Document: gwd-gis-018-00
Category: Standards Track July 2001

# Grid Information Services for Resource Sharing

## Status

This document is a draft in the GISWG, a part of the Global Gridforum. This document's status and style are roughly similar to an IETF Internet-Draft and should be regarded in a similar fashion. In particular it is only a work in progress and should not be thought of as a standard endorsed by the Global Gridforum.



## Abstract

[Need a difft title for this and GISWG tag]

The document provides a specification for the Metacomputing Directory service of the Grid information service. It is not a complete specification of the MDS, it also functions as a roadmap or a framework for other documents that specify components of the MDS in detail, implementation examples, best practices, extensions, and the like.

This document is intended to provide the Grid community with sufficient understanding to build information services into resources that will interoperate with other Grid information servers.

The basic architectural specification is outlined, followed by sections giving a more detailed specification of various components. In these sections references to developing or planned documents will be given. Concluding sections and appendices will describe implementation advice and operational experience. These sections may also refer to documents that describe their subject matter in greater detail than permitted here.

It is not the intention at this time to enforce the rigid normative reference rules of the IETF; each document should stand on its own and travel through the Gridforum's acceptance policy on its own calendar, without affecting the status of any document.

This document is based closely on [GISDRS].

## Table of Contents

## 1 Introduction

Locating resources is a significant problem for participants in the grid. The Grid architecture as envisioned in "Anatomy of the Grid" [AOTG] and "Grid Information Services for Distributed Resource Sharing" [GISDRS] is a very distributed and rather dynamic collection of resources and services. Grid users, administrators, and the Grid services themselves need directories to keep track of these entities and to maintain relationships between them.

Readers of this document should be familiar with [GISDRS] and [AOTG]. "Anatomy of the Grid" provides the architectural context for GIS, describing its place in the hierarchy of layers: Grid information services are a significant part of the "Resource" and "Collective" layers. [AOTG] also defines and describes in detail the "Virtual Organization" (VO). For convenience here, we can define VO as a set of institutions, users, and resources, grouped together for some purpose on the Grid. Some alternate definitions of VO in the information services context will be suggested later. "Grid Information Services for Distributed Resource Sharing"

provides the architecture and much of the specification of this document, as well as considerable background material, use cases, argument for the architecture, and compare-and-contrast examples of similar services. This document will not delve into these topics but they are useful for understanding architecture in depth. [GISDRS] specifies a directory infrastructure rather than a single directory, an infrastructure that supports both resource location and resource monitoring. These requirements are mapped, where possible, to Internet standard protocols and services. In some cases additional protocols or extensions are specified.

The MDS architecture resembles the neck of an hourglass, using the metaphor for the protocol architecture in [AOTG]. At the "top", from the point of view of the user or the application, a large set of services is possible. At the neck, a very small number of mandatory components are specified. These components support a very large number of resources.

## 1.1 Goals

Experience has shown that Grid developers (developers of significant services or protocols) usually require a thorough understanding of the information services infrastructure to implement their service in an interoperable configuration. This document must provide them a clear, unambiguous specification of how the information services work, and what they have to do to make their service "GIS – enabled". In some cases the details they need will be found elsewhere, and this document will tell them where to find these detailed specifications. The architecture itself must be flexible enough to allow new services and resources with different characteristics to join, without requiring an overhaul of the GIS in place.

[A proof of this might be the demonstration of, eg , how to make an existing service with its own data storage based on Oracle GIS – enabled, or how to adapt NDS to work as one of the directory types.]

Data modeling and the management of meta-data is also a significant problem for developers, but this is out of this document's scope. [Except that we may want to support OID and/or schema registries, because of their optional support in [GISDRS]; see below].

## 1.2 Justification

The substantial argument for this architecture is in [GISDRS]. A simple version of it from [AOTG] bears repeating. The MDS overlaps the "Connectivity", "Resource", and "Collective" layers of the Grid protocol architecture, with its core in the Resource layer. The Resource layer provides the access and security layer between user (or other layers) and resource. The GIS role is to provide state and naming information about individual resources. A very small number of concepts (protocols and services) are chosen to perform this GIS function. These protocols are chosen for their robustness (or suitability), identification as industry or Internet standards, and simplicity. By focusing on the minimum set of concepts with these qualities, interoperability is assured.

