de Alfonso, Carlos; Caballer, Miguel; Carrión, José V.; Hernández, Vicente

gCitizen: A Grid Middleware for a Transparent Management of the Information about Citizens in the Public Administration

gCitizen: A Grid Middleware for a Transparent Management of the Information about Citizens in the Public Administration

Carlos de Alfonso[1], Miguel Caballer[1], José V. Carrión[1], Vicente Hernández[1]

[1] Departamento de Sistemas Informáticos y Computación, Universidad Politécnica de Valencia
{calfonso,micafer,jocarbur,vhernand}@itaca.upv.es

Abstract

This paper proposes the Grid technology as an integration method of information, existing procedures and resources in the Public Administration. The exposed work supposes, from the point of view of the electronic government, an advance of future trends by means of the usage of Grid technology. On the other hand, from the perspective of Grid technology, the electronic government opens a non-evident field of application for this emergent paradigm of distributed computing. This paper explains gCitizen, which is a Grid middleware based on the GT4 components and WSRF implementation (which are the state-of-the-art in middleware for Grid computing), incorporating new protocols and services which cover the requirements for the integration purposes in the eGovernment frameworks. The system architecture has been designed to enable using the services deployed without a specific a priori knowledge of them. The gCitizen middleware also defines a data model to provide interoperability in the exchange of the information among the different gCitizen services.

Key words: eGovernment, Grid, WRSF, Globus, Interoperability, Integration

This paper is Available online at
www.jtaer.com

1   Introduction

The Public Administration is divided into specialized parts which have their own data about the citizens. Despite having most of that information in an electronic format, it is not usually used in a coordinated way due to the absence of standards for data exchange on the eGovernment processes. On the other side, Grid technologies have been traditionally applied to scientific computing challenges. Nevertheless, that is a simplification of the Grid concepts, which were conceived for the integration of general heterogeneous systems. Currently there is a lack of protocols, mechanisms, etc. for discovering and using other services than computation or storage.

This work proposes Grid technology as an integration method of the information which is managed by the Public Administration. Grid enables a framework to improve the interoperability in the administration processes, providing important features as transparency, ubiquity, security, etc.

The gCitizen project [4] is based on the GT4 [18] components and WSRF [29] implementation, as this framework is the state-of-the-art in middleware for deploying wide area Grids. Globus Toolkit has also been erected as the current de facto standard in Grid technologies. Further than the Globus Toolkit core components, it has been developed new protocols and services which cover the requirements for the integration purposes: General Addressing Convention which provides location independence to services, a Distributed Service Discovery Architecture which avoids the need of central components for discovering services, a General Logging framework for distributing the logs in the system, an ontology for eGovernment Grid services which enables a plug-and-play usage, among others. This set of components provides features such as (1) visibility, for determining whether a service has been deployed or not; (2) mobility of services, which would ease the enhancement of the infrastructure, fabric scalability, etc., but also would enable the creation of ad-hoc infrastructures based on itinerant services provided by mobile devices; or (3) redundancy of services, which enable the component scalability, load balancing, etc.

The architecture of the system developed enables the use of the services deployed without a priori knowledge of them. This feature is achieved consulting the information properties of the services, and using any of the predefined functions of the gCitizen services.

The key component for the exchange of information is a flexible and interoperable data model which has been designed, and the mapping rules from the existing data models to the arranged one. This data model is based in the e-GIF definitions, but adapted to the particular needs of the Spanish government and the applications which this government manages.

The paper is organized as follows: section 2 introduces the eGovernment concept and the ICT requirements for this kind of environments. Section 3 shows a brief introduction to Grid technology and its uses in other fields than scientific computation. Next, the section 4 introduces the gCitizen project with the different components designed and developed. Finally, section 5 summarizes and introduces some further work to be carried out.

2   ICT and eGovernment

In the 90s, the governments noticed the opportunity of applying ICTs [5], creating the denominated Electronic Government or eGovernment. Up to now, the governments have created legislation for the adoption of the ICTs with the purpose of the enhancement of their work. So, the administrative units are applying current technologies for substituting the paper and registries. This usage of the eGovernment technologies is more specifically called eAdministration.

The public administration usually applies the information systems provided by the computer science technology in different aspects of its daily work. Nevertheless most of these systems only apply to an internal level of the administration. They have the main disadvantage that the usage of ICTs have not been consolidated in every area in which have been applied. Therefore it resorts to conventional techniques. Some of the reasons that make difficult the introduction of the ICTs in the public administration are commented next.

- **Different levels of ICT deployment**. Each administration has grown in an independent way, according to its economic and social possibilities. This causes that some administrative units have their procedures fully computerized, while others hardly have computers.

- **Incomplete applications**. In many cases, the information systems and the computing solutions proposed for the administrative scope do not cover all the requirements (legal, functional, etc.) of the administrative procedures. This causes that its usage is reduced and its lack of functionality

supposes a negative predisposition to adopt more suitable tools, from the point of view of the users.

- **Usage of proprietary applications**. The computer science solutions proposed for the administration have been traditionally proprietary. This has caused a very important dependence on the companies that developed them. At the same time this dependence has limited the options of choosing among different approaches and points of view for the new solutions development.

- **Lack of adoption by the users**. The personnel in the administration are the final user of the computing systems. As many users, they are usually reticent to changes and they only accept those systems which guarantee a more comfortable way to work. The changes do not have to suppose a conceptual change of their work or to learn the procedures again.

The result of the lack of a global framework for ICT is a set of information systems which usually can not interchange information among them. So, it is interesting to integrate all these systems in order to have a scenario in which the distinct administrative units would be able to collaborate.

2.1   eGovernment Requirements

The eGovernment environments have special requirements, due to its complex organization, and the kind of information that they handle [5]. Some other requirements have also appeared with the generalization of the ICTs in the administration and with the advent of new technologies:

- **Security**. According to laws, only some people are allowed to read or to modify data in the administration, and in many cases only with the consent of the person who is the target of the access. That is why it is essential to grant the identity of the people who access the data or carries out with an administration procedure, and authorise them according to legal issues.

