Enterprise Grid Data and Storage Provisioning Problem Statement and Approach

Developed by:

Enterprise Grid Alliance
Data Provisioning Working Group

*** Version 1.0 ***

As approved by EGA

28-February-2006

This Page Intentionally Blank

Copyright Notice

Intellectual Property Notice

Revision History

Version	*State*	*Date*	*Comment*
1.0		28 February 2006	Initial v1.0 document for Public Release

Contents

Figures

Preface

The purpose of this document is to provide a foundation of the overall data provisioning and storage problem within enterprise grids and to provide a statement of future direction. The document builds on the EGA Reference Model by identifying the unique data and storage provisioning needs of commercial enterprise grid computing. As such, the reader is highly encouraged to first read the Enterprise Grid Alliance Reference Model v1.0 document before reading this document.

The document decomposes the high-level provisioning problem into a number of sub-problems many of which are aspects of traditional Storage Management and Storage Resource Management. The document examines whether suitable interfaces into the actual grid resources exist, and if so, where. The area of Asset Management is examined in more detail and provides an example of an approach for moving forward.

The Enterprise Grid Data and Storage Provisioning Problem Statement and Approach document consists of the following sections:

The **Introduction** provides an overview of the Grid Management Entity as defined in the EGA Reference Model and details the grid data and storage provisioning challenges within the enterprise data center.

The **Problem Statement** further clarifies the dynamic nature of grid provisioning and how it is at odds with traditional storage management practices.

The **Management Issues and Example Problems** section gives more details about the general problems of grid data and storage management including asset discovery and management, provisioning and capacity management, data movement and backup, policy management, alert management, billing, and content provisioning and discovery.

The **Asset Management** section provides a deeper look into the discovery, commissioning and decommissioning, and capabilities as related to Service Level Objectives of asset management within enterprise grids.

Comments regarding content and format are welcome, and may be directed to the EGA Grid Data Provisioning working group (ega_datawg@mail.gridalliance.org).

1 Background

1.1 The Enterprise Grid Alliance

The purpose of the *Enterprise Grid Alliance (EGA)* consortium is to drive the adoption of grid computing and the technologies that enable its deployment and use within enterprise data centers. The EGA is pragmatic and is focused on short and medium term goals that accelerate adoption, as well as the long-term objectives in enterprise grid computing.

Grid computing is typified by a focus on sharing and managing pools of network-distributed resources to deliver applications and services. Grid computing environments may also be typified by -

- The use of network distributed, shareable pools of discrete resources to achieve greater strategic agility, architectural flexibility, performance, scaling, resilience and utilization
- A focus on managing services, rather than on managing individual, discrete components particularly as enterprise grids turn networks into arbitrarily rich and complex fabrics of resources
- Flexibility or mutability, as service components may be regularly composed, re-purposed or re-provisioned in response to changing goals, regulations, and business objectives or simply in response to a service's needs
- Application or service architectures that are disaggregated or distributed in nature and which leverage the properties of the fabric of resources. Examples include traditional multi-tiered applications such as ERP or CRM, Service Oriented Architectures (SOAs) or decomposable compute intensive workloads
- The consolidation of computing components into [typically] a smaller number of larger resource pools that promote easier provisioning, increased service availability, greater resource utilization and simplified management
- The standardization of components and their interfaces, configurations, processes and applications which all serve to promote highly automated and resilient architectures that can efficiently respond to changing business and service requirements

Enterprise Grid Computing is specifically the use of grid computing within the context of a business or enterprise, rather than perhaps for academic or research purposes. While there may be some overlap of requirements between enterprise grid architectures and other variants, it is clear that there are unique requirements and challenges associated with the adoption of grid architectures within an enterprise, especially in the operational sense.

Enterprise Grid Architectures are typically managed by a single enterprise, entity or business. A single organization is responsible for creating and managing a shareable networked pool of resources, composing higher-order components and services from individual resources, and delivering services that not only are capable of meeting a set of defined goals and requirements but also help drive value for the business. Resources can take the form of compute, network, storage and even service capabilities. The resources and services may or may not all be owned by a business. Subject to policy and business objectives, an organization may choose to leverage resources and services from another entity such as a service provider or an outsourcing or managed services firm. What defines the boundaries of the enterprise grid is its sphere of management responsibility and control. An enterprise grid may be confined to a single data center or it may extend across several. There are typically no geographic limitations to the size and scope of an enterprise grid architecture. The resources and services managed within enterprise grid architectures are however typically under the management responsibility and control of a single organization regardless of their actual physical location.

The EGA and the scope of each of the EGA working groups is described in greater detail in the publications: “Accelerating the Adoption of Grid Solutions in the Enterprise” and “Enterprise Grid Alliance Reference Model v1.0”, available on the EGA web site – http://www.gridalliance.org.

1.2 The Goals of the Data Provisioning Working Group

The initial focus of all of the EGA’s working groups is on *commercial, enterprise applications* within a single data center. Commercial, enterprise applications are the lifeblood of most organizations. They are involved with the delivery of content and services to customers, partners, employees and shareholders as well as organizing and managing supply chain and business operations. Often, these services tend to be multi-tiered however; this architectural artifact is not a requirement. Commercial, enterprise applications also may have both batch and interactive components, may be geographically distributed, may be based on commercial or open source software packages, and are often highly customized for a given organization’s needs. Examples of such applications include but are not limited to CRM, ERP, BI, etc. Such applications may or may not be enterprise grid enabled, by default. Applications that are not enterprise grid enabled may still be able to participate in an enterprise grid architecture through the use of a connector, proxy, or other mechanism.

The EGA expects to extend the scope of its working groups into multi-data center models as well as *technical enterprise applications* at a future date. Technical enterprise applications tend to be more compute intensive and less interactive.

