Storage Resource Management at Fermilab

Vanderbilt University
IBP/Grid Workshop (April 7-8, 2005)

Timur Perelmutov (FNAL)
Robert Kennedy (FNAL)

SRM Collaboration

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

VANDERBILT UNIVERSITY

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

VANDERBILT UNIVERSITY

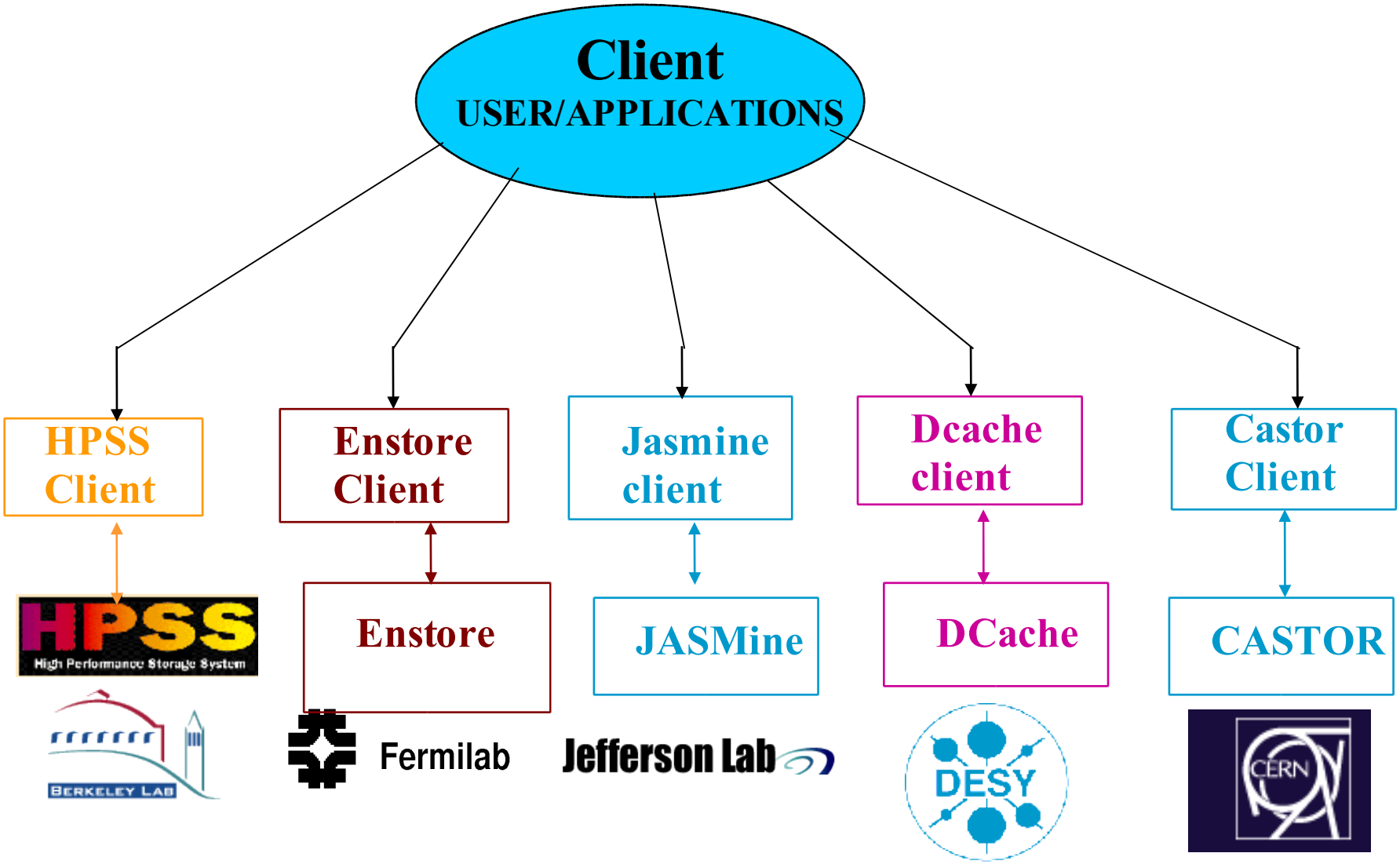
SRM Motivation (2)