- **Privacy**. The data which is stored in the administration is the subject of some laws. Also the target of the data (citizen or person on behalf of a business) has to specifically permit the access to these data. Privacy of data must be specially taken into account when using ICTs, because they are conceptually more vulnerable to security break outs.

- **Interoperability**. The distinct entities in the administration usually use distinct data sources and data models (as by now there is no standard for storage or data processing in eGovernment). There is a need of increasing the level of interoperability among systems in the public administration, for enabling the collaboration and increase the efficiency.

- **Geographically distributed resources**. The administration is divided into several areas which are physically distributed in a country, city, etc. It must be taken into account that these resources are not likely to be in a private network, and they would be acceded through public networks.

- **Ubiquitous access** Notwithstanding the geographical distribution of the resources, in a common administrative workspace the access should be granted to users, independently from the point of access. This feature would enable mobility of the users and approach to citizens and business.

- **Fault tolerance**. It is essential that the failure of one component of the system doesn't make the whole system to stop working. Some techniques and features must be introduced to solve local problems, but not affecting the whole system.

- **Transparency**. Despite the redundancy of components, the possible heterogeneity and different location of the elements of the system, the ICTs must provide all the functionality to the user transparently, so it doesn't need to know the internal structure of the system to make it work.

- **Scalability**. The system must enable the addition of new components efficiently without any damage on the performance of the system. Also the latency of the addition of new components to the system must be reduced to the minimum, in order to enable the deployment of new components in the system.

- **Information Locality**. Although the information can be accessible from any point of the system, it must be under the control of the proprietary of the data.

3   Grid Technology

The Grid concept appeared at mid 90s, and it was revealed as a solution for some computational problems, enabling the execution of lengthy tasks in shared resources which were geographically

distributed. The main objective of Grid is to provide the user with the vision of a self managed powerful computer, built up of heterogeneous resources.

Joint to this concept appeared the Globus project [19]. It proposed to share resources owned by distinct organizations, in a transparent and ubiquitous way. The main result was Globus Toolkit 2.0 (GT2), which became the "de facto" standard for the deployment of computational Grids. The key application was to share and coordinate the resources owned by distinct entities for facing challenging projects related to high throughput in scientific fields.

Grid middlewares are evolving to find the convergence with the Service Oriented Architectures (SOA), and particularly with the Web Services [8]. This junction responds to the fact that the Web Services model is a modern technology, widely accepted and supported by developers and providers, for implementing applications oriented to the Internet. On the other side, the Grid technology provides with concepts and a philosophy which widens the scope in which Web Services are applied.

A result of this convergence is the Web Service Resource Framework (WSRF), which consists in an extension of the capabilities of the current Web Services technologies for the introduction of some features required by the Grid concepts (such as lifetime, state, etc.).

3.1   Integrating other resources

Since the appearance of Grid technology, it has been fostered by some scientific projects in which it has been needed to gather computing power or storage capacity. Some examples of these projects are Unicore [17], CrossGrid [1], Eurogrid [26], GEMMS [7] or EGEE [32]. Such fashion of resource gathering has introduced the terms "computing grids" and "data grids".

The decrease of the variety of virtual resources responds to the scope of the projects which have developed this technology. But, according to the principles introduced in [20], it is a simplification of the Grid concept. From the point of view of the Grid technology, it is important to widen the scope of the resources to be shared. There is a lack of protocols, mechanisms, etc. for discovering and using other services than computation or storage. Also there is a lack of ontologies about the management of these "other" services.

At this point, the success parameters diverge from those related to high performance. Currently, the main challenge is to provide the abstractions, methods and protocols which enable a wider usage of Grid technology.

4   gCitizen: Grid for eGovernment

The aim of the gCitizen project is to create a Grid middleware for the transparent management of the information about the citizens in the public administration, and the administrative procedures in which the citizens are involved. The vision of the project is outlined in the Figure 1.
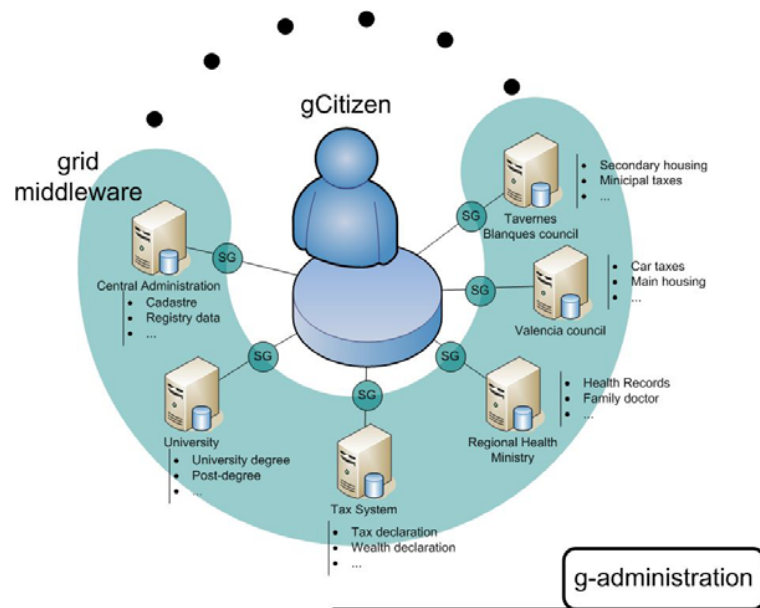

Figure 1: Vision of the project

This paper is Available online at
www.jtaer.com

One of the main aims of the gCitizen project is not being invasive within the current systems. This is a key issue when deploying the middleware in the government environments, because in these frameworks it is not possible to replace every application and start over. In this case, the gCitizen middleware coexists with the current systems.

