GGF DOCUMENT SUBMISSION CHECKLIST (include as front page of submission)	
	COMPLETED (X) - Date
1. Author name(s), institution(s), and contact information	Sept 22, 2003
2. Date (original and, where applicable, latest revision date)	Sept 22, 2003
3. Title, table of contents, clearly numbered sections	Sept 22, 2003
4. Security Considerations section	Sept 22, 2003
5. GGF Copyright statement inserted (See below)	Sept 22, 2003
6. GGF Intellectual Property statement inserted. (See below) **NOTE that authors should read the statement.**	Sept 22, 2003
7. Document format - The GGF document format to be used for both GWD's and GFD's is available in MSWord, RTF, and PDF formats. (note that font type is not part of the requirement, however authors should avoid font sizes smaller than 10pt).	Sept 22, 2003

Mary Thompson, LBNL
Von Welch, University of Chicago
Marcus Lorch, Virginia Tech
Rebekah Lepro, NASA,Ames
David Chadwick, University of Salford

OGSA Attributes: Requirements, Definitions, and SAML Profile

Status of This Memo

This document has been submitted to the Global Grid Forum OGSA OGSA-Authz Working Group for consideration as recommendations document in that area of OGSA authorization.

The latest version of this document can be found at:

https://forge.gridforum.org/projects/ogsa-authz/document/draft-OGSA-attributes-v6/en/1/draft-OGSA-attriubtes-v6.doc

Abstract

This document specifies elements and vocabulary for expressing attribute assertions to be used in the context of the Open Grid Services Architecture (OGSA). A profile for specifying subject attributes using SAML AttributeAssertions is also included. The intention of defining standard formats and meanings for these assertions is to facilitate compatibility between issuers of attribute assertions and the authorization systems that consume them.

Contents

1. Introduction

This document is a companion to the “OGSA Authorization Requirements” GWD [OGSA-authz-req] and assumes that the reader is familiar with that paper. Many terms used in this document are defined in a common glossary that is included at the end of this document.

Most authorization systems that make decisions based on access control policy consider attributes of an initiator in addition to identity. Basing all access control on the initiator identity alone requires an extremely verbose and inflexible policy that does not scale well as more principals are added to the policy.

The intention of this document is to allow for interoperability between attribute authorities (AA) which issue attribute assertions, the policy writers who define access policy, and access decision functions (ADFs) that make decisions based on the initiator’s attributes and resource policy. In a typical Grid environment there may be several authorities that assert attributes for users. Various domains will want to write authorization policy based on such attributes. Standard methods for discovering, guaranteeing integrity and transporting these assertions as well as common formats and vocabularies for expressing their assertion semantics are needed to enable the various pieces of a Grid to interact.

A number of methods for requesting and encoding attributes already exist (e.g., X.509 Attribute Certificates [RFC3281], SAML Attribute Assertions [SAML] and XACML Attributes [XACML]. This document does not intend to define a new method or dictate the use of an existing method. Instead, it documents the functionalities needed to support OGSA Authorization and defines a profile for encoding these functions using SAML Attribute Assertions. It is expected that other profiles will be defined for the use of other mechanisms in OGSA.

Section 2 defines the conventions and namespaces used in this document. Section 3 presents an overview of the requirements for the use and content of subject attributes in the OGSA authorization context. Section 4 provides a non-normative discussion of current attribute mechanisms. Section 5 contains a normative set of definitions for attributes to be used in OGSA authorization. Section 6 contains a normative profile for expressing OGSA attribute assertions using SAML.

2. Conventions used in this Specification

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC-2119 [RFC2119].

The following namespace prefixes may be used in XML examples in this document. Note that the choice of any namespace prefix is arbitrary and not semantically significant.

Table 1: Name spaces used in this specification.

