



**HAL**  
open science

## Intelligent Interactive Services

Ana-Maria Roxin, Christophe Dumez

► **To cite this version:**

| Ana-Maria Roxin, Christophe Dumez. Intelligent Interactive Services. 2009. hal-01072020

**HAL Id: hal-01072020**

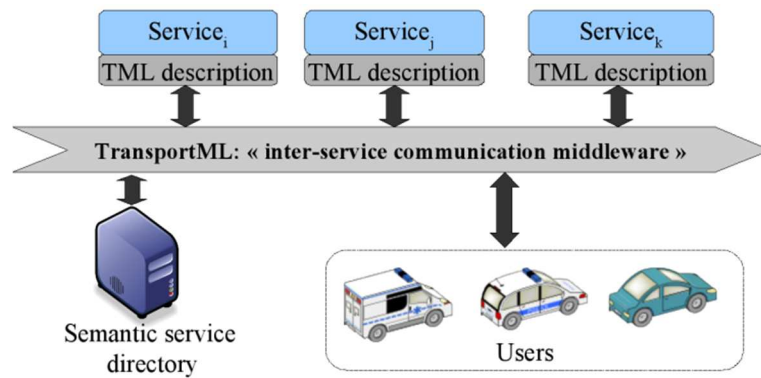
**<https://hal.science/hal-01072020>**

Submitted on 7 Oct 2014

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

## D5.1 Intelligent Interactive Services



07 May 2009

## Document information

Short title	ASSET–Road D5.1 Intelligent Interactive Services
Client:	
Contractor:	UBM University of Technology of Belfort-Montbéliard
Order no.:	
Edited by:	UBM University of Technology of Belfort-Montbéliard
Version:	0.4
Authors:	Ana Roxin, Christophe Dumez
Created on:	10/04/2009
Last saved:	07/05/2009 by Ana Roxin

## Revision Log

Version	Date	Action	Partner
0.1	10 April 2009	Original draft	UBM
0.2	22 April 2009	Modification / Editing	UBM
0.3	30 April 2009	Modification / Editing / Updated references	UBM
0.4	07 May 2009	Final corrections / Executive summary / Conclusion	UBM

## Contents

<b>Executive summary</b> .....	<b>6</b>
<b>Terminology</b> .....	<b>7</b>
<b>1 Introduction – Defining LBSs</b> .....	<b>8</b>
1.1 LBS usages .....	8
1.2 LBS Application Taxonomy .....	10
<b>2 TransportML requirements for LBS</b> .....	<b>12</b>
2.1 LBS Application Categories relevant to the ASSET Project .....	12
2.2 LBS Characteristics .....	14
2.3 Requirements for the TransportML middleware .....	14
<b>3 Related Work</b> .....	<b>16</b>
3.1 Service-Oriented Architecture (SOA) .....	16
3.2 Service Discovery .....	16
3.3 Ubiquitous and Pervasive Computing (UPC) .....	16
3.4 Location Information .....	17
3.5 Context Integration .....	17
3.6 Contract Model .....	18
3.7 Asymmetric Cryptography and Electronic Signature .....	18
<b>4 The TransportML Architecture</b> .....	<b>19</b>
<b>5 TransportML components</b> .....	<b>22</b>
5.1 The Service Discovery Framework .....	23
5.2 Secure Communications.....	25
5.2.1 The Security Manager	25
5.2.2 Digital Signature Creation and Validation Processes	25
5.2.3 Common Horizontal Facilities	25
<b>6 Data Communication</b> .....	<b>27</b>

6.1	Choice of the Transmission Medium.....	27
6.2	Communication between mobile nodes (vehicles) and the TransportML middleware .....	27
6.2.1	Infrastructure-based systems	28
6.2.2	Ad-Hoc/Multihop Systems	29
6.3	Communication between Services and TransportML middleware .....	32
<b>7</b>	<b>Test prototype .....</b>	<b>33</b>
7.1	Prototype operation .....	33
7.2	TransportML unified language .....	35
7.3	Involved Services .....	38
7.4	Test scenario Architecture .....	39
7.5	Performance Metrics.....	39
<b>8</b>	<b>Conclusion.....</b>	<b>40</b>
<b>9</b>	<b>References.....</b>	<b>41</b>

## **Executive summary**

This document defines Location-Based Services and their requirements with regard to the TransportML middleware. The architecture of the middleware is also discussed, as well as the TransportML language used for service interaction.

The introduction defines the concepts of Location-Based Services and discusses their usages. The related taxonomy is also presented.

The second part investigates LBS characteristics and the requirements they impose for the TransportML middleware.

Next are discussed the TransportML global architecture and the TransportML components.

The sixth part deals with data communication requirements, and presents two data communication technologies that have been chosen for testing purposes.

Finally, the developed TransportML test prototype is presented. Some performance metrics are also listed.

## Terminology

B2B Business-to-Business

B2C Business-to-Consumer

CALM Communication, Air-interface Long and Medium range

CSP Traditional Client to Server Paradigm

DTD Document Type Definition

EGNOS European Geostationary Navigation and Operation System

EU European Community

FCC Federal Communications Commission

GPRS General Packet Radio System

GPS Global Positioning Satellite

HTML Hyper-Text Markup Language

I2I Infrastructure-to-Infrastructure

LBS Location-Based Services

OWL Web Ontology Language

OWL-S Web Ontology Language for Services

PC Pervasive Computing

PDA Personal Digital Assistant

SCP Adaptive Services to Client Paradigm

SEP Spontaneous Service Emergence Paradigm

SMS Short Message Service

SOA Service-Oriented Architecture

SOAP Simple Object Access Protocol

SOUPA Standard Ontology for Ubiquitous and Pervasive Applications

TCP/IP Transport Control Protocol / Internet Protocol

UC Ubiquitous Computing

UML Unified Modelling Language

UPC Ubiquitous and Pervasive Computing

V2I Vehicle-to-Infrastructure

V2V Vehicle-to-Vehicle

XML eXtensible Markup Language



# 1 Introduction – Defining LBSs

The term “location-based services” (LBS) is a recent concept which denotes applications integrating geographic location (i.e., spatial coordinates) with the general notion of services. Examples of such applications include emergency services, car navigation systems, tourist tour planning, or “yellow maps” (combination of yellow pages and maps) information delivery.

With the development of mobile communication, these applications represent a novel challenge both conceptually and technically. Clearly, most such applications will be part of everyday life tomorrow, running on computers, personal digital assistants (PDAs), phones, and so on. Providing users with added value to mere location information is a complex task. Given the variety of possible applications, the basic requirements of LBS are numerous. Among them we can cite the existence of standards, efficient computing power, and friendly yet powerful human–computer interfaces.

A LBS is a service that makes use of the location of the requester in order to provide more relevant information. A LBS can also be assigned to a given area, meaning that it will only reply to mobiles located in its coverage zone.

## 1.1 LBS usages

Approximately 15% of current operator income in Western Europe and 20% in Asia is already based on data services. User location is an important dimension in this new data-service world: not only does it allow companies to completely conceive new service concepts (i.e., tracking applications), but it also has the potential to make many messaging and mobile Internet services more relevant to customers as information is adjusted to context (i.e., weather information adjusted to the region one is in). In addition, location information can considerably improve service usability.

As a result of these multidimensional benefits of location information, operators are coming to consider it as their “*third asset*” besides voice and data transmission. Important investments are being made to extract, use, and market it.

Location services are mainly used in three areas: military and government industries, emergency services, and the commercial sector. Besides the military use of location data, emergency services have turned out to be an important application field. Every day, 170,000 emergency calls are made in the United States. Of those, one-third originate from mobile phones, and, in most cases, people do not know where they are precisely in order to guide support to the correct location [7]. As a result, the U.S. FCC set an October 2001 deadline for commercial wireless carriers to provide the caller’s location information in a 911 emergency call. This means that when placing an emergency call from a mobile phone, a caller’s phone position is automatically transmitted to the closest emergency station. Consequently, people in such situations do not have to explain where they are, but are located in seconds. Ultimately, few carriers were able to meet the original deadline so the FCC relaxed the date for wireless E911 services. It is expected that it takes several years before the system reaches full coverage with high precision.

In Europe, the EU has followed a similar path. Statistics reveal that 50% to 70% of the 80 million “real” EU-wide emergency calls each year originate from mobile phones [2]. Some industry sources even argue that approximately 5000 lives could be saved each year in the region with automatic positioning of emergency calls. As a result, the EU Commission has passed Article 26 of the “*Directive of universal service and users’ rights relating to electronic communications networks and services*”<sup>1</sup>. This article asks member states to develop national regulations for mobile operators enforcing the automatic positioning of emergency calls: “*Member states shall ensure that undertakings which operator public telephone networks make a caller location information available to authorities handling emergencies, to the extent technically feasible, for all calls to the single European emergency call number 112*”.

“*Technical feasibility*” in this context means that unlike in the United States, European regulators do not enforce the highest accuracy levels such as GPS for locating emergency cases. Although GPS allows a cell phone to be located accurately, European operators have the right to start out with the accuracy levels their mobile networks can provide right now. Because more than 80% of European operators have implemented so-called Cell-ID technology [1] for mobile positioning, only very low accuracy levels can be offered for now in emergency situations: 100 meters potentially in urban areas, but only up to 3-kilometer accuracy in rural areas. A debate has started whether the latter is enough accuracy in the mid-term and ethically defensible by operators in case of life losses.

The accuracy debate leads to the third area of location use and probably the most ubiquitous one in the future: the commercial use of positioning information. For some time, marketers have been unsure whether lower levels of accuracy as they are obtained from Cell-ID would be sufficient to launch compelling consumer and business services. Yet, early service examples show that the accuracy level required depends very much on the service. Even with Cell-ID, location information can successfully be integrated by operators into many existing and new applications that enhance current value propositions and usability.

At a high level, the company Ericsson has developed a scheme of what accuracy levels it considers to be necessary for different types of applications. Table 1 gives an overview of this scheme.

Application	Accuracy
Directions	High
Traffic information	Low
Car navigation	Medium to High
Fleet management	Low
Car tracking	Medium to High
Asset tracking	High
Emergency	High

---

<sup>1</sup> 2002/22/EC of 7th March 2002

<b>Sensitive goods transportation</b>	High
---------------------------------------	------

Table 1. Overview of LBS applications and level of accuracy required<sup>2</sup>.

## 1.2 LBS Application Taxonomy

Analysts and researchers have taken several approaches to classify LBS applications. A major distinction of services is whether they are person-oriented or device-oriented.

- **Person-oriented LBS** comprises all of those applications where a service is user-based. Thus, the focus of application use is to position a person or to use the position of a person to enhance a service. Usually, the person located can control the service (e.g., friend finder application).
- **Device-oriented LBS** applications are external to the user. Thus, they may also focus on the position of a person, but they do not need to. Instead of only a person, an object (e.g., a car) or a group of people (e.g., a fleet) could also be located. In device-oriented applications, the person or object located is usually not controlling the service (e.g., car tracking for theft recovery).

