Aaauthreach Project Technical Report

Security in Computer Grid

Contributor: Yuri Demchenko <demch@science.uva.nl>

Abstracts

This chapter will describe the Grid Security Services Architecture (GSSA) as it is defined in the Open Grid Forum (OGF) documents, extending on some of its components and enabling technologies used in Grid security. The OGF standardisation work created a strong basis for secure and reliable operation of the global Grid infrastructure. The chapter provides an overview of the security related standards concerning secure Grid services communication, secure credentials and security mechanisms for authentication and authorisation.

The chapter provides detailed information about the Virtual Organisation (VO) concept and related operational procedures that are used to support user associations as collaborative communities in Grid environment. VOs and VO Membership Service (VOMS) are considered as a standard-de-facto for user attributes management and AuthZ in Grid.

The defined security services and mechanisms take direct implementation in the present Grid middleware frameworks as Globus Toolkit, gLite and Unicore. Short information is provided about tools for secure credentials management.

The chapter analyses the security aspects of the different types of Grids and some practical use cases that require extended functionality from the security services that need to support dynamic security context management and stateful services management.

The presented overview of the current development in the Grid standardisation and practice provides a good background for discussion about the possible areas of research to extend currently used and implemented security services models and frameworks that includes but not limited to: defining Complex Resource Provisioning model; supporting user/AuthZ session management; extending User Controlled Security Domain in Virtualised Workspace Service (VWSS); adding managed object support with policy obligations in computing oriented Grid applications; and exploring such emerging security concepts as Identity Based Cryptography (IBC) for building dynamic security associations in multi-domain Grid applications.

Copyright note

1 Introduction

In less than a decade Grids have developed from initial research idea to production ready technology and infrastructure. The initial Grid definition in one of the Grid foundational papers the “Anatomy of the Grid” [1] actually described the goal of this new technology at that time: “Grid systems and applications aim to integrate, virtualise, and manage resources and services within distributed, heterogeneous, dynamic “virtual organizations”. The more detailed Grid definition developed in later works included such main components as distributed infrastructure, dynamics, virtualisation, and user-defined security – the components that provide a framework for coordinated collaborative resource sharing in dynamic, multi-institutional virtual organizations (VO) [1, 2].

The Open Grid Services Architecture v1.5 (OGSA) published by the Open Grid Forum (OGF)[1] in 2006 defines the Grid as "A system that is concerned with the integration, virtualization, and management of services and resources in a distributed, heterogeneous environment that supports collections of users and resources (virtual organizations) across traditional administrative and organizational domains (real organizations) [3]. In the recently published document GFD.113 the Grid definition is extended to "Scalable, distributed computing across multiple heterogeneous platforms, locations, organisations" [4]. The document also defines the following characteristics and goals of Grids in general:

- Dynamic Resource provisioning
- Management of Virtualised Infrastructure
- Resource pooling and sharing
- Self-monitoring and improvement
- Highest quality of service

The following Grid types are identified depending on usage and required common functionality:

Cluster Grids – that have predominantly homogeneous structure and focused on shared use of high performance computing resources.

Collaboration Grids – that are targeted at supporting collaborative distributed group of people over multiple domains and involving heterogeneous resource.

Data Center Grids – are actually adding provider specific aspects in managing resources, users, their associations and supporting whole provisioning life-cycle.

Currently Grids found their place as a technology to build problem or project oriented collaborative environment that combine high-end compute, storage and visualization resources. However, wider Grids usage is limited by the complexity of the Grid resources setup and management. Cloud computing is emerging as targeting simpler use cases and potentially wider user community that require large volume but simpler computational. The recent GFD-I.150 informational document by OGF positions Clouds as providing higher-level abstraction to access Grid and compute cluster resources [5].

Grid security is identified as one of priority areas but in the recent and current developments at OGF it is mostly focused on the short-term goals to achieve interoperability of currently being developed Grid infrastructures, in particular such main security services and mechanisms as Authentication (AuthN), Authorisation (AuthZ) and Web Services Protocol Security [3]. As a fact of accomplishment of this priority goal, the OGF recently published a set of documents: OGSA Security Profile 2.0 (GFD.138 [6], also referred to as "Express Authentication Profile"), Secure Communication Profile 1.0 (GFD.132 [7]), and Secure Addressing Profile 1.0 (GFD.131 [8]).

It can be also mentioned that there is a gap between OGSA Security model/services definition and existing practical Grid implementation in large Grid projects such as LCG/EGEE[2], OSG[3]. These Grid infrastructures use different implementations of Grid middleware and successfully made them working together. Some practical interoperability initiatives came out of these projects and brought to OGF, but many others still remain developed outside of the OGF standardisation process. This

[1] http://www.ogf.org/
[2] http://www.eu-egee.org/
[3] http://www.opensciencegrid.org/

chapter provides a summary of ongoing research, development, practices and identified problems in Grid security.

2 Security in Grid Resources and Users Management

The three types of Grids defined in OGF Roadmap document [4] the Cluster Grids, the Collaboration Grids and the Data Center Grids provide a good basis for identifying basic common and specific security functionalities required for each case. It is not a goal of this paper to make detailed specification of all required security functionalities but we simply point on or refer to some differences between required/suggested security services/infrastructure operations.

Although the Cluster Grids deal with potentially homogeneous computing environment, the major security challenge/problem here is that the required security solutions need to bridge between open services oriented environment (basically using Web Services or other messaging platform over open Internet or networking environment) and generally trusted execution environment. These two realms use different operational and security models which we discuss later.

The Collaboration Grids need to solve a task of managing distributed multidomain/multi-organisational users and resources associations, which in current Grid practice called Virtual Organisations. Such associations may be static or created dynamically, and Grid resources also may be assigned to VO statically or provisioned dynamically for some experiments. And so, the security infrastructure needs to support inter-domain attributes, policies, and trust management.

The Data Center Grids bring the whole spectrum of the security aspects and problems related to typical provider operation. We can just mention that few of them are related to defining a general Grid resource provisioning, securing virtual execution environment, and user session management.

It is important to discuss another use case the provisioning of the dedicated high-speed network infrastructure. Although network provisioning tends to use the Grid middleware and consequently manage network as Grid resources, it can bring a new experience and the generic solutions from the multidomain network resource provisioning which can be used for developing common provisioning and security architecture for Grid enabled resources.

Based on their extensive experience in both networking and high-performance Grid computing, authors have a good opportunity to bring together and combine experience from two areas to develop effective and easy manageable security solutions for both Grid and network resources.

3 Enabling technologies in Grid Security

Current OGSA/Grid Security services model adopted Web Services security model which in its own turn inherited approach and basic concepts of the Open System Interconnection (OSI) Security Architecture and consequently the client/service security model. However, Grid operation generically deals with managed objects, which are jobs, processes and assigned resources. This creates a gap between inherited limitations of the OSI client/server security model, that may be considered generically stateless, when trying to solve managed objects security problems which in general require stateful services.

To position correctly the discussed here Grid Security Services Architecture (GSSA) and understand shortage of many proposed and currently used solutions in Grid security, we will revisit the basic security concepts in networking and computing that provide a foundational base for building consistent GSSA, in particular, the OSI Security Architecture and the security concepts used in Trusted Computing Base (TCB) such as Reference Monitor (RM), Multi-Level Security (MLS), Clark-Wilson integrity and manageability model, which were resulted from mainframe oriented security research in 1970s-80s.

Such an overview will also provide a necessary context for considering few some research areas in Grid Security.

3.1 From OSI/Internet to Web Service and OGSA Security

Current Internet infrastructure and networking technologies are built in compliance with the Open Systems Interconnection (OSI) model. The OSI Security Architecture (ISO7498-2/X.800 [9]) provides a common framework and approach for developing secure protocols and applications. The ISO/ITU standards specify the basic security services and mechanisms and their relation to the OSI layers. The standard also suggests relations between security services and security mechanisms. The OSI security architecture is fully applicable to the Internet TCP/IP protocol stack due to their direct mapping at the data, network, and transport layers.

Security services, in the context of the OSI security architecture, are defined as services, provided by a protocol layer of communicating or interacting systems, which ensure adequate security of the systems or data transfers. To ensure openness and interoperability of interacting systems, the services are defined for specific OSI layers and may use one or more security mechanisms. Security policies are used to manage security services and can be a part of an application specific security service implementation.

The philosophy behind OSI security architecture is that security services and mechanisms can be added independently using standard/specified interfaces (as illustrated on Fig. 1). The following are inherited key features of the OSI/Internet security architecture:

- Internet/OSI model suggests that interconnected systems are managed independently and communicated using protocols specific to each OSI/Internet layers.
- Trust relations between systems established mutually or via 3rd trusted party, a group of system can create an administrative and trust domain.
- Public Key Infrastructure (PKI) provides a basis for trust management, authentication and key exchange
- Communication and security protocols can use a session related security context.

The same philosophy was inherited by the whole development of the Internet and web based applications and later by the Web Services Architecture (WSA) [10, 11] and consequently by OGSA services model [3].

The WS-Security services model uses actually the same approach in defining security services interfaces which use the SOAP message header for adding security related information and context. This makes the security services independent from the main service call which is typically placed into

the SOAP message body [10]. In this respect WS-Security services can be also considered as orthogonal to main services and in general arbitrary combined. This confirms that current Web Services Security architecture inherited basic principles from the OSI/Internet Security Architecture.

