Open Science Grid

Document Name	**A Blueprint for the Open Science Grid**
Version	**Snapshot v0.6**
Date last updated	Sept 12th, 2004
OSG Activity	Blueprint

This document is being prepared through consensus of the participants in the Blueprint Activity, and subject to review by a Review-Circle. Since the document is being updated constantly, the latest version is accessible from http://www.opensciencegrid.org/documents/.

V0.6	9/12/04		Post Blueprint meeting
V0.5	9/6/04		Prepare for Blueprint meeting
V0.4.2	8/15/04		Distributed to Joint Committees
V0.4.1	8/08/04	Additions	Ian Fisk, Rob Gardner, Ruth Pordes
V0.4.0	8/01/04	Comments and changes	Wyatt, Ruth
V0.3.5	7/30/04	Comments and changes	Abhishek Rana, Jerome Lauret, Conrad Steenberg, Frank Wuerthwein
V0.3.0	7/19/04	Tidy up	
V0.0.0	7/15/04	Blueprint face to face	Distributed to Review Circle who attended phone call

1 Introduction

The Open Science Grid Consortium will build a sustained production national infrastructure of shared resources, benefiting a broad set of scientific applications. The organization and framework for the consortium is described at `http://www.opensciencegrid.org`. This Blueprint for the Open Science Grid provides the guiding principles and roadmap for the building and operation of the infrastructure and will provide a basis for planning a coherent technical program of work. The Blueprint does not provide the actual plan or decisions on technologies for implementation. The Open Science Grid Consortium will work through a set of self-organized Activities. Writing, editing and evolving this document is one of the first such Activities. The current document is a ‘snapshot’ and will subsequently evolve into the Blueprint document after intermediate draft versions.

The OSG infrastructure is being built and deployed through a set of Activities, each of which involve some or all of the participants in the Consortium. A series of Activities OSG-0, OSG-1 etc will build iterative releases of the infrastructure which will be usable by and supported for a set of applications defined within the Activity. Within each Activity there are a dynamic and evolving set of participants, applications, services, and resource providers; making contributions to building and use of the OSG-N infrastructure is flexible and subject to ongoing negotiation with the associated activity, it is not statically defined at the start.

In this document, first the fundamental definitions within the scope of OSG are presented, followed by principles and requirements. Next, selected issues are exposed through building up a set of simple use cases; then a discussion of some (but not all) aspects of the architectural decomposition is included; followed by sections on specific end-to-end components – to date Grid Lifecycle, Security and Operational Infrastructures. The document then makes an initial list of Services.

The Open Science Grid infrastructure relies on many diverse projects (research, development, design, operations) and groups who may be participants in the Open Science Grid Consortium but whose projects are outside the boundary of the organization’s framework itself. This Blueprint takes account of this structure and in general refers to the documents of these projects rather than duplicate the information here.

The Blueprint is guided by overarching principles to make the infrastructure – both conceptually and in practice – as simple and flexible as possible, to build from the bottom up a system which can accommodate the widest practical range of users of current Grid technologies, in a context which maximizes the future convergence of those users to greater commonality in technology choice. The production quality of the infrastructure leads to additional requirements and principles in support of sustained and robust operations.

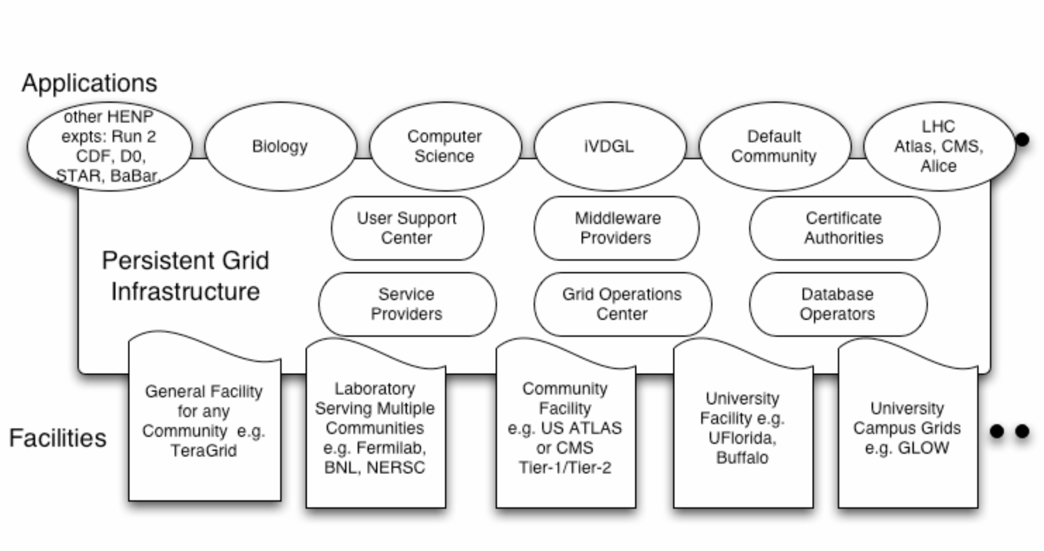
Applications
other HENP expts: Run 2 CDF, D0, STAR, BaBar,
Biology
Computer Science
iVDGL
Default Community
LHC Atlas, CMS, Alice
Persistent Grid Infrastructure
User Support Center
Middleware Providers
Certificate Authorities
Service Providers
Grid Operations Center
Database Operators
Facilities
General Facility for any Community e.g. TeraGrid
Laboratory Serving Multiple Communities e.g. Fermilab, BNL, NERSC
Community Facility e.g. US ATLAS or CMS Tier-1/Tier-2
University Facility e.g. UFlorida, Buffalo
University Campus Grids e.g. GLOW

Figure 1: The Open Science Grid

2 Definitions

The basic terms are defined within the scope of the Open Science Grid. An attempt has been made to define a useful set of simple definitions upon which the end to end infrastructure can be built. Definitions that follow dictionary definitions and standard usage are not repeated here.

- **User** – A person who makes a request of the Open Science Grid infrastructure.
- **Resource Owner** – has permanent specific control, rights and responsibilities for a Resource associated with ownership.
- **Agent** – A software component in OSG that operates on behalf of a User or Resource Owner or another Agent.
- **Consumer** – A User or Agent who makes use of an available Resource or Agent or Service.
- **Provider** – Makes a Resource or Agent or Service available for access and use.
- **Ownership** – A state of having absolute or well-defined partial rights and responsibilities for a Resource depending on the type of control. OSG considers two such types: actual Ownership and Ownership by virtue of a Contract/Lease. A Lessee is a limited Owner of the Resource for the duration of the Contract/Lease.
- **Service** – A method for accessing a Resource or Agent.
- **Site** – A named collection of Services, Providers and Resources for administrative purposes. A Facility is a collection of Sites under a single administrative domain.
- **Virtual Organization** – A dynamic collection of Users, Resources and Services for sharing of Resources (Globus definition). A VO is party to contracts between Resource

Providers & VOs which govern resource usage & policies. A subVO is a sub-set of the Users and Services within a VO which operates under the contracts of the parent

- **Virtual Site** is a set of sites that agree to use the same policies in order to act as an administrative unit.

- **Dynamic Workspace** – A persistent, extensible, managed collection of objects and tools hosted on a grid.

- **Policy** – A statement of well-defined requirements, conditions or preferences put forth by a Provider and/or Consumer that is utilized to formulate decisions leading to actions and/or operations within the infrastructure.

- **Contract** – Agreement between Consumer(s) and/or VO(s) and/or Provider(s) expressed through Policies. Simplest contract is a consumer-provider match based on their policies.

- **Delegation** – An entrustment of decision-making authority during transfer of request for work or offer of resources from a User or Agent to another Agent or Provider, or vice versa. The latter is provided with a well-defined scope of responsibility and privilege at each such layer of transfer of request or offer.

- **Economy** – Set of benefits made and costs accrued as seen by Consumers and Providers.

- **Security** – Control of and reaction to intentional unacceptable use of any part of the infrastructure.

- **Grid** – A named set of Services, Providers, Resources, and Policies, overlapping and/or including other Grids operating as a coherent infrastructure in support to the contracting Virtual Organizations. Providers may delegate their contracts with the participating VOs to the Grid administration.

Referenced Definitions:

- **Namespace -** `http://en.wikipedia.org/wiki/Namespace`

- **Resource -** Item 2 and 5 at `http://dictionary.reference.com/search?q=resource`

Notes:
There are approximate pairs of definitions that correspond to each other: User/Owner and Consumer/Provider. These pairs are not perfectly symmetric as User strictly refers to a person while Owner generally refers to an institution. There is some symmetry at the agent level such that both members of a pair delegate to engage in contracts in order to achieve their 'economic objective' within their expressed policies.

2.1 The Open Science Grid

The Open Science Grid (OSG) is the grid under the governance of the Open Science Grid Consortium operated as a sustained and production infrastructure for the benefit of the Users. Other grids may operate under the governance of the OSG Consortium, for example the grid that validates the infrastructure before it becomes the OSG. The Open Science Grid includes facility, campus, and community grids that participate in the Consortium; The Open Science Grid interacts with grids external to the Consortium through federation and partnerships.

The Open Science Grid VO is open to those Users and VOs that have contracts with the OSG.

3 Principles, Best Practice and Requirements

Principles are basic rules and guidelines that govern (guide and influence) the fundamental aspects of the model, methods and architecture.

Best Practices are guidelines to be adhered to, as much as is possible, in practice.

