GGF DOCUMENT SUBMISSION CHECKLIST (include as front page of submission)	
	COMPLETED (X) - Date
1. Author name(s), institution(s), and contact information	X
2. Date (original and, where applicable, latest revision date)	X
3. Title, table of contents, clearly numbered sections	X
4. Security Considerations section	X
5. GGF Copyright statement inserted (See below)	X
6. GGF Intellectual Property statement inserted. (See below) **NOTE that authors should read the statement.**	X
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).	X

This page intentionally left blank to align document for 2-sided printing.

GWD-R (proposed)

Von Welch, NCSA
Rachana Ananthakrishnan, Argonne National Laboratory
Frank Siebenlist, Argonne National Laboratory
David Chadwick, University of Salford
Sam Meder, University of Chicago
Laura Pearlman, Information Sciences Institute
~~May~~December, 2004

Use of SAML for OGSA Authorization

Status of This Memo

This document has been submitted to the Global Grid Forum OGSA Security 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

Abstract

This document defines an open grid services architecture (OGSA) authorization service based on the use of the security assertion markup language (SAML) as a format for requesting and expressing authorization assertions. Defining standard formats for these messages allows for pluggability of different authorization systems using SAML.

Table of Contents

1 Introduction

This specification defines the use of Security Assertion Markup Language (SAML) [SAML] for requesting and expressing authorization assertions and decisions from an OGSA authorization service and allow for the communication of authorization decisions from such a service to a service fielding a request from a client. This specification is written to meet the requirements for OGSA Authorization stated in [OGSAAuthzReq].

The SAML AuthorizationDecisionQuery element is defined as the message to request an authorization assertion or decision and an ExtendedAuthorizationDecisionQuery message is specified to allow for more expression of desired parameters of the response. A SimpleAuthorizationDecisionStatement is specified to allow an easy to parse response to a request as opposed to an enumeration of rights.

Section 2 describes the conventions and namespaces used in this document. Section 3 discusses the relationship of this document to the ongoing work in the OASIS standards body, Section 4 contains a non-normative description of SAML extensions defined in this document and Section 5 is a normative definition of those extensions. Section 6 is normative and defines how SAML elements should be used to form OGSA authorization assertions and requests. Section 7 contains the minimal WSDL for the authorization service portType. The document concludes with Acknowledgements, GGF copyright and intellectual property statements, author affiliation and contact information, references and a glossary.

0 contains a non-normative description the portions of SAML that pertain to its use in this document. Appendix B discusses known intellectual property claims on SAML

2 Conventions use 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].

It is assumed that the reader is familiar with the SAML [SAML], Open Grid Services Infrastructure [OGSI] and Open Grid Service Architecture [OGSA] documents. This document uses terminology as defined in the Authorization Glossary as produced by the GGF Working Group on Authorization Frameworks and Mechanisms [Authz-Glossary].

This specification uses namespace prefixes throughout. These prefixes are listed in Table 1~~Table 1~~. Note that the choice of any namespace prefix is arbitrary and not semantically significant.

Table 1: Namespace prefixes used in this specification

Prefix	Namespace
Saml	urn:oasis:names:tc:SAML:1.0:assertion
Samlp	urn:oasis:names:tc:SAML:1.0:protocol
ogsa-saml	http://www.gridforum.org/namespaces/2003/06/ogsa-authz/saml/
Sd	http://www.gridforum.org/namespaces/2003/03/serviceData

3 Relationship to Ongoing SAML Activities in OASIS

This section is a **non-normative** discussion of the relationship of this document to the standards activities ongoing in the OASIS standards body with regards to SAML.

At the time of this writing, SAML 1.1 is the latest version of the SAML specification released by the OASIS Security Services Technical Committee [OASIS-SSTC] and in is upon this version of SAML that this document is based. It is also unclear at this time what the future of the authorization functionality of SAML will be with the upcoming 2.0 release of SAML. It is very likely that there will be substantial revision, possibly with a deprecation of the current SAML

authorization functionality which would be subsumed by the eXtensible Access Control Markup Language Technical Committee [OASIS-XACML] in version 2.0 of their specification.

However, the OGSA Authorization working group decided to press ahead with the use of the 1.1 version of SAML despite its uncertain future. The primary reason for this was the urgent need in the Grid community for a solution. Instead of waiting for a more stable solution to emerge, it was decided that we should proceed with a solution at the present time. This would both provide a standard for current implementers and allow real world experience to be gained which we could use to help with developments of a future standard either in GGF or OASIS.

It should also be noted that SAML was chosen to due the availability of an open source implementation [OpenSAML].

4 Overview of Extensions

This section provides **non-normative** discussion of the extensions in Section 5 of this specification.

The goals of these extensions are to allow an entity requesting an authorization decision to indicate the following desires in regards to the response and for the responder to oblige those requests if it can and desires:

- To request a simple decision in regards to that query instead of a list of allowed rights of the subject.
- To request either the assertion(s) or response be signed.
- To provide one or more URIs for services from which attributes regarding the subject may be obtained.

