EGI-InSPIRE

UMD ROADMAP

EU DELIVERABLE: D5.2

Document identifier:	EGI-D5.2-272-v9.doc
Date:	**24/03/2011**
Activity:	**SA2**
Lead Partner:	**EGI.eu**
Document Status:	**FINAL**
Dissemination Level:	**PUBLIC**
Document Link:	https://documents.egi.eu/document/272

Abstract

The Unified Middleware Distribution (UMD) is the integrated set of software components that EGI makes available from technology providers within the EGI Community. These components are packaged to provide an integrated offering for deployment on the EGI production infrastructure. This first version of the UMD Roadmap focuses on the overall structure of the UMD Roadmap in order to solicit feedback from the EGI Community. It presents a framework for technology providers to identify which capabilities they wish to address, and for the user and operations communities in EGI to identify additional capabilities they may wish to see supported within the infrastructure. The process by which this feedback will be integrated into future versions of the UMD Roadmap is described.

I. COPYRIGHT NOTICE

II. DELIVERY SLIP

	Name	Partner/Activity	Date
From	Michel Drescher	EGI.eu	11/02/2011
Reviewed by	**Moderator:** Peter Solagna **Reviewers:** Andrea Cristofori Emmanouil Paisios Robert Lovas	EGI.eu INFN LRZ EDGI	 11/02/2010
Approved by	**AMB & PMB**		15/02/2010

III. DOCUMENT LOG

Issue	Date	Comment	Author/Partner
2	22/12/2010	Major update to the document.	Michel Drescher, EGI.eu
3	19/01/2011	Revised section 2, and dependencies	Michel Drescher, EGI.eu
4	26/01/2011	All chapters except references revised. IGE contributions still missing.	Michel Drescher, EGI.eu
5	28/01/2011	References & IGE contributions added.	Michel Drescher, EGI.eu
6	31/01/2011	Document ready for internal review.	Michel Drescher, EGI.eu
7	02/02/2011	Comments from internal review incorporated.	Michel Drescher, EGI.eu
8	11/02/2011	Comments from external review incorporated.	Michel Drescher, EGI.eu
9	15/02/2011	Addressed one final comment from one reviewer	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 document specific glossary is provided in section 15.

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

This second version of the Unified Middleware Distribution (UMD) Roadmap iterates on the framework of Capabilities introduced in the first version of the UMD Roadmap. Taking feedback and recent development into account, the Capabilities have been refined and re-grouped according to the common area of function. This version of the UMD Roadmap introduces the concept of functional dependencies between Capabilities. In a distributed environment certain Capabilities – that is, their implementation as a service or common domain language – must be present for implementations of other Capabilities to function properly. Those functional dependencies illustrate the overall composition of the EGI production infrastructure and allow Technology Providers to structure and plan their software contributions aligned to the Capabilities without disrupting the operation of the infrastructure as a whole.

The UMD Roadmap will be formally reviewed and re-issued every six months during the course of the project, but will also exist as a 'living' document in between these releases to allow a continual process of community comment and feedback. The process by which input from the EGI Community is collected prioritised and integrated into the UMD Roadmap through the Technical Coordination Board (TCB) is briefly described in the document.

TABLE OF CONTENTS

1 INTRODUCTION

1.1 *Purpose*

This document describes the framework that will be used to build the Unified Middleware Distribution (UMD) Roadmap and the process by which contributions from the community will be integrated into future versions of the UMD Roadmap. The UMD Roadmap therefore describes the structure of the UMD and the process by which its functional capabilities will evolve over time in response to feedback from the end-user and operations communities. It is a document that will be formally reviewed and re-issued every six months during the course of the project, but will exist as a 'living' document in between these times to allow a continual process of community comment and feedback.

1.2 *Structure and composition of the roadmap*

To ease the adoption, i.e. provision of components that implement a particular functionality of the Grid, deployment and maintenance of the production infrastructure the UMD Roadmap identifies Capabilities that must be present, either all together or in various combinations, in order to satisfy the users' requirements.

Capabilities are scoped around well known and well-defined activities that are sufficiently disparate, independent, and fulfil a discrete task in a higher-level orchestrated sequence of actions taken by the user (manually or automated) in order to reach the aspired goal.

The boundaries of capabilities are described and defined by clear interfaces, and wherever possible, publicly accepted and developed standards are introduced. For example, the grouping of OGSA-BES, JSDL and HPC-BasicProfile standards provides a sufficient boundary description of the Compute Job Capability described further below. An aspired evolution of the implementations and uptake of standards is given in the final section for each Capability.

As the number of identified Capabilities grows, more structure and overview about different families of related capabilities is necessary. Hence the capabilities are classified into the following areas:

1) Information
2) Compute
3) Storage
4) Data
5) Instrumentation
6) Security
7) Virtualisation
8) Operations
9) Clients

Chapter two describes on a high level the processes that drive the evolution of the UMD Roadmap and, through it, the technology used in the EGI production infrastructure. The relationships between the communities, EGI.eu, and the Technology Providers are highlighted, and the key sources and scope for providing the production infrastructure are illustrated.

The chapters three to 11 describe the UMD Capabilities aligned with the indicated classification.

Chapter 12 describes the EGI Platform as a while, followed by conclusions given in chapter 13.

Chapter 14 provides references for further reading, and the document is ended with a glossary of terms used in this document.

2 TECHNOLOGY WITHIN EGI

EGI provides a production quality e-Infrastructure to researchers across Europe that interoperates with other e-Infrastructures worldwide. Those two key features, production quality and interoperation, stem from the predominant process of globalisation of collaborating user communities and increased networking across user communities.

To ensure collaboration, interoperation and at the same time production quality of the provided e-Infrastructure, EGI adopts a set of important concepts and processes, which are further described in greater detail in the following sections.

The first edition of the UMD Roadmap [R 34] provided an initial overview of software categories and capabilities, which are envisioned to exist or are already provided in EGI's production infrastructure. The continuous adaptation and evolution of demand and supply is captured in the existing and future editions of the UMD Roadmap.

2.1 Technology Landscape

To satisfy the needs of end-users and operators, the software that is installed on the resources will have to be drawn from many different sources. Several techniques and approaches are employed to manage the complexity and difficulty of satisfying the requirements of many different user communities.

The software selected to comprise the complete Technology Landscape within EGI is sourced from three different provider communities, complemented with different levels of support, visibility and influence on the future evolution of the Technology deployed within EGI.

2.1.1 Categories of software sources

The foundation of the software infrastructure is comprised of general-purpose software (not developed within EGI as the primary use case) needed to satisfy the higher-level distributed computing infrastructure. For example, Operating Systems such as Scientific Linux, Fedora or Red Hat Enterprise Linux, RDBMS systems such as MySQL, PostgreSQL or Ingres are examples of general-purpose software that may be incorporated into the software stack used by EGI. This type of software will be taken and used from public sources such as SourceForge for open source software, or any other dedicated source such as corporate download servers for commercial software products, if applicable. EGI's support of and influence on the software drawn from these sources is very limited if not existent, and EGI typically incorporates that type of software, as required.

On top of a fabric of general-purpose software, a well-defined set of components is installed that are clearly driven by use cases for DCIs. Those use cases are common across the different user communities that EGI engages with. The software that falls into this category is typically called the UMD and includes components that provide job submission facilities, or the scheduling of file transfers.

