COSMOS

Cultivate resilient smart Objects for Sustainable city application

Grant Agreement Nº 609043

D7.5.2 Smart events and protocols for smart public transport (Year 2 Functionality)

WP7: Use cases Adaptation, Integration and Experimentation

Version: 1.0
Due Date: 31/10/2015
Delivery Date: 30/10/2015
Nature: Report
Dissemination Level: Public

Lead partner: Empresa Municipal de Transportes de Madrid S.A. (EMT)
Authors: Sergio Fernández Balaguer (EMT)
Andrés Recio (EMT)
Internal reviewers: Juan Sancho (ATOS)

www.iot-cosmos.eu

The research leading to these results has received funding from the European Community's Seventh Framework Programme under grant agreement n° 609043

Version Control:

Version	Date	Author	Author's Organization	Changes
0.1	15/10/2015	Sergio Fernández Andrés Recio	EMT	First updates to the D7.5.1 (Y1), creating the initial version of D7.5.2 (Y2)
0.2	19/10/2015	Sergio Fernández Andrés Recio	EMT	Second version
0.3	26/10/2015	Juan Sancho	ATOS	Internal review
0.4	28/10/2015	Francois Carrez	UNIS	Internal review
1.0	30/10/2015	Sergio Fernández Andrés Recio	EMT	Final version

Table of Contents

1 Introduction ... 6
2 Smart Labs and research projects ... 7
2.1 Smart Labs and research projects in Y2 ... 8
3 The commercial approach in the Smart Cities environment ... 9
4 Smart Transport in the field of Smart Cities ... 10
5 Smart Cities Public models vs. Private models ... 11
5.1 Data model dissemination during Y2 ... 11
6 COSMOS within the Smart Public Transport infrastructure. Standardization and protocols proposal ... 12
6.1 Virtual Entities associated or connected with public transport: schema and ontology definition ... 12
6.2 New Virtual Entities for Y2 ... 12
6.3 Specialization of data model in the storage system during Y2 ... 12
6.4 Big Data Model for COSMOS Machine Learning (new data model concept for Y2) ... 13
7 Transport oriented events server ... 15
7.1 Smart Events integration client ... 15
7.1.1. Queues and messages for Y2 ... 15
7.1.2. VEProt description ... 16
7.2 Transport events exchange protocol ... 17
7.2.1. Header context ... 17
7.2.2. Control Header ... 18
7.2.3. Virtual Entity Identifier or Request Original System ... 19
7.2.4. Identifiers of the target system ... 19
7.2.5. Identifiers of instructions and data ... 20
7.2.6. Changes done in Y2 within the VEProt datagram model ... 21
7.2.7. Response indicators ... 24
8 COSMOS opportunities under the Smart Transport context ... 26
9 Conclusions ... 27
10 Recommendations for Y3 ... 28
11 Bibliography ... 30

Table of Figures

Figure 1: Madrid mobility lab 8
Figure 2: Public and private cooperation schema 27
Figure 3: Proposed local working group for Y2 28

Acronyms

Acronym	Meaning
AENOR	Spanish Association for Standardization and Certification
AMQP	Advanced Message Queuing Protocol
CEP	Complex Event Processing
DDBB	Data bases
DDP	Distributed Data Protocol.
ICT	Information and Communications Technology
ITS	Intelligent Transportation Systems
JSON	JavaScript Object Notation
JSON-LD	JavaScript Object Notation Linked
NoSQL	Not only SQL
RB	Reactive Box
RDF	Resource Description Framework
SQL	Structured Query Language
TCP	Transmission Control Protocol
UC	Use-Case
UDP	User Datagram Protocol
UNE	A Spanish Standard
VE	Virtual Entity
VEProt	Virtual Entity Process for Reactive and Ontological Things
WGS84	World Geodetic System 1984

1 Introduction

As stated in the last D7.5.1, the development of intelligent systems oriented to Smart Cities requires coordination and the sum of synergies which, slowly, could lead to the expected changes. A clear example is the development of specialized sensors by companies and industries that have emerged in the wake of the niche market. This would not have been possible without setting up intelligent elements that offer business opportunities for those companies. In the meantime, these mechanisms and sensors have enabled smart cities to have physical and virtual elements on which to rely. Thus, the slow and constant work on this regard has meant to a mutual enrichment for sides, public and private sectors.

To achieve this, it has been important the investment in R&D of many entrepreneurs and managers who are anticipating the future, but also the work of theoretical researchers, docents and universities which have established guidelines and protocols for the virtualization of cities avoiding becoming chaotic. In addition to that, the role of cities, becoming city labs where to test and incorporate real projects has made possible the knowledge of the requested infrastructure to keep on working on this field.

As aforementioned before, within the objectives of COSMOS the aim is to follow the same approach, which equivalence will be expressed in D7.5.2 (the current document addressing the Y2 achievements) and the future D7.5.3 (for Y3) deliverables. This development model will start from classical systems for integrating transport data events towards a model of virtualization of such events, following guidelines for standardization, orchestration of services and public open data policies, designing protocols to be used in the exchange of information by citizens, businesses and institutions, even beyond the model of public transportation. That will be the contribution of COSMOS within the research field of Smart Cities and, more specifically, within the concept of smart transport. The transport oriented protocols which are being defined under the framework of COSMOS are the basement of the system predictive models that will deployed between Y2 and Y3.

2 Smart Labs and research projects

Multiple projects worldwide are serving for the development of information technologies aimed at creating ecosystems that allow learning how to use Information and Communications Technology (ICT) and Intelligent Transportation Systems (ITS) in order to make smarter and more sustainable cities. The R&D efforts focus on several strategies, including:

1. Sensing systems: there are new business opportunities under multiple pathways thanks to new adapted microcontrollers oriented to signal acquisition and actuators as well as thanks to the expansion of reduced cost sensors together by an increased supply promoted by an industry in full expansion. It is therefore essential to have laboratories that can quantify and qualify the different technologies that are appearing. In Spain, scenarios such as Malaga, Zaragoza and Santander, to give some examples, provide unique opportunities for research in the field of intelligent sensing and intelligent control of objects, mainly within the fields of energy and communication. In these cities, large companies have deployed laboratories in large areas of the city including data acquisition and control of different types of elements of the city, always talking under the framework of smart labs.
2. Event management and information systems in real time: the construction of platforms oriented to the analysis of events such as queue managers and other information models based on the observation of changes, provide trained technologies for the exchange of status information among sensors. Currently, multiple commercial technologies enable the deployment of event-driven systems.
3. Storage capacity and Big Data: new storage systems, based on schemas and objects, known as No-SQL databases, are allowing scaling data solutions covering cloud infrastructures, from which status can be stored in order to apply methods of learning and experience through the use of data and the relationship between them.
4. Definition of schemes and digital information entities: multiple standards allow machines to know about entities and objects, by its ontology and semantics, providing structured repositories and methods of consultation and access to information enabling intelligent definition methods of any element of the city, especially affecting sensors and actuators.

