Gap-SLC

Nutzung von kurzlebigen Zertifikaten in portalbasierten Grids

VOMS SAML Service – prototypische Implementierungen

– Task 4, Deliverable D4 –

– Förderkennzeichen 01IG09003A-D –

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

Date	**02.02.2011**
Version	**1.0**
Type	**Draft**
Status	**In Progress**
Authors	**M. Haase, S. Pinkernell, B. Fritzsch**

Inhaltsverzeichnis

1 Einleitung

Die Autorisierung auf den Grid-Ressourcen soll unterschiedliche Anforderungen erfüllen. SAML-Attribute bieten hierbei eine gute Möglichkeit, Informationen über den Nutzer zu transportieren und darauf Autorisierungsentscheidungen zu gründen. Die Attribute können dabei aus unterschiedlichen Quellen stammen. Konkret wurde hier der Fall untersucht, dass einerseits der Identity Provider Attribute liefert. Da er in der Regel an der Heimateinrichtung des Nutzers angesiedelt und mit dem dortigen Identity-Management-System verknüpft ist, kann er Informationen über die Zugehörigkeit des Nutzers zu einer bestimmten Einrichtung liefern. Mit dem verteilten Arbeiten gewinnen aber neben den traditionellen Organisationseinrichtungen auch die Virtuellen Organisationen (VO) an Bedeutung. Daher liefert das VO Management andererseits wichtige Informationen über den Nutzer, die seine Zugehörigkeit zu einem Projekt repräsentiert. Im Task 4 wurde dieser Ansatz genutzt, um dies in einer Testumgebung prototypisch zu demonstrieren.

Das Dokument beschreibt zunächst die technischen Grundlagen und die aufgebaute Testumgebung. Die im Rahmen des Projekts entwickelten Module werden in ihrem Zusammenspiel erläutert und ihre Funktionsweise erklärt.

2 Überblick über die Komponenten

Dieser Abschnitt beschreibt die Systeme, die im VO-Management im GapSLC-Projekt betrachtet bzw. verwendet wurden. Zum Vergleich sei auf die Studie verwiesen, die im Vorprojekt IVOM erstellt wurde [IVOM].

2.1 VOMRS

Der Virtual Organization Membership Registration Service ist eine von FermiLab entwickelte Lösung zum Registrieren und Verwalten von Benutzern, Benutzerattributen, und Gruppeninformationen. VO-Administratoren bzw. deren Repräsentanten verwalten diese Informationen für jede VO, die einen VOMRS betreibt. VOMRS kann auch zur aktiven Benachrichtigung von Ressourcen-Administratoren über neue Benutzer verwendet werden, wird mit dieser Funktionalität jedoch in D-Grid nicht eingesetzt.

VOMRS besteht aus einer Server-Komponente, die verschiedene Interfaces anbietet (Web, SOAP, sowie Kommandozeile). Pro VO wird eine eigene Datenbank angelegt. Bei einer Registrierung authentifiziert sich ein Benutzer mit seinem Grid-Zertifikat, dessen CA in der Datenbank vorliegen muss. Nach Überprüfung der E-Mail-Adresse muss der Repräsentant, den der Benutzer ausgewählt hatte, für die Aufnahme des Benutzers in die VO und für das Setzen der passenden Rollen und Rechte sorgen.

2.2 VOMS

Im Gegensatz zum VOMRS wird der Virtual Organization Membership Service als zeitkritischer Dienst verwendet, der wie ein Login-Server arbeitet. VOMS kann auch Attributzertifikate ausstellen, welche von gLite benötigt werden. Ursprünglich verfügte der VOMS im Vergleich zum VOMRS nicht über dieselbe ausgereifte und benutzerfreundliche Funktionalität zur Benutzerregistrierung, weshalb in D-Grid für jede VO normalerweise eine Kombination vom VOMRS und VOMS eingerichtet wurde. VOMRS verfügt über verschiedene Möglichkeiten, seine Daten zu einem VOMS zu synchronisieren.

Mittlerweile hat der VOMS im Zuge seiner Entwicklung viele Funktionalitäten vom VOMRS, welcher nicht mehr weiterentwickelt wird, übernommen, so dass nun auch VO-Lösungen mit nur-VOMS vorzufinden sind. In D-Grid wird jedoch vorerst (Ende 2010) noch an der Verwendung des VOMRS festgehalten.

2.3 SAML VOMS

Bei der im vorigen Kapitel beschriebenen Version von VOMS-Admin können die VO-Rollen und Attribute des Nutzers nur in Form von Attribut-Zertifikaten bezogen werden. Über eine Erweiterung der Software stehen diese Informationen mittlerweile zusätzlich auch codiert als SAML Assertion zur Verfügung. Diese VOMS SAML Service genannte Funktion wurde von OMII Europe [OMII] entwickelt und ist seit Version 2.0.18 Bestandteil der Software VOMS-Admin.

Auch in dieser Version muss zunächst das (ggf. kurzlebige) Zertifikat des Nutzers an der VO registriert werden. Anschließend können - wie gehabt - vom VO-Repräsentanten einige VO-Rollen und -Attribute vergeben werden. Über eine Axis-Schnittstelle können dann die VO-Rollen und –Attribute in Form einer SAML Assertion abgefragt werden. Der Nutzer authentifiziert sich dabei mit dem schon bei der Registrierung verwendeten Zertifikat.

2.4 VO-Management im D-Grid-Umfeld: GRRS

Um die D-Grid-VO-Management-Architektur zu verstehen, ist die Abbildung aus [VOM] wohl am anschaulichsten. Sie wird in Abbildung 1 repliziert.

