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

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

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

April 23, 2009

GLUE v. 2.0.1 – Reference Realizations to LDAP Schema

Status of This Document

This document provides information to the Grid community regarding the LDAP Schema realization of the GLUE information model (v.2.0). Distribution is unlimited. This implementation is derived from the specification document "GLUE Specification v. 2.0.1", April 23, 2009. This document is a draft.

Abstract
The GLUE specification is an information model for Grid entities described in natural language enriched with a graphical representation using UML Class Diagrams. This document presents a realization of this information model as XML Schema, LDAP Schema and SQL Schema.

Contents

1. Introduction

The GLUE 2.0 Information model defined in [glue-2] is a conceptual model of Grid entities. In order to be adopted by Grid middlewares, a realization in terms of a concrete data model is needed.

This document provides the normative realization of the GLUE 2.0 conceptual model in terms of an LDAP Schema. The approach followed to map the entities and relationships in the conceptual model to the concrete data model are also described.

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. LDAP Schema Realization

3.1 Approach

There are many approaches to realize the GLUE conceptual model as an LDAP Schema. The GLUE LDAP rendering maps an entity in the GLUE information model to a specific LDAP entry which utilizes a single objectclass. As a result there is one to one correspondence between GLUE LDAP entries and GLUE entities.

3.2 OID Assignments

The Glue LDAP rendering utilizes the sub tree of 1.3.6.1.4.1.6757 which is assigned to the Global Grid Forum. An overview of the main use of the sub tree is given in tables 1, 2 and 3 representing the main entities, computing service entities and storage service entities respectively.

Take into account that OID numbers include the concrete chapter number in which the entity for that OID is reference in the GLUE 2.0 Specifications. (i.e. Entity is described in chapter 5.1, thus its OID is 1.3.6.1.4.1.6757.100.1.1.5.1).

Main Entities	
OID	**Entity**
1.3.6.1.4.1.6757.100.1.1.5.1	Entity <<abstract>>
1.3.6.1.4.1.6757.100.1.1.5.2	Extension
1.3.6.1.4.1.6757.100.1.1.5.3	Location
1.3.6.1.4.1.6757.100.1.1.5.4	Contact
1.3.6.1.4.1.6757.100.1.1.5.5	Domain <<abstract>>
1.3.6.1.4.1.6757.100.1.1.5.5.1	AdminDomain
1.3.6.1.4.1.6757.100.1.1.5.5.2	UserDomain
1.3.6.1.4.1.6757.100.1.1.5.6	Service
1.3.6.1.4.1.6757.100.1.1.5.7	Endpoint
1.3.6.1.4.1.6757.100.1.1.5.8	Share <<abstract>>
1.3.6.1.4.1.6757.100.1.1.5.9	Manager <<abstract>>
1.3.6.1.4.1.6757.100.1.1.5.10	Resource <<abstract>>
1.3.6.1.4.1.6757.100.1.1.5.11	Activity
1.3.6.1.4.1.6757.100.1.1.5.12	Policy <<abstract>>
1.3.6.1.4.1.6757.100.1.1.5.12.1	AccessPolicy
1.3.6.1.4.1.6757.100.1.1.5.12.2	MappingPolicy

Table 1: Main Entities

Abstract classes are not implemented as objects in the LDAP Schema, but they have an OID reserved because attribute belonging to that class are referenced to the OID subtree of its class. (i.e. Entity.CreationTime has OID 1.3.6.1.4.1.6757.100.1.1.5.1.1).

Computing Service	
OID	**Entity**
1.3.6.1.4.1.6757.100.1.1.6.1	ComputingService
1.3.6.1.4.1.6757.100.1.1.6.2	ComputingEndpoint
1.3.6.1.4.1.6757.100.1.1.6.3	ComputingShare
1.3.6.1.4.1.6757.100.1.1.6.4	ComputingManager
1.3.6.1.4.1.6757.100.1.1.6.5	Benchmark
1.3.6.1.4.1.6757.100.1.1.6.6	ExecutionEnvironment
1.3.6.1.4.1.6757.100.1.1.6.7	ApplicationEnvironment
1.3.6.1.4.1.6757.100.1.1.6.8	ApplicationHandle
1.3.6.1.4.1.6757.100.1.1.6.9	ComputingActivity
1.3.6.1.4.1.6757.100.1.1.6.10	ToStorageService

Table 2: Computing Service

Storage Service	
OID	**Entity**
1.3.6.1.4.1.6757.100.1.1.7.1	StorageService
1.3.6.1.4.1.6757.100.1.1.7.2	StorageServiceCapacity
1.3.6.1.4.1.6757.100.1.1.7.3	StorageAccessProtocol
1.3.6.1.4.1.6757.100.1.1.7.4	StorageEndpoint
1.3.6.1.4.1.6757.100.1.1.7.5	StorageShare
1.3.6.1.4.1.6757.100.1.1.7.6	StorageShareCapacity
1.3.6.1.4.1.6757.100.1.1.7.7	StorageManager
1.3.6.1.4.1.6757.100.1.1.7.8	DataStore
1.3.6.1.4.1.6757.100.1.1.7.9	ToComputingService

Table 3: Storage Service

3.3 Directory Information Tree

The LDAP DN is constructed following the hierarchical relationships that exist between entities in the Glue information model. The resulting Directory Information Tree –DIT– is shown in figure 1 and SHOULD be used as a guide when constructing the DN. The top of the DN is the object o=grid. Below this there MAY be any number of domain entities which represent the hierarchical nature of the domains in the grid environment. For example a computing center C, participating in a national grid infrastructure N, which is part of a wider international infrastructure Z SHOULD constructed the following DN.

dn: EntityId=C, EntityId=N, EntityId=Z, o=grid

Where C, N and Z are either 'AdminDomain' or 'UserDomain' objects.