The Grid technology has been selected because its characteristics properly fit with the eGovernment requirements. The selected running environment for the deployment is the Globus Toolkit 4 (GT4), which is an implementation of the WSRF standard.

This infrastructure also provides some of the technical features in which the gCitizen middleware is based:

- Security: GT4 provides security in the access and the transmission using the GSI layer. It also enables the authentication of users in the system, and thus enables the accounting procedures.

- Transparency: the WSRF standard provides protocols and interfaces to enable the access to the services, independently of the system where the services are deployed or the language used for their development.

- Ubiquitous access: The WSRF services deployed are easily accessible from any point of the network, as the technology is based on standard Internet addressing protocols.

- Scalability: The SOA enables the deployment of new components, without affecting the already existing services.

- Redundancy: The service oriented paradigm enables the deployment of several instances of the same service.

Moreover the SOA approach enables the improvement of the middleware, by creating new services which provide some additional features which are not provided by GT4.

Notwithstanding the Grid middleware provides features which cover some of the requirements for an eGovernment architecture, there are some lacks that must be covered in order to be applicable to this environments.

The gCitizen project completes the Grid middleware by adding new components and concepts, but also developing new services which are not provided by the GT4 layer. These modifications are based on the eGovernment requirements, but they can also be applied to other environments which have similar restrictions. An overview of the components of the gCitizen architecture is depicted in Figure 2.
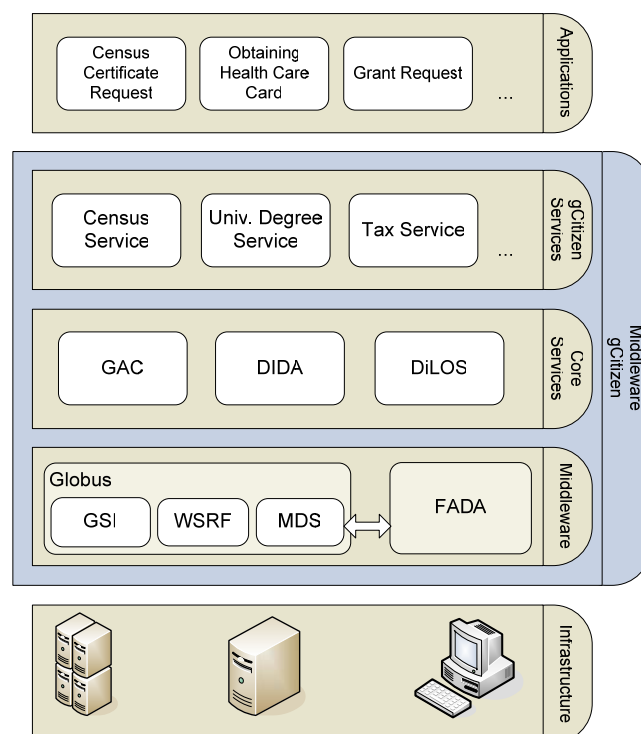

Figure 2: gCitizen architecture overview

This paper is Available online at
www.jtaer.com

4.1   General Addressing Convention

In a SOA, it is very important the way to refer to the services in the system, in order to find and to accede to them. The WSRF standard is based on the Uniform Resource Identifier (URI) [6] of the services in order to identify them. This method is location dependent, so a change in the deployment of the service would change its identifier. Also, two services are the same just because the user is conceived of it, but it is not possible guess the equivalence.

In this sense, the full qualified functionality of the services (the entity which provides it, the function performed, etc.) is used in gCitizen as the main "identifier" for the services. So if two services have the same functionality and are provided by the same administrative unit, they are indeed the same service. Using this kind of identifiers, the service naming system is location independent, because it does not apply any reference to the machine where the service has been deployed. The location independence feature also provides some important characteristics such as:

- Service Mobility: the services can change their location maintaining the same identifier. It enables the undeployment of a service from its location and its deployment in a new site (upgrading servers, maintenance, etc.).

- Transparent Redundancy: Different "physical" instances of the services are able to use the same identifier, in case they provide the same functionality. It enables a load balancing system, which may be implemented by the underlying resolving layer.

The gCitizen naming schema follows the approach of the LDAP Distinguished Names [24], [33]. The DN are currently used in grid environments to identify users and hosts, so it is the natural solution to use it for identifying the services. The fields of the DN used in the gCitizen schema are the following:

- Country Name (C): It represents the country which is the scope of the service, using the two letter code representation (ISO 3166).

- Region Name (A): This value stores the regional level of governance.

- State or Province Name (ST): This field represents the provincial level of governance.

- Locality (L): The locality refers to the local level of governance. It is equivalent to the city or town in some countries.

- Organization Name (O): This field shows the organization which is the main scope of the service.

- Organizational Unit Name (OU): This field represents the department or section which is under the scope of the service. As in standard DN, this field is likely to be repeated to enable the full representation of the different levels in the departmental structure.

- Common Name (CN): This field indicates the name of the service, and it may identify the functionality of the service in the organization qualified in previous fields.

An example of usage:
```
        /C=ES/A=C. Valenciana/ST=Valencia/L=Valencia
         /O=Ayto. Valencia/OU=Informatica/CN=Padron
```

The construction of the DN using this scheme fits with the hierarchical and departmental structure of the public administration, showing the location of the services in the hierarchy of the Administration.

In the implementation, the name of the service is integrated as a Resource Property, for enabling its access using the standard WSRF functions. It also enables searching for it in the GT4 Index Service where it may be added.

Currently, the OGSA Naming Working Group is working on the WS-Naming recommendation [21] that construct a location independent naming system. This architecture is based in the existence of resolvers, which are the responsible of translating the names into references. In contrast, the gCitizen system relies on existing components and the Discovery Architecture, which is described below.

4.2   Distributed Discovery Architecture

The discovery services in the current service oriented technologies are usually based in the maintenance of an index or directory service that gathers the information of all the services (i.e. WS using UDDI [12] servers or GT4 with the MDS4 [28]). Most of them provide node aggregation, replication of the information, grouping them hierarchically, etc.

