EGEE-III

gLite Security Architecture

EU DELIVERABLE: MJRA1.4

Document identifier:	EGEE-III-MJRA_1_4-935451-v1_0.doc
Date:	**30.1.2009**
Activity:	**JRA1**
Lead Partner:	**SWITCH**
Document status:	**DRAFT**
Document link:	https://edms.cern.ch/document/935451/1

Abstract: This document describes the overall global security architecture of the gLite middleware. It gives a high level view of the current situation and describes the foreseen evolution during the lifetime of the EGEE-III project.

Delivery Slip

	Name	Partner/Activity	Date	Signature
From				
Reviewed by	Moderator and reviewers			
Approved by	AMB & PMB			

Document Log

Issue	Date	Comment	Author/Partner
0-1	July 10, 2008	TOC	C.Witzig/SWITCH
0.2	Oct 3, 2008	First draft, call for comments from SCG	C.Witzig/SWITCH
0.3	Oct 31, 2008	Submitted to EGEE review	C.Witzig/SWITCH
0.9	Dec 1, 2008	Integrated all comments received (reviewers, SCG, D.Kouril, A.Frohner)	C.Witzig/SWITCH
1.0	Jan 30, 2009	Integrated last comments from reviewer	C.Witzig/SWITCH

Document Change Record

Issue	Item	Reason for Change

TABLE OF CONTENTS

1. INTRODUCTION 5

1.1. PURPOSE 5
1.2. DOCUMENT ORGANISATION 5
1.3. APPLICATION AREA 6
1.4. REFERENCES 6
1.5. DOCUMENT AMENDMENT PROCEDURE 7
1.6. TERMINOLOGY 7

2. EXECUTIVE SUMMARY 11

3. OVERVIEW OF THE GLITE SECURITY ARCHITECTURE 12

3.1. DEFINITION OF A SECURITY ARCHITECTURE 12
3.2. TRUST, AUTHENTICATION AND AUTHORIZATION: A TERMINOLOGY 14
3.3. VIRTUAL ORGANIZATIONS, SITES AND COMMON SERVICES 15

4. AUTHENTICATION 17

4.1. IDENTITY CREDENTIAL FORMATS 18
4.2. SHORT-LIVED CREDENTIAL SERVICES 19
4.3. BOOTSTRAPPING AUTHENTICATION 19
4.4. ENFORCING VALIDITY CONSTRAINTS 19
4.5. REVOCATION 20
4.6. CERTIFICATE RENEWAL 20
4.7. DELEGATION 20
4.8. RENEWAL OF PROXY CERTIFICATES 21
4.9. ANONYMITY, PRIVACY, PSEUDONYMITY 22
4.10. UNRESOLVED ISSUES AND WORK PLANNED IN EGEE-III 23

5. USER KEY MANAGEMENT 24

5.1. WORK PLANNED IN EGEE-III 24

6. AUTHORIZATION 25

6.1. INTRODUCTION 25
6.1.1. Work Planned in EGEE-III 26
6.2. THE GLITE AUTHORIZATION SERVICE 26
6.2.1. What is a Policy? 27
6.2.2. Service Goals 27
6.2.3. Authorization Service Architecture 27

7. ISOLATION AND SANDBOXING 29

7.1. SECURING THE HOSTED TO NATIVE INTERFACE 29
7.2. NETWORK ISOLATION 30
7.3. UNRESOLVED ISSUES AND WORK PLANNED IN EGEE-III 30

8. IDENTITY SWITCHING ON THE WORKER NODES 30

9. DATA MANAGEMENT 32

9.1. UNENCRYPTED DATA STORAGE 32
9.1.1. Disk Pool Manager (DPM) and LHC File Catalogue (LFC) 32
9.1.2. The File Transfer Service (FTS) 32
9.1.3. Work Planned in EGEE-III 33
9.2. ENCRYPTED DATA STORAGE 33
9.2.1. Medical Data manager 33

9.2.2. Support and Evolution *33*
10. LOGGING, TRACING AND AUDITING **34**
10.1. WORK PLANNED IN EGEE-III 34
11. CONCLUSION **35**

TABLE OF TABLES

Table 1: Table of references 6
Table 2 Glossary 7
Table 3 Security Architecture Facilities 12
Table 4 Security Architecture Components 16

1. INTRODUCTION

1.1. PURPOSE

The purpose of this document is twofold:

1. It gives an overview of the current gLite security architecture. It only describes briefly the components and refers to more detailed descriptions where they exist.
2. It summarizes current issues and indicates the work planned on the security components within the lifetime of the EGEE-III project.

The audience of this document are the members of the EGEE-III JRA1 activity, Grid system administrators as well as interested Grid users.

The global security architecture of gLite was described four years ago during the first phase of the EGEE project in [R1], revised in [R2] and assessed in [R3] (EGEE-I deliverables DJRA3.1 and DJRA3.4). This document continues the work of these two deliverables and therefore follows their basic structure and terminology. However, the evolution of the security architecture over this period of time did not affect all components equally. Whereas certain areas were already remarkable mature four years ago, others underwent drastic changes in the meantime. Thus, we decided to rely heavily on the original text and adapt it to the current situation wherever possible, i.e. in those areas where the framework presented in [R1, R2, R3] did not alter significantly. Footnotes label these sections clearly. On the other hand, there are several sections that had to be completely rewritten in order to present the current situation.

1.2. DOCUMENT ORGANISATION

This document is organized as follows:

- Section 2 gives the executive summary.
- Section 3 gives an overview of the gLite middleware and its main services. The components of the security architecture are listed.
- Section 4 describes the authentication, which is based on X.509 certificates. Lifetime issues, bootstrapping, revocation and renewal are described. The concepts of anonymity, pseudonymity and privacy within the context of the gLite middleware are discussed.
- Section 5 describes briefly issues relating to the management of the private key by the user.
- Section 6 gives an overview of the authorization. This is the area, where the main security effort is concentrated in EGEE-III. We first summarize the current situation and then present the planned work for the gLite authorization service.
- Section 7 presents the concepts of sandboxing and network isolation, which protect the local site infrastructure.
- Section 8 explains pilot jobs and the need for identity switching on the worker node, which is enabled by the stop-gap measure of developing a site central authorization service.
- Section 9 gives an overview of the data management security concepts, most notably of the encrypted storage.
- Section 10 describes the issues related to logging, tracing and auditing.
- Section 11 describes the conclusions and closes the document.

1.3. APPLICATION AREA

This document applies to the implementation of the security architecture of the gLite middleware (version 3.1) within the scope of the EGEE-III project.

1.4. REFERENCES

Table 1: Table of references

R1	Global Security Architecture (EGEE-I EU Deliverable DJRA3.1) https://edms.cern.ch/document/487004/
R2	Global Security Architecture, rev1 (EGEE-I EU Deliverable DJRA3.3) https://edms.cern.ch/document/602183/1.3
R3	Security Architecture Assessment (EGEE-I EU Deliverable DJRA3.4) https://edms.cern.ch/document/686044/
R4	Shibboleth Interoperability through dedicated SICS (EGEE-II EU Deliverable MJRA1.4) https://edms.cern.ch/document/770102/1
R5	Shibboleth Interoperability with Attribute Retrieval through VOMS (EGEE-II EU Deliverable MJRA1.5) https://edms.cern.ch/document/807849/1
R6	Grid Components Reengineering Workplan (EGEE-II EU Deliverable MJRA1.3) https://edms.cern.ch/document/756544/2
R7	R. Housley et al. RFC3280: Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. http://www.ietf.org/rfc/rfc3280.txt
R8	S. Tuecke et al. RFC3820: Internet X.509 Public Key Infrastructure (PKI) Proxy Certificate Profile. http://www.ietf.org/rfc/rfc3820.txt.
R9	J. Hahkala, H.Mikkonen, M.Silander, J.White Requirements and Initial Design of a Grid Pseudonymity System; Proceedings of the 2008 High Performance Computing and Simulation Conference (HPCS 2008), Nicosia, Cyprus, 3-6 June 2008.
R10	Authentication and Authorization Infrastructure (AAI) in a nutshell. http://switch.ch/aai/support/documents/#flyer
R11	M. Myers et al. RFC2560: X.509 Internet Public Key Infrastructure Online Certificate Status Protocol (OCSP). http://www.ietf.org/rfc/rfc2560.txt
R12	Alfieri R. et al. VOMS, an Authorization System for Virtual Organizations. In *Grid Computing, First European Across Grids Conference*, 2004
R13	S. Farrell and R. Housley. RFC3281: An Internet Attribute Certificate Profile for Authorization. http://www.ietf.org/rfc/rfc3281.txt.
R14	Overview of gLite authorization mechanisms. EGEE-II MJRA1.7 https://edms.cern.ch/document/887174/1
R15	Globus Workspace project: http://workspace.globus.org/
R16	Grid Security Tracing and Logging Policy https://edms.cern.ch/document/428037
R17	MyProxy Credential Management Service: http://grid.ncsa.uiuc.edu/myproxy/
R18	Privilege Project: http://www.fnal.gov/docs/products/voprivilege/
R19	Hydra: https://twiki.cern.ch/twiki/bin/view/EGEE/DMEDS

1.5. DOCUMENT AMENDMENT PROCEDURE

Amendments, comments and suggestions should be sent to the authors. The procedures documented in the EGEE "Document Management Procedure" will be followed: http://project-EGEE-III-na1-qa.web.cern.ch/project-EGEE-III-NA1-QA/EGEE-III/Procedures/DocManagmtProcedure/DocMngmt.htm.

1.6. TERMINOLOGY

This subsection provides the definitions of terms, acronyms, and abbreviations required to properly interpret this document. A complete project glossary is provided in the EGEE glossary http://egee-technical.web.cern.ch/egee-technical/documents/glossary.htm.

