TRUST RELATIONSHIPS AND SINGLE SIGN-ON IN GRID BASED DATA WAREHOUSES

Xiaoyu Li[a] and Maree Pather[b]

[a] Department of Information Technology, Nelson Mandela Metropolitan University

[b] Department of Applied Informatics, Nelson Mandela Metropolitan University

[a] xiaoyu.li2@nmmu.ac.za, 0728524877

[b]maree.pather@nmmu.ac.za, +27 41 504-3580, PO Box 77000, Port Elizabeth, 6031

ABSTRACT

This project proposes a Grid system for a pre-defined virtual organisation, to facilitate the operations of a non-governmental organisation (NGO) with a specific set of initiatives, operating within a specific demographic boundary. The framework for this project is essentially an OGSA-based Grid Services architecture, comprising distributed data warehouse sources. Authentication and access control in such a system requires designers to establish appropriate trust relationships across trust-boundaries. This paper focuses on this particular security aspect, exploring authentication, single sign-on, delegation, and authorization mechanisms, as supported by the Globus Toolkit's Grid Security Infrastructure, to establish trust relationships and manage credentials.

KEY WORDS

X.509 Proxy Certificate, MyProxy, GSI, authentication, CAS, Single sign-on, delegation

TRUST RELATIONSHIPS AND SINGLE SIGN-ON IN GRID BASED DATA WAREHOUSES

1 INTRODUCTION

This paper is based on wider research into OGSA-based Grid services architecture, comprising a Decision-Support system (which utilizes a Data Warehouse, Data Marts, and near-line Operational Data Sources that are hosted by distributed organizations). Within this framework, specific patterns for collaboration, interoperability and security are included. This paper focuses on trust relationships as espoused by a single sign-on (SSO) design.

A Grid system integrates a distributed and heterogeneous collection of locally managed users and resources. The Globus Toolkit Grid Security Infrastructure (GSI; as per www.globus.org) consists of a set of components for addressing different security issues (such as authentication, delegation and authorization) within a Grid system. The main advantage of GSI is that the general security issues are solved at infrastructure level rather than application level; the applications need only deal with application-specific policy. The GSI, version 4, (GSI4) contains both Web Services and pre-Web Services components; both use the same authentication mechanism. The envisaged security solution will discuss the use of TLS with X.509 public key certificate for authentication; X.509 proxy certificate for single sign-on and delegation; the MyProxy protocol (pre-Web Services GT4 component) as an online credentials repository, and Community Authorization Services (CAS) with Security Assertion Markup Language) SAML for authorization.

The remainder of this paper is organized as follows. In section 2, Grid security requirements will be discussed. In section 3, a high-level Grid security architecture is introduced in brief. This architecture illustrates a set of abstract concepts and mechanisms for addressing security issues within a general Grid environment (without mentioning any specific security technologies). In section 4, the mechanisms of authentication, delegation, single sign-on, and authorization, implemented by GSI, are introduced. In section 5, an integrated perspective is posited.

2 SECURITY CHALLENGES IN THE GRID ENVIRONMENT

A Grid system integrates distributed heterogeneous resources hosted by multiple organizations. All involved organizations are considered as members of a VO, in order to facilitate such resource sharing and collaborated activities across organizations. The security challenges within a pre-defined VO are discussed in this section. First, some Grid-based security terminology needs to be clarified (Foster, Kessleman, Tsudik & Tuecke, 1998):

a) In Grid systems, a subject is generally a user, a process operating on behalf of a user, a resource, or process acting on behalf of a resource.

b) An object is a resource that is being protected by the security policy.

c) A trust domain is a collection of both subjects and objects governed by single administration and a single security policy.

d) A credential is a piece of information that is used to prove the identity of a subject (such as passwords and certificates).

Subjects, generally, provide data from their own Data Warehouse, Data Marts, and near-line Operational Data Sources as Grid resources. Typically, a Web portal would provide a brokered interface to Grid applications. A Grid application may be responsible for invoking specific data access services for completing user-group-specific tasks. Figure 1 shows the architecture of a typical Grid, and an example of a single typical data request retrieving data from distributed data sources across multiple organizations.

