Cooperative Cities extend and validate mobility services

WP4000

D4.1 - Test report on interfaces, data structures and working service deliveries

Version

1.0

Dissemination level

Public

Co-Cities is a Pilot Type B Project funded by the
European Commission, DG Information Society and Media
in the CIP-ICT-PSP-2010-4 Programme

European Commission
Information Society and Media

Contract Number:

270926

Acronym:

Co-Cities

Title:

Cooperative Cities extend and validate mobility services

Contractual date of delivery:

30th November 2012

Actual date of delivery:

1st February 2013

Main author(s) or editor(s):

Michele Masnata (SOF)

Marco Garré (SOF)

Other author(s):

Said Rahma Rodriguez (ATO)

Roman Pickl (FLU)

Lara Pérez Eguireun (MLC)

Fernando Zubillaga (MLC)

José Manuel Arnesto (Telvent – MLC)

Axel Burkert (PTV)

Kamil Darebný (TMX)

Tomáš Tvrzský (TMX)

Version History:

Version	Date	Main author(s)	Summary of changes
0.1	21/05/2012	Michele Masnata	First Table of Contents
0.2	12/09/2012	Michele Masnata	First draft contents
0.3	17/09/2012	Michele Masnata	Added contents for Tuscany Region and Softeco app
0.4	09/10/2012	Michele Masnata	Corrections and updates
0.5	07/11/2012	Roman Pickl, Fernando Zubillaga	Added description of systems in Bilbao and Vienna
0.6	07/11/2012	Tomáš Tvrzský	Added description of systems in Prague, Reading and for iOS mobile app
0.7	05/12/2012	Michele Masnata, Marco Garré	IR3.1 finalized and available. Based on the validation methodology defined there the following contents and results are added: • Synthesis of the technical validation principles and methodologies • RDSS validation description for Tuscany Region • Synthesis of results from checklists (IR4.1) of Alpha test • Results of alpha test
0.71	09/12/2012	Roman Pickl	RDSS validation description for Vienna added
0.72	13/12/2012	Michele Masnata	Corrections
0.73	17/12/2012	Roman Pickl	Integration of RDSS results in Vienna.
0.8	20/12/2012	Axel Burkert, Michele Masnata, Marco Garré	Added RDSS validation description for Bilbao and Munich. Added description of system implementation in Munich. Added validation results available at that date from the Sites.
0.9	23/01/2013	Said Rahma Rodriguez, Roman Pickl, Axel Burkert	Review and quality check
0.91	29/01/2013	Kamil Darebný	Integration of validation results for Prague and Reading.
0.99	30/01/2013	José Manuel Arnesto	Integration of description of system in Bilbao

Version	Date	Main author(s)	Summary of changes
1.0	01/02/2013	Said Rahma Rodriguez, Roman Pickl, Axel Burkert, Michele Masnata	Corrections, Final check. Release of final version 1.0

List of the Co-Cities Project Partners

Partner no.	Partner name	Partner short name	Country
1	AustriaTech Gesellschaft des Bundes für technologiepolitische Maßnahmen GmbH	ATE	AT
2	Softeco Sismat S.P.A	SOF	IT
3	Telematix Software, a.s.	TMX	CZ
4	Fluidtime Data Services GmbH	FLU	AT
5	Brimatech Services GmbH	BRI	AT
	Left intentionally Blank		
7	The Regional Organiser of Prague Integrated Transport	PID	CZ
8	POLIS-Promotion of Operational Links with Integrated Services	POL	BE
9	Atos Origin Sociedad Anonima Espanola	ATO	ES
10	PTV Planung Transport Verkehr AG.	PTV	DE
11	Asociacion Cluster Del Transporte Y La Logistica De EUSKADI	MLC	ES
12	Regione Toscana	FIR	IT
13	Reading Borough Council	RED	UK
14	MemEx S.R.L.	MEM	IT

Table of Contents

List of Tables

List of Figures

1 Executive Summary

The Co-Cities system and its set of components have been developed in WP4000 within two main groups of activities:

1. Development of the necessary components for running the Commonly Agreed Interface (CAI) locally, along the principles and technical specification established in WP3
2. Development of the mobile services by these Co-Cities partners who, besides being Technical Partners, are also playing the role of Traffic Information Service Providers.

In parallel to that another core part of the Co-Cities project was developed: the Reference Platform, acting as a common tool for a number of technical validation and evaluation operations. The Deliverable D4.2 [8] specifically describes the Reference Platform while the deliverable at hand reports on the activities described in points 1 and 2 above.

Forming the basis for the implementation, the development of the Co-Cities system is the core part of the technical activities of the project together with the definition of the technical specification of the system.

The software components providing the basic functionalities of the Co-Cities Service-Oriented Architecture, as defined in the Technical Specification (Deliverable D3.1 [4]), features a number of key aspects which have been strictly considered in the development:

- Co-Cities defines two types of communication flows: **Information provision** (from local providers to the Traffic Information Service Providers (TISPs) and finally to the end users) and **feedback provision** (from the end users to the local level).
- A number of features to be provided for the communication between TISP and local provider have been defined in the project analysis phase (WP2). This includes the identification of a number of **use cases**, the identification of basic service **domains**, the definition of **user groups**, the definition of the necessary **cooperative features** as well as the definition of requirements to guarantee the compatibility with the **In-Time specification [11]**.
- From the analysis, a complete technical specification for the Co-Cities system has been produced in WP3. The specification enables the development of the software components forming the **Commonly Agreed Interface** (CAI) which enables a seamless access to data from different sites by the TISPs. This includes both services for the provision of information from local content providers to TISPs and services for the provision of feedback data from the end user to the TISP and then to the local sites.
- All software components have to be developed on basis of this specification in order to be **interoperable** as foreseen by the basic Co-Cities principles and objectives.

- A final verification of the correct **integration** of all services along the **Co-Cities value chain** is achieved to assess and verify the interoperability. A definition of the Co-Cities value chain is also provided.

2 Scope of the document

The Deliverable 4.1 specifically reports on the implementation of the local CAIs and the mobile services which were carried out in work package 4000. The activities are specifically carried out in Sub-Work Package 4200 but are also coordinated by a specific Sub-Work Package (SWP 4100) in order to ensure that the conformance to the basic principles of service interoperability is kept. Following In-Time [11], Co-Cities is aiming at developing enhanced and cooperative services over a Common Interface, whose technical specification has been produced in WP3000.

The assessment of this conformance is executed within two different Sub-Work Packages: SWP 4400 and SWP 4500. The first performs a validation of the concepts of every single service and the peak performance levels per single service and per interface implementation in one single Site. In the second, the information services of all service providers are tested together in each city in order to perform a complete test over a standard mix of information services and with the planned mobile devices.

A methodology for the technical validation has been developed by considering the requirements of the test and the features to be assessed. The Technical Validation Plan, necessary to execute the integration test has been developed in WP3 and specifically in Internal Report 3.1 [6]. A resume of the guidelines for technical validation is included for convenience in section 7 of this deliverable.

The detailed outcomes of these test phases are reported in details and respectively in Internal Reports IR4.1 and IR4.2. A comprehensive synthesis of the results of the integration tests is presented in section 8 of this report.

The present deliverable provides detailed information about all service interfaces developed by the Co-Cities test sites, all mobile applications developed by the Co-Cities TISPs and all combinations of mobile functionalities using the respective local service interfaces from the Sites. Only the combinations of Co-Cities compliant services are considered as valid.

The developed service interfaces and TISP functionalities are described in the document in detail together with the details on the technological implementation which is different from city to city and from TISP to TISP as the Co-Cities specification (like In-Time) is technologically “neutral”.

The main technical elements described in the report are:

1. **Mobile applications provided by TISPs**
   The end user applications provide information in the domains of travel and traffic and add proper Co-Cities services (feedback functionalities and travel and traffic information services, enhanced in a cooperative way).

2. **Web Services of the CAI**
   Web services of the CAI provide data via adaptation components and web services with specific WSDL. The adapters establish the connections to local systems and translate from local specific data format into the Co-Cities format.
3. **Feedback Services**
   Feedback services are web services implemented from a well known WSDL. They can be used or "invoked" by TISPs and make feedback data available for local systems.

To fully understand this document a technical background of the Co-Cities concepts is necessary. Although a synthesis of the Co-Cities technological principles is given in the first part of the document, a comprehensive presentation of the technical background is out of the scope of this report.

Project deliverable D3.1 [4] provides all necessary technical contents and references. A more detailed technical presentation, instead, is provided within this document for a key aspect of the information and feedback provision specifically developed in Task 2 of Sub Work Package 4200: the service delivery chain defined over the group of actors who operate in Co-Cities.

2.1 Link to the Co-Cities project structure

The work package WP4000 is structured in five sub work packages (SWP) whose goals are specified as follows in the FPP/DoW (Full Project Proposal/Description of Work):

- SWP 4100 - "Coordination of development and procurement"
- SWP 4200 - "Development of interfaces, data structures and service contents"
- SWP 4300 - "Development of reference platform for services and cities integration levels (A, B, C)"
- SWP 4400 - "Integration tests for single services (alpha test)"
- SWP 4500 - "Integration test for single cities (beta test)"

The sources of requirements for the Co-Cities System development and Integration are:

- Mainly the results from the Co-Cities SWP 3100 - "*Specification of ITS system, interfaces, extensions and modules, reference system*" [4], the SWP 3200 - "Interfaces and mechanisms for end user involvement" [5], the SWP 2100 - "*Service definition and use cases*" [1], the SWP 2200 - "*User group definition and selection*" [2], the SWP 2300 - "*Validation strategy for existing systems, including extensions and reference system test cases*" [3].

But also the CAI architecture, interface definition, service and data model coming from the results of In-Time Project [11].

- Supplementary source Information, work done and results coming from the SWP3300 - *"Validation process from service providers point of view"*, SWP3400 - *"Validation process from cities point of view"*, SWP 4200 - *"Development of interfaces, data structures and service contents"* and SWP 4300 - *"Development of reference platform for services and cities integration levels (A, B, C)"*.
- The FPP/DoW (Full Project Proposal/ Description of Work) as a reference to be fulfilled.
- The extensive experience of the Co-Cities consortium partners in Traffic and Travel services and management systems, and communication infrastructure.
- The capacities of the Co-Cities consortium, to setting up, to planning and performing integration activities in the context of the Co-Cities Project.

2.2 Link to the Co-Cities deliverables

The following Deliverables are linked with Deliverable D4.1.

Document	Name and nature of dependency
D2.1	*"Report of cooperative cities services and set use cases" ("Service definition and use cases")* the nature of the dependency is related to the definition and description of the use cases or scenarios and the corresponding services domains and feedback services, as well as the feedback dataset derived from use cases definition.
D2.2	*"List of user groups and interaction process" ("User groups and interaction process")* the nature of the dependency is related to the definition and description of the stakeholders, user groups and segment per services, user involvement process, as well as the definition of user interaction channels.
D2.3	*"Validation strategy for existing systems, including extensions and reference system test cases"* the nature of the dependency is related to the definition and description of the testing and validation strategy overview, as well as the general approach and generic requirements for validation activities.
D3.1	*"ITS system specification description and reference platform for validation"* the nature of the dependency is related to the detailed description of data and service model of Co-Cities CAI (Commonly Agreed Interface), system boundary and general agreements used during the modeling process, specification of the feedback services, as well as a description and specification of the reference platform for validation
D3.2	*"Interfaces and mechanisms for end user involvement"* the nature of the dependency is related to the definitions provided for the end user systems and services which are the basis for the development of TISPs mobile apps.

Document	Name and nature of dependency
IR3.1 (SWP3300/SWP3400)	*"Validated technical interfaces, reference platform and service packages from the service providers"* the nature of the dependency is related to the detailed validation planning activities.
IR4.1	*"Cooperative Services test report"* the nature of the dependency is related to the results of the technical validation (alpha test)
IR4.2	*"Service interfaces and installations test report"* the nature of the dependency is related to the results of the technical validation (alpha test) and the detailed description of the system installation and the developed service interfaces
D4.2	*"Reference platform, it´s configuration and usage manual"* the nature of the dependency is related to the detailed description of the Reference Platform, its usage and configuration

Table 1: Document links

2.3 Intended audience

The main interested groups are the participating cities of Co-cities and the responsible partner for the setup and preparation of the demonstration phase, the TISP´s and mobile application providers and the scientific community for the assessment and evaluation aspects of user behavior of mobile services.

Partners involved in the integration tasks, system integrator, people who will test, validate and evaluate the Co-Cities pilots and system related with, and all users and stakeholders that will participate in the implementation and the execution of the cities pilots and associated scenarios.

On the technical level system architects, information systems designers, system developers and applications, software engineers, and other audiences when designing services and applications taking into account relevant standards and recommendations of standards bodies like IETF, ITU, ISO, W3C, ...

Entities responsible for mobility in cities and regions interested in the extension of their mobility related information services in an open and competitive way for TISPs and mobile application providers.

2.4 General remarks

a) This document follows the ISO/IEC Directives, Part 2: Rules for the structure and drafting of International Standards w.r.t. the usage of the word “shall”. The word “shall” (not “must”) is the verb form used to indicate a requirement to be strictly followed to conform to this specification.

b) In whole of this document, the definition of the terminology “reference platform for validation” or shortly “reference platform” is stated as:

   A reference or core system, which provide the components, functionalities and applications for:

   - Collecting and monitoring the relevant data, characteristics or attributes related to the performance and quality of services and data.
   - Validating the elements, services and data related to the service domains and associated scenarios or group of use cases.
   - Evaluating and reporting the results of monitored and validated services and data.

   In this context, the reference platform should be understood as a “Collecting, Monitoring, Validation and Evaluation System Reference”, as it is defined in this general remark section.

3 Introduction

The Co-Cities architecture can be seen as an evolution of the technical infrastructure developed during the European eMOTION [12] and In-Time [11] Projects. Both, the In-Time and Co-Cities infrastructures, focus on Multi-modal Real-time Travel & Traffic Information services for improved:

- Reliability (up to date information about delays, etc…)
- Comfort (short transit, reduced waiting time, etc…)
- impact on travel behaviour: co-modality
- pan-European multimodal Real-Time Travel Information

The main objective of the Co-Cities project is to facilitate the **Cooperation** between the different actors, in the provision of enhanced, value-added, multi-modal RTTI services to travellers.

In a basic view the actors involved in the Co-Cities infrastructure are:

a) **Local content/service providers** (e.g. Local Authorities, Transport Service Operators) operating typically at regional level and providing base services. These can be independent ones or grouped together to form a **Regional Data and Service Server (RDSS)**
b) Independent **Travel/Traffic Information Service Providers** (TISPs) offering Value-added services in typical application scenarios where both pre-trip and on-trip information can be obtained and combined by the end users for their needs.
c) The end users who consume the provided services and provide feedback via the TISPS' applications.

For the achievement of the cooperative behaviour, the Co-Cities operates in a Service Oriented Architecture (SOA) whose core element is a **Common Interface (CAI: Commonly Agreed Interface)** ensuring interoperability between local providers and TISPs with respect to data and services available in different sites and related to different travel and transport domains and systems.

The CAI offers a unique, well known point of access of heterogeneous content and services but also defines how **Feedback Information** can be exchanged between TISPs and Local Sites to provide the Cooperative elements and behaviour enabling the enhancement of the information provision itself. By means of mobile services the end user first obtains the information and then provides feedback using a number of "feedback functionalities". These, using the CAI, can send feedback data back to the local systems. Locally all feedbacks can be processed and used to improve the associated information and this, eventually, can be re-introduced into the loop.

Any organisations that is able to provide information or services in the domain of traffic and transport for a specific region (such as Traffic Control Centres, Public Transport operators, etc.)

can become a ***Co-Cities follower*** as long as the Co-Cities CAI is used for such provision of data and services as well as for feedback data exchange.

Traffic Information Service Providers, on the other hand, can decide to become Co-Cities followers by building fixed/mobile application using the available Co-Cities services/data.

The CAI is built "on top" of local systems/services: there is no need of internal modifications of existing systems. The system infrastructure is modular and scalable: components for the management of specific content/services (e.g. traffic information, parking information, etc…) can be built in different phases. Further contents/services can be added progressively, when available/needed.

The specification is based on open standards and generic enabling technologies. The implementation process is platform-independent to ensure openness and flexibility.

3.1 The Commonly Agreed Interface - CAI

The Co-Cities Commonly Agreed Interface (CAI) or simply "Common Interface" as it is sometimes called in the following of the document, operates as the main element to flexibly inter-connect local systems and TISPs by providing a set of Data Services in different domains like Static Road Traffic Information, Dynamic Road Traffic Information, etc. and other services for e.g. Routing, Map exchange etc.

The components of the Interface can be physically set up locally (in the same site or infrastructure where the original content/services reside) or as a physical gateway hosted in a remote server and connected to the local provider. In both cases, the CAI enables the access to data and services for the TISPs in a well-known way regardless of the specific technical (platform-specific) implementation.

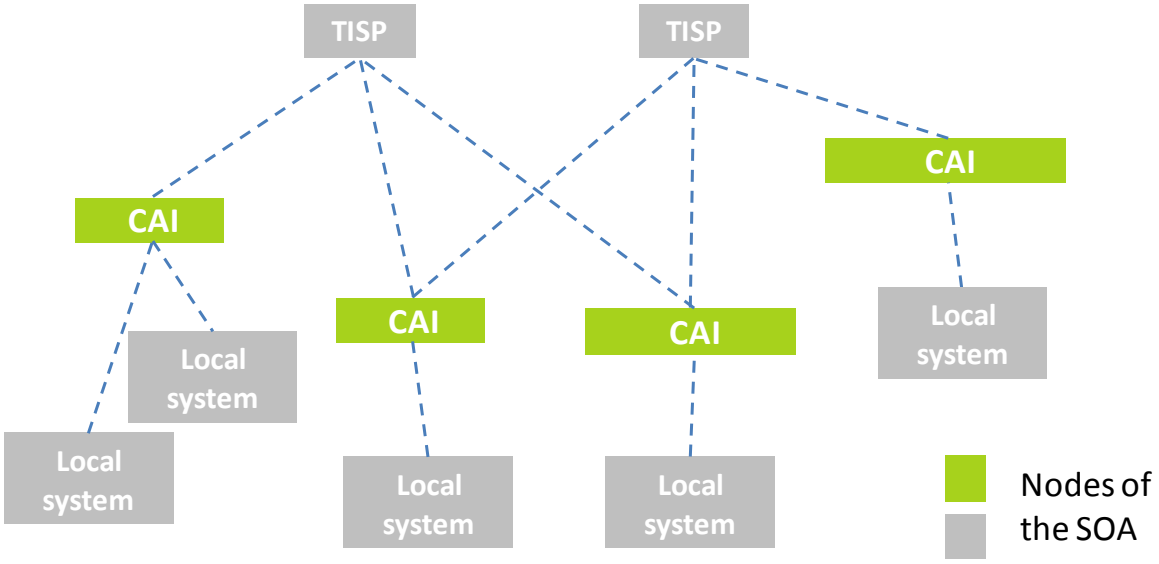

Figure 1 – example of distribution of nodes in the SOA

This **interoperability** between the nodes of the SOA is achieved by translating the different (local) offerings from traffic and transport operators in a common format by means of software components called **Adapters.** Once transformed, data is provided via web services. These components are running for local service/data provision. Local Sites can be considered as the "producers" of the CAI services while the TISPs can be considered as the consumers. A reverse process of transformation, in fact, can be ideally seen for the TISPs who read data and eventually use it for the provision of services by means of their custom technology and interfaces.

The provision of user feedback relies on the very same interoperability principles and constraints. Like for the aspects of information provision, the Co-Cities technical specification defines the transformation associated to feedback data with all details about the target data types and the necessary service interfaces.

The main pillars of the Co-Cities technical specification defining the interface are:

- The **data model**, which defines how the common information is structured (e.g. it defines the data structure to describe a traffic message, a parking place etc.)
- The **service model** which defines the services through which the information is exchanged between local sites and TISPs.

The specification, usable for the practical implementation of the services, includes:

- A formal definition of the **format** for **data** and **information exchange** between the

interface (CAI) and TISP applications/services. This includes an XML *Application Schema* defined in GML *(Geographic Markup Language).*

- A formal definition – in *WSDL (Web Services Description Language)* – of the (web) **services** enabling access to data and information available using the data model.

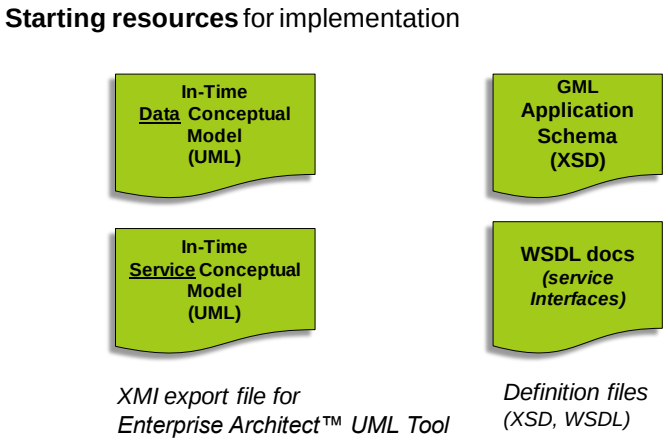

Figure 2 – resources for implementation

The technical specification has been defined as the necessary element for the development of the Co-Cities System in terms of:

- Development of the necessary components for running the Commonly Agreed Interface locally.
- Development of the mobile services by the Traffic Information Service Providers.
- Development of the Reference Platform as a common tool for a number of technical validation and evaluation operations. Deliverable D4.2 [8] specifically describes this component.

For the implementation it has to be noted that while the specification of the CAI is strict in terms of common data and service format, the development of the necessary software components is neutral with respect to the technology used. For this reason, in the present report different technological choices are presented as solutions specifically tailored for the local infrastructure or technological choices operated independently by the partners. These choices are fully acceptable as long as the service interface complies with the Co-Cities specification.

The Co-Cities technical specification is described in details in Project Deliverable D3.1 [4].

3.1.1 Technical elements of the CAI

The first step in the process of defining the Co-Cities Common Interface, which has also influenced the initial phases of the implementation process is the classification of services:

1. **End User Services** are services providing traffic and mobility information to the end users, typically via mobile devices. These are defined for a number of **information domains**.
2. '**Feedback functionalities'** are defined for each of the information domains of the end user services.
3. End user services and feedback functionalities belong to the **B2C** side of the service definition. One or more **B2B** services are defined as building blocks of the B2C elements. All B2B components form the Co-Cities CAI.

Regarding the specific **domains** defined for services and feedback functionalities, a rationalization has been achieved from the existing classification formulated for In-Time. The identified Co-Cities service domains are:

a) Road Traffic

b) Parking

c) Public Transport

d) Multimodal Journey Planning

e) Point Of Interest (for more general feedbacks)

In these domains, B2B and B2C services for the provision of information have been defined starting from the In-Time technical specification. Based on the use cases defined in the project's analysis phase the **Feedback services** have been deducted for the achievement of the feedback functionalities and operations emerging from the Use Cases.

Specifically, three types of B2B feedback services have been defined:

- One "feedback service" is used to express **the Overall Quality of the end user service** provided by the TISP
- Several "feedback services" (one for each Service Domain) are used to express the **Quality of existing data** service
- Several "feedback services" " (one for each Service Domain) are used to **submit new data**

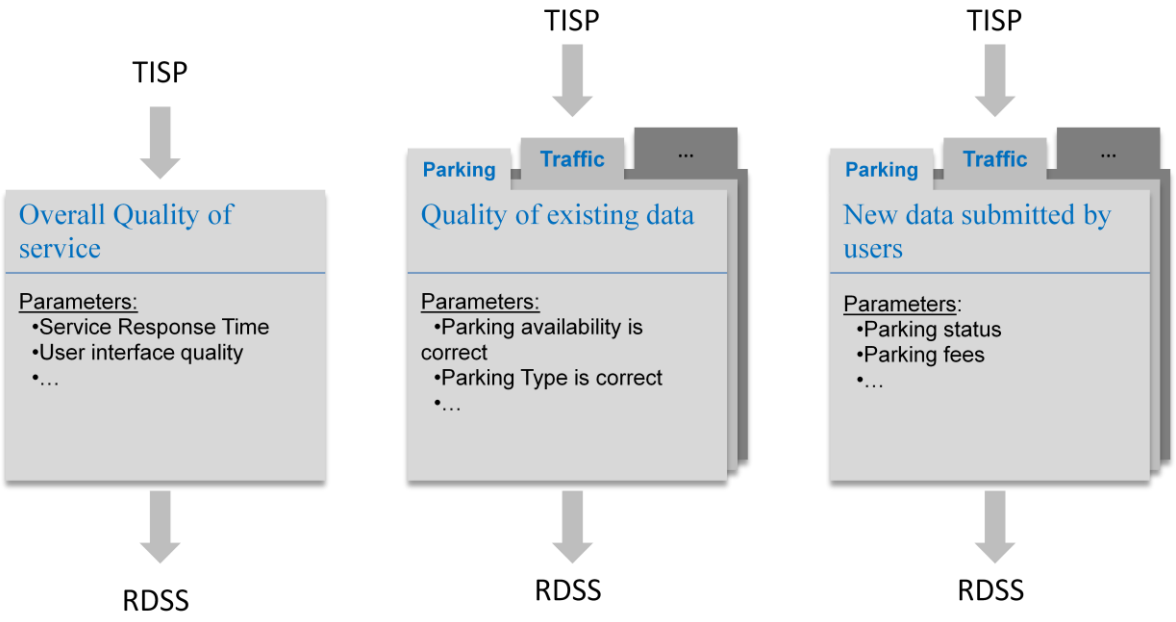

Figure 3 - Types of 'feedback services'

The feedback service are defined as a web service "running" (having a URL) on RDSS side. A TISP is able to access it by a PUSH mechanism. The list of feedback services defined for Co-Cities is in the following table.

Note that there is a single instance of the "Overall Quality of Service" feedback service. A technique to operate a differentiation to associate the feedback service to a specific domain is by associating a specific requestID (parameter of the request) to a specific domain.

Besides the different types of services defined for each service domain there is a *pushRoadQualityFeedback(RoadQualityFeedback)* service which is specifically used to provide a feedback on the quality of data for the Map-Based information which characterizes some Use Case scenarios.

N	Feedback Service	Main service method
1	Overall Quality of Service	*pushQualityFeedback(ServiceFeedback)*
2	Quality of data for Parking Information	*pushQualityFeedback(ParkingQualityFeedback)*
3	Quality of data for Traffic Information	*pushQualityFeedback(TrafficQualityFeedback)*

N	Feedback Service	Main service method
4	Quality of data for Public Transport Information	*pushQualityFeedback(PublicTransportQualityFeedback)*
5	Quality of data for Point of Interest Information	*pushQualityFeedback(PoiQualityFeedback)*
6	Quality data for Map based information	*pushRoadQualityFeedback(RoadQualityFeedback)*
7	Quality of data for Journey Planning	*pushQualityFeedback(JourneyPlanningQualityFeedback)*
8	New data for Parking Information	*pushDataFeedback(ParkingFeedback)*
9	New data for Traffic Information	*pushDataFeedback(TrafficFeedback)*
10	New data for Public Transport Information	*pushDataFeedback(PublicTransportFeedback)*
11	New data for Point of Interest Information	*pushDataFeedback(PoiFeedback)*
12	New data for Journey Planning	*pushDataFeedback(JourneyPlanningFeedback)*

Table 2 – Feedback Services in Co-Cities

3.2 Co-Cities mobile apps

Co-Cities mobile apps (applications) are designed and developed on top of the Co-Cities Common Interface to offer two basic types of functionalities to the end users:

- Provision of information in the travel and traffic domains defined for Co-Cities. The information is generated from the different Co-Cities Sites and provided by them over the CAI.
- Possibility to provide feedback information, from the end user to the local site, over the existing information in the travel and traffic domains defined for Co-Cities and with three feedback types as defined in 3.1.1.

Key feature of each TISP app is the full compliance with the Co-Cities specification which is the condition to develop only one single app and to deploy it as a multi-site solution for all Co-Cities-compliant sites with no further adaptation or change necessary except for referencing a different set of service URLs.

B2B data can be read, used and organized by the TISP regardless of the specific site from which data comes from. Data can be used to:

- Organize the information and build a user interface for its provision (e.g. map-based, list-based etc.)
- Build additional B2C services on top of it (e.g. Journey planning on top of road data, use location based services for parking, traffic, public transport etc. in combination with the Journey Planning, etc.)

In general, TISPs are free to deploy mobile solutions with custom interfaces and functionalities thus looking completely different although using the same B2B interface for data and feedback exchange.

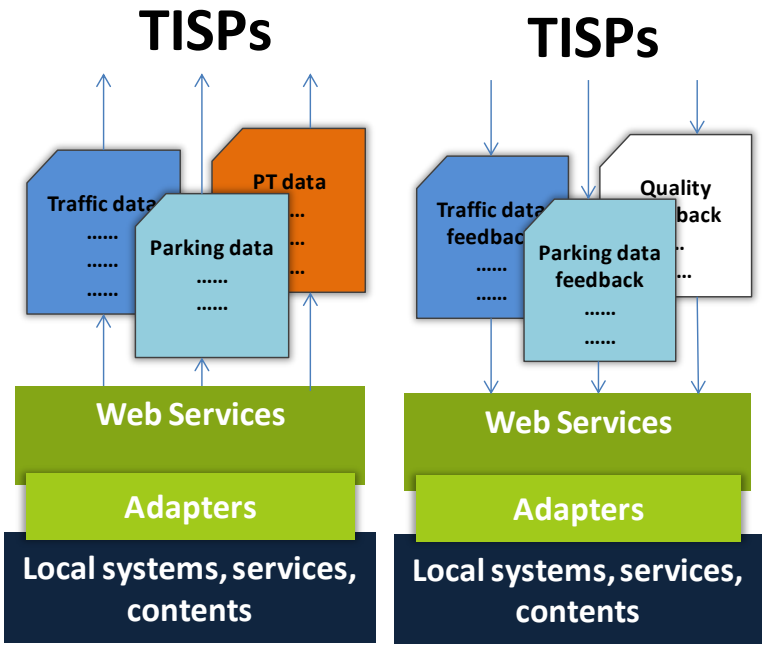

Figure 4 - Interaction of mobile apps with the Co-Cities interface

3.3 Development of the Co-Cities system

A number of software components are developed to build up the Co-Cities system from the technical specification designed in Work Package 3000:

- The components of the Commonly Agreed Interface (CAI): Adapters and web services
- The mobile services (apps) developed by the Traffic Information Service Providers.
- The Reference Platform as the common tool for a number of technical validation and evaluation operations.

For the development of all necessary components of the Co-Cities architecture, the approach can be neutral with respect to the specific technology used as long as the indications of the technical specification are fully respected. Different technological choices have then been adopted and are presented in this report for the development of services and their integration both in the specific local infrastructure and as a full working node of the Co-Cities SOA.

The following key aspects have to be strictly considered in the development:

- Co-Cities defines two types of communication flows: **Information provision** (from local providers to the Traffic Information Service Providers –TISPs- and finally to the end users) and **feedback provision** (from the end users to the local level).
- A number of features to be provided for the communication between TISP and local provider have been defined in the project analysis phase (WP2). This includes the identification of a number of **use cases**, the identification of basic service **domains**, the definition of **user groups**, the definition of the necessary **cooperative features** and the requirements for the compatibility with the **In-Time specification**.
- From the analysis, a complete technical specification for the Co-Cities system has been produced in WP3. The specification enables the development of the software components forming the **Commonly Agreed Interface (CAI),** which enables a seamless access to data from different sites by the TISPs. This includes both services for the provision of information from local content providers to TISPs and services for the provision of feedback data from the end user to the TISP and then to the local sites.
- All software components have to be developed on top of this specification in order to be **interoperable** as foreseen by the basic Co-Cities principles and objectives.
- A final verification of the correct **integration** of all services along the **Co-Cities value chain** is achieved to assess such interoperability.

