

~~GWD-R, GWD-I or GWD-C,~~

Authors:

Sergio Andreatto* (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

Deleted: GWD-P-R

GLUE WG

<http://forge.ogf.org/sf/sfmain/do/viewProject/projects.glue-wg>

~~December 9, 2008~~

Deleted: May

GLUE Specification v. 2.0 (revision 2 after public comment)

Deleted: 20

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..... | 5 |
| 2. | Notational Conventions..... | 5 |
| 3. | General Statements..... | 5 |
| 4. | Template..... | 6 |
| 5. | Conceptual Model of the Main Entities..... | 7 |
| 5.1 | Extension..... | 8 |
| 5.2 | Location..... | 8 |
| 5.3 | Contact..... | 10 |
| 5.4 | Domain..... | 10 |
| 5.4.1 | AdminDomain..... | 10 |
| 5.4.2 | UserDomain..... | 11 |
| 5.5 | Service..... | 12 |
| 5.6 | Endpoint..... | 13 |
| 5.7 | Share..... | 14 |
| 5.8 | Manager..... | 14 |
| 5.9 | Resource..... | 15 |
| 5.10 | Activity..... | 15 |
| 5.11 | Policy..... | 16 |
| 5.11.1 | AccessPolicy..... | 16 |
| 5.11.2 | MappingPolicy..... | 17 |
| 6. | Conceptual Model of the Computing Service..... | 18 |
| 6.1 | ComputingService..... | 19 |
| 6.2 | ComputingEndpoint..... | 20 |
| 6.3 | ComputingShare..... | 21 |
| 6.4 | ComputingManager..... | 24 |
| 6.5 | Benchmark..... | 27 |
| 6.6 | ExecutionEnvironment..... | 27 |
| 6.7 | ApplicationEnvironment..... | 29 |
| 6.8 | ApplicationHandle..... | 29 |
| 6.9 | ComputingActivity..... | 30 |
| 6.10 | ToStorageService..... | 32 |
| 7. | Conceptual Model of the Storage Service..... | 33 |
| 7.1 | StorageService..... | 34 |
| 7.2 | StorageServiceCapacity..... | 35 |
| 7.3 | StorageAccessProtocol..... | 35 |
| 7.4 | StorageEndpoint..... | 35 |
| 7.5 | StorageShare..... | 36 |
| 7.6 | StorageShareCapacity..... | 37 |
| 7.7 | StorageManager..... | 37 |
| 7.8 | StorageResource..... | 38 |
| 7.9 | ToComputingService..... | 38 |
| 8. | Relationship to OGF Reference Model..... | 40 |
| 9. | Security Considerations..... | 41 |
| 9.1 | Communication security..... | 41 |
| 9.1.1 | Confidentiality..... | 41 |
| 9.1.2 | Data integrity..... | 41 |
| 9.1.3 | Peer Entity authentication..... | 41 |
| 9.2 | Non-repudiation..... | 41 |
| 9.3 | System security..... | 42 |
| 9.3.1 | Unauthorized usage..... | 42 |
| 9.3.2 | Inappropriate Usage..... | 42 |
| 9.4 | Specific attacks..... | 42 |
| 9.4.1 | Eavesdropping..... | 42 |
| 9.4.2 | Replay..... | 42 |

| | | |
|---------|--|----|
| 9.4.3 | Message insertion | 43 |
| 9.4.4 | Deletion..... | 43 |
| 9.4.5 | Modification | 43 |
| 9.4.6 | Man-in-the-middle | 43 |
| 9.4.7 | Denial of service attacks..... | 43 |
| 10. | Author Information | 44 |
| 11. | Contributors & Acknowledgements..... | 44 |
| 12. | Intellectual Property Statement | 45 |
| 13. | Disclaimer..... | 45 |
| 14. | Full Copyright Notice..... | 45 |
| 15. | References..... | 45 |
| 16. | Appendix A: Place-holder values for unknown data | 47 |
| 16.1 | Use cases..... | 47 |
| 16.2 | Place-holder values | 47 |
| 16.3 | Extended booleans | 47 |
| 16.4 | Simple strings | 48 |
| 16.4.1 | Fully qualified domain names | 48 |
| 16.4.2 | IPv4 address | 48 |
| 16.4.3 | IPv6 addr..... | 49 |
| 16.4.4 | Integers..... | 49 |
| 16.4.5 | File path | 49 |
| 16.4.6 | Email addresses..... | 49 |
| 16.4.7 | Uniform Resource Identifier (URI)..... | 50 |
| 16.4.8 | X509 Distinguished Names..... | 51 |
| 16.4.9 | Fully Qualified Attribute Name (FQAN) | 51 |
| 16.4.10 | Geographic locations | 51 |
| 17. | Appendix B: Data Types | 53 |
| 17.1 | ExtendedBoolean t..... | 53 |
| 17.2 | LocalID t..... | 53 |
| 17.3 | ContactType t..... | 53 |
| 17.4 | PolicyScheme t..... | 53 |
| 17.5 | DN t..... | 54 |
| 17.6 | Capability t..... | 54 |
| 17.7 | ServiceType t..... | 55 |
| 17.8 | QualityLevel t..... | 55 |
| 17.9 | EndpointTechnology t..... | 55 |
| 17.10 | EndpointHealthState t..... | 55 |
| 17.11 | ServingState t..... | 56 |
| 17.12 | DateTime t..... | 56 |
| 17.13 | Staging t..... | 56 |
| 17.14 | JobDescription t..... | 56 |
| 17.15 | SchedulingPolicy t..... | 56 |
| 17.16 | ReservationPolicy t..... | 57 |
| 17.17 | ComputingManagerType t..... | 57 |
| 17.18 | NetworkInfo t..... | 57 |
| 17.19 | Benchmark t..... | 57 |
| 17.20 | Platform t..... | 57 |
| 17.21 | CPUMultiplicity t..... | 58 |
| 17.22 | OSFamily t..... | 58 |
| 17.23 | ParallelSupport t..... | 58 |
| 17.24 | AppEnvState t..... | 58 |
| 17.25 | ApplicationHandle t..... | 58 |
| 17.26 | OSName t..... | 59 |
| 17.27 | License t..... | 59 |
| 17.28 | ComputingActivityType t..... | 59 |
| 17.29 | ComputingActivityState t..... | 59 |

| | |
|---|--------------------|
| 17.30 StorageCapacity t | 60 |
| 17.31 StorageAccessProtocol t | 60 |
| 17.32 AccessLatency t | 60 |
| 17.33 RetentionPolicy t | 61 |
| 17.34 ExpirationMode t | 61 |
| 17.35 StorageManagerType t | 61 |
| 17.36 StorageResourceType t | 61 |

Deleted:

| | |
|--|--------------------|
| 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 |

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].

The mapping to concrete data models is defined in a separated document [glue-real] (new mappings may appear in the future). Profile documents SHOULD appear to define how to generate and use the information in production scenarios (e.g., a profile can decide that an attribute which is optional in the conceptual model, is considered mandatory in a certain Grid infrastructure; or that optional attributes are never published).

Formatted: Body Text,
Justified, Space After: 14,15 pt

Deleted: ¶

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 Système International 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).

Formatted: English (U.S.)

In Appendix A, we provide guidelines for place-holder values that MUST be used when the attributes have no good default value or when the attribute cannot be measured for some reason.

Deleted: 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 regards 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 its 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 regarding 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. Such extensions MUST NOT be considered part of the GLUE specification, nevertheless we RECOMMEND submitting them to the GLUE WG for consideration.

4. Template

In order to enrich the UML Class Diagrams with additional information, a table for each UML class is provided. The descriptive table is 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. As regards the multiplicity, the value of zero means that it is allowed to refrain from publishing a value for the related property even though this can be measured.

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 | |

Deleted: ¶
¶

Deleted: 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.¶

Formatted: Tabs: 3,17 cm, Left

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.

