EDG Replica Manager and Replica Location Service

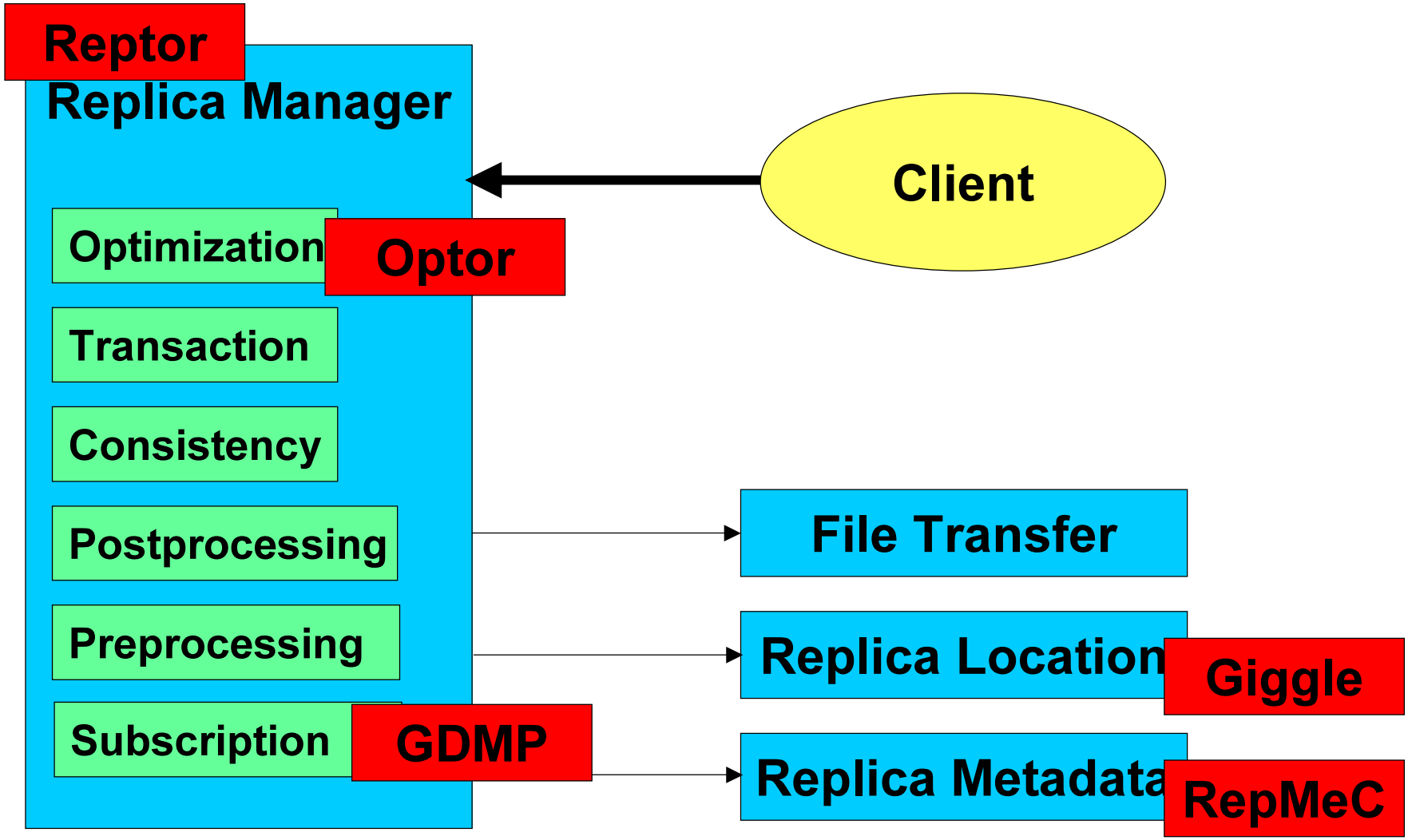
Status and Plans

Leanne Guy

Data Management Work Package: WP2

Leanne.Guy@cern.ch http://cern.ch/grid-data-management

EDG Replication Services



Replica Manager components (1)

