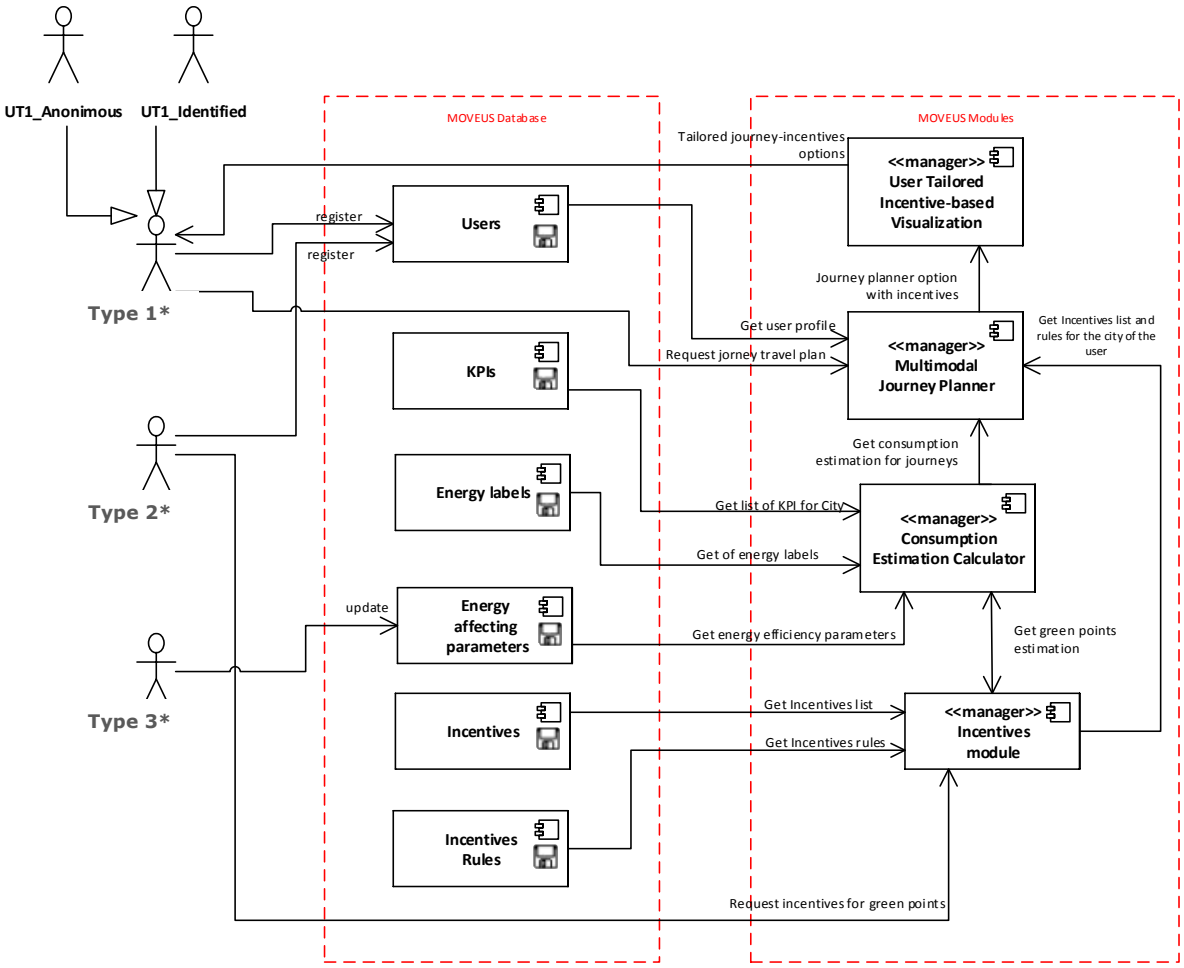


Figure 17 Energy Consumption/ Carbon Footprint Estimation functional view

MoveUs Database - high level definition

Users	Registry of users described by: • ID • User Type • Name • E-mail • Etc. Plus User-specific attributes
KPI	Set of Key Performance Indicators related with energy efficiency and carbon footprint in the transportation domain. Described by: • Description • Calculation • Relation to transportation modality • Etc.

Energy labels	Set of translations of energy efficiency values for users. Described by: • Description Equivalence to user meaningful values, e.g. cost,
Energy affecting parameters	Set of parameters that are affected by KPIs in the different living labs. Described by: • Description • KPIs related to this parameter • Positive or negative effect for energy consumption and carbon footprint • Etc.
Incentives	Data on incentives based on the LL where it is used. Described by: • Description • Type • Unit of measure – green points • Etc.
Incentives Rules	Data that define the relation of energy efficiency and carbon footprint for incentives. Described by: • Description • Relation-equivalence to green points • Etc.

Table 12 EC/CF datastores

MoveUs Modules

User Tailored Incentive-based Visualization	Access the Coupons/Deals database
Multimodal Journey Planner	Manages the request of a consumption estimation calculator per journey Receives user requests Access incentives rules Access incentives list
Consumption Estimation Calculator	Access the Energy Labels database Access the KPIs database Access the Energy affecting parameters
Incentives module	Manage the incentives for each one of the cities subscribed to MoveUs Platform Access incentives rules Access incentives list

Table 13 EC/CF related modules and operations

Functions/Operations/services

3. **Consumption Estimation Calculator**

    a. **Provision of consumption estimation for journeys**

A Type_1_user can request a journey plan with incentives values for the different options, which is related to the energy efficiency choices. Consumption Estimation Calculator provides the conversion to green points for each option to the Multimodal Journey Planner. The module executes the following operations:

i. The details of each one of the journey options are received from the Multimodal Journey Planner including the modality of transportation, distance of each journey and walking distances if needed.

ii. The Consumption Estimation Calculator evaluates each one of the options by requesting the list of KPI of the city from the KPI DB related with each transportation type of the journey options.

iii. Once the KPIs are selected, the Energy Efficiency Parameters DB is queried to get the list of parameters related to the KPIs for each transportation type.

iv. Then a query to the Energy label DB is performed to assign a meaningful value to the energy calculations that will be performed based on the journey, related KPIs and Energy Efficiency Parameters.

v. The consumption estimation values are calculated for each journey with a value based on green points

vi. This information is retrieved to the MJP to assign an incentive value based on the green points estimation.

b. Request of list of KPIs for each city registered in MoveUs Platform

The User Type 1 issues a request to the module Multimodal Journey Planner. Consumption Estimation Calculator gets the list of KPIs for the specific city where the user is placed to compute the energy consumption.

c. Request of Energy Labels

The consumption estimation calculator requests the energy label for calculation of the energy values per journey.

d. Request of Energy Efficiency Parameters

A User_Type_3 can update the list of energy efficiency parameters, which are used by the consumption estimation calculator.

e. Provision of energy efficiency equivalence

A User_Type_3 can request the equivalences of incentives of green points for incentives.

Interfaces (to be refined in WP5)

- SendCurrentEnergyConsumptionRequest
- SendHistoricalEnergyConsumptionRequest

- SendPeriodHistoricalEnergyConsumptionRequest
- SendenergyIncentivesRequest
- SendEnergyEficiencyTipsRequest
- RequestRouteEnergyConsumptionEstimation

6.3.4 MoveUs PT Operations Management

Definitions

- **Smart Prioritization**: Service that gives priority to specific vehicles (public buses) on intersections controlled by traffic lights, optimizing travel time and efficiency.
- **OBU.** On Board Unit.

Users

- **Type 4 (UT4)**: Final users
  - **UT4_Anonymous**: User that wants to keep their identity anonymous
  - **UT4_Identified**: Users that provide their information to the MoveUs platform
- **Type 8 (UT8)**: Field/on-board devices, sensors and External Traffic/Transport Service Operator Systems
  - **UT8_BusOBU**: On-board Unit (Bus)
  - **UT8_CarPoolingSys:** Car Pooling Service Operator Systems/Backend
  - **UT8_PTBusSys:** Bus Public Transport Operator Systems/Backend
  - **UT8_BikeHiringSys**: Bike hiring Service Operator Systems
  - **UT8_TrafficCntlSys:** Centralized Traffic control System

Functional View

The functional view of the PT Operations Management is depicted in the following figure.

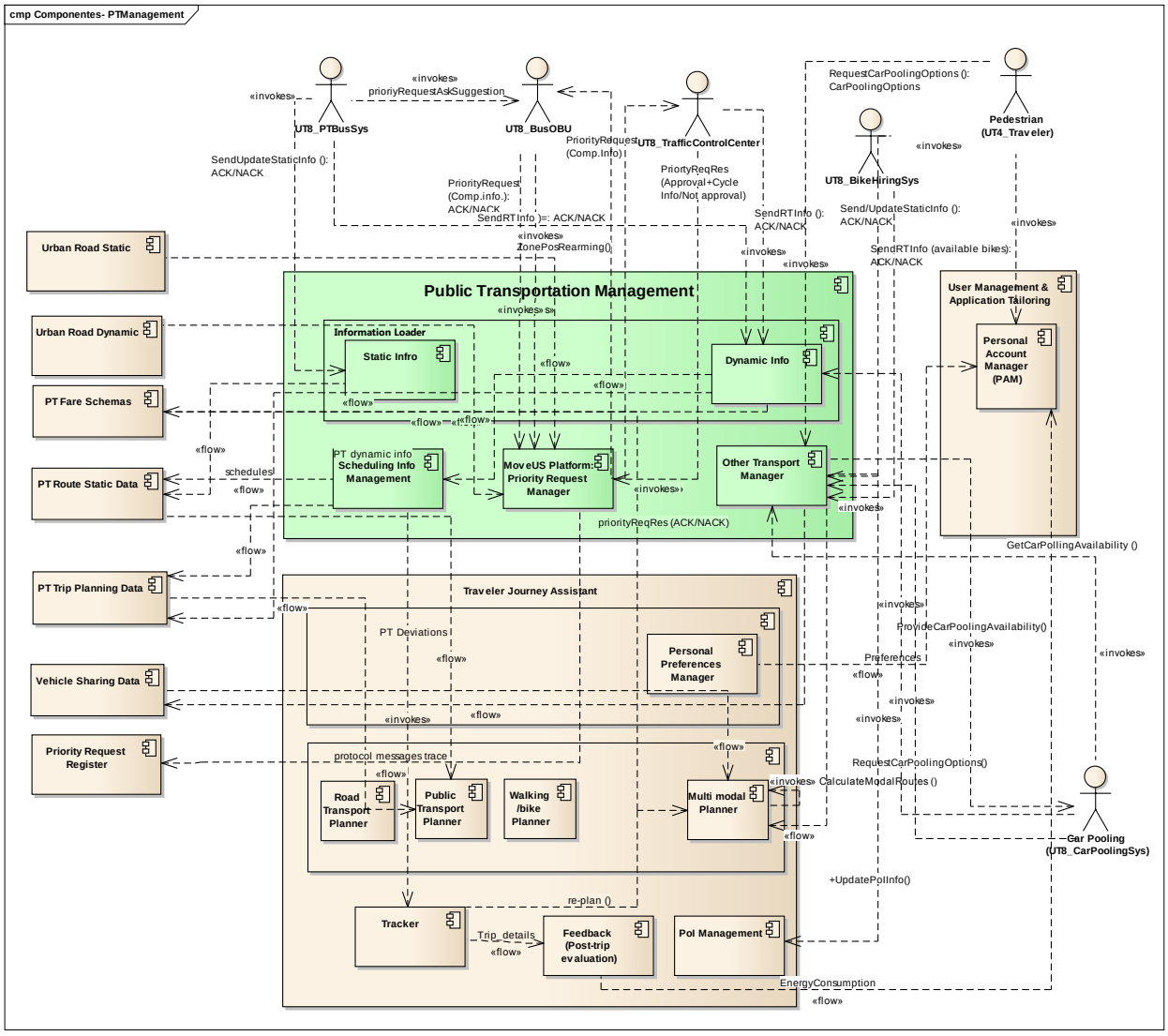


Figure 18 PT Operation Management functional view (in green)

Trip Plans Data	The result of the trip planning process is used along the on-trip phase, as reference to identify perturbations affecting estimation times and travel viability.
Urban Road Static Data	Contains information about the road network and the traffic conditions within it for use in planning trips. It shall be possible to integrate current and predicted data for different date/time combinations. It used to planning.
PT Route Static Data	Static information of Public Transport service: routes, stops, prices.
PT Route Dynamic Data	Analogous to Road Trip Planning Data, it integrates the information about the services provided by the Public Transport operations plus the fares that will be charged and shall be for use in planning trips.
General Trip Preferences (GTP) Data	Contains the personalised data needed to support the Traveller during all of their trips, from the trip planning, trip execution and finally, being updated once finalized.
Personal Mobility Data	Repository of historical information, where all the information sent by the application capture module resides. This information is used in the queries associated with mobility analysis.
Vehicle Sharing Data	Repository of the planned pooling services as candidates for the accomplishment of a trip.
Priority Request Register	Register of the protocol messages exchanges to support the priority request functionality.

Table 14 PT Operation datastores

MoveUs Modules

Information Loader	
Static Information	Capture traffic static information
Dynamic Information	Capture traffic dynamic information
Scheduling Info Management	Identifies and manages expected scheduling deviations.
Priority Request Manager	Implements the Priority Change Request functionality
Other transport management	Manages information from other public transport systems

Table 15 PT Operation modules

Functions/Operations/services

1. **Information Loader**

    a. **Static Information**

        i. The Public Transport Operator sends the static information of its transport service.

ii. In the context of MoveUs, will provide the following: Bus stop geo-position, Bus lines, planned schedules & frequency. All this information is stored on the PT Route Static Data store (according the schema defined in the Simplified Dynamic Public Transport Model package) and shared with the PT Planner System.

iii. If fare or tariff information is available, will be saved on specifically defined PT Fare Schemas (extending with GTFS format) or using existing attributes of PT Route Dynamic Data store.

b. Dynamic Information

The Public Transport Operator sends updated information about provided services. For each bus: arrival time at bus stop, estimated travel time and incidences.

i. Insert dynamic information on the PT Trip Planning Data.

ii. Call Scheduling Management Component to identify deviations between schedule and real service planning.

2. Scheduling Info Management

i. When the component receives a new estimation of timings, it evaluates the deviations versus the expected journey.

ii. If deviations are higher than a predefined threshold, then it calculates new expected times and calls the SendServiceDeviation of the Tracker module.

iii. Store deviations on the PT Trip Planning Data for QoS analysis.

iv. Receive the dynamic timing of the Public Transport Services.

3. Priority Request Manager

i. In this case, the MoveUs platform serves as a gateway between the OBUs system (On-board bus system)/Bus Control Center and the Traffic Control System; the transactions are stored in the internal database and re-sent to destination. This will enable subsequent audits and performance adjustment.

ii. Any request will be handled as a transaction, an individual, indivisible operation, succeeded or failed as a complete unit. Any request will be uniquely identified and traced across the different phases, storing the following information:

- State, that could take the following values: requested, request revocation, sent to traffic control, accepted, rejected, finished.
- Transitions, with a timestamp and origin/target states.

a. Receive Priority Request from OBU

i. Receives a request from the OBU when it enters on the detection zone. This information including complementary information like the bus identification code, the bus line, direction and delay is stored into the Bus Priority Request Datastore.

ii. MoveUs answers with an ACK or a NACK to the priority request, according to the status of connection to traffic system and the current situation of the crossing (normal, no priority request pending).

iii. Rearming Zone is set to prior operation in course.

b. Re-send of request to the Local Traffic Controller (LTC) (Bus at the detection zone)

i. MoveUs platform selects from the Urban Road Static Data (specifically crosses information), the specific LTC (Light Traffic Controller) that manages the next traffic light in the bus route, according to its scheduled route and current position.

ii. MoveUs directly re-sends the priority request message to the selected LTC, with the relevant additional information needed. This step is time-critical for the correct provision of the service.

c. LTC response management

i. The MoveUs platform receives the response from the LTC, an ACK (or an NACK message) as a result from the assessment of the request action. Further information might be also included in the ACK message from the LTC to MoveUs if needed, like cycle duration and cycle state, etc.

ii. MoveUs platform stores the transaction information, changing its state and time-stamp.

iii. MoveUs will direct the ACK or NACK message to the OBU/Bus Control Center that requested the priority.

d. Re-arming Zone position setting

i. When the bus is at the re-arming zone, the OBU will inform MoveUs platform with a message.

ii. The priority service operation in the crossing gets back to normal operation.

iii. The request transaction is updated as finished.

iv. Afterwards, MoveUs will send such information message to the LTC.

4. **Point of Interest Management**

   a. **Store the static information of Point of Interest (PoI)**

      i. Insert the PoI static info on the PoI data structure, composed by the location and entrance position if known (according the schema defined in the PoI Data Model).

      ii. Specifically, two POI types are extended accordingly in MoveUs platform with specific information: Parking and Bike/Car Hiring. The information will be accessible from the planner, as inter-modal exchange gates.

      - SendStaticInfoPoI (PoIInformation)
      - Send Static Info SendBikeHiringStaticInfo (Location geo-position, bike slot numbers)
      - Send Static Info SendParkingStaticInfo (Location geo-position, parking slot numbers)

   b. **Recover static information of Point of Interest (PoI)**

      i. Returns the static information of the selected point of interest (PoI) from PoI Data datastore, in case the position is filled in, the selection covers the radius area around the location. The parameter PoI enables selecting specific types of PoI (e.g. restaurants, crosses, etc), defined by the PoI.PoICategory.

      - RequestStaticInfoPoI ([PoIType]*,position, radius): [PoIInformation*]

   c. **Delete the static information o Point of Information (PoI)**

      i. Delete the PoI static info on the PoI data structure.

      ii. Select which users are subscribed to the specific information related the PoI type and unsubscribe them.

5. **Other transport management**

   a. **Bikes/Car Hiring**

      i. Setting and update of Bike/Car Hiring Static Information.

      The Bike/Car Hiring Company sends its static information, geo-location and bike slots available to MoveUs platform. This information is stored in the PoI repository, being accessible to the multimodal planner.

      ii. Set Bike Dynamic Information.

      The Bike Hiring Company sends its static information, geo-location and bike slots available to MoveUs platform. This information is stored in PoI repository, being accessible to the multi-modal planner.

- Send Real time Info SendRTInfo (Available bikes) (M) (I)

b. Car Pooling Management

i. Request of Car Pooling Options

Returns the car pooling travels matching the desired shift (start and end location), and Car pooling preferences.

Steps:

i. For each registered car pooling company send a RequestCarPoolingOptions request.

ii. The pooling information is stored on the Vehicle Sharing Data structure, replacing previously existing trips for the source company.

- RequestCarPoolingOptions (TripRequestData): CarPoolingOptions (G) (I)

Where TripRequestData: CarPoolingPreferences || OriginDestinationRequesType (origin, destination)

ii. Offer a plan travel for car sharing support

Shares a plan travel as candidate for car sharing and sends its description to the registered car pooling/sharing companies.

- SendCarPoolingOffering (CarPoolingOptions): ACK/NACK (G) (I)
- CarPoolingOptions = MV_CarPoolingTrips*

iii. Pooling service confirmation (Optional)

Car pooling company confirms the pooling service for a specific itinerary and sends an identifier, to manage the coordination.
MoveUs platform marks the travel (Trip Planner Plan Data) as shared and assigns the received identifier. Parts will coordinate externally through the Pooling Service Company System.

