**Review of Document:**

EGI PROFILE FOR THE USE OF THE GLUE 2.0 INFORMATION SCHEMA, v0.5  
 <https://documents.egi.eu/public/ShowDocument?docid=1324> [EGIGLUE2]

**Author:** Florido Paganelli

NorduGrid Collaboration

ARC Middleware

**email:** florido.paganelli@hep.lu.se

**Date:** 2012-09-13

Table of Contents

[1. Introduction and background 1](#__RefHeading__105_713472725)

[2. Review and Technical notes on the document 2](#__RefHeading__108_713472725)

[2.1 General remarks 2](#__RefHeading__110_713472725)

[2.2 GFD.147 and the EGI profile 3](#__RefHeading__112_713472725)

[2.3 Other general remarks. 4](#__RefHeading__116_713472725)

[3. Comments with reference to pages and sections 7](#__RefHeading__118_713472725)

[4. Remarks on consequences on Operations 11](#__RefHeading__120_713472725)

# 1. Introduction and background

This review has been done by NorduGrid ARC middleware[ARC] information system developers. The focus is mainly on implementation details and functional needs for ARC to be nicely integrated in the EGI infrastructure by leveraging GLUE2.

This document contains general remarks and detailed per-section comments about the choices.

This is considered a first iteration of the document review.

In fact, a detailed evaluation of the semantics of the GLUE2 attributes defined in the reviewed document, and their impact on the current ARC implementation of GLUE2 is yet to be done. Another document will be produced to discuss such details.

ARC uses a GLUE2 LDAP rendering and LDAP schema based on a proposed OGF draft[G2LDAP], and a GLUE2 XML rendering based on a proposed XSD schema[G2XML].

ARC GLUE2 LDAP implementation was written having in mind the experience with Glue1; it tries to adhere to a hierarchical information system structure inspired by the gLite BDII information system, but tries to preserve the fundamental ideas of ARC information system.

This hierarchical structure was never described in any document so far, but nowadays these are contained in a draft GLUE2 LDAP realization document within the OGF GLUE2 group[G2LDAP]. The latest review of this draft is currently pending for review from other members of the GLUE2 Working Group.

gLite BDII follows a hierarchical three level model, where information is pulled from a top-level system (top-bdii) from an intermediate level (site-bdii) which in turn pulls information from a local or resource level (resource-bdii).

ARC information system follows a hierarchical model where information is pushed from the local or resource level up to an index level (ARC EGIIS). Therefore, the local levels are the key place where to gather information. The index level just holds pointers to the local levels.

The GLUE2 LDAP implementation tries to follows this approach: a local/resource level provides both resource level information and site level information. This is done leveraging the GLUE2 concept of Domain. This information can then be pulled by top-level BDII, that should eventually filter unwanted information.

An working implementation of this approach is being tested within the EMI project and has been so far successful, and it involved close collaboration within the ARC and BDII developers team. Work is ongoing to test this new approach and preserve backward compatibility with the former strict three-level hierarchical approach of top-level BDII.[EMIARCBDII]

The above architecture was presented[ARISGLUE2] during the last EGI Community Forum in Munich, 26-30 March 2012, and has been currently submitted as a proceeding.

The structure of this document is as follows:

Section 2 gives a detailed review of the document. General notes are followed by per-section comments.

Section 3 gives some comments on the operational point of view.

In the following sections, 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>).

# 2. Review and Technical notes on the document

## 2.1 General remarks

The reviewed document details quite well the requirements needed for the correct publishing of information for each GLUE2 attribute.

Unfortunately, it completely lacks an overview of what the current information systems look like (those described in chapter 5), how GLUE2 information is used in these systems, and what kind of GLUE2 entities MUST be published for each of the information system products to work.

SB: The profile document is not intended to provide a description of the EGI information system, currently based on the BDII, or its future evolution. The majority of the profile relates to GLUE 2 attributes in general and is not specific to the BDII technology, but it is possible that the document will need to be revised to some extent if the technology changes.

ARC developers believe that is crucial to show how information is aggregated on different levels (resource, site, top) in the BDII system, how this is related to GOCDB, and how it would eventually be used by EMIR.

The document seems to silently assume that BDII and gLite are the EGI grid. We found strange that there is no reference on how ARC and UNICORE middlewares are supposed to join EGI information system, especially taking into account that some strategies have been presented in information system workshops held during EGI meetings.