Requirements will be formal statements (we are not there yet) that provide goals and constraints on the designs and implementations. Requirements affect functional aspects of the architecture, and can be presented through a set of Use Cases. The goal is to have a minimal set of requirements for participation in the OSG infrastructure.

The Principles, Best Practices and Requirements are not necessarily targeted for initial deployments of OSG. They are directed towards the long-term goals and requirements for the final infrastructure.

3.1 Principles

Principles are intended to apply to end-to-end use cases as well as the common infrastructure. For example, they are meant to be applied to the error handling, monitoring, information, security and management infrastructures, as well as the services and applications.

The OSG infrastructure must always include a phased deployment, with the phase in production having a clear operations model adequate to the provision of production-quality service.

Policy should be the main determinant of effective utilization of the resources. This implies that without governing policy there would be full utilization of the resources.

The OSG architecture will follow the principles of symmetry and recursion.

Services should work toward minimizing their impact on the hosting resource.

Services are expected to protect themselves from malicious input and inappropriate use.

All services should support the ability to function and operate in the local environment when disconnected from the OSG environment. This implies the local environment has control over its local namespace.

OSG will provide baseline services and a reference implementation. Use of other services will be allowed.

The OSG infrastructure will be built incrementally. The roadmap must allow for technology shifts and changes.

Users are not required to interact directly with resource providers.

The requirements for participating in the OSG infrastructure should promote inclusive participation both horizontally (across a wide variety of scientific disciplines) and vertically (from small organizations like high schools to large ones like National Laboratories).

VOs that require services beyond the baseline set should not encounter unnecessary deployment barriers for the same.

3.2 Best Practice

The OSG architecture is Virtual Organization based. Most services are instantiated within the context of a VO.

Services may be shared between VOs. It is the responsibility of the Service and Resource Providers to manage the interacting policies and resources.
Resource providers should provide the same interface to local use of the resource as they do to use by the distributed services.

Every service will maintain state sufficient to explain expected errors. There shall be methods to extract this state. There shall be a method to determine whether or not the service is up and useable, rather than in a compromised or failed state.

The OSG infrastructure will support development and execution of applications in a local context, without an active connection to the distributed services.

The infrastructure will support multiple versions of services and environments, and also support incremental upgrades.

The OSG infrastructure should have minimal impact on a Site. Services that must run with superuser privileges will be minimized.

System reliability and recovery from failure should guarantee that user's exposure to infrastructure failure is minimal.

Resource provider service policies should, by default, support access to the resource. The principle 'services should protect themselves' thus implies that services should additionally have the ability to instantaneously deny access when deemed necessary.

Allocation and Use of a Resource or Service are treated separately.

Services manage state and ensure their state is accurate and consistent.

3.3 Requirements

Published information from resource providers, sites and services must be accurate.

All services must be (recursively) discoverable by the OSG discovery service. Registration implies name, contact identifier and other specific information.

Users, resources and service providers must accept the OSG Acceptable Use Policy. Services which receive delegated credentials additionally agree to be honest stewards.

A User must be a member of at least one participating organization (at least for the time being).

A service must be offered to at least one VO.

The minimal requirements for participating in the OSG will be: the ability to advertise services in the common infrastructure; to accept use of one or more resource by applications running on the infrastructure; and to abide by the security requirements.

A minimal requirement on a Site is to provide some resources for OSG services and transient storage space for any job input and output. The amounts required for useful participation will evolve.

VOs, Sites and service providers will need to cooperate in order to permit the tracing of each transaction to a responsible user. (May not be the original user but a VO administrative user for example).

Policy of a resource provider takes precedence over the policy of a site which takes precedence over the policy of a VO which takes precedence over the Workspace (or sub-VO) policy.

3.3.1 Resource Providers & Sites

Sites can act as an administrative unit for: contracts with VOs; resource management and allocation; and providing services shared between VOs.

Sites may support a subset of the infrastructure, services and types of resource. A site should advertise its capacities and capabilities.

Sites must provide at least the well-defined set of OSG minimum services.

Sites need to be able to trace the responsible User when accessed.

Sites may deny access to a particular User and/or a VO based on security as well as contract and policy constraints. Permanent and durable storage space is provided by agreements between a VO and one or more Sites.

3.3.2 Virtual Organizations and Dynamic Workspaces

Sub-VOs operate under the context (contracts and policies) of the parent VO.

The execution environment is the responsibility of and within the scope of the VO and/or the Dynamic Workspace.

A VO must support use of VO based Dynamic Workspaces to the level of single transactions.

Validation of the infrastructure is the responsibility of the VO for their particular applications.

Resources and services can be shared by, and transferred between, VOs and Dynamic Workspaces.

VOs may have latency requirements (as well as performance requirements).

4 Discussions

4.1 Namespaces

A namespace is a collection of names in which all names are unique within their semantic groups. Names in a distributed system can be organized in namespaces, which can be represented as directed graphs. The process of looking up a name is known as name resolution, and a knowledge of how and where to start resolution is generally referred to as closure mechanism.

An ideal namespace management scheme is expected to rely not on maintaining globally unique absolute names, but rather on schemes that exploit the relative uniqueness of names in the local namespaces.

OSG will consider various namespaces and their management. Namespaces may be defined by and potentially shared between any entity in a grid or VO.

Each Service potentially has namespace scope and responsibilities to manage. E.g. Physical – Device level; Logical – within the VO; User – meta-data driven.

OSG Namespace Requirements:

Data (files) registered to a Grid are identified by a unique identifier within the GUID[1] namespace.

The "opensciencegrid" namespace is available for use by Services, Providers. VOs and Sites through a contract with the Open Science Grid consortium. Request for and review of such names is through the appropriate Technical Group.

4.2 Data Management

Data is stored in named containers, which can be nested and which are registered to the grid as Files. Files are given a unique identifier in the GUID namespace. OSG is agnostic on the question of the mutability of containers.

A Physical File Name (PFN) is the storage location of a file. The name identifies a storage resource and location in which the data is stored. The name must allow the Storage Element and the name of the file as stored to be identified.

A Logical File Name (LFN) is unique within the defined namespace. A Logical File Namespace is generally defined and managed within the scope of a VO. VOs may share LFN namespaces. Logical File Names are human readable and normally structured with the syntax of a unix file path and name.

[1] UUIDs/GUIDs, ISO/IEC 11578:1996 http://www.iso.ch/cate/d2229.html, or DCE 1.1: Remote Procedure Call http://www.opengroup.org/publications/catalog/c706.htm

The Replica Catalog Service provides management of files registered with the grid. The Replica Catalog Service maps a file namespace (either the GUID or an LFN namespace) to the Physical File Names of replicas of the file contents.

It stores the declared GUID or LFN together with the initial PFN, access control information for the file, and a checksum associated with the file contents. As the file is replicated or moved to other storage resources, the Replica Catalog Service maintains the mappings to the Physical File names of replicas.

It is assumed that replicas contain identical data.

Storage Management

VO's contract for storage space with storage resource providers (ranging from guaranteed to opportunistic use). A site providing storage for multiple VOs may manage the resources as a common service with common policies and operational procedures and dynamic mechanisms for resource sharing and allocation. These are transparent to the User (and the VO?)

A Storage Service maintains its own namespace .A Storage Element must provide services to achieve the necessary mapping between its local resource namespace and the logical and physical namespaces of the files managed by the Replica Catalogs.

The Open Science Grid supports a Logical File Namespace that may be used by any User or VO using the OSG. The semantics of the namespace is allows uniqueness of the LFN within OSG.

The Open Science Grid will also provide an LFN Mapping Service to map an LFN structure between OSG and a grid it is federating with (e.g. Teragrid)

The Open Science Grid provides a Replica Catalog Service which any user or VO may use.

4.3 Ownership and Leases

A Consumer can sublet a resource to another Consumer. This transfers allocation of the resource & appropriate privileges, subject to policy and contracts, to another Consumer, but does not transfer Ownership.

4.4 Discovery Service

There is a need for a top-level Discovery Service, the main functions of which will be: (a) given a kind of service, return a list of service instance references; (b) given a service instance name, return a service instance reference. Discovery Service will operate hierarchically.

4.5 Architecture of a Service

Some of the OSG principles and best practices affect the architecture of each Service.

Services can enhance robustness through self-management and monitoring – for example, ensuring that if the service crashes it is automatically restarted.

The Replica Catalog Service must be distributed such that there are local catalogs well connected to the Storage Elements.

No service should present a single point of failure.

5 Use Cases

We plan to describe simple use cases to expose details of the principles, architectures and larger goals. This will take several iterations:

Use cases[2]:	
	File sharing
	VO Software Installation at a site
	VO Service Installation at a site
	Large scale file production at multiple sites with input of exe/control info from user resource
	Large scale file production at multiple sites with input of exe/control info/data from VO SE
	Verification operation at a specific site for a specific VO application
	Large scale file movement from a VO SE to another VO SE or to a user resource

5.1 File Sharing

User1 creates a File and wants to provide its Contents to User2 via the OSG.

User1 establishes or uses a shared VO/Dynamic Workspace that includes both User1 and User2. The VO/Dynamic Workspace has or is populated with the necessary resources, policies and contracts needed to support file sharing on OSG.

User1 creates a file in the local file system, “Publishes”[3] it to the OSG infrastructure, obtaining a unique reference, and is then at liberty to delete it from the local file system.

```
        echo "Hello World" > ~/Foo
        myPublishService = "discover the publishing service for UserVO"
        returnCode = myPublishService.Publish(~/Foo,Boo)
        #"Boo" now exists in the "default" namespace of the UserVO
        rm -f ~/Foo
```

User1 contacts User2 and says "check out Boo" that is published via the Publishing Service "UserVO" which User1 can initially find via the discovery service of "osg".

