Journal of Grid Computing (2004) 2: 301–311

Authentication and Authorization Mechanisms for Multi-Domain Grid Environments

Linda A. Cornwall[1], Jens Jensen[1], David P. Kelsey[1], Ákos Frohner[2], Daniel Kouřil[3], Franck Bonnassieux[4], Sophie Nicoud[4], Károly Lőrentey[5], Joni Hahkala[6], Mika Silander[6], Roberto Cecchini[7], Vincenzo Ciaschini[7], Luca dell'Agnello[7], Fabio Spataro[7], David O'Callaghan[8], Olle Mulmo[9], Gian Luca Volpato[9], David Groep[10], Martijn Steenbakkers[10] and Andrew McNab[11]
[1]*CCLRC, Rutherford Appleton Laboratory, United Kingdom*
[2]*CERN, European Organization for Nuclear Research, Switzerland*
[3]*CESNET, Czech National Research Network Operator, Czech Republic*
[4]*CNRS, French National Center for Scientific Research, France*
[5]*ELTE, Eötvös Loránd University, Hungary*
[6]*HIP, Helsinki Institute of Physics, Finland*
[7]*INFN, Istituto Nazionale di Fisica Nucleare, Italy*
[8]*Department of Computer Science, Trinity College, Dublin, Ireland*
[9]*KTH, Royal Institute of Technology, Sweden*
[10]*NIKHEF, National Institute for Nuclear and High Energy Physics, Netherlands*
[11]*Schuster Laboratory, University of Manchester, United Kingdom*

Key words: authentication, authorization, grid, security

Abstract

This article discusses the authentication and the authorization aspects of security in grid environments spanning multiple administrative domains. Achievements in these areas are presented using the EU DataGrid project as an example implementation. It also gives an outlook on future directions of development.

1. Introduction

The implementation and operation of secure services running on a large-scale Grid leads to many challenges. The highly distributed nature of the resources used by even just one Virtual Organization (VO), spanning many different management and security domains, raises both technical and policy issues. This paper presents achievements in solving some of these problems, supported by colleagues in the *European DataGrid* (EDG) project [1]. We refer to a comprehensive security architecture, but in this paper focus on components in the area of Grid authentication and authorization. The concepts that are presented here have demonstrated robustness in thorough testing by the EDG project.

In the initial conception it was assumed that each middleware function could take care of its own security needs independently. Subsequently the cross-cutting aspects of security were better appreciated. In order to understand this we collected and documented the security requirements of the EDG project, both for middleware and the applications. Full details are presented in [2] by the DataGrid *Security Coordination Group* [3]. The 112 requirements identified are distributed across the many different areas of security, including Authentication, Authorization, Confidentiality, Integrity, and Non-repudiation. Building on this work, the security experts of various middleware development groups discussed compatibility issues and created a comprehensive security architecture [4] to meet the security requirements. It is *this* architecture we refer to.

The development and deployment of new security components were also driven by the growth of several European Grid testbeds (EDG, DataTAG, GridPP, LCG, etc.) spanning multiple organizations and utilizing services which were compatible with the grid at different levels. An evolutionary approach led to the development of several components which provide similar functionality to existing, non-grid solutions, thus easing the integration of existing production clusters with the grid infrastructure.

The security architecture for authentication and authorization allows:

– The User to be authenticated by a service.
– The service to gather additional information associated with the user or the actual session (e.g. group membership, role, period of validity).
– The service to gather additional information associated with the protected service or object (e.g. file permissions).
– The checking of local policy applicable to the situation (e.g. a temporarily disabled user).
– The making of an authorization decision based on the identity of the user and the additional information.
– The Users to access resources in a global Grid environment without the need for individual accounts at various sites, while allowing resource providers to keep control over access to their resources.

A primary principle has been to keep authentication and authorization separate while recognizing that authentication often includes some implicit limited authorization, such as the right to belong to a scientific community.

The authentication and delegation framework is based on the *Globus Grid Security Infrastructure* [5], which is itself an extension of the *Public Key Infrastructure* [6, 7].

Due to concerns about a single point of failure or attack, a single Certification Authority (CA) is not sufficient. For example, in EDG it was decided that an appropriate scale was one CA for each participating country. Hierarchical or cross-signed arrangements of multiple CAs are not compatible with Globus GSI, so a coordinated group of peer CAs appears to be the most suitable choice.

The authorization framework consists of both global authorization management components, i.e. those implementing VO policies, and local authorization enforcement components, i.e. those implementing site or resource-owner policies.

The remainder of this paper is organized as follows. Section 2 gives an overview of authentication, Section 3 describes the global authorization components, Section 4 uses the example of running jobs on an EDG Computing Element (CE) to present the local authorization components. Sections 5 and 6, respectively, are devoted to authorization components for Java-based web services and for use with Apache web servers. Future directions are described in Section 7.