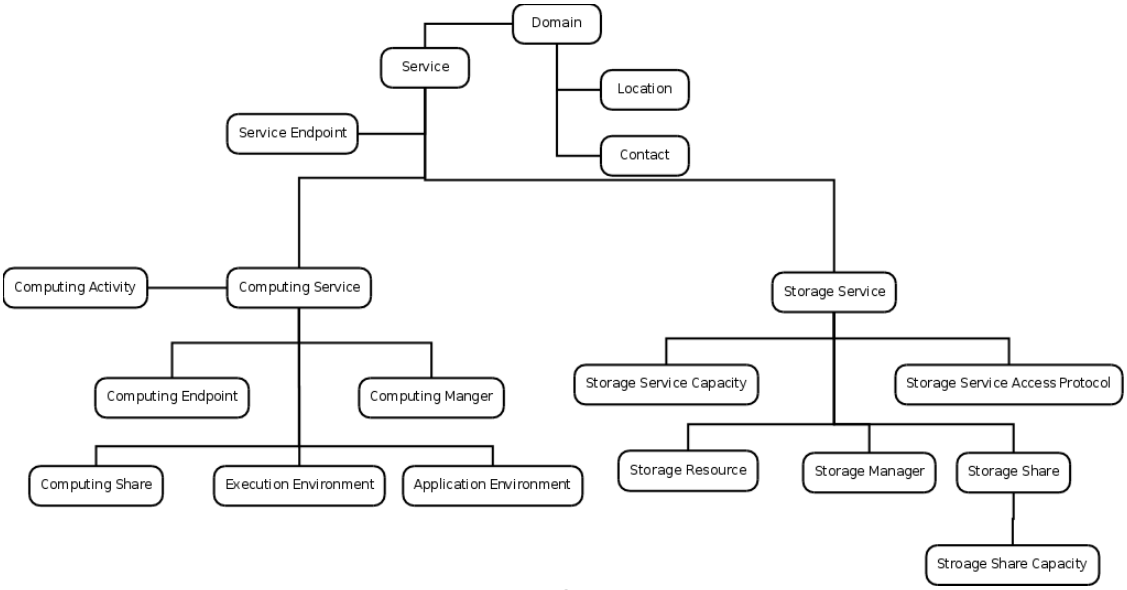

Figure 1. The Glue LDAP DIT

3.4 Data types

For the implementation of the different data types, we have decided to just use two different types of the standard LDAP v3 attributes set referred in [rfc2252].

- **IA5 String**, with OID 1.3.6.1.4.1.1466.115.121.1.26
- **Integer**, with OID 1.3.6.1.4.1.1466.115.121.1.27

“Integer” will be used for types UInt32 and UInt64 of the original GLUE 2.0 Specification and “IA5 String” will be used for every other type.

This also means that data type integrity will not be held in the LDAP implementation itself, but must be ensured by other means.

3.5 Relationships

LDAP is not a relational database[1] [2] [3], but a directory. Thus, LDAP does neither provides nor ensures relationships.

To implement relationships in LDAP we have decided that in each entity A that has a relationship with entity B, a new attribute will be created in both entity A, referencing entity B, and in entity B, referencing entity A.

This mechanism is similar to the one used in relational databases with foreign keys, except for several facts:

- In a relational database, when implementing a one to many and a one to one relationship, you just implement the foreign key in one of the sides, while here we implement it in both.

[1] LDAP is not relational: http://mysqldump.azundris.com/archives/74-LDAP-is-not-relational.html
[2] LDAP is not a relational database: http://www.oreillynet.com/sysadmin/blog/2006/05/ldap_is_not_a_database.html
[3] LDAP vs RDBMS: http://www.openldap.org/doc/admin24/intro.html#LDAP%20vs%20RDBMS

- In a relational database, when implementing a many to many relationship, you must create a new table that holds all relations due to the fact that it can not hold multivalued attributes. In LDAP we support multivalued attributes so we don't need any intermediate table.
- Relation databases ensure relationship integrity, LDAP does not.

4. Security Considerations

Using LDAP to implement GLUE 2.0 Specifications raises several considerations specially in the field of data integrity.

LDAP is not a relational database, thus it can not ensure relationship integrity. This must be ensure by other means.

LDAP can not ensure all data types referred in the GLUE 2.0 Specifications, thus this implementations uses the generic ones "IA5 String" and "Integer" specified in [rfc2252].

5. Author Information

Sergio Andreozzi, INFN

Stephen Burke, RAL

Felix Ehm, CERN

Laurence Field, CERN

Gerson Galang, ARCS

David Horat, CERN

Balazs Konya, Lund University

Maarten Litmaath, CERN

Paul Millar, DESY

JP Navarro, ANL

6. Contributors & Acknowledgements

We gratefully acknowledge the contributions made to this document.

7. Intellectual Property Statement

8. 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.

9. Full Copyright Notice

10. 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-2] GLUE 2.0 Specification, *https://forge.gridforum.org/sf/go/doc146239*
[ogf-ns] Standardised Namespaces for XML infosets in OGF,
http://www.ogf.org/documents/GFD.84.pdf
[xsd-oe] XForms 1.0. Open Enumeration.
http://www.w3.org/TR/2002/WD-xforms-20020118/slice6.html#model-using-openenum
[xsd-ap] Advanced XML Schema Patterns for Databinding Version 1.0,
http://www.w3.org/TR/xmlschema-patterns-advanced/#group-Unions
[rfc2252] Lightweight Directory Access Protocol (v3): Attribute Syntax Definitions,
http://www.ietf.org/rfc/rfc2252.txt