In addition to this first classification of services, two types of application design are distinguished: push and pull services [3].

- **Push services** imply that the user receives information as a result of his or her whereabouts without having to actively request it. The information may be sent to the user with prior consent (e.g., a subscription-based terror attack alert system) or without prior consent (e.g., an advertising welcome message sent to the user upon entering a new town).
- **Pull services** in contrast, mean that a user actively uses an application and, in this context, “pulls” information from the network. This information may be location-enhanced (e.g., where to find the nearest cinema).

Table 2 gives an overview of the LBS service dimensions with some application examples.

---

<sup>2</sup> Source: Bellocci, V., Genovese, S., Inuaggiato, D., and Tucci, M. (2002, July 18). “Mobile Location-Aware Services: 2002 Market Perspective”, Ericsson, Division Service Architecture and Interactive Solutions.

	Push Services	Pull Services
<b>Person-oriented Information</b>	The user gets an alert that a terror alarm has been issued by the city he/she is in.	The user looks for the nearest hospital in his/her area and wants navigation instructions to get there.
<b>Device-oriented Tracking</b>	An alert is send to the user from an asset-tracking application that one of his/her shipments has just deviated from its foreseen route.	The user requests information on where a truck fleet is currently located.

**Table 2. Categories and examples of LBS applications.**

Most of the early location services in Europe have been pull services, especially information services. Push services have not come to flourish yet. Unproven economics and privacy concerns are the main reasons for this situation.

## 2 TransportML requirements for LBS

Generally speaking, *an information service is a network-accessible and computer-based system to collect, process, filter, transmit, and disseminate data that represents information useful for a specific purpose or individual.* Along the same lines, *an LBS refers to the additional integration of position location information as part of the data processed by the information service.* Thus an LBS provides and delivers information to its users in a highly selective manner, by taking users' past, present, or future location and other context information into account.

Often, location services and location application services are distinguished. *A location service provides specific geographic location information about mobile terminals, such as cell phones, PDAs, or with sensors tagged on moving objects. A location application service refers to the information service that exploits this location information about a mobile terminal to offer highly customized information content to the mobile user or to third parties (i.e., other mobile terminals or static users and applications).* Many early location application services were based on the user providing the necessary position information voluntarily by submitting a street name or a zip code to the application.

This part of the document focuses on location application services and the middleware technology required for supporting their operation. No distinction is made between the two categories, and we uniquely refer to this kind of application as location-based services or location-based applications, always assuming that the required location information exploited by the ulterior information service is made available somehow. To back up this assumption, state-of-the-art location position identification technology is reviewed in [27] by Roxin et al.

We define middleware (a.k.a. middleware platform or middleware system) as *a set of services that facilitate the development and deployment of distributed applications in heterogeneous environments.* Middleware consists of a set of services exposing interfaces, a programming model, and an interaction model to the application developer. For the context of LBS, this refers to the services, abstractions, and models that implement mobile user coordination, information correlation, and information dissemination.

The objective of this part is to identify LBS application categories relevant to the ASSET project and identify requirements they impose on our TransportML middleware platform.

### 2.1 LBS Application Categories relevant to the ASSET Project

Table 3 presents a classification of LBS that are relevant to the ASSET project.

Service category	Example Application	Characteristics
<b>Tracking services</b>	<ul style="list-style-type: none"> <li>- Goods, vehicle and fleet</li> <li>- Security of entities (cars)</li> <li>- Maintenance and assistance</li> </ul>	Tracking requests are often initiated by a remote monitoring entity.

	- Workforce dispatching		Push-based and pull-based.
<b>Selective information dissemination</b>	Targeted dissemination information)	content (traffic	Proactive, event-based and condition-triggered. A priori stateless.
<b>Emergency support services</b>	- Emergency - Ambulance, fire, police dispatching - Roadside assistance	112 police	Mobile-user-initiated Pull-based A priori stateless.

**Table 3. LBS application categories relevant to the ASSET project.**

To better understand the requirements imposed by different applications on the underlying TransportML middleware model, a finer-grained classification of application is helpful. This classification is as follows:

- **Tracking services.**  
 These services track the geographic whereabouts of, with mobile terminals equipped, entities (e.g., users, trucks, and packages) and support requests to establish the location of these entities, their progress and state change along a route, or perspective future location. Applications include fleet tracking, taxi monitoring and dispatching, workforce management, mobile supply-chain management, child support and security, tracking of elderly and sick persons, and goods and package tracking. This latter example is often supported by tracking the object as it passes through a statically fixed control point (i.e., a bar code reader). Active badge systems fall in this category as well [4].  
 Characteristic for these kinds of applications is the potentially very large number of entities that must be tracked simultaneously, the need to maintain state between different tracking points, and the inverse nature of the tracking request initiator (i.e., a remote monitoring entity is tracking the objects). Applications are pull- and push-based in nature.
- **(Selective) information dissemination services.**  
 This category refers to services that disseminate content to mobile users correlated with the subscriber’s location, context, and profile. A prime example is (selective) advertisement dissemination. Disseminated content may include e-coupons or simply advertisements. At one extreme, all users entering the range of a particular dissemination source could receive notifications. At the other extreme, highly selective correlations between a user’s interests and the advertisement content could selectively target individual users.  
 Characteristic for this category is the push-based character of the application. That is, the dissemination is initiated by the supporting middleware technology, with little or no user intervention. The support of selective content correlation requires a user profile and requires user identification information to be available for each user location quote.
- **Emergency support services.**  
 These services have driven the development and deployment of location positioning technology in North America and Europe. This information serves police, fire fighters, ambulances, and automotive support crews.

These services rely entirely on the available position location technology and do not, by themselves, impose any specific requirements on the middleware technology.

## 2.2 LBS Characteristics

From the above presentation of applications, the following LBS characteristics can be extracted. These characteristics are independent from each other and one application can implement more than one characteristic. These characteristics must be taken into account when implementing the middleware technology. Figure 1 gives a sum up of the criteria that determine LBS characteristics [29].

## 2.3 Requirements for the TransportML middleware

TransportML location-based middleware platform must do the following [29]:

- Manage the mobility inherent to all LBS applications by supporting disconnected operations and supporting mobility-awareness in the middleware.
- Manage changes in the underlying network topology that may occur in very dynamic settings, such as ad-hoc location-based services.
- Manage a potentially very large number of information providers (in some cases, comparable with the numbers of information consumers).
- Propagate notifications for thousands of information consumers simultaneously, which results in managing large amounts of content sent to the system for filtering, matching, and correlating.
- Manage high volatility of users' interests (e.g., profile updates, insertion, and deletion).
- Process diverse content formats, ranging from topic-tagged blobs and collections of attribute-value pairs to HTML and XML marked-up data, as well as easily support evolving and future data formats.
- Support high availability despite node failures (e.g., guarantee notification delivery).
- Perform security functions, such as subscriber and publisher authentication, secure content distribution (e.g., not all subscribers may be allowed to receive all publications that match their subscriptions).
- Support privacy consideration, allowing subscribers to opt for the propagation of their location information to selected applications only.
- Support high rates of information input (e.g., news, location information per user).

#### Criterion 1 – Information delivery policy

- Push-based applications rely on the traditional publish/subscribe paradigm, where information is pushed to the user, based on a given event occurrence or a given condition trigger [6].
- Pull-based applications imply that the user polls the server in search for information updates. It is the user who must request information from the server.

#### Criterion 2 – User profile gathering

- Direct mode – the user's profile is obtained directly from the user.
- Indirect mode – the user's profile is obtained from third parties or by analyzing the user's interaction pattern.

#### Criterion 3 – Interaction scenarios

- Both the user and the service provider are mobile – this applies in the case of mobile ad-hoc location-based applications, notably for friend finder applications.
- Only the user is mobile – notably in vehicle tracking applications and targeted advertising.
- Only the service provider is mobile – for example for an automatic airport check-in.
- Both the user and the service provider are stationary – in this case no dynamic management is needed for location information.

#### Criterion 4 – Statefulness of interaction

- Stateful interactions characterize applications in which the LBS maintains state across multiple service requests [6].
- Stateless interactions characterize applications where each request is processed independently from other requests [6].

#### Criterion 5 – Sources of information

- Static information sources mostly concern POI databases or traditional GIS.
- Dynamic information sources vary according to user's position, the time of day. One may cite traffic information or weather forecasts.

#### Criterion 6 – Sources and accuracy of location information

- The site performing measurements and position calculus,
  - Network-based positioning – the network performs position calculus.
  - Terminal-based positioning – the terminal performs position calculus.
  - Terminal-assisted positioning – the terminal only performs measurements and then forwards the results to the network, which performs position calculus.
- The type of network on which they are implemented and operated.
  - The satellite infrastructures cover large geographical areas and are achieved by stand-alone infrastructures of several satellites. Satellite positioning is always terminal-based.
  - The cellular infrastructures refer to cellular networks (GSM, GPRS, etc.). Cellular networks operators use several methods to obtain the position of a mobile device.
  - The indoor infrastructures are based on radio, infrared or ultrasound systems, deployed in indoor environments and having limited communication range.

**Figure 1. LBS Applications' Characteristics**



## 3 Related Work

### 3.1 Service-Oriented Architecture (SOA)

For the ASSET Road project, we proposed a service-oriented architecture, in which business services can interact with one another via the middleware TransportML. Since it is essential to ensure information and process flows between and across heterogeneous business environments, an application's business logic has to be presented as a business service via a platform-independent XML interface. SOA is a branch of distributed computing that helps organizations share logic and data among multiple applications and usage modes. The basic architecture of SOA involves three fundamental elements: service providers, consumers and a service discovery and composition system [13].

### 3.2 Service Discovery

In order to develop a platform that can easily adapt to user requirements and services' availability, an appropriate service discovery system must be implemented. More precisely, the service discovery framework must allow new services to be added, updated or removed dynamically. It must also allow users to discover services dynamically and according to their current context. In other words, the TransportML service discovery component should exhibit self-adapting and self-organizing capabilities.

This discovery service may benefit from spatial registries [31] which combine scalability and local perspectives to register and discover LBSs.

### 3.3 Ubiquitous and Pervasive Computing (UPC)

Service-oriented architectures are often designed according to the client-server paradigm. In a server-centric model version, there is a risk of single point of failure. In a peer-to-peer model version, the infrastructure and communication networks and all information about services are supposed stable and always available. Still, in a dynamic environment, the overhead to continuously update information about services in registries could be huge. Moreover, traditional client-server paradigms don't allow taking full advantage of ubiquitous and pervasive computing.

According to Gaber's classification, interaction paradigms can be classified into three categories: the traditional client to server paradigm (CSP) and two alternative paradigms, the Adaptive Services to Client Paradigm (SCP) and the Spontaneous Service Emergence Paradigm (SEP). SCP and SEP suits more ubiquitous and pervasive environments and requirements respectively [20, 12]. Recall that the main objective of Ubiquitous Computing (UC) is to provide users information and service accesses anytime and irrespective to their location, while in Pervasive Computing (PC), the main objective is to provide spontaneous emergent services created on the fly by mobiles that interact by ad-hoc connections [20, 12]. In other words, SEP with PC may be viewed as a collective use and cooperation of mobile computers and wireless devices that exist and are available in the users' physical environment.

