Security in distributed metadata catalogues

Nuno Santos[1], and Birger Koblitz[2*]

[1] *Distributed Systems Laboratory, Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland*
[2] *European Organization for Nuclear Research (CERN), Geneva, Switzerland*

SUMMARY

Catalogue Services provide the discovery and location mechanisms that allow users and applications to locate data on Grids. Replication is a highly desirable feature in these services, since it provides the scalability and reliability required on large Data Grids and is the basis for federating catalogues from different organisations. Grid Catalogues are often used to store sensitive data and must have access control mechanisms to protect their data. Replication has to take this security policy into account, making sure that replicated information cannot be abused but allowing some flexibility like selective replication for the sites depending on the level of trust in them. In this paper we discuss the security requirements and implications of several replication scenarios for Grid Catalogues based on experienced gained within the EGEE project. Using the security infrastructure of the EGEE Grid as a basis, we then propose a security architecture for replicated Grid Catalogues, which, among other features, supports partial and total replication of the security mechanisms on the master. The implementation of this architecture in the AMGA Metadata Catalogue of the EGEE project is then described including the application to a complex scenario in a Biomedical application.

KEY WORDS: Grid Computing, Security, Distributed Databases, Metadata

1. Introduction

File Catalogues and Metadata Catalogues play a vital role on Data Grids, providing users and applications the means to discover and locate data among the many sites of a Grid. File Catalogues map logical filenames to the physical location of one or more replicas of the file, while Metadata Catalogues store a description of the contents of the files which is specific to the application and is used to search for files based on their contents.

As Grid Catalogues often contain sensitive information, they typically have authentication and authorisation mechanisms. To provide seamless integration with the surrounding Grid Infrastructure, Grid Catalogues use the standard security mechanisms of the Grid where they are deployed. Most Grids currently in operation use the Grid Security Infrastructure (GSI) model, where users are managed by a Virtual Organisation (VO) [1] and identified by Grid Certificates issued by their VO. To integrate with the GSI model, a Grid Catalogue must authenticate users using their Grid Certificates and use the subjects of

*Correspondence to: Mailbox J02300, IT/PSS, CERN, CH-1211 Genève 23, Switzerland (Birger.Koblitz@cern.ch)
Contract/grant sponsor: European Commission; contract/grant number: INFSO-RI-508833 (Birger Koblitz)
Contract/grant sponsor: Portuguese Foundation for Science and Technology (FCT); contract/grant number: SFRH/BD/17276/2004 (Nuno Santos)

these certificates for authorisation. These basic mechanisms, with some variations on the implementation of authorisation, are used by most of the Grid Catalogues in existence today.

Catalogues services implemented as a single central server are straightforward to protect using these well-known mechanisms. But to cope with the large size of a typical grid, Catalogues often have to be distributed and/or replicated. This creates several new security challenges, which must be addressed in the context of the existing Grid Security mechanisms.

In a standalone catalogue there is a single instance of the data that must be protected; this single catalogue acts as a central point where the security police is defined and enforced. When the data is replicated, it becomes much harder to enforce the security policy and prevent unauthorised access. There are two main dimensions to this problem: protecting the mechanisms supporting replication and enforcing the security policy across replicas.

Each new replica is a potential target for attack, increasing the vulnerability of the system. In addition, the process of transferring the data between replicas creates an additional attack window, either from snooping of data over the wire or from a malicious node posing as a legit replica.

Concerning the enforcement of the security policy, when data is replicated the node receiving the data (slave) has the power to do whatever it wants with it, including disregarding the security policy by allowing unauthorised users to access it. The node owning the data (the master) must trust the slave to enforce the security policy. If this is not possible, then it should not allow sensitive data to be replicated. When the slave is trusted, then the master must also have a way to communicate the information on the security policy to the slave, together with the data.