Where CarPoolingOptions = MV_CarPoolingTrips*

- SendCarPoolingOfferingConfirmation (CarPoolingOptions): ACK/NACK (G) (I)

iv. Car Pooling Service Company Register/unregister

A company can register/unregister from MoveUs platform service. In this case the company information will be removed from the Users (**UT7_CarPoolingOp**) together with all its information allocated on Vehicle Sharing Data.

6. **Public Transport Vehicle Position**

It manages the current position of the vehicle along the route. When received, PT Route Dynamic Data is updated accordingly. Deviations of travel plan are identified in the Scheduling Info Management. The position received has only informative character.

- SendPTPosition (trip, x,y) (M) (I)

6.3.5 MoveUs Traffic Management Module

Definitions

- **Logic configuration**: Set of parameters that define the behavior of MoveUs prediction algorithm. Specifically, it includes temporal ranges, applicable algorithm per range, historical road information window and selection rules.
- **Data fusion**: Integration and combination of multiple, commonly heterogeneous data sources into a consistent, accurate, complete and more reliable model. This process is supported by different algorithms and techniques.
- **Map matching**: Association of a sorted sequence of vehicle/user positions to a reference road network deployed on a digital map.
- **Smart Crossing**: Provision of smartest crossing options to the pedestrians by means of specific crossings and automatic request and reaffirmation.
- **Energy Consumption:** See 5.1.2.2
- **Incentives**: See 5.1.2.1

Users

- **Type 1 (UT4)**: Final users
  - **UT_Anonymous**: User that wants to keep their identity anonymous
  - **UT4_Identified**: Users that provide their information to the MoveUs platform
- **Type 8 (UT8)**: Field devices and External Traffic/Service Systems
  - **UT7_TrafficCntlSys**: Traffic/Mobility Center
  - **UT7_BTNetwork**: Bluetooth Network sensor
  - **UT7_TLC**: Traffic Light Controllers

Functional View

The functional view of the Traffic Management is depicted in the following figure.

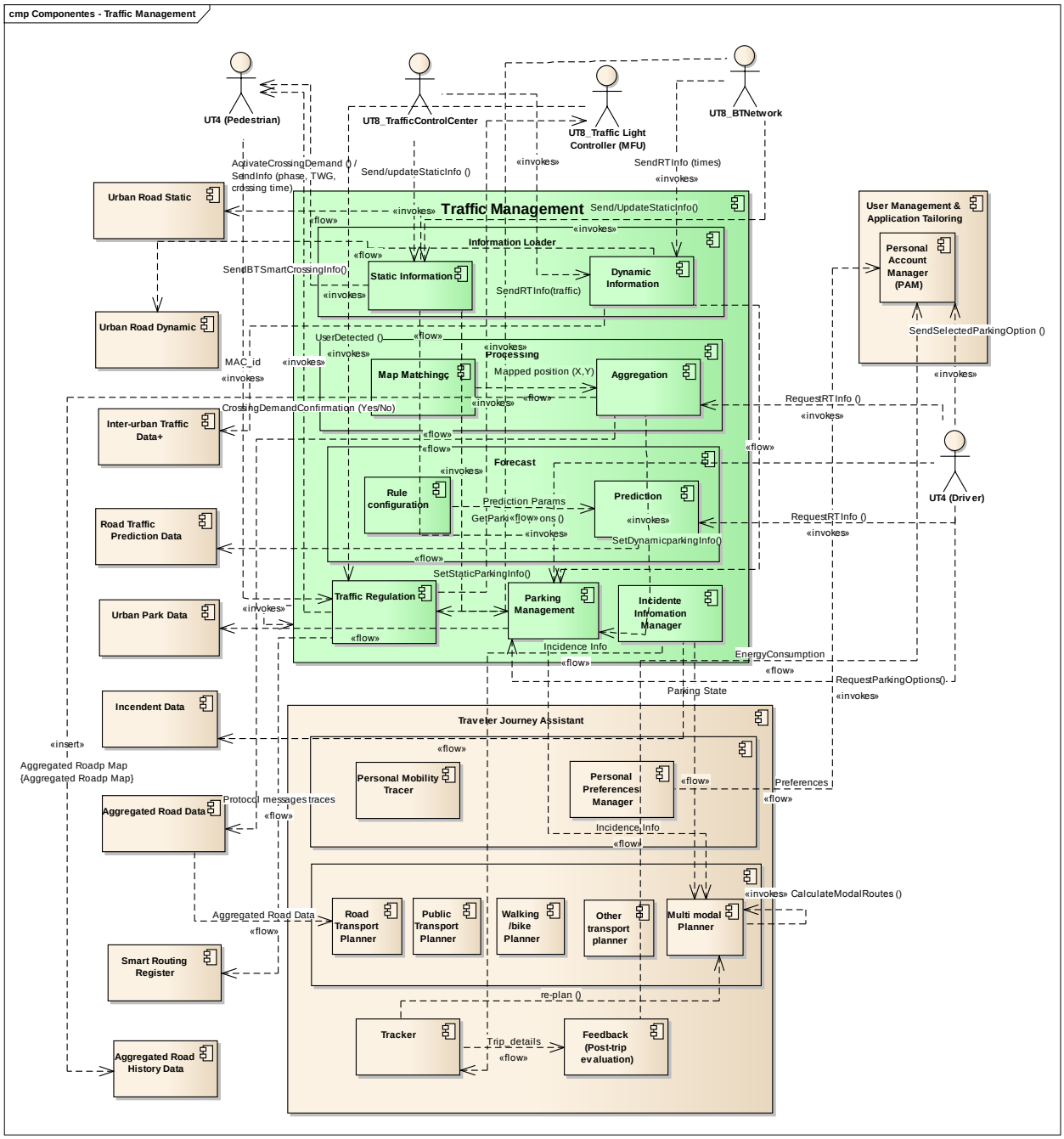


Figure 19 Traffic Management functional view (in green)

MoveUs Database - high level definition

Urban Road Static Data	Network data model, whose structure is supported by an interconnected set of network points, lines and road junctions. Also distances and segment characteristics as type, number of lanes/ways.
Urban Road Dynamic Data	Depending of the deployment, different inputs from FCD, stations / fixed sensors (e.g. Bluetooth sensor network) and many other field devices constitute measured dynamic data from equipment or outside stations, e.g. traffic and weather measurements.
Inter-urban Traffic Static Data	Analogous to urban but referred to surrounding inter-urban segments.
Inter-urban Traffic Dynamic Data	Analogous to urban but referred to surrounding inter-urban segments.
Road Traffic Prediction Data	Given a validity period (now or future) and forecast scenarios (e.g. short, medium and large), is defined by a sequence of road state predictions.
Urban Park Data	Parking data, including static: location, access, cost information, operating hours, restrictions, fares and also dynamic, availability and free slots.
Incident Data	Include accidents, abnormal traffic states, obstructions and activities undertaken by the operator as maintenance or construction.
Aggregated Road Data	Reference representation of the monitored road network (Urban Road Static), to which additional layers overlap. On the various links between nodes, it will manage vehicle information crossing at each instant of time (stored position and velocity), and measures from stations / fixed sensors (e.g. Bluetooth sensor network).
Aggregated Road Historic Data	Historical repository of Aggregated Road Information, storing a predefined temporal interval.

Table 16 Traffic Management datastores

Modules

Processing	
Aggregation	Aggregates (according to a defined data fusion algorithm) the available traffic/mobility data
Map matching	Maps mobility (GPS) traces on the reference street map.
Information Loader	
Static Information	Captures traffic static information for urban and relevant inter-urban areas and populates internal data-bases.
Dynamic Information	Captures traffic dynamic information for urban and relevant inter-urban areas and populates internal databases.
Forecast	

Logic Configuration	Configures the behavior of forecast and prediction module
Prediction	Predicts traffic/weather according module configuration.
Incidence Information Manager	Manages incidences, integrates external incidences and implements pattern matching for incidence identification.
Traffic Regulation Manager	Implements the Smart Crossing functionality

Table 17 Traffic Management modules

Description of operations / functions / services

Identified interfaces (to be refined in WP5), based on use cases and city service description are marked with 'o'.

1. **Processing**

This module processes incoming traffic/mobility information, improving its reliability, filtering and removing sampling errors, aligning samples to reference road maps and aggregating/synthetizing information.

a. Map matching

i. Alignment of a sequence of positions to a reference road map.

This Alignment will be invoked on two different ways, being different the underlying algorithm: triggered when a new specific traffic measure is received or once delimited a vehicle track.

1. Map Matching of isolated locations. The measures do not necessarily correspond to the same path or their temporal order.

   The method returns an ordered segment/nodes list, making a further trajectory matching easier. The component returns the projection of the measured position on the corresponding segment, in terms of proximity, an accuracy measure.

2. Map Matching of a trip. In this case, the reliability of the algorithm decreases with low sampling rates, noise and specific scenarios as bridges, two-sides segments or crosses. Analyzing the track as an entity enables the use of heuristics and trajectory predictions, based on the temporal sequence of positions.

The map matching can be executed on off-line or on-line; in the latter case, the track mapping functionally is tentative.

- PositionMapMatching ([PosX, PosY, Speed])
- TrackMapMatching (Track)

ii. Aggregation

Its aim is to apply data fusion techniques in traffic state estimation, according to the different real-time traffic data available. Three kinds need to be distinguished here: the raw field data from floating car

data (FCD), the estimated travel time per section from static infrastructure points (corresponding to the Bluetooth sensors network covering a limited part of the city and the aggregated traffic measures, as level of service and estimated travel times from external traffic control system (considering urban and surrounding inter-urban roads).

This aggregation will be invoked on two different ways: triggering when a new specific traffic measure or complete track is received.

- AggregatePosition ([PosX, PosY, Speed]/[BTSensor_Id, travel_time]). Updates the aggregated traffic information associated to the specific segments where the measure is taking place. The parameter could be one measure or a set of them.
- AggregateTrack ([Track]). Updates the aggregated information, with travel tracks, an ordered sequence of positions.

In any case, the component executes the following operations:

1. The AggregatedRoadInfo structure is updated with the available information, collected in the current time window. Once the refresh time-out expires, the data fusion algorithm is launched, obtaining as result the new aggregated information for those updated segments.
2. The AggregatedRoadInfo structure is updated accordingly, being current traffic estimations accessible for new travel planning.
3. The Tracker module is notified, CheckTripAvailability, checking the availability of the different planned trips (those involving updated segments and time-constraints).

If requested by the user or external entity

Provision of Traffic Information

1. MoveUs platform returns the aggregated information of any segment of the road, providing its level of service and travel time.

2. Information Loader

a. Static Information

i. Store the static view of the urban and inter-urban road

1. Inserts the traffic static info on the Traffic Static Info structure: Crossing geo-position, phases_id, traffic_section_id. All this information is stored on the Urban Road Static store (according to the schema defined in the Static Road DataModel package and TrafficRegulation.StaticModel).

   Note: An analysis of the alignment between both topologies, the general road map and the one for traffic control, is required.

2. Initializes the Aggregated Road Map Data, removing dynamic information.

3. With any change, the aggregated road map version is updated. The new map is available for Road Planning activities.
4. The historical backup system will remain awaiting the arrival of dynamic traffic information.
5. If the static information partially covers the global map, the area/nodes/PoI will be updated accordingly, remaining the previous state the same for the rest of the map.
6. Inserts cross information as PoI. Calls to Insert/Update function of the PoI management system.
    - Send/UpdateTrafficStaticInfo ().

ii. Store the static information of Point of Interest (PoI)

1. Inserts the PoI static info on the PoI data structure, composed by the location and entrance position if known (according the schema defined in the PoI Data Model).
2. Select which users are subscribed to the specific information related the new/updated PoI type.
3. Specifically, two POI types are extended accordantly in MoveUs platform with specific information: Parking and Bike/Car Hiring. The information will be accessible from the planner, as inter-modal exchange gates.

    - SendStaticInfoPoI (PoIInformation)
    - Send Static Info SendBikeHiringStaticInfo (Location geo-position, bike slot numbers)
    - Send Static Info SendParkingStaticInfo (Location geo-position, parking slot numbers)
    - GetPoIInformation (position): [PoIInformation]*
    - GetDynamicPoIInformation (PoI, category, location, validation time): [PoIDynamicInformation]*

iii. Store the static information of the Bluetooth network

1. Inserts/Updates the static information of a Bluetooth network reader: identifier, its address and the information needed to determine its relative position in the intersection and each one being assigned to a specific road segment.
2. Links the reader to the Aggregated Road Map, specifically to the "nearest" segment, identified as traffic_section_id or according to the map matching algorithm selected if not available.

    - Send/UpdateBTNetworkStaticInfo (Detectors position, traffic_section_id).

Dynamic Information

i. Store dynamic information of Traffic Data

1. Inserts the dynamic information on the traffic network, corresponding to the level of service/estimated travel time in traffic section_id, and incidents).

2. Appends this information to the corresponding segment of the Aggregated Road Data, for the next aggregation and data fusion iteration.
    - Send/UpdateTrafficDynamicInfo ()

ii. Store dynamic information of the Bluetooth network

1. Inserts the dynamic information of a Bluetooth network reader, corresponding to sensor activations due the vehicle/driver proximity.
2. Appends this information to the corresponding segment of the Aggregated Road Data, for the next aggregation and data fusion iteration.
3. Additionally, setting its status and availability on the Urban Road Dynamic Data. This information is also updated on the Aggregated Road Data, enabling/disabling subsequent operations.
    - SendBTNetworkDynamicInfo ()
    - SendBTNetworkDynamicStatusInfo ()
    - RequestDynamicRoadTrafficInformation ()

iii. Store dynamic information of the Traffic Light Controllers

1. Inserts/Updates the dynamic information of a traffic light controller, setting its status and availability on the Urban Road Static Datastore. This information is also updated on the Aggregated Road Data, enabling/disabling subsequent operations.
    - Send/UpdateBTNetworkDynamicInfo ()
    - Send/UpdateTrafficControlDynamicInfo ()

3. Forecast

a. Logic Configuration

Definition of the prediction behavior, specifically the following parameters should be updated according to the demand variability and relevant factors:

- Period: [Start date-End date]/[Month]/[Week]/[Day]/[Hourly]
- Range: [Month]/[Week]/[Day of the week]/[hourly]/[Minutes].
- Frequency: [Month]/[Week]/[Day of the week]/[hourly].
- Segments: Sub-network to be analyzed.
- Factors: Weather, incidences, events (Tentative)

The configuration of forecast behavior will be realized manually by the MoveUs platform administrator by writing/editing specific configuration files.

b. Prediction

Prediction is carried out following a set of steps: intermediate traffic data preparation, prediction itself and result capture and transfer.

i. **Prepare the intermediate traffic data from historical repositories**

Executes the following operations:

1. Selects relevant information (identified in the prediction parameters) from the dynamic traffic and environmental data stores from which a traffic characterization can be elicited.
2. Crosses traffic information with factors (e.g. weather situations, incidences and events), happening in the selected period.

Groups and aggregates register information and records on Road Traffic Prediction Data.

- PrepareHistoricalInternalData ()

ii. **Generate Prediction**

At periodic intervals (with frequency configurable) creates the predictions of short and medium term urban traffic data using appropriate algorithms that may be different in content and scope (e.g. involving traffic state but also other measures as weather, incidences). As input, the historical repository, previously prepared for querying.

- GeneratePrediction ()

iii. **Request Prediction Data**

The system returns the existing prediction information based on the current parameters and selected temporal period. When the validity forecast is future, this process is launched on demand, setting as initial state the most probable scenario.

- RequestPredictionData (period)

When completed, the results are stored being reused by next queries and/or sent to requester.

- PreparePrediction ()

Provision of Predicted Traffic Information

MoveUs platform returns the aggregated information of any segment of the road, providing its level of service and travel time.

- SendRTInfo

4. **Incidence Information Manager**

a. **Incidence Notification**

i. When the first trigger input data flow is received, the data that it contains shall be loaded into the Incident Data structure. Status is marked as active.
ii. If the second trigger input data flow in (b) contains revised incident data, the data in the store of Incident Data shall be updated. Status is unmarked (deactivated).
iii. The activation of an alarm will take priority over all other activities within the component.
iv. The integrity and contents of the store of Incident Data shall be maintained and managed to make the most efficient use of the space available whilst optimizing data access time.
v. Notifies to Tracker Component the information of the incidence.
vi. Select the users whose preferences include tracks.
vii. Notifies to those users subscribed to the state of the affected trips.

- SendIncidenceInfo ()

5. Traffic Regulation Manager (Smart Crossing)

a. Service Initialization

i. MoveUs platform locates the user through the GPS of the personal device, obtains the Device Id and requests the user to indicate a username and a password.
ii. The platform receives the location and identification (UTM, Device Id (BT-MAC)), enables the service for the user (User Management. Services) and subscribe the user to any change of cross PoIs in the current city.
iii. The user logs in and starts the application in the smartphone; it sends its coordinates to MoveUs, and then the service is ready to be used.
iv. The static crossing information (BT detection zones and smart crossings (UTM), stored into the Urban Road Static Data (crossing information), is sent to the user and any change notified.

- SendBTSmartCrossingInfo()

b. Crossing demand coordination (regardless of the Cross system used)

i. The LTC will forward to MoveUs the information message(s) received from the BT receptor
ii. MoveUs will confirm if the Device Id belongs to a registered user or not, checking the User Management Data (User.Services), before continuing with the service.
iii. Once MoveUs has received the information and has validated the user, it will send a message back to the personal device requesting the user to confirm its intention of crossing.
iv. The answer is sent back to the platform that will subsequently send a demand for crossing to the LTC.
v. Sends the demand for crossing to the MFU (Traffic Light Controller), providing the following information: requester

user Id, crossing Id and phase Id. This information is collected from the urban Road Static and Dynamic Info (information from the nearest cross and current phase).

vi. Insert request information into the Smart Crossing Request Datastore.

vii. Receives the information of the crossing from the Traffic Light Controller: Time to green (TWG) and Maximum time on Green (TmaxG). Updates the Smart Crossing Request Data and the Urban Road Dynamic Information if applies.

viii. Traffic Light Information. During the smart crossing service activation, the MFU is sending the state of the traffic light phases: activate/extend pedestrian green phase, green phase on, green phase off and user times (time for crossing).

6. **Parking Manager**

   a. **Parking Options**

      i. The System returns the parking options near the current location. Information is extracted from Urban Parking Data.

6.3.6 MoveUs Traveler Journey Assistant

Definitions

- **Multi-modal**. Transportation performed with at least two different means of transport.
- **Leg.** Part of a journey, that can be timed (associated to a timetable), frequency (the service is deployed under a defined frequency), continuous (without timing constraints) and interchange (between different transport modes).
- **GTP.** General Trip Preferences.
- **PoI.** Point of Interest.

Users

- **Type 1 (UT4)**: Final users
  - **UT4_Anonymous**: User that wants to keep their identity anonymous.
  - **UT4Identified**: Users that provide their information to the MoveUs platform.

Functional View

The functional view of the Travel Journey Assistant is depicted in the following figure.