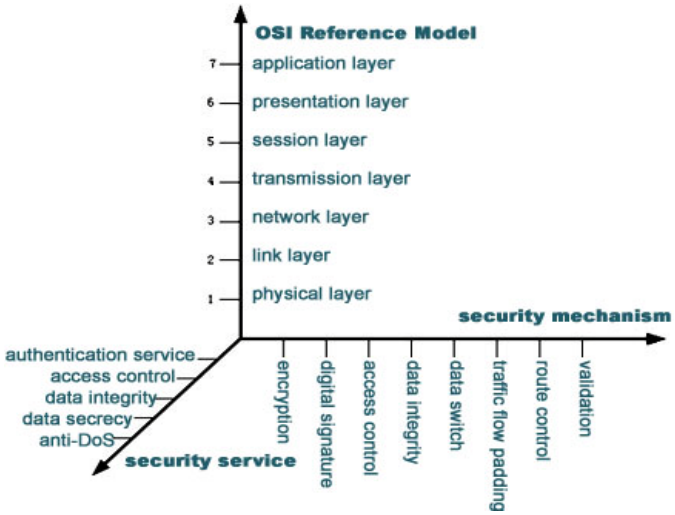


Figure 1. Relation between OSI security services, mechanisms and OSI reference model layers

We can also make an observation that the introduction of the Web Services Resource Framework (WSRF) [12] and recent developments of the Web Services Resource Transfer (WS-RT) [13] and WS-Management [14] (Web Services for Management - http://www.dmtf.org/standards/wsman/) are other attempts to address the problems of managing stateful processes in Grids with generically stateless Web Services.

3.2 Trusted Computing Base and Reference Monitor Concept

Reference Monitor (RM) concept was proposed by J.P. Anderson in the report “Computer Security Planning Study” (1972) [15] and was used as a basis for developing Trusted Computing Base (TCB) concept and architecture. As originated from the military research, the RM property provides a basis for Multi-Level Security (MLS) that can be abstracted as:

- **Complete mediation**: The security rules are enforced on every access, not just, for example, when a file is opened.
- **Isolation**: The reference monitor and databases must be protected from unauthorized modification.
- **Verifiability**: The reference monitor’s correctness must be provable. That is, it must be possible to demonstrate mathematically that the reference monitor enforces the security rules and provides complete mediation and isolation.

The following can be regarded as the basic security models used in TCB and MLS:

- Bell–LaPadula (BLP) MLS policy model [16] to protect data confidentiality that can be described as “No write down” and “No read up

- Biba model [17] to ensure data integrity that can be described as “No write up” and “No read down”. Biba model can be applied to control and management data protection in an open environment.
- Clark-Wilson data integrity policy model [18] that defines both policy enforcement and certification rules that can be shortly summarised as:
  - Authentication of all user accessing system
  - Logging and auditing all modifications
  - Well-formed transactions
  - Separation of duties

The Clark-Wilson model was initially proposed to ensure reliable business operation, it is used in developing internal OS security management policies, and in Grids it can be also applicable for creating Grid Data operational security policies.

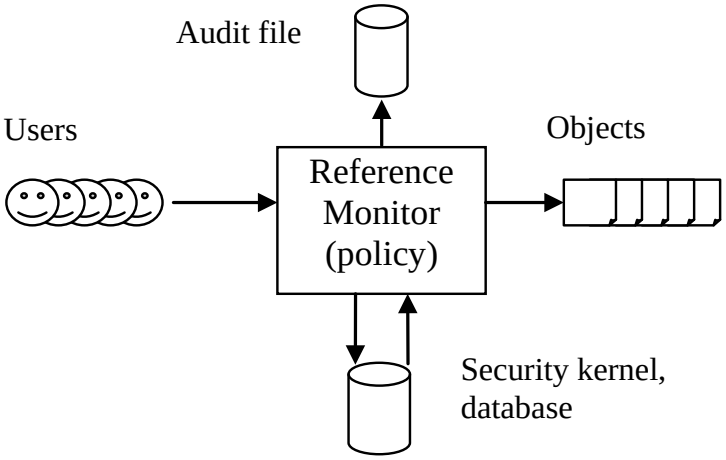


Figure 2. Reference Monitor model (applied in an “orthogonal” way to all system calls).

3.3 Trusted Computing Platform Architecture (TCPA)

The TCPA [19, 20] provides a basis for building and managing controlled secure environment for running applications and processing (protected) content and can be considered as TCB development for open networking environment.

The TCPA defines the five abstract layers: platform, system (including OS), service/application, and user identity. It is built around the functionality of the Trusted Platform Module (TPM) [21] - a chip built-in into the computer system or a smartcard chip that provides a number of hardware based cryptographic functions to ensure integrity and trust relation between TCPA layers. The following TPM functions are specifically targeted to improve privacy protection in TPM based systems: Endorsement Key (EK) that allows anonymous TPM identification through “zero knowledge” cryptography (without revealing actual identity or secret), the Direct Dynamic Attestation (DAA) that can securely communicate information about the static or dynamic platform configuration. In respect to the trust management, the TPM provides a platform-tied “root of trust” that can be used for secure platform registration and as an initial trusted secure session initiation (also referred to as “trusted introduction”).

The TCPA defines five abstraction layers: platform, system (including OS), service/application, and user identity. It is built around the functionality of the Trusted Platform Module (TPM) [21] - a chip built-in into the computer system or a smartcard chip that provides a number of hardware based cryptographic functions to ensure integrity and trust relation between TCG layers:

- Asymmetric key functions for on-chip key pair generation using hardware random key generation; private key signatures; public key encryption and private key decryption.
- An Endorsement Key (EK) that can be used by a platform owner to establish that identity keys were generated in a TPM, without disclosing its identity.
- Direct Autonomous Attestation (DAA) that securely communicates information about the static or dynamic platform configuration, which is internally stored in TPM in the form of hashed values.
- Protection of communication between two TPM.
- Monotonic counter and the tick counter to enable transaction timing and sequencing.

TPM provides a platform-tied “root of trust” that can be used for secure platform registration and as an initial trusted secure session initiation (or “trusted introduction”).

The TCPA has been developed with the following philosophy [20]: incremental implementation; available as opt-in functionality; the possibility of anonymous TPM identification through “zero knowledge” cryptography; the possibility to migrate (or backup) TPM keys to another TPM without disclosing them in clear. Trusted platform (TP) lifecycle includes six phases supported by three types of infrastructure: pre-deployment/provisioning (supports manufacturing, delivery phases), deployment (supports deployment, identity registration, operation phases), and redeployment/retirement (supports recycling and retirement phases). In this respect the TCPA lifecycle stages can be naturally integrated with the discussed below the Complex Resource Provisioning model.

The TCPA Trusted Network Connect (TNC) platform [22] is focused on establishing and enforcing security policies before and after endpoints or clients connect to multi-vendor environments. Among other requirements that improve end-points administration, TNC defines end-point configuration measurements against compliance security policies before the connection to the network is allowed. The TNC uses the IETF AAA Authorisation Framework [21] to add TPM based policy enforcement mechanisms to the TCG network infrastructure layer. On other hand, the TNC describes how the TPM functionality can be used to improve security of communications between AAA components in an open multidomain environment, in particularly to support “trusted introduction” of new network devices and reliable key distribution in multidomain network/resource provisioning.

In section 7.3 below we will discuss how the TCPA and TPM can be used to build user-controlled virtual workspace service.

4 Grid Security Services Architecture

4.1 OGSA Grid Security Services Model

OGSA Security Services model is a part of the general OGSA [3]. OGSA security architectural components are required to support, integrate and unify available security models, mechanisms,

protocols, platforms and technologies to enable a variety of systems to interoperate. Security services group encompass issues relating to the management and verification of credentials; privacy and integrity; and policy.

OGSA Security Services model defines all scope of services required to ensure end-to-end security of Grid services and applications: authentication, confidentiality, message integrity, policy expression and exchange, authorisation, delegation, single logon, credential lifespan and renewal, privacy, secure logging, assurance, manageability, firewall traversal, and messaging layer security.

Establishing secure communication or context involves policy exchange and evaluation between service requestor and service provider. Policy can specify supported authentication mechanisms, integrity and confidentiality requirements, trust policies, privacy policies, and identity constraints. The security (and trust) model must provide a mechanism by which authentication credentials from the service requestor domain can be translated into the service provider domain, and trust relations are established.

Security domain for Grid services and applications may be defined by VO created on the base of agreement and establishing its own trust domain. VO members remain administratively independent and may continue running their own security services, the VO may provide a bridge for establishing trust relations between requestors and providers from different administrative and trust domains inside VO. The security model must provide a mechanism by which authentication credentials from the requestor domain can be translated into service provider domain.

OGSA Security Services model incorporates existing and emerging WS-Security standards [11] and includes the following layers and components (see Fig. 3, from the bottom up):

1) Communication/transport Security Layer defines network infrastructure security and uses such network security services as SSL/TLS, IPSec, VPN, SASL, and others.

2) Messaging Security Layer is based on currently well defined and supported by different Web Services platforms SOAP/WS-Security [23]. It also uses relevant XML Security mechanisms: XML Signature [24], XML Encryption [25], and SAML [26] security token exchange format. At this level security mechanisms are directly incorporated into OGSA services and definitions/formats.

3) Policy Expression and Exchange Layer defines set of policies applied to Grid Services and Grid operational environment which are required to ensure multi-domain and multiplatform compatibility. Policy layer provides necessary policy information for the Service/Operational Security layer. The WS-Policy specification [??] provides a framework to describe policies in a standard way and mechanism to include policies into service definition.

4) Services/Operational Layer defines security services/mechanisms for secure operation of Grid services in an open environment and includes:

- Secure Context Management
- Identity and Credential Translation and Federation
- Authorisation and Access Control Enforcement
- Auditing and Non-repudiation

Some of layers and components are described in more details below.