In this article we analyse the security requirements of replicated Grid catalogues in the context of the EGEE[†] Grid project's infrastructure. The EGEE Middleware, called gLite, is based on the Globus Toolkit [2] and uses its Grid Security Infrastructure (GSI) [3] as the basis for its own security. In addition, the EGEE Grid has its own service for managing users and groups called Virtual Organisations Membership Service (VOMS) [4]. We also propose a security architecture for replicated Grid Catalogues and describe its implementation in the AMGA metadata catalogue, which is part of the EGEE Middleware. Although we consider this particular deployment scenario, we believe our conclusions are general enough to be applied to other Grids projects.

In addition to the description of the requirements of the EGEE project, which led to the implementation of AMGA, we will show how our implementation can also be used in a complex biomedical scenario in the Health-e-Child project, which extensively uses AMGA's replication and security features.

The remainder of this paper is organised as follows. Section 2 provides the context for the article, by describing the security used on standalone Grid Catalogues using AMGA as an example. Section 3 discusses the security requirements of replicated catalogues on the context of the existing Grid security infrastructure. Section 4 brings the previous two sections together, by describing how the security mechanisms of AMGA were extended to support replication. An application of AMGA in a complex biomedical scenario follows in section 5. Section 6 discusses the related work on security in Grid Catalogues, and Section 7 concludes this article.

[†]`http://public.eu-egee.org/`

2. Context: Security and Replication on Grid Catalogues

Before discussing security on replicated Grid Catalogues, it is worth to provide a brief overview of the security mechanisms of a typical Grid Catalogue when in standalone mode. For that we will briefly describe the AMGA metadata catalogue of the EGEE project, which is a typical example of a Grid Catalogue. It is also the basis for our work, and we come back to it later in the article when we describe the implementation within AMGA of the security mechanisms we have developed for replicated catalogues. We also describe the replication model of AMGA, to better put on context the security challenges faced when replicating a catalogue.

2.1. Security Infrastructure of the EGEE Data Grid - VOMS

AMGA's security mechanisms are fully integrated with the mechanisms proved by the EGEE Grid. The basis for the EGEE Grid's security infrastructure is GSI and on top of GSI, a service to manage the security policy of a VO, the Virtual Organisation Membership Services (VOMS) [4], which was developed by the European DataGrid(EDG)[‡] project, a predecessor of EGEE.

VOMS manages the membership of the VO users, keeping track of the users, the groups to which they belong, the roles they have on those groups and, for more fine-grained policies, the capabilities of the users (a free-form string). This information is combined in the form of user attributes, consisting of triples of group, role, capabilities. This information is included in the proxy certificate of the user when it is created, so that grid services accessed by the user or by services acting on his behalf can use this information for authorisation decisions.

2.2. Security in the AMGA Metadata Catalogue

AMGA stores *entries* representing the entities that are being described, typically files. These entries are grouped into *collections*, which have a variable number of user-defined *attributes*, called the *schema* of the collection. Attributes are key/value pairs with type information and each entry assigns an individual value to the attributes of its collection. AMGA structures metadata in a hierarchy, similar to a file-system: collections can contain both entries and other collections. This hierarchical model has the advantage of being natural to users as it resembles a file-system, and of providing good scalability as metadata can be organised in sub-trees that can be queried independently. More details about AMGA can be found on [5], in the following, we will focus on AMGA's security features.

Figure 1 illustrates the main elements of the security mechanisms in AMGA. Access control follows the UNIX security model. Internally, each entry has an owner, and two sets of read, write and execute permissions: one for the user's group and another for the other users. In addition, ACLs can be associated with collections, specifying the read, write and execute permissions for arbitrary groups. It is also possible to have ACLs associated on a per-entry basis, but as this imposes a significant overhead on entry access, collections must be created explicitly with support for this feature.