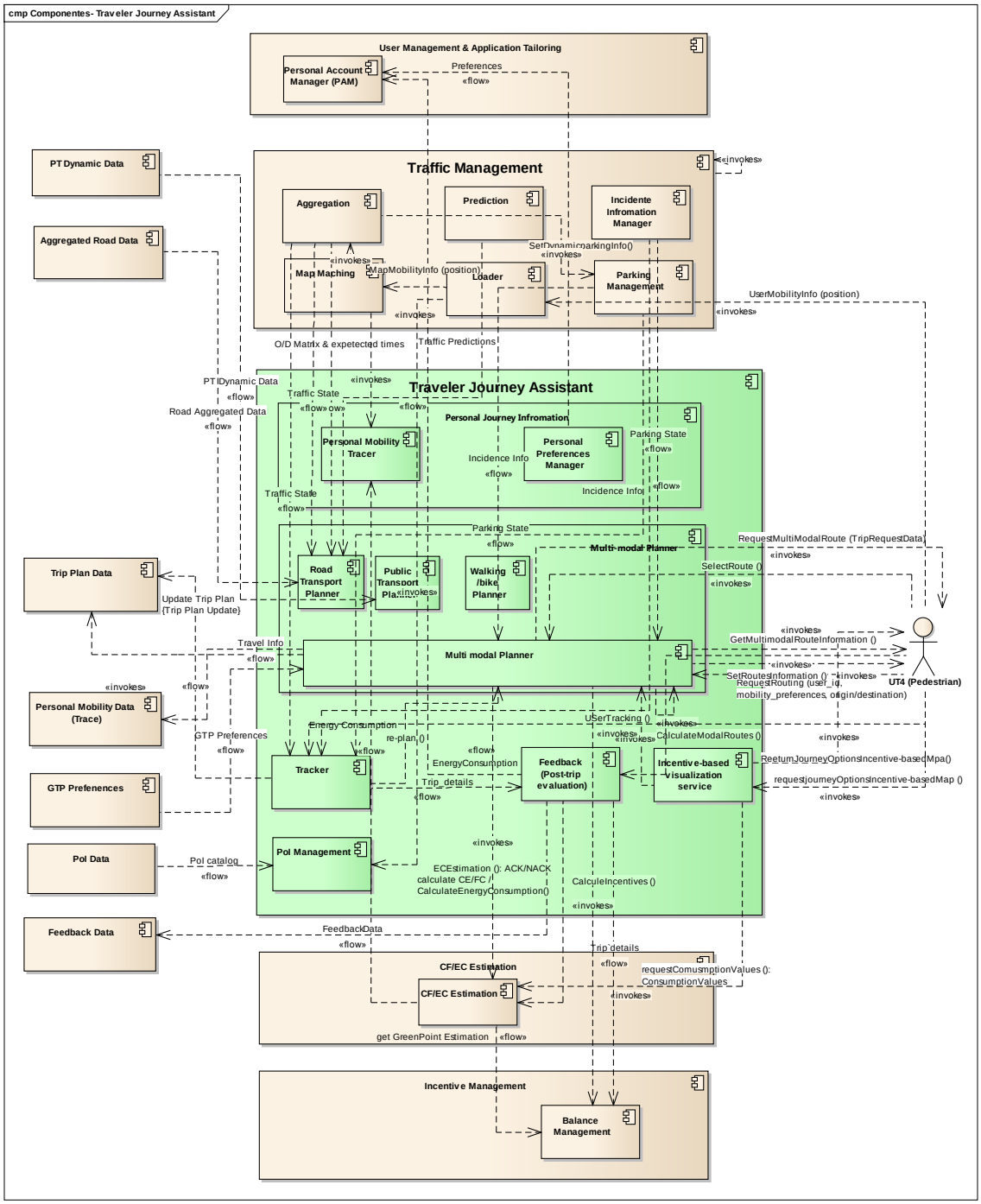


Figure 20 Journey Planning functional view (in green)

Trip Plan Data	The result of the trip planning process is used along the on-trip phase, as a reference to identify perturbations affecting estimation times and travel feasibility.
Travel Information Data	Personal Mobility Trace Data, stores the trips and traces of the MoveUs platform users.
General Trip Preferences (GTP) Data	Contains the personalised data needed to support the Traveler during all of their trips, from the trip planning, trip execution and finally, being updated once finalized.
Personal Mobility Data	Repository of historical information, hosting all the information sent by the application capture module. This information is used in the queries associated with mobility analysis. Two main blocks are identified: location traces (GPS) and track traces, where the full tracks are persistent in DB.

Table 18 Journey Planning datastores

MoveUs Modules

Personal Mobility Tracker (Personal Mobility Footprint)	Captures and manages the personal mobility trace of a traveler, composed by the GPS coordinates and a timestamp for each sequential capture.
Personal Preferences Manager	Manages mobility personal preferences (Mobility-related preferences and parameters of General User Profile ()).
Road Transport Planner	Calculates the smartest route, using actual road speed data, real-time traffic information (traffic state, expected times and incidence data). For each route, it associates a set of measures, based on which the user will make a decision.
Public Transport Planner	Calculates the smartest route using route and fare information, PT real-time information (delays, time to travel and incidence data). For each route, it associates a set of measures, based on which the user will make a decision.
Walking/Bike Transport Planner	Calculates the smartest route by walking or riding bike. No energy consumption neither carbon footprint is generated.
Other (Renting/Sharing) Transport Planner	Calculates the smartest route considering renting and sharing options (e.g. car, bike).
Multi-modal Planner	Calculates the best multi-modal journey option in terms of the different identified parameters (times, cost, energy efficiency, incentives, etc.).
Tracker	Continuously tracks the traveler/driver position, depending on which it identifies deviations from the intended trip plan and notifies the user. Traces are stored.
Post-trip Evaluator (Feedback)	Once the travel finishes, updates the personal data (preferences and incentives) according to the trace and

	provides a trip evaluation based on user feedback and quality measures.

Table 19 Journey Planning modules

Description of operations / functions / services

1. **Personal Journey information**

Although the User Management module manages all the user mobility preferences, this component realizes the calculus to speed up subsequent access in real time.

a. **Personal Mobility Tracker (Personal Mobility Footprint)**

i. **Identification of frequent trips**

In off-line mode, selects previous trips of the user from Personal Mobility Data (User_Tracks) in a predefined period of time and group by year, month, day of the month, day of the week and time (is timing slide of a predefined duration).

This information will be stored on GTP Data repository or associated to travel profile (identify MV_travellerProfile.frequentTrips) for trip suggestion. This decision will be addressed during the detailed design of the database.

b. **Multi-modal Trip Planner**

The multi-modal planner is supported by integrated/local specific-transport mode planners: road planner (traffic), public transport planner, walking/bike planner and other transport planners.

i. **Road Transport Planner**

1. **Location of nearest node.**

Given a specific position GPS coordinates), the system returns the nearest node of the reference network.

2. **Routing / Routing through some locations.**

Computation of different trajectories on the road network between two coordinates through a sequence of points, ordered according the desired optimization criteria (in terms of time, cost, energy consumption/footprint and incentives). If only one criterion is selected, the function will return the best route, which maximizes the cost function.

The number of generated trajectories is configurable.

Steps:

1. Depending of date and time of travel planning, recover current and/or predicted traffic information (Aggregated traffic Information / Predicted traffic data schemas). In the last, recovered information will correspond to the urban road static map, assigned to the current time and situation (e.g. incidence).
2. The different routes are assessed in terms of the optimization criteria. For each route segment, check its length and travel time, and accumulate at the route level. This information is obtained from the Aggregate Urban Road Map.
3. Calculate energy consumption measures, by calling RequestEnergyConsumption with each route as parameter. This information will be attached to the route (JP_leg.EnergyConsumption).
4. Calculate associate incentives, by calling CalculateIncentives (route). This information will be attached to the route (PT_Journey.MV_Incentive).
5. Select optimization criteria from User.PreferredCriteria (MV_CriteriaRating).
6. Generates a list of candidate travel plans, ordered by the selected optimization criteria.

RequestTrip (tripRequestData). As input tripRequestData = *OriginDestinationRequestType, InputPreferences* + *UserId.*

ii. Public Transport Planner

1. Routing / Routing through some locations.

Depending on the timing criteria, i.e. departure time of desired arrival time, the planner generates the best trip according to different criteria: quickest trip or fewest transfers, ensuring or minimizing the arrival time. The origin and destination must be public transport lane stops.

iii. Walking/Bike Planner

Computation of different trajectories on the road network between two coordinates through a sequence of points, ordered according the desired optimization criteria (in terms of time, cost, energy consumption/footprint and incentives). The routes will be ordered according to the user PoI (User.POI preferences) and accessibility constraints.

iv. Multi-modal Planner

The planner allows the combination of different public and private transport systems, and also considers the intermediate displacements (e.g. by bike, walking), renting/devolution of bikes, car parking and car pooling (Tentative).

If no specific preferences are introduced, search travelling preferences of userId from User schema, specifically attributes of travel preferences (from Preferences.TravelProfile.TransporRating), POI preferences and accessibility constraints, from personal data.

As output for any mode combination, the component must provide: step-by-step instructions, geo-coded data necessary for a graphical representation of the trip, the energy consumption of the trip and associated incentives.

c. Tracker

In case of setting the user the flag "vehicle tracking" of his/her profile, the system will save the current position. The position messages will be received at a configurable timely basis (e.g. 1, 5, or 10) depending on the transport mode and/or the trace configuration.

Once received a new position, if an incident notification exists, the user will be notified to enable the option of re-routing.

The trace information will be saved on Personal Mobility Data store.

i. Implement trip plan and track traveler

Continuously monitor for the receipt of the traveler GPS location, mapping its position on the reference map determining the current location of the traveler; monitoring the advance of predefined trip and detecting plan changes/deviations. Determine which part of the trip plan to implement and providing estimated travel times. Continuously monitor for receipt of the implement updated trip plan data flow.

When a new trip plan is received, store its contents internally, starting from the last know location of the Traveler.

- SendUserTracking (mobility selection, tracking) (M) (I)
- SetTravelerPosition (x,y)

ii. Monitor trip plan implementation for traveler

1. Upon any data flow containing updates of travel conditions (e.g. traffic condition changes (current and short time), incidences reported and public transport schedule delays) verify its impact on predefined and saved trip planned.
2. For each of the pending or on-going plans, check if any of its legs is affected by the deviation, and evaluate the new expected travel times.
   If the flag real-time reporting is set or vehicle is registered as green client, reports an alarm and timing.

iii. End Journey Notification

When the end of journey notification arrives, the process executes the flowing steps:

1. Update user mobility Information (trip, including energy measures and incentives).
2. Trip confirmation and storage.
3. Call Sendfeedback of the Feedback component, with the feedback information associated with the trip.

- JourneyEndNotification (trip, feedback)

d. Feedback Management

If a trip completion report for evaluations is fulfilled by a user, store this information into the Feedback Data; when not available and quality/satisfaction metrics have been previously defined by transport planners and public transport operators, report will be marked accordingly.

Evaluation must be done by segment, with the correctness of information (Y/N) at different places: destination information, last mile information, starting time information, start position information, arrival position.

Overall evaluation feedback is reported with a scale from 1 to 5 in the following concepts: quality of data, response time, quality of service, user interface, and reliability.

- SendQualityfeedback (segment, feedback correctness)
- SetTrafficFeedback (UserId, incidence, pos)
- RequestQualityfeedback (UserId)
- UpdateQualityfeedback (UserId, feedbackId)

6.3.7 Other MoveUs Modules

6.3.7.1 ***Geo-coder***

This module supports the location service: relating search strings, names, coordinates or locations in the traffic network to each.

- ResolveMobilityLocation (UserLocation): (X,Y)
- ResolveAddresss(UserLocation): (x,Y)

6.3.7.2 Electronic Wallet Registry

This module manages the information related to external payment systems, able to be used on MoveUs Transactions.

a. Available payment systems management

The platform administrator will be able to Add, update or delete any of the payment providers.

New Payment entities are create, setting its name and electronic payment details: server location network address (URL, etc.), component/service to be used for the communication.

The configuration of this information must be done by the Platform Administrator.

b. Selection of Available payment systems

The MoveUs platform manages a register of banking or payment institutions available for each user. Select available payments from User.AvailablePayments and returns connection details if known.

By default, when user is undefined, returns the full list of payment systems.

- RequestAvailablePaymentSystems (City): [ExternalLocalServices]*

6.4 Global Technical Architecture

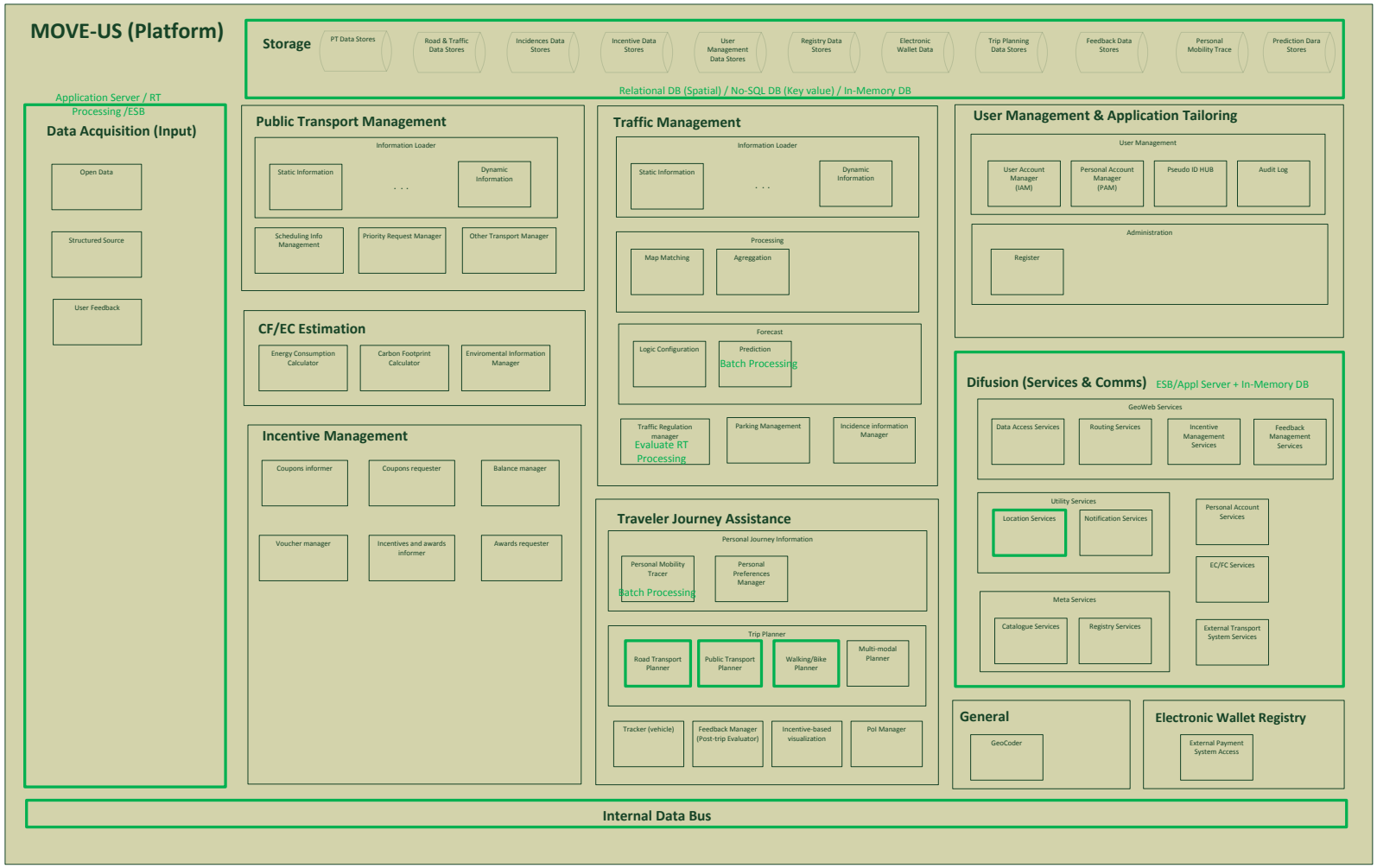


Figure 21 Global Technical Architecture (Capabilities: Storage, RT/Batch processing, middleware (EBS, messages))

6.5 Technological options

Emerging new technologies have been deployed under the term of massively parallel-processing (MPP) to address some of the issues tackled in MoveUs.

6.5.1 Storage Technologies

We need repositories capable of storing a huge volume of data and with distribution, scalability and performance characteristics. NoSQL databases present a simple, lightweight mechanism for storage and retrieval of data that provides higher scalability/availability, distribution and high performance in load and query time.

NoSQL Databases are classified according different types (Yen, 2009):

- Storage key-value: KeySpace, Flare, Schema Fre…
- Storage key-value (cached): Memcached, Repcached, Coherence…
- Storage key-value (eventually consistent): Dynamo, Voldemort, Dynomite…
- Data structure server (In-memory DB): Redis
- Object-oriented database: SopeDB, DB40, Shoal…
- Document Repository: Couch DB, Mongo DB, Scalaris …
- Column repository: BigTable, HBase, Cassandra DB, Hipertable…
- Graph-based data bases: AllegroGraph, Neo4j, Titan, Trinity, VelocityGraph

These repositories provide a persistence solution, where high performance and availability are present. For each datastore or data block, we will assess the different available technologies, in order to give response to their specific needs in terms of non-functional requirements (e.g. scalability, replication capabilities, consistency checking, reliability and data volume support). The use of any of these databases implies the migration of the relational model defined in D3.1 [7].

Cloud options

The most relevant and widely used cloud services that offer storage capabilities are listed below. The selection of storage cloud services shown below is focused in such services that offer both SQL and NoSQL storage services.

- Storage services in **Google Cloud Platform**:
  - Cloud Datastore[1], which is a service that offers a managed, NoSQL, schemaless database for storing non-relational data.
  - Cloud SQL[2], for storing and managing data using a fully-managed, relational MySQL database. Google handles replication, patch management and database management to ensure availability and performance.

[1] https://cloud.google.com/products/cloud-datastore/
[2] https://cloud.google.com/products/cloud-sql/

- Cloud Storage[3], which is a durable and highly available object storage service.

- Storage services in **Amazon Web Services**:
  - Amazon Simple Storage Service (Amazon S3), a scalable file storage in the cloud.
  - Amazon Relational Database Service (Amazon RDS), a fully managed service that offers a choice of MySQL, Oracle, SQL Server, or PostgreSQL database engines, scale compute & storage.
  - Amazon DynamoDB, fast and high-scalable NoSQL data store.
  - Amazon SimpleDB, a NoSQL database service for smaller datasets.
  - Amazon Elastic MapReduce's HBase[4], an open source, non-relational, distributed database.
  - Amazon ElastiCache, in-memory cache service.
  - Amazon RedShift, Petabyte-scale data warehouse service that makes it simple and cost-effective to efficiently analyse all data using user's existing business intelligence tools.
  - Amazon Glacier, low-cost storage for data archiving and backup.
  - Amazon Elastic Block Store (Amazon EBS), which provides block level storage volumes for use with Amazon EC2 instances. This service allows the user to create storage volumes and attach them to Amazon EC2 instances. Once attached, the user can create a file system on top of these volumes, run a database, or use them in any other way he/she would use a block device.
- Storage services in **Microsoft Azure**:
  - SQL Database, managed relational SQL Database-as-a-service.
  - DocumentDB, a highly-scalable, NoSQL document database service.
  - Cache, high-throughput, low-latency data access to build fast and scalable applications
  - Storage

In addition to the storage cloud services, a set of **virtual machines can be provisioned** from the compute services (IaaS) offered by the abovementioned cloud providers (i.e. Compute Engine[5] service in Google Cloud Platform, Amazon Elastic Compute Cloud (Amazon EC2)[6], Microsoft Azure Virtual Machines[7]). Once the virtual machines are suitably dimensioned and provisioned, MoveUs required storage datastores (SQL as well as NoSQL databases) can be installed and configured in those virtual machines.