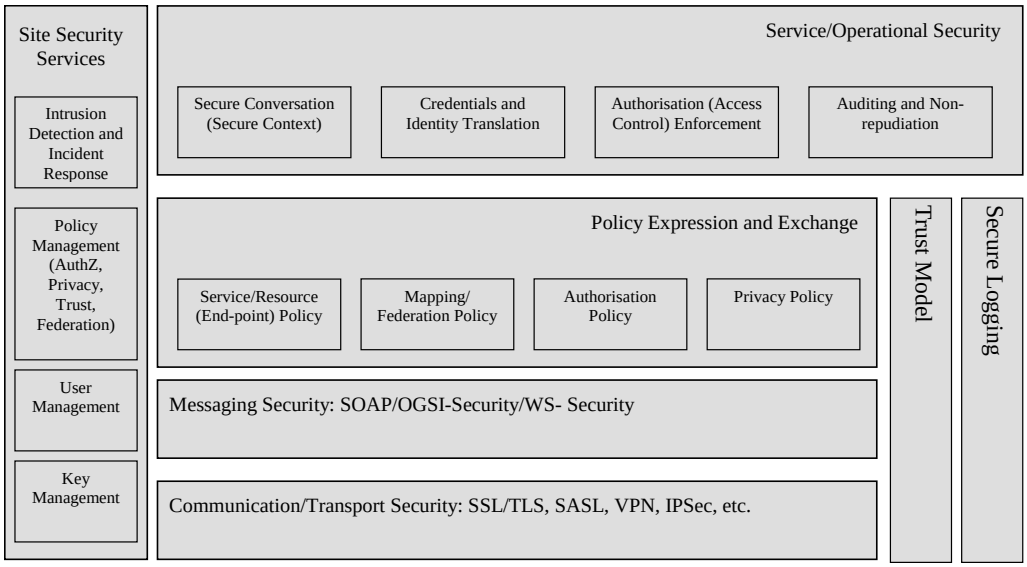


Figure 3. Components of the OGSA Security Services Model.

a) Policy Expression and Exchange Layer

Interacting Grid services need to confirm to certain requirements in order to securely interact. It is important that service or resource requestors have access and understand policies associated with the target service. As a result, both the service requestor and service provider select acceptable security profile. It is also important to mention that the privilege to acquire Security Policy is given by the hosting environment to authenticated and authorised entities only.

Policy expression and exchange layer includes (but not limited to) the following policies:

- Local site policy and resource access policy, including VO policy
- Identity association/mapping and federation policy
- Trust policy, and
- Privacy policy

Policy layer provides necessary information for policy enforcement modules of the Service/Operational Security layer. It is suggested that policies expression should confirm to WS-Policy [27] (and WS-SecurePolicy [28] and WS-PolicyAttachment [29] extensions) that provides extensible framework that can be configured for specific applications based on several common attributes including privacy, security token requirements, token and other related information encoding, supported algorithms.

Trust policy management provides a mechanism by which level of trust to the claims and assertions presented by others/entities is defined, and expressed in the Policies. Trust management issues are addressed by WS-Trust [30] defined in the WS-Security.

Privacy policy management provides a mechanism to exchange and evaluate requestor and provider privacy policy to protect user anonymity or withhold private information.

b) Secure Context Management

Secure Conversation service adopts and leverages WS-SecureConversation specification [31] to maintain consequent messages exchange between the Grid services that may span different VO's

and over open network environment. Secure Conversation will maintain secure context established during initial mutual authentication for the period of active communication session between interacting application end points. Secure Conversation will operate at the SOAP message layer providing also binding with the policies associated with the end points.

c) Identity and Credential Translation and Federation

Grid services and applications typically span over multiple VO/locations and security domains that maintain their independent security services and policies. Operations between entities in different domains will require mutual authentication. Different security domains may incorporate different format and semantics for requestor/provider identities and credentials. Interoperation will require federation of the involved domains and identity and credentials translation or mapping. This federation may also be accomplished through trusted proxies or broker services. Identity mapping and federation is a subject to VO or local policies.

OGSA Identity specification will define how the identity name for an OGSA entity should be constructed based on the entity's identity established within their security domain. The specification considers cross-realm uniqueness, anonymity, and identity mapping. Other specifications will define cross-realm mapping for generic names, policy and credentials.

Specific for Web services and Grid services, delegation mechanism allows for a requestor to delegate some subset of their rights based on credentials delegation in order to fulfil the request. Delegation is based on credentials delegation by the authenticated entity and uses identity assertion profile to express identity assertion associated with a request, credential or communication context.

Identity and credential translation service can be built on two currently available identity management specifications WS-Federation [32] (together with other complementary specifications WS-Trust, WS-Policy, WS-SecureConversation and Liberty Alliance Project [33].

d) Authorisation and Access Control

Authorisation and Access Control security service is a key part of the managed security in an open service oriented environment. Authorisation is typically associated with a service provide or resource owner, who control access to a resource based on provided by requestor credentials or attributes that define requestor's privileges or roles bound to requestor's identity. Separation of Authentication and Authorisation services allows dynamic role based access control management and virtual association between interacting entities, and provides a basis for privacy in an open environment.

Authorisation and Access Control service in Grid applications/VO will re-use models proposed in WS-Authorisation that describes how access policies are specified and managed. Exchange of Authentication credentials and Authorisation attributes is typically based on security token definition and exchange protocols defined in SAML [26] and XACML [34].

e) Auditing and Non-repudiation

Auditing and non-repudiation are necessary components for security services assurance and policy enforcement. They provide secure logging functionality that is required for many higher level audit related functionalities. Some limited auditing functionality may be required for other services at the Service/Operational Security level, in particular, timestamping.

f) Security Services Management

Effective and reliable operation of the security services requires underlying security services management and may include:

- key management for cryptographic functions;
- user management including user registry and related role or privilege management;
- policies management that includes local operational security policies, services security policies and trust management;
- intrusion detection and incident response capability.

These functions are related to local Grid sites or VO's.

4.2 OGSA Basic Security Profile

An OGSA Basic Security Profile (GFD-R-P.138 [6]) defines a basic level of security for OGSA based Grid services. It is defined as a profile of the WS-Interoperability (WS-I) [35] and comprises of the two standards: Secure Communication Profile (GFD-R-P.132 [7]) and Secure Addressing Profile (GFD-R-P.131 [8]).

The Secure Communication Profile provides a basis for ensuring interoperability between WSRF based Grid services. The profile specifies interoperability requirements (or conformance statements) concerned with the security mechanisms that can be used to ensure authentication, integrity and confidentiality properties in communication between Grid services. More specifically, this profile refines the WS-I Basic Security Profile [36] and the WS-SecurityPolicy [28] specifications and serves to define normative, "well-known" policy documents identifying commonly-used secure communication mechanisms. WS-SecurityPolicy provides a flexible, extensible approach for specifying the security tokens, cryptographic algorithms, and protocol mechanisms (both at the transport and message levels) needed to securely communicate with a given Web service resource. The normative policies defined in the Secure Communication profile can be referenced by name and composed within resource-specific security policies. The security mechanisms implied by these named policies are well-defined by external profiles that are incorporated by reference, and this document serves as a point of further refinement for these mechanisms.

The WS-Addressing Core [37] specification is often used to address Web Service resources using endpoint reference (EPR) data structures. The WS-Addressing definition of the EPR describes the encapsulation of the network protocol and endpoint information for a given resource, but does not specifically indicate how the EPR can also be used to convey the secure-communication mechanisms (and ancillary security tokens) required by that resource. Such security requirements can be described using WS-SecurityPolicy policy documents.

The Secure Addressing Profile document normatively refines the WS-Addressing Core specification in order to facilitate the inclusion of such WS-SecurityPolicy assertions within WS-Addressing endpoint references. It is often the case that WS-Addressing EPRs may be stored and exchanged by any number of intermediaries (such as directory services) before being consumed for actual communication. A particular interaction scenario may require guarantees of trust regarding the

identity of the minter and the integrity of the EPR. This document also normatively describes how XML-Signature is used provide such guarantees.

5 Practical Grid Security

5.1 Authentication and Authorisation in Grids

Authentication in Grids is based on PKI and can use different types of (user) credentials (X.509 Public Key Certificates (PKC) [38], SAML assertions [26], Kerberos tickets [39], password, etc.). Delegation (restricted and full) is a necessary mechanism in Grids to manage distributed Grid job submission and staged execution. Delegation is implemented by using X.509 Proxy Certificate [40] as used in the Globus and gLite middleware or special type of SAML assertion for delegation as used in the UNICORE6 middleware. The Proxy certificate is generated by the user client or other entity acting on behalf of a user based on user master PKC or previous Proxy certificate.

Authorisation is based on the VO attributes assigned to a user by the VO and typically managed by the VO membership service (VOMS) [41]. VOMS attributes are provided as VOMS Attribute certificate [42] or VOMS SAML attribute assertion [43] and typically included into the Proxy. These user capabilities will be evaluated by the Authorisation services against the policy when requesting access to a resource. In fact, Proxy with AC/assertion can be treated as user session credential and support simple session management functionality.

Interoperability of Authentication and Authorisation services is an important problem in Grid to enable consistent security services across global heterogeneous Grid infrastructure. This is achieved by establishing/documenting best practices, developing related standards at Open Grid Forum as well as by using industry standards and profiling general security standards for Grids. The OGSA Basic Security Profile [6] and Grid Certificate Profile [44} standards provide such examples.

A set of standards for authorisation service interoperability have been developed by the OGSA Authorization Working Group (OGSA-AuthZ WG). The goal is to leverage authorization related frameworks, standards and practices from other application areas, in particular, the Web services world (e.g. SAML, XACML, the WS Security suite) and define specification how the existing solutions should be used or profiled for Grid services involving multiple authorisation domains.

The following OGSA-AuthZ WG documents provide a basic set of standards for interoperability in Grid authorisation:

- "Functional components of Grid Service Provider Authorisation Service Middleware" [45] that specifies the major functional components and their interaction scenarios to built interoperable pluggable authorisation service middleware for distributed Grid services. The document introduces and describes functionality of such important components as the Credentials Validation Service (CVS) and Context Handler. The CVS provides functionality to check authenticity of the supplied in authorisation request credentials and validate them (including translation or mapping if necessary) for use in the specific authorisation context or scenario. The Context Handler provides all functionality and aggregates all components that support communication between PEP and PDP and their interaction with other authorisation service components such as CVS, Obligations Handler or external attribute service.
- "Use of WS-Trust and SAML to Access a Credential Validation Service (CVS)" [46] that specifies a credential validation protocol between the PEP and a credential validation service (the returned result is a set of validated attributes).
- "Use of XACML Request Context to obtain authorisation decision" [47] that specifies the authorisation protocol between the PDP and the PEP that suggests using standard XACML request and response messages format over general Web Services protocol or over SAML-XACML protocol. The document also describes the use of policy obligations which are defined in the XACML authorisation framework as conditions that must be enforced by the PEP depending on the PDP policy decision. Policy obligations are returned in the XACML response as a part of authorisation decision.
- "Use of SAML to retrieve Authorization Credentials" [48] – the document provides general recommendations for using SAML protocol and assertions format to request authorisation credentials from Attribute Authority Service.

