

## Article

# Design of a Cooperative ITS Architecture Based on Distributed RSUs<sup>†</sup>

Asier Moreno<sup>1,2</sup>, Eneko Osaba<sup>1,2</sup>, Enrique Onieva<sup>1,2</sup>, Asier Perallos<sup>1,2,\*</sup>, Giovanni Iovino<sup>3</sup> and Pablo Fernández<sup>1,2</sup>

<sup>1</sup> DeustoTech – Fundación Deusto, Avda. Universidades, 24, 48007 Bilbao, Spain

<sup>2</sup> Faculty of Engineering, University of Deusto, Avda. Universidades, 24, 48007, Bilbao, Spain; asier.moreno@deusto.es (A.M.); e.osaba@deusto.es (E.O.); enrique.onieva@deusto.es (E.O.); pablo.fernandez@deusto.es (P.F.)

<sup>3</sup> INTECS SpA, Via Umberto Forti 5, 56121 Pisa, Italy; giovanni.iovino@intecs.it

\* Correspondence: perallos@deusto.es; Tel.: +34-944-139-073

<sup>†</sup> This work is an extension of the paper “Cooperative decision making ITS architecture based on distributed RSUs”, presented at the 9th International Conference on Ubiquitous Computing & Ambient Intelligence (UCAmI 2015), Puerto Varas, Chile, 1–4 December 2015.

Academic Editors: Vladimir Villarreal and Carmelo R. García

Received: 29 April 2016; Accepted: 19 July 2016; Published: 22 July 2016

**Abstract:** This paper describes a new cooperative Intelligent Transportation System architecture that aims to enable collaborative sensing services. The main goal of this architecture is to improve transportation efficiency and performance. The system, which has been proven within the participation in the ICSI (Intelligent Cooperative Sensing for Improved traffic efficiency) European project, encompasses the entire process of capture and management of available road data. For this purpose, it applies a combination of cooperative services and methods for data sensing, acquisition, processing and communication amongst road users, vehicles, infrastructures and related stakeholders. Additionally, the advantages of using the proposed system are exposed. The most important of these advantages is the use of a distributed architecture, moving the system intelligence from the control centre to the peripheral devices. The global architecture of the system is presented, as well as the software design and the interaction between its main components. Finally, the process followed with the aim of testing the architecture functionality is described. This experimentation has been carried out in two real scenarios, in Lisbon (Portugal) and Pisa (Italy).

**Keywords:** Intelligent Transportation Systems; Wireless Sensor Networks; distributed intelligence; wireless communications; Ubiquitous computing

## 1. Introduction

The aim of this work is to define an innovative and fully distributed architecture to enable cooperative sensing in Intelligent Transportation Systems (ITS) environments. The feasibility of the proposed approach has been proven within the participation in the ICSI European project [1], developing a reference end-to-end implementation targeted to both urban and highway scenarios.

The system encompasses the entire process of capture and management of available road data, enabling the generation of services to promote transportation efficiency. The main idea behind the project relies on a local distributed intelligence, operating on a limited geographical scale, where data are timely distributed and processed without the need to contact a central subsystem. For this purpose, the concept of “gateway” (GW) is introduced. Briefly described, a GW is a logical entity that offers capabilities similar to those provided by centralized approaches (e.g., data storage or event processing). The main advantage of the GWs is that they operate on a local scope with the possibility of exchanging messages with other GWs.

This document provides a description of the overall system, defining the high level architecture and the role of its different elements. The way the intelligence and the sensed-data are provided and the concepts behind the design choices are explained.

Moreover, a set of test services developed to validate the system and based on real use cases are described. Besides that, the results of the field trials are shown. These trials have been performed in the cities of Lisbon (Portugal) and Pisa (Italy), with the aim of testing the benefits of the architecture.

In classical transport systems, open loop control governs the interaction between the user/driver and the control centre. In recent years, these systems have been evolving towards ITS [2], where a closed loop interaction between users/drivers and the transportation infrastructure can be found, enabled by cooperative V2X communications and cellular networks [3]. While some of the enabling technologies are entering their mature phase, there is still the need of a complete integrated solution that can take the most benefits from a real-time analysis of the data gathered, and an appropriate reaction on the transportation system.

Nowadays, a centralized architecture for traffic management systems is commonly used. In this case, all the information is gathered by a Traffic Management Centre (TMC) from motorways authorities, weather stations, regional data collection and measurement points on the road. Then, the information is analysed, and decisions are taken in the TMC, sending back to the on-road infrastructure signals. Thus, the architecture of classical ITS systems is purely hierarchical, with sensed data flowing from the leaves (i.e., road-side or vehicle-installed sensors) to the root (i.e., the traffic management centre).

Usually, this kind of approach presents some disadvantages [4]:

- It does not scale adequately with the inclusion of a significant number of new elements and the increase in the amount of data collected, showing a lack of flexibility and making difficult its maintenance.
- This vertical architecture is not suitable to accept and integrate changes in ITS, like new components produced in research centres and in the industry.
- It exhibits latency and security issues related to the centralization of the communications. This latency produces a bottleneck in the system.

For these reasons, research activities in ITS, and especially in C-ITS (Cooperative ITS), have changed the vision behind the definition of new ITS architectures [5], switching from the hierarchical and vertical approach to a new vision, which is more horizontal and distributed [6].

In recent years, important research projects, like CVIS [7], adopted and developed this new strategy to achieve cooperativeness in ITS: vehicles, road-side infra-structure, central systems and personal devices are no longer linked to each other through a static and vertical relationship. Instead, they are seen as “nodes” belonging to a common network. Similar projects like Safespot [8] and COMeSafety<sup>2</sup> [9] use this approach to increase road safety allowing direct communication between vehicles (C2C). The DRIVE C2X European project [10] is another example of recent implementations of C-ITS architectures, establishing a common reference system for C2X communications and performing successful large-scale field trials. With these new approaches, traffic and travel models and ITS applications specialized to operate on a local scale can be configured in order to fit the needs of the surrounding areas. In addition, they can cooperate exploiting the distribution platform.