GT4 provides a method for the service discovery, which is based on the Index Services (IS). These services enable to index other services, but also another IS, creating hierarchical organizations. The IS

This paper is Available online at
www.jtaer.com

exposes the Resource Properties (RP) of the registered services enabling the user to query for services, regarding to their RPs, using XPath [14].

MDS4 creates a hierarchical tree of IS. Therefore when the user tries to access all the information of the system he must contact the top level IS. This scheme introduces a central component which can act as a bottleneck or as a single point of failure.

The gCitizen middleware enables a distributed discovery system, which avoids these central points, but also provides with another features which are needed in an eGovernment framework:

- Ubiquitous access: It enables the access to the information from different points.

- Transparency: The complexity of the system is hidden to the user.

- Scalability: Enable to easily deploy new components.

- Fault tolerance: When a component fails, the system detects it and performs the needed operations to stabilize the system.

- Visibility: If a service is deployed in the system, an operation which tries to search for it, will success. This feature also enables the service mobility, service redundancy, and other features.

The DIDA Architecture [2] is built up on the Globus Toolkit 4 Index Services, adding a layer on top of the GT4 system, which connects the different ISs in the system creating a unique virtual server and thus avoiding cascade node organizations.

Previous works [30] have used P2P techniques, which uses contact servers, to create a view of a virtual server. In this architecture it has been used the Federated Advanced Directory Architecture (FADA) [31], which enables the creation of the completely distributed directory of IS services.

The key issues of the DIDA architecture are the topology of the network and the Distributed Discovery Services. Both are described below:

Network topology. Opposite to the standard P2P networks, the gCitizen discovery architecture proposes the usage of topologies which may be tested and corrected in case of problems, in order to increase the stability of the system. The DIDA architecture is based on an enhanced 2-Regular topology which considers the usage of two kinds of network links:

- Strong links: used to create a 2-Regular graph. It is implemented by forcing each FADA node to know at least 2 neighbours.

- Weak links: some additional links used to improve the performance and recovery of the network.

Using this kind of topologies, in case of failure of one node, the connectivity to the other components in the system is guaranteed. Failures may be detected, enabling the system to correct the problem and thus stabilizing the system.

Distributed Discovery Services. The entry points to the discovery system are created by deploying Distributed Discovery Services (DDS). These components are standard WSRF services which accept XPath for solving queries for searching for services. These queries are used by the DDS service for querying the IS indexed which are deployed in the FADA system.
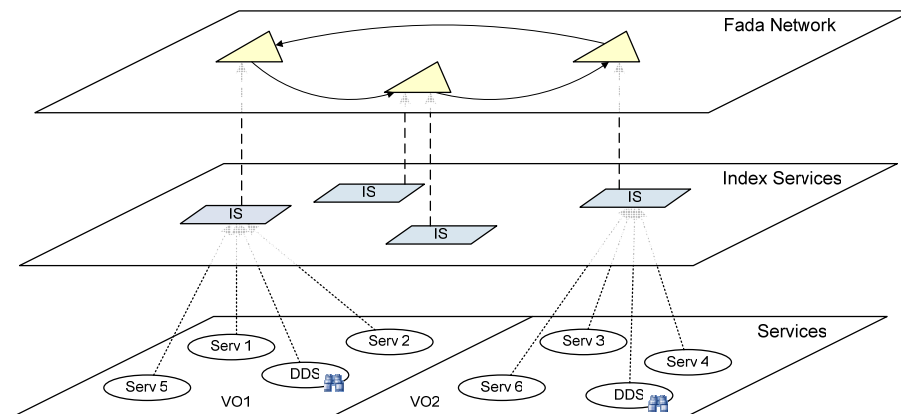

Figure 3: DiDA architecture scheme

This paper is Available online at
www.jtaer.com

The DDS are aware of the existence of FADA nodes, and uses them for discovering any IS which is deployed in the system. Later, each IS is queried using XPath and the results are gathered and returned to the client of the DDS. The DDS perform all the operations transparently to the user, hiding the complexity of the underlying system.

With the DiDA architecture described, the functional scheme of the system is the following:

1. The client sends the search request to its nearest DDS with the XPath query.

2. The DDS contacts with a FADA node to search all the ISs of the system.

3. The ISs search request is broadcasted to all the FADA nodes (using the FADA internal protocol).

4. The FADA node returns to the DDS the list of ISs.

5. The DDS contacts with each IS making the XPath Query.

6. At last the results are gathered to the client.

4.3   Distributed General Logging Framework

In an eGovernment environment any information involved in a transaction must be stored, providing the real "registry" function. In a similar way in Grid environments almost any operation is susceptible to be registered for its late analysis. Control of the access to the services, type of the resources used by someone, time of usage, changes in the state of a service, etc. are some examples of events which are candidate to be registered. Later it will be possible to assign a price to the resources, extract statistics of usage or guess who has misused a resource.

In an eGovernment framework the main resource to be shared is the information about the citizens or enterprises, owned by each of the administrative units. So in gCitizen as in many ICT projects, it occur some electronic transactions which treat with these data which are likely to be registered.

In an ICT environment, the usual procedure to register the operations is the creation of a log file which contains lines describing the activity. This model translated to the gCitizen project implies the registration of a huge quantity of information which would be hard to store or manage. The Distributed Log System (DiLoS) scatters logs through a Virtual Organization for obtaining features such as backup, redundancy, ease of access to logs and decentralization.

The DiLoS architecture is composed by two kinds of elements. On one hand, the gCitizen services which provide the logs and need to be integrated into the DiLoS architecture; on the other hand a specific service called "Log" which will coordinate the distributed information.

Every gCitizen service store their registries about the operations performed. Using the interfaces provided by the DiLoS architecture, the gCitizen services would send their local registries to the proper DiLoS log service.