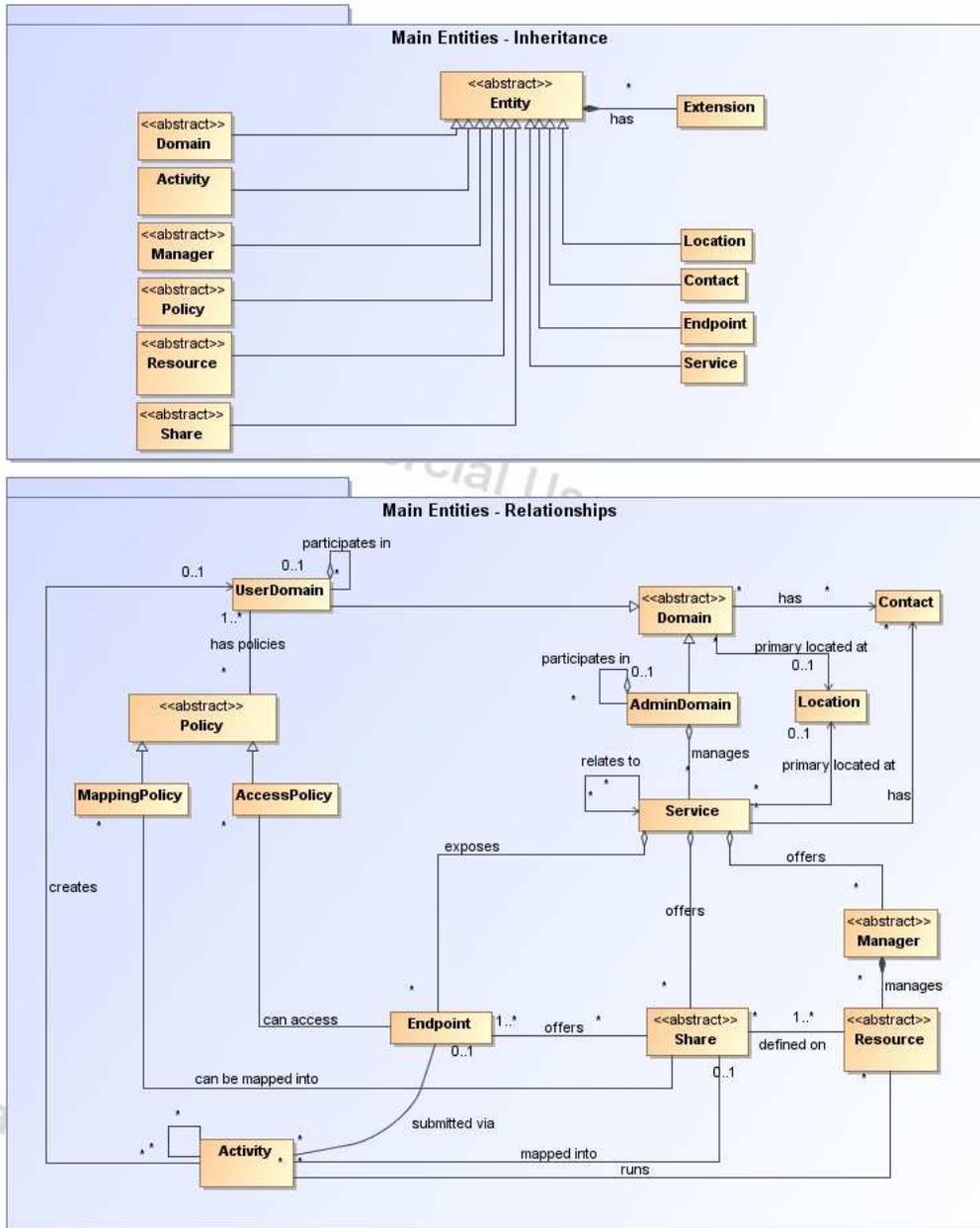


Figure 1 Entities and relationships for the Main Entities conceptual model

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 |
| <u>OtherInfo</u> | <u>String</u> | <u>*</u> | | <u>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</u> |
| 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 | | | <u>A geographical region where the granularity can vary from an exact position to spanning different countries not necessary connected</u> |
| 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) |

Deleted: A geographical position

| Association End | Mult. | Description |
|---------------------------|-------|--|
| Service.ID | * | The location is related to zero or more services |
| Domain.ID | * | The location is related to zero or more domains |
| <<abstract>> | * | |
| 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 |
| 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 | |

Deleted: OtherInfo ... [2]

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 |
| 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 | |

Deleted: OtherInfo ... [3]

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 |
| <i>ID</i> | <i>URI</i> | 1 | | <i>A global unique ID</i> |
| <i>Name</i> | <i>String</i> | 0..1 | | <i>Human-readable name</i> |
| <i>Description</i> | <i>String</i> | 0..1 | | <i>A description of the domain</i> |
| <i>WWW</i> | <i>URI</i> | * | | <i>The URL identifying a web page with more information about the domain</i> |
| Property | Type | Mult. | Unit | Description |
| Distributed | ExtendedBoolean_t | 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 (no particular format is defined) |
| 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 |

Deleted: OtherInfo ... [4]

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 |
| <i>ID</i> | <i>URI</i> | 1 | | <i>A global unique ID</i> |
| <i>Name</i> | <i>String</i> | 0..1 | | <i>Human-readable name</i> |
| <i>Description</i> | <i>String</i> | 0..1 | | <i>A description of the domain</i> |
| <i>WWW</i> | <i>URI</i> | * | | <i>The URL identifying a web page with more information about the domain</i> |
| 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 | URL | * | | The Endpoint URL 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 |

Deleted: I
Deleted: ID

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/>).

Deleted:

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 namespace-based classification (the namespace can be related to a middleware name, an organization or other concepts: org.glue and org.ogf are reserved) |
| QualityLevel | QualityLevel_t | 1 | | Maturity of the service in terms of quality of the software components; the value corresponds to the highest quality level among the available endpoints |
| 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. |
| 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 | |

Deleted: middleware

Deleted: OtherInfo ... [5]

| | | |
|---------------------------|-------|--|
| 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 issued by the CA are accepted for the authentication process |
| DowntimeAnnounce | DateTime_t | 0..1 | | The timestamp for the announcement of the next scheduled downtime |

Deleted: s

| | | | | |
|----------------------------------|--------------|--------------|--------------------|---|
| 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 | 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, then 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 provided 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 17.4). 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.

Deleted: 17.3

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. The evaluation algorithm for the rules SHOULD be defined by the policy scheme.

If an entity instance is associated to different Policy instances, each of them based on a different scheme, then the evaluation process SHOULD consider each set of policies independently. This means that the evaluation SHOULD rely on a certain policy scheme which is selected and understood by the consumer, and not by composing policies expressed using different schemes.

Deleted: 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 |
| <i>No extra properties are defined in the specialized entity</i> | | | | |
| 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 | |

Deleted: ¶

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 |
| <i>No extra properties are defined in the specialized entity</i> | | | | |
| 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.

When evaluating the mapping to a certain share using the algorithm implied by the policy scheme, if multiple solutions are available, then the consumer SHOULD NOT make any assumption on which share will be assigned to its activity and it SHOULD request a certain share explicitly.

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.

