GWD-R, GWD-I or GWD-C
Authors:

Sergio Andreozzi* (editor), INFN
Stephen Burke, RAL
Felix Ehm, CERN
Laurence Field*, CERN
Gerson Galang, ARCS
Balazs Konya*, Lund University
Maarten Litmaath, CERN
Paul Millar, DESY
JP Navarro, ANL
*co-chairs

GLUE WG
http://forge.ogf.org/sf/sfmain/do/viewProject/projects.glue-wg

May 20, 2008

GLUE Specification v. 2.0

Status of This Document

This document provides information to the Grid community regarding the specification of the GLUE information model. Distribution is unlimited. This document is based on the document version 42 as available in the GLUE Working Group document repository.

Copyright Notice

Copyright © Open Grid Forum (2008). All Rights Reserved.

Trademark

Open Grid Services Architecture and OGSA are trademarks of the Open Grid Forum.

Abstract

The GLUE specification is an information model for Grid entities described using natural language and enriched with a graphical representation using UML Class Diagrams. As a conceptual model, it is designed to be independent from the concrete data models adopted for its implementation. Rendering to concrete data models such XML Schema, LDAP Schema and SQL are provided in a separate document.

Contents

1. Introduction 4
2. Notational Conventions 4
3. General Statements 4
4. Template 5
5. Conceptual Model of the Main Entities 6
5.1 Entity 7
5.2 Extension 7
5.3 Location 7
5.4 Contact 9
5.5 Domain 9
5.5.1 AdminDomain 10
5.5.2 UserDomain 10
5.6 Service 11
5.7 Endpoint 12
5.8 Share 12
5.9 Manager 12
5.10 Resource 12
5.11 Activity 12
5.12 Policy 12
5.12.1 AccessPolicy 12
5.12.2 MappingPolicy 12
6. Conceptual Model of the Computing Service 12
6.1 ComputingService 12
6.2 ComputingEndpoint 12
6.3 ComputingShare 12
6.4 ComputingManager 12
6.5 Benchmark 12
6.6 ExecutionEnvironment 12
6.7 ApplicationEnvironment 12
6.8 ApplicationHandle 12
6.9 ComputingActivity 12
6.10 ToStorageService 12
7. Conceptual Model of the Storage Service 12
7.1 StorageService 12
7.2 StorageServiceCapacity 12
7.3 StorageAccessProtocol 12
7.4 StorageEndpoint 12
7.5 StorageShare 12
7.6 StorageShareCapacity 12
7.7 StorageManager 12
7.8 StorageResource 12
7.9 ToComputingService 12
8. Relationship to OGF Reference Model 12
9. Security Considerations 12
10. Author Information 12
11. Contributors & Acknowledgements 12
12. Intellectual Property Statement 12
13. Disclaimer 12
14. Full Copyright Notice 12
15. References 12
16. Appendix A: Place-holder values for unknown data 12
16.1 Use cases 12
16.2 Place-holder values 12
16.2.1 Fully qualified domain names 12

16.2.2 IPv4 address 12
16.2.3 IPv6 addr..... 12
16.2.4 Integers 12
16.2.5 File path 12
16.2.6 Email addresses 12
16.2.7 Uniform Resource Identifier (URI) 12
16.2.8 X509 Distinguished Names 12
16.2.9 Fully Qualified Attribute Name (FQAN)..... 12
16.2.10 Geographic locations..... 12
17. Appendix B: Data Types 12
17.1 LocalID_t 12
17.2 ContactType_t 12
17.3 PolicyScheme_t..... 12
17.4 DN_t..... 12
17.5 Capability_t 12
17.6 ServiceType_t..... 12
17.7 QualityLevel_t 12
17.8 EndpointTechnology_t..... 12
17.9 EndpointHealthState_t..... 12
17.10 ServingState_t 12
17.11 DateTime_t 12
17.12 Staging_t 12
17.13 JobDescription_t 12
17.14 SchedulingPolicy_t 12
17.15 ReservationPolicy_t..... 12
17.16 ComputingManagerType_t 12
17.17 NetworkInfo_t 12
17.18 Benchmark_t 12
17.19 Platform_t 12
17.20 CPUMultiplicity_t 12
17.21 OSFamily_t..... 12
17.22 ParallelSupport_t 12
17.23 AppEnvState_t..... 12
17.24 ApplicationHandle_t..... 12
17.25 OSName_t..... 12
17.26 License_t 12
17.27 ComputingActivityType_t..... 12
17.28 ComputingActivityState_t 12
17.29 StorageCapacity_t..... 12
17.30 StorageAccessProtocol_t 12
17.31 AccessLatency_t 12
17.32 RetentionPolicy_t..... 12
17.33 ExpirationMode_t..... 12
17.34 StorageManagerType_t..... 12
17.35 StorageResourceType_t..... 12

1. Introduction

In this document, we present a conceptual information model for Grid entities described using natural language and enriched with a graphical representation using UML Class Diagrams. As a conceptual model, it is designed to be independent from the concrete data models adopted for its implementation. Rendering to concrete data models such XML Schema, LDAP Schema and SQL are provided in a separate document. From the semantic viewpoint, the concrete data models should represent the same concepts and relationships of the conceptual information model; nevertheless they can contain simplifications targeted at improving query performance or other aspects of interest.

This information model is based on the experience of several modeling approaches being used in current production Grid infrastructures (e.g., GLUE Schema 1.x [glue-1.x], NorduGrid schema [ng-schema], Naregi model [naregi-schema]). The main supporting use cases are collected in the use cases document [glue-usecases].

2. Notational Conventions

The key words “MUST”, “MUST NOT,” “REQUIRED,” “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” are to be interpreted as described in RFC 2119 (see http://www.ietf.org/rfc/rfc2119.txt).

3. General Statements

The Information Model and its renderings MUST be considered case-sensitive. Each GLUE entity MUST have either an ID or LocalID attribute (except is made for the Entity and the Extension classes). The ID is a global identifier, while the LocalID is an identifier local to a container entity which is specified in the definition. Both ID and LocalID SHOULD be persistent, in the sense that they SHOULD NOT change during the life of the related entity. They are needed for recognition or for access to the characteristics of the related entity over time and across different information sources.

Both ID and LocalID MUST NOT be interpreted by the user or the system as having any meaning other than an identifier. In particular, there is no relationship between an ID and a network endpoint. All ID property values must be valid URIs. The usage of URN (Uniform Resource Name, a subset of Uniform Resource Identifier or URI) is RECOMMENDED.

As regards unit of measure, multiple of bytes MUST refer to the SI (*Le **S**ystème **I**nternational d'Unités*) prefix (http://en.wikipedia.org/wiki/SI_prefix), therefore GB is 10^9 Bytes and not 2^{30} Bytes (the latter are GibiBytes).

In Appendix 16, we provide guidelines for place-holder values that MUST be used when the attributes have no good default value or when the information provider is unable to obtain a dynamic value.

The terms “attribute” and “property” MUST be considered synonyms in the scope of this document.

As regads the extensibility, two main approaches are introduced to extend the information associated to the existing classes: the OtherInfo property and the Extension class. The OtherInfo

property is present in several classes, its type is string and is multiplicity is *. This SHOULD be used for associating a flat list of tags to a certain class instance.
The Extension class is associated to the Entity class (therefore also to all the derived classes) and enables to link key,value pairs to any GLUE class instance. This SHOULD be used when there is the need for advertising a more structured information, for instance an attribute not present in the model with the related value.

Both solutions are proposed because they have a different impact in the implementations: the OtherInfo approach is easier to query, nevertheless it may require parsing in case of concatenation of different chunks of information (e.g., attribute name and attribute value). The Extension class offers a two-dimensional construct, nevertheless it is more complex to query.

The extensibility regarging the addition of new classes and associations is not supported at the conceptual level. We RECOMMEND to create specialization of the conceptual model and to implement them by extending the concrete data models.

4. Template

In order to enrich the UML Class Diagrams with additional information, a table for each UML class is provided. The descriptive table si composed by three parts.

The first part refers to the whole entity and presents the entity name, the entity from which it inherits and the description of what the entity is.

The second part refers to the properties of the class; for each of them, the following characteristics are described: the property name, the data type, the multiplicity concerning how many values are allowed (* means zero or more), the unit of measurement and a description. For easy of reading, the properties that are inherited from a parent class are also listed.

The third part refers to the associations (association, composition, aggregation or association class) that the class may hold with other classes. For each association, the associated class endpoint is described in terms of the associated end class and key property, the multiplicity (i.e., the number of instances of the associated class that are allowed) and a description. The inherited associations are also reported in the "inherited association end" if they are not redefined in the "association end". The template structure is the following:

Entity	Inherits from			Description
Inherited Property	Type	Mult.	Unit	Description
Property	Type	Mult.	Unit	Description
Association End			Description	
Inherited Association End		Mult.	Description	

5. Conceptual Model of the Main Entities

This section introduces the main entities of the GLUE information model. They capture the core concepts relevant in a Grid environment. The main entities SHOULD be used to derive specialized information models. In Figure 1, the classes and the related relationships are presented in the form of a UML Class Diagram.

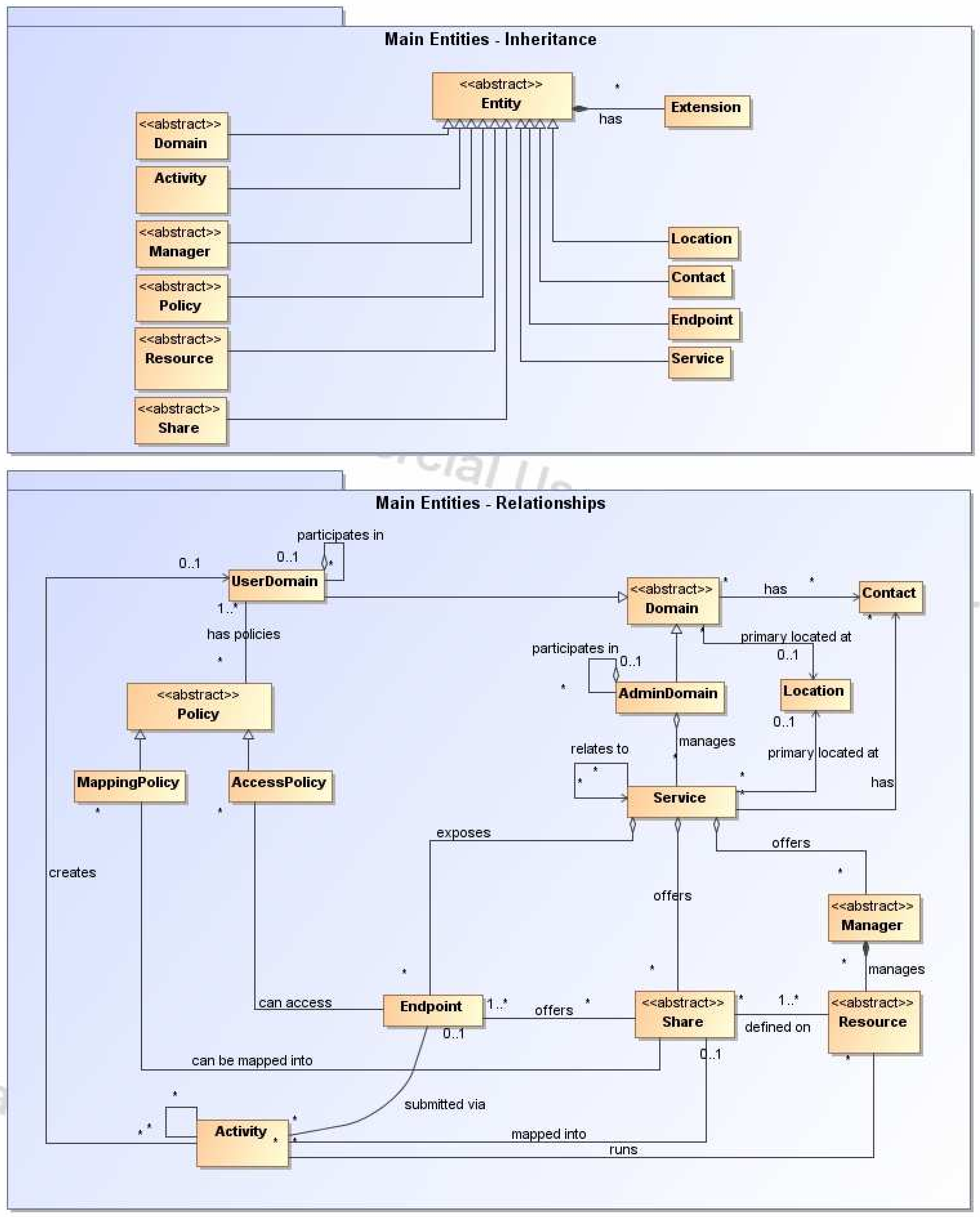

Figure 1 Entities and relationships for the Main Entities conceptual model

Entity

Entity	Inherits from			Description
Entity <<abstract>>				Abstract root concept from which all the other concepts are derived (except the Extension class); it has metadata about information creation and validity plus a key-value pair extension mechanism
Property	Type	Mult.	Unit	Description
CreationTime	DateTime_t	0..1		Timestamp describing when the entity instance was generated
Validity	UInt64	0..1	s	The duration after CreationTime that the information presented in the Entity MAY be considered relevant. After that period has elapsed, the information SHOULD NOT be considered relevant
Association End		Mult.	Description	
Extension.Key		*	The entity can be associated to zero or more key-value pairs	

This entity is the root entity from which all the GLUE classes inherit (exception is made for the Extension class). The specialized classes will inherit both the association to the Extension class and both the properties CreationTime and Validity. While the inheritance to the Extension class is reported in each table, the inheritance of the two properties is not explicitly listed.

5.1 Extension

Entity	Inherits from			Description
Extension				A key,value pair enabling the association of extra information not captured by the model with an Entity instance
Property	Type	Mult.	Unit	Description
Key	String	1		An identifier local to the container class instance; typically an attribute name not present in the model; this identifier is not supposed to be unique; several instances of this class can hold the same value for this attribute
Value	String	1		A value for the attribute
Association End		Mult.	Description	
Entity		1	The key, value pair is associated to an entity instance	

5.2 Location

Entity	Inherits from			Description
Location	Entity			A geographical position
Property	Type	Mult.	Unit	Description
LocalID	LocalID_t	1		An opaque identifier local to the associated Service or Domain
Name	String	1		A human-readable name
Address	String	0..1		Street address
Place	String	0..1		Name of town/city
Country	String	0..1		Name of the country
PostCode	String	0..1		Postal code
Latitude	Real32	0..1	degree	The position of a place north or south of the equator measured from -90°to +90°with positive values going north and negative values going south
Longitude	Real32	0..1	degree	The position of a place east or west of the primary meridian (located in Greenwich, UK) measured from -180°to +180°with positive values going east and negative values going west (the value -180°is excluded from the range)
Association End		Mult.	Description	
Service.ID		*	The location is related to zero or more services	
Domain.ID <<abstract>>		*	The location is related to zero or more domains	
Inherited Association End		Mult.	Description	
Extension.Key		*	The entity can be associated to zero or more key-value pairs	
ComputingService.ID		*	The location is related to zero or more computing services	

StorageService.ID	*	The location is related to zero or more storage services
AdminDomain.ID	*	The location is related to zero or more admin domains
UserDomain.ID	*	The location is related to zero or more user domains

The location entity can be used for describing geographical positions of domains and services. The aim is to provide a simple way to express geographical information and it is not intended to be used in complex geographical information systems. The accuracy of latitude and longitude should be defined in a future interoperability profile defined by projects adopting this specification.