Prefix	Namespace
ogsa-saml	http://www.gridforum.org/namespaces/2003/06/ogsa-authz/saml/
operation	http://www.gridforum.org/namespaces/2003/06/ogsa-authz/saml/action/operation
sde-read	http://www.gridforum.org/namespaces/2003/06/ogsa-authz/saml/action/sde/read
sde-modify	http://www.gridforum.org/namespaces/2003/06/ogsa-authz/saml/action/sde/modify
wildcard	http://www.gridforum.org/namespaces/2003/06/ogsa-authz/saml/action/wildcard
saml	urn:oasis:names:tc:SAML:1.0:assertion

samlp	urn:oasis:names:tc:SAML:1.0:protocol

3. OGSA use and requirements for Subject Attributes.

Attributes provide information about entities that can be used in addition to [or in lieu of] the entity's identity to make authorization decisions. Attributes can be associated with the initiator of an action, often referred to as the subject in an authorization context, with the resources, with the actions, and with the environment. This document will focus on subject attributes, i.e. those associated with the initiator. However, it will keep in mind the other types of attributes when deciding on the components of an attribute, since it is desirable to have a common format to represent arbitrary attributes.

Attributes normally make a positive statement, i.e. the holder has the attribute with the following value(s). In theory, one could make a negative statement, i.e., the holder does not have the attribute with the following value(s). Instead, the absence of a positive attribute statement is used. Policy statements could use attributes in either an additive manner, e.g. if the user has the attribute, he has the following rights, or negative, if the user has the attribute, he is denied some rights. The negative case is insecure in a push system, where the initiator may fail to push an attribute that would deny rights. Even in a pull system, the attribute repositories must be in a closed domain, so that no attributes would be missed in a search. Combining policy statements that make both positive and negative assertions about rights is much more complicated than a simple additive scheme.

The OGSA Authorization Requirements document [OGSA-Authz-Req] identifies several authorization scenarios that require attribute information be passed between two parties. The first group of scenarios are variants of the push model, in which the initiator retrieves its credentials from a trusted third party, such as a VO manager, and passes them to the Grid service controlling access to some resource. Other scenarios are variants of the pull model, in which the initiator passes the Grid service a reference element from which the authorization service (ADF) retrieves the necessary credentials. In order to satisfy the trust relationships between various Grid sites, the assertion must contain sufficient information such that the relying party can determine who made the assertion and that the content was not corrupted in transit. A clean way to solve both of these requirements is to use digitally signed *attribute assertions* that associate an issuer, a holder (also referred to as the subject), validity dates and possibly other conditions, with an attribute. If the two parties communicate over an unsecured channel, the issuer must digitally sign each shared assertion. If shared via a secure and authenticated connection, the assertions may be unsigned for efficiency.

An attribute assertion may optionally include some constraints that the issuer wishes to impose on the attribute. These conditions should be simple since they will be combined with any conditions included in the applicable authorization policy. However, they are the only way for the issuer of the attribute to limit its use and validity. Some uses of this feature are to restrict the caching of an attribute, to limit its use to less than a certain level of delegation and to have it take effect only during certain hours of a day.

Attributes are most commonly used in a additive manner.

One of the more obvious requirements of attribute assertions for Grids is the need for extensibility in defining attribute names, values and conditions. On the other hand, in order to allow for the interoperability of different Grid services which enforce authorization (AEF), authorization services (ADF), the attribute issuers, and the policy writers, we need to specify a basic set of elements for attribute assertions and identifiers and values for attributes.

The following section will examine some of the current attribute standards in order to see what is applicable for OGSA.

4. Existing Attribute Standards

4.1 X.509 Attribute Certificate

The IETF PKIX working group defined an X.509 Attribute Certificate that binds attributes to a holder and is digitally signed by an attribute authority. This certificate definition was motivated by the desire to keep attributes out of X.509 public key certificates and encourage the separation of identity and privileges.

The requirements of these certificates include: [RFC3281]

Note: In the context of the X.509 specification an attribute type is identified by its object identifier [OID] which explicitly refers to a schema definition which defines the everything about the attribute, including the name, number and data types of the values and for enumerated types the actual values.

