



Enabling Grids for E-science

gLite Service Discovery

*Charaka Palansuriya, EPCC
EGEE JRA4 F2F – 12 July 2005*

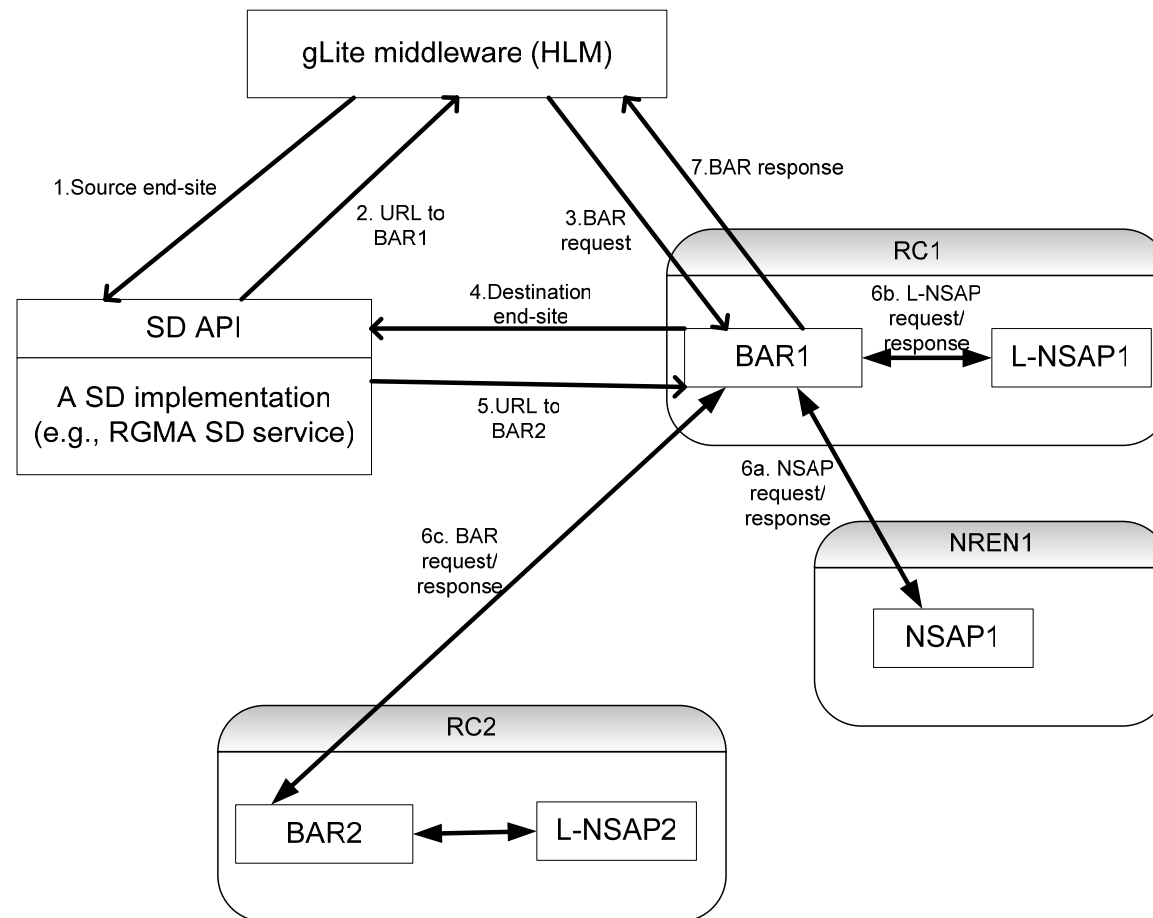
www.eu-egee.org



- **Why BAR requires Service Discovery ?**
- **An architecture for BAR Service Discovery**
- **gLite Service Discovery API**