High Energy Physics Collaborations span multiple institutions where

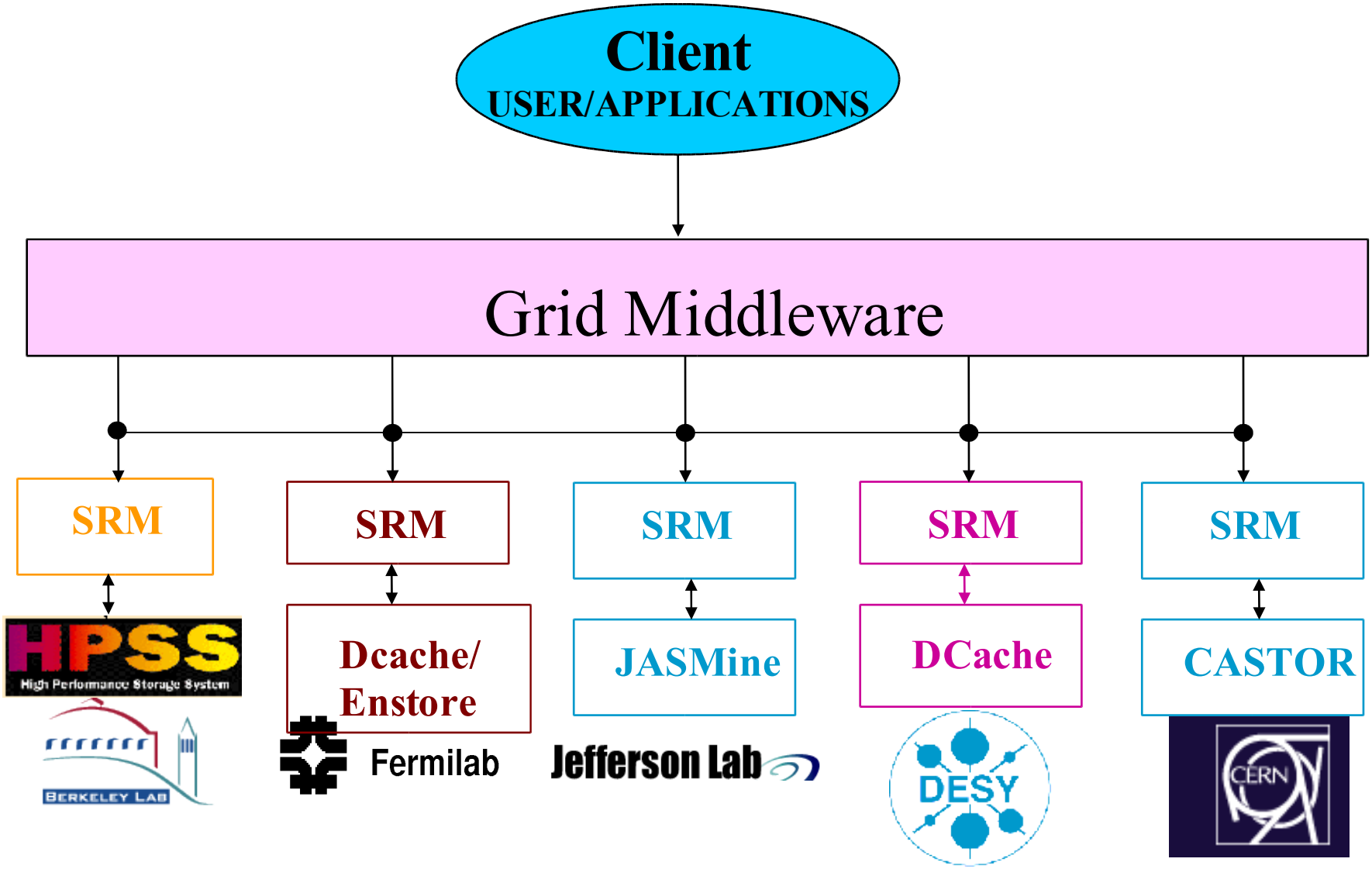
- A large variety of types of Storage Systems exist
  - Single Disk / Raid
  - Robotic Tape Storage System (Enstore, HPSS)
  - Distributed Disk Cache (dCache)
  - Hierarchical Storage System (dCache - Enstore)
- Heterogeneous environments and proliferation of custom Mass Storage Systems (MSSs) exist
- 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