6.5.2 Batch-processing / Analysis

Programming models are needed that are able to parallelize and manage the distribution of large volumes of data.

1. Map Reduce programming model and Hadoop

[3] https://cloud.google.com/products/cloud-storage/
[4] http://docs.aws.amazon.com/ElasticMapReduce/latest/DeveloperGuide/emr-hbase.html
[5] https://cloud.google.com/products/compute-engine/
[6] https://aws.amazon.com/ec2/
[7] http://azure.microsoft.com/en-gb/services/virtual-machines/

MapReduce is a programming model for processing large data sets, and the name of an implementation of the model by Google. MapReduce is typically used to do distribute computing on clusters of computers. The model is inspired by the map and reduce functions commonly used in functional programming.

Hadoop is an implementation of this computational paradigm.

2. Client tools for the platform Hadoop
   The following list names the main tools related to Hadoop and Big Data:

   a. Hive: Hive is a Hadoop-based data warehouse developed by Facebook.
   b. Pig: Pig Latin is a Hadoop-based language developed by Yahoo.
   c. Flume: Flume is a framework for populating Hadoop with data.
   d. Mahout: Mahout is a data mining library.
   e. Whirr: Whirr is a set of libraries that allows users to easily spin-up Hadoop clusters on top of Amazon EC2, Rackspace or any virtual infrastructure.
   f. Sqoop: Sqoop is a connectivity tool for moving data from non-Hadoop data stores, such as relational databases

3. Graph programming model like Pregel
   Pregel adopts a non-traditional programming model, for the sake of maximizing parallelism and scalability. A graph algorithm is implemented as a single computation function written in a vertex-centric, message-passing and bulk-synchronous way.

Cloud options

Due to the relevance that Big Data Analytics has achieved when it comes to deal with large volume of data, cloud service providers have updated their cloud service offer by including services based on Big Data technologies such as Apache Hadoop.

- **BigQuery[8] in Google Cloud Platform**: a columnar database to run large-scale queries in seconds.
- **Amazon Elastic MapReduce[9] (Amazon EMR) in Amazon Web Services.** Amazon EMR uses Hadoop to distribute the data and process it across a resizable cluster of Amazon EC2 instances.
- **HDInsight[10] in Microsoft Azure**, a Hadoop distribution powered by the cloud. This means HDInsight is architected to handle any amount of data, scaling from terabytes to petabytes on demand.

In addition to the existing Big Data analytics cloud services, a set of virtual machines can be provisioned for this kind of analytics purposes by using cloud computing services offered by the primary cloud service providers (as explained in the previous section, see section 6.5.1). For instance, a virtual machine or set of virtual machines can be allocated and Apache Hadoop open-source framework can be installed and configured, as it is explained in this guide[11] which helps a user learn how to configure and use a Hadoop cluster on Google Cloud Platform by using Google Compute Engine and Google Cloud Storage.

[8] https://cloud.google.com/products/bigquery/
[9] https://aws.amazon.com/elasticmapreduce
[10] http://azure.microsoft.com/en-gb/services/hdinsight/
[11] https://developers.google.com/hadoop/?_ga=1.182239712.658518425.1410939095

6.5.3 Real Time Analytics

Here, we can find solutions able to deal with timing constraints of data analysis process and storage. Process must be done in real-time. Two main options: CEP (Complex Event Processing) and IMDG (In-Memory-Data Grids). CEPs capture information from messages streams, databases or applications in real time, also defines taxonomies of interrelated events, supporting their identification and automatic actions or warnings (e.g. alarms). Several solutions are currently in the market (e.g. WSO2, Esper, Siddhi, Drools Fusion and others), with different transport protocol, data format, runtime engines and pattern/event definition richness. Although, commonly web-service oriented, new versions and plugins, extend their capabilities to high volume data analysis.

Also, in the last times, we can find a new concept IMDG (In-Memory Data Grid), able to process huge datasets in real time, using the system main memory as storage. This is specially indicated for local and volatile data, being distributed as plugins or libraries for reference applications servers. Some examples are: VMWare Gemfire, Oracle Coherence3, IBM extreme Scale, Terracota or JBoss Infiinispan.

Also there exists several platforms specific for real-time data analysis, some reference samples are the following: Teradata, a parallel massive processing system, based on the concept of "shared nothing", Hadapt based on Hadoop, manages structured an also non-structured data, Storm & Kafka, characterized by high fault tolerance and scalability, identifies several streaming steps: collection, transportation and process and finally, Spark & Shark, with similar capabilities but implements a functional development approach. Other options are: HP Vertica, Brisk, Parstream.

Cloud options

Real time data processing needs have produced the appearance of cloud services related to real time analytics lately.

- **Amazon Kinesis[12] in Amazon Web Services**, a service for real-time processing of streaming data at massive scale. Amazon Kinesis can collect and process hundreds of terabytes of data per hour from hundreds of thousands of sources, allowing the user to easily write applications that process information in real-time. Data can be sent to a variety of other AWS services such as Amazon Simple Storage Service (Amazon S3), Amazon DynamoDB, or Amazon Redshift.
- **Big Compute in Microsoft Azure**, which provides on-demand compute resources that enable the user to run large parallel and batch compute jobs in the cloud, by provisioning high performance A8 and A9 compute instances on Azure (i.e. by allocating high performance capacities virtual machines from the IaaS offer in Azure).

The last cloud solution is also applicable to any IaaS cloud service that offers high performance capabilities. Once high performance characteristics virtual machines are provisioned, MoveUs required real time analytics solutions can be installed and

[12] http://aws.amazon.com/kinesis/

configured in them. E.g. this guide[13] explains how to install and configure Shark in Amazon EC2.

6.5.4 Enterprise Service Bus (ESB) / Application Servers

MoveUs software architecture is based in a service oriented architecture or SOA. City pilots and their city partners allow and provide access to their information and functionalities using services. MoveUs applications and users consume functions and data accessing to services.

An enterprise service bus allows the communication between processes in a service oriented architecture where providers and consumers use a common layer to communicate with and avoid "wired" connections between consumers and providers.

There are several solution providers for ESB in open source or community license modes:

- **Mule ESB.** MuleSoft is a company with a very well established ESB product called Mule ESB. They have a community version with a set of the functions of their enterprise product. The Anypoint Studio developer tools are graphical tools that allow a fast deployment of services, mediations and transformations. The community have also some free connectors to popular services. Some drawbacks of the community edition are the lack of deployment management, performance analysis tools and high availability. MuleSoft has also other middleware products but all of them in the enterprise edition.

- **Red Hat JBoss Fuse Service Works**. Red Hat JBoss middleware solution that includes the Core ESB, an evolution of the Fuse ESB, a development framework and a service governance.

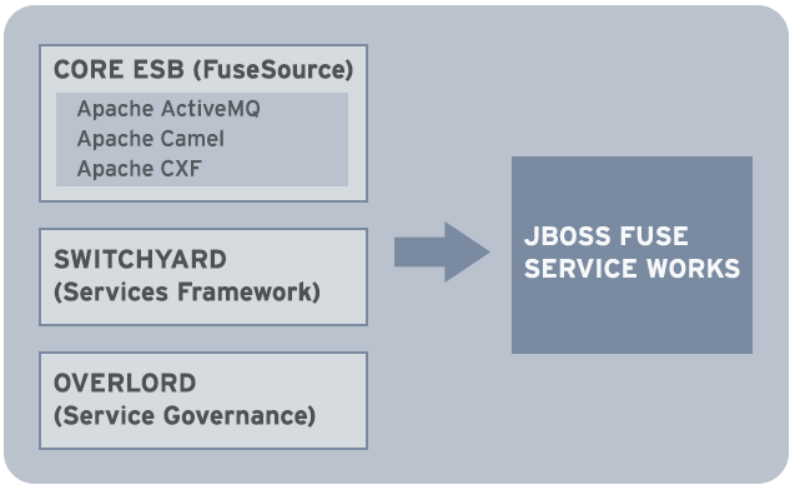


Figure 22 JBoss Fuse scheme

[13] https://github.com/amplab/shark/wiki/Running-Shark-on-EC2

JBoss Fuse Service Works adds to the ESB core: structured service development framework, service orchestration, rules processing, service governance and business transaction monitoring.

The product is well documented in the open source mode.

- **WSO2 ESB.** WSO2 seems to have an open source product for every need in middleware platforms, all developed around a common foundation that they call WSO2 Carbon. The ESB product is called WSO2 ESB and includes tools for develop, deploy, test, management and performance analysis.

  A representation of the set of middleware products appears in the following image taken from their web site:

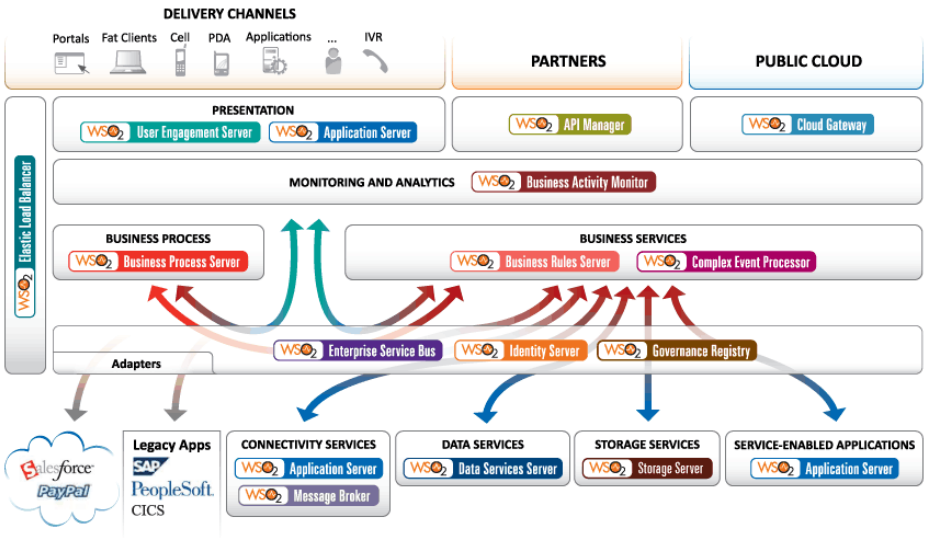


Figure 23 WSO2 ESB

- **Talend ESB.** Talend is a company with a set of open source products for a middleware platform. Talend, like MuleSoft offers open source products vs enterprise products, having the open source products a set of the enterprise's functionalities. In the case of Talend, the open source products are not reduced to the ESB, having products for:

  - Big Data integration.
  - Data Integration.
  - Data Quality.
  - Master Data Management (MDM).
  - Enterprise Service Bus (ESB).
  - Business Process Management (BPM).

Cloud options

There are many cloud services that offer application server solutions, since the most PaaS cloud services provide a deployment infrastructure for applications created by the user using programming languages, libraries, services, and tools supported by the provider.

- **App Engine[14] in Google Cloud Platform**. Support for some of the most popular programming languages (Python, Java, PHP and Go) and existing frameworks such as Django, Flask, Spring and webapp2. It provides auto scalability to 7 billion requests per day.
- **Azure websites[15] in Microsoft Azure.** It is a fully managed Platform-as-a-Service (PaaS) that enables the user to build, deploy and scale enterprise-grade web applications. Support for ASP.NET, Java, PHP, Node.js or Python. Moreover Azure includes compatible solutions provided by third partners that work with Microsoft Azure and that the user can purchase depending on web application's needs; e.g. support for MySQL, DocumentDB, Search, MongoDB, Redis, and Azure Table Storage.
- **Openshift Online[16]**. OpenShift Online is Red Hat's public cloud application development and hosting platform that automates the provisioning, management and scaling of applications.

Regarding ESB cloud solutions, there are no many in the cloud market. But the most relevant are the following ones:

- **CloudHub[17]**. It is an enterprise-class integration platform as a service (iPaaS) that securely connects SaaS and on-premise applications at cloud speed. Created by the Mule ESB owners.
- **Jboss FUSE Enterprise Service Bus by Openshift Online**

As it has been explained in the previous sections, a set of virtual machines can be allocated by using IaaS services offered by the primary cloud service providers (e.g. Compute Engine[18] service in Google Cloud Platform, Amazon Elastic Compute Cloud (Amazon EC2)[19], Microsoft Azure Virtual Machines[20]) in order to install and configure specific ESB software required by MoveUs. This allows also to deploy in what now is called as hybrid environments, with systems deployed as IaaS (ESB) and systems deployed as PaaS (provider services, storage services, database services, …).

6.5.5 Message Queue Systems

RabbitMQ implements a broker architecture, meaning that messages have persistence and are queued on a central node before being sent. This approach makes RabbitMQ very easy to use and deploy, enabling advanced functionalities. This point reduces its scalability and slows the interactions.

ZeroMQ is a very lightweight messaging system specially designed for high throughput/low latency. The programmer must implement advanced functions as persistence by combining various pieces of the framework, as a counterpart Zmq is very flexible.

[14] https://cloud.google.com/products/app-engine/
[15] http://azure.microsoft.com/en-gb/services/websites/
[16] https://www.openshift.com/products/technologies
[17] https://www.mulesoft.com/platform/saas/cloudhub-ipaas-cloud-based-integration
[18] https://cloud.google.com/products/compute-engine/
[19] https://aws.amazon.com/ec2/
[20] http://azure.microsoft.com/en-gb/services/virtual-machines/

ActiveMQ Integrates the advantages of the previous solutions: can be deployed with both broker and P2P topologies, while eases the deployment of functionalities beyond the message exchange.

Finally, the three options provide an APIs for the most common languages (C++, Java, .Net, Python, Php, Ruby, …), are documented and are actively supported by a community.

Cloud options

There are some cloud solutions that provide message queue on the cloud, such as:

- **IronMQ**[21]**.** It is an easy-to-use highly available message queuing service, which is built for distributed cloud applications with critical messaging needs. It provides on-demand message queuing with advanced features and cloud-optimized performance.
- **Amazon Simple Queue Service (SQS)**[22]. It is a fast, reliable, scalable, fully managed message queuing service provided by AWS.
- **Openshift Online** is a PaaS solution that includes message queue system.

In addition to the existing cloud solutions, required message queue systems can be deployed in virtual machines provisioned by MoveUs project.

Moreover, in case the MoveUs cloud platform is deployed in a private cloud infrastructure managed by the MoveUs project, there are more options such as Zaqar[23] system which is included into OpenStack[24] IaaS technology.

[21] http://www.iron.io/mq
[22] http://aws.amazon.com/sqs/
[23] https://wiki.openstack.org/wiki/Zaqar
[24] http://www.openstack.org/

6.5.6 Summary of Cloud options

As it is shown in previous sections, there are plenty of public cloud services that provide technological solutions for the MoveUs project needs. The next table shows a summary of these cloud services.

Services/ Capabilities	Google Cloud Platform	Amazon Web Services	Microsoft Azure	Openshift Online	Fi-WARE
Storage services					
SQL	Cloud SQL	Amazon RDS	SQL Database		GE GIS Data Provider (GeoServer 3D) GE POI Data Provider (POI-DP)
NoSQL	Cloud Datastore	Amazon Elastic MapReduce's HBase Amazon DynamoDB Amazon SimpleDB	DocumentDB		GE BigData Analysis (Hadoop, Cassandra)
Others	Cloud Storage	Amazon S3 Amazon ElastiCache Amazon RedShift Amazon Glacier Amazon Elastic Block Store	Storage Cache		GE In-Memory DB (SAP HANA (adaptor))
Batch processing and analysis					
Hadoop-based		Amazon Elastic	HDInsight		GE BigData Analysis

Services/ Capabilities	Google Cloud Platform	Amazon Web Services	Microsoft Azure	Openshift Online	Fi-WARE
		MapReduce			(COSMOS)
Others	Big Query				
Real time analytics		Amazon Kinesis	Big Compute		GE CEP (IBM Proactive Technology Online) GE Gateway Data Handling (EspR4FastData) GE Stream-oriented (Kurento)
Enterprise Bus Systems				Jboss FUSE Enterprise Service Bus	GE Publish/Subscribe Context Broker (ORION)
Application servers	App Engine		Azure websites	Jboss	Apache Tomcat (internal)
Message Queue Systems		Amazon Simple Queue Service			REST
Virtual machines (IaaS)	Compute Engine	Amazon Elastic Compute Cloud	Microsoft Azure Virtual Machines		GE IaaS Data Center Resource Management (IBM Implementation) GE Monitoring (TID Implementation)
PaaS Manager	Appengine		Windows *Azure* PaaS		GE PaaS Manager –

Services/ Capabilities	Google Cloud Platform	Amazon Web Services	Microsoft Azure	Openshift Online	Fi-WARE
					(Pegasus)
Data location	US, Europe, Asia[25]	US Standard, US West (Oregon), US West (Northern California), EU (Ireland), Asia Pacific (Singapore), Asia Pacific (Tokyo), Asia Pacific (Sydney), South America (Sao Paulo), and GovCloud (US) regions [26]	United States: US Central (Iowa), US East (Virginia), US East 2 (Virginia), US North Central (Illinois), US South Central (Texas), US West (California) Europe: Europe North (Ireland), Europe West (Netherlands) Asia Pacific: Asia Pacific Southeast (Singapore), Asia Pacific East (Hong Kong) Japan: Japan East (Saitama Prefecture), Japan West (Osaka Prefecture) Brazil: Brazil South (Sao Paulo State), One-way replication to US South Central (Texas)[27]	US-East Region (in the Amazon AWS)[28], Europe	Malaga, Seville and Santander among others.

Table 20 Summary of cloud options

[25] https://developers.google.com/compute/docs/zones
[26] http://aws.amazon.com/about-aws/global-infrastructure/
[27] http://azure.microsoft.com/en-us/support/trust-center/privacy/
[28] https://www.openshift.com/forums/openshift/server-location

In case MoveUs requires the provisioning of IaaS (virtual machines) for the installation and configuration of specific technological solutions (because of the lack of these services in the current cloud market or due to the conditions are not adequate), the MoveUs cloud platform administrator must control operating systems, storage systems, and deployed applications; and limited control of select networking components. The MoveUs cloud platform administrator has also to be in control of the needed configuration related to the (auto) scalability of the virtual machines.

As other cloud architecture option, MoveUs cloud platform can be deployed into a private cloud infrastructure managed by MoveUs project by using IaaS and PaaS technologies such as OpenStack (IaaS), Red Hat CloudForms (IaaS), OpenShift by Red Hat (PaaS) and CloudFoundry (PaaS). Once the required IaaS and PaaS technologies are installed, the needed programs on storage, Big Data analytics, ESBs, etc. can be installed and configured over IaaS and/or PaaS layers.

6.6 City-specific Technical Architecture

This section presents the technical architecture tailored for each city pilot.

