



Fachgebiet 3.2
AAI/VO

Werkzeugunterstütztes Management virtueller Organisationen im D-Grid

Autoren

Michael Schiffers (LMU/LRZ)

Wolfgang Ziegler (Fraunhofer SCAI)

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IG07014 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt bei den Autoren.

Inhalt

1	MANAGEMENT SUMMARY.....	5
2	EINORDNUNG	5
2.1	Hintergrund	5
2.2	Definitionen	5
2.3	Vorgehensweise.....	5
3	ANFORDERUNGEN AN DIE TOOL SUITE	6
3.1	Referenzinstallation.....	6
3.2	Zentrale Dienste.....	6
3.2.1	Kern-D-Grid.....	6
3.2.2	DFN-Verein	6
3.3	Rollen im D-Grid VO-Management	6
3.3.1	Administratoren der lokalen D-Grid Ressourcen (Resourceadmin)	7
3.3.2	D-Grid Operation Center (Operator).....	7
3.3.3	VO-Repräsentant (VO-Manager).....	8
3.3.4	Community Sprecher (VO-Initiator)	9
3.3.5	D-Grid Software-Administrator (Softwareadmin)	9
3.3.6	D-Grid Daten-Administrator (Dataadmin)	10
3.3.7	D-Grid Entwickler (Developer)	10
3.3.8	VO-Mitglieder	10
3.4	Interaktionen zwischen den Rollen	11
4	BASISARCHITEKTUR DER SUITE	11
4.1	Grundkonzept.....	11
4.2	Building Blocks.....	12
5	FEASIBILITY-STUDIEN ZUR SUITE-ENTWICKLUNG	12
5.1	Managebarkeit zustandsbehafteter Web Service Resources mit Apache Muse.....	12
5.1.1	Ziele	12
5.1.2	Bewertung.....	13
5.2	Vereinheitlichung von VO-Managementschnittstellen über heterogene Grids	14
5.2.1	Ziele	14
5.2.2	Fazit	14
5.3	Handhabung von Attributen in heterogener Grid Middleware.....	15
5.3.1	Realisierung der Shibboleth Integration durch GridShib CA und UNICORE 5 plug-in....	16
5.3.2	Realisierung der VOMS Integration durch eine Keystore-Erweiterung	18
5.3.3	Weitere Ansätze zur Handhabung von Benutzerattributen	19
5.4	Zusammenfassung.....	19
6	REALISIERUNG DER TOOL SUITE	20
7	DEPLOYMENTASPEKTE	20
8	FAZIT UND EMPFEHLUNGEN	20

1 Management Summary

Um die Heterogenität von Ressourcen, Diensten, Sicherheitsniveaus und Policies im D-Grid-Betrieb zu verschatten, ist neben einer VO-Managementarchitektur, die von diesen Spezifika abstrahieren kann, ein „Werkzeugkasten“ (*tool suite*) erforderlich, der eine solche Abstraktion unterstützt und das Management virtueller Organisationen im D-Grid erleichtert. Ziel dieses Dokuments ist die Beschreibung eines Rahmenwerkes einer solchen Werkzeugunterstützung. Die von den D-Grid-Communities geforderte generische VO-Struktur und ein dazu passendes VO-Management-Rahmenwerk sind Gegenstand eines früheren Berichts [generischeVO] gewesen, der die Grundlage des hier vorgestellten Rahmenwerkes bildet. Die Zielumgebung (*target environment*) des Rahmenwerkes wird durch die für das D-Grid definierte Referenzinstallation gegeben, durch die im Kern-D-Grid betriebenen und bereitgestellten zentralen Dienste wie VOMRS, VOMS und durch den vom DFN betriebenen Short Lived Credential Service (SLCS).

Sämtliche Realisierungsaspekte werden in der finalen Version dieses Berichtes zum Meilenstein 12 dargestellt.

2 Einordnung

2.1 Hintergrund

Die zunehmende Dynamik Virtueller Organisationen sowohl im Innenverhältnis (Fluktuation von Mitgliedern, Fluktuation von Rollen) als auch aus der Sicht der Communities (Fluktuation von VOs) kann in Zukunft nicht mehr manuell administriert werden. Stattdessen wird für das Management virtueller Organisationen eine adäquate „Management Tool Suite“ erforderlich, die einerseits Administratoren, VO-Manager, Grid Operations Centers und VO-Initiatoren in ihren entsprechenden Sichten auf das VO-Management unterstützt, andererseits aber nahtlos ins D-Grid mit seinen heterogenen Middleware-Strukturen „deployed“ werden kann. Einige Einzelfragen wurden zwar im Kontext früherer Projekte ([rahmenkonzept], [betriebskonzept], [ivom]) untersucht. Das Ziel der hier betrachteten Aufgabenstellung ist jedoch umfassender und Infrastruktur-näher.

2.2 Definitionen

Im Rahmen dieses Dokuments wird unter „**D-Grid**“ die in der Referenzinstallation [referenzinstallation] beschriebene Infrastruktur verstanden und unter „**Kern-D-Grid**“ die zentralen Dienste [kerndgrid], die von den D-Grid-Installationen genutzt werden können.

Unter „**Tool Suite**“ wird ein Middleware-unabhängiges technisches Rahmenwerk für die Attribut-basierte Autorisierung im Kontext von Referenzinstallation und Kern-D-Grid verstanden.

2.3 Vorgehensweise

Die Entwicklung eines Management-Rahmenwerkes erfordert eine saubere Modellierung virtueller Organisationen und deren Interaktionsmechanismen sowohl nach innen als auch nach außen. Diese werden im Abschnitt 3 untersucht. Die Abbildung der Interaktionen und der an ihnen beteiligten Rollen auf die durch die Referenzinstallation und das Kern-D-Grid vorgegebene Infrastruktur geschieht im Abschnitt 4. Die eigentliche Entwicklung des Rahmenwerkes wird im Abschnitt 6 ausgeführt. Da diese primär auf den Standards des Web Services Stacks basieren wird, wird vorher im Abschnitt 5 dessen prinzipielle Anwendbarkeit diskutiert. Dazu werden die Ergebnisse einiger im Vorfeld durchgeführter Feasibility-Studien dargestellt. Abschnitt 7 befasst sich schließlich mit dem Deployment-Aspekt.

3 Anforderungen an die Tool Suite

3.1 Referenzinstallation

Das Ziel (*target environment*) für die nachfolgenden Ausführungen bildet die D-Grid Referenzinstallation [referenzinstallation], zunächst in ihrer aktuellen Version (zur Zeit Release 1, http://dgiref.d-grid.de/wiki/Release_1), später in der für Sommer 2009 geplanten Version 2. Im Release 1 sind das Globus Toolkit in der Version 4.0.7, UNICORE in der Version 5 und gLite in der Version 3.1 vertreten, auf der Datenmanagementseite OGSA-DAI 4.0.7 und dCache 1.9. Dedizierte Authentifizierungs- und Autorisierungswerkzeuge sind zwar vorgesehen, sie sind jedoch formal kein Bestandteil der Referenzinstallation, sie gehören stattdessen zum Kern-D-Grid.

