ISSN 1392 – 124X INFORMATION TECHNOLOGY AND CONTROL, 2007, Vol.36, No.1A

SANDS VO: E-LEARNING SYSTEM ARCHITECTURE FOR GRID ENVIRONMENT

A. Jukniute, G. Paulikas

Kaunas University of Technology
Studentų St. 50, LT–51368 Kaunas, Lithuania

Abstract. Network oriented e-learning systems meet common challenges of modern distributed systems and need a secure and effective way to manage and reuse shared resources in the constantly changing heterogeneous environments. Web service based Service Oriented Architectures and Grid software provide a technological and infrastructural background for advanced realization of the systems supporting dynamic Virtual Organizations (VO). The material of this paper is drawn from the experience gained during the implementation of a prototype for the Science and Studies (SandS) VO system on Globus Toolkit of Grid middleware. The paper presents the architectural framework of SandS VO focusing on security issues and dynamic nature of the distributed learning environment which is illustrated by the first fully implemented service of this e-learning system – the Assessment Service.

1. Introduction

In recent years, e-learning has gained the great popularity as a convenient way of providing teaching methods and sharing knowledge all over the world. With the evolving development of network technologies the amount of information and services available on the Internet continues to rise. No doubt this tendency will last in the future. But currently e-learning systems are reaching their limits due to the limitations in resource sharing, management, scalability and integration in the dynamic environment [1].

At present, Service Oriented Architecture (SOA) is being promoted as the next evolutionary step in software architecture. Traditional architectures provide statically organized content, distributed, but not connected resources; on the contrary, SOA allows sharing of information and resources including distributed resource integration. In fact, SOA is the set of services, which are autonomous, discoverable and communicate among themselves. The communication of services is independent, loosely coupled and is defined using standard description language. SOA and service-orientation are implementation independent paradigms that can by realized using any suitable technology [5]. Today SOA Web services are commonly based on open standards (WSDL, SOAP, UUDI).

Though many organizations have adopted e-learning, the software and hardware facilities they use vary greatly. This causes major difficulties in sharing teaching resources. Web service techniques enable the integration of different information systems within grids and solve this problem [12].

While the Grid is often thought of in terms of providing a distributed system of high-performance compute resources, this is only one aspect of successful use of Grid computing [10]. We can consider distributed computing resources as services, delivered by different organizations, which are used for enriching and improving e-learning environment. Grid technologies can satisfy the needs of effective e-learning system – dynamic resource and user management, resource sharing, integration and discovery, security among different, multi-institutional administrative domains.

With the Globus Toolkit (GT4) move to the Open Grid Standards Infrastructure (OGSI) and finally to Web Services Resource Framework (WSRF), Grid has matured to its latest development stage, adopting a service-oriented approach. WSRF defines conventions for managing state so that applications can reliably share changing information. In combination with WS-Notification and other WS-* standards, the result is to make grid resources accessible within a web service architecture. GT4 for grid computing emerged as a middleware that addresses the common issues of web service based distributed systems, providing the building blocks for constructing the secure distributed systems for VOs.

2. Virtual Organization

In a broad sense of informatics a **virtual organization** (VO) is defined as one or more organizations interconnected and augmented by means of modern information and communication technologies. The driving force of virtualization includes the rapid change of the market, the availability of advanced technology and the need to increase efficiency. Various dimensions of an organization are apt to virtualize: location, interfaces and boundaries, processes, structures and products/services [7].

The more precise way to describe a VO is achieved by identifying the goals and of connecting the concerned organizations and individuals into the single entity and the means that glue separate organizational pieces together. I. Foster et al. [6] specifies the VO as a set of individuals and/or institutions defined by sharing rules aimed at coordinating the resource sharing and problem solving in dynamic multi-institutional environment. Sharing implies direct access to various computational resources: processing cycles, data, software, devices, etc. This requires the virtualization of accessed resources and full control over the process of providing and consuming them. The rules of sharing are based on the security of communication among partners participating in the VO system. Authentification and successive authorization of subjects engaged in pursuing common goals is essential to control the processes in the VO (scheduling, load management, data replication, information services, etc.).

