A Grid Authorization Model for Science Gateways

Tom Scavo
National Center for Supercomputing Applications
1205 W. Clark St., Room 1008
Urbana, IL 61801 USA
+1 217 265 8759
tscavo@ncsa.uiuc.edu

Von Welch
National Center for Supercomputing Applications
1205 W. Clark St., Room 1008
Urbana, IL 61801 USA
+1 217 265 7139
vwelch@ncsa.uiuc.edu

ABSTRACT

Since the number of TeraGrid Science Gateways is expected to grow at least an order of magnitude in the next few years, a lightweight gateway deployment model is sought, one that facilitates growth but meets the security requirements of the TeraGrid. To this end, we present a new authorization model for science gateways based on the community account model. This authorization model significantly increases the information flow between the gateway and the resource provider, without requiring new wire protocols or extensive new middleware infrastructure. Instead, the model complements existing technology and promises to leverage emerging federated identity deployments directly.

Categories and Subject Descriptors

C.2.4 [**Computer-Communication Networks**]: Distributed Systems – *client/server, distributed applications.*

General Terms

Management, Security.

Keywords

TeraGrid, Science Gateway, authorization, attribute-based access control, SAML, GridShib, Shibboleth.

1. INTRODUCTION

In the traditional grid computing model, the user logs in directly to the server, interacting with a resource at a relatively low level, using, for example, a command-line shell. A grid portal, on the other hand, puts a comfortable layer of abstraction between the user and the resource by presenting the user with a web-based interface. This medium generally being preferred by users, the grid portal continues to be a popular model for grid computing.

A *TeraGrid Type I Science Gateway*[1] is a type of web portal that requests a grid service on behalf of the user. It is anticipated that the number of science gateways will increase by an order of magnitude (or more) in the next few years [9]. Thus a lightweight gateway deployment model is of interest, one that facilitates growth but meets the security requirements of the TeraGrid [13].

A distinguishing characteristic of a science gateway is the underlying authorization model. Of particular interest is a science gateway based on the *community account model* [12], that is, a gateway that authenticates as itself to the resource provider using a proxy certificate issued and signed by a so-called community credential. On the server side, the resource maps a gateway request to a single community account shared by all users.

This paper presents a new authorization model for grids in general and for science gateways based on the community account model in particular. This authorization model significantly increases the information flow between the gateway and the resource provider, without requiring new wire protocols or extensive new middleware infrastructure. The underlying information model is based on *Security Assertion Markup Language* [7], a widely deployed technology on today's campuses (under the guise of *Shibboleth*[2]). Thus the authorization model complements existing technology and, as suggested in the Epilogue, promises to leverage existing Shibboleth deployments directly.

2. GRID SECURITY INFRASTRUCTURE

In Globus Toolkit 4.0 (GT4), security is based on *Grid Security Infrastructure* (GSI) [11]. Under GSI, users authenticate themselves by presenting trusted X.509 certificates to relying parties, either X.509 end entity certificates [4] or X.509 proxy certificates [10]. A distinguishing feature of GSI is its reliance on X.509 proxy certificates. Presentation of the proxy certificate may occur at the transport level (GSI Transport) or the message level (GSI Secure Message and GSI Secure Conversation). At the transport level, an X.509 proxy certificate is transmitted via SSL/TLS, while at the message level, WS-Security X.509 Token Profile [8] is used.

Figure 1 illustrates a Globus web services client authenticating to a Globus web service using a GSI proxy credential. How the client obtains the proxy credential in the first place depends on how the corresponding end entity credential is managed. Specifically, the proxy is either created using a command-line tool like `grid-proxy-init`[3] or retrieved from a MyProxy server using `myproxy-logon`[4]. In any event, the client presents the proxy certificate and proves possession of the corresponding

Scavo, T. and Welch, V. A Grid Authorization Model for Science Gateways, International Workshop on Grid Computing Environments 2007, November 11, 2007, Reno, NV, USA. An electronic version of this document will be available at http://casci.rit.edu/proceedings/gce2007

[1] http://www.teragrid.org/programs/sci_gateways/

[2] http://shibboleth.internet2.edu/

[3] http://www.globus.org/toolkit/docs/4.0/security/prewsaa/rn01re05.html

[4] http://grid.ncsa.uiuc.edu/myproxy/MyProxyLogon/

private key. In the simplest case, the service uses the distinguished name (DN) in the X.509 certificate as a basis for access control, that is, the service consults an access control list of DNs, called a *gridmap*. Authorization based on gridmap files is an example of *identity-based access control*.