2. Authentication

The user obtains a certificate proving his identity signed by the *Certification Authority* (CA). This is the basis of every trust relationship among the participants, so a strict procedure has been set up for the connection between two principals.

Once the trust relationship has been established via a trusted set of Certification Authorities, participants can use this to mutually authenticate each other in a connection. In normal usage a user will authenticate using a “short-time proxy” certificate, which is a proxy the user generates using his long-term certificate, typically valid for 12 hours [8].

The implementation of this authentication may vary according to the chosen language and environment: in EDG the services written in C mostly use the *GSI* library itself; the Java/Tomcat web services use the *EDG Java Security* package; and the Apache based web services use the modified *mod_ssl* module (see Section 6).

In parallel to the authentication, a client application may choose to delegate its credentials so that the remote process may act on its behalf. Currently, EDG uses the *GSI*’s delegation protocol [5] for this, but its limitations required the introduction of the *G-HTTPS* [4] protocol in the web services.

2.1. *Certification Authority Coordination*

A *Certification Authority Coordination Group* (CACG) was established in 2001 to define a common authentication infrastructure that was trusted by relying parties in Europe, North America and Russia.

Each CA seeking approval is required to write its own Certification Policy and Certification Practice Statement (CP/CPS) [9] and demonstrate to the group that the setup is secure. This is usually done in person at a formal meeting where detailed questions about the CP/CPS, the practices, the Registration Authority (RA) structure, etc. are answered. The CA is checked against an agreed list of minimum requirements (see Section 2.3). After satisfying this peer review a CA will be recognized as an ‘accepted’ CA and the root certificate added to the repository.

2.2. *CA Status*

In a European Grid context, currently there are 28 approved national or organizational certification authorities, each with associated registration authorities that check identity, with CNRS (France) acting as a 'catch-all' CA for those who do not have local CA, with appropriate RA mechanisms. The certification authorities use a variety of software to provide the service. Systems based on OpenSSL, including OpenCA, are popular. Commercial products are used by a number of CAs.

A repository (TACAR) [10] was created to provide a central point for relying parties to retrieve certificates, revocation lists and policies. CA information is available in RPM (RedHat Package Manager) format to allow easy installation.

In April 2004, the members of the CACG formed the European Policy Management Authority for Grid Authentication in e-Science (EUGridPMA) [11], and this approach to grid authentication policy management was endorsed by the EU eInfrastructures Reflection Group (eIRG) [12]. Similar PMAs now exist in the Asian Pacific [13] and are being proposed for the Americas. Interoperation between these PMAs is fostered via the International Grid Federation (IGF) [14].

2.3. *Minimum Requirements for a CA*

One of the major activities of a grid authentication policy management body is in creating and maintaining a set of minimum requirements and best practices for operating a CA that is "acceptable and trustworthy" as defined by the relying parties and related grid projects, taking into account the level of risk associated with the assets the projects seek to protect. By way of example, for the EUGridPMA these minimum requirements have evolved in an iterative discursive fashion – largely as a result of the difficulties that arise when operating between different linguistic, administrative, network and security domains as occur across national boundaries. The significant points are presented below.

PKI Structure. Within each country, large region or international organization there should be a single certification authority with a wide network of registration authorities.

Certification Authority. The CA hardware must be physically secure and operated offline unless it uses secure cryptography hardware. Each CA must have a CP/CPS. There are restrictions on the CA key, certificate and certificate revocation lists. Records must be kept and the CA must allow audits to take place.

Registration Authority. The RA must validate the identity of a person based on photographic identification and/or official documents. The RA must communicate securely with the CA and must keep records.

End-entity Certificates. Private keys must have a minimum length of 1024 bits, a maximum lifetime of 1 year and must not be generated by the RA or CA.

The latest version of the Minimum Requirements document of the European Policy Management Authority for Grid Authentication in e-Science (EU-GridPMA) is publicly available from http://www.eugridpma.org/.

2.4. *Delegation*

Sometimes it is necessary for a user to grant access rights to either another user or, more commonly, an agent acting on behalf of the user. For example, in EDG, if a user asks a Replica Manager (RM) to copy a file from one Storage Element (SE) to another, the RM must be able to transfer the file on behalf of the user. As another example, a job that is running on an EDG Worker Node (WN) may need to read or write files on behalf of the user who submitted the job. Within the security architecture of [4] this task is solved with delegation.

2.4.1. *Delegation Mechanism*

User credentials are delegated to an agent by creating a new derived proxy [5] (new certificate and private key) in the agent and copying the entire proxy certificate chain to the agent. This applies whether the proxy is a GSI proxy or a VOMS proxy (see later).

The user must otherwise trust the receiving agent to act only in the way the user intended (e.g. only execute the job as specified by the user) and in a manner appropriate to the service description (e.g. act as a compute element).