The main contribution of the work presented in this paper is not only the design of a new C-ITS architecture, but its practical application and the use of distributed data to make collaborative decisions based on learning capabilities. A set of test services developed to validate the system and based on real use cases are described and used to generate two different real world scenarios. The test results regarding the execution of the use cases in the trial scenarios will be thoroughly described in order to understand the viability of the system.

The remainder of this paper is structured as follows. In the following section (Section 2), the proposed architecture is presented. In Section 3, the main components of the system are deeply

described. In Section 4, a description of the test applications and services is made. Additionally, the experimentation carried out is described in Section 5. Finally, conclusions and future work are explained in Section 6.

## 2. Cooperative Decision-Making ITS Architecture

A new cooperative architecture is proposed in this paper. In this architecture, the intelligence of the system is distributed among some of the elements in the infrastructure, which host a software platform for running ITS applications. In line with this, communication with the remote centres happens only for the transmission of aggregated data for long-term operations (e.g., data mining, software upgrades, and logging). On the other hand, real-time data will be processed and stored locally in the system infrastructure, nearby the source of the events.

The developed architecture relies on a local distributed storage and intelligence, which operates on a limited geographical scale without the need to contact the control centres. Moreover, sensed data are processed in a cooperative manner performing content aggregation and integration since the earliest stages. On the other hand, only statistical data for long-term evaluation are sent to the TMC. Two key concepts are defined in order to achieve distributed intelligence and cooperative sensing: the GW and the local/global areas:

- **Gateway (GW).** A GW refers to a physical entity, usually installed inside a Road Side Unit (RSU), which implements the reference architecture, the subsystems Connectors, the Data Distribution Platform (DDP) and the Collaborative Learning Unit (CLU). The functionality of these components will be detailed in Sections 3.1–3.3, respectively. A GW is able to join Local/Global Areas, and it is connected to different subsystems (VS: Vehicular Subsystems, PS: Personal Subsystems, RS: Road-side Subsystems, etc.)
- **Local/Global Area.** An area is composed of a set of GWs (at least one), communications among these GWs (when more than one is present), and some criteria to define the area perimeter (e.g., based on the density of population, traffic, ICT elements, etc.).

A logical view of the architecture, which shows the previously introduced concepts of GW and local/global area, is reported in Figure 1. All of the Local Areas are logically connected to a global one supporting inter-area communication, while physical communication happens through the backhaul technologies and the Internet.

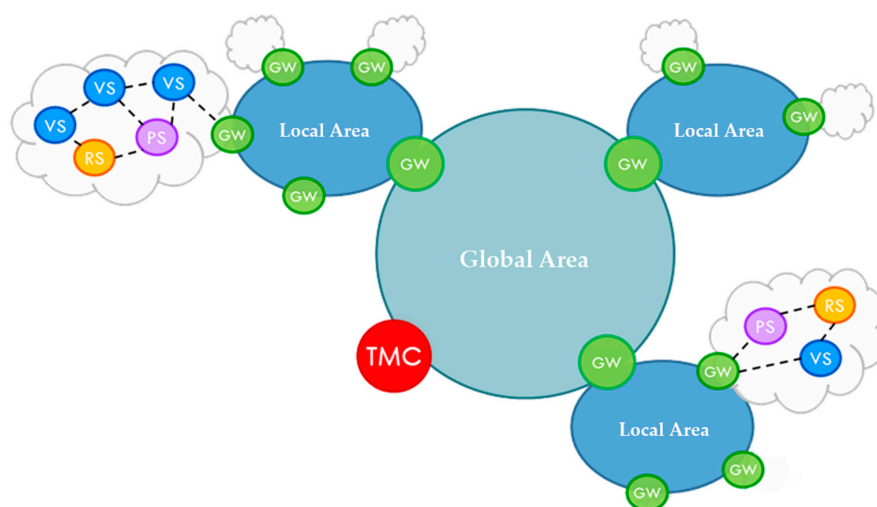


Figure 1. Global system architecture.

Local storage is offered by each GW of the Local Areas, allowing decisions based also on previous historical data. However, since the applications for real-time processing are mostly concerned about

the recent past, only recently sensed data will be kept in the GWs. On the other hand, long-term measurements and storage based on aggregated data for management purposes will be done by the TMC only.

### 3. Software Architecture of Platform on GWs

Based on the generated distributed system, the GWs will analyse the gathered information. All this analysis is made autonomously and without requiring online TMC dependences or human participation. They also determine the best traffic strategies for dealing with roadway incidents. In this way, the distributed system enhances the scalability and overcomes the weakness of centralized approaches. For this purpose, each of the GWs hosts a software platform designed for running ITS applications. This software architecture (Figure 2) is composed of various modules classified in logical layers:

- **Connectors.** They are used to extend the GW interoperability via integration interfaces with external subsystems and technologies such as Wireless Sensor Networks, Vehicular Networks, Road-side subsystems, etc. The functionality of the Connectors and the communication between the subsystems and the proposed architecture will be further described in Section 3.1.
- **Data Distribution Platform (DDP)/Logical Layer.** The Logical Layer provides the basic capabilities needed by the ITS services and the CLU in order to interact with the physical layer and the data available. The key component of the logical layer, the DDP, includes the mechanism to communicate between applications and services by managing an events-based architecture. The functionality of the DDP will be further described in Section 3.2.
- **Collaborative Learning Unit (CLU).** Along with the DDP, CLUs are other key software elements of the proposed architecture, which are responsible for providing intelligence to the whole system. These units are able to learn from each situation in order to provide different actions plans in real time. The functionality of the CLUs will be further described in Section 3.3.
- **ITS Services and Applications.** Higher level services or end-user applications implementing traffic and travel models to respond to roadway incidents. An example of the services that can be provided will be shown in Section 4.
- **External Components.** Software components running in the same OS, and are required by other components in order to work properly.

