Technical Strategy for the Open Grid Forum 2007-2010

Status of This Document

This document provides information to the Grid community on the overall technical strategy of the Open Grid Forum (OGF). It does not define any standards or technical recommendations.

This is an evolving document and as a result even formally published versions are subject to regular revision. The contents reflect the broad (but not universal) agreement of the Technical Strategy Committee, but are not intended to restrict or direct activities within the OGF. Rather this strategy document points in a direction that appears to the Technical Strategy Committee to be valuable and important at this time.

Copyright Notice

Abstract

This document describes the overall OGF technical strategy spanning a timeframe from 2006 to 2010. The technical strategy ultimately describes the output of the OGF standards working groups as well as the requirements (in the form of capabilities or functions) that serve as the inputs to standards working groups. This version of the technical strategy is represented in the form of a roadmap of standards working group output over time with specific short-term milestones and target deliverables. It is expected that later versions will address technical strategy in the context of Enterprise and e-Science activities, such as the Vendor Adoption Forums and the Grid Interoperability Now (GIN) activity.

Contents

1. Introduction

OGF's mission (www.ogf.org) is to build an international community to accelerate Grid adoption by providing an open forum for grid innovation and developing open standards for Grid software interoperability; and, this mission is based on that belief that standards-based grid computing is critical to enabling business value and scientific discovery.

In the spirit of openness as well as enabling business value and scientific discovery, the TSC was established to ensure that there is an alignment between the OGF's technical strategy and the needs of the business and scientific community. Formulation of a technical strategy is the primary output of the TSC. The TSC meets on a regular basis and consists of members of the OGF community who represent the Grid community of users, architects, developers, vendors, etc.

1.1 Purpose of this Document

This document, the Technical Strategy Document (TSD), is intended to capture and communicate OGF's technical strategy. "A strategy is a long term plan of action designed to achieve a particular goal, as differentiated from tactics or immediate actions with resources at hand" [1]. The OGF TSD is intended to:

1. Provide a concise view of the OGF technical direction and priorities.

2. Provide a mechanism to align key stakeholder requirements with OGF technical directions and priorities

3. Provide an indication of where more effort is needed, and what actions are needed to promote specific standards within the industry.

1.2 Document Structure

The structure of this document is as follows. In Section 2, we present a concise statement of the three-year goal of the Open Grid Forum. In Section 3 we outline the alignment process adopted by the Technical Strategy Committee to align and prioritize the technical strategy with our stakeholders, while section 4 we identify the current result of this process in the form of high value use cases or scenarios that need to be addressed to meet our goals. In section 5, we provide a fairly comprehensive list of Grid capabilities and functions drawn from the use case work of the OGSA-WG. Section 5, outlines the tactical priorities in the form of a roadmap for several identified specifications and presents a gap analysis table indicating ares for further standards focus.

1.3 Background

The long-term vision of Grid can be summed up as follows: "Distributed computing across multiple administrative domains." The notion of *distributed computing* as used in this definition includes a wealth of highly complex technologies, some still the focus of research. The reality of this definition complicates matters further by including operation across multiple domains of administrative control. The security, privacy, economic, and political aspects of Grids increase by orders of magnitude with the introduction of Internet scale operation.

The concept of Grid has grown from serendipitous cycle recovery projects such as SETI@Home, to planned desktop cycle sharing via tools such as Condor, to Grids built on dedicated resources, ranging from blade servers in a corporate data center to trans-national collections of supercomputers. Our focus is on standards and tools to effectively build and utilize the last of these.

We believe that Grid is composed from the following characteristics and goals:

- Infrastructure virtualization
- Resource pooling and sharing
- Self-monitoring and improvement
- Dynamic resource provisioning
- Highest quality of service

Not all of these are in every Grid, but every Grid has several of them.

We find it helpful to use a taxonomy of different "Grids" in discussions. This is not a strict taxonomy such as used by botanists, but instead a shorthand notation that points toward a particular usage style:

- **Collaboration Grids**. These Grids involve multiple organizations and individuals, security domains, protocols, discovery mechanisms, and heterogeneous hardware, collaborating to share their resources to make the most effective use of it for their combined user communities. This is the original and long-term vision of Grids and should no be confused with the domain of collaboration tools such as Access Grid.

- Data Center Grids. These Grids are in most ways as complete technically as collaboration Grids and involve the complete dynamic life cycle of service deployment, provisioning, management and decommissioning as part of their normal operation.

  At first glance, they may appear to be missing the aspect of multiple administrative domains, but that is typically an illusion. While the funding may come from a single source, and the administration carried out by a single organization, there is typically just as much tension among the various user entities as in a Collaboration Grid.