Once the Grid scales to a larger number of sites, in particular multidisciplinary or commercial research where not all collaborators can trust each other, there is good reason to review the security architecture. Directions for future research could include limiting the proxy's strength (like the Globus limited proxy), or endowing the proxy with a specific purpose (e.g. can

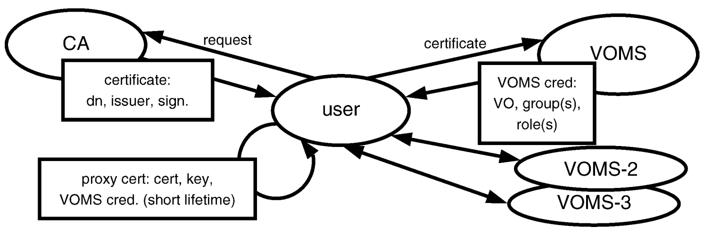

Figure 1. Authentication and Gathering of Authorization credentials.

only be used to read a fixed list of files from a fixed list of SEs), etc.

3. Global Authorization

In an access control decision there are several rules and policies to take into account: global (e.g. organizational membership) and local (e.g. banned users). Both pieces of information have to be available in order to make the decision. Grid access is granted according to membership of Virtual Organizations (VOs).

In the early versions of the Globus software, this membership information was recorded in a local *grid-mapfile*. This required a user to have an account on all resources they wished to have access to, and their DN was mapped onto that account via the *grid-mapfile*.

The problem with the *grid-mapfile* is the same as with a local *passwd* file in a cluster environment: synchronization. All of them have to be up to date so that the user can access all resources uniformly.

3.1. *VO LDAP Server and Grid-mapfile Generation*

Managing *grid-mapfiles* locally does not scale, but two simple concepts offer basic solutions to the problems: *VO directory services*, and *pool accounts*.

As an example of the first of these, the *VO LDAP server* [15] is a directory service for managing VO membership, published via an LDAP service. It allows administrators to manage a natural hierarchy of both membership of VOs and membership groups within VOs, similarly to the user management tools of traditional cluster environments (e.g. NIS, NIS+, LDAP).

The second concept resolves the scaling problem of grid-mapfiles: the creation of real accounts on the local fabric for the mapping of Grid users. This problem was solved by leasing *pool accounts* (see Section 4.1) to users, and by creating dynamic mapping of the DNs onto these pool accounts based on VO membership in a grid-mapfile. If a user is a member of a VO, his DN is automatically mapped onto a pool account that may be leased to any member of the VO. This allows for a "coarse grained" authorization based on VO membership.

An implicit further problem is the distribution of the grid-mapfile. There are many ways this can be accomplished; for example in EDG it is done via the *edg-mkgridmap* utility. The *edg-mkgridmap* utility is run as a cron job on each node each day, accessing the VO servers, *pulling* the membership list, and generating a daily updated grid-mapfile. Both VOMS servers (see below) and VO LDAP servers are able to serve edg-mkgridmap to enable the grid-mapfile to be created.

3.2. *The Virtual Organization Membership Service*

The basic global authorization solution, grid-mapfiles, described in Section 3.1 has many shortcomings. The two most important are: (a) the file might only be updated periodically, meaning users will have to wait until then before they can start using the system; (b) a user can only be mapped onto one VO as the mapping is based on identity, he cannot choose to use different VOs for different jobs.

These difficulties led to the concept of a *Virtual Organization Membership Service* (VOMS), a central database of VO members, a joint effort between the EDG and the *DataTAG* [16] projects.

VOMS allows an authenticated and authorized administrator to manage the VOMS database, and an authenticated user (or any principal) to request membership of a VO, and request group membership, role, and capability entitlements [17]. Once the user has been granted the appropriate VO membership and attributes within a VO, he may request a short lived credential. The user runs `voms-proxy-init` with optional arguments relating to which VOs, groups, roles, and capabilities he wishes for his current credential. VOMS issues a short lived Attribute Certificate [18] to the authenticated user, which the user may then present to resources on the Grid.

The credential is in the form of a private, non-critical extension included into the user's X.509 proxy certificate. The inclusion allows transparent transfer of the credential, allowing basic GSI services to use the proxy as before. VOMS allows use of a push model, the service does not have to pull VO membership of all potential users. To provide backward compatibility VOMS can also fall back to a pull model if need be to serve membership information for the grid-mapfile generation.

VOMS supports membership of multiple VOs, hence a VOMS proxy certificate may contain information from more than one VO. Figure 1 shows how credentials of multiple VOs are collected.

4. Running Jobs on a CE

As a concrete example, this section considers the submission of Jobs, i.e. computational processes, through the Globus gatekeeper to the EDG Computing Element (CE) as a way of presenting the local authorization enforcement components. As will be seen, this highlights the need for additional components, which we have satisfied with *SlashGrid*, *LCAS* and *LCMAPS*. The submission can be either a direct request or a job scheduled by the Workload Management System (or Resource Broker) – either way, the job arrives with a delegated proxy certificate which may also include VOMS credentials.