Finally, each user community has a set of very domain specific requirements on software. Those requirements cannot be satisfied by either general-purpose software or middleware as such. Specific software that satisfies those needs is often developed and maintained within the user communities, or, where applicable, general-purpose software is extended or integrated in a domain-specific way to solve the pertinent user community's problems.

2.1.2 EGI Technology Roadmap

The EGI Technology Roadmap describes the software that is taken from all three software sources described above. Though not explicitly described in one document, references, dependencies, influence and impact between the components of all three sources are taken into account in the EGI Technology Roadmap.

Because of the complexity and importance for the EGI production infrastructure today, the software components that comprise the middleware of the EGI software landscape are described in a document of their own, the UMD Roadmap – this document.

The components within both the UMD and the EGI Technology Roadmaps are described in terms of their *capabilities* (i.e. consisting of specific *functionality* aspects that will be delivered in response to particular user communities over time) delivered through the production of *software releases* from a set of *technology providers*. These technology providers may deliver open-source software from within or external to the EGI Community (as opposed to open source software developed within the EGI-InSPIRE project) or relevant commercial software solutions. This allows the EGI to be technology neutral and to respond to the needs of its users – both the researchers using the infrastructure and the operations staff providing it – by selecting the best available technology for delivering the capabilities that are required.

The EGI Technology Roadmap will over time aim to:

1. Define technologies that will help EGI resource providers to reliably deliver services to more communities with the same manpower or less effort than they do today.
2. Define stable platform interfaces for each Capability to produce a platform that allows the user and developer communities to add value through their own software into the EGI environment.
3. Define a set of software components from within (i.e. the UMD Roadmap) and external to the EGI Community drawn from open-source or commercial technology providers that are able to contribute to delivering the production infrastructure, some of which may be made available through the EGI Software Repository
4. Identify a programme of support, maintenance and development for the software components that form the core functionality within the production infrastructure built on top of the base operating systems.

Through these aspects, the EGI Technology Roadmap defines the technical architecture of the software deployed on the European Grid Infrastructure and the environment in which the Unified Middleware Distribution (UMD) has to operate.

The EGI Technology Roadmap will be issued alongside the UMD Roadmap by the TCB.

2.1.3 UMD Roadmap

UMD is the set of software components that the EGI Community needs to provide to bridge the gap between the functionality it needs to deliver as a generic base to its user community and the ‘out of the box’ solutions available from mainstream commercial or open-source providers. The UMD Roadmap describes the capabilities of the software within UMD and how the functionality within each capability will evolve over time in response to the requirements coming in from the community. The current software used to deliver the production infrastructure to end-users can be grouped into three areas:

1. Components that provide functionality required by a particular community and are best supported by teams within that community. They are deployed into the environment provided by the EGI Technology Roadmap.
2. Components from outside the EGI Community described in the EGI Technology Roadmap. These may be components that have emerged from within the EGI Community and have now reached sufficient maturity, stability and functionality such that they have use beyond the e-infrastructure community. One route towards broader adoption is to contribute these components to mainstream software distributions and to use their associated communities as a way to expand the support structures. These components are then imported back into the EGI Community through the EGI Technology Roadmap defining these software distributions as a basis for the infrastructure. The EGI Technology Roadmap could also include commercially provided/supported software.
3. Components that are still being developed or deliver functionality that is only needed within the e-Infrastructure community as described in the UMD Roadmap.

Over time it is expected that components will move between these three categories as they mature and their uptake and usage changes. Components in the second group, i.e. components contributed to mainstream open-source software distributions, will help reduce the bespoke integration needed within EGI and promote the uptake of these components by a wider (non-Grid) community. Such activity contributes greatly to the sustainability of these components as they are exposed to a much wider user base than just a deployment within UMD. The availability of such capabilities within these distributions and their expected functionality will be described in the EGI Technology Roadmap. As the user communities mature, many of these capabilities are expected to be satisfied by the offerings provided by the services within EGI. Those communities requiring additional specialised capabilities beyond those described in the UMD Roadmap and EGI Technology Roadmap will need to demonstrate that their capability has applicability to other user communities, or to develop and deploy these capabilities themselves.

The UMD Roadmap describes the technical architecture of the UMD and documents how it is expected to change over time – the components that will be added or removed and the functionality that will appear or become deprecated. The roadmap will effectively be a living document – requirements will change or be refined over time and the technology providers will adapt their release schedules. Formally, the changes in the UMD Roadmap will be reviewed every six months with changes being made available between these formal reviews to solicit community feedback.

2.1.4 Concurrent implementations

It is possible for the TCB to endorse more than one component that answers the need for a particular capability in the UMD Roadmap. This allows different technical approaches, delivering the same capability to be explored and different implementations tuned to the different deployment environments that need to be provided to satisfy the EGI community's diverse needs. The ability for the TCB to endorse multiple implementations of the same capability is dependent on the definition and adoption of common interfaces by the different technology providers. This is an essential pre-condition in order that end-user communities are able to transparently use one of the deployed implementations without having to adapt their client code or workflow.

2.2 Technology Evolution: Process and Governance

EGI has to provide a technical infrastructure that needs to satisfy many stakeholders. Its most important class of stakeholder are the end-user communities that span many diverse disciplines, physically distributed across the European Research Area. These communities have many different approaches to using the infrastructure and different technologies are needed for federating the distributed resources provided by the collaborating resource providers in order to support their end-user communities. The resource providers are generally federated into national or domain specific groups (i.e. NGIs or EIROs) and bring together the individual sites from a resource provider into an integrated operational and management structure.

Together, the User Communities and the Operations Community have requirements that must be met by the production infrastructure provided by EGI, particularly the software deployed on the infrastructure resources. Established through policies and procedures, those communities formulate, discuss, prioritise and communicate their requirements to EGI. Technology Providers then step up and deliver implementations of those requirements that are deployed on the resource infrastructure to satisfy the requirements of the users. As needs and requirements over time shift into different directions, the software deployed on the infrastructure needs constant and regular reassessment and update in order to align again with the expressed needs of the user communities, leading to a virtuous cycle of constant evolution of the delivered technology.

EGI's process of requirements driven technology evolution is described in great detail in two formal documents published by the User Community Support Team in MS305 [R 30], and the Operations Management Board in Deliverable D4.1 [R 35]. Both documents describe the underlying processes to gather, filter and prioritise the respective community requirements for the production infrastructure. Regularly the prioritised requirements are presented to the Technology Coordination Board to determine a final prioritisation for implementation and uptake (for new capabilities and requirements).

2.3 Collaboration with new Technology Providers

Technology providers wishing to have their components considered for inclusion into the UMD Roadmap should email cto@egi.eu. They should indicate:

1) The capability in the UMD Roadmap (identified in sections three and onward of this and future versions of this document) addressed by the component.
2) The current user community for the software.
3) The current support structure for the software.
4) The desired level of support (integrated/contributed/candidate/offered).
5) The license under which the software is provided to EGI.

At regular intervals the TCB will consider such requests drawing on the expertise and knowledge within the TCB, the DMSU, and the broader EGI Community as to the suitability of the component. If adopted into the UMD Roadmap usage of the component will be monitored and the support level revised as appropriate.

All Technology Providers wishing to achieve integrated, contributed or candidate status will be expected to enter into a Memorandum of Understanding (or an equivalent contractual commitment for a commercial provider) with EGI.eu as to the support commitment they are willing to provide. Those technology providers wishing to achieve integrated status will be expected to enter into a Service Level Agreement (SLA) or equivalent with EGI.eu to provide support for their components and be integrated into EGI.eu's support structure.