6.6.1 Madrid Technical Architecture

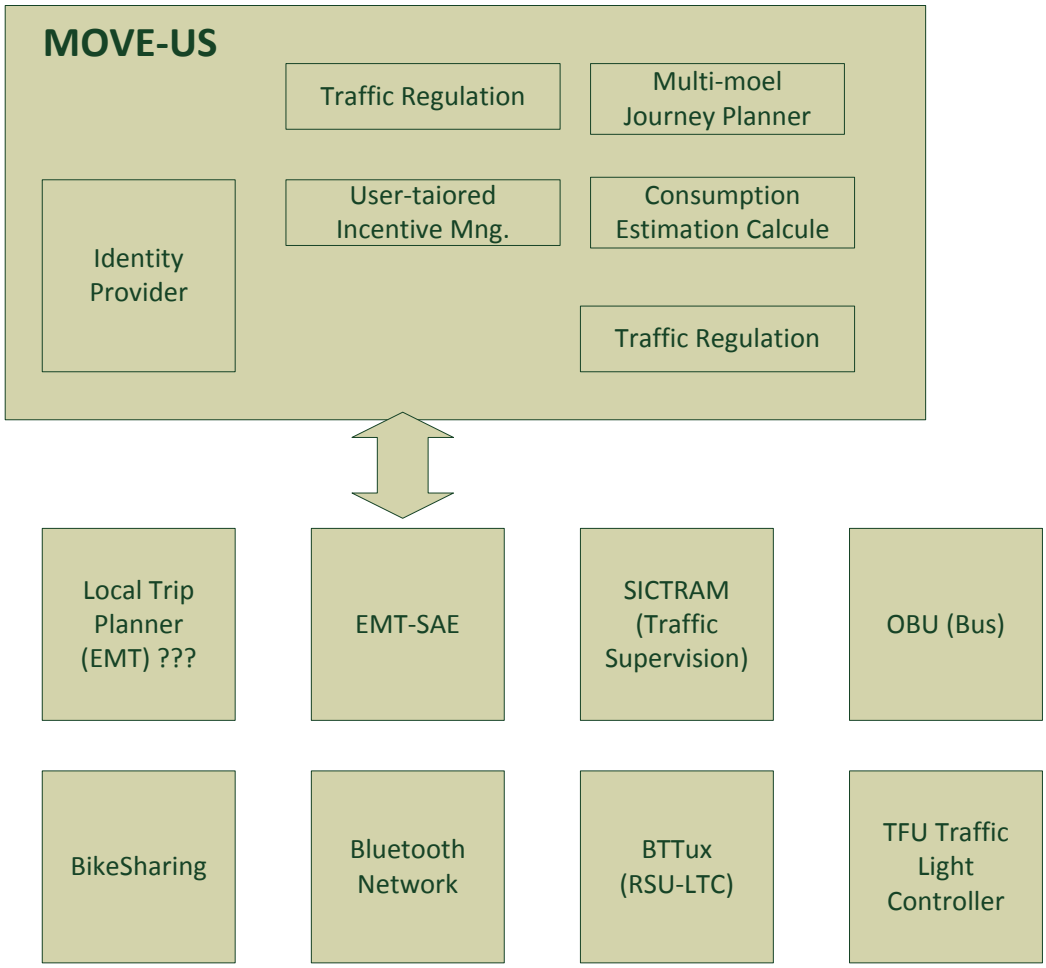


Figure 24 Madrid Technical Architecture

6.6.2 Genoa Technical Architecture

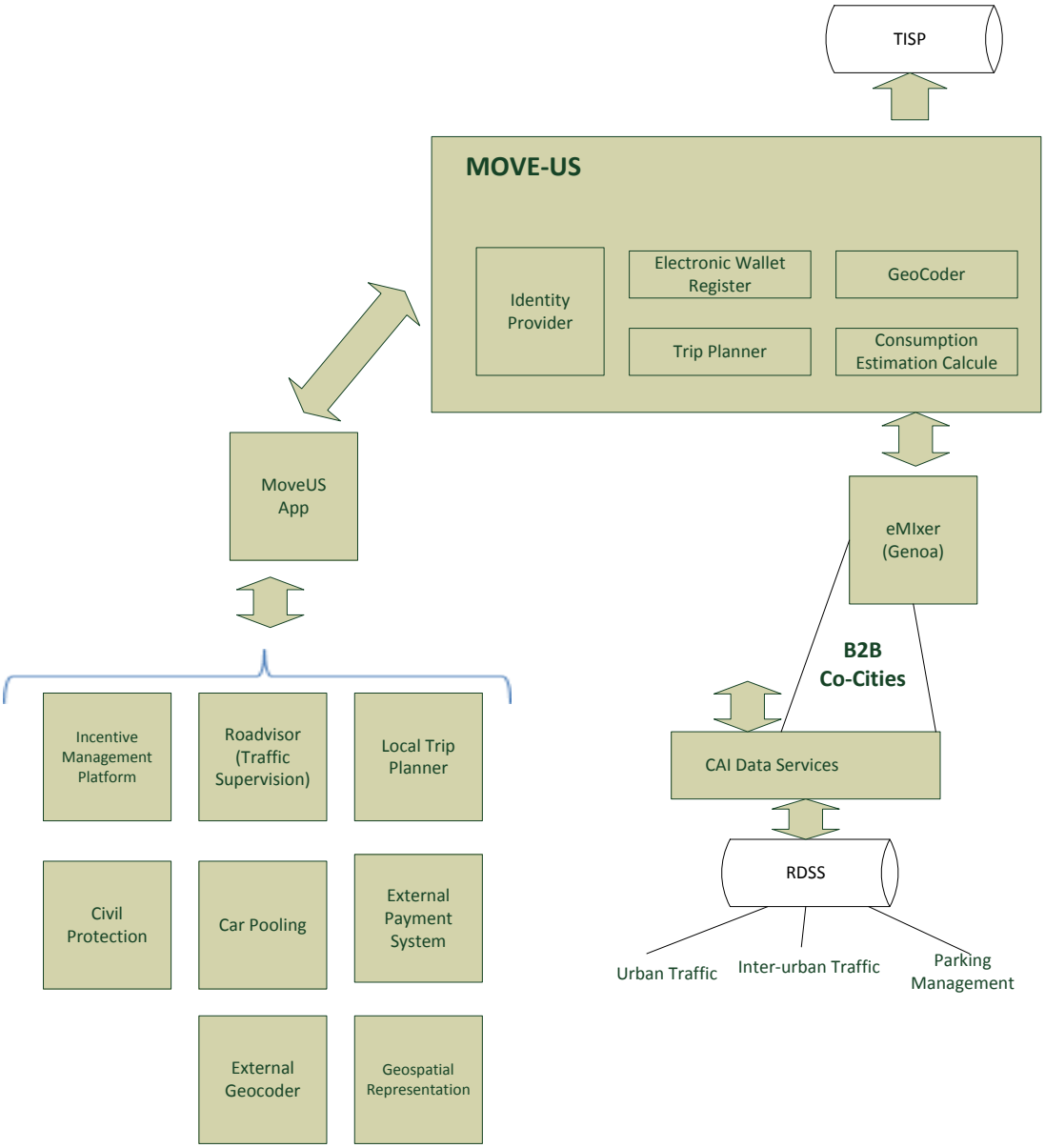


Figure 25 Genoa Technical Architecture

6.6.3 Tampere Technical Architecture

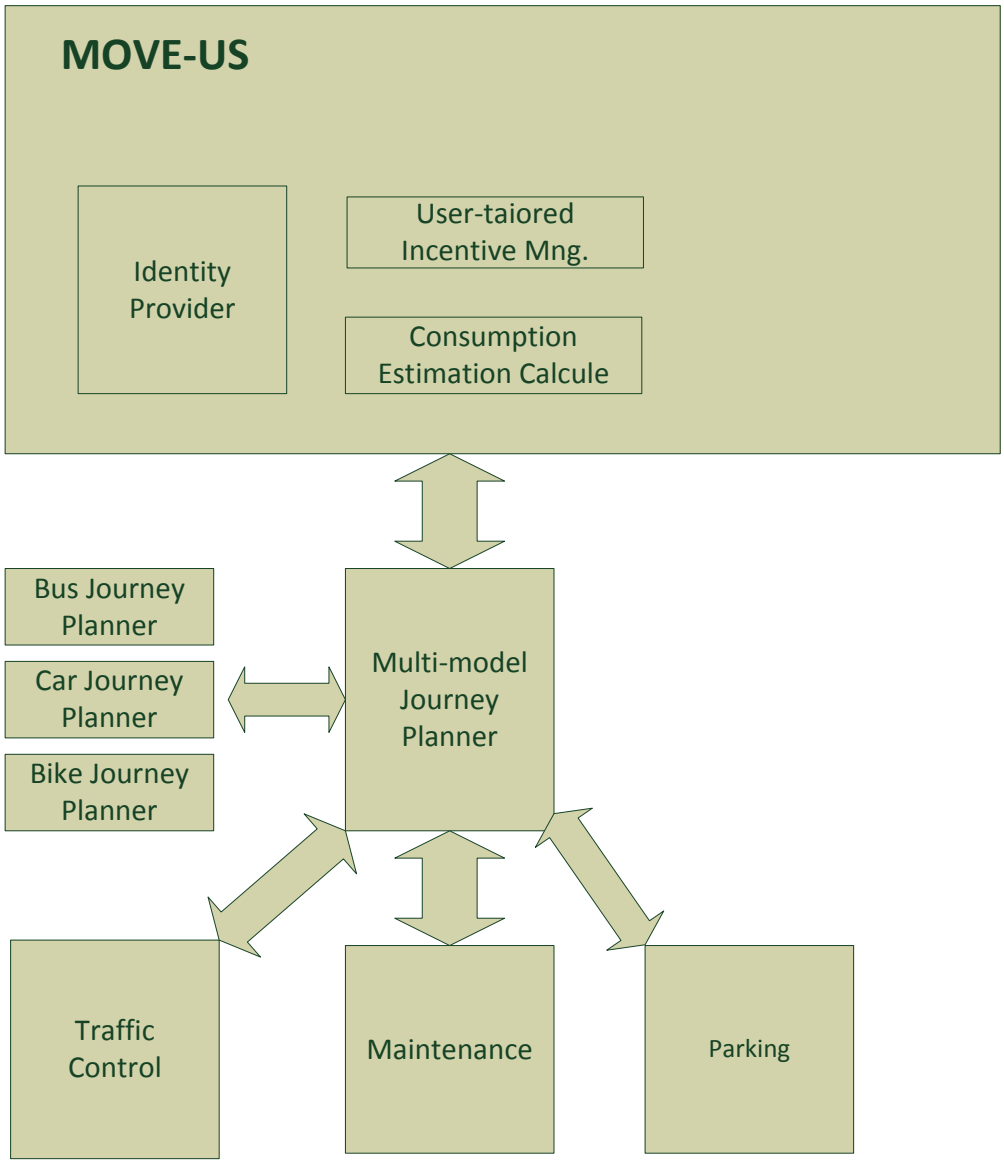


Figure 26 Tampere Technical Architecture

7 Physical/Deployment Viewpoint

Description of physical/deployment viewpoint elements.

7.1 Global Technical Architecture

The figure below shows the global deployment model that takes into account elicited non-functional requirement. This deployment model points out the physical systems that the technical components (both processing and data systems) require to be deployed.

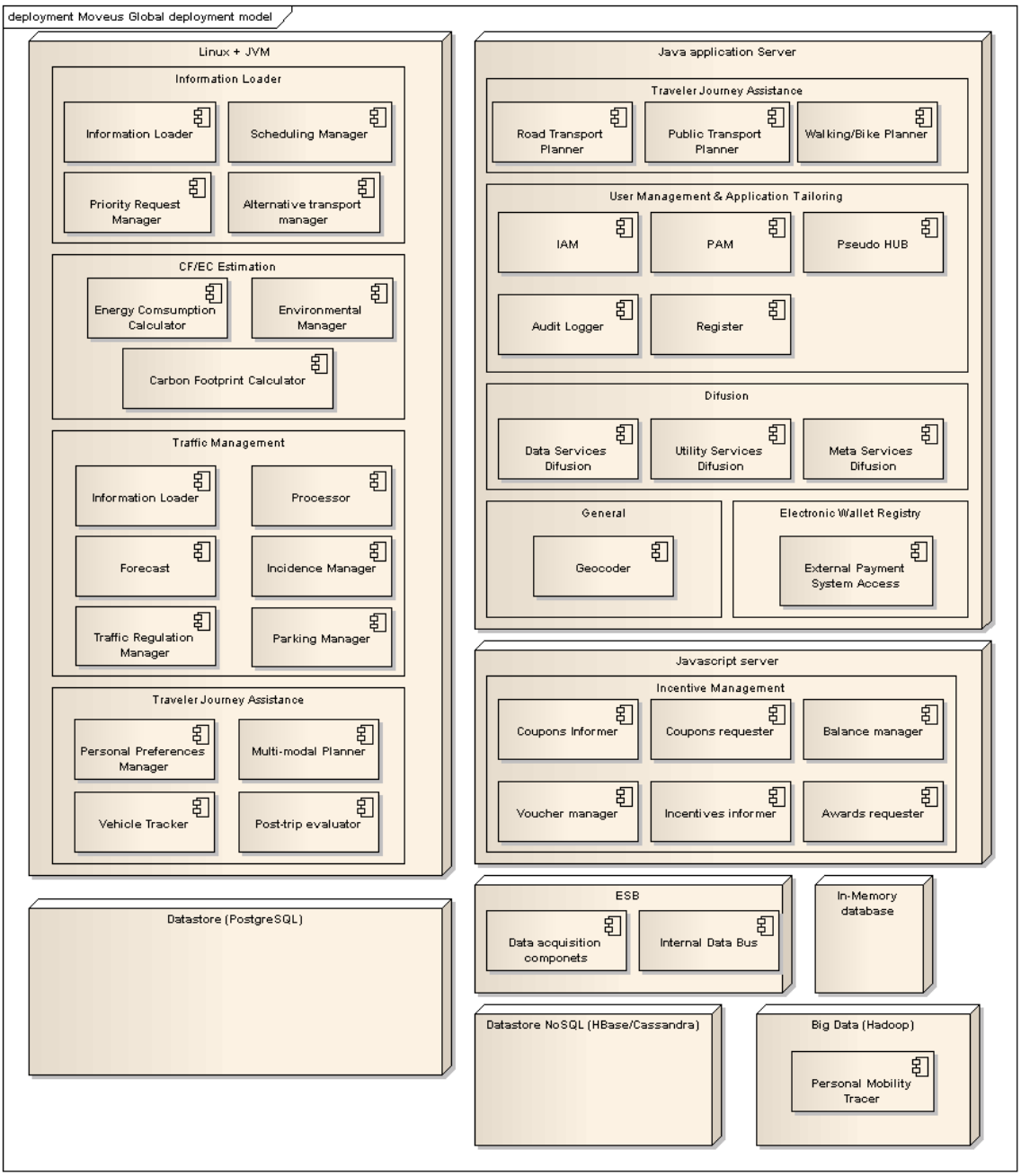


Figure 27 Global Technical Architecture

To sum up the **required deployment platform is composed of the following physical systems**:

- At least 1 machine with Linux operating system and Java Virtual Machine installed
- 1 Java application server such as JBoss.
- 1 Javascript server such as Node.js.
- 1 ESB
- 1 SQL datastore, which is required to be Spatial Database. At least 4 instances are required.
- 1 NoSQL datastore, which is required to be HBase (preferred) or Cassandra.
- 1 Big Data system, which is required to be Hadoop.
- 1 In-memory database

7.2 Cloud deployment options

This section describes the different cloud deployment options based on the MoveUs global deployment model defined previously.

7.2.1 Public cloud infrastructure based architecture

Given the vast variety of public cloud services (see section 6.5), this section describes the most likely cloud scenarios based on the non-functional requirements that the technical components have.

7.2.1.1 Public PaaS based architecture

Based on the list of the required physical systems aforementioned in section 7.1 and based on the public PaaS services offered by the most relevant cloud service providers, we can define a PaaS based architecture as it is shown as follows.

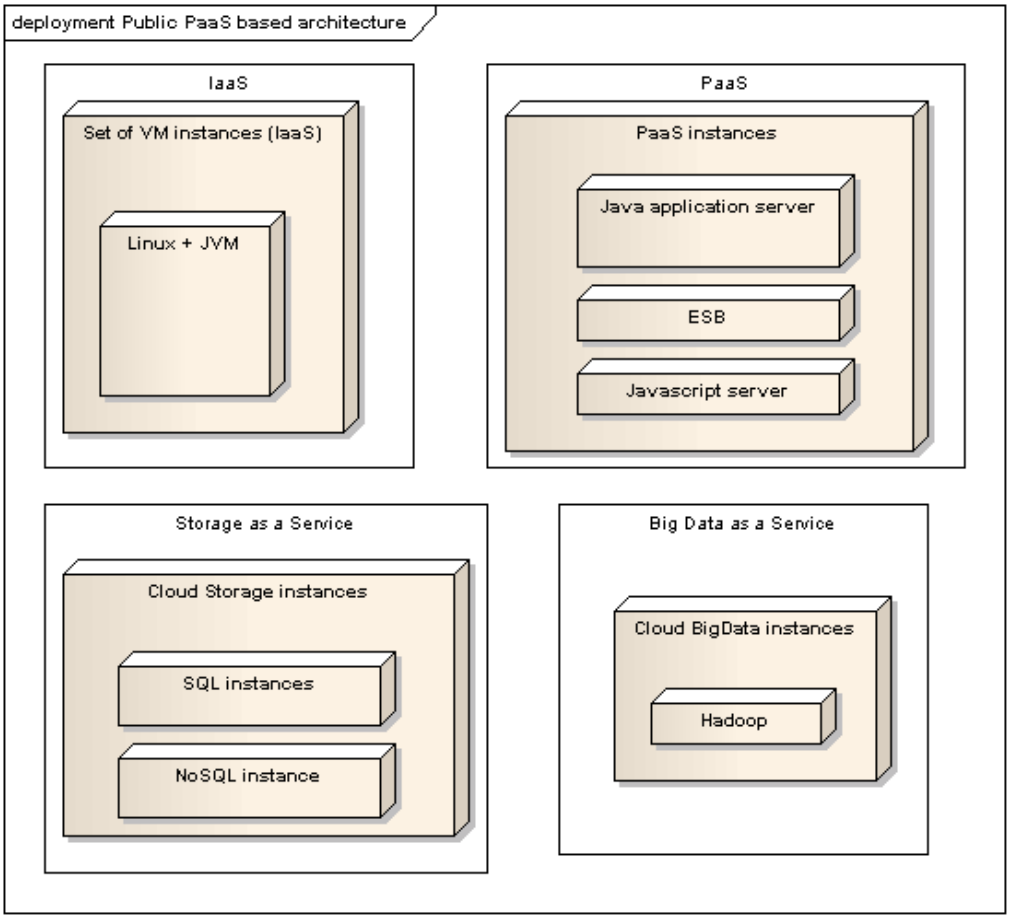


Figure 28 Public PaaS based architecture

The public cloud services that best fulfil the requirements of the MoveUs components for the PaaS based architecture are the following:

- Public PaaS candidates: OpenShift Online that includes java application server, javascript server and ESB.
- Public IaaS candidates: Google Cloud Platform's Compute Engine service, Amazon Elastic Compute Cloud (Amazon EC2), Microsoft Azure Virtual Machines.
- Public Storage as a Service candidates: Amazon Relational Database Service that offers PostgreSQL, Amazon Elastic MapReduce's HBase[29].
- Public Big Data as a Service candidates: Amazon Elastic MapReduce, Microsoft Azure HDInsight.

7.2.1.2 Public IaaS based architecture

Unlike the PaaS based architecture, the public IaaS based architecture lacks the provisioning of the PaaS infrastructure. Instead, a set of virtual machines are provisioned in order to install and configure there the systems needed for application deployment and execution (i.e. java application server, javascript server and ESB). There are two possible solutions in that case:

[29] http://docs.aws.amazon.com/ElasticMapReduce/latest/DeveloperGuide/emr-hbase.html

1. To build and install a PaaS over a set of virtual machine instances (IaaS) in order to be able to scale and update the MoveUs own platform.
2. To install, configure and maintain each system separately, i.e. the java application server, the javascript server and the ESB, over a set of virtual machine instances (IaaS).