3.3.1 Coordination of the development activities

The development, integration and test of the Co-Cities system is the most significant and demanding technical task achieved in Year 2 of the project. These activities have been necessarily done in a distributed way:

- All six pilot sites had to set up the Co-Cities CAI and connect it to the local systems (Service/Content Providers).
- The TISPs had to set up the Co-Cities mobile app.

Every component and element of the Co-Cities chain, developed by different partners, had to be fully interoperable with the remaining part of the system in order to ensure that the Co-Cities system could work, as a whole, along the specifications.

For these reasons, the three aspects (development, integration and test), have been strictly coordinated to facilitate the common understanding and achievement of the software development, starting from the technical specification produced in WP3000 with a specific monitoring process performed over the entire duration of the development.

This process aims at ensuring that the whole system has all necessary interoperability features which make the locally developed interfaces and the mobile applications compliant each other and fully conform to the technical specification.

The local sites -represented as Regional Data and Service Servers (RDSSs)- develop the CAI and the TISPs develop the mobile applications based on the technical specification. Once the respective parts are complete a verification of their interoperability (e.g. the possibility for a mobile app to exchange data with a site) is performed. If the verification has no positive outcome the necessary corrective actions are immediately taken by the responsible partner and the verification is repeated again until the outcome is positive.

This process can be summarized with the following diagram.

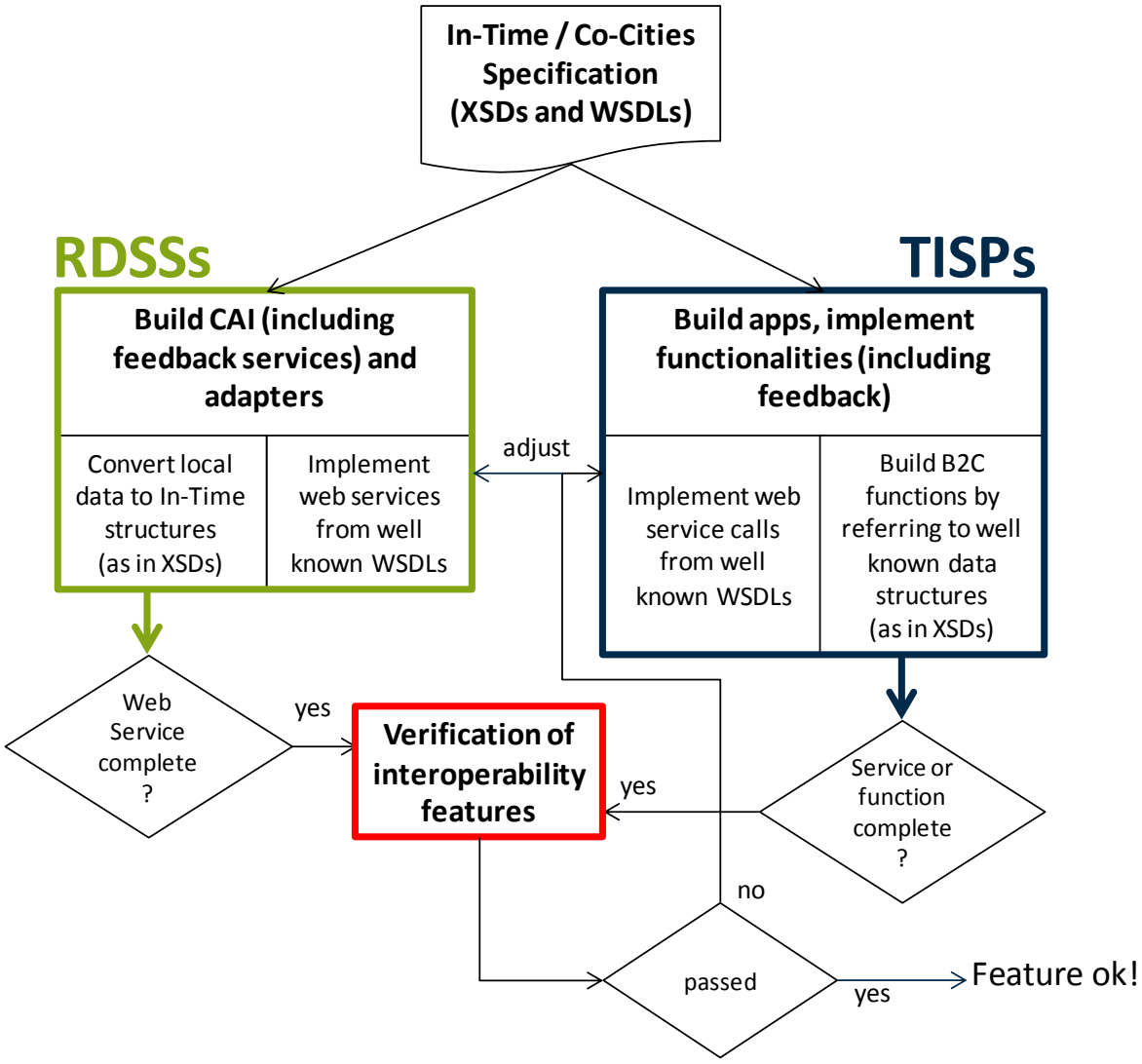

Figure 5 – technical monitoring process for the development activities

4 The Co-Cities Service delivery chain

While the Project Deliverable D3.1 defines the Co-Cities technical specification with all details about the target data types and the necessary service interfaces, a more detailed definition of the Co-Cities Service Chain has been formulated in Work Package 4000 in order to specify the communication flow for both, information provision and feedback provision. The service chain highlights the technical features which enable the cooperative behaviour of the Co-Cities services as well as the necessary elements to be considered for the integration of services into the Co-Cities SOA.

The definition of the service delivery chain with the process steps for the single service and their dependencies between each other has been carried out in Sub Work Package 4200.

This section explains the different actors involved in the information exchange, as well as the chain structure and the possible scenarios which can be applied.

The actors involved within a communication are the end user, the TISP, the Co-Cities CAI and the local systems (RDSS), who typically plays the role of accessing local systems via adaptation components.

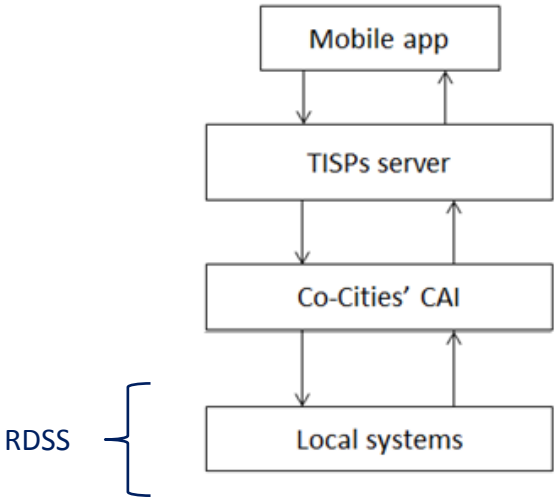

Figure 6 - service chain basic architecture

- *End user (mobile app):* The end user is the citizen that uses the services provided by Co-Cities. It can act both as an old-fashioned consumer, only receiving data from the Co-Cities services, and as provider, receiving and creating data and feedback related to the services.
- *TISP (Traffic Information Service Providers):* TISPs contribute their relations to the end users, customers and the capacity to extend the service concepts and user acceptance.

- *CAI (Commonly Agreed Interface):* standardized access layer to local servers enabling a uniform access to real-time multimodal traffic data and services typically available in different formats from local providers. The CAI facilitates the cooperation between local content/service providers and TISPs in the provision of value-added local services to travellers including the provision of feedback data.
- *RDSS (Regional Data/Service Servers):* provides a centralized access point to data and services on local/regional level and manage the feedback.

4.1 Service delivery chain behaviour

The service delivery chain admits two different behaviours:

1. Service provision to the end user
2. Feedback provision

4.1.1 Service provision to the end user

This is the typical behaviour of a client-server application. The end user demands some information related to one of the services provided by the city, using the TISP's mobile app. The request is analysed and processed within the service delivery chain and a response arrives on side of the end user via the mobile app, as explained in Figure 7.

Using the mobile app the end user sends a message to the correspondent TISP's web service requesting specific information sets.

The TISP can apply caching techniques to improve the performance. In this case it analyses the request and check if it can be processed without forwarding it to the RDSS via the CAI. If this is possible, the response is sent back to the end user. Otherwise the request is formatted and forwarded to the appropriate CAI's web service.

The CAI analyses the message and adapts it before sending it to the local systems (RDSS). Once the message arrives on side of the RDSS the requested data are retrieved from the local servers and sent to the end user´s mobile app through the CAI and the TISP systems.

It has to be noted that the data exchange between the mobile app and the TISP server typically uses a proprietary data format.

The information exchange is explained in more detail in Figure 8 and further explanations can be found in the "Real scenario example" section, easing the comprehension of the behaviour of this functionality within the service delivery chain.

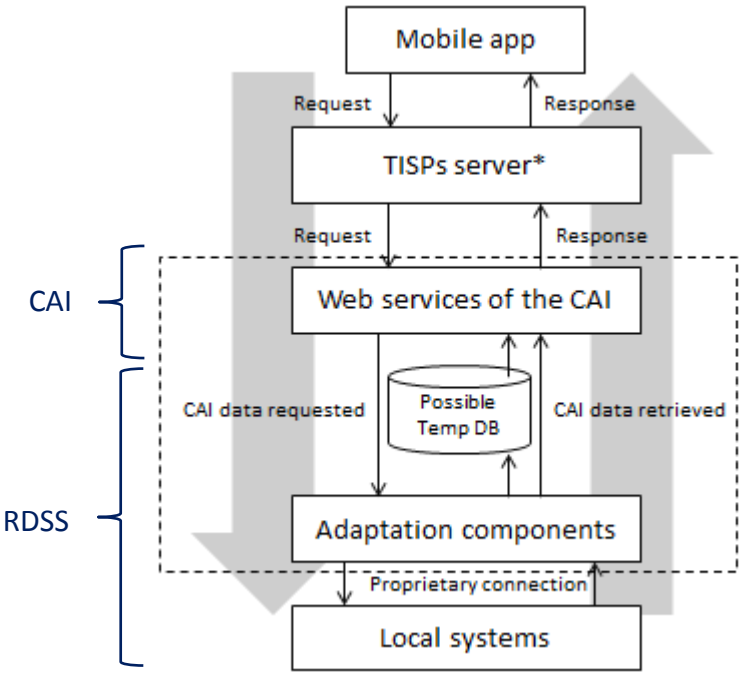

Figure 7 - Service provision to the end user

*the presence of a TISP server component depends of the organization and business model of the TISP.

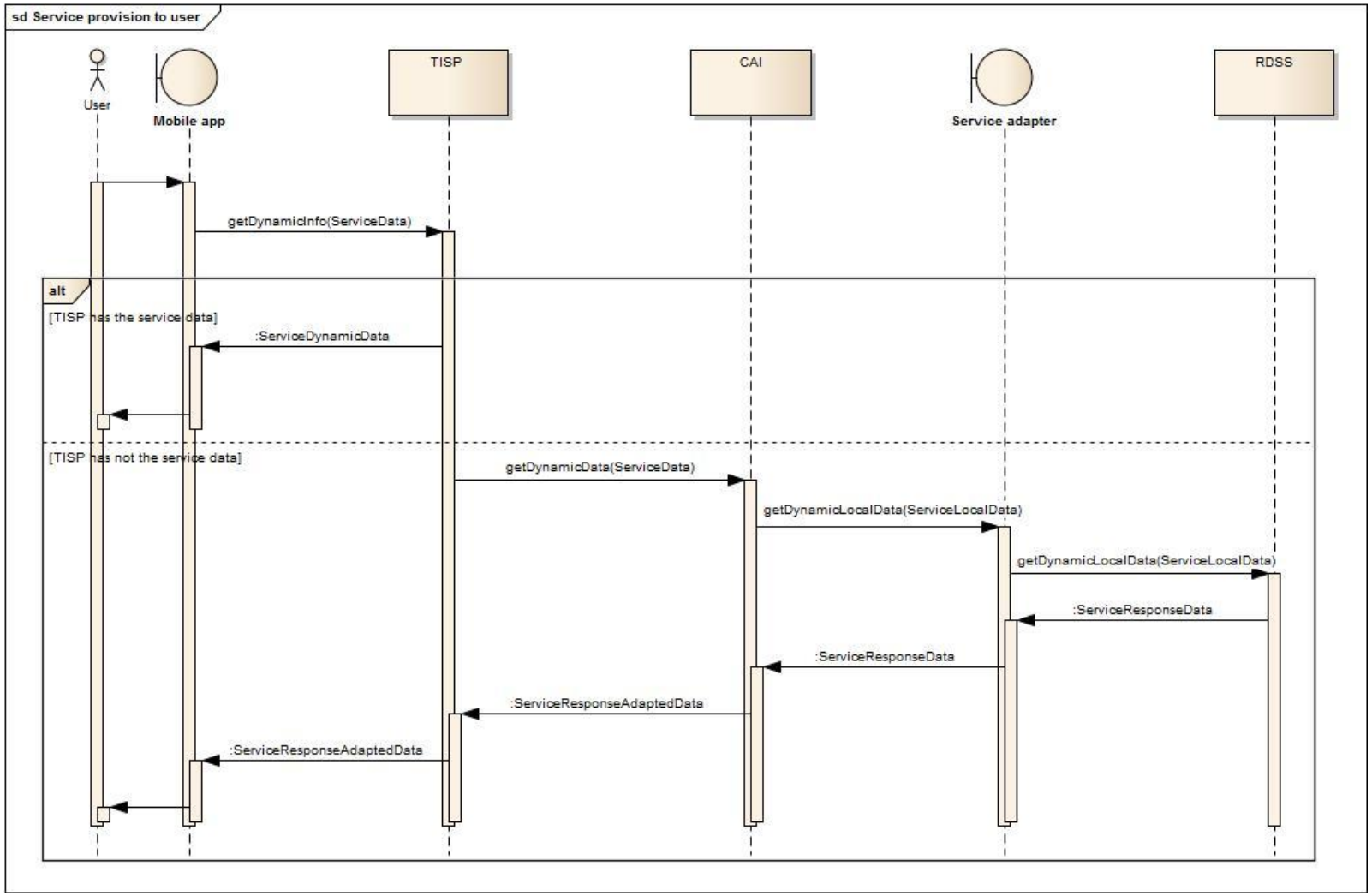

Figure 8 - Service provision to the end user – sequence diagram

4.1.2 Feedback data provision

The end user provides feedback to the RDSS servers, which will be used to improve the quality of the services provided by the RDSS.
The end user sends a feedback message to the TISP using the mobile app. The TISP and the CAI forward the information to the RDSS, where it is usually stored in a Data Base (DB) specifically defined for feedback data storage and for the Co-Cities project.
Once this information is correctly stored a confirmation message is sent back from the RDSS to the CAI.

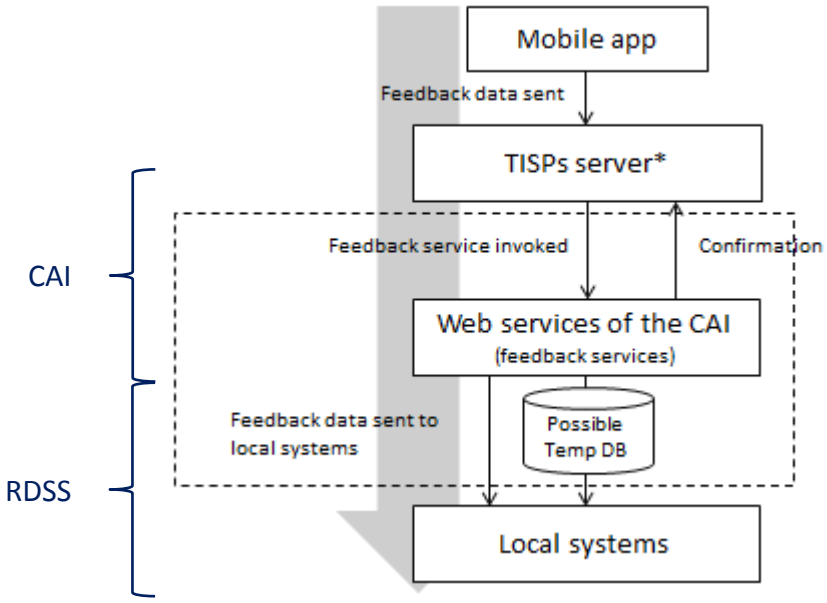

Figure 9 - Feedback data provision

Two exemplary application cases of this feature are the user sending feedback based on previously received information using the mobile application (Figure 10) or the user sending feedback related to the Co-Cities service quality (Figure 11). A third case is the provision of new data sets via the feedback channel, although this aspect, strictly speaking, can be considered a user feedback based on services received (the first application case described above).

In the diagrams the Feedback Result Message is the confirmation about the correct delivery of the feedback.

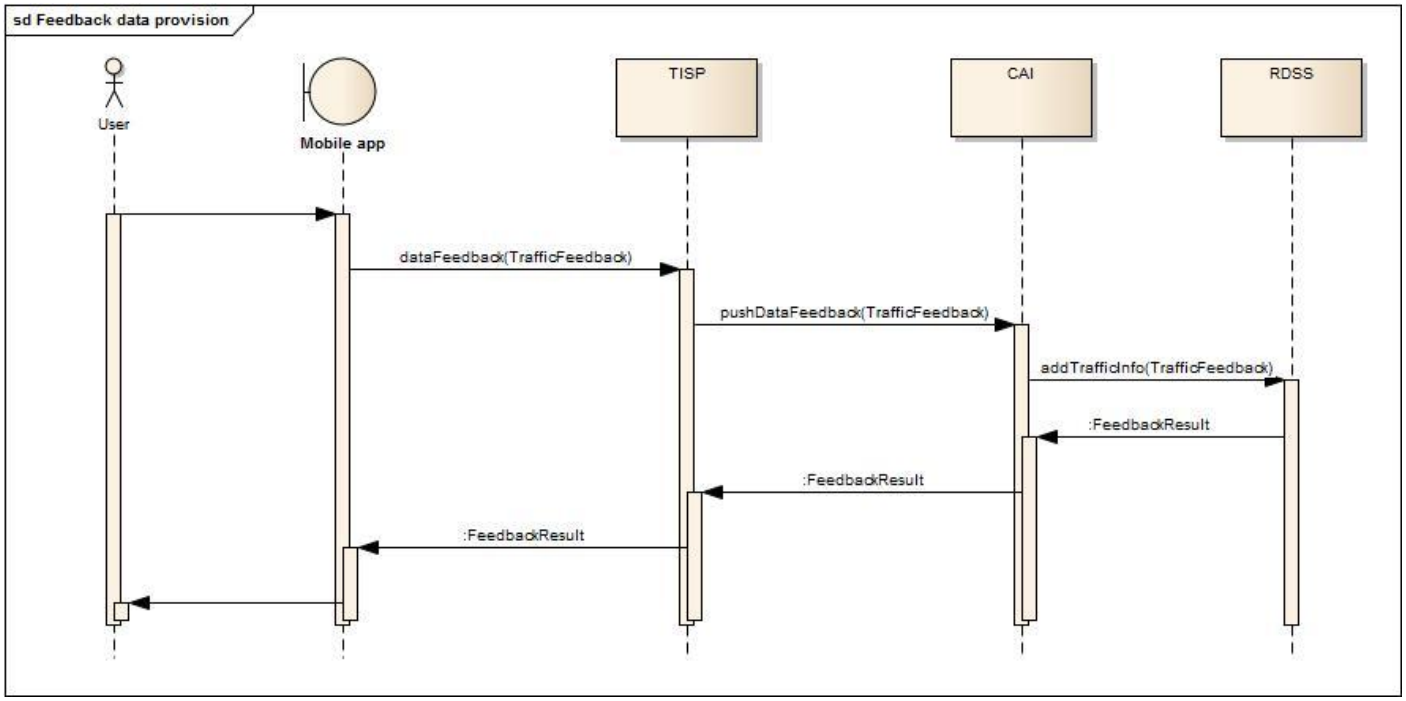

Figure 10 - Feedback data provision - Sequence diagram

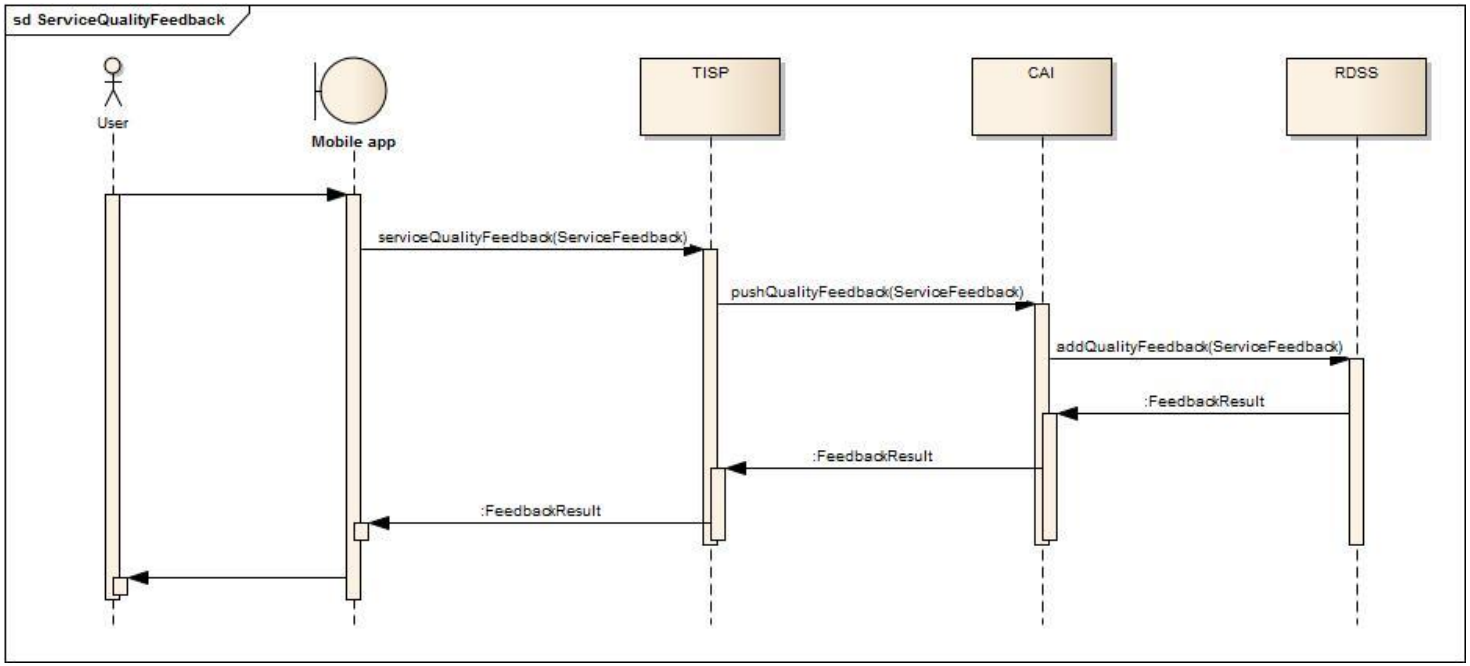

Figure 11 – Service quality feedback - Sequence diagram

4.2 Real scenario example

To ease the comprehension of the service delivery chain a real scenario example has been defined, related to both possible behaviours:

4.2.1 Example of service provision to the end user

A user downloads the Co-Cities app, related to his city, and installs it on his mobile phone.
Later, the user opens his Co-Cities mobile app and demands current traffic information (dynamic road traffic information).

From the user's mobile app a TISP web service is invoked, public method *getDynamicTrafficInfo*. This service has as parameter the name/number of a route of the road network.

The TISPs analyses the route. If it can directly provide the current traffic situation of the route it sends back the answer message to the app. If not, it adapts the incoming request message and invokes the CAI's dynamic service invoking the public method *getDynamicTrafficData*.

The CAI receives the incoming message and redirects it to an "Adaptation service component", which will modify the incoming request message and data, to adapt it to the RDSS, the local servers of the city.

The web service associated with the traffic information local server receives the request message, analyses it and sends a response message to the "Adaptation service component". Then, this adapter forwards the response message, modified to be adapted to the CAI's format.

The CAI, again, adapts the response message to be sent and understood by the TISP who send the corresponding message, typically in proprietary format, back to the mobile app.

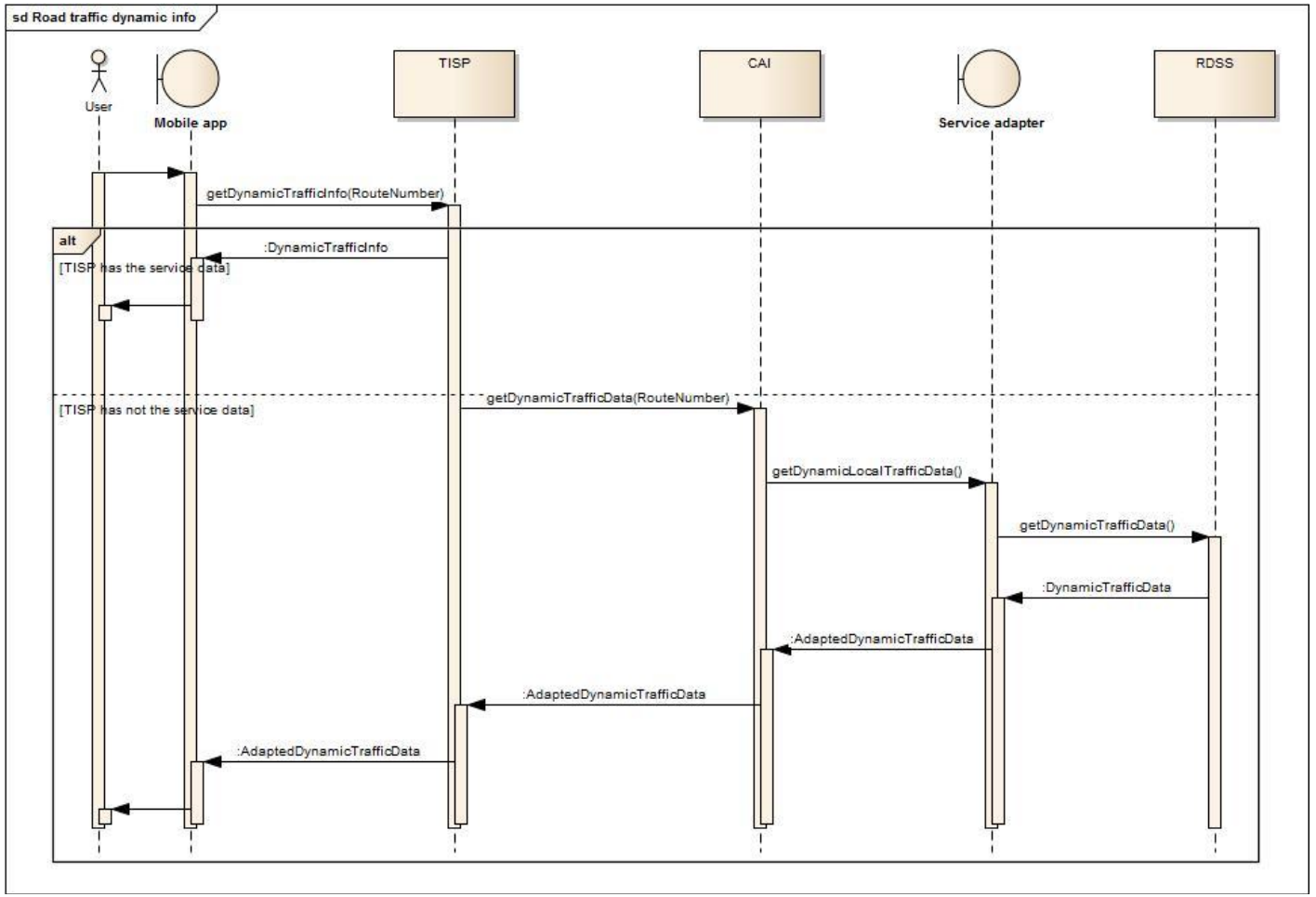

Figure 12 – Road traffic information service - Sequence diagram

4.2.2 Examples of feedback data provision

Example 1

The user encounters a traffic jam. The mobile app detects it (using the user's mobile GPS) and asks several questions to the user related to the traffic jam.

Once the app collected the required information, a traffic feedback (*TrafficFeedback*) is generated and sent to the TISP feedback service in proprietary format. The TISP adapts the format of the request message and sends it to the CAI's feedback service invoking the public method *pushDataFeedback* (*TrafficFeedback*).

While the feedback services of a city will be done *ad hoc* within the Co-Cities project, no "Adaption module" to reformat the feedback received by the CAI into local RDSS formats was required. Thus, the CAI are directly invoking the correspondent public method of the feedback service in the city servers with the feedback information. However, it might very well be possible (especially within an existing feedback treatment process environment) that a RDSS has specific format requirement concerning the feedback which have to be considered, very much as in the service delivery chain, by the CAI.

If this last operation is correct, the CAI sends a positive confirmation back to the mobile app (via the TISP). Else, the CAI sends a failure message to the mobile app.

Example 2

The user sends feedback to the RDSS related to the service quality of Co-Cities and its application. The service delivery chain behaviour is similar to the previous section.

