BMBF-Projekt

„Nutzung von kurzlebigen Zertifikaten in portalbasierten Grids (GapSLC)“

– Förderkennzeichen 01IG09003 –

„Service Grids für Forschung und Entwicklung“
des Bundesministeriums für Bildung und Forschung (BMBF)

Task 3: Deliverable D3

Arbeitspaket:	Task 3
Autor(en):	Stefan Pinkernell (AWI)
Version:	1.0
Publikationsdatum:	20.07.2010
Kontakt:	Stefan Pinkernell
Email:	Stefan.Pinkernell@awi.de

GEFÖRDERT VOM

Bundesministerium
für Bildung
und Forschung

Inhalt

1 Ausgangslage

Die Weiterentwicklung der Autorisierungsverfahren in den meisten Grid-Middleware-Systemen (Globus, gLite, UNICORE) konvergiert auf SAML als Medium der Autorisierungsinformation. Entsprechende Entwicklungsarbeiten, wie z.B. GridShib für das Globus Toolkit haben erste Ergebnisse geliefert.

Autorisierungsentscheidungen sollen in vielen Communities im D-Grid bei den Anbietern von Ressourcen getroffen werden. Dazu werden Autorisierungsinformationen durch ein VO-Managementsystem (VO-Attribute) bereitgestellt. Weitere Autorisierungsmerkmale, wie vor allem der identifizierende Name, stammen vom Identity Provider der Heimateinrichtung (Campus-Attribute). Bei durchgängigem Einsatz von SAML als Medium für die Autorisierungsinformationen werden diese idealerweise den Ressourcenanbietern als (in einem Proxy-Zertifikat eingebettete) SAML-Zusicherungen übermittelt. Diese bestehen aus einer SAML-Assertion mit den Campus-Attributen, die vom Campus-IdP erzeugt und von ihm signiert wurde, sowie einer SAML-Assertion mit den VO-Attributen, die aus dem VO-Managementsystem stammt und von ihm signiert wurde.

Damit könnte der Vertrauenskontext jeder SAML Assertion über den ganzen Verlauf der Entscheidungskette intakt gehalten werden. Diese Ideallösung ist derzeit jedoch nicht umsetzbar. Die verfügbare Version von GridShib for Globus Toolkit unterstützt nur die Auswertung genau einer, in ein X.509 Zertifikat eingebetteten, SAML-Assertion. Damit kann entweder nur die SAML-Assertion mit den Campus-Attributen oder nur diejenige mit den VO-Attributen in das Grid zu den Ressourcenanbietern transportiert werden.

Die GridShib-Entwickler haben diese Funktionslücke erkannt. Es gibt jedoch keine Planung die fehlende Funktionalität, die Interpretation von mehr als einer SAML Assertion in einem Zertifikat, zu implementieren, da GridShib über TeraGrid finanziert wird und dort diese Erweiterung derzeit nicht benötigt wird.

Auf der Basis verfügbarer Software-Komponenten soll für diese Lösung das Trust-Proxy-Verfahren zur Anwendung kommen. Damit übernimmt das Portal die Funktion einer Attributautorität, in dem es die Autorisierungsinformationen aus den verschiedenen Quellen zusammen fasst und eine neue, selbst signierte SAML Assertion ausstellt, die sowohl Campus- als auch VO-Attribute enthalten kann. Bei diesem Verfahren muss besonders sichergestellt werden, dass die Integrität der Vertrauensbeziehungen für alle Parteien nachvollziehbar bleibt.

Die verfügbaren Software-Komponenten sind die GridShib SAML Tools zur Einbettung einer SAML Assertion in ein X.509 Zertifikat und GridShib for Globus Toolkit zur Extraktion und Interpretation einer SAML Assertion aus einem X.509 Zertifikat.

Die Arbeiten zu diesem Task umfassen daher die Weiterentwicklung der Gridshib SAML Tools, um die technische Möglichkeit, eine am Portal zusammengefasste und signierte Assertion auswerten zu können, zu realisieren.

Ferner soll das in Task 1 entwickelte Tool zur Nutzung von kurzlebigen Zertifikaten erweitert werden, um SAML Assertions von unterschiedlichen Quellen zu laden und eine neue Assertion mit allen Informationen auszustellen. Diese Arbeiten sind auch schon als Vorarbeiten für Task 4 anzusehen, bei dem dann ebenfalls die Assertion von einem SAML VOMS mit den VO Attributen dazukommt.

2 Grundlagen

2.1 SAML Assertions

SAML ist ein ab 2001 von OASIS entwickeltes XML Framework zur Übertragung von Informationen zur Authentifizierung und Autorisierung [1]. Hauptanwendungsgebiet neben Single-Sign-On Systemen sind Autorisierungsdienste.

In der SAML Spezifikation sind SAML-Assertions, SAML-Protokoll, SAML-Bindings und SAML-Profile definiert. Für diesen Task wird nur ein Teil der Spezifikation genutzt: SAML Assertions, in denen die Attribute enthalten sind.

Es existieren zwei Haupt-Versionen: SAML 1 [2][3] und SAML 2 [4][5]. Die Version 2 setzt sich mittlerweile immer mehr durch. Eine wachsende Anzahl von Identity Providern fährt diese Version und stellt damit die Campusattribute aus der Shibboleth-Umgebung zur Verfügung. Jedoch gibt es daneben auch noch einige Installationen von SAML 1 und insbesondere die verwendete Version von Gridshib for Globus Toolkit benötigt noch SAML Assertions Version 1.

