GLUE2 LDAP DRAFTs: 2010 vs 2012

Florido Paganelli, Lund University

OGF Teleconf, 22nd October, 2013

Outline

- History
- Quick answers to Stephen's questions
- DRAFT2010 vs DRAFT2012
- Distributed LDAP Hierarchy, ARC + gLite/BDII integration

History

- History seems always different depending on who talks about it...
- LDAP rendering defined in 2009/10
  - gLite- specific
  - No contribution from ARC
- ARC was the first to have his own GLUE2 LDAP tree realization
- EGEE implementation was different from 2009/10 draft
- EMI (2010-2013) invested large effort in GLUE2 LDAP implementation

All existing implementations are much different than 2009/10!

History
LDAP Realization Document

- In May 2012, with an update in July, we tried to syncronize the document with existing implementations:
  - **gLite/BDII**
  - **ARC**
- The document is the result of discussion between ARC and gLite/BDII developers.
- Outcome:
  - http://redmine.ogf.org/dmsf_files/125?download=
  - The above has issues identified and track changes, and was the foundation to EMI implementation.

Stephen's questions

Q: Who is the editor of the document?

A: By revising the 2010 document we thought we took over the editorship. We are willing to finish this work.

Q: "Freeze old (2010) version and start a new one?"

A:There is a revised version already. Why is that not taken into account?

Q: "Remove all references to the tree?"

A: This needs complex discussion. ARC believes that without DIT interoperability server-side is impossible. More about it later.

Q: Separate document for the BDII architecture?

A: No need for separate document for BDII, but definite DIT modelling is needed.

Q: Who is the document for?

A: Technology implementors. It was even specified in 2010.

DRAFT2010 vs DRAFT2012

Image from openclipart.org

STRUCTURAL vs AUXILIARY

In: DRAFT2010 Affects: LDAP Schema

- All classes deriving from Entity will be of type "Structural".

- All other classes will be of type "Auxiliary".

- STRUCTURAL: entries that define **branching nodes** of the LDAP DIT
- AUXILIARY: suitable for entries that are leafs of a LDAP DIT
- Problem: objects like GLUE2ComputingService **cannot be used as a branching point** in a tree

STRUCTURAL vs AUXILIARY a bit more detailed...

In: DRAFT2010 Affects: LDAP Schema

- *All classes deriving from Entity will be of type "Structural".*
- *All other classes will be of type "Auxiliary".*

RFC4512 says:

- STRUCTURAL:
  *[...] - DIT structure rules only refer to structural object classes; the structural object class of an entry is used to specify the **position** of the entry in the DIT; [...]*
- AUXILIARY:
  *[...] commonly used to **augment** the sets of attributes required and allowed to be present in an entry.*
  [...]
- Problems:
  - *position* is a defined RDN inside the DN. Objects like GLUE2ComputingService have no own attribute in the RDN, so they **cannot be used as part of the DN, (i.e. positioned)**
  - ComputingService is a **different object than a Service** (remember the discussion in the mailing list?) not just an augmented Service. Hence it shouldn't be AUXILIARY.

STRUCTURAL vs AUXILIARY

In: DRAFT2012 Affects: LDAP Schema

- EMI GLUE2 schema was corrected by changing most of the object classes to STRUCTURAL
  - Production LDAP servers are already running with such schema
- Pros:
  - Removes the branching limitation
  - Transparent for clients

▶ Section 3.4 needs to be changed

DirectoryString attribute type

In: DRAFT2010

- DirectoryString type was used for all the strings by the document
- Problem: Schema implementation **didn't follow** that,
  it used **IA5String** (see comments B3, B4 in the draft)

DirectoryString

In: DRAFT2012
Affects: LDAP Schema and DIT

- DirectoryString is the selected string type
- EMI Schema implementation **follows the DRAFT2010 choice**
- Known issue(?): type prevents from publishing **empty** attributes.
  ▶ Should be mentioned in a revision of the document.