All these elements together and in combination are being experimented in different city labs, producing a paradigm shift in the new world that is being built, which is a combination between the classic concept of a city and the integration of elements that improve its energy efficiency, mobility and accessibility, among others. In summary, Smart Cities are emerging thanks to technology and the construction of living labs in cities, and this is the approach that COSMOS is facilitating in the Madrid use case.

2.1 Smart Labs and research projects in Y2

During Y2, with the support and under the auspices of COSMOS project, the participant partners have considered the possibility of making in Madrid the development of an intelligent laboratory, not only from an experimental point of view but also from a practical one. Therefore, and thanks to EMT initiative, a laboratory has been deployed to serve as a testing place to ongoing projects by providing feedback coming from user experience. This laboratory has led to the complete definition (and not only under theoretical and functional aspects) of the protocol for the exchange of events oriented to smart mobility. In addition to that this laboratory has been conceived as a tool for making real analysis of events, therefore it includes real information about public transportation in Madrid, including real time data, such as the position of the buses or expected arrival times to bus stops.

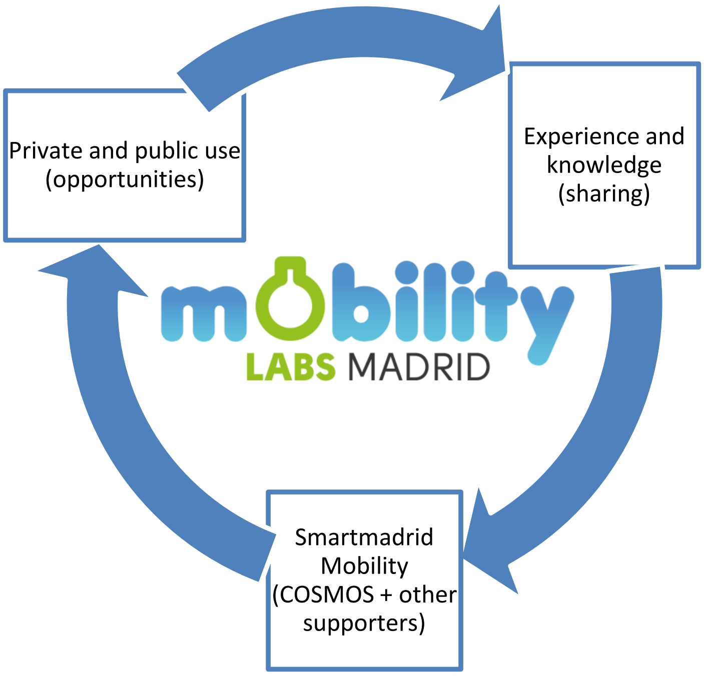


Figure 1: Madrid mobility lab

This laboratory has enabled, among others:

1. Obtaining knowledge and experience of existing semantic models, as well as tool learning such as “Protége” and gain experience in IoT oriented schemes such as Semantic Sensor Network.
2. Providing an experimentation environment to universities and companies in order to investigate, learn or develop new products.
3. Creating post-graduate courses at universities offering as a learning and practical field the laboratory itself.
4. Giving talks and disseminating COSMOS project in multiple environments, events and forums.

3 The commercial approach in the Smart Cities environment

One aspect to consider in the current deployment of Smart Cities is the possibility that large companies and corporations literally take technological control of cities. This is mainly due to their privileged position in industry and research sectors, which can involve, in many cases, the deployment of various components (sensors, devices, event management systems, etc.) based on proprietary protocols and proprietary encryption systems that prevent a democratic growth approach of infrastructure. The risk is particularly high in models of municipal concessions, in which the exploitation and management of the various systems and resources of the city are delivered during a certain number of years by companies that are responsible for its management. These companies can, in turn, install sensing and control systems based on closed protocols and infrastructure, aimed basically to control their range of management and to ensure that the operation is performed in the best economic and energy efficiency conditions.

However, although the purpose for these action plans is legitimate and belongs to the normal operating field, its design has several disadvantages and risks, including:

- Prevents the constant technological adaptation: as it is based on a closed model and usually supported only in the field of the deployed system itself.
- Produces opacity phenomena in the information: as citizens and other companies have not the means to interact with the smart city infrastructure.
- It may block business models based on local industry: as local industry might have a lack of integration possibilities for expanded uses of the network of sensors or actuators.

4 Smart Transport in the field of Smart Cities

One of the objectives pursued when advancing in the implementation of models of Smart Cities is an optimized transport from both the functional and economic point of view, including (especially when talking about public sector):

- Energy efficiency for cost savings. More energy efficient vehicles pose a significant reduction in costs, since fuel and energy is one of the biggest expenses in a public transport company. Vehicles with improved aerodynamics and propulsion systems with higher performance are part of the core of the research activities of vehicle manufacturers.
- Reducing emissions to achieve cleaner cities. Using cleaner fossil fuels and new technologies (catalysts, electric vehicles, hybrid vehicles with self-production energy, etc.)
- Optimization of routes and services. Inefficient or costly routes, obstacles on public roads, delays by blockages or slow traffic are causes of poor or inefficient service level. Better services can be achieved through optimization algorithms and processes, as well as applying greater control over the regulation of the fleet. Applying new regulatory measures and policies, such as dedicated lanes for public transport, increase the efficiency of the service.
- Quick resolution of incidents to avoid possible downtime. The public transport is subject to various endogenous and exogenous factors that will prevent planning to be accomplished. The affectation of an incident will largely depend on the knowledge of the events by the managers and the ability of a quick decision making transfer.
- Management of smart ticketing, with the aim of improving co and inter-modality. One factor that highly affects the efficient use of public transport is the creation of integrated tariff systems. Furthermore, the creation of registration systems for selling and accessing to public transport guarantee the control of demand, promoting management and spending management policies based on real data. The use of large areas of multimodal exchange stations and the analysis of origin-destination matrices coupled with fast access systems and contactless technologies for accessing or paying ensure comfortable and economical use of public transport.
- Instant detection of incidences. The use of sensors and alarms (detectors) in public transport vehicles ensure the anticipation of an event with major consequences. To this end, industrial sector is increasingly introducing elements for early detection, including Machine Learning processes as predicting issues before they actually occur is also very important.
- Integration with other systems of the city. Public transport is not an isolated element of the city, and its optimum performance largely depends on its links with other city infrastructures. Therefore it is necessary to foster joining relations between different infrastructures and systems creating supra-intelligent mechanisms. Some examples are the prioritization for buses at traffic lights or the coordination with security and emergency systems.
- Customer Information Systems. Ensuring that the public transport user has reliable information that allows instantaneous trip planning or, to take instantaneous decisions if the usual trip option is failing.
- Accessibility. Public transport should not be effective only with a majority of citizens but to all citizens. For this reason, it must provide means of access and service knowledge using channels and mechanisms accessible for all.

To achieve the aforementioned objectives, technologies and information systems pose one of the greatest challenges and, in turn, one of the main tools in the field of transportation, and ITS and ICT initiatives must be efficient and be based in technological models that guarantee the points described above.