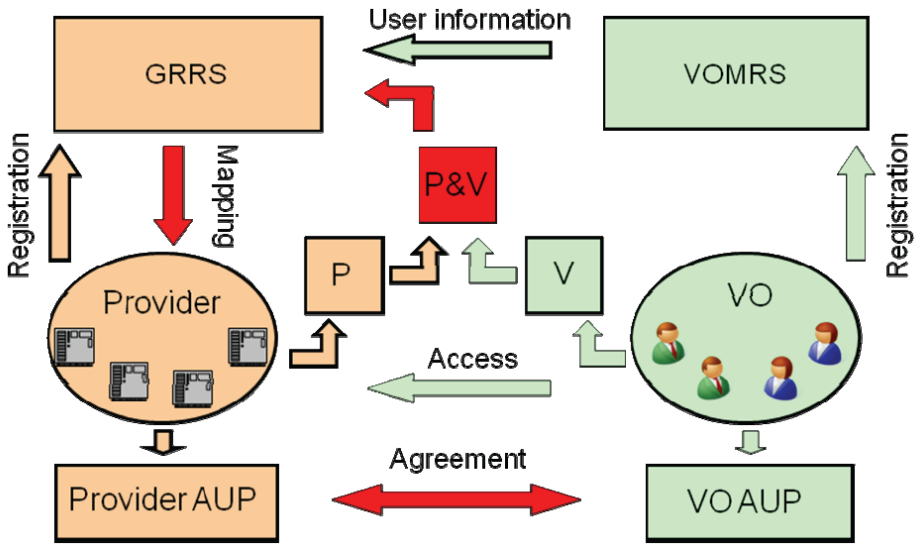

Abbildung 1: D-Grid-VO-Management

Die zentrale Komponente ist dabei der GRRS (Grid Ressource Registry Service). Er hält eine zentrale Datenbank vor, die alle für die Infrastruktur wichtigen Daten der zur Verfügung gestellten Ressourcen abspeichert: die Netzwerknamen der Ressourcen, Beschreibung der Ressourcen, E-Mail-Adressen und DNs der Grid-User-Zertifikate der Administratoren, sowie die Grid-Host-Zertifikate für die Identifikation der Ressource im Grid.

Der GRRS ist eine D-Grid-Eigenentwicklung und spielt mit dem VOMRS zusammen. Er sorgt dafür, dass die Ressourcen, die die unterstützten Middlewares Globus Toolkit, gLite bzw. UNICORE anbieten, mit dem Skript `dgridmap` das Mapping von Benutzerzertifikaten zu Systemaccounts in der jeweils benötigten Form erhalten. Dabei wird sichergestellt, dass nur Benutzer und VOs im Mapping sind, die auch auf den jeweiligen Ressourcen laut Policy zugelassen sind.

3 Aufbau der Testbeds

Für die Tests wurden Short-Lived Certificates (SLCs) aus dem Test-SLC Service des DFN verwendet, die im Gegensatz zu Zertifikaten aus dem „offiziellen“ DFN-SLCS nicht akkreditiert sind. Dies hat organisatorische Gründe, da der Beitritt einer Einrichtung zum DFN-SLCS an strenge Bedingungen gebunden ist. Da die vom Test-SLCS ausgestellten Zertifikate nicht in der Produktiv-Infrastruktur des D-Grid akzeptiert werden können, wurde hier eine Entwicklungsumgebung aufgebaut, die der VO-Management-Infrastruktur des D-Grid weitestgehend ähnelt. Somit konnte in einer Testumgebung ohne Auswirkungen auf die Produktiv-Infrastruktur des D-Grid entwickelt werden. Die Ergebnisse sind aber später auf diese übertragbar.

Bei den hier vorgestellten Entwicklungen handelt es sich z.T. um Erweiterungen von Entwicklungen aus den anderen Projekt-Tasks, deren bereits bestehende Testbeds teilweise ebenfalls übernommen werden konnten. Für die Entwicklungen zu diesem Task kommen ein

VOMS- bzw. ein VOMRS-Server hinzu. Zunächst wurde der VOMS/VOMRS Server (vgl. Kap. 3.2.1) aufgesetzt, der hauptsächlich von DAASI verwendet wurde. Parallel dazu wurde ein weiterer Server installiert und betrieben, auf dem die neuere Version von VOMS Admin mit SAML läuft. Beide Server können parallel genutzt werden, so dass die Arbeiten dazu entkoppelt sind.

3.1 Testbed AWI

3.1.1 VOMS

Für den VOMS-Server wurde die Software VOMS Admin Server 2.0.18 aus gLite 3.1 installiert [VOMS1]. Als Basis wurde eine Maschine mit Scientific Linux (Release 4.8 – Berillium) verwendet, da die gLite Software eng an diese Distribution angepasst ist.

3.1.2 Shibboleth SP mit Liferay Portal

Der Testrechner für das Portal Delegation Szenario läuft unter Ubuntu 10.04 LTS und gehört über einen Shibboleth Service Provider Version 2.2.1 der DFN-Test-Föderation an. Für den Test der Servlets wurde ein Tomcat-Container (Version 5.5) verwendet. Die Tests des neu entwickelten Shibboleth Auto Login Moduls und der Portal Delegation Portlets wurden mit der Portal-Software Liferay Version 6.0.5 (mit Tomcat 6) durchgeführt. Zudem sind auf dieser Maschine Teile von Globus Toolkit 4.0.8 installiert, um Grid-Jobs an die Test-Ressourcen abschicken zu können.