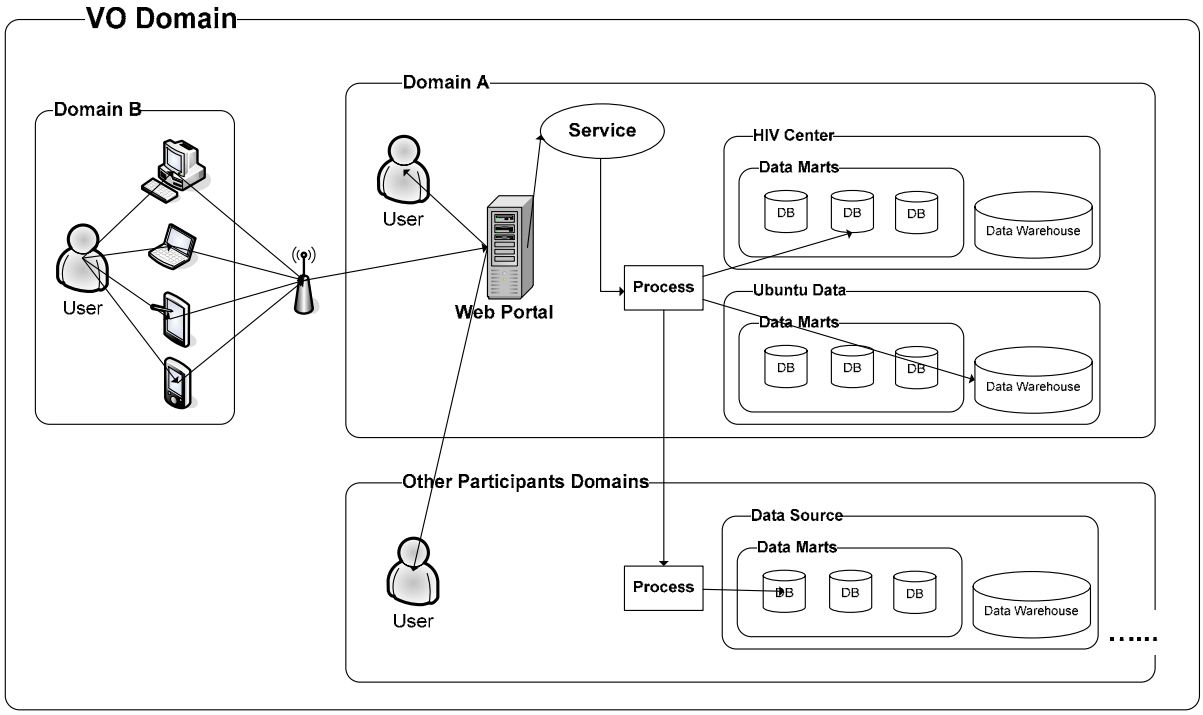

Figure 1. VO Architecture

The typical security requirements can be summarized as follows.

a) This architecture integrates distributed users with data sources, which are managed locally by their owners. Each organization can be thought as a trust domain. Operations that are confined to a single trust domain are subject to local security policy. Operations across multiple trust domains require multiple authentications. For example, a single data request may access different data sources hosted in different organizations/trust domains.

b) As both global and local subjects exist, a remote user can have a global user name, used to access the services portal, and also a local user name, defined by a local trust domain. For persisting credentials, a global subject (identity and role) can be mapped to a local subject (identity and role). Local security policy will dictate the permission sets of virtual local (mapped) and true local subjects.

c) User credentials (such as passwords, private keys, etc.) must be protected during interaction either across different trust domains or within a single trust domain. A secure transmission protocol is required to ensure privacy and integrity when these credentials are transported through the network.

d) One user's request may involve many processes on many distributed resources. It is necessary that a user only sign-on once for a long-lived programme or process. A program or process should be allowed to act on behalf of a user and be delegated a subset of the user's rights. Processes running on behalf of the same subject within the same trust domain may share a single set of credentials.

e) VO resources are located within multiple organizations. Each organization retains ultimate control over the (local) policies that control access to its resources.

The inter-domain (global) security policy must be able to interoperate with, rather than replace, the diverse intra-domain (local) policies.

3 A GRID SECURITY ARCHITECTURE

This section introduces a high-level Grid security architecture, as depicted in GIS4. Figure 2 (Foster, et al., 1998) shows an overview of this security architecture. Two types of proxy are defined: a user proxy and a resource proxy. Four related protocols are defined: user-proxy-creation protocol, resource-allocation protocol, resource-allocation-from-a-process protocol and mapping-registration protocol (Foster, et al., 1998).

A user proxy is a session manager process given permission to act on behalf of a user for a limited period of time. Once the user proxy has been created, the user may be disconnected in order to eliminate the need to have the user's credentials available for every security operation. It reduces the possibility of the credentials being compromised during operations. Further, the lifetime of the user proxy credentials is under control of the user. A resource proxy is an agent used to translate between inter-domain security operations and intra-domain (local) mechanisms. It is allocated by user proxy, and is responsible for scheduling the access to a resource and for mapping a computation onto that resource.

When a user logs on to the Grid system, it creates a user proxy by using the user-proxy-creation protocol. The user proxy then allocates a resource and creates processes by using the resource-allocation-protocol. A process may allocate additional resources by using resource-allocation-from-a-process protocol. The mapping registration protocol can be used to define a mapping from a global subject to a local subject. The following should be noted in this regard:

a) User proxy credentials should be signed by the user's long-lived credentials and contain all information (user-id, local host name, etc.) required for authentication. The integrity of user proxy credentials is protected by local security policy.

b) A user proxy requiring access to a resource first determines the identity of the resource proxy for that resource. It then issues a request to the appropriate resource proxy. If the request is successful, the resource is allocated and a process created on that resource. The request can fail because the resource is not available, or because of authentication failure or authorization failure.

c) The resource-allocation protocol is used to issue a request to a resource proxy from a user proxy. The user proxy and resource proxy authenticate each other. The resource proxy checks if the user who signed the proxy's credentials is authorized by local policy to make the allocation request. At this time, the verification may require accessing a mapping table maintained by the resource proxy (for mapping the user's credentials onto a local user-id). A single resource allocation request may result in the creation of multiple processes on the remote resources. All such processes are created with the same credentials.

d) It is a common case that the resource allocation is initiated dynamically from a process created by a previous resource-allocation request. The user proxy decides whether to honour the request through authentication between user proxy's credentials and process credentials. The resulting process handle is signed by the user proxy and returned to the requesting process.