Enterprise computing commonly involves massive, terabyte-scale amounts of data. Grid computing efforts have not yet addressed many of the data management requirements these data volumes place upon enterprise grid deployments. The goal of the EGA Data Provisioning working group is to identify the requirements of data provisioning in enterprise grids and to develop usage scenarios and reference implementations. The initial focus of the working group is on bulk operations and simple paradigms, such as deployment/redeployment. Additional work includes developing usage scenarios and a mapping of existing specifications to these scenarios, evaluating current possibilities for standards-based management of heterogeneous storage provisioning, and identifying gaps in vendor functionality that hinder development of standards based heterogeneous storage provisioning. Based upon these findings, the working group will determine how any unmet requirements would be best satisfied, such as through submitting these requirements to another organization or collaborating closely with another organization. Subsequent focus will be on incremental and fine-grained data operations.

2 Introduction

The overall charter of the Data Provisioning Working Group of the EGA is to advance the state of the art of data provisioning in grid-oriented enterprise data centers. This paper is an overview of the problem and a statement of future direction.

According to the *Enterprise Grid Alliance Reference Model* [EGA 1], "*Enterprise Grids* are typically managed by a single enterprise - i.e. an entity, the business, is responsible for managing a networked pool of resources, and a set of services, and for managing the assignment of resources to services in order to meets its goals. The resources and services may or may not be owned by that business, for example in the case of managed services or a service provider/outsourcer. What defines the boundaries of the enterprise grid is management responsibility and control. An enterprise grid may be confined to a single data center or it may extend across several."
The EGA Reference Model defines a *Grid Management Entity* (GME), which is the intermediary between applications and grid resources, as well as the management interface into them used by the administrator. Figure *i* of the Reference Model document, reproduced here, illustrates this:

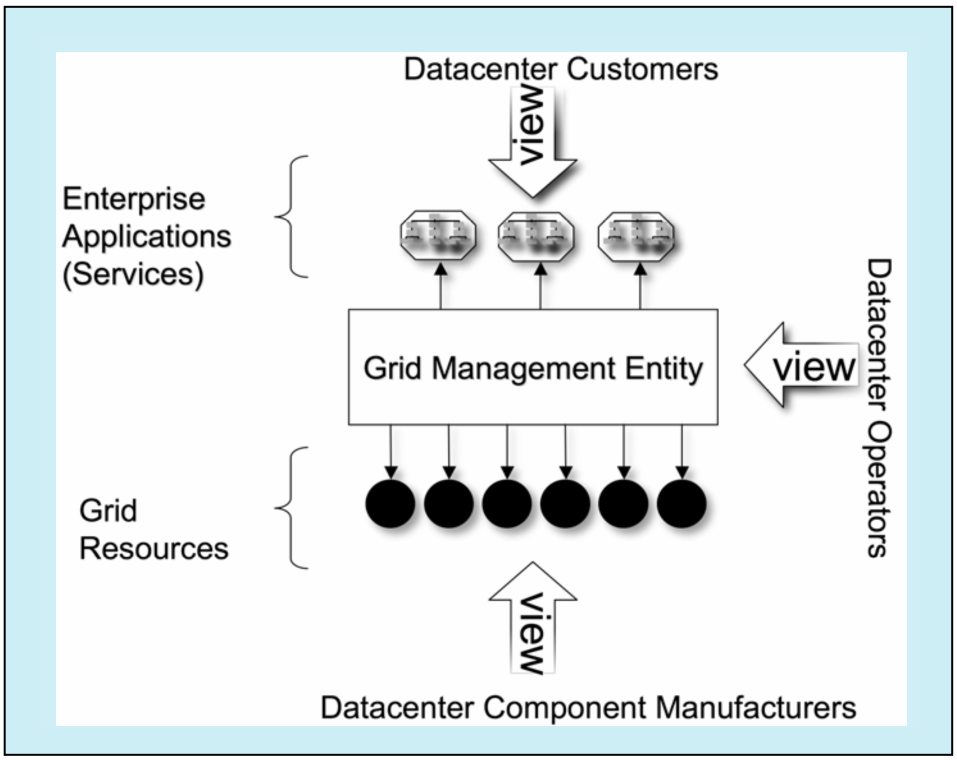

Figure i - Grid Management Entity

The GME is essentially hierarchical and distributed in nature. The GME itself may be decomposed in the same way as applications and services, i.e. into grid components. To do data provisioning, the GME 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. For example, a user or application requests a data container conforming to a predefined *Service Level Agreement* (SLA) of four TB containers and meeting specific *Recovery Point* and *Recovery Time Objectives* (RPO and RTO). To provision this container, the GME must:

- Find an appropriately sized storage container of the correct access type (Fibre Channel (FC), iSCSI, NFS, CIFS), possibly using best-fit logic
- Secure the container so that only the requester can write to it and read from it
- Set up the needed switching between the requester and the storage
- Set up a VLAN or Fibre Channel mask to establish security and prevent accidents
- Engage any needed over-the-wire or at-rest cryptographic security mechanisms
- Hand the client back an address to use for the data container

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 needed to be frequently snapshotted and/or replicated for Disaster Recovery (DR), RPO/RTO 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, then, is the grid data provisioning problem in a nutshell. This high-level view decomposes rapidly into a number of other problems—each a significant subject in its own right—as one begins to drill down. 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.

This paper decomposes the high-level provisioning problem into a number of sub problems. Many of these are aspects of traditional Storage Management (SM) and Storage Resource Management (SRM). If a traditional discipline such as storage capacity management is required to do a good job of data provisioning, we will include it and examine its requirements.

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. This paper will not attempt this, except superficially. However, in one area (Asset Discovery), we will take a deeper dive as an example of our intended approach moving forward.