2.2 OpenSaml

OpenSAML ist eine Bibliothek für die Sprachen Java und C++, um SAML zu verarbeiten [6]. Es existieren mehrere Entwicklungszweige dieser Bibliothek: OpenSaml Version 1 ist nur in der Lage, SAML 1 Assertions zu verarbeiten. Das Projekt Gridshib verwendet eine modifizierte Version von OpenSAML 1, die von den Gridshib-Entwicklern speziell angepasst wurde. Erst mit der Version 2 von OpenSAML, bei der es sich um eine komplette Neuentwicklung handelt, ist es möglich, sowohl SAML 1 als auch SAML 2 Assertions zu verarbeiten.

Für die Arbeiten an diesem Task wurde OpenSaml 2.3.1 in der Java Version benutzt, die sowohl die Informationen aus SAML 1 Assertions extrahieren als auch eine neue SAML 2 Assertion erstellen kann.

2.3 Shibboleth

Shibboleth bietet ein Verfahren zur verteilten Authentifizierung und Autorisierung auf Basis von SAML im Umfeld der Webservices und Webanwendungen [7]. Bei diesem Single-Sign-On-Verfahren muss sich der Nutzer zu Beginn der Session bei seiner Heimateinrichtung anmelden und kann dann ohne erneute Anmeldung auf teilnehmende Dienste, auch anderer Anbieter, zugreifen.

Ein Shibboleth-System besteht dabei aus drei unabhängigen Teilen: Die Heimateinrichtungen stellen zumindest einen Identity Provider bereit, an dem sich die Nutzer anmelden können. Anbieter von Inhalten nutzen den Service Provider, um eine Autorisierungsentscheidung für ihre Dienste zu treffen zu können. Zudem existiert ein Discovery Service (WAYF – Where are you from), der den noch nicht authentifizierten Nutzer zur Anmeldung an seine Heimateinrichtung weiterleitet.

2.4 Gridshib for Globus Toolkit

Um SAML basierte Autorisierung im Globus Toolkit zu ermöglichen, wird die Software Gridshib for Globus Toolkit eingesetzt. Es handelt es sich dabei um ein Plug-In für die Middleware Globus Toolkit [8][9]. Dieses Plug-In ermöglicht die Nutzung von SAML mit X.509 Zertifikaten und erweitert die Grid Security Infrastructure (GSI) damit um die Nutzung von Attributen zur attributbasierten Autorisierung [11].

Die Gridshib SAML Tools bilden einen weiteren Teil dieser Software [12]. Als stand-alone Software eingesetzt werden diverse Tools bereitstellt, wie z.B. ein Tool um SAML Assertions auszustellen, ein Binding Tool um SAML Assertions an Proxy Zertifikate zu binden, etc. Teile der Gridshib SAML Tools werden auch als Java-Bibliothek von Gridshib for Globus Toolkit verwendet und mussten im Rahmen dieses Tasks erweitert werden.

2.4.1 Auswertung durch Globus + Gridshib for Globus Toolkit

Bei der Auswertung der SAML Assertions kommen verschiedene Komponenten zum Einsatz. Es besteht die Möglichkeit, mehrere Policy Information Points (PIP), an denen Informationen gesammelt werden, sowie Policy Decision Points Points (PDP) zu konfigurieren, an denen die Autorisierungsentscheidungen getroffen werden. In Abbildung 1 ist die Kette der PIP und PDP von Gridshib dargestellt. Interessant für diesen Task ist speziell der SAMLPDP, an dem Informationen aus einer an das verwendete Proxy-Zertifikat gebundene SAML Assertion ausgewertet wird [10].

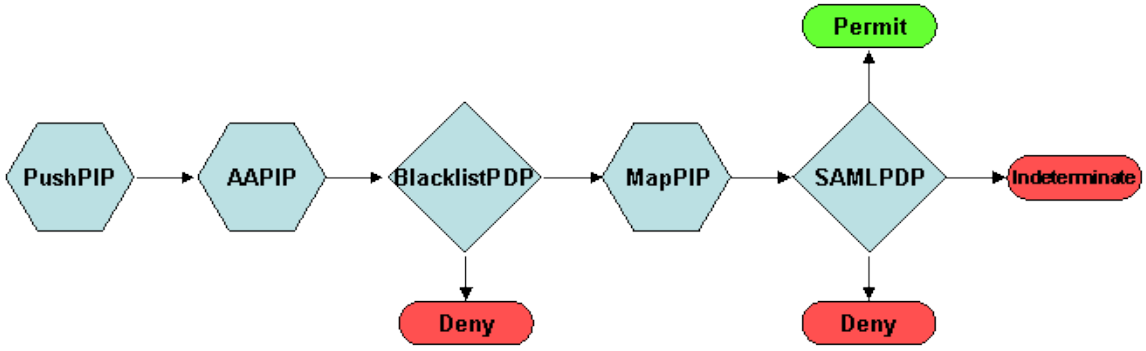

Abbildung 1: Gridshib Policy Decision Point [10]

3 Erstellung einer SAML Assertion

Im Rahmen von Task 1 wurde ein Java Servlet erstellt, das den Bezug von kurzlebigen Zertifikaten in einem Grid-Portal-Szenario implementiert. Dieses Servlet wurde für diesen Task stark erweitert und ist nun zusätzlich in der Lage, Campus Attribute vom Identity Provider aus einer Shibboleth Infrastruktur abzuholen und eine neue SAML Assertion auszustellen. Dies ist zwar nicht direkt Teil dieses Tasks, allerdings wird die SAML Assertion wird für die Entwicklung und Tests der Auswertungssoftware benötigt. Zudem sind diese Entwicklungen auch schon als Vorarbeiten zu Task 4 anzusehen, bei dem zusätzlich zur Assertion vom IdP mit den Campus Attributen auch noch eine Assertion vom SAML VOMS hinzukommt. Bei der Entwicklung wurde daher darauf geachtet, spätere weitere Assertions einfach mit aufzunehmen zu können.