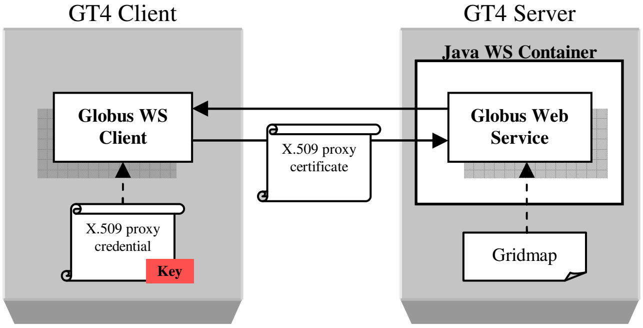

Figure 1. Grid Security Infrastructure

As grids have grown in size, the overhead associated with gridmap file management has likewise grown. Thus there has been considerable interest in alternative, more scalable approaches to authorization that permit fine-grained access control decisions. One such alternative approach is called *attribute-based authorization*. Instead of basing its access control decision on coarse-grained identity, a service instead relies on certain user attributes associated with the client request. For example, policy may stipulate that any member of a particular user group may access the service. Individual users may come and go, but the user group persists and forms the basis of a more robust access control decision. Policy may combine group membership with other attributes (such as roles) to permit fine-grained access control decisions.

2.1 Introducing SAML into GSI

OASIS *Security Assertion Markup Language* (SAML) is an XML standard for exchanging authentication and authorization data between security domains[5]. Here we concentrate on the core features of SAML as a markup language [5], which are independent of the various bindings and profiles [6] defined in the OASIS SAML set of specifications.

Current versions of GT4 use SAML to convey authorization decisions[6] but other security information such as attributes are not leveraged by GT4. GridShib, a Globus incubator project, attempts to fill this gap.

GridShib [2] is a project whose goal is to introduce attribute-based authorization to Globus-based grids. GridShib distributes four software components[7], two of which will be discussed here: GridShib SAML Tools and GridShib for Globus Toolkit. These two software components produce and consume (resp.) SAML assertions bound to X.509 certificates.

2.2 An X.509-bound SAML Token

The OASIS Web Services Security Technical Committee [8] has published the WS-Security Core Specification and a handful of token profiles including the WS-Security X.509 Token Profile and the WS-Security SAML Token Profile (see Figure 2). In fact, Globus GSI Secure Message implements the WS-Security X.509 Token Profile. In contrast, GridShib relies on a hybrid security token that we call an *X.509-bound SAML Token*.

By extension, an implementation of the WS-Security X.509 Token Profile (such as GSI Secure Message) is automatically an implementation of the X.509-bound SAML Token Profile employed by GridShib. This is a distinct advantage since it obviates the need to implement yet another GSI wire protocol (such as WS-Security SAML Token Profile). In other words, since GridShib binds SAML assertions to GSI-compliant X.509 certificates, no modifications to existing GSI protocols are required.

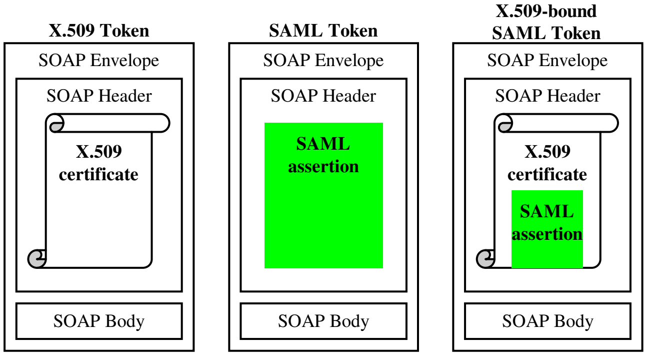

Figure 2. WS-Security Tokens

2.3 GridShib-enabled GSI

GridShib for Globus Toolkit (GT) is a plug-in for Globus Toolkit 4.0 (and later). Here we concentrate on the ability of GridShib for GT to consume X.509-bound SAML assertions issued by the GridShib SAML Tools. These SAML assertions contain attributes, which are used by the GT authorization framework to make local access control decisions.

