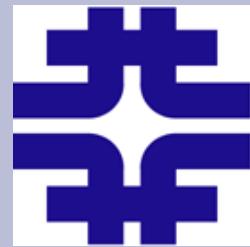




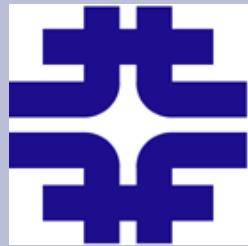
# Storage Resource Management at Fermilab



**Timur Perelmutov  
Don Petravick**

**Fermi National Accelerator Lab**

# Fermilab Storage



- We have covered:
  - Enstore – permanent tape storage
    - Scales to several petabytes
    - Flexible media selection
    - Encp access protocol
  - Dcache – distributed disk storage
    - Scales to hundreds of terabytes
    - Commodity hardware
    - Pluggable transfer protocols
  - PNFS – namespace and file metadata server for Enstore and Dcache
- Next: **SRM** – Storage Resource Manager

### **Jefferson Lab**

Bryan Hess  
Andy Kowalski  
Chip Watson

### **Fermilab**

Don Petravick  
Timur Perelmutov

### **LBNL**

Arie Shoshani  
Alex Sim  
Junmin Gu

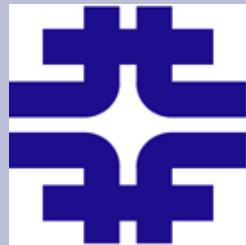
### **EU DataGrid WP2**

Peter Kunszt  
Heinz Stockinger  
Kurt Stockinger  
Erwin Laure

### **EU DataGrid WP5**

Jean-Philippe Baud  
Stefano Occhetti  
Jens Jensen  
Emil Knezo  
Owen Synge

# SRM Motivation (1)

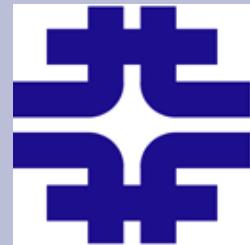


Grid Architecture promises Reservation and Scheduling  
of the Following **Shared** Resources

- Computing Resources
- Network Resources
- Storage Resources (often neglected)