5 Smart Cities Public models vs. Private models

Smart Cities offer a new range of niches in which private exploitation system models of transportation, energy and other sensorial elements are not the only viable infrastructure, as are not either those exclusively municipal or public. On the contrary, mixed and juxtaposed models between private and public systems can be a way to ensure continuity, scaling and co-participation in building smarter cities.

The proposed COSMOS model is intended to provide a possible architecture equivalent to those which have been the classical ones that have come to be the benchmark in the construction of cities, that is: public bodies concerned and in charge of ensuring the transmission and exchange of information from which any organism, company or citizen can create their business models or research activities. The proposed model is based on standardization, certification and co-participation of systems emulating the traditional functioning of cities, since this model is adapting naturally to the ecosystem of human societies.

5.1 Data model dissemination during Y2

During Y2, an important effort has been put into the dissemination of the data model oriented to mobility events, mainly through two different strategies:

1. Introducing COSMOS model, thanks to several workshops, to different departments of Madrid City Council, explaining clearly the benefits and projection of using such model for the welfare of the city. Two of the aforementioned workshops were done at the highest level, to top managers and heads of the departments of IT and Traffic and Energy Management control centers of the city.
2. Publicly publishing the model in Github and publicly disseminating the repositories and the available information, together with examples about the system functioning code to capture transport and mobility events.

6 COSMOS within the Smart Public Transport infrastructure. Standardization and protocols proposal

6.1 Virtual Entities associated or connected with public transport: schema and ontology definition

The definition of an intelligent transport model involves designing proper ontologies. Transport COSMOS model schemas and contents are being built on RDF and JSON-LD models. The definitions are proposed under European standards through the Spanish Association for Standardization and Certification, AENOR (CTN 178 AENOR group as defined in UNE 178301). The datagram message headers are being defined under schemes of context (@context) according the proposed standard by the JSON-LD community.

The Virtual Entities which are being defined under semantic schemes are:

- Line: Each journey trip made by the public transport service (bus line).
- Bus: Each of the transport units serving in a line (bus).
- Driver: Driver performing each specific service on the bus.
- Traveller: Bus user connected or integrated into the COSMOS infrastructure.
- Electronic Panel: Information device providing useful information to travellers about the service or incidents.
- Sensor: reading device or actuator connected to the transport infrastructure (bus or informative electronic panel).
- Event: Each of the messages containing information related to transportation. An event can be one bus passing by a bus stop, a traveller getting on the bus, an engine temperature alarm and, in general, any information generated in the system which gets integrated into the COSMOS infrastructure.

The semantic definition of virtual entities is available in open data format in the GITHUB web site:

https://github.com/madridopenlabmobility/MOBILITY-MADRID-virtual-entities

6.2 New Virtual Entities for Y2

During Y2 new important virtual entities have been aggregated:

- Bus Stop: It contains the information and attributes of each EMT bus stop
- Bus Events.: It collects each bus position every 20 seconds
- Bus Sensor: Specific bus sensors have been connected to COSMOS system in order to provide readings for the Machine Learning

6.3 Specialization of data model in the storage system during Y2

One of the most important topics to decide standardization and data storage structure is the criteria that will affect, undoubtedly, to the scaling and diversification potential of the structure and data that support it.

A key decision in the storage and management model of data of the different Madrid VE has been to establish the advantages and disadvantages of an unstructured model with respect to the traditional model that manages the transactional information of Madrid (unstructured

DDBB vs. relational DDBB). This information, stored in large SQL and Oracle servers, conditions the growth and scalability of the model, so finally it has been decided to separate the traditional model from that one oriented to VE which is the one that feeds COSMOS.

The separation of models in the field of transport means a serious challenge in establishing systems for data synchronization, especially in real-time environments. For this purpose, any model based on relational systems must contain mechanisms for detecting changes in the database and must be able to be observed in order to be transferred and transformed into an unstructured storage. Only through a homogeneous and timely flow over time exchange mechanisms can be really applied. The clearest example of this calculation is to obtain the bus arrival time a certain bus stop. This data, maintained in the kernel memory of the fleet control system must be transformed and built into a new model based on observation and IoT oriented.

Another important issue is how the storage and information management system integrates within the platform that will provide service to COSMOS model and other IoT based models. Since the final decision of Madrid Local Cloud solution for COSMOS is based on the construction of the Meteor based integrator element, it was considered that the most suitable model was the NO-SQL database together with the inclusive feature of that platform to monitor changes based in MongoDB queues. Both facts have been key in taking the decision about which database to use.

6.4 Big Data Model for COSMOS Machine Learning (new data model concept for Y2)

Within an intelligent transportation model, Big data can be considered as one of the most important elements when analyzing the processing and storage of information architecture. Factors such as the system capacity, the strategy of accessing to information, the organization of data or the speed of obtaining information can pose serious problems when using the system. Within the scope of COSMOS and always thinking about Madrid scenario, and especially thinking about the specific Madrid UC, several analysis have been done until taking the decision on how the various structures on which store the information will be, always considering using the same database server model based in MongoDB. Among the most important considerations in shaping the form of collections and documents we can include:

1. Memory size of indexes in order to have at least five years storage of historic data, considering the segmentation in two data servers: the first server stores the last 12 months with a complete index for each collection in RAM memory, while a second server stores five years without any index in memory.
2. Accessing and data organization strategy:
    a. For models based on data collected by buses and managed with the shape of trips, JSON structures nested arrays are used, which principal hierarchy is the identifier of the service and its basic data: driver, vehicle, line, start, finish and other basic elements. Then there is a main array whose elements contain each of the trips made and their identification data, such as schedules, number, etc. Each of the trips, contain an array with each of the titles of transport used and the number of travelers obtained.
    b. For models based on data collected by the bus and managed as events, JSON structures separated by nature are used but grouped by a main document that contains the identifying data of the service, as specified in the previous point. Next, each object (JSON subdocument) contains an identification of the type of event that contains and associated data with a fully variable structure. Thus, there are step events by bus stop, or every time a passenger gets on the bus,

or whenever there is an incident, a sensor reading, etc. All information is geo-referenced.

   c. Position event-based data obtained through the Reactive Box data are stored with the original format in which they were managed through that system. Special needs users tracking data belong to this category, as well as their check points, caregiver alarms and the historic data of each bus step by each bus stop.

3. Storage. In order to store the huge amount of data, there are special storage spaces within the Madrid storage lockers, always counting with local space, that is, without transferring to Clouds outside Madrid own infrastructure. Storage space has been reserved for five years, independently of COSMOS cloud model itself, since it is considered a part of the internal strategy of EMT for our own transport research activities.

7 Transport oriented events server

The city information should flow through systems and nodal connections capable to ensure the exchange of useful information for citizens and businesses. Communication architecture should be based on open standards and be public infrastructure through which data can be consumed and reused openly (or at least, those data that are capable of being shared). Thus, the information could be exchanged in multiple modes, that is, from citizens and private sectors to municipal systems and vice versa.

In order to share this information within the COSMOS system, Madrid proposes a Local Cloud based on an event server (Reactive Box) that would act as the only point of connection and exchange of mobility oriented events. This event server must be:

- Adaptive: allowing the addition of new layers of information quickly and dynamically, compatible with dynamic information models based in objects or virtual entities
- Reactive: allowing subscription to objects or entities by clients and multiple systems.
- Safe: supporting models based on public information and private data.