Die folgende Liste zeigt die wesentlichen Punkte, die vom Servlet abgearbeitet werden.

(1) Für den Aufruf des Servlets ist eine Shibboleth-Authentifizierung notwendig. Der Nutzer wird daher zunächst zum „Where are you from“-Server und dann zum Identity-Provider der Heimateinrichung weitergeleitet. Der erfolgreich authentifizierte Nutzer wird dann wieder an das Portal zurückgeleitet.
(2) Am Portal findet sich lediglich ein Button, mit dem das Portal Delegation Verfahren gestartet wird. Dabei wird der Nutzer zu Web-Seiten des DFN weitergeleitet, was jeweils Nutzerinteraktion in Form einiger Klicks bedeutet.

(3) Schließlich gelangt der Nutzer zurück ans Portal, wo im Hintergrund vom Nutzer weitgehend unsichtbar, einige weitere Schritte ablaufen . Zu diesem Zeitpunkt liegt das fertige kurzlebige Zertifikat aus Schritt 2 schon am Portal vor.
   a. Eine Assertion mit Campus Attributen des in Schritt 1 authentifizierten Nutzers wird geladen. Die Attribute werden ausgelesen und in internen Datenstrukturen zwischengespeichert.
   b. Eine neue, portal-signierte SAML Assertion wird erstellt, die alle gesammelten Attribute aus Schritt 3a enthält.
   c. Ein Proxy-Zertifikat wird vom in Schritt 2 bezogenen kurzlebigen Zertifikat abgeleitet, in das die neue SAML Assertion aus Schritt 3b integriert wird.

(4) Eine Übersichtsseite mit Informationen zum Kurzlebigen- und Proxy- Zertifikat, sowie zur eingebetteten SAML Assertion wird angezeigt. Je nach Implementierung wird dieses Proxy-Zertifikat im Dateisystem des Servers abgelegt.

3.1 Implementierung

In der Klasse CampusAssertionT ist das Demo-Servlet implementiert. Einige Teile, die von mehreren unterschiedlichen Servlet Varianten genutzt werden, wie beispielsweise die Erzeugung eines Schlüsselpaares, eines PKCS-10 Requests [14], etc., sind in die Klasse ServletHelper ausgelagert.

Im ersten Schritt wird durch dieses Servlet der Portal Delegation Prozess in Gang gesetzt, bei dem zunächst ein Schlüsselpaar, bestehend aus einem privaten und einem öffentlichen Schlüssel, erzeugt wird [15]. Der öffentliche Schlüssel wird benutzt, um einen PKCS-10 Zertifikat Request zu erstellen, welcher direkt an den DFN-SLCS gesendet wird [16]. Im Folgenden ist eine kurze Interaktion mit dem Nutzer notwendig, der ein paar Bestätigungs-Klicks tätigen muss. Nach einigen Augenblicken wird das kurzlebige Zertifikat an das Servlet gesendet. Dieser Prozess ist auch noch einmal detailliert unter [13] beschrieben.

Im Unterpaket tools.saml2 finden sich die beiden Klassen Saml1 und Saml2, die dazu dienen, SAML Assertions der jeweiligen Version zu verarbeiten. Beide Klassen sind dazu in der Lage, eine SAML Assertion sowohl ‚from scratch' zu erstellen, als auch eine bestehende Assertion zu parsen und den Unmarshall-Prozess durchzuführen. Zusätzlich gibt es Methoden, um das interne SAML Objekt wieder in eine XML Datei umzuwandeln, Attribute und sonstige Informationen direkt zu extrahieren, etc.

Von der Klasse Saml2 wird Gebrauch gemacht, um die vom Identity Provider stammende SAML2 Assertion auszuwerten. Die Attribute werden mit Hilfe der Klasse SAMLAttributes, die ebenfalls Teil dieses Pakets ist, zwischengespeichert. Mit Hilfe der Klasse Saml1 wird dann eine neue SAML Assertion erstellt, die alle zuvor gesammelten Attribute aus der Klasse SAMLAttributes enthält. Diese neue SAML Assertion wird mit Hilfe des Portal Zertifikats signiert und soll am Globus Container die Attribute zur Autorisierung dienen.

Geholt und ausgewertet wird die Assertion mit den Campus Attributen von der Klasse CampusAssertionRetriever. Dieser Prozess wird angestoßen, direkt nachdem das kurzlebige Zertifikat vom DFN-SLCS an das Servlet zurückgesendet worden ist.

Bisher wird nur die Campus Assertion als Quelle für Attribute genutzt. Es ist geplant, später ebenfalls eine Assertion von einem SAML VOMS zu beziehen und diese Attribute hinzuzufügen [17]. Die

Klassen SAMLAttributes zum Zwischenspeichern und Saml2 zum auswerten sind bereits so ausgelegt, auch unterschiedliche Quellen bedienen zu können. Es wäre noch notwendig analog zum CampusAssertionRetriever ein Modul zu entwickeln, dass die Assertion vom SAML VOMS holt.

Sind alle Attribute eingesammelt wird die neue SAML1 Assertion erstellt und signiert. Dafür zuständig ist die Klasse Saml1. Zuletzt wird mit Hilfe der Klasse SamlProxy ein Proxy-Zertifikat erstellt und die SAML Assertion an dieses gebunden.