SRM provides Reservation and Scheduling of  
the Storage Resources

## SRM Motivation (2)

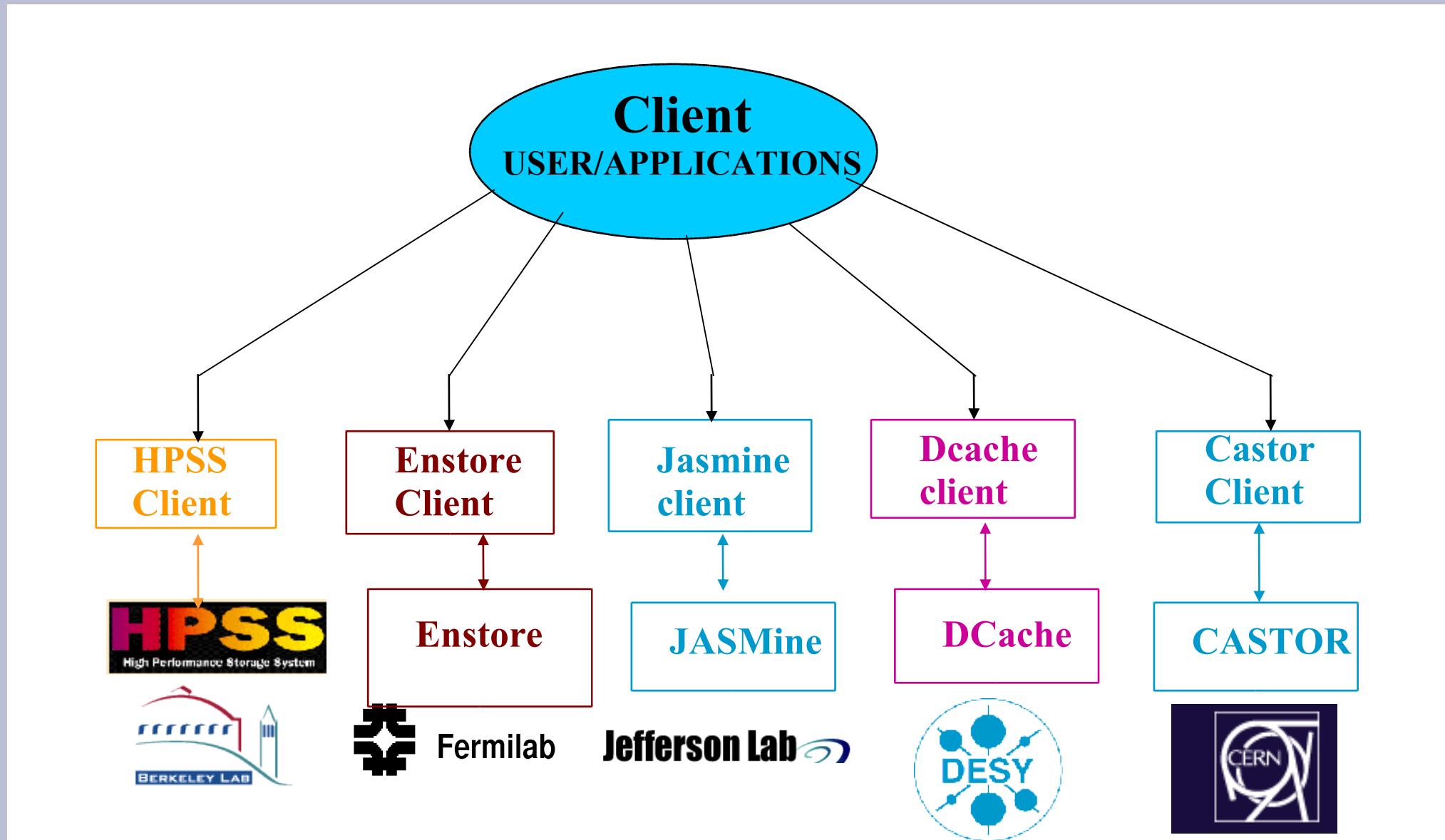


High Energy Physics Collaborations span multiple institutions where

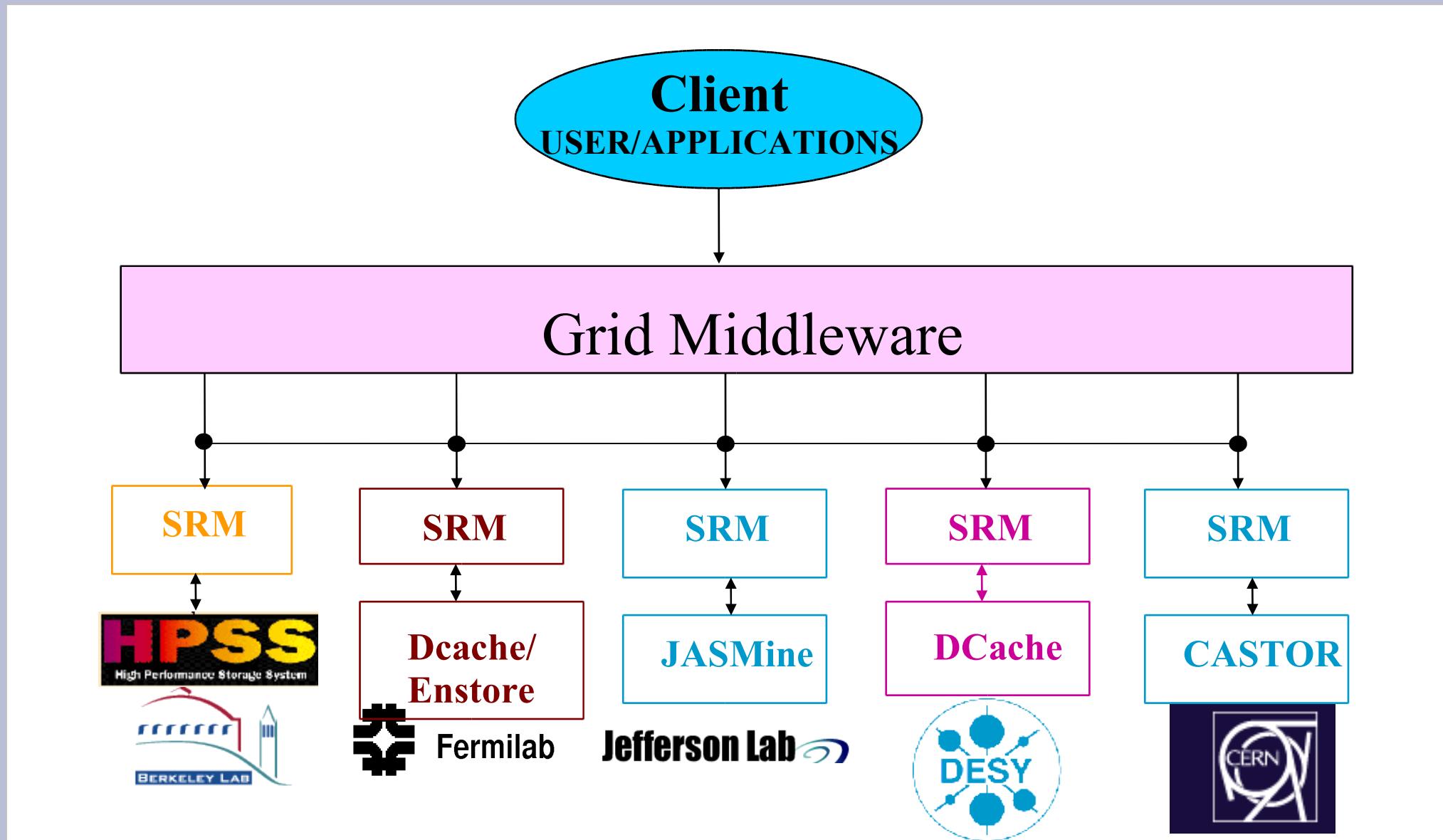
- A large variety of types of Storage Systems exist
  - Single Disk / Raid
  - Robotic Tape Storage System (Enstore, HPSS)
  - Distributed Disk Cache (dCache)
  - Hierarchical Storage System (dCache - Enstore)
- Heterogeneous environments and proliferation of custom Mass Storage Systems (MSSs) exist
- User applications often need to access data at multiple institutions on multiple MSSs

SRM provides Standardized Uniform Access to Heterogeneous Storage

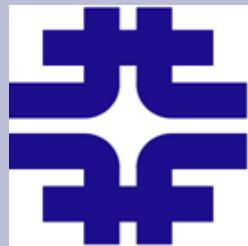
# Access to Multiple MSS



# Uniform Access via SRM



# Storage Classification



Storage systems can be classified by:

- Persistence of data
  - Permanent
  - Temporary
- Data access availability
  - Data immediately available
  - Data needs advanced reservation before utilization (tapes need to be mounted, files need to migrate to disks, etc.)
- Supported transfer protocols
  - File transfer protocols
  - POSIX like access protocols

Need management interface that supports all of the above

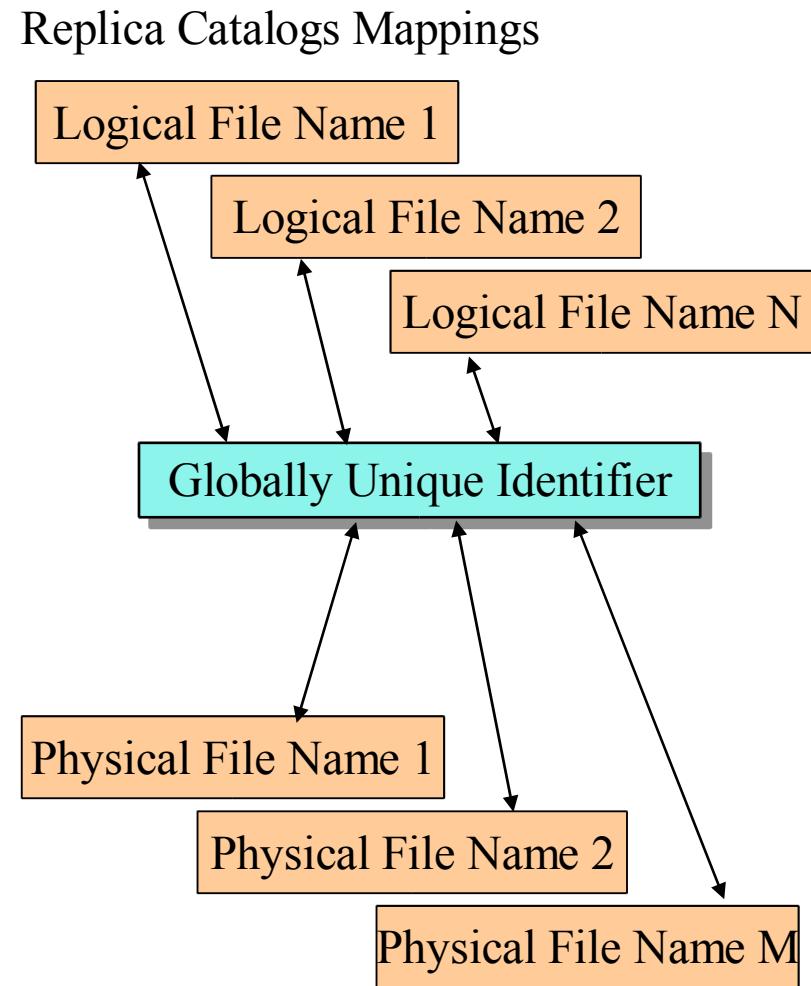
# Storage Resource Managers

SRMs are middleware components that manage shared storage resources on the Grid and provide:

- Uniform access to heterogeneous storage
- File Transfer Protocol negotiation
- Dynamic Transfer URL allocation
- Access to permanent and temporary types of storage
- Advanced space and file reservation
- Reliable transfer services

