Security in Distributed, Grid, and Pervasive Computing
Yang Xiao,(Eds.) pp. – - –

Chapter 12

GRID SECURITY ARCHITECTURE: Requirements,fundamentals, standards, and models

José L.Vivas
Departamento de Lenguajes y Ciencias de la Comunicación
University of Málaga
E-mail: jlvivas@lcc.uma.es

Javier López
Departamento de Lenguajes y Ciencias de la Comunicación
University of Málaga
E-mail: jlm@lcc.uma.es

José A. Montenegro
Departamento de Lenguajes y Ciencias de la Comunicación
University of Málaga
E-mail: monte@lcc.uma.es

1 Abstract

Grid computing is concerned with the creation of distributed virtual organizations across multiple control domains to enable the sharing of diverse remote resources. Due to its multi-institutional nature, securing the Grid is one of the main challenges in Grid computing. In this paper we provide an overview of the Grid security fundamentals, standards, requirements, models, architecture, and use patterns. We survey the major security challenges and requirements for Grids, the main Grid security models that address these requirements, current Grid security architectures, emerging Grid security standards and standard bodies, the convergence of Grid and Web services, and the emerging Enterprise Grids.

2 Introduction

A Grid may be defined as a collection of computing resources distributed over a local or wide area network, and available to an end user as a single large computing system. Originally, the Grid focused on the areas of computing power, data access, and storage resources. It was intended for large-scale and distributed scientific computing that required efficient and dynamically determined access to large amounts of data and computational resources that are distributed along several independently administered networks. However, the use of Grid computing has been expanding lately to include deployment of Grid technologies within the context of business [46], which significantly widens the range of applicability of Grid technologies. Standard interfaces for business services have also been leveraged by Grid computing. Grid computing has been targeting such differing areas as finance, medicine, decision-making, collaborative design, and utility computing. The focus today is on coordinated resource sharing distributed across virtual organizations. However, shareable on-demand resources in commercial applications greatly complicates resource sharing and introduce new challenges related to federated security and integration.

Fundamental to Grid computing is the notion of scalable virtual organization (VO) [1], which may be defined as a dynamic set of individuals and/or institutions that share resources and services according to a set of well-defined rules and policies. The Grid vision is to provide unlimited power and information access to end users through the creation of dynamic VOs for secure and agile resource sharing among individuals and organizations. VOs may span several administrative domains, each one with its own security requirements and policies. Hence, interoperability among the multiple domains involved in a VO requires that VO-defined policies comply with domain-level policies, while at the same time maintaining a clear separation among virtual and real protection domains in a context in which they may superpose and intersect each other in a variety of ways.

Security has been a central issue in Grid computing from the outset, and has been regarded as the most significant challenge for Grid computing [6]. This is particularly true for Enterprise Grids. Significant compromises in security might be the result of an inadequate understanding of the security implications of a Grid. The security requirements and policies are determined largely by the architectures developed for these types of applications, which are distinguished from client-server architectures by the fact that Grid environments assume a dynamic and simultaneous use of a large

number of resources from a number of administrative domains. Although the intention has been from the outset to use available security mechanisms as much as possible, this requirement could not be met by mechanisms that were devised largely for insulating and protecting networks from their environment, as in intranets and virtual private networks. As a result, novel security technologies have been evolving all the time within the Grid community, including solutions for the management of credentials and policies, new resource management protocols for co-allocation of multiple resources and for secure remote access to data and computing resources, and new information query protocols and data management services [7].

The requirements of Grid computing are to a great extent in contradiction with the security policies and mechanisms related to administrative domains, since the objective of a Grid is basically to circumvent the barriers imposed by these mechanisms by the establishment, in an ad hoc manner and for any desired period of time, of a virtual domain emulating the behavior of real domains. In consequence, Grid computing has given rise to new security challenges, both for providers and for users, that could not be immediately met by available security technologies as the latter were intended to meet a set of requirements that were in many cases in contradiction to the requirements associated with Grid computing. In order to provide resources to non-local members, system administrators must accommodate mechanisms and policies that are not completely under their control and which force them to open some previously closed access points. Therefore, the task of Grid security engineering has been largely to reconciliate these antagonistic set of requirements, thus enabling components to be administered independently, according to local policies, and allowing users to achieve the desire level of quality of service regarding confidentiality, integrity and availability requirements.

A key challenge here is the assignment of users, resources, and organizations to a VO. Security issues related to this task include the specification of federation, delegation, and access control among the participants. A further requirement, which gives rise to new security problems for current administrative domains, is the need to have hundreds of processes in different domains collaborating with each other in order to carry out a particular computing task. This kind of computation requires the dynamic establishment of multiple trust and security relationships among processes, turning authentication, delegation and authorization into major challenges.

The presence of multiple administrators raises also many issues concerning accountability and responsibility. Other key issues concern interaction

with firewalls and the process of creation and destruction of VOs. The requirements associated with Grid computing can therefore not be met within the framework of client-server relationships with tight access control by individual domains.

Recently we have seen an evolution towards a Grid system architecture based on Web services concepts and technologies and message-level security [2]. Grid computing is rapidly turning into a multifaceted discipline driven by international bodies and research projects, attracting also the interest of commerce and industry. This development, together with the adoption of Web services, has exerted a great impact on its architecture, infrastructure, standards and protocols, and also produced a much more fragmented landscape [20]. As a result there is presently no broad consensus on which standards to follow and on the implementation of the architecture.

Enterprise Grid computing poses also new challenges and unique requirements. In Enterprise Grids typically a single organization is responsible for managing a shareable set of resources and composing higher-order services with value for the business. Those resources may be owned by several businesses, e.g. independent service providers or outsourcing services firm, with no geographic limitations. Unique security requirements are associated with Enterprise Grids because of the needs of organizational security, privacy, and regulatory compliance goals.

In this work we present the security requirements and challenges encountered in Grid environments. We provide an overview of the Grid security fundamentals, standards, requirements, models, architecture and use patterns, survey the major security challenges and requirements, the Grid security models addressing these requirements, current Grid security architectures, emerging Grid security standards and standard bodies, the current convergence of Grid and Web services, and the emerging Enterprise Grids. We focus mainly on the security model associated with the service-oriented Open Grid Services Architecture (OGSA) [24], and the OGSA suite of security services and components. It is our hope that this paper will give those with a background in computer security, but otherwise unacquainted with Grid computing, a good introduction the field. We concentrate on the high level aspects of Grid computing, and omit questions about mechanisms, technologies, and implementation.

The rest of this work is organized as follows. In Section 3 we give an overview of Grid security standards and corresponding standard bodies. Section 4 is dedicated to a description of use patterns, general requirements, assumptions, and challenges concerning Grid computing. Finally, In Section 5

we concentrate on the presentation of the most important Grid security models and architectures.

3 Grid Security Standards

The requirements concerning integration and interoperability in Grids call for an extensive use of standard interfaces. Standardization is a key to the realization of the Grid vision, enabling the portability, interoperability and reusability of components and systems, as well as discovery, access, allocation and monitoring of services and resources in Grid environments. By facilitating the adoption of good practices, it is also important for security in general.

In this chapter we present the most relevant standards bodies (Section 3.1) and standards (Section 3.2) related to Grids and Grid security. Since web service security standards are now an integral part of Grid computing, we dedicate Section 3.3 to a presentation of WS-Security standards. For more details see also [20].

3.1 Standard bodies

The main standard body for the Grid is the Global Grid Forum [40], which works together with industrial organizations and has a decisive impact over the definition of security requirements and the adoption of infrastructures. Other important standard-setting bodies in Grid computing are the World Wide Web Consortium, the Web Services Interoperability Organization, the Advancement of Structured Information Standards, and the Distributed Management Task Force. These and other relevant standard bodies are presented below.

Global Grid Forum. The GGF [40], was formed in 1998 and consists of community-initiated working groups developing best practices and specifications for Grid computing. GGF creates four types of documents: informational, experimental, community practice, and recommendations. Work is divided in seven areas, one of which is concerned with technical and operational security issues in Grid environments, including authentication, authorization, privacy, confidentiality, auditing, firewalls, trust establishment, policy establishment, scalability, and management. GGF drafts define the delegation protocol for remote creation of *X.509 Proxy Certificates and* GSS-

API [22] extensions for Grid computing. There are currently three groups working on security:

- *Open Grid Service Architecture Authorization* (OGSA AUTHZ-WG), whose objective is to define specifications to facilitate interoperability and plug-ability of authorization components in the OGSA framework.
- *Firewall Issues*(FI-RG).
- *Trusted Computing* (TC-RG), whose purpose is to evaluate how the capabilities of TC can be used in a Grid context.

OASIS. Founded in 1993, OASIS [41] is a not-for-profit global consortium that promotes standards for e-business, focusing primarily on higher-level functionality, including security, authentication, and reliable messaging. There is a committee dedicated to the development of security standards for e-business and Web services applications. OASIS is responsible for the WS-Security standard, recognized as the foundation for securing distributed applications and Web services.