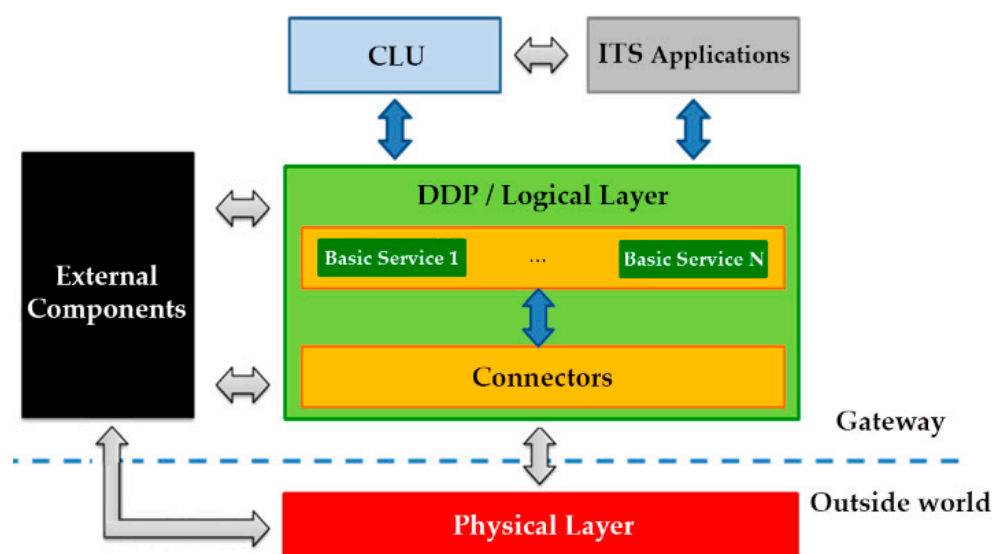


Figure 2. Software Architecture of platform on GWs.

Regarding the runtime environment, the architecture will be executed on a GNU/Linux operating system. The choice of GNU/Linux has been made due to its open-source nature, and due to the success of this OS, which is flexible enough to run on a wide set of hardware.

All of the services, connectors and applications will be provided as OSGi [11] bundles exposing their functionalities to other bundles through the OSGi service layer. Thus, an OSGi reference framework and a JAVA Runtime Environment are required in order to be able to run the GWs software platform.

### 3.1. Connectors and External Subsystems

The GW's Connectors are responsible for the effective integration of external subsystems into the proposed platform. Each external subsystem providing or receiving data from the GWs needs to be integrated through the development of a specific connector bounded on one side to the DDP, while on the other side providing APIs towards the subsystem components. Thus, each Connector is internally composed of four modules, which are in charge of the following: communication towards the DDP, communication towards the subsystem, configuration, and data model mapping between the subsystem and the DDP.

To enable the above-described mapping functionalities, the Connectors need to implement standard APIs provided by DDP basic services, mainly devoted at enabling a bidirectional communication towards the subsystem behind the connector. Such APIs can be used, for instance, to configure the subsystem components, or to provide feedbacks to devices belonging to the subsystem. They can also use external components that are executed on the same host. External components can be applications, services, bundles or other software elements which have not been designed within the architecture but are required by one or more ITS services in order to properly operate.

The use of Connectors in the GW's architecture is a powerful solution able to adapt heterogeneous subsystems into the DDP data representation and communications model. Moreover, the use of properly defined Connectors allows the integration of legacy ITS systems in the platform. This fact guaranties the reusability of already developed technologies.

### 3.2. DDP: Data Distribution Platform

The role of the DDP is to enable a scalable and highly adaptable system able to receive data from the architecture subsystems, and to provide a set of capabilities (basic services) used by connectors, ITS applications and other high level services (e.g., the logic behind the CLU).

The DDP has been mainly designed as a set of components capable of interacting among them. Each component can provide services through defined APIs or can use services provided by other components. ITS applications can serve both as applications or high level services extending the capabilities of the DDP's basic services. Another use of ITS applications is to display processed information to the end-user. These applications exploit the capabilities provided by the DDP's basic services and can directly communicate with the connectors if a direct interaction with a particular subsystem is needed.

The GW's reference architecture presented in Figure 2 shows how different architectural components of the DDP interact. The darker interfaces (arrows) in the figure will be defined and implemented by the proposed platform. These will be used by the higher level applications/services and the connectors in order to communicate with the logical layer facilities. The high level interfaces, coloured in light grey, are instead already existing interfaces defined outside the project realm necessary to integrate particular functionalities required by the DDP. Examples of this kind of interfaces are: interfaces required to access external components, interfaces provided by higher level ITS services or applications, and interfaces required to communicate with the external world. The specific architecture connectors use both internal and external interfaces in order to make a bridge between already developed ITS systems and the architecture, as well as among different GW hosts.

The DDP is, therefore, responsible of providing the mechanisms for the communication between the different layers using a publisher/subscriber (P/S) events-based architecture. In this way, other software components would only have to be concerned to inform the DDP about the events in which they are interested. Instead of implementing a high number of point-to-point links between sensors, actuators and GWs at all layers, the DDP will implement a kind of data space, using an appropriate middleware [12], directly supporting many-to-many communications.

The facilities provided by the DDP combined with the logic of services and applications will enable a proper elaboration of all the available information. This fact makes possible the realization of cooperative sensing and ITS, as it is presented in Section 5.

### 3.3. CLU: Collaborative Learning Unit

As stated previously, the CLU, as a high level ITS service, is responsible for providing intelligence to the system, executing action plans in real time according to the events received and the current state of the infrastructure. To achieve this autonomous operation, CLUs will include a set of components that will respond to the events received via the DDP (Figure 3).