Glossary

Table 2 Glossary

AA	Attribute Authority. See http://www.ietf.org/rfc/rfc3820.txt
AAI	Authentication and Authorization Infrastructure
AC	Attribute Certificate: A special case of an X.509 certificate. See http://www.ietf.org/rfc/rfc3281.txt for details.
ACL	Access Control List: List of rules regulating access to a resource or entity
AP	Authentication Profile
BDII	Berkeley Database Information Index: An LDAP-based information system, where information about the current state of the Grid is stored.
CA	Certificate Authority: An internal entity or trusted third party that issues, signs, revokes and manages digital certificates.
Certificate	Information issued by a trusted party. Used to identify an individual or system.
CE	Computing Element: a Grid-enabled computing resource
Credentials	Evidence asserting the user's right to access certain systems (e.g. username, password, etc)
CRL	Certificate Revocation List
CSR	Certificate Signing Request
CREAM	Computing Resource Execution and Management: The new gLite CE framework developed by INFN.
DN	Distinguished Name to uniquely denote a user or entity. The DN is typically used in directories like X.500 and LDAP, but also within X.509 certificates.
DPM	Disk Pool Manager
EGEE	Enabling Grids for E-sciencE: EU funded Grid project
End Entity	System (individual, host, service) that receives a certificate expressing its identity
EES	Execution Environment Service: Service internal to the new authorization system, which is being implemented within EGEE-III.

EUGRIDPMA	International organization to coordinate the trust fabric for e-Science Grid authentication in Europe. See http://www.eugridpma.org for details.
Federated Identity	Management and use of identity information across security domains, e.g. between members of a federation. Federated identity inevitably deals with issues like liability, security, privacy and trust.
FQAN	Fully Qualified Attribute Name: A string containing VOMS group and (optionally) role information.
FTS	File Transfer Service: A service, which allows to transfer a collection of files between a source and a destination.
GA	Generic Attribute: VOMS attributes in the format of key-value pairs.
GACL	Grid Access Control List: a Grid specific ACL
GID	Group IDentifier in the Unix operating system
gLite	Middleware stack developed by the EGEE project
Globus	A Grid middleware stack. See http://www.globus.org for details.
GLUE	Grid Laboratory Uniform Environment: An abstract information model for the Grid. See http://forge.ogf.org/sf/projects/glue-wg for details.
GSI	Grid Security Infrastructure: the PKI infrastructure that has been customized and extended for the Grid by the Globus project.
IGTF	International Grid Trust Federation: Body with the goal to harmonize and synchronize PMAs policies to establish and maintain global trust relationships in e-Science. See http://www.igtf.org for details.
Internet2	A consortium, led by the research and education community, which promotes advanced networking in the USA.
ITU	International Telecommunication Union: International organization established to standardize and regulate international radio and telecommunication.
ITU-T	International Telecommunication Standardization Sector
LCAS	Local Centre Authorization Service: An authorization framework, which authorizes Grid users based on their X.509 credential.
LCMAPS	Local Centre MAPing Service: A framework, which maps Grid credentials to local accounts on a host attached to the Grid.
LFC	LHC File Catalogue: A file catalogue developed by the LCG project.
LRMS	Local Resource Management System: the batch system behind the Grid CE framework.
MICS	Member Integrated Credential Service: An IGTF profile for issuing (long-lived) X.509 certificates to End Entities based on an identity management system operated by an institution.
MWSG	Middleware Security Group: An informal working group for security architects and knowledgeable security individuals from EGEE, OSG, OMII-Europe and other Grid projects.
MyProxy	A Grid service used for storing and issuing certificates as well as renewing expiring certificates. It was developed by the Globus project. [R17]
NREN	National Research and Education Network

OSCP	Online Certificate Status Protocol
OSCT	Operational Security Coordination Team: Team that is responsible for ensuring the overall security coordination in EGEE.
PAP	Policy Administration Point
PDP	Policy Decision Point
PEP	Policy Enforcement Point
PKI	Public Key Infrastructure: Processes and technologies used to issue and manage digital certificates, enabling third parties to authenticate individual users, services and hosts.
PMA	Policy Management Authority: Body responsible for defining minimum standards for the CP/CPSs of a PKI infrastructures and accrediting against those standards
Principle	Participant in a security operation. It can be a service or a user.
Pseudonymity	Describes a state of disguised identity resulting from the consistent use of another, unknown name.
Resource Provider	Synonym for site.
SAML	Security Assertion Markup Language: XML standard for exchanging authentication and authorization data.
SCAS	Site Central Authorization Service
SE	Storage Element: a Grid enabled storage resource.
Shibboleth	Federated identity management solution from Internet2/MACE (Middleware Architecture Committee for Education). It is the name of the architecture as well as the name of the open source implementation.
Short-lived X.509 certificate	An X.509 certificate with a life time of less than 1 million seconds (approx. 11 days)
Site	Organization having administrative control of resources provided to the Grid. This may be at one physical location of spread across multiple locations.
ssss	Shamir's secret sharing scheme: A scheme for distributing a secret among a group of participants. http://point-at-infinity.org/ssss/
STS	Security Token Service: A Web service that authenticates clients by validating credentials that are presented by a client. It can issue to a client a security token for a successfully authenticated client.
Trusted Third Party (TTP)	Entity, which facilitates interactions between two parties who both trust the third party. They use this trust to secure their own interactions. A CA is an example of a TTP.
UI	User Interface: host from where the user interacts with the Grid software in the gLite middleware environment.
UID	User IDentifier in the Unix operating system
VO	Virtual Organization
VOMS	VO Management Service (may sometimes also refer to the server hosting the service.)
WMS	Workload Management System: a central broker for job submission and management

WN	Worker Node: the entity where jobs get executed
X.509	A standard for public key infrastructures. It defines among other things standard formats for certificates. See http://www.ietf.org/rfc/rfc2459.txt for details.
XACML	eXtensible Access Control Markup Language - a declarative access control policy language implemented in XML

2. EXECUTIVE SUMMARY

This document gives an overview of the gLite security architecture and its planned evolution over the lifetime of the EGEE-III project. It points out that a consistent security model requires that the security "building blocks" (called facilities) must span across Grid services in a consistent manner. The following groups of building blocks are identified and described: authentication, authorization, logging and auditing, key management, delegation, isolation and sandboxing.

Today, that consistency has not yet been reached across all Grid services. Whereas certain facilities, (e.g. authentication) are being treated consistently, others (e.g. authorization and logging) are implemented in an at best non-uniform, at worst inconsistent manner.

The goals of the security architecture work within EGEE-III aims to address the following issues:

1. Development of a new authorization service, which allows applying a consistent and accurate authorization policy across the Grid.
2. Support for identity switching at the worker node.
3. Improvements in the renewal of short-lived proxy certificates, which will allow reducing their lifetimes.
4. Targeted improvements and stop-gap measures in order to ease the handling security incident.

Last, but not least, it should be stressed that the implementation of the security architecture of a middleware stack such as gLite is an ongoing evolutionary process, which requires increased collaboration between the developers of the security software, Grid deployment and operational teams as well as site security officers, typically in an environment of limited resources.

3. OVERVIEW OF THE GLITE SECURITY ARCHITECTURE

This document captures, at a high level, the global security architecture of the gLite middleware stack and outlines its planned development within the EGEE-III project.

The gLite security architecture evolved over the course of several EU funded projects and parallel ongoing efforts, such as the Open Grid Forum (OGF) or the Globus Alliance. Thus, the current system is the result of a continuous development with input from general security concepts of the software development on one hand and the actual experience of deploying and securely operating the EGEE infrastructure on the other hand. It should also be kept in mind, that the security infrastructure must not only satisfy the security requirements, but it must also be practical and easy to use by site administrators, security personnel and users.

In this chapter we present the main building blocks of the gLite security architecture. We start with the definition of the term “security architecture” and define our usage of the terms “trust”, “authentication” and “authorization”. We then continue with a description of the virtual organization and its relation to sites before enumerating the main security building blocks. In doing so, we try to strike a balance between presenting high-level security concepts and its actual implementation in gLite.

3.1. DEFINITION OF A SECURITY ARCHITECTURE

We define the term “security architecture”[1] as “a set of features and services that tackles a set of security requirements and can handle a set of use cases”.

We also note that the term service is used here in a broader sense than (web) service, which is the common jargon in other documents, and also includes infrastructure and user-driven services such as certification, auditing and incident response procedures.

According to [R2], the security architecture is composed of building blocks, called facilities. Each facility can be viewed as an atomic piece of functionality that facilitates some security requirement. The facilities should be factored orthogonally against each other, such that their use is optional, and keeping the number of interdependencies to a minimum, allowing for individual advancements of respective technologies and mechanisms used in the implementation of our architecture.

The facilities, along with a short description, are listed in table 1. The definitions were taken from [R2]

Table 3 Security Architecture Facilities

Functionality	**Facility**	**Short Description**
Logging and Auditing	Logging	A common infrastructure for the recording of system events for tracking, accountability, and auditing purposes.
	Auditing	The auditing system uses information recorded about system and user activity for the purposes of accountability and security assurance.
Authentication	Identity credentials and trust infrastructure	The identification of users, agents, hosts, and services, when they interact with each other.

[1] As defined in [R1].