The authentication mechanisms are depicted on Figure 2. Grid Certificates, either GSI or VOMS, is the preferred way of authenticating users on a Grid setting, so AMGA fully supports this option. For authorisation, it would be possible to use directly the information included in the certificate, like the Distinguished Name (DN) or the role and capabilities information encoded in a VOMS certificate. In fact,

[‡] `http://eu-datagrid.web.cern.ch/eu-datagrid/`

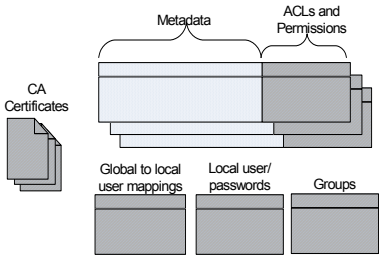

Figure 1. The elements of the security policy of AMGA are CA certificates stored in the server's file system, database tables storing credentials and mappings to local users and ACLs and permissions attached to any metadata entry.

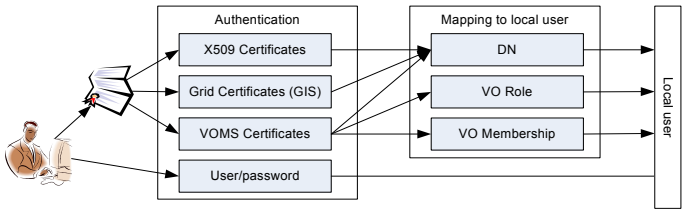

Figure 2. Authentication and mapping of global users to local users in AMGA.

this is done by some catalogues, like the Globus RLS [6]. But this approach has some limitations. One problem is the need for the user to deal with often very long DNs making the command line interfaces cumbersome to use. Another is the tight coupling between the local and the GSI/VOMS policy, as the local policy would always have to be defined in terms of GSI/VOMS users. In particular, it is not easy to add local users (without GSI/VOMS certificates) to the catalogue, and changes to the GSI/VOMS policy would often require changes to the local policy. This makes the catalogue harder to manage and limits its flexibility.

The solution used by AMGA is to support a fully independent local policy with its own users and groups. Authorisation is done solely using this local policy. This level of indirection between the GSI/VOMS and the local security policy requires a mechanism to map between the two of them. This mapping is done after authentication according to a policy defined by the catalogue administrator, based on either the DN, the VO role or the VO membership defined in the user certificate.

2.3. Replication on the AMGA Metadata Catalogue

AMGA implements replication using an asynchronous, master-slave model, described in detail in [7]. Asynchronous replication is used for coping with the high latency of Wide-Area Networks, since synchronous replication is known for its lack of scalability on WANs [8]. Master-slave replication was selected because it is the simplest model that covers the needs of the majority of our target applications, which have simple write patterns. The master-slave model works well as long as writes are not common or originate from the same geographical location.

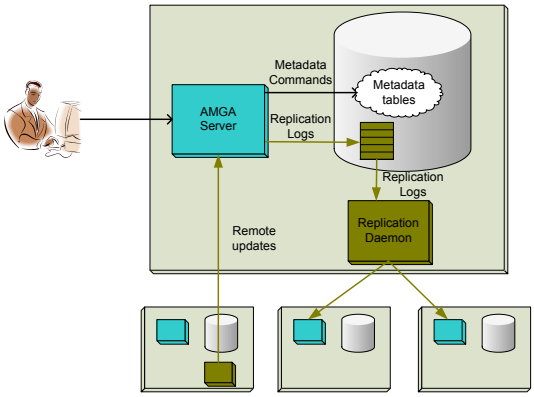

Figure 3. Replication architecture of AMGA.

Figure 3 illustrates the replication architecture of AMGA. After executing a command resulting in an update to the metadata (1), the master writes this command together with some additional context information to a log table on its database back-end (2). These commands are then shipped to slaves (3) that will execute them locally to bring themselves up-to-date. A node can function both as master and slave (4) at the same time.