The DiLoS system is inspired in the architecture proposed by Syslog [25], which is a system logging utility used on UNIX systems. DiLoS takes their concepts, and translate them into a Grid distributed environment.

Every Log service is associated to an administrative unit, which are grouped in levels, according to the governance dependences among them. A common configuration would deploy one DiLoS log service per unit. Nevertheless, the architecture considers the possibility of having more than one instance at each level for enhancing the performance, backup capacity, load balancing, etc. of the system.

Anyway, the service which needs to contact a DiLoS log service would need to discover the appropriate instance. At this situation, it will use the configured DDS from DIDA, and will form a query for the log service at the appropriate organizational level, according to the GAC naming convention. Once the service is discovered, the implementation of the service will be the responsible of deciding the instance to which contact for sending the data.

The DiLoS log services assigned to a unit are able to transfer their data to the ones in the upper units, in order to perform an integration of the data for accomplishing the laws, backups, etc.

At the end, each administrative unit is the responsible of deciding whether to send logs to upper levels or not, according to its governance policies. So a unit is able to isolate a set of log data from upper levels where this information is not important. Figure 4 summarizes the functional scheme of the DiLoS architecture.

This paper is Available online at
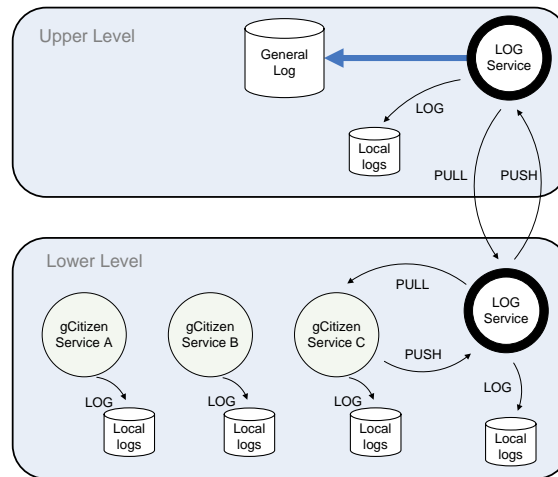www.jtaer.com

Figure 4: DiLoS functional architecture

The interface of the services which are integrated in the DiLoS architecture (log services and gCitizen services) contains four operations. A brief description of these functions is shown below (more detailed description in [3]):

- PULL: It is a public method included in every gCitizen service. It is invoked by a DiLoS log service which calls for log registries.
- PUSH: It is a public method implemented by the DiLoS log services. It is invoked by other service when they need to transmit its registries to the log service in an upper level.
- LOG: It is a private method which is implemented by the gCitizen services. It is called by the service itself to locally save a log. Later it would be transmited to DiLoS log services using PUSH operations.
- QUERY: This public method returns a set of logs which accomplish with a pattern (date, user, name service).

4.4   gCitizen Data Model

The key issue when dealing with the integration of the information is the data model which is going to be applied. As seen in previous sections, there is lack of a common framework for the eGovernment in which data models should have been created.

Up to now, the exchange of information among the different administrative units or between the administration and the citizen does not use a standard format. There are several projects and initiatives IDABC [13], SAGA [23], OASIS eGovernment TC [10] which are working on the definition of a standard format, in order to improve the interoperability of the information systems in the administration. Most of them only propose a set of rules for promoting the communication among the different participants in the eGovernment processes. But any of them agree in using XML [9] as the language for the exchange of information.

The UK Gov Talk project [15] in the United Kingdom has proposed e-GIF (e-Government Interoperability Framework) for the exchange of information between the administrations and the public sector. It consist in a set of XML documents and XML schemas (XSD) [16] which define a set of basic elements and their appropriate types, called Government Data Standards (GDS).

In the gCitizen project it has been proposed a data model which is based on the e-GIF definitions, but adapted to the particular needs of the Spanish government and the applications which this government manages.

The data model is structured in different parts, each one referring to some field of the life of a citizen. Furthermore it has an identification part that is used by the services of the gCitizen system to identify the citizen with which corresponds the associated data. The part of the model dedicated to the identification is composed by a set of essential information (such as name and surname, the date of birth and the main address) about the person who the model represents.

In order to develop some test applications, other sections have been initially included in the data model:

- Census: all the census data stored by the administration about a citizen.

- Contact: further contact information about the citizen such as telephone numbers, emails, addresses, etc. which may be provided by the internal databases of the services.

- University degrees: the information about the different degrees obtained the citizen at any university.

The use of XML as interchange format enables easily to evolve the model, including new sections or adding new elements in the existing sections.

4.5   Plug and Play for eGovernment Services

The gCitizen architecture has been designed to enable the services to be used immediately once they are deployed in the system. In this sense the services must behave as *plug-and-play* components.

The discovery architecture provided with gCitizen guarantees that any service which has been deployed in the system can be found (or it notifies an error). Usually it is needed to know the functionality of the services in order to use them. The gCitizen middleware provides interfaces and ontologies so that the services are self-contained, and provide information about themselves. These properties enable its usage use without an a priori knowledge.

The key issues which enable this behaviour are described below:

Information Properties. Each service publishes WSRF properties that will enable the user to obtain information about its functionality. There are two kind of properties published: the DN of the service in the GAC addressing system, and the "Information Sections" managed by this service.

The name used in the GAC addressing system provides information about the semantics of the services: the hierarchy of the name provides information about the entity which provides the service, and CN identifies the function provided.

Other information that every gCitizen service publishes is the "Information Sections". These RPs show what sections of the gCitizen data model are managed by the publishing service. These properties enable the users to find the services which provide the information related to the data model section which they need.

This information will enable to find services with the required functionality without a priori knowledge of them.

Functional Properties. Every service in the gCitizen framework implements a set of minimal functions, using a specific interface. The semantics of the specified functions are the same for every service, but the implementation is adapted by each service to the specific data and the semantics of the component. These functions enable the operation with the services without a specific knowledge of the services.

