

A Data Engine for Science Gateways: enabling easy data transfer and sharing

Mr. Riccardo Rotondo¹

Consortium GARR

Via dei Tizii, 6 – 00185 Rome, Italy

E-mail: riccardo.rotondo@garr.it

Dr. Marco Fargetta

Consorzio COMETA

Via S. Sofia, 64 – 95123 Catania, Italy

E-mail: marco.fargetta@ct.infn.it

Prof. Roberto Barbera

Department of Physics and Astronomy of the University of Catania and

INFN – National Institute of Nuclear Physics, Division of Catania

Viale A. Doria, 6 – 95125 Catania, Italy

E-mail: roberto.barbera@ct.infn.it

Grid infrastructures allow users to access and use computational and storage facilities distributed in different locations around the world. Science Gateways (SG) are recently emerging as high-level web environments that ease Grid access and use. However, although SGs simplify job management, providing a web interface by which jobs can be submitted with a few “clicks”, their capabilities to do an effective and intuitive data management are still at an early stage. Actually, Grid storage elements use dedicated protocols not supported by common applications such as web browsers. This makes a smooth integration of data management services in a Science Gateway quite difficult. In this work we show the Data Engine, a new SG component providing users with the ability to move data to storage elements and share them in an easy and clever way. The component operates between the Grid storage and the user, providing a file-system-like experience.

The International Symposium on Grids and Clouds (ISGC) 2012

Academia Sinica, Taipei, Taiwan

February 26 – March 2, 2012

¹ Speaker

² The MyWorkspace area in the Science Gateway portal consists of a full set of page with several portlet providing all the functionalities needed to the user for the Grid Interaction

1. Introduction

Nowadays, information and technology innovations run mostly on the web. The idea of owning a private server farm buried under a basement is starting to become obsolete and technology achievements have broken old schemas where all resources were stored on a central unit with the results that data are not anymore bound to a site but free to move.

In the past, scientists were forced to work near the place where computing resources were hosted and data stored. Now, the wide diffusion of large-bandwidth networks allows remote access of digital resources wherever they are kept by the appliances that most fit users' needs.

The increased adoption of web-based technologies has triggered the creation of dynamic environments providing services and resources to users. In the Grid infrastructure context, domain-specific portals, known as Science Gateways, have started to spread. They are used not only as resource aggregators but also as collaboration environments able to overcome and encompass physical and institutional boundaries enabling a fully web-2.0 surfing experience.

With the adoption of this new paradigm, several institutions and research entities have focused their efforts in order to conjugate services offered by distributed platforms like Grids with social functionalities available in Science Gateways. Several software modules have been written in order to customise and enable such portals with applications able to use the wide computational power offered by a Grid infrastructure.

In this paper we present the Data Engine, a web environment based on standards and characterised by easy access and simple usage. The paper is organised as follows. In Section 2, the Science Gateway paradigm is presented and the reference model used to create our framework is discussed. Section 3 presents the technologies used to create the Data Engine while Section 4 describes user workflow and experience in a realistic use case. Conclusions are then drawn in Section 5.

2. The Science Gateway paradigm

The introduction of the Grid paradigm more than 10 years ago was a big revolution in the access and use of huge computational resources either in terms of storage and CPU. The related technologies were widely adopted among the most important research centres around the world.

Aside the large adoption in research infrastructures, the Grid has partially failed to reach the wide research communities it was intended to from its providers. In fact, many aspects of the Grid paradigm are too complex, such as the Virtual Organisation (VO) concept, the strong certificate-based security infrastructure, the many middleware implementation with general lack of standards, the command line based interface and so on. As a result, non-IT-experts users give up losing all the benefit they could obtain from this new technology. On the other hand, this complexity is crucial to implement a trust among the resource providers that share their facilities and cannot be circumvented just to ease user interaction.