In early versions of Globus, the *gatekeeper* service simply used the grid-mapfile to make an authorization decision and map the job into a local account.

The first change in this scheme was the proposal for pool accounts (see below in Section 4.1) and, the above-mentioned generation of the grid-mapfile. However, at this point the basic authorization and mapping mechanism was unchanged.

There were a couple of weak points in this concept: users could not choose their role (set of assigned Unix groups) in the local fabric and could not acquire local credentials in some large clusters (Kerberos or AFS tokens). These requirements led to the notion of the components LCAS and LCMAPS (see below in Section 4.3).

The final component in this section is the credential renewal mechanism for long running jobs, where the *MyProxy* server plays the key role. This is necessary because the "short-time" proxy certificates used for job submission may expire before the job has completed.

4.1. *Pool Accounts*

The Pool Accounts [19] mechanism extends the Globus grid-mapfile system with a directory of lock files which hold the mappings between a current user's DN and a single pool account which is uniquely leased for the duration of the job.

The functionality can be implemented in any language or environment that supports Unix filesystem operations, or as a modification to the Globus grid-mapfile handling functions (therefore available to the Gatekeeper and GridFTP services derived from Globus).

A more generalised version of this system is the combination of LCMAPS and Job Repository (see Section 4.4) components.

4.2. *Slashgrid*

Slashgrid [19] is a conceptual framework for building filesystems where access control is dependent on Grid rather than Unix credentials. This is largely motivated as an extension to the Pool Accounts concept, and the design is an alternative to conventional Unix filesystems, which may be used by site administrators to simplify the management of disks to which Grid jobs have access.

The key feature of Slashgrid is the implementation of a local filesystem (layered on an existing one), which allows access control to be specified in terms of attributes found in Grid Credentials (see Section 6.1). It complements the idea of pool accounts by providing access to permanent files on local disk using the grid identity, regardless of the actual numeric userid of a pool account.

4.3. *LCAS*

LCAS, the Local Centre Authorization Service [20, 21], is an authorization decision engine that adds access controls to the *gatekeeper*. The gatekeeper accepts job requests from external sources (like the end-user or the workload management system) over an authenticated channel.

The concept of LCAS is that different independent authorization modules may be "plugged-in", thus creating a flexible system. The plug-in framework enables multiple independent authorization modules to collectively grant or deny access to the resource. The decision is based on the requested resources (expressed via the Resource Specification Language, or RSL), the identity of the requester, and the authorization credentials presented by the end-user in the proxy

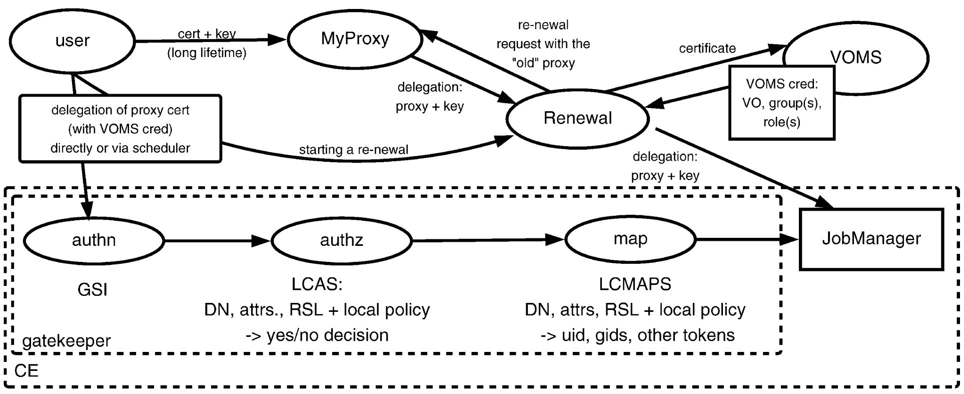

Figure 2. Authentication and Authorization in a Computing Element.

certificate. If VOMS is used, this will be the VO, Group, Role, and Capability combinations the user has acquired from VOMS.

As part of the implemented LCAS system, the following plug-ins are provided:

– A module that inspects an allowed-user and banned-user list only based on the requester's subject identifier (DN).
– A wallclock-time limiting module defining "fabric opening hours".
– A VOMS module that compares VOMS attributes against a site-local access control list.

External parties can develop their own plug-ins to provide additional functionality without the need to re-compile the LCAS framework. The framework can also be used for other services, like the *GridFTP* server.

One such plugin (the Site AuthoriZation, or SAZ, plugin [22]) has been written by personnel of the Fermi National Accelerator Laboratory (IL, USA) to provide two-step authorization in case multiple fabrics must incorporate a centrally coordinated authorization decision.

4.4. *LCMAPS*

LCMAPS is a Local Credential MAPping Service [20, 21] which allows the acquisition of credentials (like Unix user ids) for Grid jobs that run on the Local fabric. LCMAPS offers detailed support for plug-in modules.