The nodal server systems should be, undoubtedly, municipally controlled considering the approach of Smart Cities, and orchestration of services should correspond to infrastructures under the rule of public sector, beside the fact that private contents compliant with the protocol and standardization standards of the city could pass through it or be exchanged through it.

COSMOS proposed model for the intelligent management of public transport event, is based on JavaScript Meteor architecture, supported by Node.js. This infrastructure allows real-time client subscriptions to exchange standardized events information. The management of change for publishing queue results towards subscribers would be made by observing changes in a Mongo DB database.

7.1 Smart Events integration client

In order to provide systems that enable rapid integration of events within the transportation environment, COSMOS proposes the development of a specific piece of software that allows the exchange of information between different transport entities. This system, called VEProt (Virtual Entity Process for Reactive and Ontological Things), is a software client that will be released under an Open Source model on Github. It aims at easing the exchange of events and at consuming/sending information about transportation as well as any other useful data using the events server of the city.

7.1.1. Queues and messages for Y2

An important issue that had to be analyzed in depth is the MessageBus mechanism of interaction within the Madrid infrastructure because, although the Reactive Box system has a mechanism of subscription-watching in real time through which the different virtual entities can be obtained, this system behaves, for security reasons, as a unidirectional mechanism. This is, as a pure event generator. Due to this fact, the model considers how VEProt messages coming from the Complex Event Processing (CEP) are inserted into the RB. On this regard, the following issues have been taken into account:

- Speed response: A reactive system must contain entry mechanisms that do not involve delays or blockages in the entrance. Therefore, any concept based on web services or REST instances of any kind must be removed.
- Concurrent messages: Similarly, the system must be dynamic and absorb an amount of data dimensioned to handle the expected flow of information, which could exceed 2k events per second, also guaranteeing the continuation in case of system crashes.
- Security control: All messaging flows must be subject to strict security rules on the connection, so that no event of entry into the RB and involving event or status changes in a VE can be violated by any foreign element to the object itself or to any instance stored in the associated layers.
- Standardization: The messaging flow management must require a standardized scheme based on VEProt, since this is the datagram chosen for the exchange of VE within the city of Madrid.

Among the different models analysed for incoming messages into the RB system, including alarms managed by the CEP within the scope of Madrid UC, it has been selected Rabbit MQ for its adaptability to internal requirements rose within the internal communication model system. The server has been published in amqp.emtmadrid.es

7.1.2. VEProt description

VEProt is a logical system for the exchange of information between the different processes that can interact within an open scheme with proprietary or third parties systems. Its purpose is to provide a means of transportation and normalized information flow by which two systems or devices can communicate.

VEProt uses JSON as the exchange protocol handshaking and its functionalities are implemented through listening ports; these ports are configurable according to each subsystem model. Thus, it is possible to develop variable and adaptable configurations to each device according to port availability. VEProt sends data frames from IP ports towards the devices or subsystems which are message recipients and, in turn, can listen and receive responses via another port.

The VEProt system, designed globally, is the set of processes and protocols that allow a low-level object within the paradigm of IoT, to communicate status and data and to connect sensors and react instantly to changing events. This system, originally designed to work within different devices in the field of smart city mobility, can be moved naturally to any platform or system which requires interoperability and reactivity, especially in city environments, providing a unified mechanism for both communication and processing of various existing elements and also for instant sharing information with the public.

VEProt is based on a message scheme containing the rules, format and information needed so the different message subsystems know the attributes, relationships and the information to be exchanged among VEs.

VEProt can communicate remote Virtual Entities (not visible to each other) through reactive mechanisms by using DDP (into Event Server of Municipality of Madrid) or, can communicate local VE through UDP / TCP Standard protocols.

VEProt is defined in two ways, depending on its behaviour. The first definition is in terms of the set of processes that enables a device, sensor or system to operate autonomously. These processes are developed through a multi-platform programming language (Python) and they are located within a local area or ecosystem, to which sensors or devices are connected.

In summary, the transformation of a sensor or device into a VE requires:

1. A manufacturer driver or a low-level routine resident in the system that has the device connected.
2. A communication interface with VEProt, consisting of the transformation process of the different actions, status and characteristics of the device into those of a VE by defining a datagram.
3. Installing VEProt software that allows two-ways communication among the device or sensor with other systems or with other VE, as well as performing autonomous actions.

Conversion processes of events from devices/sensors to virtual entities are performed at the far end of the communication, that is, at those points where the information can be processed (generally a computer) and which have contact with the device information at a low level (communication with the driver, access to the communication port, access to the interruption, etc.).

Communication frames (datagram) are defined by a common language, through which they can indicate:

- Who is it?
- What kind of Virtual Entity is it?
- What characteristics and qualities does it have?
- What actions does it want to perform on another system?
- How can it communicate?
- What does it want to communicate?

7.2 Transport events exchange protocol

The exchange of events information within a Smart City has to be based on business rules and structures of dynamic information, but known in advance.

Currently, there are models of information in a very early stage or which are not fully developed to allow open communication of events that permit co-management participation and exchange of public-private information. Therefore, COSMOS aims to provide a new vision that guarantees the possibility of exchanging any information.

This structure must ensure the inclusion of multiple sensors and data in the system, while different observers can subscribe to get the information that is of interest to them using the event server.

The proposed datagram contains the following structure:

7.2.1. Header context

The virtual entities included in the message manifest themselves through this structure that defines the context elements. Further details about the implementation of the elements of the dictionary definition can be found in http://www.w3.org/TR/json-ld/

Example:


```
"VE:prot":
[
  {
    "@context": "http://cosmos.org/contexts/cities",
    "city": "Madrid",
    "homepage": "http://www.emtmadrid.es/",
    "domain": "http://cosmos.org/contexts/domain#transport",
    "layer": "transport.emt.bus.cpucore"
    "VE": "source",
    "@type": "bus"
  },

  {
    "@context": "http://cosmos.org/contexts/cities",
    "city": "Madrid",
    "homepage": "http://www.madrid.org/",
    "domain": "http://cosmos.org/contexts/domain#sensor",
    "layer": "transport.emt.bus.sensor"
    "VE": "target",
    "@type": "co2sensor",
  }
]
```


7.2.2. Control Header

The header contains general information of the message frame.

- VE:header. Unique identifying label indicating that the text below is a header.
- VE:version. Contains the VEProt version used for communication.
- VE:id. Allows the introduction of a unique identifying value for the entity.
- VE:datetime. Date and time of the request or generated message.
- VE:dateexpir. Message expiration date and time.
- VE:coordinateX. Geographic longitude coordinate in which the message is generated. Format must be WGS84.
- VE:coordinateY. Geographic latitude coordinate in which the message is generated. Format must be WGS84.
- VE:coordinateZ. Altitude in meters in which the message is generated.
- VE:size. Message size (for check of integrity).