To hide the complexity of Grid infrastructures and widen their usage to virtual research communities not used to dig into middleware internals, a new paradigm has recently arisen which aims at creating an application environment where users can access the infrastructures

without having to cope with their complexities. This new paradigm, named Science Gateway (SG), was initially defined in the context of *TeraGrid* project [1], [2] as follows:

The Science Gateway paradigm has been introduced to overcome the difficulties met by users of a Virtual Research Community to access the provided tools. Its main goal is to hide the underlying complexity of a system, which gathers together heterogeneous resources, and prevent undesired access to non-authorized users. In more detail, a Science Gateway is defined as “a framework of tools that allows scientists to run applications with little concern for where the computation actually takes place. This is similar to cloud computing in which applications run as Web services on remote resources in a manner that is not visible to the end user. However, a science gateway is usually more than a collection of applications. Gateways often let users store, manage, catalogue, and share large data collections or rapidly evolving novel applications they cannot find anywhere else. Training and education are also a significant part of some Science Gateways.

A Science Gateway should provide users with a set of well-defined web-based applications properly configured to run in a Grid infrastructure and takes on behalf of the user all the steps needed to authenticate and authorise him/her.

The simple idea behind the paradigm of Science Gateway is quite difficult to implement in all the aspects and some compromises between simplicity for the users and provided features are unavoidable. These compromises make the differences among different implementation of the Science Gateway paradigm.

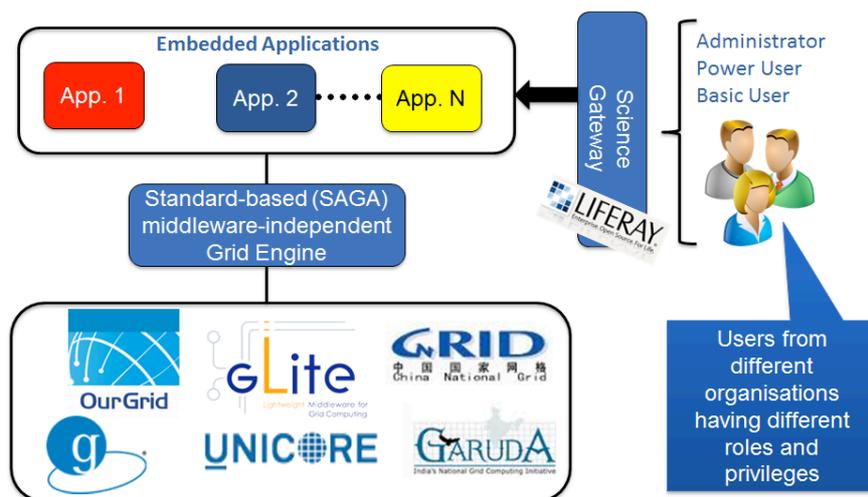


Figure 1 – The Science Gateway reference model

At a higher level a Science Gateway can be represented like in figure 1. The Science Gateway provides its user community, belonging to different organisations and having different

roles and privileges, with the access to domain/project-specific applications which can execute in a Grid infrastructure. A software layer, called the Grid Engine, takes care of job and data management and user tracking in a middleware-independent way.

2.1 The Reference Model

The work reported in this paper has been used to create a group of Science Gateways developed in the context of different projects and initiatives at continental and inter-continental level: DECIDE [3], EarthServer [4], EUMEDGRID [5], GISELA [6], and INDICATE [7]. These SGs, developed by our group from a year, share the same model, inspired to the one depicted in figure 1, with differences limited only to the target audiences and the provided applications.

The web interface of the Science Gateway is provided by Liferay [8]. This is a modern award winning framework for web portals supporting the “portlet” standards JSR168 and JSR286 and the most important web 2.0 technologies. The use of portlet standards allow to develop each application as a portlet and deploy it in different Science Gateways in a straightforward way.

For the interface towards the Grid infrastructure, a software layer has been developed. This API aims at hiding the complexity and make the system interoperable with different middleware and infrastructures. This layer, named Grid Engine, exploits the SAGA OGF standard [9-10] to interact with the Grid and provide a light and easy API to portlet developers. The implementation is based on JSAGA [11], a Java library implementing the SAGA standard, and on the Cryptographic Token Interface Standard (PKCS#11) [12] for the management of robot certificates and their proxies [13].

Therefore, to port an application inside the Science Gateway, developers have just to follow only the portlet standard and use the Grid Engine API, avoiding all interactions with the underlying Grid middleware.

In the rest of the paper we will describe the Data Engine, the component of the Grid Engine devoted to data management.