An diesem Service Provider stehen folgende Campus-Attribute aus der Shibboleth-Umgebung zur Verfügung:

- user id (urn:oid:0.9.2342.19200300.100.1.1)
- eduPersonPrincipalName (urn:oid:1.3.6.1.4.1.5923.1.1.1.6)
- eduPersonAffiliation（urn:oid:1.3.6.1.4.1.5923.1.1.1.1)
- cn (urn:oid:2.5.4.3)
- e-Mail (urn:oid:0.9.2342.19200300.100.1.3)
- eduPersonScopedAffiliation (urn:oid:1.3.6.1.4.1.5923.1.1.1.9)
- surname (urn:oid:2.5.4.4)
- eduPersonEntitlement (urn:oid:1.3.6.1.4.1.5923.1.1.1.7)
- given name (urn:oid:2.5.4.42)
- targeted-Id (urn:oid:1.3.6.1.4.1.5923.1.1.1.10)

3.1.3 Test-Ressource

Als Test-Ressourcen stehen zwei Rechner (Ubuntu 10.04 LTS) mit Globus Toolkit 4.0.8 und Gridshib for Globus Toolkit 0.6 zur Verfügung, um die vom Portal Delegation Portlet generierten Credentials testen zu können. Auf einer der beiden Maschinen ist zudem die in Task 4 entwickelte Erweiterung installiert, mit der auch die am Portal zusammengestellten SAML Assertions akzeptiert werden.

3.2 Testbed DAASI

3.2.1 VOMRS/VOMS

Es wurden auf einem am AWI befindlichen Rechner (Ubuntu 10.04 LTS) ein VOMRS und ein VOMS-Server aufgebaut. Zur Installation wurde das Virtual Data Toolkit (VDT, eine Distribution des Open Science Grid) verwendet, welches in Version 2.0.0 beide Systeme enthält. Die installierten Versionen sind:

- VOMRS 1.3.4a
- VOMS Admin 2.0.15-
- VOMS Client 1.8.8-2p1
- VOMS Server 1.8.8-2p1

Zusätzlich wurde MySQL als Datenbank für beide Systeme und Tomcat5 als Servlet Container verwendet. Es wurden mehrere Test-VOs erstellt und die Synchronisierung VOMRS → VOMS konfiguriert. Alle APIs des VOMRS wurden getestet, also die Kommandozeile auf dem Rechner selbst, die Web-Schnittstelle und die Web-Service-Schnittstelle mit dem enthaltenen SOAP Client

In D-Grid werden Attribute bei der Registrierung im VOMRS vom Benutzer erhoben, die nicht alle in der VOMRS-Standardinstallation spezifiziert sind. Diese sind:

- First name
- Last name
- Phone
- Nationality
- Street or P.O. Box
- Institute or Department
- Zipcode
- City
- Country

Die VOMRS-Instanzen wurden entsprechend konfiguriert, dass genau diese Benutzerinformationen bei der Registrierung erfragt werden.

3.2.2 Globus Toolkit

Auf einem Rechner an der SUB Göttingen, der auch als Testmaschine in TextGrid dient, wurde ein Globus Toolkit 4.2.1 installiert, wobei v.a. GridFTP genutzt wird. Auf diesem Rechner befinden sich ebenfalls zwei völlig funktionale TextGrid-Instanzen, von denen eine für die GAP-SLC-Arbeiten verwendet wird. Zu dieser Instanz gehören u.a. die Dienste TG-crud und TG-auth* mit openRBAC, das auf einer openLDAP-Datenbank operiert.

4 Lösungen

Die spezifischen Anforderungen aus den einzelnen Grid-Communities wurden in Einzellösungen erfüllt, die jedoch soweit als möglich gemeinsam entwickelte Bestandteile enthalten.

4.1 C3Grid / Portal

Im C3Grid soll folgender Use case verwirklicht werden:

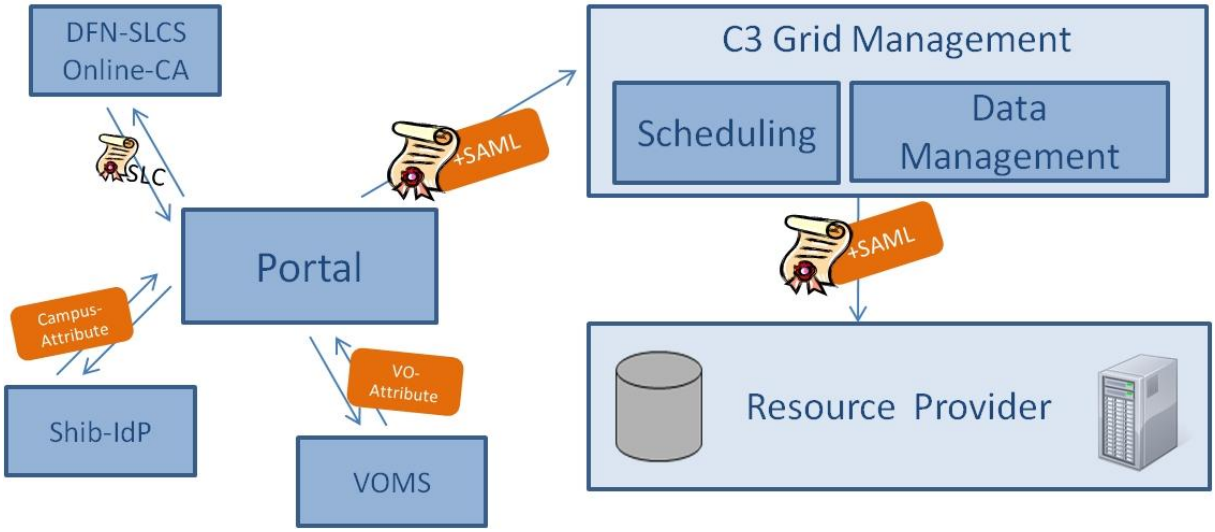

Abbildung 2 C3Grid-Use Case

Zunächst bezieht das Portal Delegation Portlet ein kurzlebiges Zertifikat und sammelt danach Attribute aus unterschiedlichen Quellen. Bei diesen Attributen handelt es sich um Campus Attribute aus der Shibboleth Umgebung, sowie Rollen und Attribute aus einer Virtuellen Organisation. Dabei liegen sowohl die Campus- als auch die VO-Attribute jeweils in Form einer SAML Assertion vor.