5.3 Contact

Entity	Inherits from			Description
Contact	Entity			Information enabling to establish a communication with a person or group of persons part of a domain
Property	**Type**	**Mult.**	**Unit**	**Description**
LocalID	LocalID_t	1		An opaque identifier local to the associated Service or Domain
URL	URI	1		URL embedding the contact information. The syntax of URI depends on the communication channel
Type	ContactType_t	1		Type of contact
OtherInfo	String	*		Placeholder to publish info that does not fit in any other attribute. Free-form string, comma-separated tags, (name, value) pair are all examples of valid syntax
Association End		**Mult.**		**Description**
Service.ID		*		The contact is related to zero or more services
Domain.ID <<abstract>>		*		The contact is related to zero or more domains
Inherited Association End		**Mult.**		**Description**
Extension.Key		*		The entity can be associated to zero or more key-value pairs
ComputingService.ID		*		The contact is related to zero or more computing services
StorageService.ID		*		The contact is related to zero or more storage services
AdminDomain.ID		*		The contact is related to zero or more admin domains
UserDomain.ID		*		The contact is related to zero or more user domains

This entity can be used to represent contact information for requests related to different areas (e.g., user support, security or sysadmin). The various types of contact are identified by the Type attribute. In case of time-depend contact information, the instances of this entity should represent only the active contact information.

There are several specifications recommending how to embed contacts into URI. The following specifications SHOULD be used:

- telephone and fax: http://www.ietf.org/rfc/rfc2806.txt
- email: http://www.ietf.org/rfc/rfc2368.txt
- irc: http://www.w3.org/Addressing/draft-mirashi-url-irc-01.txt

5.4 Domain

Entity	Inherits from			Description
Domain <<abstract>>	Entity			A collection of actors that can be assigned with roles and privileges to entities via policies. A domain may have relationships to other domains.
Property	**Type**	**Mult.**	**Unit**	**Description**
ID [key]	URI	1		A global unique ID
Name	String	0..1		Human-readable name
Description	String	0..1		A description of the domain
WWW	URI	*		The URL identifying a web page with more information about the domain
OtherInfo	String	*		Placeholder to publish info that does not fit in any other attribute. Free-form string, comma-separated tags, (name, value) pair are all examples of valid syntax
Association End		**Mult.**		**Description**
Contact.LocalID		*		A domain can be contacted via zero or more contacts
Location.LocalID		0..1		A domain is primary located at one location
Association End		**Mult.**		**Description**
Extension.Key		*		The entity can be associated to zero or more key-value pairs

This is an abstract entity and it MUST NOT be instantiated. It SHOULD be used in order to derive specialized entities.

5.4.1 AdminDomain

Entity	Inherits from	Description
AdminDomain	Domain	A collection of actors that MAY be assigned administrative roles and privileges to services via policies. An AdminDomain manages services that can be geographically distributed, nevertheless a primary location should be identified.

Inherited Property	Type	Mult.	Unit	Description
ID [key]	*URI*	*1*		*A global unique ID*
Name	*String*	*0..1*		*Human-readable name*
Description	*String*	*0..1*		*A description of the domain*
WWW	*URI*	*		*The URL identifying a web page with more information about the domain*
OtherInfo	*String*	*		*Placeholder to publish info that does not fit in any other attribute. Free-form string, comma-separated tags, (name, value) pair are all examples of valid syntax*

Property	Type	Mult.	Unit	Description
Distributed	Boolean	0..1		True if the services managed by the admindomain are considered geographically distributed by the administrators themselves
Owner	String	*		Identification of the person or legal entity which pays for the services and resources

Association End	Mult.	Description
Service.ID	*	An AdminDomain manages zero or more Services
AdminDomain.ID	*	An AdminDomain aggregates zero or more AdminDomains
AdminDomain.ID	0..1	An AdminDomain participates in another AdminDomain

Inherited Association End	Mult.	Description
Extension.Key	*	The entity can be extended via key-value pairs
ComputingService.ID	*	An AdminDomain manages zero or more Computing Services
StorageService.ID	*	An AdminDomain manages zero or more Storage Services
Contact.LocalID	*	A domain can be contacted via zero or more contacts
Location.LocalID	0..1	A domain is primary located at one location

An AdminDomain can be composed by other AdminDomains in a hierarchical structure. This structure MAY represent a “participates in” association.

5.4.2 UserDomain

Entity	Inherits from	Description
UserDomain	Domain	A collection of actors that can be assigned with user roles and privileges to services or shares via policies

Inherited Property	Type	Mult.	Unit	Description
ID [key]	*URI*	*1*		*A global unique ID*
Name	*String*	*0..1*		*Human-readable name*
Description	*String*	*0..1*		*A description of the domain*
WWW	*URI*	*		*The URL identifying a web page with more information about the domain*
OtherInfo	*String*	*		*Placeholder to publish info that does not fit in any other attribute. Free-form string, comma-separated tags, (name, value) pair are all examples of valid syntax*

Property	Type	Mult.	Unit	Description
Level	UInt32	0..1		The number of hops to reach the root for hierarchically organized domains described by the “composed by” association (0 is for the root)
UserManager	URI	*		The Endpoint ID managing the users part of the domain and the related attributes such as groups or roles
Member	String	*		An identifier for a user in this user domain

Association End	Mult.	Description
Policy.ID <<abstract>>	*	A User Domain has associated zero or more policies
UserDomain.ID	*	A User Domain aggregates zero or more User Domains
UserDomain.ID	0..1	An User Domain participates in another User Domain

Inherited Association End	Mult.	Description
Extension.Key	*	The entity can be extended via key-value pairs
Contact.LocalID	*	The domain can be contacted via zero or more contacts
Location.LocalID	0..1	A domain is primary located at one location
AccessPolicy.ID	*	A User Domain has associated zero or more access policies
MappingPolicy.ID	*	A User Domain has associated zero or more mapping policies

In the GLUE Information Model, the Virtual Organization can be realized by using the concept of UserDomain. If the VO has an internal structure, this can be represented by using different domains related to each other. A Virtual Organization (VO) comprises a set of individuals and/or institutions having direct access to computers, software, data, and other resources for collaborative problem-solving or other purposes. Resources utilized by a VO are expected to be accessible via network endpoints and constrained by defining utilization targets called shares. The VO can exhibit the internal structure in terms of groups of individuals, each of them being a UserDomain. UserDomains can be hierarchically structured. This structure can be represented via the “participates in” association.

As regards the UserManager, a commonly used implementation is the VOMS (Virtual Organization Membership Service, http://voms.forge.cnaf.infn.it/) .

5.5 Service

Entity	Inherits from	Description
Service	Entity	An abstracted, logical view of actual software components that participate in the creation of an entity providing one or more functionalities useful in a Grid environment. A service exposes zero or more endpoints having well-defined interfaces, zero or more shares and zero or more managers and the related resources. The service is autonomous and denotes a weak aggregation among endpoints, the underlying managers and the related resources, and the defined shares. The service enables to identify the whole set of entities providing the functionality with a persistent name.

Property	Type	Mult.	Unit	Description
ID [key]	URI	1		A global unique ID
Name	String	0..1		Human-readable name
Capability	Capability_t	1..*		The provided capability according to the Open Grid Service Architecture (OGSA) architecture [OGF-GFD80] (it is given by the sum of all the capabilities provided by the related endpoints)
Type	ServiceType_t	1		The type of service according to a middleware classification
QualityLevel	QualityLevel_t	1		Maturity of the service in terms of quality of the software components
StatusPage	URI	*		Web page providing additional information like monitoring aspects
Complexity	String	0..1		Human-readable summary description of the complexity in terms of the number of endpoint types, shares and resources. The syntax should be: endpointType=X, share=Y, resource=Z.
OtherInfo	String	*		Placeholder to publish info that does not fit in any other attribute. Free-form string, comma-separated tags, (name, value) pair are all examples of valid syntax

Association End	Mult.	Description
Endpoint.ID	*	A service exposes zero or more endpoints
Share.LocalID <<abstract>>	*	A service offers zero or more shares
Manager.ID <<abstract>>	*	A service offers zero or more managers
Contact.ID	*	A service has zero or more contacts
Location.ID	0..1	A service is primary located at a location
Service.ID	*	A service is related to zero or more services
Service.ID	*	A service is related to zero or more services
Inherited Association End	Mult.	Description
Extension.Key	*	The entity can be extended via key-value pairs

The simplest Service aggregates an endpoint, no share, no manager and no resource (e.g., a metadata catalog service). In the context of a Service, the same resource can be exposed via multiple endpoints based on the defined shares. For instance, in the area of storage systems, two endpoints implementing SRMv1 [srmv1] and SRMv2.2 [srmv2] interfaces respectively can expose the same resource via different endpoints offering different interface version; in the area of computing systems, the CREAM [cream] and GRAM [gram] endpoints can expose the resources locally managed by the same manager (typically a batch system). Endpoints, Shares, Managers and Resources MUST belong to precisely one service.

5.6 Endpoint

Entity	Inherits from			Description
Endpoint	Entity			A network location having a well-defined interface and exposing the service functionalities
Property	Type	Mult.	Unit	
ID [key]	URI	1		A global unique ID
Name	String	0..1		Human-readable name
URL	URI	1		Network location of the endpoint to contact the related service
Capability	Capability_t	1..*		The provided capability according to the OGSA architecture
Technology	EndpointTechnology_t	0..1		Technology used to implement the endpoint
Interface	URI	1		Identification of a type and version of the interface
InterfaceExtension	URI	*		Identification of an extension to the interface
WSDL	URI	*		URL of the WSDL document describing the offered interface (applies to Web Services endpoint)
SupportedProfile	URI	*		URI identifying a supported profile
Semantics	URI	*		URI of a document providing a human-readable description of the semantics of the endpoint functionalities
Implementor	String	0..1		Main organization implementing this software component
ImplementationName	String	0..1		Name of the implementation
ImplementationVersion	String	0..1		Version of the implementation (the syntax MAY be: major.minor.patch)
QualityLevel	QualityLevel_t	1		Maturity of the endpoint in terms of quality of the software components
HealthState	EndpointHealthState_t	1		A state representing the health of the endpoint in terms of its capability of properly delivering the functionalities
HealthStateInfo	String	0..1		Textual explanation of the state endpoint
ServingState	ServingState_t	1		A state specifying if the endpoint is accepting new requests and if it is serving the already accepted requests
StartTime	DateTime_t	0..1		The timestamp for the start time of the endpoint
IssuerCA	DN_t	0..1		Distinguished name of Certification Authority issuing the certificate for the

				endpoint
TrustedCA	DN_t	*		Distinguished name of the trusted Certification Authority (CA), i.e., certificates issues by the CA are accepted for the authentication process
DowntimeAnnounce	DateTime_t	0..1		The timestamp for the announcement of the next scheduled downtime
DowntimeStart	DateTime_t	0..1		The timestamp describing when the next downtime is scheduled to start
DowntimeEnd	DateTime_t	0..1		The timestamp describing when the next downtime is scheduled to end
DowntimeInfo	String	0..1		Description of the next scheduled downtime

Association End	Mult.	Description
Service.ID	1	An endpoint is part of a Service
Share.LocalID <<abstract>>	*	An endpoint can pass activities to zero or more Shares
AccessPolicy.ID	*	An endpoint has associated zero or more AccessPolicies
Activity.ID	*	An endpoint has accepted and is managing zero or more Activities
Inherited Association End	**Mult.**	**Description**
Extension.Key	*	The entity can be extended via key-value pairs

For Grid services requiring a richer set of properties for the endpoint, specific models can be derived by specializing from the Endpoint entity and adding new properties or relationships. The current proposal contains the ComputingEndpoint specialization (see Section 6.2) and the StorageEndpoint specialization (see Section 7.4).

The endpoint network location MUST be encoded as a URI. When available, standard schemes for the encoding SHOULD be used (e.g., for Java Messaging Service http://www.ietf.org/internet-drafts/draft-merrick-jms-uri-03.txt).

Concerning the SupportedProfile property, if there is no recommended URI for the identification of a certain profile, then suggestions for choosing them are: main URL of the document specifying the profile or target namespace URI (in case of XML Schema representation of the profile).

5.7 Share

Entity	Inherits from	Description
Share <<abstract>>	Entity	A utilization target for a set of resources managed by a local manager and offered via related endpoints. The share is defined by configuration parameters and characterized by status information

Property	Type	Mult.	Unit	Description
LocalID [key]	LocalID_t	1		An opaque identifier local to the associated Service
Name	String	0..1		Human-readable name
Description	String	0..1		Description of this share

Association End	Mult.	Description
Endpoint.ID	1..*	A share is consumed via one or more endpoints
Resource.ID <<abstract>>	1..*	A share is defined on one or more resources
Service.ID	1	A share participates in a service
Activity.ID	*	A share is consumed by zero or more activities
MappingPolicy.ID	*	A share has zero or more mapping policies
Inherited Association End	**Mult.**	**Description**
Extension.Key	*	The entity can be extended via key-value pairs

This is an abstract entity and it MUST NOT be instantiated. It SHOULD be used in order to derive specialized entities.

5.8 Manager

Entity	Inherits from			Description
Manager <<abstract>>	Entity			A software component locally managing one or more resources. It can describe also aggregated information about the managed resources.
Property	Type	Mult.	Unit	Description
ID [key]	URI	1		A global unique ID
Name	String	0..1		Human-readable name
Association End		Mult.	Description	
Service.ID		1	A manager participates in a service	
Resource.ID	<<abstract>>	1..*	A manager manages zero or more resources	
Inherited Association End		Mult.	Description	
Extension.Key		*	The entity can be extended via key-value pairs	

This is an abstract entity not meant to be instantiated. It SHOULD be used in order to derive specialized entities.

The manager refers typically to a local manager service which specific details are abstracted by a middleware software component (endpoint). Examples of managers are: for computing resources, batch systems such as OpenPBS or LSF; for storage resources, GPFS.

5.9 Resource

Entity	Inherits from			Description
Resource <<abstract>>	Entity			An entity providing a capability or capacity, managed by a local software component (manager), part of a logical service, reachable via one or more endpoints and having one or more shares defined on it. A resource can refer to a category with summary information on the available instances.
Property	Type	Mult.	Unit	Description
ID [key]	URI	1		A global unique ID
Name	String	0..1		Human-readable name
Association End		Mult.	Description	
Manager.ID	<<abstract>>	1	A resource is managed by a manager	
Share.LocalID	<<abstract>>	*	A resource provides capacity in terms of shares	
Activity.ID		*	A resource runs zero or more activities	
Inherited Association End		Mult.	Description	
Extension.Key		*	The entity can be extended via key-value pairs	

5.10 Activity

Entity	Inherits from			Description
Activity	Entity			An activity is a unit of work managed by a service and submitted via an endpoint; when accepted by the endpoint, than it can be mapped to a share and can be executed by a local manager via one or more resources; an activity can have relationships to other activities being managed by different services, therefore it shares a common context.
Property	Type	Mult.	Unit	Description
ID [key]	URI	1		A global unique ID
Association End		Mult.	Description	
UserDomain.ID		0..1	An activity is managed by a user domain	
Endpoint.ID		0..1	An activity is submitted to an endpoint	
Share.LocalID	<<abstract>>	0..1	An activity is mapped into a share	
Resource.ID	<<abstract>>	0..1	An activity is executed in a resource	
Activity.ID		*	An activity is related to zero or more activities	
Activity.ID		*	An activity is related to zero or more activities	
Inherited Association End		Mult.	Description	
Extension.Key		*	The entity can be extended via key-value pairs	

Grid jobs (named Computing Activities in GLUE) are example of activities for a Computing Service. An interesting type of relationship for jobs derives from its propagation through several services. For instance, a broker service submits a Grid job to a selected execution service, upon completion the execution service submits a logging record to an accounting service. Each of these services will have associated an instance of a Grid job related to the lifecycle of the job within the service. All instances refer to the same conceptual job submitted by the user.