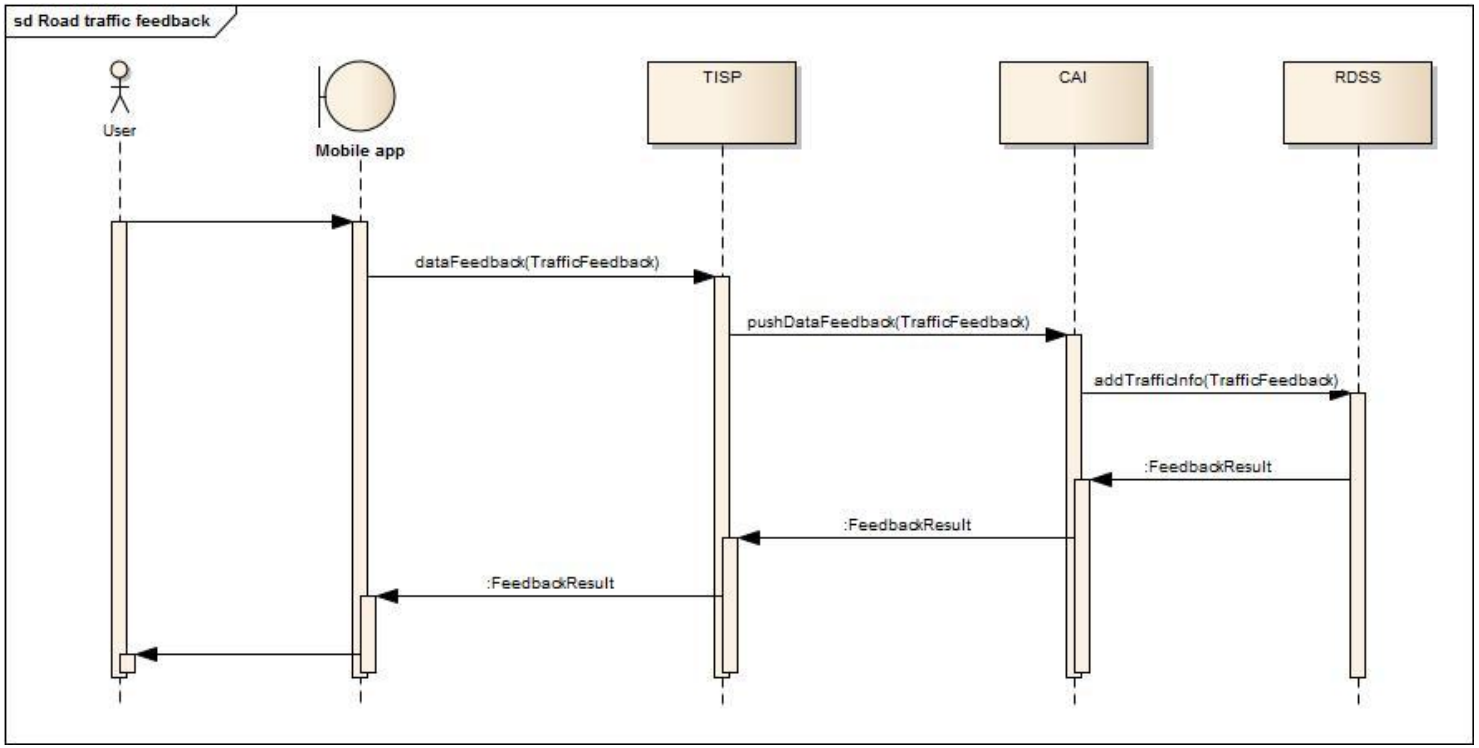

Figure 13 - Example of feedback data provision

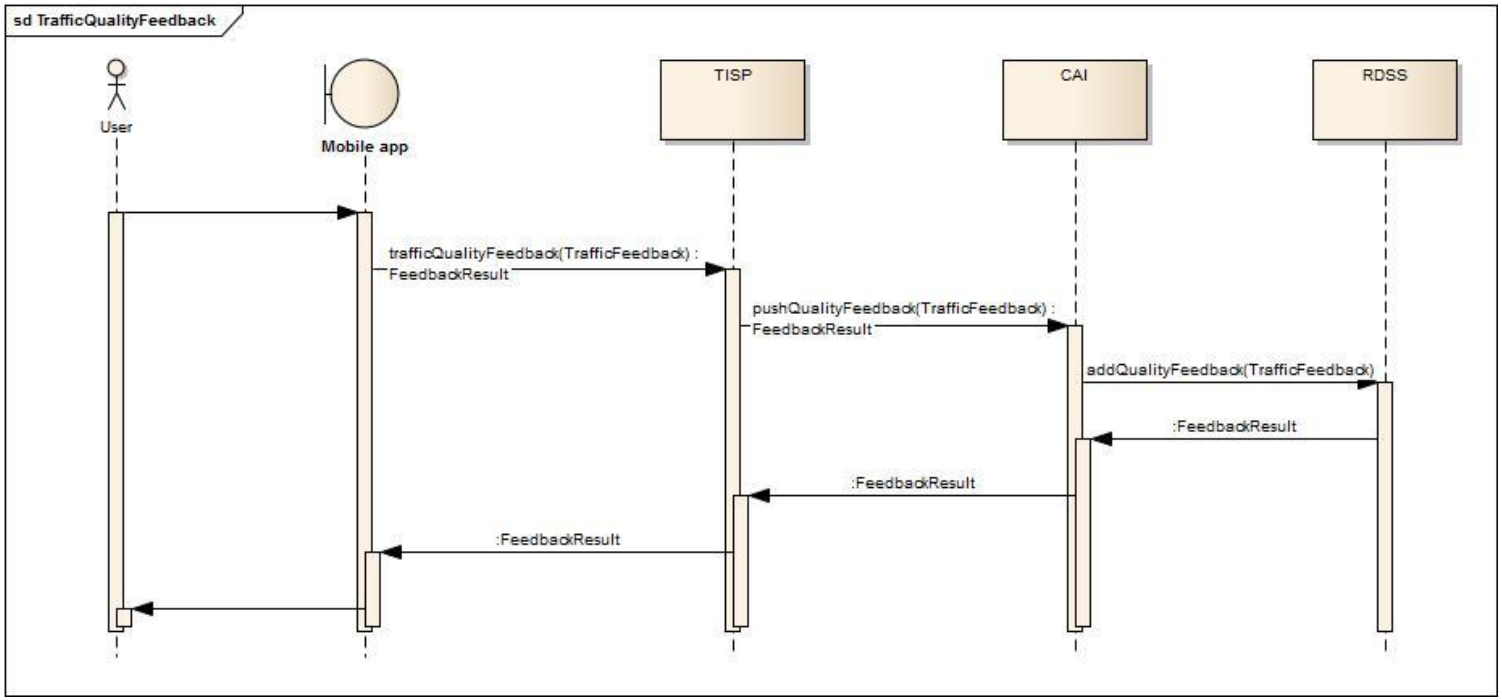

Figure 14 – Another Example of feedback data provision

4.3 Service provision to the end user after feedback processing

For the sake of completeness, in a complete real scenario example it can be interesting to figure out the possibility to provide new data to the end user as a consequence of (or as a data generated based on) the processing of feedback information.

In this case, feedback information can be collected by local systems via the RDSS who provides the necessary feedback services (part of the CAI) and act as a 'translator' to send the appropriate data to the specific local systems who needs it. This is done according to the previous functional schemas described for feedback data provision.
Local systems can process feedback data according to their specific policies and objectives (see also the Co-Cities Deliverable 3.1 [4]) and this may result in the generation of a new piece of information which can be eventually redistributed via the Co-Cities CAI. This might e.g. be the case if several end users report a congestion event which is not yet contained in the respective end user service delivery.

Once the new information is available the process of re-distribution follows exactly the same steps previously described by the functional schemas related to the process of information provision. This stage of the flow is described in Figure 15.

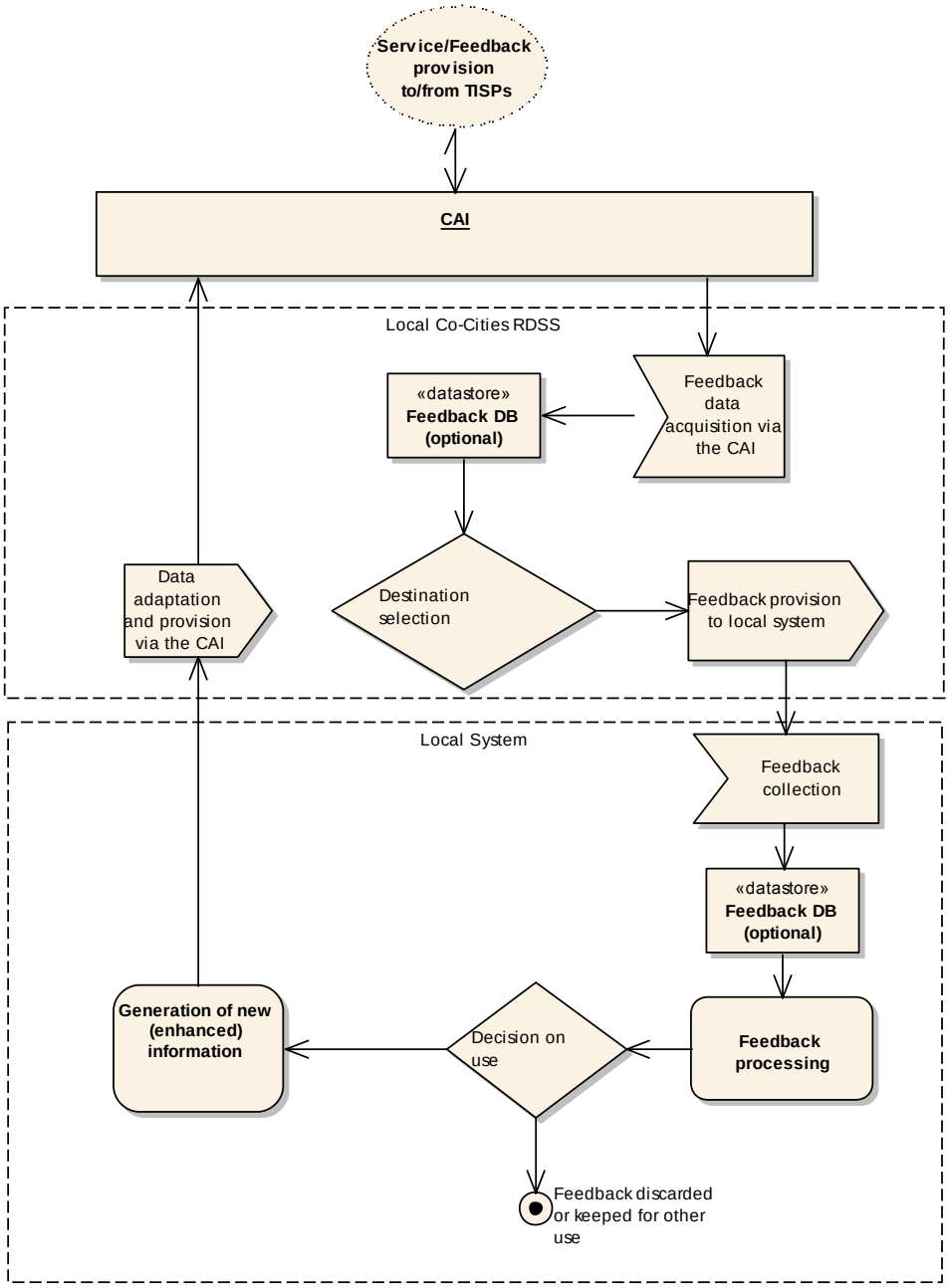

Figure 15 - New processed feedback data to the end user

5 Report on developed B2B interfaces

This chapter reports on the B2B interfaces which have been developed in the Co-Cities pilot sites. For each city a development report of the B2B interfaces (including feedback) is provided.

This includes:

- Introductory contents
- Content and Service provision: a report of the B2B Service interfaces for service provision developed on top of existing local contents and services
- Feedback interfaces: a report of the B2B interfaces developed in order to collect feedbacks. These interfaces are used by the TISPs to be fed by data entered by the end user by means of the interactive functionalities of the mobile apps
- Technological choices: describes which technology has been used for the implementation of the B2B service interfaces and for the implementation of feedback service interfaces (e.g. WFS servers DBMSs etc.)

For a better understanding and identification of the content and service offerings from each site it has to be remarked that the end user services are provided by the TISPs by using the necessary B2B services (interfaces) only under the following conditions:

- The B2B service for the provision of information must provide concrete (no empty) data.
- For each B2B feedback service associated to the overall quality of service or to the quality of data, a B2B service for the provision of information must exist in the same service domain. The provision of new data for existing parking places is also available not only where the necessary B2B feedback service exist but also where the related services for the provision of parking information is in place.

Under these conditions the availability of a B2B interface in a specific domain and in a certain site does not necessarily implies its usage for the end user service provision. The TISPs operated the appropriate selection of B2B services in each site in order to offer a correct and consistent offering to the end users.

Chapter 8 reports on the validation results and provides the exact view in terms of end user functionalities for each combination of TISPs and pilot sites.

5.1 Bilbao

Bilbao was not part of the In-Time project and therefore no CAI existed there. Therefore both the data and the feedback services had to be addressed.

After analyzing the existing resources that were available at the Bilbao Council it was decided, based on a functional point of view and on the available data, that the best line of action was to implement the CAI integrated with the so called ICM (Integrated City Management) platform,

already deployed at the Bilbao Council. The reasons for that lay in the fact that the ICM platform was already receiving and integrating information originating from the city Council databases, so significant parts of the data integration required for the implementation of the CAI were already achieved.

The available data feeds within the city Council were reviewed and the following B2B service domains were identified for Co-Cities:

- Traffic information, both static and dynamic.
- Information about parking lots, both curbside and on parking lots, including the position of the lots and its current occupation.
- Information about the bus lines that lie in the responsibility of the city council, including the location of the stops, the schedules and more.

Other functionalities like a routing service were excluded, the development being focused on providing the needed information over the CAI so third parties could offer this service in the future.

On the other hand, the feedback services belonging to the CAI as defined by the Co-Cities technical specification were developed separately, given the fact that the ICM platform, in its existing form, was not going to be able to support these services.

The technological partners that carried out the development of the CAI in Bilbao were:

- **Telvent**: in charge of the implementation and deployment of the data services belonging to the CAI as defined in the In-Time project.
- **Softeco**: in charge of the implementation and deployment of the feedback services as defined by the Co-Cities specification.

5.1.1 Content and Service provision

The table presents the B2B service interfaces developed in Bilbao and the service domains associated to these services.

B2B Service	Service domains covered
Public Transport	Static and dynamic public transport data
Parking	Static and dynamic parking data
Traffic	Static and dynamic traffic data

Table 3 - B2B interfaces developed in Bilbao

5.1.2 Feedback interfaces

The table presents the B2B feedback service interfaces developed in Bilbao and the related service domains to which these services are associated.

Feedback Service	Service domains
Overall quality of service	Traffic, Parking, Public Transport
Feedback on quality of provided data	Traffic, Parking, Public Transport
New data	Traffic

Table 4 - Feedback services interfaces developed in Bilbao

5.1.3 Technological choices

As said above, the implementation of the CAI in Bilbao was developed based on the so called ICM platform. An overall diagram of the way that integration was achieved can be seen in the following picture:

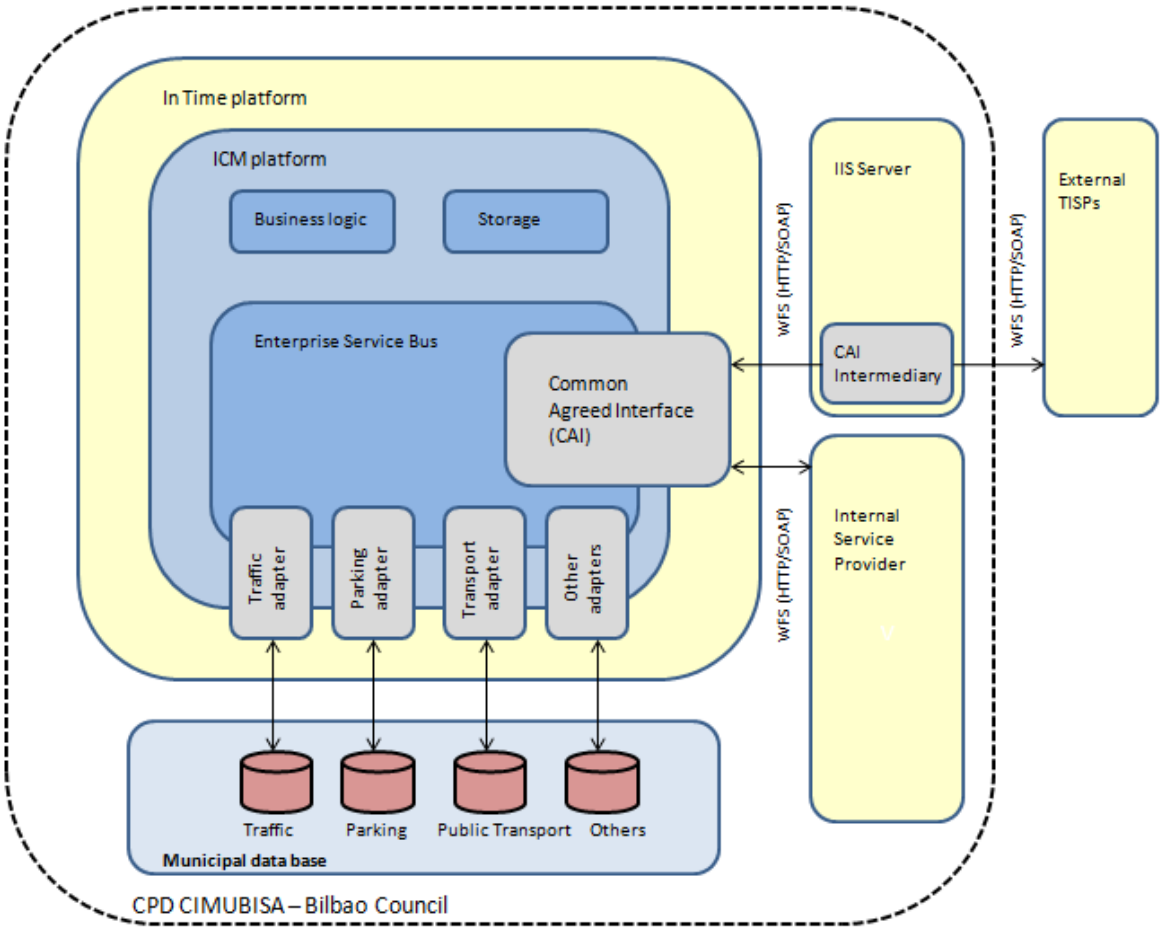

Figure 16 – Implementation of the CAI over the ICM platform at Bilbao Council

5.1.3.1 The ICM platform

This platform provides a whole range of functionalities related to the smart city concept, mainly focused, at this time, on urban mobility issues. Therefore, it is capable of receiving information from many different and heterogeneous sources, integrating it into a well-defined data model and storing it so that the business logic can take advantage of it for providing end user functionalities.

It is composed of the following Functional Entities:

- **Enterprise Service Bus**: using a flexible architecture, currently based on *Mule*, it is capable of receiving and integrating information and sending it to the storage Functional Entity.

- **Storage**: the received information is stored in different ways depending on its nature, so a relational database is usually used for most of the data, while also supporting other alternatives like *NoSQL* storage or *pub-sub* spaces. A *Microsoft SQL Server 2008* relational database is currently used in Bilbao.

- **Core and Business Logic:** it is here that the business logic, related to traffic, parking and public transportation is provided. A modular *SOA* architecture is used here, deployed over a *JBoss J2EE* container.

- **Operator Console:** it is the graphical user interface that operators in the traffic control center of the city council use to get information about the current status of urban mobility in the city and react to upcoming events. This user interface is accessible using a web browser and is based on *Flex* technology.

- **Public Web:** the platform is capable of providing information to third parties in addition to the operators working for the organization where the platform is deployed. A web interface is also provided to be used by the common public.

- **Business Intelligence:** in all applications where this is required, a separate functional entity provides analysis capabilities to extract higher level information from the stored data stored.

The development of the CAI interface involved the Enterprise Service Bus, the Storage and the Core and Business Logic Functional Entities. The details are described in the following section.

5.1.3.2 The CAI implementation

The development of the CAI required modifying several parts of the ICM platform, mainly to add new functionalities to them. The parts affected were the following:

- **Enterprise Service Bus**: providing the CAI data services accessing the information stored in the city council databases that was not used by the ICM platform before. New adapters have been developed to get information from the relevant databases.

- **Storage:** as new information sets have to be stored, the ICM data model was extended to support the new needs.

- **Core and Business logic**: in order to coordinate the execution of the new adapters within the Enterprise Service Bus, as well as the storage of the information, the necessary control logic was added to this Functional Entity.

Once these additions to the ICM platform were made, the information to be provided through the CAI and its data services was available into the *ICM* data model and stored according to it. The information stored within the ICM database was used to provide the *WFS* interface of the CAI itself.

That was achieved by deploying an instance of the Open Source platform *GeoServer* providing the *WFS* interface and accessing the needed information from the ICM database. The data is translated according to the data model defined in Co-Cities by means of the corresponding configuration files.

5.1.3.3 Deployment

For the implementation and deployment of the CAI in Bilbao the following steps have been followed:

- **First stage**: during this first stage, systems belonging to the council have not been affected as the development was performed within the premises of Telvent.

- **Second stage**: after carrying out a first validation of the developed system, the computing center of Bilbao Council has been involved. A virtual machine has been requested for the deployment of *GeoServer*, while the extensions to the ICM platform have been deployed in a pre-existing virtual machine where the ICM platform already resided.

  The operating system of both machines is Windows Server 2008 R2. Full connectivity between both machines has been established while considering that the *WFS* interface provided by the *GeoServer* had to be freely accessible from the Internet. This has required modifying the relevant rules of those firewalls used in the data center.

- **Third stage**: once the validation tasks of the previous stage have been finalized, the possibility has been considered to deploy *GeoServer* on the same machine as the ICM platform lay. However, this possibility has been finally discarded to keep the ICM platform and the *GeoServer* separate and ease their management.

5.2 Munich

The Munich test site already participated in the In-Time project providing different In-Time services like parking data, road weather and road traffic information, POI information as well as fully dynamic intermodal journey planning services.

Unfortunately, related to a change in the backbone system and the changed provision of the intermodal journey planner results implemented within this process, new contractual ancillary conditions emerged. These ancillary conditions are not suitable to support the In-Time services approach which includes, as required players/TISPs who provide the regional information in a super-regional frame also on basis of professional (and paid) services.

Hence Munich cannot, in contrary to the original planning, provide the intermodal routing services via the In-Time interfaces until an updated regulation can be defined with the local actors. These experiences will be utilized within the discussions of the Co-City concept with cities to understand how an applicable process for long-term service provision should be designed to support the long-term engagement of both, RDSS and TISPs alike.

Munich thus provides parking services and traffic information services in the frame of Co-Cities. While parking services cover the user feedback, the traffic information services contain the submission of new data (e.g. a user-detected congestion event).

5.2.1 Content and Service provision

The table presents the B2B service interfaces developed in Munich for content and service provision and the Service domains associated to these services.

B2B Service	Service domains covered
Parking	Static Parking Data
Points of Interest	Static POI Data
Traffic	Road Traffic Messages

Table 5 – B2B interfaces developed in Munich

5.2.2 Feedback interfaces

The table presents the B2B feedback service interfaces developed in Prague and the related Service domains to which these services are associated.

Feedback Service	Service domains
Overall quality of service	Road Traffic, Parking[1]
Feedback on quality of provided data	Parking
New data	Road Traffic

Table 6 – Feedback services interfaces developed in Munich

5.2.3 Technological choices

The In-Time CAI was already implemented on top of the VIB services within the In-Time project based on the In-Time specifications.

The Co-Cities feedback interface was implemented as a set of software components according to the Co-Cities specifications. The Co-Cities feedback interface is, strictly speaking, independent of the In-Time Service interface.

As data storage, the Co-Cities client system stores the received feedback and provides it visually on a screen and in a list for the traffic editor to review and for taking over into the live system service environment.

From technical perspective, the feedback management system utilizes Windows Server 2008 and a jBOSS 7.1.1 application server environment for the required processes to receive and manage the incoming feedbacks. The following figure provides a short overview on the part of the feedback management system:

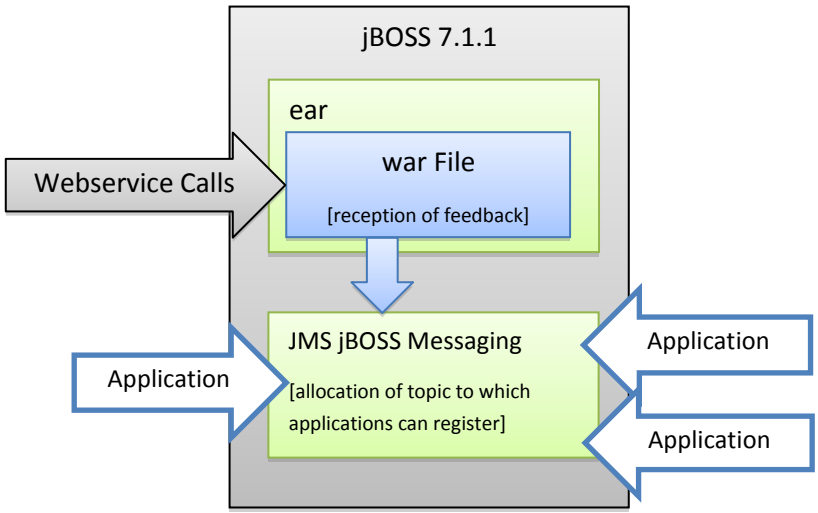

Figure 17 – Feedback management system in Munich – data management and distribution

[1] Technically, the "overall quality of service" is sent with one single web service for all domains. A specific domain is selected with a parameter. The information in the table indicates the domains for which the information is actually provided thus making this type of feedback applicable in practice.

Additionally, the system comprises a visualization unit, which presents the feedback to the operator as well as a tool that allows generating (operator-side) feedback utilizing the same technology as “normal” end user feedback.

The following figure provides a short overview about the components mentioned above and their interaction.

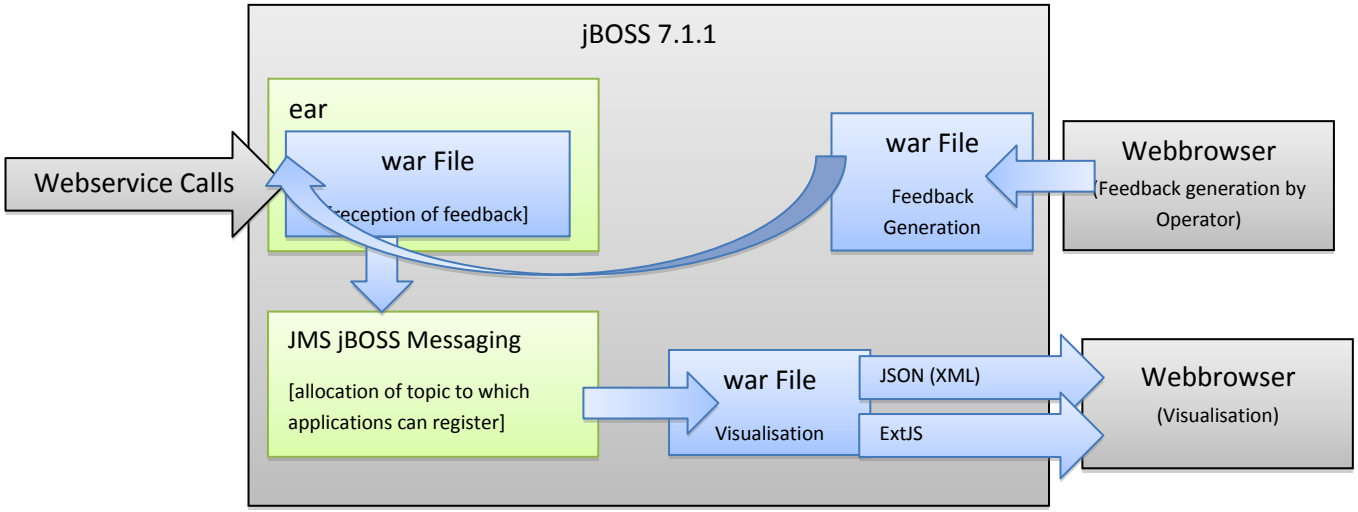

Figure 18 – Feedback management system in Munich – Visualisation and Operator Feedback Generation

For visualization and feedback generation by the operator (e.g. if the operator receives a feedback by phone), web browser technology has been utilized.

The visualization is realized utilizing open layers (for the map part) and extGWT (for the table part) technologies. The following figure illustrates the visualization tool.

Figure 19 – Operator´s screen in Munich

For the feedback generation, the Google Windows Toolkit was used to compile the related code into Java Script and enable the generation of Co-Cities feedback from a web browser (see figure below).

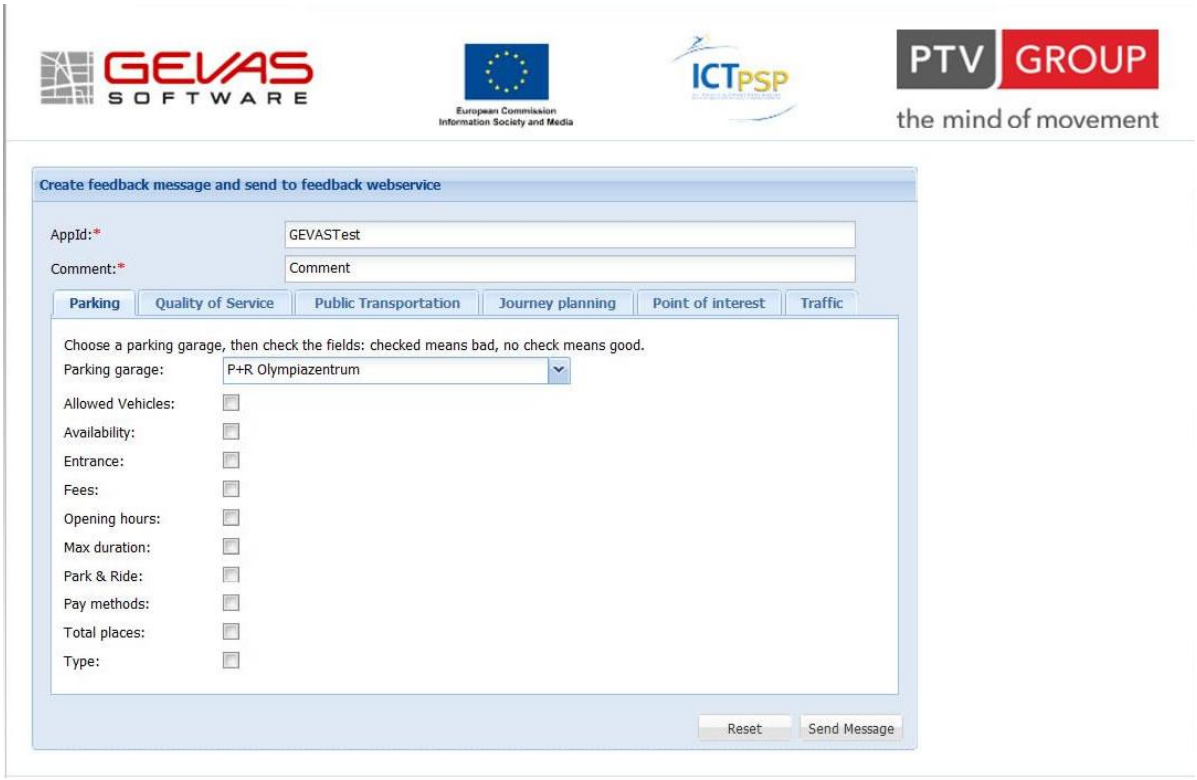

Figure 20 – Feedback generation for the operator in Munich

As the technologies which have been implemented are well known and standard in today´s IT domain, the solutions developed in this project can easily be extended to other feedback types in the future.

5.3 Prague

Prague was not part of the previous In-Time project and therefore the entire CAI had to be implemented from the beginning. Based on the available local services it was decided that the main focus would be on the multimodal journey planning service.

The company ROPID is responsible for managing public transport information for this region. Most of this information is publicly available online in a static form as timetables mainly in a PDF format. The RDSS for Prague uses this static information to build a local database of stop points and a search graph, which in return is used for journey planning service.

The data adaptation according to Co-Cities requirements was made only once. This solution requires no further communication with local services when processing incoming requests.

Two main types of In-Time B2B services are available in Prague site:

a) **Location-Based Services** in the domain of Public Transport. It provides locations of bus and train stops.
b) **Multi-modal Journey Planning** using public transport, car, walking and bicycle modes.

Feedback services are also available as part of the web services solution. Feedback data is stored locally in a database for future processing.

5.3.1 Content and Service provision

The table presents the B2B service interfaces developed in Prague and the Service domains associated to these services.

B2B Service	Service domains covered
Public Transport	Public Transport
Journey Planning	Journey Planning, Road Traffic, Walking, Public Transport

Table 7 – B2B interfaces developed in Prague

5.3.2 Feedback interfaces

The table presents the B2B feedback service interfaces developed in Prague and the related Service domains to which these services are associated.

Feedback Service	Service domains
Overall quality of service	All
Feedback on quality of provided data	All
New data	Public Transport, Journey Planning

Table 8 – Feedback services interfaces developed in Prague

5.3.3 Technological choices

The CAI was implemented as a set of software components along the Co-Cities specifications. The solution does not require steady a connection to local services. All required data are preprocessed and stored in a database or proprietary file formats.

The entire RDSS solution uses Windows Server 2008 R2. Microsoft SQL Server 2008 R2 is used for storing public transport stop points and raw feedback data.

Multimodal journey planning service uses Dijkstra's algorithm for public transport mode and A* algorithm for other modes. Graphs for public transport search algorithm were generated from static time tables. Graphs are stored in a proprietary format that enables effective search on this graph.

