LIFE+10 ENV/IT/000389

INTEGREEN

Action 3: System design

D.3.1.1

Data management unit and environmental stations front-end design

Project Coordinating Beneficiary	Municipality of Bolzano
Project Associated Beneficiary n.2	TIS innovation park (TIS)
Project Associated Beneficiary n.3	Austrian Institute of Technology (AIT)

Document history

Date	Document Author(s)	Document Contribution
31/03/14	Roberto Cavaliere, Paolo Valleri, Stefano Seppi (TIS), Reinhard Kloibhofer, Franco Fresolone, Wolfgang Ponweiser (AIT)	Document finalization

Dissemination level: CO[1]
Delivery month: M31
Status: submitted to EC

[1] **PU** = Public; **CO** = Confidential (for members of the consortium only, including the European Commission); **RE** = Restricted, i.e. only a selected target of stakeholders defined by project consortium and validated by European Commission can access to the document. The final dissemination level will be defined in cooperation between the INTEGREEN consortium and the European Commission.

Table of Contents

Table of Figures

Table of Tables

1 Introduction

1.1 Purpose of the document

The main objective of Task 3.1 Action n.3 is to obtain a complete technical definition of the new Environmental Supervisor Centre within the proposed INTEGREEN architecture (Figure 1). This document covers the entire set of components foreseen at the traffic management centre layer, expect for the vehicle data-source and the entire front-ends layer (in the original denomination named as "*vehicle-to-centre front end*" and "*public web interfaces*"), which are reported in the deliverable D.3.1.2 [1]. The work on the prototype mobile system in on the contrary covered by the activities of Task 3.2, which is reported in deliverables D.3.2.1 and D.3.2.2 [2]- [3], respectively.

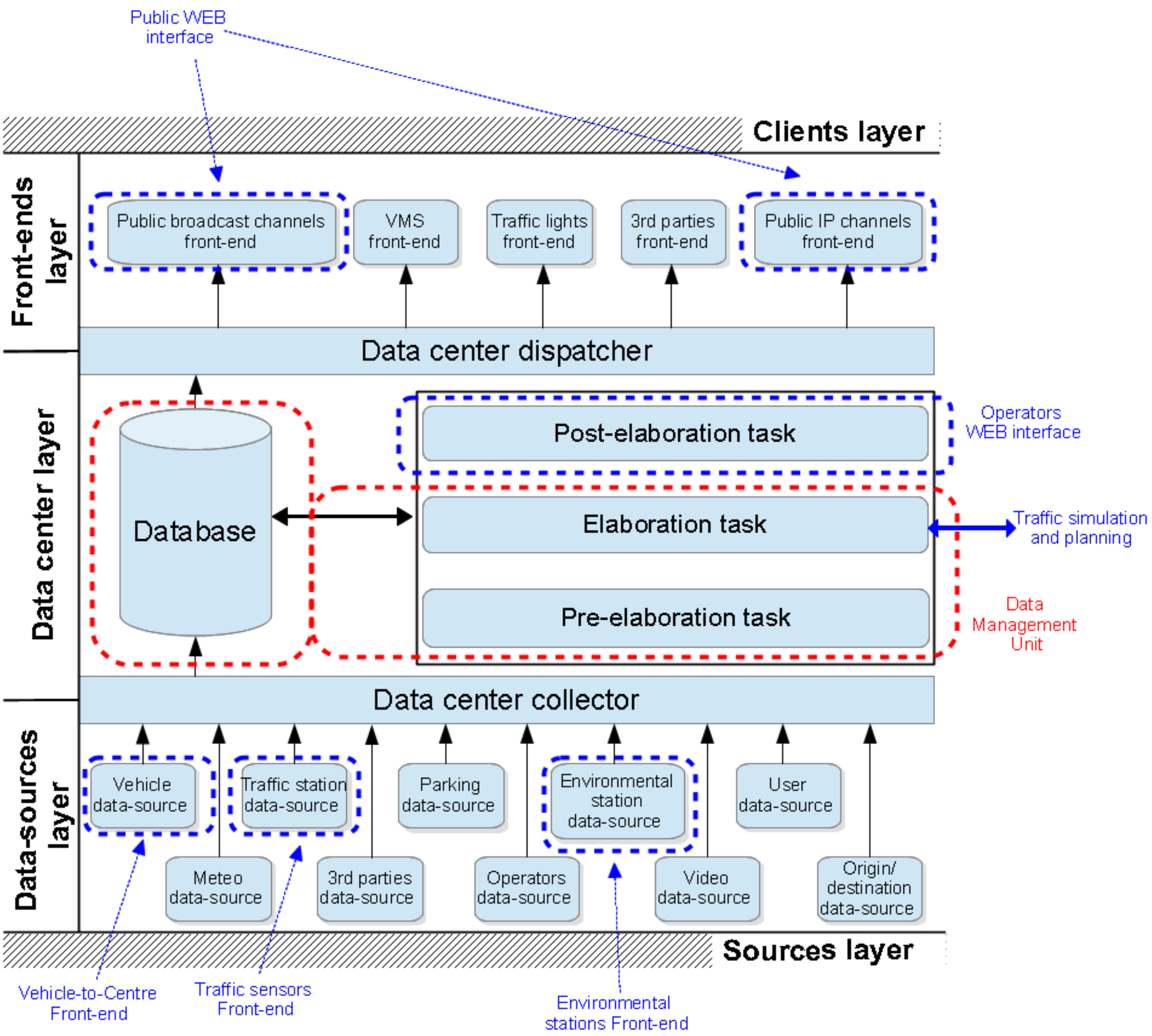

Figure 1: INTEGREEN system architecture **[4]**.

This document aims to represent a specific plan for its following actuation, which is foreseen in Action n.4.

The functional and logical architecture of the single components and of the system as a whole will be outlined. Hardware and software components will be selected and where necessary designed ad hoc, and communication specifications, protocols and data models are clearly specified.

1.2 Specification definition methodology

The design process is the second step of the V-model approach followed within the project (Figure 2). This approach is a very common engineering methodology which is very often applied for the deployment of ITS projects like INTEGREEN, as stated in a very comprehensive handbook provided by the U.S. Department of Transportation – Federal Highway Administration [5].

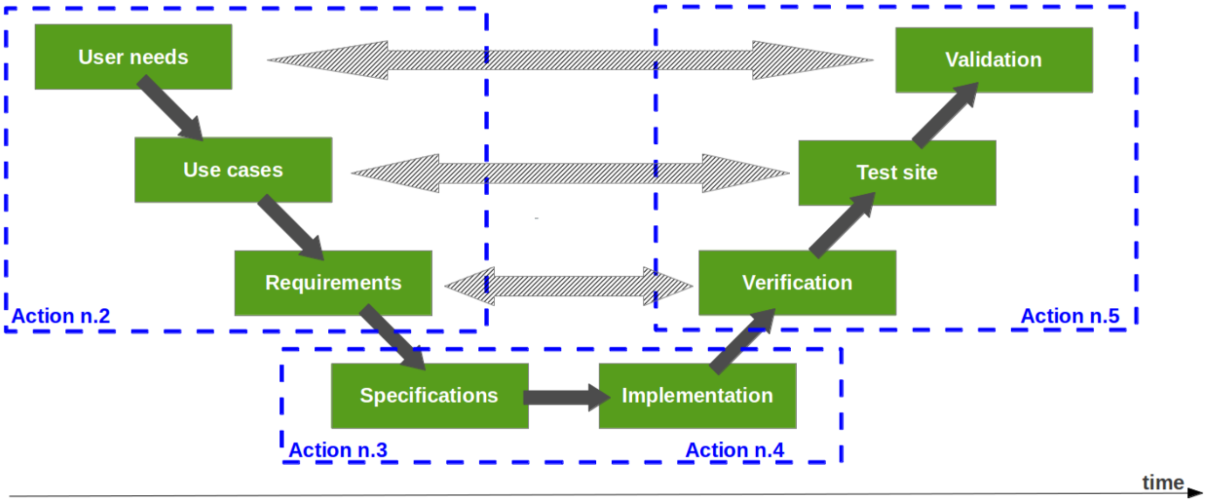

Figure 2: The V-model approach adopted in INTEGREEN **[4]**.

The system design is developed on the base of the system requirements identified in Action n.2. In the previous steps, the focus has been mainly on the investigation of the problem to be solved; this specification phase is the first moment of the project in which the proposed solutions are analyzed and evaluated at a low level, and represents a very important bridge with the system implementation that is going to be successively performed, as illustrated in Figure 3.

Two levels of design are typically introduced:

- **High-level design**, also commonly referred as *architectural design*. IEEE 610, the IEEE Standard Computer Dictionary defines architectural design as "*the process of defining a collection of hardware and software components and their interfaces to establish the framework for the development of a computer system*" [6]. ITS projects may include several computer systems, a communications network, distributed devices, facilities, and people. High-level design defines a framework for all of these project components.

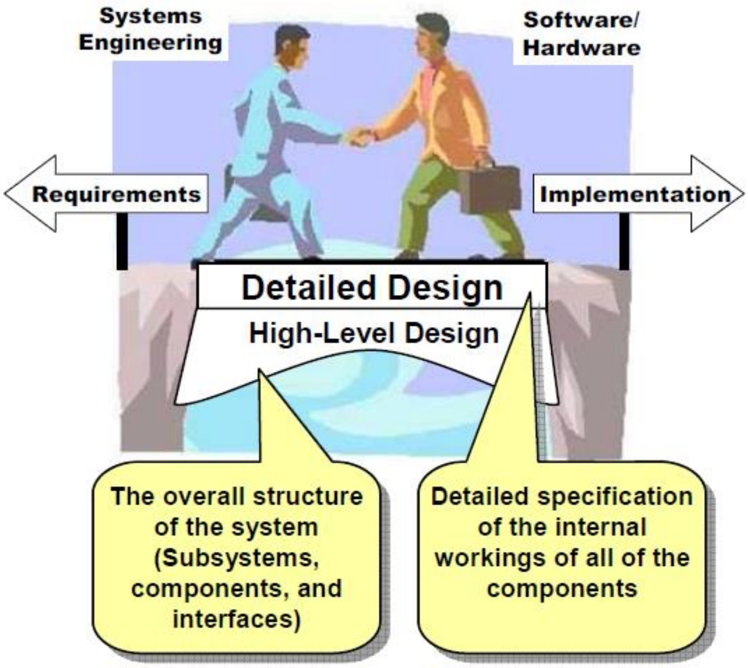

Figure 3: A focus of the role of the design process in the V-model approach **[5]**.

- **Detailed design**, which is the complete specification of the software, hardware, and communications components, defining how the components will be developed to meet the system requirements. The software specifications must be defined in a way to allow the software team to write the individual software module, while the hardware specifications must be detailed enough so that the hardware components can be fabricated or purchased.

In INTEGREEN, most of the first high-level design step of the Supervisor Centre has been already covered within Action n.2, in particular in Chapter n.6 and n.7 of D.2.1.1 where the entire system architecture and a specific detail of the role of each component has been clearly defined. This document will therefore concentrate more on the detailed design of the system elements which are considered in this deliverable. It is worth noting that a key element during this phase, that will be strongly considered in this decision-making process, is the actual and future capability of the system to be developed, integrated, maintained and upgraded / exploited over time. Fortunately, this process is simplified by the fact that the proposed intervention relies on ITS implementations that are already in place, and therefore represent just a “smart” integration of what’s already existing.

1.2.1 High-level design process

The high-level design process is typically organized on top of the following set of actions, as illustrated in Figure 4:

- **Evaluation of the off-the-shelf components**. More specifically, the purpose of this activity is to identify the hardware / software components that need to be purchased, reused, or developed from scratch. One of the key elements to be considered here is related to the necessity to avoid considering specifications that can be supported by

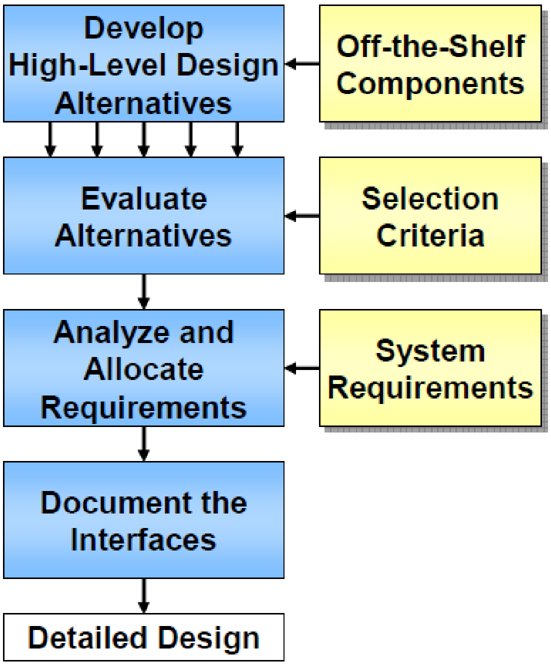

Figure 4: The flow of activities in the high-level design process **[5]**.

custom and proprietary solutions, and that create possible “lock-in” effects with vendors.

- **Develop and evaluate high-level design alternatives**. A certain system component, even the simplest one, can be implemented through very different approaches. Potential alternatives may have a significant impact on performance, reliability and life-cycle costs; therefore it is necessary to find the optimal solution based on pre-defined selection criteria, which cover typically:
    - existing physical and institutional boundaries;
    - ease of development;
    - integration & upgrading.
- **Analyze and allocate requirements**. The main goal of this activity is to allocate the full set of requirements to the basic elements of the system design, evaluated in its full granularity.
- **Document the interfaces and identify standards**. This task deals with the identification, documentation and management of the internal and external interfaces (i.e. the interfaces between the components within the INTEGREEN system, and with other components outside it, respectively). This a very important phase for INTEGREEN, since many of the data collected in the Supervisor Centre are provided by external services managed by local agencies. It is therefore necessary to foresee a negotiation time for the activation of these interfaces, in order that they are fully compatible with the project’s needs.

1.2.2 Detailed design process

The detailed design process is typically organized on top of the following set of actions, as illustrated in Figure 5:

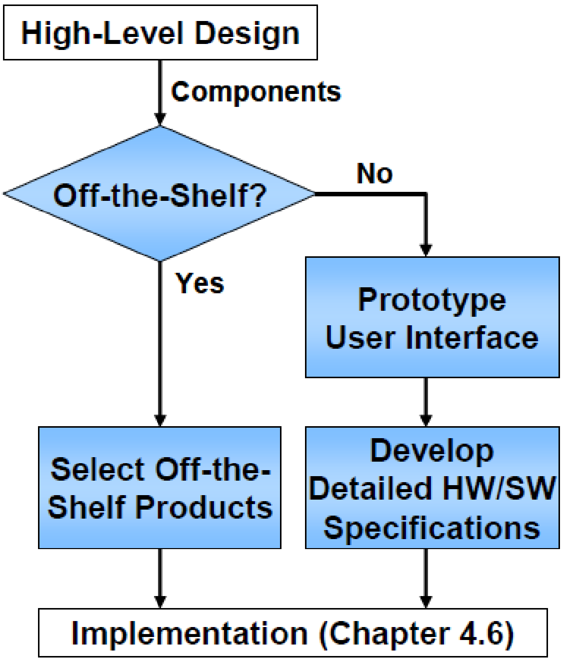

Figure 5: The flow of activities in the detailed design process **[5]**.

- **Design a first prototype user interface**. In computer science, it is more and more common to address complex software projects with "lean" approaches, which are based on the idea to develop a certain architecture in an iterative way, starting from a very simple implementation of it. The advantage of this approach is that a first implementation can be available quite quickly, and target user groups can be actively involved in order to provide continuous feedbacks in the successive implementation process.

- **Develop detailed hardware and software component design specifications**. In the high-level process, each component is defined in terms of its functionality and performance, with particular focus on its interfaces to external systems and other components. The level of detail in the detailed design specifications is greater than that in the high-level design, since the in this phase it is exactly specified how a component is going to be implemented in order to meet the requirements.

- **Select off-the-shelf products**. In the case of purchasing of equipment from external vendors (a case covered by INTEGREEN as well), there is the need to define a set of specifications to be used e.g. in a public tender in order to tell potential sub-contractors the type of products one is interested in. This can be carried out with different logics and approaches; in INTEGREEN, in order to have as much a technology-independent architecture, the idea is to follow a performance-based approach, in which it is given the maximum freedom to subcontractors to suggest the right technology solution capable to satisfy a certain set of specifications.

1.2.3 System design with FRAME

At the end of the requirements' process of Action n.2, which was conducted thanks also to a reverse engineering approach (i.e. with the idea to postpone the definition of requirements after a preliminary evaluation of the technical specifications), project partners have identified the potential added value to consider in the design process (at least in its first stage) a reference design tool, able to provide different "views" of the system. This decision was also made on top of the consideration that other EU urban realities may be interested to replicate the technical experience carried out in the project, but this would have been difficult without any specific output coming from a standardized design tool.

For this reason, INTEGREEN has decided to "map" its reference architecture with FRAME, the European Intelligent Transport System Framework Architecture which was developed in a set of different projects funded by the European Commission under the 6th and 7th Framework Programme for Research and Technological Development [7]. The intention was to create a minimum stable framework able to guarantee the integrated and interoperable deployment of ITS within the European Union. FRAME's area of interest, if matched with the V-model approach, is exactly located in the scope of the requirements and high-level design, as illustrated in Figure 6. This kind of framework is therefore a very appropriate instrument for covering the transition between Action n.2 and Action n.3, with the possibility then to easily jump in to the detailed design phase.

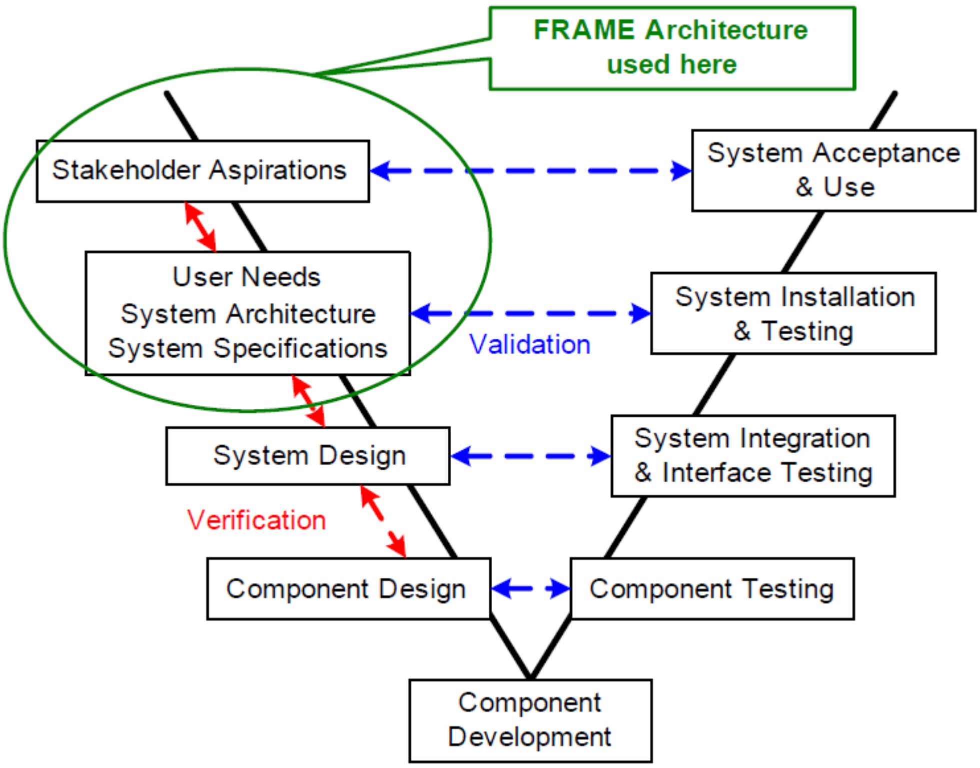

Figure 6: The FRAME design process within a V-model approach **[7]**.

1.3 Document structure

This deliverable is organized as follows. Chapter 2 covers the results of the initial design work done with FRAME. The following chapters report the detailed design activities made for each of the system components covered by this deliverable. More specifically, Chapter 3 illustrates the design choices made at the data sources layer, which include the roadside stations as well, while Chapter 4 presents the overall design of the data center layer.

2 The FRAME model of INTEGREEN

2.1 An introduction to FRAME

The need for ITS architectures started to be recognized in the early 1990’s, when the number of possible applications and services that ITS could provide increased greatly. Instead of producing unique architectures for each deployment, it was immediately realized that it would have been much more efficient to have a reference ***Framework Architecture***, from which individual ITS architectures could be developed. There are several obvious advantages for doing this, the most relevant are:

- it is much more quicker and cheaper to produce a comprehensive ITS architecture starting from a Framework Architecture, instead than producing an own one from zero;

- each derived ITS architecture has the same properties as the Framework Architecture from which it has been produced. Therefore, the use of similar equipment in different deployments is facilitated, and their potential market is thus potentially extended.

The first Framework Architecture to be produced was the National ITS Architecture for the USA, which was first published in 1996 and now is available in its Version 7.0, fully synchronized with the ITS architecture for Canada [8]. This architecture is now capable to cover very broad ITS domains, including aspects that are still at a research stage such as the connected vehicle services (Figure 7). A software application called Turbo Architecture (version 7.0) is available for free as well as other numerous instruments (such as guidelines, indications, tutorials, etc.) at disposal mainly (but not only) to regional agencies for the harmonized deployment of ITS.

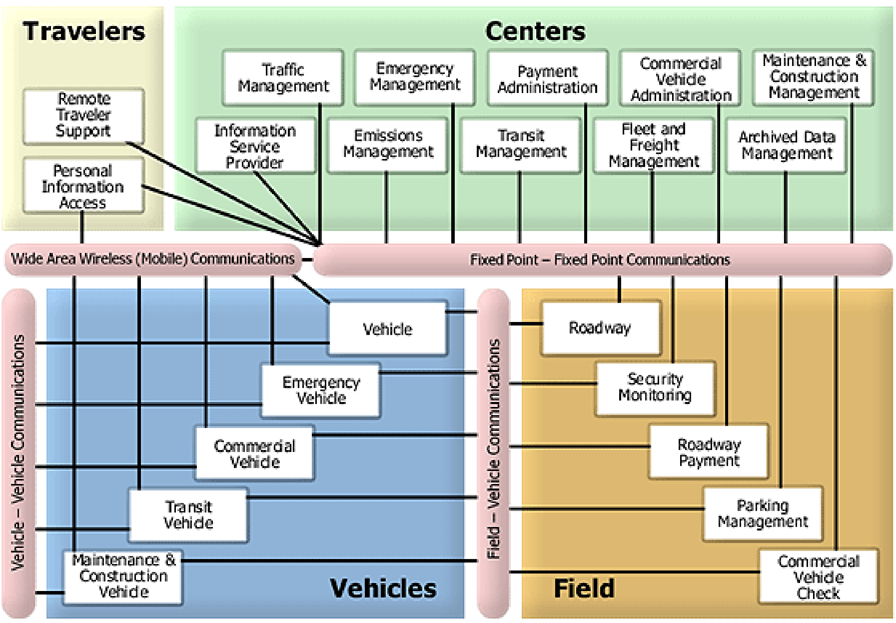

Figure 7: The high-level architecture diagram of the National ITS Architecture adopted in the USA **[8]**.

In Europe, the first initiative in this direction has been KAREN, a project funded by the European Commission in 2000. Most of the documentation which is already available online at the website http://www.frame-online.net was produced in this project, which was completed later by its follow-up FRAME-S, concluded in 2004. Finally the E-FRAME project, concluded in 2011, has managed to further extend the architecture so that cooperative systems fundamentals developed in key research & development projects such as COOPERS, CVIS and SAFESPOT are included as well (Figure 8).

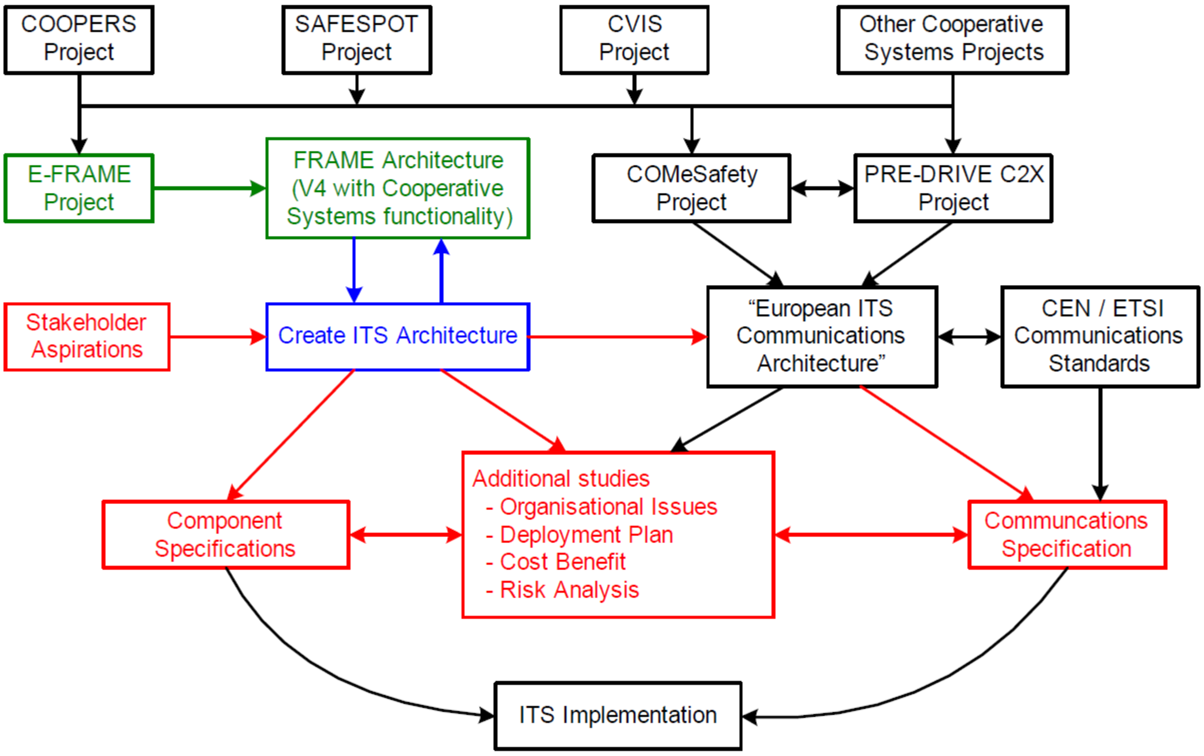

Figure 8: FRAME and the link to other international initiatives **[7]**.

Although these two Framework Architectures have many similarities there are also a number of significant differences. The most relevant one is related to the fact that while the US National ITS Architecture shows the relationship between *physical components*, with the possibility for users to choose those components capable to satisfy their requirements, the European ITS Framework Architecture shows the relationship between *functions* only. This means that users must first choose the functions that they need to satisfy their requirements, and then make their own decisions as to how they will be allocated to physical components. The advantage of this approach, which is more coherent with the independent position of the Member States within the European Union, is that a technology-independent view of a certain ITS deployment can be defined, with the consequence that it can be still maintained and easily extended when new technologies are available.

It is worth noting that FRAME has also being standardized through a specific ISO standard, namely **ISO TR 26999:2012**, a technical report describing in detail the entire FRAME methodology. FRAME has also contributed to the definition to other key standards, in particular **ISO 14813:2007**, in which a description of a reference architecture model for the ITS sector is provided [9]- [10].

2.2.1 Why a Framework Architecture?

When starting an ITS project, the danger that one may encounter is to jump very quickly into the "more exciting" stages of design and implementation, so that the use of new and advanced technology can be reached as quickly as possible. The risk is that the outputs of the initial phases of the V-model will not be complete and/or correct, and it becomes very expensive to rectify the discrepancies that can begin to manifest during the implementation process. A qualitative analysis of the order of costs to make substantial changes at a certain stage of an ITS implementation is illustrated in **Fehler! Verweisquelle konnte nicht gefunden werden.**. A work done in the 1980's and 1990's by organizations such as the Software Engineering Institute at Carnegie Mellon University demonstrated the existence of a so-called "***10:100:1000 Rule***", which indicates that the cost of correcting faults in a system increases exponentially (by about a factor of 10) during each successive stage of a lifecycle.

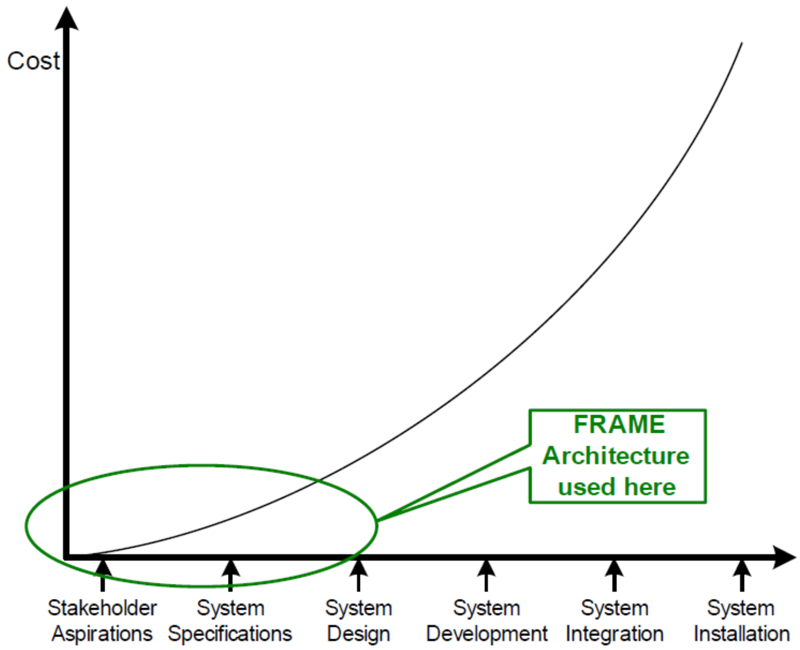

Figure 9: The cost to introduce changes during an ITS implementation **[7]**.

2.2 The FRAME architecture creation process

The creation process of a FRAME architecture is graphically illustrated in Figure 10. The steps are the following:

1. **Collection of stakeholders aspirations**, i.e. statements that express, in a non-structured way the expectations and desires of the various stakeholders for the services that the final ITS implementation will provide.

2. **User needs definition**, i.e. the translation of the stakeholders aspirations in a consistent fashion, with a clear meaning and testable properties. In FRAME, a set of 550 pre-defined user needs are available, but with the possibility to create own ones in case of need.

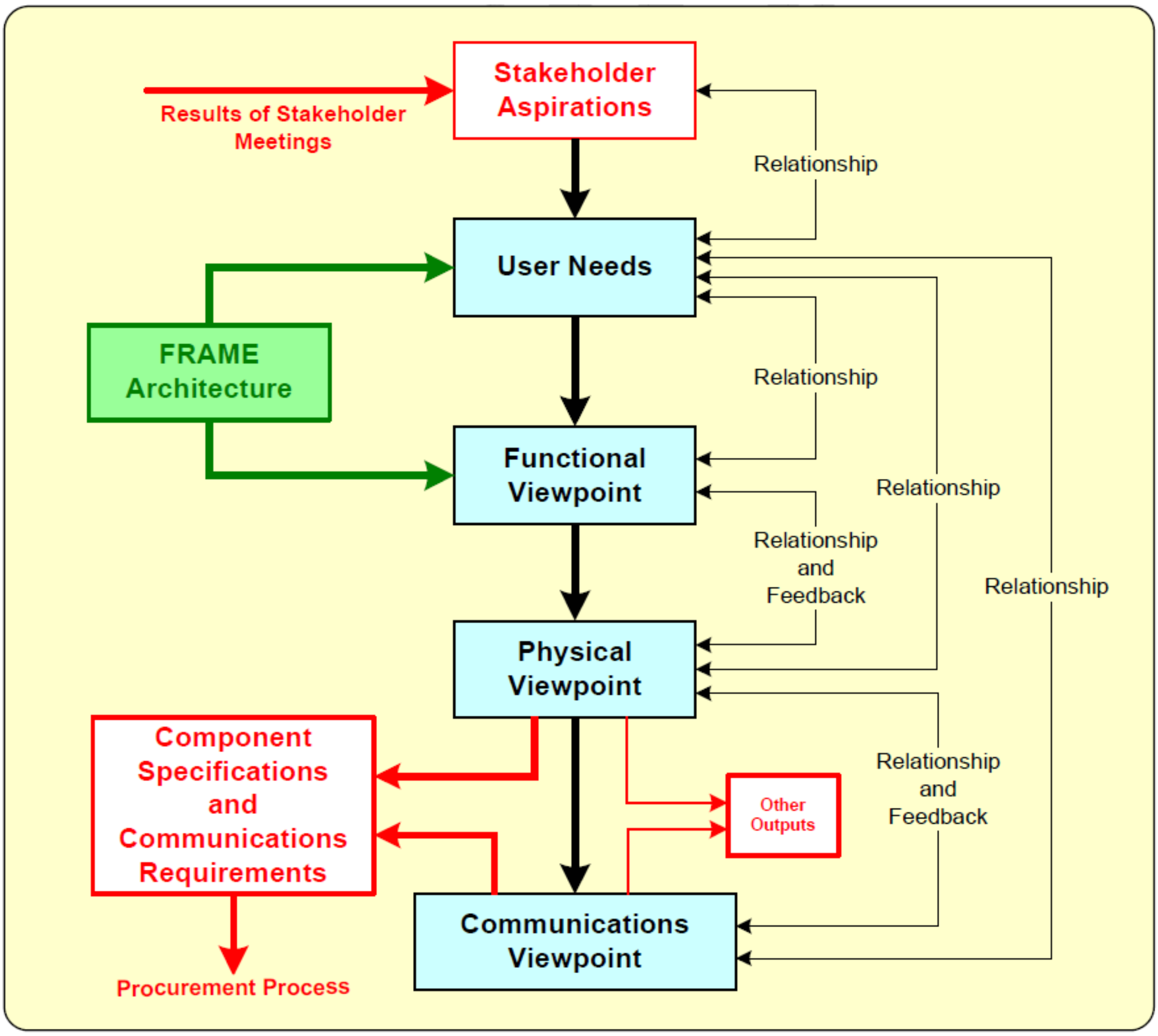

Figure 10: The creation process of a FRAME architecture **[11]**.

3. **Creation of the Functional Viewpoint**, in which the system functionalities are selected. In FRAME, this is modeled as a data flow diagram containing not only functions but also data stores and terminators, with all associated data flows. This is an alternative to UML approach, based on software-engineered use cases and object models.

4. **Creation of the Physical Viewpoint**, in which the each component of the functional viewpoint is mapped within a subsystem or a module within a subsystem, as indicated in Figure 11.

5. **Creation of the Communications Viewpoint**, in which the physical interfaces between systems are finally specified, e.g. in terms of data types, communication channel, speed and security.

An additional viewpoint that has been recently added is the **Organizational Viewpoint**, which tries to cover the responsibility issue, i.e. an answer to the typically hot question "who is responsible for what?".

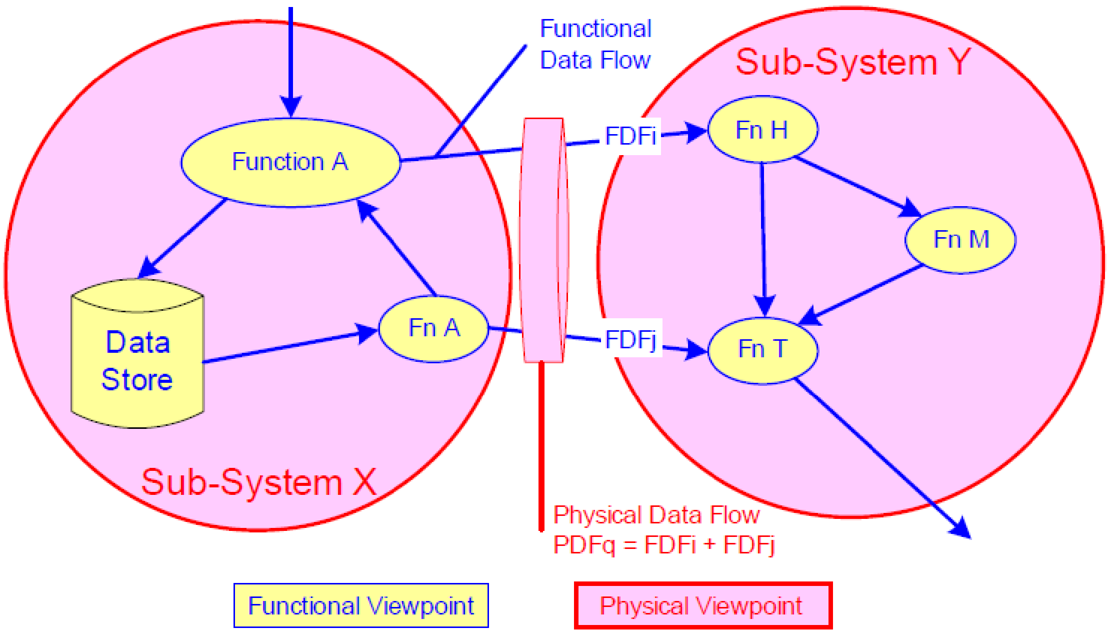

Figure 11: Functional and Physical Viewpoint "layers" in FRAME **[7]**.

2.3 The FRAME architecture tools

In order to create a FRAME architecture, two tools are available: the *Browsing Tool* and the *Selection Tool*. The Browsing Tool gives the ability to *browse* through the FRAME architecture, while the Selection Tool gives the possibility to *select* a sub-set of the overall Functional Viewpoint which satisfies a specific set of ITS user needs, and which will correspond to a specific Physical Viewpoint.

2.3.1 The Browsing Tool

As already anticipate, during the different EU projects many reference pre-prepared architecture elements have been developed in order to (i) facilitate the work of FRAME users, and above all (ii) to ensure that different FRAME users can use the same basic elements during this process, and thus ensure harmonization and functional interoperability. In order to have the possibility to quickly analyze all those elements, a HTML-based GUI viewable through a standard Internet browser has been introduced (Figure 12). It gives the possibility to navigate in a very user-friendly way within the Data Flow Diagrams (DFD) in a hierarchical manner, thus understanding the existing relationships between different ITS systems or domains at different levels.

In Figure 13, Figure 14 and Figure 15 it is possible to have an idea of the different layers of DFDs. A FRAME user can easily click the hierarchical menu at the left side or directly in the diagrams in order to move from one layer to another. Once the root of a DFD is reached, one can get a detail of a specific low-level function, with a synthetic indication of the covered facilities, associated functional requirements, an indication of the parent higher level function, the list of input and output logical data flows, and the list of associated user needs (Figure 16).