3 SECURITY CAPABILITIES

Security capabilities form an important foundation of 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.

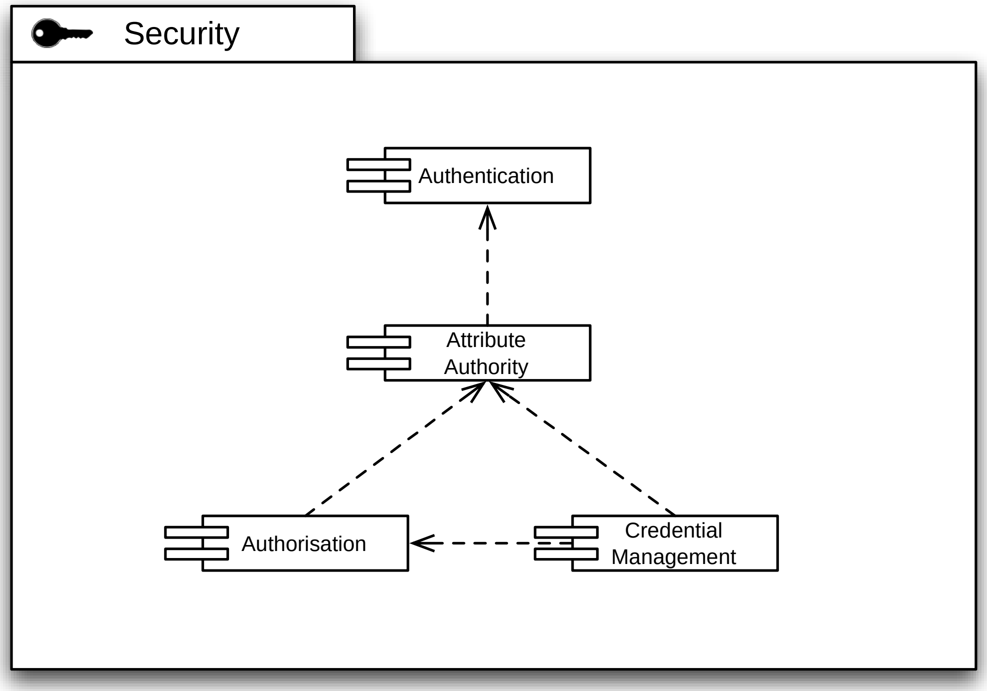

Figure 1: Dependencies of Security Capabilities.

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.

3.1 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.

3.1.1 Supported Interfaces

The primary authentication token within the infrastructure is the X.509 certificate and its proxy derivatives. The certificates and any proxy schemes must follow specifications that are fully integrated into the https protocol (as opposed to the httpg protocol). Wherever possible the user should not be exposed to certificates and their associated handling.

An alternative, standards based Authentication interface is SAML 2.0 [R 6] allowing for federated authentication solutions geared towards the end-user of the Grid.

OpenID [R 7] is a community driven family of specifications that provides federated authentication mechanisms.

Regardless the infrastructure and access interfaces to authentication solutions, a common language between services that provide security facilities and those services and clients who consume them is necessary. Two popular languages for security attributes are eduPerson/eduOrg [R 41] and eduGAIN [R 42]. While eduPerson is mostly used in the US related Grid DCI communities, eduGAIN is a fast-growing service for federated identity provisioning. In this context, eduGAIN's Metadata policy [R 43] is of particular interest as it defines a language for modelling security attributes for use in SAML/XACML combinations for both Authentication and Authorisation.

3.1.2 Implementation Roadmap

Most authentication mechanisms used in DCIs these days rely on X.509 certificates to authenticate end-users. Components are readily available from EMI and IGE in that domain, for example SLCS [R 8] and STS [R 8].

However clear user demand requires a shift towards flexible, and lightweight federated authentication mechanisms based on SAML 2.0. Solutions are available, but need adaptation towards the needs of the Grid community. The TERENA project offers a migration path from X.509 based user authentication towards federated, SAML based user authentication (and authorization) bridging to X.509 based proxy authentication for service invocation.

Both IGE and the Globus Alliance in the US share the common goal of replacing httpg with https in GT5, and through these efforts IGE aims to deliver this capability in 2012.

EMI has considered eduGAIN as a language for modelling security attributes; and eduPerson as an alternative was not communicated to EMI as a requirement for modelling security attributes. Therefore, EMI is currently developing a model for an EMI-wide common profile for security attributes [R 44].

3.2 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.

3.2.1 Supported Interfaces

An interface is needed to issue proxy certificates relating to the roles and groups that an individual has within a VO. The corresponding attributes may also be delivered back to the client through SAML assertions.

3.2.2 Implementation Roadmap

Contributions are expected from EMI and IGE.

IGE will provide initial support for SAML-capable VOMS within GT5.2 in the IGE v2.0[1] release in early 2012.

3.3 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.

3.3.1 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 [R 31], which provides for authentication and authorization for delegated access to specific data. Facebook, amongst others, is a main driver of OAuth.

3.3.2 Implementation Roadmap

EMI delivers a varying set of Authorisation components today. With the delivery of EMI v2 in April 2012 a common authorisation framework, Argus, provides for this capability [R 8].

In IGE v2.0, due in early 2012, IGE will provide direct support for interfacing with the LCAS/LCMAPS account mapping mechanism in GT5.2, including the capability to talk to SAML and XACML capable authorization services including EMI Argus.

3.4 Credential Management Capability

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

[1] IGE v2.0 is currently a placeholder for a set of functionality not yet determined.

3.4.1 Supported Interfaces

One of the key functionalities in this area is the linking of institutional authentication systems (for example, Shibboleth [R 9], OpenID, etc.) to the transparent issuing of certificates for use in the infrastructure through identity federations. This should be provided for operational deployment through the use of web portals and web service interfaces.

CILogon also provides a bridge between SAML (Shibboleth) and X.509. CILogon is comprised of MyProxy and the GridShib CA.

The Globus Alliance is pushing towards MyProxy CA + OAuth to allow web-based sign-in to a (3rd party) identity provider with retrieval of a proxy cert. This enables a user to authenticate to a MyProxy server without exposing passwords to the intermediary web site, yet delegate the proxy cert from the MyProxy server to that intermediary web site.

However, there are no standardised interfaces available in this area.

3.4.2 Implementation Roadmap

See for example the work on Short-Lived Certificate Services (SLCS) [R 8] and the Terena Certificate Service (TCS) [R 10].

Globus already supports the MyProxy proxy certificate storage, retrieval and delegation service, which will be provided in IGE v1.0 (IGE-packaged GT 5.0.3) in April 2011. IGE plans to deliver support for the AdHoc GUI front-end to VOMS [R 36] for managing VOs in 2012. In addition, the Globus Alliance plans to add support for MyProxy + OAuth to Globus Online, although no fixed dates are available yet. Other mechanisms for the issuing of certificates based on institutional authentication credentials are currently being investigated, including the approach offered by CILogon.

4 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.