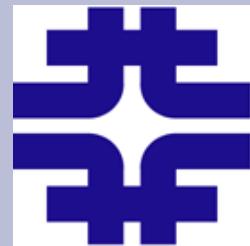
# Data life cycle on the grid overview

- Collection or generation
  - Data is collected from the CMS detector
  - Data generated by the analysis
- Storage
  - Data is stored on a shared storage system
- Registration
  - Data is registered in a Replica Catalog by via a Replica Registration Service (RRS)
- Replication
  - Data is replicated to a storage geographically close to the consumer
- Replica registration
  - RRS
- Delivery and consumption



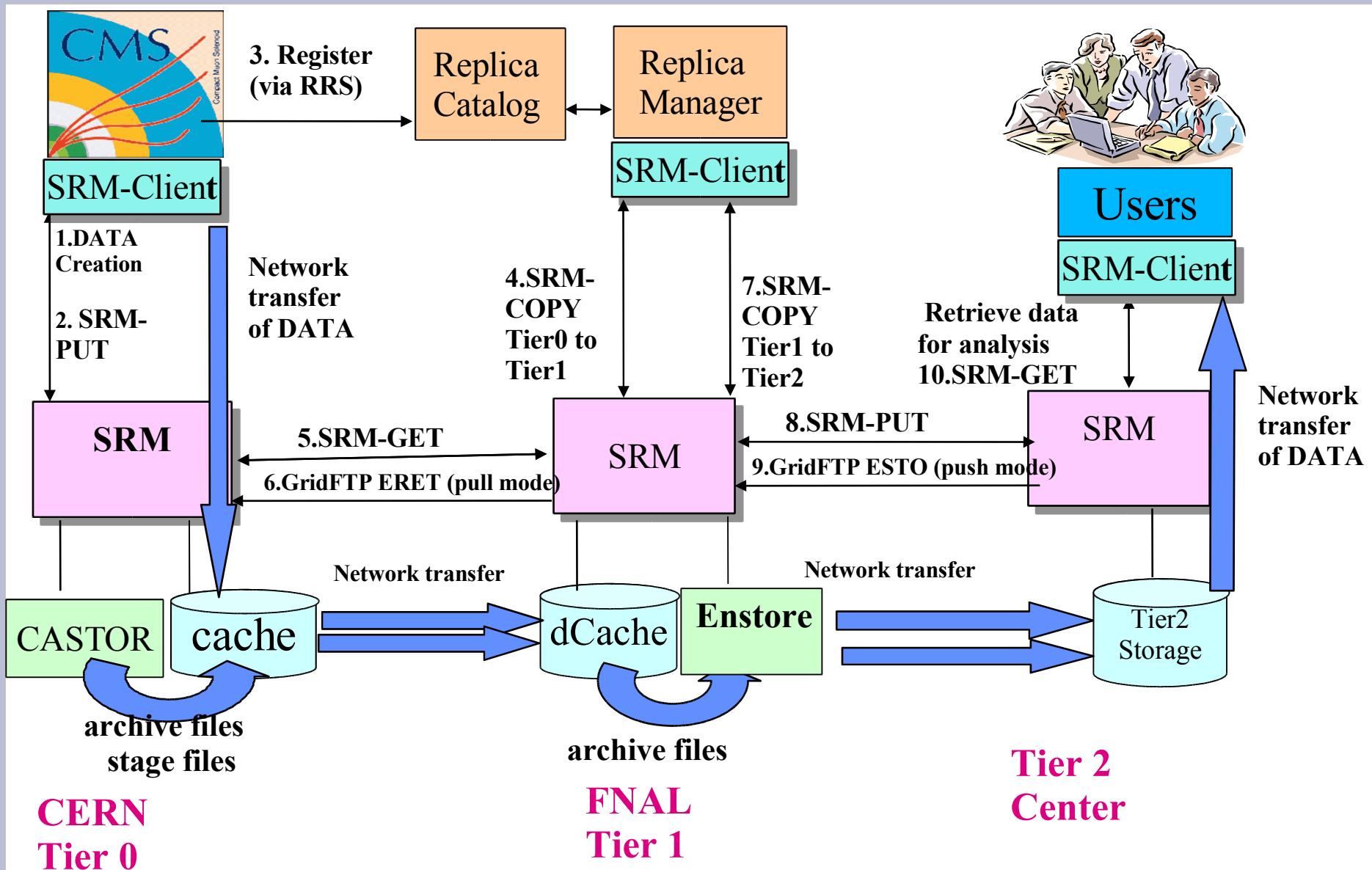
# Illustration

## US-CMS DATAGRID



- US-CMS is building a data grid to facilitate physics data analysis at academic institutions across the United States.
- Data grid 3 tier architecture.
  - Tier 0 CERN, Geneva, Switzerland.
  - Tier 1 consists of 5 regional centers, FERMILAB in Batavia, IL is a North American Center.
  - Tier 2 consists of 25 centres, 5 of these are in North America.
- SRM copy is used as a management protocol and reliable replication service for movement of data from tier 0 to tier 1 centers and from tier 1 to tier 2.

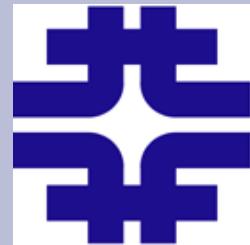
# Data life cycle on the grid and a role of SRM on the GRID



# Storage Resource Manager versions

- Two SRM Interface specifications
  - SRM v1.1 provides
    - Data access/transfer
    - Implicit space reservation
  - SRM v2.1 adds
    - Explicit space reservation
    - Namespace discovery and manipulation
    - Access permissions manipulation
- Fermilab SRM implements SRM v1.1 specification
- SRM v2.1 by the end of 2004

# SRM Protocols and Groups of Functions



SRM interface consists of the following groups of functions:

- Space Management Functions – v2.1
- Data Transfer Functions – v1.1 and v2.1
- Directory Functions – v2.1
- Permission Functions – v2.1
- Status Functions – v1.1 and v2.1

SRM v2.1 in the rest of this talk

# SRM Interface Details

## Space Management Functions

- SrmReserveSpace
- SrmReleaseSpace
- srmUpdateSpace
- srmCompactSpace
- srmGetSpaceMetaData
- srmChangeFileStorageType
- srmGetSpaceToken

## Directory

- SrmMkdir
- srmRmdir
- srmRm
- srmLs
- srmMv

## Data transfer functions

- srmPrepareToGet
- SrmPrepareToPut
- srmCopy
- SrmRemoveFiles
- srmReleaseFiles
- srmPutDone
- srmAbortRequest
- srmAbortFiles
- srmSuspendRequest
- srmResumeRequest

## Status Functions

- srmStatusOfGetRequest
- srmStatusOfPutRequest
- srmStatusOfCopyRequest
- srmGetRequestSummary
- srmExtendFileLifeTime
- SrmGetRequestID

## Permission

- srmSetPermission
- srmReassignToUser
- srmCheckPermission

# SRM Interface Details

## Space Management Functions

- **SrmReserveSpace**
- **SrmReleaseSpace**
- srmUpdateSpace
- srmCompactSpace
- srmGetSpaceMetaData
- srmChangeFileStorageType
- srmGetSpaceToken