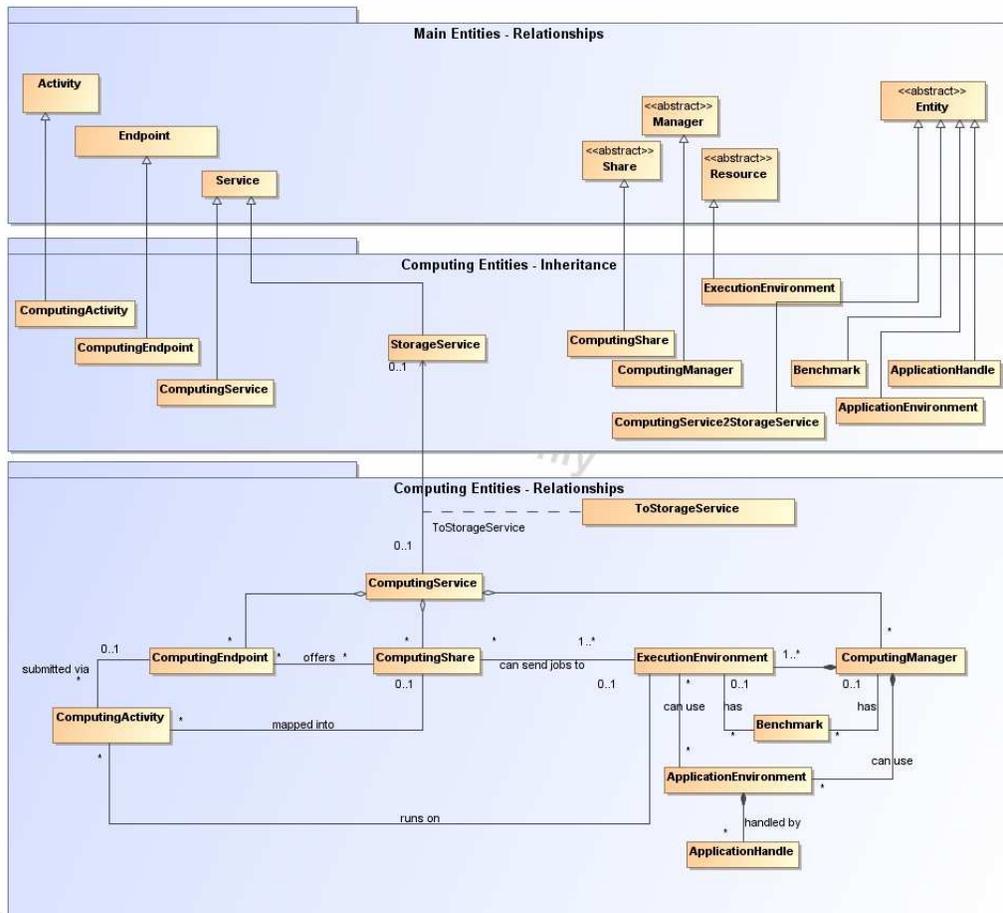


Figure 2 Entities and relationships for the Computing Service conceptual model

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

Throughout the specification, we also use the concept of storage extent to mean the capabilities and management of the various media that exist to store data and allow data retrieval.

Formatted: Font: Bold

6.1 ComputingService

| Entity | Inherits from | Description | | |
|---|----------------|--|--|---|
| ComputingService | Service | <p>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.</p> <p>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.</p> | | |
| Inherited Property | Type | Mult | Unit | Description |
| ID <i>[key]</i> | URI | 1 | | A <i>global unique ID</i> |
| 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 | | <u>The type of service according to a namespace-based classification (the namespace can be related to a middleware name, an organization or other concepts; org.glue and org.oqf are reserved).</u> |
| 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. |
| Property | Type | Mult | Unit | Description |
| TotalJobs | UInt32 | 0..1 | job | Number of total Grid jobs (sum of RunningJobs, WaitingJobs, StagingJobs, SuspendedJobs and PreLRMSWaitingJobs); this number 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 Grid jobs that are staging files in/out |
| SuspendedJobs | UInt32 | 0..1 | job | Number of Grid jobs which started their execution, but are suspended (e.g., for preemption) |
| PreLRMSWaitingJobs | UInt32 | 0..1 | job | Number of Grid 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 <i>[redefines Endpoint.ID]</i> | | * | A computing service exposes zero or more computing endpoints | |
| ComputingShare.LocalID <i>[redefines Share.LocalID]</i> | | * | A computing service offers zero or more computing shares | |
| ComputingManager.ID <i>[redefines Manager.ID]</i> | | * | 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 | |

Deleted: The type of service according to a middleware classification

Deleted: OtherInfo ... [6]

Deleted: numer

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 same 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

In this class, we have properties that can be used to publish summary information of jobs submitted via a certain endpoint. Such properties are optional and are not always measurable (e.g., in case of a stateless endpoint).

Formatted: Justified
Formatted: Normal, Justified

| 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 | 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 |

| | | | | |
|---|-----------------------|--------------|---|---|
| <i>ServingState</i> | <i>ServingState_t</i> | 1 | | A state specifying if the endpoint is accepting new requests and if it is serving the already accepted requests |
| <i>StartTime</i> | <i>DateTime_t</i> | 0..1 | | The timestamp for the start time of the endpoint |
| <i>IssuerCA</i> | <i>DN_t</i> | 0..1 | | Distinguished name of Certification Authority issuing the certificate for the endpoint |
| <i>TrustedCA</i> | <i>DN_t</i> | * | | <i><u>Distinguished name of the trusted Certification Authority (CA), i.e., certificates issued by the CA are accepted for the authentication process</u></i> |
| <i>DowntimeAnnounce</i> | <i>DateTime_t</i> | 0..1 | | The timestamp for the announcement of the next scheduled downtime |
| <i>DowntimeStart</i> | <i>DateTime_t</i> | 0..1 | | The starting timestamp of the next scheduled downtime |
| <i>DowntimeEnd</i> | <i>DateTime_t</i> | 0..1 | | The ending timestamp of the next scheduled downtime |
| <i>DowntimeInfo</i> | <i>String</i> | 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 |
| <u>TotalJobs</u> | <u>UInt32</u> | <u>0..1</u> | <u>job</u> | <u>Number of total Grid jobs (sum of RunningJobs, WaitingJobs, StagingJobs, SuspendedJobs and PreLRMSWaitingJobs); this number does not consider the local jobs</u> |
| <u>RunningJobs</u> | <u>UInt32</u> | <u>0..1</u> | <u>job</u> | <u>Number of running Grid jobs</u> |
| <u>WaitingJobs</u> | <u>UInt32</u> | <u>0..1</u> | <u>job</u> | <u>Number of Grid jobs waiting in the underlying computing managers (i.e., Local Resource Manager System or LRMS's)</u> |
| <u>StagingJobs</u> | <u>UInt32</u> | <u>0..1</u> | <u>job</u> | <u>Number of Grid jobs that are staging files in/out</u> |
| <u>SuspendedJobs</u> | <u>UInt32</u> | <u>0..1</u> | <u>job</u> | <u>Number of Grid jobs which started their execution, but are suspended (e.g., for preemption)</u> |
| <u>PreLRMSWaitingJobs</u> | <u>UInt32</u> | <u>0..1</u> | <u>job</u> | <u>Number of Grid jobs that are in the Grid layer waiting to be passed to the underlying computing manager (i.e., LRMS)</u> |
| 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 associated zero or more AccessPolicies | |

Formatted: Font: Italic

Deleted: Distinguished name of the trusted Certification Authority

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 |
| <i>LocalID</i> [key] | <i>LocalID_t</i> | 1 | | An opaque identifier local to the associated Service |
| <i>Name</i> | <i>String</i> | 0..1 | | Human-readable name |
| <i>Description</i> | <i>String</i> | 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 that can be granted to a single-slot job upon user request (unnormalized value) |
| MaxMultiSlotWallTime | UInt64 | 0..1 | s | The maximum obtainable wall clock time that can be granted to a multi-slot job upon user request; this value is measured from the start of the first slot up to the release of the last slot, (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, then 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 |

