

DESY Grid Status



* e-mail: Michael.Ernst@desy.de



Grid Activities at DESY

- DESY Grid Testbed2
- **Enabling Grid for E-Science in Europe (EGEE) (SA1)**
- D-GRID (AK1,2,4,5)
- GHEP
- **International Lattice Data Grid (ILDG)**
- (**Linear Collider Data Grid (LCDG)**)

- SRM/dCache based Storage Element
- Global Catalogs for Metadata and Replica Location Service

<http://grid.desy.de/>



DESY Grid Testbed2 ...

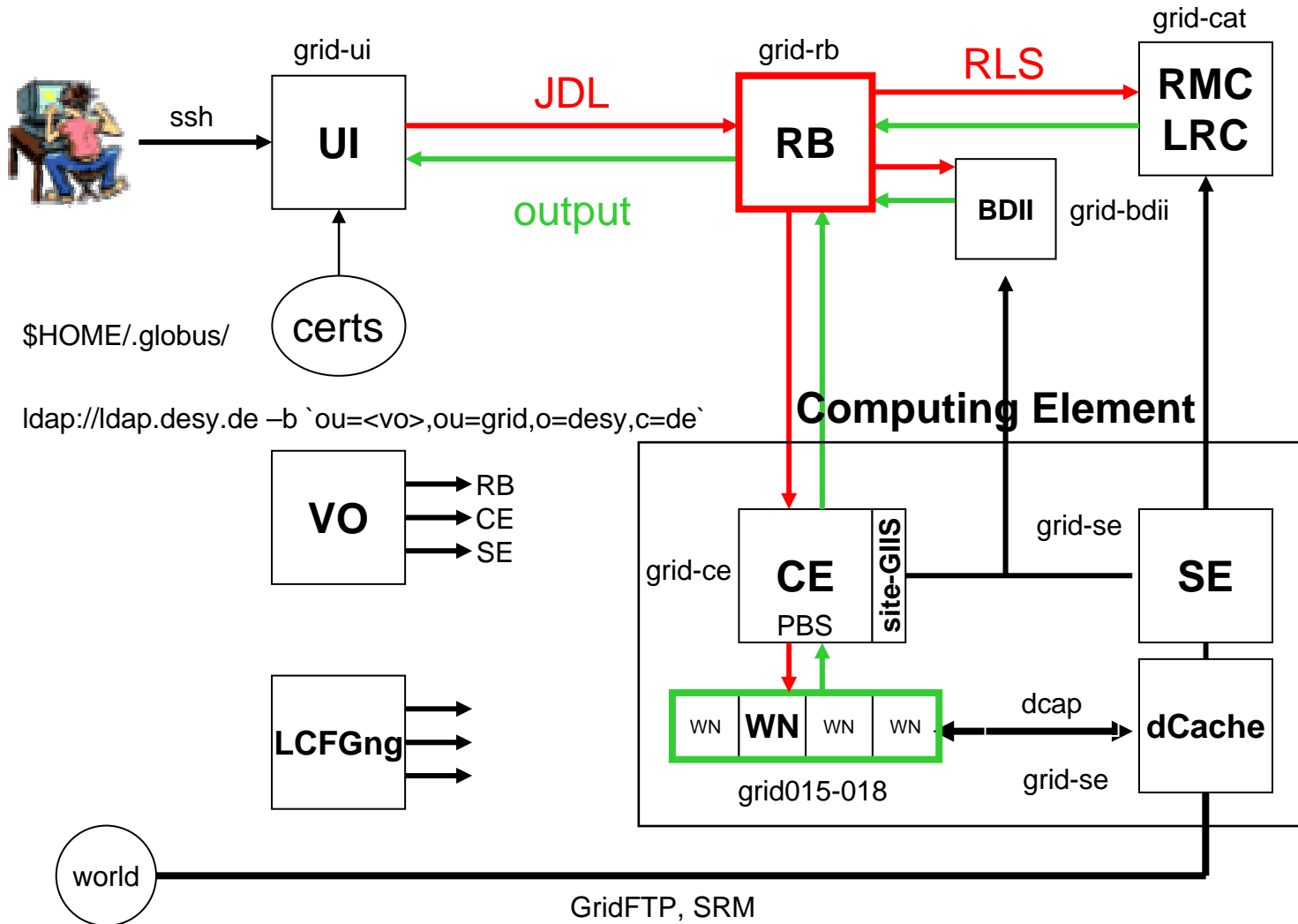
<http://grid.desy.de/>

- Setting up a uniform Grid infrastructure which serves the needs of all DESY experiments and groups („*Federating Resources*“)
- 2-tier model: integration testbed and a production system
- The Production system could be part of GHEP and EGEE
- Installation is LCG-2-based exploiting LCFGng, see: http://grid.desy.de/testbed2/lcg-2_lcfgng-install.html
- DESY maintains VOs for DESY experiments and groups
- Provisioning of resources to VOs is a matter of *policies*
- Shares can be divided by means of the batch system



... DESY Grid Testbed2 ...

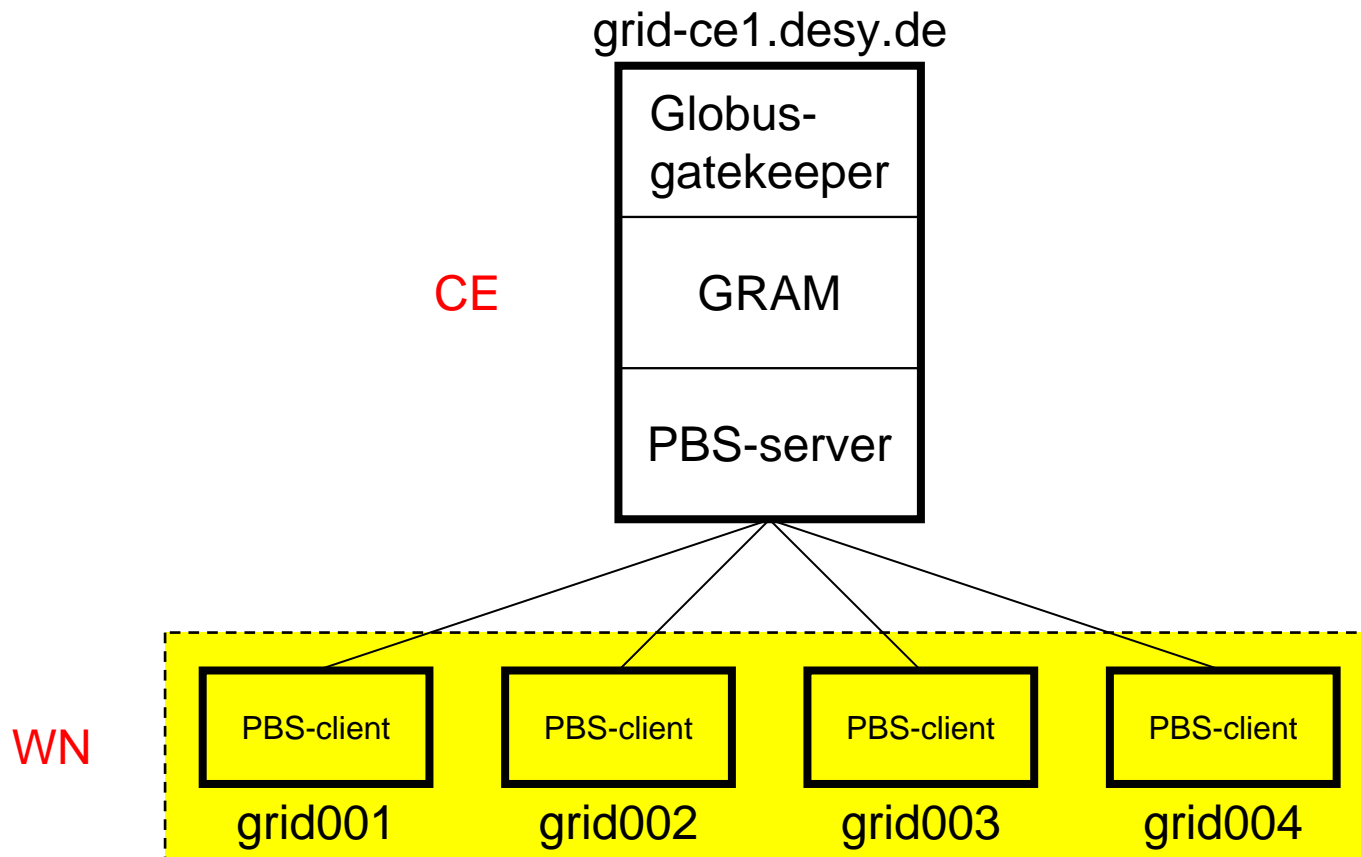
<http://grid.desy.de/>





Computing Element (TB2)