	Short-lived credential services	Short-lived credential services (SLCS) leverage the organizational or federated authentication infrastructure and issue (short-lived) Grid credentials to their users.
	Enforcing validity constraints	Additional validation tests are required to assess that credentials (in particular proxy certificates) are adhering to established operational policies.
	Revocation	The process of invalidating a credential, and the secure distributed propagation of such status change information.
	Certificate renewal	The automated, yet controlled and managed, renewal of short-lived credentials and authorization assertions.
	Anonymity, privacy, pseudonymity	The controlled protection of user identities and data, and escrow of the same for management and authorities.
	Bootstrapping authentication	A collective term for the initial security mechanisms that can be used to acquire a Grid credential.
	Credential store	The controlled secure management of user credentials, to permit the enforcement of key hygiene.
Key Management	Key hygiene	Enforcement of proper handling practices of user held credentials.
	Encryption key management	The secure management of cryptographic keys that in turn protect data stored in encrypted form.
Authorization	Authorization services	A collective term for the (centralized) services used to manage access control (typically one per VO)
	Mutual authorization	The process in which both parties authorize each other before engaging in a message exchange, typically to avoid information leakage.
	Authorization framework	A framework that collect and combines policies and information from several sources using pluggable extensions.
	Authorization interfaces to existing systems	An authorization interface to existing and legacy systems, which allows a combined and flexible decision-making process by taking into account information, assertions and policies from a variety of authorities.
Delegation	Delegation	The capability of transferring rights and privileges to allow for a principal (e.g. an application or a user) to act on your behalf.
Sandboxing	Securing the hosted to native interface	Isolating the (user-provided) applications from each other and from the local system as much as possible, while preserving the appearance of transparent access to shared remote resources.

	Network isolation	Dynamically adapting firewall policy to enforce strict rules yet being able to obey the connectivity needs of (some) users and applications.

3.2. TRUST, AUTHENTICATION AND AUTHORIZATION: A TERMINOLOGY

The concepts of trust, authentication and authorization are often poorly defined or confused. We attempt in this section a short description of these terms with a focus on X.509 credentials.

Under *Trust* we understand the process of ensuring that the issuer of a credential, and the credential itself, is trustworthy.

Under *Authentication* we understand the process of ensuring a credential is valid and belongs to the individual that presents it.

Under *Authorization* we understand the process of checking that a person has the rights to perform an operation. Authorization can be issued based on several criteria, such as, for example, the identity of the person or attributes provided about the person by a trusted third party.

Note that trust and authentication are often meshed together, but as a matter of fact they should be treated separately. For example, you can establish trust without authentication. Authentication and authorization are often (wrongly) understood as being just two sides of the coin. Today, it is generally accepted that a good design clearly separates authentication and authorization.

Trust evaluation in a PKI infrastructure may include:

- Ensuring the certificate is not expired.
- Ensuring the signature of the certificate still matches.
- Ensuring the certificate is issued by a trusted CA (this includes checking the trust chain).

Authentication in a PKI infrastructure may include:

- Ensuring the user has the private key to the certificate that he provides. (Note: This does not mean that the holder of the private key is the intended recipient of the private key).
- Ensuring the user continues to provide the correct SSL session key.
- Performing a trust evaluation on a certificate.

Authorization in a Grid environment may include:

- Ensuring that the DN is present in a configuration file that lists all authorized users.
- Ensuring that an attribute[2] listed in the extension of the certificate has a certain value in order to perform an operation, e.g installing software.

[2] The user's proxy certificate may contain a so-called attribute certificate, which contains attributes describing the user as member in a VO (see section 3.3).

3.3. VIRTUAL ORGANIZATIONS, SITES AND COMMON SERVICES

In Grid environments users are organized in so called *virtual organizations* (VO). They allow the management of users across different institutions without the need to observe intra-institutional management structures and policies.

Sites[3] on the other hand correspond to local installations of computing resources (clusters, storage, etc). Agreements are made between the individual sites and VOs that give the members of a VO access to the resources of the sites. In order to do so, sites must install the services of the chosen middleware. It is important to note that the support of VOs by sites does not mean that the site hands over some of its autonomy, particularly in security issues, to VOs. On the contrary, the local site autonomy must be preserved and respected by the VO and its members at all times.

Besides VO and sites, there are also a set of *common services* that act as a glue between VOs and sites. They can be considered as a part of the underlying Grid infrastructure and are typically operated by some of the larger sites on behalf of the infrastructure. Examples are information services, credential stores, management services for VOs etc.

Figure 1 shows the high level picture of the gLite components divided into the elements of VOs, sites and common services. The gLite middleware consists of the following software components:

- User Interface UI: Component through which the user interacts with the Grid (submitting of jobs, querying component status, access to data stored on the Grid, etc). The user needs to possess a X.509 certificate in order to interact with the Grid services, because the gLite security model is currently entirely PKI-based.
- CA: A certificate authority provides the X.509 credential to the user. The CAs are coordinated through the international Grid trust federation (IGTF) and are outside the purview of the gLite security architecture. Note that CAs also issue host or service certificates to identify hosts and services.
- VO services: Users are permitted to access the Grid due to their VO membership. Currently; there is only one supported VO service, the Virtual Organization Management Service VOMS.
- Common services: These are services that span the Grid and may serve one or more VOs at the same time. Examples are the information system, file catalogues and resource brokers.
- Credential services: This is a special class of services, which act either
  - As an attribute authority (AA)[4],
  - As proxy certificate store and renewal service (MyProxy [R17])
  - as a short-lived credential service (SLCS[5]). The only supported SLCS currently is the gLite SLCS service (see [R4]) with a new service STS under development.

[3] We accepted the terminology of the Joint Security Policy Group in this document and we refer to the glossary for a definition of the term “site”.

[4] VOMS is the AA in use. VASH is a service, which allow to transfer attributes from another AA into VOMS (see [R5]).

[5] Originally (i.e. in [R1]), the term SICS stood for site integrated credential store. It was later used interchangeably with the term SLCS [R4], which we use in this document for this service. This accommodates the fact that the concept of credential management and translation services has been changing over the last few years.

- Site-specific software components: These are the Grid services that the individual sites operate. They comprise of Compute Elements (CE) and Storage Element (SE) services. The user can either access these services directly or through resource brokers and/or file catalogues.

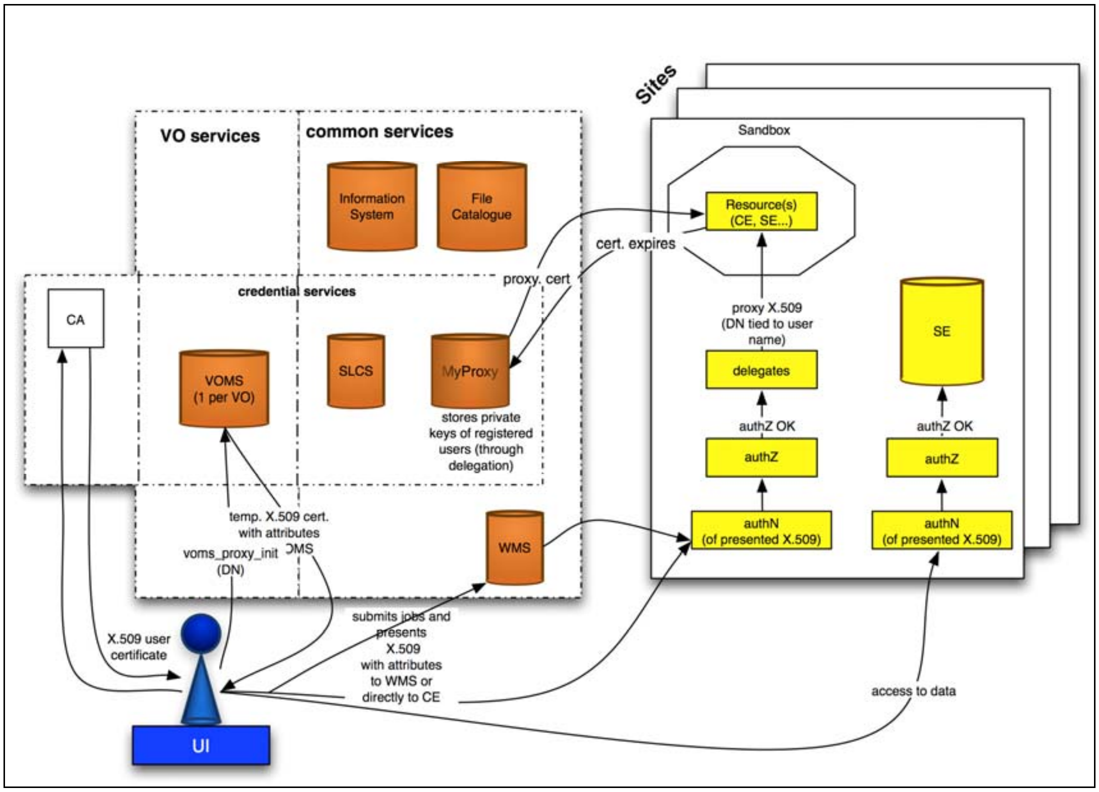

Figure 1 Overview of the gLite architecture

Table 4 lists the gLite supported software components [R6]. Each of these components must support the needed security features as given by the above-mentioned facilities.

Table 4 Security Architecture Components

Class of Service	**Software Component**	**gLite supported instance**
VO services	VO management system	Virtual Organization Membership Service (VOMS)
	Attribute transfer interface between VOs and AAIs	Voms Attributes for SHibbolth service (VASH)
Common services	Information system	Berkeley Database Information Index (BDII)
	File catalogues	LCG File Catalogue (LFC)
	Store for proxy certificates	MyProxy

	Short-lived (site-integrated) credential services	Short-Lived Credential Service (SLCS)[6], Security Token Service (STS)
	Resource broker	Workload Management Service (WMS)
	File Transfer Service	File Transfer Service (FTS)
Site-specific services	Compute Element (CE)	lcg-CE, Compute Resource Execution and Management Service (CREAM)
	Storage Element (SE)	Disk Pool Manager (DPM), Hydra
	Authorization service	Site Central Authorization Service (SCAS), gLite authorization service

4. AUTHENTICATION

Authentication[7] (AuthN) is concerned with identifying entities (users, agents, and services) when establishing a context for message exchange between principles. One of the key aims for Grid authentication is enabling single sign-on for the user – using a single identity credential with “universal” value across many different infrastructures, different communities and virtual organizations and multiple applications. As such, attention must be paid to the fact that the same identity is also to be used for other purposes: accessing non-Grid resources such as networks and web resources; or in interaction with government and administration.

