The EU DataGrid
Data Management

The European
DataGrid Project Team

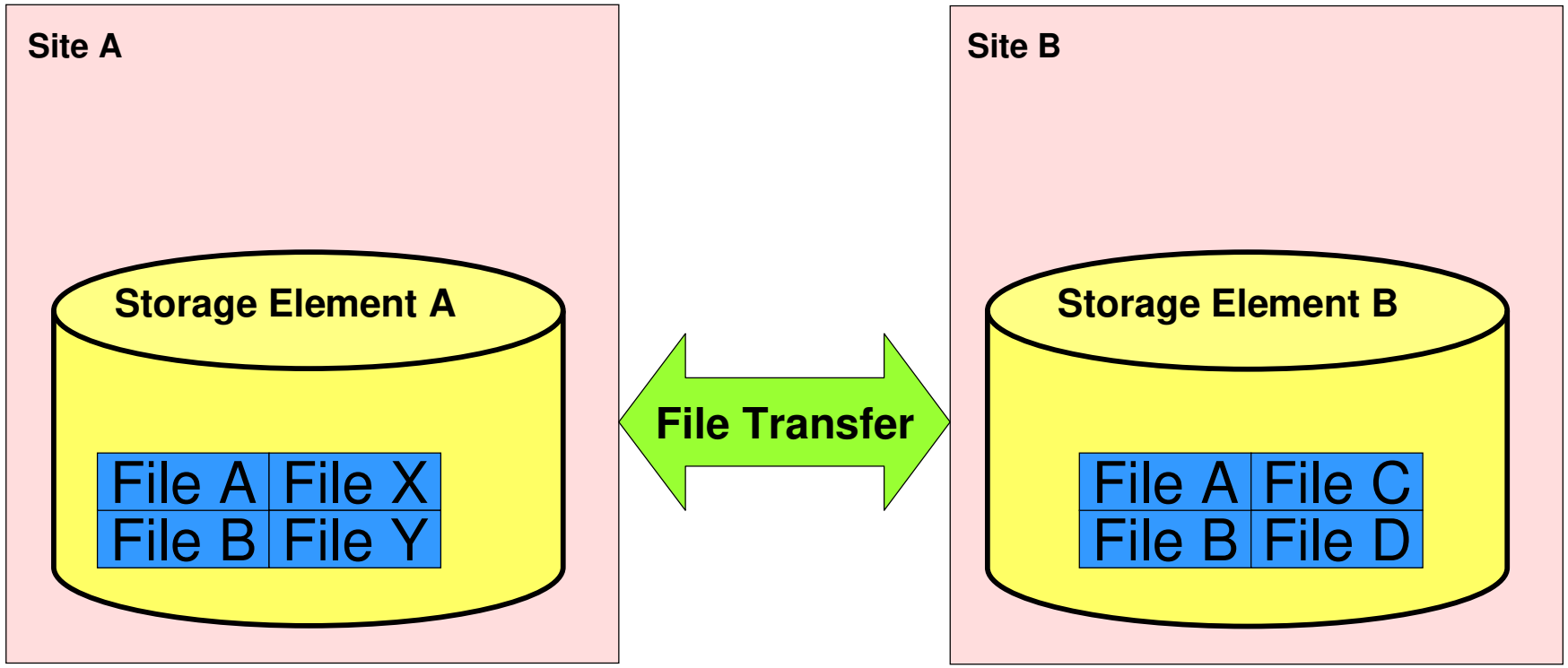
http://www.eu-datagrid.org

Overview

- Data Management Issues

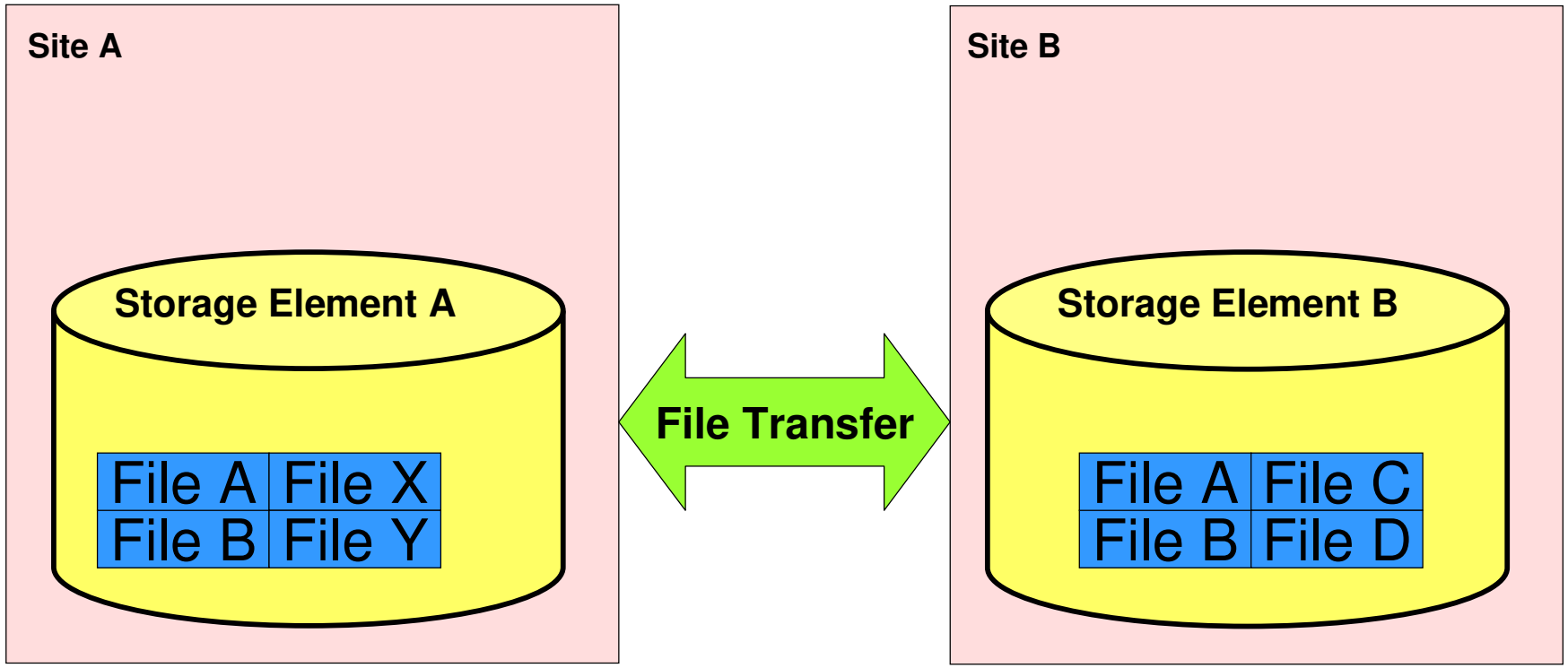
- Main Components
  - Replica Manager
  - Replica Location Service
  - Replica Metadata Catalog
  - Replica Optimization Service

File Management Motivation



File Management Motivation

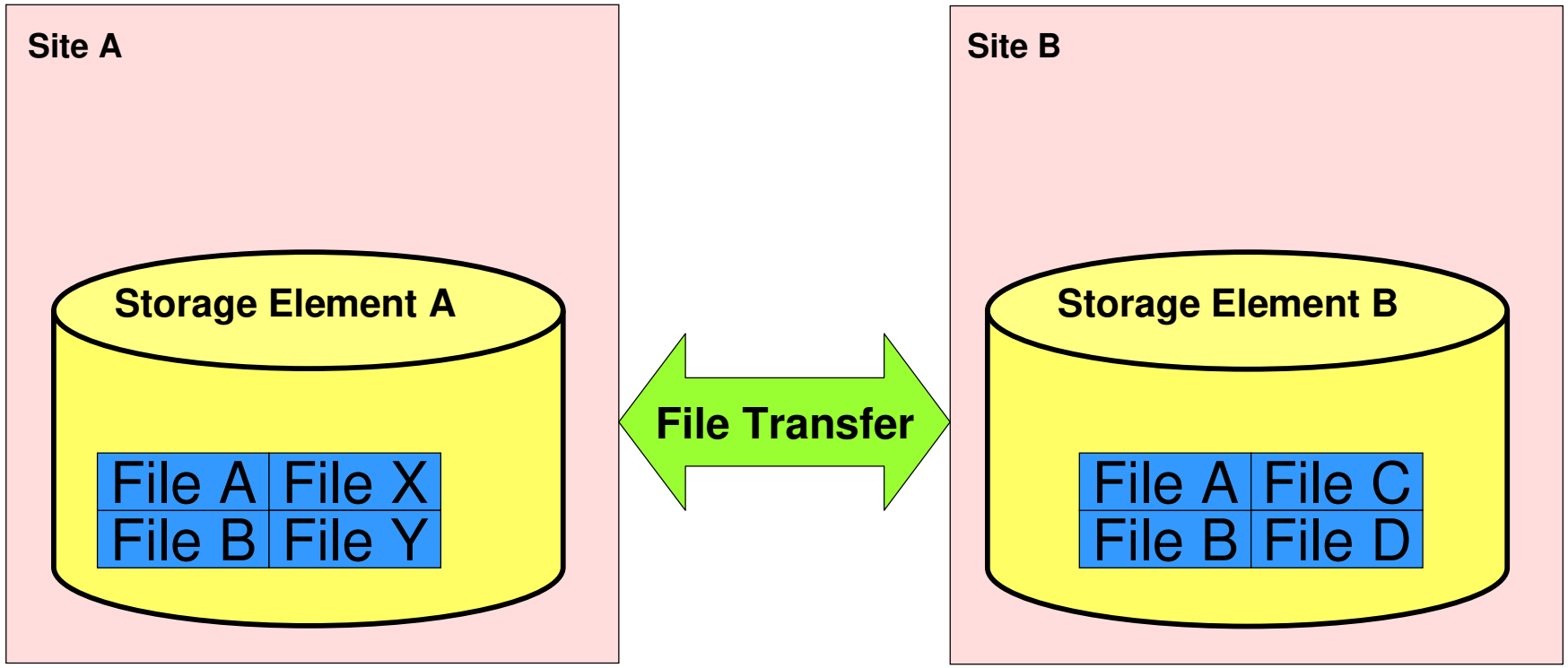
Replica Catalog:
Map Logical to Site files