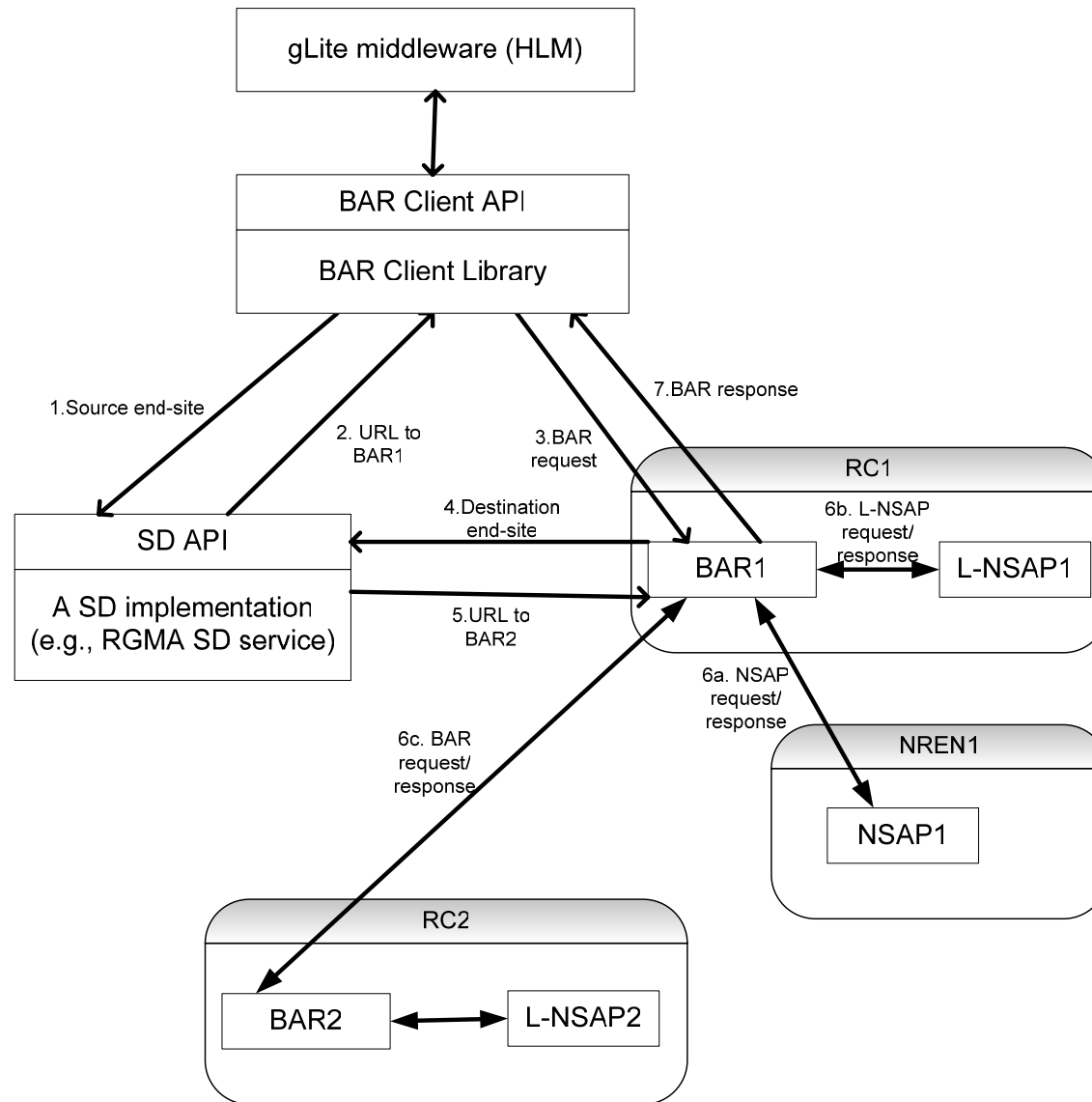
- A BAR web service is used to make an advance reservation of a bandwidth-oriented service (e.g. VLL, GDFT) between two end sites.
- A mechanism is required for gLite middleware (HLM) to locate an unique BAR service base on an end site.
- The gLite Service Discovery (SD) service “may” provide this mechanism

- Assuming a single BAR service is deployed at each RC



- **This implies that HLM has to know how BAR service data is,**
 - **organised**; e.g., BAR service data could map a BAR URL to each unique EGEE end site (or it could organised end sites in terms of RCs and map BAR URLs to RCs),
 - **to be used**; e.g., for a given unique EGEE end site a URL to a BAR service is returned,
 - **to be extended**; e.g., how can a RC add its end points to BAR service data ?

- **We can make the above tasks easier for HLM by providing a BAR Client library with a published API.**
 - MJRA4.4 already contains a simple BAR client API – used by BAR JSP client.



- **Pros:**

- The same gLite SD mechanism can be used to discover the correct NSAP service for a given end point. May be L-NSAP too, if necessary.
- HLM likely to use a simple BAR client API that supports location of the correct BAR service based on an end point than a complicated mechanism that it needs know in order to locate the correct BAR service.

- **Cons:**

- More effort is require by the BAR team to enhance the existing BAR client library and to maintain an API to be used by external software.
- gLite Service Discovery API is immature – likely to evolve.
- gLite WMS may perform similar functionality.

- **The SD API provides uniform access to service details published by information systems.**
- **SD user guide discusses a SD implementation for R-GMA information system.**
- **To use a SD implementation, a set of JAR files are required,**
 - SD API JAR file
 - An implementation of the SD interface (for R-GMA)
 - JAR files associated with the information system itself.

- **Lets assume,**
 - that BAR web service data is registered with an R-GMA information system.
 - the service data for each BAR web service is published such that end point of a BAR web service can be retrieved by submission of key-value pair -> BAR-EndSite
- **Then code to obtain this end point could be,**

```

• public EndPoint getEndPoint(String endSite) {
•
•     ServiceDiscovery sd =
•         new org.glite.rgma.discovery.RGMAServiceDiscovery();
•
•     // Use service data to find the BAR service corresponding
•     // to the endSite. The following should return information
•     // about a single BAR service.
•     Service[] barService = sd.listServicesByData(
•         new ServiceData[] {
•             new ServiceData("BAR", endSite)
•         }, null, null, null, true);
•
•     //return the end point to the service
•     return barService[0].getEndPoint();
• }

```