  For example, in the *Utility Computing* use case, a Data Center Grid exists inside a single Enterprise, but provides services for many individual political/security domains on an infrastructure managed with grid protocols, subject to varying service level agreements and payment schemes. This results in multiple domains sitting on top of an integrated domain, with a complex hierarchy of security constraints, resource lifetimes and performance requirements.

- Cluster Grids. Aimed at high performance/throughput computing, these Grids are mostly workload scheduling environments. They tend to be less dynamically deployed and more homogeneous in their construction. Their services are either generic in nature, *e.g.*, a job submission service, or provide the same service all the time. The provisioning decisions may be almost entirely driven by service level agreements for a fixed set of services and customers. They do not typically support the whole service provisioning life cycle.

It is perhaps better to think of these (and others) as a set of perspectives, taken from different points against the same vision of Grid as a pervasive, scalable, shared, resilient, integrated platform.

Much of the work of the OGF has its origins in the ongoing efforts taken from the GGF and EGA activities. Although OGF remains open to new and innovative approaches to Grid computing, much (but by no means all) of the work outlined here has been underway for some time as part of either the "Open Grid Services Architecture" or the "Reference Model and Use Cases". These two bodies of work continue to inform and guide our strategy going forward.

2. Goal of the Open Grid Forum: 2007 – 2010

Concisely put, the Goal of the Open Grid Forum for the 2007 to 2010 time frame is given below.

> **The Open Grid Forum should commit all its available resources to the goal that before this decade is out, commercial and academic organizations will build real operational grids using OGF-defined components.**

No other single technical goal can more completely focus the activities of our united organization or more clearly define its success, and no other goal will be more challenging or difficult to achieve. Furthermore, achieving this goal will require us to draw energy from all stakeholders within the organization.

One important aspect of this goal is that it is defined in terms of specific use case patterns and the specifications or practices needed to enable these scenarios. In some cases, the development of a particular specification may still be in a very early and immature state - more of a collection of community-initiated practices. Thus, it is anticipated that each high-level use case pattern will identify a number of capabilities or functions, for which there may be concrete specifications. However in some cases there may be some gaps that must be filled by community practices until further technical and/or political maturity occurs in the standards arena. To build a strategy around the goal we must:

- Identify and focus on the main, common use-cases, patterns, and scenarios that commercial and academic grids require.
- Provide best practice and other documents that allow communities to evaluate and adopt Grids today and provide a pathway for the standards process.
- Identify and complete the core architectural standards required to build robust, commercially viable, grid solutions.
- Mobilize the whole of OGF, all the working, community and research groups to meet this challenge.
- Encourage software developers, in the open source and commercial communities, to adopt and implement these standards in products and offerings. They must do so early and often, as this is part of the standards process.
- Hold regular alignment summits where the OGF functions and key stakeholders review the technical strategy and update this document based on lessons from OGF activities and the complete Grid community. Here we would review lessons from the Grid interoperability work in OGF (eScience), OGF best practice documents as well the Enterprise Voice of Community and other forums.

3. Technical Strategy Alignment Process

Figure 1 depicts a high-level view of the OGF Technical Strategy Alignment process.

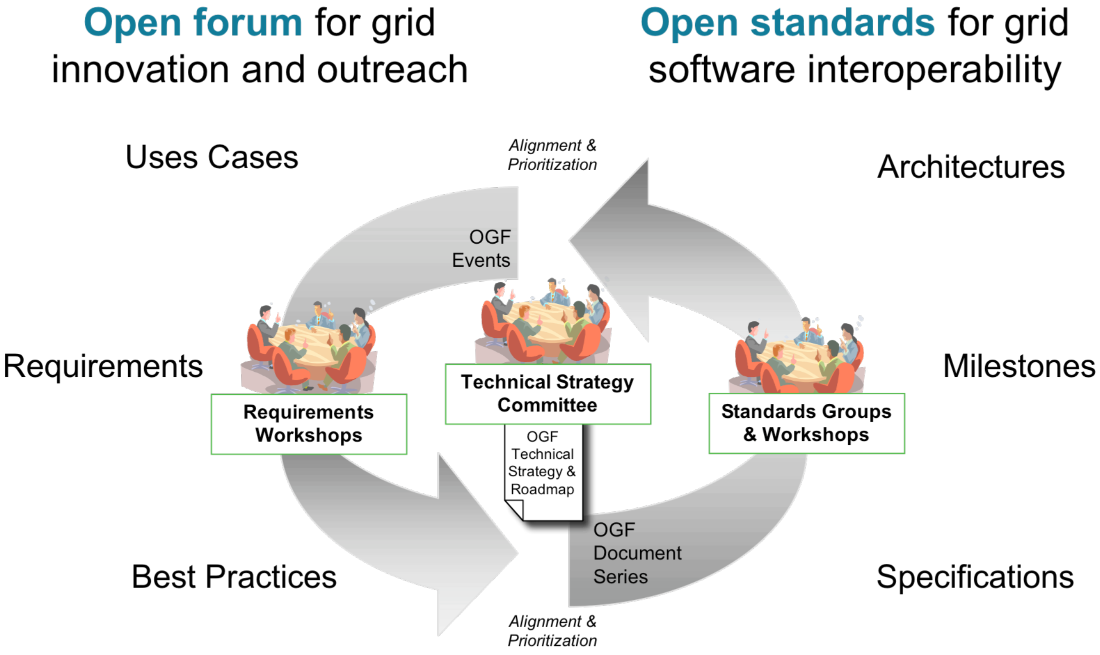
Open forum for grid innovation and outreach
Open standards for grid software interoperability
Uses Cases
Alignment & Prioritization
Architectures
OGF Events
Requirements
Requirements Workshops
Technical Strategy Committee
Standards Groups & Workshops
Milestones
OGF Technical Strategy & Roadmap
OGF Document Series
Best Practices
Specifications
Alignment & Prioritization