5.11 Policy

Entity	Inherits from			Description
Policy <<abstract>>	Entity			Statements, rules or assertions that specify the correct or expected behavior of an entity
Property	**Type**	**Mult.**	**Unit**	**Description**
LocalID	LocalID_t	1		An opaque identifier local to the Service to which the associated entity belongs to
Scheme	PolicyScheme_t	1		Scheme adopted to define the policy rules
Rule	String	1..*		A policy rule (for the basic policy scheme, syntax is provide in the Appendix)
Association End		**Mult.**	**Description**	
UserDomain.ID		1..*	A policy is related to a user domain	
Inherited Association End		**Mult.**	**Description**	
Extension.Key		*	The entity can be extended via key-value pairs	

This is an abstract entity not meant to be instantiated.

In this document, we provide the definition for a “basic” scheme (see Appendix 17.3). Such a scheme is designed to be simple and is inspired by real world scenarios in current production Grid systems. The Rule property implicitly contains the reference to the User Domains, therefore, in the concrete data model mapping, we RECOMMEND to not representing the association between User Domain and Access Policy or Mapping Policy explicitly since it is already captured by the Rule.

For a given entity to which policies are associated (i.e,, Endpoint and AccessPolicy, Share and MappingPolicy), several instances of the Policy class can be defined. This is allowed in order to enable to advertise policies using different schemes. We RECOMMEND that only one instance per policy scheme is associated to the same entity instance.

For a given entity instance, if it is associated to several policy instances with different policy schemes, then these policy instances SHOULD be expected to be consumed independently.

5.11.1 AccessPolicy

Entity	Inherits from			Description
AccessPolicy	Policy			Statements, rules or assertions that provide coarse-granularity information about the access by actors to an endpoint
Inherited Property	**Type**	**Mult**	**Unit**	**Description**
LocalID	*LocalID_t*	*1*		*An opaque identifier local to the Service to which the associated entity belongs to*
Scheme	*PolicyScheme_t*	*1*		*Scheme adopted to define the policy rules*
Rule	*PolicyRule_t*	*1..**		*A policy rule (for the basic policy scheme, syntax is provide in the Appendix)*
Property	**Type**	**Mult.**	**Unit**	**Description**
No extra properties are defined in the specialized entity				
Association End		**Mult.**	**Description**	
Endpoint.ID		1	An access policy is related to an endpoint	
Inherited Association End		**Mult.**	**Description**	
Extension.Key		*	The entity can be extended via key-value pairs	
UserDomain.ID		1..*	An access policy is related to a user domain	

This entity can be used to express which UserDomains can access a certain service endpoint. The granularity of these policies SHOULD be coarse-grained and suitable for pre-selection of services. The actual decision on the service side is performed by an authorization component that can contain a finer-grained set of policy rules that in some case can contradict the published coarse-grained policy rules. Examples of actors involved in this entity are userDomains representing VOs or groups.

5.11.2 MappingPolicy

Entity	Inherits from			Description
MappingPolicy	Policy			Statements, rules or assertions that provide coarse-granularity information about the mapping of user domain requests to a share
Inherited Property	Type	Mult	Unit	Description
LocalID	*LocalID_t*	*1*		*An opaque identifier local to the Service to which the associated entity belongs to*
Scheme	*PolicyScheme_t*	*1*		*Scheme adopted to define the policy rules*
Rule	*PolicyRule_t*	_1..*_		*A policy rule (for the basic policy scheme, syntax is provide in the Appendix)*
Property	Type	Mult.	Unit	Description
No extra properties are defined in the specialized entity				
Association End		Mult.	Description	
Share.LocalID	<<abstract>>	1	A mapping policy is related to a share	
Inherited Association End		Mult.	Description	
Extension.Key		*	The entity can be extended via key-value pairs	
UserDomain.ID		1..*	An access policy is related to a user domain	

This entity can be used to express which UserDomains can consume a certain share of resources. The granularity of these policies SHOULD be coarse-grained and suitable for pre-selection of services. The actual decision on the service side is performed by an authorization component that can contain a finer-grained set of policy rules that in some case can contradict the published coarse-grained policy rules.

6. Conceptual Model of the Computing Service

The conceptual model of the Computing Service is based upon the main entities and uses specializations of Service, Endpoint, Share, Manager, Resource, and Activity entities. Further computing related concepts such as Application Environment, Application Handle and Benchmark are introduced.

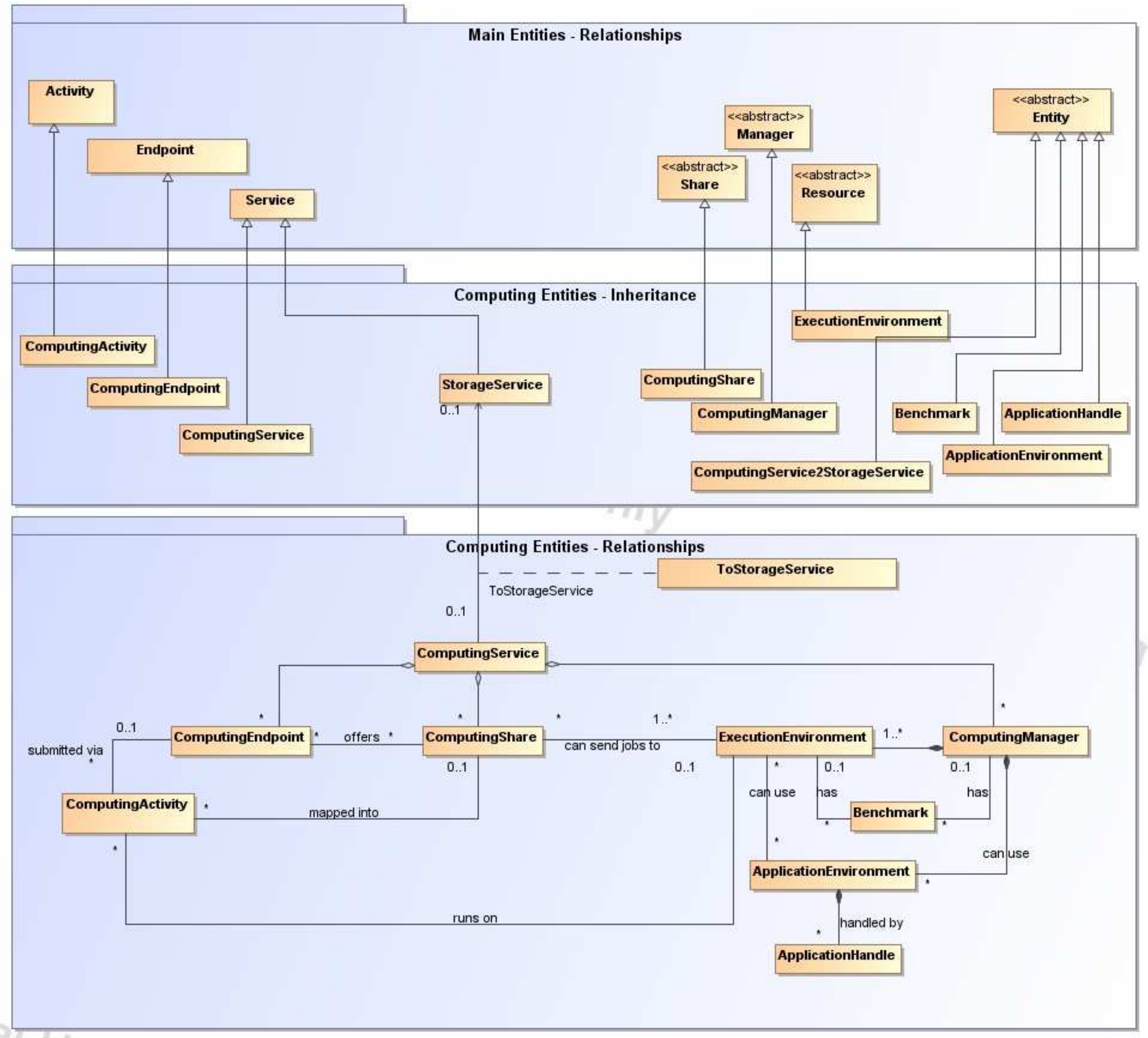

Figure 2 Entities and relationships for the Computing Service conceptual model

In this section, we extensively use the concept of physical CPU, logical CPU and slot:

- a physical CPU is defined by the socket, that means there is one physical CPU per socket; (e.g., a multi-core CPU counts as one physical CPU)
- a logical CPU corresponds to a CPU as visible by the operating system running either on a real or virtual machine
- a slot is a portion of executable time in a logical CPU offered by an execution environment instance which can be consumed by a job
    - usually, there is one slot per logical CPU, nevertheless a logical CPU can be shared across different slots
    - jobs can consume several slots at the same time (e.g., MPI jobs); a multi-slot job is counted as one job

6.1 ComputingService

Entity	Inherits from			Description
ComputingService	Service			An abstracted, logical view of actual software components that participate in the creation of a computational capacity in a Grid environment. A computing service exposes zero or more computing endpoints having well-defined interfaces, zero or more computing shares and zero or more computing managers and the related execution environments. The computing service is autonomous and denotes a weak aggregation among computing endpoints, the underlying computing managers and related execution environments, and the defined computing shares. The computing service enables to identify the whole set of entities providing the computing functionality with a persistent name.
Inherited Property	Type	Mult	Unit	Description
ID [key]	*URI*	*1*		*A global unique ID*
Name	*String*	*0..1*		*Human-readable name*
Capability	*Capability_t*	*1..**		*The provided capability according to the OGSA architecture (it is given by the sum of all the capabilities provided by the related endpoints)*
Type	*ServiceType_t*	*1*		*The type of service according to a middleware classification*
QualityLevel	*QualityLevel_t*	*1*		*Maturity of the service in terms of quality of the software components*
StatusPage	*URI*	***		*Web page providing additional information like monitoring aspects*
Complexity	*String*	*0..1*		*Human-readable summary description of the complexity in terms of the number of endpoint types, shares and resources. The syntax should be: endpointType=X, share=Y, resource=Z.*
OtherInfo	*String*	***		*Placeholder to publish info that does not fit in any other attribute. Free-form string, comma-separated tags, (name, value) pair are all examples of valid syntax*
Property	Type	Mult	Unit	Description
TotalJobs	UInt32	0..1	job	Number of total jobs (sum of RunningJobs, WaitingJobs, StagingJobs, SuspendedJobs and PreLRMSWaitingJobs); this numer does not consider the local jobs
RunningJobs	UInt32	0..1	job	Number of running Grid jobs
WaitingJobs	UInt32	0..1	job	Number of Grid jobs waiting in the underlying computing managers (i.e., Local Resource Manager System or LRMS's)
StagingJobs	UInt32	0..1	job	Number of jobs that are staging files in/out
SuspendedJobs	UInt32	0..1	job	Number of jobs which started their execution, but are suspended (e.g., for preemption)
PreLRMSWaitingJobs	UInt32	0..1	job	Number of jobs that are in the Grid layer waiting to be passed to the underlying computing manager (i.e., LRMS)
Association End		Mult.	Description	
ComputingEndpoint.ID [redefines Endpoint.ID]		*	A computing service exposes zero or more computing endpoints	
ComputingShare.LocalID [redefines Share.LocalID]		*	A computing service offers zero or more computing shares	
ComputingManager.ID [redefines Manager.ID]		*	A computing service offers zero or more computing managers	
StorageService.ID				
Inherited Association End		Mult.	Description	
Extension.Key		*	The entity can be extended via key-value pairs	
Contact.ID		*	A computing service has zero or more contacts	
Location.ID		0..1	A computing service is primary located at a location	
Service.ID		*	A computing service is related to zero or more services	

The simplest computing service is formed by a computing endpoint exposing an interface for job submission and control.

In case of a single computing manager whose execution environments are exposed by multiple computing endpoints, both computing manager, execution environments and computing endpoints MUST be considered as part of the same computing service. In case of a single computing endpoint exposing execution environments managed by different computing managers, then the computing endpoint, the execution environments and the related computing managers MUST be considered as part of the computing service.

The computing service always aggregates computing endpoints, computing shares, computing managers and execution environments forming a connected set. In other words, Endpoint A exposing Execution Environment A of Manager A via Share A and Endpoint B exposing Execution Environment B of Manager B via Share B form two different computing services. On the other side, Endpoint A exposing Execution Environment A of Manager A via Share A and Endpoint B exposing Execution Environment A of Manager A via Share B form one Computing Service.

6.2 ComputingEndpoint

Entity	Inherits from			Description
ComputingEndpoint	Endpoint			Endpoint for creating, monitoring, and controlling computational activities called jobs; it can be used to expose also complementary capabilities (e.g., reservation, proxy manipulation)
Inherited Property	**Type**	**Mult**	**Unit**	**Description**
ID [key]	*URI*	*1*		*A global unique ID*
Name	*String*	*0..1*		*Human-readable name*
URL	*URI*	*1*		*Network location of the endpoint to contact the related service*
Capability	*Capability_t*	*1..**		*The provided capability according to the OGSA architecture*
Technology	*EndpointTechnology_t*	*0..1*		*Technology used to implement the endpoint*
Interface	*URI*	*1*		*Identification of a type and version of the interface*
InterfaceExtension	*URI*	***		*Identification of an extension to the interface*
WSDL	*URI*	***		*URL of the WSDL document describing the offered interface (applies to Web Services endpoint)*
SupportedProfile	*URI*	***		*URI identifying a supported profile*
Semantics	*URI*	***		*URI of a document providing a human-readable description of the semantics of the endpoint functionalities*
Implementor	*String*	*0..1*		*Main organization implementing this software component*
ImplementationName	*String*	*0..1*		*Name of the implementation*
ImplementationVersion	*String*	*0..1*		*Version of the implementation (e.g., major version.minor version.patch version)*
QualityLevel	*QualityLevel_t*	*1*		*Maturity of the endpoint in terms of quality of the software components*
HealthState	*EndpointHealthState_t*	*1*		*A state representing the health of the endpoint in terms of its capability of properly delivering the functionalities*
HealthStateInfo	*String*	*0..1*		*Textual explanation of the state endpoint*
ServingState	*ServingState_t*	*1*		*A state specifying if the endpoint is accepting new requests and if it is serving the already accepted requests*
StartTime	*DateTime_t*	*0..1*		*The timestamp for the start time of the endpoint*
IssuerCA	*DN_t*	*0..1*		*Distinguished name of Certification Authority issuing the certificate for the endpoint*
TrustedCA	*DN_t*	***		*Distinguished name of the trusted Certification Authority*
DowntimeAnnounce	*DateTime_t*	*0..1*		*The timestamp for the announcement of the*

				next scheduled downtime
DowntimeStart	*DateTime_t*	*0..1*		*The starting timestamp of the next scheduled downtime*
DowntimeEnd	*DateTime_t*	*0..1*		*The ending timestamp of the next scheduled downtime*
DowntimeInfo	*String*	*0..1*		*Description of the next scheduled downtime*
Property	Type	Mult.	Unit	Description
Staging	Staging_t	0..1		Supported staging functionalities
JobDescription	JobDescription_t	*		Supported type of job description language
Association End		Mult.	Description	
ComputingService.ID [redefines Service.ID]		1	A computing endpoint is part of a Computing Service	
ComputingShare.LocalID [redefines Share.LocalID]		*	A computing endpoint can pass activities to zero or more computing shares	
ComputingActivity.ID [redefines Activity.ID]		*	An endpoint has accepted and is managing zero or more Activities	
Inherited Association End		Mult.	Description	
Extension.Key		*	The entity can be extended via key-value pairs	
AccessPolicy.ID		*	A computing endpoint has assocated zero or more AccessPolicies	

6.3 ComputingShare

A computing share is a high-level concept introduced to model the utilization target for a set of execution environments defined by a set of configuration parameters and characterized by status information. In clusters managed by a batch system, the simplest way to set up a computing share is to configure a batch queue, nevertheless, the same computing share can be implemented using different batch system configuration strategies.

