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 1: Deliverable D1

Arbeitspaket:	Task 1
Autor(en):	Stefan Pinkernell (AWI) Bernadette Fritzsch (AWI) Martin Haase (DAASI) Stefan Funk (DAASI) Peter Gietz (DAASI)
Version:	1.0
Publikationsdatum:	6.5.2010
Status:	internal
Kontakt:	Stefan Pinkernell
Email:	Stefan.Pinkernell@awi.de

GEFÖRDERT VOM

Bundesministerium
für Bildung
und Forschung

Inhaltsverzeichnis

1 Einführung

Die Nutzung der Kern-D-Grid-Ressourcen ist bisher an eine Authentifizierungs- und Autorisierungs-Infrastruktur gebunden, die auf persönlichen X.509 Zertifikaten beruht, wobei die Nutzerzertifikate von einer in EUGridPMA akkreditierten Zertifizierungsstelle stammen müssen (siehe z.B. [1]). Die bisherigen Erfahrungen in D-Grid haben gezeigt, dass es jedoch breite Nutzerkreise gibt, die Probleme beim Umgang mit den persönlichen Zertifikaten haben, so dass für sie eine hohe Einstiegsschwelle in das Grid besteht. Die Probleme betreffen dabei mehrere Bereiche von Zertifikatserlangung bis zur -anwendung: hoher Aufwand für den Nutzer durch wiederholtes persönliches Ausweisen bei der Beantragung bzw. Verlängerung von Zertifikaten, teilweise fehlende regional Authorities (z.B. an Unternehmen) und Probleme bei eventuell notwendiger Formatkonvertierung der Zertifikate stehen beispielhaft für mögliche Hürden.

Hier können kurzlebige Zertifikate (short lived credential, SLC) genutzt werden, da sie vom Anwender jeweils bei Bedarf relativ einfach angefordert werden können und da wegen ihrer kurzen Lebensdauer der Verwaltungsaufwand für den Nutzer entfällt. Das vorliegende Dokument beschreibt einen Anwendungsfall für die Nutzung von SLCs in einer portalbasierten Umgebung. Dabei wird zur Authentifizierung des Nutzers Shibboleth eingesetzt, das sich in den vergangenen Jahren zunehmend international verbreitet hat und in einigen Ländern die Grundlage der nationalen e-Science und Grid-Infrastruktur bildet (bspw. [2] und [3]).

2 Der DFN-SLCS

Der DFN bietet seit 2008 einen Dienst zur Ausgabe von kurzlebigen X.509 Zertifikaten an, der auf der Authentifizierungs- und Autorisierungs-Infrastruktur DFN-AAI (Details dazu unter [4]) basiert. Die Gültigkeitsdauer dieser Zertifikate ist auf maximal 1.000.000 Sekunden begrenzt, was etwa 11,5 Tagen entspricht. Beantragt werden diese Zertifikate online. Dazu enthält der SLCS eine Online Zertifizierungsstelle, die die kurzlebigen Zertifikate automatisch ausstellen kann, sowie Online-Schnittstellen um Zertifikat-Requests (CSR) einreichen und kurzlebige Zertifikate direkt ausliefern zu können.

Derzeit existieren beim DFN zwei von der technischen Implementierung her identische Varianten dieses Dienstes:

(1) Der EUGridPMA-akkreditierte SLCS der DFN-PKI: Die Nutzung dieses SLCS ist an einige Voraussetzungen gebunden. So muss die Person einen Eintrag in einem in die DFN-AAI integrierten Identity Provider (IdP) haben, wo für sie der URN "urn:geant:dfn.de:dfn-pki:slcs" im Attribut "eduPersonEntitlement" eingetragen ist. Weiterhin sind an die Einrichtung und den Betrieb der SLCS-RA strenge Anforderungen gestellt. Zu Details siehe [5]

(2) Daneben gibt es noch einen nicht akkreditierten SLCS, der als Test-System in der Test-AAI verfügbar ist.

Da ein kurzlebiges Zertifikat aus dem akkreditierten SLCS eine vergleichbare Vertrauensbasis wie ein persönliches Zertifikat darstellt und daher von den Ressourcenprovidern als gleichwertig behandelt wird, sind bei seiner Handhabung ähnlich

strenge Anforderungen zu erfüllen. So muss der private Schlüssel zu den kurzlebigen Zertifikaten ausschließlich der direkten Kontrolle des jeweiligen Nutzers unterliegen und darf niemals beispielsweise in einem Portal abgelegt werden.

Der webbasierte Ablauf des SLC-Bezugs ist in Abbildung 1 schematisch dargestellt und gestaltet sich wie folgt: Zunächst ruft der Nutzer die Website des DFN-SLCS auf (1). Von dort aus wird er zunächst zum Where-Are-You-From-Service (WAYF) weitergeleitet (2) und selektiert dort aus einer Liste seine Heimateinrichtung, zu dessen Identity-Provider-Seite er daraufhin weitergeleitet wird (3). Dort authentifiziert er sich mit seinem Nutzernamen und Passwort. Nach erfolgreicher Authentifizierung wird er auf die Website des DFN-SLCS zurückgeleitet (4), wo nun ein Zugang möglich ist. Von dieser Seite aus kann dann eine Java-Webstart-Applikation (Gridshib CA Credential Retriever) gestartet werden, die auf dem Rechner des Nutzers ein Schlüsselpaar und einen Zertifikat-Request erzeugt (5). Dieser Zertifikatrequest wird an die Online CA gesendet (6). Das kurzlebige Zertifikat wird dann zusammen mit dem privaten Schlüssel im Dateisystem des Rechners des Nutzers gespeichert.