## Directory

- SrmMkdir
- srmRmdir
- srmRm
- srmLs
- srmMv

## Data transfer functions

- srmPrepareToGet
- **SrmPrepareToPut**
- srmCopy
- SrmRemoveFiles
- srmReleaseFiles
- **srmPutDone**
- srmAbortRequest
- srmAbortFiles
- srmSuspendRequest
- srmResumeRequest

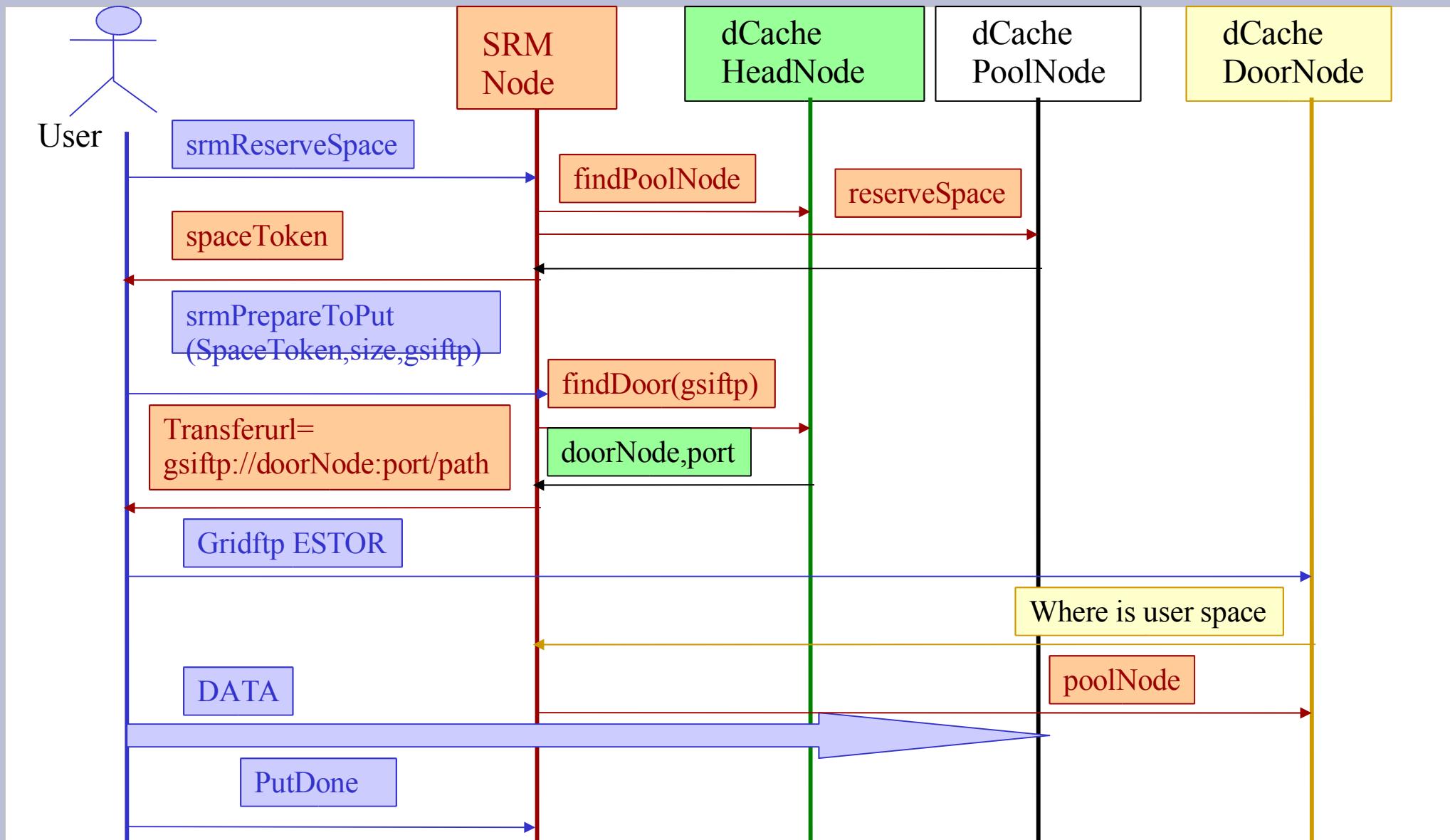
## Status Functions

- srmStatusOfGetRequest
- **srmStatusOfPutRequest**
- srmStatusOfCopyRequest
- srmGetRequestSummary
- srmExtendFileLifeTime
- **SrmGetRequestID**