3. DOGS: Data On Grid Services

Service’s architecture has been designed in order to meet all requirements listed in the previous section. We designed components decoupling the backend from the frontend so to reduce the complexity of the implementation. Even if the interfaces are meant to intercommunicate, the wide adoption of standards makes easy to customise the system with different components.

As shown in figure 2, at the backend level the Data Engine is responsible for collecting requests from the front-end in order to allow access to Grid Storage Elements. Since Grid infrastructures require a strong authentication mechanism to provide their functionalities, all the commands exchanged are based on a proxy generated from a robot certificate. While the backend delivers requests to Grid, the frontend is responsible for exposing to the user an easy

and intuitive, file-system based, web interface. Users can manage data through the Science Gateway so they have not to access them from a command-line enabled Grid User Interface. The main aim of the frontend is to hide the underlying complexity and provide users with the possibility to move/copy their files on the Grid in the same way they move data from a folder stored on an local drive to another. In the following subsections, from the lower to the upper level, we will show in detail the technologies used to realise DOGS (Data On Grid Services), the main component of the Data Engine.

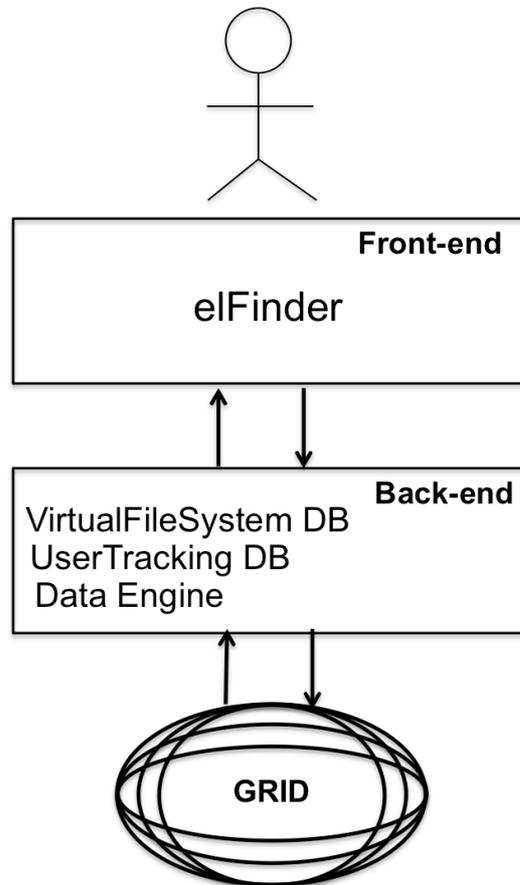


Figure 2 – Schematic view of the Data Engine

3.1 SAGA and JSAGA

Among the different ways to make an application interfacing Grid services in a user-transparent way, we adopted the Simple API for Grid Applications (SAGA), a standard defined by the Open Grid Forum (OGF). The standard defines abstract methods to access Grid resources in a middleware-independent way and perform job and data management. The success of the SAGA standard is underlined by several practical implementations developed for different programming languages. As a result of the portal framework chosen, totally written in Java, we decided to build the Data Engine using JSAGA, a Java implementation of SAGA [11]. The Data Engine powered by JSAGA is the component located at the lower level of the backend. It is

responsible for all the interactions with Grid Storage Elements and its calls and implementation are totally hidden to the end users.

3.2 Hibernate and the Database Model

Data engine is the DOGS module responsible for the access to Grid services and it is a fundamental part of it. The Data Engine realises a direct transfer between the Science Gateway and the Grid Storage Elements. In order to simulate and offer a file system like interface to end users, DOGS includes additional modules. Grid is only *where* files will be stored but from a user point of view this is as important as *how* files are organised and listed. A good data service is the one that provides users with the possibility to arrange files in folders ordered in a tree in the same way a real file system does on a physical disk. A database consisting of 3 main tables, represented in figure 3, achieves this task. Every operation on a file such as upload, copy, move or delete implies adding or editing some entries in the correct table. When users simply need to browse the file tree, in order to obtain the list of files, all the information stored in the database is used to create the user view of the file hierarchy. A good solution to manage databases in Java is Hibernate [14], an object-relational mapping (ORM) library providing a framework able to realise a mapping between MySQL tables and Java objects. At the base of the success of this framework is the independence from the Relational Database Model (RDBM) used. DOGS uses MySQL but, thanks to Hibernate, we could attach whatever database model.