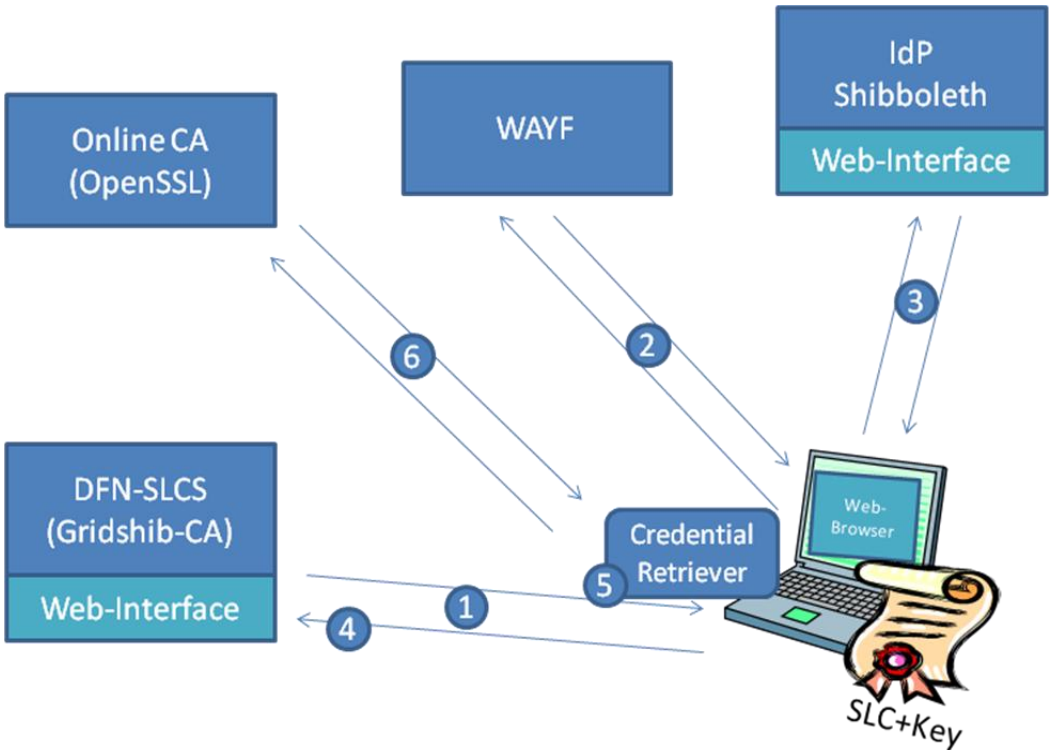

Abbildung 1: Bezug eines SLC über das Web-Interface

3 Nutzung des SLCS im portalbasierten Grid-Zugang

3.1 Konzept

Im Grid-Portal-Szenario übernimmt das Portal das Management der Zertifikate. Geplant ist die Realisierung des folgenden Szenarios: Der Nutzer bezieht webbasiert nach dem oben beschriebenen Verfahren ein SLC (Schritte 1-6 in Abbildung 2), dessen persönlicher Schlüssel auf seinem privaten Rechner verbleibt. Mit Hilfe eines noch zu entwickelnden Upload Tools wird ein Proxy in das Portal geladen (Schritt 7) und kann dann zur Initiierung der im Grid ablaufenden Prozesse genutzt werden. Der prinzipielle Aufbau des Upload

Tools ist bereits von DFN ausgearbeitet, vor einer Implementierung soll aber noch der wirkliche Bedarf in den Communities ermittelt werden.

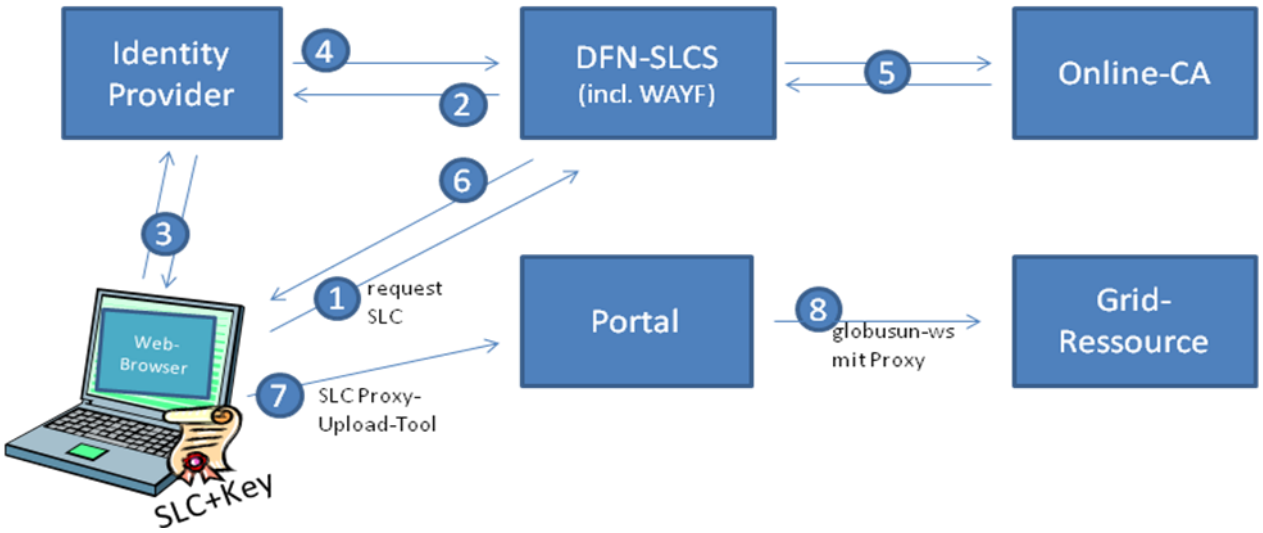

Abbildung 2: Ablauf im Grid-Portal-Szenario

Da das Proxy-Upload-Tool, mit dem das vom SLC abgeleitete Proxy zum Portal hochgeladen werden kann, zur Zeit noch nicht verfügbar ist, wurde als Zwischenlösung ein vereinfachter Ablauf entworfen und umgesetzt, der in Abbildung 3 dargestellt ist. Dabei wird das SLC nach dem Login direkt am Portal bezogen und samt privatem Schlüssel auch dort abgelegt. In einem zweiten Schritt kann ein Proxy vom SLC abgeleitet und im weiteren Verlauf vom Portal genutzt werden. Dazu meldet sich der Nutzer am Portal per Nutzername und Passwort an. Dies kann je nach Funktionsumfang des Portals wahlweise über eine lokale Nutzerdatenbank oder aber auch per Shibboleth-Login umgesetzt werden. Daraufhin werden am Portal ein Schlüsselpaar und ein Zertifikat-Request generiert. Per Portal Delegation wird der Nutzer auf die Website der Gridshib-CA weitergeleitet und, falls noch nicht geschehen, mittels Shibboleth über die DFN-AAI Föderation authentifiziert. Nach erfolgreicher Authentifizierung wird ein kurzlebiges Zertifikat ausgestellt und der Nutzer samt Zertifikat an das Portal zurückgeleitet.

