Grid Computing in Europe: From Research to Deployment

Dr. Rüdiger Berlich[1] **Dr. Marcel Kunze**[1] **Dr. Kilian Schwarz**[2]

[1]Forschungszentrum Karlsruhe Hermann-von-Helmholtz-Platz 1 76344 Eggenstein-Leopoldshafen / Germany
[2]Gesellschaft für Schwerionenforschung mbH Planckstrasse 1 64291 Darmstadt / Germany

Email: `ruediger.berlich@iwr.fzk.de`

Abstract

Grid Computing has, over the past few years, matured sufficiently to make it a viable solution for real-world problems. However, there are many different toolkits today that allow to build a Grid environment. And while this wealth of different solutions may offer techniques applicable to the widest possible range of computational problems, their very availability directly contradicts the inherent promise of the "World Wide Grid" to offer a compatible and standardised infrastructure. The European Union project EGEE[1] aims at a consolidation of existing efforts and will assist in the deployment of the resulting Grid middleware gLite by offering support and training to new users, both in academia and industry.

This paper introduces the historical development and present scope of European Grid projects and middlewares leading up to EGEE. Beyond its European scope, the paper tries to clarify their relationships to worldwide initiatives and to give an insight into the lessons learned during the development. The paper concludes with the introduction of a national Grid deployment project – the German D-Grid initiative.

1 Overview

2007 will see the advent of the **L**arge **H**adron **C**ollider[2] (R. Ostojic 2001), a particle accelerator built in the tunnel of the now decommissioned **L**arge **E**lectron **P**ositron Collider (LEP) at CERN[3] near Geneva, Switzerland. In contrast to LEP, LHC will collide protons with protons or, alternatively, heavy ions, such as lead. As a result of the collision, the kinetic energy and rest-mass of these particles is transformed into other particles that can be detected in large detector systems[4] built around the interaction points. One collision is usually referred to as an "event". The peculiarities of these events, together with high particle-energies and -beam intensities, lead to data rates unprecedented by todays most modern particle physics experiments. The estimated amount of data emanating from all four LHC experiments, ALICE (F. Carminati et al. 2004), Atlas, CMS and LHCb will exceed 4 Petabyte per year, or $4 * 10^{15}$ bytes. It must be stressed that this represents only the data that is available for processing and storage. The data measured in the detectors themselves exceeds the data going into processing by a factor of 10 million.[5]

It is obvious that, in order to be able to process this huge amount of data, new techniques had to be developed well in advance of the start-up of LHC. For example, no widespread knowledge is available in the design of storage systems[6] capable of providing random, simultaneous access to thousands of scientists in a multitude of different, worldwide distributed locations. Data access must be transparent to users, should be efficient, and may not neglect security considerations.

In addition to the sheer need for storage and computing power, the availability of ready-to-use computing resources in different geographic locations, coupled with the political force to invest money locally - investments in LHC range in the billion dollar area - do not allow for centralised resources.

Similar problems as in particle physics exist in many areas, ranging from distributed medical databases such as MammoGrid (S.R. Amendolia et al. 2004) to the processing of satellite data (ENVISAT of the European Space Agency). Similar security and transparency requirements also exist in less data-intensive areas such as the coupling of globally distributed radio telescopes (Virtual Observatory).

Many different projects tried to address the technical challenges common to most Grid flavours[7].

2 Basic Building Blocks

2.1 Globus

In 1995, 17 supercomputer centers, virtual reality laboratories and development centers joined forces to demonstrate the distributed execution of a number of supercomputer applications during *Supercomputer '95* in San Diego. The software infrastructure of the project, then called I-WAY[8] (I. Foster et al. 1997), came back to life in 1996, when new funding allowed further development. This early Grid middleware, later called Globus (I. Foster, C. Kesselman 1997), has been co-developed by two veterans of Grid Computing, Ian Foster of Argonne National Laboratory and Carl Kesselman of University of South Carolina. Their book "The Grid, Blueprint for a new Comput-

[1]"**E**nabling **G**rids for **E**Science in **E**urope", or more recently just "**E**nabling **G**rids for **E**Scienc**E** in order to reflect EGEE's geographically broadened scope

[2]LHC

[3]Centre Europeenne pour la Recherche Nucleaire

[4]often simply referred to as "experiment"

[5]In the case of the CMS experiment, the data measured in the detector itself will be in the range of 1 PB/s, which, by the use of triggers selecting only interesting data, is reduced to about 100 MB/s.

[6]comprising both hard- and software

[7]It is important to note, that many middleware packages carry very similar names to the projects they emanated from.

[8]**I**nformation **W**ide **A**rea **Y**ear)

ing InfraStructure", published in 98, also crafted the name "The Grid" (I. Foster, C. Kesselmann 1998).