It should be noted that in the traditional client to server paradigm (CSP), the client requests a service by first knowing its existence and has to provide its exact location. In the SCP alternative paradigm, the service is directly addressed to the client, via an intelligent middleware. The paradigm SEP allows a service to spontaneously emerge, without any prior planning and according to ad-hoc collective use and cooperation of mobile devices in the surrounding environment [20, 12]. SCP and SEP integrate self-adaptation and self-organization possibilities that enable the development of ubiquitous and pervasive applications. TransportML will therefore use these two paradigms for delivering intelligent services to users.

### 3.4 Location Information

The process of implementing LBSs mainly depends on the reliability of location information. Hence, the positioning technology used must ensure the relevance, the continuity and the availability of the information. Indeed, location-based applications can be indoor, outdoor, or supported both within buildings and outdoors. Depending on the positioning technology used, location information can be represented in different ways and in different granularities [28], varying from centimeters to meters or greater. For example, the accuracy delivered by GPS is about 20m. When using an augmentation system such as EGNOS, the accuracy drops down to about 5m.

Location information representation models are also an important issue to index entities in the environment. In [28], Roxin et al. have proposed three types of spatial models: a flat graph model, a hierarchical graph model and a real-time graph model. Each model provides a given level of granularity of information details to location-based applications.

The variety of positioning technologies and short-range networking technologies continue to improve in their reliability and accuracy. An extensive review of location systems for ubiquitous computing is given in [27].

### 3.5 Context Integration

It is worth noting that context representation and manipulation are important issues for the ASSET project. Location information is one fundamental type of context information. But in order to deliver intelligent LBSs, other types of context information must be taken into account.

Indeed, an LBS is said to be context-aware since it behaves in a way that proves knowledge of the user's current context. The same piece of information will be presented differently according to one or several parameters that reflect the user's context. The ensemble of these parameters is termed context information. Context information can be divided into the following two categories [24]:

- The primary context information consists in spatial information (identity, time and location). This kind of context information may be used in order to index entities.
- The secondary context information consists of additional aspects of a given entity, generally divided according the context they refer to (personal, technical, spatial, social or physical context).

In the context of TransportML, context will be modelled through an OWL-S ontology, that will define a service profile (or service type) hierarchy. The main advantage of this type of modelling is to allow a seamless and easy to use service discovery module (see Figure 7).

### **3.6 Contract Model**

To address services' access issues, it is necessary to guaranty the user access to utilize the allowed and available services. More precisely, an access control must be granted to services in order to regulate their usage. The main problem resides in managing access policies to disparate services that are not under the control of a single system designer/administrator. The computational contract model presented in [26], which defines an exchange process between clients and service providers, is used for the TransportML middleware. It should be also noted that security is one of the fundamental issue for the ASSET Road project.

### **3.7 Asymmetric Cryptography and Electronic Signature**

TransportML messages must intrinsically include security information used to verify their integrity, authenticity and perform access rights checks. Historically, modern asymmetric cryptography is based on {public key, private key} key pairs. Electronic signatures generation and verification is basically digital signatures processes along with digital certificates integration [16]. The aim of electronic signature is to verify signed contents integrity and authenticity rather than searching for information confidentiality.

Since XML has become a standard to formalize networked communications, it has enlarged its usage to cryptography and digital signature encoding formats [18] and tends to replace the ASN.1 BER and DER encodings, although its suffers from many security weaknesses (e.g., infinite loops in DTDs, multiple DTD conformant XML encodings for same information, etc.) [21].

Considering that our TransportML middleware relies on XML-based information exchanges, XML digital signature and encryption schemes [18][30] seem to be the most adapted solutions to acheive security goals.

## 4 The TransportML Architecture

TransportML is the result of extensive research work carried out on ASSET French partner (UBM) part. TransportML can be considered as a standard interface between location-based services which expose their data according to SOA principles. More precisely, TransportML allows Web services to interact automatically, meaning that there is no need for a predefined collaboration scenario. Indeed, the scenario is determined at execution time, by using the TransportML-compliant semantic description of the services (Figure 2). As a consequence, TransportML is highly dynamic and extensible, allowing new services to join the community and benefit from its communication bus, in an easy manner.

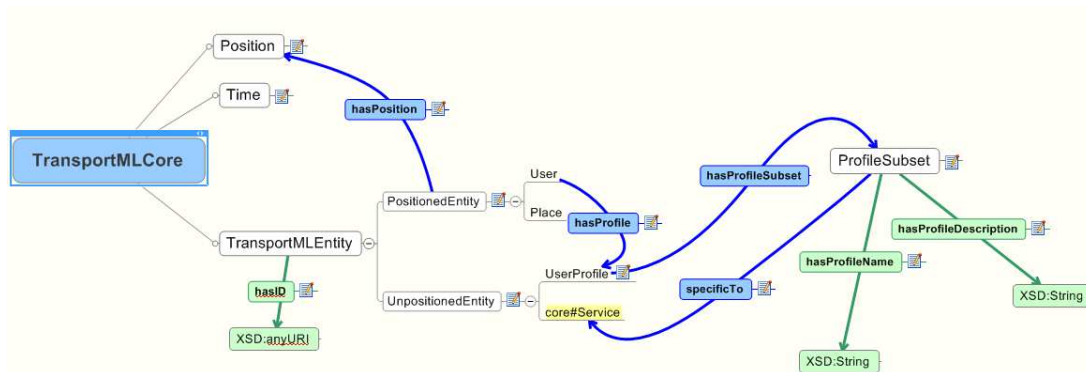


Figure 2. TransportML core ontology.

The TransportML architecture extends services using a TransportML-compliant semantic description. This standard description is provided by the service when registering itself with the semantic service directory. Such directory is used for advanced service discovery, which is possible thanks to semantic enhancements (through the extended OWL-S service ontology – see Figure 7). Whenever TransportML receives a client request (e.g. “I am looking for an itinerary from point A to point B that avoid roadwork”), the request is expressed semantically in order to determine which services are usable and the order in which they should be invoked. Once a suitable collaboration scenario is established, TransportML contacts the services which were selected and returns the result to the client.

The TransportML architecture is designed for a pervasive environment, so every component of this architecture must be uniquely identified. The overall architecture of the TransportML middleware is presented in Figure 3. Such architecture allows providing value-added services, resulting from the collaboration between existing services maintained by different entities. In this context, a service is a standard way for companies, associations or others organization to expose their internal knowledge.

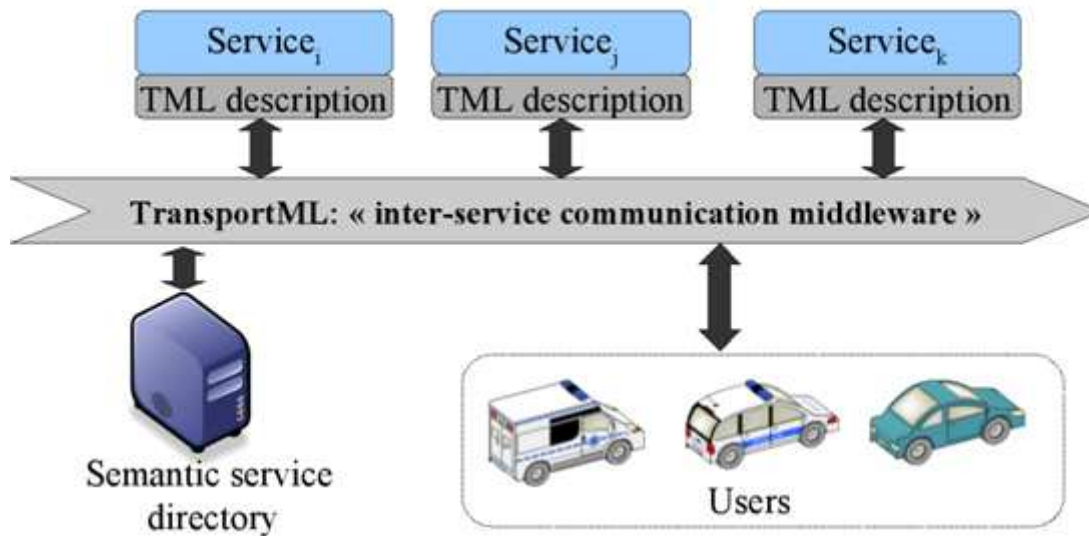


Figure 3. The TransportML middleware – global view.

The platform architecture depicted by Figure 4 comprises two layers: a vertical layer which contains different business services to be coordinated (snow clearance, emergency services, firemen services, etc.), and a horizontal layer composed of the TransportML middleware. The main objective of this middleware is to allow coordination, cooperation and synchronization between all business services of the vertical layer by exchanging information and enabling resource sharing. These operations are ensured by a horizontal communication layer composed of communication networks and positioning systems.

The implementation of the platform handles communications security, actors' authentication, access rights management, time-stamping and traceability and manages different interactions between users.

TransportML-compliant services are known as service providers. They make information available to clients (service requesters), mentioning the required credentials to access the service. This publication does not contain the whole information but a small description so that requesters are able to determine the information relevancy.

The TransportML architecture implies for services to be semantically described, and then advertised by service providers. During the process of service advertisement/publishing, the service's contextual information is published. A discovery protocol (detailed below) allows determining which service suits best a user's request.

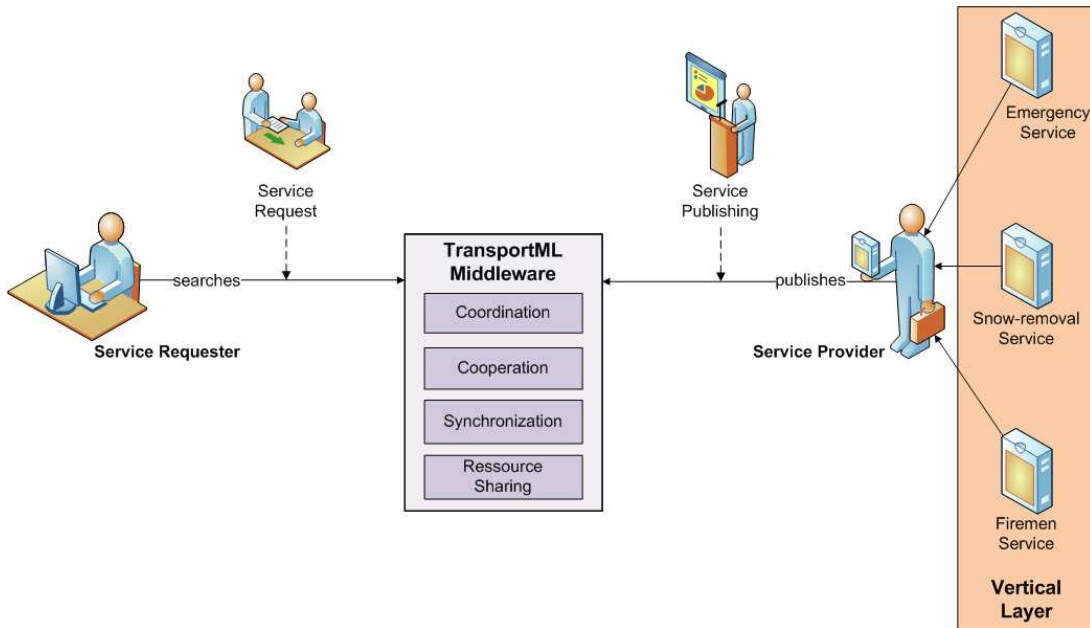


Figure 4. The reference architecture for the TransportML middleware

The UML use-case model illustrated by Figure 5 describes the main information providers and requesters actions.