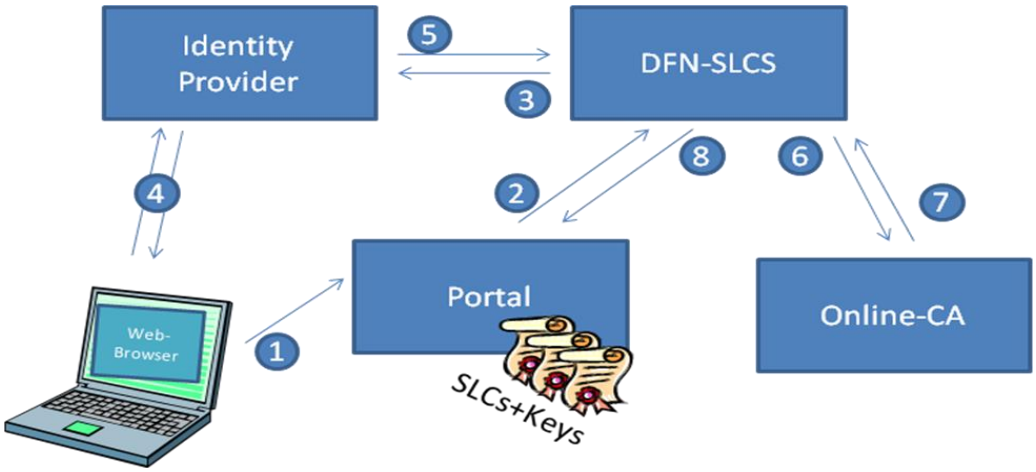

Abbildung 3: Konzept Portal Delegation

Der wesentliche Unterschied zum in Abbildung 2 beschriebenen Szenario besteht darin, dass das Portal im Auftrag des Nutzers das SLC holt und das Zertifikat sowie der persönliche Schlüssel auf dem Portalserver liegen. Letztes stellt eine Verletzung der Nutzungsbedingungen für den akkreditierten DFN-SLCS dar. Daher wird in diesem Verfahren nur der nicht-akkreditierte Dienst genutzt. Die damit gemachten Erfahrungen sind aber wegen der Baugleichheit der beiden Dienste sofort auf das eigentliche Grid-Portal-Szenario übertragbar, sobald das upload tool verfügbar ist.

Um Portal Delegation nutzen zu können muss das Portal allerdings einige Bedingungen erfüllen:

- Der Zugriff muss per https geschützt sein.
- Das Portal muss ein Schlüsselpaar sowie einen Zertifikat-Antrag erstellen können.
- Der Benutzer muss authentifiziert werden
- Die Session muss von der Erzeugung des Schlüsselpaars bis zu dem Zeitpunkt, an dem das Zertifikat zurückgegeben wird, aufrechterhalten bzw. gespeichert werden können.

3.2 Technische Umsetzung

3.2.1 Testumgebung

Die vorgestellten Lösungen wurden in einer Testumgebung aus virtuellen Linux-Maschinen implementiert, getestet und eingerichtet. Folgende Basisinstallation wurde dabei verwendet:

- Ubuntu Version 8.04 LTS
- Apache Webserver Version 2.2.1 und Apache Tomcat Version 5.5

Zudem wurde ein IdP in der Version 2.1.4 eingesetzt. Es ist auch möglich, zu Testzwecken einen Test-IdP – z.B. vom DFN – zu nutzen. Dazu wird auf der Seite des WAYF-Servers beispielsweise der „DFN Test-IdP 2.x“ ausgewählt.

In dieser Testumgebung wurden die Tests mit dem Perl-Script und der Java-Servlet Portierung erfolgreich durchgeführt.

3.2.2 Bezug über Perl-Script

Für das Beziehen des kurzlebigen Zertifikats wird im Paket GridShib-CA das Perl Script gridshib-ca-demo-portal.cgi bereitgestellt. Dieses Script wurde in den lokalen Adressen sowie der Adresse des DFN-SLCS modifiziert und in die Testumgebung integriert. Dazu waren folgende Anpassungen notwendig:

In der Originalversion werden die Daten (Zertifikat-Request, etc.) vom gridshib-ca-demo-portal-CGI-Script aus per HTTP-POST an das portalLogin CGI-Script an der Gridshib-CA übertragen. Dies führt zu der Fehlermeldung „ERROR: No portal URL given“. Offensichtlich werden einige der Headerinformationen während des „Shibboleth-Login-Dance“ an einer der zahlreichen Zwischenstationen nicht weitergeleitet. An welcher Stelle genau dieser Fehler auftritt, wurde nicht weiter verfolgt.

Daher wurde die Übertragungsmethode modifiziert. Statt die Daten per HTTP-POST im Header zu senden, werden sie nun mittels HTTP-GET als Parameter-Wert-Paare als Teil der URL übergeben. Die mit dieser Methode übertragbare Datenmenge ist zwar geringer als bei HTTP-POST, reicht für diese Zwecke aber aus. Allerdings können bei HTTP-GET die zu übertragenden Daten (Zertifikat-Request, eigene Portal-Adresse) einfacher durch Dritte ausgelesen werden. Mit Einbinden den Proxy-Upload-Tool wird diese Sicherheitslücke in Zukunft aber geschlossen.