Since the metadata commands are independent of the database back-end, replication works even between AMGA servers using back-ends from different vendors. The AMGA Server is only responsible for saving the commands to the replication log. The remainder of the functionality is implemented by the replication module, which is an independent daemon that can run on a different machine for better performance.

3. Security Requirements of Replicated Catalogues

Replicating a catalogue introduces new security challenges that do not apply to standalone catalogues. A first issue is to protect the mechanisms used for replication, including ensuring that no unauthorised node is able to replicate data from a master and that the data is kept safe during transmission. These problems can be addressed easily using the traditional authentication and authorisation mechanisms, as well as data encryption for transmission.

A more challenging problem is enforcing the security policy across several nodes. The data stored on the master has an associated security policy defining who can access it and under what conditions. The master enforces this policy with its client, using authentication and authorisation mechanisms. When the data is replicated, all the nodes with a replica of the data must collaborate to enforce the security policy. Whether other nodes can be trusted to do so depends on the level of trust between the master and the replica. A node that is not trusted cannot be granted access to sensitive data. If a node is trusted, then the master can send the data together with the information describing the security policy, so that the slave can enforce it locally.

In practice, the security policy of the master can either be extended (partially or totally) to the slave, or can be discarded, allowing the slave to define its own. The correct approach to this problem depends on several

factors, including the environment where the master and the slaves are deployed, the applications using the catalogue, and the level of privacy of the data.

There are three levels of privacy that data in a catalogue can have: public, VO-private or site-private. *Public data* can be accessed by any user, regardless of its VO, although the VO usually reserves the right to restrict the write-access. *VO-private* data can only be accessed by VO members, possibly only by a restricted group inside the VO. Finally, *site-private* data is owned by sites federating data to the VO.

In the case where the data is publicly accessible, there is no need to replicate security information together with the metadata, because every user has read access and because replicas are read-only (by design of the replication mechanism). However, the site might need to restrict the access to prevent denial-of-service attacks.

In the second case, where the data is VO-private, sites must replicate the entire security information to ensure that other nodes will enforce the same security restrictions. Since the only sites allowed to replicate the data will be VO sites and these are bound by contracts with the VO, they are trusted to respect and enforce the security policy defined at the master.

An example of the last scenario is the Medical Data Manager [9] developed by EGEE's BioMed community, where each institute (e.g., laboratories, hospitals, clinics) generates its own data and has its own security policy, however users and VO membership are still coordinated by the VO. One of the reasons for this setup are the local privacy laws, which normally restrict transfer of sensitive information to other jurisdictions. In this case, no replication of the metadata itself is done. However, users can be managed globally at the VO level and this information can be replicated to the sites. Sites may then define their local security policy in terms of these users. For instance, they may grant restricted access to specific users. In addition, the VO can organise federation of the individual catalogues into a global namespace.

In practice, Grid Catalogues might be also deployed locally under the control of the respective site, for instance, to use the replication features to provide increased scalability and performance of a local catalogue service. In this case, replicating the entire security configuration of the master is the simplest solution in terms of deployment and management. Another scenario consists of sites external to the VO that want to replicate publicly-readable data. In this case, the security policy of the master is of no interest to the site replicating the data, since it is not part of the same security infrastructure. Therefore, the site usually does not replicate the security information, instead defining its own policy.

The following summarises the requirements discussed above:

Extension of part or the totality of the security policy to slaves In Grid Catalogues this includes information about users, groups, ACLs and permissions. Users and groups are entities that exist on their own, but ACLs and permissions are always associated with collections and entries. As discussed above, it should be possible to replicate a collection with or without the associated security information.

Access control to replication A catalogue acting as a master should be able to define a fine-grained access control policy based on the collection and on the identity of the slave that is requesting a replica. For instance, a catalogue might allow replication of a collection only to nodes inside the same institute, or it might have some collections that should not be replicated to any other node. The same applies to the replication of the security policy, which should have access control mechanisms of its own to control who can replicate it.

