

## Chapter 2

# Supporting e-Science Applications on e-Infrastructures: Some Use Cases from Latin America

**Roberto Barbera, Francisco Brasileiro, Riccardo Bruno, Leandro Ciuffo,  
and Diego Scardaci**

**Abstract** In this chapter, we describe a successful methodology to support e-Science applications on e-Infrastructures put in practice in the EELA-2 project co-funded by the European Commission and involving European and Latin American countries. The heterogeneous requirements of the e-Science applications, coming from several scientific fields, makes difficult to provide them with a support able to satisfy all the different needs. Usually, the grid middleware adopted, gLite in the case of EELA-2, provides applications with general tools not able to meet specific requirements. For this reason, a really powerful e-Infrastructure has to offer some additional services to complete and integrate the functionalities of the grid middleware. These services have to both increase the set of functionalities offered by the e-Infrastructure and make easier the tasks of developing and deploying new applications. Following this methodology, EELA-2 deployed 53 e-Science applications out of the 61 supported in total, in its enriched e-Infrastructure during its life.

---

R. Barbera  
Division of Catania, Italian National Institute of Nuclear Physics,  
Via Santa Sofia 64, Catania 95123, Italy

Department of Physics and Astronomy, University of Catania, Catania, Italy

F. Brasileiro  
Department of Systems and Computing, Universidade Federal de Campina Grande,  
Campina Grande, Brazil

R. Bruno • L. Ciuffo • D. Scardaci (✉)  
Division of Catania, Italian National Institute of Nuclear Physics, Via Santa Sofia 64,  
Catania 95123, Italy  
e-mail: diego.scardaci@ct.infn.it

## 2.1 Introduction

In this chapter, we describe a successful methodology to support e-Science applications on e-Infrastructure starting from the experience of the EELA-2 project [9] co-funded by the European Commission and involving European and Latin American countries. In order to satisfy all heterogeneous applications requirements and to simplify the access to the grid infrastructure, a set of special services has been developed to enhance the functionality of the gLite middleware [12] and provide users with a richer platform.

The middleware services developed have widened the number of potential applications taking benefit of the grid e-Infrastructure and, moreover, have speeded up the porting of applications to run in the grid. It can be considered general and applied in other similar contexts.

This chapter is organized in the following way. Section 2.1 introduces e-Science in Latin America. Section 2.3 describes the main characteristics of applications involved in the EELA-2 project. Section 2.4 presents the additional services developed, according to applications requirements, to enhance gLite functionalities. Section 2.5 describes how these services have been adopted by the EELA-2 applications. Future perspectives and conclusion are then drawn in the last two sections (Sects. 2.6 and 2.7, respectively).

## 2.2 Related Work

Latin America has been one of the regions supported by the European Commission through the Information and Communications Technology (ICT) scheme. Grid computing has been funded through the EELA [8], and EELA-2 projects with a noticeable success.

The EELA-2 project (E-Science Grid facility for Europe and Latin America) is EELA's second phase. It aims at building and operating a production grid infrastructure, which provides computing and storage resources from selected partners across Europe and Latin America. During its first phase, a collaborative human network was established by means of several advances in the grid infrastructure setup, deployment of new certification authorities, support of pilot applications, and organization of many workshops and training events that made the Latin American research community aware of what grid could do.

The current EELA-2 production infrastructure operates middleware Core Services located in both Europe and Latin America [11], and works with grid sites that have demonstrated to have an adequate maturity level in par with what is expected from a production environment. It also maintains virtual organization services, with one main Virtual Organization (VO) – named “prod.vo.eu-eela.eu” – gathering the infrastructure services. It is also worth mentioning that EELA-2 adopts both the gLite middleware and the OurGrid [6, 16] middleware; the former powers its service infrastructure, while the latter is used to set up a complementary opportunistic infrastructure.

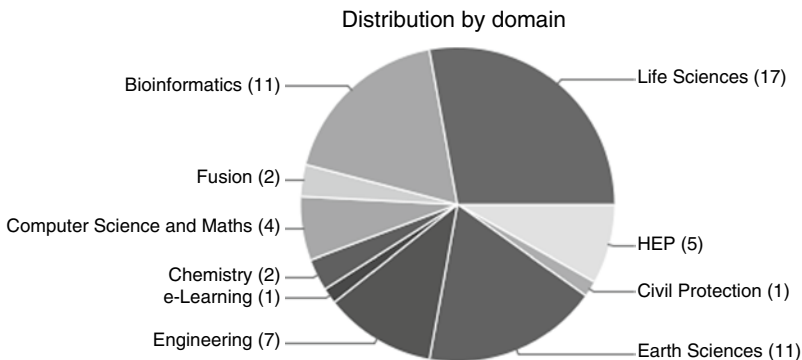
## 2.3 Supporting e-Science Applications in Latin America

It is well known that the development of grid computing was initially driven by the requirements of efficient processing of the huge amounts of data generated by High Energy Physics experiments such as those running at the CERN Large Hadron Collider (LHC) [5]. However, investments to promote grid computing within new scientific communities in several regions of the world have been attracting new research groups interested in investigating the potential benefits that grid computing can bring for their academic pursuits. Indeed, grid has proven to be a very effective way of tackling intensive computing needs such as those found in climate simulations and drug discovery.

Based on our experience supporting 61 applications in the framework of the EELA-2 project, we have noticed that grid users may be broadly divided into three groups:

- Those participating in collaborative experiments which require High-Throughput Computing (HTC) across many computing and storage clusters.
- Those that have computational and storage demands that cannot be handled by their local resources in a reasonable time – these users require to access extra resources belonging to others only for the purpose of attending the excess in their workload.
- Those with modest computational needs that could be easily handled by a local cluster or storage server; in this case, the affiliation of these groups with a large grid project might allow them to overcome the digital divide just by granting access to extra computing resources.

Such a diversity of users is one of the consequences of the grid expansion across many institutions/countries facing different maturity levels of Information Technology (IT) infrastructures, network connections, and e-science awareness. This is also reflected on the application's profile. On one hand, EELA-2 supports applications that run thousands of parallel jobs per week which last for many hours and handle gigabytes of data, but on the other hand, there is also bag-of-task application that runs one single job on an occasional basis and consumes relatively much less computing resources. Figure 2.1 shows the current distribution of the 61 EELA-2 applications per scientific domain.



**Fig. 2.1** Distribution of EELA-2 applications per scientific domain

## 2.4 Developing e-Infrastructure Services for e-Science Applications

To cope with all heterogeneous aspects presented above and to meet the requirements of the applications, a set of special services has been developed. These services enhance the functionality of both the gLite and the OurGrid middleware giving the EELA-2 users access to richer middleware, reducing the amount of application development, and generally accelerating the adoption of grid technologies.

The identification and the development of these services has been conducted with the goal of increasing the reach and the usability of e-Infrastructure by assembling/reengineering existing technologies and developing new ones that could facilitate the installation, management, and use of the grid infrastructure. By increased reach, we mean more sites belonging to the grid infrastructure, with more resources being shared, more users being served, and a more diverse range of applications being supported. By increased usability, we mean the addition of services that can ease the tasks of developing and deploying new applications, as well as installing, managing, and deploying the core infrastructure. Therefore, the development of these additional services has focused mainly on the design, implementation, and deployment of new infrastructure-oriented and application-oriented grid services.

