The Security Architecture for Open Grid Services

July 17, 2002, Version 1

Nataraj Nagaratnam[1], Philippe Janson[2], John Dayka[3], Anthony Nadalin[4]
Frank Siebenlist[5], Von Welch[6], Ian Foster,[5,6] Steve Tuecke[5]

[1] IBM Corporation, Research Triangle Park, NC 27703

[2] IBM Corporation, Zurich Research Lab, Switzerland

[3] IBM Corporation, Poughkeepsie, NY 12601

[4] IBM Corporation, Austin, TX 78759

[5]Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL 60439

[6]Department of Computer Science, University of Chicago, Chicago, IL 60637

Abstract

This document proposes a strategy for addressing security within the Open Grid Services Architecture (OGSA). It defines a comprehensive Grid security architecture that supports, integrates and unifies popular security models, mechanisms, protocols, platforms and technologies in a way that enables a variety of systems to interoperate securely. *This security architecture is intended to be consistent with the security model that is currently being defined for the Web services framework used to realize OGSA's service-oriented architecture. The document presents a security model, describes a set of security components that need to be realized in the OGSA security architecture, and presents a set of use patterns that show how these components can be used together in a secure Grid environment.*

Table of Contents

1 Introduction

Research and development efforts within the Grid community have produced protocols, services, and tools that address the challenges arising when we seek to build scalable *virtual organizations* (VOs). For the purpose of this paper, a virtual organization is defined as a set of individuals and/or institutions sharing resources and services under a set of rules and policies governing the extent and conditions for that sharing. As stated in [ANA]**,** "the sharing that Grid environments are concerned with is not primarily file exchange but rather direct access to computers, software, data, and other resources, as is required by a range of collaborative problem-solving and resource-brokering strategies emerging in industry, science, and engineering. This sharing is, necessarily, highly controlled, with resource providers and consumers defining clearly and carefully just what is shared, who is allowed to share, and the conditions under which sharing occurs."

What distinguishes a VO from a classical organization is that it may gather individuals and/or institutions that have agreed to share resources and otherwise collaborate on an ad-hoc, dynamic basis, while they continue to belong to different real organizations, each governed by their own set of internal rules and policies. This poses a challenge when combined with the fact that an individual or institution may be a member of several VOs simultaneously. From a security point of view, one is thus confronted with protection domains that may superpose, straddle, and intersect one another in many different ways. Within this context, we require interoperability among domains while maintaining a clear separation of the security policies and mechanisms deployed by both virtual and real organizations.

The technologies that have evolved from the Grid community include security solutions that support management of credentials and policies when computations span multiple institutions; resource management protocols and services that support secure remote access to computing and data resources and the co-allocation of multiple resources; information query protocols and services that provide configuration and status information about resources, organizations, and services; and data management services that locate and transport datasets between storage systems and applications. These core technologies are evolving to include a programming model as proposed by the Open Grid Services architecture (OGSA) [PSY], which describes a set of characteristics that an OGSA service provider must adhere to and how service requestors should interact with it. These technologies take into account the fact that Grid service application topologies include a broad combination of mobile devices, gateways, proxies, load balancers, demilitarized zones (DMZs), outsourced data centers, and globally distributed, dynamically configured systems. Many of these systems rely on the ability for message processing intermediaries to forward messages.

A fundamental construct underlying many of the required attributes of the Grid services architecture is that of *service virtualization*. It is virtualization of Grid services that underpins the ability to map common service semantic behavior seamlessly onto native platform facilities. Current OGSA design work focuses on the adaptation of the Web Services Description Language (WSDL) for this purpose [PSY], although other interface definition languages (IDLs) could also be used.

Controlling access to services through robust security protocols and security policy is paramount to controlling access to VO resources and assets. Thus, authentication mechanisms are required so that the identity of individuals and services can be established, and service providers must implement authorization mechanisms to enforce policy over how each service can be used. The requirement for composition complicates issues of policy enforcement, as one must be able to apply and enforce policy at all levels of composition and to translate policies between levels of composition. For example, when running a data mining query against a distributed collection of databases, we might need to enforce not only database-specific access control policies based on the identity of the requestor but also resource consumption policies associated with the VO.

To address these challenges, this paper proposes an evolutionary approach to creating secure, integrated and interoperable Grid services based on a set of security abstractions that unify formerly dissimilar technologies. The following sections discuss the security challenges encountered in Grid environments (Section 2), and translate those challenges into requirements (Section 3). The paper then presents an architecture for a Grid security model that addresses the identified security challenges and requirements.

2 Security Challenges in a Grid Environment

The security challenges faced in a Grid environment can be grouped into three categories: integration with existing systems and technologies, interoperability with different "hosting environments" (e.g., J2EE servers, .NET servers, Linux systems), and trust relationships among interacting hosting environments. Relationships among these three categories of challenges are depicted in Figure 1.

2.1 The Integration Challenge

For both technical and pragmatic reasons, it is unreasonable to expect that a single security technology can be defined that will both address all Grid security challenges and be adopted in every hosting environment. Existing security infrastructures cannot be replaced overnight. For example, each domain in a Grid environment is likely to have one or more registries in which user accounts are maintained (e.g., LDAP directories); such registries are unlikely to be shared with other organizations or domains. Similarly, authentication mechanisms deployed in an existing environment that is reputed secure and reliable will continue to be used. Each domain typically has its own authorization infrastructure that is deployed, managed and supported. It will not typically be acceptable to replace any of these technologies in favor of a single model or mechanism.

Thus, to be successful, a Grid security architecture needs to step up to the challenge of integrating with existing security architectures and models across platforms and hosting environments. This means that the architecture must be *implementation agnostic*, so that it can be instantiated in terms of any existing security mechanisms (e.g., Kerberos, PKI); *extensible*, so that it can incorporate new security services as they become available; and *integratable* with existing security services.

2.2 The Interoperability Challenge

Services that traverse multiple domains and hosting environments need to be able to interact with each other, thus introducing the need for interoperability at multiple levels:

- At the *protocol level*, we require mechanisms that allow domains to exchange messages. This can be achieved via SOAP/HTTP, for example.
- At the *policy level*, secure interoperability requires that each party be able to specify any policy it may wish in order to engage in a secure conversation—and that policies expressed by different parties can be made mutually comprehensible. Only then can the parties attempt to establish a secure communication channel and security context upon mutual authentication, trust relationship, and adherence to each other's policy.
- At the *identity level*, we require mechanisms for identifying a user from one domain in another domain. This requirement goes beyond the need to define trust relationships and achieve federation between security mechanisms (e.g., from Kerberos tickets to X.509 certificates). Irrespective of the authentication and authorization model, which can be group-based, role-based or other attribute-based, many models rely on the notion of an identity for reasons including authorization and accountability. It would be nice if a given identity could be (pre)defined across all participating domains, but that is not realistic in practice. For any cross-domain invocation to succeed in a secure environment, mapping of identities and credentials must be made possible. This can be enforced at either end of a session through proxy servers or through trusted intermediaries acting as trust proxies.

2.3 The Trust Relationship Challenge

Grid service requests can span multiple security domains. Trust relationships among these domains play an important role in the outcome of such end-to-end traversals. A service needs to make its access requirements available to interested entities, so that they can request secure access to it. Trust between end points can be *presumed*, based on topological assumptions (e.g., VPN), or *explicit*, specified as policies and enforced through exchange of some trust-forming credentials. In a Grid environment, presumed trust is rarely feasible due to the dynamic nature of VO relationships. Trust establishment may be a one-time activity per session or it may be evaluated dynamically on every request. The dynamic nature of the Grid in some cases can make it impossible to establish trust relationships among sites prior to application execution [COMP]. Given that the participating domains may have different security technologies in their infrastructure (e.g., Kerberos, PKI) it then becomes necessary to realize the required trust relationships through some form of federation among the security mechanisms.

The trust relationship problem is made more difficult in a Grid environment by the need to support the dynamic, user-controlled deployment and management of *transient services* [PSY]. End users create such transient services to perform request-specific tasks, which may involve the execution of user code. For example, in a distributed data mining scenario, transient services may be created at various locations both to extract information from remote databases and to synthesize

summary information. Challenges associated with user-created transient services include the following.

