CHAPTER 7

INVESTIGATING INTEROPERABILITY ISSUES IN
EU-INDIA GRID PROJECT

7.1 GENERAL

This chapter discusses the interoperability issues between European and Indian Grid infrastructure and implements methodologies to overcome some of them. The main objective of EU-India Grid interoperability project is to aggregate the computational resources available across Europe and India, and to use them effectively in solving computationally intensive scientific applications. As both European and Indian Grid are based on different middleware, the interoperability initiative leads to several issues with respect to job submission, resource management and monitoring, security, and data transfers. The European Grid is based on gLite middleware and Indian Grid Garuda is a combination of in-house software developed by C-DAC, India, Globus toolkit and industry grade components. In this work, several interoperability issues have been investigated and a preliminary interoperable system to facilitate remote job submission, resource discovery between gLite and Garuda Grid infrastructure is proposed. An ExtendedUI is created to interoperate between the middleware and its prototype implementation is presented. This ExtendedUI consists of all the libraries and adapters for job submission across globus and gLite based Grid resources. Gridseed tool is used to setup an experimental Grid infrastructure consisting of resources based on both gLite and Globus middleware and the extendedUI has been tested for job submission and information aggregation.
7.2 INTEROPERABILITY INITIATIVES

The emergence of standards related to Grid computing technology and associated tools enabled many research institutes and organizations to deploy Grid for their own applications such as Molecular Docking, Life Sciences, etc. Currently most of the countries are providing more funds to encourage research and deployment of Grid infrastructure to solve high end applications such as Disaster Management, Drug Discovery and High Energy Physics. However, the middleware that power these Grids are different. They have different security infrastructure, different resource monitoring and job management mechanisms. Further, there was no issue of interconnecting a Grid with another one during the inception phase. Hence, before the idea of standardizing the middleware architecture was initiated, different middleware were developed and they have been widely used by various research institutes to develop their Grid Infrastructure based on their needs.

Recently, there are initiatives to interconnect different national Grids. In this work, one such initiative which aims to interconnect European and Indian Grid infrastructure to aggregate huge computational resources spanning across these two Grids are discussed. Funded by the European Commission, the Enabling Grids for E-sciencE (EGEE) is a seamless Grid infrastructure to integrate current national, regional and thematic Grid efforts to support scientific research. At present it consists of 250 sites in 48 countries and 68000 CPUs to cater to 8000 users on 24X7 basis. The gLite middleware distribution developed by the Enabling Grids for E-sciencE (EGEE) project of Europe is now deployed by about 80 percent of the sites connected to the EGEE infrastructure, making it the main middleware distribution used in production. gLite makes use of components from other Grid middleware projects and is designed as a modular system to allow users to tailor the system to their specific needs by deploying the services they
require, rather than having to use the system as a whole. Garuda is one of the major Grid initiatives of Government of India to provide computing power and help research in the field of high energy physics, disaster management, astronomy, bioinformatics, etc. It connects and aggregates high end computational resources from 45 research laboratories spanning over 17 cities across the nation. Garuda middleware is built with in-house developed software, Globus Toolkit and standard commercial products. Globus Toolkit is the base on which other components are implemented. Currently Garuda is using Globus 2.4 but there is a plan to migrate to Globus 4.x version. Garuda Integrated Development Environment (G-IDE) and Paryavekshanam are software developed by C-DAC for program development and monitoring of the Grid, respectively. Moab from Cluster Resource Inc has been used for metascheduling.

The EU-India Grid interoperability initiative will lead to high level collaborative research activities between the participating organizations of Europe and India. In this case, there is a problem of interoperability which is to be solved between these two Grids as they were built using different middleware. To interoperate two Grid infrastructures each with different middleware, there can be two approaches; an adapter based approach that translates the specific design aspect from one Grid middleware to another and, a standard based approach in which the components of both the Grids follow specific international standard. But there is still no de-facto standard for most of the Grid, and developing such standard is also a very long process. So the scope of this research work is restricted to short term interoperation using adapter approach. In this work, an architecture for interoperation between gLite and Globus based Garuda Grid infrastructure is proposed. An extended User Interface (UI) is developed for job submission across both the middleware. Suitable adapters have been implemented in the extended UI for switching between middleware for job submission and resource management.
Further, gLite and Globus based Grid infrastructure have been created using Gridseed tool (2007), a training toolkit developed by ICTP, Trieste, as part of EU-India Grid interoperability project. With this architecture, it is possible to achieve interoperation and interoperability between gLite, Globus 2.4 and Globus 4.0 middleware based Grid infrastructure.

