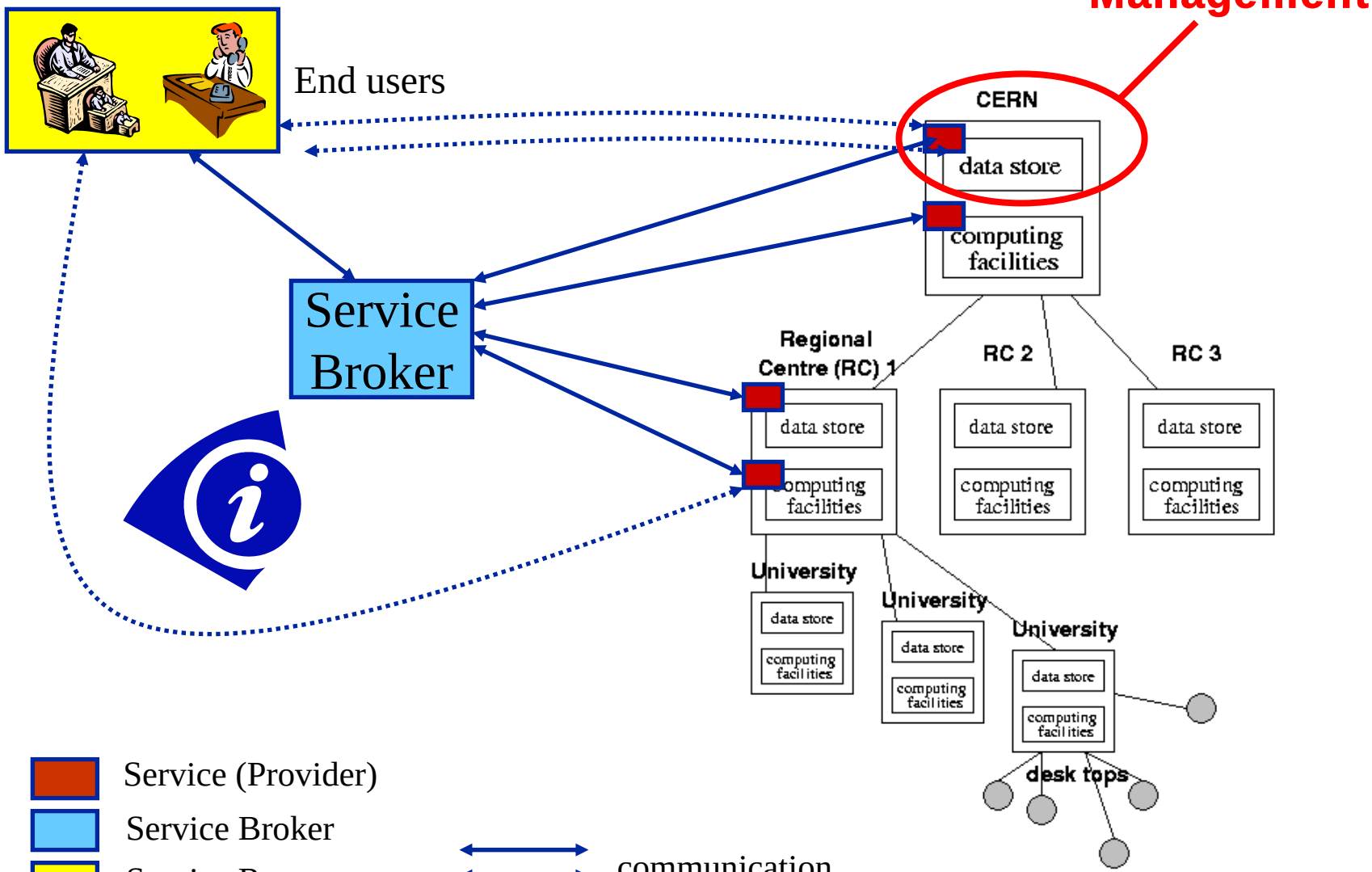
Data Management

Heinz Stockinger

CERN & INFN

A Reference Grid



(Grid) Storage Systems

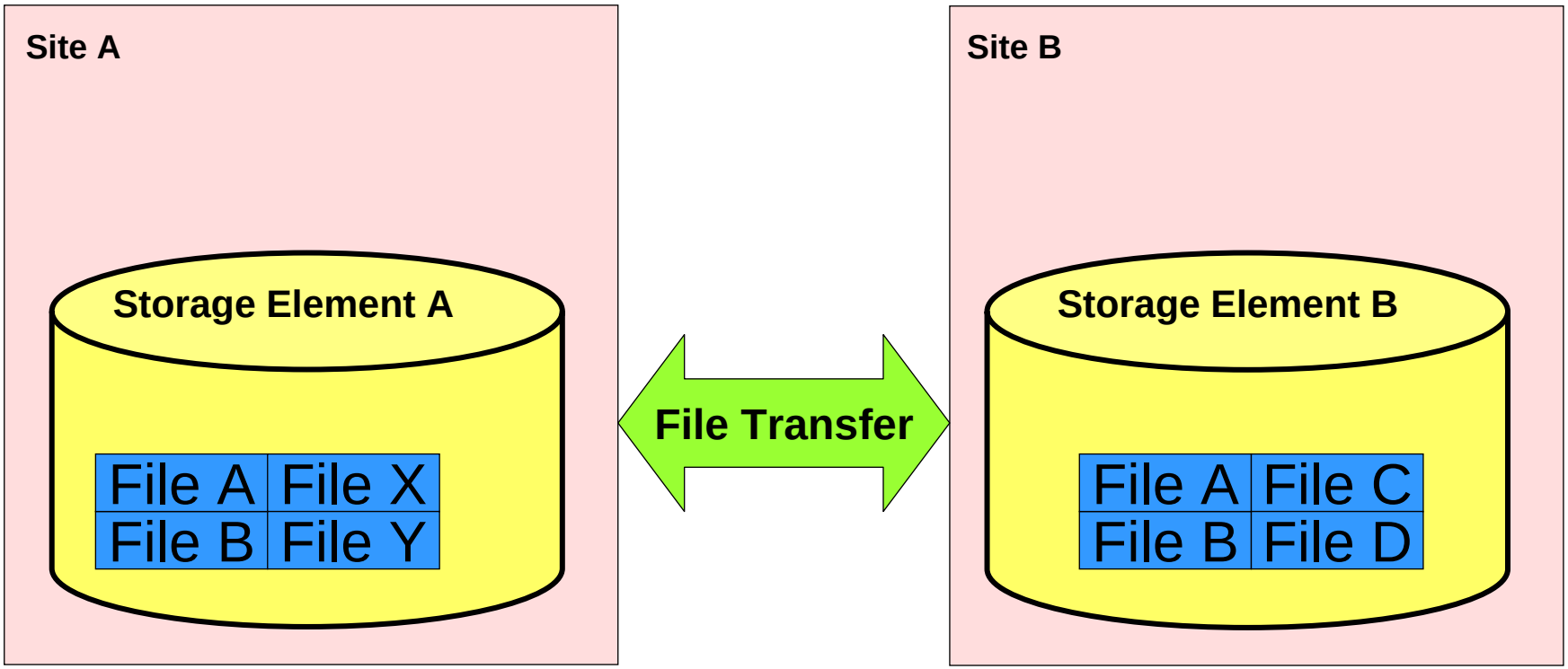
Data Transfer

Data Cataloging / Metadata Management

Replica Management

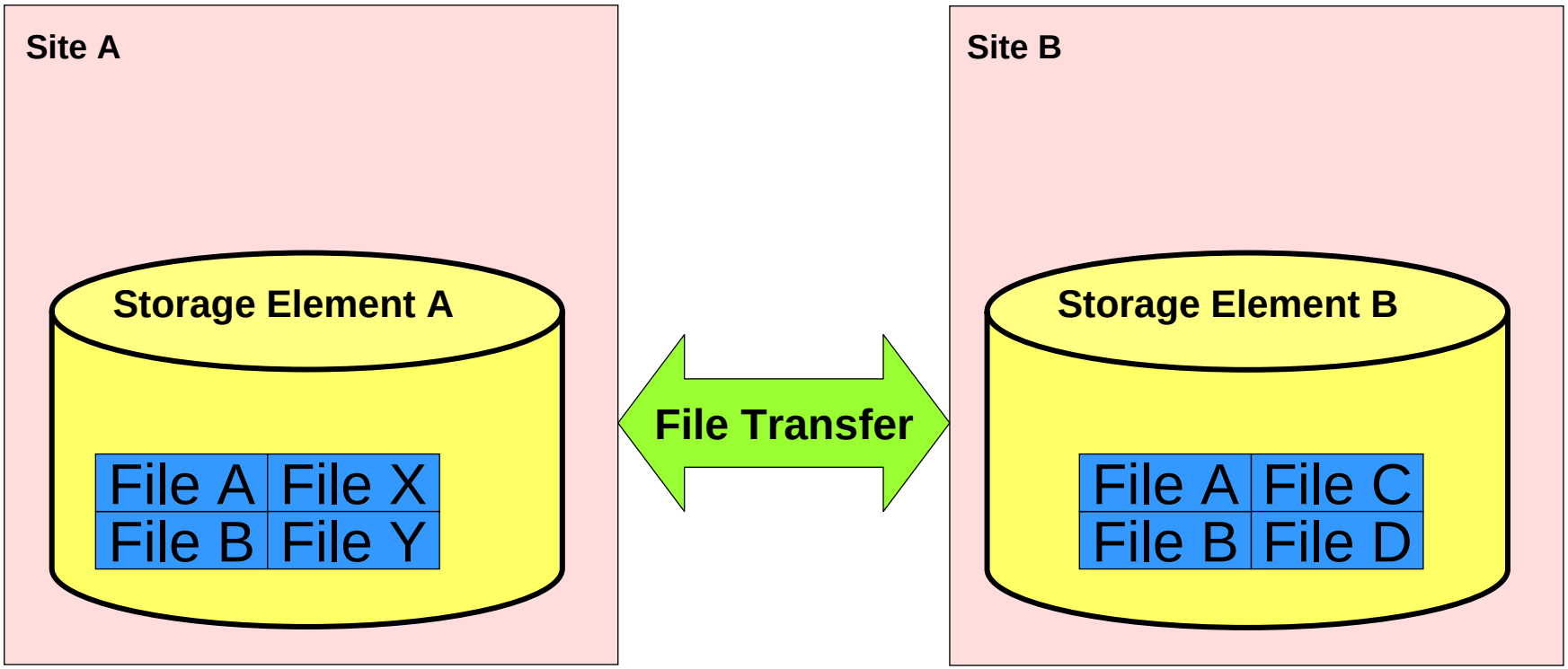
Selected Related Solutions

File Management Motivation



File Management Motivation

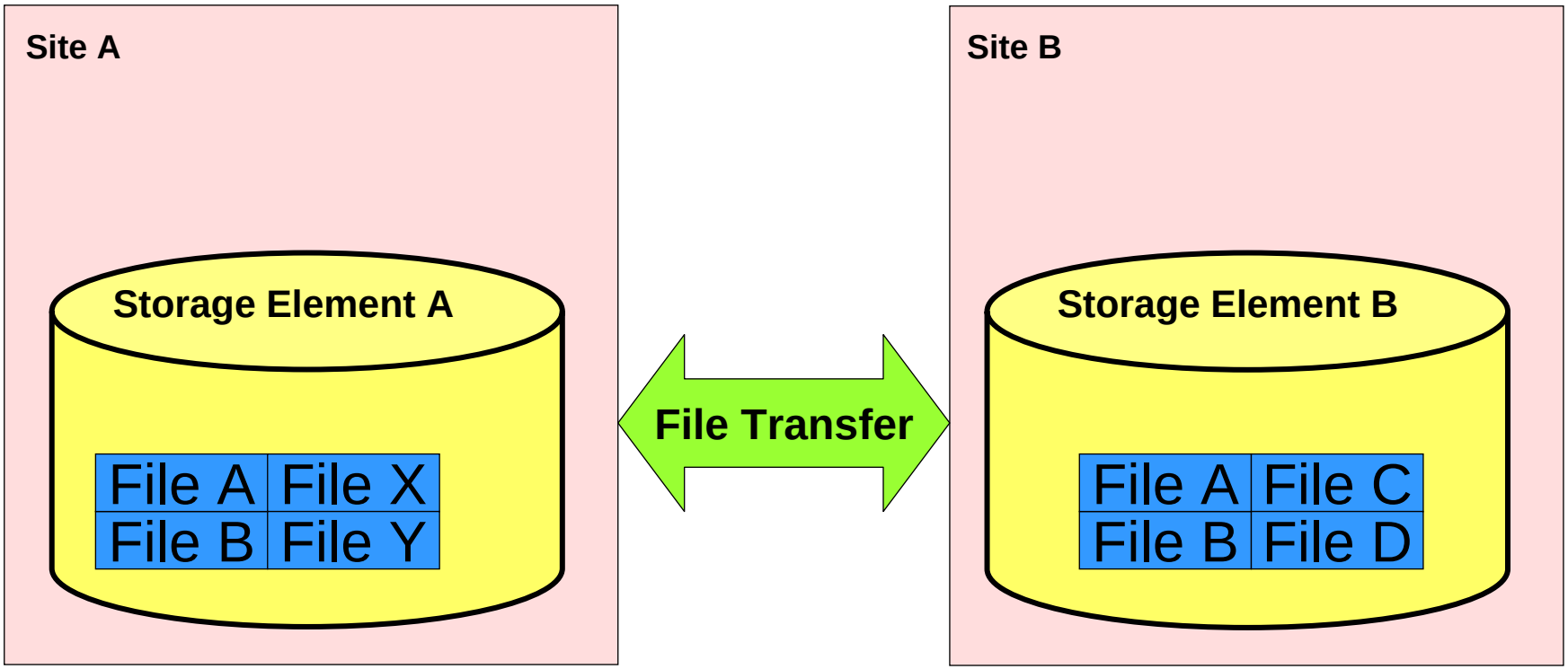
Replica Catalog:
Map Logical to Site files



File Management Motivation