The infrastructure-oriented services have been developed to provide alternatives to ease the installation, management, and use of the e-Infrastructure. To this end, the following services have been built:

- A gateway between gLite and Ourgrid [2, 16], a simpler peer-to-peer (P2P) technology, to provide alternative ways to make resources available to the grid infrastructure and to simplify the access to the infrastructure for new users and applications
- The porting of the gLite User Interface and Computing Element to the Microsoft Windows platform [17] to facilitate the access to the infrastructure to Windows users and to allow Windows Applications to use the infrastructure
- The Storage Accounting for Grid Environments (SAGE) [19], a system to measure the usage of storage resources in a gLite based on grid infrastructure whose main task is to collect information from physical devices and make this data available to system administrator to account the storage at higher levels

The application-oriented services have been developed according to the feedback of the users and site administrators:

- The Grid Storage Access Framework (GSAF) [20], an object-oriented framework designed to access and manage Data Grid via APIs. It provides developers with a development tool to write application that adopts grid as Digital Repository hiding the fragmentation and the complexity of the Data Grid Services. GSAF also provides a basic transaction layer for multiservice operation (i.e., synchronization of data operations).

- The Secure Storage [18] is a service for the gLite middleware which provides the users with a set of tools in order to store confidential data in a secure way and in an encrypted format (e.g., medical or financial data) on the grid storage elements. The data stored through provided tools is accessible and readable by authorized users only, preventing also the administrators of the storage elements to access the confidential data in a clear format (e.g., insider abuse problem).
- OPeNDAP Meta-Finder, a tool for searching geographical information data sets available at the Web through OPeNDAP servers. By filling a form, the user can search for data sets containing some wanted attributes and variables such as spatial coverage, temporal coverage, atmosphere temperature, precipitation, etc. To make queries easier, tag clouds are shown with some suggestions of tags to classify published data sets.
- The Watchdog [3] is a tool that allows users to watch the status of a running job when it runs on a working node tracing the evolution of produced files.
- The lcg-rec-\* tools allow users to perform recursive Grid File Operations such as copies and deletion.
- DIRAC is an alternative to overcome an infrastructure limitation of not having many Computing Elements able to support MPICH2 jobs.

In the following sections, we describe in detail the additional services that caused a major impact in terms of increasing the reach and the usability of the EELA-2 e-Infrastructure.

### ***2.4.1 A Gateway Between gLite and OurGrid***

The gateway technology we use is the 3G-bridge proposed in the context of the EDGeS project [4]. Jobs originated in one system run in the other in a completely transparent way, using the standard gLite user interface and the OurGrid broker. Nevertheless, jobs may carry additional requirements to force their execution cross-platforms.

A middleware-specific adaptor is responsible for converting the jobs from their particular native format to a canonical format defined by the 3G-bridge; the translated jobs are stored in the gateway database to be later dispatched to the appropriate grid. The gateway software constantly inspects the gateway's database looking for jobs that need to be dispatched. For each grid to which the gateway interfaces, there is an instance of a middleware-specific plug-in that implements a job management API defined by the 3G-bridge. Upon detecting that a job needs to be dispatched, the gateway identifies to which grid the job should be submitted and calls the appropriate functions in the correspondent plug-in instance. The plug-in is responsible for creating a new job in the appropriate format and submitting it to the grid to which it is associated. When the job is completed, the database is updated and, eventually, the user interface is notified.

We have developed a Java implementation of the 3G-bridge and plug-ins for both gLite and OurGrid. The OurGrid plug-in is implemented by a modified version of

the OurGrid Broker that implements the 3G-bridge API; a peer representing the gateway is instantiated to serve this Broker, and jobs are submitted to the OurGrid system through the Broker as if they had been submitted by the user. The gLite plug-in uses a standard gLite User Interface to submit jobs to the gLite back end.

Suitable adaptors have been developed to map OurGrid and gLite jobs sent to the gateway into jobs described in the canonical format and stored in the 3G-bridge database. The adaptor that allows OurGrid jobs to be stored in the gateway's database for execution in the gLite system is implemented by a modified version of the OurGrid Worker.

This new Worker is called a Front-End Worker, since it does not manage a resource, but simply provides access to resources managed by other components. The Broker sees the Front-End Worker as a normal Worker and dispatches jobs to it in the same way it does for other Workers. The fact that the task is actually executed in a gLite grid is completely transparent from the Broker's perspective. For gLite, we have implemented a modified version of the CREAM (Computing Resource Execution and Management) Computing Element [7] that is able to store gLite jobs in the 3G-bridge database.

The common syntax for the jobs was obtained by the definition of a MySQL database schema [15] that allows the storage of jobs in a canonical format. Regarding the availability of extension points, it could be reached by the development of a plug-in framework that uses a common interface, called GridHandler, allowing the implementation of middleware-specific plug-ins. So, each deployed plug-in implements a different strategy of job execution management on the destination grid back end. The software was written in Java [13] and fully deployed as an Apache Tomcat [1] application.

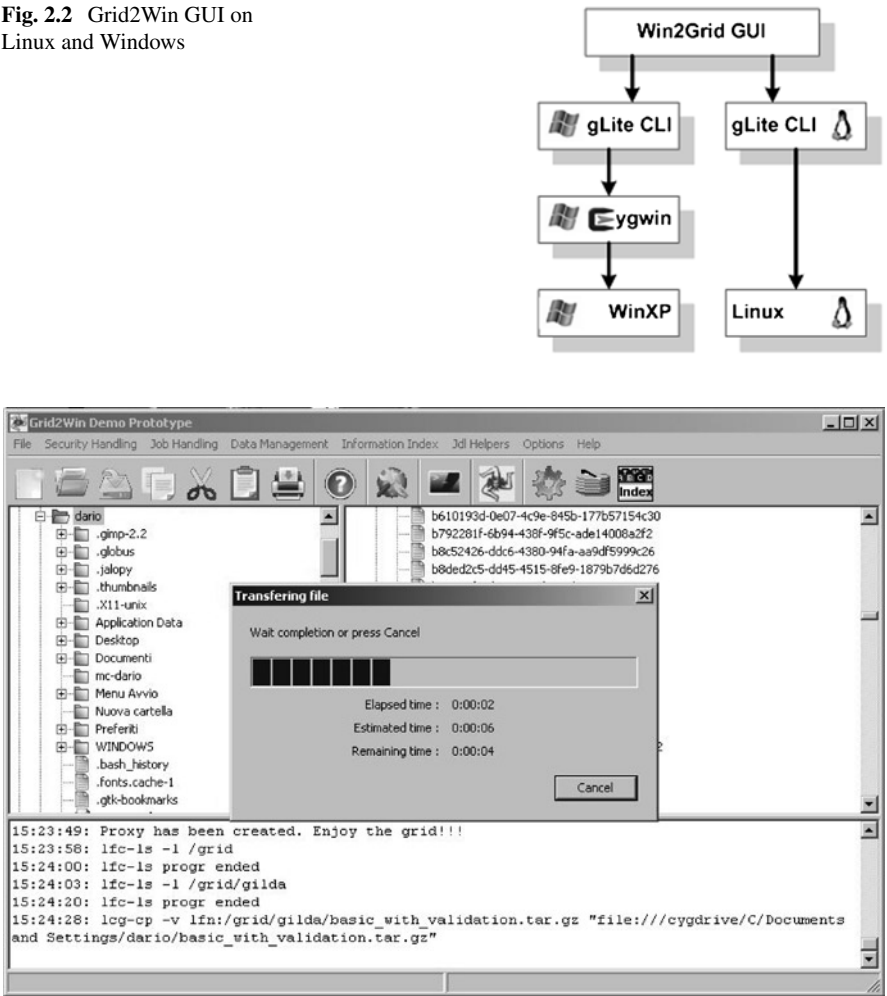
### ***2.4.2 Porting gLite to the Microsoft Windows Platform***