7.3 COMPLEXITY IN INTEROPERABILITY

In this section, the main issues that affect seamless joint usage of the European and Indian Grid infrastructure are discussed. The major interoperability issues which have been foreseen are listed below:

- Mechanism for aggregation of resource information
- Issues related to security infrastructure towards authentication and authorization and
- Job submission to the resources across two Grids.

An adapter based system is implemented in a centrally located user interface through which job submission to both European and Indian Grid can be accomplished. This system consists of various adapters for interoperation of different middleware components of both the Grids. The adapters proposed in our system are aimed to interoperate between gLite and Globus 2.4 and 4.x based Garuda infrastructure. This is because of the fact that Garuda Grid will migrate to Globus 4.x in future. Further, the Globus 2.4 is developed based on Component based Model whereas the Globus 4.0 version is based on Web Service framework. Hence, there is a need to develop different adapters for Globus 2.4 and Globus 4.0 based Garuda middleware.

Since core components of Garuda, like information services and job management is based on Globus, the characteristics of gLite, Globus 2.4 and
Globus 4.0 middleware are compared with respect to interoperability issues mentioned above.

7.3.1 **Globus vs gLite - Information Service**

The Information Service in both Globus 2.4 and gLite middleware is built with MDS component of Globus middleware. The MDS in Globus 2.4 is implemented with two services, namely Grid Index Information Service (GIIS), which provides an aggregate directory of lower-level data, and Grid Resource Information Service (GRIS), which runs on a resource and acts as a modular content gateway for the resource. A GRIS registers with a GIIS, and one GIIS may register with another using a soft-state protocol that allows dynamic cleaning of dead resources. However, for scalability reason in gLite middleware, a new component called the Berkeley Database Information Index (BDII) is used to store and publish data from the local GRISes. Although both middleware use LDAP for storing low level resource information, they use different schemas and hence publish almost the same information but in different formats. Data in the MDS in gLite conforms to the LDAP implementation of GLUE schema, whereas in Globus, it follows its own schema called MDS schema. So by querying LDAP database, it is not possible to obtain resource information of both the middleware. Further, in Globus 4.0, the MDS component has been implemented as a web service called WS-MDS. It has its own web service interface for interacting and aggregating resource information from the underlying middleware.

Hence, separate adapters have to be developed to query the information sources from these middleware.
7.3.2 Globus vs gLite – Execution Management Service

In gLite middleware, Workload Management System (WMS) component governs the job submission and execution. In addition to this, it can also discover suitable Grid resources across the Grid that matches the application requirements with the help of Condor Matchmaking system implemented in WMS. The Execution Management service in Globus is implemented in GRAM component. GRAM simplifies the use of remote systems by providing a single standard interface for requesting and using remote system resources for the execution of jobs. GRAM is designed to provide a single common protocol, API for requesting and using remote system resources, a uniform, flexible interface for local job scheduling. Jobs are described using Resource Specification Language (RSL). In Globus 4.0 version, this component is implemented as a web service called WS-GRAM. Hence, it has web service interface for job submission and also RSL used to describe the job is in XML format. With this diverse nature of components, there is a need for an adapter to negotiate with both the middleware for job execution on behalf of user from a single UI.

7.3.3 Globus vs gLite – Grid Security Infrastructure

The Grid Security Infrastructure of both gLite and Globus enable secure authentication and communication over an open network. It is based on public key encryption X509 certificates, and the secure sockets layer communication protocol, with extensions on single sign-on and delegation.

However, in gLite, an additional service called Virtual Organization Management Service (VOMS) is included which provides information about the roles and privileges of users within the VO. In gLite, X509 certificates are created with VOMS extension. So there is a need for a
component that can map the certificate of Globus users to gLite based Grid resources and vice versa.