Replica Catalog:
Map Logical to Site files

Replica Selection:
Get ‘best’ file

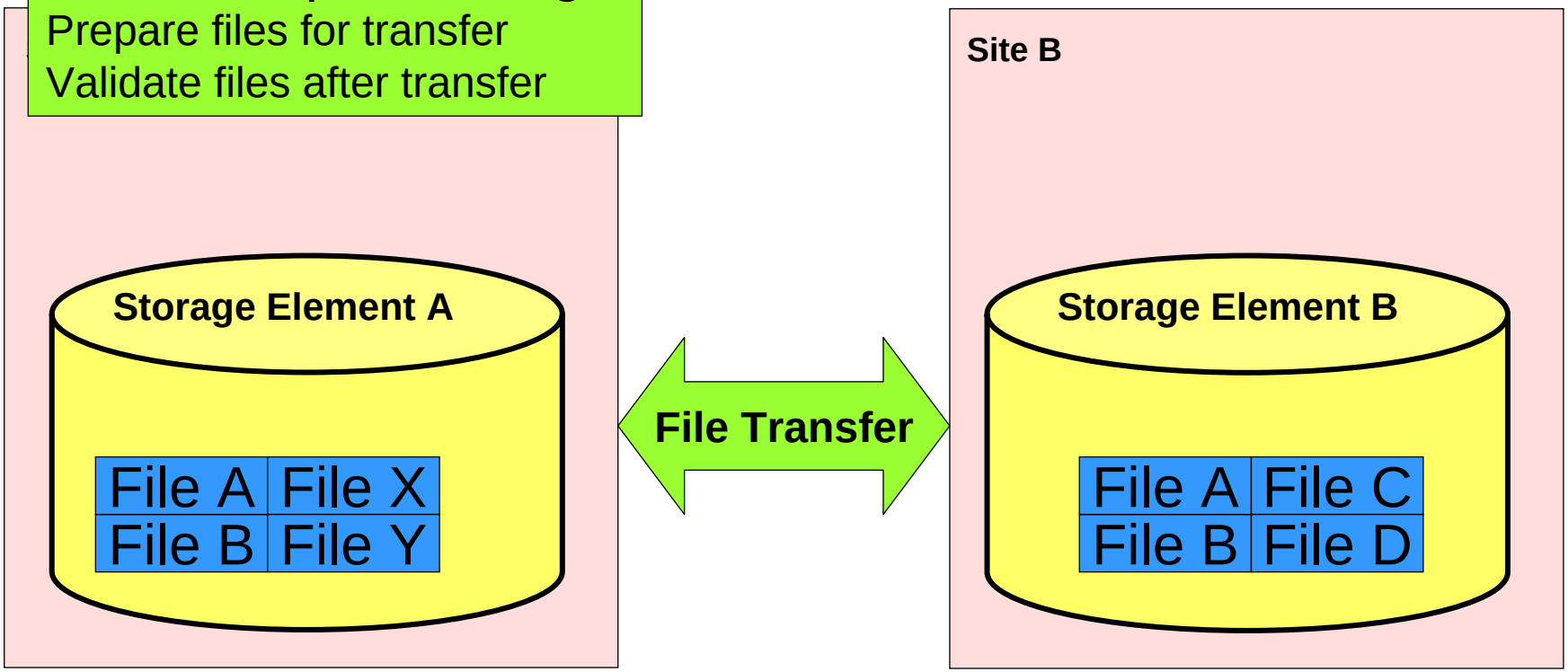


File Management Motivation

Replica Catalog:
Map Logical to Site files

Replica Selection:
Get ‘best’ file

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



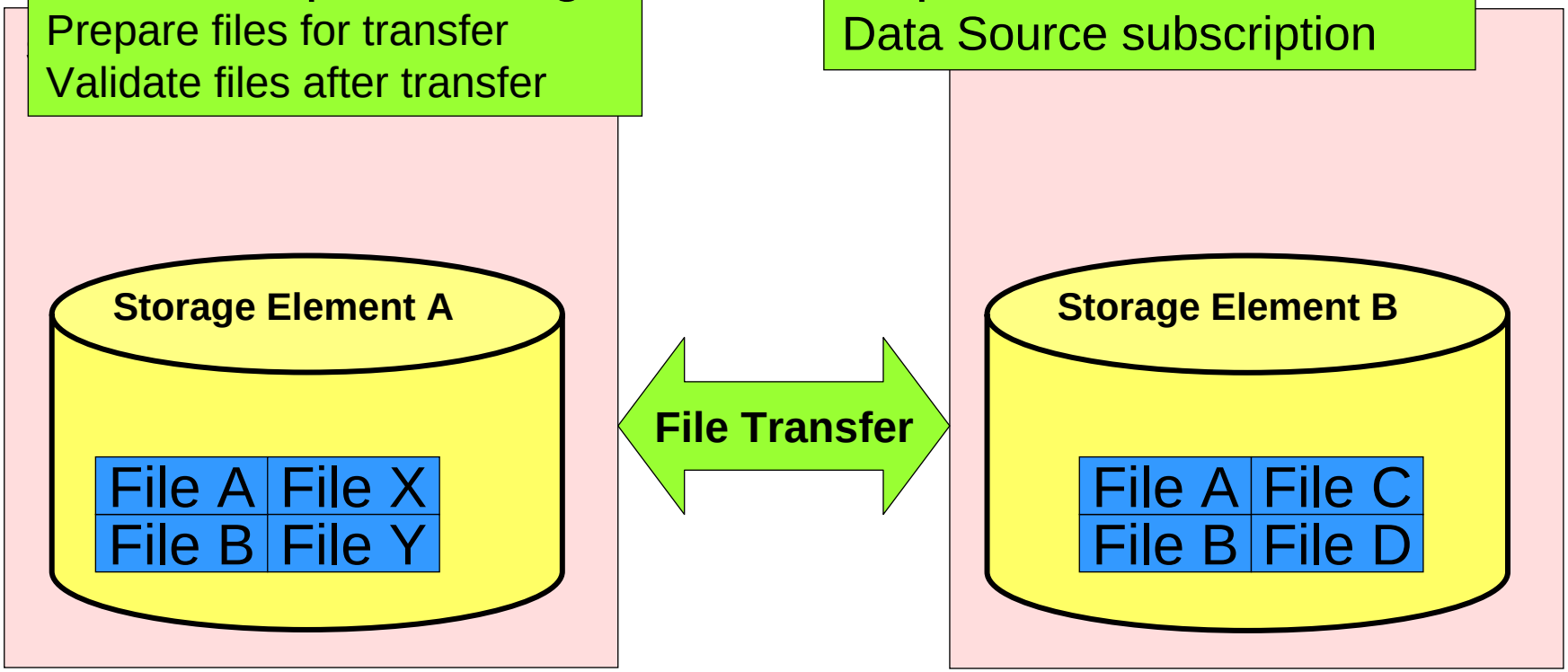
File Management Motivation

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

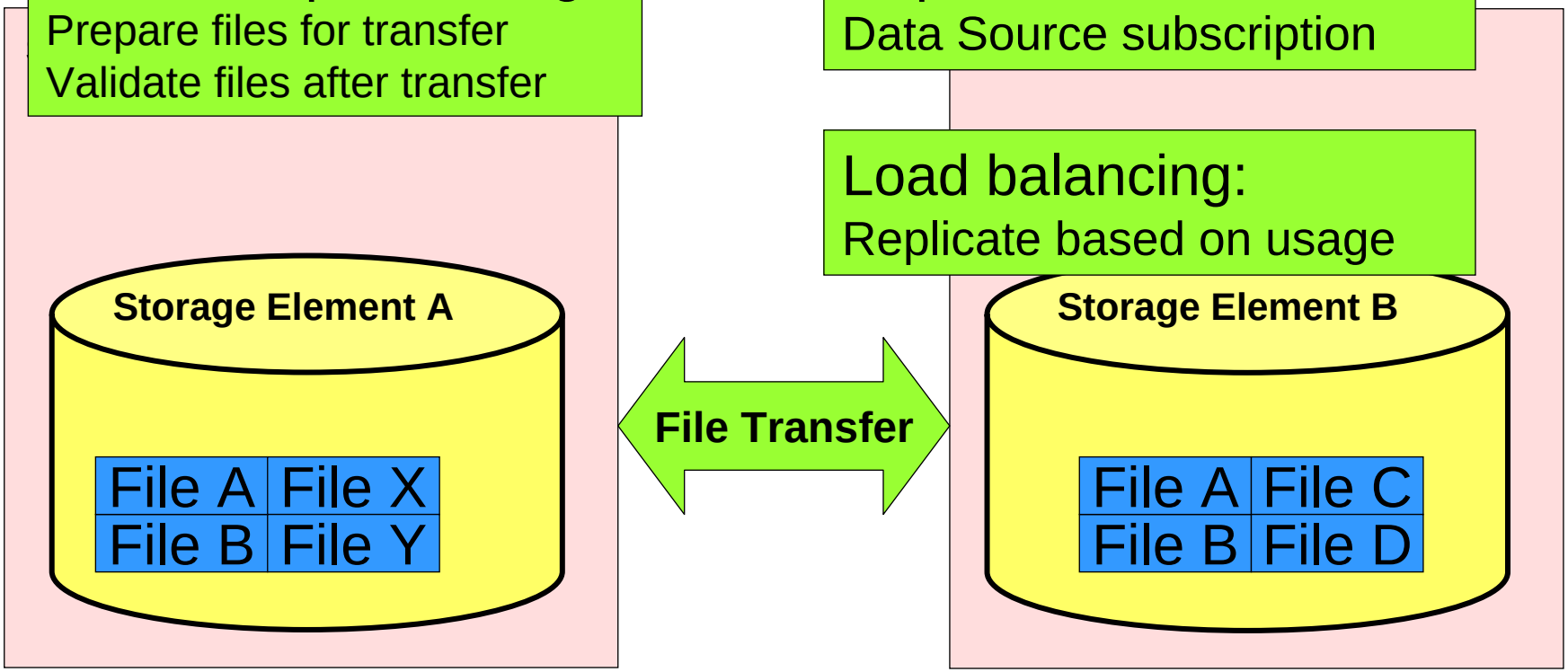
Replica Catalog:
Map Logical to Site files