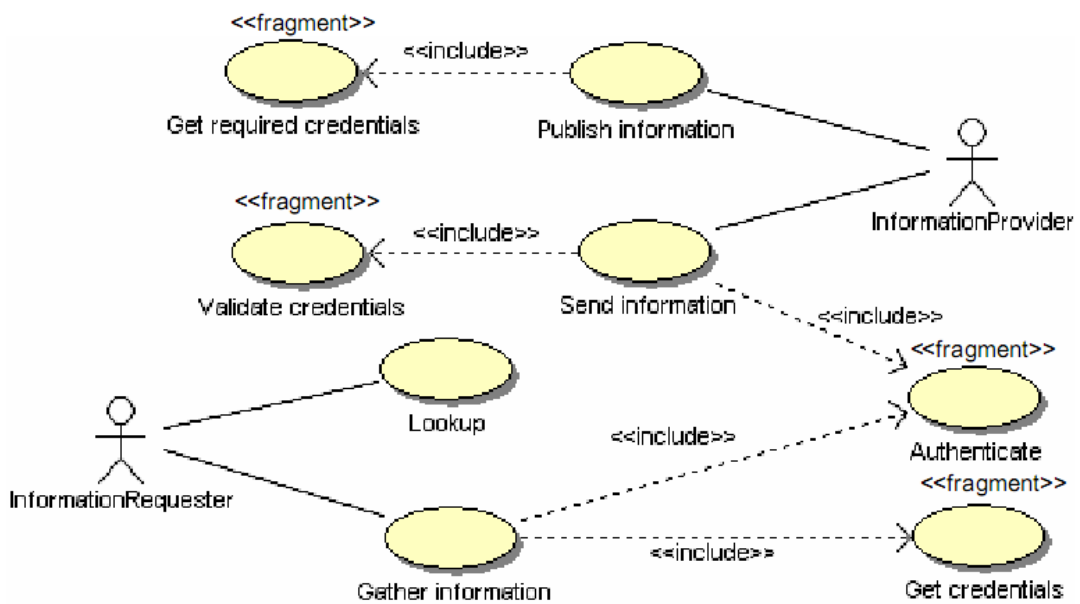


Figure 5. TransportML main use cases.

Providers publish information summaries made visible to requesters (using a lookup process). Providers can then send information when requesters ask for them, providing the required credentials. This operation can be compared with RSS feeds syndication.

## 5 TransportML components

As depicted by Figure 6, information providers publish information about their services to be consulted by user's mobile agents. This information is associated with required credentials known as publication rules represented by a set of expected credentials. This access rules information is given to requesters so that they can check that they own the appropriate credentials to satisfy the information access requirements. When asking for a given service available on an information provider, a requester provides the set of required credentials and creates a published information request. This request is digitally signed by the requester's security manager before sending it to the information provider to get the published information. The information provider verifies the requester's digital signature and checks the credentials validity and matching with the expected credentials by means of its own security manager. The whole information is then digitally signed and sent back to the requester as far as the latter's credentials are sufficient.

Each information and information summary holds an obsolescence delay which indicates that a new request must be performed to get up-to-date information after this maximum amount of time.

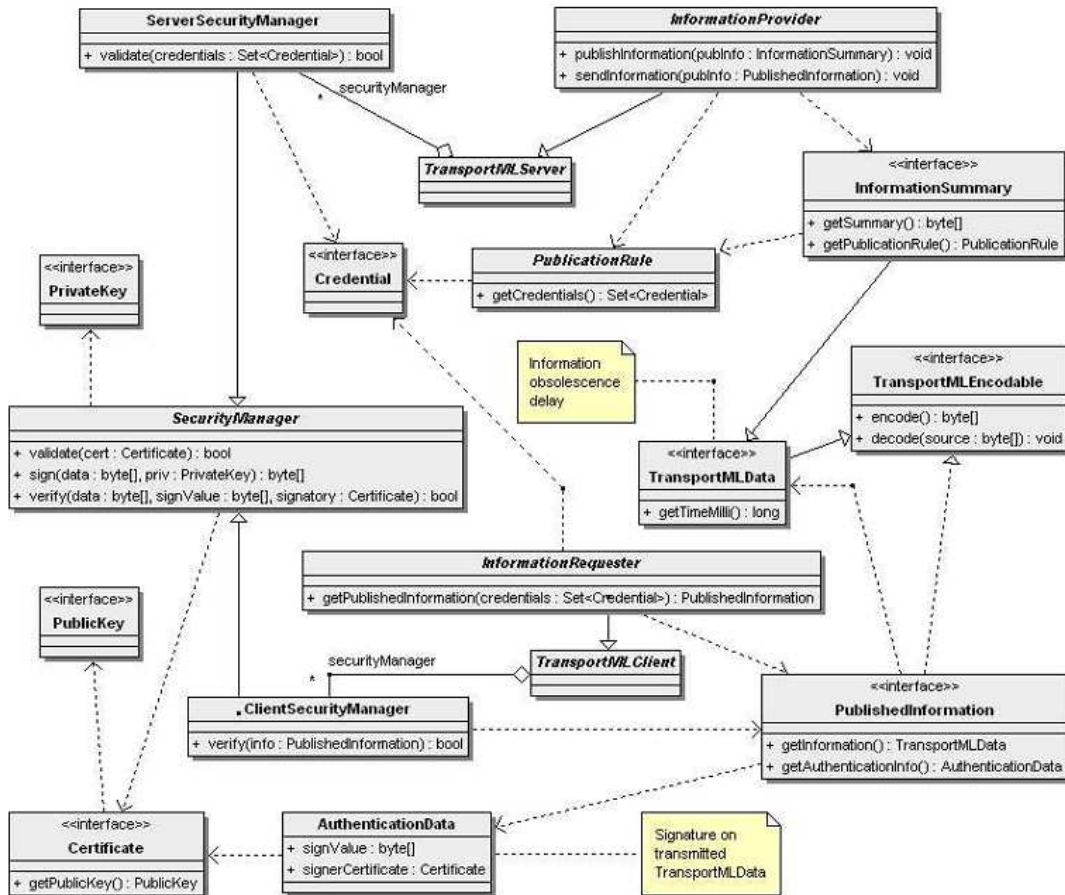


Figure 6. TransportML components' organization

Independently from the network layer (TCP/IP), TransportML derives from web services Simple Object Access Protocol (SOAP) exchange format and adds authentication information provided by XML-based electronic signatures [18].

## 5.1 The Service Discovery Framework

The TransportML middleware is responsible for discovering services answering a particular user query. As it must allow gathering information from several business services, we adopted the Semantic Web vision.

A Semantic Web Service is different from a traditional Web Service, in that it contains semantic annotations that automate the service life cycle [14]:

- The **Service Modeling Phase**, during which is described the service which is being looked for;
- The **Service Discovery Phase**, during which the service requester tries to locate the eventual service providers able to deliver the service modeled in the first phase;
- The **Service Definition Phase**, during which the service requester and the service provider come together in order to define the details of the delivered service;
- The **Service Delivery Phase**, during which other service provision related interactions occur between the two parties.

A service semantic annotation indicates “*what the service does*”. The vocabulary needed for such descriptions is defined in terms of ontologies [14]. Ontologies allow determining main vocabulary terms that characterize service’s capabilities in a “machine-processable” language [23].

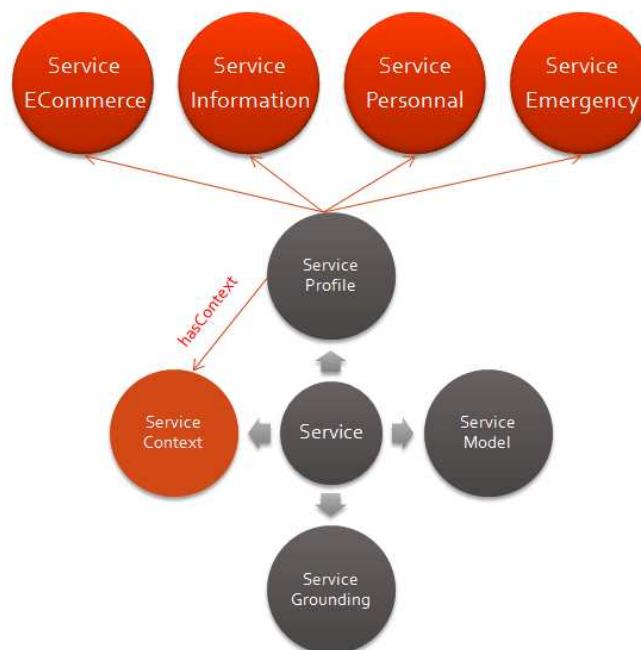


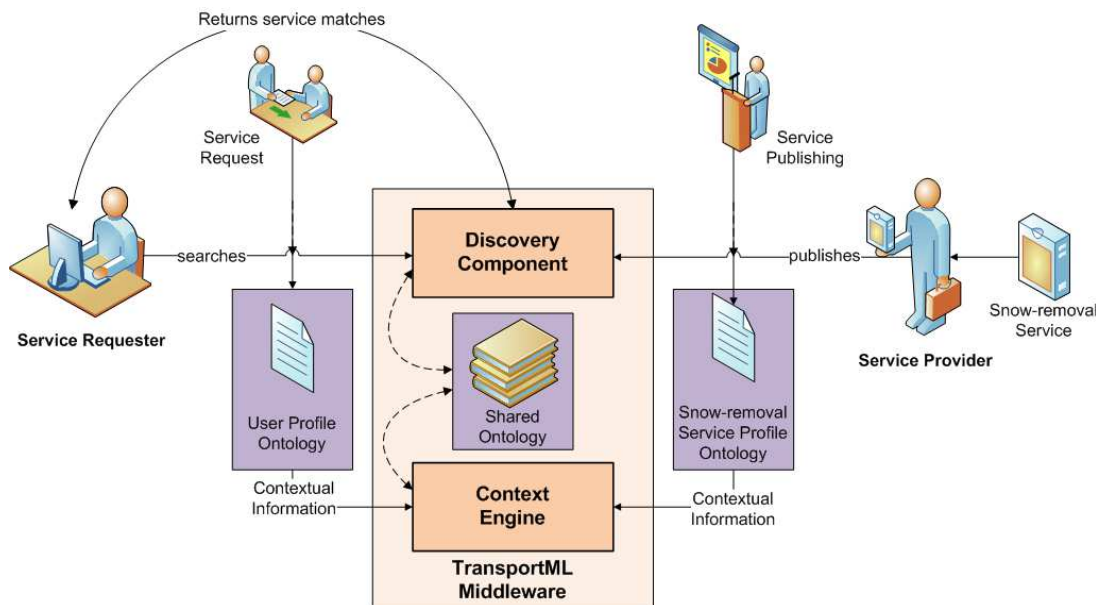
Figure 7. TransportML ontology for service profiles.