There are two different module types: "acquisition" and "enforcement". The acquisition modules collect information on the credentials to be used for a particular request, but do not enforce these credentials. This task is performed by enforcement plugins.

Plug-in modules are available providing the following functionality:

– Mapping onto a local Unix account and group. This is a static mapping from the user's DN to a uid based on a plain-text grid-mapfile.
– Mapping onto *pool accounts*, but extended so that a Unix Group is also set.
– Full VOMS support. This also allows VOMS groups, roles, and capabilities to be mapped onto Unix groups, possibly taken from a pool of groups similar to the *pool accounts* system.
– Mapping from the DN onto local AFS tokens. This is carried out if, for example, the local home directory is on an AFS file system.
– POSIX in-process enforcement which set the real and effective user and group ID for the current process.
– Updating a fabric-central (LDAP) user directory for user ID and group ID information. This is required in the environment of a cluster fabric that uses a batch system for running jobs, since the in-process enforcement mentioned above will only affect the credential used for submitting the job to the local batch system, and not the actual job execution on back-end worker nodes in the cluster.

In a similar way as for LCAS, other plug-ins may be written to provide functionality specifically required by a site.

The Job Repository (JR) [20, 21] maintains a record of the credential information associated with all jobs running inside the fabric. By incorporating data persistence in the architecture (using a database as an archiving back-end), the JR can also be used to obtain information on credential mappings that were in effect for a particular job in the past. The information is provided to the JR via the LCMAPS plug-in mechanism, and by modification to the job management system.

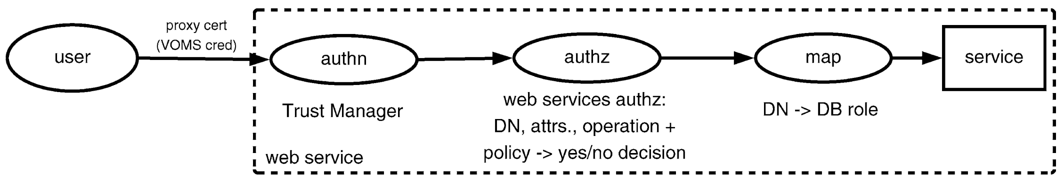

Figure 3. Authentication and Authorization on Java based web services.

4.5. *Credential Renewal*

In the architecture, jobs are required to have a valid certificate for all the time they are running. However, a job usually has a credential with a lifetime of a few hours, which is often shorter than the time spent by the job waiting or executing in queues.

The Credential Renewal service has been conceived to support long-running jobs. It uses the MyProxy package [23], where users store their long-time proxy before they submit jobs. Typically, users store a proxy of their CA signed certificate which is valid for a few weeks.

The MyProxy server restricts access to users' proxies to a very small set of clients, which always have to prove possession of an older user's proxy before retrieving a newer one. The Credential Renewal service must be among these allowed clients.

The Credential Renewal service registers the proxy certificate of a submitted job and, by periodically contacting the MyProxy server, ensures that a new proxy is issued prior to the old one expiring. If the registered Proxy certificate contains VOMS information, the Credential Renewal service also contacts the appropriate VOMS servers and renews this information as well (see Figure 2).

5. Java Based Web Services

If middleware components are implemented as Java based web services, it is important to develop a security system for them which is compatible with other parts of the security architecture.

The Java Trust Manager [24] is a tool that allows authentication to take place in the Java environment using proxy certificates. Java Services run inside the Tomcat servlet container, and authentication takes place on connection to this service container (see *authn* on Figure 3).

The Java Trust Manager allows a service to authenticate itself to the client with either a CA signed certificate (such as the host certificate) or a proxy of the host certificate. A user will usually authenticate with a proxy.

5.1. *Authorization for Java Web Services*

A Java authorization manager [24] has been designed as an integral part of the security architecture to carry out authorization within Java web services (see *authz* and *map* on Figure 3).

The coarse grained authorization filters the user requests according to the system settings on front of the web service. Only the principals in the grid-mapfile, in a local policy, or with a valid proxy certificate are allowed to access the service. It also presents the relevant attributes (either the VOMS roles, groups and capabilities or the attributes derived from the users' DN by this package) to the service, which can then make the service-specific fine grained authorization decisions based on these attributes (e.g. role and group).

Medium grained authorization is a further refinement of this method and contains lists of actions inside a service the principal with the corresponding role can access. For example, only an administrator can remove files from the replication system, a production manager can add files and a normal user can only replicate and access files.

The authorization system has an optional web interface that can be used by an administrator to securely monitor, manage, and update the configuration at runtime (no restart or downtime required).

6. Apache Authorization

Apache authentication is part of the GridSite [25] development from the UK GridPP [26] project. GridSite is a set of extensions to the Apache web server and a toolkit for Grid credentials. This too needs to be compatible with the remainder of the security architecture.