This is never completely satisfactory. The variety of resources that need to be supported and the variety of uses to which these resources are put make some incompatibility likely, and the choices themselves are likely to have some imperfections. Allowing the GIS at the Resource layer to be infinitely extensible might meet most of these objections but sacrifice interoperability. MDS also extends upwards to the "Collective" layer, where an open specification is allowed. In the long run this may be found inadequate.

[Something to the effect that this is a reasonable and representative functionality; also describes a particular set; someone else can use this for some kind of dialectic]

### 1.3 Plan

The Architectural Specification will list a set of requirements (developed in [GISDRS]) and specify an architecture to satisfy them. This architecture will be developed in the context and terminology of the Grid architecture. The Component Specification will describe components and specifications for the essential Resource Layer components and a few [or one?] optional Collective Layer components, often by referencing a specification document for these components.

Specifications will conclude with Implementation Guidelines. Descriptions of representative implementations will conclude the document.

## 2 Architectural Specification

### 2.1 Requirements

Information is old. We cannot provide a consistent, known global state (state over the whole or even subsets of the Grid) [a Heisenbergian principle]. Alternative approaches that provide (on a local basis) a consistent view of global state are out of scope of this document.

Efficient delivery of state information from any one source is required.

Services must respond robustly to the failure of a component. This means that the failure of one component must not prevent obtaining information about other components. Further, components must assume that failure is the rule. The components are must be distributed and administration decentralized.

Services must timestamp failures.

The information service must support a few mandatory discovery and enquiry mechanisms.

The GIS must support a rich set of discovery and monitoring strategies, including:

Hierarchical resource groupings

Multiple naming schemes

Search strategies

Cost vs timelines algorithms

[Need help supporting this]

We require a clean separation between discovery and enquiry protocols

[note two dialectics here discovery <> monitoring, discovery <> enquiry.]

The GIS must support robust authentication and policy framework.

Data objects require time stamps and confidence estimate (TTL) attributes

Grid entities require an information server. Experience with alternate designs has shown that it is useful to separate the service that supports an individual entity’s reporting from the service that supports views of many different entities (particularly where these entities are large in number and the views are relatively complex). This is in accord with the Grid

architecture, which would place the first of these services in the Resource layer or below, and the second in the Collective layer.

[define entity here]

Two protocols are needed. An enquiry protocol supports queries of views of resources, and individual resources. A registration protocol identifies components of the registration service to each other.

There are two general kinds of applications that the information service needs to support: discovery applications and monitoring applications. Discovery applications ask for a snapshot in time, and ask for information explicitly [example]. Monitoring applications are concerned with variance over time and require change notification.

The information service needs to support a wide variety of these discovery and monitoring applications. Queries against the information service need to allow different balances of query expressiveness, timeliness, and cost. [V=Q+T+C]. The information service must not impose itself on Grid resources and services, and should require a minimum of changes to Grid resources and services (preferably none).

[The information service supports discovery of other grid services and resources. As a grid resource itself it needs to bootstrap this discovery process.]

[There is a discussion of globally unique identifiers and names that appears in the GRIP section, but I think belongs here. I think the requirement is that there MAY be GUIDs and OIDs, but they are not MUSTS. However, making the IP an LDAPv3 – conforming server makes OIDS mandatory. In that case there is a need for an OID registry. Perhaps that needs to be added to the LDAP-server sections below.

I think there is also another subtle derived mandatory requirement for naming but I need to think thru what GISDRS says about this more fully. The IP's must be VO-neutral. Neutrality is provided by using DNS names for the server? (Is that the rqmt/spec?) But do the base names for these IP servers need to be unique? Globus effectively does this in a non-mandatory way by building in the dns locale in two separate ways: one set in the "hn" component, and one in the "dc" components. What parts of this are mandatory and what parts optional?]

## 2.2 Specification

Timestamps and TTL's are covered by another document (or documents), which describe the two attributes and define them in terms useful to the other specified services and protocols and API's.

[There is a debate about dynamic vs static data that may be contributed here. There is also a specification document for an LDAP extension "Dynamic LDAP" [ref] that might be contributed in the components below; it would have to be optional.]