The interface proposed contains the following set of basic functions, whose semantics are defined in the architecture: Insert, Read, Delete, Modify, Validation, Subscription (to notifications), and Cancellation (of subscription).

The input and output parameters for any of these functions will be a XML fragment of the gCitizen data model. To enable the proper use of these functions without a priori knowledge of the model, the system will enable a process of discovery of this data model. Each service implements two functions which provide an XML document containing the input and output scheme for each implemented function. So, the aggregation of every XML fragment obtained from the services will create a complete vision of the data model. It also enables the natural evolution of the data model, and thus the scalability of the system.

To use a gCitizen service without an a priori knowledge of its functionality the user must follow the next steps:

1. Search the service using the Discovery System: To find the suitable service the user can use the "Information Properties" of the service. These properties are published using XML, enabling the user to make the query using XPath.

2. Select what standard functions is intended to be used.

3. Query the service about the sections of the data model needed as input for the selected function.

4. Complete the input fragment with the available data.

5. Call the selected function using the XML generated.

6. Get the results (also in XML).

This paper is Available online at
www.jtaer.com

Furthermore, each service is also likely to implement specific operations, which improve its functionality. Nevertheless, in order to use these functions, the users must know the particular implementation of the service.

5   Services and Applications Developed

In order to demonstrate the functionality of the gCitizen architecture, it has been deployed a testbed which contains most of the components and services described in this document. These services and components are hosted by several units in the Public Administration. They get the data from the original applications or services previously developed by these units.

There have also been developed some applications which implement distinct administrative procedures involving some of the services deployed by the administrative units. These applications have been created in the form of web portals, and they are hosted by external entities in order to demonstrate the feasibility of the integration model and the architecture itself.

One of the test applications is the coordination of the update of the citizen registries. The citizen registry contains data about the citizens which are associated to each municipality. The problem arises as each municipality manages its own registry using its own mechanisms (specific applications, ad-hoc databases, etc.). Also there are entities which manage the registries of several municipalities.

In this use case, the main gCitizen service used is the so called "Padron" (using the Spanish name). The service has been deployed by some significant entities, which represent most of the levels of governance involved in the management of the registry. The organizational structure of the deployment has been done according to the governmental structure of these entities. The services are thus deployed as follows:

- The citizen registry of Cullera municipality at Cullera.

- The citizen registry of Marines and Tavernes municipalities at the Provincial Government of Valencia

- The citizen registry of Naquera, at the Universidad Politécnica de Valencia (UPV).

The services cover a wide variety of use cases. Besides the use cases of the self managed registry (Cullera), and those which are delegated to an upper level of governance (Marines and Tavernes), it has been deployed a service at the UPV, in order to demonstrate an external hosting.

Any of the Padron services deployed interact with the databases managed by the applications already used for the current management of the citizen registry. The civil servants are able to work with their existing applications, but the data which is being managed is also acceded by an upper level.

Further than the eGovernment services, there have been deployed some gCitizen infrastructure services. These services are used to build distributed architecture which integrates the services which provide the eGovernment facilities. The middleware testbed deployment involves the regional government of Valencia ("Generalitat Valenciana", GVA), the provincial government of Valencia ("Diputación de Valencia", DVA) and the Technical University of Valencia (UPV).

Although the number of nodes is low, they are representative of a general deployment as they involve some of the most important public administration in the region of Valencia. The infrastructure is made of a FADA deployment (used by the DiDA architecture), and some Globus Toolkit based nodes (in order to deploy the DiLOS services, the Index Services, and the gCitizen services).

The gCitizen infrastructure is composed by the next nodes:

- One Windows node (WU) as a FADA Server, in which there have been installed Windows XP Pro, FADA 5.2.6.1, Java 1.5.0 01. This node is deployed at the UPV network.

- One Linux node (LG) as a FADA Server, which also contains a GT4 installation with MDS support. This node is deployed at the GVA network.

- Two nodes (LU and LD) with Linux Fedora Core 3 and Globus Toolkit 4 (Version 3.9.5) installed. These nodes are deployed at the UPV and DVA networks.

Each of the institution has deployed the DDS service in their GT4 nodes, in order to have three different services which provide access to the DiDA system. These points of access enable a structured access to

the service search system, as they allow the queries to the granted services and applications (according to the governmental structure).

According to the naming convention, the Padron services deployed in the system are named as follows:

```
 /C=ES/A=Valencia/ST=Valencia/L=Marines/O=Ayto. Marines/OU=Gestion/CN=Padron
 /C=ES/A=Valencia/ST=Valencia/L=Tavernes/O=Ayto. Tavernes/OU=Gestion/CN=Padron
 /C=ES/A=Valencia/ST=Valencia/L=Cullera/O=Ayto. Cullera/OU=Gestion/CN=Padron
 /C=ES/A=Valencia/ST=Valencia/L=Naquera/O=Ayto. Naquera/OU=Gestion/CN=Padron
```

These names are used in the DiDA search system, in order to locate a physical deployment of the services, and interact with them.

According to this distribution, one particular execution of the pilot general application created to implement the administrative procedure of moving one citizen from one municipality to another, follows the next steps (shown in Figure 5):

1. Select the XPath query to search in the DiDA system for the Padron from Tavernes. The XPath query should be similar to:

    ```
        "//ns5:Entry[/ns5:Content//tns:SID/CN='Padron' and
     /ns5:Content//tns:SID/L='Tavernes']/ns5:MemberServiceEPR/ns6:Address"
    ```

2. Contact to the nearest DDS service (in this case, the one deployed at the LU node), and send the query to it (the client needs to know the physical reference of the DDS service, as it is part of the infrastructure).

3. The DDS service will use the DiDA system to obtain the results of the XPath query made from any IS deployed in the system.

4. In parallel, repeat the three previous steps in order to get the reference to the Padron service from Cullera. In this case, the XPath query is very similar to the previous one.

