The Intelcities e-City Platform: a framework for a new generation of local e-government services

Rémi Vankeisbelck[1], François Giraud-Carrier[2], Jérôme Duchon[3], Alain Zarli[1]
[1]*CSTB, 290 route des Lucioles, BP 209, 06904 Sophia Antipolis Cedex, France*
[2]*Derbi, 18 boulevard de la Bastille 75012 Paris, France*
[3]*CS, rue Brindejonc des Moulinais 31500 Toulouse, France*
f.giraud-carrier@derbi.fr / alain.zarli@cstb.fr

Abstract: The research presented in this paper has been undertaken in the context of the Intelcities project of the 6th framework Information Society Technologies programme of the European Commission. The main objective of Intelcities is to create a new and innovative set of interoperable, interactive, citywide on-line e-government applications and services, to meet the future needs of both citizens and businesses. This paper presents the e-City Platform, an integrated information system for cities based on OSS principles, and illustrates it through a working prototype for the delivery of 'joined-up' e-administration city services.

1 Introduction

The work and activities presented in this paper have been undertaken in the context of the 6th framework Information Society Technologies research programme of the European Commission. A major integrated project, the Intelcities initiative, has been launched in January 2004 within the "networked businesses and governments" area. The main objective of this project is to create a new and innovative set of interoperable e-government services to meet the needs of both citizens and businesses, providing interactive citywide on-line applications and services for users that will make all aspects of "what is going-on" in the city available to all. The aim is to help achieve the EU policy goal of the "Knowledge Society" by 2010 through new forms of electronic governance of cities and greater social inclusion through enhanced access to services by citizens and businesses.

Intelcities is addressing the research and development of an integrated information system for cities based on OSS[1] principles – known as the e-City Platform or e-CP. This is to integrate the complete range of current and developing day-to-day e-government and city management services with new e-planning and urban regeneration services addressing the city of tomorrow.

The project is structured around three groups of interrelated activities:

1. ICT research and innovation with a focus on integration and interoperability issues;
2. Prototype studies, including development and demonstration activities within cities;

[1] Open Source Software

3. Change management, including knowledge management, capacity building and e-governance development and innovation.

This paper more specifically concentrates on the first two activities. It illustrates the integration and interoperability dimension of the e-CP through a working prototype for the delivery of enhanced e-administration services.

The paper first introduces the concept of enhanced city service, a transformation in the delivery of public services leading to user-centred, highly accessible, seamlessly integrated, transactional services. It then details the methodology for the development of the e-CP leading to the definition and presentation of its conceptual architecture. The concepts and technical components of the platform architecture have been carefully designed independently of any technology implementation in order to allow as much as possible replacement of a functional module or technology by another and the reuse of existing OSS.

As an illustration, the paper places the e-CP in context through one of Intelcities' demonstrators dedicated to employment and professional training and developed at the request of the cities of Marseille and Nice (France). The development process of an innovative employment and training service that uses the e-CP to deliver enhanced services is presented and commented.

2 The concept of enhanced city service

Throughout Europe, the modernisation of public services through the adoption of information and communication technologies is in motion. There are, all around us, evidences of a universal shift toward modern online public services (e-services). This phenomenon is reflected by a global consensus and remarkable results obtained at all levels of government in Europe (see the "eEurope 2005 Mid-term Review" published on February 2004).

The introduced concept of enhanced city service refers to a new generation of local online public services that are to be one-stop, "end-up" or "joined-up" services. The objective is to allow the city's inhabitants to access and deal with interactive and transactional global services that are not only information services but genuine and **complete** procedures. The development of such services especially imply a partnership of the main stakeholders providing basic services and contents that will be aggregated and syndicated within the enhanced service. An ambient, enhanced service is:

- a service that can be delivered on different types of devices, in particular "democratic" devices such as mobile phones, televisions, kiosks and PCs (Accessibility dimension);
- a transactional service (bi-directional exchange of data), and not just the provision of information, in order to provide the ability, for citizens, to get meaningful answers to their requests (Interaction dimension);
- requiring several providers to collaborate and integrate their services to deliver a global, comprehensive answer to a particular query (the one-stop city shop) (Integration dimension).