We exploit the capability of the *GridShib SAML Tools* to issue SAML assertions and optionally bind these assertions to X.509 proxy certificates (see Figure 3). Instead of using `grid-proxy-init` or MyProxy to create a proxy credential, a user installs GridShib SAML Tools on the client system and, through a command-line interface, issues a SAML assertion and embeds it in a proxy certificate. For example, a simple command such as the following might be used:

```
$ gridshib-saml-issuer --user trscavo --authn --x509
```

In the above command, the argument to the `--user` option is the SAML Subject, the `--authn` option indicates a SAML `AuthenticationStatement` is desired, and the `--x509` option indicates that the output should be an X.509 proxy credential.

[5] http://en.wikipedia.org/wiki/SAML

[6] http://www.globus.org/toolkit/docs/4.0/security/cas/

[7] http://gridshib.globus.org/download.html

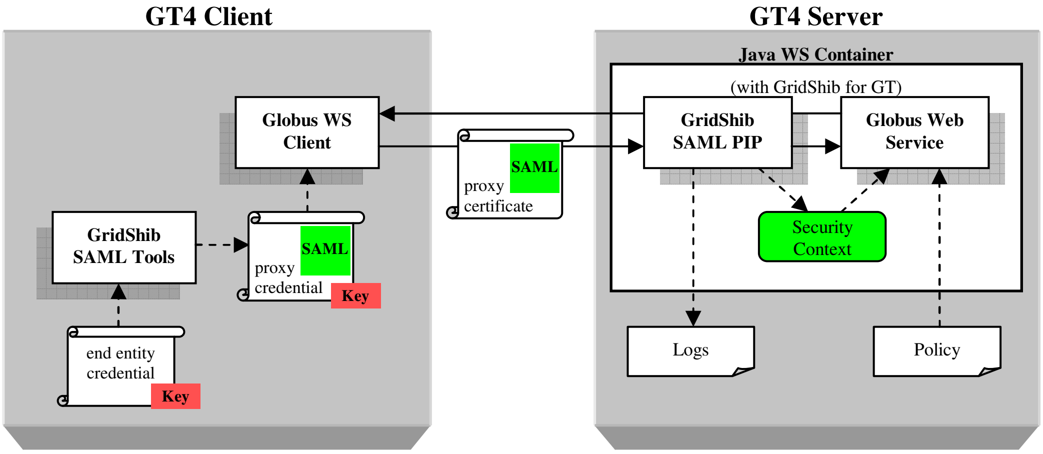

Figure 3. GridShib-enabled GSI

Note that the client in Figure 3 is an ordinary Globus web services client, that is, the client is not aware that the proxy certificate contains a SAML assertion. Indeed the service need not be aware that the proxy certificate contains a SAML assertion either. If the service is not GridShib-enabled, the SAML assertion is simply ignored by the GT runtime.

If, on the other hand, the service has GridShib for GT installed, a SAML assertion policy information point (PIP) will extract and parse the bound SAML and populate a security context with the resulting security information. Subsequently, an appropriately configured service combines the information from the security context with locally declared policy to arrive at an access control decision.

Note that the SAML assertion PIP can be configured at the container level or the service level. If configured at the container level, the security context is available to all services running in the container.

3. THE TERAGRID SCIENCE GATEWAY

For the remainder of this paper, we focus on the *TeraGrid Type I Science Gateway*, a type of web portal that requests a grid service on behalf of the user. Of particular interest is a science gateway based on the community account model.