In dieser Variante wird nur das das Proxy Zertifikat mit eingebetteter SAML Assertion auf der Festplatte gespeichert. Der Ordner muss über einen entsprechenden Eintrag in einer Properties Datei konfiguriert werden. Das kurzlebige Zertifikat sowie der private Schlüssel werden für die Dauer der Session im Arbeitsspeicher gehalten aber nicht auf der Festplatte abgelegt. Damit steht dieses Verfahren nicht in Konflikt mit den Anforderungen zur Speicherung von Schlüsselmaterial in den entsprechenden Policies [19].

4 Erweiterung der Gridshib SAML Tools

Gridshib unterscheidet zwar konzeptionell zwischen dem Aussteller des Zertifikats und dem Aussteller der SAML Assertion, jedoch haben die Entwickler bisher nur solche Anwendungsfälle vorgesehen, bei denen beide identisch sind und die Signatur des Zertifikats auch gleichzeitig die Assertions einschließt. Dritte Akteure kommen in ihren use cases nicht vor [18].

Dies deckt sich leider nicht mit unserem Anwendungsfall, wo das Zertifikat vom DFN-Verein ausgestellt und signiert wird und erst durch das Portal eine eigene Assertion eingefügt wird. Um trotzdem eine Vertrauensbasis für die zugesicherten Attribute zu gewährleisten, muss also auch diese eingefügte Assertion nochmals signiert werden.

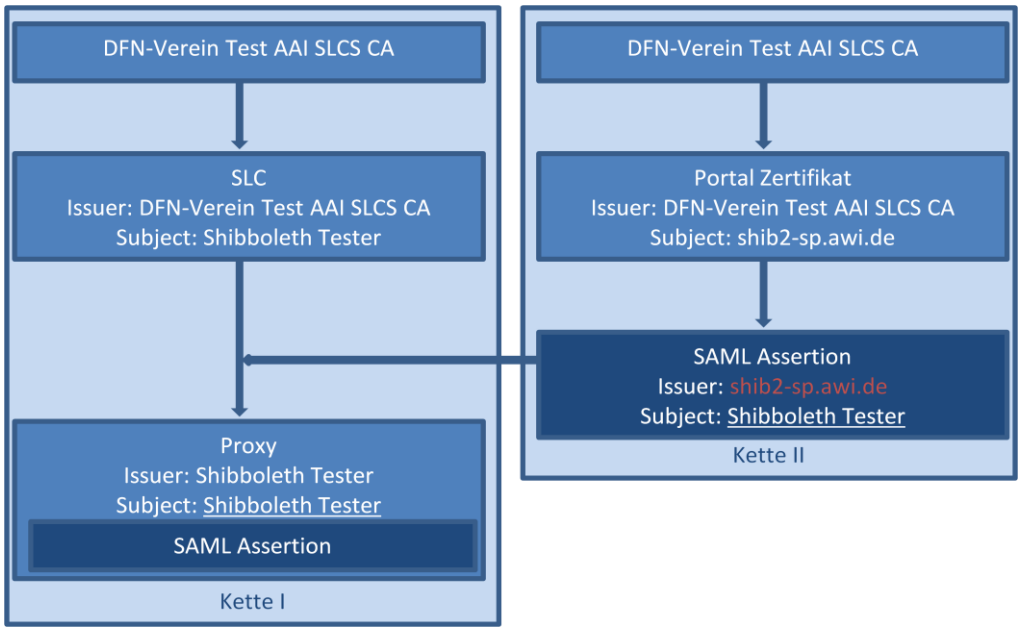

Abbildung 2: Beteiligte Zertifikatketten

Die am Portal erzeugte SAML Assertion muss eine eigene Signatur enthalten, die allerdings auf eine andere Zertifikatkette zurückgeht als die des umgebenden X.509 Zertifikats (vgl. Abbildung 2). Dieser Anwendungsfall ist von den Gridshib Entwicklern leider nicht vorgesehen, so dass eine solche Assertion von der Originalversion der Software nicht für die Auswertung verwendet sondern einfach ignoriert würde.

Daher wurde im Rahmen dieses Task eine Bibliothek modifiziert, die dazu dient, zu entscheiden, ob eine Assertion zur Auswertung genutzt werden darf. Dazu werden zunächst einige Checks durchgeführt, bevor die Assertion akzeptiert wird:

- Syntaxcheck der SAML-Assertion
- Überprüfung der Signatur der SAML Assertion
- Diverse Checks mit Hilfe der Klasse ProxyPathValidator: Extension Check, Validity Check, Restricted Policy Check, CRL Check, etc.
- Test, ob Issuer der Assertion in der trustedAuthoritiesList eingetragen ist
- Test, ob SAML Assertion Subject und Proxy Subject identisch sind

Ergebnis ist eine modifizierte Java Bibliothek, die aus einer einzigen jar-Datei besteht. Um die Erweiterungen nutzen zu können muss am Server nur diese eine Datei ausgetauscht werden. Eine Anleitung dazu, Konfigurations-Einstellungen für den Server, sowie Tests der Software finden sich in Kapiteln 5.

4.1 Implementierung

Die Klasse SAMLUtil aus den Gridshib-SAML-Tools wird vom Attribute Acceptance Policy Information Point (AAPIP, vgl. Abbildung 1) genutzt, um die in das Proxy-Zertifikat eingebettete SAML-Assertion zu validieren und muss für unseren Zweck ein wenig modifiziert werden, um die durch die Zertifikat-Kette II signierte und in das Proxy-Zertifikat eingebettete SAML-Assertion ebenfalls auswerten zu können. Dabei wird zunächst geprüft, ob die Assertion auch wirklich für das Proxy-Zertifikat ausgestellt wurde, in das sie eingebettet ist. Dazu wird ein einfacher String-Vergleich des Inhalts des Subject-Feldes der Assertion mit dem Inhalt des Subject-Feldes des umgebenden Proxy-Zertifikates durchgeführt.