File Management Motivation

Replica Catalog:
Map Logical to Site files

Replica Selection:
Get 'best' file

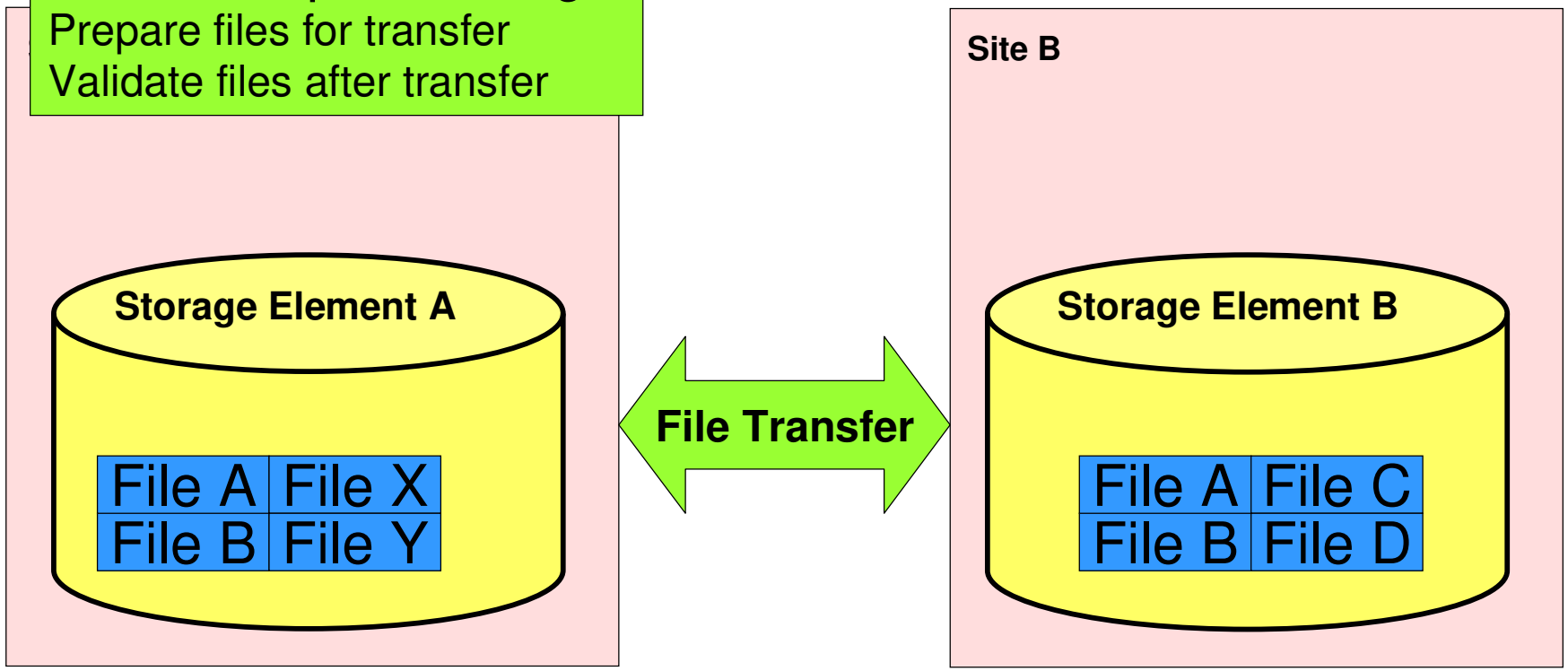


File Management Motivation

Replica Catalog:
Map Logical to Site files

Replica Selection:
Get 'best' file

Pre- Post-processing:
Prepare files for transfer
Validate files after transfer



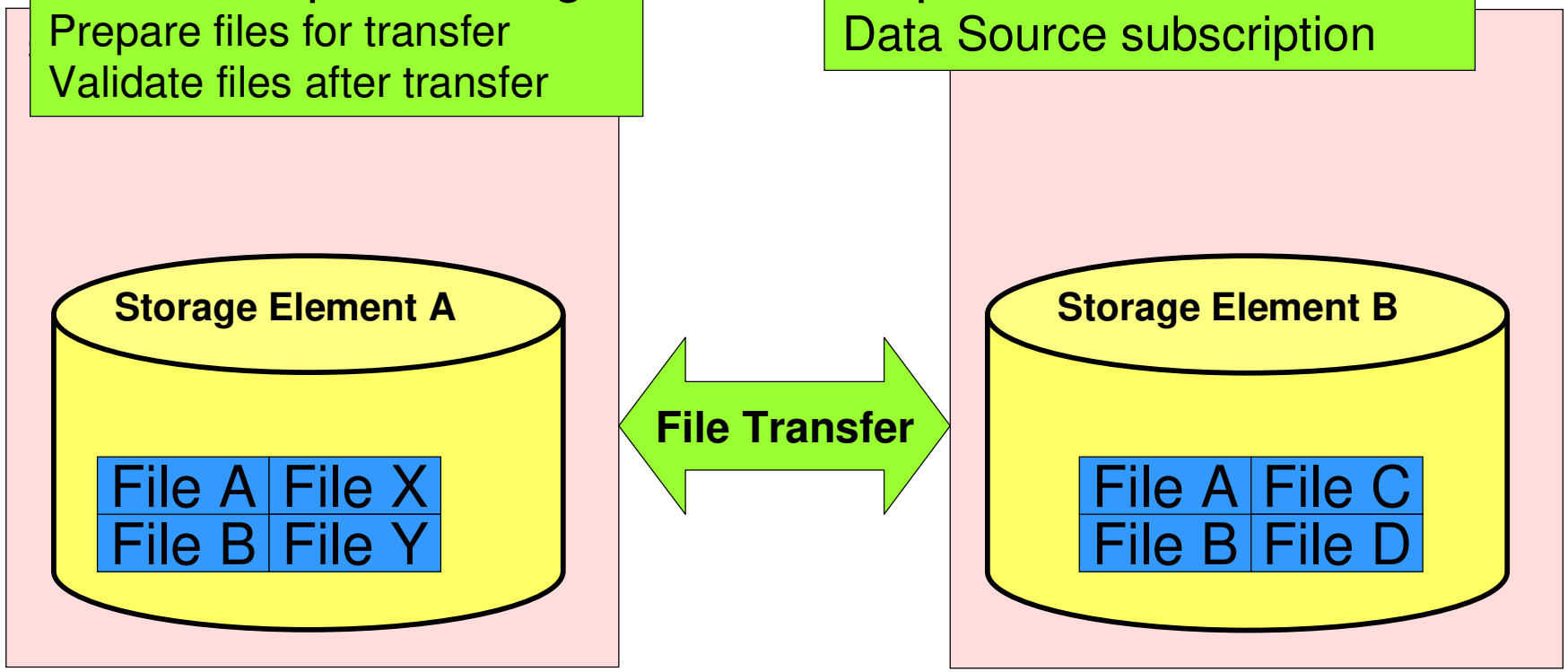
File Management Motivation

Replica Catalog:
Map Logical to Site files

Replica Selection:
Get 'best' file

Pre- Post-processing:
Prepare files for transfer
Validate files after transfer

Replication Automation:
Data Source subscription



File Management Motivation

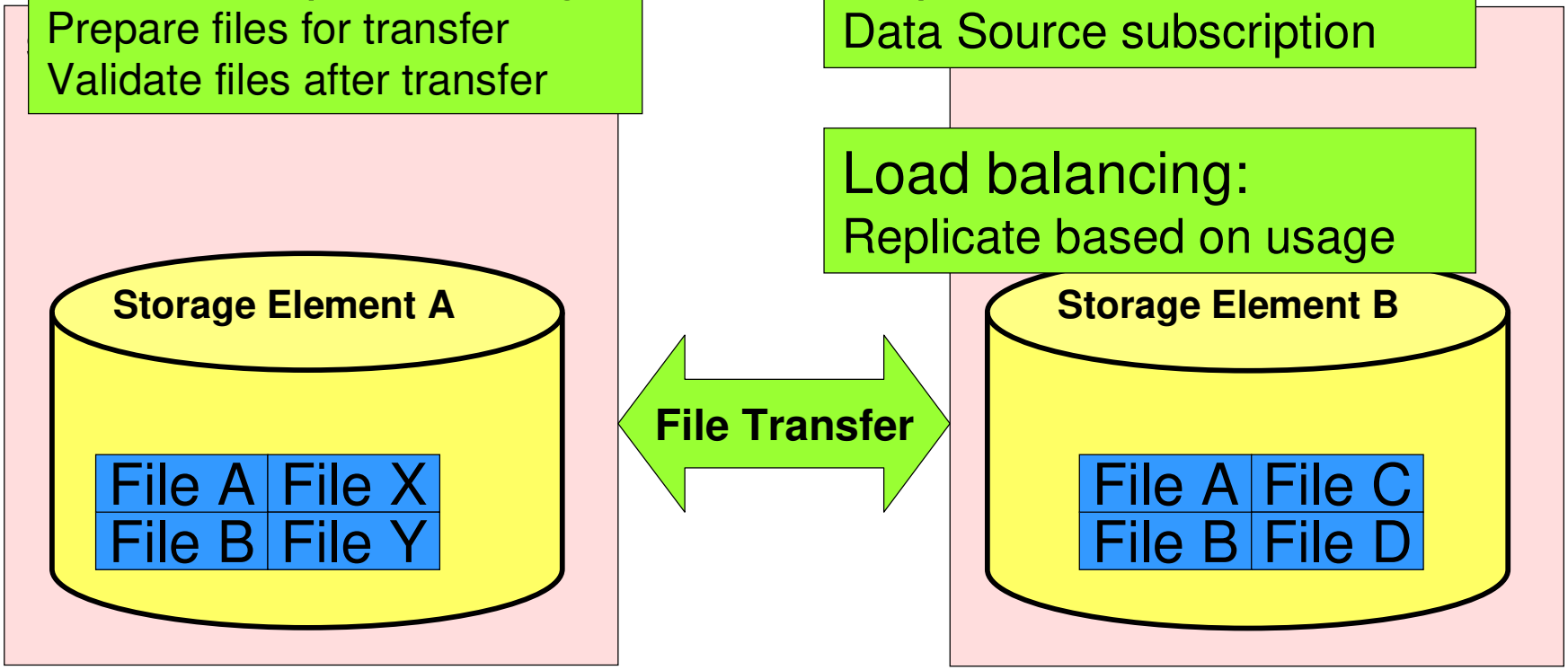
Replica Catalog:
Map Logical to Site files

Replica Selection:
Get 'best' file

Pre- Post-processing:
Prepare files for transfer
Validate files after transfer

Replication Automation:
Data Source subscription

Load balancing:
Replicate based on usage