Establish trust relationship between nodes Nodes must authenticate each other as a pre-requisite for access control to the replication.

4. Implementation of Replicated Security in AMGA

The security mechanisms supported by AMGA related to replication address three different issues: authentication between the nodes participating in the replication, access control mechanisms allowing the master to decide which other nodes are allowed to replicate which parts of its catalogue, and extension of the security policy of the master to slaves when requested. We will discuss each one of these in the next sections.

4.1. Authentication Between Nodes

When a slave contacts a master, it must authenticate using the same mechanisms as used by clients to authenticate to an AMGA server, including user/password and certificate authentication. Although plain X509 Certificates is the most appropriate mechanism for mutual authentication between nodes, the other mechanisms increase the flexibility of AMGA. Password authentication has the benefit of being simple to setup, while grid proxy certificates allow delegation from a user to a catalogue. The subjects associated with nodes are mapped to the same set of users as the AMGA clients, since this simplifies management by having a single interface for user management.

4.2. Access Control

There are two permissions in AMGA that control replication. The first is specified on a collection-basis and grants the owner (user or group) the right to replicate the collection and all its sub-collections. This allows administrators to control who is allowed to replicate which parts of the catalogue. The other permission, specified at catalogue level, controls the replication of users and groups information (see discussion on next section), by specifying who is allowed to replicate them.

4.3. Extension of Security Policy through Replication

Most of the information describing the security policy can be replicated to slaves, as shown in Figure 4.

Authentication Credentials (1) When replicating the security information, it is convenient if the user credentials stored at the master are also made available to the slave in an automated way. For certificate-based authentication, this is ensured by the Grid environment. In EGEE, all grid nodes have the gLite Middleware installed, which includes the certificates of the CA authorities, thus AMGA can simply assume that the required certificates are installed locally. For password-based authentication, the passwords are managed internally by AMGA, so they must be replicated together with the users and groups. AMGA does not store the password directly, instead it keeps only their hashes, which limits the damage in case of a security breach. These hashes are sent to the slave together with the user information. Sending the hashes is more secure than sending the passwords, but even so they are susceptible to a brute-force attack. In the end, it is the responsibility of the administrator to decide which sites are to be trusted with this information.

Users and Groups (2) Groups and users exist as separate entities from the metadata collections and must therefore be handled independently. Therefore, it is possible to replicate them separately. A node is only allowed to have a single security policy, either defined locally or adopted from the master. Hence, to replicate the users and groups from a remote node, it must not have any local ones defined. In addition, it is not allowed to replicate them from two masters at the same time. The rationale is that a node will replicate the users and

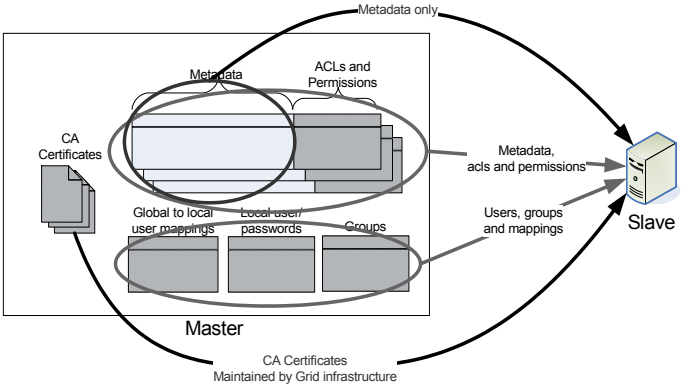

Figure 4. Different possibilities for replication of the security information on AMGA.

groups only when it belongs to the same institution as the master, therefore sharing the same security policy. The root user is a special case, in that it is never replicated and that the local root user is always preserved, to ensure that the catalogue can still be managed locally.

Mapping from global subjects to local users (3) The mappings between global and local users must also be replicated so that the security policy can be defined completely on the master. Therefore, they are replicated together with the user and groups information.