In version 2, which is still the foundation of many middleware packages in particle physics, the functionality of Globus is, by design, limited. In itself it can be regarded more as a toolkit, from which more sophisticated Grids can be constructed, than as a ready-to-use Grid solution.

While Globus is capable to act globally, its users must know the target resources address. Usage of Globus-2 thus does not exhibit high levels of automation.

Globus did however add ground-breaking functionality in many areas, with the Grid Security InfraStructure (GSI), responsible for authentication and authorisation, just being an example.

Meanwhile newer versions of Globus are available (version 3 and 4). However, due to a shift in paradigms in these new versions Globus 2 is still widely used in many Grid projects.

2.2 Importance of Resource Brokers

A must-have component of a Grid that has intentionally been left out in Globus 2 is a Resource Broker (RB). Its duty is to match job requirements with resource capabilities and to assign jobs to resources accordingly. Users then do not need to choose a specific target machine anymore, nor do they need to know about geographic location or ownership (as long as their credentials match the target resource's requirements).

According to the Grid definition of Ian Foster, co-founder of the Globus Alliance, "*Grid Computing is coordinated resource sharing and problem solving in dynamic, multi-institutional virtual organisation*". In this context, participating institutes and compute centers will usually be geographically distributed, with each one contributing its own dedicated computing resources. This results in an inhomogeneous, non-local computing environment.

By coupling these diverse resources, it is the Resource Broker that assembles a Grid from separate resources, such as clusters. A Grid can thus often be compared to a Meta-Cluster, or in other words a cluster of clusters.

While a Grid Resource Broker must be able to interface to local batch submission systems, such as PBS[9] or LSF[10], a batch submission system only acts locally in its own cluster resources. It can thus not be compared to a Resource Broker, which acts globally across inhomogeneous resources.

2.3 Workload Management Systems

Even a highly specialised workload management system like Condor (T. Tannbenbaum et al. 2002), which provides the possibility to compare job requirements with the capabilities of the compute nodes[11], operates only on a local cluster. In the case of Condor, though, an extension via Globus is possible, thus adding the capability to Grid-enable Condor (then called Condor-G). Using Globus as the basis for Grid functionality implies, however, that no automated selection of target resources can be done.

NIMROD (Abramson et al. 2000), another tool, manages the execution of parametric studies across distributed computers in a local area network. It uses its own parametric modeling language for description.

NIMROD/G extends NIMROD with wide-area execution capabilities by mapping of individual computations to appropriate remote sites. Information about their physical characteristics and availability are available from the Globus directory service (MDS). Local resources like batch queues are accessed via the Globus Resource Allocation Manager (GRAM). Before submitting any jobs to a specific cluster, a Nimrod Resource Broker (NRB) is started on the local cluster, providing additional capabilities beyond that of GRAM like, for example, file staging.

3 The LHC Computing Grid

Early proposals to address the technical challenges of LHC arose from the MONARC[12] study, which aimed at the creation of feasible models for the computing of LHC experiments and the establishment of guidelines for the experiments in building their distributed computing models. One of MONARC's successes was the establishment of a multi-tier model (see figure 1), whose essentials are still valid today. In this model, distributed data- and compute-centers are presented in a hierarchy, with different tasks depending on their corresponding level in the hierarchy[13].

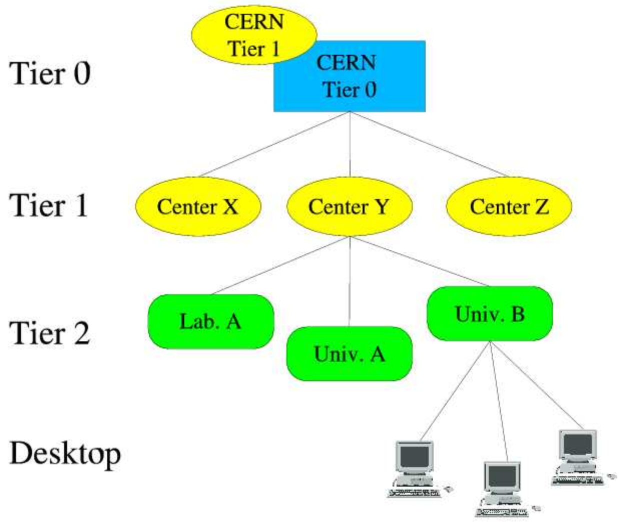

Figure 1: One of MONARC's successes was the establishment of a multi-tier model, whose essentials are still valid today.