Besides OGF driven standardisation activity, there are numerous community driven initiatives to ensure Grid middleware interoperability. They are built around mentioned above Grid projects and consortia. One of such initiatives the joint OSG-EGEE Authorisation interoperability Working Group has produced the common XACML-Grid attributes and policy profile [49] that is being jointly implemented by partner projects. The profile version 1.0 documented a number of common attributes and policy models for typical Grid applications and formalised use of the policy obligations in Grid defining the Reference Model for Obligations Handling (OHRM) [50]. The pluggable Generic AAA Toolkit (GAAA-TK) Java library [51] has been developed as a part of the Phosphorus[4] project that was focused on developing the combined on-demand Grid and network resources provisioning for Grid based applications. The GAAA-TK library supports XACML-Grid profile and implements the OHRM as pluggable modules that can be used together with the major Grid middleware.

Trust management is another important component of the Grid security and PKI based authentication and delegation. Trust relations are represented by a certificate chain that include Grid Certification Authority (CA) certificate and may include a number of successively generated Proxies. It is important to notice that global trust relations in Grids are maintained by the International Grid Trust Federation[5] (IGTF).

[4] http://www.gridpma.org/gridpma.html
[5] http://www.ist-phosphorus.eu/

5.2 Proxy Certificate and Delegation in Grid

Many Grid usage scenarios require a remote service to act on a user's behalf, e.g. a job running on a remote site needs to be able to talk to other servers to transfer file, and it therefore needs to prove that it is entitled to use the user's identity (this is known as delegation).

A Proxy Certificate [40] (or Proxy) allows limited delegation of rights. Strictly speaking, a Proxy is a "not qualified" certificate as it is issued by a user client and not a CA. To make a Proxy, a new public/private key pair is created and a new certificate is built containing the public key using a name with the form of the following example:

/C=UK/O=eScience/OU=CLRC/L=RAL/CN=john smith/CN=proxy

It is signed with the certificate's long-term private key. Proxies normally have a rather short lifetime, typically 12 hours. A new proxy can be generated from the existing proxy. The currently recommended practice limits maximum delegation length by 10, what means that the proxy validation/trust chain should not be longer than 10.

When a Grid job is submitted, the proxy certificate, the private key for the proxy and the normal certificate (but not the long-term private key) are sent with it. When the job wants to prove its delegated identity to another service, it sends it the proxy certificate and the standard certificate, but not the proxy private key. This information is sufficient to prove that the remote service has the right to use the delegated identity. If the remote service needs to further delegate rights to the identity to other services, it may create a new proxy based on the first one and give that proxy to the other services, lengthening the certificate validation chain.

In terms of security, a proxy is a compromise. Since the private key is sent with it, anyone who steals it can impersonate the owner, so proxies need to be treated carefully. There is no mechanism for revoking proxies, so in general, even if someone knows that one has been stolen, there is little they can do to stop it being used. On the other hand, proxies usually have a lifetime of only a few hours, so the potential damage is fairly limited.

The proxy contains information about the identity of the user, that is, the user's Distinguished Name (DN), a public and private key pair, and is signed by the original certificate. The proxy may also contain user attributes related to their membership in particular VOs including assigned groups and roles which can be used for authorisation in Grid. This information is provided by the VO Membership Service (VOMS) [41] and can be expressed in a form of the X.509 Attribute Certificate or SAML Attribute assertion.

At the time the proxy is created, one or more VOMS servers are contacted, and they return an Attribute Certificate that is signed by the VO and contains information about group membership and any requested roles within the VO.

The VOMS operates as an authorisation Attribute Authority providing user attributes (e.g., VO name, groups, roles, etc.) in a form of X.509 Attribute Certificate or SAML assertion to the user or to the Grid resource that require user authorisation. When accessing a Grid resource a user or a user

client on a user's behalf can present VOMS attributes/credentials as a part of a specially created VOMS Proxy which is a Proxy Certificate with embedded VOMS AC or SAML assertion.

Proxy containing VOMS attributes is called VOMS Proxy and used for authorisation in Grid. To create a VOMS proxy, the ACs are embedded in a standard proxy, and the whole thing is signed with the private key of the parent certificate. Services can then decode the VOMS information and use it as required, e.g. a user may only be allowed to do something if he has a particular role from a specific VO. One consequence of this method is that VOMS attributes can only be used with a proxy, they cannot be attached to a CA-issued certificate.

One other thing to be aware of is that the proxy and each AC has its own lifetime. Typically each AC has the same expiration time as the proxy as a whole, but it is possible that they may be different depending on VO policies and on the times specified when the proxy is created. VOMS servers usually limit the AC lifetime to a maximum of 24 hours, although a higher limit has been agreed in some cases. Differing expiration times often causes authorization problems on the grid.

More information about VOMS operation is provided in section 6 below.

5.3 Grid Security Middleware and Secure Credentials Management Tools

Grid infrastructure and applications rely on the Grid middleware that provides a common communication/messaging infrastructure for all resources and services exposed as Grid services, and also allows for a uniform security configuration at the service container or messaging level. This significantly simplifies development of Grid-based applications and allows developers to focus on application-level logic.

The major Grid middleware implementations and frameworks that currently run production Grid services are Globus Toolkit being developed by Globus Alliance[6] and currently primarily used in OSG[7] Consortium, gLite[8] middleware being developed by the European Grid project EGEE[9], and UNICORE6[10] Grid middleware developed and used by the Unicore Grid infrastructure and applications.

The Grid security middleware is an important component of the general Grid middleware and allows secure Grid services invocation and secure access to the distributed Grid resources. It is also important to mention that OGSA Grid services architecture/infrastructure needs to bridge between open Web services based user and job management infrastructure and typically UNIX based protected execution environment of the computer clusters and farms, what creates a number of specific security problems with identity switch/mapping, security context and security session management.

One of the important functional and structural components of the gLite and Globus security middleware is the gLExec module that provides a gateway between open Grid infrastructure

[6] http://www.globus.org/
[7] http://www.opensciencegrid.org
[8] http://glite.web.cern.ch/glite/
[9] http://www.eu-egee.org/
[10] http://www.unicore.eu/

environment and protected task execution environment of the Computer Element (CE) or Worker Node (WN) [52, 53]. When handling Grid jobs submission workflow, it contacts Site Central Authorisation Service (SCAS) [54, 55] which consequently involves such services as Local Center Authorisation Service (LCAS), Local Credential Mapping Service (LCMAPS) and Execution Environment Service (EES) to obtain authorisation decision and map Grid credentials to CE/WN pool accounts.

MyProxy Credentials Management Service

MyProxy [56] is open source software for managing X.509 Public Key Infrastructure (PKI) security credentials (certificates and private keys). MyProxy combines an online credential repository with an online certificate authority to allow users to securely obtain credentials when and where needed. Users run `myproxy-logon` command to authenticate and obtain credentials, including trusted CA certificates and Certificate Revocation Lists (CRLs).

Rather than storing Grid credentials on each machine a user can store them in a MyProxy repository and retrieve a proxy credential from the MyProxy repository when needed. MyProxy was designed with the following usage scenarios in mind.

After obtaining a certificate from a Certificate Authority (CA), a user can store a proxy credential based on that certificate in the MyProxy repository using the myproxy-init command. By default, `myproxy-init` stores a credential valid for 7 days, but it can also be used to generate longer-lived credentials. Then, whenever a credential is needed to access Grid resource, the user can retrieve a short-lived proxy from the MyProxy repository with the myproxy-logon command. This makes it easy to access your credentials without needing to manually copy certificate and key files between systems, which is prone to error and can be a cause of security problems.

Storing credentials in a MyProxy repository allows users to easily obtain RFC 3820 proxy credentials, without worrying about managing private key and certificate files. They can use MyProxy to delegate credentials to services acting on their behalf (like a Grid portal) by storing credentials in the MyProxy repository and sending the MyProxy passphrase to the service. They can also use MyProxy to renew their credentials, so, for example, long-running jobs don't fail because of expired credentials. A professionally managed MyProxy server can provide a more secure storage location for private keys than typical end-user systems. MyProxy can be configured to encrypt all private keys in the repository with user-chosen passphrases, with server-enforced policies for passphrase quality. By using a proxy credential delegation protocol, MyProxy allows users to obtain proxy credentials when needed without ever transferring private keys over the network.

For users that don't have PKI credentials yet, the MyProxy Certificate Authority (CA) provides a convenient method for obtaining them. The MyProxy CA issues short-lived session credentials to authenticated users. The repository and CA functionality can be combined in one service or can be used separately.

MyProxy provides a set of flexible authentication and authorization mechanisms for controlling access to credentials. Server-wide policies allow the MyProxy administrator to control how credentials may be used. Per-credential policies provide additional controls for credential owners. MyProxy supports multiple authentication mechanisms, including passphrase, certificate, Kerberos, Pubcookie, VOMS, PAM, LDAP, SASL and One Time Passwords (OTP).

Short Lived Credentials Service (SLCS)

In recent years Authentication and Authorization Infrastructures (AAI) were introduced in the academic and research sector. In most cases the AAI projects were driven by associations of higher education networks (EDUCAUSE[11] in US) and National Research and Education Networks (NREN) in Europe. There was a strong motivation for adding interoperability between these national/campus AAIs mostly using Shibboleth Attribute Authority Service (SAAS) [57] and Grid middleware. The EGEE project in cooperation with NREN community has developed and deployed two credential mapping services bridging campus AAI and Grid access control infrastructure: the Short Lived Credentials Service (SLCS) [58] and VOMS Attributes from Shibboleth (VASH) service [59].

The SLCS is a service that issues short lived X.509 credentials based upon successful authentication at a user's institutional Identity Provider. This service operates as an X.509 certificate factory that issues a user a short lived X.509n certificate that can be used for accessing grid resources as a normal identity certificate that typically has longer life time. The certificate is accepted by the International Grid Trust Federation (IGTF) managing CA policies interoperability, and therefore can be used with most existing Grid infrastructures.