Reptor: Replication Manager

- Replication management system.
- Entry point for all clients
- Triggers automated replication of files

Giggle: Replica Location Service

- Local Replica Catalog services LRC: LFN-PFN mappings
- Replica Location Index services RLI: index on LFNs
- Set of configurable servers

GDMP: GRID Data Mirroring Package

- Automated replication of files all over the GRID Storage Elements
- Automatic updating of the replica catalog

RepMec:Replication Metadata Catalogue

- An instance of Spitfire with RDBMS backend and specialized schema

Replica Manager Components (2)

Optor: Optimisation service

- Replica Selection based on economic modelling
- Automated replication for load balancing

Processing

- Hooks for pre- and postprocessing while replicating

Transaction

- Ensure atomic ‘replication’ functionality
- Robustness of service

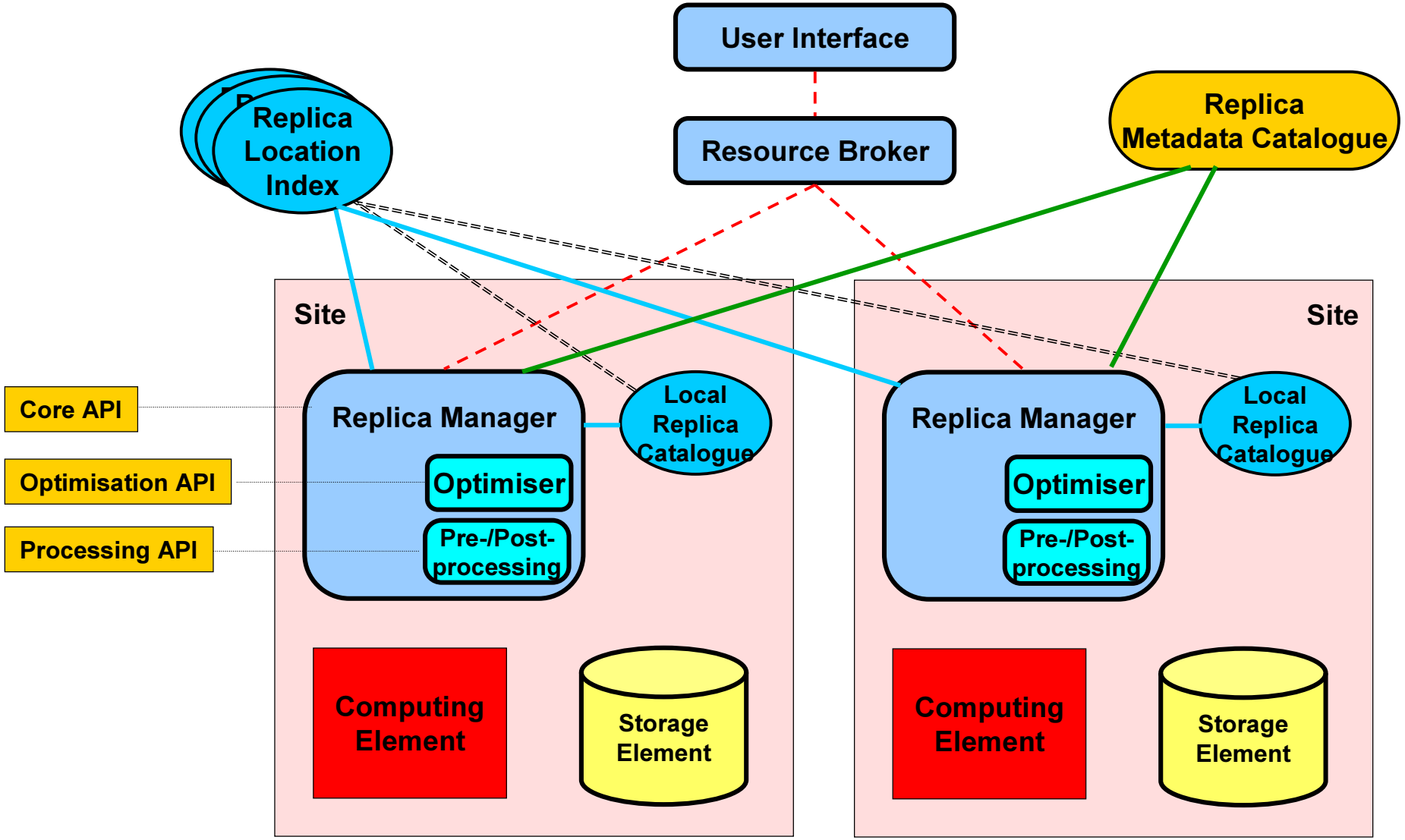
Consistency

- Check consistent state of Replication Services
- Ensure consistent view of files in RLS and SRM
- Ensure consistent Master file attribute

