Experiences and Requirements for Interoperability between HTC- and HPC-driven e-Science Infrastructures

Morris Riedel, Achim Streit, Daniel Mallmann, Felix Wolf and Thomas Lippert

Abstract Recently, more and more e-science projects require resources in more than one production e-science infrastructure, especially when using HTC and HPC concepts together in one scientific workflow. But the interoperability of these infrastructures is still not seamlessly provided today and we argue that this is due to the absence of a realistically implementable reference model in Grids. Therefore, the fundamental goal of this paper is to identify requirements that allows for the definition of the core building blocks of an interoperability reference model that represents a trimmed down version of OGSA in terms of functionality, is less complex, more fine-granular and thus easier to implement. The identified requirements are underpinned with gained experiences from world-wide interoperability efforts.

1 Introduction

Many applications take already advantage of a wide variety of e-science infrastructures that evolved over the last couple of years to production environments. Along with this evolution we observed still slow adoption of the Open Grid Services Architecture (OGSA) concept originally defined by Foster et al. in 2002 [8]. While OGSA represents a good architectural blueprint for infrastructures in general, we argue that the scope of OGSA is actually to broad to be realistically implementable for today's production e-science infrastructures in particular. This has mainly two reasons. First, the process of developing open standards that are conform to the whole OGSA ecosystem take rather long, including the precise specification of all the interconnections of these services and their adoption by the respective middleware providers. Second, the launch of OGSA-conform components within production e-science infrastructures take rather long. Although some aspects of OGSA are (or become) relevant to the e-science infrastructures (execution management

Morris Riedel, Co-Chair of Grid Interoperation Now (GIN) Group of the Open Grid Forum (OGF)
Jülich Supercomputing Centre, Forschungszentrum Jülich GmbH, e-mail: m.riedel@fz-juelich.de

and service oriented concepts), many services are still very inmature (e.g. advance reservation, service level agreements, virtualization, fault detection and recovery) or many concepts have not been widely adopted in Grid middleware technologies (e.g. service lifetime management, service factories, or notification patterns).

The absence of a realistically implementable reference model is diametral to the fundamental design principles of software engineering and has thus lead to numerous different architectures of production e-science infrastructures and their deployed technologies in the past. To provide some examples, the Enabling Grids for e-Science (EGEE) [17] infrastructure uses the gLite middleware, the TeraGrid [21] infrastructure uses the Globus middleware, the Distributed European Infrastructure for Supercomputing Applications (DEISA) [1] uses the UNICORE middleware, the Open Science Grid (OSG) [19] uses the Virtual Data Toolkit (VDT) and NorduGrid [23] uses the ARC middleware. Most elements of these technologies and infrastructures are not interoperable at the time of writing because of limited adoption of open standards and OGSA concepts.

The lack of interoperability is a hinderence since we observe a growing interest in conveniently using more than one infrastructure with one client that use interoperable components in different Grids. Recently, Riedel et al. [10] provided a classification of different approaches of how to use e-science infrastructures. Among simple scripts with limited control functionality, scientific application plug-ins, complex workflows, and interactive access, there is also infrastructure interoperability mentioned as one approach to perform e-science. Many e-scientists would like to benefit from interoperable e-science infrastructures in terms of having seamless access to a wide variety of different services or resources. In fact many scientific projects raise the demand to access both High Throughput Computing (HTC)-driven infrastructures (e.g. EGEE, OSG) and High Performance Computing (HPC)-driven infrastructures (e.g. DEISA, TeraGrid) with one client technology or Web portal.

Although one goal of OGSA is to facilitate the interoperability of different Grid technologies and infrastructures in e-science and e-business, we state that the requirements for interoperability in e-science infrastructures have to be specified much more precisely than within OGSA. Therefore, this paper defines a set of requirements based on lessons learned obtained from interoperability work between production e-science infrastructures. The goal is to identify a suitable set of requirements to definde the necessary building blocks for a reference model that is much closer oriented towards the interoperability of production e-science infrastructures than OGSA. This reference model should not replace OGSA but rather trim it down in functionality by dropping several parts of it and refining other parts that are mostly relevant to interoperability of e-science infrastructures.