ACLs and permissions (4) The ACLs and permissions associated with metadata collections are only replicated if the slave requests it. This is supported as an option to the command that initiates replication of metadata collections.

4.4. Limitations

Replicating sensitive data, while convenient, increases the exposure to attacks. For instance, when a master allows a node to replicate metadata whose access is restricted, it is trusting the slave to respect and enforce the security policy associated with the metadata. But in fact, this is out of the control of the master, since the slave can disregard the mandatory security policy and expose the metadata to users who shouldn't have access to it. This problem can only be addressed by careful administration of the master, which should only allow replication of sensitive data to fully trusted nodes. Detection of security policy violations and dissuasion measures negotiated between sites may also be effective, but these are legal measures outside of the scope of the catalogue itself. A related issue is the increased exposure of the user passwords when they are replicated. The risk is minimised by storing and replicating only the hashes of the passwords, but even so, the hashes are susceptible to brute force attacks. By being replicated in several nodes, an attacker has more chances of compromising one of the nodes and getting hold of them. Once again, this risk must be addressed by careful system administration.

The replication mechanisms of AMGA, being based on asynchronous replication, impose a propagation delay between the master and the slaves that can range from a few seconds to several hours (if there is no

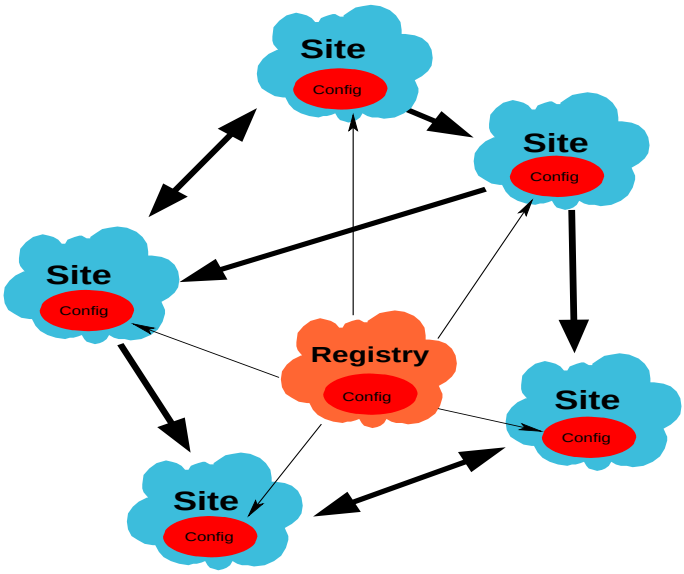

Figure 5. Node configuration in the distributed metadata catalogue of the Health-e-Child project. Sites replicate their data among each other, while the configuration is kept in a central catalogue and replicated to all participating sites.

connectivity between the slave and the master). During this period where the security information is not updated, the master and the slaves will be enforcing different security policies, which can be problematic. For instance, a user whose access was blocked at the master might still be able to access the data at a slave. The alternative would be to use synchronous replication for the security information, but this is not possible because the metadata and the security information are tightly coupled (access control information is part of the metadata) which means they must be replicated together and they must follow the same model.

5. Use case: Replication of Sensitive Metadata in Health-e-Child

In the following we will describe the usage of AMGA and in particular the application of the replication and security features in a prototype of the Health-e-Child project [§]. Health-e-Child is a large EU funded project that aims to provide a comprehensive view of children's health by integrating biomedical data from many hospitals and research centres using Grid technology. The data includes medical images and laboratory data, but also demographics and physican's notes. The integration of the vast amounts of corresponding metadata stored on different sites is implemented in the current prototype with AMGA. In the following we will describe several problems faced by the Health-e-Child project in their metadata management and how they have been solved using AMGA.

[§]For more information on Health-e-Child see `http://www.health-e-child.org`

Both data and metadata in Health-e-Child are highly sensitive and are subject to many legal restrictions, which depend on local law. The project has therefore set up a scheme to manage the data in which data including metadata is stored locally in the places where it is produced, and then controls the information flow via access restrictions when data is aggregated for the creation of studies comprising data of many participating institutions.