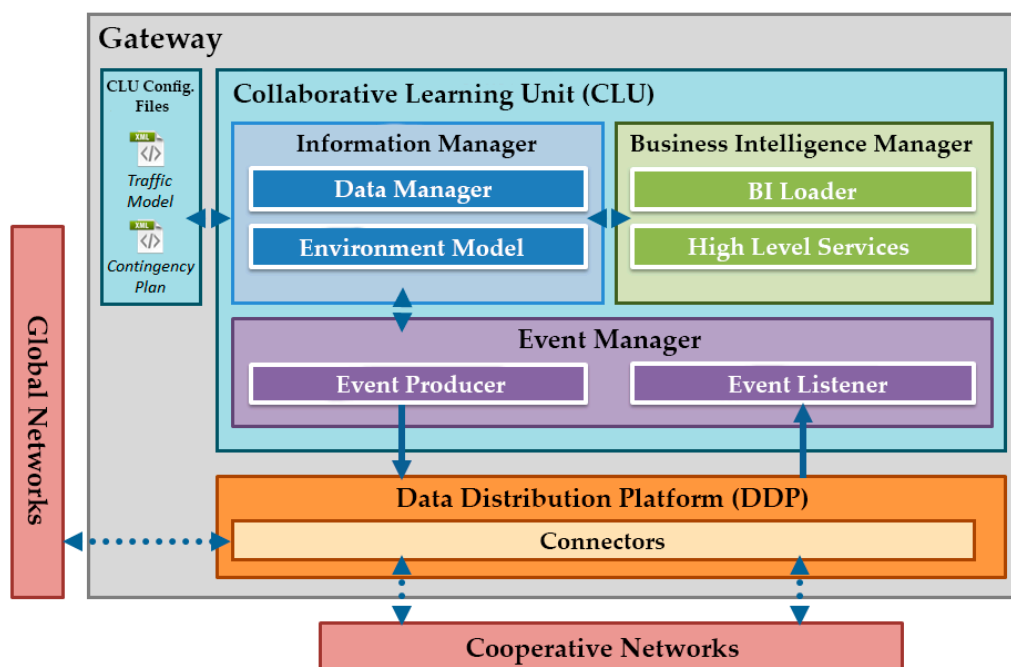


Figure 3. Architecture and technical design of the CLU.

The distributed traffic management will be achieved through the collaboration of the CLUs in the area enabling cooperative intelligence. The output will also feed up the cooperative networks, resulting in notifications that influence the reality of interest.

Additionally, the CLU itself is composed of three main elements:

- **Events manager:** this module controls the flow of the events at the CLU side; communicating with the DDP to obtain relevant events for the CLU considering its geographical position, the GW's attached wireless sensor network (WSN), the position of other GWs, etc.
- **Information manager:** this module provides updated information about the GW environment (related sensors, road conditions, etc.) to the CLU services. It also processes the traffic models and the contingency plans configuration files. These two files, updated periodically by the TMC, allow the behaviour change of the CLUs.



- **Business intelligence manager:** this component has a set of high-level services responsible for responding to incidents that arise on the road. These incidents can be, among others, accidents, delays or high CO<sub>2</sub> levels. In line with this, services implemented by the CLU will have different behaviours, because of the variety of the situations. These behaviours will be dynamically adjusted based on the learning capabilities provided by the artificial intelligence integrated in the CLUs.

### 3.3.1. CLU Data Model

CLUs will be always updated with new traffic models and contingency plans reported by the TMC. Moreover, using information reported by the system, a decision support system in the TMC will help the traffic manager to update and improve traffic models and contingency plans.

The traffic models consist of two data sets, which may be updated allowing the change in the CLUs behaviour. On the one hand, topological information (position) of the elements within the Local Area of the CLU, as well as the WSN assigned to the same GW, is established. On the other hand, it should also store a table with the set of rules that clearly specify what information is relevant to that particular CLU. In this way, it can properly subscribe to events provided from the DDP. These rule tables may be updated in real time so that the behaviour of the CLU can be adjusted dynamically.

### 3.3.2. CLU Intelligence

One of the main challenges of this architecture is to develop stable and distributed algorithms based on probabilistic reasoning, and not requiring very high computational resources. As the situations that may occur are rich and varied, services implemented by the CLU will have different behaviours. These behaviours are dynamically adjusted based on the learning capabilities provided by artificial intelligence integrated in the CLUs. In this way, CLUs have to implement some techniques to solve the different problems that could appear. These problems and techniques have been categorized in Table 1 according with the role of the CLU and the kind of problem.

**Table 1.** Different Problematic being faced by CLU.

Category	Problems Faced	Techniques
Routing	- Alternative routes	- Genetic Algorithms
	- Alternative transport	- Fuzzy Decision Support Systems
	- Route guidance	- Probabilistic Models
Regulation	- Ramp-metering algorithms	- Fuzzy Control
	- Access to pollution-monitored area	- Hybrid Models
		- Particle Swarm Optimization
Prediction	- Level of congestion	- Probabilistic Models
	- Travel time for vehicles	- Bio-inspired optimization
	- Pollution	- Time-series prediction
Monitoring	- State of traffic	- Particle filters
	- Free parking slots	- I2V Communications
	- Incidents	

In the category involving routing and regulation cases, any problematic that requires the CLU to take an action or decision is included. With respect to the problems requiring routing decisions, the CLU has to solve routing problems as a necessary task for the implementation of several use cases (presented in Section 4.1). This is because of the fact that almost in every incident detected by the CLU

it has to provide a warning signal and (when possible) an alternative route to avoid the inconvenience to the rest of vehicles. The routing problems can have three different natures:

1. **Alternative routes:** this problematic is referred to the classical routing done to face demands received by the CLU. In this category routes given to regular vehicles can be found.
2. **Alternative transport:** these routes involve the parking of the vehicle and the taking of an alternative (public) transportation.
3. **Route guidance and emergency support:** routing made for special vehicles are included here. In this case, special vehicles (emergency and authorities) that must be routed with care have to be considered. It is important to note that these vehicles are allowed to enter any restricted area.

The prediction problems contain all problems and techniques that estimate the probability of some event to appear, like congestion and high pollution levels; or the possible values of some problems like the travel time for vehicles. Many of the problems inside this category need additional information. In order to provide predictions in a distributed way, information regarding sensors located in the area of interest gathered by a CLU is processed by data mining techniques. With the processing of the collected data, the internal model of the future state of the variable of interest.