VANDERBILT UNIVERSITY

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

VANDERBILT UNIVERSITY

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

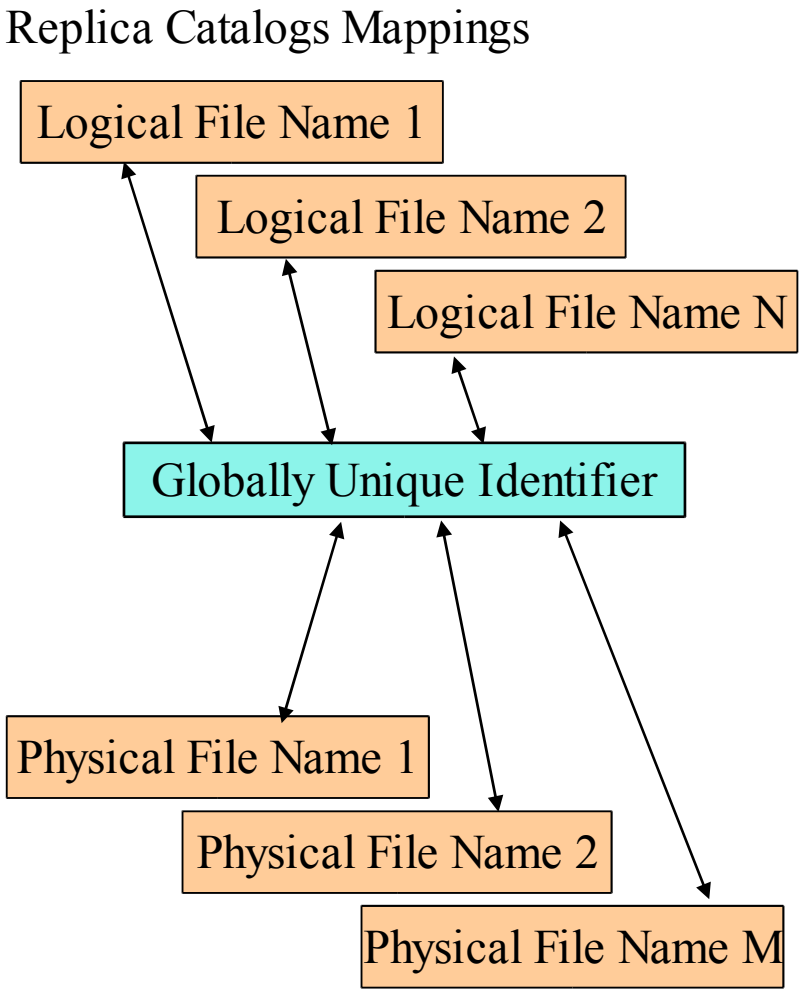
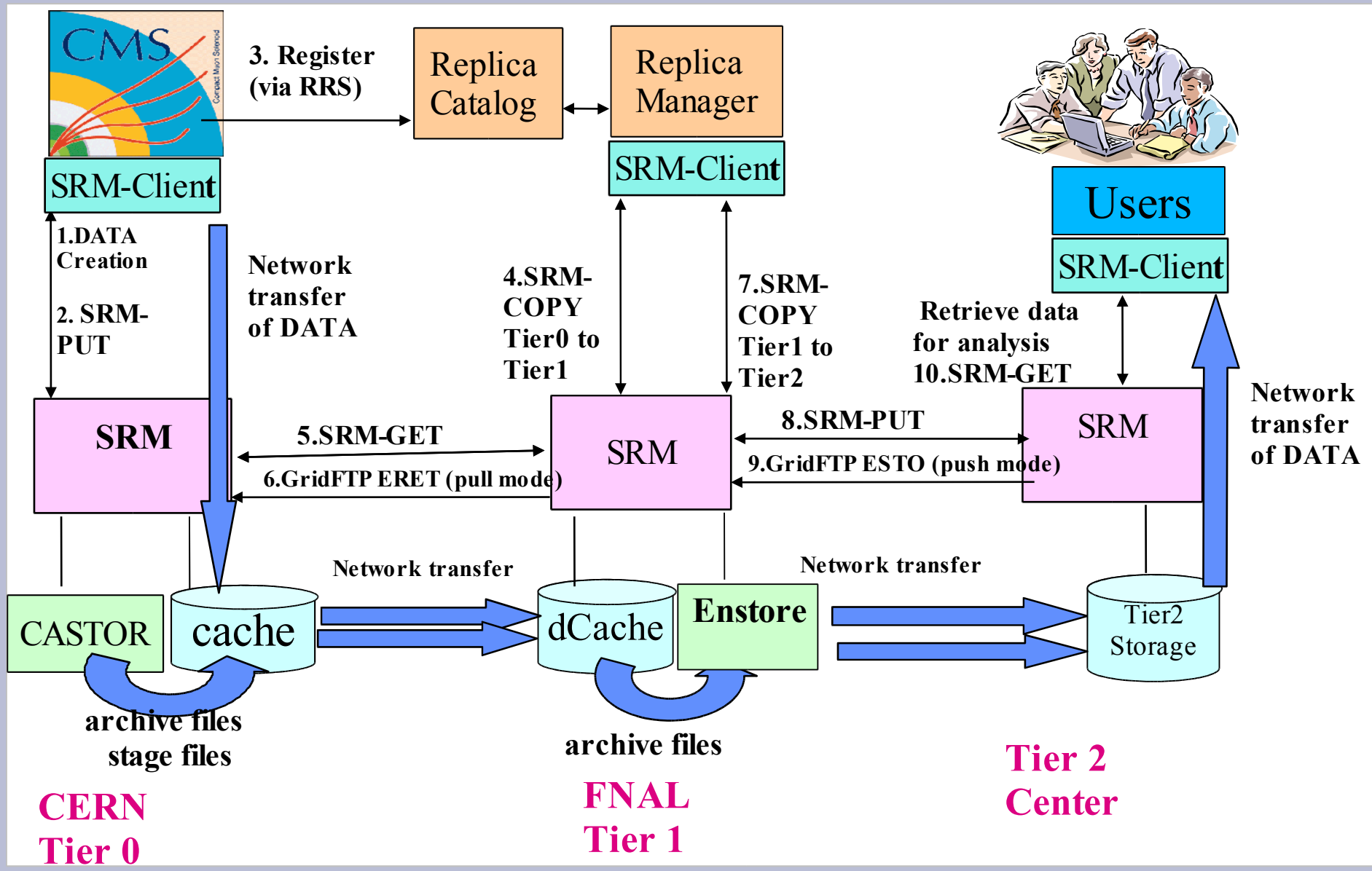