File Manage

Replica Manager:
‘atomic’ replication operation
single client interface
orchestrator

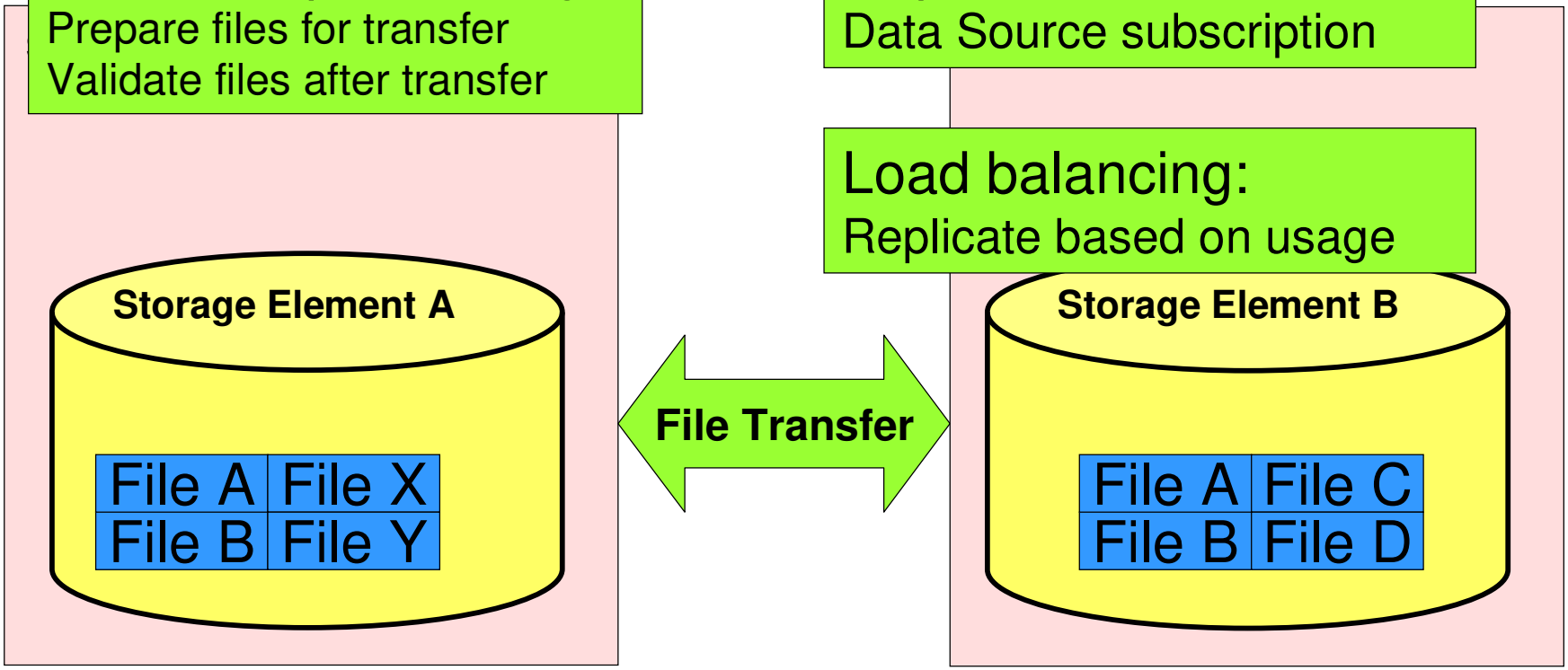
Replica Catalog:
Map Logical to Site files

Replica Selection:
Get ‘best’ file

Pre- Post-processing:
Prepare files for transfer
Validate files after transfer

Replication Automation:
Data Source subscription

Load balancing:
Replicate based on usage



File Manage

Replica Manager:
‘atomic’ replication operation
single client interface
orchestrator

Replica Catalog:
Map Logical to Site files

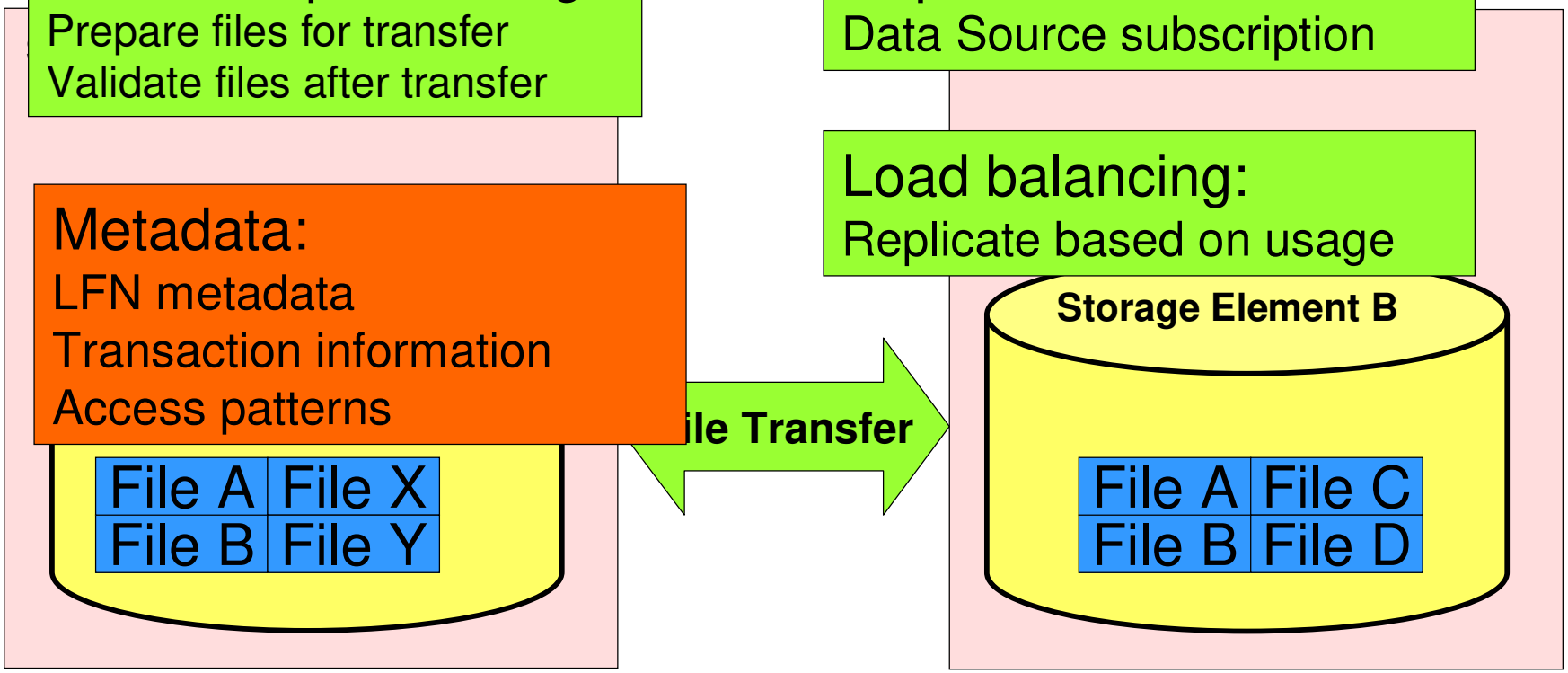
Replica Selection:
Get ‘best’ file

Pre- Post-processing:
Prepare files for transfer
Validate files after transfer

Replication Automation:
Data Source subscription

Load balancing:
Replicate based on usage

Metadata:
LFN metadata
Transaction information
Access patterns



File Manage

Replica Manager:
'atomic' replication operation
single client interface
orchestrator