History of computer science shows that often complex architectures were less used than their trimmed down versions. For instance, the complex SGML was less used than its smaller version XML, which was less complex and well-defined and thus fastly become a de-facto standard in Web data processing. Also, the ISO/OSI reference model originally consisted of seven layers, while its much more successful trimmed down version TCP reference model become the de-facto standard in networking. We argue that the same principles can be applied with OGSA by defining a more limited, but more usable reference model. This becomes also increasingly important in the context of economic contraints since the rather huge OGSA requires massive amounts of maintenance while our idea of a reference model should significantly reduce these maintenance costs by providing only a small subset of functionality, but this in a well-defined manner.

This paper is structured as follows. Following the introduction, the scene is set in Section 2 where we list some of our interoperability projects that helped to identify specific requirements for interoperable e-science infrastructures in Section 3. A survey of related work is described in Section 4, while this paper ends with some concluding remarks.

2 Experiences in Interoperability

This section gives insights into several important interoperability projects that provided valuable lessons learned in terms of interoperability between many production e-science infrastructures. Lessons learned and gained experiences out of these projects lay the foundation for our requirement analysis in Section 3.

The OMII-Europe project [5] initially started to work on an e-science infrastructure interoperability use case application of the e-Health community area named as the Wide In Silico Docking on Malaria (WISDOM) project [9]. More recently, this work is continued in DEISA and collaboration with EGEE. The WISDOM project aims to significantly reduce the time and costs in drug development by using in silico drug discovery techniques.

Technically speaking, the overall scientific workflow can be splitted in two parts as described in Riedel et al. [4]. The first part uses the EGEE infrastructure for large in silico docking, which is a computational method for the prediction of whether one molecule will bind to another. This part uses HTC resources in EGEE with so-called embarassingly parallel jobs do not interact with each other. Applications that are used in this part of the workflow are AutoDock and FlexX that are both provided on the EGEE infrastructure. The output of this part is a list of best chemical compounds that might be potential drugs, but do not represent the final solution.

The second part uses the outcome of the first part of the scientific workflow and is concerned with the refinement of the best compound list using molecular dynamics (MD) techniques. For this part, the e-scientists use massively parallel resources in DEISA with the highly scalable AMBER MD package. All in all, the goal of this interoperability application is to accelerate drug discovery using EGEE and DEISA together in the in silico step before performing in vitro experiments.

The framework that enabled the interoperability in this application lead to several job and data management requirements that are listed in the next section. Also, this application can be actually saen as an example for a whole class of interoperability applications that require access to both HTC and HPC resources. Similiar activities within the same class are interoperability efforts performed between the EUFORIA project [7], DEISA, and EGEE. Also e-scientists of the EUFORIA project and thus members of the known ITER community require access to HTC resources (via EGEE) for embarassingly parallel fusion codes and access to HPC resources (via DEISA) for massively parallel fusion codes. The lessons learned from this interoperability project are similiar to the ones in WISDOM, but slightly different in terms of security and workflow settings. This is due to the fact that the e-scientists of EUFORIA would like to use their own Kepler workflow tool.

Other experiences in interoperability have been gained in interoperability work between the EU-IndiaGrid project [11], OMII-Europe, EGEE, and DEISA. The EU-IndiaGrid project works together with specialists of DEISA to enable interoperability between the Indian Grid GARUDA, EGEE and DEISA. Finally, Riedel et al. describes in [2] many different activities of the Grid Interoperation Now (GIN) group of the Open Grid Forum (OGF). All these activities and their lessons learned also contributed to the identification of requirements in the following Section.

3 Requirements for Interoperability

The experiences and lessons learned from numerous international interoperability projects and efforts lead to several specific requirements for the interoperability between HTC- and HPC-driven infrastructures. First and foremost, the cooperation between Grid technology providers (i.e. gLite, Globus, UNICORE, ARC) and deployment teams of different infrastructures (e.g. EGEE, TeraGrid, DEISA, NorduGrid) represents an important social requirement that is often highly underrated. We argue that the right set of people from different Grid technology providers have to sit together with different infrastructure deployment teams to discuss technical problems in order to achieve interoperability in terms of technologies in general and thus of infrastructures in particular. To ensure outreach to the broader Grid community, outcome of this cooperation should be fed back to OGF to encourage discussions in the respective working groups.