- *Identity and authorization*. It must be possible to control the authorization status (e.g., identity) under which transient services execute.
- *Policy enforcement*. Users may want to establish policies for services that they "own," to control, for example, who can access them and what actions they can perform. However, these policies must necessarily be bounded by policies enforced by the service provider that hosts the user service.
- *Assurance level discovery*. A user may want to take into account the assurance level of a hosting environment when deciding where to deploy services. Thus, this information must be discoverable. Issues of concern may include virus protection, firewall usage for Internet access, and internal VPN usage. One approach to providing this information is to use an accreditation mechanism in which a third-party accreditation agency attests to the level of security provided [NEUMAN].
- *Policy composition*. Security policy on instantiated services can be generated dynamically from multiple sources: not just the resource owners, but from the entity whose request created the service and the VO in which the entity's membership entitles them to do so.
- *Delegation*. Transient services may need to be able to perform actions on a user's behalf without their direct intervention. For example, a computational job running overnight might need to access data stored in a different resource. Since there may be no direct trust relationship between the VO in which the service is running and the VO in which it wishes to make a request, the service needs to be able to delegate authority to act on the user's behalf. A number of secondary issues flow from this requirement. For example: how can a user minimize the credentials they delegate to a transient service to reduce their exposure? And what happens if the credentials delegated to the service expire before it has completed it's task?

Controlled access to VO resources and services is clearly a critical aspect of a secure Grid environment.

Given the dynamic nature of Grids and the scale of the environment, serious challenges exist and need to be addressed in the area of security exposure detection, analysis, and recovery.

In summary, security challenges in a Grid environment can be addressed by categorizing the solution areas:

(a) integration solutions where existing services needs to be used, and interfaces should be abstracted to provide an extensible architecture;

(b) interoperability solutions so that services hosted in different virtual organizations that have different security mechanisms and policies will be able to invoke each other; and

(c) solutions to define, manage and enforce trust policies within a dynamic Grid environment.

A solution within a given category will often depend on a solution in another category. The dependency between these three categories is illustrated in Figure 1. For example, any solution for federating credentials to achieve interoperability will be dependent on the trust models defined within the participating domains and the level of integration of the services within a domain. Defining a trust model is the basis for interoperability but trust model is independent of interoperability characteristics.

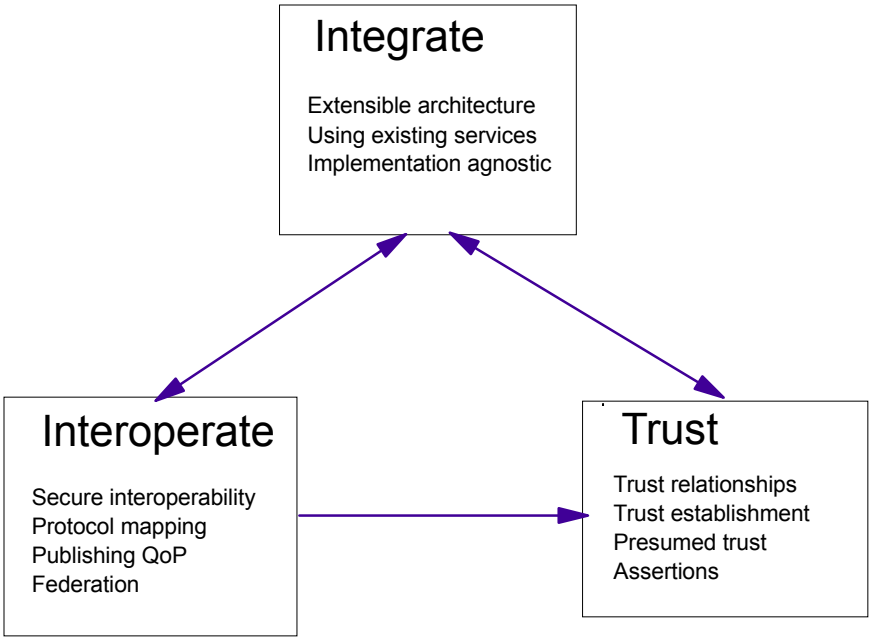

Figure 1: Categories of security challenges in a Grid environment

Similarly level of integration implies a level of trust as well has a bearing on interoperability.

In a Grid environment, where identities are organized in VOs that transcend normal organizational boundaries, security threats are not easily divided by such boundaries. Identities may act as members of the same VO at one moment and as members of different VOs the next, depending on the tasks they perform at a given time. Thus, while the security threats to OGSA fall into the usual categories (snooping, man-in-the-middle, intrusion, denial of service, theft of service, viruses and Trojan horses, etc.) the malicious entity could be anyone. An additional risk is introduced, when multiple VOs share a virtualized resource (such as a server or storage system) where each of participating VOs may not trust each other and therefore, may not be able to validate the usage and integrity of the shared resource. Security solutions that focus on establishing a perimeter to protect a trusted "inside" from an untrusted "outside" (e.g., firewalls, VPNs) are of only limited utility in a Grid environment.

The size of some Grid environments introduces the need to deal with large-scale distributed systems. The number, size, and scalability of security components such as user registries, policy repositories, and authorization servers pose new challenges. This is especially true in the area of inter-domain operations where the number of domains explodes. Many cross-domain functions that may be statically pre-defined in

other environments will require dynamic configuration and processing in a Grid environment.

3 Grid Security Requirements

We now proceed to translate the preceding general discussion of the Grid security problem into specific Grid security requirements.

Recall that the goal and purpose of Grid technologies is to support the sharing and coordinated use of diverse resources in dynamic, distributed VOs: in other words, to enable the creation, from distributed components, of virtual computing systems that are sufficiently integrated to deliver desired qualities of service. Security is one of the characteristics of an OGSA-compliant component. The basic requirements of an OGSA security model are that security mechanisms be *pluggable* and *discoverable* by a service requestor from a service description. This functionality then allows a service provider to choose from multiple distributed security architectures supported by multiple different vendors and to plug its preferred one(s) into the infrastructure supporting its Grid services.

OGSA security must be seamless from edge of network to application and data servers, and allow the federation of security mechanisms not only at intermediaries, but also on the platforms that host the services being accessed.

The basic OGSA security model must address the following security disciplines:

- *Authentication*. Provide plug points for multiple authentication mechanisms and the means for conveying the specific mechanism used in any given authentication operation. The authentication mechanism may be a custom authentication mechanism or an industry-standard technology. The authentication plug point must be agnostic to any specific authentication technology.
- *Delegation*. Provide facilities to allow for delegation of access rights from requestors to services, as well as to allow for delegation policies to be specified. When dealing with delegation of authority from an entity to another, care should be taken so that the authority transferred through delegation is scoped only to the task(s) intended to be performed and within a limited lifetime to minimize the misuse of delegated authority.
- *Single Logon*. Relieve an entity having successfully completed the act of authentication once from the need to participate in re-authentications upon subsequent accesses to OGSA-managed resources for some reasonable period of time. This must take into account that a request may span security domains and hence should factor in federation between authentication domains and mapping of identities. This requirement is important from two perspectives:

    a) It places a secondary requirement on an OGSA-compliant implementation to be able to delegate an entity's rights, subject to policy (e.g., lifespan of credentials, restrictions placed by the entity)

    b) If the credential material is delegated to intermediaries, it may be augmented to indicate the identity of the intermediaries, subject to policy.

- *Credential Lifespan and Renewal*. In many scenarios, a job initiated by a user may take longer than the life span of the user's initially delegated credential. In those cases, the user needs the ability to be notified prior to expiration of the credentials, or the ability to refresh those credentials such that the job can be completed.
- *Authorization*. Allow for controlling access to OGSA services based on authorization policies (i.e., who can access a service, under what conditions) attached to each service. Also allow for service requestors to specify invocation policies (i.e. who does the client trust to provide the requested service). Authorization should accommodate various access control models and implementation.
- *Privacy*. Allow both a service requester and a service provider to define and enforce privacy policies, for instance taking into account things like personally identifiable information (PII), purpose of invocation, etc. (Privacy policies may be treated as an aspect of authorization policy addressing privacy semantics such as information usage rather than plain information access.)
- *Confidentiality*. Protect the confidentiality of the underlying communication (transport) mechanism, and the confidentiality of the messages or documents that flow over the transport mechanism in a OGSA compliant infrastructure. The confidentiality requirement includes point–to–point transport as well as store-and-forward mechanisms.
- *Message integrity*. Ensure that unauthorized changes made to messages or documents may be detected by the recipient. The use of message or document level integrity checking is determined by policy, which is tied to the offered quality of the service (QoS).
- *Policy exchange.* Allow service requestors and providers to exchange dynamically security (among other) policy information to establish a negotiated security context between them. Such policy information can contain authentication requirements, supported functionality, constraints, privacy rules etc.
- *Secure logging*. Provide all services, including security services themselves, with facilities for time-stamping and securely logging any kind of operational information or event in the course of time - securely meaning here reliably and accurately, i.e. so that such collection is neither interruptible nor alterable by adverse agents. Secure logging is the foundation for addressing requirements for notarization, non-repudiation, and auditing.
- *Assurance*. Provide means to qualify the security assurance level that can be expected of a hosting environment. This can be used to express the protection characteristics of the environment such as virus protection, firewall usage for Internet access, internal VPN usage, etc. Such information can be taken into account when making a decision about which environment to deploy a service in.
- *Manageability*. Explicitly recognize the need for manageability of security functionality within the OGSA security model. For example, identity management, policy management, key management, and so forth. The need for security management also includes higher-level requirements such as anti-virus protection, intrusion detection and protection, which are requirements in their own rights but are typically provided as part of security management.

