GWD-I (draft-ggf-ogsa-sec-roadmap-01)
GGF OGSA Security Workgroup
http://www.ggf.org/ogsa-sec-wg

Frank Siebenlist, ANL
Von Welch, UC
Steven Tuecke, ANL
Ian Foster, ANL
Nataraj Nagaratnam, IBM
Philippe Janson, IBM
John Dayka, IBM
Anthony Nadalin, IBM

July, 2002

OGSA Security Roadmap

Global Grid Forum Specification Roadmap towards a Secure OGSA

Status of this Memo

This memo is submitted to the GFD Editor for consideration to be published as a Informational Document (GFD-I). Distribution is unlimited. The latest version of this document may be found at: http://www.gridforum.org/ogsa-sec-wg/

Abstract

This document is a roadmap enumerating a set of proposed specifications to be defined in the Global Grid Forum in order to ensure interoperable implementations of the OGSA Security Architecture. The specifications in this roadmap leverage existing and emerging Web Services security specifications.

Table of Contents

1. Introduction

This document describes a set of proposed OGSA security specifications to be defined in one or more working groups in the Global Grid Forum (GGF)[GGF]. These specifications, when combined with existing and emerging Web services (WS) security specifications and emerging Open Grid Services Architecture (OGSA) specifications [OGSA, GSS Spec], can be used to implement the OGSA security architecture as described in [OGSA Security Arch].

This roadmap leverages other WS security specifications as much as possible. In situations where other specifications are still emerging and will not be available at the time they are needed, tactical solutions are proposed that should be reconciled with the appropriate specification when they become available.

The specifications described in this roadmap are intended to be agnostic towards existing security mechanisms. To facilitate adoption, the specifications should seek solutions that allow implementations by as many existing security mechanisms and bindings as possible.

We propose that this security roadmap be maintained as a living document that reflects the progress made in the GGF and that adapt to changes in the evolving WS and OGSA specifications. We also expect that over time, as problems are better understood and other specifications evolve, the need will arise for more specifications not contained in this roadmap. Also, as other specifications emerge and evolve, it is possible that some specifications proposed in this document may no longer be necessary. Changes to this roadmap will be made as required.

This document is structured as follows. First, we give a short overview of the OGSA Security Architecture as described in [OGSA Security Arch](Section 2). Then we discuss WS Security specifications that we believe can be leveraged to implement this architecture (Section 3) and enumerate the specifications we believe need to be completed in the GGF to complete any standardized implementation of the security architecture (Section 4).

2. OGSA Security Architecture Summary

A detailed description of the OGSA security architecture is given in the "Open Grid Services Security Architecture" document [OGSA Security Arch]. This section gives a short summary to introduce the different components and to show how they are related to existing and emerging WS security specifications.

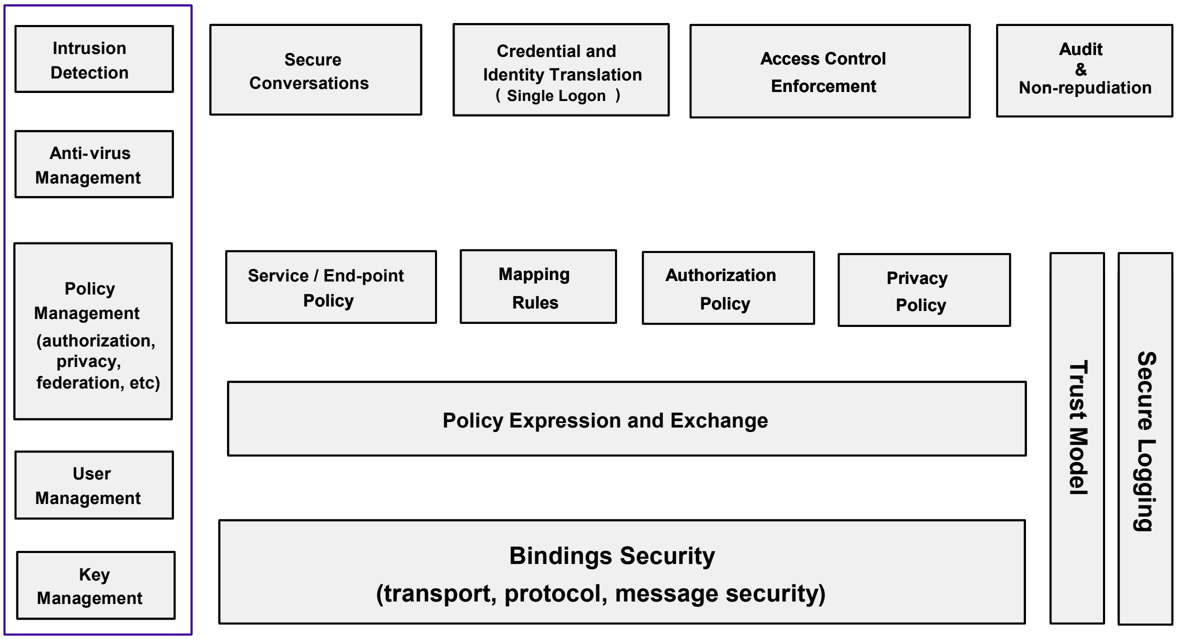
Intrusion Detection
Anti-virus Management
Policy Management (authorization, privacy, federation, etc)
User Management
Key Management
Secure Conversations
Credential and Identity Translation (Single Logon)
Access Control Enforcement
Audit & Non-repudiation
Service / End-point Policy
Mapping Rules
Authorization Policy
Privacy Policy
Policy Expression and Exchange
Bindings Security (transport, protocol, message security)
Trust Model
Secure Logging