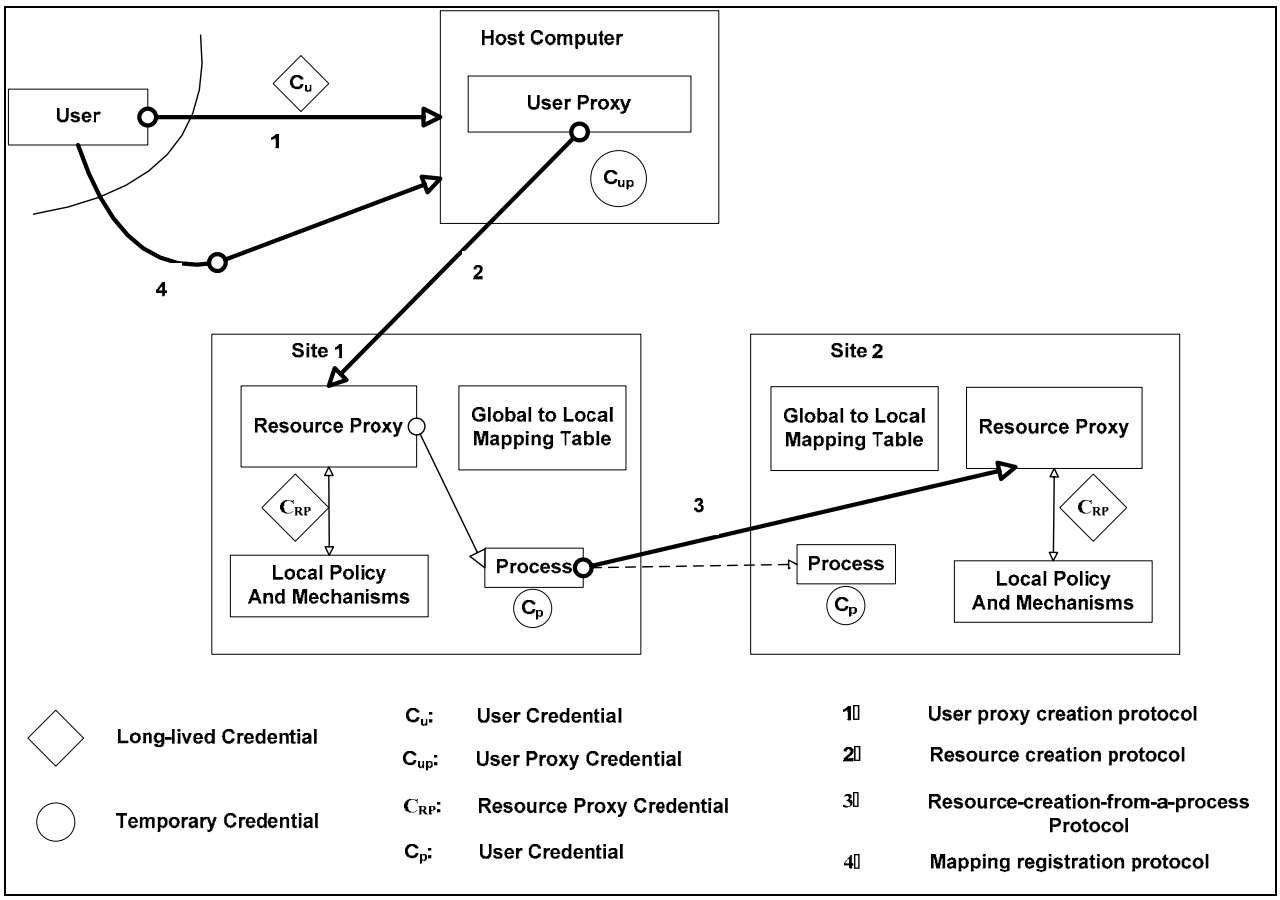

Figure 2. Grid Security Architecture

This approach uses a user proxy to interact with the resource proxy to achieve single sign-on (SSO) and delegation. Authentication occurs between a user proxy and a resource proxy. Consequently, the SSO leverages the existing trust relationship between a user and a resource that was established when the user was initially granted access to the resource. The user proxy and process authenticate each other when a resource allocation request is issued by a process. The resource-allocation request is successful only when the user is authorized by the resource on the basis of local policy.

4 GRID SECURITY INFRASTRUCTURE

The architecture discussed in the previous section defined protocols in abstract terms, rather than in terms of specific security technologies. Hence, these protocols can be implemented by using any modern security technologies and mechanisms. Grid Security Infrastructure (GSI), as it appears in the Globus Toolkit, is essentially an implementation of the architecture discussed in section 3. It primarily focuses on authentication and message protection; defining SSO algorithms and protocols; cross-trust domain authentication protocols; and delegation mechanisms for users, and for processes executing on a user's behalf. GSI defines a set of protocols, libraries, and tools that allow users and applications to securely access resources. It acts as Grid security middleware to provide infrastructure level security functionalities for addressing security issues in a Grid environment. In the following sub-section, the authentication, delegation and authorization mechanisms used by GSI will be introduced.