- *Firewall traversal*. A major barrier to dynamic, cross-domain Grid computing today is the existence of firewalls. As noted above, firewalls provide limited value within a dynamic Grid environment. However, it is also the case that firewalls are unlikely to disappear anytime soon. Thus, the OGSA security model must take them into account and provide mechanisms for cleanly traversing them—without compromising local control of firewall policy.
- *Securing the OGSA infrastructure*. The core Grid service specification (OGSI) presumes a set of basic infrastructure services, such as handleMap, registry, and factory services. The OGSA security model must address the security of these components. In addition, securing lower level components (e.g., DNSSEC) that OGSI relies on would enhance the security of the OGSI environment

As Grid computing continues to evolve to support e-business applications in commercial settings, the requirements and functions discussed in this roadmap will form the foundation for standards-based interoperability not only between real organizations within a VO (intra VO) but also across organizations belonging in different VOs (inter VO). On this foundation applications and infrastructure can be built to establish trust relationships that are required for commercial distributed computing, enterprise application integration and business-to-business (B2B) partner collaboration over the Internet.

4 Grid Security Model Principles

From a security point of view, the virtualization of a service definition encompasses the security requirements for accessing that service. The need arises in the virtualization of security semantics to use standardized ways of segmenting security components (e.g., authentication, access control, etc.) and to provide standardized ways of enabling the federation of multiple security mechanisms. The benefits of having a loosely-coupled, language-neutral, platform-independent way of linking and securing applications within organizations, across enterprises, and across the Internet is fundamental to the problem set addressed by the OGSA architecture. Therefore, abstracting security components as a single security model enables organizations to use their existing investments in security technologies while communicating with organizations using different technologies.

As evident from the Grid security requirements, securing Grid services is a fundamental requirement behind the security model proposed here. While providing the required security infrastructure, the environment may use the security functions and components, which may be exposed as Grid services. Therefore, the principles underlying the Grid security model can be categorized as:

a) A security model to secure Grid services in general and

b) Security services built to provide the necessary functionality.

The following subsections discuss these two categories.

4.1 Secure Invocation of Grid Services

The Grid security architecture must ensure that OGSA services when invoked by a service requestor adhere to policy constraints as levied by the hosting environment.

Such policy may include a specific type of credential, integrity and confidentiality requirements and so forth for successful invocation of the service. This architecture must also enable service requestors to dynamically select services which meet policy constraints levied by the service requestor, as a service requestor may select a service provider which best meets the requestor's policies.

A Grid service must be able to define or publish the Quality of Protection (QoP) it requires and the security attributes of the service Aspects of the QoP include security bindings supported by the service, the type of credential expected from the service requestor, integrity and confidentiality requirements, etc. The security attributes of the service can include information such as service identity. This enables service requestors to discover a service based on the requestor's security characteristics. Additionally, service requestors will be able to evaluate their invocation policies based on the security attributes of the service. Note that that there may be policy restrictions on the visibility of the service's security attributes.

From the service provider's point of view requests to invoke Grid services by service requestors are subject to policy checks defined by the service's access policies. For example, some policies may require that the service provider will only allow the invocation of a service after the service requestor has authenticated itself first, and provides an appropriate credential when invoking the service.

These requirements highlight the need for establishing standard mechanisms for conveying and enforcing the quality of protection, security attributes and access policies associated with services and requesters.

4.2 Grid Security Services

The suite of security services and the primitives, which are required as building blocks, provide a rich set of services to application logic hosted in an OGSA environment. This does not imply that application components need to be aware of security semantics per se. Rather, an OGSA implementation, which uses a hosting environment, may govern via authorization policy whether a given service can be instantiated. Depending on the application or hosting environment, a generic set of security primitives provide a robust foundation for applications and so forth.

As described in 4.1, secure invocation of Grid services brings out the need for a security model that reflects the security components that need to be identified and defined based on the Grid security requirements. Section 5 describes a security model based on those requirements and identifies a set of security components that need to be defined and formalized as specifications. As described in Section 4.2, some of the security components can be realized as Grid security services and these are discussed in Section 7.

5 Grid Security Model

Industry efforts have rallied around Web services (WS) as an emerging architecture which has the ability to deliver integrated, interoperable solutions. Ensuring the

integrity, confidentiality and security of Web services through the application of a comprehensive security model is critical, both for organizations and their customers – which is the fundamental starting point for constructing virtual organizations. The secure interoperability between virtual organizations demands interoperable solutions using heterogeneous systems. For instance, the secure messaging model proposed by the Web Services Security roadmap [WSR] document supports both public key infrastructure (PKI) and Kerberos mechanisms as particular embodiments of a more-general facility and can be extended to support additional security mechanisms.

The security of a Grid environment must take into account the security of various aspects involved in a Grid service invocation. This is depicted in the Figure 2.

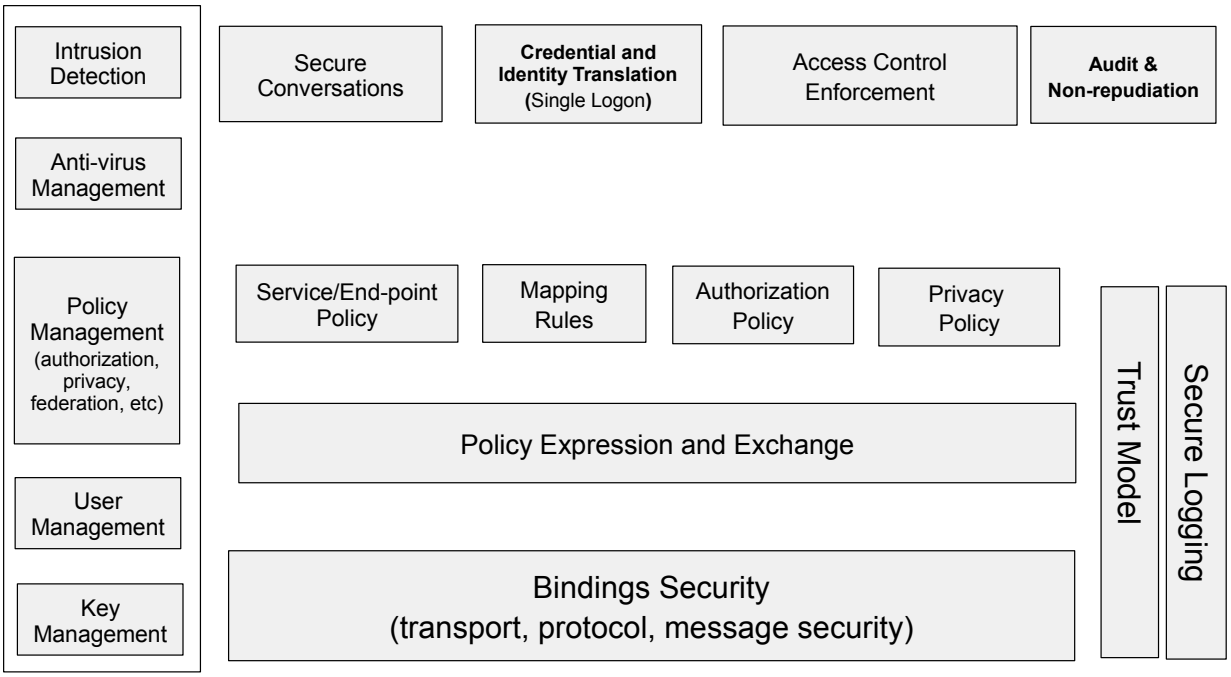

Figure 2: Components of Grid Security Model

A web service can be accessed over a variety of protocols and message formats it supports, as defined by its bindings [GRIDSPEC]. Given that bindings deal with protocol and message formats, they should provide support for quality of service, including such security functions as confidentiality, integrity, and authentication.

Each participating end point can express the policy it wishes to see applied when engaging in a secure conversation with another end point. Policies can specify supported authentication mechanisms, required integrity and confidentiality, trust policies, privacy policies, and other security constraints. Given the dynamic nature of Grid service invocations, end points will often discover the policies of a target service and establish trust relationships with it dynamically.