The broad scale of emerging VOs and heterogeneity of participating entities foster a new wave of technologies for VO implementation. The loosely coupled service based architectures based on standard internet protocols ensure the level of interoperability that is crucial to fulfill these requirements. An equally important component of modern VO is the infrastructure that facilitates the implementation of VO system by supplying common building blocks. Architectural and some technical issues are addressed by SOA, while infrastructural aspects are encountered in **grid** computing. Merging the features of both innovative computing models allows to separate applications from services (SOA) and both applications and services from underlying infrastructure and system resources (grid) [8]. The combined solution empowers the effective implementation of dynamic distributed systems for VOs, improving availability, reliability and scalability of resource utilization when pursuing common VO goals.

3. System Architecture of Science and Studies VO

The current trends of ubiquitous dynamic e-learning VOs leverage the development of open network environments with systems of distributed learning resources to support them. The ability to successfully employ multiple resources of heterogeneous VO members can be ensured by relying on open standards of communication, namely Web services. Other features of such distributed software architectures are usually encountered in enterprise systems and involve high levels of security, control and dynamism of distributed resources.

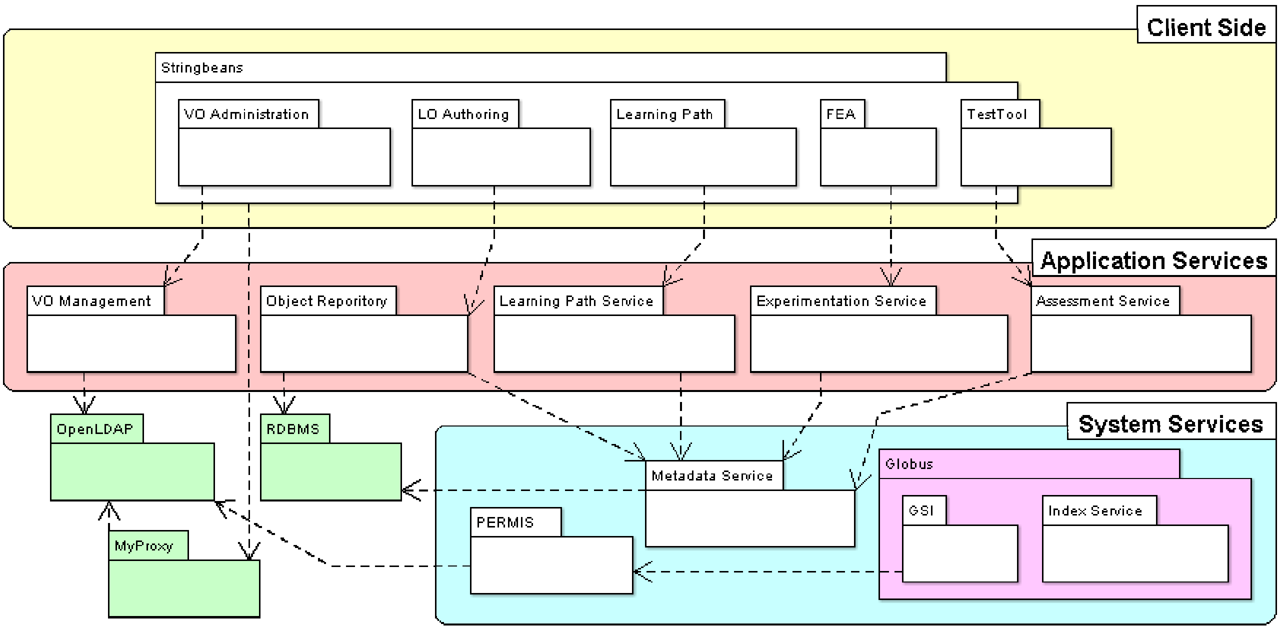

Figure 1. SandS VO architecture

EU funded research and implementation SPD (Single Programming Document) project at Kaunas University of Technology is aimed at improving the quality of human resources by creating the **Science and Studies** (SandS) VO based on grid technologies. The technological result of this project will consist of the implementation of distributed e-learning system that will be deployed in campus computing nodes and populated with sample learning resources. SandS VO system implementation will benefit both the potential users of the e-learning material of VO and as a reference for future developers of similar distributed environments. As project approaches its midpoint, the design of the system framework is finished and some pilot learning services are partially implemented. This paper presents the overall architecture of the VO system and elaborates on main challenges met during system design phase.

The system architecture of the SandS VO encompasses three conceptual layers:

- **System services and backend software.** The core of this distributed system consists of Globus grid middleware and supporting system services. SandS VO uses just a subset of default Globus services, basically **Globus Security Infrastructure** (GSI) [11] and **Index Service**. GSI ensures the required level of security while Index Service provides a service registry that glues the dispersed learning services into solid organization. These systemic Globus services are mandatory for all SandS VO educational services.