4.1 Using Public Key Infrastructure for Authentication

Authentication between two entities (users and resources) in a Grid means that each party establishes a level of trust in the identity of the other party. An authentication protocol sets up a secure communication channel between the authenticated parties, so that subsequent messages can

be sent without repeated authentication steps, although it is possible to authenticate every message. The identity of an entity is typically some token or name that uniquely identifies then entity.

The GSI authentication mechanism is based on Public Key Infrastructure (PKI). In practise, GSI uses a X.509 public key certificate as identity token. An X.509 certificate contains a public key, a subject name in the form of a multi-component distinguished name (DN), and a validity period and is signed by a trusted third party, or certification authority (CA) (Housely, Polk, Ford & Solo, 2002). The associated private key is owned by the correct remote subject with whom an encryption or digital signature mechanism will be used. Other standards documents refer to these certificates as "public key certificates" or X.509 certificates. The Global GridForum (GGF-www.ggf.org) document (Thompson et al., 2003) also uses the term "identity certificate". A Grid CA is defined as a CA that is independent of any single organization and is responsible for signing certificates for individuals who are allowed to access the Grid resources, hosts or services running on a single host. It is different from a traditional organizational CA. Organizational CA only issues certificates for members of its organization, and these certificate are used to access resources within this organization. In identity certificates issued by an organizational CA, the DN often contains a number of attributes (e.g., organizational unit, location, and email) retrieved from the organization's directory (such as X.500 and LDAP directory). Since a Grid CA is independent of the organizations to which its subscribers belong, it does not have a way to verify much information about a subscriber or to know when such information changes. The prudent approach for a Grid CA is to put as little information in the certificate as possible.

In brief, a GSI user generates a public and private key pair and obtains an X.509 certificate from a trusted CA. The X.509 certificates are used with the TLS (Transport Level Security) protocol to ensure a secure authenticated connection between two parties. The TLS protocol provides communication privacy and data integrity between two communicating applications over the Internet (Dierks & Allen, 1999). X.509 certificates are exchanged between entities (users and resources). The certificates are first tested by checking the expiration dates, possible revocation, acceptable key usage, and signature, by a trusted CA. If the certificates pass all these checks, their public keys are then used to build a challenge handshake to prove that each entity that sent a certificate has the corresponding private key. Passing these tests gives each party a level of confidence that it has established a secure connection to the party represented by the certificate presented. The veracity of an entity's identity is only as good as the trust placed in the CA that issued the certificate, so the local administrator installs these certificates, which are then used to verify the certificate chains. The current version of GSI uses X.509 Proxy Certificate (Tuecke, Welch, Engert, Pearlman & Thompson) as temporary credentials called proxy credentials for authentication. GSI treats X.509 Certificate and X.509 Proxy Certificate equivalently. Proxy credentials also enable GSI to support SSO and delegation, which are discussed in the next section (Welch et al., 2004).

4.2 Proxy Credentials for Single sign-on and Delegation

GSI provides single sign-on and dynamic delegation by using proxy credentials. Proxy credentials are short-term credentials created by a user, and in fact is a short-term binding of user's DN to an alternate private key. It allows users to be authenticated once and delegate proxy credentials to processes on remote hosts. Proxy certificates use the format prescribed for X.509 public key certificates. Unlike a public key certificate mentioned in section 4.1, the issuer of a proxy certificate is identified by a pubic key certificate or another proxy certificate rather than a CA certificate. As mentioned, GIS treats X.509 Certificate and X.509 Proxy Certificate equivalently. This approach allows proxy certificates to be created dynamically without requiring the normally heavy-weight vetting process associated with obtaining public key certificates from a CA.

Single sign-on in Grid environment allows the user to manually authenticate once in order to create a Proxy Certificate which can be used repeatedly to authenticate for some period of time without compromising the protection on the users' long term private key. This is started by creating

a new key pair, consisting of a public key and private key. The public key is then encoded in a certificate request. The Proxy Certificate is created by using user's private key associated with its long-term public key certificate to sign the certificate request containing the public key of the new key pair. The newly generated Proxy Certificate binds the new public key to a new name and delegates some or all the user's privileges to the new name. It can then be used with the new private key by the bearer to authenticate to other parties. The user's long-term private key will not be involved in authentication until the Proxy Certificate expires. Proxy Certificate can also be used for delegating privileges from an issuer to another party in a light-weight manner over a network connection without the exchange of private key. The Proxy Certificate has a short life-time. It and its private key are stored in a local file and protected in a less secure manner than the long-term private key. In this paper, the MyProxy protocol is used to store proxy credentials rather than using a local file. This will be discussed in section 4.3.

4.3 MyProxy Protocol

MyProxy is an online credentials repository for Grid systems (Novotny, Tuecke & Welch, 2001). In a typical scenario, a Grid Web portal, combining a Web server and Grid-enabled software, provides users an interface to access all Grid applications using standard Web browsers. Most Grid portals require that the user delegates to the server the right for that server to act on the user's behalf, in order to initiate and monitor operations for that user on Grid resources. GSI supports such delegation, but the standard Web security protocols do not. MyProxy bridges this incompatibility between Web and Grid security protocols, thus enabling Grid portals to use GSI-protected resources in a secure and scalable manner. It allows long-lived keys to be secured on the remote server, while allowing convenient access to short-lived proxy credentials as needed.

