EGI-InSPIRE

EGI Technology Roadmap (including the UMD Roadmap)

EU DELIVERABLE: 5.4

Document identifier:	EGI-D5.4-v16
Date:	**07/09/2011**
Activity:	**SA2**
Lead Partner:	**EGI.eu**
Document Status:	**FINAL**
Dissemination Level:	**PUBLIC**
Document Link:	https://documents.egi.eu/document/612

Abstract

The EGI Technology Roadmap, of which the Unified Middleware Roadmap describing the technical components coming from within the community is a subset, describes the long-term evolution of the EGI production infrastructure. This release of the EGI Technology Roadmap provides a classification of the functional capabilities needed within and to support the production infrastructure and the scope of their deployment. To understand the sustainability of the software components available for each capability, the current and projected dependency of these components is classified. Together, this analysis provides a basis for future discussions as to the sustainability options within EGI's current technology and its consumers. The short-term technical-level description of future UMD releases based on the currently available technology roadmaps provided from the external technology providers is also provided.

I. COPYRIGHT NOTICE

II. DELIVERY SLIP

	Name	Partner/Activity	Date
From	Steven Newhouse, Michel Drescher	EGI.eu	27/07/2011
Reviewed by	**Moderator:** Sy Holsinger **Reviewers:** Morris Reidel Robert Lovas	EGI.eu/NA2 FZJ/External MTA-SZTAKI	31/8/2011
Approved by	**AMB & PMB**		6/09/2011

III. DOCUMENT LOG

Issue	Date	Comment	Author/Partner
1	27 July 2011	First draft	Steven Newhouse, Michel Drescher, EGI.eu
2	29 August 2011	Revised following external review	Michel Drescher, EGI.eu

IV. APPLICATION AREA

This document is a formal deliverable for the European Commission, applicable to all members of the EGI-InSPIRE project, beneficiaries and Joint Research Unit members, as well as its collaborating projects.

V. DOCUMENT AMENDMENT PROCEDURE

Amendments, comments and suggestions should be sent to the authors. The procedures documented in the EGI-InSPIRE "Document Management Procedure" will be followed: https://wiki.egi.eu/wiki/Procedures

VI. TERMINOLOGY

A complete project glossary is provided at the following page: http://www.egi.eu/about/glossary/. A technology-orientated glossary is provided at https://wiki.egi.eu/wiki/EGI_Technology_Glossary.

VII. PROJECT SUMMARY

To support science and innovation, a lasting operational model for e-Science is needed − both for coordinating the infrastructure and for delivering integrated services that cross national borders.

The EGI-InSPIRE project will support the transition from a project-based system to a sustainable pan-European e-Infrastructure, by supporting 'grids' of high-performance computing (HPC) and high-throughput computing (HTC) resources. EGI-InSPIRE will also be ideally placed to integrate new Distributed Computing Infrastructures (DCIs) such as clouds, supercomputing networks and desktop grids, to benefit user communities within the European Research Area.

EGI-InSPIRE will collect user requirements and provide support for the current and potential new user communities, for example within the ESFRI projects. Additional support will also be given to the current heavy users of the infrastructure, such as high energy physics, computational chemistry and life sciences, as they move their critical services and tools from a centralised support model to one driven by their own individual communities.

The objectives of the project are:

1. The continued operation and expansion of today's production infrastructure by transitioning to a governance model and operational infrastructure that can be increasingly sustained outside of specific project funding.
2. The continued support of researchers within Europe and their international collaborators that are using the current production infrastructure.
3. The support for current heavy users of the infrastructure in earth science, astronomy and astrophysics, fusion, computational chemistry and materials science technology, life sciences and high energy physics as they move to sustainable support models for their own communities.
4. Interfaces that expand access to new user communities including new potential heavy users of the infrastructure from the ESFRI projects.
5. Mechanisms to integrate existing infrastructure providers in Europe and around the world into the production infrastructure, so as to provide transparent access to all authorised users.
6. Establish processes and procedures to allow the integration of new DCI technologies (e.g. clouds, volunteer desktop grids) and heterogeneous resources (e.g. HTC and HPC) into a seamless production infrastructure as they mature and demonstrate value to the EGI community.

The EGI community is a federation of independent national and community resource providers, whose resources support specific research communities and international collaborators both within Europe and worldwide. EGI.eu, coordinator of EGI-InSPIRE, brings together partner institutions established within the community to provide a set of essential human and technical services that enable secure integrated access to distributed resources on behalf of the community.

The production infrastructure supports Virtual Research Communities (VRCs) − structured international user communities − that are grouped into specific research domains. VRCs are formally represented within EGI at both a technical and strategic level.

VIII. EXECUTIVE SUMMARY

The EGI Technology Roadmap describes the long-term evolution of the EGI production infrastructure. Locating EGI within the larger EU vision for e-Infrastructures in 2020, this document describes the underlying principles that drive the evolution of the EGI production infrastructure.

Based on this ground-setting framework, the EGI Technology Roadmap describes the architecture of the federated production infrastructure by examining the available service portfolio (which goes beyond *software services*) with respect to the targeted operational deployment. In order to analyse long-term sustainability options for existing solutions in the EGI community, the Technology Roadmap defines three assessment criteria: Functional classification, Deployment classification, and Maintenance classification.

The functional classification of the deployed software in EGI Capabilities describes effectively a service-oriented architecture, and allows any technology provider to establish a relationship with EGI and to provide components that meet a particular capability. Further classifying the EGI Capabilities into four different deployment scenarios helps identifying the sourcing of operational efforts in deploying and managing the software satisfying the respective EGI Capability. Finally, a classification by maintenance levels determines from where to acquire the necessary efforts to develop new software and maintain existing software (i.e. fixing bugs, and security vulnerabilities).

Having put the three assessment criteria in place, the Technology Roadmap analyses the available software and their sustainability options for each EGI Capability, grouped by functional area in a two-dimensional “radar” display indicating the current deployment and maintenance situation, and the possible transition to future sustainability options through arrows pointing into that direction. Contributions of existing Technology Providers to the UMD Roadmap are synthesised into overviews of planned activities in the medium and long-term future, followed by a short-term outlook on the UMD Roadmap, and concluded with the assessment of available implementations.

The overall document concludes with summarising the analysis provided in detail in the earlier sections of the document, stressing the fact that core infrastructure needs must be predominantly satisfied using generic software with minimal customisation, in order to sustain its long-term operation.

Finally, the appendix provides the details for each EGI Capability as referenced in the main part of the document.

TABLE OF CONTENTS

TABLE OF TABLES

TABLE OF FIGURES

1 INTRODUCTION

The European Union's (EU's) vision for Europe in 2020 is that of becoming a smart, sustainable and inclusive economy. Of the seven flagship initiatives established to implement this vision, two of which are of particular relevance to EGI in establishing smart growth across Europe – the Digital Agenda for Europe (DAE) [R 1] and the Innovation Union (IU) [R 2]. Investment in e-Infrastructures represents the leading edge of the DAE in the European Research Area (ERA), while having an integrated e-Infrastructure across the ERA that underpins the innovation expected within the IU.

Therefore the importance of e-Infrastructures in Europe 2020 [R 3] (the funding initiative that will follow the current 7th Framework Programme) by providing an integrated approach to supporting innovation across the whole of the ERA by providing standardised interfaces to interoperable services that eliminate the barriers to the free movement of knowledge across Europe cannot be ignored. The European Grid Infrastructure (EGI), through its network of national and domain specific resource providers, is therefore ideally placed in providing integrated and federated access to compute, storage and other resources needed to support innovation within the EU. EGI supports the IU by bring the ERA on-line, and is able to address key issues within the DAE that are relevant to the research community such as fragmented services, standards and interoperation that reduce the barriers to the movement of knowledge across Europe.

The EGI Technology Roadmap describes the high-level approach to technology adoption and deployment being used within EGI to ensure its alignment with the EC's vision for e-Infrastructures in Europe in 2020. The detailed technical implementations that will need to be developed to meet the needs of its end-user and operations communities is described in the UMD Roadmap (section 4.3) and has a shorter-term horizon of 2-3 years.

Separating out these two aspects is a critical first step in exposing some of the sustainability issues facing the provision of a pan-European infrastructure that needs to support diverse user communities each with different software environments that need to be deployed and operated at a European wide scale. For instance, who is responsible for maintaining the operational infrastructure as opposed to the services that run in the operational infrastructure? How much of the current technology used in production is sourced from outside the EGI (where other communities can contribute to its support) as opposed to be solely used within EGI (which the EGI community has to support)?

This first version of the EGI Technology Roadmap builds on previous versions of the UMD Roadmap [R 4 and R 5] by identifying the capabilities and the different consumers of the capability (resource providers, virtual research communities and end-users) that exist within the current and planned production infrastructure. The products that deliver each of these capabilities are assessed to determine their main sources of support – generic components coming from outside the EGI community, custom components that are developed by the EGI community, components contributed by individual user communities, etc. This analysis is used to assess the sustainability of the capabilities currently planned within EGI.

2 PURPOSE

The purpose behind the EGI Technology Roadmap is to establish a multi-year view as to how EGI will evolve from a technology perspective from the previous project-based structures to a sustainable pan-European e-Infrastructure. Sustainability is a critical aspect of this vision, which is dealt with in EGI's Sustainability Plan [R 6]; however, at the heart of any sustainability strategy there are three key points relevant for the Technology Roadmap:

- Clearly defining services that are attractive, unique and needed by their consumers
- Sourcing these services from the most effective technology solutions available
- Delivering the defined services to a high-quality to the available resources

Defining the services that 'add value' to the consumers and providers of the e-Infrastructure within EGI is critical, and these services will inevitably change over time. The EGI Design Study [R 7] project defined a number of EGI Global Services that have now been instantiated within the EGI-InSPIRE project and form a basis for future service evolution.

The main focus during the first year of the EGI-InSPIRE project has been establishing the EGI with EGI.eu as its coordinating hub and transitioning to an operations infrastructure based primarily on national resource providers. While the consolidation of this and other activities continues, it is appropriate to consider challenges facing the EGI community during the remainder of the EGI-InSPIRE project. The sustainability discussions need to be focused around where, as a community, we want need to focus available resources given the legacy that we are starting with:

- Transitioning to a virtualised infrastructure: Seen as critical in providing an infrastructure that can be more flexible and responsive to users' needs and expand the user base.
- Integrating and provisioning virtualised resources on a commercial basis: Establishing the necessary operational procedures (functional, monitoring, accounting, etc.) for users to use commercially provided resources (either from within or external to the EGI community).
- Structuring EGI's technical activities into areas where further innovative development of software and deployment models are still needed (suitable for EC funding) and the routine operational activities that are then sustained through the EGI community.

The focus of this initial version of the EGI Technology Roadmap is to define a technical architecture that supports and helps structure the on-going sustainability discussions within the EGI community and to indicate how the current capabilities within the roadmap will evolve.

3 ARCHITECTURE

3.1 Principles

EGI's principal activity is the coordination of a federated European wide distributed computing infrastructure to support high-throughput data analysis tasks for multi-national research communities. Its main tasks supporting this goal are to:

- Ensure the secure integration of the distributed resources in the infrastructure through the provision of services to the provider community that allow the resources to be coordinated, managed, monitored and accounted for.
- Provide for a federated infrastructure that is open in the sense of different resource types (compute, storage, data, virtualised, etc.) and open in the sense of different providers (commercial, research, academic, etc.) that are integrated through open standards or specifications where they are available.
- Coordinate the delivery of the services (technical, infrastructure, human) necessary to integrate the distributed resources and to ensure their effective operation either through open-source software within the EGI community, outside of the EGI community or from commercial providers.
- Provide services that will help internationally distributed research communities to discover and exploit the infrastructure for the effective use to support high throughput data analysis.
- Provide the means for user communities to deploy the services they need on the resources they need.

3.2 Service Portfolio

EGI.eu coordinates the provision of a portfolio of services for national resource providers in order to federate their resources into a secure integrated environment for the use by international Virtual Research Communities (VRCs)_within the ERA and their international collaborators. These services can be grouped into three areas relating to the consumers:

- **Resource Providers**: Services that support the secure operation of an integrated federated infrastructure composed of resources from individual national or domain specific resource infrastructure providers. They provide the *management* infrastructure necessary to administer and maintain the *operational* infrastructure made available for end-users[1]. These services are logically centralised even if they are operated by another organisation for EGI.eu on behalf of the EGI community.
- **Virtual Research Communities**: Services that need to be deployed within the production infrastructure to expose the resources used within their community for transnational access. EGI.eu facilities the installation of these services (through the verification and staged rollout processes) that are then operated by the local resource providers through their distribution in the EGI Software Repository. Many of these services are released within the UMD for local site deployment to provide particular functional capabilities. This model is expected to evolve for these functional capabilities to be provided using the EGI Virtualisation capabilities (see appendix A) by experts within the VRC.
- **End-Users**: Services that help end-users within a community to make the best use of the pan-European infrastructure available to them. The key feature of the EGI User Support platform is to enable the integration and branding of these logically centrally provided technical services into the environments exposed by NGIs and VRCs to their user communities.

[1] Those services are typically transparent and not directly accessible by end-users, though the outcome may be public information, for example availability and reliability data for each resource centre.

In addition, services are provided to the whole EGI Community to support its coordination and continued development. Many of these services are based around software (either operated centrally by a single site or packaged and distributed for operation at many resource centres), which needs to be developed and maintained, in addition to any costs of operation. The operational deployment models vary depending on the consuming audience but can be *broadly* grouped as:

	Deployment Scope			
Consumer	Central	National	Community	Resource Centre
Resource Provider	X	X		
Virtual Research Community			X	X
End-user	X		X	

Table 1: Service consumer and deployment scope

Services that need to be distributed and deployed many times in different environments require significantly more engineering effort than software that is deployed and used in one location. In addition, deploying and operating software at multiple resource centres within the community frequently costs the community more effort in total than the cost of a centrally provided service to which everyone contributes.

The EGI Community has a number of services to facilitate the distribution of software. These range from the wikis and mailing lists to help in the coordination of software development, to the EGI Software Repository that provides a central point for binary software components to be made available for automated installation. The repository can also facilitate the workflow around software verification needed to validate deployment as part of the production infrastructure, or potentially more lightweight as part of a community contribution.

Recognising the different service consumers, and the types of services they are consuming helps inform the sustainability of the software based upon which the EGI community depends.

3.3 Software Sustainability

Over the last decade, the establishment of a production infrastructure for HTC and HPC data analysis required significant bespoke software development to meet the particular needs of the community. This activity tracked, and in many ways was enabled by, the growth of the open source software community around the world. However, that explosive growth in functional demand from within our own community and software development activity in the open source community have rarely been reconciled or analysed.