World Wide Web Consortium. The W3C [42] is an international organization initiated in 1994 to develop Web standards and guidelines, and promote common and interoperable protocols. It created the first Web services specification in 2003, focusing on SOAP and the Web Services Description language (WSDL).

Distributed Management Task Force. The DTMF [43] is an industry-based organization founded in 1992 to develop management standards and interoperability for Enterprise and Internet environments. It formed an alliance with the GGF in 2003 in order to build a unified approach to the provisioning and sharing and management of Grid resources and technologies. Two working groups are dedicated to security issues.

- *Security Protection and Management (SPAM) Working Group.* The goal of this working group is to ease the manageability of heterogeneous security systems within an enterprise or service provider environment.
- *User and Security Working Group.* The objective of this working group is to provide a set of relationships between the representations of users, their credentials, privileges and permissions, and the resources and resource managers involved in security management.

Internet2. Internet2 [44] is a consortium of groups from academia, industry and government, formed in 1996 to develop and deploy advanced network applications and technologies for research and higher education. Several Internet2 working groups target Grid standards, e.g. the *Higher Education PKI Technical Activities Group*, *the Peer-to-Peer Working Group*, and the *Shibboleth* project. Internet2 is part of the *EDUCAUSE/Internet2 Computer and Network Security Task Force*, which promotes practices and solutions for the protection of information assets and critical infrastructures for higher education, and is advised by *SALSA*, an oversight group consisting of technical representatives from the higher education community. The *SALSA-NetAuth Working Group* deals with the data requirements and implementation, integration, and automation technologies associated with understanding and extending network security management.

Liberty Alliance. The Liberty Alliance [45] is an international alliance of companies, non-profit and government organizations formed in 2001 to develop an open standard for federated network identity that supports network devices and addresses technical, business, and policy challenges concerning identity and web services. It has developed the Identity Federation Framework, which enables identity federation and management.

Web Services Interoperability Organization. The WS-I [47] is an open industry organization that promotes Web services interoperability across platforms, operating systems and programming languages. WS-I provides guidance, recommended practices, and supporting resources. WS-I creates and supports generic protocols for the interoperable exchange of messages between Web services. There are currently six working groups, one of them dedicated to security, the Basic Security Profile Working Group, which is developing an interoperability profile dealing with transport security, SOAP messaging security and other security issues. A set of usage scenarios and related message exchange patterns is being developed by the Working Group. A Working Draft with interoperability and security recommendations was released in March 2006 [33].

Enterprise Grid Alliance. The EGA [46] consortium is an open, vendor-neutral organization formed to develop Enterprise Grid solutions and accelerate the deployment of Grid computing in enterprises. EGA promotes open, interoperable solutions, and best practices focusing exclusively on the

needs of enterprise users. The EGA is addressing requirements for deploying commercial applications in a Grid environment. Initial focus areas include reference models, provisioning, security and accounting. The EGA's Grid Security Working Group (EGA-GSWG) is dedicated to the identification of the unique security threats, issues and requirements associated with Enterprise Grid architectures and computing.

3.2 Grid Security Standards

The de-facto standard middleware for Grid computing is the Globus Toolkit (GT) [48], and for Grid security the GT's *Grid Security Infrastructure* (GSI) [49]. The Globus Toolkit is an open-source software that provides a set of services supporting collaboration across dynamic, multi-institutional virtual organizations. GSI was implemented by the Globus Toolkit, and uses X.509 identity and proxy certificates. GSI is based on standard technologies, such as TLS and secure Web Services specifications.

The most important Grid standard today is the *Open Grid Service Architectures* (OGSA) [50] presented below in Section 5.1. OGSA is promoted by the OGSA Working Group of the Global Grid Forum, created in September 2002 to draft specifications. The Globus Toolkit has adopted this standard in the latest versions.

The first instantiation of OGSA was the Open Grid Services Infrastructure, OGSI v1.0 [23] released in June 2003. OGSI is based on the concept of *Grid service*. Dissatisfaction with OGSI, which required modifications to standard WSDL, led to an effort to define an alternative infrastructure based on pure Web services specifications. On January 2004 the WS-Resource Framework (WSRF) [51] was announced. WSRF contains specifications for expressing the relationship between stateful resources and Web services. After revision, the final result was submitted to two OASIS technical committees, the WS-Resource Framework (WSRF) TC and the WS-Notification (WSN) TC. Several specifications were standardized by both committees.

Alternatives to the WSRF include:

- **Basic Profile from the WS-I**: the Basic Profile [33] contains guidelines for using Web service standards SOAP, WSDL, and UDDI.
- **Web Services Grid Application Framework**: the WS-GAP [15] proposes to extend basic Web services functionality in order to meet the needs of Grid applications; it uses the Web services standard WS-Context to make services stateful.

- **WS-I+** from the Open Middleware Infrastructure Institute (OMII): OMII [39] is an institute established by the UK e-Science Programme to act as a center for expertise in Grid middleware. The OMII specified a roadmap to allow the capture of generic middleware components from multiple projects in a way that facilitates interoperability with Grid Services standards and OGSA developments. WS-I+ [19] is a superset of WS-I's Web Services specifications, where the extra specifications are considered helpful in building e-Science Grids.

3.3 WS-Security

WS-Security [26] is a Web service standard initially released by Microsoft in October 2001. In April 2002 IBM and Microsoft released a joint "Security in a Web Services World" document [27]. This defined a security framework for Web Services, the first of which is WS-Security. Later specifications for Web Services security include *WS-Trust* [28], *WS-Policy* [29], *WS-SecureConversation* [31], *WS-Federation* [32], *WS-Privacy* (unpublished), and *WS-Authorization* (unpublished). In 2002 WS-Security specification was submitted to the OASIS standards body. A Web Services Security group was formed in OASIS in order to develop WS-Security as an OASIS standard. WS-security standards are now an integral part of Grid computing.

WS-Security is primarily for securing SOAP messages. It defines security tokens in SOAP messages and how they and other parts of a SOAP message can be encrypted and signed by XML Security specifications, i.e. *XML Signature* and *XML Encryption*. WS-Security includes specifications such as WS-Trust, WS-Policy, and WS-SecureConversation.

WS-security defines element names in order to package security tokens into SOAP messages. On top of it there is a conceptual model that abstracts different security technologies into "claims" and "tokens." A claim is a statement relating a subject with a property, e.g. an identity, and may be used for access control. A token is an XML representation of security information, e.g. a password, X.509 digital certificates, or a Kerberos ticket. Further specifications build on these concepts and shows how to apply for security token, how tokens are related to identity, and how to associate security information with a Web service.

Interoperability across domains with different security technologies are as important for Web services as for Grids, and similar solutions apply. WS-Security provides a level of abstraction for companies using different security

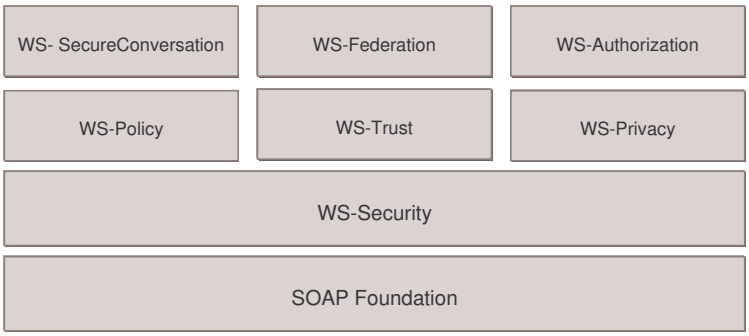

Figure 1: WS-Security model.

technologies to communicate securely using SOAP. In this way, existing or new security technologies and infrastructures can be used for both Web and Grid Services security.

The WS-Security model defines scenarios where the integrity and the confidentiality of SOAP messages are ensured while traversing intermediaries, which may themselves perform security functionality. Additional specifications, such as WS-Trust and WS-Policy, define how security tokens are issued.

The model for WS-security is shown in Figure 1. Each specification depends on its predecessors. SOAP, which is transport-independent, is at the base of the diagram. WS-Security is on top of SOAP. It provides a means for encrypting and signing portions of a SOAP message using XML Signature and XML Encryption, and for enclosing security tokens in a SOAP message. We shortly describe below the XML and Web service standards that are relevant for Grid security.

XML Signature. *XML Signature* (XML-SIG) [16] was the first XML security standard to reach recommendation status. XML Signature is a building block for WS-Security. It provides integrity for data, and is used also for authentication and non-repudiation. WS-Security provides a SOAP binding for XML signature by defining how an XML signature can be placed in a SOAP message. XML Signature makes it possible to express a signature in a standardized XML format, and to sign only part of an XML document. It contains a KeyInfo element that can be used to reference the public key of the signer.