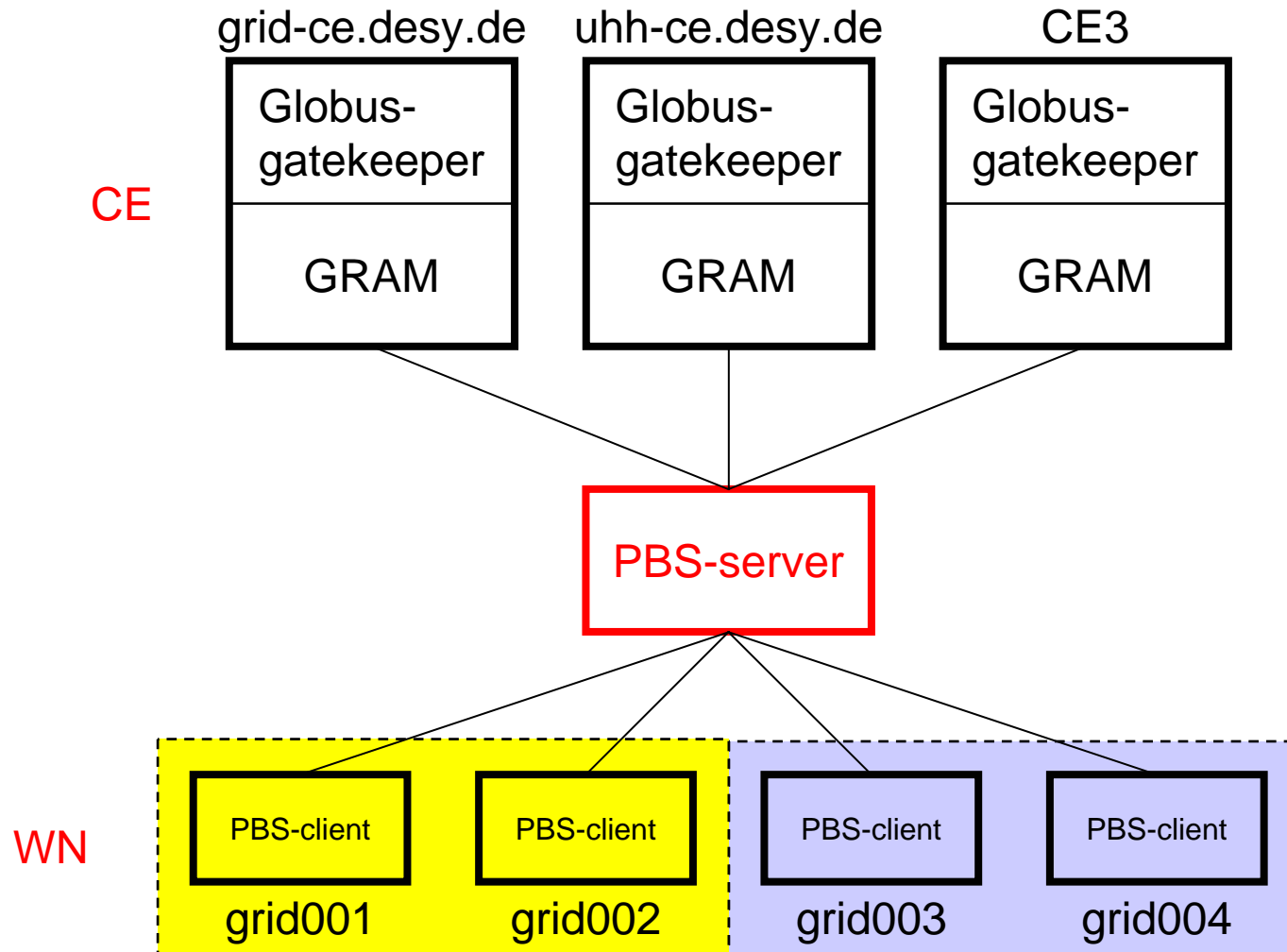
<http://grid.desy.de/>





Computing Element (planned)

<http://grid.desy.de/>





... DESY Grid Testbed2

```
ernst@grid-ui1: edg-job-list-match env.jdl
Selected Virtual Organisation name (from JDL): dteam
Connecting to host grid-rb1.desy.de, port 7772
```

```
*****
```

COMPUTING ELEMENT IDs LIST

The following CE(s) matching your job requirements have been found:

```
e5grid03.physik.uni-dortmund.de:2119/jobmanager-lcgpbs-infinite
grid-ce.physik.uni-wuppertal.de:2119/jobmanager-pbs-long
grid01.phy.ncu.edu.tw:2119/jobmanager-lcgpbs-infinite
gridkap01.fzk.de:2119/jobmanager-pbspro-default
gtbcg12.ifca.unican.es:2119/jobmanager-lcgpbs-short
gw39.hep.ph.ic.ac.uk:2119/jobmanager-lcgpbs-infinite
hotdog46.fnal.gov:2119/jobmanager-pbs-infinite
lcg-ce.usc.cesga.es:2119/jobmanager-lcgpbs-infinite
lcg00125.grid.sinica.edu.tw:2119/jobmanager-lcgpbs-long
lcg02.ciemat.es:2119/jobmanager-lcgpbs-infinite
lcg2ce.ific.uv.es:2119/jobmanager-lcgpbs-infinite
lcgce02.gridpp.rl.ac.uk:2119/jobmanager-lcgpbs-infinite
lcgce02.ifae.es:2119/jobmanager-lcgpbs-dteam
lxn1181.cern.ch:2119/jobmanager-lcgpbs-infinite
t2-ce-01.mi.infn.it:2119/jobmanager-lcgpbs-infinite
tbn18.nikhef.nl:2119/jobmanager-pbs-qshort
zeus02.cyf-kr.edu.pl:2119/jobmanager-lcgpbs-short
lcg00125.grid.sinica.edu.tw:2119/jobmanager-lcgpbs-infinite
wn-04-07-02-a.cr.cnaf.infn.it:2119/jobmanager-lcgpbs-dteam
```



... DESY Grid Testbed2

Data Management Service information

ernst@grid-ui1: edg-rm -vo=ildg printInfo

LRC endpoint <http://grid-rc.desy.de:8080/ildg/edg-local-replica-catalog/services/edg-local-replica-catalog>

RMC endpoint <http://grid-rc.desy.de:8080/ildg/edg-replica-metadatacatalog/services/edg-local-replica-catalog>