    ```
        "//ns5:Entry[/ns5:Content//tns:SID/CN='Padron' and
     /ns5:Content//tns:SID/L='Cullera']/ns5:MemberServiceEPR/ns6:Address"
    ```

5. The result of the queries will be the physical references to the two Padron services.

6. Using the physical references the client application (web portal) will contact the services for using it.

7. The client application gets gCitizen data model generated by the Padron from Tavernes, using the basic data of the citizen (the data fields corresponding to the identification part of the data model).

8. The client application deletes the reference to the citizen which has been got in the previous step, using the gCitizen standard "Delete" operation.

9. The client application inserts the data model obtained in the previous steps in the Padron from Cullera, using the gCitizen standard "Insert" operation.

This paper is Available online at
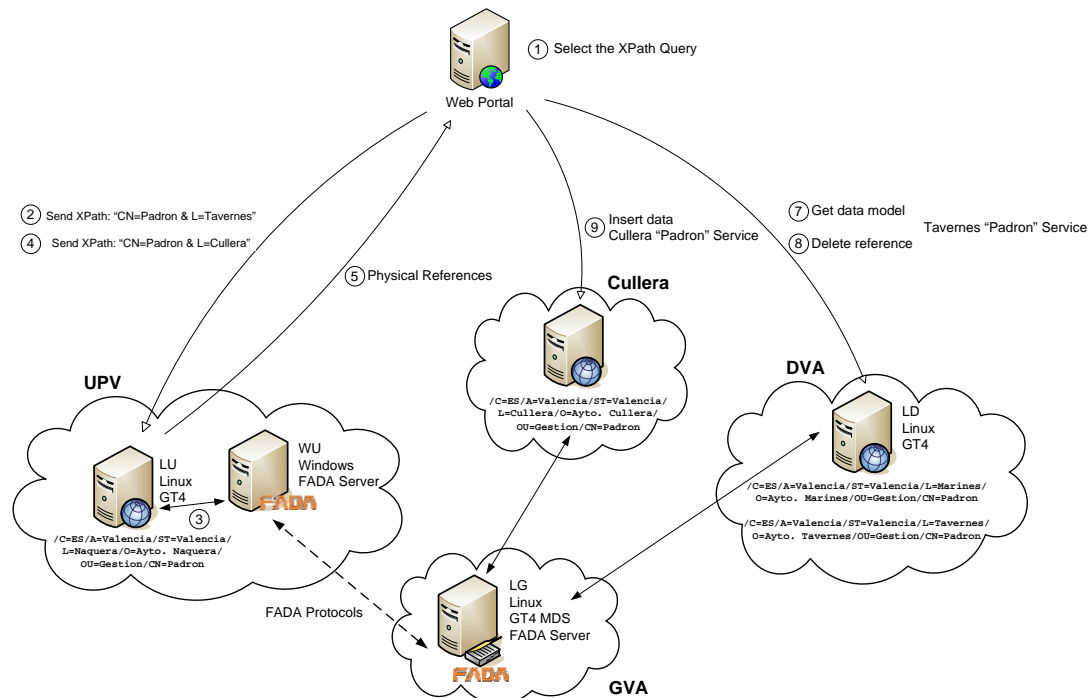www.jtaer.com

Figure 5: Testbed Scheme

As it has been described in the previous sections is easy to deploy new services to the system, and use them immediately. There are two possibilities: if the service is owned by an existing organization it must be deployed in one of the GT4 running nodes and thus registering it in the Index Service of that node. It is also possible to install new certified GT4 nodes in which deploy the services, and register it in the Index Service. The Index service must be associated to any of the FADA nodes, in order to be found by DiDA and thus being ready to use.

6   Conclusions and Further Work

Grid technology has focused in sharing computing resources and storage space, although its concepts aim at sharing more heterogeneous resources which are further than these from the scientific domains. Currently there is a lack of protocols or methods for dealing with "other" resources.

Even applying the concepts from the Grid technology, the success parameters introduced in common Grid applications (performance, speed-up, number of CPUs, etc.) are not applicable in this approach. In contrast, the aim is to provide protocols and standards which enable the creation of global collaborative environments.

In this paper there are described a set of components which provide features such as (1) visibility, for determining whether a service has been deployed or not; (2) mobility of services, which ease the enhancement of the infrastructure, fabric scalability, etc., but also enable the creation of ad-hoc infrastructures based on itinerant services provided by mobile devices; or (3) redundancy of services, which enable the component scalability, load balancing, etc.

Some of these characteristics are enabled by the discovery architecture, which is based on a stable topology which may be tested and corrected in case of failure. Notwithstanding its benefits, the current version of this architecture presents scalability problems which would affect the performance in case of the system grow in excess. In this sense, some other topologies and caching techniques are to be studied in order to enhance its scalability.

Although applying the proper technologies for the eGovernment, the key component is the definition of a flexible and interoperable data model, and the mapping rules from the existing data models to the agreed one. As described in previous sections, the ICTs were only applied at an internal level, in the public administration. In this sense the interoperability issues were not as important as they have been currently revealed. The gCitizen project exposes the feasibility of creating upper level interoperable frameworks which make use of the existing governmental resources.

Further than the interoperable issues, the gCitizen middleware proposes an attempt of facing the problem of the eGovernment as a whole. Up to now most of the initiatives have focused on solving emergent and urgent problems. The addressing of such new issues isolates parts of the whole problem and increases the distance among the existing partial solutions. Any of these problems is only a small part from a whole whose succeeding approach passes through the integration of the different existing parts, and the reconsideration of some of them.

There have been recently created some interoperability initiatives such as the European Regional Network for Interoperability (ERNI) [27] which try to enhance the electronic interchange of the information among the governmental units. These groups have gathered the expertise of the different stakeholders and the succeeding initiatives (such as gCitizen) and try to establish a Common Interoperability Framework for the eGovernment.