To summarize, there is no overall view of how a GLUE2 compliant information system should work.

It's completely unclear what entities should be published at the resource level and which entities at the top level. ARC information system relies on a definite GLUE2 resource-level structure that might not be suitable to be fed as-is to a top level.

SB: The profile specifies the content of the information system as a whole, which currently means a top-level BDII. It does not prescribe how that information is collected and aggregated from the individual sites and services.

## 2.2 GFD.147 and the EGI profile

Several open choices present in the GLUE2 GFD.147[GDF147] are not resolved in the rendering documents, and are not appointed by this profile either.

ARC developers think that the EGI Profile is a good opportunity to define these choices in a very precise way, for the profile to be effective. A positive outcome might be forwarded to the GLUE2 Workgroup as a recommendation.

These are:

**URIs must be better defined**

There is still no agreement on how URIs should look like. These are used in IDs and several other attributes and for each attribute description on how to craft these needs extreme care. This is needed for the sake of interoperability.

As a general recommendation, NorduGrid suggests to follow RFC2141 and define namespaces not simply bound to EGI, to allow future services to use GLUE2.

An example is in the discussion on *AdminDomain*, on page 8 in this document.

**IDs format**

In GFD.147, IDs are URI. Once a decision about what a URI is has been taken, there must be a canonical way of building these IDs in a correct and unified way across EGI middlewares. The document seems not to care about this detail, and it even suggests NOT to use URI

SB: I agree that it would be desirable for the GLUE Working Group to define a format for IDs which guarantees uniqueness, and I’ll add a comment that if such a format is defined it should be used.

in the case of AdminDomain IDs, thus breaking GLUE2 standard.  
NorduGrid opinion is that enforcing a standard means follow its recommendation, and not twisting them to some existing situation.

SB: IDs for AdminDomains and UserDomains (site and VO names) are defined by existing practice over many years and their use is widespread in existing middleware and operational tools. It is not practical to change that.

The most important thing for *ID*s is their **persistency**, as they are basically keys of a distributed database. Persistency is poorly defined in page 10 as a “reasonable choice”.

SB: I don’t see an easy way to give concrete guidelines for persistency; the requirements are defined by their use in external services rather than from the information system itself, However, our experience with GLUE 1 is that this has not led to any particular problems in practice.

ARC developers think that is EGI crucial task to define the operational aspects of such persistency, i.e. duration of a *Service* *ID* and conditions to which the *ID* is subject to change in a very detailed way.

**Open enumerations**

GFD.147 leaves space for the creation of open enumerations. However, some of these enumerations universally identify aspects of the resources described by the GLUE2 model as created by every middleware developer.

Examples are *Service Type*s, *Endpoint InterfaceName*s and *Endpoint Capabilit*ies.

The document incorrectly states at page 19 that "there are currently no defined uses" for the Capability attribute. This is currently not true, as ARC clients rely on these Capabilities to discover services and endpoints. A more accurate survey by EGI should be enforced among the key middlewares before writing down such statements. The coming EMI Execution Service[EMIES] interface makes extensive use of these capabilities to describe features of its unique *Endpoint*s.

SB: I will remove that comment, and simply say that the correct Capabilities should be published according to the schema.

## 2.3 Other general remarks.

**Clearly define which entities are mandatory to publish and in which context.**

It is very important to define which GLUE2 entities needs to be published. A detailed overview is missing all across the document, and recommendations are given here and there in a difficult to track manner. A specific section regarding this would be preferred.

SB: I will attempt to add more information, but I don’t think it’s possible to be comprehensive because it depends greatly on the specific circumstances. The main purpose of the profile is to define the content of GLUE 2 entities if they are published.

It is as well important to know in which context these entities needs to be published. For example, a detailed information on what to publish on each bdii level (resource, site or top), what to put in GOCDB, what EMIR should contain.

SB: The profile applies to the content of the information system, which at present means a top-level BDII, and I will say that more explicitly. It does not define the content of the GOC DB, which is managed separately, and it does not directly define the content of the site and resource BDIIs as they are implementation details. EMIR is not currently deployed in the production Grid; as the information system evolves it may be necessary to update the profile accordingly.

Without the above, mandatory nature of some attributes can be misinterpreted as “the entity whose this Mandatory attribute belongs MUST be published”.