- Deleted:** per slot
- Deleted:** the
- Deleted:** MaxTotalWallTime
- Deleted:** total
- Deleted:** the
- Deleted:**
- Deleted:** this property is a limit for the sum of the wall clock time used in all the slots occupied by a multi-slot job
- Deleted:** a

| | | | | |
|-----------------------------|--------------------|------|--------|---|
| 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 |
| MaxMainMemory | UInt64 | 0..1 | MB | The maximum RAM that a job can use; if the limit is hit, then the LRMS could kill the job |
| GuaranteedMainMemory | UInt64 | 0..1 | MB | The guaranteed RAM that a job can use |
| MaxVirtualMemory | UInt64 | 0..1 | MB | The maximum RAM that a job can use; if the limit is hit, then the LRMS could kill the job |
| GuaranteedVirtualMemory | UInt64 | 0..1 | MB | The guaranteed virtual memory 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 | ExtendedBoolean_t | 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, LocalRunningJobs, WaitingJobs, LocalWaitingJobs, StagingJobs, SuspendedJobs and PreLRMSWaitingJobs); this number 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 |
| SuspendedJobs | UInt32 | 0..1 | job | Number of jobs which started their execution, but are suspended, e.g., for preemption (local and Grid) |
| LocalSuspendedJobs | UInt32 | 0..1 | job | Number of local jobs which started their execution, but are suspended (e.g., for preemption) |
| StagingJobs | UInt32 | 0..1 | job | Number of Grid jobs that are staging files in/out |
| PreLRMSWaitingJobs | UInt32 | 0..1 | job | Number of Grid 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 |

Deleted: MaxMemory ... [7]

Deleted: Boolean

Deleted: StagingJobs ... [8]

Deleted: (

Deleted:)

| | | | | |
|--|---------------------|--------------|--|---|
| 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

For this entity, we define the working area as an allocated storage extent that holds the home directories of the Grid jobs.

Formatted: Normal

| 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 | ExtendedBoolean_t | 0..1 | | True if the computing manager (i.e., LRMS) supports advance reservation | |
| BulkSubmission | ExtendedBoolean_t | 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 | |

Formatted Table

Deleted: Boolean

Deleted: Boolean

Formatted: Right, Right: 0,45 cm

| | | | | |
|--------------------------------|------------------------------|-----------------|---------------|---|
| | | | | 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 |
| Homogeneous | ExtendedBoolean_t | 0..1 | | True if the computing manager has only one type of execution environment |
| NetworkInfo | NetworkInfo_t | * | | Type of internal network available among the managed execution environment instances; if many values are published, then the various types of network may be available only within subsets of the execution environment instances; the execution environment properties SHOULD be checked |
| LogicalCPUDistribution | String | 0..1 | | Classification of the managed execution environment instances aggregated by number of logical CPUs. Syntax: X1:Y1, ..., Xn:Yn where I is the i-th group of execution environments with the same number of logical CPUs, Xi is the number of logical CPUs in each execution environment instance and Yi is the number of execution environment instances. |
| WorkingAreaShared | ExtendedBoolean_t | 0..1 | | True if the working area is shared across different execution environment instances (i.e., cluster nodes); this property applies to single-slot jobs |
| WorkingAreaTotal | UInt64 | 0..1 | GB | Total size of working area available to all the single-slot 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 single-slot 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 single- |

- Deleted: S
- Deleted: Homogeneity
- Deleted: Boolean
- Deleted: 0..1
- Deleted: all

| | | | | |
|------------------------|-----------------|------|----|--|
| | | | | slot 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 |
| WorkingAreaMPIShared | ExtendedBoolean | 0..1 | | True if the working area is shared across different execution environment instances (i.e., cluster nodes); this property applies to MPI jobs |
| WorkingAreaMPITotal | UInt64 | 0..1 | GB | Total size of working area available to all the MPI Grid jobs either as a shared area across all the execution environments (WorkingAreaMPIShared is true) or local to a certain execution environment (WorkingAreaMPIShared 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 |
| WorkingAreaMPIFree | UInt64 | 0..1 | GB | Free size of working area available to all MPI Grid jobs either as a shared area across all the execution environments (WorkingAreaMPIShared is true) or local to a certain execution environment (WorkingAreaMPIShared 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 |
| WorkingAreaMPILifeTime | UInt64 | 0..1 | s | Guaranteed lifetime of the MPI 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 subsequent 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) |

Deleted: Boolean

Deleted: A working area is an allocated storage extent that holds the home directories of the Grid jobs; t

Deleted: his property is t

Deleted: the

Deleted: con

| | | | | |
|----------------------------------|-------------------------|--------------|---|--|
| 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 POSIX accessible disk space with transient to permanent allocation to the users |
| 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 | |

Deleted: OtherInfo ... [9]
Formatted Table

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 | |

Formatted: Normal

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 environment 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 |

Deleted:
Deleted: r

| Property | Type | Mult. | Unit | Description |
|---|-------------------|--------------|--------------------|---|
| Platform | Platform_t | 1 | | The architecture platform of this execution environment |
| VirtualMachine | ExtendedBoolean ↓ | 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 | ExtendedBoolean ↓ | 1 | | Permission for direct inbound connectivity, even if limited |
| ConnectivityOut | ExtendedBoolean ↓ | 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 |

Deleted: Boolean

Deleted: Boolean

Deleted: Boolean

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 |
| AppName | String | 1 | | Name of the application environment |
| AppVersion | 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 |
| <i>ID [key]</i> | <i>URI</i> | <i>1</i> | | <i>A global unique ID</i> |
| 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.

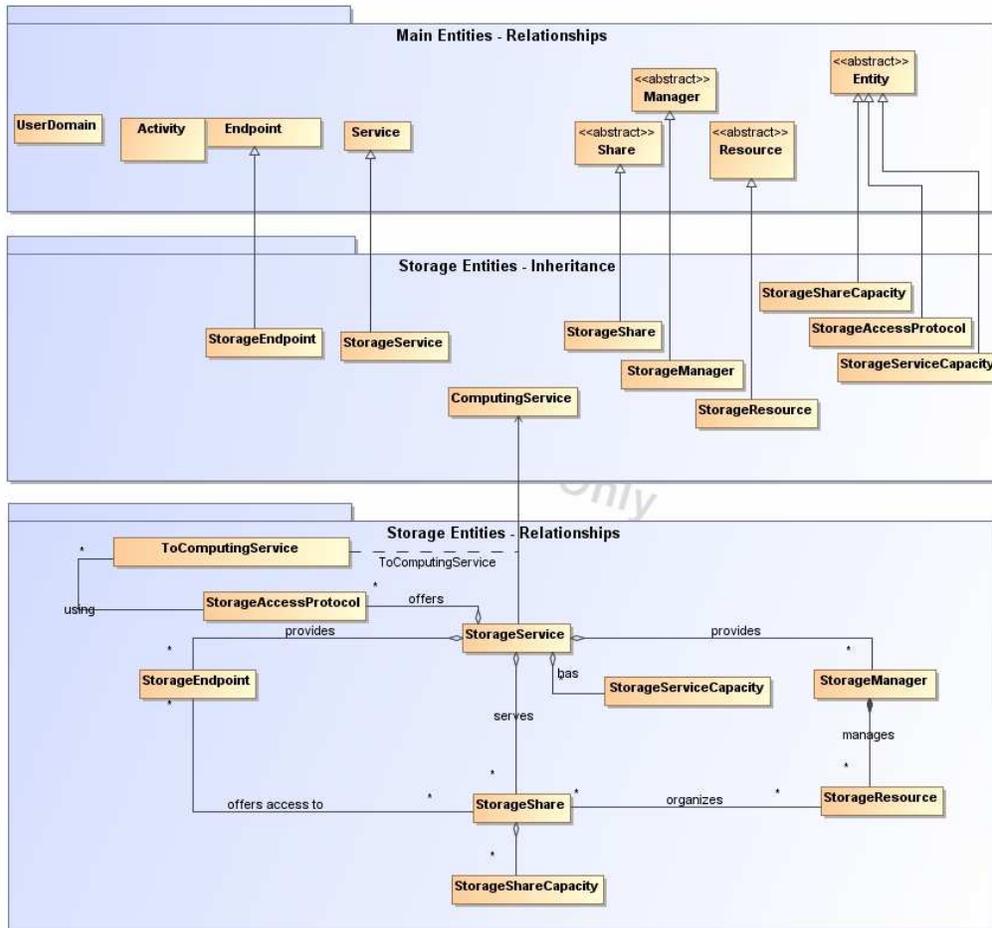


Figure 3 Entities and relationships for the Storage Service conceptual model

As explained in Section 6, we use the concept of storage extent to mean the capabilities and management of the various media that exist to store data and allow data retrieval.