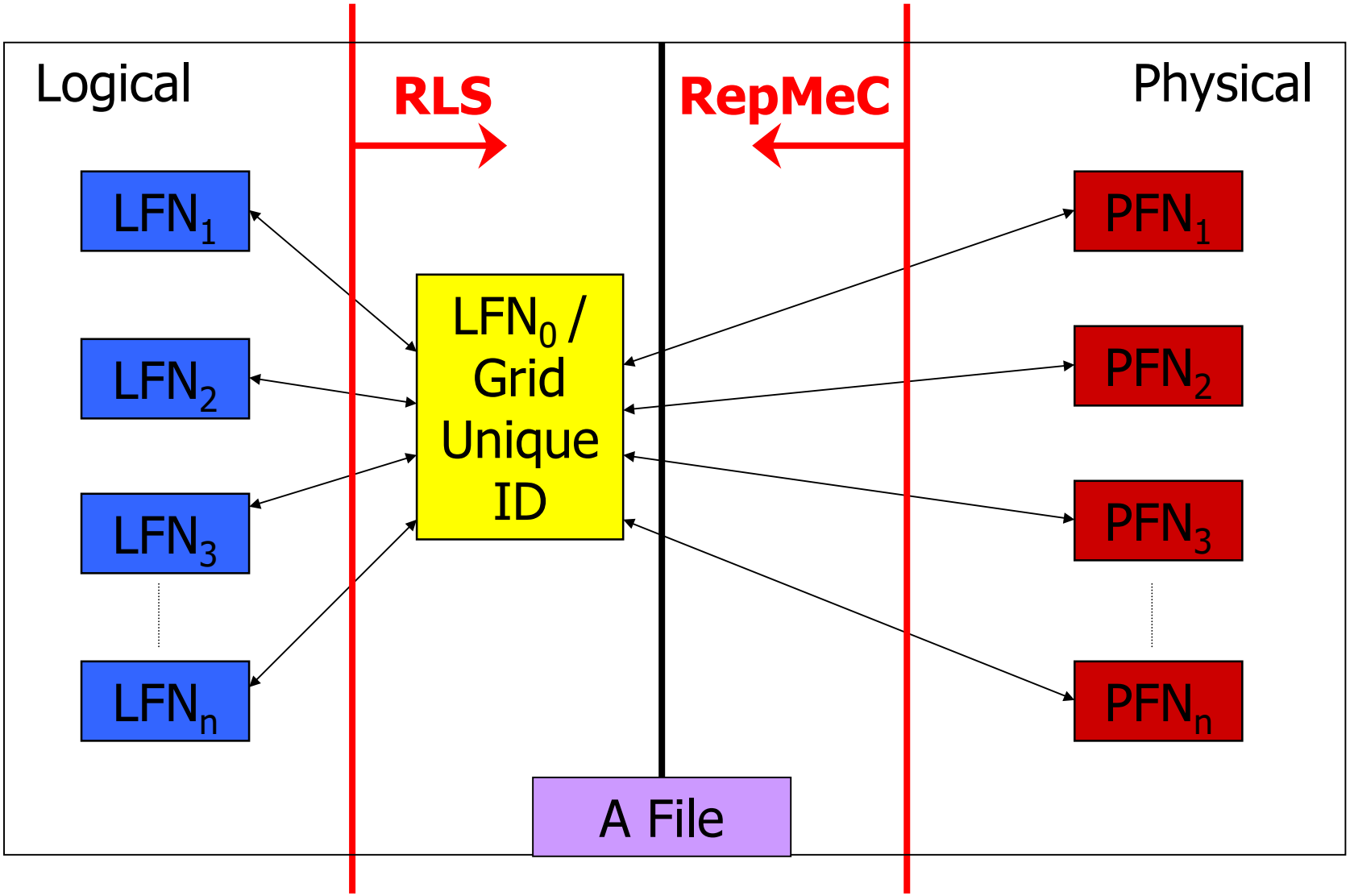
File Transfer

- GridFTP and other protocols

Replica Manager Architecture



File mappings



Replica location problem

Given a unique logical identifier for some given data, determine the physical location of one or more physical instances of this data

Replica Location Service:

- maintains information on the physical location of files
- maintains mapping between the logical identifier of data and all its physical instances
- provides access to this information

RLS Requirements

- Versioning & read only data
  - distinct versions of files can be uniquely identified
  - data published to the community are immutable
- Size
  - scale to hundreds of replica sites, 50 x 10^8 LFNs, 500 x 10^8 PFNs
- Performance
  - 200 updates/second, average response time < 10ms
- Security
  - knowledge and existence of private data must be protected
  - storage system protects integrity of data content
- Consistency
  - view of all available PFNs not necessarily consistent
- Reliability
  - no single point of failure,
  - local and global state decoupled,
    - failure of remote component does not hinder access to local component

Giggle Framework

Giggle: A Framework for Constructing Scalable Replica Location Services

- Joint collaboration between WP2 and Globus
- Paper submitted to SC2002

- Independent local state maintained in Local Replica Catalogues : LRCs
- Unreliable collective state maintained in Replica Location Indices : RLIs
- Soft state maintenance of RLI state
  - relaxed consistency in the RLI, full state information in LRC