We rebuilt parts of both the Globus Toolkit and the gLite middleware to allow the execution of the gLite middleware on Microsoft Windows operating systems. This work allowed us to provide a gLite Windows User interface, facilitating the access to the infrastructure to Windows users and, moreover, a gLite Windows Grid Site, allowing Windows Applications to use the infrastructure.

The Grid2Win GUI is a Graphical User Interface running on both Windows and Linux operating systems. We designed the Grid2Win GUI to operate in cross-platform style. We used the wxWidget cross-platform library that gives us the certainty to build on Windows, Linux, MacOS, and other operating systems (provided that a gLite command line user interface exists for such operating system). Figure 2.2 illustrates the elements involved in the execution of gLite UI over different operating systems using Grid2Win.

This GUI (Graphical User Interface) gives the possibility to use the grid in a easy way, like any other windows program, allowing to download/upload files with just one mouse click, submitting a job, wizard-driven authentication process,

**Fig. 2.2** Grid2Win GUI on Linux and Windows

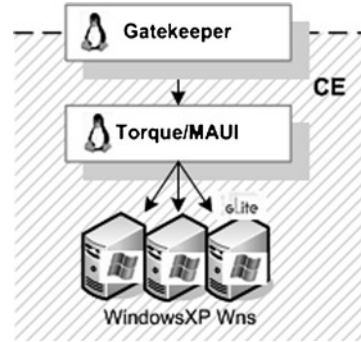


**Fig. 2.3** A snapshot of the Grid2Win GUI

wizard-driven Job Description Language (JDL) creation, etc. Since this GUI is not linked to the underlying UI, it works even if the command line programs change (provided that their correspondent binaries still exist and the interfaces are still the same which is likely to be the case). Figure 2.3 shows a snapshot of the interface.

The deployment of a gLite grid site on Microsoft Windows allowed computational nodes to run Windows-based applications. To achieve this goal, our first approach to Microsoft Windows computing farms was based on the Torque/MAUI scheduler, a free version of PBS (Fig. 2.4). The farm is composed of a unique central node with Linux operating system in which the Gatekeeper and the Torque/MAUI head node are running (a gLite Computing Element) and a set of Worker Nodes with our rebuilt version of the gLite WN package.

**Fig. 2.4** Computing element with Microsoft Windows Worker Nodes



### 2.4.3 Grid Storage Access Framework

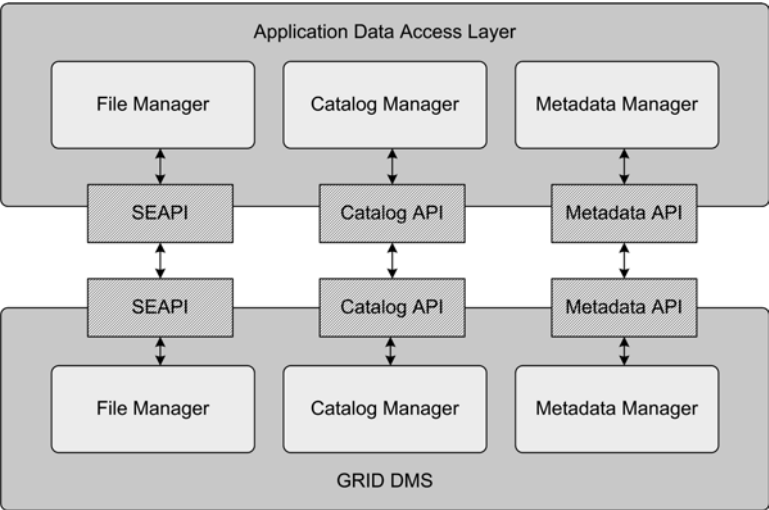
Grid Storage Access Framework (GSAF) is a tool developed to access and to manage Data Grid via APIs that solves several common design problems of the applications providing developers with a tool to write application that adopts grid as Digital Repository hiding the fragmentation and the complexity of the Data Grid Services.

Within a grid infrastructure, files are stored inside Storage Elements (SE); the grid services that take care of data persistency). They can be replicated on several SEs for ubiquity, security, and sharing purposes. Relationships between locations of files, replicas, and their logical names are kept within a specific File Catalogue Service. For each file, it is possible to associate some descriptive attributes (metadata schema), and values (schema instance) can be assigned to them, modified, quantified, and queried through a specific Metadata Catalogue Service.

Since Data Grid Services (File Catalogue, Metadata Catalogue, and Storage Elements) are independent from each other and work in a “stand-alone” mode, applications that want to adopt them must implement specialized and decoupled software components according to a vertical architecture (Fig. 2.5).

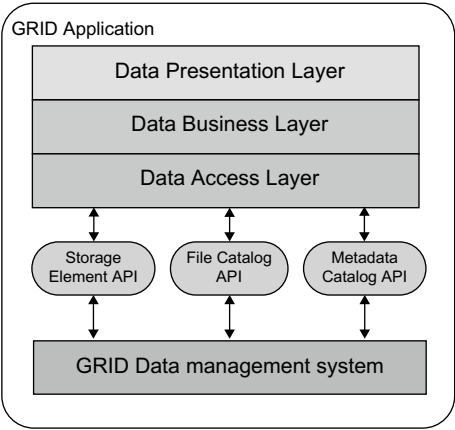
This fragmentation leads to the following problem: for each application, developers must always write the same code in order to include data services capabilities and take care about the atomicity, coherence, and the synchronization of data manipulation. Furthermore, there are neither tools nor services that help clients (both end user and software modules) to maintain semantic coherence and integrity consistency among files stored on the SEs and their entries inside the File Catalogue and the Metadata Catalogue. These are considered big limitations of the Grid Storage System and our effort has been in the direction of addressing them via a software layer that can solve both problems. We can say that GSAF implements a kind of Software Engineering Pattern for grid applications, offering a standard solution for a common problem according to the principle of write once and use everywhere.

The analysis of the architecture of a generic enterprise application emphasizes the problem of the fragmentation described above. Typically, this kind of application follows the three-layer architecture model, derived from the MCV (Model, Control, View) design pattern (Fig. 2.6).