Für die Überprüfung der Signatur der SAML-Assertion wird die Klasse ProxyPathValidator aus dem Globus Toolkit verwendet. Diese Klasse dient ursprünglich dazu, die Unterschriftenkette eines Proxy-Zertifikates zu überprüfen. Dazu wird auf die Standard-Ablageorte für Root-Zertifikate zugegriffen und versucht, die Zertifikatkette des Proxy-Zertifikats bis zu einem Zertifikat aus diesem Ordner zurückzuverfolgen.

Die Funktionalität dieser Klasse kann genutzt werden, um das Zertifikat des Ausstellers der SAML-Assertion (in Abbildung 2 ist dies das Portal-Zertifikat) zu überprüfen. Dazu ist es nicht notwendig, diese Klasse zu erweitern. Sie kann direkt von der Klasse SAMLUtil aus verwendet werden.

Ist all dies gegeben wird in der Klasse AAPIPImpl überprüft, ob der Aussteller der SAML-Assertion überhaupt vertrauenswürdig ist. Dazu existiert im Verzeichnis /etc/metadata die Datei trustedAutoritiesList, in der alle zulässigen Aussteller von SAML-Assertions aufgelistet sind. All dies setzt natürlich voraus, dass der entsprechende DN des umgebenden Proxy-Zertifikats auch im grid-mapfile eingetragen ist.

In der Originalversion gab die Methode SAMLUtil::isAssertionValid(…)bei Verwendung signierter SAML-Assertions immer „false“ zurück. Nun werden zunächst einige Tests durchgeführt um dann je nach Ergebnis „true“, um die Assertion zu akzeptieren, oder „false“, um die Assertion abzuweisen, zurückzugeben.

All diese Überprüfungen werden nur in den Methoden consumeSAMLAssertion und isAssertionValid) durchgeführt. Insgesamt wurde nur der Code einer einzelnen Klasse (SAMLUtil) verändert. Dieser modifizierte Code steht unter http://aforge.awi.de/gf/project/gapslc/frs/ zum Download bereit. Um die Änderungen zu installieren, muss nur die Bibliothek mit dem geänderten Code ausgetauscht und der Globus-Container neu gestartet werden (vgl. Kap. 5.3).

5 Integration in Testumgebung mit Globus Toolkit 4.0.8

Dieses Kapitel beschreibt, wie die Software installiert und benutzt werden kann. Als Testsystem standen dazu zwei Linux-Rechner bereit, die jeweils unter Ubuntu Linux Version 8.04 LTS liefen.

Auf dem Rechner, der das Portal simuliert, lief das Servlet in einem Apache Tomcat Version 5.5 in Verbindung mit dem Apache Webserver Version 2.2.1. Zu beachten ist zudem, dass für die Nutzung des DFN-SLCS der Portal-Rechner beim DFN bekannt gemacht werden muss. Um Grid-Jobs abschicken zu können wurde Globus ws-core Version 4.0.5 verwendet.

Auf dem zweiten Rechner läuft das Globus Toolkit Version 4.0.8 und die Gridshib for Globus Toolkit Version 0.6.1. Mit Gridshib for Globus Toolkit wurde hier eine Attribut-basierte Autorisierung konfiguriert.

5.1 Installation des erweiterten Portal Delegation Servlets

Die Installation des Servlets ist sehr einfach durchzuführen. Benötigt wird nur ein geeigneter Application Server, z.B. Tomcat, und ein Apache Webserver.

Zu beachten ist, dass das Servlet schon per Shibboleth geschützt sein muss. Im Unterverzeichnis sites-enabled der Apache Konfiguration muss die Shibbolisierung der Site eingetragen werden:

```
<Location "/slc">
   AuthType shibboleth
   ShibRequireSession On
   ShibRequireAll On
   ShibRedirectToSSL 443
   require valid-user
   ShibExportAssertion On
</Location>
```

Um Tomcat zusammen mit dem Apache Webserver verwenden zu können wurde mod_jk benutzt. Im Unterverzeichnis conf.d muss daher in der Datei mod-jk.conf folgender Eintrag hinzugefügt werden:

```
JkMount /slc* ajp13_worker
JkMount /slc ajp13_worker
```

Wird die war-Datei, in der die Software u.a. vorliegt, in den webapps-Ordner von Tomcat kopiert wird automatisch ein Autodeploy-Prozess gestartet. Konfiguriert wird das Servlet über eine

Properties-Datei, die unter /usr/local/etc/slc.properties vorliegen muss. In dieser Datei muss der Speicherort für die Proxy-Zertifikate, die URL des DFN SLCS, die eigene Portal-URL, der Pfad zum Portal-Zertifikat und –Schlüssel, sowie der Name der Campus-Assertion vom IdP konfiguriert werden. Eine Beispiel Properties Datei liegt dem Source Code bei.

Das Servlet dient lediglich zur Demonstrationszwecken. Um die Software in einem Portal zu integriert zu nutzen, sollte nicht das Servlet sondern ein Portlet verwendet werden. Dazu bietet sich der Standard JSR-168, bzw. der neuere JSR-286 an. Eine Portierung und Anpassung an das eigene Portal ist nicht sehr aufwendig. Statt der doGet bzw. doPost Methoden kann z.B. die Methode doView aufgerufen werden um das Portlet zu starten. Zudem wird die Methode processAction genutzt, wenn das kurzlebige Zertifikat vom DFN-SLCS zurück kommt.