3.2 Zentrale Dienste

Neben der Infrastruktur, die für die einzelnen D-Grid-Ressourcen durch die Referenzinstallation bereitgestellt wird, sind im D-Grid Dienste vorgesehen, die nicht lokal auf den Ressourcen installiert werden, sondern als zentrale Dienste von allen D-Grid Installationen genutzt werden. Derzeit werden diese Dienste an zwei Stellen bereitgestellt: im D-Grid selbst durch das Kern-D-Grid und außerhalb des D-Grid durch den DFN-Verein.

3.2.1 Kern-D-Grid

Im Kern-D-Grid werden zwei wesentliche Dienste betrieben, die für die Attribut-basierte Autorisierung auf D-Grid Ressourcen benötigt werden: der VO Membership Service (VOMS) und der VO-spezifische VO Management Registration Service (VOMRS). VOMS stellt auf Anforderung des Benutzers die für diesen Benutzer in einer Virtuellen Organisation definierten Attribute in einem Proxy-Zertifikat zur Verfügung. VOMRS dient der Definition und Pflege der Virtuellen Organisationen, der Mitglieder dieser Virtuellen Organisationen und der jeweils für die Virtuellen Organisationen definierten Attribute.

3.2.2 DFN-Verein

Der DFN-Verein betreibt seit 2008 eine Testinstallation für einen Short Lived Credential Service (SLCS). Dieser Dienst ermöglicht es, auch für D-Grid Benutzer, die nicht über Zertifikate der D-Grid PKI verfügen, sich kurzlebige Zertifikate ausstellen zu lassen. Insbesondere können diese SLC verwendet werden, um Benutzerattribute zum Zweck der Autorisierung zum jeweiligen Service Provider zu transportieren. Der DFN-Verein besitzt eine Akkreditierung seines SLCS durch die EUGridPMA¹ und wird nach eigenen Angaben noch in 2009 den Testbetrieb in einen Produktionsbetrieb überführen.

3.3 Rollen im D-Grid VO-Management

In [generischevo] und [attribute] wurden die für den Kontext dieses Berichtes erforderlichen Rollen und die notwendigen Attribute in Anlehnung an das VO-Rahmenkonzept [rahmenkonzept] und das D-Grid Betriebskonzept [betriebskonzept] identifiziert und deren technische Umsetzung dargestellt. Die wesentlichen organisatorischen Konzepte dahinter seien hier noch einmal kurz zusammengefasst. Die erforderlichen Rollen und die Aufgaben, die den verschiedenen Rollen zugeordnet werden, sind abhängig vom jeweiligen Kontext. Daher sind in der folgenden Übersicht die Beschreibungen der Rollen und Aufgaben nach den bisher identifizierten administrativen Ebenen im D-Grid gegliedert. Die Umsetzung im Kern-D-Grid und in der Referenzinstallation umfasst heute allerdings nur eine Untermenge der hier beschriebenen Aufgaben und Rollen². Eine schrittweise Umsetzung der in diesem Bericht

¹ www.eugridpma.org

² Stand Januar 2009

dargestellten Strukturen im Kern-D-Grid und der Referenzinstallationen wird in den nächsten beiden Jahren erwartet. Der aktuelle Stand wird im Folgenden entsprechend annotiert.

Es ist an dieser Stelle zu bemerken, dass für zahlreiche in diesem Kapitel beschriebene Aufgaben bisher weder Rollen noch Attribute für die Autorisierung definiert waren. Vielmehr findet die Autorisierung implizit durch Authentifizierung statt, d.h. die Identität eines Benutzers (abgebildet auf eine lokale Benutzerkennung) autorisiert diesen, bestimmte administrative Aufgaben durchzuführen. Beispiele sind die Pflege von VOs und Attributen im VOMRS Server, das Verwalten von Mitgliedern einer VO oder die Vergabe von Rechten lokaler Benutzer, auf die VO-Mitglieder abgebildet werden.

3.3.1 Administratoren der lokalen D-Grid Ressourcen (Resourceadmin)

Die lokalen Administratoren nehmen eine zentrale Aufgabe bei der Autorisierung der Benutzer auf den Ressourcen wahr, indem sie die Rechte lokaler Benutzerkennungen (userids) der Ressourcen definieren. Durch Abbildung von VO-Mitgliedern auf diese lokalen Benutzerkennungen werden die VO-Mitglieder mit den jeweiligen Rechten der Benutzerkennung ausgestattet, auf die sie abgebildet werden.

Rolle	Attribut	Aufgaben	Beispiel	Status
Administratoren der lokalen D-Grid Ressourcen	/<vo-name>/admin/ressourcenadmin	Unterstützung für Autorisierung auf Ressourcen	Rechte lokaler Benutzer auf die VO-member abgebildet werden konsistent vergeben Auskunftsdienst für VOs, deren Mitglieder auf lokale Ressourcen zugreifen können (sollen)	Konzeptphase
		Integration mit vorhandenen Verfahren der Autorisierung		Konzeptphase

3.3.2 D-Grid Operation Center (Operator)

Diese Ebene ist noch nicht in D-Grid implementiert, befindet sich aber in Planung und wird bis zur nächsten Version des Betriebskonzepts realisiert und im Betriebskonzept verankert werden.

Rolle	Attribut	Aufgaben	Beispiel	Status
D-Grid Operations Center	/<vo-name>/admin/operator	Autorisierung	Prozesse zum konsistenten Abbilden von VO-memberships auf lokale Nutzer	Konzeptphase
		Unterstützung der formalen		Konzept-

	Übereinkunft über Ressourcennutzung mit den VO-Initiatoren		phase
	Integration mit vorhandenen Verfahren der Autorisierung (VOMRS)		Konzeptphase

3.3.3 VO-Repräsentant (VO-Manager)

Das eigentliche Management einer D-Grid VO fällt in den Verantwortungsbereich des VO-Repräsentanten (siehe auch [workflow]). Für jede VO wird während des Gründungsprozesses durch den Community Sprecher ein VO-Repräsentant bestimmt, dem die Rolle *VO-Manager* zugeordnet ist. Diese Rolle autorisiert den VO-Repräsentanten die folgenden Aufgaben durchzuführen.