The authentication model for EGEE uses the concept of *trusted third parties* (TTPs): entities that are not related to any *relying party* except through a trust relationship. Underlying that trust relationship is the digital signature of the TTP, based on conventional asymmetric cryptography. The TTP will bind the cryptographic key pair to one or more identifiers that represent the entity. Although theoretically a single TTP could service the entire community, in practice a mesh of TTPs exist. The mesh defines a common authentication domain, by grouping the resources, users, and services that agree to use a common set of TTPs for authentication. The authentication domain, however, does neither imply common rights-of-access nor does it constitute an “infrastructure” of sorts. Although the EGEE project assumes a set of TTPs is used, it does not suggest that all entities accept this set. This introduces an additional failure mode that higher-level services should cover anticipate and handle.

The strength of an authentication credential issued by a TTP is dependent on three things:

- The trust on the TTP, in particular the TTP’s operations, procedures and general conduct.

[6] Note that the SLCS is also, at the same time, a CA as it issues short-lived certificates

[7] Large sections of this chapter have been taken from [R1] and adapted to the current situation. This reflects the mature status of the authentication in the security architecture. See also section 3.2 for a descriptions of the terms trust and authentication.

- The quality of the original identity vetting. No amount of technology can overcome weak identity checking at the source. The use of appropriately qualified authorities in the credential issuing process is important. As such, the use of government-issued credentials (like PKI-enabled passports or photo-IDs) is encouraged.
- The security of the private data needed to prove possession of the credential. This is further discussed in Section 5.

The International Grid Trust Federation (IGTF) and its European constituent EuGridPMA have been established and expanded over the course of the EGEE project. They are the orchestration bodies for the TTPs and define Authentication Profiles (AP) for identity provisioning.

We will continue to base our authentication on the work of these PMAs as well as actively contribute to their work. The advantages of such a standardized approach across the many partners of EGEE as well as across several Grid middlewares and Grid infrastructures cannot be overemphasized.

4.1. IDENTITY CREDENTIAL FORMATS

In accordance with the standards as defined by IGTF, X.509v3 public key certificates [R7] are used to express identity assertions. These certificates are issued by Certification Authorities (CA), which are accredited by IGTF. Different types of CAs exist (see below), as well as different means of delivering certificates to end-entities. Whatever the delivery mechanism or operational mode of the CA, the authentication is based on at least the distinguished name (DN) of the subject contained therein.

While the distinguished name may contain information about the user such as name, organizational affiliation and email address, we do not make any assumption on these fields. The distinguished name is only considered as a unique identifier, which bears no semantics. In order to facilitate a single-sign on for Grid resources as part of the authorization process, our security architecture also supports *proxy certificates* as defined in [R8]. Proxy certificates are normal X.509 identity certificates, but equipped with an extension that ensures their rejection by applications that do not support proxy certificates.

Two different types of certificates can be distinguished based on their lifetime:

1. Long-lived (typically one year) certificates, which are issued according to the “classic” AP or the “Member Integrated Credential Service” (MICS) AP. A MICS certificate is issued by an automated system based on a pre-existing identity management system maintained by an organization or federation.
2. Short-lived (i.e. less than 1 mio seconds, approx. 11 days) certificates, which are issued according to the Short-lived Credential Service (SLCS) AP.

It should be noted that most CA accord to the “classic” AP, with MICS and SLCS based CAs only gaining attention recently.

Over the past few years, security architectures, which do not use X.509 certificates as identity credentials, have gained significant interest. Authentication and Authorization Infrastructures[8] (AAI) based on the concept of federated identity have been established in several European countries, often coordinated by National Research and Education Networks (NRENs). They typically comprise tens or even hundreds of thousands of users These AAIs use SAML assertions to express authentication

[8] There exists several implementations of AAIs. Shibboleth of Internet2 has gained the widest acceptance in Europe.

information and user attribute values. Whereas we do not foresee replacing PKI as the basis of the security architecture over the next years, we do expect interoperability issues to gain in importance. The key concept to implement is the generic support of security tokens without requiring a specific token format. Security Token Services (STS) can then be used to exchange a given token into one of a different format.

4.2. SHORT-LIVED CREDENTIAL SERVICES

Currently, two types of SLCS services can be distinguished: SLCS tied to an organizational identity management system (such as the Kerberos-enabled CA (kCA) operated at Fermilab) and SLCS tied to an identity management federation (such as the SWITCHaai Shibboleth-based authentication and authorization infrastructure of Switzerland (see [R10])).

Whereas we expect an expanded use and accreditation of federation-based SLCS, it is rather unlikely that many organizational SLCS[9] will be accredited. We see two reasons for this trend:

1. Federations (at least in Europe) are typically coordinated by national organizations, often the National Research and Education Networks (NRENs). Their accreditation scales well with the national CAs as propagated by IGTF. Therefore it is natural to assume that SLCS will primarily be deployed on a national level.
2. The accreditation of organizational SLCS on the other hand does not scale on an international level. We expect the services to rather be deployed in campus Grids, where an IGTF accreditation may not be needed.

4.3. BOOTSTRAPPING AUTHENTICATION

The process of bootstrapping authentication is well established for services and users. For services the administrator requests a service certificate from an accredited CA, which he installs on the host. Typically, the Grid service must then be configured to use it.

For users, it consists of obtaining a certificate from an accredited CA. Before accessing Grid services, the user must create[10] a proxy certificate, which contains a VO specific extension, the VOMS Attribute Certificate (AC). The AC contains the set of attributes that the user has within the context of the VO. (See section 6 and [R14] on the usage of these attributes in authorization decisions.)

4.4. ENFORCING VALIDITY CONSTRAINTS

Proxy certificates are typically stored with a weaker protection level (stored in clear text and safeguarded only by local file system privileges). As a consequence, security policy often declares that proxy certificates should not be trusted if issued with a longer validity: A lifetime of 24 hours is normally requested, with longer-lived proxy certificates mandated to be stored in a keystore such as MyProxy [R17]. Currently, this is an unresolved issue that should be resolved (see also 4.10).

[9] By organizational SLCS we mean a SLCS CA operated by an institution. Examples are universities that issue certificates based on their local identity management system (e.g. Kerberos).

[10] This process is described in detail in [R14]

4.5. REVOCATION

There are good reasons why the binding between the public part of a key pair and an identifier, or a set of identifiers, should be revoked. These include the compromise of the associated private key or invalidation of one or more identifiers in the binding. The longer the key-to-identifier binding is considered valid, the higher the probability that such a binding will become invalid.

The TTP is responsible for revoking credentials it has issued, but it is up to the relying parties (both services and requesters in case of mutual authentication) to ensure that this revocation information is consulted. The basic revocation information is distributed as a Certificate Revocation List (CRL) [R7]. These CRLs act upon identity certificates issued by traditional CAs only and are as of today[11] not applicable to SLCS CA.

Timely identity revocation is needed to prevent exploitation of credentials that have been compromised. The allowed response time as specified by sites is in the order of 10-60 minutes. However, this time cannot be achieved through the periodic distribution of CRLs to all parties at a pan-European (or larger) scale.

Therefore, any software component that performs certificate validation must be able to check the validity of the credentials in real-time. The Online Certificate Status Protocol [R11] (OCSP) is a widely used protocol that facilitates the validation or credentials in real-time. Such services have been proposed in the past. However, as of now no such service has been deployed such that CRLs remain for the foreseeable future the main revocation mechanism.

4.6. CERTIFICATE RENEWAL

Certificates are equipped with a validity timestamp and, as such, they expire and need to be renewed. There are several techniques that a CA may use to facilitate a trusted remote credential renewal, which we do not cover here. For instance, the CA may choose to: a) simply issue a new certificate to the existing user-held key pair; b) require the user to generate a new key pair; c) countersign the new key with the old and so on.

4.7. DELEGATION

It is often the case that Grid users need to delegate some subset of their privileges to another (dynamically created) entity on relatively short notice and for a brief amount of time. For example, a user needs to move a data set from a source to a destination, where he needs to use it in a computation. Therefore he wants to grant the necessary rights to access the dataset and storage to a file transfer service. The latter then acts on behalf of the user. Since such actions may be difficult to predict, the need to arrange delegation ahead of time is prohibitive.

In gLite, the delegation mechanism is based on the creation and propagation of proxy certificates across the Grid infrastructure. Thereby, a principle (user or Grid service) who needs a delegated credential, creates a new key pair and sends a Certificate Signing Request (CSR) back to the requestor, which signs it using the private key of his (proxy) certificate. This step is repeated across all elements involved in the chain of actions on the Grid. The proxy certificate contains not only the public but also the private key and is stored on the file system. It is not protected by a passphrase, but only by file access restrictions. Anybody gaining read access to the proxy can thus impersonate the user.

[11] This issue is currently being reconsidered by IGTF.

An important security aspect in regards to delegation is the principle of least privilege: you only want to delegate as much privileges as is necessary. This is hard to accomplish in reality and is currently not supported with the exception of the "limited proxy certificate". These are proxies that inherit all rights of its parent except job submission.

4.8. RENEWAL OF PROXY CERTIFICATES

The renewal of proxy certificates must be considered separately. It is necessary in order to support long running jobs on the Grid.