Today the **L**HC **C**omputing **G**rid[14] (Les Robertson 2004) has taken over from MONARC as the main effort to make the LHC computing model a reality. Like MONARC proposed, LCG builds on a multi-tier model. In this model, the Tier-0 center at CERN is responsible for the initial reconstruction and storage of raw events and their distribution to Tier-1 centers. It can take over additional duties, where required. Tier-1 centers, themselves featuring large compute clusters and storage capacities[15], perform data-heavy analysis, reprocessing of data

[9] Portable Batch Submission System

[10] Load Sharing Facility

[11] In Condor, a resource's capabilities are published using the "Condor Class Ads"

[12] **M**odels **O**f **N**etworked **A**nalysis At **R**egional **C**enters

[13] With the advent of todays fast networks this distinction becomes more and more blurred, though, and a hierarchy is introduced more through available services and through the regional importance of a compute center.

[14] "LCG"

[15] As an example, the GridKa cluster at Forschungszentrum Karlsruhe / Germany, a Tier-1 center in the LCG, at the time of writing of this document features 1070 CPUs and 220 Terabytes of harddisk space, coupled through a highly efficient parallel filesystem – GPFS

and also host regional support efforts. Tier-2 centers provide managed disk storage, perform simulation of particle physics events and furthermore support end user analysis by providing computational power.

It must be stressed that LCG is a deployment project rather than a development project: it identifies, tests, debugs and packages solutions to support LHC's demanding computing requirements. Large-scale "data challenges" aim at identifying problem areas, so they can be fixed well in advance of the start-up of LHC.

LCGs current middleware, also commonly called LCG[16], in turn builds on the **E**uropean **D**ata **G**rid middleware EDG, but incorporates a number of additional components. Most notably, LCG uses components of the **V**irtual **D**ata **T**oolkit (VDT), such as Condor and iVDGL.

3.1 The European Data Grid

Different Grid projects, like the European Union project "**E**uropean **D**ata **G**rid" (EDG), tried to extend Globus with middleware services of higher functionality, such as a Resource Broker.

EDG, a three-year project, started in 2001 with a major focus on development, and finished with a successful final EU-review in March 2004. EDG is supported by 21, mostly European, organisations both from industry and science and was meant to cover biological and medical image processing as well as earth observation alongside the more "traditional" high-energy physics applications.

As quoted from the project presentation, EDG's main goal was to "develop and test the technological infrastructure that will enable the implementation of scientific collaboratories, where researchers and scientists will perform their activities regardless of geographical location". EDG comprised 12 work packages. Of special importance for the further development of the European Grid infrastructure are work packages 1 (Resource Broker) and 2 (Replica Management Tools). Jointly they contributed to EGD's middleware, like the project itself commonly called "EDG".

The Replica Location Service (RLS), in conjunction with the Replica Metadata Catalogue (RMC), maps logical filenames (LFNs) to physical filenames (PFN) via a Global Unique Identifier (GUID).

The logical filename is a name attributed to a file by the Grid user, whereas the physical filename additionally contains the storage location. More than one replica of the same file may exist, resulting in several PFNs. However, all replicas can be addressed through one globally unique identifier (GUID). One GUID can in turn be referred to by many different logical filenames.

RLS and RMC are based on a relational database architecture. Command line access to the database is implemented through a Java client. Similarly, commands like *edg-job-submit* are implemented in the Python language. As the corresponding virtual machine or interpreter must be loaded for each command, execution times can be relatively large. Given that most scientists will want to submit several thousand jobs simultaneously, this delay can present a problem.

The EDG middleware implements all fundamental grid services, such as information services, resource discovery and monitoring, job submission and management, brokering, data- and resource management.

The EDG project can undoubtedly be credited with having led to a quantum leap in the adoption and development of Grid techniques in Europe. Despite of all of its success there do of course remain areas to be developed further. Criticism of the EDG project often relates to the push architecture of its middleware,

In a push architecture, the Resource Broker polls all Computing Elements[17] to find out about the load of the worker nodes. If this load is sufficiently low, new jobs are assigned to them. In this scenario, if the load has changed significantly during the time span in between two polling cycles, jobs could be assigned to inappropriate nodes.

In contrast to the push model, a pull architecture leaves the decision about the best time to start a new Grid job to the computing element itself instead of the Resource Broker.

EDG's middleware is still based on Globus 2, which is likely to go out of support on a short time scale. Versions 3 and 4 of Globus are based on Grid- and Web services, which would require a redesign of the EDG middleware, if Globus 4 was to be used as its basis. This makes it likely that there will be the demand for a more modern middeware implementation soon.

The successor of the LCG-2 middleware will be called gLite, which will be based on Web services and developments of other Grid projects that are described in the following.