- Issuers of ACs should be able to define their own attribute types for use within closed domains.
- Some standard attribute types, which can be contained within ACs, should be defined. Examples include "access identity," "group," "role," "clearance," "audit identity," and "charging identity."
- Standard attribute types should be defined in a manner that permits an AC verifier to distinguish between uses of the same attribute in different domains. For example, the "Administrators group" as defined by Baltimore and the "Administrators group" as defined by SPYRUS should be easily distinguished.
- It should be possible to "target" an AC at one, or a small number of, servers. This means that a trustworthy non-target server will reject the AC for authorization decisions.

A X.509 Attribute certificate typically has a single subject (called the holder), a number of attributes of possibly varying types, each having its own schema that defines the number and data types of its values. Multiple holders are possible but not widely used. Attribute certificates are ASN.1 encoded, have one validity period and specify the issuer who signed it. They also allow optional extensions that can be used to constrain certificate validity.
The attribute types that have been defined are: id-aca-authenticationInfo, id-aca-accessIdentity, id-aca-chargingIdentity, id-aca-group, id-at-role, id-at-clearance. Extensions contain information about the attribute and how to verify it, e.g. revocation locations, keyInfo and audit Identity.

4.2 SAML Attribute Assertions

SAML (Security Assertion Markup Language) defines an XML-based protocol for querying and expressing authentication, attribute and authorization assertions about principals. Attribute assertions for a particular subject may be requested via an AttributeQuery wrapped within a SAML request. According to protocol semantics, a SAML response to that request contains zero or more relevant assertions.

SAML also defines an assertion language such that assertions may exist independently to this protocol. Each SAML assertion is a generic packaging of a set of statements pertaining to a particular category (Attribute, AuthorizationDecision or Authentication) into a standard XML structure. Each assertion holds meta-data specific to the assertion itself, such as the issuer identity represented by a string, assertion identifier, and protocol version numbers as well as conditions and advice. Assertion validity dates are a specific form of a condition. Other standard condition definitions address caching and intended audience restrictions.

As the assertion is the packaging of asserted data, SAML specifies that digital signatures be attached at this level. However, a single SAML assertion can wrap multiple attribute statements. Each attribute statement contains a single subject identity, and one or more attributes, each with zero or more values. Attributes are identified within a statement by an AttributeDesignator. An AttributeDesignator specifies a namespace uri and an attribute name local to that namespace.

4.3 XACML Attributes

XACML (extensible Access Control Markup Language) is designed to express access control policy and the context carried with an initiator when requesting an authorization. Both the policy and the request context use attributes. In addition to subject attributes, XACML defines a standard representation for environment, action, and resource attributes. Within an access control policy, an XACML attribute is conceptually specified by a combination of the unique attribute identifier in uri form, a data type and the attribute issuer, and an indicator for its required presence in any context to be evaluated against this policy. This data is defined as an XML complex type named AttributeDesignatorType. The Attribute element is the central abstraction of a request context that will be evaluated against and XACML access control policy. This element comprises meta-data and an attribute value [XACML]. This meta-data contains the attribute identifier, data-type and issuer so that the PDP may identify any matches with an attribute designator in a policy.

Attributes may be associated with a specific subject in a request context. Further, each subject within a request context may be categorized by the presence of attribute represented by a SubjectAttributeDesignatorType derived from the basic AttributeDesignatorType. XACML defines a number of attribute identifiers for use within a Subject Attribute Designator. They have uris of the form urn:oasis:names:tc:xacml:1.0:subject: <id> and include subject-id, subject-category, subject-id-qualifier, key-info, authentication-time, authentication-method, request-time, start-time, ip-address, dns-name. XACML also defines a naming convention to use any identifiers defined in LDAP, e.g., http://www.ietf.org/rfc/rfc22565.txt#userPassword.