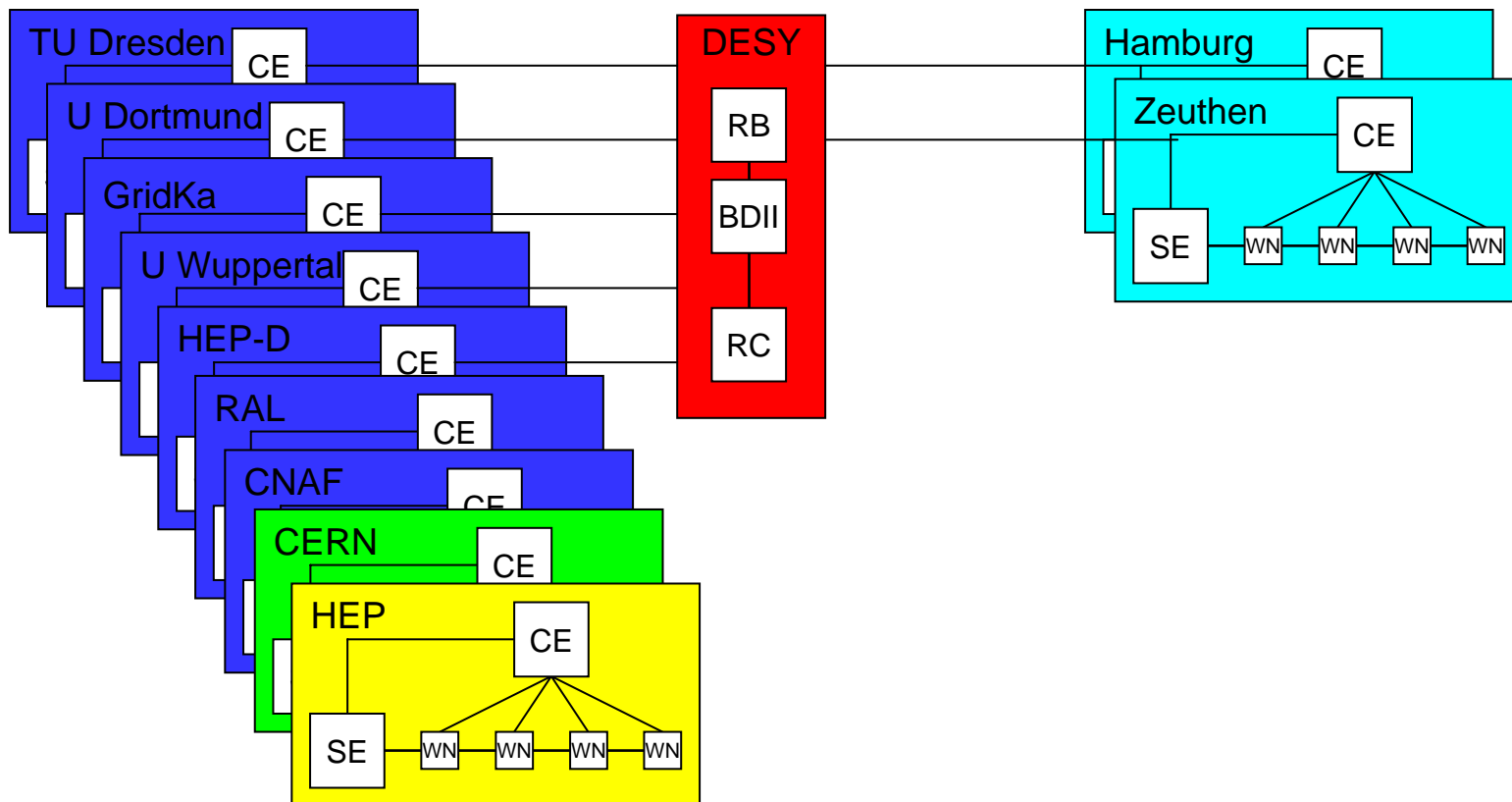
<http://grid.desy.de/>



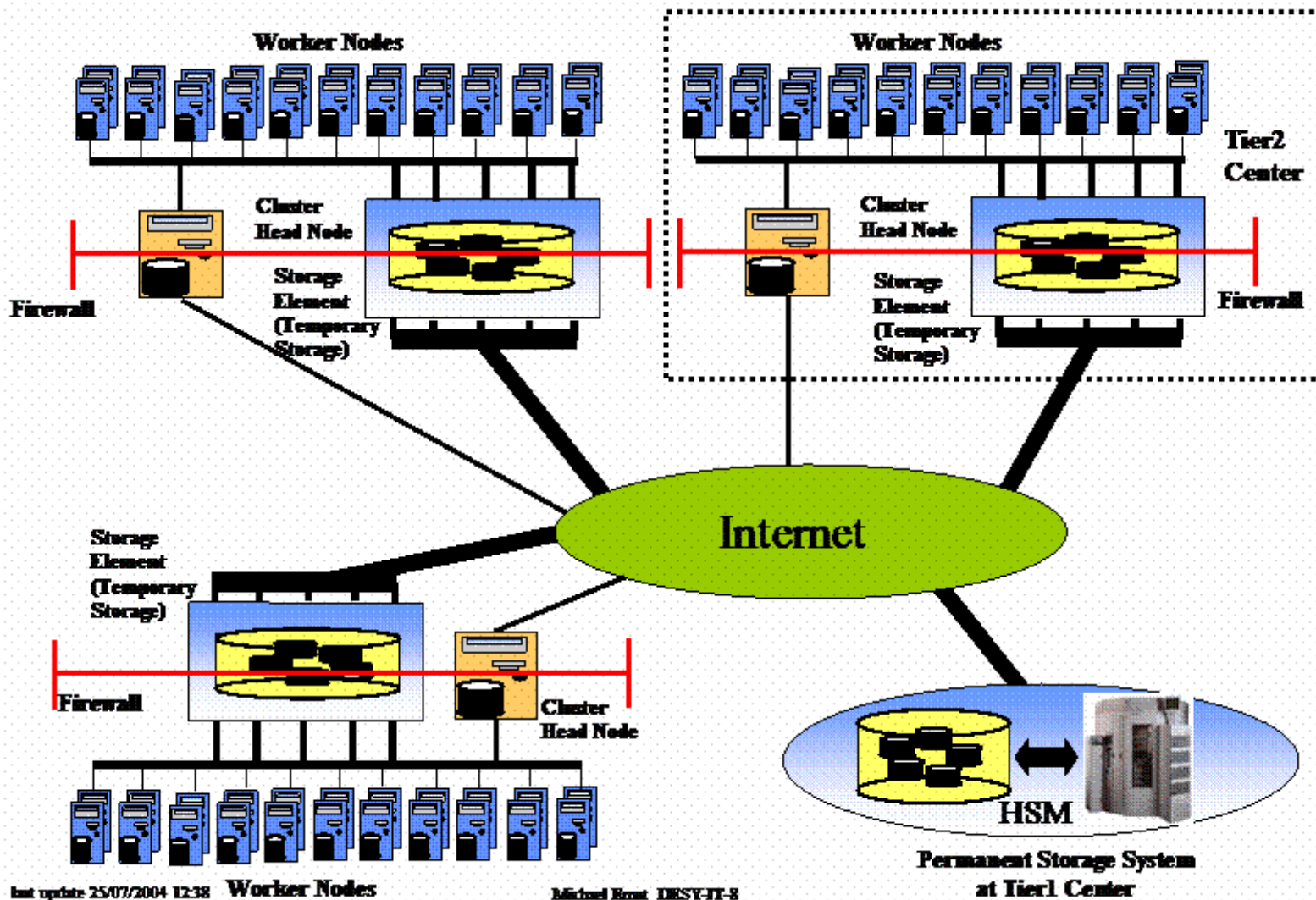
... DESY Grid Testbed2 ...

DESY & German & LHC VOs

DESY & German VOs



<http://grid.desy.de/>

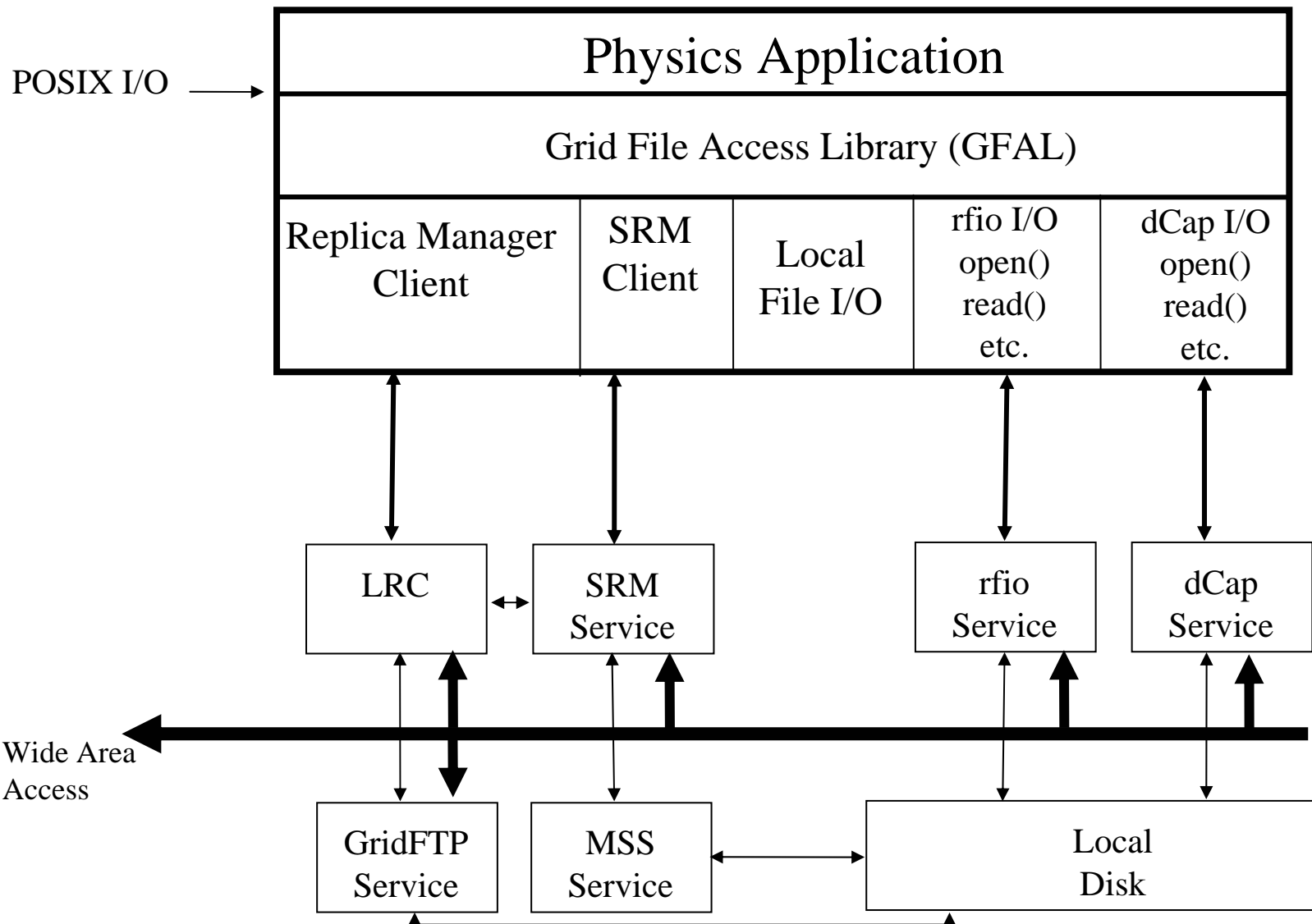


last update: 25/07/2004 12:38

Michael Ernst DESY-IT-8



The Functional View

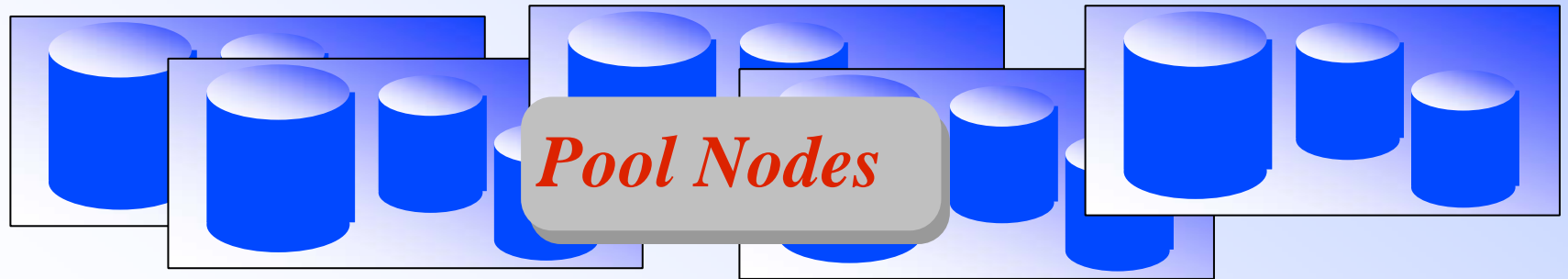
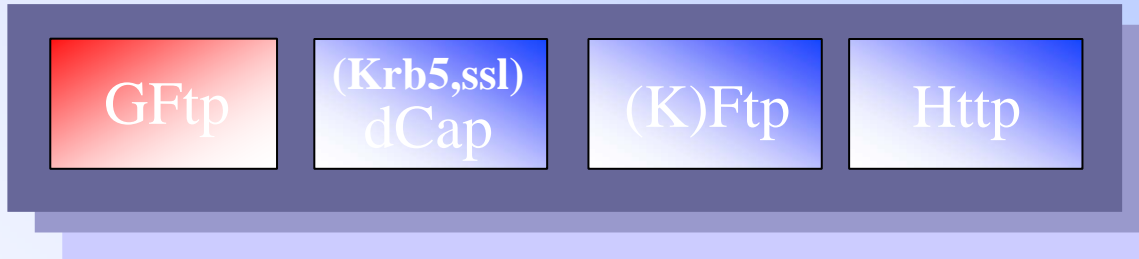


<http://grid.desy.de/>

dCache Components

I/O Door Nodes

Admin Doors





Where do SRMs belong in the Grid architecture?