In the current prototype, sites are managed according to a schema shown in Figure 5. Information on participating sites and registered users is stored in a central database using AMGA and it is mandatory for the participating sites to replicate all of this information. The metadata itself is entered by the individual participating sites, which share it among themselves by direct replication of (parts of) the data including the access permissions. Certificates signed by a central authority are used to authenticate participating sites, with the Distinguished Name of the certificates being published in the registry catalogue so that they can be verified.

All metadata is being replicated to all participating sites by the prototype and presented by the local AMGA catalogues as a whole, allowing to perform queries on the entire globally available metadata locally at a site. Filtering of data (e.g. to fulfil legal regulations) is currently not done by the prototype. Apart from the security related information on users and participating sites, no central coordination of the metadata entered into the system is done, uniqueness of patient IDs and the like is achieved by using hash values or generated GUIDs.

Interestingly the replication system is also used for the configuration of the system. When a new node enters the distributed setup, it first uses its credentials to replicate the configuration of the distributed system from the registry catalogue and updates the central configuration table in order to register itself on the system. It also adds information about the provided data into this central configuration table, which is being replicated to the other participating sites. Scripts monitoring the information in the local AMGA databases will detect the new site and start replication of the provided data. It must be stressed that the registry catalogue is a central point of failure only for changes of the user and site catalogue. All participating systems will continue to work using their local copy in case the registry fails.

The prototype described here is currently under evaluation. Additional requirements will mandate some changes in the future, the most important one being the implementation of a distributed query system, which can complement the replication of data. This could also improve security as sensitive data could be distributedly stored to reduce the damage when a single system is compromised.

6. Related Work

The LHC File Catalogue [10] developed by the LCG project has similar security mechanisms as AMGA, including GSI authentication, VOMS based authorisation and access control using Unix style ACLs and permissions. It does not have an independent security policy, lacking local mechanisms for managing users and groups. The users are created on the fly from the subjects on the certificates used for authentication, while the groups are created from the VOMS enhanced certificates or from the grid-mapfiles. LFC has no replication mechanisms of its own.

The Storage Resource Broker (SRB) [11] includes the Metadata Catalogue Service (MCAT) which, in the terminology that has been used throughout this article, is both a file and metadata catalogue. It supports authentication using GSI, secure passwords and tickets, and has its own security policy with access control based on local users. MCAT also supports federation and replication [12]. When federated, each MCAT server retains local control of its users and access control policy, although they are aware of users from other

zones and can grant them access to local resources. This means that the security policy is always established locally, and therefore never extended to other MCAT servers.

The Globus project contains both a file catalogue, the Replica Location Service (RLS) [6], and a metadata catalogue, the Metadata Catalogue Service (MCAT) [13]. They both support GSI authentication. RLS has coarse-grained ACLs to apply to the whole catalogue, while MCAT support fine-grained ACLs at the object level. The local subjects used for the ACLs are, in both cases, the DNs of the user certificates, so their local policy depends on the global credentials. MCAT has no support for replication, while RLS uses index servers to provide a global list of files available on different replicate catalogues, but the authorisation decisions are performed only on the local catalogues, so no security information is pushed into the index server.

7. Conclusions

In this article we have discussed the security requirements of Grid Catalogues, focusing on those required to securely replicate the catalogues. Two types of requirements were identified: the catalogues must have authentication and authorisation mechanisms among themselves to control who is allowed to replicate the metadata, and they must support as an option the extension of the security policy associated with the metadata to the slaves. We have proposed an architecture to address these requirements and described its implementation on the AMGA metadata catalogue. To our knowledge, this is the first time the problem of extending the security policy of a Grid Catalogue to its replicas is addressed. We believe that such a feature is essential to replicate Grid Catalogues containing sensitive information, since it significantly simplifies management of the security policy by providing a single point of administration on the master, as opposed to a situation where each node must be administrated separately. The usefulness of our approach has been demonstrated by its application in the Health-e-Child project for a complex scenario involving highly sensitive data.