Illustration
US-CMS DATAGRID

- 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 v1.1provides
    - 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

VANDERBILT UNIVERSITY

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

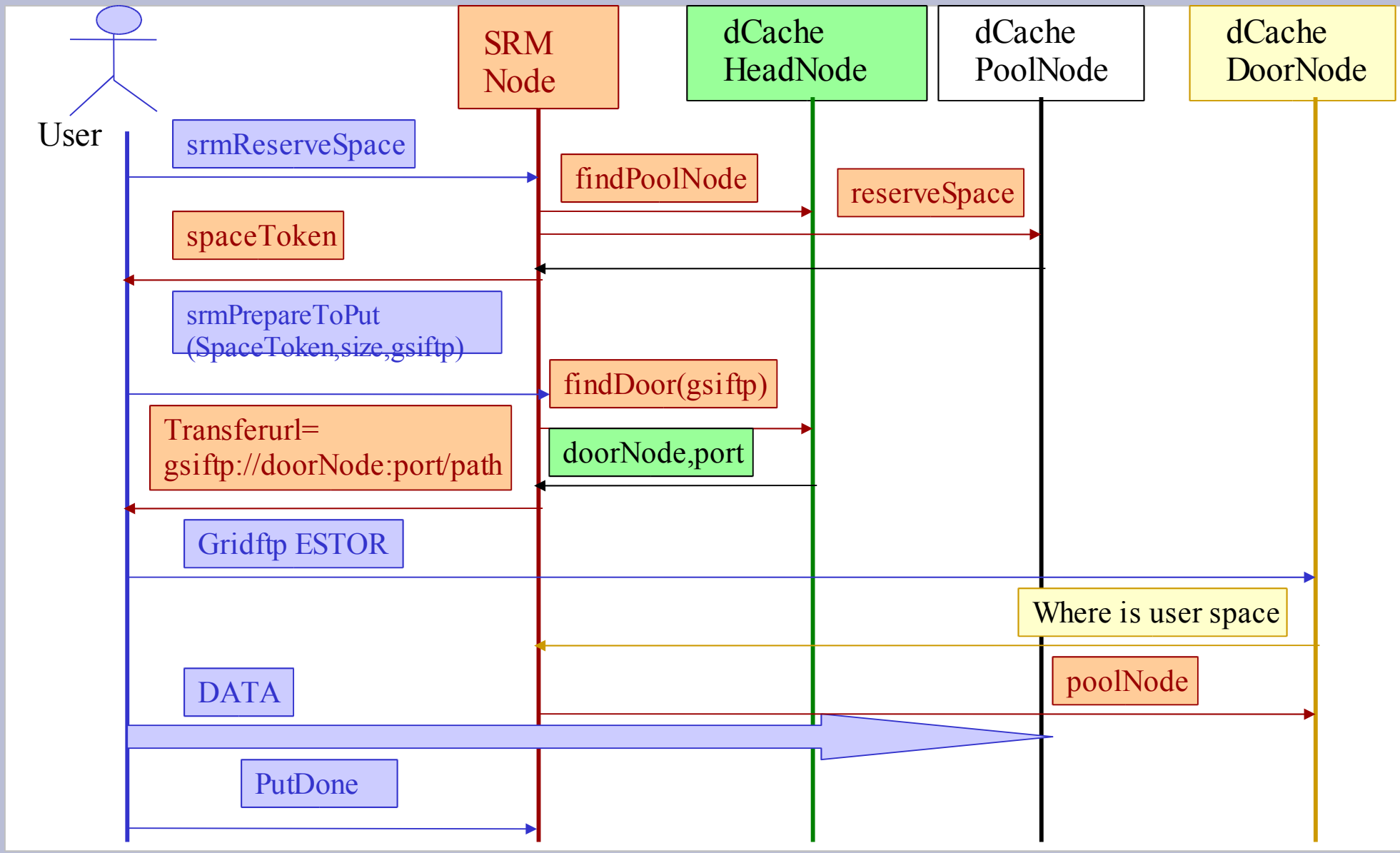
Directory
- SrmMkdir
- srmRmdir
- srmRm
- srmLs
- srmMv

Data transfer functions
- srmPrepareToGet
- SrmPrepareToPut
- srmCopy
- SrmRemoveFiles
- srmReleaseFiles
- srmPutDone
- srmAbortRequest
- srmAbortFiles
- srmSuspendRequest
- srmResumeRequest