Desweiteren müssen für das Script die Adresse für PortalLogin am Test-SLCS sowie die eigene Portaladresse konfiguriert werden. Dazu kann entweder direkt das Script gridshib-ca-demo-portal.cgi modifiziert werden, oder die Attribute werden in der Datei Config.pm im Perl-Paket Gridshib-CA angepasst.

Zu erwähnen wäre an dieser Stelle auch, dass der IdP und die Adresse des Portals beim DFN bekannt gemacht werden müssen, um den Service nutzen zu können.

Für die Realisierung des Portal Delegation Szenarios sind mehrere Komponenten notwendig, die bei den unterschiedlichen Schritten im Szenario zum Einsatz kommen. Ihre Rolle wird in Abbildung 4 schematisch dargestellt und im Folgenden kurz erläutert (siehe auch [6]).

Über die Weboberfläche (1) kann der Nutzer mit Hilfe einer Schaltfläche den Zertifikat-Antrag an die Gridshib-CA senden. Dafür werden intern drei Felder benutzt, in denen die zuvor in der Datei Config.pm konfigurierten Werte auftauchen: Unter „certificateRequest“ wird der Zertifikat-Antrag im PEM-Format abgelegt, unter „portalURL“ wird die Adresse des Portals angegeben, an die das generierte Zertifikat gesendet werden soll und unter „portData“ finden sich automatisch generierte Information für eine spätere Validierung.

Nach Anklicken des Buttons muss sich der Nutzer, falls noch nicht geschehen, spätestens jetzt per Shibboleth-Login mit seinem Nutzernamen und Passwort authentifizieren. Nach erfolgreicher Authentifizierung gelangt der Nutzer schließlich auf die Seite des Perl-Scripts portalLogin.cgi (2). Dort wird neben den bisher erwähnten Feldern ein zusätzliches Feld mit einem verschlüsselten Token hinzugefügt. Durch das Token soll sichergestellt werden, dass der Nutzer der Weiterleitung zugestimmt hat und nicht von einem schadhaften Portal weitergeleitet wurde. Mit diesem Formular wird der Zertifikat-Antrag an das Script genereateCred.cgi weitergeleitet (3) und von dort an die eigentliche Online-CA (4), wo das Zertifikat erstellt und an das Perl-Script generateCred.cgi zurückgegeben wird. Dort stehen in folgenden drei Feldern Informationen zur Verfügung: Das Feld „status“ kann den Wert „success“ oder „rejected“ enthalten, im Feld „certificate“ ist das X.509 Zertifikat im PEM-Format abgelegt und im Feld „portData“ finden sich die Informationen zu Validierung aus dem ersten Schritt.

Im letzten Schritt (6) sendet der Nutzer das Zertifikat zurück an das Portal, wo das Zertifikat dem zuvor erstellten Schlüsselpaar zugeordnet wird, indem der Inhalt des Feldes „portData“ ausgewertet wird. Zudem wird geprüft, ob der Modulus des privaten Schlüssels und der Modulus des zurückgegebenen Zertifikates übereinstimmen.

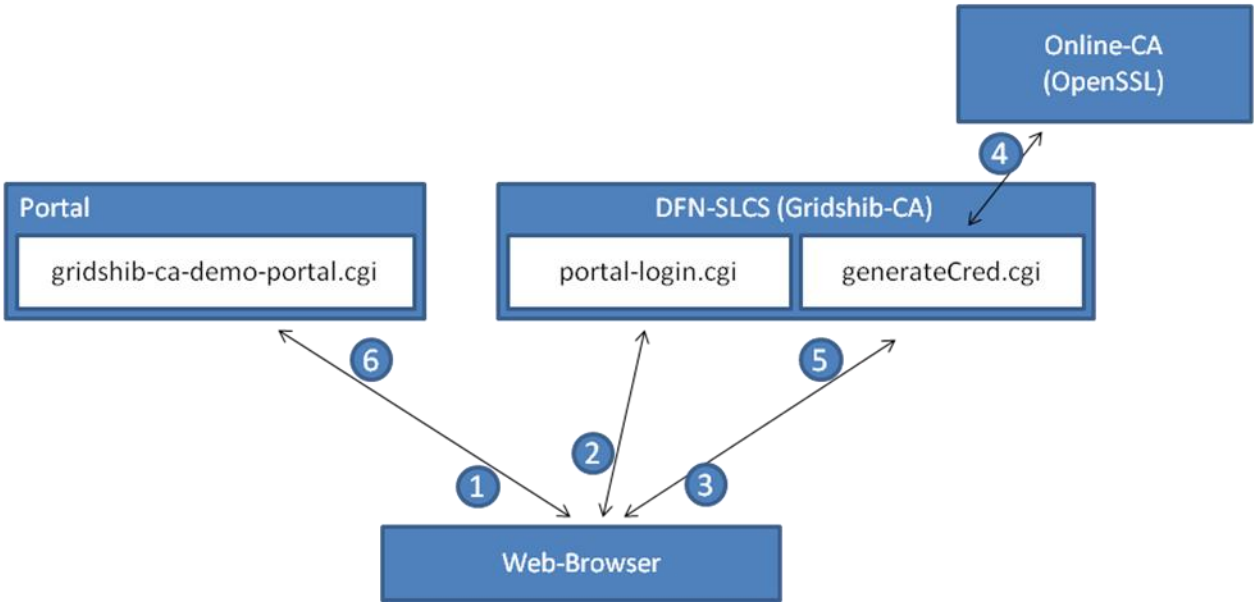

Abbildung 4: Beteiligte Komponenten bei Portal Delegation

3.2.3 Portierung des Perl-Scripts als Java-Servlet bzw. Java-Portlet

Da die im C3-Grid eingesetzte Portal Software auf Java-Technologie basiert wurde in einem zweiten Schritt das Script gridshib-ca-demo-portal.cgi in ein Java-Servlet portiert. Dieses Servlet kann in einen Tomcat-Server deployed werden. Wie das originale CGI-Script auch stellt es eine Web-Oberfläche bereit, von der aus ein kurzlebiges Zertifikat bezogen werden kann. Tests mit einem solchen Servlet sind mit wenig Aufwand durchführbar, bevor die endgültige Version dann als Portlet ins C3-Grid Portal integriert werden kann.