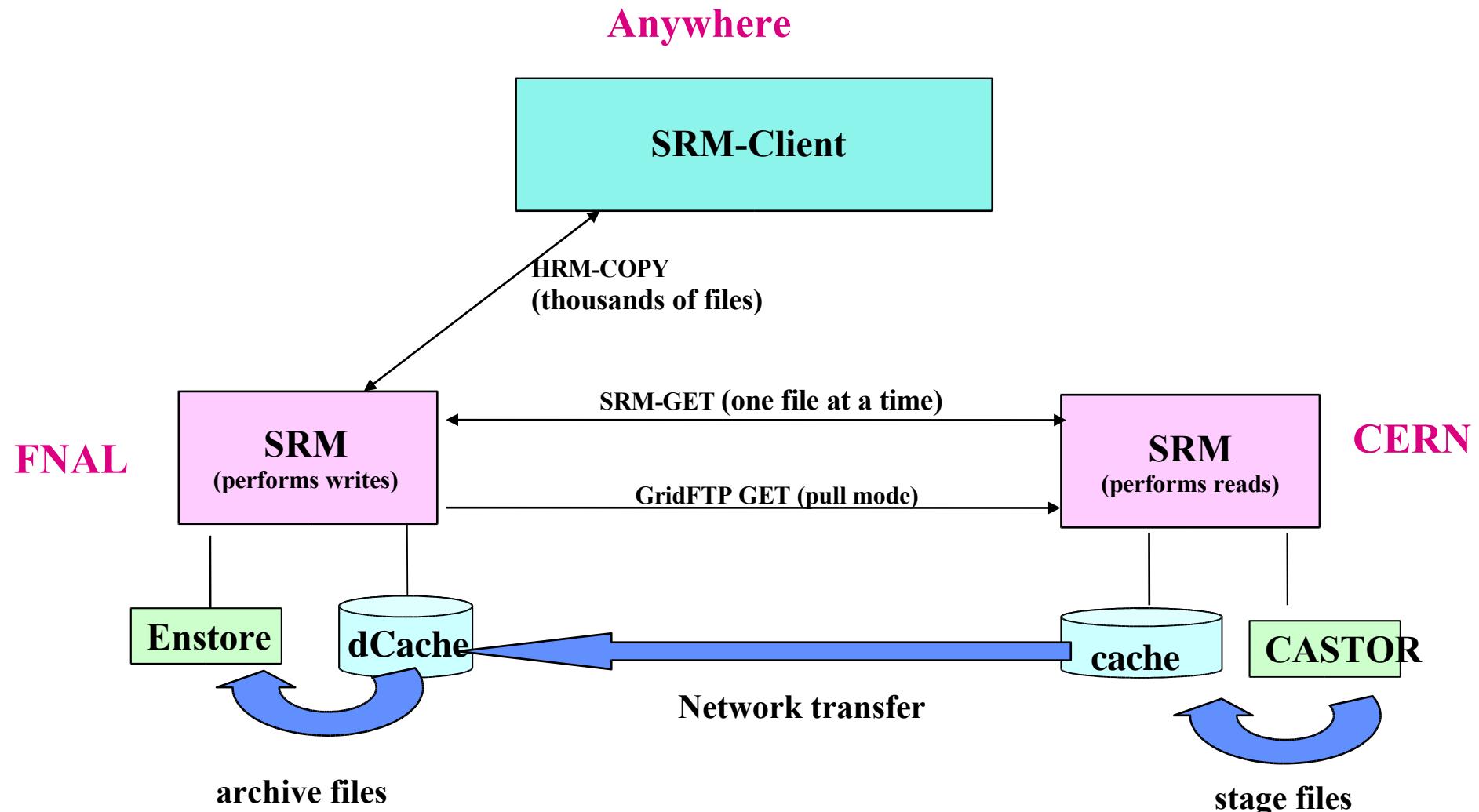
## Permission

- srmSetPermission
- srmReassignToUser
- srmCheckPermission

# Srm Example 1 - srmPrepareToPut



# Srm Example - Robust Replication



# SRM Interface Details

## Space Management Functions

- SrmReserveSpace
- SrmReleaseSpace
- srmUpdateSpace
- srmCompactSpace
- srmGetSpaceMetaData
- srmChangeFileStorageType
- srmGetSpaceToken

## Directory

- SrmMkdir
- srmRmdir
- srmRm
- srmLs
- srmMv

## Data transfer functions

- **srmPrepareToGet**
- SrmPrepareToPut
- **srmCopy**
- SrmRemoveFiles
- srmReleaseFiles
- srmPutDone
- srmAbortRequest
- srmAbortFiles
- srmSuspendRequest
- srmResumeRequest

## Status Functions

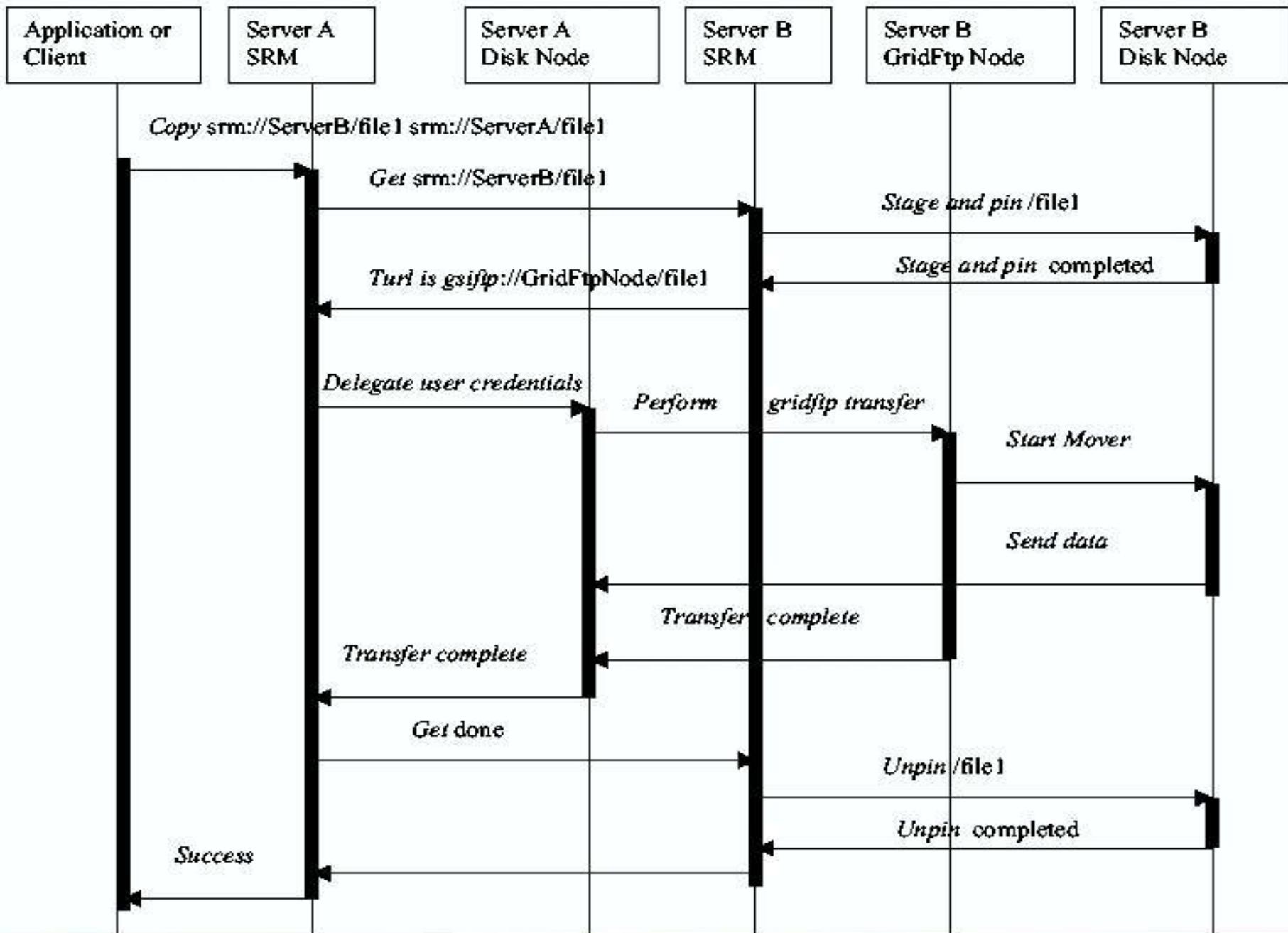
- **srmStatusOfGetRequest**
- srmStatusOfPutRequest
- **srmStatusOfCopyRequest**
- srmGetRequestSummary
- srmExtendFileLifeTime
- SrmGetRequestID