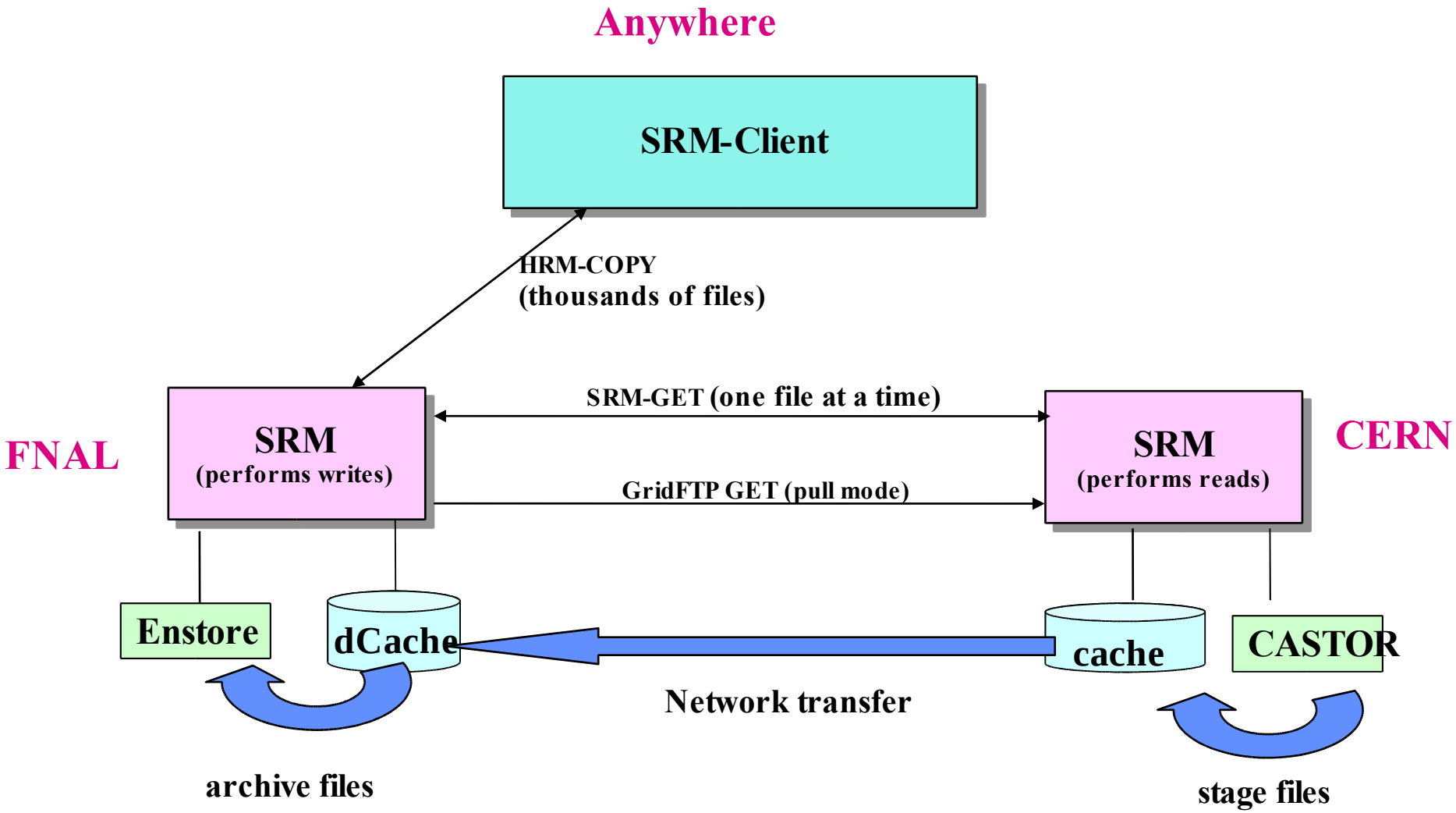
Status Functions
- srmStatusOfGetRequest
- srmStatusOfPutRequest
- srmStatusOfCopyRequest
- srmGetRequestSummary
- srmExtendFileLifeTime
- SrmGetRequestID