The renewal procedure starts with delegating (see section 4.7 and Figure 2) a proxy of the user's certificate into an online keystore such as MyProxy [R17]. The delegation procedure will ensure that a newly created public and private key pair has cryptographically become a part of the certificate chain, which will be stored in a MyProxy service. This stored delegated proxy must be of a longer-lived type, which means that it should have a maximum lifetime in the order of two weeks. The user then typically submits jobs using a short-lived proxy with a validity length in the order of one day.

When the short-lived proxy is about to expire a renewal service[12] or daemon must detect this and prior to expiration it tries to request a new delegation from the previously used online keystore. The renewal service must authenticate to the keystore using the almost expired delegated proxy. The keystore will use the stored long-lived delegated proxy to create a new short-lived delegation to trusted renewers. This procedure can continue as long as a valid proxy resides in the keystore. The entire certificate chain needs to be valid at each level in the chain.

In a VOMS-enabled (see section 6) deployed Grid infrastructures such as gLite, a few extra steps have to be taken when renewing the proxy certificate, as not only the proxy certificate but also the VOMS AC must be renewed. This could be taken care of by the keystore itself. However, the currently deployed keystore MyProxy does not offer this functionality. Therefore the renewal service must contact the user's VOMS server and assemble the VOMS extension into the proxy.

[12] This functionality is currently been provided by the renewal service, which is a part of the gLite Workload Management Service (WMS). The FTS service also supports a renewal mechanism for long lasting transfers.

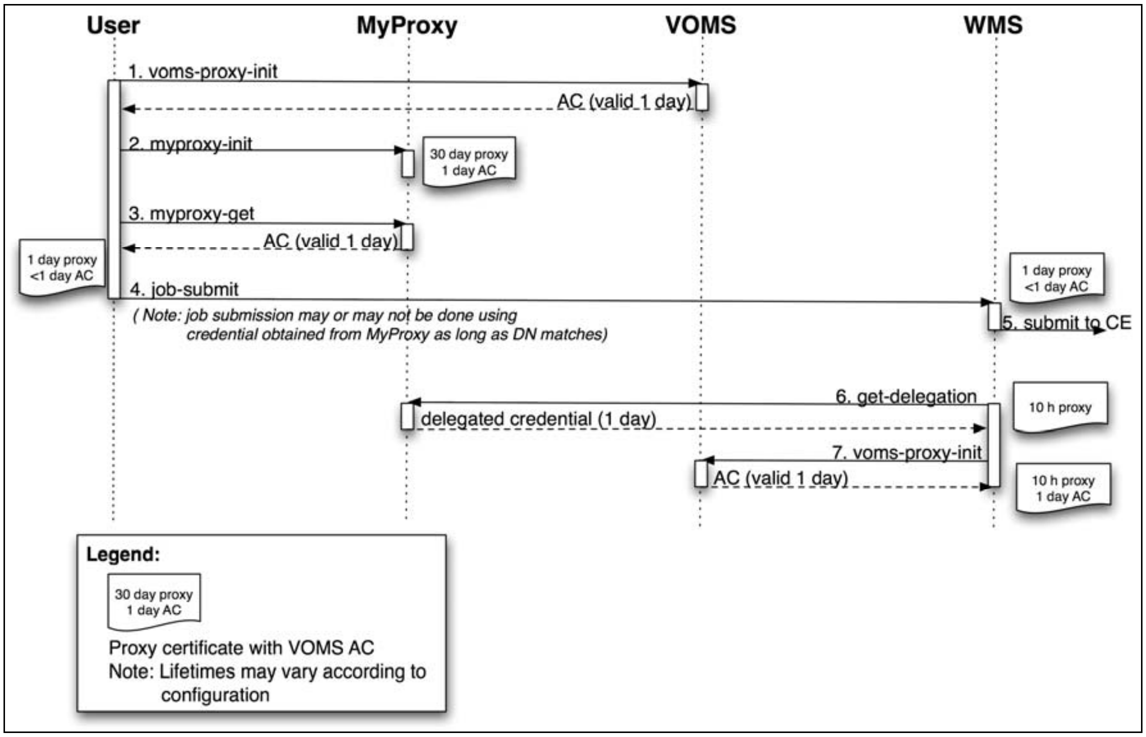

Figure 2 Proxy Renewal Mechanism

4.9. ANONYMITY, PRIVACY, PSEUDONYMITY

Application and user requirements[13] call for anonymous use of the system: an outsider should not be able to deduce a particular user's activities, such as how much of the resources the user consumes or what applications are run. Such *information creep* is of serious concern to applications in areas of highly competitive research, such as biomedicine.

We note that Grids are open-ended distributed systems. As such, protecting ourselves against information creep is prohibitive. Even if all information and message exchanges were made on encrypted and authenticated connections, analyzing the message exchange patterns would still allow an adversary to deduce information on how the system is being used. In fact, true anonymity can only be achieved by all parties sending continuous streams of (bogus) data to all other interacting parties at all time: this is a common tactic in the military and intelligence world.

Likewise, the only way in which you can obtain true privacy is by not sharing information or data at all, which is contradictory to the very nature and basis for Grid computing. In addition, many sites attached to the Grid require in their local site policy the full disclosure of the identity of the user regardless whether this is a local user of the site or a remote user accessing the site through the Grid. Thus, for these requirements, we provide best-effort solutions by the addition of a pseudonymity service and defer on privacy and anonymity.

The pseudonymity service swaps the user's real identity for a pseudonym, thus hiding it from

[13] Note that national as well as EU legislation also set requirements on privacy and anonymity.

immediate exposure in logs and on the network. The pseudonymity service acts in all regards as another TTP, with the addition that it is also trusted to maintain the relationship between the pseudonym and real identity in confidential and secret manner. This trust has to be also kept unless law enforcement or a similar legitimate body requires it as part of, for example, an investigation on malicious use.

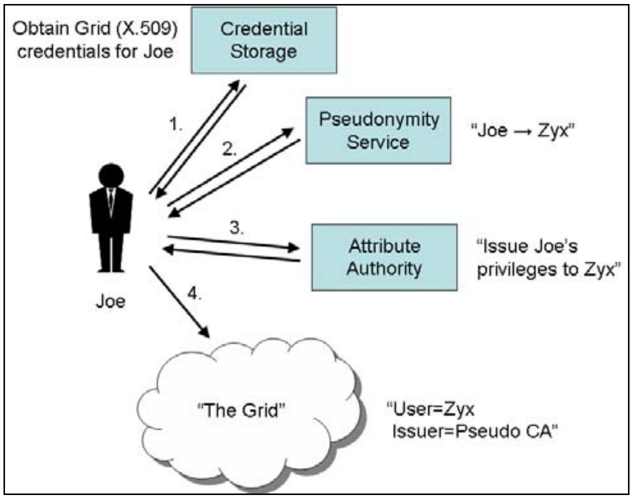
Obtain Grid (X.509) credentials for Joe
Credential Storage
1.
2.
Pseudonymity Service
"Joe → Zyx"
3.
Attribute Authority
"Issue Joe's privileges to Zyx"
Joe
4.
"The Grid"
"User=Zyx Issuer=Pseudo CA"

Figure 3 Use of the pseudonymity service

To acquire a pseudonym identity, a user performs the following steps (see **Figure 3**):

1. The user obtains their normal authentication credential, for example: *Joe*.
2. The user authenticates using *Joe* to the pseudonymity, service, which then issues a one-time identity, *Zyx*, to the user.
3. The user authenticates twice to the attribute authority using *both* the *Joe* and *Zyx* credentials, thus proving possession of them both. The attribute authority can then bind authorization attributes, originally issued to *Joe*, to the one-time identity.
4. The user is now bootstrapped with a pseudonym credential *Zyx* and the necessary access privileges bound to that credential to make use of Grid services.

In the Grid VO model, the user's real identity is revealed to the pseudonymity service (the authentication trust anchor) and the VO service (the authorization trust anchor). We work under the assumption that we do not need to hide the true identity of a user from the other members of the same VO.

The pseudonymity service has been implemented during EGEE-II and shares code with the SLCS service [R9].

4.10. UNRESOLVED ISSUES AND WORK PLANNED IN EGEE-III

Only code maintenance is planned for most components of the authentication infrastructure.

Currently, there are the following non-resolved issues with the validation of proxy certificates:

1. Proxy certificates cannot be revoked. If a compromise were suspected, then the only

countermeasure one can take is to revoke the user's certificate. This is the one reason for requesting a short lifetime for a proxy.

2. The restrictions on the lifetime of a proxy certificates are currently very weak. This is because the proper renewal of the AC within the proxy is not yet deployed in the EGEE infrastructure. Once this is done, the lifetime of proxies can be limited to 24 hours and proxies having a longer life can be rejected. In addition, a review should be done in order to make sure that the relevant Grid services enforce this restriction. It should be stressed, that the handling of the proxy certificates as done today is one of the main deficiencies of the current security architecture.
3. Today, some VOs are using proxy certificates of a much longer lifetime that what is assumed by the security architecture (one or even two weeks long). This is one of the weakest spots of the architecture at this point and must be dealt with at a technical as well as managerial level. With improvements on the proxy renewal under way, it can be expected that the lifetime can be shorten during the lifetime of the project. This requires close coordination with the VOs.

As mentioned above, a VOMS-enabled keystore would simplify the proxy renewal mechanism and lead to a cleaner separation of security-related responsibilities. This task is planned for 2009.

As part of the interoperability work between gLite and Shibboleth a security token service (STS) is being developed, which is based on the WS-Trust standard. EGEE collaborates with Internet2 in this area.

No work is planned on privacy and anonymity. This is clearly a missing functionality given the fact that is required by the current EU legislation.

