Grid, Storage and SRM

Introduction

Open Science Grid

Storage and Grid

- Grid applications need to reserve and schedule
  - Compute resources
  - Network resources
  - Storage resources
- Furthermore, they need
  - Monitor progress status
  - Release resource usage when done
- For storage resources, they need
  - To put/get files into/from storage spaces
  - Unlike compute/network resources, storage resources are not available when jobs are done
  - files in spaces need to be managed as well
    - Shared, removed, or garbage collected

Open Science Grid

Motivation & Requirements (1)

- Suppose you want to run a job on your local machine
  - Need to allocate space
  - Need to bring all input files
  - Need to ensure correctness of files transferred
  - Need to monitor and recover from errors
  - What if files don’t fit space?
    - Need to manage file streaming
  - Need to remove files to make space for more files

Motivation & Requirements (2)

- Now, suppose that the machine and storage space is a shared resource
  - Need to do the above for many users
  - Need to enforce quotas
  - Need to ensure fairness of space allocation and scheduling

Motivation & Requirements (3)

- Now, suppose you want to run a job on a Grid
  - Need to access a variety of storage systems
  - mostly remote systems, need to have access permission
  - Need to have special software to access mass storage systems

Motivation & Requirements (4)

- Now, suppose you want to run distributed jobs on the Grid
  - Need to allocate remote spaces
  - Need to move files to remote sites
  - Need to manage file outputs and their movement to destination sites

Storage Resource Managers

Open Science Grid

What is SRM?

- Storage Resource Managers (SRMs) are middleware components
  - whose function is to provide
    - dynamic space allocation
    - file management

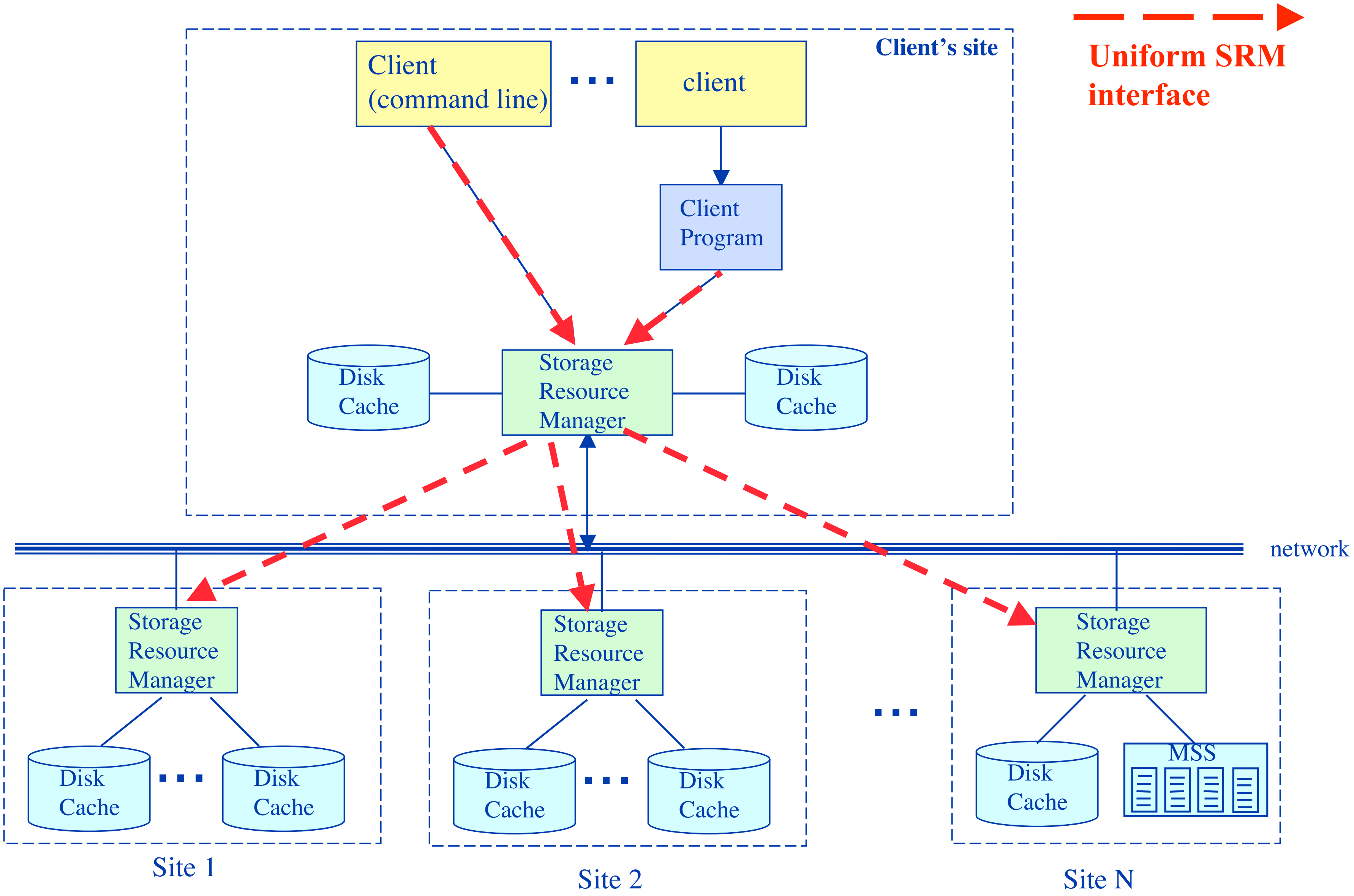
    on shared storage resources on the Grid
  - Different implementations for underlying storage systems are based on the same SRM specification