Replica Catalog:
Map Logical to Site files

Replica Selection:
Get be[illegible]

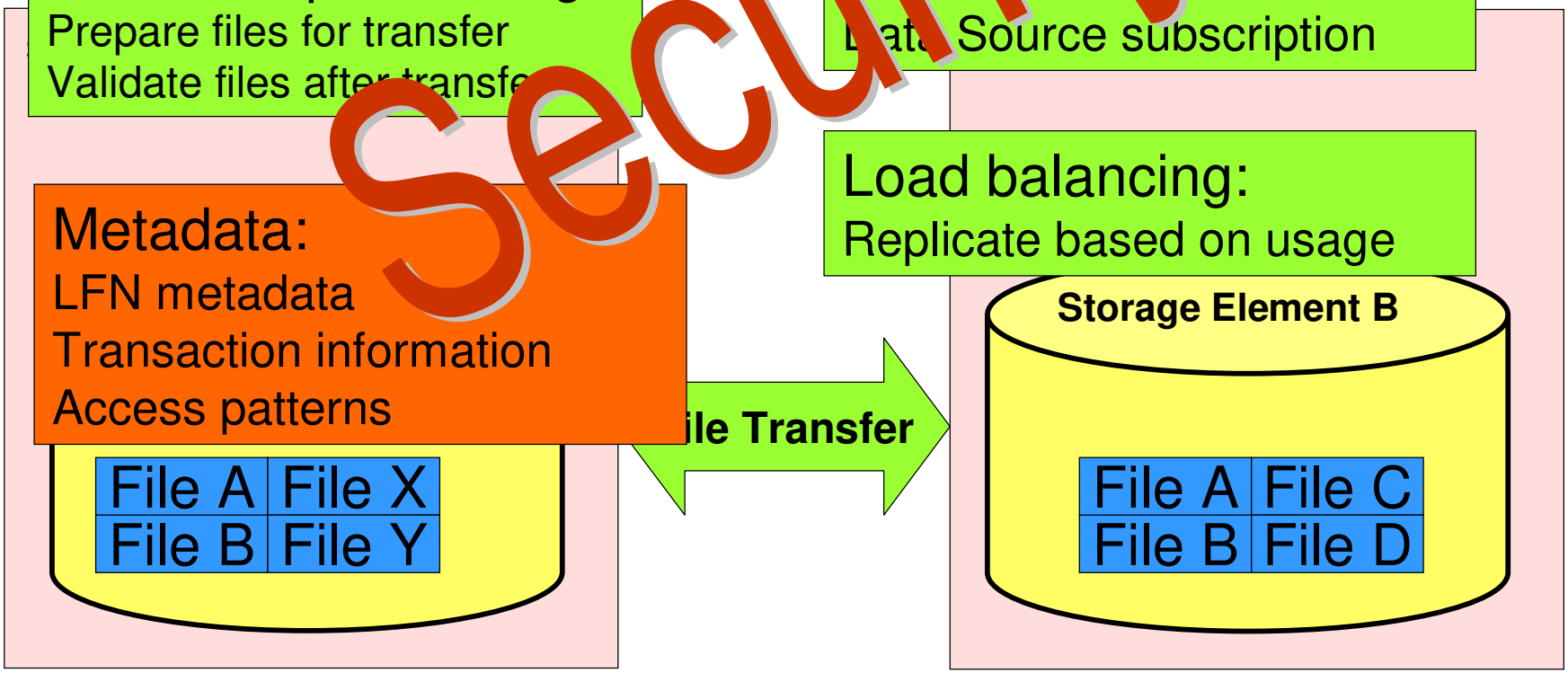
Pre- Post-processing:
Prepare files for transfer
Validate files after transfe[illegible]

Replication Automation:
Data Source subscription

Security

Metadata:
LFN metadata
Transaction information
Access patterns

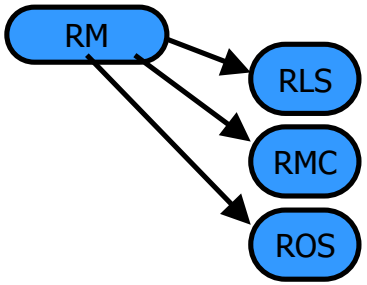
Load balancing:
Replicate based on usage



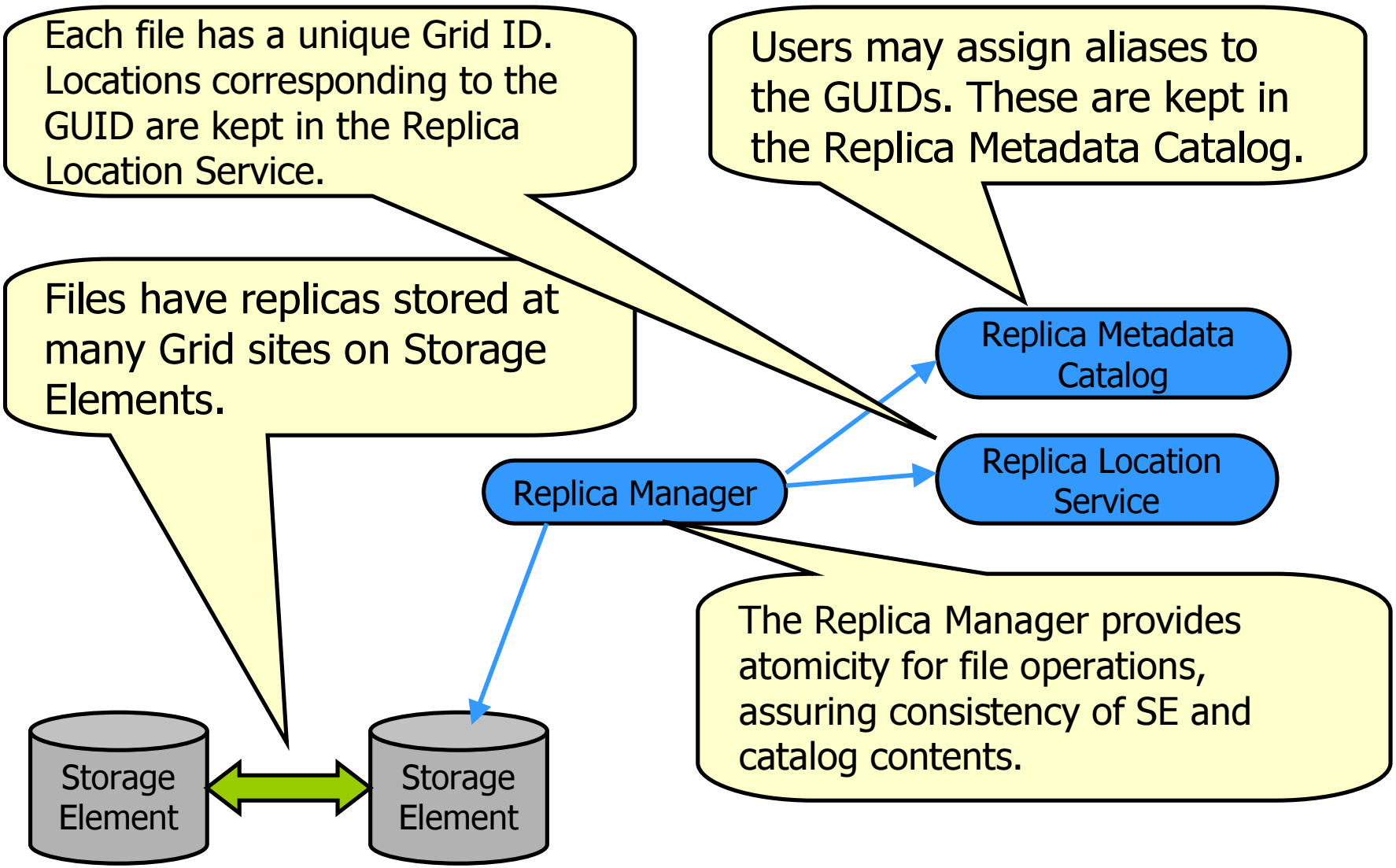
Data Management Tools

- Tools for
  - Locating data
  - Copying data
  - Managing and replicating data
  - Meta Data management