Mit diesem Servlet ist es möglich, analog zur Perl-Implementierung, im Testbed ein kurzlebiges Zertifikat zu beziehen. Zudem wird auch gleich automatisch ein Proxy vom SLC abgeleitet, das dann zur Authentifizierung und Autorisierung im Grid zur Verfügung steht.

Zusätzlich wurde eine Version erstellt, die, im Gegensatz zu der Perl-Variante, die privaten Schlüssel zum SLC nicht auf der Festplatte des Servers/Portals speichert, sondern nur innerhalb der Session im Arbeitsspeicher vorhält.

Dies bedingt eine kleine Änderungen zum in Kapitel 3.2.2 dargestellten Ablauf: Wie beschrieben wird durch das Perl-Script zunächst ein Schlüsselpaar erstellt und auf der Festplatte abgelegt. Da es bei Verwendung von Perl-CGI-Scripts keine Session Verwaltung gibt, wird der Dateiname, unter dem das Schlüsselpaar abgelegt wurde, zusammen mit dem Zertifikatrequest über den Parameter „portData“ an den DFN-SLCS gesendet. Im letzten Schritt, bei dem das fertige SLC zurückgesendet wird, wird diese Information wiederum mit gesendet, um das SLC dem richtigen Schlüsselpaar zuzuordnen.

Da im Gegensatz dazu bei der Verwendung von Servlets (bzw. in ein Portal integrierte Portlets) auf eine Sessionverwaltung zurückgegriffen werden kann, wird diese Information nicht mehr unbedingt benötigt. Außerdem gibt es auch keinen Dateinamen mehr, der dafür

verwendet werden könnte. Daher wird das Feld „portData“ in der aktuellen Version nicht mehr verwendet, sondern nur noch mit Dummy-Daten gefüllt.

Eine Überprüfung, ob das zurückerhaltene Zertifikat tatsächlich zum privaten Schlüssel passt findet noch immer statt, indem der Modulus der Zertifikats mit dem des privaten Schlüssels verglichen wird.

Die Software steht unter http://aforge.awi.de/gapslc zum Download bereit.

Die Hinweise zur Installation finden sich in dem dort abgelegten README File.

3.2.4 Integration in C3-Grid-Portal

Auf Basis des Java-Servlets wurde ein Portlet erstellt und in das C3-Grid-Portal eingebunden. Damit stehen nun kurzlebige Zertifikate und davon abgeleitete Proxy-Zertifikate am Portal zur Verfügung. Derzeit wird daran gearbeitet, von diesem SLC abgeleitete Proxies auch an den folgenden Stationen im Grid-Workflow, wie z.B. dem Daten Informations Service, Daten Management Service, Datenprovider, etc. nutzbar zu machen. Ziel ist die vollständige Integration in C3-Grid. Darauf aufbauend soll später eine SAML Assertion mit Campus- und VO-Informationen zusammengestellt und mit gesendet werden. An den Ressourcen Providern wird dann eine feingranulare Autorisierung anhand dieser zusammengestellten Informationen möglich (vgl. Abbildung 5).

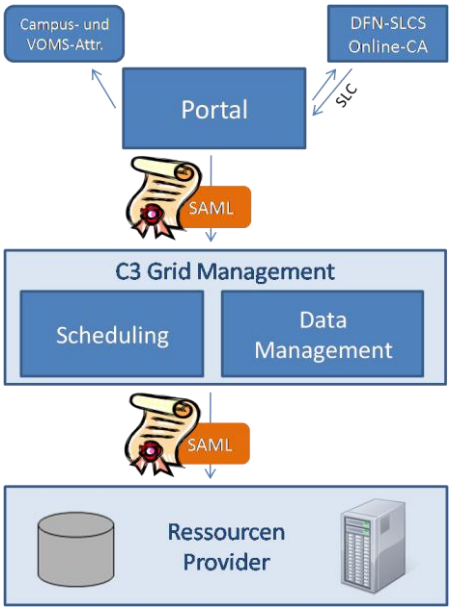

Abbildung 5 Integration in C3-Grid

Folgender Ablauf ist geplant:

1. Authentifizierung am Portal per Shibboleth.
2. Portal bezieht ein SLC vom DFN-SLCS.
3. Aus Campus- (und VOMS-) Attributen wird eine SAML Assertion zusammengestellt und durch das Portal signiert.
4. Proxy-Zertifikat wird abgeleitet (SAML Assertion wird dabei in Proxy integriert)
5. Proxy (+SAML) werden durch Scheduling und DMS an die RP weitergereicht.
6. Bei den RP: Autorisierungsentscheidung anhand der Informationen aus der SAML Assertion.

4 Nutzung des SLCS in TextGrid über Rich Client (TG-lab)

4.1 Konzept

In TextGrid wird ein Eclipse-basierter Rich Client, das „TextGridLab“ benutzt, der eine web-basierte Authentifizierung implementiert [7]. Deshalb ist das Szenario „Portal Delegation“ (vgl. Abbildung 3) auch hier anwendbar. Das SLC wird in der TextGrid-Middleware gespeichert und mittels eines Security-Tokens, der TextGrid SessionID (SID), referenziert. Bei allen vom TextGridLab initiierten Operationen – sowohl auf Ressourcen (TG-crud) als auch bei der Interaktion mit der Autorisierungskomponente (TG-auth*) – wird diese SID verwendet.