In complex batch systems, a batch queue can be configured with different set of policies for different set of users. This implies that each set of users obtains a different utilization target. Such a scenario can be represented by different computing shares.

In general, given a number of shares to be set up, it is possible to adopt different configuration strategies in the underlying system. Regardless the selected approach, the external behavior does not change. The main goal of the computing share concept is to abstract from such implementation choices and to represent the externally observable behavior.

The computing share supports also heterogeneity by being able to have associations to different execution environments.

Entity	Inherits from			Description
ComputingShare	Share			A utilization target for a set of execution environments defined by a set of configuration parameters and characterized by status information
Inherited Property	Type	Mult	Unit	Description
LocalID [key]	*LocalID_t*	*1*		*An opaque identifier local to the associated Service*
Name	*String*	*0..1*		*Human-readable name*
Description	*String*	*0..1*		*Description of this share*
Property	Type	Mult.	Unit	Description
MappingQueue	String	0..1		Name of a queue available in the underlying computing manager (i.e., LRMS) where jobs of this share are submitted (different shares can be mapped into the same queue; it is not foreseen that a single share can be mapped into many different queues)
MaxWallTime	UInt64	0..1	s	The maximum obtainable wall clock time per slot that can be granted to the job upon user request (unnormalized value)
MaxTotalWallTime	UInt64	0..1	s	The maximum obtainable total wall clock time that can be granted to the job upon user

				request; this property is a limit for the sum of the wall clock time used in all the slots occupied by a multi-slot job (unnormalized value)
MinWallTime	UInt64	0..1	s	The minimum wall clock time per slot for a job (unnormalized value); if a job requests a lower time, than it can be rejected; if a job requests at least this value, but runs for a shorter time, than it might be accounted for this value
DefaultWallTime	UInt64	0..1	s	The default wall clock time per slot allowed to a job by the computing manager (i.e., LRMS) if no limit is requested in the job submission description. Once this time is expired the job will most likely be killed or removed from the queue (unnormalized value)
MaxCPUTime	UInt64	0..1	s	The maximum obtainable CPU time that can be granted to the job upon user request per slot (unnormalized value)
MaxTotalCPUTime	UInt64	0..1	s	The maximum obtainable CPU time that can be granted to the job upon user request across all assigned slots; this property is a limit for the sum of the CPU time used in all the slots occupied by a multi-slot job (unnormalized value)
MinCPUTime	UInt64	0..1	s	The minimum CPU time per slot for a job (unnormalized value); if a job requests a lower time, than it can be rejected; if a job requests at least this value, but uses the CPU for a shorter time, than it might be accounted for this value
DefaultCPUTime	UInt64	0..1	s	The default CPU time per slot allowed to each job by the computing manager (i.e., LRMS) if no limit is requested in the job submission description (unnormalized value)
MaxTotalJobs	UInt32	0..1	job	The maximum allowed number of jobs in this share
MaxRunningJobs	UInt32	0..1	job	The maximum allowed number of jobs in running state in this share
MaxWaitingJobs	UInt32	0..1	job	The maximum allowed number of jobs in waiting state in this share
MaxPreLRMSWaitingJobs	UInt32	0..1	job	The maximum allowed number of jobs that are in the Grid layer waiting to be passed to the underlying computing manager (i.e., LRMS) for this share
MaxUserRunningJobs	UInt32	0..1	job	The maximum allowed number of jobs in running state per Grid user in this share
MaxSlotsPerJob	UInt32	0..1	slot	The maximum number of slots which could be allocated to a single job (defined to be 1 for a computing service accepting only single-slot jobs)
MaxStageInStreams	UInt32	0..1	stream	The maximum number of streams to stage files in
MaxStageOutStreams	UInt32	0..1	stream	The maximum number of streams to stage files out
SchedulingPolicy	SchedulingPolicy_t	0..1		Implied scheduling policy of the share
MaxMemory	UInt64	0..1	MB	The maximum RAM that a job can use
MaxDiskSpace	UInt64	0..1	GB	The maximum disk space that a job can use excluding shared area such as cache
DefaultStorageService	URI	0..1		ID of the default Storage Service to be used to store files by jobs in case no destination Storage Service is explicitly stated
Preemption	Boolean	0..1		True if the computing manager (i.e., LRMS) enables preemption of jobs; a preempted job is supposed to be automatically resumed
ServingState	ServingState_t	1		A state specifying if the share is open to place new requests and if it is open to offer the already present requests for execution
TotalJobs	UInt32	0..1	job	Number of total jobs in any state (sum of

				RunningJobs, WaitingJobs, StagingJobs, SuspendedJobs and PreLRMSWaitingJobs); this numer includes the local jobs
RunningJobs	UInt32	0..1	job	Number of running jobs submitted via any type of interface (local and Grid)
LocalRunningJobs	UInt32	0..1	job	Number of running jobs submitted via a local interface
WaitingJobs	UInt32	0..1	job	Number of jobs waiting in the underlying computing managers (i.e., LRMS's) submitted via any type of interface (local and Grid)
LocalWaitingJobs	UInt32	0..1	job	Number of jobs waiting in the underlying computing managers (i.e., LRMS's) submitted via a local interface
StagingJobs	UInt32	0..1	job	Number of jobs that are staging files in/out
SuspendedJobs	UInt32	0..1	job	Number of jobs which started their execution, but are suspended (e.g., for preemption)
PreLRMSWaitingJobs	UInt32	0..1	job	Number of jobs that are in the Grid layer waiting to be passed to the underlying computing manager (i.e., LRMS)
EstimatedAverageWaitingTime	UInt64	0..1	s	Estimated time to last for a new job from the acceptance to the start of its execution
EstimatedWorstWaitingTime	UInt64	0..1	s	Estimated worst waiting time assuming that all jobs run for the maximum wall time
FreeSlots	UInt32	0..1	slot	Number of free slots
FreeSlotsWithDuration	String	0..1	slot:s	Number of free slots with their time limits. Syntax: ns[:t] [ns:t]* where the pair ns:t means that there are *ns* free slots for the duration of *t* (expressed in seconds); the time limit information is optional
UsedSlots	UInt32	0..1	slot	Number of slots used by running jobs
RequestedSlots	UInt32	0..1	slot	Number of slots which are needed to execute all waiting and staging jobs
ReservationPolicy	ReservationPolicy_t	0..1		Type of reservation policy
Tag	String	*		UserDomain-defined tag (the values SHOULD use namespace to avoid collision)
Association End		**Mult.**	**Description**	
ComputingEndpoint.ID [redefines Endpoint.ID]		*	A computing share can be consumed via one or more computing endpoints	
ExecutionEnvironment.ID [redefines Resource.ID]		1..*	A computing share is defined on one or more computing resources	
ComputingService.ID [redefines Service.ID]		1	A computing share participates in a computing service	
ComputingActivity.ID [redefines Activity.ID]		*	A computing share is being consumed by zero or more computing activities	
Inherited Association End		**Mult.**	**Description**	
Extension.Key		*	The entity can be extended via key-value pairs	
MappingPolicy.ID		*	A share has zero or more mapping policies	

As regards CPU Time and Wall Time related properties, there is the need for a way to normalize them depending on the computing capacity of the execution environment. The approach proposed in GLUE is to add two attributes in the Execution Environment (see Section 6.6) which refer to the scaling factor to be used to compute the CPU/Wall time that a job will get if it will be assigned to such an execution environment via a certain share. It is important that a job will get always at least the advertised CPU/Wall time. This means that the reference Execution Environment for the normalization should be always the fastest among those available in the whole Computing Service. For this Execution Environment, the scaling factor MUST be equal to 1. The CPU/Wall time values published by a share refer to the time that the job will get when mapped to this Execution Environment. For the other Execution Environments, the time should be normalized according to the defined scaling factors.

6.4 ComputingManager

Entity	Inherits from			Description
ComputingManager	Manager			A software component locally managing one or more execution environments. It can describe also aggregated information about the managed resources. The computing manager is also known as Local Resource Management System (LRMS).
Inherited Property	**Type**	**Mult**	**Unit**	**Description**
ID [key]	*URI*	*1*		*A global unique ID*
Name	*String*	*0..1*		*Human-readable name*
Property	**Type**	**Mult.**	**Unit**	**Description**
Type	ComputingManagerType_t	1		Type of the computing manager (i.e., LRMS)
Version	String	0..1		Version of the computing manager (i.e., LRMS)
Reservation	Boolean	0..1		True if the computing manager (i.e, LRMS) supports advance reservation
BulkSubmission	Boolean	0..1		True if the computing manager (i.e, LRMS) supports the bulk submission
TotalPhysicalCPUs	UInt32	0..1	Ph.CPU	Number of managed physical CPUs accessible via any of the available endpoints (there is one physical CPU per socket)
TotalLogicalCPUs	UInt32	0..1	Log.CPU	Number of managed logical CPUs accessible via any of the available endpoints (a logical CPU corresponds to a CPU visible to the operating system)
TotalSlots	UInt32	0..1	slot	Number of managed slots
SlotsUsedByLocalJobs	UInt32	0..1	slot	Number of slots used by jobs submitted via local interface
SlotsUsedByGridJobs	UInt32	0..1	slot	Number of slots used by jobs submitted via a Grid interface
Homogeneity	Boolean	0..1		True if the computing manager has only one type of execution environment
NetworkInfo	NetworkInfo_t	0..1		Type of internal network available among all the managed execution environment instances
LogicalCPUDistribution	String	0..1		Classification of the managed execution environment instances aggregated by number of logical CPUs.Syntax: X*1*:Y*1*, …, X*n*:Y*n* where I is the i-*th* group of execution environments with the same number of logical CPUs, X*i* is the number of logical CPUs in each execution environment instance and Y*i* is the number of execution environment instances.
WorkingAreaShared	Boolean	0..1		A working area is an allocated storage extent that holds the home directories of the Grid jobs; this property is true if the working area is shared across different execution environment instances (i.e., cluster nodes)
WorkingAreaTotal	UInt64	0..1	GB	Total size of working area available to all the Grid jobs either as a shared area across all the execution environments (WorkingAreaShared is true) or local to a certain execution environment (WorkingAreaShared is false); if the computing manager supports individual quota per job/user, this is not advertised; in case of non-shared working area with different local space allocation, the advertised total size is the minimum available across all the execution

				environment instances
WorkingAreaFree	UInt64	0..1	GB	Free size of working area available to all the Grid jobs either as a shared area across all the execution environments (WorkingAreaShared is true) or local to a certain execution environment (WorkingAreaShared is false); if the computing manager supports individual quota per job/user, this is not advertised; in case of non-shared working area, this attribute represents the minimum guaranteed free working area available in any execution environment instance at the beginning of the job execution
WorkingAreaLifeTime	UInt64	0..1	s	Guaranteed lifetime of the Grid job files present in the working area; the lifetime is related to the end time of the job; after the expiration of the lifetime, the files are not guaranteed to exist
CacheTotal	UInt64	0..1	GB	Total size of a shared temporary storage area where frequently accessed data can be stored for rapid access by consequent Grid jobs
CacheFree	UInt64	0..1	GB	Free size of a shared temporary storage area where frequently accessed data can be stored for rapid access by consequent Grid jobs; in the computation of the free size, files which are not claimed by any job can be considered as deleted
TmpDir	String	0..1		The absolute path of a temporary directory local to an execution environment instance (i.e., worker node). This directory must be available to programs using the normal file access primitives (open/read/write/close)
ScratchDir	String	0..1		The absolute path for a shared directory available for application data. Typically a POSIX accessible transient disk space shared between the execution environment instances. It may be used by MPI applications or to store intermediate files that need further processing by local jobs or as staging area, specially if the execution environment instances have no internet connectivity
ApplicationDir	String	0..1		The path of the directory available for application installation. Typically a PO-SIX accessible disk space with transient to permanent allocation to the users
OtherInfo	String	*		Placeholder to publish info that does not fit in any other attribute. Free-form string, comma-separated tags, (name, value) pair *are all examples of valid syntax*

Association End	Mult.	Description
ComputingService.ID [redefines Service.ID]	1	A computing manager participates in a computing service
ExecutionEnvironment.ID [redefines Resource.ID]	1..*	A computing manager manages one or more execution environments
ApplicationEnvironment.LocalID	*	A computing manager can use zero or more application environments
Benchmark.LocalID	*	A computing manager has zero or more associated benchmarks

Inherited Association End	Mult.	Description
Extension.Key	*	The entity can be extended via key-value pairs

The Operating System can be the simplest case of computing manager. A typical example of computing manager is a batch system (i.e., LRMS).