Since 1998, the Apache web server has had support for user authentication with X.509 certificates

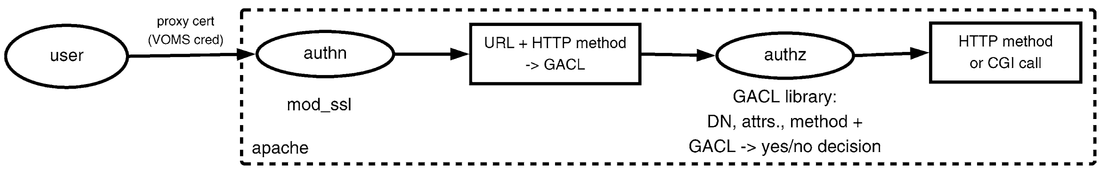

Figure 4. Authentication and Authorization in Apache.

via the mod_ssl loadable module. The GSI proxies are an extended form of these X.509 user certificates, but are not acceptable to the standard version of mod_ssl.

To take advantage of Apache's stability, performance, and ubiquity, the mod_ssl module has been modified to use the GridSite library to verify GSI proxies and to check the extended chain of trust they use. The extended version of this module is called mod_ssl-gridsite [25].

If the proxy is valid, the modified mod_ssl-gridsite makes the associated user identity available to the rest of the Apache framework (see *authn* on Figure 4), and so available to existing technologies used for building dynamic content and services which are not in themselves aware of GSI.

The GridSite library also extracts VOMS credential attributes from the proxy certificate if present, and mod_ssl-gridsite makes them available to the rest of Apache by the standard mechanisms.

6.1. *GridSite Authorization*

To complement the Grid authentication extensions to Apache, a new Apache module, mod_gridsite, has also been created to implement authorization based on X.509, GSI and VOMS credentials, and add support for writing to files via the HTTP PUT method in addition to the standard reading via HTTP GET (see *authz* on Figure 4).

This makes it possible to use Apache as a high performance file-server supporting Grid credential authorization, similar to the GridFTP server but using HTTP(S).

The mod_gridsite module uses GACL, the "Grid Access Control Language" [19] provided by the GridSite/GACL library. This allows access control to be specified in terms of attributes found in Grid Credentials. This simple language is being developed to be a subset of the XACML [27] schema.

Complementing mod_gridsite, a standalone utility provides tools for the interactive management of files, directories, and GACL access control files via a web interface.

6.2. *G-HTTPS*

Delegation is one of the major components of the trust model we use, but is absent from standard HTTPS. The G-HTTPS [4] extensions add additional methods and headers to HTTP over SSL which allow GSI proxies to be delegated over an HTTPS connection without modifying the underlying SSL or HTTP implementations.

A "proof of concept" implementation of G-HTTPS in C has been implemented as part of GridSite.

7. Future Directions

The security architecture and the various components implemented and presented in this paper have successfully satisfied many of the security requirements [2]. Some problems, however, have either not been fully solved or tested. We suggest that some of these areas are worthy of consideration by other Grid projects for future development, and give a short summary of them here; more details are given in [28].

Certificate Request Applet. Java applets have been proposed, and implemented, for making certificate requests. This is useful for web-based CA interfaces, partly to overcome browser support problems. Other advantages include the ability to sign the applet, validate the user's input, and check the strength of the user's passphrase.

Automated Evaluation of Trust in CAs. The CACG developed a tool to assist in the evaluation of the policies and procedures of each of the trusted certification authorities. The trust matrices, which provide a structured view of the features of each CA, and the peer assessments within the CACG, could be developed further to enable more automation in this area.

Online Certificate Services. Traditionally, grid certification authorities have been operated disconnected from any network, reducing the risk of compromise of the CA signing key. Online certificate services are those which store private keys, and generate or sign certificates on a network-connected system. In LCG [29], one of the CAs is a Kerberized Certification Authority (KCA) [30], and ESnet is proposing a minimum requirements profile for online services, which should allow policy management of this type of service. The SLAC Virtual Smart Card system [31] has also been discussed by the EUGridPMA.

Delegating Specific Rights. If a user creates a proxy and delegates this to a service to act on their behalf, they are delegating all the rights that their proxy allows. For example, if a user delegates a VOMS proxy, that will delegate all the rights that are defined by that VOMS proxy. To solve this problem, we would need to consider another form of delegation, where the user only grants the right to carry out a single specified action or set of actions.

User to User Delegation. In the authorization model there is no provision for a user to delegate their rights to another user. To gain a right, a user must be given that authorization by a VO administrator, and the user may only delegate that right to certain services.

Full Use of VOMS Credentials. We believe that when all appropriate services are fully able to use VOMS credentials, a very effective authorization system will be in place.

Mutual Authorization. In some circumstances it is necessary to authorize a service to carry out an action, as well as authorizing user access to a service. For example, some computers in a student lab may be set up to allow a particular VO access to those resources, but the VO will not want confidential data to be processed there.