Factor the service into two layers: one layer is the Information Provider (GRIS) service, which supports the individual entity, and the other is the Aggregate Directory (GIIS) service.

[Need more complete explanation drawn from [GISDRS] to explain how all this works. Info flow diagram]

The enquiry protocol at the Resource layer must be a single protocol, to promote interoperability. It may contain other protocols at the Collective layer. The registration protocol at the Resource layer must be a single protocol to promote interoperability.

[It seems to me there is a null? protocol which is missing, which is the one between the GRAM and the Info Provider LDAP server. What should be done about this]

Alternate definitions of VO (functional definitions):

VO is a security policy scope.

VO is a search scope.

(These are not completely compatible.)

VO is isomorphic with a set of name-based AD's (in effect a VO is a common namespace).

[To support the discovery/bootstrap process, outline methods and implementation, either in a long section or additional document. This document should list several alternatives like the "taxonomy of discovery" ldap ietf draft. [GISDRS] discusses an SLP implementation.]

## 2.3 Security

GSI security must be implemented.

[This needs a lot of expansion or another document]

Information Providers implement the security policy of the resource they are serving, so that authentication and access control in the IP directory server must reflect that policy. To implement this, the GSI PKI must be implemented. [This should be described further in the LDAP or the security profile. Implementers of alternate directories will need to know how to do this.] Access control must reflect the GSI AC model. [which is]

[The GRIP protocol, which communicates between directories, needs to implement the GSI authZ and authN in some fashion, that is it needs to support the security model.]

[The GRRP protocol must support the GSI authZ model (that is the pki for authentication). Does it need to support authN at all? Either you register or you don't…; however the AD information mining means that it must be accorded a status as a grid user for the purpose of reading information from the IP with GRIP.]

# 3 Component Specification

## 3.1 Services

### 3.1.1 Information Providers (GRIS)

#### 3.1.1.1 Requirements

[I haven't identified many service-specific requirements for this from the paper. Assume:]

Info Provider must support Grid security.

Info Provider must implement support for GRAM (see above)

[Entities, or objects, need not be enumerable. This requirement is in the protocol section of [GISDRS].]

IP's need to know what AD's in which to register.

3.1.1.2 Specification

IP's must implement GSI.

IP's must implement the LDAP protocol (information, functional, and security models). The DIB (or data store) is not specified (which is acceptable to the LDAP standard). However, the functional model implies that the query methods, add, update, delete, the information model must be supported. This functional model may be implemented by a local data store (eg a conventional LDAP server); referrals of various types to other servers or services; a set of functions that perform operations and collect information that is returned in a manner conforming to the LDAP information model; or combinations of the above. Referrals may be LDAP knowledge references (the usual meaning of referral) or URL's to other services.

Server may support objects that are derived or otherwise not static or enumerable.

Discovery of AD techniques needed [lists, looking up in supporting service like SLP, DNS SRV, other info sources. Also discussed in another section involving bootstrapping]

3.1.2 Aggregate Directories (GIIS)

3.1.2.1 Requirements

A scope for searching is required.

Name-based location servers must be implemented.

Specialized AD's based on other criteria may be provided.

[AD's may need techniques for discovering IP's]

3.1.2.2 Specification

[AD's may be implemented as a special case of IP's?]

AD's provide a VO search scope.

AD's must use GRIP and GRRP to communicate with Information Providers.

AD's reply to queries about IP's. They may implement their query protocol as the LDAPv3 query (search) subset. [I think this should be expressed differently; LDAP should be the one mandatory to implement query protocol. This promotes interoperability but leaves the door open for more sophisticated or specialized queries, and other protocols.]

AD's must implement [at least the listener side?] of the GRRP protocol [this is the same as the first specification, only more specific?]

AD's may support any data model, query language, or set of protocols.

[AD's must also support GSI.]

AD's should mirror the organizational structure of the VO.

Specialized AD's are based on inherent name-based location AD's.

[AD's must implement GRIP (LDAPv3) for the purpose of querying IP's that have registered and extracting information. Should GRIP be a mandatory-to-implement protocol for AD's to promote interoperability? Recommended? Completely optional?]