5. USER KEY MANAGEMENT

The experiences of the past years have shown that the management of private keys by the end user is a significant entry barrier for the wide adoption of Grid technology. PKI technology has turned out to be not easily understood by many users, and the average user often ends up leaving weakly protected copies of this certificate on various untrusted systems.

Several other technologies were investigated during the first and second phase of EGEE, but no final conclusions have been reached [R1, R2].

5.1. WORK PLANNED IN EGEE-III

No dedicated activity is planned in this area in EGEE-III. However, it may very well be that user communities and/or national AAI federations may better be suited to address the issues surrounding the user key management than EGEE itself.

We mention at this point three trends that may remedy this situation in the medium term:

1. Some user communities have found that the entry barrier can be significantly lowered through community specific portals. It can be expected that some of these portals will be using other authentication methods than X.509 certificates, e.g. federated identity.

2. The PMAs have started to investigate how different classes of X.509 certificates can be supported (robot certificates, use of certificate in portal technologies etc).
3. The first SLCS CAs have been accredited, and the first experiences with them have been very positive. They allow the issuance of short-lived credentials that are practically invisible to the end user.

6. AUTHORIZATION

6.1. INTRODUCTION

Authorization (AuthZ) is concerned with allowing or denying access to services based on *policies*. The core problem with authorization in the Grid environment is how to handle the overlay of policies from multiple administrative domains (VO policies, operational procedures, policies of the local sites) and how to combine them.

Early Grid deployments based authorization on the identity of the user (e.g. the Distinguished Name of the certificate). Whereas this approach is very simple and intuitive, it doesn't scale and doesn't support more sophisticated authorization policies.

Attribute-based authorization overcomes the limitations of identity-based authorization and is the authorization mechanism that has been chosen for gLite. It requires two kinds of components:

1. Attribute Authorities (AA), which associate a user with a set of attributes in a trusted manner to a relying party by way of digitally signed assertions. Note, that several attribute authorities in a Grid deployment may assert attributes. gLite currently only supports one AA - the VO Membership Service (VOMS) [R12]. VOMS assigns two types of attributes[14] to the user: memberships in an arbitrary number of hierarchical groups, and roles that the user may or may not assert on a case-by-case basis. VOMS issues X.509 attribute certificates (AC) [R13], which contain a list of strings, so-called Fully Qualified Attribute Names (FQAN), corresponding to the role and group membership associated with a particular user. These ACs are then embedded in the user's proxy certificate and are thus available to the Grid resources at authorization time. A detailed description of the group and role mechanism and the role of AC can be found in [R14].
2. The relying party (the resource) evaluates the attribute assertions and includes them as "evidence" added to a *context*, which in turn is used when evaluating an access request against local policy. As an example, a CE evaluates the FQANs and authorizes the user based on their values (see [R14] for details).

Whereas the gLite middleware offers with the VOMS service a well-established Attribute Authority, no such standard service exists for the evaluation and enforcement of authorization decisions. Today, the authorization is done in the different gLite services through different software modules such as LCAS [R14], GACL and file-level ACLs.

[14] VOMS also supports a third type of attributes: the Generic Attributes (GA). They are arbitrary key-value pairs that the VOMS administrator defines. Thus, they are VO-specific and have no meaning for the generic infrastructure. Therefore, they are not used by gLite services nor are policies for their usage defined. We do not consider them in this document any further.

This situation has the following consequences:

1. There is no standard in gLite to express authorization policies for Grid components.
2. A given resource has no way to advertise its authorization policies to clients in a consistent manner[15].
3. There is no standard way to ban users at the level of a resource, site, VO or globally.
4. It is not possible to use different attributes for different authorization requests. The authorization attributes depend solely on the order, in which they are listed in the VOMS AC. In particular, the first FQAN is taken as primary FQAN for the job handling as well as data management services. Thus, it is not possible to process a job with a given primary FQAN and store the output data of the job using another primary FQAN (see [R14] for details).
5. The logging of the various authorization decisions is distributed within a given site. Therefore it is hard for a site administrator to investigate and understand incidents.

6.1.1. Work Planned in EGEE-III

The shortcomings of the current authorization framework have been investigated in EGEE-II. We refer to [R14] for a description of the different authorization mechanisms in the job management and data management of gLite. The improvement of the current situation is one of the main focuses of JRA1 in EGEE-III and comprises work in four areas:

1. Specific improvements in various Grid services such as a general library to evaluate pattern matching algorithms of required attributes against the attributes of the user’s proxy certificate.
2. Development of the Site Central Authorization Service (SCAS), which is a short-term solution that allows distributed processes at a site to authorize users and obtain a mapping to a local account. It is tightly coupled to the fact that identity switch must be supported on the WN (see section 8).
3. Development of the gLite Authorization Service; a new service, which allows expressing and publishing authorization policies in a consistent manner. It is described in section 6.2. Note that the SCAS service will become one of the components of this new service.
4. Modification of the job submission such that different primary FQANs are being used by the job handling and for the data management software.

6.2. THE GLITE AUTHORIZATION SERVICE

As described above, gLite currently lacks a consistent and accurate application of authorization policy across the Grid. This is the main focus of the gLite authorization service, which is under development in EGEE-III.

As most, but not all, of the data available on the grid is public, many individuals think of authorization only in terms of access control to services in general or service operations in specific. The current scope of the authorization service is thus initially limited to this problem space and does not specifically attempt to address other problems spaces (e.g. data access authorization). However, every effort has been made to ensure this service is applicable in other areas as well.

[15] E.g. Computing Elements publish their access policies to shares through the information system as a mapping between so-called VOViews and accepted FQANs. See [R14] for details.

6.2.1. What is a Policy?

The core concept of the authorization service is that of a policy. Policies, however, are nothing new in gLite. The gridmap files currently used are an example of a very simple policy. However some deployers may wish to communicate more robust policies, for example:

- Do not allow jobs from anyone on the site's banned list.
- Do not allow jobs from anyone on a Grid-wide banned list[16].
- Allow only jobs that execute programs present on the site's application white list.
- Allow individuals from the site to submit work to site resources and, from 8.00 – 17.00, give their work a priority of “highest”.
- Allow any individual to submit work to site resources. From 8.00 – 17.00 given their work a priority of “low” and a priority of “normal” any other time.
- All jobs started by a pilot job[17] must use a credential associated with the same VO that from which the pilot job originated.

As can be seen from the examples, policies may pull in information from other sources (e.g. a Grid-wide banned list), be based on submitter, environment (e.g. time), and job (e.g. program to execute) properties, and require certain responsibilities (e.g. giving work a specific priority) for an affirmative decision to be valid.

6.2.2. Service Goals

The primary goal of the authorization service is to accurately and consistently apply authorization policies across Grid resources. This process starts by allowing authoritative individuals to write and maintain their policies thus increasing the chance that the policy will be accurate. These policies are then automatically distributed across the Grid in a relatively short period of time (i.e. minutes to hours) ensuring that all services are eventually operating with a consistent, up-to-date, set of policies. Finally, all services use the same policy evaluation engine and so always end up with consistent results.

At a more technical level the service also attempts to provide very good tools and information to deployers and troubleshooting staff. For example, robust audit logs that may be used to track users and simple command line tools for adding new policies or determining an effective, point-in-time policy. Additionally, the components of the service are designed to be very scalable and highly resistant to failure, when deployed in a network configuration. The authorization service client is kept as small and simple as possible to allow for easier deployment (i.e. to avoid library dependency conflicts) and lower the barrier of entry in creating clients in other languages. Lastly, a great deal of effort is invested to make sure that components can be deployed in numerous models in order to allow a balancing of factors like complexity, latency, and response time.

6.2.3. Authorization Service Architecture

The authorization service is made up of four individuals components (see Figure 4).

[16] E.g. the Operational Site Coordination Team (OSCT) of EGEE could operate such a Grid-wide banning list.

[17] Pilot jobs are explained in section 8.

The Policy Administration Point (PAP) is the repository for policies. Generally speaking, each organization that contains an author of policies will run a PAP. The Policy Decision Point (PDP) is the policy evaluation engine. The Policy Enforcement Point (PEP) acts as the client to the authorization service. It sends authorization decision requests to the PDP and provides the resulting information back to the invoking code (e.g. CREAM, glexec). The Execution Environment Service (EES) works with the PDP to provide the constraints, or context, under which an authorization decision is valid (e.g. run the job under this UID/GID).

The communication between components is performed using the eXtensible Access Control Markup Language (XACML) profile for the Security Assertion Markup Language (SAML). EGEE, in cooperation with the OSG and Privilege Project [R18], has created a profile of this specification that allows the authorization components from gLite, Globus, and the Privilege project to interoperate (as already mentioned in section 8). The policies exchanged between PAPs are expressed in the XACML policy language.

