



omii europe

open middleware infrastructure institute



EU project: RI031844-OMII-Europe

Project no: **RI031844-OMII-Europe**

Project acronym: **OMII-Europe**

Project title: **Open Middleware Infrastructure Institute for Europe**

Instrument: **Integrated Infrastructure Initiative**

Thematic Priority: **Communication network development**

D:JRA2.0 Report on Grid Activities relevant to the identification of new services

Due date of deliverable: 31 October 2006

Actual submission date:

Start date of project: **1 May 2006**

Duration: **2 years**

INFN

Revision [RC3]

Project co-funded by the European Commission within the Sixth Framework Programme (2002-2006)		
Dissemination Level		
PU	Public	X
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Document Control Sheet

Document	Title: Report on Grid Activities relevant to the identification of new services	
	ID: D:JRA2.0	
	Version: 1.0 RC3	Status: Final
	Available at: http://omii-europe.org	
	Software Tool: Microsoft Word 2002 SP3	
	File(s): DJRA2.0.doc and DJRA2.0.pdf	
Authorship	Edited by:	Sergio Andreozzi (INFN)
	Written by:	Sergio Andreozzi (INFN), Antonia Ghiselli (INFN), Chunming Hu (BU), Jinlei Jiang (TU), Balazs Konya (Lund University), Morris Riedel (FZJ), Davy Virdee (UDEIN), Li Zha (ICT)
	Contributors:	INFN, FZJ, TU, ICT, BU, UEDIN
	Reviewed by:	Project Management Committee
	Approved by:	Project Management Committee

Document Status Sheet

Version	Date	Status	Comments
0.10	23 October 2006	Draft	Version sent to the internal reviewers
0.11	26 October 2006	Draft	Reviewed by DV – made some minor corrections.
1.0 RC1	8 November 2006	Release Candidate	
1.0 RC2	14 November 2006	Release Candidate	Reviewed by DV with minor corrections
1.0 RC3	24 November 2006	Final	Changes based on reviews

Executive Summary

The JRA2 activity is targeted at the identification of appropriate emerging middleware services from the global Grid initiative that are not already considered by the OMII-Europe. The purpose is the definition of priorities for the placement of such services in the OMII-Europe repository.

This document is the first deliverable of the JRA2 activity that is planned to be revised at month 12 and month 18. The purpose of this document is the analysis, identification and prioritization of the services that should be included in the OMII-Europe repository through a re-engineering process towards the adoption of community standards. The re-engineering activity for the porting of the identified services will be performed via a subcontract of 100,000 Euros to be used during the second year of the project; therefore it should be sized to such an effort.

The analysis and comparison described in this document started by considering the Grid platforms that contribute to the OMII-Europe project namely: gLite, Globus, UNICORE, VEGA-GOS, CROWNGRID and OMII-UK. Thanks to an external contribution, the ARC Grid platform was included.

The comparison has been performed based on the consideration that third generation Grid systems are based on community standards. Therefore, the first phase has been the study of the Open Grid Service Architecture (OGSA) specification and the creation of a hierarchy of capabilities that each Grid middleware should provide (see Section 2). The second phase has been the mapping of each of the selected platforms into each of the identified capabilities (see Section 3). The third phase has been the comparison of the various platforms based on each elementary capability (see Section 4). This comparison is summarized in Table 10 Table 10 Middleware comparison by capabilities and a descriptive part is also provided for those capabilities not yet covered by the OMII-Europe. The last phase has been the evaluation and selection of Grid capabilities to be covered by the subcontract. This selection has been based on a Risk-Cost-Value model (Section 5).

Given the results of our analysis, we recommend that the subcontract of 100,000 Euros should be invested on the following capabilities given in decreasing preference order: information modeling, security.authorization, data.storage.management and data.managment.transfer.

We also recommend that the JRA3 Security should analyze how the OMII-related Grid platforms address the following aspects: (1) authentication, there are common building blocks, nevertheless it should be verify the effectiveness of compatibility; (2) delegation, UNICORE has a different approach than the others, therefore an interoperable solution should be found; (3) authorization, this is the top-level part of the work; a common and interoperable solution among the different solutions should be evaluated; the inclusion of an authorization service as an activity of JRA2 should be planned only if synergies with JRA3-security are possible, that is if JRA3-security considers as a key activity the authorization aspect.

Table of Contents

1	Introduction.....	6
2	Approach.....	7
3	Platforms Descriptions.....	11
4	Analysis.....	20
5	Cost-Risk-Value Based Selection.....	26
6	Conclusions.....	27
A	Description of the Grid Platforms.....	28
B	Acknowledgements.....	56
C	References.....	56

List of Tables

Table 1 Template for Grid platform description.....	8
Table 2 gLite decomposition.....	11
Table 3 GLOBUS decomposition.....	12
Table 4 CROWN decomposition.....	13
Table 5 VEGA-GOS decomposition.....	14
Table 6 UNICORE 5 decomposition.....	15
Table 7 UNICORE 6 decomposition.....	16
Table 8 OMII-UK decomposition.....	17
Table 9 ARC decomposition.....	18
Table 10 Middleware comparison by capabilities.....	20
Table 11 Cost-Risk-Value comparison.....	26

1 Introduction

The main purpose of the Open Middleware Infrastructure Institute for Europe (OMII-Europe) is the provision of key software components for building e-Infrastructures within the European Research Area (ERA) [TA]. The activity aims at facilitating the development and porting of a common set of identified application level services to a number of major Grid software distributions, and further to develop tighter interoperability between different Grid distributions. The main strengths of the project are interoperability among a set of relevant Grid middleware platforms via the adoption of emerging standards and usability. If standards are not yet mature or are lacking, the project can contribute via the participation in the appropriate forums and with early implementation across multiple Grid distributions.

In this context, the activity of Joint Research Activity (JRA) 2 aims at identifying appropriate emerging middleware services from the global Grid initiative that are not part of the initial plans [TA]; furthermore it aims at defining priorities for the placement of such services in the OMII-Europe repository. Such a placement activity can be preceded by a re-engineering activity targeted at the adoption of a community-based standard. For this activity, a subcontract of 100,000 Euros is available and can be allocated for the second year of the project.

This document is the first deliverable of the JRA2 activity that is planned to be revised at month 12 and month 18. The purpose of this document is the analysis, identification and prioritization of the services that should be included in the OMII-Europe repository through a re-engineering process towards the adoption of community standards. The analysis and comparison started considering the Grid platforms that contribute to the OMII-Europe project, namely gLite, Globus, UNICORE, VEGA-GOS, CROWNGRID and OMII-UK. Thanks to an external contribution, also the ARC Grid platform was included.

The comparison has been performed based on the consideration that third generation Grid systems are based on community standards. Therefore, the first phase has been the study of the Open Grid Service Architecture (OGSA) specification and the creation of a hierarchy of capabilities that each Grid middleware should provide (see Section 2). The second phase has been the mapping of each of the selected platforms into each of the identified capabilities (see Section 3 for a summary view and Appendix A for a more detailed description). The third phase has been the comparison of the various platforms based on each elementary capability (see Section 4). The last phase has been the evaluation and selection of Grid capabilities to be covered by the subcontract. This selection has been based on a Risk-Cost-Value model (see Section 5). Final conclusions are given in Section 6.

2 Approach

The analysis process for the selection of which missing Grid capabilities for a Grid middleware based on the content of the OMII-Europe repository is a task that should consider several constraints. First, the selected capabilities should be re-engineered towards the adoption of a community standard in order to enable interoperability and wide acceptance. Second, the selected capabilities should encounter a large interest for their inclusion. Third, the re-engineering and inclusion process must be achievable in a one-year timeframe with a subcontract of 100K Euro. Fourth, the analysis, selection and motivation should be performed by Month 6, that is by this deliverable.

Considering these constraints, we have decided to perform the analysis and selection activity as follows. First of all, we have limited the analysis to the Grid platforms that are represented in the OMII-Europe project because a direct knowledge was available. This means that the following Grid middlewares have been investigated: gLite, Globus, CrownGrid, Vega-GOS, OMII-UK and UNICORE. External contributions of descriptions of other middlewares were possible if provided by extra effort. We received a description of the ARC middleware.

The comparison has been performed based on the consideration that third generation Grid systems are based on community standards. In particular the architecture of the various middlewares is converging towards the Open Grid Service Architecture (OGSA) [OGSA]. The OGSA is a conceptual architecture defined by the Open Grid Forum that identifies a set of capabilities that a Grid middleware should provide in order to implement the Grid paradigm. This conceptual architecture is service-oriented [SOARM] and builds on top of the Web Services Architecture [WSA] plus a set of extensions (e.g., Web Services Resource Framework [WSRF]).

Following this consideration and keeping in mind that not all selected platforms can be considered fully service-oriented, we have decided to focus the comparison based on the OGSA capabilities. The term *capability* is defined in [OGSAGLOS] as “*a set of one or more services that together provide a function that is useful in a Grid context*”. In the same document, the term *service* is defined as “*a software component participating in a service-oriented architecture that provides functionality and/or participates in realizing one or more capabilities*”. Both sentences create a circular definition that should be resolved in a future revision of the document. For the purpose of our document, we define capability as the “capacity to be used for a specific purpose”. Following this definition, a Grid middleware provides a certain capability if the middleware is able to be used for the purpose identified by the capability description.

By considering these aspects, we have identified a hierarchical decomposition of the OGSA capabilities and we have constructed a table enabling to decompose a particular Grid platform based on these capabilities (see Table 1). Such a table acts as a template for platform description and is composed by two main parts: a summary section and a capability section. The former provides general information about the platform being analyzed (e.g., name, last stable version and licence), while the latter contains the list of the supported capabilities with reference to the subsystem name, the development status (planning, alpha, beta, production, mature, inactive), the adopted standards, and the standards that are foreseen for future adoption.

Table 1 Template for Grid platform description

Name	Name of the middleware
Last Stable Version	Version of the last stable release
Dissemination Website	URL of the dissemination website
Source Code Repository	URL of the code repository
Reference Document Architecture	URL of the most relevant architectural document
Licence	Software Licence used to release the code
Deployment Size	Number of administrative institutions that install and manage a site using this middleware

	Subsystem Name	Development Status	Adopted Standard	Foreseen Standard
Security.Authentication				
Security.CredentialStorage				
Security.Delegation				
Security.Authorization				
Security.AttributeAuthority				
Security.IdentityMapping				
Security.Accounting				
Data.Transfer				
Data.Management.Transfer				
Data.Management.Replica				
Data.Management.Storage				
Data.Naming.Resolver				
Data.Naming.Scheme				
Data.Access.Relational				
Data.Access.XML				
Data.Access.FlatFiles				
Information.Model				
Information.Discovery				
Information.Logging				
Information.Monitoring				
Information.Provenance				
ExecMan.BES				
ExecMan.JobDescription				
ExecMan.JobManager				
ExecMan.ExecutionAndPlanning				
ExecMan.CandidateSetGenerator				
ExecMan.Reservation				

In the remaining part of this section, you can find a synthetic description of each capability.

Security

Security.Authentication

This capability is related to the capacity of providing authentication mechanisms for Grid users, machine and services.

Security.CredentialStorage

This capability is related to the capacity of providing an online credential repository that allows users to securely obtain credentials when and where needed.

Security.Delegation

This capability is related to the capacity for a user to give a service the authority to undertake specific activities or decisions on its behalf.

Security.Authorization

This capability is related to the capacity of handling authorization aspects, making authorization decisions about the subject and the requested mode of access based upon combining information from a number of distinct sources.

Security.AttributeAuthority

This capability is related to the capacity of associating a user with a set of attributes in a trusted manner to a relying party, by way of digitally signed assertions.

Security.IdentityMapping

This capability is related to the capacity of mapping Grid-level credentials to local level credentials (e.g., mapping a user X.509 certificate into a UNIX account).

Security.Accounting

This capability is related to the capacity of systematically recording, reporting, and analyzing the usage of resources.

Data***Data.Transfer***

This capability is related to the capacity of moving a file from one network location to another. It refers to the actual transfer (e.g., as performed by protocols like FTP, GridFTP, or HTTP).

Data.Management.Transfer

This capability is related to the capacity of managing a transfer of files from the start to the completion.

Data.Management.Replica

This capability is related to the capacity of managing the creation of file replicas upon request.

Data.Management.Storage

This capability is related to the capacity of managing a storage resource, from simple systems like disk-servers to complex hierarchical systems.

Data.Naming.Resolver

This capability is related to the capacity of resolving one name to another (for example, search the associated abstract name to a certain human-oriented name).

Data.Naming.Scheme

This capability is related to the capacity of attaching names to data resources. (To evaluate if it should moved to the main category infrastructure instead of data). In OGSA, a three-level naming scheme is defined: (1) human-oriented name, (2) abstract name and (3) address.

Data.Access.Relational

This capability is related to the capacity of providing access to a relational data source.

Data.Access.XML

This capability is related to the capacity of providing access to an XML data source.

Data.Access.FlatFiles

This capability is related to the capacity of providing access to a flat file.

Information

Information.Model

This capability is related to the capacity of modelling resources based on a community accepted definition.

Information.Discovery

This capability is related to the capacity of locating unknown resources or services, possibly satisfying a set of requirements.

Information.Logging

This capability is related to the capacity of recording data, often chronologically.

Information.Monitoring

This capability is related to the capacity of periodically observing measurements, transform them and make available to users or other applications.

Information.Provenance

This capability is related to the capacity of providing long-term storage of information related to Grid activity and to let this information be accessed by users or other applications.

ExecMan

ExecMan stands for Execution Management.

ExecMan.BES

This capability is related to executing a job or set of jobs.

ExecMan.JobDescription

This capability is related to the capacity of letting users be able to describe a job submission request based on a machine-processable language.

ExecMan.JobManager

This capability is related to the capacity of managing the execution of a job or set of jobs from start to finish.