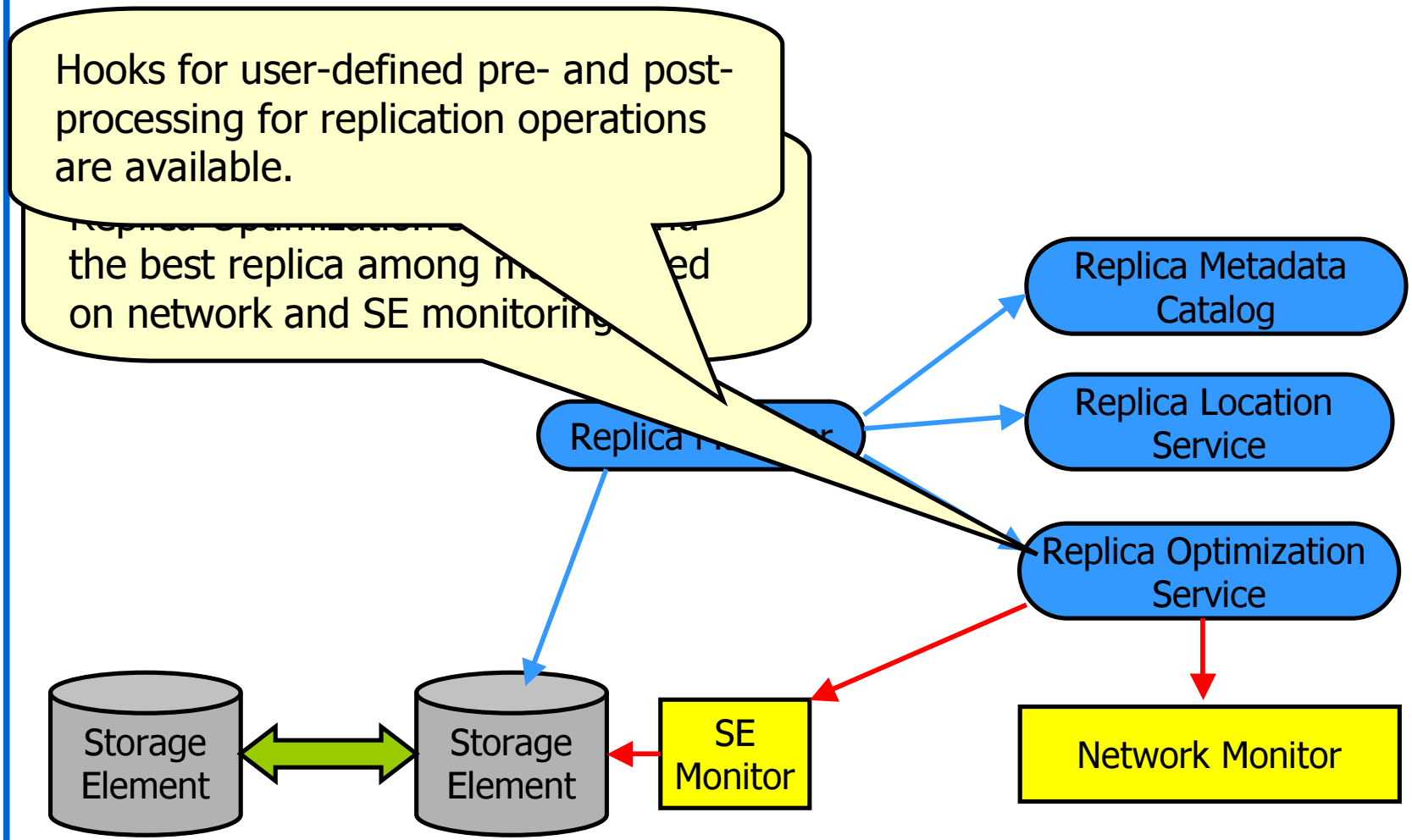
- On EDG Testbed you have
  - Replica Location Service (RLS)
  - Replica Metadata Service (RMC)
  - Repica Optimisation Service (ROS)
  - Replica Manager (RM)



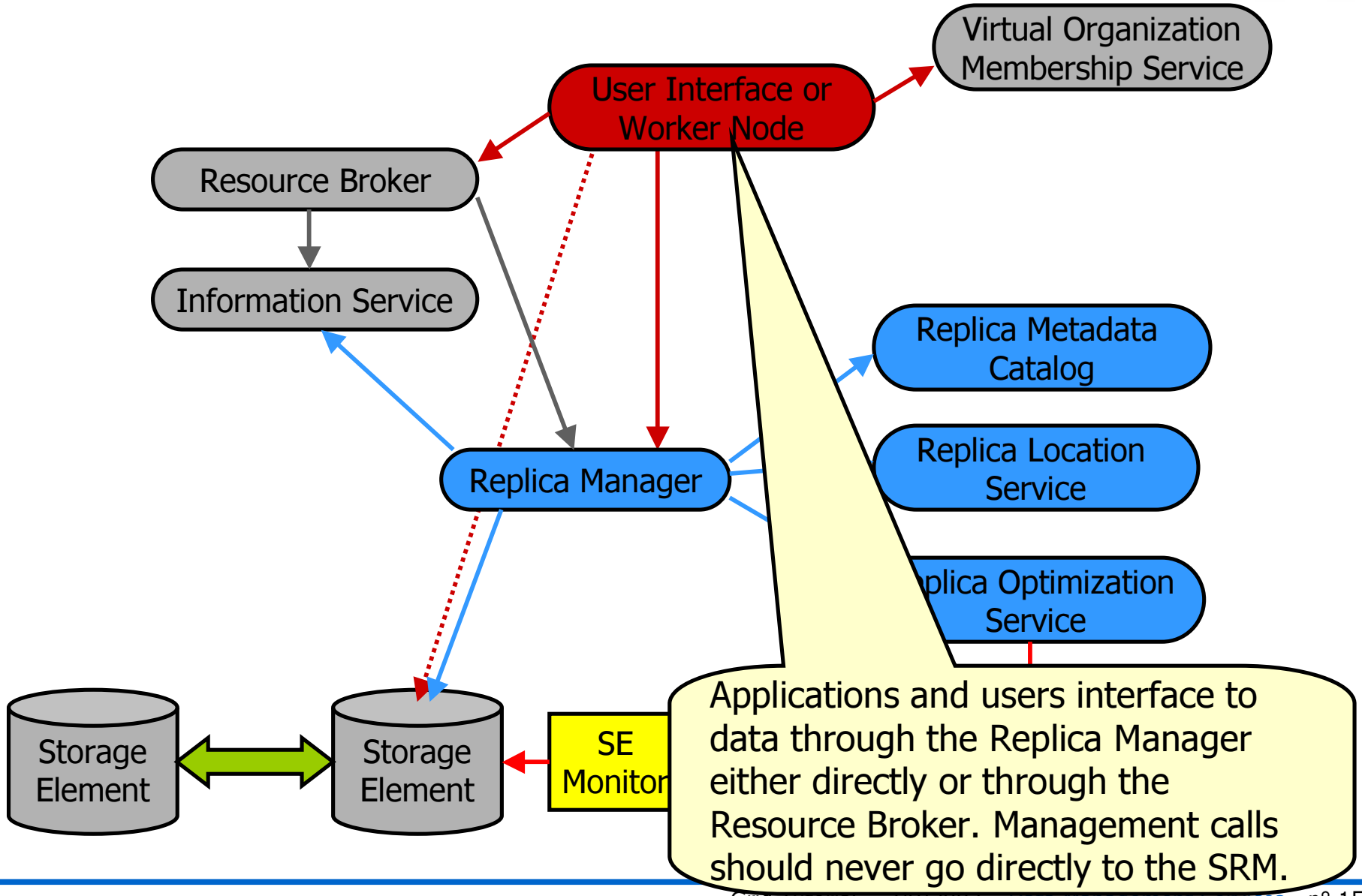
Replication Services: Basic Functionality



Higher Level Replication Services

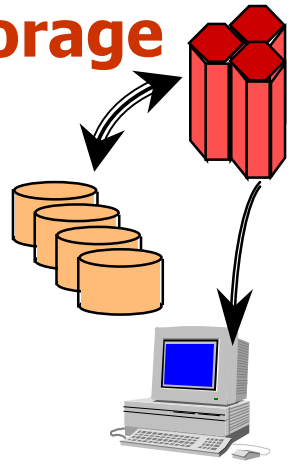


Interactions with other Grid components



Storage Resource Management (1)