XML Encryption. *XML Encryption* (XML-ENC) [17] allows confidentiality to be satisfied on an end-to-end basis. Portions of an XML document can be selectively encrypted, and encrypted data can be expressed using XML. XML may also express an encrypted key, information about how an agreement was reached on the encrypted key, reference to the encrypted data, information about the data type of the encrypted document, and the encryption method used. XML-ENC uses the *KeyInfo* block from XML Signature.

XKMS. The *XML Key Management Specification v2.0* (XKMS 2.0) [18] is a W3C recommendation that specifies protocols for distributing and registering public keys, suitable for use in conjunction with XML-SIG and XML-ENC. XKMS comprises two parts, the *XML Key Information Service Specification* (X-KISS) and the *XML Key Registration Service Specification* (X-KRSS). X-KISS is a protocol that allows a client to delegate part or all of the tasks required to process XML Signature elements to an XKMS service. X-KRSS defines a protocol for registration and management of public key information.

SAML. The *Secure Assertion Markup Language* (SAML) [37] is an OASIS specification, later extended by the Liberty Alliance Project and the Internet2 Shibboleth group, concerned with access control for authenticated entities based on a set of policies. SAML allows trust assertions concerning authorization, authentication, and attributes of specific entities, to be specified using XML. An assertion is either a claim, a statement, or a declaration, and can be accepted as true to the extent that the certification authority that issued the claim can be trusted. Thus, authorities for attributes and authentication are crucial elements in the SAML model. SAML also defines a client/server protocol for exchanging XML messages. Typically, the underlying transport protocol is SOAP running over HTTP. SAML also enables portable trust by supporting authentication assertions between multiple administrative domains, a capability that is very important for Grid Services. Furthermore, it allows the mapping of access control elements between different systems. SAML has been proposed as a message format for expressing and requesting authorization assertions from an OGSA authorization service [11]. SAML 2.0, which became an OASIS standard in March 2005, added features to enable communication between SAML authorities, to enhance authentication methods, and to protect privacy.

XACML. The *Extensible Access Control Markup Language* (XACML) [38], developed at OASIS, is a language for expressing access control policies. XACML has the ability to express the complex policies that are not embedded into application code, and can also associate actions, called obligations, with access control decisions. Important for Grid services and context-based authorization is the ability endowed by XACML to base decisions on a resource's properties, or on environmental factors such as date, time, and location. It may also take into account properties such as role or group membership of the all the entities involved in a request, including intermediaries to the request. Of fundamental importance for Grids is the ability of XACML to operate in large-scale environments with multiple administrators that create policies. Specific features have been defined to enable XACML and SAML to work together. XACML 2.0 was approved as an OASIS Standard in February 2005.

WSS: SOAP Message Security. The *Web Services Security (WSS): SOAP Message Security v1.0* specification [21] defines the use of *security tokens* and *digital signatures* to protect and authenticate SOAP messages. Three main mechanisms are provided: message integrity, message confidentiality, and the ability to send security tokens as part of a message, which can be associated with message contents. Message integrity and confidentiality are provided by the encryption and digital signature of XML elements in the message. WSS 1.0 became an OASIS Standard in 2004, and WSS 1.1 in February 2006.

WS-Policy. The *WS-Policy* specification [29] defines a policy data model and an extensible grammar for expressing the capabilities, requirements and general characteristics of a Web Service. It is used to convey the conditions for an interaction between a Web service requestor and a Web service provider. WS-Policy defines fundamentals used for creating security policies, such as the type of security tokens a service will accept, supported algorithms for encryption and data signatures, and privacy attributes. WSDL bindings are typically used in order to attach policy information to a Web Service.

WS-SecurityPolicy. The *WS-SecurityPolicy* [30] defines a set of security policy assertions for use with the WS-Policy framework with respect to security features provided in WSS: SOAP Message Security, WS-Trust and WS-SecureConversation.

WS-Trust. *WS-Trust* [28] is thought to enable applications to construct trusted SOAP message exchanges. It defines extensions that build on WS-Security to broker trust relationships and to provide a framework for requesting and using security tokens, managing trusts, and establishing and assessing trust relationships,. The extensions provide also methods for issuing, renewing, and validating security tokens. Trust relationships can be direct or brokered. In the latter case a *trust proxy* is used to read the WS-Policy information and request security tokens from an issuer. WS-Security is able to transfer security tokens using XML Signature and XML Encryption for integrity and confidentiality. The trust model allows also delegation and impersonation.

WS-Privacy. *WS-Privacy* (unpublished) uses a combination of WS-Policy, WS-Security, and WS-Trust to communicate privacy policies. For privacy, incoming SOAP requests are required to contain claims that the sender conforms to desired privacy policies. These claims are encapsulated into verifiable security tokens with the help of the WS-Security specification. WS-Privacy defines also how to express privacy requirements in WS-Policy descriptions, and WS-Trust is used evaluate the privacy claims included in SOAP messages.

WS-Secure Conversation. *WS-SecureConversation* [31] defines extensions that build on WS-Security and WS-Trust to provide secure communication across messages. WS-SecureConversation is designed for the SOAP message layer, and has been described as "SSL at the SOAP level." Since there is no concept of a session for a group of SOAP messages, WS-SecureConversation allows a requestor and a Web Service to mutually authenticate using SOAP messages, and also to establish a mutually authenticated security context that uses session keys, derived keys, and per-message keys. Asymmetric encryption is used to negotiate a symmetric key which can be used for a series of SOAP messages, thus avoiding message-level authentication. WS-SecureConversation builds upon WS-Security and WS-Trust to securely exchange contexts in order to negotiate and issue keys.

WS-Federation. *WS-Federation* [32] acts at a layer above WS-Policy and WS-Trust, and explains how federated trust scenarios and relationships may be constructed and managed using WS-Security, WS-Policy, WS-Trust, and WS-SecureConversation. It describes how to manage and broker the trust

relationships in a heterogeneous federated environment, including support for federated identities and management of pseudonyms. WS-Policy and WS-Trust are used to determine which tokens are consumed, and how to apply for tokens from a security token issuance service.

WS-Authorization. *WS-Authorization* (unpublished) describes how access control policies for a Web Service may be specified and managed. The specification is extensible with respect to both authorization format and authorization language, and supports both ACL-based and RBAC-based authorization.

4 Grid Security Requirements, Challenges, and Use Patterns

The special security requirements of Grid applications derive mainly from the dynamic nature of Grid applications and the notion of virtual organization (VO), which requires the establishment of trust across organizational boundaries. In this kind of environment, security relationships can be dynamically established among hundreds of processes spanning several administrative domains, each one with its own security policies. Important requirements in this context are heterogeneity and site autonomy: a site must keep control over its resources and usage policies. As a result, the Grid security requirements are complex and pose significant new challenges. Existing intra-domain security solutions and infrastructures must be integrated into the overall security architecture and interoperate with inter-domain security solutions, since organizations are not as a rule prone to change their internal security requirements and policies in order to become a part of a wider organization. Complex patterns of trust between the different organizations within a VO must thus be established by entities that must be able to determine the identities and rights of other entities to ensure that only legitimate ones may access the required resources. Grids focus on the users and their needs, allowing them to take advantage of multiple distributed resources located in several administrative domains. *Authentication* and *authorization*, and in general *trust policies and management*, have thus been major challenges in Grid security. Moreover, two general nonfunctional requirements for Grids have deep implications for security: integration and interoperability. In Section 4.3 we present these requirements, but to make the presentation more concrete we show first some typical usage scenarios in

Grid computing highlighting the corresponding security concerns involved, and introduce the underlying assumptions about Grids as well as some terminology. Finally, Section 4.4 is dedicated to a presentation of the security requirements of the emerging Enterprise Grid Computing.

4.1 Underlying assumptions and terminology

We give here a short account of the terminology and underlying assumptions related to a Grid system, its participants, entities, components and specific policies [3].

The participants involved in a Grid computation include *subjects* or *users*, *user proxies* operating on behalf of the user, *resources*, and *resource proxies* which are agents or processes operating on behalf of resources. *Credentials* are piece of information about the identity of a user, such as passwords and certificates. Trust domains are administrative units with a local security policy, and consists of users and resources.

A Grid environment consists of a *Virtual Organization*, multiple trust domains with local security policies that cannot be overridden by the Grid security policy. Operations confined to a trust domain are subject solely to local policy, and can be implemented by a variety of mechanisms.

Subjects must have globally defined names besides local names, and there may exist partial mappings from global to local subject names. If there is a mapping in a trust domain for a determined global name that has been globally authenticated, then the subject is assumed to be locally authenticated also. The identity of the user needs to be passed transparently between sites during the execution of a job. This is the basis of *single sign-on*, which is made possible by the existence of a global identity. Access control decisions are always made locally on the basis of the local name.