The MyProxy credentials repository system consists of a repository server and a set of client tools for delegating to and retrieving credentials from the repository. In order to meet the goals of the MyProxy approach, there are two basic steps for using the repository: delegation of proxy credentials to the repository and retrieving the credentials from the repository (Novotny, Tuecke & Welch, 2001).

a) A user starts by using the myproxy-init client program along with its permanent credentials to contact the repository and delegate a set of proxy credentials to the server along with authentication information and retrieval restrictions. Authentication information in this process consists of a user identity and a pass phrase. It is used to authenticate any retrieval operations. This user identity is different from the user's DN; it is actually hand-typed by the user at later times. The user identity identifies the account storing the proxy credentials in repository server. Both can be tested by the repository to ensure they comply with any local policy (for example, the pass phrase must meet a certain length). The only available retrieval restriction that can be placed on delegations by the repository is the maximum lifetime of the proxy credentials. These restrictions are intended to be expanded in future versions.

b) A user, or service acting on behalf of the user, uses the myproxy-get-delegation client program to contact the repository server and request a delegation of the user's credentials. During this process, the user must provide the identity and pass phrase for verification. After verifying this authentication information and checking and restrictions that the user presented with the delegation, the repository will delegate proxy credentials back to the user or service.

c) The credentials delegated to the repository normally have a lifetime of a week. This life time can be changed to any length of time desired. The credentials delegated to repository can be destroyed at any time by using the myproxy-destroy client program.

The first step to using MyProxy in a Grid portal is to delegate proxy credentials to the repository by using myproxy-init client program. Then the user may connect to the Grid portal by

using a Web browser and provide the authentication information (user identity and pass phrase) through a Web form at a different time and place. The Grid portal then uses myproxy-get-delegation program to connect to the MyProxy repository and authenticates itself using its own Grid credentials. The user's authentication information (user ID and pass phrase) is also transferred to repository server at the same time for requesting a proxy credential for the user. The repository would delegate a proxy credential for the user back to the portal after all necessary verification (portal's Grid credentials, user's authentication information). The Grid portal then can securely access the Grid resources by using standard Grid applications. The operation of logging out of the Grid portal deletes the user's delegated credentials on the portal; otherwise, the credentials will expire when the lifetime lapses.

4.4 Community Authorization Services

The Community Authorization Services (CAS) solution in the Grid environment addresses the management and policy issues introduced by a VO. A VO is a dynamic collection of resources and users unified by a common goal and potentially spanning multiple administrative domains. In a Grid environment, resource providers and consumers are controlled by dynamic and complex policies. These policies determine who can use which resources and for which purposes. CAS describes an approach (Pearlman, Welch, Foster, Kesselman & Tuecke, 2002) to represent, maintain and enforce such policies that provide a scalable mechanism for specifying and enforcing these policies. It allows for a separation of concerns between site policies (local policies) and VO policies. Site policies are about what the VO is allowed to do and the VO policies are about what the individual user is allowed to do as a VO member. The CAS architecture builds on PKI-based authentication and delegation mechanism which is also supported by GSI. As discussed above, GSI uses X.509 Proxy Certificates to provide credentials for the user and to allow for delegation and SSO.

CAS functions as "push-model" authorization services (Vollbrecht et al., 2000) (Lorch et al., 2003). The CAS concept can be simply divided into two parts: policy management and enforcement. Policy management concerns the maintenance of VO policies and communication between VO policies and sites' policies. The sites then combine their local policies with the VO policies and enforce this combined policy. The VO's portion of this combined policies are maintained by the VO through the CAS server, and the resource provider's policies regarding the VO are maintained by the resources provider using the same mechanisms used for non-VO users. Resource providers deploy CAS-enabled services (i.e., services modified to enforce the policy in the CAS credentials) onto resources assigned to a VO for policy enforcement. A user first requests CAS credentials for accessing those resources from the VO's CAS server. CAS credentials contain a policy statement of that user's rights, cryptographically signed by the CAS server. The user then makes a request to the resource with these CAS credentials. A CAS-enabled service deployed by the resource provider verifies the validity of the CAS credentials (e.g., signature). After this verification, both site policies and VO policies are enforced by taking the following steps: 1). Enforce the site's policies regarding the VO. 2). Enforce the VO's policies regarding the user. 3). optionally, enforce any additional site policies in regard to the user.

GSI has developed two CAS prototypes (Pearlman, Kesselman, Welch, Foster & Tuecke, 2003) according to the mechanism introduced above. The first one uses restricted proxy certificate issued by the CAS server to the user. The second one (CAS alphaR2) use a combination of a proxy certificate issued by the user with a signed policy assertions issued by the CAS server.

4.5 Integration of Grid Security Infrastructure with the OGSA security model

GSI is the portion of the Globus Toolkit that provides the fundamental security services according to the mechanisms discussed above. The version 4 of GSI (GSI4), corresponding to the Globus Toolkit version 4 (GT4), integrates with the OGSA security mechanism to allow applications and

users to operate in the Grid in a seamless and automated manner (GSI4 is not the first implementation of the OGSA mechanism).

