



Softwaretechnische Analyse und Design (AP 1.3)

Version 0.23

Autoren:

Timo Baur (LRZ)

Anne Milbert (AEI)

Mathilde Romberg (FZJ)

Gevorg Poghosyan (FZK)

Tobias Lindinger (LMU)

28.10.2008

1	<i>Einleitung</i>	3
2	<i>Überblick</i>	3
3	<i>Analyse</i>	4
3.1	Automatische Datenmigration	4
3.2	Bidirektionale Gateways	5
3.3	Korrelation der Ressourcendaten mit VO-Information	5
3.4	VO-sensitive Zugriffsverfahren	6
3.5	Standard-konforme Zugriffsschnittstellen	7
3.6	Föderation mehrerer D-MON Dienste	8
3.7	Nutzer in verschiedenen Rollen	8
4	<i>Modulares Design des D-MON Informationsdiensts</i>	9
4.1	Automatische Datenmigration	9
4.2	Die Monitoring-Datenbank	10
4.3	Föderation von Datenbanken mehrerer Ressourcenanbieter	11
4.4	Benutzerschnittstellen der Monitoring-Datenbank	11
5	<i>Gesamtsystem</i>	13
5.1	Benutzer-Interfaces	14
5.2	Servermodule	14
6	<i>ANHANG A – Mapping der Informationsmodelle</i>	17
6.1	Globus	17
6.2	UNICORE	18
6.3	gLite	20

1 Einleitung

Basierend auf den Ergebnissen aus AP1.1 (Sammlung von Anforderungen, [AP1.1]), AP1.2 (Evaluierung und Auswahl verfügbarer Komponenten, [AP1.2]) und AP2.1 (Aufbau des Testbetts, [AP2.1]) behandelt dieses Dokument die notwendigen Weiterentwicklungen. Es werden hier die zu entwickelnden Komponenten durch eine softwaretechnische Analyse und ein anschließendes Design spezifiziert. Nach einem kurzen Überblick über das Gesamtsystem im Kapitel 2 werden im Kapitel 3 die Möglichkeiten analysiert. Kapitel 4 enthält für alle Komponenten die Spezifikationen der Methoden und Schnittstellen. Dabei wird insbesondere darauf geachtet, eine Zusammenarbeit der Komponenten untereinander sowie mit den Komponenten des Virtual Organisation (VO) Managements zu erreichen. Zusammenfassend wird in Kapitel 5 das Gesamtsystem dargestellt.

2 Überblick

Abbildung 1 zeigt den initialen Aufbau des Testbetts (siehe [AP2.2]). Das Testbett enthält je eine der im vorhergehenden Arbeitspaket ausgewählten Komponenten. Im Zusammenspiel sollen diese später eine horizontale (middleware-integrative) und vertikale (VO-integrative) Lösung ergeben. Die Grafik zeigt relevante Datenquellen in Form von Monitoringsystemen dreier Middlewares, eine Datenbank mit den zum VO-Management relevanten Daten, eine zentrale SQL-Datenbank zum Zusammenführen und Korrelieren aller Daten, sowie ein Instanz des Portalframeworks Gridsphere, welche zum Anzeigen der Daten dient. An den durch einen Punkt markierten funktionalen Übergängen müssen Erweiterungen vorgenommen und Schnittstellen definiert werden.

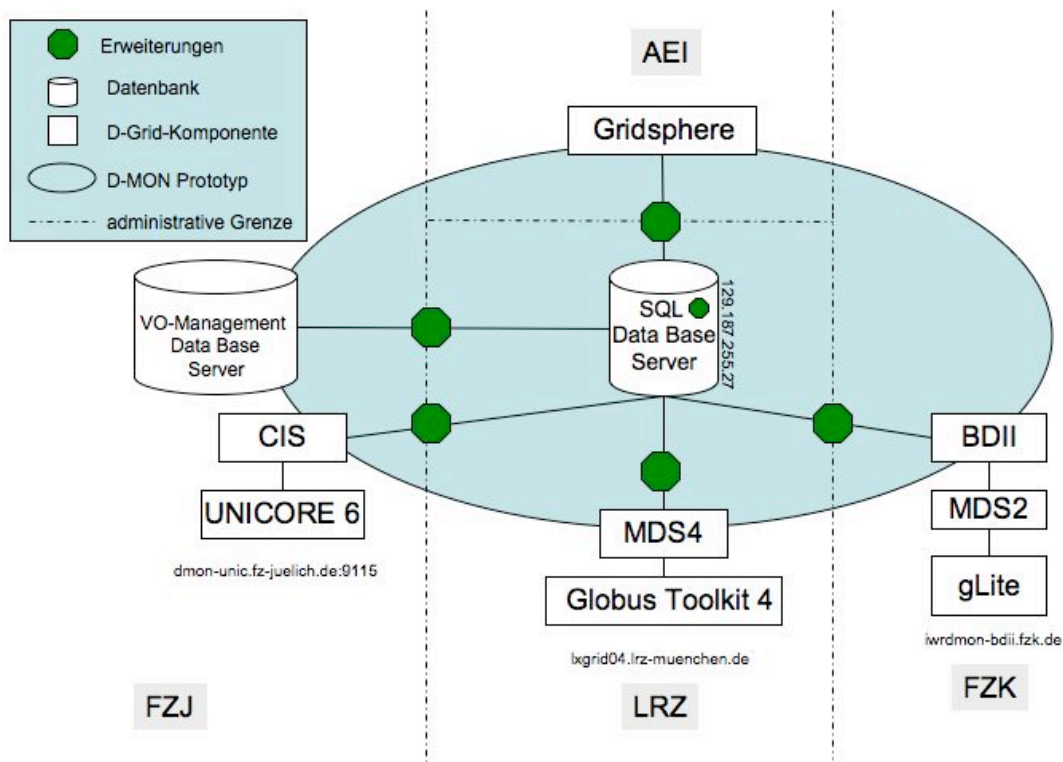


Abbildung 1: Aufbau des Testbett

3 Analyse

Die Sammlung von Anforderungen an ein D-Grid Monitoring-System [AP1.1] und die Auswahl der verfügbaren Komponenten [AP1.2] haben zur Identifikation der neu zu entwickelnden notwendigen Basiskomponenten geführt, wie sie in Abbildung 1 dargestellt sind. Für die Anforderung „horizontale Integration“, die den Datenaustausch zwischen den bestehenden Informationssystemen einfordert, sind Adapterkomponenten zu schaffen, wie sie in Abbildung 1 als Erweiterungen gekennzeichnet sind. Diese Adaptern müssen neben der Übersetzungsfunktion auch eine Datentransferfunktion bieten. Im Folgenden wird die Gesamtfunktionalität wie sie aus dieser Anforderung resultiert als automatische Datenmigration bezeichnet (Kapitel 3.1). Nach den Anforderungen an die horizontale Integration sollen alle Adaptern bidirektional funktionieren, um die Informationen auch über jede der beteiligten Middlewares abrufen zu können. Dieser Aspekt wird in Kapitel 3.2 behandelt. Die Anforderung „vertikale Integration“ umfasst die Integration von Monitoring- und Virtual Organisation (VO) Management-Daten. In den Kapiteln 3.3 und 3.4 wird dieses softwaretechnisch analysiert. Weitere Details zum theoretischen Hintergrund und verwandte Arbeiten finden sich beispielsweise in [Ba08].