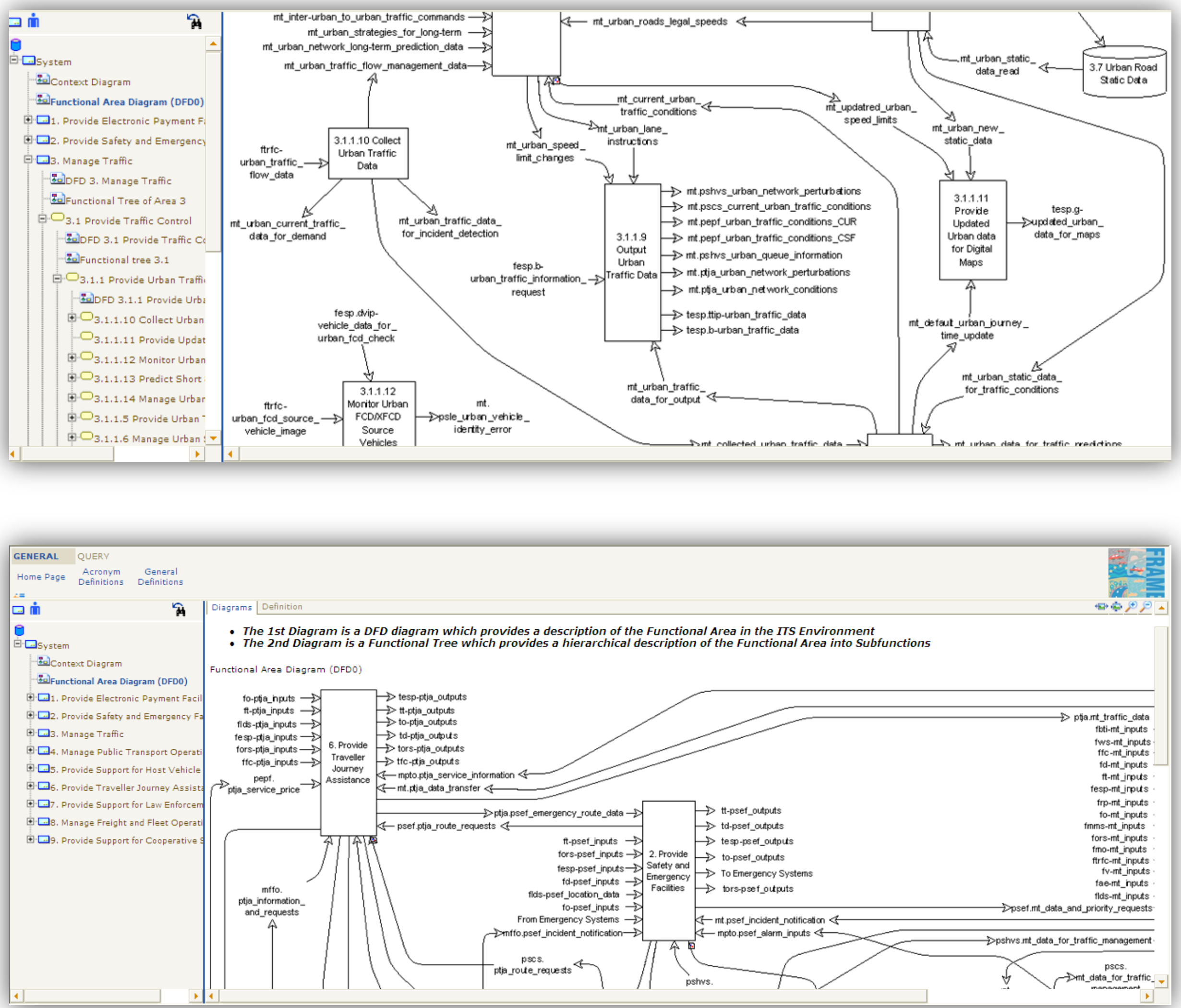

Figure 12: The home screenshot of the Browsing Tool.

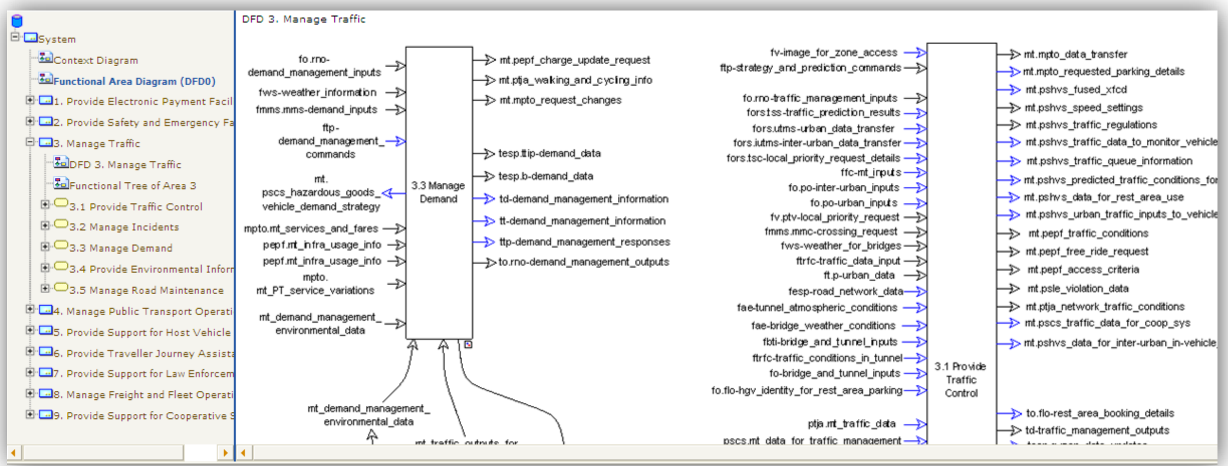

Figure 13: The Browsing Tool – Data Flow Diagram at "layer" n.1.

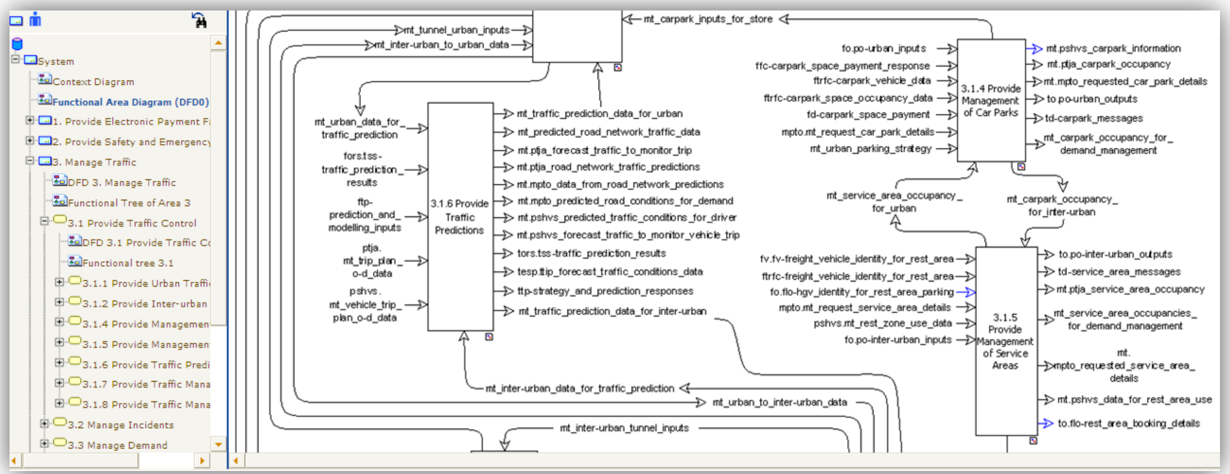

Figure 14: The Browsing Tool – Data Flow Diagram at "layer" n.2.

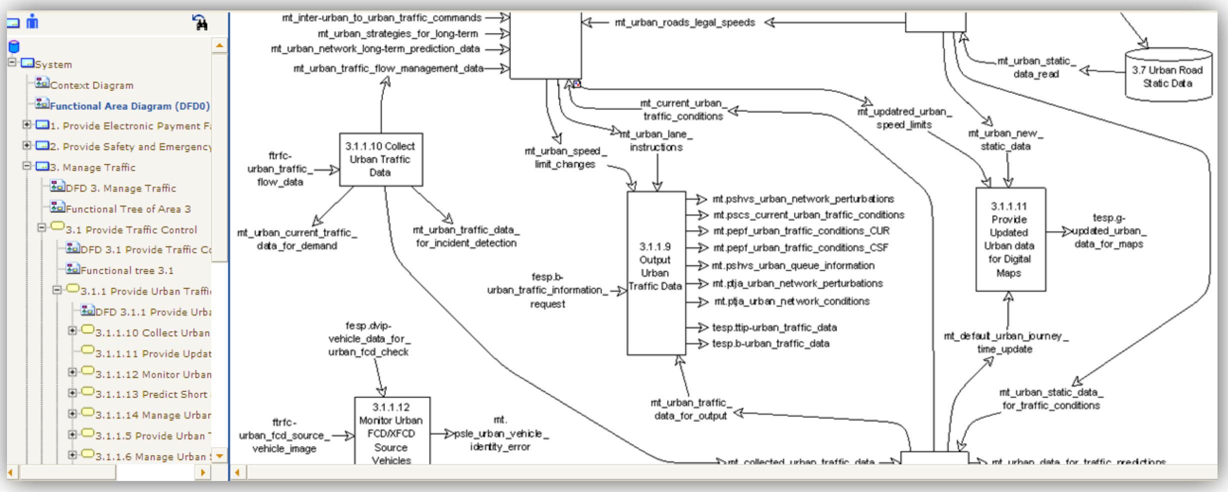

Figure 15: The Browsing Tool – Data Flow Diagram at "layer" n.3.

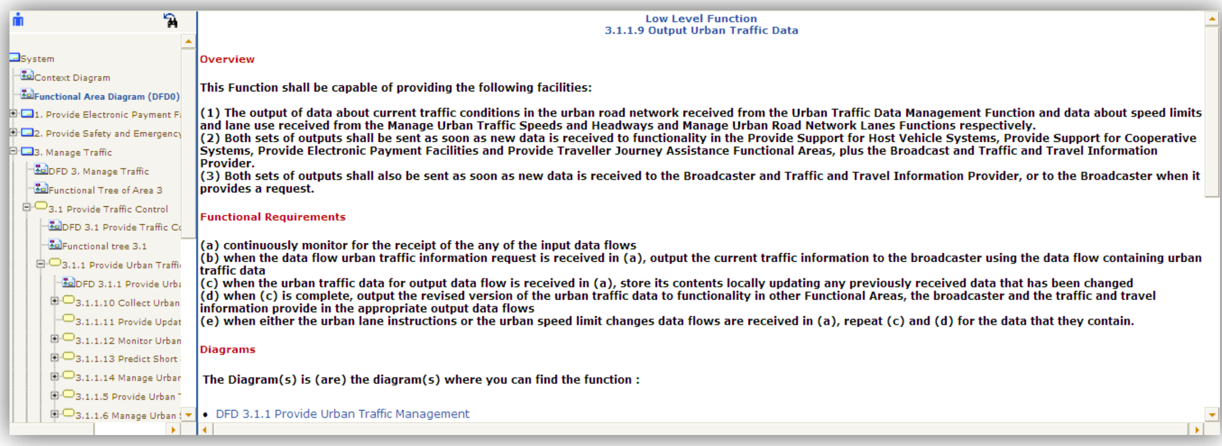

Figure 16: Description of low-level functions in the Browsing Tool.

2.3.2 *The Selection Tool*

The Selection Tool is the operative instrument that the ITS specialist must use, while checking the Browsing Tool, in order to develop his/her own FRAME architecture. The covered interest area of the Selection Tool is indicated in Figure 17.

More specifically, the tool is a simple stand-alone program that can gives the possibility to create the desired viewpoints by following the above design process, i.e. – if we just focus on the Functional Viewpoint – by selecting (i) the set of most appropriate user needs, (ii) the set of low-level functions (the boxes in the Browsing Tool), which are in the condition each of them is to address a certain number of user needs, (iii) the set of data flows (the arrows in the Browsing Tool), which connect the functions, and (iv) the set of terminators and actors, i.e. external entities to which certain data flows may be connected. This process must be carried out in an iterative way, since it is quite common to make accidentally some inconsistencies with respect to the overall FRAME architecture.

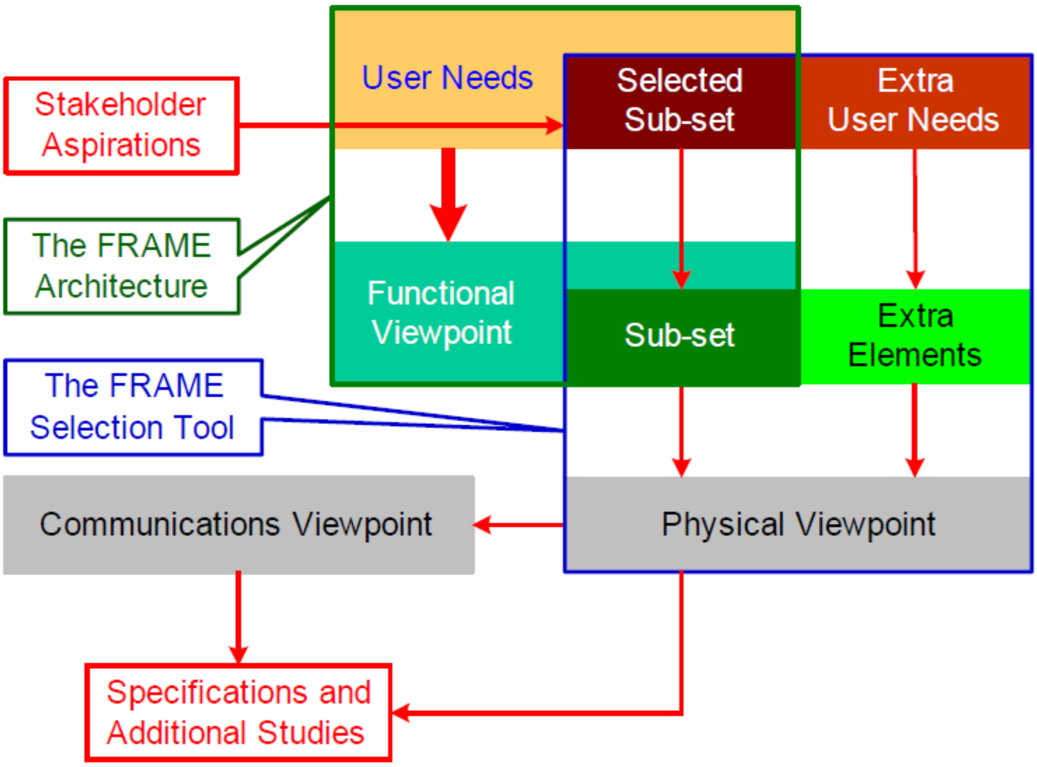

Figure 17: The domain covered by the Selection Tool **[7]**.

For simplicity sake, user needs are classified in terms of reference ITS domains, which are:

- management activities;
- law reinforcement;
- financial transactions;
- emergency services;

- travel information;
- traffic management;
- intelligent vehicle systems;
- freight and fleet operations;
- public transport.

In this way, a FRAME user can directly go into the class of interest without controlling the full set of user needs. A similar logic is applied for the low-level functions as well, as presented in the Browsing Tool. In this case FRAME calls them “areas”, and are divided in:

- Area 1 – Provide Electronic Payment Facilities;
- Area 2 – Provide Safety and Emergency Facilities
- Area 3 – Manage Traffic
- Area 4 – Manage Public Transportation Operations
- Area 5 – Provide Support for Host Vehicle Services
- Area 6 – Provide Traveler Journey Assistance
- Area 7 – Provide Support for Law Enforcement
- Area 8 – Manage Freight and Fleet Operations
- Area 9 – Provide Support for Cooperative Systems

Once the Functional Viewpoint is completed (eventually integrated with own user needs and functional elements in the architecture), it is possible to design the Physical Viewpoint. In this case, the steps to be carried out are (i) the definition of the sub-systems and the allocation of functions to them, (ii) the creation of modules and again the allocation of function as well as data stores to them, and finally (iii) the creation of the physical data flows. It is worth noting that a Physical Viewpoint must be specifically associated with a certain Functional Viewpoint. Once this association is completed, the Functional Viewpoint is “cloned” and it is not possible to use it anymore. Once all this selection process is complete, the tool gives the possibility to have as output a certain number of “reports”. The main actual drawback of FRAME is that these kind of output are quite uncomfortable for a human being, and unfortunately their interpretation is not immediate if the mapped ITS implementation is complex, which is most of the cases. In the following screenshots it is possible to have a quick overview of the main steps during the viewpoints’ process creation.

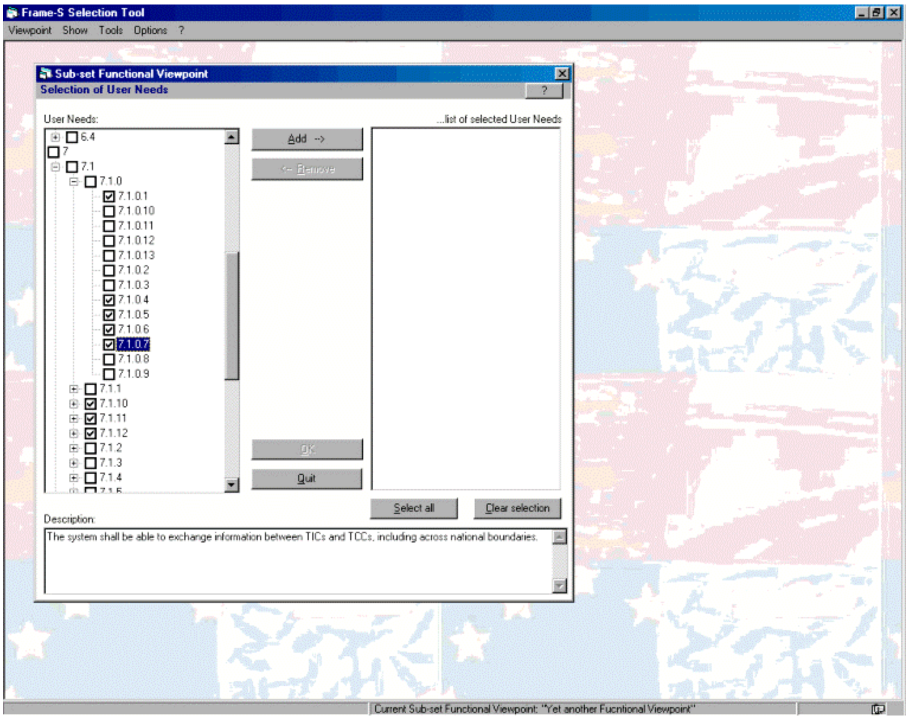

Figure 18: FRAME Selection Tool: user needs selection **[11]**.

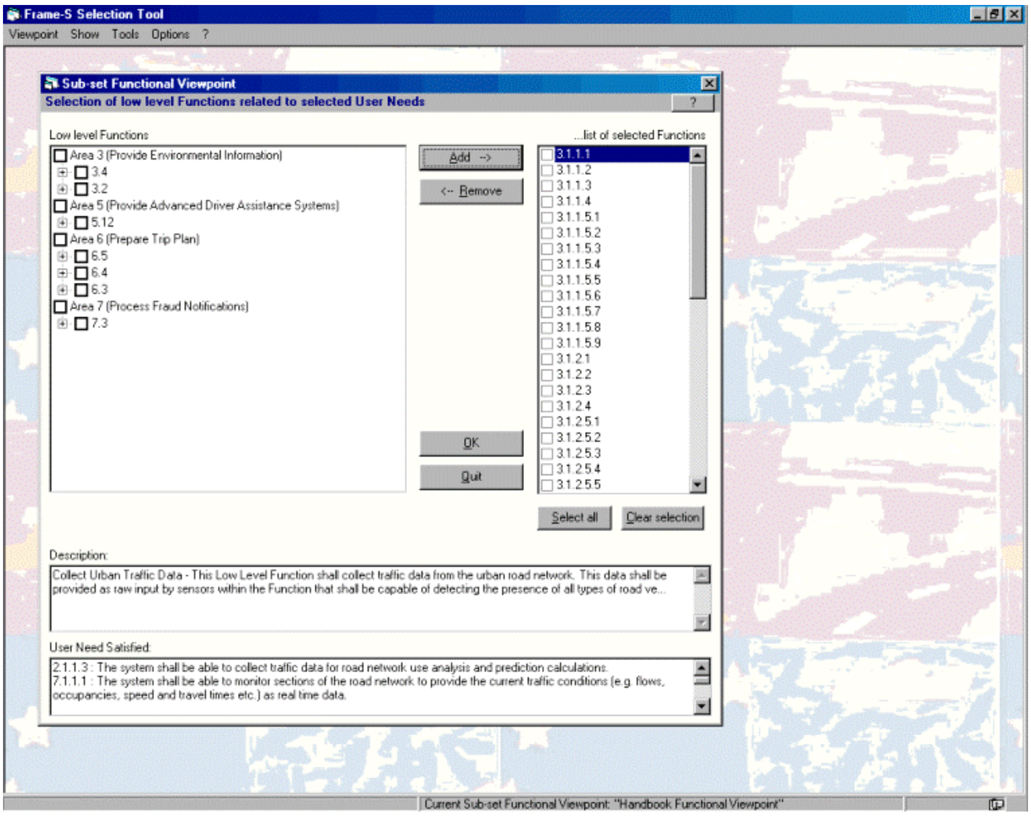

Figure 19: FRAME Selection Tool: low-level functions selection **[11]**.

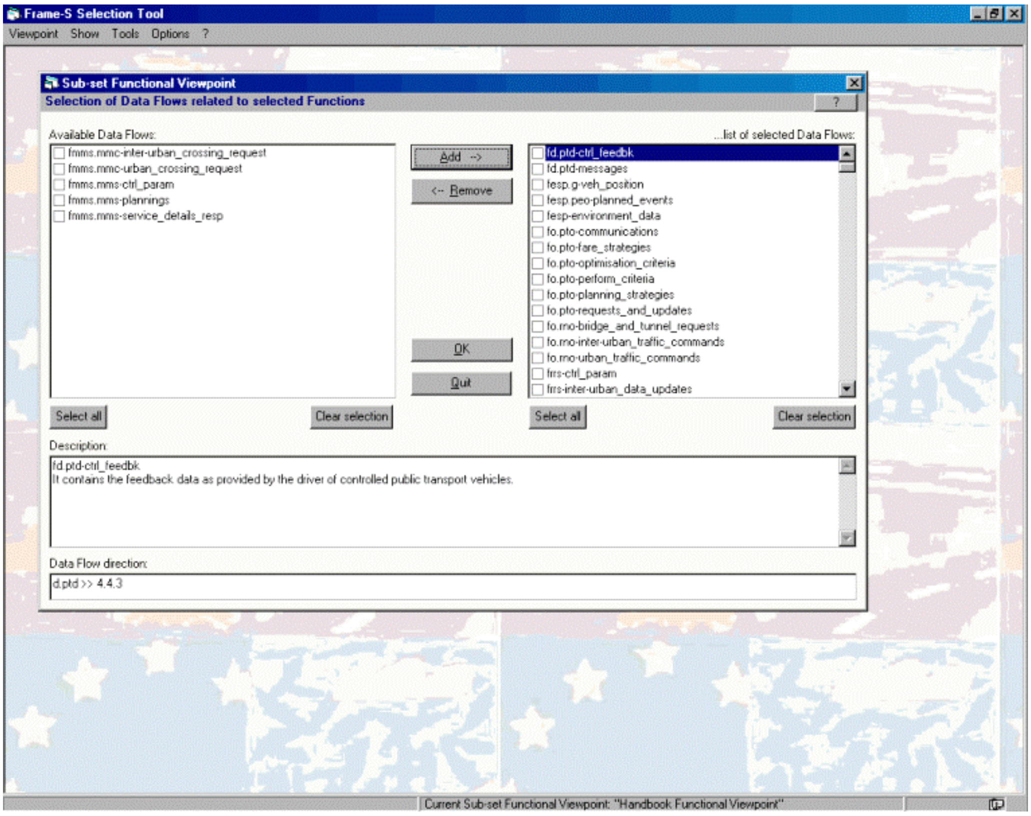

Figure 20: FRAME Selection Tool: data flows selection **[11]**.

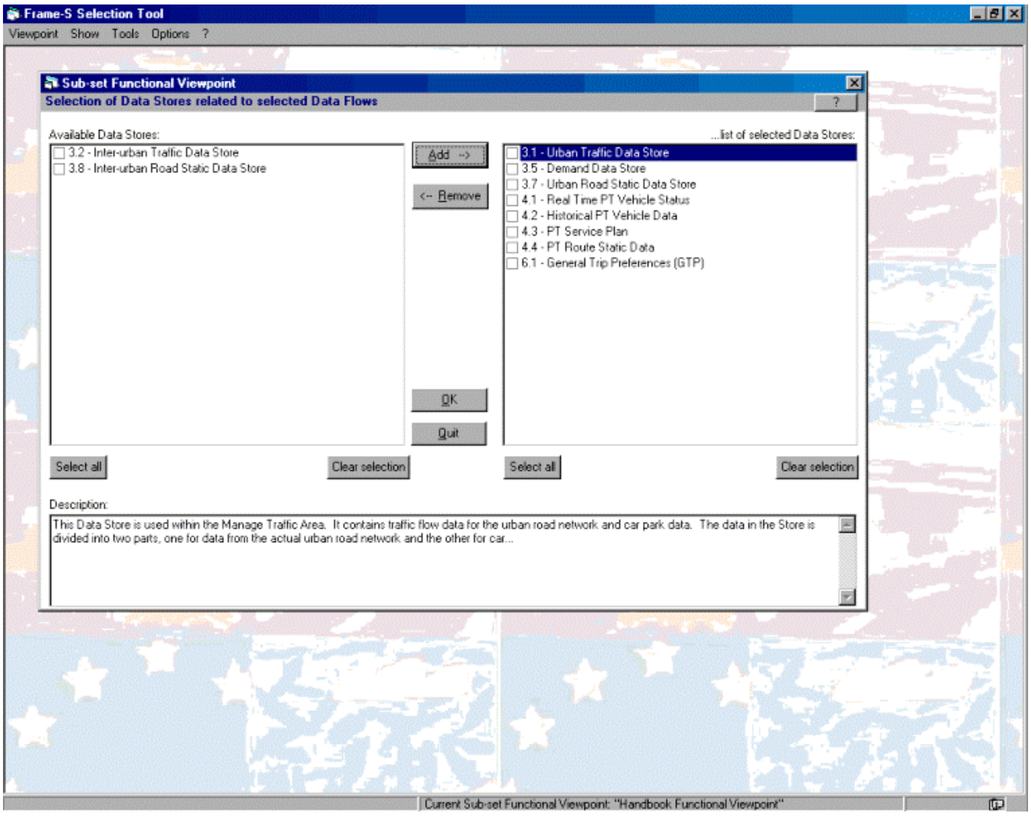

Figure 21: FRAME Selection Tool: terminators and actors selection **[11]**.

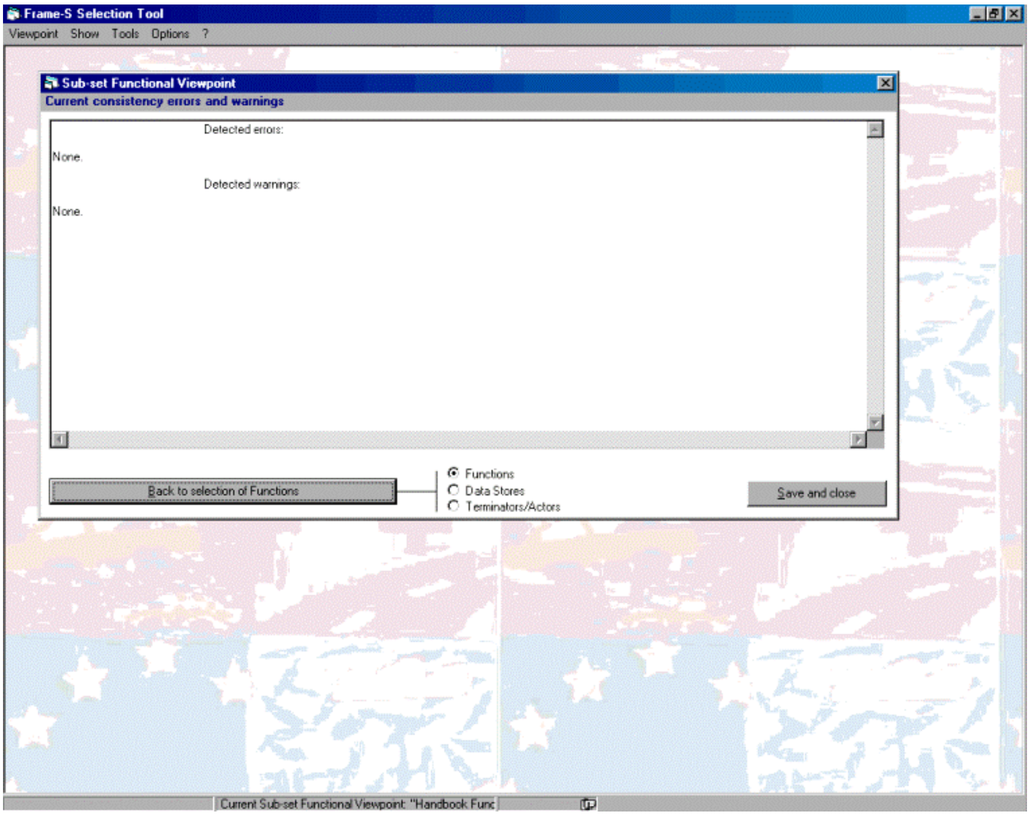

Figure 22: FRAME Selection Tool: functional viewpoint consistency check **[11]**.

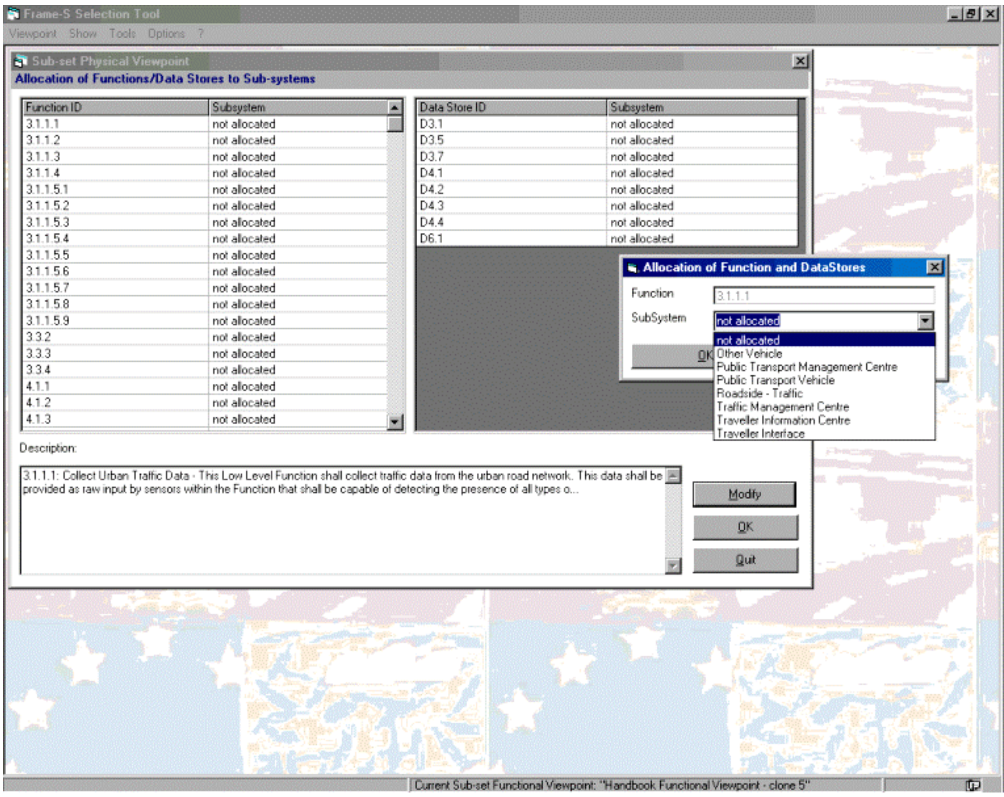

Figure 23: FRAME Selection Tool: mapping of low-level functions to sub-systems within the Physical Viewpoint **[11]**.

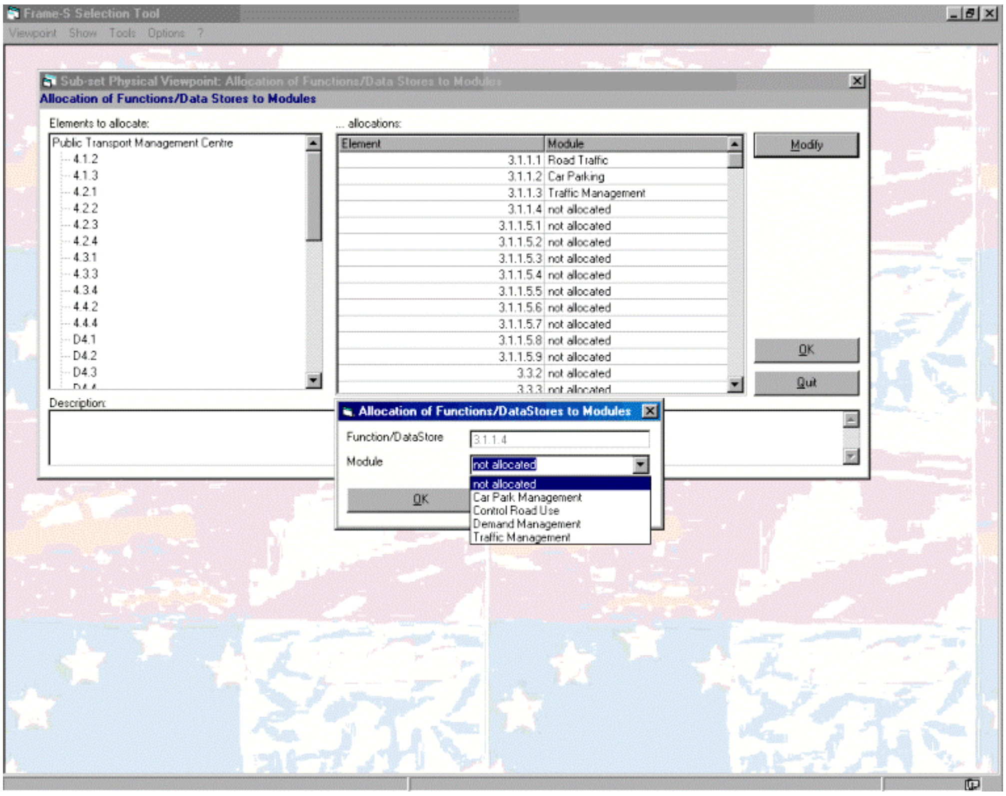

Figure 24: FRAME Selection Tool: mapping of low-level functions and data stores to modules within the Physical Viewpoint **[11]**.

2.4 The INTEGREEN FRAME architecture

In INTEGREEN, the work of “FRAME mapping” has been quite simple in light of the relevant work already done during the requirements definition process. Apart of the “internal” added value of using this instrument in the design process just before the detailed design process, INTEGREEN consortium immediately realize the potential outcomes from having a FRAME-compliant functional and physical viewpoint, which can be used not-only by third parties as a starting point for similar implementation in other urban areas, but also by the consortium itself (and in particular for the Municipality of Bolzano) in the perspective of further extending this reference architecture, in particular in direction of cooperative systems scenarios. Indeed, it is important to notice that the functionalities covered by the INTEGREEN FRAME-architecture covered a domain which is broader than the one depicted during the requirements process, because of the intention to maintain a generalization approach that is more suitable with respect to these indicated purposes.

As already stated, one of the main limitation of FRAME is in terms of outputs availability. In INTEGREEN partners have made some additional reports to “post-process” the raw outputs made available by the Selection Tool in order to present the INTEGREEN viewpoints in a more clear way, and trying to put in evidence the key elements of the entire architecture (e.g. the environmental dimension). All these post-processed outputs are available in form of a set of annexes which are a natural extension of these deliverable.

3 Environmental Supervisor Centre high-level design

The definition of how to effectively implement the proposed Environmental Supervisor Centre illustrated in Figure 1 has been one of the key initial design elements, since this determines all subsequent design choices presented below. The objective of this design task has been to find a viable solution that could fit with the project goals and constraints and at the same time is able to not put into any risk the current traffic management activities carried out within the Traffic Control Centre of Bolzano [4].

3.1 Reference requirements

Many high- and low-level criteria have been considered during this crucial initial decision-making process. The most important ones can be summarized as follows:

- the effective possibility to implement and introduce the **functionalities of the entire INTEGREEN system**;
- the minimization of the overall **disturbance to daily traffic control activities** carried out by the traffic operators at the Traffic Control Centre of Bolzano, and the possibility to involve them into the project activities in a very smooth way;
- the minimization of **time and effort resources** needed from a system integration point of view;
- the maximization of the **overall system security**;
- the **possibility to easily exploit the proposed architecture** towards novel monitoring technologies and use cases and scenarios, thus ensuring the capability to smoothly include the most consolidated outputs coming from this implementation within the daily traffic management activities while maintaining a reference cloud area where research and development activities can continue in cooperation with the local industry and research organizations.

Based on these reference criteria, three different integration scenarios, presenting different advantages and disadvantages, have been therefore proposed. The details of each of one are briefly presented in the next three paragraphs.

3.2 Integration Scenario 1: complete “physical” integration

In the first integration scenario proposed, the Environmental Supervisor Centre is completely integrated within the actual Traffic Control Centre (**Fehler! Verweisquelle konnte nicht gefunden werden.**). All the fixed and mobile stations send their collected measurements to this central point on a real-time basis, where they are stored and dynamically processed in order to determine an actual estimation of the current traffic and air pollution conditions, which the traffic operators can continuously check. The most relevant information are directly distributed from this centre to interested service providers through the Internet.