Formatted: Normal, Left

7.1 StorageService

| Entity | Inherits from | Description | | |
|---|----------------|--|------|---|
| StorageService | Service | <p>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</p> <p>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.</p> | | |
| Inherited Property | Type | Mult | Unit | Description |
| ID | 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 namespace-based classification (the namespace can be related to a middleware name, an organization or other concepts; org.glue and org.ogf are reserved) |
| 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. |
| Property | Type | Mult | Unit | Description |
| No extra properties are defined in the specialized entity | | | | |
| Association End | | Mult. | | Description |
| StorageEndpoint.ID [redefines Endpoint.ID] | | * | | 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 |

Deleted: The type of service according to a middleware classification

Deleted: OtherInfo [10]

Deleted: Local

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 | |

Deleted:

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 |
| 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 | |

Deleted: OtherInfo ... [11]

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 |
| <i>ID</i> [key] | URI | 1 | | A global unique ID |
| <i>Name</i> | String | 0..1 | | Human-readable name |
| <i>URL</i> | URI | 1 | | Network location of the endpoint to contact the related service |
| <i>Capability</i> | Capability_t | 1..* | | The provided capability according to the OGSA architecture |
| <i>Technology</i> | EndpointTechnology_t | 0..1 | | Technology used to implement the endpoint |
| <i>Interface</i> | URI | 1 | | Identification of a type and version of the interface |
| <i>InterfaceExtension</i> | URI | * | | Identification of an extension to the interface |
| <i>WSDL</i> | URI | * | | URL of the WSDL document describing the offered interface (applies to Web Services endpoint) |
| <i>SupportedProfile</i> | 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 | * | | <i>Distinguished name of the trusted Certification Authority (CA), i.e., certificates issued by the CA are accepted for the authentication process.</i> |
| 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 storage 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 associated zero or more AccessPolicies | |

Formatted: Font: Italic

Deleted: Distinguished name of the trusted Certification Authority

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 |
| <i>Inherited Property</i> | <i>Type</i> | <i>Mult.</i> | <i>Unit</i> | <i>Description</i> | |
| 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 |
| 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 |

Deleted: OtherInfo ... [12]

A storage share represents a utilization target of one or more storage capacities 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 capacities. The status of each type of storage capacity as regards the usage by the user domains is described by the StorageShareCapacity.

Deleted: a
Deleted: extent
Deleted: extents
Deleted: extent

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 |

Deleted: n

The storage share capacity is useful to express the usage information of a 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 |
| 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 |

Deleted: OtherInfo ... [13]

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 storage resource refers to a category with summary information on the capacity | | |
| Inherited Property | | Type | Mult. | Unit |
| ID | [key] | URI | 1 | |
| Name | | String | 0..1 | |
| Property | | Type | Mult. | Unit |
| Type | | StorageResourceType_t | 1 | |
| Latency | | AccessLatency_t | 1 | |
| TotalSize | | UInt64 | 0..1 | GB |
| FreeSize | | UInt64 | 0..1 | GB |
| UsedSize | | UInt64 | 0..1 | GB |
| 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 |

Deleted: OtherInfo ... [14]

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 |
| LocalID | | LocalID_t | 1 | |
| NetworkInfo | | NetworkInfo_t | 0..1 | |
| Bandwidth | | UInt32 | 0..1 | Mb/s |
| Association End | | Mult. | 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 |

Deleted: OtherInfo ... [15]

8. Relationship to OGF Reference Model

In this section, we describe the integration of the GLUE information model with the OGF Reference Model [rm]. The reference model defines the concept of Grid Component. In GLUE, a root concept called Entity is defined. Such a root concept can be defined as a specialization of the GridComponent concept, that means that all properties are inherited by the GLUE classes. In Figure 4, we represent this relationship by a UML class diagram.

Formatted: Justified

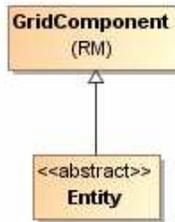


Figure 4 GLUE and Reference Model integration

Formatted: Justified

9. Security Considerations

This section considers security implications when using the GLUE 2.0 conceptual model. It follows the advice given in RFC-3552.

As the conceptual model of GLUE 2.0 provides limited scope for embedding security information many of these concerns listed here are delegated to the concrete data models and to the underlying software implementations. Nonetheless, some points are independent of which concrete data model is employed so some discussion is appropriate.

When deploying an information service conforming to the GLUE 2.0 conceptual model, consideration should be given to the points discussed below.

9.1 Communication security

The GLUE conceptual model is independent of how information is stored and how that information is exchanged between agents. Because of this, concern for communication security is largely delegated to the underlying concrete data model and software implementations.

Formatted: Heading 2, Justified
Formatted: Justified

9.1.1 Confidentiality

The GLUE conceptual model contains information that may be personal or confidential in nature. Contact details and indications of end-user activity may fall into this category.

Formatted: Heading 3, Justified
Formatted: Justified

Conforming implementations should identify which components of the data should be considered confidential and appropriate precautions should be in place to safeguard against disclosure to unintended audiences.

9.1.2 Data integrity

The information within GLUE has many potential uses, from operational to accounting. How accurate the information is may depend on many factors, including the integrity of software agents that publish data and the transport used to propagate information.

Formatted: Heading 3
Formatted: Justified

The software used to provide an information service may cache GLUE information. If so, the caches provide additional points where data integrity may be compromised.

9.1.3 Peer Entity authentication

No explicit description of the agents that publish information is included within the GLUE conceptual model. This prevents authentication information from being included within the abstract model.

Formatted: Heading 3
Formatted: Justified

In general, support for peer-entity authentication is delegated to the concrete data model or the underpinning software. In many cases the agents will act on behalf of some AdminDomain; if so, elements of peer entity authentication (e.g., public/private key-pairs) may be included using the described schema extension mechanisms provided issues with data integrity are understood.

9.2 Non-repudiation

The GLUE conceptual model contains no explicit description of the publishing agents that provide GLUE information. This prevents explicitly support for non-repudiation. In many cases a set of publishing agents will provide information for Services in some AdminDomain. If so, then it is the AdminDomain that asserts the non-repudiation of the data the publishing agents provide.

Formatted: Heading 2
Formatted: Justified

Non-repudiation may require information from whoever asserts the non-repudiation of the data; for example, a cryptographic certificate of some AdminDomain. If the publishing agent is identified with an AdminDomain then this information may be included using the schema extension mechanisms of the AdminDomain (via OtherInfo or Extension). It is also possible for this information to be included in fields specific to the concrete data model or it may be provided outside of the GLUE conceptual model.

In addition, information may be published with corresponding non-repudiation information, such as a cryptographic signature. Signatures may be included using schema extensions (OtherInfo or Extension) or may be included in fields specific to the concrete data model.

9.3 System security

← Formatted: Heading 2
← Formatted: Justified

The GLUE conceptual model intended use is to provide an abstract view of a grid system. There are many processes that may make use of this information, each may depend on the GLUE conceptual model to undertake work.