Data are stored on **disk pool servers** or **Mass Storage Systems**

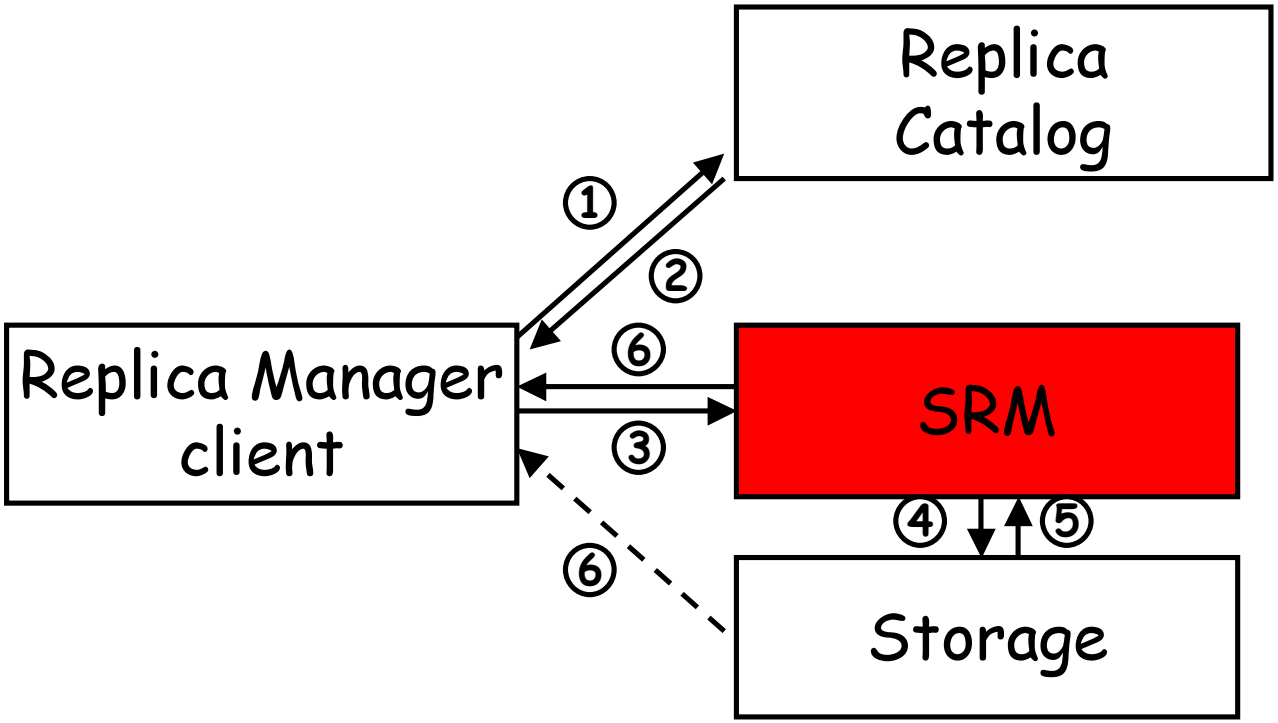


- storage resource management needs to take into account
  - Transparent access to files (migration to/from disk pool)
  - File pinning
  - Space reservation
  - File status notification
  - Life time management
- SRM (Storage Resource Manager) takes care of all these details
  - SRM is a Grid Service that takes care of local storage interaction and provides a Grid interaface to outside world
- In EDG we originally used the term Storage Elemement
  - now we use the term SRM to refer to the new service

Storage Resource Management (2)

- Original SRM design specification: LBL, JNL, FNAL, CERN
- Support for **local policy**
  - Each storage resource can be managed independently
  - Internal priorities are not sacrificed by data movement between Grid agents
- Disk and tape resources are presented as a **single element**
- Temporary **locking**/**pinning**
  - Files can be read from disk caches rather than from tape
- **Reservation** on demand and advance reservation
  - Space can be reserved for registering a new file
  - Plan the storage system usage
- File **status** and **estimates** for planning
  - Provides info on file status
  - Provide estimates on space availability/usage
- EDG provides one implementation as part of the current software release

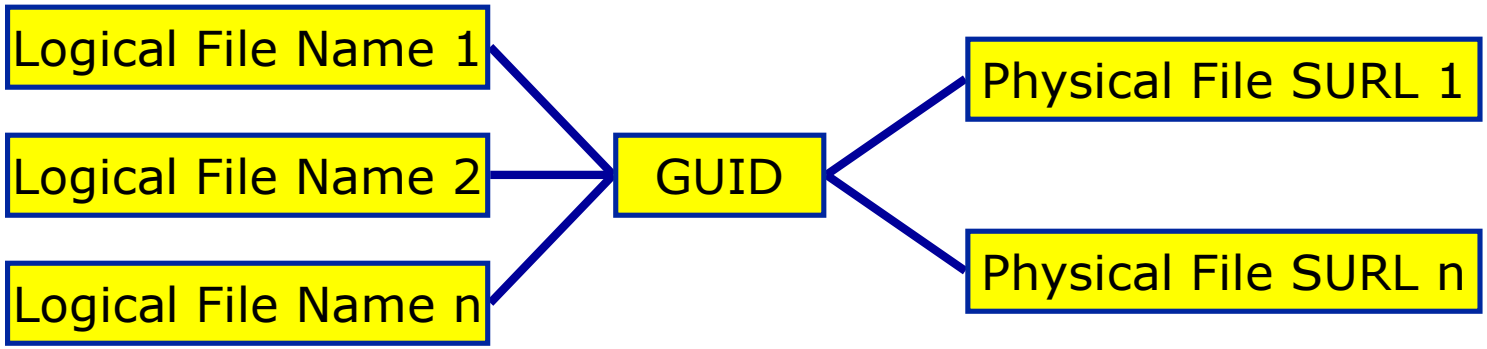
Simplified Interaction Replica Manager - SRM



1. The Client asks a catalog to provide the location of a file
2. The catalog responds with the name of an SRM
3. The client asks the SRM for the file
4. The SRM asks the storage system to provide the file
5. The storage system sends the file to the client through the SRM or
6. directly

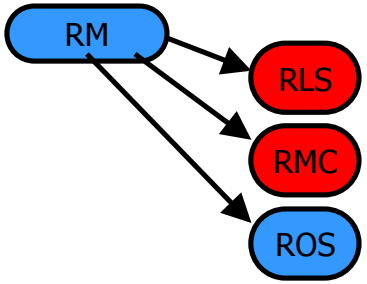
Naming Conventions

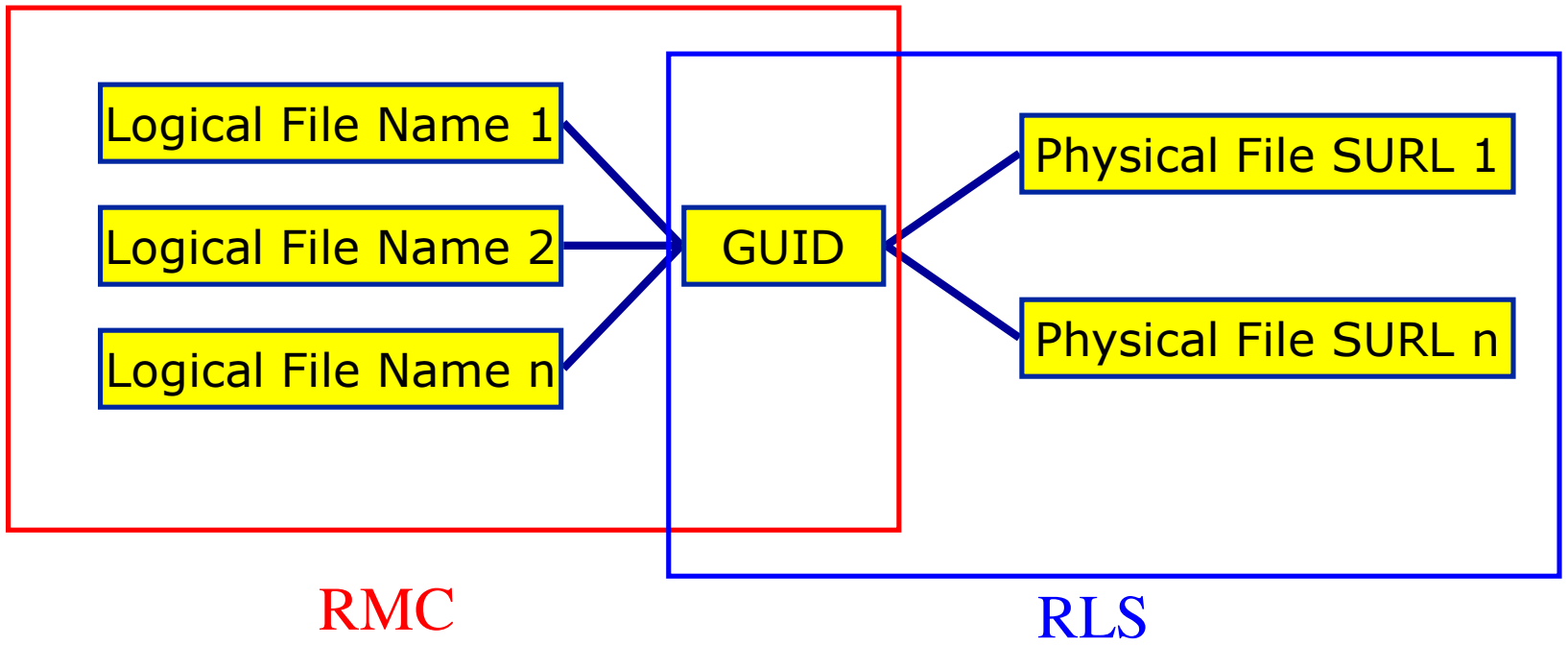
- Logical File Name (LFN)
  - An alias created by a user to refer to some item of data e.g. "lfn:cms/20030203/run2/track1"
- Site URL (SURL) (or Physical File Name (PFN))
  - The location of an actual piece of data on a storage system e.g. "srm//:pcrd24.cern.ch/flatfiles/cms/output10_1"
- Globally Unique Identifier (GUID)
  - A non-human readable unique identifier for an item of data e.g. "guid:f81d4fae-7dec-11d0-a765-00a0c91e6bf6"