The relevance, dynamism and power of local e-governance is characterised by the ability for cities to rapidly deploy resources and mobilize critical mass partnerships. This is a key requirement to achieve integrated e-government and multi-platform access, supporting a move to a demand-driven approach that emphasises service delivery, end-user value for all and "functionality". In the context of this paper, two important objectives of the Intelcities research are emphasized:

- Help cities modernise the delivery of public local services by providing them with a framework and platform that supports the integration of legacy systems (the city's and its partners') as well as the adoption of reusable components and services.
- Propose a "vertically" integrated set of public services, i.e. one that involves national, regional and local government services, a decentralised approach to e-government that emphasizes the local dimension so as to be closer to citizens' needs, more human and interconnected with the city planning activities.

3 The e-CP: principles, architecture and main components

The nature of the Intelcities project and the large number of stakeholders involved in the development of its components requires a solid strategy for managing the different e-Governance services available to the end-users from the participating cities along with the issues of integration and interoperability of legacy systems communicating through the e-CP. This situation also presents an important concern in terms of the integration of the different applications since these applications are independently developed by these cities. We first describe the methodology followed to design the e-CP Architecture and secondly, present an overview of this architecture and its main underlying mechanisms.

3.1 Development methodology

In order to tackle the above mentioned objectives and to establish a long lasting communication platform, an Enterprise Architecture (EA) approach was considered to develop what we call the "e-CP Architecture". This section briefly introduces the consecutive steps and methodologies used to define the e-CP Architecture. This includes the presentation of a Business Reference Model showing the processes, interactions and business objects captured from the analysis of the e-Governance domain, and presents the development of the three Architectural Components forming the complete e-CP Architecture: the Conceptual, Technical and System Architectures:

- The Conceptual architecture analyses the e-Governance domain in order to identify the business reference models in terms of processes, interactions, business objects, etc.
- The Technical architecture analyses the e-Services developed in the context of Intelcities in order to identify the technical needs in terms of interoperability. It defines the structure of the platform, independently of the technology chosen; and
- The System architecture applies technology to the technical needs in order to define the overall architecture. This way, technological limitations can be identified, as well as their consequences on the entire architecture.

The three aforementioned architecture layers are included in the development process of the platform as shown in Fig. 1 below:

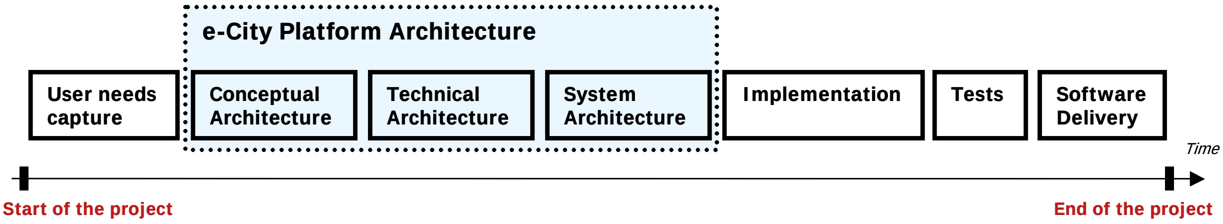

Figure 1: Development process of the e-CP

In order to define a high-quality architecture, we took advantage of the Rational Unified Process methodology (RUP), currently used in the business domain. The reason for choosing

RUP is that it covers the entire development process lifecycle and enables high quality software product. RUP is a Use-Case-driven methodology, architecture centric and incremental which facilitates the use of the Unified Modelling Language "UML" in an effective way. The use of RUP is believed to help defining an architecture which should cover all the requirements expressed by the users.

3.2 Scope and objectives of the e-CP

One of the first steps in the EA framework is the analysis of the business domain, in our case, e-Governance. The diagram below introduces the business object model of the e-Governance domain and shows the essential concepts and their relationships.