Once a service requestor and a service provider have determined each other's policies, they can establish a secure channel over which subsequent operations can be invoked. Such a channel should enforce various qualities of service including

identification, confidentiality, and integrity. The security model must provide a mechanism by which authentication credentials from the service requestor's domain can be translated into the service provider's domain and vice versa. This translation is required in order for both ends to evaluate their mutual access policies based on the established credentials and the quality of the established channel.

5.1 Binding Security

The set of bindings to be considered includes SOAP (SOAP/HTTP, SOAP over a message queue or SOAP over any other protocol) and IIOP bindings. The security of a binding is based on the security characteristics of the associated protocol and message format. If new protocols or message formats are introduced, care should be taken to address security requirements in those bindings so that, at a minimum, suitable authentication, integrity, and confidentiality can be achieved.

HTTP is an important protocol to consider because of its transparency to firewalls and wide adoption. In the case of bindings over HTTP, requests can be sent over SSL (i.e., "https") and thus SSL can provide authentication, integrity and confidentiality. However SSL ensures these qualities of service only among participating SSL connection end points. If a request needs to traverse multiple intermediaries (firewalls, proxies, etc), then end-to-end security needs to be enforced at a layer above the SSL protocol.

In the case of SOAP messages, security information can be carried in the SOAP message itself in the form of security tokens defined in the WS-Security specification [WSR]. SOAP messages can also be integrity and confidentiality protected using XML Digital Signature and XML Encryption support respectively. Signature and encryption bindings defined in WS-Security can be used for this purpose.

Web services can be accessed over IIOP when the service implementation is based on CORBA [CORBA]. In the case of IIOP, the security of the message exchange can be achieved by using the Common Secure Interoperability specification, version 2 (CSIv2)[CSI]. This specification is also adopted in J2EE [J2EE].

In addition to, or in lieu of, binding-level security requirements, network security solutions (e.g., firewalls, IPSec, VPN, DNSSEC, etc.) remain useful components for securing a Grid environment. Firewalls can continue to enforce boundary access rules between domains and other network level security solutions can continue to be deployed in intra-domain environments. Grid services deployment can take the topology into consideration when defining security policies. At the same time, deployment assumptions may be surfaced as policies attached to firewalls and network architecture.

The Grid security model must be able to leverage security capabilities of any of these underlying protocols or message formats. For example, in the case of SOAP over HTTP requests, one can use WS-Security for end-to-end security functionality, HTTPs for point-to-point security, and SSL, TLS or IPSec for other purposes. Security requirements for a given Web service access will be specified and honored based on the set of policies associated with the participating end points. For example, a policy associated with a Web service can specify that it expects SOAP messages to be

signed and encrypted. Thus, service requestors accessing that service would be required to use WS-Security to secure their SOAP requests.

Addressing the security of the service bindings will address the requirements related to integrity and confidentiality of messages, achieving delegation facilities, and facilitating firewall traversal.

5.2 Policy Expression and Exchange

Web Services have certain requirements that must be met in order to interact with them. For example, a service may support specific message encoding formats or may require specific security credentials to perform a specific action. A hosting environment has access to policies associated with a hosted web service so that it can enforce the invocation requirements when the service is accessed. It is important for service requestors to know about the policies associated with a target service. Once the service requestor knows the requirements and supported capabilities of a target service, it can evaluate the capabilities and mechanisms that the service provider supports. At the end of the evaluation, both the service requestor and the service provider together select the optimal set of bindings to converse with one another. Note that the ability to acquire this knowledge is a privilege given by the hosting environment's policy.

In a dynamic environment like the Grid, it is important for service requestors to discover these policies dynamically and make decisions at runtime. Such policies can be associated with the service definition (e.g., WSDL), service data (i.e. part of Grid service specification), or exchanged between service requestor and service provider (e.g., service provider can return a fault that contains information about the policy, or through some negotiation). It should be noted that discovering and reacting to policies can be part of the bindings themselves. For example, in the case of IIOP bindings, service requirements and capabilities are defined as part of the service reference (IOR) as a security tagged component [CSI].

In addition to service provider policies that need to be exposed to a service requester (or similarly service requestor policies to the service provider), there may be other policies that a service requestor or a service provider's environment needs to know but not necessarily expose in order to ensure a secure environment. For example, a service provider may have a set of authorization policies that indicate authorized requestors and this policy need not be (most likely will not be) exposed to service requestors. Similarly, service requestors may have policies specifying the identity of service provider's hosting environments it may trust.

Based on the web services roadmap document [WSR], WS-Policy will describe how both service providers and service requestors can specify their requirements and capabilities. WS-Policy will be fully extensible and will not place limits on the types of requirements and capabilities that may be described; however, the specification will likely identify several basic service attributes including privacy attributes, encoding formats, security token requirements, and supported algorithms. Grid service policies will also be specified and defined based on WS-Policy. In the case of Grid services, these policies can be exchanged in a variety of ways including but not limited to,

SOAP messages, service data (part of Grid service), part of bindings (e.g., CORBA security tagged component) or by using a policy discovery service.

Policy expression and exchange facilities will address the Grid security requirements to exchange policy between participating end points, securing the OGSI infrastructure and play a critical part to achieve secure association between the end points.

The bindings and exchange layers discussed so far allow service requestor and service provider to discover one another's policy. The next layer of the model deals with the nature and enforcement of these policies: secure association between service end points, mapping of identities and translation of credentials across domain boundaries between them, authorization policies and privacy policies, which together form the basis for enforcing control of access to protected services. These are reviewed in the following sections.

5.3 Secure Association

A service requester and a service provider are likely to exchange more messages and submit requests subsequent to an initial request. In order for messages to be securely exchanged, policy may require service requester and service provider to authenticate each other. In that case, a mechanism is required so that they can perform authentication and establish a security context. This security context can be used to protect exchange of subsequent messages. As an added benefit, using the established security context will improve the performance of secure message exchanges. The period of time over which a context is reused is considered a session or association between the interacting end points. Security context establishment and maintenance should be based on a web service context (to be) defined within web or Grid service specifications.

The notion of a context is tightly coupled with the bindings. Many existing protocols (e.g. IPSEC, SSL, IIOP) and mechanisms (e.g. Kerberos) already support secure association contexts. For example, in the case of IIOP, context establishment is based on the CSIv2 specification. In the case of SOAP, the context can be carried and secured as part of the SOAP messages. WS-SecureConversation will describe how a Web service can authenticate service requestor messages, how service requestors can authenticate service providers, and how to establish mutually authenticated security contexts. WS-SecureConversation will be designed to operate at the SOAP message layer so that the messages may traverse a variety of transports and intermediaries. Therefore, in the case of SOAP bindings, the Grid security model should adopt WS-SecureConversation to establish security contexts and exchange message securely. Alternatively, depending on the constraints of a VO other technologies (e.g., SASL, BEEP, etc) may be used. Therefore, the mechanism used to establish security contexts between end points will be based on the bindings used as well as the policy associated with the end points.

Facilitating secure association is required to establish the identity of a requestor to the service provider (and vice versa) so that the service provider (and service requestor) can satisfy the requirements to authenticate the identity on the other end

and then enforce authorization and privacy policies based on the established identity. The identities of the requestor and service provider are required for auditing purposes, so that audit logs will contain information about accessing identity.

5.4 Identity and Credential Mapping/Translation

A Grid environment consists of multiple trust (VOs) and security domains. Operations between entities in different domains will typically require mutual authentication. However the assumption that all domains may share a global user registry is unrealistic. Hence when operations between entities cross real domain as well as virtual organization boundaries, the identity of service requestors and providers, as well as their respective credentials as expressed in their home domain may not be syntactically or even semantically meaningful in their communication partner's domain. Enabling interoperation will thus require "federating" the involved domains and their respective security mechanisms, for example a Kerberos and a PKI domain.

This federation will typically be accomplished through mapping or translation of identities and/or credentials is required through proxies, gateways or trusted intermediaries. The mapping/translation components at this layer are responsible for implementing these functions as directed by corresponding policies. The definition of these policies is the subject of suitable management functions and trust models to be discussed later. The resulting federation framework forms the basis for addressing the requirements for single authentication and delegation.

WS-Federation will define how to construct federated trust scenarios using the WS-Security, WS-Policy, WS-Trust, and WS-SecureConversation specifications [WSR]. The Grid security model should perform federation based on the WS-Federation specification.

5.5 Authorization Enforcement

Policies required in the Grid security model also include authorization policies. Authorization is a key part of a security model and requires special mention. Each domain will typically have its own authorization service to make its own access decisions. In an Internet environment, authorization is typically associated with a service provider such that it controls access to a resource based on the identity of the service requestor. Clients, or service requestors, typically trust the server, or service provider. In case they do not, service provider authentication through SSL is one mechanism to establish service requestor trust in the service provider. In a Grid environment, or even a B2B environment, more stringent rules apply from the service requestor's side. Service requestors evaluate their relationship with the service provider's environment prior to deciding whether to trust the service provider to handle the request.