EGI benefits from open-source software activities such as Linux as a common operating system, security libraries such as OpenSSL, and web service engines such as Apache for some of its software components. Although code from these projects has been used within the community, the software frequently had to be modified to meet our needs rather than simply being reused through stable public interfaces. Considerable effort has been invested by the EMI project during its first year of activity to re-engineer many of the software components used within EGI to eliminate some of these dependencies and reuse components directly off the shelf through established code repositories from outside the EGI community.

In considering the long-term sustainability of the software activities within EGI there are two dimensions to be considered beyond the functional classification of the capability: deployment need and maintenance cost.

3.3.1 Functional Classification

For many of the technology components needed within EGI to satisfy a defined community capability, EGI expects there to be multiple providers. Indeed, having multiple technology providers able to satisfy a particular capability helps contribute to EGI's sustainability by removing its dependence and lock-in with a single supplier. Defining these capabilities by functional (e.g. open standards or specifications) and non-functional (e.g. performance, availability, reliability) requirements allows any technology provider, both commercial or research based, to establish a relationship with EGI and to provide components that meet a particular capability.

The functional classifications of the current capabilities are defined as:

- **Security:** Describe the needs of secure identity and access management in EGI.
- **Information:** Define the language and infrastructure necessary to describe, disseminate and publicise information about EGI resources.
- **Operations:** Capabilities necessary for efficient federated management of EGI.
- **Storage:** Define the needs to configure, manage and use large amounts of storage space in EGI for otherwise unstructured data.
- **Data:** Describe the requirements on remote and federated access on structured data, such as databases, or metadata on (structured or unstructured) data.
- **Compute:** Describe the needs around remote job execution and submission, job management and higher-level job related services such as scheduling and workflow.
- **Virtualisation:** Describe the functionality required for remote execution of virtual machines (VMs), from VM image format to remote management and image distribution.
- **Instrumentation:** Manage and control a scientific instrument (such as a radio telescope) using Grid technology.
- **Clients:** Define the requirements for client-orientated access to Grid services from an end-user perspective, ranging from high-level API definitions on command line tools suitable for manual invocation or scripting, and including graphical clients.

3.3.2 Deployment Classification

Implementations of the EGI Capabilities are deployed in several different scenarios and target the different use cases of the provided software. The following classification describes the predominant deployment scopes in which EGI Capabilities may be deployed.

- **Core Capability**: This is a capability critical to EGI in securely federating, accounting for and monitoring the resources that constitute the production infrastructure and is managed and deployed by EGI.
- **Resource Capability**: This is a capability that is deployed by EGI to expose resources in the production infrastructure for use by user communities.
- **Community Capability**: This is a capability that is critical to a particular user community and may be dependent on the Core or Deployed Capabilities provided by EGI but is not deployed or managed by EGI.
- **User Capability**: This is a capability that is needed by an individual user (e.g. client tools) and may be dependent on other deployed capabilities.

3.3.3 Maintenance Classification

Currently, EGI is heavily dependent on its own community to provide many of these capabilities as open-source software components. These components, alongside others, are verified within the production environment and released within the Unified Middleware Distribution (UMD). This reliance on the EGI community for these components inhibits the EGI community's move towards sustainability as too much of the technology needed by EGI is only supported by EGI. As the support that can be provided by the EGI community is limited – either provided directly by in-kind effort or national or European project funding – the amount of software that needs this support or the effort needed to support the software needs to be reduced. While software re-engineering and consolidation activities can reduce the required support effort, transitioning components to support models that include other communities could significantly spread the support burden. The support base for particular components can therefore be classified as:

- **Generic software**[2]: Software components that fall into this category are often referred to as "Plain Off The Shelf" (POTS) software; it is supported by communities outside of EGI and is re-used 'as is' within EGI's production infrastructure through custom configurations. Well-known examples are RDBMS (e.g. Oracle databases), Web Servers (e.g. Apache httpd), Web Containers (e.g. Apache Tomcat) or scripting languages (Python, PERL, etc.). Software components in this category are seen as being sustainable and should be maximised.
- **Custom Infrastructure**: The software component originates from within EGI and is driven by requirements needed to support EGI's production infrastructure. It may use generic components as its base or contribute material back into the generic infrastructure components; however, the maintenance effort within EGI available to support these components is limited.
- **User Community**: The requirements for such a software component and the support for such a software component originate in a user community. Sustainability is the responsibility of the community.
- **Other**: Components in this category do not fall clearly into any of the other categories and their number should be driven to zero.

The UMD represents primarily the set of custom infrastructure components needed within EGI to establish the production infrastructure. These components have been developed to meet specific use cases coming from the EGI community and have therefore been through additional verification of their functionality and integration into the production environment beyond any testing activities done by their developers. These custom infrastructure components may depend on other generic infrastructure components (either distributed as part of UMD or available through on-line repositories) or on user community components (distributed as part of UMD for convenience).

Figure 1 below recognises the long-term strategy to rely increasingly on generic components as the basis of EGI's deployed infrastructure. Currently, technology providers such as EMI and IGE are supported by the EC to maintain and develop components that have bespoke functionality infrastructure providers (e.g. EGI) and particular user communities. The funded support that the EGI Community will be able to provide on behalf of the resource providers is limited – indicated by the custom software band in the diagram. Much of the technology deployed in EGI provides access to site resources for the use of particular user communities. Support of these technology components should fall to the user communities that use them. It is recognised that there may be some technology components that are truly generic across all user communities and these could be maintained centrally if resources permit. There will inevitably be 'other' components that do not fall into any of the three

[2] Earlier versions of the EGI Technology Roadmap and the UMD Roadmap (D5.1, D5.2) referred to this category as "EGI Technology components", implying that only the EGI operational infrastructure should use generic software.

other categories following this analysis. Understanding the criticality of these components to the different stakeholders is one of the motivations for this analysis.

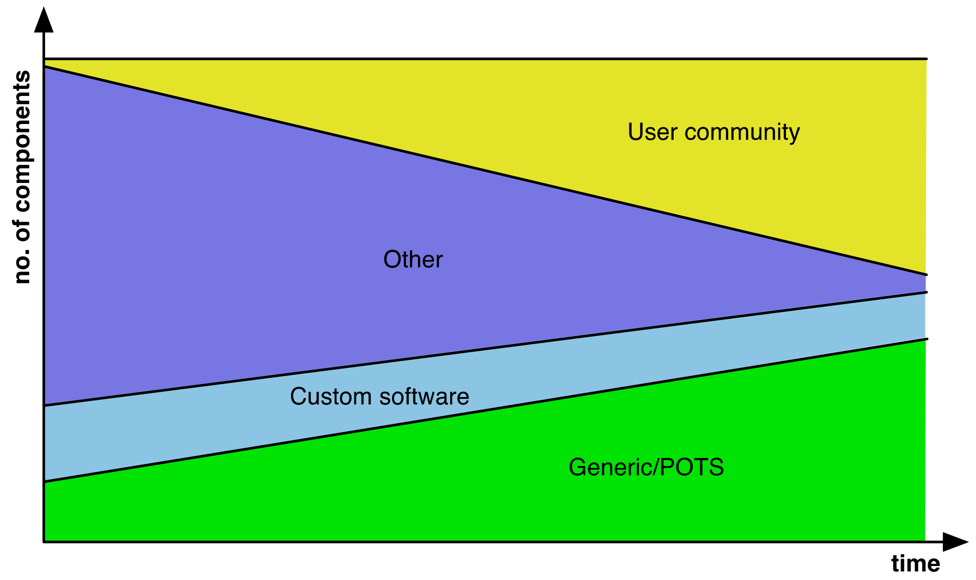


Figure 1: A possible projection of software maintenance evolution

3.3.4 Analysis

In considering the two aspects of functional capability and software maintenance, there are a number of conclusions that can be made:

- It is clearly desirable that as much of the software used within the EGI community should be generic so the other users of the software outside the EGI community can share the support burden.
- The more specific a given functionality is to a single community the more appropriate it is that that community undertake the development and support. Unsustainable tensions frequently arise when a community just consuming software that it does not contribute to demands functionality from the providing community.
- The software development resources that the EGI community can invest in customising and maintaining existing software solutions are small.

The table below indicates where it is most (+) or least (-) desirable for maintenance responsibility of a particular component to lie. The amount of the respective sign indicates the practicality of this being realised (no sign indicates a neutral stance).

	Maintenance			
Deployment Scope	Generic	Custom	User	Other
Core	+++	++	--	----
Resource	++		+++	----
Community	+		+++	--
User			+++	-

Table 2: A feasibility matrix software in different deployment scope

4 IMPLEMENTATIONS

4.1 EGI Capabilities

An EGI Capability represents the synthesis of requirements from different user communities that are grouped into a functional interface. A specific functional capability is grouped by functional area (Security, Compute, etc.) and by their deployment scope within the production infrastructure. Therefore, capabilities are no longer described in terms of software middleware (i.e. bound to a specific layer in the software architecture of the infrastructure as a whole), but in terms of a functional need across the whole architecture. This allows easier assessment of each capability in terms of its importance and deployment scope within the community, but also to explore the sustainability options available to implementations of a particular capability.

The tables presented in the following sections provide a “radar”-like assessment of EGI Capabilities, indicating perceived trends both in sustainability and deployment options for software implementing the respective capability, or capabilities. Each capability is individually represented by a small box containing an abbreviated name (the full name and description is given in Appendix A: EGI Capabilities). Arrows attached to the boxes indicate the expected shift in each capability that is needed to improve its sustainability by its main consuming community, and the projected deployment scenario in the future. Dotted arrows indicate potential shifting into the given direction.

Clearly, some functional capabilities can be classified as being needed in more than one deployment area; for example an Authentication and Authorisation Infrastructure (AAI) Capability is needed to manage access to the infrastructure and to control access to the services running in the infrastructure for a particular community). However, that does not necessarily mean that the same technology will be deployed to meet the needs of this capability for each consumer of the capability.

4.1.1 Security

The available Authentication (AuthN) solutions overwhelmingly reuse existing generic software and specifications, such as X.509 certificates, PKI infrastructures, etc. When considering alternatives for the support of new user communities, existing generic solutions should be considered to maintain the already achieved sustainability in this area. Available implementations and specifications are Shibboleth [R 8] and OpenID [R 9], which both use SAML 2.0 [R 10] as their language to convey authentication tokens.

VOMS (including VOMS-Admin) is predominantly used to deliver the Attribute Authority (Attr.A) capability. On the verge of being integrated into major Linux distributions, VOMS issues RFC 3281 [R 11] compliant attribute certificates, or SAML tokens as requested and required.

Authorisation (AuthZ) is delivered through many different bespoke solutions, often tightly integrated into existing high(er) level services, such as storage access (DPM). A scalable solution is provided by EMI with the Argus Authorisation framework, which allows a distributed, scalable and highly flexible deployment of EGI-wide authorisation using publicly defined standards (SAML and XACML [R 12]). EGI expects Argus as a prime candidate for release into the public domain as generic software to be re-used in a wider community.

EGI deploys MyProxy as Credential Management (Cred. Mgmt.) solution for research communities to store RFC defined proxy certificates. It is expected that existing and new user communities will move towards adopting authorisation solutions already established in their community once the integration of these approaches into the production infrastructure is established. It is therefore expected that community-specific authentication solutions will eventually be deployed and maintained by the user communities themselves.

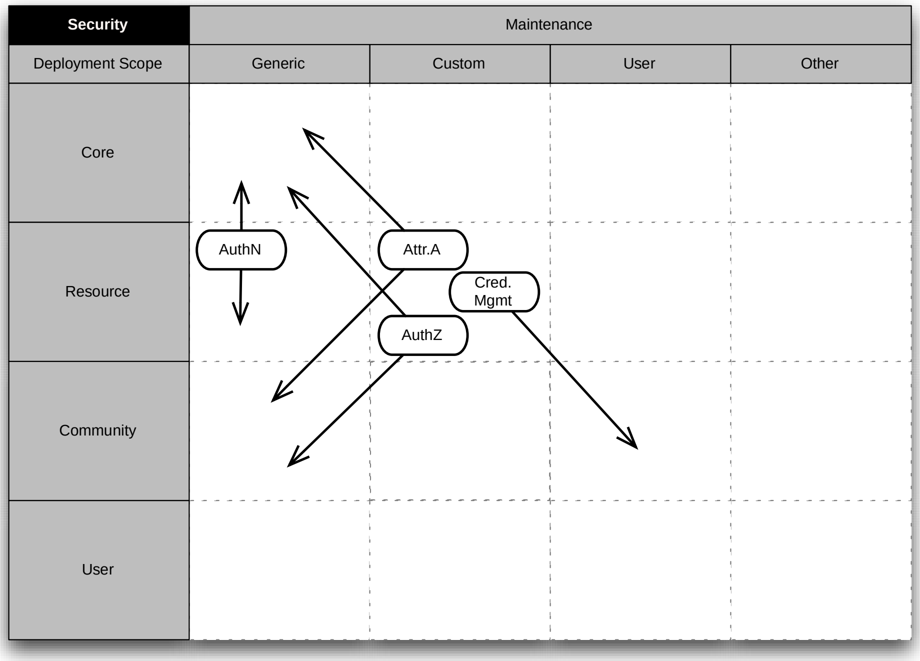


Figure 2: EGI Security Capabilities radar

4.1.2 Information

The Information Model (Model) is defined as the sole usage of GLUE, which exists in two different iterations as GLUE Schema 1.3 [R 13], and GLUE Specification 2.0 (GLUE2) [R 14]. EGI clearly sees a strong need to converge towards the use of GLUE2 throughout all components of the production infrastructure. Extensions to GLUE2 may be deployed to satisfy a requirement that have been defined in an open process engaging all stakeholders and compliant with the original specification. If applicable, rendering and parsing libraries for the GLUE2 schema available in different programming languages such as C, C++ and Java should be sourced from the pool of generic software.

The publication and search facilities for Information Discovery (Discovery) should remain stable in its current usage scenario. A clear need exists to maintain Information Discovery at the core of EGI as well as at the Resource level. A tightly coupled evolution of the Information Model and the Information Discovery is expected. The solutions used at EGI's core should evolve in a more conservative manner than the solutions provided for community-wide consumption at the resource level.

EGI uses Messaging (Messaging) as the core infrastructure for dissemination of information, such as accounting and monitoring information. Based on ActiveMQ [R 15] the maintenance and administration of the messaging infrastructure benefits from the use of a generic software solution from outside the EGI community. ActiveMQ makes use of the JMS 1.1 [R 16] specification – a specification developed as a Java Community Process, and as such all but an industry standard. Sustainability of the messaging infrastructure should aspire to broaden from mere product-based sustainability to product and interface(s)-based sustainability, allowing a broader pool of available software to provide and access to the messaging infrastructure beyond the operations community. With messaging attracting more interest from the EGI communities, this capability is likely to be deployed by user communities in the foreseeable future for next-generation distributed scientific research. At first, seen as an insular community-specific solution, the EGI Core messaging might in the future be provided as a global service to user communities to use within their Resource level.