3.2 AliEn

The **Ali**ce Grid **En**vironment "AliEn" (P.Saiz et al. 2003) is a lightweight, fully functional Grid-framework which has been developed by the ALICE experiment – one of the four LHC experiments. The necessity to develop its own Grid software emerged because large scale simulations had to be done already at a very early stage of the experiment, mainly for the Technical Design Report. At this time, though, the EU's EDG project and its middleware were still in their early stages. ALICE thus had to operate in an area of conflict, where its computing requirements could not yet be matched by the agreed Grid solution of all four LHC experiments – the EDG-based middleware LCG.

In order to be able to make use of LCG or its components at a later time, AliEn has been built in a distinctively modular way. Even if the underlying architecture should change, ALICE users should not experience the necessity to adjust to a new environment. Therefore a constraint to the development of the AliEn environment has been that the interface to the system should remain the same.

Unlike most Globus-based frameworks, AliEn has been based on Web-Services from the very beginning, which facilitates the implementation of future standards such as OGSA[18] and WSRF[19]. Also, more than 95% of AliEn consists of standard Open Source components, thereby largely reducing the development and maintenance requirements.

All of these external components are used without modification. The individual modules are glued together by using the scripting language Perl, which offers easy-to-use database access and SOAP[20] support as well as a good Open Source implementation of cryptography modules. The AliEn architecture can be seen in figure 2.

[16]At the time of writing of this paper, LCG's middleware is available in version 2.2, often referred to as LCG-2

[17]In the Grid context, a Computing Element is the front end machine to a number of worker nodes, usually implemented as a cluster. It hands Grid jobs over to a local job scheduler, such as the Portable Batch Submission system ("PBS")

[18]**O**pen **G**rid **S**ervice **A**rchitecture

[19]**W**eb **S**ervice **R**essource **F**ramework

[20]Simple Object Access Protocol

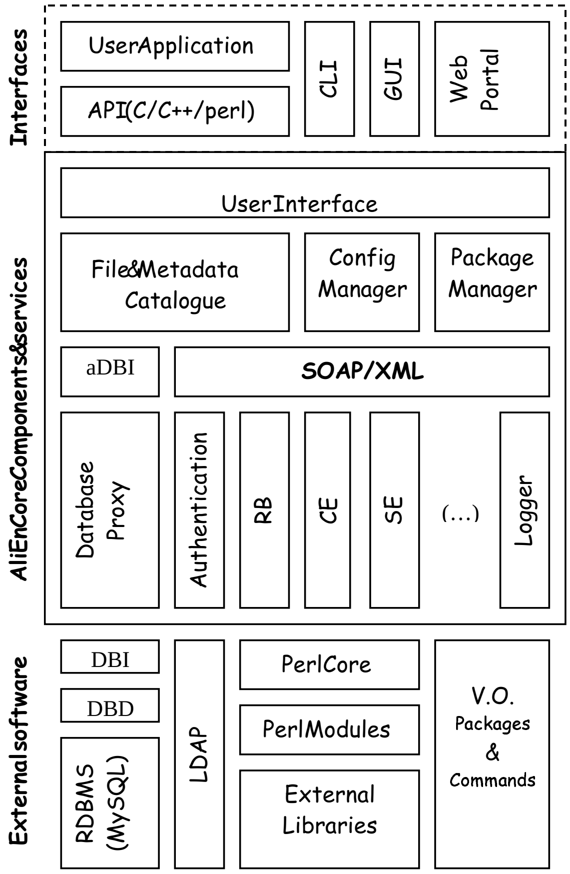

Figure 2: The AliEn architecture at a glance. The modules are organised according to their functionality.

A core piece of AliEn is the file catalogue, which acts as a globally distributed filesystem. Like in the case of the EDG Replication Services, it maps logical filenames (LFNs) to their physical pendants (PFNs). Via the user interface, in its appearance very similar to the bash shell and featuring most Linux filesystem commands, the AliEn user has worldwide access to all data registered in the file catalogue. Exclusive read/write permissions can be given to individual LFNs, just like in an ordinary UNIX filesystem. The technical implementation of the file catalogue is based on the relational database MySQL. The package manager, being another central component of the AliEn Grid environment, is able to manage the relevant software of a specific Virtual Organisation[21] and automatically installs software packages on any registered site, if and when needed. The AliEn Resource Broker (RB) uses a pull architecture (as described in chapter 3.1).

The relatively loose coupling between the various resources and the resource broker allows it to connect other Grid systems as AliEn Computing and Storage Elements to an existing AliEn system (see figure 3). Interfaces to the European Data Grid middleware and to LCG-2 have been implemented. In this way AliEn can act as a *Metagrid*.

AliEn is in production use. More than 400000 ALICE jobs ran under AliEn control since 2001.

Despite the technical benefits of the AliEn middleware, it is a specialised solution for the ALICE experiment. While it has been shown to be possible to use AliEn in other projects not even related to particle physics (such as the UK MammoGrid), porting AliEn to another environment has proven to be difficult[22].