Figure 1. Security Component Layering

The different components needed for the secure deployment of a distributed environment are shown in Figure 1. In this layering, application-specific components such as "secure conversation," "credential and identity translation," "access control enforcement," and "audit and non-repudiation," depend on policies and rules for "authorization," "privacy," "identity/credential mapping," and "service/end-point." In order to apply and manage these policies, one needs languages for "policy expression and exchange" and means to securely communicate through bindings to transport protocols. On the side,one can identify a "trust model" component that defines where the trust anchors are, and how trust is derived. The "secure logging" component is a requirement for the auditing of any policy decision.

Finally, the left box in the picture groups components that are required for the management and administration of the infrastructure. These management components are also subject to policy enforcement.

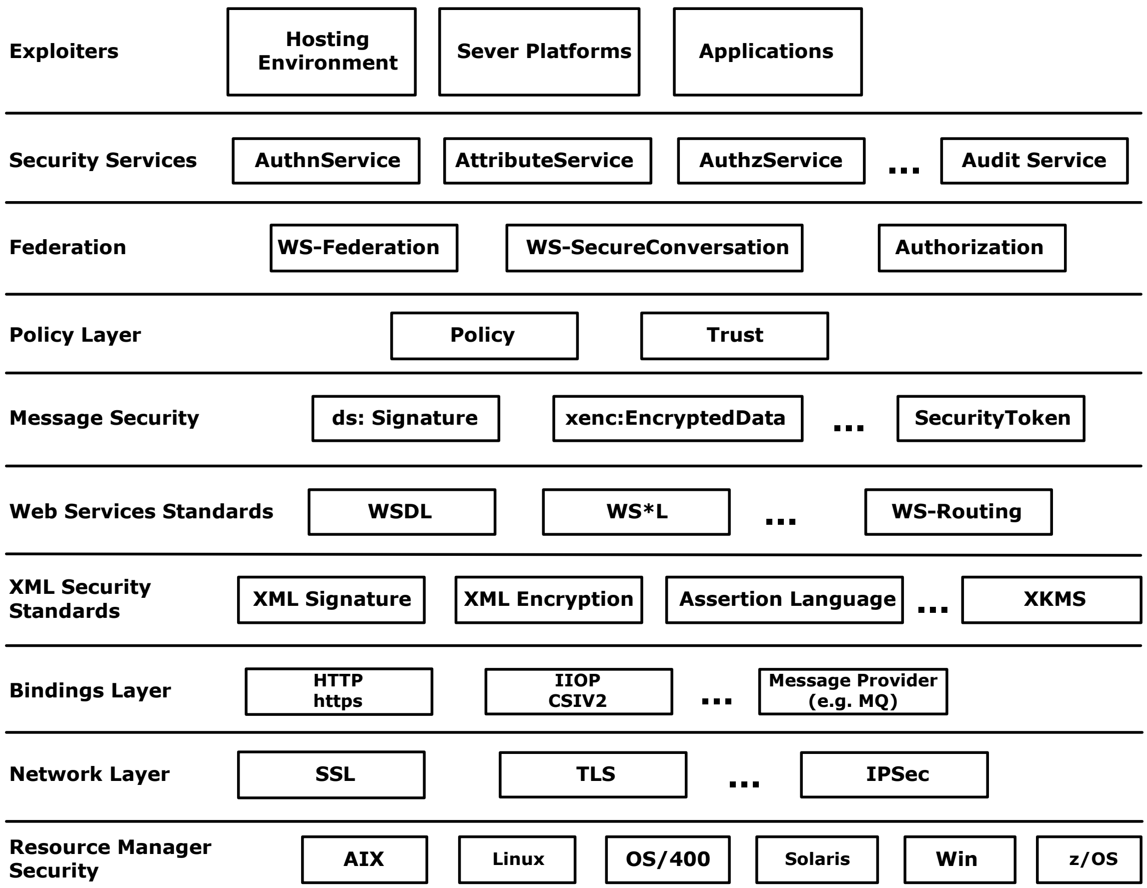

Figure 2. Security Specifications "Stack"

Figure 2 illustrates the layering of existing security technologies and standards and shows how these fit into the Grid security model. Moving from the machine and OS security on the bottom to the applications and server environment at the top, one can identify different layers that either are built and depend on their lower neighbors, or are a level up in abstraction.

The same or similar functions can be implemented at different levels, with different characteristics and tradeoffs. For example, security can be an inherent part of a network and binding layer. In the case of the network layer, it can be provided via IPSec or SSL/TLS. In the case of the binding layer, it can be provided by HTTPS and in the case of IIOP, by CSIv2 [CORBA, CSI]. 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 here: XML Digital Signature [XML-Signature], XML Encryption [XML-Encryption], XML Key Management Service (XKMS) [XKMS], and assertion languages (e.g., SAML [SAML]). Built on top of XML standards are the Web services standards, including WSDL [WSDL].

Alternatively, or as well, message-level security mechanisms can be used to achieve end-to-end security instead of depending on underlying hop-by-hop security technologies like

SSL/TLS [SSL, TLS]. In the case of SOAP payloads, security is based on WS-Security [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.