Services' semantic descriptions are publicly advertised in order to be consulted by mobile agents representing the users. In order to match these service descriptions to users' requests, a semantic matching engine must be implemented.

TransportML will operate in open and distributed environments, involving numerous interactions between users' devices and software components. As traditional semantic service discovery protocols mostly rely on keyword searching techniques, a novel semantic-based service discovery architecture must be defined. The TransportML service discovery framework will support mobility, semantic heterogeneity and also secure communications.

Every user will have a user profile represented through the user ontology. Every service will have its service profile ontology that will define its capabilities and requirements. Both user and service ontology are described using the OWL-S ontology (see Figure 2 for the TransportML core ontology and Figure 7 for the service ontology). Figure 8 illustrates the concept of the TransportML service discovery architecture.



**Figure 8. The TransportML Service Discovery Architecture.**

The main role of the discovery framework is to determine which service answers best a user's request. In doing so, the user's profile ontology must be matched against the service's profile ontology (Figure 7). The resulting matching process needs a common ground on which ontology comparisons can be based [19]. Therefore context information will be represented through a shared global ontology, notably the SOUPA ontology (Standard Ontology for Ubiquitous and Pervasive Applications) [15]. The SOUPA ontology defines generic concepts used in pervasive and ubiquitous computing environments: person, time, space, event, etc. This allows having common context knowledge for the two ontologies to be matched. The result of the matching process will be a list of available services, ranked according their relevance to the user's request.

A recent publication [31] demonstrates the benefits of Spatial Registries to register and discover location-based services. The Discovery Component of TransportML may deploy a module to invoke Spatial Registries, thus allowing scalability while preserving a local vision when matching best services.

## 5.2 Secure Communications

### 5.2.1 The Security Manager

TransportML takes advantage of an internal security manager which allows applications security and TransportML services security separation. TransportML security manager is composed of client-side and service-side modules.

The client-side module makes sure that the outgoing request satisfies a-priori the corresponding service execution. The request can be cancelled in case the requester does not hold the mandatory credits or does not support required security protocols or algorithms.

The service-side security manager first verifies that the incoming request conforms to the expected XML format (according to DTD, XML Schema and Schematron constraints for example). The manager then checks that the request digital signature is valid and that the signatory has the necessary credits to ask for the service.

### 5.2.2 Digital Signature Creation and Validation Processes

Messages authentication and integrity relies on digital signatures along with X.509 version 3 public key certificates [22] (modeled using "Certificate") which bind a public key and an identity. Digital signatures are associated with messages (requests – responses) so that receivers can verify their integrity and validity using revocation lists [22] or on-line protocols [25].

Messages can be wrapped into secured communications protocols such as TLS [17] to ensure confidentiality, when needed. Client-side TransportML security manager is composed of two complementary layers: requester's signature creation layer and accreditations gathering. Accreditations' composition is performed as defined in [16].

Server-side TransportML security manager is also composed of two layers: the first layer aims at verifying the requester's digital signature and the second layer validates the requester's signature associated accreditations (credentials) by means of Access Control Lists [16]. It globally considers all digital signatures to allow or deny accessing the service.

### 5.2.3 Common Horizontal Facilities

TransportML includes complementary facilities related to vertical business services interactions such as information publication and security.

The information publication facility allows vertical services sharing part of their proprietary information along with visibility. This visibility flag depends on requesters' credits and make the information opaque or unavailable to non-authorized third parties.

Getting access to the whole protected information may require complementary credentials held by other requesters. These credentials can be obtained my means of a security and composition facility. This facility is provided to help requesters

gather credentials and therefore allow services composition. A possible implementation may derive from [16].

## 6 Data Communication

The very concept of LBSs rests on the ability to communicate while being mobile. In traditional communication systems, the need for LBS has never been keenly felt; it only became necessary to consider this concept as communication in a mobile environment became possible. Such mobile communication is enabled by wireless transmission.

TransportML can be built on top of many different system architectures. Their common task is to provide communication among different entities, whether they are mobile or fixed, and the TransportML middleware uses this facility to communicate (e.g., position information between a mobile terminal and a distant service provider). But the differences in system architectures will affect what type of communication support TransportML can expect (e.g., with regard to the speed and cost of updating location information).

This chapter introduces the basics of the most common architectures of mobile communication systems and examines what functionality they can offer to the realization of the TransportML middleware.

### 6.1 Choice of the Transmission Medium

In order to build a wireless communication system, a transmission medium is required that can transport information without requiring a tethered connection between two communicating peers. In principle, different wireless media, such as ultrasound, infrared light, or electromagnetic waves in the radio spectrum, can be used. Each of these media has its specific advantages and disadvantages; however, for the purposes of LBS considered in this book, radio wave communication is the appropriate choice because it can support sufficiently high data rates (between a few kilobits per second up to some tens of megabits per second) over acceptable distances (ranging from a few meters to hundreds of meters or even kilometres) even when the participants are moving about. These properties, however, cannot all be simultaneously maximized; there are some inherent tradeoffs between them, which are discussed in the next section. In brief, the larger the distance or the higher the speed, the lower the possible data rate.

### 6.2 Communication between mobile nodes (vehicles) and the TransportML middleware

No matter how the trade-off between data rates, mobility, and range is cast, wireless communication does have a limited range: It is not possible for two arbitrarily distant partners to communicate with each other. Hence, provisions have to be made by a mobile communication system to enable such long-range communication. Essentially, three different approaches can be taken: (1) infrastructure-based systems, and (2) ad-hoc multihop systems.

## 6.2.1 Infrastructure-based systems

To overcome the limited range of wireless communication, a wired infrastructure is introduced. A mobile terminal communicates wirelessly with a device, commonly called a base station or an access point that is connected to a fixed, wired network. Such a base station receives the mobile terminal's wireless communication and sends the data toward its actual destination, which could be another mobile terminal communicating with another base station or a device directly located to a tethered network.

The typical structure of such a communication system contains several base stations, each covering a certain area; for each terminal in this area, the base station ensures that data can be transmitted to and from the wired network. Because these areas are usually called cells, this type of system is also often called a cellular system.

In this category, our team has selected the GPRS system as a testing candidate.

GPRS is a packet oriented mobile data service available to 2G cellular communication system users. GPRS provide data rates between 56-114kbit/s. The strength of GPRS comes from good coverage in Europe. It is also well suited to V2I (vehicle-to-infrastructure) communication because it is robust and almost unaffected by the speed of the mobile. Finally, GPRS use is made easy because the required equipment is cheap. However, GPRS suffers from some shortcoming too, such as the necessary subscription to a cell phone provider, its high latency and its relatively low throughput compared to other technologies such as Wifi (802.11 b/g/n/p).

With packet switching, a mobile station does not have to set up a connection to occasionally send or receive a packet, but can do so whenever data become available. The charging model, correspondingly, can be applied only to the amount of data that is actually transmitted, enabling an always-on usage pattern. GPRS can and typically will be combined with extended data rate via the wireless link, where the allocation of data rates for sending and receiving need not be symmetric—typical values like 53 kbits/second for receiving and 26 kbits/second for sending are expected [5].

Delay class	Packet size 128 byte		Packet size 1024 byte	
	Mean	95% percentile	Mean	95% percentile
1	<0,5s	<1,5s	<2s	<7s
2	<5s	<25s	<15s	<75s
3	<50s	<250s	<75s	<375s

**Table 4. Delay class specified for GPRS [6].**

In addition, GPRS allows specifying different quality-of-service profiles, which proscribe service precedence, the user data throughput, a choice between three reliability classes, and several delay classes. The delay classes in particular will

have a keenly felt impact on the practical usage because the GPRS standards only put rather weak demands on their implementations that will likely be quite different from the behaviour of a fixed Internet. Table 4 lists the delay values that the standard proscribes for two different packet sizes. As the table shows, even for low delay classes, there is a considerable chance that the packets will be substantially delayed. This fact renders highly interactive application patterns, which are likely, typical for LBS, problematic.

In addition, the variance of the packet delays can be quite high as the differences between the mean and the 95% percentile demanded by the standard is quite high. Large variance in packet delays is a difficult problem for many higher-layer protocols such as TCP to cope with. Actual measurements with the first instances of such GPRS systems corroborate the presence of highly variable delay or roundtrip times.

GPRS is a good candidate for V2I communication, which is widely used in this test. Its latency and its throughput are acceptable in the context of the considered test. As a consequence, GPRS will be used in this test for communication between vehicles and services, as presented in Figure 9.