User2 gets the file and reads the contents.

[2] file coordination requirements as mods to the use cases below: e.g. requirements to restrict production sites based on existence or access status of exe/control info/data at the site; requirements to deliver inputs or outputs with an ordering algorithm; requirements to execute sequential operations on files within and across production sites (simulation chains, merging, verification).

[3] It is clear that “Publish” is in architecture, design and implementation, the concatenation of many separate services, each of which will need to support many use cases besides this one.

```
myPublishService = "discover the publishing service for UserVO1"
returnCode = myPublishService.Get(Boo,/tmp/Loo)
cat /tmp/Loo
"Hello World"
```

Implicit in this use case is the fact that `myPublishService.Publish` not only registers the file but also stores the file using some storage service.

This use case addresses namespace management issues involved in remote file access between different users working on their individual local file systems. This is accomplished by 2 OSG Services: (a) Publish Services (PSs), (b) a Discovery Service [sec 4.3] to find an instance or more of PSs.

The PS conceptually fulfills the closure mechanism [sec 4.1] involved in resolution of distributed names in OSG. Figure 2 shows the flow of requests for this use case, and provides a logical view of how the PS namespace is used to manage different namespaces of the file owner, file consumer, and the storage services. It is the responsibility of the PS to use various Catalog Services (Metadata, File, and Replica) during this process. On receiving a ‘publish file’ request from User1 that contains a filename chosen (BOO in the figure) by this user, PS requests Metadata Catalog to register necessary metadata of the file. Thereafter, PS registers the file with the File Catalog using a name that is unique in the PS namespace. There is a one-to-one namespace correspondence between the PS and the File Catalog. Using the Replica Catalog, PS registers the file as a mapping to the storage element names and the filenames on each storage service.

In addition, there is a Reliable File Transfer Service (RFTS) that guarantees reliability and QoS, and works with the other services.

It is assumed that, in reality each referenced service will depend on and reference other services in any implementation.

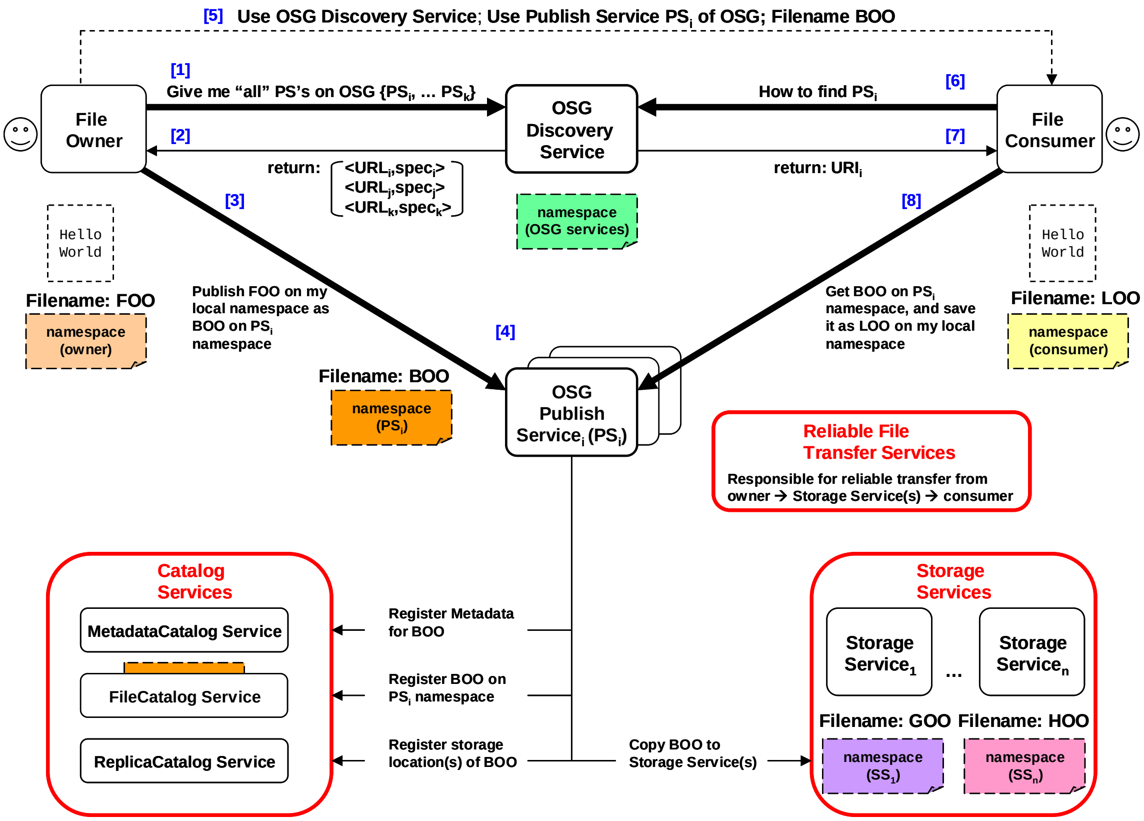
[5] Use OSG Discovery Service; Use Publish Service PS_i of OSG; Filename BOO
[1]
Give me “all” PS’s on OSG {PS_i, … PS_k}
File Owner
[2]
return: <URL_i,$spec_i$> <URL_j,$spec_j$> <URL_k,$spec_k$>
OSG Discovery Service
How to find PS_i
[6]
[7]
return: URI_i
File Consumer
namespace (OSG services)
[3]
[8]
Hello World
Filename: FOO
namespace (owner)
Publish FOO on my local namespace as BOO on PS_i namespace
[4]
Get BOO on PS_i namespace, and save it as LOO on my local namespace
Hello World
Filename: LOO
namespace (consumer)
Filename: BOO
namespace (PS_i)
OSG Publish $Service_i$ (PS_i)
Reliable File Transfer Services
Responsible for reliable transfer from owner → Storage Service(s) → consumer
Catalog Services
MetadataCatalog Service
FileCatalog Service
ReplicaCatalog Service
Register Metadata for BOO
Register BOO on PS_i namespace
Register storage location(s) of BOO
Copy BOO to Storage Service(s)
Storage Services
Storage $Service_1$
…
Storage $Service_n$
Filename: GOO
Filename: HOO
namespace (SS_1)
namespace (SS_n)

Figure 2: Publish File Use case (with Discovery Service and Publish Service)

The services exposed are both those offered as part of the OSG software stack and also VO provided services. (Further exploration of this use case is also available at `http://www.ppdg.net/pa/ppdg-pa/blueprint/files/OSG_PublishService.ppt`, and will be moved to a separate document later).

Figure 2b shows the expected cardinalities in the deployment of the catalogs, the Publishing Service, and the storage system elements. These are not expected to remain the same, since implementation may demand a different level of multiplicity. In the figure, **`1..*`** denotes one or many objects, **`1`** denotes a single object, following standard UML conventions. The cardinalities involved in association of PSs with VOs need to be further explored in future discussions.

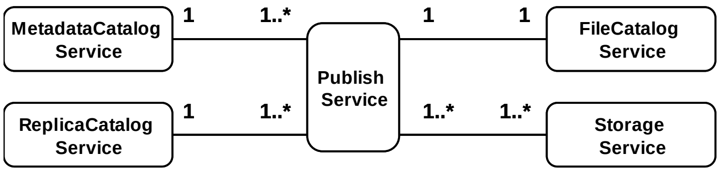

Figure 2b: Multiplicity of Association for PS, Storage and Catalog Services

Responsibilities of various actors/components in this Use case:

Key:

- responsibility particular to this actor.

o responsibility general to either services or consumers

? responsibility necessary but so far not determined

User1: File Owner:

- Select which Publish Service to use from available set (might outsource this to some selection agent given criteria).
- Negotiate contract with Publish Service (myPS).
  get current service offering definition from myPS.
  check that SLA offered by myPS is acceptable.
  broker SLA complaints from User2.
- Maintain User2's Access Control Policy (ACP) for the file stored in myPS.
- Communicate with User2 name of file ("Boo") and name of myPS.

o Get authentication (authN) and authorization (authZ) tokens sufficient for requests made of myPS.

o Abide by Acceptable Use Policy (AUP).

Use2: File Consumer:

o Get AuthN and AuthZ tokens sufficient for request of myPS (and any other service).

o Abide by Acceptable Use Policy (AUP).

Publish Service:

- Maintain namespace consistency (for space containing "Boo")
- Maintain the link between nameBoo and name in storage service namespace ("Goo").
- Maintain the link between nameBoo and User1's ACP.
- Provide method for transfer of ownership.
- Provide method for User1's revision of ACP (unless ACP storage is outsourced).
- Maintain contact method and service definition information in Discovery Service.
- Negotiate contract with Storage Service sufficient to meet its SLA requirements.

o Meet SLA (including participate in problem resolution method).

o Authenticate and authorize (AA) User1 sufficient for the request.

o Authenticate and authorize User2 sufficient for the request.

o Provide informative error messages back to failed requests.
??Request sufficient AA tokens if missing. ??

Storage Service:

- Maintain namespace consistency (for space containing "Goo").
- Maintain the link between nameGoo and the physical storage name(s).
- Enforce ACP specified in contract with myPS.
  (perhaps not only give files to myPS ?)

? Maintain the Acceptable Use Policy (??)

- Maintain the Privacy Policy.

o Meet SLA including participate in problem resolution method:
Specify what level of reliability, protection from loss, etc. are promised.

Specify what level of integrity checking is performed.

- o Authenticate and authorize (AA) myPS sufficient for the request.
- o Provide informative error messages back to failed requests.