Figure 1 : Technical Strategy Alignment Process

The right half of the alignment process is concerned with the standards working groups and their production of specifications and reference architecture. The left half of the alignment process is concerned with the inputs to the standards working groups. This input represents requirements from the Grid community at large and may be represented in the form of use cases or best practices. The requirements are gathered from various requirements gathering groups such as the Enterprise Grid Requirements Research Group (EGR-RG) or the Storage Networking Community Group (SN-CG), or the Telco Community Group. Each of these groups meets to capture requirements that are particular that group. Each group's requirements are then rolled up into a merged and prioritized list that is then brought forward to the Technical Strategy Committee (TSC).

The TSC represents the key part of the alignment process where requirements are matched-up against the current technical direction of the standards working groups, and a determination is made as to the degree of alignment/misalignment that exists between the requirements and the technical direction. These three parts, the requirements gathering, the standards work and the technical alignment, all operate simultaneously and in parallel.

When merged and prioritized requirements are brought forward to the TSC, assessment criteria will be applied to each requirement in order to determine the appropriate response to that requirement.

The assessment criteria for incoming requirements are as follows:

1) What is the degree of alignment with OGF objectives, goals and current technical direction?

   a) Is the requirement just an extension to what OGF is already doing?

   b) Is requirement a minor tweak to the current technical direction?

   c) Is there a working group that already exists that would be a natural fit to handle the requirement?

   d) How universal is the requirement to the community at-large?

2) How important and beneficial is this requirement to the community?

   a) How universal is the requirement to the community at-large?

   b) Is this requirement a top-line priority to a particular segment of the community?

3) Do the resources exist to address the requirement?

   a) Do the resources exist to actually work-on the requirement?

   b) Do the skills/knowledge/expertise exist to help address the requirement?

   c) What is the interest level in working on the requirement?

4) Does this requirement exclude other industries or vendors?

5) What is the magnitude (resources and time) of the effort needed to address the requirement?

6) How complex or risky is the requirement?

7) What is the timeframe in which the requirement needs to be addressed in order or it to be useful?

Once the assessment criteria for the incoming requirements have been applied the next step is to determine the appropriate action/response to each requirement. The range of actions/responses is as follows:

1) Send the requirement to an existing working group for whom the requirement would be a natural fit.

2) Start a new standards working group to work-on the requirement.

3) File the requirement as pending due to current lack of interest or resources.

4) Ignore as out of scope for OGF.

5) If the requirement is already being addressed by an existing standards working group, make the connection between the source of the requirement and the working group. Make certain to include in this work in the TSD. If a specification is already published, send an open letter to the vendor/developer community to suggest implementation.

6) Form a new standards working group to create a specification for existing technology.

7) Form a new Enterprise Group to develop a Best Practices Document that might offer an interim solution (during standards development) or may turn-out to be a prescribed permanent solution.

8) Develop a new OGF processes to handle addressing the requirement.

9) Refer to an existing technology specification from outside OGF.

10) Declare the requirement out of scope if it is not consistent with the OGF mission or objectives and may not be a “Grid” issue.

4. High Priority Capabilities

Table 1 lists capabilities identified in the OGSA Use Cases document [16] and as a result of surveys carried out by the Technical Strategy Committee. This list is not complete, nor has any priority been associated with each capability at this stage.