While the relatively small number of contributors[23] allowed for a very flexible work style and fast development cylces, AliEn is today hardly supported anymore. Most developers of AliEn have today moved on to gLite as the joint successor of LCG-2 and AliEn.

3.3 NorduGrid and ARC

The ARC[24] middleware, developed by the NorduGrid collaboration of the universities of Oslo (Norway) and Lund (Sweden), has been used by ATLAS in its first and second data challenge.

In comparison to EDG's middleware, ARC uses fewer Globus-2 components (mostly from the Grid Security Infrastructure), but nevertheless implements a very similar functionality to EDG. Like AliEn, ARC tries to make use of standard Open Source solutions such as OpenLDAP, OpenSSL and SASL

At the time of writing of this paper, ARC was only available in a pre-1.0 version 0.4.4 . A first ARC release dates back to May 2002.

4 The EGEE project

With EDG having had its successful final review in March 2004, a major contributor to LCG went out of development. The new EU project EGEE (**E**nabling **G**rids for **E**-Science in **E**urope) has been established as a successor of EDG. Figure 4 shows a comparison of Grid projects related to EGEE.
Unlike its predecessor, the focus of EGEE is less on technical development and research, but on the provision of production services to Grid users. Only 16% of EGEE's 32 Million Euro budget – to be spent over the course of two years – are available for the development of a new middleware, *gLite*, testament to the fact that its main components, LCG-2 and AliEn, already represent close-to production quality solutions. 48% of EGEE's budget will be invested in Grid operations, including the running of a Grid Operations Center ("GOC") plus several Regional Operations Centers ("ROC"), as well as a global Grid user support infrastructure[25]. Another 28% of the budget are invested into "networking" activities[26], such as dissemination, outreach, and user training.

As part of the further development of EGEE and European Grid operations, also the user base is expected to broaden, with other user groups beyond particle physics, such as bioinformatics, astronomy and geophysics expected to join soon.

EGEE thus makes the important step from Grid research to Grid deployment.

Despite its strong focus on operational Grid issues, the development of the gLite middleware is a core part of EGEE. Combining several, formerly independent, middlewares into one product is an important step towards standardisation.

[21] A **V**irtual **O**rganisation (VO) is a means of controlling access to resouces. Users can only access computing and data resources belonging to their VO. This feature is nowadays used in most Grid implementations

[22] As an example, one might stumble across hardwired paths in the code when trying to port the middleware to a different particle physics experiment.

[23] The number of contributers to AliEn was below 10 at any given time

[24] Advanced Resource Connector

[25] handled by Forschungszentrum Karlsruhe/Germany together with Academia Sinica / Taiwan

[26] in the sense if "social networking", not in the sense of "building computer networks"

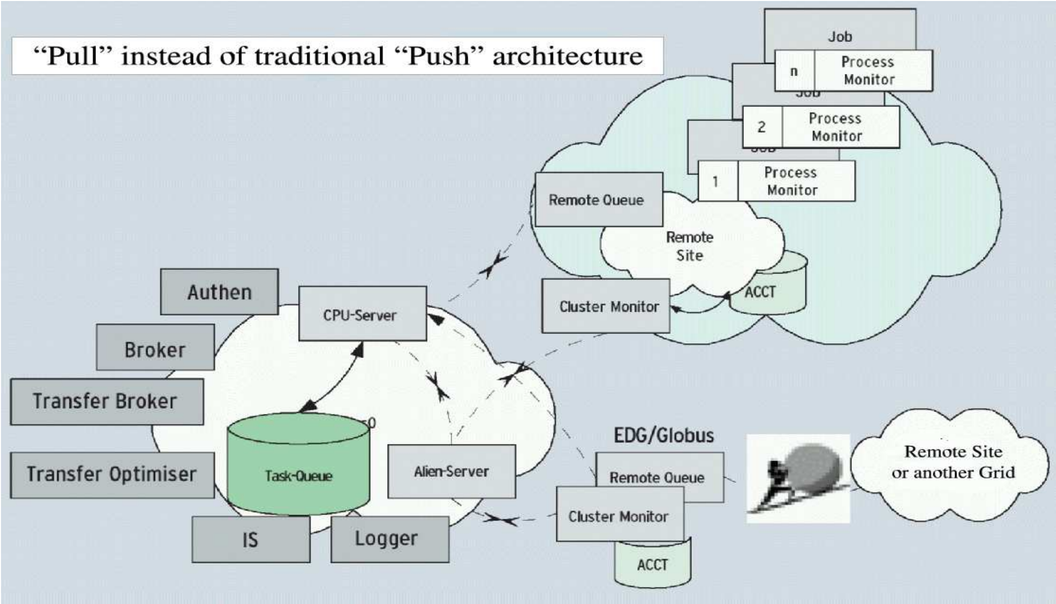

Figure 3: An AliEn server can connect to an AliEn CE, but also to a completely different Grid system. An AliEn server and client with the corresponding services can be operated together with other Grid systems.