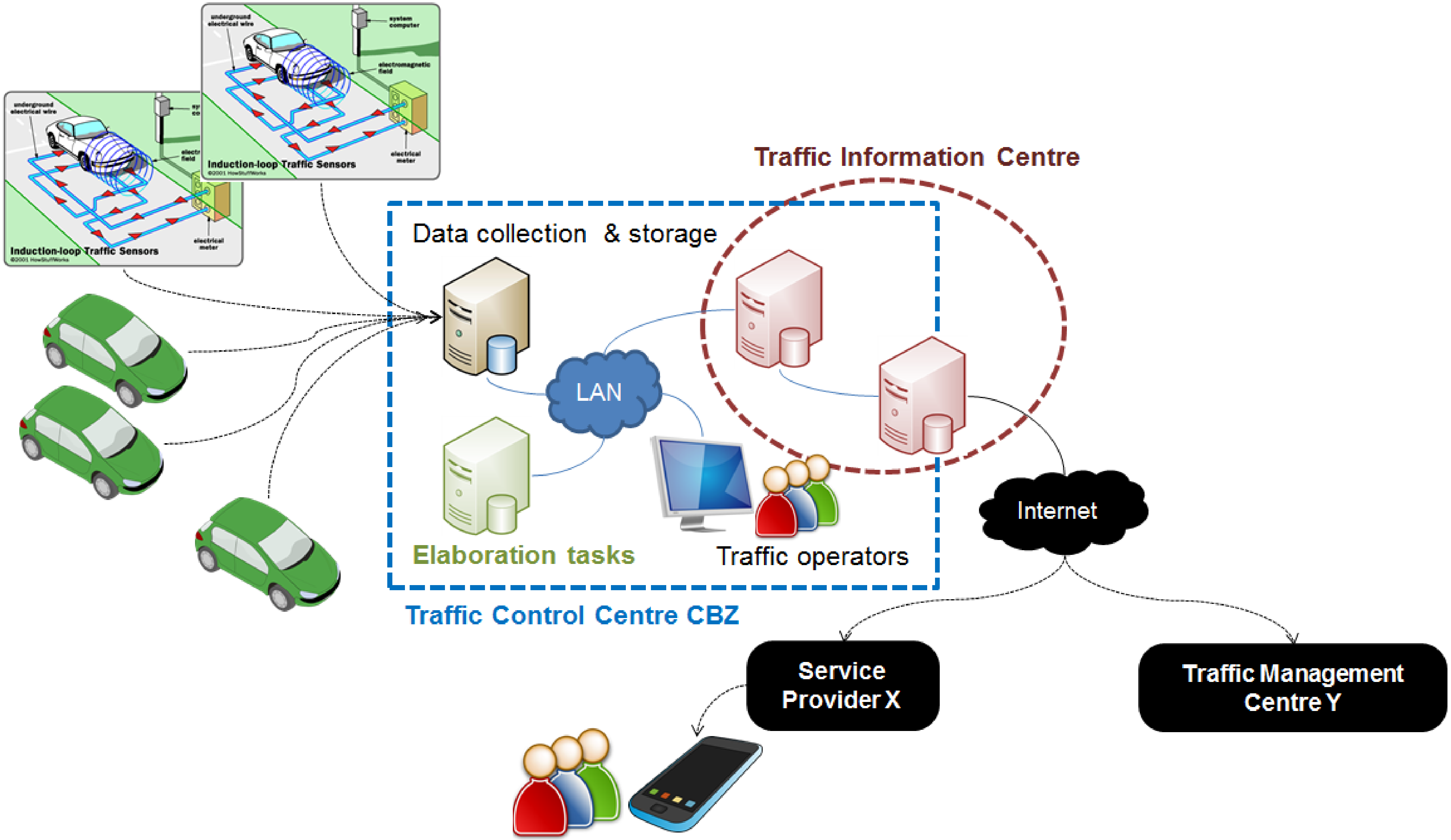

Figure 25: Integration Scenario 1: Environmental Supervisor Centre completely integrated in the Traffic Control Centre.

3.3 Integration Scenario 2: partial "physical" integration

In the second integration scenario proposed, the Environmental Supervisor Centre is integrated within the actual Traffic Control Centre but the Traffic Information Centre responsible for the distribution of real-time traffic information to external providers is located outside it (Figure 25). It is actually the same as proposed in the integration scenario n.1, with the difference that the distribution point for the delivery of the real-time information is located outside the Traffic Control Centre. In order to accomplish this, a dedicated and secure virtual channel between the Traffic Control Centre and the Traffic Information Centre must be foreseen.

3.4 Integration Scenario 3: "distributed" integration

Finally, in the third integration scenario proposed, the Environmental Supervisor Centre is located outside to the Traffic Control Centre, but represents an external expansion of it (Figure 26). In this scenario, the operations of the Traffic Control Centre are not directly altered; only the additional roadside installations are integrated within the actual monitoring system. The entire Environmental Supervisor Centre platform (included the Traffic Information Centre) is implemented in a completely external location, but with a dedicated connection with the Traffic Control Centre. This connection is logically bidirectional, since there is not only the necessity to transfer on a real-time basis the data gathered on the field, but also to give access to all the traffic / air pollution elaboration outputs, as well as other additional data streams coming from novel sources (e.g. the mobile stations) to the traffic operators.

3.5 Comparison between proposed integration scenarios

Table 1 presents an evaluation matrix between reference evaluation criteria and proposed integration scenarios.

Evaluation criteria	Integration scenario n.1	Integration scenario n.2	Integration scenario n.3
INTEGREEN functionalities	**High**. The proposed scenario guarantees the proper implementation of the proposed functionalities of the INTEGREEN system.	**High**. The proposed scenario guarantees the proper implementation of the proposed functionalities of the INTEGREEN system.	**High**. The proposed scenario guarantees the proper implementation of the proposed functionalities of the INTEGREEN system.
Disturbance to daily traffic control activities	**High**. The physical integration of the novel components proposed in INTEGREEN can provide significant inconvenient situations to traffic operators' staff.	**Medium**. The partial physical integration of the novel components proposed in INTEGREEN (elaboration tasks only, with novel measurement stations) can limit the number of inconvenient situations to traffic	**Low**. Being no physical integration at the Traffic Control Centre, there will be no inconvenient situations for traffic operators' staff.

		operators' staff.	
Time and effort resources	**Medium**. There is a non-negligible effort foreseen for this possible integration process, but at least the integration process takes place in one single physical place.	**High**. There is the need to integrate most of the INTEGREEN components in the Traffic Control Centre and to find a virtual space where to host the Traffic Information Centre.	**Low**. The implementation process of the prototype Environmental Supervisor Centre can start from scratch, and therefore most of the integration issues can avoided from the beginning through this approach. The only little overhead is to develop data sources components for the real-time collection of data from the monitoring stations that are managed by the Traffic Control Centre. This scenario presents also the advantage that successive releases of the entire platform can be very quickly introduced.
Overall system security	**High**. Being INTEGREEN fully integrated in the physical Traffic Control Centre, the security risk is clearly minimized.	**Medium**. In this hybrid situation the security risk can be considered medium since the data flow between the Traffic Control Centre and the external network is maintained at reasonable low levels.	**Low**. This integration scenarios presents the highest number of interactions between the Traffic Control Centre and the external network, and therefore is the one that is more exposed to security risks.
System exploitation	**Low**. There is little possibility to increase the functional capabilities of the Traffic Control Centre.	**Medium**. There is the possibility to easily exploit the Traffic Information Centre, while it is more difficult to expand the functionalities of the Traffic Control Centre.	**High**. This scenario offers the best exploitation opportunities.

Table 1: High-level design of Environmental Supervisor Centre: integration scenarios evaluation matrix.

Based on these overall considerations, the decision has been to move in the direction of integration scenario n.3. The price for this choice is likely to be in the security field, but proper technical countermeasures can be adopted in order to properly manage this issue. The advantage of this scenario is in particular in its enormous exploitation capabilities, which can take advantage of some of the strength points of other integration scenarios, as presented in Figure 27. The future perspective is to step by step move into the Traffic Control Centre the most robust and cost-effective experimental solutions, which have given proper evidence of their added value, and let it migrate to a real *Traffic Management Centre*. The Environmental Supervisor Centre could evolve to a very advanced open ITS research & development platform, where very advanced elaborations can be studied and assessed in cooperation with the local industry and the research organizations. An experimental Traffic Information Centre can be maintained so that advanced traveler information services can be offered to local travelers. This proposed exploitation scenario can open the doors in the future to very ambitious Field Operational Tests, covering for example vehicle-to-X (V2X) communication-based cooperative systems.

3.6 Environmental Supervisor Centre implementation strategy

The prototype version of the Environmental Supervisor Centre is going to be hosted, at least for the life time of INTEGREEN, at the premises of Associated Beneficiary n.2 TIS. The Free Software & Open Technologies of TIS, i.e. the internal centre of TIS which is involved in the project, has in fact matured in the last years a broad experience in the building and management of server infrastructures for the development of complex demonstrative projects. This virtual workspace represents typically the ideal environment where to start and then consolidate similar R&D works, in cooperation with other involved parties through open innovation approaches. Once a prototype has demonstrated to be sufficiently stable through testing and validation activities, the strategy of TIS is to make a step backward and let third parties (typically private companies that have been involved in such projects from an early stage) the task to host such server infrastructures in "production" mode, following specific business models. Activities more oriented to R&D can then continue on top of these experimental platforms. This kind of perspective fits perfectly with the needs of INTEGREEN, and its possible future exploitation scenarios; the involvement of external subcontractors will therefore play an important role also from this perspective.

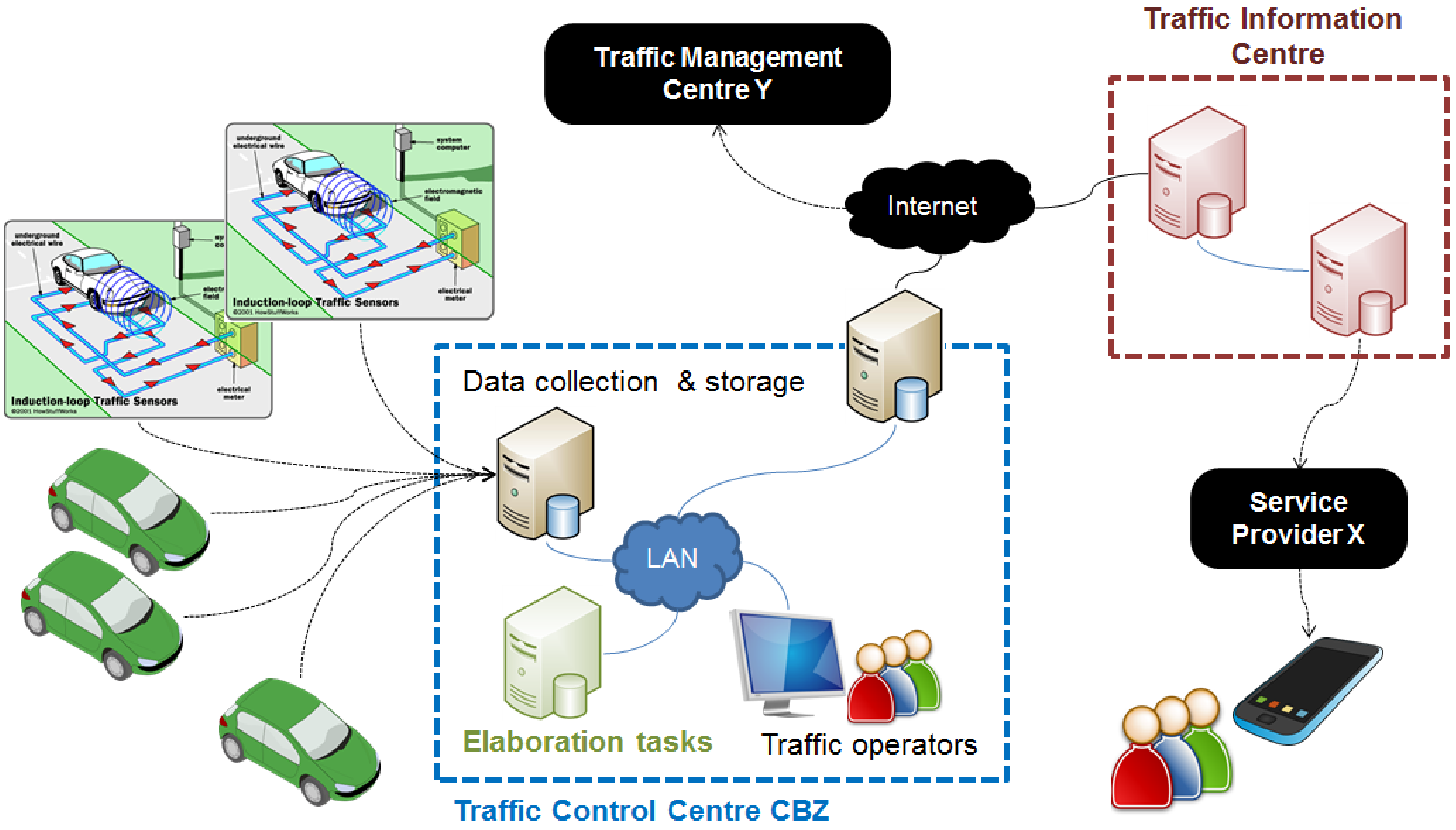

Figure 25: Integration Scenario 2: Environmental Supervisor Centre integrated in the Traffic Control Centre but with external Traffic Information Centre.

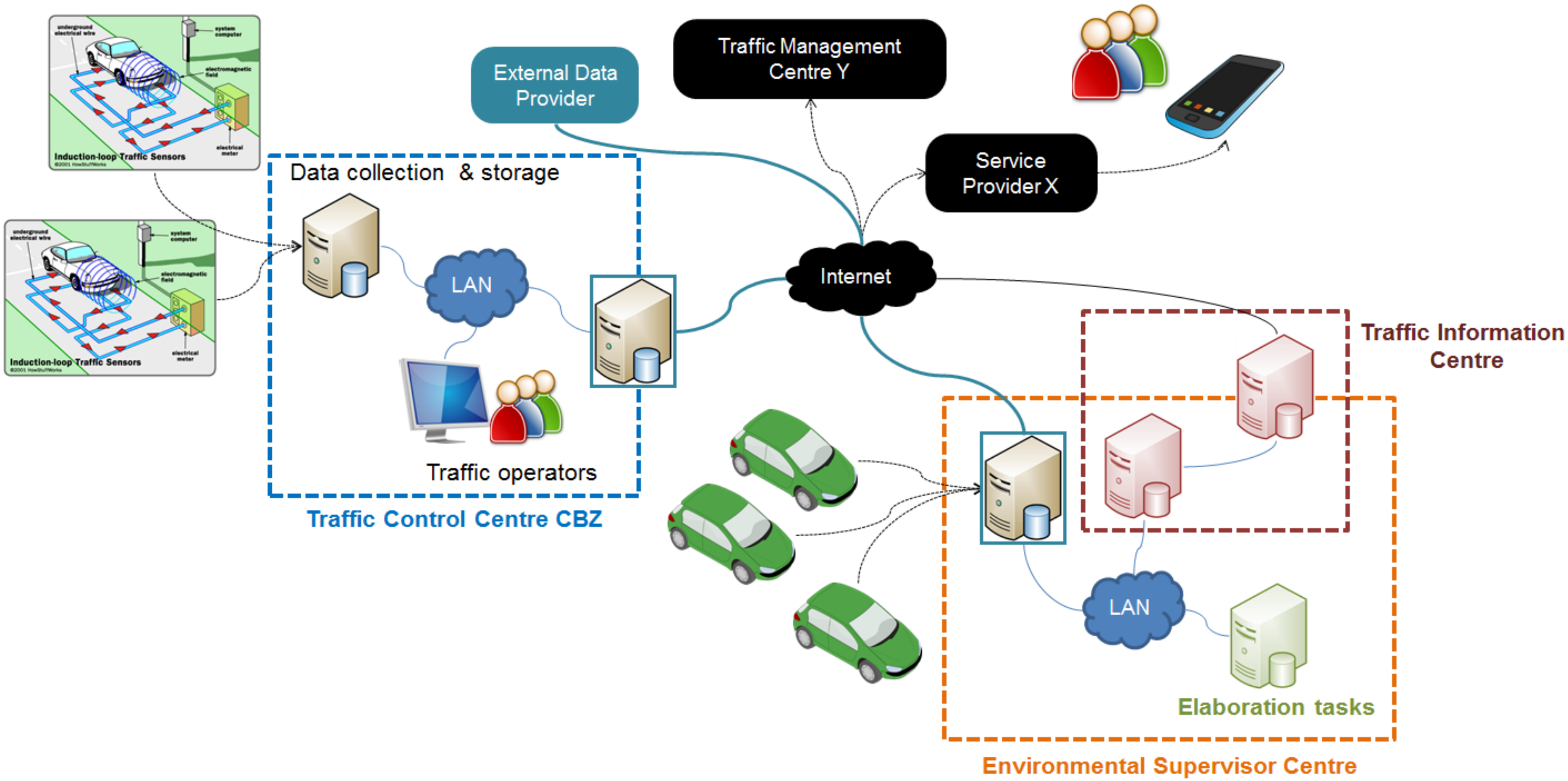

Figure 26: Integration Scenario 3: Environmental Supervisor Centre as external expansion of the Traffic Control Centre.

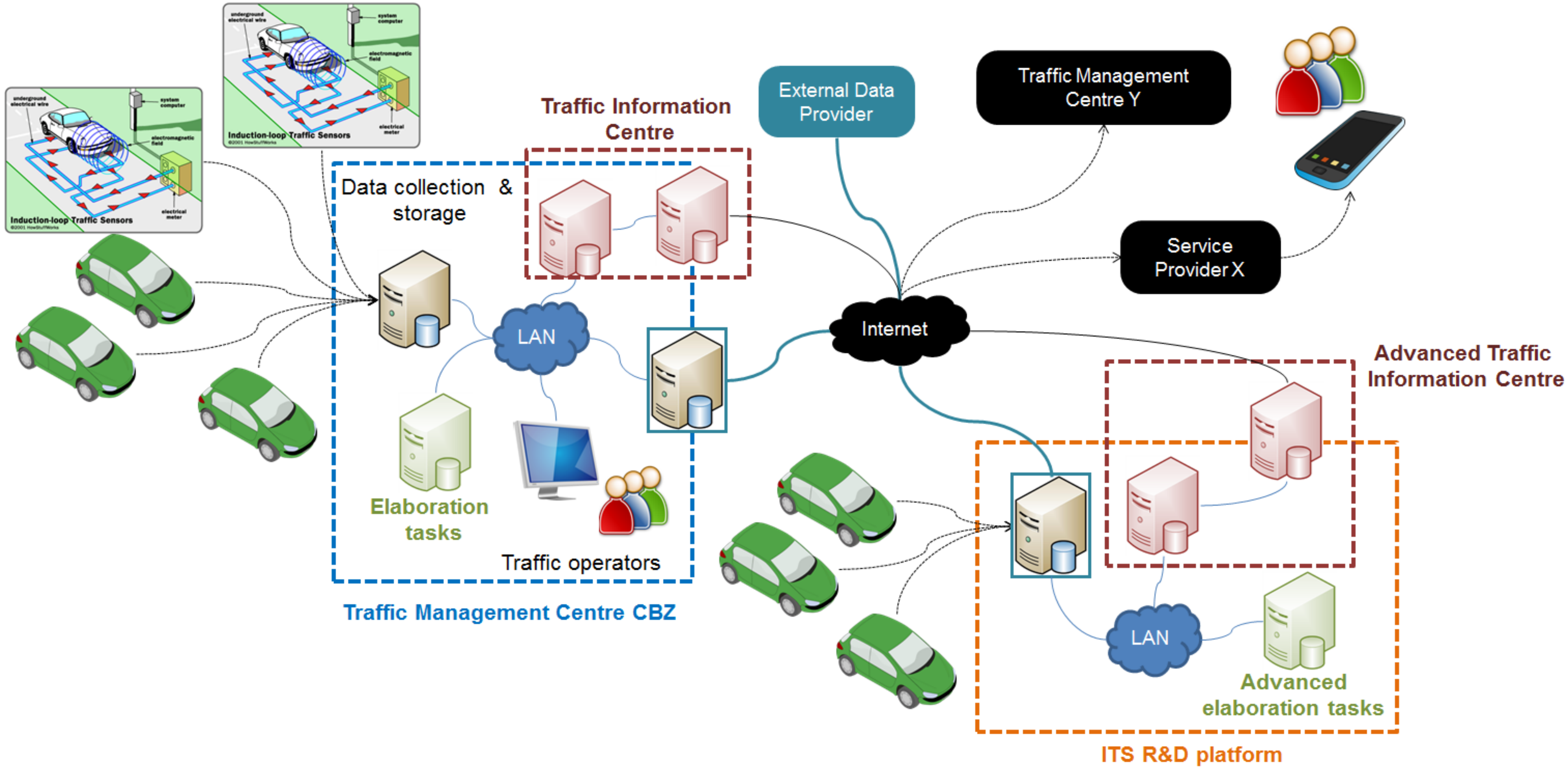

Figure 27: Possible exploitation scenario based on integration scenario 3.

4 Data sources layer design

4.1 DATEX II: the reference data transfer technology

The compatibility with reference ITS standards is a central element within the INTEGREEN system, and this for a twofold reason. On one side, there is the possibility to re-use basic software components, in particular for the data transfer interfaces, which gives as a consequence the possibility to significantly reduce the implementation effort. Second, in particular as far as the front-end layer of the architecture is concerned, the adoption of standard technology ensures that the system can be easily exploitable, and interfaced with third parties (e.g. service providers who want to build real-time travel information services to end-users). In fact, in a future scenario, those service providers may be able to have access to multiple supervisor centers covering different road networks domain and thus have the possibility to provide harmonized and non-interrupted services among all Europe.

When one thinks at data transfer standards in the ITS domain, the immediate association goes to **DATEX II** technology, which was introduced and is now managed in the scope of **EasyWay**, the trans-European investment programme managed by the European Commission DG MOVE for the harmonization of ITS services in Europe. An important work of standardization has been carried out in the past years, in partnership with CEN. In particular, DATEX II series standards are continuously developed and maintained by the CEN Technical Committee 278 (**CEN TC/278** “Road Transport and Telematics”), which works in strict cooperation with **ETSI TC ITS** (responsible for the standardization of the enabling information and communication technologies for the ITS domain) and **ISO TC 204**, which deals with the standardization harmonization at a global level (Figure 28).

The first three parts of the DATEX II CEN standard (**CEN 16157**), finalized in 2011, are already published as Technical Specifications. These three parts deal with the most mature and widely used parts of DATEX II: the modelling methodology (*context and framework*) as **Part 1**, *location referencing* as **Part 2** and the most widely used DATEX publication for traffic information messages (called *situation publication*) as **Part 3.** A fourth and fifth Part of CEN DATEX II series, VMS and measured and elaborated data publications (respectively), have been already approved and are waiting to be published as well. This set of standards are all based on release 2.0 of DATEX II, even if a version 2.2 is already available.

DATEX II technology is primarily focused on two main reference scenarios: the exchange of real-time travel information among (i) <u>two traffic management centers</u> and (ii) <u>a traffic management centre and a service provider</u>. In the latter scenario, the idea is to combine DATEX II with **TPEG** (*Transport Protocol Expert Group*), the standards proposed by the *Traveller Information Services Association* (**TISA**) as evolution of the RDS-TMC technology for the standardized delivery of end-user services on broadcast and IP channels [12]. In terms of exchanged data, the covered domains are:

- road and **traffic-related events**;
- **operator actions** (e.g. roadworks);

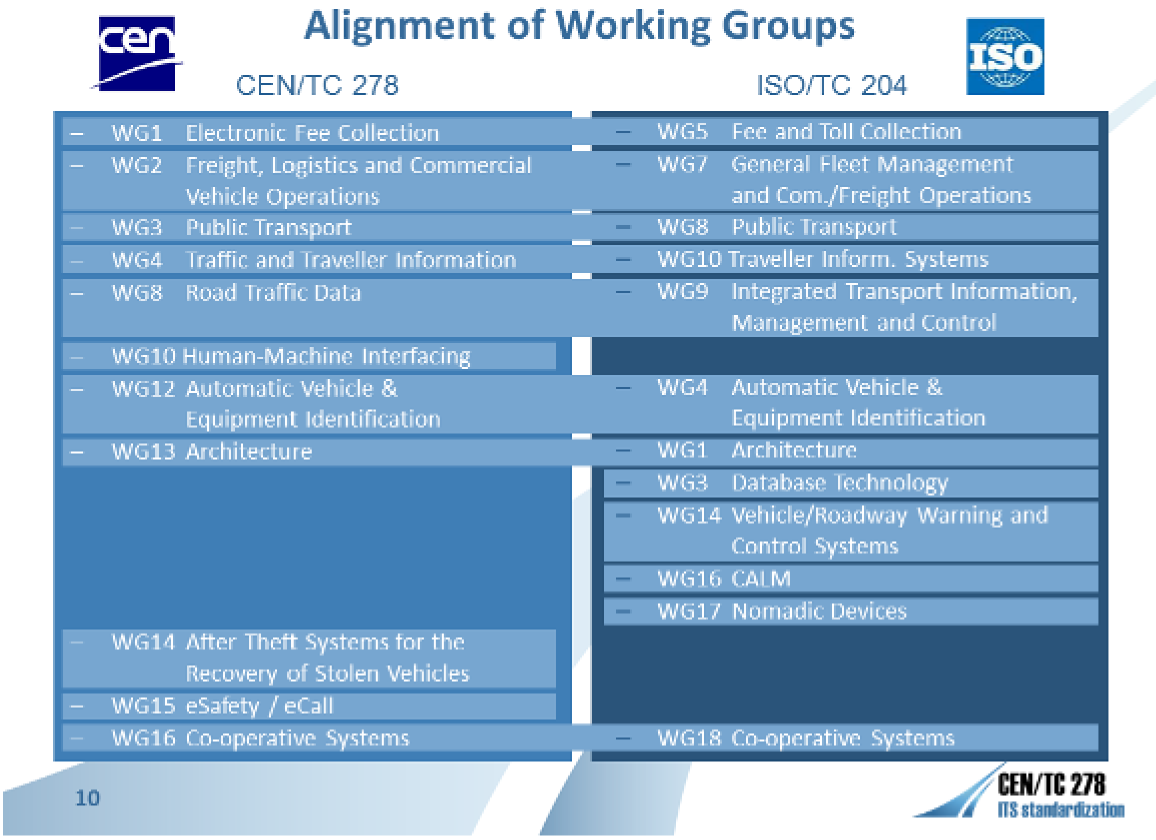

Figure 28: The relationship between CEN and ISO in the ITS standardization domain (Source: www.itsstandards.eu).

- **non-road event information** (including multi-modal information);
- **elaborated data** (e.g. travel times, traffic status);
- **measured data** (e.g. traffic or weather measurements);
- messages displayed on the **Variable Message Signs**.

4.1.1 Overview of data package exchanged in DATEX II

A more detailed overview of the data package supported by DATEX II data exchange technology is given in Table 2.

Data class	Categories
Road and traffic-related events	• **Abnormal traffic** (e.g. long queues, stop&go situations, etc.) • **Accidents** • **Obstructions** (i.e.: animal presence / vehicle presence / obstructions due to environment (avalanches, flooding, fallen trees, rock falls, etc.) / obstructions due to infrastructure (fallen power cables, etc.) / other obstructions including people

	• **Activities** (e.g. public event, disturbance, etc.) • **Incident on equipment or system** (e.g. Variable Message Sign out of order, tunnel ventilation not working, emergency telephone not working, etc.) • **Conditions**: driving conditions related to weather (e.g. ice, snow, etc.) or not (e.g. oil, etc.), related to environment (e.g. precipitation, wind, etc.).
Operator actions	• **Network management** (e.g. road closure, alternate traffic, contraflow, etc.) • **Traffic control** (e.g. rerouting, temporary limits, etc.) • **Roadworks** (e.g. resurfacing, salting, grass cutting, etc.) • **Roadside assistance** (e.g. vehicle repair, helicopter rescue, food delivery, etc.) • **Sign settings** (i.e. VMS messages)
Non-road event information	• Events that do not directly take place on the road (e.g. transit service information, road operator service disruption, car park information)
Elaborated data	• **Travel times** (e.g. free-flow travel times, instantaneous travel time, etc.) • **Traffic status** (attribute with five values: free flow, heavy, congested, impossible, unknown) • **Traffic values** (i.e. traffic measurements aggregated on periodic basis: flow, speed, headway, concentration and individual vehicle measurements) • **Weather values** (i.e. environmental measurements aggregated on periodic basis: precipitation, wind, temperature, pollution, road surface condition and visibility)
Measured data	• **Traffic values** (flow, speed, headway, concentration and individual vehicle measurements) • **Weather values** (precipitation, wind, temperature, pollution, road surface condition and visibility) • **Travel times** (i.e. direct measurements from outstation values: elaborated time, free flow time, normally expected time) • **Traffic status** (i.e. direct measurements from outstation values: attribute with five values: free flow, heavy, congested, impossible, unknown)
VMS messages	• **VMS messages** (including textual messages and pictograms), including information about equipment status and position.

Table 2: Complete data package that can be exchanged over DATEX II.

4.1.2 DATEX II data models

Three alternative data models are supported by DATEX:

- **Level A "*core model*",** typically able to cover most of the data exchange scenarios, which the users can use to assemble a specific data publication need. This level is considered the minimum set that all DATEX II must fulfil in order to assure the interoperability.
- **Level B "*extended core model*"**, which is an instrument to extend the basic data

dictionary of level A. In fact, specific implementations can have particular requirements which cannot be completely fulfilled through the minimum standard level. Thanks to level B, users can extend the data dictionary as needed, but in order to ensure the maximum backward interoperability with level A implementations. This means that an authority working at level A must be able to accept and correctly decode a publication sent by a provider using a level B implementation.

- **Level C**, which include all those implementation that are DATEX II "concept-compliant" (i.e. in terms of modelling rules and exchange protocols) but not compliant with level A/B content models.

It is also worth noting that for two parties the level A could also be too much for their actual needs. In this case, it is possible to create a so-called "**profile**", which contain just the classes that are needed for implementing the data exchange service.

4.1.3 DATEX II exchange mechanisms

DATEX II standard has been defined with the idea to introduce a functional model which parties could easily refer to, independently from the technologies and software environment in which a technical implementation of web-service can be carried out. The only (obvious) condition is that two actors must be linked through the Internet.

The functional model is specified in **UML** (*Unified Modelling Language*), which is quite a standardized and common language in the software engineering domain. The UML-based model can be easily translatable in a **XSD / XML** (*eXtensible Markup Language*) schema which is needed to produce XML exchange files. So, even if XML is not the only possible "model specification", this is typically the standard approach in which DATEX II is implemented, in conjunction with a **SOAP** (*Simple Object Access Protocol*) running over **HTTP**.

Exchange mechanism can following either a **pull mode** or a **push mode**. In the first case, data is requested by a client from a provider, typically on a periodically basis; on the contrary, in the second case, the server (or "supplier") sends to the client the information package when a change is available ("*push on occurrence*") or on a periodic basis. The "push on occurrence" mode is typically chosen in scenarios in which the client needs to get the information as soon as possible.

The exchange system relies on a **publish / subscribe** pattern, in which the supplier publishes the information, whereas the client subscribes it in order to receive the traffic information. The supplier exchange system is composed by two main subsystems: (i) the publisher subsystem, which makes data available and creates the payloads publications (e.g. situations, traffic view, measured and elaborated data, locations); and (ii) the delivery subsystem, which adds specific exchange information and performs the physical data. A the client side, there is only a receiver subsystem which must be able to catch the data through a pull or push approach (Figure 29).

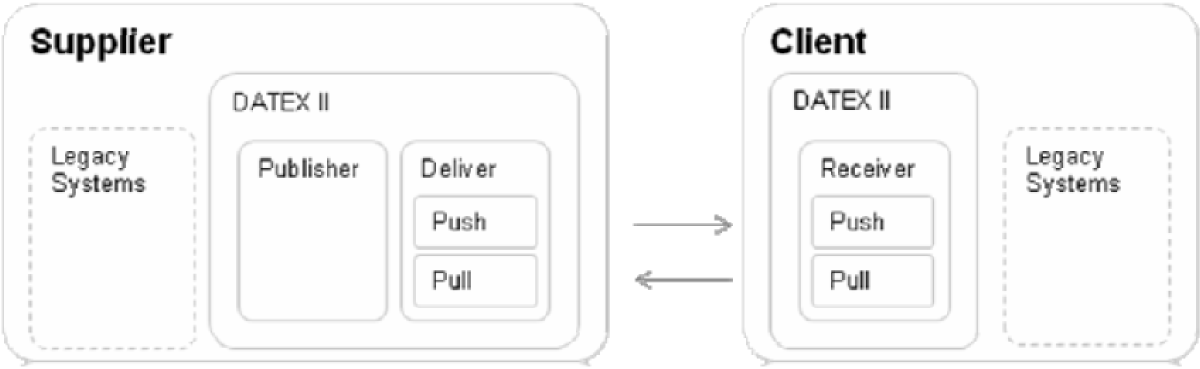

Figure 29: The reference architecture for a DATEX II data exchange interface **[13]**.

The subscription is an important part of the overall exchange system, since it is a mechanism with which client and supplier specify the payload type to be exchanged. In the specification, there is no particular restriction on how this must be implemented; the payload can in fact be defined either by the client or the supplier, introducing all necessary refinements.

4.1.4 DATEX II levels of service

DATEX II is a standard which aims to create the possibility to enable harmonized services among Europe not only in terms of content and functionality, as already discussed, but also in terms of *availability*. In fact, a road user would expect to have certain services to be offered in specific road environments, depending on its particular nature. In order to address this, EasyWay has defined eighteen so-called "***Operating Environments***" (OE), i.e. pre-defined road area types classified in terms of (i) physical characteristics, (ii) network topology and (iii) traffic characteristics [14]. These OEs are then matched with a set of reference service levels, which are defined in the perspective of the road operator wanting to publish information related to the road network that he's controlling (Figure 31). Three services levels are defined, according to the information detail which is published (Figure 30).

In the target scenario of Bolzano, the associated OE is N1, i.e. "*road corridor or network, at most seasonal flow-related traffic impact, possibly safety concerns*", which suggests:

- a service level "1" for the publication of road event information and travel times;
- a service level "1" for minimum implementation scenarios, and a service level "3" for optimum implementation scenarios concerning the publication of measured data and traffic status;
- a service level "1" for minimum implementation scenarios, and a service level "2" for optimum implementation scenarios concerning the publication of VMSs messages;

LEVELS OF SERVICE: DATEX II				
Criteria Nature of service	Level 0 (no service)	Level 1[7]	Level 2	Level 3
Road event information	None	Minimum set of data relative to safety as mentioned in the ITS Directive[8]	Road and traffic related events with operator actions	Non-road event information linked with multimodal information
Travel Time	None	Short-distance travel time like along ring road	Long-distance travel time including forecast	
Measured data and traffic status	None	Traffic data and/or traffic status	Weather data	Both
Messages displayed on VMS	None	Availability of the messages linked with safety	Availability of road events messages	Availability of all messages displayed

Figure 30: The levels of services defined in DATEX II **[13]**.

4.1.5 Limitations of DATEX II

DATEX II is an ideal instrument for the data transfer between traffic management centers, and in fact most of the applications respond to this kind of use cases. This technology , being not conceived to cover all possible ITS domains and applications, presents some significant limitations that typically do not allow to fulfill the entire set of requirements of a complex ITS implementation, like the one which is expected to be achieved in INTEGREEN. The most relevant ones are mainly two:

- not all the data that be handled in an ITS system are covered by DATEX II; for example, public transportation information is covered in only a very little way;
- DATEX II does not natively support OGC services for the standard geo-publication of

Name of Service			EasyWay OPERATING ENVIRONMENT																	
Criteria for the Levels of Service [reference <DG spec>]			C1	T1	T2	T3	T4	R1	R2	R3	R4	R5	R6	R7	R8	S1	S2	N1	N2	P1
Road event information	C	Level 3																		O
	B	Level 2	O								O				O					
	A	Level 1	M	O	M	M	M	O	M	O	M	O	M	O	M	M	M	M	M	M
	/	*Service unavailable*																		
Travel Time	C	Level 3																		
	B	Level 2														M	M	M		
	A	Level 1				M	M			M	M			M	M				M	M
	/	*Service unavailable*	/	/	/			/	/			/	/							
Measured data and traffic status	C	Level 3														O		O		
	B	Level 2																		
	A	Level 1	O		M	M	M		M	M	M		M	M	M	M	M	M	M	M
	/	*Service unavailable*		/				/				/								
Messages displayed on VMS	C	Level 3	O																	O
	B	Level 2			O		O		O		O		O		O	O	O	O	O	
	A	Level 1	M		M	M	M		M	M	M		M	M	M	M	M	M	M	M
	/	*Service unavailable*		/				/				/								

Recommendations for LoS per OE: M Minimum LoS recommended; O Optimum LoS recommended

Figure 31: The Operating Environments and Levels of Services matrix of DATEX II **[13]**.

information. This means that a service provider must re-arrange the data provided through a DATEX II service in order to offer a more user-friendly map-based service to travelers.

In order to face these problems, there have been a series of EU projects funded under both the umbrella of the 6^{th} Framework Programme for Research and Technological Development (FP6) as well as the Competitiveness and Innovation Framework (CIP), which have proposed an harmonized approach based on the variety of existing ITS standards for the standard geo-publication of real-time traffic and travel information, according to the INSPIRE directive as well [15]. The design choices made at the data center layer have tried to take in active account the results of these projects in order to maximize the actual (and future) system capabilities, and exploit the freely available tools made available by these international initiatives which are going to be presented more in detail in the following chapter.

In INTEGREEN, DATEX II can represent the right choice for properly implementing most of the functionalities of the 3^{rd} parties data source and front-end, which goes exactly in the direction of providing a bi-directional interface with other traffic (and not only) control / information centers. On the contrary, as far as the connection with the system components located at the Traffic Control Centre of the Municipality of Bolzano, in light also of the perspective given in the above paragraphs, the strategy is to introduce, in cooperation with the private company charged for this task (Famas System S.p.A), very simple “internally-specified” web services which are able to provide, in a very simplified way, access to all available measured data. On the other side, the interfaces managed by the data sources gathering data from external services providers (e.g. the Environmental Data Source collecting the air pollution measurements taken by the Local Agency for the Environment) have to follow the custom specifications exposed by these third parties, which however must be tuned in order to satisfy the INTEGREEN constraints.

An overview of the specific design choices made for each data source is given in the next paragraphs. Where relevant, a complete description of the roadside units installations and capabilities is given, too.

4.2 Traffic Station Data-Source

The Traffic Station Data Source is the software component at the Environmental Supervisor Centre which is responsible to gather on a real-time basis the data gathered by the fixed traffic stations installed within the road network of interest. For completeness sake, the tables below illustrate again the main specific requirements defined in the previous Action n.2 for this data source.

ID	TSDS_1
Name	Data type
Description	Each record of generated data must contain the following fields: • number of vehicles counted within the window observation; • speed profile (i.e. all vehicles shall be classified in an aggregated way according to their speed to a speed class of a maximum interval of 10 [km/h], with lower boundary class 0-15 [km/h] and upper boundary class all speeds higher than 70 [km/h]);

	● vehicle category (according to the "9+1"- classes Italian standard classification); ● travel direction.
Rationale	Basic parameters for determining the traffic status according to the reference theories available in the literature.
Type	Functional
Priority	Must

Table 3: Traffic Station Data-Source: requirement TSDS_6 (data type).