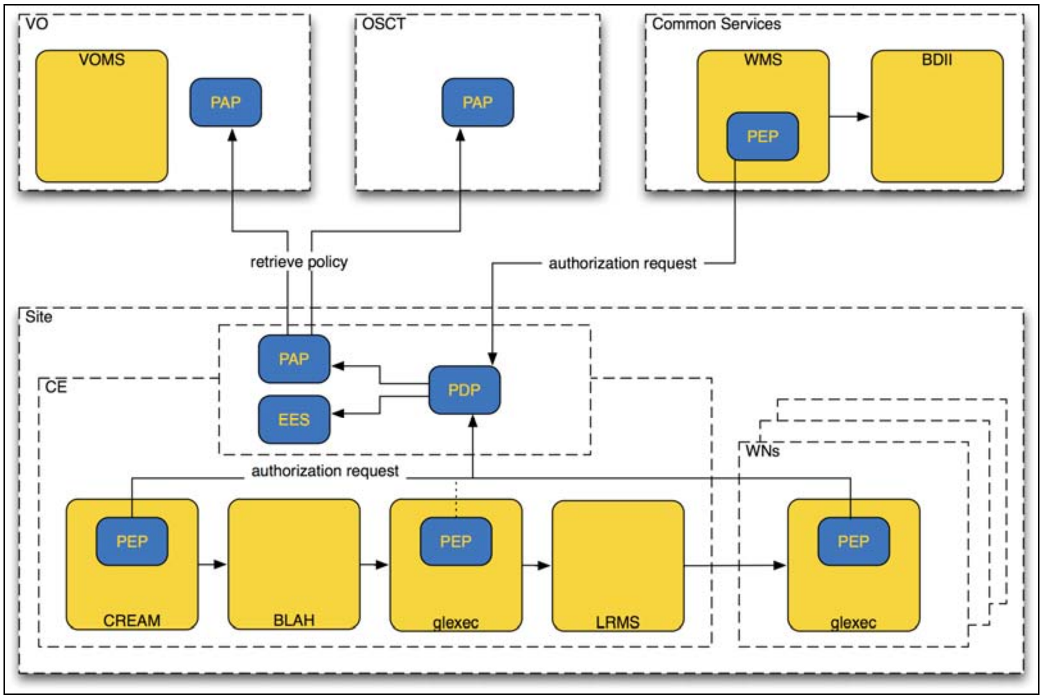

Figure 4 Architecture of the gLite Authorization Service

The effective policy for a given authorization decision is the composition of a site's local policy as well as any remote policies. The site has final control over which remote policies are pulled and how those policies affect the overall policy decision. For example, a site may indicate that if any policy, local or remote, indicates that the request is allowed than the returned result should be “permit”. Alternatively a site may indicate that its policy should overrule remote policies, or vice versa.

Lastly, the deployment model depicted in Figure 4 is likely to be the model used in most cases. However, deploying the PAP, PDP, and EES components is not without some amount of work. An organization may chose to use an alternative deployment model if they wish. PAP, PDP, and EES components may be hosted and run by a third party. Therefore an organization may outsource any, or all, of these components as a means of reducing the complexity at their site at a cost of increased request/response latency.

7. ISOLATION AND SANDBOXING

Access to remote computing or storage resources should on one hand be completely transparent to the user and on the other hand as secure as possible. The latter, however, requires among other things access limitations, which often are in contradiction to the former.

We consider two aspects in this section:

1. Securing the hosted to native interface, i.e. sandboxing the user's job on a Grid resource as it runs
2. Network isolation of the Grid resource.

The first aspect must be supported by the Grid middleware. The second is a site-specific network configuration issue, which must be respected by the Grid services.

7.1. SECURING THE HOSTED TO NATIVE INTERFACE

The user's actions should be *sandboxed* in order to minimize the impact on the local system. Within the sandbox the user's job is free to do whatever it wants, and the sandbox must be removed as soon as the job is finished. The generic creation of sandboxes, based only on an entity's Grid credentials, such as VO membership, relieves the local system administrator from the burden of administering local users at his site.

gLite currently supports as sandbox environment pre-configured accounts, into which a Grid job will be mapped. This requires two software components:

1. One that maps the Grid credentials into local credentials (i.e. one Unix UID and one or more Unix GIDs). In gLite this functionality is currently provided by LCMAPS.
2. One that provides a trusted "setuid" functionality. In gLite this functionality is provided by glexec.

A more elegant and more secure way to secure the hosted to native interface would be to run the Grid user processes inside a virtual machine[18], which provides the user job with a complete operating system. The virtual machine, contained in a single file, can be discarded after the Grid user processes are finished. The long-term negative effects of an adversary gaining root privileges in a virtual machine are negligible, and the network connections to/from a virtual machine can easily be controlled. However, today there are still performance degradations, mainly in the access to data. We assume that over time they will be overcome. However, new issues arise with the use of virtual machines. Firstly, traceability must also be guaranteed by virtual machines. Secondly, site administrators must, for example, in the case of a compromise, have some efficient way to perform

[18] It should be pointed out that by virtualization one typically means running virtual machines on the WN. On the other hand, as long as the middleware also allows job execution on the CE, then that execution should also be virtualized in order to completely sandbox any user program.

forensic analysis to the discover the cause of the breach and the extent of the damage done.
The gLite authorization service, currently under development, will provide hooks to support other execution environment systems, such as virtual machines (see section 6.2.3).

7.2. NETWORK ISOLATION

Firewalls set the boundaries of network utilization by enforcing (strict) rules on connectivity. Rules such as “no inbound connectivity allowed to worker nodes” are widely accepted and enforced among the sites. They wish to enforce an even more strict policy on the network use to avoid the chance of their site being liable for playing a role in any malicious use such as attacks on other connected systems on the Internet. This strict policy could mean “No network activity at all to or from the worker nodes”, rendering some applications unable to operate. To solve this problem, sites need to be able to secure the network at the fullest by default. However, they also need to be able to allow exceptions to their default policy.

There is currently no standard way to resolve these conflicting requirements. The implementations of the network isolations vary between sites. The fact is that some Grid services require outbound connectivity from the WN to services outside the site (e.g. for data access or access to external schedulers).

7.3. UNRESOLVED ISSUES AND WORK PLANNED IN EGEE-III

The need for a mediator service in the form of a Dynamic Connectivity Service was put forward in [R2]. Further investigation in this area is needed in order to design and implement such a service. Furthermore, the acceptance of such a service by the site administrators will have to be clarified as sites may be opposed to dynamically change firewall settings. There is currently no work planned in EGEE-III in this area.

8. IDENTITY SWITCHING ON THE WORKER NODES

Ideally, a request to execute a Grid job at a given site is mapped at the CE to a local identity, under which the job will be executed on the WN as scheduled by the local resource management system. As explained above (section 7.1), this is performed by an identity switching mechanism, called glexec. In this scenario, once the local identity has been determined, it would not be changed, until the requested job has ended.

However, in gLite the concept of *pilot jobs* has been developed, which is a special kind of job that is submitted to a computing resource. Once a pilot job starts to run on a WN, it pulls one or more jobs from a dedicated service and executes them one after the other. The advantage of the pilot job concept is that it allows selecting the next job at the time, when a slot on a site has become available. In other words, the user’s job will not be assigned to a given site early and risks to remain waiting for a long time until a slot has become available in the Local Resource Management System (LRMS). Instead, the user’s job is waiting in a special scheduler, which takes the scheduling decision at the time a slot on the Grid is available. This is called *late binding*, in contrast to *early binding.*

Pilot jobs offer the option of either running the user's job, called payload job, under the identity of the pilot job, or under the identity corresponding to the payload job's proxy certificate. The latter option requires that the identity must be switched on the WN.

In order to support the identity switch on the WN, the above-mentioned modules LCAS[19] and LCMAPS[20] have been ported into a web-service, called Site Central Authorization Service (SCAS).

Figure 5 shows the submission of a pilot job (1), which leads to glexec being called using the proxy certificate of the pilot job (3,4) followed by a submission to the LRMS (5). Once the job is executed on the WN (6), it pulls in the payload jobs from an external scheduler (7,8) and executes glexec using the proxy certificate of the payload job (9). glexec in turn calls SCAS (10,11) for authorization and obtains the identity to switch into for the payload job. Once the payload job is finished (12), the pilot job either terminates or pulls in another payload job from the external scheduler (7,8).

The communication between the client and SCAS is based on the SAML2/XACML2 standard and a common protocol has been formulated together with the Globus and OSG project.

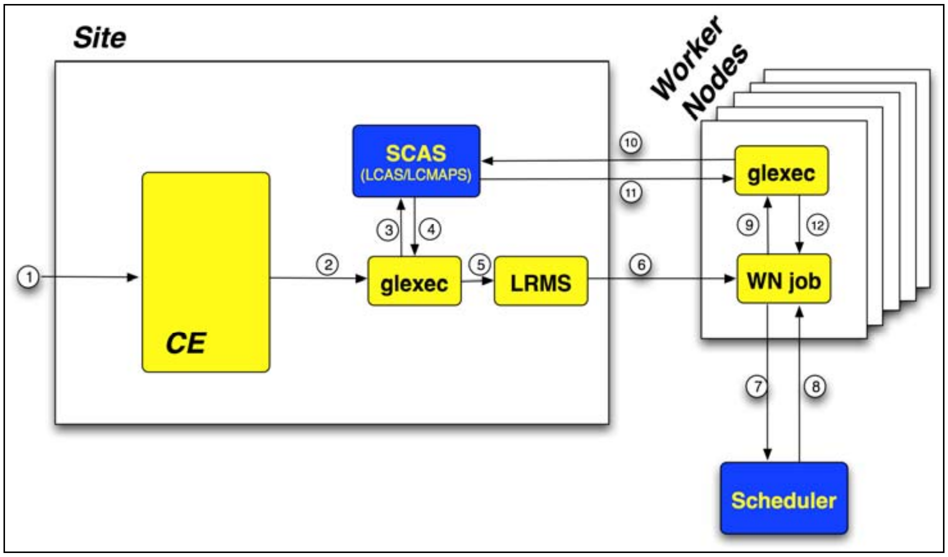

Figure 5 Site Central Authorization Service (SCAS)

It should be noted that SCAS is a short-term stop-gap solution, designed to enable the identity switching on the WN in a consistent manner at a site as quickly as possible. It also provides a central log for the identity switching at a given site. Its deployment is expected in late 2008/early 2009. As the new authorization service will become available (see section 6.2), SCAS will evolve into the so-called Execution Environment Service (EES), which initially covers only the LCMAPS functionality. In the

[19] LCAS authorizes a user based on the information in his proxy.