OID Assignments

In: DRAFT2010

Problems:

- The proposed OID numbering system was **not extensible** (comment B13):
  - x.x.5.5.1-6: Domain attributes
    - Adding a new domain attribute breaks the system, it should be x.x.5.5.7, but...
  - x.x.5.5.7: AdminDomain ObjectClass
  - MISTAKE: considering attributes and object classes OIDs at the same “level”
- Entity OID numbers are coupled to OGF.147 section numbers (!)
  - Adding new entities or sections to the model document breaks the system
- Inheritance rules are not applied consistently, but only for Policy and Domain. (comment B12)

OID Assignments

In: DRAFT2012

- EMI Schema implementation redefines all the OIDs
  - Introduces a simple extensible schema, different OIDs “levels” for object classes and attibutes
  - Domain OID: x.x.1.1.5
    - Domain attributes OID: x.x.1.1.5.1-3
    - Adding a new attribute would just be x.x.1.1.5.4 and so on.
  - AdminDomain OID: x.x.1.1.6
- Deployed in production
  - Transparent for queries, doesn't affect clients
- TODO: sync the draft with the implementation.

Taxonomy consistency

Draft 2009	Draft 2012	Brief explanation
LDAP implementation	LDAP Realization	
Grid Middleware	technology providers	Where it applies
GLUE2 abstract schema	GLUE2 abstract model	
object pair in the abstract schema	Entity attribute in the abstract model	Taxonomy not in sync with GFD.147
LDAP object	LDAP Object Class	Sync with LDAP RFCs taxonomy, consistent use in the document
LDAP object	LDAP entry	Sync with LDAP RFCs taxonomy, consistent use in the document, to define a complete entry in a LDIF file. (i.e. multiple object classes can be grouped in a entry)
GLUE1	Glue1	
URL to GFD incorrect	URL to GFD updated	Due to publication of GFD
Unclear definition of model to LDAP rendering	Clarified in section 3.4	
Missing Boolean datatype in 3.5	Added boolean datatype in section 3.5	
inconsistent or missing typography for different items	Consistent typography for different items (i.e. bold for LDAP object classes, and so on)	

DIT - DRAFT2010

- Contained a description of a **gLite-oriented** tree
- **No** real server **implementation** followed this tree, as of today that is completely obsolete.
- Lousy description of DIT in Section 3.7 lead to **very different implementations**
- Section 3.7 gives no explanation of existing hierarchy of services in production systems. In short, **it does not discuss aggregation at all**.
- **GLUE2GroupID is not an ID** (see comment B7)

Section 3.7 needed a major rewrite.