4.1.1 Bezug von SAML-Assertions im Portal

Der Bezug der Campus-Attribute wurde bereits in [SLC3] beschrieben. Im Rahmen von Task 4 wurde das Servlet erweitert, um zusätzlich VO-Rollen und -Attribute vom VOMS-Server zu integrieren.

Für den Bezug der als SAML 2 Assertion codierten Attribute und Rollen vom VOMS Server wurde ein bereits existierende Test-Client [VOMS2] genutzt, der angepasst und in das Portal Delegation Portlet integriert wurde. Dabei wird das vom Portal Delegation Portlet bezogene SLC direkt vom eingebetteten Client zur Authentifizierung am VOMS-Server verwendet

In der eingesetzten Lösung wird das Zertifikat des Nutzers am Client (von der verwendeten Axis-Bibliothek) gespeichert und beim nächsten Mal wiederverwendet. Daher ist dieses Verfahren nur für Einzelnutzung anwendbar. Vor dem Einloggen eines weiteren Nutzers muss jedes Mal der Container neu gestartet werden, was nicht praktikabel ist. Von den

Entwicklern war nur der Use Case vorgesehen, für jeden Benutzer (bzw. für jedes Nutzerzertifikat) einen eigenen Prozess zu verwenden.

In unserer Implementierung ist der Client jedoch selbst in einen Server-Prozess eingebunden und muss – ohne Neustart – mehrere Benutzer bedienen können. Es handelt sich dabei um ein generelles Problem, das im Apache-Wiki bereits dokumentiert wurde und wofür ein Workaround existiert [AXIS]. Konkret handelt es sich um eine Anpassung der Klasse *SocketFactoryFactory* aus der Axis-Bibliothek sowie die Ergänzung um einige weitere Klassen

Die Anpassungen wurden für unseren Anwendungsfall übernommen, so dass nun für mehrere Nutzer automatisch die zu dem jeweiligen Zertifikat in einer VO hinterlegten VO-Rollen und –Attribute geladen werden.

4.1.2 Zusammenführung Campus- und VO-Attribute

Die bezogenen Attribute müssen nun an den Ressourcenanbieter weitergeleitet werden. Technisch ist es kein Problem, alle gesammelten SAML Assertions in ein vom kurzlebigen Zertifikat abgeleitetes Proxy Zertifikat einzubetten. Doch leider kann die auf den Grid-Ressourcen eingesetzte Software nur jeweils eine einzige eingebettete SAML 1 Assertion auswerten. Aus diesem Grunde werden die in den SAML 2 Assertions enthaltenen Attribute zunächst gesammelt und in einer vom Portal neu ausgestellten und signierten SAML 1 Assertion vereint, die dann in das Proxy-Zertifikat eingebettet werden kann.

In Analogie zu den Task 3 werden dazu die einzelnen SAML Assertions nacheinander abgerufen und die darin enthaltenen Daten in einer eigenen Datenstruktur zwischengespeichert. Danach wird eine neue SAML Assertion durch das Portal generiert, die zwischengespeicherten Attribute in die neu erzeugte SAML Assertion eingefügt und diese mit dem Schlüssel des Portals signiert. Dafür konnten die in Task 3 entwickelten Teile der Software wiederverwendet werden, da bei Konzeption bereits mehrere Datenquellen für Attribute vorgesehen waren.

4.1.3 Integration in bestehende Infrastruktur

Das entwickelte Portlet kann über die Portal-eigenen Interfaces im laufenden Betrieb deployed werden. Mit der enthaltenen Datei portlet.xml wird das Portlet konfiguriert.

Der Name der VO, von der die Rollen abgefragt werden sollen, muss für jeden Nutzer individuell konfigurierbar sein. Dazu werden sog. „Custom Fields“ genutzt: Der Liferay Administrator muss dazu einmalig ein Feld mit dem Schlüssel ‚voName‘ anlegen. Für dieses Feld benötigt der jeweilige Nutzer ein Schreib- und Leserecht, was ebenfalls vom Administrator konfiguriert werden muss.

Vom Nutzer muss, bevor das Portal Delegation Portlet verwendet wird, einmalig der Name der VO konfiguriert werden. Dies geschieht über das Benutzer-Menü von Liferay, wo die Unterkategorie „Custom Fields“ zu finden ist. Die hier gesetzten Werte werden von Liferay gespeichert und müssen lediglich bei einem Wechsel der VO erneut konfiguriert werden.

4.2 TextGrid / Web Services

4.2.1 Bisheriger Status

Bevor die Arbeiten aus diesem Kapitel vorgenommen wurden, war das VO-Management in TextGrid wie in **Fehler! Verweisquelle konnte nicht gefunden werden.** dargestellt.