In the first solution the installed PaaS technology offers scaling and updating capabilities, whereas in the second case the scalability capability is given and limited by the scalability functions configured in the IaaS.

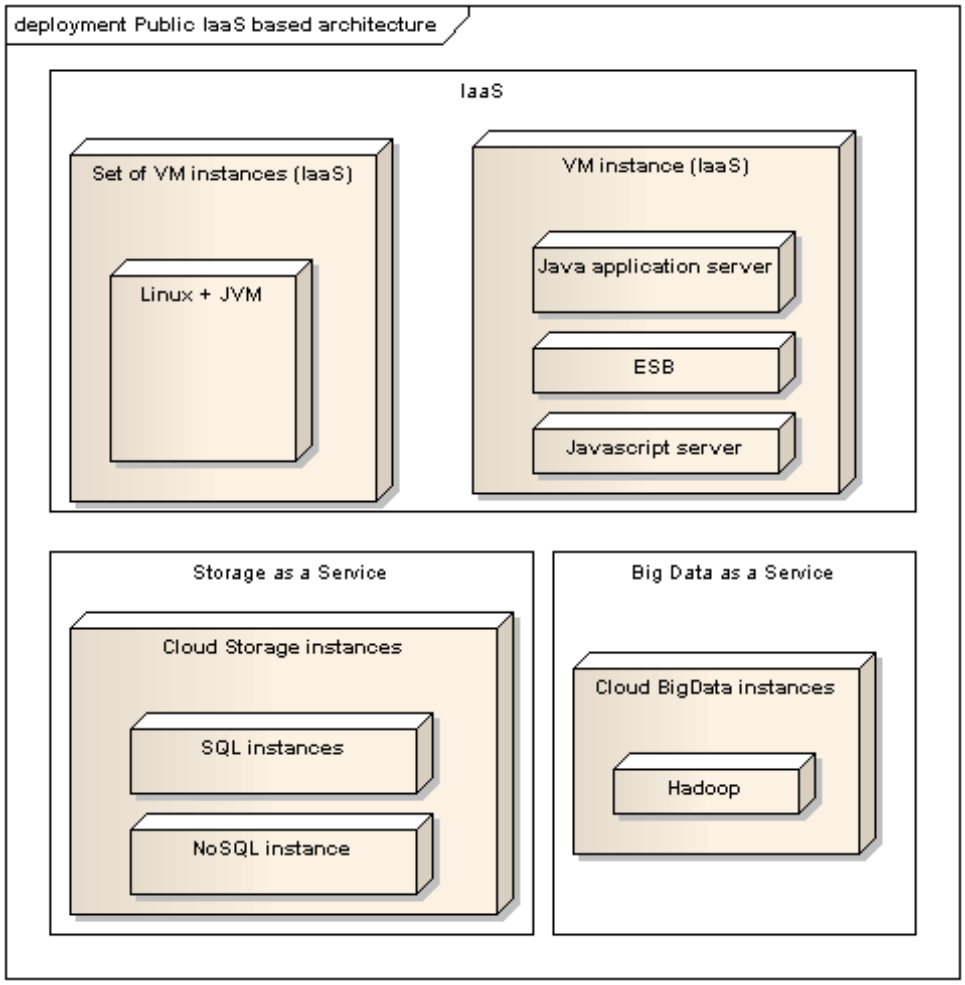


Figure 29 Public IaaS based architecture

The public cloud services that better fulfil the requirements of the MoveUs components for the IaaS based architecture are the following:

- Public IaaS candidates: Google Cloud Platform's Compute Engine service, Amazon Elastic Compute Cloud (Amazon EC2), Microsoft Azure Virtual Machines.
- Public Storage as a Service candidates: Amazon Relational Database Service that offers PostgreSQL, Amazon Elastic MapReduce's HBase.
- Public Big Data as a Service candidates: Amazon Elastic MapReduce, Microsoft Azure HDInsight.

The candidate PaaS technologies are: OpenShift community by Red Hat, CloudFoundry.

7.2.2 Private Cloud infrastructure based architecture

In the private cloud infrastructure based architecture, the whole MoveUs platform will be based on the IaaS technology installed by the administrator such as Openstack. The rest of the storage, big data, execution environment, etc. systems that are required will be installed and configured over virtual machines provisioned and controlled through the IaaS technology.

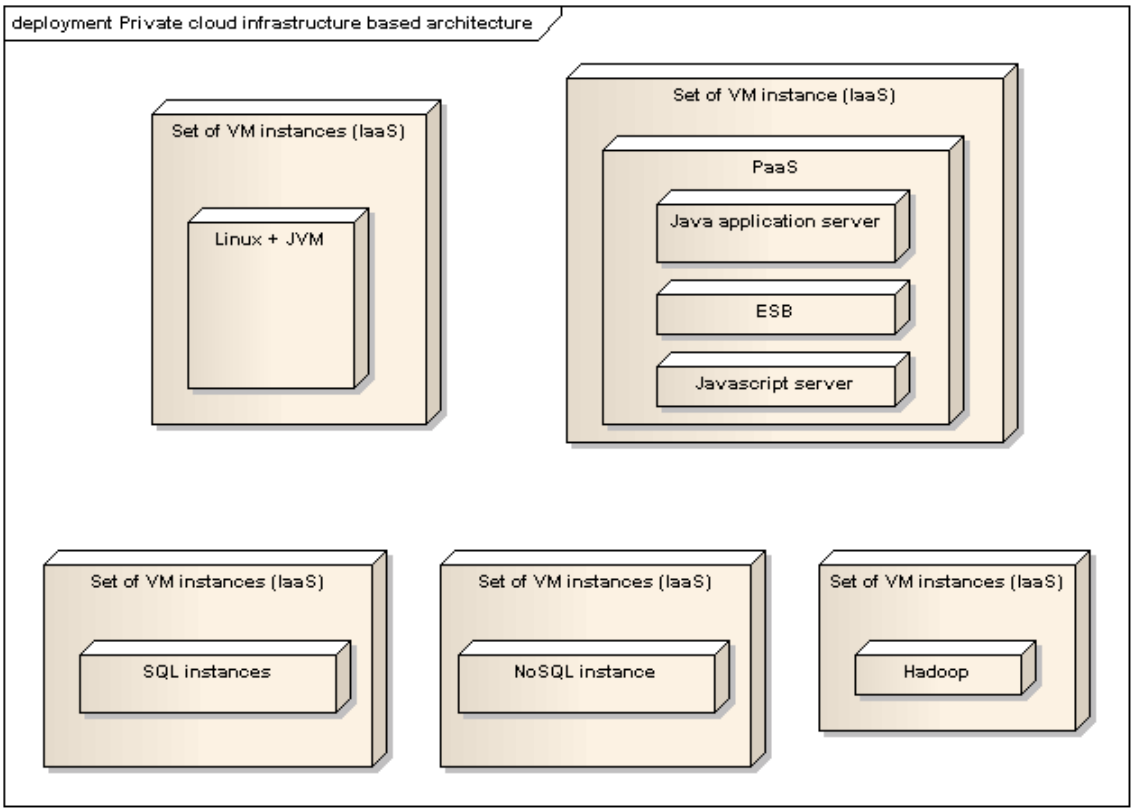


Figure 30 Private Cloud Infrastructure-based

The technologies to be installed and therefore the technologies which will part of the private cloud infrastructure are the following:

- IaaS technology candidates: OpenStack
- PaaS technologies candidates: OpenShift community by Red Hat, CloudFoundry
- SQL technologies: PostgreSQL (required)
- NoSQL technologies: HBase (required)
- Big Data technologies: Hadoop (required)

7.2.3 Hybrid Cloud infrastructure based architecture

As it is defined by the NIST, the hybrid cloud deployment model is *a composition of two or more distinct cloud infrastructure (private, community, or public) that remain unique entities, but are bound together by standardized or proprietary technology that enables data and application portability* [9].

In case of the MoveUs architecture, the hybrid cloud infrastructure that best fulfils the requirements is the one in which high performance data processing is deployed into the public cloud whereas the processes, services and applications bound to low performance demand are deployed into the private cloud installed and managed by MoveUs.

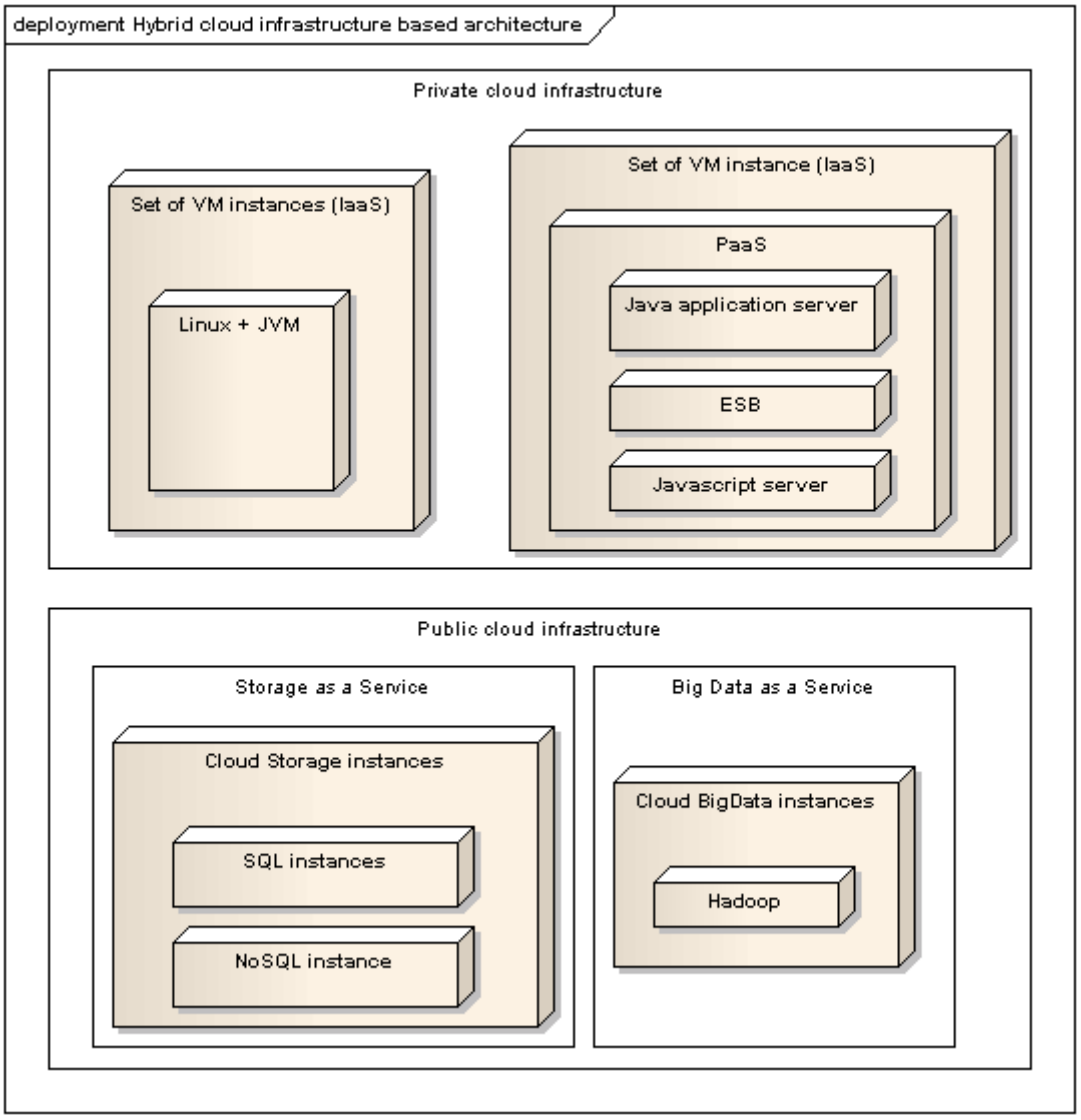


Figure 31 Hybrid Cloud infrastructure based architecture

However, in order to address privacy and security issues (see D3.4) an alternative hybrid cloud infrastructure can be implemented, in which the personal data is stored in a private cloud infrastructure (both on and off premises can be evaluated, and can be operated by the organization, a third party or a combination of them [9]).

8 Conclusions

This deliverable addresses the definition of the MoveUs cloud-based mobility management platform. On the one hand, primary requirements have been considered such as the integration of heterogeneous information and the integration of advanced algorithms to manage and add value to these data, while ensuring high availability, scalability and real time response. On the other hand, a commitment has been made by the project to allow developers and 3rd parties to exploit the stored information and spawn new business models.

Along the task execution, candidate reference architectures and projects with similar challenges to MoveUs project, both technological and also functional, have been identified and analyzed. Nevertheless, the use cases and defined city services highlight existing gaps, and force to integrate approaches, concepts and outcomes from different system engineering domains.

The parallel execution of the definition activities: system architecture (T3.2), city service specifications (T3.3), *security/privacy implications (T3.4)* and the platform algorithms (e.g. calculation of energy efficiency, trip planner, data fusion and prediction) have motivated an iterative working approach during this phase of the project.

Although the definition process has been technologically neutral, the definition of a technical/architecture identifying the key technologies able to deal with the needs of MoveUs has been achieved. This high level view is needed as a basis of any cloud provider comparison and assessment. In particular, storage technologies, batch-processing /analysis, real time analysis and internal message queue systems are explored, as well as their capacity of being deployed in cloud environments.

Final technology and cloud deployment choices will be part of WP5, but the basis and the information to make decisions are already present in this document.

9 References

[1] eMotion Project. http://www.emotion-project.eu/

[2] In-Time Project. http:// www.in-time-project.eu/

[3] Co-cities Project. http://www.co-cities.eu/

[4] National ITS Architecture (U.S. DoT). http://www.its.dot.gov/arch/

[5] FRAME Framework. http://www.frame-online.net

[6] MoveUs deliverable D2.2 - Use cases, incentives-based model concept and common specifications for the pilots

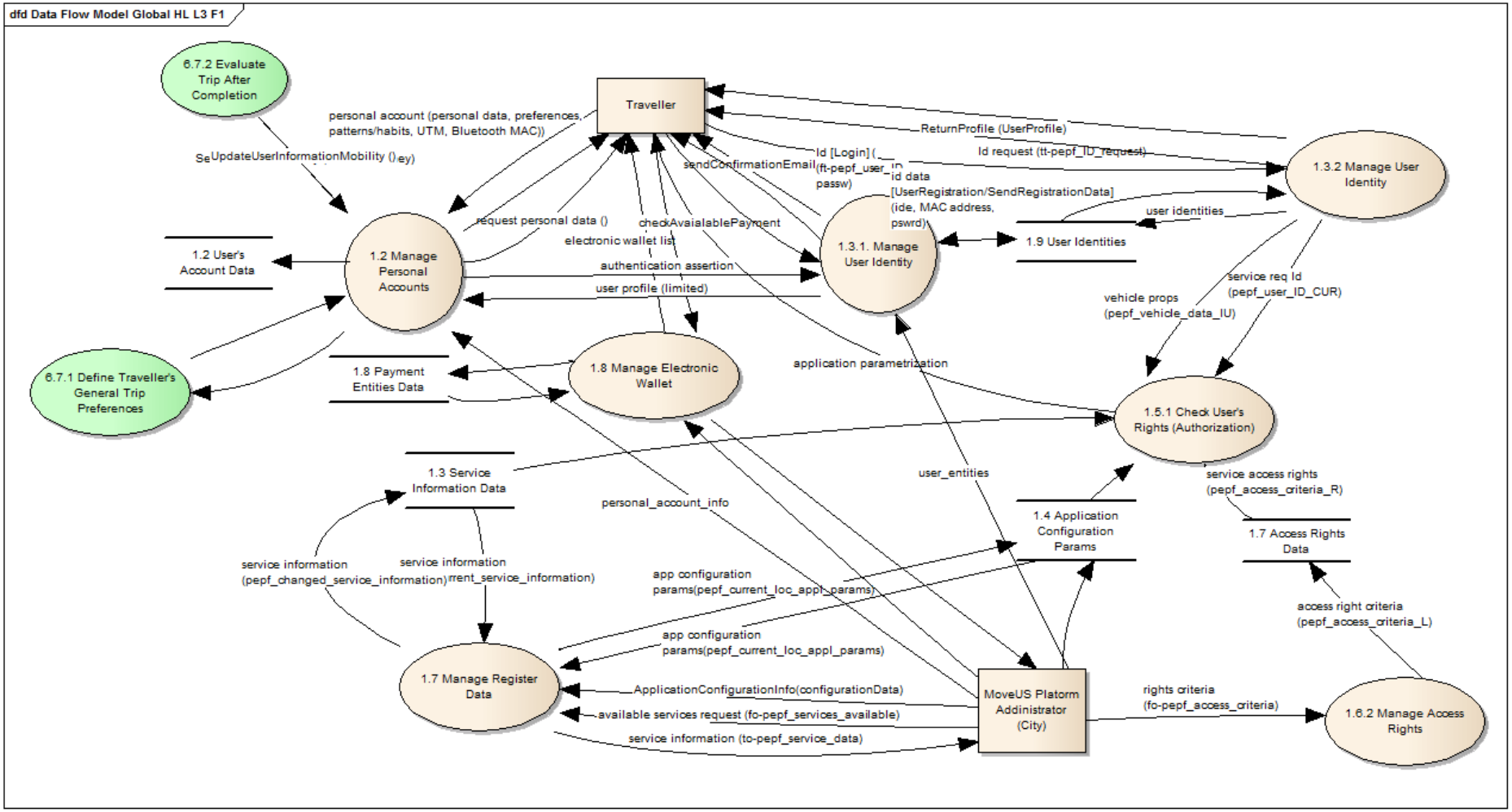
[7] MoveUs deliverable D3.1 - Data models, object models and ontology definition