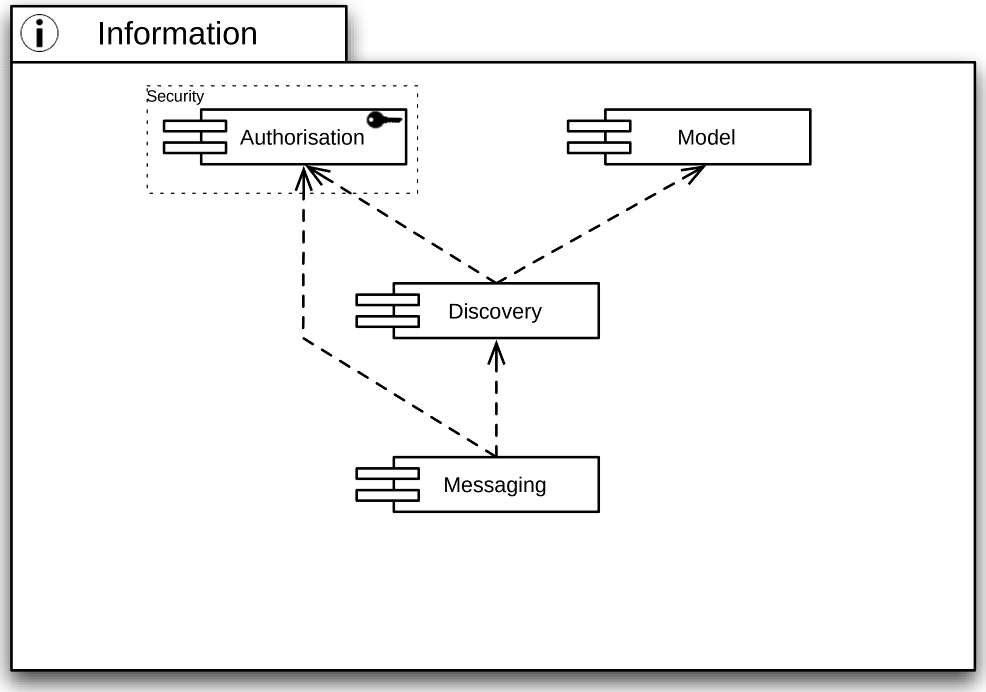

Figure 2: Dependencies of Information Capabilities

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.

4.1 *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.

4.1.1 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 [R5].

4.1.2 Implementation Roadmap

EMI will start to support GLUE2 using a subset in its components of the Compute and Data area services, beginning in April 2011 with the delivery of EMI1.0. By April 2012, when delivering EMI2.0 EMI will fully support GLUE 2.0.

Globus' Integrated Information Service (IIS) [R 37], as a generalized information service, will support GLUE2 (as well as other schemas). Within the Globus Alliance, proof-of-concept support for GLUE2 within IIS already exists, and the selection and integration of an existing GLUE2 implementation into IIS will be started in early 2011. IGE will track these developments and include them in future development cycles as appropriate.

4.2 *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.

4.2.1 Supported Interfaces

The Java Message Service (JMS) [R 38] 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 [R 11].

Emerging from the financial industry, *AMQP* [R 12] 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.

4.2.2 Implementation Roadmap

JMS is widely used today in Grid Middleware and elsewhere, with the most popular implementation perhaps being ActiveMQ.

AMQP is just available as version 1.0 awaiting production implementations. AMQP 1.0 will be voted for final status once more than two production implementations are available [R 47]. Meanwhile, stable production implementations for versions 0.9.1 are available, such as RabbitMQ [R 13], AMQPI (AMQP Infrastructure) for Fedora 10 (by the Fedora Project) [R 14] or QPID by Apache Project [R 15].

The Globus Alliance is currently investigating support for AMQP, and has conducted some prototyping in this area. IGE will track these developments with the intention to support AMQP.

4.3 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.

4.3.1 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.

4.3.2 Implementation Roadmap

EMI provides BDII, a well-known Information providing service offering LDAPv3 access, from its first release EMI 1.0, due in April 2011.

Within the Globus Alliance, the next generation of IIS is currently being investigated. IGE will track developments in this area for later consideration.

5 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.

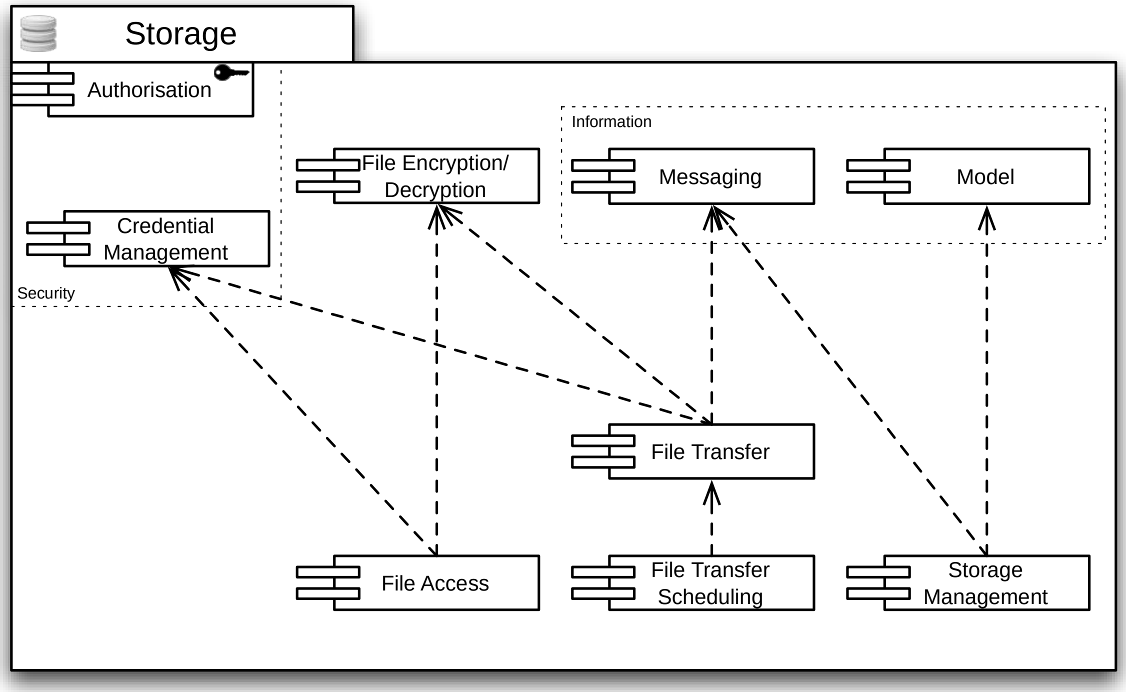

Figure 3: Dependencies of Storage Capabilities

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.

5.1 *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.

5.1.1 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.

5.1.2 Implementation Roadmap

EMI delivers from the start the Hydra [R 8] service implementing a proprietary key-handling interface for file encryption and decryption tasks.

Information about an implementation from IGE is currently not available.

5.2 *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.

5.2.1 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.

5.2.2 Implementation Roadmap

EMI will deliver POSIX access to remote files via StoRM from the start. By April 2011 (i.e. EMI 1.0) DCache will provide a full POSIX interface, while DPM will only allow prototypical POSIX access by then. With the release of EMI 2.0 in April 2012 all EMI implementations will provide full POSIX access to remote files. [all R 8]

From a Globus perspective, implementations providing POSIX based file access already exist to provide abstract virtual file system-oriented access to grid-based file resources, such as Grifi [R 32] and Parrot [R 33]. IGE is currently investigating these options.

5.3 *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.