XACML does not use namespaces for attribute identifiers, does not attach conditions to them, and does not have a specification for signed assertions with validity dates. Instead, the context in which the attribute is embedded may be secured by some means outside of the scope of XACML.

From Anne Anderson - It is hard to link the attribute namespace (namequalifier) and the attribute name when making policy statements without having a complex format for referencing attributes. Scott Cantor has experience with this problem in using SAML attributes in Shibboleth. It is why XACML chose to have a one-part name.

If you want to distinguish "permisRole" from "BarcelonaRole" then define two different URIs for them. If you want to know they have the same format, then associate the same DataType with both of them.

4.4 Shibboleth

Shibboleth [SHIB], the Internet2 architecture for sharing web resources with access control, defines attributes about its users to the sites. They have extended the XACML naming scheme to include an LDAP schema for eduPerson [EP] that builds on the inetOrgPerson [LDAP]. They specify the names of attributes to be the attributes defined in eduPerson schema, e.g., eduPersonPrincipalName, eduPersonAffliation, eduPersonExtGroupMembership,

4.5 Commonalties and Differences

Below are a number of ways that attribute assertions can be modeled that seem to have similarities between the three formats discussed above.

- Number of subjects supported
- Representing multiple values
- Predefined attribute identifiers
- Digital signatures

Below are a number of ways that attribute assertions can be modeled that seem to have differences between the three formats discussed above.

- Attribute identifier format

- Attribute meta-data
- Encoding
- Association with a subject or principal

There is typically a single subject who is the holder of one or more attribute(s) (attribute certificates can accommodate multiple holders but its not recommended). A named attribute may have one or more values associated with it. X.509 and SAML can associate conditions with the attribute. Assertion signing is mandatory for X.509 certificates, optional for SAML AttributeAssertions, and not defined in XACML. All three systems allow an attribute to have multiple values. SAML and X.509 allows grouping of several attributes per subject.

5. Standard OGSA Attributes

5.1 Standard Attribute Elements

This section contains a normative specification for the abstract attribute elements.

In order to store attributes in non-secure repositories and to transmit them across unsecured connections, optionally signed attribute assertions are required. The attribute element should be useable in policy statements and should be able to hold environment, action and resource attributes as well as subject attributes.

These assertions MUST contain the identity of the issuer, the holder of the attribute(s) and one or more attributes. They SHOULD have begin and end validity dates and MAY have additional conditions. The attributes MAY have conditions and MAY be typed. The attributes MAY be named in a flat name space or MAY have a namespace component. The name and value elements MUST be extensible.

```
Attribute Assertion
  Issuer
  Condition (0 or more)
  Holder/Subject
  Attribute (1 or more)
     Name
    Value (0 or more)
    Data Type (0 or 1)
    Condition (0 or more)
  Signature (optional)
```

MRT Should validity be separate from other conditions. Can validity be optional? Do we need to have different validity periods for each attribute. If so can they just be placed in different assertions.

Do we need to specify data types for each attribute?

[RSL – What about standardizing attribute meta-data definitions, such as issuer identity, attribute value datatype, etc? Are we making assumptions about attribute type here? Is the attribute type an actual datatype or is it tuple of [attribute identifier, attribute datatype)? Are we assuming that the attribute name is the attribute identifier? What about a naming scheme for such identifiers OID, Namespace and Name, AttributeId are three different approaches taken in AC, SAML and XACML which are each noted.]

Anne Anderson - Argument in favor of including a data type. An XACML PDP must know not only the syntax of an attribute value, but also the semantics for how to handle it in functions (compare it for greater or less than, add it to another value, etc.). If attribute values were defined as schema instances, then not only would the PDP have to locate and process the schema associated with each attribute, but the PDP would also have to be augmented with code that understands the semantics of the schema-defined information.

5.2 Standard Attribute Types

This section contains a normative specification of attribute names and meanings.