- Compression of soft states
  - compress LFN information based on knowledge of logical collections
- Membership and partitioning information maintenance
  - RLS components change over time : failure, new components added
  - Service discovery and system policies

Local Replica Catalogue (LRC)

- Maintains replica information at a single site
  - Complete locally consistent record
  - Queries across multiple sites not supported
- Maintains mappings between LFNs and PFNs on associated storage systems
  - Coordinates its contents with those of the storage system
- Responds to the following queries:
  - Given an LFN, find the set of PFNS associated with that LFN
  - Given a PFN, find the set of LFNS associated with that PFN
- Supports authentication and authorisation when processing remote requests
- Periodically sends information about its state to the RLIs

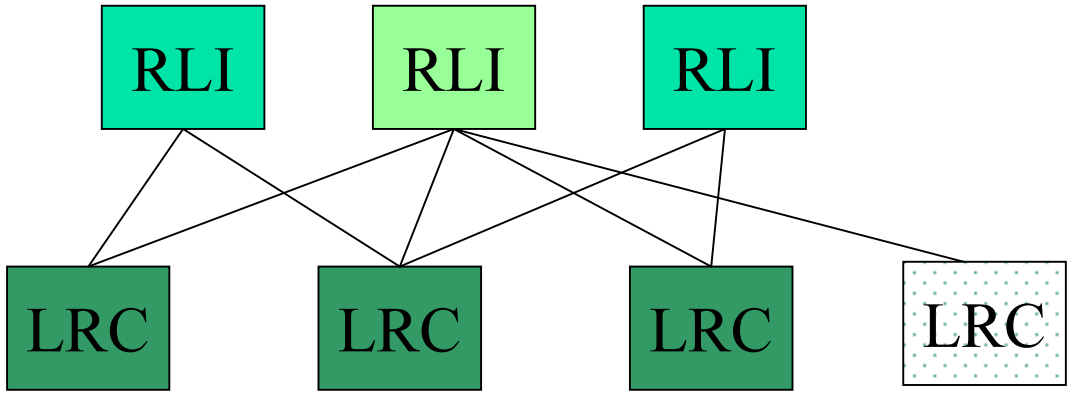
Replica Location Index (RLI)

Index structure needed to support queries across multiple sites

- One or more RLIs to map LFNs to LRCs
  - Structure w.r.t LRCs can be freely defined
  - redundancy, performance, scalability
- Geographical partitioning – all PFNs of a set of LRCs are indexed
- Namespace partitioning 1 – for load balancing purposes
- Namespace partitioning 2 – only LFNs adhering to a specified pattern are indexed for all LRCs
  - possibly not good for load balancing
- Many identical RLIs may be set up for load balancing

RLS Architecture (1)

A 2 level RLS layout: The RLIs contain pointers to LRCs only.