6.5 Benchmark

Entity	Inherits from			Description
Benchmark	Entity			Benchmark information about an entity providing computing capacity
Property	Type	Mult.	Unit	Description
LocalID	LocalID_t	1		An opaque identifier local to the Computing Service
Type	Benchmark_t	1		Type of benchmark
Value	Real32	1		Value
Association End		Mult.	Description	
ExecutionEnvironment.ID		0..1	A benchmark can be related to an execution environment	
ComputingManager. ID		0..1	A benchmark can be related to a computing resource	
Inherited Association End		Mult.	Description	
Extension.Key		*	The entity can be extended via key-value pairs	

6.6 ExecutionEnvironment

Entity	Inherits from			Description
ExecutionEnvironment	Resource			A type of environment available to and requestable by a Grid job when submitted to a ComputingService via a Computing Endpoint; the type of envonrment is described in terms of hardware, operating system and network characteristics; the information about the total/available/used instances of this type of execution environment are also included
Inherited Property	Type	Mult.	Unit	Description
ID [key]	*URI*	*1*		*A global unique ID*
Name	*String*	*0..1*		*Human-readable name*
Property	Type	Mult.	Unit	Description
Platform	Platform_t	1		The architecture platform of this execution environment
VirtualMachine	Boolean	0..1		True if the execution environment is based on a virtual machine (in this case, the values of the other attributes are related to the virtualized environment and not to the hosting environment)
TotalInstances	UInt32	0..1		Number of execution environment instances
UsedInstances	UInt32	0..1		Number of used execution environment instances; an instance is used when, according to the policies of the Computing Manager (i.e., LRMS), it cannot accept new jobs because it already runs the maximum number of allowed jobs
UnavailableInstances	UInt32	0..1		Number of unavailable execution environment instances because of failures or maintenance
PhysicalCPUs	UInt32	0..1		Number of physical CPUs in an execution environment instance
LogicalCPUs	UInt32	0..1		Number of logical CPUs in an execution environment instance
CPUMultiplicity	CPUMultiplicity_t	0..1		Information about the multiplicity of both physical CPUs and cores available in an execution environment instance
CPUVendor	String	0..1		Name of the physical CPU vendor
CPUModel	String	0..1		Physical CPU model as defined by the vendor
CPUVersion	String	0..1		Physical CPU version as defined by the vendor
CPUClockSpeed	UInt32	0..1	MHz	Nominal clock speed of the physical CPU
CPUTimeScalingFactor	Real32	0..1		Factor used by the Computing Manager (i.e., LRMS) to scale the CPU time (CPU Time divided by CPUTimeScalingFactor); for the reference execution environment, this attribute is equal to 1
WallTimeScalingFactor	Real32	0..1		Factor used by the Computing Manager (i.e., LRMS) to scale the Wall time (Wall Time divided by WallTimeScalingFactor)
MainMemorySize	UInt64	1	MB	Amount of RAM (if many jobs run in the same execution environment, they compete for the total

				RAM)
VirtualMemorySize	UInt64	0..1	MB	The amount of Virtual Memory (RAM+Swap)
OSFamily	OSFamily_t	1		Family of the operating system
OSName	OSName_t	0..1		Name of the operating system
OSVersion	String	0..1		Version of the operating system
ConnectivityIn	Boolean	1		Permission for direct inbound connectivity, even if limited
ConnectivityOut	Boolean	1		Permission for direct outbound connectivity, even if limited
NetworkInfo	NetworkInfo_t	*		Type of internal network available among the execution environment instances
Association End		**Mult.**	**Description**	
ComputingManager.ID [redefines Manager.ID]		1	An execution environment is managed by a computing manager	
ComputingShare.LocalID [redefines Share.LocalID]		*	An execution environment provides capacity in terms of computing shares	
ComputingActivity.ID [redefines Activity.ID]		*	An execution environment runs zero or more computing activities	
ApplicationEnvironment.LocalID		*	An execution environment offers zero or more application environments	
Benchmark.LocalID		*	An execution environment has zero or more associated benchmarks	
Inherited Association End		**Mult.**	**Description**	
Extension.Key		*	The entity can be extended via key-value pairs	

Each execution environment instance is under the responsibility of a Computing Manager (i.e., LRMS). An execution environment can be realized in several ways. Examples are a computing node or a virtual machine image that can be requested by a job (different virtual machine images can coexist on the same node). The description about individual software packages is considered by the ApplicationEnvironment class.

6.7 ApplicationEnvironment

Entity	Inherits from			Description
ApplicationEnvironment	Entity			Description of the application software or environment characteristic available within one or more execution environments
Property	**Type**	**Mult.**	**Unit**	**Description**
LocalID	LocalID_t	1		An opaque identifier local to the Computing Service
Name	String	1		Name of the application environment
Version	String	0..1		Version of the application environment
Repository	URI	0..1		URL of a service which offers a repository and/or a name service for this application environment
State	AppEnvState_t	0..1		State about the installation
RemovalDate	DateTime_t	0..1		Date and time after which the application can be removed
License	License_t	0..1		The type of license
Description	String	0..1		The description of this application environment
BestBenchmark	Benchmark_t	*		Type of benchmark which best identify the sensitivity of this application to the performance aspect
ParallelSupport	ParallelSupport_t	0..1		The type of supported parallel execution framework
MaxSlots	UInt32	0..1	slot	Maximum number of slots that can be used to run jobs using the application environment at the same time
MaxJobs	UInt32	0..1	job	Maximum number of jobs that can use the application environment at the same time
MaxUserSeats	UInt32	0..1	user seat	Maximum number of user seats that can use the application environment at the same time
FreeSlots	UInt32	0..1	slot	Available number slots that can be used to run jobs using the application environment at the same time
FreeJobs	UInt32	0..1	slot	Number of new jobs that could start their execution and use the application environment at the same time
FreeUserSeats	UInt32	0..1	user seat	Free seats for additional users that can use the

				application environment at the same time
Association End		Mult.	Description	
ExecutionEnvironment.ID		*	An application environment can be used in zero or more execution environments	
ComputingManager.ID		1	An application environment is part of a computing manager	
ApplicationHandle.LocalID		*	An application environment can be handled via zero or more application handles	
Inherited Association End		Mult.	Description	
Extension.Key		*	The entity can be extended via key-value pairs	

There is no recommendation for the Name property of the Application Environment. In some deployment scenario, the definition of namespace-based Names or guidelines for unique application names can be defined; application repository services relying on the unique application names can be provided. This aspect is considered out of scope for GLUE.

The Application Environment is suggested to be used also for describing application software or special environment setup in terms of a simple tag. In this case, the Name property should be used.

6.8 ApplicationHandle

Entity	Inherits from			Description
ApplicationHandle	Entity			Technique for bootstrapping and/or accessing the application
Property	Type	Mult.	Unit	Description
LocalID	LocalID_t	1		An opaque identifier local to the Computing Service
Type	ApplicationHandle_t	1		Type of handle for an application environment
Value	String	1		Actionable value to trigger the handle method
Association End		Mult.	Description	
ApplicationEnvironment.LocalID		1	An application handle can be used for one application environment	
Inherited Association End		Mult.	Description	
Extension.Key		*	The entity can be extended via key-value pairs	

6.9 ComputingActivity

Entity	Inherits from			Description
ComputingActivity	Activity			An activity managed by an OGSA execution capability service (the computing activity is traditionally called job)
Inherited Property	*Type*	*Mult*	*Unit*	*Description*
ID [key]	*URI*	*1*		*A global unique ID*
Property	Type	Mult.	Unit	Description
Name	String	0..1		Human-readable name as specified by the user in the job description document
Type	ComputingActivityType_t	0..1		Type of computing activity
IDFromEndpoint	URI	0..1		The job ID as assigned by the computing endpoint
LocalIDFromManager	String	0..1		The local ID of the job as assigned by the computing manager (i.e., LRMS)
JobDescription	JobDescription_t	0..1		Job description language used to specify the job request
State	ComputingActivityState_t	1		The state of the job according to the Grid state model for jobs
RestartState	ComputingActivityState_t	0..1		The state from which a failed job can restart upon a client request
ExitCode	Int32	0..1		The exit code as returned by the executable of the job
ComputingManagerExitCode	String	0..1		The exit code provided by the computing manager (i.e., LRMS)
Error	String	*		Error messages as provided by the

				software components involved in the management of the job
WaitingPosition	UInt32	0..1		For a waiting job in the computing manager (i.e., LRMS), the position of the job in the queue
UserDomain	String	0..1		User domain selected by the job owner in the job submission request (an owner can belong to several user domains, it should decide which one to choose when submitting a job)
Owner	String	1		The Grid identity of the job's owner; in case of anonymity is required, the value CONFIDENTIAL should be advertised
LocalOwner	String	0..1		The local user name to which the job's owner is mapped into
RequestedTotalWallTime	UInt64	0..1	s	The total wall clock time requested by the job; for multi-slot jobs, it represents the sum of wall clock time needed in each required slot
RequestedTotalCPUTime	UInt64	0..1	s	The total CPU time requested by the job for multi-slot jobs, it represents the sum of CPU time needed in each required slot
RequestedSlots	UInt32	0..1	slot	The number of requested slots
RequestedApplicationEnvironment	String	*		Serialization of the Name and Version of the requested Application Environment to match the Name and Version properties of the Application Environment (the serialization of the Name and Version is delegated to the implementers)
StdIn	String	0..1		The name of the file which is used as the standard input of the job
StdOut	String	0..1		The name of the file which contains the standard output of the job
StdErr	String	0..1		The name of the file which contains the standard error of the job
LogDir	String	0..1		The name of the directory which contains the logs related to the job and generated by the Grid layer (usually the directory is private to the job)
ExecutionNode	String	*		Hostname associated to the execution environment instance (i.e., worker node) running the job; multi-node jobs are described by several instances of this attribute
Queue	String	0..1		The name of the Computing Manager (i.e, LRMS) queue to which this job was queued
UsedTotalWallTime	UInt64	0..1	s	The totally consumed wall clock time by the job (in case of multi-slot jobs, this value refers to the sum of the wall clock time consumed in each slot)
UsedTotalCPUTime	UInt64	0..1	s	The totally consumed CPU time by the job (in case of multi-slot jobs, this value refers to the sum of the consumed CPU time in each slot)
UsedMainMemory	UInt64	0..1	MB	The RAM used by the job
SubmissionTime	DateTime_t	0..1		Time when the job was submitted to a computing endpoint
ComputingManagerSubmissionTime	DateTime_t	0..1		Time when the job was submitted to the Computing Manager (i.e., LRMS) by the Grid layer
StartTime	DateTime_t	0..1		Time when the job entered in the

				Computing Manager (i.e., LRMS) running state
ComputingManagerEndTime	DateTime_t	0..1		Time when the job entered its final Computing Manager (i.e., LRMS) state
EndTime	DateTime_t	0..1		Time when the job entered its final Grid state
WorkingAreaEraseTime	DateTime_t	0..1		A working area is an allocated storage extent that holds the home directories of the Grid jobs; the time when the dedicated working area of this job will be removed
ProxyExpirationTime	DateTime_t	0..1		The expiration time of the proxy related to the job, in case of proxy with attribute certificates having different expiration times, then this value represent the minimum expiration time among all the values
SubmissionHost	String	0..1		The name of the host from which the job was submitted
SubmissionClientName	String	0..1		The name of the software client which was used to submit the job
OtherMessages	String	*		Optional job messages provided by either the Grid Layer or the Computing Manager (i.e., LRMS)
Association End		**Mult.**	**Description**	
ComputingEndpoint.ID [redefines Endpoint.ID]		0..1	A computing activity is submitted to a computing endpoint	
ComputingShare.LocalID [redefines Share.LocalID]		0..1	A computing activity is mapped into a computing share	
ExecutionEnvironment.ID [redefines Resource.ID]		0..1	A computing activity is executed in an execution environment	
Inherited Association End		**Mult.**	**Description**	
Extension.Key		*	The entity can be extended via key-value pairs	
UserDomain.ID		0..1	An activity is managed by a user domain	
Activity.ID		*	An activity is related to zero or more activities	
Activity.ID		*	An activity is related to zero or more activities	

In this specification, the Computing Activity refers to simple jobs or element of collections or workflow. The description of the relationships between jobs part of a collection or workflow may be considered in future revisions of the specification.

As regards the State property and the related ComputingActivityState_t type, we notice that currently there is no commonly accepted state model. Each production Grid middleware defined and is using its own state model. As regards the standardization process, the OGSA-BES specification defines a simple state model. The middleware providers started to define their own extensions to the BES state model, nevertheless they differ and do not enable interoperability. Given the current scenario, we RECOMMEND to use namespace in state model values, so that every middleware provider can publish the computing activity state according to its definition. We expect that an extension to the core BES state model common to all the middleware providers and suitable for production scenarios can be defined by a profiling activity of the BES/ /JSDL/GLUE specifications.

6.10 ToStorageService

Entity	Inherits from			Description
ToStorageService	Entity			Description of a POSIX access via a file system technology enabling the computing service to access the associated storage service
Property	Type	Mult.	Unit	Description
LocalID	LocalID_t	1		An opaque identifier local to the Computing Service
LocalPath	String	1		The local path of the computing service enabling to access a remote path in the associated storage service (this is typically an NFS mount point)
RemotePath	String	1		The remote path in the storage service which is associated the local path in the computing service (this is typically an NFS exported directory)
Association End		Mult.	Description	
ComputingService.ID		1	Is associated to a computing service	
StorageService.ID		1	Is associated to a storage service	
Inherited Association End		Mult.	Description	
Extension.Key		*	The entity can be extended via key-value pairs	

7. Conceptual Model of the Storage Service

The conceptual model of the Storage Service is based upon the main entities and uses specializations of Service, Endpoint, Share, Manager, Resource, and Activity entities. Further storage related concepts such as Storage Service Capacity, Storage Share Capacity and Storage Access Protocol are introduced.

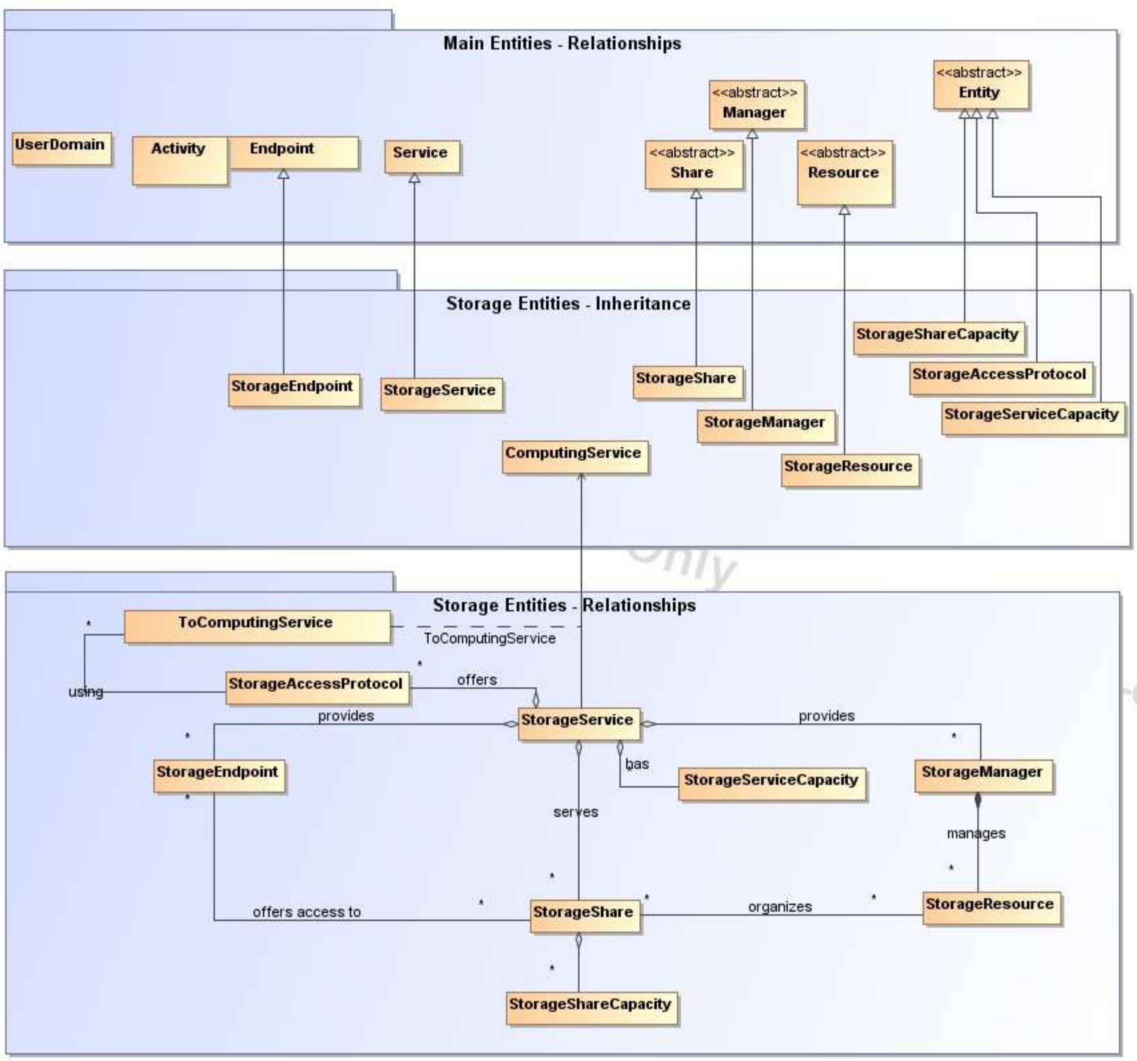

Figure 3 Entities and relationships for the Storage Service conceptual model

7.1 StorageService

Entity	Inherits from	Description
StorageService	Service	An abstracted, logical view of actual software components that participate in the creation of a storage capacity in a Grid environment. A storage service exposes zero or more endpoints having well-defined interfaces, zero or more storage shares and zero or more storage managers and the related storage resources. The storage service also offers zero or more storage access protocols and provides summary information about the global capacity by means of the storage service capacity The storage service is autonomous and denotes a weak aggregation among storage endpoints, storage shares, storage managers, storage access protocols and storage service capacity. The storage service enables to identify the whole set of entities providing the storage functionality with a persistent name.