Die Vision von TextGrid ist – unter Anderem – eine möglichst einfache Kooperation von textbasiert arbeitenden Wissenschaftlern, die gemeinsam an im Grid befindlichen Ressourcen arbeiten. Dies wird ermöglicht durch ein feingranulares Rechtesystem (Role-based Access Control) und eine Organisation von Ressourcen in Projekten. Zu jeder zu einem Projekt gehörenden Ressource können einzelne Rechte (z.B. Lesen, Schreiben) an Rollen vergeben werden. Benutzer können wiederum in Rollen eingetragen werden. Dieses Konzept beschreibt u.A. einen Weg, wie dieses Rechtesystem auch auf dem Grid-Rechner umgesetzt werden kann, damit etwa die Mitarbeiter eines Projekts auf gemeinsame Ressourcen zugreifen können, obwohl sich diese in den Home-Verzeichnissen der einzelnen Besitzer befinden.

Um erste Erfahrungen zu sammeln, wurde – ähnlich wie bei C3-Grid – zunächst ein nicht-akkreditierter Dienst verwendet. Der Ablauf ist in Abbildung 5 wiedergegeben und wird im Folgenden beschrieben.

4.2 Technische Umsetzung

4.2.1 Testumgebung

Das System besteht aus den in Abbildung 5 gezeigten Komponenten:

1. Dem Eclipse-basierten Rich Client (TextGridLab), der eine web-basierte Authentifizierung ermöglicht (auf einem beliebigen Benutzerrechner)
2. Dem von einem Shibboleth Service Provider geschützten Portal (eine virtueller 64-bit OpenSuSE 11.0 Rechner mit Apache 2.2.8). Diese Komponente interagiert mit
    a) Dem Shibboleth Identity Provider der Heimatorganisation, bei der der Benutzer seinen Account hat (zum Bezug von Authentifizierungsinformation und Attributen)
    b) Dem gleichfalls von einem SP geschützten SLCS des DFN (zum Bezug des SLCs)
3. Einem Rechner (OpenSuSE 10.1 64 bit), der eine Kopie der TextGrid-Middleware enthält mit den Komponenten
    a) TG-crud – Ein Axis2 Web Service zum Anlegen, Lesen, Aktualisieren und Löschen von Dateien, der den Zugang zum Grid mittels JavaGAT kapselt
    b) TG-auth* - ein PHP-basierter Web Service zur rollenbasierten Autorisierung (openRBAC)

c) Einem eigens für die Verwendung mit SLCs entwickelten Synchronisierungsskript, das RBAC-Operationen in UNIX-Access Control Lists übersetzt (Perl)

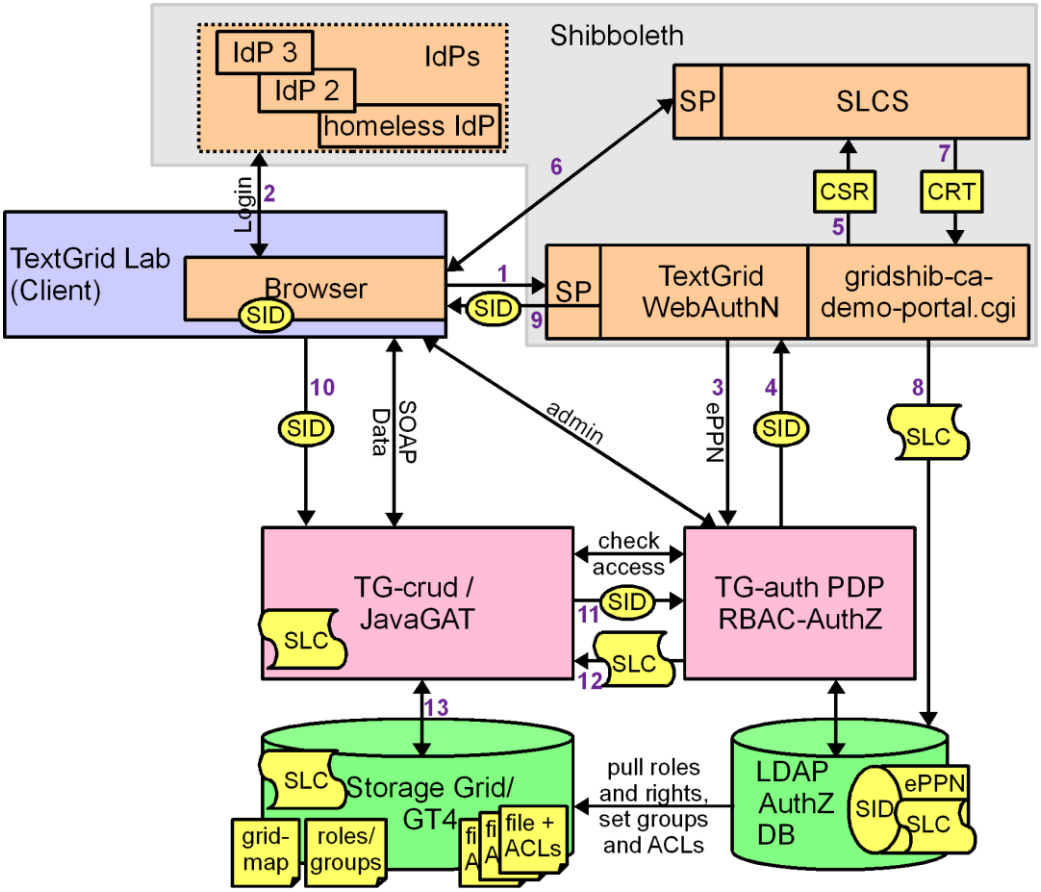

Abbildung 6: Bezug von SLCs per Portal Delegation bei Verwendung des Richt Clients von TextGrid

4.2.2 Vorgehensweise beim Bezug von SLCs

Über den im Rich Client integrierten Web-Browser wird eine zentrale Authentifizierung über Shibboleth ermöglicht. Nach Zugriff auf die durch einen Shibboleth Service Provider geschützte Web-Authentifzierungs-Ressource (Schritt 1), Auswahl des IdPs (nicht im Bild) und Login (2) gelangt der Benutzer zum ersten Teil des Portals. Dort wird die Benutzerkennung (eduPersonPrincipalName) vom SP übernommen und an die zentrale TextGrid-Autorisierungskomponente TG-auth geschickt (3). TG-auth schickt über eine verschlüsselte Verbindung eine eindeutige SessionId (SID, gelb) zurück (4), die als Sicherheitstoken fungiert.