3 Problem Statement

The dynamic nature of grid provisioning is at odds with traditional storage management practices. The data and storage resources within an organization or grid are difficult to manage – heterogeneous environments, vendor specific management tools, incompatible technologies, and inconsistent management procedures and policies all contribute to the problem. Administrators use many tools with many options that must be carefully planned and sequenced to accomplish configuration updates or equipment installation, decommissioning, or “repurposing”. In almost all scenarios, the possibility of unplanned downtime, performance problems, or data loss is significant. These factors combine to make storage administrators one of the most conservative and change-resistant groups in the industry.

Current best practices in the face of these problems entail setting up a homogeneous environment, selecting a core compatible set of vendors’ management tools, selecting compatible and interoperable technologies and options, and developing consistent procedures and policies. When this is achieved, one begins to think about how to realize better capacity utilization than the current 30% that is often quoted. This is where grid technology can make a contribution.

Today, the most commonly used tool to track complex environments is an Excel spreadsheet. This spreadsheet is probably never totally correct in an enterprise environment, which can be loosely characterized in two ways:

- an environment so large and complex that something is always broken
- an environment large enough that something is always being upgraded

Because manually keeping track of this myriad of constantly changing details is so difficult, automation is key to managing a grid. To determine what to automate, we take a top-down view of data management.

Data management includes the definition of *service level objectives* (SLOs)[1], the association of SLOs to managed targets, monitoring targets for SLO conformance, and generating alerts when a target violates one of its SLOs. During the SLO definition, monitoring, and problem analysis phases, many different managed targets from many vendors may need to be examined. Thus, there is a need for—and the GME provides—a single management ecosystem in which all these targets may be correlated. SLOs—aggregated into an SLA—can also be used to describe the results of a proposed new configuration or configuration delta. Managing storage for clustered hosts or for replication technologies adds yet more complexity.

Implementation of SLOs for data management requires the underlying system to do what is traditionally called *storage management* (SM). In general, all of the storage available to a client of a grid today—except its own local storage—is networked. Storage networks include elements such as hosts, disk arrays, NAS filers, NAS heads, tape libraries, software volume managers and virtualizers, host HBAs or NICs, and network switches. The storage management problem is to discover, configure, and monitor these elements in a manner conformant with enterprise policies and SLAs.

[1] We take the simplistic view in this paper that SLAs are agreements to conform to the performance parameters specified by one or more Service Level Objectives (SLOs).

4 Grid Data and Storage Management Issues and Example Problems

What makes storage management requirements interesting in an Enterprise Grid is that data and applications are much more fluid in a computing infrastructure. Consequently, whole new sets of requirements are born.

This section gives more details about the general problem of *grid data and storage management.* Management of storage-related entities in the EGA context covers the two basic areas of ***device management*** and ***data management***. Device management is broadly defined as functions associated with management of the physical storage hardware. This includes device configuration, asset management alert reporting, root case analysis and other areas related to the maintenance and operation of physical components. Data management is defined as the functions associated with the logical contents of the storage. These areas include data backup, data copy, and charge back. A GME must of course also manage the compute and switching infrastructure of the grid.

Ideally, data and storage management for a grid will follow a *utility computing* model. This means consumers view storage as a resource with predictable behavior that is simple to manage. Storage behavior is predictable when it follows its *service level agreement*, or SLA. As long as the SLA is satisfied, neither the physical location of the storage nor the hosting hardware matters. The cost of using storage is a function of the SLA, capacity used, duration of use, time window, identity of the storage consumer, and other variables.

Grid storage administrators routinely have to solve the problems that are itemized in the section below. This list of problems is not meant to be comprehensive, merely an illustration of the scope of the problem space. Furthermore, while the problems are categorized into general areas, their solutions tend to involve many functional areas at once.

4.1 Device Management

4.1.1 Asset Discovery, Asset Management and Topology

Asset Management deals with physical assets and their locations, owners, lifecycles, cost and so on. Using the Storage Industry Networking Association (SNIA) dictionary definition, "topology deals with the questions of what components are directly connected to the other components from the standpoint of being able to communicate." [SNIA] Topology then is the logical view, the way in which the physical resources are connected.

To manage resources, they must first be discovered. An asset database provides a starting point, but such databases often drift from the true picture of the Enterprise Grid. Ideally, assets are discovered completely automatically with any administrative input such as location identifiers and billing information captured via automatic means. Discovery may use SLP, SNMP, or vendor unique protocols.

Once discovered, assets must be managed and tracked. Useful information includes manufacturer/model, vendor, purchase date, purchase cost and depreciation function, physical location, owner, admin contacts, purpose, billing parameters, and usage policies. Other desirable information includes support cost and renewal information, retirement schedule and method, and security information.

Asset Management generates data that is used by client side services such as those that display grid topology. Typically, these services read a database created by an asset discovery mechanism and display the relationships found in a graphical form. Administrators can see views of the topology filtered in several different ways – by application, by line of business, by physical location, and so forth. Clicking on an icon representing a storage element allows drill-downs. Understanding the topology can be very helpful in provisioning and troubleshooting. For example, when a resource goes down, topology can help figure out which applications and services are affected.

Sample Asset Management problems include:

- When allocating storage, how can we be sure that it is reachable from all the hosts in the grid? Issues include SAN "name server zoning", VSANs, VLANs, array LUN masking, switch hard port zoning, usage policy, lack of a physical connection, or host configuration issues.
- We want to move a host from one cluster to another. Will it be able to reach all the storage it needs to as a member of the new cluster? If not, what changes to switches, arrays, host configuration, etc. will be required to make the storage reachable?
- Which hosts have access to which storage?
- What LUNs are configured on a disk array? Which hosts use the LUN?
- A host or other component is at the end of its lifecycle and is being replaced or upgraded.