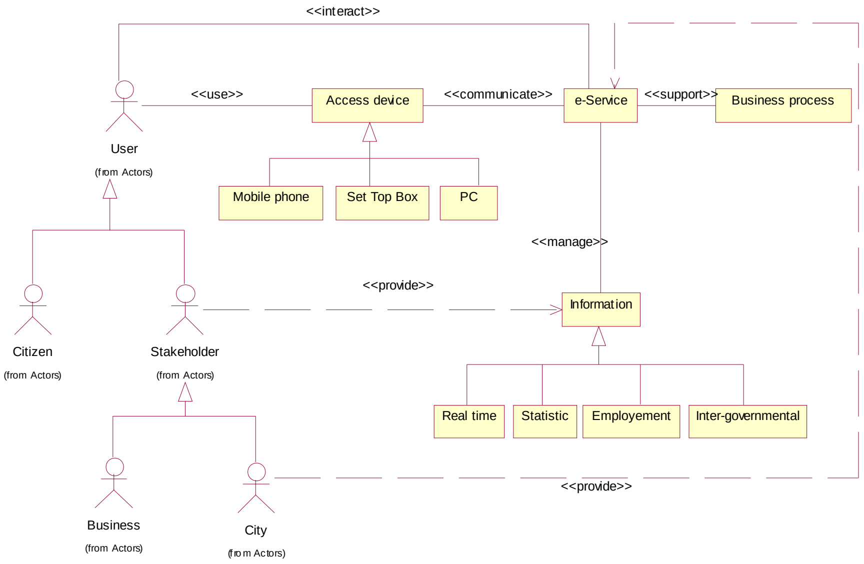

Figure 2: Business Object Model diagram

These concepts are:

- *User* uses *Access device* as a support to interact directly with e-Governance services in order to access information. The list of *Access device* is not exhaustive and devices such as Variable Message Signs for traffic information or PDA should also be addressed in the project.
- *Real time* is the information relative to day-to-day life.
- *Statistic* represents information such as demography, crimes, etc.
- *Inter-governmental* is the information used internally by the administration.
- *Business process* represents processes of the governance domain that manage *Information*. It includes passport renewal, tax declaration, etc.
- *e-Service* represents e-Governance service. It supports *Business process* in order to improve the activities of public sector organisation and manages *Information*. *e-Services* are provided by *City*.

The technical objectives we want to achieve with the e-CP when implementing the aforementioned concepts are double: **integration** and **interoperability**.

1. Integration of legacy and new city services, to allow these systems to be more focused on the business issues by relying on offered facilities for infrastructure management ;
2. Interoperability between these city services, to allow them to share their functional capabilities and datasets in order to define new services with more added value.

When following these objectives, we have to pay attention to the heterogeneity of the services which range from simple transmission of electronic forms to real-time communications for traffic management and interaction with 3D datasets. We present hereafter a distributed architecture that tackles such goals.

3.3 The e-CP architecture

Analysis of e-Governance concepts within Intelcities, together with the requirements of citizens and cities has lead to the adoption of a distributed architecture as shown below.

3.3.1 Conceptual Architecture

This distributed architecture uses an n-tier decomposition which can be seen as follow:

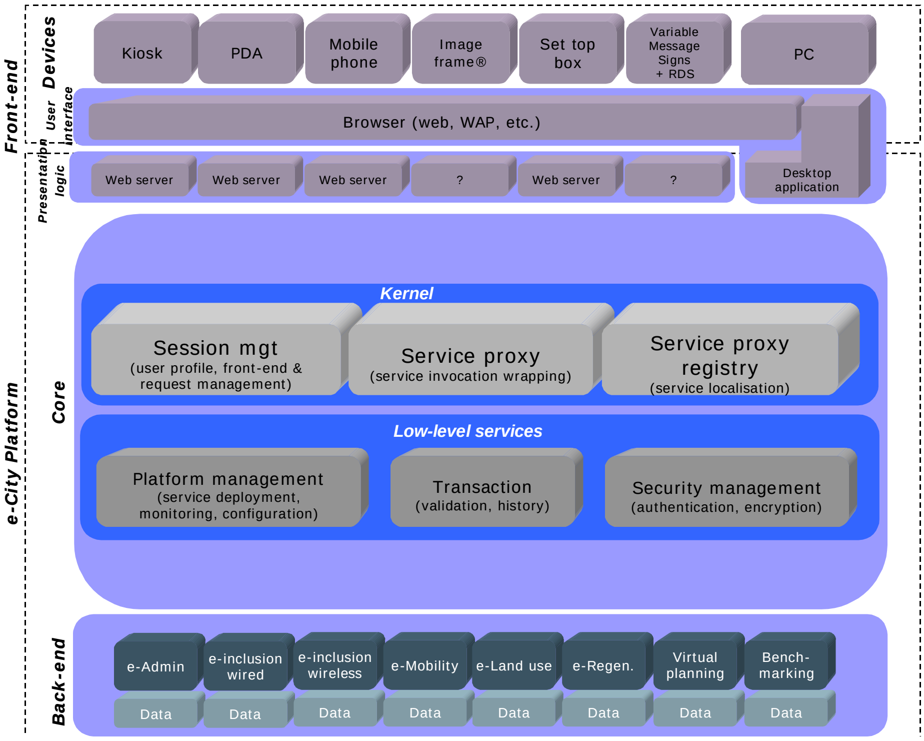

Figure 3: the e-CP conceptual architecture

At the bottom of the figure, the back-end services to be developed are represented. They can either be services developed within the e-CP framework or city legacy systems. They encompass the business and data layers of city services. At the top, the front-end layer allows users to interact with the system through different devices while the application layer encompasses the communication mechanism with the middleware of the platform. The e-CP core is the set of infrastructure services that provides transversal medium or low-level features required in ICT systems like security, transactions, network communications, benchmarking, etc. It can be broken down into the following elements:

- Kernel services : session and service management ;
- Low-level services: platform administration and monitoring services like authentication or data encryption).

The two main advantages for adopting such an architecture are :

1. Client layers (front-end and back-end layers) can access common infrastructure services which decrease the complexity of these clients. This is the integration goal of the e-CP.
2. Back-end services can communicate through a common mechanism to build more complex city applications. This is the interoperability goal of the e-CP.

The architecture underlying mechanisms (session management, service proxy, security, etc.), have been designed independently of the implementation, in order to get a stable conceptual architecture even if the techniques and the technologies used for the implementation change during the Intelcities project(s) : alpha and beta releases, pre-product, product.

3.3.2 Technical Architecture

We present here the main techniques that have been mapped onto the afore mentioned e-CP concepts. The key modules for achieving the above mentioned integration and interoperability goals are located in the *e-CP core*, inside the *Kernel* box on Figure 3. So we explain hereafter the role of these software components, what is their added value, and how they are used with a typical example of *front-end connection* to *back-end service*.

To begin, let us explain what we call *Service* and *Client* inside e-CP context. The *Service* role represents a software component able to handle *Client* requests for accessing shared operations and procedures (i.e. business logic) and indirectly the data. The available public methods are defined in the API of the *Service*. In our conceptual architecture on Figure 3, this role is fulfilled by *back-end services*. The *Client* role represents a software component requesting a *Service* through the *e-CP core*. Any application that wants to interact with *Services* must implement such a component. In our conceptual architecture, this role is either fulfilled by *front-end applications* or *back-end services*.

3.3.2.1 Service Proxy

The *e-CP core* is based on a key concept, which is called *Service Proxy*. This mechanism allows injecting transversal low-level features when the *Client* requests the *Service*.

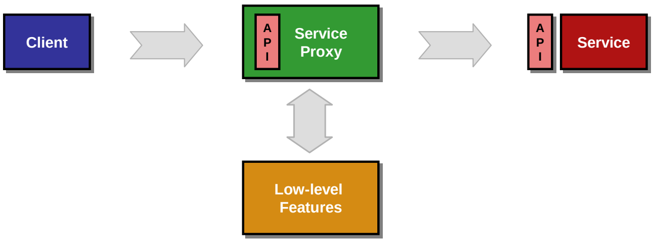

Figure 4: The Service Proxy concept

The *Service Proxy* is part of the *e-CP core* and is a way to wrap a *Service* interface (API) in order to inject transversal low-level features when invoking the *Service* (e.g. security checks, load-balancing). This way, *Clients* can request *Service* public methods transparently through the *Service Proxy*, as well as they would directly request the real *Service*. This mechanism is a way to add features to *Services* without being intrusive in the *Service* itself which is **essential** to deal with legacy systems. The *Service* can run remotely without being integrated physically into the *e-CP*. The Service Proxy is generated inside the *e-CP core* during the deployment process of a Service.

