# Applying Grid security and virtual organization tools in distributed publication databases

Marko Niinimaki and Vesa Sivunen
Helsinki Institute of Physics, CERN Offices
{Marko.Niinimaki, Vesa.Sivunen}@cern.ch
CERN / EP
1211 Geneve 23
Switzerland
`http://wikihip.cern.ch`

April 30, 2003

**Abstract**

In a single document repository or a library system, the authentication and authorization issues can be handled relatively simply with traditional technologies. However the situation is different in distributed and heterogeneous publication databases. The European DataGrid database access and security design is, among other things, able to take care of the authentication of each user, and the roles of different users across organizational borders. In this paper, we suggest how these Grid tools can be used for accessing distributed publication databases in a controlled and secure way.

## 1 Introduction

In the context of a single publication database, authentication and authorization ("who is this user", "what queries or updates is he authorized to issue") can be handled with conventional techniques. For instance (i) a library user is provided with a browser interface and the library catalog web application has the right to query everything in the database, whereas (ii) the information updater interacts with the database using an application that has the rights to update everything.

Conditions are different with a distributed system. Even if the database updates are managed locally, the publishers often want to limit the query authorization. For instance some publications are accessible to everyone, and some only for customers that have an agreement with the publisher. Moreover, the database implementations can vary, and thus a direct mapping of user X with a database role "can query publications Y" in each database is not practical.

The problems can be solved with technologies and software that exists already and that is freely available. In our paper, we present European DataGrid database access and security design, and discuss how it can be utilized in connection with distributed publication databases. The main parts of the system are called TrustManager and Virtual Organization Management Server (VOMS). TrustManager takes care of authentication of one user to one server at a time, whereas VOMS handles information relates to roles of different users across organizational boundaries. All the communication explained below is based on digital certificates (see [7], [6]).

## 2 Technology and terminology

### 2.1 Grid

Our design applies Grid technologies. Foster and Kesselman describe the Grid as follows: "The Grid is a software infrastructure that enables flexible, secure, coordinated resource sharing among dynamic collections of individuals, institutions and resources." [1] An essential part of our system is the Grid Security Infrastructure (GSI), which allows secure connections to potentially all computers in the Grid. GSI is based on public key encryption, X.509 certificates, and the Secure Sockets Layer (SSL) communication protocol [4]. The design uses HTTP/HTTPS protocol to access remote databases. Access to common http and https ports (80, 443) is normally allowed through possible firewalls. The user authentication is based on a common certificate, thus separate user IDs or passwords are not needed.

The best known GSI implementation is a part of Globus software. Globus (http://www.globus.org) is a de-facto standard for distributed computing platforms. Globus incorporates a large body of secure data transfer and remote program execution tools. Globus'es GSI is used as the backbone of TrustManager, too.

## 2.2 Spitfire and TrustManager

Spitfire is a project of Work Package 2 within the European Data Grid Project (EDG) [5]. It offers a Java servlet that accepts database request using HTTP/HTTPS protocols and displays the results in XML. TrustManager is EDG's authentication software that can be used in connection with Spitfire; it analyzes the users rights to execute queries based on his user (Grid) certificate.

In TrustManager the authentication is based on hand shaking protocol in Secure Sockets Layer (SSL) and Transport Layer Security (TLS). The server and client send each other their X.509 certificates and messages encrypted by their private keys that are related to the public keys included in the X.509 certificates. This way the server and the client authenticate themselves as owners of the certificate [3].

A certificate chain is established when a user receives a certificate from a certificate authority (CA). TrustManager gets called when the server receives a connection and when the client makes a connection. The certificate chain containing is checked by the TrustManager. The signature in each of the certificates has to be made by the previous certificate in the chain and the distingushed name (DN) of the issuer of a certificate must be the subject DN of the previous certificate. The chain has to end with the certificate that is one of the CA certificates in trust settings. None of the certificates can be expired. The TrustManager throws an exception if the chain is invalid [3].

## 2.3 VOMS

It is not practical to look after authentication or authorization information for everyone at every site. This is why EDG introduces two concepts called Virtual Organization (VO) and Resource Providers (RP). The VO is an abstract entity grouping users, institutions and Resources in a single administrative domain. The RP facility offers recourses to other parties [3].

Virtual Organization Membership Service (VOMS) creates a certificate, which proves the identity and that the entity (like a user) has a membership and certain other attributes of a VO. A trusted service (CA in authentication; VOMS in authorization) signs a set of information (users identity; users groups) with its private key. Its validity can be checked at the services using the public key of this trusted service. The information is included in the users proxy certificate, which is signed by the users normal certificate. Because the VOMS certificate is a proxy, the

