GRID Service and Access Management within User Service Environment

1. Introduction

One of the most important issues concerning utilizing the GRID resources is the management of access to the GRID and its services. The PROGRESS project, currently under development at the Poznań Supercomputing and Networking Center, introduces the User Service Environment, which serves as a layer for grouping the functions of a GRID access environment needed for making the use of the GRID resources most comfortable to the end users. The most important USE functions are implemented in the PROGRESS USE framework, which is presented in this paper.

2. Overview

The User Service Environment (USE) can be understood as a group of GRID service providers. It provides services that facilitate building computing portals and other client applications accessing the GRID. The most important functions of USE are:

- user authentication and authorization
- providing access to computing applications
- providing workgroup services
- services management

USE is a base for building GRID access client applications. The placement of the User Service Environment within the GRID environment is presented in **Fig. 1**.

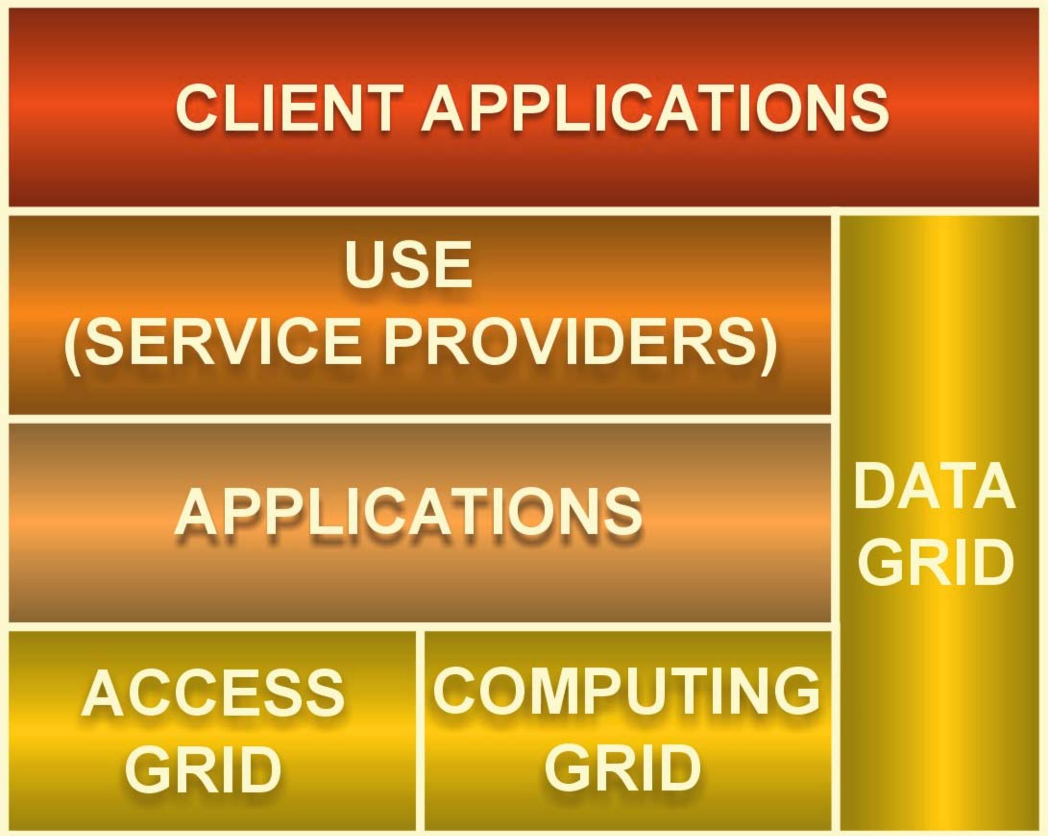

Fig. 1 The placement of User Service Environment within the GRID environment

3. USE Service provider

A typical USE service provider consists of the following modules:

- kernel
- services
- applications
- means of communication with GRID brokers (a computing broker, a data broker and others)

The kernel is responsible for user authentication and authorization.

Services provide users with the possibility to access computing applications, submit computing jobs, communicate with each other, share their work results and manage the GRID environment. USE introduces a separation between the presentation layer and core services. Thanks to such approach, the services may now be grouped into a service provider and the presentation layer may be implemented by a client application of the service provider. There are no limitations imposed on the form of client applications: they can be stand-alone personal computer applications or typical web applications. The grouping of services allows creating many client applications that use one and the same service provider for user request handling. Moreover, one client application may simultaneously use two or more service providers for user request handling.