4.1 Extended Authorization Query

This document defines an extended authorization query which adds the following features to the standard SAML Authorization query:

- A mechanism to allow a requestor to indicate their interest in a simple authorization response rather than a full set of AuthorizationDecisionStatements. The intent is to allow a PEP to request an easily parsed decision regarding any number of requested actions. The response allows the PEP to know easily if all actions where allowed or any where denied without having to parse a list of statements.
- A abstract mechanism, AuthorizationAdvice, to allow a requestor to pass information to the PDP which it may choose to use in making in decision. This document also defines once such element, SubjectAttributeReferenceAdvice, which allows a requestor to pass a pointer to the source of attribute information regarding the subject.
- A mechanism to allow a requestor to indicate their preference in regards to whether the response is signed and how. This is useful for saving work on the PDP in situations where some clients may be passing the response on to another party (e.g. in a push mode of operation) while others will be direct consumers and hence don't need any signatures when the transport layer provides sufficient security.

4.2 Simple Authorization Decision Statement

In the SAML authorization query protocol, a resource normally sends a query to the decision service with an enumeration of the actions being attempted by a requestor. The decision service responds with an assertion containing the set of actions that the requestor is authorized to perform.

While this functions well for situations where the resource may be interested in knowing what subset of the actions the requestor is allowed to perform, in "all or nothing" situations where the resource is only interested in knowing if the requestor can perform all the enumerated actions, it

requires the resource to process the entire list to verify if all the actions originally requested are listed.

This specification defines a new StatementType, the SimpleAuthorizationDecisionStatement element, which contains a reference to the original ExtendedAuthorizationDecisionQuery and a simple boolean decision in regards to that query as a whole. This allows an easy-to-parse decision to be rendered on the query as a whole, as well as potentially significantly reducing the bandwidth needed to transmit the decision.

5 SAML Extensions

This section is **normative**. It defines extensions to the SAML extensions for use in OGSA authorization. See the previous section for a non-normative description of these extensions.

These extensions are made to the SAML 1.1 schema using the type derivation method as described in Section 6.3 of [SAML].

5.1 Element <ExtendedAuthorizationDecisionQuery>

The ExtendedAuthorizationDecisionQuery element allows the entity making the query to indicate its preferences in regards to the query response. This element extends the SAML AuthorizationDecisionQuery element.

An ExtendedAuthorizationDecisionQuery element contains the following additional attributes:

RequestSimpleDecision [Optional]

> This elements indicates the requestor's preference in regards to having the response in the form of a single SimpleAuthorizationDecisionStatement (as defined in this document) instead of as one or more SAML AuthorizationDecisionStatment elements.

Recipient [Optional]

> ~~This element is used to indicate the intended recipient of the response. When a SimpleAuthorizationDecisionStatement is requested, the recipient element will be included in that statement to help prevent replay of the element to entities other than the recipient.~~This element is depreciated and its use should be avoided.

RequestSigned [Optional]

> This element is used to request that a signature be included with the response. This element should contain the QName of the element to be signed - i.e. samlp:Response or saml:Assertion. A responder to a query with this attribute set SHOULD sign the response as request, however is under no obligation to and MAY return an unsigned response (or one signed in a different manner than requested).

An ExtendedAuthorizationDecisionQuery element contains the following additional elements:

AuthorizationAdvice [Optional]

> This abstract element allows for additional information to be included with the query that the responder MAY use when rendering a decision. This element is defined in Section 5.1.1 of this document.

The following schema fragment defines the <ExtendedAuthorizationDecisionQuery> element and its ExtendedAuthorizationDecisionQueryType complex type:

```
<element name="ExtendedAuthorizationDecisionQuery"
type="ExtendedAuthorizationDecisionQueryType"/>
<complexType name=" ExtendedAuthorizationDecisionQueryType">
    <complexContent>
        <extension base="samlp:AuthorizationDecisionQuery">

            <attribute name="RequestSimpleDecision" type="boolean" use="optional"
              default=”false”/>
```

```
            <attribute name=”Recipient” type=”anyURI” use=”optional”/>
            <attribute name=”RequestSigned” type=”QName” use=”optional”/>
            <sequence>
               <element ref=”ogsa-saml:AuthorizationAdvice” minOccurs=”0"
                  maxOccurs="unbounded” />
            </sequence>
        </extension>
    </complexContent>
</complexType>
```

5.1.1 Element < AuthorizationAdvice>

The <AuthorizationAdvice> element is an extension point that allows for additional information to be included with an authorization query that MAY be used by the responder.

The following scheme fragment defines the <AuthorizationAdvice> element and its AuthorizationAdviceAbstractType complex type:

```
<element name=”AuthorizationAdvice” type=”ogsa-saml:AuthorizationAdviceAbstractType”/>
<complexType name=”AuthorizationAdviceAbstractType” abstract=”true”/>
```

5.1.2 Element <SubjectAttributeReferenceAdvice>

The <SubjectAttributeReferenceAdvice> element supplies a statement that the designated attributes associated with the specified subject may be obtained from the referenced URI. Its purpose is to advise the PDP as to where it may find attributes of the subject when working in the *credential pull mode* of operation.

<SubjectAttributeReferenceAdvice> is of type SubjectAttributeReferenceAdviceType, which extends the AuthorizationAdvice AbstractType with the addition of the following:

AttributeDesignator [Any number]

> These elements list the attributes that may be located at the referenced URI. If this component is absent, then it implies that all attributes can be found at the referenced URI.