3.1 The Community Account Model

A science gateway based on the *community account model* (Figure 4) is relatively easy to implement, but it is well known that the basic model has some significant drawbacks [12]. In particular, all requests from the science gateway look the same to the resource provider, so there is no opportunity for fine-grained access control since the end user (let alone the user's attributes) is unknown to the resource provider. Other disadvantages of the basic community account model include the following:

- The community account model (necessarily) exhibits course-grained authorization at the resource. Since there is only one account—the community account—there is a single set of privileges applied across the board to all members of the community. Even if the gateway had a fine-grained access control framework in place, there is no simple way to communicate attribute and authorization information to the resource.
- The community account model requires a rather convoluted approach to auditing. In the event of an incident, the resource provides details from its logs (timestamp, command, arguments), which the gateway must correlate with its own logs. Then the gateway administrators contact the user, attempt to resolve the problem, and report back to the resource administrators who may follow up with their own investigation. In this sense, the gateway is an awkward man-in-the-middle between the user and the resource.
- If a user or process misbehaves, the resource provider has no choice but to disable access to the entire community (by removing the DN of the community credential from the gridmap file).
- In the science gateway model, where the gateway is acting on behalf of the user, accounting is difficult even if we assume a free flow of information between the gateway and the resource. At first glance, the gateway seems like a natural place to centralize accounting, but upon closer inspection, we find this leads to some fairly significant issues. On the other hand, for the resource to handle accounting, it would need to have a persistent identifier for the user and need to know the user's community allocation, which is often determined by the community at the time the user joins the community. So neither party, gateway nor resource, is equipped to do accounting on its own.

Existing Science Gateways[8] are working to overcome the limitations of the basic community account model, some quite successfully. It seems, however, that none of the existing models is lightweight enough to scale to large numbers of science gateways, which explains why the number of gateways remains small. Thus we propose an alternate approach that is both lightweight and secure.

The proposed model incorporates GridShib SAML Tools at the gateway and GridShib for GT at the resource provider. With these two components installed, the gateway passes information that the resource can use for fine-grained access control, auditing and incident response.

[8] http://www.teragrid.org/programs/sci_gateways/programlist.php

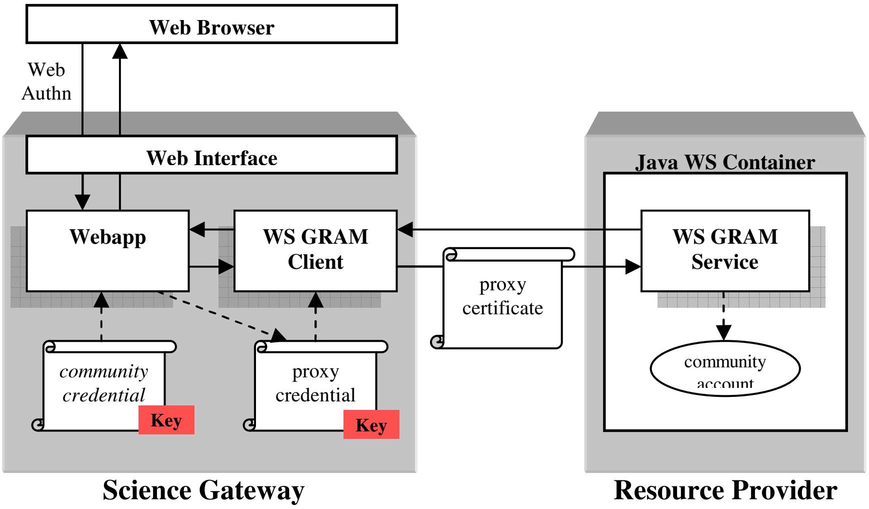

Figure 4. Science Gateway with Community Credential

3.2 A GridShib-enabled Science Gateway

After the user authenticates to the portal on the front channel and indicates via the UI that a back channel request is required, the portal initiates a grid request on behalf of the user. The portal issues a SAML assertion containing the user's authentication context and attributes, binds the SAML assertion to a proxy certificate signed by the community credential, and then authenticates to the resource by presenting the SAML-laden proxy certificate. See Figure 5.

So the basic idea is to bind a SAML assertion to the proxy certificate that the science gateway presents to the resource provider. The SAML assertion may carry three types of security information:

1. *Authentication context*
2. *Attributes*
3. *Authorization decisions*

Authentication context is information about the act of authentication at the portal. Although web authentication is out of scope (i.e., the model does not assume or require a particular type of authentication), details such as authentication method, authentication instant, and the IP address of the authenticated user are presumably of interest to the resource provider and may be used for access control.

Of the three types of information in an assertion, attributes are most useful for the purposes of access control. Examples of such attributes include e-mail address, persistent identifier, entitlements, and most importantly, group memberships and roles.