Rolle	Attribut	Aufgaben	Beispiel	Status
VO-Repräsentant	/vo-name/ admin/vo-manager	Verfahren für Akzeptanz/Ablehnung der VO-Richtlinien für D-Grid und bei Akzeptanz Bestätigung durch Community Sprecher		Konzeptphase
		Prozesse für Agreements über Ressourcennutzung		Konzeptphase
		Prozesse zur technischen VO-Gründung		Konzeptphase
		Unterstützung für Mitgliederverwaltung	Prozesse für Gruppenbildung Aufnahme von Mitgliedern Prüfung der Möglichkeit von Gruppenmitgliedschaften Prüfung VO-AUP Akzeptanz durch Mitglied Statusänderung von Mitgliedern Rollen- und Attributverwaltung Beenden der Mitgliedschaft	Konzeptphase
		Prozesse zur technischen VO-Auflösung		Konzeptphase

		Integration mit vorhandenen lokalen Verfahren des VO Management		Konzeptphase
--	--	---	--	--------------

3.3.4 Community Sprecher (VO-Initiator)

Die Community Sprecher haben im D-Grid die Aufgabe auf Anforderung aus der jeweiligen Community eine neue VO zu gründen oder eine VO aufzulösen, wenn sie nicht mehr gebraucht wird. Dafür wird dem Community Sprecher das Attribut *VO-Initiator* zugeordnet

Rolle	Attribut	Aufgaben	Beispiel	Status
Community Sprecher	/<vo-name>/ admin/vo-initiator	Unterstützung der formalen Schritte zur VO-Gründung	Autorisierung Übereinkommen mit den D-Grid Operation Center über Ressourcennutzung	Konzeptphase
		Unterstützung der formalen Schritte zur VO-Auflösung	Autorisierung Übereinkommen mit dem D-Grid Operations Center über Ressourcennutzung auflösen	Konzeptphase
		Gruppenstruktur in VO etablieren (wenn gewünscht)		Konzeptphase
		Integration mitzeitigem Verfahren der VO Initiierung/Beendigung		Konzeptphase

3.3.5 D-Grid Software-Administrator (Softwareadmin)

Diese Rolle wurde ist neu und wurde auf dem Workshop am 12.1.09 mit dem Kern-D-Grid und der Referenzinstallation vereinbart.

Rolle	Attribut	Aufgaben	Beispiel	Status
D-Grid Software-Administrator	/<vo-name>/ admin/softwareadmin	Verwaltung von Software im Verzeichnis seiner VO	Installation, Upgrade, Deinstallation	In Vorbereitung

3.3.6 D-Grid Daten-Administrator (Dataadmin)

Diese Rolle wurde ist neu und wurde auf dem Workshop am 12.1.09 mit dem Kern-D-Grid und der Referenzinstallation vereinbart.

Rolle	Attribut	Aufgaben	Beispiel	Status
D-Grid Daten- Administrator	/<vo-name>/admin/dataadmin	Verwaltung von Daten auf Speicherressourcen im Bereich seiner VO		In Vorbereitung

3.3.7 D-Grid Entwickler (Developer)

Diese Rolle wurde ist neu und wurde auf dem Workshop am 12.1.09 mit dem Kern-D-Grid und der Referenzinstallation vereinbart.

Rolle	Attribut	Aufgaben	Beispiel	Status
D-Grid Entwickler	/<vo-name>/[member/]developer	Entwicklung von Software für eine VO	Zugang zur Express-Queue (falls auf der entsprechenden Ressource implementiert)	In Vorbereitung

3.3.8 VO-Mitglieder

Auf der Ebene der VO-Mitglieder gibt es kein Dienste für die Verwaltung von VOs oder VO Benutzern bzw von Software oder Daten dieser VOs. Die einzige Funktionalität, die auf dieser Ebene implementiert wird, ist ein Dienst, der es einem VO-Mitglied erlaubt, sich über die für dieses Mitglied gespeicherten Attribute zu informieren.

Rolle	Attribut	Aufgaben	Beispiel	Status
VO-Mitglied	/<vo-name>/member/	Nutzung des D-Grid	Zugang zu Ressourcen abhängig von den Attributen Nutzung des Auskunftsdienstes für Zugriff auf die eigenen Daten	Vorhanden

3.4 Interaktionen zwischen den Rollen

Die Interaktionen zwischen diesen Rollen dienen vorzugsweise der Gründung virtueller D-Grid-Organisationen und der Aufnahme neuer Mitglieder. Sie sind in [rahmenkonzept], [betriebskonzept] und [workflow] ausführlich beschrieben.

4 Basisarchitektur der Suite

4.1 Grundkonzept

Um den im Abschnitt 4 dargestellten Anforderungen zu genügen, wird die Basisarchitektur der Suite in ein klassisches Schichtenmodell gemäß Abbildung 1 eingebettet.

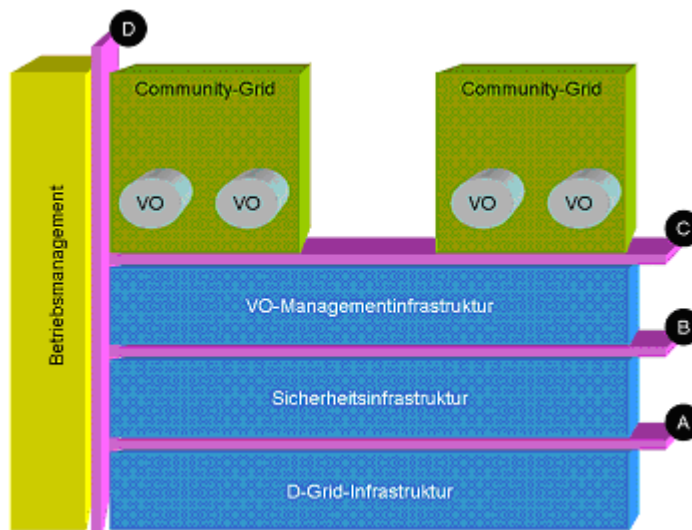


Abbildung 1: Allgemeines Schichtenmodell zum VO-Management

Die dabei auftretenden Schnittstellen A, B, C und D werden wie folgt bedient:

Schnittstelle A:

Schnittstelle A wird durch die in der Referenzinstallation vorgegebene und im IVOM-Projekt [ivom] bzw. Folgeprojekten des dritten Calls erweiterte D-Grid-Middleware bedient. Eine Implementierung der Schnittstelle A wird deshalb hier nicht betrachtet.

Schnittstelle B:

Das VO-Management verwendet wesentliche Funktionalitäten der D-Grid-Sicherheitsinfrastruktur über die Schnittstelle B. Einige der an dieser Schnittstelle erforderlichen Funktionalitäten sind zwar schon im IVOM-Projekt [ivom] diskutiert worden (bzw. werden in Folgeprojekten des dritten Calls diskutiert werden), erforderliche Erweiterungen werden weiter unten untersucht.

Schnittstelle C:

Das VO-Management stellt den Communities für deren VO-Management Dienste an der Schnittstelle C zur Verfügung. Die Tool Suite bedient Schnittstelle C.

Schnittstelle D:

Orthogonal dazu stehen die Deployment- und (Betriebs-) Management-Dienste, die die für ein Attribut-basiertes VO-Management entwickelten Lösungen dem operativen Betrieb zuführen und spezifische Management-Dienste bereitstellen. Die Tool Suite bedient Schnittstelle D.

4.2 Building Blocks

Aus den Ausführungen im Abschnitt 3 ergeben sich grundsätzliche Anforderungen an die Building Blocks der Tool Suite-Architektur:

Der aktuelle Kontext des Grid-Systems muss erkannt werden.