Reference Attribute [Required]

> This attribute provides the URI from which the attributes may be obtained.

The following schema fragment defines the <SubjectAttributeReferenceAdvice> element and its SubjectAttributeReferenceAdviceType complex type:

```
<element name="SubjectAttributeReferenceAdvice"
   type="ogsa-saml: SubjectAttributeReferenceAdviceType"/>
<complexType name="SubjectAttributeReferenceAdviceType">
     <complexContent>
        <extension base="AuthorizationAdviceAbstractType">
           <sequence>
              <element ref="saml:AttributeDesignator" minOccurs=”0” maxOccurs=”unbounded”
/>
           </sequence>
           <attribute name="Reference” type=”anyURI” use="required" maxOccurs="unbounded"/>
        </extension>
    </complexContent>
</complexType>
```

5.2 Element <SimpleAuthorizationDecisionStatement>

The <SimpleAuthorizationDecisionStatement> element specifies the decision made about a corresponding SAML AuthorisationDecisionQuery request. Its purpose is to allow a response to the statement as a whole without enumeration of the rights in the response, which in turns allows for easier processing of the response by the requestor.

It has the complex type SimpleAuthorizationDecisionStatementType, which extends the StatementAbstractType by adding the following to it:

Decision [Required]

The decision made by the responder.

Recipient

If the ExtendedAuthorizationDecisionQuery to which Statement is in response, contained a Recipient attribute, this attribute MUST be present and its value MUST match the value of this field in the ExtendedAuthorizationDecisionQuery.

The following schema fragment defines the <SimpleAuthorizatonDecisionStatement> element and its SimpleAuthorizationDecisionStatementType complex type:

```
<element name="SimpleAuthorizationDecisionStatement"
type="SimpleAuthorizationDecisionStatementType"/>
<complexType name="SimpleAuthorizationDecisionStatementType">
    <complexContent>
        <extension base="saml:SubjectStatementAbstractType">
            <attribute name="Decision" type="saml:DecisionType" use="required"/>
            <attribute name=”Recipient” type=”anyURI” use=”optional”/>
        </extension>
    </complexContent>
</complexType>
```

6 SAML Authorization Element Usage in OGSA

This section is **normative**. It describes how SAML Authorization elements are used to meet OSGA requirements for authorization assertions and decisions as described in [OGSAAuthzReq]. It first describes the use of the AuthorizationDecisionQuery and ExtendedAuthorizationDecisionQuery elements, which are used by entities to request authorization assertions or decisions from an authorization service. This is followed by a description of the statements that can be returned in the response, either one or more standard AuthorizationDecisionStatement elements or a SimpleAuthorizationDecisionStatement element.

6.1 (Extended)AuthorizationDecisionQuery

A client MUST request an authorization decision using either an AuthorizationDecisionQuery or an ExtendedAuthorizationDecisionQuery (as defined in Section 5.1). This section describes constraints on fields in these elements.

The AuthorizationDecisionQuery element MUST include the following elements:

- A *Subject* element containing a *NameIdentifier* element specifying the identity of the initiator of the action being authorized.
- A *Resource* element specifying the resource (or domain of resources) to which the request to be authorized is being made.
- One or more *Action* elements specifying the action(s) being requested on the resource(s).

The query MAY include the following elements:

- Optionally one or more *Evidence* elements containing one or more supporting credentials about the initiator (or pointers to them), plus any contextual information, plus a public key certificate chain that may be used to authenticate the initiator.

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

6.1.1 NameIdentifier Element

This element, contained in the Subject element, contains the name of the initiator. The syntax of the NameIdentifer element is unchanged from the SAML specification. In some scenarios, the authorization service (PDP) MAY require the Subject 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 Subject.

6.1.1.1 X.509 Proxy Certificate Format Identifier

The SAML specification defines how some common identity types are asserted. This document defines how entities authenticated using X.509 Proxy Certificates [ProxyCerts] should be encoded. The SAML specification, in Section 7.3.3, defines method for expressing X.509 subject names that MUST be used for X.509 Proxy Certificate authenticated identities with the subject name of the end entity certificate that issued the proxy certificate chain as the subject name to be encoded.

The URI for this method is urn:oasis:names:tc:SAML:1.1nameid-format:X509SubjectName

6.1.1.2 Wildcard Subject Identifier

This document defines a method to be used in order to obtain public rights, that is, rights available to any subject. To indicate that such a request is being made, the NameIdentifier element MUST contain the following URI as the Format attribute:

```
http://www.gridforum.org/ogsa-authz/saml/2003/06/NameIdentifier/any
```

The Subject string MUST be "*", i.e., an asterisk.

6.1.2 SubjectConfirmation Element

When a subject was authenticated using the Grid Security Infrastructure and a X.509 Identity or Proxy Certificate, the SubjectConfirmation element should contain the X.509 certificate chain presented by the subject as follows:

The ConfirmationMethod element should contain the following URI:

```
http://www.gridforum.org/ogsa-authz/saml/2004/01/am/gsi
```

The SubjectConfirmationData element should contain the certificate chain presented by the subject encoded as a certificate path (i.e. an X509PKIPathv1 element) as described in [WSS-X509].