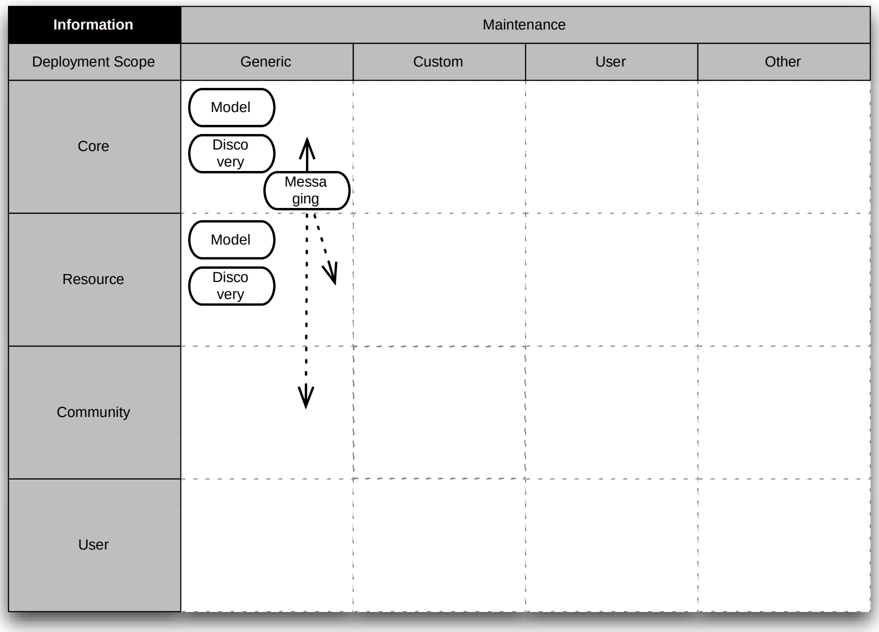


Figure 3: EGI Information capabilities radar

4.1.3 Operations

Accounting (Acct.) is provided as an integration of an Oracle database with custom components for the visualisation and reporting of the accounted data. However, accounting as such is a well-known, and well-understood business topic, that generic solutions should be preferred over any custom-made toolkit. More specifically, the evolution of the accounting infrastructure should be based on technically generic software (as opposed to generic in the sense of maintenance) that allows for very flexible accounting data declaration and aggregation rules to configure how the data that is fed in (via the messaging infrastructure) will be further processed using that generic software.

Monitoring (Monit.) the reliability and availability of the infrastructure are key operational tasks. Based on the widely available Nagios framework [R 17] (available in a free edition, and a commercial, support-enabled version) the development of custom probes to collect the availability and reliability of the deployed Grid services has been handed over to the Technology Providers. However, a significant part of the EGI monitoring infrastructure is provided as a custom reporting and monitoring system (SAM), which might be subject to reassessment for replacement by generic software components.

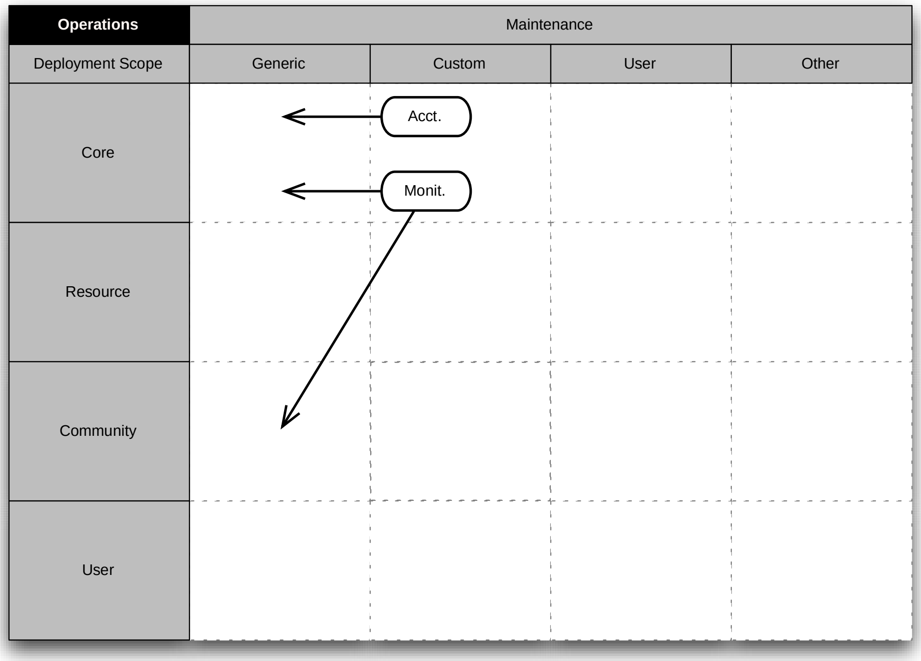


Figure 4: EGI Operations capabilities radar

4.1.4 Storage

Managing and handling Petabytes of storage (Storage Mgmt.) is a key operational feature for EGI as such large amounts of storage space cannot be provisioned on demand. The deployed software that exposes this storage resource has been developed by the EGI community for the EGI community. Its specialisation may hinder movement to a wider support base.

Accessing files (File Access) and transferring them (File Transfer) to other, alternative storage locations (or into the execution environment to be processed) are supported by custom solutions especially written for the Grid community. Protocols for the access and transfer of otherwise unstructured data are often combined to hide the complexities of dealing with remotely stored data. Such a capability is fundamental to the needs of many different user communities therefore there are few reasons not to provide a solution for the use by all of EGI's communities. However, the sustainability of the provided solution should aim for re-using or completely transitioning to generic software.

Scheduling file transfers (F. T. Sched.) is a capability that is often closely linked with workflows or to optimise the use of networking capacity. Such custom software is only deployed for some user communities and is considered to remain in the scope of the user communities.

The issue of data confidentiality (Encr./Decr.) is not a frequent requirement across EGI's user communities. As such, the responsibility of the maintenance of the used software should be kept with the user communities that are in need of data confidentiality.

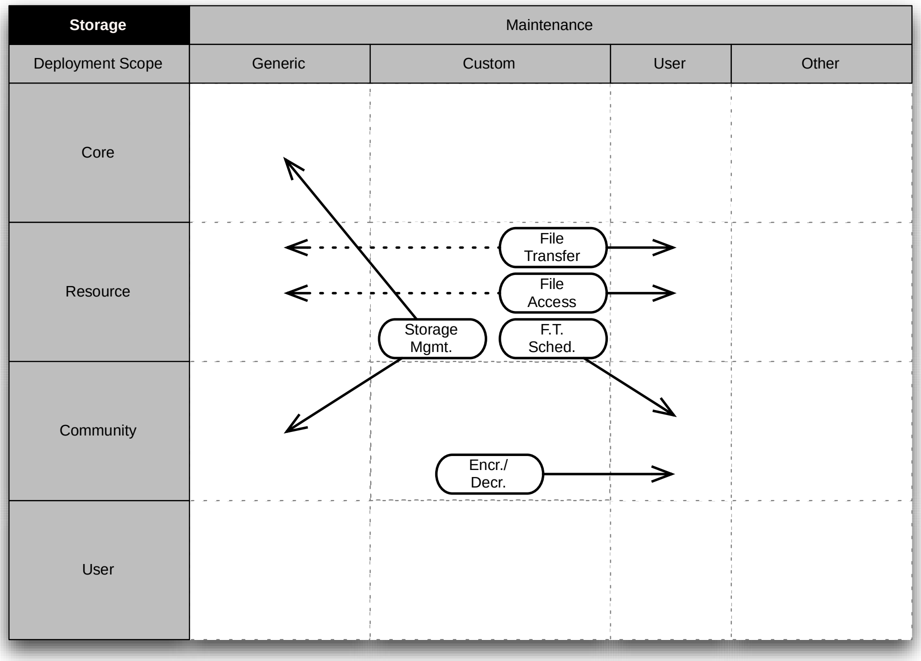


Figure 5: EGI Storage capabilities radar

4.1.5 Data

Access to structured data (Data Access), for example in form of relational data structures, is clearly a community specific need. Deployment and evolution of remote data access services should be maintained within the user communities that make use of such services.

Metadata catalogues (Metadata Catalogue) provide a mapping of human-readable names of files or data resources to system-level handles of the same resource (e.g. in the form of a UUID). Although different communities will have different metadata schemas, the basic functionality may be shared. Any heterogeneity of requirements and use cases for metadata management pushes the maintenance responsibility to the respective user communities for deployment within the community or potentially at the level of an individual user.

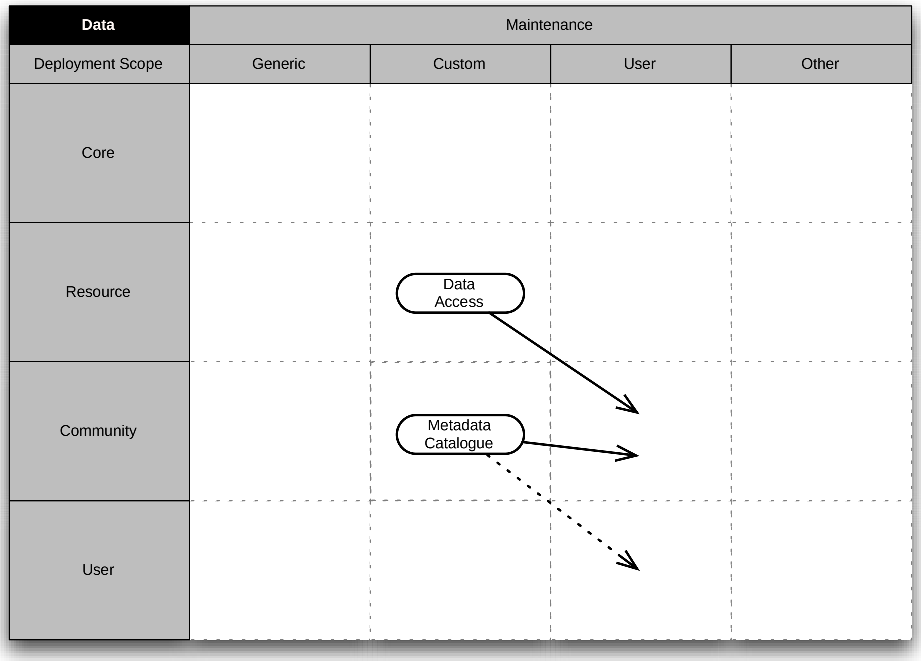


Figure 6: EGI Data capabilities radar

4.1.6 Compute

Job Execution (Job Exec.) is one of the most well-understood and used capabilities in the EGI production infrastructure. However, different models on how to provide for the Job Execution capabilities exist, though most of them are composed of a remote accessible interface to a local resource management system (such as SGE), aspiring to provide a uniform job execution layer across heterogeneous environments. Associated with the execution of jobs is the need to have jobs execute in parallel. Given the number of alternative MPI implementations (MPI1, MPI2, OpenMP) the maintenance of such a specialised parallel computing paradigm (Parall. Job) would be expected to find its sustainability in the user communities that make use of that capability.

Workflows (Workfl.) and Interactive Job Management (Inter. Job Mgmt.) are typical user community-related capabilities in the job management area. Many attempts have been made to converge on workflows, whether on the level of the description language or on the access interface (e.g. several attempts undertaken in OGF), so it is expected that workflows will remain a domain-specific capability in terms of software implementation. Interactive job management is until now not of cross-community interest, so there seems to be very little benefit in provisioning this capability on a resource level.

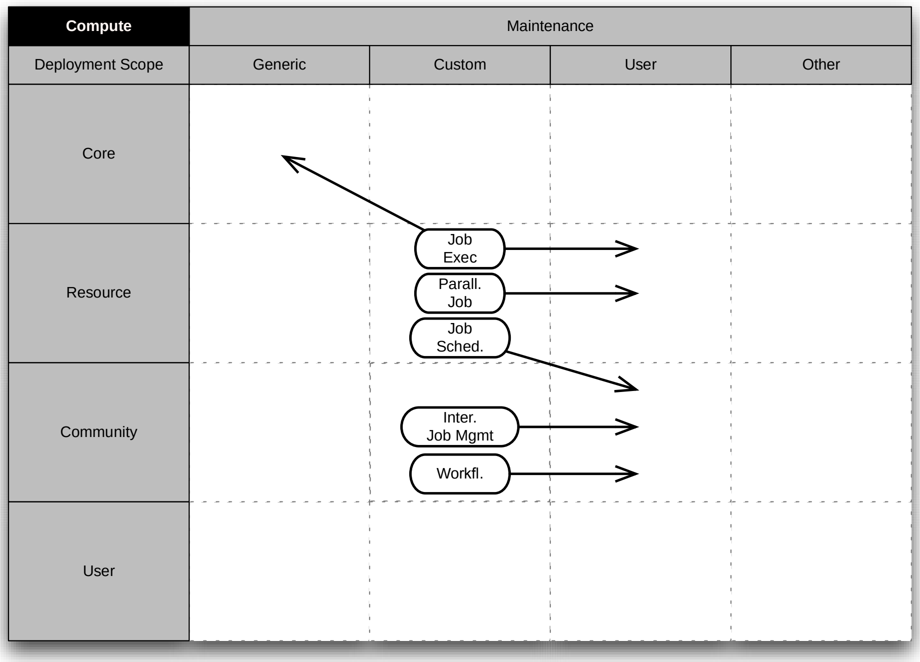
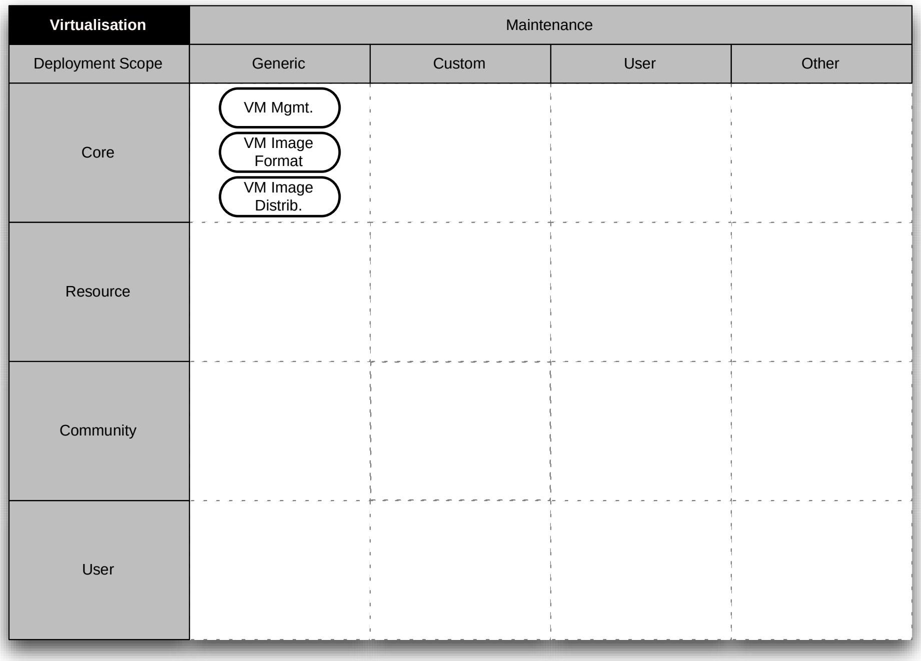


Figure 7: EGI Compute capabilities radar

4.1.7 Virtualisation

Virtualisation Capabilities were included in the previous version of the roadmap for the first time. Virtualisation Capabilities are becoming critical for EGI's sustainability as they allow for extremely flexible support of various different user communities. Reducing the necessity for each resource administrator to develop expertise in every application, or service that is deployed in the production infrastructure, is essential in order to support more diverse user communities. Managing the utilisation of the concrete resources (often referred to as the "bare metal") irrespective of the actual software (potentially domain-specific) contained within the Virtual Machines (VM) then becomes possible on a uniform level through generic software available for Virtual Machine Management (VM Mgmt.) in particular, and general data centre management. While aspiring to use general-purpose software for VM Management, EGI must not introduce custom-made or incompatible VM Image formats (VM Image Format) or infrastructure for VM Image Distribution (VM Image Distrib.)