<http://grid.desy.de/>

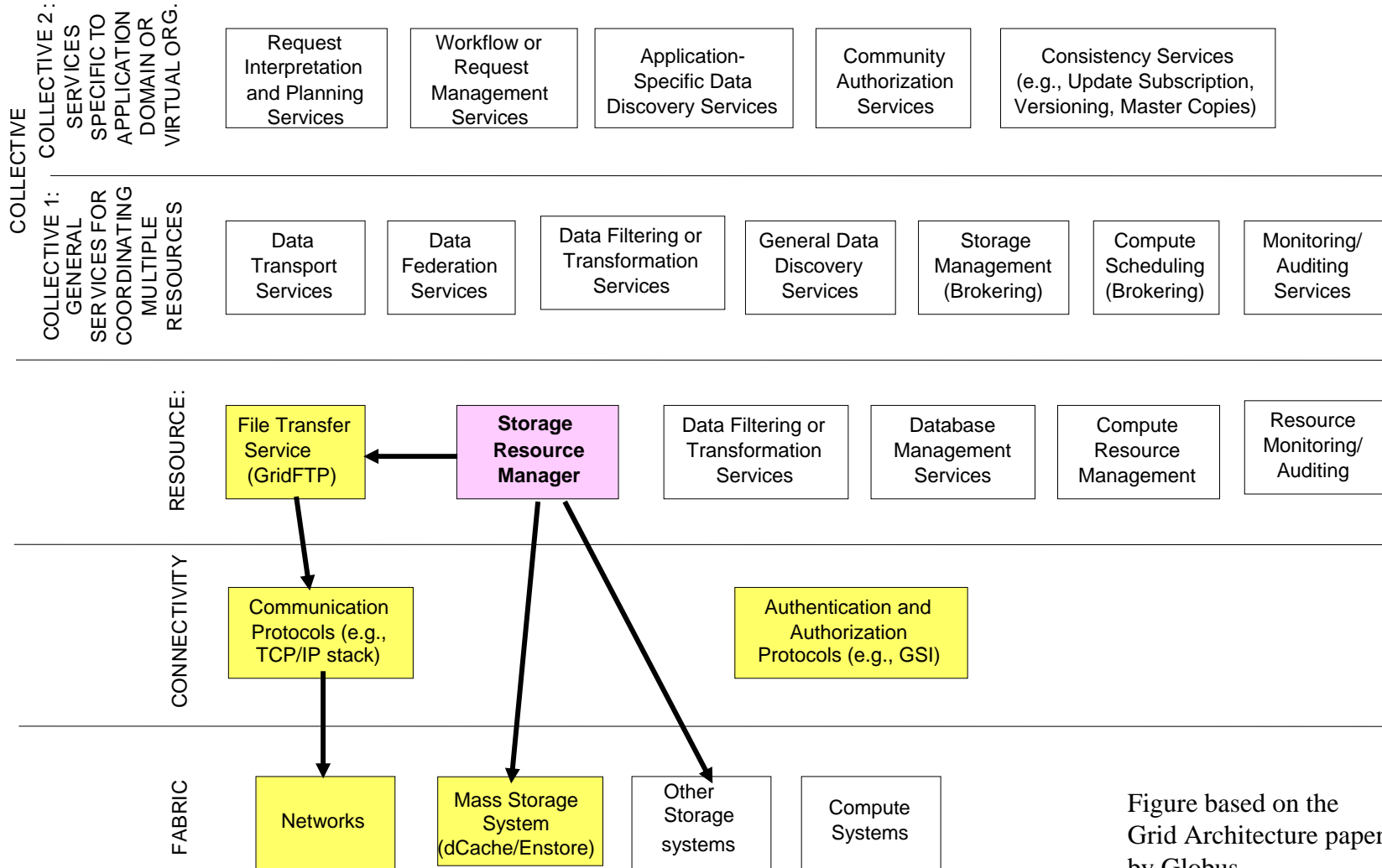
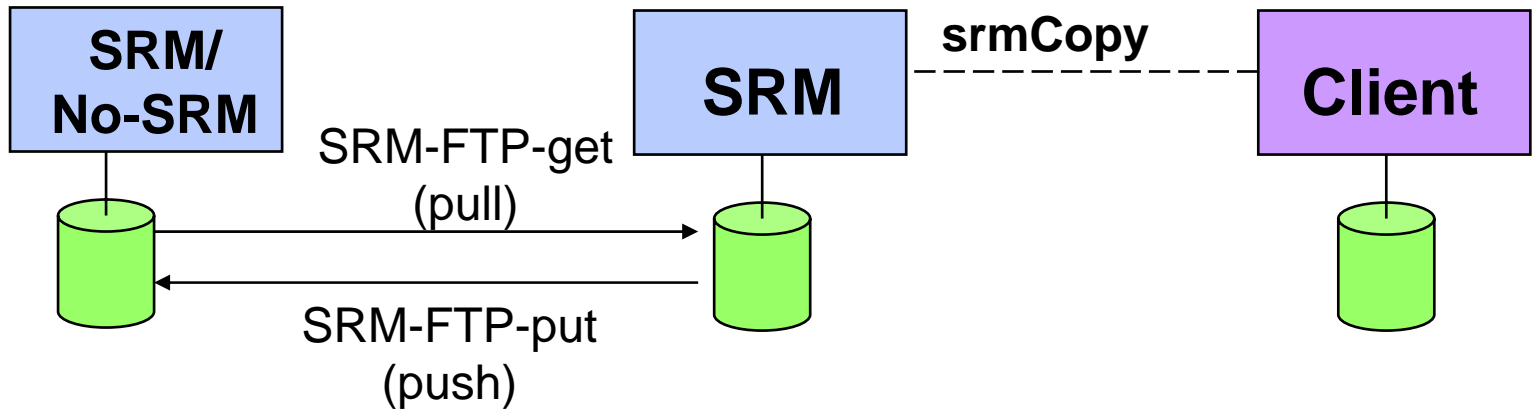
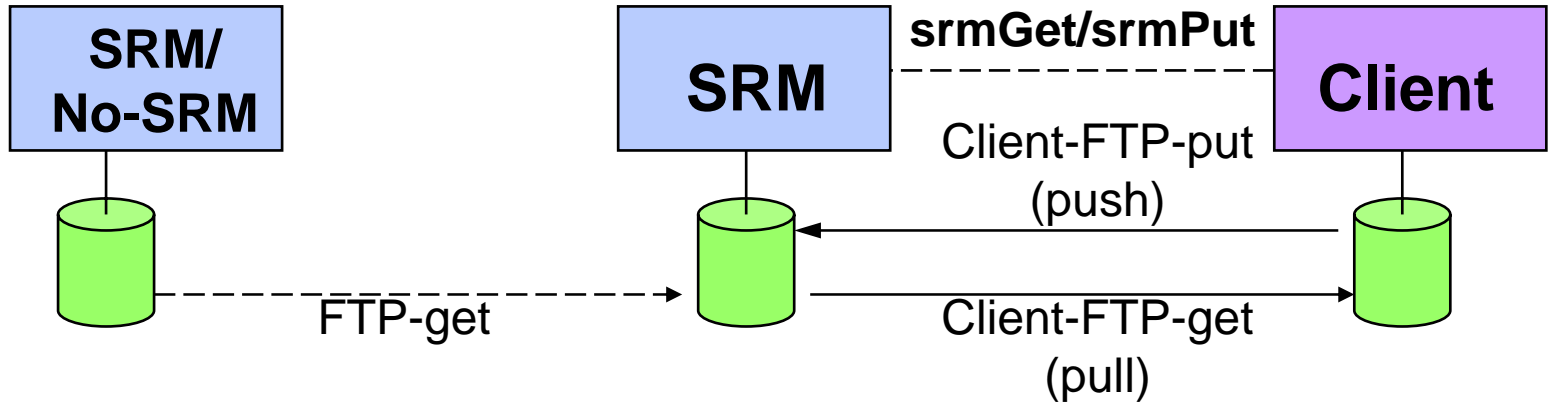


Figure based on the Grid Architecture paper by Globus



File movement functionality: srmGet, srmPut, srmCopy

<http://grid.desy.de/>





Summary

<http://grid.desy.de/>

- DESY has a working Grid infrastructure
- Member in the LCG-2 TestZone

- DESY has a working ResourceBroker (RB)
- DESY has installed Catalog Services (RMC/LRC) for supported VOs

- Tests with international sites successful

- “DESY” VOs officially supported by LCG

- We have a motivated community at DESY to use the Grid
- DESY is Tier-0/1 for DESY experiments

- DESY is setting up a production-level system (20 new servers)
- Resources for DESY, GEHP, EGEE, etc.



Requirements for Grid Data Management

- Terabytes or petabytes of data
 - Often read-only data, “published” by experiments
- Large data storage and computational resources shared by researchers around the world
 - Distinct administrative domains
 - Respect local and global policies governing how resources may be used
- Access raw experimental data
- Run simulations and analysis to create “derived” data products



Requirements for Grid Data Management

- Locate data
 - Record and query for existence of data
- Data access based on metadata
 - High-level attributes of data
- Support high-speed, reliable data movement
 - E.g., for efficient movement of large experimental data sets
- Support flexible data access
 - E.g., databases, hierarchical data formats (HDF), aggregation of small objects
- Data Filtering
 - Process data at storage system before transferring