Discovery Service:

- - Maintain namespace of services.
- - Maintain a link between namemyPS and its contact methods and service definition.
- - Describe organization principle of returned matches between queries and service.
- - Provide contact method description and service definitions for services matching a request.
- ? Would hierarchies of grids come about by levels of discovery services.?
- o Provide informative error messages back to failed requests.
- o Defend service against attacks aimed at:
  - Overflow of namespace entry size.
  - Overflow of namespace.
  - Dilution of namespace with bogus entries.

Questions:

Owner must tell Consumer the name of the file and which PS to use.

The PS (may have VO identity) and the storage service must work on access controls. Out-of-band access? What if someone came in with the physical name of the file and accessed it outside of the knowledge/control of the publish service? Does PS pass policy down to the Storage Service to have the latter enforce it.

Add a use-case where the owner comes into the picture to change the access rights on the file, and how this works in the PS and Storage Service interactions/relationship.

Selection of PS based on throughput (network location and capabilities).

What is the dividing line between Publish and Storage Service responsibilities. Scenarios relating storage services and lifetimes.

Consumer access to Storage goes through PS, should this be only on the way to get to Storage?

Is there too much state accreting in Publish Service? Much of the PS activity is to coordinate amongst other services, like File Catalog Service and Replica Catalog Service.

Acceptable Use Policies - do Storage Services individually have them? How could they be enforced in real-time? What can be done in real-time would be structural issues like secure use of certificates and privacy (of files)?

The name of the Storage Service name must be unique in some namespace. It has not been addressed at this point whether this is guaranteed by virtue of a URI explicitly, or via the Discovery Service's namespace. Also, the implications of multiplicity in the relationship of PS to Replica and Metadata Catalogs, needs to be addressed in future. Specifically, it needs to be decided how the mapping in these two catalogs is guaranteed to be unique if more than one PS share the same Replica and Metadata Catalog. It would conceptually suffice if the same mechanism as used in guaranteeing the uniqueness of the Storage Service name is employed here. The Replica Catalog would thus be a map of objects of the type `Storage_Service:filename`.

Overwriting/modifying of a file is not addressed by this use case.

We also have not discussed how two different Publish Services may synchronize themselves. This is clearly another important use case given an OSG principle that all services should function in a local environment, disconnected from the OSG.

6 Architectural Decomposition

This section includes a set of sketches to explore the architectural decomposition and it will grow with interface and service definitions, and dependencies as we proceed.

6.1 Basic OSG Components

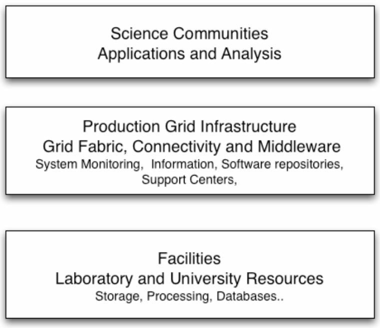
Science Communities
Applications and Analysis
Production Grid Infrastructure
Grid Fabric, Connectivity and Middleware
System Monitoring, Information, Software repositories,
Support Centers,
Facilities
Laboratory and University Resources
Storage, Processing, Databases..

Figure 3: OSG Architecture

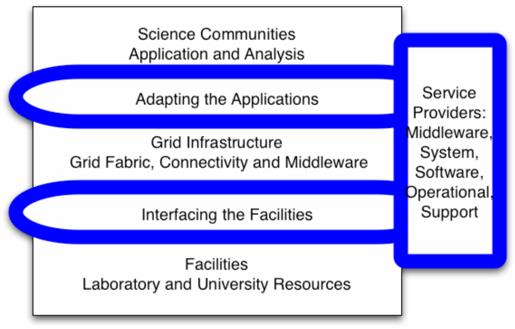

Figure 4: Missing Capabilities

6.2 Symmetry & Recursion relating Users, Resources, and VOs

OSG aims to federate across heterogeneous grid environments, large-scale distributed enterprises and communities. To facilitate this task, the OSG infrastructure views VOs as recursively-defined entities comprising of users, resources, and sub-VOs [see appendix]. The different ways a VO can be formed is shown in Figure 5. In this figure, users and resources organize themselves as VOs in order to enter into contracts resulting from negotiations based on their respective sets of policies. These contracts are manifested at the middleware level as matchmaking, and the related services are provided by the VOs.

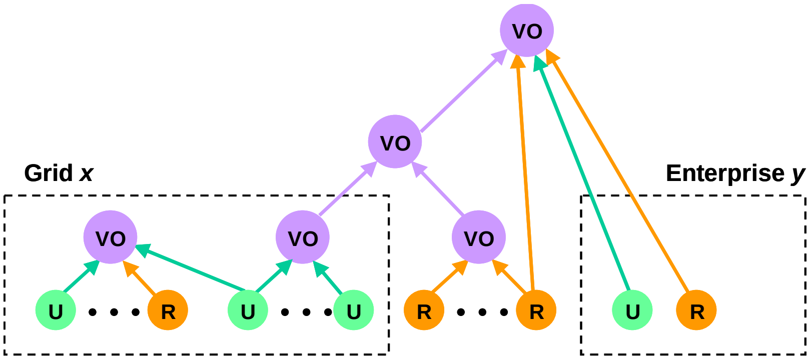

Figure 5: VO Hierarchy and Recursive VO Formation in OSG

VOs may choose to enter into sub-contracts in order to more effectively utilize their resources or better satisfy their users. For sake of simplicity, agents and services are not shown along with VOs in this figure. A VO can be solely a resource-provider or consumer or both. Figure 5b shows symmetry in this relationship by considering a typical flow of request from a user to a resource owner (via a resource provider). This figure takes into account Agents with delegated rights and policies, communicating and working together to establish end-to-end functionality. Users, Providers and Agents play roles of producers and consumers depending on the direction of workflow being considered. However, the conventional nomenclature for this role has been followed (the bold line in the figure) throughout this document. Policy representation and policy reconciliation generally implies delegation of responsibility in such a heterogeneous and dynamic environment. (This delegation may or may not include forwarding of identity and role of the user and/or resource. E.g., the cache management system of a VO generally can not be required to know which user requested what data movement as files in cache are used by more than one user. On the other hand, access to a user’s quota does of course require a user’s identity/role to be forwarded.)

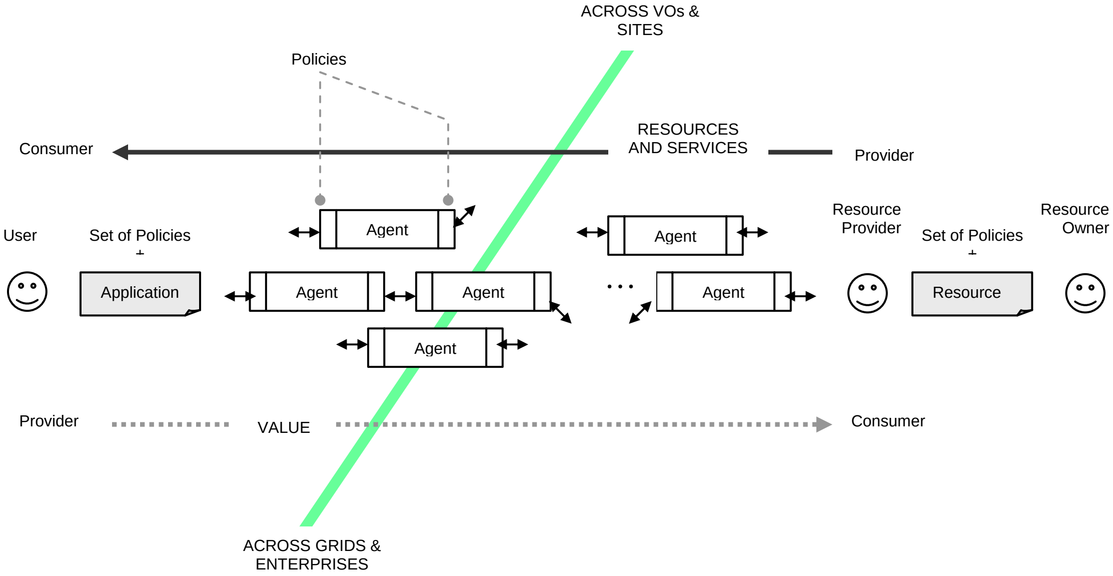

Figure 5b: Symmetry between Consumers and Providers

Each functional level in this model may have the capability to monitor its appropriate use. To make this relationship fault-tolerant, OSG may explore looking into error recovery and rollback mechanisms that would allow a workflow request to trace back by following only a limited number of steps.

6.2.1 Relationship between VOs, Grid Infrastructure, and Sites/Facilities

As a federation of grids, OSG infrastructure considers VOs and Sites to be dynamically associated with one another as shown in Figure 6.

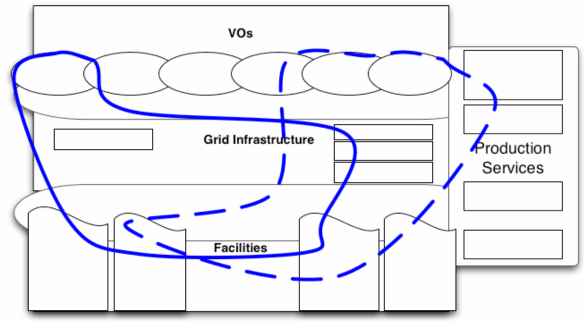

Figure 6: VO Environments

This is made possible at an operational level by timed leases of Resources by Sites to the consumer VOs. Once party to a contract, a consumer VO takes the responsibility to dynamically deploy VO-specific services on Sites for the period of the lease. OSG will provide a persistent grid services layer and service specifications to guarantee interoperability, as well as reference implementations for those services. This includes both services provided by Sites as well as VOs. Both the Site and the consumer VO have the freedom to do monitoring and accounting in such an environment.