Virtualisation	Maintenance			
Deployment Scope	Generic	Custom	User	Other
Core	VM Mgmt. VM Image Format VM Image Distrib.			
Resource				
Community				
User				

Figure 8: EGI Virtualisation capabilities radar

4.1.8 Instrumentation & Clients

Currently, no known Remote Instrumentation (Remote Instr.) capability is deployed in the production infrastructure, although its importance to very specialised user communities is acknowledged. The sustainability of EGI is not linked to the support of remote instrumentation and there is no foreseen change in this in the immediate future.

There is a clear benefit in providing a consistent high-level Client API (Client API) deployed in the infrastructure to provide a uniform functional API across all software services deployed on the resource level. Sustainability however will have to be driven by the user communities that use, or wish to use, such a client-related API deployed by EGI on behalf of these user communities.

Complementing either a deployed cross-resource Client API, or specific services deployed in the EGI production infrastructure, client tools (Client Tools), such as graphical user interfaces, command line tools or interactive shells provide convenient end-user access to Grid resources and community services. Sustainability will have to be driven by the respective user communities that make use of the offered software.

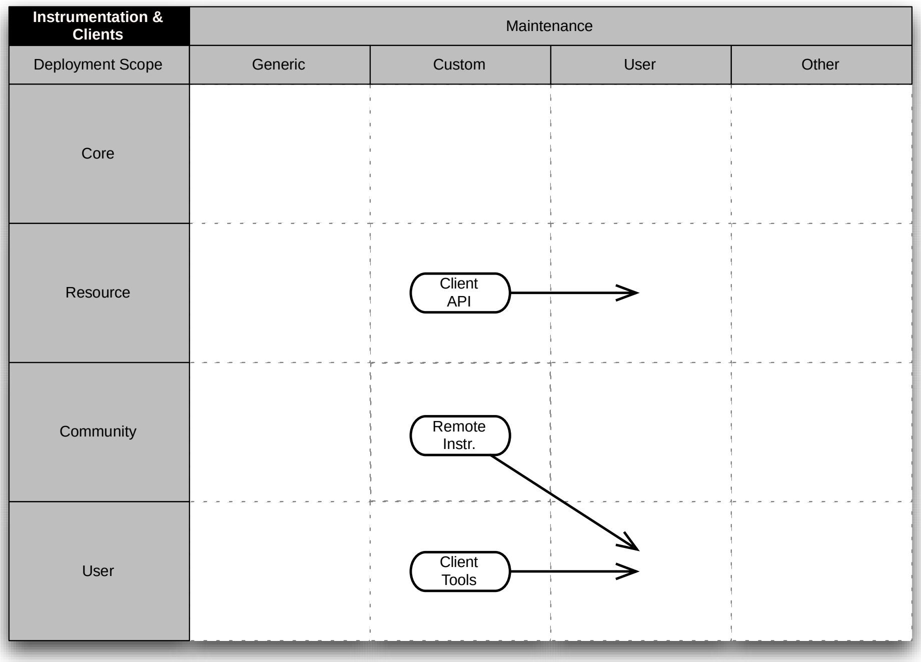


Figure 9: EGI Instrumentation and Client capabilities radar

4.2 Technology Provider Roadmaps

4.2.1 European Middleware Initiative (EMI)

This section provides a synthesis of the EMI technical development plan [R 18].

4.2.1.1 Security

EMI took the strategic decision to consolidate the provided security infrastructure. This includes not only a harmonisation and re-use of the client libraries in their published products, but also consolidation and integration with the security services that are part of the EMI software portfolio. EMI's security architecture is firmly rooted in the use of X509 based public key infrastructures, being extended with basic support for SAML 2.0 support. For both types of authentication tokens, EMI is developing a common profile of attributes to identify users and their associated VOs.

EMI provides VOMS as the reference implementation for attribute authorities. Support and maintenance for VOMS is planned to be released into public domain and as part of standard Linux distributions such as scientific Linux and Debian, by 2011 or at the latest by the end of the project. Currently there are no plans known to standardize the access and management interface of VOMS. On the backend side VOMS already issues authentication tokens in both SAML 2.0 and RFC 3281 compliant attribute certificates.

In support EMI's security infrastructure consolidation its distributed authorisation framework, Argus, will be the integration point for distributed software services provided by EMI and other technology providers (for example, IGE, see below). Based on the standardised SAML 2.0 and XACML 2.0 interfaces, the first release of Argus has already been deployed in the EGI infrastructure. Initially integrated with few EMI services, the integration and support base of Argus will be continuously expanded until the second major EMI release in April 2012.

Instead of providing a credential management solution on their own, EMI is already integrating with the MyProxy service, delivered by IGE (see below).

4.2.1.2 Information

With the delivery of EMI-1, EMI started the transitioning process to full support of GLUE2. By mid 2012, EMI will deliver EMI-2, which will fully support GLUE2 in all its provided services and components. The first service fully supporting GLUE2 was BDII as part of the EMI-1 release in 2011. EMI will continue maintaining BDII throughout the duration of the project.

EMI is investigating the use of a messaging infrastructure as an alternative means for communication between distributed services, but no clear direction or roadmap has been developed.

4.2.1.3 Operations

Like other technology providers, EMI is not directly delivering solutions in this area. EMI has committed to taking over the maintenance and further development of Nagios probes to support the core EGI monitoring capability.

EMI also provides CPU accounting functionality through sensors (i.e. log file parsers) that extract the information directly at the functional service (e.g. CREAM). An intermediate component, the APEL publisher, takes that information and passes it to the EGI APEL accounting repository through the afore-mentioned messaging network. The various numbers and types of accounting modules are subject to consolidation and harmonisation in EMI. Furthermore, EMI is actively contributing and developing a standardised set of storage accounting records through the OGF Usage Record (UR) working group. Currently, no publication date for standardised storage record language is known.

4.2.1.4 Storage

EMI is consolidating the access interfaces and protocols for storage services towards a set of already developed public domain standards such as HTTP, WebDAV and POSIX. Support for those is partially already provided by EMI's storage services StoRM, dCache and DPM.

Full support for POSIX is planned to be delivered with EMI's second major distribution, EMI-2 in spring 2012. GridFTP is supported by all relevant EMI services and will be complemented by increased support for HTTP and WebDAV, although without clear delivery dates. EMI's existing storage services that already support the SRM interface, such as StoRM and DPM provide full interoperability through the use of the FTS service. EMI plans to publish FTS in Autumn 2011 in a re-engineered version.

By April 2012, with the release of EMI-2, EMI plans to provide a unified service implementation of the SRM standard, followed by a consolidated client side access to SRM services through a common set of libraries with the release of EMI-3 in April 2013. EMI is planning to release Hydra as a file encryption and decryption service in Autumn 2011.

4.2.1.5 Data

EMI is providing two complementary services for matter data management. AMGA provides facilities for mapping user generated matter data onto location independent data descriptors while LFC provides a mapping mechanism for said location independent data descriptors onto concrete data location pointers for direct use with services that support SRM.

4.2.1.6 Compute

EMI provides three independent distributed computing services that implement job execution and parallel jobs support. Through active contribution to OGF Production Grid Infrastructure (PGI) working group, EMI is collaboratively developing the PGI Execution Service (ES) specification that in turn will be supported by the provided compute services. The timeline and delivery of the ES specification support depends on the progress made in the OGF PGI working group. However, some execution services within EMI (e.g. UNICORE) support the standardised interfaces OGSA-BES [R 40], JSDL [R 41] and HPC-Basic Profile [R 42]. UNICORE components are part of the Unified Middleware Distribution (see section 4.3 for more detail).

Beyond support for basic interactive job management, planned to be delivered in September 2012, EMI has no plans for any evolution of this capability. With WMS (Workload Management System) EMI provides the job scheduling capability within EMI-1.

4.2.1.7 Clients

EMI provides an extensive set of client tools in support of its provided Grid services. Currently bundled in four different installation profiles, EMI is consolidating its Client tools portfolio by first introducing common EMI grid middleware access libraries, followed by a further consolidation into one client tools offering for end users.

EMI provides partial support for the SAGA API. The requirements and effort for complete support of SAGA API need to be evaluated and eventually planned.

4.2.2 Initiative for Globus in Europe (IGE)

This section provides a synthesis of the IGE roadmap contributions grouped by EGI Capability area.

4.2.2.1 Security

IGE's strategy for security capabilities can be described as harmonisation and integration wherever possible. A strong push towards re-using either generic software or software that is very popular in the Grid community is evident.

To provide compatibility with common security capabilities required in the EGI production infrastructure, IGE is working together with the Globus Alliance to complete the transition from httpg to https based on standard SSL and X.509v3 certificates in an IGE release in 2012. In support for Virtual Organisations, IGE plans to start integrating with VOMS on the basis of exchanging SAML tokens in early 2012.

Furthermore, IGE moves towards standards based authorisation solutions using SAML 2.0 and XACML 2.0, with EMI's Argus explicitly listed as an integration candidate. Concrete integration work will be undertaken in 2012 and it is unclear by when this will be completed.

IGE, through Globus, provides the MyProxy certificate storage and management solution. To accommodate the expressed User Community requirement to support more authentication and access mechanisms besides X.509 certificates, IGE, with the Globus Alliance, is investigating two parallel solutions for an enhanced Credential Management system:

a) CILogon [R 19], a solution that integrates Shibboleth based federated institute authentication with MyProxy certificate issuance to access Grid resources.
b) A combination of MyProxy and OAuth [R 20], a similar solution as CILogon, but based on the popular web-based federated authentication mechanism OAuth.

4.2.2.2 Information

IGE is supporting GLUE as a critical language to describe and publicise information about a Grid production infrastructure. Predominantly using GLUE 1.3, the Globus Integrated Information Service (IIS) provides GLUE2 support as a proof-of-concept. Transitioning to full GLUE2 support has started, but an end date is not yet known.

Tracking the development and rise of interest in messaging infrastructures, IGE is committed to support AMQP [R 21] through the Globus Alliance. However, no definitive completion dates are known.

4.2.2.3 Operations

Like other Technology Providers, IGE is not directly delivering solutions in this area. IGE has committed to taking over the maintenance and further development of Nagios probes to support the core EGI Monitoring Capability. To provide for Accounting information, IGE is working towards the inclusion of Grid-SAFE [R 22] into its product portfolio, as a local accounting aggregator that will be capable to forward accounting records in UR format to the EGI global accounting service in 2012.

4.2.2.4 Storage

IGE, through the Globus Alliance, is providing Globus Online, a storage cloud for location-independent data access and management features. Through Grifi [R 23] and Parrot [R 24], IGE is looking into providing POSIX based file access to remote storage, hiding the complexities of accessing and manipulating remote data behind a local data access façade. This is made possible through a combination of Globus Online and GridFTP, which have been both available as standard IGE service offerings since mid-2011.

Through this integrated offering, dedicated file transfer or file transfer scheduling services become obsolete, and therefore IGE will not provide implementations for those capabilities. Also, the design allows data encryption applied transparently at the user or community end of the storage access infrastructure. Consequently, there are no plans of IGE to provide dedicated data encryption and decryption services.

4.2.2.5 Data

IGE is working towards providing a standalone community based Web Service exposing an access interface that implements the standards WS-DAIX and WS-DAIR (both standardised in the OGF OGSA-DAIS WG [R 25]), both in SOAP and RESTful renderings. The first release is expected in IGE 2.0 in early 2012.

In support for user communities to organise their data references, IGE is providing the Globus Replica Location Service (RLS) as a community service to map abstract or human readable file names into location descriptors. RLS is part of IGE 1.0.

4.2.2.6 Compute

IGE delivers with GRAM5 a Grid compute service that accepts classic Job submissions as well as parallel job definitions, as part of IGE-1. IGE plans to provide support for OGSA-BES, JSDL and HPC Basic Profile in early 2012.

IGE plans to integrate GridWay [R 26] as a job scheduling solution into its IGE-2 distribution in early 2012. By that time, GridWay will also support OGSA-BES, JSDL and HPC Basic Profile on both the service level, and as a client to any compute service that implements these standards. Additionally, GridWay will support native CREAM job submissions. GridWay provides basic DAG workflow execution, but IGE has no plans to provide workflow support beyond that since community-specific workflow solutions (such as Taverna) already interoperate with GridWay and GRAM5.

To allow users to interact with the currently running compute jobs, IGE has been offering GSISSH as part of IGE-1 since mid-2011.

4.2.2.7 Clients

IGE plans to support the SAGA API in Globus Toolkit 5.2 and GridWay in 2012. In support of the SAGA project, which is providing the API mapping libraries to satisfy the Client API Capability (see below), IGE will maintain the Globus-specific SAGA plug-ins to jointly deliver the Client API capability by 2012. The exact way to take over the plug-in maintenance, and further support of the SAGA project for the plug-in development will also be investigated in early 2012.

4.2.3 A Simple API for Grid Applications (SAGA)

The SAGA project is planning to provide an implementation for the “Client API” capability to enable user communities to integrate on a common API across all available Grid resources. The architecture of the SAGA implementation includes the definition of two interfaces: The public interface for the user-level applications is defined and maintained in OGF by the mandate of the SAGA WG – the SAGA project implementation acts at the same time as reference implementation for the SAGA interface specification. The second interface is exposed as a capability provider interface, through which Grid middleware provider may bind their implementations to SAGA through a plug-in.