Using SLCS simplifies user credentials management and decrease risk of compromising long-lived user credentials and the potential impact of their misuse.

The SLCS provides also a bridge between Grid security infrastructure and widely implemented Shibboleth based campus Authentication and Authorisation Infrastructure [57] that actually doesn't require possession of the X.509 certificates by users. In this way the SLCS allows using user campus/universities credentials to access Grid.

VOMS attributes from Shibboleth (VASH)

The VASH (VOMS Attributes from Shibboleth) [59] service consolidates authentication and authorization information from a user's institutional Identity Provider (IdP) and a user Virtual Organization (VO). The credentials of both the IdP and the VO can be used for managing access to Grid resources.

VASH pushes a (sub-) set of the user's Shibboleth attributes to the VOMS server. On the VOMS server these attributes are stored within the existing VO user profile. The decisions to push, and later to maintain up-to-date Shibboleth attributes on the VOMS server are proactive, which means they are fully user driven.

The VASH service transfers a user's Shibboleth attributes to VOMS upon request of the user. There is one instance of this service for every Shibboleth federation and every VO. The main advantages of the current VASH design are:

- The X509 user certificate remains unchanged (no attributes in the X.509 extensions).
- The Shibboleth credentials are transparently integrated into VOMS ACs. Using the credentials for authorization in grid requires minimal changes at the grid service itself. In particular, no Shibboleth specific code has to be added to the grid service and no changes are needed at the Shibboleth IdP.

[11] http://www.educause.edu/

- VOMS servers do not have to be changed and do not need to become Shibboleth Service Providers (contrary to a model, in which VOMS would pull the Shibboleth user attributes).
- The VO user registration does not have to be changed. This is important, as some VOs have implemented their own registration software and procedures (e.g. VOMRS).
- The administrative domains of VOMS and Shibboleth are fully decoupled, i.e. the VOMS administrator manages only VO specific information and the Shibboleth IdP administrator manages only campus relevant information.
- For the Shibboleth IdPs this service is just another Shibboleth Service Provider (web resource).
- The user itself as registered in the VOMS server manages the mapping of the user identity between the Shibboleth IdP and the DN of the user's X.509 certificate. Therefore, the administrative burden for the VASH administrator is kept to a minimum. In addition, the new service becomes a repository where the mapping of the Shibboleth IdP and the DN is stored and made available to other (properly authorized) services.

6 Virtual Organisations in Grid

This section briefly presents the Virtual Organisation (VO) concept in Grid/OGSA and describes widely used VO management tool Virtual Organisations Membership Service (VOMS).

Grid resource and service virtualisation, together with provisioning, are two key concepts in the OGSA [3]. OGSA Security is built around the Virtual Organisation (VO) concept and targeted for the enforcement of the security policies within a VO as an association of users and resources. VO provides a framework for inter-organisational and inter-domain trust management. When registered with a VO, an external user will be able to access to the enterprise/provider internal network based on his/her VO membership and relationship between the VO and the enterprise or provider. Access is typically enforced by a firewall, VPN gateway or Application Level gateway.

VO is actually a form of the user and resource federation that can be dynamic by its nature.

Typically, the VOs security services are created on the basis of the VO members' security services and may interact with them. A VO may run its own security services. Examples of such services are: credential validation services, trust services, authorisation services, and attributes services. But still many other services will remain in member domains and their authority need to be translated into VO domain through established trust relations and shared/translated semantics.

Figure 4 below illustrates conceptual model for VO security services and their interaction with VO members' security services. VO may run own security services, e.g. credential validation service, trust service, authorisation service, and attributes service as shown on the picture. But still many other services will remain in member domains and their authority need to be translated into VO domain through established trust relations and shared/translated semantics.

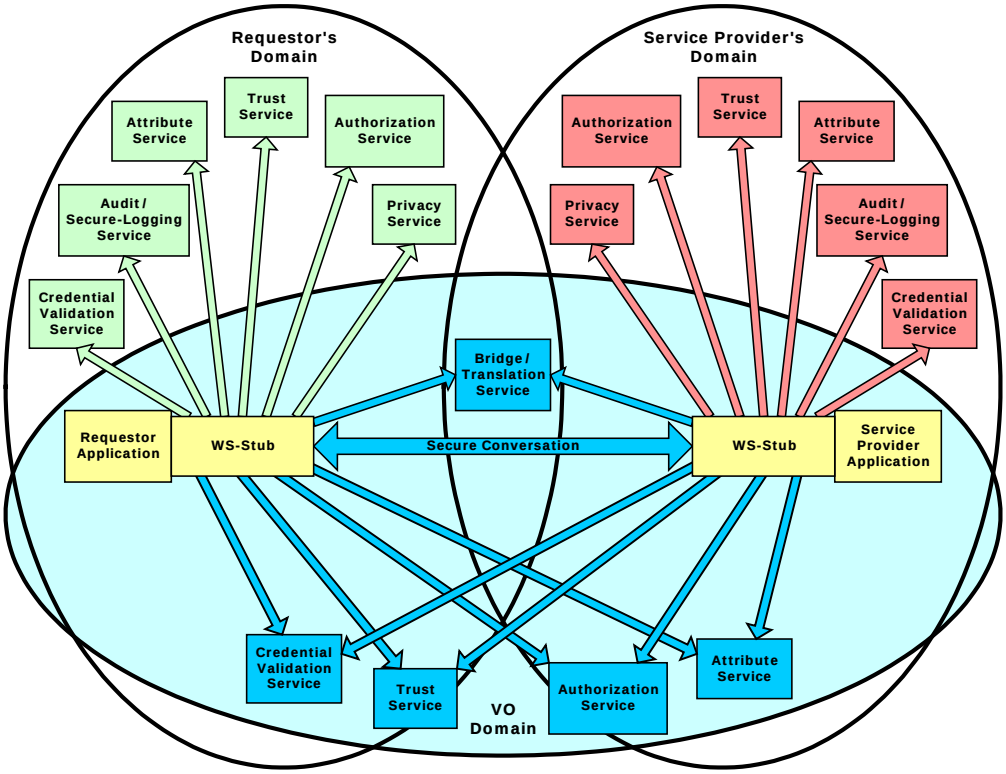


Figure 4: Security services in a virtual organization setting [3]

6.1 The Virtual Organization Membership Service (VOMS)

The Virtual Organization Membership Service (VOMS) has been developed in the framework of EU project EGEE and a part of international cooperation between EGEE and OSG Consortium in US [41]. VOMS goal is to solve the problems of granting users authorization to access the resources at VO level, providing support for group membership, roles and capabilities.

In VOMS design, a VO is represented as a complex, hierarchical structure with groups and subgroups what is required to clearly separate VO users according to their tasks and home institutions. From an administrative point of view, the management of each group can be independently delegated to different administrators. The administrators of each group can create subgroups and grant administration rights to these subgroups; they cannot modify memberships in any other subgroup. A group is basically a set of users, which may also contain other groups. In general a user can be a member of any number of groups contained in the VO hierarchy.

Every user in a VO is characterized by the set of his attributes defining their group membership, roles and capabilities in the scope of the VO that can be expressed in a form of 3-tuples (group, role, capability). The combination of all 3-tuple values forms unique attribute, the so-called “Fully Qualified Attribute Name" (FQAN). In general an FQAN has the following form [42]:

```
/VO[/group[/subgroup(s)]][/Role=role][/Capability=cap]
```

For example, the FQAN corresponding to the role Administrator in the group Nerds of the VO campus.example.org is:

```
/campus.example.org/Nerds/Role=Administrator
```

The VOMS system consists of the following parts (see Figure 5) [41]:

User server: receives requests from client and returns information about the user.

User client: contacts the server presenting a user's certificate and obtains a list of groups, roles and capabilities of the user.

Administration client: used by VO administrator to add users, create new groups, change roles.

Administration server: accept the request from the admin client and updates the database.

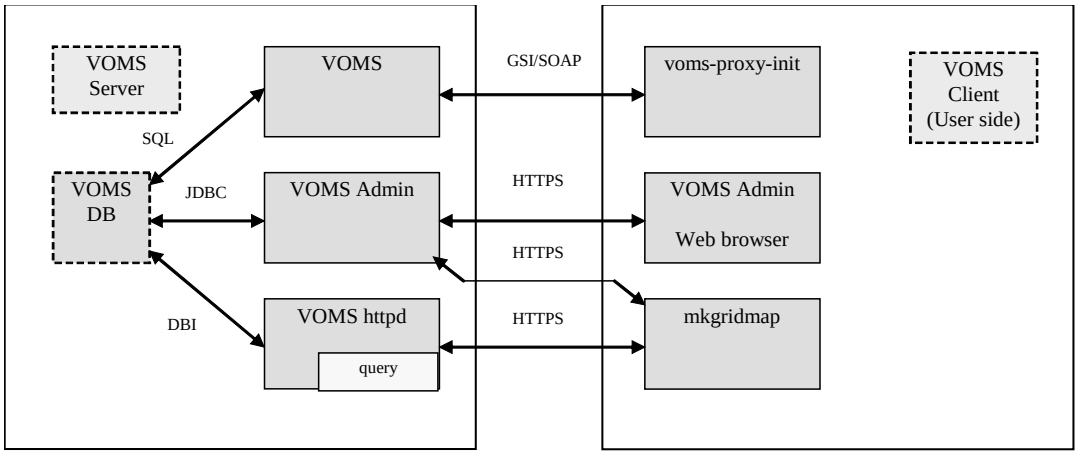


Figure 5. VOMS System Architecture.

In Grid user or service request authorisation is based on user VO credentials or attributes that are defined by the VOMS Attribute Certificate. In the basic scenario, user obtains VOMS Certificate via User (VOMS) client, embed it into their Proxy Certificate (ProxyCert) [40] and send it together with the Service Request to the Grid Service or Resource where it is used for user authorisation. The procedure includes the following steps (see Figure 6):

1. The user and the VOMS Server authenticate each other using their certificates (via the standard Globus API);
2. The user sends a signed request to the VOMS Server;
3. The VOMS Server verifies the user's identity and checks the syntactic correctness of the request;
4. The VOMS Server sends back to the user the required information (signed by itself);
5. The user checks the validity of the information received;
6. The user optionally repeats this process for other VOMS's to collect membership information in other VO's;

7. The user creates the proxy certificate containing all the information received from the VOMS Server in a (non-critical) extension;
8. The user may add user-supplied authentication information (e.g., Kerberos tickets).