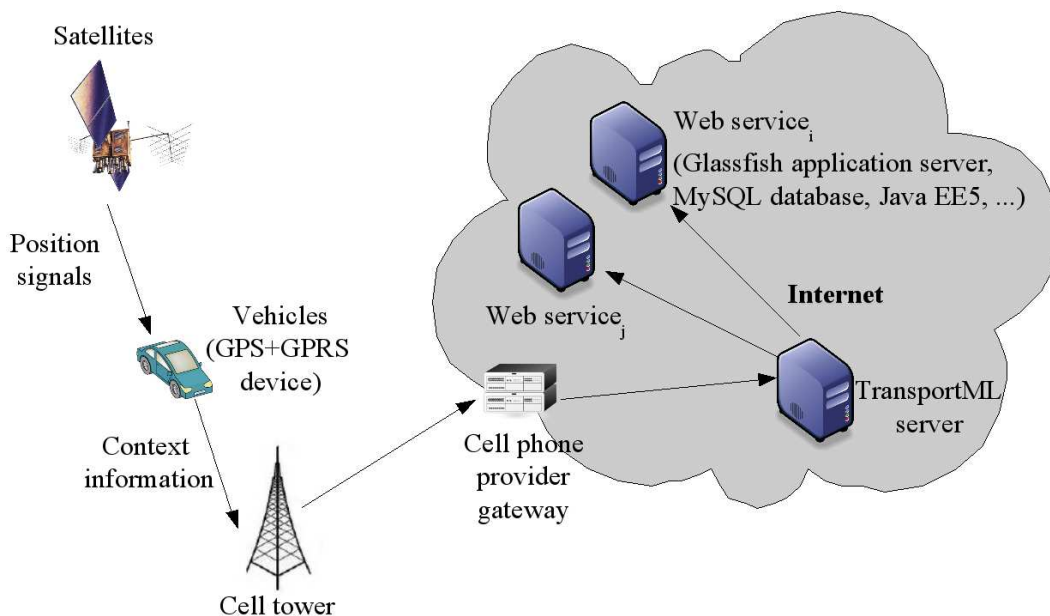


Figure 9. GPRS Communication Model for TransportML

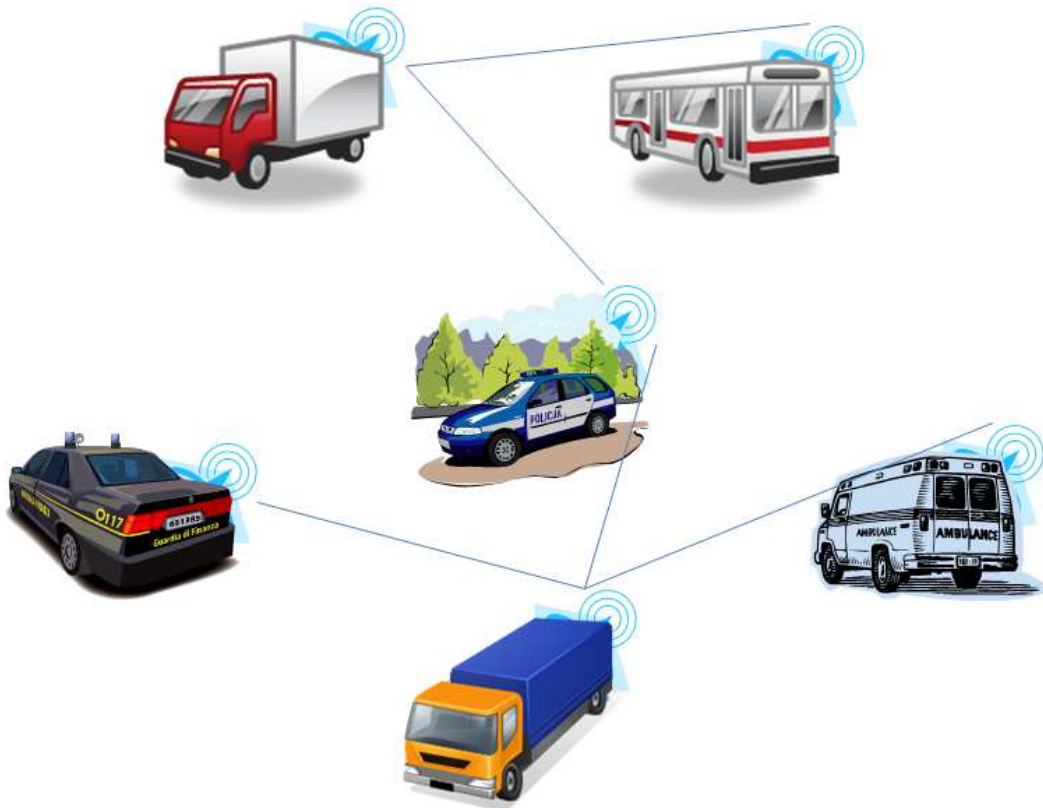
## 6.2.2 Ad-Hoc/Multihop Systems

An infrastructure is nice if faraway terminals want to communicate with each other. In some scenarios, however, all terminals could talk directly to each other as they are in their immediate vicinity; think of some laptops in a conference room, exchanging files via a wireless medium. In such a situation, infrastructure is neither necessary nor useful; it is conceptually much simpler to spontaneously set a network between these terminals in an ad-hoc (i.e., “for a specific purpose,” “improvised”) fashion. Figure10 shows such an ad-hoc network where all terminals communicate directly with each other.



**Figure 10.** Illustration of a direct-communication ad-hoc network.

Such ad-hoc networks are even conceivable when the terminals are not all in mutual communication range of each other. A terminal in the middle can then act as a “*relay*” for data coming from one terminal and forward the message toward its destination: a message travels over several radio hops, whereas in an infrastructure-based network, only one or two radio hops (if both sender and receiver are mobile) are used. Typical scenarios for such multihop ad-hoc networks are disaster relief operations (e.g., fire-fighters communicating wirelessly after an earthquake has disrupted existing infrastructure), construction sites or mining operations where setting up infrastructure is not possible, or cars informing each other of the traffic situation ahead (so-called vehicular networks). Figure 11 shows an example of such multihop ad-hoc networks; the lines indicate which members of this network can directly communicate with which other members.



**Figure 11. Illustration of a multihop ad-hoc network.**

A common characteristic of both single and multihop ad-hoc networks is the need to be self-organized in setting up and maintaining the network, without relying on additional outside infrastructure, even though members of the network are moving.

In this category, our team has selected the 802.11p wireless communication standard as a testing candidate. 802.11p is a draft amendment to the IEEE 802.11 standard [9]. The main goal of this draft is to offer wireless access in the vehicular environment, with regard to ITS applications' requirements [10]. The main enhancements defined concern data exchange between high-speed vehicles and between the vehicles and the roadside infrastructure. Communications occur in the 5.9GHz band. Indeed, the EC recently allocated 30 MHz spectrum in the 5.9GHz band for road safety applications [8].

The 802.11p draft is based on the ISO Communication, Air-interface Long and Medium range (CALM) architecture standard for vehicle-based communications. This standard defines a set of wireless communication protocols and air interfaces for a variety of communication scenarios in ITS. The CALM standard enables the following communication modes [11]:

- Vehicle-to-vehicle via ad hoc networking (V2V);
- Vehicle-to-Infrastructure (V2I);
- Infrastructure-to-Infrastructure (I2I) point-to-point connection where conventional cabling is undesirable.



The IEEE 802.11p promises “always-connected” communications that will allow drivers to benefit from more complete and up-to-date information about traffic hazards and congestion. This is the main reason why IEEE 802.11p communication should be tested for the ASSET project.

### 6.3 Communication between Services and TransportML middleware

Communication between services is generally achieved using Ethernet because they consist of Web services exposed on the Internet. The strength of this kind of communication is that it is standard (and thus interoperable), due to the very nature of the Web. This is even more interesting because TransportML aims at making services interact in a standard and interoperable manner. This type of communication represents I2I communication.

The communication architecture used in the test scenario is presented in Figure 12.

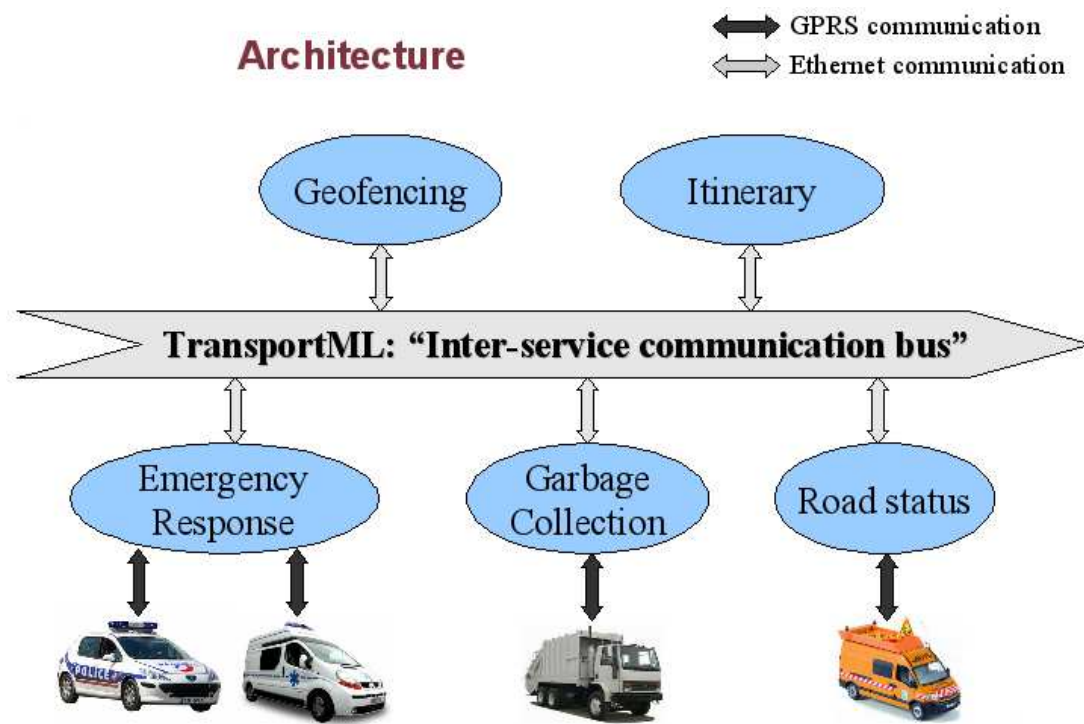


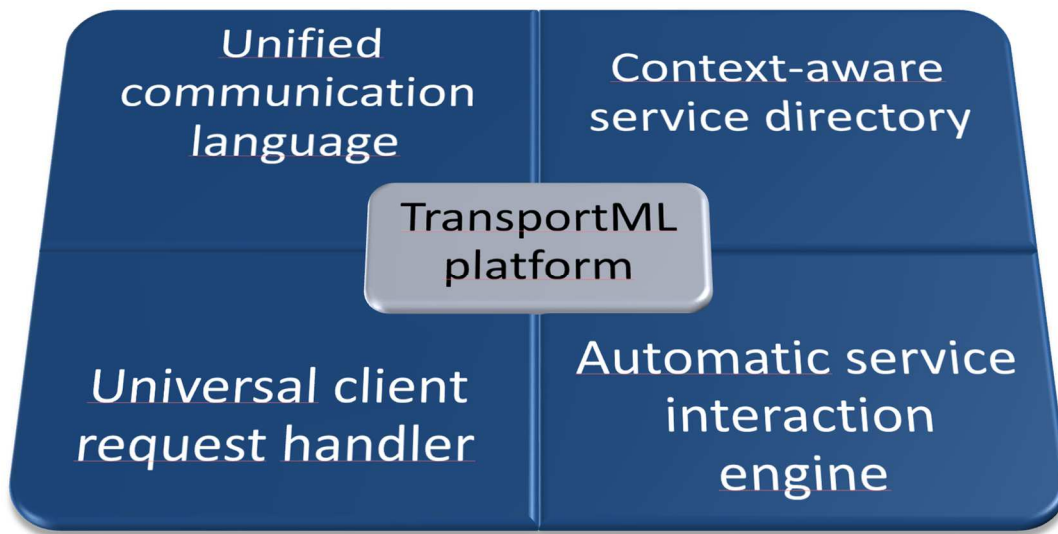
Figure 12. TransportML Global Communication Scheme.