3.3.2.2 Session Management

In order to request a *Service*, a *Client* (i.e. a *back-end service* or *front-end application*) will have to open a *Client Session*. This session will be created only once for the first request. Opening a *Client Session* doesn’t require authentication. The same session will be used until the disconnection of the *Client* or if the session timeout delay has expired. This way, the *e-CP core* will always identify the *Client* that makes a given request. Some *Service* methods may require a user identity in parameter for processes dependant of the user identity. Before invoking such a method, *Client* will have to open a *User Session* through a *Client Session* after providing valid user credentials. After validating the credentials, the *User Session* is created and associated with the corresponding user identity. Finally, *Client* will be able to give a valid *User Session* in parameter of the *Service* methods that require a user identity.

3.3.2.3 Service Localisation

The service localisation concept is the way *Clients* will be able to locate the *Service Proxy* associated with the wanted *Service*. The component *Service Proxy Registry* will associate Service identification information (e.g. name, description etc.) with corresponding *Service Proxy*. A *Client* get a reference on a *Service Proxy* in order to request a *Service*. The *Client* requests the *Service Proxy Registry* with identification information on the wanted *Service*. From this request, the *Service Proxy Registry* is able to provide the corresponding *Service Proxy* reference. For a better understanding of how these techniques are chained together, Figure 5 shows a scenario of a *front-end application* connecting and communicating with a *back-end service* via the *e-CP core*:

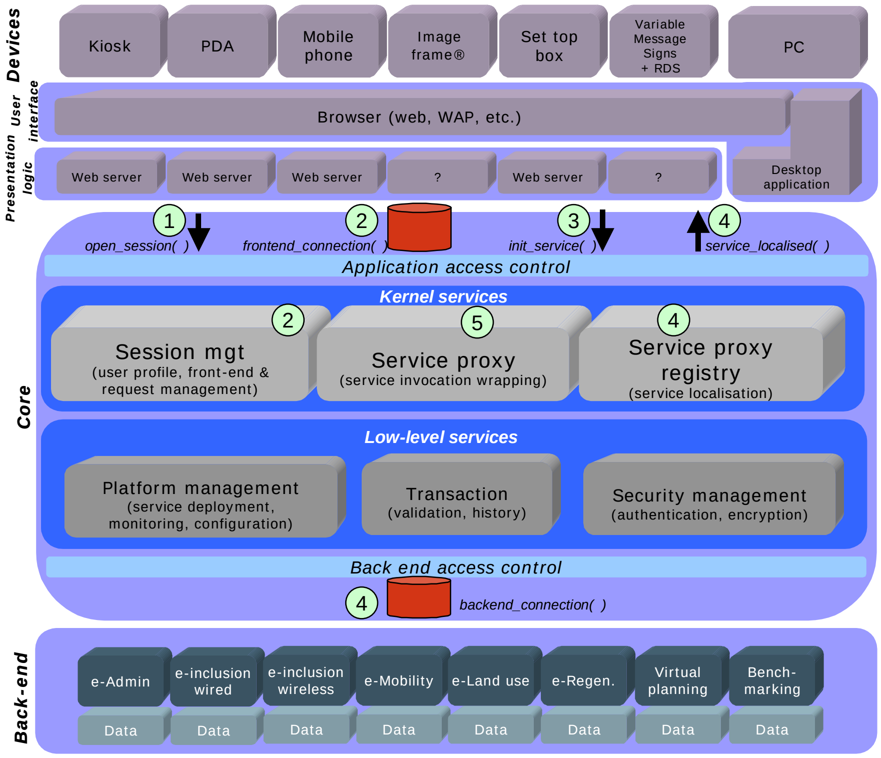

Figure 5: Connection scenario with the e-CP

1. The *front-end application* requests the *core* to open a session. This includes exchange of information such as the service security policy and other hints (e.g. benchmark requirements).
2. The *session management* module establishes a connection with the *front-end application*.