As illustrated in Figure 2 and described above, the Grid security model will adopt and build on a variety of existing and evolving standards. As many different environments will need to interoperate, the technologies used within a particular hosting environment can be exposed as part of its policy so that interoperability can be achieved.

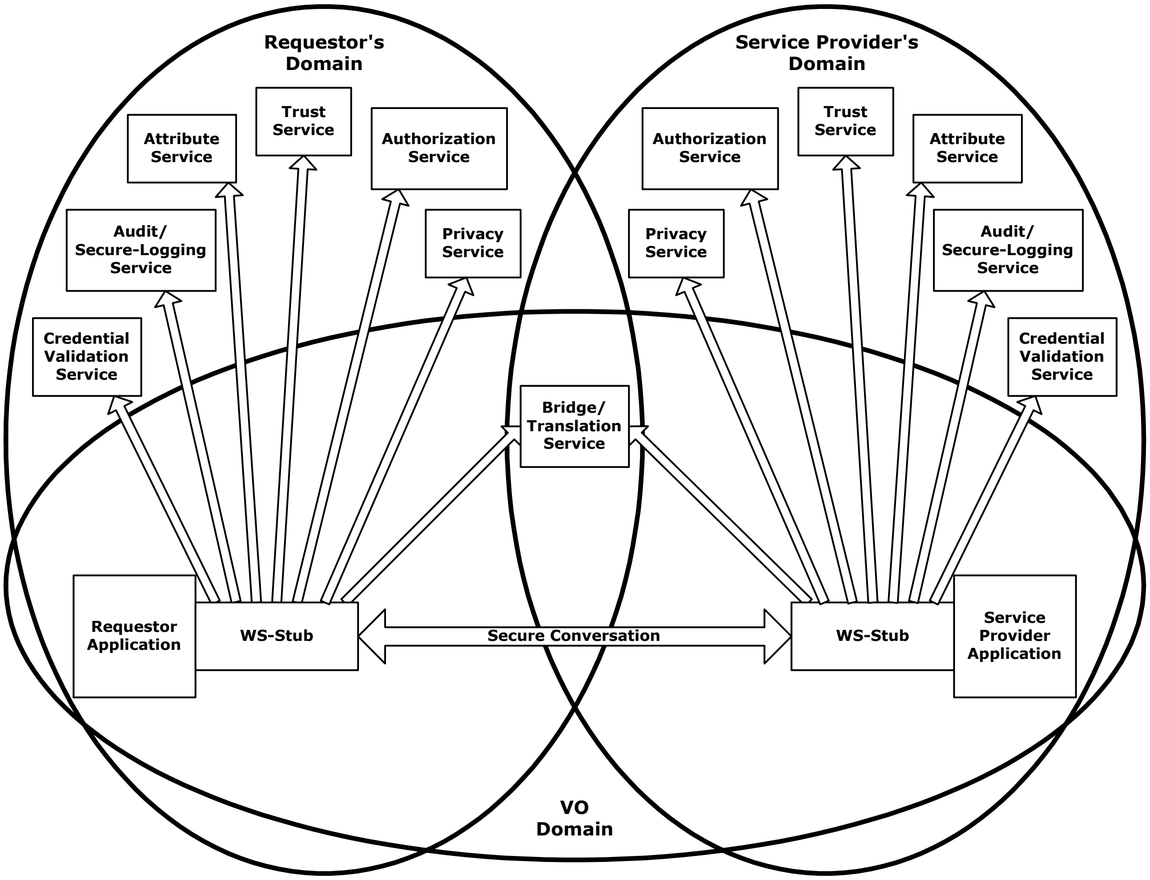

Figure 3. Grid Security Services

Key relationships among requestor, service provider, and many of the security services are depicted in Figure 3. Here, we assume a Virtual Organization setup in which the requestor and service provider are each subject to the policy set in their respective domain, while the Bridge/Translation Service has credentials in both domains and is able to federate the requestor and service provider by issuing different identity and capability assertions that can be validated in each domain.

All the call-out interfaces to the security services from the requestor and service provider, indicated by outgoing arrows in Figure 3, must be specified in terms of OGSA interfaces.

Compliant implementations can make use of existing services and defined policies through configuration. Compliant security service implementations of a particular security related service type can provide the associated and possibly alternative security services.

All security service providers are also OGSA-compliant services, which means that they adhere to the same specified serviceTypes and associated portTypes as normal OGSA service providers. Furthermore, all security service providers are subject to the same policy enforcement as application service providers and requestors.

3. Web Service Security Specifications

This roadmap proposes a set of OGSA security specifications that leverage heavily existing and emerging WS security specifications, building in particular on the framework described in the WS Security Architecture [WS-Security]. The WS Security Architecture describes a framework of layered modules: WS-Security, WS-Policy, WS-Federation, WS-SecureConversation, WS-Privacy, WS-Trust and WS-Authorization. Specifications written for these modules will serve as building blocks for the OGSA security specifications proposed in this roadmap.

Other specifications that may be of interest to the OGSA Security community for possible leverage are XKMS [XKMS], SAML [SAML], XrML [XrML], XACML [XACML], WS-Routing [WS-Routing], WS-Referral [WS-Referral], XML-Signature [XML-Signature] and XML-Encryption [XML-Encryption].

Some of these WS specifications define a generic framework for which specific profiles must be defined to describe message formats and interfaces. The definition of these profiles is essential to guarantee interoperability and pluggability. A number of the specifications proposed in this roadmap will define profiles to fit these WS security specifications.