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

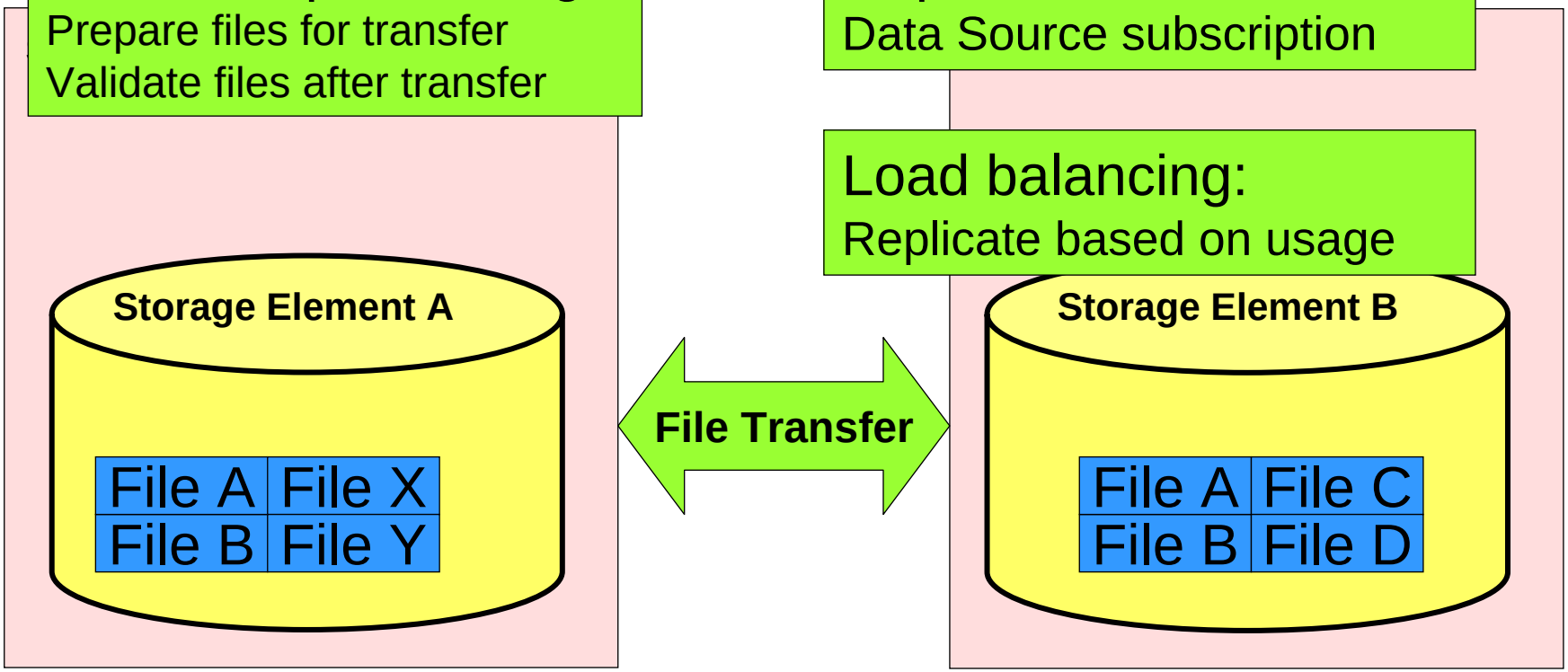
Replica Catalog:
Map Logical to Site files

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

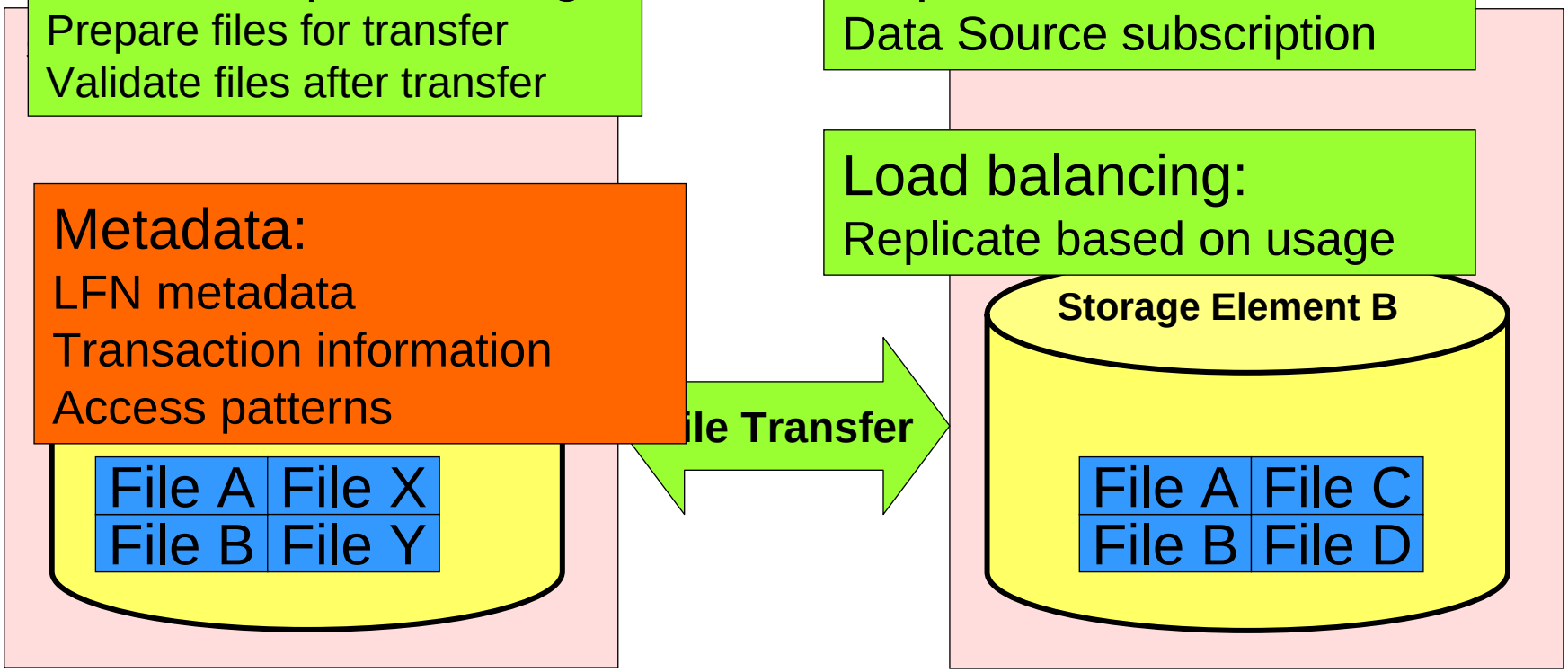
Replica Selection:
Get ‘best’ file

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]

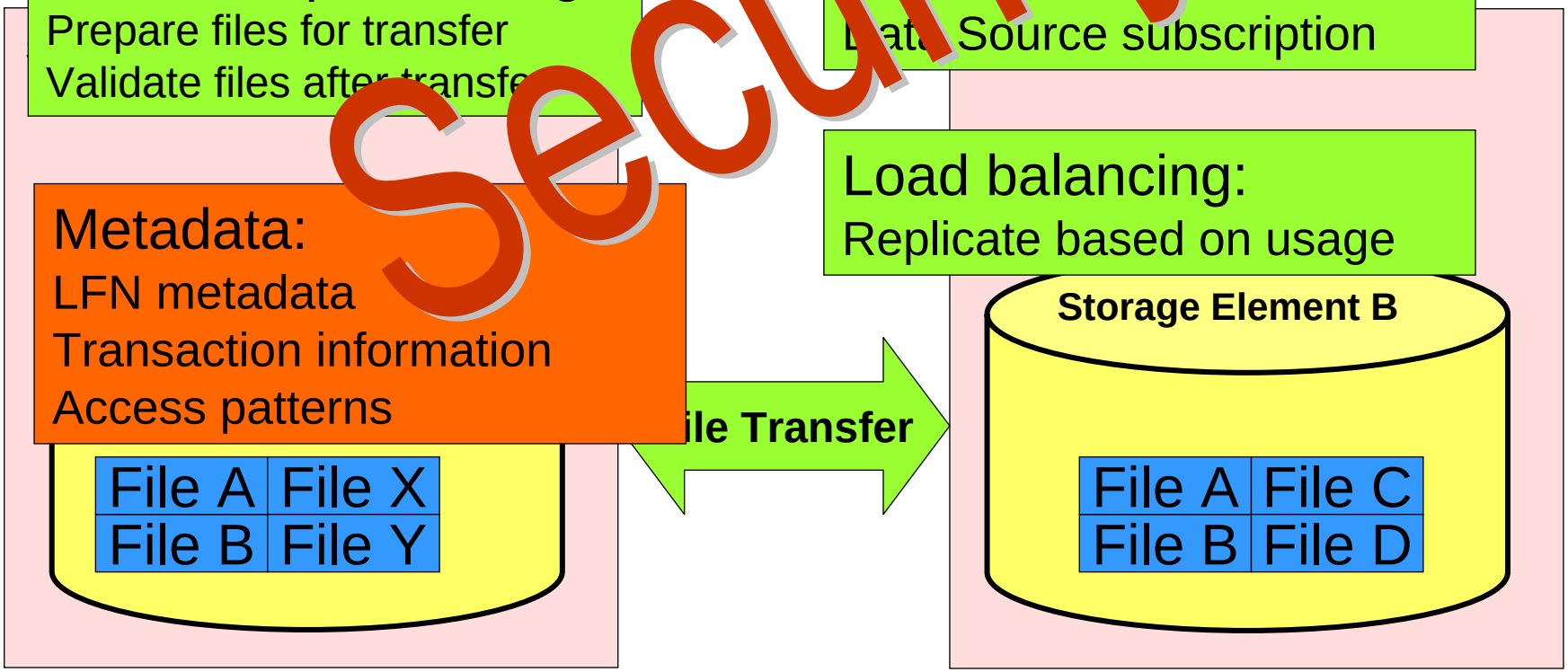
Pre- Post-processing:
Prepare files for transfer
Validate files after transfe[illegible]

Replication Automation:
Data Source subscription

Security

Metadata:
LFN metadata
Transaction information
Access patterns

Load balancing:
Replicate based on usage



4 Basic Categories

- (Grid) Storage Systems
- Data Transfer
- Data Cataloging / Metadata Management
- Replica Management

- General:
  - Simulation of all the components

Data Management Overview

- Discuss the 4 basic categories presented before
- Focus on data management experience in EGEE, EDG and closely related projects
- Give examples in each of the categories
  - Most of them are based on state-of-the-art systems that are currently used with applications mainly in EGEE and related projects
- Related work in data management and solutions that are not directly included/involved in current EGEE-0 architecture

(Grid) Storage Systems

Data Transfer

Data Cataloging / Metadata Management

Replica Management

Selected Related Solutions

Data Storage

Interface to data

software

Data Structure

Database Management System

General Storage Software

etc.