The implementation of the authorization engine in each domain may also follow different models (e.g., role based authorization, rule based authorization, capabilities, access control lists, etc). WS-Authorization will describe how access policies for a Web service are specified and managed. In particular it will describe how claims may be specified within security tokens and how these claims will be

interpreted at the end-points [WSR]. The Grid authorization model should build on top of WS-Authorization. It should take into account that every domain is likely to have its own authorization model, authorization authority and management facilities. Defining an authorization model will address the requirement provide a secure Grid environment by controlling access to Grid services.

Grid computations may grow and shrink dynamically, acquiring resources when required to solve a problem and releasing them when they are no longer needed [ANA]. Each time a computation obtains a resource, it does so on behalf of a particular service requestor and based on a set of privileges associated with the requestor. Identity based authorization is typical in most resource managers. It is necessary that any identity asserted by an end client (a service requestor) be recognizable and valid in service provider's domain, facilitated by the identity and credential mapping functions. This is independent of whether the domain can associate the asserted identity with a real end user. There are circumstances where a user may want to remain anonymous, or use a different (possibly shared) identity. As long as an asserted identity can be associated with a set of privilege attributes or rights that can be evaluated and used to make access decisions, it does not matter if the identity is mapped to a real end user. Though a real user identity may not be required to perform authorization, it may be required to map the asserted identity to an end user for non-repudiation purposes, by tracing through a set of mapping layers.

5.6 Privacy Enforcement

Maintaining anonymity or the ability to withhold private information is important in certain service environments. Organizations creating, managing, and using Grid services will often need to state their privacy policies and require that incoming service requests make claims about the service provider's adherence to these policies. The WS-Privacy specification will describe a model for how a privacy language may be embedded into WS-Policy descriptions. The Grid security model should adopt WS-Privacy in addition to WS-Policy to enforce privacy policies in a Grid environment. The general practices and rules defined by the P3P effort [P3P] can prove useful in privacy policy enforcement.

While the authorization and privacy functions in the Grid security model build upon the WS-policy, WS-Authorization and WS-Privacy components, they do so by partitioning policy-related functions into specific functionality by abstracting the expression and exchange of policies from actual policy itself. Mechanisms to express, expose and exchange policies are covered by the policy expression and exchange layer in the proposed Grid security model. Enforcement of policies pertaining to service end-points, federation, authorization and privacy should be built upon WS-SecureConversation, WS-Federation, WS-Authorization and WS-Privacy in the WS security architecture. The security policy management functions are discussed later in this paper.

5.7 Trust

Each member of a VO is likely to have a security infrastructure that includes authentication service, user registry, authorization engine, network layer protection and other security services. The security policies, authentication credentials and

identities belonging to that member organization are likely to be managed, issued and defined within the scope of the organization – i.e., a security domain. In order to securely process requests that traverse between members of a VO, it is necessary for the member organizations to have established a trust relationship. Such trust relationships are essential for services accessed between the members to traverse network checkpoints (e.g., firewalls) and satisfy authorization policies associated with a service achieved by translating credentials from one domain to another (e.g., Kerberos to PKI) and mapping identities across security domains. Therefore, defining and establishing these trust relationships in a Grid environment, i.e. defining VO membership, is a necessary foundation of the security model. Such a model needs to define direct or mutual trust relationships between two domains, as well as indirect trust relationships brokered through intermediaries. These relationships will then often materialize as rules for mapping identities and credentials among the involved organization domains.

The Grid trust model should be based on the web services WS-Trust specification. Importantly, due to the dynamic nature of Grids, trust relationships might also need to be established dynamically using trust proxies that act as intermediaries. Trust can be established and enforced based on trust policies defined either a-priori or dynamically. Once such a model is defined, this will play a role in defining how trust assertions are to be consumed by a service provider or a requester as the case may be. The model will also form the basis to satisfy the requirements to achieve single logon based on trust of asserting authority or trust on requesting member of a VO.

5.8 Secure Logging

The Grid security model explicitly calls for secure logging functionality as the necessary foundation for many higher-level audit-related services. Similar to trust model and security management, secure logging is a basic service that is applicable to other components in the model.

5.9 Management of Security

The Grid security model groups all security management functions applicable to various aspects of binding, policy and federation. These include key management for cryptographic functions, user registry management, authorization, privacy and trust policy management and management of mapping rules which enables federation. It may also include the management of intrusion detection, anti-virus services and assurance information enabling service requestors to discover what security mechanisms and assurances a hosting environment can offer. Addressing the management of various aspects of the security infrastructure will satisfy the manageability requirement on the Grid environment.

The following section captures the standards and technologies as they relate to the proposed Grid security model.

6 Relationship to Security Standards

The Grid environment and technologies address seamless integration of services with existing resources and core application assets. As discussed in the Grid Security

Model section, the Grid security model is a framework that is extensible, flexible, and maximizes existing investments in security infrastructure. It allows use of existing technologies such as X.509 public-key certificates, Kerberos shared-secret tickets and even password digests. Therefore, it is important for the security architecture to adopt, embrace and support existing standards where relevant. Given Grid services

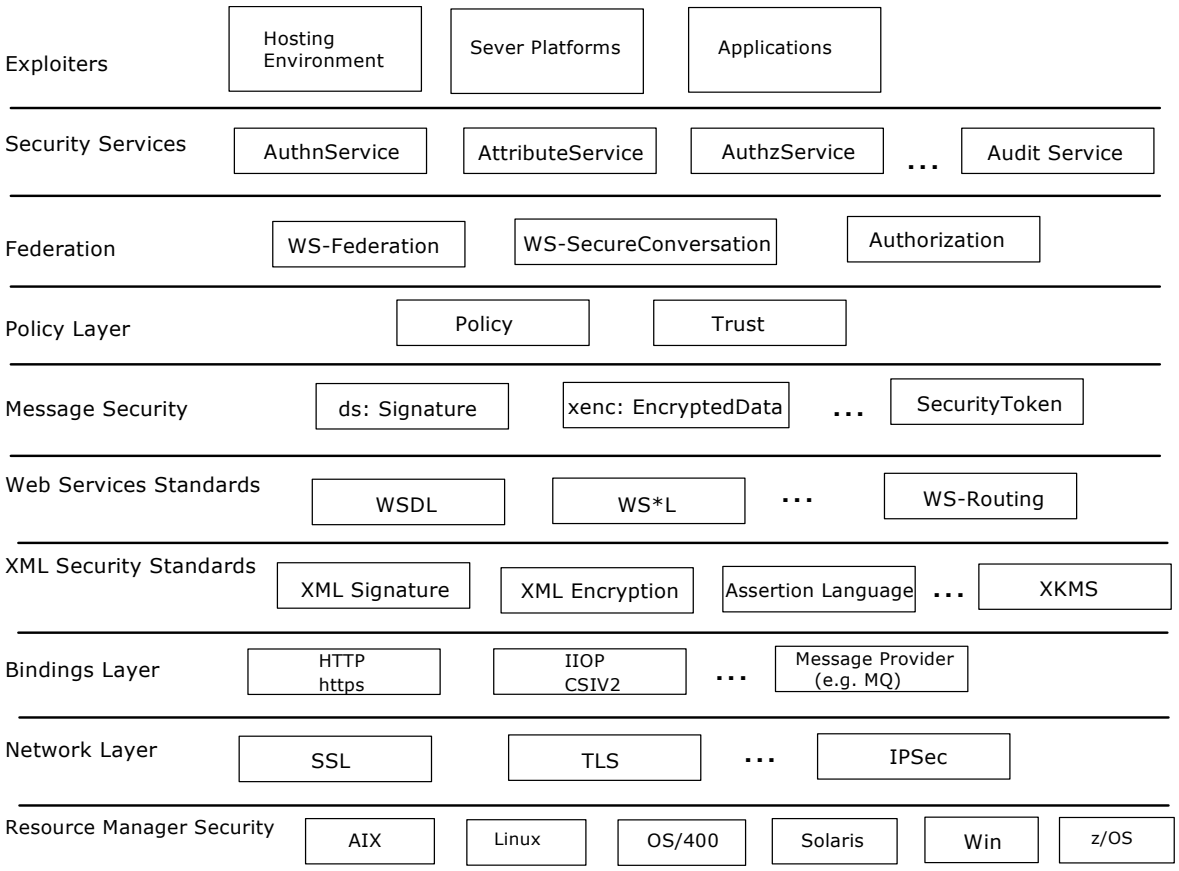

Figure 3: Building blocks for Grid security architecture

are based on web services, Grid security model will embrace and extend the Web services security standards proposed under the WS Security roadmap [WSR].