In the above mentioned symmetry in OSG architecture, it is important *where* a *decision* is made. Distributed systems fundamentally should allow components to have as little knowledge as suffices the need. Robustness, however, is dependent on effectual error-propagation and thus decision-making points. There is a trade-off involved since too many decision-making junctions in the workflow route may become an overhead.

6.3 Job and Data Management

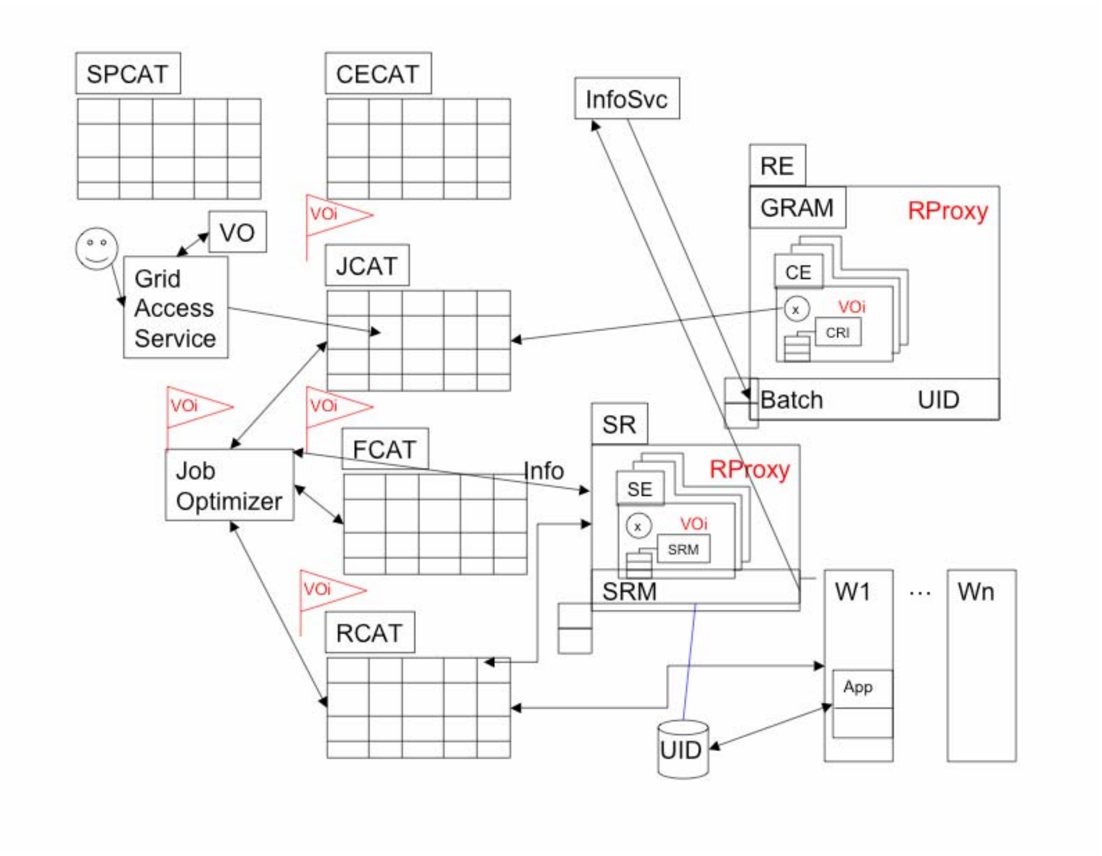

Figure 7: Job and Data Management Components

There was consensus to accept this architecture as a baseline model for review.

CE	Compute Element
CRI	Interface to Compute Resource (Condor-G)
GRAM	Interface to Compute Element
Job Optimiser	looks at list of files, decides which SR to use based on replica locations, to minimize data movement for instance. A VO can feed information from the SE (VO specific) to the Job Optimizer to tune this picture appropriately. Information also can flow from the SE to the SR.
RE	Resource
RProxy	Remote Proxy. Adapts from grid to local infrastructure (e.g. translate for private network)
SE	Storage Element
SR	Storage Resource
SRM	Interface to Storage Resource
CECAT	Compute Allocation Catalog
JCAT	Job Catalog. Has rules to control who is next in the compute center (RE). The RE can push out information that can influence the decision made by the JCAT

	who is going next... perhaps based on availability of RE's required services.
RCAT	Replica Catalog (a Reliable File Transfer Job Queue)
SPCAT	Space Allocation Catalog
WN	Worker Node
VOi	Possibility for VO-specific CE and/or SE implementations.

The architecture sketch depicted in Figure 7 is based on the notion that job & data management are conceptually symmetric, especially at the level of the job optimizer. In both cases, a VO leases resources from sites. It maintains catalogues of available and requested resources, and matches them based on policy driven optimization of workload throughput. This matching takes into account co-location of data and CPU as needed.

The architecture places minimal requirements on the sites. The responsibility for providing functionality is shifted to the VOs as much as possible. The latter is motivated by the notion that VOs are by definition internally cohesive wheras sites are distinct and may generally differ in a variety of ways.

In the following we first discuss some of the details alluded to in Figure 7, and then list a number of broad topics that require further discussion.

6.3.1 GRAM, Batch, and SRM

The GRAM plays a central role in that it is the site's generic interface to attract deployment of services by VOs. If a VO wants to acquire hardware resources to install services it may do so via the GRAM.

In principle, one could imagine a future in which the symmetry between job & data management is carried all the way through to the site infrastructure. A VO could lease disk & CPU to deploy its own storage elements by submitting installation scripts via the site's GRAM interface. This would imply that the site's batch system advertises the capability of its "batch slots" in sufficient detail to allow aggregation into efficient and performant storage systems.

In practice, efficient and performant storage systems that scale well are very difficult to build, and require a great deal of attention to hardware, OS, filesystem, and networking details. We thus assume that, at last for the near term, the symmetry between job & data is badly broken at the site level.

Sites will expose their storage to the VO via an SRM interface. Access to this interface is possible from outside the site. The services provided by the site's SRM interface form the moral equivalent of GRAM & batch system, and thus allow for the job-data symmetry seen by the job optimizer as discussed above.

6.3.2 The role of queuing in RE & CE

There is a separation of the Resource Element and the Compute Element, both of which may implement their own queues. Whether or not this is done in practice depends on the VO's preference between push & pull architecture for the JCAT/CECAT, as well as its desired control of scheduling policy.

In a pull architecture, the VO may submit relatively thin, possibly single-slot CE's via GRAM to the site's batch. The CE would then advertise its availability to the CECAT, possibly via Rproxy to overcome firewall rules. Job optimizer would then match entries in CECAT and JCAT, and submit the corresponding job to the CE directly, i.e without involvement of the GRAM.

In a push architecture, the VO may either deploy its own aggregation of a site’s batch slots into a CE that was deployed via GRAM, or use the GRAM/batch combination as a CE. The latter, obviously simpler arrangement, seriously limits the range of policy that the VO can implement. E.g. A site’s batch system is not required to guarantee fair share of resources among members of a VO as this would require a concept of hierarchical fair share that few if any batch systems support today.

6.3.3 Batch, SRM & InfoSvc

Batch and SRM advertise their state via a well defined, version controlled, schema. The definition of this schema falls under the purview of the OSG. In order for a site to join the OSG it needs to advertize its resources via an OSG compliant schema.

Questions:
Is there an OSG level InfoSvc?yes
Are sites required to advertise to this OSG level InfoSvc? no

Can VOs operate their own InfoSvce independent of OSG, and if so, yes
How would they obtain ads from sites? E.g is there a Subscription Model? Possibly there has to be. How does a VO which is going to “glide in all its services” to a generic site once it knows that that site is “usable” do this without an “information supermarket” or subscription?

6.3.4 Open Issues

(1) Processes at a grid site must access local and global resources with the privileges of the requestor. Such processes may include file transfer agents and batch worker nodes, among others. Access to grid resources uses the identity provided by the requestor's certificate to determine access privileges. Access to local operating system resources uses the uid of the running process, which may be leased only for the duration of the process to the requestor. Another requestor could subsequently get the same uid, and another process by the same requestor could get a different uid. However the resources may have a lifetime longer than the process that accesses them. A method is required to determine access privileges to local OS as well as local site resources for grid users.

(2) We have a general principle that a site may disallow access at the granularity of either VO or user. It is unclear what this means in practice. In particular, is there a guaranteed maximal latency for site access denial? And if yes, what does this imply for propagating signals indicating termination of access? For discussion at the October blueprint meeting

(3) Detailed use cases for storage have not been discussed yet. E.g., how are stage-in/out dealt with? Is there a concept of a "transient file", E.g. A file produced on a worker node that requires additional processing (e.g. Concatenation) prior to storage in a strategic SE.

(4) We still need to carefully discuss the file access methods. In particular, we need to clarify what we mean by "POSIX-like" file access, and how that relates to the Q's (1) and (3). POSIX file interfaces may be simulated through operating system specific libraries or file system drivers.

6.4 Interfacing the Facilities

Facilities and sites are responsible for administering and supporting the services, resources and infrastructure within their administrative domains. These include storage, processing, network, and database services as well as the security, operations, and policy infrastructures.