File system

hardware

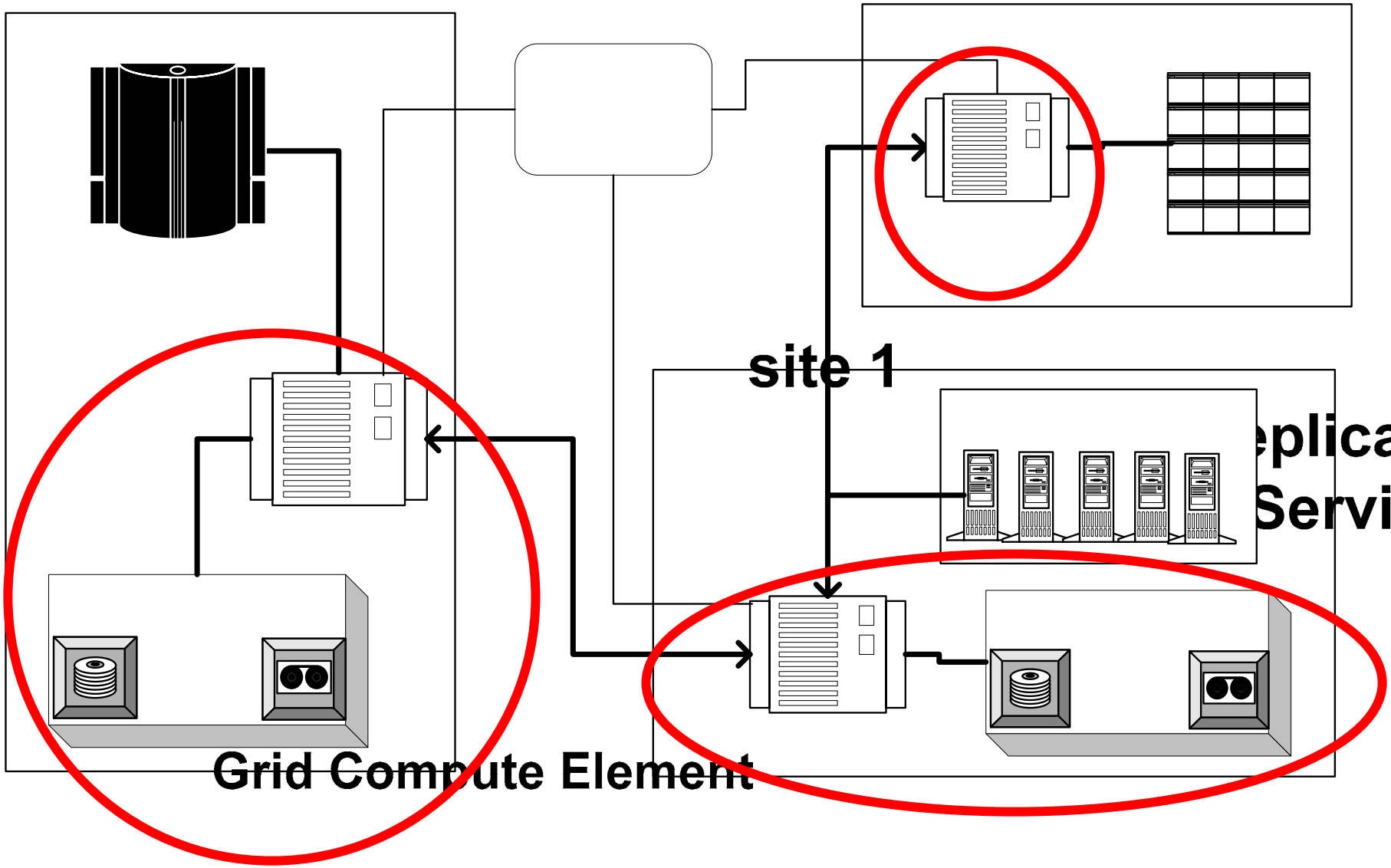
Primary Storage (main memory)

Secondary Storage (hard disk)

Tertiary Storage (tape, optical disks etc.)

Access latency increases

Grid Storage Devices

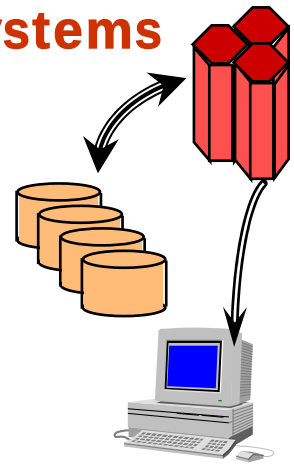


Grid Storage Requirements

- Manage local storage and interface to Mass Storage Systems like
  - HPSS, CASTOR, DiskeXtender (UNITREE), ...
- Provide a unique interface
- Support basic file transfer protocols
  - GridFTP, FTP, ...

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



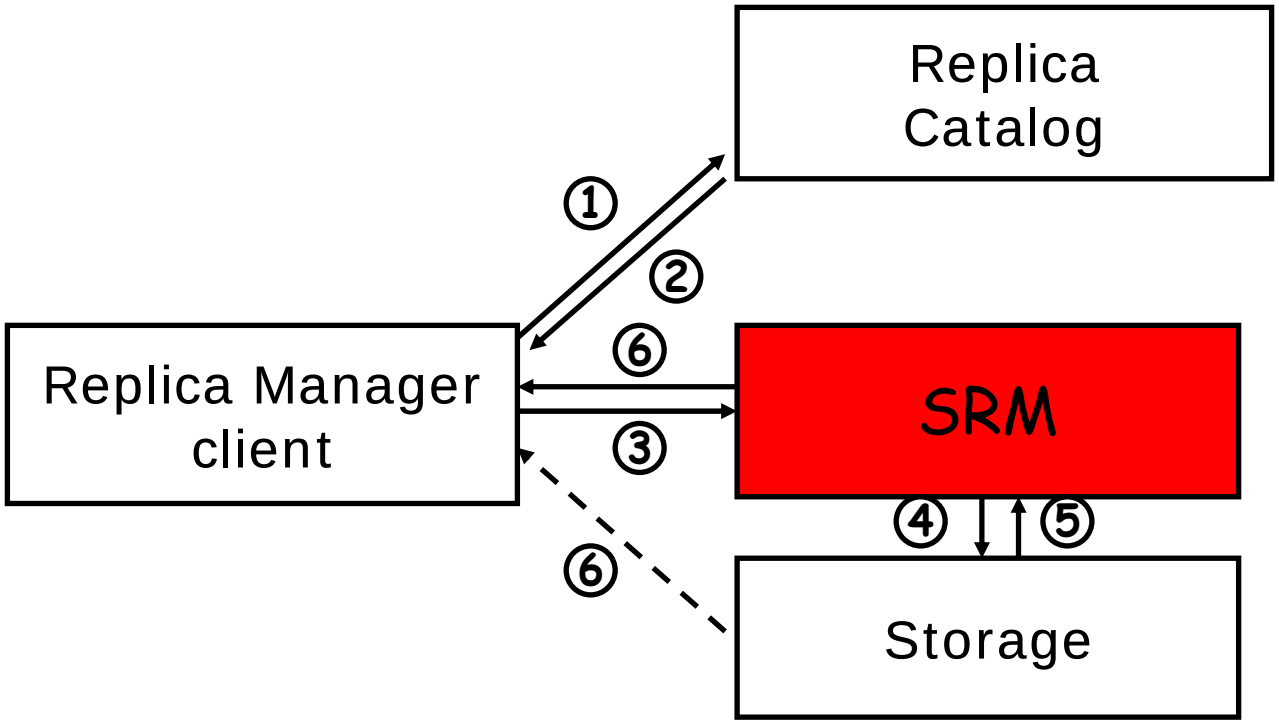
Storage Resource Management (2)

- Original SRM design : LBL, JNL, FNAL, CERN
  - Research Group in GGF created (http://sdm.lbl.gov/gsm/)
- 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

Storage Resource Management (3)

- SRM gives you a consistent interface to mass storage regardless if data is stored on secondary or tertiary storage
- Can be regarded as “low level” file assess similar to FTP
  - GET/PUT file
- Standard does not include the access to the “objects” in the file
  - POSIX file system semantics like seek/read/write are not supported
  - Need to use additional file I/O library to access files in the storage system (e.g. GFAL see later)
- Advantage over POSIX: file pinning, caching, reservation

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

SRM Interface Details

- Current version of specification: 2.1

- Functions for
  - Space Management
    - Reserve, release, update, …
  - Permission
  - Directory
    - mkdir, rmdir, rm, ls, mv
  - Transfer
    - prepareToPut, prepareToGet, copy
    - statusOfGetRequest, statusOfPutRequest
    - abortRequest, suspendRequest, …

(Grid) Storage Systems

Data Transfer

Data Cataloging / Metadata Management

Replica Management

Selected Related Solutions

Data Transfer Requirements

- Here we consider only file level granularity
  - No object streaming etc.
- Secure and efficient point-to-point file transfer over Wide Area Network links
- Needs to interact with existing Grid Security Infrastructure (GSI)
- Utilise network bandwidth
  - “Optimal” file transfer in close connection with network optimisation

GridFTP

- Data transfer and access protocol for **secure and efficient** data movement
- Standardised in the Global Grid Forum
- **extends** the standard **FTP** protocol
  - Public-key-based Grid Security Infrastructure (GSI) or Kerberos support (both accessible via GSS-API
  - Third-party control of data transfer
  - Parallel data transfer
  - Striped data transfer Partial file transfer
  - Automatic negotiation of TCP buffer/window sizes
  - Support for reliable and restartable data transfer
  - Integrated instrumentation, for monitoring ongoing transfer performance

GridFTP Implementations

- Original implementation comes from Globus Alliance
  - Part of Globus Toolkit
- Another implementation (client and server) from Fermilab
  - Part of GGF working group and proposed several extensions to GridFTP
- Several storage systems provide GridFTP interfaces:
  - Castor
  - EDG's SRM implementation
  - etc.
- GridFTP is probably the most widely accepted protocol (next to GSI) that results from GGF work

General Optimisation Considerations

- What do we want to optimise?

- Performance of single user?
  - Transfer one file as quickly as possible
  - Utilise the us much as possible of network link

- Throughput of several users?
  - Is it “fair” to optimise performance of single user and then neglect other concurrent users?
  - Allow for several users at the same time and optimise aggregated throughput

(Grid) Storage Systems

Data Transfer

Data Cataloging / Metadata Management

Replica Management

Selected Related Solutions

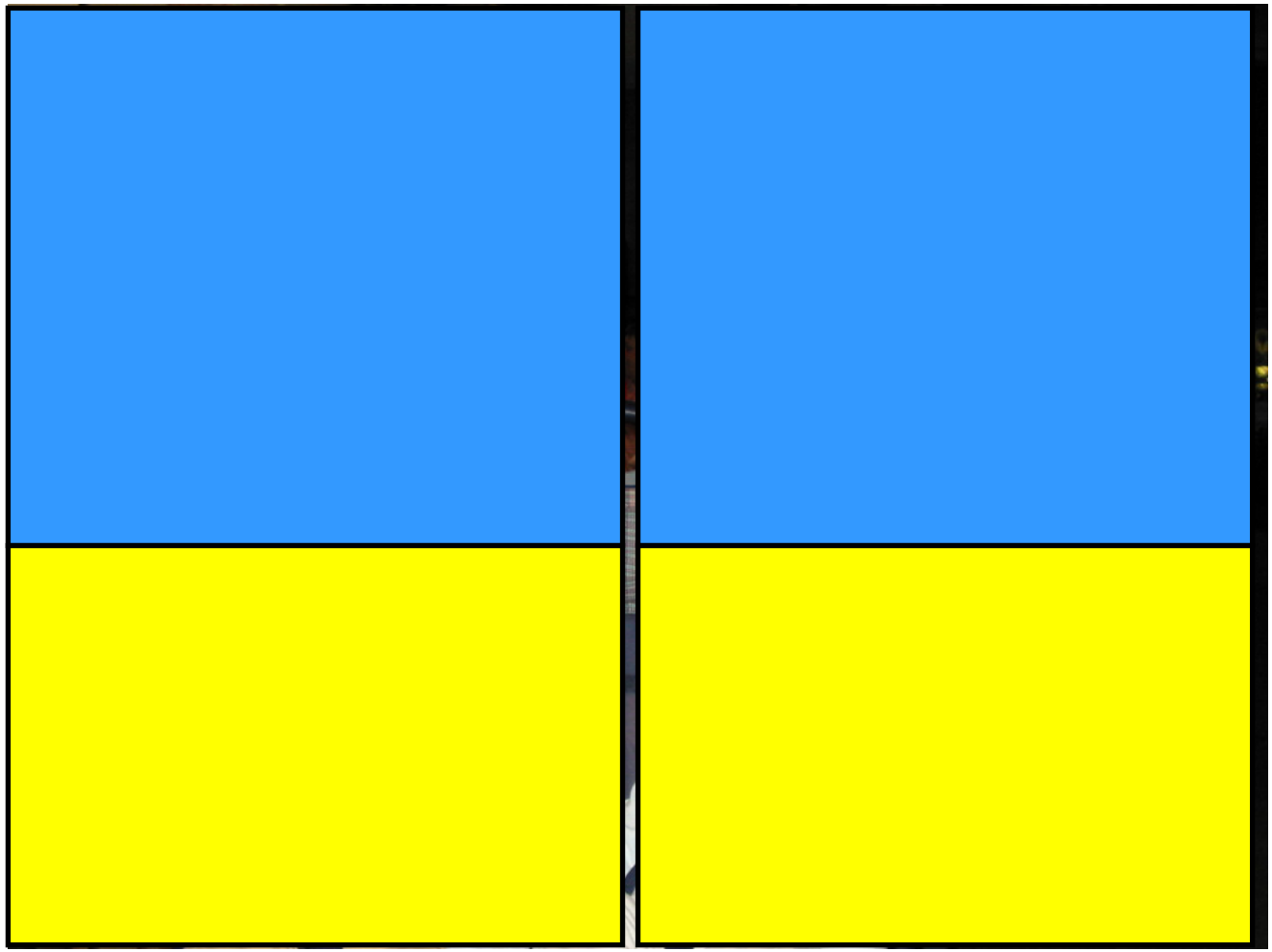
Cataloging Requirements

- Replication is well known in distributed systems and important for Data Grids
- Identical replicas need to be identified and located
- Replica location information can be regarded meta-data management
- Potentially, millions of files need to be registered and located
- Requirement for performance as well as distributed architecture to prevent single-point of failure

What is a Replica ?