ID	TSDS_2
Name	Data type (optional)
Description	Each record of generated data should contain the following additional fields related to each single vehicular transit: ● gap; ● headway; ● lane (in the case of roads with multiple lanes in the same direction); ● sensor occupation time (i.e. the total amount of time in which a vehicle is in the monitoring field of the traffic sensor); ● vehicle length.
Rationale	Parameters which can be considered not mandatory but which can increase the level of analysis and elaboration of the generated data.
Type	Functional
Priority	Should

Table 4: Traffic Station Data-Source: requirement TSDS_2 (data type (optional)).

ID	TSDS_3
Name	Data frequency update
Description	The maximum time interval between two consecutive generated data packets delivered to the traffic station data-source must be 5 minutes.
Rationale	Guarantee minimum system performance
Type	Performance
Priority	Must

Table 5: Traffic Station Data-Source: requirement TSDS_3 (data frequency update).

The main design choices related to this specific system component have mainly covered the following aspects:

- improvement of the initial **roadside traffic monitoring system**;
- creation of a web-based **connection with the Traffic Control Centre** and specific web-services for the access to the data stored in this protected environment;
- definition of the technical modalities and protocols for the **real-time data transfer service**;
- evaluation of the **data collection exploitation opportunities** that can be developed after the project's end.

In order to get a proper assistance in the execution of these sub-task, the Municipality of Bolzano has decided to give a contract to an external subcontractor, i.e. the local company

Famas System. This company is responsible of all the ITS systems that are at present managed by the Traffic Control Centre, and therefore has all the basic background knowledge to properly cover this list of activities, together with the project partners. The role of Famas System is not only limited to this design phase to the project, but will also cover the implementation process as well the final test and validation activities.

4.2.1 Traffic monitoring stations installations

As already described in deliverable D.2.1.1, the technology which is actually in use for the fixed traffic detection is based on inductive loops. During the early phase of this design process, specific field verification analysis put unfortunately in evidence the necessity to reactivate a certain number of loops which revealed to be damaged, and therefore not in the condition to provide reliable traffic data any more.

For this reason, project partners decided to allocate some project effort on this specific reactivation activity, in order to have a certain number of fixed traffic detection installations capable to properly collecting traffic data within all the urban network of the city of Bolzano. This extended requirement coverage, which appears to be broader with respect to the effective needs of INTEGREEN (in fact, the main area of interest is limited to a certain reference demo route, highlighted in blue in Figure 32), is actually considered at this stage because of the possibility to properly use this data in order to feed the traffic simulation model, as it will be more specifically analyzed in the section of this report dealing with this system component.

The overall fixed traffic installations which will be available for the INTEGREEN demonstrative activities are illustrated in Figure 32. The fixed stations colored in green are the traffic stations which are going to be repaired in the scope of the INTEGREEN project, and are put in correspondence of these crucial points:

- traffic station at the **intersection between Resia Street and Druso Street** (Figure 33). This a very important point for the overall road network of the city, since it links the “access” traffic coming from the Oltradige area (including part of the traffic coming from Merano through the local highway) and the “inside” traffic on two of the most crowded roads of the city (Resia Street and Druso Street);

- traffic station at the **intersection of Maso della Pieve Street** (Figure 34). Thanks to this fixed detection point, it is possible to estimate how the traffic generated by the local residential area of San Giacomo is split into the directions of the industrial area and the other residential area of Oltrisarco;

- traffic station in **Vittorio Veneto Street** (Figure 35). Thanks to this fixed detection point, it is possible to estimate the “access” traffic that connects the city centre with the old road connection of Merano, in correspondence of the city hospital.

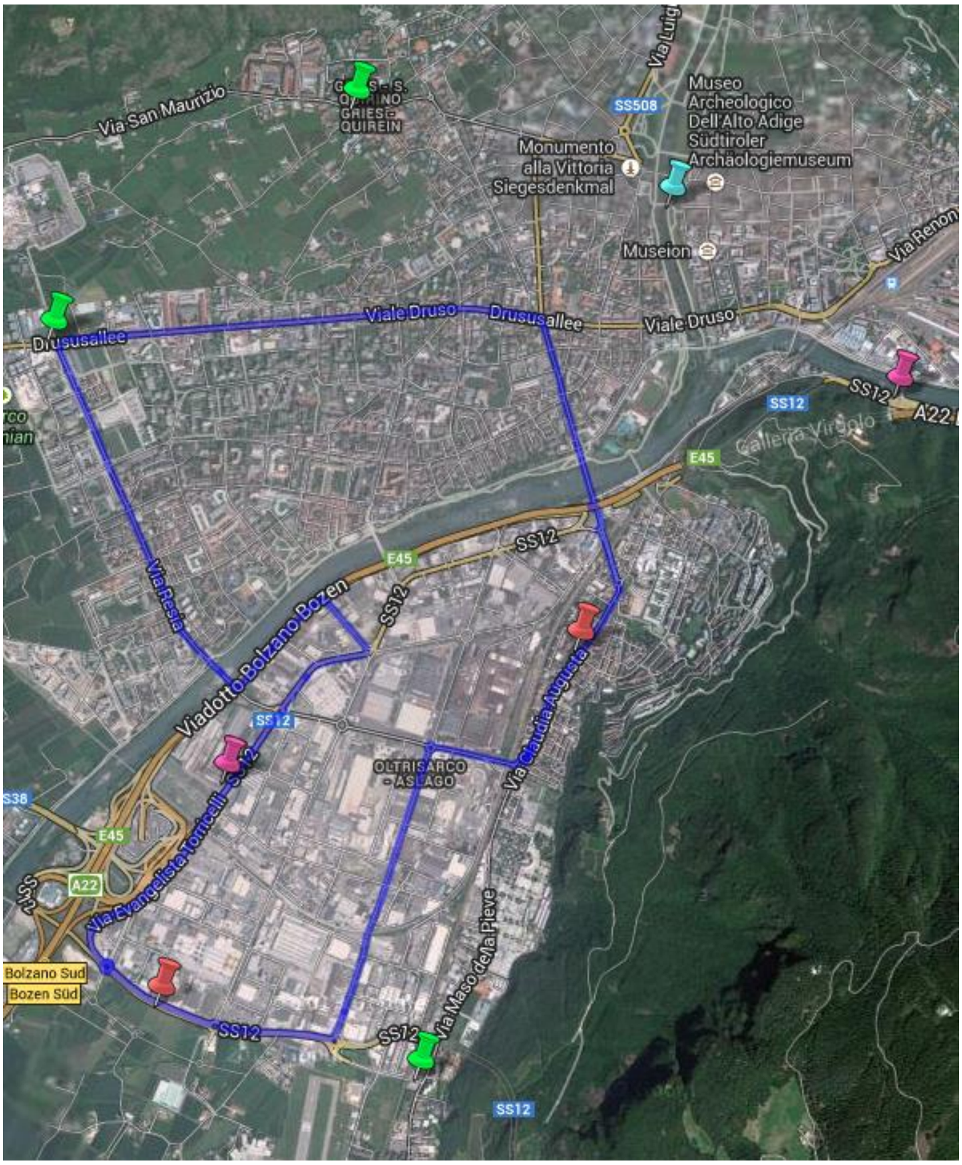

Figure 32: The map of fixed traffic detection installations considered in the INTEGREEN project.

Two further fixed stations are going to be repaired outside the scope of INTEGREEN (highlighted in the map in pink color), namely:

- traffic station in correspondence of **Torricelli Street** (Figure 36), which will detect the “access” traffic coming from the A22 toll highway and from the local highway connecting Merano to Bolzano, as well as the “inner” traffic generated by the industrial area of the city;
- traffic station in correspondence of **Virgolo Bridge** (Figure 37), which will detect the “access” traffic coming from the north-east gate of the city, which is generated in particular by the traffic flows exiting from the A22 (Bolzano North gate) and by the road SS12 which connects Bolzano to another important urban centre of the Province, Bressanone.

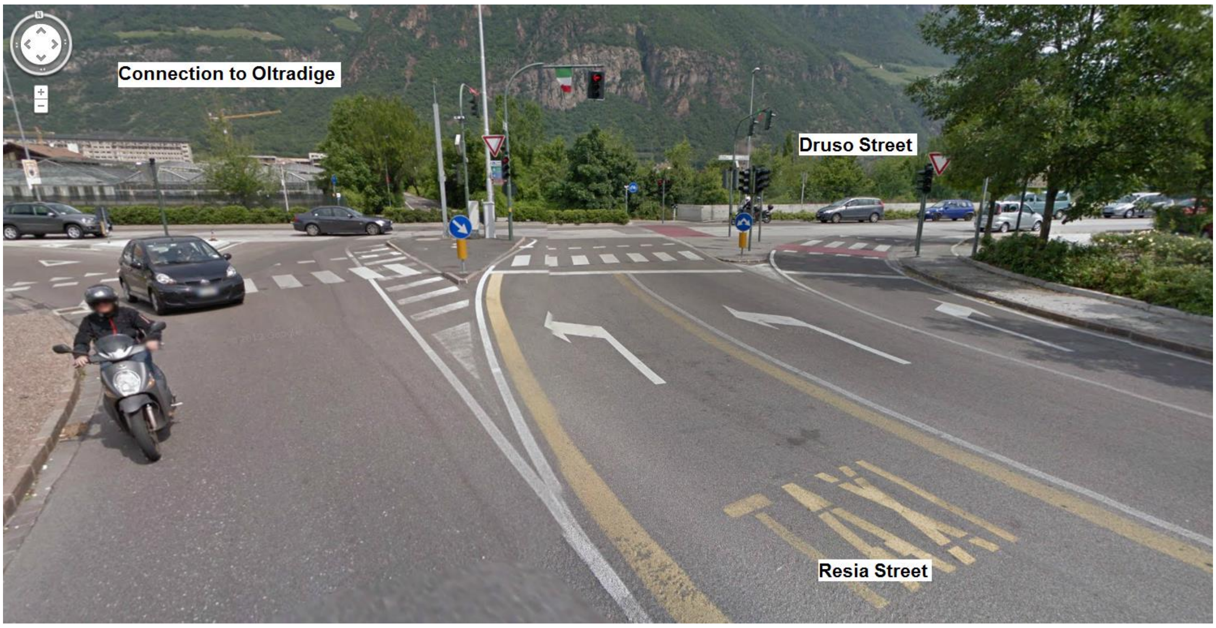

Figure 33: The intersection between Resia and Druso street.

Figure 34: The intersection in correspondence of Maso della Pieve street.

Figure 35: The traffic detection location in correspondence of Vittorio Veneto street.

Figure 36: The traffic detection location in correspondence of Torricelli street.

Figure 37: The traffic detection location in correspondence of Virgolo bridge.

The map in Figure 32 presents furthermore the expected installation locations of the **two integrated traffic / air pollution monitoring stations** which will be purchased within the INTEGREEN project through a public tender managed by the Municipality of Bolzano. One detection area is going to be the industrial area of the city, which is characterized by intense traffic flows (in particular during the commuting periods of the day) and therefore source of non-negligible air pollutant emissions (Figure 38). The second installation area is going to be individuated within the Oltrisarco residential district, where an “official” air quality station owned by the Local Agency for the Environment of the Province of Bolzano is available (Figure 39). This will allow from one side to perform initial calibration tests on the air pollution sensors that are going to be installed, and on the other side to make specific traffic / environmental measurements in correspondence of a sensitive area which could be specifically targeted by the “eco-friendly” traffic policies that are going to be experimented within Action n.5. It is worth noting that the air pollution measurement stations are portable instruments, which could be easily moved from a detection point to another for specific test analysis also on other sensitive roads of the INTEGREEN demo route, like for example Resia Street, Druso Street or Roma Street. For more details about the technical capabilities of the fixed air pollution monitoring units chosen for the INTEGREEN testing activities, please refer to paragraph 4.5.

Finally, INTEGREEN will also manage to create a remote real-time connection with the **bicycle counter system** which is already in function since several years in correspondence of one of the urban bicycle roads connecting to the historical city centre (Figure 40). This data will be not only useful for monitoring purposes (e.g. in order to evaluate the trend evolution of bicycle flows), but also for providing interesting multi-modal suggestion to local travelers (e.g. sharing the information that lots of people are actually using the bicycle in order to move within the city).

Figure 38: The integrated traffic / air pollution detection location in correspondence of Einstein street.

Figure 39: The integrated traffic / air pollution detection location in correspondence of Oltrisarco district.

Figure 40: The bicycle counter display of the city of Bolzano (Source: byccibeton.files.wordpress.co).

4.2.2 Traffic monitoring stations technical details

From a technical point of view, traffic detection tasks will be implemented by means of conventional **inductive loops** installed in the road asphalt. Inductive (magnetic) loops are based on the idea to detect the variations of their inductance as a consequence of a vehicular transit. This variation is then converted in frequency variations of a detector oscillator (Figure 41).The inductance variations are measured through a loop arranged in one or more insulated tires put in a shallow slot sawed in or across the road pavement. Different loop detectors exist (i.e. saw-cut, trenched-in and preformed), and determine the way the loop needs to be installed (and maintained) in the road pavement. Inductive loops can not only determine volume, presence and occupancy of the vehicles, but also their speed and their type, determined as a function of post-processing operations based on fundamental traffic variables relationships and automatic pattern recognition of the inductance variation signal, respectively. This method has been to be accurate and reliable, but however suffers of significant installation and maintenance costs as well as high failure rates. At present, is still probably the most reliable automatic traffic detection technology available on the market, even if a lot of alternative detectors now exist, including non-intrusive ones [16].

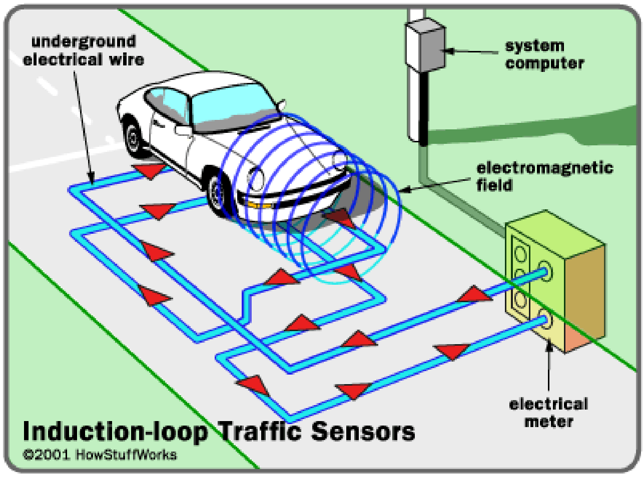

Figure 41: Principle of an inductive loop (Source: auto.howstuffworks.com).

In order to manage the connection with the Traffic Control Centre and the on-site pre-processing and aggregation of the collected data, partners have decided to use an interesting product recently put in commerce by Famas System which is called **MROAD 500** (Figure 42). The interesting feature of this technical solution, apart its high communication capacity and its low energy demand, is in its capability to acquire data from heterogeneous sensors monitoring not only traffic but also weather and road surface conditions. This solution could be therefore used in the future in order to have a wider number of "integrated" roadside monitoring stations. The interesting perspective is that thanks to this remote control unit it will be also possible in the future to easily move towards more advanced monitoring solutions, e.g. non-invasive traffic detection sensors or even wireless magnetometers. The unit can be easily controlled and managed in the back-end through a common browser thanks to an on-board web server. Local checks can be made using diagnostic leds on the front cover of the device. It is also possible to locally download the data and upload new firmware updates. The data transfer is done automatically and on a regular basis on a FTP server, using standard TPC/IP protocols. The complete technical specification of MROAD 500 is reported in Table 6.


Figure 42: The MROAD 500 roadside unit (Source: Famas System).

Specification	Description
Dimensions	285.5 x 171.5 x 95.5 [mm] (H x L x D)

Casing Material	ABS with aluminium cover
Protection Class	IP65
Power Supply	12 VDC – 24VDC
Power Consumption	~ 0.8 [W] (with external interfaces switched off)
Application Range	Temperature: -40 [°C] + 80 [°C] (except Ethernet interfaces) Relative Humidity: 0 – 100 [%], no condensation
Electronics	CMOS technology with real-time clock and watchdog
Operation System	Embedded Linux
Date / time	Automatic synchronized through NTP server
Collected data	• **Traffic**: date, time, lane, direction, speed, length, distance, interval, class • **Meteo**: date, time, values measured by sensors; • **Road surface status**: road surface temperature and at -4 [cm], road surface conditions (dry, wet, wet with salt, salt concentration)
Transmitted data	Raw, calculated, aggregated, accumulated data based on selectable intervals; diagnostic data
Memory	2 GB SD Card
Communication	• Web Server • 4 indication LEDs • 1 or 3 buttons with LCD display • 1 Ethernet 10/100T interface, TCP/IP • 1 USB interface • 1 GSM/GPRS/UMTS modem
Serial Interfaces	2 RS-485 interfaces with power supply (SCAD-BUS and STAR-BUS)
Sensor Interfaces	• 8-16 inputs for inductive sensors • 8 analogue inputs for meteorological sensors (with 12 bit resolution converter) • 1 input for road sensors and precipitation / visibility sensor via SCAD-BUS • 1 input for STAR sensors (non-inductive traffic) via STAR-BUS
I/O	• 4-8 digital sensors (1 reserved for the cabinet door contact) • 4 relay contacts

Table 6: The specification of the MROAD500 roadside unit.

4.2.3 Traffic data real-time transfer service details

The integration approach illustrated in the previous chapter determined the necessity to activate an **interface with the Traffic Control Centre**, since at present the centre is a Local Area Network (LAN) without any external connection to the Internet. The design of this fundamental task has been carried out directly by the company Famas System in strict cooperation with the internal office of the Municipality of Bolzano, responsible for the management of the overall internal ICT infrastructure, and has included:

- the **dimensioning of the connection line**, capable to host at minimum several [Mbit/s];
- the **selection of the hardware and software equipment** (e.g. router, firewall) needed for the secure physical activation of the line;
- the definition of a **new plan for the IP addressing** of the machines located at the Traffic Control Centre, as well as the **selection of the ports** in correspondence of which the data packets can be transmitted or received.

For the purpose of the real-time data transfer service required in INTEGREEN, the communication will take place on a classical HTTP communication based on TCP/IP. As far as the technical modalities and protocols considered for the implementation of this web-service, partners at the beginning evaluated the opportunity to use at this level of the architecture DATEX II. However, the purpose of this service is not to provide a public service to the end-users (intended as service providers), as foreseen at the front-end layer of the Supervisor Centre, but to have an "internal" service, simple and optimized for the current needs. For this reason, partners involved in this task have decided to use a very simple remote data transfer technology, i.e. **XML-RPC**, a basic *Remote Procedure Call* (RPC) approach that many consider at the base of the more complex *Simple Object Access Protocol* (SOAP). XML-RPC is a protocol running over HTTP that allow a client and a server to exchange information based on a minimum set of XML vocabulary and which is completely described by:

- the XML-RPC *data model*, a set of types used for the transfer of parameters, values and the management of communication errors. Six different data types are considered in the specification, four "basic" and two "complex" which are created through a series of heterogeneous sets of basic data types (Table 7).

Type	Description	Example
int	32-bit integer numbers	<int>27<int> <i4>27<i4>
double	64-bit floating numbers	<double>27.31415</double> <double>-1.1465</double>
boolean	True / false	<boolean>1</boolean> <boolean>0</boolean>
string	ASCII text	<string>Hello</string> <string>bonkers! @</string>
dateTime.iso8601	Date in ISO/IEC 8601 format (CCYYMMDDTHH:MM:SS)	<dateTime.iso8601> 20021125T02:20:04 </dateTime.iso8601> <dateTime.iso8601> 20020104T17:27:30

		</dateTime.iso8601>
base64	Binary information codified in base 64 (RFC 2045 specification)	<base64> SGVsbG8sIFdvcmxkIQ== </base64>
array	Arrays of heterogeneous data types, multi-dimensional arrays	**Heterogeneous arrays** <value> <array> <data> <value><boolean>1</boolean></value> <value><string>Test</string></value> <value><int>-91</int></value> <value><double>42.14159265</double></valu e> </data> </array> </value> **Multidimensional arrays** <value> <array> <data> <value> <array> <data> <value><int>10</int></value> <value><int>20</int></value> <value><int>30</int></value> </data> </array> </value> <value> <array> <data> <value><int>15</int></value> <value><int>25</int></value> <value><int>35</int></value> </data> </array> </value> </data> </array> </value>
struct	Structs of "members"	<value> <struct> <member> <name>givenName</name> <value><string>Joseph</string></value> </member> <member> <name>familyName</name> <value><string>DiNardo</string></value> </member> <member> <name>age</name> <value><int>27</int></value> </member> </struct>

		</value>

Table 7: The XML-RPC protocol data model.

- the XML-RPC *request format*, which specifies how a XML-RPC call must be prepared. The request must be contained in a XML document in which the root element is called "*methodCall*", which contains in turn the elements "*methodName*" and "*params*", which indicated the name of the procedure to be remotely called and the associated parameters' values to be transferred, respectively;

- the XML-RPC *response format*, which is structured as a request format, but with a "*methodResponse*" root element (instead of "*methodCall*") and without any "*methodName*" element. The output of the remote procedure which is called is given in a unique single parameter, which can be organized as an array in case of multiple values. It is worth noting that a "fault" element contained in the "*methodResponse*" root element can be used in case an exception has taken place.

An example of XML-RPC communication is given below, and gives an idea of the extreme simplicity of this data transfer technology. In this case, the remote procedure calculates the area of a circle by giving in input the length of its radius.

```
POST /xmlrpc HTTP 1.0
User-Agent: myXMLRPCClient/1.0
Host: 192.168.124.2
Content-Type: text/xml
Content-Length: 169
<?xml version="1.0"?>
<methodCall>
    <methodName>circleArea</methodName>
    <params>
        <param>
            <value><double>2.41</double></value>
        </param>
    </params>
</methodCall>

HTTP/1.1 200 OK
Date: Sat, 06 Oct 2012 23:20:04 GMT
Server: Apache.1.3.12 (Unix)
Connection: close
Content-Type: text/xml
Content-Length: 124
<?xml version="1.0"?>
<methodResponse>
    <params>
        <param>
            <value><double>18.24668429131</double></value>
        </param>
    </params>
</methodResponse>
```

A XML document can be automatically generated through software code implemented in different development languages. One of the most widespread choices is to use Java, both at client and server side, for example availing of the libraries put at disposal by the project Apache XML-RPC [17].

For the real-time data transfer needs under investigation, the design of the service has been carried out between the company Famas System (which will cover the server side of it) and the Associated Beneficiary TIS innovation park (which will run the client at the data source side).

A reference list of procedures to be implemented is given in Table 8. The first two procedures give only static metadata referred to each traffic station managed at the Traffic Control Centre. The method *TrafficWebService.getTrafficStationLanesAttributes* gives an overview about the capability for a certain traffic monitoring station to make a complete classification of the detected vehicles (and not only in terms of light / heavy vehicles), to count vehicles based on their speed profile and/or to provide further aggregated information such as gap, headway and sensor occupation time. Once this preliminary knowledge is available at the client side, it is possible to transfer the data by calling the proper four latter methods, e.g. if the complete vehicle classification data is available, it is possible to call the method *TrafficWebService.getTrafficStationLaneCompleteClassifiedData*, if the speed profile details are available, it is possible to call the method *Traffic WebService.getTrafficStationLaneSpeedData* and finally if optional traffic data are available as well the method *TrafficWebService.getTrafficLaneStationDataOptional* can be called.

Name	Description	Input type	Output type
TrafficWebService.getTrafficStations	Returns the full list of traffic stations sets. Each set contains: (i) an id; (ii) the name of the traffic station; (iii) the number of monitored lanes	-	Multidimensional array. Each "node" array contains: (i) int; (ii) string; (iii) int
TrafficWebService.getTrafficStationLanesAttributes	Returns a list of sets associated to each lane of a specific traffic station. Each set contains (i) the complete basic traffic data set; (ii) the number of vehicles aggregated per speed profile and (iii) additional, optional, aggregated traffic data.	Int (traffic station identifier)	Multidimensional array. Each "node" array contains three Boolean values
TrafficWebService.getTrafficStationLaneBasicClassifiedData	Returns the basic set of traffic data collected in the given window observation associated to a specific traffic station lane, i.e. the number of detected light / heavy vehicles.	Two int (traffic station identifier, lane identifier) and two dateTime.ISO (start and end time)	Array two int: light / heavy vehicles
TrafficWebService.getTrafficStati	Returns the complete basic set of traffic data collected in	Two int (traffic station identifier,	Array ten int – number of

onLaneCompleteClassifiedData	the given window observation associated to a specific traffic station lane, i.e. with a full detail of the number of vehicles associated to each vehicle class	lane identifier) and two dateTime.ISO (start and end time)	vehicles classified according to ITALY 9+1 classification mode
TrafficWebService.getTrafficStationLaneSpeedData	Returns the traffic data collected in the given window observation associated to a specific traffic station lane, classified as a function of the pre-determined speed classes	Two int (traffic station identifier, lane identifier) and two dateTime.ISO (start and end time)	Array of seven int, one for each speed class
TrafficWebService.getTrafficStationLeneDataOptional	Returns an additional, optional set of aggregated traffic data calculated over the given window observation associated to a specific traffic station lane.	Two int (traffic station identifier, lane identifier) and two dateTime.ISO (start and end time)	Array of three double: gap, headway, sensor occupation time

Table 8: The XML-RPC procedures for the real-time traffic data transfer between Traffic Control Centre and Supervisor Centre.

It is worth noting that the seven classes defined for the speed profile data package are the following:

- class 1: 0-15 [km/h]
- class 2: 15-25 [km/h]
- class 3: 25-35 [km/h]
- class 4: 35-45 [km/h]
- class 5: 45-55 [km/h]
- class 6: 55-70 [km/h]
- class 7: >70 [km/h]

An example of the XML client-server communication related to the request of the four latter procedure is given below.

TrafficWebService.getTrafficStationLaneBasicClassifiedData

```
POST /xmlrpc HTTP 1.0
User-Agent: myXMLRPCClient/1.0
Host: 192.168.124.2
Content-Type: text/xml
Content-Length: 169
<?xml version="1.0"?>
```

```
<methodCall>
    <methodName>
    TrafficWebService.getTrafficStationLaneBasicClassifiedData
    </methodName>
    <params>
         <param>
             <value>
                <array>
                   <data>
                      <value><int>101</int></value>
                      <value><int>1</int></value>
                   </data>
                </array>
             </value>
         </param>
    </params>
</methodCall>

HTTP/1.1 200 OK
Date: Sat, 06 Oct 2012 23:20:04 GMT
Server: Apache.1.3.12 (Unix)
Connection: close
Content-Type: text/xml
Content-Length: 124
<?xml version="1.0"?>
<methodResponse>
    <params>
         <param>
             <value>
                <array>
                   <data>
                      <value><<dateTime.iso8601>20140125T13:25:00
                      </dateTime.iso8601>></value>
                      <value><int>45</int></value>
                      <value><int>8</int></value>
                   </data>
                </array>
             </value>
         </param>
    </params>
</methodResponse>
```

TrafficWebService.getTrafficStationLaneCompleteClassifiedData

```
POST /xmlrpc HTTP 1.0
User-Agent: myXMLRPCClient/1.0
Host: 192.168.124.2
Content-Type: text/xml
Content-Length: 169
<?xml version="1.0"?>
<methodCall>
    <methodName>
    TrafficWebService.getTrafficStationLaneBasicClassifiedData
    </methodName>
```

```
    <params>
         <param>
             <value>
                <array>
                   <data>
                      <value><int>101</int></value>
                      <value><int>1</int></value>
                   </data>
                </array>
             </value>
         </param>
    </params>
</methodCall>

HTTP/1.1 200 OK
Date: Sat, 06 Oct 2012 23:20:04 GMT
Server: Apache.1.3.12 (Unix)
Connection: close
Content-Type: text/xml
Content-Length: 124
<?xml version="1.0"?>
<methodResponse>
    <params>
         <param>
             <value>
                <array>
                   <data>
                      <value><<dateTime.iso8601>20140125T13:25:00
                      </dateTime.iso8601>></value>
                      <value><int>14</int></value>
                      <value><int>13</int></value>
                      <value><int>20</int></value>
                      <value><int>5</int></value>
                      <value><int>7</int></value>
                      <value><int>4</int></value>
                      <value><int>0</int></value>
                      <value><int>0</int></value>
                      <value><int>0</int></value>
                      <value><int>0</int></value>
                   </data>
                </array>
             </value>
         </param>
    </params>
</methodResponse>
```

TrafficWebService.getTrafficStationLaneSpeedData

```
POST /xmlrpc HTTP 1.0
User-Agent: myXMLRPCClient/1.0
Host: 192.168.124.2
Content-Type: text/xml
Content-Length: 169
<?xml version="1.0"?>
```

```
<methodCall>
    <methodName>
    TrafficWebService.getTrafficStationLaneBasicClassifiedData
    </methodName>
    <params>
         <param>
             <value>
                <array>
                   <data>
                      <value><int>101</int></value>
                      <value><int>1</int></value>
                   </data>
                </array>
             </value>
         </param>
    </params>
</methodCall>

HTTP/1.1 200 OK
Date: Sat, 06 Oct 2012 23:20:04 GMT
Server: Apache.1.3.12 (Unix)
Connection: close
Content-Type: text/xml
Content-Length: 124
<?xml version="1.0"?>
<methodResponse>
    <params>
         <param>
             <value>
                <array>
                   <data>
                      <value><<dateTime.iso8601>20140125T13:25:00
                      </dateTime.iso8601>></value>
                      <value><int>16</int></value>
                      <value><int>18</int></value>
                      <value><int>24</int></value>
                      <value><int>7</int></value>
                      <value><int>2</int></value>
                      <value><int>0</int></value>
                      <value><int>0</int></value>
                   </data>
                </array>
             </value>
         </param>
    </params>
</methodResponse>
```

TrafficWebService.getTrafficStationLaneDataOptional

```
POST /xmlrpc HTTP 1.0
User-Agent: myXMLRPCClient/1.0
Host: 192.168.124.2
Content-Type: text/xml
Content-Length: 169
```

```
<?xml version="1.0"?>
<methodCall>
    <methodName>
    TrafficWebService.getTrafficStationLaneBasicClassifiedData
    </methodName>
    <params>
        <param>
            <value>
               <array>
                  <data>
                     <value><int>101</int></value>
                     <value><int>1</int></value>
                  </data>
               </array>
            </value>
        </param>
    </params>
</methodCall>

HTTP/1.1 200 OK
Date: Sat, 06 Oct 2012 23:20:04 GMT
Server: Apache.1.3.12 (Unix)
Connection: close
Content-Type: text/xml
Content-Length: 124
<?xml version="1.0"?>
<methodResponse>
    <params>
        <param>
            <value>
               <array>
                  <data>
                     <value><<dateTime.iso8601>20140125T13:25:00
                     </dateTime.iso8601>></value>
                     <value><double>4.35</double></value>
                     <value><double>0.47</double></value>
                     <value><double>0.02</double></value>
                  </data>
               </array>
            </value>
        </param>
    </params>
</methodResponse>
```

As far as the the retrieval of the **bicycle counter data**, a specific web-service will be used. This will be a simplification of the one proposed for the transfer of vehicular traffic data, as presented in Table 9.

Name	Description	Input type	Output type
BicycleWebService.getCounterS	Returns the full list of bicycle counter stations	-	Multidimension al array. Each

tations	sets. Each set contains: (i) an id; (ii) the name of the station.		"node" array contains: (i) int; (ii) string;
BicycleWebService.getCounterStationsData	Returns the number of detected bicycles in the given window observation associated to a specific counter station.	Two int (traffic station identifier, lane identifier) and two dateTime.ISO (start and end time)	One int: (number of bicycles)

Table 9: The XML-RPC procedures for the bicycle counter data transfer between Traffic Control Centre and Supervisor Centre.

Based on the indication of requirements TSDS_3, the client must be implemented such that all the available traffic data are requested every five minutes, following this data transfer logic and protocols. If for any reason it was not possible to complete the data transfer correctly, either from the roadside stations to the Traffic Control Centre or from the Traffic Control Centre to the Supervisor Centre, client and server must be able to implement a logic that allows the client to retrieve all the missing data, including the one associated to previous window observations. This can be simply handled by letting the client sharing with the server the timestamp associated to the last data delivery, which is compliant with requirements DSL_11. It is worth noting that the software client implementing the traffic data source, similarly to any other data source, must be able to decode the XML-RPC response and control if the data is pre-validated and formatted as expected, as stated in requirements DSL_3 and DSL_4, respectively. An idea for implementing this in an efficient way could be to define a common library of procedures that can be shared among all different data sources.

Finally, in order to guarantee that the source is trustworthy (requirement DSL_10), there must be some mechanism to ensure that the channel between the Traffic Control Centre and the Supervisor Centre is fully secure. This could be simply guaranteed through the use of a ***Virtual Private Network*** (VPN), which is able to logically extend a private network (such as the LAN located at the Traffic Control Centre) over the Internet. A VPN is in the condition to guarantee that the data communication can only take place after an authenticated remote access and through proper encryption techniques.

4.2.4 Future exploitation opportunities

Through such a remote real-time traffic data transfer service, the Traffic Station Data Source is intrinsically in the condition to gather more traffic data from the Traffic Control Centre, once it will be available.

An interesting exploitation opportunity for this data source component could be furthermore the possibility to receive on a real-time basis even the traffic data managed at regional level that might be of interest for the urban area of Bolzano (highlighted in yellow in Figure 43). In order to implement this, a second client must be managed at this stage, capable to connect to a different reference server and to retrieve the data following a different logic and protocol (i.e. SOAP). The service in question could be prepared by Südtiroler Informatik AG (SIAG), the local in-house IT organization owned by the Autonomous Province of Bolzano which is responsible of the automatic data storage of all remote field installations owned by the regional administration, similarly to what has been implemented at the Environmental Station Data Source for the real-time data transfer of the air pollution data managed by the Local Agency for the Environment (see paragraph 4.5 for more details). Thanks to this reference code, it is possible to foresee a reduced time / cost effort for the future activation of this web service.

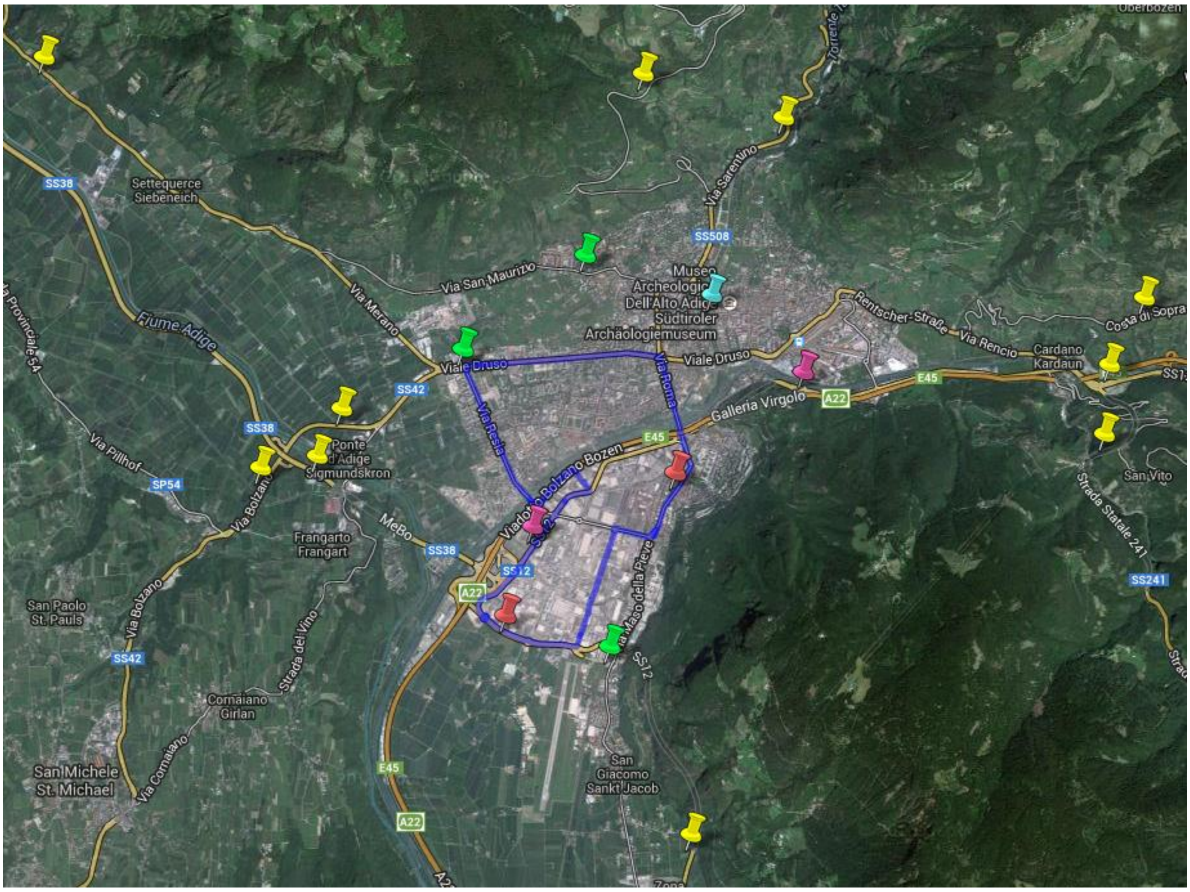

Figure 43: The complete map of fixed traffic detection installations (extended to the regional traffic detectors network).

4.3 Parking Data-Source

The Parking Data Source is the software component at the Environmental Supervisor Centre which is responsible to gather data regarding the current occupancy status of the main parking areas distributed in the city of Bolzano. For completeness sake, the tables below illustrate again the main specific requirements defined in the previous Action n.2 for this data source.

ID	PDS_1
Name	Data type – generated data
Description	Each record of generated data must contain the number of free slots of a specific parking area.
Rationale	Minimum set of generated data to be collected by the parking source.
Type	Functional
Priority	Must

Table 10: Parking Data-Source: requirement PDS_1 (data type – generated data).

ID	PDS_2
Name	Data type - basic information
Description	The parking data-source must have basic information about the number of available slots that each controlled parking area can manage.
Rationale	Mandatory information in order to evaluate the current level of occupancy of the parking areas.
Type	Functional
Priority	Must

Table 11: Parking Data-Source: requirement PDS_2 (data type – basic information).

ID	PDS_3
Name	Data frequency update
Description	The maximum time interval between two consecutive generated data packets delivered to the parking data-source must be 5 minutes.
Rationale	Guarantee minimum system performance which is compatible with the one requested to the traffic stations data-source.
Type	Performance
Priority	Must

Table 12: Parking Data-Source: requirement PDS_3 (data frequency update).