Project	*EDG / DataGrid*	*LCG*	*ALICE experiment*	*EGEE*	*D-Grid*
Project Focus	middleware development and research	middleware deployment, development where needed	building a Grid middleware suitable for experiment's needs	middleware consolidation, deployment, training, support	middleware integration, application support, Grid deployment
Middleware	EDG	LCG-2	AliEn, LCG-2	LCG-2 (EGEE-0), gLite (EGEE-1)	various (gLite, LCG-2, Unicore, Cactus)
Geographical Scope	focus on Europe	world wide	world wide	focus on Europe	Germany
User communities	all interested parties, industry	mainly physicists working at LHC	members of ALICE collaboration	all interested parties, industry	all interested parties, industry
Remarks	Project finished with successful review in March 2004; 9.8 million Euro over a 3-year period. 30 million including national projects of 21 partners	ongoing project, LCG-2 middleware partially based on EDG middleware	small developer base, specialised target audience; web services based middleware	ongoing project, successor of EDG project; 32 Million Euros over 2 years; gLite partially based on LCG-2 and AliEn	ramping up; Expected funding: 300 Million Euros (equally split between research institutions, the German federal ministry of education and research, and industry)
Home	www.eu-datagrid.org	lcg.web.cern.ch/LCG	www.cern.ch/alice	www.eu-egee.org	www.d-grid.de

Figure 4: Comparison of Grid projects related to EGEE

4.1 The EGEE middleware gLite

The LCG middleware was so far predominantly used as a means of running large scale, distributed Monte Carlo simulations[27]. Now the focus starts to shift towards a more analysis-driven environment. While large-scale detector simulations can be managed by a relatively small number of people, event analysis is done by huge numbers of independently acting physicists. This paradigm shift implies a different means of interacting with the Grid middleware.

Strongly influenced by this, the ARDA[28] workgroup, consisting of members of all 4 LHC experiments as well as representatives from LCG and the Grid Application Group (GAG), gave recommendations for the future EGEE middleware, published in the ARDA architecture.

gLite started as a fork of AliEn, and many AliEn developers are now members of the gLite team. It must be stressed again, though, that the final gLite-version will be based on a mixture of LCG-2 and AliEn components, and also many components and ideas foreign to AliEn and LCG-2 find their way into gLite. As an example, gLite is influenced by the Global Grid Forum (GGF) and OGSA. Further ideas are contributed by NorduGrid's middleware ARC, as discussed in section 3.3. Thus, the next generation middleware of the EGEE project is truly a "best-of-breed" framework.

Like AliEn, the EGEE middleware gLite is based on Web services. It thus allows an easier compliance with upcoming standards based on the same principle, such as OGSA. Additionally, a web service-based middleware is easier to integrate into existing environments, due to its independence of operating systems and hardware architectures.

Due to its initial origin in the AliEn world, the current gLite prototype shares many of its features.

The gLite Computing Element is based on the AliEn CE and, as was described for AliEn, works with a pull-architecture. There is however now an additional choice to switch the CE to "push mode" like in EDG. Also the gLite user interface is mostly

[27] In High Energy Physics, the Monte Carlo method allows the detailed simulation of particle decays and detector responses

[28] **A** **R**oadmap to **D**istributed **A**nalysis

based on the AliEn shell with it's intuitive bash-like behaviour.

The Workload Management System (WMS) of gLite is a combination of the EDG WMS (including logging and bookkeeping), the AliEn TaskQueue and the new Information Supermarket (ISM). This way a decoupling between the collection of information concerning resources and its use can be achieved.

Also the gLite File & Replica Catalog is decomposed into the File Catalog Interface based on the AliEn file catalog (dealing with operations on the logical namespace) and the Replica Catalog Interface, which originated from the Globus/EDG RLS[29] and is responsible for replica operations based on a GUID[30]. The interface decomposition leads to service interfaces with well-defined semantics.

gLite security is based on GSI[31] security and grid-mapfiles, as well as VOMS[32] from the Authorization Working Group of EDG. This service is basically a simple account database supporting multiple VOs, to which the user may "log in".

Finally, the open source software license of gLite is based on the EDG license.

The current gLite Prototype is deployed already at a number of test sites.