Table 2: Capabilities of Grids

Category	Capability
Security	Multiple Security Infrastructures[i]
	Perimeter Security Solutions[ii]
	Virtual Organization[xvi]
	Encryption[iii]
	Certification[xiv]
	Authentication[iv]
	Authorization[xv]
	Web Service Protocol Security
Operations	Instantiate New Services[xiv]
	Deployment[xvi]
	Provisioning[xvi]
	Service Level Management[xvi]
	Notification[xvi]
	Messaging[xvi]
	Logging Service[xvi]
	Service and Resource Monitoring[xvi]
	Metering and Accounting[xv]
	Policy[xvi]
	Policy Management[xvi]
	Administration[xiv]
	Systems Management[xiv]
	Aggregation of Services and Resources[xiv]
Resource Management	OGSA-Naming[v]
	Resource Discovery[xvi]
	Resource Brokering[vi]
	Job Management[xvi]
	Choreography, Orchestration and Workflow[xvi]
	Resource Virtualization[xvi]
	Information Model[xvi]
	CPU Scavenging[vii]
	Legacy Programs[xvi]
	Reservation[xvi]
Data	Data Movement
	Data Access
	Data Integration
	Data Management
	Data Provisioning[xvi]
	Metadata[xvi]
Application Development	Application Debugging[xvi]
	Application APIs[viii]
Foundations	Communication Protocols[xvi]
	Architecture[xvi]
	Grid Semantics[ix]
	Grid Fabric [xvi]
System Properties	Fault Tolerance[x]
	Load Balancing[xi]
	Failure Recovery[xvi]
	Self-Management[xvi]

Based on the current state of play within the standards development activities of OGF, the input available from the former EGA, the results of a recent community survey, and discussions with key stakeholders within the community, the following capabilities have been identified as priority targets to meet the stated goal in Section 2.

4.1 Grid Security

Several critical security areas need to be addressed in the near term. These are focused on creating interoperability standards supporting scaleable access control for basic Grid end-to-end use cases. For the purpose of this document, we group these broadly under Authentication, Web Service Protocol Security, and Authorization.

4.1.1 Authentication

Authentication deals with the process of verifying the identity, and attributes, associated with a principal(s) within the grid environment. Existing systems rely on a variety of security credential types as the basis for principal authentication. These include: name-password pairs; X.509v3 certificates; proxy certificates; attribute certificates, Kerberos tickets; and SAML tokens.

The existing credential types allow for considerable variability in how identities and attributes are encoded and there are multiple authentication algorithm standards, which may be used. Given this situation, it is important for the grid community to develop standard authentication profiles to serve as a basis for interoperability. Guidelines on the use of multiple credentials are also needed. Grid environments may require a principal to present a set of credentials obtained from multiple authorities (i.e., X.509 CAs; SAML Authorities; VOMS Servers [6]) to supply all information needed to authorize an action.

In some environments, the ability to revoke credentials, preventing their continued use for authentication, is deemed critically important. Standards exist for handling X.509 revocation, though unique aspects of grid environments suggest grid-specific profiles are needed. For some other credential types the grid community may need to develop new revocation approaches.

The grid use cases also identify a critical need to delegate from one principal to another, typically, between a user and a job running on their behalf. Several proprietary approaches to handling this requirement have been developed. These should be supplemented with interoperability standards defining how delegation credentials are securely transferred and subsequently used when accessing resources.

4.1.2 Web Service Protocol Security

There is growing interest in using a Service Oriented Architecture, based on standardized web service protocols (e.g., SOAP over HTTP), within grid systems. There is a large body of specifications defining composible functionality layered on these basic protocols. These cover things such as addressing, routing, session negotiation, and security.

The WS-I Basic Security Profile 1.0 [9] defines a collection of normative profiles that provide guidance for interoperable secure communication based on these specifications. This addresses basic communication security needs such as message authentication, integrity, and confidentiality and is intended to address a broad set of operational environments. A grid-specific profile is desirable which highlights the recommended options and functionality required to address grid messaging requirements.

In addition, the grid community needs recommended guidelines and standards for how to leverage these basic secure protocol capabilities for more complex interactions. For example: mutual authentication; session negotiation; conveyance of delegation credentials; use of credential renewal services; and so forth.

4.1.3 Authorization

There are a number of authorization systems currently available for use on the Grid as well as in other areas of computing, such as Akenti [3], CAS [4], PERMIS [5], XACML. On the abstract level these types of authorization services have similar semantics: they are given a description of the initiator (identity, attributes, and possibly externally determined privileges); a description of an action being requested; details about the target resource; and any contextual information such as time of day. In response, they provide an authorization decision indicating whether the requested action should be performed or rejected, possibly with supplemental information such as auditing data.

These existing systems were developed independently over a number of years. Not surprisingly, they require the using grid service (typically a resource access gateway) to express the authorization query inputs, and communicate with the system, using a proprietary mechanism. One also finds some architectural differences in these systems, which impacts the calling grid service's logic. For example, whether credential validation and decoding is a separate function from the authorization query. These differences impact grid system development by forcing service developers to select an authorization system early in the development process. Once selected, replacing, or extending, that system to meet the needs of specific operational environment can be complex and costly.