Instead of passing attributes that the resource uses to make an access control decision, the gateway can predetermine an authorization decision, encode it in a SAML assertion, and pass the decision to the resource. For example, the Globus Community Authorization Service[9] uses SAML in this way.

Finally, authorization decisions may be used to limit the scope of the containing proxy certificate. If, for example, the resource uses GSI delegation, an authorization decision embedded in the proxy certificate can be used to limit the scope of the delegation. Used in this way, SAML authorization decisions bound to proxy certificates are an extension of the simple all-or-none proxy policy rules specified in RFC 3820 [10].

As a simple example, suppose the gateway wishes to assert two attributes: a group membership ("isMemberOf") and a role ("eduPersonEntitlement"). Using the SAML Tools, the gateway issues a SAML assertion (Appendix A) and binds it to a proxy certificate.

Since the "isMemberOf" attribute is fixed, it is specified in a configuration file (Appendix B). The value of the "eduPersonEntitlement" attribute, on the other hand, varies from request to request, so it is specified on the command line:

```
$ gridshib-saml-issuer --user trscavo --authn --x509    \
    --outfile /tmp/gridshib-proxy.pem --properties      \
      Attribute.mail.Name=                              \
        urn:mace:dir:attribute-def:eduPersonEntitlement \
      Attribute.mail.Value=                             \
        http://www.teragrid.org/names/roles/guest
```

Taken together, the configuration properties (Appendix B) and the command-line options (above) produce the SAML markup shown in Appendix A.

Note that a SAML assertion issued by the GridShib SAML Tools need not be signed. This is because the assertion inherits its signature from the containing certificate, which was issued by the same entity. We call such an assertion a *self-signed assertion*, the simplest type of assertion that can be bound to an X.509 certificate.

The user identity and attributes exposed in the SAML assertion allow for fine-grained access control, strong auditing and effective incident response. Additional infrastructure is needed, however, to address accounting and other lingering issues with the community account model.

3.3 Policy at the Resource Provider

As mentioned earlier, with GridShib for GT installed at the resource provider, a SAML security context exposes the security

[9] http://www.globus.org/toolkit/docs/4.0/security/cas/

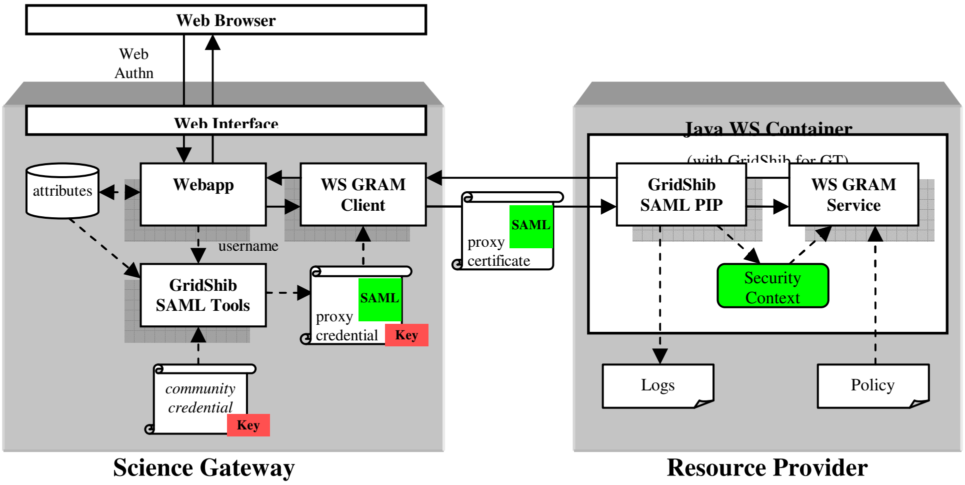

Figure 5. GridShib-enabled Science Gateway

information bound to the proxy certificate. A policy decision point (PDP) combines the information in the security context with one or more policy files to arrive at an access control decision.

For example, suppose a service declares the following two-step policy:

1. Access is limited to a user who is a member of a recognized community.
2. Access is further limited to a user acting in an acceptable role.