Database tables related to the virtual file system are not the only ones used. The European Grid Infrastructure Portal and User Traceability policies require to track every single Grid transaction performed by the Science Gateway. Since all actions done by the Data Engine use same proxy (issued by the robot certificate), the DOGS database keeps trace of files' owner in order to keep private files uploaded by users while a users' tracking database stores other information such as the IP address associated to the user who is connecting to the Science Gateway.

3.3 elFinder

The components described in the previous sub-section act behind the scenes. A good service for the user is the one able to offer a dynamic, simple and intuitive interface, such as the one a user is used to on his/her personal computer. The most used web technology to build user interfaces is Javascript, a scripting language executing in web browser and interacting with the users to provide a richer user experience. So,, we choose elFinder [15], an open-source web file browser that uses Javascript and jQuery UI, as the frontend for our service. Every click on its interface calls a specific triggers an action in the backend. Part of the work has consisted in the integration of the elFinder connector with the corresponding Data Engine actions managing the files on the Grid.

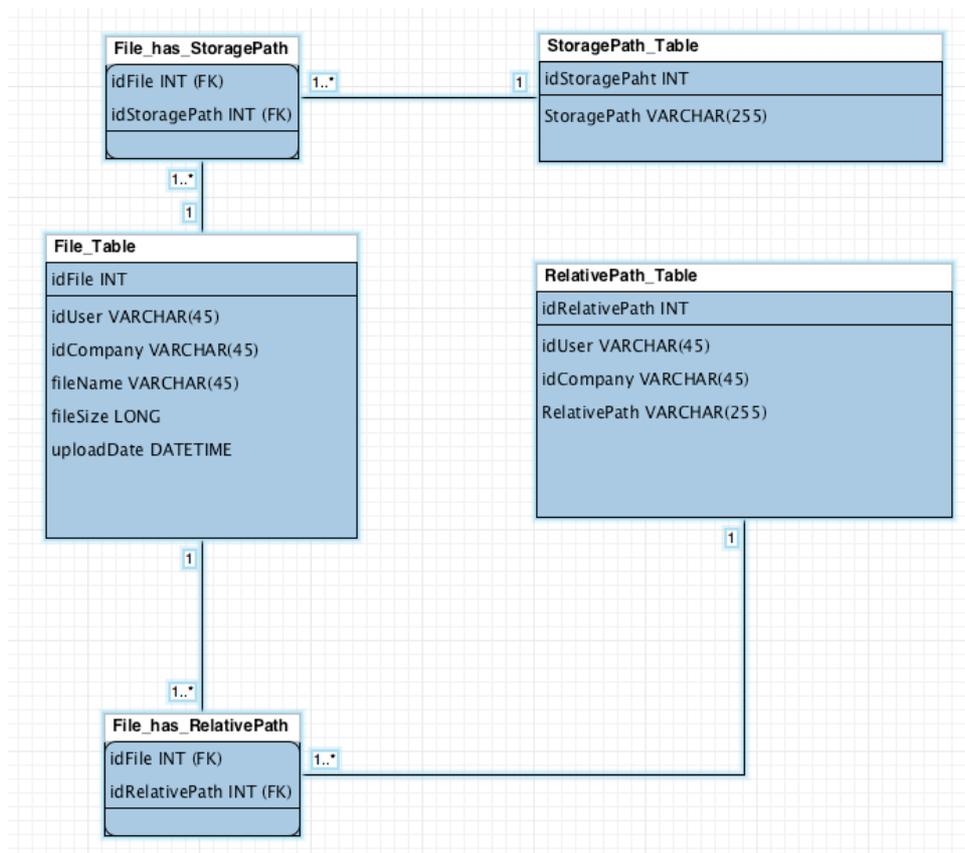


Figure 3 – Structure of the database implementing the virtual file system of the Data Engine

4. User Experience

In this section we describe, as an exemplar use of the Data Engine, the workflow done by users to access one of our Science Gateways and upload some data on a Grid Storage Element.