In order to make models readable, as well as easy to code and share among CLUs, tree models are used [13]. State of the art decision tree algorithms can be launched both in a local or remote way. Resulting models (tree or set of rules) can be easily shared within all the connected nodes. Some examples of this sort of algorithms can be the well-known C4.5 [14], the ID3 [15] or the Support Vector Machine [16]. This kind of techniques have been successfully applied to a broad range of tasks, from learning to diagnose medical cases [17,18] to school performance prediction [19]. These models are also needed to predict the possible future levels of CO<sub>2</sub>. This fact could be useful for control the restricted areas into the urban use cases of monitoring and reduction of air pollution.

The fourth group of problems is the monitoring group. The monitoring actions taken by the CLU will be oriented to send/receive information about the state of the environment. The collected information will be necessary for a proper decision making to be done by the CLU. Monitoring techniques record the accidents, stopped vehicles on roads, blocked lanes and any other event that could happen in the vehicular flow. It also monitors the slots of all intermodal parking, which is very important to decide correct alternative routes and alternative transports.

#### 4. ITS Test Applications and Services

The proposed ITS architecture must be able to prevent and warn about many anomalous situations that can occur in the circulation, as well as to act in consequence. To make this possible, it is necessary to analyse in detail all the possible events, and to select the most effective method or algorithm to solve each use case. The implementation of a set of ITS services running on the proposed architecture has been established in order to validate the correct operation of the whole system.

##### 4.1. Use Case Analysis

To perform the tests, a working platform to simulate the necessary information has been created. This platform allows the behaviour analysis of the vehicles before, during and after the incidents. Table 2 shows the different use cases analysed with this platform.

**Table 2.** Use cases analysed to test the proposed architecture.

Identifier	Type	Description
UC1	Urban	Monitoring and reduction of air pollution
UC2	Urban	Cooperative parking lots monitoring
UC3	Highway	Alternative paths signalling
UC4	Highway	Monitoring of anomaly in traffic flows
UC5	Highway	Traffic jam warning



SUMO (Simulation of Urban Mobility) [20], an open source tool for microscopic road traffic simulation, has been used to simulate the behaviour of the vehicles. The solution for urban use cases will be implemented in the city of Pisa, Italy. For this case, Pietrasantina and Aurelia streets have been selected. This selection has been made due to the existence of alternative paths on major roads, as well as an intermodal parking. Figure 4 shows the map used for the urban simulations.



**Figure 4.** Map in SUMO used for the simulation: (a) Pisa city centre. In blue the selected area for the urban use cases simulation; (b) Zoom over the selected area, with a 350 slots parking facility; (c) Simplified model for the simulations, with the main road connections and the parking facility.

During the simulation, the system monitors the CO<sub>2</sub> levels of the restricted area. When the system detects that these levels continue to rise, being probable that the maximum levels can be exceeded, the restricted area is closed. Once the restricted area is closed, only authorized vehicles are allowed to access it, while the rest of the vehicles must be re-routed. In addition, the Pietrasantina's Parking has been defined in a way that the number of free slots is available for simulation purposes. This parking has more than 350 slots. With the aim of simulating intermodal applications, the bus station has been incorporated, carrying drivers from/to the Pisa city centre.

On the other hand, to perform the simulations of the highway use cases, a 10 km section of the A5 highway of Portugal has been used. In this use case, the system continuously receives the events that happen on the road, such as accidents on the road or stopped vehicles. Once an incident is reported to the system, it initiates a process to prevent further damage, and it warns other vehicles as soon as possible. Additionally, for the traffic jam warning use case, the simulator receives the flow of the highway in real time. In this way, when a traffic jam is detected, the system starts the process of alerting nearby vehicles to prevent them entering the jam. In addition, vehicles with the possibility of using an alternative route are informed about the incidence, and provided with the possibility of using an alternative route.

#### 4.2. Test Environment

Along this subsection, it will be exposed how the architecture functionality and capabilities were tested. Before the start of the planned field trials, which will be performed in Pisa (Italy) and Lisbon (Portugal), two datasets were created to test the CLU implementation and the operation of the GW. These datasets correspond to the urban and highway scenarios, defining the value of the sensors and the events generated by them for a certain period of time. The objective is to see not only the flow of events generated by the use cases, but rather the communications between CLUs as well:

- Collaborating and communicating with other components of the architecture, with the aim of elaborating environment information in a distributed way.
- Enabling cooperative intelligence, and learning from each situation in order to provide different actions plans depending on the events received from the reality of interest.

Figure 5 shows the global architecture of the designed test environment, which is composed of two main elements: the simulator bundle, an OSGi module integrated in the architecture, which recreates the ingoing events, and the demonstration web service and application, which interacts with the system via the simulation bundle to show real-time values of the sensors.