[Editor's note: It's not clear the X509PKIPathv1 element is specified yet in that document, should verify this before final version of this document.]

6.1.3 Resource Element

The Resource element is defined as a URI and is not changed from the SAML specification.

6.1.3.1 Grid Services

If the resource being referred to is a Grid service the resource element MUST contain the Grid Service Handle (GSH) of the service as described in [OGSI].

6.1.3.2 Wildcard Resource

This specification also defines a wildcard resource. This has two different meanings depending on whether it is in a query (request to a PDP) or a statement (response from a PDP):

- In an AuthorizationDecisionQuery or ExtendedAuthorizationDecisionQuery, the use of the wildcard resource URI states a desire by the entity making the query to learn the subject's rights on all the resources of which the authorization service is aware. Typically such a query will be used by an initiator who will cache the results and present them to resources later in a *decision push mode* of authorization.
- In an AuthorizationDecisionStatement, it states the subject has the given privileges on all resources that accept the authorization service as authoritative. This statement may be used when the authorization service is the authority for a group of resources with identical policy.

This wildcard URI MUST be specified as follows:

```
http://www.gridforum.org/ogsa-authz/saml/2003/06/resource/any
```

The Resource string must be "*", i.e., an asterisk.

6.1.4 Action Elements

The Action element describes the operation or method to be authorized. The Action element is composed of a string describing the operation and a URI specifying the namespace of the action.

6.1.4.1 Grid Service Operation Invocation

This specification defines the following namespace:

```
http://www.gridforum.org/namespaces/2003/06/ogsa-authz/saml/action/operation
```

This namespace is used to define an operation invocation on a Grid Service, specified in the Resource element, by the specified Subject. The action string should contain the name of the operation being invoked.

Note that operations regarding service data MUST be handled as actions on the service data itself as described in the following section.

6.1.4.2 Grid Service Data Access

[OGSI] defines service data elements (SDEs) associated with a Grid Services and methods for finding, setting and deleting SDEs. These actions are encoded in SAML Action elements by using the Action namespace to indicate the type of access (find, set or delete) and the Action value to indicate the name of the SDE on which the access is being attempted.

This scheme is intended to work with the queryByServiceDataNames QueryExpression and the setByServiceDataName and deleteByServiceDataNames UpdateExpressions as defined in Section 9.2 of [OGSI]. More complicated forms of these expressions may not fit into this scheme and it is expected they will require a more complicated method of encoding the expression and response.

```
http://www.gridforum.org/namespaces/2004/01/ogsa-authz/saml/action/sde/find
```

This namespace MUST be used to indicate a findServiceData operation (or its equivalent) being invoked on the specified Grid Service by the specified Subject. The action string MUST contain the QName of the Service Data element being accessed.

```
http://www.gridforum.org/namespaces/2003/06/ogsa-authz/saml/action/sde/set
```

This namespace is used to define the modification of a ServiceDataElement. The action string should contain the QName of the Service Data element being modified.

```
http://www.gridforum.org/namespaces/2003/06/ogsa-authz/saml/action/sde/delete
```

This namespace is used to define the deletion of a ServiceDataElement. The action string should contain the QName of the Service Data element being modified.

6.1.4.3 Wildcard Action

This specification also defines a wildcard action. This action has two different meanings depending on whether it is in a query or an assertion:

- In an AuthorizationDecisionQuery or ExtendedAuthorizationDecisionQuery, it states a desire to learn all of the subject's rights on the specified resource. An example of where this might be used, is by a policy enforcement point co-located with a resource, that expects a number of requests from a subject and will use a wildcard action query to obtain all of the subjects rights which it will cache as to do further access control without the contacting the authorization service.
- In an AuthorizationDecisionStatement, it states the initiator has all privileges on the resource. This will often be the case where the initiator is the policy authority for the resource in question.

This wildcard action MUST be specified as follows. The namespace URI MUST be:

```
http://www.gridforum.org/namespaces/2003/06/ogsa-authz/saml/action/wildcard
```

The Action string must be "*", i.e., an asterisk.

6.1.5 Evidence Elements

The AuthorizationDecisionQuery and ExtendedAuthorizationDecisionQuery elements may contain zero or more SAML Evidence elements which may be used to hold, either directly or by reference, supporting credentials regarding the initiator, as well as environmental parameters.

6.1.5.1 ReferenceStatement Element

Reference statements MAY be included within Evidence elements, in order to signal the *credential pull mode* of operation to the PDP. Reference statements MAY be included instead of, or as well as, credentials in Evidence elements, and it is a local matter for the PDP to determine how to handle the presence of one, both or neither elements.

If a Reference statement is present, then the Format attribute of the NameIdentifier element of the Subject element of the Reference statement SHOULD be #X509SubjectName, and the value MUST correspond to that of the Subject element of the AuthorizationDecisionQuery.

The value of the Reference URI is not further constrained by this specification.