  The system layer also has two non-Globus services, supplementing the functionality of GSI and Index Service. **PERMIS** authorization service is used to control access to learning resources across VO and is tightly coupled with authentification duties of GSI. **Metadata Service** is developed exclusively for this project and assists in accumulating, publishing and searching of meta information about the static contents (**Learning Objects**, or LO) and services scattered over SandS VO deployment nodes.

  System layer backend servers provide the data storage utilities for information about VO members as well as for the data of some application layer services. SQL **relational database managements system** (RDBMS) fulfills the needs of simple and effective storage engine encountered in Metadata Service and other services from the application layer. **OpenLDAP** and **MyProxy** servers keep authentification and authorization information of the VO members in the form of X.509 certificates along with personal profiles of VO users.

- **Application services** expose the educational resources of SandS VO to their users. These services are created from scratch (Learning Path Service), reengineered from existing learning software to make them grid-aware (Assessment Service) or provided as wrappers over proprietary systems (Experimentation Service) to accommodate the needs of the distributed e-learning environment. **Assessment Service** has the most advanced realization stage so it is used as a sample case of grid service implementation further in the article. **Learning Path Service** allows to create the knowledge acquisition scenarios by composing the new services from already available service instances and follow the constructed paths to pursue individual learning goals.

  The other two services, **VO Management** and **Object Repository**, support the backbone of SandS VO by providing convenient access to VO member information for administrators and LO storage space for authors of educational material.

- **Client Side.** Considering the demand to enable the easy user access to VO resources from any networked machine the SandS VO project chose a web based portal for its UI representation. Distributed nature of VO system and employment of web service technologies led to the choice of **Stringbeans** JSR-168 portal with WSRP support [4]. The GUI portions of SandS VO e-learning subsystems are implemented as Java portlets that deliver the functionality of the corresponding grid services to the user. In cases like **TestTool** portlet when sophisticated GUI can't be implemented using plain HTML markup the external program is started from the portlet. The separate security scheme of Stringbeans portal requires synchronizing the information of authentification and authorization of client side and system layer security domains (more on this in the following section).

3. SandS VO Security Subsystem

The security framework of SandS VO distributed system is built on GSI that ensures the fundamental aspects of safe communication among members of VO and provides full control of the shared resources. Globus toolkit offers multiple security features:

- **Communication protocol security** guarantees privacy and integrity of the transmitted information and authentification of the parties involved in the communication. Globus gives a freedom of choice between the transport level and message level (SOAP) security schemes. The latter is available in Secure Message or Secure Conversation forms. SandS VO uses Secure Conversation that is more flexible than transport security, more effective than Secure Message and is the only scheme that supports credential delegation.
- X.509 PKI certificates provide the cornerstone of VO member **authentification** and are required for the full-fledged security of communication. As manual management of digital certificates by users themselves has no substantial advantages and even may create a breach in system security, PKI certificates are stored at server side and retrieved on demand. MyProxy credential management service [9] is a central repository of security credentials and ensures the retrieval of user **proxy** certificates that are obligatory for credential delegation and single sign-on to the VO.
- **Authorization** policy of the authentificated VO members enforces the sharing rules of resources that are made available to the VO partners. As most access control decisions are made by services that expose some functionality to VO users, client side authorization must be present only for credential delegation. Server side authorization is the place that concentrates the majority of access control

logic to VO resources. Globus, as usual, presents several options for the given task: clients can be authorized by 1) ACL-like (Access Control List) gridmap files, 2) host credentials or 3) SAML-callouts (Security Assertion Markup Language) that delegate authorization decisions to OGSA-Authz compliant service. SandS authorization policy avoids the first two methods as they make access control administration cumbersome and limit the desirable dynamic characteristics of the distributed VO system.

SAML-callout server side authorization enables to delegate access control to the third-party **PERMIS** service that makes authorization decisions using RBAC (Role Based Access Control) [2, 3]. Besides access resolution at run time PERMIS has the facilities to manage the authorization policies and user privileges. PERMIS authorization is built around the notion of a **role** – a user is granted one or more roles while a service exposes different subsets of its functionality to different roles. Role assignment utilizes the X.509 PMI (Permission Management Infrastructure) attribute certificates that are saved in SandS VO LDAP server. Authorization policies are expressed in XML files that are installed at the authorization decision making engine.

- Arbitrary **levels of security** (container, service and resource) allow to choose the required level for SandS VO system by tightening the security measures (e.g. when client side authorization is required for credential delegation) or liberating them (e.g. for operation of system layer Index Service).

The elements of the security subsystem of SandS VO system and application service layers fit together seamlessly as they are derived from a monolithic Globus framework. A different picture is emerging at the boundary between Stringbeans user GUI authorization and authentification and the lower security layers. Stringbeans possesses an autonomous mechanism for user authentification based on internal database, JAAS (Java Authentication and Authorization Service) or LDAP. While LDAP authentification is suitable for overall SandS VO architecture, PERMIS roles are saved as attribute certificates and can't be acquired directly from LDAP server (though their certificates are stored there). This raises the need to develop an interface to service layer authorization data that will be used by Stringbeans portal to extract approximate user permissions to access services. This information is used to construct a portal view for the logged in user. In case the real access to the functionality provided by some service is denied the corresponding portlets should be closed gracefully and that is achieved by direct SAML calls to PERMIS service (see Figure 2).

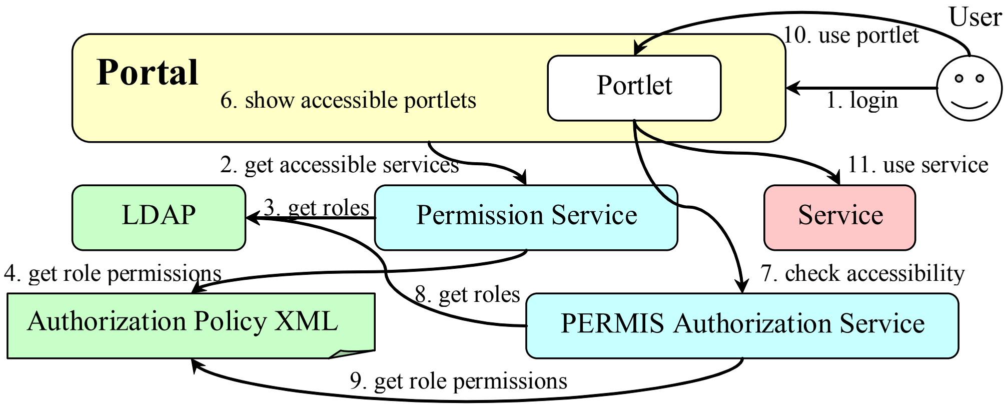

Figure 2. Interactions of portal UI and service layer elements for the synchronization of authorization

4. GT4 Based Assessment Service

The Assessment Service (AS) designed for the first SandS prototype is one of the first fully realized VO learning services. Having this service as a sample case for a grid service realization, the paper describes service's architectural view and implementation details.