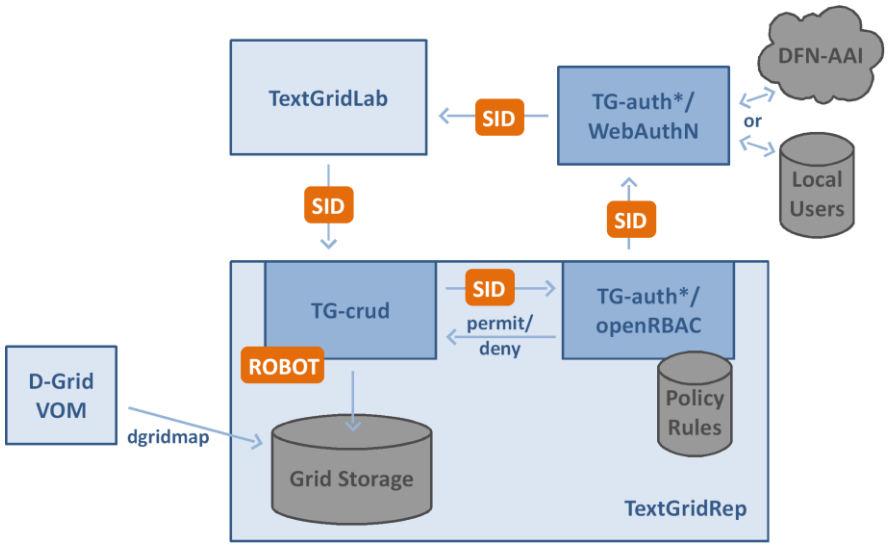

Abbildung 3: VO-Management-Infrastruktur in TextGrid

Zwei Dinge sind an dieser Abbildung im vorliegenden Kontext wichtig: Zum einen wird ein ROBOT-Zertifikat eingesetzt, mit dem Dateioperationen auf dem Grid-Speicher geschehen. Dieses ROBOT-Zertifikat wird über das normale D-Grid-VO-Management verwaltet, d.h. im VOMRS zur VO „textgrid“ hinzugefügt (einmalig) und vom GRRS über das dgridmap-Skript auf einen Benutzeraccount abgebildet (täglich). Die eigentliche Benutzerverwaltung geschieht über ein Web-basiertes Skript (WebAuthN), das mit der Autorisierungskomponente (openRBAC) kommuniziert. Die Benutzer für WebAuthN kommen entweder aus einer separaten lokalen LDAP-Benutzerdatenbank oder über Shibboleth / die DFN-Test-AAI. Die lokalen Benutzer müssen sich vorher mit E-Mail-Verifikation und manueller Überprüfung der Authentizität bei TextGrid registrieren. Benutzer, die über die DFN-AAI kommen, können ohne explizite Registrierung in TextGrid arbeiten; hier genügt die Übermittlung eines eduPersonPrincipalNames vom Identity Provider der Heimathochschule, wodurch die Identifizierbarkeit gewährleistet ist.

Das andere ist die Verwendung einer SessionId (SID), mit dem anstatt eines persönlichen Zertifikats der Benutzer identifiziert wird. Die SID wird von openRBAC nach Anstoß durch WebAuthN erzeugt und an den Rich Client, das TextGridLab, zurückgegeben. Sobald im Namen des Benutzers Dateioperationen oder auch Operationen auf anderen TextGrid-Services getätigt werden müssen, erfolgt dies unter Angabe der SID. Es liegt in der Verantwortung des Dienstes mit dem ROBOT-Zertifikat (TG-crud), dass Dateioperationen tatsächlich im Namen des Benutzers erfolgen. Insbesondere muss TG-crud bei der Autorisierungskomponente mittels der empfangenen SID prüfen, ob der Benutzer auf die angeforderte Datei in der beabsichtigten Weise zugreifen darf. Es liegt hier also ein

klassisches PEP-PDP-Szenario vor: TG-crud ist der Policy Enforcement Point, und TG-auth* bzw. openRBAC der Policy Decision Point.

Diese Infrastruktur hat den Vorteil, dass sie kollaboratives Arbeiten leicht ermöglicht, da Objekte im Grid-Speicher auf unterster Ebene nicht nach Benutzeraccounts getrennt sind und somit keine Beschränkungen durch das Dateisystem bestehen. Zu jeder Ressource kann feingranular in openRBAC verfügt werden, welche Rollen was dürfen.

Die Komponenten kommunizieren über Web Services miteinander, also SOAP und/oder REST. Es war eine Designentscheidung zu Beginn von TextGrid, dass diese statt WSRF-Services verwendet werden, da für letztere zum damaligen Zeitpunkt kaum taugliche Bibliotheken vorhanden waren. Somit können (externe) Programmierer einfacher TextGrid-Dienste implementieren. Der Zugriff auf den Grid-Speicher wird von TG-crud gekapselt, der nach oben Web Services anbietet, nach unten aber über die GAT-Bibliotheken das Globus Toolkit anspricht.

4.2.2 Neuer Status

Durch die Einführung von SLCs in TextGrid – es wird das Portal Delegation Szenario verwendet – ergeben sich auch Implikationen für das VO-Management. Zum Vergleich mit der ursprünglichen Architektur ist in Abbildung die neue Architektur schematisch dargestellt.