9.3.1 Unauthorized usage

← Formatted: Heading 3
← Formatted: Justified

The GLUE conceptual model has no explicit description of end-users of the schema information and no explicit description of authorized usage. In general, is assumed that any authorization controls for access to the GLUE information is provided by specific concrete bindings and software implementation.

It may be possible to identify a UserDomain with those agents authorised to use GLUE information and embed authorization information using described schema extension mechanisms, provided issues with data integrity are understood.

9.3.2 Inappropriate Usage

← Formatted: Heading 3
← Formatted: Justified

The GLUE conceptual model provides no mechanism for describing appropriate usage and does not include a data-processing model, so providing a description of inappropriate usage is considered out-of-scope.

Individual grids may describe what they consider appropriate usage of GLUE information and implement appropriate procedures to ensure this policy is enacted.

9.4 Specific attacks

← Formatted: Heading 2
← Formatted: Justified

RFC-3552 describes several specific attacks that must be considered. These are detailed below.

9.4.1 Eavesdropping

← Formatted: Heading 3
← Formatted: Justified

Some information described in the GLUE conceptual model may be sensitive in nature; this may include contact details and descriptions of user activity. Appropriate care should be taken to prevent unintended access or disclosure to an unintended audience.

9.4.2 Replay

← Formatted: Heading 3
← Formatted: Justified

Grid operations may depend on information provided in the GLUE conceptual model.

If a system implementing the GLUE 2.0 conceptual model is susceptible to a replay attack then it is possible for part (possibly all) of the information in the conceptual model to be reverted to some previous state as seen by some (possible all) end users. Please note that this is a specific case of the more general modification attack.

A replay attack may result in disrupted service. If security attributes, such as authorization, are embedded within the GLUE conceptual model then a replay attack may result in inappropriate access to data.

Underlying concrete models and software implementations should prevent replay attacks.

9.4.3 Message insertion

Formatted: Heading 3
Formatted: Justified

The ability to insert information is key to providing accurate information. However, inserting incorrect information may have a detrimental effect to the running systems; for example, there are attributes in the conceptual model that accept multiple values. If incorrect values are included, the systems may suffer.

Many aspects of GLUE provide service discovery. Inserting false information would allow unauthorised services to publish their presence and attract activity. This may be used as a basis for further attacks.

Underlying concrete models and software implementations should ensure that any agent's ability to insert information is limited and appropriate.

9.4.4 Deletion

Formatted: Heading 3
Formatted: Justified

The ability to delete information from an information service could interfere with normal operations; for example, if Services are removed then activity that would use those services may be affected; if AdminDomains are removed then normal operation procedures may be impossible; if security components are removed (such as X509 certificates) then facilities such as non-repudiation may become ineffectual.

Underlying concrete models and implementing software should ensure that any ability of an agent to delete information is limited and appropriate.

9.4.5 Modification

Formatted: Heading 3
Formatted: Justified

The ability for an agent to modify information stored in an information service is key to providing accurate information. However, concrete data models and software implementation should place limits such that the agents' ability to modify information is controlled and appropriate.

9.4.6 Man-in-the-middle

Formatted: Heading 3
Formatted: Justified

For a system implementing the GLUE conceptual model, a successful man-in-the-middle attack may lead to arbitrary modification of data (see 9.4.5). It may also allow deleting existing data (see 9.4.4) or adding additional data (see 9.4.3). This may have severe influence on the systems based on GLUE information.

Underlying concrete models and implementing software should understand the risk from man-in-the-middle attacks and provide appropriate security against them.

9.4.7 Denial of service attacks

Formatted: Heading 3
Formatted: Justified

A Denial of Service attack is one that attempts to prevent normal operation of systems. Perhaps, the most obvious is to prevent or corrupt the flow of information.

Systems using the GLUE conceptual model should understand the consequences of a partial or complete lack of information. Appropriate measures should be taken to ensure the systems continue to run to the extent possible.

10. Author Information

Sergio Androozzi, 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.

Deleted: ¶

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."¶

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-real] GLUE 2.0 – Reference Realizations to Concrete Data Models, <http://forge.ogf.org/sf/go/doc15221>,~~
 [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>

Formatted: Font: Not Italic

Formatted: Font: Not Italic

Formatted: Indent: Left: 0 cm, First line: 0 cm

Deleted: ¶

Formatted: Font: Not Italic

[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

[rm] **D. Snelling, P. Strong. Open Grid Forum Reference Model v2.0. OGF Draft. 23 Feb 2008.**
<https://forge.gridforum.org/sf/go/doc14766>

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 Extended booleans

The reserved value "undefined" SHOULD be used. The way to express that no value is published MUST be defined in the documents defining the realization to concrete data models (e.g., [glue-real]).

← - - - Formatted: Heading 2, Left

← - - - Formatted: nobreak, Left

← - - - Formatted: Left

16.4 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.

Formatted: Bullets and Numbering

16.4.1 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 information, 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

Formatted: Bullets and Numbering

16.4.2 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.

Formatted: Bullets and Numbering

16.4.3 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.

Formatted: Bullets and Numbering

16.4.4 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.

Formatted: Bullets and Numbering

16.4.5 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

Formatted: Bullets and Numbering

16.4.6 Email addresses

It must use an undefined FQDN for the domain.

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

Formatted: Bullets and Numbering

example@ggf.org

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
```

16.4.7 Uniform Resource Identifier (URI)

Formatted: Bullets and Numbering

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
```

16.4.8 X509 Distinguished Names

Formatted: Bullets and Numbering

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
```

16.4.9 Fully Qualified Attribute Name (FQAN)

Formatted: Bullets and Numbering

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 16.4.1). It must have no subgroup(s) or Capability specified.

Deleted: 16.3.1

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
```

16.4.10 Geographic locations

Formatted: Bullets and Numbering

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 a 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 ExtendedBoolean_t

Formatted: Heading 2

Closed enumeration

| Value | Description |
|-----------|------------------------------|
| false | boolean false |
| true | boolean true |
| undefined | the value cannot be measured |

17.2 LocalID_t

Formatted: Bullets and Numbering

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]

17.3 ContactType_t

Formatted: Bullets and Numbering

Open enumeration

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

Deleted: security ... [16]

17.4 PolicyScheme_t

Formatted: Bullets and Numbering

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]):

Deleted:

- 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

example@ggf.org

- o matches the user proving to have a certificate identified by this DN
- fqan:/vo_a
 - o matches all the users proving to be part of the vo_a
- fqan:/vo_a/group_a
 - o matches all the users proving to be part of group_a or one of its subgroups
- fqan:/vo_a/group_a/Role=prod
 - o matches all the users proving to be part of group_a and having the Role prod

17.5 DN_t

Formatted: Bullets and Numbering

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

17.6 Capability_t

Formatted: Bullets and Numbering

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

| Value | Description |
|---|--|
| data.access.flatfiles | capacity of providing access to a flat file |
| 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.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.management.transfer | capacity of managing a transfer of files from the start to the completion |
| 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.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) |
| executionmanagement.candidatesetgenerator | capacity of determining the set of resources on which a unit of work can execute |
| executionmanagement.dynamicvmdeploy | capacity of dynamically deploying a virtual machine image in a worker node |
| executionmanagement.executionandplanning | capacity of building schedules for jobs, that is, the capability of defining mappings between services and resources, possibly with time constraints |
| executionmanagement.jobdescription | capacity of letting users be able to describe a job submission request based on a machine-processable language |
| executionmanagement.jobexecution | capacity of executing a job or set of jobs. |
| executionmanagement.jobmanager | capacity of managing the execution of a job or set of jobs from start to finish |
| executionmanagement.reservation | capacity of managing reservation of resources for future usage |
| information.discovery | capacity of locating unknown resources or services, possibly satisfying a set of requirements |
| information.logging | capacity of recording data, often chronologically |
| information.model | capacity of modelling resources based on a community accepted definition |
| 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. |
| security.accounting | capacity of systematically recording, reporting, and analyzing the usage of resources |
| 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.authentication | capacity of providing authentication mechanisms for Grid users machine and services |
| security.authorization | capacity of handling authorization aspects, making authorization decisions about the subject and the requested mode of access based |

| | |
|-----------------------------------|--|
| | <u>upon combining information from a number of distinct sources</u> |
| <u>security.credentialstorage</u> | <u>capacity of providing an online credential repository that allows users to securely obtain credentials when and where needed</u> |
| <u>security.delegation</u> | <u>capacity for a user to give a service the authority to undertake specific activities or decisions on its behalf</u> |
| <u>security.identitymapping</u> | <u>capacity of mapping Grid-level credentials to local level credentials (e.g., mapping a user X.509 certificate into a UNIX account).</u> |