Two policy files are needed in this case, both of which must be satisfied. In the first policy file (Appendix C), we assume there are two communities recognized by this particular service: nanoHUB and GISolve. Community membership is indicated by the values http://www.nanohub.org and http://www.gisolve.org, respectively. At step 1, the PDP returns PERMIT if and only if the security context contains an "isMemberOf" attribute with either of these values. Note that the latter are URIs used here to represent community membership.

Further suppose that the service accepts requests from users acting in one of three roles: admin, user, or guest (Appendix D). At step 2, the PDP returns PERMIT if and only if there is an "eduPersonEntitlement" attribute having one of these three values.

Note that both policy files associate a username with each attribute name-value pair. If that name-value pair is contained in the X.509-bound SAML assertion, the corresponding username is added to the security context. The service uses this information to map the request to an appropriate system account (tg-admin, tg-user, or tg-guest, resp.), much like a grid-mapfile maps DNs to system accounts.

3.4 Comparison with other Grid Authorization Models

The most successful attribute-based grid authorization model in use today is the *Virtual Organization Membership Service* (VOMS) [1], which is used in both Europe (EGEE) and the U.S. (OSG). A VOMS server issues a signed X.509 attribute certificate (AC) [3] that a client typically binds to an X.509 proxy certificate (cf. Figure 3). In that sense, the GridShib approach is similar to VOMS since both models bind attributes to proxy certificates.

Aside from the different attribute formats, a major difference between VOMS and GridShib is that VOMS requires the requester to be the subject. In other words, VOMS will not issue an AC to a requester acting on behalf of the subject. For this reason, a gateway can not call out to a VOMS server to obtain attributes, and thus VOMS can not be used as a basis for gateway security.

Instead of resolving attributes locally, a science gateway could call out to a SAML Attribute Authority and request attributes on behalf of the subject (which the GridShib SAML Tools supports, in fact). However, not only is attribute query more difficult to deploy, but a protocol to request a forwardable SAML token is not readily available and would have to be specified. All things considered, local attribute resolution appears to be the simplest option open to the science gateway.

4. EPILOGUE

In the current model, the browser-facing components of the science gateway are considered out of scope. If the number of science gateways is to grow as expected, this aspect of gateway infrastructure deserves consideration as well. Along these lines, it is well known that password management, from both the user's and administrator's point of view, represents a significant investment in infrastructure. To eliminate passwords, we will consider the possibility of introducing federated identity at the science gateway.

The federating technology with the greatest presence on today's campuses is *Shibboleth*, an open source implementation of the SAML Browser Profiles [6]. The advantages of Shibboleth include the following:

- **Ubiquity:** Shibboleth has a large (and growing) installed base on campuses around the world.
- **Manageability:** Shibboleth mitigates the need to manage passwords at the science gateway (which is a huge administrative burden).

- **Usability:** For users, Shibboleth means fewer passwords to remember and fewer user interfaces to become familiar with.
- **Security:** Shibboleth relies on existing campus identity management infrastructure that is more likely to be associated with positive identity vetting, strong authentication, and privacy protection (motivated by FERPA).

Since Shibboleth is based on SAML, the grid authorization model presented here complements existing campus infrastructure. Finally, to minimize the impact of deploying Shibboleth at the science gateway, we will also investigate the possibility of standing up a TeraGrid-wide proxy service that mediates authentication at the user's home institution and asserts community attributes to the science gateway.

5. ACKNOWLEDGMENTS

The GridShib work was funded by the NSF National Middleware Initiative (NMI awards 0438424 and 0438385). Opinions and recommendations in this paper are those of the authors and do not necessarily reflect the views of NSF.

The Science Gateway integration work is funded by the NSF TeraGrid Grid Integration Group through a sub-award to NCSA.

“Globus Toolkit” is a registered trademark of the University of Chicago. “Shibboleth” is a registered trademark of Internet2.

6. REFERENCES

[1] Alfieri, R., Cecchini, R., Ciaschini, V., dell’Agnello, L., Frohner, A., Löréntey, K., and Spataro, F.. *From gridmapfile to voms: managing authorization in a grid environment*. Future Generation Comp. Syst., 21(4):549–558, 2005.