This is the case for *Extension* attributes on page 12, or the *Contact.Detail* attribute on page 14 are mandatory attributes. Does this mean that *Extension* and *Contact* entities must be published at all times?

The general comment should be that Mandatory attributes MUST be published IF AND ONLY IF their entity MUST be published.

SB: I think the correct statement is that Mandatory attributes MUST be published if their entity is in fact published, and I will say that more explicitly.

**Semantic is somewhat gLite/BDII centric**

Most of the semantic across the document is very gLite/BDII-centric. For example, the author takes for granted the concept of site. To ARC developers knowledge, there is no existing document that explains and defines such a concept not even in BDII documentation. If documents exist, please refer to them.

SB: The site is an operational concept in EGI, for example all resources are registered in the GOC DB as belonging to a site and GGUS tickets can be assigned to a site. The information system reflects that usage, it doesn’t define it.

NorduGrid ARC does not have a “site” concept. ARC only has a resource concept, and the closest thing to a site concept is the LDAP index distinguished name. In the ARC GLUE2 rendering, we agreed that publication of *AdminDomain* information at the local/resource level can behave in a similar way to the Glue1 site concept.

We suggest the following: the assumption of "Site equals AdminDomain" at page 14 must be better defined. First an introduction of the Site concept is needed, and then a clear definition of what an AdminDomain has to do with it and how the concept of distributed domain adheres to it has to be explained.

Another example of gLite-centrism is on page 52. When speaking of BDII clients, details of the environment variable set to use top-level BDII is explained. It is probably referring to gLite clients. ARC clients do not use anything like that. Please define what kind of clients is the author speaking about. A non-grid audience may assume that all clients behave like that by design.

SB: All clients which query top-level BDIIs *should* behave like that since it is the currently-defined method to obtain the information.

The last paragraph of section 5.1 seems again to ignore that resource level in ARC is documented and published, and can be accessed directly by any consumer. This is a key concept in ARC.

SB: As above, the profile describes the EGI information system which is currently provided by the top-level BDII.

ARC wants this information to be aggregated by top-level BDII, hence it is important that these differences are stated in such document. Moreover, ARC will probably need top-level BDII to filter such information in the way top-level bdii finds it suitable, and not the other way around, as already stated before.

For top-bdii to be the primary information system for EGI, as stated in page 51, it is not fair for it to mandate a structure at the resource level.

SB: It does not mandate any structure at the resource level, that’s up to the implementation.

EGI profile SHOULD NOT impose undesired attributes not to be published on the local level. Presence of undesired attributes at a top-level might need to be further discussed.

That's why we think the tree structure used by pure gLite components needs to be discussed and explained in the context of an information system that federates different middlewares. We think that the top-level should be smart enough to leave all ARC functionality intact without imposing functional modifications of interoperation.

Relevant comments about this subject are in the notes later in this document: about the *Member* attribute on page 8, about *ApplicationEnvironment*s on page 10, about *ComputingActivities* on page 10.

In general, it is not clear how other middlewares are using GLUE2 top-level BDII. How will UNICORE benefit of BDII within the EGI infrastructure?

**Better definition of which attribute is static or dynamic.**

It is not clear from the classification on page 5-6 what is the expected dynamic nature of entities for EGI. There is a need for a clear distinction between what is supposed to be a static, dynamic or semi-dynamic attribute. In the latter cases, a precise definition of the lifetime of an attribute must be given.

SB: In general I think the lifetime of attributes is inherent in their definition, it’s defined by the rate at which the underlying property changes. I will explain that, and add an explicit comment for attributes which may change very frequently like the number of running jobs.

To this extent, the persistency example on page 10 of the EGI profile is rather strange: why should a StorageService hold an ID until there is files? From an operations point of view, that is the same service even if there is no data served. Might be more interesting to know if an ID

change is triggered by the addition of a new component to the StorageService, or by its relocation in some other facility....

SB: The point is that files stored in a StorageService may be catalogued using a reference to the ID, hence if the ID changes the reference will fail.

NorduGrid proposes to define an additional classification, specifically targeting attributes with a dynamic nature, plus a detailed explanation of their time to live.

**Need for examples.**

This profile would benefit from examples all across the description. Especially when complex situation arise, i.e. how does EGI devise the publication of a distributed *AdminDomain*? How should the record(s) look like? Where is the shared information (i.e. Location) retrieved?