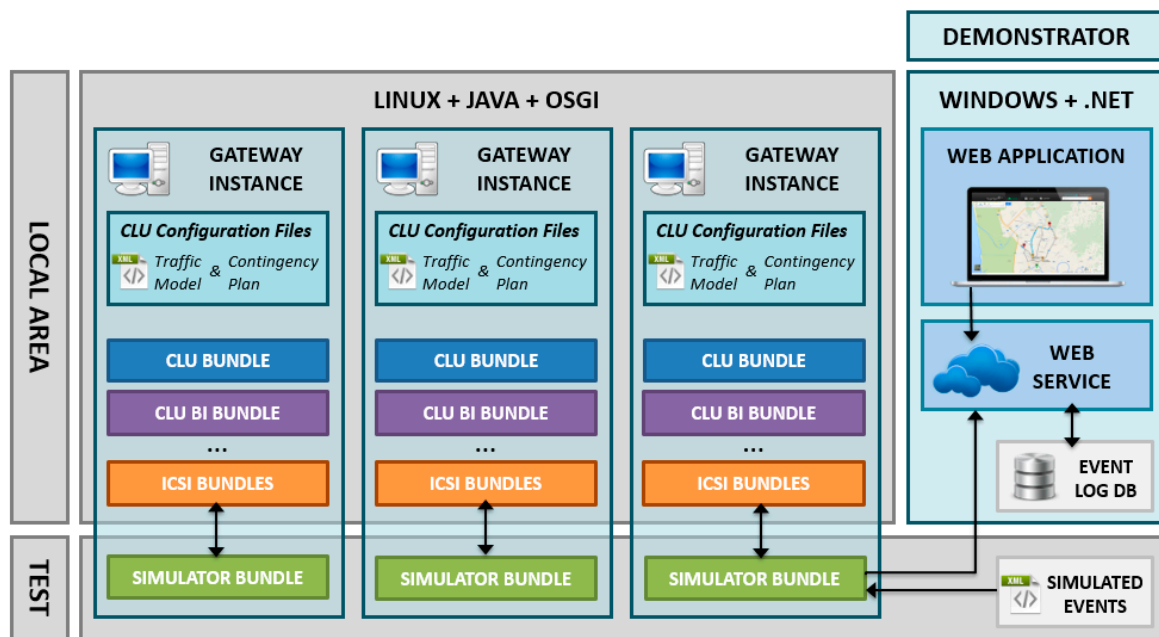


Figure 5. Architecture Test Environment.

#### 4.2.1. Simulator Bundle

As can be seen in Figure 5, three GW instances are deployed. These instances are complete SW architecture implementations running in a GNU/Linux machine in three different processes. These processes communicate with each other via event messages. The CLU OSGi set of bundles and the needed configuration files are also included and running on each of the instances, so a complete Local Area is being simulated in the provided test environment.

The CLU Simulator bundle is also included in the instances for testing purposes. It will read the temporally sequenced list of sensor values from the provided use case simulation datasets. Then, using the Event Processing Language (EPL), it will generate the corresponding Data Model to publish the simulated event inside the GW's Event Manager, just as if it were a real event sent by an external sensor attached to the GW.

The functionality of this bundle is required to adequately test the reception of event messages from the cooperative networks. In the following integration tests (detailed in Section 5) this bundle will be replaced seamlessly with real information coming from the ICSI infrastructure.

#### 4.2.2. Demonstration Web Application

A web application for the real time monitoring of the sensors and actuators has been incorporated to the testing solution. It is possible to monitor in real time the vehicles position, the monitored events (e.g., traffic jam, accidents, etc.) and the messages generated by CLUs on each GW according to the contingency plans.

For vehicles position tracking the positioning information carried by CAM messages generated by On Board Units (OBUs) is used. Each vehicle that is equipped with an OBU can be tracked. This track is possible thanks to the DDP event processor service, which is used to process the CAM messages. These messages are received by the GWs in the RSUs. All of the events generated on a particular GW are also propagated to the other GWs located in the same Local Area, including the one running the web application, in charge of displaying visual information for the monitored area.

To test some of the use cases, the web application offers the capability of generating some of the events that should be triggered by the traffic management centre. In this case, the injected events are also forwarded to the other GWs in the Local Area, where the DDP and the CLU can process them and react properly. Then, the active events in the system can be monitored on the platform, where it is possible to cancel them if necessary (e.g., an accident event can be cancelled by the web application after the accident is clear and the road condition has been restored). With this application (Figure 6), the administrator of the system, or the responsible for carrying out the final deployment, can test the functionality and the correct operation of the system in a simulated environment prior to the physical deployment of the RSUs on the road.

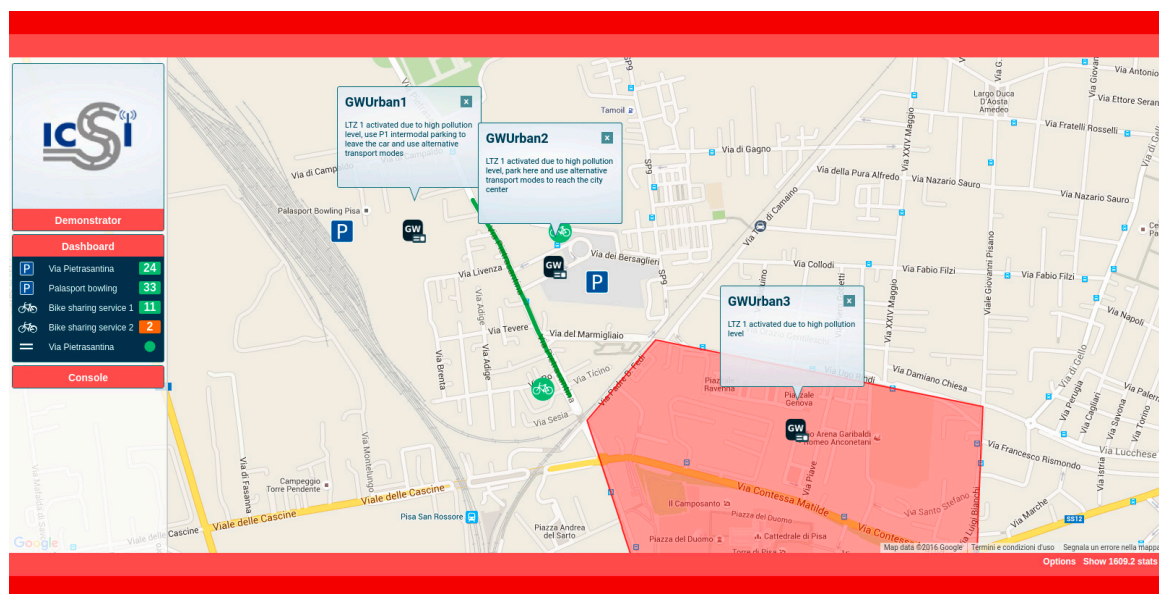


Figure 6. Demonstration Web Application.

The web application is a Single Page Application developed in C# with a map-based interface based on Google Maps API. The application is a RIA (Rich Internet Application) [21], and follows the MVC architecture. For the presentation layer design, HTML5, CSS3 and client programming languages like JavaScript, JQuery and Ajax have been used. Figure 6 shows a screenshot of the developed demonstration web application.