5.3.1 Supported Interfaces

The GridFTP protocol [R2] 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.

5.3.2 Implementation Roadmap

There are many implementations supporting various file transfer protocols, of which FTS [R 8] from EMI will be delivered from the start.

From an IGE perspective, Globus already provides GridFTP as a core service, which will be provided in IGE v1.0 (IGE-packaged GT 5.0.3) in April 2011.

5.4 *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.

5.4.1 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.

Industry standard solutions that are commonly used as integration points for file transfer scheduling are FTS [R 17] and RFT [R 16] provided by EMI and IGE, respectively.

5.4.2 Implementation Roadmap

Commonly known implementation of the File Transfer Scheduling Capability are RFT, and FTS.

FTS will be delivered by EMI right from the start, combining the Capabilities File Transfer, and File Transfer Scheduling, into one Product. However, RFT is deprecated in GT5 in favour of the relatively new approach GlobusOnline (www.globusonline.org), a cloud based file storage and transfer solution provided by the Globus Alliance. IGE will provide a File Scheduling capability through a European instance of Globus Online in early 2012.

5.5 *Storage Management Capability*

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

5.5.1 Supported Interfaces

The most commonly used specification is SRM (Storage Resource Management) from the OGF [R 18]. 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.

5.5.2 Implementation Roadmap

EMI will deliver implementations of SRM early on with varying level of SRM support, beginning with its first release EMI 1.0 in April 2011. It will include, but not limited to, DPM and StoRM (from the gLite community), UAS-D (from UNICORE), and dCache from the dCache project itself. By April 2012 all delivered Storage Management implementations will provide a unified SRM interpretation and implementation. By April 2013 client-side access to Storage Management will be unified towards a common set of libraries that will support SRM.

IGE currently has no plans in this area.

6 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.

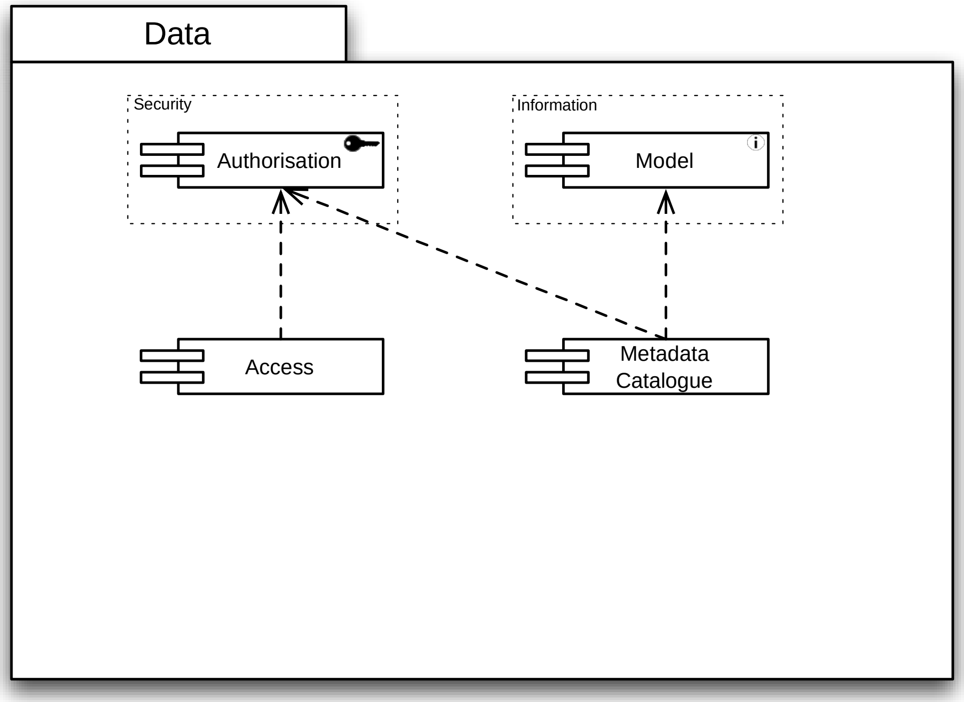

Figure 4: Dependencies of Data Capabilities

6.1 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.

6.1.1 Supported Interfaces

The OGF family of standards developed in the DAIS WG provide standardised access to relational (WS-DAIR) [R 19] and XML structured data (WS-DAIX) [R 20].

6.1.2 Implementation Roadmap

GREiC [R 46] provides proprietary means to manage efficiently, securely and transparently collections of bioinformatics data for in silico experiments.

OGSA-DAI provides a reference implementation of the standard WS-DAIX interfaces [R 21], which will be supported within GT5.2 in the IGE v2.0 release in early 2012.

6.2 *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.

6.2.1 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.

6.2.2 Implementation Roadmap

EMI's complementary services AMGA and LFC together provide the described functionality from the start with its AMGA component [R 8]. At the same time, LFC and Amga serve as integration points for higher-level services.

7 COMPUTE CAPABILITIES

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

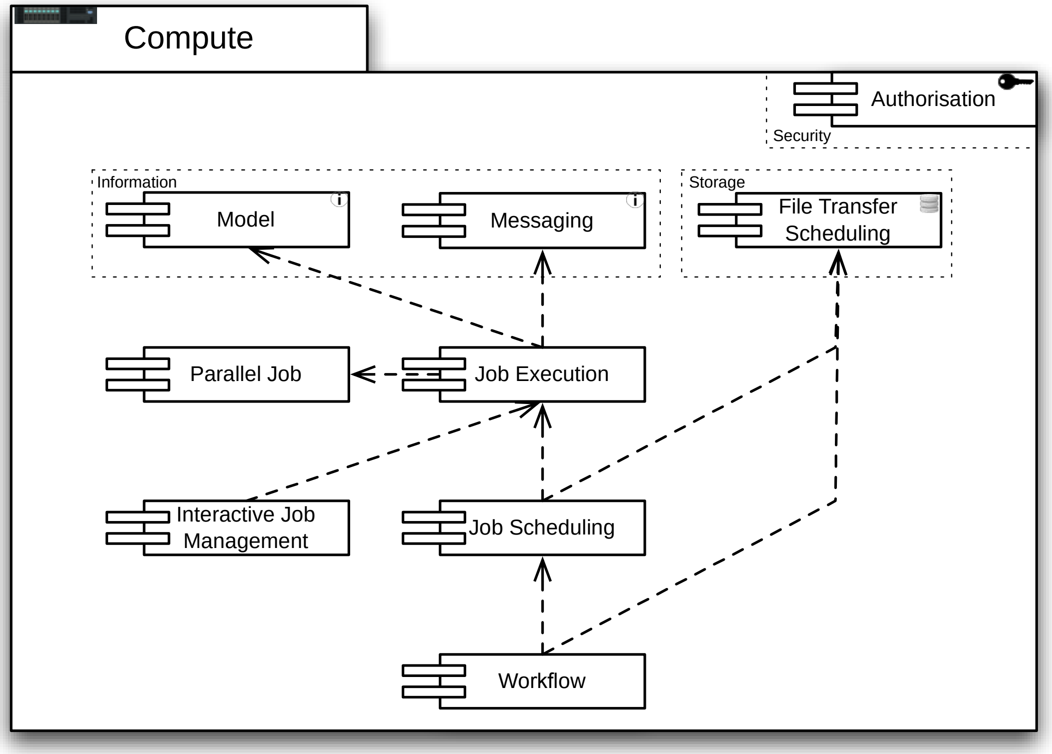

Figure 5: Dependencies for Compute Capabilities

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 specified storage locations, 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.