Mutual authentication is required when a operation involve entities located at different trust domains. It is possible for a user to delegate a subset of his rights to a process to act on his or her behalf, thus enabling the execution of long-lived processes without user interaction, as well as the creation by a process of new processes. Moreover, processes running on behalf of a single subject may share the same set of credentials, thus enhancing the scalability of the security architecture by avoiding the need to issue a unique credential for each process.

4.2 Typical usage scenarios

A variety of scenarios are typical of Grid environments [5]. We briefly present some typical ones, together with some security issues each one bring forth.

A job execution request. A user submits a request to initiate a job, accompanied by a description of the job and the user's Grid credentials, either personal credentials or VO-issued credentials. The request is thereafter evaluated by different policy evaluation points (PEPs) against both local and VO policies. If the request is authorized it is mapped to a set of local credentials and enforced by local enforcement mechanisms. During job execution the user may make management requests to the job [12].

Resource allocation. Resource allocation can be initiated by a user proxy or a process. The first step is to identify the resource proxy. Mutual authentications is then executed, upon which a request, possibly signed, will be sent to the resource. The resource check the requester's credentials, and if authorized the resource is allocated and a process is created on that resource if needed. The request can fail either because of an allocation, authentication, or authorization failure. It is the responsibility of the resource to enforce local authorization policies.

A job execution on a specified Grid computer with local I/O. Here the user designates the execution host and submits a job, possibly together with the code, to a Grid gateway, i.e. a process that accepts remote resource requests. The job uses only remote computation cycles and possibly temporary file storage, input data is uploaded at job submission, the output is returned along the connection for job submission. The security requirements in this case include: (i) mutual authentication of user and Grid gateway on host; (ii) Grid gateway on host must map Grid ID to a local one; (iii) request must be submitted by the Grid gateway to the resource gateway in a manner that enables the job to run as the authorized local user. Authorization to use the resource is performed here by the Grid gateway.

A job execution on a specified Grid computer with non-local I/O. In this case the remote job must access non-local files, and therefore delegation in some form becomes thus necessary. Additional security requirements are as follows: (i) if file transfer must occur before execution, authorization must be given to transfer these file on behalf of the user, and delegation

becomes thus necessary; (ii) otherwise, credentials must be obtained upon startup to obtain the data; (iii) a Kerberos ticket from the user may be needed since the remote job writes output to a local file server of type AFS or DFS; (iv) if the output is in the form of files that must be copied back to the user's machine submitting the job, credentials to be authorized with the Grid gateway on the local machine are needed, as well as some form of delegation.

A job execution requiring a combination of resources from multiple sites. In this case, a user starts a coordinated job that needs to combine resources from multiple sites. Specific resources may be selected by a third party service such as a scheduler, eventually following some explicit QoS or other kinds of user requirements. Remote execution at multiple sites may thus be required, together with the corresponding data manipulation. Possible security requirements associated with this scenario might involve (i) authorization to execute the required jobs or access data in each of the target Grid machines according to the user's credentials, and, recursively, to access any resources that might be requested by any of the started processes; (ii) authentication, single or mutual, for any agents involved during job execution, starting with the user; (iii) mapping of Grid IDs to local IDs; (iv) possibly some kind of credential or privilege delegation, since the scheduler or any remote job might be required to act on the user's behalf.

A job execution requiring advanced scheduling. In some jobs, advance reservation of data storage, network bandwidth or compute cycles may be required. Possible security requirements associated with this scenario are: (i) delegation of a user's rights to a scheduler to make reservations; (ii) bandwidth reservations may require that a bandwidth broker knows at reservation time that the user's connection will come from an authorized site; (iii) the user should be able to authenticate itself as the entity that made the reservation; in the context of group membership and reservation made on behalf of a group, the user should be able to prove group membership; (iv) non-repudiation: the resource proxy should not be able to falsely deny granting of reservation.

Job control. A job might be disconnected by the user and reattached later, possibly from another location, or a user might want to monitor a job's progress or enter steering information. Another user or collaborator

may be allowed to monitor the job at some specific time. Possible security requirements associated with this scenario are: (i) access policy for a job may be required; (ii) authentication by the collaborator; (iii) auditing may be required since the Grid software must provide a means of identifying which Grid user started a local job.

Accessing Grid Information Services. Information Services are present in most Grid architectures for helping in the location of services and determining their status and availability. Typically, users will be able to read from the Directory Service, and entities such as processes will be able to enter information and set access policies for their information. Possible security requirements associated with this scenario are: (i) authentication between users and the Information Services; (ii) implementation of required access control policy by the Information Service; (iii) confidentiality or message integrity on the communication from the publisher to the Information Service; (iv) the Information Service must be trusted by the publisher.

Setting or querying security parameters. Entities in a Grid environment may want to have the capability to constrain the manner in which they interact with other. For instance, a user or resource provider may want to define message integrity and confidentiality parameters, stakeholders may want to set authorization policies or to revoke access, principals may want to specify trust Grid hosts, require confidentiality on stored data, etc. All these scenarios are complex and meeting the requirements is in general difficult. For further details consult [5].

Auditing use of Grid resources. A typical scenario of this kind is when a Grid administrator may want to check a list of past requests and allowed or denied accesses. This implies that (i) the resource gateway must keep an unforgeable log of all access including time of access and user identity; (ii) access to the log should be carefully restricted; (iii) a mechanism must exist to signal troublesome access requests. The usefulness of such a log file depends on how trusted a server is. Restricted access to the log may also be desirable. In this case there should be mechanisms to restrict access to the logs.

A typical service request scenario. In this scenario, drawn from [10] and illustrated in Figure 2, we show an example involving Grid services,

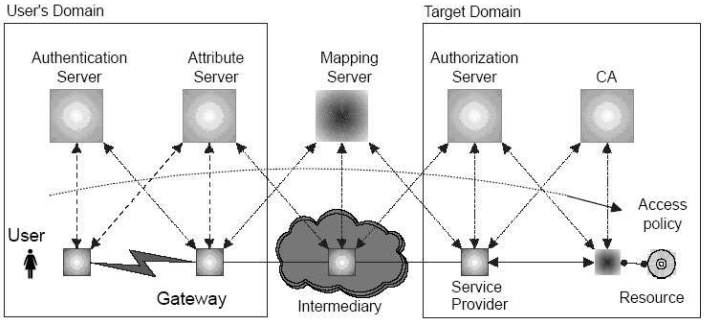

Figure 2: A service request scenario involving intermediaries.

explained in Section 5.

A user in his or her own domain wishes to invoke a Grid service in the target domain. The user first authenticates to an authentication server local to his or her domain, and obtain an identity credential. Thereafter the request is routed through a gateway, which may consult an attribute server to obtain the user's privilege attributes and rights. The assertions are then sent together with the service request. The request may be routed through an intermediary which is able to translate the assertions into a form that is understandable by the target domain and forwards the request according to a set of policies. Thereafter the target may receive the request and validate the certificate, and if successful, it can map the user's identity to a local one and make the appropriate authorization decisions using he locally defined policies.

The scenarios shown above illustrate many features associated with Grid environments [3]:

- user population, resources pool, and the group of processes running on different sites are potentially large and dynamic.
- processes may communicate by a variety of mechanisms such as unicast or multicast.
- different authentication and authorization mechanisms can be present in a single job computation, according to the local security policies of the sites involved.
- a user may be associated with different local name spaces or credentials.

- local authentication, authorization and access control may apply at different sites.

- individual users may be associated with different local name spaces, credentials and accounts at different sites.

4.3 Security Requirements

We present below the most common general security requirements and challenges associated with Grids.

Authentication. Authentication mechanisms and policies are supposed to constitute the basis on which local security policies can be integrated within a VO [3]. Because of its complexity and heterogeneity, in a Grid environment it is desirable to separate authentication from authorization. Difficult issues with respect to authentication in Grids are scalability, trust across different certification authorities, revocation, key management, and delegation. Since processes with delegated authority act on behalf of their owner, there is a question of authentication in delegation, which becomes even more complex when delegation is chained. Key management is also an issue for several reasons. Due to user mobility users may require a portable medium for media storage: Furthermore, users may have different credentials, for instance to cover different roles, which in practice means that numerous key pairs might become necessary.

Confidentiality. Both privacy and intellectual property concerns require confidentiality in the use of data. Encryption is one of mechanisms used to enforce confidentiality. The nature of Grids forces data to be stored in accessible online databases. Confidential code may be requested to execute on a remote host, and confidential data may need to be used at remote locations. Data may also need to be replicated at multiple sites, and thus should be stored in an encrypted form and remain consistent throughout. Furthermore, not only data but also users and resources may have privacy requirements, and users may be protected under privacy laws to which all components must adhere. Mechanisms for protection of confidentiality should also protect against the deducibility of data. Finally, laws regarding privacy rights and encryption vary among countries and must be taken into account when deploying Grid technologies across international borders.