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

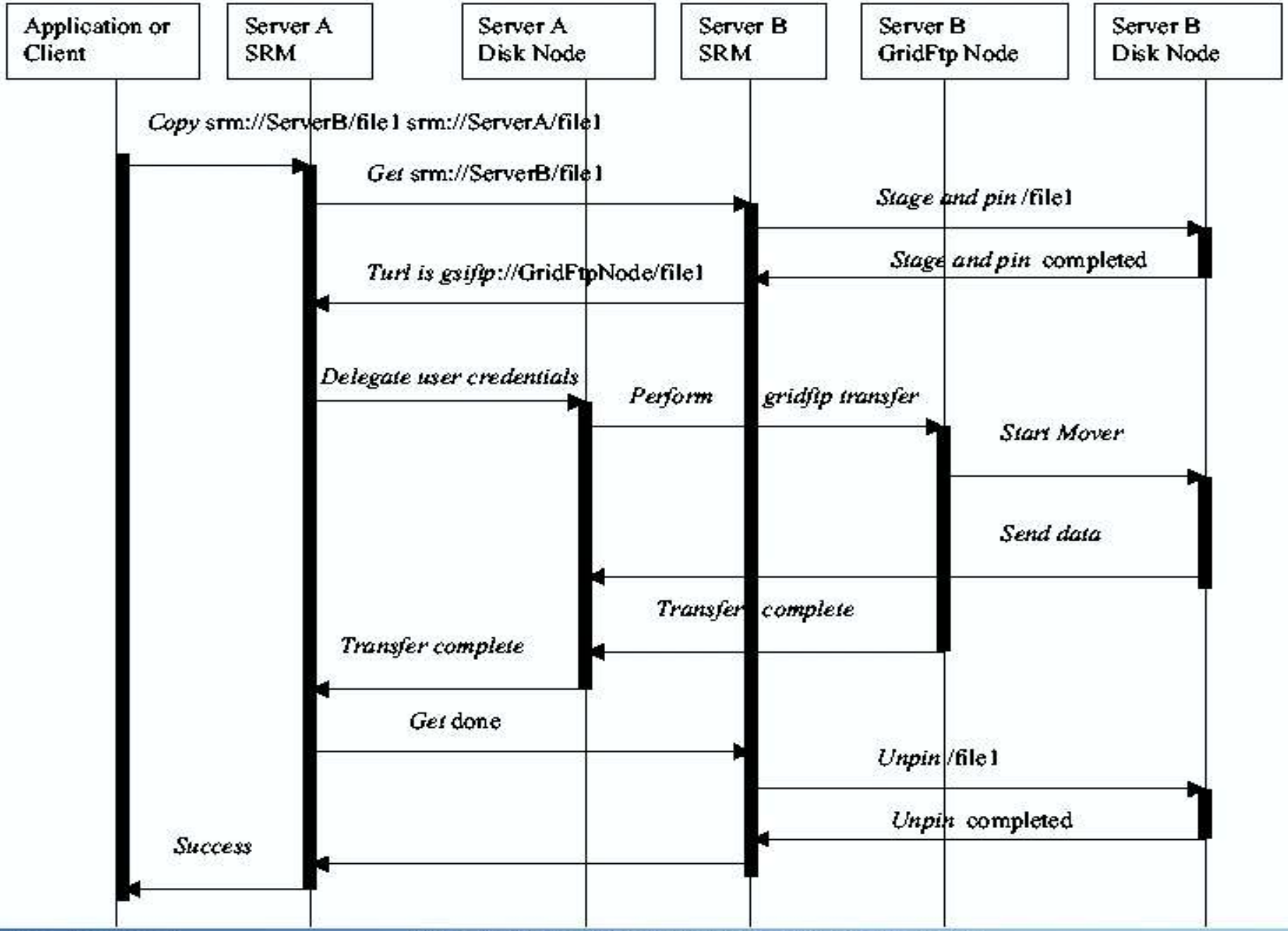
Directory
- SrmMkdir
- srmRmdir
- srmRm
- srmLs
- srmMv

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"