### 7.4 INTEROPERABLE ARCHITECTURE

To address the complexity while interoperating, the architecture, shown in Figure 7.1, is proposed which includes necessary components such as User Interface, Job Submission adapter and Information service adapter which have been described in detail in the following sections.

![Figure 7.1 Interoperability Architecture](image)

**The User Interface** component contains necessary functionalities for interacting with Grid environment transparent to the underlying middleware. It includes Globus client libraries as well as gLite based User Interface libraries for job submission across both the Grids. It enables the user to query the top BDII of the gLite Grid and WS-MDS of Globus based Garuda Grid for obtaining resource information. It can also interact with WMS of gLite and WS-GRAM component of Globus for job submission. It
allows the user to submit the job described using JDL as well as RSL so that they can run their job either in gLite or Garuda resources.

The **Information Service adapter** aggregates the resource information across Globus and gLite Grid middleware. Since the information schema used for publishing resource information in these middleware is different, an adapter is needed to contact the respective middleware for collecting resource information. In gLite middleware, the MDS collects the resource information and stores it in LDAP server. This resource site information is then stored in BDII. The Information service adapter implements a script that contacts the BDII for retrieving resource information across gLite based Grid. Similarly, in Globus based Grid, the resource information is obtained through LDAP server and registered with GIIS. In this case, the adapter uses a script to contact GIIS for resource information retrieval in Globus based Grid. This information shall be effectively used for discovering suitable resources depending on the requirements of the job submitted by the user.

The most important component in the architecture, the **Job Submission Adapter (JSA)**, contains appropriate software to submit jobs to suitable resources present across both Globus and gLite Grid infrastructure. The gLite middleware provides in-built matchmaking mechanism through its WMS component. However, in Globus middleware, the matchmaking component is not provided and it needs to be developed externally. Hence, in Garuda Grid, a semantic based resource discovery mechanism described in chapter 4 is implemented which performs matchmaking of resource request against the available resources. It uses an ontology representation of available Grid resources thereby forming a resource ontology knowledge base. Algernon inference engine is used to retrieve resource information from the knowledge base. The Grid Resource information is populated onto the
ontology by interacting with information database maintained in Garuda Grid. As soon as a suitable resource is discovered, depending on the middleware installed in it, the adapter prepares the job and schedules it to that resource. For instance, if a user submits his job described in JDL, it contacts the WMS component of gLite Grid for finding suitable resource. If a resource that matches the requirements is found in gLite Grid, then the job submission adapter sends the job to WMS of gLite for execution. If the suitable resource is not available in gLite Grid, it then invokes the information service adapter. If a resource is found for job execution in Garuda Grid, job submission adapter converts the JDL description of the job into RSL and schedules the job to Globus using the Globus client libraries. Similarly, if the user submits the job described in RSL, it contacts the matchmaking mechanism of Garuda Grid to find which in turn queries WS-MDS for suitable resource. If found, it schedules the job to that resource through GRAM component of Globus middleware. If not, then, it contacts information service adapter, which in turn queries top BDII of gLite for resource information. If resource is available, then, Job Submission adapter converts the RSL description to JDL and submit the job to WMS of gLite Grid which in turn discovers suitable resources and submits the job to it.

In this architecture, there is a need for certification authority to establish secure communication channels which enables mutual authentication and authorization between the Grid resources. The complexity in this case is the additional attribute in the form of VOMS in gLite X509 certificate in gLite Grid. In this architecture, it is proposed to use Grid proxies without VOMS extension when submitting job to Globus resources and create VOMS proxies when submitting job to gLite resources.
7.5 EXPERIMENTATION WITH GRIDSEED

The proposed architecture has been tested in a controlled Grid environment created using Gridseed (GS), a tool developed by the EU-India Grid project that simplifies the setting up of a full fledged Grid infrastructure based on the gLite middleware by means of VMware machines.