Open Science Grid

SRMs role in grid

- SRMs role in the data grid architecture
  - Shared storage space allocation & reservation
    - important for data intensive applications
  - Get/put files from/into spaces
    - archived files on mass storage systems
  - File transfers from/to remote sites, file replication
  - Negotiate transfer protocols
  - File and space management with lifetime
  - support non-blocking (asynchronous) requests
  - Directory management
  - Interoperate with other SRMs

Client and Peer-to-Peer Uniform Interface



History

- 7 year of Storage Resource Management (SRM) activity
- Experience with system implementations v.1.1 (basic SRM) – 2001
  - MSS: Castor (CERN), dCache (FNAL, DESY), HPSS (LBNL, ORNL, BNL), JasMINE (Jlab), MSS (NCAR)
  - Disk systems: dCache (FNAL), DPM (CERN), DRM (LBNL)
- SRM v2.0 spec – 2003
- SRM v2.2 – enhancements introduced after WLCG (the World-wide LHC Computing Grid) adopted SRM standard
  - Several implementations of v2.2
  - Extensive compatibility and interoperability testing
  - MSS: Castor (CERN, RAL), dCache/{Enstore,TSM,OSM,HPSS} (FNAL, DESY), HPSS (LBNL), JasMINE (Jlab), SRB (SINICA, SDSC)
  - Disk systems: BeStMan (LBNL), dCache (FNAL, DESY), DPM (CERN), StoRM (INFN/CNAF, ICTP/EGRID)
- Open Grid Forum (OGF)
  - Grid Storage Management (GSM-WG) at GGF8, June 2003
  - SRM collaboration F2F meeting – Sept. 2006
  - SRM v2.2 spec on OGF recommendation track – Dec. 2007

Who’s involved…

- CERN, European Organization for Nuclear Research, Switzerland
- Deutsches Elektronen-Synchrotron, DESY, Hamburg, Germany
- Fermi National Accelerator Laboratory, Illinois, USA
- ICTP/EGRID, Italy
- INFN/CNAF, Italy
- Lawrence Berkeley National Laboratory, California, USA
- Rutherford Appleton Laboratory, Oxfordshire, England
- Thomas Jefferson National Accelerator Facility, Virginia, USA