The main design choices related to this specific system component have mainly covered the following aspects:

- creation of a web-based **connection with the Traffic Control Centre** and specific web-services for the access to the data stored in this protected environment;
- definition of the technical modalities and protocols for the **real-time data transfer service**;

Even these tasks have been carried out in cooperation with the local company **Famas System**.

4.3.1 Parking areas under monitoring

The map of the parking areas which are actually managed by the system and which will be included in the INTEGREEN architecture is presented in Figure 44. Most of them are located in the city centre of the city.

4.3.2 Parking data real-time transfer service details

The real-time transfer service of the parking data avails of the interface to the Traffic Control Centre already illustrated in the previous paragraph. Similarly as for the Traffic Station Data Source, the **XML-RPC** technology has been considered for accomplishing this task.

A reference list of procedures to be implemented is given in Table 13. In this case, only two procedures are considered, one for the retrieval of static information about all parking areas (*ParkingWebService.getParkingAreas*),and one for the reception of the real-time occupancy of a specific parking area. The associated timestamp is included in the output in order to check the freshness of the data.

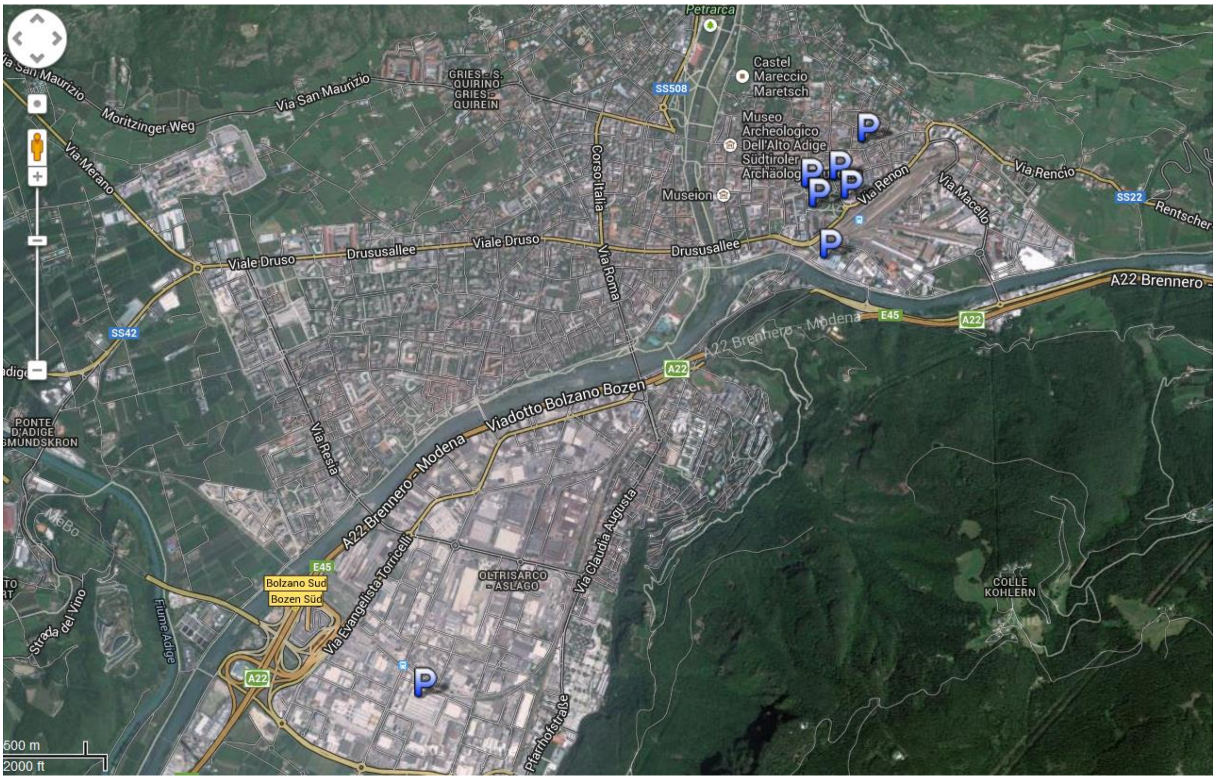

Figure 44: The map of the parking areas supported by INTEGREEN.

Name	Description	Input type	Output type
ParkingWebService.getParkingAreas	Returns the full list of parking areas sets. Each set contains: (i) an id; (ii) the name of the parking area; (iii) the number of available parking slots	-	Multidimensional array. Each “node” array contains: (i) int; (ii) string; (iii) int
ParkingWebService.getParkingData	Returns the real-time parking of a given parking area	One int (parking area identifier)	Array of heterogeneous values: real-time free slots (int); associated timestamp (dateTime.ISO)

Table 13: The XML-RPC procedures for the real-time parking data transfer between Traffic Control Centre and Supervisor Centre.

4.4 O/D Data-Source

The Origin/Destination (O/D) Data-Source (ODDS) is in charge to gather the data coming from the origin/destination source which the Municipality of Bolzano has in plan to introduce in its monitoring system. It will be based on ANPR technology and aims to identify the passages of vehicles at specific points through the automatic recognition of their number plate and to let them match them at different locations in order to statistically quantify on a real-time basis (i) the traffic flows on a specific O/D couple and (ii) the correspondent travel times. The O/D data source is interested in receiving both the generated data collected by the monitoring stations as well as the pre-elaboration outputs calculated on a remote server. For completeness sake, the tables below illustrate again the main specific requirements defined in the previous Action n.2 for this data source.

ID	ODDS_1
Name	Data type – raw generated data
Description	The O/D data-source must receive the raw generated data collected by a specific monitoring station of the O/D source, located at a specific point of the road network, i.e.: • the total number of vehicles recognized within the window observation; • a vehicle identifier which is anonymously associated to the number plate of each passing vehicle.
Rationale	Minimum set of raw generated data to be collected by the O/D source.
Type	Functional
Priority	Must

Table 14: O/D Data-Source: requirement ODDS_1 (data type – raw generated data).

ID	ODDS_2
Name	Data frequency update
Description	The maximum time interval between two consecutive generated data packets delivered to the O/D data-source must be 5 minutes.
Rationale	Guarantee minimum system performance
Type	Performance
Priority	Must

Table 15: O/D Data-Source: requirement ODDS_2 (data frequency update).

4.4.1 Experimental Bluetooth travel times detection system

During this high-level design and planning process of the traffic-related roadside monitoring units, enriched by the technological analysis carried out in Action n.2, partners (and in particular TIS) have started considering the opportunity to combine this future ANPR cameras network with an additional low-cost one composed by non-invasive units capable to collect data to be used for the real-time calculation of the vehicular travel times and eventually for making interesting origin / destination analysis concerning the urban trip behaviors. This detection methodology is based on a novel technique based on the idea to scan the Bluetooth devices which are present in the vehicles. By comparing their anonymous identifiers at different locations, one could determine in quite accurate way the travel time needed for moving between the two points of interests, as graphically illustrated in Figure 45.

Bluetooth is a wireless technology standard for exchanging data over short distances from fixed and mobile devices. Without delving into the features and details of this wireless technology for the sake of clarity it is however relevant to mention that from the physical point of view the Bluetooth uses the microwave radio frequency spectrum operating in the range of 2400 [MHz] to 2483.5 [MHz]. Moreover the transmitted data are divided into packets and each packet is transmitted on one of the 79 designated Bluetooth channels. The Bluetooth technology overlaps with the standard Wi-Fi 802.11 which operates almost in the same frequency range, and therefore the efficiency of a Bluetooth detector can be drastically

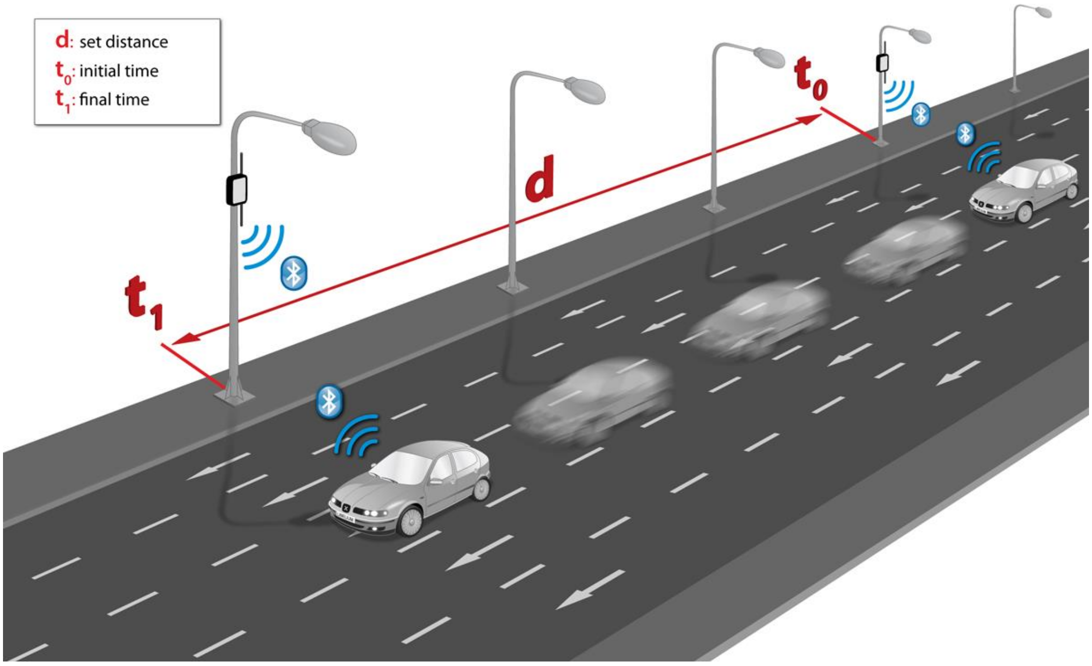

Figure 45: Graphical explanation of the Bluetooth vehicular travel times detection technology (Source: Libelium).

reduced when deployed in the presence of a nearby public/private Wi-Fi access point. Which means that in these situations the number of Bluetooth detections will be lower than the potential ones. It is also worth to mention that not all the Bluetooth devices can be detected, more in detail when a Bluetooth device is switched on it can be set in three different states namely:

- **visible**, which means that the Bluetooth device is visible and it will reply to any other 'interested' Bluetooth devices in the communication range;
- **invisible**, which means that the device is not detectable even if it is physically placed in the communication range of another device interested in starting a communication;
- **limited**, which is an intermediate state that allows the device to be discoverable only by well-defined Bluetooth devices, more in detail it will be discoverable only by those devices it has coupled with before.

Most of the time the Bluetooth in-car devices, the only one that are of interest for the purposes of these monitoring, are placed by default to visible. The most relevant reason for that choice comes to simplify as much as possible the interaction among the in-car devices with the personal drivers' ones; as a matter of fact most of the times the in-car devices such as headset, navigation and entertainment system are places as visible. From our experience this approach is not followed by the modern smartphone due to the fact that are places as limited or better invisible by default. Given that **a Bluetooth detector on the road side will not be able to detect the smartphones in the car unless the user has voluntarily changed that parameter.**

The most relevant **advantages** of this monitoring approach state to the really simple deployment compared to the most common technology, i.e. ANPR: in this case, the cameras need to be installed on top on the monitored lane. Given this constraint, they are typically place at suitable locations, e.g. in correspondence of traffic lights which in many cases make the deployment even more complex for the very delicate work environment. Even more, the cameras need a much higher bandwidth in order to transfer all the gathered images; on the other side, Bluetooth units transfer only an identifier coupled by a timestamp for each detection. This gives a lot of advantages also in terms of power supply demand, which opens theoretically the opportunity to power these units through renewable energy source (i.e. solar panels).

This monitoring approach has on the contrary some few **drawbacks** that must be taken into account. The most relevant ones are related to the wireless technology used to spread the Bluetooth messages which does not provide any practical way to circumscribe the signal. In particular, as stated before the detection of in-car Bluetooth devices leverages on the detection of any Bluetooth signal spread in the detection area of the probe. By doing that although the detecting probes are installed as closest as possible to the road sides there are no guarantee that other 3' parties Bluetooth signals, signals that does not belong to any in-car device, can be avoided. In other words Bluetooth devices such as printers and portable audio speakers in the nearby houses can be wrongly exchanged for in-car devices and successfully logged along the real ones. Moreover, a car could own more than one Bluetooth devices, altering the effective number of relevant detections. Given these premises, one must be very careful to use this data in order to estimate the overall traffic flow passing through a specific monitoring point.

The vehicle traffic monitoring through the detection of Bluetooth devices can be described as a chain of three main and consecutive phases in which on the one hand the raw Bluetooth detections data are received as input and on the other hand estimations of the current travelling elapsed times between two monitoring stations are produced as output. More in details, the three phases can be summarized as follows:

- **Detection phase**. In this task, accomplished by each autonomous entity belonging to the distributed network of detectors deployed on the roads, the Bluetooth signals are physically detected. The raw data (i.e. a list of Bluetooth device identifiers paired with the correspondent timestamp of the detection) is moreover delivered on a continuous basis to a remote central unit.

- **Data elaboration phase**: this task, entirely covered in the back-end, is in charge to continuously compute the estimate travel times. More specifically:

    - the raw data are used in input in order to extract origin-destination *matches*, which are directly associated to a Bluetooth device identifier detected by a certain couple of stations;

    - the differences between the two timestamps characterizing each match are then used to estimate the overall travel time on a certain origin-destination couple by considering an appropriate aggregate indicator.

- **Data visualization phase**: finally, through an appropriate data publication system and graphical user interfaces, the most relevant information are presented to end-users in form of charts or graphs, with the possibility to analyze the traffic trends.

In this section, only the detection phase of the Bluetooth monitoring system is considered. More details about the other two phases are given in the next chapter (data elaboration) and in the deliverable D.3.1.2 (data visualization).

The detection process is designed as follows. Since any in-car Bluetooth device set as visible (discoverable mode) will transmit its own identifier on demand, the roadside Bluetooth detector is configured as an active node that continuously perform a scanning process in the surrounding space (over the whole set of frequency channels)in order to check the presence of other Bluetooth devices, and permanently stores in a local database each Bluetooth identifier along with timestamp. This detection operation is quite complex to design in practice, in order to minimize the number of missed detections: for example, if a car pass too fast in front of a detector, the in-car device will presumably receive the request but it will be too far away from the detector in order to positively deliver its response (Figure 46).

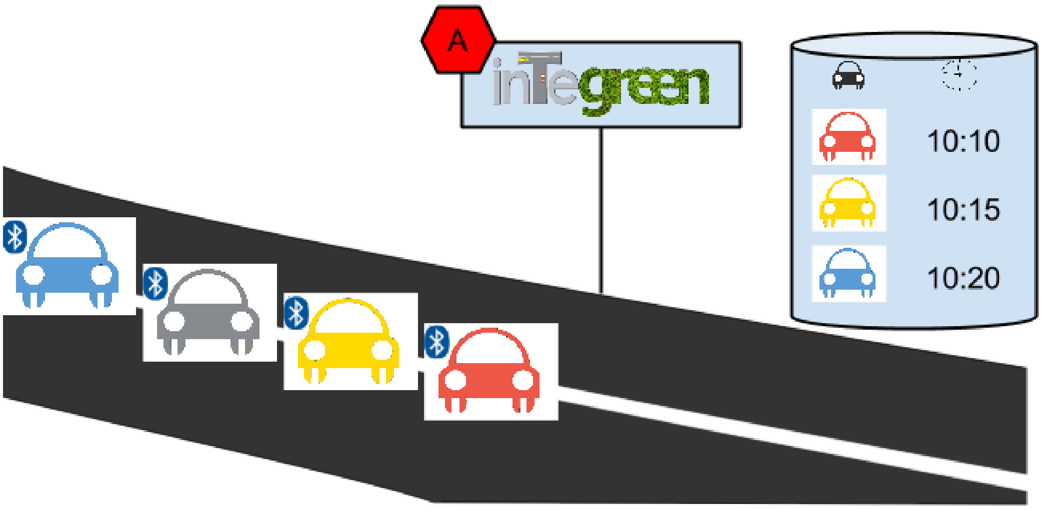

Figure 46: Bluetooth vehicular detection process (graphical explanation).

4.4.2 Planned installations

During this design phase, preliminary surveys have been carried out in under to understand how to properly cover all the reference demo route of INTEGREEN. The results of these investigations are reported in the map illustrated in Figure 47. Nine fixed installations are going to provide precise estimations of several road stretches in the city, which are of particular importance for the overall traffic circulation conditions. Each elementary road stretch is going to be characterized by two consecutive Bluetooth detectors installations. Thanks to this set of planned locations, an additional monitored test route could be included in the monitoring area, i.e. Galilei Street, one of the most important access roads connecting the A22 gate “Bolzano South” in direction city centre. A comprehensive list of all these road stretches (to be considered in both travel directions) is given in Table 16.

Figure 47: The installation plan of the Bluetooth detectors.

Link	Bluetooth detectors couple
1 – Resia Street	Installations at Resia Bridge and intersection Druso Street - Resia Street
2 – Druso Street	Installations at intersection Druso Street - Resia Street and intersection Roma Street - Druso Street
3 – Roma Street	Installations at intersection Roma Street - Druso Street and Roma Bridge
4 – Claudia Augusta Street	Installations at Roma Bridge – Oltrisarco district
5 – Volta Street	Installations at Oltrisarco district and Galvani Street
6 – Galvani Street	Installations at Galvani Street and roundabout Galvani Street – Einstein Street
7 – Einstein Street	Installations at roundabout Galvani Street – Einstein Street and roundabout Einstein Street – Torricelli Street
8 – Torricelli Street	Installations at roundabout Einstein Street – Torricelli Street and Siemens Street
9 – Arginale road (gates Righi Street and Resia Street)	Installations at Siemens Street and Resia Bridge
10 – Galvani Street (additional link)	Installations at Siemens Street and Roma Bridge

Table 16: The road stretches defined on top of the Bluetooth monitoring system.

Since it is planned to have Bluetooth detectors capable to work with continuous power supply, the selected locations are all in correspondence of traffic light intersections, which guarantee this kind of operative requirement. The roadside cabinets offer moreover a suitable place where to place the detectors in a safe and secure way. Other potential location candidates which could be considered for future installations, or for alternative ones (e.g. in case of high number of static Bluetooth / Wi-Fi devices in the communication range interfering with the scanning process) are presented in the figures below.

4.4.3 Bluetooth vehicular data real-time transfer service details

The information stored for each detection involves a **Bluetooth identifier** (i.e. an anonymization of its MAC address) and the **relative timestamp**. All log are stored in a simple but powerful database placed directly in the detector which allows all data gathered to be queried by using a given timestamp. Thanks to this **local database**, an external application could be easily able to synchronize the local storage with the remote storage without losing any precious data loss. In addition as soon as the data is correctly stored in the remote server, the local copy is removed due to security and privacy concerns.

Figure 48: Alternative site for hosting detector to be placed on Resia Bridge.

Figure 49: Alternative sites for hosting detectors in Druso Street.

Each Bluetooth detector periodically, i.e. every few minutes, synchronizes the remote data center database with the last data gathered. Differently from other data sources, in this case the data is delivered directly to the O/D Data Source, without passing through the Traffic Control Centre. Another significant difference in terms of approach is that in this case the data source has a "slave" position, i.e. it must only receive the data which is periodically transmitted; in the previous cases, the data sources were mostly in a "master" position, in the sense that are designed to periodically call the available web services methods to retrieve the data.

The data delivery phase needs to guarantee no packages lost, namely if a roadside unit is not able to send the data, i.e., temporary internet connection missing, as soon as it will be online again it will send all data not synchronized yet. For the implementation scheduled in INTEGREEN, the plan is to ensure the connectivity of each Bluetooth detector through a dedicated **GPRS cellular modem**. The limited amount of data to be transferred, the available bandwidth and the today connection costs are in the condition to properly satisfy the performance requirements of INTEGREEN as a system at reasonable costs. It is however worth noting that other communication technologies could be considered in the future to cover this data transfer. Last but not least, the data must be delivered throughout a secure channel implemented by a **VPN**, so to avoid any potential security issue over this communication path (Figure 50).

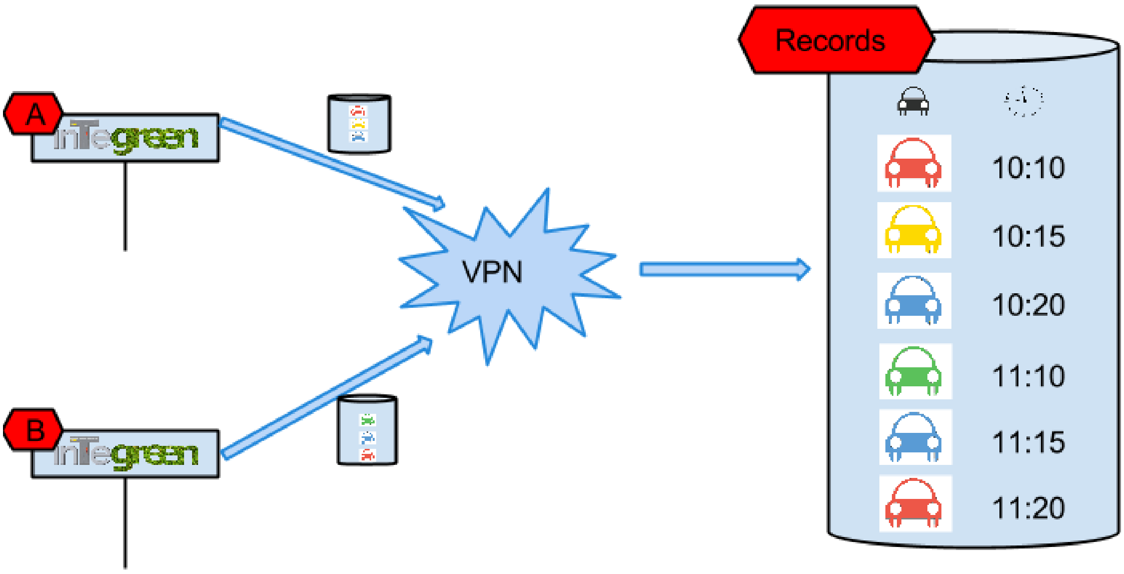

Figure 50: Remote delivery of Bluetooth vehicular raw data.

4.5 Environmental Station Data-Source and Meteo Data Source

The Environmental Station Data-Source (ESDS) is in charge to receive specific data about the air pollutants concentration at a street level. On the contrary, the Meteo Data-Source (MDS) represents the reference aggregation point for all the weather information gathered by the meteo stations distributed in the area of the city of Bolzano. It is worth noting how air pollution stations could be in the conditions to provide meteorological information as well (requirement ESDS_2). For completeness sake, the tables below illustrate again the main specific requirements defined in the previous Action n.2 for these data sources.

ID	ESDS_1
Name	Data type – environmental parameters
Description	Each single generated data record must contain one value of air pollution concentration for each of the following pollutants: • CO • O_3 • NO_X • VOC • $PM_{2.5}$

Rationale	Focus on main air pollutants emitted by traffic which cause the main problems in the city of Bolzano. CO is considered as well for joint comparisons with values gathered through the vehicle data-source.
Type	Functional
Priority	Must

Table 17: ESDS Data-Source: requirement ESDS_1 (data type – environmental parameters).

ID	ESDS_2
Name	Data type – meteorological parameters
Description	Each single generated data record must contain one value for each of the following meteorological parameters: • temperature; • humidity; • speed and direction of the wind.
Rationale	To have data that can calibrate the dispersion models at the data center layer.
Type	Functional
Priority	Must

Table 18: ESDS Data-Source: requirement ESDS_2 (data type – meteorological parameters).

ID	ESDS_4
Name	Data frequency update
Description	The maximum time interval between two consecutive generated data packets delivered to the environmental station data-source must be 15 minutes.
Rationale	Guarantee minimum system performance
Type	Performance
Priority	Must

Table 19: ESDS Data-Source: requirement ESDS_4 (data frequency update).

ID	MDS_1
Name	Data type
Description	Each single generated data record must contain one value for each of the following meteorological parameters: • temperature; • humidity; • wind speed and direction; • type, intensity and quantity of precipitation;
Rationale	Minimum set of generated data to be collected by the meteo source.
Type	Functional
Priority	Must

Table 20: MDS Data-Source: requirement MDS_1 (data type).

ID	MDS_3
Name	Data frequency update
Description	The maximum time interval between two consecutive generated data packets delivered to the meteo data-source must be 15 minutes.
Rationale	Guarantee minimum system performance which is compatible with the one requested to the environmental stations data-source
Type	Performance
Priority	Must

Table 21: MDS Data-Source: requirement MDS_3 (data frequency update).

4.5.1 Air pollution and meteorological stations installations

In paragraph 4.2 the expected installation locations of the two **integrated traffic / air pollution monitoring stations** which will be purchased within the INTEGREEN project have already been presented. Before entering into the details of the technical capabilities of the chosen fixed air pollution monitoring units, it is important to underline the accurate preparation work concerning the **definition of the technical specifications** that these two units must be in the conditions to fulfil. These specifications have been written by taking in considerations both the reference requirements defined in Action n.2 as well as the technological capabilities of the state-of-art techniques, an activity which has been carried out in strict relation to the low-cost air pollution sensors analysis carried out in Task 3.2 and which is most extensively presented in D.3.2.2. The list of specifications, which is fully reported in Annex 1 at the end of this report, has also taken into account the particular **type of installation** intended to be considered in the project. Standard installation guidelines (that are normally followed by the local agencies for the environment protection that are formally responsible of the air quality monitoring job), indicate certain reference distances between the installations and intersection points as well as the roadway for different types of monitoring activities [18]- [19]. In particular, for stations considered at high exposure to traffic, the following specifications must be considered:

- at least 4 [m] far from the nearest roadside level;
- at least 25 [m] far from the nearest traffic light, intersection, bus stop and similar point of interest.

In INTEGREEN, however, **<u>the focus is on air pollution measurements taken as much as possible near to the emission source</u>**, in order to:

- minimize the emissions' contribution provided by sources different from road traffic (and thus maximize the correlation studies);
- maximize the potential correlation between static and mobile measurements in correspondence of these fixed detection points.

The technical challenge is to verify the proper functioning and long-term stability of the chosen sensors in this complex environment, in which the values of air pollutants are expected to be quite high and therefore typically different from the laboratory tests in which they are calibrated. Cross-sensitivity effects could in fact significantly reduce the SNR of the measured signals and properly be compensated in post-processing.

It is worth noting that the specifications given to the private companies invited to the negotiation process have been technologically-independent, so to give the maximum space for the possible different technological alternatives by the companies and thus to be in the condition to make an overall cost/benefit assessment of the products available on the market. The winning proposal of Famas System is based on the idea to have **thick film semiconductor-based sensors** as well **PM detectors based on light-scattering technology** for the air pollution monitoring part (NO_2, O_3, CO and $PM_{2.5}$, respectively). This solution is at the start of the art as far as the real-time air pollution measurement is concerned, and is specifically developed by an Italian company called **Unitec**. Apart of being in condition to make quick air pollution measurements, with a good compromise in terms of accuracy / resolution with respect to the given technical specifications, the interesting aspect of this choice is that **semiconductor sensors are used within the on-board environmental monitoring unit of the mobile system as well**, making therefore detailed comparative analysis possible and reasonable. The sensor called **SENS3000** is integrated together with the full integrated PM light scattering module within a compact air quality station, called **ETL ONE** (Figure 51). The station can be directly remotely connected to the Traffic Control Centre thanks to an embedded cellular modem, without any necessity in this task to pass through the aforementioned MROAD500 remote unit. The advantage of this station is that it very compact and rather light, and therefore easily portable and mountable, e.g. to a pole of a traffic light (please refer to for all detailed specifications). This gives the opportunity to mount this component e.g. near an official air quality station for initial calibration purposes, and then move them in correspondence of specific interest points covered by complete traffic detection loops during certain testing activities. It is worth underlining how an active sample air flow is guaranteed in correspondence of the air pollution sensor by means of a small fan. Selectivity and precision are reached using special semiconductor oxides with appropriate filters.

Traditional meteorological sensors for the monitoring of ambient parameters (air temperature, relative humidity, wind speed and direction, solar radiation), which are needed to properly interpret the air pollution concentrations, are mounted separately and directly connected to the MROAD500 unit. Conventional measurement techniques such as PTC sensors, hygrometers, mechanical anemometers and thermopiles are going to be used.

Figure 51: The ETL ONE air quality station and detail of possible installation (Source: UNITEC).

Figure 52: The SENS3000 thick film semiconductors sensors (Source: UNITEC).

Specification	Description
Sensors	CO – NO_2 – O_3 – C_6H_6 – CH_4
Analog output	0-5 VCC (not linear: calibration curve provided)
Precision	< 2% (< 10% for O_3)
Power Supply	+/- 12 VDC
T90 Response Time	< 3 [s]
Operating Environment	Temperature: -20 [°C] + 50 [°C] Relative Humidity: 5 – 95 [%], no condensation
Power Consumption	150 [mA]
Operation System	Embedded Linux
Dimensions	55 x 94 [mm] (diameter x height)
Weight	220 [g]
Execution	Aluminium anodized cylindrical enclosure
Background noise	0.1 % f.s.
Accuracy	+/- 10 [µg/m^3] (+/- 0.5 [mg/m^3] for CO; +/- 1 [µg/m^3] for C_6H_6)
Resolution	0.1 [µg/m^3] (0.1 [mg/m^3] for CO; 1.0 [µg/m^3] for O_3 and CH_4)
Low Detection Limit	0.1 [µg/m^3] (0.1 [mg/m^3] for CO; 20 [µg/m^3] for O_3)

Range	• **CO**: 0-100 [mg/m^3] / 0-80 [ppm] • **NO_2**: 0-500 [μg /m^3] / 0-400 [ppb] • **O_3**: 20-500 [μg /m^3] / 10-200 [ppb] • **C_6H_6**: 0-100 [μg /m^3] / 0-30 [ppb] • **NO_X**: 0-800 [μg /m^3] / 0-500 [ppb] • **CH_4**: 0-300 [μg /m^3] / 0-400 [ppb]
Zero Span Drift	< 2.5% / 6 [months] (3 [months] for O_3)
Data Exchange	MODBUS RS485 (serial communication protocol for connecting industrial electronic devices)

Table 22: The technical specifications of the SENS3000 thick film semiconductors sensors.

Specification	Description
Continuous monitor	Orthogonal nephelometry (light scattering)
Range	0,001 - 100 [mg /m^3]
Resolution	0,1%
Accuracy	+/- 5%
T90 Response Time	< 10 [s]

Table 23: The dust module technical specifications.

Specification	Description
Operator Display	2 lines x LCD 16 characters
Operating Environment	Temperature: -10 [°C] + 50 [°C] Relative Humidity: 10 – 95 [%], no condensation
Storage Environment	Temperature: -20 [°C] + 50 [°C]
Dimensions	540 x 710 x 310 [mm]
Weight	15 [kg]
Execution	IP55 (outdoor installation)
Mounting Kit	Pole mounting kit
Power Supply	220 [V] – 50 [Hz]
Power Consumption	30 [W]

Output	RS232, ASCII string
Data logger	1 data / s as elementary Storage of hourly / 15 [minutes] average
Remote communication	GSM / GPRS / UMTS with VPN functionalities
Software	@COM3000 remote data handling and elaboration software (report, graphs, statistics, configuration)
Solar panel option	Solar panels + battery regulator for 24 [hours] autonomy. Designed to power an ETL with up to 5 sensors, modem GSM and meteorological station

Table 24: The ETL ONE technical specifications.

Thanks to the active involvement of the **Local Agency for the Environment**, an online interface with the reference air quality stations presented in D.2.1.1 has also been created. In particular, the official NO_2 mean values calculated on a hourly basis related to the two air quality stations positioned at a relevant roadside level (i.e. Adriano Square and Claudia Augusta street) will be transmitted as soon as they are available to Environmental Stations Data Source. A similar approach has been followed for the retrieval of official real-time data concerning the meteorological conditions in the city. In this case, the reference department of the Autonomous Province of Bolzano that has been actively involved for this task has been the **Hydrographic Office**, which manages a broad network of meteorological stations all around the provincial territory. Thanks to this further cooperation, it has been possible to design an online interface to the data coming from the meteorological station of Bolzano. No technical specifications the meteorological measurements instruments are unfortunately available. The map with the existing air pollution and meteorological stations which will be included in the INTEGREEN system is presented in Figure 53.

4.5.1 Air pollution and meteorological data and metadata package

The package consisting of all relevant air pollution and meteorological data and metadata is structured as follows:

Metadata

- ***stationCode***: an alphanumeric identifier of the measurement station (e.g. "*BZ4*")
- ***stationName***: a string identifier of the measurement station (e.g. "*Via Claudia Augusta*")
- ***measurementTypes***: an array of array consisting of all different parameters measured by the station. Each parameter is characterized by:
    - ***elementCode***: an alphanumeric identifier of the measured parameter (e.g. "*NO2*")
    - ***measurementUnit***: a string identifier of the unit of measurement of the

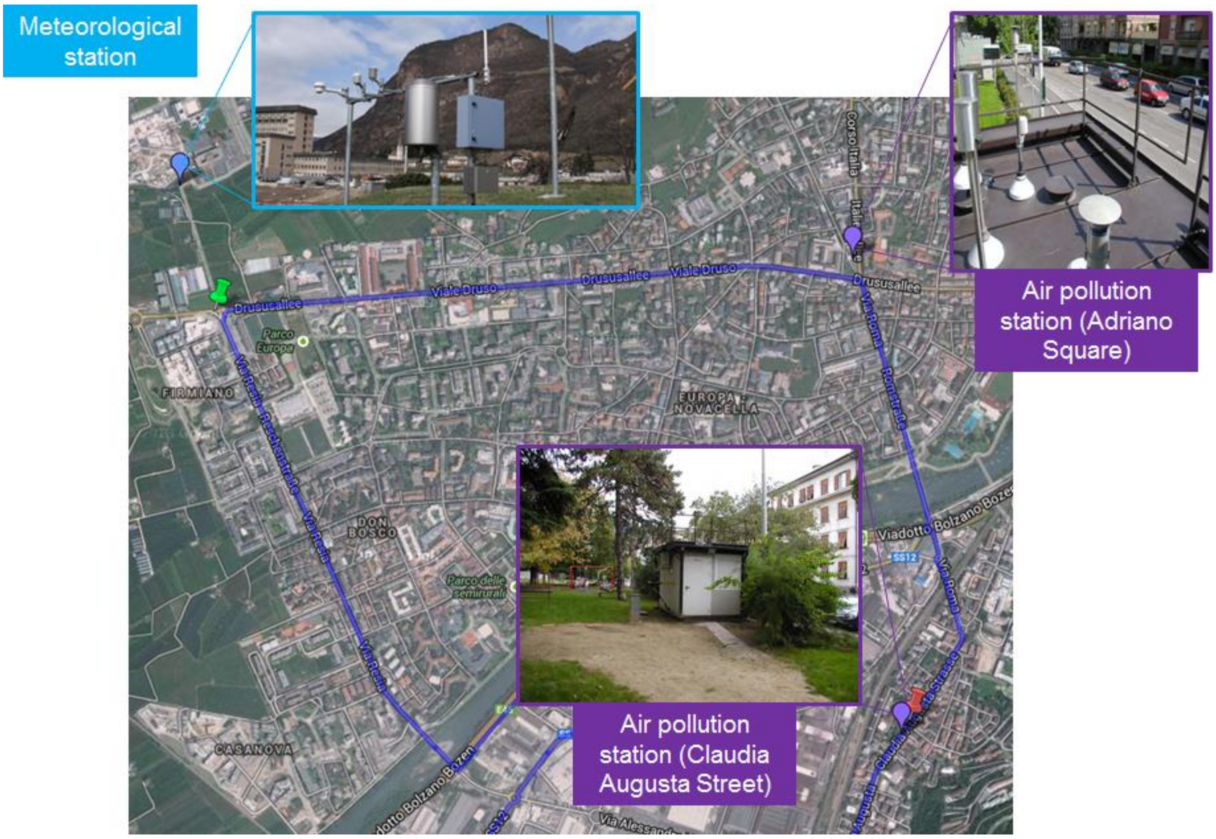

Figure 53: The map with the existing air pollution and meteorological stations included in the INTEGREEN system.

- associated parameter (e.g. "*µg /m3*")
- ***measurementType***: a string identifier associated to the type of measurement which is provided (e.g. "*mean value*")

- **Data**
  - ***date***: the reference measurement timestamp (in epoch format)
  - ***value***: the associated measurement.

This package structure can be considered as reference model for both the ESDS and the MDS, given the similarity of the measurements of interest.

4.5.2 Air pollution and meteorological data real-time transfer service details

In order to design the transfer service of the real-time air pollution measurements carried out by the Local Agency for the Environment, a strict cooperation with SIAG, the in-house company of the Province of Bolzano which actually manages the IT infrastructure for the collection and storage of the air pollution data coming from the provincial monitoring network, has been activated. In this case, based on the technological platform in use by SIAG, the chosen data transfer protocol has been **SOAP** (*Simple Object Access Protocol*). SOAP is a very well-known and used W3C specification which allows distributed software components to exchange messages through XML language independently from the transport layer chosen in the ISO/OSI stack [20]. A SOAP message is organized as presented in Figure 54. The root element is called "**Envelope**" and contains two fields, namely (i) an optional **header**, which contains meta-information covering e.g. routing or security aspects; and (ii) a **body**, which contains the message in strict sense and eventually optional faults which describe the way exceptions must be handled. Optional annexes in MIME format are possible for the exchange of binary data.The transmission of a SOAP message can be directly carried between a transmission entity ("*SOAP sender*") and a receiving one ("*ultimate SOAP receiver*"), or foresee one or more *SOAP intermediaries* which can have different roles, e.g. they can simply route the messages or elaborate their contents.

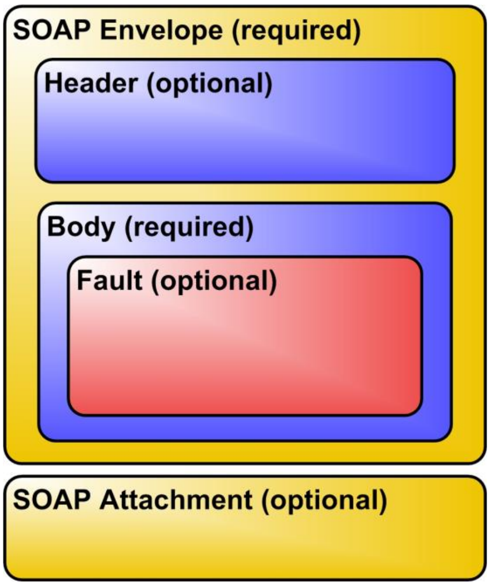

Figure 54: The SOAP message format **[20]**.