6.2 Assertion Element

The SAML Assertion element is used by one entity to assert the capabilities of another. While an Assertion element can contain a variety of SAML statements, for the purposes of this document we consider only AuthorizationDecisionStatements, SimpleAuthorizationDecisionStatements (defined in this document) and AttributeStatements. The first two may be returned in response to AuthorizationDecisionQueries, whilst the latter may be presented in the Evidence elements of (Extended)AuthorizationDecisionQueries.

When returned by an authorization service to an entity, the Assertion element will be enveloped in a SAML Response element as described in the SAML specification.

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.
- Any number of *AuthorizationDecisionsStatements*
- Any number of *AttributeStatements* in Evidence elements

- An optional *Signature* element allowing the Assertion to be verified.

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

6.2.1 Conditions Element

Implementations SHOULD NOT use this element unless they are confident it will be understood by the PEP.

The Conditions element contains optional time constraints and any number of Condition elements (note difference in plurality between Conditions and Condition element names) on the returned assertion. Condition elements serve as an abstract element for extension, and should be used to express the policy conditions on operands and context/environment that the authorization service was unable to evaluate due to insufficient information being provided by the client. It is envisioned that future specification will be able to extend the Condition element to return fine-grained policies for parameters on operation invocation and service data access, using for example elements of XACML.

6.2.2 Advice Element

The Advice element MAY be ignored by the recipient of the assertion, therefore it MUST NOT contain any information essential to the operation of the PEP. Information that MAY be placed into the Advice Element includes: evidence supporting the assertion, and identification of the policy used in making the assertion.

6.2.3 AuthorizationDecisionStatement Element

The AuthorizationDecisionStatement element contains the same elements as the AuthorizationDecisionQuery, and also includes a Decision attribute.

The Decision attribute can take the value of Permit, Deny or Indeterminate. Indeterminate MUST be returned if the PDP could not render a decision do to error or lack of information.

6.2.4 AttributeStatement Element

The AttributeStatement element MAY be included in the Evidence element of an AuthorizationDecisionQuery, to signify attributes of the subject that were used when rendering the authorization decision. For example, when RBAC is being used, the attribute statement(s) could contain the role(s) of the initiator.

6.2.5 Signature Element

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

6.2.6 Required Assertion Fields

Major Revision

MUST be set to 1

Minor Revision

MUST be set to 1

AssertionID

SHOULD be set to a statistically unique 128 bit number

Issuer

This MUST be a string unambiguously identifying the issuer. A URI MAY be used. Where the Issuer name is an X.500 DN, it MUST have the format as specified in RFC 2255 [RFC 2255]. For example, if the issuer was a PDP with distinguished name of cn=PERMIS ADF, o=University of Michigan, c=us, the URI would be:

ldap:///cn=PERMIS%20ADF,o=University%20of%20Michigan,c=US

IssuerInstant

MUST be the date/time that the Assertion was issued in UTC form as specified in Section 1.2.2 of [SAML].

7 SAML Authorization Service PortType

This normative section has the WSDL that define the interface (operation and service data elements) that an OGSA Authorization service MUST define in its WSDL. These MUST be defined in addition to the basic Grid Service WSDL defined in Section 19.1 of [OGSI]. Authorization services MAY also define other service data or operation in addition to those defined in this section.

7.1 OGSA Authorization Service Service Data Declarations

The OGSA Authorization service portType includes the following serviceData elements:

7.1.1 supportedPolicies

This element MAY contain identifiers for any or all access control policies that authorization service is capable of rendering decisions regarding.

```
<sd:serviceData name="supportedPolicies"
        type="xsd:anyURI"
        minOccurs="0" maxOccurs="unbounded"
        mutability="mutable"
        modifiable="false"
        nillable="false"/>
```

7.1.2 supportsIndeterminate

This element expresses the authorization service's ability to return an Indeterminate decision. It is expected that some legacy systems may not allow the returning of Indeterminate.

```
<sd:serviceData name="supportsIndeterminate"
        type="xsd:boolean"
        minOccurs="1" maxOccurs="1"
        mutability="static"
        modifiable="false"
        nillable="false"/>
```

7.1.3 signatureCapable

This element expresses the authorization service's ability to sign the assertions and responses.

```
<sd:serviceData name="signatureCapable"
        type="xsd:boolean"
        minOccurs="1" maxOccurs="1"
        mutability="static"
        modifiable="false"
        nillable="false"/>
```

7.2 OGSA Authorization Service Operations

The OGSA Authorization service portType includes the following operations:

7.2.1 SAMLRequest

Input

- SAML Request Message

Output

- SAML Response Message

This operation defines the basic mechanism for which queries are sent to the authorization service and responses are returned. Faults will be encoded in the response in the standard SAML manner, so no faults are defined at the WSDL level.

```
<!-- The body of the request is exactly a samlp:Request -->
<message name="SAMLRequestMessage">
        <part name="body" element="samlp:Request"/>
</message>

<!-- The body of the corresponding response is exactly a samlp:Response -->
<message name="SAMLResponseMessage">
        <part name="body" element="samlp:Response"/>
</message>

<portType name="SAMLRequestPortType">
        <operation name="SAMLRequest">
           <input message="tns:SAMLRequestMessage"/>
           <output message="tns:SAMLResponseMessage"/>
        </operation>
</portType>
```