It is expected that some WS security specifications will not meet all OGSA security requirements in the particular area they are addressing. In this case, those WS security specifications must be modified to enable integration into the OGSA Security framework. These modifications will have to be described in GGF specifications and parallel effort should be made to work with the appropriate standards organization to reconcile the differences.

As a number of WS-* specifications are emerging, some may not be available at the time they are needed to draft specifications in this roadmap. In those cases, tactical solutions should be generated that allow the OGSA security specification to be generated. These tactical solutions should be documented as specifications in the GGF, and be reconciled as the associated WS security specifications become available.

4. OGSA Security Specifications

This section describes the OGSA security specifications that we propose be defined within the GGF. Each of the subsections specifies first a high-level goal followed by one or more subsections, refining the goal and describing the proposed specifications needed to address a portion of the goal.

The proposed OGSA security specifications leverage existing and emerging specifications and standards as described in Section 3. These OGSA security specifications have a circumscribed purpose to extend and modify other (in particular, WS security) specifications as needed to meet OGSA-specific requirements. In situations when OGSA security specifications in the critical path of the overall OGSA roadmap rely on other specifications that are expected but have not yet emerged, tactical specifications are proposed with the intention that they will be later reconciled.

A number of the OGSA security specifications proposed in this roadmap will specify an OGSA service. In the scope of this document an OGSA service is defined as the WSDL defining a serviceType as defined in [OGSA spec] as well as the semantics for a service implementing that serviceType.

4.1. Naming

The OGSA Security Architecture document defines a number of requirements that demand that OGSA be able to assign names to users, services, groups, attributes, and actions (methods). In particular:

- **Authorization enforcement:** It is expected that most policy evaluation implementations will require names for the requestor, service provider, their attributes, the requested action, etc., in order to perform the evaluation. In order to create a standard interface to arbitrary policy evaluation services for use by OGSA services, the forms of these names need to be standardized. In addition to make sure that policy evaluation is done correctly, this dictates that names be unique across different realms.
- **Attribute binding:** Attributes are often bound to an entity via the name of the entity. Since the binding can be done by one entity and then evaluated by a different entity, possibly in a different realm, the method for expressing names needs to be consistent.
- **Auditing:** The names of entities and actions, in particular, will often be put into audit logs. Since the entity doing the logging may be different than the entity parsing the audit log, the method for expressing names needs to be consistent.

While in many cases an existing mechanism will provide a name—in particular, users and services will have identities from their authentication credentials—it must be specified how the name from the mechanism should be canonicalized for use in OGSA.

Recall that OGSA allows for the dynamic creation of stateful transient services. This capability introduces another challenge. Service requestors must be able to establish trust of these transient services since they will potentially send them sensitive data or delegate credentials to them as part of a request. Thus, these services need unique, assertable identities so that requestors can make authorization decisions either based on the identity itself and/or on attributes that are associated with the identity.

We propose four specifications to address the naming requirements discussed: one for the naming of entities, one for naming targets or actions on OGSA services, one for naming attributes and groups and one for the naming of transient services.

4.1.1. OGSA Identity Specification

Names for OGSA entities (users or services) are required for all requirements listed above. Authorization enforcement will require names for the entities (users or services) involved in the request – the requestor and the service provider. Auditing will require names for the entity in the audit log. And attribute binding will need names for the entity to which the attribute is being bound. Because of this, a standard means for naming entities is seemed as a priority.

This specification defines how the identity (i.e. name) for an OGSA entity should be formed based on the entity's identity established by within their security realm. This specification should consider the following issues:

- **Cross-realm uniqueness:** A unique name from one realm is not necessarily guaranteed to be unique across all realms. The name will need appropriate canonicalization to make sure it is unique.
- **Anonymity:** There will be scenarios where anonymous usage is allowable and even desirable. The naming scheme should enable this.
- **Identity Mapping:** Policy may dictate that an entity is known by multiple names (e.g. ,it may have both a Kerberos principal name and a PKI subject name). This specification should be careful not to hinder policies and services that express and perform these mappings.

4.1.2. OGSA Target/Action Naming Specification

Many authorization policies will also require a name for the action being invoked. This specification defines how an action (i.e., a request from a requestor to a service provider) is described.

For coarse-grained policy evaluation, this description may be nothing more than the name of the portType on the named service.

For finer-grained policy evaluation, this description may need to include a representation of the arguments being used to invoke the action.

This specification should describe a name format for describing actions in a standard manner to allow for pluggable policy evaluation modules.

(Current work to identity serviceDataDescriptions in the larger OGSA community may resolve portions of this requirement.)

4.1.3. OGSA Attribute and Group Naming Specification

In order to allow attributes and groups defined in one realm to be used in other realms, a standard method of expressing these attributes and groups names is required. This will allow writers and evaluators of a policy to have a consistent approach for naming attributes and groups and ensure that the intended attribute or group is used by both parties. This specification should describe an appropriate canonicalized naming method for group and attribute names.

4.1.4. Transient Service Identity Acquisition Specification

This specification defines a method that a transient service instance can use to obtain an unique identity. It is possible that this specification will entail the description of one or more OGSA services.