To provide an example, Grid deployment teams from the infrastructures EGEE, DEISA and NorduGrid as well Grid technology providers such as gLite, UNICORE, and ARC had a meeting at CERN to discuss how the job exchange interoperability could be significantly improved within Europe. The result of this workshop was given as an input to the OGF GIN group and will be further discussed with related OGF standardization groups.

Technical requirements are illustrated in Figure 1, which indicates that the requirements stated in this section can be found in four different layers. The plumbings for interoperability are orthogonal to all these layers and thus represent a mandatory requirement. The term plumbings refers to the fact that they basically affect any layer significantly although often realized behind the scenes and thus not visible to end users. This section highlights the requirements of the plumbings as well as the job and data management layer. We argue that the network layer is already interoperable mainly through GEANT and thus not considered to be important in our requirement analysis. Furthermore, we state that the different infrastructures and resource layers (i.e. HTC Grid, Cloud, HPC Grid) are given in our analysis as unchangeable elements. In this context, we argue that rather commercial-oriented Clouds like the Amazons Elastic Computing Cloud (E2C) [15] are currently out of the scope of scientific use cases and thus not part of our analysis but listed to provide a complete and realistic picture.

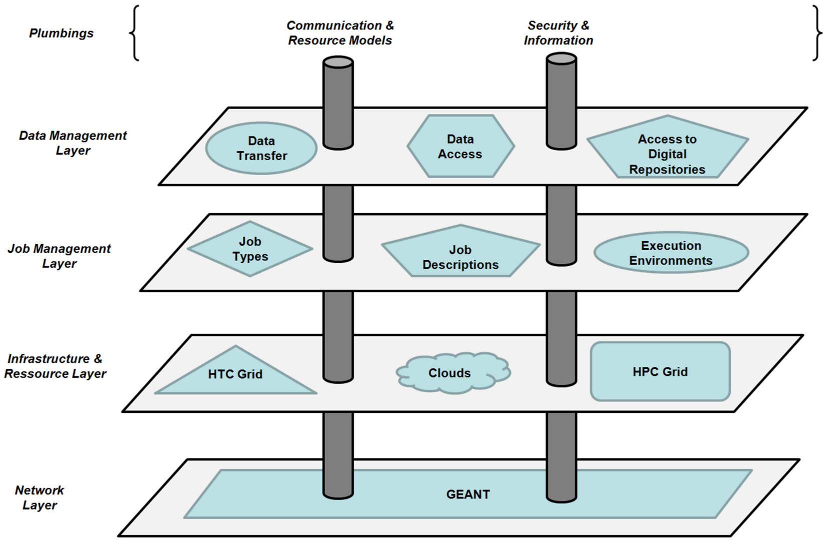

Fig. 1 Interoperablity requirements within four different layers, while two (often behind the scenes realized) plumbings are orthogonal to them because they significantly affect every layer.

3.1 Plumbings

Today, the most production infrastructures use proprietary protocols between their components and thus communication between components of infrastructure A and B is not possible. To provide an example, DEISA deployed UNICORE 5 that uses the proprietary UNICORE Protocol Layer (UPL) [6] for communication. EGEE deployed gLite that also uses proprietary protocols such as the WSMProxy protocol between the User Interface (UI) [20] and Workload Management System (WMS) [20]. Hence, the deployed components of both Grids cannot interact with each other. As a consequence, the communication technology between major elements must be the same, especially when connecting job and data management technologies of different Grids. The Web services technology based on HTTPS and Simple Object Access Protocol (SOAP) [12] are good foundations to exchange commands and control information, but not to perform large data transfers. In addition, the underlying resource model representation should be compliant with the WS-Resource Framework (WS-RF) [3] or WS-Interoperabiltiy [13] stack to enable common addressing techniques such as WS-Addressing [14].

Even if the communication relies on WS there is still a wide variety of deployed security models in the production infrastructures starting from different certificate types (e.g. full X.509 vs. X.509 proxies) to different delegation techniques (e.g. proxy delegation vs. explicit trust delegation) that do not allow for interoperable applications. Also, many different security technologies are implemented (e.g. VOMS proxies [18] vs. SAML-based VOMS [16]) or proprietary definitions (e.g. authorization attributes) are used. Therefore, another important requirement is to agree on a common security profile that consists of numerous relevant security specifications to enforce security policies. In addition to this profile, there must be a detailed specification of VO attributes and delegation contraints/restrictions encoding that does not exist today, but are required to ensure fine-grained authorization interoperability (including delegation) along the different infrastructures.