Physical Name on Storage

Logical File Name (LFN)



Stockie

LollyPop

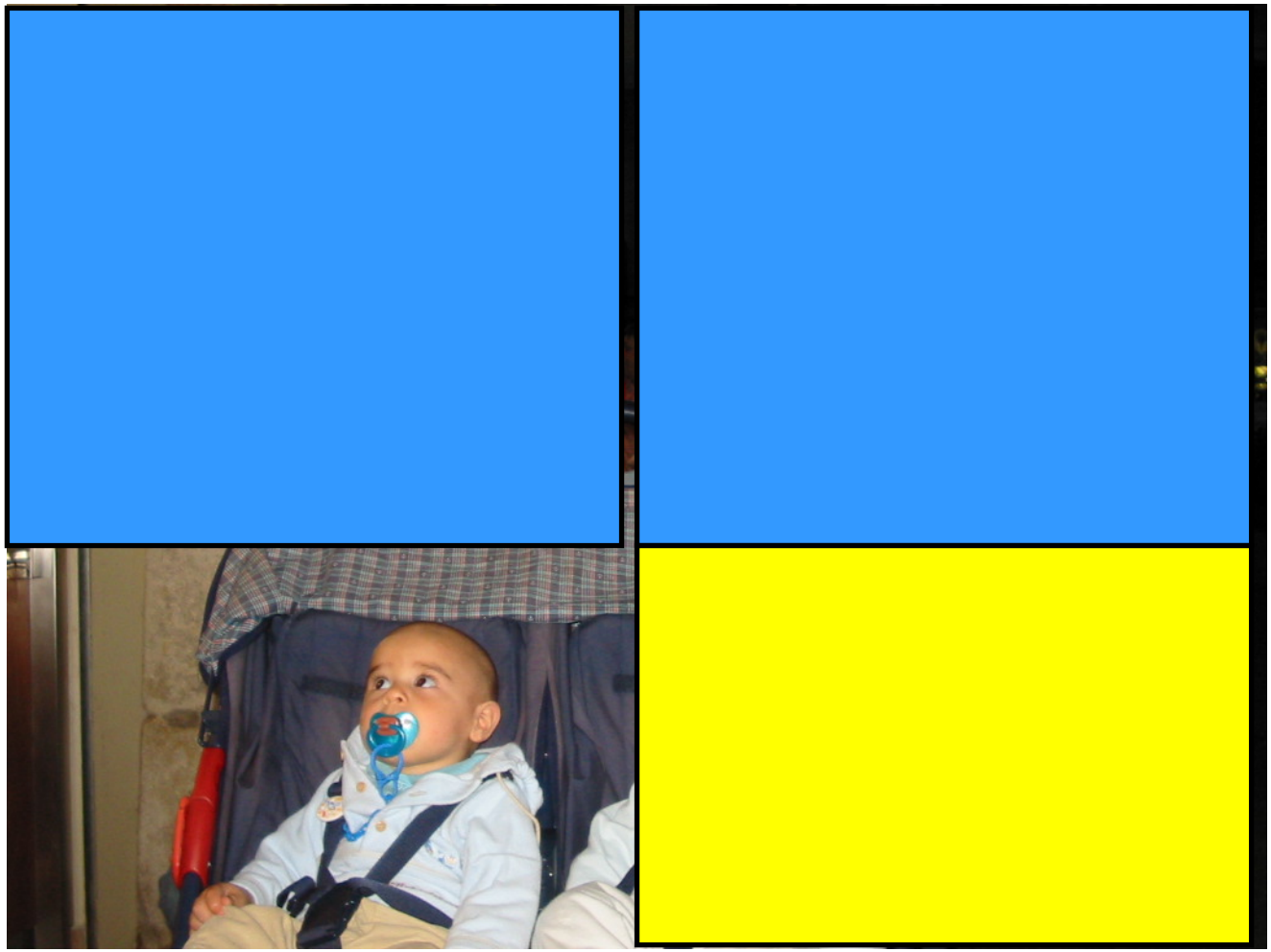
What is a Replica ?

Physical Name on Storage

Logical File Name (LFN)

Stockie

LollyPop



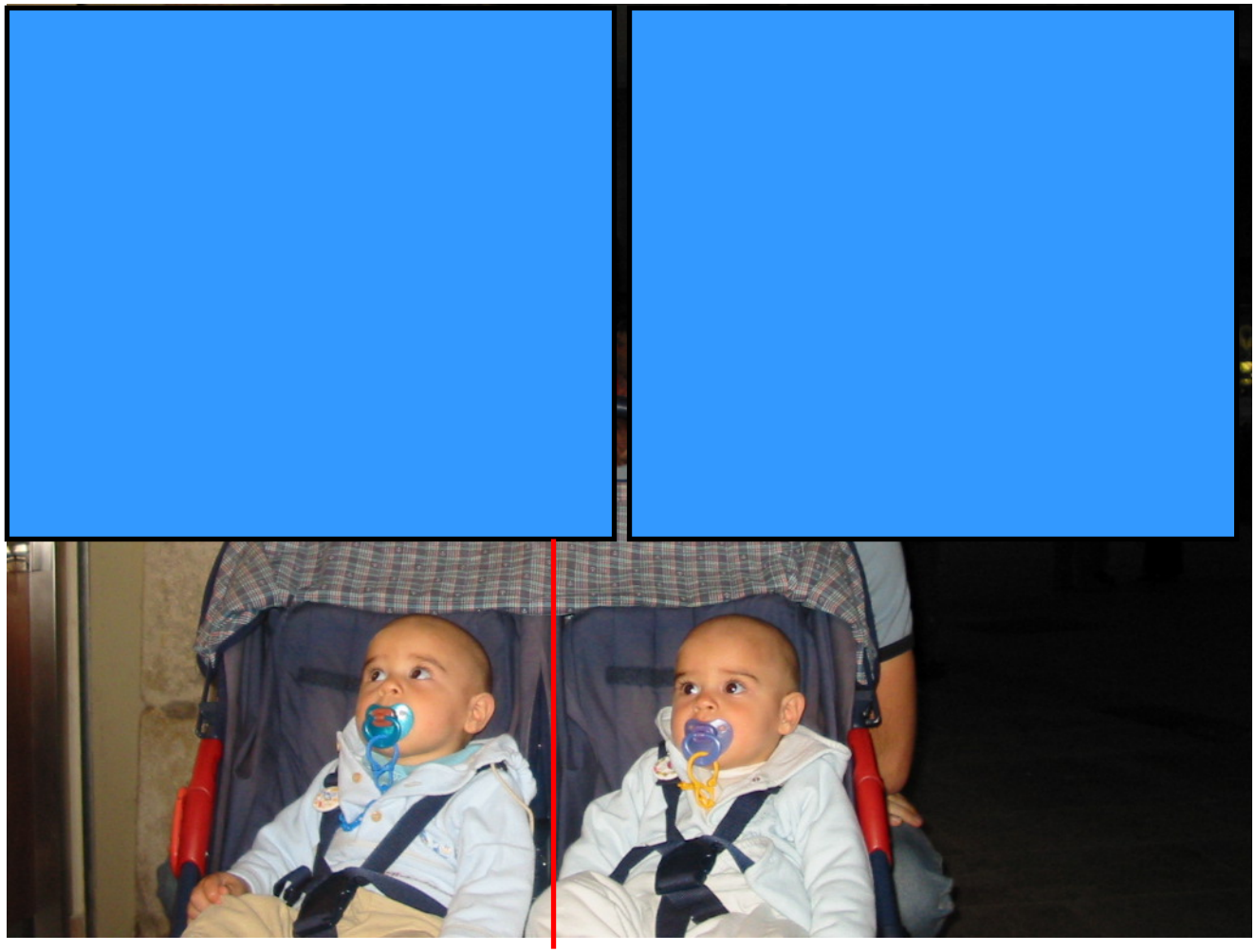
What is a Replica ?

Physical Name on Storage

Logical File Name (LFN)

Stockie

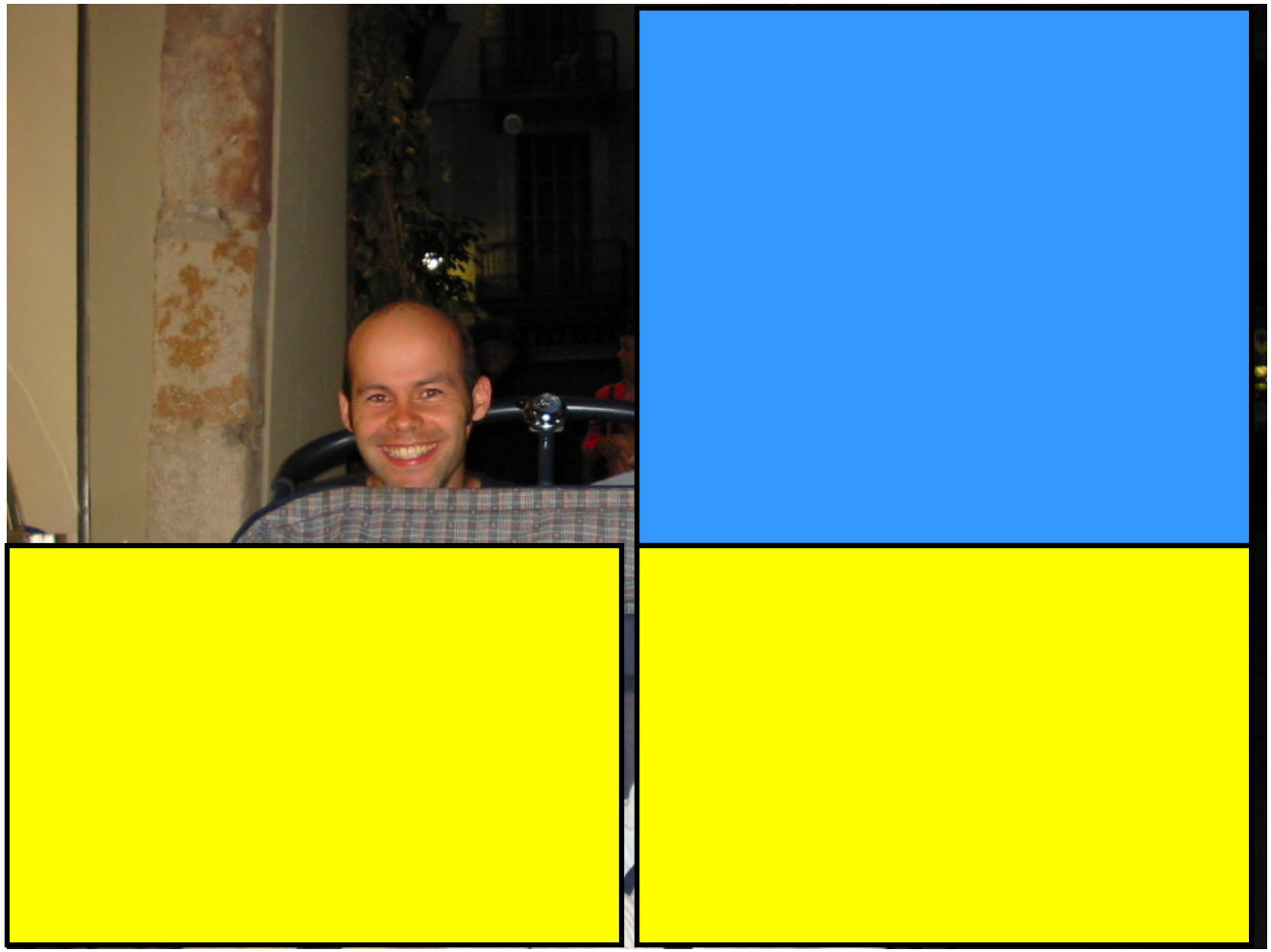
LollyPop



What is a Replica ?