Following the SandS VO vision – to provide interactive learning services with modeling capabilities, an existing application – TestTool (TT) was used for AS realization. TT is highly interactive assessment system developed in Kaunas University of Technology based on the graphical components. TT was modified and redesigned to make it grid-aware.

Further in this article the architecture and implementation details of AS are presented. Figure 3 shows the inner design of AS, identifying the communicating services and user interaction through VO portal.

When migrating to the new grid version of TestTool, two main architectural modifications were made. First, TT system users and assessment engine were separated in the service implementation relying on global VO user authentication and authorization policies. Second, the idea of test composition from distributed learning objects, hosted in different environments, led us to the solution where TT questions are not stored in AS but rather discovered through Metadata and retrieved from the Object Repository

Services. Moreover, such an implementation ensures global access control and better reuse possibilities avoiding the data duplication.

The kernel of Assessment Service actually follows factory pattern. The use of factory pattern enables the dynamic creation of Grid service instances based on the Grid service description. Testing Factory is always available static service responsible for new Testing Services instances and related resources creation. This approach gives us the possibility to create new tests on demand, considering test as a temporary service, thus having a limited life cycle.

Testing Service is one Web Service implementing TestTool logic, which can be divided in two logical parts – Tutor and Students operations. The state information of Testing Service contains some basic metadata of particular test and is kept separated in the Testing Resource. The exposed state is relevant when publishing created test instance in the global services registry – GT4 Index Service.

The user of service can have a (PERMIS) role of Student, Tutor or Researcher. Researcher is the role which encompasses both Student and Tutor capabilities when using AS. AS communicates with other VO services and in order to be properly authorized by these services, it must perform the request on user's behalf. User credential delegation is performed when the connection is made to other service which requires authorization.