7.3 Full WSDL

The following is the WSDL used for the SAML-based authorization service. The first WSDL is for the SAML-specific portions of the authorization service. The second shows the SAML WSDL combined with the OGSI Grid Service WSDL to create a OGSI SAML Grid Authorization Service.

```
<definitions name="AuthorizationService"
   targetNamespace="http://www.gridforum.org/namespaces/2004/03/ogsa-authz/saml"
   xmlns:samlp="http://www.oasis-open.org/committees/security/docs/draft-sstc-schema-protocol-
19.xsd"
   xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/";
   xmlns="http://schemas.xmlsoap.org/wsdl/";
>
   <message name="SAMLRequestInputMessage">
     <part name="body" element=" samlp:Request "/>
   </message>

   <message name="SAMLRequestOutputMessage">
     <part name="body" element=" samlp:Response"/>
   </message>

   <gwsdl:portType name="SAMLRequestPortType">
     <operation name="SAMLRequest">
       <input message="tns:SAMLRequestInputMessage"/>
      <output message="tns:SAMLRequestOutputMessage"/>
     </operation>
   </gwsdl:portType>

   <sd:serviceData name="supportedPolicies" type="xsd:anyURI" minOccurs="0"
        maxOccurs="unbounded" mutability="mutable" modifiable="false" nillable="false"/>
   <sd:serviceData name="supportsIndeterminate" type="xsd:boolean" minOccurs="1"
        maxOccurs="1" mutability="static" modifiable="false" nillable="false"/>
   <sd:serviceData name="signatureCapable" type="xsd:boolean" minOccurs="1" maxOccurs="1"
        mutability="static" modifiable="false" nillable="false"/>
```

```
</definitions>
```

An OGSI SAML Authorization Service:

```
<definitions name="AuthorizationService"
      targetNamespace="http://ogsa.globus.org/samples/authzService">

 <import location="../../ogsi/ogsi.gwsdl"
      namespace="http://www.gridforum.org/namespaces/2003/03/OGSI"/>

 <import location="../../security/authorization/authz_port_type.gwsdl"
      namespace="http://www.gridforum.org/namespaces/2004/03/ogsa-authz/saml"/>

 <gwsdl:portType name="AuthzServicePortType" extends="ogsi:GridService
      authz:SAMLRequestPortType"/>
```

</definitions>

8 Security Considerations

This specification defines an authorization service based on the SAML specification for OGSA and is completely about security. Implementers of this specification need to take be aware that errors in implementation could lead to denial of service or improper granting of service to unauthorized users.

In particular, mutual authentication between the client and the PDP is highly desirable and strongly recommended. PDP implementations SHOULD sign assertions when they do not have an authenticated connection to the evaluator of the assertion, and MAY sign them when they do have. PDP implementations MAY be unwilling to respond to authorization decision queries from clients who are not authenticated.

9 Acknowledgements

Rebekah Lepro-Metz for XML advice and feedback.

The basic SAML operations WSDL in Section 7.2 was taken from a version by Irving Reid of Baltimore Technologies (in email to the OASIS SSTC: http://lists.oasis-open.org/archives/security-services/200302/msg00008.html).

10 Author Information

Von Welch
NCSA
vwelch@ncsa.uiuc.edu

Rachana Ananthakrishnan
Argonne National Laboratory
ranantha@mcs.anl.gov

Frank Siebenlist
Argonne National Laboratory
franks@mcs.anl.gov

Sam Meder
University of Chicago
meder@mcs.anl.gov

Laura Pearlman
Information Sciences Institute
University of Southern California

laura@isi.edu

David Chadwick
Information Systems Institute
University of Salford
d.w.Chadwick@salford.ac.uk

11 Glossary

This document uses the terms as defined in the Authorization Glossary as produced by the GGF Working Group on Authorization Frameworks and Mechanisms [Authz-Glossary].

The following additional terms are used in this document.

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)

12 Intellectual Property Statement

13 Full Copyright Notice

14 Normative References

[Authz-Glossary] Lorch, M., Skow, D., Thompson, M., "Authorization Glossary", Global Grid Forum, February 2004.

[OGSI] "Open Grid Service Infrastructure, Version 1.0", Global Grid Forum, April 5, 2003.

[RFC 2255] T. Howes, M. Smith. "The LDAP URL Format", RFC 2255, Dec 1997

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

[SAML] OASIS, Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML) V1.1, May 2003.

[WSS-X509] "Web Services Security X.509 Certificate Token Profile", Working Draft 10, 19th August 2003

15 Informational References

[OASIS-SSTC] OASIS Security Services Technical Committee. http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security. January, 2004.

[OASIS-XACML] OASIS eXtensible Access Control Markup Language Technical Committee. http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml. January, 2004.

[OGSA] Foster, I., C. Kesselman, J. Nick, S. Tuecke, "The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration", XXX

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

[OpenSAML] OpenSAML - an Open Source Security Assertion Markup Language implementation. http://www.opensaml.org. January, 2004.