Specifically, given that OGSA is a service oriented architecture based on Web services (i.e. WSDL based service definitions), the *OGSA security model needs to be consistent with web services security model*. The web services security roadmap [ws-security-roadmap] provides a layered approach to address web services, and also defines SOAP security bindings.

Figure 3 illustrates the layering of security technology and standards that exist today and how they fit into the Grid security model.

In a service-oriented environment, every decision engine should be constructed as a service. But given these services are hosted in a Grid environment, and the hosting environment itself needs to use security technologies and solutions, for bootstrapping reasons, there is a limited set of trust anchors that the system needs to start building on. Such trust anchors can be certificate authorities, trusted Kerberos domains, authorization authorities or simply minimal set of security pieces that are tied to the hosting environment. For example, if configuration is stored as files in the file system, a trust anchor can be the file system security manager itself. Therefore, the security model is based on certain resource managers provided by the hosting environment or the platforms themselves.

As discussed in previous sections, security can be an inherent part of a network and binding layer. In the case of the network layer, IPSec, SSL [SSL] or TLS [TLS] can provide it. In the case of the binding layer, it can be provided by HTTPS; in the case of IIOP, it can be provided by CSIv2. In a messaging environment, the message provider (e.g., MQ) can provide end-to-end message security.

Given the increasing use of XML, the security standards in the XML space play an important role: XML DigitalSignature, XML Encryption, XML Key Management Service (XKMS), and assertion languages (e.g., SAML). Built on top of XML standards are the Web services standards, including WSDL.

Message level security provides means to achieve end-to-end security instead of depending on underlying hop-by-hop security technologies like SSL. In the case of SOAP payloads, security is based on WS-Security and the areas it addresses: digital signature, encryption and security tokens. As described in the Grid security model, the policy layer and the federation layer will be built based on the underlying security layers and technologies.

A number of security functions can be provided through a service oriented approach as well. Security services can be built based on the core underlying technologies and solutions. These services can be built from scratch and exposed, or alternatively existing security functionality (e.g., authorization function embedded in an operating system) can be exposed as an authorization service. Such security services can be exposed as Web services themselves – so that they can be discovered, bound to and invoked. Components that can exploit these services can be broadly categorized to be the platforms (e.g., an AIX platform can use the Kerberos service hosting in a z/OS platform), hosting environment (e.g., a J2EE application server such as IBM's WebSphere can use an external security service through its pluggable authorization framework), or applications themselves (e.g., a financial application may have its own payload digitally signed using a digital signature service).

As illustrated in the building blocks of Grid security model, and described above, existing and evolving standards will be adopted or recognized in the Grid security model. As many different environments will need to interoperate, the technologies

each hosting environment uses can be exposed as part of its policy so that interoperability can be achieved.

7 Security as Services

To achieve integration and interoperability while securing Grid services, existing security technologies may be (re)used. Exposing existing security solutions as services, as well as building new required security functions as services will achieve a level of abstraction that helps provide an integrated, secure Grid environment. Like any other service, security services should be exposed as web services (i.e., with a WSDL definition) and should expose functionality while hiding implementation details. No specific security technology should be hard-coded in order to secure these services. Services must be secured using the Grid security model (e.g., a service request must be protected using WS-Security) and thus be protected using a variety of security mechanisms supported in hosting environments in which they will be deployed.

Given that a request to a security service can be intra-domain or inter-domain, the hosting environment will need to take federation into account. For example, a user registry service must take into account that it may need to evolve and support federation with other registry services.

An OGSA infrastructure may use a set of primitive security functions in the form of services themselves. A set of Grid security services may include:

- An authentication service: An authentication service is concerned with verifying proof of an asserted identity. One example is the evaluation of a User ID and password combination, in which a service requestor supplies the appropriate password for an asserted user ID. Another example involves a service requestor authenticating through a Kerberos mechanism, and a ticket being passed to the service provider's hosting environment, which determines the authenticity of the ticket before the service is instantiated.

- Identity mapping service: The identity mapping service provides the capability of transforming an identity which exists in one identity domain into a identity within another identity domain. As an example, consider an identity in the form of an X.500 Distinguished Name (DN), which is carried within a X.509 V3 digital certificate. The combination of the subject DN, issuer DN and certificate serial number may be considered to carry the subject's or service requestor's identity. The scope of the identity domain in this example is considered to be the set of certificates that are issued by the certificate authority. Assuming that the certificate is used to convey the service requestor's identity the identity mapping service via policy may map the service requestor's identity to a identity which has meaning (for instance) to the hosting environment's local platform registry. The identity mapping

service is not concerned with the authentication of the service requestor; rather it is strictly a policy driven name mapping service

- Authorization service: The authorization service is concerned with resolving a policy based access control decision. The authorization service consumes as input a credential which embodies the identity of an authenticated service requestor and for the resource that the service requestor requests, resolves based on policy, whether or not the service requestor is authorized to access the resource. It is expected that the hosting environment for OGSA compliant services will provide access control functions, and it is appropriate to further expose an abstract authorization service depending on the granularity of the access control policy that is being enforced.

- VO Policy service: The VO policy service is concerned with the management of policies. The aggregation of the policies contained within and managed by the policy service comprises a VO's policy set. The policy service may be thought of as another primitive service, which is used by the authorization, audit, identity mapping and other services as needed.

- Credential Conversion service: The credential conversion service provides credential conversion between one type of credential to another type or form of credential. This may include such tasks as reconciling group membership, privileges, attributes and assertions associated with entities (service requestors and service providers). For example, the credential conversion service may convert a Kerberos credential to a form which is which is required by the authorization service. The policy driven credential conversion service facilitates the interoperability of differing credential types, which may be consumed by services. It is expected that the credential conversion service would use the identity mapping service.

- Audit Service: The audit service similarly to the identity mapping and authorization services is policy driven. The audit service is responsible for producing records, which track security relevant events. The resulting audit records may be reduced and examined as to determine if the desired security policy is being enforced. Auditing and subsequently reduction tooling are used by the security administrators within a VO to determine the VO's adherence to the stated access control and authentication policies.

- Profile Service: The profile service is concerned with managing service requestor's preferences and data which my not be directly consumed by the authorization service. This may be service requestor specific personalization data, which for example can be used to tailor or customize the service requestor's experience (if incorporated into an application which interfaces with end-users.) It is expected that primarily this data will be used by applications which interface with a person.

- Privacy Service: The privacy service is primarily concerned with the policy driven classification of personally identifiable information (PII). Service providers and service requestors may store personally identifiable information using the Privacy Service. Such a service can be used to articulate and enforce a VO's privacy policy.

8 Use Patterns

This section discusses the application of the proposed roadmap to a common use pattern. This use pattern illustrates an end-to-end security flow from service requestor to a service provider – highlighting the security aspects involved in request flows between the requestor and the service provider, as well as the security aspects involved when service requests are delegated between services.

8.1 Typical e-business Use Pattern

Bob the traveler surfs the web looking for the best possible hotel rates through a travel agency, BagsPacked travel. In order to request travel services from the agency, which includes automated hotel reservations based on user supplied criteria, Bob opens an account with BagsPacked Travel. This is accomplished by the invocation of the RegisterNewUser service, which entails providing a payment authorization token (e.g., credit card number) and the hotel selection criteria.. The payment authorization token has a limited lifespan, which enables Bagspacked travel to work on behalf of Bob for a limited period of time. Bob supplies the criteria that include the scheduled arrival and departure dates, and a price ranges that Bob is willing to pay for the room.

As shown in Figure 4, Bob submits the request over a firewall friendly protocol (e.g., https) connection. Bob, as the service requestor, trusts his connection to the travel agency based on access policy that Bob has previously established. The RegisterNewUser service hosted by its Sales department in turn securely invokes the Accounting department's CreateNewUser service. The trust basis for the invocation is the trust relationship established through the Kerberos authentication mechanism used by both the organizations in BagsPacked Travel.