Developing interface standards for the major functional components within an authorization system will provide a uniform way for grid services to interact with authorization services. It also provides a basis for existing and future authorization systems to evolve while maintaining compatibility with deployed grid services. It is expected authorization services will either adopt these standard interfaces or provide a mapping to their proprietary interfaces. The, generally accepted, authorization service architectural model which has evolved over the past few years provides a basis for this work. It envisions three major functional components: a Policy Enforcement Point (PEP); a Policy Decision Point (PDP); and a Credential Validation Service (CVS). The standards will describe these functional components, their interfaces, and the various ways they are expected to interact.

4.2 Application Provisioning

Job submission, and indeed any sort of workload manager, implies the ability to discover, describe, provision and manage the lifetime and lifecycle of an appropriate application code onto an identified computing resource. In many instances, this can be done at a very high level, but some scenarios will require very specific descriptions at the application layer. This, in turn, may place requirements for a specific operating system and version, possibly implying a certain patch level and hardware requirements. EGA's Reference Model describes the overall flow of activity involved in provisioning a high-level component and decomposing the required work into accessible quanta: ACS and CDDLM are specific proposals/WGs that attack the problems of describing and managing the lifetime of specific applications.

4.3 Job Submit

The simplest job submit use case is a high-throughput compute cluster that is managed by a batch job scheduler and that is used only from within an organization. Aspects to consider include user interface (semantics only, not GUI issues), state model, and resource descriptions. With respect to the user interface, users expect to be able to submit jobs, query the status of running jobs, cancel a job, and list jobs belonging to them. The state model needs to capture, at a minimum, the concepts of running and finished, as well as a state before execution commences (pending or queued).

Users expect to be able to discover something about a job service before they attempt to use it. However, given the complexity of the resource modeling domain, only a small set of standardized properties can be specified, such as number of CPUs/compute nodes needed, memory requirements, disk requirements, etc.

A number of common use cases that extend this simple use case should also be addressed. In particular, being able to describe a service's fault tolerance model, to handle extended functionality offered by specialist schedulers, to provide notification of job status to the user, and to advertise and request other aspects of quality of service.

4.4 File Movement

The TSC has identified a need to define an interface that standardizes the process of invoking the movement of large amounts of data. This capability covers the problems of discovering data transport protocols available at the data's source and destination locations and agreeing on one of them, and the actual invocation of the agreed data movement, including direct data movement and 3rd party data movement. Executing a data movement includes the invocation of the transport protocol itself, and applying the previously agreed parameters where appropriate. While the data movement is executing, control and management operations on the data movement are necessary, such as "cancel," "suspend," and "resume." Progress information, including general status information, must be provided to interested parties as well.

The OGF GridFTP standard [10] provides most of the capabilities just described, but does not define a Web Services interface or address discovery issues. Further work is required to address those concerns.

4.5 Data Provisioning and Data Grids

Data intensive grids are of increasing importance and require components to handle files, different types of databases, caching, transport, metadata, federation leading to managed data, information and knowledge. One needs to address provisioning and management at both the data and storage levels.

To do data provisioning, the GME (Grid Management Entity from the EGA Reference Model) must become the intermediary between the compute, switching and storage infrastructures that make up the set of grid resources. The dynamic nature of grid-based applications requires provisioning on several levels at once to achieve what may look to the end user like an atomic operation.

Data provisioning typically requires at least three steps: initial population, keeping the data in sync, and cleaning up the data when it is no longer needed. If the container must be populated with an initial data set from somewhere, additional work is required. There may be an opportunity to use cloning technology to greatly enhance the efficiency of the copy operation.

After the initial provisioning step, the data may need to be frequently snapshotted and/or replicated for Disaster Recovery or other purposes. At the end of the job, results may need to be copied elsewhere to a location of the client's specification. In addition, all temporary copies of any data may need to be securely "shredded" when the user or application is done with the container and its offspring. Yet the user's desire is simple: a data container conforming to some service level that the system has previously advertised.

This high-level view decomposes rapidly into a number of other problems, each a significant subject in its own right. In addition, at bottom one needs APIs that actually perform the provisioning and monitoring operations in order to build a GME that can offer the convenient and dynamic abstraction of a grid, which holds so much promise.

Once that decomposition has been done, we are in a position to examine whether suitable interfaces into the actual grid resources exist, and if so, where.

4.6 Grid APIs

Some people believe that a barrier to broad adoption of the Grid paradigm is the continued evolution of the underlying programming interfaces. Developers of both end user applications and middleware services need programming interfaces that provide stability across both different middleware technologies and changes in the underlying protocols through either different approaches or versions.