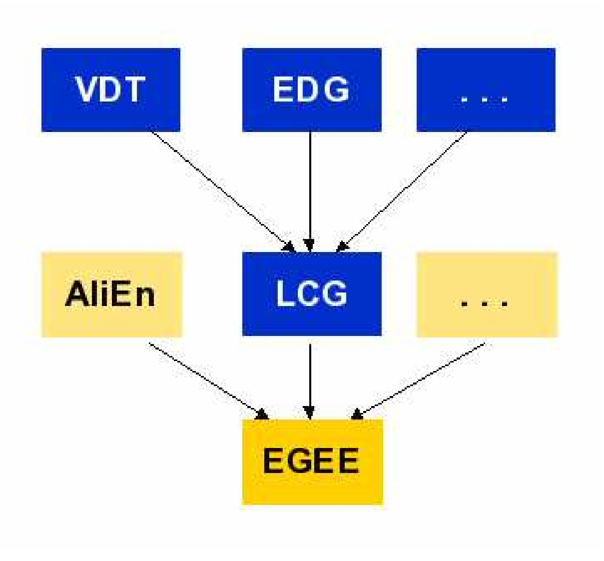

Figure 5: The new EGEE middleware gLite is built on the LCG and AliEn middlewares.

4.1.1 Technical Evolution

For some time gLite will coexist with the LCG-middleware. After the first inofficial pre-release of gLite, planned for December 2004, the EGEE middleware can be installed on the LCG project's pre-production testbed, in parallel to the existing LCG-2 installation.

Under the pre-condition of the approval of all LHC experiments, as soon as it will be clear that gLite is able to satisfy the computational needs of LHC experiments, gLite can take over as the main LCG middleware.

In the meantime, LCG-2 will concentrate on large-scale productions, whereas gLite will focus on distributed data analysis, albeit profiting as far as possible from the LCG-2 infrastructure.

It is unlikely that the migration to EGEE-1 (aka gLite) will happen before mid-2005.

[29] **R**eplica **L**ocation **S**ervice

[30] **G**lobal **U**nique **ID**entifier

[31] Grid Security InfraStructure – a Globus component

[32] VO Management Service

A smooth transition from the Globus-2 based LCG-2 middleware to the Web-Service based gLite might not always be possible. Therefore the availability of support-services as well as training is important to ease the migration.

It is important to note that, by combining several formerly independent Grid middlewares, one of EGEE's major achievements will be standardisation.

5 National Grid infrastructure – D-Grid

With Grid Computing quickly evolving into a real-world technology, capable of solving even the most demanding compute problems, it is mandatory for national Grid infrastructures to be in sync with the technologies used in other countries. Grid Computing is not a local phenomenon.

As an example, the 10th Global Grid Forum in Berlin (Germany) in March 2004 saw the inauguration of the German D-Grid initiative[33]. Its goal is to further enhance Germany's Grid infrastructure, based on networks, computer resources, modular middleware components and standard interfaces, ready-to-use Grid applications and scientific communities. As was announced on a press conference running alongside GGF10, the German federal ministry of education and research (BMBF) expects a funding for this project of 300 million Euros over 5 years, with one hundred million being contributed each by industry, research institutions and the BMBF.

D-Grid expects to be in full production in 2008, serving many different scientific communities ranging from climate research and high-energy physics over bio-informatics and medicine to life sciences.

Five work groups have been formed – "Cooperation and Operation", "Middleware and Services", "Management Methods and Autonomic Computing", "Networking" and "Data and Information Management" – jointly they provide the input for the D-Grid integration project (IP). The latter will be complemented by 3-5 Grid projects of the scientific communities. It is the purpose of the IP to build a "Meta Grid", making sure that the community Grid projects and their middlewares can interact with each other. Middleware packages being discussed in this context include Cactus, LCG-2, Unicore[34] and gLite.

D-Grid will also create an inventory of existing Grid middleware packages as part of the "Middleware and Services" work package, in order to re-use the best components.

A close cooperation with EGEE is likely, not the least because there is a large overlap between the German EGEE-federation and the D-Grid community[35]. Ties with other European Grid deployment projects such as the UK e-Science program are possible.

While D-Grid is a relatively new initiative, it can build on other projects and a quickly growing network- and computing-infrastructure. Given the large amount of funding the project is thus likely to be a success.

[33] D stands for "Deutschland" (Germany)

[34] Unicore is a software project funded by the German government with the aim to connect Germany's largest Supercomputer centers and to create a uniform access built on existing technologies. Unicore is written in Java for portability reasons and today provides the functionality of a fully grown Grid middleware, including a graphical user interface. A special feature of Unicore is its ability to take into account job dependencies and work flow mechanisms. Interfaces to Globus have been created in the EU-funded Grid interoperability project "GRIP".