Finally, we observe that many production Grids rely on different information schemas (e.g. GLUE vs. CIM) and information systems (e.g. MDS, BDII, CIS). Up-to-date information is crucial for interoperability with impacts on job and data management technologies. To provide an example, the information where a job with a predefined amont of CPUs defined can be executed should be obtained from an information service that relies on an information schema. To date, the production infrastructures cannot directly exchange information to enable interoperability due to different deployed mechanisms. As a consequence, the same information schema (.e.g. GLUE2) must be used in the technologies deployed in the infrastructures. In addition, information services need standard mechanisms to be queried for this kind of information.

3.2 Job Management

Figure 1 illustrates the job management layer with three different requirement elements. According to the different types of Grids within the infrastructures and resources layer, we also have different types of computational jobs that should be executed on the infrastructures. This in turn leads to the development of different technologies that are used for computational job submission. HTC Grid infrastructures mostly run cycle farming jobs (also named as embarassingly parallel jobs) that do not require a efficient interconnection between the CPUs. As a consequence, middleware packages deployed on these infrastructures (e.g. gLite) use brokering technologies (e.g. WMS) that submit the job on behalf of the user to any free resource for execution. In contrast, HPC Grid infrastructures mostly run massively parallel jobs that do require an efficient interconnection between cpus and thus cannot be run on any PC pool or similiar farming resource. In turn HPC-driven middleware (e.g. UNICORE) enables end users to manually choose one particular HPC resource due to the general massively-parallel job type, but also because often HPC jobs are even tuned to run on a specific HPC architecture (i.e. memory or network topology requirements). In addition, HPC applications are often compiled on one specific platform and not easy transferable to another. Therefore, we raise the requirement that both cycle farming jobs and massively parallel jobs should be supported in technologies that are deployed on interoperable infrastructures.

Many technologies that are deployed on production Grids still rely on proprietary job description languages derived from the different requirements of the underlying resource infrastructure. In other words it is not possible to exchange jobs directly between different e-science infrastructures although many of the base functionality is the same (e.g. excutable location). To provide an example for different job description languages, TeraGrid deployed Globus, which uses the Resource Specification Language (RSL) [22], DEISA deployed UNICORE, which uses the Abstract Job Object (AJO) [6] and gLite, which uses the Job Definition Language (JDL) [20]. Hence, the seamless submission of computational jobs from one client to different infrastructures can only be ensured when all technologies are compliant with the same description language such as the Job Submission and Description Language (JSDL) [24] and their specification extensions like Single Program Multiple Data (SPMD). Hence, the middleware should provide interfaces that accept jobs in this description (e.g. OGSA-BES [26]). Although the progress of JSDL is quite far the emerging amount of extensions break again the obtained interoperability in terms of job submission. To support even more complex jobs we also state the requirement for a common workflow language among the Grid middleware technologies, but research in this area reveals that this seems not to be achievable because of the huge differences in definition of complex application jobs.

The last requirement of the job management layer is related to the execution of the computational job itself. At the time of writing there is no agreement about a common execution environment that is needed when performing cross-Grid jobs that require for instance special visualization libraries or rather general message passing interface libraries. In addition, many applications make use of the environment variable provided via the Grid middleware systems in the started process at the HTC or HPC resource. Since there is no common definition available on this variable setting, jobs cannot be run on resources that provide access via a different Grid middleware system. In this context, virtual machines are not still widely adopted in production infrastructures and early analysis reveals that the performance does not satisfy end-user demands especially in HPC Grids. As a consequence, we raise the demand for the definition of an execution environment profile that has been recently started within GIN of OGF (i.e. Worker Node Profile).