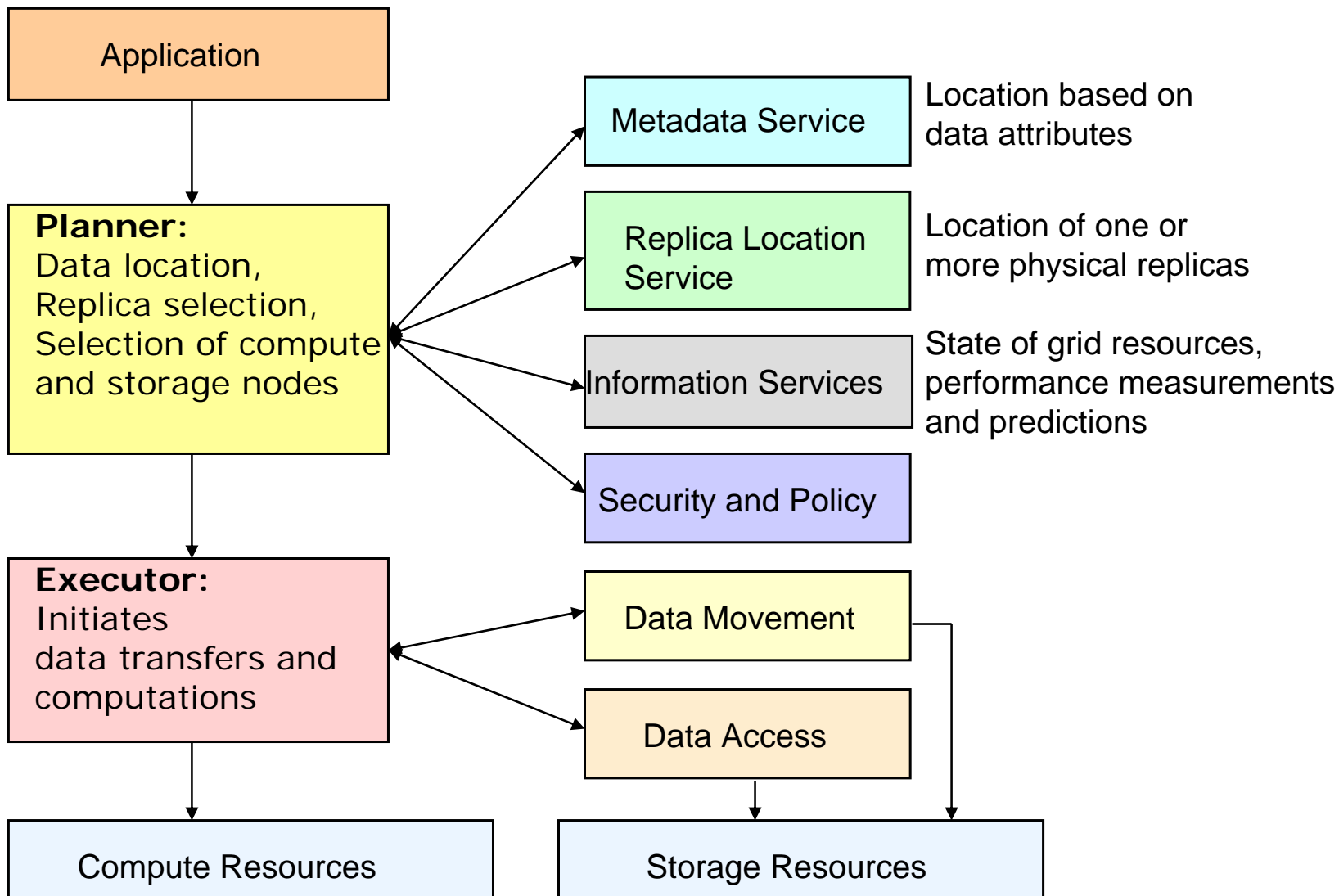
Requirements for Grid Data Management

- Planning, scheduling and monitoring execution of data requests and computations
- Management of data replication
 - Register and query for replicas
 - Select the best replica for a data transfer
- Security
 - Protect data on storage systems
 - Support secure data transfers
 - Protect knowledge about existence of data
- Virtual data
 - Desired data may be stored on a storage system (“materialized”) or created on demand



Functional View of Grid Data Management

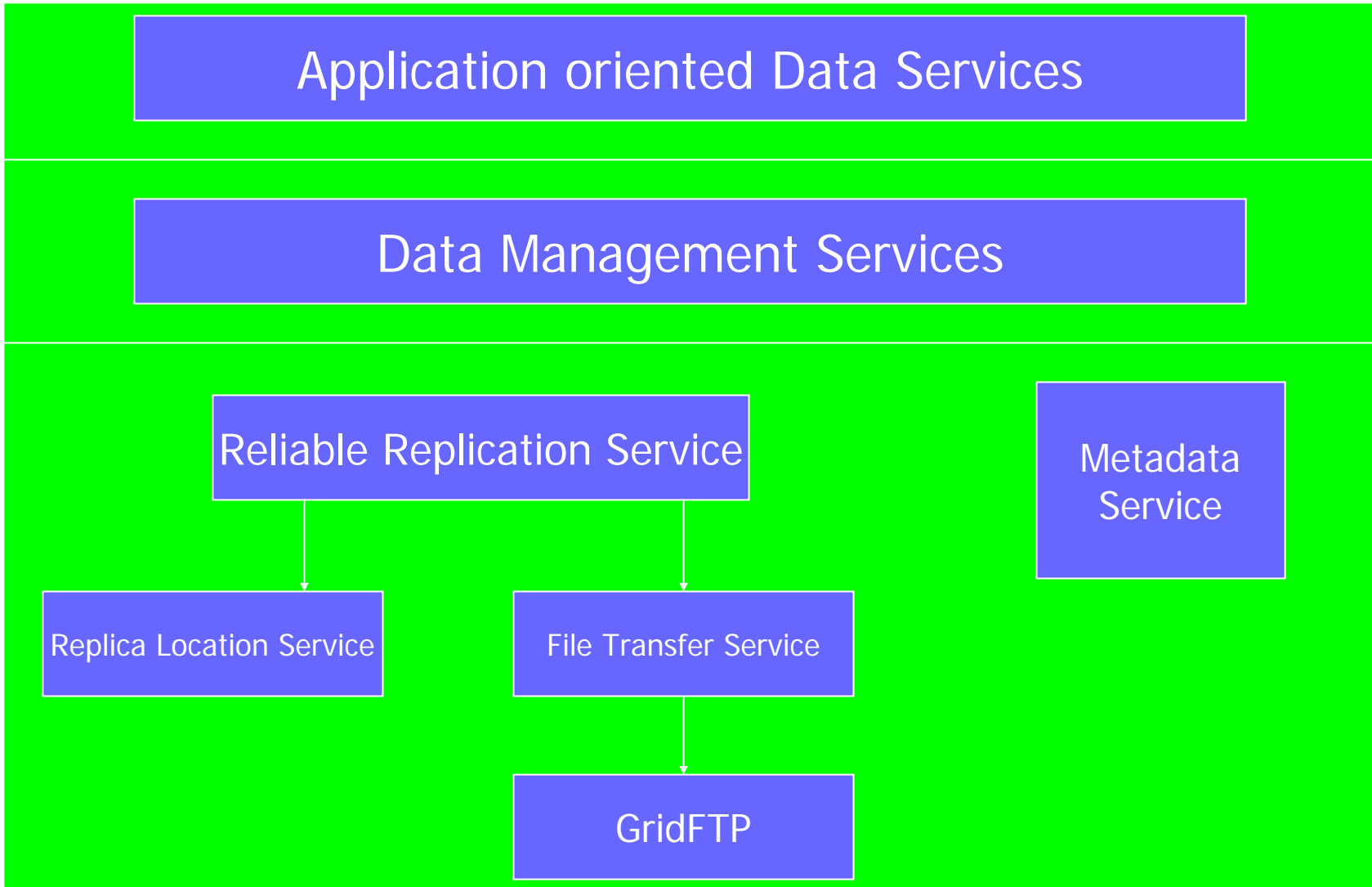
<http://grid.desy.de/>





Data Grid Architecture incl. RLS

<http://grid.desy.de/>





Components

<http://grid.desy.de/>

- GridFTP protocol for secure, efficient WAN data transfer
- File Transfer service for reliable transfer of files between storage systems
- Replica Location Service (RLS)
- a reliable replication service that provides coordinated, fault-tolerant data movement and RLS updates
- a metadata service containing information that describes logical files
- one or more higher level data management services (master copy management, workflow management, etc.)
- application-oriented data services that implement application-specific semantics and policies such as preferences for replica selection or access



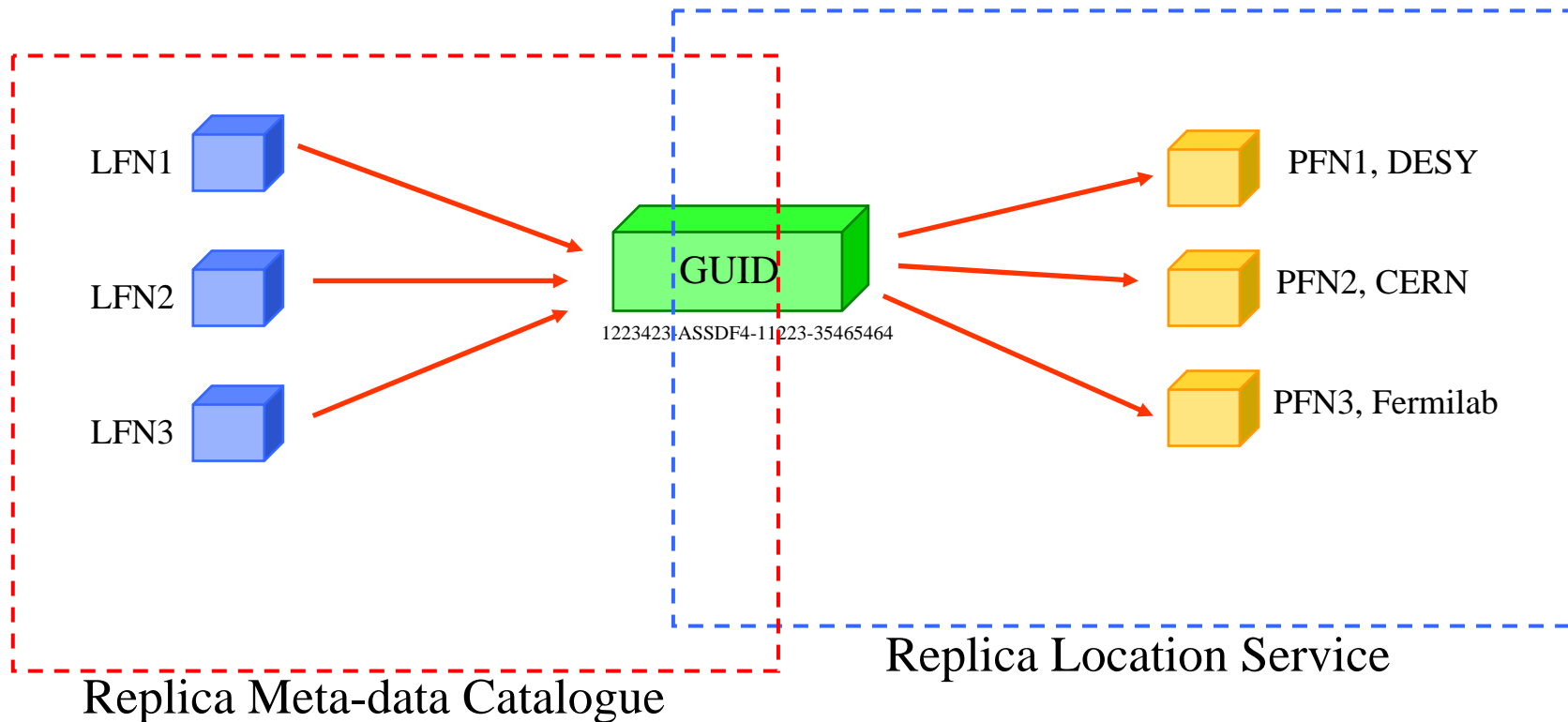
Replica Location Service (RLS)