Physical Name on Storage

Logical File Name (LFN)



Stockie

LollyPop

What is a Replica ?

Physical Name on Storage

Logical File Name (LFN)

Stockie

LollyPop

General Naming Conventions

- Require a logical-to-physical filename mapping
- Simplest case if files reside on disk systems
  - Logical File Name (LFN) … logical identifier
  - Physical name on storage … hostname + physical location
- However, once Mass Storage Systems are used, there is another storage hierarchy. In addition, several protocols can be used to access the file.
  - Logical File Name (LFN)

    Any string
  - Storage URL (SURL)

    srm://hostname/name-on-storage
    - contains host information to find physical file
    - Independent of protocol transport protocol
  - Transfer File Name (TFN)

    protocol://hostname/name-on-storage
    - Contains protocol to actually access the file
    - Expressed as a URI

Filename examples

Logical File Name	**my-lfn** **logical-name**
Storage URL	**srm://host1.cern.ch/data/myfile** **srm://example.org/dir1/dir2/dir3/name** **srm://example.org/dir1/dir2/dir3/name2**
Transfer File Names	**gridftp://host1.cern.ch/data/myfile** **gsiftp://host2.cern.ch/data/myfile** **http://example.org/data/filename**

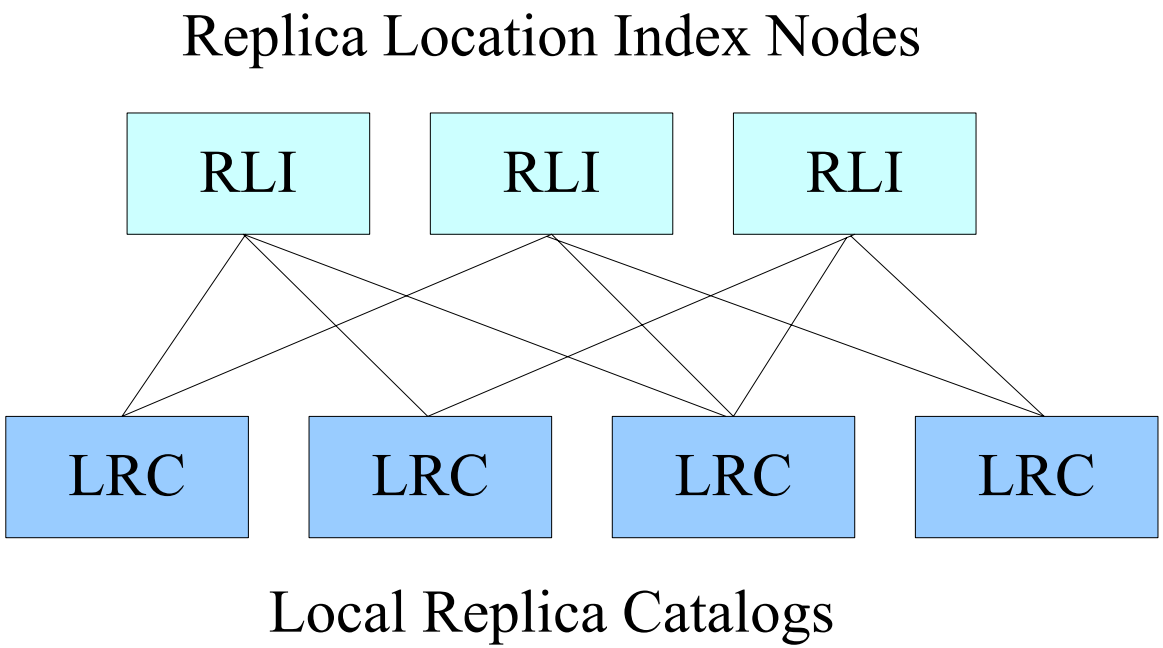
Replica Location Service (RLS) - 1

- **A Replica Location Service (RLS)** is a distributed registry service that records the locations of data copies and allows discovery of replicas

- Maintains mappings between logical identifiers and target names
  - Physical targets: Map to exact locations of replicated data
  - Logical targets: Map to another layer of logical names, allowing storage systems to move data without informing the RLS

- RLS was designed and implemented in a collaboration between the DataGrid project (WP2) and the Globus Alliance
  - First implementation was available in the Globus Toolkit 2
  - Later also EDG made an implementation

Replica Location Service (RLS) - 2



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

A Flexible RLS Framework

Five elements:

1. Consistent Local State:
   - Records mappings between logical names and target names and answers queries
2. Global State with relaxed consistency:
   - Global index supports discovery of replicas at multiple sites; relaxed consistency
3. Soft state mechanisms for maintaining global state:
   - LRCs send information about their mappings (state) to RLIs using soft state protocols
4. Compression of state updates (optional):
   - reduce communication, CPU and storage overheads
5. Membership service:
   - for location of participating LRCs and RLIs and dealing with changes in membership

RLS – Globus Implementation

- RLS is part part of GT2 and GT3
- Service is implemented in C
  - C and Java clients are available
- Uses Postgress DBMS to store filenames
  - Originally, MySQL but there we licensing issues with MyODBC (i.e. the C implementation of MySQL's ODBC)
- GSI authentication

- For details: Ann Chervenak, Ewa Deelman, Leanne Guy, Ian Foster, Wolfgang Hoschek, Adriana Iamnitchi, Carl Kesselman, Peter Kunszt, Matei Ripeanu, Heinz Stockinger,Kurt Stockinger, and Brian Tierney. Giggle: A Framework for Constructing Scalable Replica Location Services. In Proc. of the Int'l. ACM/*IEEE Supercomputing Conference (SC 2002)*, Baltimore, USA, November 2002.

(Grid) Storage Systems

Data Transfer

Data Cataloging / Metadata Management

EDG's additions to RLS

Replica Management

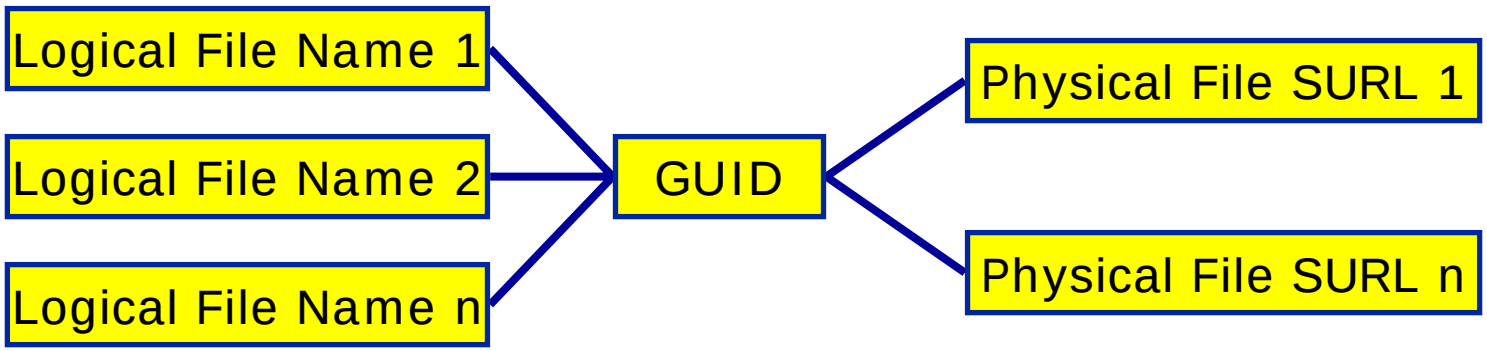
Selected Related Solutions

RLS – EDG Implementation

- EDG provides another implementation of the RLS
- Design is identical but differences in implementation
  - Uses web service approach
    - Protocol is not interoperable with Globus implementation
    - Protocol convergence will be achieved once both implementations use OGSA or "follow-up" (WSRF) approach
- Additional service (Replica Metadata Catalogue, RMC) added
  - Stores additional metadata for files
- Slight change in naming conventions
  - GUID (Globally Unique Identifier) added which basically replaces the LFN
  - LFN is used as an alias for the GUID
  - Allows for N:M relations LFN – StorageURL
  - Replica Manager client (explained later) provides global view

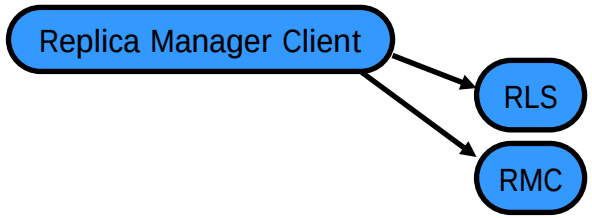
Naming Conventions in EDG

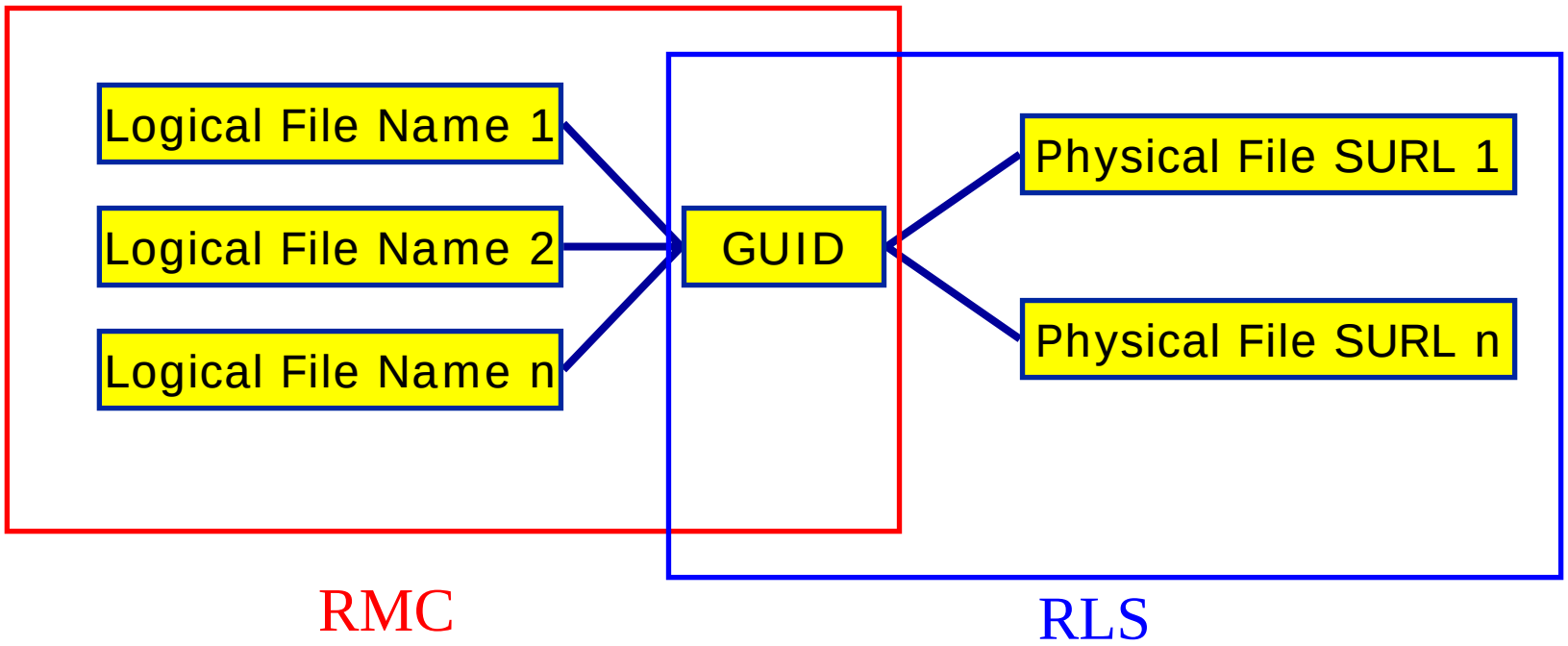
- Logical File Name (LFN)
  - An alias created by a user to refer to some item of data e.g. “lfn:cms/20030203/run2/track1”
- Site URL (SURL) (or Physical Name (PFN))
  - The location of an actual piece of data on a storage system e.g. “srm://pcrd24.cern.ch/flatfiles/cms/output10_1”
- Globally Unique 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