Example:


```
"VE:header": {
    "VE:version": "1.00",
    "VE:id": "8CDA1667-C31B-4723-8154-90B8992EDA2A",
    "VE:datetime": "2014-04-20 13:23:05.01234 UTC",
    "VE:dateexpir": "2014-04-20 14:00:05.01234 UTC",
    "VE:coordinatey": "40.6719856",
    "VE:coordinatex": "-4.089738",
    "VE:coordinatez": "853",
    "VE:size": "243"
}
```

7.2.3. Virtual Entity Identifier or Request Original System

It allows that the message's addressed or the message's intermediate manager knows about the message sender. The data included is:

- VE:source. Fixed label that indicates that the structure contains the identification of the source system of the message.
- VE:id. Unique identifier of that object within its data layer.
- VE:systemsource. System/Entity code that generates the information. The receiver system must recognize that code as it allows the segmentation and analysis of the message contents.
- VE:subsystemsource. Information origin subsystem. It will allow the segmentation of the information of a virtual entity in multiple schemes, depending on the specialization.
- VE:versionsource. Identifier of the version of the generated information by the system source, regarding the general diagram.
- VE:functionsource. Function that emits the message.
- VE:modelistening. Listening mode for the answers.
- VE:port. Listening port
- VE:address. Listening address for the message answer.

Example:

```
"VE:source": {
    "VE:id": "1234",
    "VE:systemsource": "EMTBUS",
    "VE:subsystemsource": "CORECPU",
    "VE:functionsource": "INTEGRATOR",
    "VE:versionsource": "3.02.76",
    "VE:modelistening": "SKT",
    "VE:address": "82.45.133.2",
    "VE:port": "1424"
}
```

7.2.4. Identifiers of the target system

It contains relevant information about who is the final addressee of the data frame. This way, segmentations can be done in the subscription mechanisms by receiver system, so only the systems that need that frame will be the consumers of that information.

It defines the system to which VEProt must send the information. It contains the following tags within its structure:

- EV:target. Unique label that identifies this block.
- EV:systemtarget. Identifier of the destination system. This is an identifier for the subscriber through which the subscriber who is the message addressee.
- VE:subsystemtarget. Identifier of the destination subsystem. A subscriber can identify its message and recognize where it will be redirected within its system.
- VE:functiontarget. Invoking function at the addressee system. It indicates the function type that must be invoked in the destiny subsystem.

Example:

```
"VE:target": {
    "VE:systemtarget": "VALUE",
    "VE:subsystemtarget": "VALUESENSOR",
    "VE:functiontarget": "READ"
}
```

7.2.5. Identifiers of instructions and data

They define the structure of the virtual entity that contains the data from the message body (on array type structures).It contains the following values:

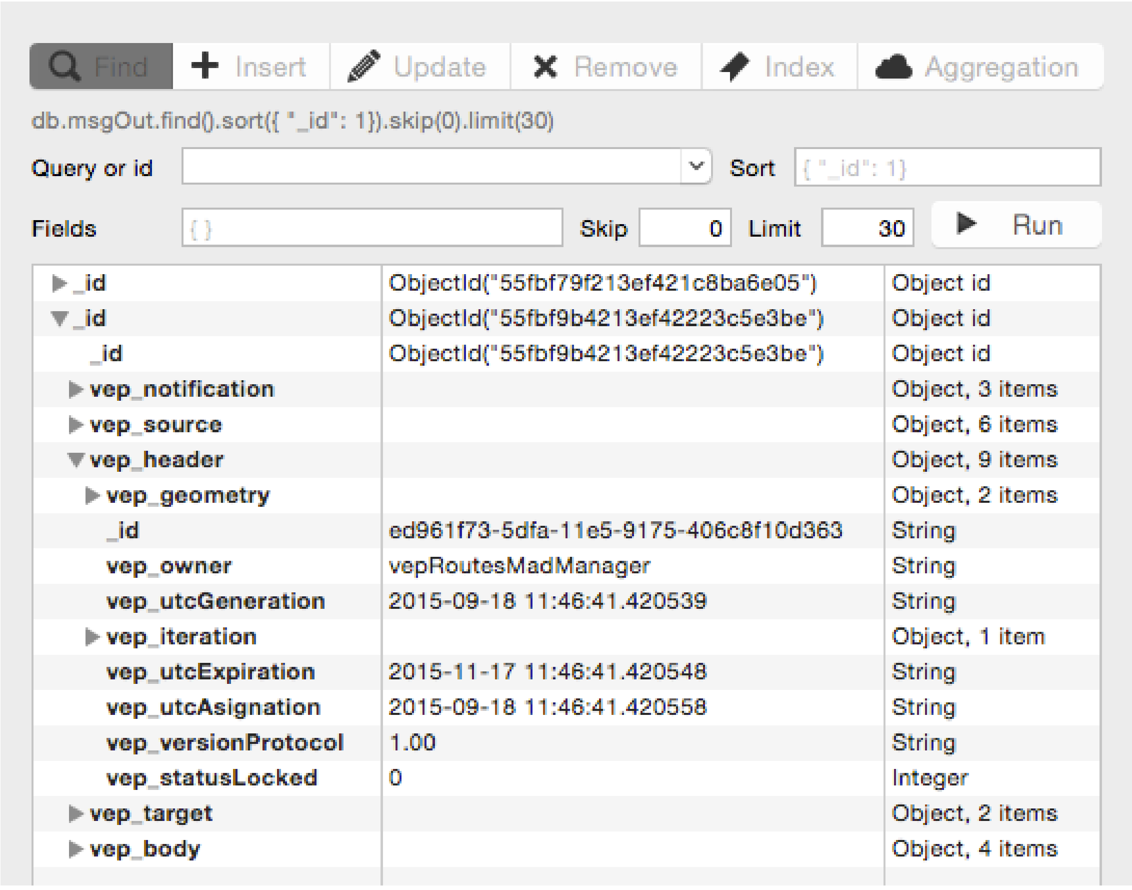
- VE:message. Unique label of the object. Message information exchanged between two virtual entities. Several of such structures may exist within the message to be exchanged.
- VE:id. Unique identifier of the data section within the overall message. It may simply be a sequential value.
- VE:confirmreq. Indicator response requirement. Value 0 = No requirement 1.- requirement.
- VE:timeout. Information or content validity time in milliseconds.
- VE:mode. Execution mode indicator. Value 0=Asynchronous (VEProt generates the call through one of its threads and continues with the execution of other processes or functions). Value 1= Synchronous (VECore stops the execution of its threads until response or timeout requested) call.
- VE:fiability. Reliability of the message; it will contain a percentage value whose maximum will be 100.
- VE:type. Defines the data type within the message.
- EV:subtype. It contains the data subtype within the type category. Some examples may be: XLS, MP3, TXT, EXE, etc.
- VE:encrypt. Encryption type (if there is any).
- VE:user. User that will be required to connect to the destiny resource.
- VE:passw. Password that will be required to connect to the destiny resource.
- VE:publickey. Public key that will be used for the encryption (coding-decoding) of the communication (if there is any).
- VE:data. Data set to be communicated by the virtual entity. The content will depend on each frame type.