Having started from a diverse set of use cases collected from the 'grassroots' OGF community, the Simple APIs for Grid Applications (SAGA) working group has developed an application programming interface (API) specification that is agnostic to the underlying middleware. This API includes functional support for job submission and management, resource discovery, and data management, access and replication. This is on top of generic support for asynchronous notification, error reporting and security. As the semantics of the generic API stabilizes and moves forward to standardization, work continues on generating language specific bindings and the solicitation of new use cases to drive a second round of API development.

5. Tactical Priorities and Roadmap

Table 3 lists the specifications, where identifiable, needed to provide the capabilities outlined in Section 4. The table is organized as follows:

- Capability: The capability for which this specification is required.
- Specification Name: The short name of the specification where possible. If no specification exists yet, this entry is left blank. Note that there may be several specifications addressing a given capability.
- Current Status: The current status of this specification on the following scale from Concept through Deployment. The levels on the scale are roughly sequential, but not all steps are always taken.
    - Concept: Concept exists and (proprietary) proof of concept implementations exist.
    - WrkGrp: Working Group formed to create the specification.
    - Draft: Draft specification exists.
    - Interop: Reference Implementations and Working Group lead interoperability tests.
    - Spec: Specification completed to OGF Proposed Recommendation.
    - Full Rec: Specification completed to OGF Full Recommendation.
    - Product: Available as a supported product, including Open Source based service contracts.
    - Deploy: Deployment observed in a production setting, commercial or technical.
- Milestone 1: The month, year and target status for the first milestone with respect to the specification. These milestones need not point to the next stage in the status list.
- Milestone 2: The month, year and target status for the second milestone with respect to the specification.
- Area: OGF technical area of responsibility for the specification.

Table 3: Simplified Specification Roadmap

Capability	Specification	Status	Milestone 1	Milestone 2	Area
Grid APIs	SAGA	Draft	Feb 07: Spec	Dec 07: Product	Applications
	DRMAA	Product	Jan 07: Full Rec	Dec 07: Deploy	Applications
	Grid RPC	Interop	Jan 07: Full Rec	Dec 07: Deploy	Applications
Job Submit	JSDL 1.0	Product	Feb 07: Deploy		Compute
	OGSA-BES	Draft	Mar 07: Spec	Dec 07: Product	Compute
	HPC Profile	Draft	Mar 07: Spec	Dec 07: Product	Compute
File Movement	DMI	WrkGrp	Dec 06: Spec	Dec 07: Product	Data
	ByteIO	Interop	Oct 06: Spec	Dec 07: Product	Data
	GridFTP	Product	Dec 06: Deploy		Data
Data Provisioning and Data Grids	WS-DAI	Spec 74	Aug 07: Product	Mar 08: Deploy	Data
	WS-DAIR	Spec 76	Aug 07: Product	Mar 08: Deploy	Data
	WS-DAIX	Spec 75	Aug 07: Product	Mar 08: Deploy	Data
Application Provisioning	CDDLM	Spec 69	Aug 07: Product	Mar 08: Deploy	Management
	ACS	Spec 73	Aug 07: Product	Mar 08: Deploy	Management
Authentication	OGSA-AuthN	Concept			Security
	OCSP Profile	Draft			Operations
Authorization	OGSA-AuthZ	WrkGrp	Jan08:Spec		Security
	Distributed Audit	Concept			Security
Web Services Protocol Security	OGSA-SBP-Core	Draft	Aug 07: Product	Mar 08: Deploy	Architecture
	OGSA-SBP-SecChan	Draft	Aug 07: Product	Mar 08: Deploy	Architecture
	Grid Extended Interaction Profiles	Concept			Security

Table 3 is not a complete list of OGF activity nor is it a statement of the overall importance of these specifications with respect to the rest of the work in OGF. These specifications merely address the priority capabilities set out in section 4; other work in OGF continues independently.

The contents and schedule represented in Table 3 will change over time, based on input from stakeholders as to perceived priorities, chairs in terms of available resources to meet milestones, and general input from the community.

5.1 Medium Term and Gap Analysis

Table 4 contains the capabilities from Table1 not addressed in the short term as listed in Table 3. These provide some insight into the future directions and gaps in the Grid roadmap. This list, like the others contained herein will evolve over time.

The “Specification” column describes the current state of play with respect to each capability. If there are existing activities for this capability, then these are listed. If there is active work in another organization, the name of that organization is listed. Each capability may also be out of scope for the OGF or be an implementation specific capability. Otherwise it is a gap.