Replica Metadata Catalog (RMC) vs. Replica Location Service (RLS)

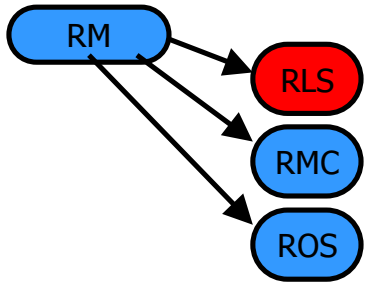
- RMC:
  - Stores LFN-GUID mappings
- RLS:
  - Stores GUID-SURL mappings



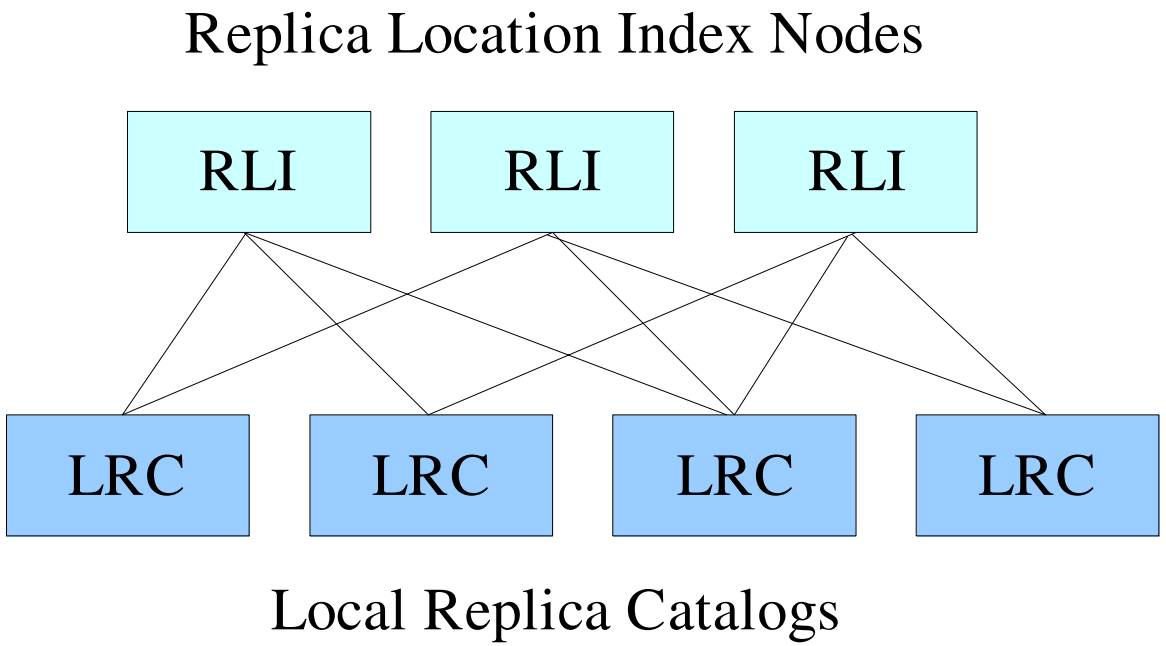


Replica Location Service (RLS)

- The Replica Location Service is a system that maintains and provides access to information about the physical location of copies of data files.
- It is a distributed service that stores mappings between globally unique identifiers of datafiles and the physical identifiers of all existing replicas of these datafiles.
- Design is a joint collaboration between Globus and EDG-WP2



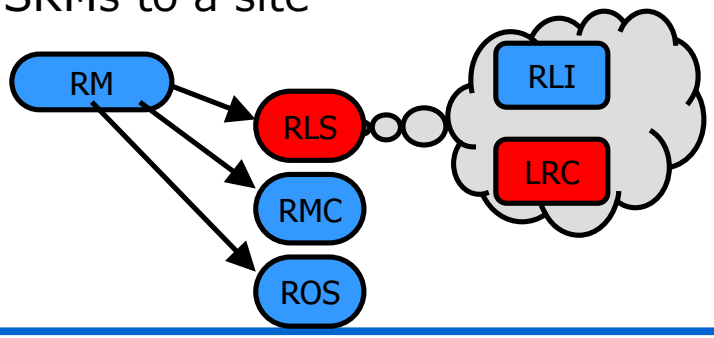
Replica Location Service RLS



- Local Catalogs hold the actual name mappings
- Remote Indexes redirect inquiries to LRCs actually having the file
- LRCs are configured to send index updates to any number of RLIs
- Indexes are Bloom Filters

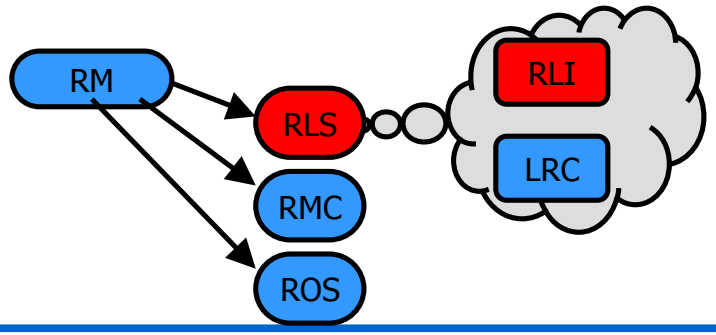
RLS Components (1)

- **Local Replica Catalog (LRC)**
  - Stores GUID to SURL (PFN) mappings for a single SRM
  - Stores attributes on SURL (PFN),
    - e.g file size, creator
  - Maintains local state independently of other RLS components
    - complete local record of all replicas on a single SRM
  - Many Local Replica Catalogs in a Grid
    - can be configured, for instance one LRC per site or one LRC per SRM
  - Fairly permanent fixture
    - LRC coupled to the SRM, SRM removal is infrequent
    - new LRCs added with addition of new SRMs to a site



RLS Components (2)

- **Replica Location Index (RLI)**
  - Stores GUID to LRC mappings
  - Distributed index over Local Replica Catalogs in a Grid
  - Receives periodic soft state updates from LRCs
    - information has an associated expiration time
    - LRCs configured to send updates to RLIs (push)
  - Maintains collective state
    - inconsistencies due to the soft state update mechanism
  - Can be installed anywhere
    - not inherently associated with anything
  - Uses Bloom Filter Indexes



LRC Implementation

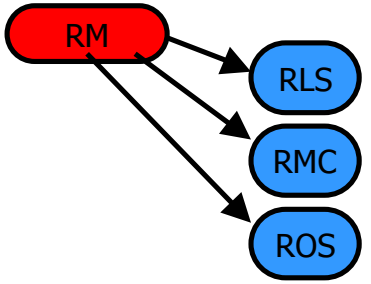
- LRC data stored in a Relational Database
  - Runs with either Oracle 9i or MySQL
- Catalogs implemented in Java and hosted in a J2EE application server
  - Tomcat4 or Oracle 9iAS for application server
  - Java and C++ APIs exposed to clients through Apache Axis (Java) and gSoap (C++)
- Catalog APIs exposed using WSDL
- Vendor neutral approach taken to allow different deployment options

RLI Implementation

- Updates implemented as a push from the LRCs
  - RLIs less permanent than LRCs
- Bloom filter updates only implemented
  - O(10^6) (or more ☺) entries in an LRC
  - May contain false positives, but no false negatives
    - rate depends on the configutation of the bloom filter
- Bloom filters stored to disk
- Impractical to send full LRC lists to multiple RLIs
  - fine for tests
  - not scalable in a production environment
- Implemented in Java as a web service as for the LRCs

User Interfaces for Data Management

- Users are mainly referred to use the interface of the **Replica Manager client**:
  - Management commands
  - Catalog commands
  - Optimization commands
  - File Transfer commands



- The services RLS, RMC and ROS provide additional user interfaces
  - Mainly for additional catalog operations (in case of RLS, RMC)
  - Additional server administration commands
    - Should mainly be used by administrators
    - Can also be used the check the availability of a service

The Replica Manager Interface – *Management Commands*

- copyAndRegisterFile args: source, dest, lfn, protocol, streams
  - Copy a file into grid-aware storage and register the copy in the Replica Catalog as an atomic operation.
- replicateFile args: source/lfn, dest, protocol, streams
  - Replicate a file between grid-aware stores and register the replica in the Replica Catalog as an atomic operation.
- deleteFile args: source/seHost, all
  - Delete a file from storage and unregister it.
- Example

```
edg-rm --vo=tutor copyAndRegisterFile
file:/home/bob/analysis/data5.dat
-d lxshare0384.cern.ch
```

The Replica Manager Interface – *Catalog Commands (1)*

- registerFile args: source, lfn
  - Register a file in the Replica Catalog that is already stored on a Storage Element.
- unregisterFile args: source, guid
  - Unregister a file from the Replica Catalog.
- listReplicas args: lfn/surl/guid
  - List all replicas of a file.
- registerGUID args: surl, guid
  - Register an SURL with a known GUID in the Replica Catalog.
- listGUID args: lfn/surl
  - Print the GUID associated with an LFN or SURL.