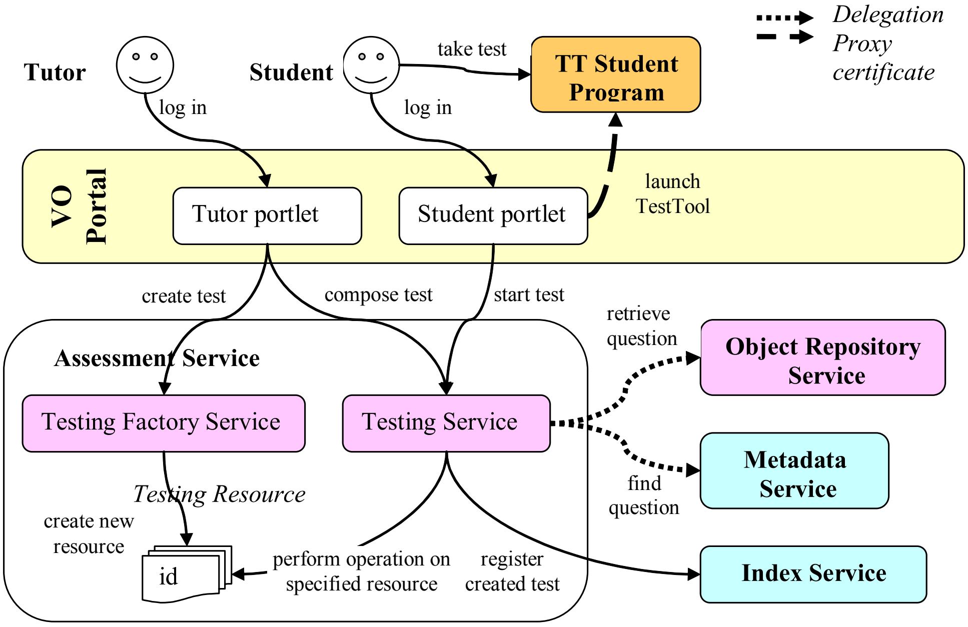

Figure 3. Relations between Assessment Service and other VO components

Depending on the user's role, different TT portlets are shown when logged in the portal. While having simple portlet like as a GUI for the service, the authentication and authorization are performed seamlessly as the portal already has the user proxy certificates, obtained at the time user logged in. This is the case with the Tutor portlet. The Student portlet represents much more complex scenario. TestTool graphical environment can't be implemented using HTML that's why the portlet contains the link, which launches external Java application at client's computer. Security is ensured by passing the user credentials from the portal to application at run time using secure connection.

References

[1] **Z. Abbas, M. Umer, M. Odeh, R. McClatchey, A. Ali, F. Ahmad.** A Semantic Grid-based E-Learning Framework (SELF). *5th IEEE International Symposium on Clust Computing and the Grid Cardiff, Wales, UK* 9th – 12th *May* 2005, *http://arxiv.org/ftp/cs/papers/0502/0502051.pdf.*

[2] **D.W. Chadwick, A. Otenko, E. Ball.** Implementing Role Based Access Controls Using X.509 Attribute Certificates – the PERMIS Privilege Management Infrastructure. *Security and Privacy in Advanced Networking Technologies, NATO Science Series, IOS Press*, 2004, 26-39.

[3] **D.W. Chadwick, A. Otenko.** RBAC Policies in XML for X.509 Based Privilege Management, Security in the Information Society: Visions and Perspectives. *IFIP TC11 17th* Int. *Conf. On Information Security* (SEC2002), *Cairo, Egypt, Kluwer Academic Publishers, May* 2002, 39-53.

[4] **W. Chokry.** StringbeansTM Installation and Setup Guide. *Nabh Information System*, 2006, *http://www.nabh.com/doc/sb_install_guide.html.*

[5] **T. Erl.** Service-Oriented Architecture (SOA). *Concepts, Technology, and Design*, 2006

[6] **I. Foster, C. Kesselman, S. Tuecke.** The Anatomy of the Grid, International Journal of High Performance Computer Applications, 15(3), Fall 2001, 200-222.

[7] **D. Gould.** Virtual Organization, 2000, *http://www.seanet.com/~daveg/Virtual%20Organizing.pdf.*

[8] **E. Kourpas.** Grid Computing. *Past, Present and Future, IBM, June* 2006.

[9] **J. Novotny, S. Tuecke, V. Welch.** An Online Credential Repository for the Grid. *MyProxy, Proceedings of the Tenth International Symposium on High Performance Distributed Computing* (HPDC-10), *IEEE Press, August* 2001, 104-111.

[10] **V. Reklaitis, K. Baniulis, T. Okamoto.** Shaping e-Learning Applications for a Service Oriented Grid. *Proceedings of* 2*nd International LEGE-WG Workshop on e-Learning and Grid Technologies: a Fundamental Challenge for Europe, Paris, France* – 3*rd* & 4*th March* 2003, *http://www.lege.ktu.lt/pdf/200303_paris.pdf.*

[11] **B. Sotomayor.** The Globus Toolkit 4 Programmer's Tutorial, Chapter III. GT4 *Security, University of Chicago*, 2005, *http://gdp.globus.org/gt4-tutorial/multiplehtml/pt03.html*.

[12] **Ch. Yang, H. Ho.** An e-Learning Platform Based on Grid Architecture. 2005, *http://www.iis.sinica.edu.tw/JISE/2005/200509_06.pdf.*

Received March 2007.

DOI: 10.5755/j01.itc.36.1.11821