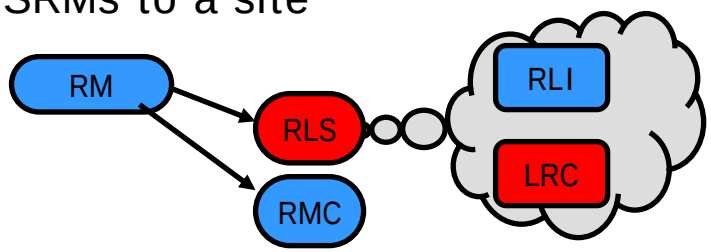




RLS Components (1)

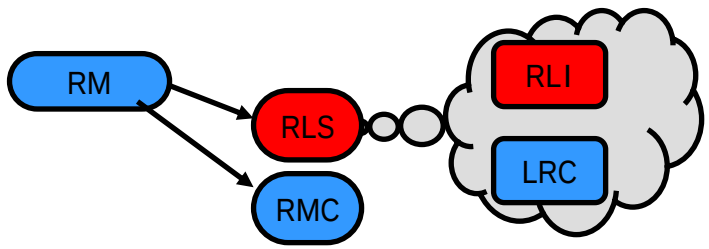
Local Replica Catalog (LRC)

- 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

(Grid) Storage Systems

Data Transfer

Data Cataloging / Metadata Management

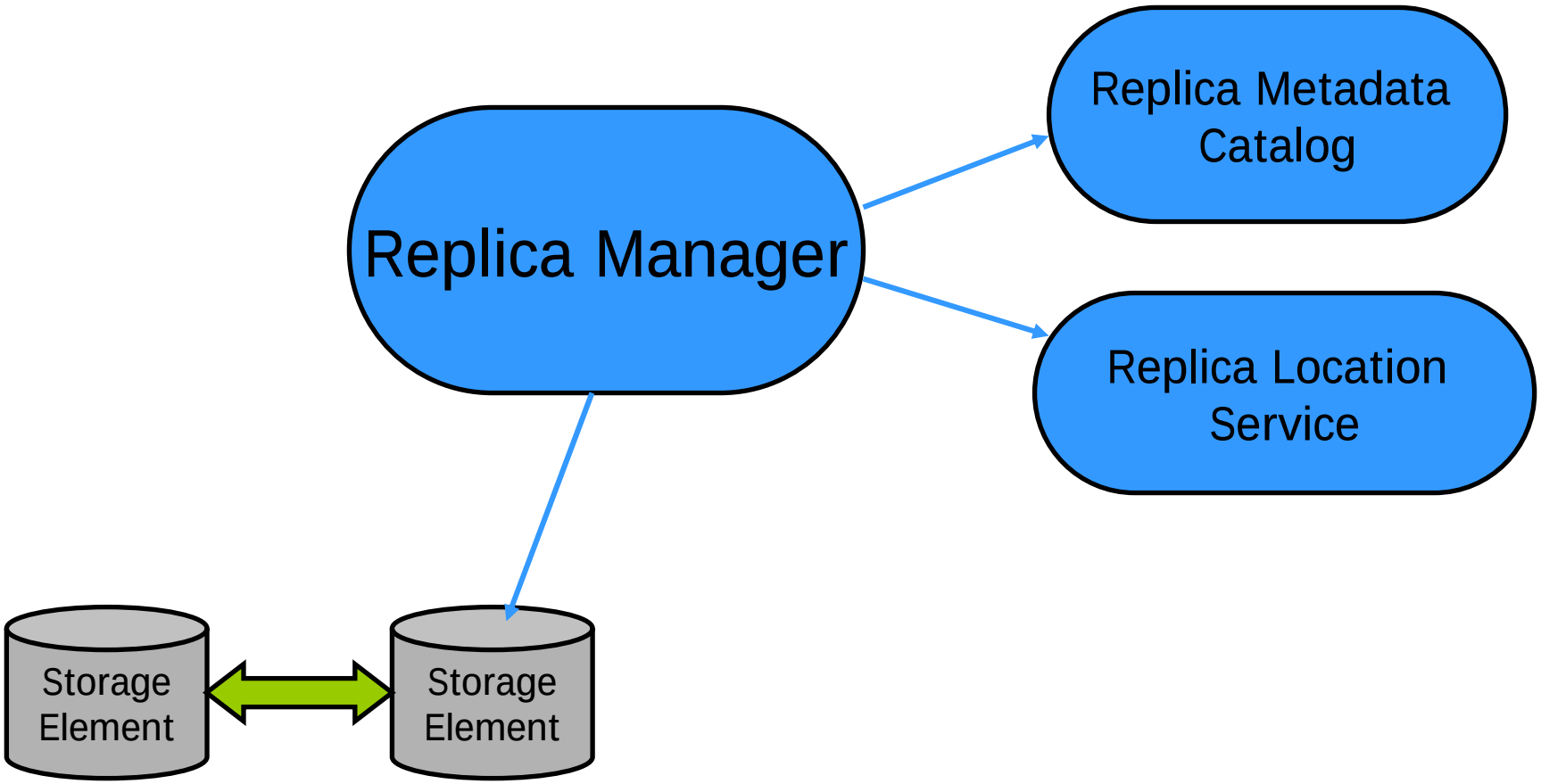
Replica Management

Selected Related Solutions

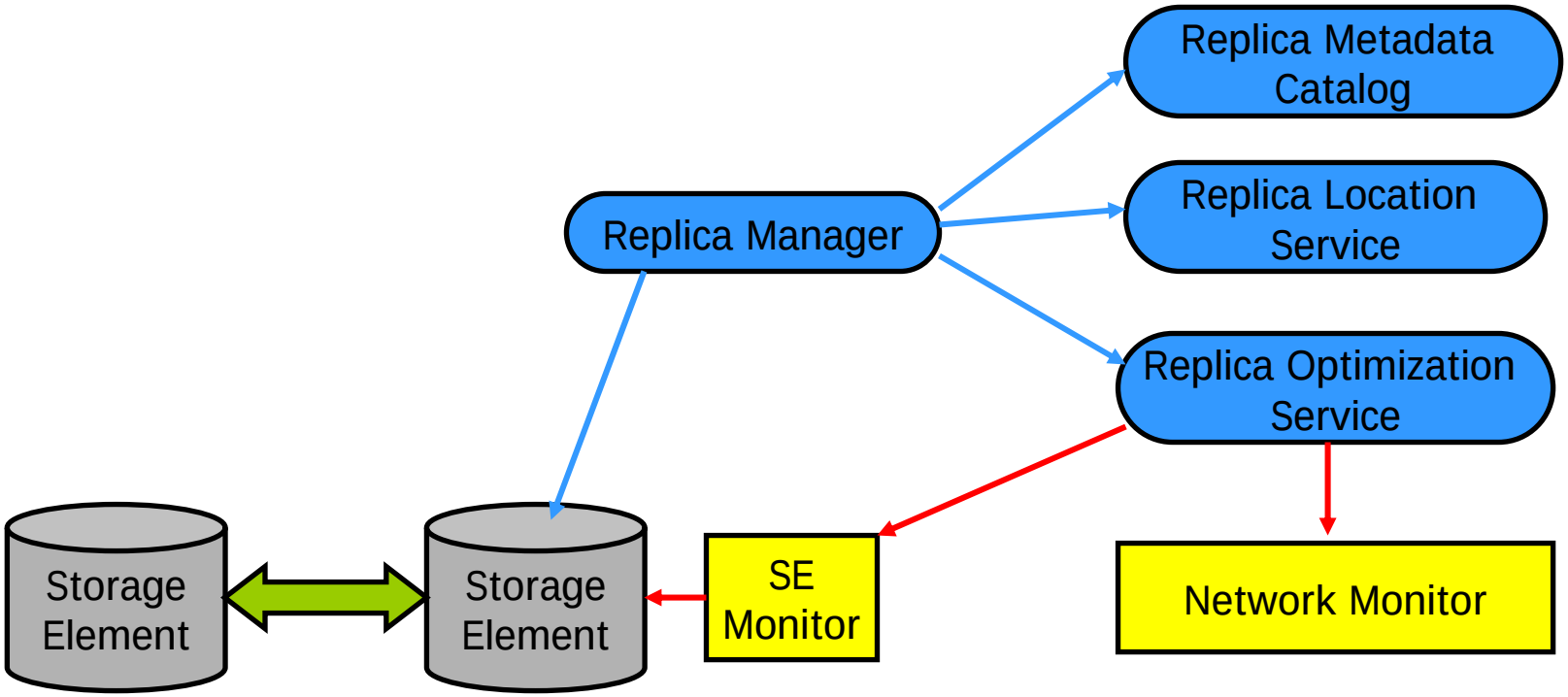
Replica Management

- Combine file transfer and cataloging as an atomic transaction
  - Keep consistency with catalogues and storage systems
  - Optimise replica selection (i.e. choose best locations to get file if several replicas are availble)
- “simpler” for read-only files
- “More difficult” if file replicas can be changed
  - Need to keep replicas in synch