Multiply indexed LRC for higher availability

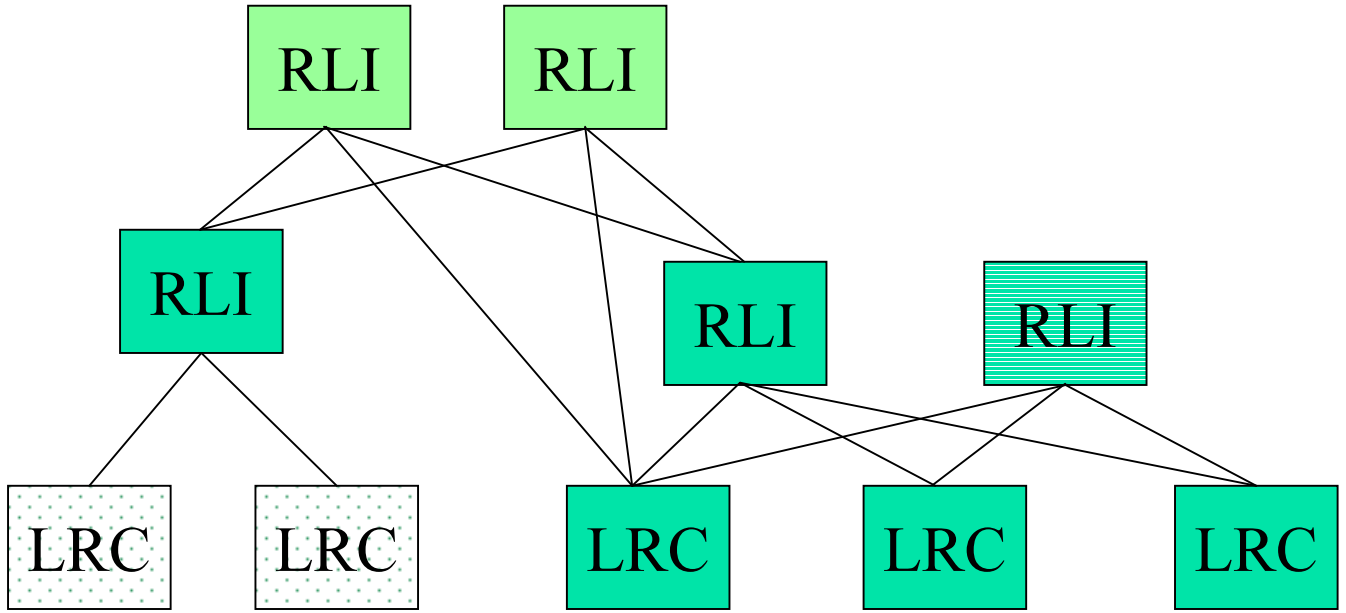
LRC indexed by only one RLI

RLI indexing over the full namespace (all LRCs are indexed)

RLI indexing over a subset of LRCs

RLS Architecture (2)