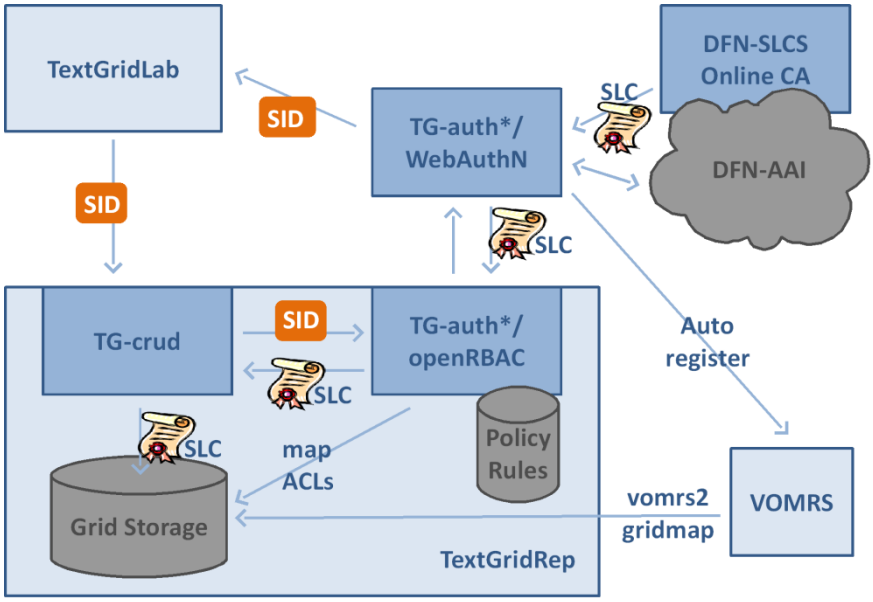

Abbildung 4: TextGrid-Architektur mit Einbezug des SLCS und D-Grid-kompatiblen VO-Management

Die wichtigste Neuerung ist der Einbezug des DFN-SLCS, der SLCs für die über die DFN-AAI / Shibboleth authentifizierte Benutzer ausstellt. Somit müssen alle TextGrid-Benutzer Mitglied einer Organisation sein, die einen Identity Provider in der DFN-AAI betreibt und ihre Benutzer für den SLCS freischaltet. (Hierzu muss das Attribut eduPersonEntitlement mit dem Wert urn:geant:dfn.de:dfn-pki:slcs im Benutzereintrag vorliegen.) Die TextGrid-eigene lokale Benutzerverwaltung existiert nicht mehr. Wenn im Zuge der Shibboleth-Authentifizierung ein SLC vom SLCS ausgestellt wird, sorgt die Komponente WebAuthN dafür, dass das SLC von

openRBAC gespeichert wird. Dies kann nun durch TG-crud angefordert werden, der es anstatt des ROBOT-Zertifikats für die Authentifizierung gegenüber der Grid-Ressource einsetzt. Details hierzu sind in [SLC1] zu finden.

Die Auswirkungen auf das VO-Management werden nun in den folgenden Abschnitten beschrieben, wobei die Verortung der einzelnen Komponenten aus Abbildung ersichtlich wird.

4.2.3 Schnittstelle TextGrid-Authentifizierung – VOMRS

Die Eigenschaft, dass sich Benutzer, die sich über die DFN-AAI authentifiziert haben, automatisch für TextGrid zugelassen werden, soll erhalten bleiben. Hierzu muss für D-Grid-Konformität dafür gesorgt werden, dass im Zuge der TextGrid-Authentifizierung diese Benutzer automatisch im VOMRS registriert werden (vgl. Pfeil „Autoregister“ in Abbildung). Dazu wurden folgende Schritte implementiert:

- Einmalige Registrierung eines Benutzers mit Administratorrechten in der VO, dessen Zertifikatspaar (ein langlebiges Zertifikat) auf dem Rechner mit dem WebAuthN-Skript hinterlegt wurde
- Erhebung der Benutzerinformationen, die vom VOMRS nach D-Grid-Policy gefordert werden. Das entsprechende Web-Formular wird vom WebAuthN-Skript angezeigt und es erfolgt eine Speicherung in openRBAC, damit diese Attribute nur beim ersten Login erfragt werden.
- Verwendung der Java-Klasse fnal.vox.vomrs.client.SoapClient, um eine Registrierung des Benutzers als Member der VO zu erwirken, mit dem oben erwähnten langlebigen Zertifikat des Administrators. Es werden die folgenden VOMRS-API-Funktionen aufgerufen:
    - getMembers – zur Überprüfung, ob Benutzer bereits vorhanden ist
    - registerMember – zur eigentlichen Registrierung mit allen Benutzerinformationen
    - setMbrRegistrationStatus – zur Bestätigung der Registrierung

Das Skript wird bei jeder Authentifizierung in TextGrid ausgeführt, wobei die VOMRS-Registrierung nur gemacht wird, wenn der authentifizierte Benutzer ein SLC bekommen hat.

Der Quellcode der Autoregistrierungsklasse sowie der gesamten WebAuthN-Komponente befindet sich im Subversion-Repository von TextGrid unter der URL https://develop.sub.uni-goettingen.de/repos/textgrid/trunk/middleware/tgauth/info.textgrid.middleware.tgauth.webauth.

4.2.4 Synchronisierung von VOMRS-Benutzern zur Ressource

Wie in Abschnitt 2.4 beschrieben, bewirkt in D-Grid die Komponente GRRS die Abbildung von VOMRS-Benutzereinträgen auf System-Accounts auf den Grid-Ressourcen. Der GRRS

ist eine Eigenentwicklung speziell für D-Grid und wird als zentraler Dienst am FZ Jülich bereitgestellt.

Da laut Entwickleraussagen die Erstellung einer eigenen Instanz in einer anderen Umgebung nicht möglich ist, wurde für das Testbed ein Pendant entwickelt. Das Skriptvomrs2gridmap (vgl. Abbildung) ruft die Mitgliederinformationen aus einer VOMRS-Instanz ab und erstellt ein Grid-Mapfile, wie es von Globus Toolkit 4 benötigt wird. Im Detail funktioniert das Skript folgendermaßen:

- Zunächst werden die DNs aller Mitglieder vom VOMRS bezogen. Dazu wird wie in Abschnitt 4.2.3 beschrieben die Java-Klasse fnal.vox.vomrs.client.SoapClient verwendet, die die API-Methode getMembers aufruft.
- Alsdann wird das bestehende Grid-Mapfile eingelesen.
- Etwaige nicht im Grid-Mapfile, aber im VOMRS vorhandene Einträge werden hinzugefügt. Dabei werden gleichzeitig die hierzu nötigen Systemaccounts erzeugt.