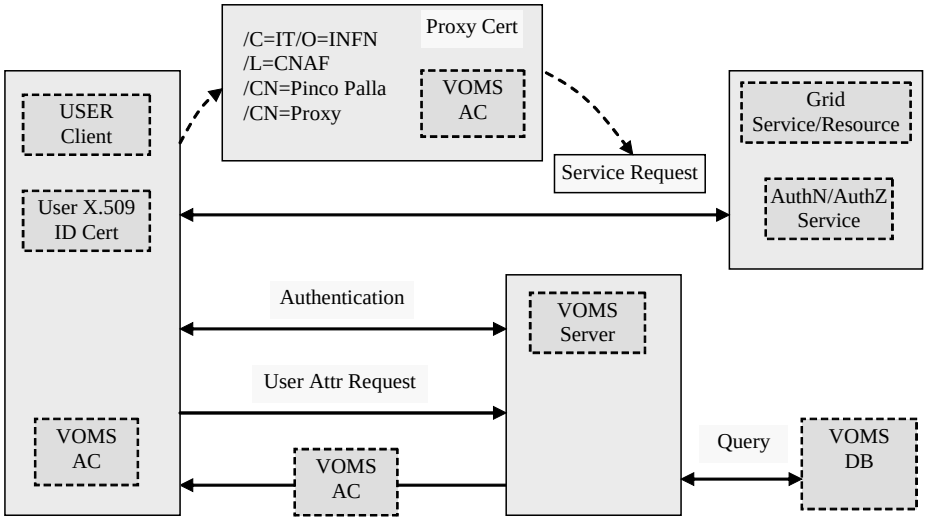


Figure 6. Interaction between VOMS server and user client when obtaining VOMS Attribute Certificate that is further presented in the service request by user.

VOMS server returns user X.509 Attribute Certificate (AC) that contains information about user VO and optionally about user group and role. Future version of VOMS server is claimed to support SAML Attribute assertion format. At the Resource, the authorization information provided by VOMS needs to be extracted from the user's proxy certificate and evaluated against the local access control policies in order to make the authorization decision.

The Administration Server communicates over SOAP protocol and can be easily integrated into WS-based Globus Toolkit. It consists of five sets of routines grouped into services: (1) the Core that provides the basic functionality for the clients; (2) the Admin that provides the methods to administrate the VOMS database; (3) the History that provides the logging and auditing functionality (the database scheme provides full audit records for every changes); (4) the Request that provides an integrated request handling mechanism for new users and for other changes; and (5) the Compatibility, which provides a simple access to the user list for the `mkgridmap` utility. Two administrative interfaces (web and command line) are available.

VOMS infrastructure suggests that VO may have few VOMS servers with synchronised membership databases, however one VOMS server can serve multiple VO's. Central/main membership database is maintained by a VO and must contain information/attributes for all registered VO members. Currently, only user attributes are stored in VOMS database. There is ongoing discussion about providing VO credentials to the resources as well.

User Server and Clients (Core VOMS System) is developed by INFN, Administration Server and Client (Admin Interface) is developed at CERN. VOMS is available as open source software under EGEE license.

6.2 VO Management in the EGEE Project

The current VO management practice in the LCG and EGEE projects, provide a good example of the instant implementation of the VO concept. The approach is however project based and project oriented. The Major VO membership management tool is the VOMS, which supports user registration procedures with the VOMS Admin server automated workflow.

Within EGEE, user communities are organised on the basis of Virtual Organisations (VO). An individual can only use resources if they are a member of a virtual organisation. It is the VO that is enabled to use a resource and restrictions can be placed on which individuals or groups/roles within a VO can gain access to a resource. Many resources of varying types can be hosted by each Resource Centre (RC). Not all VOs are enabled on all resources at all Resource Centres. Indeed, it is expected that a VO will bring resources from within its own community to connect to and form part of the EGEE infrastructure, thus increasing the total pool of resources available. Resource Centres and regions are encouraged to allocate a small percentage of their resources to new VOs to help boost the community onto the e-Infrastructure.

The following documents define VO management framework in LCG/EGEE:

VO Registration Procedure [60] and Virtual Organisation Registration Security Policy [61] - describe steps a new Virtual Organisation (VO) should take and information provide in order to be configured and get integrated in the LCG/EGEE infrastructure;

VO Operations Policy [62], Virtual Organisation Membership Management Policy [63] – provide a set of requirements which the VO managers need to comply in managing the VOs and their membership;

VO Portal Policy [64] – specifies policies applied to different portals operated by VOs participating in the Grid. In particular the document defines four classes of portals depending on the provided functionality for job submission and corresponding authentication strength requirements.

Grid Acceptable Use Policy (AUP) [65] – defines a set of responsibilities placed on the members of the VO and the VO as a whole through its managers. It aims to ensure that all Grid participants have sufficient information to properly fulfil their roles with respect to interactions with a VO. All Grid/VO users shall agree at AUP and their agreement is filed at the stage of the user registration with VO.

There are a number of steps a Virtual Organisation (VO) should take in order to get registered, configured and integrated in the LCG/EGEE infrastructure described in the VO Registration procedure and VO Security Policy:

1. **Naming the VO.** VO name should use the DNS naming style and resemble project and/or team. It should use the dedicated domain name and host certificates in order to prepare a properly managed system environment for VO-related data, scripts, web pages and transactions.

2. **VO integration into existing EGEE infrastructure** should be requested from the designated EGEE body Operations Advisory Group (OAG) [??Operations Advisory Group (OAG) Procedures and policy report - https://edms.cern.ch/file/724636/5/EGEE-II-DSA1.2-724636-v4.0.pdf] which will estimate required resources (computing power and load, storage size, etc.) and propose possible VO applications hosting and resources allocation

between candidate hosting sites and Grid Regional Centers (RC) and also fix requirement to RC to participate in the VO. As a result of this stage a VO manager is appointed and a CIC (Core Infrastructure Centre) appointed to provide VO user management service to the new VO.

3. **Setting-up a VO.** The VO management selects a site where to run the VO database (VODB) server and the Registration service/database (where the acceptance of the Grid Usage Rules by the user is registered). There can be few options for particular implementations of the VO services.

4. **Populating a VO.** Candidate entries in the VODB are passed through successful Registration process and Registration database additions. Suggested mechanisms to bootstrap and update a VODB depends on the selected technology and may be use LDAP based solution or integrated Registration and VODB solution based on VOMS. As soon as a VO is configured, the VODB contents must be propagated to the Grid sites in order to be matched to the users' credentials at job submission time.

5. **Organising support structure for the VO.** This requires designated group of people to manage VO procedures both registration and user support, including VO-wide Security Incident response. A VO Support Manager is responsible for building this structure and becomes a member of the EGEE Support Task Force.

The need for such strict procedures is motivated by the distributed character of the Grid resources and potentially unlimited possibility to use (and misuse) these resources.

7 Suggested Research Areas for Grid Security Services Architecture

This section discusses possible research areas in the Grid security that can contribute to the definition of more consistent GSSA. In particular, this includes but not limited to defining Complex Resource Provisioning (CRP) model to provide basis for Grid security services integration with the upper layer scientific workflow, provisioning and authorisation session management, defining mechanisms to express and communicate security context between services and domains, user centric and user controlled security services environment, secure invocation of a remote virtualised execution environment. Initial analysis of these problems and solutions was proposed in [66].

7.1 Complex Resource Provisioning Model

The whole lifecycle of the Grid resources provisioned on-demand can be abstracted to the common Complex Resource Provisioning (CRP) model [67]. Such abstraction can provide a basis for defining GSSA that should answer the major Grid operational models.

A typical on-demand resource provisioning process includes four major stages, as follows:

(1) resource reservation;

(2) deployment (or activation);

(3) resource access/consumption;

(4) resource de-commissioning after it was used.

In its own turn, the reservation stage (1) typically includes three basic steps:

(a) resource lookup;

(b) complex resource composition (including alternatives), and

(c) reservation of individual resources.

It may be also observed that for long running scientific experiments (months and years) there may be a need to define another CRP stage “(5) resource relocation” that could include combination of all 4 basic stages (1) - (4) starting from de-commissioning the resource that should be relocated, e.g. changing lightpath, or moving jobs/experiment to another Data Center. We assume that in case of relocation the general security context, and reservation ID in particular, should be inherited from the initial reservation. In the current applications and experiments this problem is reduced to either moving virtual execution environment image or just moving data. However, if a relocation is required for multiple involved machines or more security restriction are applied to data, this case will need more precise definition of the security model and related procedures.

The reservation stage may require the execution of complex procedures that may also request individual resources authorisation. This process can be controlled by the central advance reservation system or meta-scheduling system and driven by the provisioning workflow and related security policy. At the deployment stage the reserved resources are typically bound to the reservation ID, which we will refer to as the Global Reservation Identifier (GRI).

The de-commissioning stage is considered as an important stage in the whole resource provisioning workflow from the provider point of view and may include such important actions as global provisioning/access session termination and user/process logout, log information sealing, accounting and billing which are again currently considered as separates actions outside of the general provisioning workflow.

The rationale behind defining different CRP workflow stages is that they may require and can use different security models for policy enforcement, trust and security context management, but still may need to use common dynamic security context.

Defining and applying CRP model for Grid specific resource provisioning will aim two goals: building consistent security architecture that will ensure integrity of the whole resource life-cycle, and provide a better formalised framework for Grid services integration into more general e-Science workflow.

7.2 Authorisation Session Management

Authorisation and/or provisioning session management is considered as an important function when applying access control to managing stateful processes and resources.