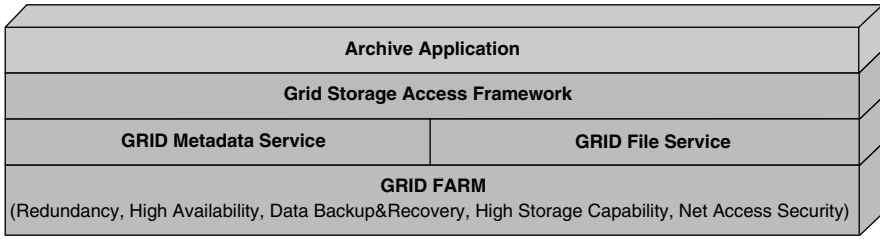


**Fig. 2.5** Grid data services integration schema

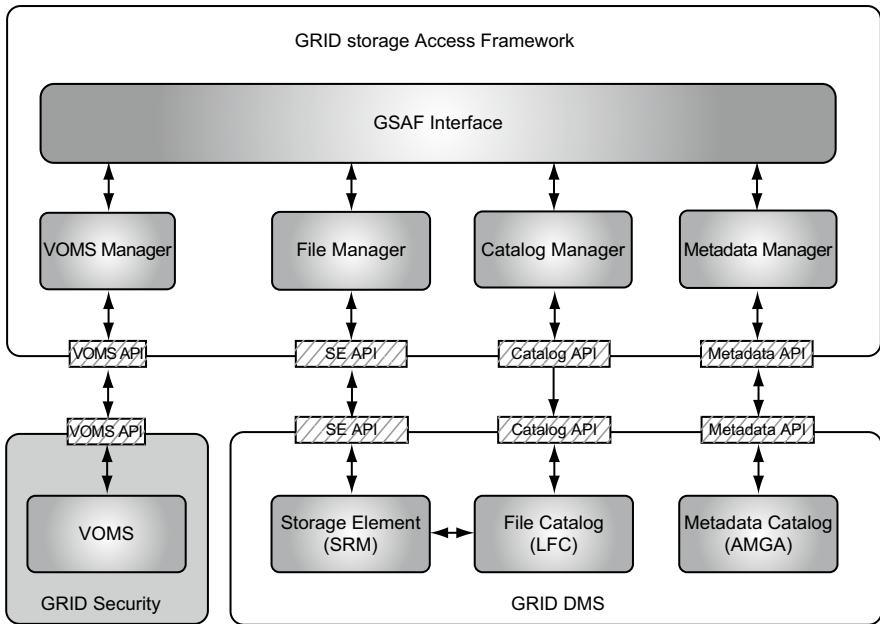
**Fig. 2.6** Grid data services integration schema



As shown in Fig. 2.7, the Data Presentation Layer consists of the graphical interfaces that make the user able to interact with the application. The Data Business Layer collects all software components that implement the behavior of the given application. The Data Access Layer is made of software components that allow the application to manage data and metadata, typically ASCII files, XML files, digital objects, SQL data, etc. It is evident how developing enterprise applications over grid means to replace the traditional Data Access Layer with an appropriate interface that allows business components to manage data stored on the Data Management System (DMS) and presentation objects to search and retrieve data from it.



**Fig. 2.7** Layered architecture view of GSAF



**Fig. 2.8** Architectural framework view of GSAF

GSAF represents this interface (Fig. 2.7). It is an object-oriented framework built on top of the gLite Metadata Service and gLite File Services that exposes classes and related methods. These, in turn, hide the complexity and the fragmentation of the several underlying Application Programming Interfaces (APIs) for applications located above it.

GSAF wraps each gLite Data Services through a corresponding software module built on top of the related API. Figure 2.8 shows the architecture of the framework. The File Manager provides functions to interact with the Storage Element allowing the upload/download of files to/from the SE. The Catalogue Manager is related to the File Catalogue Service and provides functions to register and de-register files. It also allows browsing and managing virtual directories. The Metadata Manager interfaces the Metadata Catalogue Service wrapping all metadata-related functions.

The Virtual Organization Membership Service (VOMS) Manager is responsible to get a valid proxy from the VOMS server. Finally, the main module, the GSAF Interface, works as a dispatcher for the application requests and as a coordinator of the underlying components. It provides a unique interface and takes care of both data atomicity and coherence including the transaction capability.

#### 2.4.4 The Secure Storage Service

The Secure Storage Service (SSS) for the gLite middleware provides users with a set of tools to store in a secure way and in an encrypted format confidential data, for example, medical or financial data, on the grid storage elements. The data stored through the provided tools is accessible and readable only by authorized users. One of the important contributions of this service is that it solves the insider abuse problem, as well as it prevents the administrators of the storage elements to access the confidential data in a clear format.

The Secure Storage Service has been designed to be integrated into the gLite middleware. It consists of the following components (Fig. 2.9):

- **Command Line Applications:** Commands integrated into the gLite User Interface to encrypt and upload, as well as download and decrypt files on the storage elements.

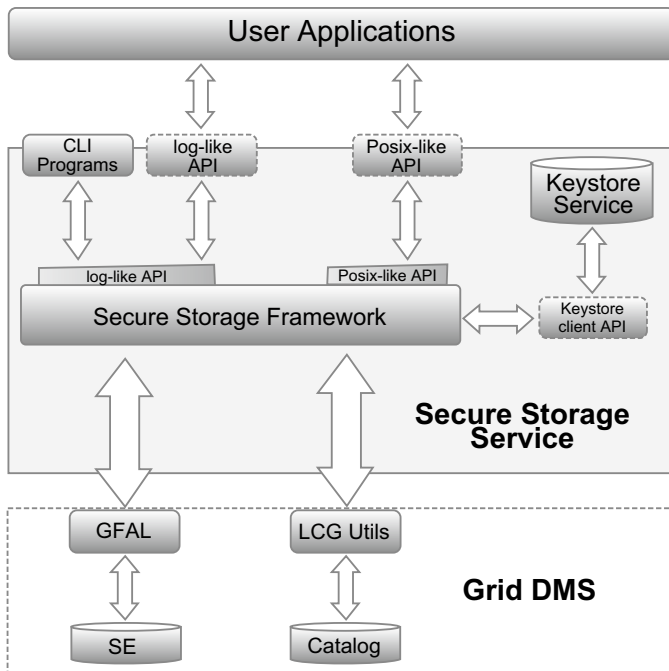


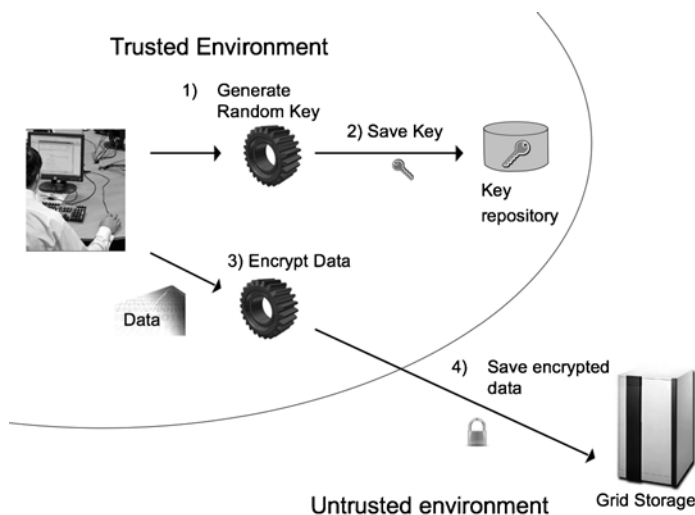
Fig. 2.9 Secure storage service architecture

- An Application Program Interface: The API allows the developer to write programs able to manage confidential data using the Secure Storage Service.
- The Keystore: A new grid element used to store and retrieve users' keys in a secure way.
- The Secure Storage Framework: A component of the service, internally used by the other components. It provides encryption/decryption functions and other utility functions. It takes care of the interaction with the Grid Data Management System.