Das Skript wird in regelmäßigen Zeitabständen (alle fünf Minuten) durch den CRON-Demon auf der Ressource ausgeführt.

Der Quellcode des Skripts ist im erwähnten Subversion-Repository unter https://develop.sub.uni-goettingen.de/repos/textgrid/trunk/middleware/tgauth/info.textgrid.middleware.tgauth.vomrs2gridmap verfügbar.

4.2.5 PDP – ACL Synchronisierung zur Ressource

Bedingt durch den Wegfall des ROBOT-Zertifikats und die Verwendung persönlicher Zertifikate (also in diesem Fall SLCs) werden auf der Grid-Ressource nicht mehr alle Dateien unter demselben Systemaccount geschrieben, sondern unter verschiedenen Accounts in dem jeweiligen Home-Directory der Benutzer. Mit der rollenbasierten Autorisierung TG-auth*/openRBAC können jedoch vielfältige Autorisierungspolicies implementiert werden, die über die Möglichkeiten von Discretionary Access Control (DAC, dem Autorisierungs-mechanismus in einem UNIX-Dateisystem) hinausgehen. So werden z.B. einem Benutzeraccount verschiedene Rollen zugeordnet (z.B. „Projektleiter in Projekt A“, „Bearbeiter in Projekt B“), die in einer Session selektiv aktiviert werden können. Zu jeder Ressource wiederum werden in openRBAC Berechtigungsattribute vorgehalten, die über den Zugriff entscheiden (z.B. „Projektleiter in Projekt B dürfen die Ressource publizieren“, oder „Bearbeiter in Projekt A dürfen sie aktualisieren“).

Die Hauptfrage ist nun, wie die Information über die Zugriffspolicy, die TG-auth* vorhält, auf der Grid-Ressource genutzt werden kann. Dazu gibt es im Wesentlichen zwei Modelle:

- Das Pull-Modell: die Synchronisierung der Policies von TG-auth* zur Ressource in kurzen Zeitabständen, damit sie dort zur Verfügung stehen. Dieser relativ einfach zu implementierende Ansatz wird im vorliegenden Abschnitt behandelt.

- Das Callout-Modell mit PEP und PDP: die Ressource tritt als Policy Enforcement Point auf, der für Autorisierungsentscheidungen einen Policy Decision Point befragt und diese dann durchsetzt. Dieser Ansatz, der den Vorteil hat, dass die Entscheidung immer auf absolut aktuelle Policies beruht, allerdings erhebliche Entwicklungsarbeiten voraussetzt, wird im Abschnitt **Fehler! Verweisquelle konnte nicht gefunden werden.** behandelt.

Zur Umsetzung des Pull-Modells muss man sich zunächst im Klaren sein, dass hier eine Übersetzung von feingranularen RBAC-Policies auf DAC-Policies stattfinden muss. Dies ist im allgemeinen Fall nicht ohne Informationsverlust möglich [FKC]. Dennoch verbleibt ein relativ großer Teil von RBAC (so, wie in TextGrid implementiert), der in DAC abgebildet werden kann. Hier handelt es sich um die Berechtigungen „create“ (auf Projekten) und „read“, „update“ und „delete“ (auf Dateien) – dies sind genau die Operationen, die der Dienst TG-crud zur Verfügung stellt und die hier von Relevanz sind.

Die Idee ist im Wesentlichen die, dass TextGrid-Projekte als Verzeichnisse im Home-Directory der Benutzer angelegt werden. Mitglieder eines Projekts werden nun zu Mitgliedern einer UNIX-Gruppe, die die entsprechenden Accounts als Mitglieder hat. Die erwähnten Berechtigungen auf Projektverzeichnissen und Dateien werden nun mit erweiterten Access Control Lists (POSIX ACLs) modelliert.

Im Einzelnen werden die folgenden Schritte ausgeführt für eine Synchronisierung:

- Die Daten werden direkt aus der LDAP-Datenbank von openRBAC bezogen, und zwar regelmäßig (z.B. alle 5 Minuten) und nur die jeweils neuesten Änderungen.
- Es wird das Grid-Mapfile und die group-Datei des Systems geparst.
- Es werden die neuen Einträge im LDAP verarbeitet, abhängig von Typ:
    - Für Rollen: lege die entsprechende Systemgruppe an, falls noch nicht vorhanden. Alle Mitglieder dieser Rolle (sofern im Grid-Mapfile vorhanden) werden zu der Gruppe hinzugefügt.
    - Für Dateiressourcen: Es wird das Projektverzeichnis (TG-crud hat es beim Schreiben der Datei bereits angelegt) und der Dateiname festgestellt. Verzeichnis und Datei bekommen zunächst alleinige Nutzungsrechte durch den Eigentümer. Nun wird für jeden Berechtigungseintrag (also jede Rolle-Berechtigung-Kombination) dieser Ressource folgendes durchgeführt:
        - Für Berechtigung „create“: die der Rolle entsprechende Systemgruppe erhält die Nutzungsrechte „wx“ auf dem Projektverzeichnis
        - Für „read“: die entsprechende Systemgruppe erhält die Rechte „rx“ auf dem Verzeichnis und „r“ auf der Datei.
        - Für „update“: „x“ auf dem Verzeichnis und „w“ auf der Datei
        - Für „delete“: „wx“ auf dem Verzeichnis.