3. The *front-end application* requests for *back-end service* initialisation.
4. The *service proxy registry* localises the concerned *back-end service*, establishes the connection with it and returns *service proxy* reference to the *front-end application*.
5. The *front-end application* communicates with the *back-end service* via the *service proxy*, the *service proxy* being in charge of wrapping transparently needed features (data encryption, load balancing, etc.) to the communications.

3.4 What's next?

In order to succeed in the development of the e-Governance platform for cities, we have defined a high-quality architecture that analyse the e-Governance domain and we provided a technical solution. In the alpha release, the kernel of the platform makes use of Corba middleware for infrastructure services communication but also for the communication with clients at back-end and front-end. The main advantage of such technology is to have implementations available for a wide choice of languages such as C++, Java or ADA whereas other middleware are constrained to one implementation or platform (e.g. J2EE with Java). The Intelcities ecosystem spans from e-administration transactions to interactive 3D visualisation requests which, even if they all do not pass through the middleware, require efficient communications.

At this stage of development, we have presented an alpha release of the e-CP that is currently being integrated by several cities. This integration will allow implementing simple business use cases to validate the proof of concept. In the next release, features below will be implemented or, at least investigated allowing cities to deploy their services onto more complex cases, including legacy systems.

- Security (authentication, ACL) with generic profiles authentication, "pluggable" user management system and access control to services. Thereby, e-Administration services will rely on the platform's built-in mechanisms for Users and Roles management, authentication, access control, etc.
- 2-Phase Commit distributed transactions: they will be included to a provide e-Administration with advanced transactional features, like online job submission, etc.
- Integration to existing frameworks and technologies: the e-CP requires technologies that already exist or are under investigation, like Aspect Oriented Programming, Code Generation techniques and Declarative Services. Even more, the concept itself is very close to existing Component Technologies like Entreprise Java Beans or CORBA Component Model. During its future development, the e-CP should rely on these "base bricks" where needed instead of competing with it.
- Techniques like load-balancing, clustering, replication, and quality of service are not yet investigated. Of course, the e-CP relies on standard middleware so it can be extended to include these features. This work is planned for the final e-CP specification release.

4 The e-CP demonstration: the employment and training service

4.1 Demonstration scope

The demonstrator aims at validating intra-service processes as well as eCP integration mechanisms. It should allow the testing of the Use Cases and processes specified by the cities, as well as make the proof of concept for already specified integration features and low-level services in the platform.

4.2 Scenario

The demonstrator supports the process of providing information to citizens on employment and training opportunities. The scenario goes in 2 steps: The person looking for a training programme will fill an electronic form that details his or her situation (age, current qualification, financial requirements, etc.). The system will return the opportunities and public assistance that apply to him or her together with the list all available training programmes. The job search follows the same pattern.

4.3 Prototype architecture

The e-CP relies on a fully modular, multi-tier, service-oriented architecture, as explained above. This demonstrator shows integration of the so called INTELCITIES "e-Admin" back-end service into the e-CP Core, as well as the invocation of this service via a web interface.

Figure 6 below shows the overall architecture of the prototype application. On this figure, one will notice the "regular" three tiers (Presentation Layer, Business Layer and Resource Layer), as well as the additional "e-CP Layer" that provides transversal, infrastructure-level features. Let's have a closer look to this architecture:

- The "back-end" groups Resource, Business and e-CP layers. Its role is to encapsulate the "service" and to provide it to clients. Each back-end tier handles part of this role as described in the following sections.
- Resources Layer; Employment and Training data are at the moment stored into and retrieved from a single Relational Database Management System (MySQL). In a second step, the service should include various employment and training data sources. It could also include geographic, transport or economic data.
- This Business Logic layer encapsulates business logic that can be executed through method invocations on the service. It relies on the Resource Layer to manipulate Employment and Training data, compute it, and provide consistent answers to the client's requests. On the technical point of view, business logic is encapsulated in regular Java classes, and exposed to clients through a CORBA Façade. The IDL interface describes available business methods.