- Deleted:** security.authentication ... [17]
- Formatted:** English (U.S.)
- Formatted:** Bullets and Numbering
- Deleted:** the middleware name
- Deleted:** (e.g., for gLite services, org.glite.[service type])
- Deleted:** ¶

17.7 ServiceType_t

The RECOMMENDED syntax is in reverse-DNS style. The first element is a top-level domain, while the second element is a namespace, (the namespace can be related to a middleware name, an organization or other concepts; org.glue and org.ogf are reserved). Open enumeration.

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

- Deleted:** org.glite.wms ... [18]

17.8 QualityLevel_t

Closed enumeration

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

- Deleted:** development ... [19]

17.9 EndpointTechnology_t

Open enumeration.

| Value | Description |
|-------------------|--|
| <u>corba</u> | <u>The endpoint is implemented using CORBA technologies</u> |
| <u>indi</u> | <u>The endpoint is implemented using JNDI</u> |
| <u>legacy</u> | <u>The endpoint is implemented using legacy technologies</u> |
| <u>webservice</u> | <u>The endpoint is implemented as a Web Service</u> |

- Deleted:** webservice ... [20]

17.10 EndpointHealthState_t

Closed enumeration

| Value | Description |
|-----------------|---|
| <u>critical</u> | <u>It was possible to check the state of the endpoint and either it was not running or it was above some "critical" threshold</u> |

- Formatted:** Bullets and Numbering

| | |
|-------------------------|---|
| ok | It was possible to check the state of the endpoint and it appeared to be functioning properly |
| other | It was possible to check the state of the endpoint, but this is not covered by the defined states |
| unknown | It was not possible to check the state of the endpoint |
| 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 |

Deleted: ok ... [21]

17.11 [ServingState_t](#)

Formatted: Bullets and Numbering

Closed enumeration

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

Deleted: production ... [22]

17.12 [DateTime_t](#)

Formatted: Bullets and Numbering

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'

17.13 [Staging_t](#)

Formatted: Bullets and Numbering

Closed enumeration:

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

Deleted: none ... [23]

17.14 [JobDescription_t](#)

Formatted: Bullets and Numbering

Open enumeration:

| Value | Description |
|--------------------------------|---|
| condor | Condor |
| egee:idl | EGEE Job Description Language |
| globus:rsi | Globus RSL |
| nordugrid:xrsl | Nordugrid XSRL [XSRL] |
| ogf:jsdl:1.0 | Job Description Submission Language 1.0 |

Deleted: ogf:jsdl:1.0 ... [24]

17.15 [SchedulingPolicy_t](#)

Formatted: Bullets and Numbering

Open enumeration:

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

Deleted: fairshare ... [25]

17.16 ReservationPolicy_t

Closed enumeration:

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

Formatted: Bullets and Numbering

Deleted: none ... [26]

17.17 ComputingManagerType_t

Open enumeration:

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

Formatted: Bullets and Numbering

Deleted: lsf ... [27]

17.18 NetworkInfo_t

Open enumeration

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

Formatted: Bullets and Numbering

Deleted: 100megabite ... [28]

17.19 Benchmark_t

Open enumeration

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

Formatted: Bullets and Numbering

Deleted: specint2000 ... [29]

17.20 Platform_t

Open enumeration:

| Value | Description |
|-----------------------|--|
| amd64 | AMD 64bit architecture |
| i386 | Intel 386 architecture |

Formatted: Bullets and Numbering

| | |
|-------------------------|---|
| itanium | Intel 64-bit architecture |
| powerpc | PowerPC architecture |
| sparc | SPARC architecture |

Deleted: i386 ... [30]

17.21 CPUMultiplicity_t

Formatted: Bullets and Numbering

Closed enumeration:

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

Deleted: singlecpu-sin ... [31]

17.22 OSFamily_t

Formatted: Bullets and Numbering

Open enumeration:

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

Deleted: linux ... [32]

17.23 ParallelSupport_t

Formatted: Bullets and Numbering

Open enumeration:

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

Deleted: mpi ... [33]

17.24 AppEnvState_t

Formatted: Bullets and Numbering

Open enumeration:

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

Deleted: notinstallable ... [34]

17.25 ApplicationHandle_t

Formatted: Bullets and Numbering

Open enumeration:

| Value | Description |
|----------------------------|---|
| executable | Access based on running directly the main executable of the application (this |

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

Deleted: module ... [35]

17.26 OSName_t
Open enumeration:

Formatted: Bullets and Numbering

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

Deleted: scientificlinux ... [36]

17.27 License_t

Formatted: Bullets and Numbering

Closed enumeration:

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

Deleted: opensource ... [37]

17.28 ComputingActivityType_t

Formatted: Bullets and Numbering

Closed enumeration:

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

Deleted: single ... [38]

17.29 ComputingActivityState_t

Formatted: Bullets and Numbering

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

- namespace:state
- namespace:state:substate

Open enumeration:

| Value | Description |
|------------------------------|---|
| 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.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: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: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 |

Deleted: bes:pending ... [39]

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

17.30 StorageCapacity_t

Open enumeration:

Formatted: Bullets and Numbering

| Value | Description |
|----------|-------------|
| online | |
| nearline | |
| offline | |
| cache | |

17.31 StorageAccessProtocol_t

Open enumeration:

Formatted: Bullets and Numbering

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

Deleted: gsiftp ... [40]

17.32 AccessLatency_t

Closed enumeration:

Formatted: Bullets and Numbering

| Value | Description |
|--------------------------|--|
| 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 |
| online | Files are always on disk, hence cannot have their latency improved |

17.33 RetentionPolicy_t

Deleted: online ... [41]

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 |

Deleted: custodial ... [42]

17.34 ExpirationMode_t

Formatted: Bullets and Numbering

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 |
| releasewhenexpired | Support for files that have finite lifetimes and on expiration will be removed by the storage system |
| 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 |

Deleted: neverexpire ... [43]

17.35 StorageManagerType_t

Formatted: Bullets and Numbering

Open enumeration:

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

Deleted: castor ... [44]

17.36 StorageResourceType_t

Formatted: Bullets and Numbering

Open enumeration:

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

Deleted: disk ... [45]

| | | |
|--------|--|----|
| 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 |

Page 10: [2] Deleted **Sergio Andreozzi** **12/3/2008 1:00:00 PM**

| | | | | |
|-----------|--------|---|--|--|
| 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 |
|-----------|--------|---|--|--|

Page 10: [3] Deleted **Sergio Andreozzi** **12/3/2008 1:00:00 PM**

| | | | | |
|-----------|--------|---|--|--|
| 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 |
|-----------|--------|---|--|--|