SB: I think this would be enormously difficult to do in a way that would cover all possibilities. I would suggest that specific cases are discussed as they arise, documented in the EGI wiki and added to the profile document at a future revision.

**Implicit tree/hierarchical structure needs to be explicit.**

The document seems to implicitly mandate a tree structure, probably implied by the BDII three-level structure and its relationship with GOCDB.

SB: The document should not be BDII-specific. I will check for such references, and either remove them or make it clear that they relate to that technology only.

However, this model is never explicitly explained and the rules are poorly described here and there, from time to time, with controversial statements. Examples are as follows:

Page 14, for domains: "In general the DN structure in the BDII will make it impossible to publish any service information for a site if the *AdminDomain* is missing". This is implicitly telling that there is a structure and the *AdminDomain* MUST be published. But then, what about the local/resource level? Is it mandatory to publish it there? What happens to *Distributed* domains?

What is the DN structure in this context? where is it defined? At least give some pointers.

Then later in the document, in section 5.1, it says that "clients should not rely on the structure of the LDAP Directory Information Tree below the root DN"

What is this LDAP directory structure? how is the root DN related to other DNs? all these concepts are not explained, not even in some reference document.

SB: 5.1 is an appendix with a brief description of the BDII, and it has a reference to the BDII documentation.

**Validation**

Validation of values is very ambitious and very little will be implemented by technology and software providers along the lines of this profile. Most of the checks in place nowadays are now related to data types and not semantics.

NorduGrid thinks that if EGI needs such a semantic validation, it must produce such a validator itself.

SB: A validator is being produced. The intention is to integrate it into the EGI operational test framework.

# 3. Comments with reference to pages and sections

*Page 9, Section 2.1.3 ID*

GFD.147 does not define any special scheme for IDs, the model just says they have to be URIs.

A proper format for IDs needs to be decided in a formal and acknowledged way, at least across partners who want to be compliant with the EGI profile. The reviewer's proposal would be to adopt RFC2141 for the scheme and define a set of namespaces suitable for the different elements. UUIDs might be part of the ID.

SB: See the response above.

About this subject, it would be nice to clarify with some examples of what the author means by "ID MUST contain a string which is interpretable as a DNS name ....". A DNS name itself is not a URI, cause at least a scheme should be defined. To some extent, it is suggested to have a read to http://www.w3.org/TR/uri-clarification/ about this matter.

SB: I will add an example.

*Page 10, Section 2.1.4 Name*

In some cases it will be nice to make this field mandatory, for example for the *AdminDomain* record. See further for discussion.

SB: The Name is just a human-readable description of the object so I don’t see any need for it to be mandatory, since the content by definition can’t be relied on.

*Page 11, Section 2.1, Entity attributes table*

Some attributes marked as Mandatory must appear in all the objects.

SB: Apart from the ID only the profile name and version are marked as mandatory, and that is only the case if it is desired to assert compliance with the profile. Given that objects may be generated in many different ways I don’t see a general way to infer compliance from one object to another. However, I will add something to allow a Service object to assert compliance for every dependent object.

It's probably too much unnecessary information. For example, this information will be replicated in every *ComputingShare, ExecutionEnvironment, Benchmark* and every *Endpoint*. Currently ARC CE has 11 *Endpoint*s.

Publishing it in only in *AdminDomain* or *Service* records SHOULD be enough.

*Page 12, Section 2.3 Location*

The description of Location object is restricted to its association with Domains, but this is incomplete.

According to GFD.147, also Services can have their own Location, and the model seems not to put any constraint on the associations between a Service's Location.ID and an AdminDomain's Location.ID.

SB: That’s true. I would regard it as an unusual case for a Service to have an associated Location, but I will add some text to cover it.

These two Location records could be different.

This makes sense also thinking of a Domain as an authority, located somewhere, and a Service as an actual piece of software running in some different place.

*Page 12, Section 2.3.2 Place*

Is there any mandatory value in the case of a Distributed domain?

Any recommendation, i.e. OtherInfo contents?

SB: I have no particular view – it’s inherently difficult to define the location of a distributed domain.