- 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

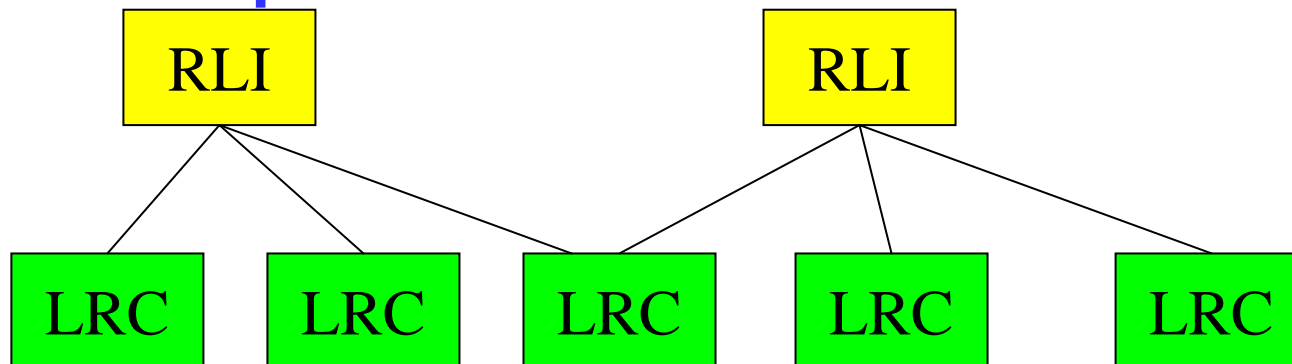


LFNs, PFNs, GUIDs

<http://grid.desy.de/>



Replica Location Indexes

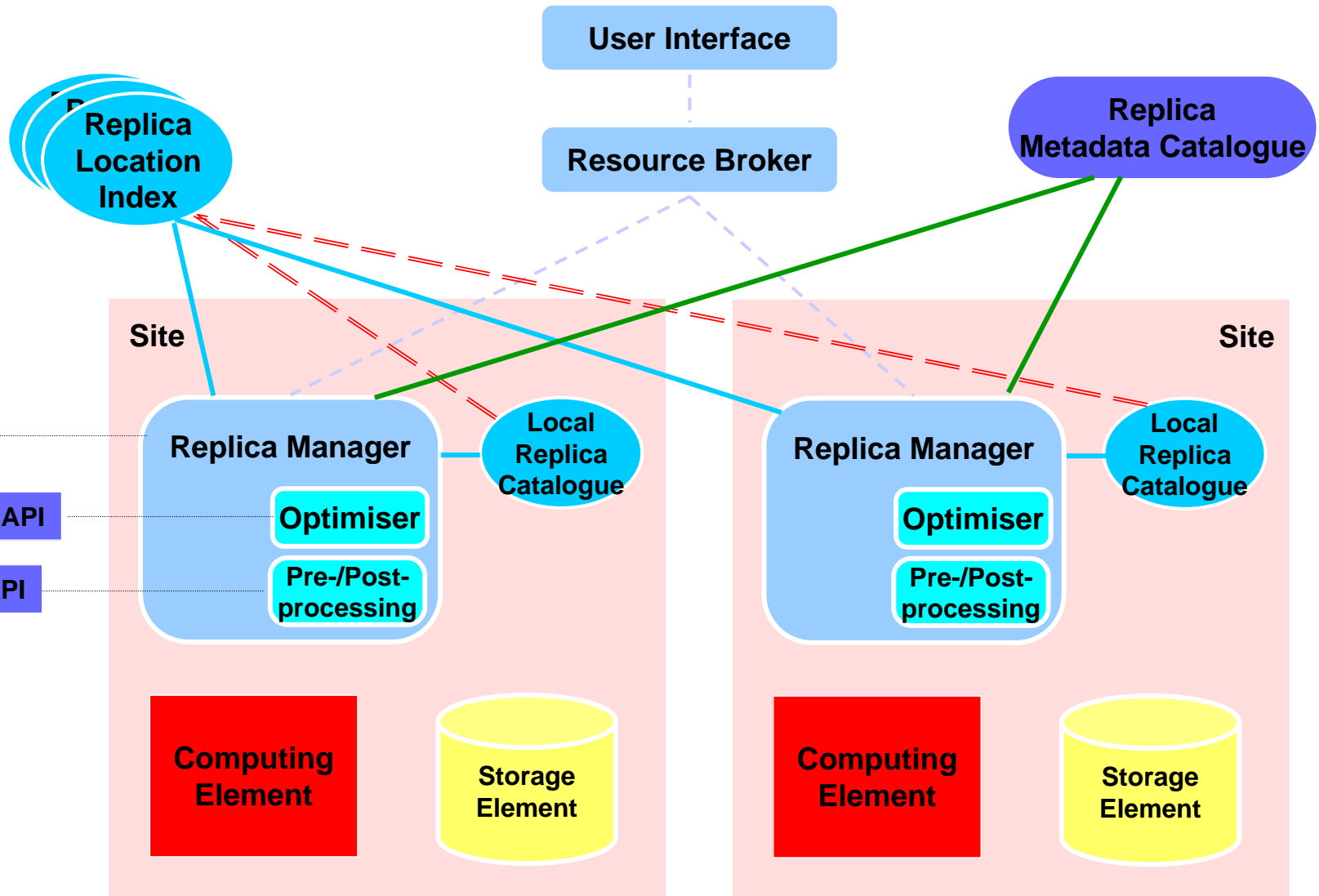


Local Replica Catalogs

- LRCs contain consistent information about logical-to-target mappings on a site
- RLIs nodes aggregate information about LRCs
- Soft state updates from LRCs to RLIs: relaxed consistency of index information, used to rebuild index after failures
- Arbitrary levels of RLI hierarchy



Replica Manager Architecture

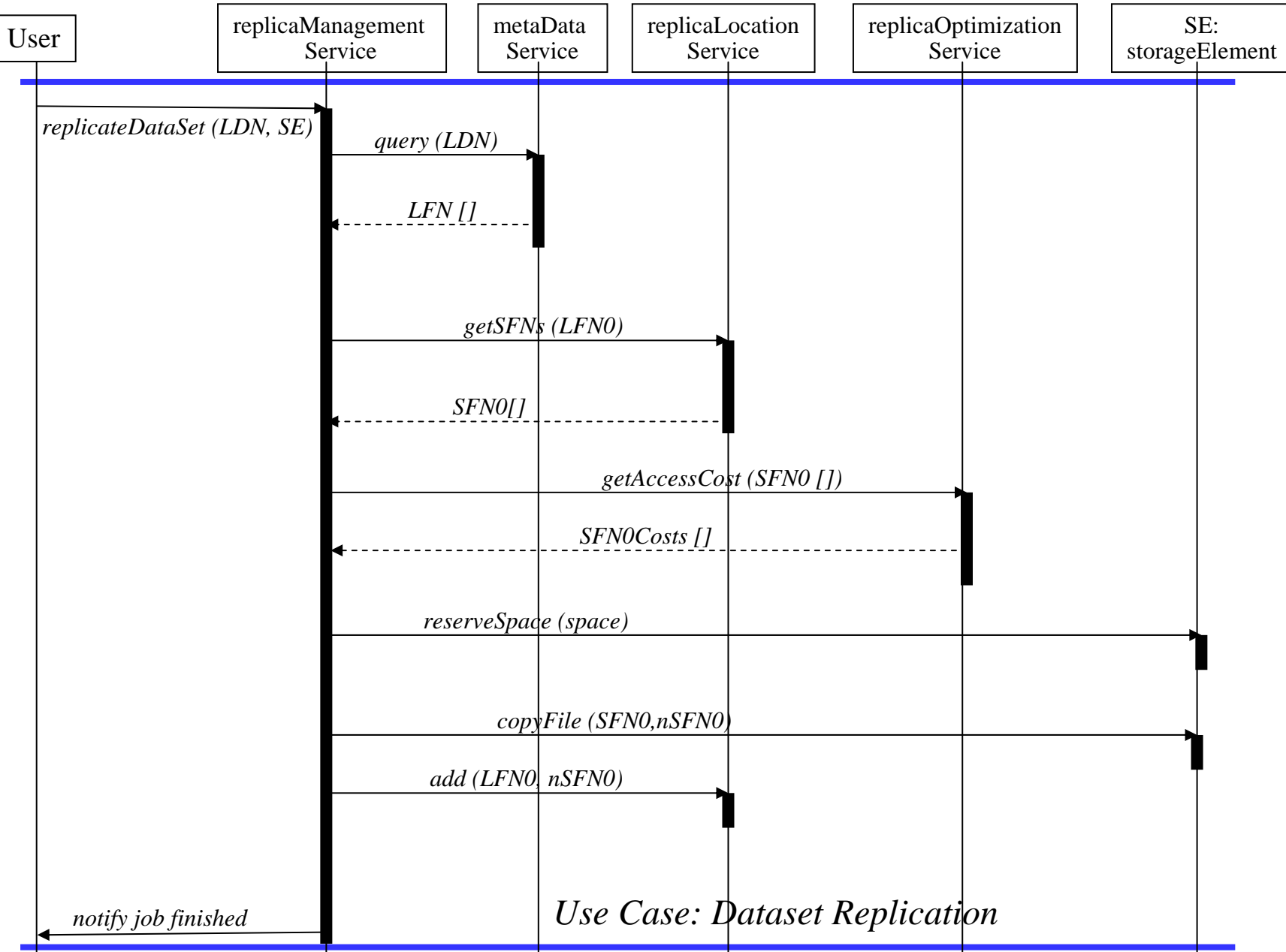


d.desy.de/

Core API

Optimisation API

Processing API





Status of Catalog Installation at DESY (07/23/2004)