Inherited Property	Type	Mult	Unit	Description
ID [key]	*URI*	*1*		*A global unique ID*
Name	*String*	*0..1*		*Human-readable name*
Capability	*Capability_t*	*1..**		*The provided capability according to the OGSA architecture (it is given by the sum of all the capabilities provided by the related endpoints)*
Type	*ServiceType_t*	*1*		*The type of service according to a middleware classification*
QualityLevel	*QualityLevel_t*	*1*		*Maturity of the service in terms of quality of the software components*
StatusPage	*URI*	***		*Web page providing additional information like monitoring aspects*
Complexity	*String*	*0..1*		*Human-readable summary description of the complexity in terms of the number of endpoint types, shares and resources. The syntax should be: endpointType=X, share=Y, resource=Z.*
OtherInfo	*String*	***		*Placeholder to publish info that does not fit in any other attribute. Free-form string, comma-separated tags, (name, value) pair are all examples of valid syntax*

Property	Type	Mult	Unit	Description
No extra properties are defined in the specialized entity				

Association End	Mult.	Description
StorageEndpoint.ID [redefines Endpoint.LocalID]	*	A storage service exposes zero or more storage endpoints
StorageShare.LocalID [redefines Share.LocalID]	*	A storage service serves zero or more storage shares
StorageManager.ID [redefines Manager.ID]	*	A storage service provides zero or more storage managers
StorageAccessProtocol.LocalID	*	A storage service offers zero or more storage access protocols
StorageServiceCapacity.LocalID	*	A storage service has zero or more storage service capacities
Inherited Association End	**Mult.**	**Description**
Extension.Key	*	The entity can be extended via key-value pairs
Contact.ID	*	A service has zero or more contacts
Location.ID	0..1	A service is primary located at a location
Service.ID	*	A service is related to zero or more services

The storage service can expose storage endpoints enabling to manage or access different types of storage capacity. The usage of storage capacity is typically constrained by policies, thus implying service differentiation. Each homogenously constrained storage capacity is described by the storage share concept.

The storage capacity used to create shares is locally managed by a storage manager and provided by storage resources.

7.2 StorageServiceCapacity

Entity	Inherits from			Description
StorageServiceCapacity	Entity			Description of the size and usage of an homogenous storage extent; the storage extent is aggregated at the storage service level by type
Property	Type	Mult.	Unit	Description
LocalID	LocalID_t	1		An opaque identifier local to the Storage Service
Type	StorageCapacity_t	1		Type of storage capacity
TotalSize	UInt64	0..1	GB	Size of dedicated storage extent
FreeSize	UInt64	0..1	GB	Size of free storage extent
UsedSize	UInt64	0..1	GB	Size of used storage extent
ReservedSize	UInt64	0..1	GB	Size of reserved storage extent
OtherInfo	String	*		Placeholder to publish info that does not fit in any other attribute. Free-form string, comma-separated tags, (name, value) pair *are all examples of valid syntax*
Association End		Mult.	Description	
StorageService.ID		1	A storage service capacity is related to one storage service	
Inherited Association End		Mult.	Description	
Extension.Key		*	The entity can be extended via key-value pairs	

7.3 StorageAccessProtocol

Entity	Inherits from			Description
StorageAccessProtocol	Entity			A type of protocol available to access the available storage capacities
Property	Type	Mult.	Unit	Description
LocalID	LocalID_t	1		An opaque identifier local to the Storage Service
Type	StorageAccessProtocol_t	1		The name of the protocol
Version	String	1		The version of the protocol
MaxStreams	UInt32	0..1	stream	The number of parallel streams this protocol supports
OtherInfo	String	*		Placeholder to publish info that does not fit in any other attribute. Free-form string, comma-separated tags, (name, value) pair are all examples of valid syntax
Association End		Mult.	Description	
StorageService.ID		1	A storage access protocol is related to one storage service	
ToComputingService		*	A storage access protocol can be used by zero or more computing services	
Inherited Association End		Mult.	Description	
Extension.Key		*	The entity can be extended via key-value pairs	

7.4 StorageEndpoint

Entity	Inherits from			Description
StorageEndpoint	Endpoint			Endpoint for managing storage shares or for accessing them; it can be used to expose also complementary capabilities part of the storage service
Inherited Property	Type	Mult	Unit	Description
ID [key]	*URI*	*1*		*A global unique ID*
Name	*String*	*0..1*		*Human-readable name*
URL	*URI*	*1*		*Network location of the endpoint to contact the related service*
Capability	*Capability_t*	*1..**		*The provided capability according to the OGSA architecture*
Technology	*EndpointTechnology_t*	*0..1*		*Technology used to implement the endpoint*
Interface	*URI*	*1*		*Identification of a type and version of the interface*
InterfaceExtension	*URI*	***		*Identification of an extension to the interface*

WSDL	*URI*	*		*URL of the WSDL document describing the offered interface (applies to Web Services endpoint)*
SupportedProfile	*URI*	*		*URI identifying a supported profile*
Semantics	*URI*	*		*URI of a document providing a human-readable description of the semantics of the endpoint functionalities*
Implementor	*String*	*0..1*		*Main organization implementing this software component*
ImplementationName	*String*	*0..1*		*Name of the implementation*
ImplementationVersion	*String*	*0..1*		*Version of the implementation (e.g., major version.minor version.patch version)*
QualityLevel	*QualityLevel_t*	*1*		*Maturity of the endpoint in terms of quality of the software components*
HealthState	*EndpointHealthState_t*	*1*		*A state representing the health of the endpoint in terms of its capability of properly delivering the functionalities*
HealthStateInfo	*String*	*0..1*		*Textual explanation of the state endpoint*
ServingState	*ServingState_t*	*1*		*A state specifying if the endpoint is accepting new requests and if it is serving the already accepted requests*
StartTime	*DateTime_t*	*0..1*		*The timestamp for the start time of the endpoint*
IssuerCA	*DN_t*	*0..1*		*Distinguished name of Certification Authority issuing the certificate for the endpoint*
TrustedCA	*DN_t*	*		*Distinguished name of the trusted Certification Authority*
DowntimeAnnounce	*DateTime_t*	*0..1*		*The timestamp for the announcement of the next scheduled downtime*
DowntimeStart	*DateTime_t*	*0..1*		*The starting timestamp of the next scheduled downtime*
DowntimeEnd	*DateTime_t*	*0..1*		*The ending timestamp of the next scheduled downtime*
DowntimeInfo	*String*	*0..1*		*Description of the next scheduled downtime*
Property	Type	Mult.	Unit	Description
No extra properties are defined in the specialized entity				
Association End		Mult.	Description	
StorageService.ID [redefines Service.ID]		1	A storarge endpoint is part of a storage service	
StorageShare.LocalID [redefines Share.LocalID]		*	A storage endpoint can pass activities to zero or more storage shares	
Inherited Association End		Mult.	Description	
Extension.Key		*	The entity can be extended via key-value pairs	
AccessPolicy.ID		*	An endpoint has assocated zero or more AccessPolicies	

7.5 StorageShare

Entity	Inherits from			Description
StorageShare	Share			A utilization target for a set of storage resources defined by a set of configuration parameters and characterized by status information
Inherited Property	*Type*	*Mult*	*Unit*	*Description*
LocalID [key]	*LocalID_t*	*1*		*An opaque identifier local to the associated Service*
Name	*String*	*0..1*		*Human-readable name*
Description	*String*	*0..1*		*Description of this share*
Property	Type	Mult.	Unit	Description
ServingState	ServingState_t	1		A state specifying if the share is open to place new requests and if it is open to offer the already present requests for execution
Path	String	1		A namespace where files are logically assigned to when they are stored into this share
SharingID	LocalID_t	1		Local ID common to the storage shares which use the same storage share capacities (‘dedicated’ is a reserved term and means that the storage share capacities are not shared with other storage share capacities part of different storage shares)
AccessLatency	AccessLatency_t	1		The maximum latency category for a file stored in

				this share to be made available for reading
RetentionPolicy	RetentionPolicy_t	*		The quality of retention, which indicates the probability of the storage system losing a file
ExpirationMode	ExpirationMode_t	0..3		Support for files with infinite and/or finite lifetimes, and what actions the storage service may take upon the expiration of a file
DefaultLifeTime	UInt32	0..1	s	The default lifetime assigned to the file if no explicit lifetime is specified
MaximumLifeTime	UInt32	0..1	s	The maximum lifetime that can be requested for a file
Tag	String	0..1		A user defined tag for additional information
OtherInfo	String	*		Placeholder to publish info that does not fit in any other attribute. Free-form string, comma-separated tags, (name, value) pair are all examples of valid syntax
Association End		**Mult.**	**Description**	
StorageEndpoint.ID [redefines Endpoint.ID]		*	A storage share is consumed via zero or more endpoints	
StorageResource.ID [redefines Resource.ID]		*	A storage share is defined on zero or more storage resources	
StorageService.ID [redefines Service.ID]		1	A storage share participates in a storage service	
StorageShareCapacity.LocalID		*	A storage share offers zero or more storage share capacities	
Inherited Association End		**Mult.**	**Description**	
Extension.Key		*	The entity can be extended via key-value pairs	
MappingPolicy.ID		*	A share has zero or more mapping policies	

A storage share represents a utilization target of a storage extent which policies are homogeneous. If many user domains are mapped to a storage share via a mapping policy, then they compete to the usage without any differentiation. A storage share can have many types of storage extents. The status of each type of storage extent as regards the usage by the user domains is described by the StorageShareCapacity.

7.6 StorageShareCapacity

Entity	Inherits from			Description
StorageShareCapacity	Entity			Description of the size and usage of an homogenous storage extent available to a storage share
Property	**Type**	**Mult.**	**Unit**	**Description**
LocalID	LocalID_t	1		An opaque identifier local to the Storage Service
Type	StorageCapacity_t	1		Type of storage capacity
TotalSize	UInt64	0..1	GB	Size of dedicated storage extent
FreeSize	UInt64	0..1	GB	Size of free storage extent
UsedSize	UInt64	0..1	GB	Size of used storage extent
ReservedSize	UInt64	0..1	GB	Size of reserved storage extent
OtherInfo	String	*		Placeholder to publish info that does not fit in any other attribute. Free-form string, comma-separated tags, (name, value) pair are all examples of valid syntax
Association End		**Mult.**	**Description**	
StorageShare.LocalID		1	A storage share capacity is related to one storage share	
Inherited Association End		**Mult.**	**Description**	
Extension.Key		*	The entity can be extended via key-value pairs	

The storage share capacity is useful to express the usage information of an homogenous storage extent allocated to a share. Such usage information refers to the user domains which are related to the storage share via mapping policies.

7.7 StorageManager

Entity	Inherits from	Description
StorageManager	Manager	The primary software component locally

				managing one or more storage resources. It can describe also aggregated information about the managed resources.
Inherited Property	**Type**	**Mult**	**Unit**	**Description**
ID [key]	*URI*	*1*		*A global unique ID*
Name	*String*	*0..1*		*Human-readable name*
Property	**Type**	**Mult.**	**Unit**	**Description**
Type	StorageManagerType_t	1		Type of the storage manager
Version	String	0..1		Version of the storage manager
OtherInfo	String	*		Placeholder to publish info that does not fit in any other attribute. Free-form string, comma-separated tags, (name, value) pair are all examples of valid syntax
Association End		**Mult.**	**Description**	
StorageService.ID [redefines Service.ID]		1	A storage manager participates in a storage service	
StorageResource.ID [redefines Resource.ID]		*	A storage manager manages zero or more storage resources	
Inherited Association End		**Mult.**	**Description**	
Extension.Key		*	The entity can be extended via key-value pairs	

7.8 StorageResource

Entity	Inherits from			Description
StorageResource	Resource			Abstracted of a sufficiently homogeneous storage device providing a storage capacity, managed by a local software component (storage manager), part of a storage service, reachable via zero or more endpoints and having zero or more shares defined on it. A storge resource refers to a category with summary information on the capacity
Inherited Property	**Type**	**Mult.**	**Unit**	**Description**
ID [key]	*URI*	*1*		*A global unique ID*
Name	*String*	*0..1*		*Human-readable name*
Property	**Type**	**Mult.**	**Unit**	**Description**
Type	StorageResourceType_t	1		Type of storage resource
Latency	AccessLatency_t	1		The actual latency category for a file stored in this resource to be made available for reading
TotalSize	UInt64	0..1	GB	Size of storage extent
FreeSize	UInt64	0..1	GB	Size of free storage extent
UsedSize	UInt64	0..1	GB	Size of used storage extent
OtherInfo	String	*		Placeholder to publish info that does not fit in any other attribute. Free-form string, comma-separated tags, (name, value) pair are all examples of valid syntax
Association End		**Mult.**	**Description**	
StorageManager.ID [redefines Manager.ID]		1	A storage resource is managed by a storage manager	
StorageShare.LocalID [redefines Share.LocalID]		*	A storage resource provides capacity in terms of zero or more storage shares	
Inherited Association End		**Mult.**	**Description**	
Extension.Key		*	The entity can be extended via key-value pairs	

7.9 ToComputingService

Entity	Inherits from			Description
ToComputingService	Entity			Description of the network link quality between a storage service and a computing service and of a potentially dedicated access protocol that the computing service can use to access the storage service
Property	**Type**	**Mult.**	**Unit**	**Description**
LocalID	LocalID_t	1		An opaque identifier local to the

				Storage Service
NetworkInfo	NetworkInfo_t	0..1		Type of network available among the storage service and computing service
Bandwidth	UInt32	0..1	Mb/s	The nominal bandwidth available between the storage service and computing service
OtherInfo	String	*		Placeholder to publish info that does not fit in any other attribute. Free-form string, comma-separated tags, (name, value) pair are all examples of valid syntax
Association End			**Description**	
StorageAccessProtocol.LocalID		1	The storage service can be accessed via an access protocol by a certain computing service	
ComputingService.ID		1	Is associated to a computing service	
StorageService.ID		1	Is associated to a storage service	
Inherited Association End		**Mult.**	**Description**	
Extension.Key		*	The entity can be extended via key-value pairs	

8. Relationship to OGF Reference Model

In this section, we describe the integration of the GLUE information model with the OGF Reference Model. The reference model defines the concept of Grid Component.