Sites have services used by local or remote users not on the common grid infrastructures. Sites and facilities will support local grid infrastructures which will federate with or partially be made accessible to the Open Science Grid, local resources that will be shared with VOs accessing them through OSG and local VOs that will want to use both the local resources as well as share those available through the OSG infrastructure.

These will be taken into account when defining each service (interfaces, capabilities, architecture) as well as in engineering the infrastructure.

6.5 Areas of Responsibility

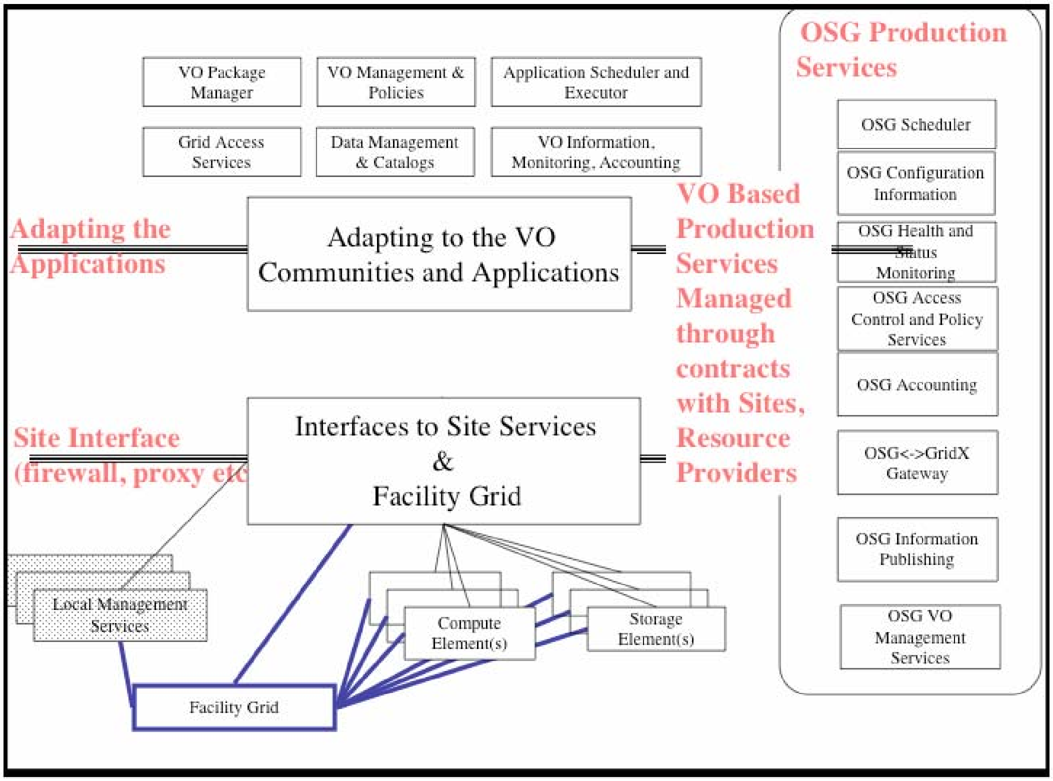
OSG Production Services
VO Package Manager
VO Management & Policies
Application Scheduler and Executor
Grid Access Services
Data Management & Catalogs
VO Information, Monitoring, Accounting
OSG Scheduler
OSG Configuration Information
VO Based Production Services Managed through contracts with Sites, Resource Providers
Adapting the Applications
Adapting to the VO Communities and Applications
OSG Health and Status Monitoring
OSG Access Control and Policy Services
OSG Accounting
Site Interface (firewall, proxy etc
Interfaces to Site Services & Facility Grid
OSG<->GridX Gateway
OSG Information Publishing
Local Management Services
Compute Element(s)
Storage Element(s)
OSG VO Management Services
Facility Grid

Figure 8: OSG Responsibilities

6.6 Storage Services

For the foreseeable future, support for three types of disk volumes is likely to be required:

(1) Storage Elements:
Storage Elements distinguish themselves by supporting an SRM interface as well as a posix-like interface. In general, not all sites will support the same Storage Element implementation. Sites will thus differ in the details of their posix-like interfaces, and it is the VO's responsibility to make this transparent to their members. To be able to do so, the site needs to advertise the type of posix-like interface in some fashion to the VO. Access to both SRM and the posix-like file IO interface is controlled via certificate based authentication.

(2) Shared Disk Volumes:
Sites will want and need to support at least some shared (e.g. NFS exported) disk volumes for VOs to install software distributions as discussed in Section 5.2. These spaces may be read-only from the compute nodes, and are generally less robust and less scalable than Storage Elements. Write access to these disks may require special privileges, and may thus not be open to all users of the site. Read access may be global to all users of the site, across VO membership.

(3) Stateless disk on compute nodes:
All compute nodes need to support some amount of locally accessible disk space that is guaranteed to the batch slot environment in which the user application is executing. This disk volume is stateless in the sense that its content is erased when the lease of the batch slot expires. Some combination of VO and site needs to guarantee that users can not trample on themselves, or on others. Read/write privileges of this space thus may be restricted to the application that has leased the space in conjunction with the batch slot.

7 Development & Deployment Grids

Open Science Grid has aspects of development, deployment and support for stable production operations. These activities place very different demands on grid infrastructure. In order to meet the needs of both development and stable production it is necessary to partition the grid computing resources at least logically.

There should be a Grid Laboratory Infrastructure on which new services are deployed, integrated and tested. There should be validation and transition procedures and contracts that cover transition of a stable “release” version of the Test Grid infrastructure to the production OSG infrastructure.

OSG is proposing to build on the computing infrastructure model already deployed by several of the stakeholders. In these models there are typically three classes of grids that range from very small, very volatile, and very responsive to very large and very stable grid environments. The three classes have been given different names on the single VO infrastructures, but they can be generalized to test, integration, and production. Configuration Management services should support dynamic and auditable reconfiguration of a Site and Resources from one of these grids to another.

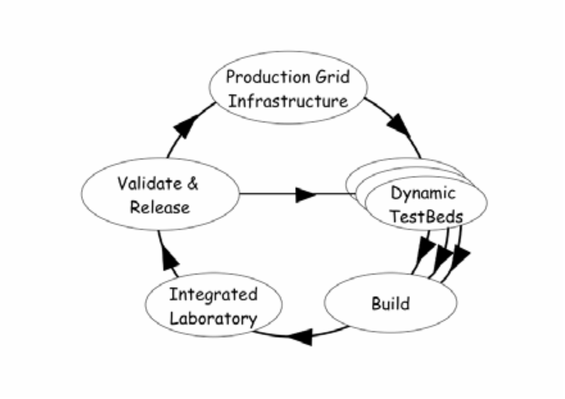

Figure 9: Grid Lifecycle

7.1 Existing Common Infrastructure

1. Test Grids:
Test Grid environments are used to verify packing, installation and configuration of new components for the OSG. The scale of the test grids can be very small, with only a handful of systems at any given site required to perform the needed validation steps. The ability to test a wide variety of platforms is critical to the success of the test program, so a diverse set of test sites is required. Packaging, configuration, and configuration management are some of the most technically difficult aspects of bringing up a large and efficient grid infrastructure. In order to test and develop these, speed of the response on the test grid is important.

2. Integration Grid:
Integration grid environments are used to test the functionality, stability, and scalability of grid services before they are declared to be production quality. The scale at which the integration grid operates needs to be sufficiently close to the scale of the production grid to diagnose potential scaling problems before they are exposed to production operations. Before a service is declared robust enough for the production grid it should be tested on an integration grid facility within at least an order of magnitude, and preferably larger, of the complexity of the target production facility.

The integration grid has two primary functions: to verify individual services in development and to integrate the developing services with the existing infrastructure and VO applications as a precursor to promotion to production grid resources. OSG is likely to have several independent development activities progressing at any given time and it will often be helpful to have dedicated resources. OSG should aim to have sufficiently flexible configuration management tools that sites can be moved from the production grid to the integration grid and back as needs dictate. The integration grid should be partitionable into test beds to verify specific services and combined into larger grids to validate services and environments together. The integration grid will not have a fixed membership and sites may join for defined periods of time before rejoining the production grid.

3. Production Grid Infrastructure:
The production grid infrastructure is used for stable production running. There are expectations for scale, responsiveness, and stability for the resources that make up the OSG production grid.

Services and applications running on the production grid infrastructure are expected to have been carefully validated on integration grid resources. All sites that participate in the OSG production grid may not have the same services, but the information services on the OSG should fully and accurately describe the grid services available. Upgrades and changes in service versions on the production grid need to be organized to ensure a stable and predictable environment.

7.2 Areas for Development

There are a variety of areas in the management of the grid layers that need to be understood for the Development and Deployment grid programs to be successful. A number of the projects listed below contain technical development and others require documentation, communication, and understanding.

1. Formal “roll out” procedures need to be established to specify how services can be validated in preparation for moving from the integration grid to the production grid.

2. A management structure is needed for the integration grid and the production grid. There is an expectation that the component testing on the integration grid is largely self-organized by the VO performing the validation, but it is unclear how the transitions between membership in an integration grid and a production grid are managed. It is not obvious how an organization asks for a period of time on an integration grid or how one schedules the larger scale integration for validating a grid environment for promotion to the production grid.

3. There is a clear need for advanced configuration management tools. As sites move between integration and production grids the classification and service versions need to be accurately published. The configuration management tools need to consistently configure the site, and ensure that the configuration is reliably published.

4. How are changes in the infrastructure communicated, agreed to, controlled? There is an expectation that OSG will eventually support the multiple versions of services and even multiple grid architectures. The flexibility needs to be tempered by the desire for stable and robust production operations.