Gridseed consists of a set of VMware virtual machines, each them dedicated to a specific Grid element. In order to install configure and work with a gLite Grid, the basic hardware requirement is to have enough physical machines connected to local area network with the VMware player and/or VMware server installed. VMware based Gridseed machines are then booted in a specific order: all central services (CA/VOMS/TopBDII/WMS etc) must be up and running before a Grid site can operate. The configuration of each service proceeds in parallel with its installation and hence there is, therefore, a precise ordering for the configuration and booting of a full gLite Grid. The first machine to boot up is CA. It consists of an Apache web server, a DNS, an implementation of NTP server and a Certification Authority based on scripts. For other services, a script is provided in ‘/etc/init.d’ so that, when a particular service is started on new virtual machine, it takes host certificate automatically from CA using client-server model, where CA is the server.

All the virtual machines are synchronized through NTP server provided in CA. Different scripts are available to configure services depending on the infrastructure. For example, if two CE’s were booted, then a script would add this information in BDII’s bdii-update.conf file so that it can fetch information from CE. Gridseed encapsulates this knowledge and leverages on it when automatically configuring a Grid on the fly. Gridseed is designed to create and configure a gLite Grid made up of 1 WMS, 1 Top BDII, 1 LFC, 1 VOMS, up to 99 UI, up to 255 Sites each one consisting of 1
CE + 1 SE + up to 252 WN. More information about Gridseed can be found in https://euindia.ictp.it/Grid-seed.

**Figure 7.2 Experimental Setup for Interoperation**

In our experimental setup for interoperability experiments, Gridseed is used to create gLite based Grid infrastructure first. The gLite UI machine allows users to submit job to the gLite Grid. A script is provided in this machine to obtain user certificate for every user from CA. The other central services were booted and up for proper operation of Grid followed by a Computing Element and a set of Worker Nodes. This infrastructure has been then tested for proper operation such as resource discovery, job submission through WMS.
Once Gridseed Grid infrastructure is setup, a separate VMware based Virtual Machine was booted and Globus middleware version 4.0.0 is installed. Host certificate and User certificate for this machine have been obtained from Gridseed CA machine after properly modifying the CA script provided in Gridseed. Also, this machine is time synchronized with gLite Grid by editing the NTP script present in Gridseed CA machine.

In our adapter based approach, the users can submit their job described either with JDL or RSL. If suitable resources are not found in one Grid infrastructure, the adapter can prepare the job for the other middleware and submit. The proposed adapter components must be installed in a machine in which both gLite based UI and Globus client libraries are available. In the Gridseed based experimental setup, GT4 client in gLiteUI machine is installed for cross job submission between Globus and gLite middleware. It has been termed as extendedUI because all the proposed adapters have been implemented in this UI as shown in Figure 7.2.

![Flowchart of the Interoperation](image_url)
A software has been written in UI machine which can accept job from the users that is described either by JDL or RSL and is sent to switch module. If JDL is submitted, the switch would try to submit the jobs to gLite resources. The gLite resource discovery module queries the ‘topBDII’ through WMS for finding out suitable resources matching the requirements specified in the JDL. This module has been implemented using the user interface command provided by gLite middleware to query information services. However, if the resources are not available in gLite Grid, the switch component will convert the JDL into RSL and sends the job to Globus job adapter. The Globus job adapter is implemented using GT4 client libraries and hence the job can be scheduled to Globus 2.4 as well as Globus 4.0 middleware. Similarly, if RSL job is submitted, the adapter would try to submit the jobs to Globus resources. If the resources are not available in Globus Grid, the switch component will convert the RSL into JDL and sends the job to gLite job adapter. The gLite job adapter is implemented using gLite based UI libraries with which the job can be scheduled to gLite WMS middleware. This entire flow of operation is depicted in Figure 7.3.

In this work, an architecture based on extendedUI is proposed to achieve interoperation and interoperability between gLite and Garuda Grid. The centrally located extendedUI can also be implemented as a web service acting as Grid portal through which users can submit job. This architecture does not require any piece of software to be installed in the participating Grid resources and hence preserving their autonomy. The extendedUI can be developed as an independent Grid component which can be downloaded and installed by the users who wish to use both gLite and Garuda resources.