References

[1] ACC CYFRONET AGH-UST. (2000), The CrossGrid Project Website. [Online]. Available: http://www.eu-crossgrid.org/project.htm
[2] C. Alfonso, M. Caballer, and V. Hernández, DIDA, a Distributed Discovery Architecture for Grid Environments, in Proceedings Cracow Grid Workshop, 2005, pp. 106–113.
[3] C. Alfonso, M. Caballer, José V. Carrión, and V. Hernández, Distributed General Logging Architecture for Grid Environments, in Proceedings VECPAR 2006.
[4] C. Alfonso, M. Caballer, José V. Carrión, and V. Hernández, gCitizen: Using Grids for Creating an Integrated View of the Citizen in an eGovernment Framework, in Proceedings of the Fifteenth IEEE International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises, 2006, pp. 233–238.
[5] M. Bangemann. et al. Europe and the global information society, Bangemann report. European Commission, High-Level Group on the Information Society, Technical Report, [Online]. Available: http://europa.eu.int/ISPO/infosoc/backg/bangeman.html
[6] T. Berners-Lee, R. Fielding, L. Masinter. Uniform Resource Identifier (URI): Generic Syntax. RFC 3986, Internet Engineering Task Force (IETF). 2005.
[7] G. Berti. (2002). Grid Enabled Medical Simulation Services (GEMMS) [Online]. Available: http://www.ccrl-nece.de/gemss
[8] D. Booth, H. Haas, F. McCabe, E. Newcomer, M. Champion, C. Ferris, D. Orchard (2004), Web Services Architecture, [Online]. Available: http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/
[9] T. Bray, J. Paoli, C. M. Sperberg-McQueen, E. Maler, F. Yergeau (2006), Extensible Markup Language (XML) 1.0 (Fourth Edition), W3C Recommendation, [Online]. Available: http://www.w3.org/TR/REC-xml/
[10] H.J. van Burg (2002), OASIS eGovernment TC Website [Online] Available: http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=egov
[11] CERN. (2002). LHC Computing Grid Project Website. [Online]. Available: http://lcg.web.cern.ch/LCG
[12] L. Clement, A. Hately, C. von Riegen, T. Rogers, UDDI Version 3.0.2 Specification, [Online]. Available: http://uddi.org/pubs/uddi_v3.htm.
[13] E. Commission. (2005), IDABC Interoperable Delivery of European egovernment services to public Administrations, Business and Citizens Website. [Online]. Available: http://europa.eu.int/idabc/
[14] S. DeRose and J. Clark (1999), XML Path Language (XPath) Version 1.0, W3C Recommendation, [Online]. Available: http://www.w3c.org/TR/xpath
[15] e-Government Unit of Cabinet Office. (2001). Uk Gov Talk information on policies and standards for e-government Website. [Online]. Available: http://www.govtalk.gov.uk/
[16] D. C. Fallside, P. Walmsley (2004), XML Schema Part 0: Primer Second Edition, W3C Recommendation, [Online]. Available: http://www.w3.org/TR/xmlschema-0/
[17] U. Forum e.V. (1997), The Uniform Interface for Computing Resources. [Online]. Available: http://www.unicore.org
[18] I. Foster, Globus Toolkit Version 4: Software for Service-Oriented Systems. IFIP International Conference on Network and Parallel Computing, Springer-Verlag LNCS 3779, pp 2–13. 2005.
[19] I. Foster and C. Kesselman. Globus: A metacomputing infrastructure toolkit, vol. 11, no. 2, pp 115–128, 1997.
[20] I. Foster and C. Kesselman. The Grid: Blueprint for a new computing infrastructure. Morgan Kaufmann Publishers, San Francisco, USA. 1999.
[21] A. Grimshaw and M. Pereira. (2005), OGSA naming wg Website. [Online]. Available: https://forge.gridforum.org/projects/ogsa-naming-wg
[22] L. Hooghe and G. Marks.Multi-Level Governance and European Integration, 1st ed. Rowman & Littlefield Publishers, Lanham, Maryland. 2001.
[23] KBSt. (2003), SAGA: Standards and Architectures for eGovernment Applications Website [Online] Available: http://www.kbst.bund.de/saga
[24] S. Kille. A string representation of distinguished names. RFC 1779. Internet Engineering Task Force (IETF). 1995.

[25] C. Lonvick. The BSD Syslog protocol. RFC 3164, Internet Engineering Task Force(IETF), 2001.
[26] D. Mallmann. (2000), Eurogrid Website. [Online]. Available: http://www.eurogrid.org
[27] Regional Ministry for Infrastructures and Transport, Generalitat Valenciana (2006), esiig Red ERNI [Online] Available: http://www.esiig.gva.es/index.php?option=com_content&task=view&id=79&Itemid=66&lang=en
[28] J.M. Schopf, M. D'Arcy, N. Miller, L. Pearlman, I. Foster and C. Kesselman (2005), Monitoring and discovery in a web services framework: Functionality and performance of the globus toolkit's MDS4, Preprint ANL/MCS-P1248-0405, [Online]. Available: http://www-unix.mcs.anl.gov/~schopf/Pubs/mds4.hpdc06.pdf
[29] D. Snelling, I. Robinson and T. Banks. (2004), OASIS Web Services Resource Framework (WSRF) TC Website. [Online]. Available: http://www.oasis-open.org/committees/wsrf
[30] D. Talia and P. Trunfio. Web services for peer-to-peer resource discovery on the grid. DELOS Workshop: Digital Library Architectures, pp 73–84. 2004.
[31] Techideas. (2005), Federated advanced directory architecture Website. [Online]. Available: http://fada.sourceforge.net/
[32] TERENA. (2004), Enabling Grid for E-sciencE Website. [Online]. Available: http://public.euegee.org
[33] W. Yeong, T. Howes and S. Kille. Lightweight directory access protocol. RFC 1777, Internet Engineering Task Force (IETF). 1995.