OGSA is a set of technical specifications that aligns Grid technologies with emerging Web Services technologies. It defines standard Web service interfaces and behaviours that add to Web services the concepts of stateful services and secure invocation, as well as other capabilities needed to address Grid-specific requirements (that are not relevant to this paper). These interfaces and behaviours define what is called a "Grid service" and allow users to manage the Grid service's life-cycle, as allowed by policy, and to create sophisticated distributed services. The OGSA security model casts all security functionalities as Grid Services to allow them to be located and used by applications. The security services defined in OGSA includes authentication, identity mapping, authorization, credentials conversion, audit, secure logging and privacy (Foster et al., 2005). The OGSA security model uses a sophisticated container-based hosting environment (such as J2ME, .Net) to handle security for applications and to allow security to adapt without having to change the application. In order to establish trust relationships, two entities need to be able to find a common set of security mechanisms understood by both of them. The use of hosting environments and security services enables OGSA applications and services to adapt dynamically and use different security mechanism, so using publishing security policy to allow clients to discover dynamically what credentials and mechanism are needed to establish trust with the services. The standard for exchange of security token is also need to be specified for interoperability. These features allow the security mechanism to be supplied by the surrounding Grid infrastructure instead of being instantiated in an application, which enables applications only need to focus on application-specific policy.

5 A GENERALISED SECURITY MODEL

A typical Grid Services system requires security functions, including authentication, authorization, single sign-on and delegation. This is provided by GSI4. GSI4 contains both Web Services components and pre-Web Services components; MyProxy is a pre-Web Services component. GSI4 Web Services components can be divided into four distinct functions: message protection, authentication, delegation and authorization (Globus, 2005).

a) In OGSA-based Grid services systems, most capabilities and functionalities are represented as Web Services. The Web Services-based portion uses SOAP (W3C, 2000) for communication. Message protection can be provided either by transporting SOAP messages over TLS, or by signing and/or encrypting portions of the SOAP message using the WS-Security standard (Message-level Security) (OASIS, 2004). TLS provides for both integrity protection and privacy (via encryption). The WS-Security standard and the WS-SecureConversation specification provide SOAP message protection. The WS-SecureConversation specification (IBM et al., 2002) allows for an initial exchange of messages to establish a security context which can then be used to protect the subsequent SOAP messages.

b) GSI4 supports authentication and delegation through the use of X.509 credentials. Authentication with X.509 credentials can be accomplished either via TLS, or via digital signature as specified by WS-Security. X.509 end entity certificates (EECs) (Housely et al., 2002) are used to identify persistent entities such as users and services. It normally acts as long-term credentials. The X.509 EEC provides each entity with a unique identifier (i.e., DN) and a method to assert the identifier to another party through using a private and public key pair bound to the identifier by the certificate. The standard X.509 Proxy Certificates are used to support delegation and single sign-on. It normally acts as short-term credentials which allow bearers of X.509 EECs to delegate their privileges temporarily to another entity. An X.509 Proxy Certificate has its own public and private key pair. It can be signed by X.509 EECs or another X.509 Proxy Certificate, but it can not sign an X.509 EEC. Using X.509 Proxy Certificate avoids transferring user's X.509

EEC's private key frequently for authentication or authorization. GSI treats X.509 EECs and X.509 Proxy Certificate equivalently. GSI4 also supports username/password authentication.

c) GSI4 uses the security assertion markup language (SAML) (OASIS, 2002) as the format for the policy assertions issued by the CAS server.

The MyProxy protocol is responsible for providing an online credentials repository, and for delegating X.509 proxy certificates to the Grid Web portal. It allows the Grid Web portal to act on the user's behalf without involving the user's long-term credentials; it allows users to access their credentials from anywhere using standard Web browsers. This process is completed by the following steps (assuming an X.509 certificate issued by a CA has already been installed on a client):

a) Delegate a X.509 Proxy Certificate to the MyProxy credentials repository server. The client that stores the user's X.509 EEC (long term user's credentials) sends a 'Put' request (Basney, 2005) to the repository server along with a user-id, pass phrase and lifetime through the use of myproxy-init client program. The repository server accepts the request and generates a new public/private key pair and then sends a certificate request, containing the repository's public key, to the client. The client then sends a X.509 Proxy Certificate containing the public key from the certificate request, signed by its X.509 EEC's private key, followed by the corresponding certificate chain, back to repository server. The repository then stores this X.509 Proxy Certificate for later retrieval from Grid Web portal. The user ID and pass phrase specify the account for storing the proxy certificate and used to authenticate its retrieval operation. The stored X.509 expires at the lifetime. This process is authenticated via TLS.

b) Users can log on the Grid Web portal at a different time and place by using a standard Web browser. The Grid Web portal sends a 'Get' request (Basney, 2005), along with the user-id and pass phrase generated in the first step, to the MyProxy repository server to retrieve the stored user's X.509 Proxy Certificate through the use of the myproxy-get-delegation client program. If the repository's response indicates success, the Web portal generates a new public/private key pair and then sends a certificate request, containing the Web portal's public key, to the repository server. The repository server then sends a X.509 Proxy Certificate, containing the public key from the certificate request, signed by the private key of the stored user's X.509 Proxy Certificate, followed by the corresponding certificate chain, back to the Web portal. This process is authenticated via TLS and also User ID and pass phrase. This process enables the repository to delegate proxy credentials for the user back to the Web portal, and then the Web portal can act on the user's behalf to securely access the Grid resources by using standard Grid applications.