[2] Barton, T., Basney, J., Freeman, T., Scavo, T., Siebenlist, F., Welch, V, Ananthakrishnan, R., Baker, B., Goode, M., and Keahey, K.. *Identity Federation and Attribute-based Authorization through the Globus Toolkit, Shibboleth, Gridshib, and MyProxy*. In 5th Annual PKI R&D Workshop, April 2006. http://grid.ncsa.uiuc.edu/papers/gridshib-pki06-final.pdf

[3] Farrell, S. and Housley, R.. *An Internet Attribute Certificate Profile for Authorization*. IETF RFC 3281, April 2002. http://www.ietf.org/rfc/rfc3281.txt

[4] Housley, R., Polk, W., Ford, W., and Solo, D.. *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*. IETF RFC 3280, April 2002. http://www.ietf.org/rfc/rfc3280.txt

[5] Maler, E. et al. *Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V1.1*. OASIS Standard, September 2003. Document ID oasis-sstc-saml-core-1.1. http://www.oasisopen.org/committees/download.php/3406/oasis-sstc-saml-core-1.1.pdf

[6] Maler, E. et al. *Bindings and Profiles for the OASIS Security Assertion Markup Language (SAML) V1.1*. OASIS Standard, September 2003. Document ID oasis-sstc-saml-bindings-1.1. http://www.oasisopen.org/committees/download.php/3405/oasis-sstc-saml-bindings-1.1.pdf

[7] *OASIS Security Services (SAML) Technical Committee*. http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security

[8] *OASIS Web Services Security (WSS) Technical Committee*. http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wss

[9] *TeraGrid Authentication, Authorization and Account Management Workshop*. Argonne National Laboratory, Argonne, Illinois, August 30–31, 2006. http://www-fp.mcs.anl.gov/tgmeeting/AAA-Agenda.htm

[10] Tuecke, S., Welch, V., Engert, D., Pearlman, L., and Thompson, M.. *Internet X.509 Public Key Infrastructure (PKI) Proxy Certificate Profile*. IETF RFC 3820, June 2004. http://www.ietf.org/rfc/rfc3820.txt

[11] Welch, V. (ed.) *Globus Toolkit Version 4 Grid Security Infrastructure: A Standards Perspective*. http://www.globus.org/toolkit/docs/4.0/security/GT4-GSI-Overview.pdf

[12] Welch, V., Barlow, J., Basney, J., Marcusiu, D., and Wilkins-Diehr, N. *A AAAA model to support science gateways with community accounts*. Concurrency and Computation: Practice and Experience, 19(6):893–904, 2006. http://dx.doi.org/10.1002/cpe.1081

[13] Welch, V., Foster, I., Scavo, T., Siebenlist, F., Catlett, C., Gemmill, J., and Skow, D. *Scaling teragrid access: A testbed for identity management and attribute-based authorization*. In TeraGrid 2007, June 2007. http://grid.ncsa.uiuc.edu/papers/welch-tg07-Idm-final.pdf

Appendix A

Here is an example of a self-issued assertion (with whitespace added for readability) created by GridShib SAML Tools:

```
<saml:Assertion
  xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  AssertionID="_3f89fecd323e4f1fa0f897b0ebb5f81c"
  IssueInstant="2007-10-29T16:46:35.673Z"
  Issuer="https://gridshib.gisolve.org/idp"
  MajorVersion="1" MinorVersion="1">
  <saml:AuthenticationStatement
    AuthenticationInstant="2007-10-29T16:46:34.872Z"
    AuthenticationMethod="urn:oasis:names:tc:SAML:1.0:am:unspecified">
    <saml:Subject>
      <saml:NameIdentifier
```

```
        Format="urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified">
        trscavo
      </saml:NameIdentifier>
    </saml:Subject>
  </saml:AuthenticationStatement>
  <saml:AttributeStatement>
    <saml:Subject>
      <saml:NameIdentifier
        Format="urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified">
        trscavo
      </saml:NameIdentifier>
    </saml:Subject>
    <!—FriendlyName="isMemberOf" -->
    <saml:Attribute
      AttributeName="urn:oid:1.3.6.1.4.1.5923.1.5.1.1"
      AttributeNamespace="urn:mace:shibboleth:1.0:attributeNamespace:uri">
      <saml:AttributeValue xsi:type="xsd:string">
        http://www.gisolve.org
      </saml:AttributeValue>
    </saml:Attribute>
    <saml:Attribute
      AttributeName="urn:mace:dir:attribute-def:eduPersonEntitlement"
      AttributeNamespace="urn:mace:shibboleth:1.0:attributeNamespace:uri">
      <saml:AttributeValue xsi:type="xsd:string">
        http://www.teragrid.org/names/roles/guest
      </saml:AttributeValue>
    </saml:Attribute>
  </saml:AttributeStatement>
</saml:Assertion>
```