This specification might consider the following approaches:

- **Factory/Hosting environment granted:** The service factory or hosting environment act as a naming authority for instances they create. This approach essentially amounts to the factory or hosting environment defining a namespace.
- **Requestor granted:** The requestor that requested the instantiation of the service instance grants it a name. This approach is essentially what occurs with proxy certificates in the 2.0 version of the Globus Toolkit.
- **Self-named:** The instance generates its own name either using a standard UUID method or by using some other unique (or statistically unique) identifier such as a public key.

4.2. Translating between Security Realms

As described in the OGSA Security Architecture document, entities need the ability to translate between security realms—for example, to request an action at a remote service provider. This translation may lead to the entity encountering different security mechanisms, different organizations with different namespaces and trust roots, or both. To address the difficulties that arise in this context, we must define OGSA services for converting identities, names and policies between realms as well as services for converting credential formats.

4.2.1. Identity Mapping Service Specification

This specification defines an OGSA service that allows a client requestor to determine what identity mappings are allowed, by policy, for a particular pair of realms. This specification addresses a critical issue for cross-realm interoperability and is seen as an appropriate first step. This service also needs a management interface that allows appropriately authorized entities to manage this policy.

4.2.2. Generic Name Mapping Specification

This specification generalizes the previous specification to define an OGSA service for mapping any sort of defined name for groups, attributes, actions, etc.

4.2.3. Policy Mapping Service Specification

Building on the previous name mapping specification, this specification defines an OGSA service for mapping policies between security realms. This specification should consider the WS-Policy specification.

4.2.4. Credential Mapping Service Specification

This specification defines an OGSA service that enables the conversion of credentials from one security realm to another in order to enable inter-realm interoperability. It is probable that such a service would also find use as an intra-realm credential conversion service (e.g. to allow requestors to obtain credentials for different identities or rights). So this specification should seek to allow this as well. In is envisioned that this service could use the identity mapping service to help with policy decisions, though it still may require it's own policy management interface.

4.3. Authentication Mechanism Agnostic

OGSA must support multiple authentication mechanisms, including Public Key Infrastructure (PKI) and Kerberos. It will be desirable for OGSA implementations to support multiple mechanisms concurrently in order to support bridging of authentication domains.

Public Key Infrastructure.

We want to enable OGSA implementations to work with different PKI variants: in particular, X.509 [X509] but also PGP [PGP], AADS [AADS] and SPKI [SPKI]. In all cases, an OGSA implementation has to be able to adhere to local certificate path validation policies in order to function in environments where PKI systems are already deployed.

To meet this PKI-agnostic goal, a certificate validation service interface should be defined that can be used within OGSA implementations to:

- Parse a certificate and return desired attribute values.
- Perform path validation on a certificate chain according to the local policy and with local PKI facilities, such as certificate revocation lists (CRLs) or through an online certificate status protocol [OCSP].
- Return attribute information for generic KeyInfo values, thus allowing one to use different certificate formats or single keys, or to pull attribute information from directory services instead of certificates.

Kerberos.

Kerberos [KERBEROS] uses its own message format and protocol for authentication and ticket requests. To facilitate the use of Kerberos in the OGSA context, we can wrap Kerberos protocol messages for authentication and ticket requests in OGSA WS protocols, so that client-to-Kerberos server communications use the same wire protocol, routing capabilities, and discovery mechanisms as client-to-PKI server communications. In this way, we also leverage the OGSA WS ability to route more easily through firewalls and network address translation (NAT) systems.

4.3.1. Certificate Validation Service Specification

One standard that may meet the listed goals for an Certificate Validation Service is XKMS [XKMS]. XKMS has the potential to bind attribute to Public Keys as well as to Kerberos principals.

Further investigation is needed to see if XKMS provides all the required features for such an assertion validation service. The XKMS-specification is currently being revised within the W3C [W3C].

4.3.2. OGSA-Kerberos Services Specifications

This specification defines OGSA services to enable the tunneling of Kerberos authentication protocols.

4.4. Pluggable Session Security

A session-based security solution that communicates on the message level is needed for conversations that require multiple request/reply exchanges and in which the security context may extend through one or more intermediaries. For connection-based protocols such as SSL/TLS the security context is defined by the socket endpoint connection, which inhibits the establishment of an end-to-end security context when intermediaries are involved.

Session-based communication through intermediaries is common in many Grid specific scenarios, and it is therefore important for OGSA to support that functionality as soon as possible. This capability is to be addressed in the future within the WS Security Architecture by the WS-SecureConversation framework. However, the WS-SecureConversation specification is still work in progress, and thus we suggest that a tactical specification is required that describes how existing Generic Secure Services API (GSSAPI)-accessible [GSSAPI] mechanisms can be used to provide the required session-based security. Once the WS-SecureConversation specification is available, this tactical specification should be reconciled to reflect the standardized protocol.

4.4.1. GSSAPI-SecureConversation Specification

This tactical specification defines how GSSAPI-accessible mechanisms can be used for context establishment and per-message protection for OGSA peer-to-peer communication. The scheme should allow the use of any GSSAPI-accessible mechanism, in particular the Grid Security Infrastructure [GSI] and Kerberos.

The name “GSSAPI-SecureConversation” reflects that this specification is meant to fill temporarily the void that will be filled by the WS-SecureConversation specification.

4.5. Pluggable Authorization Service

The OGSA Security Architecture document describes the need for authorization services that can be called by both requestor and service providers to enforce policy.