On top of these components there are web services hosted in IIS 7 and WFS server that together provide a CAI interface. Web services are implemented using Microsoft .NET Framework. A skeleton version of each service was generated by a tool wsdl.exe that is part of the .NET Framework tools. WFS server is based on a custom solution. It is designed specifically to transform WFS requests to SQL commands. Majority of the operations are performed in a database. Requests that cannot be fully translated to database commands are executed in a database as much as they can be and the results are processed "locally" – i.e. in the WFS server code.

5.4 Reading

In Reading it was necessary to build the entire CAI including the In-Time part. Reading provides public transport data, car park static and dynamic data.

Two main types of In-Time B2B services are available in Reading site:

a) **Location-Based Services** in the domain of Public Transport and Parking. This includes locations of bus and train stops and available parking places.
b) **Multi-modal Journey Planning** using public transport, car, walking and bicycle modes.

Feedback services are also available as part of the web services solution. Feedback data is stored locally in a database for further processing.

5.4.1 Content and Service provision

The table presents the B2B service interfaces developed in Reading and the Service domains associated to these services.

B2B Service	Service domains covered
Public Transport	Public Transport
Parking	Parking
Journey Planning	Journey Planning, Road traffic, Walking, Public Transport

Table 9 – B2B interfaces developed in Reading

5.4.2 Feedback interfaces

The table presents the B2B feedback service interfaces developed in Reading and the related Service domains to which these services are associated.

Feedback Service	Service domains
Overall quality of service	All
Feedback on quality of provided data	All
New data	Public Transport, Journey Planning

Table 10 – Feedback services interfaces developed in Reading

5.4.3 Technological choices

Reading was developed using similar technological choices as in Prague. This includes:

- Microsoft Windows Server 2008 R2
- Microsoft SQL Server 2008 R2
- Microsoft IIS 7
- Proprietary WFS server
- .NET Framework

Reading provides public transport timetables in two formats. The first one is in CIF (Common Interface Format) format stored in files and the second one uses SOAP to form a database requests. RDSS for Reading was built using the first method. Search graph for Dijkstra's algorithm was generated from CIF files.

Car park static and dynamic information is available in a proprietary format using HTTP requests. RDSS periodically downloads data and stores it in a local MS SQL database to satisfy incoming requests from CAI. Skeleton code for web services was generated from provided WSDL files using the wsdl.exe utility.

5.5 Tuscany Region

The B2B CAI services in Tuscany Region extend the In-Time service offerings currently existing in the Regional Capital Florence.

This extension is possible by accessing regional systems, de-centralized and centralized, providing data and services for Co-Cities.

The de-centralized systems mainly originate from In-Time and are already connected to the In-Time CAI. These include the AVM system of Florence's Public Transport Company ATAF providing dynamic Public Transport information, the "BusBussola" Journey Planning System providing journey planning in Florence and the Parking system managed by Firenze Parcheggi. These are already connected to In-Time.

The main central node of the Regional Transport and Traffic infrastructure is the **Mobility Information Integration Center (MIIC)** where data from local Administrations is collected into a

unique data repository by means of a dedicated Regional Network which connects all de-centralized organizations who are able to provide local data using a common, agreed protocol for data exchange (thus adopting the same basic principles of Co-Cities).

The second main Centralized System used by Co-Cities in Tuscany Region is the **Regional Journey Planner,** which provides an integrated Journey Planning service at regional level.

Technologically, each local source (centralized and de-centralized) is connected to an adaptation **component,** which translates the contents into the In-Time/Co-Cities format. This can be done regardless the specific physical connection, which is often imposed by local constraints. The provision of data is then achieved using the CAI web services set up locally along the indications of the Co-Cities specification.

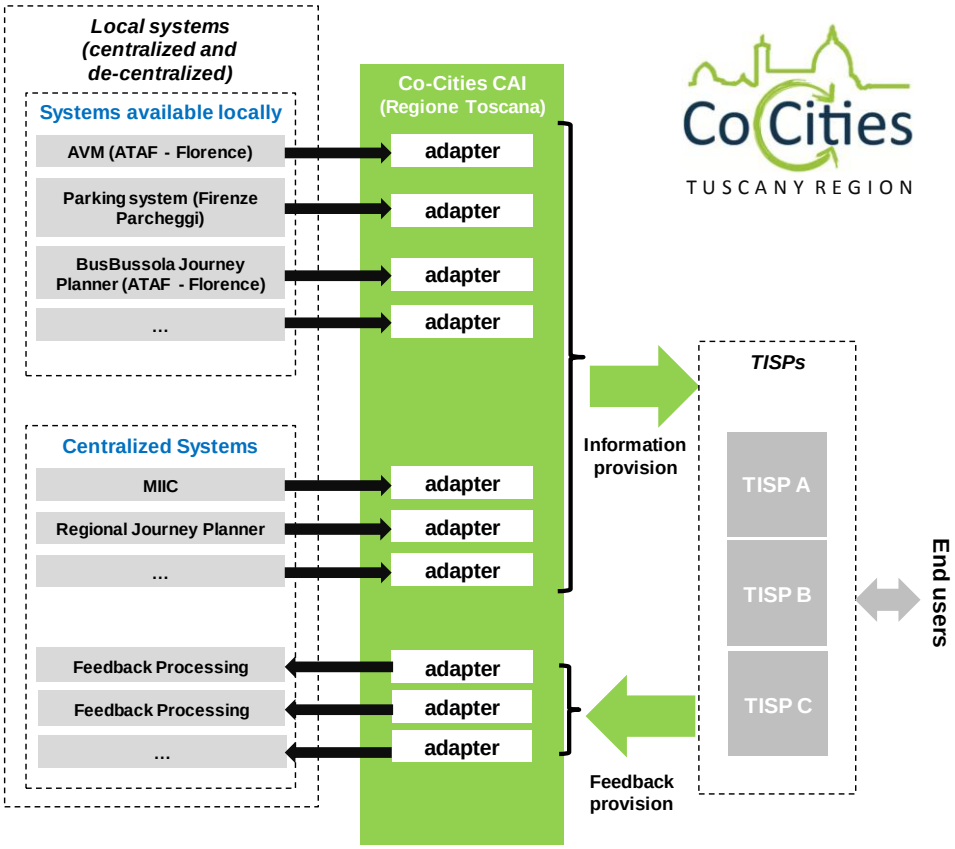

Figure 21 – Co-Cities in Tuscany Region - Functional view

Two main types of In-Time B2B services are available in Tuscany region site:

a) **Location-Based Services** in the domain of Public Transport, Parking and Road Traffic. This includes: location of bus stops and dynamic schedules for the stop, real-time information on free places available in parking and dynamic road traffic events.

b) **Multi-modal Journey Planning** using public transport, car and walking modes.

Feedback Services are also available as web services that can be invoked by TISPs for feedback data provision. Feedback data is sent by the feedback services to a **Feedback repository** for further processing.

5.5.1 Content and Service provision

The table presents the B2B service interfaces developed in Tuscany Region and the Service domains associated to these services.

B2B Service	Service domains covered
Public Transport	Public Transport
Parking	Parking
Traffic	Traffic
Journey Planning	Journey Planning, Road traffic, Walking, Public Transport

Table 11 – B2B interfaces developed in Tuscany Region

5.5.2 Feedback interfaces

The table presents the B2B feedback service interfaces developed in Tuscany Region and the related Service domains to which these services are associated.

Feedback Service	Service domains
Overall quality of service	All
Feedback on quality of provided data	All
New data	Traffic

Table 12 – Feedback services interfaces developed in Tuscany Region

5.5.3 Technological choices

The CAI is set up as a set of software components connected to local systems and providing web service interfaces along the Co-Cities specifications.

All nodes (regional systems and the CAI software components) are connected each other via the Internet or by means of dedicated connections depending on local constraints. A custom connection from a local system to the CAI components is typically necessary because of the specific

infrastructural conditions or security policies, which can be found locally. Custom connections include:

- Custom web services available or developed on top of local systems and invoked by the CAI components, through a VPN if necessary
- Datasets available via HTTP requests, over a VPN if necessary

The group of web services of the CAI forms the Co-Cities node which is, in principle, not physically associated with any of the existing regional nodes and can run in an independent infrastructure. It can be set up either in a dedicated physical server or run on an existing server.

The CAI includes both the In-Time B2B services and the Feedback Services.

The processing of **Feedback information** is carried out by nodes which can be called **Feedback Processing Units** typically run by Local Authorities. These need to be connected to the Co-Cities Feedback repository which is fed by the Co-Cities Feedback Services.

In principle the Central Administration (Regional Authority) can have a Feedback Processing Unit but also other de-centralized authorities like Public Transport Companies, Traffic Management Centers etc. can have units for processing feedback data in the respective, specific domains.

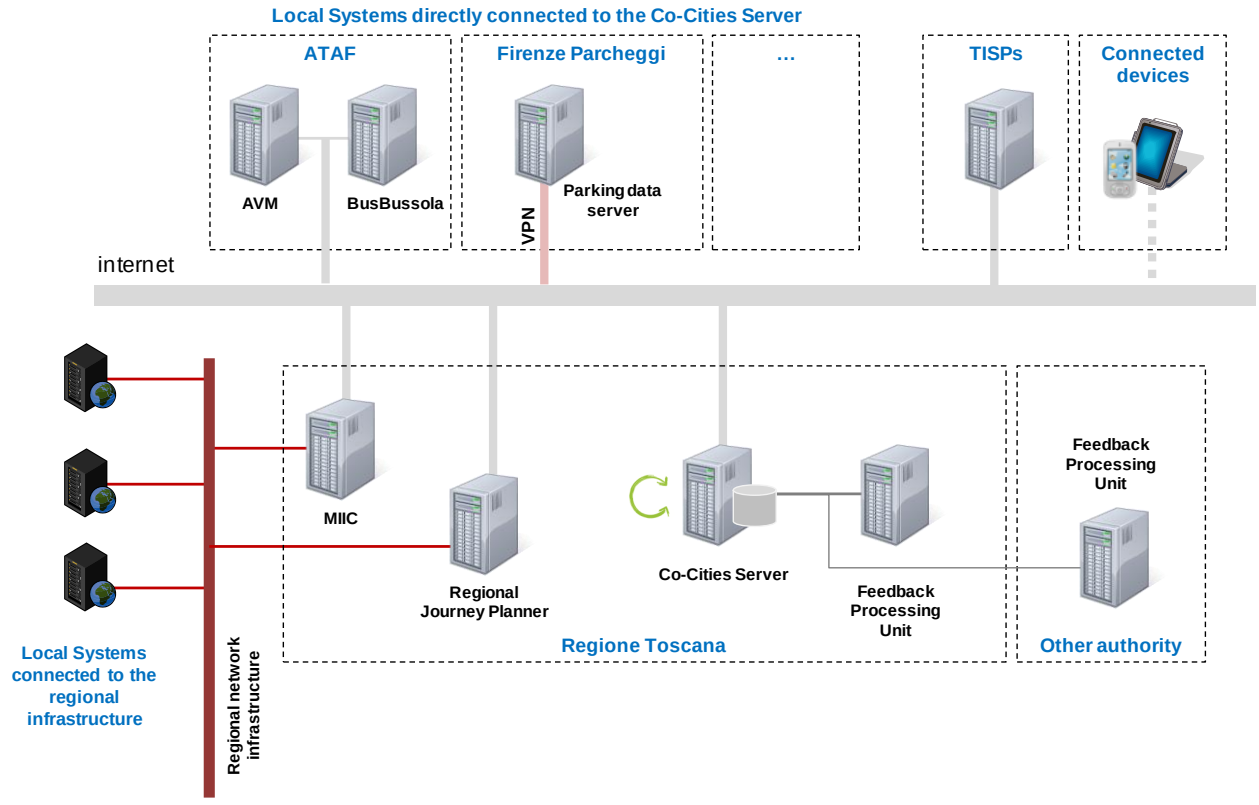

Figure 22 – Co-Cities in Tuscany Region - Physical view

The integration of the Regional Data/Services is achieved on the basic Co-Cities requirements:

- Co-Cities specifications developed in Deliverable D3.1 [4] and subsequent extraction of necessary features (simplified XSDs) - at data translation level;
- Use of WFS/WMS interfaces and other service interfaces described by the appropriate WSDL enabling TISP to make requests and get responses from the local B2B server
- Use of feedback services documented as part of the Co-Cities technical specification

The technological choices for the integration of the components underlined by the above points can be summarized as follows:

- Use and configuration of GeoServer Open Source WMS/WFS Server platform for data services
- Set up of a temporary DB (Data Base) jointly with GeoServer (Postgres)
- Local data adaptation achieved through the development of a mapping scheme between the existing data format and the model adopted in the CAI; this is achieved by developing specific software components (adapters)
- Local Service adaptation by means of a layer added in top of the existing services, able to wrap and adapt them to the Co-Cities Infrastructure (e.g. this applies to Journey planning service)
- Custom implementation of feedback services

B2B services for the provision of Information

For Location Based Service B2B provision, OGC WFS data services are used as indicated by the Co-Cities specification.

GeoServer[2] is an open source software server written in Java and a reference implementation of the OGC Web Feature Service (WFS), Web Map Service (WMS) and Web Coverage Service (WCS).

In the Tuscany Region the GeoServer is used to develop the OGC-based part of the Co-Cities service interface. This includes all proper In-Time services (except for the Journey Planning service). GeoServer works jointly with a supporting DB in Postgres technology.

[2] http://www.geoserver.org

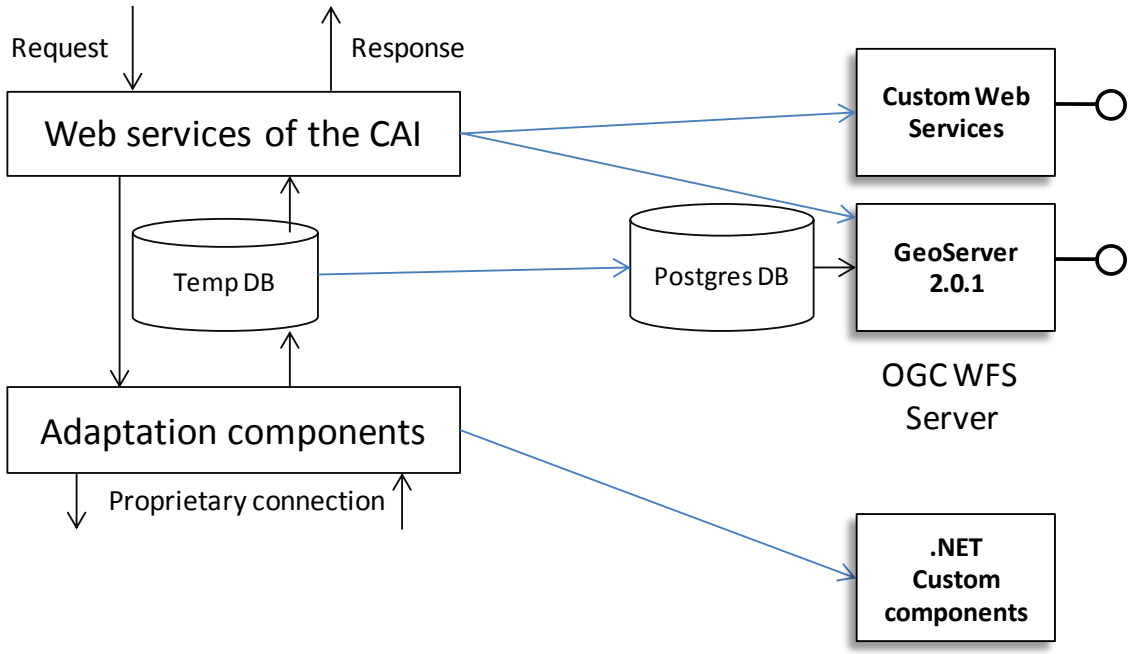

Figure 23 – Information request processing and related components in Tuscany Region CAI

Data adaptation through WFS / WMS services is obtained by specifically set up GeoServer using the XSD schemas[3]. At the same time, the Postgres DB temporarily stores all datasets which have to be provided through the WFS server. The DB is filled by data fetchers, adaptation components built as .NET Custom entities. These are responsible for retrieving data from local systems and to adapt them/store them in Postgres DB.

[3] GeoServer includes an extension, called Application-Schema which enables a two-ways mapping of simple features into complex ones. This is especially useful in order to use the Co-Cities XSD-based specification where objects are often represented by means of complex features.

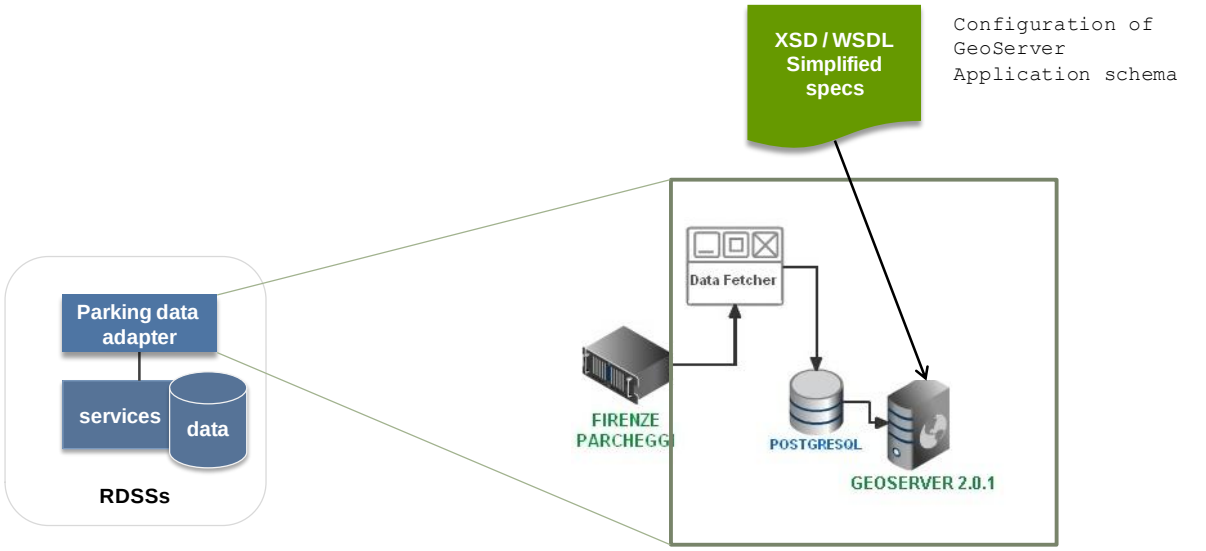

Figure 24: Structure of a data adapter using GeoServer in Tuscany Region

The process of adaptation and provision through the Co-Cities CAI is different for the Journey Planning Service. In this case the Adaptation process can be simply intended as a wrapping operation over the existing local service provision. No temporary storage is necessary in this case.

The custom service wrapper is created in .NET technology along the directives of the Co-Cities specifications.

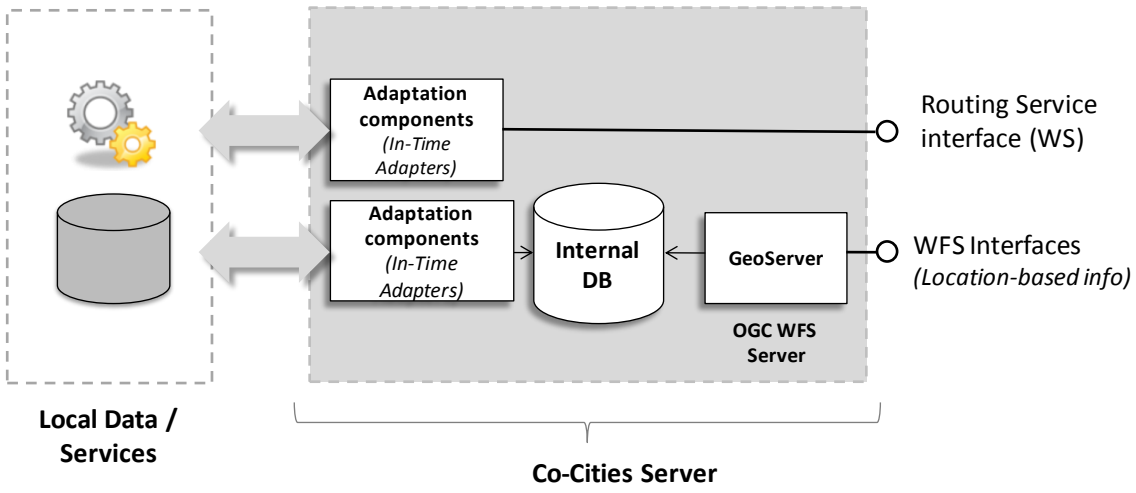

Figure 25: CAI implementation in Tuscany Region

Feedback functionalities

The achievement of Feedback functionalities in Tuscany Region is based on the development of a set of services as part of the Co-Cities CAI. From a technological point of view these are Custom Web Services developed in .NET technology.

A supporting DB (Postgres) is used in order to store data retrieved by the Feedback services before they are dispatched to local processing units.

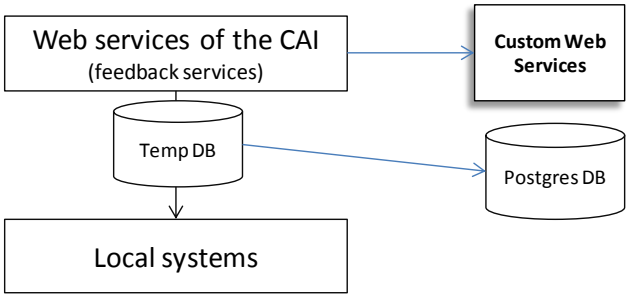

Figure 26 – Development of feedback services in Tuscany mapped to the service chain schema

5.6 Vienna

While the In-Time CAI and the services 9 (parking) and 17 (multi-modal journey planner) were already implemented and described during the In-Time project, the development and implementation of the feedback services i.e. the CAI Co-Cities extension and its web services for feedback data provision are described in the following subchapters.

5.6.1 Content and Service provision

The table presents the B2B service interfaces implemented during the in-Time project in Vienna and the Service domains associated to these services.

B2B Service	Service domains covered
Parking	Parking
Journey Planning	Journey Planning, Road traffic, Walking, Public Transport

Table 13 – B2B interfaces implemented in Vienna

Note: The parking service has been implemented, but is not available anymore at the time of writing.

5.6.2 Feedback interfaces

The table presents the B2B feedback service interfaces developed in Vienna and the related Service domains to which these services are associated.

Feedback Service	Service domains
Overall quality of service	All
Feedback on quality of provided data	Journey Planning, Public Transport, Traffic
New data	Journey Planning, Public Transport, Traffic

Table 14 – Feedback services interfaces developed in Vienna

5.6.3 Technological choices

The feedback services can be invoked by the TISP apps to provide feedback data via web services. These custom web services were implemented using the following software stack:

- Web Server: Tomcat 7.0 with Java 1.6
- Database: PostgreSQL 8.4.12
- Application Framework: Spring Framework 3.1
- Jax-WS Implementation: Apache Axis 2

The following integration steps were conducted to comply with the Co-Cities requirements and specifications:

1. Downloaded the latest Co-Cities Package from the repository
2. Generated the SOAP web services using a top-down approach from the Co-Cities WSDL and XSD files with wsimport (part of the JDK)
3. The resulting classes were implemented using Apache Axis 2 (Jax-WS)
4. Set up an PostgreSQL 8.4.12 database to store received feedback
5. Developed backend application in Spring 3.1
6. Hosted the web services and backend application on Tomcat 7.0
7. Provided the web service URLs to all TISPs

During implementation two bugs were found in the WSDL/XSD files, which were resolved quickly together with Softeco.

The feedback data is stored to a Feedback repository and visualized in a backend for further processing and dispatching via email.

Figure 27: Backend Application

6 Report on Development of mobile apps

6.1 Softeco

Softeco is one of the Co-Cities Traffic Information Service Providers (TISPs) offering mobile services to the end users. The technological solutions are based on Softeco's e-miXer[4], a software platform able to inter-connect different systems by using a set of protocols which can be different from case to case. In Co-Cities these protocols comes exactly from the CAI specification.

e-miXer Co-Cities is Softeco's Co-Cities mobile app running on Android and connected to the CAI through e-miXer of which it is a client. The app can be used in a variety of mobility scenarios like planning a trip from the current position to a specified address or POI with additional dynamic information on traffic, public transport or parking available along the route or at start/destination.

e-miXer Co-Cities has been specifically developed for the Project as a unique app for all Co-Cities sites thus fully applying the concept of a single access interface for all sites running the CAI.

A key aspect enabling the app working over an expansible set of sites is the *e-miXer Co-Cities Registry* which indicates exactly which Sites and which Services for each site are in place and must be visible within the app.

The *e-miXer Co-Cities Registry* is not intended to be a centralized element available to all Co-Cities sites. A centralized registry, in fact, can operate separately; nevertheless, a connection and synchronization between the two may be envisages and in general this would keep a consistency of the actual available offerings across the different TISP platforms[5].

[4] E-miXer is a flexible environment enabling integration, processing and delivery of Multimodal Real Time Traffic and Travel Information services which can be fully applied to In-Time and Co-Cities (more details are in section 6.1.4)

[5] This synchronization is not one of the activities foreseen by the Project DOW and has not been developed.

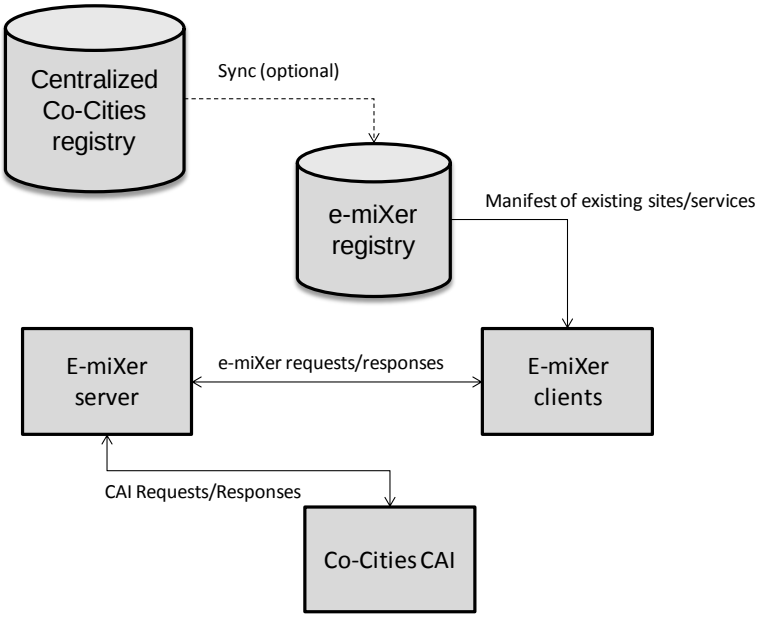

Figure 28 – basic e-miXer components

6.1.1 Features and functionalities of the mobile services

When the app is started the first time, users can select which city/site he/she wishes to receive services for (usually the location). This option can easily be changed later using the application settings menu, e.g. if the user would travel to another Co-Cities city.

From the main menu the current position can be shown and used to search all elements nearby (e.g. all stop points and parking places), otherwise a search for a specific domain can be operated by selecting the corresponding menu.

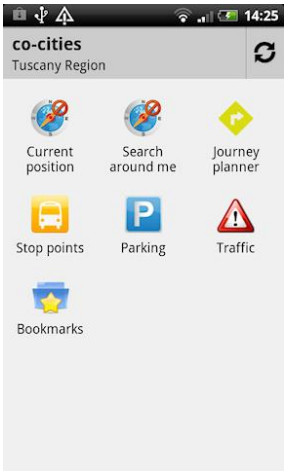

Figure 29 – Softeco application - Main menu for the selected Site

The domain menus are associated to the Co-Cities service categories. The app dynamically retrieves from the registry the information about the available services in the active site and shows them accordingly. In general the following domains can be present:

- Public transport
- Parking
- Traffic
- Journey Planning

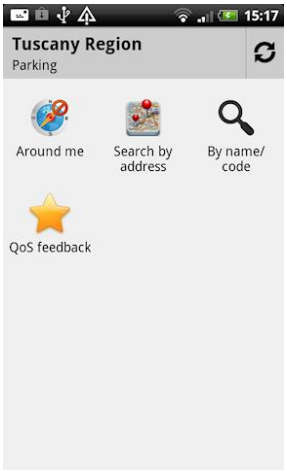

Figure 30 – Domain menu for parking

6.1.1.1 Location Based Services

Location Based Services allow searching information (Traffic, Parking and Public Transport) near a point (the current position or an address) or by giving the exact information about the element to be found.

The request of information is first sent to the e-miXer server and from here to the B2B Co-Cities services, which are running in each site. The response of the local CAI is sent back to the e-miXer server and from here to the app. The sequence of operations is the same as the Co-Cities value chain.

Once the search is done all elements found are shown as a list or on the map. Users can switch between the two views using the appropriate commands.

Detailed information is accessible by selecting a single element.

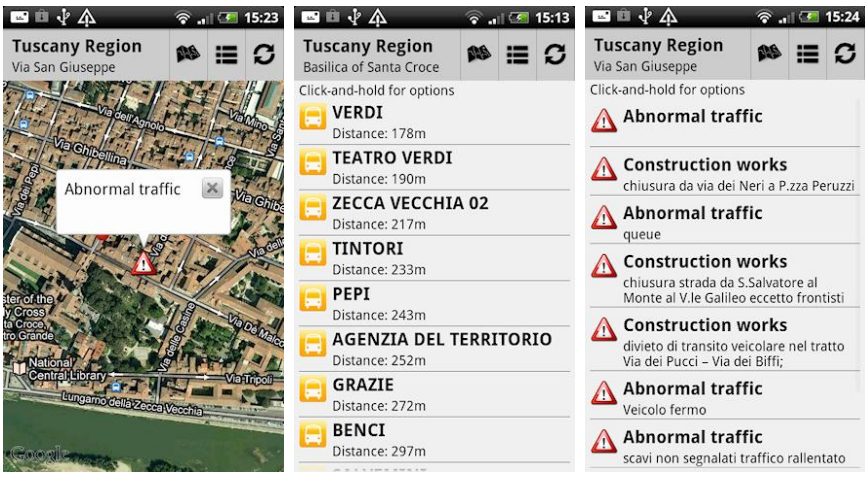

Figure 31 – Examples of response rendered in the map and as a list

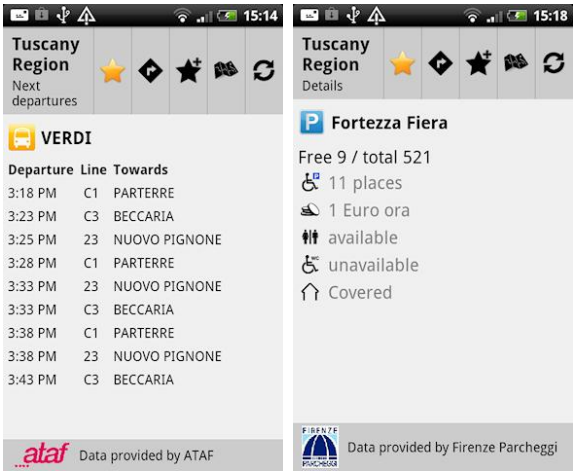

Figure 32 – Examples of details of a request

Several options are available for an element using the commands on the top toolbar. One option allows saving the element into a list of bookmarks. In general, bookmarks are pre-formatted searches, organized into domains and quickly accessible when desired.

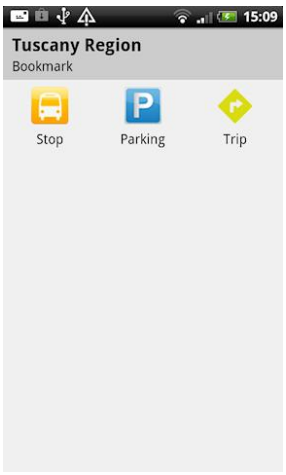

Figure 33 - Bookmarks

In case of elements of type PT stops or Parking it's possible to use them as a start or destination point of a journey to be planned.