3.3 Data Management

Figure 1 illustrates three major requirements at the data management layer. Starting with the lack of a common defined set of data transfer techniques, we observe that the most e-science infrastructures adopt different concepts. Many Grid infrastructures (TeraGrid, EGEE, OSG) adopt GridFTP for large data transfers while other infrastructures (e.g. DEISA) rather rely on distributed parallel file systems (e.g. IBM Global Parallel File System). While many other technologies exist such as secure FTP, ByteIO, and parallel HTTP, we raise the demand for an interface that abstracts from the different concepts and provides one standard interface for data transfer. Recently, OGF worked on the Data Movement Interface (DMI) that seems to satisfy this requirement but which is still work in progress, especially in terms of third-party credential delegation that is used, for instance, for GridFTP transfers underneath.

We also state the requirements need to explicitly define the link between job description and data transfer specifications that are required when jobs are using data staging in/out functionalities. That means all computational jobs that may even be submitted through OGSA-BES compliant interfaces and use JSDL would fail since the data staging definitions within the JSDLs are not defined and thus are implemented in proprietary ways in the Grid middlewares. More recently, work in OGF has been performed on the HPC File Staging Profile (FSB) that aligns OGSA-BES interfaces with FTP-based data transfers. However, since FTP is also not widely adopted within e-science infrastructures, we still require a more sophisticated set of profiles that support a wider variety of data transfer technologies.

Closely aligned with data transfer are data access and storage technologies such as SRM [28] implementations (e.g. DCache, Storm, Castor). That means several functions of SRM interface implementations (e.g. moveTo operation) use the underneath data transer technologies (e.g. GridFTP) to transfer the data. The same is valid for WS-DAIS [29] implementations (e.g. OGSA-DAI, or AMGA Metadata Catalog) that rely on GridFTP for the transport of large blob files when, for instance, relational databases are used. In addition to the use of SRM interface implementations in infrastructures (e.g. EGEE) and WS-DAIS interface implementations in infrastructures (e.g. NGS), several infrastructures also rely on SRB or iRods (e.g. OSG) that neither provide a SRM nor WS-DAIS interface. In order to use all these technologies transparently during cross-Grid job submissions with data staging, the listed data access technologies must provide a standard interface such as SRM or WS-DAIS. Also, we require a precise definition how these interface can be used during data staging using JSDL-compliant jobs.

More recently, we also indicated end-user requirements to support the access to digital repositories that deliver content resources such as any form of scientific data and output, including scientific/technical reports, working papers, articles and original research data. In this context, metadata is important to store/extract data conveniently. As a consequence, we require middleware that allows for the attachment/detachment of semantics to computed data on the infrastructures before any data is stored. Also, we require different services such as search, collection, profiling, or recommendation that bridge the functionality between Grid middleware and digital repositories. In this context, the DRIVER project [25] is working on a good step towards the right direction in inter-networking European scientific respositories. But the precise links between deployed job and data management technologies and digital repositories are not defined yet although required more in more in scientific use cases that use computational resources within Grids to perform efficient queries for knowledge data in repositories.

4 Related Work

As already mentioned in the introduction of this paper, the OGSA initially defined by Foster et al. [8] defines an architecture model taking many requirements from e-science and e-business into account. Our work is motivated by lessons learned from e-science infrastructure interoperability efforts that raise the demand for an interoperability reference model that is trimmed down in functionality compared to OGSA, is less complex than OGSA, and thus realistically to implement and to specify in more detail than OGSA.

Another reference model that is related is the Enterprise Grid Alliance (EGA) reference model [27]. The goal of this model is to make it easier for commercial providers to make use of Grid computing in their data centers. This model comprises three different parts: a lexicon of Grid terms, a model for classifying the management and lifecycles of Grid components, and a set of use cases demonstrating requirements for Grid computing in businesses. In contrast to our work, this reference model is rather focussed on business requirements, while we take requirements mainly from the scientific community into account.

5 Conclusion

In this paper we raised the demand for an interoperability reference model based on experiences and lessons learned from many interoperability projects. We can conclude that the requirements identified from these efforts lead to a reference model that can be seen as a trimmed down version of the OGSA in terms of functionality and complexity. The requirements defined in this paper can be used to specify the core building blocks of a realistically implementable reference model for interoperable e-science infrastructures based on experiences and lessons learned over the last years in GIN and other interoperability projects. We foresee that many building blocks of this reference model are already deployed on the infrastructures and only minor additions have to be done in order to achieve interoperable e-science infrastructures. The definition of this reference model is work in progress.