8 Security Infrastructure

We aim to understand the end to end infrastructure including security policies and contracts.

8.1 Core

1. Public Key Infrastructure
GSI is the authentication protocol for the near future so Users will have to have the ability to acquire Proxy credentials. While OSG will not have to operate this infrastructure, authentication methods available to the VO's users and acceptable for the resource requests they need to make must be developed.

2. A robust authentication challenge process.
Third party authentication means that a resource provider is relying on the source of a Proxy for assurance that the agent asserting an identity is authorized to do so. When occasion gives cause for that authentication to be challenged, there must be a method for relying parties to have problems investigated and resolved.

3. VO membership service
An OSG service will negotiate, as part of its contract with a VO, how to determine if a requesting identity is a member of the VO. For the default VO, some general OSG membership service will need to be defined. In the near term, this might be a union of all the specific VO membership services.

4. Acceptable Use Policy
A baseline acceptable use policy must be in place for the default VO. Other VOs may have more restrictive AUPs that are invoked if users assert their VO membership authorization. No enforcement requirements are put on the resource providers unless specified in their contracts with the VO.

8.2 Higher Level

1. Defined system of expressing authorization attributes
The attributes presented by consumers to a policy decision point must have a consistent schema with the policy to be checked. Methods are needed for enforcing a standard attribute schema or matching policies with attribute authorities.

2. A Policy Enforcement Point control (or set of controls) needs to be implemented for services that are expected to enforce authorization decisions.

3. Audit
Services which accept delegated credentials must be auditable to resolve claims of challenged authentication and exposed risk. Action logs on grid services must be sufficient to determine which identity was associated with all processes and which AA tokens might be exposed in an incident.

4. Incident Response
A network of communication points and agreed set of expectations for resolving incidents would be useful. This is probably a human moderated "service".

5. User and Resource Administration Contact Information
Some procedure for putting resource providers (or incident response staff) in touch with users or other resource provider administrators is needed.

6. Authentication Service
Dealing with certificate revocation is a high maintenance load. An online service which acted like the credit card clearinghouses may be a desirable service. Some of the CAs are planning OCSP responders which may address some of this issue.

7. Restricted Proxies
A good amount of the current security concerns and demands are in response to the current "all or nothing" delegation schemes. A robust, usable method of restricted delegation would relieve some of this pressure.

8. Intrusion Detection
Some capability to efficiently monitor for unacceptable activity will be needed. Not obvious where the equivalent of the network border router would be. Work needs to be done to better understand the needs and capabilities here.

9. Testing
There needs to be some program of testing to identify vulnerabilities and work to get them fixed. This is more important than in the past since users have come to trust all available services rather than making specific decisions who to deal with. This testing program will likely violate the AUP for general users, so some method of deputizing trusted agents will be needed. Some process for dealing with unresolved vulnerabilities will be needed.

10. Recovery Procedures
Processes for disabling parts of the grid and restoring them to known good state will need to be developed and streamlined. The current procedures are not widely understood and are much too slow to weather an incident involving more than a handful of actors.

11. Policy Management
The problem of specifying policy, locating the policy decision functions, and maintaining the (distributed ?) policy at the policy decision points has to be addressed. This area is in the very early stages of development. Will need to start simple and not get beyond our ability to debug. This will likely introduce single points of failure and have a strong operational influence.

Things to Consider:

12. Sandboxes.
Better technology for limiting the risk by restricting the network space available to processes and/or the executables that may be run would relax the level of concerns. Current technology is both ineffective and difficult to work with.

13. Untrusted Terminals
It is looking like it may be practically impossible to adequately secure the users workstations, laptops, etc. to prevent a significant level of compromises of these machines. Current interest in technologies like OneTimePasswords (OTP) is a consequence of this. Similarly, we may have to come up with ways of dealing with revocation and/or restriction of proxies (or AA tokens in general).

14. Privacy concerns of stored data.

15. Support for Identify modification and multiple identities and credentials.

9 Policy Infrastructure

(This section will discuss the policy and contract infrastructure not related to security. It is likely that the implementations will be the same.)

1. Support for dynamic site policies.

2. Policy Reconciliation vs. Policy Enforcement.
Policy plays a central role in the OSG since it enables participation by a diverse set of users and resource owners, across organizational boundaries.

Both users and owners need to be able to express their policies. A framework needs to be in place that allows both enforcement and reconciliation of dynamic policies. Enforcement is generally restrictive. It’s the basis for trust for both users and owners that their policies are adhered to.

Policy reconciliation is enabling; it allows both users and owners to reach their *economic* objectives.

The challenges in policy enforcement are largely in reliable delegation and guarantees on maximum latencies for revocation. E.g., a site may change its policy and expect the new policy to be enforced within some time limit. How are contracts revoked that no longer satisfy the changed policy?

For policy reconciliation, the challenges lie in allowing policies that are sufficiently expressive. A very powerful policy reconciliation paradigm that OSG expects to employ is the concept of *matchmaking* between offers of and requests for resources. However, this is unlikely to be sufficient. In addition to simple matching, OSG infrastructure will most likely require a concept of preference, or ‘rank’, a notion of quota on aggregated resources, and the possibility to encode hierarchically structured policies.

An example of a hierarchically structured policy that needs to be supported is hierarchical fair share. A site invariably has policies that express preferences for some VOs over others. The VO in turn needs to express policies that reflect (some of) its organizational structure. It is thus a requirement for the end-to-end policy infrastructure to allow for policy expressions that may be depicted in form of a decision tree, or a directed acyclic graph.

10 Operational Infrastructure

The Operational Model for Open Science Grid will involve a distributed structure of support between the VO administrators, the Grid Operations Center, the service and technology providers and the Sites. The Users are not expected to contact the Sites directly. All problems are expected to be tracked and pertinent information gathered and published in order to build up a base of knowledge, lessons to be learned, and input for future planning.

1. There will be a defined User Support model.

2. Operational support will operate through the VOs who will triage problems reported. There will be identified responsibilities for VOs, resource, site and service providers.

3. The infrastructure will offer infrastructure packaging, distribution, configuration management services.

4. The operational infrastructure will need to publish accounting and monitoring information with the help of Service Providers and Resource Owners.

5. The operational infrastructure will provide proactive communication channels for incident response, notification of certificate expiry, fault handling etc.

6. Customer support centers will provide support services that include ticket systems and management.

11 Technology Roadmap

This section will be expanded to include a complete list of services and capabilities and an indication of how far from the blueprint current capabilities are thought to be.

The goal of the OSG in implementing the technology roadmap is to provide reference end to end implementation to help participants with minimal implementations, and promote a broad base of innovation and participation.

When mapping the roadmap to implementations, consideration of benefit vs. cost will be done and a reasonable cost/benefit point will be one of the inputs to decision making.

12 Appendices for Discussion at October Meeting

12.1 Relationships between VOs

```
VO[x] = {U[i];   R[j];   VO[y]}
         0..m    0..n    0..p
```

where at least one of `m|n` is non-zero and `R`'s imply presence of a `U` in the role of resource-owner

- A user `U` can either be a consumer or a resource-owner or a service-provider or any combination;
- A service is a method to access a resource or agent (services being implicit & abstract, are not considered in this model/diagram);
- A resource `R` has value to some consumer, is owned, and can be leased to a consumer or a `VO`;
- A `VO[y]` follows the definition of `VO[x]` recursively.

Different ways a `VO` can be formed:

```
VO[x] = { U[1..m] }
VO[x] = { R[1..n] }
VO[x] = { U[1..m]; R[1..n] }
VO[x] = { U[1..m]; VO[1..p] }
VO[x] = { R[1..n]; VO[1..p] }
VO[x] = { U[1..m]; R[1..n]; VO[1..p] }
```

Using the notation `Entity[1..k]` to indicate `Entity[1], Entity[2], ... Entity[k]`

12.2 VO Software Installation at a site

A VO wants to install some application software for latter use at the site by its members. We assume that at least in the near term, sites may provide disk volumes that are NSF exported as discussed in Section 6.5, and implemented in Grid3 via /scratch/data and /scratch/apps.

In this section we describe how the mechanics for this is expected to work out for the foreseeable future.

User1 is the present VO software coordinator. She has prepared a new release of the VO software on some reference platform and wants to push this software out to some sites for use by the VOs members.

As software coordinator, she is privileged to attach the special role of "<vo>soft" to her cert. She uses this to submit an installation job to a site via the sites GRAM interface. Based on her role she is mapped to a UID on the site that gives her write access to a quotaed disk volume that is NFS exported read only to all compute nodes on the cluster. This NFS volume may be world readable within the site.

We need to work out carefully the responsibilities of site, VO, and OSG with regard to what roles have what privileges, and how to communicate the total quota per VO, and how this is sliced up into different roles.

12.3 VO Service Installation at a site

VOs need to be able to deploy their own services at a site. This requires the site to advertise resources that are capable of hosting services. Such resources may include externally visible ports on dual-homed systems that straddle public-private network boundaries.

These services will be deployed in analogy to the VO based software installation, except that statefulness is not restricted to disk space alone. Services persist for extended periods of time, probably months after installation in order to be useful for the VO. The VO may need to update these services with new versions of its services at its discretion. The VO will want to register the dynamically assigned hostname(s) and port number(s) with the appropriate discovery service.

All of this has to be done keeping site security concerns in mind. It requires trust between the site security personnel and the VO. It is likely that not all sites will support all types of dynamically deployed VO services that exist within the OSG.

We need to work out carefully the details for some realistic examples of dynamically deployed services and gain some experience in practice.

