Integrating glue 2.0 in gLite WMS

S.Monforte (INFN-CT)

JRA1 All Hands Meeting (Prague)

5-7 November 2008

www.eu-egee.org

glue schema 2: first impressions

- **The new glue 2.0 schema definition despite the initial complexity is generally well conceived**
  - conceptual models of both computing and storage service
    - based on simple specialized entities
    - connected by essential and well formed relations
  - describing meaningful entities

- **Henceforth in the presentation we will focus only on the computing models**
  - the same considerations and implementation design apply at MM level for the storage model

glue schema 2.0 and MM

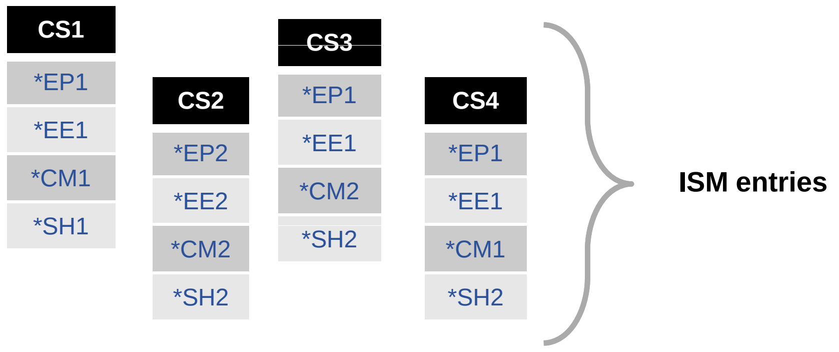
- **The computing entities required for MM purposes are basically**
  - ComputingService
  - ComputingEndPoint
    - AccessPolicy
  - ComputingShare
    - MappingPolicy
  - ExecutionEnvironment
    - ApplicationEnvironment
  - ComputingManager
- **The *ComputingService* aggregates all the other entities forming a connected set**
  - provides the relations between entities describing how a given *endpoints* may expose a certain *environment* handled by the relevant *manager* via a particular *share*

glue2.0 and MM: memory model

- **glue2.0 information once acquired are converted in ClassAd**
  - each sensible entity is represented by its own classad
  - stored in the WMS as a sort of flyweight design pattern
    - *reducing the memory image of the WMS*
- ***The ISM will be a simple collection of pointers to the relevant data***
  - *each item stores pointers to the aggregated entities*
    - *representing a ComputingService entity*
- ***glue2.0 provides different authorization policies***
  - *AccessPolicy*
    - *ComputingEndpoint*
  - *MappingPolicy*
    - *ComputingShare*
- ***Both Access and Mapping policies should be matched against user credentials***

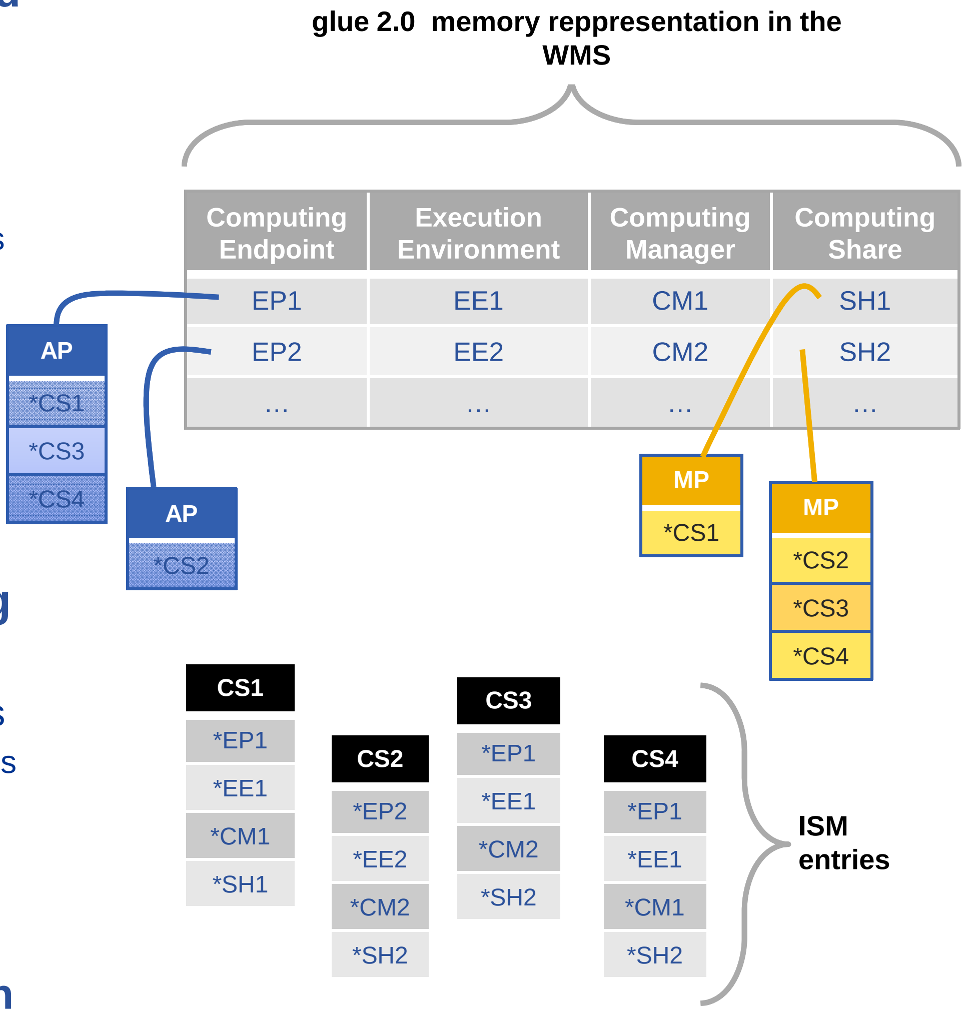
glue 2.0 memory reppresentation in the WMS