7.1 Job Execution Capability

The compute 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.

7.1.1 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.

7.1.2 Implementation Roadmap

Starting in April 2011 EMI provides varying support for the mentioned standardised interfaces and protocols, even though this is communicated as temporary. By October 2011, EMI will provide at least one compute service implementation that implements the agreed set of interfaces and messages defined by the OGF PGI Working Group [R 8].

IGE will deliver, in IGE v2.0 in early 2012, support for OGSA-BES, JSDL and the HPC Basic Profile within GT5.2.

7.2 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.

7.2.1 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 [R6]
2. MPI: Message Passing Interface 2.x [R7]
3. OpenMP [R8]

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 standard HPC-SPMD, which is an extension to the JSDL 1.0 standard.

7.2.2 Implementation Roadmap

EMI provides a proprietary and non-standardised way to run parallel jobs. By September 2012, projected to be available in release EMI2.x, EMI will support a common, standards based parallel job execution framework.

From an IGE perspective, Globus already provides MPI support in GRAM5, to be provided in IGE v1.0 (IGE-packaged GT 5.0.3) in April 2011

7.3 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.

7.3.1 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.

7.3.2 Implementation Roadmap

EMI will deliver basic, limited interactive access to compute jobs by September 2011. Beyond that no guarantees are made towards any evolution of this capability.

Globus already provides near real-time job interaction through the streaming of results or globus-job-get-output, and also through the querying of a job's status. Furthermore, Globus offers gsissh, an ssh based on X.509 certificates, to log in interactively on the remote site. These capabilities will be provided in IGE v1.0 (IGE-packaged GT 5.0.3) in April 2011.

7.4 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.

7.4.1 Supported Interfaces

No standard interfaces for Compute Job Scheduling are known. The OGF DCIFED Working Group [R 22] 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.

7.4.2 Implementation Roadmap

Mainly EMI and IGE are expected to provide implementations from early on, however incompatible, proprietary, or both.

EMI will deliver a Compute Job Scheduling implementation with its first major release EMI 1.0 in April 2011.

IGE will initially deliver Compute Job Scheduling via Gridway within the IGE v1.0 (IGE-packaged GT 5.0.3) in April 2011. The IGE v2.0 release in early 2012 will see GridWay supporting the OGSA-BES, JSDL and HPC Basic Profile standards at the service level and as a client to other compute endpoints that use these standards.

7.5 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.

7.5.1 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.

7.5.2 Implementation Roadmap

There are many workflow engines and it is not the purpose of this capability to implement workflow engines but to ensure that the EGI Platform provides the interfaces needed to support their use.

From an IGE perspective, GridWay already supports the specification and execution of DAG workflows, and Globus has already been demonstrated to work with other workflow engines such as Taverna. In addition, Swift is a high-throughput parallel scripting language deeply integrated with GT5 for running parameter sweep style jobs over DCIs, to be delivered within the IGE v1.0 (IGE-packaged GT 5.0.3) in April 2011.

The SHIWA project [R 50] provides an approach for scientific workflows to be mapped on concrete DCIs to allow the execution of large-scale scientific experiments.

8 OPERATIONS CAPABILITIES

Maintaining and administering a production infrastructure poses a number of requirements on the deployed components.

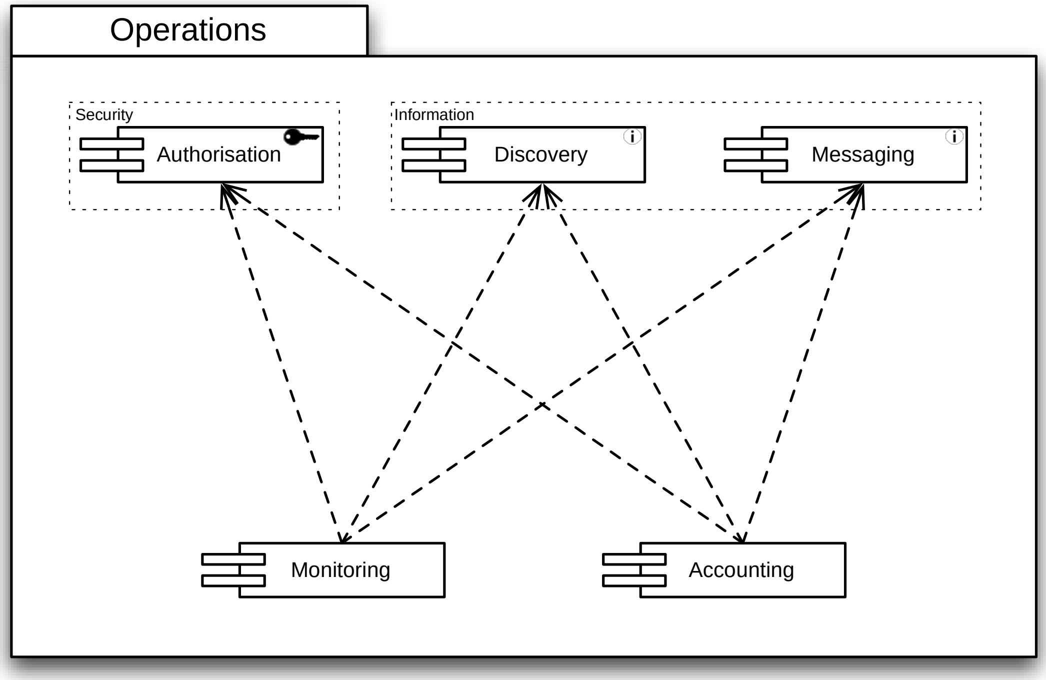

Figure 6: Dependencies of the Operations Capabilities

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.

8.1 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.

8.1.1 Supported Interfaces

Both programmatic and web based interfaces are needed to access this capability. The current integration point in EGI's production infrastructure is a framework based on Nagios [R 48]. Standardised monitoring interfaces are necessary, but are currently not available.

8.1.2 Implementation Roadmap

The monitoring capability is implemented through probes provided for each functional capability that is being monitored and a framework within which these probes reside. The probes are provided by either the operations teams or the technology providers, and are integrated into a framework deployed within the production infrastructure.

EGI delivers the Service Availability Monitor [R 49] service based on the Nagios framework.

From an IGE perspective, Nagios probes for various services already exist for GT5.

8.2 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.

8.2.1 Supported Interfaces

OGF defines a record format for accounting data, Usage Record (UR), GFD.98, [R 24]. OGF also suggests a draft of an access interface and a format for aggregated accounting records through the Resource Usage Service WG [R 23]). 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 [R 45] extension to UR [R 51, R 52], the OGF UR WG is now standardising accounting records for storage services.

An alternative approach to collection of accounting data is not to poll services (or service instances) for accounting data but to re-use a messaging infrastructure (see section 0) to have services publish their usage data to a dedicated accounting service.

8.2.2 Implementation Roadmap

EMI will deliver accounting record for their compute capabilities by mid-2011, perhaps in an intermediate minor release. No commitment is given for a standardised set of accounting record language for storage accounting. However, harmonisation across EMI is underway through the development of StAR.

IGE plans to deliver an accounting capability that utilises the RUS and UR standards via GridSafe in 2012.

9 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.