The definitions of attribute identifiers and data types MUST be understood by attribute authorities and policy writers. They MAY need to be understood by an initiator in order to gather up the required attributes before contacting a Grid service. Ideally, attributes can be opaque to the Grid services, authorization services and any attribute repositories. For many attributes such as group, role and affiliation the ADF can verify that an initiator has the required attribute by doing a case sensitive string comparison between a required attribute and the ones that a user presents. On the other hand supporting wild-cards in attributes, or non-string values, requires the ADF to understand the data types. Also, the data type of environment or initiator context attributes such as IP address or disk quotas, MUST be understood and evaluated by either the AEF or ADF.

Some generally useful attribute types are “group”, “role”, “account id”, aka. “charging identiy”, “project id”, “clearance”, “citizenship”, and “VO membership”. With the exception of citizenship and possibly clearance, none of these has a standard set of values. It is anticipated that these names would be defined in a /www.gridforum.org/namespaces/2003/06/ogsa-authz/attributeType namespace. It is anticipated to follow the Shibboleth and XACML examples and use and extend the attributes of eduPerson and inetOrgPerson. It would also help interoperability to accept the XACML request context attributes.

[RSL – need to standardize both attribute names and legal values]

[RSL – What about Liberty attributes?]

5.3 Standard conditions

This section contains a normative specification of attribute conditions and their meanings.

An attribute authority constrains the use of subject attribute via conditions. Conditions SHOULD be kept simple because if a relying party does not understand how to process the condition, it MUST not use the attribute. We need to support single value conditions, like DoNotCache, conditions that are equal to one or more values, e.g. audienceRestriction and conditions that are expressed by algebraic expressions combining terms and values that are known by the policy writers, the authorization service and the Grid service (AEF). Unlike the conditions placed on authorization decisions, most attribute conditions can be evaluated by an authorization service, since they tend to be more resource independent. Some generally useful conditions on attributes are:

- Time of day, e.g. time >= 8:00 & time <= 17:00
- Days of week, e.g. day != sat & day != sun
- Making one attribute depend on the existence of another e.g role=administrator if project=Atlas

The standard vocabulary for such expressions includes the relational operators: =, !=, <, >, <=, >=, &, |, times of day hh:mm with a 24 hr clock, days of the week: sun-sat.. XACML functions provide a possible vocabulary and format for relational expressions. They are defined in the XACML v1.1 document, starting on page 100. E.G. String-equal, integer-equal, boolean-equal, date-equal, time-equal, x500name-equal, string-greater-than, string-greater-than-or equal, etc. .

[RSL – Are we assuming that conditions are expressed as functions?]

ML: As mentioned in an email if we want to I would recommend to reuse XACML functions here. However, using XACML may be quite verbose, so if we want to be human readable we may want to use the relational operators mentioned above, but I think there may be encoding issues. (<,>,& have to be escaped in XML- mrt) XACML conditions are evaluated by ADF, while the attributes may be extracted from the request context and assembled for submission to the PDP by the PEP who typically would not have the code for evaluating XACML rules.

[RSL - I think that these are good discussion points. i.e. What motivates the need for human readability?] I find it hard to determine a useful subset of the standard XACML functions (as all seem useful without a concrete use case) and would almost think that if one uses an XACML library all the functions are available to the PDP. However, a context manager (which has typically no XACML evaluation engine) is the component that would have to evaluate these conditions and determine if the attribute should be honored (provided to the PDP) or not.

6. SAML profile for attribute assertions

This section contains a normative specification of how the attributes and conditions defined in the previous sections should be expressed using SAML. This document does not require the use of SAML for expressing Attribute Assertions in OGSA, but only defines how it MUST be used if chosen by the implementer.

The SAML Assertion element is used by one entity to assert the statements about a principal. While an Assertion element can contain a variety of SAML statements, for the purposes of this document we consider only AttributeStatements. The Assertion element includes the following elements:

- An optional *Conditions* element specifying the conditions for use of the assertion.
- An optional *Advice* element specifying advice for use of the element.
- Zero or more *AttributeStatements* specifying attributes.
- An optional *Signature* element allowing the Assertion to be verified.