Any interaction between two parties requires that authorization decisions be made on both ends. The requester verifies that its policy allows it to invoke the request on the service provider, while the server checks whether its policy allows the request to be serviced. The policy rules on each end can be simple or complicated, but must in general be able to answer the question of whether a subject is allowed to invoke—or service—a particular operation/action on a resource, i.e., the target. Policy decisions can be communicated back by the caller as either Boolean results or authorization/capability assertions.

It is desirable from a deployment and management perspective that an authorization service be *pluggable*: i.e., that it can be integrated into applications via a well-defined interface. Pluggability facilitates deployment of OGSA components in environments where existing policy enforcement tools are in place.

4.5.1. OGSA-Authorization Service Specification

This specification defines an OGSA service that, as discussed above, provides policy evaluation functions for authorization decisions. In defining this specification, we must consider various specifications that are addressing different aspects of this functionality, including SAML, XACML, XrML, and WS-Authorization. [SAML, XACML, XRML, WS-AUTHORIZATION]

4.6. Authorization Policy Management

This subsection and the three that follow (Sections 4.7, 4.8, and 4.9) address security policy management issues.

A number of services will have authorization policy that can be managed by traditional site administrators and, particularly in the case of transient services, by users.

It should be noted that in addition to emerging WS-Policy specifications there is an effort in the larger OGSA community on general policy management. Work on authorization policy management should build on this effort as much as possible and strive to provide feedback on any missing requirements.

This Section describes proposed authorization policy management specifications.

4.6.1. Coarse-grained Authorization Policy Management Specification

This specification defines mechanisms for managing coarse-grain authorization policy (e.g., access control lists: ACLs) imposed by an OGSA service on a requestor. This specification should also address management of policy regarding trust roots: e.g., whom should a service trust to assign identities.

4.6.2. Fine-grained Authorization Policy Management Specifications

A set of further specifications may be defined to support the management of more sophisticated and fine-grained policies. Some possible examples include the following.

- Authorization policy based on required attributes in addition to, or instead of, identities and policy regarding trust roots for attributes.
- Policy regarding the circumstances under which delegation is acceptable.
- Policy based on fine-grained details of an action as described in Section 4.1.2.

4.7. Trust Policy Management

All entities in an OGSA environment will make policy decisions based on the trust they have in the claims and assertions presented by others. In some cases, this trust is *implicit*, as in the case of claims and assertions made by the entity itself. In other cases, trusting entities may be configured to trust other *anchored* entities, as a means of achieving closure. In most cases, however, trust in a statement made by an entity has to be derived through assertions about that entity by others, and this chain has to end with the entity itself or a trust-anchored entity.

The need to trust one entity to make a statement about another introduces policy decisions. These are essentially authorization assertions, and the policy that governs this decision has to defined and managed.

The WS Security Architecture has an emerging WS-Trust [WS-TRUST] module that addresses trust management issues. The distributed and dynamic nature of the Grid environment, and its frequent crossing of administrative boundaries through ad-hoc created virtual organizations, will put high demands on this WS-Trust specification.

4.7.1. OGSA Trust Service Specification

This specification defines an OGSA service that will use the WS-Trust specification to manage and publish trust policies.

4.8. Privacy Policy Management

Requestors that seek to maintain anonymity or to withhold private information will want to inspect a service provider's stated privacy policy and its adherence to that policy. On the other end, service providers may need the ability to adapt their data collection level based on the stated privacy level of the requester. Here we can refer to the general practices and rules defined by the P3P effort [P3P].

The WS-Privacy specification will define a model for how a privacy language may be embedded into WS-Policy descriptions, thus allowing requestors and service providers to exchange their respective privacy policy statements. With the help of WS-Authorization, the agreed privacy policy can then be enforced.

As Grid applications cross the borders of many organizations, countries, and even continents, they may encounter, and have to adhere to, different legislation concerning privacy policy. For OGSA to be successful on a global scale, a tiered privacy policy enforcement model has to be built into the foundation.

4.8.1. Privacy Policy Framework Specification

Building on the emerging WS-Privacy, WS-Policy, and WS-Authorization specifications, this specification defines a framework through which privacy policy can be stated and enforced.

4.9. VO Policy Management

A virtual organization (VO) comprises a collection of users and services that span multiple physical institutions. This broad span can make it difficult to maintain consistent VO information such as membership and policies. We envision a set of services that allow a VO to distribute policy information so as to ensure that policies are consistently enforced.

4.9.1. VO Policy Service Specification

This specification defines one or more OGSA services that act as a repository of VO membership and policy information. These services should support both push (e.g., giving a credential to a user to present to a resource) and pull (a resource requesting information about a user) models for interaction.

Note that since resources that are part of a VO are distributed at a number of different sites, a service cannot assume consistent interpretation of attributes and groups across the VO. This means that policy expressions must be either specific to the resource that is going to evaluate them, or self-descriptive so as to avoid misinterpretation.

4.10. Delegation

It will often be necessary for a requestor to delegate some subset of their rights to a service provider in order for that service provider to fulfill the request. For example:

- A requestor submits a computational job to a compute resource that needs access to data on a data store, when there is no existing trust relationship between the compute resource and the data store.
- A requestor making a request to a travel agent to make a reservation may need to delegate to the travel agent the right to make payment on behalf of the requestor.