4.1.2 Provisioning, Application Services and Capacity Management

Storage provisioning involves configuring physical storage systems such as disk arrays, NAS filers, switches, and host operating systems in order to provide access to storage according to policy and SLA requirements. This is a key value proposition for storage management. Provisioning is simple in concept and very complex in the concrete. Much of the complexity comes from the fact that provisioning requires control and management at many layers of the storage stack, from the application all the way into the storage array. Unfortunately, until today there have been few standardized interfaces for these layers. It is often the case that each vendor provides their own mechanisms for managing their component in the storage stack. SNIA, with SMI-S is evolving a standardized approach to these controls and provide standardized management at each layer of the storage stack, and this will enable the development of robust grid management services. We strongly support this effort.

The GME must also communicate with Application Services. For example, another form of asset discovery involves identifying the applications that use the managed resources. Host-side agents are normally required for this, and are a feature of middleware and management applications that understand resource usage to this level.

Provisioning must determine and track what storage topologies are required by the applications on whose behalf storage requests are being made. For example, databases frequently desire to store their logs and table spaces on different containers.

Capacity planning is a frequent pain point, experienced in the early stages of planning a grid and at many steps before and after storage is provisioned. This means monitoring capacity usage in a way that can be used dependably to govern budgets and timetables for future acquisitions. Doing this correctly can help greatly with proper utilization of resources, as a frequent cause for underutilization is over provisioning brought on by fear that future expansion needs will not be met.

Sample problems for these issues include:

- A database cluster will soon require additional storage. Where can this storage be allocated?
- How much unconfigured capacity (perhaps parameterized by quality of service) is available on a given storage device?
- Given an application's storage resources, are there any backup and DR implications that are not addressed by each component's SLO? I.e. do backups need to be coordinated, etc.?
- Which applications are using which storage resources? Given a storage element, what are all the applications that would be affected if it would totally fail?
- Given a LUN on a disk array and a given host, what is the host device pathname for the LUN?
- Assuming a grid exists a long time with ever changing applications, will fragmentation of data become an issue?
- Need to safely decommission a storage element.

4.1.3 Root Cause Analysis

Root Cause Analysis is generally a client of a monitoring service. When there is a cascading I/O error, Root Cause Analysis is responsible for finding the element that originally failed, and to the extent possible, why. Issues include running out of space on cascading file extend errors, or finding the bottleneck on a database transaction that repeatedly takes too long. An example of Root Cause Analysis is:

- Data accessibility just went away. Which component failed and why?

4.2 Data Management

4.2.1 Data Copy, Data Movement and Backup

The most basic use case in a grid scenario is that an application is provisioned, some storage for that application is also provisioned and then populated with a base data set for further processing. This implies that data copy is a fundamental need in a grid.

There are various levels of data copy support available in a grid environment. The most efficient is a copy-on-write clone. Currently and for the foreseeable future, this will be a vendor-specific operation. The least efficient is a host-side copy. In the middle, high-value storage environments are able to do storage-to-storage copies using various replication technologies.

Fundamental copy operations include homogeneous cloning, homogeneous and heterogeneous copying, deletion and/or shredding of data (a copy plus a shred effects a move), snapshot and replication/mirroring.

Backup is the process of making redundant copies of data. Backing up data addresses two kinds of data loss; data loss from equipment failure and data loss from user error. The latter is by far the most common reason for losing data. Historically, the backup process involved putting a copy of data on tape. Various forms of copying to tape were employed including copying all files to copying just the files that had changed since some point in the past. Lately, backup to disk is more viable due to the decrease of storage costs.

Example issues include:

- I need a temporary copy of an application data set.
- How can I effectively move data from one container to the other?
- I need a copy of data set X as it existed a specified time ago.
- In the event of a disaster, I need things to be up and running again in four hours.

4.2.2 Policy Management and Quotas

A variety of technologies may be used to set and enforce policies. Quotas are a common one. Quotas are used to ensure that users do not use more than their allocation of storage. Quota management allows a user to grow their storage consumption, as the application requires. Greater storage consumption results in larger storage costs billed back to the user. However, quota management allows the enterprise to establish hard limits as to the maximum space the user is permitted to use. Example quota and policy problems are:

- Which disk arrays are “usable” by the order processing department? This is an example of a restriction of usage due to enterprise policy.
- Which users have access to a given file system or LUN?
- What auditing policy is in effect for a given file system or LUN?
- What compliance policies are in effect for a given file system or LUN?
- What secrecy and security policies--e.g. encryption on the disk, over the wire, signing—are in effect for a given file system or LUN?
- What alerts and notifications are configured for a given container?
- How can I prevent abuse of file system space by users?

4.2.3 Monitoring, Auditing and Alert Management

These functions actually span both the device and data layers in a grid, as both layers must be monitored and may generate alerts.

Modern best practice recommends a systematic table driven approach to collecting data from all the managed targets. This data can be stored persistently, and summaries can be generated and stored persistently. This data is often stored in an SQL database, allowing storage management client applications to perform analysis via SQL queries. Metric collection is important as a foundation for billing, root cause analysis (RCA), capacity planning and other alert related activities.

The information collected from the managed targets can take the form of asynchronous alerts and time-driven sampling. Asynchronous alerts are usually of the nature of failure status indication for a failing component. Time-driven sampling data is usually of a performance or a capacity utilization nature. These two types of data serve different purposes, but having a common store and retrieval interface provides great value to clients of the data. Clients can subscribe to the alerting service to receive alerts, subject to the approval of the Grid Security service (as will be defined by the EGA Security Working Group.)

Application Management is an emerging technological area attempting to provide an understanding of the impact that storage network and storage devices have on applications. Is the SLO that the application promises to its clients achievable given the SLOs of the storage elements the application depends on? What coordination between the application and the underlying storage is necessary to achieve the data consistency requirements of the application?