As explained in [16], all our portals are configured as Service Providers registered in different Identity Federations whose Identity Providers (IDPs) are trusted. Users will grant access to portal's resource only at the end of an authentication and authorisation process. The former is demanded to user's IDP while the latter to an LDAP server connected with the Science Gateway.

Once users have successfully signed-in, according with their roles and privileges they are allowed to perform operations on the Grid like running an application, accessing their files, etc.. All these functionalities are available through the *MyWorkspace portlet* in *MyWorkspace area* of the Science Gateway portal² visible only to logged users and sketched in figure 4. Figure 4 shows as well the components working behind the scenes when users try to upload a file. The first call made by the Data Engine is addressed to the eTokenServer [13], the service where robot certificates are stored, responsible for the issuance of the proxies needed to sign Grid transactions. After proxy generation, the Data Engine will transfer files, first to the portal and

² The MyWorkspace area in the Science Gateway portal consists of a full set of page with several portlet providing all the functionalities needed to the user for the Grid Interaction

from there to a Grid Storage Element according to the VOMS (Virtual Organisation Membership Services) extension of the proxy. In the meanwhile, all the information related to the file as owner, size, storage element(s) selected to store it, and location on user's virtual file system will be stored in the DOGS database while the other tracking information regarding user's IP address or timestamp of the transaction will be stored in the UserTracking database.

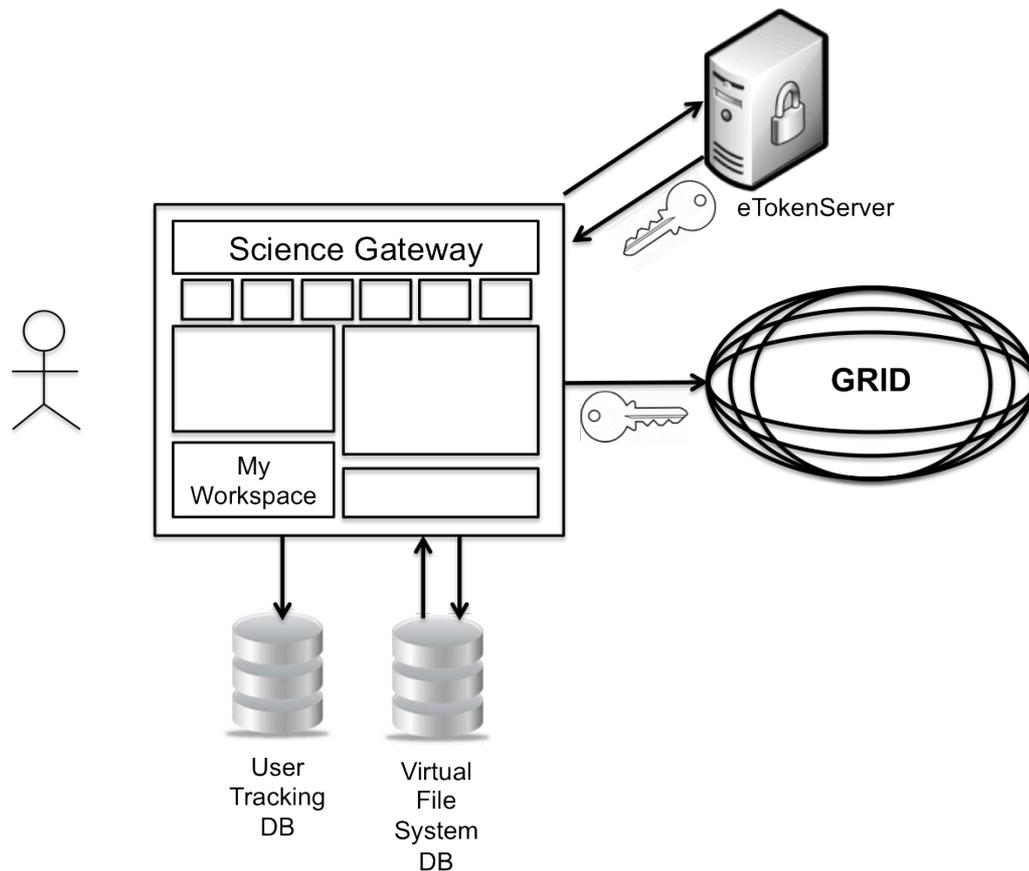


Figure 4 – Sketch of the MyWorkspace panel and its functions

The user interface based on eFinder is shown in figure 5. It takes clear inspiration from the file browser of a desktop/laptop operating system and it includes all functionalities that users expect to find in a file management environment, such as the possibility to have some commands associated to the right-click on the mouse, or the functionality to recognise file extensions and automatically organise files in folders listed in a tree.