Um die Heterogenität der D-Grid-Komponenten auszugleichen, sind Transformatoren notwendig, die Tool Suite Kommandos und Komponentennachrichten Middleware-spezifisch anpassen. Erforderlich sind diese für das Globus Toolkit 4.0.x und 4.2, für UNICORE 5 und 6 und für gLite 3.1.

Ebenso sind Kommandos an und Nachrichten von Kern-D-Grid-Diensten zu behandeln. Dies betrifft (SAML-)VOMS, VOMRS und GRRS.

Komponenten für die Integration von Shibboleth sind erforderlich (Identity Provider, WAYF, GridShib). Dies bedeutet auch die Erweiterung des eduPerson-Schemas um D-Grid-Attribute (siehe [ivom]).

Die Verwendung des DFN SLCS muss automatisiert werden.

Eine daraus resultierende Software-Architektur wird in die finale Version dieses Dokumentes im Rahmen des Meilensteins 12 integriert.

5 Feasibility-Studien zur Suite-Entwicklung

Im Rahmen des Projektes und des Projektvorfeldes wurden mehrere Arbeiten durchgeführt, um die Machbarkeit der hier angedachten Management-Suite auszuloten. Die wesentlichen Ergebnisse dieser hier als *Feasibility-Studien* deklarierten Arbeiten sollen im folgenden kurz dargestellt werden.

5.1 Managebarkeit zustandsbehafteter Web Service Resources mit Apache Muse

5.1.1 Ziele

Diese Arbeit ist im Rahmen eines Fortgeschrittenen-Praktikums an der LMU entstanden [hoeber]. Ziel der Arbeit war die Überprüfung des Web Services Distributed Management Frameworks [wsdm] für dessen grundsätzliche Eignung als Implementierungsmodell einer Management-Suite. Die Arbeit analysiert konkret eine WSDM-Implementierung durch Apache Muse [muse]. Als Testbeispiel diente ein Apache HTTP Web Server, der über WSDM gemanagt wurde (Server Start, Server Stopp, Server-Konfiguration, Event Notification).

Web Service Distributed Management (WSDM) ist ein Projekt des OASIS WSDM Technical Committee mit dem Ziel eines Web Service basierten Managements von Web Services (Management of Web Services (MUWS)) bzw. durch Web Services (Management using Web Services (MOWS)). Dazu beschreibt WSDM die Management Informationen und die so genannten Capabilities für bestimmte Geräte, Anwendungen und Komponenten mit Hilfe der Web Service Description Language (WSDL).

Das Muse Project von Apache ist eine Java-basierte Implementierung der WSDM-Teilspezifikationen WS-ResourceFramework [wsrf] und WS-BaseNotification [wsn]. Muse erlaubt die Entwicklung von Web Service Interfaces für Management-Zwecke, ohne dass sich der Entwickler mit den Details der Spezifikationen näher auseinander setzen muss.

Apache Muse bietet folgenden Funktionsumfang:

- Implementierung der Spezifikationen WSRF 1.2, WSN 1.3, WSDM 1.1 und WS-Mex Port Typen

- Standalone-Implementierung des WSDM Event Formats 1.1
- Kompatibilität mit WS-Addressing 1.0 und SOAP 1.2
- WSDL-to-Jave Code Generierungs-Tool
- Generierung von WS-Resource- und Client-seitigem Code anhand von WSDL

Da die Entwicklung von WSDL Dokumenten nicht trivial ist, wird in Muse ein WSDL Template angeboten, das bereits eine Vielzahl von Properties und Operationen beinhaltet, die nur noch angepasst oder erweitert werden müssen. Anhand dieses WSDL-Templates werden mittels `wsdl2java` der benötigte Code-Skeleton generiert sowie relevante Dateien angelegt. Muse erzeugt dabei eine fest vorgegebene Dateistruktur.

Standardmäßig sind in den Muse Resources folgende Capabilities als Klassen implementiert:

- WSRP ResourceProperties
- WSMEX MetadataExchange
- WSN NotificationProducer
- WSDM MUWS Identity
- WSDM MUWS Description
- WSDM MUWS OperationalStatus
- MyCapability (um weitere spezifische Capabilities zu implementieren)

5.1.2 Bewertung

Apache Muse bietet zwar eine Java-Implementierung von WSDM, WSRF und WSN, ohne gründliche Kenntnis der Einzelspezifikationen ist es allerdings nahezu unmöglich, sinnvolle Web Services zu definieren. Das Muse beiliegende WSDL-Template, das bereits einen Großteil der WSDM-, WSRF- und WSN-Spezifikationen definiert, bietet jedoch eine gute Grundlage, um auch komplexere Web Services zu erstellen. Der größte Aufwand bei der Entwicklung der Services besteht in der konkreten Implementierung der Java Klassen der eigenen Capabilities. Die eigentliche Logik der Web Services ist dort anzutreffen und muss vollständig eigenhändig programmiert werden. Muse bietet also nur die Implementierung der Spezifikationen, also die grundlegenden Funktionen wie zum Beispiel: Abfragen von Properties, Verteilen von Nachrichten, Verwalten von Subscriptions und Empfangen von Nachrichten.

Muse liegt seit mehr als eineinhalb Jahren in Version 2.2 vor. Muse 3.0 ist bereits seit über einem Jahr angekündigt und soll viele Neuerungen und Vereinfachungen mit sich bringen.

Muse setzt mit Axis2 in einem Tomcat momentan zwingend Java 1.5 voraus und ist im aktuellen Stand nicht mit Java 1.6 lauffähig.

Da ein Web Service im Prinzip stetig weiterentwickelt wird, sei es durch Erweiterung des Funktionsumfangs oder durch Abänderung der logischen Implementierung, führt dies zu großen Problemen wenn der Web Service auf Muse basiert. Mittels `wsdl2java` werden zwar die Java-Klassen der benutzerspezifischen Capabilities generiert, doch eine Erweiterung des Funktionsumfangs erfordert eine Änderung des WSDL-Dokuments und eine Neu-Generierung der Capabilities mittels `wsdl2java`. Unglücklicherweise können jedoch die Code-enthaltenden Java-Klassen nicht überschrieben werden. Auf die neu-definierten Funktionen kann deshalb nicht zugegriffen werden. Wird jedoch das Überschreiben mittels `-overwrite` oder durch explizite Anpassung innerhalb der `.overwrite` Datei erzwungen, so besitzen die Capabilities zwar alle neuen und alten Funktionsgerüste, aber die alte Logik ist

nicht mehr vorhanden. Der Entwickler wird also gezwungen, die Logik der alten Funktionen zuerst zu sichern, danach mittels –overwrite eine Neugenerierung zu erzwingen, um anschließend die alte Logik wieder in die neuen Klassen zu kopieren. Erst danach kann er den neuen Funktionen die neue Logik hinzufügen.

Ein großes Problem sind Fehlermeldungen von Muse. Diese sind alles andere als gut. Es wird – wenn überhaupt – die Java-Fehlermeldung geworfen, manchmal nicht einmal diese, sondern nur ein einfaches „null“. Man weiss in diesem Fall weder woher, noch warum die Meldung geworfen wurde. Fehlersuche in Muse-Umgebungen ist schwer und mühselig!