The SAGA project supports, among others, the Globus Toolkit 5, gLite 3.2, Condor and SSH as back-ends. With the inclusion into the EGI production infrastructure, the maintenance of the middleware plug-ins for SAGA will transition from the SAGA project to the respective middleware providers: IGE (see above) is investigating the inclusion of the SAGA support for Globus into IGE software for 2012. No definitive statement was given by EMI up to the time of writing this roadmap.

4.2.4 Main Stream Open Source Community Components

Parallel to EGI Technology Providers, software and technology is made available in the general-purpose software landscape. This section provides an overview of software, classified as generic software as defined in section 3.3.3, that is identified as a potential complement or replacement of software that is currently sourced from within the EGI community.

4.2.4.1 Messaging

The AMQP (Advanced Message Queuing Protocol) Working Group aims at providing a programming language-neutral and platform-agnostic message queuing specification. The specification was published in version 1.0; the vote for being released as a final specification is expected for October 2011. Several competing but interoperating implementations cover both commercial solutions (e.g. Red Had Enterprise MRG) and open source (e.g. VMware RabbitMQ, Apache QPID). Being developed outside the EGI community, any of those products has the potential for delivering messaging needs in the EGI community, whether core infrastructure or user community-specific, or both.

A viable alternative is the open source STOMP framework [R 27] with broad support for programming and scripting languages. Several independent implementations exist as listed at STOMP's website.

4.2.4.2 Workflow

There are many workflow engines within the open source scientific and computer science community, as well as more business process orientated workflow implementations. For example, the SHIWA project [R 28] provides an approach for scientific workflows to be mapped on concrete Distributed Computing Infrastructures (DCIs) to allow the execution of large-scale scientific experiments.

4.2.4.3 Monitoring

The core component of the Monitoring capability is the Nagios framework. Probes are provided by the technology providers to extract monitoring data from the services. Current probes vary greatly in the type of data they extract. While some probes extract service availability information from a service client point of view, some probes extract service management data (such as job queue length).

It is expected that three types of monitoring data will be extracted from services, evolving with a clearer definition of the semantics of each type and target audience:

- Service availability ("up and running" over time).
- Service Management, exposing key service status information, e.g. thread pool size, memory consumption.
- Service Quality, to measure how well in non-functional terms the service is delivered, e.g. number of job submissions accepted per minute.

While the EGI production infrastructure perhaps intrinsically will never be in the position to use generic software for *gathering* monitoring data, using available solutions for *processing* the gathered data is a viable option in transitioning towards a business intelligence oriented operational infrastructure and management. Using techniques such as Online Analytical Processing (OLAP), i.e. storing gathered data in a format and structure suitable for generic reporting solutions, allows to focus on the expertise in modelling and gathering the business-relevant information, and use available solutions such as Jasper [R 29], Pentaho [R 30], and Eclipse BIRT [R 31] to generate reports necessary to manage and evolve the production infrastructure.

4.2.4.4 VM Management

Many different implementations for VM Management are available, ranging from low-level VM execution support (e.g. Linux Kernel KVM) to high-end VM Management solutions, such as VMware solutions. Various implementations of Open Cloud Computing Interface (OCCI) exist, such as OpenNebula and Eucalyptus, which should be taken into consideration before providing custom solutions.

4.2.4.5 VM Image Format

It is important to embrace and understand the notion of the difference between a binary VM disk image format and an interoperable VM container definition (i.e. the OVF "envelope"). OVF supports a range of currently used disk formats, but provides an extensibility point for the future use of new disk image formats. VM Management solutions must therefore provide support not only for the various metadata artefacts defined in the OVF specification, but also for a variable set of VM disk formats as described in the specification (e.g. ISO images, VMware vmdk format).

4.2.4.6 Remote Instrumentation Capability

Some proprietary implementations are available from the GridCC, DORII and DILIGENT projects. Also, EGEE's RESPECT programme included applications for remote instrumentation.

4.2.4.7 Operating Systems

There are uncountable numbers of Operating Systems available. The by far the largest ecosystem of different Operating Systems is available in the GNU/Linux community, where countless different flavours of comparatively few base distributions follow many different support and software update policies and procedures. Surprisingly, virtually all those different distributions can be traced back to two, perhaps three influential base systems in terms of software support, and low-level package distribution and management infrastructure:

- RedHat, using the RPM based package management infrastructure,
- Debian, using the DEB package distribution infrastructure, and
- SUSE, originally based on the own-developed YaST packaging format, now adopting RPM format, but still using YaST as package management tool.

To support new communities in a sustainable way, EGI should consider the consolidation of its Operating System base towards a set of selected distribution providers that satisfy the following requirements:

- Within each published distribution, provide different installation profiles for different general use cases, such as server, desktop, mobile.
- Provide clear policies of distribution support in terms of support for applications as such, and application updates.
- Offering dual licensing of the distribution portfolio, or similar models for the sourcing of paid expert level support (e.g. a free distribution base, but paid support beyond community contributions).

Candidate for evaluation against such a set of criteria are, almost naturally, distributions that are backed by commercial providers, such as Canonical, Red Hat or Novell.

4.3 UMD Roadmap

The UMD Roadmap describes the planned release schedule for UMD major and minor releases over the upcoming 6 months and beyond based upon the information we have available from our technology providers. Any software that is considered for inclusion into UMD will be taken through the EGI Software Provisioning process [R 32] before it is made available for production use. This snapshot of the UMD Roadmap will be updated at regular intervals as more information becomes available (https://wiki.egi.eu/wiki/UMD_Release_Schedule).

UMD Major releases (such as UMD 1.x, UMD 2.x, etc.) will be supported and updated while EGI is provided with updates from it technology providers for at most two consecutive years from the date of their initial release. UMD major releases will be made when non-backwards compatible changes need to be made to the software components within it. Minor releases within a major series (e.g. UMD 1.x.0) may introduce new functionality, but existing interfaces and behaviours will remain. While EGI aspires to follow this plan as closely as possible, the UMD Release Plan is dependent on the software provided by our external technology providers and the quality of information (if provided) and their ability to meet their announced release dates. Therefore, there will be no guarantee on release dates and contents other than the timely communication of changes as they become apparent. In particular, the time taken to include a product into a certain UMD release greatly depends on diligent and volunteer-based Staged Rollout activity [R 33]. If no volunteer picks up this Product and exposes it to the Production Infrastructure, then that given Product is in danger of not being included in the planned UMD release and will be shipped later.

Specifically for UMD 1, an initial phase of frequent releases is planned without fixed publication dates to provide as early as possible a most complete UMD as the limited efforts allow. Once this goal is achieved, EGI is planning to provide regular UMD updates on a three-month schedule. UMD 1 will be available on Scientific Linux 5 (x86_64) and be supported until April 2013. UMD 1.0.0 in July 2011 provided an initial release of software components from EMI 1.0. IGE products are tentatively scheduled for inclusion in UMD 1.2.

4.3.1 UMD 1.2.0

EGI plans to release UMD 1.2.0 on 12 September 2011. The predecessor version, UMD 1.0.0 and UMD 1.1.0 were already released into production as per original schedules. Since software release schedules may change frequently, the UMD Release Schedule and planned contents are maintained at https://wiki.egi.eu/wiki/UMD_Release_Schedule. With the release of UMD 1.2.0, the initial provisioning of software in the UMD is finished. Although the UMD will not initially include SAGA software, it will provide software solutions to cover all of the EGI capabilities deemed most critical for its user community.

4.3.2 Outlook

With UMD 1.3.0 and beyond, EGI will strive to enter into a regular update cycle of quarterly updates. This however requires that EGI is regularly updated, in due time, with specific release plans of the Technology Providers. This represents a significant change in internal processes for the existing Technology Providers as opposed to the last decade where releases were made with little forward planning. While Technology Providers are well aware of this requirement, it proves difficult to change product team processes towards detailed planning and communication. Therefore, it is at this point in time not possible to reliably plan ahead into the future and provide UMD release schedules that go beyond UMD 1.2.0 in mid-September 2011.

5 CONCLUSIONS

5.1 *Implementation & Deployment Analysis*

The analysis provided in this report of the different functional capabilities and their deployment scope within EGI enables the groups (resource providers, virtual research communities, end-users) that are the major consumers of the capability to start identifying how to sustain the capabilities that are critical for them.

Capabilities that are essential for EGI's core functionality are supported by generic components for the information, monitoring and virtualisation (even though these have not yet been frequently deployed) capabilities. The security capabilities need to move away from their current heavily customised software base to one that better leverages generic software. While the accounting requirements for EGI are unlikely to become generic, other production infrastructure providers can potentially support them.

The adoption of common protocols (e.g. http, WebDAV) within some of the storage services allows the EGI community to start to leverage existing client tools. However, many of the storage, data and compute related capabilities deployed on resources or for the community have bespoke protocols and requirements to particular communities, and therefore the support by these communities of these services would provide the best long-term sustainability,

5.2 *Funding analysis*

The operational core of the production infrastructure that is dependent on the EGI core capabilities (as assessed in section 4.1) needs to strive to be self-sufficient for the software maintenance, development and its operation. It is therefore critical that a significant proportion of this software needs to be drawn from generic components (supported primarily outside of EGI) with minimal customisations being needed by EGI. Many of these capabilities provide a foundation for all other activities within EGI and therefore 'opting-out' of these services by individual resource providers would be very difficult.

The usage of different capabilities deployed on the resources can vary significantly between communities. Clearly, capabilities used by just one or a few communities need to draw heavily on these communities for their maintenance, development, operation and support. It is possible for capabilities that truly span all communities could be candidates for support through the infrastructure. However, given the probable differences in usage between capabilities by different communities, a usage based cost model may be appropriate for proportioning costs.

5.3 *Next steps*

This first version of the EGI Technology Roadmap has been based on the capability definitions presented in previous versions of the UMD Roadmap. It has also categorised these capabilities with respect to the scope of their deployment – within the infrastructure, to provide access to a resource, to provide a coordination capability to a community, or as a client environment to provide access to these capabilities. The implementations currently available for these capabilities are also categorised by their maintenance support model – generic by a broad community, custom within the EGI community, or dependent on the user community for its support.

The categorisation of functionalities and the classification of the implementations will now be socialised within the community and feedback gathered. Alongside this feedback an assessment will be made as to the capacity of the infrastructure and the user communities to maintain and operate the software that is allocated to them. The capacity of these consumers and the categorisations can then be balanced to ensure a long-term sustainable solution is available for all of EGI's stakeholders.

6 REFERENCES

R 1	Digital agenda for Europe, http://go.egi.eu/DigitalAgendaEurope
R 2	European Innovation Union, http://go.egi.eu/InnovationUnionEurope
R 3	Europe 2020, http://go.egi.eu/Europe2020
R 4	UMD Roadmap, 1st edition, EU Deliverable D5.1**,** http://go.egi.eu/UMDRoadmap-1
R 5	UMD Roadmap, 2nd edition, EU Deliverable D5.2, http://go.egi.eu/UMDRoadmap-2
R 6	EGI Sustainability Plan, EU Deliverable D2.7, http://go.egi.eu/313
R 7	EGI Design Study Project, http://web.eu-egi.eu/
R 8	Shibboleth, http://shibboleth.internet2.edu/
R 9	OpenID, http://openid.net/
R 10	SAML 2.0, OASIS, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security
R 11	Internet Attribute Certificate for Authorisation, IETF RFC 3281, http://www.ietf.org/rfc/rfc3281.txt
R 12	XACML 2.0, OASIS, http://www.oasis-open.org/committees/xacml/
R 13	GLUE Schema 1.3, OGF, https://forge.gridforum.org/sf/go/doc14185
R 14	GLUE Specification 2.0, OGF, http://www.ggf.org/documents/GFD.147.pdf
R 15	ActiveMQ, http://activemq.apache.org/
R 16	JMS 1.1, JSR-914, http://jcp.org/aboutJava/communityprocess/final/jsr914/index.html
R 17	Nagios, http://nagios.org/
R 18	EMI Technical Development Plan, D1.3.2, http://cdsweb.cern.ch/record/1277543
R 19	CILogon, http://www.cilogon.org/
R 20	OAuth, http://oauth.net
R 21	AMQP, Advanced Message Queuing Protocol, http://www.amqp.org
R 22	Grid-SAFE, http://sourceforge.net/projects/gridsafe/
R 23	Grifi "GridFTP File System", http://grifi.sourceforge.net/

R 24	Parrot Virtual File System, http://www.cse.nd.edu/~ccl/software/parrot/
R 25	OGF DAIS WG, http://forge.gridforum.org/projects/dais-wg
R 26	The GridWay metascheduler, http://www.gridway.org
R 27	STOMP “Streaming Text Oriented Message Protocol”, http://stomp.codehaus.org/
R 28	SHIWA Workflow engine, SHIWA project, http://shiwa-workflow.eu
R 29	Jaspersoft, http://www.jaspersoft.com
R 30	Pentaho, http://www.pentaho.com/
R 31	Eclipse BIRT, http://www.eclipse.org/birt
R 32	EGI Software Provisioning Process, MS508, https://documents.egi.eu/document/505
R 33	EGI Staged Rollout process, MS409, https://documents.egi.eu/document/478
R 34	EGI Quality Criteria version 1, http://go.egi.eu/qualitycriteria-1
R 35	EGI Quality Criteria version 2, to be published
R 36	Usage Record WG, OGF, http://www.ggf.org/gf/group_info/view.php?group=ur-wg
R 37	DCI Federation WG, OGF, http://www.ggf.org/gf/group_info/view.php?group=dcifed-wg
R 38	Open Cloud Computing Interface WG, OGF, http://www.ggf.org/gf/group_info/view.php?group=occi-wg
R 39	Virtualization Management (VMAN), DMTF, http://www.dmtf.org/standards/vman
R 40	OGSA-BES: Open Grid Services Architecture – Basic Execution Service, http://ww.ogf.org/documents/GFD.108.pdf
R 41	JSDL: Job Submission and Description Language, version 1.0: http://www.gridforum.org/documents/GFD.136.pdf
R 42	HPC-BP: High Performance Computing Basic Profile, http://www.ogf.org/documents/GFD.114.pdf
R 43	

APPENDIX A EGI CAPABILITIES

The EGI Capabilities were developed from the beginning of EGI, and documented in the first and second iteration of the UMD Roadmap [R 4 and R 5]. Capabilities describe a specific functionality that is required to deploy and operate the EGI production infrastructure, using technology neutral language wherever possible. EGI Capabilities may describe a service deployed in the production infrastructure, or a cross cutting language that is required to allow services to interact and interoperate with each other.