It also carries the following attributes:

- The issuer (the attribute authority)
- The issue instant (date/time)

The following subsections describe the use and extensions to these elements for OGSA.

6.1 Conditions Element

Implementations are advised to be conservative in their use of this element and only include it when they are confident it will be understood. Relying parties MUST not use an attribute if they do not understand how to evaluate any of its conditions. Implementations MAY support only standard conditions.

The Conditions element can contain optional time constraints and/or zero or more Condition elements (note difference in plurality between element names) on the assertion. Several basic condition types, such as cache behavior or audience restrictions, are directly defined in the specification [SAML] as well as an abstract condition element that serves as an extension point. These extended conditions should be used to express particular constraints that the attribute authority wishes to place on the use of the attribute by the subject. One of the most obvious uses for this is to limit the time of day that a subject can act in a specific role.

MRT We need to define the condition schema for conditions consisting of algebraic expressions or choose to use XACML functions.

6.2 Advice Element

This specification recommends against the use of the Advice element. Implementations SHOULD NOT use this element and MAY only include it when they are confident that it will be understood.

6.3 AttributeStatement Element

The Attribute Statement contains the following elements:

- Subject element
- One or more attributes consisting of

- Attribute name and name space
- One or more attribute values

[RSL – How can we allow attribute meta-data to be specified? How should namespaces and names be concatenated to derive an attribute identifier?]

When the assertion encapsulating the Attribute Statement is passed across an insecure network, it MUST be signed by the attribute authority.

6.3.1 Subject Element

This element contains the name of the attribute holder. The Subject and contained NameIdentifer elements are unchanged from the SAML specification. The exact use of these elements is driven by the authentication mechanism used by the client. In some scenarios, the authorization service (ADF) MAY require the holder and client names to be the same. In other scenarios, the authorization service MAY allow trusted clients to request authorization decisions on behalf of any initiator.

The SAML specification defines how some common identity types are asserted. The Grid Security Infrastructure (GSI) is a common Grid authentication mechanism that uses X.509 based identities. The SAML specification defines a URI for X.509 subject names (#X509SubjectName) that SHOULD be used for GSI authenticated identities. Note that SAML specifies the LDAP encoding of DNs [RFC2253].

[RSL – do we need to say anything else about the format of data within the field?]

6.4 Signature Element

This specification places no constrains on the Signature elements. Implementations MUST sign assertions when they do not have an authenticated and secure connection to the evaluator of the assertion.

Security Considerations

This specification defines the elements and use of attributes for authorization services. Implementers of attributes need to be aware that errors in implementation could lead to denial of service or improper granting of service to unauthorized users. Users of attribute assertions should be aware of the situations in which they must require and verify signed assertions.

Author Information

Mary R. Thompson
Lawrence Berkeley National Laboratory
MRThompson@lbl.gov

Von Welch
University of Chicago
welch@mcs.anl.gov

Markus Lorch
Department of Computer Science
Virginia Tech
mlorch@vt.edu

Rebekah Lepro
NASA, Ames
bekah@nas.nasa.gov

David Chadwick
Information Systems Institute

University of Salford
d.w.Chadwick@salford.ac.uk

Glossary

The following terms are abbreviations are used in this document.

AA – Attribute Authority Principal that is trusted to issue attribute assertions.

ACI – Access Control Information (from ISO 10181-3). Any information used for access control purposes, including contextual information.

ADF – Access control Decision Function (from ISO 10181-3). A specialized function that makes access control decisions by applying access control policy rules to an access request, ADI (of initiators, targets, access requests, or that retained from prior decisions), and the context in which the access request is made.

ADI – Access control Decision Information (from ISO 10181-3). The portion (possibly all) of the ACI made available to the ADF in making a particular access control decision.

AEF – Access control Enforcement Function (from ISO 10181-3). A specialized function that is part of the access path between an initiator and a target on each access request and enforces the decision made by the ADF.