Bereits bestehende Zugriffsverfahren aus der Authentication and Authorisation Infrastruktur des D-Grid Integrationsprojekt [DGI-AAI] sowie dem VO-Management (Rahmenkonzept [MiSchZi07], Interoperabilität und Integration der VO-Management Technologien [IVOM], AAI/VO [DGI2-FG3]) werden hierbei berücksichtigt. Standardisierung ist für die spätere Verbreitung des Monitoring-Dienstes wichtig, weswegen Konformität zum Open Grid Services Architecture (OGSA) Web Services Resource Framework (WS-RF) für die Schnittstellen von vornherein festgelegt wurde (Kapitel 3.5). Vom Gesamtsystem wird erwartet, dass es skalierbar ist und keinen single-point-of-failure bietet. Dieses soll durch ein föderiertes System, das an jedem Standort Daten verteilt vorhält, erreicht werden (Kapitel 3.6). Einen entscheidenden Platz nehmen die VO-spezifischen Sichten auf die Monitoring Daten ein, die über einen Command Line Client sowie über Gridsphere Portlets zur Verfügung gestellt werden sollen. Zusätzlich sind die Rollen der Anfragenden wichtig, um Views für konkrete Anwendungs-Szenarien erstellen zu können. Diese werden in Kapitel 3.7 beleuchtet.

3.1 Automatische Datenmigration

Für die automatische Migration der Monitoringdaten zwischen den verschiedenen Informationssystemen an verschiedenen, administrativ autarken Standorten und auf der Basis verschiedener technischer Plattformen muss ein Migrationsprozess stattfinden, der sowohl den Zugriff auf die Datenquellen und deren Schnittstellen, sowie die Konvertierung der Daten in das in [AP1.2] ausgewählte gemeinsame Datenmodell GLUE 2.0 übernimmt. Erfolgt dieser Prozess automatisch und regelmäßig, kann so eine andauernde horizontale Datenintegration sichergestellt werden. Wenn notwendig, kann der Prozess auch on-demand angestoßen werden, beispielsweise um ein aktives Pull-Verfahren zu erlauben.

Der Migrationsprozess ist im Wesentlichen der aus dem Data-Warehousing bekannte [ChDa97] Extract, Transform, Load Prozess (siehe Abbildung 2).

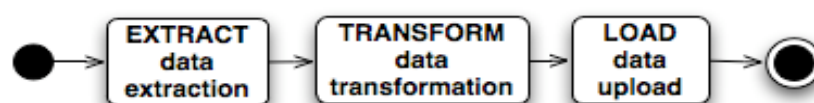


Abbildung 2: Datenmigrationsprozess

Der Teilprozess EXTRACT führt dabei die Extraktion der Daten aus den verschiedenen eingesetzten Monitoringsystemen bzw. Informationsdiensten als Datenquellen durch.

Der Teilprozess TRANSFORM ist zuständig für eine Datentransformation. Diese realisiert die Abbildung der Informationsmodelle von den Quellsystemen auf das gemeinsame Zielmodell GLUE 2.0.

Der Teilprozess LOAD erledigt dann den Upload der Daten in die gemeinsame Datenbank und die dafür notwendigen Verfahren.

3.2 Bidirektionale Gateways

Der Anforderung, dass Daten aus jedem Informationssystem auch in jedem anderen zur Verfügung stehen soll, damit keine Middleware bevorzugt wird, lässt sich in dem entworfenen System dadurch entgegenkommen, dass alle Informationen aus den verschiedenen Informationssystemen in eine gemeinsame Datenbank zusammengeführt werden. Über middleware-unabhängige Schnittstellen können die Daten dann aus der gemeinsamen Datenbank abgerufen werden.

Dies ermöglicht auch ein Weitergeben von Monitoringdaten über Middlewaregrenzen hinweg, so dass Daten, die von einem Informationsdienst bereitgestellt werden, in einem anderen auch verfügbar gemacht werden können, um middleware-spezifischen Diensten deren Nutzung zu ermöglichen. Dazu bietet sich an, relevante Daten aus der gemeinsamen Datenbank zu extrahieren und in middleware-spezifische Informationsdienste zu integrieren.

3.3 Korrelation der Ressourcendaten mit VO-Information

Der nach außen angebotene Teil der Ressourcen und Dienste innerhalb einer Grid-Infrastruktur ist zur Nutzung durch bestimmte VOs reserviert.

In einigen Fällen, wie beispielsweise bei den Einträgen im BDII, sind die Ressourceninformationen deshalb mit zusätzlichen Parametern versehen, welche die VO-Zugehörigkeit der Ressourceninformation beschreiben („tagging“). In anderen Fällen sind die Daten nicht mit einer solchen Information versehen oder diese Information ist durch die Verknüpfung mehrerer Informationsdienste miteinander implizit gegeben. Wo nicht anders möglich, müssen die Informationen nachträglich einzelnen VOs zugeordnet werden.

Um diese Zuordnung zuverlässig vornehmen zu können muss eine Instanz vorhanden sein, welche eine entsprechende Zuordnungsentscheidung trifft, ein sogenannter ‚Policy Decision Point‘ (zu einer detaillierteren Diskussion siehe [Ba08]). Solche Instanzen sind im D-Grid zumindest auf technischer Ebene bereits vorhanden (auf einer organisatorischen Ebene wären sie dringlich, beispielsweise als Gremium oder Koordinationsgruppe, klar zu definieren bzw. zu instanzieren. Es handelt sich dabei um die im VO-Management Rahmenkonzept [MiSchZi07] theoretisch geforderten und im Betriebskonzept für die D-Grid Infrastruktur [Gü07] festgelegten Komponenten, welche die Zuordnung von Ressourcen und Nutzern zu VOs sowie deren Rollen und Rechte innerhalb der VOs bereitstellen. In den im D-Grid eingesetzten Monitoringsystemen wird die Entscheidung über die Zuordnung, welche Ressource durch welche VO genutzt (und damit auch überwacht) werden darf, bisher üblicherweise verteilt von den jeweiligen Ressourcenanbietern selbst auf der Basis ihrer Verpflichtungen getroffen. Formalisiert wird diese Information dann mit Hilfe eines GRRS (Grid Resource Registry Service) in entsprechenden Datenbankeinträgen zentral eingetragen (siehe auch [MiSchZi07]). Die Unstrukturiertheit dieses Verfahrens gibt Raum für Fehler und eine sehr grobe Unschärfe und Statik der Einträge. So sind in der Regel keine klaren Service Level Agreements (SLAs) vereinbart. In der Praxis hat sich auch gezeigt, dass die Einträge für fast alle VO's nahezu identisch sind. Für dynamische Zuweisungen von Ressourcen zu

VOs muss hier eine feinere Granularität erreicht werden, welche sich nicht nur auf das Monitoring auswirken dürfte, sondern auch zu einem genaueren Abbild der real vorhandenen Organisationsformen führt und damit ein genaueres Ressourcenscheduling sowie eine angepasste und damit wirtschaftlichere Ressourcenkapazitätsplanung erlaubt.

Der GRRS-Dienst im D-Grid erfüllt dennoch die Funktionalität der in Abbildung 1 als VO-Management Database Server bezeichneten Einheit. Zur Nutzung der Daten aus dem GRRS durch den D-MON Dienst wurde von uns ein weiterer ETL-Prozess etabliert, welcher die Daten ausliest und in geeigneter Weise zur Laufzeit in die D-MON Datenbank einpflegt.

3.3.1 Zuordnungspolicy

Die Zuordnung von Ressourcen und Diensten zu VOs wird über die Definition von Mappings (AccessPolicies) und VOs (UserDomains) unterstützt.

Schwieriger gestaltet sich die Zuordnung von Jobs (ComputingActivity) zu einer VO, da nur der gLite Informationsdienst die VO Information mitführt. Aus der Ressource, auf der der Job ausgeführt wird, und der Benutzernummer, unter der der Job bearbeitet wird, lässt sich nicht eindeutig auf die zugehörige VO schließen, da sowohl eine Ressource mehreren VOs zur Verfügung stehen kann als auch ein Benutzer Mitglied in mehreren VOs sein kann. Wenn die Schnittmenge mehr als eine VO enthält, ist keine Aussage möglich. Hier kann nur auf die Middleware-Entwickler Einfluss genommen werden, damit die Middleware entsprechend angepasst wird und sie die VO Information z.B. immer mitführt.

3.3.1 Generierung VO-spezifischer Datensätze

Auf der Basis der Zuordnungspolicies können nun VO-spezifisch Datensätze zu den einzelnen Datensätzen erstellt werden. Diese können beispielsweise als Datenbankviews realisiert werden.

3.3.2 Konsistenzprüfung

Durch das Zusammenführen der Informationen über die Zusammensetzung der VOs aus der GRRS Datenbank und den Informationsdiensten der verschiedenen Middlewares werden ggf. bestehende Inkonsistenzen beim Monitoring der gleichen Objekte durch verschiedene Plattformen sichtbar. So lassen sich beispielsweise alle Einträge von Ressourcen identifizieren, die keinen korrespondierenden Eintrag im GRRS haben, und zum anderen werden Abweichungen von Zustandsbeschreibungen durch verschiedene Middleware-konfigurationen, z.B. Inkonsistenzen in der Beschreibung und Bezeichnung gleicher Ressourcen deutlich.

Eine Konsistenzprüfung auf die Einhaltung von Service Level Agreements (Dienstgütevereinbarungen) kann insofern erfolgen, wie die Einträge zwischen Ressourcenanbietern und VOs entsprechend diesen Vereinbarungen abgebildet werden und sich diese auf in der Datenbank abgebildete Parameter beziehen.

3.4 VO-sensitive Zugriffsverfahren

Um allen anfragenden Instanzen nur die aufgrund Ihrer Mitgliedschaft in bestimmten VOs für sie bestimmten Datensätze zur Verfügung zu stellen muss an den Zugriffspunkten GUI bzw. Datenbank eine VO-sensitive Autorisierung stattfinden, welche feststellt, welchen VOs eine anfragende Instanz (z.B. ein Nutzer oder ein anderer Grid-Dienst) angehört, und die Daten entsprechend bereitstellt. Dies kann erreicht werden, wenn die Daten mit der Zuordnung von Nutzern und Diensten zu VOs, welche aus den VO-Managementsystemen bezogen werden kann, abgeglichen werden.

Der Zugriff auf diese Zuordnung lässt sich in D-Grid auf folgende drei Arten gestalten:

- die zentrale VOMRS-Komponente (Virtual Organization Management Registration Service) in Jülich bietet relevante Benutzerinformation (Nutzer → VO) in folgenden Formaten: grid-mapfile für GT4 und gLite, UUDB für Unicore, OGSA-DAI [MiSchZi07].
- Über einen VOMS-Server (Virtual Organization Membership Service) in Karlsruhe kann auf Basis dieser Daten zusätzlich eine attribut-basierte Autorisierung realisiert werden, welche über das Ausstellen von Attribut-Zertifikaten mit VO-Zugehörigkeit sowie eventuell definierten Gruppenzugehörigkeiten und Rollen innerhalb der VO funktioniert. Dieser Server funktioniert derzeit aber nicht mit allen Middlewares [MiSchZi07]. Ein entsprechender Client könnte hier aber Abhilfe schaffen.
- Insgesamt herrscht hier noch ein diverses Spektrum verschiedener Ansätze und Zugriffsformate vor, welches im DGI-2 Projekt AAI/VO [DGI2-FG3] mit Hilfe eines einheitlichen SAML¹-VOMS Servers integriert wird. Entsprechende Ansätze für eine künftige Authentifizierung und Autorisierung mittels SAML und VOMS sind auch bei OMII-Europe und dem OGF vorhanden und ihr Einsatz würde dem state-of-the-art entsprechen. SAML bietet eine einheitliche und sichere Methode zum Übertragen und Auswerten der Attribute (SAML Assertions).

Für die Konzeption eines übergreifenden Middlewaredienstes erscheint SAML-VOMS als zukunftsweisendes Mittel zur VO-basierten Autorisierung und Authentisierung der Nutzer und der Zuweisung einzelner Zugriffsrechte auf bestimmte Daten. Insbesondere steht SAML für eine einheitliche, hinreichend flexible und sichere Übertragungsmethode dieser Daten. Leider ist AAI/VO noch nicht beendet und die Implementierung eines SAML-VOMS Systems für das D-Grid noch nicht verfügbar. Da D-MON zum derzeitigen Projektstand allerdings noch keine harten Anforderungen an die Sicherheit hat, d.h. Ressourcendaten aller VOs noch öffentlich verfügbar sein dürfen, genügt es zunächst, eine spätere Erweiterung auf SAML vorzusehen. Dennoch sollte es bereits möglich sein, bestimmte VOs zu übergeben, für welche Monitoringdaten übergeben werden sollen. Wenn dann eine spätere Integration von SAML-VOMS vorgenommen wird, muss dann nur noch die für einen Benutzer mögliche Auswahl an VOs beschränkt werden.

3.5 Standard-konforme Zugriffsschnittstellen

Neben der Verwendung eines aktuellen standardisierten Informationsmodells wie GLUE 2.0 sollen in D-MON auch WS-RF kompatible Schnittstellen als bevorzugt unterstützte Zugriffsverfahren auf den Dienst geschaffen werden, so wie sie durch das Open Grid Forum standardisiert wurden. Dies erlaubt die nahtlose Integration des D-MON Monitoringdienstes in OGSA-kompatible Grid-Infrastrukturen.

Als bereits in weiten Teilen realisierte Schnittstellen-Komponente für den Datenbankzugriff auf die D-MON Datenbank bietet sich hierfür die Anbindung OGSA-DAI (OGSA-Database Access Interface) an, welches relativ einfach auf die D-MON MySQL-Datenbank aufgesetzt werden kann. OGSA-DAI unterstützt sowohl SQL-Abfragen über WS-RF, als auch XML-Abfragen und erlaubt ferner eine Föderation mehrerer OGSA-DAI Zugriffspunkte. Es existiert ein Add-In Modul, welches eine Autorisierung per VOMS erlaubt. Ferner existiert ein Prototyp, der OGSA-DAI und SAML koppelt. Letztere Lösung sollte mittelfristig angestrebt werden, nicht zuletzt, um auch mit den aktuellen Entwürfen aus dem IVOM Projekt Schritt zu halten.

¹ Security Assertion Markup Language

3.6 Föderation mehrerer D-MON Dienste

Der Aufbau des D-MON Systems kann zentral erfolgen, d.h. es gibt einen einzigen integrierenden Dienst mit Repository, oder aber föderiert, d.h. es gibt mehrere dieser Dienste, welche jeweils an einzelnen Sites aufgesetzt werden und die Daten untereinander austauschen. Hierfür können die unterliegenden Datenbanken föderiert aufgesetzt werden. Auch ist es möglich, OGSA-DQP als verteilten Anfrageprozessor zu verwenden oder spezielle Protokolle zur Verteilung der Daten zu implementieren. Für den Prototypen soll zunächst ein zentrales Setup gewählt werden, welches sich einfacher aufsetzen und warten lässt.

3.7 Nutzer in verschiedenen Rollen

Die künftigen Nutzer des Systems treten in verschiedenen Rollen auf, z.B. als einfacher Benutzer oder als VO-Repräsentant. Für die Realisierung der Zugriffskontrolle auf die in der DMON Datenbank aggregierten Daten wurden in der Anforderungsanalyse [AP1.1] die verschiedenen Rollen definiert. Jede dieser Rollen entspricht einer durch einen Filter limitierten Sicht auf die Daten. In den folgenden Abschnitten werden diese Rollen, d.h. ihre Funktion und benötigte Informationen, die diesen zugeordnet sind, beschrieben, um anschließend einige grundlegende Portlets für das Benutzerinterface entwerfen zu können, aber auch um entsprechende Authentifizierungsmechanismen implementieren zu können.

Jede der hier genannten Nutzergruppen verfolgt verschiedene Interessen und hat deswegen verschiedene Anforderungen an die Informationen, die ihr angezeigt werden und die Repräsentation derselben. Im Folgenden wird, als Grundlage für die Erstellung entsprechender Templates, das Informationsbedürfnis einzelner beispielhafter Rollen noch einmal konkretisiert.

3.7.1 VO-Nutzer/Endnutzer

Ein Benutzer in einer VO nutzt deren Ressourcen, um Daten zu speichern und Jobs ausführen zu lassen. Benutzer haben verschiedene Ansätze zur Ressourcenauswahl, so ist es manchen egal wo ihre Anwendung gerechnet wird, während andere gezielt eine ganz bestimmte Ressource auswählen wollen. Dazu benötigen die Benutzer Informationen über die Ressourcen und deren Auslastung. Daneben sollten Benutzer Informationen über den Fortgang ihrer Jobs bzw. Workflows sehen, wobei dies bei engen Kollaborationen auch die Jobs der gesamten VO einschließen könnte. Als Basis sollte aber ein Benutzer nur seine eigenen Jobs sehen können.

3.7.2 VO-Repräsentant

Ein Repräsentant einer VO hat die Aufgabe, die zur VO gehörenden Benutzer und die der VO zugehörigen Ressourcen zu verwalten. Das Anlegen einer neuen VO wird vom VO-Repräsentanten nach Absprache mit dem betroffenen Community-Sprecher und den Ressourcen-Gebern beantragt. Das eigentliche Anlegen (und auch Löschen) einer VO erfolgt durch die zentrale Stelle, die die VOMRS Services betreibt. Zur Erfüllung seiner Aufgaben sollte der VO-Repräsentant vom Monitoringsystem Informationen über den Zustand der Ressourcen einer VO sowie den Umfang der Aktivitäten der VO-Nutzer erhalten. Dies umfasst unter anderem die Auslastung der Ressourcen einer VO sowie die Anzahl der Jobs.

3.7.3 Zentrale Infrastrukturdienste und D-Grid Operation Center

Zentrale Grid-Dienste wie beispielsweise Scheduler gehören nicht zu einer bestimmten VO und werden (ggf. künftig) von den Betreibern der D-Grid Infrastruktur bereitgestellt.

Wesentliche Informationen, die von diesen Diensten benötigt werden sind Verfügbarkeit, aktuelle Kapazitäten, Benchmarks sowie quasi-statische Informationen über die Ressourcen. Laut D-Grid Betriebskonzept soll ein D-Grid Operation Center (D-GOC) mit Grid-Operateuren (Grid Operator im Dienst, GOD) bei ständiger Erreichbarkeit aufgebaut werden. Ziele für die Schaffung eines GOC sind u.a. eine zeitlich durchgehende Unterstützung von Nutzern, VOs und Ressourcen-Providern bei Problemen und Fragen sowie eine durchgehend mögliche Zuordnung von Trouble-Tickets zu VOs, oder Sites. Ein D-GOC benötigt zu diesem Zweck die Verfügbarkeitsinformationen aller Ressourcen und Middleware-Dienste sowie historische Daten und Wartungsinformationen.

4 Modulares Design des D-MON Informationsdiensts

Im obigen Kapitel wurden die Problemstellungen für die zu entwickelnden Komponenten analysiert und geeignete Lösungen vorstrukturiert. Im Folgenden werden nun gezielt Module spezifiziert, um diesen Problemstellungen zu entsprechen.

4.1 Automatische Datenmigration

Für die Funktionalitäten eines automatischen Datenmigrationsprozesses werden die folgenden softwaretechnischen Module definiert.

4.1.1 Extract

Im Rahmen des D-MON Projektes kommen die Monitoring- und Informationsdienste UNICORE CIS, MDS4 DefaultIndexService sowie BDII als Datenquellen in Frage.

Am Ende des Extraktionsprozess stehen XML-Dateien mit den Inhalten dieser Datenquellen. Es werden die folgenden Softwarekomponenten zur Extraktion der Daten verwendet:

- MDS4 wsrf-query
- UNICORE Command Line Client
- LDAP 2 XML

4.1.2 Transform

Die Datentransformation der im Extraktionsprozess gewonnenen XML-Dateien von den Quelldatenmodellen zu dem gemeinsamen Zieldatenmodell erfolgt mit Hilfe von XSL-Transformationen. Eine genaue Spezifikation, wie die aktuelle Attribute der im Prototypsystem gegebenen Quellschemata auf die Zielsprache GLUE 2.0 abgebildet werden, findet sich im Anhang A. Das Ergebnis des Transformationsprozesses ist eine entsprechend vorbereitete SQL-Datei, welche Anweisungen für das Eintragen der Daten in die auf GLUE 2.0 basierenden Tabellen einer SQL-Datenbank enthält.

4.1.3 Load

Die nunmehr extrahierten und transformierten Monitoringdaten in Form der SQL Anweisungs-Datei werden nun mit Hilfe eines SQL-Clientprogrammes in die zentrale Datenbank hochgeladen.

Das Ergebnis ist in Abbildung 3 zu sehen, sie zeigt integrierte Einträge aus verschiedenen Monitoringdiensten.

name	type	domainID	informa...	sourceAddr	insertTime
UNICORE 6 Target System Factory Service			CIS	134.94.105.15	2008-09-16 14:18:37.0
alnitak.ari.uni-heidelberg.de	org.teragrid.ws-gram	AstroGrid-D	MDS4	129.187.254.39	2008-08-07 11:25:16.0
anticyclone.dkrz.de	org.teragrid.ws-gram	dkrz.de	MDS4	129.187.254.39	2008-08-07 11:25:17.0
aprilia.izbi.uni-leipzig.de	org.teragrid.ws-gram		MDS4	129.187.254.39	2008-08-07 11:25:16.0
arminius-grid.uni-paderborn.de	org.teragrid.ws-gram	PC2	MDS4	129.187.254.39	2008-08-07 11:25:16.0
astar.aip.de	org.teragrid.ws-gram	AstroGrid-D	MDS4	129.187.254.39	2008-08-07 11:25:16.0
astrodata01.gac-grid.org	org.teragrid.ws-gram	AstroGrid-D	MDS4	129.187.254.39	2008-08-07 11:25:16.0
bladekemper21.informatik.tu-muenchen.de	org.teragrid.ws-gram	AstroGrid-D	MDS4	129.187.254.39	2008-08-07 11:25:16.0
buran.aei.mpg.de	org.teragrid.ws-gram	AstroGrid-D	MDS4	129.187.254.39	2008-08-07 11:25:16.0
c3grid-gt.E-Technik.Uni-Dortmund.DE	org.teragrid.ws-gram	c3grid-gt.e-t...	MDS4	129.187.254.39	2008-08-07 11:25:17.0
c3grid.rrz.uni-koeln.de	org.teragrid.ws-gram	de.uni-koeln	MDS4	129.187.254.39	2008-08-07 11:25:17.0
cashmere.aip.de	org.teragrid.ws-gram	AstroGrid-D	MDS4	129.187.254.39	2008-08-07 11:25:16.0
ce-1-fzk.gridka.de	pbspro-gLite3	FZK-LCG2	BDII	iwrmon-bdii.fzk.de	2008-09-19 17:13:20.0
ce-2-fzk.gridka.de	pbspro-gLite3	FZK-LCG2	BDII	iwrmon-bdii.fzk.de	2008-09-19 17:13:20.0
ce-3-fzk.gridka.de	pbspro-gLite3	FZK-LCG2	BDII	iwrmon-bdii.fzk.de	2008-09-19 17:13:20.0

Abbildung 3: Einträge aus verschiedenen Monitoringsystemen in der D-MON Datenbank

4.2 Die Monitoring-Datenbank

Die im Prototypen verwendete Datenbank basiert auf einem MySQL-System, welches das derzeit aktuelle Referenzdatenschema von GLUE 2.0 verwendet. Dieses Schema wurde von der OGF Arbeitsgruppe GLUE entwickelt und wird auf den entsprechenden Seiten zum Download angeboten.

4.2.1 Initialisierung der Datenbank

Die Initialisierung der Datenbank erfolgt mit Hilfe des Referenzdatenschemas von GLUE 2.0, sowie der Einbindung mehrerer entfernter Tabellen des Remote Managements. Des Weiteren wurden Trigger zur automatischen Verarbeitung von VO Informationen sowie mehrere Views und Funktionen zur Generierung von VO spezifischer Sichten erstellt.

4.2.2 Schnittstelle zum VO-Management

Die Anbindung des VO-Managements erfolgt durch eine direkte Einbindung der Datenbanken von VOMS und GRRS auf Datenbankebene. Die entsprechende Zuordnung von Ressourcen und Services zu VOs wurde durch einen Trigger realisiert, der aktiviert wird, wenn Monitoring Daten durch die Informations-Provider eingefügt oder erneuert werden.

4.2.3 Trigger

Der Trigger erzeugt für jede Kombination aus Ressource / Service und erlaubter VO einen eigenen Datensatz in der GLUE 2.0 Relation „AccessPolicy“. Diese Datensätze werden im Anschluss für die Erstellung von VO spezifischen Sichten verwendet.

4.2.4 OGSA-kompatibler Datenbankzugriff

Um einen zum OGSA-Standard kompatiblen Zugriff auf die integrierten Monitoringdaten zu erlauben, stellt D-MON eine Anbindung an OGSA-DAI bereit. Hierfür wurde eine DMON-Datenresource definiert, auf welche per OGSA-DAI zugegriffen werden kann.

Für das D-MON Projekt wurde ein Command Line Client entwickelt, der durch das Setzen einer SQL-Umgebungsvariable die Auswahl einer bestimmten VO durch einen anfragenden Dienst erlaubt. Damit dies fehlerfrei möglich ist, wurde OGSA-DAI in Rücksprache mit den Entwicklern auch verbessert, so dass Umgebungsvariablen nun auch session-persistent akzeptiert werden.

Die Erstellung VO-basierter Views erfolgt dann aufgrund der gesetzten Datenbankvariable sowie der Information aus der GLUE 2.0 Relation „AccesPolicy“ jeweils für eine bestimmte VO.

4.3 Föderation von Datenbanken mehrerer Ressourcenanbieter

Zunächst wurde eine zentrale Datenbank für den Prototypen gewählt. In einem nächsten Schritt soll diese föderiert werden. Dabei ist es wünschenswert, dass verschiedene relationale Datenbanken bei den Ressourcenanbietern eingesetzt werden können. Erste Tests ergaben die Möglichkeit der Föderation mehrerer SQL-Datenbanken, MySQL, Postgre SQL, ... können in einer Föderation zusammenarbeiten.

4.4 Benutzerschnittstellen der Monitoring-Datenbank

4.4.1 Command-Line Client

Insbesondere für Anfragen durch andere Grid-Dienste im D-Grid, die ggf. auf der Basis verschiedener Middlewares arbeiten, aber auch für versierte Benutzer, ist ein Command-Line Client notwendig. Der OGSA-DAI Command Line Client wurde erweitert, so dass er zusätzlich zur eigentlichen SQL Abfrage einen weiteren Parameter unterstützt, welcher die VO spezifiziert, in deren Kontext die DB-Anfrage ausgeführt werden soll. Hierzu waren eigene Entwicklungen notwendig, da das Ausführen mehrerer SQL Kommandos in derselben Session nativ mit OGSA-DAI nicht möglich war.. Der D-MON OGSA-DAI Client nimmt SQL Anfragen auf GLUE2.0 Tabellen an. Abbildung 4 zeigt beispielhaft das Ergebnis einer Abfrage von vorhandenen ComputingServices im Kontext der VO „vi“.

```
SQLQuery: SELECT name, type, totalJobs, runningJobs, waitingJobs, informationProvider FROM ComputingService_VO
VO: vi
uk.org.ogsadai.resource.request.status.COMPLETED
```

name	type	totalJobs	runningJobs	waitingJobs	informationProvider
arminius-grid.uni-paderborn.de	org.teragrid.ws-gram	0	0	0	MDS4
dgiref-globus.fzk.de	org.teragrid.ws-gram	0	0	0	MDS4
dgrid-glitecl.rz.rwth-aachen.de	sge-dgiseq	null	null	null	BDII
dmon-unic.fz-juelich.de\$9115\$FZJ-JUGGLE-DMON	de.fzj.unicore.tsf	0	0	0	CIS
udo-gt01.grid.uni-dortmund.de	org.teragrid.ws-gram	151	151	0	MDS4
udo-gt03.grid.tu-dortmund.de	org.teragrid.ws-gram	0	0	0	MDS4

Abbildung 4: Ausgabe des D-MON OGSA-DAI Kommandozeilen-Klienten

4.4.2 Rollenbezogene Web-Portlets für Gridsphere

Für die Anwendung im World Wide Web wurde ein Gridsphere-Portal entwickelt, welches für verschiedene Nutzergruppen verschiedene Portlets zur Verfügung stellt.

Jede dieser Nutzergruppen verwendet die Informationen für einen anderen Zweck. Deswegen unterscheiden sie sich maßgeblich in ihrem Bedürfnis nach der Art der Information, deren Granularität und der Darstellung oder Repräsentation der Information innerhalb des Portals. In den nun folgenden Abschnitten werden zuerst die von jeder Nutzergruppe/Rolle benötigten Informationen aufgeführt und dann die verschiedenen Möglichkeiten zur Darstellung dieser Informationen vorgestellt. Hierbei wird der Fokus besonders darauf gerichtet nur Informationen anzuzeigen, die für die Nutzergruppe von Interesse sind. Ziel

dabei ist es, dass diese Informationen gut aufbereitet und schnell erfassbar sind, um die Hürden für die Benutzung der Plattform möglichst gering zu halten.

Für die Portlets sind, wie im Anforderungsdokument in [AP1.1] skizziert, die Folgenden Ansichten notwendig. Derzeit wird noch prototypisch an einer Umsetzung für erste Versionen solcher Portlets gearbeitet.

4.4.2.1 VO–Nutzer/Endnutzer

Für das Ressourcen- und Job-Monitoring durch den Endnutzer müssen die folgenden Daten zur Verfügung gestellt werden:

- Statische Information der Rechen-Ressourcen (Anzahl der Nodes, Anzahl der cores, RAM, Ressource Management System, Interconnect, Architektur, Betriebssystem,...),
- Statische Information der Datenressourcen (Kapazität, gesamt, für die VO und für den Benutzer, Zugriffsmöglichkeiten),
- Aktueller Status der Ressourcen (up, down oder in Wartung, Anzahl der Jobs gesamt, aus der VO, des Benutzers),
- Aktueller Status der Datenressourcen (belegter Platz durch VO und durch Benutzer),
- Detaillierter Job-Status für alle Jobs des Benutzers und gegebenenfalls auch der VO.

4.4.2.2 VO-Repräsentant

Dem VO-Repräsentanten müssen Informationen über die zur VO gehörenden Benutzer und ihre Aktivitäten auf den Ressourcen sowie Informationen zu den Ressourcen selbst zur Verfügung gestellt werden:

- Für alle Ressourcen, die von der VO genutzt werden dürfen:
 - Ort,
 - Organisation,
 - Name der Ressource,
 - Kurzbeschreibung der Rechen-Ressource (statische Information: Anzahl der Nodes, Anzahl der Cores, RAM, Ressource Management System, Interconnect, Architektur, Betriebssystem) bzw. Kurzbeschreibung der Datenressource (Kapazität, ...),
 - Status der Ressource (dynamische Information: up, down oder in Wartung, Anzahl der Jobs gesamt und aus der VO),
- Status der zur VO gehörenden Services (Verfügbarkeit, Service spezifische Informationen, ...),
- Liste mit Status der zur VO gehörenden Jobs (über alle Ressourcen),
- Übersicht über das aktuell zur VO gehörende Datenvolumen, akkumuliert und pro Ressource,
- Übersicht über alle Benutzer in der VO mit den gespeicherten Eckdaten.

4.4.2.3 Zentrale Infrastrukturdienste

Für weitere zentrale Grid-Dienste wie beispielsweise Scheduler, müssen die folgenden Daten bereitgestellt werden:

- VO spezifische Informationen, siehe hierzu die obigen Angaben für VO-Nutzer und VO-Repräsentant
- Statistisch-historische Daten, wie:
 - EstimatedResponseTime,
 - Tests (Ergebnisse von durchgeführten Tests in einem bestimmten Zeitraum),

- Benchmarking-Ergebnisse und quasi-statische Information,
 - Status (befindet sich die Ressource im Produktions- oder Testzustand),
 - Policy (maximal erlaubte CPU Benutzung, Maximale Wartezeit für einen Job),
 - Jobs (aktuelle Anzahl von wartenden Jobs, Anzahl von ausgeführten Jobs usw.),
 - Prices (Nutzungskosten für fortgeschrittene Scheduling Dienste, die Ressourcen über ein Geschäftsmodell vermitteln),

4.4.2.4 D-Grid Operation Center

Der Blickwinkel eines GOC auf die Infrastruktur des Grids ist aufgrund seiner Aufgaben sehr global. Zur Lösung sehr spezieller Probleme kann aber ein sehr hoher Detailgrad im Monitoring notwendig sein. Unter Umständen muss das GOC auch Nutzersichten nachvollziehen können, eine sehr flexible Darstellung der Informationen im Monitoring Portal ist deshalb wichtig.

Für ein zentrales D-GOC müssen die folgenden Daten bereitgestellt werden:

- VO spezifische Informationen, siehe hierzu die obigen Angaben für VO-Nutzer und VO-Repräsentant,
- Statistisch-historische Daten, siehe hierzu die obigen Angaben für VO-Repräsentant,
- Software Versionen der Anwenderprogramme,
- Kontaktdaten der Verantwortlichen/Ansprechpartner für jeden Aufgabenbereich.
- Wartungsmeldungen

5 Gesamtsystem

In diesem Kapitel werden zusammenfassend noch einmal alle D-MON Module genannt, so wie sie dann im Release-Paket zusammengefaßt werden:

- Server
 - ETL-Komponenten für Globus MDS4, gLite BDII und UNICORE CIS installierbar je Site oder je Monitoringsystem
 - SQL-Konfiguration
 - ETL-Komponente zum Einfügen von GRRS-Daten in die Datenbank
 - Gridsphere Konfiguration und Portlets
- Client
 - OGSA-DAI Client

Ein Gesamtmodell mehrerer, domänen- (bzw. Ressourcenanbieter-) übergreifender und föderiert zusammenarbeitender D-MON Module an verschiedenen Standorten zeigt Abbildung 4.

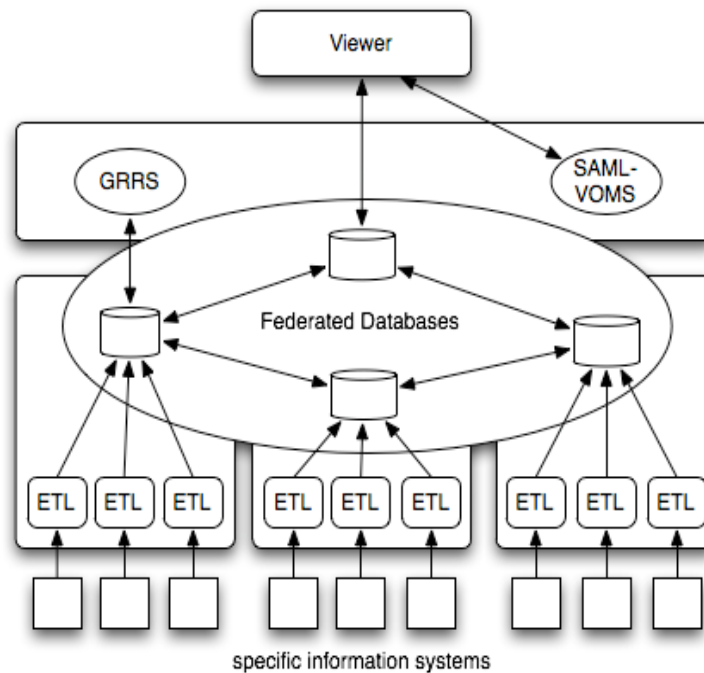


Abbildung 5: Übersicht über das Gesamtsystem

5.1 Benutzer-Interfaces

Für den Benutzer stehen eine Kommandozeilen- und ein Webclient zur Verfügung. Der D-MON Command-Line Client ist im Wesentlichen ein OGSA-DAI Client, welcher in der Lage ist, SQL-Queries an die D-MON Datenbank zu beantworten.

Der Webclient wird durch eine Gridsphere-Server Installation und einem Satz Portlets, welche speziell für den D-MON Informationsdienst bereit gestellt werden, realisiert.

5.2 Servermodule

Serverseitig stellt D-MON einen Satz aus Modulen bereit, die an jeder D-Grid Site installiert werden können, um einen site- und middleware-übergreifenden Informationsdienst zu realisieren.

In Abbildung 5 sind die verschiedenen Middlewares, gekennzeichnet mit MW, sowie der ETL-Prozess, die D-MON Datenbank wie auch das standardisierte Zugriffsinterface zu erkennen.

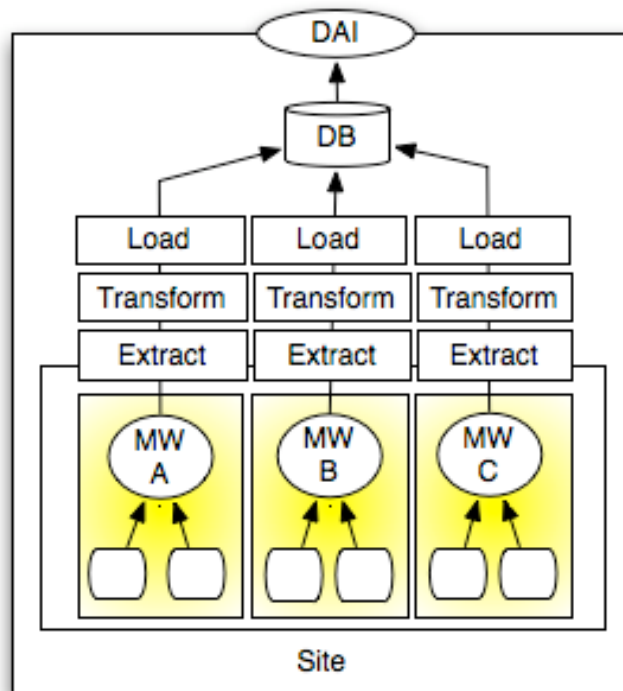


Abbildung 6: Monitoring-Modul

Literatur

- [Ba08] Baur, T., Functional Analysis and Architecture for Interoperable and DVO-specific Grid Monitoring Services, In Proceedings of the Fourth IEEE International Conference on eScience (eScience 2008) (to appear), IEEE Computer Society Press, Indianapolis, USA, Dezember, 2008.
- [ChDa97] Chaudhuri, S. and Dayal, U.: An overview of data warehousing and OLAP Technology. SIGMOD Rec. 26, 65-74, 1997.
- [Gü07] Gürich, W. et. Al.: Betriebskonzept für die D-Grid Infrastruktur, D-Grid Report, 2007
- [MiSchZi07] Milke, J-M., Schiffers, M., Ziegler, W.: Rahmenkonzept für das Management Virtueller Organisationen im D-Grid (Version 1.1), D-Grid Report, 2007
- [AP1.1] Sammlung von Anforderungen, D-MON Projekt Deliverable, Januar 2008 http://www.d-grid.de/fileadmin/user_upload/documents/D-MON/Anforderungsanalyse-Version1.0.pdf
- [AP1.2] Middleware-übergreifendes Monitoring: Evaluierung und Auswahl von Komponenten, D-MON Project Deliverable, February 2008 http://www.d-grid.de/fileadmin/user_upload/documents/D-MON/Komponenten-AP2.1-final.pdf
- [AP2.1] Planung und Aufbau Testbett, D-MON Project Deliverable, March 2008 http://www.d-grid.de/fileadmin/user_upload/documents/D-MON/AP2_2-final.pdf
- [AP2.2] Planung und Durchführung von Tests, D-MON Project Deliverable, March 2008 http://www.d-grid.de/fileadmin/user_upload/documents/D-MON/AP2_2-final.pdf

- [DGI-AAI] Authentication and Authorization Infrastructure, Teil des D-Grid Integrationsprojekts
<http://dgi.d-grid.de/index.php?id=100>
- [IVOM] Interoperabilität und Integration der VO-Management Technologien im D-Grid (IVOM)
<http://www.d-grid.de/index.php?id=314>
- [DGI2-FG3] AAI/VO im DGI-2 Fachgebiet3 (Sicherheit)
http://www.rrzn.uni-hannover.de/dgi-2_fg-3.html#c12378

6 ANHANG A – Mapping der Informationsmodelle

Mapping der in den Anforderungen im Dokument zu AP 1.1 genannten Kennzahlen auf das GLUE 2.0 Schema (siehe Datei Mapping Tabellen.doc und GT4 Mapping.doc)

6.1 Globus

GT4 GLUECE to GLUE 2.0 Mappings

<i>Objectclass</i>	<i>Attribute</i>		<i>GLUE20.AdminDomain</i>
Site	UniqueID	→	ID
Site	Name	→	Name
Site	Description	→	Description

<i>Objectclass</i>	<i>Attribute</i>		<i>GLUE20.Location</i>
Site	UniqueID	→	LocalID
Site	Name	→	Name
Site	Location	!	Place
Site	Location	!	Address
Site	Location	!	Country
Site	Location	!	Postcode
Site	Longitude	→	Longitude
Site	Latitude	→	Latitude

<i>Objectclass</i>	<i>Attribute</i>		<i>GLUE20.Contact</i>
Site	UniqueID	→	adminDomainID
Site	SysAdminContact	→	URL
	„sysadmin“	→	localID
	„sysadmin“	→	Type

<i>Objectclass</i>	<i>Attribute</i>		<i>GLUE20.Contact</i>
Site	UniqueID	→	adminDomainID
Site	Web	→	URL
	„web“	→	localID
	“web”	→	Type

<i>Objectclass</i>	<i>Attribute</i>		<i>GLUE20.ComputingService</i>
ComputingElement	HostName	→	ID
ComputingElement	HostName	→	Name
	„org.teragrid.ws-gram“	→	Type
ComputingElement	State/RunningJobs	→	runningJobs
ComputingElement	State/WaitingJobs	→	waitingJobs
ComputingElement	State/RunningJobs+		
	State/WaitingJobs	→	totalJobs
Site	UniqueID	→	domainID

<i>Objectclass</i>	<i>Attribute</i>	<i>GLUE20.ComputingEndpoint</i>	
ComputingEndpoint	ID	→	ID
ComputingEndpoint	Name	→	Name
ComputingEndpoint	URL	→	URL
ComputingEndpoint	Technology	→	Technology
ComputingEndpoint	InterfaceName	→	InterfaceName
ComputingEndpoint	Implementor	→	Implementor
ComputingEndpoint	ImplementationName	→	ImplementationName
ComputingEndpoint	ImplementationVersion	→	ImplementationVersion
ComputingEndpoint	QualityLevel	→	QualityLevel
ComputingEndpoint	HealthState	→	HealthState
ComputingEndpoint	HealthStateInfo	→	HealthStateInfo
ComputingEndpoint	ServingState	→	ServingState

<i>Objectclass</i>	<i>Attribute</i>	<i>GLUE20.ComputingManager</i>	
ComputingElement	HostName	→	ID
ComputingElement	HostName	→	Name
ComputingElement	LRMSType	→	Type
ComputingElement	LRMSVersion	→	Version
ComputingElement	TotalCPUs	→	TotalPhysicalCPUs
ComputingElement	serviceID	→	serviceID

6.2 UNICORE

Der UNICORE 6 Common Information Service (CIS) stellt seine Daten bereits im GLUE 2.0 Schema zur Verfügung. Das zugehörige XSLT-Script hat hier nur die Funktion die Daten aus dem CIS in die relationale D-MON Datenbank einzuspeisen. Derzeit sammelt der CIS nur Informationen zum Ressourcen-Monitoring, Job-Monitoring Informationen stehen nicht zur Verfügung, wie die folgende Tabelle zeigt. Zu jeder Objektklasse werden in der Datenbank noch die Provenance Angaben informationProvider und providerAddr gespeichert.

UNICORE CIS (GLUE2.0) to GLUE 2.0 Mappings

<i>Objectclass</i>	<i>Attribute</i>	<i>GLUE20.AdminDomain</i>	
AdminDomain	ID	→	Id
AdminDomain	Name	→	Name
AdminDomain	Description	→	Description
AdminDomain	Distributed	→	Distributed
<i>Objectclass</i>	<i>Attribute</i>	<i>GLUE20.Location</i>	
Location	LocalID	→	LocalID
Location	Name	→	Name
Location	Address	→	Address
Location	Place	→	Place
Location	Country	→	Country
Location	Postcode	→	Postcode
Location	Latitude	→	Latitude

Location	Longitude	→	Longitude
<i>Objectclass</i>	<i>Attribute</i>		<i>GLUE20.Contact</i>
Contact	LocalID	→	LocalID
Contact	URL	→	URL
Contact	Type	→	Type
<i>Objectclass</i>	<i>Attribute</i>		<i>GLUE20.ComputingService</i>
ComputingService	ID	→	ID
ComputingService	Name	→	Name
ComputingService	Type	→	Type
ComputingService	QualityLevel	→	QualityLevel
ComputingService	TotalJobs	→	TotalJobs
ComputingService	RunningJobs	→	RunningJobs
ComputingService	WaitingJobs	→	WaitingJobs
<i>Objectclass</i>	<i>Attribute</i>		<i>GLUE20.ComputingEndpoint</i>
ComputingEndpoint	ID	→	ID
ComputingEndpoint	Name	→	Name
ComputingEndpoint	URL	→	URL
ComputingEndpoint	Technology	→	Technology
ComputingEndpoint	InterfaceName	→	InterfaceName
ComputingEndpoint	Implementor	→	Implementor
ComputingEndpoint	ImplementationName	→	ImplementationName
ComputingEndpoint	ImplementationVersion	→	ImplementationVersion
ComputingEndpoint	QualityLevel	→	QualityLevel
ComputingEndpoint	HealthState	→	HealthState
ComputingEndpoint	HealthStateInfo	→	HealthStateInfo
ComputingEndpoint	ServingState	→	ServingState
<i>Objectclass</i>	<i>Attribute</i>		<i>GLUE20.ComputingManager</i>
ComputingManager	ID	→	ID
ComputingManager	Name	→	Name
ComputingManager	TotalPhysicalCPUs	→	TotalPhysicalCPUs
ComputingManager	TotalLogicalCPUs	→	TotalLogicalCPUs
<i>Objectclass</i>	<i>Attribute</i>		<i>GLUE20.ExecutionEnvironment</i>
ExecutionEnvironment	ID	→	ID
ExecutionEnvironment	Platform	→	Platform
ExecutionEnvironment	PhysicalCPUs	→	PhysicalCPUs
<i>Objectclass</i>	<i>Attribute</i>		<i>GLUE20.ApplicationEnvironment</i>
ApplicationEnvironment	LocalID	→	LocalID
ApplicationEnvironment	Name	→	Name
ApplicationEnvironment	Version	→	Version

6.3 gLite

gLite BDII(GLUE1.2) to GLUE2.0 Mapping

<i>Objectclass</i>	<i>Attribute</i>	<i>GLUE20.AdminDomain</i>	
GlueSite	<i>DistinguishedName must be tokenized before (,mds-vo-name=local,o=grid') and after ('o-name=')</i>	→	ID
GlueSite	GlueSiteName	→	Name
GlueSite	GlueSiteDescription	→	Description
<i>Objectclass</i>	<i>Attribute</i>	<i>GLUE20.Location</i>	
GlueSite	GlueSiteUniqueID	→	LocalID
GlueSite	GlueSiteName	→	Name
GlueSite	GlueSiteLocation	→	Location
GlueSite	GlueSiteLocation	→	Address
GlueSite	GlueSiteLocation (<i>before</i> “,”)	→	Place
GlueSite	GlueSiteLocation	→	PostCode
GlueSite	GlueSiteLocation (<i>after</i> “,”)	→	Country
GlueSite	GlueSiteLatitude	→	Latitude
GlueSite	GlueSiteLongitude	→	Longitude
<i>Objectclass</i>	<i>Attribute</i>	<i>GLUE20.Contact</i>	
GlueSite	GlueSiteUniqueID	→	LocalID
GlueSite	GlueSiteDescription „Security“ „SysAdmin“ „UserSupport“ GlueSiteWeb	→	Type
GlueSite	GlueSiteUserSupportContact GlueSiteSysAdminContact GlueSiteSecurityContact	→	URL
GlueSite	GlueSiteSponsor	→	OtherInfo
GlueSite	GlueSiteOtherInfo	→	OtherInfo
<i>Objectclass</i>	<i>Attribute</i>	<i>GLUE20.ComputingManager</i>	
GlueCETop	GlueCEInfoApplicationDir	→	ApplicationDir
GlueCETop	GlueCEInfoDataDir	→	TmpDir
GlueCETop	GlueCEInfoGatekeeperPort	→	OtherInfo
GlueCETop	GlueCEInfoHostName	→	Name
GlueCETop	GlueCEInfoJobManager	→	OtherInfo
GlueCETop	GlueCEInfoLRMSType	→	Type
GlueCETop	GlueCEInfoLRMSVersion	→	Version
GlueCETop	GlueCEInfoDefaultSE	→	OtherInfo
GlueCETop	GlueCEInfoTotalCPUs	→	TotalPhysicalCPUs
<i>Objectclass</i>	<i>Attribute</i>	<i>GLUE20.ComputingService</i>	
GlueClusterTop	GlueClusterName	→	Name
GlueClusterTop	GlueInformationServiceURL	→	StatusPage
GlueClusterTop	GlueClusterUniqueID	→	ID
GlueSubCluster	GlueClusterService (<i>after</i> “jobmanager-“)	→	Type

<i>Objectclass</i>	<i>Attribute</i>		<i>GLUE20.ComputingShare</i>
GlueCETop	GlueCEName	→	Name
GlueCETop	GlueCEPolicyAssignedJobSlots	→	MaxTotalJobs
GlueCETop	GlueCEPolicyMaxCPUTime	→	MaxCPUTime
GlueCETop	GlueCEPolicyMaxRunningJobs	→	MaxRunningJobs
GlueCETop	GlueCEPolicyMaxTotalJobs	→	MaxTotalJobs
GlueCETop	GlueCEPolicyMaxWallClockTime	→	MaxWallTime
GlueCETop	GlueCEPolicyPriority	→	SchedulingPolicy
GlueCETop	GlueCEStateEstimatedResponseTime	→	EstimatedAverageWaitingTime
GlueCETop	GlueCEStateFreeJobSlots	→	FreeSlots
GlueCETop	GlueCEStateRunningJobs	→	RunningJobs
GlueCETop	GlueCEStateStatus	→	Description
GlueCETop	GlueCEStateTotalJobs	→	TotalJobs
GlueCETop	GlueCEStateWaitingJobs	→	WaitingJobs
GlueCETop	GlueCEStateWorstResponseTime	→	EstimatedWorstWaitingTime
GlueCETop	GlueCEUniqueID	→	LocalID

<i>Objectclass</i>	<i>Attribute</i>		<i>GLUE20.ExecutionEnvironment</i>
GlueCETop	GlueCEHostingCluster	→	LocalID
GlueSubCluster	GlueHostMainMemoryRAMSize	→	MainMemorySize
GlueSubCluster	GlueHostMainMemoryVirtualSize	→	VirtualMemorySize
GlueSubCluster	GlueHostNetworkAdapterInboundIP	→	ConnectivityIn
GlueSubCluster	GlueHostNetworkAdapterOutboundIP	→	ConnectivityOut
GlueSubCluster	GlueHostOperatingSystemName	→	OSName
GlueSubCluster	GlueHostOperatingSystemRelease	→	OSFamily
GlueSubCluster	GlueHostOperatingSystemVersion	→	OSVersion
GlueSubCluster	GlueHostProcessorClockSpeed	→	CPUClockSpeed
GlueSubCluster	GlueHostProcessorModel	→	CPUModel
GlueSubCluster	GlueHostProcessorVendor	→	CPUVendor
GlueSubCluster	GlueSubClusterUniqueID	→	LocalID
GlueSubCluster	GlueSubClusterPhysicalCPUs	→	PhysicalCPUs
GlueSubCluster	GlueSubClusterLogicalCPUs	→	LogicalCPUs

<i>Objectclass</i>	<i>Attribute</i>		<i>GLUE20.Benchmark</i>
GlueSubCluster	GlueSubClusterUniqueID	→	parentID
GlueSubCluster	SPECint2000	oder →	Type
GlueSubCluster	SPECfloat2000	oder →	Value
GlueSubCluster	GlueHostBenchmarkSI00	oder →	Value
GlueSubCluster	GlueHostBenchmarkSF00	oder →	Value
GlueSubCluster	GlueSubClusterName	→	LocalID

<i>Objectclass</i>	<i>Attribute</i>		<i>GLUE20.StorageService</i>
GlueSE	GlueSEUniqueID	→	ID
GlueSE	GlueSEName	→	Name

<i>Objectclass</i>	<i>Attribute</i>		<i>GLUE20.StorageEndpoint</i>
GlueSE	GlueSEUniqueID	→	ID
GlueSE	GlueSEName	→	Name
GlueSE	GlueSEPort	→	URL
GlueSE	GlueSEArchitecture	→	Technology
GlueSE	GlueInformationServiceURL	→	ImplementationVersion