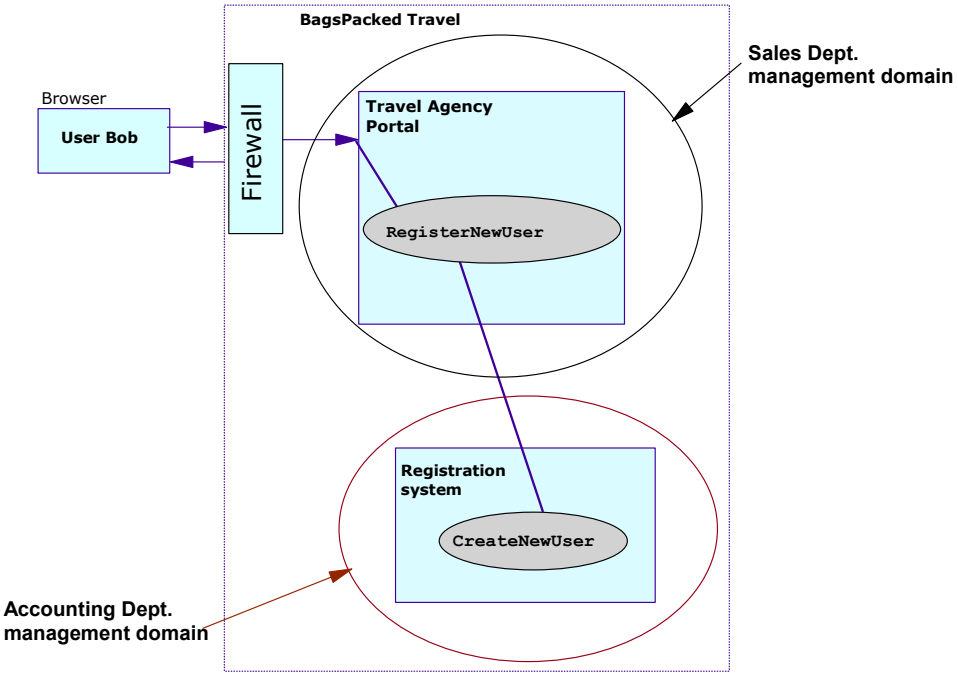

Figure 4: Service requests within a virtual organization

As a part of the account creation process, the CreateNewUser service evaluates the authorization token supplied by Bob. For future transactions that involve booking of hotel rooms, Bob will supply on the service request a authorization, plus the criteria required to carry out the hotel reservation.

The sales and account departments of the travel agency share resources and services in order to perform their business effectively. Even though they may have organizational boundaries reflected through their network or security domains, they form a virtual organization based on their trust relationship and through sharing of their resources. This trust relationship is statically defined by using Kerberos as the mechanism of establishing trust between these two organizations in the travel agency. However, between Bob and the travel agency, a dynamic trust relationship has been established. This trust is based in an authorization token – a limited life span credential, which Bob has delegated to the travel agency that enables the agency to reserve a hotel room on his behalf. In this interaction between Bob and the travel agency a virtual organization has been dynamically created which crosses organizational boundaries.

Once the authorization token is successfully validated, BagsPacketTravel process the criteria and selects a hotel that meets the criteria established by Bob. As illustrated in Figure 5, the BookHotelReservation service hosted by the travel agency, dynamically looks up a hotel reservation service, ReserveRoom, in order to complete the request. Among a possible set of resulting service instances, the BookHotelReservation service chooses one instance, hosted by SleepyLand Hotels,

based on a set of policy constraints which may have been further refined by the union of the criteria which Bob supplied, and the policies of the travel agency.

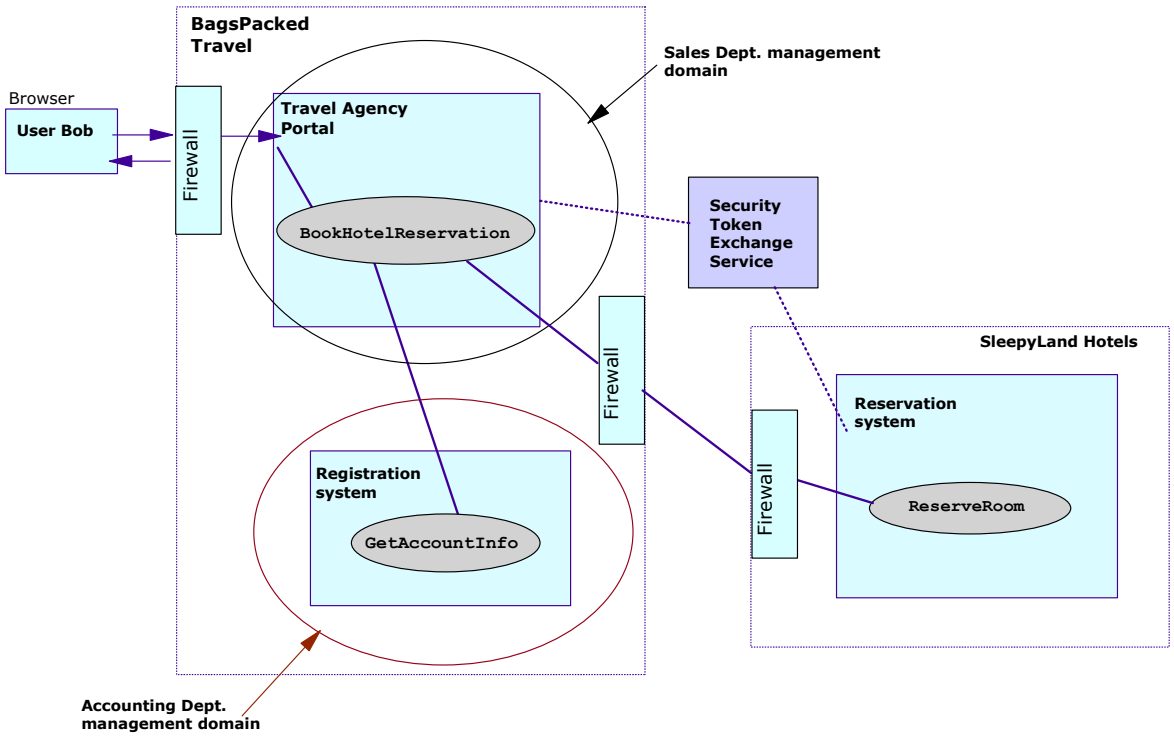

Figure 5: Service requests across virtual organizations

The SleepyLand Hotels' services are secured using a public key infrastructure. They can accept X509 certificates issued by a trusted certificate authority in order to access their services. This information is published through the security policies associated with the ReserveRoom service.

In order to invoke SleepyLand's service, the BookHotelReservation service talks to a security token exchange service hosted by a mutually trusted third party. It obtains an X509 certificate in exchange for the travel agent's Kerberos ticket. It then submits the request to reserve a hotel room using the X509 certificate issued by the exchange service, and includes with the request the payment authorization token which supplied by Bob on his initial request to the travel agency. Upon successful validation of the certificate and enforcing an appropriate authorization policy, the ReserveRoom service makes a room reservation for Bob. The result is sent back to the end user through the BookHotelReservation service.

The travel agency and the hotel chain have their own virtual organization where they share resources and service. Though they have a loosely coupled trust relationship, requests must traverse between these virtual organizations. The difference in security mechanisms and trust models between these VOs and the dynamic invocation pattern illustrated in this use case highlight a set of security challenges to be addressed in the Grid environment.

8.2 Scenario Involving Intermediaries

Figure 6 illustrates a flow from an end user requesting a target service, where the request traverses through intermediaries. Assume that a user, wishes to invoke a Grid service, e.g. at an intermediary node which will eventually result in accessing some resource at some N-step remote service provider (right in the figure).

The user can obtain a credential by authenticating to an authentication server, local to his domain, and present that credential as part of the service request. When the request gets routed through a gateway, the gateway may consult an attribute server

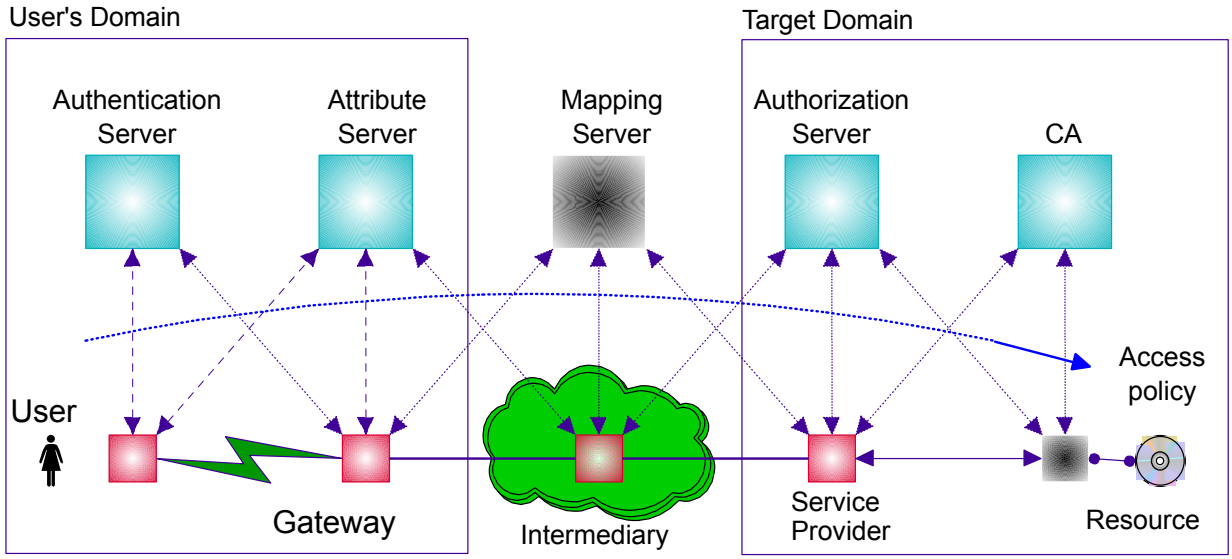

Figure 6: Service requests that traverse through intermediaries

to obtain the user's privilege attributes and rights and send the assertions with the request.