SLO enforcement is another important client of a monitoring service. SLO enforcement monitors storage networks and nodes for violations of SLOs associated with these elements. SLO parameters may include performance and availability. This provides the ability to identify storage problems before they cause downtime and improves business continuity processes based on system behavior.

More active forms of SLO usage are possible. In some environments, Information Lifecycle Management (ILM) uses SLOs. Inflexion points in the lifecycle of a given data set are modeled and realized as SLO changes. These changes may or may not result in a change in the storage resources that ultimately host the data.

Monitoring involves more than simply accumulating chargeback information. It must also handle notification of unusual events and conditions. For example:

- How much storage is the order processing department using?
- Are there any performance issues that need to be addressed?
- Are any applications overusing their stated bandwidth allocation?
- Of the configured capacity allocated to an application, how much free space is remaining?
- Which host is filling up a given shared file system?
- Which host is clocking the CPU on a given storage resource?

4.2.4 Content Provisioning and Discovery

In a sense, searching the data in a grid for some given items is outside the scope of the GME. However, the architecture of a given grid may affect how this search is done. For example, a grid consisting entirely of fixed content storage will be searched in a much different way than a grid that provisions only file systems or LUNs. An example of content provisioning is:

- Is some given content "present" in a given container? Example: a federal judge has just asked for copies of all emails that contain the name of a specific securities company.

4.2.5 Billing and Chargeback

Billing and charge back enable customer billing and visibility into departmental storage costs. This allows for integration between an internal enterprise billing system and the storage management application for user charge back. Cost of storage will generally correspond to quality of storage. There are various ways of charging for storage; some enterprises charge for capacity that is actually used, while others charge per container, for example. There may be a separate provisioning charge as well, in addition to a "dollars per terabyte-day" kind of figure.

Chargeback rates may be an input for provisioning and capacity management to determine which tier of storage a department wishes to use. An example billing problem is:

- How much should be charged to Engineering for their use of grid resources in the past month?

5 Asset Management – A Deeper Dive

To illustrate our intended approach going forward, we will do a partial in-depth analysis of one of the functional areas laid out in the last section. A note about tools and protocols will help explain our intended direction.

5.1 CIM, SMI-S and GME Modeling

To represent the managed devices and management entities in a grid we need a set of modeling tools and management protocols that are able to span all of the required domains.

The EGA currently plans to base its work on CIM Modeling [DMTF] and the SNIA SMI-S specification [SMI-S]. CIM modeling provides classes, associations, and events while the SMI-S specification defines profiles that further specify behavior of class instances, as well as specific event types that must be supported. Note that WBEM has not been mentioned since the focus is on *modeling*, not actual access. Indeed, it is probable that actual invocation of management APIs to managed targets will include some mapping of CIM/SMI-S to web services or may use a few vendor proprietary APIs. In all cases, however, a model will be followed that is based on CIM/SMI-S.

One of the things we try and achieve with our model is to specify which aspects of the data provisioning problem fall exclusively under the purview of the GME. The definition of the GME in the Reference Model does not make this distinction. In this section and in the following one that describes asset management in greater detail we try and categorize the various services that may be present in a grid environment.

In general, we include in the GME any service that we consider fundamental to the functioning of the grid. Anything that does not meet this standard gets moved into the category of *grid-aware services*.

An example of a GME component is an asset database. No asset management function is possible without such a database – therefore we consider it to be a core GME component.

Services that we will not include in the GME include

- Any service that repackages what the GME provides for reasons of convenience or user-friendliness. A topology viewer is an example of such a service.
- Any service whose functionality is extremely dependent on the business environment. A billing service (which can differ significantly based on the billing model in use) falls into this category.

Clearly, our motivation to develop requirements for systems will be stronger for components that we consider to be part of the GME proper than for those grid-aware components that lie outside it.

The following diagram lays out what the protocol ecosystem is, in our opinion, likely to look like for the foreseeable future.