Integrity. Many applications have strong code or data integrity concerns. The trust status of remote resources is important when data arises from remote processing as the accuracy of results can be trusted only to the extent that the remote host generating the data is trusted. Integrity is also an issue with regard to delegation, since the set of rights that have been delegated must not be modified maliciously.

Authorization and access control. Authorization is the process by which a subject is eventually allowed to access some resource. In Grids local access mechanisms should be applied whenever possible, and the owner of a resource should be able to enforce local user authorization. Users need also a consistent way to get authorization to access Grid resources across organizations. The first condition a user must meet in order to access the Grid is that he or she is a member of the VO, but eventual roles played by the user or other attributes may also be taken into consideration. Authorization by identity is very common, but in a Grid context resource owners may want to grant access based on e.g. roles, group membership, creditworthiness, static or dynamic and context-based attributes. Confirmation that a user has the VO membership and the required roles and attributes must be possible to obtain.

A resource provider in a Grid environment must have reached some form of agreement with the VO to allow the use of the resource. The VO may wish to specify a portion of the resource usage policies, to manage jobs running on VO resources, or to give some group of users the ability to manage those jobs. The authorization policy system must thus be able to combine policies from the resource owner and the VO, express policies about resource usage, manage VO-wide jobs and resource allocations, and dynamically enforce fine-grained policies about resource usage [12].

Revocation. Revocation is crucial for authentication in case of a compromised key, and for authorization when a VO is terminated or a user proves untrustworthy.

Distributed trust. Trust is a complex theoretical issue. A Grid must be constructed in a dynamic fashion from components whose trust status is hard to determine. For instance, a user that trusts R may not necessarily trust R to delegate the user's rights further. Determining trust relations between participant entities in the presence of delegation is important, and

delegation mechanisms must rely upon stringent trust requirements.

Freshness. Freshness is related to authentication and authorization and is important in many Grid applications. Validity of a user's proof of authentication and authorization is an issue when user rights are delegated and the duration of a job may span several weeks. Furthermore, some applications may want to state the number of times a given user may access a resource, a non-trivial problem when one user's rights are delegated to another user that may thereafter wish to access the resource.

Scalability. A Grid must be easy to extend and capable of progressive replacement. Fault recovery and dynamic optimization should be usually possible, and degradation should happen gracefully.

Trust. Trust refers to the assured reliance on someone or something. Since VOs can span multiple security domains, trust relationships between domains are of paramount importance. Sites in a Grid must be able to enter into trust relationships with Grid users and maybe other Grid sites as well. In a Grid environment trust is usually established through exchange of credentials, either on a session or a request basis. Due to the dynamic nature of Grid environments, trust can scarcely be established prior to session execution.

Single sign-on. A user should be able to authenticate only once, whereupon he may acquire, use and release resources without further authentication. This is required since a user may want to access a large number resources with different patterns of availability, access control policies, etc, that cannot be determined statically. Moreover, users may want to initiate computations running for long periods of time without needing to remain logged on all the time.

Delegation. Privilege delegation for operations executed by a proxy is a basic requirement for Grid environments, among other reasons in order to satisfy the single sign-on requirement. Delegation of user rights depends upon the security requirements of the application. Delegation is hard to achieve securely in practice, since enabling the delegation of a user's rights gives rise to many unresolved subtle issues and has a great impact on the overall security of a system [13].

Privacy. Privacy is the ability to keep information from being disclosed to determined actors. Privacy can be important in many Grid applications, for instance in medical and health Grids [14].

Non-repudiation. Non-repudiation refers to the inability to falsely deny the performance of some action. It is specially important in e-commerce involving money transactions. With the advent of Enterprise Grid this requirement becomes very important.

Credentials. A credential is a piece of information that may be used to prove the identity of a subject, e.g. a password or a private key. Interdomain access requires a uniform way of expressing the identities of users or resources, and must thus employ a standard for the encoding of credentials. Furthermore, user credentials must be protected.

Exportability. Code is required to be exportable and executable in multinational testbeds. As a result, bulk encryption cannot be required.

Secure group communication Authenticated communications for dynamic groups is required since the composition of a process group may change dynamically during execution.

Multiple implementations It should be possible to enforce security requirements with distinct security technologies and mechanisms.

Interoperability In the context of Grids, interoperability means that services within a single VO must be able to communicate across heterogeneous domains. Interoperability guarantees that services located in different administrative domains are able to interact at multiple levels. This gives rise to many serious security concerns related to authentication, privacy, authorization and policy enforcement. Services may be hosted in domains with different security mechanisms and policies, and interoperability between these services will depend on the trust models adopted.

With regard to policy management, security interoperability means that the security policies established by different parties in a VO can be made compatible, thus allowing the establishment of secure communications channels and security contexts following mutual authentication. This requires that users in different domains be able to identify each other. As result,

mechanisms for identity federation, mapping of identities, and credentials, must be made available, since global identities would be very impractical.

Interoperability with local security solutions Access to local resources is normally enforced by local security policies and mechanisms. Interoperability between sites and domains with differing local policies is necessary in a Grid environment. In order to accommodate interdomain access, one or several entities in a domain may act as agents of external entities for local resources.

Integration In order to allow the use of existing services and resources, integration requirements call for the establishment of an extensible architecture with standard interfaces. Security integration is facilitated by the use of existing security mechanisms. The latter is also in part a consequence of the requirement for site autonomy with regard to security policies, and also of the fact that no single security technology would be able to address the inherent complexity of Grid computing.

Uniform credentials and certification infrastructure A common way of expressing identity, e.g. by a standard such as X.509, is necessary for interdomain access.

4.4 Enterprise Grid Computing

Enterprise Grid Computing [34] is the use of Grid computing in the context of a business or enterprise. There are many requirements and challenges that are unique for Enterprise Grid Architectures, managed by a single enterprise or business. Resources consists basically of computing, network, storage, and service capabilities. Resources and services need not necessarily be owned by an organization, they may also be available through service providers or outsourcing firms. The boundaries of the Enterprise Grid are defined by its sphere of management responsibility and control. An Enterprise Grid may extend across several data centers, and no geographical limitations exist.

Based upon the assessment of threats and risks, many security requirements have been highlighted that are specific for Enterprise Grids, which we show below, together with more general requirements, following [34]. We follow he terminology used in this document, for details see Section 4.4. In

this model, a Grid consists of entities called components, and the Grid Management Entity (GME) is a logical entity that manages those components and their mutual relationships.

Confidentiality. Communication must be secure between Grid components for confidentiality, and the confidentiality of sensitive data must be preserved through the life cycle of Grid components.

Integrity. Grid components must be validated for security and integrity in accordance with the Grid security policy; integrity checks must be executed to guard against tampering the wire; images used to provision Grid components and settings during configuration processes, as well as information preserved from provisioning resources, must be validated for integrity.

Availability. Availability must be often enforced since it is obviously very important in many Enterprise Grids.

Identification. All components and user communities must be uniquely identifiable, and identities must be preserved.

Authentication. Communicating entities must be able to authenticate to each other; the GME must provide a functionality equivalent to an ordinary AAA (Authentication, Authorization, Auditing) server, including support for policy-based, extendible and strong authentication mechanisms, and for role-based resource access control.

Authorization. Grid components must be authorized to communicate with each other; authorization can tan be strict or loose depending on the nature of the organization.

Auditing. It must be possible to track and resolve the dynamic binding of Grid components; audit data must be meaningful also after reprovisioning or decommission of audited components.

Separation of Duties and Least Privilege. The standards of access control policy, separation of duties and least privilege, apply to Enterprise Grids.

Defense in Depth. Traditional defense in depth measures such as DMZs (demilitarized zones) should be preserved in Enterprise Grids; additional security measures can be taken by utilizing security measures to reinforce systemic security at every layer of the DAGs (directed-acyclic graphs) provided by the EGA reference model.

Secure failures. The GME and the Enterprise Grid as a whole must be designed to fail securely, i.e. Grid components must not be able to enter a vulnerable state.

Grid lifecycle security. A number of security requirements associated with the life cycle and reuse of Grid components are unique for Enterprise Grids; these include:

- *Secure packaging*: Grid components must be logically packaged for provisioning from resources. This allows components to be logically isolated from each other, packages to be easily modifiable, revised and managed for integrity. Packages should be also digitally signed or encrypted according to the security policy of the site.

- *Secure update of deployed components*: secure communication with components to query state, update and check pointing changes should be provided.

- *Secure archival*: it should be easy to extract needed imformation from a provisioned resource.

- *Secure reuse of Grid components.*

Interoperable security. Support for interoperable security across heterogeneous Grid components must de provided since a homogeneous environment cannot be assumed in Enterprise Grids.

Secure isolation. Since shareable pools of Grid components may be used, the same secure isolation requirements associated with physically or logically silo-ed environments apply for Enterprise Grids.