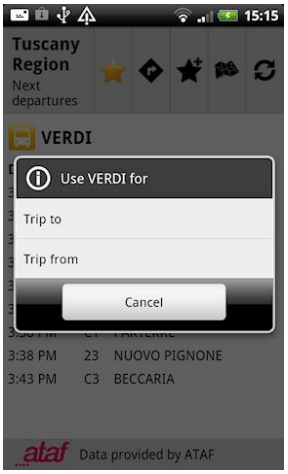

Figure 34 – Use of an element found to plan a Journey

6.1.1.2 Journey Planning

In order to plan a journey, users have to indicate a minimum set of data (an origin and a destination) and optionally, the departure/arrival date and time if these have to be different from the current ones. The next selection is about the preferred modes of transport selectable from the available ones.

Once these parameters are indicated the request is sent and a response is obtained with all trips suitable for the requested journey.

The selection of one trip returns its details with a step-by-step instruction list of the operations to be executed for each trip segment.

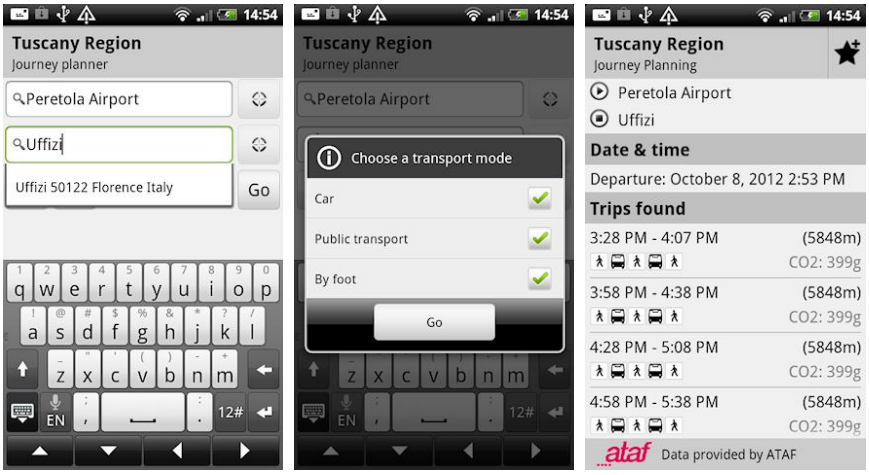

Figure 35 – Journey Planning- request and list of trips

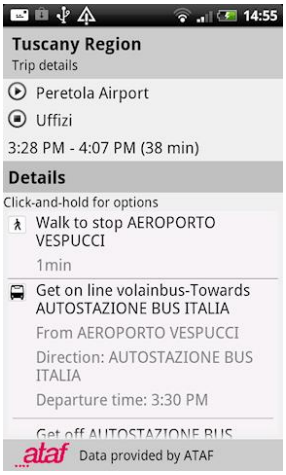

Figure 36 – Details of a trip

A trip found can be saved in the bookmark list for future reference. Further operations are available on the segments as it's possible to find POIs (Points Of Interest) nearby. These functions are available with a click-and-hold on the segment.

A single click on the segment opens the map with the selected segment highlighted.

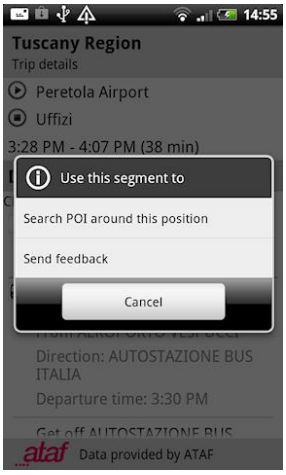

Figure 37 – options on a segment of a trip

6.1.1.3 Feedback functionalities

Feedback functionalities are available to provide full compliance with the Co-Cities requirements.

As indicated by the technical specification, users have the possibility to provide feedbacks as:

- Feedback on the overall quality of the service
- Feedback on the quality of provided data
- New data feedback

Each type of feedback functionality can be activated in a different way as indicated in the following.

Feedback on the overall quality of the service

The functionality is accessible with a shortcut on the sub-menu of the domain where this service is available like in the following figure.

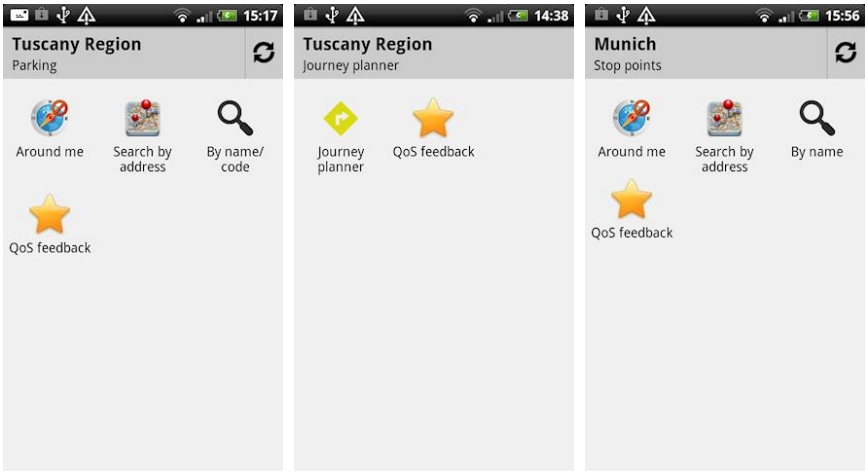

Figure 38 – Quality of service feedback shortcut for different domain menus

This opens a feedback panel with a free-text field and the 1 to 5 stars quality indicators associated to the service features to be evaluated as defined by the Co-Cities technical specification.

These values are common for each domain[6].

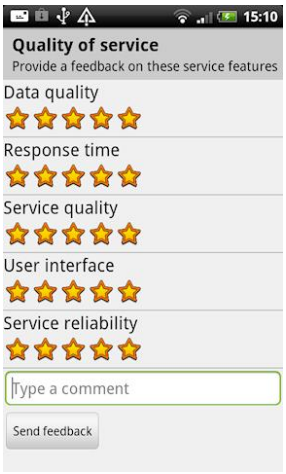

Figure 39 – Values for the quality of service

Feedback on the quality of provided data

The feedback functionality concerning the quality of the provided data is accessible with a shortcut on the top menu available from the screen where the detailed data is shown, like in the following figures.

[6] The quality values are common and one single QoS (Quality of Service) feedback service is defined in the specification. Nevertheless, by using the service parameters, it's possible to distinguish between the feedbacks associated to each service domain so that they can be sent and stored separately.

Figure 40 – Quality of Data feedback functionality available for detailed data

For the journey planning service the quality of data is related to each trip segment. In this case to make the functionality accessible for the segment it's necessary to click-and-hold the segment itself.

For the Quality of Data feedback a panel opens with a number of selectors, one for each relevant data item as defined by the Co-Cities technical specification. Users can switch the selectors to indicate a "good" or "bad" quality of the information received comparing it to the observed situation.

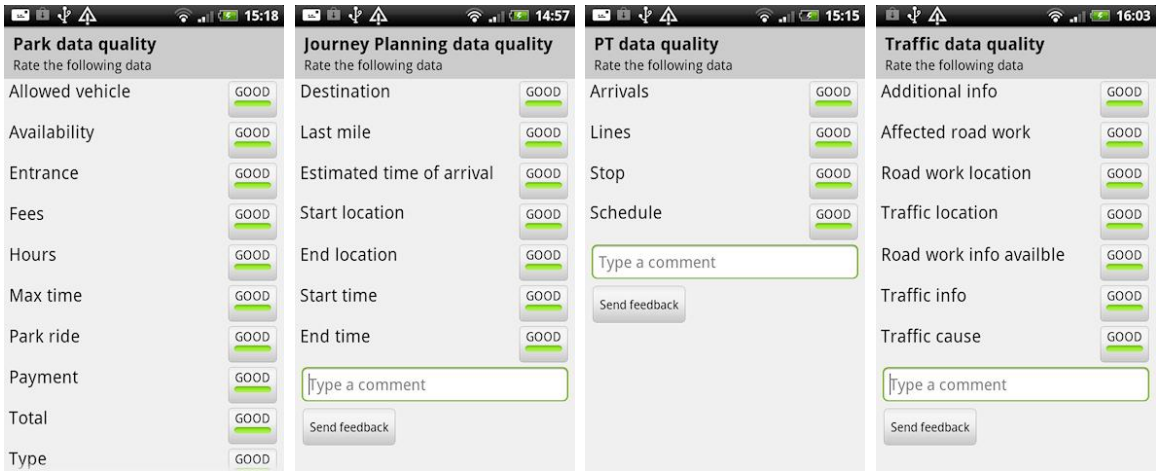

Figure 41 – Quality of Data feedback functionality for each service domain

Provision of New data

The New data feedback functionality is available for:

- New traffic event data

- New parking data
- Occupancy of a public transport vehicle

The functionality “New parking data” enable the provision of new features (or amendments to existing features) for an existing parking whose information can be retrieved by the user with the app. For this reason the access point to this functionality can be found in the context where the parking information is displayed: the page with the parking details (menu command) or the list of the parking found (click-and-hold over the selected item).

The functionalities “new traffic event data” and “occupancy of a public transport vehicle”, instead, refers to newly created data items not necessarily related to existing objects displayed into the application. For this reason the access point of these functionalities is the domain menu where a dedicated command can be found to add the new item.

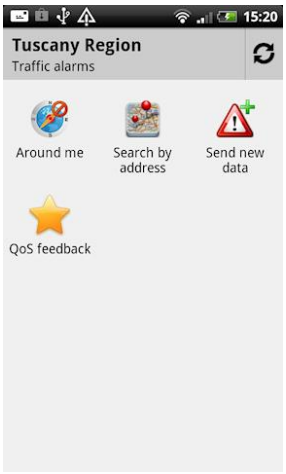

Figure 42 – Function to send a feedback as new road traffic data

Using the function “**New data on existing parking places”** the user can provide the following information:

- **Estimated available places**
- **N. of places for disabled people** (optional)
- **Fares** (optional)
- **Type of parking**
- **Free-text comment** (optional)

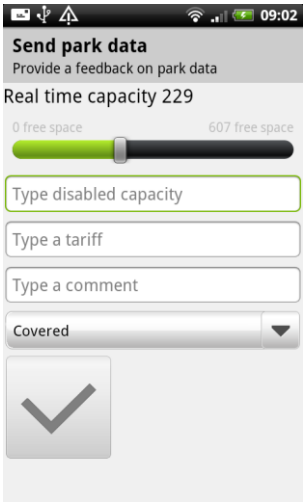

Figure 43 – Provision of new parking data

Using the function "**new traffic events"** the end user can provide the following information**:**

- **Type of event**
- **Location**
- **Date and Time** (Automatically retrieved)
- **Free text** for additional description, comments etc. (optional)

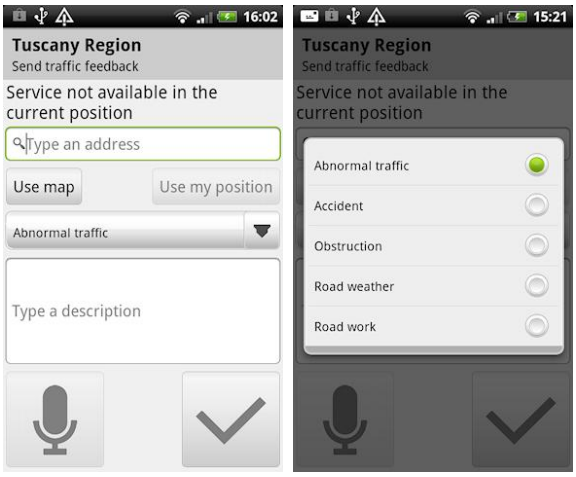

Figure 44 – New traffic data feedback

A speech-to-text function is available to easily provide the feedback information. In this case the current position is automatically retrieved by the system[7] and no free-text description can be

[7] This option is activated if the service is available in the current position. The GPS positioning system should be activated in the mobile device for a more precise identification of the position.

indicated. Once the feedback is generated, it is sent to the Co-Cities feedback services (Co-Cities CAI) and data is collected and eventually processed by each Site.

For **New data on estimated occupancy of a public transport vehicle** the user can provide the following information:

- **Estimated level of occupancy**
- **Position** (Automatically retrieved)
- **Date and time** (Automatically retrieved)
- **The number of the bus line**
- **The number of the bus** (optional)
- **Free-text comment** (optional)

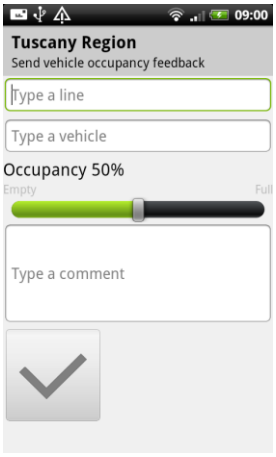

Figure 45 – Provision of new data on the estimated occupancy of a public transport vehicle

6.1.2 B2C service provision over available B2B services

The following table describes the available B2C mobile services over the available B2B services provided by the sites.

B2C Service	Service domains covered	B2B (CAI) service used
Public Transport information (LBS)	Public Transport	Public Transport WFS
Parking information (LBS)	Parking	Parking WFS
Traffic information (LBS)	Traffic	Traffic WFS
Journey Planning (including Walking, Cycling, Car and Public Transport modes)	Journey Planning, Road traffic, Cycling, Walking, Public Transport	Multimodal Journey Planning web service

Table 15 – B2C services provided by Softeco application

6.1.3 Feedback service provision using available B2B feedback service interfaces

The following table describes the Feedback functionalities available from the mobile app and the related B2B feedback service interfaces available and used from the cities.

Feedback Functionality	Service domains covered	B2B (CAI) service used
Overall quality of service	All domains	pushQualityFeedback(ServiceFeedback)
Feedback on data quality	All domains	pushQualityFeedback(ParkingQualityFeedback) pushQualityFeedback(TrafficQualityFeedback) pushQualityFeedback(PublicTransportQualityFeedback) pushQualityFeedback(JourneyPlanningQualityFeedback)
New data	Traffic PublicTransport Parking	pushDataFeedback(TrafficFeedback) pushDataFeedback(PublicTransportFeedback) pushDataFeedback(ParkingFeedback)

Table 16 – feedback functionalities provided by the Softeco application

6.1.4 Technological choices

The Co-Cities mobile application is developed as a client of the software platform e-miXer developed by Softeco.

e-miXer is a set of software components enabling integration, processing and delivery of Multimodal Real Time Traffic and Travel Information services. It operates in a Service Oriented Architecture environment and allows the full integration of data/services from external mobility and traffic sources (e.g. Local Authorities, mobility operators, etc.) while providing enhanced, value-added B2C and B2B services. Typical applications of e-miXer include, for instance: traffic monitoring and supervision, user information through various delivery channels (e.g. SMS, email, mobile apps, interactive voice response, etc.), infomobility web portals, etc.

e-miXer can operate as a middleware hosted remotely and connected to local systems or it can be installed locally in the technical infrastructure of a customer or organization.

e-miXer can interconnect and combine data from different systems through custom adaptation components (adapters) and make them accessible for Traffic Information Service Providers (TISPs) via a number of B2B interfaces. The platform, whose design is evidently inspired from the In-Time/Co-Cities principles, can flexibly be configured either as a producer and a consumer of the CAI B2B services by means of an appropriate configuration of the Adapters and the B2B interfaces.

A more detailed view of the e-miXer architecture is shown on the following figure.

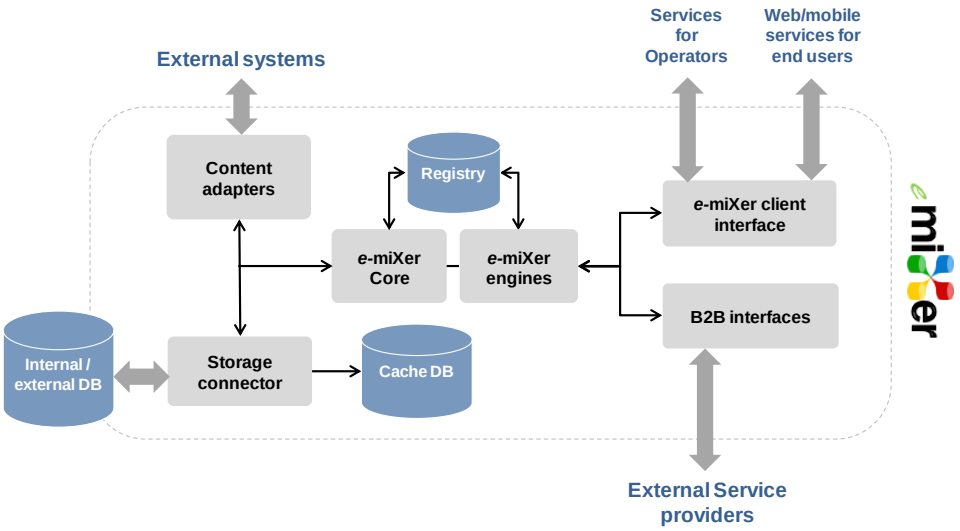

Figure 46 - e-miXer Architecture

The e-miXer core drives the process of data acquisition based on parameters hosted by the registry. The storage connector enables a temporary caching or permanent storage of the information. The e-miXer engines processes data for specific purposes whenever necessary (e.g. routing calculation, traffic data processing etc.).

The e-miXer client interfaces finally enables the provision of services for the end users (both public and restricted access) while the B2B interfaces are available for external service providers like the TISPs.

For each stage of the service chain along which it operates, the appropriate combination of modules is dynamically activated in order to achieve a specific functionality from a connection to a data source to content delivery through a specific channel. New e-miXer modules can be developed on the basis of emerging needs and new features can be added to existing systems by integrating the appropriate additional modules.

The e-miXer environment is set up and configured to be connected with the existing CAI interfaces and to work in combination with the mobile application.

The sequence of Requests and Responses is achieved as described for the Co-Cities delivery chain (chapter 4): the request for a specific site is first sent from the mobile app to the e-miXer server (through the B2B interfaces) and from here to the appropriate Co-Cities CAI (using the e-miXer adapters) whose address is stored in the registry. The response is sent from the local CAI to the e-miXer server (again through the e-miXer adapters) and from here to the mobile app, through the B2B interface. The behavior is similar for feedback information.

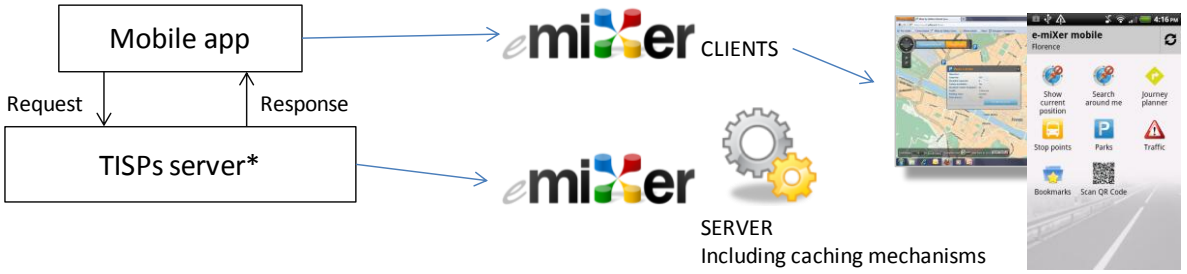

Figure 47 – Softeco's e-miXer Client-Server environment

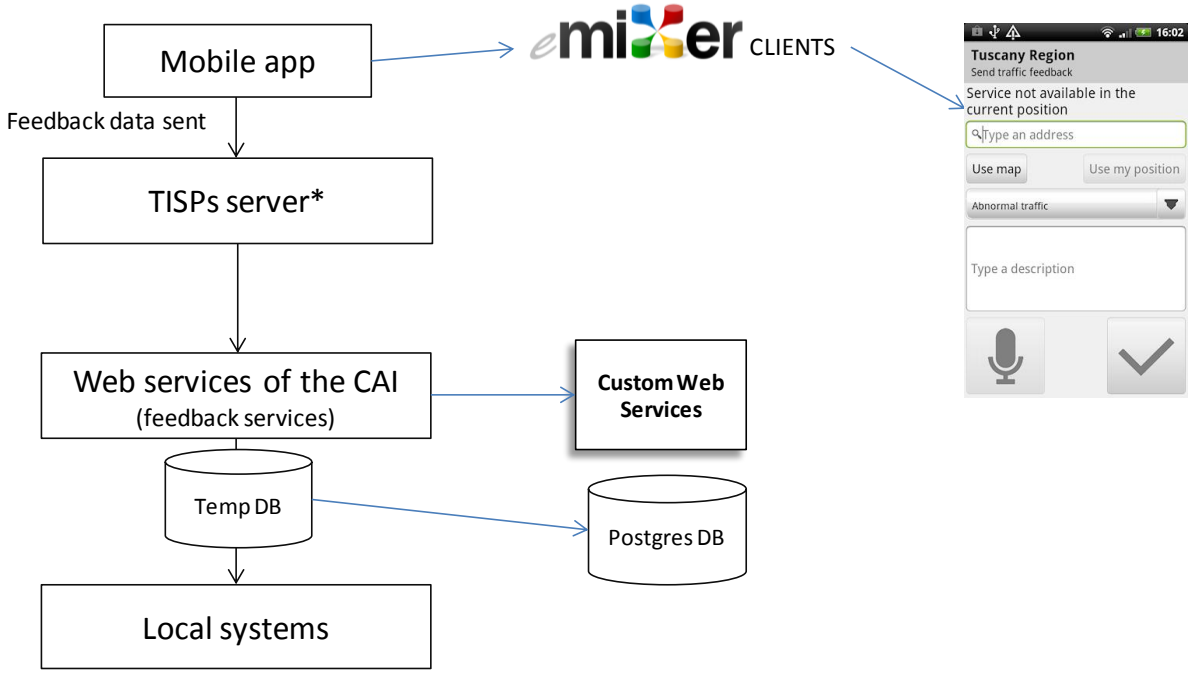

Figure 48 – Components for feedback management

6.1.5 Supported sites

The following table describes how the Softeco application supports the Co-Cities Sites

Function or Service	Site
LBS (Public transport, Parking, Traffic information)	All sites where the necessary B2B service is available
Journey Planning Service	All sites where the necessary B2B service is available
Feedback Services (overall Quality of service)	All sites where the necessary B2B service is available

Function or Service	Site
Feedback Services (Quality of data)	All sites where the necessary B2B service is available
Feedback Services (new data - Traffic)	All sites where the necessary B2B service is available

Table 17 – tables supported by Softeco application

6.1.6 Technical Specification

Name of the application: e-miXer Co-Cities

Platform: Android v2.2 and above

Downloadable from: Google Play Store

Size of package: 1,3 MB

URL: https://play.google.com/store/apps/details?id=it.softeco.freetomove.cocities

6.2 Telematix

Telematix is one of the Co-Cities Traffic Information Service Providers (TISPs) offering mobile services to the end users. Telematix has concentrated its effort to demonstrate the functionality in Prague and Reading, however the other cities are supported as well. The application is developed for the iOS platform and its main functionality is to provide multi modal journey planning services to the end users.

6.2.1 Features of the mobile services

The application offers journey planning services for Prague, Reading, Vienna and Brno. Users can use this application to provide feedbacks to the cities of Prague, Reading and Vienna. These feedbacks include the overall quality of service, public transport vehicle occupancy and journey planning quality.

The application uses the native user interface controls of iOS. Any user that is familiar with iOS should not have any difficulties navigating the user interface. The application is logically separated into four parts - Connection, Favorites, Recent and Settings. When the application starts the connection screen appears.

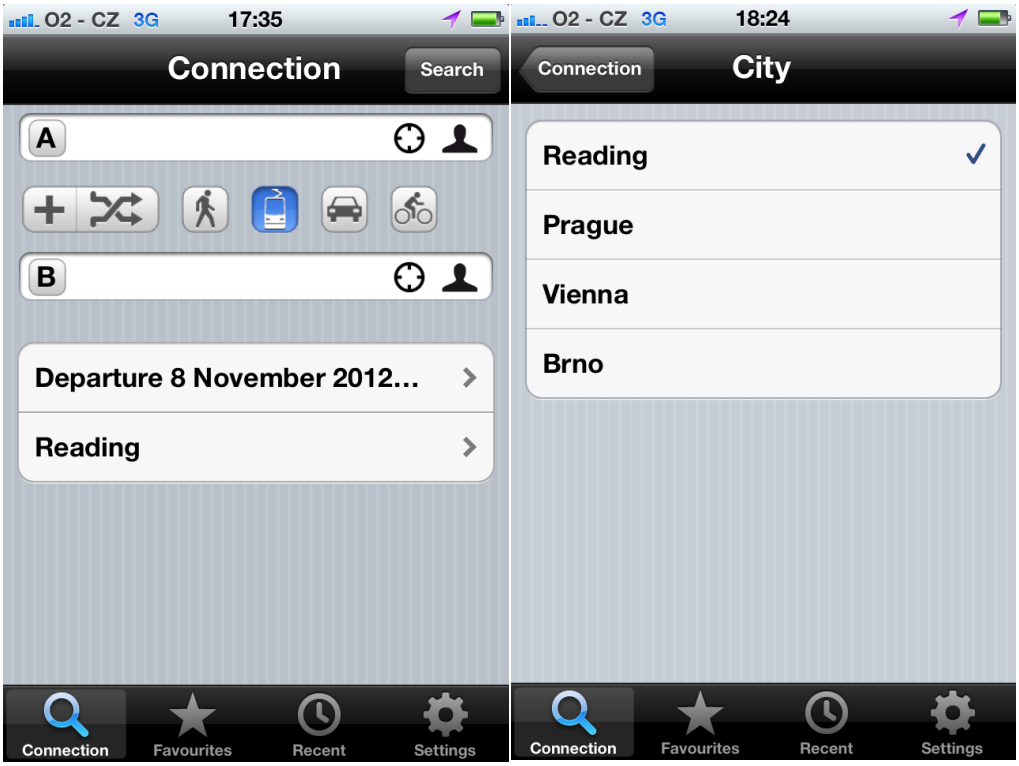

Figure 49 – Connection screen and city selection in Dynavix Multi

The main purpose of the connection screen is to plan a multi modal journey. From this screen it is possible to navigate to a screen where the user can operate a change between sites. The application supports adding intermediate waypoints along the planned route. Individual waypoints can be selected by entering an address, choosing a stop point or choosing a point in map. After pressing a search button the user is presented with optional routes. Each route contains navigation instructions and the result can be displayed in the map. Additionally, favorite routes can be saved for later use. The history screen contains previously requested routes.

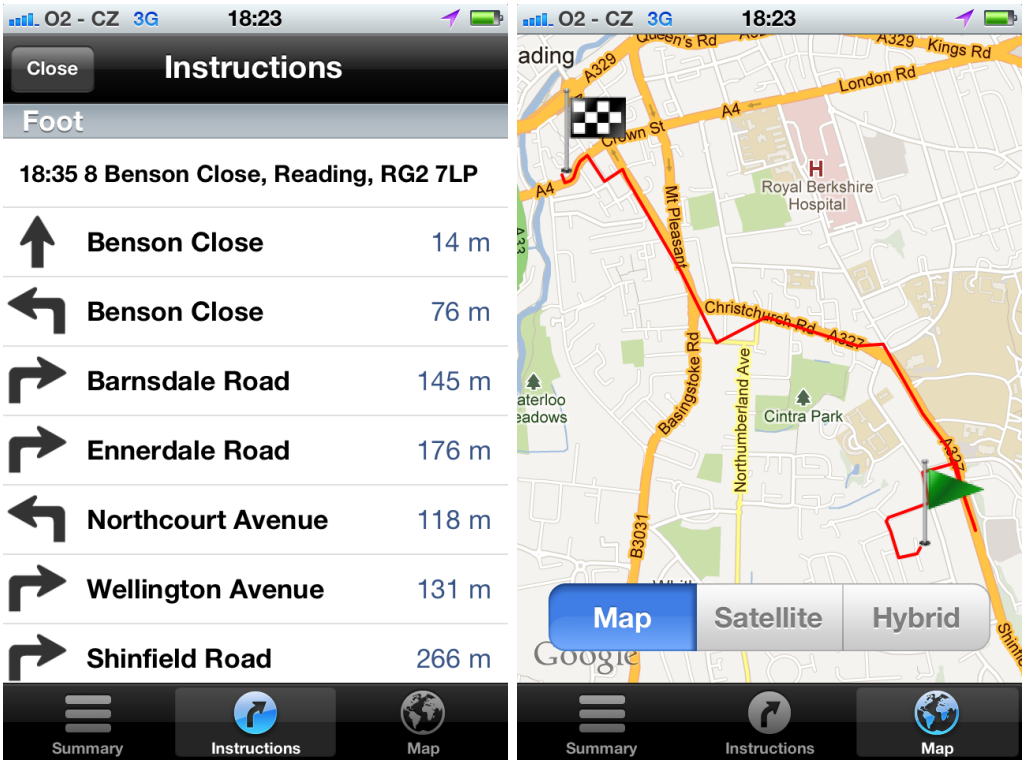

Figure 50 – Navigation instructions and route displayed on the map in Dynavix Multi

There are three types of user feedbacks that can be sent through the CAI from this application. Public transport occupancy and journey planning quality feedbacks are accessible from the instructions screen for each part of a public transport journey. The feedback for overall quality of service is located in the settings screen.

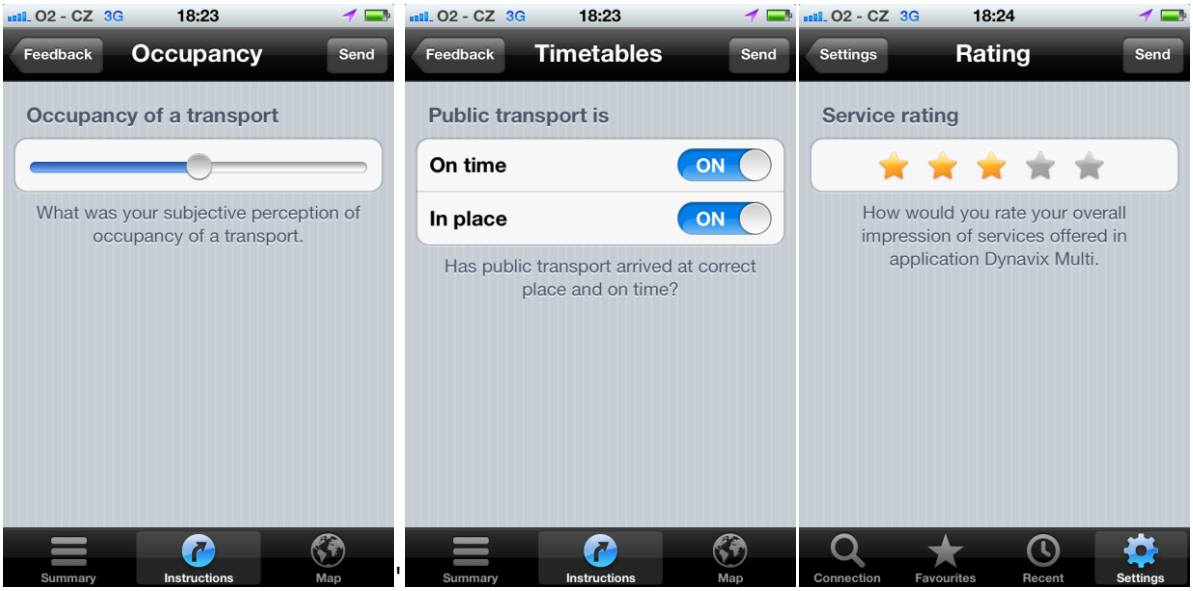

Figure 51 – Screens for sending feedbacks in Dynavix Multi

There are 1 to 5 stars quality indicators to evaluate the services.

6.2.2 B2C service provision over available B2B services

The following table describes the available B2C mobile services over the available B2B services provided by the Cities.