DIT - DRAFT2012

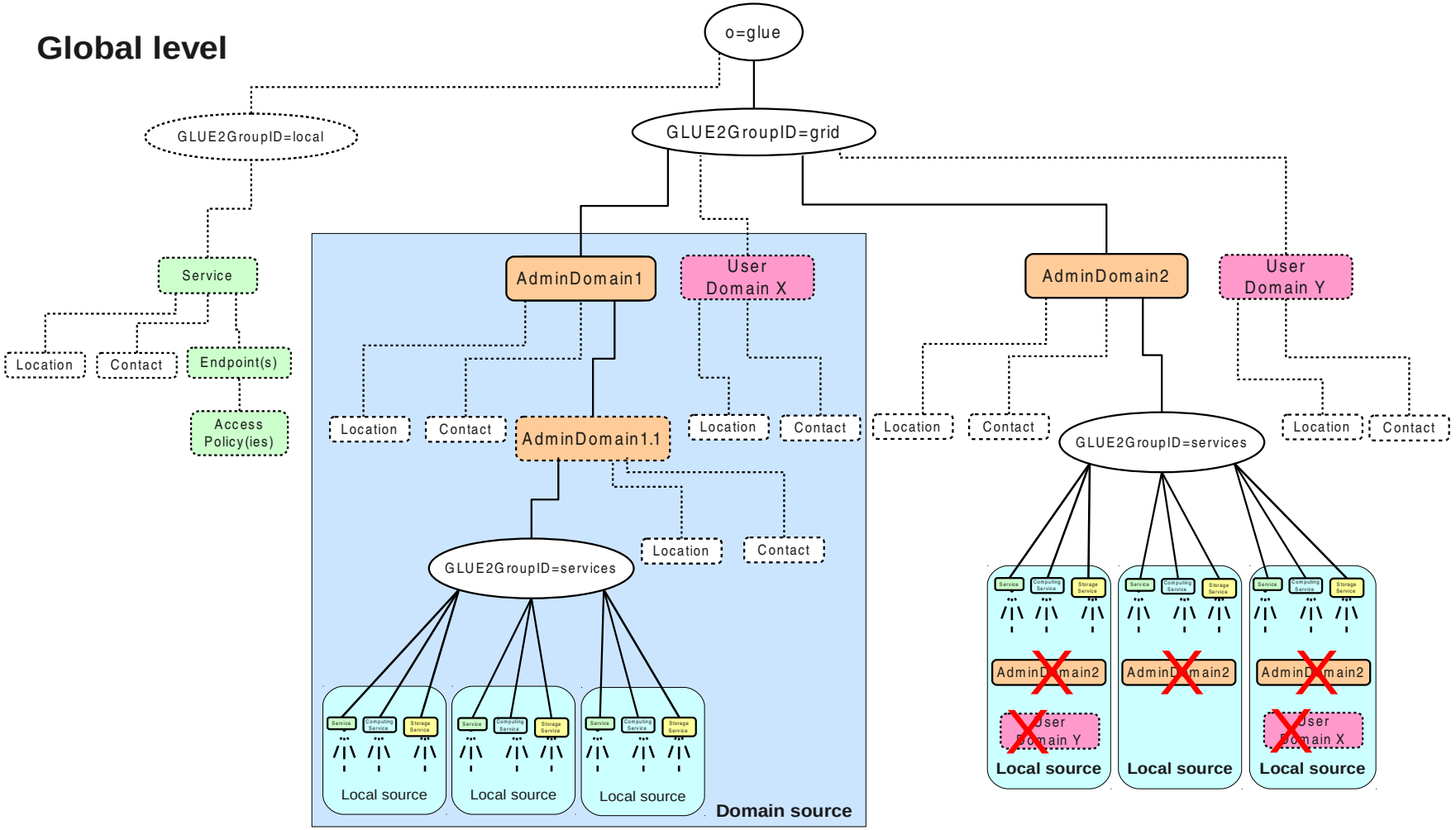
- Comes with a **completely rewritten section 3.7 (DIT)**
- Accurately defines the **mapping** between **model entities** and **LDAP entries**
- Explains how the DIT is constructed in terms of RDNs and DNs
- Presents a **minimal** set of restrictions on the structure of a GLUE2 LDAP tree
  (i.e. root of the tree, how to group services, where to place extensions, relationship between AdminDomain and Service objects...)
- Corresponds to status of existing **deployments**
- Introduces consistent treatment of grouping elements.
  - Grouping is not defined in GFD.147 and MUST be explained in this document.

- The new section 3.7 can be reviewed and moved to an appendix, but not in another document.