Client – the entity making a decision request to the ADF (it could be the target, the initiator, or a proxy acting on behalf of the initiator)

Contextual information – Information about or derived from the context in which an access request is made (e.g. time of day).

Environmental parameters – same as contextual information.

Initiator – An entity (e.g. human user or computer-based entity) that attempts to access other entities (from ISO 10181-3).

OID - Object Identifier, a strings of numbers allocated in a hierarchical manner, so that, for instance, the authority for "1.2.3" is the only one that can say what "1.2.3.4" means. The formal definition of OIDs comes from ITU-T recommendation X.208 (ASN.1). OIDs are assigned by the Internet Assigned Numbers Authority (IANA)

PDP – Policy Decision Point (from RFC2904), same as ADF

PEP – Policy Enforcement Point, (from RFC2904) same as AEF

Privilege – An attribute or property assigned to an entity by an authority

Relying party - The entity that uses information such as attribute assertions, or authorization assertions to allow some actions.

Subject - same as initiator (used by SAML and XACML)

Target – An entity, usually a resource, to which access may be attempted (from ISO 10181-3).

Intellectual Property Statement

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

References

[Akenti] Thompson, M., et al., "Certificate-based Access Control for Widely Distributed Resources," in Proc. 8th Usenix Security Symposium. 1999.

[Authz] Welch, V., et al, OGSA Authorization Requirments, June, 2003.

[CAS] Pearlman, L., V. Welch, I. Foster, C. Kesselman, S. Tuecke, "A Community Authorization Service for Group Collaboration," Proceedings of the IEEE 3rd International Workshop on Policies for Distributed Systems and Networks, 2002.

[EP] “Class Object Specification 200210”, http://www.educause.edu/eduperson/

[LDAP] Smith, M., “Definition of the inetOrgPerson”, RFC2787, April 2000

[OGSA-Authz-Req] Welch, V., Siebenlist, F., Chadwick, D., Meder, S., Pearlman, L. “OGSA Requirements” June 2003 https://forge.gridforum.org/docman2/ViewCategory.php?group_id=119&category_id=449

[OGSA-Service] Welch, V., Siebenlist, F., Chadwick, D., Meder, S., Pearlman, L. “Use of SAML for OGSA Authorization”, Sept 2003 https://forge.gridforum.org/docman2/ViewCategory.php?group_id=119&category_id=450

[OGSI] Foster, I., C. Kesselman, J. Nick, S. Tuecke, "The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration," Open Grid Service Infrastructure WG, Global Grid Forum, June 22, 2002.

[PERMIS] Chadwick, D.W., Otenko, O.," The PERMIS X.509 Role Based Privilege Management Infrastructure", Proceedings of 7th ACM Symoisium on Access Control Models and Technologies (SACMAT 2002).

[Roadmap] Siebenlist, F., et al, "OGSA Security Roadmap," OGSA Security WG, Global Grid Forum, July, 2002.

[RFC2904] Vollbrecht, J., et al, " AAA Authorization Framework", RFC 2904, August 2000.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels," BCP 14, RFC 2119, March 1997.

[RFC2253] Wahl, M., Kille, S., Howes, T., " Lightweight Directory Access Protocol (v3): UTF-8 String Representation of Distinguished Names", Dec. 1977

[RFC3281] Farrell, S., Housley, R. "An Internet Attribute Certificate Profile for Authorization", RFC 3281, May 2002.

[SAML] "Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML) v1.1, July 13, 2003, OASIS Security Services Technical Committee, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security, June, 2003.

[SHIB] Erdos, M., Cantor, S., "Shibboleth-Architecture DRAFT v0.5, http://shibboleth.internet2.edu/docs/draft-internet2-shibboleth-architecutre-05.pdf

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

[XACML] "OASIS eXtensible Access Control Markup Lanugage (XACML) Committee specification 1.0", Dec. 2002,OASIS Security Services Technical Committee, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml, Feb, 2003.