This could be done by giving resources credentials from the Virtual Organization, so the user's software checks the credentials – and a job is only submitted if the resource is authorized. VOMS could be used in this way, as there is no reason why a resource cannot be issued with VOMS credentials in the same way a user is.

Authorization to Access Distributed Information. For information services (such as R-GMA) which gather information from several sources, authorization gets more complicated. This is particularly a concern if some of the information is highly confidential.

This may be taken care of by ensuring that information is only placed on authorized hosts and trusting the service to obey all access rules. Additional safeguards may be put in place by only allowing confidential information into the system if an authorized principal has requested it or by encrypting any confidential information.

Confidentiality. Certain applications, such as biomedical applications handle confidential data. The storage, processing, movement and retrieval of such data in the Grid environment is a complex problem, involving various security components, that needs more work in the future.

An example design was presented in [4] for storing a confidential file, to meet the bio-medical confidentiality requirements, but this has not been implemented.

Further Development of LCAS and LCMAPS. In the near future, LCAS will provide additional interfaces for the Authorization Call-Out mechanism in GT3, the current production version of the Globus Toolkit, and it will serve as the authorization framework for native services in the EGEE middleware. LCAS will take on the use of XACML [27] for expressing VO access rights. The development will be streamlined with respect to the authorization for hosted services. In the long run, LCAS will provide authorization services in federated Grids, and move toward an engine for negotiating compatible authorization methods among different Grids.

The Job Repository will continue to be enhanced to support more complex queries and aid site administrators and users in problem tracking and accounting.

Authorization Standards. We have taken an active role in the work of the Global Grid Forum, where interoperability schemes are currently discussed and investigated. We have been especially active in the Authorization Frameworks and Mechanisms Working Group, the Site Authentication, Authorization and Accounting Research Group and the Open Grid Services Architecture Authorization Working Group.

We expect this to result in convergence between our VOMS, GACL and LCAS systems and the relevant Grid authorization standards.

In addition, other encoding formats should be investigated, such as the XML based format defined in SAML by OASIS [32].

Acknowledgments

The authors wish to acknowledge the support of the EU and many national funding bodies, institutes and projects that enabled their staff to participate in the activities of the DataGrid Security Coordination Group. We are also thankful for the contributions of L. Gommans (University of Amsterdam), Philippa Hopcroft (née Broadfoot), G. Lowe (Oxford University), B.A. Coghlan (Trinity College, Dublin), J. Astalos (Slovak Academy of Sciences), R.D. Cowles (Stanford Linear Accelerator Center), U. Epting (Forschungszentrum Karlsruhe), T.J. Genovese (ESnet/LBNL), J. Gomes (Laboratório de Instrumentação e Física Experimental de Partículas), M. Gug (CERN), A.B. Hanushevsky (Stanford Linear Accelerator Center), M. Helm (ESnet/LBNL), C. Kanellopoulos (Aristotle University of Thessaloniki), R. Marco (Instituto de Física de Cantabria), I. Neilson (CERN), D. Quesnel (Canarie, Ontario), I. Schaeffner (Forschungszentrum Karlsruhe), L. Shamardin (Moscow State University), D. Skow (Fermi National Accelerator Laboratory), M. Sova (CESNET), A. Wäänänen (NBI, Copenhagen), P. Wolniewicz (Poznań Supercomputing and Networking Center), W. Xing (University of Cyprus).

We are grateful to all of our colleagues in the EU DataGrid and other projects for many illuminating discussions and for providing very useful feedback on our designs and implementations.