SRM : Concepts

SRM: Main concepts

- Space reservations
- Dynamic space management
- Pinning file in spaces
- Support abstract concept of a file name: Site URL
- Temporary assignment of file names for transfer: Transfer URL
- Directory management and authorization
- Transfer protocol negotiation
- Support for peer to peer request
- Support for asynchronous multi-file requests
- Support abort, suspend, and resume operations
- Non-interference with local policies

Site URL and Transfer URL

- Provide: Site URL (SURL)
  - URL known externally – e.g. in Replica Catalogs
  - e.g. srm://ibm.cnaf.infn.it:8444/dteam/test.10193
- Get back: Transfer URL (TURL)
  - Path can be different from SURL – SRM internal mapping
  - Protocol chosen by SRM based on request protocol preference
  - e.g. gsiftp://ibm139.cnaf.infn.it:2811//gpfs/sto1/dteam/test.10193
- One SURL can have many TURLs
  - Files can be replicated in multiple storage components
  - Files may be in near-line and/or on-line storage
  - In a light-weight SRM (a single file system on disk)
    - SURL may be the same as TURL except protocol
- File sharing is possible
  - Same physical file, but many requests
  - Needs to be managed by SRM implementation

Transfer protocol negotiation

- Negotiation
  - Client provides an ordered list of preferred transfer protocols
  - SRM returns first protocol from the list it supports
  - Example
    - Client provided protocols list: bbftp, gridftp, ftp
    - SRM returns: gridftp
- Advantages
  - Easy to introduce new protocols
  - User controls which transfer protocol to use
- How it is returned?
  - The protocol of the Transfer URL (TURL)
  - Example: bbftp://dm.slac.edu//temp/run11/File678.txt

Types of storage and spaces

- Access latency
  - On-line
    - Storage where files are moved to before their use
  - Near-line
    - Requires latency before files can be accessed
- Retention quality
  - Custodial (High quality)
  - Output (Middle quality)
  - Replica (Low Quality)
- Spaces can be reserved in these storage components
  - Spaces can be reserved for a lifetime
  - Space reference handle is returned to client – space token
  - Total space of each type are subject to local SRM policy and/or VO policies
- Assignment of files to spaces
  - Files can be assigned to any space, provided that their lifetime is shorter than the remaining lifetime of the space

Open Science Grid

Managing spaces

- Default spaces
  - Files can be put into an SRM without explicit reservation
  - Default spaces are not visible to client
- Files already in the SRM can be moved to other spaces
  - By srmChangeSpaceForFiles
- Files already in the SRM can be pinned in spaces
  - By requesting specific files (srmPrepareToGet)
  - By pre-loading them into online space (srmBringOnline)
- Updating space
  - Resize for more space or release unused space
  - Extend or shorten the lifetime of a space
- Releasing files from space by a user
  - Release all files that user brought into the space whose lifetime has not expired
  - Move permanent and durable files to near-line storage if supported
  - Release space that was used by user

Space reservation

- Negotiation
  - Client asks for space: Guaranteed_C, MaxDesired
  - SRM return: Guaranteed_S <= Guaranteed_C,
    best effort <= MaxDesired
- Types of spaces
  - Specified during srmReserveSpace
  - Access Latency (Online, Nearline)
  - Retention Policy (Replica, Output, Custodial)
  - Subject to limits per client (SRM or VO policies)
  - Default: implementation and configuration specific
- Lifetime
  - Negotiated: Lifetime_C requested
  - SRM return: Lifetime_S <= Lifetime_C
- Reference handle
  - SRM returns space reference handle (space token)
  - Client can assign Description
  - User can use srmGetSpaceTokens to recover handles on basis of ownership

Open Science Grid

Directory management

- Usual unix semantics
  - srmLs, srmMkdir, srmMv, srmRm, srmRmdir
- A single directory for all spaces
  - No directories for each file type
  - File assignment to spaces is virtual
- Access control services
  - Support owner/group/world permission
    - ACLs supported – can have one owner, but multiple user and group access permissions
    - Can only be assigned by owner
    - When file is requested from a remote site, SRM should check permission with source site