[ProxyCerts] Welch, V., Foster, I., Kesselman, C., Mulmo, O., Pearlman, L., Tuecke, S., Gawor, J., Meder, S., Siebenlist.F., "X.509 Proxy Certificates for Dynamic Delegation" *3rd Annual PKI R&D Workshop, 2004.*

SAML Authorization Overview

The SAML specification [SAML] defines a number of elements for making assertions and queries regarding authentication, authorization decisions and attributes. It also supports extensibility by allowing applications to define their own elements. In this section we give a brief **non-normative overview** of the elements related to authorization, and the additional elements needed for Grid authorization. Readers are encouraged to review the SAML specification for more details.

A.1 SAML Version

This specification is based on the SAML v1.1 specification. This specification defines a number of extensions to SAMLv1.1 which are described in Section 4, that are necessary for Grid authorization,. The authors are aware that OASIS is currently working on SAMLv2.0. Indeed, the authors are working closely with the OASIS organization to help ensure that SAMLv2.0 contains the extensions described in this specification (and if not syntactically identical, then at least having the same semantic content). Once SAMLv2.0 has been published, it is the intention of the authors to migrate this specification to SAMLv2.0.

A.2 SAML Authorization Model

As shown in Figure 1~~Figure 1~~, SAML defines a message exchange between a policy enforcement point (PEP) and a policy decision point (PDP) consisting of an AuthorizationDecisionQuery (2) flowing from the PEP to the PDP, with an Assertion returned containing some number of AuthorizationDecisionStatements (3). We also define an extension to SAML to support exchanges in which a client can issue an AuthorizationDecisionQuery to a server, and have an Assertion returned containing a simple AuthorizationDecision.

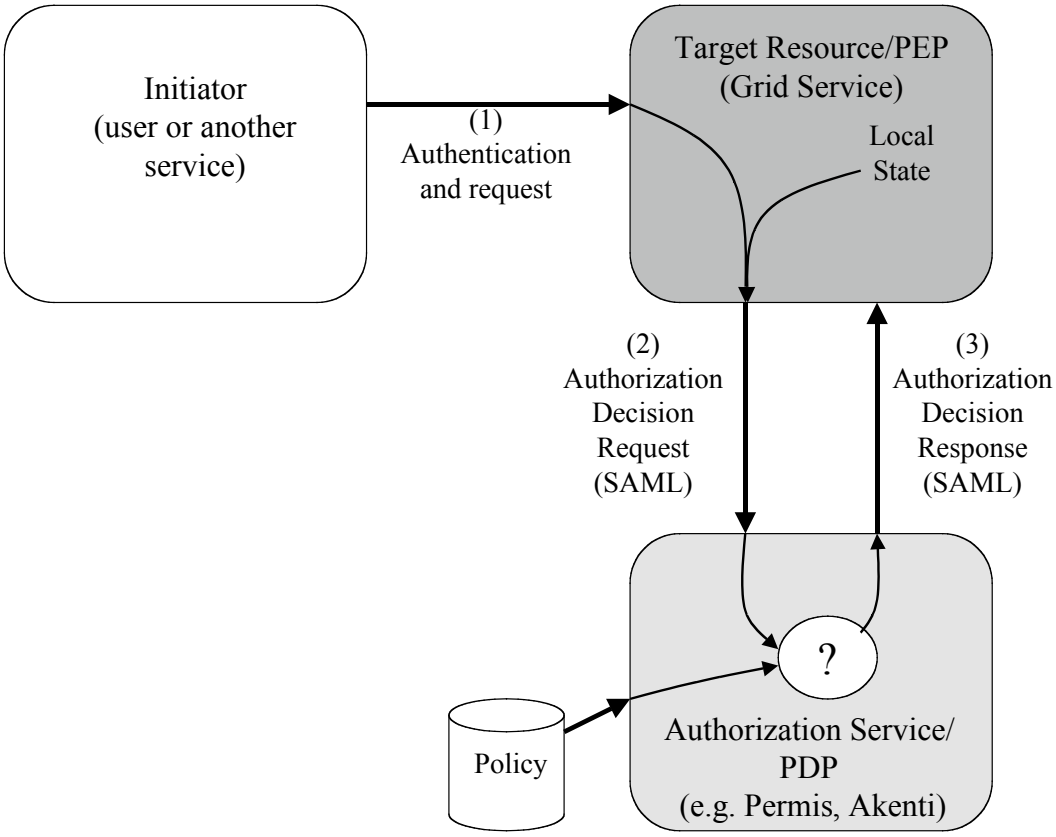

Figure 1: SAML message flow. (1) A request arrives at the target resource. (2) The Grid Service generates and sends a SAML AuthorizationDecisionQuery to an Authorization

Service. (3) The service evaluates the request against policy and returns a response encoded as a SAML Assertion.

In the following sections we describe the AuthorizationDecisionQuery and the Assertion element, and the elements that are used to compose these.

A.3 Action Element

The Action elements allows for the expression of actions that may be attempted by entities and expressed in policy. This element consists of a string and a URI defining a namespace for the action described in the string.

For example the SAML specification defines a namespace for HTTP operations that defines actions of GET, HEAD, PUT, POST.

A.4 Resource Element

The Resource element is used to identify the target on which the policy is being asserted or requested. This element is simply a URI.