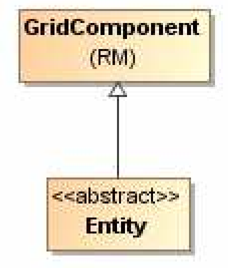

Figure 4 GLUE and Reference Model integration

9. Security Considerations

Please refer to RFC 3552 (http://www.ietf.org/rfc/rfc3552.txt) for guidance on writing a security considerations section. This section is required in all documents, and should not just say "there are no security considerations." Quoting from the RFC:

> "Most people speak of security as if it were a single monolithic property of a protocol or system, however, upon reflection, one realizes that it is clearly not true. Rather, security is a series of related but somewhat independent properties. Not all of these properties are required for every application.
>
> We can loosely divide security goals into those related to protecting communications (COMMUNICATION SECURITY, also known as COMSEC) and those relating to protecting systems (ADMINISTRATIVE SECURITY or SYSTEM SECURITY). Since communications are carried out by systems and access to systems is through communications channels, these goals obviously interlock, but they can also be independently provided."

10. Author Information

Sergio Andreozzi, INFN
Stephen Burke, RAL
Felix Ehm, CERN
Laurence Field, CERN
Gerson Galang, ARCS
Balazs Konya, Lund University
Maarten Litmaath, CERN
Paul Millar, DESY
JP Navarro, ANL

11. Contributors & Acknowledgements

We gratefully acknowledge the contributions made to this document (in no particular order) by Shiraz Memon, Matt Viljonen and Steve Traylen.

12. Intellectual Property Statement

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

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

13. Disclaimer

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

14. Full Copyright Notice

Copyright (C) Open Grid Forum (2008). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the OGF or other organizations, except as needed for the purpose of developing Grid Recommendations in which case the procedures for copyrights defined in the OGF Document process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the OGF or its successors or assignees.

15. References

[glue-wg] The GLUE Working Group of OGF, *https://forge.gridforum.org/sf/projects/glue-wg*
[glue-usecases] GLUE 2.0 Use Cases (early draft), *https://forge.gridforum.org/sf/go/doc14621*
[glue-1.x] The GLUE Schema 1.3, *https://forge.gridforum.org/sf/go/doc14185*
[ng-schema] The NorduGrid/ARC Information System, NORDUGRID-TECH 4, *https://forge.gridforum.org/sf/go/doc14273*
[naregi-schema] NAREGI information and data model, *https://forge.gridforum.org/sf/go/doc14300*
[ogf-ts] Technical Strategy for the Open Grid Forum 2007-2010. GFD-I.113. http://www.ogf.org/documents/GFD.113.pdf
[omii-jra2-djra2.1] Sergio Andreozzi, Antonia Ghiselli, Chunming Hu, Jinlei Jiang, Balazs Konya, Morris Riedel, Davy Virdee, Li Zha. D:JRA2.0 Report on Grid Activities relevant to the identification of new services http://omii-europe.org/OMII-Europe/News/DJRA20.pdf

[XSRL] NorduGrid XRSL (Extended Resource Specification Language) - http://www.nordugrid.org/documents/xrsl.pdf
[EBNF] Extended Backus–Naur form. ISO/IEC 14977 : 1996(E) http://www.cl.cam.ac.uk/~mgk25/iso-14977.pdf
[srmv1] Storage Resource Manager (SRM) Joint Design. http://sdm.lbl.gov/srm-wg/doc/srm.v1.0.pdf
[srmv2] Storage Resource Manager Interface Specification V2.2. http://sdm.lbl.gov/srm-wg/doc/SRM.v2.2.html
[cream] gLite CREAM (Computing Resource Execution And Management). http://grid.pd.infn.it/cream/
[gram] Globus Resource Allocation Protocol. http://www.globus.org/api/c-globus-2.2/globus_gram_documentation/html/index.html
[ogf-gfd80] I. Foster, H. Kishimoto, A. Savva, D. Berry, A. Grimshaw, B. Horn, F. Maciel, F. Siebenlist, R. Subramaniam, J. Treadwell, J. Von Reich. The Open Grid Services Architecture, Version 1.5. OGF GFD-80. 24 Jul 2006

16. Appendix A: Place-holder values for unknown data

Whilst people endeavor to provide accurate information, there may be situations where specific GLUE attributes may be assigned place-holder (or dummy) values. These place-holder values carry some additional semantic meaning; specifically, that the correct value is currently unknown and the presented value should be ignored. This appendix describes a set of such place-holder values.

Some attributes within the GLUE schema are required whilst others are optional. If the attribute is optional and the corresponding information is unavailable, the information provider must either publish a place-holder or not to publish the attribute. If the attribute is required, then the information must either publish a place-holder value or refrain from publishing the GLUE object.

If a place-holder value is published, it must conform to the scheme described in this appendix. This is to increase the likelihood that software will understand the nature of the information it receives.

This appendix describes place-holder values that have be chosen so they are obvious "wrong" to humans, unlikely to occur under normal operation and valid within the attribute type. This also allows for detection of failing information provider components.

16.1 Use cases

There are two principle use-cases for place-holder values, although others may exist.

Scenario 1. a static value has no good default value and has not been configured for a particular site.

Some provisions for GLUE Schema provide templates. These templates may contain attributes that have no good default value; for example, supplying the correct value may require site-specific knowledge. Whilst it is expected that these attributes be configured, it is possible that this does not happen, so exposing the attributes' default values.

Scenario 2. information provider is unable to obtain a dynamic value.

A dynamic value is provided by an information provider by querying the underlying grid resources. This query will use a number of ancillary resources (e.g., DNS, network hardware) that might fail; the grid services might also fail. If an attribute is required and the current value is unobtainable, a place-holder value must be used.

16.2 Place-holder values

This section describes a number of values that can be represented within a given address space (e.g., Strings/UTF-8, Integers, FQDNs, IPv4 address space). Each of the different types are introduced along with the place-holder value and a brief discussion on usage, rational
and any other considerations.

16.3 Simple strings

(ASCII/UTF-8) should use "UNDEFINEDVALUE" or should start "UNDEFINEDVALUE:"

Upper-case letters make it easier to spot and a single word avoids any white-space issues.

A short error message can be incorporated into the message by appending the message after the colon.

Examples:
UNDEFINEDVALUE
UNDEFINEDVALUE: unable to contact torque daemon.

Using UNDEFINEDVALUE is a default option for strings that have no widely-known structure. If a value is of a more restrictive sub-type (e.g., FQDNs, FQANs, URIs) described below, then the rules for more restrictive form must be used.

16.3.1 Fully qualified domain names

They must use a hostname ending either "example.org" for scenario 1, or "invalid" for scenario 2.

RFC 2606 defines two second-level domains: "example.org" and "example.com". These domains have the advantage of ending with a recognisable TLD, so are recognisable as a DNS name. Default configuration (scenario 1, above) must use DNS names that end "example.org"

RFC 2606 also reserves the "invalid" Top-Level-Domain (TLD) as always invalid and clearly so. For dynamic information gathering, a value ending "invalid" must be used.

In both cases, additional information may be included by specifying a prefix to "example.org" or "invalid". This may be used to specify the class of machine that should be present. For dynamic infomation, if the class of machine is not published then the FQDN "unknown.invalid" must be used.

Examples:
www.example.org
your-CE.example.org
unknown.invalid
site-local-BDII.invalid

16.3.2 IPv4 address

It must use 192.0.2.250

There are several portions of IPv4 addresses that should not appear on a network, but none that are reserved for documentation or to specify a non-existent address. Using any address leads to the risk of side-effects, should this value be used.

The best option is an IP address from the 192.0.2.0/24 subnet. This subnet is defined in RFC 3330 as "TEST-NET" for use in documentation and example code. For consistency, the value 192.0.2.250 must be used.

16.3.3 IPv6 addr

It must use 2001:DB8::FFFF

There is no documented undefined IPv6 address. RFC 3849 reserves the address prefix 2001:DB8::/32 for documentation. For consistency, the address 2001:DB8::FFFF must be used.

16.3.4 Integers

It must use "all nines"
For uint32/int32 this is 999,999,999
For uint64/int64 this is 999,999,999,999,999,999

For integers, all numbers expressible within the encoding (int32/uint32/etc.) are valid so there is no safe choice.

If an unsigned integer is encoded as a signed integer, it is possible to use negative numbers safely. However, these numbers will be unrepresentable if the number is stored as an unsigned integer. For this reason a negative number place-holder must not be used.

The number was chosen for three reasons. First, attribute scales are often chosen to reduce the likelihood of overflow: numbers towards MAXINT (the large number representable in an integer domain) are less likely to appear. Second, repeated numbers stand out more clearly to humans. Finally, the statistical frequency of measured values often follows Benford's law, which indicates that numbers starting with "1" occur far more frequently than those starting with "9" (about six times more likely). For these reasons, information providers must use all-nines to indicate a `place-holder`.

16.3.5 File path

It must start either "/UNDEFINEDPATH" or "\UNDEFINEDPATH".

As with the simple string, a single upper-case word is recommended. The initial slash indicates that the value is a path. Implementations must use whichever slash is most appropriate for the underlying system (Unix-like systems use a forward-slash). Software should accept either value as an unknown-value place-holder.

Additional information can be encoded as data beyond the initial UNDEFINEDPATH, separated by the same slash as started the value. Additional comments should not use any of the following characters: \ [ ] ; = " : | , * .

Examples:
/UNDEFINEDPATH
\UNDEFINEDPATH
/UNDEFINEDPATH/Path to storage area
/UNDEFINEDPATH/Broker unavailable

16.3.6 Email addresses

It must use an undefined FQDN for the domain.

RFC 2822 defines emails addresses to have the form: <local-part> '@' <domain>

The <domain> must be an undefined FQDN; see above for a complete description. For email addresses, information providers should use "example.org" for scenario 1. and "unknown.invalid" for scenario 2.

The <local-part> may be used to encode a small amount of additional information; for example, it may indicate the class of user to whom the email address should be delivered. If no such information is to be encoded the value "user" must be used.

Examples:
user@example.org
user@unknown.invalid
site-local-contact@example.org
local-admin@example.org

16.3.7 Uniform Resource Identifier (URI)

It is schema-specific

RFC 3986 defines URIs as a "federated and extensible naming system." All URIs start with a schema-name part (e.g., "http") and no schema-name has been reserved for undefined or documenting example values.

For any given URI schema ("http", for example), it may be possible to define an unknown value within that name-space. If a GLUE value has only one valid schema, the undefined value must be taken from that schema. If several schemata are possible, one must be chosen from the available options. This should be the most commonly used.

Take care with the URI encoding. All place-holder URI values MUST be valid URIs. If additional information is included, it must be encoded so the resulting URI is valid.

For schemata that may include a FQDN (e.g., a reference to an Internet host), an undefined URI must use an undefined FQDN; see above for details on undefined FQDNs.

URI schemata that reference a remote file (e.g., "http", "ftp", "https"), additional information may be included as the path. The FQDN indicates that the value is a place-holder, indicating an unknown value, so information providers should not specify "UNDEFINEDPATH".

For "file" URIs, the path part must identify the value as unknown and must use the forward-slash variant; see above for details on undefined paths.

For "mailto" URIs [RFC 2368] encapsulates valid email addresses with additional information (such as email headers and message body). Place-holder mailto URIs MUST use an unknown email address (see above). Any additional information must be included in the email body.

There may be other schemata in use that are not explicitly covered in this section. A place-holder value should be agreed upon within whichever domain such schemata are used. This place-holder value should be in the spirit of the place-holder values described so far.

Examples:
http://www.example.org/
httpg://your-CE.example.org/path/to/end-point
httpg://unknown.invalid/User%20certificate%20has%20expired
mailto:site-admin@example.org
mailto:user@maildomain.invalid?body=Problem%20connecting%20to%20WLMS
file:///UNDEFINEDPATH
file:///UNDEFINEDPATH/path%20to%20some%20directory

16.3.8 X509 Distinguished Names

It must start /O=Grid/CN=UNDEFINEDUSER

X509 uses a X500 namespace, represented as several Relative Domain-Names (RDNs) concatenated by forward-slashes. The final RDN is usually a single common name (CN), although multiple CNs are allowed.

Unknown DN values must have at least two entries: an initial O=Grid followed immediately by CN=UNDEFINEDUSER.

Additional information can be encoded using extra CN entries. These must come after CN=UNDEFINEDUSER.

Examples:

```
/O=Grid/CN=UNDEFINEDUSER
/O=Grid/CN=UNDEFINEDUSER/CN=Your Grid certificate DN here
/O=Grid/CN=UNDEFINEDUSER/CN=Cannot access SE
```

16.3.9 Fully Qualified Attribute Name (FQAN)

It must use a VO of "vo.example.org" (for scenario 1.) or "unknown.invalid" (for scenario 2).

The "VOMS Credential Format" document,

http://edg-wp2.web.cern.ch/edg-wp2/security/voms/edg-voms-credential.pdf

states that FQANs must have the form:

```
/VO[/group[/subgroup(s)]][/Role=role][/Capability=cap]
```

Where VO is a well-formed FQDN. Unlike FQDNs, VO names must be lower-case. The place-holder value for FQAN is derived from the place-holder FQDN (see Section 16.3.1). It must have no subgroup(s) or Capability specified.

Any additional information must be encoded within a single Role name. Care should be taken that only valid characters (A-Z, a-z, 0-9 and dash) are included.

Examples:

```
/vo.example.org
/vo.example.org/Role=Replace-this-example-with-your-FQAN
/unknown.invalid
/unknown.invalid/Role=Unable-to-contact-CE-Error-42
```

16.3.10 Geographic locations

It must use longitude 0 degrees, latitude 0 degrees.

Meridians of longitude are taken from (-180,180] degrees, whilst parallels of latitude are taken from [-90,90] degrees. For a place-holder value to be a valid location, it must also be taken from these ranges.

By a happy coincidence, the (0,0) location is within the Atlantic Ocean, some 380 miles (611 kilometers) south of the nearest country (Ghana). Since this location is unlikely to be used and repeated numbers are easier for humans to spot, (0,0) must be used to specify an place-holder location.

17. Appendix B: Data Types

This section contains the definition of property types defined within this model. The enumeration types can be either closed or open. For properties which type defines a closed enumeration, one of the defined values MUST be chosen; any other value is not valid. For properties which type defines an open enumeration, one of the defined values MAY be chosen, nevertheless any other value compatible with the string type and with the recommended syntax is allowed.

The enumeration values MUST be lower-case.

17.1 LocalID_t

The base type is the string with the following restrictions:

- first char in a-zA-Z
- following characters in [\w\-\.\:]
    - \w = [a-zA-Z_0-9]

17.2 ContactType_t

Open enumeration

Value	Description
security	Contact for persons responsible for the security
sysadmin	Contact for the system administration
usersupport	Contact for the user support
general	Contact for persons to ask about general issues

17.3 PolicyScheme_t

Open enumeration

Value	Description
basic	The basic scheme
gacl	GridSite Access Control List

For the *basic* policy scheme, the following syntax MUST be used (defined in EBNF form [EBNF]):

- basic rule ::= [‘DENY:’] (DN_RULE | FQAN_RULE | ‘ALL’)
- DN_RULE ::= ‘dn:’ DN_NAME
- FQAN_RULE ::= ‘fqan:’ VO_NAME (‘/’ GROUP_NAME)* (‘/Role=’ ROLE_NAME)?
- VO_NAME ::= [a-zA-Z0-9-_\.]+
- GROUP_NAME ::= VO_NAME
- ROLE_NAME ::= VO_NAME

The ‘DENY override’ matching algorithm MUST be used.