## Permission

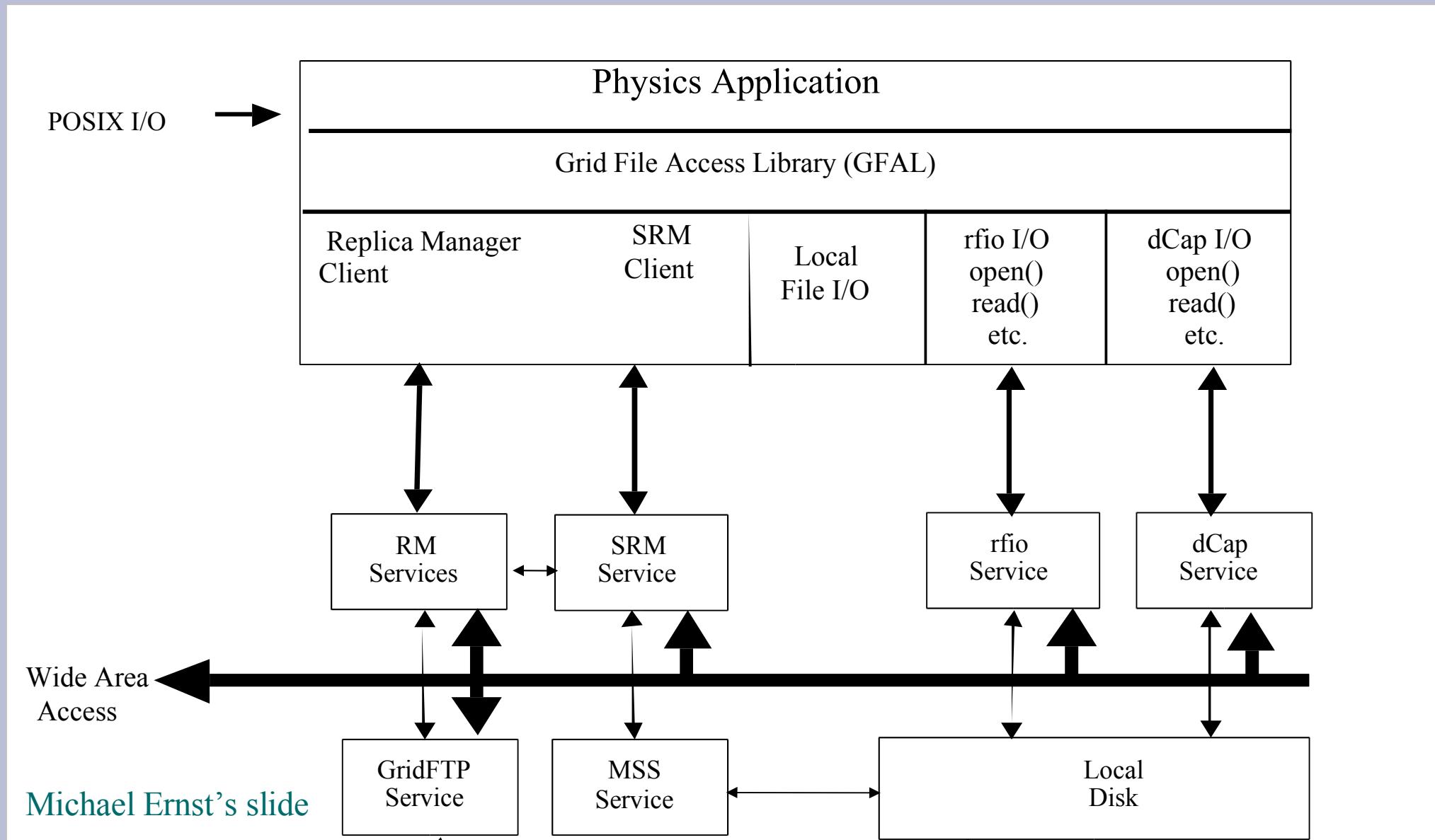
- srmSetPermission
- srmReassignToUser
- srmCheckPermission