5.2 Bezug eines Credentials

Das Credential zum ausführen eines Grid-Jobs kann mit Hilfe des Servlets bezogen werden, wobei zur Bedienung nur ein Browser notwendig ist. Das Demo-Servlet ist dabei unter der in der Properties Datei konfigurierten Domain unter dem Namen /slc zu erreichen(z.B. https://shib2-sp.awi.de/slc). Das Servlet speichert das Proxy-Zertifikat mit der eingebetteten SAML Assertion in dem dafür konfigurierten Ordner ab (vgl. Kap. 5.1). Eine detaillierte Anleitung zur Vorgängerversion findet sich zudem auch unter [13].

Um das Proxy Zertifikat verwenden zu können ist es u.U. zudem noch notwendig, die Berechtigungen anzupassen. Bei unseren Tests wurden die Unix-Dateirechte auf 400 gesetzt, um das Programm globusrun-ws benutzen zu können. Dies wurde auf der Konsole durchgeführt, nicht mit Hilfe des Servlets.

5.3 Installation der modifizierten Saml Tools Bibiothek

Die Integration der modifizierten Bibliothek in ein installiertes Globus Toolkit 4.0.8 unter Verwendung von Gridshib for Globus Toolkit 0.6 ist ebenfalls sehr einfach durchzuführen.

Die modifizierte Klasse org.globus.gridshib.security.util.SAMLUtil ist in den Gridshib-SAML-Tools enthalten. Mit Hilfe des ebenfalls in diesem Paket enthaltenen Ant-Scripts können die SAML-Tools übersetzt werden, wobei die Software auf mehrere Jar-Dateien aufgeteilt wird. Die von uns benötigte modifizierte Klasse ist in der Bibliothek **gridshib-common-0_5_0.jar** enthalten.

Bei der Installation von Gridshib for Globus Toolkit werden einige Bibliotheken benötigt, darunter auch die Datei gridshib-common-0_5_0.jar. Diese Datei ist schon im Installationspaket von Gridshib enthalten.

Um die Änderungen im Globus-Container nutzen zu können, muss die modifizierte Version der Klasse SAMLUtil übersetzt und die entsprechende Jar-Datei erstellt werden. Dafür kann das Ant-Script genutzt werden. Dabei wird im Projekt-Unterverzeichnis ‘lib‘ die Bibliothek gridshib-common-0_5_0.jar angelegt. Diese muss in das ‘lib‘-Verzeichnis des Globus-Containers kopiert werden. Damit die Änderungen wirksam werden, muss der Globus-Container neu gestartet werden.

Es genügt, die modifizierte Bibliothek gridshib-common.jar im Unterverzeichnis ‘lib‘ des Globus-Ordners auszutauschen und den Globus-Container anschließend neu zu starten. Zudem ist zu beachten, dass alle nötigen Wurzelzertifikate vorhanden sind.

5.4 Anpassung der Konfiguration des Globus Containers

Wird die modifizierte Bibliothek verwendet, stehen weiterhin alle durch Gridshib for Globus Toolkit bereitgestellten Features zur Verfügung, wie z.B. die Möglichkeit Blacklists zu erstellen, auf Basis der enthaltenen Attribute eine Autorisierung sowie das Mapping auf lokale Nutzerkonten vorzunehmen, etc. Details dazu und weitere Hinweise zur Konfiguration der PIPs und PDPs finden sich in der Dokumentation zu Gridshib for Globus Toolkit.

Beispielkonfiguration der Attribute für Autorisierung
in Datei: /etc/grid-security/policy/global-authz-policy.xml

```
<AttributePolicy
   xmlns="http://gridshib.globus.org/namespaces/2005/08/policy"
   xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion"
   xmlns:saml1="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>urn:geant:dfn.de:dfn-pki:slcs</saml:AttributeValue>
    </saml:Attribute>
   </listOfAttributes>
 </entry>
</AttributePolicy>
```

Achtung: Die Attribute müssen genau so eingetragen werden, wie sie auch in der SAML Assertion vorliegen. Dabei ist auch der korrekte Namespace zu beachten!

Beispielkonfiguration der verwendeten Gridshib PDP und PIP
Ausschnitt aus Datei: /opt/globus-4.0.8/etc/globus_wsrf_core/ global_security_descriptor.xml

```
<authz value="global:org.globus.gridshib.SAMLAssertionPushPIP
           global:org.globus.gridshib.AttributeAcceptancePIP
           global:org.globus.gridshib.SAMLBlacklistPDP
           global:org.globus.gridshib.SAMLMapPIP
           global:org.globus.gridshib.SAMLAttributePDP"/>
```

Beispielkonfiguration der Attribute für Autorisierung
Ausschnitt aus Datei: /opt/globus-4.0.8/etc/globus_wsrf_core/ server-config.wsdd

```
<!-- GRIDSHIB CONFIGURATION, SCOPE: GLOBAL -->
<parameter name="global-metadataPath" value="/etc/grid-security/metadata"/>
<parameter name="global-attributeAuthzPolicyFile" value="/etc/grid-security/policy/global-authz-
policy.xml"/>
<parameter name="global-attributeMappingPolicyFile" value="/etc/grid-security/policy/global-
mapping-policy.xml"/>
<parameter name="global-blacklistPrincipalNamesFile" value="/etc/grid-security/blacklists/blacklist-
principal-names.txt"/>
<parameter name="global-blacklistIPAddressesFile" value="/etc/grid-security/blacklists/blacklist-ip-
addresses.txt"/>
<parameter name="global-consultDefaultGridmap" value="false"/>
<parameter name="global-enableGridmap" value="false"/>
```

Im Beispiel für die Datei global-authz-policy.xml ist dargestellt, wie ein Attribut zur Autorisierung konfiguriert werden kann. Es ist auch möglich, sowohl mehrere Werte für ein Attribut, als auch

mehrere Attribute anzugeben. In der Datei server-config.wsdd wird eingetragen, dass die Datei global-authz-policy.xml für die Autorisierung verwendet werden soll. Dieses Beispiel zeigt nur die globale Konfiguration des Servers. Einzelne Teile, wie z.B. gram, rft, etc. können natürlich auf separat konfiguriert werden.