Example:

```
"VE:message": [
    {
      "VE:id": "9DBC2368DA1E-56A1-895F-143A56BC87DF",
      "VE:confirmreq": "1",
      "VE:timeout": "1000",
      "VE:fiability": "100%",
      "VE:type": "BIN",
      "VE:subtype": "B64",
      "VE:encrypt": "none",
      "VE:user": "any",
      "VE:passw": "any",
      "VE:publickey": "any",
      "VE:data": {
        "port": "COM4",
        "PARITY": "true",
        "velocity": "9600"      }
    }
]
```

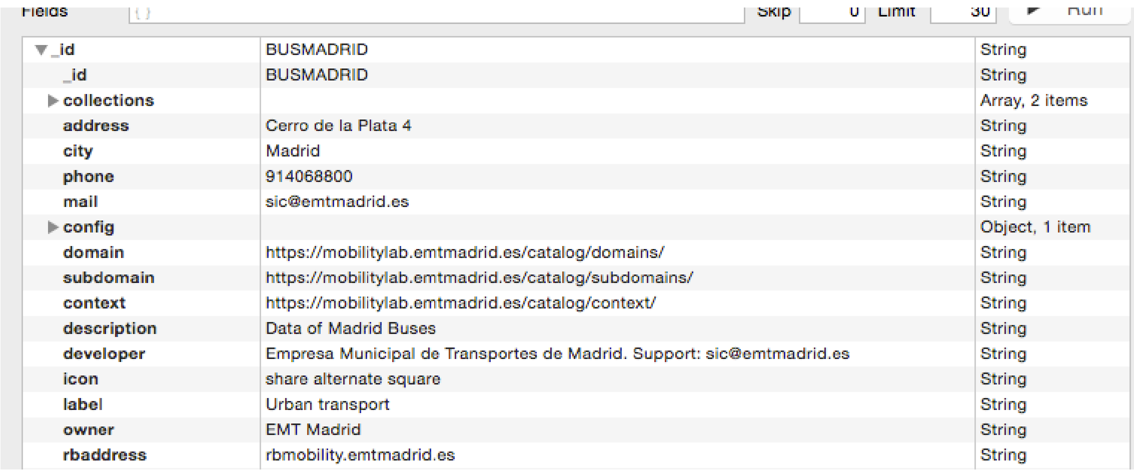
7.2.6. Changes done in Y2 within the VEProt datagram model

Since the implementation of the first Reactive Box system for the Madrid laboratory, certain problems were detected making incompatible the datagram protocol designed for interaction with the real software and storage elements. One of the first problems encountered was the notation "VE" of each element of the datagram produced a clear misunderstanding mistake within MongoDB schemes when using it in certain languages of instantiated classes, especially in C#. This makes that the released datagram version 1.00 which is currently working for the Y2 has changed the character ":" by "_". For example, the header object of the datagram is now represented as follows:



▶ _id	ObjectId("55fbf79f213ef421c8ba6e05")	Object id
▼ _id	ObjectId("55fbf9b4213ef42223c5e3be")	Object id
_id	ObjectId("55fbf9b4213ef42223c5e3be")	Object id
▶ vep_notification		Object, 3 items
▶ vep_source		Object, 6 items
▼ vep_header		Object, 9 items
▶ vep_geometry		Object, 2 items
_id	ed961f73-5dfa-11e5-9175-406c8f10d363	String
vep_owner	vepRoutesMadManager	String
vep_utcGeneration	2015-09-18 11:46:41.420539	String
▶ vep_iteration		Object, 1 item
vep_utcExpiration	2015-11-17 11:46:41.420548	String
vep_utcAsignation	2015-09-18 11:46:41.420558	String
vep_versionProtocol	1.00	String
vep_statusLocked	0	Integer
▶ vep_target		Object, 2 items
▶ vep_body		Object, 4 items

This has impacted directly on the notation defined originally for creating RDF definitions, since the original idea was that each element of the datagram was an element of a semantic definition. Finally, since it did not fit into the model, the ontological definition of the elements is currently being defined within the "_layers" collection of the storage database, reflecting the domain, subdomain and context of all elements of all collections.



▼ _id	BUSMADRID	String
_id	BUSMADRID	String
▶ collections		Array, 2 items
address	Cerro de la Plata 4	String
city	Madrid	String
phone	914068800	String
mail	sic@emtmadrid.es	String
▶ config		Object, 1 item
domain	https://mobilitylab.emtmadrid.es/catalog/domains/	String
subdomain	https://mobilitylab.emtmadrid.es/catalog/subdomains/	String
context	https://mobilitylab.emtmadrid.es/catalog/context/	String
description	Data of Madrid Buses	String
developer	Empresa Municipal de Transportes de Madrid. Support: sic@emtmadrid.es	String
icon	share alternate square	String
label	Urban transport	String
owner	EMT Madrid	String
rbaddress	rbmobility.emtmadrid.es	String

The aforementioned definition also applies to the "_schema" collection at each element level, though this change has meant delays in the semantic definition of objects.

▼ _id	ROUTESMAD.alarms	String
_id	ROUTESMAD.alarms	String
▶ __id		Object, 3 items
▼ instant		Object, 3 items
__description	No description	String
__type	date	String
__path	.instant	String
_description	no description	String
▼ _id	ROUTESMAD.bustrack	String
_id	ROUTESMAD.bustrack	String
▼ instant		Object, 3 items
__description	No description	String
__type	date	String
__path	.instant	String
▼ geometry		Object, 5 items
__description	No description	String
__type	object	String
__path	.geometry	String
▶ type		Object, 3 items
▶ coordinates		Object, 3 items
▶ __id		Object, 3 items
_description	no description	String

Another important change while operating VEProt model in real conditions has been the redefinition of certain elements to make it operational as a whole. The most important of those changes has been the "vep_body" object whose wealth of content now responds much better to the requirements model posed by COSMOS, even in its first conception, allowing each VE not only to reveal what it is or what it contains, but also letting those VE to contain segments of encrypted and public data or even snippets of executable programs transferred into the datagram itself, stating simultaneously the VE and building at the receiver an operable and intelligent self-image, if necessary. To do this, we have introduced a content analyzer that represents in an array which data type contains every element, supporting multiple message objects within a datagram. It also contains elements for building logs and trace in remote of the activity of the entire system.

▼ vep_body		Object, 4 items
▼ vep_type		Array, 3 items
▼ 0		Array, 1 item
0	layerData	String
▼ 1		Array, 1 item
0	ebusData	String
▼ 2		Array, 1 item
0	logData	String

It has also been included a scheme (vep_requisites) so that the receiver of the message analyzes whether the VE being received through the datagram contains elements compatible with its operative system, memory, capacity, etc. Added to this, it contains an object that defines implementation plans (vep_plan) for VEs manifested as processes running on client systems.

Finally, as mentioned above, the data object (vep_data) has been replaced by an array of elements that allows multiple instances, features, messages, values or actions traveling in the same datagram and can be managed as a set or following an order. A special but typical case is the property of merging two VEs, and applies especially when the person with special needs is on board the bus, as shown in the following example.