<http://grid.desy.de/>

- Both the Local Replica Catalog (LRC) and the Replica Metadata Catalog (RMC) are installed and are part of the DESY LCG Testbed2
 - Distinct Instances of LRC/RMC for
 - DESY (catch all), Baikal, H1, HERA-B, HERMES, Icecube, ILDG, LC, ZEUS
 - Catalog endpoints published via the Grid Information Service (GRIS)
 - EDG Replica Manager (Client Software) installed on the UIs and on the WNs can be used to transfer files into/out of Grid-enabled storage systems (Classic and SRM/dCache SEs)
 - As part of the Job Description Language (jdl) Logical File Names (LFN) can be specified to retrieve files as job input data and to transfer results to a Permanent Storage system (e.g. the Production dCache/OSM system at DESY)
 - JAVA CLI and C++ API (much faster than CLI)



ReplicaManager (RM)

- Management Commands
 - copyAndRegisterFile Put local file into Grid Storage & register it in the Catalog(s); can be both LRC and RMC with a single command
 - BulkCopyAndRegister Do multiple copyAnd Registers
 - replicateFile Replicate an existing file to a certain Grid Storage, update catalog
 - deleteFile Delete a file from Storage (can be asked to remove all replicas) and remove entry from Catalog (LRC only)
 - getTurl Get a TURL given a SURL and a protocol



ReplicaManager (RM)

- Catalog Commands

- registerFile Register a file in the catalog
- RegisterGUID catalog Register a file with a known GUID in the catalog
- unregisterFile Unregister a file from the catalog
- listReplicas List all replicas of a logical file name
- listGUID List the GUID of a known LFN or SURL
- addAlias Add an LFN alias to an existing GUID
- removeAlias Remove an alias to GUID mapping
- printInfo Print all info service data



ReplicaManager (RM)

- File Transfer Commands
 - copyFile Copy a file to local store by using the LFN as source and a local destination
 - list List the contents of a directory on an SE (SRM or GridFTP)



Example of a simple Grid Job

```
Requirements = other.GlueCEUniqueID == "grid-ce2.desy.de:2119/jobmanager-lcgpbs-short";  
Executable = "/bin/hostname";  
StdOutput = "std.out";  
StdError = "std.err";  
OutputData = {  
  [  
    OutputFile="std.out";  
    LogicalFileName="lfn:my_test_result";  
    StorageElement="grid-se2.desy.de"  
  ]  
};
```



Metadata

- Information that describes data files or data items
- Application-specific
 - Temperature, longitude, latitude, depth
 - Time, duration, sensor
- Application-independent
 - creator, logical name, time created, access control
 - notion of a data collection—data collected during an experiment, data collected over a certain time interval
 - notion of a view--users might want to group the data in a way that they want to look at it



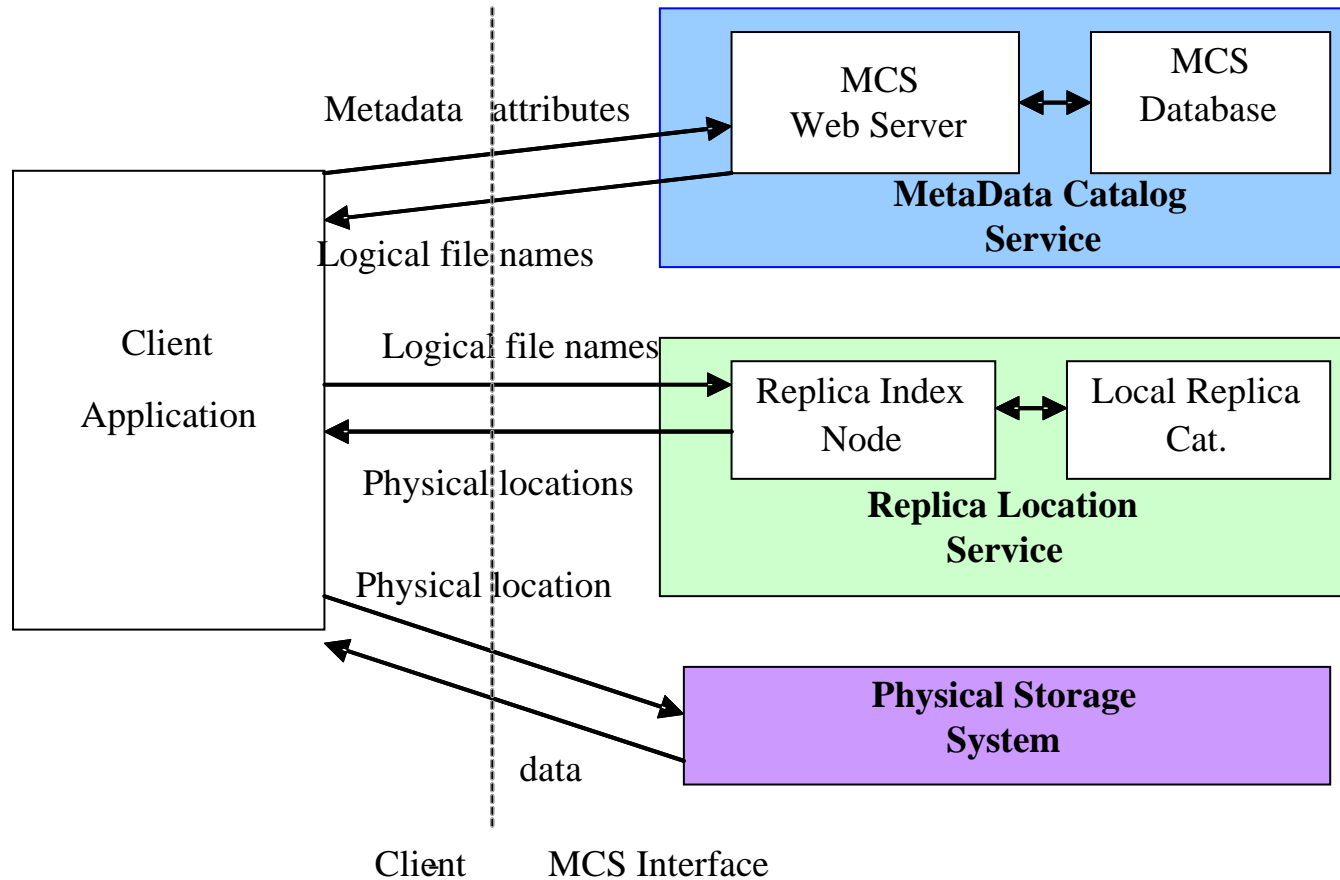
Types Of Metadata

- Physical file metadata
 - Depends on the actual location of the file
 - Depends on the characteristics of a given storage system
- Logical file metadata
 - How data files were created or modified
 - By whom, when, what process, or instrument, or what simulation or analysis software was run on which computational engine with which input parameters)
 - description of what the data stored in a file represent
 - precipitation measurements over South America for December 1998
 - particle collisions at HERA for a period of 1 second
 - file format information (e.g., netCDF vs. XML vs. ASCII vs. binary).



MCS Usage Scenario

<http://grid.desy.de/>





Metadata Service

- *Storing attributes:*
 - provide a mechanism for storing various metadata attributes associated with logical files.
- *Querying:*
 - return the names of all logical files that possess the attributes present in the queries
 - respond to queries about one or more attributes of a logical file.
- *Extensibility:*
 - support for user-defined attributes
 - capability to access application-specific metadata catalogs external to MCS
- *Consistency:*
 - maintain strict consistency over its contents
 - if the MCS is replicated, then all copies of the metadata database must be updated atomically



Metadata Service (cont)

<http://grid.desy.de/>

- *Support for logical collections:*
 - Support a tree hierarchy of collections with well defined rules for delegation of authorization rights to child collections.
 - Support simple queries that list the logical files in a collection
 - Respond to attribute-based queries on logical collections.
- *Support for authentication and authorization:*
 - provide authentication based on the Grid Security Infrastructure (GSI)
 - provide authorization/access control
 - based on attributes
 - Based on community policies in the Community Authorization Service.



Metadata Service (cont)

- Support for Logical Views
 - Aggregate of any acyclic selection of logical files, logical collections or other logical views.
 - Logical files and logical collections may belong to many different logical views.
- *Creation information*
 - record information about the creator of logical files, collections and views as well as creation times.
- *Annotations*
 - allow users to add descriptive text as annotations to logical files, logical collections and logical views.
- *Audit records*
 - provide the ability to log all the accesses to a particular data item, including the identity of the user and the action that was performed.



Metadata Service (cont)

- *Transformation history*
 - provide the capability to store records of transformations on a dataset
 - its creation and subsequent processing.
 - the identity of data modifiers
 - information about analyses run and the input parameters used
- *Master copy support*
 - provide support for associating master copy attributes with logical files
 - answer queries about these attributes
 - provide a means of locating the master copy
- *Versioning*
 - provide support for multiple versions of a particular logical file.