Trust relationships. Trust relationships in Enterprise Grids include relationships between users, administrators, applications, and services to the GME and Grid components; important questions here include how trust is established, maintained and terminated, and how trust violations are detected and addressed.

5 Grid Security Architectures and Models

In the early days of Grid computing, the definition of Grid was centered on computational aspects. A computational Grids was defined as "a hardware and software structure that provides dependable, consistent, pervasive, and inexpensive access to high-end computational capabilities" [4]. Several custom middleware solutions were created, but interoperability was hard to achieve. Later, the focus changed to coordinated resource sharing according to well-defined policies, easier integration, security, and QoS aspects. With the advent of Web services, we have seen in the last years a merging of Web and Grid services technologies. Today, the Open Grid Services Architecture (OGSA), announced in February 2002 by the Global Grid Forum, has become the standard architectural model for Grid systems. Section 5.1 is entirely dedicated to OGSA and OGSA security, and in Section 5.2 we give a brief presentation of the Enterprise Grid Alliance reference model security.

5.1 OGSA

The Open Grid Services Architecture (OGSA) is a service-oriented architecture (SOA) that represents an evolution towards a Grid system architecture based on Web services concepts and technologies, autonomic computing principles, and open standards for integration and interoperability.

Components in Web services are typically defined in terms of access methods, bindings of access methods to chosen communication mechanisms, and service discovery mechanisms. Some mechanisms are becoming de facto standards in Web services, such as the *Simple Object Access Protocol* (SOAP) [35], which uses XML technologies for messaging with HTTP as the underlying transport protocol, and the *Web Services Description Language* (WSDL) [36], in which signatures and bindings to protocols may be expressed in an XML document.

OGSA builds on concepts and technologies from both the Grid and Web services, and introduces the notion of *Grid service*, a potentially transient kind of Web service that conforms to a set of conventions for Grid interaction

expressed as WSDL interfaces, extensions and behaviors. OGSA also adds to Web services the important notion of a stateful service, as well as secure invocation methods and other capabilities. These additions to Web services are basically what defines a Grid service.

OGSA was created to meet the challenges related to the integration of services across VOs running on top of different native platforms [2]. In the context of Grid services, for instance, access control to resources amounts to controlling access to services through security protocols and policies. OGSA defines a set of core capabilities and behaviors addressing several aspects of Grid computing and the need for standardization. It specifies a set of characteristics describing how service requestors should interact with OGSA service providers. An important concept related to Grid services is the notion of *service virtualization*, which enables mapping of service semantics onto native platform facilities. OGSA also envisages mapping of security parameters between domains.

5.1.1 OGSA Security

Web Services standards did not meet all Grid security requirements from the beginning and were thus expanded with new service definitions. Grid requirements played a central role in the definition of WSDL 2.0 and in the review of *WS-Security*, a standard for creating secure message exchanges that provides mechanisms for authentication, confidentiality, encryption and message integrity. OGSA introduces new challenges for security.

Web Services security specifications include he Web Services Security Policy (*WS-Policy*) [30], XACML [38], SAML [37], WS-Security [26] for security token exchange, as well as the standards *WS-SecureConversation* [31] and *WS-Trust* [28] for authentication , establishment of security contexts, and trust relationships. However, OGSA introduces new challenges for security, and the specifications above have to be extended to address specific Grid security requirements.

The *OGSA security model* builds on Web Services security with specific extensions to cope with the challenges posed by virtual organizations. Security arises at various levels of the OGSA architecture. WS-security is used to allow service requests to provide suitable tokens, for purposes of e.g. authentication and authorization. For user authentication, delegation and single sign-on, the OGSA uses the Grid Security Infrastructure (GSI) [49] protocol. End-to-end message protection is also required by the OGSA architecture, and provided by mechanisms such as XML encryption and

digital signatures. Security components are also rendered as services, e.g. the OGSA authorization service which uses *WS-Agreement* (unpublished to date) along with SAML and XACML.

In the context of Grid services, some security challenges gain a new dimension:

- **Integration**: reuse of existing services and interface abstraction for extensibility.

- **Interoperability**: services located in different VOs and with different mechanisms and policies should be able to invoke each other

- **Trust relationship**: services should make access requirements available in order to enable access to them, and trust policies should be specified and enforced, e.g. through exchange of credentials. Moreover, heterogeneity calls for some form of federation among security mechanisms

Special security challenges related to trust relationships are associated with the notion of *transient services*, a class of Grid services that implements an interface that creates new Grid service instances [2] . Transient services are created by end users to perform some request-specific task which may involve execution of user code. Those challenges include:

- the requirement that it must be possible to control the *authorization status* under which transient services execute.

- *policy enforcement* by service providers even when users want to establish policies for the transient services they create.

- availability of the *assurance level* of a hosting environment for the benefit of the end user, including privacy, virus protection, firewall and VPN usage.

- *security policy composition* in the case that several policies are generated from different sources.

- *authority delegation* to enable transient services to perform actions on behalf of a user.

The OGSA *security model* stipulates that security mechanisms should be *pluggable* and *discoverable* by service requestors from a service description, enabling service providers to select their preferred mechanisms.

The Global Grid Forum's OGSA 1.0 [24] document targets security requirements including authentication and authorization, security infrastructures, perimeter security solutions, isolation, delegation, policy exchange, intrusion detection and protection, and secure logging. It also specifies security services associated with message integrity, confidentiality and privacy, auditing, intrusion prevention, and access control. We show these requirements below:

- **authentication:** plug points for multiple authentication mechanisms should be provided.
- **delegation:** support should be provided for enabling delegation of access rights from requestors to services, and for the specification of delegation policies.
- **single sign-on:** authentication to a VO should happen only once per session for the end user.
- **credential renewal:** the user should be notified whenever the expiration time of a credential is approaching.
- **authorization:** various access control models should be allowed, and access to OGSA services based on the authorization policies of each service should be possible, as well as the specification of invocation policies by requestors.
- **privacy:** both service requestors and providers should be able to specify and enforce privacy policies.
- **confidentiality:** confidentiality should be possible to maintain both in point-to-point transport and store-and-forward mechanisms.
- **message integrity:** unauthorized changes to a message should be detectable.
- **policy exchange:** service requestors and providers should be able to exchange policy information in order to establish a security context.

- **secure logging:** facilities for time-stamp and reliable logging are required, and are the basis for other important security requirements such as notarization, non-repudiation, and auditing.

- **assurance** means should be provided to qualify the security assurance level of a hosting environment, for instance with regard to virus protection or firewall usage.

- **manageability** security functionality, should be manageable, e.g. identity, policy or key management.

- **firewall traversal** mechanisms should be provided for cleanly traversing firewalls without compromising local control of firewall policy.

- **securing the OGSA infrastructure:** security of the components of the OGSA infrastructure must be provided.

5.1.2 OGSA Security Services

OGSA security services are intended to support the enforcement of security policies. The architecture is assumed to be implementation-agnostic, extensible, and easy to integrate with existing security services. OGSA components must enable systems to interoperate securely since services may traverse multiple domains. Also, due to heterogeneity of security infrastructures, required trust relationships are supposed to be established through some form of federation among the security mechanisms.

The model for security services in OGSA v1.0 [24] proposes a language to understand and describe *security policies*, which are defined as statements about *entities*, *interaction mechanisms* and *contexts*. The statements specify restrictions on associated *attribute values* and *properties*, and their *relationships*. Entities refer to *users*, *subjects* or *services*, and interact through mechanisms within a context. *Interaction mechanisms* refer to the different communication protocols, such as HTTP, SOAP or SSL/TSL. A *context* is related to interactions, and is a way of putting them in perspective, for instance by the establishment of a secure association. The *policy statements* are thus expressed in terms of entities, resources, and environment characteristics, and involve aspects such as authorization, authentication, trust, identity mapping, delegation, and assurance levels.

Security services are designed to support security policy enforcement, and are defined as "entities with interaction patterns that facilitate the ad-

ministration, expression, publishing, discovery, communication, verification, enforcement and reconciliation of the security policy." [24]

With regard to security, Grid applications differ from Web services by focusing on security services that enable cross-organizational interactions among entities. These entities have specific attributes and properties within a virtual community that differ from those in their home domain. Hence, the OGSA security services model has to support the concurrent enforcement of multiple policies that have to be evaluated each one within its own context.

Delegation of rights is needed in order to let services work on behalf of other entities. Since those services may become compromised, the delegated rights are limited to those rights that are truly needed by the service according to the *least-privilege delegation model*. This model requires the non-trivial calculation of the adequate number of rights required by the invoked service operations. The idea is to use the job directives expressed in a suitable language to specify the job requirements, which are matched against the capabilities of resources according to a language used to express resource capabilities. The latter should thus be able to match up with the language used to express job directives.