A.1 SECURITY CAPABILITIES

Security capabilities form an important foundation of a distributed production infrastructure, for obvious reasons. The challenge is, however, to carefully model the Security Capabilities so that no unintentional dependencies creep into the architecture, and that a clear boundary definition allows for scalable distribution of Security Capabilities within the production infrastructure.

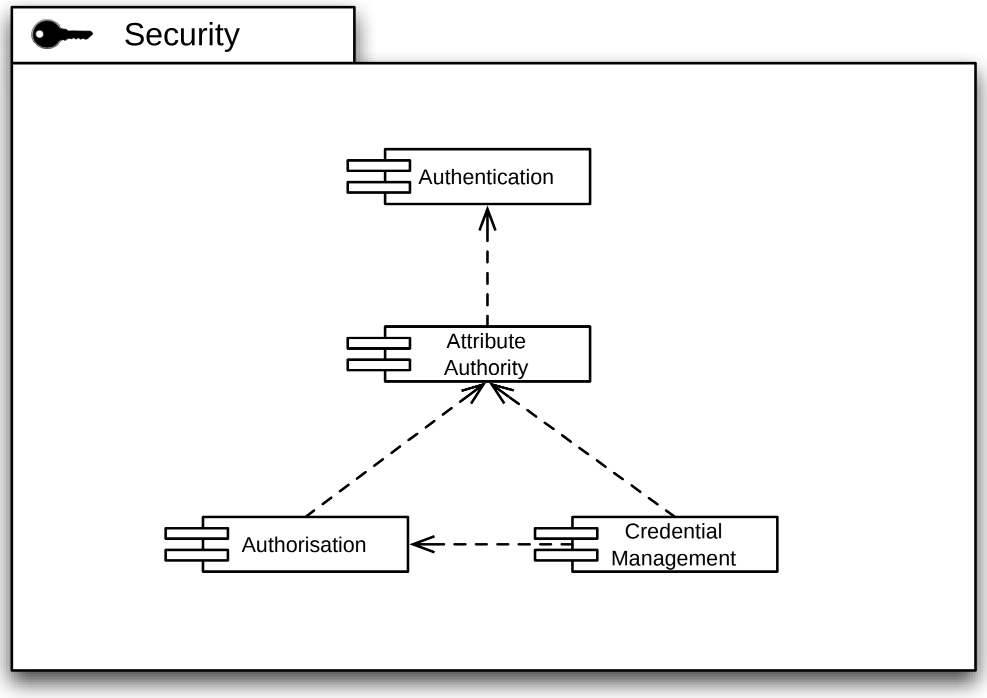


Figure 10: Dependencies of the Security Capabilities

The establishment of a secure technical identity of an individual across the production infrastructure is a key capability. The acceptance of a set of identity providers (whether federated or centralised as in the current model of X.509 based authentication) is required to lay the basis for an identity/attribute approach to the identification of a user within the infrastructure.

Accepting that a technical identity is not enough to accommodate all use cases for the Grid, an Attribute Authority establishes the concept of roles, or context-based identity of a Grid user: The same user may within one context be an administrator of a site, but at the same time a scientist conducting research using the Grid (as a scientific user) in a different context. Albeit the same technical identity, the roles in both contexts are clearly separated yet securely affixed to the pertinent user identity. In many cases, the Authentication Authority and the Attribute Authority may be identical. However, in federated authentication scenarios the Attribute Authority is in most cases located within the perimeter of one or more VOs the user may be affiliated with.

Based on the attributes securely affixed to an identity, resources and services in the Production Infrastructure must make decisions to allow or deny access, based on the attribute-decorated identity information and any rules stored at that resource or service.

Many use cases of the Grid require the concept of delegation of trust, mostly for procedural purposes, or to comply with policy on a certain site. For those use cases, issuing credentials on demand based on a long-term established identity is a key feature of Credential Management. It is important though that access to on-demand issuing of credentials is guarded through authorisation mechanisms. Interestingly enough, Credential Management itself thus provides at the same time Authentication and Authorisation functionality from a service point of view.

A.1.1 Authentication Capability

An authentication token that is strongly bound to an individual must be applied consistently across the software used within the production infrastructure. The authentication system must be capable of supporting a delegation model.

Irrespective of the actual format, and infrastructure employed, authenticating an individual is a two-step process of first creating and issuing the token, and second to establish the trust in the presented token by cryptographically verifying the integrity against a set of well-defined trust anchors.

Simple authentication infrastructures, such as classic username and password systems often co-locate token issuance and trust establishment in a single conceptual location, for example a username and password file (e.g. "/etc/passwd" in Linux systems) or a distributed replication (e.g. the EGI.eu SSO system).

More complex authentication infrastructures separate the token issuance from establishing the trust in a presented authentication token. A PKI infrastructure as it is employed in the EGI production infrastructure takes the token issuance service completely offline, reducing the actual establishment of trust as a configuration detail: The architecture of the PKI allows a generic implementation of a cryptographically secure verification of a X.509v3-based certificate chain, resulting in a fundamental trust decision (i.e. to trust or not to trust the presented authentication token) based on configuration – as available in the EGI Software Repository (https://repository.egi.eu).

A.1.1.1 Supported Interfaces

The primary authentication token within the infrastructure is the X.509v3 certificate and its proxy derivatives. Accessing resources through protocols that are secured using SSL or TLS (e.g. plain socket, or https connections) must employ X.509v3 certificates

An alternative, standards based Authentication language is SAML 2.0 allowing for more flexible federated authentication solutions geared towards the end-user of the Grid.

OpenID, and Shibboleth are community driven solutions based on SAML, providing federated authentication mechanisms.

PKI-based Authentication (and eventually, also Authorisation) is sustainable in slow-changing environments; otherwise the management effort becomes too high.

In an environment, where individual user affiliation with a project, a VO, or an experiment may range from months to years, a reliable prediction of domain-specific AAI requirements is difficult across all current and future user communities.

A.1.2 Attribute Authority Capability

Resources within the production infrastructure are made available to controlled collaborations of users represented in the infrastructure through Virtual Organisations (VOs). Access to a VO is governed by a VO manager who is responsible for managing the addition and removal of users and the assignment of users to groups and roles within the VO.

The main service that any Attribute Authority provides is the issuance of signed tokens that express a subject’s extended information such as VO membership, special roles within an organisational context etc. Typically, an implementation comes with a proprietary administration interface or functionality.

A.1.2.1 Supported Interfaces

Two types of interfaces are required for an interoperable Attribute Authority implementation:

- A language and format containing the actual attribute statement
- An access interface for clients to contact the Attribute Authority service. Administration interfaces are considered an implementation detail.

SAML 2.0 provides for a standardised language to express authoritative statements about a subject’s attributes, particularly in scenarios that employ mechanisms of federated identity. The prevalent means of expressing subject attributes reuse PKI X.509 certificates, standardised in RFC 3281. Currently, there is no standardised access interface known for Attribute Authority services to implement.

A.1.3 Authorisation Capability

The implementation of access control policy – authorisation – needs to take place on many levels. Sites will wish to restrict access to particular VOs and individuals. Sites or VOs may wish to stop certain users accessing particular services. The infrastructure as a whole may need to ban particular users. Policy Enforcement Points (PEPs) will be embedded into many components throughout the infrastructure and will use Policy Decision Points (PDPs) to drive access control decisions.

In a service oriented Grid infrastructure the Policy Decision Point provides the fundamental service to other services, including any number of Policy Enforcement Points. Often a Policy Information Point (PIP) is co-located with a Policy Decision Point, providing human-readable renderings of the technical access policies stored in the PDP.

A number or use cases mandate that implementations must support distributed deployment of the PDP (and perhaps the PIP), for example for performance reasons, or policy management reasons. In such scenarios all PDP services must expose an identical set of access interfaces and information description languages.

A.1.3.1 Supported Interfaces

SAML offers basic Authorisation mechanisms. However, the combination of SAML and XACML from OASIS form a perfect couple for any authorisation needs.

XACML provides clear definitions and scope for PEPs and PDPs, allowing different implementations along those interface definitions deployed in the infrastructure.

Another industry quasi-standard is OAuth , which provides for authentication and authorization for delegated access to specific data. Facebook, amongst others, is a main driver of OAuth.

A.1.4 Credential Management Capability

The Credential Management capability provides an interface for obtaining, delegating and renewing authentication credentials by a client using a remote service.

A.1.4.1 Supported Interfaces

One of the key functionalities in this area is the linking of institutional authentication systems to the transparent issuing of certificates for use in the infrastructure through identity federations. This should be provided for community deployment through the use of web portals and web service interfaces.

Being fundamentally a community-driven service, it nonetheless sits on the seam towards the use of the generic production infrastructure and thus depends on the currently EGI-wide deployed authentication AAI infrastructure.

With the drive towards a clearer and stricter separation of Core and Community Capabilities, the technical dependencies of Credential Management implementations will transition to depend on AAI solutions chosen for the respective community domain.

A.2 INFORMATION CAPABILITIES

Information is key in distributed infrastructure. Both users and administrators need to know which services are deployed in the infrastructure, which resources are available for consumption or are saturated with compute or storage requests by users, etc.

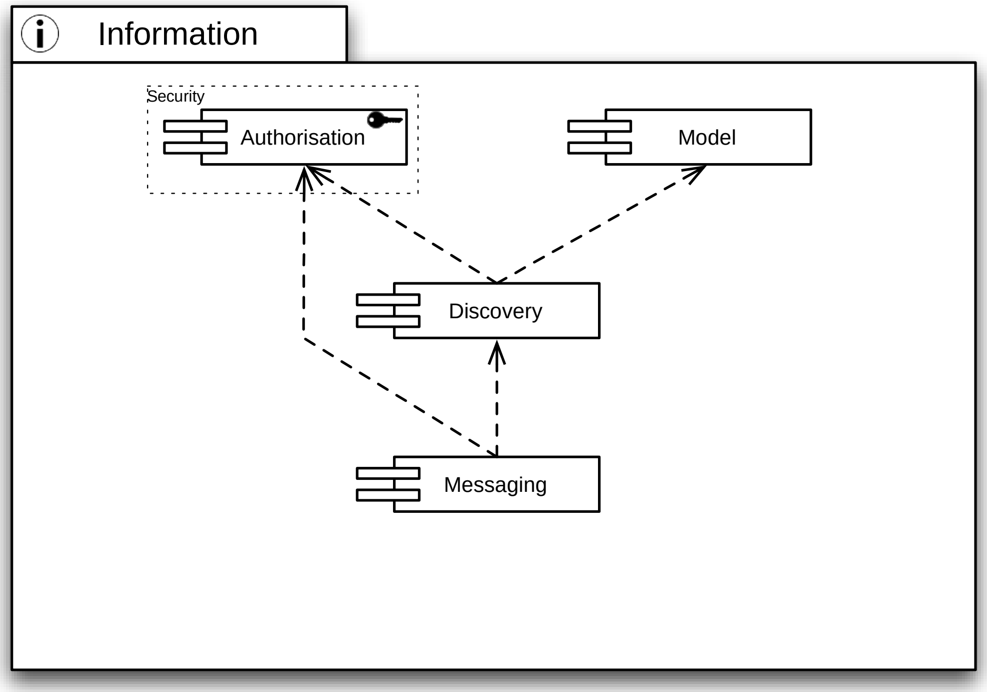


Figure 11: Dependencies of the Information Capabilities

It is quite obvious that a common language must be available. The Model Capability provides such a language of modelling resources present in the infrastructure, their connections and dependencies, and a common understanding of how to interpret the constructs of language elements to model a given resource.

With a common language at hand, presence and availability discovery of services and resources is possible without ambiguity. Through a well-known "lighthouse" approach in discovery services, users may learn of, or discover, services that may be helpful in solving the user's needs by querying those information services with search expressions.

Connecting services with each other is the primary use case for a messaging infrastructure in order to serve a set of given inter-service communication patterns. However, to programmatically (or automatically) connect those services with each other, the messaging facilities and channels must be well known – hence they must be discoverable and searchable through the Discovery Capability.

However, not all services or messaging endpoints may be accessible to any user that is generally allowed to access the Grid. Hence proper distributed Authorisation services are required to allow any level of granular access to services deployed in the infrastructure.

A.2.1 Information Model Capability

When exchanging information about services, resources, and data a common understanding of the metadata describing such entities is necessary. A common definition of the terms, the syntax, and semantics of basic and complex metadata structures ensures interoperability and integration of systems from different Technology Providers, when exchanging metadata in requests and responses.

A.2.1.1 Supported Interfaces

The information about the resources is described using the GLUE schema from the Open Grid Forum. Currently this is GLUE 1.3 with migration underway to GLUE 2.0.

A.2.2 Messaging Capability

Any kind of distributed computing system faces the challenges of participating services having to communicate with each other. Though many such challenges deal with system design and architecture, common messaging language, protocols and patterns connect the services to each other. If not the only patterns, most commonly used in distributed systems are *store-and-forward* (JMS calls this point-to-point) and *publish-subscribe* patterns of using messaging.

Establishing a messaging infrastructure solves many scalability issues commonly found in distributed systems.

Within distributed systems, a message 'bus' provides a reliable mechanism for data items to be sent between producers and (multiple) consumers. Such a capability, once established, can be reused by many different software services.

A.2.2.1 Supported Interfaces

The Java Message Service (JMS) is the de-facto standard for Java based messaging systems. In its current version 1.1, JMS is used widely in the commercial world even as means for Enterprise Application Integration. Supporting *publish-subscribe* and *point-to-point* modes, JMS provides for all messaging patterns in distributed computing. Although not language agnostic, adapters for programming languages other than Java are readily available through Apache ActiveMQ.

Emerging from the financial industry, AMQP provides a standard interface for messaging on the lowest integration layer, the wire layer. The AMQP Working Group aims to develop a messaging protocol that is eventually standardised and stewarded by a recognised SDO, such as the IETF. AMQP is fully language agnostic, and defines a message format at the byte level, intentionally leaving the payload structure unspecified. Those two features allow indiscriminate implementation for any given programming language, and offers integration and interoperability for any kind of applications, architectures and networks.

Messaging is clearly a Core Capability required for efficient and scalable management of the EGI production infrastructure.

Messaging gains popularity in custom software solutions as a means to establish asynchronous communication to gain scalability, or for software components that are otherwise difficult to integrate on the access layer. Therefore Messaging is an emerging capability to be incorporated into User community-sourced solutions.

A.2.3 Information Discovery Capability

Information discovery is a capability that helps find the required resources that have been registered with it within the production infrastructure. The information collected about such resources is made available through well-known instances that provide the data to some logical collection, infrastructure wide, regional, site, domain, etc.

Clients to such service must be able to search, filter, and order the available information until their initial request is satisfied.