## 5. Field Experimentation

This section presents a summary of the results and the process followed with the aim of testing the architecture functionality using real data coming from predefined ICSI scenarios. As has been mentioned in previous sections, two scenarios have been selected for this experimentation, corresponding to the two field trials scheduled in Pisa (Italy) and Lisbon (Portugal).

For the first test scenario, a location in the access point of the Pisa city centre has been selected. Real data about historical pollution levels in the city has been incorporated to the experimentation. On the other hand, in Lisbon, the system was tested along the A5 highway, and the trials were executed in order to assess the performance of the communications, namely in terms of the IT2S communication.

Real data about the vehicles traffic flow in the A5 highway has been incorporated in order to test the CLU in a context as close to the real one as possible.

The main goal of the experimentation is to show that the system, by the aggregation of distributed sensors data and the implementation of collaborative intelligence, can provide relevant information related to the analysed use cases to the users, and therefore improve their decision making while using their vehicles.

### 5.1. Urban Scenario in Pisa, Italy

Pisa is a city in Tuscany, central Italy, on the right bank of the mouth of the River Arno on the Tyrrhenian Sea. It is the capital city of the Province of Pisa. In this city, some areas have a controlled access through Restricted Traffic Zones (RTZs) and Low Emission Zones (LEZ). These LEZs are a way to reduce the pressure of non-residential traffic in highly touristic destinations. The objective of LEZs is to control the pollution level in highly congested and populated zones.

In this context, and having the collected data about the traffic flow and the parking availability, the proposed urban test scenario includes the implementation of the following use cases:

- Alternative transport services;
- Monitoring and reduction of air pollution;
- Alternative paths signalling/route guidance; and
- Cooperative parking slots monitoring.

The system constantly monitors the pollution of the roads in the RTZ and LEZs of Pisa. As has been explained before, when it predicts that the level of pollution can exceed the threshold, it suggests leaving the car in the parking area, continuing the trip using alternative transport services. In addition, the system estimates in real-time the number of free slots in the parking lot. In this way, it can recommend the most appropriate parking lot to leave the vehicle. This fact highlights the opportunity to provide intermodal transport solutions. Table 3 shows the tasks performed by the different components of the architecture in order to accomplish the requirements of the test scenario.

**Table 3.** Urban scenario related tasks.

Task	Test Equipment			
	GW	Sensor	GUI	CLU
Traffic flow monitoring in the selected area		✓		
Parking space vacancy monitoring in via <i>Pietrasantina</i> and <i>Palasport</i> parking areas		✓		
Air pollution level monitoring in city centre		✓		
Data management and event publishing	✓			
Event processing, alert messages generation				✓
Evolution of congestion and pollution levels prediction for the next period of time				✓
Suggestion of new parking area target				✓
Display the availability of both parking areas			✓	
Display the LTZ area in the map			✓	
Suggestion of alternative transport modes			✓	

The verification of some of these use cases have been reproduced in the laboratory due to the need of producing events related to current pollution levels.

### 5.2. Highway Scenario in Lisbon, Portugal

The A5 highway of Lisbon is a 25 km (16 miles) long motorway that connects the capital city of Portugal to Cascais. The first section of this infrastructure was opened in 1944, becoming the first



motorway in Portugal and one of the firsts in the world. Nowadays, it is the most travelled motorway of the country and one of the most congestion prone ones. Six GWs were installed on the motorway road side cabinets. The RSUs were interconnected, and together with the GWs made possible the implementation of the platform on the field trial location.

Field trials were performed by partners of the ICSI project (IT, BRISA and INTECS). The following summarizes the tested use cases, not being the purpose of this work the detailed description of those trials, which will be subject of forthcoming works. In this context, the proposed highway test scenario includes the implementation of the following use cases:

- Monitoring of anomaly in traffic flows (congestion);
- Accident warning; and
- Road works warning.

The proposed ITS distributed architecture provides in-route traveller information about traffic and road conditions according to both static and dynamic rules. In this way, drivers who are approaching a traffic jam can take some precaution measures, like reducing the speed in advance.

Each RSU's GW is configured to get *Abnormal Traffic* events (e.g., accident or roads work warning) from the next GW on the road. Additionally, each GW also gets *Vehicle Counter* events. These events come from the GW's attached sensors, and they are delivered to the CLU in order to detect congestion using the implemented artificial intelligence. In line with this, if congestion is detected, a *Congestion Level* event is launched. Each GW is listening for *Congestion Level* events from itself and from the next GW on the road in order to act with foresight and warn the drivers about expected traffic jumps.

The produced messages have been also successfully received at the GUI Web Platform and the HMI, included as a mobile application inside the vehicles, informing that it is recommended to take exit to avoid traffic congestion or alerting about an accident with foresight. Figure 7 shows a snapshot of the demonstrator application GUI for the highway test scenario.



**Figure 7.** Alert messages received by the user: fragment of the demonstrator web application.

## 6. Conclusions

This work, based on the ICSI European project, exposes a new paradigm in the ITS domain, moving the system intelligence from the control centre to the peripheral devices. The presented architecture uses a set of gateways, installed in the road infrastructure to process and gather all the sensors data independently and in a distributed way, without the need to contact a central subsystem.

The presented architecture, with the participation of the sensors networks on the road, the data distribution platform, the collaborative learning units and, finally, the ITS applications, encompasses the entire process of capture and management of available road data, enabling the generation of services to promote transportation efficiency and performance, among others.

This paper provides a description of the overall system architecture explaining the role of the different elements of the system and the way the intelligence and the sensed-data are provided.

Furthermore, a more specific view of the Collaborative Learning Unit, the component responsible for providing intelligence to the system, is presented.

Besides that, the developed demonstration environment has been presented. This environment counts with a web application able to display the complete operation of the CLUs with real time monitoring of the sensors, actuators, and the messages generated by CLUs on each GW.