Various existing security mechanisms support delegation in various manners. While it is expected that these mechanisms will continue to be used, it is desirable to specify a standard delegation method that can be used uniformly, independently of the underlying authentication mechanisms. Technologies that can perhaps be used to express these different assertions in a standardized way include SAML, XrML, XACML, and WS-Security. For any of these options, specifications are required to define profiles for the use of assertions so as to ensure interoperability among OGSA components.

4.10.1. Identity Assertion Profile Specification

This specification defines a profile for expressing identity assertions that allow an entity to assert the identity name associated with a key, a request, or a communication channel.

WS specifications that can potentially be leveraged are SAML Identity Assertion [SAML] and WS-Security/Policy/Federation [WS-SECURITY, WS-POLICY, WS-FEDERATION]].

4.10.2. Capability Assertion Profile Specification

This specification defines a profile to express capability (e.g., attribute and authorization) assertions.

WS specifications that can potentially be leveraged are SAML Attribute/Authorization Assertion, XACML, XrML, and WS-Security/Authorization. [SAML, XACML, XRML, WS-SECURITY, WS-AUTHORIZATION]

4.11. Firewall "Friendly"

As described in the OGSA Security Architecture, OGSA implementations should be able to interoperate through firewalls, Network Address Translating (NAT) routers and other similar infrastructure.

A scenario of particular difficultly is the case of a client behind a firewall that needs to receive notification events from services. Since users tend to have less control over this infrastructure than the administrators of services do, changes in the firewall policy to aid in this scenario are less likely.

A related scenario, not traditionally thought of as a firewall scenario, is a cluster configuration where access to the front end node is unrestricted, but the compute nodes are on a private network unreachable to the outside world. This scenario is similar since the compute nodes may still need to communicate with entities outside the private network, but are prevented from doing so directly.

4.11.1. OGSA Firewall Interoperability Specification

This specification defines the functionality that OGSA service requestors and providers must support in order to interoperate through intervening firewalls. This specification may also define new OGSA services needed to assist in this effort.

WS specifications that can potentially be leveraged are WS-Routing, WS-Referral and WS-Policy. [WS_ROUTING, WS-REFERRAL, WS-POLICY]

4.12. Security Policy Expression and Exchange

For requestors to be able to interact securely with a service, they have to be able to find the relevant security policy about that service in order to:

- determine whether the requestor's policy allows interaction with that service's identity;
- determine whether the security mechanisms supported by the service are compatible with the those supported by the requestor;
- determine whether the requester's policy and capability can match the service's QoP;
- determine the security information needed to authenticate the service.

There are many different ways the requestor can find this information, and some protocols, like SSL, have certain procedures baked-in.

4.12.1. Grid Service Reference and Service Data Security Policy Decoration Specification

This specification will describe how a requestor determines the information that is required to communicate securely with a service. This specification would define an XML-based format for these polices as well as a scheme for decorating the GSR and Service Data. This decoration of the service's Grid Service Reference (GSR) and Service Data would allow policy to be conveyed to the potential requestors.

The WS-Policy module of the WS Security Architecture promises to address many of our policy expression language requirements. We will work to influence and to make our specification conform to the upcoming WS-Policy standard.

4.13. Secure Service Operation

An incoming request received by a service provider may be subjected to a variety of policy checks as it is passed down through various levels of the service provider implementation—which may include hand off to independent "application" processes. In order to allow different levels to make their own policy decisions, all asserted identities and attributes should be passed through.

The different policy checks applied by the service should be described, and exit points defined, such that a service can interoperate with externally defined services for certificate validation, attribute assertion, authorization policy evaluation, and secure logging. As noted above, the ability to interface to such services can facilitate integration with existing site authentication and authorization infrastructures.

Note that because an OGSA service interfaces comprises both operations and service data, OGSA policy enforcement mechanisms should also control access to service data. One can envision the need for fine-grained access control policy at the service data element level.

4.13.1. Secure Service's Policy and Processing Specification

This specification defines the various policy checks that a service is expected to perform, and defines interfaces to specified external security services.

4.13.2. Service Data Access Control Specification

This specification defines both coarse- and fine-grained access control policy that should be enforced on Service Data accesses.

4.14. Audit and Secure Logging

Any implementation of OGSA will have all the requirements for auditing that are common to any distributed system. All sorts of information concerning authentication and authorization will need to be logged in managed manner.

The complete specification of an Audit Framework is probably outside of the scope of the GGF, but standardizing and audit service and audit management interface will greatly facilitate the overall management of secure logging.

4.14.1. OGSA Audit Service Specification

This specification defines an OGSA Audit Service that allows requestors to submit information for inclusion in secure logs. An associated management interface would control policy on the logging – e.g. filters on what logs are actually stored, possible notification if certain logs are received, etc.

4.14.2. OGSA Audit Policy Management Specification

This specification defines the management interface (porttype, operations and message formats) for normal OGSA services such that they can be externally managed as to what audit entries they generate and how those entries are logged (e.g. to what Audit Service).

5. Table of Proposed Specifications

The following table lists all the specifications proposed in the roadmap, grouped by category.