Appendix B

This is a simple example of a configuration properties file read by the GridShib SAML Tools:

```
# GridShib SAML Tools: SAML Assertion Issuer Tool config properties

# Identity Provider entityID
IdP.entityID=https://gridshib.gisolve.org/idp

# SAML NameIdentifier
NameID.Format=urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified
NameID.Format.template=%PRINCIPAL%

# SAML Attribute
Attribute.isMemberOf.Name=urn:oid:1.3.6.1.4.1.5923.1.5.1.1
Attribute.isMemberOf.Value=http://www.gisolve.org

# X.509 Issuing Credential
certLocation=file:///etc/grid-security/community-cert.pem
keyLocation=file:///etc/grid-security/community-key.pem
```

Appendix C

This policy file requires membership in at least one of the two listed communities:

```
<attributePolicy
  xmlns="http://gridshib.globus.org/2005/08/policy"
  xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion">
  <entry>
    <listOfAttributes>
      <!-- FriendlyName="isMemberOf" -->
      <saml:Attribute
        AttributeName="urn:oid:1.3.6.1.4.1.5923.1.5.1.1"
        AttributeNamespace="urn:mace:shibboleth:1.0:attributeNamespace:uri">
        <saml:AttributeValue>http://www.nanohub.org</saml:AttributeValue>
      </saml:Attribute>
    </listOfAttributes>
    <listOfUsernames>
        <username>nanohub</username>
    </listOfUsernames>
  </entry>
  <entry>
    <listOfAttributes>
      <!-- FriendlyName="isMemberOf" -->
      <saml:Attribute
        AttributeName="urn:oid:1.3.6.1.4.1.5923.1.5.1.1"
        AttributeNamespace="urn:mace:shibboleth:1.0:attributeNamespace:uri">
        <saml:AttributeValue>http://www.gisolve.org</saml:AttributeValue>
      </saml:Attribute>
    </listOfAttributes>
    <listOfUsernames>
        <username>gisolve</username>
    </listOfUsernames>
  </entry>
</attributePolicy>
```

Appendix D

This policy file recognizes three roles, at least one of which must be satisfied:

```
<attributePolicy
  xmlns="http://gridshib.globus.org/2005/08/policy"
  xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion">
  <entry>
    <listOfAttributes>
      <saml:Attribute
        AttributeName="urn:mace:dir:attribute-def:eduPersonEntitlement"
        AttributeNamespace="urn:mace:shibboleth:1.0:attributeNamespace:uri">
        <saml:AttributeValue
          >http://www.teragrid.org/names/roles/admin</saml:AttributeValue>
      </saml:Attribute>
    </listOfAttributes>
    <listOfUsernames>
        <username>tg-admin</username>
    </listOfUsernames>
  </entry>
  <entry>
    <listOfAttributes>
      <saml:Attribute
        AttributeName="urn:mace:dir:attribute-def:eduPersonEntitlement"
        AttributeNamespace="urn:mace:shibboleth:1.0:attributeNamespace:uri">
        <saml:AttributeValue
          >http://www.teragrid.org/names/roles/user</saml:AttributeValue>
      </saml:Attribute>
    </listOfAttributes>
    <listOfUsernames>
        <username>tg-user</username>
    </listOfUsernames>
  </entry>
  <entry>
    <listOfAttributes>
      <saml:Attribute
        AttributeName="urn:mace:dir:attribute-def:eduPersonEntitlement"
        AttributeNamespace="urn:mace:shibboleth:1.0:attributeNamespace:uri">
        <saml:AttributeValue
          >http://www.teragrid.org/names/roles/guest</saml:AttributeValue>
      </saml:Attribute>
    </listOfAttributes>
    <listOfUsernames>
        <username>tg-guest</username>
    </listOfUsernames>
  </entry>
</attributePolicy>
```