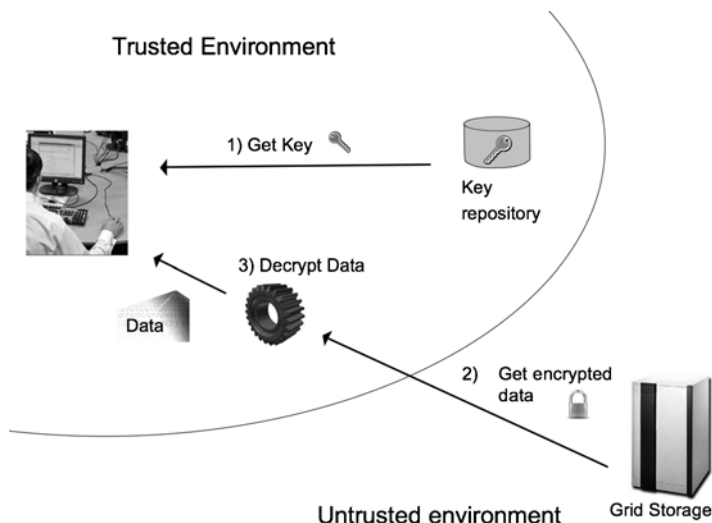
Advanced Encryption Standard (AES) algorithm is the default encryption algorithm used in the Secure Storage Service with a 256-bit long key. However, the service is able to support new symmetric algorithms, thanks to its modular architecture.

This service provides a set of new Command Line Applications on the gLite User Interface. These applications allow users to manage confidential data in a secure way. The Command Line Applications of the Service used to upload/download data to/from storage elements are described in the following:

- *lcg-scr*: the input parameters of this command are a local file, a storage element, a Logical File Name (LFN), and a list of users authorized to access the file. The command generates an encryption key, encrypts the input file, and uploads it on the storage element, registering its LFN in an LFC (LCG File Catalogue) file catalogue. Moreover, it stores the key generated and used to encrypt the file in the keystore. An ACL will be created and associated to the encryption key on the keystore. This ACL will contain all users authorized to access the file, represented by a list composed by pairs of Distinguished Name (DN), and Fully Qualified Attributes Name (FQAN). Figure 2.10 depicts the steps involved in the execution of this command.



**Fig. 2.10** *lcg-scr* command: (1) A new random secret key is generated, (2) the key and the ACL are saved on the keystore, (3) the input file is encrypted inside user-trusted environment, (4) the encrypted file is uploaded on the grid storage element



**Fig. 2.11** `lcg-scp` command: (1) Get the secret key from the keystore; this operation fails if the user is not authorized, (2) download the encrypted file on the local machine, (3) decrypt the file and save it on the local file system

- *lcg-scp*: the input parameter of this command is an LFN. It downloads the encrypted file identified by the input LFN, gets the key to decrypt the file from the keystore, decrypts the file, and then stores it on the local file system. This command successfully returns only if the user is an authorized user (the authorization procedure is described later on in this document. Figure 2.11 shows the steps involved in the execution of the `lcg-scp` command.

### 2.4.5 The Watchdog

The Watchdog is a tool that allows grid users to control and monitor the execution of their own jobs. The need to monitoring and control the job execution while the grid application runs on the Worker Nodes is one of the primary needs of grid developers since they can easily interact with the running jobs.

Monitor and control grid jobs are not only useful during the early development phases of grid applications, but they are also useful to aid the application users when exploiting the production infrastructure. Thanks to a Job monitoring and control tool, grid jobs may produce real-time results or even let the application be piloted by the user. It is also possible to stop unnecessary computations avoiding costly waste of time for both users and infrastructure resources.

The early design of the Watchdog has been developed to assist all grid applications requesting long computation time. The idea behind the Watchdog was to inspect the execution of a grid job monitoring, timely, a set of files produced by

the grid job while it is running on the Worker Node. Soon this simple idea has been enriched with different features and capabilities, creating a very flexible and powerful tool to monitor and also control the job execution.

The EELA-2 grid infrastructure currently relies on the gLite middleware version 3.1 and this release of the middleware does not provide any flexible way to monitor and control the grid job execution. The gLite 3.1 only offers a basic and native support for the job monitoring named the Job Perusal mechanism. The Workload Management System (WMS) handles directly the Job Perusal monitoring system, and its behavior is directly piloted by a set of specific statements of the Job Description Language (JDL). Exactly like the Watchdog, the Job Perusal mechanism allows to check periodically the content of files produced by the jobs while they are running on the WN. Although the Job Perusal mechanism provides a standard way to monitor grid job files, it cannot be considered as flexible as the Watchdog is. Moreover, the Job Perusal mechanism has been introduced since gLite v3.1 and few grid developers feel confident with this newly introduced middleware capability while very few tests have been made so far. In the EELA-2 community, there are no applications that are currently adopting the Job Perusal.

The Watchdog consists of few bash script files; this decision has been taken to help grid application developers to easily customize the Watchdog behavior accordingly to their personal needs. The grid application user has just to set up a configuration file and include the Watchdog files into the Job Input Sandbox.

The Watchdog does not need any special installation to be done by the grid site administrators and it has been built only on top of existing gLite services, this implies that the use of the Watchdog does not compromise the grid security or the Job execution performances.

The way the Watchdog uses to monitor and control the job execution consists of mainly two possibilities: Timely, keep track of Job files content (log files, output streams, etc.) and allow the execution of commands or scripts sent by the user while the grid job runs. In the Watchdog terminology, the monitor files are called “file snapshots” and these files may report the whole content of the file or just the last changes happened during the last time interval. The list of files and the kind of snapshots can be changed anytime. It is also possible to change the time interval and stop/resume/end the monitoring while the job runs.

The Watchdog interaction between the monitored jobs and the user can be done in three different fashions:

- Using an LFC file catalog
- Using an AMGA server
- Using a mounted Network File System (NFS) accessible from the WN; if one is foreseen by the site administrators (AFS, NFS, etc.)

Recent efforts have been made to improve the quality of the user interface developing a command line interface as helper tool in order to easily get file snapshots from monitored files and submit commands to be executed on the Worker Node by the Watchdog on the user behalf.

## 2.5 Exploit the Additional Grid Services

In this section, we describe how the additional grid services have been adopted by the applications.