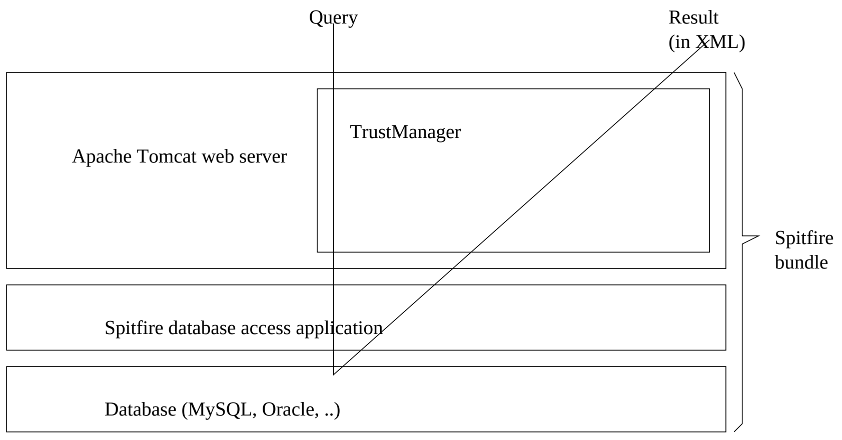

Figure 1: A block diagram of the elements in a Spitfire bundle.

lifetime of the authorization certificate can not be longer than the lifetime of the certificate which it is based on [3].

## 3 Implementation

In Figure 3, a general layout of a "bundle" consisting of TrustManager, Tomcat web server and Spitfire database application is described. There, a client issues a query to the web server part of the bundle. The client can be either a web browser or a standalone query program, but in both cases the client must "possess" a client certificate that it can provide to the server. TrustManager, embedded in Tomcat web server, inspects the client's certificate and if it is valid, authenticates the user to the service. Spitfire web application receives and executes the query and outputs the results in XML format. Spitfire can access relational databases by several manufacturers, including Oracle and MySQL.

Figure 3 illustrates the situation of distributed publication databases, where there are several "bundles", each in one collaborating organization. This enables us to use the virtual organization membership information for user authentication. There, with each query (or session), TrustManager first inspect the identity of the client as in Figure 3. Then TrustManager queries VOMS server in order to find out what are the roles of this user. If, for instance, the client is a program that retrieves indexes of publications, it may have a role that enables it to retrieve all publications. On the other hand, if the client belongs to a user that does not have an agreement with the publisher, it may only get a role that enables it to retrieve "free of charge" publications. The role is then mapped with the database, and (if the authorization was successful), the result is returned as in Figure 3.

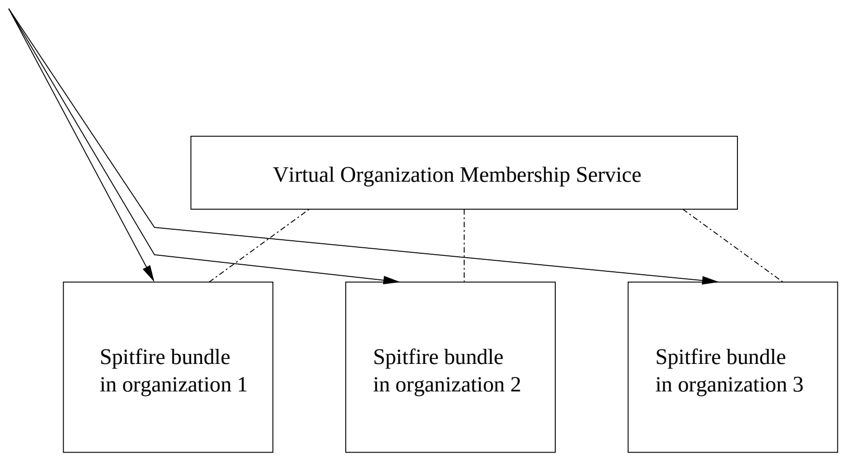
Virtual Organization Membership Service
Spitfire bundle in organization 1
Spitfire bundle in organization 2
Spitfire bundle in organization 3

Figure 2: Distributed databases.

## 4 Conclusions

The main benefits of the design are: (i) it separates authentication and role authorization in manageable modules; (ii) There is no need to write user id - password -pairs (that would be difficult in an application program anyway) because of certificates; (iii) It is generic, i.e. can access databases of different manufacturers; (iv) it outputs XML that can be easily processed further.

The software is open source, and available at http://ppewww.ph.gla.ac.uk/cgi-bin/cvsweb.cgi. For further details, see [2].

## References

[1] I. Foster, C. Kesselman, and S. Tuecke. The anatomy of the Grid: Enabling scalable virtual organizations. *International Journal of Supercomputer Applications*, 15(3), 2001.

[2] A. Frohner. Datagrid security design. Technical report, CERN, 2003.

[3] Security Coordination Group. Datagrid security design. Technical report, European DataGrid Project, 2003.

[4] The Globus project. Commodity Grid Kits. Available on: http://www-unix.globus.org/cog/, 2002.

[5] Project Spitfire. Project Spitfire. Available on: http://spitfire.web.cern.ch, 2001.

[6] The Globus Project. Overview of the grid security infrastructure. Available on: http://www.globus.org/security/overview.html, 2002.

[7] ITU X.509. Information technology - open systems interconnection - the directory: Public-key and attribute certificate frameworks. Technical report, ITU, 2000.