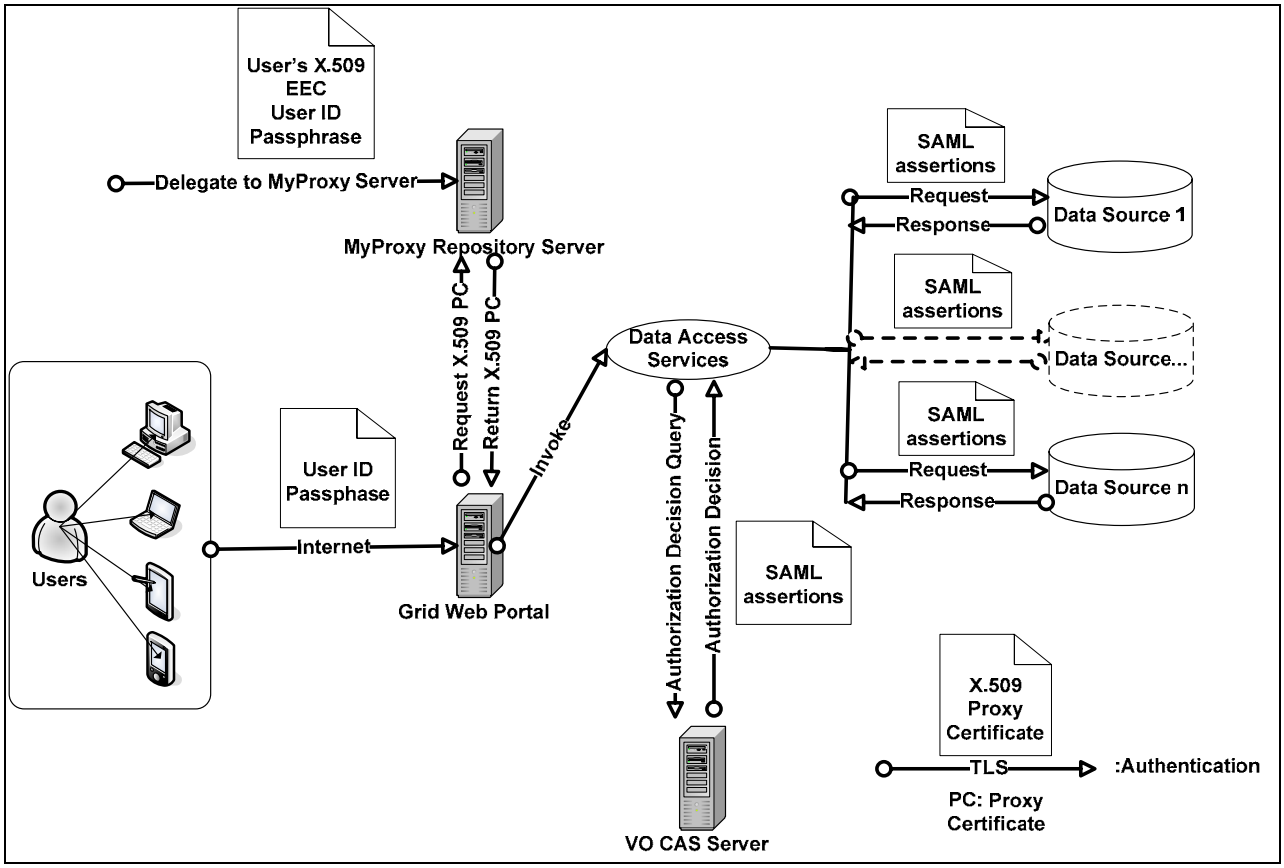

Figure 3. Generalised Security Architecture

These two processes enable the Grid Web portal to act on the user's behalf to access Grid resources through using a X.509 Proxy Certificate rather than the user's X.509 EEC. It provides an effective way to protect the user's private key. Then the Grid Web portal can act as the user to invoke the deployed Grid services to perform user's tasks. The Web Services components of GSI4 are used to provide authorization, delegation and authentication services for the subsequent operations. After the user logs on the Grid Web portal, the user can launch an application in order to perform a data analysis task to retrieve some data from the data sources (e.g. a data mart or data warehouse) hosted by two organizations. The application conducts this task by invoking a data access service (OGSA Grid Service). This service queries data from two data source respectively. Before the two data source respond to the query requests, the user's identity, rights, etc. need to be verified both by the VO and the data source owner according to VO policies and organizations local policies. This is conducted by a VO CAS server and the organizations' own authorization mechanisms. SAML is used as the format for the policy assertions issued by the CAS server. The communications among Web portal, CAS server, services and data sources use SOAP messages protected either by transporting SOAP messages over TLS, or by signing and/or encrypting portions of the SOAP message with X.509 certificate using the WS-Security standard. The following steps describe how the user is authorized by VO:

a) A Grid Web portal acts as the user by using the user's X.509 Proxy Certificate. The Web portal use GSI4 delegation services to delegate another X.509 Proxy Certificate to data access service for authentication and authorization before it is allowed to access data sources. The data services send a signed SAML AuthorizationDecisionQuery request to the VO CAS server to indicate which data source it desires to access and what actions (i.e., read, write, etc.) it desires to take.

b) The VO CAS server establishes the user's identity (user's DN from X.509 Proxy Certificate) which is used to determine the rights as established by the VO policy. It then returns a signed SMAL assertion containing AuthorizationDecisionStatement, the user's identity and some subset of the user's requested actions.

c) The data access service presents the SAML assertion to a data source along with the authenticated data request. The data source uses the SAML assertion, subject to local policy regarding how much authority was delegated to the CAS service, to authorize the data query request, and returns the data according to the user's rights and requested actions, described by SAML assertion. The SAML assertions can also be used to make multiple requests to multiple data resources.