*Page 14, Section 2.6 AdminDomain*

The definition of ID given here breaks GLUE2 URI recommendation for no good reason, in the reviewer's opinion. The reviewer thinks that the NAME field should be MANDATORY and carry the AdminDomain human-readable name. Information providers MUST produce a URI-compliant ID based on the Name field, in the form of some recommended URI scheme (see previous comments on section 2.1.3).

Example:

AdminDomain.Name = FI\_HIP\_T2

AdminDomain.ID = urn:AdminDomain:FI\_HIP\_T2

The above would still preserve human readability "at a glance" in a LDAP tree, and is compliant to RFC2141.

SB: As stated above, site names defined in the GOC DB are well-established and embedded in many of our operational tools and practices, so in this case GLUE usage must follow that. Uniqueness is ensured by the requirement to register the name in the GOC DB – these are indeed unique keys and not just names.

*Page 14, Section 2.6.2*

Should distributed domains enforce some kind of AdminDomain hierarchy when publishing?

Example: a distributed AdminDomain MUST have at least one

child AdminDomain, that is, a "local" domain joining the distributed one?

SB: At present I don’t see a need to require such a hierarchy, but it is possible, for example GRIF does it like that.

*Page 12, Page 14 (Detail attribute)*

Are these mandatory attributes to be added if and only if their entity is added? Or entities with mandatory attributes must always be published?

SB: The former. I’ll add a general statement about that.

*Page 16, Section 2.7 UserDomain*

The same observations about AdminDomain Name and ID apply for UserDomain.

SB: Again, VO names are a special case.

*Page 17,* *Section 2.7.4 Member*

ARC used to publish DNs in the NorduGrid schema. We would like to use *Member* to reflect this behaviour until a better solution that leverages better *Polici*es and *UserDomain* is found. ARC agrees that is bad to publicly disclose certificates' Distinguished Names.

However, the *Member* attribute is marked as undesirable. ARC thinks that the author did not want to publish such an attribute at the top-level. If so, then the top-level should secure itself from publishing such information, and client SHOULD NOT expect such information at the top level, but they MAY find it at the resource level.

SB: As above, the profile only specifies the content at the top level.

*Page 18, 2.9 Endpoint*

Wrong usage of MUST, SHOULD is more appropriate. In fact if a *Service* has an *Endpoint*, it MUST be known, otherwise clients run the risk of having unexpected surprises while searching for *Endpoint*s. There can be *Service*s without *Endpoint*s, and it that case publication is impossible. SHOULD fits both cases.

SB: Paul Millar made the same point, and I agree. I’ve rewritten it.

*Page 19, Section 2.9.2 Capability*

ARC developers requirement is that the *Capability* attribute MUST be published, and therefore MUST be MANDATORY instead of RECOMMENDED.

SB: OK.

The GLUE2 group SHOULD provide a clear list of these *Capabilities* and existing open enumerations with their descriptions. If the GLUE2 group is not able to do it, EGI MUST have a list of *Capabilities* of *Service*s running within its partners, that SHOULD be

part of the profile or MUST be referenced somewhere on the net as an authoritative source. *Service*s MAY publish different capabilities, but in that case EGI clients are not supposed to be able to parse those.

Clients MUST be able to rely on *Capability* information to perform

discovery and understand the operations allowed on a target endpoint.

This is of more importance since *ComputingServices* will have

*ComputingEndpoints* with different capabilities

(as stated on Page 27, Section 3.3), hence the *Capability*

field becomes crucial for discovery.

*Page 19, Section 2.9.4 InterfaceName*

The GLUE2 group SHOULD provide a clear list of these *InterfaceName*s and existing open enumerations with their descriptions. If the GLUE2 group is not able to do it, EGI MUST have a list of *InterfaceName*s of *Service*s running among its partners, that SHOULD be part of the profile or MUST be referenced somewhere on the net as an

authoritative source. *Service*s MAY publish other *InterfaceName*s,

but in that case EGI clients are not supposed to be able to parse those.

Clients MUST be able to rely on *InterfaceName* information to perform

discovery and understand the nature of the endpoint.

SB: I’ll add an explicit reference to the GLUE WG respository.

*Page 25, Section 2.11 Policy, AccessPolicy, MappingPolicy*

The second paragraph on page 25 says that "In EGI there SHOULD be a list of VO names to which the Rules relate".

Since section 2.7 says that a UserDomain is basically a VO, maybe this list of names should actually be enriched by a list of UserDomain.IDs instead of VO names.