Das Fazit der Arbeit: Muse steckt noch in sehr kleinen Kinderschuhen. Muse bietet zwar viele komfortable Funktionen und die Implementierung der Spezifikationen ist natürlich begrüßenswert, für die Unterstützung der Entwicklung von Diensten muss jedoch noch einiges geleistet werden. Muse ist daher im aktuellen Stand nicht wirklich für den produktiven Einsatz geeignet. Vor allem in Hinblick darauf, dass die Spezifikationen sowieso so gut wie nicht im Einsatz sind. Auch unter dem Gesichtspunkt, dass WSDM und WSM (DMTF's Web Services for Management-Spezifikation) über kurz oder lang zu einer gemeinsamen Spezifikation verschmelzen werden, die dann von Muse 3.0 implementiert werden soll.

5.2 Vereinheitlichung von VO-Managementschnittstellen über heterogene Grids

5.2.1 Ziele

Ein wichtiger Aspekt in einer Grid-Umgebung ist die Authentifizierung und Autorisierung der Benutzer, die heute in sehr unterschiedlicher Art und Weise in den Grid-Middleware-Konzepten implementiert wird. Damit die Verwendung eines Grids und dessen Ressourcen vereinfacht wird, müssen die Authentifizierungs- und Autorisierungsmechanismen der verschiedenen Middleware-Technologien vereinheitlicht werden.

Im Rahmen einer Diplomarbeit [kirchler] an der Technischen Universität München wurde ein neues Konzept verfolgt. Es wurde eine zusätzliche Abstraktionsschicht eingeführt, welche die Authentifizierung und Autorisierung von der Grid-Middleware entkoppelt und in eine darüber liegende Schicht, den so genannten VO-Layer, integriert. Der VO-Layer hat die Aufgabe, die Authentifizierungs- und Autorisierungsanfragen der Benutzer entgegenzunehmen, zu prüfen und anschließend an die entsprechende Middleware-Technologie weiterzuleiten. Der VO-Layer fungiert somit als Proxy zwischen dem Benutzer und der Grid-Middleware, welche die Ressource bereitstellt.

Damit die Autorisierung überhaupt vereinheitlicht werden kann, wurde eine generische VO-Struktur entwickelt (siehe auch [genrischevo]), auf die die verschiedenen VO-Konzepte der einzelnen D-Grid Communities abgebildet werden können. Dadurch kann jede Community ihr eigenes VO-Konzept beibehalten, die Autorisierung aber wird dadurch vereinheitlicht.

5.2.2 Fazit

Es konnte gezeigt werden, dass die wesentlichen Authentifizierungs- und Autorisierungsanforderungen im D-Grid mit dem VO-Layer-Konzept erfüllbar sind. Der VO-Layer fungiert zur Authentifizierung als Proxy zwischen dem Benutzer und der eigentlichen Grid-Middleware. Der Benutzer gibt wie gewohnt sein Zertifikat an, anschließend wird der VO-Layer aktiv und übermittelt die Zertifikatinformationen mittels eines Grid-Befehls an die entsprechende Grid-Middleware. Dort erfolgt der eigentliche Authentifizierungsvorgang und bei positiver Authentifizierung erhält der Benutzer ein Proxy-Zertifikat, welches ihn ermächtigt, die vom Resource Provider bereitgestellten Ressourcen zu benutzen. Die Authentifizierung ist somit mittels des VO-Layers einheitlich möglich, die Heterogenität der Middleware wird verschattet.

Auch zur Autorisierung fungiert der VO-Layer als Proxy zwischen Benutzer und der eigentlichen Grid-Middleware. Der Benutzer möchte auf einer Ressource des Grids einen Job ausführen, benötigt dazu aber die Zugriffsrechte auf die Ressource. Zunächst überprüft der VO-Layer, ob der Benutzer überhaupt berechtigt ist, auf die gewünschte Ressource zuzugreifen. Dazu wird geprüft, ob sich beide in der selben VO befinden (eine Folge der zugrunde liegenden VO-Definition). Falls dies zutrifft, wird überprüft, ob der Benutzer die nötigen Zugriffsrechte anhand der spezifizierten Gruppen, Rollen und Fähigkeiten besitzt. Dazu prüft der VO-Layer, ob der Benutzer mindestens einen gleichen Fully Qualified Attribute Name (FQAN) wie die angeforderte Ressource besitzt. Ist dies der Fall, darf der Benutzer den Job auf der Ressource ausführen, ansonsten wird der Zugriff auf die Ressource bereits durch den VO-Layer unterbunden. Wird der Zugriff genehmigt, startet der VO-Layer den eigentlichen Job auf der entsprechenden Grid-Middleware. Dazu wird – durch einen Grid-Befehl - der Job an das Zielsystem übergeben. Die Grid-Middleware autorisiert den Benutzer erneut - wobei diese Autorisierung stets erfolgreich verlaufen muss. Daraufhin wird der Job zur Ausführung gebracht und der Benutzer kann im Anschluss das Ergebnis des Jobs abrufen.

Das Ziel, alle im D-Grid verwendeten VO-Konzepte mit Hilfe einer generischen VO-Struktur zu unterstützen, wird durch die eingeführte VO-Struktur erreicht. Es ist möglich, VOs mit beliebig vielen Benutzern, Ressourcen und sub-VOs zu etablieren. Benutzer und Ressourcen können ihrerseits in beliebig vielen VOs vertreten sein. Da viele D-Grid Communities das Konzept der Gruppen, Rollen und Fähigkeiten in verschiedenen Ausführungen umsetzen (nur Gruppen, nur Rollen oder Kombinationen aus den dreien), werden zusätzlich zur Entität VO auch noch die Entitäten Gruppe, Rolle und Fähigkeit eingeführt. Aus diesen drei Entitäten können beliebige Tripel (Gruppe, Rolle, Fähigkeit) generiert werden. Diese so genannten FQANs geben für einen Benutzer an, welche Zugriffsberechtigungen er in einer VO hat. Für eine Ressource legen sie fest, welche Zugriffsbedingungen für den Zugang nötig sind. Da nicht alle D-Grid Communities alle drei Entitäten unterstützen, kann eines oder mehrere der drei den Wert "NULL" annehmen. Dadurch wird eine generische VO-Struktur geschaffen, welche von den D-Grid Communities – unabhängig von deren VO-Konzept - einheitlich verwendet werden kann.

Neben der Authentifizierung und Autorisierung der Benutzer ist ein einheitliches und einfaches Management der VO-Struktur durch den VO-Layer möglich. Es hat sich gezeigt, dass die schon heute für den Betrieb verwendeten Datenbanken (VOMS, GRRS) weiter verwendet werden können. Die Weiterleitung von Änderungen in der VO-Struktur an die darunter liegenden Grid Middleware-Technologien erfolgt periodisch, wobei aus der VO-Struktur für jeden Hostrechner - auf dem sich eine Ressource befindet - eine neue ACL in Form eines gridmap-file (für Globus und gLite Ressourcen) oder einer UADB (für UNICORE Ressourcen) generiert wird. Diese Dateien werden anschließend auf die jeweiligen Hostrechner übertragen.