Such a request may be routed through some intermediary, which can convert the assertions into a form understood by the target domain (e.g. based on WS-Federation). For example, the intermediary may convert the authentication credential (e.g., Kerberos ticket) into a credential form that target domain can work with (e.g., X509 certificate). Additionally, intermediary can honor a set of policies when forwarding the request, including the mapping rules and delegation policies.

When the request is received, the target can validate the certificate. Upon successful validation, it can derive an identity based on the certificate and make authorization decisions using the locally defined authorization policies. This example illustrates the value of looking at the problem of starting from an authentication credential in one domain and going through delegation and mapping to a target server, that makes authorization decision based on credential mapping performed by an intermediary.

This roadmap suggests an approach for a set of standardized OGSA compliant services which addresses the needs of credential exchange and propagation with WS-Security, policy services which may be used by the work hosting environment based on WS-Policy, the needs of validating and mapping identities with WS-Federation, and lastly secure session and context establishment with WS-SecureConversation.

These scenarios did not address specific bindings. This roadmap suggests a binding and implementation agnostic architecture that fosters the interoperability between heterogeneous systems, addressing the emerging needs of the Grid environment reflected in a business scenario.

9 Summary

As Grid services are adopted and applied widely, as definition of an organizational boundary fade away in support of virtual organizations, as virtual organizations continue to evolve to support intermediaries such as firewalls, load balancers, and messaging hubs, and as awareness of the threats organizations face becomes more well understood, the need for a security architecture within OGSA grows clear.

In this document, we propose a comprehensive OGSA security architecture and a set of security components that encapsulate the required security functionalities. This paper reflects the challenges and requirements we have identified thus far in an OGSA environment. Based on those requirements, the proposed security architecture, by extending and leveraging (rather than replacing) existing security technology and assets, will enable businesses and organizations to more rapidly develop secure, interoperable Grid services.

10 Terminology

Because terminology varies between technologies, this document defines several terms that may be applied consistently across the different security formats and mechanisms. Consequently, the terminology used here may be different from other specifications and is defined so that the reader can map the terms to their preferred vocabulary.

- **Grid Service -** A *Grid service* supports a set of standard interfaces for reliable invocation, lifetime management, access control, notification, and upgradeability. The characteristics of a Grid service are discussed in [PSY] and [GRIDSPEC]. A service that adheres to these conventions we call a *Grid service*.

- **Web service** – The term "Web service" is broadly applicable to a wide variety of network based application topologies. In this document, we use the term "Web service" to describe application components whose functionality and interfaces are exposed to potential users through the application of existing and emerging Web technology standards including XML, SOAP, WSDL, and HTTP. In contrast to Web sites, browser-based interactions or platform-dependent technologies, Web

services are services offered computer-to-computer, via defined formats and protocols, in a platform-independent and language-neutral manner.

- **Domain –** Typically used in the context of this paper to refer to a collection of services which are managed and shared within some organizational boundary, which may include a physical organizational boundary, a virtual organization boundary as scoped by a registry, etc.

- **Subject** – The subject of the security token is a principal (e.g., a person, an application or a business entity) about which the claims expressed in the security token apply. Specifically, the subject, as the owner of the security token possesses information necessary to prove ownership of the security token.

- **Grid Service Policy** – Web services have complete flexibility in specifying the claims they require in order to process messages. Collectively we refer to these required claims and related information as the "Web Service Endpoint Policy". Endpoint policies may be expressed in XML and can be used to indicate requirements related to authentication (e.g. proof of user or group identity), authorization (e.g. proof of certain execution capabilities), or other custom requirements.

- **Intermediaries** – As SOAP messages are sent from an initial requester to a service, they may be operated on by intermediaries that perform actions such as routing the message or even modifying the message. For example, an intermediary may add headers, encrypt or decrypt pieces of the message, or add additional security tokens. In such situations, care should be taken so that alterations to the message do not invalidate message integrity, violate the trust model, or destroy accountability.

- **Bindings –** Protocol and message format that facilitates invocation of a service.

- **Virtual Organization** A set of individuals and/or institutions defined by such sharing rules form what we call a *virtual organization* (VO).

11 Acknowledgements

We are thankful for the valuable comments from Brian Carpenter, Francis Hildenbrand, Jeffrey Nick, Jeffrey Frey, Karl Czajkowski, Carl Kesselman, Laura Pearlman, Sam Meder, Doug Engert and many others.

12 References

[ANA] **The Anatomy of the Grid: Enabling Scalable Virtual Organizations**. I. Foster, C. Kesselman, S. Tuecke. *International J. Supercomputer Applications*, 15(3), 2001.

[PSY] **The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration**. I. Foster, C. Kesselman, J. Nick, S. Tuecke; January, 2002.

[COMP] **A Security Architecture for Computational Grids**. I. Foster, C. Kesselman, G. Tsudik, S. Tuecke. *Proc. 5th ACM Conference on Computer and Communications Security Conference*, pp. 83-92, 1998.

[NEUMAN] Lai, C., Medvinsky, G. and Neuman, B.C. Endorsements, Licensing, and Insurance for Distributed System Services. in *Proc. 2nd ACM Conference on Computer and Communication Security*, 1994.

[WSR] **Security in a Web Services World: A Proposed Architecture and Roadmap,** http://www-106.ibm.com/developerworks/library/ws-secmap/

[SSL] **The SSL Protocol Version 3.0**. http://home.netscape.com/eng/ssl3/draft302.txt.

[TLS] RFC 2246: **The TLS Protocol**. ftp://ftp.isi.edu/in-notes/rfc2246.txt.

[CORBA] **The Common Object Request Broker: Architecture and Specification, Version 2.3.1.** The Object management Group (OMG), http://www.omg.org/cgi-bin/doc?formal/99-10-07.

[CSI] **Common Secure Interoperability Version 2 Final Available Specification.** The Object Management Group (OMG), http://www.omg.org/cgi-bin/doc?ptc/2001-06-17.

[J2EE] **Java 2 Platform, Enterprise Edition, v1.3 (J2EE)**. http://java.sun.com/j2ee.

[P3P] **The Platform for Privacy Preferences 1.0 (P3P1.0) Specification, W3C Recommendation** 16 April 2002, http://www.w3.org/TR/P3P/

[GRIDSPEC] **Grid Service Specification**. S. Tuecke, K. Czajkowski, I. Foster, J. Frey, S. Graham, C. Kesselman; Draft 2, 6/13/2002, http://www.globus.org

13 Contact Information

Nataraj Nagaratnam
IBM Corporation
4205 S Miami Blvd
Research Triangle Park, NC, 27703
Email: natarajn@us.ibm.com

Philippe Janson
IBM Corporation
Zurich Research Laboratory
SAEUMERSTRASSE 4
RUESCHLIKON, 8803
Email: pj@zurich.ibm.com

John Dayka
IBM Corporation
2455 South Road
Poughkeepsie, NY 12601
Email: dayka@us.ibm.com

Anthony Nadalin
IBM Corporation
9442 Capitol of Texas Highway North
Austin, TX 78759
Email: drsecure@us.ibm.com

Frank Siebenlist
Argonne National Laboratory
236 More Avenue, Los Gatos, CA 95032
Email: franks@mcs.anl.gov

Von Welch
University of Southern California, Information Sciences Institute
Email: welch@mcs.anl.gov

Ian Foster
Argonne National Laboratory & University of Chicago
Email: foster@mcs.anl.gov

Steven Tuecke
Distributed Systems Laboratory
Mathematics and Computer Science Division
Argonne National Laboratory
Argonne, IL 60439
Phone: 630-252-8711
Email: tuecke@mcs.anl.gov