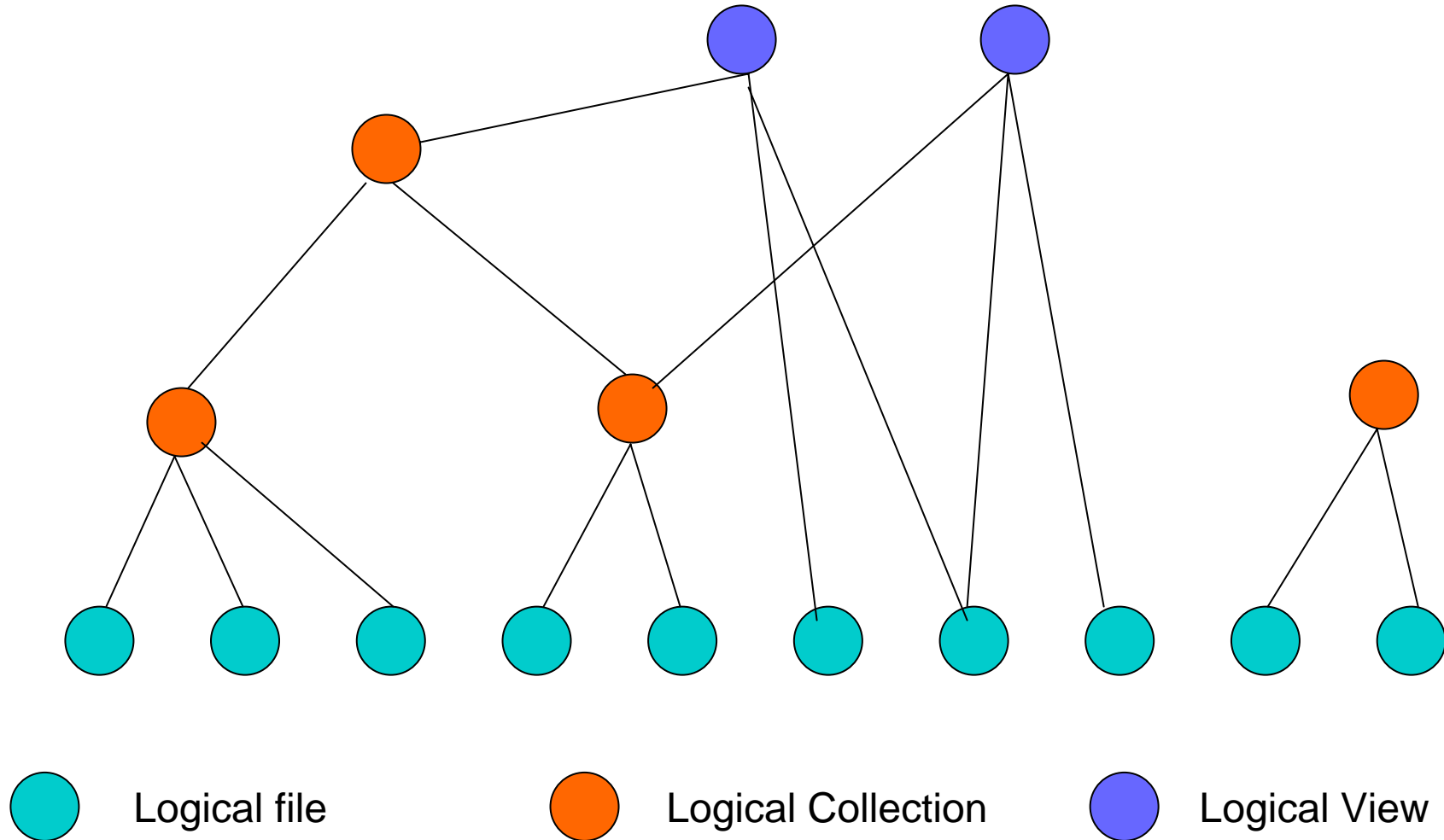
Metadata Service (cont)

- *Support for containers*
 - interface with an external container management service that constructs containers and extracts individual files from within containers.
 - provide attributes that enable logical files to be associated with containers via a particular external container management system.
- *Performance:*
 - should provide short latencies on query and update operations and support relatively high query and update rates.
- *Scalability:*
 - should scale to support information about millions of logical files and thousands of logical collections and logical views.



Data Model

<http://grid.desy.de/>





Proposed Schema

- *logical file metadata:*
 - logical file name
 - data type
 - version number
 - master copy—points to a Local Replica Catalog
 - container information
 - information about the creator
 - last modifier of the data



Supported Metadata

- *logical collection metadata:*
 - collection name
 - description
 - set of files in a collection
 - annotations on the collection
 - information about the creator and modifier(s)
 - collection hierarchy information (parent collection id)



Supported Metadata

- *logical view metadata*
 - view name
 - view attributes
 - description
 - view creator and/or modifier
 - logical files, collections, sub views within a view.
- *authorization metadata*
 - used in the absence of an external authorization service (CAS)
 - specifies access privileges on logical files or collections.
- *writers of metadata*
 - includes contact information
- *audit metadata*
 - records actions performed via the metadata service



Supported Metadata

- *annotation metadata*
 - used to describe logical files, collections, and views
 - Timestamped

- *external catalog access*
 - provides information needed to contact external metadata catalogs

- *user-defined metadata, attributes on*
 - logical files
 - logical collections
 - logical views
 - provides extensibility beyond pre-defined metadata attributes

- **Creation history**
 - Initially, textual description
 - Eventually, more complex provenance information that can be queried



Authorization Information

<http://grid.desy.de/>

- associated with both individual logical files and logical collections.
- must be maintained for both individual users and a Community Authorization Service (CAS or VOMS).
- logical collections allow authorization on groups of files without requiring that permissions be specified on each logical file.
- a logical file may belong to at most one logical collection
- the access permissions for a collection apply to all logical files within the collection.
- in addition to the permissions specified on the collection, the user might impose additional access restrictions on individual logical files.
- the access to the file attributes is determined by the intersection of the access permissions on the file and collection.



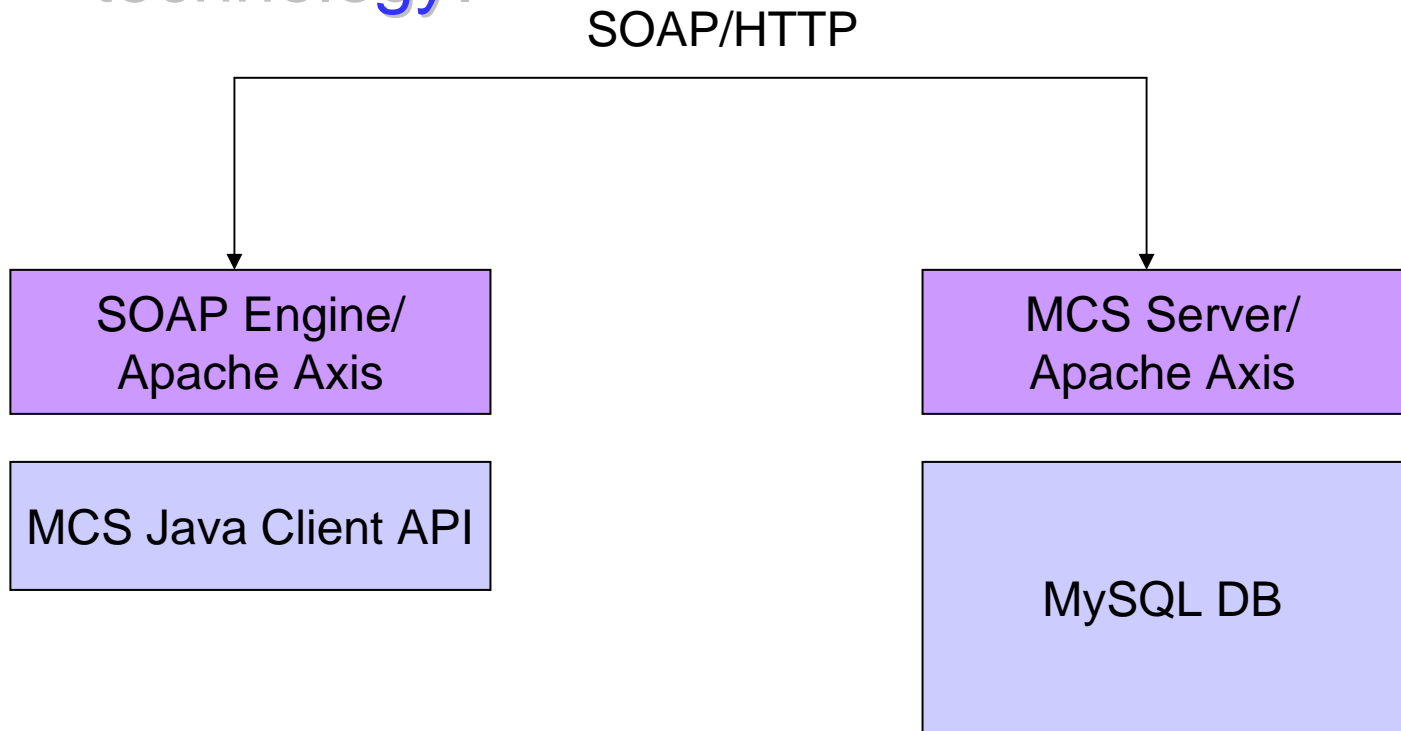
Accessing external catalogs

- The user or application can then use this information to further query the external catalog
- External_Data_id The domain or application specific data identifier used in the external database
- Database_type The external database type (relational or XML)
- Database_name The name of the database
- Database_desc General description of database content
- Table_name The table in the external database
- Host_name The hostname where the external database is located
- Host_address The ip address of the host where the database is located



Current Design

- Initial Implementation
 - Simple, centralized Metadata Service
 - based on open source relational database technology.





Current Functionality

■ Data Access

- Querying Database based on attributes
- Querying attributes of an object
- Querying collection or view contents
- Querying based on user defined attributes
- Retrieving XML metadata

■ Data Publishing

- Creating a logical file, collection or a view
- Modifying attributes
- Deleting a logical file, collection or a view
- Annotating a logical file, collection or a view
- Adding contents to a view
- Storing XML metadata
- Grant/revoke authorization (dn based)



Future Directions

- Continue gathering requirements from the community
- Evaluate the database technology
 - Initial implementation relational
 - Possible use of XML db
- Authorization
 - Modeled but not fully implemented
 - Better understanding
- Supporting provenance information
- Federation of heterogeneous metadata services
- Bulk import and export of large amounts of metadata into the service
- Container management