ACKNOWLEDGEMENTS

This work was performed within the LCG-ARDA project and the authors would like to thank Massimo Lamanna (CERN) for his valuable feedback and suggesting this interesting field of work in the first place. Viktor Pose (Dubna, Russia) performed intensive studies and testing of AMGA. We would also like to thank Andre Schiper for his helpful comments on the several draft versions of this paper. Finally, we would like to thank the Health-e-Child project members for the fruitful collaboration on understanding replication and security issues.

REFERENCES

1. Foster I. The anatomy of the grid: Enabling scalable virtual organizations. *Euro-Par '01: Proceedings of the 7th International Euro-Par Conference Manchester on Parallel Processing*, Springer-Verlag: London, UK, 2001; 1–4.
2. Foster I. Globus toolkit version 4: Software for service-oriented systems. *International Conference on Network and Parallel Computing (IFIP)* 2005; **3779**:2–13.
3. Foster I, Kesselman C, Tsudik G, Tuecke S. A security architecture for computational grids. *CCS '98: Proceedings of the 5th ACM conference on Computer and communications security*, ACM Press: New York, NY, USA, 1998; 83–92, doi:http://doi.acm.org/10.1145/288090.288111.
4. Alfieri R, Cecchini R, Ciaschini V, dell'Agnello L, Frohner A, Lorentey K, Spataro F. From gridmap-file to voms: managing authorization in a grid environment. *Future Generation Computer Systems* April 2005; **21**(4):549–558.
5. Santos N, Koblitz B, Pose V. The amga metadata service. *Journal of Grid Computing* July 2007; doi:10.1007/s10723-007-9084-6.
6. Chervenak A, Palavalli N, Bharathi S, Kesselman C, Schwartzkopf R. Performance and scalability of a replica location service. *Proceedings. 13th IEEE International Symposium on High performance Distributed Computing (HPDC-13)*, IEEE Computer Society: Washington, DC, USA, 2004; 182 – 191.

7. Santos N, Koblitz B. Distributed Metadata with the AMGA Metadata Catalog. *Workshop on Next-Generation Distributed Data Management - HPDC-15*, 2006.
8. Gray J, Helland P, O'Neil P, Shasha D. The dangers of replication and a solution. *SIGMOD '96: Proceedings of the 1996 ACM SIGMOD international conference on Management of data*, ACM Press: New York, NY, USA, 1996; 173–182, doi: http://doi.acm.org/10.1145/233269.233330.
9. Montagnat J, Jouvenot D, Pera C, Akos Frohner BKNSCL Peter Kunszt. Bridging clinical information systems and grid middleware: a medical data manager. *HealthGrid 2006*, 2006.
10. Baud JP, Caey J, Lemaitre S, Nicholson C, Smith D, Stewart G. Lcg data management: From edg to egee. *UK e-Science All Hands Meeting, Nottingham, UK*, 2005. URL `http://ppewww.ph.gla.ac.uk/preprints/2005/06/`.
11. Baru C, Moore R, Rajasekar A, Wan M. The sdsc storage resource broker. *CASCON '98: Conference of the Centre for Advanced Studies on Collaborative Research*, IBM Press, 1998.
12. Rajasekar A, Wan M, Moore R, Schroeder W. Data grid federation. *PDPTA - Special Session on New Trends in Distributed Data Access*, 2004; 541–546.
13. Deelman E, Singh G, Atkinson MP, Chervenak A, Hong NPC, Kesselman C, Patil S, Pearlman L, Su MH. Grid-based metadata services. *16th International Conference on Scientific and Statistical Database Management (SSDBM'04)*, 2004.