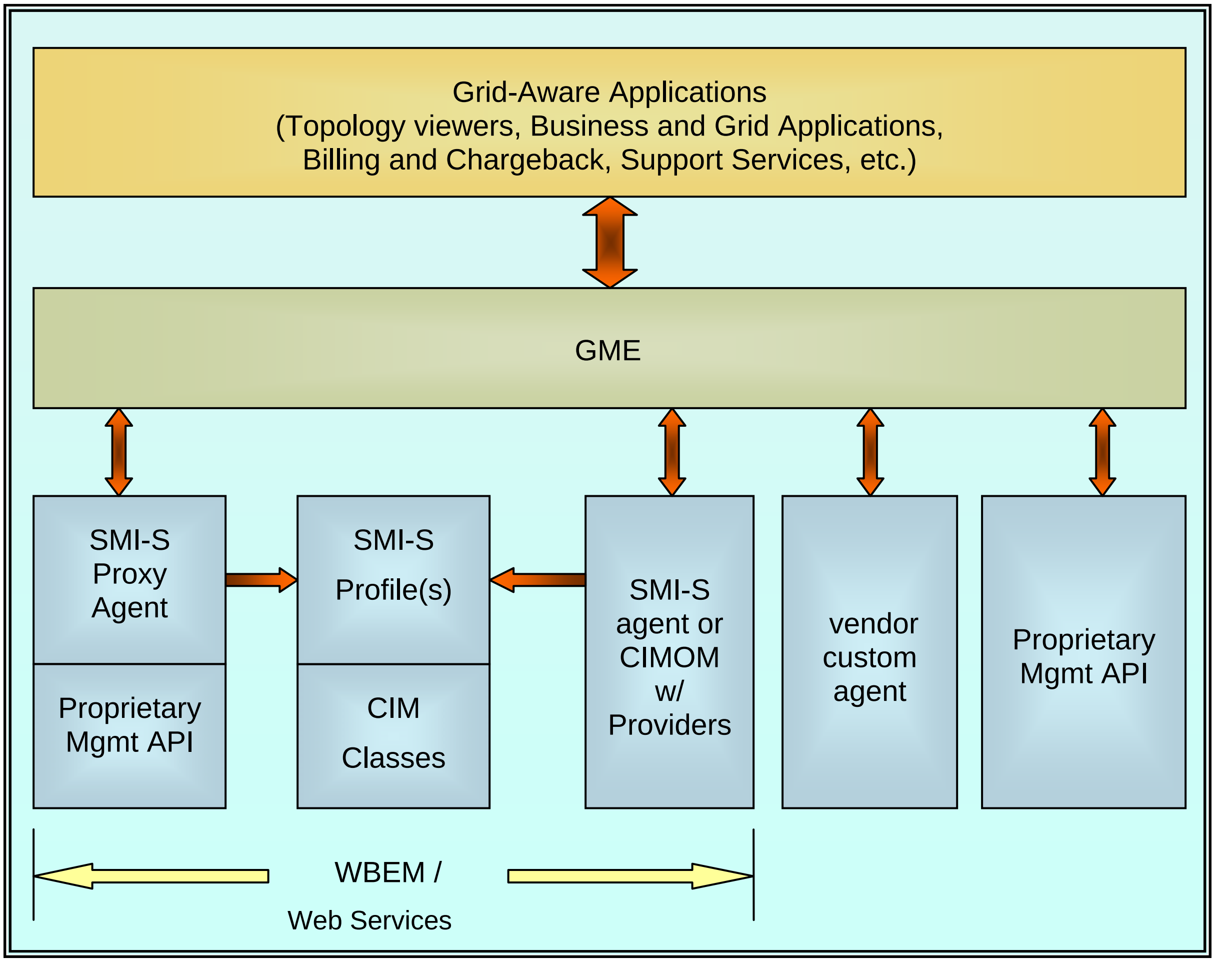

Figure ii - Protocol Ecosystem

Note that this model allows for proprietary APIs for managed targets as well as proprietary proxy agents that manage underlying managed targets. This is because while all existing data and storage management vendors profess to want a standard interface to managed devices, it is also universally true that they will bow to necessity when their business calls for it and use proprietary APIs. The need for this is decreasing, as vendors of managed target devices realize the business value in providing management via standard APIs, but is unlikely to disappear. There will always be vendors on the bleeding edge, for example, who have functionality that has not yet made it into a standard. It matters little in this case whether a vendor-specific extension to CIM is used to manage the feature, or some other API. The management application must still understand the semantics of this vendor-specific feature and manage it accordingly. Arguments over API syntax are missing the point, therefore, which is that vendor-specific value-added functionality that has not been standardized must be managed as such.

5.2 A Deeper Look – Asset Management

Having identified a number of functional modules, we wish to map them to standardized APIs that a GME or GME-aware service can use to implement the functionality needed for data provisioning. As an illustration of our intended approach, this section will briefly address the problem from the standpoint of Asset Management.

Asset Management as it relates to data provisioning encompasses several subtopics:

- Discovery
- Commissioning and decommissioning
- Capabilities as related to SLOs
- Chargeback units and quantities
- Security parameters related to access and use
- Data access protocol support
- Size and availability data (allocation statistics)
- Topology

In this section, we will discuss the first three items on the above list and defer discussion of the rest to a follow on paper.

The following diagram illustrates the relationship of the above list items within the context of the EGA Reference Model. This shows how the GME interacts with client applications and underlying resources to provide the *image* of the grid that clients will use.

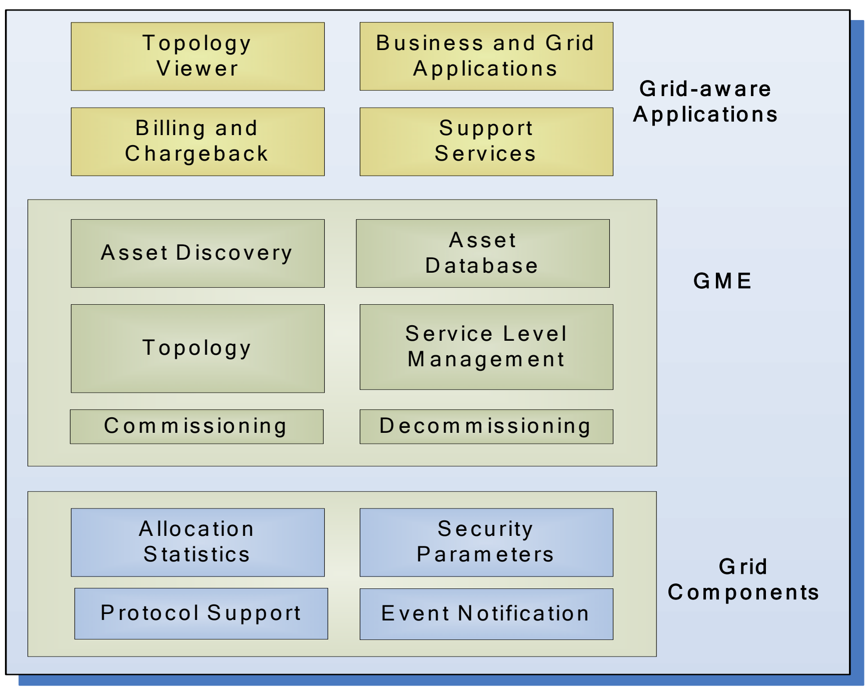

Figure iii - Asset Management in Context of EGA Reference Model

With respect to asset management, the underlying Storage Manager may have many other less high-level interests that are not listed above, e.g.:

- Ownership
- Physical location
- Vendor information
- Purchase information and depreciation schedules
- Upgrade and replacement schedules

- Physical security parameters
- Administrative contacts
- Software/firmware/hardware mappings
- Support costs
- Support contacts and parameters

These are details that must be considered when drawing up all the requirements for asset management.