Page 11: [4] Deleted **Sergio Andreozzi** **12/3/2008 1:00:00 PM**

| | | | | |
|-----------|--------|---|--|---|
| OtherInfo | String | * | | Placeholder to publish info that does not fit in any other attribute. Free-form string, comma-separated |
|-----------|--------|---|--|---|

| | | | | |
|--|--|--|--|---|
| | | | | <i>tags, (name, value) pair are all examples of valid syntax</i> |
|--|--|--|--|---|

| | | | | | |
|-----------------------------|--------|-------------------------|--|--|--|
| Page 12: [5] Deleted | | Sergio Andreozzi | | 12/3/2008 1:00:00 PM | |
| 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 | |

| | | | | | |
|-----------------------------|--------|-------------------------|--|--|--|
| Page 19: [6] Deleted | | Sergio Andreozzi | | 12/3/2008 1:00:00 PM | |
| 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 | |

| | | | | | |
|-----------------------------|--------|-------------------------|----|------------------------------------|--|
| Page 23: [7] Deleted | | Sergio Andreozzi | | 12/3/2008 4:48:00 PM | |
| MaxMemory | UInt64 | 0..1 | MB | The maximum RAM that a job can use | |

| | | | | | |
|-----------------------------|--------|-------------------------|-----|--|--|
| Page 23: [8] Deleted | | Sergio Andreozzi | | 12/9/2008 4:33:00 PM | |
| StagingJobs | UInt32 | 0..1 | job | Number of jobs that are staging files in/out | |

| | | | | | |
|-----------------------------|--------|-------------------------|--|--|--|
| Page 27: [9] Deleted | | Sergio Andreozzi | | 12/3/2008 1:01:00 PM | |
| 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 | |

| | | | | | |
|------------------------------|--------|-------------------------|--|--|--|
| Page 34: [10] Deleted | | Sergio Andreozzi | | 12/3/2008 1:01:00 PM | |
| 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 | |

| | | | | | |
|------------------------------|--------|-------------------------|--|--|--|
| Page 35: [11] Deleted | | Sergio Andreozzi | | 12/3/2008 1:01:00 PM | |
| 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 | |

| | | | | | |
|------------------------------|--------|-------------------------|--|--|--|
| Page 37: [12] Deleted | | Sergio Andreozzi | | 12/3/2008 1:01:00 PM | |
| 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 | |

| | | | | | |
|------------------------------|--------|-------------------------|--|--|--|
| Page 38: [13] Deleted | | Sergio Andreozzi | | 12/3/2008 1:02:00 PM | |
| 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 | |

| | | | | | |
|------------------------------|--------|-------------------------|--|--|--|
| Page 38: [14] Deleted | | Sergio Andreozzi | | 12/3/2008 1:02:00 PM | |
| 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 | |

| | | | | | |
|------------------------------|--------|-------------------------|--|--|--|
| Page 38: [15] Deleted | | Sergio Andreozzi | | 12/3/2008 1:02:00 PM | |
| 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 | |

| | | | | | |
|------------------------------|--|-------------------------|--|-----------------------------|--|
| Page 53: [16] Deleted | | Sergio Andreozzi | | 12/3/2008 3:47:00 PM | |
| 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 | | | | |

| | | | | | |
|------------------------------|--|-------------------------|--|-----------------------------|--|
| Page 55: [17] Deleted | | Sergio Andreozzi | | 12/3/2008 3:48:00 PM | |
| 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.identitymapping | 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 |

| | |
|--------------------------|--|
| 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 |

Page 55: [19] Deleted Sergio Andreozzi 12/3/2008 3:48:00 PM

| | |
|----------------|--|
| 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 |

Page 55: [20] Deleted Sergio Andreozzi 12/3/2008 3:48:00 PM

| | |
|------------|---|
| 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 |

Page 56: [21] Deleted Sergio Andreozzi 12/3/2008 3:48:00 PM

| | |
|----------|---|
| 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 |

Page 56: [22] Deleted Sergio Andreozzi 12/3/2008 3:49:00 PM

| | |
|------------|--|
| 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 |

Page 56: [23] Deleted Sergio Andreozzi 12/3/2008 3:49:00 PM

| | |
|--------------|---|
| 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 |

Page 56: [24] Deleted Sergio Andreozzi 12/3/2008 3:49:00 PM

| | |
|----------------|---|
| 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 |

Page 56: [25] Deleted Sergio Andreozzi 12/3/2008 3:49:00 PM

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

Page 57: [26] Deleted Sergio Andreozzi 12/3/2008 3:49:00 PM

| | |
|-----------|---|
| 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 |

Page 57: [27] Deleted Sergio Andreozzi 12/3/2008 3:49:00 PM

| | |
|---------------|--------------------------------|
| Isf | 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 |

| Page 57: [28] Deleted | | Sergio Andreozzi | 12/3/2008 3:49:00 PM |
|------------------------------|---|-------------------------|-----------------------------|
| 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 | | |

| Page 57: [29] Deleted | | Sergio Andreozzi | 12/3/2008 3:49:00 PM |
|------------------------------|--|-------------------------|-----------------------------|
| 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 | | |

| Page 58: [30] Deleted | | Sergio Andreozzi | 12/3/2008 3:49:00 PM |
|------------------------------|---------------------------|-------------------------|-----------------------------|
| i386 | Intel 386 architecture | | |
| amd64 | AMD 64bit architecture | | |
| itanium | Intel 64-bit architecture | | |
| powerpc | PowerPC architecture | | |
| sparc | SPARC architecture | | |

| Page 58: [31] Deleted | | Sergio Andreozzi | 12/3/2008 3:49:00 PM |
|------------------------------|---|-------------------------|-----------------------------|
| 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 | | |

| Page 58: [32] Deleted | | Sergio Andreozzi | 12/3/2008 3:50:00 PM |
|------------------------------|---|-------------------------|-----------------------------|
| 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 | | |

| Page 58: [33] Deleted | | Sergio Andreozzi | 12/3/2008 3:50:00 PM |
|------------------------------|--|-------------------------|-----------------------------|
| mpi | Parallel execution based on mpi library | | |
| openmp | Parallel execution based on openmp library | | |
| none | No supported parallel execution | | |

| Page 58: [34] Deleted | | Sergio Andreozzi | 12/3/2008 3:50:00 PM |
|------------------------------|--|-------------------------|-----------------------------|
| 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 | | |

| Page 59: [35] Deleted | | Sergio Andreozzi | 12/3/2008 3:50:00 PM |
|------------------------------|--|-------------------------|-----------------------------|
| 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) | | |

| Page 59: [36] Deleted | | Sergio Andreozzi | 12/3/2008 3:50:00 PM |
|------------------------------|-----------------------|-------------------------|-----------------------------|
| 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 |

Page 59: [37] Deleted Sergio Andreozzi 12/3/2008 3:50:00 PM

| | |
|------------|--|
| 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 |

Page 59: [38] Deleted Sergio Andreozzi 12/3/2008 3:51:00 PM

| | |
|-------------------|--|
| 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 |

Page 60: [39] Deleted Sergio Andreozzi 12/3/2008 3:51:00 PM

| | |
|----------------|---|
| 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 |

Page 60: [40] Deleted Sergio Andreozzi 12/3/2008 3:51:00 PM

| | |
|----------|---|
| gsiftp | FTP with GSI authentication |
| file | POSIX access |
| nfs | Network File System protocol |
| afs | Andrew File System protocol |
| rftio | Remote File Input/Output protocol |
| gsirftio | 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 |

Page 61: [41] Deleted Sergio Andreozzi 12/3/2008 3:51:00 PM

| | |
|----------|--|
| 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 |

Page 61: [42] Deleted Sergio Andreozzi 12/3/2008 3:51:00 PM

| | |
|-----------|-------------------------|
| 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 |

Page 61: [43] Deleted

Sergio Androozzi

12/3/2008 3:51:00 PM

| | |
|--------------------|--|
| 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 |

Page 61: [44] Deleted

Sergio Androozzi

12/3/2008 3:51:00 PM

| | |
|---------|--|
| 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 |

Page 61: [45] Deleted

Sergio Androozzi

12/3/2008 3:52:00 PM

| | |
|---------|--|
| 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 |