Computing Endpoint	Execution Environment	Computing Manager	Computing Share
EP1	EE1	CM1	SH1
EP2	EE2	CM2	SH2
…	…	…	…



glue2.0 and MM: memory model

- **The new AuthzFramework will be used during the MM**
  - in gLite 3.1 authorization check is performed at classad match evaluation time
    - authorization constraints are expressed as requirements inserted in the CEAd
- **Keeping tracks of**
  - Services exposing Endpoints
    - access policy
  - Services exposing Shares
    - mapping policy
- **simplifies the Authorization filtering phase of the MM**
  - for each EP calling the Authz API it yields
    - the set of CS matching the user credentials via AP
  - for each SH exposed by matching CS
    - remove CS not matching user credentials via MP
- **The actual ClassAd match will be then performed on resulting CSs**



glue2.0 and MM

- **The actual classad representation of a ComputingService will be generated on the fly at MM time by composing the relevant data**
  - *two possible solutions*
    - *all information flattened in one classad*
      - *attribute names prefixed by the name of entity*
        - current behaviour in gLite 3.1
      - *linear construction complexity O(n)*
        - ClassAd::Update, ClassAd::Merge
    - structured nested classads
      - *immediate construction complexity O(1)*
        - swapping pointers to the relevant data
          - ClassAd::Insert, ClassAd::Remove
      - *Users may specify attributes in the requirements expression of the JDL following the schema structure*
        - *other.ComputingService.ExecutionEnv.<attr>*
        - *member(<value>, other.ComputingService.Share.<attr>)*

```
ComputingService = [
  Endpoint = [
     ...
  ]
  Share = [
     ...
  ]
  ExecutionEnv = [
     ...
  ]
  Manager = [
     ...
  ]
  ...
]
```

Enabling Grids for E-sciencE

glue2.0 and MM: matching AEs

- ***ExecutionEnvironments offers zero or more ApplicationEnvironments***
  - *providing the description about individual software packages*
- ***AEs can be shared between different EEs***
  - *again storing the APs using a flyweight pattern may reduce the required memory*

```
...
  ExecutionEnv = [
    ApplicationEnv = {
      [ LocalID = ... ],
        ...
      [ LocalID = ... ]
    }
  ]
...
```

- ***AEs can be inserted in the Environment***
  - *as a list of ClassaAds*
    - *users should use gang-matching expressions*

```
anyMatch(
  other.ComputingService.ExecutionEnv.ApplicationEnv,
target.ParallelSupport == true
);
```

```
...
ExecutionEnv = [
  ID = ...
  ApplicationEnv =
      generateAE( ExecutionEnv.ID );
]
...
```

    - *defining a specialized classad plugin function*
      - *generate the list of relevant AEs at classad evaluation time*
        - *if and only if AEs constraints are actually specified in the requirements expression*

Conclusions

- **Supporting the glue 2.0 schema in the gLite WMS means**
  - restructuring part of the MM engine and the ISM
    - this is anyway required for other reasons:
      - *integrating the new Authz framework*
      - *overcome some limitation of the current MM engine*
        - bug #37911: ISM purchaser should handle Glue 1.3 subcluster software entity
        - bug #36394: gang-matching should be updated for the new SE schema (glue 1.3, SRM 2)
- **Is there any real machine compliant with glue 2.0 so that we can start playing with the schema ?**
- **Is there any deadline scheduled for the transition to glue 2.0 ?**