Werden in einer Datei mehrere Attribute bzw. mehrere Werte für ein Attribut angegeben, so wird „PERMIT“ zurückgegeben, sobald ein Attribut vorhanden ist. Dies entspricht somit einer logischen ODER Entscheidung. Wenn auf zwei oder mehr Attribute gestestet werden soll, die alle vorhanden sein sollen (logisches UND), so müssen diese auf mehrere Dateien verteilt werden. Dazu wird im Security Descriptor ein weiterer SAMLAttributePDP eingetragen werden und in der Datei server.config ein zweiter attributeAuthzPolicyFile-Eintrag.

5.5 Test der Software

Zunächst wurde mit dem in Kapitel 3 vorgestellten Servlet das notwendige Credential zum Ausführen von Grid-Jobs erzeugt. Dazu wurde vom Servlet zunächst das kurzlebige Zertifikat und die SAML-Assertion mit den Campus Attributen bezogen, eine neue, durch das Portal signierte SAML-Assertion mit Campus Attributen und schließlich ein Proxy-Zertifikat mit eingebetteter SAML-Assertion erstellt (vgl. Kap. 5.1 und 5.2).

Dieses Proxy-Zertifikat wurde genutzt, mit dem Programm globusrun-ws einen Grid-Job auf einem Testrechner mit Globus und der (modifizierten) Gridshib for Globus Toolkit Version aufzuführen. Dazu war es im Testsystem notwendig, auf dem Rechner, von dem aus der Job abgeschickt wurde, die Variablen X509_USER_CERT, X509_USER_KEY und X509_USER_PROXY alle auf das Proxy-Zertifikat zeigen zu lassen, das zudem die Unix-Rechte 400 haben sollte.

5.5.1 Verwendung Originalvariante der Bibliothek

Wird die modifizierte Bibliothek nicht verwendet, kann die eingebettete SAML Assertion nicht ausgewertet werden, so dass die enthaltenen Attribute nicht zur Autorisierung herangezogen werden. Der abgeschickte Grid-Job wird daher nicht ausgeführt.

```
$/opt/globus/globus-4.0.8/bin/globusrun-ws -submit -s -F gs4gt5.awi.de:8443 -c /bin/date
Delegating user credentials...Done.
Submitting job...Failed.
Cleaning up any delegated credentials...Done.
globusrun-ws: Error submitting job
globus_soap_message_module: SOAP Fault
Fault code: soapenv:Server.userException
Fault string: org.globus.wsrf.impl.security.authorization.exceptions.AuthorizationException:
"/C=DE/O=DFN-Verein/OU=DFN-PKI/OU=SLCS/OU=Alfred-Wegener-Institut/CN=Stefan Pinkernell -
Stefan.Pinkernell@awi.de" is not authorized to use operation:
{http://www.globus.org/namespaces/2004/10/gram/job}createManagedJob on this service
```

5.5.2 Mit der modifizierten Bibliothek

Bei Verwendung der modifizierten Bibliothek, werden auch die Attribute der durch das Portal signierten SAML Assertions in die Autorisierungsentscheidung mit aufgenommen. Befinden sich die entsprechenden Attribute in der SAML Assertion und ist das Proxy-Zertifikat gültig und vertrauenswürdig, so wird der per globusrun-ws gestartete Job ausgeführt (vgl. Kap. 5.3 und 5.4).

```
$ /opt/globus/globus-4.0.8/bin/globusrun-ws -submit -s -F gs4gt5.awi.de:8443 -c /bin/date
Delegating user credentials...Done.
Submitting job...Done.
Job ID: uuid:0623816e-7879-11df-a92a-000c292b9aa5
Termination time: 06/16/2010 12:25 GMT
Current job state: Active
Current job state: CleanUp-Hold
Tue Jun 15 14:25:06 CEST 2010
Current job state: CleanUp
Current job state: Done
Destroying job...Done.
Cleaning up any delegated credentials...Done.
```

5.5.3 Attribute zur Autorisierung nicht vorhanden

In diesem Fall wurde ein weiterer SAMLAttributePDP, mit einem Attribut, das nicht in der Assertion vorhanden ist, am Container konfiguriert. Die Ausführung des Grid-Jobs wird somit nicht zugelassen.

```
$ /opt/globus/globus-4.0.8/bin/globusrun-ws -submit -s -F gs4gt5.awi.de:8443 -c /bin/date
Delegating user credentials...Done.
Submitting job...Failed.
Cleaning up any delegated credentials...Done.
globusrun-ws: Error submitting job
globus_soap_message_module: SOAP Fault
Fault code: soapenv:Server.userException
Fault string: org.globus.wsrf.impl.security.authorization.exceptions.AuthorizationException:
"/C=DE/O=DFN-Verein/OU=DFN-PKI/OU=SLCS/OU=Alfred-Wegener-Institut/CN=Stefan Pinkernell -
Stefan.Pinkernell@awi.de" is not authorized to use operation:
{http://www.globus.org/namespaces/2004/10/gram/job}createManagedJob on this service
```