A web-service based on SOAP has been defined in order to allow a simple but efficient transmission of the data. The methods supported by the service are presented in Table 25; an example of the XML request / response message is reported below.

Name	Description	Input type	Output type
getHourlyData	Returns the hourly averages of the air pollution measurements. At server side, only the hourly averages of the last 24 [hour] are stored	An array containing two strings(station Code and elementCode) and two and two dateTime.ISO (start and end time)	An array containing: a dateTime.ISO (reference measurement time) and a double (measured value)
getDailyData	Returns the daily averages of the air pollution measurements.	An array containing two strings(station Code and elementCode) and two and two dateTime.ISO	An array containing: a dateTime.ISO (reference measurement time) and a double (measured value)

		(start and end time)	

Table 25: The SOAP methods for the data transfer of the reference NO_2 measurements in the city between the Local Agency for the Environment and Supervisor Centre.

```
<soapenv:Envelope
      xmlns:soapenv=http://schemas.xmlsoap.org/soap/envelope/
      xmlns:rem="http://luft.services.siag.it/remarks_v1.0.wsdl">
<soapenv:Header/>
<soapenv:Body>
    <rem:getHourlyData>
        <stationCode>BZ4</stationCode>
        <elememtCode>NO2</elememtCode>
        <remarkedFrom>1373805560</remarkedFrom>
        <remarkedTo>1373894537</remarkedTo>
    </rem:getHourlyData>
</soapenv:Body>
</soapenv:Envelope>

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
        xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <getHourlyDataResponse
         xmlns="http://luft.services.siag.it/remarks_v1.0.wsdl">
      <remark xmlns="">
        <stationCode>BZ4</stationCode>
        <elementCode>NO2</elementCode>
        <date>1373806800</date>
        <value>16.6361</value>
      </remark>
      <remark xmlns="">
        <stationCode>BZ4</stationCode>
        <elementCode>NO2</elementCode>
        <date>1373810400</date>
        <value>14.0735</value>
      </remark>
      <remark xmlns="">
        <stationCode>BZ4</stationCode>
        <elementCode>NO2</elementCode>
        <date>1373814000</date>
        <value>13.1344</value>
      </remark>
    </getHourlyDataResponse>
   </s:Body>
</s:Envelope>
```

As far as the service for the real-time transfer of the meteorological data from the Hydrographic Office is concerned, it has been necessary to use as data format a very particular one called **ZRXP**, a line-oriented text file format developed by the company Kisters AG. A file in the ZRXP format consists of one or more segments (blocks), whereby each segment is divided into basic data and time series data (values), introduced by a basic data header. An example of a ZRXP message in the reference use case application is reported below. The lines which start with "##" indicate lines with optional parameter or a comment; lines starting with "#" contain mandatory fields.

```
## Exported ZRXP Block for 30_a.Realtime
#SNAMEETSCH BEI BRANZOLL;*;SANR8555;*;SWATEREtsch;*;CNR252746;*;
##SRW677968;*;SHW142650;*;SPNP226.960;*;SFGEBIETEtsch;*;
#CNAMEQ;*;CMW48;*;CTYPEn-min-equi;*;CUNITm3/s;*;
#RTYPEMomentanwerte;*;RORPRproduktion;*;
## ParameterNr=252746
#XVLID555QQ0030AREALTIME;*;REXCHANGE8555QQ0030AREALTIME;*;
```

The interpretation of this header is the following. Please note how there is no empty space between the reference field and its associated value(s).

- SNAME**ETSCH BEI BRANZOLL**: the field "SNAME" is associated to the name of the meteorological station.
- SANR**8555**: the field "SANR" is associated to a specific alphanumeric code related to the meteorological station.
- SWATER**Etsch**: the field "SWATER" is associated to the name of a reference river.
- CNR**252746**: the field "CNR" is associated to a channel number.
- SR**W677968** and SHW**5142650**: the fields "SRW" and "SHW" indicate latitude and longitude of the meteorological station by taking in consideration the UTM WGS84 reference system.
- SPNP**226.960**: the field "SPNP" indicate the altitude of the meteorological station (in meters).
- SFGEBIET**Etsch**: similarly to SWATER, this field indicates the zone of the basin in which the meteorological station is located.
- CNAME**Q**: the field "CNAME" indicates the reference measurement type. All possible values (with associated codes) are reported in Table 26.
- CMW**48**: the field "CMW" indicates the number of daily measurements.
- CTYPE**n-min-equi**: the field "CTYPE" indicates how the measurements are taken in the time domain. The typical value is "n-min-equi", i.e. equidistant.
- CUNIT**m3/s**: the field "CUNIT" indicates the unit of measurement.
- RTYPE**Momentanwerte**: the field "RTYPE" indicates the type of measurement which is reported, which can be the max or min value ("*Max*" or "*Min*", respectively), the

average (“*Mittelwerte*”), the sum over a period (“*Summen*”) or the instantaneous value (“*Momentanwerte*”).

- RORPR**produktion**: the field “RORPR” indicates if the measurements are original (i.e. with or without any pre-processing task, “produktion” and “original”, respectively).
- XVLID**555QQ0030AREALTIME**: the field “XVLID” indicates the ID in the database.
- REXCHANGE**8555QQ0030AREALTIME**: the field “REXCHANGE” indicates the number of the data exchange in the database

Code	Meteorological parameter
Q	Flow rate
LT	Air Temperature
TD	Dew point at 2 [m] height
N	Precipitation
W	Hydrometric level
WR	Wind direction
WG	Wind speed
WG.Boe	Wind speed gust
LF	Relative Humidity
LD	Atmospheric pressure
HS	Snow height
SD	Solar radiation duration
GS	Global radiation
GS.OBEN	Direct global radiation
GS.UNTEN	Reflected global radiation
OFT	Ground temperature

Table 26: Specification of all possible meteorological measurements managed by the meteorological stations of the Hydrographic Office of Bolzano.

An example of the ZRXP data is the following:

```
20040325130000 0.000000 131080

20040325133000 75.502744 131072
```

```
20040325140000 -777 268435456
```

The first value is the **timestamp**, in the format YYYYMMDDhhmmss. The second value is the **measurement**; a reference code (-777) indicates missing measurement. The third value is a **flag** indicating possible problems concerning the given measurement.The ZRXP file is going to be regularly published and updated on a limited access web site. The MDS will need to have a software client capable of regularly connect to this web site, get the data and parse the ZRXP file according to the given specification.

As far as the air pollution and meteorological data gathered by the instruments purchased through the project, again a simple data exchange service based on XML-RPC is considered. The structure of this service is designed by taking into account the other XML-RPC services and the data and metadata package defined before, and can handle both air pollution and meteorological values.

Name	Description	Input type	Output type
EnvironmentalWebService.getStations	Returns the full list of environmental stations sets. Each set contains: (i) an id; (ii) the name of the environmental station; (iii) a set containing the types of measurement supported, which contain (i) the name of the environmental parameter, (ii) the reference measurement unit and (iii) the type of measurement reported (min, max, mean, sum, instantaneous).	-	Multidimensional array. Each “node” array contains: (i) int; (ii) string; (iii) array, containing three strings.
EnvironmentalWebService.getData	Returns the set of environmental data collected in the given window observation associated to a specific measurement station and parameter.	A int (environmental station identifier), a string (measurement parameter identifier) and two dateTime.ISO (start and end time)	Array of doubles: all measurements taken in the given window observation

Table 27: The XML-RPC procedures for the real-time environmental data transfer between Traffic Control Centre and Supervisor Centre.

4.6 Operators Data-Source

Other potential information that could be integrated at the Environmental Supervisor Centre is those related to the presence of road works and more generally to city events which may disrupt the normal traffic circulation in the city. The most relevant requirements defined in Action n.2 are reported in the tables below.

ID	ODS_1
Name	Data type
Description	Each operator notification must be associated to one of the following categories: • roadworks; • general city events.
Rationale	Characterization of possible data types that can be introduced by authorized operators.
Type	Functional
Priority	Must

Table 28: ODS Data-Source: requirement ODS_1 (data type).

ID	ODS_3
Name	Operator role
Description	The additional information can be introduced in the data-source only by one or more operators having specific privileges.
Rationale	Security
Type	Non-Functional
Priority	Must

Table 29: ODS Data-Source: requirement ODS_3 (operator role).

4.6.1 Experimental system for road works information collection

At present, there is an internal initiative by the Municipality of Bolzano for the creation of an automatic GIS platform called **AGIT** ("*Atti per la gestione unificata del territorio*") able to collect and store electronically all the requests for the occupation of the public land. This platform will be composed by:

- a **(GIS) client application**, in which selected public servants of the Municipality of Bolzano can insert and/or visualize (i) all the temporary grants to use the public land, including information such as nature of request, spatial area affected, duration etc. and (ii) all the temporary warrantees concerning deviations and/or limitations in the normal road circulation;

- a **(GIS) server application**, which is in charge to properly store and publish all geo-referenced public land use information.

Thanks to this platform, it will be possible to get a map-based overview of the areas affected by these requests, including those having an impact on road traffic. Given these premises, AGIT is already able to satisfy most of the native source requirements. The idea of INTEGREEN is to investigate the possibility to have a service which the Supervisor Centre can use through its ODS to continuously integrate relevant set of road works and city event information.

4.6.2 Road works data and metadata package

INTEGREEN is interested mostly in information that can have an immediate impact on normal traffic flows. This are typically construction sites that are directly built inside the road infrastructure, which produce as a consequence a reduction of the nominal road capacity. Indeed, even construction sites opened outside the road infrastructure can significant consequences on urban traffic, since they can alter the mobility demand of a specific centroid.

In light of these considerations, the reference data package can be the following:

- "**activities**", i.e. human actions external to the traffic stream or roadway which could disrupt traffic (e.g. public events such as fairs, concerts, soccer games, etc. but also construction building far from the road network).

- **road works**, i.e. maintenance, installation or construction activities that are located directly or very near the road infrastructure, and that have an immediate impact on traffic;

- **network management information**, i.e. changes to the configuration or usability of the road network, whether by legal order or by operational decisions.

Each data package must be supported by a set of reference metadata, including (ref. ODS_2):

- all relevant **details** characterizing one of these events (e.g. for network management information, the type of vehicles which cannot circulate);

- the **temporal duration** and the **spatial extension**, which can be modeled as 1D (line) or 2D (area) object depending on its nature.

4.6.3 Road works information real-time transfer service details

At this design stage the approach is to establish a standard **OGC-compliant service** (e.g. **WFS**) that the INTEGREEN system could integrate through the ODS in order to get the up-to-date information of interest. This in light of the decision of the Municipality of Bolzano to publish all AGIT information through a GIS client such as GeoServer using one or more of the supported OGC services.

Since the implementation of the AGIT platform is still currently on-going, the perspective for the project is just to put the premises and/or initially test this kind of integration, which is going to be likely finalized and fully exploited after the project's end. For example, at this stage it is not fully clear how to cover mapping of all available data set into the three identified information categories, whether directly at the AGIT system side or at the ODS

4.7 Other data-sources

In the INTEGREEN architecture, three additional data sources have been introduced:

- the **3rd Parties Data Source** (3PDS), which is in charge to gather measurements and elaborated data from other centers;

- the **Video Data Source** (VDS), which is expected to collect notifications coming automatically the from automatic processing tools of video-surveillance system;

- the **User Data Source** (UDS), which intends to receive the notifications coming directly from the local travelers through one or more end-user applications.

These data sources represent together with the ODS an added value for INTEGREEN, to be implemented following a best-effort approach, since they can improve the amount of data that such a Supervisor Centre can use in order to monitor the current conditions of traffic in the city. A perspective of the design activities carried out for each of these components and of their potential implementation is given in the next sub-paragraphs.

4.7.1 3rd Parties Data-Source

The most important requirements for the 3PDS are reported from D.2.1.1 in the following tables.

ID	3PDS_1
Name	Data type
Description	Each single received data must be linkable to one of the following categories: • traffic; • parking; • air pollution / emissions; • general events; • roadworks: • accidents; • road conditions; • public transport; • others.
Rationale	Characterization of possible data types received by 3rd parties sources.
Type	Functional
Priority	Must

Table 30: 3PDS Data-Source: requirement 3PDS_1 (data type).

ID	3PDS_4
Name	Standard data transfer
Description	The data provided by each 3rd parties source must be exposed with a standard data exchange protocol (e.g. DATEX II, SIRI, VDV, etc.).
Rationale	Standard interoperability
Type	Interface
Priority	Must

Table 31: 3PDS Data-Source: requirement 3PDS_4 (standard data transfer).

The most relevant 3rd parties for INTEGREEN are mainly three:

- the **Autostrada del Brennero S.p.A.**, the organization managing the A22 toll highway;
- the **Road Department of the Autonomous Province of Bolzano**, managing the data of the regional road network;
- the company **SASA**, the urban public transportation operator.

Thanks to the involvement activities carried out in Task 6.3, the premises have been put for the inclusion at the Supervisor Centre of the data package defined in requirement 3PDS_1 from these 3rd parties. It is worth noting SASA could provide not only information concerning **current limitations in the public transportation service** they offer (e.g. strikes, temporary bus lines changes, etc.) through this data source, but also real-time information coming from the AVM system through the Vehicle Data Source (see D.3.1.2 for more details on this).

The considered data transfer technologies will be:

- **DATEX II** for the exchange of data will other traffic control centers;
- **VDV** for the exchange of data with **public transportation operators**;
- eventually other standard protocols such as SOAP for the exchange of historical information managed in separated databases.

As far as the implementation perspectives are concerned, it is planned to (i) put the premises for an active DATEX II data exchange with the A22 highway, and (ii) give the 3PDS the ability to include the information of SASA concerning temporary limitation of their service.

4.7.2 Video Data-Source

The most important requirement for the VDS is reported from D.2.1.1 in the following table.

ID	VDS_1
Name	Data type
Description	Each notification must be associated to one of the following categories: • accident; • traffic jam; • other events.
Rationale	Characterization of possible data types received by the video source.
Type	Functional
Priority	Must

Table 32: VDS Data-Source: requirement VDS_1 (data type).

The VDS will be probably not fully developed within INTEGREEN, since at present there is no elaboration processes running on top of the current video streams gathered by the video-surveillance network which could provide this kind of information through a real-time automatic service. Some specific tests could however be specifically organized during the testing and validation activities of Action n.5, depending on the progress and the results of the overall verification process.

4.7.3 User Data-Source

The most important requirement for the UDS is reported from D.2.1.1 in the following table.

ID	UDS_1
Name	Data type
Description	Each user notification delivered to the data-source must be associated to one of the

	following categories: • accident; • traffic jam; • other general event.
Rationale	User notification minimum set of data
Type	Functional
Priority	Must

Table 33: UDS Data-Source: requirement UDS_1 (data type).

In this case, there are concrete opportunities to test an early implementation of a similar application thanks to at least one of the RTTI application which is actually already under test in Bolzano, in particular, the demo applications developed within the European project **Co-Cities**, funded by the CIP-ICT PSP programme of the EC, which has developed an automatic mechanism for integrating the feedbacks coming from the users into a RTTI data storage [21].

In INTEGREEN, the intention is to “close this bi-directional loop” with the local travelers by creating an online **interface with this central collection point**, in order to actively integrate the notifications and the comments directly sent by the end-users. The perspective of this would be to have an extended coverage of the current traffic (and environmental) conditions, since users could actually act as “probes” themselves, and to decrease the time for detecting a particular event (e.g. an accident), in particular in areas that are not covered by fixed / mobile detection units. Through the Co-Cities application (available for both Android and iOS devices), travelers can provide notifications about the presence of (i) **abnormal traffic**; (ii) **accidents**; (ii) **obstructions** (i.e. obstacles on the roadway); (iv) **road weather events**; and (v) **road works**.

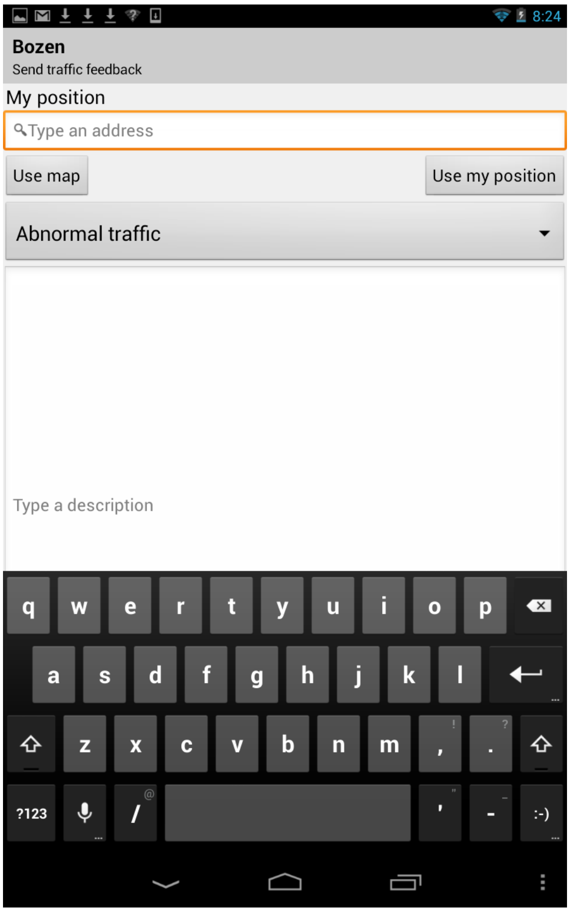

Figure 55: The Co-Cities Android application (traffic notification interface).

4.8 Other data-sources

All this set of data sources design activities can be well summarized by the graphical scheme reported in Figure 56, where all the planned channels activated at this layer are illustrated, including a qualitative indication of the temporal implementation perspectives. It is worth noting there that several further roadside ITS installations are planned at the Traffic Control Centre of the Municipality of Bolzano, which are indicated in this reference architecture.

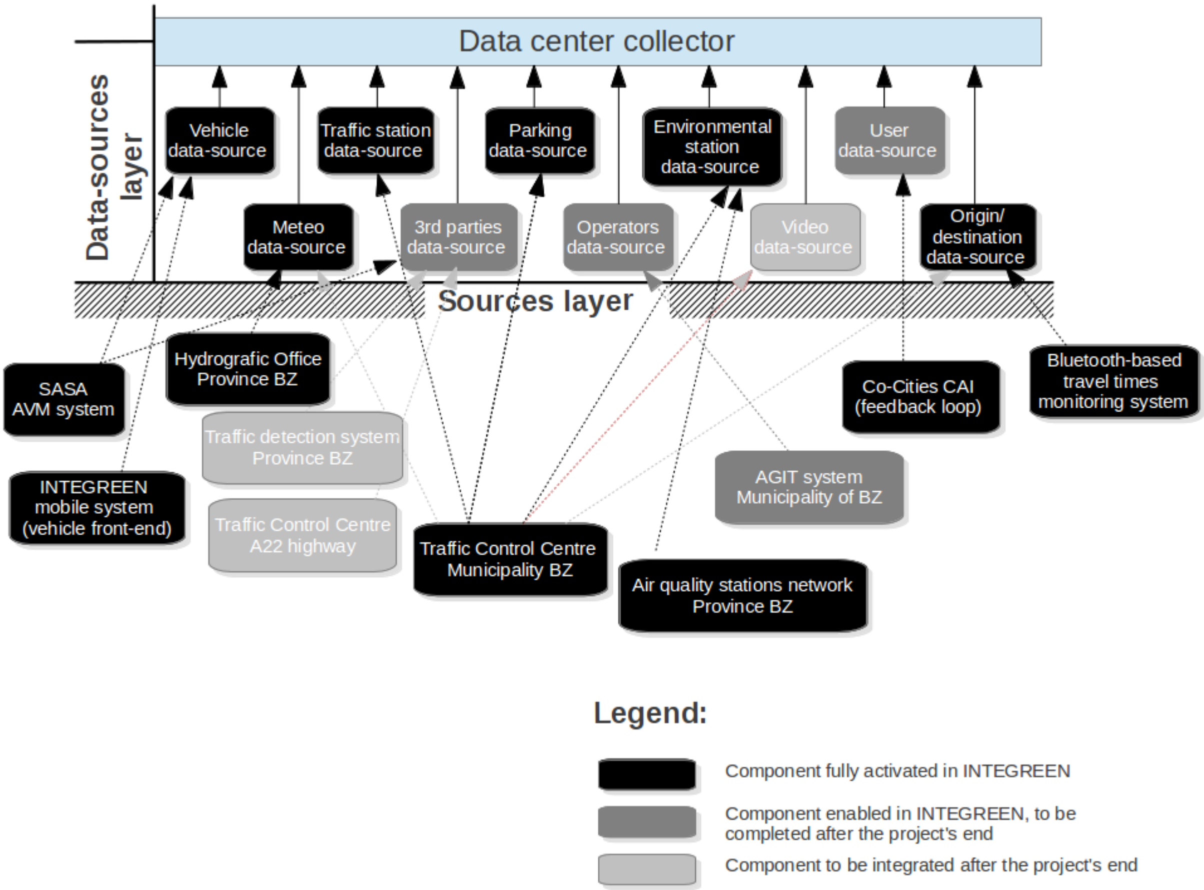

Figure 56: An overall overview of the design choices at the data sources layer.

5 Data Center Layer design

The Data Center Layer is the core engine of the Supervisor Center. It plays a twofold role, since on the one hand it is in charge to apply the proper elaboration activities/routines onto the coming data gathered by the several data-sources connected through the Data Center Collector, while on the other hand it is in charge to make available both the validated generated data and the elaborated ones to the several front-ends through the Data Center Dispatcher. In this latter case, further elaborations might be necessary in order to respond to the coming queries.

Further requirements that must be considered in the design of this core layer are in terms of (i) **security**, i.e. sufficient protection measures must be planned in order to prevent any kind of external attacks, (ii) **performance**, in order to guarantee the “real-time” constraints of the system, and (iii) **flexibility and scalability**, since it should be easily possible in the future to significantly extend the type and amount of measured and elaborated data.

All different data flows entering from the data sources layer are centrally managed in this environment, which is composed mainly by a database, where all the raw validated data are stored. Three core elements interact directly with the database: on one side the Data Center Collector and Dispatcher, which have the fundamental role of providing to external entities an abstract and controlled view of the internal database (to the data source and to the front-ends, respectively), and on the other sides a set of “elaboration tasks”, i.e. automatic routines that are in charge to process the data and extract relevant information related to the current traffic and air pollution conditions.The first step of the design process applied to this specific application domain is to define a **reference implementation architecture** able to properly manage this complex mechanism of accesses to the database. Collector and dispatcher have a specular role, as graphically illustrated in, with the possibility to have two logically-separated software components but capable to share the same software architecture (e.g. libraries) within a common environment.

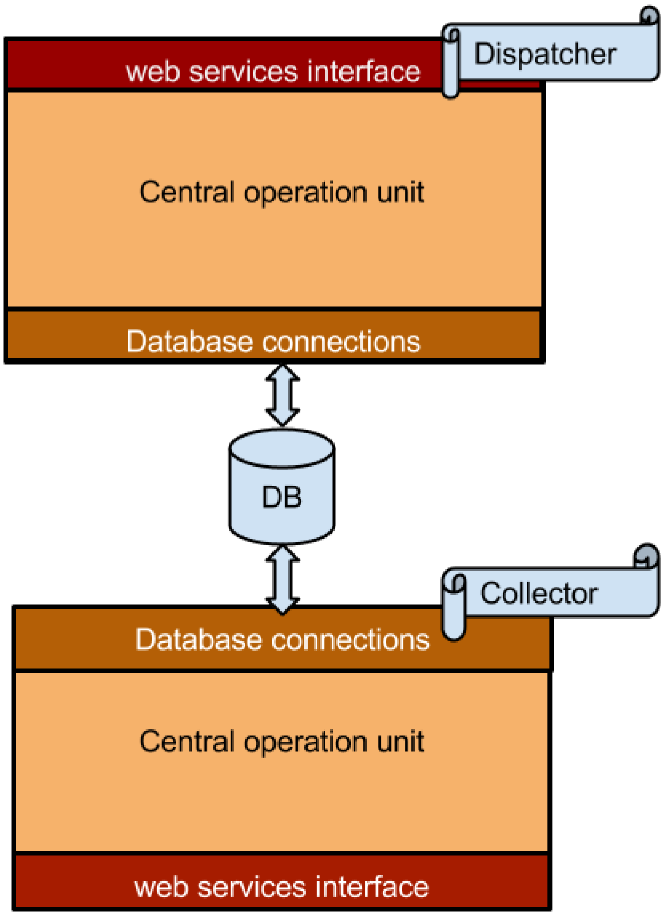

Figure 57: The specular architecture of Data Center Collector and Dispatcher.

From a high-level point of view, it is first of all important to indicate how the entire Supervisor Centre is going to be managed from a server point of view. Two main design alternatives have been investigated during this phase:

- an **architecture schema with a load balancer**, with three servers, one database and two replicators, and three application servers, one of which with a load balancer able to redirect the requests to the other two application servers (Figure 58);

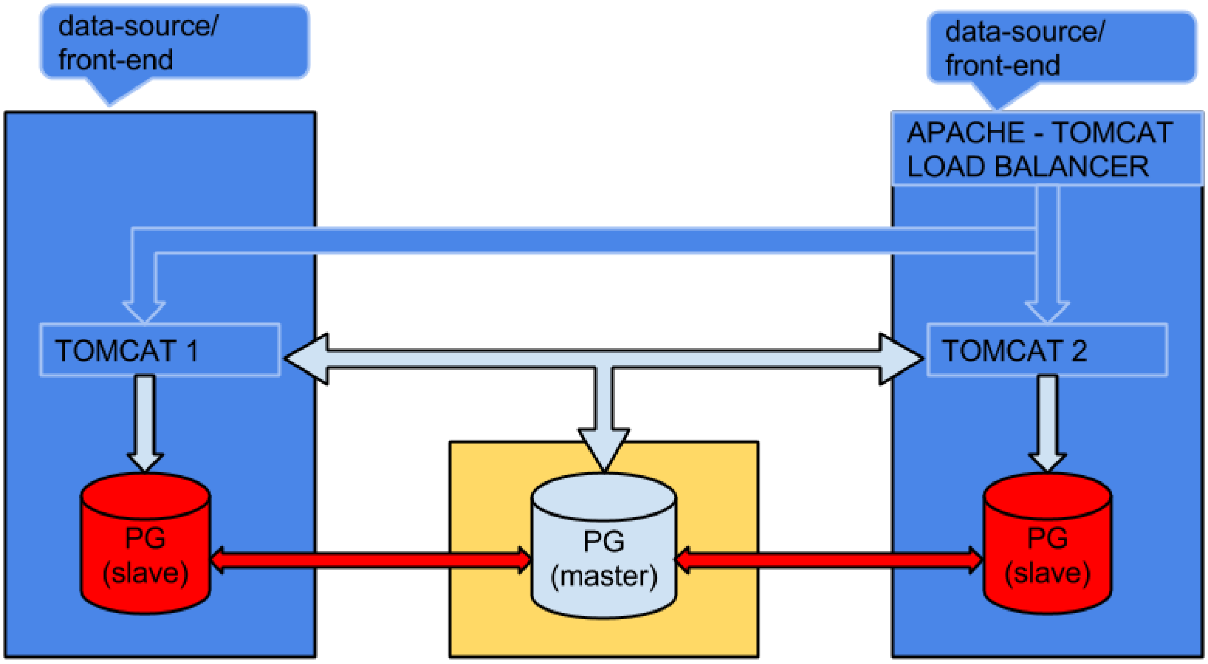

Figure 58: The Data Center Layer architecture schema proposal nr. 1 with a load balancer.

- an **architecture schema without a load balancer**, with two servers, one database and one replicator (in read-only mode), and one application server (Figure 59).

Being the second architecture less complex to manage and characterized by a reduced number of physical components, **the decision has gone to this second approach**. More complex approaches could be exploited once the amount of data to be managed will grow. This choice considers two databases, a "master" one which is accessible by the elaboration tasks in read/write mode only, and a "slave" one, which is accessible by the data center collector and dispatcher in read-only mode. The two databases, to be hosted in two different servers, must continuously synchronize in order to guarantee at any time the freshness of the available raw data and elaborated information at both sides. This master / slave structure can be automatically maintained thanks to the different automatic features which is guaranteed by several Database Management Systems (DBMS), for example the "***streaming replication***" one supported by Postrgres, which is going to be the choice for INTEGREEN.

This choice is optimal from different points of view, even from a scalability and security perspective; e.g. thanks to this approach, it is immediately guaranteed the impossibility for the dispatcher to provoke any changes in the contents stored in the database, and to limit its function to a simple forward of relevant information at the upper front-end layer. It is worth

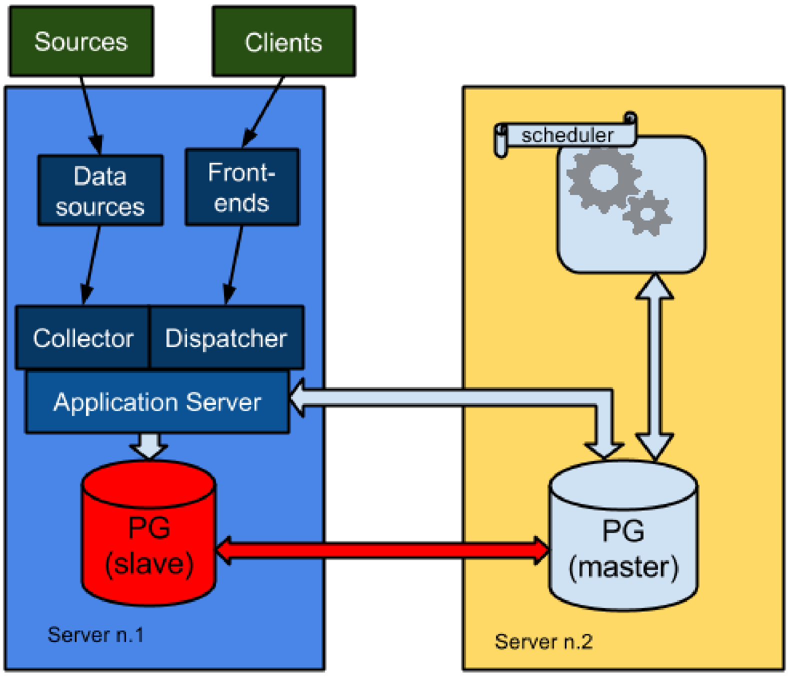

Figure 59: The Data Center Layer architecture schema proposal nr. 2 without a load balancer.

noting that this kind of architecture poses the question on how properly satisfy some requirements introduced at both data source - and front-end layers, in particular the **<u>need to have all the components properly "isolated"</u>** (ref. DSL_2 and FEL_2). Data sources and front-ends are no more than pieces of software; they require and rely on several entities to be executed such as application libraries, database connections, operating system libraries, runtime tools and an operating system. In a common environment these software entities are executed and shared by several applications; moreover they can be seen as placed into different abstraction levels as shown in Figure 60. At the lowest level there is the physical network in charge of connecting physical servers. Above this it is possible to find the real operating system which is in charge of providing the necessary abstractions to allow independent virtual machines to be executed. Each virtual machine has got its own guest operating system which involves several libraries and applications. The application server is one of the several application running on top of the guest operating system, this software is designed to allow several and independent applications to be executed and be able access the underlying levels.

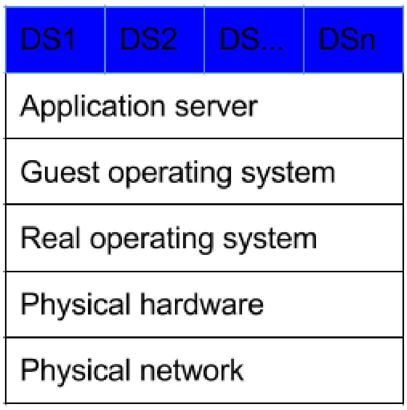

Figure 60: Supervisor Centre software components isolation.

As far as the choice of the **application server** is concerned, three different competitive open source alternatives have been taken into consideration, namely JBoss, Glassfish and Tomcat. An initial empirical comparison of the different comparison has been carried out in order to specifically analyze advantages and disadvantages of the different alternatives. Useful indications have been collected by considering the extensive report of Rebellabs [22]. For the deployment purposes of the Supervisor Centre of INTEGREEN the most important implementation requirements are:

- possibility to make quick and dynamic system components **deployments**;
- minimize problems of **out of memory**;
- easy integration with **GeoServer** application which is used at the front-end layer (more details on this in deliverable D.3.1.2);
- web-based **application server manager** that could be easily used by non-core developers for restarting specific system components in case of issues;

The results of this initial comparison are summarized in Table 34. Performance tests take into account start-up / restart time, deploy time and application initialization. Despite the well-known out of memory problems of Tomcat, its excellent interoperability with GeoServer which is unfortunately not observed e.g. for JBoss has led to the decision to use as application server **Tomcat**, and not JBoss which has proofed to be probably the most reliable solution.

Feature	JBoss	Tomcat	Glassfish
Download and installation complexity	**Easy**. Dimension: 127 [Mbytes]	**Very easy**. Dimension: 12.8 [Mbytes]	**Easy**. Dimension: 30 - 50 [Mbytes]
Tooling support	**Very high**. Good cross IDE support, able to make light server configuration changes.	**Very high**. The big three IDEs, Eclipse, IntelliJ IDEA and Netbeans, all have integration support for Tomcat out the box.	**High**. Info about where to find plugins is poor, great IDE support once the plugins are found.
Server configuration	**Easy**. Single configuration file (300 XML lines). Well structured “subsystems”. It uses a command-line interface for reloading operations.	**Easy**. Restarts required for configuration changes, scattered across multiple files, small file, easy to update, nice examples in comments	**Not Easy**. One should have advanced system administration competences for this…
Performance	**Good**.	**Very good**.	**Good**.
Open standards compliance	**Excellent.** Full Java EE compatible.	**Sufficient.**	**Excellent.** Full Java EE compatible.

Application Manager	**Very good.** Enterprise quality administration, reload/restart information, need to configure users before use, extensive feature configuration, modern	**Sufficient.** Simple, limited functionality console.	**Excellent.** Clean and simple administration console, not extensive feature configuration.
Costs & licensing	**GNU LGPL**. Free with commercial support.	**Apache 2.0 / EPL**. Free but no commercial support	**CDDL 1.1**. Free with commercial support (Oracle GlassFish Server).

Table 34: A comparison of advantages and disadvantages of different Java Application Servers.

Another important aspect of this design process has been the individuation of a set of reference tools capable to monitor in real-time the performance and proper functioning of the Supervisor Centre, and in some case of the remote field units as well (e.g. Bluetooth detectors). The basic tools considered for accomplishing these tasks have been:

- **Ganglia**, a scalable distributed monitoring system for high-performance computing environments developed and maintained as open source project by the University of California, Berkeley [23]. Ganglia is going to be considered in order to remotely monitor the performance of the hardware equipment of the Bluetooth detectors and to continuously improve their behaviour thanks to this monitoring capability. Ganglia can monitor parameters such as the amount of available RAM, disk usage, temperature of hardware components but also statistical information about specific tools such as for example the number of query executed by the central database unit in a given time frame

- **Nagios** and **Pingdom**, which are two well-known open source tools capable to monitor the status and the performance of a server environment. While Nagios will be used for advanced system administration tasks, Pingdom is going to provide very immediate alerts concerning the availability of the external web services, i.e. at the data source and front-end layer side;

Moreover, the logic of a tool called "**TestingMachine**" developed inside a local ERDF project coordinated by TIS for the automatic testing of web-services in the e-government domain is going to be adapted for the project needs in order to verify the proper availability of the multiple web-services that are used at the Supervisor Centre. For more information about this project please refer to the web-site https://testingmachine.eu/.

5.1 Data Center Collector

The Data Center Collector (DCC) is in charge to provide a single and unique point with whom all the data-sources must interact in order to deliver the data which they gathered from all their respective sources. In addition to this, it has the role to interact with the database and to store the data in an appropriate manner.

5.1.1 Functionalities design

The DCC is a unique software entity that must be able to:

- identify all **data sources** providing the data and the associated sources in a secure way, i.e. through proper authentication mechanisms;
- establish a secure connection to the **database** and properly store the received data according to its data model.

The DCC must be able to handle the data reception process independently from the number of data sources providing the data, and all this without producing delays in the generated data transfer.

Implementing these functionalities is not that complex given the today's computer science technologies. Secure connections can be easily managed through proper protocols at the application layer of the ISO/OSI stack, e.g. **SSH** (Secure Shell).

Assuming that all necessary initialization procedures have been already covered (e.g. number of data sources available, metadata shared, etc.), the list of operations that the DCC must implement are the following:

- opening a secure connection with both the data sources and the database;
- retrieve the data from the data sources;
- store the data in the database.

5.1.2 Data transfer protocols

Since most of the data sources get the data using the **XML-RPC** protocol, the decision has been to consider this protocol even at this stage in order to cover this communication link. In this way common functions and software entities can be shared with the possibility to significantly improve the efficiency of the overall Supervisor Centre chain implementation.

5.2 Database

The Database (DB) is the core of the Supervisor Center, where all the information, data, output results gathered and elaborated at this layer are stored. A DBMS with GIS extension is needed in order to properly cover all the requirements defined in Action n.2.

Different alternatives are today available at the state-of-art. Most of them are proprietary, and used for commercial applications, for example Oracle Spatial, ESRI ArcSDE or Microsoft SQL Server Spatial. In INTEGREEN, the interest is on open source tools in order to avoid the creation of licenses dependencies and limit the open development of this architecture. One of the most reliable alternative, which between the most popular DBMSs at all, is **PostgresSQL** with its spatial extension **PostGIS**. Another possible alternative, which is however very little used, is MySQL Spatial.

Being PostgreSQL and PostGIS at the base of the GIS reference implementation proposed by **FreeGIS.net**, project co-funded by the program Interreg Italy-Switzerland coordinated by TIS which has introduced a reference system for the standard publication of geo-information through the use of open data formats (e.g. following the guidelines of the INSPIRE directive and the specifications of OGC) and free and open source software components, **the decision has been to use this DBMS for the INTEGREEN Supervisor Centre**. FreeGIS.net is conceptually at the base of the front-end layer choices described in D.3.1.2.