5.3 Handhabung von Attributen in heterogener Grid Middleware

Die D-Grid Infrastruktur orientiert sich wesentlich an den Anforderungen der D-Grid Communities. Diese Communities, zum Teil in internationale Kooperationen eingebunden, haben aber unterschiedliche Anforderungen, die nicht durch eine homogene Infrastruktur erfüllt werden können. Dadurch ergab sich in D-Grid die parallele Nutzung von drei unterschiedlichen Middlewaresystemen: Globus Toolkit 4, gLite und UNICORE 5.

Eine erste Bestandsaufnahme im Rahmen des D-Grid Projekts zur Entwicklung eines Rahmenkonzepts für das Management Virtueller Organisationen im D-Grid [rahmenkonzept] ergab u.a. die Notwendigkeit einer Harmonisierung der drei Middlewaresysteme hinsichtlich der in zur Autorisierung verwendeten Technologie.

Der Grad an Unterstützung für attributbasierte Autorisierung, die im Grid zum de facto Standard geworden ist, und die dafür eingesetzte Technologien wichen allerdings so stark voneinander ab, dass eine Harmonisierung im Rahmen des VO-Management Projekts nicht möglich war.

Zur Realisierung wurde daher ein weiteres Projekt definiert, das sich speziell mit der Interoperabilität der unterschiedlichen Ansätze beschäftigt: IVOM – Interoperabilität und Integration der VO-Management Technologien im D-Grid [ivom].

UNICORE 5 verfügte zu Beginn von IVOM über eine rein identitätsbasierte Autorisierung (realisiert durch X.509 Benutzerzertifikate), während Globus Toolkit 4 und gLite bereits über einen ähnlichen Mechanismus verfügen, der es erlaubt, Attribute eines Nutzers, die im Rahmen seiner Zugehörigkeit zu einer Virtuellen Organisation festgelegt werden, in Proxy-Zertifikaten verpackt zu transportieren.

Parallel zu den existierenden Ansätzen, die Autorisierung über (Proxy-)Zertifikate zu realisieren, die eine PKI voraussetzen, entwickelte sich eine Technologie, die ohne PKI auskam, stattdessen zur Authentifizierung eines Nutzers die Mechanismen seiner Heimateinrichtung zu verwenden und auch die Attribute für die Autorisierung beim Service Provider von seiner Heimateinrichtung zu beziehen. Der am weitesten verbreitete Ansatz dazu ist Shibboleth [shibboleth] und einige Communities waren und sind stark an einer Verwendung dieser alternativen Technologie zur Autorisierung interessiert.

Wesentliche Aufgaben von IVOM waren es daher, zum einen Voraussetzungen und Technologie für eine Nutzung der Shibboleth in D-Grid zu evaluieren und zum anderen für UNICORE 5 die notwendigen Erweiterungen zu definieren und zu implementieren, um für attributbasierte Autorisierung mit Globus Toolkit 4 und gLite kompatible Technologie zu bereit zu stellen.

Gleichzeitig wurden Erweiterungen für UNICORE 5 realisiert, die es erlauben, Attribute aus einer Shibboleth Umgebung in UNICORE 5 zur Autorisierung zu verwenden.

5.3.1 Realisierung der Shibboleth Integration durch GridShib CA und UNICORE 5 plug-in

Da der DFN für den SLCS Testbetrieb die GridShib-CA nutzt und diese für die Erstellung von dynamisch erzeugten, kurzlebigen Zertifikaten (Short Lived Credentials, SLCs) verwendet wird, wird für die UNICORE 5 Integration mit Shibboleth eine Lösung realisiert, die ebenfalls auf der GridShib-CA basiert.

Die GridShib-CA ist für das Abbilden einer Shibboleth-Identität auf eine Grid-Identität in Form eines SLCs zuständig. Das SLC enthält dafür die vom Shibboleth IdP bereitgestellten Benutzerattribute als SAML-Assertions.

Der UNICORE-Client soll die SLCs aufnehmen können, diese in ein Benutzerzertifikat transformieren und in den Keystore des Benutzers einfügen. Zusätzlich muss die UUDB die SAML Assertions des Benutzers extrahieren können und diese in ein SAML-Objekt transformieren, um Funktionalitäten wie Validierung, Verifizierung und das Entnehmen der Benutzerattribute zu ermöglichen. Abbildung 2 stellt die Architektur und den Ablauf der Kommunikation zwischen den Komponenten dar.

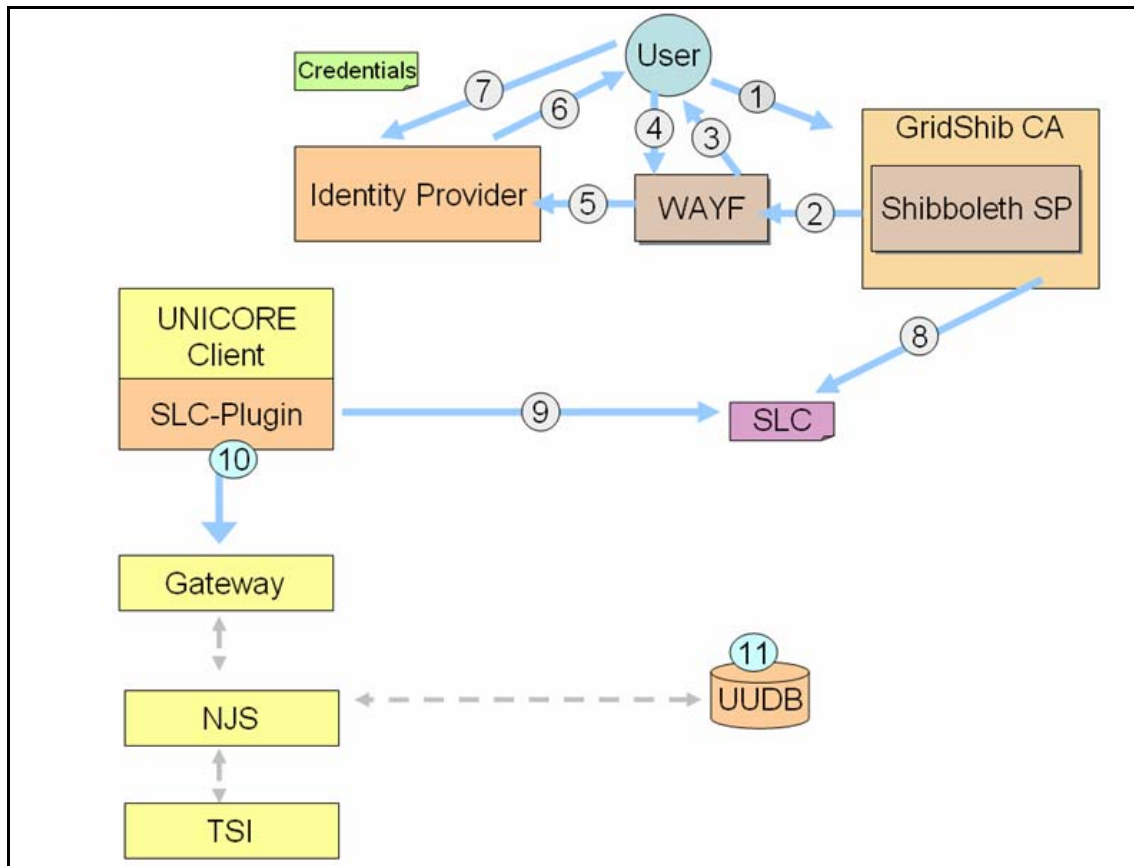


Abbildung 2: Integrationslösung mit GridShib CA und UNICORE 5-Plugin

Folgende Schritte werden durchlaufen, um Attribute eines Benutzers von seiner Heimateinrichtung für die Autorisierung in UNICORE zu verwenden:

1. Der Benutzer greift auf den Online-Dienst GridShib CA zu.
2. Der Dienst GridShib CA fungiert als Service Provider und leitet den Benutzer an einen WAYF-Service weiter.
3. Der WAYF-Service gibt dem Benutzer die Möglichkeit, seine Heimateinrichtung auszuwählen.
4. Der Benutzer wählt seinen IdP aus.
5. Der WAYF-Service leitet den Benutzer per Redirect zu seinem IdP weiter.
6. Der Benutzer gibt dem IdP seine Credentials weiter.
7. Der IdP authentifiziert den Benutzer anhand seiner Credentials und schickt die erhaltenen SAML-Assertions dem GridShib CA weiter.
8. Nach einer erfolgreichen Authentifizierung erstellt die GridShib-CA ein SLC. Die erhaltenen SAML-Assertions werden in die Extension des SLCs eingebettet.

9. Das zu implementierende SLC-Plugin erhält das SLC des Benutzers von der GridShib CA und verwendet dieses als Benutzerzertifikat für UNICORE.
10. Das SLC, welches nun als Benutzerzertifikat dient, wird an das Gateway und den NJS weitergeleitet. Dies setzt aber voraus, dass das Gateway und der NJS den erstellten SLCs vom GridShib CA vertrauen. Der NJS sendet dann das SLC an die UUDB weiter.
11. Die UUDB extrahiert die SAML-Assertions aus der Zertifikatserweiterung. Sie entnimmt aus den SAML-Assertions die Benutzerattribute und trifft anhand dieser eine Autorisierungsentscheidung

Durch die Verwendung der GridShib-CA kann eine Interoperabilität zur Middleware Globus ermöglicht werden, da Globus ebenfalls die GridShib-CA nutzen kann. Die Middleware gLite verwendet eine modifizierte GridShib-CA für die eine Anpassung notwendig wäre. Für das Realisieren der Interoperabilität der verschiedenen Grid-Middleware gegenüber VO-Benutzern durch Shibboleth muss lediglich eine einheitliche Darstellung der SLCs erarbeitet werden. Das UNICORE-SLC-Plugin kann leicht erweitert werden und mit anderen VO-Management-Systemen, die SLCs generieren, benutzt werden.

5.3.2 Realisierung der VOMS Integration durch eine Keystore-Erweiterung

Hierbei verwendet der Benutzer sein Benutzerzertifikat, um Zugang zum UNICORE und dem VOMS-System zu erhalten. Sein Benutzerzertifikat und sein privater Schlüssel werden in UNICORE wie üblich in Form eines P12-Keystores abgespeichert. Der UNICORE-Client wird um ein Proxy-Plugin erweitert, welches das VOMS-Proxy-Zertifikat in die Certificate-Chain des Benutzer-Keystores einfügt.

Abbildung 3 und die nachfolgende Auflistung veranschaulichen Architektur und Kommunikationsablauf:

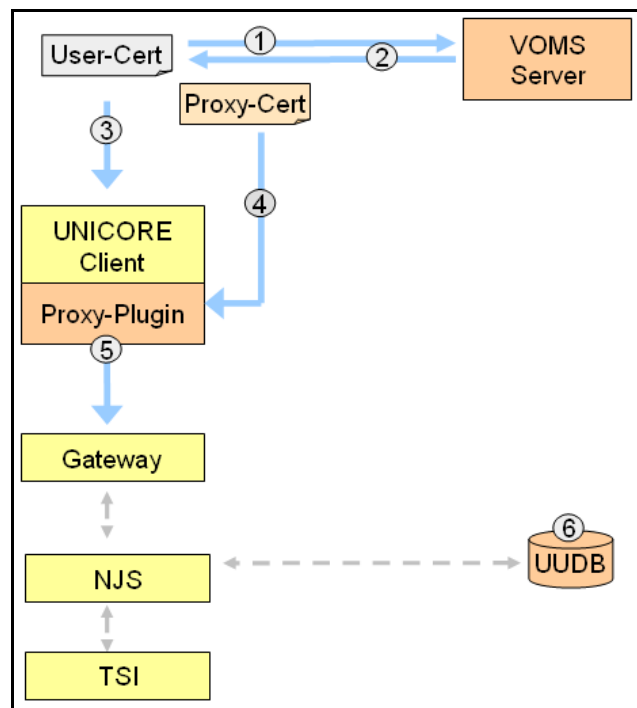


Abbildung 3: Integrationslösung durch eine Keystore-Erweiterung

1. Der Benutzer führt *voms-proxy-init* aus und spezifiziert seinen VOMS-Request. Dadurch wird sein VOMS-Request an den VOMS-Server gesendet.
2. Der VOMS-Server validiert den VOMS-Request und sendet die benötigten Autorisierungsinformationen in Form eines Attribut-Zertifikats an den Benutzer zurück. Der *voms-proxy-init*-Mechanismus erstellt anschließend ein Proxy-Zertifikat und fügt das Attribut-Zertifikat in ihm hinzu. Der Benutzer signiert das Proxy-Zertifikat mit seinem privaten Schlüssel.
3. Der Benutzer startet den UNICORE-Client und authentisiert sich standardmäßig mit seinem Benutzerzertifikat.
4. Der Benutzer führt das Proxy-Plugin aus und gibt sein ProxyZertifikat an das Plugin weiter.
5. Das Proxy-Plugin validiert und verifiziert das Proxy-Zertifikat. Anschließend erweitert das Plugin das Certificate-Chain des Benutzer-Keystores um das Proxy-Zertifikat.
6. Die erweiterte UUDB extrahiert die VO-FQANs aus dem ProxyZertifikat und bildet die VO-FQANs auf lokale Accounts ab

Die NJS-Kern-Implementierung und der UNICORE 5-Authentisierungs-Mechanismus werden mit diesem Realisierungsvorschlag nicht verändert. Interoperabilität mit anderen VO-Management-Systemen ist gegeben. Es muss kein zusätzliches Benutzer-Credential erstellt werden. Diese Integrationslösung kann von den D-Grid-Communities sofort genutzt werden.