This e-CP Layer is where the e-CP comes into scene, to handle all transversal features in the application as a whole. The e-CP imposes to clients to request service invocation to the platform itself, so that it can perform transversal tasks when needed (security checks, transaction management). The concept of "Service Proxy" allows the e-CP to define additional behaviours and "decorate" the service itself in order to control its invocation. This is very close to the Aspect Oriented Programming paradigms, applied to distributed, heterogeneous city services. Eventually, the user (Front-end) can interact with the service by filling in forms and browsing results via a regular web-based application based on J2EE and Struts. This application is a Client of the e-CP: Struts Actions, Servlets and JSPs in this application make use of the back-end service through the e-CP Layer in a transparent fashion.

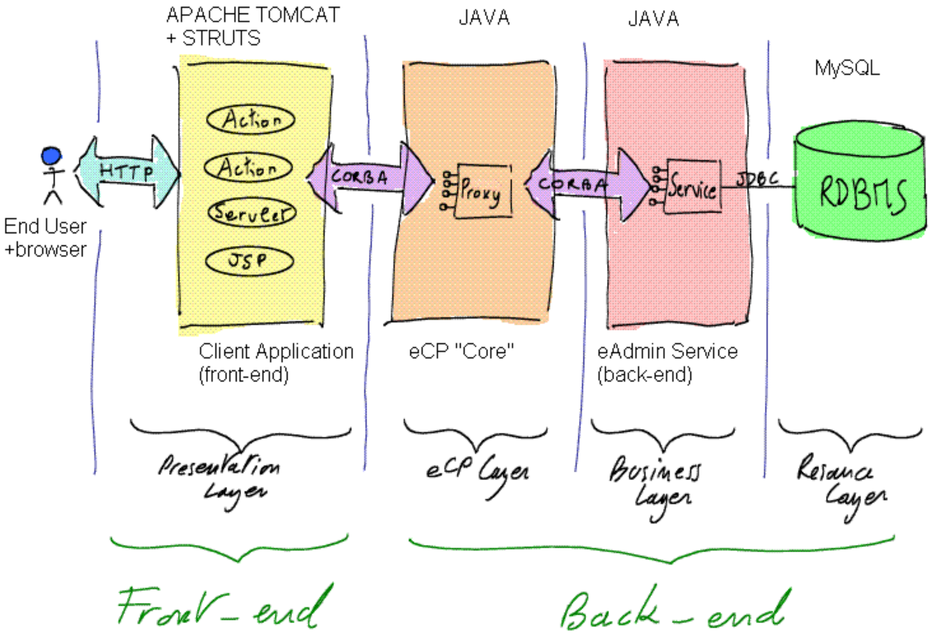

Figure 6: Prototype architecture

5 Conclusion

New forms of electronic governance are at an experimental stage today throughout Europe, with the underlying objective of ensuring that all the applications and services can be made to work together in a seamless and interoperable manner. Intelcities is a major initiative in this field, with the specific ambition to develop an open information-integrated system (the so-called e-CP) that will ensure the interoperability between new e-Government services and their integration with legacy systems, as well as the handing of all low-level features (e.g. security, transactions, etc.). One of the business fields[2] targeted by the Intelcities project is e-administration, and more precisely the provision of joined-up city services, whose objective is to develop “ambient intelligence” in cities for the benefits of all citizens. Especially, this paper has introduced to this new kind of services, that should lead for citizens to accessing interactive and transactional global services – and that are indeed not only information services, but genuine and complete procedures, to be based on further commitment and agreement among various different bodies providing basic services and content that will be aggregated / syndicated within some enhanced service. This paper has presented the groundings to future works for generalisation and full deployment, under a Pan-European perspective, of a new generation of ambient, interactive on-line City services.

[2] Other considered business fields are*: e-Inclusion and e-Participation, e-Mobility and e-Transport, e-Land-Use, and e-Regeneration and urban planning.*