The security model discussed above consists of: 1). Using MyProxy protocol as an online credentials repository to store and retrieve user's X.509 Proxy Certificate. 2). A Grid Web portal using an X.509 Proxy Certificate retrieved from a MyProxy repository to perform operations on the user's behalf. 3). The CAS, using SAML for policy assertions, to indicate user's rights and requested actions on Grid resources. 4). The TLS and X.509 Certificates to provide authentication between two entities both on transport-level and message level. All components are combined to enable the Grid system work in a secure way.

6 CONCLUSION

The typical modern Grid system has two main features: 1). It is an OGSA-based system which uses standard OGSA services to provide some basic capabilities (i.e., the capability to access distributed data sources). 2). It use a Grid services Web portal to provide a Web-based interface allowing users to access Grid resources by using standard Web browsers at any time and place. This paper discusses a security model based on the mechanisms and implementations of the Globus Toolkit 4 Grid Security Infrastructure. This model provides infrastructure-level security, through using GSI middleware, to provide, inter alia, authentication, authorization, delegation and SSO. It integrates the selected components from GSI to establish trust relationship within the proposed OGSA-based Grid system, comprising a Decision-Support system. The model ensures participants in the pre-defined VO to have secure access to distributed data and information services through a Grid services Web portal.

7 REFERENCES

Basney, J. (2005). MyProxy Protocol. Global Grid Forum. GFD-E.054, November 2005.

Dierks, T., Allen, C. (1999). The TLS Protocol Version 1.0 <RFC2246>. IETF RFC 2246, Retrieved February 15, 2006 from http://www.ietf.org/rfc/rfc2246.txt.

Foster, I., Kesselman, C., Tsudik, G. and Tuecke, S. (1998). A Security Architecture for Computational Grids. 5th ACM Conference on Computer and Communications Security, 83-91. 1998.

Foster, I., Kesselman, C. and Tuecke, S. (2001). The Anatomy of the Grid: Enabling Scalable Virtual Organizations. International Journal of Supercomputer Applications, 15 (3), 200-222. 2001.

Foster, I., Kishimoto, H., Savva, A., Berry, D., Djaoui, A., Grimshaw, A., Horn, B., Maciel, F., Siebenlist, F., Subramaniam, R., Treadwell, J., and J. Von Reich, J. (2005). The Open Grid Services Architecture, Version 1.0. Global Grid Forum OGSA-WG. GFD-I.030, January 2005.

Globus (2005). GT4.0: Security. http://www.globus.org/toolkit/docs/4.0/security/.

Housley, R., Polk, W., Ford, W. and Solo, D. (2002). Internet X.509 Public Key Infrastructure Certificate and CRL Profile <RFC3280>. IETF. Retrieved February 18, 2006 from http://www.ietf.org/rfc/rfc3280.txt.

IBM et al. (2002). Web Services Secure Conversation Language (WS-SecureConversation) Version 1.0. December 18, 2002.

Lorch, M. et al. (2003). Conceptual Grid Authorization Framework and Classification. Global Grid Forum Authorization Frameworks and Mechanisms–WG. GFD-I.038. February 2003 (Revised November 2004).

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

OASIS (2002). Security Assertion Markup Language (SAML) V1.0. November 5, 2002.

OASIS (2004). Web Services Security: SOAP Message Security. 15 March 2004.

Pearlman, L., Welch, V., Foster, I., Kesselman, C. and Tuecke, S. (2002). A Community Authorization Service for Group Collaboration. IEEE 3rd International Workshop on Policies for Distributed Systems and Networks. 2002.

Pearlman, L., Welch, V., Foster, I., Kesselman, C. and Tuecke, S. (2003). The Community Authorization Service: Status and Futures. Computing in High Energy Physics (CHEP03). 2003.

R. Thompson, M. et al. (2003). CA-based Trust Issues for Grid Authentication and Identity Delegation. Global Grid Forum Grid Certificate Policy WG. GFD-I.17, June 2003.

Tuecke, S., Welch, V., Engert, D. and Thompson, M. (2004). Internet X.509 Public Key nfrastructure (PKI) Proxy Certificate Profile <RFC3820>. IETF. Retrieved February 18, 2006 from http://www.rfc-archive.org/getrfc.php?rfc=3820.

Vollbrecht, J. et al. (2000). AAA Authorization Framework <RFC2904>. IETF. Retrieved March 25, 2006 from http://www.ietf.org/rfc/rfc2904.txt?number=2904.

Welch, V., Foster, I., Kesselman, C., Mulmo, O., Pearlman, L., Tuecke, S., Gawor, J., Meder, S. and Siebenlist, F. (2004). X.509 Proxy Certificates for Dynamic Delegation. 3rd Annual PKI R&D Workshop. 2004.

W3C (2000). SOAP (Simple Object Access Protocol) 1.1 Specification. Retrieved July 5, 2005 from http://www.w3.org/TR/2000/NOTE-SOAP-20000508.