To enable search and discovery on various levels of the infrastructure it is important to reiterate that any implementation of the Information Discovery Capability must at the same time make use of the Information Model capability defined earlier in this document.

A.2.3.1 Supported Interfaces

The LDAPv3 (RFC 4530) protocol and search syntax is used to query information from the information discovery services and to encapsulate the information payload relating to the services being offered within the production infrastructure that is exchanged between instances.

There are currently no interfaces defined for an interoperable management of the data that is published through this Capability, except for implementation specific interfaces.

However, a desirable integration might re-use implementations of the Messaging Capability to enable scalable dissemination and management of the published information.

A.3 OPERATIONS CAPABILITIES

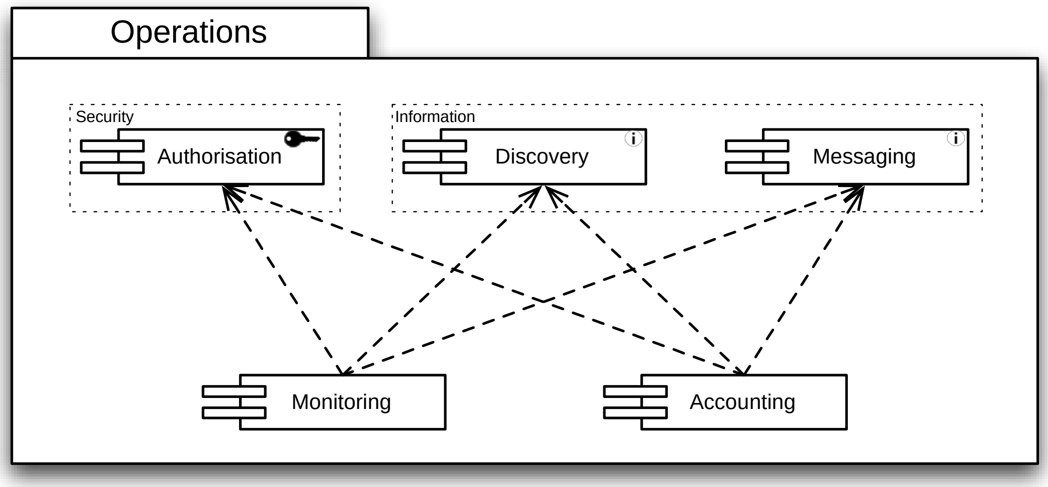


Figure 12: Dependencies of the Operations Capabilities

Maintaining and administering a production infrastructure poses a number of requirements on the deployed components. Monitoring the production infrastructure is a key capability that is necessary to determine in real time the current state of the infrastructure. Resources that are made available for usage are either known or must be discovered, and flexible access control protects resources that, for example, are not available for users that are part of a different infrastructure federation. As the state of the production infrastructure resources is inherently dynamic, changes in a resource's state (e.g. availability, or load) are propagated through messaging facilities and endpoints.

Similarly, to account for how much of the provided resources are used, the Accounting Capability requires identical subsequent capabilities to provide to the operations community vital information for budget calculations and, eventually, billing facilities.

A.3.1 Monitoring Capability

A production infrastructure is primarily defined by its availability, reliability and security – if its user community cannot rely on it then it is not an infrastructure. All of the resources within the infrastructure need to be monitored for the community to be assured of the quality. Such a monitoring capability is essential for the operational staff attempting to deliver the production infrastructure and the end-users seeking out reliable resources to support their research.

It is important in a service-oriented environment to distinguish between the services that are monitored, and the service that provides monitoring. In EGI *all* production infrastructure related services (e.g. a compute job submission service) must be monitored for a defined set of parameters (see below).

The service that provides the monitoring capability itself is delivered by EGI through the Service Availability Monitor framework (SAM).

A.3.1.1 Supported Interfaces

The EGI SAM framework provides a monitoring infrastructure that is based on three key cornerstones relevant for this Capability: Nagios and related service probes as the functional monitoring data, the MyEGI portal to visualise and report the monitoring results, and a messing infrastructure (shared with the EGI Accounting infrastructure) based on Apache MQ that ties the monitoring data extraction tool with the MyEGI data presentation layer.

This architecture defines two interfaces, which reside on the monitoring data extraction layer and the monitoring data presentation/querying layer, respectively.

The interface on the extraction layer is rather an integration point in that service-specific probes are provided that plug into the Nagios service-monitoring tool.

On the monitoring data presentation/query layer, both a programmatic (API) and web based access interface are necessary, but currently not available.

A.3.2 Accounting Capability

The use of resources within the e-Infrastructure must be recorded for a number of reasons. From statistical analysis of usage patterns, prediction of resource shortage up billing of the actual use of resources are just some common use cases for the usefulness of accounting data.

Structurally, the Accounting Capability is very similar to the Monitoring Capability in its deployment and ownership of the various components: The Accounting Capability is realised by EGI through the APEL accounting repository and accounting portal. The data that is stored in the repository and visualised through the portal is supplied by the functional services directly, or through functionally orthogonal modules that parse service log files and translate the found information into accounting information.

The accounting information extraction and the central accounting data repository are linked together by an EGI-wide network of messaging brokers.

A.3.2.1 Supported Interfaces

OGF defines a record format for accounting data [R 36], Usage Record (UR). It also suggests a draft of an access interface and a format for aggregated accounting records through the Resource Usage Service. Convergence to the UR standard exists only insofar as it is used together with various incompatible extensions for VO support. Based on EMI's submission of their StAR extension to UR, the OGF UR WG is now standardising accounting records for storage services.

A.4 STORAGE CAPABILITIES

Storage capabilities are necessary for any kind of long(er) term availability of data, whether raw (e.g. taken directly from an instrument), or digested in any kind of form or shape.

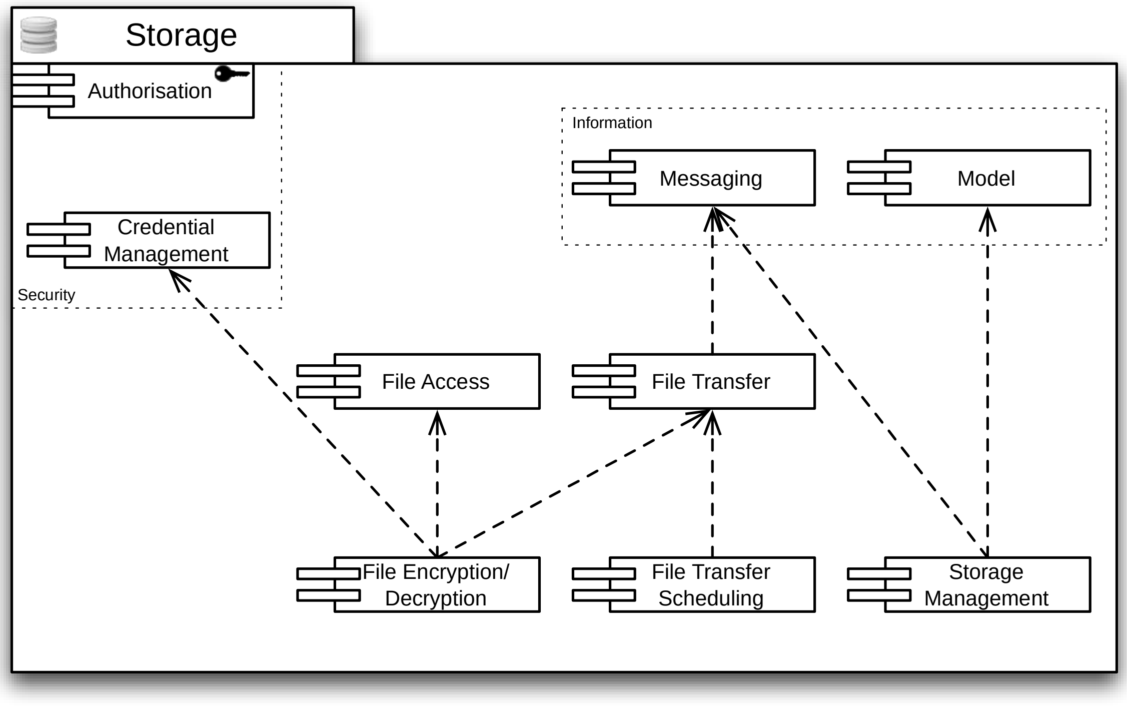


Figure 13: Dependencies of the Storage Capabilities

For any Storage Capabilities, Authorisation is necessary to allow access control to the provided resources and services. Hence instead of populating the diagram with a plethora of arrows that disguise the important dependencies in this area, this general prerequisite is symbolised by attaching the Authorisation Capability to the outer boundary of the Storage Capabilities.

Managing Storage resources is an important task in day-to-day Grid business. Storage resources may have to be taken offline, into maintenance, or newly created resources must be set up, and integrated into the infrastructure. Once the management tasks are completed, the changes must be propagated through the infrastructure using the deployed messaging capabilities. Indirectly, this requires access to the Discovery services available in the infrastructure to choose the correct messaging pipelines.

Files on the storage infrastructure are accessed every day in business. Some of them may be encrypted, and need to be decrypted in a seamless way. Credential Management is necessary to delegate the decryption (or encryption) process from the user to the File Access, or the File Transfer service, respectively. At the same time, file transfers consume resources that need to be properly accounted. Hence the resource consumption is properly forwarded to the accounting service through the messaging infrastructure.

A.4.1 File Encryption/Decryption Capability

Sensitive data needs to be stored securely. Before being stored in a remote file store the file may need to be encrypted and then on retrieval de-encrypted before use. The capability should also provide solutions relating to the storage of the keys needed to perform these tasks.

A.4.1.1 Supported Interfaces

There are no standardised interfaces for File Encryption/Decryption, whether seamless, transparent or integrated. The encryption and decryption steps are distinct tasks in a small workflow for a compute job. However, the key-handling interface will be described in future versions of the roadmap following input from the EGI Community.

A.4.2 File Access Capability

File Access provides an abstraction of a file resource that may be located remotely on a storage element anywhere in the Grid. The physical nature of the storage element or the storage means are unknown, whether disk array, tape, distributed file system, or else. Access to the file resource includes bulk read and bulk write, block read and write and perhaps striped block read and write.

A.4.2.1 Supported Interfaces

There exist many different standards that are appropriate for File Access implementations. Available standards fall into two categories that differ in their mode of access for the client:

- Interfaces that expose the remote nature of File Access
- Interfaces that integrate with local file system access (and hide the remote nature of the file resource).

The single most widespread interface for local file system based file access is POSIX (Portable Operating System Interface for Unix), defined by IEEE. Many different implementations map POSIX to specific, and proprietary, file systems, such as FAT, VFAT, ext2/3/4, ReiserFS, XFS, DFS, to name but a few. Popular protocols that map remote file resources into the local file system of any given system are CIFS (former SMB), NFS, or btrfs. While those protocols are mainly suited for permanent or semi-permanent access to remote files, FUSE allows elastic, transient and temporary access to remote files without root/Administrator involvement, bridging any suitable remote file access protocol into any user's home directory (or writable file system areas) for on demand access using common POSIX-compatible local file access methods.

Non-integrated, proprietary File Access interfaces are DCAP, RFIO, and XROOTD, which are all deployed to various degrees in EGI infrastructure. Protocols that were not originally designed for File access but are nonetheless suitable are HTTP(S) (even in parallel) and WebDAV.

A.4.3 File Transfer Capability

Files are stored at different physical locations within the production infrastructure and are frequently used at other locations. It is necessary for the files to be efficiently transferred over the international wide area networks linking the different resource centres.

Typically, a dedicated service provides the File Transfer capability, offered through potentially many instances of the same service software for scalability reasons.

A.4.3.1 Supported Interfaces

The OFG GridFTP protocol is used extensively in production infrastructures around the world alongside protocols such as http/https that have been developed outside of this community. This protocol provides the functionality to read/write and list data files stored on remote locations.

A.4.4 File Transfer Scheduling Capability

The bandwidth linking resource sites is a resource that needs to be managed in the same way compute resources at a site are accessed through a job scheduler. By being able to schedule wide area data transfers, requests can be prioritised and managed. This would include the capability to monitor and restart transfers as required.

A.4.4.1 Supported Interfaces

The only known standardised interface that allows File Transfer Scheduling is the Data Movement Interface (DMI) developed in OGF. At least two implementations are known that fully implement the interfaces defined within DMI, but no production implementation has been reported.

A.4.5 Storage Management Capability

Storage Management refers to the ability of managing a storage resource, from simple hard disk-based systems to complex hierarchical systems.

Typical deployments are to bundle a management interface with the respective local storage element into one service, so that it provides both the File Access, and the Storage Management capability. Alternatively, a dedicated service implementing Storage Management is capable of managing many remote storage nodes.

A.4.5.1 Supported Interfaces

The most commonly used specification is SRM (Storage Resource Management) from the OGF. However, ambiguities in the interface definition and description need to be addressed before unified SRM based management of storage resources can be achieved. Moreover, different implementations, for example from IGE and EMI, vary in adoption of the standard, and a common subset or full adoption must be agreed upon, before interoperability between providers can be achieved.

A.5 DATA CAPABILITIES

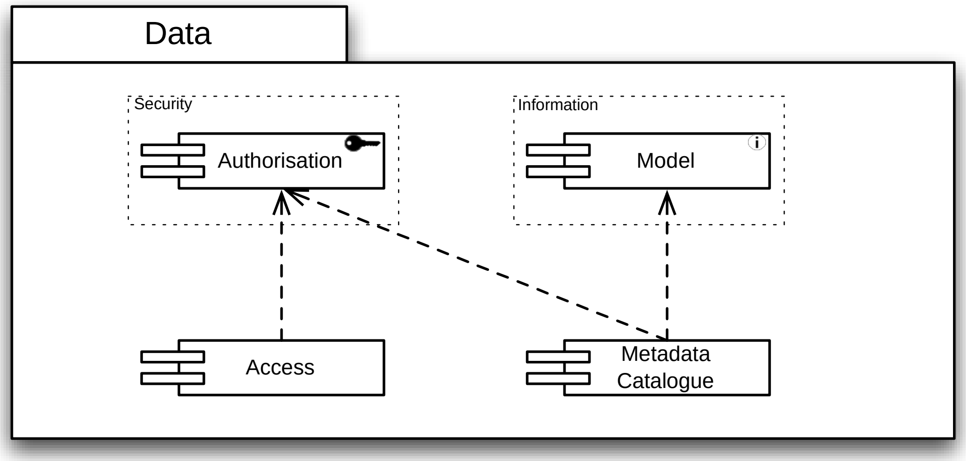


Figure 14: Dependencies of the Data Capabilities

Data Access becomes increasingly important in contemporary distributed computing infrastructures.

Fine-grained access control to distributed data sets is required to protect copyrighted material or data covered by non-open access licenses from unauthorised access.