VANDERBILT UNIVERSITY

Features and Plans

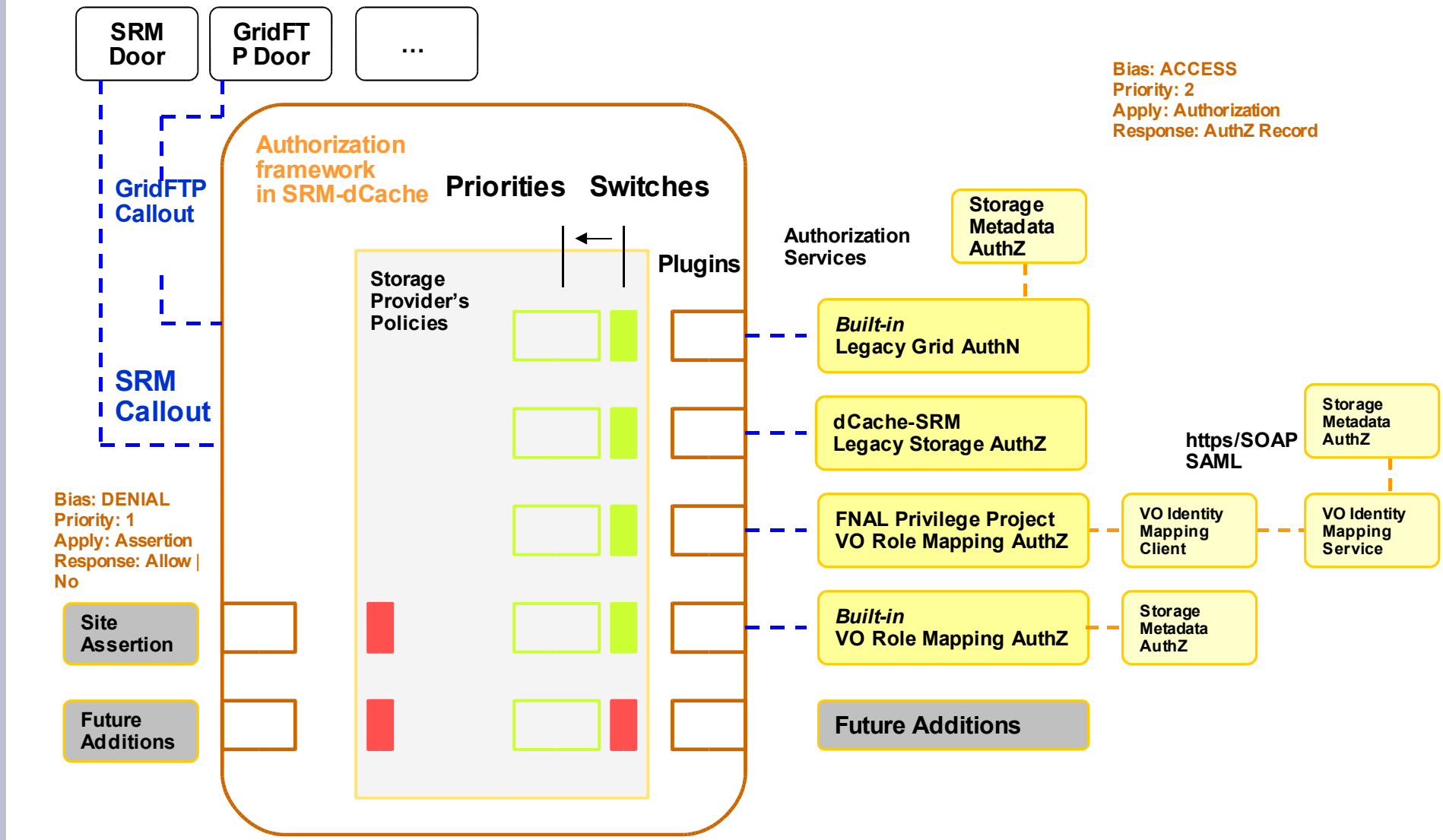
- Implicit Space Reservation: Initial version deployed. More to do.
  - Space reserved for a single file transfer, implicit in get/put/copy.
  - Reservation is not long-term, but just for time of transfer.
- SRM V2.1 API: Prototyped. Initial version being developed.
  - First feature we will deploy: srm-ls. High priority for our clients.
  - Expect this to be ready in 1-2 months, features released as available.
- Auditing, Accounting, Monitoring: SRM schema is "growing".
  - Already have GRIS-GIIS path working: total space and space used.
  - Will coordinate with parallel efforts to define initial schema.
  - Need SRM DB interface, query tools, path to MonAlisa for full A&A.
- Broader Client Shell Toolset: Depends on our V2.1 API work
  - V1.1 tools already in LCG dCache distribution (Michael Ernst)
  - Preparing for distribution of V1.1 tools in OSG channel as well.

Features and Plans 2

- VO Authorization Module Integration: Work done by Abhishek Singh Rana in context of SRM-dCache.
  - Pluggable Authorization Mechanism architecture
  - Priorities and switches allow fine-tuning of authorization policy
  - Integrated with Site-Wide VO Authorization Service
  - This is the long-term Grid solution, adapting to existing practice. US-CMS T1 uses scripts and VO tools now, until this is ready.
  - Discussions begun: mapping high-level storage authorization policies to performant low-level mechanisms (ACLs, ...)
- Storage Authorization is superset of existing VO Authorization
  - Some storage systems (dCache, ...) need to know "where" in the SE namespace that a username-role is authorized to work, for instance.
  - Critical in a multi-VO SE, but still important in a VO-specific SE. One physics group should not be able to affect another's data storage.

VO Authorization Module Architecture

(slide by Abhishek Singh Rana)



VANDERBILT UNIVERSITY

Resources

- FNAL SRM implementation proven in field
  - SRM-dCache for US-CMS T1 was successful in SC2
- FNAL SRM at API V1.1, going to V2.1
  - More features, functionality in 1-2 months
- Planning for wider use: SRM service, SRM client shell tools
  - SRM-dCache use is expanding: LCG, US-CMS, OSG channels.
  - Can be used with other storage systems too.
  - Expanding staff, and "external" contributors welcome (open source).
- *"Storage is hard."*
  - Storage maintains state where compute worker nodes do not.
  - Access to storage meta-data must be performant, not just access to data.
- *"Agree to APIs and protocols, compete on implementations."*
  - Healthy variety of storage solutions here at workshop. Customers win! 8^)
- FNAL-SRM link: http://www-isd.fnal.gov/srm