## 7 Test prototype

### 7.1 Prototype operation

The TransportML platform prototype is composed of four core components as presented in Figure 13:

- A standard, unified and road-related language called TransportML (“ML” stands for “Markup Language”). TransportML is used as a standard communication language between the platform and other entities (users, services, etc.). The TransportML format is predefined, so there is no ambiguity.
- A context-aware directory where service providers can register their services so that the platform is aware of their existence and location. During the service registration, information regarding service capabilities should be provided. As a consequence, advanced service lookup is possible and allows finding services that can fill in a given TransportML document subsection. This component is a Web service providing two methods *register()* and *unregister()*.
- An automatic service interaction engine whose task is to elaborate on-the-fly composition (i.e. “*interaction*”) scenario using the service directory for discovery. Once the scenario is determined, the engine is in charge of contacting the services and matching/merging their input/output into TransportML documents. Most of TransportML intelligence is embedded in this core component.
- A universal (i.e. *standard*) user request handler which is able to process any user request, as long as it is formatted as a valid TransportML document. This core component is a standard Web service, providing a method *handleRequest(TML)* which takes as an input a TransportML document and returns a TransportML document with subsections filled in. This way, TransportML platform provides a standard and interoperable interface for user/client interaction.



**Figure 13. TransportML platform presentation**

As presented in Figure 14, TransportML platform complies with SOA principles. The platform is composed of two standard Web services: the universal user request handler for user/client interaction and the context aware service directory which provides means for registration of local Business Web services. Also, the automatic interaction engine acts as a client for local business services for the purpose of Web service invocation.

As a consequence, communication between the TransportML platform and other entities is achieved using the SOAP protocol. SOAP relies on XML for exchanging structured information with Web services. In our case, the exchanged payload information is a valid TransportML document.

In SOA, Web services' interface is described using WSDL (*Web Service Description Language*) that relies also on XML to present a Web service capabilities (i.e. provided methods with their input and output data).

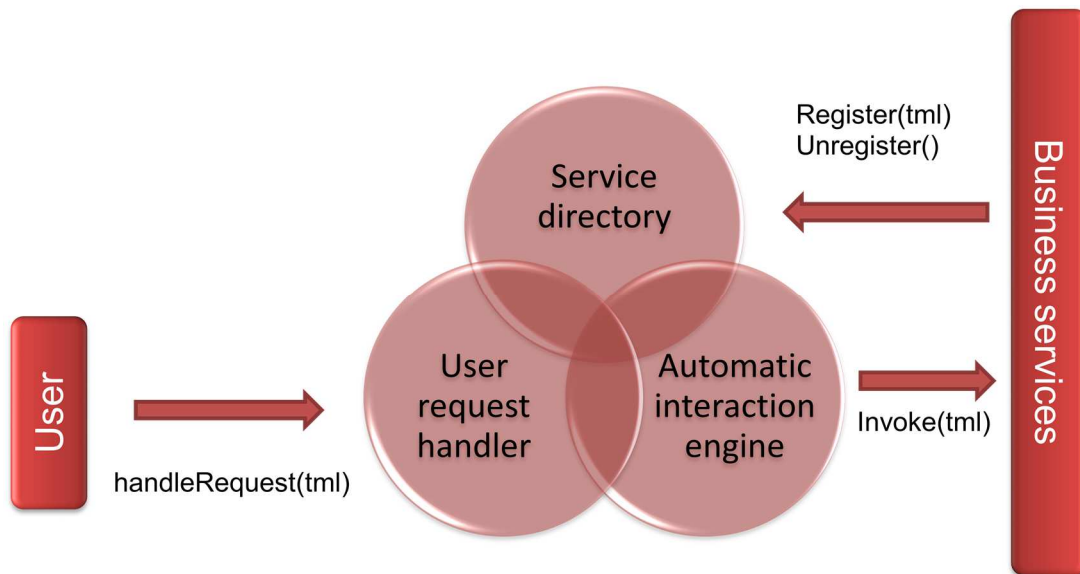


Figure 14. Interactions with TransportML platform

## 7.2 TransportML unified language

TransportML (short for “*Transportation Markup Language*”) is a transportation-centric language for formatting exchanged information between entities and the TransportML platform.

A standard and unified communication language is a prerequisite for automatic service interaction. Indeed, in TransportML platform, the interaction scenario is built on-the-fly, by examining a TransportML document corresponding to an user request.

An example of a valid TransportML document corresponding to an itinerary computation request is presented in Figure 15. As one can see, a TransportML document consists of XML document having a `<tml_document>` root element. This root element has one child element corresponding to the type of request: `<route>`, which in our case specifies that the user is asking for a route. We also refer to this element as a section of the TransportML document. The `<route>` element contains several children corresponding either to parameters, optional information or request result. In the selected example, the `<waypoints>` element contains the route request parameters. At least two waypoints must be provided : the start point and the end point. The `<advised_areas>` and `<discouraged_areas>` tags correspond to optional data which can be filled in by third-party services, before itinerary computation, in order to provide added value. Finally, the last child element, `<itinerary>`, contains the response to user’s request. All these elements are referred to as subsections of a TransportML document.

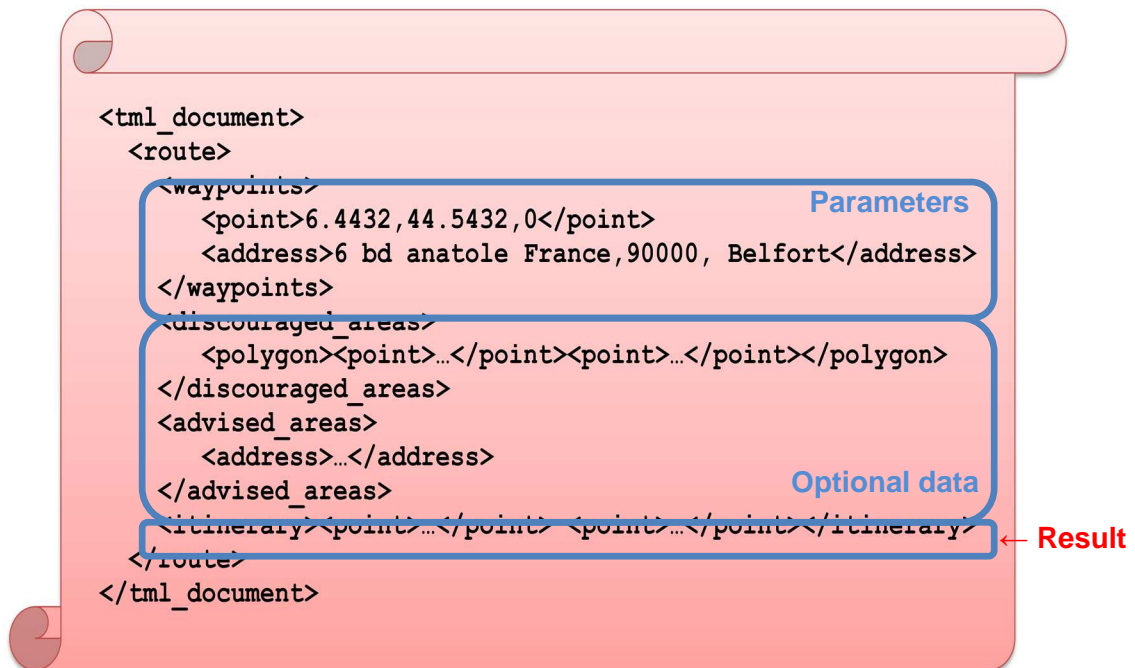


Figure 15. TransportML document example.

As illustrated by Figure 15, locations represent an important part of a TransportML document. TransportML currently supports three ways to express locations :

- <point>**

A point is characterized by its GPS coordinates (expressed after the WGS84 standard): longitude, latitude and altitude. It is the more precise and efficient way to express a punctual location.

A point is represented as follows:

```
<point>longitude,latitude,altitude</point>
```

e.g. : <point>6.4765,47.4356,0</point>
- <address>**

An address is a human-readable way to store an exact location. It is characterized by the following attributes: *street, postal code, city, country*. Since TransportML platform is only able to work on GPS coordinates, such an address is automatically converted to the corresponding (longitude, latitude) couple. This process is called *geocoding*.

An address is represented as follows:

```
<address>street, postal code, city, country</address>
```

e.g. : <address>3 rue jean jaurès, 90000, Belfort, France</address>
- <polygon>**

A polygon allows to express a non-precise location, that is to say an area. A polygon is characterized by at least three precise locations (i.e. <point> or <address>). Linking these precise locations forms a polygon, which defines *an area*.

A polygon is represented as follows:

```
<polygon><point>...</point><address>...</address></polygon>
```

e.g. <polygon> <point>6.4765,47.4356,0</point>

```
<point>6.4865,47.4356,0</point>
<point>6.4765,47.5356,0</point> </polygon>
```

User requests are expressed as valid TransportML documents. An example of such request is depicted in Figure 16. As one can see, the user provided the parameters (i.e. starting location and destination). Also, the user provided the optional subsections (<discouraged\_areas> and <advised\_areas>) but left them empty. This allows specifying that the user is interested in these subsections and that the TransportML platform should try to find third-party services able to fill in these sections. The same behavior is expected for the <itinerary> section, which corresponds to the request result.

```
<tml_document>
  <route>
    <waypoints>
      <point>6.4432,44.5432,0</point>
      <address>6 bd anatole France,90000, Belfort</address>
    </waypoints>
    <discouraged_areas/>
    <advised_areas/>
    <itinerary/>
  </route>
</tml_document>
```

**Figure 16.** User request example.

As mentioned earlier, TransportML documents are used to elaborate on-the-fly service interaction scenarios. This is achieved through the following steps (Figure 17):

1. TransportML platform receives a TransportML user request, such as the one in Figure 16. This request is then passed to the automatic service interaction engine.
2. The service interaction engine contacts the context-aware service directory in order to discover services that can fill in optional subsections (e.g. the <discouraged\_areas> section). Once matching services are found, the engine invoke those services and provides them the TransportML document corresponding to the initial request.
3. Once the interaction engine receives the resulting TransportML documents from the invoked services, it merges them in order to get a single valid TransportML document.
4. The interaction engine now has a TransportML document with parameters and optional subsections (filled in by third-party services). It is now time to discover services which are able to fill in the result subsections that the user request (e.g. the <itinerary> subsection). To achieve this task, the engine needs to contact the service repository again and select one best-matching service for each result subsection.
5. As for the optional subsections, the interaction engine will contact the select business services in order to fill in the result subsections.
6. Once the TransportML document is complete, the platform is able to answer the user request by sending him back the document.

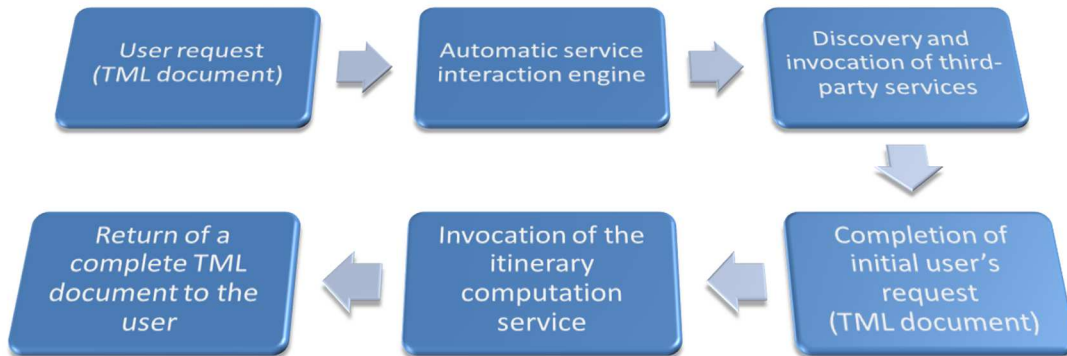


Figure 17. Steps of completion for a TransportML-compliant user request.