5.5.4 Ungültige SAML Assertion

Wird eine SAML Assertion mit einer Unterschrift eines nicht verifizierbaren Portal-Zertifikats benutzt, so ist auf der Client-Seite folgende Fehlermeldung zu sehen:

```
globusrun-ws: Error submitting job
globus_soap_message_module: SOAP Fault
Fault code: soapenv:Server.userException
Fault string: java.rmi.RemoteException: Job creation failed.; nested exception is:
      java.lang.Exception: Unable to retrieve a username mapping for authenticated user
/C=DE/O=DFN-Verein/OU=DFN-PKI/OU=SLCS/OU=Alfred-Wegener-Institut/CN=Stefan Pinkernell –
Stefan.Pinkernell@awi.de.
```

In der Log-Datei var/container.log des Globus-Containers findet sich folgende Fehlermeldung, die auf eine fehlerhafte Signatur schließen lässt:

```
16.11.2009 16:11:20 org.globus.gridshib.security.util.SAMLUtil isAssertionValid
SCHWERWIEGEND: null
. Caused by COM.claymoresystems.cert.CertificateVerifyException: The signature of
'C=DE,O=GridGermany,OU=Alfred-Wegener-Institut,CN=shib2-sp.awi.de' certificate does not match
its issuer
```

6 Zusammenfassung + Ausblick

Für diesen Task wurde das im Zuge von Task 1 entwickelte Servlet erweitert, um SAML Assertions verarbeiten zu können. Damit ist es ist nun möglich, eine SAML Assertion mit Campus Attributen vom Identity Provider zu beziehen und die Attribute aus dieser SAML Assertion zu extrahieren. Aus diesen gesammelten SAML-Attributen kann dann eine neue, vom Portal signierte SAML Assertion erstellt werden.

Durch die Erweiterung der SAML Tools Bibliothek auf Seiten des Globus Containers ist es nun möglich, durch Dritte signierte SAML Assertions auszuwerten und die enthaltenen Attribute in die Autorisierungsentscheidung mit einfließen zu lassen.

Wie in Kapitel 5.5 dargestellt, ist durch diese Erweiterungen eine Autorisierung an anhand von Attributen in der SAML Assertion bereits möglich.

Natürlich könnte man, wenn nur eine Quell-SAML Assertion mit Attributen vorliegt, diese auch direkt in das Proxy Zertifikat einbetten und am Globus Container darauf zurückgreifen. Bei den hier vorgestellten Entwicklungen wurde aber schon darauf geachtet, in Zukunft auch weitere SAML Assertions als Quelle für weitere Attribute mit aufnehmen zu können. Damit wären die Entwicklungen auch schon als wichtige Vorarbeiten für Task 4 zu nennen. Geplant ist, Informationen zur virtuellen Organisation des Nutzers aus einer vom SAML VOMS stammenden SAML Assertion mit in die neu erstellte SAML Assertion mit aufzunehmen und diese ebenfalls zur Autorisierung zu nutzen.

7 Referenzen

[1] OASIS SAML, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security

[2] J. Hughes et al., Technical Overview of the OASIS Security Assertion Markup Language (SAML) V1.1. OASIS Committee Draft, May 2004. http://www.oasis-open.org/committees/download.php/6837/sstc-saml-tech-overview-1.1-cd.pdf

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

[4] N. Ragouzis et al., *Security Assertion Markup Language (SAML) V2.0 Technical Overview.* OASIS Committee Draft, March 2008. http://www.oasis-open.org/committees/download.php/27819/sstc-saml-tech-overview-2.0-cd-02.pdf

[5] S. Cantor et al. *Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0.* OASIS Standard, March 2005. http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf

[6] OpenSaml, https://spaces.internet2.edu/display/OpenSAML/Home/

[7] Shibboleth, http://shibboleth.internet2.edu/

[8] Gridshib, http://gridshib.globus.org/about.html

[9] Gridshib for Globus Toolkit, http://gridshib.globus.org/docs/gridshib-gt-0.6.1/readme.html

[10]Gridshib Quick Start, http://gridshib.globus.org/docs/gridshib/quick-start.html

[11]GSI, http://www.globus.org/security/overview.html

[12]Gridshib SAML Tools, http://gridshib.globus.org/docs/gridshib-saml-tools-0.5.0/readme.html

[13]GapSLC D1: Vereinfachung der Handhabung des kurzlebigen Zertifikats (SLC) für Nutzer, http://gap-slc.awi.de/documents/GapSLC-D1-V1.0.pdf

[14]RSA Laboratories, PKCS#10 v1.7:Certification Request Syntax Standard, Mai 2000, ftp://ftp.rsasecurity.com/pub/pkcs/pkcs-10/pkcs-10v1_7.pdf

[15]Portal Delegation, http://gridshib.globus.org/docs/gridshib-ca-0.5.1/portal-delegation.html

[16]DFN-SLCS, https://www.pki.dfn.de/index.php?id=slcs

[17]Benjamin Henne, Test des VOMS-SAML-Services des DGrid VOMS, DGI2 FG3.2, Juni 2009, http://dgi.d-grid.de/fileadmin/user_upload/documents/DGI2-FG3/FG3-2/DGI-2_FG-3.2_Test_VOMS-SAML-Service.pdf

[18]SAML in X.509 validation, http://dev.globus.org/wiki/SAML_in_X.509_Validation

[19]EuGridPMA, Protection of private key data for end-users in local and remote systems, http://www.eugridpma.org/guidelines/pkp/pk-protection-1.0-20091016.pdf