SB: This may not be phrased very well, but the point is the other way round: the Policy objects have references to DomainIDs, and in the EGI context that means to VO names.

*Page 29, Section 3.4 ComputingShare*

Incorrect statement:

"The class ComputingShare represents a group of jobs which are scheduled

in a uniform way by an LRMS"

The *ComputingShare* class does not group anything, and for sure does

not group jobs, but summarizes resources that are available for job

execution on that specific share.

It identifies utilization targets for the jobs to be executed, eventually in the form of LRMS queues, but not limited to that.

SB: I don’t agree: the ComputingShare does represent a group of jobs, for example the TotalJobs attribute counts the jobs in that group which are currently known to the LRMS. However I can perhaps phrase it better.

*Page 31, Section 3.4.12 MaxSlotsPerJob*

As for today, the concept of Slots per Job is not clear, especially the way this relates to cores and nodes on an underlying LRMS. The reviewer understands the need for having this attribute as mandatory, but warns that this information will be interpreted by clients in an ambiguous way.

SB: There are working groups looking at, for example, parallel and whole-node scheduling, and it may be that they will produce further recommendations in this area.

*Page 36, Section 3.5.3 OtherInfo*

How is the *CPUScalingReference* relates to the *Benchmark* record? Would

it be better to use GLUE2 *Benchmark* instead of this *OtherInfo* field?

SB: This is an existing GLUE 1 attribute which will simply be reflected into the GLUE 2 publication. If GLUE 2 usage becomes the default it may be that this will be revised – in any case the attribute is marked as Optional.

*Page 41, Section 3.8 ApplicationEnvironment*

As a suggestion, *ApplicationEnvironment*s might be grouped to allow faster queries, or to allow clients to strip out all these records if not needed.

This, however, will not reduce data size on a global information level.

The reviewer thinks that a feature in the GLUE2 Model (publishing detailed *ApplicationEnvironment* information) should not be dropped due to limitations of some information system.

The data problem should be overcame in some other way, preserving the richness of GLUE2 information.

As stated in general remarks, If any data issue arises within the top-level, then the top-level itself should cope with that. ARC would like to keep its local level tidy and consistent with the GLUE2 model.

SB: We have to deal with the practical limitations of the system we have, so we will have to keep this under review. The LHC experiments are moving to distribute their software using cvmfs, in which case they may stop publishing the large number of names currently in the system.

*Page 43, Section 3.10 ComputingActivity*

The assertion that *ComputingActivities* SHOULD NOT be inserted in the information system is not precise, or at least gLite-centric.  
ARC reads it this way: *ComputingActivities* records will NOT be aggregated by the   
top-level BDII information system software.  
EGI profile cannot prevent such information to be published by clusters at a resource level. This would disrupt ARC functionality. See general comments about this issue.

SB: As discussed above, the profile applies to the content of the top BDII only.

*Page 51, Section 5.1 BDII*

The site concept is not clear for ARC, specify that top-level BDII MUST be able to read information also from ARC resource level.

This section is gLite-centric and is completely missing EGI views on ARC and UNICORE integration via GLUE2.

Structure of the resource level should be recommended and documented.

SB: This is only intended to be a brief summary as an appendix, with a pointer to external documentation with the details. The way the information is collected by a top BDII is up to the implementation, and it’s entirely possible for gLite, ARC and UNICORE services to have different methods.

*Page 52, Section 5.1 BDII*

The document states that "Clients should not rely on the structure of the LDAP directory Information Tree below the root DN, but should follow relationships between objects using their ID and ForeignKey attributes."

ARC knows that this is currently not possible, and when possible, not reliable.

SB: I disagree: ten years of experience with GLUE 1 has told us that it’s both possible and desirable. This was one of the basic design principles for the LDAP rendering used in the BDII, and I regard it as essential. The discussion on the XML rendering has reached essentially the same conclusion, for much the same reasons.

This knowledge has been acquired while developing ARC to top-level BDII integration within the EMI project[EMIARCBDII].

More than that, the proposed approach is hard to implement. Consider the following remarks:

1) It is bad design to use an LDAP database (hierarchical by design) and then rely on clients themselves to preserve relational data integrity on a distributed database (which is rather associative than purely relational).