Security services should provide the required security functional capabilities. Figure 3, extracted from the OGSA 1.0 document [25], shows key relationships among service requestors, providers, and security services. It illustrates how different security services are invoked by the service requestor or the service provider. It can be seen that call-outs are made from within the stubs and thus are transparent to applications. Policy enforcement should be in part established in this way, thus keeping security-specific code at a minimum for application developers. The figure also shows that call-outs are made to different security service instances managed in different organizations, allowing compliance through configuration with the services and security policies of the requestor, the provider, and the VO. It can be seen also that the service requestor and the service provider are within the same VO but each is subject to their respective domain's policies. Requestor and provider are federated by the Bridge/Translation service that has credentials in both domains and may thus issue identity and capability assertions that can be validated in both domains. Outgoing arrows represent the interfaces to the security services from the requestor and provider, which must be specified in terms of OGSA interfaces.

Many of the following capabilities are considered in the OGSA 1.0 document:

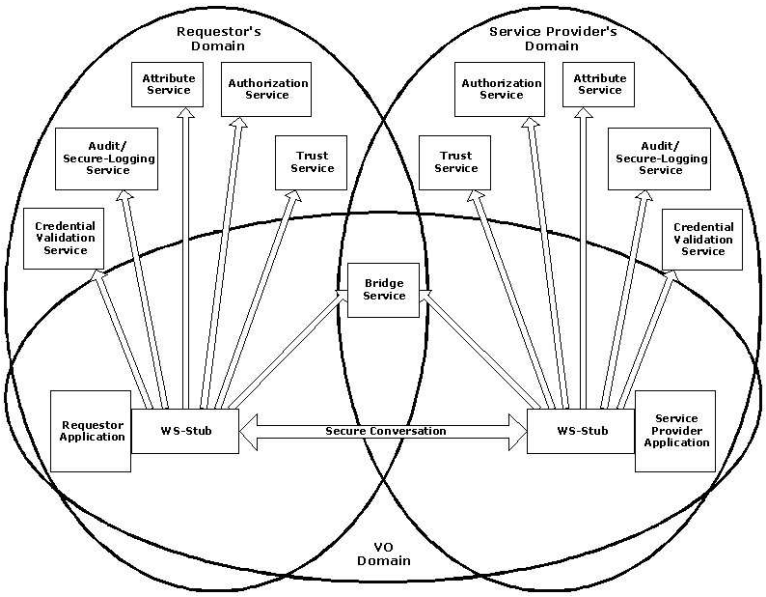

Figure 3: Security services in a virtual organization setting.

Authentication. This capability is part of the Credential Validation Service and Trust Service shown in Figure 3. Examples of authentication services are a combination of user-id and password or Kerberos authentication.

Identity mapping. This capability is provided by the Trust, Attribute and Bridge/Translation services. Identity mapping provides the possibility of associating identities existing in different identity domains.

Authorization. This service provide means to make policy-based access-control decisions. Resource access is typically authorized or denied according to the resource access policy and the requestor's credentials. It is expected that the hosting environment provides access control functions.

VO Policy. This service is concerned with the policy management. The policy service may be requested by services such as the authorization, audit, and identity mapping services.

Credential conversion. The capability of converting a credential from one type to another is provided by the Trust, Attribute and Bridge/Translation

Services. Credential conversion may enable the reconciliation of group membership, privileges, attributes and assertions associated with service requestors and providers, and facilitates also the interoperability of differing credential types. Credential conversion may require the service of identity mapping.

Audit and secure logging. The audit service is policy-driven and responsible for recording security-relevant events. This service is typically used by security administrators within a VO to check adherence to access-control and authentication policies. Auditing requires that events are logged in a secure fashion. Logging services and secure access to logs in a distributed setting is a complex problem since logs may reside in different administrative domains. Logs should be secured and tamper-proof, and capable of ensuring message integrity. Among the events that requires auditing are security events, e.g. an intrusion, which should be dealt with by the security services.

Profile. This service concerns the management of the preferences and personalized data of the service requestor that may not be directly consumed by the authorization service. This data may be used by applications that interface with a person.

Privacy. This service is concerned with the classification of personally identifiable information (PII) that may be stored by provider or requestors.

Figure 4, from [10], provides a view of the relationships between the components of the Grid security model as a layered stack of related services. The layering shows that application-specific components such as Secure Conversations depend on policies and rules for the components at the layer below, e.g. *Service/End-point Policy* or *Authorization Policy*. Further, the figure also shows that in order to apply and manage the policies and rules of a layer, e.g. the one in which the Authorization Policy resides, languages for *Policy Expression and Exchange* are required, as well as secure communication mechanisms through bindings to transport protocols or message security. Management components such as *Intrusion Detection* or *Policy Management* are shown in the left box in the picture.

Figure 5, extracted from [8], shows how the layering of existing security technologies and standards fit into the Grid security model. A determined

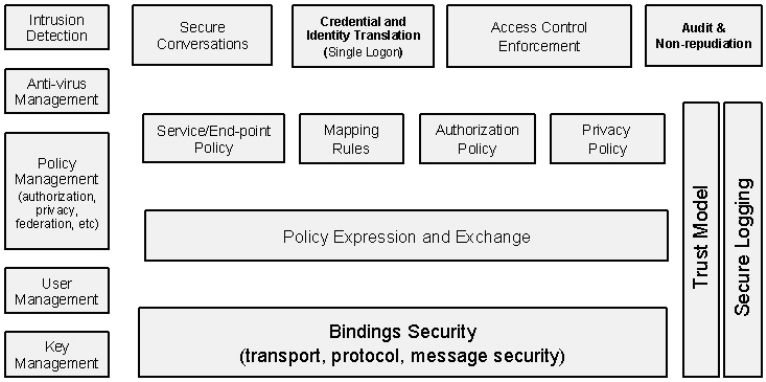

Figure 4: Components of Grid Security Model.

security function can be implemented at different levels, for instance at the network layer via IPSec or SSL/TLS, which provide only point-to-point security. SOAP and WS-Security, on the other hand, provide message level mechanisms at higher levels that can be used to achieve end-to-end security.

All OGSA security interfaces need to be standardized. Compliant implementations are supposed to be able to make use of existing services and policies through configuration, and to provide the associated and possibly alternative security services.

Invocations of OGSA services are usually subject to the enforcement of relevant security policies. OGSA security services may be closely connected to other services of higher level, and one security service may be a consumer of other OGSA services.

The Global Grid Forum produced a roadmap [7] leveraging existing and emerging Web Services security specifications and enumerating a set of proposed specifications to ensure interoperable implementations of the OGSA security architecture. The proposal builds on the framework described by the WS Security Architecture [26], which consists of layered modules including WS-Security, WS-Policy, WS-Federation, WS-SecureConversation, WS-Privacy, WS-Trust, and WS-Authorization. These modules are proposed to become building blocks for OGSA security. A set of profiles for WS security specifications has been proposed. It is recommended that WS security specifications are modified in Global Grid Forum specifications when they do no meet OGSA security requirements. The OGSA security specifications proposed include services such as naming, delegation, audit and secure logging, translation between security realms, and authorization, trust, privacy

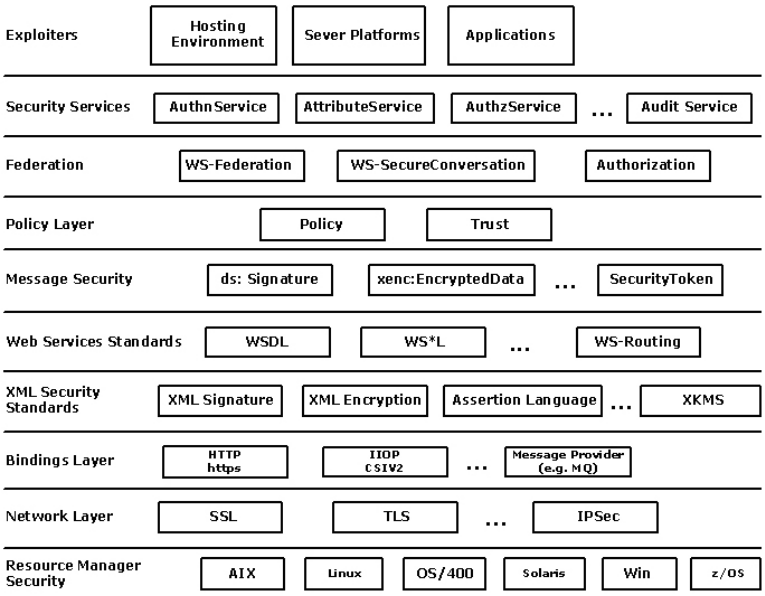

Figure 5: Security Specifications "Stack".

and VO policy management. Other proposed specifications concern support for multiple authentication mechanisms, authorization services plug-ability, security policy expression and exchange, interoperability through firewalls, and secure service operation.