Nun beginnt der Vorgang der *Portal Delegation*. Hierzu wurde eine angepasste Version des Skripts gridshib-ca-demo-portal.cgi verwendet. Die Modifikationen beschränkten sich dabei im Wesentlichen auf die Anpassung an die Umgebung (Endpoints, verwendete Bibliotheken), das Durchreichen der SID, und das Verschicken des SLC samt privatem Schlüssel im PKCS#12-Format an die Autorisierungsdatenbank. Im Einzelnen wird ein privater Schlüssel samt Zertifikatsrequest generiert, letzterer wird an den SLCS-Service geschickt (5). Der SLCS ist ebenso durch einen SP geschützt, so dass er auch auf die bereits erfolgte Authentifizierung, sowie auf Benutzer-Attribute des IdP zugreifen kann (6).

Nach einer gewissen Zeit antwortet der SLCS mit dem Zertifikat (7). Ist nach den notwendigen Bestätigungen durch den Benutzer der Vorgang der Portal Delegation beendet, wird zunächst das SLC in der Autorisierungdatenbank gespeichert (8). Nun wird die SID an den Rich Client zurückgegeben (9) und der Benutzer kann mit seinem persönlichen SLC im Grid arbeiten. Allerdings wird das SLC nicht direkt im System des Benutzers verwaltet, sondern es wird die SID als Referenz zum SLC des Benutzers verwendet. Das TextGridLab speichert die SID und verwendet sie in allen Anfragen an TextGrid-Dienste (10). Der zentrale TG-crud-Dienst löst die SID-Referenz zum SLC auf (11,12) und verwendet das zugehörige SLC bei allen Grid-Zugriffen (13).

Wegen aktuellen Performanzproblemen beim Bezug des SLC (s.u.) wurde vor der Portal Delegation (5) eine Option eingebaut, damit der Benutzer entscheiden kann, ob er mit einem SLC arbeiten möchte. Wünscht er dies nicht, wird die SID sofort an den Rich Client zurückgegeben (9), der sie fortan benutzen kann, und der Authentifizierungsvorgang ist beendet. Wenn in diesem Fall TG-auth ein noch gültiges SLC dieses Benutzers vorfindet, wird dieses weiterverwendet, auf die Gefahr hin, dass es im Laufe seiner Sitzung abläuft. So muss bei der voreingestellten Gültigkeitsdauer von 12 Stunden ein Benutzer nur am Beginn seines Arbeitstags tatsächlich das SLC beziehen und kann später das vorhandene Zertifikat verwenden, falls ein erneutes Login im TextGridLab nötig sein sollte.

4.2.3 Abbildung von TextGrid-Rollen in der Grid-Umgebung

Um die rollenbasierte Policy von TG-auth* nachzubilden, wurde auf dem Grid-Rechner ein Skript installiert, welches regelmäßig Rollen und Berechtigungen auf Ressourcen aus der Autorisierungsdatenbank abgreift.

Rollen werden als UNIX-Gruppen auf dem Grid-Rechner modelliert. Ist ein Benutzer mit seinem ePPN für eine Rolle registriert, wird im Grid dessen User-ID in die entsprechende Gruppe eingetragen. Wird eine Ressource angelegt, wird über die erweiterten POSIX-ACLs (Access Control Lists) der Zugriff für die Rollen/Gruppen modelliert, so dass die Situation der Berechtigungen aus der TextGrid-Autorisierungdatenbank wiedergespiegelt wird. Somit können die TextGrid-Daten im Home-Verzeichnis eines jeden Benutzers (dem Ersteller der Ressource) abgelegt werden und andere Projektmitglieder haben dennoch Zugriff gemäß ihren Rechten in TextGrid. Die TextGrid-Projekte werden als Verzeichnisse unterhalb des Benutzer-Home-Verzeichnisses modelliert und Ressourcen als darin befindliche Dateien.

4.2.4 TG-crud

Der TG-crud Service (Create/Retrieve/Update/Delete) wurde an das SLC-Scenario wie oben beschrieben angepasst. Um den Nutzern von SLCs Zugriff auf Dateien anderer Nutzer – nach ihren jeweiligen Rollen in TextGrid – zu gewähren, wird momentan der absolute Pfad der Dateien (incl. Hostnamen des Grid-Hosts) im RBAC abgelegt. Zukünftig soll dies über die logicalFiles des Globus Toolkit erfolgen, die eine logische Adresse als Identifikator haben und mehrere physikalische Orte für eine Datei beinhalten, an denen die Datei vorliegt. Dies ist momentan mit dem auf dem Testsystem installierten Globus Toolkit 4.2.1 noch nicht möglich, da die entsprechenden Adaptoren für das GT4.2.1 seitens JavaGAT noch nicht existieren. Die logische Adresse für eine TextGrid-Datei wäre sinnvollerweise die TextGrid-URI, die ein jedes TextGrid-Objekt besitzt. Eine Vorhaltung

von absoluten Pfaden wäre dann in TG-auth* oder TG-crud nicht mehr nötig. Mit diesen logischen Dateien ist ein ReplicaManagement in der TextGrid Produktivumgebung bereits implementiert (mit Globus Toolkit 4.0.8). [8][9]

Generell bleibt noch die Frage offen, wie die Nutzer der SLCs in das grid-mapfile aufgenommen werden, in dem die Zertifikatinhaber per DN auf die lokalen Nutzeraccounts gemapped werden, in denen dann gearbeitet wird, und wer die lokalen Nutzer und deren Home-Verzeichnisse bei einem ersten Zugriff anlegt.