References

1. Distributed European Infrastructure for Supercomputing Applications (DEISA) http://www.deisa.eu, Cited 10 October 2008
2. Riedel, M. et al.: Interoperation of World-Wide Production e-Science Infrastructures, accepted for Concurrency and Computation: Practice and Experience Journal, (2008)
3. The Web Services Resource Framework Technical Committee http://www.oasis-open.org/committees/wsrf/, Cited 10 October 2008
4. Riedel, M. et al.: Improving e-Science with Interoperability of the e-Infrastructures EGEE and DEISA. In: Proceedings of the 31st International Convention MIPRO, Conference on

Grid and Visualization Systems (GVS), Opatija, Croatia, ISBN 978-953-233-036-6, pages 225–231 (2008)

5. The Open Middleware Infrastructure Institute for Europe
http://www.omii-europe.org, Cited 10 October 2008
6. Streit, A. et al.: UNICORE - From Project Results to Production Grids, Advances in Parallel Computing 14, Elsevier, 357–376 (2005)
7. EU Fusion for ITER Applications
http://www.euforia-project.eu, Cited 10 October 2008
8. Foster, I. et al.: The Physiology of the Grid. In: Grid Computing - Making the Global Infrastructure a Reality, John Wiley & Sons Ltd, pages 217–249 (2002)
9. Wide In Silico Docking on Malaria Project
http://wisdom.eu-egee.fr, Cited 10 October 2008
10. Riedel, M. et al.: Classification of Different Approaches for e-Science Applications in Next Generation Computing Infrastructures. Accepted for publication in: Proceedings of the e-science Conference, Indianapolis, Indiana, USA (2008)
11. Wide In Silico Docking on Malaria Project
http://www.euindiagrid.org/ , Cited 10 October 2008
12. Gudgin, M. et al.: SOAP Version 1.2 Part 1: Messaging Framework, W3C Rec. (2003)
13. WS-Interoperability (WS-I)
http://www.ws-i.org , Cited 10 October 2008
14. Box, D. et al.:WS-Addressing (WS-A), W3C Member Submission. (2004)
15. Amazons Elastic Computing Cloud (EC2)
http://aws.amazon.com/ec2, Cited 10 October 2008
16. Venturi, V. et al.: Using SAML-based VOMS for Authorization within Web Services-based UNICORE Grids, In:Proceedings of 3rd UNICORE Summit 2007 in Springer LNCS 4854, Euro-Par 2007 Workshops: Parallel Processing, pages 112–120 (2007)
17. Enabling Grids for e-Science Project
http://public.eu-egee.org, Cited 10 October 2008
18. Alfieri, R. et al.: From gridmapfile to voms: managing authorization in a grid environment, In: Future Generation Comp. Syst., 21(4):, pages 549–558 (2005)
19. Open Science Grid (OSG)
http://www.opensciencegrid.org, Cited 10 October 2008
20. Laure, E. et al.: Programming The Grid with gLite, Computational Methods in Science and Technology, Scientific Publishers OWN, 33–46 (2006)
21. TeraGrid
http://www.teragrid.org, Cited 10 October 2008
22. Foster, I. et al.: Globus Toolkit version 4: Software for Service-Oriented Science, In: Proceedings of IFIP International Conference on Network and Parallel Computing, LNCS 3779, pages 213–223 (2005)
23. NorduGrid
http://www.nordugrid.org/ , Cited 10 October 2008
24. Anjomshoaa, A. et al.: Job Submission Description Language (JSDL) Specification, Version 1.0, OGF GFD 136
25. DRIVER Project
http://www.driver-repository.eu , Cited 10 October 2008
26. Foster, I. et al.: OGSA - Basic Execution Services, OGF GFD 108
27. EGA Reference Model
http://www.ogf.org/documents/06322r00EGA_RefMod-Reference-Model.pdf, Cited 10 October 2008
28. Sim, A. et al.: Storage Resource Manager Interface Specification Version 2.2, OGF GFD 129
29. Antonioletti, M. et al.: Web Services Data Access and Integration - The Core (WS-DAI) Specification, Version 1.0, OGF GFD 74