SMI-S is the SNIA's storage management initiative specification. It is a large specification, composed mostly of *profiles* that specify server and client behavior. Clients use HTTP to make requests to CIMOMs (*CIM Object Managers*). The CIMOMs in turn use *providers* (code modules furnished by device vendors) to query or manage the server devices in question. Information is returned in CIM (*Common Information Model*) format. The DMTF (*Distributed Management Task Force*) at present owns most of the *schema* used by SMI-S. CIM is a "white box" system, meaning that the "model" of managed devices is exposed, instead of being concealed behind an opaque interface. The classes in the CIM schema are what comprise this model. Each profile defines a set of mandatory classes and properties that must be implemented for conformance with the profile.

A less dense way of saying the above is that clients use the SMI-S spec to figure out what a given server will support, what its resources are, and what services it offers. Between the specification and the CIM classes that it mandates, server behavior is defined in as standard a way as cooperating competitors are able to make it.

A very strong point of SMI-S is the extensive certification and testing infrastructure SNIA has built up around the SMI-S specification. The CTP (*Conformance Test Program*) and the SNIA Technology Center in Colorado Springs allow vendors to test against other vendor's gear to find interoperability problems. Frequent "plug fests" bring vendor engineers together face to face. Additionally, experience has shown that once a server vendor's profile passes the CTP test for that profile and a client vendor has successfully used the profile, there are likely to be few if any problems with profile implementations of other server vendors. This is a huge and unprecedented step forward in storage management.

5.2.1 Discovery

Asset discovery in the present context means finding all the assets that are of interest to a data provisioning service. This includes arrays, virtualizers, NAS devices, Web and FTP servers [potentially], Fibre Channel and Ethernet switches. For simplicity, lets assume that compute resources are discovered and provisioned separately. We consider switching to be part of data provisioning, however, as it is the delivery mechanism.

CIM environments provide discovery via SLP of CIMOMs, which in turn must discover endpoint resources. If one is relying on CIM, then the admin must ensure that all grid storage and switching resources are manageable via CIM. This is not a given as of the date of this paper. While all major vendors are moving aggressively towards this model, productization is just beginning, and full capability is probably still some years away.

Nonetheless, the Data Provisioning Working Group considers it to be the best available method going forward for those planning to implement a GME-based grid. Part of the charter of this paper and its follow-ons is to provide requirements for grid-based data provisioning to SNIA so that issues that have not already been addressed may be.

The traditional method of discovery for many devices is to use SNMP. By querying a well-known OID on any device with an ethernet interface and SNMP running, the device type may be determined. Similarly, there are in-band SCSI queries that can be used to discover resources on fibre-channel networks. These methods do not achieve as much as the CIM method, as there remains much correlation to hand-coded tables of vendor-specific functionality to be done.

Key things that must be found out—from a data provisioning point of view—during discovery are

- Server type (Storage, Switch)
- In-band access type (NFS, CIFS, HTTP, FTP, WebDAV, FC, IP) and addresses
- Management protocols and addresses
- Admin contact address and method

These are enough to a) decide what it is, b) find out what it can do, c) find out how to manage it and d) tell someone if attempts to do that fail.

More traditional SM and SRM applications will also be interested in finding out all of the items summarized in the first part of this section. Though one could think of obtaining that information also as discovery, we take a more narrow view here, and discuss the other items separately.

When assets are added dynamically, such as when more disks are added to an array, the asset management database needs to be brought into sync with the new reality. CIM provides *indications* for this purpose, and SMI-S specifies several mandatory indication filters that the asset management service can subscribe to for receiving notification of these events.

5.2.2 Commissioning and Decommissioning

In a large grid it is a given that resources will be added, broken and removed from the grid on a fairly frequent basis. Grids are somewhat like RAID arrays in that respect; regardless of the relative unreliability of the underlying disks, we expect the array itself, and especially the data on it, to be as dependable as gravity.

The same elements called out in discovery are sufficient for commissioning. However, it is not reasonable to expect the GME to do a complete scan of the environment periodically just to find out if new resources exist. There must also be a way of notifying the GME that new resources for discovery exist. Moreover, when a resource is decommissioned, the GME must be notified. One imagines that ideally, the GME will prevent final decommissioning of a resource until no other entity is actually using it anymore, but that is an implementation detail.

CIM offers *Indications*, which are alerts triggered by events such as "InstCreation" and "InstDeletion". By subscribing to these indications on objects of type CIM_ComputerSystem, the GME can receive notification of such events and act accordingly.

5.2.3 Capabilities as Related to SLOs

An SLO is a set of performance goals. The managers of a given device—generally the sysadmins at this point in time—match device capabilities against the SLOs required by applications.

So in order to find out whether a data resource can meet an SLO, its capabilities must be known. Key data SLO-related capabilities and characteristics are

- Available capacity
- Available transport protocols
- Security ACLs. These will be in the security context of the GME and control the rights of various principals and their applications to allocate resource. There may also be other ACLs related to other security domains.
- RPO (achievable *Recovery Point Objective* goals). Achieving the specified RPO generally involves snapshots, transaction logging or some form of local replication, and is ultimately the responsibility of the storage vendor's provider code to arrange. That is the theory and

the desire expressed by many in the storage community anyway. There are no automated implementations at present.