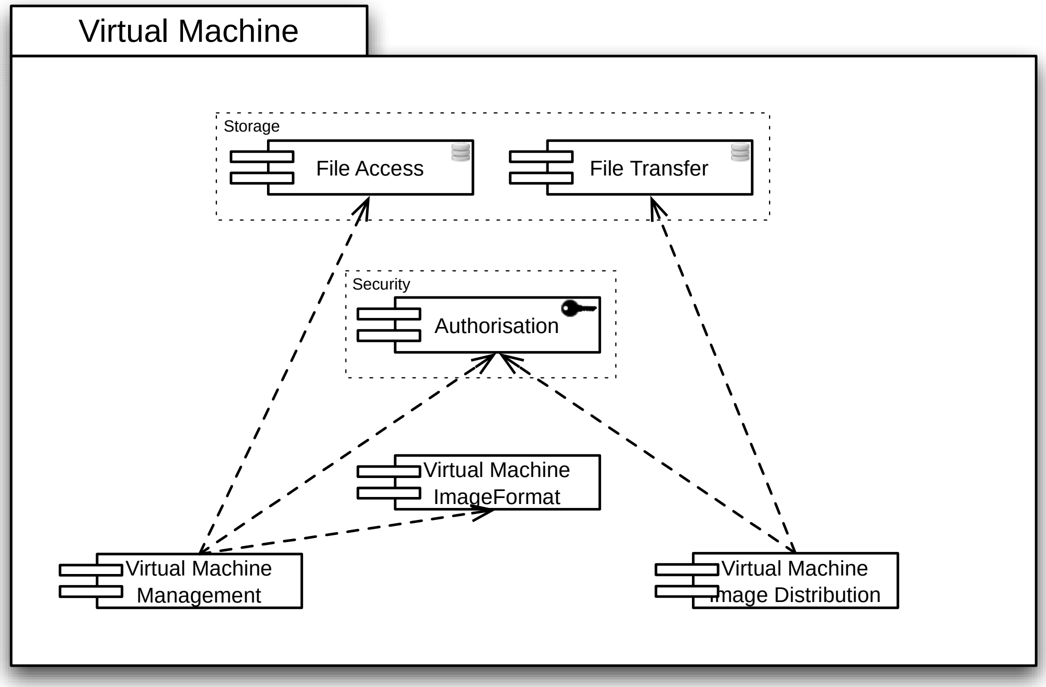

Figure 7: Dependencies of Virtualisation Capabilities

Central to interoperable Virtualisation is a commonly agreed image format for the virtual machines. The ImageFormat 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.

9.1 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.

9.1.1 Supported Interfaces

The OCCI (Open Cloud Computing Interface) [R 25] 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.

9.1.2 Implementation Roadmap

Various implementations of OCCI are available today from commercial providers and projects, such as Eucalyptus, OpenNebula, SLA@SOI, or Platform Computing.

Integration into UMD may happen not until mid-2012.

9.2 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.

9.2.1 Supported Interfaces

The OVF (Open Virtualisation Format) [R 26] from DMTF 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.

9.2.2 Implementation Roadmap

Many different virtualisation platforms support OVF today, ranging from Open Source solutions, such as VirtualBox (Oracle) to commercial solutions, such as VMWare ESX.

9.3 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.

9.3.1 Supported Interfaces

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

9.3.2 Implementation Roadmap

Protocols and interfaces need to be developed or adopted from elsewhere after consultation with relevant EGI communities.

10 INSTRUMENTATION CAPABILITIES

10.1 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.

10.1.1 Supported Interfaces

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

10.1.2 Implementation Roadmap

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

11 CLIENT CAPABILITIES

11.1 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 benefit domain specific application developers from evolving middleware while it may all the way easier to maintain a client side API for the most common Grid Use Cases that keeping track of and synchronising middleware interfaces.

11.1.1 Supported Interfaces

OGF provides the SAGA API as an approach to a common, lightweight and simple API for client-side abstraction of distributed computing resource.

11.1.2 Implementation Roadmap

This crosscutting capability needs close interaction and synchronisation with all distributed service types it interfaces with. Particularly, the implementation of middleware bindings (i.e. adaptors to SAGA) should be under control of the middleware service implementers, a model that has already been adopted for the development of Nagios plugins for the current EGI monitoring infrastructure.

A number of SAGA implementations exist that currently support GT5, including jGlobus and the SAGA C++ Reference Implementation maintained at the Louisiana State University Center for Computation and Technology (LSU/CCT) [R 39]. jGlobus 2.0, currently in alpha, supports client interfaces to GridFTP and GRAM, and Globus Online support is being investigated. In addition, the LSU implementation continues to support, to an increasing degree, the OGSA-BES and HPC Basic Profile standards in addition to JSDL. This aligns with IGE's plans to support these standards in GT5.2 and GridWay in 2012.

12 EGI PLATFORM

The EGI Platform is the collection of EGI Interfaces from the different EGI Capabilities and represents a commitment between the technology provider and the consumer of the interface (end-user, developer or operations) for stability and planned evolution that will be managed by the TCB and validated by the EGI.eu Technology Unit on behalf of the community. The EGI Platform is expected to evolve into a stable set of service interfaces or programming APIs (depending on the interface) that can be considered a solid foundation upon which to build services, applications or tools. The EGI Platform as a whole, and the individual Interfaces are versioned and changes are planned as part of a software component's release roadmap. Any changes to an interface are planned in consultation with its users and announced as part of the roadmap.

Wherever possible the EGI Interfaces will reference defined standards interfaces (from organisations such as the Open Grid Forum, IETF, W3C, OASIS, etc.), protocols or APIs provided by third parties. Advice is given to the community about interfaces that are emerging from the community and are 'candidate' interfaces within the UMD Roadmap. These interfaces are potentially subject to rapid change. If there is no consensus in the community on a common interface for a particular capability then such gaps are also noted.

The functional and non-functional aspects of the interfaces are defined and verified by the EGI.eu Technology Unit through quality criteria. Early drafts of these quality criteria for some capabilities are available [R1] and these will be expanded during the course of the project.

13 CONCLUSION

This document describes the functional capabilities of the EGI production infrastructure. Wherever possible existing or evolving standardised interfaces were used to define the external boundaries of capabilities. Following the use cases of distributed computing infrastructures, the functional dependencies between UMD Capabilities as described in the UMD Roadmap illustrate the composition and architecture of the EGI production infrastructure.

When possible, components and aspired release dates were taken from published release schedules [R 40] or development plans [R 8] as provided by the Technology Providers.