12.4 Issues in Data Management

Section 6.3 discussed similarities between job and data management. In the present section we focus on issues that are unique to data management. This is by no means complete, and will be elaborated upon in the discussion of use cases in Section 5.

The underlying assumption in all of this is that data management is a VO responsibility. The VO obtains the physical spaces for storing the date via contracts with storage services. These contracts are dynamic in the sense that data management includes moving data around in order to safe guard against data loss in light of expiring contracts.

We assume that we are dealing with files as fundamental entities in which data is stored. However the types of files that need to be managed range from primary data via multiple stages of processed and derived data all the way to scientific publications. It may thus span a great variety of different formats, and includes being able to implement a broad range of policies.

Data management is tightly connected to virtually all aspects of grid activities. It should thus be expected that implementations, and even detailed decomposition of services may differ between VOs as well as over time. We thus attempt here to discuss conceptual issues that have to be dealt with rather than a detailed service decomposition.

12.4.1 Namespaces

Data management is a complex activity requiring an interplay of a variety of namespaces, and maps between them. The most obvious is the file logical namespace. It uniquely defines all files managed by a single instance of the data management system.

Files need to be stored somewhere, and data management needs to name the resources where it stores those files. It may prove useful to distinguish logical from physical resource names, thus including a resource logical namespace.

Files are owned by users, and data management has to maintain a variety of policies based on file ownership. It may thus prove useful to include a user namespace.

Files have attributes and file management has state that needs to be managed. Both attributes and state have meaning, and need to be referred to while they are tracked. There is thus a metadata namespace which is most likely heavily scoped to provide order among different types of metadata.

Grouping files into datasets, or metadata into groups, may be conceptualized as constraints on the file metadata. It may prove useful to introduce a logical constraint namespace in order to manage this.

12.4.2 Modifying files

People make mistakes they want to correct. This leads to a natural desire to modify files. However, file modification in distributed systems is very difficult to implement without significant performance loss because it leads to "cache coherence" requirements.

One may be tempted to disallow modification of grid files. To modify a file may thus require a sequence of file export, file modification, file import. File modification would thus be reduced to file versioning. The inherent space inefficiency of such a design decision may be tolerable if modifications are rare.

12.4.3 Import, Export, and Synchronization

The OSG principle: "services need to function and operate in a local environment when disconnected from the OSG environment" poses significant challenges to data management. It requires that a user may "export" a partial replica of the full data management environment, use that partial replica (i.e. modify its content), and import it back in, synchronizing it with the master copy.

This synchronization will in general be rather complex as somebody else may also have "checked out" a partial overlap. The synchronization may be manageable if modification of files and metadata is restricted to "adding new versions" rather than true modifications implying irretrivably discarding previous information.

However, if a mechanism for export, import, and synchronization can be developed then that same mechanism may be used for a variety of use cases. E.g., site autonomy requires distributed databases, partial or complete merging of data management system instances, etc. may be accomplished using the same mechanisms. This would for example enable dynamically created sub-VOs to have persistency of data beyond the VO lifetime by merging data management instances back into parent VOs. It also increases scalability.

12.4.4 Guaranteeing file integrity

As storage media densities increase bit error rates are expected to at best stay constant. Increasing data volumes per transfer thus leads to increasing error rates per transfer. It is thus mandatory that data management includes data integrity checks in order to guarantee file integrity. This is a non-trivial end-to-end requirement.

12.5 Virtual Site

Physical sites or facilities provide an administrative context for users and organizations. Some aspects of this administrative context like user registration, well defined acceptable use policies, and security incidence response are also needed for administration of a grid. In principal, each VO could settle these issues with each site. However, in practice, it is likely to be more efficient if sites and facilities negotiate a common administrative context, thus forming a "virtual site" or a "virtual facility".

OSG spans across sites which differ in their administrative context. E.g., sites differ in their expectations regarding security, privacy, accounting, or acceptable use. There may thus be a well defined namespace of virtual sites and a given VO may be able to operate on only a subset of these.
It is conceivable that the namespace of virtual sites evolves to a namespace of administrative context constraints. However, initially it is quite likely that the OSG will be a single virtual facility.

12.6 Thoughts on Requirements & Use Cases for the OSG Discovery Service

One of the goals for OSG is to have a "discovery service". OSG is expected to support many thousands of registered users. It should be expected that a significant fraction of the suppoer load may be attributable to users not reading manuals, or not re-reading manuals when things change. A typical request might be "I used ... 6 months ago, but when I do this now I get the following error message:".

A discovery service can help minimize this support load in two ways. It can make changes in computing infrastructure transparent to the user, and it can provide the user a means to explore the available services, similar to an interactive manual. The former is essential in a highly dynamic grid environment. The latter is desirable.

12.6.1 Making Changes transparent to the User

As part of the OSG principles, we require that VOs and sites may update versions of their services without this affecting all of OSG. A VO may use this by having, e.g., a 3-tiered rollout of services classified as "development" "integration", and "production". The version number of a service that is labeled as "production" will thus change over time. The VO will want to have the majority of its users use whatever version is labeled "production" rather than exposing fixed version number to their user community. A VO would thus be able to retire an old version of a service simply by changing labels.

The discovery service could provide the label. E.g., a client access to a service may always be directed via the discovery service in order to discover the location, port, maybe even some details about protocol.

The discovery service may provide sufficient flexibility for the infrastructure to merge services, or service providers. E.g. imagine a temporary sub-VO creating data that persists beyond the lifetime of the sub-VO and its services. When the sub-VO disappears, a larger VO may take over the data, and serve it from its archival storage. A user that tries to access the now defunct sub-VO's data may be automatically routed to the archival storage services of the parent VO by the discovery service.

By structuring discovery services in a hierarchical fashion, we may have a very thin layer at the top that never changes. E.g., imagine a top level web services interface that is tied into "`http://www.opensciencegrid.org/discoveryService`", a URL that could well remain unchanged for decades to come. If the services at this layer remain basic, referring to others in a hierarchical fashion, then this might be implementable thin enough to withstand the test of time.

12.6.2 Interactive Manual Functionality

In addition to the relay functionality described above, users might expect to be able to discover interactively the syntax and API of some services, as well as "browse" for others.

A user may start by asking the discovery service about its methods for discovering services, and get back something like:

`ShowServiceCategories()` returns list of service categories.

`ShowVOs()` returns list of VOs.

`ShowServices(VO=CMS,Category=discoveryService)` returns list matching Requirements.

`ShowMethods(CMS-DiscoveryService-Devel)` returns Methods of the service named.

`GetAgent(CMS-DiscoveryService-Devel)` returns agent that can execute Methods.

This is meant as an illustration only, not as a proposal for a sensible implementation. The user might then drill down further, exploring interfaces as she goes along. A VO might make a design decision that all of its services are required to have a README method that details how to do a simple test of the service. A user might thus use the discovery service to discover the range of services their VO supports, and then query the services themselves for their usage.

13 References

The Open Science Grid Consortium, `http://www.opensciencegrid.org/`.

Open Science Grid white paper, v2.3, August 10, 2003.

Open Science Grid Presentation to Fermilab Computing Division, May 14, 2003, `http://cdinternal.fnal.gov/Org2003/BriefingMtg/2003Briefings/2003-05-CDbriefing.pdf`.

R. Pordes, L. Bauerdick, V. White, *The Open Science Grid*, abstract submitted to CHEP 2004.

OSG Security Technical Group, `http://www.opensciencegrid.org/techgroups/security/`.

OSG Storage Technical Group, `http://www.opensciencegrid.org/techgroups/storage/`.

The Grid 2003 Project, `http://www.ivdgl.org/grid2003/`.

The Grid 2003 planning document, v21, GriPhyN 2003-33/Grid3 2003-3, Nov 5, 2003.

International Virtual Data Grid Laboratory (iVDGL), `http://www.ivdgl.org/`.

The Virtual Data Toolkit (VDT), `http://www.cs.wisc.edu/vdt/`.

LHC Computing Grid Project (LCG), `http://lcg.web.cern.ch/LCG/`.

The Particle Physics Data Grid (PPDG), `http://www.ppdg.net/`.

The Grid Physics Network Project, `http://www.griphyn.org/`.

I. Foster, and C. Kesselman (eds.), The Grid 2: Blueprint for a new Computing Infrastructure, Morgan Kaufmann, 2004.

H. Wang, S. Jha, M. Livny, P. McDaniel, *Security Policy Reconciliation in Distributed Computing Environments*, IEEE Fifth International Workshop on Policies for Distributed Systems and Networks (POLICY 2004), New York , June 2004.

I. Foster, N. Jennings, C. Kesselman, *Brain Meets Brawn: Why Grid and Agents Need Each Other*, The 3rd International Conference on Autonomous Agents & Multi-Agent Systems (AAMAS 2004), New York City, July 2004.

Grid Resource Allocation and Management (GRAM), `http://www.globus.org/gram/`.

A. Shoshani, A. Sim, and J. Gu, *Storage Resource Managers: Essential Components for the Grid*, Grid Resource Management: State of the Art and Future Trends, Kluwer Publishing, 2003.

B. Clifford Neuman, *Scale in Distributed Systems*, Readings in Distributed Computing Systems, IEEE Computer Society Press, 1994.

Tim Berners-Lee, J. Hendler, O. Lassila, *The Semantic Web*, Scientific American, May 2001.

R. Raman, M. Livny, and M. Solomon, *Matchmaking: Distributed Resource Management for High Throughput Computing*, Proceedings of the Seventh IEEE International Symposium on High Performance Distributed Computing, Chicago, July 28-31, 1998.