The same fine-grained access control is necessary to protect searches and indexing activities for data catalogues.

A.5.1 Data Access Capability

Many communities are moving to the use of structured data stored in relational databases. These need to be accessible for controlled use by remote users as any other e-Infrastructure resource.

A.5.1.1 Supported Interfaces

The OGF family of standards developed in the DAIS WG provide standardised access to relational (WS-DAIR) and XML structured data (WS-DAIX)

A.5.2 Metadata Catalogue Capability

The metadata catalogue is used to store and query information relating to the data (files, databases, etc.) stored within the production infrastructure. An integral part of this functionality is not only to query about the existence of a file that may satisfy the needs of the enquiring user, but also the ability to resolve to a concrete description of the location of the file itself.

Typically, metadata catalogues are provided by dedicated service implementations and deployments for scalability reasons, and to provide metadata catalogues that feature different metadata sets for different user communities.

A.5.2.1 Supported Interfaces

To be described in detail in future versions of the roadmap following input from the EGI Community. Functionalities include the ability to store and query information relating to the data item including, location, mapping of persistent storage identifiers to the locations of the stored data.

At the moment, there are no standard interfaces known.

A.6 COMPUTE CAPABILITIES

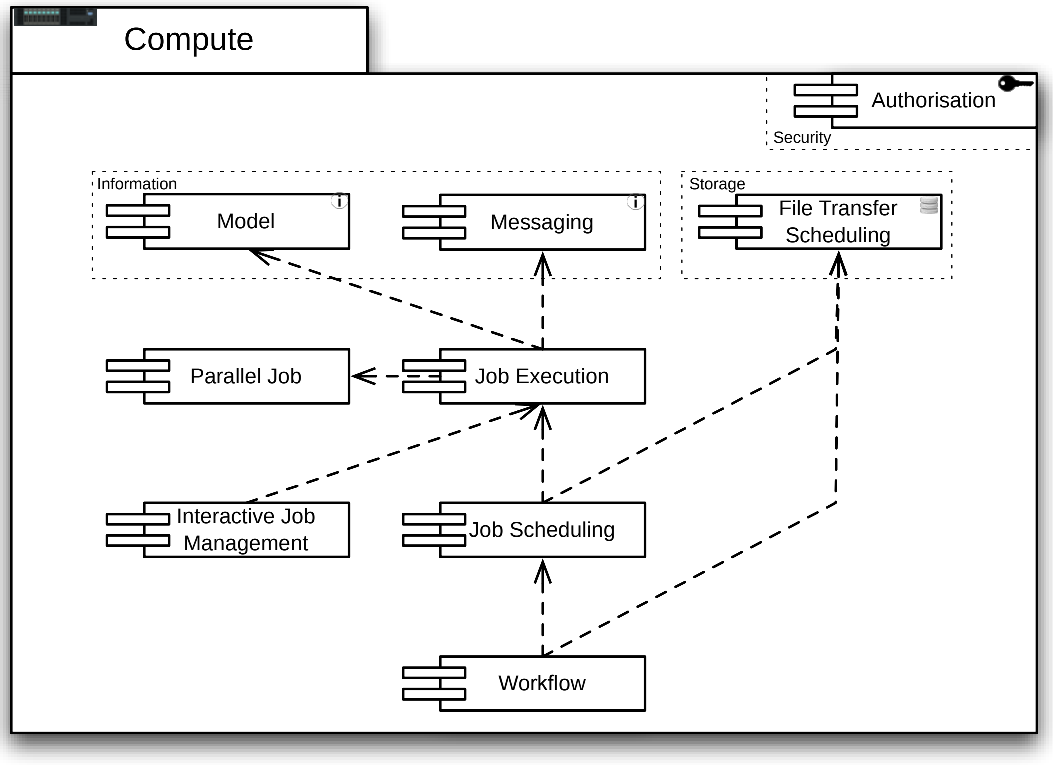


Figure 15: Dependencies of the Compute Capabilities

Capabilities providing access to distributed computing resources are the heart of most distributed computing infrastructures.

Just as Storage capabilities the Compute Capabilities require proper access control mechanisms delivered through generally available implementations of Authorisation.

The Job Execution Capability plays a central role in this group of capabilities. Through messaging usage records are provided for the accounting service, and using a common language for modelling ensures that no ambiguities exist in the execution and accounting of submitted compute jobs. To provide support for parallel job execution, appropriate libraries and job models are necessary. Compute Jobs need to be scheduled to accommodate user constraints such as resource usage, finishing time, and operational constraints such as average resource utilisation and load. Input files and output files need to be transferred from and to a specified storage location, forming already a simple, basic workflow for components to provide. Higher-level workflows include scheduling compute jobs and file transfers, among other domain specific tasks that are not covered in the UMD Roadmap.

A.6.1 Job Execution Capability

The Job Execution Capability relates to the ability to describe, submit, manage and monitor a work item on a specific site submitted for either queued batch or interactive execution.

A.6.1.1 Supported Interfaces

There are a number of different proprietary interfaces currently in production use that provide the ability to describe, submit and manage an interactive or batch work item on a specific site. Activity within the Open Grid Forum in recent years has led to specifications in this area: Job Submission Description Language (JSDL), the Basic Execution Service (BES), High Performance Computing Basic Profile (HPC-BP) and HPC File Staging Profile (HPC-FSP) specifications. These specifications and the experiences derived from them are forming the basis of ongoing activity within the OGF Production Grid Infrastructure Working Group. It is expected that the output from this activity will eventually lead to the interfaces that will be supported by EGI.

A.6.2 Parallel Job Capability

The parallel programming paradigm is gaining greater use in the user communities within EGI. The infrastructure does not provide support at the programming level – it is not needed – but provides support for controlling the distribution of processes to physical machines within a cluster. The ability to have fine-grained control over the placement of processes for an MPI or OpenMP application is a key differential between this capability and a conventional batch job capability.

A.6.2.1 Supported Interfaces

To support parallel jobs, three individual issues must be solved: Computing nodes must bear the correct parallel job library (both provider, and specific version), their ability to run parallel jobs using either of the provided libraries must be properly advertised in information systems (hence are discoverable and searchable), and the job submission must actually indicate the use of the parallel job capability.

Existing libraries for parallel job programming are:

1. MPI: Message Passing Interface 1.x
2. MPI: Message Passing Interface 2.x
3. OpenMP

Concerning the Information Model and Discovery, GLUE 2.0 provides the necessary element, i.e. class `ApplicationEnvironment`, property `ParallelSupport`.

Submitting parallel jobs is supported in the OGF SPMD Application extension to JSDL 1.0.

A.6.3 Interactive Job Management Capability

For certain use cases of distributed computing interactive access to the running job is necessary. A certain form of communication and control to the running job is required, for example to monitor the job progress or intermediate output in near-real time, or to stop, restart, or even interactively manipulate certain parameters of execution.

A.6.3.1 Supported Interfaces

There are no standardised interfaces to interactive job control known. The most common communication channels used are ssh access to the process environment for normal shell based access to program parameters and processes, or sockets that offer a limited, usually proprietary, command shell to the process itself.

A.6.4 Job Scheduling Capability

Compute Job Scheduling capability refers to the ‘end-to-end’ service that can be delivered to a user in response to their request for a job to be run. This includes managing the selection of the most appropriate resource that meets the user’s requirements, the transfer of any files required as input or produced as output between their source or destination storage location and the selected computational resource, and the management of any data transfer or execution failures within the infrastructure.

A.6.4.1 Supported Interfaces

No standard interfaces for Compute Job Scheduling are known. The OGF DCIFED Working Group [R 37] is chartered to address this gap, but neither interfaces nor their expected publication dates are known. Meanwhile some existing implementations of Compute Job Schedulers support simple scheduling capabilities described in OGF standards such as BES and JSDL.

The implementations provided for Compute Job Scheduling should use compatible interfaces for the batch compute capability.

A.6.5 Workflow Capability

The ability to define, initiate, manage and monitor a workflow is a key capability across many user communities. Workflows are by nature highly domain-specific even though a number of properties and features are shared between them. However, the various workflow systems may have requirements that need to be supported within EGI's core infrastructure.

A.6.5.1 Supported Interfaces

EGI does not have a stance on any workflow system, or any standardised access interfaces for workflow engines as long as the core infrastructure defined through the UMD Capabilities, provides sufficient support for domain specific workflow engines.

EGI considers workflows, especially those that go beyond simple DAG type of job trees, as a domain-specific feature. While needed across many different domains, convergence on a common set of workflow features and interfaces is seen as very unlikely in the future.

A.7 VIRTUALISATION CAPABILITIES

Virtualisation provides powerful opportunities for alternative approaches to the provisioning of distributed computing resources. Three main capabilities are required to successfully provision virtualised resources.

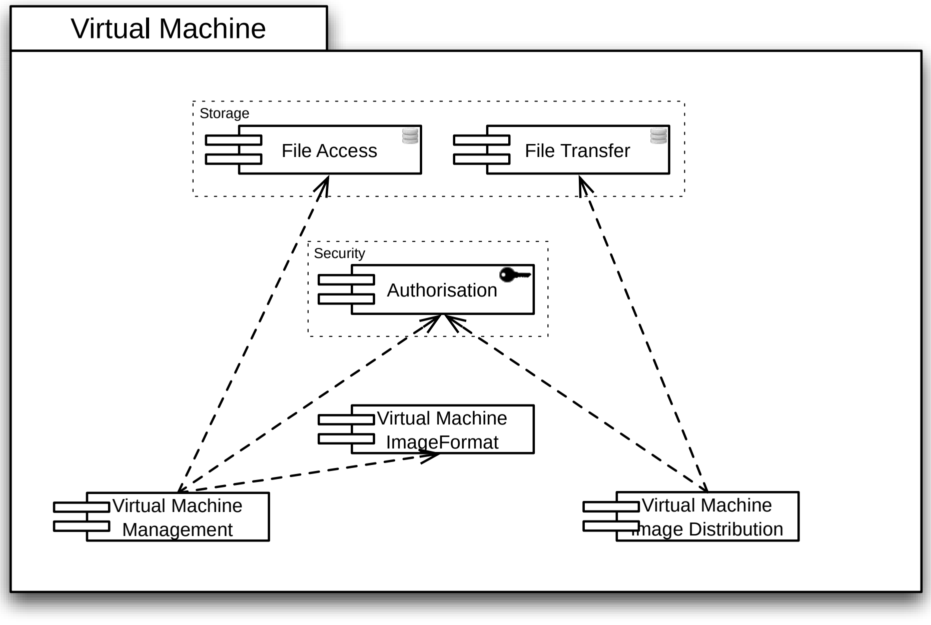

Virtual Machine
Storage
File Access
File Transfer
Security
Authorisation
Virtual Machine ImageFormat
Virtual Machine Management
Virtual Machine Image Distribution

Figure 16: Dependencies of the Virtualisation Capabilities

Central to interoperable Virtualisation is a commonly agreed image format for the virtual machines. The Image Format Capability is therefore of identical paramount importance to Virtualisation as the Information Model Capability for almost all other aspects of a distributed production infrastructure.

Managing virtual machine images requires proper access to the image repositories, which are in turn protected by access control.

Distributing image files requires proper authorisation (e.g. a distribution agent must be authorised to access an image stored in one location to transfer it to a different location to make it available for execution by a different user.) and file transfer facilities to allow the instantiation of the virtual machine in close proximity to the input and output data the contained appliances require.

A.7.1 Virtual Machine Management Capability

The core functionality is for authorized users to manage the virtual machine life cycle and configuration on a remote site (i.e. start, stop, pause, etc.) Machine images would be selected from a trusted repository at the site that would be configured according to site policy. Together this would allow site managers to determine both who could control the virtual machines running on their sites and who generated the images used on their site.

A.7.1.1 Supported Interfaces

The OGF OCCI WG [R 38] provides extensive management capabilities for many different kinds of distributed computing management functionality. The Core specification describes the foundation of all OCCI related renderings and extensions. The HTTP Rendering specification describes the RESTful rendering of OCCI-based management is rendered in HTTP communication with various OCCI based resources. Finally, the OCCI Infrastructure specification defines a standardised set of extensions, mix-ins and attributes for infrastructure resources, such as virtual machines (even storage) that allow for standards based Virtual Machine management functionality.

A.7.2 Virtual Machine Image Format Capability

This capability refers to the ability of portable binary formats for Virtual Machine images that may run on different hypervisors deployed in EGI.

A.7.2.1 Supported Interfaces

The OVF (Open Virtualisation Format) from DMTF [R 39] provides a standardised format for Virtual Machine Images that may be deployed on many different virtualisation platforms available on the EGI infrastructure. Recently published as an ANSI standard, widespread adoption is nearly guaranteed.

A.7.3 Image Distribution Capability

As virtual machine images become the default approach to providing the environment for both jobs and services, increased effort is needed on building the trust model around the distribution of images. Resource providers will need a mechanism for images to be distributed, cached and trusted for execution on their sites.

A.7.3.1 Supported Interfaces

There are no standardised interfaces available for federated distributed VM Image distribution.

A.8 INSTRUMENTATION CAPABILITIES

A.8.1 Remote Instrumentation Capability

Instruments are data sources frequently encountered within e-Infrastructures. As part of a distributed computing architecture providing remote access to manage and monitor these instruments is becoming increasingly important within some communities.

A.8.1.1 Supported Interfaces

There are no standardised interfaces known for the Remote Instrumentation Capability.

A.9 CLIENT CAPABILITIES

A.9.1 Client tools

Client tools aim to provide an access layer for end-users to efficiently use the Grid infrastructure in a way that integrates either seamlessly, or with minimal effort in the user's daily work to achieve his or her research. Client tools range from command line clients to more complex graphical user interfaces geared towards the use of the Grid resources directly or indirectly.

A.9.1.1 Supported Interfaces

There are no supported interfaces available or necessary for the Client tools Capability except that Client tools are *consumers* of other interfaces and APIs provided by the exposed Grid resources, and therefore must be carefully evaluated when changing interfaces at a lower level.

A.9.2 Client API Capability

Instead of addressing interface heterogeneity on the service level, an alternative approach proposes the abstraction of distributed services on the client side, providing a common interface to client application developers. Adopting a client API may protect domain specific application developers from evolving middleware by maintaining compatibility with a client side API for the most common Grid Use Cases rather than keeping track of and synchronising middleware interfaces.

A.9.2.1 Supported Interfaces

OGF provides the SAGA API specification as an approach to a common, lightweight and simple API for client-side abstraction of distributed computing resource. The SAGA API itself maps semantically very well onto interfaces that are standardised on a lower level of the middleware stack, such as GLUE, BES, DRMAA, GridFTP, and several others. In general, SAGA can be implemented on any DCI that provides (a subset of) SAGA semantic capabilities.