[20] LCMAPS determines the local identity (Unix UID and one or more GIDs), based on the information contained in the user's proxy certificate.

long run, the EES will support the provisioning of other execution environments such as workspaces [R15] or virtual machines.

9. DATA MANAGEMENT

9.1. UNENCRYPTED DATA STORAGE

Several data storage systems, that store the data unencrypted, are supported by gLite. The Disk Pool Manager (DPM) has the most mature authorization system. It is planned that the other two, Castor and dCache, will support the same model in the future. We therefore only describe the DPM model in this document.

9.1.1. Disk Pool Manager (DPM) and LHC File Catalogue (LFC)

All the files managed by DPM on a given installation are owned by the same Unix account, which serves as a management account. Every DPM instance contains a name service (a local database), where the actual file name, ownership and access information resides. The services exposed by DPM consult this name service for obtaining authorization information. Internally to DPM, every user and group is represented in this local database by a virtual UID and virtual GID, which is independent of any Unix UID or GID for the hosting operating system. These virtual UID/GIDs are just representations of individual users and groups of users.

The authorization mechanisms in DPM are based on how DPM maps the user credentials (proxy certificate with or without VOMS AC) to the virtual UID and GIDs. The mechanism is depends whether the proxy certificates contains a VOMS AC or not. In both cases the virtual UID is derived from the DN and the virtual GID either from the FQANs or a data mangement specific mapfile.

LFC, a file catalogue, is using the same model, whereas its virtual UID and GIDs are not correlated to the values used by DPM.

More information on the DPM security model can be found in [R14].

9.1.2. The File Transfer Service (FTS)

The File Transfer Service (FTS) is a service that moves data from one SE to another over a defined channel between the sites. FTS operates on individual files as well as sets of files. It has dedicated interfaces for managing the network resources and to display statistics about transfers. Optionally, it also supports lookup and registration of files in catalogues. The authorization is based on defined roles, to which the user is mapped using their DN. Mapping based on FQAN will be supported in the future.

FTS renews expiring proxy certificates through the use of MyProxy, similar to the renewal mechanism as described in section 4.8).

More information on the FTS security model can be found in [R14].

9.1.3. Work Planned in EGEE-III

Only maintenance work will be performed on DPM and FTS. It is considered to add quotas to DPM. The DPM security model is being extended to CASTOR.

9.2. ENCRYPTED DATA STORAGE

The encrypted data service solution provides a system that encrypts/decrypts files and stores/retrieves them from GFAL library-compliant storage systems. For additional security and redundancy, the encryption keys used to encrypt the files are split and stored in separate keystores (Hydra servers).

Hydra [R19] is the central element in the encrypted file storage solution. The Hydra service comprises at least one Hydra server and a set of command-line interfaces to perform the basic commands. The Hydra server consists of a MySQL database and a Tomcat server that acts to contain the actual Hydra software, written in Java.

The encrypted data service solution uses the Hydra servers to store the pieces of the encryption key and their associated Access Control List (ACL) information. Therefore the key pieces are only accessible by their respective owners. These encryption keys are split using a particular scheme, Shamir Secret-Sharing Scheme (SSSS) that provides another layer of protection. The encryption key may be split into "M" fragments and may be recovered with possession of "N" fragments, where N<M. These parameters "M" and "N" are configurable by the number of Hydra servers deployed and the SSSS algorithm respectively. In this case, the security and reliability of the overall service has been increased: an attacker would have to gain access to at least "N" Hydra keystores to recover and encryption key; a legitimate user can tolerate the loss of M-N keystores and still recover their key.

In addition, a legitimate administrator of a Hydra keystore would not have access to the full encryption key.

Encryption keys are generated in the Hydra system based upon the concept of a globally unique identity (GUID) of the file in question to be encrypted. This GUID can be retrieved from sources such as an LCG File Catalog (LFC) that contains the logical names for files inside a Disk Pool Manager (DPM) storage element. The GUID is necessary for retrieval and decryption of files.

9.2.1. Medical Data manager

In addition to the Hydra service there are some additional components available to form a complete medical data manager system.

There is an interface between the DPM storage element and the Digital Imaging and Communications in Medicine (DICOM) storage system. This DICOM system is designed for internal hospital usage and the DPM-DICOM interface provides a method to use medical images securely in a Grid environment.

In addition, a metadata catalog, such as AMGA, is used to store the medical-specific metadata of the images that are transferred to the DPM storage element.

9.2.2. Support and Evolution

According to the general work plan in EGEE-III, there are no new developments for Hydra/EDS/Medical Data Manager. The only requests for new functionality that will be accepted are those coming specifically from the Biomed (and other) user communities as a result of their usage of the system. These requests will be brought to the EGEE-III technical management board (TMB) and approved as official work within the constraints of the effort assigned.

The support of these components has been assigned to JRA1 members and the bugs submitted are handled. This work also requires some coordination with the Data Management group as these components draw heavily on the concepts and technologies of both security and data management.

10. LOGGING, TRACING AND AUDITING

Being able to trace and audit past actions is a prerequisite for operating a Grid infrastructure in a secure manner. However, it is only possible if good logging information is produced by the Grid services. For example, logging information is needed to

- Trace and time-stamp security incidents.
- Provide evidence of incidents (to enable action to be taken).
- Derive incident reports.
- Conduct security audits.

In order to meet these requirements, the information logged must be “useful” (correct and complete information, uniform format, easy to digest, easy to correlate log entries related to the same event or user interaction). It is impossible to state a priori exactly what information is of interest to log and what is not; as we don’t know under what circumstances we will consume the information.

The Joint Security Policy Group (JSPG) has formulated the logging and tracing requirements [R16] and guidelines exist in the EGEE developer’s guide. However, their implementation is neither monitored nor enforced through code reviews.

10.1. WORK PLANNED IN EGEE-III

There is only very limited dedicated effort planned in EGEE-III to address shortcomings in the area of logging, tracing and auditing due to limited available manpower. However, the issue has been clearly raised by the Operational Site Coordination Team (OSCT) and improvements will have to be made on a best effort basis by JRA1. Key in this regard will be an increased collaboration between Middleware Security Group (MWSG) and the OSCT.

Two areas have been particularly targeted:

1. A combined view of the different log files produced by the various services at the CE is needed in order to allow the local site administrators and security personnel to understand past decisions by the authorization service and the LRMS. This should become available together with the new authorization service.
2. The evaluation and analysis of the information stored in the Logging and Bookkeeping Service shall be improved.

11. CONCLUSION

This document describes the security building blocks of the gLite middleware and the work planned within the lifetime of the EGEE-III project. It explains how the security architecture is broken down into "building blocks", called facilities, that should be implemented across all Grid services in an uniform and consistent manner. The following facilities are identified: authentication, authorization, logging and auditing, key management, delegation as well as isolation and sandboxing.

The consistency of the security architecture has not yet been reached for all facilities: E.g. authentication is today treated consistently and is – equally important – a well-established process in the Grid environment through the IGTF standard. Authorization, on the other hand, is today implemented inconsistently and only supports very simple policies.

The work plan of the EGEE-III project only foresees targeted and limited changes in the security architecture in the following areas:

1. Development of a new authorization service, which allows applying a consistent and accurate authorization policy across the Grid (described in section 6.2)
2. Support for identity switching at the worker node in order to enable pilot jobs (described in section 8)
3. Improvements in the renewal of short-lived proxy certificates, which will allow reducing their lifetimes (described in section 4.8)
4. Targeted improvements and stop-gap measures in order to ease the handling security incident.

Besides identifying these work items, the document also makes the following statements on important developments in the area of Grid security:

1. The experience of operating the Grid over the last few years has shown that a "good" security architecture not only supports secure services, but is also simple to deploy and operate. Most importantly, easy to use tools must support incident handling.
2. Today's Grid security architecture is based on PKI. While it has been very successfully in establishing the Grid, it has also turned out that
    a. The average user has shown difficulty in handling long-term user certificates. This limits the widespread adoption of Grid technology.
    b. The average user has many possibilities of leaving weakly protected copies of his certificate on untrusted systems
    c. Many proxy certificates with no protection (but albeit short(er) lifetimes) are stored on several hosts of the Grid during the execution of jobs.

   The following remedies are identified:

    a. Automated issuance of (often short-lived) certificates based on existing identity management systems, most notably the emerging authentication and authorization infrastructures in Europe.
    b. Development of community specific portals, which have another authentication and authorization system than the Grid. In this case, the credentials have to be "translated" between the two security domains. Various technologies are currently been discussed, which comprise of using special classes of certificates or security token services.

   c. Better documentation on procedures for the users as well as site administrators are needed.
   d. Interoperability with other security infrastructures will gain in importance over the next few years with security token services bridging the gap between different infrastructures. However, it should be pointed out that much work is needed, not only on the software development side but equally important on the policy side as well.

   In the long run, i.e. beyond the lifetime of EGEE-III, other security concepts than PKI may prevail and may very well be integrated into the Grid. Most notably, the concept of Federated Identity has gained an enormous following in all the national Authentication and Authorization Infrastructures (AAI) that have been established mainly by the NRENs in Europe.

2. Isolation and sandboxing are important security requirements for the Grid. It is expected that the use of other execution environments (e.g. virtual machines) will find increase in the future. They will be supported in gLite through the execution environment service, which is being developed as part of the authorization service. No work is planned in the area of network isolation.
3. Privacy and anonymity are important - not only because some communities need it, but also because the current EU legislation requires it.

Last but not least, it should be stressed that the implementation of the security architecture is an ongoing evolutionary process, which requires increased collaboration between the developers of the security software, Grid deployment and operational teams as well as site security officers, typically in an environment of limited resources. The middleware security group and operational security coordination team are committed to close collaboration during the EGEE-III and beyond.