Replication Services: Basic Functionality



Higher Level Replication Services



Replica Selection Process

- Assumption: files are partially replicated to several sites
  - User wants to access file as quickly as possible independent of location
- Goal: selecting a single “best” replica
- Response time for accessing a file locally is minimal
- Access time also includes the data transfer time from the remote to the local site
- Two main performance parameters: network and storage

$$\min(file\_transfer_i(site_{local}, site_{remote}i))$$

$$file\_transfer_i = access\_cost_{network} + access\_cost_{storage}$$

Access Cost for Hierarchical Storage Manager (HSM) Systems

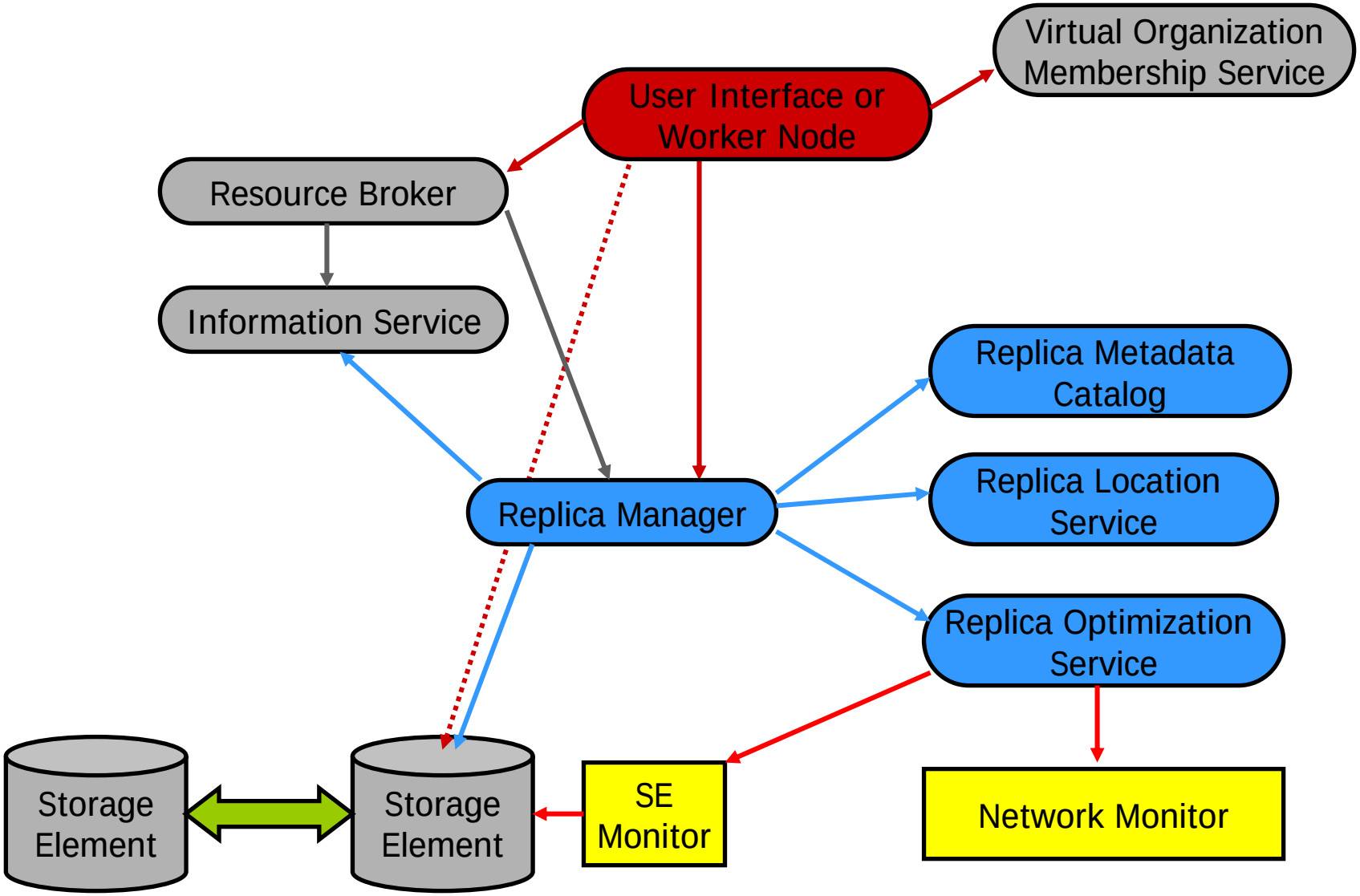
- Depends on:
  - current load of HSM system
  - number of available drives
  - performance characteristics of the drives
  - data localization (cache, tape)
  - data compression rate
- **$access\_cost_{storage} = time_{latency} + time_{transfer}$**

$$time_{latency} = t_w + t_u + t_m + t_p + t_t + t_d$$

$$time_{transfer} = size_{file} / transfer\_rate_{cache}$$

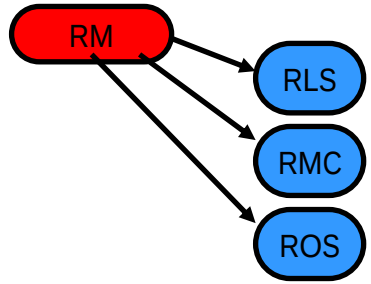
- **W**aiting for resources
- **U**nloading idle tape
- **M**ounting required tape
- **P**ositioning
- **T**ransfer from tape to disk cache
- **D**isk cache latency

Interactions with other Grid components



User Interfaces for Replica 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.

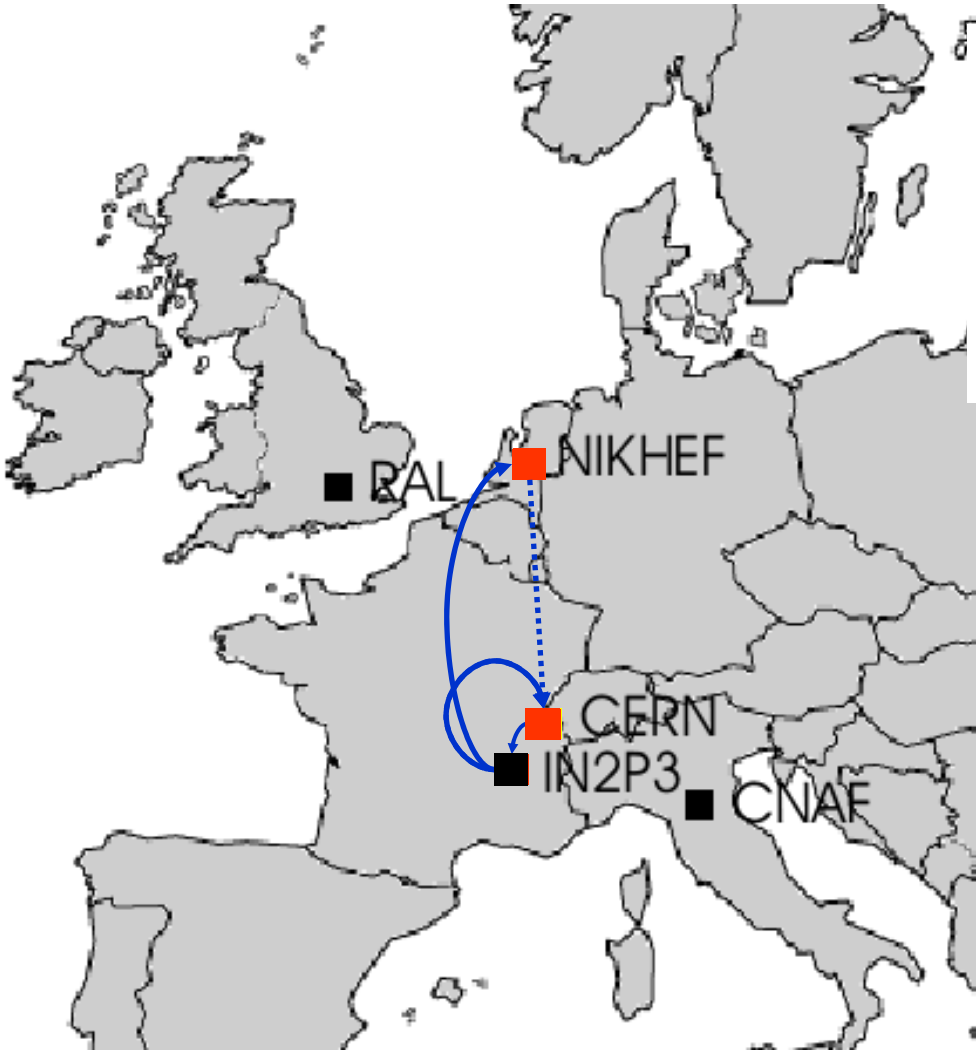
The Replica Manager Interface – *Optimization Commands*

- 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

(Grid) Storage Systems

Data Transfer

Data Cataloging / Metadata Management

Replica Management

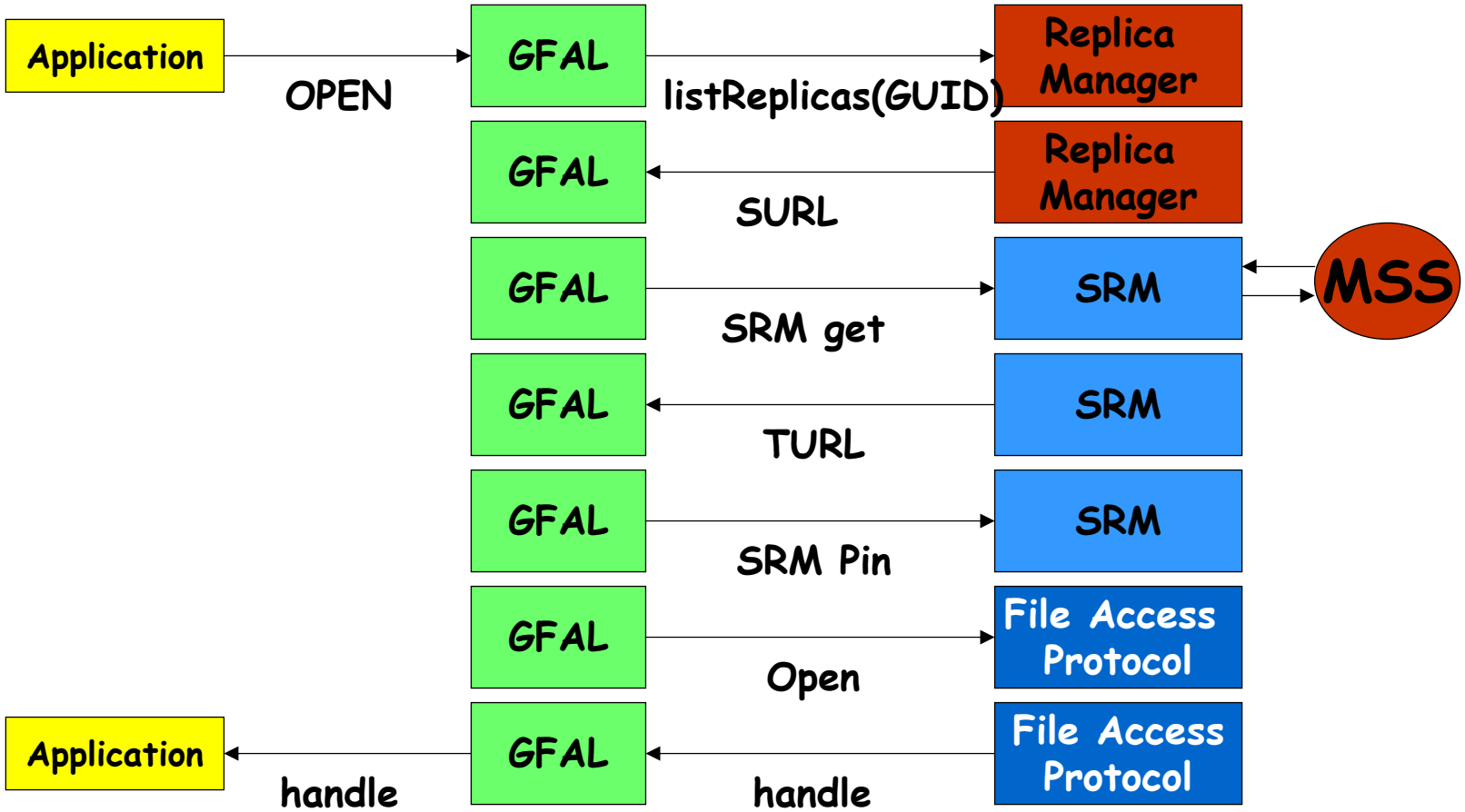
Selected Related Solutions

Grid File Access Library

Grid File Access Library (GFAL) - 1

- Since SRM does not support read/write POSIX functionality, an “extension” has been provided by LCG project
- Hides the orchestration of interactions between
  - Replica Manager Services
  - SRM
  - and the file access mechanism between Worker Nodes and Storage Elements
- Presents a POSIX interface for normal file operations (Open/Seek/Read/Write/Close...)
- It assumes local accesses although the architecture permits local and wide-area access

Grid File Access Library (GFAL) - 2



Conclusion

- Several Grid data management are solutions already in place and in use
- Still more work is required to allow for transparent access to data distributed and replicated in a Grid
- Data consistency of replicated data is not yet dealt with sufficiently
  - Some on-going prototypes

Abbreviations

- DAI – Data Access & Integration
- GFAL – Grid File Access Library
- GUID – Globally Unique Identifier
- HSM – Hierarchical Storage Manager
- LFN – Logical File Name
- LRC – Local Replica Catalog
- PFN – Physical File Name
- RLS – Replica Location Service
- RMC – Replica Metadata Catalog
- RLI – Replica Location Index
- ROS – Replica Optimization Service
- SRM – Storage Resource Manager
- TFN – Transfer File Name