Category	Specifications
Naming	OGSA Identity OGSA Target/Action Naming OGSA Attribute and Group Naming Transient Service Identity Acquisition
Translation between Security Realms	Identity Mapping Service Generic Name Mapping Policy Mapping Service Credential Mapping Service
Authentication Mechanism Agnostic	OGSA Certificate Validation Service OGSA-Kerberos Services
Pluggable Session Security	GSSAPI-SecureConversation
Pluggable Authorization Service	OGSA-Authorization Service
Authorization Policy Management	Coarse-grained Authorization Policy Management Fine-grained Authorization Policy Management
Trust Policy Management	OGSA Trust Service
Privacy Policy Management	Privacy Policy Framework
VO Policy Management	VO Policy Service
Delegation	Identity Assertion Profile Capability Assertion Profile
Firewall Friendly	OGSA Firewall Interoperability
Security Policy Expression and Exchange	Grid Service Reference and Service Data Security Policy Decoration
Secure Service Operation	Secure Service's Policy and Processing Service Data Access Control
Audit and Secure Logging	OGSA Audit Service OGSA Audit Policy Management

6. Security Considerations

Security related issues are discussed on a high level in this document.

7. Authors Contact Information

Frank Siebenlist
franks@mcs.anl.gov
Mathematics and Computer Science Division, Argonne National Laboratory

236 More Avenue
Los Gatos, CA 95032

Tel: 408-656-6787

Von Welch
welch@mcs.anl.gov
Department of Computer Science, University of Chicago, Chicago, IL 60637

Steve Tuecke
tuecke@mcs.anl.gov
Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL 60439

Ian Foster
foster@mcs.aml.gov
Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL 60439

Nataraj Nagaratnam
natarajn@us.ibm.com
IBM Corporation, Research Triangle Park, NC 27713

Philippe Janson
pj@zurich.ibm.com
IBM Corporation, Zurich Research Lab, Switzerland

John Dayka
dayka@us.ibm.com
IBM Corporation, Poughkeepsie, NY 12601

Anthony Nadalin
drsecure@us.ibm.com
IBM Corporation, Austin, TX

8. Acknowledgements

We are grateful to numerous colleagues for discussions on the topics covered in this paper, in particular (in alphabetical order, with apologies to anybody we've missed): Brian Carpenter, Karl Czajkowski, Doug Engert, Jeffrey Frey, Jarek Gawor, Francis Hildenbrand, Carl Kesselman, Sam Meder, Jeffrey Nick, Laura Pearlman, and Thomas Sandholm.

9. References

[AADS] **Account Authority Digital Signature Model (X9.59)**, http://www.garlic.com/~lynn/

[GGF] **The Global Grid Forum**, www.gridforum.org

[GSI-Proxy-Certs] **S. Tuecke, et. al., Internet X.509 Public Key Infrastructure Proxy Certificate Profile, Global Grid Forum draft**, draft-ggf-gsi-proxy-03, http://www.gridforum.org/2_SEC/GSI.htm

[GSS Spec] **Grid Service Specification**, http://www.gridforum.org/ogsi-wg/

[GSSAPI-Krb5] **The Kerberos Version 5 GSS-API Mechanism**, RFC 1964

[GSSAPI-SPKM] **The Simple Public-Key GSS-API Mechanism (SPKM)**, RFC 2025

[IETF] **The Internet Engineering Task Force**, www.ietf.org

[IETF-GSSAPI] **Generic Security Service Application Program Interface, Version 2, Update 1**, RFC 2743

[Kerberos] **The Kerberos Network Authentication Service (V5)**, RFC 1510

[OASIS-SAML] **Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML)**, http://www.oasis-open.org/committees/security/docs/cs-sstc-core-01.pdf

[OCSP] **Online Certificate Status Protocol**, http://www.ietf.org/html.charters/pkix-charter.html

[OGSA Security Arch] **The Security Architecture for Open Grid Services**

[OGSA] **The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration**, http://www.gridforum.org/ogsi-wg/

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

[PGP] **Pretty Good Privacy**, http://www.ietf.org/html.charters/openpgp-charter.html

[SAML] see [OASIS-SAML]

[SPKI] **Simple Public Key Infrastructure**, http://www.ietf.org/html.charters/spki-charter.html

[SSL-TLS] The TLS Protocol, Version 1.0, RFC 2246

[W3C] **The World Wide Web Consortium**, http://www.w3.org/

[WS-Authorization] see [WS-Security]

[WSDL] **Web Services Description Language (WSDL)**, http://www.w3.org/TR/wsdl

[WS-Federation] see [WS-Security]

[WS-Policy] see [WS-Security]

[WS-Privacy] see [WS-Security]

[WS-Referral] **Web Services Referral Protocol (WS-Referral)**, http://msdn.microsoft.com/library/en-us/dnglobspec/html/ws-referral.asp

[WS-Routing] **Web Services Routing Protocol (WS-Routing)**, http://msdn.microsoft.com/library/en-us/dnglobspec/html/ws-routing.asp

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

[WS-Trust] see [WS-Security]

[X509] **Internet X.509 Public Key**, http://www.ietf.org/html.charters/pkix-charter.html

[XACML] **OASIS eXtensible Access Control Markup Language (XACML)**, draft-xacml-schema-policy-14.doc, http://www.oasis-open.org/committees/xacml/

[XKMS] **XML Key Management Specification (XKMS)**, http://www.w3.org/TR/xkms/

[XML-Encryption] **XML Encryption WG**, http://www.w3.org/Encryption/2001/

[XML-Signature] **XML Signature WG**, http://www.w3.org/Signature/

[XrML] **eXensible rights Markup Language**, http://www.xrml.org/

Intellectual Property Statement

The GGF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the GGF Secretariat.

The GGF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this recommendation. Please address the information to the GGF Executive Director.

Full Copyright Notice