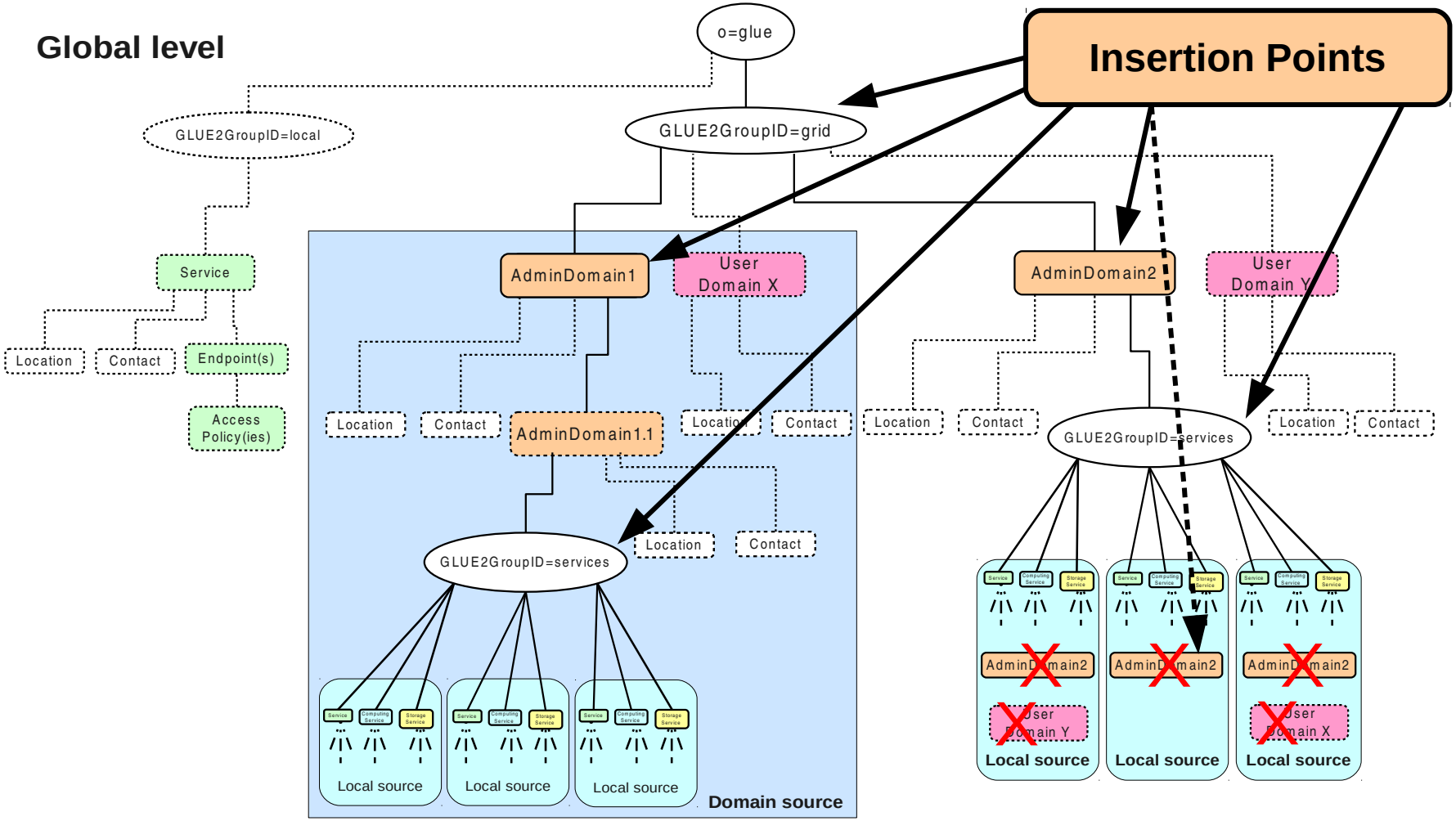
Realization Document: ARC's view

- Take the July 2012 document as a basis for further work
- Have a dedicated meeting where the changes are taken one by one and approved
  - **Should** reflect current implementation status and DIT structure agreement: the so called "*insertion points*"
- DIT issue: keep trees structures at least as **examples** of existing implementations.
- "Resource" BDII is a **central concept for ARC** and **NOT *"implementation detail"***
- Site-BDIIs is unnecessary complication for ARC.

Distributed LDAP Hierarchy, ARC + gLite/BDII integration



Distributed LDAP Hierarchy, ARC + gLite/BDII integration



NORDUGRID
Grid Solution for Wide Area Computing and Data Handling

Thanks!