The security context and session management are widely used in modern web based applications what can provide a good base for developing similar solutions for Grids that will address such

specific requirements as delegation (currently solved with the Proxy certificate), supporting policy obligations (addressed in XACML-Grid profile [49] and discussed below), and others.

The analysis of the CRP sessions properties was made in the paper [68] that suggested using for this different types of session based credentials: pilot tokens for reservation session, access tokens for access, authorisation ticket for extended context management.

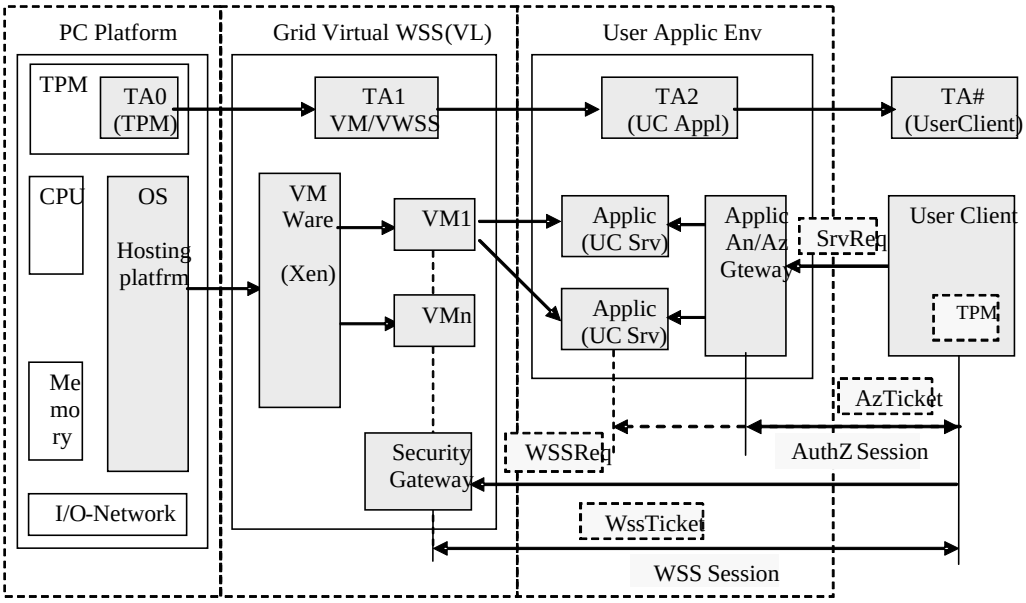
authors addressed this problem in developing AuthZ service for Grid based collaborative applications and for NRP [27, 28] by using AuthZ tickets and token that when used together can address both extended AuthZ context management and performance issues. The proposed and currently implemented in the GAAA-TK solution supports two types of AuthZ tickets: proprietary, and based on the SAML 2.0 Assertions format [29] and SAML 2.0 Profile of XACML [30].

7.3 Extending User Controlled Security Domain in Virtualised Workspace Service (VWSS)

Modern paradigm of remote distributed services and digital content providing makes security and trust relations between User and Provider more complex. A user and a service provider are two actors concerned with own Data/Content security and each other System/Platform trustworthiness

Figure 7 depicts the proposed in authors work [69] the 3-layer VWSS-UC environment for running user tasks and applications that provides integral protection of user tasks/applications at all three layers. The three layers include: a TCG based computing/hosting facility, a Grid based Virtual Workspace Service, and a User Application Environment. The solution extends the original Virtual WorkSpace Service (VWSS) concept [70] and is capable of scaling over multiple administrative and trust domain and allows for running multistage user tasks or complex resource provisioning.

A virtual workspace is created after a user request is sent to the VWSS security gateway, which checks user credentials and deploys the VM based workspace with characteristics that meet the request's requirements. Such a virtual workspace creates a trusted environment where users can run their tasks or applications. User applications and/or tasks are protected by basic security services to avoid potential data compromise or interruptions. This is first of all achieved by user Authentication (AuhtN) and Authorisation (AuthZ) provided by the Application AuthN/AuthZ Gateway. In the case of complex/multi-component services, their combinations should be secured through the applications level security context management.



- Trust Anchors: T0 (TPM) – TA1 (VM/VWSS) – TA2 (Application) – TA# (User)
- WVSS session and Application AuthZ sessions

Figure 7. Three-layer Security Model of the VWSS-UC.

For the dynamic security context management, the VWSS-UC distinguishes between a WSS session and an application/service AuthZ session that is related to the user task or application. WSS session may have wider security context but still both of the session types are based on the positive authorisation decision and will require a similar AuthZ context management. WSS sessions that includes VWSS request may also need to incorporate a negotiation stage and possibly want to verify the platform security configuration and/or integrity, which could be achieved through the TPM based mechanisms.

In the proposed architecture, the TPM with its hardware-based secure ID allows for “bootstrapping” a chain of trust to the TMP and hardware platform. This creates a continuous chain of trust from the user to the workspace environment and hosting platform: TA#-TA2-TA1-TA0., where TAn – are trust anchors as shown on the picture.

7.4 Policy Obligations – Bridging Two Security Concepts

In many Grid applications, policies may specify actions that must be performed either instead of or in addition to the policy decision. In the XACML specification [34], obligations are defined as actions that must be performed in conjunction with policy evaluation on a positive or negative decision. In this way and when using together with gLExec [53], policy obligations can be used for defining actions that will be performed by the gLExec when submitting Grid jobs to the protected execution environment.

Obligations are included into the policy definition and returned by PDP to PEP which in its turn should take actions as prescribed in the obligation instructions or statements. In the context of the GSSA,

obligations provide an important mechanism for policy decision enforcement in the provisioned Grid resources, in particular, mapping global user ID/account to local accounts or groups, assigning quotas, usage limits, etc.

The proposed obligations handling model is described in details in [50] and allows two types of obligations execution: at the time of receiving obligations from the PDP and at the later time when accessing a resource or performing an authorised action. The latter can be achieved by using AuthZ tickets that hold obligations together with AuthZ decisions.

7.5 Using Identity Based Cryptography for building Dynamic Security Associations

Trust management is an important issue and a problem in Grid security. It would not be a complete overview of possible research areas in developing a consistent GSSA if we not mention the Identity Based Cryptography (IBC) [71]. The IBC allows using recipient's public credentials to generate the encryption key when sending a message to the recipient, and the user can request the local IBC Key Generation Server (KGS) to obtain own private key.

IBC in application to Grid has been a topic for many research projects and papers in the academic community [71, 72] but it is still less known for Grid practitioners. We expect that IBC can provide a simple way of building dynamic interdomain trust relations or distributing security context between domains that doesn't have direct trust relations. Such an approach will use pre-configured IBC KGS to distribute security information between domains, and in this way "exchange" the IBC based intra-domain trust infrastructure for simpler trust and key management in dynamic multidomain applications.

8 References

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

[2] Foster, I., Kesselman, C., Nick, J. and Tuecke, S. The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration. Globus Project, 2002. [Online]. Available: http://www.globus.org/research/papers/ogsa.pdf.

[3] GFD.80 "The Open Grid Services Architecture, Version 1.5", I. Foster, H. Kishimoto, A. Savva, D. Berry, A. Grimshaw, B. Horn, F. Maciel, F. Siebenlist, R. Subramaniam, J. Treadwell, J. Von Reich. Open Grid Forum, September 5, 2006.

[4] GFD.113 "Technical Strategy for the Open Grid Forum 2007-2010", D. Snelling, C. Kantarjiev. Open Grid Forum, August 7, 2007.

[5] GFD.150 "Using Clouds to Provide Grids Higher-Levels of Abstraction and Explicit Support for Usage Modes. Version: 1.0." Shantenu, J., A. Merzky, G. Fox, Open Grid Forum, April 29, 2009.

[6] GFD.138 "OGSA Basic Security Profile 2.0", D. Snelling, D. Merrill, A. Savva, Open Grid Forum, July 28, 2008.

[7] GFD.132 "Secure Communication Profile 1.0", D. Merrill. Open Grid Forum, June 13, 2008.

[8] GFD.131 "Secure Addressing Profile 1.0", D. Merrill. Open Grid Forum, June 13, 2008.

[9] ITU-T Rec. X.800 Security Architecture for Open Systems Interconnection for CCITT applications. ITU-T (CCITT) Recommendation, 1991

[10] Web Services Architecture. W3C Working Group Note 11 February 2004. [Online]. http://www.w3.org/TR/ws-arch/