PostgreSQL is an object-oriented relational DBMS available for most operating system. Through its current version (9.2), it guarantees very advanced properties of transactional integrity and disaster recovery, and allows to manage very large and complex data models with high performance. Authentication procedures are available in order to guarantee a secure access to the tables of the database as a function of the privileges given to different user categories. The access can easily take place using standard protocols and programming languages, either or through command line interface (psql) or a more user-friendly GUI (pgAdmin), today available in its version 3. The flexibility and programmability of PostgreSQL are the most relevant properties of this DBMS, which is typically preferred to other open source or proprietary alternatives. PostGIS is the spatial enabling of PostgreSQL, and allows to save and manage spatial data following the OGC Simple Features specifications [24]- [25]. This is possible e.g. in the Java language through the Java Topology Suite (JTS). The advantage of using PostGIS is that an integration with tools like MapServer or GeoServer is immediate. PostGIS puts at disposal types, functions and indexes that allow an easy management of large quantities of spatial objects in both read and write modes. The spatial coordinates of the spatial objects are saved in dedicated tables, called *feature tables*, which can contain just one geometry type (i.e. point, line, polygon and complex types based on these basic elements). More specifically, the coordinates are saved in a field of a special type, WKT (*Well Known Text*). The spatial information is moreover completed by a set of metadata, saved in the dedicated table *geometry_columns*. Indication like the reference spatial reference system can be stored here. PostGIS supports different files types (e.g. Shape, MapInfo, DGN, GML, etc.), which can be read, converted and inserted in the database through proper OGC libraries (Figure 61).

The following example shows how simple a spatial data entry can be in PostGIS. The following lines of SQL commands create the feature table "*user_locations*" and characterize it with two fields, a numeric ID ("*gid*") and name ("*user_name*"). This table is associated with the geometry_columns table called "*the_geom*", which indicates how the spatial data type "point" is associated to the table user_locations in the spatial reference system associated to the ID 4326. Given these premises, one can now easily insert spatial data in the user_locations table by providing the basic two fields and the spatial information requested (Figure 62).

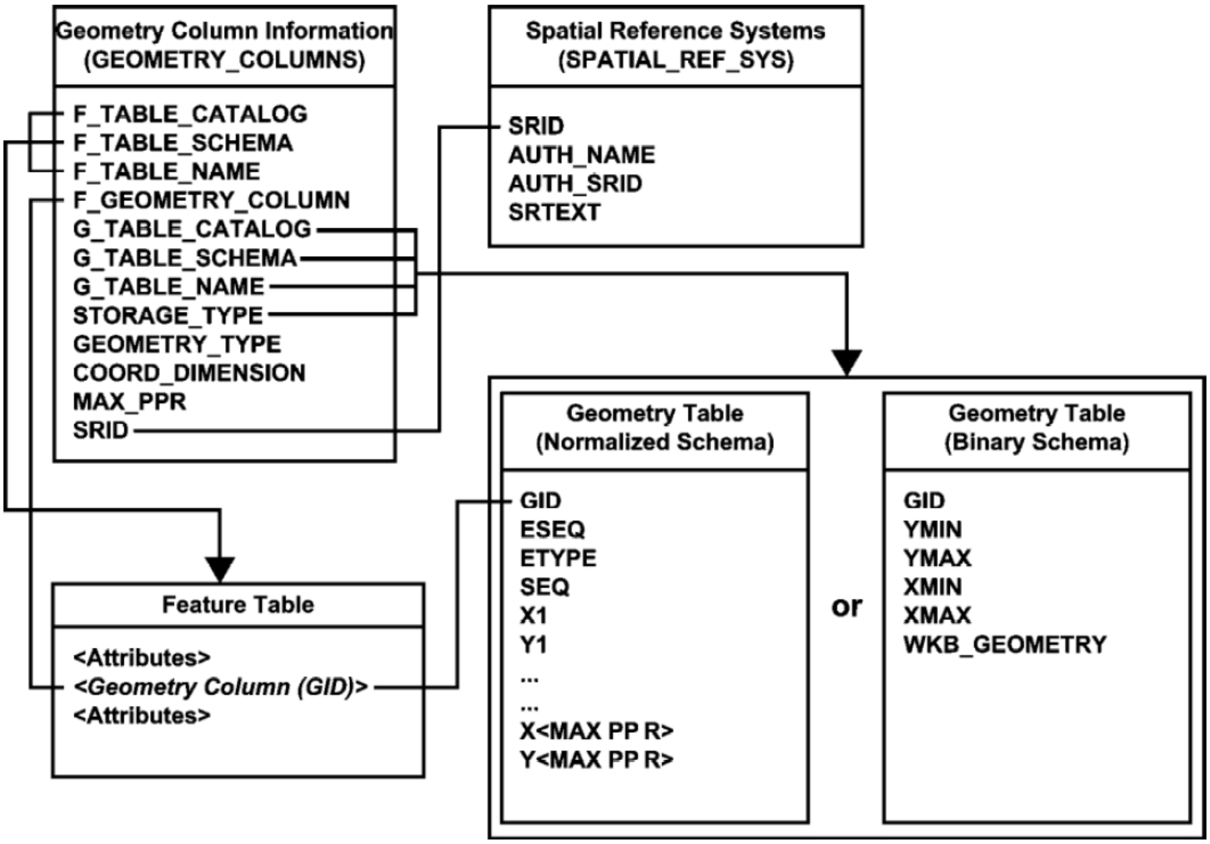

Figure 61: The SQL implementation of spatial database according to OGC specifications **[25]**.

```
create table user_locations (gid int4, user_name varchar);

select AddGeometryColumn
('db_mapbender','user_locations','the_geom','4326','POINT',2);

insert into user_locations values ('1','Morissette',GeometryFromText
('POINT(-71.060316 48.432044)', 4326));

insert into user_locations values ('2', 'Sperb',GeometryFromText
('POINT(-48.6764 -26.8916)', 4326));

...
```

```
f_table_catalog   | spatial
f_table_schema    | db_mapbender
f_table_name      | user_locations
f_geometry_column | the_geom
coord_dimension   | 2
srid              | 4326
type              | POINT
attrelid          | 8751803
varattnum         | 11
stats             |
```

```
gid       | 1
user_name | Sperb
the_geom  | SRID=4326;POINT(-48.6764 -26.8916)
```

Figure 62: Example of spatial data initialization and entry in PostgreSQL and PostGIS (Source: A. Christl: "Introduction to Spatial Data Managament with PostGIS").

5.2.1 Data model

One of the most important and crucial activities of all the design work carried out at the Data Center Layer has been the **definition of database' tables** needed to store all the various data types included in the system, which is considered for both database replications. The two peculiar elements that have driven the final choice of the proposed data structure have been:

- the approach to have as reference a **generic table called "*station*"** which is descriptive of the specific source sending the data, and **common tables for measurements and elaboration outputs**, independently from their particular nature;

- the intention to structure the tables in order to **simplify the "mapping" with the harmonized data model proposed in the aforementioned projects eMOTION, In-Time and Co-Cities**, which is used in INTEGREEN at the front-end layer side (see D.3.1.2 for more details on this aspect);

- the willingness to have a **dedicated table containing the measurements collected by the mobile system**, in order to properly manage the big amount of data that this source is going to put at disposal.

The reference data model is graphically illustrated in Figure 63. The table "station" is the reference table where all relevant metadata for all different station types are stored. One of the most important fields here is "*stationtype*", which indicates the type of monitoring station it is (traffic, parking, environmental, mobile, etc.) through a proper association to the table "*type*". Other additional tables extending table "*station*" have been defined, i.e. *carparkbasicdata* for parking areas information, *meteostationbasic* data for meteorological stations, and *linkbasicdata* for specifying road links on which travel times are computed thanks to the Bluetooth data. Table "station" is associated by default to the spatial element "point", since it is typically associated to a fixed station collecting data from a specific point; however "*station*" extensions can have other geometries, like for example *linkbasicdata* which is associated to a type "line" or *carparkbasicdata* which could be associated to more complex spatial representations (e.g. area-based).

Raw data and elaboration outputs are stored in a set of dedicated tables. Two tables are available for each type in order to store in one all the data history, which could be considered for offline elaborations, and in one the last values. The latter table could be directly accessed from the Data Center Dispatcher when providing the fresh information the front-ends, thus significantly reducing the time for executing the database queries. In short:

- ***elaboration*** (and ***elaborationhistory***), which contains the results of the automatic processing routines (to be more specifically presented in the next chapter);

- ***measurement*** and ***measurementstring*** (coupled with ***measurementhistory*** and ***measurementstringhistory***), which store the raw data coming from the Data Center Collector. In measurement, all the numeric values (double) are stored, while in

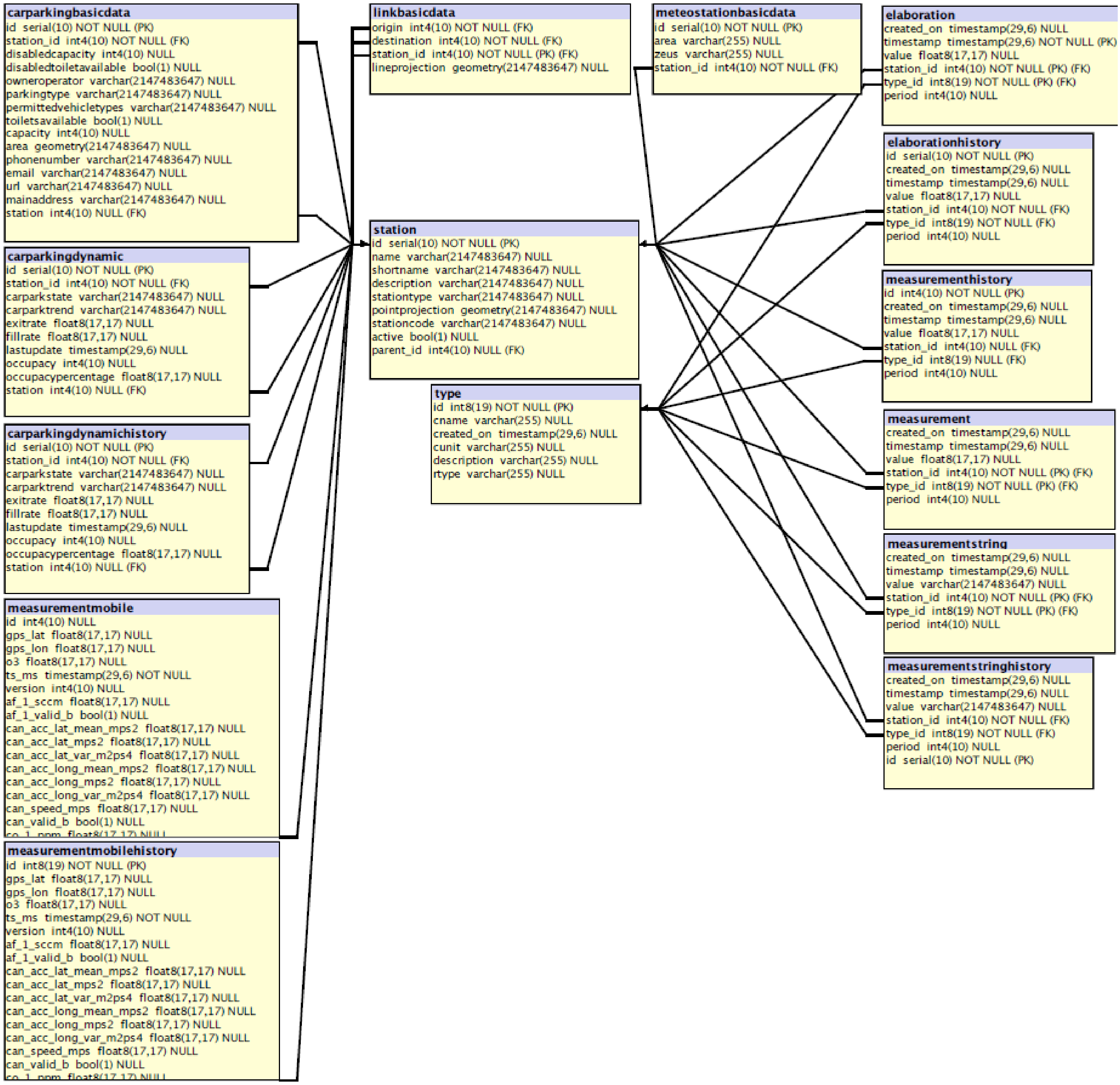

Figure 63: The detailed design of the database tables.

measurementstring all non-numeric values are stored (e.g. the identifiers associated to the Bluetooth detections);

- ***measurementmobile*** (and ***measurementmobilestring***), which store the mobile system raw data coming from the Data Center Collector (according to the data dictionary defined in D.3.1.2);

- ***carparkdynamic*** *(and* ***carparkdynamichistory****)*, which store the real-time occupancy of the car parking areas.

5.3 Data Center Tasks

A crucial part of the entire INTEGREEN system is represented by its set of elaboration tasks, which are in charge to determine a comprehensive overview of the traffic and air pollution conditions in the city. During this design process, the main elaboration tasks have been initially identified and specified, as graphically presented in Figure 64.

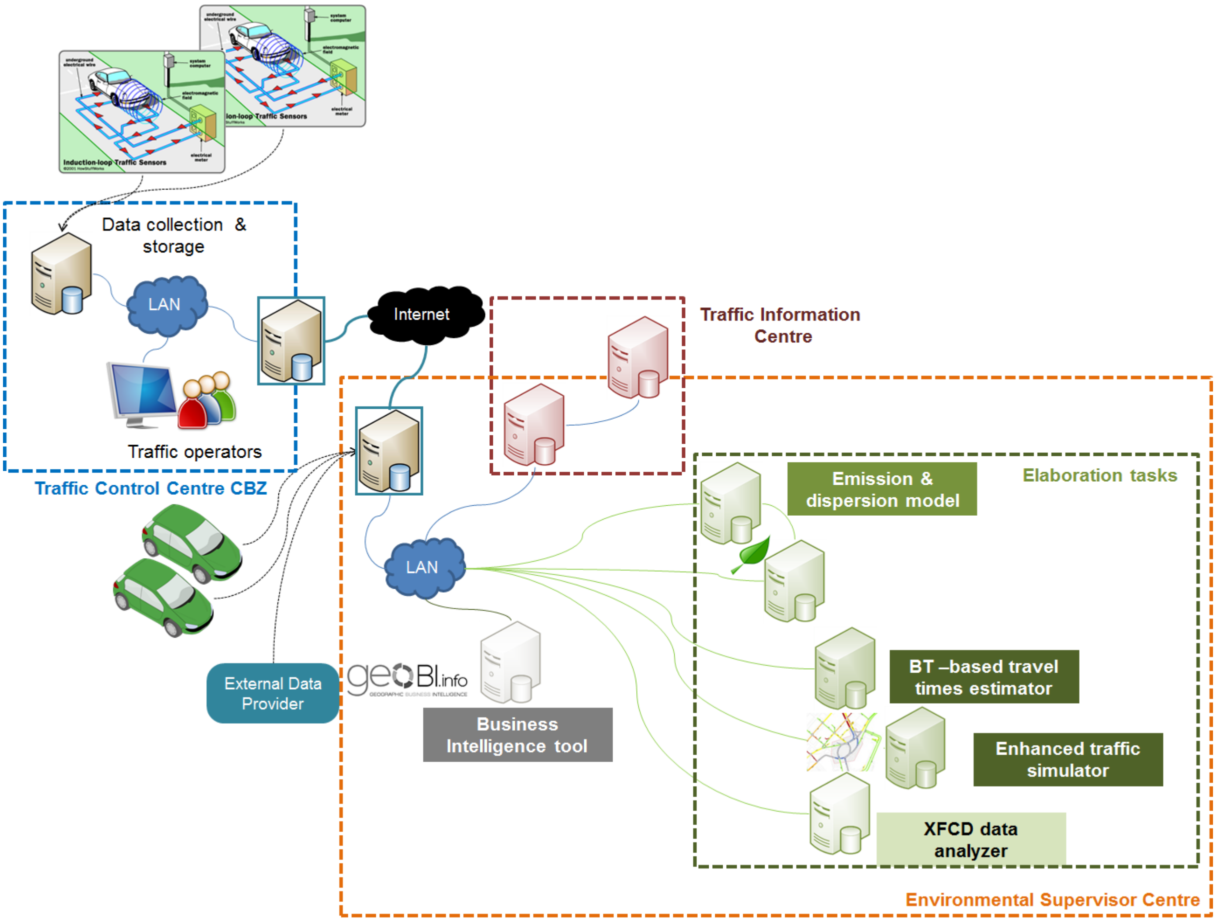

Figure 64: The elaboration tasks located inside the Supervisor Centre of the INTEGREEN system.

These tasks are going to be:

- **the emission and dispersion model**, i.e. an automatic engine capable to estimate, on a real-time basis, the vehicular air pollutant and greenhouse gas emissions and to give an overview of the spatial distribution of air pollution concentrations over the target area;
- **the Bluetooth-based travel times estimator**, i.e. a scheduler in charge of automatically analyze the Bluetooth detections in order to make real-time estimations of the vehicular travel times on specific road stretches of the urban test route;

- **the enhance traffic simulator**, which is going to be a semi-online adaptation of the available offline traffic simulator in use at the Municipality of Bolzano, which will provide updated overviews of the current traffic conditions in the city;

- **the XFCD data analzyer**, which is an automatic engine capable to process on a real-time basis the air pollution data gathered by the mobile system. It is worth noting how an automatic evaluation of the mobile traffic data (basically, position, speed and acceleration in time) – including the one provided by additional sources like e.g. the public transportation vehicles - can provide additional information to the outputs of the Bluetooth-based travel times estimator.

Detailed design considerations for each of this elaboration tasks are presented in the following pages.

5.3.1 Environmental elaboration tasks: emission and dispersion models

As far as the environmental elaborations tasks are concerned, partners have decided to include in the project a specific expertise in the environmental domain in order to properly cover and enrich this central element of the project. Because of previous cooperation activities carried out with the Municipality of Bolzano, and in light of the specific and deep knowledge of the challenges of the targeted urban environment, the decision has been to activate a specific external assistance cooperation with a local engineering company located in the business incubator of TIS (**CISMA s.r.l.**), specialized in environmental modelling and in particular in the evaluation of the dispersion of contaminants into the atmosphere.

The tasks assigned to CISMA (have been in particular the design, implementation and validation of two models capable to work on a real-time basis, namely:

- an **emission model** capable to estimate the current air pollutants' emissions produced by vehicular traffic;

- a **dispersion model** that is in the condition to show the levels of air pollutants over the city of Bolzano, in particular in correspondence of the road network.

The first step of this process has been to define a reference methodology and to select the suitable algorithms for the proper implementation of this elaboration chain, graphically represented in Figure 65. <u>The task of traffic flows assignment will be carried out be the semi-online traffic simulator</u>, as described in the reference sub-paragraph. In this way, the focus is on the selection of suitable approaches for the emission and dispersion modules. This latter one must take in consideration even:

- **meteorological data** such as wind velocity and direction as well the indication of the presence of turbulence phenomena (i.e. the stability class);

- **topological data**, more specifically a digital elevation map in order to properly model the dispersion phenomena happening the in the atmosphere layers just above the road infrastructure.

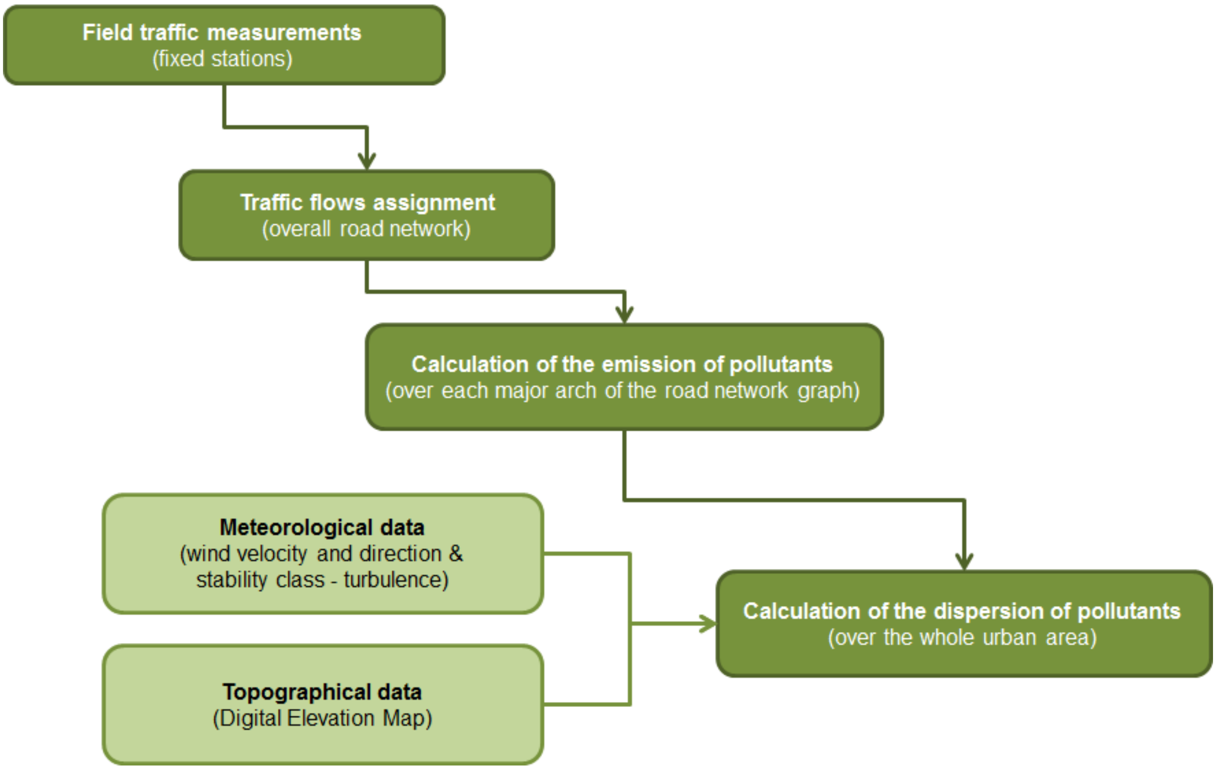

Figure 65: The environmental elaboration tasks chain proposed in INTEGREEN.

Emission model

As far as the emission model is concerned, different methodologies for emission calculation have been evaluated. The most known at the state-of-art are:

- **ARTEMIS**, which is the result of an international project supported by the European Commission which has improved the European methods for estimating and inventorying the pollutant emissions from all transport modes [26]. This is probably the reference methodology available at the state-of-art which provide the highest level of accuracy, but quite difficult to be applied at a practical level (in particular in a real-time environment like INTEGREEN) even because there is no reference to the EURO classification of circulating vehicles;

- **HBEFA**, which stands for Handbook Emission Factors for Road Transport, and is a methodology shared and in use in Austria, Switzerland and Norway which takes in consideration micro-factors such as driving behaviors and therefore of difficult application in the INTEGREEN framework because of the unavailability of this detailed information [27];

- **Mobile6** (recently replaced by **MOVES** as the official model considered by the U.S. Environmental Protection Agency), which is the reference method in use in the North America (USA and Canada) for estimating emissions from cars, trucks and motorcycles, but unfortunately of difficult application in Europe because (again) of the absence of any quantitative reference to the EURO classification of circulating vehicles adopted in Europe [28].

The final choice has fallen on **COPERT**, an European standard for the road-transport emissions calculation whose reference methodology is suggested in the European guidelines "EMEP Guidebook" [29]. One of the advantage in considering COPERT is that this methodology is already considered for the provincial emissions inventory. The reference COPERT methodology is graphically presented in Figure 66.

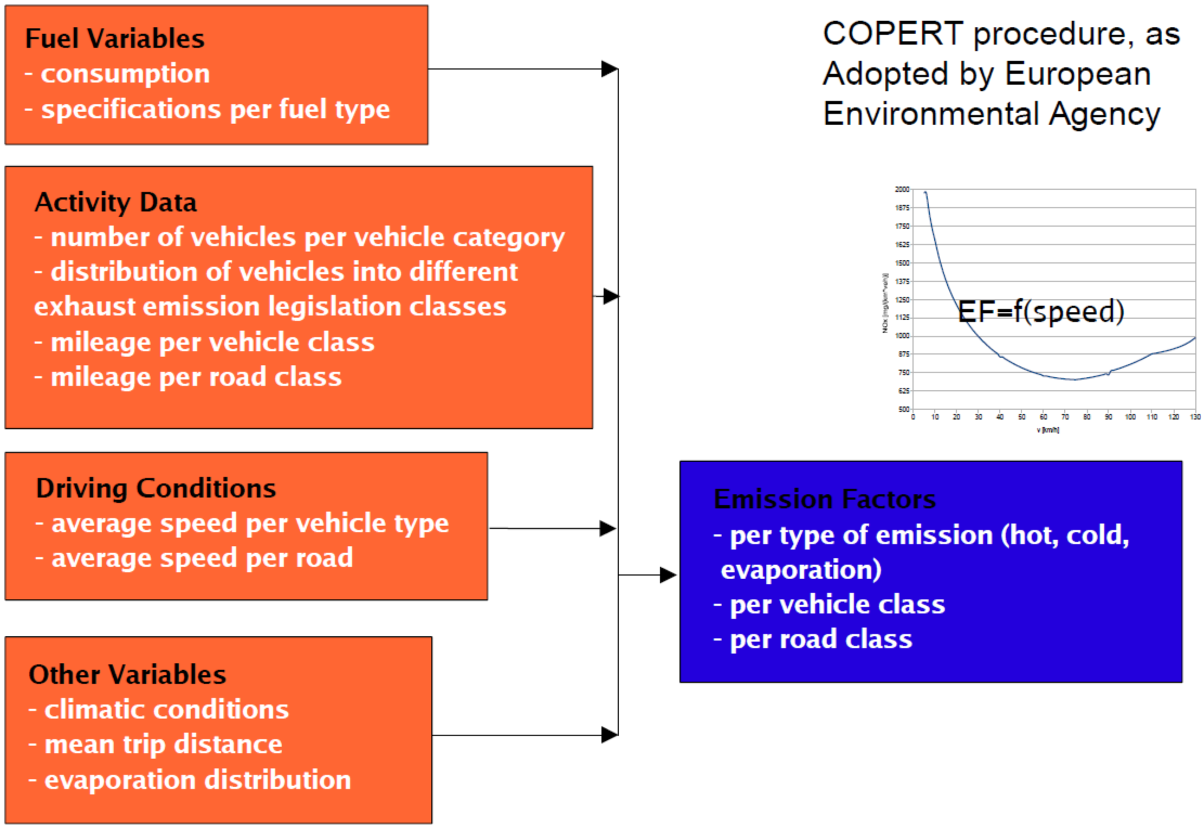

Figure 66: The COPERT methodology (Source: CISMA).

Once the decision to use COPERT was taken, most of the effort within this design action has been to simplify its methodology so that it can be used on a real-time basis, but without any relevant loss in the accuracy of the model. All the emissions factors calculation (expressed in terms of FE_{ij}, where i is the type of pollutant and j the vehicle class) can be indeed expressed through the reference generalized formula:

$$FE_{ij}\ [g/km] = \frac{A_{ij} + B_{ij} \cdot v + C_{ij} \cdot v^2}{1 + D_{ij} \cdot v + E_{ij} \cdot v^2}$$

where v is the vehicle speed and A,B,C,D and E are numerical coefficients depending on the type of pollutant and vehicle class under analysis. So, **given the number and type of vehicles in circulation, and indication of their reference speed, one can easily calculate through this relationship the associated emissions**.

The complex numerical formulations that are at the base of the emission factors calculation in COPERT will thus need to be transformed in analytic formulations, which are much more simpler to manage and quicker to compute. The approach which is needed is graphically presented in Figure 67, in which the two reference situations are presented:

- non-continuous functions (reference situation in the right side);
- step-wise formulas (reference situation in the left side);

The approach is to use proper interpolation approximations in order to minimize the computation error.

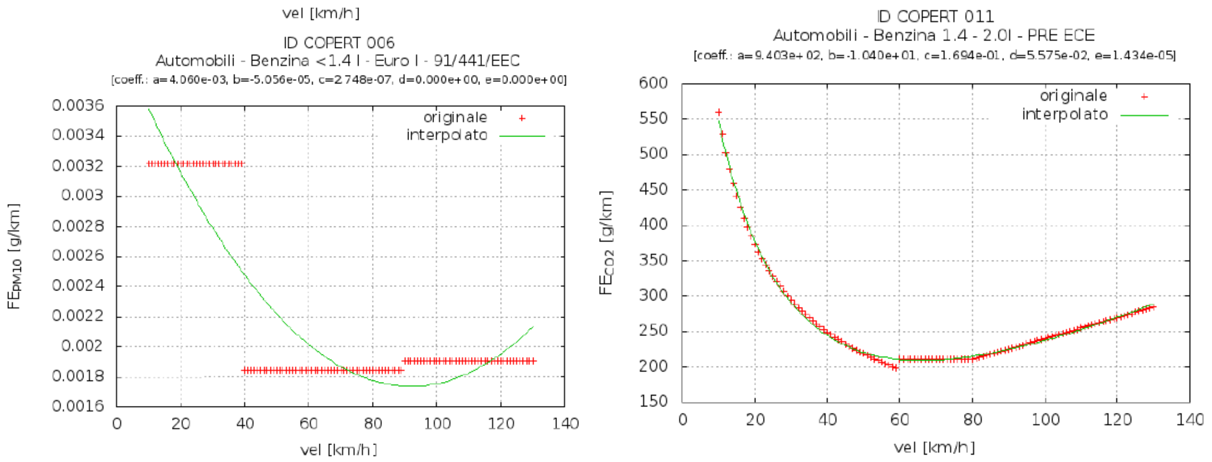

Figure 67: The emission factors approximation proposed in COPERT (Source: CISMA).

The pollutants that are going to be addressed in this work will be mainly focused on main traffic-related pollutants such as **nitrogen oxides** (NO_X), **carbon monoxide** (CO), **fine particulate matter** (PM_{10}), but will also cover greenhouse gas emissions, in particular **carbon dioxide** (CO_2) and being directly bound to it the **fuel consumptions**. Secondary pollutants are also considered, i.e. **volatile organic compounds** (VOC), ammonia (NH_3), **total suspended particulate matter** (TSP), **sulphur dioxide** (SO_2), **methan** (CH_4) and **unbrunt hydrocarbons** (HC).

The result of the emission modeling chain is therefore expected to be in terms of an array of air pollutants concentrations that are specific for a certain road stretch of the road network (more specifically of an arch of the graph representing it). Its visualization through a GIS could look like the map illustrated in Figure 68.

The reference database tables design is moreover reported in Figure 69. One of the main assumptions that must be considered in the model is that the statistical distribution of vehicle classes is mainly uniform at every location and on the long-period, with the possibility to regularly update it once more updated estimations are available.

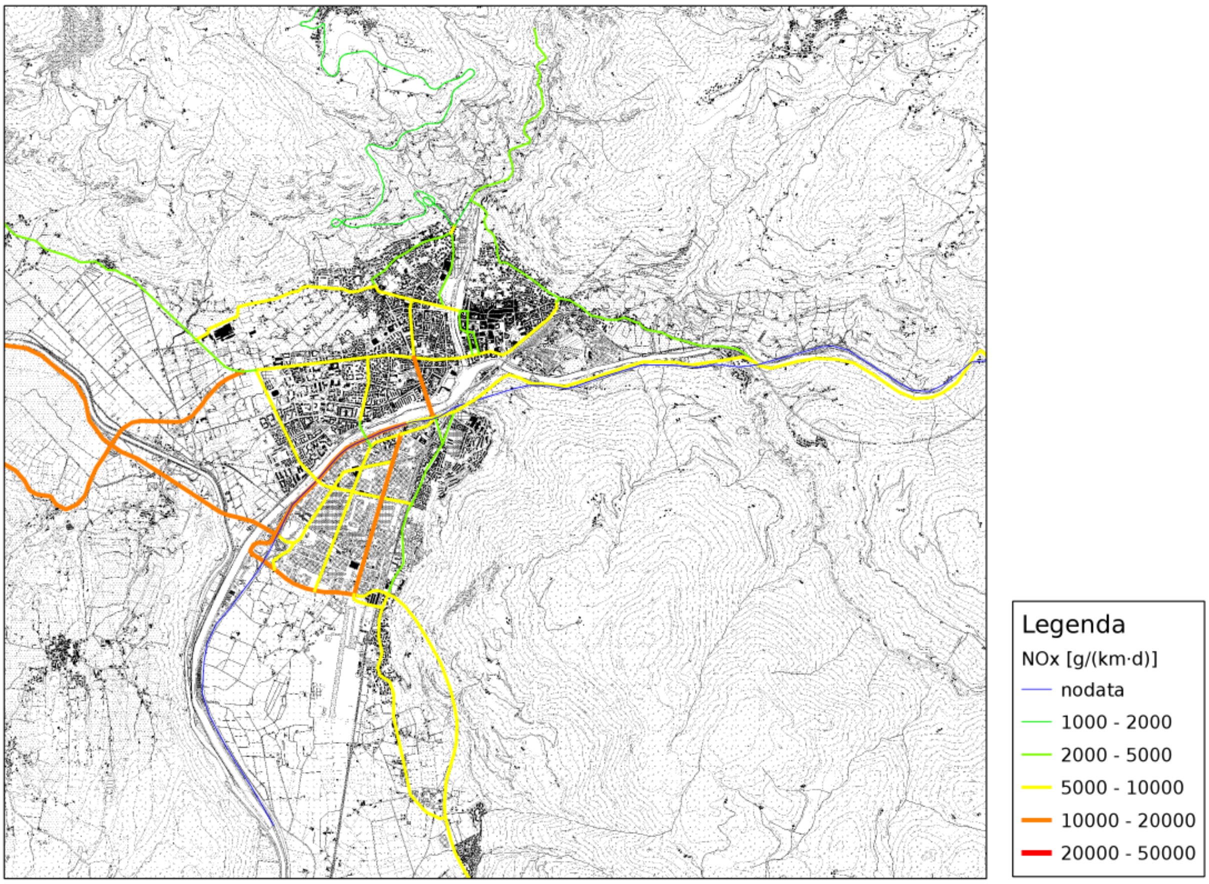

Figure 68: A possible output of the emission modelling chain (Source: CISMA).

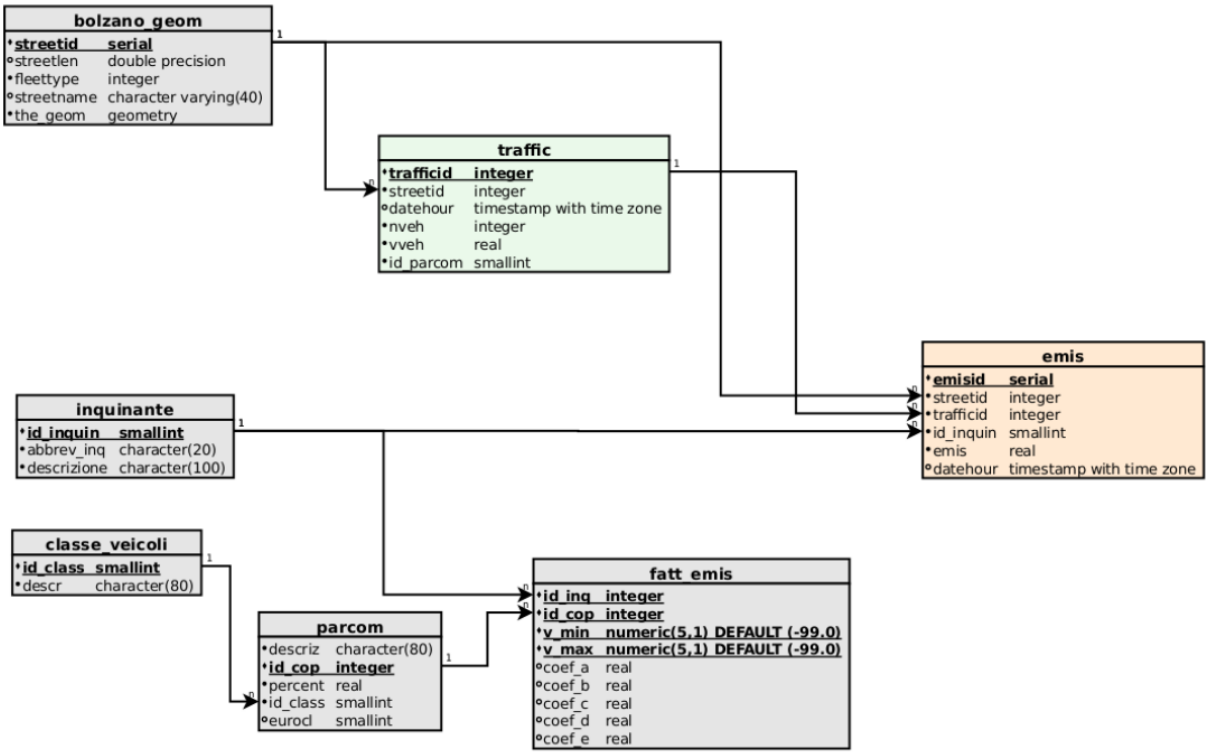

Figure 69: The database design of the emission model (Source: CISMA).

Dispersion model

The dispersion model is going to be a second modelling engine to be run in cascade to the emission model. Even in this case, different methodologies and tools have been analyzed in order to properly accomplish this task in the real-time requirements given in the project. The most known at the state-of-art are:

- **AUSTAL**, probably the state of the art of local scale air quality dispersion model for urban areas, developed and maintained by the German Federal Environmental Agency, but hardly deployable for real-time applications because of its complexity and its very high computational resources demand, which significantly limits its potential exploitation on wider road networks [30];

- **CALINE**, a free and open source dispersion model developed in in California and maintained by the California Department of Transportation. CALINE is less recent but with a simpler theory beyond it, and therefore more suitable to be run on a real-time basis [31]. It can also be adapted to peculiar situations such as emissions in “urban canyons”, over bridges, in parking lots, etc.

Given these premises, the choice has fallen on CALINE, which will produce an output similar to the one which is reported in , which is in this case a raster image with a value of air pollutants concentrations for each spatial resolution element of the target area.

