RLS Background

Project History

- WP2 proposed a distributed replica catalog design in its D2.2 deliverable.
- Globus have presented an almost identical design at GGF3 in Frascati
- At that point we started to work together and agreed to consolidate the work of both groups into a common framework and a common publication. – the Giggle Paper.
- A first release was given to us by March 2002
- A Testing/Release cycle has continued until August 2002
- A GUI was provided by August
- Project reorientation – it became clear that the RLS will not be deployed immediately on the EDG testbed
- No deployment to date but large amount of testing. Demo of the framework (Globus RLS) at SC2002

EDG-RLS

- Also in August it became clear that the needs of LCG/POOL are not fully satisfied by Globus-RLS
- Some performance problems were not solved
- Started a pure java implementation of the server, still sitting on top of the Globus RLS database, being 100% interoperable.
- Deviation of schema when implementing the GUID concept and enforcing a higher level of consistency
- EDG-RLS is providing the LRC functionality and has been deployed on MySQL/Tomcat as well as Oracle/9iAS

Choice of RLS

EDG RLS	Globus RLS
Java-based technology java through JNI Native C, C++, Java, Perl, Python, ... bindings MySQL and Oracle support, easy to extend to more DBMS LRC only at the moment. Support planned for RLIs. Uses Web Service protocols for network communication Small client, no dependencies (on GT2 or others) N:1:M logical to physical file mapping Schema has natural support for GUIDs and alias-ing. Evolution is easier, no code change necessary. SQL is in configuration file.	C-based daemon-style technology C-language binding MySQL-only backend implementation Supports LRCs and RLIs. Uses proprietary Globus Toolkit 2 (GT2) protocols for network communications N:M logical to physical filename mapping Schema is not designed to support GUIDs and aliasing Evolution is very hard due to hardcoded schema and SQL, code change required