[11] Web Services Security Roadmap (2002). [Online]. Available: http://www.ibm.com/developerworks/library/ specification/ws-secmap/
[12] Web Services Resource Framework (WSRF), Primer v1.2, Committee Draft 02, 23 May 2006. - http://docs.oasis-open.org/wsrf/wsrf-primer-1.2-primer-cd-02.pdf
[13] Web Services Resource Transfer (WS-RT) Version 1.0, August 2006. http://www.ibm.com/developerworks/ webservices/library/specification/ws-wsrt/
[14] Web Services for Management, DMTF Standard, 2 December 2008. [Online]. Available: http://www.dmtf.org/standards/published_documents/DSP0226_1.0.0.pdf
[15] Anderson, J., Computer Security Technology Planning Study, ESD-TR-73-51, ESD/AFSC, Hanscom AFB, Bedford, MA 01731 (Oct. 1972) [NTIS AD-758 206]. http://csrc.nist.gov/publications/history/ande72.pdf
[16] David E. Bell and Leonard La Padula, Secure Computer System: Unified Exposition and Multics Interpretation, ESD-TR-75-306, ESD/AFSC, Hanscom AFB, Bedford, MA 01731 (1975) [DTIC AD-A023588] http://csrc.nist.gov/publications/history/bell76.pdf
[17] Biba, K. J. "Integrity Considerations for Secure Computer Systems", MTR-3153, The Mitre Corporation, April 1977.
[18] Anderson, R., F. Stajano, J. Lee, Security Policies. [Online]. Available: http://www.cl.cam.ac.uk/~rja14/ Papers/security-policies.pdf
[19] Trusted Computing Group (TCG). [Online]. Available: https://www.trustedcomputinggroup.org/home
[20] TCG Infrastructure Working Group Reference Architecture for Interoperability. Specification Version 1.0. 16 Jun. 2005. http://www.trustedcomputinggroup.org/ specs/IWG/IWG_Architecture_v1_0_r1.pdf
[21] Trusted Platform Modules Strengthen User and Platform Authenticity. TCG Whitepaper, January 2005. [Online]. Available: https://www.trustedcomputinggroup.org/specs/TPM/ Whitepaper_TPMs_Strengthen_User_and_Platform_Authenticity_Final_1_0.pdf
[22] TNC Architecture for Interoperability. Specification Version 1.1, 1 May 2006. [Online]. Available: https://www.trustedcomputinggroup.org/specs/TNC/TNC_Architecture_v1_1_r2.pdf
[23] Nadalin, A., C. Kaler, P. Hallam-Baker and R. Monzillo (ed.): Web Services Security: SOAP Message Security 1.0 (WS-Security 2004), OASIS Standard, 200401, March 2004. http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-messagesecurity-1.0.pdf
[24] XML Signature Syntax and Processing (Second Edition), W3C Recommendation 10 June 2008. [Online] Available from: http://www.w3.org/TR/xmldsig-core/
[25] XML Encryption Syntax and Processing, W3C Recommendation 10 December 2002. [Online] Available from: http://www.w3.org/TR/xmlenc-core/
[26] Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0, OASIS Standard, 15 March 2005. [Online]. http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf
[27] Web Services Policy 1.2 - Framework (WS-Policy), W3C Member Submission 25 April 2006. [Online] Available from: http://www.w3.org/Submission/WS-Policy/
[28] Nadalin, A., M. Goodner, A. Barbir, H. Granqvist (ed.): WSSecurityPolicy 1.2. Oasis Standard, 1 July 2007. http://docs.oasis-open.org/wssx/ws-securitypolicy/200702/ws-securitypolicy-1.2-spec-os.pdf
[29] Vedamuthu, A., D. Orchard, F. Hirsch, M. Hondo, P. Yendluri, T. Boubez, Ü. Yalçinalp (eds.): Web Services Policy 1.5 – Attachment. W3C Candidate Recommendation 05 June 2007. http://www.w3.org/TR/2007/CRws-policy-attach-20070605
[30] WS-Trust 1.3, OASIS Standard, 19 March 2007. [Online] Available from: http://docs.oasis-open.org/ws-sx/ws-trust/200512/ws-trust-1.3-os.pdf
[31] WS-SecureConversation 1.3, OASIS Standard, 1 March 2007. [Online] Available from: http://docs.oasis-open.org/ws-sx/ws-secureconversation/200512/ws-secureconversation-1.3-os.html
[32] Web Services Federation Language (WS-Federation), Version 1.1. December 2006. [Online] Available from: http://specs.xmlsoap.org/ws/2006/12/federation/ws-federation.pdf
[33] The Liberty Alliance Project. http://www.projectliberty.org/
[34] Godik, S. et al, “eXtensible Access Control Markup Language (XACML) Version 2.0”, OASIS Working Draft 04, 6 December 2004. - http://docs.oasis-open.org/ xacml/access_control-xacml-2_0-core-spec-cd-04.pdf
[35] K. Ballinger, D. Ehnebuske, C. Ferris, M. Gudgin, C.K. Liu, M. Nottingham, and P. Yendluri (ed.): Basic Profile Version 1.1, Web Services Interoperability Organization Final Material, 24 August 2004. http://www.ws-i.org/Profiles/BasicProfile-1.1.html

[36] Barbir, A., M. Gudgin, M. McIntosh, and K.S. Morrison (ed.): Basic Security Profile Version 1.0, Web Services Interoperability Organization, Working Group Draft, 17 August 2006. http://www.ws-i.org/Profiles/BasicSecurityProfile-1.0-2006-08-17.html

[37] M. Gudgin and Marc Hadley (ed.), Web Services Addressing 1.0 - Core, W3C Candidate Recommendation 9 May 2006, http://www.w3.org/TR/2006/REC-wsaddr-core-20060509

[38] Housley, R., et al. (ed.) Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile, RFC 3280, 2002.

[39] The Kerberos Network Authentication Service (V5), RFC 4120. July 2005. http://www.ietf.org/rfc/rfc4120.txt

[40] Tuecke, S., et al. Internet X.509 Public Key Infrastructure (PKI) Proxy Certificate Profile, RFC 3820, 2004

[41] VOMS Architecture v1.1. [Online]. Available from: http://grid-auth.infn.it/docs/VOMS-v1_1.pdf, February 2003.

[42] Ciaschini, V., V. Venturi, A. Ceccanti, The VOMS Attribute Certificate Format. Open Grid Forum Draft, 2009. [Online]. Available from: http://www.ogf.org/Public_Comment_Docs/Documents/2009-06/VOMSACv10-editor-1.doc.pdf

[43] Venturi, V., M. Riedel, et al, Using SAML-Based VOMS for Authorization within Web Services-Based UNICORE Grids, Lecture Notes in Computer Science, Springer Berlin/Heidelberg, ISBN 978-3-540-78472-2. Pages 112-120.

[44] GFD-C.125 "Grid Certificate Profile," Groep, D., et al, Open Grid Forum, 31 March 2008 [Online]. Available from: http://www.ogf.org/documents/GFD.125.pdf

[45] “Functional components of Grid Service Provider Authorisation Service Middleware” [49 - https://forge.gridforum.org/sf/go/doc15695?nav=1]

[46] “Use of WS-Trust and SAML to Access a Credential Validation Service (CVS)”. https://forge.gridforum.org/sf/go/doc15696?nav=1 [50]

[47] “Use of XACML Request Context to obtain authorisation decision”. https://forge.gridforum.org/sf/go/doc15694?nav=1

[48] “Use of SAML to retrieve Authorization Credentials”. https://forge.gridforum.org/sf/go/doc15173?nav=1

[49] An XACML Attribute and Obligation Profile for Authorization Interoperability in Grids. [Online] Available: https://edms.cern.ch/document/929867/1

[50] Demchenko, Y., C. de Laat, O. Koeroo, H. Sagehaug, Extending XACML Authorisation Model to Support Policy Obligations Handling in Distributed Applications, Proceedings of the 6th International Workshop on Middleware for Grid Computing (MGC 2008), December 1, 2008, Leuven, Belgium. ISBN:978-1-60558-365-5. [Online] Available from http://portal.acm.org/citation.cfm?id=1462704.1462709

[51] Pluggable GAAA-TK library. [Online]. Available: http://staff.science.uva.nl/~demch/projects/aaauthreach/index.html#aaauthreach

[52] Groep, D., O. Koeroo, G. Venekamp, "gLExec: gluing grid computing to the Unix world", Journal of Physics: Conference Series 119 (2008) 062032

[53] GLExec. [Online]. Available: https://www.nikhef.nl/pub/projects/grid/gridwiki/index.php/GLExec

[54] Site Access Control. [Online] https://www.nikhef.nl/pub/projects/grid/gridwiki/index.php/Site_Access_Control

[55] Argus, The EGEE Authorisation Framework. [Online]. Available from: https://twiki.cern.ch/twiki/bin/view/EGEE/AuthorizationFramework

[56] MyProxy Credentials Management Service. [Online]. Available from: http://grid.ncsa.illinois.edu/myproxy/

[57] Shibboleth Attribute Authority Service. [Online]. Available from: http://shibboleth.internet2.edu/

[58] Short Lived Credentials Service (SLCS). [Online]. Available from: http://www.switch.ch/grid/slcs/

[59] VOMS Attributes from Shibboleth (VASH) service. [Online]. Available from:

[60] VO Registration Procedure (https://edms.cern.ch/document/503245)

[61] Virtual Organisation Registration Security Policy. [Online]. Available from: https://edms.cern.ch/document/573348

[62] VO Operations Policy [Online]. Available from: https://edms.cern.ch/document/853968/

[63] Virtual Organisation Membership Management Policy. [Online]. Available from: https://edms.cern.ch/document/428034

[64] VO Portal Policy. [Online]. Available from: https://edms.cern.ch/document/972973

[65] Grid Acceptable Use Policy (AUP). [Online]. Available from: http://www.jspg.org/wiki/Grid_Acceptable_Use_Policy

[66] Demchenko, Y., C. de Laat, O. Koeroo, D. Groep, Re-thinking Grid Security Architecture. Proceedings of IEEE Fourth eScience 2008 Conference, December 7–12, 2008, Indianapolis, USA. Pp. 79-86. IEEE Computer Society Publishing. ISBN 978-0-7695-3535-7 / ISBN 978-1-4244-3380-3
[67] Demchenko Y, A. Wan, M. Cristea, C. de Laat, "Authorisation Infrastructure for On-Demand Network Resource Provisioning", The 9th IEEE/ACM International Conference on Grid Computing (Grid 2008), Tsukuba, Japan, Sept. 29 - Oct. 1, 2008.
[68] Demchenko, Y., C. de Laat, T. Denys, C. Toinard, Authorisation Session Management in On-Demand Resource Provisioning in Collaborative Applications. COLSEC2009 Workshop, The 2009 International Symposium on Collaborative Technologies and Systems (CTS 2009), May 18-22, 2009, Baltimore, Maryland, USA.
[69] Demchenko Y., F. Siebenlist, L. Gommans, C. de Laat, D. Groep, O. Koeroo. Security and Dynamics in Customer Controlled Virtual Workspace Organisation, In Proc. HPDC2007 Conference, Monterey Bay California, June 27-29, 2007.
[70] Virtual Workspaces. [Online]. Available: http://workspace.globus.org/index.html
[71] A. Shamir. Identity-based cryptosystems and signature schemes. In G.R. Blakley and D. Chaum, editors, Advances in Cryptology - Proceedings of CRYPTO'84, pages 47{53. Springer-Verlag LNCS 196, 1985.
[72] Exploiting Hierarchical Identity-Based Encryption for Access Control to Pervasive Computing Information. [Online]. Available: http://stinet.dtic.mil/cgi-bin/GetTRDoc?AD=ADA457869 &Location=U2&doc=GetTRDoc.pdf