Finally, the summary of a set of field trials performed in Italy and Portugal under the collaboration in the ICSI European project is exposed, fulfilling the main goal of the project: the deployment of cooperative systems, based on V2X communication technologies, with the aim of enabling safer and more efficient mobility.

The proposed system is able to demonstrate the feasibility of a flexible and innovative platform from the architectural point of view that has been specialized for the ITS context, but generally applicable in other contexts and compatible with the Internet of Things paradigm, allowing considerable flexibility and scalability of the system.

**Acknowledgments:** This work has been partially funded by the European Commission under the Seventh Framework Programme. Intelligent Cooperative Sensing for Improved Traffic efficiency (ICSI) project (Grant agreement No.: 317671). This work is an extension of the paper “Cooperative decision making ITS architecture based on distributed RSUs”, presented at the 9th International Conference on Ubiquitous Computing & Ambient Intelligence (UCAmI 2015) [22].

**Author Contributions:** Asier Moreno and Eneko Osaba wrote the paper. Asier Perallos, Asier Moreno and Giovanni Iovino designed and conceived the architecture. Enrique Onieva and Pablo Fernandez conceived and designed the experiments related to CLU testing. Giovanni Iovino in collaboration with the partners of the ICSI project (CNR, CNIT, IT and Brisa) performed the experiments. All authors supervised the paper and provided substantive comments.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. ICSI: Intelligent Cooperative Sensing for Improved Traffic Efficiency. Available online: <http://www.ict-icsi.eu> (accessed on 22 April 2016).
2. Weiland, R.J.; Purser, L.B. Transportation Research Board. Intelligent Transportation Systems. In *Transportation in the New Millennium*; Transportation Research Board: Washington, DC, USA, 2000.
3. Weiß, C. V2X communication in Europe—From research projects towards standardization and field testing of vehicle communication technology. *Comput. Netw.* **2011**, *55*, 3103–3119. [CrossRef]
4. Papadimitratos, P.; La Fortelle, A.; Evenssen, K.; Brignolo, R.; Cosenza, S. Vehicular communication systems: Enabling technologies, applications, and future outlook on intelligent transportation. *IEEE Commun. Mag.* **2009**, *47*, 84–95. [CrossRef]
5. Toulminet, G.; Boussuge, J.; Laurgeau, C. Comparative synthesis of the 3 main European projects dealing with cooperative systems (cvis, safespot and coopers) and description of COOPERS Demonstration Site 4. In Proceedings of the 11th International IEEE Conference on Intelligent Transportation Systems, Beijing, China, 12–15 October 2005.
6. Paul, A.; Haley, D.; Grant, A. Cooperative intelligent transport systems: 5.9-GHz field trials. *IEEE Proc.* **2011**, *99*, 1213–1235.
7. Cooperative Vehicle-Infrastructure Systems (CVIS). Available online: <http://www.ecomove-project.eu/links/cvis/> (accessed on 23 June 2016).
8. SAFESPOT Integrated Project. Available online: <http://www.safespot-eu.org> (accessed on 23 June 2016).
9. COMeSafety<sup>2</sup>. Available online: <http://www.ecomove-project.eu/links/comesafety/> (accessed on 23 June 2016).
10. DRIVE C2X—Accelerate Cooperative Mobility. Available online: <http://www.drive-c2x.eu/project> (accessed on 23 June 2016).
11. Rellermeyer, J.; Alonso, G.; Roscoe, T. R-OSGi: Distributed applications through software modularization. In Proceedings of the 8th International Conference on Middleware, Newport Beach, CA, USA, 26–30 November 2007.



12. Eugster, P.; Felber, P.A.; Guerraoui, R.; Kermarrec, A.-M. The many faces of publish/subscribe. *ACM Comput. Surv.* **2003**, *35*, 114–131. [[CrossRef](#)]
13. Quinlan, J.R. Induction of decision trees. *Mach. Learn.* **1986**, *1*, 81–106. [[CrossRef](#)]
14. Quinlan, J.R. *C4.5: Programs for Machine Learning*; Elsevier: Amsterdam, The Netherlands, 2014.
15. Quinlan, J.R.; Michie, D. *Discovering Rules by Induction from Large Collections of Examples. Expert Systems in the Micro Electronic Age*; Edinburgh University Press: Edinburgh, UK, 1979.
16. Cortes, C.; Vapnik, V. Support vector machine. *Mach. Learn.* **1995**, *20*, 273–297. [[CrossRef](#)]
17. Masetic, Z.; Subasi, A. Detection of congestive heart failures using c4. 5 decision tree. *Southeast Eur. J. Soft Comput.* **2013**, *2*, 74–77.
18. Yao, Z.; Liu, P.; Lei, L.; Yin, J. R-c4. 5 decision tree model and its applications to health care dataset. In Proceedings of the IEEE International Conference on Services Systems and Services Management, Chongqing, China, 13–15 June 2005.
19. Adhatrao, K.; Gaykar, A.; Dhawan, A.; Jha, R.; Honrao, V. Predicting students' performance using id3 and c4. 5 classification algorithms. *Int. J. Data Min. Knowl. Manag. Process* **2013**, *3*, 39–52. [[CrossRef](#)]
20. Behrisch, M.; Bieker, L.; Erdmann, J.; Krajzewicz, D. SUMO—Simulation of Urban Mobility. In Proceedings of the SIMUL 2011 3rd International Conference on Advances in System Simulation, Barcelona, Spain, 23–29 October 2011.
21. Fraternali, P.; Rossi, G.; Sánchez-Figueroa, F. Rich internet applications. Internet Computing. *IEEE Internet Comput.* **2010**, *14*, 9–12. [[CrossRef](#)]
22. Moreno, A.; Onieva, E.; Perallos, A.; Iovino, G.; Fernandez, P. Cooperative decision making ITS architecture based on distributed RSUs. In *Ubiquitous Computing and Ambient Intelligence. Sensing, Processing, and Using Environmental Information*; Springer International Publishing: Zurich, Switzerland, 2015.



© 2016 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC-BY) license (<http://creativecommons.org/licenses/by/4.0/>).