The Replica Manager Interface – *Catalog Commands (2)*

- addAlias args: guid, lfn
  - Add a new alias to GUID mapping
- removeAlias args: guid, lfn
  - Remove an alias LFN from a known GUID.
- printInfo()
  - Print the information needed by the Replica Manager to screen or to a file.
- getVersion()
  - Get the versions of the replica manager client.

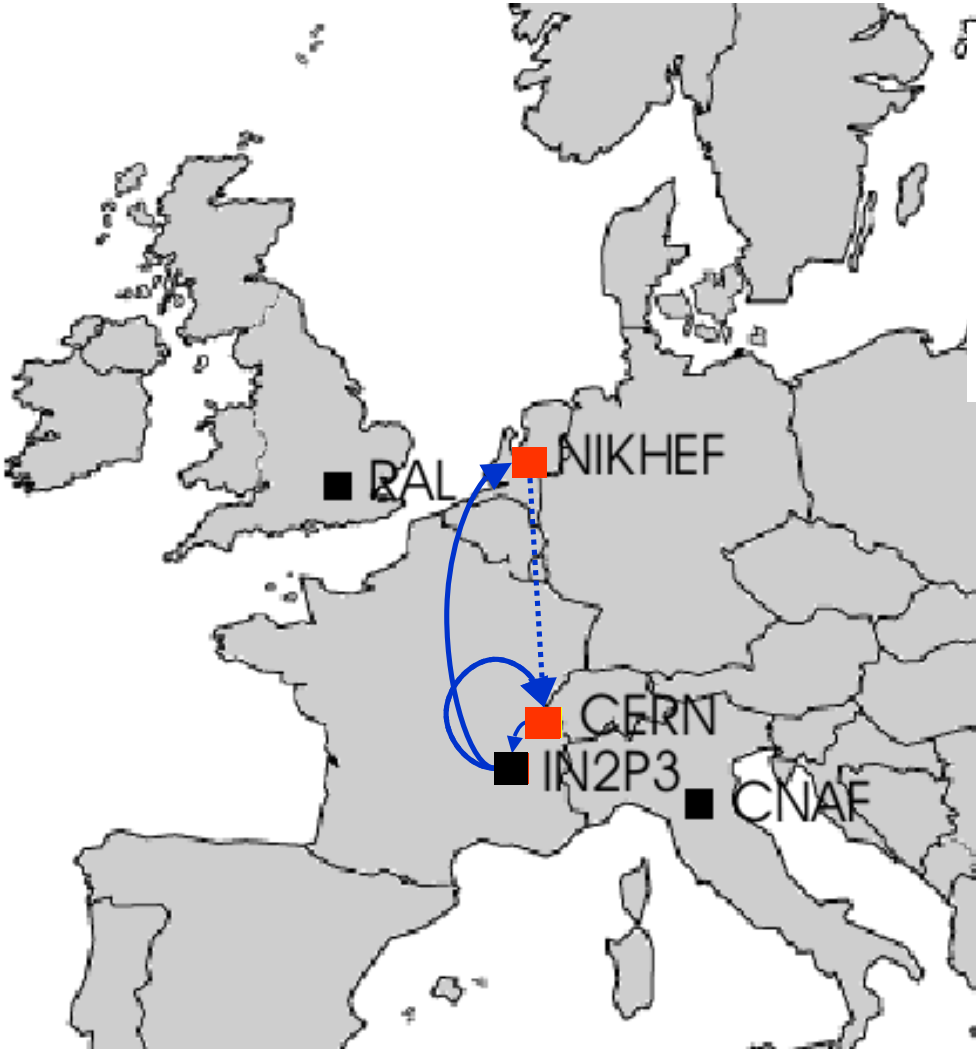
The Replica Manager Interface – *Optimization Commands*

- listBestFile args: lfn/guid, seHost
  - Return the 'best' replica for a given logical file identifier.
- getBestFile args: lfn/guid, seHost, protocol, streams
  - Return the storage file name (SFN) of the best file in terms of network latencies.
- getAccessCost args: lfn/guid[], ce[], protocol[]
  - Calculates the expected cost of accessing all the files specified by logicalName from each Computing Element host specified by ceHosts.

The Replica Manager Interface – *File Transfer Commands*

- copyFile args: soure, dest
  - Copy a file to a non-grid destination.
- listDirectory args: dir
  - List the directory contents on an SRM or a GridFTP server.

Replica Management Use Case



edg-rm copyAndRegisterFile -l lfn:higgs
CERN → LYON

edg-rm listReplicas -l lfn:higgs

edg-rm replicateFile -l lfn:higgs
→ NIKHEF

edg-rm listBestFile -l lfn:higgs
→ CERN

edg-rm getAccessCost -l lfn:higgs
CERN NIKHEF LYON

edg-rm getBestFile -l lfn:higgs
→ CERN

edg-rm deleteFile -l lfn:higgs
→ LYON

edg-rm listBestFile -l lfn:higgs
→ CERN

Metadata Management and Security

Project Spitfire

- 'Simple' Grid Persistency
  - Grid Metadata
  - Application Metadata
  - Unified Grid enabled front end to relational databases.
- *Metadata Replication and Consistency*
- *Publish information on the metadata service*

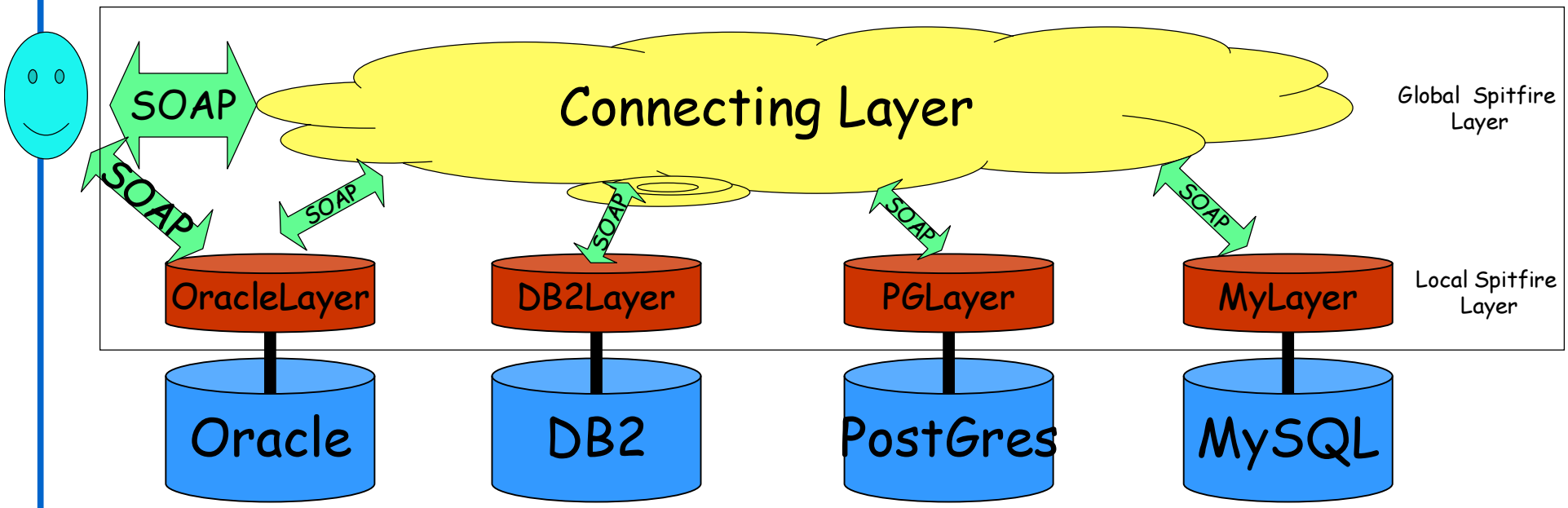
Secure Grid Services

- Grid authentication, authorization and access control mechanisms enabled in Spitfire
- Modular design, reusable by other Grid Services

Spitfire Architecture

- browser access and dedicated client access through a **uniform, secure interface**
- Web/Grid Services Paradigm
  - SOAP interfaces
  - JDBC interface to RDBMS
- Plugability and extensibility

- Atomic RDBMS is always consistent
- No local replication of data
- authentication and fine-grained authorization based on the user's certicate



Conclusions and Further Information

- The second generation Data Management services have been designed and implemented based on the Web Service paradigm
- Flexible, extensible service framework
- Deployment choices : robust, highly available commercial products supported (eg. Oracle) as well as open-source (MySQL, Tomcat)
- First experiences with these services show that their performance meets the expectations
- Further information / documentation:
  - www.cern.ch/edg-wp2