The Maturity column indicates the rough State of the Art with respect to this capability. These, in rough order of maturity, are: *Out of Scope* for the OGF, a *Gap* that OGF should be investigating, an area of Grid *Research*, an *Evolving* area either in OGF or some other SDO, or a capability with *Mature* specifications or solutions.

Table 4: Medium Term and Gap Analysis

Category	Capability	Working Group or Comment	Maturity
Security	Multiple Security Infrastructures[i]	OGSA Auth-Z	Evolving
	Perimeter Security Solutions[ii]	Firewall Issues RG	Research
	Virtual Organization[v]	VOMS work applies	Gap
	Encryption[iii]	Existing technology is currently adequate	Out of Scope
	Certification[iii]	CA Ops WG	Evolving
	Authentication[iv]	OGSA-AuthN	Evolving
	Authorization[iv]	OGSA-AuthZ	Evolving
	Web Service Protocol Security	OASIS/WSS, OGSA Secure Channel	Mature
Operations	Instantiate New Services[iii]	CDDLM-WG, degenerate workflow	Evolving
	Deployment[v]	ACS-WG, CDDLM-WG	Evolving
	Provisioning[v]	ACS-WG, CDDLM-WG	Evolving
	Service Level Management[v]	GRAAP-WG	Evolving
	Notification[v]	OASIS/WS-Notification, WS-Eventing	Mature
	Messaging[v]	OASIS/WS-Notification, WS-Eventing	Mature
	Logging Service[v]	Related to metering, see below	Gap
	Service and Resource Monitoring[v]	Grid Monitoring Architecture	Evolving
	Metering and Accounting[iv]	UR-WG and RUS-WG More needed	Evolving
	Policy[v]	WS-Policy	Evolving
	Policy Management[v]	Management standards needed for policy	Gap
	Administration[iii]	Community practices needed	Gap
	Systems Management[iii]	Reference Model-WG	Evolving
	Aggregation of Services and Resources[iii]	See OASIS WS-ServiceGroup	Mature
Resource Management	OGSA-Naming[v]	WS-Naming-WG, GFS-WG	Evolving
	Resource Discovery[v]	OASIS/WSDM	Mature
	Resource Brokering[vi]	RSS-WG	On Hold
	Job Management[v]	OGSA-BES-WG, JSDL-WG.	Mature
	Choreography, Orchestration and Workflow[v]	OASIS/BPEL, OGSA Workflow Design Team	Mature
	Resource Virtualization[v]	GridVirt-WG, CDDLM-WG	Evolving
	Information Model[v]	DMTF/CIM, GLUE-WG	Mature
	CPU Scavenging[vii]	Proprietary Solutions Exist	Mature
	Legacy Programs[v]	ACS-WG	Evolving
	Reservation[v]	GRAAP-WG, GSA-RG	Evolving
Data	Data Movement	DMI-WG, Grid-FTP	Evolving
	Data Access	GFS-WG and DAIS-WG	Mature
	Data Integration	DAIS-WG	Evolving
	Data Management	Storage Network-CG, OGSA-Data-WG	Evolving
	Data Provisioning[v]	Continuation of EGA data work, OGSA-Data	Gap
	Metadata[v]	OASIS/WSRF-RMD,	Evolving
Application Development	Application Debugging[iii]		Gap
	Application APIs[xii]	SAGA-WG,GridRPC-WG,GridCPR-WG,DRMAA-WG	Mature
Foundations	Communication Protocols[iii]	HTTP/SOAP,	Mature
	Architecture[iii]	Reference Model-WG, OGSA-WG	Mature
	Grid Semantics[ix]	Semantic Grid-RG	Research
	Grid Fabric [v]	OASIS/WSRF, NM-WG, NML-WG	Mature
System Properties	Fault Tolerance[x]	Implementation Property	Mature
	Load Balancing[xi]	Implementation Property	Mature
	Failure Recovery[v]	Implementation Property	Evolving
	Self-Management[v]	Implementation Property	Research

6. Security Considerations

All OGF documents must have this section. With respect to this document, it would be a serious omission if security specifications were not part of the OGF short term roadmap and an identified priority. Noting that this is the case, meets the requirement that this document address security in a way consistent with the nature of the document.

7. Contributors

The Editors listed on the title page are those committed to taking permanent stewardship for this document – receiving communication in the future and otherwise being responsive to its content. Their contact information is provided below:

Dr. David Snelling
Fujitsu Laboratories of Europe
Hayes Park, UB4 8FE
David.Snelling@UK.Fujitsu.com
+44-208-606-4649

Chris Kantarjiev
Oracle Corporation
400 Oracle Parkway
Redwood Shores, CA 94065
Chris.Kantarjiev@Oracle.com
+1 650 607 5521

Other contributors include all members of the Technical Strategy Committee, namely Steven Newhouse, Andre Merzky, Geoffery Fox, Robert Fogel, Hanoch Eiron, Ian Foster, Tom Maguire, and Joel Replogle. The TSC would also like to thank Mark Linesch and the chairs of a number of working groups who's documents provided much needed perspectives and content.