[8] MoveUs deliverable D3.3 - MoveUs city services: specification and design

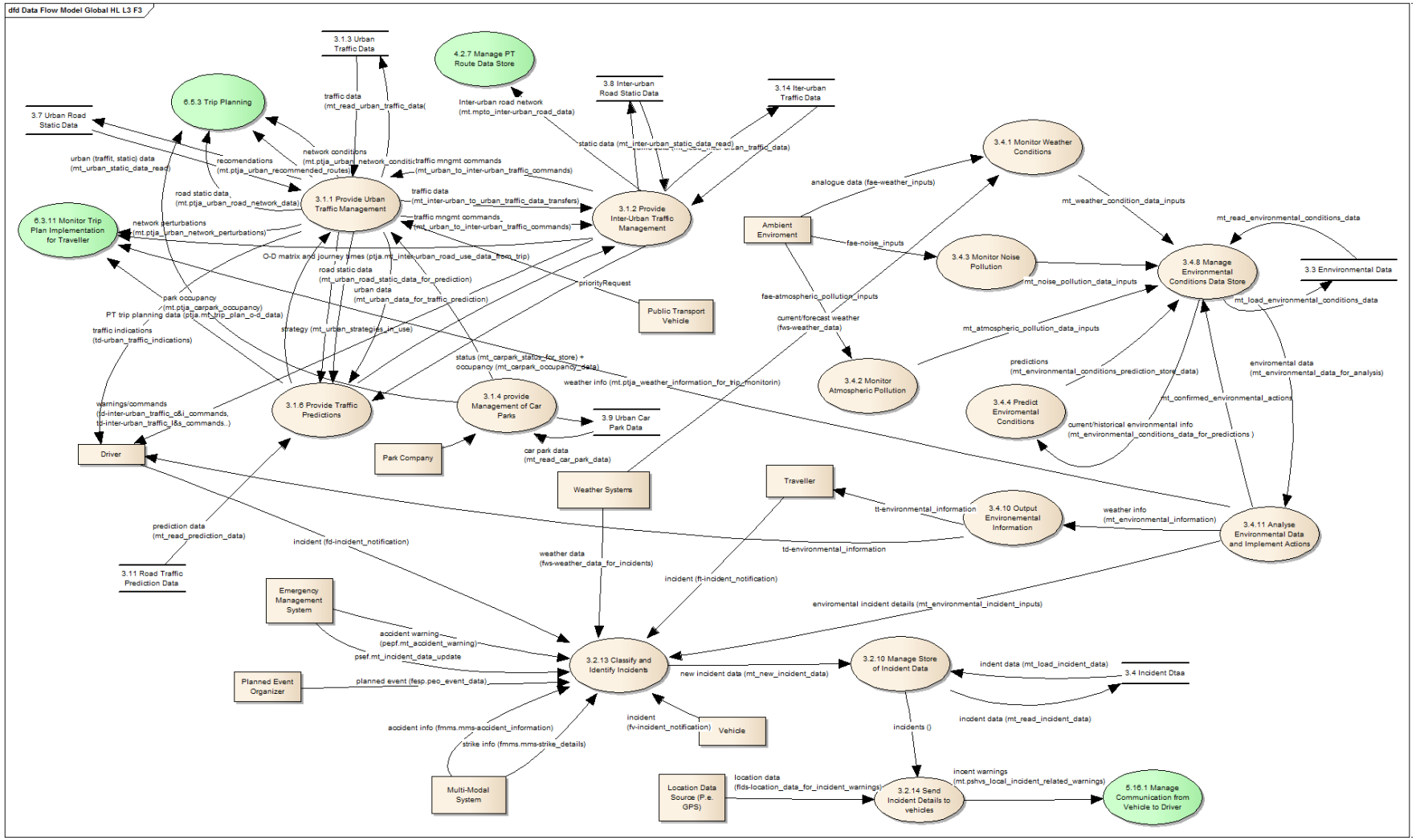
[9] The NIST Definition of Cloud Computing, National Institute of Standards and Technology of U.S. Department of Commerce, NIST Special Publication 800-145, Sept. 2011

[10] MoveUs deliverable D3.4 – Data Security and Privacy in the MoveUs system

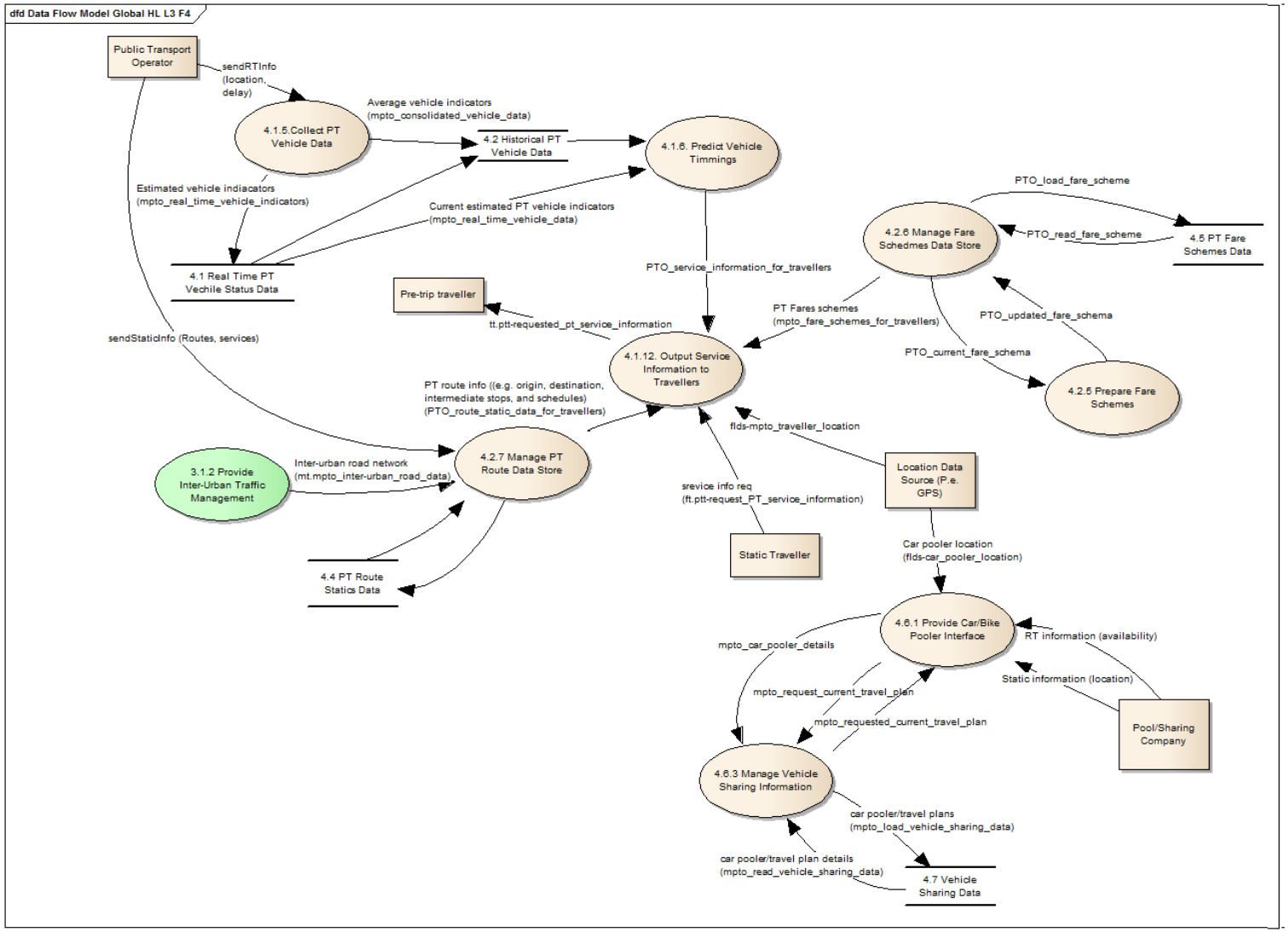
Annex 1: Data Flow Diagrams (DFDs)



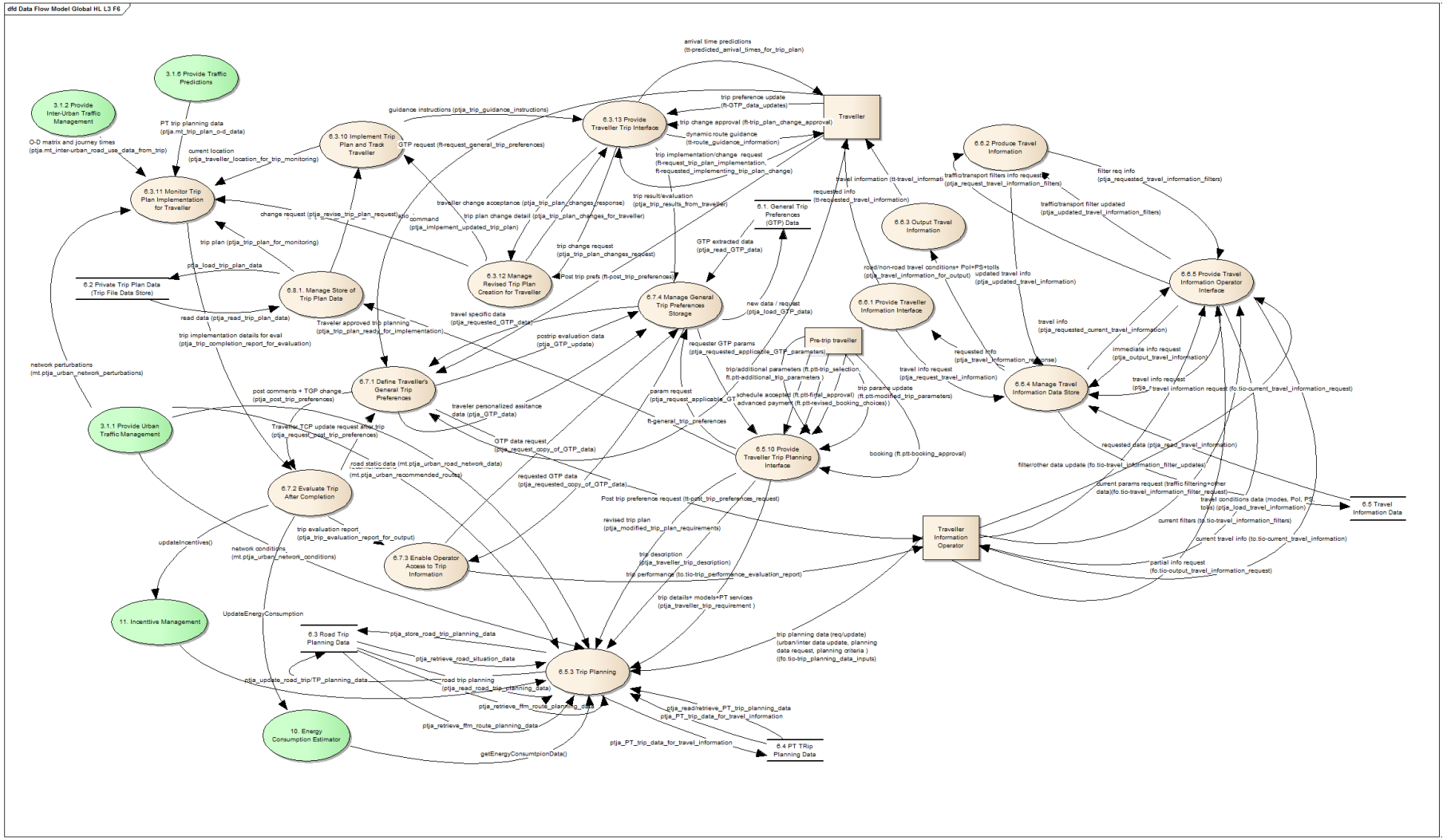
User Management and Application Tailoring DFD



Traffic Management DFD



PT Operation Management DFD



Traveler Journey Assistance DFD

Annex 2: MoveUs Platform Services

			I/O	Input	Output	Syntaxes	Origin
Journey Planning Services							
Road Static Information	Static Road Traffic Information		O	Key information is to be specified by entering the origin and destination (position or address). Other types of information which could be provided are single points or areas. Single point coordinates could be also determined by means of devices such as GPS receivers.	This service provides static information (like route, distance, specific route characteristics like road type, one ways, restrictions, road junctions etc.)	RequestRoadStaticInformation()	InTime / MoveUs
	Static Road Intersections Information		O	Key information is to be specified by the driver by entering the origin and destination (position or address). Other types of information which could be provided are single points or areas. Single point coordinates could be also determined by means of devices such as GPS receivers.	This service provides static information related with intersections. Concretely, the data structure is aligned with SAE J2735 structure, integrating road topology and involved equipment.	RequestIntersectionStaticInformation()	MoveUs
Road Dynamic Information	Dynamic Road Traffic Information		O	Key information is to be specified by the traveler by entering the following parameters: • UserId • Origin & Destination (position, address or stop) • Required Start time/Arrival times • Required stop-over ("via"-function) • In case of disturbance information possibly the type of disturbance • Time period of validity (current time, period in future for forecast)	This service provides dynamic information about current or future traffic conditions; the service can be provided as pre-trip and on-trip information service to plan a route or to select a destination depending on the current or future traffic conditions.	RequestRoadDynamicInformation()	InTime / MoveUs
	Dynamic Road Intersections Information		O	Key information is to be specified by the driver by entering the origin and destination (position or address). Other types of information which could be provided are single points or areas. Single point coordinates could be also determined by means of devices such as GPS receivers.	This service provides static and dynamic information related with intersections. Concretely, the data structure is aligned with SAE J2735 structure, integrating current intersection signal light phases, preemption or priority services requests and signal status.	RequestIntersectionDynamicInformation()	MoveUs

Parking Information	Static Parking Information		O	Provides static information like information about parking facilities (parking garage, parking lane, P&R) and routing to that facility and/or the node between different services. The traveler will be provided with static map and parking information which includes: • cost information, operating hours, timely limitations, restrictions (Gas driven vehicles) • Transfer facilities available at the stops (elevators, parking facilities, and access ways, special) (optional)	Key information is to be specified by the traveler by entering the following parameters: • Destination point (position, address) or area • Additional Parking requirements related for example to opening hours, restrictions etc.	RequestCarParkingStaticInfo()	InTime
	Dynamic Parking Information		O	Key information is to be specified by the driver by entering the following parameters: • Area of interest (Specific location, specific roads or routes, geographic or administrative areas) • Time period of validity (current time, period in future for forecast) • Costs/Fees • Special requirements (parking spaces for women only, for disabled people, etc.)	Provides dynamic information (static information like tariffs, opening times, etc; dynamic information about occupancy or vacancy) for parking neighborhood, single parking facility or all parking facilities near a specific address for specific validity period. The driver will be provided with: • Parking facilities within the specified area of interest • Occupation (current/forecast) • Costs/Fees • Special requirements (if specified)	RequestCarParkingDynamicInfo()	InTime
Bike Hiring Information	Static Bike Hiring Information		O	Provides static information like information about bike hiring facilities	Key information is to be specified by the traveller by entering the following parameters: • Destination point (position, address) or area.	RequestBikeHiringStaticInformation ()	InTime
	Dynamic Bike Hiring Information		O	Key information is to be specified by the driver by entering the following parameters: • Area of interest (Specific location, specific roads or routes, geographic or administrative areas) • Time period of validity (current time, period in future for forecast) • Costs/Fees	Provides dynamic information (static information like tariffs, opening times, etc; dynamic information about availability) . The traveler will be provided with: • Facilities within the specified area of interest • Occupation (current/forecast) • Costs/Fees	RequestBikeHiringDynamicInformation ()	MoveUs
Public Transport Information	Static Public Transport Information	Get	O	Key information is to be specified by the passenger by entering the following parameters: • Origin or Destination (Stop point or address near a stop point) • Desired Start time/Arrival times • Desired stop-over ("via"-function) • Desired mode/service/operator	This service provides static information (time table, stops positions, stop related time tables) about all PT modes (Bus, Tram, Metro, Train etc.). The passenger will be provided with: • Start time, arrival time, number of changes, where to change, waiting times • Modes • Stop Name, Direction, Line/Service/Operator, Schedule of bus/tram/metro/train/airplane • Infrastructure information about Stop Places, Stations etc. (e.g. special services for disabled people (like elevators, specific toilets etc.), location	Extended from InTime	InTime

					of entrances etc.) (optional)		
	Dynamic Public Transport Information	Get	O	Key information is to be specified by the passenger by entering the following parameters: • Origin or Destination (Stop point or address near a stop point) • Desired start time/arrival times • Desired mode/service • Time period of validity (current time, period in future for forecast)	Provides dynamic information about Public Transport for a specific stop point, service, line, time period or date. This includes: forecast of the next passages, qualitative conditions of transport service, waiting times, regularity of service, news, and transport services changed/cancelled. The passenger will be provided with: • Start time, arrival time, number of changes, alternative routes (compared), prices, • Name, Direction, Platform & Schedule of bus/tram/metro/train/airplane • Interchanges, interchange intervals, maps (e.g. walking from bus to metro) • Special services (elevators, bicycles, disabled, elderly, and children)	Extended from InTime	InTime
Walking Information	Walk		O	Key information is to be specified by the pedestrian by entering the following parameters: • Area of interest (map and position) • UserId	Provides routing service for pedestrians. The pedestrian will be provided with: • Routing, distances • Landmarks	Extended from InTime	InTime
	PoI		O	Key information is to be specified by the pedestrian by entering the following parameters: • Origin or Destination • UserId	Provides routing service for pedestrians. The pedestrian will be provided with: • Routing, distances • Landmarks • PoI	Extended from InTime	InTime
Dynamic Road Traffic	Dynamic Road Traffic Routing Information		O	Key information is to be specified as parameters: • Origin & Destination (position, address or Place such as POI) • Via Points • Trip options (e.g. Road types, roads to avoid etc.) • Area of interest (e.g. single point (e.g. for traffic cams) single road section, specific roads or routes, geographic or administrative areas) • Time period of validity (current time, period in future for forecast)	Provides dynamic information about current or future traffic conditions for specific connection from origin to destination for route planning or turn-by-turn-navigation with parameters: • Intended starting date and time or arrival date and time • Preferred or excluded road types • Fastest, shortest, most economic route The driver will be provided with: • Routing Information (Origin to Destination)	Extended from InTime	InTime

					• On-Trip information (e.g. Travel Time, Distance, Road fees etc.) • Re-routing in case of traffic incidents, traffic jams etc. • Fuel/Energy consumption, CO2 emission (optional) • Incentives		
Dynamic Public Transport	Dynamic Public Transport Journey Routing		O	Key information is to be specified by the passenger by entering the following parameters: • Origin & Destination (position, address or stop) • Preferred modes of transport • Maximum number of changes • Required special services (elevators, bicycles, disabled, elderly, children) • UserId	Provides dynamic information for specific connection from origin to destination (and via station) with parameters including e.g.: • Preferred or excluded public transport mode • Routing via intermediate stops • Bicycle transport • Maximum numbers of changes The passenger will be provided with routing information: • Start time, arrival time, number of changes, alternative routes (compared), prices, • Name, Direction, Platform & Schedule of bus/tram/metro/train/airplane • Interchanges, interchange intervals, maps (footpaths between different means of transport) • Special services (elevators, bicycles, disabled, elderly, and children) • Incentives	Extended from InTime	InTime
Dynamic Walking	Dynamic Walking Planning		O	Key information is to be specified by the pedestrian by entering the following parameters: • Origin& Destination (position, address or stop) • Desired start time/arrival time • Facilities required (elevator, ramps, subways, flyovers etc) • Obstacles to avoid (steps, stairs, multilane trunk roads)	Provides static and dynamic information (static information like route, distance, time, obstacles; dynamic information like changes in route characteristics (construction works, etc.). The pedestrian will be provided with: • Route, distance, walking time • Detour recommendations • Disturbance notifications	DynamicWalkingPlanning (criteria, num_selection, start, end)	InTime

	Dynamic Cycling Planning		O	Key information is to be specified by the cyclist by entering the following parameters: • Origin & Destination (position, address or stop) • Desired start time/arrival time • Preferred route characteristics • Average Speed • UserId	Provides dynamic information for specific connection from origin to destination with parameters including e.g.: • Preferred cycle path characteristics (main routes, separated cycle lane, max. incline etc.) • Routing via intermediate stops • Information on deviations/road works • Connection to PT, Bike&Ride, cycle parking The cyclist will be provided with: • Route, distance, travel time, route characteristics • Detour recommendations • Disturbance notifications • Connection to PT, Bike&Ride, cycle parking	DynamicCyclingPlanning (criteria, num_selection, start, end)	InTime
Multi Modal Journey Planning			O	Key information is to be specified by the driver or the scheduler by entering the following parameters: • UserId • Area of interest (e.g. single point (e.g. for traffic cams) single road section, specific roads or routes) • Time period of validity (current time, period in future for forecast)	Provides dynamic information for specialised traffic and transport information like waiting time at border crossings, ferry time tables or weather information for freight traffic. The driver or the scheduler will be provided with: • Speed, traffic volume, traffic density (concentration) • Fuel/energy consumption (i.e. calculation of CO2 consumption per route) • Restrictions (e.g. ban on vehicles etc.) • Loading Zones (location) • Fuel/Energy consumption, CO2 emission (optional) • Incentives	GetMultimodalPlanning (criteria, num_selection, start, end)	InTime
PoI	Dynamic POI (Point Of Interest) Information		O	Key information is to be specified by the driver / passenger / cyclist / pedestrian by entering the following parameters: • Point of interest (category, e.g. restaurant) in vicinity of [location] • Time period of validity (current time, period in future for forecast)	Provides POI information for an area or near an address with thematic filter. The driver / passenger / cyclist / pedestrian will be provided with: • POI information (location, opening times, fees, programme) • Journey Plan (dynamic multimodal trip info for the journey with comparison travel time/costs) including special services (shuttle bus, train, etc)	RequestPoIStaticInformation (PoI.Id / PoI category, location, time) // RequestPoIDynamicInformation (PoI,Id / PoI Category, location, time)	InTime/ MoveUs