ExecMan.ExecutionAndPlanning

This capability is related to the capacity of building schedules for jobs, that is, the capability of defining mappings between services and resources, possibly with time constraints.

ExecMan.CandidateSetGenerator

This capability is related to the capacity of determining the set of resources on which a nit of work can execute.

ExecMan.Reservation

This capability is related to the capacity of managing reservation of resources for future usage.

3 Platforms Descriptions

3.1 gLite

The gLite middleware is developed in the framework of the European Grid for E-science (EGEE) project [EGEE]. The relevant document describing its architecture is [EGEEARCH].

Table 2 gLite decomposition

Name	gLite
Last Stable Version	3.0.0
Dissemination Website	http://glite.web.cern.ch/glite/
Source Code Repository	http://jralmw.cvs.cern.ch:8180/cgi-bin/jralmw.cgi/
Reference Document Architecture	https://edms.cern.ch/document/594698/1.0/
Licence	Apache 2.0 (soon)

	Subsystem Name	Development Status	Adopted Standard	Foreseen Standard
Security.Authentication	GT GSI	mature	IETF RFC3820 , ITU-T X.509	
Security.CredentialStorage	MyProxy	mature	IETF RFC3820 , ITU-T X.509	
Security.Delegation	GT GSI	mature	IETF RFC3820	
Security.AttributeAuthority	VOMS	production		
Security.Authorization	G-PBox	beta	XACML 1.1	XACML 1.1, SAML 2
Security.Authorization	gJAF	beta	XACML 1.1	SAML 2
Security.Authorization	GridSite	production		
Security.Authorization	LCAS	production		
Security.IdentityMapping	LCMAPS	production		
Security.Accounting	DGAS	production		OGF RUS/UR
Security.Accounting	APEL	production		
Data.Transfer	GridFTP		GridFTP	
Data.Management.Transfer	FTS	production		
Data.Management.Storage	DPM	production	SRM 2.1.1	SRM 2.2
Data.Management.Storage	StoRM	production	SRM 2.1.1	SRM 2.2
Data.Naming.Scheme	LFN,TURL,SURL	production		
Data.Naming.Resolver	LFC	production		
Information.Model	GLUE Schema	production		CIM
Information.Discovery	MDS 2.x + BDII	production		
Information.Discovery	R-GMA	Mature		
Information.Discovery	Service Discovery	production		
Information.Logging	Logging And Bookeeping	production		
Information.Monitoring	R-GMA	production		
Information.Monitoring	GridICE	production		
Information.Monitoring	CEMon	production	SOAP 1.2, WSDL 1.1	
Information.Provenance	Job Provenance	beta		
ExecMan.BES	LCG CE	production	GRAM	
ExecMan.BES	gLite-CE	production		
ExecMan.BES	CREAM	beta	OGSA-BES, JSDL	OGSA-BES , JSDL
ExecMan.JobDescription	JDL	production		JSDL
ExecMan.JobManager	WMS	production		
ExecMan.ExecutionAndPlanning	WMS	production		
ExecMan.CandidateSetGenerator	WMS	production		
ExecMan.Reservation	Advance Reservation	alpha	WS-Agreement	

3.2 Globus

Table 3 GLOBUS decomposition

Name	Globus Toolkit 4
Last Stable Version	4.0.3
Dissemination Website	http://www.globus.org/toolkit/
Source Code Repository	http://www.globus.org/toolkit/downloads/4.0.3/#source
Reference Document Architecture	http://www.globus.org/toolkit/docs/4.0/
Licence	http://www.globus.org/toolkit/legal/4.0/ Falls under “Apache Public Licence” and the Globus Toolkit Public Licence”, that is, “The Globus Alliance is committed to maintaining a liberal, open source licence. The Globus Toolkit Public Licence (GTPL) allows software to be used by anyone and for any purpose, without restriction. We believe that this is the best way to ensure that Grid technologies gain wide spread acceptance and benefit from a large developer community.”

	Subsystem Name	Development Status	Adopted Standard	Foreseen Standard
Security.Authentication	GSI	production	x.509, RFC 3820	
Security.CredentialStorage	MyProxy	mature	RFC3820 , X.509	
Security.Delegation	Delegation Service	mature	X.509, WS-Trust	
Security.Authorization	CAS	mature	SAML	
Security.AttributeAuthority				
Security.IdentityMapping				
Security.Accounting	SGAS	beta		
Data.Transfer	GridFTP,	production	GFD.020 RFC959, RFC2228, RFC2389	
Data.Management.Transfer	RFT	beta		
Data.Management.Replica	DRS	beta		
Data.Management.Storage				
Data.Access.Relational	OGSA-DAI	Production	OGSA-DAI	WS-DAI
Data.Access.XML	OGSA-DAI	Production	OGSA-DAI	WS-DAI
Data.Access.FlatFiles	XIO	Production		
Data.Naming.Scheme				
Data.Naming.Resolver	RLS	beta		
Information.Model	GLUE Schema 1.1			
Information.Discovery	MDS 4	mature		
Information.Logging				
Information.Monitoring	MDS 4	mature		
Information.Provenance				
ExecMan.BES	WS-GRAM			OGSA-BES
ExecMan.JobDescription	XML-based			
ExecMan.JobManager	CSF			
ExecMan.ExecutionAndPlanning	CSF			
ExecMan.CandidateSetGenerator	CSF			
ExecMan.Reservation				

3.3 CROWNGRID

Table 4 CROWN decomposition

Name	CROWN
Last Stable Version	2.5
Dissemination Website	http://www.crown.org.cn/en
Source Code Repository	not available
Reference Document Architecture	http://202.112.128.70/~leilei/CROWN-20060523.iso
Licence	

	Subsystem Name	Development Status	Adopted Standard	Foreseen Standard
Security.Authentication	CROWN Authz	Production	X.509 RFC3820	
Security.CredentialStorage	CROWN CredMan	Production	RFC3820 WS-Trust	
Security.Delegation	CROWN CredMan	Production	RFC3820	
Security. Authorization	CROWN Authz	Production	SAML, XACML1.1	
Security.AttributeAuthority	CROWN AA	beta	SAML	
Security.IdentyMapping	CROWN CredFed	Production	WS-Secure Conversation WS-Trust	
Security.Accounting	Part of CROWN NodeServer	Production		
Data.Transfer	LDS	Prototype	FTP	
Data.Management.Transfer	MDS	Prototype		
Data.Management.Replica				
Data.Management.Storage				
Data.Naming.Resolver				
Data.Naming.Scheme				
Data.Access.Relational	OGSA-DAI	Production		
Data.Access.XML				
Data.Access.FlatFiles				
Information.Model	crown gims service	Production		
Information.Discovery	crown rlds service, SClub service, Region Registry service, Region Switch service	Production		
Information.Logging	Information providers	production		
Information.Monitoring	CROWN monitoring and statistical services	production		
Information.Provenance	CROWN RLDS service	production		
ExecMan.BES	CROWN Scheduler	Production	JSDL/BES	
ExecMan.JobDescription	CROWN Scheduler	Production	JSDL	
ExecMan.JobManager	CROWN Scheduler	Production	BES	
ExecMan.ExecutionAndPlanning	CROWN Scheduler	Production		
ExecMan.CandidateSetGenerator	CROWN RLDS	Production		
ExecMan.Reservation	CROWN Scheduler/Node Server (FIRST)	prototype		

3.4 VEGA-GOS

Table 5 VEGA-GOS decomposition

Name	Vega-GOS
Last Stable Version	2.0
Dissemination Website	http://vega.ict.ac.cn/en/gosproject.jsp?id=dir30
Source Code Repository	
Reference Document Architecture	http://vega.ict.ac.cn/en/gosdownload.jsp?id=dir5
Licence	

	Subsystem Name	Development Status	Adopted Standard	Foreseen Standard
Security.Authentication	agora	Production	X.509	
Security.Delegation	grip	Production	IETF RFC3820	
Security.CredentialStorage	agora	Production	IETF RFC3820	
Security.AuttributeAuthority	agora	Production	SAML 1.0	
Security.Authorization.Decision	agora	beta		
Security.Authorization	agora	beta		
Security.IdentityMapping	agora	beta		
Security.Accounting	Grid Batch Accounting System	Production		
Data.Transfer	HTTP	Production	HTTP1.1	GridFTP
Data.Management.Transfer	Grid File Management System	Production		
Data.Access.FlatFiles	Grid File Management System	Production		
Data.Naming.Scheme	a three-level naming scheme	beta		
Data.Naming.Resolver	meta file service in Grid File Management System	beta		
Information.Model	meta info and schema	Production		CIM
Information.Discovery	Grid Router	Production		
Information.Logging	Log4j wapper	Production		
Information.Monitoring	Grid Monitoring System	Production		GMA
Information.Provenance	N/A			
ExecMan.JobMan	Grid Batch System	Production		
ExecMan.JobDescription	xml schema	beta		JSDL
ExecMan.ExecutionAndPlanning	N/A			
ExecMan.CandidateSetGenerator	N/A			
ExecMan.Reservation	According to the local batch system			

3.5 UNICORE 5

Table 6 UNICORE 5 decomposition

Name	<i>UNICORE 5</i>
Last Stable Version	5
Dissemination Website	http://www.unicore.eu
Source Code Repository	https://sourceforge.net/projects/unicore/
Reference Document Architecture	http://www.fz-juelich.de/zam/vsgc/pub/streit-2005-UFP.pdf
Licence	Open Source under BSD licence

	Subsystem Name	Development Status	Adopted Standard	Foreseen Standard
Security.Authentication	UNICORE Gateway	production	IETF RFC3820	
Security.CredentialStorage	Java Keystores	production		
Security.Delegation	Explicit Trust Delegation (ETD)	production		
Security.AttributeAuthority	UUDB	production		SAML
Security.Authorization	UUDB	production	IETF RFC3820	
Security.IdentityMapping	UUDB	production	IETF RFC3820	
Security.Accounting	RMS	Vendor-specific		
Data.Transfer	UPL, GridFTP	production	GridFTP	ByteIO
Data.Management.Transfer	NJS			
Data.Management.Replica				
Data.Management.Storage				
Data.Naming.Resolver				
Data.Naming.Scheme	NJS, TSI, Gateway	production		
Data.Access.Relational				
Data.Access.XML				
Data.Access.FlatFiles	TSI	production		
Information.Model				
Information.Discovery				
Information.Logging	TSI, NJS, Gateway	production		
Information.Monitoring	Client	production		
Information.Provenance				
ExecMan.BES	TSI	production		
ExecMan.JobDescription	AJO	production		JSDL
ExecMan.JobManager	NJS	production		
ExecMan.ExecutionAndPlanning	NJS	production		
ExecMan.CandidateSetGenerator				
ExecMan.Reservation				

3.6 UNICORE 6

Table 7 UNICORE 6 decomposition

Name	<i>UNICORE 6 alpha</i>
Last Stable Version	UNICORE 5 (production)
Dissemination Website	http://www.unicore.eu
Source Code Repository	https://sourceforge.net/projects/unicore/
Reference Document Architecture	http://www.fz-juelich.de/zam/vsgc/pub/riedel-2006-SPU.pdf
Licence	Open Source under BSD licence

	Subsystem Name	Development Status	Adopted Standard	Foreseen Standard
Security.Authentication	UNICORE Gateway	Alpha	IETF RFC3820	
Security.CredentialStorage	Java Keystores/ VOMS	Alpha		
Security.Delegation	Explicit Trust Delegation (ETD)	Alpha		
Security.AttributeAuthority	WS-UUDB	Alpha		SAML
Security.Authorization	WS-UUDB	Alpha	IETF RFC3820	
Security.IdentityMapping	WS-UUDB	Alpha		
Security.Accounting	RMS, Accounting Service	Design-phase completed	OGF RUS, UR	
Data.Transfer	ByteIO, GridFTP, HTTP Transfer, BaseLine Transfer	Alpha	OGF GridFTP	
Data.Management.Transfer	File Transfer Service	Alpha		
Data.Management.Replica				
Data.Management.Storage	Storage Management Service	Alpha		OGF DMI
Data.Naming.Resolver				
Data.Naming.Scheme	Proprietary scheme exposed as WS-RPs	Alpha		
Data.Access.Relational	OGSA-DAI	Alpha		
Data.Access.XML	OGSA-DAI	Alpha		
Data.Access.FlatFiles	File Transfer Service	Alpha		
Information.Model	WS-RPs	Alpha		
Information.Discovery	WS-RPs	Alpha		
Information.Logging	Gateway, UNICORE 6 servers	Alpha		
Information.Monitoring	Client, WS-RPs	Alpha		
Information.Provenance				
ExecMan.BES	Target System Service	Alpha	OGF JSDL	OGSA-BES and JSDL
ExecMan.JobDescription	JSDL	Beta	OGF JSDL	
ExecMan.JobManager	Job Management Service	Alpha		OGSA-BES
ExecMan.ExecutionAndPlanning	Job Management Service	Alpha		OGSA-BES
ExecMan.CandidateSetGenerator				
ExecMan.Reservation				

3.7 OMII-UK

Table 8 OMII-UK decomposition

Name	OMII-UK
Last Stable Version	3.2 (Early November 2006)
Dissemination Website	www.omii.ac.uk
Source Code Repository	Included in distribution
Reference Document Architecture	
Licence	Various open-source (Modified BSD, Apache, GPL)

	Subsystem Name	Development Status	Adopted Standard	Foreseen Standard
Security.Authentication	Hosting Environment Authentications	Production	WS-Security (X.509 Digital Signatures)	
Security.CredentialStorage				
Security.Delegation				
Security. Authorization	Hosting Environment Authorization	Alpha	SAML 1.x Assertion	OGSA-Authz
Security.AttributeAuthority		Considering Shibboleth or VOMS		XACML
Security.IdentyMapping	GridSAM	Production		
Security.Accounting		Alpha	RUS & UR	Refinements of RUS & UR
Data. Transfer				GridFTP
Data.Management.Transfer		Considering RFT or FTS		DMI
Data.Management.Replica				
Data.Management.Storage				
Data.Naming.Resolver				
Data.Naming.Scheme				WS-Naming
Data.Access.Relational	OGSA-DAI	Production		WS-DAI, WS-DAIR
Data.Access.XML	OGSA-DAI	Production		WS-DAI, WS-DAIX
Data.Access.FlatFiles				
Information.Model		Considering GLUE or CIM		
Information.Discovery	Grimoires	Production		UDDI
Information.Logging				
Information.Monitoring				
Information.Provenance				
ExecMan.BES	GridSAM	Production	OGSA Basic Execution Service	
ExecMan.JobDescription	GridSAM	Production	JSDL	
ExecMan.JobManager				
ExecMan.ExecutionAndPlanning	Taverna & BPEL Manual workflow systems	Production	ScuFL & BPEL	
ExecMan.CandidateSetGenerator	KNOOGLE	Alpha		
ExecMan.Reservation				

3.8 ARC

The ARC open source middleware development is coordinated by the NorduGrid collaboration [NORDUGRID], with support from various projects, most notably; the EU KnowARC project [KNOWARC] and the Nordic NDGF project [NDGF]. The present design and architecture are described in the published paper [ARCPAPER], while the architecture design of the next generation ARC middleware will be made public shortly (4Q 2006).

Table 9 ARC decomposition

Name	Advanced Resource Connector (ARC)
Last Stable Version	0.4.5 (next major version is in beta testing, the release 0.6 is scheduled 4Q 2006)
Dissemination Website	http://www.nordugrid.org
Source Code Repository	http://cvs.nordugrid.org
Reference Document Architecture	http://dx.doi.org/10.1016/j.future.2006.05.008
Licence	GPL v2

OGSA Capabilities	Subsystem Name	Development Status	Adopted Standard	Foreseen Standard
Security.Authentication	ARC GridFtp server	production	IETF RFC3820 , ITU-T X.509	
Security.Authentication	ARC HTTPS container	production	IETF RFC3820 , ITU-T X.509 OGF-GFD.24	
Security.Authentication	LDAP server	production	IETF RFC3820 , ITU-T X.509 OGF-GFD.24	
Security.Authentication	ARC UserInterface	production	IETF RFC3820 , ITU-T X.509 OGF-GFD.24	
Security.CredentialRetrieval	ARC GridManager	beta-production	MyProxy GFD.54	
Security.Delegation	ARC GridManager	production	IETF RFC3820 OGF-GFD.24	
Security.Delegation	ARC HTTPS container	production	IETF RFC3820 OGF-GFD.24	
Security.Authorization	ARC GridManager: GACL	production	GACL	
Security.Authorization	ARC HTTPS container	production	GACL	
Security.IdentityMapping	ARC GridManager: mapfile	production		
Security.IdentityMapping	ARC GridManager: LCMAPS/LCAS	beta-production		
Security.Accounting	SGAS	production		
Data.Transfer	ARC GridFTP server	production	GridFTP v1	GridFTP v2
Data.Transfer	ARC HTTP(s)	production	RFC 2818	
Data.Transfer	ARC UserInterface	production	GridFTP v1 , RFC2818, RFC959	
Data.Management.Storage	ARC SSE	production	SRM 1.1,	SRM 2.2
Data.Naming.Scheme				
Data.Naming.Resolver	ARC GridManager:cache,	production	Posix	
Data.Naming.Resolver	ARC datamove module	production	GT-RLS, GT-RC, Glite-LFC	
Information.Model	ARC Schema	production		Glue 2
Information.Discovery	ARC UserInterface	production	LDAPv2	
Information.Logging	ARC Logger	production	SQL, HTTPS, SOAP	OGF-UR draft

Information.Monitoring	ARC UserInterface,	production	LDAPv2	
Information.Monitoring	ARC monitor	production	LDAPv2	
ExecMan.BES	ARC GridManager	production		
ExecMan.JobDescription	ARC xRSL	production	GT-RSL	
ExecMan.JobDescription	JSDL	beta	JSDL 1.0	JSDL 2
ExecMan.JobManager	ARC portal	alpha		
ExecMan.JobManager	Arconaut	beta		
ExecMan.JobManager	ARC UserInterface	production		
ExecMan.ExecutionPlanning	ARC UserInterface	production		
ExecMan.CandidateSetGeneration	ARC UserInterface	production		

4 Analysis

Table 10 Middleware comparison by capabilities

Capability	gLite	Globus	UNICORE 5	UNICORE 6	CROWNGrid	Vega-GOS	OMII-UK	ARC	Convergence
Security.Authentication	std: ITU X.509+RFC3820	std: ITU X.509 +RFC3820	std: ITU X.509 +RFC3820	std: ITU X.509+RFC3820	std: ITU X.509 +RFC3820+Kerberos	std: ITU X.509	std: WS-Security (X.509 Digital Signatures)	std: ITU X.509 +RFC3820	std: ITU X.509 + RFC 3820
Security.CredentialStorage	MyProxy, GFD.24	MyProxy, GFD.24	Java Keystores	Java Keystores	CROWN CredMan	WS-based service similar to MyProxy		MyProxy, GFD.54	
Security.Delegation	std: ITU X.509 + RFC 3820	std: ITU X.509 + RFC 3820 + WS-Trust	Explicit Trust Delegation (ETD)	Explicit Trust Delegation (ETD)	std: ITU X.509 + RFC 3820	std: ITU X.509 + RFC 3820 + proprietary interface		std: ITU X.509 + RFC 3820	std: ITU X.509 + RFC 3820 + WS-Trust
Security.AttributeAuthority	VOMS		UUDB	WS-UUDB	CROWN AA	Agora		VOMS	Attribute Authority + SAML
Security.Authorization	G-PBox, gJAF	CAS	UUDB	WS-UUDB	CROWN AuthZ	Agora	GridSAM		XACML, SAML
Security.IdentityMapping	LCMAPS		UUDB	WS-UUDB	CROWN CredFed				
Security.Accounting	DGAS+APEL	SGAS	RMS	RMS, RUS	Part of CROWN NodeServer			SGAS	OGF RUS/UR
Data.Transfer	GridFTP	GridFTP	UPL, GridFTP	GridFTP	LDS	HTTP		GridFTPv1	GridFTPv2
Data.Management.Transfer	FTS	RFT	NJS	JMS	MDS			Datamove	OGSA-DMI
Data.Management.Replica	lcg-utils	DRS							
Data.Management.Storage	DPM, StoRM		NJS	SMS				SSE	SRM 2.2
Data.Naming.Scheme	LFN,TURL,SURL		NJS, TSI, Gateway		LGN, LCN, PFN	EAS, VAS, PAS	WS-Naming		WS-Naming + WS-Addressing
Data.Naming.Resolver	LFC, DPM, StoRM	RLS			LDS	GFMS			
Data.Access.Relational		OGSA-DAI		OGSA-DAI	OGSA-DAI		OGSA-DAI		WS-DAIR
Data.Access.XML		OGSA-DAI		OGSA-DAI			OGSA-DAI		WS-DAIX
Data.Access.FlatFiles	GFAL, gsrifio	XIO,OGSA-DAI	TSI	TSI					ByteIO
Information.Model	GLUE Schema 1.2	GLUE Schema 1.1	Proprietary schema	Proprietary schema				ARC Schema	GLUE Schema 2
Information.Discovery	MDS 2.x, Service Discovery	MDS 4			RLDS+SCLub	Grid Router	Grimoires	LDAP v2	UDDI
Information.Logging	Logging and Bookeeping					Wrapped Log4j		ARC Logger	
Information.Monitoring	R-GMA, GridICE, CEMon	MDS 4			CROWN Monitoring	Ganglia-based		LDAPv2	
Information.Provenance	Job Provenance								
ExecMan.BES	LCG-CE, gLite-CE, CREAM	WS-GRAM	NJS	JMS	CROWN Scheduler		GridSAM	GridManager	OGSA-BES
ExecMan.JobDescription	JDL	XML-based	AJO	JSDL	JSDL	XML-based	GridSAM	GT-RSL,JSDL 1	JSDL 1.x
ExecMan.JobManager	WMS	CSF			CROWN Scheduler			UserInterface	
ExecMan.ExecutionAndPlanning	WMS	CSF	NJS	JMS	CROWN Scheduler		Taverna & BPEL Manual workflow systems	UserInterface	OGSA-RSS
ExecMan.CandidateSetGenerator	WMS	CSF			RLDS		KNOOGLE	UserInterface	OGSA-RSS
ExecMan.Reservation	WS-Agreement				CROWN Scheduler/Node				WS-Agreement

In Table 10, we present a summary comparing the different Grid platforms considered in this document. The table is structured as follows: (1) each row is related to a certain OGSA capability; (2) there is a dedicated column for each platform considered in the previous part of this paper; (3) the right-most column is dedicated to express the convergence standard that could be adopted to enable interoperability among the different Grid platforms. A line with a grey background means that the related capability is already covered by an activity within OMII-Europe, therefore the convergence path for interoperability is already planned and resources are allocated. In the following subsections, we focus on a subset of the capabilities not yet covered by OMII-Europe.

4.1 Security.Authentication

The authentication is based on the X.509 certificates plus the use of proxy for supporting other features as single-sign-on and delegation. In the evolution towards Web Services, the authentication assertion is integrated with the WS-Security and WS-SecureConversation standards at the message level [GT4SEC]. All the analyzed Grid platforms rely on these building blocks. The JRA3-security activity should verify that these components are used in an interoperable manner.

4.2 Security.CredentialStorage

Most of the analyzed middlewares make use of the MyProxy component as credential storage for End Entity and Proxy X.509 certificates, and private keys. This component might be considered as an off the shelf component.

4.3 Security.Delegation

The delegation is based on the proxy certificate. For Web Services, this is extended using WS-Trust. The only platform that has a different approach on delegation is UNICORE. We recommend that the JRA3-security activity should identify converge paths among the different approaches on delegation.

4.4 Security.Authorization

In the analyzed Grid platforms, while the authentication capability relies on ITU X.509 for all of them, the authorization aspects are faced in similar, sometimes different ways. This is an obstacle to interoperable Grid systems, especially for all the authorization management aspects at the Virtual Organization level.

The Open Grid Forum is developing a description of the functional components for a Grid authorization service [AUTHZFRAM]. This informational document does not want to mandate a rigorous architecture. The Globus project has defined a multi-policy authorization framework [GLOBUSAUTHZ] based on widely adopted standards like XACML and SAML. The EGEE project has a distributed policy framework [G-PBOX] relying on XACML. G-PBox is not yet part of the production release, while it is in the preview testbed.

While many approaches to authorization services for Grids exist, they tend to rely on the same core set of technologies related to XACML and SAML. Interoperability aspects should be evaluated, possibly by more suitable OMII-Europe activities (e.g., JRA3 security) and an agreed framework should be formalized with well-defined interfaces and architectural components to enable interoperability at the authorization level. The impact of emerging approaches related to Shibboleth should also be evaluated due to the increasing community interest [SHIB, GRIDSHIB]. This activity should be part of the JRA3 Security. While this activity is carried on, the OMII-Europe could assign to JRA2 the task of including in the repository an authorization service that already complies with relevant standards. If this activity is assigned to JRA2, then a tight relationship with JRA3-security should be established.

4.5 Information.Model

An information model is an abstraction of real world into constructs that can be represented in computer systems (e.g., objects, properties, behaviour, and relationships) not tied to any particular implementation. In current Grid systems, information models are typically used to share a common definition of Grid resources for the purpose of resource discovery, selection and monitoring.

Within the Grid community, there are several proposals for describing resources available in a Grid system. The most widely adopted is the GLUE Schema, currently in use within the whole EGEE infrastructure and in the OSG infrastructure. The GLUE Schema was started in 2002 for enabling a common definition of Grid resources and it was the product of a joint collaboration of European and American Grid related projects.

Within the OGF, there are two main information models to be mentioned. They both ground on CIM. One is the Job Submission Information Model (JSIM) that was defined to be a common schema, but it did not succeed in terms of wide adoption. The other one is a core set of terms included in the JSDL. These can be extended via an external schema identified as Resource Requirement Language (RRL) that should be possibly contributed.

In this scenario, we can argue that, while there are many attempts to define a common information model for Grid resource, a widely accepted solution among standard bodies and Grid platforms is still lacking.

The OGF viewpoint is that such a schema should be an extension of the CIM. On the other side, the community around the GLUE Schema is expanding and a major revision will start shortly. In particular, the GLUE Schema interest group planned to move the evolution of the GLUE Schema into the Open Grid Forum in order to define it as a community standard. Other projects have expressed interest in participating in this approach (e.g., ARC, UNICORE, NGS).

We believe that OMII-Europe can play an important driver role in this phase. This capability could be selected for the JRA2 activity and the purpose could be to work in relationship with the GLUE Schema group and the OGF in order to contribute to the evolution of the GLUE Schema as a Resource Requirement Language for JSDL described on the CIM meta-model and coherently integrated with OGSA-BES properties. The final result could be included in the OMII repository.

4.6 Information.Discovery

A Grid system requires an information service in order to achieve visibility of resources. This is a vital capability not yet envisioned by the OMII-Europe repository. Many solutions exist, nevertheless a reference standard is lacking. A community standard in the context of Web Services for discovery is UDDI (Universal Description, Discovery and Integration), but this has never been recognized as providing the flexibility required by a Grid environment. While a general discovery service would be beneficial for interoperability among Grids, we do not believe that this can be addressed and solved in the context of this activity in the one-year timeframe.

4.7 Information.Monitoring

In the area of Grid monitoring, there are many tools fulfilling different needs. While the Open Grid Forum defined an abstract architecture for a Grid Monitoring Architecture (GMA), there is no stated or implied convergence path among them. The monitoring information of each service that are necessary to be exposed should be accessible in a standard way. In the Web Services area, the WS-RF specification offers mechanisms that could be used to expose such information (WS-ResourceProperties), nevertheless, the WS-RF is not adopted by all services nor it will be in the near future. A possible building block could be to define a common WSDL port for each service

that enables to inspect a service status. The approach should be document-oriented in the sense that the method should return an XML document describing the service status in some format. If there are pre-WS components, then an external service should be used to expose their information (e.g., gLite CEMon).

4.8 Information.Logging

While some middleware offers a logging service, there is no defined standard interface.

4.9 Information.Provenance

The provenance capability is only offered by the gLite middleware. A standard interface and agreed architecture for this service is lacking, nevertheless a European Project is dedicated to it [EUPROV]. We do not consider this service as relevant for this phase of the OMII-Europe project.

4.10 Data.Transfer

The OMII-Europe repository does not include a data transfer service. Most of the analyzed platforms support GridFTP, while a few of them support HTTP. A possible activity is the analysis and inclusion of the best instance of GridFTP implementation. The inclusion of HTTP could be also evaluated. We should consider this activity as a low priority since these components are well established and can be treated as off the shelf components.

4.11 Data.Management.Transfer

The management of data transfer is an essential aspect for Grid systems that are mainly oriented at data sharing and require the moving of huge amount of data. Some of the analyzed platforms offer tools for reliable file transfer movement; nevertheless they do not satisfy a community standard as this is being defined [DMI].

4.12 Data.Management.Replica

Among the analyzed middlewares, the management of the creation of file replica and related registration into the resolver services is provided by gLite as a command line tool, while Globus provides a high-level service called Data Replication Service (DRS). No standard exist yet and this capability is not considered essential in this phase since it depends on other services that are not yet present into the OMII-Europe repository.

4.13 Data.Management.Storage

Along with computing and networking resources, data storage resources are the basic building blocks of a distributed computing infrastructure. All data is ultimately stored on a data storage resource. Different storage resources offer different levels of Quality of Service, and have different semantics for data access, both for reading and writing [OGSADATAARCH].

Within the OGF, the Grid Storage Management WG is working on making the de-facto standard Storage Resource Management (SRM) interface specification [SRM22] to become an OGF standard. Among the reviewed Grid platforms, only two of them provide an SRM implementation (gLite and ARC).

While the OMII-Europe already covers the aspect of data access for structured and semi-structured data via the OGSA-DAI related technologies, the aspect of storage management is not covered and should therefore be considered. A process of evaluation and selection of the most mature and flexible SRM implementation could be performed in order to include it in the OMII-Europe repository.

4.14 Data.Naming.Scheme and Data.Naming.Resolver

Each project uses a different approach to the naming of files; nevertheless they typically have a three-level naming. A common solution could be based on the WS-Naming and WS-Addressing standards. Nevertheless, such common solution does not exist yet, nor a common standard for the resolver service.

4.15 Data.Access.File

The input/output operations on remote files can be achieved by different solutions in the different middleware. An emerging specification (not yet finalized) from the OGF is ByteIO.

4.16 ExecMan.JobManager

The Job Manager has the important task of managing the execution of a job or set of jobs from start to finish. It is a complex activity that requires the interaction with many other services (e.g., discovery, data resolver, execution and planning). The standardization activity is targeted at defining a scheduling architecture that supports cooperation between different scheduling instances for arbitrary Grid resources [GSA]. There is no activity in the definition of a common interface for a job manager.

Among the analyzed platforms, gLite seems to provide the most advanced system. Nevertheless, the inclusion of such a component into OMII-Europe would require the presence of other key capabilities not yet present (e.g., the discovery service and an information model). Therefore, we do not consider this component of a high priority at a present stage.

4.17 ExecMan.ExecutionAndPlanning and ExecMan.CandidateSetGenerator

The Open Grid Forum is defining a standard specification for a service called Resource Selection Service (RSS) [OGSARSS]. This service includes both execution and planning capability and the candidate set generator capability. Some of the analyzed platforms include such features using project specific choices. This capability is not considered to be of a high interest in this phase of the project since services from which it would depend are not present (e.g., information.discovery and information.model).

4.18 ExecMan.Reservation

The reservation service is standardized by the WS-Agreement. While some of the analyzed middleware include implementation for this service, we do not consider it as essential in the current phase of the project.

5 Cost-Risk-Value Based Selection

In this section, we present the approach for selecting the OGSA capabilities not covered by OMII-Europe that can be considered in the second year of the JRA2 activity for the re-engineering and porting into the OMII-Europe repository. The selection is based on a cost-risk-value model. The cost parameter refers to the quote of FTE estimated for the activity. Its value can be maximum two because the subcontract is 100,000 Euros, thus meaning two FTE for one year. The parameters risk and value use the following enumeration: low, medium, high. The risk indicates the possibility that the re-engineering process for the inclusion into the OMII-Europe repository can fail; the value indicates the benefit that the inclusion of the considered capability can bring to the global value of the OMII-Europe repository.

For the analysis given in Section Analysis4, we report only those capabilities that we consider meaningful for inclusion in the second year of the projects as part of the JRA2 activities. The other capabilities are not selected for different reasons, for instance because there is lack of capabilities on which they rely on or because the convergence standard is either not present or not mature.

Table 11 Cost-Risk-Value comparison

Capability	Activity	Convergence towards	Current Standard Compliance	Cost	Risk	Value
Information.Model	Collaborate for GLUE Schema 2.0 as extension of CIM, RRL vocabulary for JSDL and OGSA-BES vocabulary	- RRL for JSDL - CIM-based - OGSA-BES	0%	1	Low	High
Data.Management.Storage	Select StoRM and integrate into OMII-EU; collaborate with OGF GSM WG	SRM 2.2	90%	1	Low	High
Security.Authorization	Analysis of interoperability aspect of authorization systems together with JRA3 Inclusion of G-PBox	XACML, SAML	50%	1	Medium	High
Data.Management.Transfer	Select a data movement service and adopt it to emerging OGSA DMI	OGSA-DMI	0%	1	High	High

6 Conclusions

The purpose of this document was to identify a number of appropriate set of emerging middleware services from the global Grid initiative to be included into the OMII-Europe repository. The definition of priorities for the placement of such services was also a goal. The analysis was conducted mainly concentrating on the Grid platforms already involved in the OMII-Europe project. Furthermore, contributions from external projects were considered. The comparison was performed by identifying the relevant set of capabilities described in the Open Grid Service Architecture, and by decomposing the different Grid platforms based on these capabilities. A comparative analysis has been later performed trying to identify potential convergence paths based on community standards. The final step was to reduce the number of capabilities to a meaningful, but small subset and to prioritize them based on a cost-risk-value model.

Given the results of our analysis, we recommend that the subcontract of 100,000 Euros should be invested on the following capabilities given in decreasing preference order: information.modeling, data.storage.management, security.authorization, data.managment.transfer.

We also recommend that the JRA3 Security should analyze how the OMII-related Grid platforms address the following aspects: (1) authentication, there are common building blocks, nevertheless it should be verify the effective compatibility; (2) delegation, UNICORE has a different approach than the others, therefore an interoperable solution should be found; (3) authorization, this is the highest part of the work; a common and interoperable solution among the different solutions should be evaluated; if authorization is selected as one of the JRA2 activity for the inclusion of an authorization service, than there should be a tight interaction among JRA2 and JRA3-security.

A Description of the Grid Platforms

A.1 gLite

A.1.1 Security

In gLite, security services encompass the Authentication, Authorization, and Auditing services which enable the identification of entities (users, systems, and services), allow or deny access to services and resources, and provide information for post-mortem analysis of security related events [EGEEARCH].

Security.Authentication

Identity assertions are based on X.509v3 public key certificates [PKI], whilst the single sign-on relies on proxy certificates (RFC3820) [RFC3820]. This makes PKI based authentication system with proxy credentials the starting point of authentication services in gLite. Proxies can be generated either based on long-term user credentials or via the integration with site-level credential systems (e.g., Kerberos Leveraged PKI [KLP]). Recently, an activity to generate X509 certificate from Shibboleth [SHIB] credentials was started [EGEESHIB].

Security.CredentialStorage

The long-term user credentials can be maintained in dedicated components in order to deal with user mobility or proxy renewal. The gLite middleware adopts the MyProxy [MYPROXY] components to deal with this capability.

Security.Delegation

In order to be able for Grid users to delegate some subset of their privileges to another (dynamically created) entity on relatively short notice, the use of Proxy Certificates [RFC3820] is selected. It has been the mechanism most widely adopted by the Grid community to date, as the technology needs no additional infrastructure services, and at the same time it also solves the single sign-on and dynamic entity identification problems. Besides providing some coarse-grained controls (such as describing whether all or none of a user's privileges are implied in the delegation, and the right to delegate further), there is also a placeholder for adding arbitrary, typically application-specific, policy restrictions.

Security.Authorization

The gLite architecture envisions four main authorization systems.

The first is GridSite, a GACL-based authorization system based on Apache [GRIDSITE].

The second is gJAF, the gLite Java Authorization Framework.

The third system is a local authorization system composed called LCAS (Local Centre Authorization Service) [LCAS]. This is a site-local service that can authorize users based on their name, their VO affiliation, and the resources requested. It is tightly related to LCMAPS (see Security.IdentityMapping).

The fourth is G-PBox [GPBOX], that is a policy assertion service which issues claims that gives a user (or set of users) the explicit privilege to perform an action (or a set of actions) on a certain resource (or set of resources). The G-PBox is more complex framework and as regards the source of authorization, the repository component is considered. An authorization decision is a complex task requiring the combination of information from a number of distinct sources. Different types of

policies may also exist (VO-oriented, local systems-oriented, and a mix of the two). The G-PBox component provides a Policy Decision Point (PDP) that relies on policies represented as XACML to perform the authorization evaluation. G-PBox is currently in development and deployed on the EGEE preview test-bed but not yet included in the gLite distribution.

Security.AttributeAuthority

Two main types of sources of authorization are present. The first is the Virtual Organization Membership Service (VOMS) [VOMS], which is an Attribute Authority (AA) which associates a user with a set of attributes in a trusted manner to a relying party, by way of digitally signed assertions.

Security.IdentityMapping

In gLite, the mapping of Grid credentials into a local UNIX account is performed the the LCMAPS (Local Credential Mapping Service) service [LCMAPS]. It makes sure user requests are sandboxes in local account with unique group memberships. Such accounts can span a machine or a cluster, in short, an entire administrative domain. LCMAPS is typically coupled with LCAS (Local Centre Authorization Service) [LCAS].

Security.Accounting

In the gLite middleware, there are two different and complementary accounting systems. The first is the Distributed Grid Accounting System (DGAS). It provides resource usage metering, accounting, and account balancing in a fully distributed Grid environment through an arbitrary number of distributed accounting servers and (optional) resource pricing servers. DGAS allows usage records to be forwarded from the sites' accounting servers to the VOs' accounting servers, such that accounting information can be made available (through a proper authorization mechanism) to both sites and users, while preserving a reasonable scalability that cannot be achieved through a single centralized accounting repository. The second accounting system including in gLite is APEL. It enables to transmit accounting information to a centralized database at the Grid Operations Centre (GOC) and providing graphical representations of aggregated (summary) accounting information.

A.1.2 Data

Data.Transfer

In gLite, the actual transfer of files is performed relying on the GridFTP protocol [GRIDFTP] through the SRM interface [SRM2.2].

Data.Management.Transfer

In the gLite middleware, the management of data transfer is part of a category of services called data movement. In this category, the gLite File Transfer Service FTS is a high-level data movement service, responsible for moving sets of files from one site to another while allowing participating sites to control the network (channel) resource usage. This control includes the enforcement of site and usages policies such as fair-share mechanisms on predefined channels. It is designed for point to point movement of physical files. The FTS has dedicated interfaces for managing channels and to display statistics of ongoing transfers. The FTS is also able to communicate with external Grid File Catalogues via the use of VO-specific plug-in's. This allows for instance that the file to be transferred can also be specified using an LFN (Logical File Name) [GLITEPROG].

The FTS has three interfaces that can be used for programming. The File Transfer Interface is used to submit File Transfer jobs, get status on current jobs, list requests in a given job state, cancel transfers, set priority of transfers; and to add, remove and list VO managers. The Channel Management Interface can be used to add, list and delete channels for the FTS instance, and set

channel parameters. It has also methods to add, remove and list channel managers and to apply policies for jobs that need manual intervention, such as being in HOLD state. Finally, the Status Interface can be used to list or summarize the channel and VO activity, and to list all running background Transfer Agent processes.

Data.Naming.Scheme

In gLite, the naming scheme for files is structured in three levels: Logical File Name (LFN), Transfer File Name (TURL) and Storage File Name (SURL).

Data.Naming.Resolver

In gLite, the resolution of file names is performed by the LCG File Catalogue. This is a store of information about the data and metadata that is being operated on the EGEE infrastructure. The catalogues are used to manage the Grid file namespaces and the location of the files, to store and retrieve metadata and to keep data authorization information. The LCG File Catalogue (LFC) is a secure file and replica catalogue mapping logical file names to physical replicas of the file. It supports full POSIX namespace with secure Grid access. Both central file catalogue and local file catalogue modes are supported [LFC].

Data.Management.Storage

In gLite, all storage systems exposed to a Grid are required to comply with the Storage Resource Manager (SRM) interface. The SRM interface itself is being standardized through the Open Grid Forum in the context of the Grid Storage Management Working Group [GSM]. The SRM abstraction can be built on top of different categories of storage systems, from simple disk-based systems to complex hierarchical mass-storage systems. The current release of gLite middleware includes the Disk Pool Manager [DPM] developed at CERN [CERN] and dCache [DCACHE] while Castor [CASTOR] developed by CERN and StoRM (Storage Resource Manager) [STORM] developed by INFN [INFN] and ICTP [ICTP] are provided as external products.

A.2 Information

Information.Discovery

In gLite, the discovery service is currently based on LDAP [LDAP] technology inspired to the Globus MDS 2.x [MDS2]. The tree hierarchy of components is progressively updated with a Berkeley Database backend. In the near future, the discovery interface will be replaced with a common interface defined in agreement with other Grid-related projects (TO ADD REFERENCE). The backend will be a pluggable component where MDS can be replaced with other technologies (e.g., R-GMA [RGMA]). A "boot-strapping" mechanism for the Service Discovery component is in development to remove the need of a static configuration file.

Information.Monitoring

The monitoring activity in gLite is performed by several tools that fulfil different goals. The R-GMA (Relational Grid Monitoring Architecture) is an implementation of the Grid Monitoring Architecture (GMA [GMA]) in the context of the relational data model. It makes all the information appear like one large Relational Database that may be queried to find the information required. It consists of Producers which publish information into R-GMA, and Consumers which subscribe. Other components such as registries, mediators and republishers are also present (see [RGMA]).

GridICE [GRIDICE] is a distributed monitoring tool promoting the adoption of de-facto standard Grid Information Service interfaces, protocols and data models. Further, different aggregations and partitions of monitoring data are provided based on the specific needs of different users categories (virtual organizations, grid operation centres, site). Being able to start from summary views and to drill down to details, it is possible to verify the composition of virtual pools or to sketch the sources

of problems. A complete history of monitoring data is also maintained to deal with the need for retrospective analysis.

The CEMon Web Service enables the provision of monitoring information for different services. It acts as a source of information that supports pluggable information providers. It also supports both query-response and subscription-notification information delivery [CEMON].

Information.Model

Resources that are published in the Grid Information Service for the purpose of discovery and selection are modelled relying on the GLUE Schema [GLUESHEMA]. The GLUE Schema is composed of an abstract modelling described in terms of the Unified Modelling Language (UML) Class diagrams. Mapping into concrete schemas are also provided for the following data models: relational, LDAP and XML.

Information.Logging

In gLite, the logging capability is provided by the Logging and Bookkeeping (L&B) [LB] service. This service is mainly used to tracks jobs managed by the gLite WMS (workload management system). It gathers events from various WMS components in a reliable way and processes them in order to give a higher-level view, the status of job. Virtually all the important data are fed to L&B internally from various gLite middleware components, transparently from the user's point of view. In addition, L&B provides public interfaces for querying the job information as well as registering for notifications.

Information.Provenance

In gLite, the provenance capability is provide by the Job Provenance (JP) [JP] service which goal is to provide long-term storage of data related to job live in a Grid. JP provides a query interface allowing to perform data mining. The possibility to annotate jobs stored in JP is also provided. JP is currently in development and deployed on the EGEE preview test-bed but not yet included in the gLite distribution.

A.2.1 ExecMan

ExecMan.JobManager

This capability refers to the execution and management of jobs at a certain resource such as a simple machine or a complex cluster managed by a local batch system. In gLite, there are several implementations for this capability. The LCG CE is based on the Globus GRAM 2.x [GRAM2]. The gLite CE [GLITECE] is based on GSI-enabled Condor-C. The Computing Resource Execution And Management (CREAM) [CREAM] is Web Service based and a candidate component to adopt the upcoming Basic Execution Services (BES) interface specification being defined by the Open Grid Forum [BESWG]. Both the gLite CE and CREAM rely on a common abstraction layer called BLAHP [BLAHP] to isolate the different batch systems from the Grid layer. CREAM is currently in development and deployed on the EGEE preview test-bed but not yet included in the gLite distribution.

Considering the OGSA definitions, the Job Management capability falls into the Workload Management System (WMS) [WMS] for its job management functionalities. The WMS is responsible for the distribution and management of tasks across Grid resources, in such a way that applications are conveniently, efficiently and effectively executed. The core component of the Workload Management System is the Workload Manager (WM), whose purpose is to accept and satisfy requests for job management coming from its clients. For a computation job, there are two main types of request: submission and cancellation. In particular, the meaning of the submission

request is to pass the responsibility of the job to the WM. The WM will then pass the job to an appropriate CE for execution, taking into account the requirements and the preferences expressed in the job description. The decision of which resource should be used is the outcome of a matchmaking process between submission requests and available resources. The WMS is able to perform other tasks such as managing workflows, supports proxy renewal, manages input and output of the jobs.

ExecMan.JobDescription

In gLite, the job description can be expressed using the Job Description Language (JDL) [JDL], a ClassAd-like [CLASSAD] language enabling to express the important information to enable a meta-scheduler to find both required data and suitable resources to execute and manage a simple job or complex set of jobs described in terms of a Direct Acyclic Graph (DAG). Compatibility with JSDL is being added now.

ExecMan.ExecutionAndPlanning

The job execution and planning is performed by the Workload Management System [WMS].

ExecMan.CandidateSetGenerator

Given a job description in terms of the JDL, the description of Grid resources based on the GLUE Schema and the information of data available in a Grid via replica and file catalogues, a component of the WMS called Matchmaker performs the generation of the possible set of resources that fulfil the user requirements. The final list can be ranked based on a user-defined arithmetic expression. The matchmaking process relies on the Condor Matchmaking algorithm performing selection among set of ClassAds representing both resources and user request.

ExecMan.Reservation

The gLite middleware includes a prototype of the WS-Agreement standard for performing advance reservation. This has been integrated with the WMS and tested in a use case considering the immediate reservation of storage space via an SRM 2.x compliant implementation.

A.3 Globus

A.3.1 Security

Security.Authentication

The authentication is based on GSI both in pre-WS and WS services.

Security.Delegation

GT4 supports a delegation service that provides an interface to allow clients to delegate (and renew) X.509 proxy certificates to a service. The interface to this service is based on the WS-Trust specification (the specification is not well defined enough to allow claim of compliance). Note that when delegating from a Proxy Certificate, the type of the delegated proxy will always be the same type as the initial proxy.

Security.Authorization

In addition to the grid-mapfile found in earlier versions of the Globus Toolkit, which provides access control based on a list of acceptable user identifiers, GT4 GSI uses the SAML standard from OASIS. SAML defines formats for a number of types of security assertions and a protocol for retrieving those assertions. GSI uses SAML AuthorizationDecision assertions in two ways: (1) the Community Authorization Service (CAS) issues SAML AuthorizationDecision assertions as its means of communicating the rights of CAS clients to services; (2) GSI uses a callout based on the SAML AuthorizationDecision protocol being defined in GGF [OGSASAML] to allow the use of a third party authorization decision service, such as PERMIS, for access control requests to GT4-based services.

Security.AttributeAuthority

GT4 provides distinct WS and pre-WS authentication and authorization capabilities. Both build on the same base, namely standard X.509 end entity certificates and proxy certificates, which are used to identify persistent entities such as users and servers and to support the temporary delegation of privileges to other entities, respectively.

GT4's WS security provides an Authorization Framework that allows for a variety of authorization schemes, including a "grid-mapfile" access control list, an access control list defined by a service, a custom authorization handler and access to an authorization service via the SAML protocol.

This has been extended in many ways, for example, CAS, GridShib Additionally VOMS can be used with GT4.

Security.IdentityMapping

Security.Accounting

The Globus Toolkit includes the SweGrid Accounting System (SGAS) [SGAS], that is a resource allocation enforcement and tracking service for the Grid, based on the latest Web services technologies. The SGAS has the following subcomponents:

- Bank: the central service of the accounting system that maintains and enforces allocation quotas
- LUTS: the Logging and Usage Tracking Service (LUTS) is a general purpose logging system for tracking resource usage in SGAS. It allows secure publication and query-based retrieval of usage data in the format of GGF UsageRecord XML

- **JARM:** the Job Account Reservation Manager (JARM) is a component responsible for integrating various workload managers, schedulers and local accounting systems deployed at the resource sites with SGAS. JARM is typically used as a callout to the bank during the job submission phase. The bank then issues a time-limited reservation to run the job, based on user, resource and bank policy. After the job has completed the job is logged in LUTS, and if a valid account reservation was made, JARM also charges the account in the Bank, and releases the reservation on behalf of the resource
- **PAT:** The Policy Administration Tool (PAT) component is designed to be used to manage the security policies of all of the SGAS services. It contains a command line tool that can be run in interactive or batch mode for easy scripting

A.3.2 Data

Data.Transfer

The Globus Toolkit relies on GridFTP for data transfer. It provides (1) a server implementation called globus-gridftp-server, (2) a scriptable command line client called globus-url-copy, and (3) a set of development libraries for custom clients.

Data.Management.Transfer

The Reliable Transfer Service (RFT) Service uses standard SOAP messages over HTTP to submit and manage a set of 3rd party GridFTP transfers and deletion of files and directories using GridFTP. The service also provides an interface to control various transfer parameters of the GridFTP control channel like TCP buffer size, parallel streams, DCAU etc. The user creates a RFT resource by submitting a Transfer Request (consisting of a set of third-party gridftp transfers) to the RFT Factory service. The resource is created after the user is properly authorized and authenticated. RFT service implementation exposes operations to control and manage the transfers (the resource). The resource the user created exposes the state of the transfer as a resource property to which the user can subscribe either for changes or poll for the changes in state periodically using standard WS-RF command line clients and other resource properties.

Data.Management.Replica

The Data Replication Service (DRS) combines two existing data management components: RFT and RLS. The function of the DRS is to ensure that a specified set of files exists on a storage site. The DRS begins by querying RLS to discover where the desired files exist in the Grid. After the files are located, the DRS creates a transfer request that is executed by RFT. After the transfers are completed, the DRS registers the new replicas with RLS. DRS is implemented as a Web service and complies with the Web Services Resource Framework (WSRF) specifications. When a DRS request is received, it creates a WS-Resource that is used to maintain state about each file being replicated, including which operations on the file have succeeded or failed.

Data.Naming.Scheme

There is no explicit data-naming scheme for all of GT4. EPRs are used to define service and resource endpoints in most services. RLS has a naming structure which can be found in the documentation here: <http://www-unix.globus.org/toolkit/docs/4.0/data/rls/>

RLS maintains a consistent local state maintained in Local Replica Catalogs (LRCs). Local catalogs maintain mappings between arbitrary logical file names (LFNs) and the physical file names (PFNs) associated with those LFNs on its storage system(s).

Data.Naming.Resolver

The Replica Location Service (RLS) provides the ability keep track of one or more copies, or replicas, of files in a Grid environment. This tool is especially helpful for users or applications that need to find where existing files are located in the Grid. RLS is a simple registry that keeps track of where replicas exist on physical storage systems. Users or services register files in RLS when the files are created. Later, users query RLS servers to find these replicas.

RLS is a distributed registry, meaning that it may consist of multiple servers at different sites. By distributing the RLS registry, we are able to increase the overall scale of the system and store more mappings than would be possible in a single, centralized catalogue.

The job of RLS is to maintain associations, or mappings, between logical file names and one or more physical file names of replicas. A user can provide a logical file name to an RLS server and ask for all the registered physical file names of replicas. The user can also query an RLS server to find the logical file name associated with a particular physical file location. In addition, RLS allows users to associate attributes or descriptive information (such as size or checksum) with logical or physical file names that are registered in the catalogue. Users can also query RLS based on these attributes.

Data.Access.Relational

The access to relational data is provided by OGSA-DAI, a pure Java data service framework for accessing and integrating data resources on to Grids.

Data.Access.XML

The access to XML data is provided by OGSA-DAI, a pure Java data service framework for accessing and integrating data resources on to Grids.

Data.Access.FlatFiles

The access to flat files is provided by XIO, an extensible input/output library written in C for the Globus Toolkit. It provides a single API (open/close/read/write) that supports multiple wire protocols, with protocol implementations encapsulated as drivers. The XIO drivers distributed with 4.0 include TCP, UDP, file, HTTP, GSI, GSSAPI_FTP, TELNET and queuing.

A.3.3 Information

Information.Model

The Globus Toolkit delivers an XML-based version of the GLUE Schema 1.1. Newer versions are not provided because their inclusion is delegated to middleware integrators.

Information.Discovery

The Monitoring and Discovery System (MDS) is a suite of web services to monitor and discover resources and services on Grids. This system allows users to discover what resources are considered part of a Virtual Organization (VO) and to monitor those resources. MDS services provide query and subscription interfaces to arbitrarily detailed resource data and a trigger interface that can be configured to take action when pre-configured trouble conditions are met.

The current release of MDS with GT4, called MDS4, includes two WSRF-based services: an Index Service, which collects data from various sources and provides a query/subscription interface to that data, and a Trigger Service, which collects data from various sources and can be configured to take action based on that data.

Information.Logging

Information.Monitoring

This capability is provided by the MDS4 (see Information.Discovery)

Information.Provenance

Provenance is not explicitly handled by GT4 - the applications must be aware of the need to store provenance information.

A.3.4 ExecMan

ExecMan.BES

GT4 supports the Grid Resource Allocation and Management (GRAM) interface as a basic mechanism for job execution. The GT4 GRAM server is typically deployed in conjunction with the Delegation and RFT service to address data staging, delegation of proxy credentials, and computation monitoring and management in an integrated manner. GRAM is not a resource scheduler, but rather a protocol engine for communicating with a range of different local resource schedulers using a standard message format. The current release of Globus ToolKit provides a Web service implementation of GRAM called WS GRAM.

ExecMan.JobDescription

The current Globus Toolkit enables an XML-based description of job submission.

ExecMan.JobManager

Job management capabilities are provided by the contributed Community Scheduler Framework (CSF).

ExecMan.ExecutionAndPlanning

Execution and Planning capabilities are provided by the Community Scheduler Framework (CSF).

ExecMan.CandidateSetGenerator

Candidate Set Generation is provided by the Community Scheduler Framework (CSF).

A.4 CROWNGRID

A.4.1 Security

In CROWN, the security architecture is designed to enforce secure communication and distributed access control. CROWN's security includes: (1) secure communication mechanisms; SOAP message level security are based on WS-Security, WS-Policy, WS-SecureConversation, WS-Trust, and IETF GSS-API specifications; (2) identity management mechanisms; the basic PKI or Kerberos security infrastructure is used to manage identity within a security domain; the identity mapping and credential conversation mechanism is also provided to identify the users when collaborations are conducted across domains with heterogeneous security infrastructures; (3) Policy-based Distributed Access Control mechanisms, that is a policy-based access control framework employed to established trust relationship between service requester and provider; the basic access control policy is based on XACML specification, and the authorization delegation mechanism across multiple secure domains is implemented based on an Extended Role-based Trust Management language. Besides these, a trust negotiation protocol is also provided when some privacy information in credentials or access control policies is needed to be protected.

Security.Authentication

The CROWN users and services' identity are based on X.509v3 public key certificates or Kerberos v5 Ticket. The proxy certificates are managed by CROWN Credential Management Service. In CROWN, authentication handler is used for local node authentication, and authentication service is used for domain-centralized authentication. They can authenticate Grid users or services by verifying signatures, and build and verify certificate chains. The related CA and Kerberos security infrastructures are not provided by the CROWN middleware; they can be based on the existing security infrastructures of the organization.

Security.CredentialStorage

In CROWN, the CredMan service is designed for the credential storage, which is similar to MyProxy. The client tools are also integrated with the portal, that is, a user can access his credential through portal. By using the tools, a user can easily delegate to and retrieve credential from the repository. Moreover, some client tools are provided for the user to manage the credential stored in the repository. In order to protect the credentials in the service repository, CredManService provides a protected mechanism in which user can specify authentication information and retrieval restrictions to protected his credentials in the repository.

Security.Delegation

The identity delegation mechanism is implemented with the CROWN CredMan service. At first, a user would use a CredMan client command, named credman-init, to visit the CredManService, and delegate a set of proxy credentials which are signed by the user's permanent credential to the service repository. Later when the user's credential is needed, the user, or service acting on the behalf of the user to get a proxy credential delegated from the proxy credential stored in the repository.

Some advanced technologies are also adopted from trust management for authorization delegation in multiple CROWN security domains, and a specified policy language is designed based on Role-based Trust Management Language, which can express the capability-based and attribute-based authorization delegation and some flexible attributes parameters constrains are also supported.

Security.Authorization

Policy-based authorization module in CROWN implements Policy Decision Point (PDP) in both handler and domain service. The XACML (eXtensible Access Control Markup Language) is adopted to express fine-grained access control policy in AuthzService. By using SAML assertions, the AuthzService can make authorization decision based on user attributes rather than identity.

Authorization handler intercepts each request sent to target service, and then collects attribute certificates signed by attribute authority for both user and service to form a request context, which is conducted by policy decision point to make an authorization decision for the request.

Security.AttributeAuthority

Security.IdentityMapping

The identity mapping function is provided by CredFed Service in CROWN, which can achieve identity mapping and credential conversation between X.509 certificate and Kerberos v5 ticket according to specified identity mapping policy.

First, the input credential is processed by the authentication module, which is realized by secure-conversation mode offered by underlying communication security component of CROWN-ST, to verify whether the user is the real owner of this credential. If so, the credential is then forwarded to identity mapping module, which will map the identity of user to another domain based on mapping policy. Then the new identity will be processed by credential conversion module to generate a new credential for the user. Finally, this credential is returned to the user.

Security.Accounting

Currently, the function of security accounting is a part of CROWN NodeServer Log Manager. When the service is invoked, all the accounting information related to security, such as requester identity, the secure communication policy information, are recorded.

A.4.2 Data

Data.Transfer

Considering the compatibility of OS, CROWN uses FTP protocol as the approach of file transfer. CROWN implemented the function of FTP control command by a set of web service, called Local Data Service(LDS), a component of CROWN, so CROWN provide the data transfer function without listening a static FTP port.

Data.Management.Transfer

CROWN Data Client can support the third side transfer. You simply provide one logical file name in CROWN, and the name or URL of destination LDS, then CROWN find the replica of the logical file in many LDS and then move the files from the LDSs to the destination LDS. All the transfer information is saved by the Data Client, so after the transfer failure the Data Client can restart or resume the transfer.

Data.Management.Replica

In CROWN, every logical file has a MD5 value. If a set of physical files have the same MD5 value that equals the one of the logical file, we think that these physical file are replicas of the logical file. The component MetaData Service (MDS) maintains the mapping between LGNs and one or more LCNs. When a LDS has a new physical file to share, it creates a LCN for the physical file, calculates the file's MD5 value, and then sends the LCN and the MD5 value to the MDS which it is registered to. The MDS create the mapping between LGN and the LCN, after receive the LDS's

request. If more than one LGN has the MD5 value equals the received, multiple mapping will be created.

In CROWN, the MDS is distributed. By distributing the MDS registry, we are able to increase the scale of the system, store more mappings and avoid creating a single point of failure in the Grid data management system. When user transfer a logical file from source LDS(s) to the destination LDS, a replica is created.

Data.Naming.Scheme

In CROWN, the naming scheme for files is structured in three levels: Logical File Name (LGN), Local File Name (LCN) and Physical File Name(PFN). LGN is the identifier of a set files that has the same value, LCN is the transfer URL of a physical file, and the PFN is the file path of the file in file system. More than one LCN can share one LGN, and one PFN can be linked to more than one LCN.

Data.Naming.Resolver

In CROWN, MDS resolves a LGN to a set of LCN by MD5 value, and the LDS resolves a LCN to one PFN.

Data.Access.Relational

OGSA-DAI is integrated into CROWN as component to access relational datasets.

A.4.3 Information

Information management in CROWN is co-supplied by several components including GIMS (Grid Information Model Service), Region Registry, Region Switch, RLDS (Resource Locating and Description Service) and SClub. It is noted that we choose 'RLDS' to name the subsystem implementing information management in CROWN though 'RLDS' is also one of the major components in the subsystem.

Information.Model

This capability is provided by GIMS that is responsible for managing all the information models used in a grid system. The description of a resource is modelled as a set of attribute-value pairs. GIMS does not bind to a certain information model (e.g Glue schema), instead it allows grid administrators to define, modify and delete their information models freely according to specific requirements. Thus, GIMS can be used to implement other standards like Glue schema or CIM (Common Information Model).

Information.Discovery

RLDS adopts a hybrid overlay that combines hierarchical tree and P2P to provide efficient information service. As the number of grid resources is large, many RLDS service are deployed to manage resource information and each deployed RLDS service is called a RLDS node. In the bottom layer, RLDS nodes are organized into a multi-tree framework and the root of a tree is called Region Switch. In the upper layer, all the Region Switch services communicate with each other in a P2P manner. Region Switch services can find each other through Region Registry service. Users can launch information discovery from any RLDS node, and searching is first performed within a tree; if user requirements are not met, the search will be further performed across trees through communications between Region Switch services. Beyond that, SClub service is designed to provide efficient discovery for 'hot' information. The SClub service dynamically computes the popular information, and records where to locate it. To put simply, SClub provides a short cut for discovering popular information.

Information.Logging

RLDS provides a set of interfaces for registering information entries. Information is collected through a variant of information providers. Before registering information to RLDS, information providers must transform data collected to make it conform to information models defined by GIMS, otherwise the registration will fail. RLDS itself incorporated several information providers to collect information about services and underlying resources, and it allows developers to develop other providers based on application or system requirements.

Information.Monitoring

This capability is provided by a monitoring service in CROWN. The monitoring service embraces a number of sensors that are responsible for monitoring resource status dynamically and reporting the status to RLDS periodically. The information provided currently includes both static (e.g. OS, disk size and CPU frequency) and dynamic (e.g. CPU load, available memory and available disk size) resource status. Users can access to monitoring information through RLDS interfaces.

There is also a statistic service (a base service in CROWN) developed to dynamically meter the throughput, average service response time and other performance data of service containers. This data will be integrated with RLDS in future.

Information.Provenance

The information logged by RLDS is stored in a MySQL database. Thus the information can be obtained at any time for the purpose of data mining, analysis and so on.

A.4.4 ExecMan

Execution management is carried out by CROWN scheduler in CROWN middleware. CROWN Scheduler is the execution management component in CROWN Grid. CROWN Scheduler can be divided into two parts, one is CROWN Global Scheduler and another is a CROWN Local Scheduler. CROWN Global Scheduler response for the meta-level schedule and global job management, CROWN Local Scheduler response for job execution and local job management

ExecMan.JobManager

When a job is submitted to the CROWN Scheduler, the Global Scheduler will generate a UUID for the job, which can unique identify the job. Then the statue of the job will become pending, and the statue of the job can be matchmaking, executing, error, finished and so on.

ExecMan.JobDescription

The job in CROWN Scheduler is described in JSDL (Job Submission and Description Language). JSDL is a draft specification of Open Grid Forum; it is the *de facto* specification for the Grid Job Execution Service. CROWN Scheduler also has some extension on JSDL to support Web Service-type job, PBS-type job and CROWN Credential Manager. Using JSDL as its job description language makes CROWN Scheduler have the interoperability with other Job Execution System, which also using JSDL in job description.

ExecMan.ExecutionAndPlanning

CROWN Scheduler can support different job types such as WebService-type job, PBS-type job and POSIX-type job. For the WebService-type job, CROWN Scheduler will behave as a Web Service invocation client to invoke the service which is deployed at the remote server. For the PBS-type job and POSIX-type job, CROWN Local Scheduler provides the interface to the back-end PBS system and POSIX system; it can generate the command which is needed for executing the jobs in the PBS system or POSIX system.

ExecMan.CandidateSetGenerator

CROWN Scheduler has defined a flexible interface for information service, which we called SPI. The default SPI is the interface for CROWN RLDS, according to the job execution requirement described in JSDL, CROWN Scheduler will query from RLDS to get the candidate resource set for job execution. Using the SPI defined in CROWN Scheduler, developers could also have CROWN Scheduler query candidate resource information from other information system (e.g., UDDI, Grimoires). While CROWN Scheduler got the candidate resource for the job execution from the information service or service registry, CROWN Scheduler will apply some filter and sort policy to the candidate resource set. Then CROWN Scheduler could select one or more resource in the candidate set to execution the jobs.

ExecMan.Reservation

CROWN Scheduler also provides the function for the resource reservation. CROWN Scheduler implemented a flexible capacity reservation mechanism, called FIRST, which employs the slack time-enabled request admission control with differentiated selection strategies.

A.5 VEGA-GOS

A.5.1 Security

Security.Authentication

In order to identify the user, each user of VEGA GOS can acquire a X.509 certificate that signed by CA trusted by GOS. In order to identify the service, each service that registered into GOS also needs to acquire a X.509 certificate from CA. VEGA GOS does not include a CA module. User and service provider need to apply and acquire certificates from a separated CA module.

The authentication of physical service provides function of user certificate and user proxy certificate authentication. The authentication of agora user's login provides function of uid/pass login authentication and of proxy certificate login authentication.

Security.Authorization

The Virtual service authorization provides function of service operation level authorization based on token (SAML document signed by agora authorization authority).

Security.Delegation

The proposed runtime construct called grip (grid process) [GOSARCH], which is an abstraction that corresponding to process in a traditional OS, and runs on VEGA GOS, representing a grid subject (grid user identified by DN) to access various services in grid. The grip can hold the user proxy certificate and sign the outgoing SOAP messages on the fly. At present, the grip presents five client side APIs, they are: create, bind, invoke, control and close. The developer can program the code with the five APIs simply and easily.

Security.CredentialStorage

The agora service can provide the function of credential storage, which is similar to myproxy. With the help of grip service, the grip mentioned above can be created using uid/pass pair.

Security.Accounting

Grid Batch Accounting System is constructed based on VEGA GOS v2 core function. Its main module includes batch log probe daemon, batch accounting service, batch accounting client side APIs and backend accounting database. Batch job user can query batch accounting information by the batch accounting service. When batch job user submit a job though the batch service, the batch service will record mapping of user identity (DN) to the local user name to the accounting database. After the job finished, the batch log probe daemon can collect accounting information from local batch system accounting log at intervals, and writes them to background accounting database.

A.5.2 Data

Data.Transfer

File upload/download by HTTP servlet which associated with the stand-alone file service. When client send a request to the file service for a flat file, the associated servlet will take over this request and send the file back though the HTTP protocol.

Data.Naming.Scheme

Grid File Management System [GOSIMPL] has a three-layer address space and provides file remote operation interfaces, but does not provide remote file accessing interface. The three layers of file in grid are physical file, virtual file and effective file respectively. Physical file is the grid file

available at physical file service, and physical file name is the absolute location of this file (/path/to/file or driver:\path\to\file); virtual file is the ID of virtualized physical file service, and virtual file name can uniquely determine the location of file in grid; effective file is the logical file space grid users can see (agora effective users). Users can arbitrarily organize file and directory structure in this space according to requirements. Each effective file of this space (implicitly including user ID namely user certificate DN) can be mapped to a virtual file. It is only necessary to see effective file view for end users without knowing which file server in which virtual file and physical file mapping this effective file store. Grid file system based on GOS allows developers to add functions provided by grid file system to grid applications by the use of GOS grid file system client side APIs.

Data.Naming.Resolve

The resolution of names is performed by the Grid File Management System.

A.5.3 Information

In this version of VEGA GOS, the information management named as grid router service. The address (URI) of service is registered and virtualized by grid router service. Each grid router service can manage the meta information of domain scoped services. By incrementally propagating the difference between neighbour routers, each router keeps the router ids (the virtualized router service access point which created automatically while the router start-ups at first time) globally synchronized, that is to say, the decentralized interlinked grid routers can provide a global unique (SSId) meta information space. The elements in this space are meta information of registered services which identified by combination of router id and service id.

Information.Discovery

Grid Router service provides users with the following function: (1) management of mapping from virtual service to physical serviceM; (2) global physical service locating

Information.Logging

By a uniform exception definition and the wrapper of SOAP fault, VEGA GOS can provide the capability of dynamically extending exception declarations and the capability of combining a group of exception information [EXCEPTION]. The developer of VEGA GOS can make use of this exception handling declare, analyze and process all kind of exceptions that may encountered, even throw a new exception to the up level processor. According to the emergency, the exception can be categorized as info, debug, important and vital; According to the occurred location (core, system, application level), the exception can also be categorized as core, system and application separately. Some useful information can be logged by a wrapper of Log4j. For example, the core and system level exceptions if important or vital are all logged into separate plain file for auditing. The format of log file is the same as log file generated by Log4j.

Information.Monitoring

The monitoring system in VEGA GOS can only monitor system level resources, such as cpu, mem, loadavg and hard disk capacity. It composed by monitoring data probe (ganglia or other cluster monitoring system), backend storage database, monitoring service and monitoring data aggregator. The original xml format monitoring data generated by ganglia can be collected into backend database. As the front-end, the monitoring service response to the data retrieval request and send back the data accordingly. The monitoring data aggregator can gather monitoring data from multiple monitoring services in centralized manner, and store these data into a individual database.

A.5.4 ExecMan

The execution management in VEGA GOS is called grid batch system [GOSIMPL]. The grid batch system consists of batch service, accounting service and batch driver for backend batch system (OpenPBS, LSF and so on.) interaction. The batch service has the interface of job submission, job status query, job cancellation or deletion. File stagein and stageout are supported by grid file management system.

ExecMan.BES

When a job description document is sent to batch service, the batch service will translate the request to job script and submit to backend batch system. The batch driver can provide the capability of running on behalf of other account that is because the underlying JVM cannot convert the running identity from one to another. If the request contains file stage-in or stage-out requirements, the batch service will download/upload the files from/to user global file space maintained by grid file management system. When a job finished, the batch service will write a record about computing and storage resource usage into accounting backend database for later retrieval.

There is no additional job scheduling functions in grid batch system. The job status is acquired by backend batch system, and will not be stored by grid batch system. These capabilities will be added to the new version of VEGA GOS prospectively.

ExecMan.JobDescription

The job description in grid batch system is quite simple. Only original job script, file stagein and stageout addresses are embedded into the xml schema. Anyway, the JSDL will be adapted in future version of VEGA GOS.

A.6 UNICORE 5

A.6.1 Security

The security architecture base upon standardized IETF X.509 certificates that are checked by the UNICORE Gateway component and the UNICORE User DataBase (UUDB).

Security.Authentication

The authentication of a user is performed by the UNICORE Gateway that provides the single point of entry to a UNICORE site (U-site). The X.509 certificates of the users are checked if they are valid (not expired), signed by a trusted Certificate Authority (CA) and not revoked via a Certificate Revocation List (CRL).

Security.CredentialStorage

The credentials are stored in Java keystores at the UNICORE client.

Security.Delegation

The secure delegation of jobs is possible by using the Explicit Trust Delegation (ETD). For more information, please refer to [ETD].

Security.Authorization

The authorization decision about a user is done at the UUDB by using the X.509 certificate of the user. However, the distinction of requested modes of access via combining different pieces of information is not supported in UNICORE 5.

Security.IdentityMapping

The authorization of a user is performed by the UNICORE User DataBase (UUDB) that provides mappings from X.509 certificates to dedicated user logins on supercomputers or clusters. Virtual Organization (VO) or group-wise logins are currently not wanted since supercomputing centres normally do not allow group accounts on their machine.

Security.Accounting

The accounting is left to the Resource Management System (RMS) that runs beneath the UNICORE stack.

A.6.2 Data

Data.Transfer

The handling of data is mainly done via the Network Job Supervisor (NJS) and the Target System Interface (TSI) component. For the transfer of data either, the UNICORE protocol layer (UPL) or GridFTP can be used. HTTP data transfer is also available in an alpha version.

Data.Management.Transfer

The managing of data transfers from the start to the completion is done at the NJS.

Data.Naming.Scheme

This capability is related to the capacity of attaching names to data resources. In particular, UNICORE uses a human-oriented name for a VSite which is mapped to a concrete address later.

Data.Naming.Resolver

This capability is related to the capacity of resolving one name to another (e.g., search the associated abstract name to a certain human-oriented name). In UNICORE, the human-oriented name is mapped to an address realized via the NJS and Gateway (connections file).

Data.Access.FlatFiles

The UNICORE TSI can access files on disk.

A.6.3 Information

Information.Model

The information model of UNICORE 5 is proprietary and typically provided by the UNICORE Incarnation DataBase (IDB).

Information.Discovery

The information & monitoring management is handled via the Network Job Supervisor (NJS) and the UNICORE Client provides job monitoring capabilities as well as the control of jobs.

Information.Logging

All the major components of UNICORE, namely the UNICORE Gateway, the Network Job Supervisor (NJS) and the Target System Interface (TSI), provide massive logging capabilities with different levels of verbosity. Additionally, the UNICORE Client also allows for logging.

Information.Monitoring

The information & monitoring management is handled via the Network Job Supervisor (NJS) and the UNICORE Client provides job monitoring capabilities as well as the control of jobs.

Information.Provenance

Long-term storage of information related to Grid activity is done at the USPACE of UNICORE. Note that only the one that creates the information (e.g. job outputs) is able to access this information.

A.6.4 ExecMan

ExecMan stands for Execution Management.

ExecMan.BES

The execution management is performed via the Network Job Supervisor (NJS), the core component of UNICORE that also provides massive workflow capabilities for job execution over different sites.

ExecMan.JobDescription

The Job description is done via either the UNICORE Client GUI or various command line interfaces. However, the internal representation is the Abstract Job Object (AJO), a proprietary java object analyzed at the NJS level.

ExecMan.JobManager**ExecMan.ExecutionAndPlanning**

Mappings between resources and jobs are done at the UNICORE client by the user. The scheduling of computational jobs is done via Resource Management Systems (RMS) that interacts with the UNICORE TSI.

ExecMan.CandidateSetGenerator

The resource is manually selected by users within the UNICORE Client.

ExecMan.Reservation

There is the possibility to reserve either bandwidth or computational time. However, both versions are just available as an alpha version.

UNICORE 6

A.6.5 Security

The security architecture base upon standardized IETF X.509 certificates that are checked by the new SOAP-aware UNICORE Gateway component and the Web Services-based UNICORE User DataBase (WS-UUDB).

Security.Authentication

The authentication of a user is performed by the new SOAP-aware UNICORE Gateway that provides the single point of entry to a UNICORE site (U-site). The X.509 certificates of the users are checked if they are valid (not expired), signed by a trusted Certificate Authority (CA) and not revoked via a Certificate Revocation List (CRL).

Security.CredentialStorage

The credentials are stored in Java keystores at the UNICORE client.

Security.Delegation

The secure delegation of jobs is possible by using the Explicit Trust Delegation (ETD). For more information please refer to [ETD].

Security.Authorization

The authorization decision about a user is done at the new WS-based UUDB by using the X.509 certificate of the user. However, the distinction of requested modes of access via combining different pieces of information is not supported in UNICORE 6 yet, but already planned by an integration of VOMS.

Security.IdentityMapping

The authorization of a user is performed by the WS-based UNICORE User DataBase (WS-UUDB) that provides mappings from X.509 certificates to dedicated user logins on supercomputers or clusters. Virtual Organization (VO) management is planned by the integration of VOMS.

Security.Accounting

The accounting is left to the Resource Management System (RMS) that runs beneath the UNICORE stack. However, a planned Resource Usage Service (RUS) that exposes Usage Records (UR) is already planned and will be developed for UNICORE 6 alpha soon. URs consist of accounting information gathered by the RMS. The specifications of RUS and UR are standardized by the Open Grid Forum (OGF).

A.6.6 Data

Data.Transfer

The handling of data transfers is done via the File Transfer Services that represent an abstract concept. In particular, a more specific file transfer is the RandomByteIO, StreamableByteIO transfer or the BaselineFileTransfer. It is planned to be OGSA-Data Movement Interface (DMI) compliant in future, a specification that will be starting to be standardized by the OGF soon.

Data.Management.Transfer

The managing of data transfers from the start to the completion is done via the corresponding File Transfer Service.

Data.Management.Storage

The Storage Management Service provides capabilities to manage a storage resource, from simple systems like disk-servers to complex hierarchical systems.

Data.Naming.Scheme

This capability is related to the capacity of attaching names to data resources. In particular, UNICORE uses no abstract or human readable names in the Alpha version, just addresses. However, in future it is planned to use the old naming scheme of UNICORE 5 or upcoming standard compliant specifications.

Data.Naming.Resolver

The name is resolved from the Registry Service, but only consist of the address (and port) of the corresponding service.

Data.Access.Relational

UNICORE 6 provides an alpha implementation of the OGSA-DAI specification that provides access to relational databases.

Data.Access.XML

UNICORE 6 provides an alpha implementation of the OGSA-DAI specification that provides access to XML-based databases.

Data.Access.FlatFiles

UNICORE 6 still relies on the Target System Interface (TSI) at the execution backend that provides access to files on disk.

A.6.7 Information

Information.Model

The information model of UNICORE 6 alpha is proprietary and typically base upon the exposure and management of WS-ResourceProperties.

Information.Discovery

The discovery of information can be done via simple WS-RF compliant WS-ResourceProperties message exchanges, targeting the Target System Service, Job Management Service, Storage Management Service or File Transfer Service. All these services expose their information as WS-ResourceProperties.

Information.Logging

All major components of UNICORE 6 provide logging features.

Information.Monitoring

The monitoring of information can be done via simple WS-RF compliant WS-ResourceProperties message exchanges, targeting the Target System Service, Job Management Service, Storage Management Service or File Transfer Service. All these services expose their information as WS-ResourceProperties. Furthermore, the GPE UNICORE Client allows for monitoring of submitted jobs and site status.

Information.Provenance

Long-term storage of information related to Grid activity is done at the USPACE of UNICORE. Note that only the one that creates the information (e.g. job outputs) is able to access this information.

A.6.8 ExecMan

ExecMan stands for Execution Management.

ExecMan.BES

The execution of a job is initiated via the Target System Service that can take JSDL job descriptions. The submission of jobs leads to the creation of a Job Resource that can be controlled via the Job Management Service. It is planned to move to the OGSA-BES specification of OGF in future.

ExecMan.JobDescription

The job description is OGF JSDL compliant.

ExecMan.ExecutionAndPlanning

Mappings between resources and jobs are done at the GPE UNICORE Client by the user. The scheduling of computational jobs is done via Resource Management Systems (RMS) that interacts with the UNICORE TSI at the backend.

ExecMan.CandidateSetGenerator

The resource is manually selected by users within the GPE UNICORE client, however first implementations of automatically selecting brokers are available, but still in alpha status.

ExecMan.Reservation

There is a WS-based prototype of a network reservation service available for UNICORE 6.

A.7 OMII-UK

A.7.1 Security

Security.Authentication

Incoming messages are verified that they have been digitally signed by a trusted CA using the relevant WS-Security profiles. Outgoing messages are signed using the same standard.

Security.CredentialStorage

Credentials are stored locally using the Java Keystore structure.

Security.Authorization

Access control can be centralized across a single or several containers using an Authorisation service implementing the SAML 1.1 assertion port type. The access request is evaluated using a policy and user attributes defined using PERMIS.

Security.AttributeAuthority

Currently we use PERMIS, but are considering how we can integrate with Shibboleth and VOMS as a source of attributes.

Security.IdentityMapping

The only service that currently executes service invocations as a specified user is the GridSAM job submission service which uses local 'gridmap' style mappings.

Security.Accounting

We will be integrating the Usage Records (UR) format with the Resource Usage Service (RUS) for storing accounting information.

A.7.2 Data

Data.Transfer

Currently GridSAM uses ftp and http. When we integrate a large scale data transfer mechanism it will be based around GridFTP.

Data.Management.Transfer

We are planning the integration of RFT or FTS as the web service interface to managing data transfers.

Data.Naming.Scheme

We will probably be examining the WS-Naming specification from OGF (OGSA).

Data.Naming.Resolver

This capability is related to the capacity of resolving one name to another (e.g., search the associated abstract name to a certain human-oriented name).

Data.Access.Relational & Data.Access.XML

Use of OGSA-DAI and the WS-DAI family of specifications.

A.7.3 Information

Information.Model

An object model based around either CIM or GLUE. Both show strong potential.

Information.Discovery

Use the UDDI compliant registry Grimoires to publish and search for information records, primarily services.

A.7.4 ExecMan

ExecMan.BES

We will be using the Basic Execution Service within the GridSAM service.

ExecMan.JobDescription

We are already using the Job Submission Description Language (JSDL).

ExecMan.JobManager

This is undertaken through the GridSAM web service.

ExecMan.ExecutionAndPlanning

This could also be considered as workflow where we use Taverna and BPEL. Taverna provides an environment for experimental service composition while the BPEL editor and execution engine provides the ability to provide pre-generated workflows.

ExecMan.CandidateSetGenerator

The KNOOGLE project will provide tools for service selection.

A.8 ARC

A.8.1 Security

In ARC, security services deal with authentication, authorization and eventually access control. A framework for auditing and accounting exists integrated into core services, however no dedicated services are provided for that. Security-related events are currently logged in server logs making possible grid-level tracking of incidents locally.

Security.Authentication

Authentication in ARC is done in accordance and using GSI components of the Globus Toolkit. Identity assertions are based on X.509v3 public key certificates, whilst the single sign-on relies on proxy certificates (RFC3820). Proxies are generated based on long-term user credentials.

Security.CredentialRetrieval

MyProxy is used and supported by the job execution component of ARC to renew client's credentials. This feature is implemented through MyProxy API.

Security.Delegation

Delegation in ARC is done via proxy certificates.

Security.AttributeAuthority

The ARC middleware includes the VOMS subsystem.

Security.Authorization

Authorization decisions in ARC are currently based on local system policies and typically are done on per-VO basis. The Authorization capabilities are integrated into all of the ARC components. ARC supports a variety of Authorization Sources such as VOMS, LDAP, HTTPS based user database. GACL is used for Access Control. The Grid Manager (GM) component of ARC enables flexible Authorization Decisions by providing a pluggable infrastructure for authorization plug-ins, in addition to the integrated authorization enforcement capabilities. In addition, data can be protected by using GACL technology. Most production storage elements in ARC are GACL-enabled.

Security.IdentityMapping

In ARC, mapping of Grid credentials into a local UNIX account is mostly required for job execution services because of the used local resource management systems. Most commonly used mechanism is that of grid-mapfiles, although support for LCMAPS and LCAS exist as well. Besides the aforementioned solutions, the ARC configuration enables fine-grained identity mapping for the core services. In ARC, it is also possible to separate Identity Mapping from Authorization.

Security.Accounting

ARC as such does not provide an accounting component, but has been interfaced to the SGAS (Swedish Grid Accounting System).

A.8.2 Data

Data.Transfer

Data transfer in ARC is mostly carried out via GridFTP, using the custom ARC server and clients. Transfer reliability is achieved by automatic retries and utilizing meta data such as checksums, timestamps. Data transfer via other protocols, such as HTTP(s) and FTP, is also supported.

Data.Management.Storage

Flat files are stored either in plain ARC GridFTP-based storage elements, or in so-called Smart Storage Elements. The latter are equipped with a SRM interface (v1.1 and some v2.1 features) and has a functionality of automatically registering itself and instantiated data in the data indexing system.

Data.Management.Transfer

ARC datamove module makes use of the Globus RLS and RC, gLite-LFC, gLite-Fireman, to index data and storage resources. They hold a limited amount of metadata.

Data.Naming.Resolver

Data location optimization is done by ARC services by using a combination of local to job execution services caches and the storage element records in the data indexing service.

A.8.3 Information

Information.Model

ARC resources are published in the custom designed ARC schema [ARC_schema], that includes a hierarchical manner such objects as clusters, storage elements, queues, jobs and authorized users. All objects with their attributes are published in local to each service LDAP databases.

Information.Discovery

Discovery service in ARC is based on LDAP technology, inspired by the Globus MDS2. It has the substantial difference of not caching the published information in indexing servers, keeping only pointers to local databases, such that the discovery procedure always involves every single Grid resource.

Information.Logging

In ARC, job activity logging can be configured on job execution service level and on individual job level. Job execution service would submit the logged job record to the chosen Logger database (SQL dump over SOAP). The information about each logged job will follow the Usage Record standard by OGF.

Information.Monitoring

Monitoring in ARC relies entirely on the Information system. Monitoring can be performed either via the ARC Grid Monitor (a Web tool), or via ARC User Interface. Both utilize LDAP to discover information.

A.8.4 ExecMan

ExecMan.JobMan

Job execution is done through ARC Grid Manager that accepts job requests via custom GridFTP server interface, formulates and prepares the task, and forwards it to the local resource management system. Grid Manager takes care of following up job execution, manipulating it upon user request (kill, retrieve, restart, renew credentials etc), staging necessary files and cleaning up after execution.

ExecMan.JobDescription

Job description is commonly done via extended Globus RSL 1.1 (xRSL), plus JSDL 1.0 is also supported with ARC extension, though it currently provides much less capabilities.

ExecMan.JobManager

The higher level service that encapsulates all the aspects of executing a job or a set of jobs is provided by either of the several ARC components: UserInterface, ARC portal or the Arconaut GUI.

ExecMan.ExecutionAndPlanning

Job execution planing is performed by the User Interface.

ExecMan.CandidateSetGenerator

Job candidate set generation is performed by the User Interface.

B Acknowledgements

This deliverable was possible thanks to worthy feedbacks from many people, in particular we wish to thank Andrea Ferraro and Valerio Venturi for the feedback on the security aspects, Riccardo Zappi for the worthy discussions on the data management capability decomposition and Claudio Grandi and John White for insight on the gLite middleware. We also thank the reviewers for their valuable comments about this document.

C References

- [AUTHZFRAM] Chadwick D. *Functional Components of Grid Service Provider Authorisation Service Middleware*. OGSA AUTHZ WG
- [ARC_schema] B. Kónya, The NorduGrid/ARC Information System, The NorduGrid Collaboration, NORDUGRID-TECH-4, http://www.nordugrid.org/documents/arc_infosys.pdf.
- [ARCPAPER] "Advanced Resource Connector middleware for lightweight computational Grids". M.Ellert et al., *Future Generation Computer Systems* 23 (2007) 219-240, (<http://dx.doi.org/10.1016/j.future.2006.05.008>)
- [BLAHP] Molinari E., Prelz F., Rebato D. et Al. A local batch system abstraction layer for global use. In *Proceedings of the International Conference on Computing in High Energy and Nuclear Physics (CHEP2006)*, Mumbai, India, Feb 2006.
- [BESWG] OGSA Basic Execution Services WG. <http://forge.ggf.org/projects/ogsa-bes-wg>
- [CERN] European Organization for Nuclear Research (CERN). <http://www.cern.ch>
- [CLASSAD] Solomon M.. *The ClassAd Language Reference Manual, Version 2.1*. Computer Sciences Department, University of Wisconsin, Madison, WI, USA, Oct 2003.
- [CEMON] gLite CEMon website. <http://grid.pd.infn.it/ceмон/field.php?n=Main.AboutCEMon>
- [CASTOR] Castor. <http://cern.ch/castor/>
- [CREAM] Computing Resource Execution And Management (CREAM). <http://grid.pd.infn.it/cream/>.
- [DCACHE] dCache Website. <http://www.dcache.org/>
- [DWG] OGF OGSA Data Working Group. <http://forge.ggf.org/sf/projects/ogsa-d-wg>
- [DPM] Disk Pool Manager. <https://uimon.cern.ch/twiki/bin/view/LCG/DpmGeneralDescription>
- [DMI] OGF OGSA Data Movement Interface. <http://forge.ggf.org/sf/projects/ogsa-dmi-wg>
- [EGEE] European Grid for E-science, INFOS-RI-031688, <http://www.eu-egee.org>
- [EGEERCH] JRA1 Design Team. *EGEE Middleware Architecture*. EU Deliverable DJRA1.4 <https://edms.cern.ch/document/594698/>
- [EGEESHIB] gLite Shibboleth interoperability through dedicated SICS. <https://edms.cern.ch/document/770102/1>

[ETD] Explicit Trust Delegation : Security for dynamic Grids:
www.unigrids.org/papers/explicittrust.pdf

[EXCEPTION] Qiang Yue, Hao Wang, Li Zha, et. al., An Approach to Exception Handling for Service-Oriented Systems, in Proceedings of ICWS'06, Chicago, US, 2006.

[EUPROV] EU Provenance Project. <http://gridprovenance.org>

[GACL] A. McNab, The gridsite web/grid security system: research articles, Softw.

Pract. Exper. 35 (9) (2005) 827–834.

[GLITECE] gLite Computing Element.

[GLITEPROG] Laure E. Programming the Grid with gLite. EGEE Technical Report EGEE-TR-2006-001. <http://documents.cern.ch/cgi-bin/setlink?base=egee&categ=tr&id=egee-tr-2006-001>

[GLOBUSAUTHZ] Lang B., Foster I., Siebenlist F., Ananthkrishnan R., Freeman T. *A Multipolicy Authorization Framework for Grid Security*.
http://www.globus.org/alliance/publications/papers/IEEE_NCA_AGC.pdf

[GLUESHEMA] Andreozzi S., Burke S., Field F., Fisher S., Konya B., Mambelli M., Schopf J., Viljoen M., and Wilson A.. GLUE Schema Specification - Version 1.2, Dec 2005.
http://glueschema.forge.cnaf.infn.it/uploads/Spec/GLUEInfoModel_1_2_final.pdf

[GMA] Tierney B., Aydt R., Gunter D., Smith W., Swany M., Taylor V., Wolski R. A Grid Monitoring Architecture. GFD.7 Open Grid Forum document.
<http://www.gridforum.org/documents/GFD.7.pdf>

[GOSARCH] Li Zha, Wei Li, et. al., System Software for China National Grid, IFIP International Conference on Network and Parallel Computing (NPC 2005), LNCS 3779, pp. 14~21, Beijing, China, 2005.

[GOSIMPL] X. Xie, N. Xiao, Z. Xu, L. Zha, et. al., CNGrid Software 2: Service Oriented Approach to Grid Computing, in Electronic Proceedings of AHM'05, Nottingham, UK, 2005.

[G-PBOX] Grid Policy Box. <http://gpbox.forge.cnaf.infn.it/>

[GRIDFTP] Mandrichenko I., Allcock W., Perelmutov T. GridFTP v2 Protocol Description. Open Grid Forum GFD.47 document

[GRIDICE] GridICE – website. <http://grid.infn.it/gridice>

[GRIDSHIB] GridShib. <http://gridshib.globus.org/>

[GRIDSITE] GridSite. <http://www.gridsite.org/>

[GRAM2] Globus Grid Resource Allocation Management 2.x.
<http://www.globus.org/toolkit/gram/#gramgt2>

[GSA] Grid Scheduling Architecture Research Group (GSA-RG). Open Grid Forum.
<https://forge.gridforum.org/sf/projects/gsa-rg>

[GSM] Grid Storage Manager Working Group (GSM-WG). Open Grid Forum
<http://forge.ggf.org/projects/gsm-wg>

[GPBOX] Grid Policy Box (G-PBox). <http://gpbox.forge.cnaf.infn.it/>

[GT4SEC] Globus Toolkit Version 4 Grid Security Infrastructure: A Standards Perspective. The Globus Security Team. <http://www.globus.org/toolkit/docs/4.0/security/GT4-GSI-Overview.pdf>

[KLP] University of Michigan, Center for Information Technology Integration. Kerberos Leveraged PKI. http://www.citi.umich.edu/projects/kerb_pki/

[KNOWARC] Grid-enabled Know-how Sharing Technology Based on ARC Services and Open Standards (KnowARC) project, <http://www.knowarc.eu>

[JDL] Job Description Language (JDL) Attributes Specification (Submission through the WMPProxy Service). <https://edms.cern.ch/document/590869/1>

[JP] gLite Job Provenance. <http://egee.cesnet.cz/en/JRA1/index.html>

[ICTP] International Center for Theoretical Physics. <http://www.ictp.it/>

[INFN] Istituto Nazionale di Fisica Nucleare. <http://www.infn.it>

[LB] gLite Logging and Bookkeeping Service. <http://egee.cesnet.cz/en/JRA1/index.html>

[LCAS] Local Centre Authorization Service (LCAS). <http://www.nikhef.nl/grid/lcaslcmaps/>

[LCG] LHC Computing Grid. <http://www.cern.ch/lcg>

[LCMAPS] Local Credential Mapping Service (LCMAPS). <http://www.nikhef.nl/grid/lcaslcmaps/>

[LDAP] Wahl M., Howes T., and Kille, S.. Lightweight Directory Access Protocol (v3). IETF RFC 2251, Dec 1997.

[MDS2] Globus Monitoring and Discovery Service 2.x. http://www.globus.org/toolkit/mds/#mds_gt2

[MYPROXY] MyProxy Credential Management Service. <http://grid.ncsa.uiuc.edu/myproxy/>

[NORDUGRID] The NorduGrid Collaboration, <http://www.nordugrid.org>

[NDGF] Nordic Data Grid Facility (NDGF), <http://www.ndgf.org>

[OGSA] Foster I., Kishimoto H., Savva A., Berry D., Djaoui A., Grimshaw A., Horn B., Maciel F., Siebenlist F., Subramaniam R., Treadwell J., and Von Reich J. *The Open Grid Services Architecture, Version 1.5*. OGS Draft 1.5-011.

[OGSADATAARCH] Berry D., Luniewski E., Antonioletti M., Chervenak A., Kunszt P., Laws S., Morgan M. *OGSA Data Architecture*. OGF OGSA Data WG Draft. Version 0.6.2. 29 Jun 2006.

[OGSARSS] OGSA Resource Selection Services: Specification. Draft 7 Feb 2006. <https://forge.gridforum.org/sf/go/doc13767?nav=1>

[OGSASAML] Welch, V., et al., Use of SAML for OGSA Authorization, www.globus.org/ogsa/security

[PKI] Information Technology - Open Systems Interconnection - The Directory: Authentication Framework. ITU-T Recommendation X.509 (1997 E). June 1997.

[RFC3820] Tuecke S., Welch V., Engert D., Pearlman L., Thompson M. *Internet X.509 Public Key Infrastructure (PKI) Proxy Certificate Profile*. IETF RFC 3280. Jun 2004

[RGMA] Relational Grid Monitoring Architecture. <http://www.r-gma.org/>

[SAML] OASIS Security Services (SAML) TC. http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security

[SDAPI] Service Discovery Interface API. <http://hepunix.rl.ac.uk/egee/jra1-uk/sd/>

[SGAS] SweGrid Accounting System. <http://www.sgas.se/>

[SHIB] Shibboleth. <http://shibboleth.internet2.edu/>

[SOARM] Reference Model for Service Oriented Architecture V1.0. OASIS Committee Specification 1, 2 Aug 2006. <http://www.oasis-open.org/committees/download.php/19679/soa-rm-cs.pdf>

[SRM2.2] Shoshani A., Sim A., et Al. *Storage Resource Manager Interface Specification* version.2.2. <http://sdm.lbl.gov/srm-wg/doc/SRM.v2.2.html>

[STORM] STORage Resource Manager. <http://storm.forge.cnaf.infn.it/>

[TA] OMII-Europe Annex I – “Description of Work”, 21 Mar 2006

[VOMS] Virtual Organization Membership Service (VOMS). <http://voms.forge.cnaf.infn.it/>

[XACML] OASIS eXtensible Access Control Markup Language (XACML) TC. <http://www.oasis-open.org/committees/xacml/>

[WSA] Web Services Architecture. W3C Working Group Note 11 Feb 2004. <http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/>

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