8. Intellectual Property Statement

The OGF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the OGF Secretariat.

The OGF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights, which may cover technology that may be required to practice this recommendation. Please address the information to the OGF Executive Director.

9. Disclaimer

This document and the information contained herein is provided on an "As Is" basis and the OGF disclaims all warranties, express or implied, including but not limited to any warranty that the use of the information herein will not infringe any rights or any implied warranties of merchantability or fitness for a particular purpose.

10. Full Copyright Notice

11. References

1. Definition “Strategy” from Wikipedia, http://en.wikipedia.org/wiki/Strategy, August 2006.
2. A. Barbir, M. Gudgin, M. McIntosh, and K.S. Morrison (ed.): Basic Security Profile Version 1.0, Web Services Interoperability Organization, Working Group Draft, 29 August 2005. http://www.ws-i.org/Profiles/BasicSecurityProfile-1.0-2005-08-29.html.
3. Thompson, M., et al., "Certificate-based Access Control for Widely Distributed Resources," in Proc. 8th Usenix Security Symposium. 1999.
4. Pearlman, L., V. Welch, I. Foster, C. Kesselman, S. Tuecke, "A Community Authorization Service for Group Collaboration," Proceedings of the IEEE 3rd International Workshop on Policies for Distributed Systems and Networks, 2002.
5. Chadwick, D.W., O.Otenko, " The PERMIS X.509 Role Based Privilege Management Infrastructure", Proceedings of 7th ACM Symposium on Access Control Models and Technologies (SACMAT 2002).
6. "VOMS Architecture v1.1," http://grid-auth.infn.it/docs/VOMS-v1_1.pdf, February 2003.
7. “Reference Model and Use Cases”, The Enterprise Grid Alliance, www.gridalliance.org/en/WorkGroups/ReferenceModel.asp, March 2006.
8. I. Foster, H. Kishimoto, et al, “Open Grid Services Architecture V1.5”, GFD.80, Open Grid Forum, http://www.ggf.org/documents/GFD.80.pdf, July 2006.
9. K. Ballinger, D. Ehnebuske, C. Ferris, M. Gudgin, C.K. Liu, M. Nottingham, and P. Yendluri (ed.): Basic Profile Version 1.1, Web Services Interoperability Organization Final Material, 24 August 2004. http://www.ws-i.org/Profiles/BasicProfile-1.1.html
10. I. Mandrichenko, W. Allcock, T.Perelmutov, GridFTP v2 Protocol Description, GDF.47, http://www.ggf.org/documents/GFD.47.pdf.
11. I. Foster, D. Gannon, H. Kishimoto, Jeffrin J. Von Reich, Open Grid Services Architecture Use Cases, GDF.29, http://www.ggf.org/documents/GFD.29.pdf .

[i] “Distributed operation implies a need to interoperate with and manage multiple security infrastructures.” (from: OGSA use case matrix)

[ii] “Many use cases require applications to be deployed on the other side of firewalls from the intended user clients. Inter-Grid collaboration often requires crossing institutional firewalls. OGSA needs standard, secure mechanisms that can be deployed to protect institutions while also enabling cross-firewall interaction.” (from: OGSA use cases matrix)

[iii] dictionary definitions

[iv] See Authorisation Glossary under AAA, GFD-I.042

[v] See OGSA Glossary of Terms v1.5, GFD-I.081

[vi] Resource Brokering is provided by a Brokering Service (see: OGSA use case matrix term “Brokering Service”).

[vii] “An important tool for an enterprise or VO to use to aggregate computing power that would otherwise go to waste. How can OGSA provide service infrastructure that will allow the creation of

applications that use scavenged cycles? For example, consider a collection of desktop computers running software that supports integration into processing and/or storage pools managed via systems such as Condor, Entropia, United Devices, etc. Issues here include maximizing security in the absence of strong trust." (from: OGSA use case matrix)

[viii] Programming language based APIs for Grid enabled applications

[ix] "Semantic Web technologies for Grid users and developers." (from: SEM-RG, Semantic Grid RG charter: http://forge.gridforum.org/sf/projects/sem-rg)

[x] "Support is required for fail-over, load redistribution and other techniques used to achieve fault-tolerance." (from: OGSA use case matrix)

[xi] "The GRID monitors the job performance and adjusts allocated resources to match the load and fairly distributes end users' requests to all the resources." (from: OGSA use case matrix)

[xii] Programming language based APIs for Grid enabled applications