B2C Service	Service domains covered	B2B (CAI) service used
Public Transport information	Public Transport	Public Transport WFS
Journey Planning (including Walking, Cycling, Car and Public Transport modes)	Journey Planning, Road traffic, Cycling, Walking, Public Transport	Multimodal Journey Planning web service

Table 18 – B2C services provided by Telematix application

6.2.3 Feedback service provision using available B2B feedback service interfaces

The following table describes the Feedback functionalities available from the mobile app and the related B2B feedback service interfaces available and used from the cities.

Feedback Functionality	Service domains covered	B2B (CAI) service used
Overall quality of service	Public Transport, Journey Planning, Cycling, Walking	pushQualityFeedback(ServiceFeedback)
Feedback on data quality	All domains	pushQualityFeedback(JourneyPlanningQualityFeedback)
New data	Traffic	pushDataFeedback(PublicTransportFeedback)

Table 19 – feedback functionalities provided by the Telematix application

6.2.4 Technological choices

The Dynavix Multi mobile application is developed for iOS. The application was created in C# targeting the Monotouch framework. This framework supports compilations ahead of time and therefore eliminates Apple's virtual machine restrictions by compiling C# code into native code. Furthermore, the application uses the native iOS user interface controls and online maps to display the selected journey.

The application operates in a service oriented architecture. Telematix has created a custom service for querying public transport stop points. This service is connected to other RDSS servers. It preloads stop points to an internal cache. The mobile application communicates with this service in proprietary format which is based on the SOAP. The journey planning requests are sent directly from mobile application to RDSS servers, since there was no need to do additional processing on the TISP server. Feedbacks are also sent directly to the proper feedback services based on the selected city. By selecting Monotouch technology it was possible to reuse generated code from the wsdl.exe utility to communicate with RDSS services.

6.2.5 Supported sites

The application integrates the following sites: Reading, Prague, Tuscany Region and Vienna. The city of Brno is also available without the ability to send user feedbacks as Brno services were implemented in the In-Time project. This clearly demonstrates the backwards compatibility of the Co-Cities CAI.

6.2.6 Technical Specifications

Name of the application: Dynavix Multi

Platform: iOS 4.3 and above

Downloadable from: iTunes Store

URL: https://itunes.apple.com/app/dynavix-multi-reading/id511284243?mt=8.

7 Methodology for Technical validation

7.1 TISP validation

The TISP feedback service is responsible for collecting feedbacks from end users. Data collection process can be implemented proprietarily as long as data is sent to the RDSS through the CAI feedback extensions. For example, TISPs can implement a custom web service for collecting feedback information from end user mobile phones. Once the TISP feedback service receives a feedback it can be preprocessed and stored in a local database and then send to RDSS in predefined time intervals.

In order to allow the collection of user feedback related to individual Co-Cities services, a matching between the respective use cases consumed and the feedback questions displayed is needed in the End-User service application. General requirements for clients and devices are discussed in *D3.2-Interfaces and mechanisms for end user involvement* [5].

The technical validation of the functionalities of the TISP apps is carried out with the following sequence of operations:

1. A number of checklists (found in Annex of Internal Report IR3.1 [6]) is defined to identify the functionality to be validated:
    - A checklist is defined for each service domain.
    - A checklist is divided in four parts. One for the validation of the In-Time-related functions and three for the validation of the functionalities related to the three types of Feedback. In details the 4 checklists are:
        i. Checklist for the validation of the functionalities related to the information provision for the end users. This is the validation of the In-Time services and for this reason the validation schema is derived from the In-Time one by taking into account the differences in terms of classification of the B2C services between Co-Cities and In-Time
        ii. Checklist for the validation of the end user functionalities necessary for the provision of a feedback about the General Quality of the service
        iii. Checklist for the validation of the end user functionalities necessary for the provision of a feedback about the Quality of data
        iv. Checklist for the validation of the functionalities necessary for the provision of new data as a feedback
2. The presence of each feature defined in the checklists must be checked by using the mobile app developed by the TISPs, in each City participating in the project.
3. By inspecting the behavior and output of the app, each entry of the checklist must be marked with the appropriate value as described in the checklist itself (i.e. test passed, not passed or not applicable).

7.2 Validation of Communication between TISP and RDSS System

The Reference Platform is a tool for testing and validating the Co-Cities services' complete delivery chain at the Commonly Agreed Interface (CAI) in each city. It simplifies error analyses, as it allows quickly assessing which data and in which quality was available at the interface. Furthermore comparability between cities and services is ensured.

Project partners, Traffic Information Service Providers (TISP) and Local Data Providers (RDSS) can use it to monitor the interface, run automated tests and access reports/statistics depending on their permissions.

Based on Co-Cities' architecture, lessons from In-Time, the validation strategy [3] and several discussions with project partners the following requirements for the Reference Platform were identified:

- Relay all traffic between the TISPs and the RDSS while affecting the performance of the service delivery chain as little as possible
- Log incoming and outgoing messages in a standardized way
- Aggregate and prepare these messages for further analysis and to get an up to date status
- Support a hierarchy of different user roles (Administrator/User) with different permissions (e.g. read only) on each level
- Offer monitoring and analysis capabilities in a Web-GUI to users, depending on their permissions
- Login with Username/Password
- Allow mobile access in the field

The Reference Platform Validation on the CAI level is conducted by the responsible partner/TISP by using the reports produced by the Reference Platform itself.

The configuration and set up of the Reference Platform is detailed in *D4.2-Reference platform, its configuration and usage manual* [8]. The following methodology is defined to test the full service delivery chain:

- Setup
    - The TISP has to set up the feedback URLs to invoke the feedback services via the Reference Platform
    - The RDSS has to add to a "white list" the Reference Platform's IP address in case there are any firewall restrictions
- Use the services from the mobile apps (or invoke the related web services) multiple times
- Check individual requests/responses (optional) and feedbacks/acknowledgments in the first test run and, in case of any unexpected behaviour, assess whether the problem is located on side of the TISP or the RDSS

- Check the statistics of the Reference Platform to get information about errors, response times etc.
- The logs of the Reference Platform are used to check the correct behaviour of the web services of the CAI installed at each site: using the checklist for the validation of the communication between TISP and RDSS (defined in the internal report IR3.1) the execution of each Business Case must result in a **positive** output from the Reference Platform logs.
- For the current test the execution of a business case can be identified on hands of the execution of a web service. A positive output can be identified in the log reports of the Reference Platform with the corresponding entry having a "**Success**" status and marked in Green. "Success" denotes the successful call of a web service. There may be cases where this is not enough to ensure that the expected results occur (e.g. no results for a request). Therefore, while a "Success" entry can reveal the presence/absence of the basic communication link between the TISP and the RDSS, an additional check must be performed on the integrity of data to test the complete functionality of the CAI.
- An entry in the log files has the following format:

    - Business case: name of the city and method of the web service
    - Started at
    - Duration
    - Status

An example of entries can be found in the following picture:

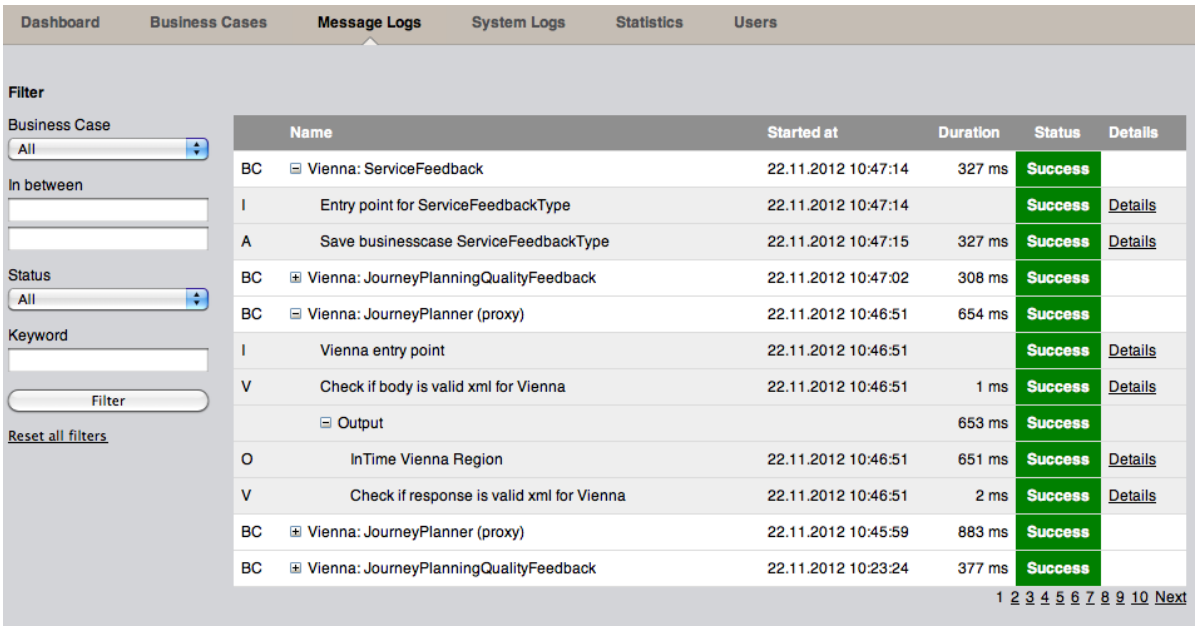

Figure 52 – example of report from the Reference Platform

7.3 Validation of RDSS Systems

The validation of RDSS systems according to a common scheme is not possible, as the RDSS have very different organisational and technical backgrounds. Hence only a very general procedure can be proposed to any follower cities which basically comprise the requirement, that **the feedback provided via the CAI is to be received correctly by the dedicated processes in a applicable amount of time**.

Due to the unique design of each RDSS system, **a common validation methodology** for "the RDSS system" is not defined and only high level guidelines should be considered.

The specific methodologies, tools used and key performance indicators relevant for the validation will differ from system to system depending on the specific design of the software implemented on side of the RDSS. This means, that every RDSS shall describe the validation methodology he applied to his specific system concept.

7.3.1 General Requirements

In principle, the RDSS components of the Co-Cities system should be validated against the requirements placed by the RDSS for those system parts.

Those requirements could contain:

- Correctness of data: the feedback received by the "final recipient process" in the RDSS system, e.g. a data base, must comply with the feedback generated by the end user. In case of processing of the feedback on side of the TISP, the correctness of data is validated against the specification of the aggregated/processed feedback while in case of direct reception of each single end user feedback, the correctness of data is validated against the single end user feedback. Example: if an end user is indicating, that the opening hours of a parking lot are not correct and should be corrected, this information must also be contained in the data set contained in the RDSS system concerning this very end user feedback.
- Latency: as the quality of an information service heavily depends on the actuality of the information, the end user feedback should be processed within a well-defined time frame. Within the validation of the RDSS system components, only the latency resulting from those processes is considered.
- Correct georeferencing: the feedback in Co-Cities contains, where applicable, location data. This location must be correctly represented in the RDSS system and needs to be validated against the location submitted with the end-user feedback.
- Correctness of feedback allocation: if a feedback is related to a data item originating from the related In-Time service.

Of course, other aspects could be contained in the specification of the RDSS components and should be considered in the validation.

The methodology for validation might differ depending on the RDSS system but should be designed to support a sound validation methodology. In other words, the validation should foresee a

sufficient number of tests to safeguard a sufficiently high significance. The validation can comprise manual or automatic validation means, e.g. the automatic or manual generation of test messages in sufficient quantity.

7.3.2 Conditions of the test

The tests are executed and the results are considered valid under the following conditions, applicable to every domain and type of feedback service in all test sites:

- The mobile app or, in general, the client responsible for providing the feedback is working correctly and is fully Co-Cities compliant[8].
- Results based on failures due to external causes like absence of connectivity are discarded and do not invalidate the result.
- Each test is repeated several times (at least 20) and at different times/dates.

7.3.3 Site specific validation

The following chapters hold the detailed description of the site specific validation methodologies and tools which shall be followed within the validation of the RDSS systems.

7.3.3.1 Bilbao

The RDSS validation in Bilbao is executed with the support of a software tool developed by Softeco providing facilities in relation to the received feedback information. A separate instance of this tool is created for Bilbao and the same validation procedure as for the Tuscany Region is considered. The supporting tool and the related validation methodology are described in section 7.3.3.5.

It can be noted that the possibility to use the same supporting tool in different Sites is another proof of interoperability, ensured by the common approach of Co-Cities.

The following sections describe how data are collected and how the validation of the correct reception of the feedback is conducted.

7.3.3.1.1 Quality of service feedback

The methodology for the validation of correct receipt of Quality of Service feedbacks in Bilbao is composed by the following operations:

1. A number of quality of service feedback data is sent to the respective service interface in Bilbao at different times and the results are shown on the e-miXer Co-Cities console.
2. Data is sent from different Co-Cities Clients and mobile apps.
3. The test is repeated several times (at least 20).

[8] The validation of the communication between TISP and the RDSS is carried out by a specific type of test. See 7.2

4. The test is successful if all data is correctly received and shown in the related e-miXer Co-Cities console panel.

The following table structure will be used to report on the results:

Period of testing	elapsed time or duration of testing
Number of test sessions	Numeric value
Number of feedbacks correctly received	Numeric value
% of success	Numeric value as a percentage

Table 20 – table for RDSS validation of Quality of Service feedback in Bilbao

7.3.3.1.2 Quality of data feedback

The methodology for the validation of correct receipt of Quality of Data feedbacks in Bilbao is composed by the following operations:

1. A number of quality of data feedback data is sent to the respective service interface in Bilbao at different times and the results are shown on the e-miXer Co-Cities console.
2. Data is sent from different Co-Cities Clients and mobile apps.
3. The test is repeated several times (at least 20).
4. The test is successful if all data is correctly received and shown in the related e-miXer Co-Cities console panel.

The following table structure will be used to report on the results:

Period of testing	elapsed time or duration of testing
Number of test sessions	Numeric value
Number of feedbacks correctly received	Numeric value
% of success	Numeric value as a percentage

Table 21 – table for RDSS validation of Quality of data feedback in Bilbao

7.3.3.1.3 New data feedback

The methodology for the validation of correct receipt of new data feedbacks in Bilbao is composed by the following operations:

1. A number of new data feedback information is sent to the respective service interface in Bilbao at different times and the results are shown on the e-miXer Co-Cities console.
2. Data is sent from different Co-Cities Clients and mobile apps.
3. The test is repeated several times (at least 20).

4. The test is successful if all data is correctly received and shown in the related e-miXer Co-Cities console panel

The following table structure will be used to report on the results:

Period of testing	elapsed time or duration of testing
Number of test sessions	Numeric value
Number of feedbacks correctly received	Numeric value
% of success	Numeric value as a percentage

Table 22 – table for RDSS validation of New data feedback in Bilbao

7.3.3.2 Munich

In Munich, the feedback service for parking data and traffic data was validated successfully.

The tools used for the validation were the visualization tool developed within the project (see figure below) and the mobile application of Softeco for entering the feedback.

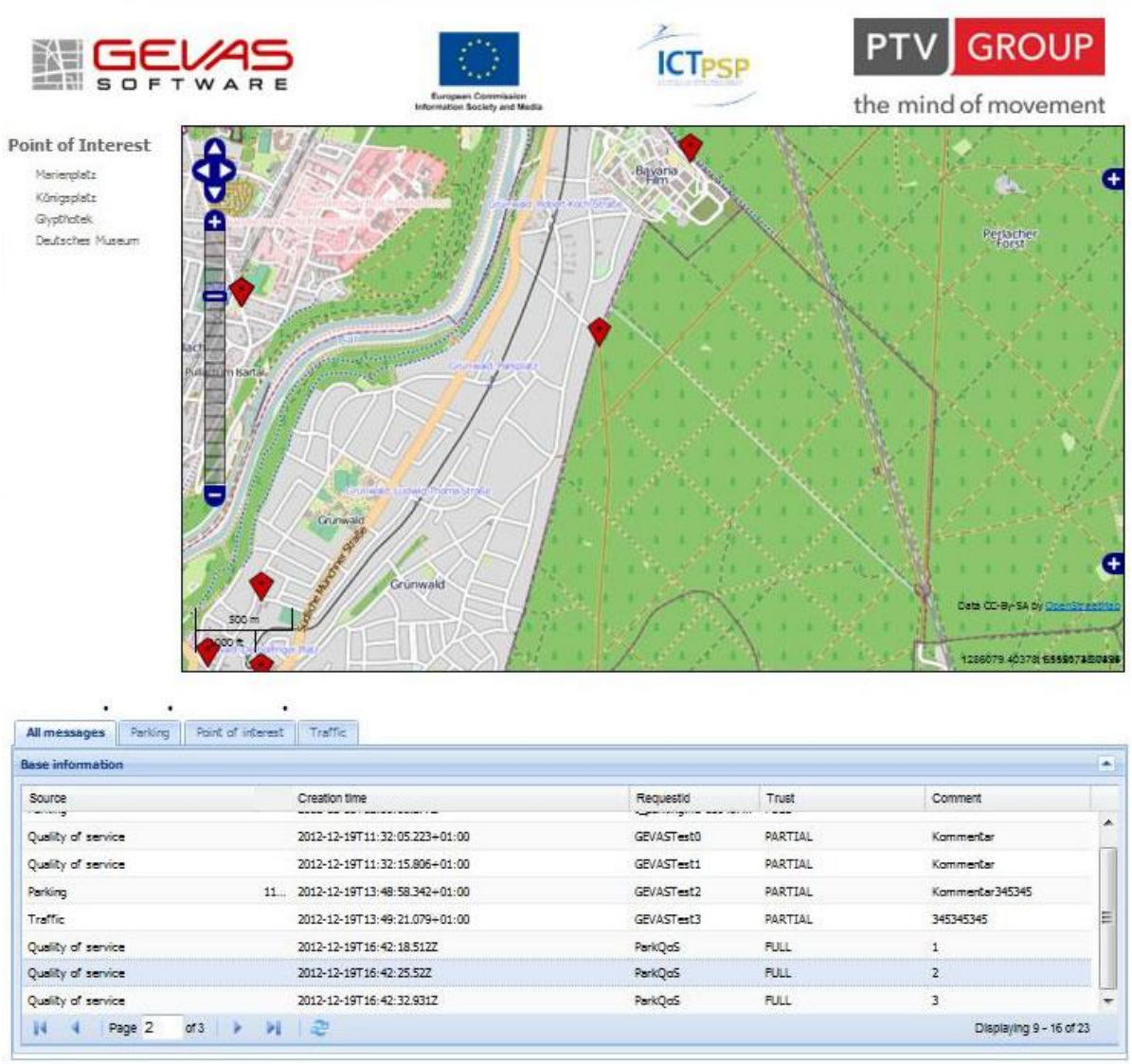

Figure 53 – Visualization of Feedback in Munich

Despite the fact, that the use of the mobile application is – strictly speaking – constituting an integration test (as all Co-Cities relevant actors are involved, the TISP, CAI and RDSS), it was defined as test setting with subsequent breaking down into potentially malfunctioning system components for error tracking.

In principle, feedback messages utilizing, in different combinations, all available feedback options including the free text field were sent to the RDSS system. 75 messages were generated as quality of data feedback for seven different parking lots and ten messages were generated for the overall quality of service feedback (5 each for the traffic and parking domain).

The detailed results are provided below. The results of the tests are displayed below as an overview. As all messages were received correctly the validation was concluded successfully and no further bug-tracking (concerning e.g. the origin of a bug – e.g. CAI/RDSS/TISP) had to be commenced.

7.3.3.2.1 Quality of service feedback

The validation in Munich concerning the quality of service feedback was executed as follows:

1. Feedback messages are generated five times for the traffic and parking domain covering all possible adjustments for each five-star rating system (but not all possible combinations).
2. The test is successful if all data is correctly received and displayed in the visualisation tool.
3. The following table structure will be used to report on the results:

Period of testing	elapsed time or duration of testing
Number of test sessions	Numeric value
Number of feedbacks correctly received	Numeric value
% of success	Numeric value as a percentage

Table 23 – Table for RDSS validation of Quality of Service feedback in Munich

Each feedback message also comprised the utilisation of the free text field which was found to work perfectly as well.

7.3.3.2.2 Quality of data feedback

The validation in Munich concerning the quality of data feedback was executed as follows:

1. A number of P&R places were selected as trial spots. These were:
   Neuperlach Süd
   Westfriedhof
   Heimeranplatz
   Gmunder Straße
   Maffeistraße
   Harthaus
   Schneiderhofstrasse 1 (there are two P&R locations with the same name in close vicinity, one of those was selected for testing)
2. Feedback is generated several times (min. 8) for each P&R location
3. The test is successful if all data is correctly received and displayed in the visualisation tool.

The following table summarizes the results:

Period of testing	Value to be specified after the test
Number of test sessions	Value to be specified after the test
Number of feedbacks correctly received	Value to be specified after the test
% of success	Value to be specified after the test

Table 24 – Table for RDSS validation of Quality of Service feedback in Munich

The feedback messages generated implemented different combinations of feedback options (e.g. allowed vehicles & Fees, all options etc.) and partly comprising a free text feedback in order to test the correct functionality of this feature.

The following figure below shows the excerpt of a test sheet on example of the P&R facility *Heimeranplatz*.

Utilising Mobile Application (provided by SOFTECO)

P&R Name	Timestamp (1)	Allowed vehicles	Availability	Entrance	Fees	Hours	Max Time	Park&Ride Status	Payment	Total Number	Type	Comment
Heimeranplatz	17.12.2012											
Feedback sent	15:44	x	x								x	testcommenthp1
Feedback received	15:44	x	x								x	testcommenthp1
Feedback sent	15:47		x	x					x	x		testcommenthp2
Feedback received	15:47		x	x					x	x		testcommenthp2
Feedback sent	15:49				x				x		x	testcommenthp4
Feedback received	15:49				x				x		x	testcommenthp4
Feedback sent	15:52				x	x		x	x			testcommenthp5
Feedback received	15:52				x	x		x	x			testcommenthp5
Feedback sent	15:58	x	x						x	x		
Feedback received	15:58	x	x						x	x		
Feedback sent	16:14					x	x	x				testcommenthp3
Feedback received	16:14					x	x	x				testcommenthp3
Feedback sent	16:16				x	x		x				
Feedback received	16:16				x	x		x				
Feedback sent	16:20	x				x	x					
Feedback received	16:20	x				x	x					
Feedback sent	16:22								x	x	x	
Feedback received	16:22								x	x	x	
Feedback sent	16:23						x	x	x			
Feedback received	16:23						x	x	x			
Feedback sent	16:25						x	x	x			
Feedback received	16:25						x	x	x			
Feedback sent												
Feedback received												

Utilising Webfrontend Feedback Application (provided by GEVAS)

(1): timestamp of backend is GMT, thus +1 h for CET

Figure 54 – Test sheet for quality of data for P&R Heimeranplatz

All feedback messages were received within (mostly significantly below) the time frame of one minute counted from the submission on side of the end user device to the reception on side of the visualisation tool. These results satisfied the requirements towards the RDSS feedback management system for actuality and correctness of data.

7.3.3.3 Prague

Telematix, responsible for the technical activities in Prague, has used existing (developed by third parties) and custom tools for validating the RDSS side. One of the custom tools Telematix has developed, named Feedbacks Viewer, has been used for visualization of received feedbacks.

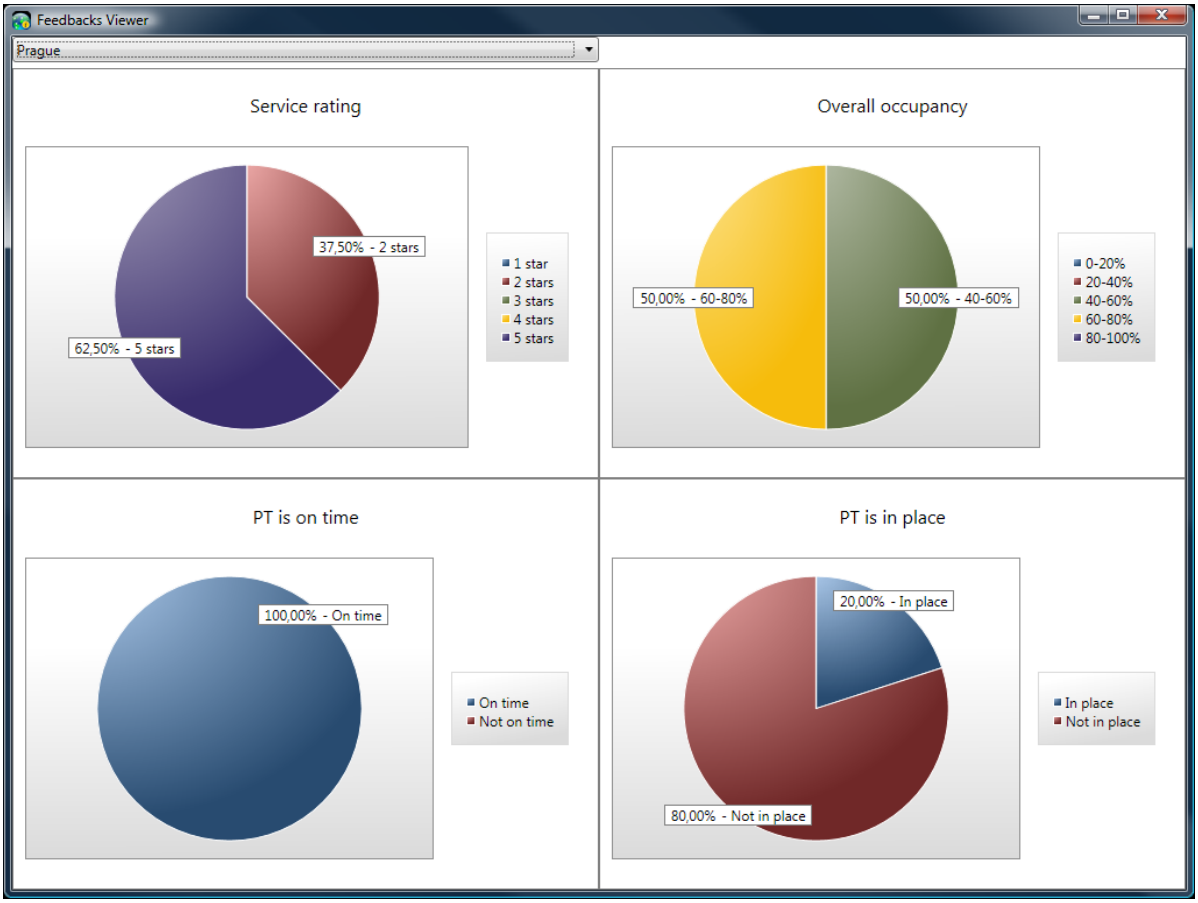

Figure 55 – Telematix Feedbacks Viewer showing Prague feedbacks.

7.3.3.3.1 Quality of service feedback

The methodology for the validation of the correct reception of Quality of Service feedbacks in Prague comprises the following steps and related operations:

1. A number of quality of service feedback data is sent to the respective service interface in Prague at different times and the results are shown using the Feedbacks Viewer tool and database visualization tools.
2. The feedback data is sent from different Co-Cities Clients and test apps.
3. The test is repeated several times (at least 20).
4. The test is successful if all data is correctly received and shown in the backend.

The following table structure will be used to report on the results:

Period of testing	elapsed time or duration of testing
Number of test sessions	Numeric value
Number of feedbacks correctly received	Numeric value
% of success	Numeric value as a percentage

Table 25 – table for RDSS validation of Quality of Service feedback in Prague

7.3.3.3.2 Quality of data feedback

The methodology for the validation of the correct reception of Quality of Data feedbacks in Prague comprises the following steps and related operations:

1. A number of quality of data feedback data is sent to the respective service interface in Prague in different times and the results are shown using the Feedbacks Viewer tool and database visualization tools.
2. The feedback data is sent from different Co-Cities Clients and test apps.
3. The test is repeated several times (at least 20).
4. The test is successful if all data is correctly received and shown in the backend.

The following table structure will be used to report on the results:

Period of testing	elapsed time or duration of testing
Number of test sessions	Numeric value
Number of feedbacks correctly received	Numeric value
% of success	Numeric value as a percentage

Table 26 – table for RDSS validation of Quality of Service feedback in Prague

7.3.3.3.3 New data feedback

The methodology for the validation of the correct reception of new data feedbacks in Prague comprises the following steps and related operations:

1. A number of new data feedback information is sent to the respective service interface in Prague at different times and the results are shown using the Feedbacks Viewer tool and the database visualization tools.
2. The feedback data is sent from different Co-Cities Clients and test apps.
3. The test is repeated several times (at least 20).
4. The test is successful if all data is correctly received and shown in the backend.

The following table structure will be used to report on the results:

Period of testing	elapsed time or duration of testing
Number of test sessions	Numeric value
Number of feedbacks correctly received	Numeric value
% of success	Numeric value as a percentage

Table 27 – table for RDSS validation of Quality of Service feedback in Prague

7.3.3.4 Reading

The technical activities in Reading were executed by Telematix along with those for Prague. The business logic behind the developed software components was similar in the two sites, therefore the technical validation RDSS-side in Reading was executed following the same methodology defined for Prague.

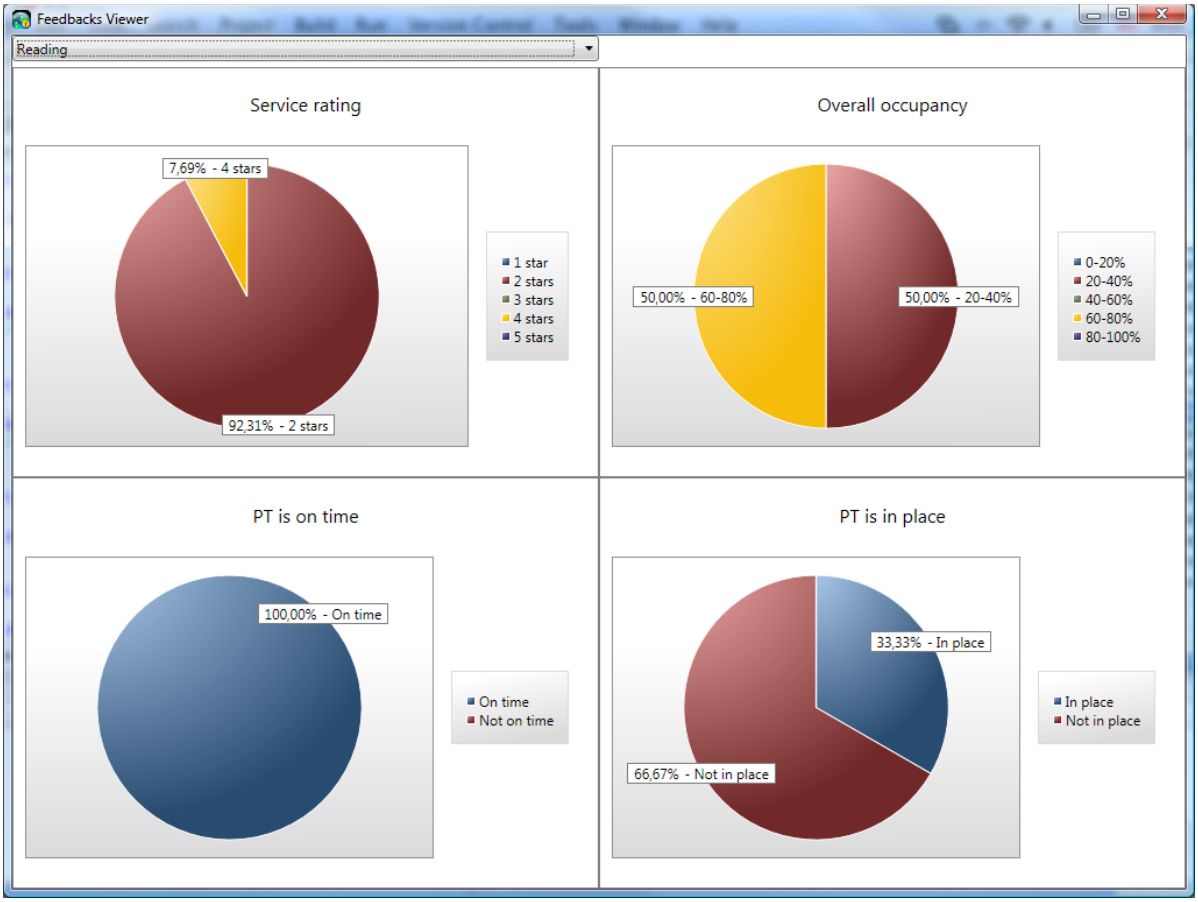

Figure 56 – Telematix Feedbacks Viewer showing Reading feedbacks

7.3.3.4.1 Quality of service feedback

The methodology for the validation of the correct reception of Quality of Service feedbacks in Reading comprises the following steps and related operations:

1. A number of quality of service feedback data is sent to the respective service interface in Reading at different times and the results are shown using the Feedbacks Viewer tool and database visualization tools.
2. The feedback Data is sent from different Co-Cities Clients and test apps.
3. The test is repeated several times (at least 20).
4. The test is successful if all data is correctly received and shown in the backend.