References

1. F. Gagliardi, B. Jones, M. Reale and S. Burke, “European DataGrid Project: Experiences of Deploying a Large Scale Testbed for E-Science Applications”, in *Performance Evaluation of Complex Systems: Techniques and Tools, Performance 2002, Tutorial Lectures*. Lecture Notes in Computer Science, Vol. 2459, Springer, 2002. http://www.edg.org
2. D. Kelsey and L. Cornwall, “DataGrid Security Requirements and Testbed-1 Security Implementation, D7.5”, The European DataGrid Project, 2002-05-28. https://edms.cern.ch/document/340234
3. R. Alfieri, R. Cecchini, V. Ciaschini, L. dell Agnello, A. Gianoli, F. Spataro, F. Bonnassieux, P. Broadfoot, G. Lowe, L. Cornwall, J. Jensen, D. Kelsey, A. Frohner, D.L. Groep, W. Som de Cerff, M. Steenbakkers, G. Venekamp, D. Kouril, A. McNab, O. Mulmo, M. Silander, J. Hahkala and K. Lorentey, “Managing Dynamic User Communities in a Grid of Autonomous Resources”, in *Proceedings of Computing in High Energy Physics 2003*, La Jolla – San Diego, March 24–28 2003.
4. A. Frohner et al., “DataGrid Security Design, D7.6”, The European DataGrid Project, 2003-03-28. https://edms.cern.ch/document/344562
5. I. Foster, C. Kesselman, G. Tsudik and S. Tuecke, “A Security Architecture for Computational Grids”, in *Proceedings of the 5th ACM Conference on Computer and Communications Security*, pp. 83–92, 1998. Describes techniques for authentication in wide area computing environments.
6. IETF, Public-Key Infrastructure (pkix) Charter.
7. C. Adams and S. Lloyd, *Understanding PKI: Concepts, Standards, and Deployment Considerations*, 2nd ed. Addison-Wesley, 2002.
8. S. Tuecke, V. Welch, D. Engert, L. Pearlman and M. Thompson, “Internet X.509 Public Key Infrastructure (PKI) Proxy Certificate Profile”, RFC3820, June 2004. http://www.rfc-editor.org/rfc/rfc3820.txt
9. S. Chokhani, W. Ford, R. Sabett, C. Merrill and S. Wu, “Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework”, RFC3647, November 2003. http://www.rfc-editor.org/rfc/rfc3647.txt
10. Policy of the TERENA Academic CA Repository, Version 1.0, 29 January 2004.
11. European Policy Management Authority for Grid Authentication in e-Science Charter, Version 1.0, 2004-04-01.
12. eInfrastructure Reflection Group, White Paper, Dublin, 2004-04-13.
13. Asia Pacific Grid Policy Management Authority Charter, Version 1.0, 2004-06-01.
14. International Grid Policy Management Authority. http://gridpma.org (2003-03-01).
15. J.A. Templon and D.A. Groep, “VO Server Information”, Documentation of the European DataGrid Project, 2004-10-23.
16. Research and Technological Development for a Data TransAtlantic Grid. http://www.datatag.org (2004-06-01).
17. R. Alfieri, R. Cecchini, V. Ciaschini, L. dell Agnello, Á. Frohner, A. Gianoli, K. Lőrentey and F. Spataro, “VOMS, an Authorization System for Virtual Organizations”, in *Proceedings of the 1st European Across Grids Conference*, Santiago de Compostela, February 13–14 2003.
18. S. Farrell and R. Housley, “An Internet Attribute Certificate Profile for Authorization”, RFC3281, April 2002. http://www.rfc-editor.org/rfc/rfc3281.txt
19. A. McNab, “Grid-Based Access Control and User Management for Unix Environments, Filesystems, Web Sites and Virtual Organisations”, in *Proceedings of CHEP 2003*, La Jolla, CA, March 2003.
20. M. Steenbakkers, “Guide to LCAS, Version 1.1.16”, 15 September 2003. Documentation of the European DataGrid Project.
21. M. Steenbakkers, “Guide to LCMAPS, Version 0.0.16”, 15 September 2003. Documentation of the European DataGrid Project.
22. D. Skow, I. Mandrichenko and V. Sehkri, “Site Authorization Service (SAZ)”, in *Proceedings of CHEP 2003*, La Jolla, CA, eConf C0303241, 2003, TUBT007 [arXiv:cs.dc/0306100].
23. J. Novotny, S. Tuecke and V. Welch, “An Online Credential Repository for the Grid: MyProxy”, in *Proceedings of the 10th International Symposium on High Performance Distributed Computing (HPDC-10)*, IEEE Press, August 2001.
24. J. Hahkala, “Guide for EDG Security for Java 1.5.6”, 13 October 2003. Documentation of the European DataGrid Project.
25. A.T. Doyle, S.L. Lloyd and A. McNab, “GridSite, GACL and SlashGrid: Giving Grid Security to Web and File Applications”, in *Proceedings of the UK e-Science All Hands Conference*, Sheffield, September 2002.

26. D. Britton, P. Clarke, J. Coles, D. Colling, A. Doyle, S.M. Fisher, A.C. Irving, J. Jensen, A. McNab and D. Newbold, "A Grid for Particle Physics – from Testbed to Production", in *Proceedings of the UK e-Science All Hands Conference*, Nottingham, September 2004.
27. eXtensible Access Control Markup Language (XACML), Version 1.0, OASIS Standard, 18 February 2003. http://www.oasis-open.org/specs/index.php#xacmlv1.0.
28. L. Cornwall et al., "DataGrid Security Implementation, D7.7", European DataGrid Project, 2004-01-27. https://edms.cern.ch/document/414762
29. LHC Computing Grid Project. http://cern.ch/lcg (2004-06-01).
30. O. Kornievskaia, P. Honeyman, B. Doster and K. Coffman, "Kerberized Credential Translation: A Solution to Web Access Control", February 2001, in *Proceedings of USENIX Security Symposium*, Washington, DC (August 2001).
31. A. Hanushevsky and R. Cowles, "Mechanisms to Secure x.509 Grid Certificates", in *Proceedings of CHEP 2003*.
32. Security Assertion Markup Language (SAML), Version 1.1, OASIS Standard, 2 September 2003. http://www.oasis-open.org/specs/index.php#samlv1.1.