SB: GLUE 2 is not a hierarchy, it’s a complex network, and hence cannot be mapped to a tree. In addition, LDAP requires every non-leaf object in the tree to exist, whereas in many cases objects in GLUE 2 are not guaranteed to exist.  
This is even present in the document. For example, in Section 1.2 Validation, about validation tools:

"All relations between objects should be published as implied by the schema and the LDAP rendering."

This sound to the reviewer as a need to mandate some minimal LDAP DIT.

SB: No, this is a reference to the ForeignKey attributes in the rendering,

Later in the same section " ... although full validation of

referential integrity is likely to be a difficult task."

If referential integrity is a difficult task, why should we burden clients?

SB: Clients don’t need to validate the references, just follow them.

There is no benefit in demanding clients or consumers preserving that. A robust information system should preserve such integrity by some other means. The only way we currently have is to preserve important associations by enforcing the DIT of the LDAP tree.

SB: Again, GLUE 2 is not a tree, and it is therefore not possible to enforce the schema relations via the DIT.

2) There is no clear specification of what an ID is and how should be defined, hence no real assumption of uniqueness and persistency can be made.

SB: The ID is a fundamental concept in the schema, and must be generated and used correctly whatever the technology.

3) Not all current systems publish ForeignKeys in an appropriate manner.

SB: That’s a matter for validation: any bugs must be fixed.

# 4. Remarks on consequences on Operations

The consequences of this schema on operations are merely connected to what kind of information should be gathered, and for what concerns NorduGrid, how this can be done by ARC and its adopters.

The focus in mostly on the OtherInfo fields that have been introduced to be MANDATORY or RECOMMENDED by the profile. ARC rendering is already compliant to most of the recommendations, but a detailed comparison of the semantics defined in the EGI Profile against actual values contained in ARC GLUE2 rendering is yet to be done, as said, it will be part of another document.

Currently there are only two remarks:

1) Regarding NDGF, it is possible to imagine it as a Distributed *AdminDomain*. With respect

to this, there must be a common agreement on how to fill Location information as in Page 12, Section 2.3. However, since Distributed AdmindDomains are poorly defined by the EGI profile, there is still no idea on how to achieve this.

SB: In general I would say that NDGF can choose any representation which suits its needs. If necessary NDGF should discuss the use of the information with clients like gstat and the Real Time Monitor which display locations on a map.

2) on Page 36, Section 3.5.3 OtherInfo: Value Share=<VO>:<share> this percentage value is marked as recommended, and is responsibility of the system administrator to set that value manually.

ARC does not publish anything like that yet, but we can make the string configurable by some means. I don't understand how is it possible to calculate that though, and how relevant it can be.

SB: Again this is a reflection of an existing GLUE 1 attribute into GLUE 2.In the past, WLCG mandated publication for its MOU resources, for EGI it’s useful. It reflects the target share for each VO – the actual outcome will depend on how the VOs submit jobs.

Bibliography

[ARC]: NorduGrid Collaboration, NorduGrid Advanced Resource Connector middleware, , http://www.nordugrid.org/arc/

[ARISGLUE2]: Florido Paganelli, Balazs Konya, The ARC InformationSystem - Overview of a GLUE2 compliant production system, 2012, https://indico.egi.eu/indico/contributionDisplay.py?sessionId=3&contribId=138&confId=679

[EGIGLUE2]: Stephen Burke, EGI profile for the use of the GLUE2.0 Information schema, 2012, https://documents.egi.eu/public/ShowDocument?docid=1324

[EMIARCBDII]: EMI project, , 2012, https://savannah.cern.ch/task/?func=detailitem&item\_id=31619

[EMIES]: Bernd Schuller, EMI Execution Service Specification, 2011, https://twiki.cern.ch/twiki/pub/EMI/EmiExecutionService/EMI-ES-Specification\_v1.14-1.odt

[G2LDAP]: Stephen Burke et al, GLUE v. 2.0 – Reference Realization to LDAP Schema, , http://redmine.ogf.org/dmsf\_files/125?download=

[G2XML]: Sergio Andreozzi et al., GLUE v. 2.0 – Reference Realization to XML Schema, , http://redmine.ogf.org/dmsf\_files/122?download=

[GDF147]: Sergio Andreozzi et al., GLUE Specification v. 2.0, 2003, http://www.ggf.org/documents/GFD.147.pdf