Examples of policies expressed using this syntax are:

- `dn:/C=XX/O=YYYY/OU=Personal Certificate/L=ZZZZ/CN=NAME SURNAME/Email=account@domain.org`
    - matches the user proving to have a certificate identified by this DN
- `fqan:/vo_a`
    - matches all the users proving to be part of the vo_a
- `fqan:/vo_a/group_a`
    - matches all the users proving to be part of group_a or one of its subgroups
- `fqan:/vo_a/group_a/Role=prod`
    - matches all the users proving to be part of group_a and having the Role prod

17.4 DN_t

Distinguished Name as defined by RFC 4514 (http://www.rfc-editor.org/rfc/rfc4514.txt).

17.5 Capability_t

List of values initially drafted from [omii-jra2-djra2.1, OGF-GFD80]. Open enumeration.

Value	Description
security.authentication	Capacity of providing authentication mechanisms for Grid users machine and services
security.credentialstorage	Capacity of providing an online credential repository that allows users to securely obtain credentials when and where needed
security.delegation	capacity for a user to give a service the authority to undertake specific activities or decisions on its behalf
security.authorization	capacity of handling authorization aspects, making authorization decisions about the subject and the requested mode of access based upon combining information from a number of distinct sources
security.identymapping	capacity of mapping Grid-level credentials to local level credentials (e.g., mapping a user X.509 certificate into a UNIX account).
security.attributeauthority	capacity of associating a user with a set of attributes in a trusted manner to a relying party, by way of digitally signed assertions
security.accounting	capacity of systematically recording, reporting, and analyzing the usage of resources
data.transfer	capacity of moving a file from one network location to another. It refers to the actual transfer (e.g., as performed by protocols like FTP, GridFTP, or HTTP)
data.management.transfer	capacity of managing a transfer of files from the start to the completion
data.management.replica	capacity of managing the creation of file replicas upon request
data.management.storage	capacity of managing a storage resource, from simple systems like disk-servers to complex hierarchical systems
data.naming.resolver	capacity of resolving one name to another (for example, search the associated abstract name to a certain human-oriented name)
data.naming.scheme	capacity of attaching names to data resources. (To evaluate if it should moved to the main category infrastructure instead of data). In OGSA, a three-level naming scheme is defined: (1) human-oriented name, (2) abstract name and (3) address
data.access.relational	capacity of providing access to a relational data source
data.access.xml	capacity of providing access to an XML data source
data.access.flatfiles	capacity of providing access to a flat file
information.model	capacity of modelling resources based on a community accepted definition
information.discovery	capacity of locating unknown resources or services, possibly satisfying a set of requirements
information.logging	capacity of recording data, often chronologically
information.monitoring	capacity of periodically observing measurements, transform them and make available to users or other applications
information.provenance	capacity of providing long-term storage of information related to Grid activity and to let this information be accessed by users or other applications.
executionmanagement.jobexecution	capacity of executing a job or set of jobs.
executionmanagement.jobdescription	capacity of letting users be able to describe a job submission request based on a machine-processable language
executionmanagement.jobmanager	capacity of managing the execution of a job or set of jobs from start to finish
executionmanagement.executionandplanning	capacity of building schedules for jobs, that is, the capability of defining mappings between services and resources, possibly with time constraints
executionmanagement.candidatesetgenerator	capacity of determining the set of resources on which a nit of workcan execute
executionmanagement.reservation	capacity of managing reservation of resources for future usage
executionmanagement.dynamicvmdeploy	capacity of dynamically deploying a virtual machine image in a worker node

17.6 ServiceType_t

The RECOMMENDED syntax is in reverse-DNS style. The first element is a top-level domain, while the second element is the middleware name (e.g., for gLite services, org.glite.[service type]). Open enumeration.

Value	Description
org.glite.wms	gLite Workload Management Service
org.glite.lb	gLite Logging and Booking Service
org.glite.fts	gLite File Transfer Service
org.nordugrid.arex	NorduGrid Resource Coupled Execution Service
org.nordugrid.isis	NorduGrid Information Index Service
org.nordugrid.storage	NorduGrid Storage Service
org.teragrid.condor-g	TeraGrid Condor-g
org.teragrid.globus-mds4	TeraGrid Globus MDS 4
org.teragrid.gpfs	TeraGrid GPFS
org.teragrid gridftp	TeraGrid GridFTP
org.teragrid.gsi-openssh	TeraGrid gsi-enabled openssh
org.teragrid.prewsgram	TeraGrid pre-WS Globus GRAM
org.teragrid.srb	TeraGrid Storage Resource Broker
org.teragrid.ws-delegation	TeraGrid WS-Delegation Service
org.teragrid.ws-gram	TeraGrid WS-GRAM Service
org.teragrid.ws-ogsadai	TeraGrid OGSA-DAI
org.teragrid.rft	TeraGrid Reliable File Transfer

17.7 QualityLevel_t

Closed enumeration

Value	Description
development	The component is under active development both in functionalities and interfaces
testing	The component has completed the development phase and is under testing
pre-production	The component has completed the development and passed the testing phase; it is being used in real world scenarios
production	The component completed the development and is considered stable for real world scenarios

17.8 EndpointTechnology_t

Open enumeration.

Value	Description
webservice	The endpoint is implemented as a Web Service
jndi	The endpoint is implemented using JNDI
legacy	The endpoint is implemented using legacy technologies
corba	The endpoint is implemented using CORBA technologies

17.9 EndpointHealthState_t

Closed enumeration

Value	Description
ok	It was possible to check the state of the endpoint and it appeared to be functioning properly
warning	It was possible to check the state of the endpoint, but it appeared to be above some "warning" threshold or did not appear to be working properly
critical	It was possible to check the state of the endpoint and either it was not running or it was above some "critical" threshold
unknown	It was not possible to check the state of the endpoint
other	It was possible to check the state of the endpoint, but this is not covered by the defined states

17.10 ServingState_t

Closed enumeration

Value	Description
production	The endpoint is both accepting and serving requests
draining	The endpoint is not accepting requests, but is serving requests in the queue
queueing	The endpoint is accepting requests, but is not serving them
closed	The endpoint is not accepting request nor is serving them

17.11 DateTime_t

The DateTime_t is based on the extended ISO 8061 format:

- [-]CCYY-MM-DDThh:mm:ss[Z|(+|-)hh:mm]

This data type maps the dateTime XSD simple type. We restrict this syntax to UTC time zone as follows:

- yyyy '-' mm '-' dd 'T' hh ':' mm ':' ss 'Z'

17.12 Staging_t

Closed enumeration:

Value	Description
none	No staging of files supported
stagingin	Automatic staging in of files supported
stagingout	Automatic staging out of files supported
staginginout	Automatic staging in and out of files supported

17.13 JobDescription_t

Open enumeration:

Value	Description
ogf:jsdl:1.0	Job Description Submission Language 1.0
egee:jdl	EGEE Job Description Language
nordugrid:xrsl	Nordugrid XSRL [XSRL]
globus:rsl	Globus RSL
condor	Condor

17.14 SchedulingPolicy_t

Open enumeration:

Value	Description
fairshare	Statistically guarantees the allocated share
fifo	First-In First-Out
random	Random choice

17.15 ReservationPolicy_t

Closed enumeration:

Value	Description
none	No reservation is supported
mandatory	Jobs must be submitted only via advance reservation
optional	Jobs can be submitted via advance reservation, but this is not required

17.16 ComputingManagerType_t

Open enumeration:

Value	Description
lsf	Platform Load Sharing Facility
sungridengine	Sun Grid Engine
openpbs	Open PBS
torque	Torque
torquemaui	Torque with MAUI
bqs	CC-IN2P3 Batch Queue System
condor	Condor
loadleveler	IBM LoadLeveler
fork	Based on fork primitive

17.17 NetworkInfo_t

Open enumeration

Value	Description
100megabitethernet	Network based on 100 MBit/s Ethernet technology
gigabitethernet	Network based on 1 GBit/s Ethernet technology
myrinet	Network based Myrinet technology
infiniband	Network based on Infiniband technology

17.18 Benchmark_t

Open enumeration

Value	Description
specint2000	SPECint2000 integer benchmark
specfp2000	SPECfp2000 floating point benchmark
cint2006	SPEC CINT 2006 integer benchmark
cfp2006	SPEC CFP 2006 floating point benchmark
bogomips	BogoMips
linpack	LINPACK benchmark

17.19 Platform_t

Open enumeration:

Value	Description
i386	Intel 386 architecture
amd64	AMD 64bit architecture
itanium	Intel 64-bit architecture
powerpc	PowerPC architecture
sparc	SPARC architecture

17.20 CPUMultiplicity_t

Closed enumeration:

Value	Description
singlecpu-singlecore	The execution environment is run by a single physical CPU with a single core
singlecpu-multicore	The execution environment is run by a single physical CPU with multiple cores
multicpu-singlecore	The execution environment is run by multiple physical CPUs with a single core each
multicpu-multicore	The execution environment is run by multiple physical CPUs with a multiple cores each

17.21 OSFamily_t

Open enumeration:

Value	Description
linux	Family of operating systems based on Linux kernel
macosx	Family of operating systems based on MacOS X
windows	Family of operating systems based on Windows
solaris	Family of operating systems based on Solaris

17.22 ParallelSupport_t

Open enumeration:

Value	Description
mpi	Parallel execution based on mpi library
openmp	Parallel execution based on openmp library
none	No supported parallel execution

17.23 AppEnvState_t

Open enumeration:

Value	Description
notinstallable	The application environment is not installed and not installable
installable	The application environment is not installed, but can be dynamically installed
installingmanually	The application environment is not installed, but is being installed manually
installingautomatically	The application environment is not installed, but is being installed automatically
installationfailed	The application environment was being installed, but the installation process failed
installednotverified	The application environment is installed, but not yet verified
installedverified	The application environment is installed and successfully verified
installedbroken	The application environment is installed, but the verification failed
pendingremoval	The application environment is installed, but will be removed as soon as possible
removing	The application environment is installed, but it is being removed

17.24 ApplicationHandle_t

Open enumeration:

Value	Description
module	Access based on loading modules via Environment Modules (http://modules.sourceforge.net/)
softenv	Access based on loading SoftEnv keys (http://www.mcs.anl.gov/systems/software/softenv/softenv-intro.html)
path	Access based on using an explicit path where the software is installed on the file system
executable	Access based on running directly the main executable of the application (this may require set-up of the environment)

17.25 OSName_t

Open enumeration:

Value	Description
scientificlinux	Scientific Linux
scientificlinuxcern	Scientific Linux CERN
ubuntu	Ubuntu
debian	Debian
centos	CentOS
fedora	RedHat Fedora
rhes	RedHat Enterprise Server
mandrake	Mandrake
suse	SUSE
leopard	Mac OS X 10.5 (Leopard)
windowsxp	Microsoft Windows XP
windowsvista	Microsoft Windows Vista

17.26 License_t

Closed enumeration:

Value	Description
opensource	Open Source license approved by the OSI (Open Source Initiative)
commercial	Commercial license
Other	Other type of license not matching any of the available values
unknown	Unknown license type

17.27 ComputingActivityType_t

Closed enumeration:

Value	Description
single	An individual stand-alone job
collectionelement	A job submitted as part of a collection of individual jobs which do not communicate among them
parallelelement	A job submitted as part of a collection of individual jobs which communicate among them
workflownode	A job submitted as part of a workflow

17.28 ComputingActivityState_t

For the values of this type, we RECOMMEND the following syntax:

- namespace:state
- namespace:state:substate

Open enumeration:

Value	Description
bes:pending	the service has created a record for an activity but not yet instantiated it on a suitable computational resource or enabled it to start execution on such a resource
bes:running	the activity is executing on some computational resource
bes:finished	(a terminal state): the activity has terminated successfully. Successful termination implies that the activity exited of its own accord rather than due to some failure in the BES or of the computational resources on which the activity was running. Note that a successfully terminating activity may nevertheless return an error code as its return value
bes:failed	(a terminal state): the activity has failed due to some system error/failure event, such as failure of a computational resource that the activity was running on
bes:terminated	(a terminal state): the client – which might be some system administrator (and hence not necessarily the client who originated the request to create the activity) – has issued a TerminateActivity request

For more information on the BES state model, see [BES].

This property type is an open enumeration. Examples of additional values are:

- a middleware provider is using its own state model defined before the BES specification:
    - NorduGrid defines the state *accepting* which can be represented as (see [ng-schema], page 28):
        - nordugrid:accepting
    - gLite WMS defines the state *scheduled* which can be represented as:
        - glite-wms:scheduled
    - gLite CREAM defines the state *registered* which can be represented as:
        - glite-cream:registered
        - (see https://edms.cern.ch/document/595770)
- a middleware provider defined an extension of BES state model which is not part of an official OGF specification
    - NorduGrid defined an extension the bes:pending by adding two substates:
        - nordugrid-bes:pending:accepting
        - nordugrid-bes:pending:accepted

17.29 StorageCapacity_t

Open enumeration:

Value	Description
online	
nearline	
offline	
cache	

17.30 StorageAccessProtocol_t

Open enumeration:

Value	Description
gsiftp	FTP with GSI authentication
file	POSIX access
nfs	Network File System protocol
afs	Andrew File System protocol
rfio	Remote File Input/Output protocol
gsirfio	RFIO with GSI authentication
dcap	DCache access protocol
gsidcap	DCAP with GSI authentication
root	File transfer protocol for the ROOT framework
https	Secured HyperText Transfer Protocol
http	HyperText Transfer Protocol

17.31 AccessLatency_t

Closed enumeration:

Value	Description
online	Files are always on disk, hence cannot have their latency improved
nearline	A file may have its only copies in a "nearly online" component of the storage system, typically a fully automated tape robot, but also a remote storage system could fit this qualification. Such a facility will need an unspecified amount of time to make a copy of the file available on the disk component of the container under consideration. When a file is not in use, its disk copies may be removed. Hence the system cannot guarantee that a file will be immediately available on disk
offline	A file may have its only copies in an offline component of the storage system, for example a tape library that is not connected to an automated tape robot. Hence an operator intervention may be needed to make a copy of a file available that has a lower latency

17.32 RetentionPolicy_t

Open enumeration:

Value	Description
custodial	Low probability of loss
output	An intermediate level and is appropriate for data which can be replaced by lengthy or effort-full processes
replica	The highest probability of loss, but is appropriate for data for which a certain amount of loss can be tolerated, in particular when other copies can be accessed in a timely fashion

17.33 ExpirationMode_t

Closed enumeration:

Value	Description
neverexpire	Support for files with infinite lifetime: they can only be removed by authorized clients, not by the storage system itself
warnwhenexpired	Support for files that have finite lifetimes, but on expiration cannot be removed by the storage system itself. The data content of an expired file may be deleted if it can be recovered from an archive. New store operations may fail for certain clients until (some of the) expired files have either been removed by authorized clients, or have had their lifetimes increased
releasewhenexpired	Support for files that have finite lifetimes and on expiration will be removed by the storage system

17.34 StorageManagerType_t

Open enumeration:

Value	Description
castor	CERN Advanced STOrage manager, disk and tape management system
gpfs	General Parallel File System, disk management system
dcache	Disk Cache, disk managing system with ability to control tape backends (e.g., Enstore)
tsm	IBM Tivoli Storage Manager, disk and tape management system
sse	Smart Storage Element, disk management system
enstore	Tape Storage system, tape management system

17.35 StorageResourceType_t

Open enumeration:

Value	Description
disk	The storage capacity is provided by magnetic disks
tape	The storage capacity is provided by magnetic tapes
optical	The storage capacity is provided by optical disks