A.5 Subject and NameIdentifier Elements

The Subject element contains a NameIdentifier element as well as some elements outside the scope of this document. In SAML authorization assertions, the NameIdentifer element serves to identify the initiator of the action being authorized. The NameIdentifer element contains a string to hold an identity that has two attributes:

- The NameQualifier attribute is a string expressing the security or administrative domain that defined the name (e.g. Kerberos realm, CA name).
- The Format attribute is a URI identifying the format of the name (e.g. X509 subject name).

A.6 AuthorizationDecisionStatement Element

The AuthorizationDecisionStatement element contains statements regarding authorization policy. Each of these statements contains a *Subject* element, identifying the entity whose rights are being expressed, a *Resource* element, identifying the resource(s) the rights apply to, an optional *Evidence* element holding the assertions the issuer relied upon in making its decision, any number of *Action* elements (expressing the allowed or denied operations) and the *Decision* attribute containing the authorization decision. The assertion may also optionally contain a *Conditions* element expressing the conditions that must be fulfilled before the authorization can be permitted and an *Advice* element providing additional information related to the authorization decision which may be ignored by the recipient.

A.7 AttributeStatement Element

This element supplies a statement by the issuer that the specified subject is associated with the specified attribute(s).

A.8 Assertion Element

The Assertion element specifies the basic information that is common to all SAML assertions, and optionally it may be signed. It can contain any number of Statements, for example, AuthorizationDecisionsStatements and AttributeStatements. It is also capable of containing statements related to authentication, but for the purposes of this document we only consider Assertions containing AttributeStatements, AuthorizationDecisions and AuthorizationDecisionStatements.

A.9 Conditions Elements

Each Assertion element may contain any number of Conditions elements. Conditions elements are specified to express policy restrictions on the assertion such as a validity time of the Assertion. However they are extendable to express arbitrary conditions on the use of the

assertion. Condition elements might typically be added to assertions if the decision engine had insufficient information to be able to evaluate the policy locally.

A.10 Advice Elements

An Assertion element may contain any number of Advice elements. Advice elements hold information related to the assertion, but they may be ignored by applications that do not support them. Examples of information that could be included in an Advice element are: an identifier of the policy that was used by the PDP when making its authorization decision, and assertions that were used by the PDP when making its authorization decision.

A.11 AuthorizationDecisionQuery Element

The AuthorizationDecisionQuery element allows for the request of AuthorizationDecisionStatements and simple AuthorizationDecision responses. It contains a Subject, Resource, optional Evidence, and any number of Action elements that identify the decisions that the initiator wants to be made; as well as a RespondWith element that identifies the type of response that the client wishes to be returned.

A.12 Evidence Elements

Evidence elements allow for queries to provide information to the PDP that may be useful for its decision-making. They are used to hold the credentials of the initiator, as well as contextual and environmental information. The initiator's credentials may be either included directly in the evidence element (as AttributeStatements), or may be included indirectly via a pointer (as ReferenceStatements). This allows the PDP to support both the credential push and pull mode of operation. In responses, they also allow the PDP to express what information it used to make its decision.

Each AuthorizationDecisionStatement and AuthorizationDecisionQuery element can contain an Evidence element. Each Evidence element can contain any number of Assertion or AssertionIDReference elements that affect the policy decision process.

Appendix B. Intellectual Property Issues with SAML

RSA (http://www.rsa.com) claims intellectual property rights on portions of the SAML specification. They offer a reciprocal license to implementers of SAML. Details of their claim and the license may be found at: http://www.rsasecurity.com/solutions/standards/saml/

Appendix C. Globus Toolkit version 4 use of Callout: URI encoding of EPR

Version 4 of the Globus Toolkit (GT4) is based on the Web Services Resource Framework (WSRF), which uses the WS-Addressing specification to identify service endpoints.

We provide here a brief description of how GT4 encodes the address for its services into a URI. This is intended to provide a non-normative example of how more complex address structures can be encoded into simple URIs.

1. The Reference Properties element is canonicalized and hashed using SHA-1
2. The hash is encoded using Base64 encoding.
3. The URI identifying the service is formed from the Address element URI , a question mark ("?") and the encoded hash of the Reference Properties element.

Here is an example of a resulting URI using this process. Everything prior to the "?" is the URI from the address element; everything after the "?" is the hash of the Reference Properties element:

http://192.168.1.100:3411/wsrf/services/AuthzCalloutTestService?q4D+31NtjfehDAnn07NUwBP2j34=

~~Appendix C.~~Appendix D. ChangeLog

This section to be deleted by the GGF editor prior to publication.

Changes from May, 2004 to December, 2004 version:

- Added Appendix C giving an example of how GT4 creates URIs from WS-Addressing elements.
- Marked Recipient field in 5.1 as depreciated since it is insufficient to stop replay attacks.

Changes from January, 2004 to ~~current version~~May, 2004 version:

- Split references into Normative and Informational
- Added WSDL to section 7
- Deleted section 6.1.5.1 due to lack of clarity and content.
- Section 5.2: Removed InResponseTo element from SimpleAuthorizationDecisionStatement since it is already in Response element.
- Cleaned up references
- Section 5.1.2: Corrected base for SubjectAttributeReferenceAdvice to AuthorizationAdviceAbstractType