### 7.3 Involved Services

TransportML test scenario in laboratory environment involves five location-based services. A location-based service is a special kind of service which makes use the location of the users in order to provide well-suited response to them (i.e. information relevant to the user's context).

Each one of these services was developed by UBM, for testing purpose. As a consequence, those services do not use yet information issued by local organizations (except advanced itinerary service which uses real geographical data) although they correspond to existing local services.

- **Waste collection service:** This service corresponds to the waste collection part of CAPM, which consists of 20 collection trucks and 58 agents which collect 37000 tons of waste per year. CAPM is tracking collecting trucks in order to optimize collection tours, ensure agents' safety, manage problems in real-time. Therefore, our test service exposes information regarding the real-time location of all collection trucks, although this information is fabricated at the moment.
- **Itinerary computation service:** the purpose of this service is to compute the fastest route between two geographical locations. The service allows the user to define waypoints as well as advised or discouraged areas. We developed this service internally using geographical data (GIS files) issued by local institutions.
- **Emergency service:** This service is developed and simulated internally in order not to disturb existing local services. Emergency vehicles are replaced by test site vehicles which are equipped with tracking devices as well as embedded computers. Vehicles will consist of both cars and motorbikes. In our test prototype, emergency vehicles correspond to users of the

TransportML platform. The objective is to reduce emergency response time thanks to intelligent interactive services and inter-organization cooperation.

- **Road status service:** In real life, this service is also provided by CAPM. It exposes information regarding road status issued by field agents which are driving throughout the area. In our test prototype, we developed similar services with fabricated information. It is composed of two sub-services:
  - o **Snow clearance service:** Provides information regarding snow clearance status: Cleared roads or areas where traffic can be difficult.
  - o **Public works service:** Provides information regarding road works or damaged road surface.

### 7.4 Test scenario Architecture

The architecture used in this test scenario is based on TransportML standard architecture which was presented earlier (Figure 3).

The resulting architecture is presented in Figure 18 and it includes all services which were introduced in previous section. The Figure illustrates that the user (here an emergency response vehicle) provides a TransportML document corresponding to a *route* request, with waypoints parameters. As one can see, some services (namely the *waste collection service* and *public works service*) are able to fill in the `<discouraged_area>` subsection, which is optional. Snow clearance service is able to fill in the second optional subsection, which is `<advised_areas>`.

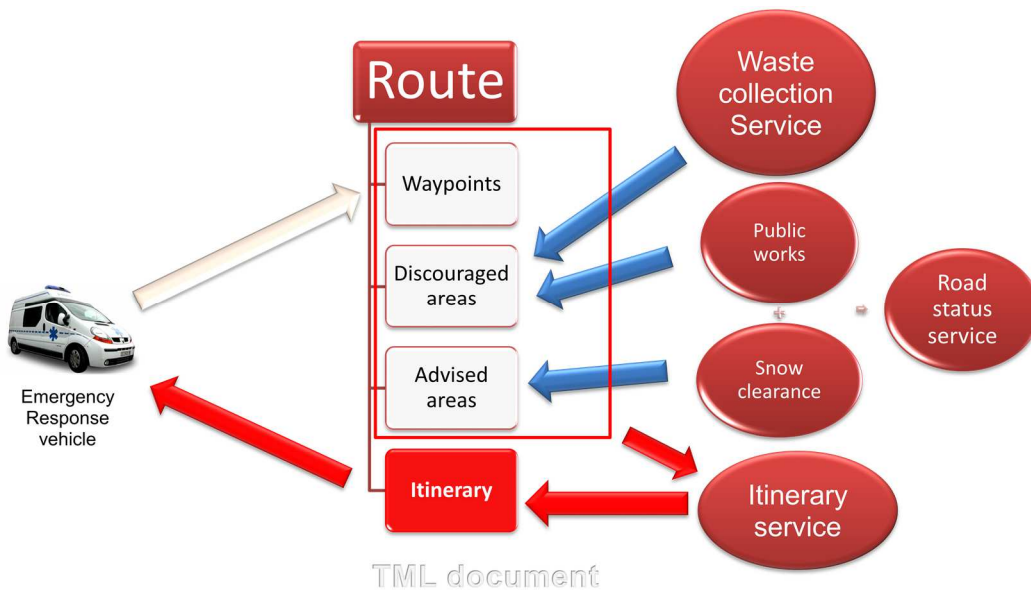


Figure 18. Test scenario architecture.

### 7.5 Performance Metrics



## 8 Conclusion

The development of middleware technologies tailored to the special needs of LBSs is essential for their rapid creation and deployment. The traditional middleware focusing on distributed computing does not provide suitable APIs and infrastructure services. Rather, it is required to offer convenient tools that hide the complexity of positioning and related tasks, for example, quality aspects, privacy, and billing, from the application developer. For this purpose, the middleware has to span the entire LBS supply chain ranging from the LBS target to the LBS user.

In order to cover all LBS' aspects, this document has investigated LBS characteristics and their requirements for the TransportML middleware. This middleware has therefore been developed with regard to these requirements. The TransportML middleware has been specifically developed to suit LBS requirements. Its architecture, components and way of functioning have been illustrated.

## 9 References

- [1] European Location-Based Services: Operator Status and Market Drivers Concise Insight, Ltd., London, 2003.
- [2] Caller Location in Telecommunication Networks in View of Enhancing E112 Emergency Services: Recommendation Towards a European Policy and Implementation Plan, Helios Technology Ltd., prepared on behalf of the Directorate General Information Society, Brussels, Luxembourg, April 2002.
- [3] Location: Not Quite Here Yet, OVUM, London, 2002.
- [4] Want R., Hopper A., Falcao V., and Gibbons J.. "The Active Badge Location System." ACM Transactions on Information Systems, 10(1):91–102, January 1992.
- [5] Roth J.. Mobile Computing Grundlagen, Technik, Konzepte. (In German.) Dpunkt Verlag, 2002.
- [6] Schiller J.. Mobile Communications. 2nd edition, Addison-Wesley, London, 2003.
- [7] Update on U.S. E911 Mandate, Mobile Location Services Conference, MLS, Amsterdam, 2003.
- [8] "Cars that talk: Commission earmarks single radio frequency for road safety and traffic", IP/08/1240, Brussels, 5 August 2008. Available online at: <http://europa.eu/rapid/pressReleasesAction.do?reference=IP/08/1240&format=PDF&aged=0&language=EN&guiLanguage=en>
- [9] Status of Project IEEE 802.11 Task Group p - Wireless Access in Vehicular Environments (WAVE): [http://grouper.ieee.org/groups/802/11/Reports/tgp\\_update.htm](http://grouper.ieee.org/groups/802/11/Reports/tgp_update.htm)
- [10] [http://en.wikipedia.org/wiki/IEEE\\_802.11p](http://en.wikipedia.org/wiki/IEEE_802.11p)
- [11] [http://en.wikipedia.org/wiki/Communications,\\_Air-interface,\\_Long\\_and\\_Medium\\_range](http://en.wikipedia.org/wiki/Communications,_Air-interface,_Long_and_Medium_range)
- [12] Bakhouya M., Gaber J., "Ubiquitous and Pervasive Application Design. Encyclopedia of Mobile Computing & Commerce, Eds. D. Taniar, Idea Group Pub, ISBN: 978-1 59904-002-8, Feb. 2007.
- [13] Bakhouya M., Gaber J., "Service Composition Approaches for Ubiquitous and Pervasive Computing Environments: A Survey". Agent Systems in Electronic Business, Vol. 3, Ed. Soe-Tsy Yuan, Idea Group Pub. 30, 2007.
- [14] Cardoso J., Sheth A. P., "Semantic Web Services, Processes and Applications", Springer, 2006.

- [15] Chen H., Perich F., Finin T., and Joshi A.. "SOUPA: Standard Ontology for Ubiquitous and Pervasive Applications", available at: <http://pervasive.semanticweb.org/ont/2004/06/>. In Proceedings of the International Conference on Mobile and Ubiquitous Systems, pages 258-267, Massachusetts, USA, August 2004.
- [16] Cottin N., "Contribution à la sécurisation des échanges électroniques en environnement réparti objet". PhD thesis, December 2003.
- [17] Dierks T., Rescorla E., "The Transport Layer Security (TLS) Protocol Version 1.2", RFC5246, August 2008.
- [18] Eastlake 3rd et al., "(Extensible Markup Language) XML-Signature Syntax and Processing", RFC3275, March 2002.
- [19] Euzenat J., Shvaiko P., "Ontology Matching", Springer-Verlag Berlin Heidelberg 2007.
- [20] Gaber J., "New Paradigms for Ubiquitous and Pervasive Applications". Proc of 1st Workshop on Software Eng. Challenges for Ubiquitous Computing, Lancaster, UK, 2006.
- [21] Gallagher T., Jeffries B., Landauer L., "Hunting Security Bugs", Microsoft Press, 2006.
- [22] Housley et al., "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) profile", RFC3280, April 2002.
- [23] Kompatsiaris Y., Hobson P., "Semantic Multimedia and Ontologies – Theory and Applications", Springer-Verlag, London, 2008.
- [24] Loke S., "Context-aware Pervasive Systems – Architectures for a New Breed of Applications", Auerbach Publications, 2007.
- [25] Myers et al., "X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP", RFC2560, June 1999.
- [26] Nait-Sidi-Moh A., Bakhouya M., Gaber J., Wack M., "A Contract-based Approach for Monitoring and Controlling Services Accesses in Ubiquitous Computing", NTMS'07 Workshop, Paris, May 2007.
- [27] Roxin A., Gaber J., Wack M., "Wireless Geolocation", Research Report, University of Technology of Belfort-Montbéliard (UTBM), France, 2006.
- [28] Roxin A., Gaber J., Wack M., "Location-based models for pervasive road networks", IEEE ICPS'07 workshop, July 2007.
- [29] Roxin A., Dumez C., Gaber J., Wack M., "Middleware Models for Location-Based Services", ACM ICPS'08 (International Conference on Pervasive Services), 2<sup>nd</sup> AUPC Workshop (Agent-Oriented Software Engineering Challenges for Ubiquitous and Pervasive Computing).

- [30] OASIS, "Web Services Security: SOAP Message Security 1.1 (WS-Security 2004)"
- [31] Cottin N., Wack M., "Dynamic Registration and Discovery of Location-Based Services Using Spatial Registries", to appear in ACM ICPS'09 AUPC workshop.