Driven by the requirements of the user and operations communities of EGI the Roadmap will continue to evolve, based on input identifying:

1) Additional capabilities not currently listed
2) Prioritisation of the listed capabilities
3) Key interfaces that need to be supported to access these capabilities
4) Implementations from technology providers able to deliver these capabilities

Feedback on the UMD Roadmap can be sent to cto@egi.eu

14 REFERENCES

R 1	https://wiki.egi.eu/wiki/EGI-InSPIRE:UMDQualityCriteria
R 2	http://ogf.org/documents/GFD.20.pdf
R 3	http://www.ggf.org/documents/GFD.129.pdf
R 4	https://documents.egi.eu/document/109
R 5	http://www.ggf.org/documents/GFD.147.pdf
R 6	SAML 2.0, OASIS, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security
R 7	OpenID, http://openid.net/
R 8	EMI Technical Development Plan, D1.3.1 http://cdsweb.cern.ch/record/1277540
R 9	Shibboleth, http://shibboleth.internet2.edu/
R 10	Terena Certificate Service, http://www.terena.org/activities/tcs/
R 11	ActiveMQ, http://activemq.apache.org/
R 12	Advanced Message Queuing Protocol, http://www.amqp.org
R 13	RabbitMQ, VMWare, http://www.rabbitmq.com/
R 14	AMQP Infrastructure, Fedora, http://fedoraproject.org/wiki/Features/AMQP_Infrastructure
R 15	Apache QPID, http://qpid.apache.org/
R 16	Reliable File Transfer, Globus Project, www.globus.org/toolkit/data/rft
R 17	EMI Data Area Work Plan and Status Report, D5.2.1 http://cdsweb.cern.ch/record/1277615
R 18	Distributed Resource Management Application, http://www.ogf.org/documents/GFD.22.pdf http://www.ogf.org/documents/GFD.133.pdf

R 19	WS-DAIR, OGF, http://www.ogf.org/documents/GFD.74.pdf and http://www.ogf.org/documents/GFD.76.pdf
R 20	WS-DAIX, OGF, http://www.ogf.org/documents/GFD.74.pdf and http://www.ogf.org/documents/GFD.75.pdf
R 21	OGSA-DAI, http://www.ogsadai.org.uk/
R 22	DCI Federation WG, OGF, http://www.ggf.org/gf/group_info/view.php?group=dcifed-wg
R 23	OGSA Resource Usage Service WG, OGF, http://www.ggf.org/gf/group_info/view.php?group=rus-wg
R 24	Usage Record WG, OGF, http://www.ggf.org/gf/group_info/view.php?group=ur-wg
R 25	Open Cloud Computing Interface WG, OGF, http://www.ggf.org/gf/group_info/view.php?group=occi-wg
R 26	Virtualization Management (VMAN), DMTF, http://www.dmtf.org/standards/vman
R 27	Simple API for Grid Applications (SAGA), OGF, http://ogf.org/gf/group_info/view.php?group=saga-wg
R 28	EGI-InSPIRE Standards Roadmap, EGI Deliverable D2.5, https://documents.egi.eu/document/206
R 29	EGI-InSPIRE User Feedback and Recommendations, EGI Milestone MS305, https://documents.egi.eu/document/211
R 30	EGI-InSPIRE Technical Note on EGI-InSPIRE MS305 https://documents.egi.eu/document/267
R 31	OAuth, http://oauth.net/
R 32	Grifi, GridFTP File System, http://grifi.sourceforge.net/
R 33	Parrot, http://www.cse.nd.edu/~ccl/software/parrot/
R 34	UMD Roadmap, first edition, EU Deliverable D5.1, https://documents.egi.eu/document/100
R 35	EGI Operations Architecture, EU Deliverable D4.1, https://documents.egi.eu/document/218
R 36	AdHoc Resource Management, GridWiseTech, http://www.gridwisetech.com/products/adhoc

R 37	Globus Integrated Information Service, The Globus Alliance, http://www.ci.uchicago.edu/wiki/bin/view/IIS/WebHome
R 38	Java Message Service, JSR-914, Java Community Process, http://jcp.org/aboutJava/communityprocess/final/jsr914/index.html
R 39	SAGA Reference Implementation, LSU/CCT http://saga.cct.lsu.edu/software/cpp
R 40	IGE Release Schedule version 2 (1 February 2011), http://tinyurl.com/63dhbmx
R 41	eduPerson/eduOrg, http://middleware.internet2.edu/eduperson/
R 42	eduGAIN, Federated Identity, GEANT project, http://www.edugain.org/
R 43	eduGAIN Metadata Policies, eduGAIN, GEANT Project, http://www.edugain.org/policy/
R 44	EMI response on evaluation of eduPerson on the TCB mailing list, Action Item 01/09 https://www.egi.eu/indico/conferenceDisplay.py?confId=268
R 45	EMI Storage Accounting Record (StAR) format, https://twiki.cern.ch/twiki/bin/view/EMI/StorageAccounting
R 46	Grid Relational Catalog Project (GRelC), http://grelc.unile.it/home.php
R 47	AMQP Review and Implementation process, AMQP Working Group http://www.amqp.org/confluence/pages/viewpage.action?pageId=4489238
R 48	Nagios, Open Source Monitoring Framework, http://www.nagios.org/
R 49	Service Availability Monitor (SAM), EGI-InSPIRE Project, https://wiki.egi.eu/wiki/WP7:_Operational_Tools
R 50	SHIWA Workflow engine, SHIWA project, shiwa-workflow.eu
R 51	EMI's StAR contribution to OGF UR-WG, http://www.ogf.org/gf/event_schedule/index.php?id=2029
R 52	OGF UR-WG taking up standardising Storage Accounting records, http://www.ogf.org/gf/event_schedule/index.php?id=2116

15 GLOSSARY

Technology Provider	A T*echnology Provider* is an organisation or a project that is collaborating with EGI to develop or deliver software for use within the production infrastructure or its user community. Typically, a Technology Provider commits to provide software that implements one or more of the EGI Capabilities described in this roadmap.
Software Component	A S*oftware Component* is provided by a technology provider in a software release and is the smallest granularity of software considered within this document.
Unified Middleware Distribution (UMD)	The *Unified Middleware Distribution* is the set of software components developed for EGI by technology providers to provide the innovation needed by EGI to satisfy its users that cannot be found elsewhere and is therefore endorsed by EGI for use within the production infrastructure.
(EGI) Capability	An EGI *Capability* (e.g. authorization, job submission, data movement, etc.) is a high-level description of activity needed by either the end-user or operations community that is defined and delivered by one or more I*nterface*s that may be supported by one or more technology providers.
(EGI) Interface	An *EGI Interface* consists of different functional aspects that together or separately are verified through the component acceptance criteria, i.e. different functional aspects of the interface have defined acceptance criteria and are verified by the EGI.eu Technology Unit. These interfaces may initially be specified through free text but will over time be defined by a particular protocol, specification or standard.
(EGI) Functionality	An *EGI Functionality* is provided as part of an EGI Interface to meet a requirement identified by the end-user or operations communities within EGI.
UMD Release	The *UMD Release* is the set of components within UMD that are released to provide an integrated software distribution. There may be components in UMD that are not included in the UMD Release but provide functionality that is only available from the EGI Software Repository.
(EGI) Platform	The *EGI Platform* is the collection of UMD Interfaces that is implemented within UMD. Note – this is distinct to the underlying OS environments supported within EGI.
(EGI) Community	The *EGI Community* are the 'DCI projects', other national or European Commission funded projects, the European National Grid Initiatives and the user communities benefiting from the European Grid Infrastructure.
Technology Roadmap	The *EGI Technology Roadmap* provides a broader view to the use of technology within the EGI production infrastructure, e.g. supported operating systems, messaging technologies, deployed software, etc. The EGI Technology Roadmap includes components sourced from outside the EGI Community.
UMD Roadmap	The *UMD Roadmap* defines the evolution of UMD, the software

	components coming from within the EGI Community, which will be deployed to deliver the production infrastructure.
Technology Coordination Board (TCB)	The *Technology Coordination Board* (TCB) brings together stakeholders within the EGI Community to prioritise their requirements into the broader EGI Technology and the UMD Roadmaps, which will be issued to together at least every 6 months.