A hierarchical RLS topology: The RLIs point to LRCs and RLIs



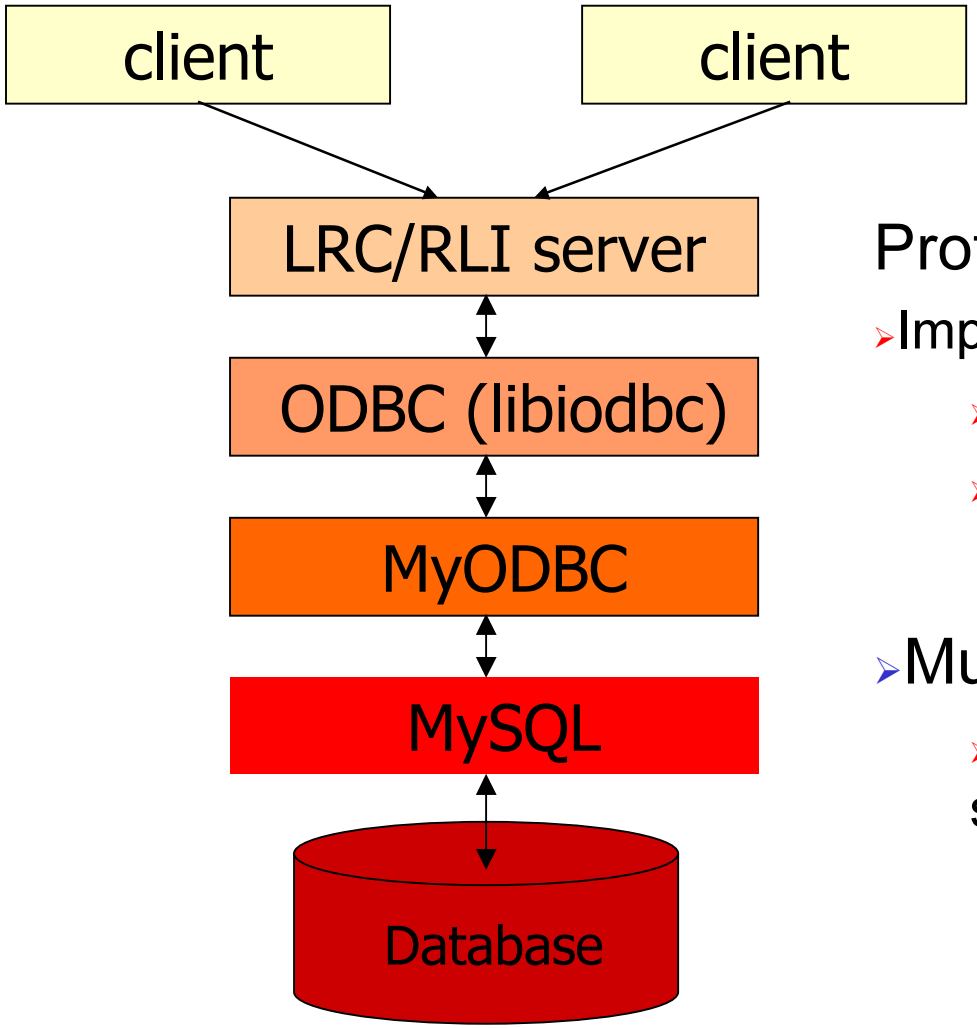
Multiply indexed LRC for higher availability

LRC indexed by only one RLI

RLI indexing over the full namespace (all LRCs are indexed)

RLI indexing over a subset of LRCs

RLS Server Prototype



Prototype implementation:

- Implemented in C; relies on
  - Grid Security Infrastructure
  - globus_io_socket layer
- Multithreaded server
  - configure as an LRC and/or RLI server

Performance results (1)

Preliminary results only !

- Performance results document will be released soon

- Platforms
  - Solaris 2.8 (US) Red Hat Linux 6.1 (CERN)
- Time to add/create/delete/read an LFN entry
  - Number of entries in LRC from 0 -> 1000k
  - ~ 15-16 ms , no noticeable increase in time with database size
- Time to perform soft state update
  - Increases linearly with the number of entries in the LRC
  - ~8 secs for 1000 entries in LRC
  - ~10000 secs for 1000k entries in the LRC
- ~1667 queries/sec, 67 updates/second

Release plans

- RLS is currently installed on 4 EDG nodes at CERN
- Current release is an alpha :
- Initial testing debugging completed
- Giggle paper submitted to SC2002 with preliminary performance results
- Final performance results expected by end July
- Expect to have an RLS - RPM by end of June 2002
- Expect to have a full set of integrated replication services for testbed2 by the end of September

Future work

- Web Services paradigm – the RLS is one of the early adopters of the Open Grid Services Architecture
  - OGSA interface will be available by the end of this year, still needs exact definition
- Compression of RLI state – bloom filters.