### 2.5.1 *Application Porting and Final Balance at the End of the EELA-2 Project*

At the end of the project on March 2010, 53 applications out of the 61 supported in total were successfully deployed and interfaced with the grid middleware (Table 2.1). It is worth mentioning that some EELA-2 resource centers also support the Virtual Organization (VOs) from well-known High-Energy Physics (HEP) experiments such as ALICE, ATLAS, LHCb, CMS, and Pierre Auger.

Several applications use the additional grid services developed by the project. These services helped to speed up the application porting on the grid infrastructure and sometimes made possible the gridification of some applications.

The complete list of applications supported by EELA-2, as well as its descriptions and references, can be retrieved from the URL <http://applications.eu-eela.eu>.

### 2.5.2 *Increasing the Reach of e-Infrastructure*

The EELA-2 infrastructure has profited from the additional services developed in many ways. The services that provide with a major benefit in raising the reach of e-Infrastructure are, first, the adoption of the OurGrid middleware, that has increased the portfolio of middleware supported by the EELA-2 infrastructure and, second, the porting of the gLite to the Microsoft Windows platform that allows Windows Applications to use the infrastructure.

The adoption of OurGrid allowed non-dedicated resources to be added to the infrastructure. Moreover, it allowed resources, dedicated or not, to be added to the infrastructure without the need to change their operating system. The seamless integration of these resources was made available thanks to the work carried out in the EELA-2 project that enhanced the OurGrid middleware to use the same kind of certificates that are used by gLite and other established grid middleware. Figure 2.12 shows a snapshot of the monitoring page for the opportunistic part of the EELA-2 infrastructure. As it can be seen, around 300 cores are available in this part of the infrastructure and the tendency is that this number will substantially increase.

Figure 2.13 shows the availability of cores in the opportunistic grid during the last quarter of 2009. As it can be seen, the total number of cores that can potentially be available to the grid kept over 300 during the whole period shown. From these, a little less than 50% are normally available.

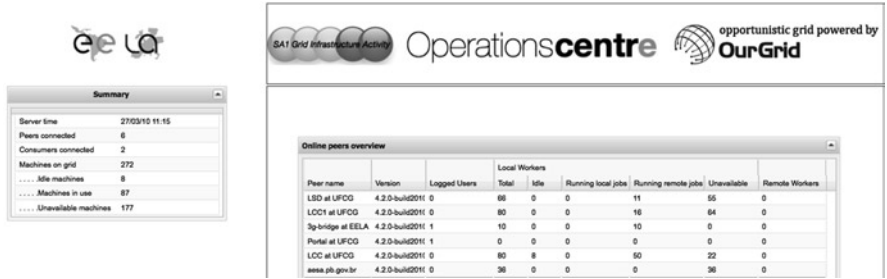
**Table 2.1** Applications deployed on EELA-2 infrastructure

Application	Scientific domain	Country
AERMOD	Earth Sciences	Cuba
AeroVANT	Engineering	Argentina
Aiuri	Computer Science and Mathematics	Brazil
BiG (Blast)	Bioinformatics/Genomics	Spain
BioMD	Life Sciences	Brazil
bioNMF	Bioinformatics/Genomics	Spain
BRAMS	Earth Sciences	Brazil
C/CATT-BRAMS	Earth Sciences	Chile/Brazil
CAM	Earth Sciences	Spain
CardioGrid Portal	Life Sciences	Argentina
CATIVIC	Life Sciences (Chemistry)	Venezuela
Cinefilia	Computer Science and Mathematics	Italy/Brazil
CIS – Classification of Satellite Images with neural networks	Earth Sciences	Ecuador
CROSS-Fire	Civil Protection	Portugal
DicomGrid	Life Sciences	Brazil
Dist-SOM-PORTRAIT	Bioinformatics/Genomics	Brazil
DistBlast	Bioinformatics/Genomics	Brazil
DKEsG	Fusion	Spain
DRI/Mammogrid	Life Sciences (e-health)	Spain
eIMRT	Life Sciences (e-health)	Spain
FAFNER2	Fusion	Spain
fMRI	Life Sciences (e-health)	Portugal
G-HMMER	Bioinformatics/Genomics	Colombia
G-InterProScan	Bioinformatics/Genomics	Colombia
GAMOS	Life Sciences	Spain
gCSMT	Earth Sciences	France
GenecodisGrid	Bioinformatics/Genomics	Spain
GrEMBOS	Bioinformatics/Genomics	Mexico
Grid Bio Portal	Bioinformatics/Genomics	Spain
GRIP – Grid Image Processing for Biomedical Diagnosis	Life Sciences	Chile
GROMACS	Life Sciences (Chemistry)	Brazil
gRREEMM	Engineering	Cuba
gSATyrus	Computer Science and Mathematics	Brazil
Heart Simulator	Life Sciences	Brazil
HeMoLab	Life Sciences	Brazil
Industry@Grid	Engineering	Brazil
Integra-EPI	Life Sciences	Brazil
InvCell	Life Sciences	Brazil
InvTissue	Life Sciences	Brazil
LEMDistFE	Engineering	Mexico
MAVs-Study	Engineering	Argentina
META-Dock	Bioinformatics/Genomics	Mexico
Phylogenetics	Life Sciences	Spain
PhyloGrid	Life Sciences	Spain

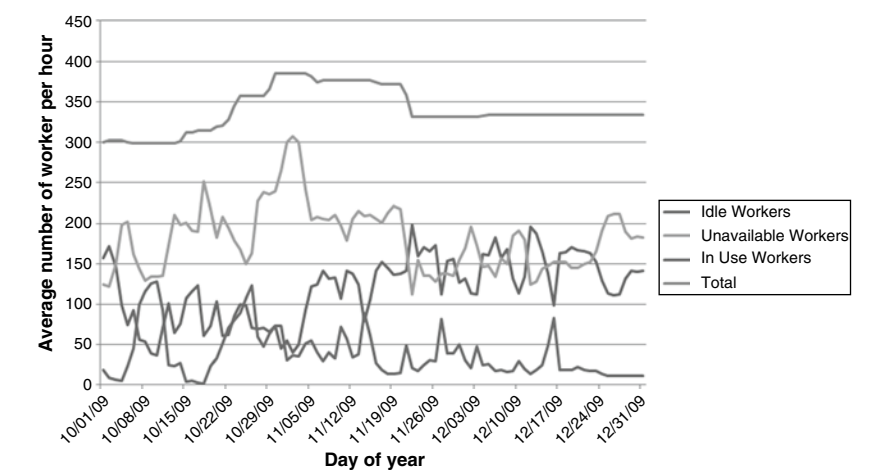
(continued)

**Table 2.1** (continued)

Application	Scientific domain	Country
PILP	Computer Science and Mathematics	Portugal
Portal de Porticos	Engineering	Venezuela
ProtozoaDB	Life Sciences	Brazil
PSAUPMP	Engineering	Mexico
SATCA	Earth Sciences	Mexico
Seismic Sensor	Earth Sciences	Mexico
SEMUM3D	Earth Sciences	France
WAM	Earth Sciences	Ireland
WRF	Earth Sciences	Spain