[35] The reason for D-Grid to be featured in this paper is the author's expectation that EGEE and gLite will play an important role in D-Grid. Please note, though, that the process of selecting an appropriate technical platform for D-Grid is still underway. No final conclusion regarding the relationship of EGEE and D-Grid can thus be drawn.

6 Conclusion

In the time since Grid Computing became a research topic in the mid-nineties, many different, technically complex Grid middlewares have been developed. From the many different routes pursued in Grid research, a number of lessons can be learned:

- Standardisation is an important goal, but can – both for practical and logical reasons – not always be achieved. National and global Grid initiatives thus must put similar emphasis on interoperability of different Grid middlewares.
- Alongside sophisticated features a consistent and user-friendly behaviour of Grid components is important to end-users of Grid systems[36].
- Research collaborations must find the right balance between development flexibility and man power. A distributed, large work force can meet higher targets than a small number of developers, but cannot move as quickly. Likewise, a small number of developers will not be able to guaranty long-time maintenance and support for a Grid middleware.
- Support and Training[37] play a crucial role in generating a critical mass of users for a Grid software.

With only relatively few middlewares evolving as "best-of-breed" packages, projects like EGEE and D-Grid can focus on Grid deployment. Combining existing technologies into gLite furthermore presents an important step towards standardisation. D-Grid, in turn, puts additional emphasis on integration. Cooperation with other national Grid initiatives helps with the establishment of a Grid infrastructure far beyond the boundaries of Europe.

7 Acknowledgement

The authors would like to thank Forschungszentrum Kalrsruhe (FZK / Germany), Gesellschaft für Schwerionenphysik (GSI, Darmstadt / Germany), the German Federal Ministry of Education and Research (BMBF) and the members of the EU project EGEE ("Enabling Grids for e-Science in Europe") for their constant help and support.

References

I.Foster, J. Geisler, W. Nickless, W. Smith, S. Tuecke (1997) *Software Infrastructure for the I-WAY High Performance Distributed Computing Experiment* Proc. 5th IEEE Symposium on High Performance Distributed Computing, pp. 562-571, 1997

I.Foster, C. Kesselman (1997) *Globus: A Metacomputing Infrastructure Toolkit* Intl J. Supercomputer Applications, 11(2):115-128, 1997

Ian Foster, Carl Kesselmann (1998) *The Grid – Blueprint for a New Computing Infrastructure*, Morgan Kaufmann Publishers (1998)

D. Abramson, J. Giddy, L. Kotler (2000) *High performance parametric modeling with NIMROD/G.* International Parallel and Distributed Processing Symposium, Cancun, Mexico, IEEE Computer Society Press, Los Alamitos, CA, 2000, PP.520–528

R. Ostojic (2001) *Status and Challenges of the LHC Construction*, IEEE Particle Accelerator Conference PAC'01, Chicago, IL, USA, 18–22 Jun. 2001, pages e-proc. 16

Todd Tannenbaum, Derek Wright, Karen Miller, and Miron Livny (2002) *Condor - A Distributed Job Scheduler*, in Thomas Sterling, editor, Beowulf Cluster Computing with Linux, The MIT Press, 2002. ISBN: 0-262-69274-0

L.Guy et al. (2002) *Replica Management in Data Grids* GGF5, Edinburgh

P.Saiz, L.Aphecetche, P.Buncic, R.Piskac, J.E.Revsbech, V.Sego (2003) *AliEn – ALICE environment on the GRID*, Nuclear Instruments and Methods 2003, 437 - 440 alien.cern.ch

The ALICE Collaboration: F. Carminati et al. (2004) *ALICE: Physics Performance Report, Volume I*, J. Phys. G: Nucl. Part. Phys. 30 1517-1763, alice.web.cern.ch/Alice/AliceNew

S.R. Amendolia et al. (2004) *MammoGrid: a service oriented architecture based Medical Grid Application* Proceedings of the 3rd International Conference on Grid and Cooperative Computing, Wuhan, China 2004

Les Robertson (2004) *LCG Status Report - 3rd Quarter 2004*, CERN-LCG-PEB-2004-19 lcg.web.cern.ch/LCG

The ARDA Project lcg.web.cern.ch/LCG/peb/arda

The Virtual Data Toolkit www.cs.wisc.edu/vdt

International Virtual Data Grid Laboratory www.ivdgl.org

The gLite Middleware glite.web.cern.ch/glite

The NorduGrid Project www.nordugrid.org

The EGEE project www.eu-egee.org

The EGEE Middleware Architecture edms.cern.ch/document/476451

Die D-Grid-Initiative www.d-grid.de

The German Ministry of Education and Research www.bmbf.de

[36] This statement reflects comments from users gathered at Grid-training events, as organised by two of the authors.

[37] Meant in the sense of training events and support centers