4.3 Nutzung eines akkreditierten SLCS

Es wird mittelfristig angestrebt, und in einer Folgeversion dieses Dokuments spezifiziert, dass auch in TextGrid eine akkreditierte Version des SLCS verwendet wird. Es ist insbesondere unschön, dass das SLC (samt privaten Schlüssel) in Schritt 8 über das Netz wenn auch verschlüsselt transportiert wird. Dies sollte generell gelöst werden. Hier einige Schritte, die dafür noch zu tun sind:

- Umbau der Architektur:
    1. Der private Schlüssel des SLC darf in einem akkreditierten System nicht auf einer zentralen Komponente liegen, also weder auf dem Portal noch in der Autorisierungsdatenbank. Stattdessen sollen davon abgeleitete (kürzer gültige) Proxies auf einem MyProxy-Server bereitgestellt werden.
    2. Der Credential Retriever (CR, eine Java-WebStart-Anwendung) kommuniziert mit dem SLCS. Es ist zu prüfen, ob der CR auch in jedem Fall mit dem eingebetteten Browser des TextGridLab startbar ist. Dies funktionierte bereits mit einem im TextGridLab mitgelieferten Xulrunner v. 1.8; in der aktuellen Entwicklungslinie wird aber der im System des Benutzers vorhandene Browser eingebettet. Hier muss WebStart noch getestet werden.
    3. Idealerweise sollte der CR auch das Hochladen des Proxys auf den myProxy-Server besorgen, wie im DFN-Szenario „Grid Portal“ vorgesehen.
- Die Benutzer von TextGrid werden Shibboleth Identity Provider an ihren Heimatorganisationen benötigen, die die erhöhten Sicherheitsanforderungen, wie durch die EUGridPMA-Akkreditierung gefordert, umsetzen.
- Es sollten Anstrengungen unternommen werden, die Performanz des SLCS zu erhöhen: gegenwärtig dauert ein Loginvorgang bei Bezug eines SLCs samt aller Benutzereingaben und -bestätigungen ca. 1 bis 2 Minuten, was bei TextGrid-Nutzern zu Akzeptanzproblemen führen könnte.

4.4 Produktive Nutzung von SLCs in TextGrid

Es ist zu klären, ob auch innerhalb von TextGrid sowohl Ressourcen mit hohen Sicherheitsanforderungen als auch solche mit niedrigen eingerichtet werden sollen. Entsprechend kann dem Benutzer die Wahl bleiben, ob er den unter Umständen komplizierteren Bezug eines SLC für notwendig erachtet – um somit auf Resourcen mit hohen Sicherheitsanforderungen zugreifen zu können, oder ob ihm für sein Vorhaben der Zugriff auf Ressourcen mit niedrigen Sicherheitsanforderungen genügt

5 Zusammenfassung

Mithilfe des CGI-Scripts gridshib-ca-demo-portal konnten SLC, geschützt durch einen Shibboleth-Login, bezogen werden. Durch die erfolgreiche Portierung in JAVA steht diese Funktionalität nun auch im C3-Grid-Portal zur Verfügung. Auch in TextGrid konnte die Kombination Shibboleth-Authentifizierung und SLC erfolgreich implementiert werden.

Um vorerst auch ohne Proxy-Upload-Tool arbeiten zu können, wurde zunächst in beiden Nutzungs-Szenarien ein pragmatische Lösungsansatz gewählt. Problematisch an dieser Lösung ist jedoch, dass der private Schlüssel des Users direkt am Portal, bzw. in der TG-Middleware abliegt. Dies stellt ein gewisses Risiko dar, da durch eine Kompromittierung des Portals/der Middleware die privaten Schlüssel der Nutzer in fremde Hände geraten könnten.

Zu beachten ist dabei auch, dass eine Verwendung in Zusammenhang mit dem durch EUGridPMA akkreditierten Short Lived Credential Service ausgeschlossen ist, da die Policies verbieten, den privaten Schlüssel, wie in unserem Fall, am Portal bzw. auf einem Server abzulegen. Diese Tests wurden daher allesamt mit dem nicht akkreditierten SLCS durchgeführt.

Da mit dieser Technik nun kurzlebige Zertifikate, bzw. die davon abgeleiteten Proxies am Portal zur Verfügung stehen, kann die Integration in den weiteren Workflow beginnen und damit Erfahrungen im Einsatz von kurzlebigen Zertifikaten gesammelt werden. Für die weitere Verwendung der Proxies im Grid ist es zudem unerheblich, wie diese zum Portal gelangt sind. Wenn das Proxy-Upload-Tool zur Verfügung steht und die privaten Schlüssel damit nicht mehr am Portal abgelegt werden müssen, kann die dann vorhandene Integration in den Komponenten weiter genutzt werden; dann natürlich auch mit den akkreditierten SLC.

6 Literatur

[1] Verwendung von Zertifikaten im D-Grid, Deliverable DGI-2, Fachgebiet 3.1, Koordination und Sicherheitsmanagement http://dgi2.d-grid.de/fileadmin/user_upload/documents/DGI2-FG3/FG3-1/DGI-2_FG-3.1_Zertifikate_im_D-Grid_v10.pdf

[2] SWITCHHaai, http://www.switch.ch/aai/

[3] UK Access Management Federation for Education and Research, http://www.ukfederation.org.uk/

[4] http://www.dfn.de/dienstleistungen/dfnaai/

[5] Certificate Policy and Certification, Practice Statement of the Public Key Infrastructure in the Deutsche Forschungsnetz, http://www.pki.dfn.de/fileadmin/PKI/DFN-PKI_SLCS-CPCPS_v11.pdf

[6] http://gridshib.globus.org/docs/gridshib-ca-1.0.0/admin/portal-delegation.html

[7] TextGrid-Architektur, http://www.textgrid.de/fileadmin/TextGrid/reports/TextGrid_Report_3_2.pdf

[8] TextGrid Manual: Tool Development, http://www.textgrid.de/fileadmin/TextGrid/reports/R3_5-manual-tools.pdf

[9] TextGrid – Installation einer Datengrid-Knotens, http://www.textgrid.de/fileadmin/TextGrid/reports/TextGrid_Report_3_6-Datengrid-Knoten.pdf