# The sequence diagram of the SRM Copy Function performing “Copy srm://ServerB/file1 srm://ServerA/file1”

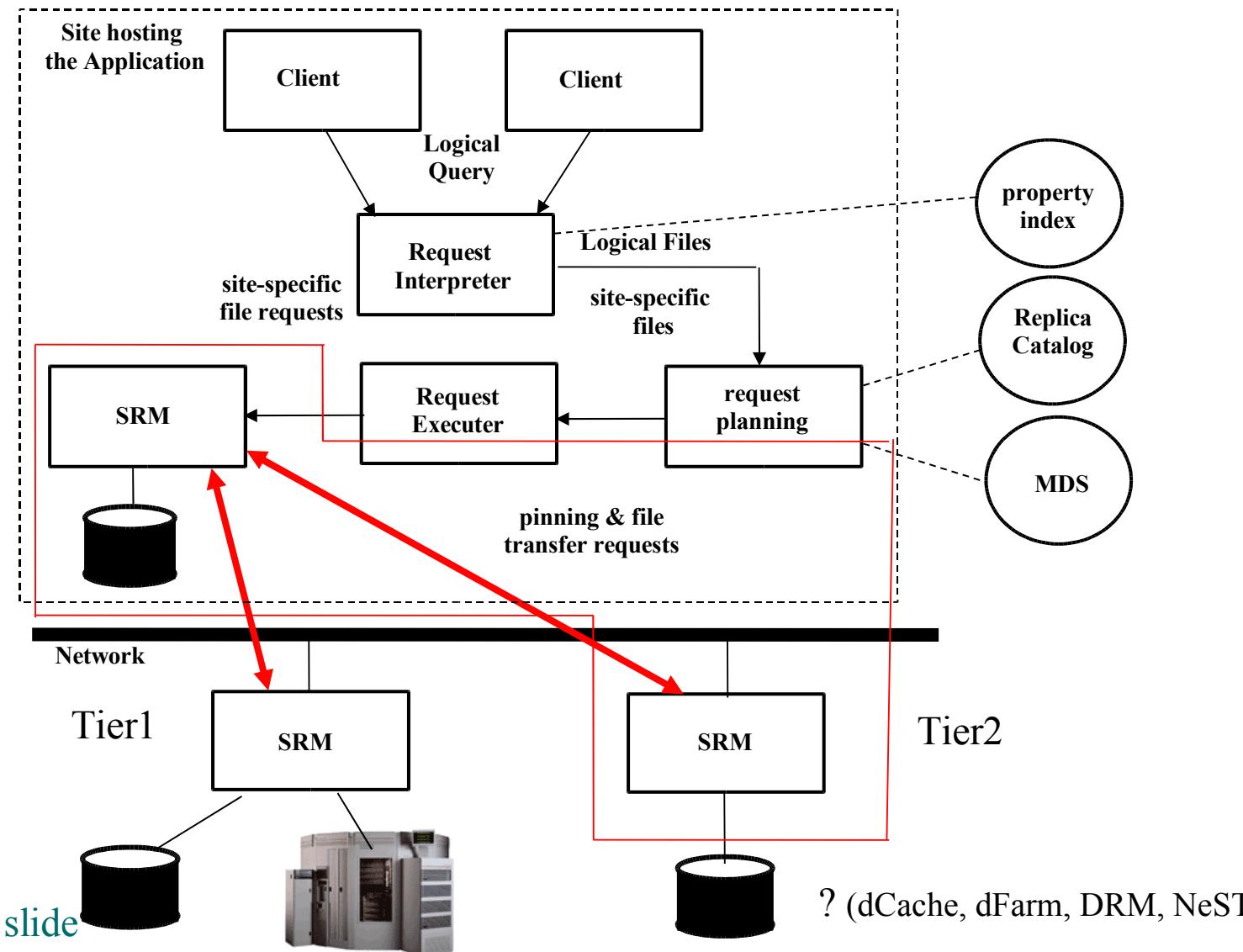


# Grid File Access Library and SRM



Michael Ernst's slide

# Client Access to CMS-DATA



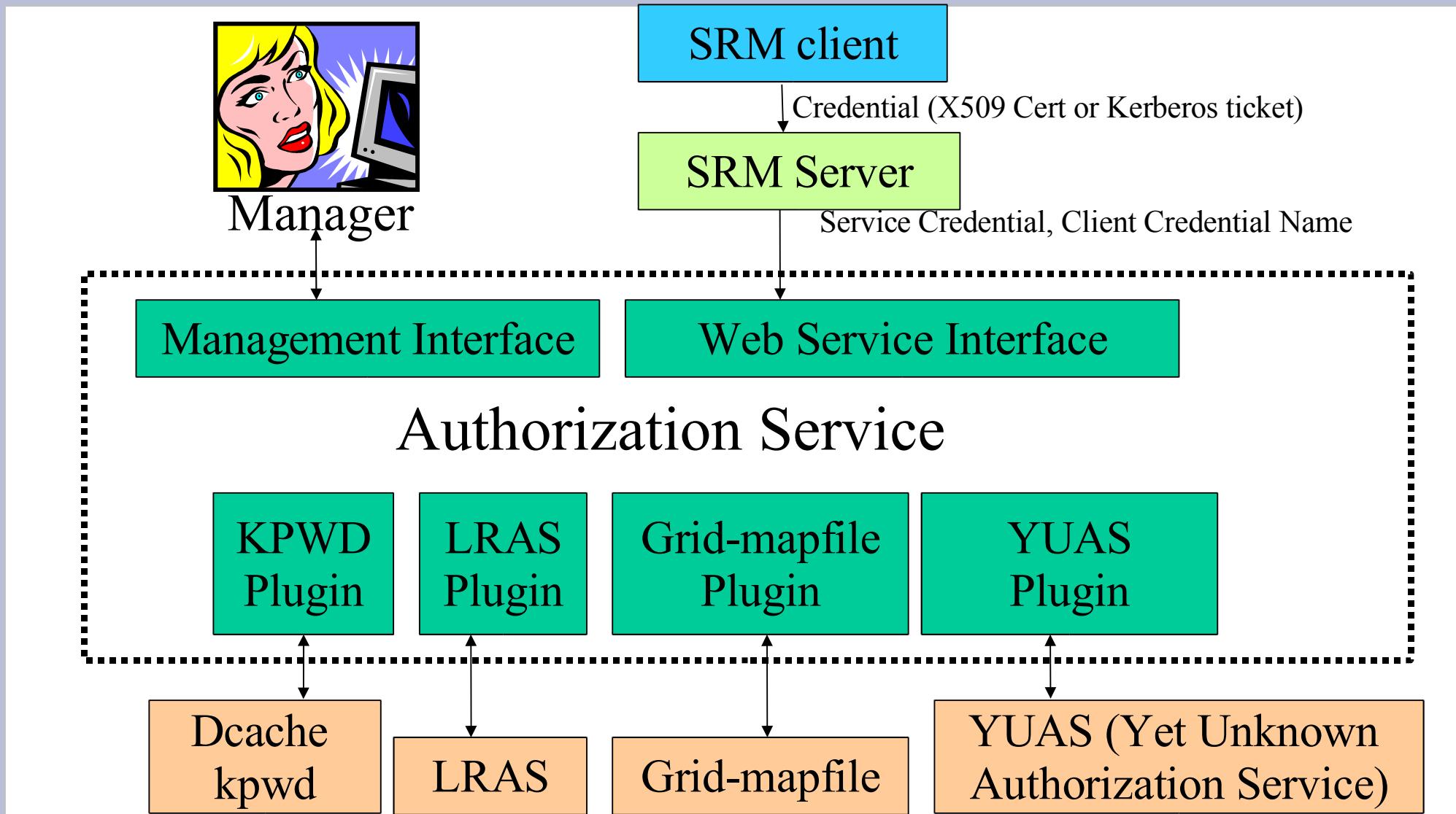
# Storage Authorization Requirements

- Multiple Authorization mechanisms exist:
  - dCache kpwd
  - Grid-mapfile
  - Local Resource Administration Service (LRAS) and other VO like services.
  - YUAS (Yet Unknown Authorization Service)
  - Example: US-CMS is using kpwd and grid-mapfile
- Storage systems such as dCache and Enstore are distributed services
  - multiple access points (doors)
  - authorization management is localized to each participating host

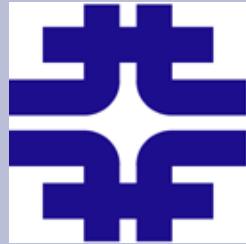
Need a centralized service with a well defined interface, which can be used by multiple systems

Need to support some or all of the above (pluggable auth. Mechanism architecture)

# Storage Authorization Architecture

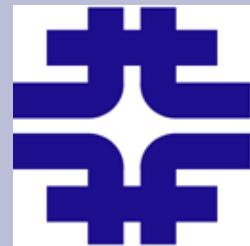


# Status of Fermilab SRM Implementation



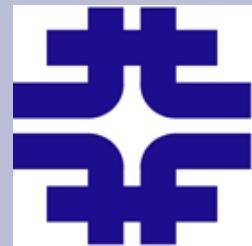
- SRM Interface to dCache Storage System
  - Data Transfer Functions (get, put and copy)
  - Load balancing, throttling, fairness
  - Scalable replication mechanism via gridftp
  - Automatic directory creation
  - Fault tolerance and reliability achieved by providing persistent storage for transfer requests and retries on failures
- SRM interface as a standalone product, adaptable to work on top of another storage system through a SRM-Storage interface
- A reference implementation of the SRM-Storage interface to a Unix File System

