



Arbeitspaket 2: Blaupausen und Beratung

Community - Überblick / Report ¹

Deliverable	2.1.1 Community - Überblick / Report
Autoren	Arbeitspaket 2: Blaupausen und Beratung
Editoren	A. Rapp, C. Grimme, H. Enke
Datum	23-10-2009
Dokument Version	1.0.0

A: Status des Dokuments

Deliverable 2.1.1, Version 1.0.0. Das Dokument ist durch das Projekt akzeptiert.

B: Bezug zum Projektplan

Die Materialsammlung dient zur ersten Aufbereitung der Erfahrungen der CGs.

C: Abstract

Die Material-Sammlung enthält in einer strukturierten Form sowohl direkte Stellungnahmen zu Aspekten der Community-Grids wie auch kurze Statements oder Referenzen, die für eine eingehende Analyse eine Übersicht und Orientierungshilfe bildet.

¹This work is created by the WissGrid project. The project is funded by the German Federal Ministry of Education and Research (BMBF).

D: Änderungen

Version	Date	Name	Brief summary
0.1.0	08.07.2009	H. Enke	Working Draft Creation
0.1.1	24.07.2009	S. Kindermann, T. Rathmann	Working Draft, climate research added
0.1.2	28.07.2009	A. Aschenbrenner	Working Draft, TextGrid
0.1.3	29.07.2009	B. Löhnhardt	Überarbeitung + Draft, MediGrid
0.1.4	29.07.2009	F. Schlünzen	Draft, HEP
0.1.5	02.08.2009	H. Enke	Draft, AstroGrid-D
0.1.6	22.07.2009	C. Grimme	Struktur für LCM-Evaluation
0.2.0	02.08.2009	H. Enke	Integration into one document
0.2.1	11.08.2009	K. Rieger	Additions to AstroGrid-D
0.2.2	19.08.2009	H. Enke	Added a section about documents and guides
0.2.3	25.08.2009	F. Schintke	Ergänzung C3Grid LCM, Docs, sonstige Korrekturen
0.2.4	27.08.2009	B. Löhnhardt	Ergänzung MediGRID LCM, Docs, sonstige Korrekturen
0.2.5	07.09.2009	T. Rathmann, S. Kindermann, B. Fritzsich	Struktur Support-Fragen, Ergänzung Support, Docs, sonstige Korrekturen
0.2.6	08.09.2009	B. Löhnhardt	Ergänzung MediGRID Support
0.2.7	29.09.2009	F. Schlünzen	Ergänzung HEPGrid Support
0.2.8	30.09.2009	F. Schlünzen	Ergänzung HEPGrid Organisation
0.3.0	13.10.2009	C. Grimme	Redaktion Kap. 4,7
0.3.1	14.10.2009	A. Rapp	Redaktion Kap. 5,6
0.3.2	14.10.2009	H. Enke	Redaktion Kap. 2,3, Einleitung Draft
0.4.0	16.10.2009	H. Enke, C. Grimme	Edits
0.4.1	20.10.2009	T. Rathmann	Edits
0.4.2	21.10.2009	O. Schmid	Textgrid Changes
0.4.3	21.10.2009	H. Enke	Edits
0.5.0	23.10.2009	H. Enke	Summary, published to the project for comments, as voted by the workgroup as of 22.10.2009.
1.0.0	09.11.2009	H. Enke	Included the comments, revised some grammar etc.

E:

Inhaltsverzeichnis

1	Summary	6
2	Einleitung	7
3	Skizze der inneren Organisationsstruktur	7
3.1	AstroGrid-D	8
3.2	HEP Grid	9
3.3	C3 Grid	11
3.4	TextGrid	11
3.5	MediGrid	12
4	Skizze der Dienste und des Aufbaus des CG - Architekturkonzepte	13
4.1	AstroGrid-D	15
4.2	HEP Grid	17
4.3	C3 Grid	18
4.4	TextGrid	19
4.5	MediGrid	20
5	Software-Lifecycle-Management	22
5.1	Aspekte des Software-Lifecycle-Managements	23
5.1.1	Ermittlung und Umgang mit Anforderungen	23
5.1.2	Organisationsstrukturen und Vorgehensmodelle	24
5.1.3	Umgang mit Risiken	24
5.1.4	Testumgebungen, Testverfahren und Qualitätsmanagement	24
5.1.5	Lebenszyklus nach der Fertigstellung und Nachhaltigkeit	25
5.2	Life-Cycle-Management im AstroGrid-D	25
5.2.1	Allgemeine Angaben zum AstroGrid-D-Projekt	25
5.3	Life-Cycle-Management im C3Grid	26
5.3.1	Allgemeine Angaben zum C3Grid Projekt	26
5.3.2	Portal (Partner AWI):	31

5.3.3	Workflow Scheduling Service (Partner TUDo)	32
5.3.4	Datenmanagement Service (Partner ZIB)	34
5.4	Life-Cycle-Management im HEP-CG	36
5.4.1	Allgemeine Angaben zum HEP-CG-Projekt	36
5.5	Lifecycle Management im MediGrid-Projekt	36
5.5.1	Allgemeine Angaben zum MediGrid-Projekt	36
5.5.2	Grid Workflow Execution Service (Fraunhofer FIRST)	37
5.6	Life-Cycle-Management im TextGrid	38
5.6.1	Allgemeine Angaben zum TextGrid Projekt	38
5.7	Vorläufige Bewertung	44
6	Interaktion mit der Community	45
6.1	AstroGrid-D	46
6.2	HEP Grid	46
6.3	C3 Grid	46
6.4	TextGrid	47
6.5	MediGrid	47
7	Support	47
7.1	AstroGrid-D	48
7.1.1	Benutzeranleitungen und andere schriftliche Support-Dokumente	48
7.1.2	Training	49
7.1.3	Weitere Support-Strukturen	49
7.2	HEP Grid	50
7.2.1	Benutzeranleitungen und andere schriftliche Support-Dokumente	50
7.2.2	Training	50
7.2.3	Weitere Support-Strukturen	51
7.3	C3 Grid	52
7.3.1	Benutzeranleitungen und andere schriftliche Support-Dokumente	52
7.3.2	Training	52
7.3.3	Weitere Support-Strukturen	53
7.4	TextGrid	53

7.4.1	Benutzeranleitungen und andere schriftliche Support-Dokumente	53
7.4.2	Training	54
7.4.3	Weitere Support-Strukturen	55
7.5	MediGrid	55
7.5.1	Benutzeranleitungen und andere schriftliche Support-Dokumente	55
7.5.2	Training	56
7.5.3	Weitere Support-Strukturen	57
8	Dokumente und Materialien aus den CG-Grids zur Weiterverwertung	57
8.1	AstroGrid-D	58
8.1.1	Dokumentation von allgemeinem Interesse	58
8.1.2	Dokumente bezüglich AstroGrid-D	58
8.1.3	Ausgewählte Publikationen	59
8.2	HEP Grid	60
8.2.1	Dokumente von allgemeinem Interesse	60
8.2.2	Dokumentation bezüglich HepCG	61
8.3	C3 Grid	62
8.3.1	Dokumentationen von allgemeinem Interesse	62
8.3.2	Dokumentationen bezüglich C3Grid	63
8.3.3	Ausgewählte Publikationen	64
8.3.4	Web-Seiten	65
8.4	TextGrid	65
8.4.1	Dokumente von allgemeinem Interesse	66
8.4.2	Dokumentation bezüglich TextGrid	66
8.5	MediGrid	68
8.5.1	Dokumentation von allgemeinem Interesse	68
8.5.2	Dokumentation bezüglich MediGRID	68

1 Summary

This document contains a collection of answers to questions put to the D-Grid Call 1 Community-Grids. It is meant as a collection of material for further study, providing easy access to otherwise mostly published, but very randomly distributed material. Since the working group is preparing in depth studies of approaches to build and run a Community Grid (CG), this document is a starting point for drill down.

Then, it is a reference document for the WissGrid team of expert consultants to build on the experiences of the CG of D-Grid.

The document is not (yet) a complete compendium of CG experiences, solutions and services and will grow through added entries from the CGs.

How to read the document:

- For a high-level view: just digest the introduction to each chapter. (For Life Cycle Management: read Sub: Evaluation)
- For technical details: some of the chapters are already more elaborately structured, (e.g. the Life Cycle Management), others are just hints to further reading
- For in depth study: read the chapters, follow the links

2 Einleitung

Der vorliegende Bericht nimmt eine Sichtung von Arbeiten der in der ersten D-Grid Phase gegründeten Community Grids (CG) vor. Entlang der Fragestellungen, die von den Kapitelüberschriften angedeutet sind, wurden Zusammenfassungen, Berichte und Literatur gesammelt. Die Intention hierbei ist, einen leicht zu benutzenden Überblick bereitzustellen, der als Material die nächsten Arbeitsschritte begleitet, strukturelle Lücken erkennen hilft und die über viele Websites und andere Publikationsorte verteilten Dokumente sammelt. Der Bericht erhebt keinen Anspruch auf Vollständigkeit und ist als ein ergänzungsfähiges Dokument konzipiert, sodass im Lauf des Projekts weitere Materialien hinzugefügt werden. Wir möchten an dieser Stelle gleich darauf hinweisen, dass dieses Dokument *in keiner Weise* als Bilanz der Arbeiten der Community-Grids (CG) verstanden werden kann. Das Augenmerk der Fragen richtet sich notwendigerweise mehr auf die kritischen Aspekte.

In den Einleitungen zu den einzelnen Kapiteln wird eine erste kritische Durchsicht des Materials vorgenommen, die die Materialien in Hinblick auf kommende Deliverables einordnet und auch auf noch vorhandene Lücken hinweist.

3 Skizze der inneren Organisationsstruktur

Organisationsstrukturen für ein Community-Grid sind notwendig von vielen, auch zufälligen Konstellationen, z.B. der Partner untereinander, bestimmt. Ein weiterer wichtiger Faktor ist die Einbindung der Grid-Struktur in einen schon bestehenden Organisationszusammenhang, die eine selbständige Strukturbildung erheblich bestimmt oder als überflüssig erscheinen lässt, wie im Falle der Hochenergie-Physik. Die Materialien zeigen eine Vielfalt von Modellen, die im Detail auch bestimmt werden durch die jeweilige Projekt-Konstellation. Es gibt jedoch einige Elemente, die immer vorhanden sind (bzw. sein sollten):

Leitung: Eine Kombination eines Lenkungsgremium, welches die inhaltlichen Entscheidungen trifft, mit einer operativen Führung, die durch ein Projektmanagement unterstützt wird. Die operative Führung wird durch ein kleineres Gremium von Experten realisiert, das Lenkungsgremium versammelt die Entscheidungssträger.

Diese Struktur ist auch über die Phase der Projektförderung hinaus geeignet für CGs.

Arbeitsgruppen: Die Arbeitsgruppen sind vor allem in der Phase des Aufbaus der Strukturen zur Realisierung der Arbeitspakete, zur Realisierung von Implementationen von Software usw. wichtig. Im Verlauf der Herausbildung des CG und bei gefestigten Kommunikationskanälen werden Ad-hoc-Gruppen jedoch effizienter. Die Arbeitsgruppen benötigen auf jeden Fall feste Verantwortlichkeiten und regelmäßige Kommunikation durch Mailing-Listen und ein- bis zweimonatliche Telefonkonferenzen, sowie F2F-Meetings.

Kommunikation: Hier sind 4 Elemente zu nennen:

Meetings: Halbjährliche Plenar-Treffen, AG-Meetings nach Bedarf. Eine Kombination ist meist nicht günstig, da oft Mehrfach-Mitgliedschaft in AGs vorliegt.

Telefonkonferenzen: Monatliche oder längere Abstände für Telefon- (bzw. Video-) Konferenzen sind für die operative Arbeit des CG elementar. Es ist darauf zu achten, daß die Konferenzen nachvollziehbar durch zentral zugängliche Protokolle + Action-Items dokumentiert werden.

Mailing-Listen: Neben einer öffentlichen Liste sollten für jede AG und die Gremien je eine Liste zur Verfügung stehen.

Website: Das CG benötigt eine Website zur Außendarstellung und eine Intranet-Site. Die Lösungen sind sehr verschieden, ratsam ist eine technische Unterstützung für eine weitgehend von allen CG-Mitarbeitern selbst zu erstellende Site, d.h. ein CMS-System, das keine/wenig Code-Erfahrung erfordert. Die Technik sollte plattformunabhängig sein. Die Protokolle von Treffen sollten im Intranet zugänglich sein.

Repository: Die CG haben jeweils eigene Produkte oder Weiterentwicklungen, die als Pakete einer Versionskontrolle unterworfen werden müssen. Hier wird nahezu überall SVN eingesetzt, oft zusammen mit einem Issue-Tracker System, welches gezielte Bearbeitung von Code-Entwicklung durch mehrere Personen erst ermöglicht.

3.1 AstroGrid-D

Untenstehendes bezieht sich auf AstroGrid-D als Projekt im Call 1

- Leitung, (wie oft Treffen/Telco etc):
11 Partner, Konsortialführer AIP.
Etwa halbjährliche Treffen, Telefonkonferenzen des Steering Board monatlich
Telefonkonferenzen der ArchitekturGruppe wöchentlich
- AGD-Organisation:
Steering Board, Projektmanagement, Architektur-Gruppe
Nach AP organisierte Arbeitsgruppen
Telefonkonferenzen der WG nach Bedarf
TaskForces nach Bedarf
- Protokolle der Sessions/Treffen/Telcos:
Werden geführt und im Intranet dokumentiert.
Aktion-Item-Liste zum Protokoll
- Mailing-Listen für Projekt (Struktur: generell, Leitung, AGs): Mailing-Listen: öffentliche Liste,
StB, ArchGruppe: Jeder darf posten.
WP-Listen: Projektmitglieder dürfen posten.
- Website: <http://www.astrogrid-d.org>
(Herausgeber: Astrophysikalisches Institut Potsdam (AIP))
- Intranet:
(Apache Lenya CMS): <http://www.gac-grid.org/intranet.html>
(Jedes Mitglied kann lesen und hat schreibenden Zugriff.)
- Software-Repository:
zuerst CVS, dann SVN
gezielte Freigabe von Code-Zweigen (ro) auf Verlangen der Workgroup,
(rw) für Projekt

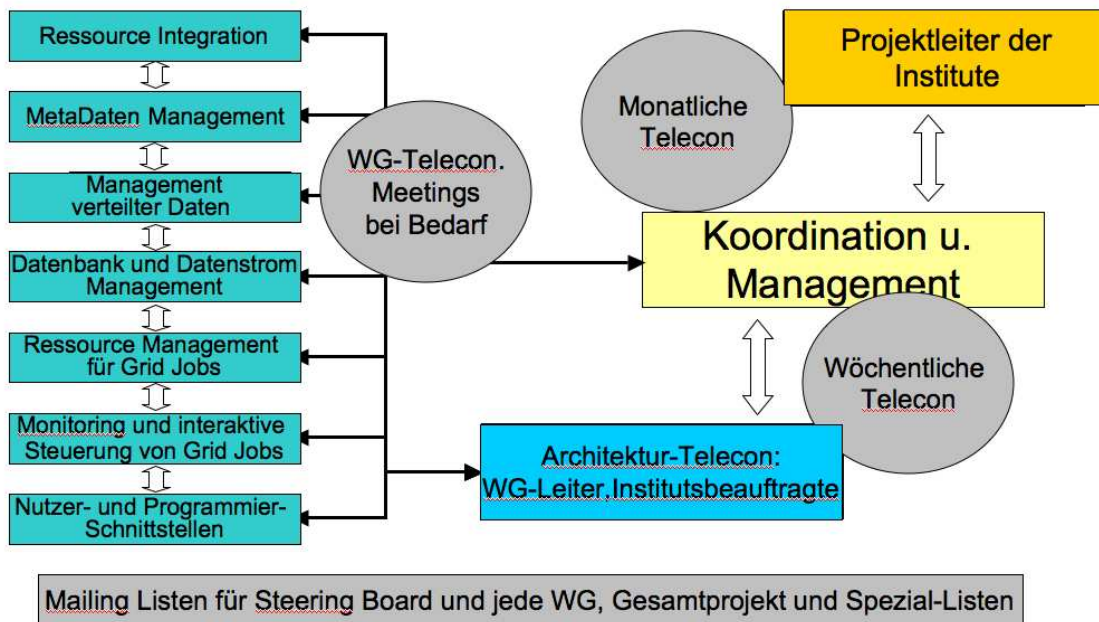


Abbildung 1: Organigramm AstroGrid-D

- Issue-Tracker:
JIRA mit OpenSource Lizenz
- Lizenzmodell für CG-Produkte:
Apache 2.0
- Anwendungen von AstroGrid-D:
Dynamo, Nbody6++, GEO600, Clusterfinder, Cactus, Robotic Telescopes, ProC, ...

3.2 HEP Grid

HEPGrid hat genau genommen keine ausgezeichnete Organisationsstruktur, da das HEPGrid sehr stark in das WLCG eingebettet ist. Die Terascale Alliance, die Teile des HEP Grids mit einbezieht, ist sehr viel besser abgegrenzt und strukturiert und eignet sich daher besser zu Dokumentationszwecken. Der Text verwendet also konsequenterweise primär die Terascale Alliance als Vorlage.

- Leitung, (wie oft Treffen/Telco etc):
20 Partner, Konsortialführer DESY.
Das Management Board kommuniziert variierend etwa einmal im Monat
Individuelle Arbeitsgruppen nach Bedarf
User Committee aus Vertretern der Experimente und den Service-Providern monatlich
- AG-Organisation:
Die AG hat besitzt als übergeordnetes Gremium das Institute Assembly in dem alle Partner

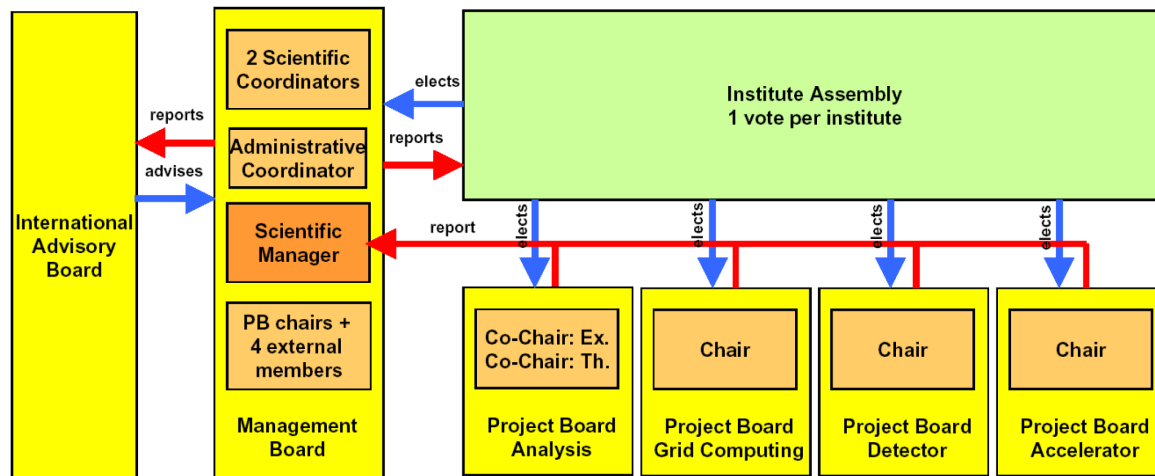


Abbildung 2: Organigramm der Terascale Alliance

vertreten sind.

Dem beigeordnet sind das Management Board und das International Advisory Board.

Nachgeordnet sind die individuellen Projekt-Gruppen.

- Protokolle der Sessions/Treffen/Telcos:
Werden geführt und in Intranet/Wiki sowie auf einem Meeting-Management-System (Indico) dokumentiert.
- Mailing-Listen für Projekt (Struktur: generell, Leitung, AGs):
Die Mailing-Listen sind analog zur AG-Organisation strukturiert.
Details über die Organisation der Listen finden sich unter <http://www.terascale.de/>
Die National Analysis Facility (NAF) innerhalb der Terascale-Alliance verfügt darüber hinaus über eine gegliederte Struktur von Mailing-Listen separiert in Helpdesk-, Intern-, Admin- und Support-Listen.
Die meisten Listen stehen nur registrierten Mitgliedern offen.
Auf die NAF-Helpdesk- und Support-Listen darf jeder posten.
- Website Terascale: <http://www.terascale.de>
Website NAF: <http://naf.desy.de>
Website HEP Grid: <http://documentation.hepcg.org/>
(Herausgeber: DESY)
- Intranet wird über die Authentisierung des CMS realisiert.
Jedes Mitglied kann lesen, nur berechnigte Mitglieder können schreiben.
- Software-Repository: werden fast ausschließlich von den Experimenten gepflegt
dCache APIs, GUI's und andere Pakete primär zur Interaktion mit der NAF werden im AFS sowie in yum-Repositories gehostet.
- Issue-Tracker:
Best Practical erweitert durch Eigenentwicklung wie GGUS-Support und krb-Authentifizierung.
- Kein einheitliches Lizenzmodell für CG-Produkte
dCache Entwicklungen zum Beispiel sind frei, aber nicht Open Source.

Andere Tools sind frei verfügbar ohne explizite Lizenzvereinbarung.

3.3 C3 Grid

- Gesamtleitung durch AWI, vierteljährliche Projekttreffen, monatliche Videokonferenz, zusätzliche thematische Videokonferenzen nach Bedarf
- Arbeitsgruppen mit fester Leitung, AG-Videokonferenzen nach Bedarf und zu spezifischen Arbeitsschwerpunkten, Arbeitstreffen zu den Arbeitspaketen nach Bedarf
- Protokolle der Treffen und Videokonferenzen werden auf dem Projekt-Wiki veröffentlicht.
- Mailing-Listen für Projekt: Gesamtliste und Listen zu den einzelnen Arbeitspaketen, alle nicht moderiert
- Informationen im WWW: Website mit MediaWiki, öffentlich zugänglich, Projektmitarbeiter schreibberechtigt
- Software-Repository: Projektmanagement-Software G-forge mit SVN-Repositories, Issue-Tracker u.s.w.
- Lizenzmodell für CG-Produkte: Open Source

3.4 TextGrid

OBS - Diese Aufstellung bezieht sich auf die erste Projektphase von TextGrid (Anfang 2006 bis Mai 2009). Mit der neuen Projektphase ist TextGrid um weitere Partner und Fachgebiete gewachsen. Generell und verkürzt kann man sagen, dass die erste Projektphase stark auf technische Infrastrukturarbeiten ausgerichtet war, während die zweite Phase stark auf Community Building und die Überführung in den Produktivbetrieb ausgerichtet ist.

- 8 Partner, Konsortialführer SUB Göttingen
Etwa Vierteljährliche Treffen, nach Bedarf (z.B. Vorbereitung Beta-Release, Vorbereitung Review, etc) auch öfter
Videokonferenzen, Telefonkonferenzen, Skype-Calls nach Bedarf, ggf. in AG-bezogenen Untergruppen monatlich und öfter
- AG-Organisation: Zwei AP-übergreifende Haupt-Arbeitsgruppen, mehrere Themen-bezogene Unterarbeitsgruppen (die teilweise nach Beendigung ihres Tasks auch wieder aufgelöst wurden). Jede AG hat einen AG-Leiter und Stellvertreter, die Vorbereitung und Moderation der AG-Treffen arrangieren, sowie Aktionen und Entscheidungen der AG verwalten.
- Protokolle der Treffen und Telefonkonferenzen werden geführt und in Wiki gesammelt. Aktionen (wer, was, bis wann) und Entscheidungen werden extrahiert und gesamtheitlich nachverfolgt.
- Mailing-Listen für Projekt: Mailing-Listen alle unmoderiert für intern gesamt, jede AG, öffentlich, Fachbeirat (hier nur TextGrid-Vertreter, ansonsten unabhängig)

- Website: Typo3-basierte Website verwaltet durch SUB, interner Bereich: Wiki (moderiert durch SUB) + verlinkter File-Vault und WebDAV
- Software-Repository: SVN-Repository — zuerst nur intern, seit TextGrid Version 1.0 auch extern lesbar und auf Anfrage schreibbar
- Issue-Tracker: Zuerst TRAC, mit Verfügbarkeit an SUB Umstieg auf JIRA
- Lizenzmodell für CG-Produkte: Open-Source; wird aktuell überarbeitet, da die vielfältig verwendeten Libraries (von Drittanbietern) jeweils unterschiedliche Einschränkungen auf die konkrete Lizenz geben (LGPL möglich?)

3.5 MediGrid

Die MediGRID-Community wurde in dem Projekt MediGRID aus dem ersten D-Grid-Call (2005 bis 2009) aufgebaut. Ziel war es, eine Infrastruktur für Grid Computing Technologien im biomedizinischen Bereich zu etablieren. Auch nach Beendigung des initialen MediGRID-Projekts existiert die MediGRID-Community weiter. In der zweiten Förderphase von D-Grid entstanden mit Services@MediGRID, MedInfoGrid und PneumoGrid weiterführende biomedizinische Grid-Projekte. Als gemeinsame Plattform aller biomedizinischen Grid-Projekte ist 2008 unter dem Dach der TMF das Forum Grid ins Leben gerufen worden. In Abbildung 3 ist der Aufbau der Organisation der MediGRID-Community skizziert.

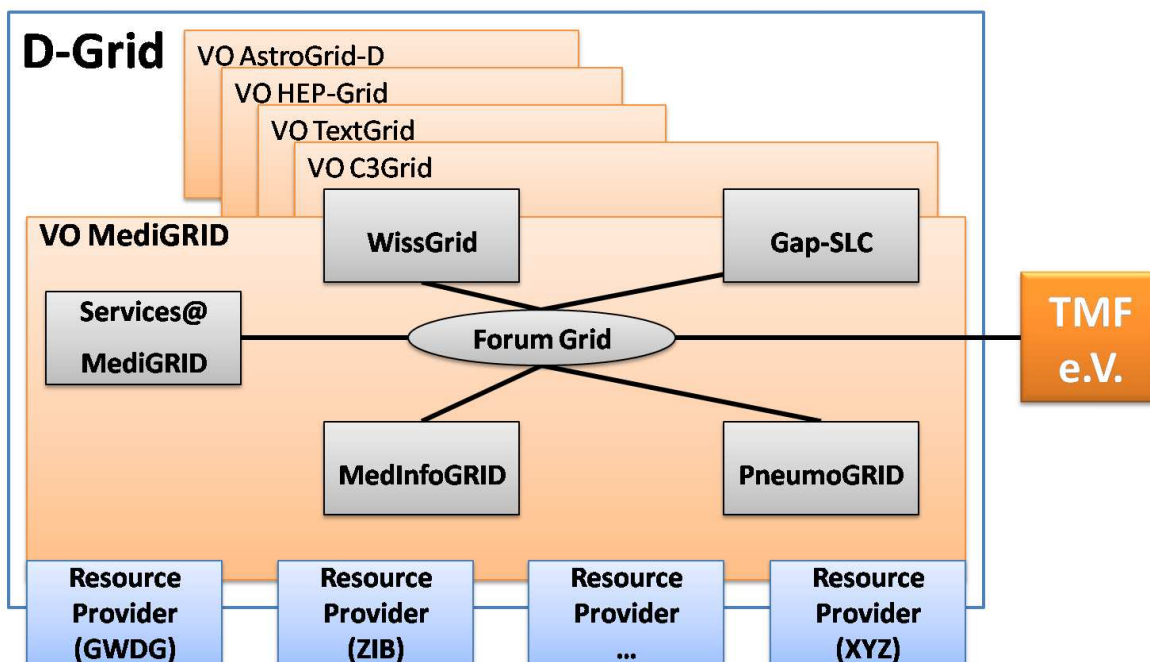


Abbildung 3: Aufbau der MediGRID-Community

Services@MediGRID ist ein Projekt, welches der MediGRID-Community angehört und die Infrastruktur von MediGRID verwendet. Die untenstehenden Informationen zu der Organisationsstruktur beziehen sich auf das Projekt Services@MediGRID.

- **Leitung (wie oft Treffen/Telco etc):**
Die Gesamtleitung erfolgt durch die UMG. Das Projekt Services@MediGRID umfasst insgesamt 9 Partner (7 akademische und 2 kommerzielle). Es finden halbjährliche Konsortialtreffen und monatliche Telefonkonferenzen (letzter Donnerstag im Monat) statt.
- **AG-Organisation (Leiter, Telco etc):**
Die Partner teilen sich auf 4 Arbeitspakete (Grid Coaching, Business enabling, Kundenprojekte, Faktura) auf. Regelmäßig finden AP-übergreifende Videokonferenzen unter den Anwendungsentwicklern statt.
- **Protokolle der Sessions/Treffen/Telcos:**
Elektronische Protokolle werden per E-Mail verteilt und auf dem Sharepoint-Server der UMG abgelegt. Auf den Sharepoint-Server haben alle Projektteilnehmern Zugriff.
- **Mailinglisten für Projekt (Struktur: generell, Leitung, AGs):**
Es wurde eine Gesamtliste für alle Projektteilnehmer und eine spezielle Liste für alle Anwendungsentwickler eingerichtet.
- **Website (Technik, Partizipation der Mitarbeiter: Intranet/Public):**
Die Pflege der Webseite erfolgt durch die UMG. Weiterhin existiert ein Wiki, in dem alle Projektteilnehmer lesenden und auf Wunsch schreibenden Zugriff haben. Zudem wird der Sharepoint-Server als Dateiablage und Kommunikationsplattform genutzt.
- **Software-Repository:**
Auf dem Sharepoint-Server erfolgt die Dateiablage der endgültigen Dokumente. Während der Erstellung sind die APs selbst für eine geeignete Ablage verantwortlich.

4 Skizze der Dienste und des Aufbaus des CG - Architekturkonzepte

Die Abschnitte dieses Kapitels bieten einen knappen Überblick zu den Grid-Strukturen, die jeweils aus CG-spezifischen Anforderungen entstanden sind. Auch hier sind verschiedene Einflüsse klar zu sehen, die charakteristisch für die Bedingungen eines CG sind:

- **Anliegen des CG - Anforderungen der Community**
Den Ausgangspunkt für Anforderungen bilden die durch die konstituierenden Partner eingebrachten Wünsche und Vorstellungen. Im Laufe des Projektes werden in der Regel die Anforderungen einer Revision und Validierung anhand von CG-Befragungen oder anderen Erfahrungen unterzogen. Einige CGs, wie z.B. das C3-Grid, haben schon vor der Konstituierung eine längere Phase gehabt, in der das Anforderungsprofil formuliert wurde mit recht klar definierten und eingegrenzten Zielen. Andere, wie z.B. das AstroGrid-D, haben zunächst recht weit gesteckte Ziele formuliert, um dann in einem Validierungs-Prozess eine Präzisierung zu erreichen.
- **Konstituierende Institutionen**
Die Institutionen welche an der Bildung des CG sich beteiligen, bringen auch bestimmte Software-Pakete, Dienste oder Applikationen ein, die entweder schon grid-befähigt sind oder portiert werden sollen. Dieses bildet in der Regel den anfänglichen Kern der CG-Arbeit.

- **Nationale Zusammenhänge**
Die Erfahrungen, die hier zu finden sind, entstanden ohne eine schon gefestigte D-Grid Struktur und etablierte Dienste. Ein Teil der CG-Arbeit hat den Aufbau des DGrid in seiner heutigen Form zum Inhalt gehabt. Hier finden neue CGs in eine völlig andere Situation vor. Wichtige konstituierende Dienste wie ein Middleware übergreifendes VO-Management, oder Ressourcen-Monitoring und Ansätze zu Accounting und Billing sind vorhanden, und weitere Dienste (wie z.B. SLA) sind durch GAP-Projekte in der Entwicklung. Das bedeutet, daß ein neues CG sich stärker auf die Bildung des CG konzentrieren kann.
- **Internationale Zusammenhänge und Kooperationen**
Hier sind zwei, teilweise gegensätzliche, Tendenzen festzustellen. Ist eine starke, international gut vernetzte Community vorhanden, wie sie durch den LHC und dessen Experimente (ATLAS etc) gegeben sind, so ist die Bildung eines nationalen CG nicht einfach zu realisieren. Sowohl die Middleware (gLite) als die gesamte Grid-Infrastruktur wird im wesentlichen vorgegeben. Die internationalen Kooperationen bilden in anderen Fachcommunities ein breites Spektrum von Rahmenbedingungen und Anforderungen. Sind z.B. in der Astronomie alle relevanten Projekte international (Instrumente), so ist gleichzeitig eine viel breitere Anforderungsstruktur festzustellen, sodaß auch die Middleware-Anforderungen keine solche Engführung wie im LHC gestatten. Die Klimaforschung (und andere Geophysikalische Forschung) hat eine sehr starke Standardisierung ihrer Daten vorangetrieben, mithin eine wesentlich andere Voraussetzung als in der Astronomie. Im Bereich Humanities sind die Arbeitsweisen, Datenhaltung, etc. der Fachdisziplinen sehr heterogen und fragmentiert. TextGrid und andere e-Humanities Projekte beteiligen sich daher an Standardisierungsinitiativen und arbeiten im Rahmen des Community-Building auf einen Mentalitätswandel hin zum vermehrten Austausch von Forschungsdaten und -tools. Kooperationen gestalten sich deshalb kleinteiliger.
- **Vorhandene Kenntnisse / Entwicklungen / Dienste**
Wie oben schon festgestellt, sind die als Ausgangspunkt vorhandenen Kenntnisse und Applikationen oft von grossem Einfluss auf die Architektur (oder innere, technische Struktur) des CG. Deutlich wird, dass z.B. beim MediGrid und dessen Folgeprojekten, die wesentliche Komponenten der derzeit genutzten Infrastruktur über einen der Partner bereits vorhanden waren. Das ist einerseits sehr effizient, bildet jedoch gleichzeitig bei der Integration in eine fachübergreifende Infrastruktur (DGrid) einige Hürden.

Deutlich ist, das einige Komponenten essentiell für ein CG sind. Hierzu zählen

- die Herausbildung einer Security-Struktur (Zertifikatsmanagement, RAs),
- die Bildung einer Virtuellen Organisation,
- die Auswahl einer geeigneten Middleware mit
- Integration von einem Kern von Community-Ressourcen und/oder
- Implementierung von essentiellen CG-Diensten (z.B. TextLab, C3 Portal).

Wie aus der Übersicht hervorgeht, ist in vielen CGs die Festlegung auf geeignete MetaDaten-Standards ein Problem, welches zu wenig Beachtung erfährt. Wo solche Standards vorhanden sind (C3), ist es für die CG-Bildung ein immenser Vorteil. Ersichtlich ist auch, daß CGs sich für eine

möglichst flexible Middleware entscheiden, wo nicht durch Vorgaben aus der (internationalen) Community diese Frage gar nicht wirklich gestellt werden kann. Die eher Daten-zentrierten Community-Anforderungen zeigen, auch durch die Vielzahl der Eigenentwicklungen, daß die Middleware-Unterstützung hier noch recht schwach ausfällt.

4.1 AstroGrid-D

- Funktionaler Aufbau der Grid-Nutzung: Job-Submission über Skripte, JSDL-Translator, Grid-Way, GAT-Client, Portal Datenmanagement: gsiscp, gsiftp, AstroGrid-D Data Management (ADM)
- Zertifikats-Management / RAs / VO-Struktur / VO-Management: RootCA: GridKA, DFN weitere CA: Ukraine, Holland, Japan RAs sind in den meisten Instituten etabliert VOMRS Server in Eigenbetrieb, SUB-VOs für Institute und Arbeitsgruppen, Skripte zum VO-Management auf Ressourcen-Ebene
- genutzte Middleware: GT4, Metadaten-Komponenten des Virtual Observatory, Eigenentwicklungen
- Nutzung von D-Grid Ressourcen:
Nutzung eigener Ressourcen und Ressourcen im Rahmen der Sonderinvestitionen.
Nutzung von D-Grid Ressourcen in erheblichem Umfang
- Middleware-Extentions:
filebasierte Daten/Metadaten-Verwaltung in Globus
Stellaris, Job-Monitoring mit Globus; TimeLine, GridResourceMap
Stream-Based Data Management
GAT (Grid Application Toolkit)
Robotic Telescopes Package
- Genutzte und konzipierte CG-Dienste:
(Informationsdienste, Job-Services, Daten-Services usw.): GridSphere-Portal
- Struktur/Standard der Metadaten für Daten/Jobs/Dienste:
Globus-XML, GLUE-Schema, Job Description Document (JDD)
- Software-Lifecycle Management / Testumgebungen :
Bugtracking (JIRA)
SVN-Repository
- Andere Support-Strukturen (Help-Desk usw): Mailing-Listen und Admin-Liste AIP E-Science Team
- Lizenzen-Management: bislang keine kommerziellen Lizenzen.

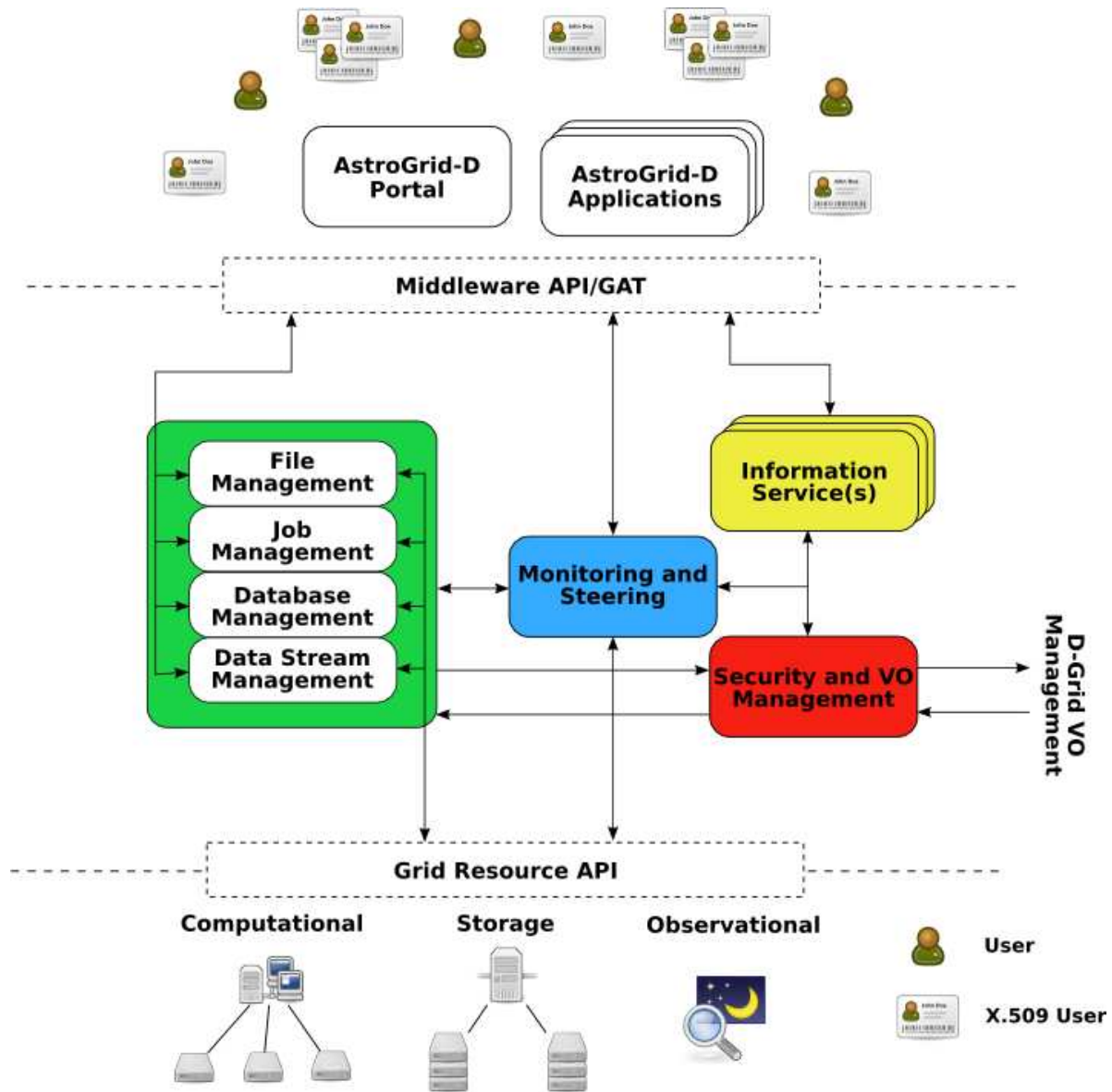


Abbildung 4: Architektur-Schema AstroGrid-D

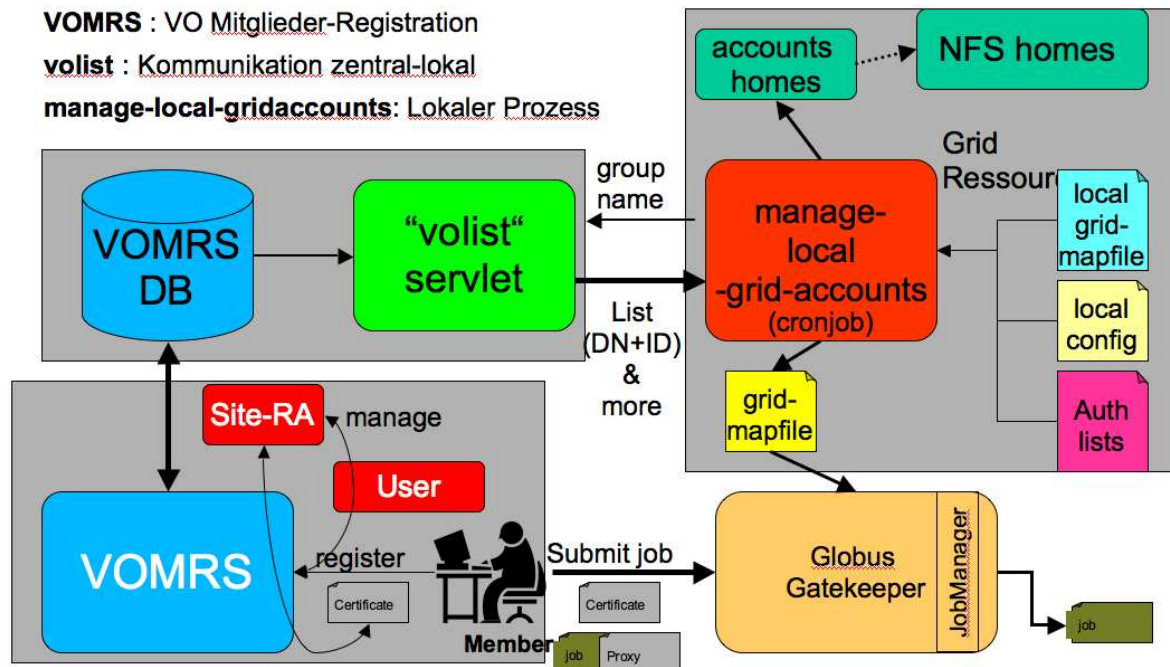


Abbildung 5: VO Management AstroGrid-D

4.2 HEP Grid

- Funktionaler Aufbau der Grid-Nutzung: Compute- und Daten-Grid.
Zugang über diverse Linux-basierte Interfaces, wie z.B. Ganga, dq2, phedex, fts.
- Certifikats-Management / RAs / VO-Struktur / VO-Management:
DESY betreibt RA für die CA am FZK (GridKa)
Betrieb von VOMRS/VOMS für eine Reihe von VOs aus HEP, Astroparticle und Theoretische Physik sowie Photon Science
Support für Photon Science spezifische VOs wie BioMed, XRAY.EU und XFEL.
- genutzte Middleware:
gLite
- Nutzung von DGrid-Ressourcen:
Betrieb einer generischen Grid-Infrastruktur, die die Ressourcen der D-Grid Sonderinvestitionen, aus DESY's WLCG Tier2-Aktivitäten und DESY Experimente enthält.
- Middleware-Extentions:
Entwicklung des Datamanagement Frameworks dCache in Kooperation mit Fermilab.
- Genutzte und konzipierte CG-Dienste
(Informationsdienste, Job-Services, Daten-Services usw) :
Informationsdienste: gLite-BDII, LCG-FileCatalog (LFC), Amga mit Oracle und Postgresql als Backend
Job-Services: glite-WMS, PBS/Maui

Daten-Services: dCache

Daten-Zugang: gsiFTP, dcap, NFS4, lcg, auch via gEclipse als Frontend.

- Struktur/Standard der Metadaten für Daten/Jobs/Dienste:
primär LFC und Amga
- Software-Lifecycle Management / Testumgebungen :
Lifecycle-Management in erster Linie mit EGEE-preProduction System (PPS), Quattor, CVS;
partiell auch SVN oder GIT.
- Andere Support-Strukturen (Help-Desk usw.):
Verteiltes ROC für DE/CH. GGUS, Mailing-Listen, Best Practical Request Tracker sowie regelmäßige Treffen mit User-Communities aus den Experimenten, regelmäßige Workshops/Tutorials
- Lizenzen-Management:
Oracle ist eventuell erwähnenswert, wird unter anderem für Atlas tagDB eingesetzt. Ist noch im Aufbau.
Photon Science Daten-Analyse beruht im Gegensatz zum HEP-Modell sehr viel stärker auf proprietärer oder closed source Software, und/oder basiert in der Regel auf personalisierten Lizenzvereinbarungen.

4.3 C3 Grid

- Datenzentrierte Grid-Nutzung
- Kein zentrales Zertifikats-Management, RAs bei den Einzelinstitutionen (AWI, ZIB, DKRZ, MPIM, IFM-Geomar, DLR, ...)
- Einzelne gemanagte VO innerhalb von D-Grid, vor allem aber Nutzung eigener Grid-Ressourcen ohne explizite VO-Struktur
- Genutzte Middleware:
Globus Toolkit GT4, dCache, Metadaten-Middleware-Komponenten der Open Archives Initiative (OAI), Eigenentwicklungen zu Metadaten/Daten
- Hauptsächlich Nutzung eigener Grid-Ressourcen, daneben D-Grid
- Middleware-Extentions:
MDS-Erweiterung zur Verwaltung der Metadaten- und Datenressourcen, Eigenentwicklung Scheduler, Eigenentwicklung Datenmanager, Eigenentwicklung Metadatensuchdienst
- Genutzte und konzipierte CG-Dienste:
Metadaten-Informationssdienst, Job-Scheduling-Dienst, Daten-Staging-Dienst, durch Portal nutzbare Workflows
- Metadaten-Standard für Daten: ISO19115/19139
- Metadaten-Standard für Jobs: JSDL-Erweiterung
- Kein zentrales Software-Lifecycle Management, Software-Lifecycle Management der Teilkomponenten durch verantwortliche Einzelinstitutionen, Betrieb einer Testumgebung

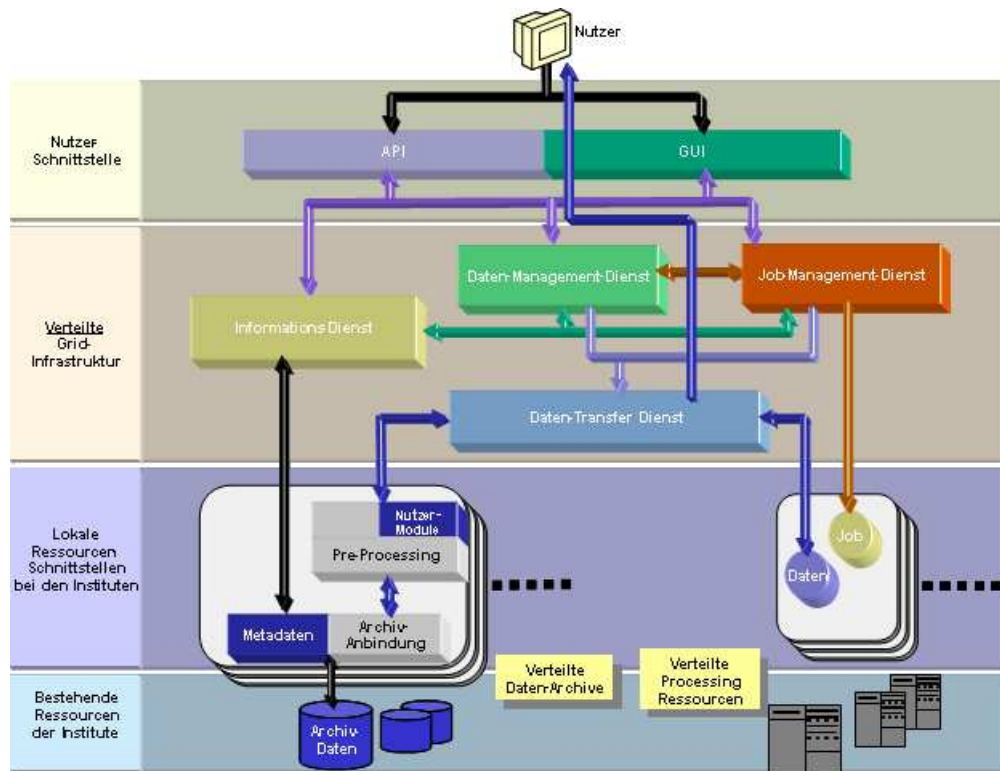


Abbildung 6: Architektur-Schema C3Grid

- Kein Lizenzen-Management

4.4 TextGrid

- Funktionaler Aufbau der Grid-Nutzung:
Daten-Grid, Zugang zunächst über Eclipse-basiertes Portal, Migration zu einer browser-basierten Lösung wird evaluiert.
- Zertifikats-Management / RAs / VO-Struktur / VO-Management:
Keine separaten Zertifikat-Strukturen, DFN RA-Netzwerk wird genutzt.
Derzeit nur eine große VO (tx - TextGrid). Wesentliche zukünftige Anforderung ist die Nutzung von Grid/Shibboleth, was über Projekte wie PERMIS, SLC, ASPiS, etc. gerade entwickelt wird. Daher sind derzeitige Strukturen bewusst temporär.
- Genutzte Middleware:
GT4, Metadaten-Middleware-Komponenten der Open Archives Community, Eigenentwicklungen zu Metadaten/Daten
- Nutzung von DGrid-Ressourcen:
Hauptsächlich Nutzung eigener Ressourcen und Ressourcen im Rahmen der Sonderinvestitio-

nen. Zukünftig Datenreplikation in D-Grid, sollte es in Zukunft SLAs für nachhaltige Speicherung (Langzeitarchivierung) geben.

- **Middleware-Extentions:**
Filebasierte Daten/Metadaten-Verwaltung in Globus, Metadaten-Suche, XML-basierte Strukturdatenverwaltung und -suche, RDF-basierte Verwaltung von Objekt-Relationen, Shibboleth/RBAC-basiertes Rechtemanagement
- **Genutzte und konzipierte CG-Dienste**
(Informationsdienste, Job-Services, Daten-Services usw.): Middleware-spezifisch — siehe voriger Punkt
Eclipse-basierter Client, der Zugriff auf Datagrid in Eclipse abstrahiert (d.h. beliebige Eclipse-basierte Tools können so ins TextGrid geschoben werden). Zusätzliche Tools in TextGrid: XML-Editor, grafischer Editor, Wörterbücher, Workflows: Streaming, Tokenizer, Lemmatizer, etc.
- **Struktur/Standard der Metadaten für Daten/Jobs/Dienste:**
Metadaten-Schema basierend auf Dublin Core.
XML-basierte Datenformate basierend auf TEI (Text Encoding Initiative).
Objekt-Struktur inspiriert durch METS (Metadata Encoding and Transmission Standard).
- **Software-Lifecycle Management / Testumgebungen:**
Zwei separate Testumgebungen (jeweils für Middleware und Client). Prozess zur Aufnahme von neuen Modulen in den Release. Dieser Prozess soll in Zukunft in einen offenen Community-Prozess ausgebaut werden.
- **Andere Support-Strukturen (Help-Desk usw.):**
Mailingliste, regelmäßige Workshops/Tutorials, etc. werden in der kommenden Projektphase installiert.
- **Lizenzen-Management** wird gerade überarbeitet (siehe letztes Kapitel).

4.5 MediGrid

- **Funktionaler Aufbau der Grid-Nutzung:**
Es erfolgt im Wesentlichen eine compute-zentrierte Grid-Nutzung über verschiedene Interfaces. Der Regelfall ist die portalbasierte Nutzung von Anwendungen/Services, die jeweils von Anwendungsentwicklern für eine größere Nutzerzahl zur Verfügung gestellt werden. Was eher nicht auftritt ist der Fall, dass Endanwender ihre eigenen Anwendungen mitbringen, selbst installieren und warten und täglich weiterentwickeln.

Im portalbasierten Fall gibt es einerseits die ausführbaren Anwendungen/Services, die auf diversen Ressourcen im Grid installiert werden. Auf der anderen Seite befinden sich die Nutzer, die über anwendungsspezifische Portlets auf die Anwendungen/Services zugreifen, Input-Daten hochladen und Parameter setzen können, Ergebnisdaten anschauen und/oder herunterladen und somit die Anwendungen steuern können. Zwischen diesen beiden sitzt u.a. der Grid Workflow Execution Service, der die Rolle eines Resource Brokers, Meta-Schedulers und einer Workflow Engine übernimmt. Bei den Portlets sind abstrakte Workflowbeschreibungen

hinterlegt, die der GWES interpretiert und auf Basis derer er die geeigneten Ressourcen ermitteln kann, um dann konkrete Workflowinstanzen mit dedizierten Ressourcen zu initialisieren und den Ablauf der Workflows auf den finalen Hardware-Ressourcen zu koordinieren.

Es gibt darüber hinaus im MediGRID auch einzelne Anwendungen, die keine portalbasierten Interfaces haben und wo statt dessen Gridjobs auf der Kommandozeile abgesetzt werden - sowohl über den GWES als auch direkt auf die Ressourcen.

- **Certifikats-Management / RAs / VO-Struktur / VO-Management:**
Es ist kein eigenes zentrales Zertifikats-Management etabliert. Stattdessen existieren RAs zur DFN-Grid-CA z.T. bei den Einzelinstitutionen. Die MediGRID Community umfasst mehrere D-Grid Projekte: MediGRID, Services@MediGrid und MedInfoGrid.
- **genutzte Middleware:**
In MediGRID wird ausschließlich Globus Toolkit GT4 verwendet.
- **Nutzung von DGrid-Ressourcen:**
Hauptsächlich erfolgt eine Nutzung eigener Grid-Ressourcen. Daneben werden auch weitere Ressourcen aus D-Grid verwendet.
- **Middleware-Extentions:**
Als Erweiterungen werden MDS und Ganglia zum Monitoring der Ressourcen und SRB/iRods für das Datenmanagement verwendet.
- **Genutzte und konzipierte CG-Dienste:**
(Informationsdienste, Job-Services, Daten-Services usw): Im MediGRID wird der Grid Workflow Execution Service (GWES) verwendet. Er dient als Resource Broker, Metascheduler und Workflow Engine zum Abbilden komplexer Grid Jobs auf die im MediGRID vorhandenen Ressourcen. Anhand von Metadaten zu Hardware-Ressourcen und Anwendungen, die im D-GRDL Format in einem Repository hinterlegt sind, kann so eine Virtualisierung der Ressourcen und Anwendungsinstanzen erfolgen.

Über ein Portalsystem (derzeit noch GridSphere, demnächst Liferay) kann auf die verschiedenen MediGRID-Anwendungen zugegriffen werden. Das Portal integriert Portlets zu den meisten Anwendungen. Diese Portlets bilden für den Endnutzer die Nutzerschnittstelle zur Parametereingabe und zum Up-/Download von Daten. Diese Portlets initiieren Grid-Workflows, die an den GWES submittiert werden. Dieser wählt über seine Broker- und Scheduling-Funktion die konkreten Instanzen auf Basis der Auslastung und anderer Parameter aus und orchestriert dann die verteilte Job-Ausführung.

Das DGAS Accounting System wird im Prinzip genutzt. In der Praxis hängt es noch an der Verabschiedung von Nutzungsvereinbarungen mit dem Betreiber RRZN. Auf Basis der DGAS Daten wird in der MediGRID Infrastruktur (Projekt: Services@MediGRID) ein sogenanntes fachliches Accounting und Billing System namens „CollAct“ der Firma c.a.r.u.s. IT AG genutzt, das es Dienstleistern ermöglicht im Auftrag mehrerer Mandanten im Grid zu agieren und die Accounting-Daten nach Mandanten zu unterscheiden, um somit auf dieser Grundlage auch eine Rechnungstellung realisieren zu können.

- **Struktur/Standard der Metadaten für Daten/Jobs/Dienste:**
Für die Software-Beschreibung wird die im DGI-Projekt von D-Grid entwickelte D-GRDL verwendet.

- Software-Lifecycle Management / Testumgebungen:
Es gibt kein zentrales Software-Lifecycle Management. Für das Software-Lifecycle Management der Teilkomponenten sind die entsprechenden Einzelinstitutionen zuständig. Es wird eine Testumgebung des Portalsystems betrieben. Das Verfahren ist wie folgt:

Die Anwendungsentwickler (bzw. Service Provider) treten als SW-Admin in Erscheinung - im Sinne der Rollenvergabe im VOM(R)S. Wenn eine neue Anwendung ins Grid gebracht werden soll, gibt es dafür einen klaren Ablauf, der in einem Dokument "Guidelines for the Deployment of Applications in MediGRID" [11] ausführlich beschrieben ist. Der Prozess umschließt, kurz zusammengefasst, die Installation der Anwendungen/Services auf den Ressourcen, die Funktionstests, den Eintrag in die Informationssysteme - dort vor allem in eine Hardware- und Serviceregistry, deren Einträge im D-GRDL-Format gepflegt werden - und die Erstellung von Portlets, deren Deployment im MediGRID Entwicklerportal und letztendlich das Portletdeployment im Produktivportal. Die Außerbetriebnahme von Anwendungen und Diensten ist derzeit nicht klar geregelt. Ebenso sind Updates von bestehenden Anwendungen/Services nicht klar geregelt. Ein zeitweiliger Parallelbetrieb von alten und neuen Versionen und entsprechende Vorwarnungen der Nutzer wären hier sinnvoll.
- Andere Support-Strukturen (Help-Desk usw):
Das Trouble-Ticket-System DGUS aus D-Grid wird genutzt. Für sämtliche Anwendungs- und Infrastrukturdienste sind Supporter aus der Community hinterlegt. Das System wurde auch in dieser Hinsicht erweitert, um bei der Problembeschreibung bereits die Anwendungen oder Dienste gezielt im Helpdesk auswählen zu können, mit denen ggf. ein Problem besteht.

Darüber hinaus gibt es eine Mailingliste für die Softwareentwickler in MediGRID. Dort können Fragen gestellt und offene Punkte diskutiert werden. Es gibt zusätzlich regelmäßige Videokonferenzen unter den Anwendungsentwicklern.
- Lizenzen-Management:
derzeit kein Lizenzen-Management

5 Software-Lifecycle-Management

Dieser Abschnitt sammelt die wichtigsten Aspekte und Ansätze des Software-Lifecycle-Managements in den wissenschaftlichen Community-Grids der ersten D-Grid Förderphase. Damit soll eine Grundlage für die Entwicklung von Empfehlungen gelegt werden, die schließlich in Form von Blaupausen neuen Communities zur Organisation des Software-Managements an die Hand gegeben werden können. Dabei sollen nicht Konzepte der allgemeinen Softwaretechnologie aufgegriffen und lediglich im Grid-Kontext beschrieben werden, sondern insbesondere Vorgehensmodelle, Organisationsformen sowie Erfahrungen aus den bereits existierenden Community-Grids aufgenommen werden. Nur aus einer solchen Erhebung lässt sich ein praxisrelevantes Konzept entwickeln, das einerseits die speziellen Gegebenheiten der Softwareentwicklung in akademischen Projekten und gegebenen Konsortialstrukturen berücksichtigt und andererseits auf hinreichende Akzeptanz stößt und so Umsetzung in nachfolgend geförderten Projekten findet.

Dazu wurden für die Sammlung der vorliegenden Inhalte mit der technischen Entwicklung beauftragte Vertreter der Communities sowie das jeweilige Projektmanagement gebeten, ihre Ansätze entsprechend der vorgegebenen thematischen Struktur zu beschreiben. Grundsätzlich sollte jedes

Community-Grid seine Ansätze in einem eigenen Abschnitt zusammenfassen. Innerhalb eines solchen Abschnittes konnten alle Partner ihr gemeinsames oder individuelles Vorgehen entsprechend der vorgeschlagenen Fragestellungen erläutern. Zusätzlich sollten die entwickelten Anwendungen und spezielle Software-Management-Entscheidungen knapp herausgestellt werden.

Die vorliegende Zusammenstellung von Aspekten des Software-Lifecycle-Managements der unterschiedlichen Projekte ist bewusst wenig editiert, um eine unverfälschte Einschätzungen der verantwortlichen Entwickler und Projektbeteiligten zu erhalten. Teilweise wurden jedoch Inhalte von persönlichen Gesprächen und aus Korrespondenzen zusammengefasst und in diese Sammlung eingepflegt. Insgesamt wurde auf eine sehr kritische Betrachtung der Vorgehensweisen Wert gelegt, um tatsächliches Verbesserungspotenzial zu identifizieren. Obwohl alle Community-Grids in ihrer ersten Phase erfolgreiche Entwicklung betrieben und die Projektziele erreicht haben, sollten im Kontext der Vorgehens- und Meinungsammlung auch problematische Aspekte des Softwaremanagements angesprochen werden, um aus gemachten Erfahrungen und auch Fehlern zu lernen. Eine eventuell überproportionale Darstellung von Problemen und Kritikpunkten am gewählten Vorgehen der einzelnen Projekte lässt explizit nicht auf den Gesamterfolg des Projektes schließen.

5.1 Aspekte des Software-Lifecycle-Managements

Dieser Abschnitt stellt eine Struktur für die Beschreibung des Software-Management-Prozesses innerhalb der einzelnen Community-Projekte zur Verfügung. Unter jedem Abschnitt befindet sich eine Liste von Aspekten, die im Text kurz adressiert werden sollten.

5.1.1 Ermittlung und Umgang mit Anforderungen

- Vorgehen bei der Ermittlung der Anforderungen (Workshop mit Nutzern, Anforderungen von Nutzern im Projekt, Analyseprozess, ...)
- Welches Gewicht hatten bei der Analyse
 - funktionale Anforderungen (Anforderungen an das erwartete Verhalten von Software)?
 - nicht-funktionale Anforderungen (Anforderungen, die Qualitätsaspekte in den Vordergrund stellen, etwa Performance, Wartbarkeit, Zuverlässigkeit, Sicherheit, ...)?
- Produkte des Anforderungsmanagements
 - Dokumentation der Anforderungen und von Änderungen während des Entwicklungsprozesses
 - Use Cases, Diagramme
 - Interfaces, Prototypen
- Werkzeugunterstützung (wurden Softwarewerkzeuge zur Unterstützung des Analyseprozesses verwendet?)
- Fazit, Meinung zum gewählten Weg, Vorschläge und Verbesserungsmöglichkeiten

5.1.2 Organisationsstrukturen und Vorgehensmodelle

- Organisation des Entwicklungsprozesses
 - Arbeitsorganisation und Arbeitsverteilung (zentrales Projekt- und Software-Management oder verteilt auf einzelne Partner und Institutionen)
 - Rollen im Projekt (Projektmanagement, technische Leitung, Kunden/Nutzer, Entwickler, Tester, ...)
 - Organisationsprozesse, Abstimmungs- und Eskalationswege für die Klärung von Problemfällen und zur Entscheidungsfindung
- Eingesetzter Entwicklungsprozess (Insbesondere soll die Fragestellung betrachtet werden, ob und wie etablierte Vorgehensmodelle der Softwaretechnologie angewendet wurden. Hier ist von großem Interesse, ob dies ein gemeinsamer Prozess, gemeinsame Unterprozesse für alle beteiligten Partner oder für jeden Partner ein individueller Prozess waren.)
- Werkzeugunterstützung (wurden Softwarewerkzeuge zur Unterstützung des Projektmanagements verwendet?)
- Fazit, Meinung zum gewählten Weg, Vorschläge und Verbesserungsmöglichkeiten

5.1.3 Umgang mit Risiken

- Analysen zur Entdeckung von Risiken im Entwicklungsprozess (Welche möglichen Problem wurden bereits vor dem Entwicklungsbeginn festgestellt?)
- Aspekte der Risikoanalyse (Welche Aspekte wurden in die Risikoanalyse einbezogen und wie wurde diesen genau begegnet?)
 - Technologiewandel (Änderung von eingesetzten Technologien über die Projektlaufzeit)
 - Finanzielle Planung (Aufwands- und Arbeitsabschätzung unter Berücksichtigung des festen verfügbaren Förderbudgets)
 - Zeitplanung (Planung von Projektziel und Zwischenzielen sowie des Projektfortschrittes unter Berücksichtigung der begrenzten Laufzeit)
- Strategien zur Abschätzung von Risiken (Wie wurden Risiken im Entwicklungsprozess bewertet?) und zur Vermeidung und Akzeptanz von Risiken
- Fazit, Meinung zum gewählten Weg, Vorschläge und Verbesserungsmöglichkeiten

5.1.4 Testumgebungen, Testverfahren und Qualitätsmanagement

- Test- und Qualitätsziele des Projektes (Wie wurden Testziele für die Softwareentwicklungen im Projekt definiert und festgehalten?; Testfälle?)
- Testverfahren für einzelne Softwareentwicklungen (Wie wurde die Software eines Partners getestet und welche Testverfahren wurden dazu genutzt? Hier ist interessant, ob und welche

manuellen oder automatischen Tests genutzt wurden². Weiterhin ist eine Darstellung des Testprozesses wünschenswert über die Definition einer Teststrategie, einer Testumgebung und der Festlegung von Anforderungen bis zur Durchführung und Analyse der Tests.)

- Testverfahren und Testumgebungen für den integrierten Betrieb der entwickelten Infrastruktur (Gab es eine Testumgebung für die Evaluation von Neuentwicklungen im Gegensatz zum Produktivsystem? Wie oben ist auch hier ist das Testvorgehen von zentralem Interesse³.)
- Test- und Zertifizierungsprozess innerhalb der Community (Gab es einen Prozess zur Verifikation der Funktionalität von Produkten innerhalb der Community?)
- Werkzeugunterstützung (wurden Softwarewerkzeuge zur Umsetzung der Tests angewendet?)
- Fazit, Meinung zum gewählten Weg, Vorschläge und Verbesserungsmöglichkeiten

5.1.5 Lebenszyklus nach der Fertigstellung und Nachhaltigkeit

- Deploymentzyklen für die entwickelte Software (Packagingprozesse und Verteilung der Software für den produktiven Einsatz. Dazu gehört ebenfalls eine Versionierung der Softwarepakete.)
- Betrieb, Wartung und Produktionsbetrieb der Software (Hier sollen die organisatorischen und technischen Maßnahmen und Prozesse zur Sicherstellung des Betriebes dargestellt werden, insbesondere gehören dazu die Wartungszyklen der Hard- und Software-Infrastruktur.)
- End-of-Life (Vorgehen zur Außerbetriebnahme von Software und der damit verbundene Prozess. Hier können etwa Schritte wie Datenrettung, Migration von Nutzern, Upgradepfade und Abschaltphasen als Bestandteile des Prozesses auftauchen.)
- Fazit, Meinung zum gewählten Weg, Vorschläge und Verbesserungsmöglichkeiten

5.2 Life-Cycle-Management im AstroGrid-D

5.2.1 Allgemeine Angaben zum AstroGrid-D-Projekt

Zum Zeitpunkt des Redaktionsschlusses für dieses Dokument lagen noch keine abschließenden Materialien vor. Bisher haben einige Aspekte bezüglich Organisationsstruktur und Softwareentwicklung bereits Einzug in andere Kapitel dieses Dokumentes gefunden. Die Datensammlung bei den Partnern aus dem AstroGrid-D-Projekt ist jedoch im Kürze abgeschlossen und wird anschließend editiert und in das vorliegende Dokument eingepflegt.

Ermittlung und Umgang mit Anforderungen In der Anfangsphase des AstroGrid-D (AGD) wurden UseCases durch gezieltes Ansprechen von Wissenschaftlern der beteiligten Institute erstellt. Die UseCases wurden entlang eines vom Projekt entwickelten Fragenkatalogs aufbereitet. Alle UseCases sind auf der Website des Projekts dokumentiert.[8]

²Black-Box-Test, White-Box-Test, funktionale und nicht-funktionale Tests, Schnittstellentests, Interoperabilitätstests, Belastungstest, Performancetest, Sicherheitstest

³Hier spielen Komponententests, Integrationstests und Systemtests eine Rolle.

Die UseCases sind in den verschiedenen Workgroups des AGD jeweils in Hinblick auf ihre besonderen Fragestellungen analysiert und in Anforderungen an die Architektur bzw. die Realisierung des Workpackages eingeflossen. Insbesondere gab es eine Arbeitsgruppe Architektur, welche die Arbeiten in einem integrativen Prozess begleitet hat. Dies hat einen starken und sehr notwendigen Gegenpol zur Tendenz der Software-Ersteller zu Verselbständigung ihres Produkts gebildet.

Organisationsstrukturen und Vorgehensmodell tbc.

5.3 Life-Cycle-Management im C3Grid

5.3.1 Allgemeine Angaben zum C3Grid Projekt

Ermittlung und Umgang mit Anforderungen: Ermittlung der Anforderungen erfolgte durch Beteiligung von Nutzern und nutzernahen Institutionen (DKRZ, AWI, FU Berlin, ...). In diesem Umfeld wurden Workshops durchgeführt und Testnutzer ausgewählt. Erste Ergebnisse wurden auf Community-Workshops und Konferenzen präsentiert und zur Diskussion gestellt. Das AP "Outreach" im C3-Grid hat die Ergebnisse analysiert und Anforderungen an die SW-Umgebung formuliert. Diese Anforderungskatalog wurde auf den regelmäßigen Planartreffen des Projekts mit den anderen APs diskutiert und eine Klassifizierung in folgende Kategorien vorgenommen:

- Kurzfristig umsetzbar (d.h. innerhalb der Projektlaufzeit),
- Mittelfristig zu realisieren (d.h. Aufgabe für ein Nachfolgeprojekt)
- Langfristige Umsetzung (d.h. Aufgabe für nachgelagerte Entwicklungen)

Basierend auf diesen Anforderungsanalysen wurden Use Cases entwickelt und der Funktionsumfang der Prototypen definiert.

Prozess der Anforderungsanalyse: Der Prozess der Anforderungsanalyse bestand aus folgenden Schritten: Während der Antragsphase wurde eine Analyse der Anforderungen in der Klima-Community durchgeführt, die bereits dort in der Planung verschiedener Arbeitsschritte und einer Grobarchitektur mündete. Im Laufe einer ersten Evaluationsphase wurden Anwendungsfälle (genauer beispielhafte Workflows der Klimawissenschaften) ausgewählt, anhand derer die prototypische Infrastruktur und zugehörige Dienste definiert wurden.

Gewichtung funktionaler und nicht-funktionaler Anforderungen: Zu Beginn des Entwicklungsprozesses wurde insbesondere auf die funktionalen Anforderungen an das System Wert gelegt, zu einem späteren Zeitpunkt standen nicht-funktionale Anforderungen mindestens gleichberechtigt im Fokus der Entwicklung. Die funktionalen Anforderungen ergaben sich vor Allem aus der initialen Auswahl von klimawissenschaftlichen Workflows und den mit der Abarbeitung verbundenen identifizierten Diensten und Aufgaben. Dabei stand zuerst die Evaluation der technischen Entscheidungen im Vordergrund. Im Verlauf des Projektes kamen verschiedene nicht-funktionale Anforderungen hinzu:

- einfache Handhabbarkeit (insbesondere bezogen auf die Nutzbarkeit der Portalanwendung)

- Zuverlässigkeit der höherwertigen Dienste wie Scheduling und Datenmanagement hin zu einer im Produktivbetrieb laufenden Infrastruktur.
- Sicherheit: diese Anforderung wurde insbesondere auf die Anforderungen von Nutzergruppen hin verstärkt betrachtet.

Produkte des Anforderungsmanagements:

- Dokumentation der Anforderungen durch Nutzerbeschreibungen und durch Ablaufdiagramme und Interaktionsdarstellungen von Diensten im Projekt-Wiki
- Definition von Schnittstellen
- Umsetzung verschiedener Prototypen zur Demonstration der funktionalen (prototypische Dienste) und nicht-funktionalen Anforderungen (Demonstratoren beim Portal)

Werkzeugunterstützung:

- Wiki zur Dokumentation der Anforderungen und Produkte des Anforderungsmanagements

Fazit, Meinungen und Verbesserungsmöglichkeiten:

- Der teilweise sehr dezentral gestaltete Prozess der Anforderungsermittlung führte zu unterschiedlichen Ansätzen und Problemen in der Entwicklung während der Konzeptionsphase.
- Kommunikationshürde zwischen Nutzern und technischen Partnern war nicht immer leicht zu überbrücken. Hier wäre sicher eine Vermittlung notwendig, genauso wie ein gemeinsames Vorgehen bei der Dokumentation der gefassten Anforderungen.
- Veranstaltungen zur Integration von weiteren Nutzeranforderungen sind sehr wichtig und sollten ggf. häufiger durchgeführt werden, um die Anforderungen der Nutzer außerhalb des Projektes besser aufzunehmen.
- Der Einsatz eines Wikis als Kollaborationsplattform ist in einem so großen Projektverbund nur bedingt sinnvoll möglich, da hier keine klaren Definitionen und Prozesse zur Dokumentation vorgegeben sind. Hier müsste ein klarer Prozess definiert werden, der die Aufbereitung der Inhalte standardisiert.

Organisationsstrukturen und Vorgehensmodell: Es gab kein ausgeprägtes, zentrales Software-Management. Das Software-Management der Einzelkomponenten wurde durch die verantwortlichen Einzelinstitutionen (AP-Leitung) durchgeführt. Für die Versionspflege wurde meist SVN verwendet.

Im Vorgehen wurde unterschieden zwischen der lauffähigen SW-Implementierung und einer parallelen Test- und Entwicklungsumgebung, sodass für die Testnutzer (möglichst) immer eine lauffähige Implementierung zur Verfügung stand. Insgesamt wurde drei Entwicklungszyklen durchlaufen.

Arbeitsorganisation und Arbeitsverteilung:

- Projektmanagement (AWI), zentrale Steuerung und Ausrichtung der Entwicklung, gestützt durch ein vierteljährliches Treffen aller Projektpartner.
- Technisches Projektmanagement war nicht zentral vorhanden, sondern in der Hand der einzelnen technischen Partner, die ihre Entwicklung im eigenen Bereich selbst organisierten. Synchronisation der Entwickler über unregelmäßige Treffen und Abstimmungsrounds (je nach gemeinsamem Bedarf)
- Rollen: Projektmanagement/Projektleitung (AWI), Kunden/Nutzer (repräsentiert durch Partner aus der Klimaforschung im Projekt), Entwickler (technische Partner im Projekt), Tester (teilweise repräsentiert durch Nutzer sowie durch Techniker). Eine feine Aufteilung im Sinne des Standard-Softwaremanagementprozesses gab es im Projekt nicht.

Entwicklungsprozess:

- Projektorganisation und Entwicklungsrichtung des gesamten Projektes durch die Projektleitung
- Entwickler organisierten sich entsprechend der Arbeitspakete. Synchronisation der Entwicklungen auf informellen Treffen oder im Rahmen der vierteljährlichen Treffen. Zusätzliche Synchronisation über monatliche Absprache per Videokonferenz.
- Festlegung von Meilensteinen für das Gesamtprojekt (Generationen 0 bis 2) der Software sowie Deliverables, die in der Verantwortung der Arbeitspakete bearbeitet und schließlich ins Gesamtprojekt integriert wurden. Daraufhin folgte eine Testphase und 'Produktivstellung' des Meilensteins.
- Eskalationswege über die Projektleitung oder über Projekt-interne Kommunikationsinfrastruktur (VC, Mailinglisten)

Werkzeugunterstützung:

- Gemeinsames SVN-Repository, das jedoch nicht von allen Partnern genutzt wurde, da z.T. lokale eigene Systeme vorgehalten wurden
- Wiki

Fazit, Meinungen und Verbesserungsmöglichkeiten:

- Die dezentrale Entwicklung und die Aufteilung der Entwicklung in einzelne Softwaremanagementprozesse der Partner bietet den Vorteil, dass bereits vor Beginn der Entwicklungsarbeiten hinreichende Absprachen bei der Konzeptentwicklung und Entkopplung des Systems getroffen werden müssen, so dass jede 'Gruppe' eigenständig entwickeln kann.

- Ein Nachteil der entkoppelten Entwicklungsteams ist die schwierige Sicherstellung der Einhaltung von Milestones und Deliverables sowie der anschließende Synchronisations- und Integrationsprozess. Insbesondere während der Entwicklung sollte mehr Synchronisationsaufwand betrieben werden, um spätere, langwierige Phasen der Fehlersuche zu vermeiden.
- Es sollte einen übergeordneten Softwaremanagementprozess geben, der einen strikteren Rahmen für die dezentrale Entwicklung der einzelnen Komponenten schafft.

Umgang mit Risiken: Zentrale Implementierungsentscheidungen wurden auf den Plenartreffen zur Diskussion gestellt und im Konsens mit allen Arbeitspaketen getroffen. Es wurde darauf geachtet, dass parallel zur aktuellen Entscheidung immer eine getestete Fall-back-Lösung zur Verfügung stand, wenn auch nicht mit dem vollen Funktionsumfang. Dieses Vorgehen war soweit erfolgreich, bis auf die AAA-Sicherheitsinfrastruktur. Hier stand keine Fall-back-Option zur Verfügung. Dieser Aspekt war jedoch auch niemals Gegenstand des Projektantrags.

Entdeckung von Risiken und Strategien zur Vermeidung:

- Erste Anforderungsanalyse, Ermittlung des finanziellen Bedarfs und Zeitplanung bereits im Rahmen der Antragsphase. Damit stand ein grober Rahmen für die Projektlaufzeit fest.
- Evaluationsphase zu Beginn des Projektes zur Feststellung folgender Daten:
 - Einzusetzende Technologien (für die Implementierung), Festlegung von zu verwendenden Standards zur Reduzierung des evtl. auftretenden Migrationsaufwandes bei Technologiewechsel
 - Genaue zeitliche Planung der verschiedenen Entwicklungsstufen (diese Planung wurde im Laufe des Projektes immer wieder verfeinert). Diese Planung hing insbesondere mit der Verteilung der Fördergelder bei den Partnern zusammen, so dass auch hier eine genauere Planung der Arbeitsaufgaben der einzelnen Partner erfolgen konnte.
- Kooperation und Kommunikation mit anderen Projekten, um Doppelentwicklungen ggf. zu vermeiden.
- Auftretende Risiken wurden im Gremium besprochen und durch einen gemeinsamen Beschluss akzeptiert oder ein alternativer Weg zur möglichen Vermeidung beschlossen. Risiken auf Ebene der technischen Entwicklung wurden von den technischen Partnern beurteilt, von diesen Partnern wurden dann Strategien zur Vermeidung entwickelt.

Fazit, Meinungen und Verbesserungsmöglichkeiten:

- In den meisten Fällen war der Rahmen, der durch den Antrag gegeben war, für die Risikoabschätzung ausreichend, da bereits in der Antragsphase alle Partner eine Aufwandschätzung abgegeben hatten.
- Die zeitliche Planung hätte jedoch feiner durchgeführt werden müssen, um auf Probleme während der Entwicklungsphasen besser reagieren zu können. So kam es mehrmals im Projektverlauf zu Verzögerungen, die eine Integration von Meilensteinen verzögerten und damit die Zeit- und Aufgabenplanung stark veränderten.

Testumgebungen, Testverfahren und Qualitätsmanagement: Definierte Versionen der C3-Grid Prototyp-Entwicklung wurden "eingefroren", als stabile Versionen auf Workshops und Konferenzen präsentiert sowie den ausgewählten Testnutzern zur Verfügung gestellt. Das Feedback wurde aufgenommen und ausgewertet. Fehler in der Testversion wurden sofort behoben. Wünsche zur Funktionserweiterung flossen in die Entwicklung der nächsten Generation ein.

Test- und Qualitätsziele:

- Als Qualitätskriterium wurden die definierten Use Cases und der darin spezifizierte Funktionsumfang herangezogen. Weitere Kriterien waren Stabilität, Performanz und Nutzerfreundlichkeit.

Testverfahren und Testumgebungen:

- Übergreifende Integrationstests,
- Workflows, die als Use Cases identifiziert wurden,
- Belastungstests,
- Testumgebung bestehend aus einer parallelen Infrastruktur zur Evaluation der Produktionsfähigkeit. Anschließende Überführung in den Produktionsbetrieb.

Test- und Zertifizierungsprozesse innerhalb der Community: Keine derartigen Prozesse im C3Grid

Fazit, Meinungen und Verbesserungsmöglichkeiten:

- Es fehlte ein strukturierter Prozess, der schließlich auch zu einer Projekt-internen Zertifizierung führen kann.
- Whitebox-Unit-Testing von Globus-Diensten gestaltet sich schwierig.
- Fehlersuche in verteilt erzeugten Logs der Grid-Infrastruktur in den unterschiedlichen Administrationsdomänen erforderte viele Telefonate, um die notwendigen Informationen zusammenzutragen (zeitlich aufwendig).

Lebenszyklus nach der Fertigstellung und Nachhaltigkeit

Deploymentzyklen:

- Bestimmt durch die Meilensteine des Projektplans: (Generationenfolge G0, G1, G2)
- Festlegung einer Zeitspanne für das Deployment und die Integration von neuen Versionen durch das Projektmanagement
- Keine regelmäßigen Zyklen definiert, insbesondere wurden Roll-Outs nicht koordiniert durchgeführt, sondern telefonisch zwischen den technischen Partnern abgestimmt.

Betrieb und Wartung: Das C3-Grid ist in der ersten Projektlaufzeit über den Prototyp-Status nicht hinaus gekommen, da im Wesentlichen die AAA-Sicherheitsinfrastruktur nicht vollständig implementiert werden konnte. Die Software-Entwickler und -Betreiber (AWI, DKRZ, TU Dortmund, ZIB) haben vereinbart, den erreichten C3-Grid Status zu erhalten und im Rahmen ihres jeweiligen Betriebs weiter zu entwickeln. Es wird parallel nach Fördermöglichkeiten gesucht, den C3-Grid Prototyp in ein Standardwerkzeug der Erdsystemforschung zu überführen.

Einen festen Wartungsplan gibt es bisher nicht. Die Partner führen nötige Arbeiten nach Ankündigung über Mailingliste durch. Dies ist jedoch kein zentral verwalteter oder überwachter Prozess.

End-of-Life:

- Bisher keine einheitliche Vorgehensweise im Projekt.
- Da die bisherigen Entwicklungen noch Arbeiten zum Aufbau der Infrastruktur waren, wurde ein Prozess zum nachhaltigen Betrieb nicht vereinbart.

Fazit, Meinungen und Verbesserungsmöglichkeiten:

- Mit Blick auf die weitere Nutzung und Entwicklung der C3Grid-Infrastruktur ist die Einführung eines Prozesses zur Sicherstellung von Betrieb und Wartung der Ressourcen erforderlich.
- Besonders wäre im C3Grid ein Prozess für Deployment/Roll-out notwendig gewesen. Die Abstimmung unter den Partnern hat oft nicht reibungslos funktioniert und somit häufig zu Verzögerungen und Problemen in der Zuverlässigkeit des Produktivsystems geführt.

5.3.2 Portal (Partner AWI):

Das Portal übernimmt in C3Grid die Rolle der graphischen Oberfläche und der Schnittstelle zwischen Anwender und Grid-Komponenten. Über verschiedene Masken baut sich der Anwender einen Job zusammen, welcher dann mittels der Workflow Specification Language (enthält JSDL-Bestandteile) beschrieben und über die Web Service Schnittstelle an den Workflow Scheduling Service zur Weiterverarbeitung geschickt wird. Das Portal basiert auf dem JSR-168 kompatiblen Portalframework "GridSphere".

Ermittlung und Umgang mit Anforderungen:

- Spezielle Treffen zwischen Anwendern und Technikern, um Anforderungen an Oberfläche und Metadaten zu spezifizieren.
- Festlegung von Entwicklungsrichtungen und konkreter Anforderungen bei Quartaltreffen der Projektpartner.
- Da wiederholt Kommunikationsschwierigkeiten zwischen Wissenschaftlern und Informatikern aufgetreten sind, sollten zukünftig genauere Spezifikationen (z.B. in Form eines Pflichtenheftes) festgeschrieben werden, um größere Abweichungen zwischen IST und SOLL zu vermeiden.

Organisationsstrukturen und Vorgehensmodelle:

- Ein Entwickler am Institut verantwortlich für Portal und Data Information Service (Volltext-index für alle Metadaten)
- Kurze Eskalationswege durch direkte Kommunikation zwischen Nutzer und Entwickler

Umgang mit Risiken:

- Im Laufe der Entwicklung war ein Rückzug der GridSphere-Entwickler zu verzeichnen, welches im März 2008 in der völligen Einstellung der Entwicklung des Portalframeworks gipfelte (keine neuen Versionen, kein Bugfixing etc. bis heute).
- Aufgrund der fortgeschrittenen Projektphase wurde mit der letzten veröffentlichten Version weitergearbeitet

Testumgebungen, Testverfahren und Qualitätsmanagement:

- Testverfahren: Funktionstests in Zusammenarbeit mit den Projektpartnern
- Keine Zertifizierungsprozesse

Lebenszyklus nach Fertigstellung und Nachhaltigkeit:

- Keine Deploymentzyklen festgelegt
- Ständiger Betrieb und fortlaufende Wartung der Software über das Projektende hinaus
- Wartungsarbeiten führen zum kurzfristigen Ausfall des Dienstes
- Zukünftig werden aufgrund der unklaren Zukunft von GridSphere die Portlets in ein anderes Portalframework überführt werden.

5.3.3 Workflow Scheduling Service (Partner TUDo)

Der Workflow Scheduling Service (WSS) übernimmt im C3Grid die Analyse und Ausführung von zuvor in einem XML-Dialekt und JSDL formulierten Arbeitsprozessen (Workflows) der Klimawissenschaftler. Die Komponente ist in der C3Grid-Architektur unterhalb des Portals angeordnet und verwaltet und optimiert die Ressourcennutzung im C3Grid, zusammen mit dem Datenmanagementdienst. Aufbauend auf dem Globus Webservice Stack wurde der WSS im Globus WSRF in Java implementiert. Durch die Umsetzung einer Service-orientierten Architektur konnte der Workflowdienst in einem getrennten Arbeitspaket entwickelt werden.

Softwaremanagement beim WSS In diesem Abschnitt werden nur die vom allgemeinen C3Grid-Vorgehen abweichenden Aspekte des Softwaremanagements betrachtet.

Umgang mit Anforderungen

- Spezielle Treffen zwischen Klimaforschern und Technikern, die auf die Definition der Anforderungen an Workflowmanagement, Scheduling und Datenmanagement ausgerichtet waren
- Gewicht der Anforderungsarten: Funktionale Anforderungen standen im frühen Projektverlauf stark im Vordergrund. Aspekte der Sicherheit, Performance usw. wurden zunächst vernachlässigt.
- Produkte: Festlegung der Anforderungen mittels schematischer Darstellungen und Ablaufdiagramme, Schnittstellendefinitionen zwischen WSS und Datenmanagement sowie Portal, prototypische Realisierung einzelner Aspekte zur funktionalen Evaluation

Organisationsstruktur und Vorgehensmodell

- Kleingruppe von Entwicklern im Institut
- Keine feste Rollenzuteilung
- Vorgehensmodell: Extreme-Programming / Pair Programming
- Interne Abstimmung, flache Hierarchie, keine dedizierten Eskalationswege

Umgang mit Risiken

- Zu Entdeckung von Risiken vor dem Entwicklungsprozess wurden technologische Studien durchgeführt
- Aspekte der Risikoanalyse: Insbesondere Technologiewandel, Zeitplanung zumindest als Rahmen durch das Projekt vorgegeben, weitere Verfeinerungen der Planung durch Zwischenziele und definierte Deliverables

Testumgebungen, Testverfahren und Qualitätsmanagement

- Testverfahren: Funktionale Tests, Unit-Tests, Integrationstests in Zusammenarbeit mit den Projektpartnern
- Testumgebung für die Langzeitevaluierung der Entwicklungen
- Keine Zertifizierungsprozesse

Lebenszyklus nach der Fertigstellung und Nachhaltigkeit

- Deploymentzyklen nicht explizit festgelegt, weder innerhalb der Entwicklungsgruppe noch im Projekt
- Versionierung der Programmpakete im Institut

- Ständiger Betrieb und fortlaufende Wartung der Software über das Projektende hinaus
- In der aktuellen Infrastruktur führen Wartungsarbeiten aber zum kurzzeitigen Ausfall der Software. Durch das existierende Testsystem kann die Downtime aber auf die Dauer des Deploymentvorgangs reduziert werden.
- Entwicklungen für die verschiedenen Infrastrukturgenerationen hatte gewöhnlicherweise Änderungen oder Anpassungen von Schnittstellen zur Folge. Hier wurde möglichst für den Zeitraum einer Generation eine Abwärtskompatibilität der meisten Schnittstellen sichergestellt. Die Migration der Schnittstellenklienten (insbesondere Portal) auf die aktuelle Version wurde unterstützt.

5.3.4 Datenmanagement Service (Partner ZIB)

Das Generation N Data Management System (GNDMS) ist ein flexibles verteiltes Grid-Daten-Managementsystem. Zum Funktionsumfang gehören u.a. der Umgang mit logischen Pfadnamen, die Durchführung von Dateitransfers, Workspace-Management, und eine kontrollierbare, automatisierte Speicherressourcenverwaltung. GNDMS unterstützt Staging und Co-Scheduling und integriert sich als Menge von WSRF-Diensten für den Globus Container in die bestehende Grid-Infrastruktur. Besonders bemerkenswerte Alleinstellungsmerkmale der entwickelten Softwarelösung sind:

- Co-scheduling für alle Datenmanagementaktivitäten
- Hot Failover und Failure Recovery für alle Datenmanagement-Aktivitäten basierend auf persistenter Speicherung des gesamten Systemzustands
- Interoperabel und grid-ready durch den Einsatz etablierter Standards wie Globus/WSRF, GSI und GridFTP
- Klare Trennung von logischer Addressierung und physikalischem Speichermanagement
- Hochgradig konfigurier- und erweiterbar durch generische Datenmanagement-Aktivitätsplugins und den Einsatz eines systemweiten Verzeichnisdienstes auf der Basis von MDS
- Komponenten für das Systemmanagement: Laufzeit-Monitoring und verteiltes Logging

GNDMS wurde im Rahmen von C3Grid entwickelt, ist aber flexibel und anpassbar und daher für den weiteren Einsatz in zukünftigen Grid-Projekten geeignet.

Softwaremanagement beim GNDMS In diesem Abschnitt werden nur die vom allgemeinen C3Grid-Vorgehen abweichenden Aspekte des Softwaremanagements betrachtet.

Umgang mit Anforderungen

- Spezielle Treffen zwischen Klimaforschern und Technikern, die auf die Definition der Anforderungen an Datenmanagement, Workflowmanagement und Scheduling ausgerichtet waren.

- Gewicht der Anforderungsarten: Funktionale Anforderungen standen im frühen Projektverlauf stark im Vordergrund. Aspekte der Sicherheit, Performance etc. wurden später adressiert.
- Produkte: Festlegung der Anforderungen mittels schematischer Darstellungen und Ablaufdiagrammen, Schnittstellendefinitionen zwischen WSS und Datenmanagement sowie Portal, prototypische Realisierung einzelner Aspekte zur funktionalen Evaluation im Wiki und als Dokument

Organisationsstruktur und Vorgehensmodell

- Kleingruppe von Entwicklern im Institut mit einem Hauptverantwortlichen für die GNDMS-Softwarearchitektur
- Vorgehensmodell: Rapid-Prototyping, Sprints, Vorbereitung von Templates zur Ausprogrammierung (durch die Datenprovider), Pair-Programming für kritische Systemkomponenten
- Interne Abstimmung, flache Hierarchie, definierte Eskalationswege

Umgang mit Risiken

- Zu Entdeckung von Risiken vor dem Entwicklungsprozess wurden Prototypen entwickelt. Die Rahmenbedingungen konnten aber nicht immer vollständig richtig abgeschätzt werden. Weiterentwicklungen an genutzten Basiskomponenten außerhalb des Projektes erforderten die Neubewertungen getroffener Technologieentscheidungen während des Projektes.
- Aspekte der Risikoanalyse: Technologiewandel, weitere Verfeinerungen der Planung durch Zwischenziele und definierte Deliverables unter Berücksichtigung des Technologiewandels (Generationenfolge G0, G1, G2).

Testumgebungen, Testverfahren und Qualitätsmanagement

- Testverfahren: Funktionale Tests, Unit-Tests, Integrationstests in Zusammenarbeit mit den Projektpartnern
- Testumgebung für die Langzeitevaluierung der Entwicklungen
- Definierte Test- und Dokumentationsprozesse für Releasekandidaten und Patches
- Keine Zertifizierungsprozesse
- *Randomized Testing*
- Telefonhotline insbesondere nach Deployment
- Interne Versionsverwaltung mit git

Lebenszyklus nach der Fertigstellung und Nachhaltigkeit

- Deploymentzyklen nicht explizit festgelegt, weder innerhalb der Entwicklungsgruppe noch im Projekt
- Versionierung der Releases (Nachvollziehbarkeit)
- Ständiger Betrieb und fortlaufende Wartung der Software über das Projektende hinaus
- In der aktuellen Infrastruktur führen Wartungsarbeiten aber zum kurzzeitigen Ausfall der Software. Durch das existierende Testsystem kann die Downtime aber auf die Dauer des Deploymentvorgangs reduziert werden.
- Entwicklungen für die verschiedenen Infrastrukturgenerationen hatte gewöhnlicherweise Änderungen oder Anpassungen von Schnittstellen zur Folge. Hier wurde möglichst für den Zeitraum einer Generation eine Abwärtskompatibilität der meisten Schnittstellen sichergestellt. Die Migration der Schnittstellenklienten (Datenprovider) auf die aktuelle Version wurde unterstützt.

5.4 Life-Cycle-Management im HEP-CG

5.4.1 Allgemeine Angaben zum HEP-CG-Projekt

Zum Zeitpunkt des Redaktionsschlusses für dieses Dokument lagen noch keine Materialien vor. Die Datensammlung bei den Partnern aus dem HEP-CG-Projekt ist angestoßen worden. Bisher haben einige Aspekte bezüglich Organisationsstruktur und Softwareentwicklung bereits Einzug in andere Kapitel gefunden. Sobald ausreichend Materialien vorliegen, werde diese editiert und in das Dokument eingepflegt.

5.5 Lifecycle Management im MediGrid-Projekt

5.5.1 Allgemeine Angaben zum MediGrid-Projekt

In Services@MediGRID ist jeder Entwickler eines Services selbst für das Lifecycle Management seines Services zuständig. Es gibt keine klaren, zentralen Vorgaben durch das Projekt. Dieser Umstand hat zur Folge, dass die Ermittlung der Anforderungen, die Risikoanalyse etc., in der Hand von jedem Entwickler liegt. Auch die dafür eingesetzten Techniken (Wiki, SVN, etc.) liegen im Verantwortungsbereich der einzelnen Entwickler und sind nicht vorgegeben.

Der geplante Fortschritt bei der Entwicklung der Services ist durch Meilensteine im Projektplan festgehalten. In den monatlichen Telefonkonferenzen erfolgt eine Rückmeldung der Partner, inwiefern die Meilensteine eingehalten werden können und an welchen Stellen Verzögerungen zu erwarten sind. Bei den halbjährlichen Konsortialtreffen werden (Teil-)Ergebnisse präsentiert und vorgestellt.

Festgelegt ist hingegen der Prozess der Anwendungsintegration [11]. Darin ist ein Verfahren beschrieben, welches bei Neuerungen bei den Portlets (also den portalbasierten Userinterfaces zu den Grid-Services) eingesetzt wird. Dieses sieht zunächst einen Test auf dem Entwicklerportal (portal2) vor. Erst bei erfolgreich absolviertem Test auf dem Entwicklerportal wird im Produktivsystem deployt. Was die Services, Anwendung bzw. Software angeht, erfolgt zunächst eine parallele Installation der neuen Version auf einer Ressource. Dann erfolgen Tests - ggf. inkl. Tests der neuen

Portlets. Wenn diese Tests erfolgreich sind, werden die neuen Versionen auf allen (dafür vorgesehenen) Ressourcen installiert, und im Portal wird ggf. das neue Portlet deployt. Da in den Portlets die Workflowbeschreibungen hinterlegt sind, geht alles andere dann von selbst, sofern die Informationssysteme aktualisiert worden sind. Dies ist aber bereits für die Tests erforderlich und beeinflusst in der Regel die alten Versionen nicht.

5.5.2 Grid Workflow Execution Service (Fraunhofer FIRST)

Die vorliegenden Informationen beziehen sich auf die zentrale Software Komponente GWES (Grid Workflow Execution Service), der im MediGRID weitgehend eingesetzten Software zur (graphischen) Modellierung, Beschreibung, dem Management und der Ausführung von Workflows.

Ermittlung und Umgang mit Anforderungen

- Anforderungen der Nutzer-Community in Medi-Grid (Projekt-interne Abstimmung, Themenbereiche der Anwendungen: Bildverarbeitung, biomedizinische Informatik und klinische Forschung.)
- Da GWES bereits vor Beginn des Projektes existierte, mussten Anpassungsarbeiten zur Umsetzung der Ziele identifiziert werden.
- Hilfsmittel: Strukturdiagramme, Datenflussdiagramme

Organisationsstrukturen und Vorgehensmodell

- Projektleitung und Konsortialführung
- Einzelne Arbeitspakete (Module) mit weitreichenden Verantwortlichkeiten für die technische Umsetzung
- Keine dedizierte IT-Koordination (siehe Abschlussbericht). Jeder Partner pflegte jedoch ein eigenes Entwicklungsvorgehen in meist kleinen Entwicklergruppen.

Umgang mit Risiken

- Risiken waren aufgrund der Adaption eines existierenden (und funktionierenden) Systems als beherrschbar einzustufen.
- Als Middlewarelösung wurde auf die im Grid-Umfeld etablierte Globus-Middleware (Version 4.0.x) gesetzt, die ebenfalls im Zusammenspiel mit dem verwendeten GWES zu betrachten ist.
- Die Zeitplanung war weitgehend durch den Projektantrag gegeben, ebenso wie die zur Verfügung stehenden Mittel.

Testumgebungen, Testverfahren und Qualitätsmanagement

- JUnit-Tests
- Integrationstest
- Unterstützung des Entwicklungsprozesses (Apache Maven), unterstützt Integrationstests mit jedem Build
- Es standen zwei Systeme, ein Testsystem und ein Produktivsystem, zur Verfügung. Das Testsystem diente zur Erprobung neuer Entwicklungen, das Produktivsystem für den stabilen Betrieb.
- Entwicklungsumgebung: IntelliJ idea V 8.x
- Source Code-Verwaltung mittels Subversion

Lebenszyklus nach der Fertigstellung und Nachhaltigkeit

- Automatisiertes Deployment
- Keine festen Vorgaben für Lebenszyklus und Deployment
- Keine zertifizierten Prozesse
- Nachhaltigkeit: Integration des GWES im DGSI-Projekt (Interoperabilität mit anderen Metaschedulern)

5.6 Life-Cycle-Management im TextGrid

5.6.1 Allgemeine Angaben zum TextGrid Projekt

Ermittlung und Umgang mit Anforderungen

Vorgehen bei der Ermittlung der Anforderungen

- Die gleichberechtigte Zusammenarbeit von Nutzern und nutzernahen Institutionen (TU Darmstadt, Uni Würzburg, Uni Trier etc.) mit (Grid-)Technikern
- Anforderungsanalyse in der Antragsphase: Anforderungen wurden insbesondere aus den konkreten Erfahrungen der beteiligten FachwissenschaftlerInnen heraus formuliert, die bereits dort in die Planung verschiedener Arbeitsschritte und einer Grobarchitektur mündete.
- Spezifikationsphase nach Projektbeginn: Im Laufe einer ersten Evaluationsphase wurden Anwendungsfälle⁴ (genauer beispielhafte Use Cases und Workflows der Geistes- und Kulturwissenschaften, insbesondere der Editionsphilologie und der Lexikographie) ausgewählt, anhand derer die prototypische Infrastruktur und zugehörige Dienste definiert wurden.

⁴TextGrid Szenarien. <http://www.textgrid.de/information/berichte.html>

- Entwicklungsphase: Nutzerworkshops, in deren Rahmen den Nutzern Prototypen der TextGrid-Infrastruktur vorgestellt wurden (z.B. philtag Workshops, Präsenz auf TEI Konferenz). Die Nutzer konnten die Infrastruktur testen und weitere Anforderungen formulieren.
- Kontinuierliche Erweiterung: Wie in jeder nachhaltigen Infrastruktur ist die Entwicklung an TextGrid nie wirklich abgeschlossen. Zu den ständigen Feedbackmechanismen und Kanälen zur Erweiterung der Anforderungen zählen: Feedback über Webseite und Mailinglisten, regelmäßige wiederkehrende Tests, Diskussion auf Konferenzen, Hackfeste mit neuen Communities, etc. (siehe auch weiter unten)

Gewichtung funktionaler und nicht-funktionaler Anforderungen: Zu Beginn des Entwicklungsprozesses wurde insbesondere auf die funktionalen Anforderungen an das System Wert gelegt, zu einem späteren Zeitpunkt standen nicht-funktionale Anforderungen gleichberechtigt im Fokus der Entwicklung. Die funktionalen Anforderungen ergaben sich vor allem aus der initialen Formulierung von editionswissenschaftlichen und lexikographischen Nutzungsszenarien. Zu den nicht-funktionalen Anforderungen zählen:

- Einfache Handhabbarkeit: Besonders wichtig bei einer nicht prinzipiell technikaffinen Community
- Einbettung in "natürliche" Umgebung: der Nutzer sollte so wenig bzw. so reibungslos als möglich zwischen unterschiedlichen Arbeitsumgebungen wechseln müssen (z.B. dezidierte Web bzw. Grid Services zur Analyse und Auswertung; offline-fähiges Textwerkzeug zur Edition und Annotation; Instituts-Hompage zur Publikation).
- Erweiterbarkeit, Nachnutzbarkeit: TextGrid ist als organisch wachsende Infrastruktur konzipiert (vgl. Partizipation der Community in Modifikationsprozess). In aktuellen Schlagworten ist TextGrid sowohl als *Infrastructure as a Service*, als *Platform as a Service*, wie auch als *Software as a Service* zu sehen – jeweils für unterschiedliche Rollen und Kompetenzen, die sich aber gegenseitig fördern.
- Nachhaltigkeit: Eine gewisse Langfristigkeit ist essenziell – Archivierung der Daten, Verfügbarkeit und Pflege der Dienste. Zum Abgleich mit der im vorigen Punkt genannten Flexibilität ist dies gestaffelt: Basisdienste aus der TextGrid Infrastruktur haben maximale Stabilität; Plattform-Dienste können sich absehbar (mit Ankündigung und Support) ändern; Software ist je nach Nutzerbasis mitunter "sehr" flexibel.
- Zuverlässigkeit der Infrastrukturdienste wie Datenmanagement und Nutzerverwaltung / Rechtemanagement. (OBS aus Sicht der TextGrid Architektur ist Datenmanagement explizit ein Basisdienst, kein "höherwertiger" Dienst.)

Produkte des Anforderungsmanagements

- In Antragsphase: Projektantrag
- Anforderungserhebung nach Projektbeginn: Szenarien⁵ mit Dokumentation der Anforderungen durch Nutzer- und Workflowbeschreibungen

⁵TextGrid Szenarien. <http://www.textgrid.de/information/berichte.html>

- Definition von Schnittstellen in TextGrid Architektur⁶
- Tool-Spezifikation⁷
- Umsetzung verschiedener Prototypen zur Demonstration der funktionalen (prototypische Dienste) und nicht-funktionalen Anforderungen auch im Rahmen von Benutzer-Workshops

Werkzeugunterstützung

- Wiki zur Dokumentation der Anforderungen und Produkte des Anforderungsmanagements
- WebDAV zum Austausch von Beispieldaten (inzwischen auch DropBox)
- Anmerkung: Erstellung von Mockups in HTML, Powerpoint, etc.
- Anmerkung: Erstellung von Architektur- und Ablaufdiagrammen in Powerpoint

Fazit, Meinungen und Verbesserungsmöglichkeiten

- Die große Heterogenität der Geistes- und Kulturwissenschaften bzw. der Textwissenschaften erforderte eine ausführliche und intensive Diskussion und Bestandsaufnahme unterschiedlicher (berechtigter) Anforderungen sowie eine klare Schwerpunktsetzung.
- Kommunikationshürde zwischen Nutzern und technischen Partnern war nicht immer leicht zu überbrücken. Hier war und ist Vermittlung notwendig.
- Veranstaltungen zur Integration von weiteren Nutzeranforderungen sind sehr wichtig und sollten ggf. häufiger durchgeführt werden, um die Anforderungen der Nutzer außerhalb des Projektes besser aufzunehmen.
- Der Einsatz von multimodalen Kollaborationsmechanismen (Wiki, Chat: Skype/IRC, Video-konferenzen, etc.) sowie von regelmäßigen Programmiersprints (persönlichen Treffen) hat sich in TextGrid als sehr erfolgreich erwiesen. Wesentlich für den Erfolg dieser Mechanismen ist aber jedenfalls ein Kernteam an Mitarbeitern, die aktiv die Kommunikation moderieren (z.B. Wiki-Bereiche aktuell halten oder Artefakte aufräumen).

Organisationsstrukturen und Vorgehensmodell

Arbeitsorganisation und Arbeitsverteilung

- Projektmanagement (SUB), übergreifende Moderation
- Kooperative Entscheidungsprozesse durch vierteljährliche Treffen aller Projektpartner, dazwischen regelmäßige virtuelle Treffen
- Begleitung durch einen internationalen Fachbeirat

⁶TextGrid Architektur (R3.2), Manual (R3.5), und Datengrid (R3.6). <http://www.textgrid.de/information/berichte.html>

⁷TextGrid Tools, R2.x. <http://www.textgrid.de/information/berichte.html>

- Intensive Zusammenarbeit vor allem auch durch AP-übergreifende AGs, die das Projekt teilweise durchgehend begleiteten: AG Textformate (Nutzer), AG Architektur (Entwickler), AG Tests, ...
- Intensive Zusammenarbeit über mehrere Kommunikationswege: Wiki, Mailinglisten, Chat, Telefon- und Videokonferenzen

Entwicklungsprozess

- Projektmoderation des gesamten Projektes durch die Projektleitung bzw. durch die AP-Struktur
- Entwickler kollaborieren eng in AGs. Im Rahmen der AP-übergreifenden AGs organisierten sich die Entwickler in wöchentlichen Absprachen über Video- oder Telefonkonferenz bzw. Chat. Zwischen diesen wöchentlichen Treffen auch Absprachen je nach Task. Programmiersprints (mehrtägige physische Treffen zur Team-Programmierung) zumindest einmal im Quartal
- Festlegung von Meilensteinen der Software sowie Deliverables mit Management über AGs und Bericht an AP-Leiter bzw im Plenum. Release-Zyklen etwa alle 8 Wochen

Werkzeugunterstützung

- Wiki
- Jira als Ticket-System
- Gemeinsames Subversion

Fazit, Meinungen und Verbesserungsmöglichkeiten

- Je nachdem wie gut Architektur und Schnittstellen im Vorhinein spezifiziert werden können, ist eine dezentrale Entwicklung möglich. In TextGrid war enge Interaktion zwischen den Entwicklern essenziell, z.B. um Mehrfachentwicklungen zu verhindern und die Entstehung einer homogenen Plattform zu fördern. Regelmäßige "Programmiersprints" von allen Technikern waren dazu sehr hilfreich, und eine noch stärkere team-orientierte (as opposed to task-orientierte) Entwicklung wäre hilfreich gewesen.
- Gerade in großen, verteilten Projekten ist es wesentlich, dass (ein Kernteam) auch wirklich dem Projekt zur Verfügung steht, nicht bloß mit 20 % ihrer Arbeitszeit — diese Zeit ist alleine schon nötig, um auf dem Laufenden zu bleiben.
- Ein übergreifender, transparenter Softwareentwicklungsprozess war nicht nur innerhalb des Entwicklerteams, sondern auch in der Interaktion mit Nutzern sehr hilfreich. Noch stärkere Agile Programmierung (release early and release often) wird in der weiteren Folge von TextGrid intensiv verfolgt — rapid prototyping, iterative Entwicklung.⁸

⁸Agile Softwareentwicklung. http://de.wikipedia.org/wiki/Agile_Softwareentwicklung

Umgang mit Risiken

Entdeckung von Risiken und Strategien zur Vermeidung

- Erste Anforderungsanalyse, Ermittlung des finanziellen Bedarfs und Zeitplanung bereits im Rahmen der Antragsphase. Damit stand ein grober Rahmen für die Projektlaufzeit fest.
- regelmäßige Projekttreffen, in denen (a) der aktuelle Stand abgefragt wurde, (b) weitere Meilensteine und Ziele geplant wurden
- enge Kooperation, wodurch schon sehr früh Verzögerungen deutlich wurden
- Auftretende Risiken wurden im Gremium besprochen und durch einen gemeinsamen Beschluss akzeptiert oder ein alternativer Weg zur möglichen Vermeidung beschlossen. Risiken auf Ebene der technischen Entwicklung wurden von den technischen Partnern beurteilt und von diesen Partnern Strategien zur Vermeidung entwickelt.
- Wo nötig auch Auslagerung von Tasks in Absprache mit dem Projektträger. Enge internationale Kooperation

Fazit, Meinungen und Verbesserungsmöglichkeiten

- In den meisten Fällen war der Rahmen, der durch den Antrag gegeben war für die Risikoabschätzung ausreichend, da bereits in der Antragsphase alle Partner eine Aufwandschätzung abgegeben hatten.
- Eine entspannte, offene Kommunikation in regelmäßigen Treffen war essenziell, um bei Problemen alternative Lösungsmöglichkeiten erst möglich zu machen. Der Blick war immer nach vorne, wodurch sich eine sehr offene Kommunikationskultur ergeben hat.

Testumgebungen, Testverfahren und Qualitätsmanagement

Test- und Qualitätsziele

- Gemeinsame Ziele in Antrag formuliert und in enger Kooperation wo nötig nachgebessert.
- Übergreifende Qualitätsziele (siehe Abschnitt zu "Anforderungen" oben)

Testverfahren und Testumgebungen

- Selbst zusammengestellte Continuous Integration Umgebung mit nightly Builds und Integrationstests
- Trennung Test- und Produktionsumgebungen. Es gibt 2 Testumgebungen (jeweils für Basisdienste und höherwertige Services) auf einem physisch separaten Server, sowie die Möglichkeit virtuelle Server zu generieren, wenn spezifische Umgebungen (andere Java-Version, anderer Web Server, wilde Threads in Container, etc) temporär nötig sind.
- Nutzertests: Beta-Version mit Feedbackmechanismen, regelmäßige Anwenderworkshops

Test- und Zertifizierungsprozesse innerhalb der Community

- Testprozesse: Siehe letzter Punkt
- Zertifizierungsprozesse: Prozess zur Aufnahme von neuen Features in den Produktionsbetrieb; Zusammenführung der Tests in einer Projekt-übergreifenden AG Tests

Fazit, Meinungen und Verbesserungsmöglichkeiten

- Tests ändern sich je nach Projektphase sehr stark. In der Entwicklung über den Verlauf von "TextGrid 1" waren im ersten Jahr lediglich "intellektuelle" Tests (i.e. menschliches Feedback) sinnvoll, im zweiten dann gegenseitige Unterstützung und lediglich interne Tests in kollaborativer Entwicklungsarbeit, und im dritten dann schließlich der Aufbau der Continuous Integration Umgebung mit externen Tests. Im Rahmen der Erweiterung von TextGrid um neue Zielgruppen wird der weitergehende Aufbau einer Community-Umgebung erweitert und strukturiert Feedback von Nutzern in von TextGrid organisierten Anwenderworkshops eingeholt.

Lebenszyklus nach der Fertigstellung und Nachhaltigkeit

Deploymentzyklen

- Bestimmt durch die Meilensteine des Projektplans, bzw. durch gemeinschaftlich abgestimmte Projektziele (z.B. Präsentation auf Konferenz xy, Ausrichtung eines Workshops im Januar 200x, etc.)
- Release-Zyklen etwa alle 8 Wochen, Abstimmung und Planung im Rahmen der AP-übergreifenden technischen AG Architektur
- Release der Beta-Version als Entscheidung des gesamten Projektteams, regelmäßiges Reporting über den aktuellen Stand im Plenum (auf physischen oder virtuellen Treffen).
- Versionierung der Softwarepakete

Betrieb und Wartung

- Produktivumgebung, die möglichst stabil gehalten wird. In der zweiten Projektphase wird dazu ein high-availability-Konzept umgesetzt.
- Zuteilung von Verantwortlichkeiten innerhalb des Projektes. Bei Engpässen bespricht das Projekt kooperativ Alternativen.
- Partner führen nötige Arbeiten mit Ankündigung über Mailingliste durch, kein zentral verwalteter oder überwachter Prozess

End-of-Life

- Aufbau eines nachhaltigen Betriebs. Derzeit Finanzierung gesichert bis 2012.
- Aufbau eines "Modifikationsprozesses" und Community-Prozesses

Fazit, Meinungen und Verbesserungsmöglichkeiten: Software-Lebenszyklus-Management ist eng verknüpft mit dem Aufbau der TextGrid Community und der Community-Prozesse. Schon jetzt wird eng mit externen Partnern z.B. in den Niederlanden (Huygens Instituut) zusammengearbeitet, bzw. Kollegen einer verwandten Software (TUStep) arbeiten an einem Gateway, den sie auch selbst warten würden.

5.7 Vorläufige Bewertung

Dieser Abschnitt schließt das Kapitel der Sammlung von Inhalten bzgl. des Software-Lifecycle-Managements mit einer vorläufigen Bewertung der Erhebungsergebnisse ab.

Bei der Betrachtung der erhobenen Daten fallen etliche Vorgehensweisen und Organisationsformen ins Auge, die bei den meisten Communities während der Softwareentwicklung ähnlich umgesetzt wurden:

Organisation in kleinen Teams: Innerhalb der meisten Communities wurde die Entwicklung von Software durch unabhängige Entwicklerteams mit klar abgegrenzten Aufgabenbereichen durchgeführt. Insbesondere wurde die Entwicklung eines speziellen Dienstes durch einen einzigen Projektpartner, selten auch durch einen wenige Partner umfassenden Verbund geleistet. Da dieses Vorgehen in allen Projekten anzutreffen ist, lässt dies auf eine übliche und unter den gegebenen Förderrahmenbedingungen angemessene Form der Entwicklung schließen.

Zusätzlich kann festgestellt werden, dass neben den dedizierten Entwicklerteams Anwender in jedem Projekt ebenfalls durch kleine Teams repräsentiert werden und die klassische Rolle der Kunden einnehmen, die Anforderungen an das zu erstellende System liefern. Daraus ergibt sich auch in akademischen Projekten ein internes Wechselspiel, wie es gewöhnlich im Softwareentwicklungsprozess auftritt.

Individuelles Softwaremanagement: Durch die Bildung spezieller Entwicklungsteams mit einer thematisch festgelegten Entwicklungsaufgabe ergibt sich offenbar verbreitet eine dezentrale und individuelle Gestaltung des Entwicklungsprozesses durch die beteiligten Projektpartner. In fast allen Projekten ist festzustellen, dass es keinen ausgeprägten, übergeordneten Software-Management-Prozess gibt. Dabei hat die Konsortial- oder Projektleitung oft nur einen oberflächlich steuernden Charakter, und nimmt nur bei weitreichenden Technologieentscheidungen oder in Plenarsitzungen direkten Einfluss auf das Entwicklungsvorgehen.

Bemerkenswert ist in diesem Zusammenhang, dass etliche an Projekten beteiligte Partner eine stärkere Organisationsform befürworten. Dies weist auf die Notwendigkeit einer genaueren Analyse des dezentralen Entwicklungsvorgehens hin und legt die Entwicklung von Prozessen zur stärkeren Zentralisierung des gesamten Entwicklungsprozesses nahe. Hier ist allerdings auf eine gewisse Balance zwischen unabhängiger Entwicklung in kleinen Teams und einem klar geführten zentralen Management zu finden.

Übergreifende Kommunikation: Die Erhebung hat gezeigt, dass es bereits während der ersten Förderphase erhebliche Anstrengungen zur Kommunikation zwischen den Projektpartnern in jedem Community-Grid gab. Die zahlreichen und regelmäßigen Treffen von Arbeitsgruppen und Projektpartnern (Techniker-, Arbeitsgruppen- und Plenartreffen) weisen auf einen erheblichen Kommunikations- und Koordinationsbedarf hin. Im Sinne des gemeinsamen und geregelten Softwaremanagements ist also die geeignete Kommunikation der beteiligten Projektpartner ein wichtiger Punkt, der in den folgenden Monaten betrachtet werden muss und Aufnahme in die Blaupause für das Softwaremanagement finden muss.

Rahmenwerk des Antrages: Als bestimmendes Element für den Verlauf der Softwareentwicklung in einem Community-Grid scheint sich der Projektantrag abzuzeichnen. Hier werden bereits Rahmenbedingungen festgelegt, die den Verlauf des Projektes stark bestimmen. Neben entscheidenden Technologiefestlegungen werden hier bereits die verfügbaren finanziellen Mittel geplant und das Risiko abgeschätzt.

Hier ist zu überprüfen, ob aus Sicht des Softwaremanagements Festlegungen und Entscheidungen bereits während der Antragsphase durch geeignete Prozesse vorbereitet und unterstützt werden können.

Neben den hier herausgestellten Gemeinsamkeiten gibt es auch weitere spezielle Aspekte, die die meisten Projekte oft in ähnlicher Weise angegangen haben, die aber hier nicht ausführlich diskutiert werden, da dies eine ausführliche und genaue Betrachtung vieler Einzelaspekte umfasst, die den Rahmen dieses Dokumentes übersteigen. In den folgenden Monaten wird auf Basis der hier erfassten Daten und ggf. zusätzlich beschaffter Detailinformationen ein erster Entwurf für eine Softwaremanagement im Kontext akademischer Grid-Projekte entwickelt.

6 Interaktion mit der Community

Die beiden Abschnitte 4 und 5 hängen als spezifisch Community-bezogene Punkte, in denen es darum geht, Nutzer außerhalb der Projektgruppe anzusprechen und mit ihnen zu kommunizieren, eng zusammen. Es lassen sich Bereiche identifizieren, die generell für alle Communities relevant sind, während andere (z.T. sehr) spezifisch erscheinen. Es ist zu prüfen, inwieweit die übergreifenden Aspekte zusammengefasst werden können, dies ließe im Gegenzug die Community-spezifischen Aspekte deutlicher hervortreten.

Der Punkt **Interaktion mit der Community** bietet kurze stichwortartige Informationen zu den Unterpunkten 'Workshops' (vom Community-Projekt ausgerichtet), 'Präsenz auf Community-Tagungen', 'Nutzung der CG-Dienste in der Community'. Alle beteiligten Communities haben eigene Schulungen durchgeführt, z.T. differenziert in fachwissenschaftliche (Anwender) und informatische (Entwickler) Workshops (z.B. HEP, TextGrid), und waren sowohl auf weiteren fachspezifischen wie informatischen Konferenzen präsent. Diese Aktivitäten sind generell wichtig für die Verbreitung der Grid-Idee und entsprechen den üblichen universitären bzw. wissenschaftlichen Kommunikationsstrukturen. Differenzen zeigen sich vor allem bei der Nutzung der CG-Dienste, was z.T. wohl durch den unterschiedlichen Entwicklungsstand der CG-Projekte bzw. durch den unterschiedlichen Grad der Akzeptanz und Verankerung der Technologie in den Communities bedingt ist. Gültig für alle scheint der Dienste-Einsatz in (z.T. kleineren Pilot-)Projektgruppen. Bei einigen gibt es Ansätze zur Integration in universitäre Curricula (z.B. AstroGrid und TextGrid), dieser Punkt scheint besonders wichtig für die Verstetigung durch die Heranführung des wissenschaftlichen Nachwuchses

an die Technologie. Selten und weniger von genereller Bedeutung, d.h. von Community-spezifischer Bedeutung, scheint kommerzielle Nutzung (MediGrid).

6.1 AstroGrid-D

- Workshops:
zu Globus (letzter 2008)
Instituts-Präsentationen
Mitarbeit / Präsenz bei GAVO und IVOA-Veranstaltungen
- Präsenz auf Community-Tagungen/Kongressen/ usw:
Beiträge und Stand auf Community-Veranstaltungen (Jahrestagung Astron. Gesellschaft)
OGF-RG Workshop mit AstroGrid UK 2008
Beiträge zu ADASS, IVOA usw.
- Nutzung der CG-Dienste in der Community:
Gruppen- und Einzelnutzung der Grid-Ressourcen
Kollaboration mit Ukraine (GRAPE)
vorbereitet wird Kurs an Uni
- Einsatz im "D-Grid Scheduler Interoperability (DGSi)" Projekt

6.2 HEP Grid

- Workshops:
Workshops, Tutorials und Schulungen im Rahmen von EGEE, D-Grid und Terascale Alliance.
Workshops im Photon Science Umfeld, z.B. im Rahmen des ESRF Upgrade Programms.
- Präsenz auf Community-Tagungen/Kongressen/ usw:
Präsenz auf nationalen und internationalen Veranstaltungen, EGEE, D-Grid, EGI.
Im Bereich Photon Science: Kooperation mit PSI und ESRF für xray.eu-VO.
Beteiligung an EU-weitem Projekt PanData mit fast allen führenden Synchrotron- und Neutronen-Quellen, darin unter anderem Präsenz auf Photon-Science spezifischen User-Meetings wie auch in EGI.
- Nutzung der CG-Dienste in der Community:
Nutzung durch die LHC-Experimente Atlas und CMS, durch ILC, die Hera-Experimente (Hermes, H1, Zeus), IceCube, ILDG und der Terascale Allianz der Helmholtz-Gemeinschaft.
Nutzung durch Photon Sciences (BioMed) fuer BioInformatics, sowie Evaluierung-Projekte in der xray.eu-VO. Evaluierung von Diensten wie Amga durch Photon Science Service Anbieter wie FLASH oder GKSS.
Keine weitergehende Nutzung durch Photon Science zur Zeit.

6.3 C3 Grid

- User-Workshops zur Sammlung von Rückmeldungen und Anregungen aus der Community und zur Vorstellung der Entwicklungen

- Präsenz auf nationalen und internationalen Klima-Tagungen
- Nutzung der CG-Dienste in der Community durch kleine Pilot-Nutzergruppe

6.4 TextGrid

- Workshops:
Release Workshops (technisch und fachwissenschaftlich separat) von TextGrid 1.0 im Rahmen des TextGrid Summits (Januar 2009). Teil der zweiten Projektphase sind User-Evaluationen und Tutorials.
- Präsenz auf Community-Tagungen/Kongressen/usw:
Präsenz auf nationalen und internationalen Veranstaltungen, sowohl in geisteswissenschaftlichen Fach-Tagungen (z.B. philtag, Digital Humanities), als auch in technischen Veranstaltungen (z.B. IEEE, OGF)
- Nutzung der CG-Dienste in der Community:
Vorbereitet werden sowohl Wissenschaft als auch Lehre: Die Nutzung im Rahmen von Universitäts-Curricula (Literaturwissenschaften) an Uni Würzburg und Uni Trier; Nutzung von TextGrid in wissenschaftlichen DFG Projekten; Kooperationen mit Grossbritannien (JISC Projekt TEXT-vre); und Kooperation mit USA (North-Western University) angedacht

6.5 MediGrid

Das „Forum Grid“ der TMF dient als gemeinsame Plattform der biomedizinischen Grid-Projekte. Die gemeinsame Abstimmung der biomedizinischen Grid-Projekte in Deutschland erfolgt somit über das Forum Grid. Weiterhin werden durch das Forum Grid die internationalen Kooperationen mit HealthGrid auf europäischer Ebene und caBIG in den USA ausgebaut.

- Workshops:
Durch Entwickler-Workshops (2008 eintägig, 2009 zweitägig) werden die Anwendungsentwickler in dem Projekt geschult und es finden Diskussionen (Rückmeldungen/Anregungen/Anfragen) statt.
- Präsenz auf Community-Tagungen/Kongressen/usw:
Es findet eine Präsenz auf nationalen und internationalen (bio-)medizin-informatischen Fachtagungen sowie D-Grid-Workshops statt.
- Nutzung der CG-Dienste in der Community:
Nutzung erfolgt innerhalb der MediGRID-Community. Durch den kommerziellen Partner Bayer Technologie (BTS) erfolgt eine Testnutzung.

7 Support

Der Punkt **Support** ist gegliedert durch konkrete Fragen, die von den jeweiligen Communities abgearbeitet wurden, so dass zum einen gute Vergleichbarkeit gewährleistet ist, zum anderen sehr

konkrete Angaben erfolgt sind. Insgesamt scheinen die Strukturen und die dazugehörigen Dokumente und Materialien recht Community-spezifisch, d.h. heterogen. Dies ist jedoch verständlich, da sich die Strukturen aufgrund der unterschiedlichen Anforderungen der Communities unterscheiden müssen; die nicht unbedingt technik-affinen Geisteswissenschaftler bspw. benötigen eine andere Art von Support als grundlegend technisch orientierte Disziplinen. Das bedeutet, dass in diesem Bereich sorgfältig differenziert werden muss und eventuell noch viel Diskussions- und Abstimmungsbedarf besteht. Es wird zu prüfen sein, inwiefern Strukturen übertragen werden können bzw. generalisierbar sind.

Beide Kapitel sind wichtige Ausgangspunkte, um die Blaupausen für das Heranführen neuer Communities an das Grid optimal ausrichten zu können. Insgesamt könnte hier wohl noch am meisten ausdifferenziert werden, denn es gibt zwar generische Elemente bzw. Instrumente wie „Workshops“, gerade die Ansprache ist aber aufgrund der heterogenen Anforderungen Community-spezifisch. Darauf wäre besonders zu achten.

7.1 AstroGrid-D

7.1.1 Benutzeranleitungen und andere schriftliche Support-Dokumente

Gibt es ein einheitliches Dokumentenmanagement? Werden Support-Dokumente evaluiert? Wie wird die Aktualität der Support-Dokumente gewährleistet?

Ein zentral verwaltetes Dokumentenmanagement gibt es nur für Deliverables und andere offizielle Dokumente. Evaluiert werden solche Dokumente durch a) die WorkGroup, b) das Projekt. Es gibt hierzu das 'Documents and Standards Recommendations', in dem das Verfahren geregelt wurde. Viele Support-Dokumente liegen im projekteigenen Web. Für Web-Seiten gibt es eine einfache *on agreement* Evaluation durch das Autorenteam / Workgroup.

Die Aktualität von Support-Dokumenten wird nicht durch eine formalisierte Prozedur gewährleistet. Im Web darf jeder, der ein Nutzerkonto besitzt, auch in jede Seite hineinschreiben. Auch Support-Seiten werden so im Team aktualisiert.

Wer stößt die Überarbeitung eines Support-Dokuments an?

Überarbeitung eines Support-Dokuments: ist im AstroGrid-Projekt nicht formal festgelegt. Es wurde meistens von den Arbeitsgruppen bzw. der Architektur-Arbeitsgruppe angestoßen.

Dienen wissenschaftliche Aufsätze oder Vortragsfolien gelegentlich als Ersatzdokumentation?

Nein

In welchen Dateiformaten liegen Support-Dokumente in der Regel vor?

Alle Dokumente stehen entweder im SVN als LaTeX, und als PDF oder HTML-Text im Intranet zur Verfügung.

Welche Support-Dokumente liegen auf einem zentralen Server, welche werden dezentral zur Verfügung gestellt (z.B. auf Servern der Software-Entwickler)?

Software-Dokumentation bei den Software-Repositories, Rest auf zentralen Servern (Wiki etc.)

7.1.2 Training

Gibt es Wegweiser (Präsenztraining/Nutzer-Workshops, Tutorien, FAQ, ...) für Anfänger in Ihrer Community? Welche?

Globus-HandsOn Workshops (2x) und Instruktion durch örtliche Experten

Gibt es Wegweiser für Anfänger / Anwender innerhalb Ihrer Community zu relevanten Informationen national/international korrelierter Aktivitäten/Initiativen - dort angebotener Trainings, Workshops etc. usw.?

Informationen zu übergreifenden D-Grid-Aktivitäten durch Mailingliste, analog für internationale Aktivitäten. Ankündigung auf Telefonkonferenzen.

Gab es eine Zusammenarbeit mit SuGI?

Nein

Gibt es Tutorien/e-Learning zu Applikationen und anderen speziellen Grid-Themen, die ohne Beteiligung von SuGI entstanden sind?

Ja. Materialien zu den Hands-On Workshops, App2Grid-Guide[4]

7.1.3 Weitere Support-Strukturen

Gibt es getrennte Support-Strukturen für verschiedene Gruppen (z.B. eine Mailingliste für Nutzer, eine weitere für Implementierer)? Welche Gruppe nutzt welche Struktur?

Nutzer-Support per Mail durch AIP-Team. Außerdem gibt es die Admin Mailing Liste.

Wie viele Hilfeersuchen gehen jährlich in etwa bei jeder dieser Support-Strukturen ein?

Wenige via Mail, viele via Telefon

Werden Hilfeersuchen durch einen Moderator/First-Level-Support auf die Support-Mitarbeiter verteilt oder liest jeder Support-Mitarbeiter alle Eingaben?

Dies wurde nicht formell geregelt.

Wird den Nutzern eine Liste von Mail-Adressen oder Telefonnummern zur Verfügung gestellt, mit Hilfe derer unter Umgehung des bestehenden Ticketsystems bzw. der bestehenden Mailingliste auch direkt bei einem Spezial-Support gefragt werden kann?

Ja, diese enthält die RAs der Institute, Admins, Ansprechpartner

Wie werden Applikationen ins Grid gebracht? Gibt es dafür Unterstützung und wie ist diese organisiert?

Unterstützung durch Entwickler / Projektmitarbeiter

7.2 HEP Grid

7.2.1 Benutzeranleitungen und andere schriftliche Support-Dokumente

Gibt es ein einheitliches Dokumentenmanagement? Werden Support-Dokumente evaluiert? Wie wird die Aktualität der Support-Dokumente gewährleistet?

Es existiert kein zentral verwaltetes Dokumentenmanagement. Die Referenzen zu Deliverables und anderen offiziellen Dokumenten wie Veröffentlichungen werden zentral gepflegt, die Objekte selbst liegen jedoch verteilt.

Da das HEPCG in die WLCG bzw. EGEE-Aktivitäten eingebunden ist, sind viele generische Support-Dokumente z.B. in die EGEE-Seiten eingeflossen.

Experiment- oder Instituts-spezifische Dokumentation wird auf Wikis bereit gestellt und gepflegt. Zum Beispiel pflegen Atlas, CMS oder ILC eigene Wikis. Die Support-Dokumente werden entsprechend von den Experimenten evaluiert. Dies erfolgt unter anderem auch in regelmäßigen User-Meetings unter Beteiligung der Service-Betreiber.

Die Aktualität von Support-Dokumenten wird nicht durch eine formalisierte Prozedur gewährleistet.

Im Rahmen der globalen HEP-Community werden insbesondere Anforderungs-Profile in Meetings abgestimmt und formalisiert.

Wer stößt die Überarbeitung eines Support-Dokuments an?

In der Regel wird die Überarbeitung eines Support-Dokuments durch die Anwender initiiert. Eine aktive Revision der Support-Dokumente ist zumindest nicht der Regelfall.

Dienen wissenschaftliche Aufsätze oder Vortragsfolien gelegentlich als Ersatzdokumentation?

Ja. Sowohl Publikationen in Form von Proceedings als auch Vortragsfolien finden Verwendung als Referenz.

In welchen Dateiformaten liegen Support-Dokumente in der Regel vor?

PDF oder Wiki

Welche Support-Dokumente liegen auf einem zentralen Server, welche werden dezentral zur Verfügung gestellt (z.B. auf Servern der Software-Entwickler)?

Spezifische Software-Dokumentation wie zum Beispiel für dCache werden bei den Repositories der Entwickler gehostet.

Die HEPCG spezifischen Support-Dokumente sind nicht zentral verfügbar. Allerdings sind alle relevanten Dokumente zentral auf den Seiten der EGEE und lokalisierten Servern wie zum Beispiel DECH, die Support für Deutschland, Schweiz und CERN anbieten.

7.2.2 Training

Gibt es Wegweiser (Präsenztraining/Nutzer-Workshops, Tutorien, FAQ, ...) für Anfänger in Ihrer Community? Welche?

Die Experimente Atlas und CMS bieten regelmäßige Nutzer-Workshops, Hands-on-Tutorials an.

Die werden auf den Wikis, dem Web und Experiment-Seiten bekannt gegeben. Im Rahmen der Terascale-Allianz, die sehr stark mit dem HEPCG überlappt, erfolgen Ankündigungen weiterhin über monatliche erscheinende Newsletter.

Gibt es Wegweiser für Anfänger / Anwender innerhalb Ihrer Community zu relevanten Informationen national/international korrelierter Aktivitäten/Initiativen - dort angebotener Trainings, Workshops etc. usw.?

Lokale wie überregionale Support-Strukturen (z.B. EGEE, DECH, DESY) bieten entsprechende Wegweiser zu Tutorien, FAQs und Workshops an.

Gab es eine Zusammenarbeit mit SuGI?

Nein

Gibt es Tutorien/e-Learning zu Applikationen und anderen speziellen Grid-Themen, die ohne Beteiligung von SuGI entstanden sind?

Ja

7.2.3 Weitere Support-Strukturen

Gibt es getrennte Support-Strukturen für verschiedene Gruppen (z.B. eine Mailingliste für Nutzer, eine weitere für Implementierer)? Welche Gruppe nutzt welche Struktur?

Die Experimente unterhalten ihre eigenen Support-Strukturen und agieren als 1st Level-Support gegenüber den Anwendern. Im NAF-Umfeld werden Mailing-Listen für Atlas, CMS, Announcements sowie für Administratoren getrennt angeboten und über ein Best-Practical Tracking System abgebildet, das gleichzeitig eine Schnittstelle zu globalen Support-Strukturen wie das GGUS Ticketing System abietet.

Wie viele Hilfeersuchen gehen jährlich in etwa bei jeder dieser Support-Strukturen ein?

Außerordentlich unterschiedlich und aufgrund der verteilten Support-Strukturen kaum zu ermitteln. Für die NAF waren es 250 generische Support-, 450 Admin-Anfragen und 38 Announcement, aber das bildet nur einen kleinen Teil der Support-Anfragen ab.

Werden Hilfeersuchen durch einen Moderator/First-Level-Support auf die Support-Mitarbeiter verteilt oder liest jeder Support-Mitarbeiter alle Eingaben?

Generische Anfragen werden von einem Help-Desk angenommen und entweder von diesem bearbeitet oder an entsprechende Support-Strukturen weitergeleitet. Dies erfolgt über das erwähnte Best Practical Tracking System.

Wird den Nutzern eine Liste von Mail-Adressen oder Telefonnummern zur Verfügung gestellt, mit Hilfe derer unter Umgehung des bestehenden Ticketsystems bzw. der bestehenden Mailingliste auch direkt bei einem Spezial-Support gefragt werden kann?

Nein. Allerdings neigen Anwender bisweilen dazu, den Weg durch die Support-Strukturen abzukürzen, sobald ein Spezial-Support bekannt wird.

Wie werden Applikationen ins Grid gebracht? Gibt es dafür Unterstützung und wie ist diese organisiert?

Die Experimente verfügen über eigene Strukturen, um Applikationen im Grid zu installieren. Support für spezielle Applikationen (z.B. ROOT, gLite User Interfaces oder Ganga) wird angeboten.

7.3 C3 Grid

7.3.1 Benutzeranleitungen und andere schriftliche Support-Dokumente

Gibt es ein einheitliches Dokumentenmanagement? Werden Support-Dokumente evaluiert? Wie wird die Aktualität der Support-Dokumente gewährleistet?

Ein zentral verwaltetes Dokumentenmanagement gibt es nur für Deliverables und andere offizielle Dokumente. Evaluiert werden solche Dokumente durch das Autorenteam selbst. Viele Support-Dokumente liegen im projekteigenen Wiki. Für Wiki-Seiten gibt es eine einfache *on agreement* Evaluation durch das Autorenteam.

Die Aktualität von Support-Dokumenten wird nicht durch eine formalisierte Prozedur gewährleistet. Im Wiki darf jeder, der ein Nutzerkonto besitzt, auch in jede Seite hineinschreiben. Auch Support-Seiten werden so im Team aktualisiert.

Wer stößt die Überarbeitung eines Support-Dokuments an?

Wer die Überarbeitung eines Support-Dokuments anstößt, ist im C3Grid-Projekt nicht formal festgelegt, es gab sowohl die *top down* Initiative durch die Projektkoordination als auch den *bottom up* Weg durch Leser oder Autoren.

Dienen wissenschaftliche Aufsätze oder Vortragsfolien gelegentlich als Ersatzdokumentation?

Nein

In welchen Dateiformaten liegen Support-Dokumente in der Regel vor?

PDF oder Wiki

Welche Support-Dokumente liegen auf einem zentralen Server, welche werden dezentral zur Verfügung gestellt (z.B. auf Servern der Software-Entwickler)?

Software-Dokumentation bei den Software-Repositories, Rest auf zentralen Servern (Wiki etc.)

7.3.2 Training

Gibt es Wegweiser (Präsenztraining/Nutzer-Workshops, Tutorien, FAQ, ...) für Anfänger in Ihrer Community? Welche?

Nutzer-Workshops

Gibt es Wegweiser für Anfänger / Anwender innerhalb Ihrer Community zu relevanten Informationen national/international korrelierter Aktivitäten/Initiativen - dort angebotener Trainings, Workshops etc. usw.?

Informationen zu übergreifenden D-Grid-Aktivitäten durch Mailingliste, analog für internationale Aktivitäten

Gab es eine Zusammenarbeit mit SuGI?

Ja [14]

Gibt es Tutorien/e-Learning zu Applikationen und anderen speziellen Grid-Themen, die ohne Beteiligung von SuGI entstanden sind?

Ja

7.3.3 Weitere Support-Strukturen

Gibt es getrennte Support-Strukturen für verschiedene Gruppen (z.B. eine Mailingliste für Nutzer, eine weitere für Implementierer)? Welche Gruppe nutzt welche Struktur?

Es gibt die C3-Support-Mailingliste für Administratoren und weitere getrennte Mailinglisten für Entwickler. Nutzer sind im C3Grid Portalnutzer und wenden sich an den Portalverwalter.

Wie viele Hilfeersuchen gehen jährlich in etwa bei jeder dieser Support-Strukturen ein?

Außerordentlich unterschiedlich, zu Demo-/Tutorienterminen viele im Vorfeld und während des Ablaufes, sonst wenige, da Prototyp-Betrieb der Infrastruktur

Werden Hilfeersuchen durch einen Moderator/First-Level-Support auf die Support-Mitarbeiter verteilt oder liest jeder Support-Mitarbeiter alle Eingaben?

Portalverwalter schickt Nutzeranfragen an die C3-Support-Mailingliste oder — wenn das Problem direkt einer Person zugeordnet werden kann — an einen Mitarbeiter.

Wird den Nutzern eine Liste von Mail-Adressen oder Telefonnummern zur Verfügung gestellt, mit Hilfe derer unter Umgehung des bestehenden Ticketsystems bzw. der bestehenden Mailingliste auch direkt bei einem Spezial-Support gefragt werden kann?

Nein

Wie werden Applikationen ins Grid gebracht? Gibt es dafür Unterstützung und wie ist diese organisiert?

Unterstützung durch Entwickler auf Anfrage

7.4 TextGrid

7.4.1 Benutzeranleitungen und andere schriftliche Support-Dokumente

Gibt es ein einheitliches Dokumentenmanagement? Werden Support-Dokumente evaluiert? Wie wird die Aktualität der Support-Dokumente gewährleistet?

Alle Dokumente werden in der Wiki verwaltet. Die finale Version ist Word bzw. PDF. Je nach Wunsch des Autorenteam werden die Dokumente in Google Docs, direkt in der Wiki, oder von Anfang an in Word verfasst. Die Moderation übernehmen die Autoren selbst.

Evaluation erfolgt bei offiziellen Dokumenten über Freigabe des gesamten Projektes. Deliverables und andere offizielle Dokumente werden auch noch einmal vereinheitlicht (Templates, etc). Interne Dokumente "leben" in der Wiki. Dokumentation für die (projekt-externen) Nutzer liegt in Form von Online-Hilfe im TextGridLab, publizierten Dokumenten, Screencasts, etc. vor.

Die Aktualität von Support-Dokumenten wird nicht durch eine formalisierte Prozedur gewährleistet. Im Wiki darf jeder, der ein Nutzerkonto besitzt, auch in jede Seite hineinschreiben. Auch Support-Seiten werden so im Team aktualisiert.

Wer stößt die Überarbeitung eines Support-Dokuments an?

Wer die Überarbeitung eines Support-Dokuments anstößt, ist nicht formal festgelegt. Oftmals geschieht dies nach Bedarf.

Dienen wissenschaftliche Aufsätze oder Vortragsfolien gelegentlich als Ersatzdokumentation?

Nein, dies ist aber bei wachsender Größe der Community denkbar.

In welchen Dateiformaten liegen Support-Dokumente in der Regel vor?

1. Wiki (intern, für Projektmitglieder)
2. PDF oder Word (für externe Nutzer)
3. Screencasts, Workshop-Video-Mitschnitte (für externe Nutzer)

Welche Support-Dokumente liegen auf einem zentralen Server, welche werden dezentral zur Verfügung gestellt (z.B. auf Servern der Software-Entwickler)?

Die gesamte Dokumentation liegt in der zentralen Projekt-Wiki. Nur Code-Dokumentation ist im SVN.

7.4.2 Training

Gibt es Wegweiser (Präsenztraining/Nutzer-Workshops, Tutorien, FAQ, ...) für Anfänger in Ihrer Community? Welche?

1. Manuals und Online-Tutorials für Nutzer und Entwickler (zielgruppenorientiert)
2. Regelmäßige Nutzer- und Entwickler-Workshops
3. Feedbackmöglichkeiten über die Website, Entwickler-Mailingliste
4. Ticketsystem (JIRA)

Gibt es Wegweiser für Anfänger / Anwender innerhalb Ihrer Community zu relevanten Informationen national/international korrelierter Aktivitäten/Initiativen - dort angebotener Trainings, Workshops usw.?

1. Regelmäßige Publikation eines Newsletters (siehe Webseite)
2. Regelmäßige Konferenzbesuche, Workshop-Präsentationen
3. Direkte Teilnahme an Projekttreffen verwandter Initiativen (z.B. Bamboo in den USA, Dariah in Europa)

Gab es eine Zusammenarbeit mit SuGI?

Video-Aufzeichnung der Konferenz "TextGrid Summit" (<http://www.textgrid.de/konferenzen/summit.html>), und auch andere Vorträge zu TextGrid sind im SUGI-Portal zu finden (<http://www.google.de/search?q=textgrid+site%3Asugi.d-grid.de>).

Gibt es Tutorien/e-Learning zu Applikationen und anderen speziellen Grid-Themen, die ohne Beteiligung von SuGI entstanden sind?

Ja, ein Text-Tutorial (siehe <http://www.textgrid.de/information/berichte.html>)

7.4.3 Weitere Support-Strukturen

Gibt es getrennte Support-Strukturen für verschiedene Gruppen (z.B. eine Mailingliste für Nutzer, eine weitere für Implementierer)? Welche Gruppe nutzt welche Struktur?

Ja, die Dokumentation ist zielgruppenorientiert aufbereitet (Nutzer, Entwickler, und auch Archive).

Wie viele Hilfeersuchen gehen jährlich in etwa bei jeder dieser Support-Strukturen ein?

Dies fluktuiert derzeit noch stark, es ist zu früh dies zu beziffern.

Werden Hilfeersuchen durch einen Moderator/First-Level-Support auf die Support-Mitarbeiter verteilt oder liest jeder Support-Mitarbeiter alle Eingaben?

Es gibt Moderatoren und ein Ticket-System.

Wird den Nutzern eine Liste von Mail-Adressen oder Telefonnummern zur Verfügung gestellt, mit Hilfe derer unter Umgehung des bestehenden Ticketsystems bzw. der bestehenden Mailingliste auch direkt bei einem Spezial-Support gefragt werden kann?

Nein, derzeit kein 24/7 Helpdesk. Anfragen gibt es aber mitunter auch persönliche, z.B. bei Konferenz-Bekanntschäften, etc.

Wie werden Applikationen ins Grid gebracht? Gibt es dafür Unterstützung und wie ist diese organisiert?

Derzeit wird das Einbringen neuer Applikationen noch im Gesamtprojekt diskutiert, bzw. hängt dies von Projektförderung ab. Sobald eine Applikation auf der "Roadmap" ist, gibt es die volle Unterstützung: Teilnahme an Programmiersprints, Einbettung in die Entwickler-Mailingliste, direkter Zugriff auf Kern-TextGrid-Entwickler.

In Zukunft ist ein erweiterter Community-Prozess denkbar, der über die nächsten Jahre entwickelt werden könnte.

7.5 MediGrid

7.5.1 Benutzeranleitungen und andere schriftliche Support-Dokumente

Gibt es ein einheitliches Dokumentenmanagement? Werden Support-Dokumente evaluiert? Wie wird die Aktualität der Support-Dokumente gewährleistet?

Ein zentral verwaltetes Dokumentenmanagement gibt es nur für Deliverables und andere offizielle

Dokumente. Evaluiert werden solche Dokumente durch das Autorenteam selbst. Einige Support-Dokumente liegen im projekteigenen Wiki.

Die Aktualität von Support-Dokumenten wird nicht durch eine formalisierte Prozedur gewährleistet. Im Wiki darf jeder, der ein Nutzerkonto besitzt, auch in jede Seite hineinschreiben. Auch Support-Seiten werden so im Team aktualisiert.

Wer stößt die Überarbeitung eines Support-Dokuments an?

Wer die Überarbeitung eines Support-Dokuments anstößt, ist nicht formal festgelegt, es gab sowohl die *top down* Initiative durch die Projektkoordination als auch den *bottom up* Weg durch Leser oder Autoren.

Dienen wissenschaftliche Aufsätze oder Vortragsfolien gelegentlich als Ersatzdokumentation?

Die Vortragsfolien insbesondere von dem Entwickler-Workshops in MediGRID [12] können den Anwendungsentwicklern als Leitfaden und Ersatzdokumentation dienen.

In welchen Dateiformaten liegen Support-Dokumente in der Regel vor?

PDF oder Wiki

Welche Support-Dokumente liegen auf einem zentralen Server, welche werden dezentral zur Verfügung gestellt (z.B. auf Servern der Software-Entwickler)?

Das MediGRID-Wiki befindet sich auf einem zentralen Server. Die Deliverables werden zentral im Sharepoint-System der Universitätsmedizin Göttingen abgelegt. Die Dokumentation beispielsweise der Anwendungsentwicklung erfolgt dezentral durch die beteiligten Projektpartner.

7.5.2 Training

Gibt es Wegweiser (Präsenztraining/Nutzer-Workshops, Tutorien, FAQ, ...) für Anfänger in Ihrer Community? Welche?

Regelmäßig (derzeit jährlich) wird ein Entwickler-Workshop [12] angeboten, bei dem die Anwendungsentwickler die Grundlagen für die Nutzung der MediGRID-Infrastruktur vermittelt bekommen.

Gibt es Wegweiser für Anfänger / Anwender innerhalb Ihrer Community zu relevanten Informationen national/international korrelierter Aktivitäten/Initiativen - dort angebotener Trainings, Workshops etc. usw.?

Informationen zu übergreifenden D-Grid-Aktivitäten durch Mailingliste bzw. Webseite, analog für internationale Aktivitäten

Gab es eine Zusammenarbeit mit SuGI?

Ja [12]

Gibt es Tutorien/e-Learning zu Applikationen und anderen speziellen Grid-Themen, die ohne Beteiligung von SuGI entstanden sind? (Ja/Nein)

Ja, es gibt Tutorien beispielsweise durch die Vortragsfolien des Entwickler-Workshops 2008 oder durch das Dokument "Guidelines for the Deployment of Applications in MediGRID" [11].

7.5.3 Weitere Support-Strukturen

Gibt es getrennte Support-Strukturen für verschiedene Gruppen (z.B. eine Mailingliste für Nutzer, eine weitere für Implementierer)? Welche Gruppe nutzt welche Struktur?

Es gibt in Services@MediGRID neben einer allgemeinen Mailingliste zwei Mailinglisten für spezielle Themen (Anwendungsentwicklung und Visualisierung). Daneben wird das DGUS-System von D-Grid verwendet.

Wie viele Hilfeersuchen gehen jährlich in etwa bei jeder dieser Support-Strukturen ein?

Grundsätzlich treffen eher wenige Anfragen ein. Dies kann zu speziellen Anlässen allerdings variieren. Häufig werden die Mailinglisten eher für Ankündigungen genutzt.

Werden Hilfeersuchen durch einen Moderator/First-Level-Support auf die Support-Mitarbeiter verteilt oder liest jeder Support-Mitarbeiter alle Eingaben?

Jeder Benutzer der Mailingliste erhält sämtliche Anfragen. Eine Moderationsstruktur existiert nicht.

Wird den Nutzern eine Liste von Mail-Adressen oder Telefonnummern zur Verfügung gestellt, mit Hilfe derer unter Umgehung des bestehenden Ticketsystems bzw. der bestehenden Mailingliste auch direkt bei einem Spezial-Support gefragt werden kann? (Ja/Nein)

Ja, in dem Dokument "Guidelines for the Deployment of Applications in MediGRID" [11] befinden sich Ansprechpartner für bestimmte Aufgaben (z.B. Portal-Administrator, Ansprechpartner für die verschiedenen Anwendungen etc.).

Wie werden Applikationen ins Grid gebracht? Gibt es dafür Unterstützung und wie ist diese organisiert?

Unterstützung durch Spezialisten auf Anfrage. Weiterhin beschreibt das Dokument "Guidelines for the Deployment of Applications in MediGRID" [11] die Details.

8 Dokumente und Materialien aus den CG-Grids zur Weiterverwertung

In diesem Abschnitt sind Materialien zusammengetragen, die während der Laufzeit der einzelnen akademischen Projekte erstellt wurden. Dazu gehören neben Projekt-spezifischen Dokumentationen von entwickelten Mehrwertdiensten und speziellen Prozessen auch Dokumente von übergreifendem Interesse, in denen Installationsanleitungen für Projekt-übergreifend genutzte Middleware zu finden sind.

Diese Dokumente werden hier — grob thematisch sortiert bzgl. jedes einzelnen Community-Grids — gesammelt, um als Referenz- und Ergänzungsmaterial für die spätere Entwicklung von Blaupausen verwendet zu werden. Tatsächlich sind Informationen aus diesen Dokumenten nicht vollständig in die hier vorliegende Materialsammlung (Deliverable 2.1.1) eingeflossen, sodass die im Folgenden angeführten Dokumente eine wertvolle Ergänzung der gesammelten Materialien darstellen.

8.1 AstroGrid-D

Das AstroGrid-D hat - im Unterschied zum DGI - eine Installation von den Sourcen des Globus-Toolkit gefordert, im Hinblick auf die Absicht, das MPICH-G2 device für das Clustering über Institutsgrenzen hinweg zu nutzen[2].

8.1.1 Dokumentation von allgemeinem Interesse

- Es sind eine Reihe von Scripten zum Konfigurieren einer Globus-Installation, unter Einschluss von Ganglia, im *globus-helper-package* zusammengestellt[3].
- H. Enke, S. White: "Running MPI Jobs on Grid Resources", 1. Juni 2008 http://www.astrogrid-d.org/project-documents/deliverables/wp1/Run_MPI_Jobs_on_Grid-D.1_4.pdf
- H. Enke, S. White: "MPI over the Grid", 10. April 2008 http://www.astrogrid-d.org/project-documents/deliverables/wp1/MPI_over_Grid-D.1.5.pdf
- M. Höggqvist: "Metadata Information Service", 22. August 2008 <http://stellaris.zib.de/files/releases/stellaris-0.3.1.tar.gz>
- Stephan Braune, Frank Breitling, Arthur Carlson, Mikael Höggqvist, Thomas Radke, Tobias Scholl, Steve White: "Advanced Prototype Implementation of Metadata Information Providers" 18. August 2008 http://www.astrogrid-d.org/project-documents/deliverables/wp2/D2_9_Information_Providers.pdf
- A. Beck-Ratzka: "GAT-Installation mit generischer Grundfunktionalität", 8. Januar 2007 <http://www.astrogrid-d.org/project-documents/deliverables/wp7/7-3.pdf>

8.1.2 Dokumente bezüglich AstroGrid-D

- F. Breitling, M. Braun: "Providing Static Metadata of Robotic Telescopes to Stellaris", 15. Mai 2007 http://www.astrogrid-d.org/project-documents/deliverables/wp2/D2_4.pdf
- F. Breitling: "Providing Dynamic Metadata of Robotic Telescopes to Stellaris" 29. April 2008 http://www.astrogrid-d.org/project-documents/deliverables/wp2/D2_7.pdf
- T. Scholl: "Data Stream Management - An Implementation Report", 31. Juli 2008 <http://www.astrogrid-d.org/project-documents/deliverables/wp4/wg4-d8-1.0.0.pdf>
- Aus den Erfahrungen, die aus dem Portieren von Applikationen auf das Grid gewonnen wurden, ist ein App2Grid-Guide entstanden[4].
- F. Breitling, T. Granzer, H. Enke, "First Integration of Robotic Telescopes" 31. August 2007 <http://www.astrogrid-d.org/project-documents/deliverables/wp5/Deliverable5.3.pdf>
- T. Röblitz: "A Broker for a Robotic Telescope Network", 31. August 2007 <http://www.astrogrid-d.org/project-documents/deliverables/wp5/D5-5.pdf>

- F. Breitling, S. Braune, H. Enke, T. Granzer, T. Rätz: "A Scheduler for a Robotic Telescope Network" 8. August 2008 <http://www.astrogrid-d.org/project-documents/deliverables/wp5/D5.8.pdf>
- T. Radke: "Implementation of Monitoring & Steering Methods for AstroGrid-D Use Cases - Final Report" 25. August 2008 http://www.astrogrid-d.org/project-documents/deliverables/wp6/WG6_D6_6.pdf
- O. Wehrens: "Installation of the GridSphere Portal Framework on a GACG Portal Server", 18. April 2006 <http://www.astrogrid-d.org/project-documents/deliverables/wp7/M2.pdf>
- O. Wehrens: "Portal User Interfaces for Metadata Management", 5. August 2008 <http://www.astrogrid-d.org/project-documents/deliverables/wp7/7-6.pdf>
- Thomas Brüsemeister beschreibt in "First steps in AstroGrid-D" wie bei AstroGrid-D die Authentifikation des Benutzers und das einreichen und überwachen eines Jobs erfolgt. Weiterhin wird der Transfer großer Datenmengen betrachtet und als Beispiel für eine Anwendung das Ausführen von NBODY6++ beschrieben[5].
- Das AstroGrid-D Data Management (ADM) stellt ein virtuelles Filesystem zur Verfügung. Es kümmert sich um die entsprechende Positionierung der korrespondierenden physikalischen Files auf einem oder mehreren Speichergeräten. Funktion und Anwendung wird in dem folgenden Tutorial beschrieben[6].
- Klaus Rieger beschreibt in "Installing Geomaint" wie GeoMaint zu installieren, zu testen und aktuell zu halten ist[7].

8.1.3 Ausgewählte Publikationen

- F. Breitling: "A standard transformation from XML to RDF via XSLT", *Astronomical Notes* 330, issue 7, pp. 755-760, 2009
- T. Scholl, A. Reiser, A. Kemper: "Collaborative Query Coordination in Community-Driven Data Grids", 18th International Symposium on High Performance Distributed Computing (HPDC 2009), June 11 - 13, 2009, Munich, Germany
- T. Scholl, B. Bauer, J. Märtl, B. Gufler, A. Reiser, A. Kemper: "Workload-Aware Data Partitioning in Community-Driven Data Grids", 12th International Conference on Extending Database Technology (EDBT 2009), March 23 - 26, 2009, Saint-Petersburg, Russia
- T. Scholl, B. Bauer, B. Gufler, R. Kuntschke, A. Reiser, A. Kemper: "Scalable Community-Driven Data Sharing in e-Science Grids", *Elsevier Future Generation Computer Systems (FGCS)*, Vol. 25, No. 3, March 2009
- T. Scholl, B. Gufler, J. Märtl, A. Reiser, A. Kemper: "P2P-Datenmanagement für e-Science-Grids" *Datenbank-Spektrum*, Vol. 8, No. 26, September 2008
- F. Breitling, T. Granzer, H. Enke: "Grid Integration of Robotic Telescopes", *Astronomical Notes* 329, issue 3, 343-346, 2008

- M. Riedel, Th. Eickermann, S. Habbinga, W. Frings, P. Gibbon, D. Mallmann, A. Streit, Th. Lippert, F. Wolf, W. Schiffmann, A. Ernst, R. Spurzem, W.E. Nagel: "Computational Steering and Online Visualization of Scientific Applications on Future Peta-Scale HPC Systems within e-Science Infrastructures", 3rd IEEE International Conference on e-Science and Grid Computing (eScience 2007), December 10 - 13, 2007, Bangalore, India
- A. Carlson, H. BÄ¶hringer, T. Scholl, W. Voges: "Finding Galaxy Clusters using Grid Computing Technology", 3rd IEEE International Conference on e-Science and Grid Computing (eScience 2007), December 10 - 13, 2007, Bangalore, India
- R Spurzem, P Berczik, I Berentzen, D Merritt, N Nakasato, H M Adorf, T BrÄ¼semeister, P Schwekendiek, J Steinacker, J Wambsganß, G Marcus Martinez, G Lienhart, A Kugel, R MÄ¶nner, A Burkert, T Naab, H Vasquez and M Wetzstein: "From Newton to Einstein - N-body dynamics in galactic nuclei and SPH using new special hardware and AstroGrid-D", J. Phys.: Conf. Ser. 78 012071; Procs. SciDAC 2007, 24-28 June 2007, Boston, Massachusetts, USA
- T. Granzer, F. Breitling, M. Braun, H. Enke, T. RÖblitz: "Providing Remote Access to Robotic Telescopes by Adopting Grid Technology", Proceedings of the German e-Science Conference, May 2-5, 2007, Baden-Baden, Germany
- M. Hoegqvist, T. RÖblitz, A. Reinefeld: "Stellaris: An RDF-based Information Service for AstroGrid-D", Proceedings of the German e-Science Conference, May 2-5, 2007, Baden-Baden, Germany
- Hovest, W., G. Stojceska, I. Saverchenko, H.-M. Adorf, T. Enßlin, Th. Riller: "Deterministic and statistical parameter space sampling – an autonomous driver for scientific workflows", Astronomical Data Analysis Software and Systems (ADASS), ASP Conference Series, Vol. XXX, (to appear). 2006
- Richard Kuntschke, Tobias Scholl, Sebastian Huber, Alfons Kemper, Angelika Reiser, Hans-Martin Adorf, Gerard Lemson, and Wolfgang Voges: "Grid-based Data Stream Processing in e-Science", Proceedings of the 2nd IEEE International Conference on e-Science and Grid Computing (eScience 2006), December 4 - 6, 2006, Amsterdam, The Netherlands

8.2 HEP Grid

Die hier gesammelten Dokumente entstammen weitgehend der InternetprÄ¶sentation des HEP-CG⁹. Dort werden die Arbeitsfortschritte des Projektes gepflegt. Es ist bei den vorliegenden Dokumenten darauf hinzuweisen, dass die Aktualisierung der InternetprÄ¶sentation mit Ablauf des Jahres 2006 endet. Die Dokumente sind also nicht aktueller als die zu diesem Zeitpunkt verfÄ¶gbaren Versionen.

8.2.1 Dokumente von allgemeinem Interesse

- dCache, The Book HTML (<http://www.dcache.org/manuals/Book/dCacheBook.pdf>)
- Pflichtenheft fÄ¼r Testsuiten (<http://documentation.hepcg.org/res/ap1/PflichtenHeft.pdf>)

⁹<http://documentation.hepcg.org/>

- UserGuide "Result Monitoring and Online Steering Tool" (<http://www.hep.physik.uni-siegen.de/grid/rmost/doc/UserGuide.pdf>)
- InstallationsGuide "Result Monitoring and Online Steering Tool" (<http://www.hep.physik.uni-siegen.de/grid/rmost/doc/InstallationGuide.pdf>)
- DevelopersGuide "Result Monitoring and Online Steering Tool" (<http://www.hep.physik.uni-siegen.de/grid/rmost/doc/DeveloperGuide.pdf>)

8.2.2 Dokumentation bezüglich HepCG

- Martin Radicke: "dCache - A Scalable Storage Element", HEPCG-Wokshop am GSi in Darmstadt, 27. April 2006 (http://documentation.hepcg.org/res/ap1/radicke_dCache-SE_darmstadt.pdf)
- Dirk Pleiter: "Using grid technologies for lattice QCD", Proceedings of Computing in High Energy Physics (CHEP06), Mumbai, India, February 2006 (http://documentation.hepcg.org/res/ap1/chep06_pleiter_ildg.pdf)
- Hinnerk Stueben: "Introduction to ILDG, the LatFor Datagrid, and I3HP Transnational Access", Lattice Practices '06 am DESY Zeuthen, 28 November 2006 (<http://documentation.hepcg.org/res/ap1/lap06-hs.pdf>)
- Hinnerk Stueben: "Metadata Catalogue", HEPCG-Wokshop an der Bergischen Universitaet in Wuppertal, 30. November 2006 (<http://documentation.hepcg.org/res/ap1/hepcg-hs.pdf>)
- Lars Schley, Stefan Freitag: "HEP CG Scheduling Architecture", HEPCG-Workshop an der Bergischen Universitaet in Wuppertal, 30. November 2006 (http://documentation.hepcg.org/res/ap1/stefanfreitag_2006-11-30-Wuppertal.pdf)
- UserGuide "Result Monitoring and Online Steering Tool" (<http://www.hep.physik.uni-siegen.de/grid/rmost/doc/UserGuide.pdf>)
- InstallationsGuide "Result Monitoring and Online Steering Tool" (<http://www.hep.physik.uni-siegen.de/grid/rmost/doc/InstallationGuide.pdf>)
- DevelopersGuide "Result Monitoring and Online Steering Tool" (<http://www.hep.physik.uni-siegen.de/grid/rmost/doc/DeveloperGuide.pdf>)
- Tutorial zur Benutzung der Grundfunktionalität des ersten Prototyps des "Result Monitoring and Online Steering Tools" (<http://www.hep.physik.uni-siegen.de/grid/rmost/tutorials/PythiaDemo/index.html>)
- Torsten Harenberg: "Job Monitoring in LCG", Gruppenvortrag auf der DPG-Frühjahrstagung 2006, Dortmund (http://www.grid.uni-wuppertal.de/eval/dpg06_harenberg.pdf)
- Ahmad Hammad, Torsten Harenberg, Matthias Hüsken, Peter Mättig, Markus Mechtel, David Meder-Marouelli: "AP2 Monitoring - Job Execution Monitor" (http://www.grid.uni-wuppertal.de/eval/JEM_Darmstadt.pdf)

- Daniel Lorenz, "The RMOST Online Steering Tool", Atlas Software Week am CERN in Genf, 12. Dezember 2006 (http://www.hep.physik.uni-siegen.de/grid/rmost/talks/CERN_1206.pdf)
- Ralph Müller-Pfefferkorn, Reinhard Neumann "User-Centric Job and Resource Usage Monitoring", HEPCG-Workshop an der Bergischen Universität in Wuppertal, 30. November 2006 (<https://indico.desy.de/contributionDisplay.py?contribId=8&sessionId=2&confId=134>)
- Gap-Analyse des automatischen Jobmanagers und der interaktiven Analyse (http://documentation.hepcg.org/res/ap3/milestone_300906.pdf)
- Anar Anar Manafov: "RGLite - A gLite plug-in for ROOT", HEPCG-Workshop am GSI in Darmstadt, 27. April 2006 (<http://documentation.hepcg.org/res/ap3/HEPCGWorkshopatGSI--RGLite.pdf>)

8.3 C3 Grid

Die folgenden Dokumentationen beschreiben grundlegende Techniken die im C3Grid entwickelt oder zur Umsetzung von gewählten Technologien durchgeführt wurden. Die Handlungsanweisungen sind C3-spezifisch; die Konzepte aber könnten auch für datenorientierte Communities außerhalb der Klimaforschung von Interesse sein. Abgesehen vom letztgenannten Howto liegen alle diese Dokumentationen auf dem C3-Wiki

<http://www.c3grid.de/wiki/>

und sind nur nach Eingabe von Benutzername und Passwort zugänglich. Benutzername und Passwort können bei Benny Bräuer (benny.braeuer@awi.de) erfragt werden.

8.3.1 Dokumentationen von allgemeinem Interesse

Alexander Papaspyrou

Installationsanleitung für GT-4

http://www.c3grid.de/wiki/index.php/Installationsanleitung_f%C3%BCr_GT-4

Diese Kurzanleitung beschreibt die Installation und Grundkonfiguration von Globus Toolkit 4.x unter Linux. Sie dient als Ergänzung zu den Dokumenten auf den Webseiten der Globus Alliance sowie den Anleitungen des Leibniz-Rechenzentrums zur Installation und soll lediglich als Überblick dienen sowie typische Probleme herausstellen.

Konfigurationsanleitung für GT4-Anbindung an zentrale Indexdienste

http://www.c3grid.de/wiki/index.php/Konfigurationsanleitung_f%C3%BCr_GT4-Anbindung_an_zentrale_Indexdienste

Diese Kurzanleitung beschreibt die Anbindung an einen zentralen Indexdienst nach erfolgter Globus Installation.

V. Winkelmann, V. Achter

WebMDS unter Tomcat 5.5 und Java SDK 1.5

http://www.c3grid.de/wiki/index.php/WebMDS_unter_Tomcat_5.5_und_Java_SDK_1.5

Diese Kurzanleitung beschreibt, was zu beachten ist, wenn der WebMDS unter Tomcat 5.5 und Java SDK 1.5 eingerichtet werden soll. MDS ist der Monitoring und Discovering Service von Globus.

8.3.2 Dokumentationen bezüglich C3Grid

C3Grid Data Archive Reference Implementation

http:

[//www.c3grid.de/wiki/index.php/C3Grid_Data_Archive_Reference_Implementation](http://www.c3grid.de/wiki/index.php/C3Grid_Data_Archive_Reference_Implementation)

Diese Dokumentation beschreibt, wie die Anbindung eines lokalen Archivs an die Grid-Umgebung vereinfacht werden kann.

Specification of C3Grid Resource Descriptions

http:

[//www.c3grid.de/wiki/index.php/Specification_of_C3Grid_Resource_Descriptions](http://www.c3grid.de/wiki/index.php/Specification_of_C3Grid_Resource_Descriptions)

Enthalten ist ein XML-Schema für die einheitliche Beschreibung von Compute- und Storage-Ressourcen im C3Grid samt Dokumentation. Hauptnutzen einer solchen Beschreibung ist die automatische Erkennung von Ressourcen durch Grid-Middleware und Portal.

WorkflowBeschreibung

<http://www.c3grid.de/wiki/index.php/WorkflowBeschreibung>

Diese Seite enthält Informationen zur Erstellung von Workflowbeschreibungen. Über Links kann auf Subdokumente zugegriffen werden. Der wichtigste dieser Links zeigt auf das folgende Dokument.

Christian Grimme, Alexander Papaspyrou

Dokumentation der Workflow Specification Language (WSL)

<http://www.c3grid.de/wiki/images/9/9b/C3GridWSLDokuG2.pdf>

Diese Dokumentation gibt zuerst einen kurzen Einblick in die Architektur des Workflowschedulingdienstes und erklärt anschließend den Aufbau und die Benutzung der WSL für die Beschreibung eines C3Grid-Workflows. Der C3Grid-Dienst zum Management und zur Planung von Workflows basiert auf einem Service-orientierten Ansatz. Dabei wird ein einzelner Workflow als Kette atomarer aber durch Abhängigkeiten verbundener Einzelaufgaben (Tasks) betrachtet. Diese Sichtweise spiegelt sich auch in der Architektur des Schedulingdienstes wider. So wird jeder einzelne Workflow von einem eigenständigen Dienst überwacht und ebenso jeder einzelne Task des Workflows.

Scheduling von Workflows

http://www.c3grid.de/wiki/index.php/Scheduling_von_Workflows

Die Seite enthält einige zum Verständnis von Scheduling und Workflow nützliche Definitionen sowie ein Anwendungsbeispiel (erweiterte Stormtrack-Anwendung) für die Workflow Specification Language (WSL).

Environment für Workflow-Module (Hinweise für WF-Module-Provider)

[http://www.c3grid.de/wiki/index.php/Environment_f%C3%BCr_Workflow-Module_\(Hinweise_f%C3%BCr_WF-Module-Provider\)](http://www.c3grid.de/wiki/index.php/Environment_f%C3%BCr_Workflow-Module_(Hinweise_f%C3%BCr_WF-Module-Provider))

Anbieter von C3Grid-Workflow-Modulen stehen vor dem Problem, ihre Module portabel und unabhängig von Site-spezifischen Gegebenheiten des Computerressourcenproviders gestalten zu müssen, obwohl Site-spezifische Tools (wie CDOs, NetCDF-Libraries etc.) eingesetzt werden oder andere lokale Informationen benötigt werden. In dieser Dokumentation wird das Paket "Environment Modules" vorgestellt. "Environment Modules" ist ein Werkzeug zur einfachen Verwaltung einer Software-Umgebung (PATH, MANPATH, etc.). Im C3Grid soll "Environment Modules" den Entwicklern von Workflow-Modulen dazu dienen, ihre Module unabhängig von lokalen Eigenheiten benötigter Tools zu gestalten. Der Entwickler muss nicht mehr wissen, wo ein benötigtes Tool installiert ist, welche zusätzlichen Shell-Variablen notwendig sind, oder welche Libraries benötigt werden. Nur noch der standardisierte Toolname ist erforderlich.

Environment für Workflow-Module - Toolnamen und Shell-Variablen

http://www.c3grid.de/wiki/index.php/Environment_f%C3%BCr_Workflow-Module_-_Toolnamen_und_Shell-Variablen

Um für Entwickler von Workflow-Modulen die unabhängige Bereitstellung von Tools zu gewährleisten, müssen die Bezeichnungen für die eingesetzten Tools C3Grid-übergreifend vereinbart werden. Dies gilt auch für die Art der Versionsnummerierung sowie die zur Verfügung stehenden Environment-Variablen. Die Dokumentation gibt Auskunft über die im C3Grid zur Verfügung stehenden Bezeichner für Shell-Variablen und Tools im Environment.

Stefan Petri

How to set up a C3-Grid Data Provider site?

<http://www.c3grid.de/fileadmin/c3outreach/generation-0/howto-setup-dataprovider.pdf>

In diesem Howto wird beschrieben, wie der Anschluss an das C3-Datengrid durch Installation von Interfaces erfolgen kann. Das Howto ist öffentlich zugänglich.

8.3.3 Ausgewählte Publikationen

B. Fritsch, W. Hiller: *Das Collaborative Climate Community Data and Processing Grid (C3Grid) Gridtechnologien für die Erdsystemforschung* in Strobl/ Blaschke/ Griesebner (Hrsg.), *Angewandte Geoinformatik 2006*, Wichmann Verlag, 2006.

W. Gentzsch, A. Reinefeld (eds.) *Special Section on D-Grid*, *Future Generation Computer Systems* 25 (2009), (DOI: 10.1016/j.future.2008.09.008).

S. Plantikow, K. Peter, M. Höggqvist, C. Grimme, A. Papaspyrou *Generalizing the Data Management of Three Community Grids*, Future Generation Computer Systems (FGCS), Volume 25, Issue 3, Pages 281-289, March, 2009.

C. Grimme, T. Langhammer, A. Papaspyrou, F. Schintke: *Negotiation-based Choreography of Data-intensive Applications in the C3Grid Project*, German e-Science Conference 2007 Baden-Baden, May 2007.

U. Schindler, B. Bräuer, M. Diepenbroek: *Data Information Service based on Open Archives Initiative Protocols and Apache Lucene*, German e-Science Conference 2007 Baden-Baden, May 2007

S. Kindermann: *Climate Data Analysis and Grid Infrastructures: Experiences and Perspectives*, Grid-Enabling Legacy Applications and Supporting End Users Workshop (GELA), Paris, France, 20 June 2006: within the framework of the 15th IEEE International Symposium on High Performance Distributed Computing

S. Kindermann: *Klimadaten und Grid-Infrastrukturen*, Jahrbuch der Max-Planck-Gesellschaft 2006.

S. Kindermann, K. Ronneberger: *Grid technology projects at DKRZ TerraFLOPS*, Newsletter of DKRZ and M&D. Ausgabe April 2006.

8.3.4 Web-Seiten

C3Grid Homepage: <http://www.c3grid.de/>

Publikationen: <http://www.c3grid.de/index.php?id=54>

Deliverables: <http://www.c3grid.de/index.php?id=52>

8.4 TextGrid

Alle im Rahmen von TextGrid entstandenen Berichte und Publikationen sind auf der Internetseite verzeichnet:

<http://www.textgrid.de/berichte.html>

Hervorzuheben sind insbesondere folgende weiterführende Dokumente:

- Öffentliche Fassung des Projektantrags für die zweite Projektphase (Juni 2009 bis Mai 2012):
http://www.textgrid.de/fileadmin/TextGrid/div/090804_Nachtrag_oeffentlich.pdf
- Informationen für Kooperationspartner:
<http://www.textgrid.de/ueber-textgrid/kooperationen.html>
- Tutorial für Entwickler (wird laufend aktualisiert):
http://www.textgrid.de/fileadmin/TextGrid/reports/developers_tutorial_090803.pdf

8.4.1 Dokumente von allgemeinem Interesse

R3.1. Bericht über Evaluation der vorhandenen Grid-Middleware-Standards und Software-Pakete unter Berücksichtigung der geplanten Dienste der Integrationsplattform und der ermittelten Anforderungen

http://www.textgrid.de/fileadmin/TextGrid/reports/TextGrid_Report_3_1.pdf

R3.2. Spezifikation der Architektur für die TextGrid-Middleware (TextGrid-Architektur)

http://www.textgrid.de/fileadmin/TextGrid/reports/TextGrid_Report_3_2.pdf

R3.3. Spezifikation aller von der TextGrid-Middlewareplattform zu bedienenden Grid-Schnittstellen (Version 1) (wird fortlaufend aktualisiert – verfügbar auf Anfrage)

R3.4. Middleware-Tests, unter Berücksichtigung der TextGrid-Werkzeuge und der TextGrid-Musterapplikationen (wird fortlaufend aktualisiert – verfügbar auf Anfrage)

R3.5. User-Manual zur Anbindung von Werkzeugen an die TextGrid-Middleware (TextGrid Manual – Tool Development)

http://www.textgrid.de/fileadmin/TextGrid/reports/R3_5-manual-tools.pdf

R3.6. User-Manual zur Installation eines Datengridknotens (für assoziierte Textarchive)

http://www.textgrid.de/fileadmin/TextGrid/reports/TextGrid_Report_3_6-Datengrid-Knoten.pdf

8.4.2 Dokumentation bezüglich TextGrid

Publikationen in wiss. Fachorganen

Aschenbrenner, Andreas / Meffert, Katja: *Wissenschaftliche Infrastruktur in den Geisteswissenschaften? – Eine Wegbeschreibung*. In: Georf Braungart, Peter Gendolla, Fotis Jannidis (Hg.): *Jahrbuch für Computerphilologie – online*, 9.8.2008 (Hg.). <http://computerphilologie.tu-darmstadt.de/jg07/aschmeff.html>

Aschenbrenner, Andreas: *Feature – Editing, analyzing, annotating, publishing: TextGrid takes the a, b, c to D-Grid*. In: *iSGTW* 30 January 2008, Jg. 54. <http://www.isgtw.org/?pid=1000828>

Rapp, Andrea: *Das Projekt „TextGrid. Modulare Plattform für verteilte und kooperative wissenschaftliche Textdatenverarbeitung – ein Community-Grid für die Geisteswissenschaften“*. *Chancen und Perspektiven für eine neue Wissenschaftskultur in den Geisteswissenschaften*. In: *Jahrbuch der historischen Forschung in der Bundesrepublik Deutschland: Berichtsjahr 2006 / hrsg. von der Arbeitsgemeinschaft historischer Forschungseinrichtungen in der Bundesrepublik Deutschland*. München: Oldenbourg, 2007, S. 61-68. http://www.ahf-muenchen.de/Forschungsberichte/Jahrbuch2006/AHF_Jb2006_FB_B1_Rapp.pdf

Neuroth, Heike / Aschenbrenner, Andreas / Lohmeier, Felix: *e-Humanities – eine virtuelle Forschungsumgebung für die Geistes-, Kultur- und Sozialwissenschaften*. In: *Bibliothek. Forschung und Praxis*, 3 (2007), S. 272-279. http://www.bibliothek-saur.de/2007_3/272-279.pdf

„TextGrid: Ein Community Grid für die Geisteswissenschaften“. In: Heike Neuroth, Martina Kerzel, Wolfgang Gentsch (Hg.): *Die D-Grid Initiative*. Göttingen: Universitätsverlag, 2007. S. 64-66.

<http://www.univerlag.uni-goettingen.de/content/list.php?notback=1&details=isbn-978-3-940344-01>

Küster, Marc Wilhelm / Ludwig, Christoph / Aschenbrenner, Andreas: *TextGrid as a Digital Ecosystem*. IEEE DEST 2007, 21.-23. Februar, Cairns, Australien, SPECIAL SESSION 3: e-Humanities for Digital Eco-systems: A Social, Cultural, Economic and Political Agenda. http://www.textgrid.de/fileadmin/TextGrid/veroeffentlichungen/DigitalEcosystem07_CameraReady-1.pdf

Aschenbrenner, Andreas / Blanke, Tobias / Dunn, Stuart / Kerzel, Martina / Rapp, Andrea / Zielinski, Andrea: *Von e-Science zu e-Humanities – Digital vernetzte Wissenschaft als neuer Arbeits- und Kreativbereich für Kunst und Kultur*. In: Bibliothek. Forschung und Praxis, 1 (2007), S. 11-21. http://www.bibliothek-saur.de/2007_1/011-021.pdf

Aschenbrenner, Andreas / Gietz, Peter / Küster, Marc Wilhelm / Ludwig, Christoph / Neuroth, Heike: *TextGrid – a modular platform for collaborative textual editing*. In: Proceedings of the International Workshop on Digital Library Goes e-Science (DLSci06), September 21, 2006, Alicante, Spain. S. 27-36. http://www.textgrid.de/fileadmin/TextGrid/veroeffentlichungen/ecdl06_textgrid.pdf

Büdenbender, Stefan / Leuk, Michael: *Daten als Dienste: Wörterbücher als Erschließungsinstrumente in der virtuellen Arbeitsumgebung "TextGrid"*. *Data as services: dictionaries as a means of accessing texts in the virtual research environment "TextGrid"*, in: it – Information Technology, Heft 4 (August) 2009, S. 191-196, Themenheft Informatik in den Geisteswissenschaften (Gastherausgeber: Moulin, Claudine / Burch, Thomas / Rapp, Andrea).

Küster, Marc Wilhelm / Ludwig, Christoph / Aschenbrenner, Andreas: *TextGrid: eScholarship und vernetzte Angebote*, in: it – Information Technology, Heft 4 (August) 2009, S. 183-190, Themenheft Informatik in den Geisteswissenschaften (Gastherausgeber: Moulin, Claudine / Burch, Thomas / Rapp, Andrea).

Öffentliche Reports und Dokumentationen (alle zugänglich über www.textgrid.de/berichte.html)

TextGrid-Szenarien / Use Cases

http://www.textgrid.de/fileadmin/TextGrid/reports/TextGrid-Szenarien_061212.pdf

TextGrid-Kerncodierung

http://www.textgrid.de/fileadmin/TextGrid/reports/TextGrid_Kerncodierung070615.pdf (Überblick)

http://www.textgrid.de/fileadmin/TextGrid/reports/Textsortenspezifische_Kerncodierung_2009.pdf (Ausführliche Dokumentation)

R1.1. Text Processing

http://www.textgrid.de/fileadmin/TextGrid/reports/TextGrid_Report_1_1.pdf

R1.2. Linking

http://www.textgrid.de/fileadmin/TextGrid/reports/TextGrid_Report_1_2.pdf

R1.3. Text Retrieval

http://www.textgrid.de/fileadmin/TextGrid/reports/TextGrid_Report_1_3.pdf

R1.4. Publishing

http://www.textgrid.de/fileadmin/TextGrid/reports/TextGrid-R1_4_Publishing.pdf

R1.5. Management von Workflow, Access, Kommunikation und Nutzer

http://www.textgrid.de/fileadmin/TextGrid/reports/TextGrid_Report_1_5.pdf

R1.6. Ontologien

http://www.textgrid.de/fileadmin/TextGrid/reports/TextGrid_Report_1_6_Ontologien.pdf

R2.1. TextGrid Tools I

http://www.textgrid.de/fileadmin/TextGrid/reports/TextGrid_Report_2_1.pdf

R2.2. Dokumentation TextGrid Tools II

http://www.textgrid.de/fileadmin/TextGrid/reports/TextGrid-R2.2_ToolsII.pdf

R5.1. Reports über Ontologie-Software (Ontologien und Wortnetze)

http://www.textgrid.de/fileadmin/TextGrid/reports/TextGrid_Report_5_1.pdf

R5.2. Report über Tests mit philologisch-linguistischen Verfahren zur Erstellung der Meta - Lemmaliste

http://www.textgrid.de/fileadmin/TextGrid/reports/TextGrid_R5_1.pdf

R6.1. Konzept für die Öffentlichkeitsarbeit

http://www.textgrid.de/fileadmin/TextGrid/reports/TextGrid_R6_1.pdf

8.5 MediGrid

MediGRID verwendet für die Installation der Ressourcen die DGI Referenzinstallation. Spezielle Anpassungen, wie die Konfigurationen des Globus Monitoring and Discovery Systems (MDS) oder des Ganga-Monitorings, werden in dem internen MediGRID-Wiki beschrieben.

8.5.1 Dokumentation von allgemeinem Interesse

In 2008 und 2009 wurden in MediGRID Entwickler-Workshops durchgeführt. Nach einer Einführung in die MediGRID-Landschaft haben Spezialisten auf den Gebieten Globus Toolkit, Datenmanagement, Workflow-Management, Portal-Entwicklung und Visualisierung ihr Wissen an die Anwendungsentwickler in MediGRID durch Vorträge und viele Übungen weitergegeben. Der Entwickler-Workshop 2009 wurde durch das Projekt SuGI begleitet und es befinden sich sämtliche Vorträge (als Folien und Flash-Video) im SuGI-Portal[12].

8.5.2 Dokumentation bezüglich MediGRID

Das Dokument "Guidelines for the Deployment of Applications in MediGRID"[11] beschreibt detailliert die Integration von Anwendungen in MediGRID. In Abbildung 7 ist beispielhaft die Anwendungsintegration von Command Line Tools in MediGRID dargestellt. Der Prozess umfasst die erforderlichen Schritte von der entwickelten Anwendung bis zu ihrer Veröffentlichung.

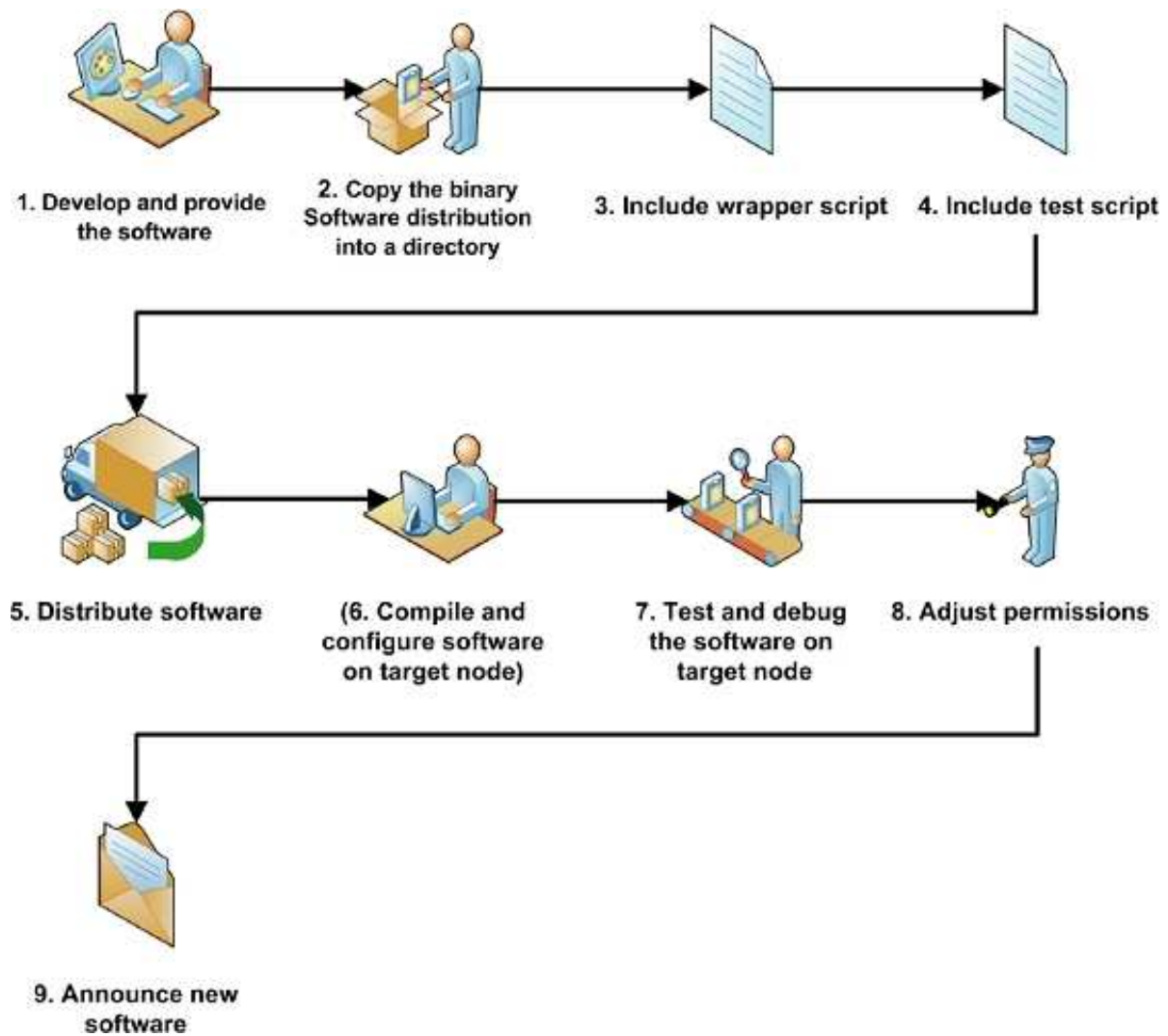


Abbildung 7: Der Prozess der Anwendungsintegration von Command Line Tools in MediGRID.

Literatur

- [1] AstroGrid-D Website, <http://www.astrogrid-d.org>
- [2] Globus Administration Configuration and Standard Installation, <http://www.astrogrid-d.org/project-products/grid-support.html>
- [3] Globus-Helper Package, http://www.astrogrid-d.org/project-products/Software/Globus-Helper_en.html
- [4] Application2Grid-Guide, <http://www.astrogrid-d.org/documents/Posters/App2Grid-Guide-AstroGrid.pdf>
- [5] First steps in AstroGrid-D, <http://www.ari.uni-heidelberg.de/mitarbeiter/bruesemeister/agdguide/book1.htm>
- [6] AstroGrid-D Data Management Tutorial <http://www.astrogrid-d.org/project-products/Software/adm/adm-tutorial.pdf>
- [7] Installing Geomaint <http://mintaka.aip.de:8080/lenya/intranet/live/workpackages/wg1/Geomaint.html>
- [8] <http://www.astrogrid-d.org/project-documents/UseCases.html>
- [9] TextGrid Website <http://www.textgrid.de>
- [10] MediGrid Website <http://www.medigrid.de>
- [11] Hoheisel, A; Sommerfeld, D; Peter, K: Guidelines for the Deployment of Applications in MediGRID, Version 1.4
- [12] Entwickler-Workshop 2009 im SuGI-Portal http://sugi.d-grid.de/de/veranstaltungen/liste-der-beitraege.html?tx_sugi_pi3%5Bcurrentpage%5D=1&tx_sugi_pi3%5Bsearchtype%5D=event&tx_sugi_pi3%5Bid%5D=21&cHash=f2d449cd5e
- [13] C3Grid Website <http://www.c3grid.de>
- [14] Der C3Grid-Nutzerworkshop 2009 wurde von SuGI aufgezeichnet und ist über die SuGI-Plattform verfügbar:
 - 1) Bernadette Fritzsich: Collaborative Climate Community Data and Processing Grid - C3Grid, [http://sugi.d-grid.de/de/schulungsinhalte/details.html?tx_sugi_pi2\[contentid\]=395&cHash=5b8bafb819](http://sugi.d-grid.de/de/schulungsinhalte/details.html?tx_sugi_pi2[contentid]=395&cHash=5b8bafb819)
 - 2) Benny Bräuer: Vorstellung des C3-Portals, [http://sugi.d-grid.de/de/schulungsinhalte/details.html?tx_sugi_pi2\[contentid\]=386&cHash=87c149012c](http://sugi.d-grid.de/de/schulungsinhalte/details.html?tx_sugi_pi2[contentid]=386&cHash=87c149012c)
 - 3) Christian Grimme und Stefan Plantikow: Technische Aspekte von C3Grid, [http://sugi.d-grid.de/de/schulungsinhalte/details.html?tx_sugi_pi2\[contentid\]=389&cHash=2770309cf0](http://sugi.d-grid.de/de/schulungsinhalte/details.html?tx_sugi_pi2[contentid]=389&cHash=2770309cf0)
 - 4) Uwe Ulbrich: Workflows, [http://sugi.d-grid.de/de/schulungsinhalte/details.html?tx_sugi_pi2\[contentid\]=432&cHash=03abe6ee91](http://sugi.d-grid.de/de/schulungsinhalte/details.html?tx_sugi_pi2[contentid]=432&cHash=03abe6ee91)
 - 5) Bernadette Fritzsich: Ergebnisse und Erfahrungen eines Community-Projekts, [http://sugi.d-grid.de/de/schulungsinhalte/details.html?tx_sugi_pi2\[contentid\]=302&cHash=993a73cf4e](http://sugi.d-grid.de/de/schulungsinhalte/details.html?tx_sugi_pi2[contentid]=302&cHash=993a73cf4e)

Zum Thema Workflow gibt es auch eine Video-Demonstration:

Volker Winkelmann: C3Grid Workflow, <http://sugi.d-grid.de/de/grid-computing/anwendungsbeispiele/c3grid/workflow-demo.html>

Außerdem gibt es das folgende Vortragsvideo.

Siegfried Makedanz, Jörg Matthes und Hans Pfeiffenberger: Authentifizierung und Autorisierung in einem Portal-basierten Grid (C3-Grid), http://sugi.d-grid.de//fileadmin/user_upload/sugi_storage/0000/4/IVOM_05_Makedanz/t001/index.htm

[15] HEP Grid <http://www.dgrid.de/index.php?id=44>