Applications are programs that are run by services to perform time consuming tasks. Examples of applications include computing applications and data searching applications.

Client applications, built on top of USE, are responsible for presenting data to the users. They obtain the data from service providers and the Data GRID (structure of this data is known to the USE client applications).

Service providers are responsible for handling requests from users of computing applications. Service providers allow accessing computing applications that are run in the Computing GRID. The Data GRID is used as a source of input data and a destination of output data resulting from computing experiments. USE client applications communicate with the data broker directly, bypassing service providers.

4. PROGRESS USE Framework

A typical GRID service provider within the User Service Environment is illustrated in **Fig. 2.** Such a service provider can be built using the PROGRESS USE Framework.

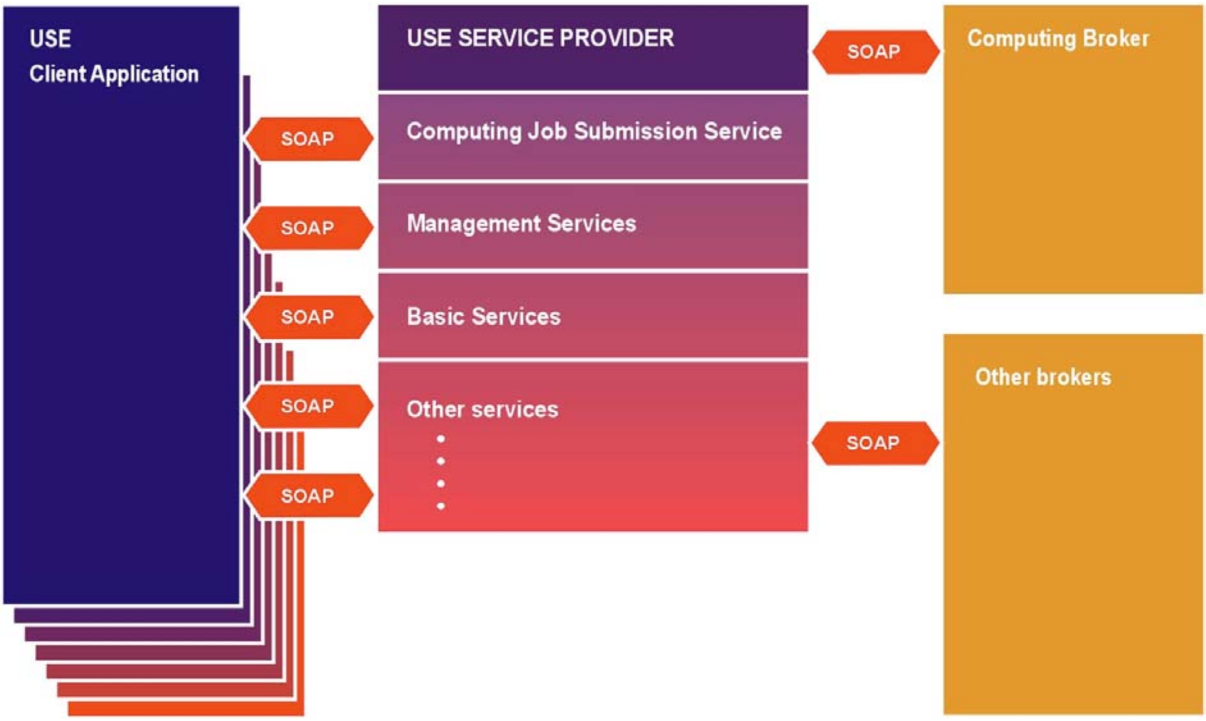

Fig. 2 GRID service provider within User Service Environment

The PROGRESS framework includes:

- user authentication
- authorization of user actions
- management services for computing applications and USE services
- a computing job submission service
- basic portal services
- a module for communication with the Computing Broker

The above makes a complete service provider, ready to handle requests coming from GRID client applications built on top of this service provider.

Each of the framework modules is built as a set of Enterprise Java Beans accessible via a Web Service. The 2.0 version of EJB specification is used for building the framework, thus allowing to use any EJB application server (with EJB 2.0 support), connected to any database system. The responses to client applications requests are sent in the form of XML documents.