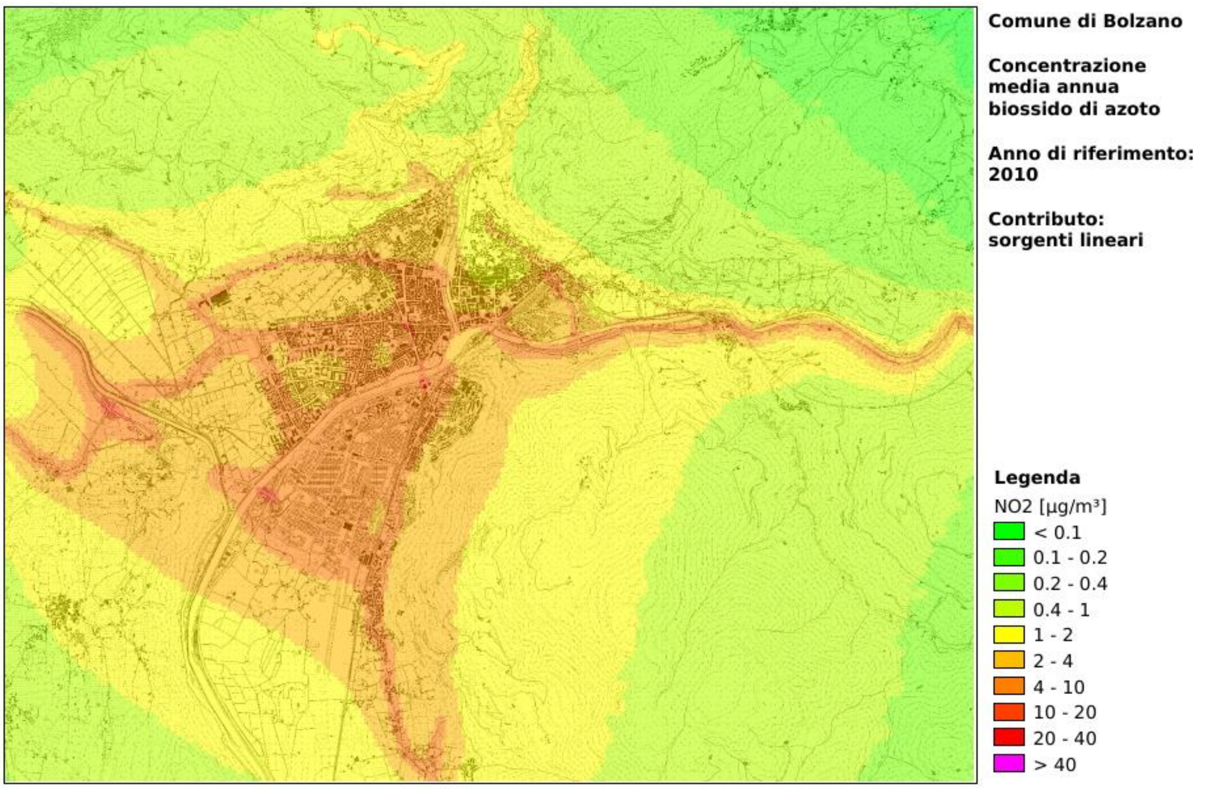

Figure 70: A possible output of the INTEGREEN dispersion model (Source: CISMA).

5.3.2 Traffic elaboration task 1: Bluetooth based travel times estimator

As already discussed, in order to properly run all this environmental elaboration chain, it is therefore fundamental to have at disposal, apart of environmental data, accurate estimations of (at least) two traffic variables, namely:

- the **traffic flows** (measured in number of vehicles in the time unit)
- the **average vehicle speed**.

A distributed estimation over the whole network of these parameters is going to be provided by the traffic simulation model, enhanced in order to be able to run in a semi-online way. Punctual estimations of these parameters are going to be provided by the fixed traffic detectors, which will effectively be used in order to train the traffic model. Moreover, **an estimation of the second parameter is going to be directly provided by properly processing the raw data records collected by the Bluetooth-based travel times detection system**. If validated by empirical measurements, the expectation is that this monitoring system could be able to provide useful information for the estimation of the traffic flows as well. This information is going to be extracted through the processing chain presented in , and well analyzed in one of the technical publications given by the project at international level [32].

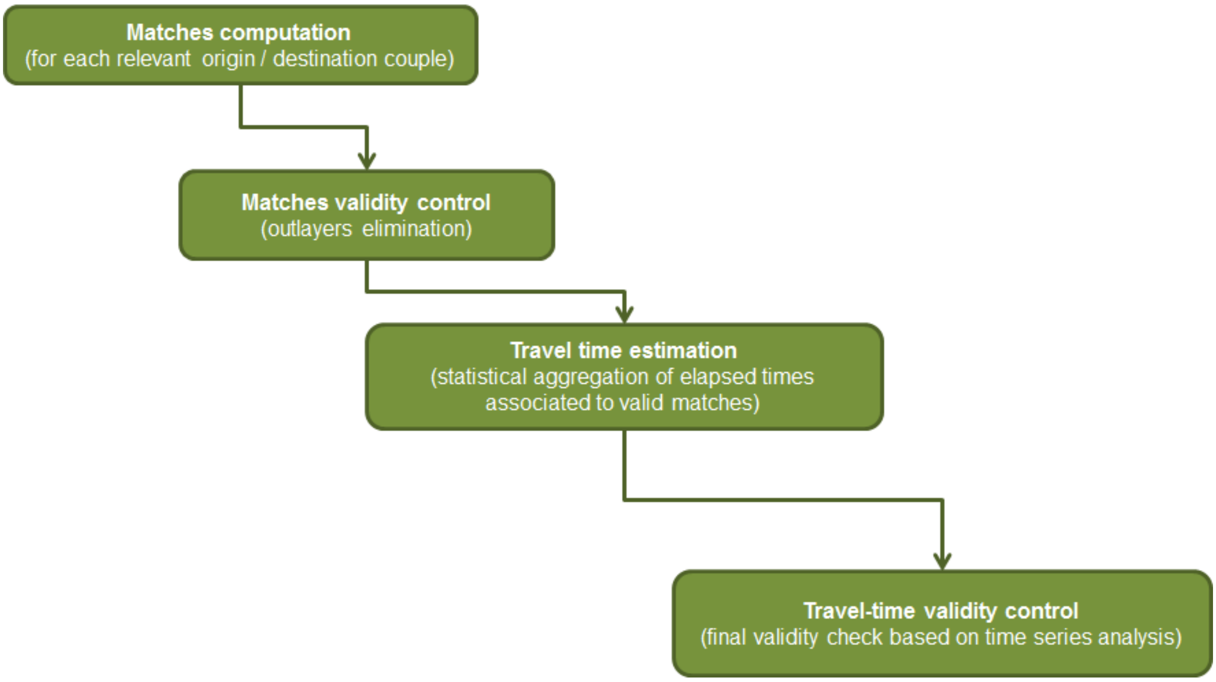

Figure 71: The Bluetooth-based travel times detection system elaboration chain.

Before delving into the details of how the data can be analyzed and validated, it is necessary to introduce the following definitions:

- a "**record**" is a data packet composed by three fields: a station identifier, a unique car identifier and the timestamp of the detection;

- a "**match**" is a couple of two records with the same car identifier but different station identifiers (O - origin, D - destination);
- the "**elapsed travel time**" is defined as the relative difference between the timestamps associated to each valid match.

Given the fact that the monitoring infrastructure is not able to automatically cut out cars that have made one or more stops between the two detection points from those that pass strait, a proper validation and elaboration process is demanded in order to accomplish this task.

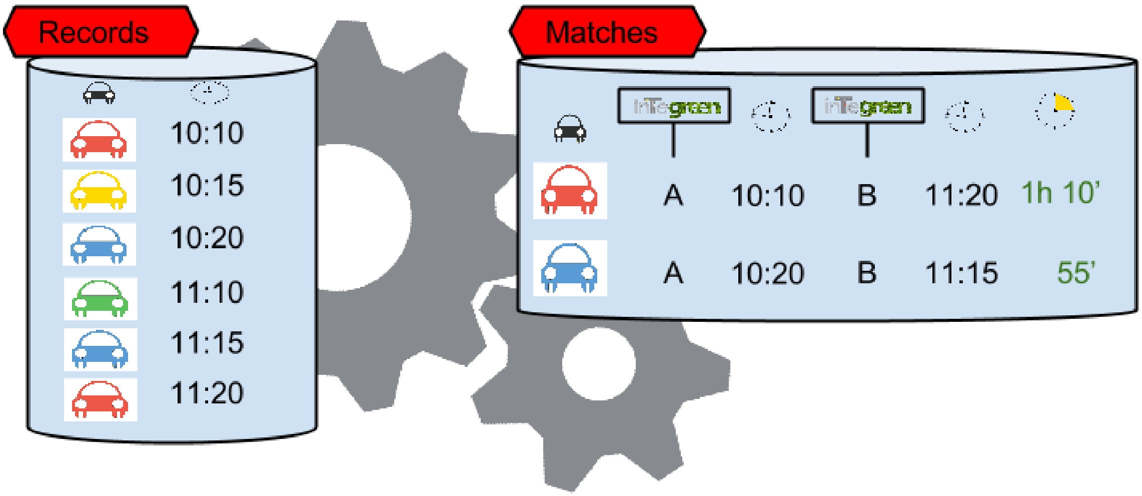

Figure 72: The Bluetooth-based travel times detection system elaboration chain.

Matches computation

The elaboration of a match representing an origin destination requires several data-analysis, as a matter of fact it is not a simple join operation of all the records detected by an origin against those detected by a destination. The right records for both the origin and destination must be chosen carefully not only to understand if a car had a stop between two origin/destination stations but also because a car could have been detected several times by an origin station and only once by the destination. In this latter case, the correct behavior would be to create only one match recognizing the right record detected by the origin station that must be coupled with the one detected by the destination. All the other logs will not be part of any match. This problem can be clearly described by the following example.

Station Code	Car Identifier	Timestamp
1	'a'	12:00
1	'a'	12:30
2	'a'	14:00
2	'a'	14:30

Nr.	Station Code Origin	Station Code Destination	Timestamp Destination	Timestamp Origin	Elapsed Time
1	1	2	12:00	14:00	2:00
1	1	2	12:00	14:30	2:30
2	1	2	12:30	14:00	1:30
2	1	2	12:30	14:30	2:00

Table 35: The Bluetooth-based travel times detection system – routines example.

The matches resulting by the simple left join require to be filtered in order to remove wrong associations. It is worth to mention that all matches marked as valid will be used in further elaboration to computed the average elapsed time, as many invalid matches we remove as much precise the estimation average elapsed time will be. The issue presented describes a very simple and possible behaviour, a car detected two times by a station (the origin probe called '1'), moved in the area of a second detector (the destination probe called '2') where it has been detected other two times.

In the example proposed, among all the 4 matches, we must mark as 'valid' only one, according to this criterion:

"a match is considered "valid" if and only if (i) the record at station O is detected in time before station D, and (ii) in the time interval identified by its two timestamps there is no other record detected by at least one of the two monitoring stations having the same car identifier"

Given this rule, the only match which is considered valid is the one associated to an elapsed time of 1:30.

Travel times estimation

On top of a set of validated matches, an average value regarding the most relevant and representative elapsed time for a given time window between the origin and destination stations is computed. It is worth noting that field monitoring system is intrinsically related to the design choices with which the Bluetooth scanning process is carried out, and therefore the resolution of the system is intrinsically limited by this factor..

An accurate procedure is needed in order to select and filter out those matches that really represent travel times between two stations without any stop for errands or other reasons. Three standard indicators have been taken into account for such task: (i) the "***lower bound***", which returns the lowest elapsed time, (ii) the "***arithmetic mean***", which returns the arithmetic mean of all elapsed times, and (iii) the "**mode**", which is calculated by returning an indicative value of the time interval where the occurrence of all elapsed times is maximum. All three indicators are unfortunately subjected to the unforeseeable character of travel demand.

The lower bound is not fully representative of the whole traffic conditions, since it can report particular car travels which might have taken advantage of a green wave or even worst that might have driven over the speeds limits. On the other hand, the arithmetic mean can be higher than the real value since it might take in consideration “valid indirect matches”, i.e. matches associated to car travels that had had one or more stops between the two detection points. Finally, the mode is at first sight the most robust indicator, but unfortunately diverges when its value tends to be representative of valid indirect matches. This pattern can be quite common when in the path between the two monitoring stations relevant attraction points are located. The solution proposed for the validation step considers the following constraint:

Given two consecutive modes m_t and m_{t+1}, where t and $t+1$ are the median values of the correspondent temporal frames, then a mode is “valid” if and only if:

$$t + m_t < (t+1) + m_{t+1}$$

A full set of modes values is considered “valid” if the above constraint is satisfied all over the entire set of temporal frames. In order to accomplish this, a recursive algorithm is executed, which allows the invalid matches to be identified and finally removed. Moreover, this method leverages us from using predefined range of correctness, threshold and numerical approximations which give to our data model a stronger elasticity to correctly work with the very dynamic traffic patterns the a city can have.

5.3.3 Traffic elaboration task 2: towards semi-online traffic simulation model

A very important added value for the project is represented by the extended traffic modeling capabilities that will be offered by the traffic simulation model that the Municipality of Bolzano already avails. The model is in particular **PTV-VISUM**, a commercial product of the Germany Company PTV, which at present can operate only in offline model and estimate the average traffic flows in the local urban road network on the base of detailed studies and surveys about the origin / destination trip patterns by local travelers. A typical output of this kind of simulators is reported in Figure 73.

How does a traffic simulation model works? The reference elaboration chain is graphically reported in Figure 74. Historical vehicular traffic data values, gathered through old-fashioned approaches such as manual traffic surveys but nowadays more commonly collected through automatic detection systems, are used through proper *measurement propagation techniques* in order to determine the traffic state in points of the routes where data are not available. In an offline model, this data, together with other information determining a reference origin / destination matrix based on a pre-defined zoning of a target area, become a static representation of a certain reference demand model. By combining this information with a reference transport network through proper *route assignment techniques*, one could have an overview of the indicative traffic flows characterizing each road stretch of the network.

Figure 73: Traffic simulation output example.

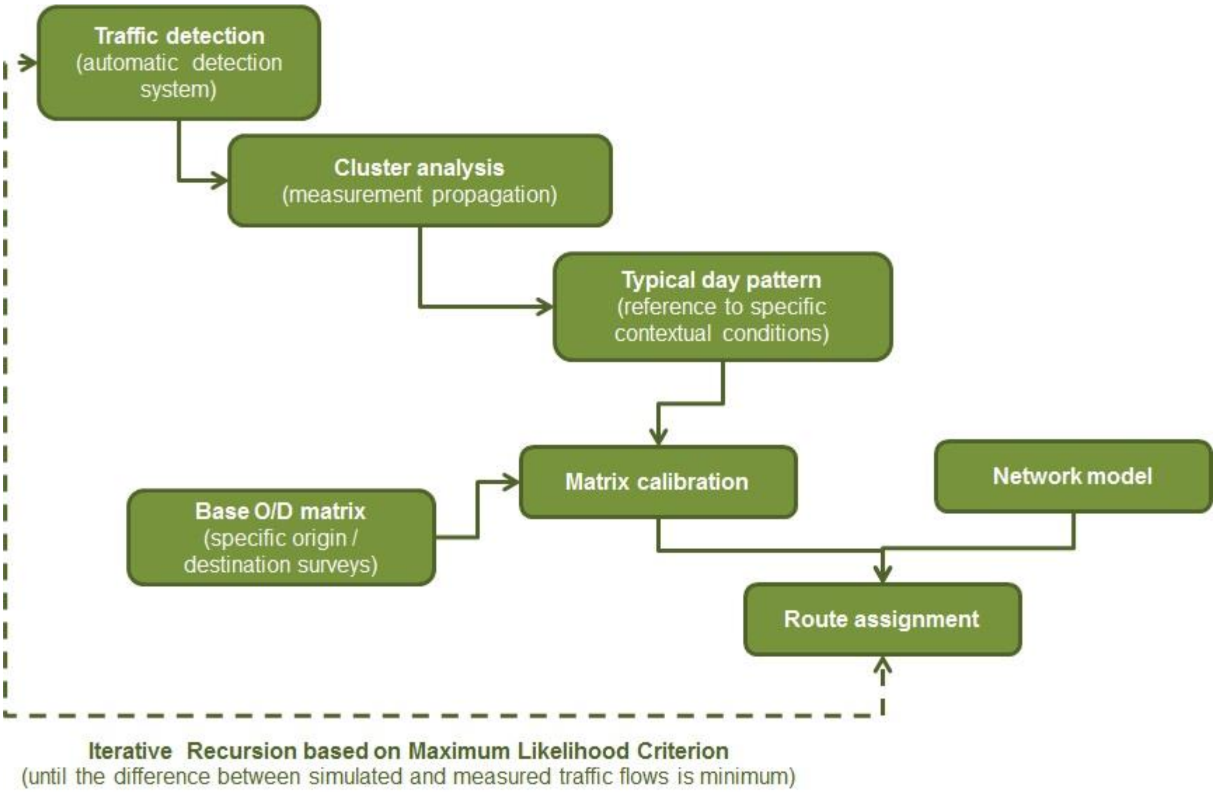

Figure 74: Traffic simulation modelling – reference offline modelling chain.

In INTEGREEN, the idea is to let this offline modelling chain that is actually in use for long-term planning purposes only evolve towards to a “semi-online” chain (Figure 75), which could support the real-time environmental chain described above. More specifically, the plan is to:

- take in consideration the **real-time traffic measurements in order to adapt the traffic flows estimations** generated by the model as a function of the reference demand model inputs, following a maximum likelihood criterion;
- **consider the “matches” calculated on top of the records gathered by the Bluetooth-based travel times detection system** in order to calculate a certain number of reference origin / destination matrixes that can been considered as a function of specific contextual conditions (e.g. working / non-working days, bad weather, etc.).

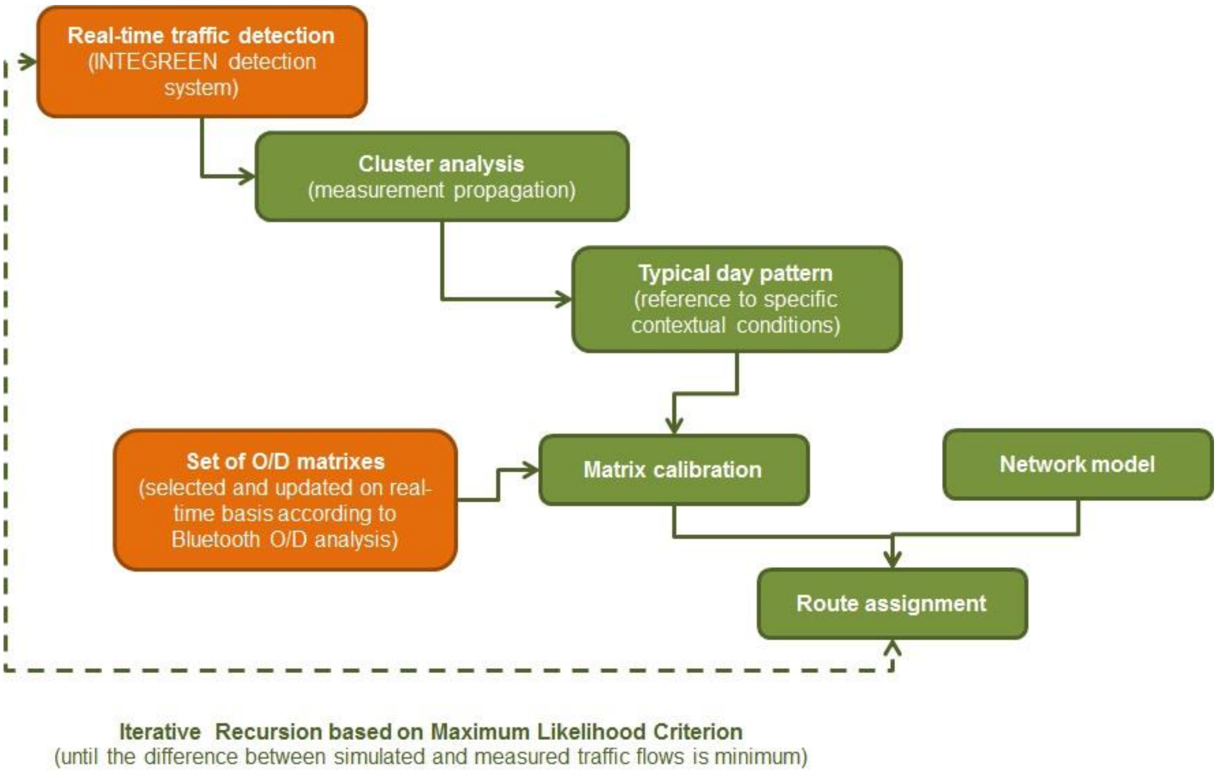

Figure 75: Traffic simulation modelling – proposed “semi-online” modelling chain.

Based on quick studies of reference state-of-art, this is one of the first attempts to let offline traffic simulation models for real-time applications. The idea to continuously update the reference O/D matrixes through the empirical field measurements collected by the Bluetooth monitoring system is also a promising innovation introduced by INTEGREEN.

The implementation of these complex functionalities are going to be assigned to an external sub-contractor, probably TPS, an Italian traffic engineering company which is already supporting the Municipality of Bolzano in the use of the simulator and therefore knows already very well the transportation requirements of the city and also the particular configuration of the tool in use.

5.3.4 XFCD data analyzer

Finally, a separate processing chain is going to be considered as far as the XFCD data provided by the INTEGREEN mobile system is concerned (Figure 76).

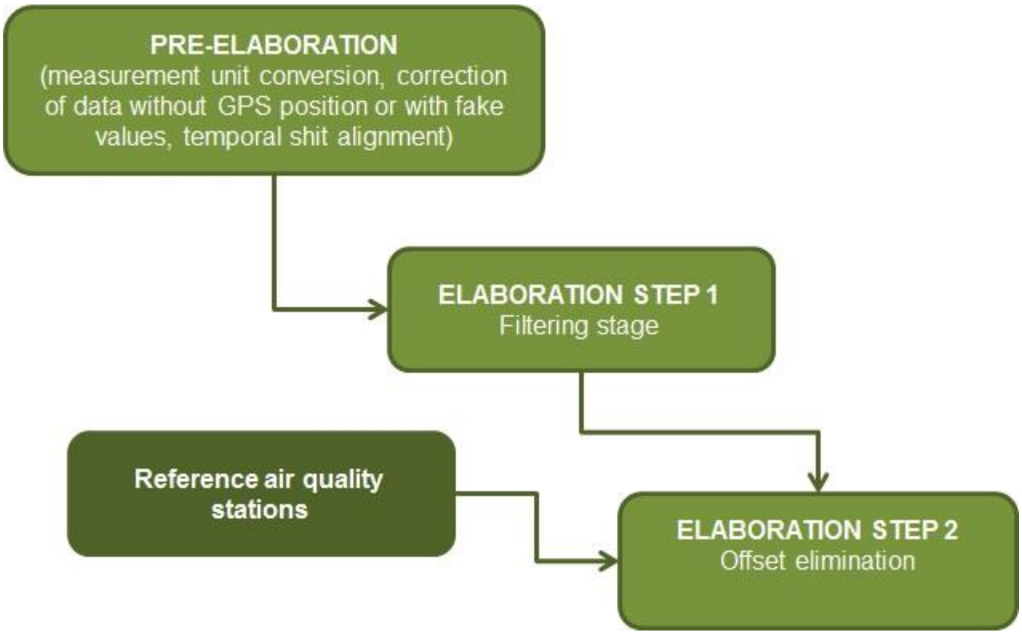

Figure 76: XFCD data analyzer– reference elaboration chain.

After having properly "**cleaned" the data** (e.g. spatial position correction, removal of the temporal shift associated to the delay introduced in the air pollution measurement system), the samples' values will be:

- properly **filtered** with one or more suitable statistical filters able to improve their SNR and minimize the intrinsic limitations of the sensors in terms of response time;
- "**aligned**" as a function of the reference values provided by the fixed air quality stations.

Different filters have been considered for the accomplishment of the first elaboration step; some of them such as the **Kalman filter** are globally more accurate and in the condition to minimize the RMSE, while others (e.g. **negative exponential filters**) can better manage the situations of very sharp air pollution conditions variations, both in space and in time. The idea is to consider both in the reference system architecture: the first ones for post-processing evaluations, and for very detailed comparisons with the output produced by the dispersion model; the second ones in "online" mode, in order to have an immediate assessment of the air pollution conditions in certain streets while the mobile system is driving. In this way, the Environmental Supervisor Centre of INTEGREEN is going to put at disposal of the traffic operators a couple of elaborations for both traffic and air pollution assessment (i.e. one coming from the two modelling chains and one coming from the automatic elaboration of the XFCD data), with the intention to clearly identify the potential added value of considering the mobile system within the proposed system architecture, which is expected to be in terms of qualitative detection of specific air pollution hotspots that are not identifiable through the dispersion model.

5.3.5 *Future offline business intelligence analysis*

Since the information available at the data center layer can be useful not only for real-time mobility and traffic applications, but also for medium- and long-term planning activities, during this design action it came out the opportunity to introduce a business intelligence tool able to permit a smart visualization of the collected data and information. The perspective opened at this stage is to adapt for the particular needs of the project a business intelligence engine developed in another ERDF project coordinated by TIS called GeoBI, which permits a user-friendly and geo-referenced presentation of the available data and information. By giving in input the historical datasets gathered by INTEGREEN, a specialized user can make interesting correlation analysis of the recorded phenomena and get useful inputs on how to improve the traffic control strategies in the medium- and long-term.

Conclusions

This report has presented the main design choices covering the data management unit and the environmental stations front-end. In particular, the following aspects have been presented:

- the results of the **FRAME** design activities, which are more specifically presented in a separate annex;
- the choice of the high-level design concerning the **integration between Traffic Control Centre and proposed Supervisor Centre**;
- the **roadside installation plan**, with **specification of the real-time data exchange** modalities (which include the interfaces defined for the continuous collection of environmental data from existing 3rd parties systems);
- the presentation of the idea to use **Bluetooth technology** as an additional mean to monitor urban traffic conditions, and its inclusion in the system architecture;
- the design choices at the **data center layer**, with specification of the **traffic and environmental elaboration tasks** that are going to be included in the INTEGREEN system and that will be implemented in Action n.4.

Bibliography

[1] INTEGREEN consortium, "D.3.1.2 "Vehicle-to-centre front-end and web interfaces design"," 2014.

[2] INTEGREEN consortium, "D.3.2.1. "On board telematic unit"," 2014.

[3] INTEGREEN consortium, "D.3.2.2 On board traffic and environmental monitoring unit," 2014.

[4] INTEGREEN consortium, "D.2.1.1 "Supervisor Centre components requirements"," 2012.

[5] U.S. Department of Transportation, "Systems Engineering for Intelligent Transportation Systems," 2007.

[6] IEEE, "IEEE Std 610.12-1990 - "IEEE Standard Glossary Of Software Engineering Terminology"," 1990.

[7] P. Jesty and R. Bossom, "Using the FRAME architecture for Planning Integrated Intelligent Transport Systems," in 4th Euro American Conference on Telematics and Information Systems (EATIS), Prague (Czech Republic), 2009.

[8] U.S. Department of Transporation - Research and Innovative Technology Administration, "National ITS Architecture Technical Deployment Support for Public Agencies," Intelligent Transportation Systems - Joint Program Office, 2013. [Online]. Available: http://www.its.dot.gov/arch/.

[9] ISO - TC 204 Intelligent transport systems, "ISO/TR 26999:2012 "Intelligent transport systems - Systems architecture: Use of process-oriented methodology in ITS international standards and other deliverables"," 2012.

[10] ISO - TC 2014 "Intelligent transport systems", "ISO 14813-1: 2007 "Intelligent transport systems - Reference model architecture(s) for the ITS sector, Part 1: ITS service domains, service groups and services"," 2007.

[11] FRAME Forum, "European ITS Framework Architecture - FRAME Selection Tool Reference Manual," 2009.

[12] European Broadcasting Union, "TPEG - what is it all about?," 2007.

[13] EasyWay project, "Data Exchange Interfaces: DATEX II - Deployment Guidelines," 2012.

[14] EasyWay project, "Information and Communication Technologies - Operating Environments (Supporting Guidelines)," 2012.

[15] European Commission, "Directive 2007/2/EC of the European Parliament and of the Council estabilishing an Infrastructure for Spatial Information in the European Community (INSPIRE)," 2007.

[16] CLEAN-ROADS consortium, "D.A2.2 "Technological instruments and constraints"," 2013.

[17] Apache Software Foundation, "Apache XML-RPC," 2013. [Online]. Available:

http://ws.apache.org/xmlrpc/.

[18] E. De Munari, I. Allegrini, N. Bardizza, N. Carfagno, N. Di Carlo, A. Gaeta, G. Lanzani, M. Malaguti, G. Marson, C. Melegari, F. Moricci, P. Pagotto and L. Ramponi, "Linee guida per la predisposizione delle reti di monitoraggio della qualità dell'aria in Italia," Agenzia per la Protezione dell'Ambiente e per i Servizi Tecnici - Centro Tematico Nazionale Atmosfera Clima Emissioni.

[19] European Environment Agency, "Criteria for EUROAIRNET - The EEA Air Quality Monitoring and Information Network," 1999.

[20] E. Cerami, Web Services Essentials - Distributed Applications with XML-RPC, SOAP, UDDI & WSDL, O'Reilly Media, 202.

[21] Co-Cities consortium, "D2.1 Report of cooperative cities services and set use cases," Programma CIP-ICT-PSP , 2011.

[22] Rebellabs, "The Great Java Application Server Debate," 2013.

[23] M. Massie, B. Chun and D. Culler, "The ganglia distributed monitoring system: design, implementation and experience," Parallel Computing, pp. 817-840, 2004.

[24] Open Geospatial Consortium, "OpenGIS Implementation Standard for Geographic information - Simple feature access - Part 1: Common architecture," 2011.

[25] Open Geospatial Consortium, "OpenGIS Implementation Standard for Geographic information - Simple feature access - Part 2: SQL option," 2010.

[26] M. André, M. Keller, A. Sjoedin, M. Gadrat and I. Mc Crae, "The ARTEMIS European Tools for Estimating the Pollutant Emissions from Road Transport and their application in Sweden and France," in Proceedings 18th International Emission Inventories Conference, 2009.

[27] P. De Haan and M. Keller, "Modelling fuel consumption and pollutant emissions based on real-world driving patterns: the HBEFA approach," International Journaly of Environment and Pollution, vol. 22, no. 3, pp. 240-258, 2004.

[28] D. Parrish, "Critical evaluation of US on-road vehicle emission inventories," Atmospheric Environment, vol. 40, no. 13, pp. 2288-2300, 2006.

[29] L. Ntziachristos, D. Gkatzofilias, C. Kouridis and Z. Samaras, "COPERT: A European Road Transport Emission Inventory Model," Information Technologies in Environmental Engineering - Environmental Science and Engineering, pp. 491-504, 2009.

[30] W. B. Faulkner, J. J. Powell, J. M. Lange, B. W. Shaw, R. E. Lacey and C. B. Parnell, "Comparison of Dispersion Models for Ammonia Emissions from a ground-level area source," Transactions of the ASABE, vol. 50, no. 6, pp. 2189-2197, 2007.

[31] T. Kear, D. Niemeier and E. A. Yura, "Using CALINE dispersion to assess vehicular PM2.5 emissions," Atmospheric Environment, vol. 41, no. 38, pp. 8747-8757, 2007.

[32] R. Cavaliere, P. Valleri and R. Kloibhofer, "The INTEGREEN approach for assessment of environmental traffic management policies," in 9th ITS European Congress , Dublin, 2013.

<u>ANNEX 1</u>: Roadside monitoring units technical specification used in the negotiation process (Italian language only)

La presente richiesta riguarda la fornitura di 3-5 stazioni fisse di misura, da installare in ambiente stradale, capaci di monitorare:

- le **condizioni del traffico**;
- i **livelli di inquinamento dell'aria**, secondo i criteri di posizionamento definiti dal Technical Report No.12 "*Criteria for EUROAIRNET – The EEA Air Quality Monitoring and Information Network*" dell'European Environment Agency (EEA).

Rilevamento delle condizioni di traffico

La stazione di misura deve essere in grado di effettuare delle misure relative ai passaggi veicolari, e permettere la raccolta, in forma aggregata su **finestre temporali di 5 minuti**, dei seguenti parametri:

- **numero di veicoli totali**, dettagliati per:
  - <u>direzione di marcia</u>;
  - <u>classe veicolare</u> (in funzione dello standard di classificazione 9+1 "ITALY10");
  - <u>profilo di velocità</u> (secondo intervalli di ampiezza pari a 10 [km/h], con intervallo inferiore 0-15 [km/h] e intervallo superiore i veicoli con velocità superiore ai 50 [km/h]);
  - <u>corsia di marcia</u> (nel caso di installazione su strada con numero di corsie maggiore di due);
- **gap medio**;
- **headway medio**;
- **tempo medio di occupazione del sensore**;
- **lunghezza media dei veicoli**;

Monitoraggio dei livelli di qualità dell'aria

La stazione di misura deve essere in grado di effettuare delle misure relative ai livelli di qualità dell'aria, con riferimento la seguente lista di inquinanti:

- **NO_2;**
- **NO** (vedi nota[2]);
- **CO;**
- **O_3;**
- **$PM_{2.5}$;**
- **C_6H_6** (opzionale);

La stazione di misura deve essere anche in grado di effettuare delle misure relative ai seguenti parametri meteorologici:

- **temperatura dell'aria;**
- **umidità relativa;**
- **velocità e direzione del vento;**
- **radiazione solare globale.**

Si richiede al fornitore di indicare, per ogni differente inquinante, le seguenti specifiche tecniche della strumentazione che verrà proposta per la realizzazione di queste misure:

- **tecnologia di riferimento;**
- **tempo di deriva dei sensori (*zero & span drift*);**
- **sensibilità;**
- **intervallo di misura;**
- **risoluzione;**
- **precisione;**
- **accuratezza: sensitività incrociata rispetto ad altri inquinanti;**
- **tempo di risposta;**
- ***sample flow rate;***
- **tipologia di interfacciamento per lo scambio dati.**

[2] In alternativa, è possibile considerare l'opzione di misurare in maniera diretta i valori di NO_X e calcolare i valori di NO in maniera indiretta come differenza rispetto ai valori di NO_2.

La soluzione tecnica proposta dovrà essere in grado di soddisfare i seguenti vincoli:

Inquinante: NO_2	
Valore minimo rilevabile	≤ 2 [ppb]
Valore massimo rilevabile	≥ 200 [ppb]
Risoluzione	≤ 1 [ppb]
Precisione	≤ 5% fondo scala
Tempo di risposta	≤ 5 [minuti]
Intervallo di acquisizione	≤ 5 [minuti]

Inquinante: NO (o NO_X)	
Valore minimo rilevabile	≤ 2 [ppb]
Valore massimo rilevabile	≥ 500 [ppb] (o ≥ 800 [ppb] nel caso di NO_X)
Risoluzione	≤ 1 [ppb]
Precisione	≤ 5% fondo scala
Tempo di risposta	≤ 5 [minuti]
Intervallo di acquisizione	≤ 5 [minuti]

Inquinante: CO	
Valore minimo rilevabile	≤ 0.1 [ppm]
Valore massimo rilevabile	≥ 10 [ppm]
Risoluzione	≤ 0,05 [ppm]
Precisione	≤ 5% fondo scala
Tempo di risposta	≤ 5 [minuti]
Intervallo di acquisizione	≤ 5 [minuti]

Inquinante: O_3	
Valore minimo rilevabile	≤ 20 [ppb]
Valore massimo rilevabile	≥ 250 [ppb]
Risoluzione	≤ 1 [ppb]
Precisione	≤ 10% fondo scala
Tempo di risposta	≤ 5 [minuti]
Intervallo di acquisizione	≤ 5 [minuti]

Inquinante: $PM_{2.5}$	
Valore minimo rilevabile	≤ 1 [$\mu g/m^3$]
Valore massimo rilevabile	≥ 400 $\mu g/m^3$
Risoluzione	≤ 1 [$\mu g/m^3$]
Precisione	≤ 10% fondo scala
Tempo di risposta	≤ 5 [minuti]
Intervallo di acquisizione	≤ 5 [minuti]

Inquinante: C_6H_6	
Valore minimo rilevabile	≤ 1 [ppb]
Valore massimo rilevabile	≥ 10 [ppb]
Risoluzione	≤ 0,1 [ppb]
Precisione	≤ 5% fondo scala
Tempo di risposta	≤ 5 [minuti]
Intervallo di acquisizione	≤ 5 [minuti]

Parametro meteorologico: temperatura dell'aria	
Valore minimo rilevabile	≤ -30 [°C]

Valore massimo rilevabile	≥ 50 [°C]
Risoluzione	≤ 0,1 [°C]
Precisione	≤ 0,1 [°C] (almeno per l'intervallo di misura -10 [°C]; + 30 [°C])
Tempo di risposta	≤ 5 [minuti]
Intervallo di acquisizione	≤ 5 [minuti]

Parametro meteorologico: umidità relativa	
Valore minimo rilevabile	≤ 1 [%]
Valore massimo rilevabile	≥ 100 [%]
Risoluzione	≤ 0,1 [%]
Precisione	≤ 0,1 [%]
Tempo di risposta	≤ 5 [minuti]
Intervallo di acquisizione	≤ 5 [minuti]

Parametro meteorologico: vento	
Valore minimo rilevabile	≤ 0.25 [m/s]
Valore massimo rilevabile	≥ 50 [m/s]
Risoluzione	≤ 0.1 [m/s] ; ≤ 1 [°]
Precisione	≤ 0.5 [m/s] ; ≤ 5 [°]
Tempo di risposta	≤ 5 [minuti]
Intervallo di acquisizione	≤ 5 [minuti]

Parametro meteorologico: radiazione solare globale	
Valore minimo rilevabile	≤ 1 [W/m²]
Valore massimo rilevabile	≥ 1400 [W/m²]
Risoluzione	≤ 1 W/m²

Precisione	≤ 5 W/m²
Tempo di risposta	≤ 5 [minuti]
Intervallo di acquisizione	≤ 5 [minuti]

La stazione di misura dovrà spedire in remoto le misure effettuate attraverso il canale telematico concordato con un **tempo di aggiornamento pari a 15 minuti**.

Caratteristiche d'insieme

Si richiede al fornitore di fornire inoltre delle specifiche tecniche riguardanti le caratteristiche d'insieme dell'apparato, in funzione delle seguenti voci:

1. **consumo di potenza e tipologia di alimentazione** (sarà considerato un valore aggiunto l'alimentazione da fonti energetiche rinnovabili e sostenibili);
2. **classe di protezione IP;**
3. **condizioni ambientali di funzionamento operativo garantito;**
4. **capacità locale di memoria;**
5. **sistema di gestione dei problemi di riscaldamento delle componenti;**
6. **canali di comunicazione dati supportati;**
7. **peso;**
8. **dimensioni:**

Sarà considerato un valore aggiunto:

9. la proposta di una stazione di monitoraggio compatta e integrata rispetto alle due componenti di misura (traffico / inquinamento dell'aria);
10. l'utilizzo di tecnologie non invasive per il rilevamento delle condizioni del traffico;

Attività del fornitore

Oltre alla fornitura della strumentazione indicata, al fornitore viene richiesto di effettuare le seguenti attività:

11. **installazione delle stazioni** (nei punti concordati col committente), comprensiva di tutte le opere civili necessarie;

12. **calibrazione in loco e verifica di corretto funzionamento;**

13. **collegamento telematico** col software centrale della Centrale della Viabilità Comunale gestita dal Comune di Bolzano;

14. **servizio di manutenzione e calibrazione** fino al 01/03/2015, con richiesta di indicare anche un canone annuo per l'estensione di tale servizio anche dopo tale data.