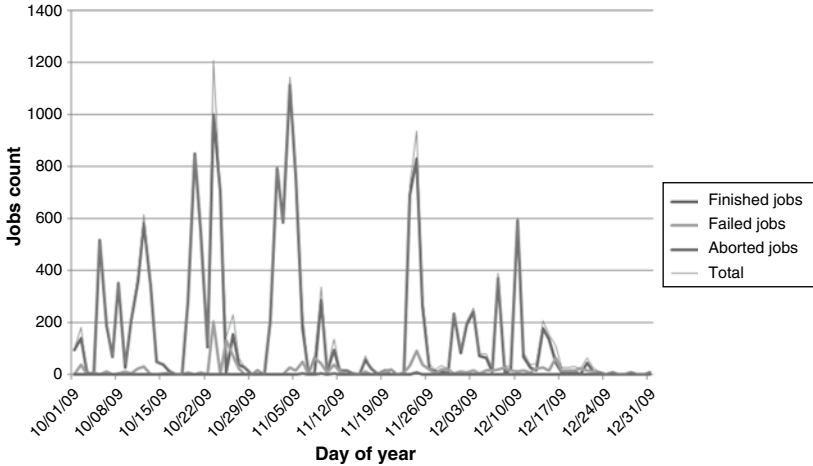


**Fig. 2.12** Opportunistic infrastructure monitoring data



**Fig. 2.13** Worker Nnodes availability for the opportunistic infrastructure (last quarter of 2009)

Figure 2.14 shows the job submission status for the opportunistic part of the infrastructure during the last quarter of 2009. As it can be seen, the number of failed jobs is fairly small. Moreover, most of the failures are due to the resources becoming



**Fig. 2.14** Job submission for the opportunistic infrastructure (last quarter of 2009)

unavailable to the grid (these resources are not dedicated to the grid). This gives evidence of the reliability of the infrastructure.

The other important contribution to increase the reach of the EELA-2 e-Infrastructure was the port of the gLite User Interface, Computing Element, and Worker Nodes components to the Microsoft Windows platform. This allowed resource centers that had staff well trained in managing MS Windows-based system to offer to their users the full power of the gLite middleware – before this port was done, the only option for these system administrators was either to leverage on virtualization technology to run gLite on Scientific Linux systems executed as guest operating systems in virtual machines run on top of the host MS Windows-based system or to provide their resources to the grid using the OurGrid middleware. Currently, there is one resource center (in INFN/Catania) running the MS Windows port of gLite and other resource center should be deployed in the months to come. More importantly, there is already one application (MAVs-Study) benefiting from this development.

### 2.5.3 *Increasing the Usability of e-Infrastructure*

The application-oriented services developed in the context of the EELA-2 project helped to solve many common problems related to application deployment and allowed the gridification of some applications with requirements not easy to satisfy using the standard gLite environment. These services have been widely advertised in the EELA-2 Web site, as well as in the various dissemination events organized during the grid schools and the gridification weeks.

The grid schools are closed retreats of 1–2 weeks where application experts work in close collaboration with tutors to interface their applications with the grid

**Table 2.2** List of applications that use JRA1 application-oriented services

Service	Application using it	Use description
Digital Archives (GSAF)	gRREEMM	Provide gRREEMM with a Java interface toward the gLite Data Management System
Secure Storage	CardioGrid Portal, HeMoLab, Seismic Sensor, AeroVANT	Applications use this service to manage their confidential data in a secure way inside the Grid infrastructure
Tagging	BRAMS and other SegHidro applications	Help in searching shared data among researchers working on water resources management
Workflow	BRAMS and other SegHidro applications	Several SegHidro applications use it to implement their workflow, starting from the water precipitation prediction provided by BRAMS
Watchdog	CROSS-Fire, InterproScan/HMMER, HeMoLab, AeroVANT, BioMD, Cinefilia	Several applications use it to monitor their long running jobs
Lcg-rec	CROSS-Fire, META-Dock	CROSS-Fire and META-Dock use it to interact in an easier way with the LCG File Catalog
DIRAC	BioMD, PSAUPMP	Ease their execution in sites with different MPI “flavors.”

middleware. At the end of the school, the applications involved are able to run on the grid infrastructure.

These events provided us with a good chance to identify applications that could benefit of the additional services. Starting with the applications requirements analysis, we identified the problems that could hinder the applications gridification and the services that could help us to remove these impediments.

For example, applications managing confidential data cannot be easily ported in a grid infrastructure because the standard gLite middleware does not provide users with tools to manage confidential data in a secure way. Instead, profiting of the Secure Storage Service, this obstacle can be removed and these applications are able to run on the grid infrastructure satisfying their requirements. As a result, a number of applications are currently using the services developed and all services are currently been used by at least one application. Table 2.2 gives a summary of this utilization.

### 2.5.4 *Gridify a Windows Application – The MAVs-Study Use Case*

In order to allow Windows application to profit from the grid infrastructure, in this section we will provide a detailed description of how we ported the Windows application MAVs-Study on the grid infrastructure. MAVs-Study – Biologically

**Table 2.3** MAVs- Study on the grid (JDL and Start Script)

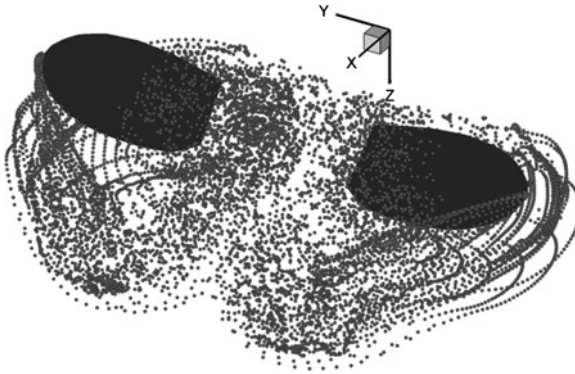
MAVs-Study on the GRID	
JDL	Start Script
<pre>Type = "Job"; JobType = "Normal"; Executable = "uvlmRF"; StdOutput = "uvlmRF.out"; StdError = "uvlmRF.err"; InputSandbox = {"uvlm.exe", "uvlmRF", "CONFIG.DAT", "LS.DAT", "RotationAngle.DAT", "StrokeDe viation.DAT", "StrokePosition.DAT"}; OutputSandbox = {"uvlmRF.err", "uvlmRF.out"}; RetryCount = 0; ShallowRetryCount=0;</pre>	<pre># Environment variables export PROD=prod.vo.eu-eela.eu export LFC_HOST=lfc.eela.ufrj.br export LCG_GFAL_INFOSYS=bdii.eela.ufrj.br:2 170 export PATH=\$PATH: # Execution date chmod u+x uvlm uvlm # Storage date FILE=results03-00.tar.gz /bin/tar -czf \$FILE *.DAT *.Dat *.TEC std.out std.err lcg-cr --vo \$PROD -l lfn:/grid/prod.vo.eu-eela.eu/MAVs- Study/test03/\$FILE -d lnx097.eela.if.ufrj.br file:\$PWD/\$FILE # End Date</pre>

Inspired, Super Maneuverable, Flapping Wing Micro-Air-Vehicles is an application developed by University of Río Cuarto and University of Córdoba (Argentina) to design micro air vehicles able to be used in different contexts as atmospheric studies, fire detection, inspections of collapsed buildings, hazardous spill monitoring, inspections in places too dangerous for humans, and planetary exploration.