- Anschließend wird die Uhrzeit der letzten Abfrage für den nächsten Lauf gespeichert.

Dieser Algorithmus wurde als Perl-Skript implementiert und auf der Ressource installiert (vgl. Pfeil „map ACLs“ in Abbildung). Das Skript ist im TextGrid-Repository unter https://develop.sub.uni-goettingen.de/repos/textgrid/trunk/middleware/tgauth/info.textgrid.middleware.tgauth.pdp2acl zu finden.

4.2.6 Alternative: SAML-XACML-Callout von Ressource zum PDP

Der Hauptnachteil der im vorhergehenden Abschnitt 4.2.5 beschriebenen Pull-Variante liegt in der Tatsache, dass abhängig vom Synchronisierungsintervall von z.B. einer Minute die Zugriffskontrolle nicht immer aktuell ist. Eine optimalere Lösung wäre deshalb, wenn die Zugriffsentscheidung direkt vom PDP getroffen würde. Hierzu wurde in openRBAC serverseitig das XACML-SAML-Protokoll implementiert [XACMLSAML]. Dieses beschreibt die Elemente <XACMLAuthzDecisionQuery> und <XACMLAuthzDecisionStatement>, welche zum Austausch zwischen PEP und PDP dienen und über das XACML-SAML-Request-Response-Protokoll transportiert werden. Ein weiterer Vorteil dieser Lösung im Gegensatz zum vorher beschriebenen Pull-Modell ist, dass alle in openRBAC spezifizierten Zugriffsregeln auf der Ressource genau umgesetzt werden können.

Der XACML-SAML-Callout ist erstmals in Globus Toolkit 4.2.1 im Zuge der flexiblen Autorisierungsinfrastruktur integriert worden, allerdings nur für die Web-Services-basierten Dienste. Es wird dabei folgendes mitgegeben:

- Subject-DN aus dem Zertifikat
- Name der Ressource (Datei)
- gewünschte Operation für Zugriff

Für TextGrid war diese Implementierung allerdings leider nicht nachnutzbar, da der Zugriff auf die Grid-Speicher-Ressourcen ausschließlich über das nicht Web-Services-basierte GridFTP erfolgt. Um den Call-Out-Ansatz weiter verfolgen zu können, müsste also die GridFTP-Server-Implementierung von Globus direkt modifiziert werden, damit dieser bei jedem Dateizugriff den externen PDP bezüglicvh der Zugriffsrechte befragen kann.

Hierzu wurde ein existierender C-Client, der das XACML-SAML-Protokoll implementiert (http://www.mcs.anl.gov/~bester/xacml/) getestet und so modifiziert, dass er mit dem openRBAC-basierten PDP kommunizieren kann. Zur Fertigstellung dieser Lösung müsste nun in einem zweiten Schritt der so angepasste C-Client in den Globus GridFTP-Server integriert werden.

5 Zusammenfassung

Mit Kollaborationen über Institutsgrenzen hinweg, wie sie in Grids oder –allgemeiner formuliert- in virtuellen Forschungsumgebungen (VREs) unterstützt werden, verschieben sich auch einige Verantwortlichkeiten und Zuständigkeiten. Zwar hat die Heimateinrichtungen eines Nutzers immer noch eine Reihe von Informationen über ihn, die sie mittels entsprechender Zusicherungen den Ressourcenanbietern übermitteln kann. Aber immer größer wird die Rolle von virtuellen Organisationen, die sich um wissenschaftliche Projekte herum organisieren. Entsprechend müssen auch aus dieser Quelle Aussagen über die Rolle eines Nutzers in der VO bei der Autorisierungsentscheidung einfließen.

Die vorliegenden Arbeiten haben dieses Thema aufgegriffen und exemplarisch Lösungsansätze aufgezeigt.

6 Literatur

[AXIS] http://wiki.apache.org/ws/FrontPage/Axis/DynamicSSLConfig

[FKC] David Ferraiolo, D. Richard Kuhn, Ramaswamy Chandramouli: Role-based Access Control. Artech House, Boston, Mass., Artech House computer security series, 2003.

[IVOM] IVOM: Interoperability and Integration of VO Management Technologies in D-Grid, Work Package 1: Evaluation of international Shibboleth-based VO Management Projects

[OMII] OMII Europe, http://www.omii-europe.org

[SLC1] Nutzung von kurzlebigen Zertifikaten in portalbasierten Grids (GapSLC) - Task 1: Deliverable D1, http://gap-slc.awi.de/documents/GapSLC-D1-V1.0.pdf

[SLC2] S. Pinkernell, B. Fritzsch, Einsatz von Portal Delegation und SAML Assertions bei der Authentifizierung und Autorisierung, http://gap-slc.awi.de/documents/portalDelegation-1.0.pdf

[SLC3] Nutzung von kurzlebigen Zertifikaten in portalbasierten Grids (GapSLC) – Task 3: Deliverable D3, http://gap-slc.awi.de/documents/GapSLC_D3_V1.0.pdf

[VOM] Betriebskonzept für die D-Grid Infrastruktur. Thomas Fieseler (Koordination).

[VOMS1] gLite 3.1, https://twiki.cern.ch/twiki/bin/view/EGEE/GLiteVOMS

[VOMS2] VOMS Client von Andrea Ceccanti, Download per CVS: cvs –d :pserver:anonymous@glite.cvs.cern.ch:/cvs/glite/ co –r glite-security-voms-admin-server_R_2_0_18_1 org.glite.security.voms-admin-server

[XACMLSAML] SAML 2.0 profile of XACML v2.0 http://docs.oasis-open.org/xacml/2.0/SAML-PROFILE/access_control-xacml-2.0-saml-profile-spec-os.pdf