Diese Architektur wurde für die Implementierung der UNICORE und VOMS-Integration gewählt, da sie eine Interoperabilität zu anderen VO-Technologien gewährleistet. Trotz des unvollständigen SSO, erfüllt dieser Vorschlag die Anforderungen an die Integration. Außerdem kann dieser Ansatz in Kombination mit der Shibboleth-Integration implementiert und verwendet werden.

5.3.3 Weitere Ansätze zur Handhabung von Benutzerattributen

Wie oben beschrieben, basiert Shibboleth auf SAML-Assertions, mit denen die Benutzerattribute zum Serviceprovider transportiert werden- Im EU-Projekt OMII-Europe wurde eine Integration von SAML-Assertions in UNICORE 6 realisiert, die die Verwendung von Benutzerattributen von einem VOMS-Server für die Autorisierung auf UNICORE 6 Ressourcen ermöglicht. Dafür wurde zusätzlich eine Erweiterung des bisherigen VOMS-Servers bereitgestellt, der zusätzlich zu Proxy-Zertifikaten auch SAML-Assertions für die Weitergabe von Attributen verwenden kann.

Dieser Ansatz lässt sich leicht mit dem oben dargestellten Ansatz zur Integration von Shibboleth Attributen kombinieren, sodass auch für UNICORE 6 sowohl VOMS-Attribute als auch Shibboleth-Attribute unterstützt werden können.

5.4 Zusammenfassung

Die Feasibility-Studien ergeben eine Einschätzung für die weitere Vorgehensweise:

WSDM ist in der Apache Muse Implementierung zur Realisierung der Tool Suite entgegen der ursprünglichen Annahme nicht geeignet. Dennoch geht an den in den WSDM- (und ähnlichen) Spezifikationen formulierten Grundkonzepten langfristig kein Weg vorbei. Die Realisierung der Tool Suite muss deshalb auf eine andere Art durchgeführt werden.

Eine unterstützende Entwicklungsumgebung ist zur Zeit nicht vorhanden. Inwieweit die zur Zeit stattfindenden g-eclipse-Arbeiten³ hier hilfreich sein können, muss in einer gesonderten Arbeit geklärt werden.

Als Korollar der im D-Grid angestrebten Vereinheitlichung der VO-Strukturen (siehe auch [generischevo], [attribute], [workflow]) bei gleichzeitiger Beibehaltung der Heterogenität kann für das VO-Management die Einführung einer zusätzlichen Abstraktionsebene Erfolg versprechend sein, insbesondere dann, wenn sich VOs überlappen können.

Die Handhabung von Attributen in heterogener D-Grid-Middleware ist im Prinzip für alle drei in Frage stehenden Middleware-Ansätze (gLite, UNICORE, Globus) verstanden und für die gleichzeitige Nutzung von VO- und Campus-Attributen auch umgesetzt. Zur Zeit ist dazu für letztere Attribute eine GridShib-CA erforderlich.

6 Realisierung der Tool Suite

Die Realisierung der Tool Suite erfolgt über eine Hierarchie von Skripten. Die genaue Umsetzung wird in eine spätere Version dieses Dokumentes integriert.

Die Implementierungen der Feasibility-Studien des Abschnitts 5 sind veröffentlicht und frei verfügbar. Die finale Version dieses Dokuments (Meilenstein 12) wird diese Implementierungen ebenso in einem separaten Appendix enthalten wie den Code der Suite.

7 Deploymentaspekte

Gegenstand der Berichtsversion zu Meilenstein 12

8 Fazit und Empfehlungen

Gegenstand der Berichtsversion zu Meilenstein 12

³ www.geclipse.eu

Abkürzungen

CA	Certificate Authority
FQAN	Fully Qualified Attribute Name
IdP	Identity Provider
IVOM	Interoperabilität und Integration der VO-Management Technologien im D-Grid
PKI	Public Key Infrastructure
SLC	Short Lived Credentials
SLCS	Short Lived Credential Service
VO	Virtual Organization
VOMRS	Virtual Organization Management Registration Service
VOMS	Virtual Organization Management Service
WAYF	Where Are You From
WSDM	Web Services Distributed Management
WSRF	Web Services Resource Framework

Referenzen

- [attribute] Definition von Attributen für die Autorisierung auf D-Grid Ressourcen, Report Fachgebiet 3.2 AAI/VO, Juli 2008
- [betriebskonzept] O. Büchner et al.: *Betriebskonzept für die D-Grid-Infrastruktur*, Version 1.1c, November 2007. http://www.d-grid.de/fileadmin/user_upload/documents/Kern-D-Grid/Betriebskonzept/D-Grid-Betriebskonzept.pdf
- [generischevo] Generische VO-Strukturen für das D-Grid, Report Fachgebiet 3.2 AAI/VO, Juli 2008
- [hoeber] Höber, M., *Realisierung und Management zustandsbehafteter Web Service Resources mit Apache Muse*, Fortgeschrittenenpraktikum Ludwig-Maximilians-Universität München, Oktober 2008, <http://www.nm.ifi.lmu.de/pub/Fopras/hoeb08>
- [ivom] Gietz, P., Grimm, C., Gröper, R., Haase, M., Makedanz, S., Pfeiffenberger, H., Schiffers, M., [A Concept for Authorization on D-Grid Resources: Report of Working Package 3 of the D-Grid IVOM-Project](#), D-Grid-Report, D-Grid IVOM, September, 2007
- [kerndgrid] <http://www.d-grid.de/index.php?id=23>
- [kirchler] Kirchler, W., *Entwicklung einer einheitlichen Autorisierungs- und Authentifizierungsschnittstelle für heterogene Grids am Beispiel D-Grid*, Diplomarbeit Technische Universität München, September 2008, <http://www.nm.ifi.lmu.de/pub/Diplomarbeiten/kirc08>
- [muse] Apache Muse. <http://ws.apache.org/muse>
- [rahmenkonzept] J.-M. Milke, M. Schiffers, W. Ziegler: *Rahmenkonzept für das Management Virtueller Organisationen im D-Grid*, November 2006. http://dgi.d-grid.de/index.php?id=118&no_cache=1&filename=VO_Rahmenkonzept_0.9a.pdf&dir=FG1/VO-Management&task=download&mountpoint=2
- [referenzinstallation] Release 1: <http://dgiref.d-grid.de/downloads/docs/Manuals-r1.pdf>
- [shibboleth] <http://shibboleth.internet2.edu/>
- [workflow] Ch. Grimm, M. Schiffers, T. Fieseler, S. Piger, Ch. Dohmen: *Schematische Darstellung von VO-Management-Workflows im D-Grid*. Juni 2008. (wird noch veröffentlicht)
- [wsdm] OASIS: Web Services Distributed Management (WSDM) TC, <http://www.oasis->

open.org/committees/tc_home.php?wg_abbrev=wsdm

[wsn] Steve Graham, Bryan Murray: *Web Services Base 2 Notification 1.2.*, <http://docs.oasis-open.org/wsn/2004/06/wsn-WS-BaseNotification-1.2-draft-03.pdf>

[wsrf] OASIS: Web Services Resource Framework, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsrf