ETSI TR 102 766 V1.1.1 (2009-10)

Technical Report

GRID;
ICT Grid Interoperability Testing Framework
and survey of existing ICT Grid interoperability solutions

Reference
DTR/GRID-0002

Keywords
analysis, directory, interoperability, methodology, testing

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Important notice

Individual copies of the present document can be downloaded from:
http://www.etsi.org

The present document may be made available in more than one electronic version or in print. In any case of existing or perceived difference in contents between such versions, the reference version is the Portable Document Format (PDF). In case of dispute, the reference shall be the printing on ETSI printers of the PDF version kept on a specific network drive within ETSI Secretariat.

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at
http://portal.etsi.org/tb/status/status.asp

If you find errors in the present document, please send your comment to one of the following services:
http://portal.etsi.org/chaircor/ETSI_support.asp

Contents

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (http://webapp.etsi.org/IPR/home.asp).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Foreword

This Technical Report (TR) has been produced by ETSI Technical Committee GRID (GRID).

1 Scope

The present document describes a grid testing framework based on existing testing and validation methodologies, best practices and tools used, in IT and Telecom sectors to obtain ICT Interoperability. It lists and compiles existing grid interoperability solutions including interoperability events, state of the Art papers, guidelines, interoperability profiles, reference implementations, use cases, test suites, test beds, testing tools, and open source developments.

2 References

References are either specific (identified by date of publication and/or edition number or version number) or non-specific.

- For a specific reference, subsequent revisions do not apply.
- Non-specific reference may be made only to a complete document or a part thereof and only in the following cases:
  - if it is accepted that it will be possible to use all future changes of the referenced document for the purposes of the referring document;
  - for informative references.

Referenced documents which are not found to be publicly available in the expected location might be found at http://docbox.etsi.org/Reference.

NOTE: While any hyperlinks included in this clause were valid at the time of publication ETSI cannot guarantee their long term validity.

2.1 Normative references

The following referenced documents are indispensable for the application of the present document. For dated references, only the edition cited applies. For non-specific references, the latest edition of the referenced document (including any amendments) applies.

Not applicable.

2.2 Informative references

The following referenced documents are not essential to the use of the present document but they assist the user with regard to a particular subject area**.** For non-specific references, the latest version of the referenced document (including any amendments) applies.

[i.1] Wang, Y., Scardaci, D., Yan, B., Huang, Y.: "Interconnect EGEE and CNGRID e-Infrastructures through Interoperability between gLite and GOS Middlewares". In: e-Science and Grid Computing, IEEE International Conference on 10-13 Dec., pp. 553-560. IEEE Computer Society (2007).

[i.2] Marzolla, M., Andreetto, P., Venturi, V., Ferraro, A., Memon, A.S., Memon, M.S., Tweddell, B., Riedel, M., Mallmann, D., Streit, A., van den Berghe, S., Li, V., Snelling, D.F., Stamou, K., Shah, Z.A., Hedman, F.: "Open Standards-Based Interoperability of Job Submission and Management Interfaces across the Grid Middleware Platforms gLite and UNICORE". In: e-Science and Grid Computing, IEEE International Conference on 10-13 Dec., pp. 592-601. IEEE Computer Society (2007).

[i.3] Riedel, M.: "Fact-Shet WISDOM Interoperability Scenario". Open middleware infrastructure institute. (2008).

[i.4] Riedel, M.: "e-Infrastructure Interoperability between Europe and India. Open middleware infrastructure institute". Presentation. OGF23. Barcelona, Spain. (2008).

[i.5] Field, L, Schulz, M.: Grid interoperability: "The interoperations cookbook". J. Phys.: Conf. Ser. 119 012001 (8pp) (2008).

[i.6] Stephen Brewer, Achim Streit, Alistair Dunlop: "Building Components for Grid Interoperability". OMII Europe (2008).

[i.7] OMII-Europe: "D:JRA2.0 Report on Grid Activities relevant to the identification of new services". Technical report (2006).

[i.8] S. Verma, M. Parashar, J. Gawor und G. von Laszewski: "Design and Implementation of a CORBA Commodity Grid Kit". Lecture Notes in Computer Science, Springer, Volume 2242/2001.

[i.9] M. Parashar, G. von Laszewski, S. Verma, J. Gawor, K. Keahey, N. Rehn: "A CORBA Commodity Grid Kit". Concurrency and Computation: Practice and Experience, Special Issue: Grid Computing Environments. Volume 14 Issue 13-15, John Wiley & Sons, Ltd., Jan. 2003, Pages 1057 - 1074.

[i.10] Simone Campana, Dario Barberis, Frederic Brochu1, Alessandro De Salvo, Flavia Donno, Luc Goossens, Santiago González de la Hoz, Tommaso Lari, Dietrich Liko, Julio Lozano, Guido Negri, Laura Perini, Gilbert Poulard, Silvia Resconi, David Rebatto, Luca Vaccarossa: "Analysis of the ATLAS Rome Production Experience on the LHC Computing Grid". First International Conference on e-Science and Grid Computing, Volume, Issue, 5-8 Dec. 2005 Page(s): 8 pp.

[i.11] Web Pages of the KnowARC project.

NOTE: Available at http://www.knowarc.eu/about.html (last visited: 9.12.08).

[i.12] KnowARC project deliverable: "D3.1-1 Interoperability Minimal Service Survey". March 2007.

NOTE: Available at http://www.knowarc.eu/documents/Knowarc_D3.1-1_07.pdf (last visited: 9.12.08).

[i.13] KnowARC project deliverable: "D3.3-1 KnowARC Standards Conformance Roadmap (2nd release)". May 2008.

NOTE: Available at: http://www.knowarc.eu/documents/Knowarc_D3.3-1_08.pdf (last visited: 9.12.08).

[i.14] KnowARC project deliverable: "D3.2-1 KnowARC-EGEE Middleware Gateways". August 2008.

NOTE: Available at: http://www.knowarc.eu/documents/Knowarc_D3.2-1_08.pdf (last visited: 9.12.08).

[i.15] KnowARC Milestone report: "M3.2 Cross-Grid Tasks execution is performed via the ARC-EGEE Gateway". September 2007.

NOTE: Available at: http://www.knowarc.eu/documents/Knowarc_M3.2_07.pdf (last visited: 9.12.08).

[i.16] ATLAS Computing Group: "ATLAS Computing - Technical Design Report". ATLAS TDR-017, CERN-LHCC-2005-022. June 2005.

[i.17] The LCG Editorial Board: "LHC Computing Grid Technical Design Report", LCG--TDR--001, CERNLHCC-2005-24, June 2005.

[i.18] Open Science Grid.

NOTE: Available at http://www.opensciencegrid.org/ fetched on 12/10/2008.

[i.19] Nordic Data Grid Facility.

NOTE: Available at http://www.ndgf.org/ fetched on 12/10/2008.

[i.20] A. De Salvo, G. Negri, D. Rebatto, L. Vaccarossa: "LEXOR, the LCG-2 Executor for the ATLAS DC2 Production System", Conference on Computing in High Energy and Nuclear Physics (CHEP04), September 2004, Interlaken (Switzerland).

[i.21] M. Branco: "Don Quijote - Data Management for the ATLAS Automatic Production System", Conference on Computing in High Energy and Nuclear Physics (CHEP04), September 2004, Interlaken (Switzerland).

[i.22] EGEE-II Collaboration: "The Gateway Approach Providing EGEE/gLite Access to Non-Standard Architectures". September 2007.

NOTE: Available at http://cdsweb.cern.ch/record/1057547/files/egee-tr-2007-001.pdf (last visited: 10.12.2008).

[i.23] Christine Hung, Ravi Madduri, Kiran Keshav, Scott Oster, Stephen Langella, Stuart Martin, Stephen Mock: "Integrating caGrid and TeraGrid". Proceeding of The 3rd annual TeraGrid Conference, TeraGrid '08 (2oo8).

[i.24] Nancy Wilkins-Diehr, Dennis Gannon, Gerhard Klimeck, Scott Oster, Sudhakar Pamidighantam: "TeraGrid Science Gateways and Their Impact on Science". In IEEE Computer. Volume: 41, Issue: 11 On page(s): 32-41. November 2008.

[i.25] caGrid.

NOTE: Available at http://www.cagrid.org fetched online on 12/11/2008.

[i.26] TeraGrid.

NOTE: Available at http://www.teragrid.org/ fetched online on 12/11/2008.

[i.27] Graeme A. Stewart, David Cameron, Greig A. Cowan, Gavin McCance: "Storage and Data Management in EGEE". Australasian Symposium on Grid computing in Research and Practice in Information Technology. Volume 68. Australian Computer Society Inc. (2007).

[i.28] Rings, T., Neukirchen, H., Grabowski, J.: "Testing Grid Application Workflows Using TTCN-3". In: International Conference on Software Testing Verification and Validation (ICST), pp. 210{219. IEEE Computer Society (2008).

[i.29] ISO/IEC 9646 (1994-1997): "Information Technology - Open Systems Interconnection - Conformance testing methodology and framework".

[i.30] ETSI ES 201 873 (parts 1 to 8): "Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3".

[i.31] Grabowski, J., Hogrefe, D., Rethy, G., Schieferdecker, I., Wiles, A., Willcock, C.: "An introduction to the testing and test control notation (TTCN-3)". Computer Networks 42(3), 375{403 (2003). DOI: 10.1016/S1389-1286(03)00249-4.

[i.32] eInfrastructure for Testing, Integration and Configuration of Software (ETICS).

NOTE: Available at http://www.eu-etics.org fetched on 11/06/08.

[i.33] Morgan, M.: "ByteIO OGSA WSRF Basic Profile Rendering 1.0". Open Grid Forum (2006).

[i.34] Nakada, H., Matsuoka, S., Seymour, K., Dongarra, J., Lee, C., Casanova, H.: GridRPC: "A Remote Procedure Call API for Grid Computing". Advanced Programming Models Research Group, GWD-I (Informational) (2002).

[i.35] A. Duarte, G. Wagner, F. Brasileiro, and W. Cirne: "Multi-Environment Software Testing on the Grid". In PADTADIV: Proceedings of the 2006 workshop on Parallel and Distributed Systems: Testing and Debugging, pages 61-68. ACM Press (2006).

[i.36] E. Gamma and K. Beck. JUnit.

NOTE: Available at http://junit.sourceforge.net fetched on 12/04/2008.

[i.37] Centre for Development of Advanced Computing: "CDAC Grid Computing Test Suites and Grid Probes".

NOTE: Available at http://www.cdac.in/HTML/npsf/gridcomputing/npsfgrid.asp fetched on 12/04/2008.

[i.38] GridRPC Working Group: "Interoperability Testing for the GridRPC API Specification". Grid Working Draft for Information (GWD-I) Open Grid Forum (OGF), November 2006.

NOTE: Available at http://www.ogf.org/Public_Comment_Docs/Documents/Nov-2006/gridrpc-interoperability-1103.pdf (last visited: 10.12.2008).

[i.39] Hidemoto Nakada, Hitoshi Sato, Kazushige Saga, Masayuki Hatanaka, Yuji Saeki, Satoshi Matsuoka: "Job invocation interoperability between NAREGI Middleware Beta and gLite". HPC Asia, 2007.

[i.40] S. Newhouse, A. Grimshaw: "Independent Software Vendors (ISV) Remote Computing Usage Primer". Technical Report. GFD-I.141 Open Grid Forum (2008).

[i.41] ETSI EG 202 237 (V1.1.2): "Methods for Testing and Specification (MTS); Internet Protocol Testing (IPT); Generic approach to interoperability testing".

[i.42] "Basic Features of the Grid Component Model", Technical Report, INRIA, 2007.

[i.43] "The Fractal Component Model, ObjectWeb Consortium", Specification, 2004.

[i.44] Basukoski, Artie and Buhler, Peter and Getov, Vladimir and Isaiadis, Stavros and Weigold, Thomas: "Methodology for component-based development of grid applications". In: Compframe/Hpc-Geco Workshop: Proceedings of the 2008 compFrame/HPC-GECO workshop on Component based high performance. ACM, New York, NY, USA. ISBN 9781605583112.

[i.45] ETSI TS 102 827 (V1.1.1): "GRID; Grid Component Model (GCM); GCM Interoperability Deployment".

[i.46] ETSI TS 102 828 (V1.1.1): "GRID; Grid Component Model (GCM); GCM Application Description".

[i.47] ETSI TS 102 829 (V1.1.1): "GRID; Grid Component Model (GCM);GCM Fractal Architecture Description Language (ADL)".

[i.48] Schulz et al, "Grid deployment experiences: the interoperability experiences between OSG and LCG", CHEP2006, Feb 2006.

NOTE: Available at http://osg-docdb.opensciencegrid.org/cgi-bin/ShowDocument?docid=370.

[i.49] Riedel et al, "Interoperation Scenarios of Production e-Science Infrastructures", e-Science 2007, Nov 2007.

NOTE: Available at http://forge.gridforum.org/sf/docman/do/downloadDocument/projects.gin/docman.root.publications.e_science_2007_ogf_workshop/doc14928.

[i.50] Riedel et al, "Experiences and Requirements for Interoperability between HTC- and HPC-driven e-Science Infrastructures".

NOTE: Available at http://www.fz-juelich.de/jsc/datapool/KojakPubs/Riedel_RequirementsInterop.pdf.

[i.51] Crouch et al, "Towards Grid Interoperability", OMII-UK, UK e-Science All Hands Meeting, September 2007.

NOTE: Available at http://www.omii.ac.uk/attach/Presentations/781.pdf.

[i.52] "ETICS User Manual", CERN EDMS 795312 v2.0.1, November 2007.

NOTE: Available at https://edms.cern.ch/file/795312/2.0.1/ETICS-User_Manual-latest.pdf.

[i.53] Object Management Group (OMG), "Common Object Request Broker Architecture: Core Specification", March 2004, Version 3.0.3.

NOTE: Available at http://www.omg.org/docs/formal/04-03-12.pdf.

[i.54] IBM: "Workload Management with LoadLeveler", IBM Redbooks.

NOTE: Available at http://www.redbooks.ibm.com/abstracts/sg246038.html.

[i.55] "Platform, Manage and accelerate compute- or data-intensive workload on HPC clusters and grids".

NOTE: Available at http://www.platform.com/Products/platform-lsf.

[i.56] Microsoft: "Microsoft Windows Compute Cluster Server 2003 (CCS)".

NOTE: Available at http://www.microsoft.com/windowsserver2003/ccs/default.mspx.

[i.57] PBS Gridworks, OpenPBS.

NOTE: Available at http://www.pbsgridworks.com/PBSTemp1.3.aspx?top_nav_name=Products&item_name=OpenPBS&top_nav_str=1&AspxAutoDetectCookieSupport=1.

[i.58] Sun, Sun Grid Engine.

NOTE: Available at http://www.sun.com/software/sge/.

[i.59] GLite.

NOTE: Available at http://glite.web.cern.ch/glite/.

[i.60] Gridsystems. Fura.

NOTE: Available at http://www.gridsystems.com/.

[i.61] Foster, I.: "Globus Toolkit Version 4: Software for Service-Oriented Systems". In: H. Jin, D.A. Reed, W. Jiang (eds.) NPC, Lecture Notes in Computer Science, vol. 3779, pp. 2-13. Springer (2005).

[i.62] Streit, A., Erwin, D., Lippert, T., Mallmann, D., Menday, R., Rambadt, M., Riedel, M., Romberg, M., Schuller, B., Wieder, P.: "UNICORE - From Project Results to Production Grids". CoRR (2005).

[i.63] China National Grid.

NOTE: Available at http://www.cngrid.org.

[i.64] VEGA GOS.

NOTE: Available at http://vega.ict.ac.cn/gos/en/index_en.htm.

[i.65] Foster, I., Grimshaw, A., Lane, P., Lee, W., Morgan, M., Newhouse, S., Pickles, S., Pulsipher, D., Smith, C., Theimer, M.: "OGSA Basic Execution Service Version 1.0, GFD-R.108". Open Grid Forum (2008).

[i.66] Anjomshoaa, A., Brisard, F., Drescher, M., Fellows, D., Ly, A., McGough, S., Pulsipher, D., Savva, A.: "Job Submission Description Language (JSDL) Specification, Version 1.0, GFD-R.136". Open Grid Forum (2008).

[i.67] Scott Cantor and John Kemp and Rob Philpott and Eve Maler: "Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML)" V2.0, 2005.

NOTE: Available at http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf, OASIS Standard.

[i.68] J. Treadwell: "Open Grid Services Architecture Glossary of Terms Version 1.6". GFD-I.120, 2007, Open Grid Forum.

[i.69] National Research Grid Initiative: "NAREGI middleware version 1.1.1".

NOTE: Available at http://middleware.naregi.org fetched on 12/16/08.

[i.70] Catlett, C.: "Global Grid Forum Documents and Recommendations: Process and Requirements, GFD-C.1". Open Grid Forum (2002).

[i.71] ETSI ES 202 553: "Methods for Testing and Specification (MTS); TPLan: A notation for expressing Test Purposes".

[i.72] GFD.136 Anjomshoaa, A. et. al.: "Job Submission Description Language (JSDL) Specification", Version 1.0, Open Grid Forum, 2008.

NOTE: Available at http://www.ogf.org/documents/GFD.136.pdf fetched on 01-10-2009.

[i.73] GFD.133 Rajic, H. et. al.: "Distributed Resource Management Application API Specification 1.0", Open Grid Forum, 2008.

NOTE: Available at http://www.ogf.org/documents/GFD.133.pdf fetched on 01-10-2009.

[i.74] GFD.114 Dillaway, B. et. al.: "HPC Basic Profile", Version 1.0, Open Grid Forum, 2007.

NOTE: Available at http://www.ogf.org/documents/GFD.114.pdf fetched on 01-10-2009.

[i.75] GFD.111 Humphrey, M. et. al.: "JSDL HPC Profile Application Extension", Version 1.0, Open Grid Forum, 2007.

NOTE: Available at http://www.ogf.org/documents/GFD.111.pdf fetched on 01-10-2009.

[i.76] GFD.108 Foster, I. et. al.: "OGSA Basic Execution Service Version 1.0", Open Grid Forum, 2008.

NOTE: Available at http://www.ogf.org/documents/GFD.108.pdf fetched on 01-10-2009.

[i.77] GFD.52 Nakada, H. et. al.: "A GridRPC Model and API for End-User Applications", Open Grid Forum, 2007.

NOTE: Available at http://www.ogf.org/documents/GFD.52.pdf fetched on 01-10-2009.

[i.78] GFD.140 McGough, A. S., Savva, A.: "Implementation and Interoperability Experiences with the Job Submission Description Language(JSDL) 1.0", Open Grid Forum, 2008.

NOTE: Available at http://www.ogf.org/documents/GFD.140.pdf fetched on 01-10-2009.

[i.79] GFD.124 Wasson, G.: "Interoperability Experiences with the High Performance Computing Basic Profile (HPCBP)", Version 1.0, Open Grid Forum, 2008.

NOTE: Available at http://www.ogf.org/documents/GFD.124.pdf fetched on 01-10-2009.

[i.80] GFD.117 Ciesnik, L. et. al.: "PBS/Torque DRMAA 1.0 Implementation - Experience Report", Open Grid Forum, 2007.

NOTE: Available at http://www.ogf.org/documents/GFD.117.pdf fetched on 01-10-2009.

[i.81] GFD.103 Tröger, P., Gietzel, B.: "Condor DRMAA 1.0 Implementation - Experience Report", Open Grid Forum, 2007.

NOTE: Available at http://www.ogf.org/documents/GFD.103.pdf fetched on 01-10-2009.

[i.82] GFD.104 Herrera, H. et. al.: "GridWay DRMAA 1.0 Implementation - Experience Report", Open Grid Forum, 2007.

NOTE: Available at http://www.ogf.org/documents/GFD.104.pdf fetched on 01-10-2009.

[i.83] GFD.105 Templeton, D., Haas, A.: "N1™ Grid Engine DRMAA 1.0 Implementation - Experience Report", Open Grid Forum, 2007.

NOTE: Available at http://www.ogf.org/documents/GFD.105.pdf fetched on 01-10-2009.

[i.84] GFD.102 Tanimura, Y. et. al.: "Interoperability Testing for The GridRPC API Specification", Open Grid Forum, 2007.

NOTE: Available at http://www.ogf.org/documents/GFD.102.pdf fetched on 01-10-2009.

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the following terms and definitions apply:

Virtual Organization (VO): comprises a set of individuals and/or institutions having direct access to computers, software, data, and other resources for collaborative problem-solving or other purposes

NOTE: VOs are a concept that supplies a context for operation of a Grid that can be used to associate users, their requests, and a set of resources. The sharing of resources in a VO is necessarily highly controlled, with resource providers and consumers defining clearly and carefully just what is shared, who is allowed to share, and the conditions under which sharing occurs [i.68].

deployment manager: entity which converts deployment information into infrastructure specific service calls or commands to perform resource reservation and application deployment

3.2 Abbreviations

For the purposes of the present document, the following abbreviations apply:

AD	GCM Application Descriptor
ADL	Architecture Description Language
AFS	Andrew Filesystem
AJO	Abstract Job Object
APEL	Accounting Processor for Event Logs
API	Application Programming Interface
ARC	Advanced Resource Connector
BDII	Berkeley Database Information Index
BES	Basic Execution Service
BSP	Basic Security Profile
CAS	Community Authorization Service
CCS	Compute Cluster Server
C-DAC	Centre for Development of Advanced Computing
CMS	Compact Muon Solenoid experiment
CORBA	Common Object Request Broker Architecture
CSF	Community Scheduler Framework
DAI	Data Access and Integration
DAIR	Data Access and Integration - The Relational Realisation
DAIX	Data Access and Integration – The XML Realization
DD	GCM Deployment Descriptor
DRMAA	Distributed Resource Management Application API
DRS	Data Replication Service
EGA	Enterprise Grid Aliance
EGEE	Enabling Grids for E-sciencE
ENEA	Energy and Sustainable Economic Development
ETICS	eInfrastructure for Testing, Integration and Configuration of Software
FTP	File Transfer Protocol
FTS	File Transfer Service
GACL	Generic Access Control Lists
GCM	Grid Component Model
GFAL	Grid File Access Library
GIN	Grid Interoperation Now
GIOP	General InterORB Protocol

GOS	Grid Operation System
GRAM	Grid Resource and Allocation Manager
GRIA	Grid Resources for Industrial Applications
GSI	Grid Security Infrastructure
GUG	Grid Underground
HPC	High Performance Computing
HPCBP	High Performance Computing Basic Profile
ICS	Implementation Conformance Statement
ICT	Information and Communication Technology
IIOP	Internet Inter-ORB Protocol
IPT	Internet Protocol Testing
ISV	Independent Software Vendor
JAF	JavaServer Faces
JDL	Job Description Language
JMS	Job Management Service
JSDL	Job Submission Description Language
LCG	Large Hadron Collider
LCMAPS	Local Credential MAPPING Service
LDAP	Lightweight Directory Access Protocol
LFN	Logical File Name
LHC	Large Hadron Collider
LSF	Load Sharing Facility
MDS	Monitoring and Discovery System
NAREGI	National Research Grid Initiative
NDA	Non-Disclosure Agreement
NG	NorduGrid
NGN	Next Generation Network
NJS	Network Job Supervisor
OGF	Open Grid Forum
OGSA	Open Grid Service Architecture
OMII	Open Middleware Infrastructure Institute
ORB	Object Resource Broker
OSG	Open Science Grid
PBAC	Process Based Access Control
PBS	Portable Batch System
PCO	Point of Control and Observation
PGI	Production Grid Interoperability
PIXIT	Protocol Implementation Extra Information for Testing
POA	Portable Object Adapter
QA	Quality Alliance
QCM	Quality Certification Model
RBAC	Role Based Access Control
RFT	Reliable File Transfer
RLDS	Resource Locating and Description Service
RLS	Replica Location Service
RNS	Resource Namespace Service
RPC	Remote Procedure Call
RSL	Resource Specification Language
SAML	Security Assertion Markup Language
SGAS	SweGrid Accounting System
SGE	Sun Grid Engine
SLA	Service level Agreement
SPMD	Single Process Multiple Data
SSE	Smart Storage Element
SUT	System Under Test
TD	Test Description
TP	Test Purpose
TSF	Target System Factory
TSI	Target System Interface
TSS	Target System Service
TTCN	Testing and Test Control Notation
TURL	Transfer Uniform Resource Locator

UDDI	Universal Description, Discovery and Integration
UPL	Unicore protocol layer
UUDB	Unicore User Database
VO	Virtual Organization
VOMS	Virtual Organisation Management System
WLCG	Worldwide LHC Computing Grid
WMS	Workload Management System
XAML	eXtensible Application Markup Language
XIO	eXtensible Input/Output
XML	eXtensible Markup Language

4 Types of grid interoperability

Depending on the view on grid, grid interoperability can be interpreted differently. The different types of interoperability include internal interoperability within a grid infrastructure, between grid infrastructures, and between grid infrastructures and other systems.

4.1 Internal grid interoperability

Interoperability within a grid infrastructure means that the services provided by a grid infrastructure or entities using and implementing them are able to communicate by well defined interfaces. This means that the services in figure 1 are able to interoperate through common, standardized (or otherwise agreed) interfaces inside the grid infrastructure.

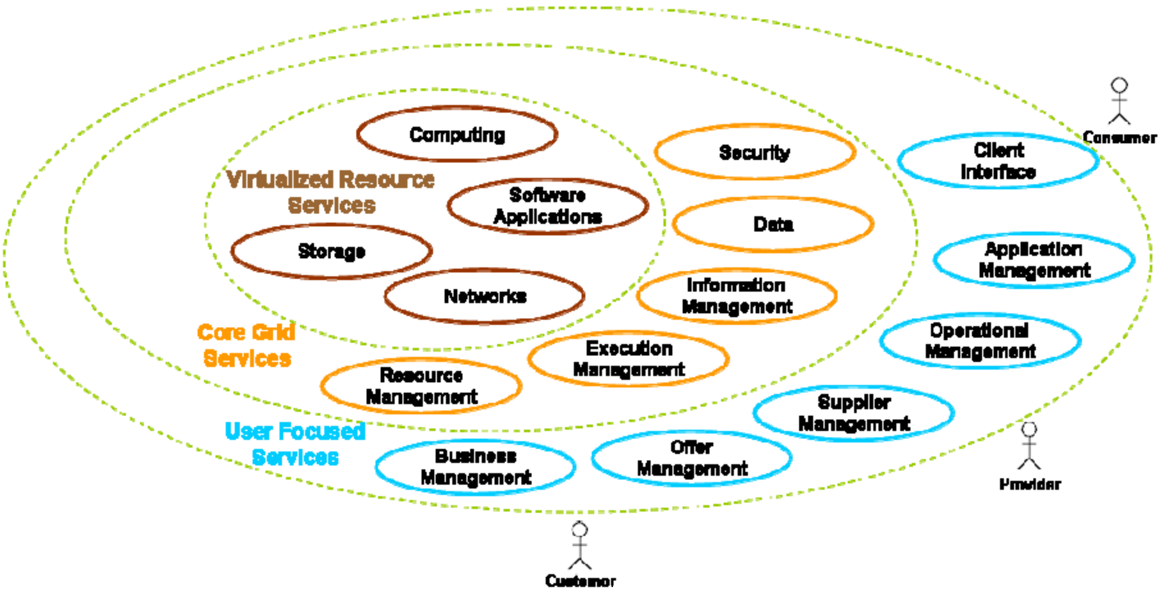

Figure 1: Conceptual model of a grid infrastructure

4.2 Interoperability between grid infrastructures

Interoperability between different grid infrastructures is usually located at user domain level, i.e. interoperability between end users. The grid infrastructure A and the grid infrastructure B, as depicted in figure 2, are able to communicate and exchange data through one or more standardized (or otherwise agreed) interfaces. More specifically, the services provided by grid infrastructure A understand the services provided by the grid infrastructure B.

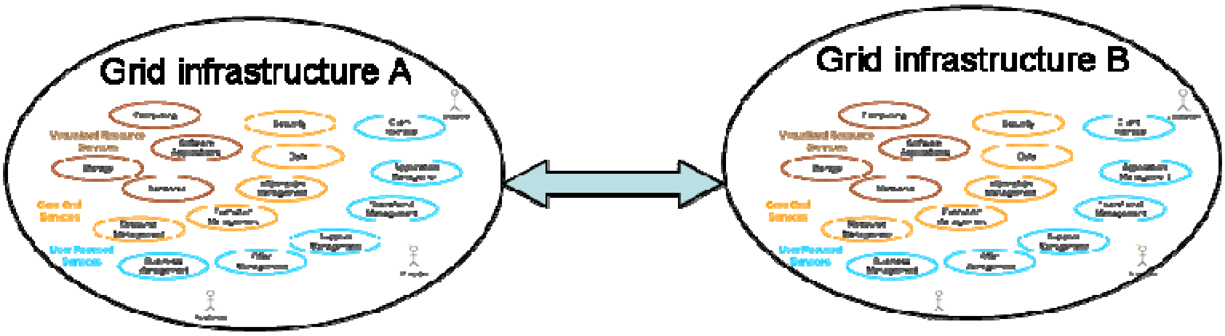

Figure 2: Communication between two grid infrastructure

4.3 Interoperability with other systems

The interoperability of a grid infrastructure with other systems is another scenario. Such another system can be, for example, a Next Generation Network (NGN), peer to peer or CORBA [i.53] system that are able to interact with a grid infrastructure in order to exchange information and data, or provide access to resources.

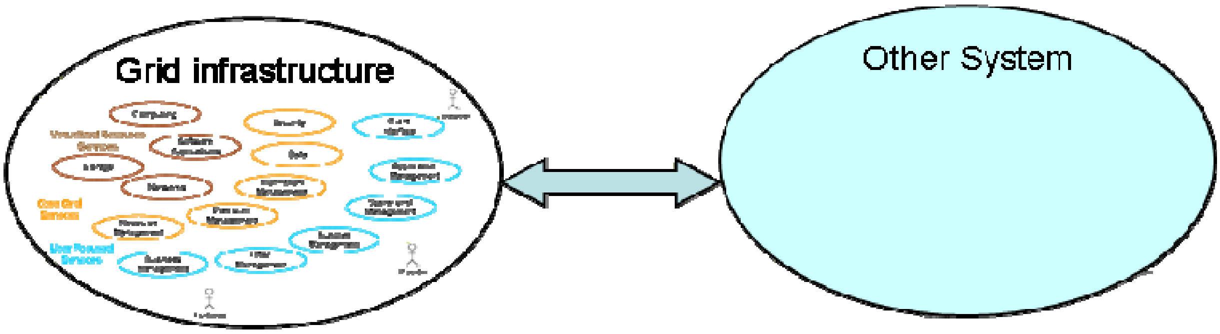

Figure 3: Interoperability between a grid infrastructure and another system

5 Approaches for achieving grid interoperability

A number of existing approaches to achieve interoperability between productive grids are presented by [i.5]. These approaches are distinguished as user driven, parallel development, gateways, adaptor and translators, and standardized interfaces.

5.1 User driven

In user driven approach, a user and the virtual organization the user belongs to use their own proprietary interfaces to access different grid infrastructures as illustrated in figure 4. The main efforts to overcome gaps in interoperation have to be taken by the user or its virtual organization because each virtual organization has to implement their own interfaces on top of each grid infrastructure. This results in a significant duplication of effort and loss of productivity [i.5].

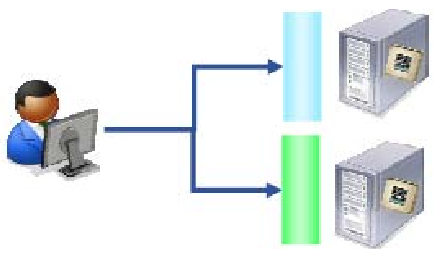

Figure 4: User driven scenario (from [i.5])

An example of this approach is the ATLAS collaboration [i.16]. Here, computing resources are provided by three different grid infrastructures each based on different grid middleware: Grid3/OSG [i.18] in the USA, NorduGrid/ARC [i.19] in Scandinavia and a few other countries, and LHC Computing Grid (LCG-2)/EGEE [i.17] in most of Europe, in Canada and in the Far East. As shown in figure 5, the modular implemented production system of ATLAS allows utilizing resources of the three grid infrastructures. The ATLAS central database is called ProdDB and it holds information about jobs whereas supervisor agents delegates these jobs to grid specific agents called executors that offer an interface to the underlying grid middleware [i.10].

An example use case is described in [i.20] where a software component is deployed as the LCG executor in order to submit jobs to the LCG. In the bottom level, grid specific tools, in case of LCG the lcg-clients, deal with the data, whereas high-level data-management across the grid infrastructures is ensured by the ATLAS data management system, i.e. the Don Quijotte [i.21], [i.27] service.

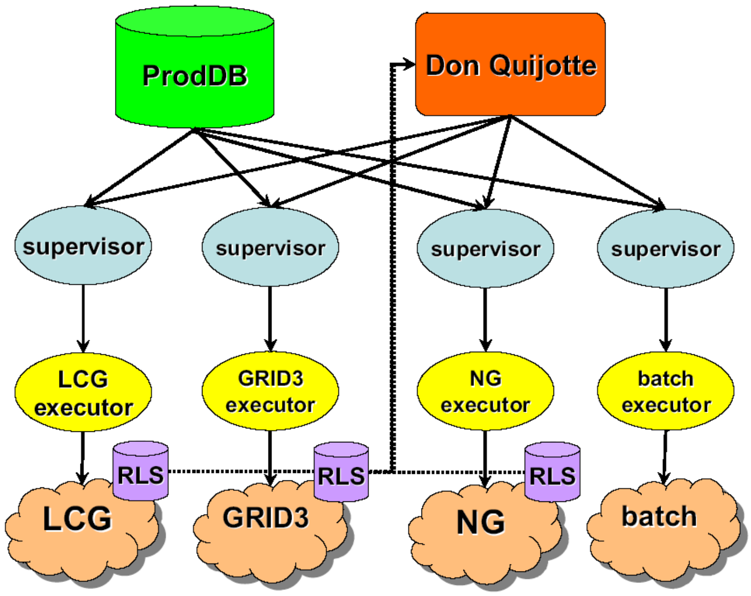

Figure 5: The ATLAS production framework (from [i.10])

5.2 Parallel deployment

In parallel deployment approach, a resource provider deploys multiple interfaces for his resources in order to make them available to virtual organizations as depicted in figure 6. If a resource provider wants to support a grid infrastructure, the required services have to be deployed on the resource which would scale with the number of grid infrastructures [i.5].

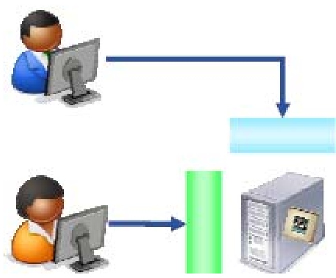

Figure 6: Site driven scenario (from [i.5])

The approach is only recommended for large resources centres like the GridKa, the German Tier-1 centre in the Worldwide LHC Computing Grid (WLCG) that supports four different LHC experiments: ALICE, ATLAS, CMS, and LHCb as well as four non-LHC high energy physics experiments. In addition, it supports project of Nordugrid, the German D-Grid, and EGEE [i.5].

5.3 Gateway

In gateway approach, a gateway service bridges the gap between two grid infrastructures as depicted in figure 7. A grid infrastructure behind the gateway service appears as a single resource. The gateways service has to support at least the minimum common set of services provided by the underlying grid infrastructure. Therefore, gateway services can be a single point of failure and a scalability bottleneck [i.5].

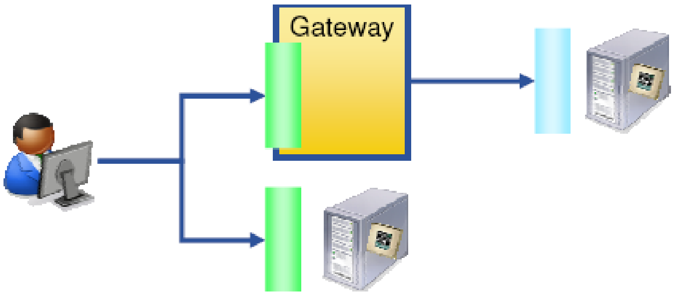

Figure 7: Gateway approach (from [i.5])

The gateway approach is applied by [i.1] to achieve interoperability between the EGEE grid infrastructure based on the gLite [i.59] middleware and the Chinese grid infrastructure CNGRID [i.63] which is based on the GOS [i.64] middleware. In this case, the gateway approach has been applied because gLite and GOS implement very different architectures.

Another example is the KnowARC project which has developed an ARC-EGEE middleware gateway that allows ARC users to use EGEE resources. A short description of the gateway implementation can be found in [i.14]. Whereas storage interoperability is in essence automatically covered by the fact that both middleware solutions utilize GridFTP-based storage elements and SRM technology, the real challenge was to bridge the interoperability gap on the computing service level. Finally, a CREAM2 WS-interface was used for accessing gLite computing resources. Further details on the gateway implementation can be found in [i.15] or on the KnowARC Web pages [i.11].

The gateway approach has also been used to provide gLite middleware access to worker nodes of unsupported architectures [i.22]. This case study presents the integration of the Italian ENEA-Grid with gLite middleware via a gateway. The gateway architecture assumes the existence of a shared file system, namely AFS for ENEA-Grid.

The integration of caGrid [i.25] with the TeraGrid [i.26] through a gateway service is described in [i.23]. The traditional caGrid analytical service is depicted in figure 8. In this scenario, the user application first queries the index service in order to get a list of available services. After the user has selected an appropriate service, the application sends the corresponding data and parameter to the service. After the service finishes the analysis, it transmits the result to the user application [i.23].

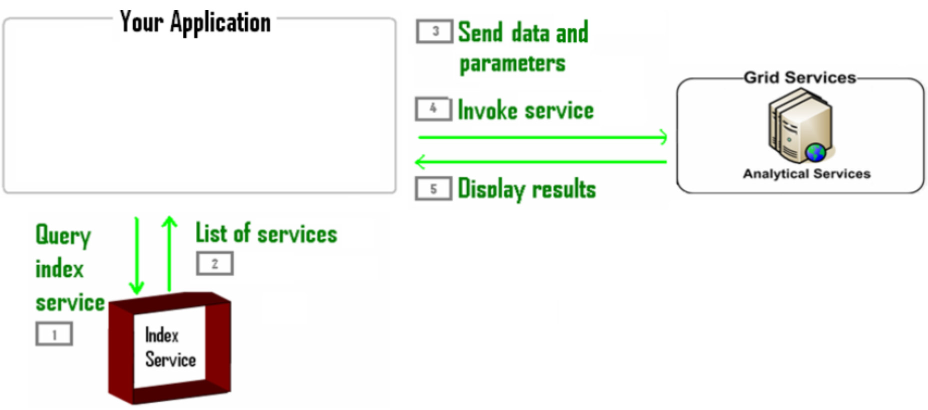

Figure 8: A traditional caGrid analytical service. (from [i.23])

The goal was to make TeraGrid resources available for the caGrid client application without applying changes on it. Therefore, the TeraGrid caGrid gateway is an extension of the traditional analysis service as depicted in figure 9 The user application still queries the index service, selects one, and invokes the caGrid service. But in this case the caGrid service is a TeraGrid aware gateway that forwards and submits the jobs to the TeraGrid. In this case, GridFTP is applied to transport the data and parameters between the TeraGrid and the gateway that forwards them to the user application.

In addition, TeraGrid supports 29 gateways developed independently by various research communities to access TeraGrids back-end resources. Hereby, TeraGrid has to provide and develop scalable service solution in order to meet the countless requirements that result from this decentralized development. Commonalities of the gateways service are present in the area of web services, community accounts, auditing, a scalable infrastructure, and flexible resource allocation and scheduling [i.24].

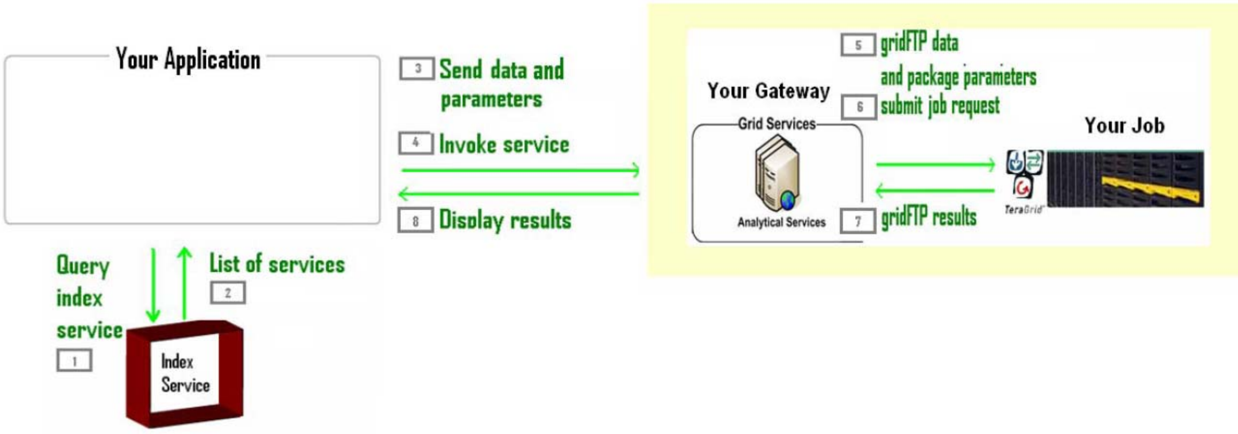

Figure 9: A TeraGrid-aware caGrid gateway service (from [i.23])

Furthermore, [i.39] present a description of an experiment on interoperation between the NAREGI Middleware Beta [i.69] and EGEE gLite. In detail, they showed the interoperability of job submission from NAREGI Middleware Beta to gLite and vice versa by applying the gateway approach. In addition, they made it possible to the NAREGI Middleware Beta to allocate gLite computing elements as its resource and, furthermore, they exposed the whole NAREGI Middleware Beta resources as one of the computing elements of gLite.

5.4 Adaptors and translators

In the adapters and translator approach, adaptors bridge incompatible grid infrastructure interfaces so that translators convert information in a format that is understandable by the other interface. Adaptors and translator are developed as new services and are included into the grid infrastructure. But existing interfaces of the grid infrastructure are not modified [i.5] and [i.12].

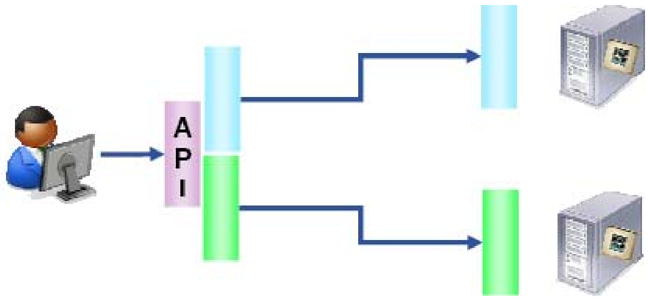

Figure 10: Adapters (from [i.5])

One example of realizing this approach is the ETSI Grid Component Model [i.42] which has recently been standardized [i.45], [i.46] and [i.47]. The GCM is an extension to the Fractal [i.43] specification so that it is feasible for grid infrastructures. The general idea behind component models is that software units (components) are provided to perform certain tasks. In the context of different applications, these components can be distributed and reused. A methodology for component-based development of grid application has been presented by [i.44].

The goal of GCM is to provide a unified access to different computing infrastructures. This is realized by specifying the deployment of applications on any grid infrastructure in a unified way. Grid infrastructures including resources and their access can be described using a standardized deployment descriptor [i.45] and the application descriptor [i.46] which are both based on XML. Both are converted into deployment information to establish a communication layer for an application over several infrastructures. A detailed description of the GCM standard is given in clause 9.1.

Another example of the adapter approach is the integration of CORBA and grid services. This can be achieved at least at two levels: a high-level where CORBA interfaces are wrapped around Grid services, or a low level where CORBA services are extended (and new services added) to support grid services. [i.8], [i.9] present a final solution which combines these approaches. The design and implementation of the presented work focuses however on a high-level integration. Additionally, they concentrated on providing CORBA applications access to grid services. Their approach also enables true interoperability between CORBA and Grid services. Figure 11 shows how a grid services depicted on the left align with CORBA services depicted on the right.

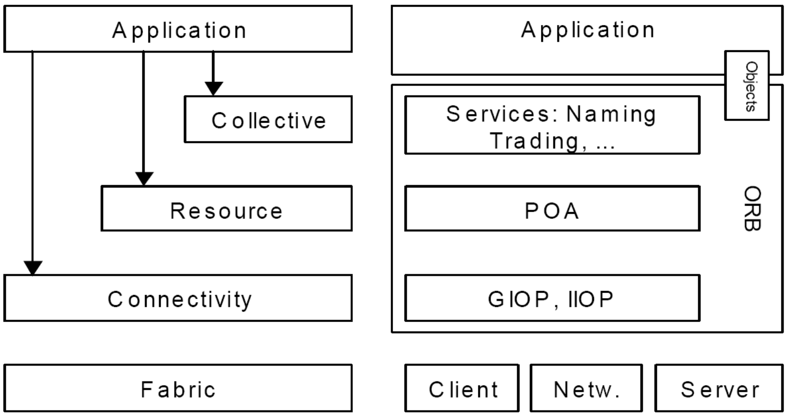

Figure 11: The Grid architecture and CORBA [i.8]

In the overall CORBA CoG Kit architecture, the CORBA Object Resource Broker (ORB) forms the middle-tier providing clients' access to CORBA server objects that interface to services on the Grid. Their implementation provides server objects for the Globus Toolkit (GT version 2.4) information management, security, remote job submission, and data access services.

5.5 Standardized interfaces

Another approach to address interoperability is the use of open, standardized or otherwise agreed upon interfaces. The interfaces that need standardization can evolve from the adaptor and translator deployment since mapping to different grid infrastructures have been identified. However, the drawback of this approach is that agreement on a common set of standard interfaces that also meet production grid requirement is shown to be very time consuming.

6 Standard adoption by grid infrastructures

This clause presents a number of studies on the adoption of standards by current grid infrastructure implementations.

6.1 gLite & UNICORE middleware assessment

Work presented in [i.2] focuses on interoperability of job submissions between gLite and UNICORE since core services are also implemented proprietarily rather than adopting agreed standards. However, UNICORE and gLite adopt common open standards such as OGSA-BES [i.65] and the JSDL [i.66] specification which means that the same client can invoke operations based on these standards within UNICORE or gLite. In addition to these interfaces, it is crucial that they are based on the same security profile using X.509 certificates for authentication and SAML [i.67] assertions for authorization in order not to be rejected [i.2].

6.2 OMII assessment

The OMII-UK includes projects for developing tools that aim at interoperability of grid such as GridSAM, JSDL application repository and OMII-SAGA. In addition, OMII-Europe focuses on achieving interoperability through common standards by implementing standards in tandem with standards development on all middleware platforms with involvement of OGF and OASIS. Their vision is an easy access and usage of grid resources in heterogeneous e-infrastructures crossing national, pan-European, and global boundaries. OMII-Europe focuses on providing common interfaces and integration of major grid software infrastructures [i.6]. In detail, common interoperable services such as database access, virtual organization management, accounting, job submission, and job monitoring have to be enabled to achieve grid interoperability. In addition, this has to be covered by an interoperable security framework and a common access to infrastructure services through a portal. Initially, their main focus lies on the interoperation of gLite, UNICORE, and Globus [i.6].

The WISDOM interoperability scenario, a real world case study of the OMII-Europe, demonstrates how the OMII-Europe components can be used. In detail, the infrastructure of the *Distributed European Infrastructure for Supercomputing Applications* (DEISA) that is based on non web service based UNICORE 5 and the *Enabling Grids for e-Science* (EGEE) that is deployed on the non web services based gLite. A step wise description of this interoperability scenario is shown in an overview in a fact sheet [i.3]. A conclusion of this case study is that users cannot use one middleware to access both. Both middleware do not implement a minimal set of standard in order to allow interoperation [i.4].

In addition, OMII has analyzed and compared different grid middleware implementations including gLite, Globus Toolkit, UNICORE 5, UNICORE 6, Vega-GOS, CROWNGrid, ARC, and OMII-UK based on community standards by [i.7]. Their concluding contribution is a comparison table of the listed infrastructures based on each elementary capability as depicted in Table 1. In addition, the convergence of standards of these implementations related to each capability is given.

Table 1: Grid middleware comparison by capabilities (from [i.7])

Capability	gLite	Globus	UNICORE 5	UNICORE 6	CROWNGrid	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: 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		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	std: ITU X.509 + RFC 3820 + WS-Trust
Security.AttributeAuthority	VOMS		UUDB	WS-UUDB	CROWN AA		VOMS	Attribute Authority + SAML
Security.Authorization	G-PBox, gJAF	CAS	UUDB	WS-UUDB	CROWN AuthZ	GridSAM		XACML, SAML
Security.IdentyMapping	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		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	WS-Naming		WS-Naming + WS-Addressing
Data.Naming.Resolver	LFC, DPM, StoRM	RLS			LDS			
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, gsirfio	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	Grimoires	LDAP v2	UDDI
Information.Logging	Logging and Bookeeping						ARC Logger	

Capability	gLite	Globus	UNICORE 5	UNICORE 6	CROWNGrid	OMII-UK	ARC	Convergence
Information.Monitoring	R-GMA, GridICE, CEMon	MDS 4			CROWM Monitoring		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	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

6.3 KnowARC assessment

The KnowARC project studied several grid middleware implementations to investigate about their possibilities to interoperate based on standards that they support. The results of these studies can be found in [i.12]. A summary taken from [i.12] is given in tables 2 and 3.

Table 2: Summary of grid middleware services (functional grid interoperability level) (from [i.12])

Functionality		ARC	gLite	GUG	OSG	UNICORE 6	GRIA	OMII
Security	**Authentication**	GSI	Extended GSI	TSL	GSI	SSL, X.509	HTTPS, X.509, WS-Security, WS-Federation	HTTPS, X.509, WS-Security, WS-Federation
	Authorization	Local policies, Gridmap, VOMS, LDAP, GACL	VOMS	Design phase	Gridmap, VOMS	UUDB	PBAC Authorization Restriction	PERMIS RBAC
	VO management	VOMS	VOMS	Design phase	VOMS	none	Business VO	VOMS-Proposal phase
Information system	**Characterization**	ARC	BDII	GUG	Glue	static	Gria	
	Discovery	ARC, LDAP		GUG, P2P GIS	BDII LDAP	none	Taverna plug-in	
	Monitoring	ARC Monitor	R-GMA	None	Monalisa	Target System WS-Resources	None	
Job execution management	**Job description**	XRSL/ JSDL	JDL	JSDL	RSL	Abstract Job Object, JSDL, BPEL	Java, scripts	
	Job submission	GRIDFTP	GRAM/ Condor-G	Exec OGSA-BES	GRAM	TSS, JMS,TSF	Exec	GridSAM
Data services	**Transport**	GridFtp	FTS	GUG storage	GridFtp	GridFtp, SCP, URL		
	Management	SRM	SRM POSIX	POSIX		WS-Resources, SMS	OGSA-DAI	OGSA-DAI

Table 3: Summary of grid middleware services (high level grid interoperability) (from [i.12])

Functionality	ARC	gLite	GUG	OSG	UNICORE 6	GRIA	OMII
Workflow	None	P-Grade	None	DAGman	Client, BPEL	GRIA Workflow plugins: Taverna and Freefluo	Scufl Taverna
Brokering	Client	Central service	GUG broker		none		
Business aspects	HP Tycoon				GRAAP-WG	SLAs	SLA Toolkit

One of the main goals of KnowARC is to create a novel, powerful Next Generation Grid middleware based on NorduGrid's Grid middleware "*Advanced Resource Connector*" (ARC). For achieving interoperability of ARC with other middlewares, the ARC proprietary interfaces are replaced with community based ones. A list of standards and their relevance for the KnowARC project can be found in [i.13]. The document lists 48 specifications, out of which 11 are classified as highly relevant and 14 as of potential relevance.

6.4 OGF assessment

OGF has identified a set of standards and specifications for facilitating access to distributed computing resources [i.40]. Afterwards, the specifications have been applied to different scenarios in order to demonstrate how distributed computing recourses can be accessed and identify the requirements for middleware providers and Independent Software Vendors (ISV). Their products and projects specification support is listed in table 4.

Table 4: Grid standard support by selected projects and products (from [i.40])

	Specifications						
Project or Product	OGSA-BSP 2.0	JS DL	OGSA-BES	HPCBP	File Staging	OGSA-ByteIO	RNS
Globus	No	Yes	Yes	Yes	No	No	No
UNICORE 6	No	Yes	Yes	Yes	No	Yes	?
USMT (Fujitsu)	Will	Yes	Yes	Yes	No	Yes	?
HPCS 2008 (Microsoft)	No	Yes	Yes	Yes	Yes	No	No
Genesis II	Will	Yes	Yes	Yes	Yes	Yes	Yes
GridSAM (OMII-UK)	No	Yes	Yes	Yes	Yes	No	No
Crown	No	Yes	Yes	Yes	No	No	No
BES++ (Platform)	No	Yes	Yes	Yes	Yes	No	No
NAREGI	No	Yes*	No	No	No	No	?
Gfarm	No	No	No	No	No	Will	No
gLite	No	Proto-type	Prototype	Proto-type	No	Proto-type	?
ARC1 (NorduGrid)	Not planned	Yes	Yes	Yes	No	Yes	No

* Provides support for JSDL SPMD

\+ Provides support for JSDL Parameter Sweep

7 Related work on grid infrastructure testing

An approach of testing grid application workflows provided by grid infrastructures based on the grid middleware Globus Toolkit 4 is outlined in [i.28]. This has been focused on using conformance testing [i.29] for test specification with the Testing and Test Control Notation (TTCN-3) [i.30] and [i.31]. This case study demonstrates how distributed testing concepts and remote communication mechanisms can facilitate very well the testing of grid infrastructures.

The eInfrastructure for Testing, Integration and Configuration of Software (ETICS) [i.32] provides a build and test system by means of reliable grid software and industry-standard best practice. ETICS is used for the build and integration process of gLite by the EGEE. The Ipv6 compliance analysis of gLite code and the distributed Ipv6 experimental tested where the Ipv6 version of Berkeley Database Information Index (BDII) was tested has been implemented with ETICS system EGEE. In addition, EGEE applied the ETICS system to build some of its high-level services, such as GridWay. Furthermore, the ongoing ETICS 2 project endeavours to support the widespread adoption of grid technologies.

OGF has done some work on test specifications for ByteIO [i.33] and GridRPC [i.34]. These test specifications are informal test descriptions that mainly follow a conformance oriented, unit testing approach rather than end-to-end functionality testing from an ETSI point of view. Other related work [i.38] is entitled "Interoperability Testing for the GridRPC API Specification" but describes instead classical conformance testing of three GridRPC API implementations. Classical conformance testing means, that the test cases check the conformance of the API implementations to the GridRPC API recommendation.

An approach for Multi-Environment Software Testing on the Grid is given by [i.35]. They developed and applied the tool GridUnit for controlling and monitoring execution of tests on several nodes of a Grid. The tests that are executed on each node are implemented using the Junit [i.35] Java-based test framework.

The Centre for Development of Advanced Computing (C-DAC) provides C-DAC Grid Computing Test Suites and Grid Probes [i.37]. These test suites have the objective to check basic capabilities of the Grid middleware itself, e.g. remote job submission, validation of proxy or mutual authentication in a Grid environment that is based on Globus Toolkit 4. These test suites are implemented in various general purpose programming languages such as C or Java.

8 Grid interoperability initiatives

8.1 ETSI

8.1.1 ETSI in brief

ETSI's core mission is the development of European telecommunications standards in work initiated by ETSI members. Due to convergence of ICT, ETSI is becoming increasingly involved in software and computing standards. Leveraging ETSI's experience in standards development, standards compliance testing, and interoperability testing ETSI has been organizing Grid Plugtests since 2004 mostly in context of an ETSI Grids@work workshop. An ETSI technical committee focused on investigating the use of grid and related technologies in the telecommunication domain was formed in 2006.

8.1.2 ETSI GRID Plugtests

This event has been typically run in the past conjunction with the Grids@Work workshop. In 2009 the scope of the event has been changed away from a programming contest towards an assessment of the GCM standard in different grid technologies. Note that until 2009 the ETSI TC GRID has not been involved in the organization and planning of this event.

8.1.2.1 Overview

These events have traditionally focused on application level grids, where Plugtest participants are writing a distributed application to perform a given task. That application is deployed on a grid infrastructure of heterogeneous resources integrated prior to the event. The Plugtest organizers have taken responsibility for securing various grid computing resources and preparing them for the deployment of the distributed application. It was then the responsibility of the participants to write a distributed application which best utilizes the available grid infrastructure. At various points in the history of the plug test there has been either a requirement or a very strong recommendation to use Pro Active Java distributed computing library for writing the application and/or the Pro Active deployment descriptors (now Grid Component Model Deployment Descriptor) to deploy the application. The participants themselves are writing a single monolithic distributed application which is deployed and run on the available grid infrastructure, and the resource providers have no requirement to provide standard interfaces or pre-configure their system. The interoperability is based on the use of a common distributed computing library (such as CORBA, Java RMI, Ibis, or ProActive) to provide a uniform interface to the underlying compute resource. Only recently has any "standard" been involved in the Plugtest, and that being the ETSI GCM Deployment Descriptor which is an XML description of an application deployment environment, specifying details regarding the software to run, necessary libraries, remote resources, and remote resource access details.

8.1.2.2 Participation Details

The ETSI Plugtests events open to participation to anyone (members and non-members of ETSI). The participation fees vary between events. Some events have not had any participation fee, Grid Plugtests have generally been run at the ETSI in Sophia Antipolis, France. In 2007 is has been run in Beijing, China. Events typically happen in October or November each year.

8.1.2.3 Organizations and companies involved

Companies and organizations have not actively participated but rather pledged and provided computing resources or access to them for the programming contest, e.g. Microsoft, Amazon. Only student teams from Universities have been involved.

Grid Plug Test 1 (2004): N-Queens (Sophia Antipolis, FR)

Participants:

> Universidad de Chile (Chile), INRIA (France), NTU (Singapore), TOURNANT (France), University of Southern California (USA)

Grid infrastructure:

> Composed total of 473 machines, bearing 800 processors, on 20 different sites in Australia, Europe, North and South America, and India. Set up and integrated by INRIA via Pro Active software with (SSH), LSF, PBS, OAR, UNICORE, Globus Toolkit 4, NorduGrid, Torque, PRUN, SUN Grid Engine, and gLite.

Grid Plug Test 2 (2005): N-Queens and Flowshop (Sophia Antipolis, FR)

Participants:

> INRIA (France), LSC/UFSM (Brazil), Beijing University of Post and Telecommunications (China), National Taiwan University and Ming Chuan University (Taiwan), University of Southampton (UK), National eScience Centre (UK), Vrije University (Netherlands), FhG SCAI (Germany), ECMWF (UK), MCNC (USA).

Grid infrastructure:

> Composed total of 13 different countries, in more than 40 sites in Australia, Brazil, China, France, Ireland, Greece, Switzerland, USA, Germany, Italy, Norway, and Netherlands, gathering 2 700 processors. Set up and integrated by INRIA via Pro Active software.

Grid Plug Test 3 (2006): N-Queens and Flowshop (Sophia Antipolis, FR)

Participants:

LSC/UFSM (Brazil), ChinaGrid/ Huazhong University of Sci. & Tech. (China), Beijing University of Post and Telecommunications (China), FIT (China), University Diego Portales (Chile), Institut d'Informatique et de Mathematique Appliquees de Grenoble (France), University of Tokyo (Japan), Vrije University (Netherlands), Poznan University of Technology (Poland)

Grid infrastructure:

Included 20 sites, gathering 4 130 cores in Australia, Chile, France (Grid'5000), Italy, Japan, China, and Netherlands. Set up and integrated by INRIA via Pro Active software with (SSH), LSF, PBS, OAR, Torque, PRUN, and SUN Grid Engine.

Web site:

http://www.etsi.org/WebSite/OurServices/Plugtests/2006GRID.aspx

Grid Plug Test 4 (2007): N-Queens and Flowshop (Beijing, CN)

Participants:

ChinaGrid/Tsinghua University (China), ACT (China), Poznan University of Technology (Poland), Beijing University of Post and Telecommunications (China), MOAIS (France), Tsinghua Technical University (China).

Grid infrastructure:

Four grids have been involved in the computations, located in France, Italy, Netherlands and Japan, and gathering 4 538 cores. Set up and integrated by INRIA via Pro Active software.

Grid Plug Test 5 (2008): Super Quant Monte-Carlo Challenge (organized by INRIA only)

Participants:

ChinaGrid/Tsinghua University (China), ACT (China), China Academy of Science (China), National University of Defence Technology (China), MOAIS (France), Tsinghua Technical University (China).

Grid infrastructure:

20 Sites from 2 grids (Grid'5000 in France and Intrigger in Japan) were involved in this test, gathering a total of 6 242 cores. Set up and integrated by INRIA via Pro Active software.

Web site:

http://www-sop.inria.fr/oasis/plugtests2008/ProActiveMonteCarloPricingContest.html

8.1.2.4 Feedback to Standardization

As this event has so far not really been based on standards therefore there has been no feedback to standardization.

8.1.2.5 Tools and test environment

The Pro Active library and the GCM Deployment Descriptor have formed so far the primary software foundation for the Grid Plugtests, although at times participants have had the flexibility to use other libraries or deployment mechanisms. The test environment has relied heavily on contributions by the French Grid'5000 research grid (consisting of approximately 15 nationally distributed compute clusters and over 3 000 compute cores). Depending on the year, other compute resources/clusters from China, Japan, Italy and elsewhere have been incorporated.

8.1.2.6 Output reports, certification

Recent Grid Plugtests have produced a summary report within 4 months of the completion of the plug test. These reports have however rarely been made publically available.

8.1.3 ETSI Grids, Clouds and Service Infrastructure event

This event has been first held for the first time in 2009. It marks a departure from the previous Grids@work approach. Similarly as before the event has a Plugtest and a workshop component. The focus of this event has however been redirected to ETSI TC Grid standard assessment and its industrial stakeholders. Also it has widened the scope of the event from grid to cloud computing technology. This is the first event in which ETSI TC GRID has been involved in the organization and planning of and ETSI grid event.

8.1.3.1 Overview

The Plugtest part has been focused so far on the assessment and /or ability to deploy applications or jobs onto different computing infrastructures. The bases of the tests are the Grid Component Model (GCM) deployment and application descriptors which have been standardized by ETSI TC Grid. Although the event is based on GCM participant are not forced to support directly the loading of the XML files. Instead it is allowed to also convert GCM descriptor information manually into the non-standardized interface format that a given infrastructure supports. This is especially needed for infrastructures that do not have their mapping standardized in the GCM DD standard. The tests mainly focus on GCM DD since it is very flexible towards new non-standardized interfaces. The GCM AD is always used in combination with the GCM DD. The main interest of the GCM standard is to gain the ability to deploy one application, i.e. multiple processes, across more than one infrastructure. The new format has essentially decoupled the ETSI Grid Plugtest from the need to use of Pro Active software, i.e. opened it up to the evaluation of any grid middleware, cluster, or cloud computing system vendor.

8.1.3.2 Participation Details

The ETSI Plugtests events open to participation to anyone (members and non-members of ETSI as well as grid/cloud technology vendors as well as observers, e.g. telecom operators). The participation fees vary between events. This event has so far been run at the ETSI in Sophia Antipolis, France.

8.1.3.3 Organizations and companies involved

Companies and organizations involved in the workshop include British Telecom, Alcatel Lucent Bell Labs, NSN, Telefonica, IBM, Intel, Sun Microsystems, Microsoft, Amazon, 451 group, Animoto, etc.

8.1.3.4 Feedback to Standardization

The main purpose of the Plugtest is used to validate existing standardized mappings in the GCM DD as well as to extend them with new ones.

8.1.3.5 Tools and test environment

The test environment is a simple IP network. Infrastructure vendors may connect to their equipment at a remote location. Deployment managers should participate locally. As the interface with the underlying infrastructures is not part of the GCM specification, the test execution may depend on each GCM implementation. It can be performed in an automated manner or manually. Tests are executed by either loading GCM DD and AD (if applicable) files to the deployment manager or by manually converting information contained in these files into commands supported by the infrastructure(s) under test.

8.1.3.6 Output reports, certification

ETSI Plugtests are covered by an NDA which allow the publication of an anonymous result report, i.e. a report that shows results without explicitly associating companies with them. This event is not intended to provide any certification.

8.2 OGF

8.2.1 OGF in brief

The Open Grid Forum is a merger between the Global Grid Forum and the Enterprise Grid Alliance (EGA). Their primary goal is provide a forum for the grid community to establish standards for grid operation (http://www.ogf.org/).

8.2.2 Standards Validation

All OGF standards (called "Recommendations") go through a process as documented in [i.70] prior to being designated an OGF standard. In order to gain "Recommendation" status an OGF standard has to be validated and shown to be working in at least two implementations.

8.2.2.1 Overview

Standards-track documents must first pass a 15-day internal review and then a 60-day public comment period. Provided both the internal and public comment period judge the submitted document acceptable, it becomes a "Proposed Recommendation". Within 24 months there must be 2 complete independent implementations and an experience report describing use of the implementations, interoperability between them, and effectiveness of the standard. These are submitted to the OGF Area Directors and reviewed over a 4 month period by 3 experts. Public comments are also solicited. The standard can then be promoted to a full recommendation, held as a proposed recommendation pending further revisions and experience (with another review in 12 months), or closed with a classification of "historical" or "obsolete".

8.2.2.2 Participation Details

Anyone can submit a document to the OGF for consideration as a standard. It will be reviewed along the way by OGF-internal Area Directors, working groups, and experts drawn from the OGF membership and beyond. It is typical, though not required, that a standards track document is "shepherded" by an OGF group. Parties involved in preparing the standard are not allowed to participate in the document review.

8.2.2.3 Organizations and companies involved

The OGF is comprised of dozens of large organizations, from public, private, and academic settings. In total there are 40 organizational members (http://www.ogf.org/Members/members_members.php) and hundreds of actively involved individuals. For example, a recent meeting in June 2008 had 458 participants (http://www.ogf.org/OGF23/participants.php).

8.2.2.4 Feedback to Standardization

Both the internal and public review periods provide critical feedback to the document authors. The purpose of the experience report requirement is to ensure the standard's authors have reviewed the issues arising from producing and operating implementations of the standard, and their interoperability. If a document is not promoted along the standards track, a report explaining why is produced.

8.2.2.5 Tools and test environment

No standard tools or test environments are specified or recommended by the OGF in the development of grid standards.

8.2.2.6 Output reports, certification

The OGF has so far six full standards ("Recommendations"):

- Job Submission Description Language (JSDL) GFD.136 [i.72].
- Distributed Resource Management Application API (DRMAA) GFD.133 [i.73].
- HPC Basic Profile GFD.114 [i.74].

- JSDL HPC Profile Application Extension GFD.111 [i.75].
- OGSA Basic Execution Service GFD.108 [i.76].
- GridRPC GFD.52 [i.77].

Each of these has an associated experience report:

- Implementation and Interoperability Experiences with the Job Submission Description Language GFD.140 [i.78].
- Interoperability Experiences with the High Performance Computing Basic Profile, GFD.124 [i.79].
- PBS/Torque DRMAA 1.0 Implementation - Experience Report, GFD.117 [i.80].
- DRMAA 1.0 Implementation - Experience Reports, GFD.103 [i.81], GFD.104 [i.82], GFD.105 [i.83].
- Interoperability Testing for The GridRPC API Specification, GFD.102 [i.84].

These can be found from the main OGF public document area: http://www.ogf.org/gf/docs/?final.

8.2.3 Grid Interoperability Now

8.2.3.1 Overview

The purpose of this group is to coordinate a set of interoperation efforts among production grids interested in interoperating in support of applications that require resources in multiple Grids. This OGF community group does not produce interoperability standards, or review standards for interoperability, but instead acts as a forum for managers of production grid environments who require or desire interoperation with other grid environments. Their work covers three main areas: discussions and experience reports on integrating production grids; comparisons and evaluations of multiple implementations of a standard; demonstrations at OGF meetings and other major grid computing conferences.

Two key events for the OGF have been the interoperability demonstrations at Supercomputing 2007 and Supercomputing 2008, where dozens of research groups and commercial companies highlighted the interoperability of their products, packages, and systems. In late December 2008 a working group was formed to agree on production grid standards profiles, named Production Grid Interoperability (PGI). They specifically will be looking at the OGF standards BES, JSDL, GridFTP, SRM, and GLUE2.

8.2.3.2 Participation Details

Anyone can participate in GIN discussions, however GIN demonstrations at conferences are limited to OGF members and organized in advance.

8.2.3.3 Organizations and companies involved

This group is dominated by the large national production grids such as LCG, NorduGrid, D-Grid, UK NGS, OSG, and TeraGrid, however commercial organizations such as Microsoft, Platform, and Altair have participated.

8.2.3.4 Feedback to Standardization

As a community group, it has no specific mandate to provide feedback on standards, however many group members are also involved with the standards working groups and as can be seen from the reports listed below, there are a number of valuable papers and reports that have been produced by the group. The PGI group does not appear to have new standards as a goal but rather the production of a standards profile (an amalgam of interconnected standards to provide an interoperable production grid framework).

8.2.3.5 Tools and test environment

GIN has no specific tools or test environment, however the members represent senior members of national grid infrastructures and have committed their own production computing infrastructures to the task of evaluating interoperability.

8.2.3.6 Output reports, certification

A number of reports are available [i.48], [i.49], [i.50] which includes GIN reports as well as reports written by GIN members.

8.3 ETICS Consortium

8.3.1 ETICS in brief

ETICS (eInfrastructure for Testing, Integration and Configuration of Software) was initiated as a project to manage the substantial amount of software developed (and in development) as part of the EGEE project, primarily used by Large Hadron Collider Computing Grid (LCG). It has been expanded and formalized into a system to facilitate software testing for large projects. The project is co-funded by the European Commission.

8.3.2 ETICS

8.3.2.1 Overview

ETICS is a software QA system. It was originally designed to facilitate the interoperability testing of LCG (now EGEE/gLite) grid software. The hundreds of grid software components behaved differently on slightly different underlying hardware, software, and network configurations. ETICS provides a system for the automated compilation and configuration of those packages on various platforms, source code analysis, unit testing, and automated report generation. It is an open system that can be used by anyone.

8.3.2.2 Participation Details

The core of ETICS is used by the EGEE developers and community, however anyone can register and download ETICS clients for use in their own development projects.

8.3.2.3 Organizations and companies involved

The ETICS consortium consists of CERN (coordinator), INFN, Engineering Ingegneria Informatica S.p.A., 4D Soft Ltd., MTA SZTAKI, Vega IT GmbH, Forschungzentrum Jülich, and the University of Wisconsin-Madison.

8.3.2.4 Feedback to Standardization

No information available.

8.3.2.5 Tools and test environment

ETICS [i.51], [i.52] is a tool and test environment combined. As a large scale system, currently hosting 77 224 packages, it acts as a software development repository. Each project can have multiple modules, modules consist of packages, and modules or packages can be versioned. There is an associated set of access permissions for each module, along with configuration and testing information. This can be used to do automated compilation on various platforms followed by execution of a test suite. The compilation results, code QA, and test results form a report.

8.3.2.6 Output reports, certification

ETICS can automatically analyze a registered project, module, or package and produce a quality certification based on the Grid Quality Certification Model (Grid-QCM). ETICS produces numerous automated reports regarding software quality. It does not do specific tests for interface compliance.

9 Testing framework for the Grid Component Model

9.1 Introduction

A generic approach to interoperability testing for Internet Protocol Testing (IPT) is presented [i.41]. Here, interoperability testing is defined as the "activity of proving that end-to-end functionality between (at least) two communicating systems is as required by the base standard(s) on which those systems are based". The document introduces the interoperability testing process that includes descriptions of basic interoperability concepts, interoperability test architectures, and interoperability test models. In addition, the process for the development of interoperability test specification and execution are presented.

In the present document, the development of an interoperability grid testing framework is described which has been derived from the generic approach to interoperability testing [i.41]. This grid interoperability testing framework is based on the ETSI GCM standards that have been introduced in clause 5.4 of this document. The GCM standards were not meant to standardize interfaces but to abstract a common set of properties from existing interfaces. In addition, it standardizes the use of these proprietary interfaces towards the respective platforms. This approach has been chosen to allow infrastructure providers not to change their interfaces for a deployment of applications.

The main focus of the testing framework is to validate that properties described in the GCM DD and AD are implemented by the equipment under test. This equipment includes one or more infrastructures as well as a GCM deployment manager as shown in the abstract architecture shown in figure 12.

Afterwards, example test purposes and test descriptions are presented which have been developed based on this framework. Due to the proprietary nature of deployment interfaces the testing framework is currently envisioned to primarily serve as the basis for development of informal test descriptions suitable for manual execution of tests, e.g. at interoperability events such as ETSI Plugtests™. However, test descriptions may also be used as a basis for test case development, i.e. an automated testing framework.

9.2 Grid Component Model

The Grid Component Model (GCM) standard contains three parts: the GCM Application Description [i.46], GCM Interoperability Deployment [i.45], and GCM Fractal Architecture Description Language (ADL) [i.47]. A generic GCM test architecture which focuses on the GCM Application Descriptor (AD) and Deployment Descriptor (DD) has been developed as shown in figure 12. Here, the user is assumed to provide a (test) application, a GCM DD XML file, as well as optionally a GCM AD XML file.

The GCM DD describes resources requested from one or more different infrastructures for an application. The GCM DD is converted by the deployment manager into the invocation of specific infrastructure services or commands. This conversion process should be done in an automated manner by a deployment manager but may need to be performed manually if the use of the infrastructure interface has not yet been standardized in [i.45]. The GCM DD is mapped to resources of the specified infrastructure(s), and then used to deploy and establish a communication layer, called the GCM infrastructure, which is used for application deployment and execution. Input and output data servers can be used to store input and/or output data of GCM applications independent of the infrastructure on which it runs. Data can be accessed remotely or locally.

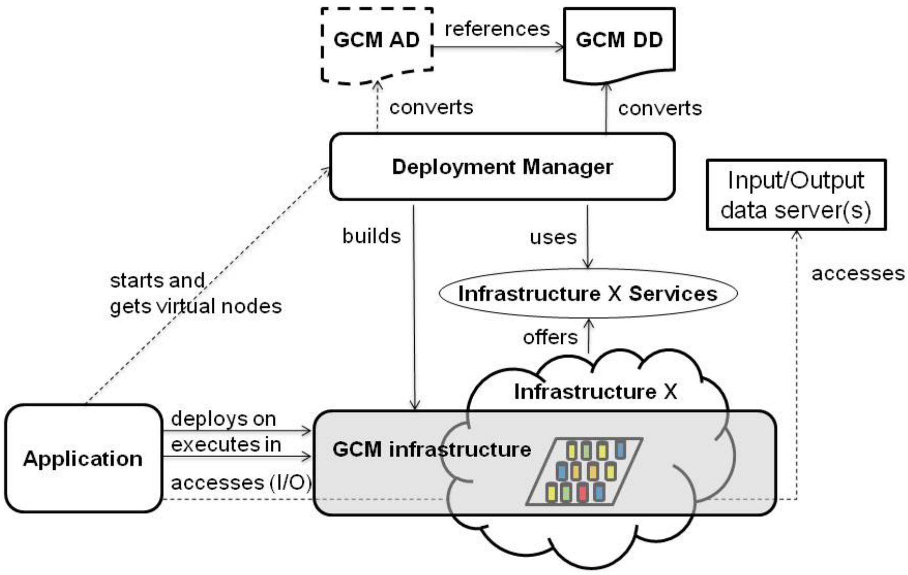

Figure 12: GCM Architecture

A GCM AD describes the requirements of an application from an underlying infrastructure, e.g. how virtual nodes (VNs) required by the application are mapped to resources defined in the GCM DD it references. In addition, an application may access data servers to read input data and write output data from locations specified in the GCM AD.A GCM DD describes resources that are expected to be available and provided by one or more infrastructures. Note that the resources requested in a GCM DD are often only a subset of all resources available in one or more infrastructures. The resources have to be accessible either in a direct or indirect manner. While infrastructures with indirect resource access offer a service that contacts a job scheduler and grants access to resources, infrastructures with direct resource access perform the deployment on the resources without any manager.

Examples of infrastructures with indirect resource access include clusters and/or grid middleware. Examples of infrastructures with direct resource access include desktop computers and cloud computing systems. A set of desktop computers may also be collected to form a group infrastructure.

The GCM DD standard contains already a number of standardized mappings to a number of different commercial as well as open source infrastructures:

- Infrastructures with indirect resource access:
  - Local resources manager including IBM LoadLeveler [i.54], Platform Load Sharing Facility (LSF) [i.55], Microsoft Compute Cluster Server (CCS) [i.56], Portable Batch System (PBS) [i.57], and Sun Grid Engine (SGE) [i.58].
  - Grid infrastructure including gLite [i.59], Fura [i.60], Globus [i.61], and Unicore [i.62].
- Infrastructures with direct resource access:
  - Desktop computer including MS Windows and Linux.
  - Cloud computing system including Amazon EC2.

Input and output data servers are used to store input and/or output data of GCM applications independent of the infrastructure on which it runs. Input and output data servers can be access independently using a supported file access protocol such as http, ftp, sftp, or file. Depending on the protocol, the data is accessed remotely or locally.

9.3 Goals

The main purpose of the tests is the assessment of the standardized GCM Deployment Descriptor (DD) and Application Descriptor (AD). The general test objective is to check that applications can be deployed and executed on a given infrastructure based on the information provided in GCM DD and AD. An infrastructure can either provide direct or indirect resource access. To access an infrastructure, its protocol need to be followed as specified in [i.45].

For a classification of functionalities that may be provided by a System Under Test (SUT), we define compliance levels as follows:

Compliance by the infrastructure:

1) An infrastructure does not support properties described in GCM AD and DD.

2) An infrastructure supports properties described in GCM AD and DD but are converted in a manual manner.

3) An infrastructure supports properties described in GCM AD and DD and are converted in an automated manner.

Compliance by the deployment manager:

1) Multiple infrastructures support fulfil level 1.

2) Multiple infrastructures support fulfil either level 1 or level 2.

3) Multiple infrastructures support fulfil level 2.

9.4 Test architecture

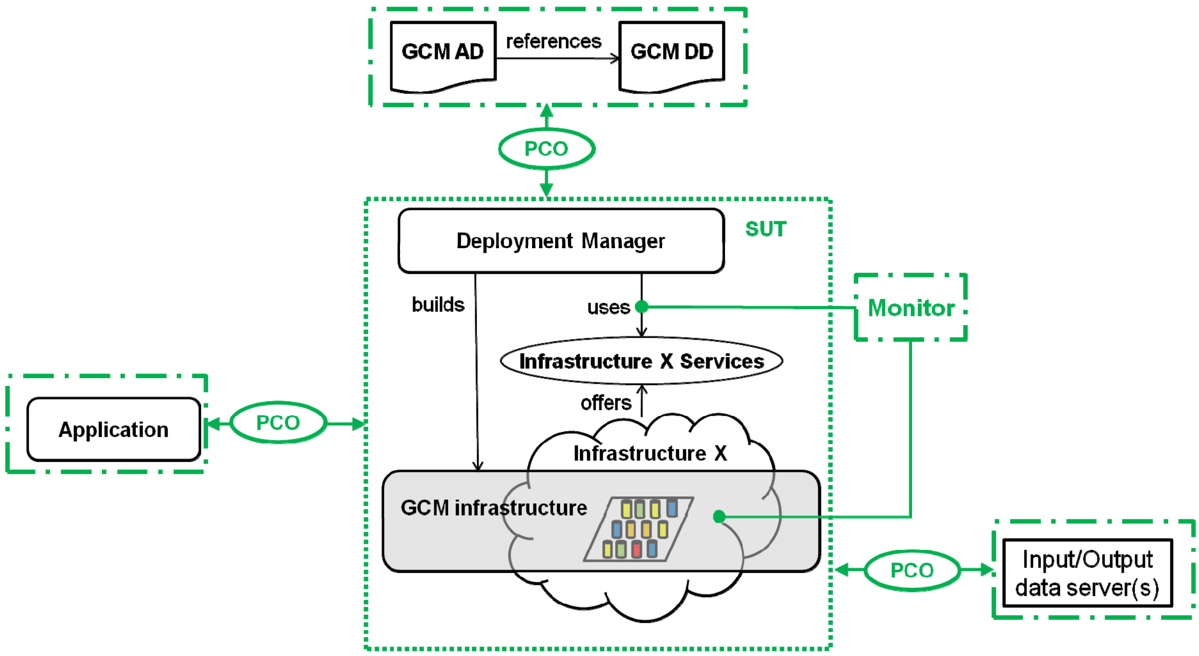

Figure 13: A test architecture for GCM-based deployment

An example test architecture is shown in figure 13. The System Under Test (SUT) consists of the Deployment Manager and one or more infrastructures. The different types of entities that compose the means of testing handle provision of GCM DD and AD files to the deployment manager associated with the infrastructure to be tested, the evaluation of responses from the deployment manager, analysis of the output produced by the application, monitoring of the processing ongoing in the infrastructure during the execution of tests, e.g. the number of processors involved in a computation, the interface(s) between deployment manager, each infrastructure and the input/output servers.

NOTE 1: The interfaces for supplying GCM DD and AD to a deployment manager have not yet been standardized.

NOTE 2: This testing architecture can also be used to access other standards related to the deployment and execution of applications on grid or cloud infrastructures, e.g. an OGSA-BES [i.65] web service based interface between the deployment manager and the infrastructure.

For each test, a specific test application is used to assess the test purpose which is based on the GCM standard. This application should be parameterizable to allow its reuse across multiple tests.

NOTE 3: It is also assumed that all applications associated with tests already physically reside in the SUT.

9.5 Test suite structure and test purposes

9.5.1 Test suite structure

The test suite structure should be based on the requirements specified in GCM DD and AD standards. Firstly, test groups should distinguish tests to be relevant to GCM DD vs. AD. For the GCM DD groups should differentiate between indirect and direct resource access. For the GCM AD groups should differentiate between virtual nodes and data location.

9.5.2 Test purposes

A Test Purpose (TP) specifies how a requirement specified in a specification can be assessed in a given test architecture. Therefore, each TP should at least include a reference to the clause in a specification where the requirement assessed in the test is stated in the standard. In addition, a TP should be dedicated to one aspect of a specific requirement or concept defined in a GCM standard. Each test purpose should be unique and have an identifier reflecting its place in the test suite structure. The formulation of the test purpose should be based on the writing style recommended in [i.71].

Specification of test purposes for GCM is not trivial since the GCM standard defines the specification of deployment information but not an interface for deployment. In case of GCM DD, test purposes should be specified on interface parameters which are common in standardized GCM mappings to different infrastructures. A test purpose should however not be specific to a single mapping. Other areas for test purpose specification are the assessment of different resource access methods (including direct, indirect and bridge) as well as requests for different number of processors and/or virtual machines from one or more infrastructures. In table 5 an example TP for GCM DD is depicted.

Table 5: Example GCM test purpose "Single processor with direct resource access"

TP ID:	TP_GCM_DD_DA_PA_001
Clause Ref:	TS 102 827 [i.45], clause 7.1
SUT role:	Deployment Manager and Infrastructure
Summary:	Ensure that an infrastructure with direct resource access provides a single processor as specified in the GCM DD

In the case of GCM AD, GCM DD information which is referenced should be assumed to be correct and tested. Test purposes here should focus on aspects and parameters only specified as part of the AD, e.g. handling of virtual nodes and input/output data location.

9.5.3 Test descriptions

A Test Description (TD) is a detailed but informal specification of the pre-conditions and test steps needed in order to cover one or more given test purposes. A test description contains the following information:

- **Identifier:**
  Each TD has a unique identifier that relates a test to its group and sub-group.
- **Summary:**
  The TD summary clarifies the purpose of a test. The summary of every TD should be unique.
- **Configuration**
  This field references the test configuration required for the test execution.
- **Specification References**
  One or more clause references to the standard based on which the test purpose has been specified

- **Test application**
  A reference to the test application which is required to execute this test. In case, the application is parameterized the pre-conditions should state constraints on these parameters. Examples for GCM test applications include:

  - Single process batch job.
  - Parallel job.
  - Virtual Node GCM Application.
  - Data Manipulation GCM Application.

- **Pre-test conditions**
  A list of all conditions that have to be fulfilled prior to the execution of a test. These conditions should also help to identify the standardized features that should be supported by the equipment referenced in the test configuration, i.e. to check if a test is applicable for a given equipment.
  Common types of pre-conditions include:

  - GCM descriptor content: GCM DD and/or AD content must fulfil certain requirements,
  - Infrastructure: type of resource access, features required to be supported (e.g. specification of all clock time), and available amount of resource (e.g. at least 2 processors).
  - Test application parameterization: number of processes, process execution time, etc.

- **Test sequence**
  Description of stimuli and observations in a numbered order which expected to be performed by an end user on interfaces offered by the SUT. Each stimulus should be followed by at least one observation.

An example TD for the GCM DD is shown in table 6. This TD describes a test to check if an infrastructure with direct resource access provides a single processor as specified in the GCM DD.

NOTE: Test descriptions for GCM DD should be specified independent of GCM AD.

Table 6: Example GCM test description "Single processor with direct resource access"

Interoperability Test Description		
Identifier:	TD_GCM_DD_DA_PA_001	
Summary:	Ensure that an infrastructure with direct resource access provides a single processor as specified in the GCM DD	
Configuration:	Single Infrastructure or single Infrastructure with a bridge	
Specification References:	GCM DD clause 7.1	
Test Application:	Single process batch job	
Pre-test conditions:	• Infrastructure provides direct resource access • GCM DD contains a direct group description with `hostList` containing one host and host description with `hostCapacity=1` for the infrastructure • Infrastructure has a processor available for use	
Test Sequence:	**Step**	**Description**
	1	User loads the GCM DD and starts the test application on the infrastructure using the deployment manager
	2	Verify that the infrastructure has created and executed the process
	3	Verify that returned application output is correct

9.5.4 Test execution

Test execution requires the evaluation of the applicability of each test (description) to all of the equipment part of the SUT. To speed up this process a GCM Implementation Conformance Statement (ICS) should be established to allow infrastructure providers to specify supported features prior to a test execution and support automatic test selection. PIXIT can be used capture infrastructure specific aspects of a GCM DD such as the access details to an infrastructure, resource identifiers, etc.

A test should be executed if all of its pre-conditions have been ensured.

A test should not be executed and recorded as being not applicable if any of its pre-conditions are not met by the one (or more) equipment part of the SUT.

History

Document history		
V1.1.1	October 2009	Publication