5.2 Enterprise Grid Alliance Reference Model Security

The EGA reference model defines an *Enterprise Grid* as a collection of interconnected (networked) *Grid components* under the control of a *Grid management entity* [34]. A *Grid component* is defined as super class of object from which all of the components that are managed within an Enterprise Grid are descended or derived. Components include servers, network components, ERP services, online bookstores, etc. Grid components can as a rule be combined together into more sophisticated components.

Components have security properties and attributes, and may define specific dependencies that can be used to support enforcement of security policies and to ensure minimal exposure. The concept of Enterprise Grid wide dependencies and constraints supports the secure provision, configuration and enabling of entire services or business functions. To help minimize

risk, these attributes, dependencies and constraints should be enforced.

The Grid Management Entity (GME) is defined by the EGA reference model as the logical entity that manages the Grid components, the relationships among them, and their entire lifecycles. The GME should support the definition and enforcement of the security policy of the Enterprise Grid. The security functions that the GME should manage include: the user identities and administrative roles; authentication of identities; authorization of actions taken by principals; access restrictions to the Grid components; capture, storage, analysis and reporting of audit-related events; key management; enforcement of secure communications across the Grid and of secure isolation of shared Grid components and services; validation of individuals of groups with regard to their expected security states; and ensuring that local an remote management and troubleshooting operations are secured in accordance with the organization's security policy.

The EGA reference model defines three life cycle states of a Grid component: provision, ongoing management, and decommission/re-purposing.

Provisioning Provisioning involves adding, creating, configuring and starting a Grid component. The security attributes and properties associated with provisioning include questions such as the identity of the provisioner of the Grid component, the provisioning history, component verification and validation, satisfaction of required dependencies, and eventual constraints on the use of the component.

Ongoing management. Management of a Grid component involves any management related activities when the component is in an active state. The security issues related to ongoing management include questions such as who is authorized to create or modify components or administrative roles, the location for performing management functions, restrictions concerning the authentication of the administrator, management of administrative roles, management of Grid components and security attributes, distribution and updating of security policies, validation of security configuration, failure detection and repercussions of failures, detection of unauthorized changes, notification of security events, access control, and user authentication.

Decommissioning and Re-purposing. Decommissioning involves the retirement or re-purposing of a service or Grid component. Relevant security issues here include authorization to decommission/re-purpose a com-

ponent's security attributes, the history and other details of a resource's provisioning/decommission/re-purpose, and conditions under which a resource can be decommissioned/re-purposed.

References

[1] I. Foster, C. Kesselman, and S. Tuecke, The Anatomy of the Grid: Enabling Scalable Virtual Organizations. *International Journal of High Performance Computing Applications,* 15 (3). 200-222.2001.

[2] I. Foster and C. Kesselman and J. Nick and S. Tuecke. The Physiology of the Grid. *Global Grid Forum*, June 2002.

[3] Ian T. Foster and Carl Kesselman and Gene Tsudik and Steven Tuecke, A Security Architecture for Computational Grids, *ACM Conference on Computer and Communications Security*, 1998.

[4] I. Foster and C. Kesselman, The Grid: Blueprint for a New Computing Infrastructure. Morgan Kaufmann: San Francisco, CA, 1999.

[5] M. Humphrey and M. Thompson, Security implications of typical grid computing usage scenarios, HPDC 10, August 2001.

[6] M. Humphrey, M.R. Thompson, and K.R. Jackson, Security for Grids (August 14, 2005). *Lawrence Berkeley National Laboratory*, Paper LBNL-54853.

[7] F. Siebenlist, V. Welch, S. Tuecke, I. Foster, N. Nagaratnam, P. Janson, J. Dayke, and A. Nadalin, OGSA Security Roadmap, Global Grid Forum 5 Document, Open Grid Security Architecture Security Working Group, July 2003.

[8] F. Siebenlist, V. Welch, S. Tuecke, I. Foster, N. Nagaratnam, P. Janson, J. Dayke, and A. Nadalin, OGSA Security Roadmap, Global Grid Forum 5 Document, Open Grid Security Architecture Security Working Group, July 2003, page 6, Copyright ©Global Grid Forum (2003), all rights reserved.

[9] G. Della-Libera et. al., Security in a web services world: A proposed architecture and roadmap. White paper, IBM Corporation and Microsoft Corporation, April 2002.

[10] N. Nagaratnam, P. Janson, J. Dayka, A. Nadalin, F. Siebenlist, V. Welch, I. Foster, and S. Tuecke. The Security Architecture for Open Grid Services. Open Grid Services Security Architecture WG, Global Grid Forum, 2.9 (Draft Version 1), July 2002.

[11] V. Welch, F. Siebenlist, D. Chadwick, S. Meder, and L. Pearlman. Use of SAML for OGSA Authorization, Global Grid Forum, 2004.

[12] K. Keahey, V. Welch, S. Lang, B. Liu, and S. Meder, Fine-Grained Authorization for Job Execution in *The Grid: Design and Implementation. Concurrency and Computation: Practice and Experience,* April, 2004, 16(5): p. 477-488.

[13] P.J. Broadfoot and G. Lowe. Architectures for Secure Delegation Within Grids. Technical Report PRG-RR-03-19, Oxford University Computing Laboratory, September 2003.

[14] J. Herveg, F. Crazzolara, S. E. Middleton, D.J. Marvin, and Y. Poullet, GEMSS: Privacy and security for a Medical Grid. In *Proceedings of HealthGRID* 2004, Clermont-Ferrand, France.

[15] S. Parastatidis, J. Webber, P. Watson and T. Rischbeck, A Grid Application Framework based on Web Services Specifications and Practices, Document version: 1.0, 12 August 2003.

[16] D. Eastlake, J. Reagle, and D. Solo. XML-Signature Syntax and Processing. World Wide Web Consortium, Recommendation REC-xmldsig-core-20020212, February 2002.

[17] D. Eastlake, J. Reagle, and D. Solo, XML-Signature Syntax and Processing. World Wide Web Consortium, Recommendation REC-xmlenc-core-20021210, December 2002.

[18] P. M. Hallam-Baker, S, H. Mysore, XML Key Management Specification (XKMS) Version 2.0, World Wide Web Consortium, Recommendation REC-xkms2-20050628, June 2005.

[19] M. Atkinson, D. DeRoure, A. Dunlop, G. Fox, P. Henderson, T. Hey, N. Paton, S. Newhouse, S. Parastatidis, A. Trefethen, and P. Watson, Web Service Grids: An Evolutionary Approach, Report UKeS-2004-05, UK e-Science Technical Report Series, 2004.

[20] M. Baker, A. Apon, C. Ferner and J. Brown, Emerging Grid Standards *IEEE Computer*, Volume 38, Number 4 (April 2005), pages 43-50.

[21] Web Services Security: SOAP Message Security 1.0 (WS-Security 2004), OASIS Standard 200401, March 2004.

[22] J. Linn, Generic Security Service Application Program Interface, Version 2, RFC 2078.

[23] Open Grid Services Infrastructure (OGSI) Version 1.0, Global Grid Forum, 27 June 2003.

[24] The Open Grid Services Architecture, Version 1.0, Global Grid Forum, 29 January 2005.

[25] The Open Grid Services Architecture, Version 1.0, Global Grid Forum, 29 January 2005, p. 43. Copyright ©Global Grid Forum (2002-2005), all Rights Reserved.

[26] Web Services Security: SOAP Message Security 1.1, OASIS Standard Specification, 1 February 2006.

[27] www-128.ibm.com/developerworks/library/specification/ws-secmap.

[28] Web Services Trust Language (WS-TrusT), February 2005.

[29] Web Services Policy Framework (WS-Policy), March 2006 Version 1.2.

[30] Web Services Security Policy Language (WS-SecurityPolicy) Version 1.1., July 2005.

[31] Web Services Secure Conversation Language, February 2005.

[32] Web Services Federation Language (WSFederation), Version 1.0, July 8 2003.

[33] Basic Profile Version 1.0. Web Services Interoperability Organizarion, 16 April 2004.

[34] Enterprise Grid Security Requirements Version 1.0, Enterprise Grid Alliance Security Working Group, 8 July 2005.

[35] www.w3.org/TR/soap.

[36] www.w3.org/TR/wsdl.

[37] www.oasis-open.org/committees/security.

[38] www.oasis-open.org/committees/xacml.

[39] www.omii.ac.uk.

[40] www.ggf.org.

[41] www.oasis-open.org.

[42] www.w3.org.

[43] www.dtmf.org.

[44] www.internet2.edu.

[45] www.projectliberty.org.

[46] www.gridalliance.org.

[47] www.ws-i.org.

[48] www.globus.org/toolkit.

[49] www.globus.org/security/overview.html.

[50] www.globus.org/ogsa.

[51] www.globus.org/wsrf.