Traffic Event Information		Get	O	Key information is to be specified by the driver / passenger / cyclist / pedestrian by entering the following parameters: • Event (name or location) • Starting point (position, address) • Time period of validity (current time, period in future for forecast)	Provides dynamic information like information about temporary parking facilities, public transport lines for temporary re-routing in the neighborhood of the event venue. Event traffic information contains dynamic traffic and public transport information provided in connection with event traffic management for big events. The driver / passenger / cyclist / pedestrian will be provided with: • Static Information (location, programme) • Journey Plan (dynamic multimodal trip info for trip to/trip from event with comparison travel time/costs) including special services (shuttle bus, train, etc) with event as source of service (POI, parking, PT etc.)	Extended from InTime	InTime
Weather Information,		Get	O	For general purpose and different transportation modes key information is to be specified by the driver / passenger / cyclist / pedestrian by entering the following parameters: • Area of interest (specific location, specific roads or routes, geographic or administrative areas) • Time period of validity (current time, period in future for forecast)	If a source is available, provides dynamic weather information for specific roads, road segments (links), route or administrative area. The current information shall be regularly updated in certain intervals (5, 15, 60 minutes). Optionally with filter for specific message type and specific validity period. The driver / passenger / cyclist / pedestrian will be provided with: • Weather current & forecast for a given time frame	Extended from InTime	InTime
Multi-modal Journey Planning		Journey Plan Request	O	Key information is to be specified by the driver / passenger / cyclist / walker by entering the following requests into his computer: •UserId, • Origin& Destination (position, address or stop) • Desired Start time/Arrival time • Mode: preferred means of transport • Mode: maximum number of changes • Mode: required special services (elevators, bicycles, disabled, elderly, children) • Mode: Costs/prices, Reservation • Mode: preferred route (e.g. avoid toll road)	It is a comparative dynamic multimodal journey planning service. Unlike the other services, which are referring always to only one mode and information type (static/dynamic) , this service provides all information at once, listing all information simultaneously and, thereby, giving the comparison of the modes to each other. The driver / passenger / cyclist / walker will be provided with: 1. Mode: route, distance, travel time 2. Mode: means of transport (Car, Bus, Tram, Metro, Train, Airplane, Bike, Walking) 3. Mode: number of changes 4. Mode: special services (elevators, bicycles, disabled, elderly, children) 5. Mode: Costs/prices, Reservation 6. Car: restrictions (e.g. ban on vehicles etc.)	Extended from InTime	InTime / MoveUs

					7. Car: fuel/energy consumption CO2 emission 8. Car: parking facilities (location, costs/prices) • Fuel/Energy consumption, CO2 emission (optional) • Incentives		
External Entities Services							
Car Pooling Options	Update		I	This data type support the definition and storage of the carpooler's profile and can be used eventually at applicative level to find and match the different trip offerings: o Indicators about habits of the carpooler (smoker, has animals etc.) o Temporal thresholds allowed for departure and arrival time o Spatial thresholds for departure and arrival position o Size of an ideal corridor around the journey where pick-ups are allowed o Preferences about other carpoolers	This service updates the user car pooling profile with the preferences and parameters necessary for the carpooling service (e.g. preferred role (driver, passenger), gender, smoker…)	SendDataOnMobility (UserId, Car Pooling Preferences): returnRegistrationStatus	MoveUs
	Request		O	This data type defines the trip definition: • Origin & Destination (position, address or stop) • Required Start time/Arrival times	This service provides offered car pooling travel plans, matching the displacement requirements and personal preferences.	RequestCarPoolingOptions (TripRequestData): [CarPoolingTrips]*	MoveUs
	Car Pooling Offer	Request	I	Shares a plan travel as candidate for car sharing and sends its description to the registered car pooling/sharing companies. The notification integrates the following information : • Calendar: set the days of the week and time when the offering is available • Role: driver, passenger or unspecified • Visibility: set if the offering is visible or not • Roundtrip: to specify whether it is a round trip • Validity: to specify until when the offering is valid	This service shares a private journey plan as candidate for car pooling service.	SendCarPoollingOffering (UserId, CarPoolingOptions): ACK/NACK	MoveUs
		Confirmation (Optional)	I	Car pooling company confirms the pooling service for a specific itinerary and sends an identifier.	MoveUs platform marks the travel (Trip Planner Plan Data) as shared and assigns the received identifier for service coordination.	SendCarPoolingOfferingConfirmation (CarPoolingOptions) : ACK/NACK	MoveUs
	Car Pooling Service Company Management	Register	I	Information of the Car Pooling organization.	This service registers the car pooling company as external service provider, resending car pooling request and notifications from final users in order to involve this transport mode as a chance for travel plans. Anyway, parts will coordinate externally through the Pooling Service Company	SendCarPoolingCompanyRegister (CarPolingCompany Data)	MoveUs

					System.		
	Unregister	Unregister	I	Information of the Car Pooling organization to be unregistered (in fact, only organization name or UserId is mandatory).	This service unregisters the car pooling company as external service provider and	SendCarPoolingCompanyUnRegister (CarPolingCompanyData)	MoveUs
Available payment systems	Available payment systems	Register	I	New Payment entities are create, setting its name and electronic payment details: server location network address (URL, etc.), component/service to be used for the communication.	This service registers the external entity as candidate for payment.	SendCarPoolingCompanyRegister (CarPolingCompanyData)	MoveUs
		Unregister	I	Information of the External payment organization to be unregistered (in fact, only organization name or UserId is mandatory).	This service unregisters the external entity as candidate for payment.	SendCarPoolingCompanyRegister (CarPolingCompanyData)	MoveUs
	Selection of available payment systems		O	As input, city in which user is seeking available payment services.	This service returns the information associated to registered payment systems. Later payment transactions themselves are not stored into MoveUs platform.	RequestAvailablePaymentSystems (City): [ExternalLocalServices]*	MoveUs
Feedback Management Services							
Feedback Management	Traffic Quality Feedback		I	Overall evaluation feedback is reported with a scale from 1 to 5 in the following concepts: quality of data, response time, quality of service, user interface, and reliability.	This service reports a quality measure of the traffic information.	SendQualityFeedback (UserId, segment, feedback correctness)	CoCities / MoveUs
	Journey Planning Quality Feedback		I	Overall evaluation feedback is reported with a scale from 1 to 5 in the following concepts: quality of data, response time, quality of service, user interface, and reliability.	This service reports a quality measure of the multi-modal journey planner.	SendQualityFeedback (UserId, segment, feedback correctness)	CoCities / MoveUs
	Traffic Feedback		I	Reporting of traffic incidence or level.	This service reports a traffic incidence or changed status, being updated on the road dynamic or incidence data.	SetTrafficFeedback (UserId, incidence, pos)	CoCities / MoveUs
	Request		O	User that reported the feedback information	This service returns the list of feedbacks reported by a selected user . Different information will be reposted based on feedback type.	RequestQualityFeedback (UserId): [Feedback]*	CoCities / MoveUs
	Update		I	User that reported the feedback information	This service returns the list of feedbacks reported by a selected user . Different information will be reposted based on feedback type.	UpdateQualityFeedback (UserId, feedback)	CoCities / MoveUs
Positioning Services							
Positioning Services	Resolve Location		O	As input, location of a road element	This service supports the location service: relating search strings, names, coordinates or locations in	ResolveMobilityLocation (UserLocation):	MoveUs

					the traffic network to each	(X,Y)	
	Resolve Address		O	As input, a full address string	This service supports the location service: relating search strings, names, coordinates or locations in the traffic network to each	ResolveAddresss(IUserLocation): (X,Y)	MoveUs
Personal Account Services							
Personal Account Service	User Management	User Registration	I	A user can be registered by providing his/her identity, Device Id, and password.	Registration of a user into MoveUs platform	UserRegistration (Id, Device Id, Passwd)	MoveUs
		Login	I	Each time a user accesses to MoveUs services logs introducing the login information (user, password).	User login to MoveUs platform. The system returns an authorization token.	Login (user_name, Passwd)	MoveUs
		Password Change	I	The user requests the change of password, providing the old and the new one.	The user change his/her password.	ChangeUserPassword (UserId, OldPass, NewPass)	MoveUs
		Password Recovery	I	A user can request the recovery of his/her account, depending of specific implementation; this request will be reported via eMail or specific message.	If forgotten, the user request an account recovery.	PasswordRecovery (UserId) (message) // eMail (with specific subject as "Password Recovery")	MoveUs
	Manage mobility personal data	Usual Trips	I/O	Including origin, destination, time and calendar (day of the year, month or week).	Two services for storing/requesting the information related to usual trips of a user.	SendUsualTrips (UserId, [UsualTrip]*) (M)(I) // RequestUsualTrips (UserId): [UsualTrip]*	MoveUs
		Mobility Preferences	I/O	Preferred transport mode (car, PT, bike, walk, sharing). This information is stored as pairs of type (transport, rating), where rating is the rating of preference.	Two services for storing/requesting the information related to user mobility preferences.	SendMobilityPreferences (UserId, MobilityPreferences) // RequestMobilityPreferences (UserId): MobilityPreferences	MoveUs
		Car pooling Preferences	I	Preferred role (driver, passenger), gender, smoker…) for pooling service.	Previously defined as "Car Poooling Services"	SendCarPoolingPreferences (UserId, CarPoolingProfile) //RequestCarPoolingPreferences (UserId): CarPoolingProfile	MoveUs

	Manage personal Data	Personal Data	I/O	PersonalData includes: name, surname gender, address, email, telephone 1/2, fax, accessibility constraints and POI preferences (for any of the predefined PoI (e.g. parkings, restaurants or merely locations).	Two services for storing/requesting information related to personal data of the user.	SendPersonalData (PersonalData): returnRegistrationStatus	MoveUs
		Vehicle Data	I/O	Vehicle Data. The information of the user registered vehicles (until 2): engine type, power, fuel consumption and other environmental-related data (to be defined according the WP4 algorithm for energy consumption/efficiency calculus).	Two services for storing/requesting information related to vehicle profile.	SendPersonalData (Vehicle Data): returnRegistrationStatus // RequestPersonalData ()+H39	MoveUs
Incentive Management Services							
Incentive Management	Rules Management	Request	O	User owner of the requested rules	As result this service returns the set of rules defined by the user.	RequestOwnedRules (UserId): [MV_I_Rule_Basic]*	MoveUs
		Remove	I	User owner and identifier of the requested rules	If the user is the owner of the rule or an administrator, the rule is removed.	RemoveRule (UserId, RuleId): ACK/NACK	MoveUs
		Update	I	User owner of the requested rules and description of the rule.	If the user is the owner of the rule or an administrator, the rule is updated according the new definition	UpdateRule(UserId, RuleDesc): ACK/NACK	MoveUs
	Currencies	Request	O	User owner of the requested currencies	As result this service returns the set of currencies defined by the user.	RequestCurrencies (UserId): [Currencies]* : [MV_IncentiveCurrency]*	MoveUs
		Remove	I	User owner and identifier of the requested currency	If the user is the owner of the currencies or an administrator, the currency is removed.	RemoveCurrency (UserId, CurrencyId (ID:int)): ACK/NACK	MoveUs
		Update	I	User owner of the requested rules and description of the currency.	If the user is the owner of the currency or an administrator, the currency is updated according the new definition	UpdateCurrency (UserId, CurrencyDesc) : ACK/NACK	MoveUs
		Create	I	User owner of the new currency and description	The currency is created according the new definition	CreateCurrency (UserId, CurrencyDesc (MV_IncentiveCurrency) : ACK/NACK	MoveUs
	Incentives	Request	O	User owner of the requested incentives	As result this service returns the set of incentives defined by the user.	RequestIncentiveList (UserId): [MV_Incentive]*	MoveUs

		Remove	I	User owner and identifier of the requested incentive.	If the user is the owner of the incentive or an administrator, the incentive is removed.	RemoveIncentive (UserId, IncentiveId (ID:int)): ACK/NACK	MoveUs
		Update	I	User owner of the requested incentive and description.	If the user is the owner of the incentive or an administrator, the incentive is updated according the new definition	UpdateIncentive (UserId, IncentiveDesc): ACK/NACK	MoveUs
		Create	I	User owner of the new incentive and description	The incentive is created according the new definition	CreateIncentive (UserId, Currency (Id: int), Rules : [Id: int]*) : ACK/NACK	MoveUs
		Access	I	User owner of the requested incentives	Access coupon and deals information	RequestCouponDealsInformation (UserId): Execute(MV_incentiveBalance where assignedTo is UserId)	MoveUs
	Payment Type	Request	O	User owner of the requested payment type.	As result this service returns the set of payment type defined by the user.	RemoveIncentivePaymentType (UserId, IncentivePaymentTypeId (ID:int))	MoveUs
		Remove	I	User owner and identifier of the requested payment type.	If the user is the owner of the payment type or an administrator, the payment type is removed.	RemoveIncentivePaymentType (UserId, IncentivePaymentTypeId (ID:int))	MoveUs
		Update	I	User owner of the requested payment type and description.	If the user is the owner of the payment type or an administrator, the payment type is updated according the new definition	UpdateIncentivePaymentType(UserId, IncentivePaymentDesc)	MoveUs
		Create	I	User owner of the new payment type and description	The payment type is created according the new definition	CreateIncentivePaymentType(UserId, IncentivePaymentDesc)	MoveUs
	Awards	Request	O	User owner of the requested award.	As result this service returns the set of awards defined by the user.	RequestAwardList (UserId): [MV_Awards]*	MoveUs
		Request List	O	No parameter is needed	This service returns the whole list of registered awards on the system.	RequestIncentiveList (UserId): [MV_Incentive]*	MoveUs
		Remove	I	User owner and identifier of the requested award	If the user is the owner of the payment type or an administrator, the awards is removed.	RemoveAwards (UserId, AwardId	MoveUs

						(ID:int))	
		Update	I	User owner of the requested award and description.	If the user is the owner of the payment type or an administrator, the award is updated according the new definition	UpdateAwards (UserId, IncentiveDesc)	MoveUs
		Create	I	User owner of the new award and description	The award is created according the new definition	CreateAwards (UserId, Award)	MoveUs
		Notify Acquisition	O	User owner of the award, identififer and acquirer.	Notify an award to the owner of an award, sending also its acquierer identifier.	NotifyRequestAward(UserId,AcquirerUserId, AwardsId): ACK/NACK	MoveUs
		Notify Issue	O	User owner of the award, identififer and acquirer.	Notify the issue of a voucher to the owner of an award, sending also its acquierer identifier.	NotifyVoucherIssue (UserId,AcquirerUserId, AwardsId): ACK/NACK	MoveUs
	Coupons	Request	O	User owner of the requested coupon.	As result this service returns the set ofawards defined by the user.	RequestCouponList (UserId, CouponId (ID:int)): [CouponDesc]*	MoveUs
		Request associated to an award	O	User owner of the requested coupon and award identifier.	As results returns the list of coupons	RequestCouponList (UserId, Award): [CouponDesc]*	MoveUs
		Remove	I	User owner and identifier of the requested coupon.	If the user is the owner of the payment type or an administrator, the payment type is removed.	RemoveCoupons (UserId,CouponId (ID:int))	MoveUs
		Update	I	User owner of the requested coupon and description.	If the user is the owner of the payment type or an administrator, the payment type is updated according the new definition	UpdateCoupons (UserId, CouponDesc)	MoveUs
		Create	I	User owner of the new coupon and description	The payment type is created according the new definition	CreateCoupons (UserId, CouponDesc)	MoveUs
		Activate	I	User owner of the coupon, coupon identifier and activation state	Activate a coupon setting its state to true/false.	SetCouponActivation (UserId, CouponId (ID:int), activated: True/false)	MoveUs
		Emit Voucher	I	User owner and identifier of the coupon associated to the new voucher.	Emit a voucher for the selected coupon.	EmitVoucher (UserId, CouponId)	MoveUs
		Notify Coupon Request	O	User owner of the coupon, identifier and acquirer.	Notify the request of a coupon to the owner of an award, sending also its acquirer identifier.	NotifyRequestCoupon(UserId,AcquirerUserId, CouponId): ACK/NACK	MoveUs

	Advertisement	Request	O	User owner of the requested advertisement.	As result this service returns the set advertisement defined by the user.	RequestAdvertisementItem (UserId): [Advertisement]*	MoveUs
		Remove	I	User ownerand identifier of the requested advertisement.	The platform checks that UserId is the owner of the advertisemet item, then remove it.	RemoveAdvertisementItem (UserId,AdvertisementId (title:string))	MoveUs
		Update	I	User owner of the requested advertisement and description.	If the user is the owner of the payment type or an administrator, the payment type is updated according the new definition	UpdateAdvertisementItem (UserId, AdvertisementDesc)	MoveUs
		Create	I	User owner of the new advertisement information and its description	The payment type is created according the new definition	CreateAdvertisementItem (UserId)	MoveUs
		Notify	O	User to be notified and advertisement information	Notify advertisement data to a user	NotifyAdvertisementData (UserId, AdvertisementInfo)	MoveUs
	Voucher	Retrieval	I	User owner of the voucher to be retrieved	Set voucher as retrieved	VoucherRetrieval (UserId)	MoveUs
		Request	O	User owner of the voucher, identifier and an indication true/false for email reporting	As result the service returns the voucher information	RequestAVoucher (UserId, VoucherId, eMail: boolean)	MoveUs
	Administration	Advertisement Condition	I	Receives the condition: interest of user and/or geographical criteria and a boolean value.	Sets the global advertisement condition to the current value	SetAdvertisementCondition ([Interest of user||Geographical criteria], TRUE/FALSE)	MoveUs
		User Advertisement Condition	I	Receives the user identifier and the selected advertisement flag as enable or disabled	Sets the user advertisement condition to the current value	SetUserAdvertisement (UserId, true/false)	MoveUs
		Update Coupons	I	User owner of the requested coupon and a description	Updates the coupon with the current description	UpdateCoupons (UserId, CouponDesc)	MoveUs
Registry Services							
Registry	Application Parameters	Get	O	User and current location (place)	The service returns the configuration parameters needed to tailor MoveUs Apps	GetApplicationConfiguration (UserId, Location)	MoveUs
		Get by coordinates	O	User and current location (coordinates)	The service returns the configuration parameters needed to tailor MoveUs Apps	GetApplicationConfiguration (UserId, X,Y)	MoveUs
	Services	Get	O	User and current location (coordinates/place)	The service returns the available services for a user in the current location	GetAvailableServices (UserId, location): MV_Service*	MoveUs