# Fermilab SRM Implementation Plans (1)

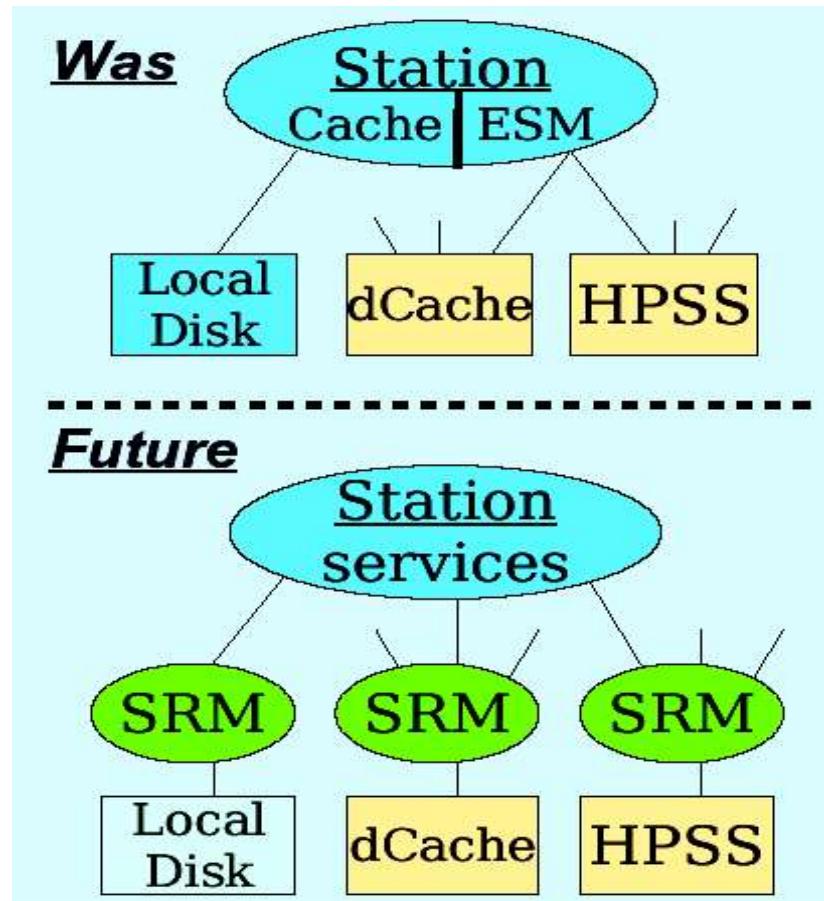


- Space Management
  - First implicit as a part of SRM Version 1.1
  - Then explicit, through the SRM V2.1
- Full implementation of SRM Version 2.1 interface
- International Lattice QCD (ILDG)
- Research utilization of Lambda Station Interface by a Storage System. Lambda Station gives optical path allocation and per flow routing
- Monitoring, Administration and Accounting interfaces

# Fermilab SRM Implementation Plans (2)

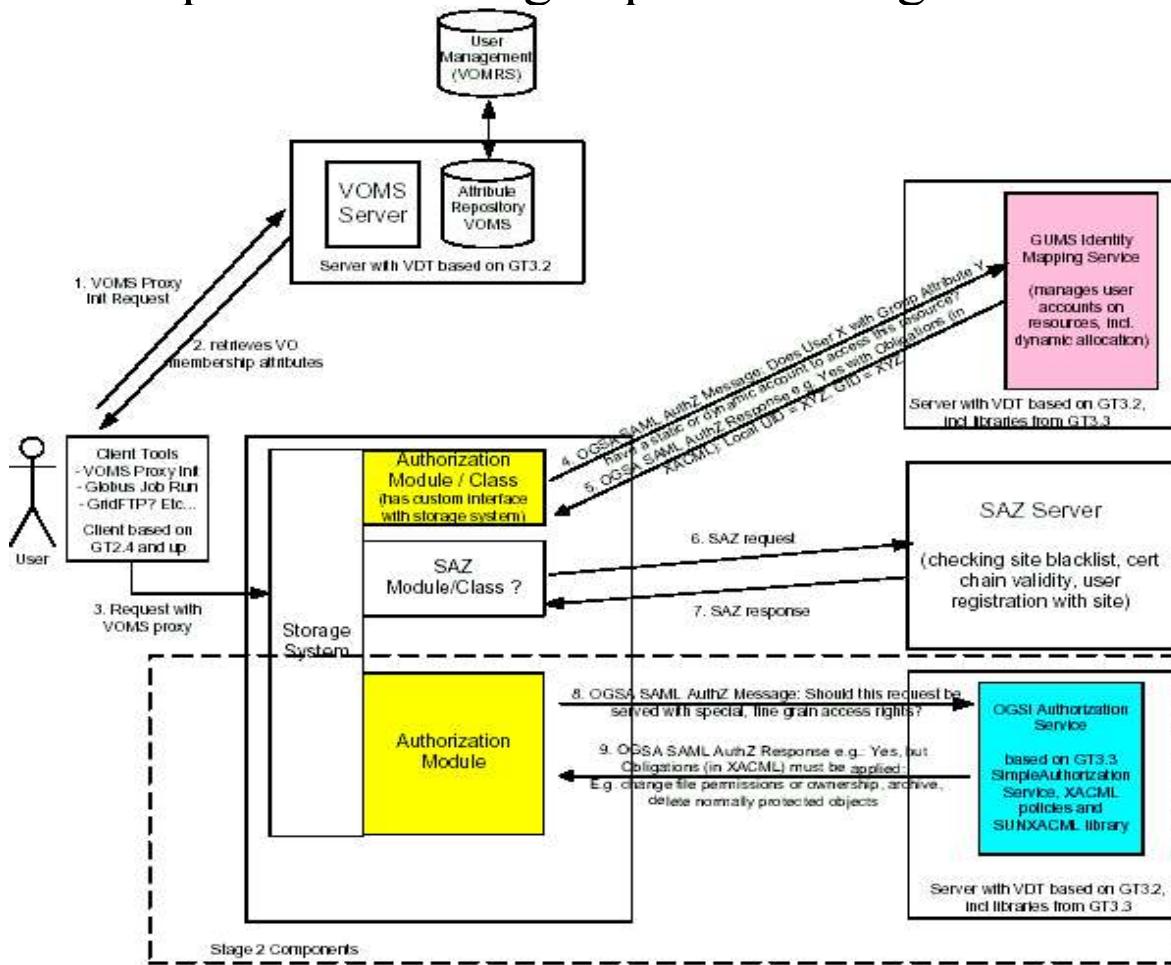


- Integration of SRM to SAMGrid Data Handling System as the universal interface to a storage



# Fermilab SRM Implementation Plans (3)

- Integration with Virtual Organization based Authorization Services, development of Storage Specific fine grain Authorization Services



# Resources

- The Storage Resource Manager Collaboration, <http://sdm.lbl.gov/srm-wg/>
- Fermilab SRM Project , <http://www-isd.fnal.gov/srm>
- Patrick Fuhrmann, dCache, Grid Storage Element and enhanced use cases,  
<http://indico.cern.ch/contributionDisplay.py?contribId=233&sessionId=10&confId=0>
- DCache, Disk Cache Mass Storage System, <http://www.dcache.org/>
- US-CMS, <http://www.uscms.org/>
- Don Petravick, Lambda Station Proposal, <http://hpc.fnal.gov/wawg/omnibus-text.pdf>
- Arie Shoshany, Replica Registration Service,  
<http://www.ppdg.net/mtgs/28jun04-wb/slides/PPDG-AH-0406-RMS-RRS.ppt>
- Robert Kennedy, SAMGrid Integration of SRMs,  
<http://indico.cern.ch/contributionDisplay.py?contribId=460&sessionId=7&confId=0>
- Michael Ernst, US-CMS Grid File Access Proposal,  
<http://www.uscms.org/sandc/reviews/doe-nsf/2003-07/docs/GFA-Proposal-Short-v1.0.pdf>
- Michael Ernst, Managed Data Storage and Data Access Services for Data Grids,  
<http://indico.cern.ch/contributionDisplay.py?contribId=190&sessionId=7&confId=0>
- Philip DeMar, LambdaStation: A forwarding and admission control service to interface production network facilities with advanced research network paths  
<http://indico.cern.ch/contributionDisplay.py?contribId=359&sessionId=11&confId=0>