▼vep_data		Array, 3 items
▼0		Object, 9 items
layer	ROUTESMAD.usrtrack	String
instant	2015-08-10 14:53:02.462053	String
idRoute	55d1801ffb954f08f8f1489f	String
dayType	LA	String
▼geometry		Object, 2 items
type	Point	String
▼coordinates		Array, 2 items
0	-3.69138338267	String
1	40.4211505276	String
idSesion	21d1701fab914f08c8e1438e	String
nameRouteUser	jmendez	String
order	3	String
dayWeek	M	String
▼1		Object, 11 items
status	5	String
direction	1	String
bus	8811	String
stop	0	String
line	008	String
trip	14	String
name	####	String
▶geometry		Object, 2 items
altitude	626.372	String
delay	-346	String
offSet	570	String

7.2.7. Response indicators

In the area of response data the invoked subsystems will return the achieved result data either as a return message or as the required data frame, provided that a response has been requested.

The content is defined as indicated below:

- VE:returntosender. Unique defining label indicating that the object is the one informing about the result of the operation. The object contains the following values:
  - VE:errorcode: Return code about the result of the request.
  - VE:errordescription. Request result description.
  - VE:datetime. Date and time of the result of the request according to the invoked system
  - VE:response. It will be an added object to each message sent within the array
  - VE:response. Array with the content of the returned data
    - VE:id. Identifier of the individual frame of the message.
    - VE:data. Values of the message.

Example:


```
"VE:returntosender": {
      "VE:,errorcode": "000",
      "VE:errordescription": "INFO:decoded ok, datagrame allowing",
      "VE:datetime": "2014-04-20 13:23:05.06543 UTC",
      "VE:response": [
            { "VE:id":"9DBC2368DA1E-56A1-895F-143A56BC87DF",
              "VE:data": {
                    "valuelevel": "18",
                    "temperature": "5°"
                 }
            }
      ]
}
```


Regarding this section, there is an important remark regarding Y2 achievements, which is about the response indicators, as they merge as new elements of the array vep_body of the datagram.

8 COSMOS opportunities under the Smart Transport context

There are multiple circumstances and needs within a mobility oriented city system that depends on systems or subsystems whose development and implementation is beyond the specific context of transport or the establishment of protocols and standards for the exchange of information within a city. Among the main ones, we could find the security elements to ensure that certain content can be transferred between ends with the highest guarantee and confidentiality. On this regard, COSMOS could provide a secure way to encrypt a message segment that can travel through a public infrastructure and within an open protocol.

Information such as personal data, images, health data, etc. could use the Hardware Security Board developed under the framework of COSMOS as an encryption mechanism “extreme to extreme”. Those client systems requiring encryption of a VEProt datagram segment can perform the encryption operation and then be translated only by authorized recipients. It also lets information storage through Storlets within the Cloud with segments of encrypted messages and be recovered afterwards with total security.

Another important element for Smart Transport environment is the Complex Event Processing engine which allows you to manage and anticipate situations where a special event somewhere in the city can affect very significantly to the overall mobility or specific mobility of a citizen. Such events, by analysing patterns of previously processed data through the CEP can provide predictions that can help efficiency in traffic and, ultimately, to the overall mobility of the city.

Therefore, analysing the course and passing of buses, the density of traffic and traffic light phases provides useful data for planning special situations or prevent changes to a value of a pre-established series. Thus, connecting a predictive events analysis system to the transportation infrastructure provides added value for the management and planning of public transport and traffic.

For this, the mobility systems of the city must provide those events to the CEP, and through its analysis, getting automatically outlier data to take the appropriate decisions; and all in real time!

A clear example or use case is when the expected arrival time of a bus, which carries a person with special needs, should be analysed in a particular place and this bus does not arrive within its forecast; or undergoes a change on the bus route; or the person with special needs gets off the bus at a wrong bus stop; or does not realize that he/she has arrived to his/her destination, etc. all the aforementioned situations that could alert a supervisor or caregiver. To this end, transport events and contextual data need to be sent to big data storage systems such as those designed in the context of COSMOS. The purpose of Madrid, as a Smart Mobility City is to analyse how COSMOS, from the point of view of a permanent events analysing system can be used to solve complex problems of the city.

In summary, efficient communication of a uniform and open data flow, integrating connectors and processes within urban public platforms, interoperating with a system like COSMOS provides the ability to dynamically compose information flows for solutions adaptable to multiple situations.

9 Conclusions

During Y2, the initial conclusions outlined in Y1 deliverable have been confirmed. This is, that the final conclusion of the study leads us to believe that the integration of the various systems of public transport within the paradigm of IoT passes through the right combination of public infrastructure and private systems, within which the public component plays the following roles:

- Supplier of a unified event information exchange protocol.
- Provider of public service infrastructure, especially server architectures able to understand, integrate and connect different systems, information providers and customers under a unified architecture.
- Coordinating and managing body for the approval and of smart devices in order to ensure the adaptability of different mechanisms and virtual organizations in the city.

On the contrary, the private component (businesses, citizens, research centres), will use the public infrastructure used for the purposes that their model or task is required, either for collaborative purposes or for direct consumption of information.