User Authentication

Users log into the USE using their unique identifier and password. After a successful authentication a user session is created. The identifier of the session is then used as a key for authenticating each user request.

Authorization of User Actions

User actions are authorized according to the information about their resource access rights. This information is stored in a database of an authorization service. Before executing a user request, a given USE service sends an authorization request to the authorization service, asking whether the user can perform a particular action on a particular resource object belonging to a given service. The authorization service answers “YES” if the user is allowed to perform the action, or “NO” if the action is not allowed for this user.

Computing Application Management Service

The computing application management service keeps up-to-date information on applications available for execution within the GRID. It allows adding and removing applications, and also updating their information. The binary code of an application is stored in the Data GRID. The application management service keeps only the Data GRID object identifier to easily find the location of the code. Each application has its set of arguments. Users may set them to customize the behavior of the application according to their needs.

USE Services Management Service

The USE services management service keeps up-to-date information on services available in the service provider. Each service is described with a URL and a URN of the web service serving its data. A service may be of single or multiple instance type. If a service can have multiple instances, it is allowed to create, remove and manage its own instances. Each instance of a service may be understood as a separate virtual object within the service, with its own space in the service database.

Computing Job Submission Service

The computing job submission service allows users to submit computing jobs to the Computing GRID. There is a XRSL (XML Resource Specification Language) document automatically generated for each job. This XRSL document is generated by the job submission service based on the requirements for the job entered by the user - if present - or based on default requirements for a given application – if the user does not specify any requirements. The XRSL is later used by the Computing Broker.

Basic Portal Services

Basic portal services are simple information sharing services. These include a news service, a link directory service and a discussion forum service. The news service allows presenting users with short messages. In addition, a message may be illustrated with a set of web links. The link directory allows creating a structured folder tree with information about web pages located in the directory folders. Information about web pages includes the title, the description and the link. Finally, the discussion forum service offers users a possibility of discussing various topics. Users may write their messages and send them to a forum, read and answer messages written by other users. The news, link directory and discussion forum services are examples of services offering multiple instances.

Communication module

The communication module is a library of java classes allowing communication with any Web Services server. The job submission service uses it to transfer XRSL documents containing computing job descriptions to the Computing Broker and also to check the status of a job submitted to the GRID.

5. PROGRESS USE Client Application

The PROGRESS USE client application is a set of Java servlets. The servlets are presentation modules for all of the framework services. Each application uses an additional library for communication with the USE via Web Services. This library eases the process of USE client development.

Servlets

The set of servlets of the client application facilitates a quick development of a USE client web application, with the use of a chosen deployment environment. The tested environments include Sun ONE Portal Server and Joshua Portal Framework. Each of the servlets is a complementary presentation module for a PROGRESS framework service (which could be understood as a service functional module). These servlets are used for:

- computing application management
- portal services management
- computing job submission
- basic portal services

The servlets use XSL Transformations as a means of translating XML data received from the service provider into HTML documents sent to the user browser. The communication with a service provider is performed using the PROGRESS communication API.

PROGRESS Communication API

The PROGRESS communication API provides Java classes allowing a simple invocation of a PROGRESS framework service method. The API is used by the client servlets and may be used by any other Java client application. The API methods for execution of framework service methods return the response as an XML document of a structure defined by the service. The API uses the SOAP protocol for communication with a service provider.

6. PROGRESS testbed

The PROGRESS USE framework is used for developing a bioinformatics service provider testbed. The testbed is being equipped with three test applications:

- DNA assembly
- Secondary protein structure prediction
- Constructing filogenetic trees and motif search

The applications are used for conducting experiments in a cluster of SUN machines constituting the PROGRESS GRID. The USE framework is deployed on the Sun ONE Application Server with an Oracle database instance serving as the service provider database system. There are two PROGRESS USE client applications – a bioinformatics computing portal and a migrating desktop. The migrating desktop is an example of a stand-alone java client application. The USE client application is implemented on both Joshua Portal Framework and Sun ONE Portal Server.

Although the testbed uses tools designed for the PROGRESS project, the USE framework could be deployed to work with any computing GRID. The framework communication module can communicate with any computing GRID broker (preferably using the Web Services technology).