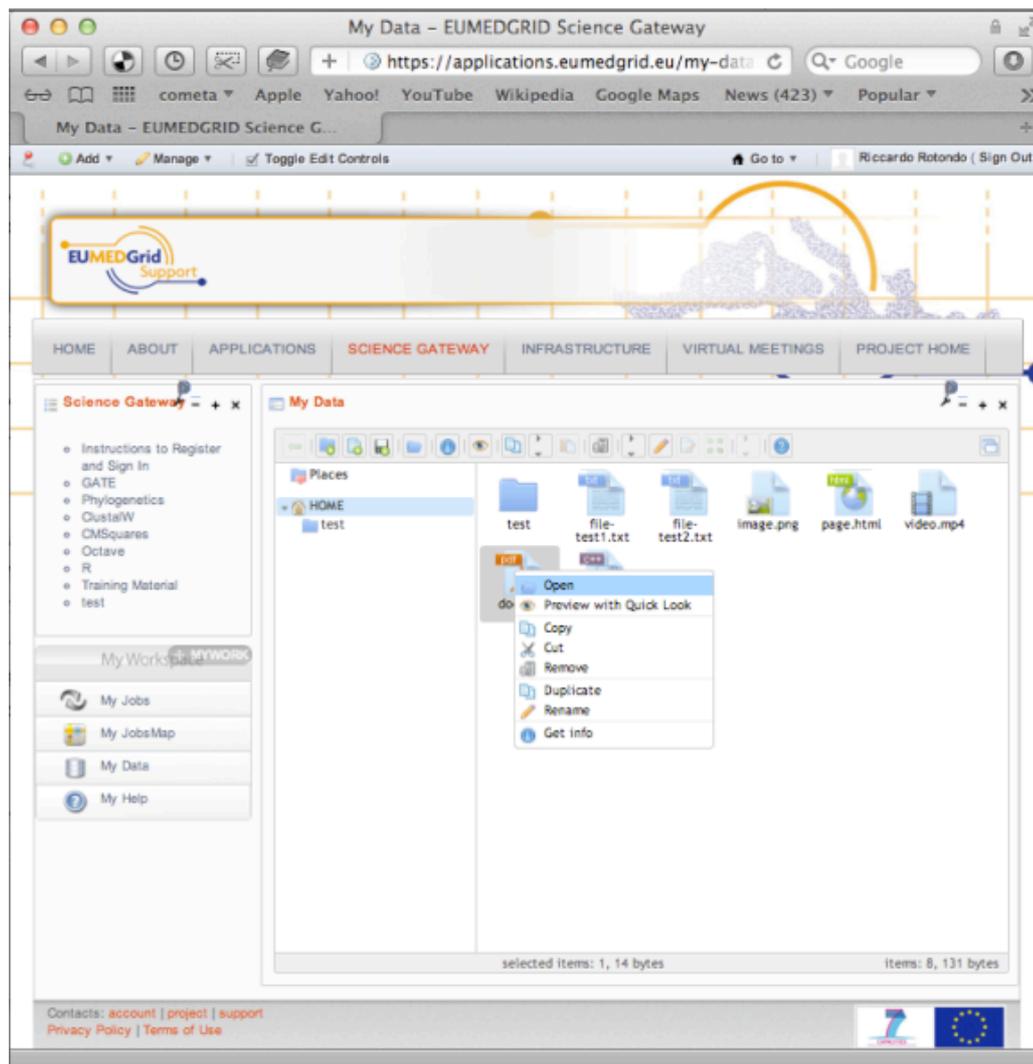


Figure 5 – Data Engine user interface based on elFinder

5. Conclusions

This work report of the implementation of DOGS (Data On Grid Services), a service to manage data stored on Grid integrated in a Science Gateway. DOGS enables users already relying on Grid computational power to benefit as well of the huge Grid storage capabilities.