The following table structure will be used to report on the results:

Period of testing	elapsed time or duration of testing
Number of test sessions	Numeric value
Number of feedbacks correctly received	Numeric value
% of success	Numeric value as a percentage

Table 28 – table for RDSS validation of Quality of Service feedback in Reading

7.3.3.4.2 Quality of data feedback

The methodology for the validation of the correct reception of Quality of Data feedbacks in Reading comprises the following steps and related operations:

1. A number of quality of data feedback data is sent to the respective service interface in Reading at different times and the results are shown using the Feedbacks Viewer tool and database visualization tools.
2. The Data is sent from different Co-Cities Clients and test apps.
3. The test is repeated several times (at least 20).
4. The test is successful if all data is correctly received and shown in the backend.

The following table structure will be used to report on the results:

Period of testing	elapsed time or duration of testing
Number of test sessions	Numeric value
Number of feedbacks correctly received	Numeric value
% of success	Numeric value as a percentage

Table 29 – table for RDSS validation of Quality of Service feedback in Reading

7.3.3.4.3 New data feedback

The methodology for the validation of the correct reception of new data feedbacks in Reading comprises the following steps and related operations:

1. A number of new data feedback information is sent to the respective service interface in Reading at different times and the results are shown using the Feedbacks Viewer tool and database visualization tools.
2. The Data is sent from different Co-Cities Clients and test apps.
3. The test is repeated several times (at least 20).
4. The test is successful if all data is correctly received and shown in the backend.

The following table structure will be used to report on the results:

Period of testing	elapsed time or duration of testing
Number of test sessions	Numeric value
Number of feedbacks correctly received	Numeric value
% of success	Numeric value as a percentage

Table 30 – table for RDSS validation of Quality of Service feedback in Reading

7.3.3.5 Tuscany Region

Following the indications and principles specified in the Technical Validation Plan (IR3.1) for the Validation RDSS side, a locally installed validation tool is used in the Tuscany Region to check all aspects of the correct reception, storage and availability of feedback information.

The tool, called e-miXer Co-Cities Console, has been developed during the project for test and demo purposes and uses Softeco's e-miXer technology to provide functionalities for a comprehensive assessment of the feedback information received through the B2B services set up as part of the feedback CAI in Tuscany Region.

The e-miXer Co-Cities Console opens with a dashboard showing statistics about the total number of feedbacks received and the list of Quality of Service Feedbacks including the date, time, Application Id (client sending the feedback) and details on the values for this service's data.

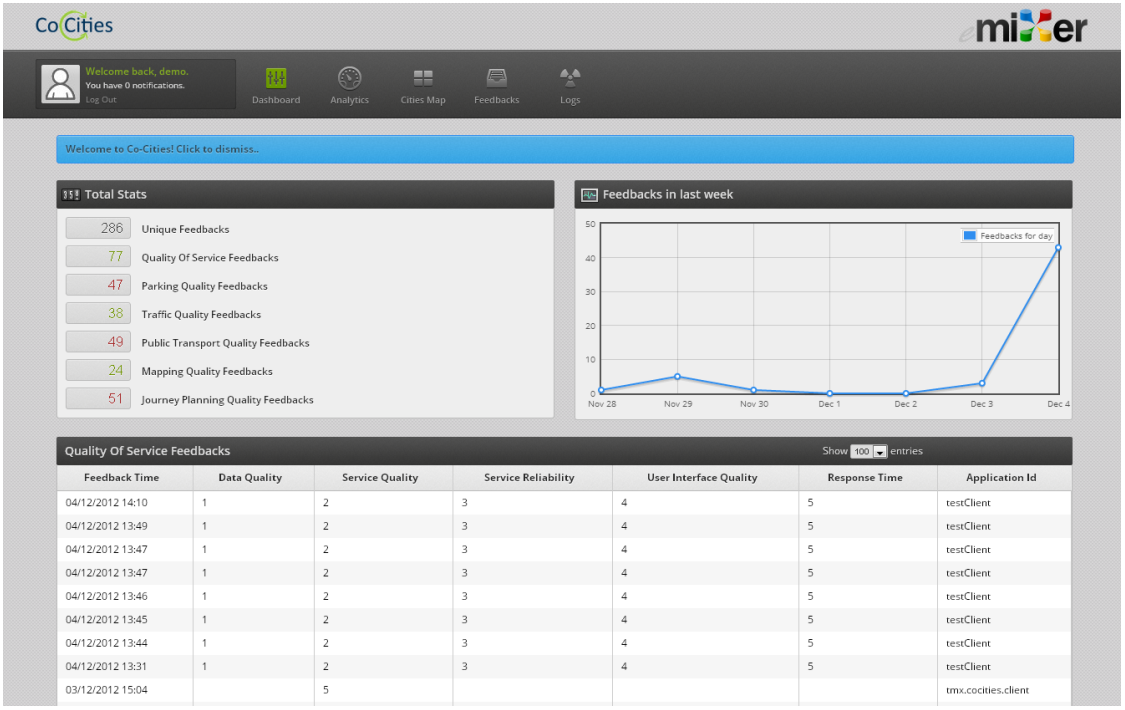

Figure 57 – Dashboard of the e-miXer Co-Cities Console

The "Analytics" menu gives access to statistics in each single domain with a day by day diagram of the feedbacks received during the last month and a list of the quality of data feedback information received for Tuscany Region with the specification of the values entered by the users as feedbacks in each domain.

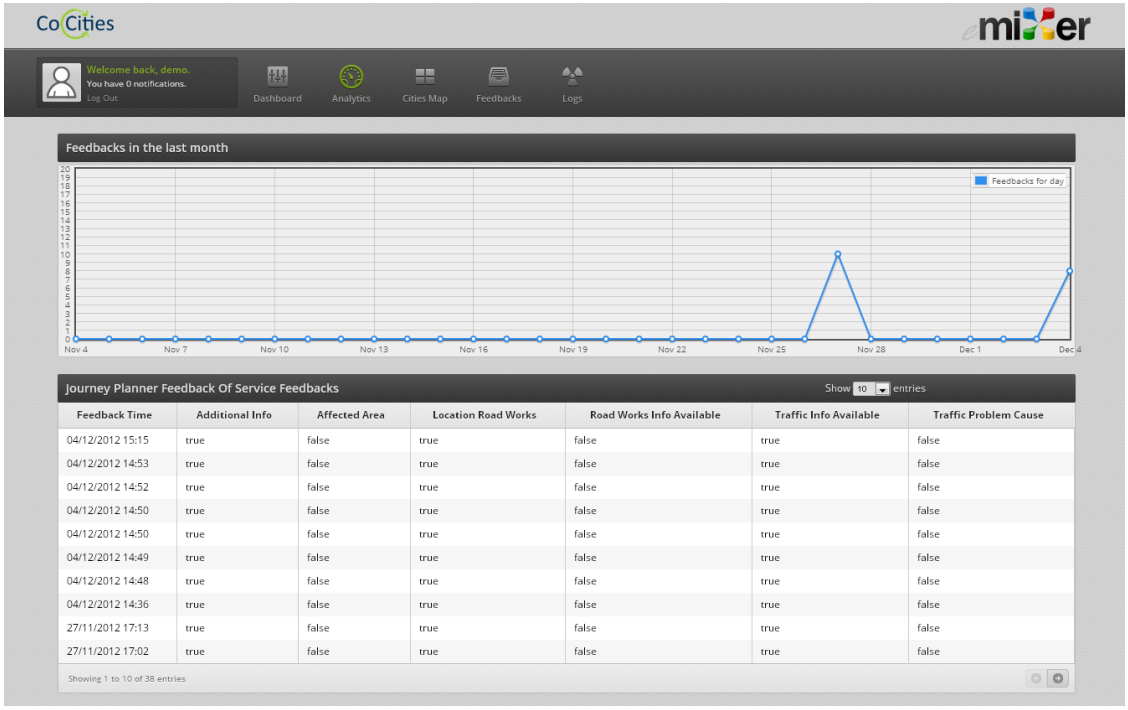

Figure 58 – Report of Quality of data feedbacks in the e-miXer Co-Cities Console

By clicking in a row of the lists containing feedback information more details are shown like the id of the client and the text entered as a comment by the end user.

The e-miXer Co-Cities console is also able to display information about new data provided via the "new data" feedback services in the domains of Public Transport, Traffic and Parking.

The Public Transport and Parking "new data" feedback information received by the CAI in Tuscany Region is shown in tables similar to those of the Quality of service and quality of data feedbacks.

For "new data" feedbacks in the domain of Traffic a special functionality is available to show not only the feedback data as a list but also to aggregate this data with spatial and temporal criteria and to eventually use the aggregated data for creating a new traffic event based on the information provided by the end user. This approach shows the potential use of Co-Cities as a real cooperative system to support traffic centres in the definition or enhancement of the end user information.

The operative schema reproduced here follows the basic steps foreseen by an administration who receives and collect data, validates them and finally introduces a new traffic data via the CO-Cities CAI thus providing enhanced information for the end user.

The following sequence of pictures illustrates the different steps of this process.

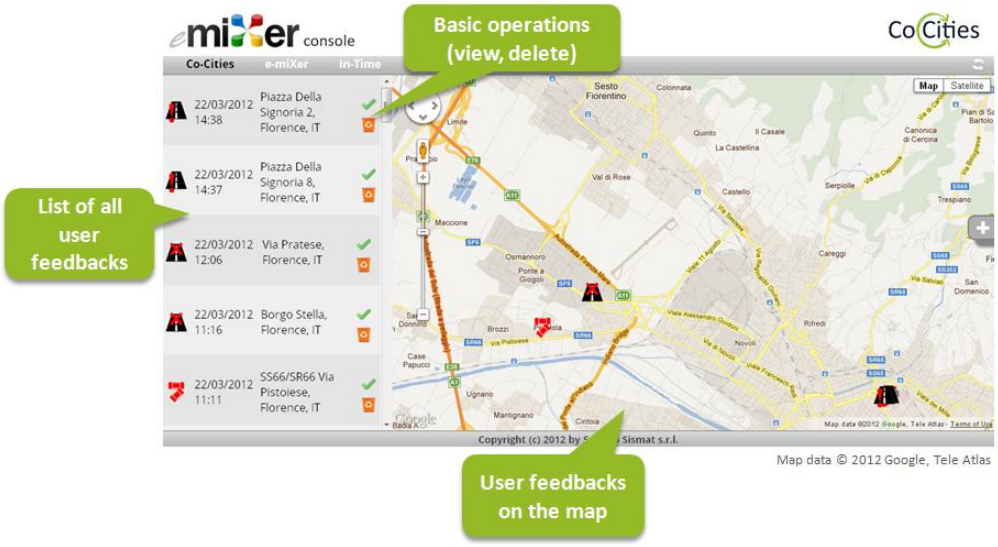

Figure 59 – New traffic data processing – step a

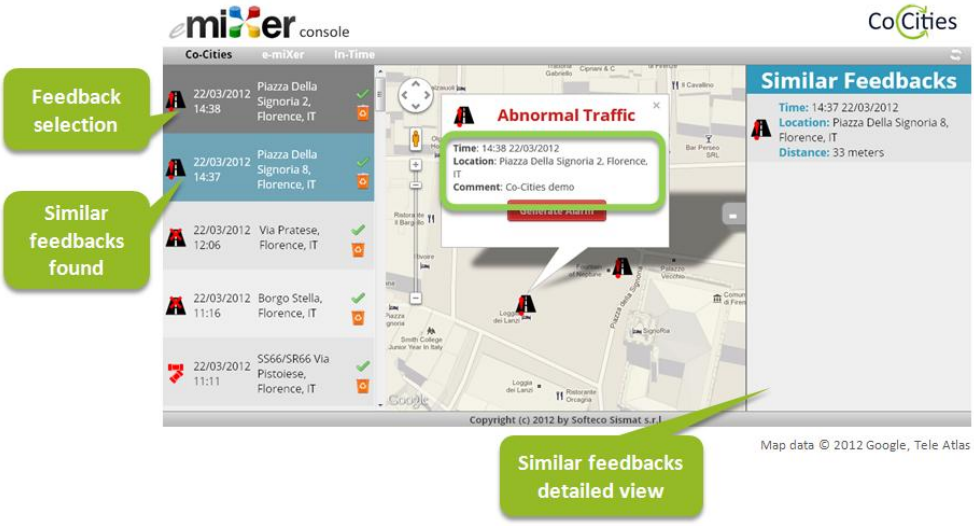

Figure 60 – New traffic data processing – step b

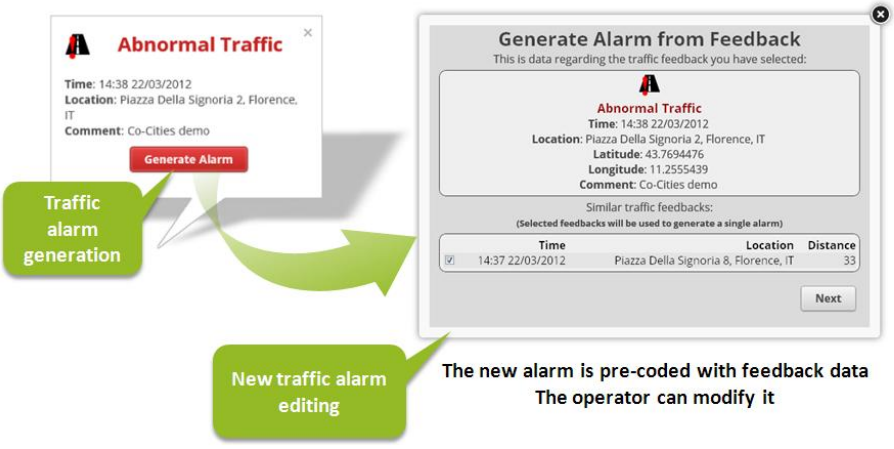

Figure 61 – New traffic data processing – step c

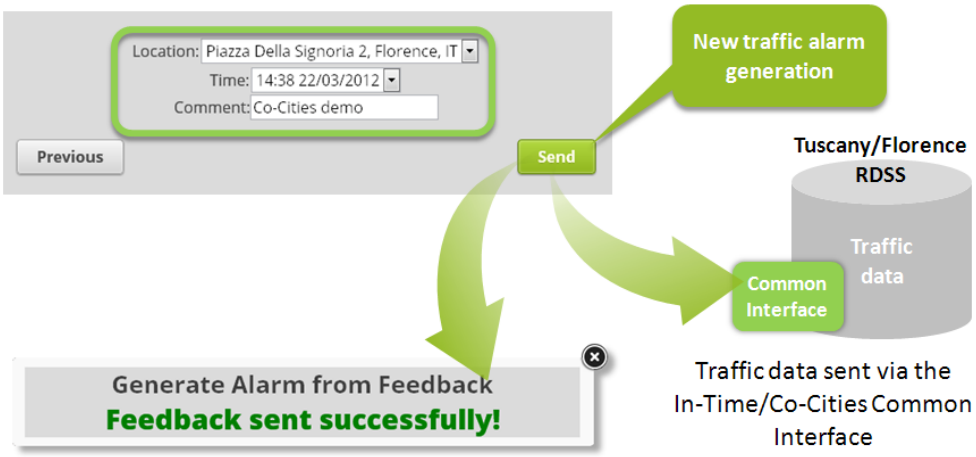

Figure 62 – New traffic data processing – step d

Figure 63 – New traffic data processing – step e

All data shown in the console and composing feedback information (of all types and for all domains) is retrieved from the internal database set up as specified in 5.5.3. This data is available for further usage and other types or channels of storage are possible and the connection between the e-miXer Co-Cities console with the data base allow to validate the correct receipt, storage and availability of feedback information in Tuscany Region either in a test environment or in production.

The following sections describe how data has been collected and how the correct feedback reception has been validated.

7.3.3.5.1 Quality of service feedback

The methodology for the validation of the correct reception of Quality of Service feedback in the Tuscany region comprises the following steps and related operations:

1. A number of quality of service feedback data sets is sent to the respective service interface in the Tuscany region at different times and the results are shown on the e-miXer Co-Cities console.
2. Data is sent from different Co-Cities Clients and test apps.
3. The test is repeated several times (at least 20).
4. The test is successful if all data is correctly received and shown in the related e-miXer Co-Cities console panel.

The following table structure will be used to report on the results:

Period of testing	elapsed time or duration of testing
Number of test sessions	Numeric value
Number of feedbacks correctly received	Numeric value
% of success	Numeric value as a percentage

Table 31 – table for RDSS validation of Quality of Service feedback in Tuscany Region

7.3.3.5.2 Quality of data feedback

The methodology for the validation of the correct reception of Quality of Data feedback in the Tuscany region comprises the following steps and related operations:

1. A number of quality of data feedback data sets is sent to the respective service interface in the Tuscany region at different times and the results are shown on the e-miXer Co-Cities console.
2. Data is sent from different Co-Cities Clients and test apps.
3. The test is repeated several times (at least 20).
4. The test is successful if all data is correctly received and shown in the related e-miXer Co-Cities console panel.

The following table structure will be used to report on the results:

Period of testing	elapsed time or duration of testing
Number of test sessions	Numeric value
Number of feedbacks correctly received	Numeric value
% of success	Numeric value as a percentage

Table 32 – table for RDSS validation of Quality of data feedback in Tuscany Region

7.3.3.5.3 New data feedback

The methodology for the validation of the correct reception of new data feedbacks in the Tuscany region comprises the following steps and related operations:

1. A number of new data feedback information sets is sent to the respective service interface in the Tuscany region at different times and the results are shown on the e-miXer Co-Cities console.
2. Data is sent from different Co-Cities Clients and test apps.
3. The test is repeated several times (at least 20).
4. The test is successful if all data is correctly received and shown in the related e-miXer Co-Cities console panel.

The following table structure will be used to report on the results:

Period of testing	elapsed time or duration of testing
Number of test sessions	Numeric value
Number of feedbacks correctly received	Numeric value
% of success	Numeric value as a percentage

Table 33 – table for RDSS validation of New data feedback in Tuscany Region

7.3.3.6 Vienna

Based on the indications and principles specified in the Technical Validation Plan (IR3.1) for the Validation RDSS side, a Backend-Prototype is used in Vienna to check all aspects of the correct reception, storage and availability of feedback information.

The tool has been developed during the Project for test and demo purposes (e.g. at the ITS WC 2012 in Vienna) and uses Fluidtimes “brivel” framework to provide functionalities for a comprehensive assessment of the feedback information received through the B2B services set up as part of the feedback CAI in Vienna.

The Backend-Prototype visualizes the received feedback in a list and on a map of Vienna.

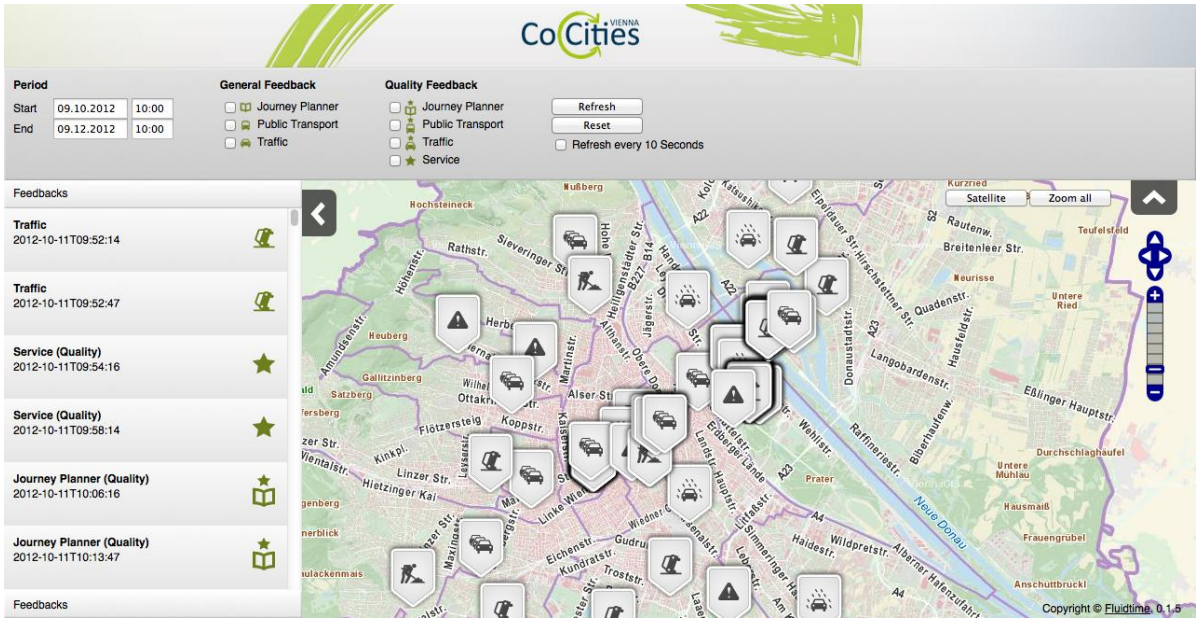

Figure 64 – Backend-prototype RDSS system in Vienna – overview

Feedback data can be filtered according to date and time as well as according to the feedback type (New data / Quality Feedback / General Service Feedback). An Auto-Refresh functionality (every 10 seconds) ensures that the operator keeps an up-to-date overview of the current situation.

Figure 65 – Backend-Prototype RDSS system in Vienna – filter and refresh functionalities

“Quality feedback” and “general service feedback” is visualized with corresponding icons in a list, while geo-referenced “new data” feedback is additionally visualized using markers on an interactive map that allows zooming, panning and switching between satellite and map layers.

Figure 66 – Backend-Prototype RDSS system in Vienna – feedback visualization

Clicking on a feedback item in the list reveals details including the date, time, Application Id (client sending the feedback) and details on the values for this service's data. Additionally the map is centred, if the data is geo-referenced.

Figure 67 – Backend-Prototype RDSS system in Vienna – feedback details

These details contain different data fields depending on the type and domain of the feedback.

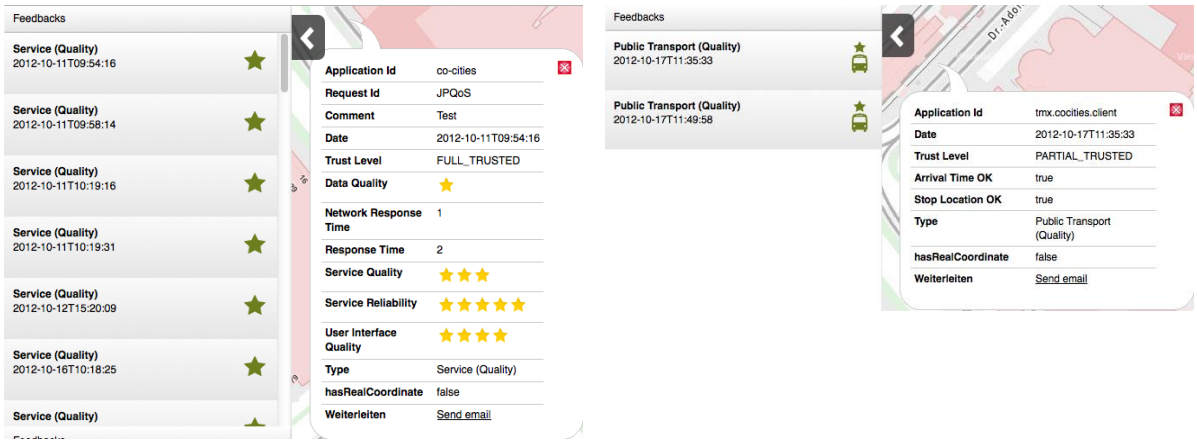

Figure 68 – Backend-Prototype RDSS system in Vienna – feedback data fields

Due to the fact that only In-Time Service 17 (Multimodal Journey Planner) is available in Vienna at the time of writing, feedback, in contrary to the approach described in Tuscany Region, cannot be automatically fed into any In-Time service. However, to support manual processing, feedback data can be forwarded via email to the responsible stakeholders by clicking on the “send email” link depicted in the figure above.

All data shown in the Backend is retrieved from the internal database set up as specified in 5.6.3. This data is available for further usage and other types or channels of storage can be utilised. The connection between the backend and the data base allows validating the correct reception, storage and availability of feedback information in Vienna either in a test or in a production environment.

The following sections describe how data is collected and how the validation of the correct feedback reception is conducted.

7.3.3.6.1 Quality of service feedback

The methodology for the validation of the correct reception of Quality of Service feedbacks in Vienna comprises the following steps and related operations:

5. A number of quality of service feedback data is sent to the respective service interface in Vienna in different times and the results are shown on backend.
6. Data is sent from different Co-Cities Clients and test apps.
7. The test is repeated several times (at least 20).
8. The test is successful if all data is correctly received and shown in the backend.

The following table structure will be used to report on the results:

Period of testing	elapsed time or duration of testing
Number of test sessions	Numeric value
Number of feedbacks correctly received	Numeric value
% of success	Numeric value as a percentage

Table 34 – table for RDSS validation of Quality of Service feedback in Vienna

7.3.3.6.2 Quality of data feedback

The methodology for the validation of the correct reception of Quality of Data feedbacks in Vienna comprises the following steps and related operations:

5. A number of quality of data feedback data is sent to the respective service interface in Vienna in different times and the results are shown on backend.
6. Data is sent from different Co-Cities Clients and test apps.
7. The test is repeated several times (at least 20).
8. The test is successful if all data is correctly received and shown in the backend.

The following table structure will be used to report on the results:

Period of testing	elapsed time or duration of testing
Number of test sessions	Numeric value
Number of feedbacks correctly received	Numeric value
% of success	Numeric value as a percentage

Table 35 – table for RDSS validation of Quality of Service feedback in Vienna

7.3.3.6.3 New data feedback

The methodology for the validation of the correct reception of new data feedbacks in Vienna comprises the following steps and related operations:

5. A number of new data feedback information is sent to the respective service interface in Vienna in different times and the results are shown on backend.
6. Data is sent from different Co-Cities Clients and test apps.
7. The test is repeated several times (at least 20).
8. The test is successful if all data is correctly received and shown in the backend.

The following table structure will be used to report on the results:

Period of testing	elapsed time or duration of testing
Number of test sessions	Numeric value
Number of feedbacks correctly received	Numeric value
% of success	Numeric value as a percentage

Table 36 – table for RDSS validation of Quality of Service feedback in Vienna

8 Report on Technical Validation

8.1 TISP validation

The detailed results of TISP validation, achieved following the validation methodology introduced in section 7 are presented in the internal reports IR4.1 [9] and IR4.2 [10].

A synthesis of the TISP validation results is presented in Annex 1 – Results of TISP validation.

The following table provides the total number of checkmarks **for the features found in the TISP mobile app** as specified in the applicable checklists. The values have been indicated by the responsible Partners following the indications of the validation methodology introduced in section 7:

- Number of positive checkmarks (indicated as √ in the header of the table) recorded during the validation.
- Number of negative checkmarks (indicated as X in the header of the table) recorded during the validation.
- Number of n/a checkmarks (indicated as n/a in the header of the table) recorded during the validation.

The meta-column marked in light-red is related to negative checkmarks, which indicates a failure in the test.

The n/a values refers to features for which at least one of the following conditions is true:

a) The feature is not supported by the TISP due to the absence of the related functionality in the mobile app.
b) The feature is not supported by the site because the related B2B service has not been developed in the site.

The differences between the values in the table indicate the prevalence of factors of type a) or b) in each situation.

Site	Number of features of the mobile app with a positive checkmark	Number of features of the mobile app with a negative checkmark	Number of features of the mobile app for which the test was not applicable
Softeco			
Bilbao	48	0	50
Munich	44	0	54
Prague	47	0	51
Reading	74	0	24
Tuscany Region	88	0	10
Vienna	31	0	67
Telematix			
Bilbao	-	-	-
Munich	-	-	-
Prague	9	0	89
Reading	9	0	89
Tuscany Region	9	0	89
Vienna	9	0	89

Table 37 – Total number of each type of checkmark - TISP validation

8.2 Validation of Communication between TISP and RDSS System

The detailed results of the validation of the Communication between TISP and RDSS systems, achieved following the validation methodology introduced in section 7 are presented in the internal reports IR4.1 [9] and IR4.2 [10].

A synthesis of the validation results is presented in Annex 2 - Results of the validation of the communication between TISP and RDSS.

In the next table the total number of checkmarks related to the results of the validation is presented to provide a more synthesized view of the achieved result.

The totals are calculated over the number of checkmarks provided **for each of the web services involved in the communication TISP-RDSS** as specified in the applicable checklists. The values have been provided by the responsible partners following the indications of the validation methodology introduced in section 7:

- Number of positive checkmarks (indicated as √ in the header of the table)
- Number of negative checkmarks (indicated as X in the header of the table)
- Number of n/a checkmarks (indicated as n/a in the header of the table)

The meta-column marked in light red is related to negative checkmarks which indicates a failure in the test.

The n/a values refers to features for which at least one of the following conditions is true:

a) The web service is not used by the TISP.
b) The web service has not been developed in the site.

The differences between the values in the table indicate the prevalence of factors of type a) or b) in each situation.

Site	Number of web services with a positive checkmark	Number of web services with a negative checkmark	Number of web services for which the test was not applicable
Softeco			
Bilbao	8	0	4
Munich	8	0	4
Prague	7	0	5
Reading	10	0	2
Tuscany Region	12	0	0
Vienna	4	0	8
Telematix			
Bilbao	0	0	12
Munich	0	0	12
Prague	3	0	9
Reading	3	0	9
Tuscany Region	3	0	9
Vienna	3	0	9

Table 38 – Total number of each type of checkmark - Validation of Communication between TISP and RDSS System

8.3 Validation of RDSS System

8.3.1 Bilbao

Following the site-specific methodology set up in 7.3.3.1 the following results have been obtained:

8.3.1.1 Quality of service feedbacks

Period of testing	22 Oct 2012 – 4 Dec 2012
Number of test sessions	20
Number of feedbacks correctly received	20
% of success	**100 %**

Table 39 –Quality of Service Feedback received by RDSS Bilbao

8.3.1.2 Quality of data feedbacks in each domain

Domain: Public Transport	
Period of testing	22 Oct 2012 – 4 Dec 2012
Number of test sessions	25
Number of feedbacks correctly received	25
% of success	**100 %**
Domain: Traffic	
Period of testing	22 Oct 2012 – 4 Dec 2012
Number of test sessions	25
Number of feedbacks correctly received	25
% of success	**100 %**
Domain: Parking	
Period of testing	22 Oct 2012 – 4 Dec 2012
Number of test sessions	20
Number of feedbacks correctly received	20
% of success	**100 %**

Table 40 –Quality of Data Feedback received by RDSS Bilbao

8.3.1.3 New data feedbacks in each domain

Domain: Public Transport	
Period of testing	22 Oct 2012 – 4 Dec 2012
Number of test sessions	20
Number of feedbacks correctly received	20
% of success	100 %
Domain: Traffic	
Period of testing	22 Oct 2012 – 4 Dec 2012
Number of test sessions	20
Number of feedbacks correctly received	20
% of success	100 %
Domain: Parking	
Period of testing	22 Oct 2012 – 4 Dec 2012
Number of test sessions	20
Number of feedbacks correctly received	20
% of success	100 %

Table 41 –New data Feedback received by RDSS Bilbao

8.3.2 Munich

8.3.2.1 Quality of service feedback

The following table summarizes the results:

Period of testing	18.12.2012
Number of test sessions	5
Number of feedbacks correctly received	5
% of success	100

Table 42 – Quality of Service Feedback received by RDSS Munich

Each feedback message also comprised the utilisation of the free text field which was found to work perfectly as well.