This application is subdivided in three stages: (1) preprocess, (2) process, and (3) post-process. The preprocess stage allows to define the start parameters for the simulation using an interactive application developed in MATLAB. The process stage computes the aerodynamic model using No-linear and no-steady vortex-lattice method (NUVLM). The final stage, developed in FORTRAN 95 and MATLAB, writes the results from the numerical simulations in an appropriate format to be interpreted by other softwares (Table 2.3).

The process stage is a batch and requires a long time to be executed in a normal workstation, about 3/4 days. Then, application developers decided to port the application on the grid infrastructure to benefit of the huge computation resources available. But the process stage has been developed in FORTRAN 95 for Windows Environment and the standard release of the gLite environment supports only Linux. Then, this application couldn't be ported on the grid environment without grid2win, the infrastructure service developed to allow Windows applications to run on gLite.

Using the gLite Windows site installed in the EELA-2 infrastructure, we were able to successfully port MAVs-Study on the grid environment. Above, we have demonstrated the JDL file and Start script used to gridify MAVs-Study while the obtained results about wings position after the first processing cycle are illustrated in Figs. 2.15 and 2.16.



**Fig. 2.15** MAVs-study wings position: Isometric view



**Fig. 2.16** MAVs-study wings position: Right view

## 2.6 Future Perspectives

The methodology to support e-Science applications introduced in this chapter allowed EELA-2 e-Infrastructure to attract and support several e-Science applications. This infrastructure will continue to be supported in the next years, thanks to new initiatives like new projects funded by the European Commission and local dedicated funds.

Indeed, one of the main objectives of the EELA project was to ensure the long-term sustainability of the e-Infrastructure beyond the term of the project. Several initiatives have been led during the EELA-2 lifetime to reach this target. In particular, we defined a long-term sustainability model, LGI (Latin Grid Initiative) that fits the Latin America reality and could be easily implemented [10, 14].

The described methodology in this chapter will continue to be used in the next years. The EELA-2 e-Infrastructure could be enriched with new additional services and, this way, the number of applications supported could be still increased. Moreover, thanks to the past experiences, the identification and the developing of new services able to satisfy specific applications requirements will be faster. In this way, the EELA-2 e-Infrastructure capabilities will grow in the years, allowing us to approach toward an infrastructure able to support almost all e-Science applications and, moreover, able to quickly adapt to new requirements and needs.

## 2.7 Conclusion

In this chapter, we described a successful methodology to support e-Science applications on an e-Infrastructure. Enriching the grid middleware adopted with additional services, developed to meet applications requirements, and to increase the appeal of the e-Infrastructure, we set up an e-Infrastructure able to satisfy the heterogeneous requirements of e-Science applications coming from different scientific domains.

We used the EELA-2 use case to show how this methodology can be applied in the real world. First, we introduced the EELA-2 project and the set of applications supported. After then, we described the services developed during the project to enrich the infrastructure. Then, we showed how these services increase the number of applications supported and make faster their deployment on the infrastructure. Finally, we explained how this methodology could evolve in the future, allowing to create an e-Infrastructure able to satisfy almost all the requirements of the e-Science applications.

## References

1. Apache Tomcat: <http://tomcat.apache.org> (2010)
2. Brasileiro, F., Duarte, A., Carvalho, D., Barbera, R., Scardaci, D.: An approach for the co-existence of service and opportunistic grids: the EELA-2 Case. In: Proceedings of the 2nd Latin America Grid International Workshop, Campo Grande, pp. 1–8 (2008)
3. Bruno, R., Barbera, R., Ingrà, E.: Watchdog: a job monitoring solution inside the EELA-2 Infrastructure. In: Proceedings of the 2nd EELA-2 Conference, Choronì (2010)
4. Cardenas-Montes, M., Emmen, A., Marosi, A., Araujo, F., Gombs, G., Terstynszky, G., Fedak, G., Kelly, I., Taylor, I., Lodygensky, O., Kacsuk, P., Lovas, R., Kiss, T., Balaton, Z., Farkas, Z.: Edges: bridging desktop and service grids. In: Proceedings of 2nd Iberian Grid Infrastructure Conference, Porto. [http://edges-grid.eu:8080/c/document\\_library/get\\_file?p\\_l\\_id=29093&folderId=11075&name=DLFE-201.pdf](http://edges-grid.eu:8080/c/document_library/get_file?p_l_id=29093&folderId=11075&name=DLFE-201.pdf) (2008)
5. CERN Large Hadron Collider (LHC): [www.cern.ch/lhc](http://www.cern.ch/lhc) (2010)
6. Cirne, W., Brasileiro, F., Andrade, N., Costa, L., An, A.: Labs of the world unite!!! J. Grid Comput. **4**(3), 225–246 (2006)
7. CREAM (Computing Resource Execution and Management): <http://grid.pd.infn.it/cream> (2010)
8. EELA: <http://www.eu-eela.org/first-phase.php> (2006)
9. EELA-2: <http://www.eu-eela.eu> (2008)

10. EELA-2 DSA1.3: <http://documents.eu-eela.eu/record/11119/files/EELA-2-DSA1.pdf?version=1> (2009)
11. EELA-2 general services: [http://eoc.eu-eela.eu/docu.php?id=central\\_services](http://eoc.eu-eela.eu/docu.php?id=central_services) (2008)
12. gLite: <http://www.glite.org> (2010)
13. Java: <http://www.java.com> (2010)
14. Marechal, B., Gavillet, P., Barbera, R.: Long-term sustainability of e-Infrastructures in LA: The EELA-2 model. Presented at CCICT Conference, Kingston (2009)
15. MySQL: <http://www.mysql.com> (2010)
16. OurGrid: <http://www.ourgrid.org> (2010)
17. Russo, D., Scibilia, F.: Grid2Win: Porting gLite to Windows Based Platforms. In: Proceedings of the 16th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, Paris, pp. 300–301. IEEE Computer Society, Evry (2007)
18. Scardaci, D., Scuderi, G.: A secure storage service for the gLite middleware. In: Proceedings of 3rd International Symposium on Information Assurance and Security, Manchester, pp. 261–266. IEEE Computer Society, Washington, DC (2007)
19. Scibilia, F., Barbera, R.: SAGE: storage accounting for grid environments. In: Proceedings of 1st EELA-2 Conference, Bogotá, Colombia (2009)
20. Scifo, S.: GSAF: grid storage access framework. In: Proceedings of 16th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, Paris, pp. 296–297. IEEE Computer Society, Evry (2007)



<http://www.springer.com/978-0-85729-675-7>

Grid Computing

Towards a Global Interconnected Infrastructure

Preve, N.P. (Ed.)

2011, XVI, 312 p., Hardcover

ISBN: 978-0-85729-675-7