[I believe there is a requirement for "shadowing" or replication. This could be derived from the architectural requirement that "failure of one component must not prevent obtaining information about other components". In terms of LDAP or any other information service this adds the burden of an altogether different protocol and service (both of which might be implemented in pure LDAP) with additional thorny security issues. Possibly this could be met by an additional AD, but this means that "replicas" as such would be less likely to be exact copies. ]

## 3.2 Protocols

### 3.2.1 Grid Information Protocol (GRIP)

#### 3.2.1.1 Requirements

GRIP is the core of the Resource Layer function of MDS.

GRIP must supply a security model capable of supporting GSI.

GRIP must support a rich information model.

GRIP must not require a specific storage model.

GRIP must not require enumerable entities.

GRIP must supply a query function.

GRIP must be deployed universally.

GRIP should support distributed operations.

GRIP must not require a consistent global state [or consistency as defined by ACID].

#### 3.2.1.2 Specification

These requirements are best met by defining GRIP as a subset of the LDAPv3 protocol specification [reference]. GRIP requires the LDAPv3 information and security models. LDAPv3 provides a rich query function but does not implement a "join" function or complex query language. LDAPv3 provides a simple API and numerous SDK's are available for it.

[This section 3.2 requires a PROFILE document.]

### 3.2.2 Grid Registration Protocol (GRRP)

#### 3.2.2.1 Requirements

GRRP notifies one directory server about the existence of another.

GRRP must not require

#### 3.2.2.2 Specification

GRRP is implemented as a "soft-state" protocol: relationship is established and maintained by notifications, and discarded after a long period of silence.

GRRP may be implemented on a variety of transport protocols. A sample LDAP implementation is provided based on the LDAP update function and SASL security.

The GRRP protocol is described in [GRRP]. The status of this document is GISWG draft.

### 3.2.3 Time protocol

[We can argue about this one]

A common time standard is required in a Virtual Organization, so that timestamps and time-to-live attributes are meaningful.

NTP [ref] is a mandatory Resource level protocol inside a Virtual Organization.

NTP configuration in a VO must converge to a common system time.

[This requires some kind of profile document. See comment about time stamps above.]

## 3.3 API's, SDK's, and Developer services

### 3.3.1 LDAP API's

GRIP is always implemented as LDAP, and GRRP is currently implemented as LDAP. Therefore the LDAP API as specified by [ref] is required. There are numerous SDK's implementing various language bindings of the API . The Globus Toolkit distributes [requires?] the OpenLDAP [ver] C SDK. [List some others?]

### 3.3.2 Aggregate Directory API's

[This should be split into LDAPv3 : the Globus toolkit supports this; and enumerate other contributions as document references; for example if someone has an alternative AD implementation based on an RDBMS and ODBC the primary document for this would probably be in the AD section above, but another reference might be here.]

### 3.3.3 Other

[DSML? Or out of scope]

### 3.3.4 [NTP]

[API is not required. Services require that computer systems implement ntp and provide services with correct time.]

# 4 Implementation Guidelines

[This section explains how a developer might configure and adapt other protocols, SDK's, and services to interoperate with other Grid information services. In particular the adaptation of OpenLDAP might be described in greater detail. Issues of data modeling and schema differences between various kinds of services and how to cope with this might be covered. Or a description of how a complex service like NWS or data grid cataloguing is adapted to this architecture could be described. But this section should finish the work of the previous sections rather than developing a use case.]

# 5 Implementation Experiences and Best Practices

[This section describes experiences different groups have had developing services based on this architecture. {GISRS} describes the MDS-2.x service

in Globus, and to a lesser extent NWS. An NWS developer has also expressed interest in describing their experience in more detail.]

5.1 MDS-2

5.2 NWS

6 Conclusion

Acknowledgements

Editor's Address

Michael Helm
ESnet/LBNL
M/S 50a-3111
Lawrence Berkeley National Laboratory
Berkeley CA 94720
mailto:helm@fionn.es.net

References

[GISDRS] Czajkowski, Fitzgerald, Foster, Kesselman, “Grid Information Service for Dynamic Resource Sharing”, HPDC-10, August 2001

[AOTG] Foster, Kesselman, Tuecke, “Anatomy of the Grid”, 2001