DOGS uses several modules in order to offer simple data management services. Every transaction is secured and authenticated thanks to the proxy generated by robot certificates stored on the eTokenServer to which the Science Gateway is securely connected.

The Data Engine is fully based on standards to ensure maximum portability and reusability. Thanks to JSAGA, its implementation, currently tested with the gLite, is not bound to any specific middleware. To keep privacy ownership on files and non-reputability of Grid transactions, two databases keep track of every users action and file information, respectively. A set of Java objects are associated with file information, by means of the Hibernate framework,

and they are queried by the eFinder web file system to represent data as they appear in a common file browser organised in directories listed in a tree.

A weakness of the proposed service is the two steps data transfer from users to storage through the portal. Actually, the current version of the middleware of the Grid Storage Elements, lacks a HTTP/HTTPS protocol interface that could allow for direct file transfers from the user machine without passing through the Science Gateway.

A Science Gateway represents a place where users can share their work together with data coming out from their analyses. For this reason we will extend, as future work, eFinder functionalities with a file sharing among users (e.g. file sharing service such as Dropbox).

Currently the first beta of DOGS is going to be release in the next few months and it will be installed in the DECIDE Science Gateways. A group of selected users, from the community of neurologists within the DECIDE project, will access files on grid using this service and provide their feedback and evaluation to integrate in the future official release. Final release will be deployed in all the Science Gateways listed in section 2.1.

Finally, considering the spread of mobile devices, the creation of an “app” for smartphones and tablets providing DOGS functionalities is in our to-do list.

References

- [1] N. Wilkins-Diehr, *Special issue: Science gateways - common community interfaces to Grid resources.*, Concurrency and Computation: Practice and Experience 19 (2007), no. 6, 743–749.
- [2] N. Wilkins-Diehr, D. Gannon, G. Klimeck, S. Oster, and S. Pamidighantam, *TeraGrid science gateways and their impact on science*, IEEE Computer 41 (2008), no. 11, 32–41
- [3] All information about the DECIDE project can be found at www.eu-decide.eu while the DECIDE Science Gateway can be accessed at application.eu-decide.eu.
- [4] All information about the EarthServer project can be found at www.earthserver.eu while the EarthServer Science Gateway can be accessed at earthserver-sg.conorzio-cometa.it.
- [5] All information about the EUMEDGRID project can be found at www.eumedgrid.eu while the EUMEDGRID Science Gateway can be accessed at applications.eumedgrid.eu/science-gateway.
- [6] All information about the GISELA project can be found at www.gisela-grid.eu while the GISELA Science Gateway can be accessed at gisela-gw.ct.infn.it/science-gateways.
- [7] All information about the INDICATE project can be found at www.indicate-project.eu while the INDICATE e-Culture Science Gateway can be accessed at indicate-gw.conorzio-cometa.it.
- [8] All information about Liferay can be found at www.liferay.com
- [9] A Simple API for Grid Applications (SAGA): <http://www.ogf.org/documents/GFD.90.pdf>
- [10] A Requirements Analysis for a Simple API for Grid Applications: www.ogf.org/documents/GFD.71.pdf
- [11] All information about JSAGA, including source code and user manual, can be found on the website grid.in2p3.fr/jsaga.

- [12] Cryptographic Token Interface Standard (PKCS#11): www.rsa.com/rsalabs/node.asp?id=2133.
- [13] R. Barbera, V. Ciaschini, G. La Rocca, S. Monforte, S., *A new "lightweight" Crypto Library for supporting an Advanced Grid Authentication Process with Smart Cards*. Proceedings of the International Symposium on Grids and Clouds and the Open Grid Forum (ISGC 2011 & OGF 31). March 19 - 25, 2011. Academia Sinica, Taipei, Taiwan.
- [14] All information about Hibernate can be found at www.hibernate.org.
- [15] All information about elFinder can be found at elrte.org/elfinder.
- [16] R. Barbera, M. Fargetta, R. Rotondo, *A Simplified Access to Grid Resources by Science Gateways*. Proceedings of the International Symposium on Grids and Clouds and the Open Grid Forum (ISGC 2011 & OGF 31). March 19 - 25, 2011. Academia Sinica, Taipei, Taiwan.