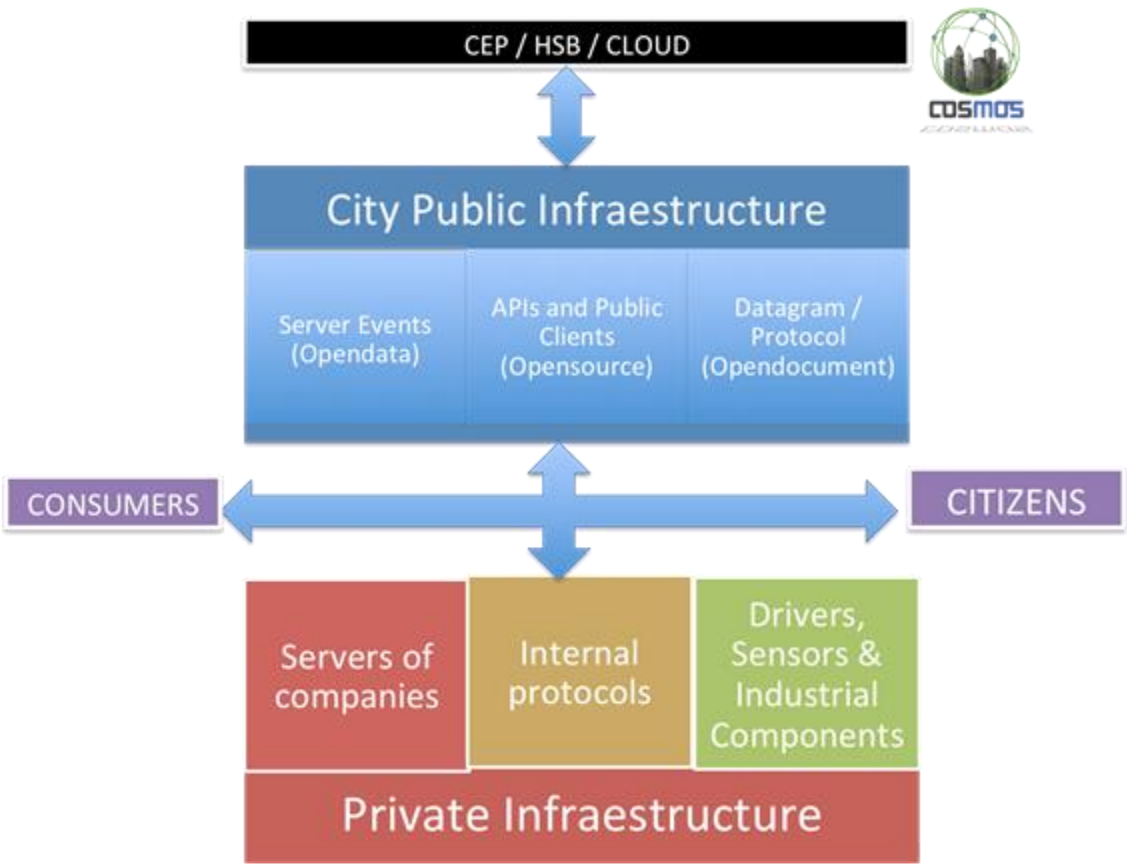


Figure 2: Public and private cooperation schema

10 Recommendations for Y3

Among the recommendations for Y2 some topics were highlighted to ensure the success of the project. For instance, the need of establishing a multi-discipline working group aimed at creating a model definition, communication and management of objects, systems and sensors for the city of Madrid. This working group has been working already during Y2, and has been responsible for the specification, development and implementation of a Smart City Mobility Lab in Madrid, working mainly in transportation topics but with an open scope.

The main aim was to constitute a group wide enough to allow the participation of the four major groups of actors potentially involved in the development of the VEProt system. This group has been the project management committee and the "seed" for the implementation of the lab infrastructure.

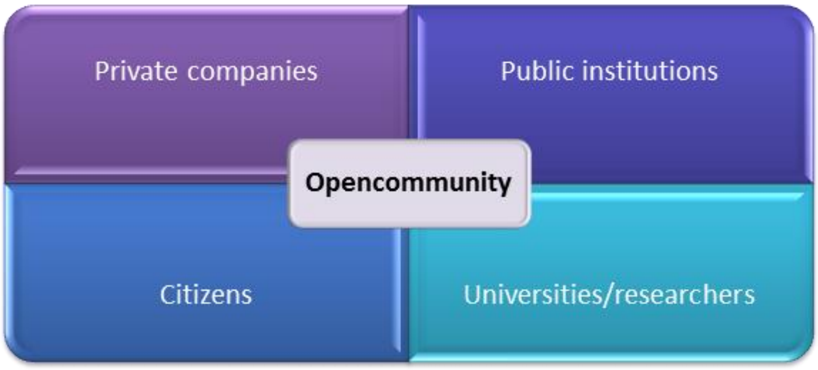


Figure 3: Proposed local working group for Y2

As stated in the previous deliverable D7.5.1 (Y1), the aforementioned community has been structured as follows:

- Project Management Committee: management function will be at a high level throughout the life cycle of the project and presentation, distribution and publication of the progress and results achieved over time.
- Audit and Oversight Committee: their role will be to establish the guarantees to ensure that the tasks that are planned are carried out with maximum quality as well as to verify that the repositories of code and documentation and new versions of VEProt have the highest reliability and warranty. This committee will also analyse the bug and the feedback from any source, establishing mechanisms for the evolution of the system.
- Coordination of code development: It shall arrange the analysis and specification of the source code to develop. It will also have the function to document and upload to the public repository the code developed.
- Developers: Those responsible for conducting the development of source code, using public repositories (GitHub, etc.)
- Specification-ontology and semantic entities: is responsible for defining the data model of virtual organizations in the city, as well as to maintain the information and the definition of the entities in public repositories. This function will be performed by specialists.

- Repositories, diagrams, web-semantics: the schemes will be used under any standardized specification for defining objects in public and accessible repositories.
- Protocol specification data: VEProt requires a standardized data exchange, to be defined, nurtured and developed by specialists and updated in a public repository.

More specifically, regarding recommendations for Y3, we would like to point out that, within the broad and complex scenario that arises in this document, COSMOS offers endless possibilities within the scope of IoT in the city of Madrid, and not only in the area of mobility and transport but also in other areas, by conjugating the reactive elements deployed, which require expert systems in Machine Learning and CEP to operate.

Therefore, we believe that the system could be ripe to raise any use case that combine data sources and combined problems in two or more aspects or topics of the city. This could be the case of analyzing the conjunction public transport/lighting, or private transport/traffic optimization and improvement of congestion, etc. These ideas, based on the successful implementation of Madrid use case could be considered as alternatives.

Additionally, the system should also follow the line of technical dissemination of the laboratory and the empowerment of its open and public use. On this regard, building an information portal to let proper dissemination and learning of Madrid reactive elements should be considered, using at the same time the research opportunities that COSMOS offers both within private and public solutions. The portal must contain various elements that allow both rapid deployment and content accessibility, such as GIT, CMS content (wordpress, etc), Wiki, etc.

Moreover, it may be useful to have specific user interfaces for startups and researchers to build new reactive elements and investigate with them, using COSMOS system as the generator of complex solutions.

11 Bibliography

1. RDF Dataset Normalization: A Standard RDF Dataset Normalization Algorithm;http://json-ld.org/spec/latest/rdf-dataset-normalization/

2. Organization of Schemas; http://schema.org/docs/schemas.html

3. Semantic Web Development Tools; http://www.w3.org/wiki/SemanticWebTools

4. Node.js Api Reference Documentation; http://nodejs.org/documentation

5. DDP, the Distributed Data Protocol; https://www.meteor.com/ddp, http://www.meteorpedia.com/read/DDP_Clients, https://github.com/foxdog-studios/pyddp

6. UNE 178301:2015, Ciudades Inteligentes. Datos Abiertos (Open Data); http://www.aenor.es/aenor/normas/ctn/fichactn.asp?codigonorm=AEN/CTN%20178#.VOBIO1OG_7p

7. Advanced Message Queuing Protocol; http://www.rabbitmq.com

8. Implantación de la Sociedad de la Información, el Conocimiento y la Innovación en la Ciudad de Madrid. Boletín Oficial de la Comunidad de Madrid Num. 6661; http://www.madrid.es/portales/munimadrid/es/Inicio/Ayuntamiento/Boletin-Oficial-del-Ayuntamiento/Buscador-Boletines/862-2012-04-18-Dec-Alcsa-Comision-Tec-Impl-Soc-de-la-Informac-el-conoc-y-la-Innov-?vgnextfmt=default&vgnextoid=9c965e7751ed6310VgnVCM2000000c205a0aRCRD&vgnextchannel=7a698db0ae967010VgnVCM1000009b25680aRCRD

9. Semantic Sensor Network definition and specifications http://www.w3.org/2005/Incubator/ssn/wiki/Main_Page

10. MongoDB https://docs.mongodb.org/manual/