8.3.2.2 Quality of data feedback

The following table summarizes the results:

Domain: Parking	
Period of testing	17/18.12.2012
Number of test sessions	75
Number of feedbacks correctly received	75
% of success	100

Table 43 – Quality of Data Feedback received by RDSS Munich

All feedback messages were received within (mostly significantly below) the time frame of one minute counted from the submission on side of the end user device to the reception on side of the visualisation tool. These results satisfied the requirements towards the RDSS feedback management system for actuality and correctness of data.

8.3.3 Prague

Following the site-specific methodology set up in 7.3.3.3 the following results have been obtained:

8.3.3.1 Quality of service feedbacks

Period of testing	28 January 2013
Number of test sessions	25
Number of feedbacks correctly received	25
% of success	**100%**

Table 44 –Quality of Service Feedback received by RDSS Prague

8.3.3.2 Quality of data feedbacks in each domain

Domain: Journey Planning	
Period of testing	28 January 2013
Number of test sessions	25
Number of feedbacks correctly received	25
% of success	**100%**

Table 45 –Quality of Data Feedback received by RDSS Prague

8.3.3.3 New data feedbacks

Domain: Public Transport	
Period of testing	28 January 2013
Number of test sessions	25
Number of feedbacks correctly received	25
% of success	**100%**

Table 46 –New data Feedback received by RDSS Prague

8.3.4 Reading

Following the site-specific methodology set up in 7.3.3.4 the following results have been obtained:

8.3.4.1 Quality of service feedbacks

Period of testing	28 January 2013
Number of test sessions	25
Number of feedbacks correctly received	25
% of success	**100%**

Table 47 –Quality of Service Feedback received by RDSS Reading

8.3.4.2 Quality of data feedbacks in each domain

Domain: Journey Planning	
Period of testing	28 January 2013
Number of test sessions	25
Number of feedbacks correctly received	25
% of success	**100%**

Table 48 –Quality of Data Feedback received by RDSS Reading

8.3.4.3 New data feedbacks

Domain: Public Transport	
Period of testing	28 January 2013
Number of test sessions	25
Number of feedbacks correctly received	25
% of success	**100%**

Table 49 –New data Feedback received by RDSS Reading

8.3.5 Tuscany Region

Following the site-specific methodology set up in 7.3.3.5 the following results have been obtained:

8.3.5.1 Quality of service feedbacks

Period of testing	27 July 2012 – 4 December 2012
Number of test sessions	50
Number of feedbacks correctly received	50
% of success	**100 %**

Table 50 –Quality of Service Feedback received by RDSS Tuscany Region

8.3.5.2 Quality of data feedbacks in each domain

Domain: Public Transport	
Period of testing	26 July 2012 – 4 December 2012
Number of test sessions	45
Number of feedbacks correctly received	45
% of success	**100 %**
Domain: Traffic	
Period of testing	26 July 2012 – 4 December 2012
Number of test sessions	35
Number of feedbacks correctly received	35
% of success	**100 %**

Domain: Parking	
Period of testing	21 September 2012 – 4 December 2012
Number of test sessions	25
Number of feedbacks correctly received	25
% of success	**100 %**
Domain: Journey Planning	
Period of testing	27 July 2012 – 4 December 2012
Number of test sessions	45
Number of feedbacks correctly received	45
% of success	**100 %**

Table 51 –Quality of Data Feedback received by RDSS Tuscany Region

8.3.5.3 New data feedbacks in each domain

Domain: Public Transport	
Period of testing	21 September 2012 – 4 December 2012
Number of test sessions	50
Number of feedbacks correctly received	50
% of success	100 %
Domain: Traffic	
Period of testing	21 September 2012 – 4 December 2012
Number of test sessions	50
Number of feedbacks correctly received	50
% of success	100 %
Domain: Parking	
Period of testing	21 September 2012 – 4 December 2012
Number of test sessions	50
Number of feedbacks correctly received	50
% of success	100 %

Table 52 –New data Feedback received by RDSS Tuscany Region

8.3.6 Vienna

Following the site-specific methodology set up in 7.3.3.6 the following results have been obtained:

8.3.6.1 Quality of service feedbacks

Period of testing	22 October 2013 – 10 December 2013
Number of test sessions	30 (15 with SOF for JP; 15 with TMX Client)
Number of feedbacks correctly received	30
% of success	**100%**

Table 53 –Quality of Service Feedback received by RDSS Vienna

8.3.6.2 Quality of data feedbacks in each domain

Domain: Journey Planning	
Period of testing	22 October 2013 – 10 December 2013
Number of test sessions	30
Number of feedbacks correctly received	30
% of success	**100%**

Table 54 –Quality of Data Feedback received by RDSS Vienna

8.3.6.3 New data feedbacks

Domain: Traffic	
Period of testing	22 October 2013 – 10 December 2013
Number of test sessions	25
Number of feedbacks correctly received	25
% of success	**100%**

Table 55 –New data Feedback received by RDSS Vienna

9 Conclusions

Co-Cities features a significant number of newly developed software components, running at a centralized level (Reference Platform, mobile apps) as well as at local level (Co-Cities Commonly Agreed Interface). These components have been developed in the central part of the Project timeline based on the technical specification developed in the first year of the project.

The development of the Co-Cities system in the six test Sites and by the Traffic Information Service Providers was supported by a strict technical coordination, necessary to ensure that the developed software was fully in line with the technical objectives fixed for the Co-Cities system, including:

- Efficient communication of all actors along the value chain
- Full interoperability between the TISPs and the Local systems
- Establishment of efficient cooperative features
- Full integration in the Co-Cities SOA along the indications of the Technical Specification

The activities resulted in the development of a number of interoperable services in all demonstration sites (Bilbao, Munich, Prague, Reading, Tuscany Region and Vienna) and across different information domains comprising public transport, road traffic, parking and multimodal journey planning.

Services required for the provision of the information from the local systems were implemented where necessary (in sites which have not participated in In-Time already) while new services for the provision of feedbacks (from the end user) have been developed in all sites. From the requirement of ensuring backwards compatibility with the In-Time interface, existing In-Time Cities have generally kept or enhanced the running services for information provision while new cities developed new ones. In one case (Tuscany Region) a geographical extension of existing In-Time services (previously only installed in Florence) to a regional level has been achieved. In all sites the new cooperative part of the interface has been developed and integrated in the existing infrastructure to collect and store feedback data. The description of the developed local CAI is contained in chapter 5.

Two dedicated Co-Cities mobile applications have been developed to run on the two mobile platforms with the highest worldwide sales[9]: Android and iOS. The description of the developed TISP apps is in section 6.

The technical Validation of the Co-Cities system has been a key pre-requisite to ensure that every technical element is correctly in place for the WP5000 evaluation phase. Overall, the technical Validation of the Co-Cities system comprised three main aspects:

- The validation TISP side, which comprises the assessment of the functionalities available TISP side for each Site and for each service domain

[9] http://en.wikipedia.org/wiki/Mobile_operating_system - as in Q3 2012

- The validation of the communication between the RDSS and the TISP which is monitored by the Reference Platform , set up as a central, rule based, configurable communication platform, serving as a reference system for all involved partners. Deliverable D4.2 specifically describes the development and usage of this component
- The validation of the correct receipt of the feedback data from the RDSS

A methodology for the technical validation has been developed in the Internal Report IR3.1 and briefly resumed in chapter 7.

As reported in details in chapter 8 and in the Internal Report 4.1, the technical validation (Alpha test and Beta Test) of the Co-Cities system was successful for the mix of services integrated in the TISP apps.

The following picture shows the groups of components specifically developed in Co-Cities and their interconnections. The bidirectional communication flow relates not only to the request/response procedures associated to the provision of the information, but also to the flow of data associated to feedback exchange as presented in details in the definition of the Co-Cities Service Delivery Chain specifically formulated as a key prerequisite for the correct execution of all development and integration activities.

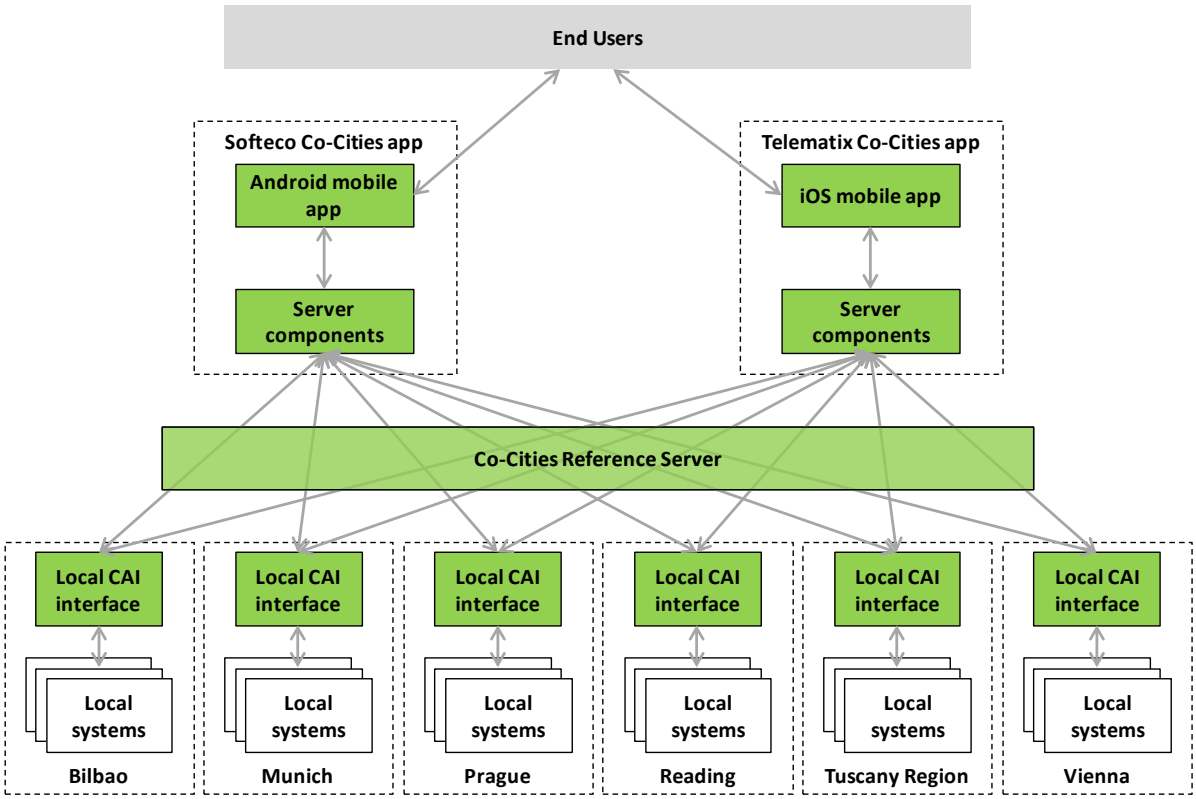

Figure 69 – Overview of the developed components in the Co-Cities SOA

The infrastructure for cooperative mobility, comprising all interface elements and an initial number of available and dedicated mobile applications, is ready to be exploited by local authorities or

organizations in any application or scenario where feedback information can be used to enhance the information quality provided to the end user. To concretely demonstrate the potential of the Co-Cities cooperative approach, the partners responsible for the technical developments designed and developed a number of prototypal web-based systems able to show how feedback data is received and how, in example application scenarios, it is processed. These systems have also been used to validate the correct system integration and deployment for the aspects related to the correct reception of feedback data. For this reason a presentation the prototypes can be found in chapter 7.

The successful development and installation of all software components, on the one hand offered the possibility to fully test the concepts, principles and behavior of the Co-Cities approach. Additionally, the availability of fully operative systems at local sites and mobile applications from TISPs is the precondition to successfully start the operational and evaluation phase in project Work Package 5000.

10 Glossary

The glossary provides the coherent terminological framework used in this document.

10.1 Abbreviations

AVM	Automatic Vehicle Monitoring
DB	Data Base
CAI	Commonly Agreed Interface
CBA	Cost Benefit Analysis
CIMUBISA	Bilbao Council's computing centre
DoW	Description of Work
ETA	Estimated Time of Arrival
ESB	Enterprise Service Bus
FCD	Floating Car Data
GPS	Global Positioning System
MIIC	Mobility Integrated Information Centre
MSS	Measurement Support System
OGC	Open Geospatial Consortium
PDF	Portable Document Format
PT	Public Transport
RDSS	Regional Data / Service Server
RTTI	Real Time Traffic Information
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
SWP	Sub Work Package
TISP	Traffic Information Service Providers
UML	Unified Modeling Language
VIB	Verkehrsinformationsagentur Bayern GmbH
VPN	Virtual Private Network
WFS	Web Feature Service

WMS	Web Map Service
WP	Work Package
WSDL	Web Services Description Language
XML	Extensible Markup Language
XSD	XML Schema Definition

11 References

The following references are used as background documents for the preparation of this document. References are categorized standards (i.e. standards and specifications from the consortium working groups or alliances and specifications or drafts standardization bodies) and other documents, publications and technical or scientific books.

[1]	Co-Cities project. Cooperative Cities extend and validate mobility services. Deliverable D2.1 *"Report of cooperative cities services and set use cases" ("Service definition and use cases")*, January 2012
[2]	Co-Cities project. Cooperative Cities extend and validate mobility services. Deliverable D2.2 *"List of user groups and interaction process" ("User groups and interaction process")*, January 2012
[3]	Co-Cities project. Cooperative Cities extend and validate mobility services. Deliverable D2.3 *"Validation strategy for existing systems, including extensions and reference system test cases"*, March 2012.
[4]	Co-Cities project. Cooperative Cities extend and validate mobility services. Deliverable D3.1 *"ITS system specification description and reference platform for validation"*, March 2012.
[5]	Co-Cities project. Cooperative Cities extend and validate mobility services. Deliverable D3.2 *"ITS system interfaces to end users and service delivery chain of the service providers"*, April 2012.
[6]	Co-Cities project. Cooperative Cities extend and validate mobility services. Internal Report IR3.1 *"Technical Validation Plan for TISPs and RDSSs"*, November 2012
[7]	Co-Cities project. Cooperative Cities extend and validate mobility services. Deliverable D4.1 *"Test report on interfaces, data structures and working service deliveries"*, May 2012.
[8]	Co-Cities project. Cooperative Cities extend and validate mobility services. Deliverable D4.2 *"Reference platform, it´s configuration and usage manual"*, May 2012.
[9]	Co-Cities project. Cooperative Cities extend and validate mobility services. Internal Report IR4.1 *"Cooperative Services test report"*, December 2013
[10]	Co-Cities project. Cooperative Cities extend and validate mobility services. Internal Report IR4.2 *"Service interfaces and installations test report"*, January 2013
[11]	In-Time project. Intelligent and efficient travel management for European cities. http://www.in-time-project.eu/

	http://www.in-time-project.eu/en/library/deliverables/
[12]	eMotion project. European-wide multi-Modal On-trip Traffic InformatION. http://www.emotion-project.eu/
[13]	HCI, Human systems Interaction, http://en.wikipedia.org/wiki/Human–computer_interaction
[14]	Human-Computer Interaction by Steve Love, IEEE Distributed Systems Online, vol. 7, no. 10, 2006, art. no. 0610-ox006.
[15]	Powell, D. (Ed.) (1991). Delta-4: A Generic Architecture for Dependable Distributed Computing. Re-search Reports ESPRIT. Project 818/2252 Delta-4 Vol.1. ISBN 3-540-54985-4 Springer-Verlag 1991
[16]	Jacobson, I., Bylund, S., Jonsson, P., and Ehneboom, S. (1995), "Modeling with Use Cases: Using contracts and use cases to build plugable architectures". Journal of Object Oriented Programming, Vol. 8, No. 2, pp. 18-24.

11.1 Normative references

OGC Standards and Specifications	Open Geospatial Consortium specifications http://www.opengeospatial.org/standards
IEEE 829-1998	IEEE Standard for Software Test Documentation
IEEE 1012-2004	IEEE Standard for Software Verification & Validation Plans (V&V)
IEEE 1059-1993	IEEE Guide for Software Verification and Validation Plans

11.2 Documents and books

W3C (2004). Web Services Architecture.	W3C Working Group Note 11 February 2004. http://www.w3.org/TR/ws-arch/
OASIS - SOA-RM (2006)	OASIS Reference Model for Service Oriented Architecture 1.0. Committee Specification 1, 2 August 2006. http://www.oasis-open.org/

12 Annex 1 – Results of TISP validation

The tables of this annex provides the following information related to the results of the TISP validation, synthesized from the relevant checklists which have been filled in by the responsible Partners:

- Number of positive checkmarks (indicated as √ in the header of the table) recorded during the validation
- Number of negative checkmarks (indicated as X in the header of the table) recorded during the validation
- Number of n/a checkmarks (indicated as n/a in the header of the table) recorded during the validation

The table holds information identifying the available feedback and service delivery functionalities which were validated:

- Part I: Availability In-Time Service
- Part II: Availability of detailed feedback functions per data/service item
- Part III: General quality feedback
- Part IV: new data feedback

Finally the total number of checkmarks are further associated to the respective service domains

The meta-column marked in light-red is related to negative checkmarks, which indicates a failure in the test.

TISP Validation	Multimodal journey planning	Parking	Traffic	Public Transport	Multimodal journey planning	Parking	Traffic	Public Transport	Multimodal journey planning	Parking	Traffic	Public Transport
Part I: Availability of In-Time Service	-	4	-	1					6	3	2	1
Part II: Availability of detailed feedback functions per data/service item	-	11	-	6					6	-	4	-
Part III: General quality feedback	-	6	-	6					6	-	6	-
Part IV: new data feedback	-	6	5	3					11	3	-	2
	√ (TEST PASSED)				X (TEST FAILED)				n/a.			
	Softeco (Android)											
Total per service domain	**0**	**27**	**5**	**16**	**0**	**0**	**0**	**0**	**29**	**6**	**12**	**3**
Total	**48**				**0**				**50**			

Table 56 – Synthesis of the results of TISP validation in Bilbao

TISP Validation	Multimodal journey planning	Parking	Traffic	Public Transport	Multimodal journey planning	Parking	Traffic	Public Transport	Multimodal journey planning	Parking	Traffic	Public Transport
Part I: Availability of In-Time Service	-	2	2	1					6	5	-	1
Part II: Availability of detailed feedback functions per data/service item	-	11	4	6					6	-	-	-
Part III: General quality feedback	-	6	6	6					6	-	-	-
Part IV: new data feedback	-	-	-	-					11	9	5	5
	√ (TEST PASSED)				X (TEST FAILED)				n/a.			
	Softeco (Android)											
Total per service domain	**0**	**19**	**12**	**13**	**0**	**0**	**0**	**0**	**29**	**14**	**5**	**6**
Total	**44**				**0**				**54**			

Table 57 – Synthesis of the results of TISP validation in Munich

TISP Validation	Multimodal journey planning	Parking	Traffic	Public Transport	Multimodal journey planning	Parking	Traffic	Public Transport	Multimodal journey planning	Parking	Traffic	Public Transport
Part I: Availability of In-Time Service	6	-	-	1					-	7	2	1
Part II: Availability of detailed feedback functions per data/service item	6	-	-	6					-	11	4	-
Part III: General quality feedback	6	-	-	6					-	6	6	-
Part IV: new data feedback	8	-	5	3					3	9	-	2
	√ (TEST PASSED)				X (TEST FAILED)				n/a.			
	Softeco (Android)											
Total per service domain	**26**	**0**	**5**	**16**	**0**	**0**	**0**	**0**	**3**	**33**	**12**	**3**
Total	**47**				**0**				**51**			

Table 58 – Synthesis of the results of TISP validation in Prague - Softeco

TISP Validation	Multimodal journey planning	Parking	Traffic	Public Transport	Multimodal journey planning	Parking	Traffic	Public Transport	Multimodal journey planning	Parking	Traffic	Public Transport
Part I: Availability of In-Time Service	6	-	-	-					-	7	2	2
Part II: Availability of detailed feedback functions per data/service item	-	-	-	1					6	11	4	5
Part III: General quality feedback	-	-	-	1					6	6	6	5
Part IV: new data feedback	-	-	-	1					11	9	5	4
	√ (TEST PASSED)				X (TEST FAILED)				n/a.			
	Telematix (iOS)											
Total per service domain	**6**	**0**	**0**	**3**	**0**	**0**	**0**	**0**	**23**	**33**	**17**	**16**
Total	**9**				**0**				**89**			

Table 59 – Synthesis of the results of TISP validation in Prague - Telematix

TISP Validation	Multimodal journey planning	Parking	Traffic	Public Transport	Multimodal journey planning	Parking	Traffic	Public Transport	Multimodal journey planning	Parking	Traffic	Public Transport
Part I: Availability of In-Time Service	6	4	-	1					-	3	2	1
Part II: Availability of detailed feedback functions per data/service item	6	11	-	6					-	-	4	-
Part III: General quality feedback	6	6	-	6					-	-	6	-
Part IV: new data feedback	8	6	5	3					3	3	-	2
	√ (TEST PASSED)				X (TEST FAILED)				n/a.			
	Softeco (Android)											
Total per service domain	**26**	**27**	**5**	**16**	**0**	**0**	**0**	**0**	**3**	**6**	**12**	**3**
Total	**74**				**0**				**24**			

Table 60 – Synthesis of the results of TISP validation in Reading - Softeco

TISP Validation	Multimodal journey planning	Parking	Traffic	Public Transport	Multimodal journey planning	Parking	Traffic	Public Transport	Multimodal journey planning	Parking	Traffic	Public Transport
Part I: Availability of In-Time Service	6	-	-	-					-	7	2	2
Part II: Availability of detailed feedback functions per data/service item	-	-	-	1					6	11	4	5
Part III: General quality feedback	-	-	-	1					6	6	6	5
Part IV: new data feedback	-	-	-	1					11	9	5	4
	√ (TEST PASSED)				X (TEST FAILED)				n/a.			
	Telematix (iOS)											
Total per service domain	**6**	**0**	**0**	**3**	**0**	**0**	**0**	**0**	**23**	**33**	**17**	**16**
Total	**9**				**0**				**89**			

Table 61 – Synthesis of the results of TISP validation in Reading - Telematix

TISP Validation	Multimodal journey planning	Parking	Traffic	Public Transport	Multimodal journey planning	Parking	Traffic	Public Transport	Multimodal journey planning	Parking	Traffic	Public Transport
Part I: Availability of In-Time Service	6	5	2	2					-	2	-	-
Part II: Availability of detailed feedback functions per data/service item	6	11	4	6					-	-	-	-
Part III: General quality feedback	6	6	6	6					-	-	-	-
Part IV: new data feedback	8	6	5	3					3	3	-	2
	√ (TEST PASSED)				X (TEST FAILED)				n/a.			
	Softeco (Android)											
Total per service domain	**26**	**28**	**17**	**17**	**0**	**0**	**0**	**0**	**3**	**5**	**0**	**2**
Total	**88**				**0**				**10**			

Table 62 – Synthesis of the results of TISP validation in Tuscany Region – Softeco

TISP Validation	Multimodal journey planning	Parking	Traffic	Public Transport	Multimodal journey planning	Parking	Traffic	Public Transport	Multimodal journey planning	Parking	Traffic	Public Transport
Part I: Availability of In-Time Service	6	-	-	-					-	7	2	2
Part II: Availability of detailed feedback functions per data/service item	-	-	-	1					6	11	4	5
Part III: General quality feedback	-	-	-	1					6	6	6	5
Part IV: new data feedback	-	-	-	1					11	9	5	4
	√ (TEST PASSED)				X (TEST FAILED)				n/a.			
	Telematix (iOS)											
Total per service domain	**6**	**0**	**0**	**3**	**0**	**0**	**0**	**0**	**23**	**33**	**17**	**16**
Total	**9**				**0**				**89**			

Table 63 – Synthesis of the results of TISP validation in Tuscany Region – Telematix

TISP Validation	Multimodal journey planning	Parking	Traffic	Public Transport	Multimodal journey planning	Parking	Traffic	Public Transport	Multimodal journey planning	Parking	Traffic	Public Transport
Part I: Availability of In-Time Service	6	-	-						-	7	2	2
Part II: Availability of detailed feedback functions per data/service item	6	-	-						-	11	4	6
Part III: General quality feedback	6	-	-						-	6	6	6
Part IV: new data feedback	8	-	5						3	9	-	5
	√ (TEST PASSED)				X (TEST FAILED)				n/a.			
	Softeco (Android)											
Total per service domain	**26**	**0**	**5**	**0**	**0**	**0**	**0**	**0**	**3**	**33**	**12**	**19**
Total	**31**				**0**				**67**			

Table 64 – Synthesis of the results of TISP validation in Vienna – Softeco

TISP Validation	Multimodal journey planning	Parking	Traffic	Public Transport	Multimodal journey planning	Parking	Traffic	Public Transport	Multimodal journey planning	Parking	Traffic	Public Transport
Part I: Availability of In-Time Service	6	-	-	-					-	7	2	2
Part II: Availability of detailed feedback functions per data/service item	-	-	-	1					6	11	4	5
Part III: General quality feedback	-	-	-	1					6	6	6	5
Part IV: new data feedback	-	-	-	1					11	9	5	4
	√ (TEST PASSED)				X (TEST FAILED)				n/a.			
	Telematix (iOS)											
Total per service domain	**6**	**0**	**0**	**3**	**0**	**0**	**0**	**0**	**23**	**33**	**17**	**16**
Total	**9**				**0**				**89**			

Table 65 – Synthesis of the results of TISP validation in Vienna

13 Annex 2 - Results of the validation of the communication between TISP and RDSS

In the following tables, the following information is provided for each TISP and for each Site:

- Number of positive checkmarks (indicated as √ in the header of the table)
- Number of negative checkmarks (indicated as X in the header of the table)
- Number of n/a checkmarks (indicated as n/a in the header of the table)

The table provides information about the result of the test for each data set contained in the related checklist for each of the web services:

- JourneyPlanning
- Parking
- PublicTransport
- Traffic
- JourneyPlanningQualityFeedback
- ParkingQualityFeedback
- TrafficQualityFeedback
- PublicTransportQualityFeedback
- ParkingFeedback
- PublicTransportFeedback
- TrafficFeedback
- ServiceFeedback

The meta-column marked in light red is related to negative checkmarks which indicates a failure in the test.

	Services involved in the communication TISP-RDSS	Softeco (Android)			Telematix (iOS)		
		√	X	n/a	√	X	n/a
Services for the provision of information	JourneyPlanning			1			1
	Parking	1					1
	PublicTransport	1					1
	Traffic	1					1
Quality of data	JourneyPlanningQualityFeedback			1			1
	ParkingQualityFeedback	1					1
	TrafficQualityFeedback	1					1
	PublicTransportQualityFeedback	1					1
New data	ParkingFeedback			1			1
	PublicTransportFeedback			1			1
	TrafficFeedback	1					1
Overall quality of service	ServiceFeedback	1					1
	Total	**8**	**0**	**4**	**0**	**0**	**12**

Table 66 – Synthesis of results of the technical validation of the communication TISP-RDSS in Bilbao

	Services involved in the communication TISP-RDSS	Softeco (Android)			Telematix (iOS)		
		√	X	n/a	√	X	n/a
Services for the provision of information	JourneyPlanning			1			1
	Parking	1					1
	PublicTransport	1					1
	Traffic	1					1
Quality of data	JourneyPlanningQualityFeedback			1			1
	ParkingQualityFeedback	1					1
	TrafficQualityFeedback	1					1
	PublicTransportQualityFeedback	1					1
New data	ParkingFeedback			1			1
	PublicTransportFeedback			1			1
	TrafficFeedback	1					1
Overall quality of service	ServiceFeedback	1					1
	Total	**8**	**0**	**4**	**0**	**0**	**12**

Table 67 – Synthesis of results of the technical validation of the communication TISP-RDSS in Munich

	Services involved in the communication TISP-RDSS	Softeco (Android)			Telematix (iOS)		
		√	X	n/a	√	X	n/a
Services for the provision of information	JourneyPlanning	1			1		
	Parking			1			1
	PublicTransport	1					1
	Traffic			1			1
Quality of data	JourneyPlanningQualityFeedback	1			1		
	ParkingQualityFeedback			1			1
	TrafficQualityFeedback			1			1
	PublicTransportQualityFeedback	1					1
New data	ParkingFeedback			1			1
	PublicTransportFeedback	1					1
	TrafficFeedback	1					1
Overall quality of service	ServiceFeedback	1			1		
	Total	**7**	**0**	**5**	**3**	**0**	**9**

Table 68 – Synthesis of results of the technical validation of the communication TISP-RDSS in Prague

	Services involved in the communication TISP-RDSS	Softeco (Android)			Telematix (iOS)		
		√	X	n/a	√	X	n/a
Services for the provision of information	JourneyPlanning	1			1		
	Parking	1					1
	PublicTransport	1					1
	Traffic			1			1
Quality of data	JourneyPlanningQualityFeedback	1			1		
	ParkingQualityFeedback	1					1
	TrafficQualityFeedback			1			1
	PublicTransportQualityFeedback	1					1
New data	ParkingFeedback	1					1
	PublicTransportFeedback	1					1
	TrafficFeedback	1					1
Overall quality of service	ServiceFeedback	1			1		
	Total	**10**	**0**	**2**	**3**	**0**	**9**

Table 69 – Synthesis of results of the technical validation of the communication TISP-RDSS in Reading

	Services involved in the communication TISP-RDSS	Softeco (Android)			Telematix (iOS)		
		√	X	n/a	√	X	n/a
Services for the provision of information	JourneyPlanning	1			1		
	Parking	1					1
	PublicTransport	1					1
	Traffic	1					1
Quality of data	JourneyPlanningQualityFeedback	1			1		
	ParkingQualityFeedback	1					1
	TrafficQualityFeedback	1					1
	PublicTransportQualityFeedback	1					1
New data	ParkingFeedback	1					1
	PublicTransportFeedback	1					1
	TrafficFeedback	1					1
Overall quality of service	ServiceFeedback	1			1		
	Total	**12**	**0**	**0**	**3**	**0**	**9**

Table 70 – Synthesis of results of the technical validation of the communication TISP-RDSS in Tuscany Region

	Services involved in the communication TISP-RDSS	Softeco (Android)			Telematix (iOS)		
		√	X	n/a	√	X	n/a
Services for the provision of information	JourneyPlanning	1			1		
	Parking			1			1
	PublicTransport			1			1
	Traffic			1			1
Quality of data	JourneyPlanningQualityFeedback	1			1		
	ParkingQualityFeedback			1			1
	TrafficQualityFeedback			1			1
	PublicTransportQualityFeedback			1			1
New data	ParkingFeedback			1			1
	PublicTransportFeedback			1			1
	TrafficFeedback	1					1
Overall quality of service	ServiceFeedback	1			1		
	Total	**4**	**0**	**8**	**3**	**0**	**9**

Table 71 – Synthesis of results of the technical validation of the communication TISP-RDSS in Vienna