- RTO (achievable *Recovery Time Objective* goals). RTO goals are used to determine which underlying storage services will be necessary in the event of a disaster. As in the previous bullet, there is much talk of automating this, but little progress to date. A hybrid solution is in order.
- Availability. Traditionally this is measured in "nines", as in "five nines", which means "99.999% availability of my data" which in turn means "my data is available all but 5 minutes per year on average across storage devices of this type". Unfortunately, these numbers are generated by marketing departments, not by any kind of independent and comprehensive system of automated metrics, so the numbers are not often meaningful except within vendor product lines. Nonetheless, some indication of availability is needed.
- Chargeback parameters. These include provisioning fees and ongoing use fees.
- Projected lifetime. One of the difficulties of data provisioning is that data must be persistent, usually past the lifetime of the job, and moving it about is expensive. This means that it should be touched by the grid manager as little as possible. Accordingly, it makes little sense to provision a job with a projected end date of *x* on equipment whose decommissioning date is in advance of that.
- Data copy capabilities. The best copy in many cases is no copy, meaning the use of copy-on-write cloning or snapshotting techniques. If a full copy must be made, storage-to-storage or switch-based copies are preferred. If one of those cannot be arranged, a host side copy must be done.
- Data security capabilities. Encryption of data in flight or at rest may be offered. Many applications will require secure deletion of no-longer-wanted data as well. This can be done, in increasing order of security, through normal OS guarantees, via US Dept. of Defense-style overwriting of sectors, via cryptographic shredding, and finally by chipping the disks themselves into very small pieces. This last is unlikely to be useful in a grid environment, we feel, though in a geographically dispersed grid the level of physical security at each site may be an interesting factor.

There are other capabilities pertinent to storage, but the above list is in the main one data provisioning is concerned with.

CIM attempts to deal with SLO issues by defining *capabilities* that are attached to entities such as a CIM_StorageProvisioningService. These capabilities are matched up against *goals* specified by the client to determine whether a requested level of service can be met, and to provision the requested container if it can. This paradigm is used in SMI-S for both block-level storage and for file systems in the newer NAS profiles.

5.3 Grid Extensions for SNIA's Asset Management

Effective Grid Asset Management requires data for each managed element that includes the element's physical position, its street address, owner, group, administrative contacts, the logical location of its maintenance logs, its purchase date, purchase cost, support cost per support time unit, support renewal date, its scheduled retirement date, retirement method, its provisioning billing cost, per-unit-time billing cost, billable unit, billing unit, physical and logical security info, host hardware UUID, supported access and administrative protocols, and a list of required software names and versions. And this is not an all-inclusive list!

The EGA presumes that the logical place for this information is in the Physical Package and System profiles. The majority of it is not currently specified in the CIM Schema. So this is part of our ongoing work, to specify required enhancements to the CIM Schema, and SNIA's profiles, so these areas may be effectively managed in a Grid context.

6 Future Work

The Data Provisioning Working Group of the EGA plans to do a detailed analysis of each of the functional areas identified in this paper and publish the results. The asset management discussion illustrates the type of analysis required for the overall data provisioning. While it is not possible at present to manage all the managed targets in a grid storage infrastructure via SMI-S, the EGA and SNIA agree in principle that it should be. Therefore, the result of the overall analysis will include all the management requirements the group has identified for the given area, list correlations to SMI-S profiles and CIM classes for those already manageable by SMI-S, and then list missing properties and methods as well, with recommendations for possible modeling at SNIA and DMTF.

A key element of the current DMTF CIM Schema is that it attempts to be fully normalized. This can result in management data that might intuitively seem to belong together being spread out over many classes. It is a core characteristic of CIM, furthermore, that class instances do not contain pointers to other class instances, as in most object-oriented languages. Instead, *associations* (instances of association classes) are used to tie together related pieces of information. A query for what looks like a related set of data may as a consequence of the design of the CIM schema result in a query on the wire for each of a dozen class instances and separate queries to traverse a dozen or more associations. A proposal to implement a selected set of denormalized classes to cut down on some of this client chattiness is working its way through SNIA at present. These classes are called *view classes*; they correspond very closely in concept to SQL views, but may be implemented differently.

In some of the functional areas, it may be that necessary information is so spread out that the group may also recommend one or more view classes in its results.

7 Conclusion

It seems clear that much of the basic functionality required to interoperably manage grid components has been established by the work of the DMTF and the SNIA. However, a closer look at Asset Management intimates that there is almost certainly a large body of as yet unmet requirements that exist at a level of detail lower than the architects of CIM and SMI-S have yet considered. The EGA is resolved to address the identification of these requirements as a significant part of its charter going forward.

8 References

[DMTF] Common Information Model (CIM), Distributed Management Task Force. Homed at http://www.dmtf.org/standards/cim.

[EGA 1] The EGA Reference Model. Available on the public side of the EGA site at http://www.gridalliance.org/en/workgroups/ReferenceModel.asp

[GGF] The Global Grid Forum. Homed at www.ggf.org

[Globus] The Globus Toolkit. Homed at www.globus.org

[SNIA] The Storage Networking Industry Association. Homed at www.snia.org.

[SMI-S] The SNIA Storage Management Initiative Specification. This spec is currently in Member Review for version 1.1. Version 1.0 is on a fast track to be an ANSI standard, and will eventually become an ISO standard as well. Subsequent versions will follow the same path. The 1.1 spec is not available to the public at present, but all significant storage management vendors are members of SNIA and therefore have access to it at http://www.snia.org/members/viewcvs/smi/spec/SMIS.pdf. Version 1.2 of the spec is currently being scoped; preliminary estimates call for versions after 1.1 to arrive at roughly one year intervals.

Enterprise Grid Alliance
2400 Camino Ramon, Suite 375
San Ramon, CA 94583
Tel: +1.925.275.6644
http://www.gridalliance.org