Advanced concepts

- Composite Storage Element
  - Made of multiple Storage Components
    - e.g. component 1: online-replica
      component 2: nearline-custodial (with online disk cache)
    - e.g. component1: online-custodial
      component 2: nearline-custodial (with online disk cache)
  - srmBringOnline can be used to temporarily bring data to the online component for fast access
  - When a file is put into a composite space, SRM may have (temporary) copies on any of the components.
- Primary Replica
  - When a file is first put into an SRM, that copy is considered as the primary replica
  - A primary replica can be assigned a lifetime
  - The SURL lifetime is the lifetime of the primary replica
  - When other replicas are made, their lifetime cannot exceed the primary replica lifetime
  - Lifetime of a primary replica can only be extended by an SURL owner.

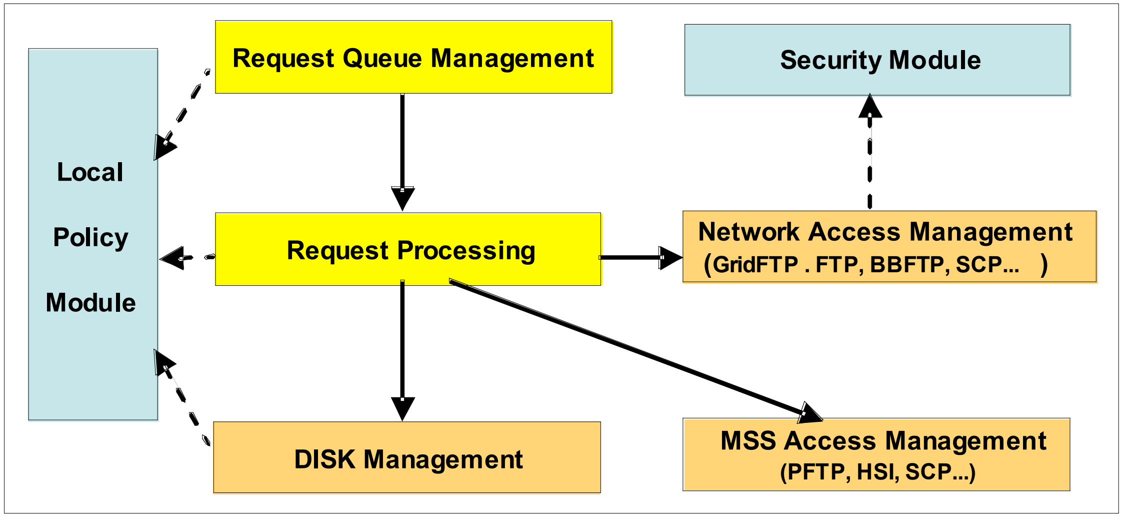
SRM v2.2 Interface

- *Data transfer functions* to get files into SRM spaces from the client's local system or from other remote storage systems, and to retrieve them
  - srmPrepareToGet, srmPrepareToPut, srmBringOnline, srmCopy
- *Space management functions* to reserve, release, and manage spaces, their types and lifetimes.
  - srmReserveSpace, srmReleaseSpace, srmUpdateSpace, srmGetSpaceTokens
- *Lifetime management functions* to manage lifetimes of space and files.
  - srmReleaseFiles, srmPutDone, srmExtendFileLifeTime
- *Directory management functions* to create/remove directories, rename files, remove files and retrieve file information.
  - srmMkdir, srmRmdir, srmMv, srmRm, srmLs
- *Request management functions* to query status of requests and manage requests
  - srmStatusOf{Get,Put,Copy,BringOnline}Request, srmGetRequestSummary, srmGetRequestTokens, srmAbortRequest, srmAbortFiles, srmSuspendRequest, srmResumeRequest
- Other functions include Discovery and Permission functions
  - srmPing, srmGetTransferProtocols, srmCheckPermission, srmSetPermission, etc.

SRM implementations

Berkeley Storage Manager (BeStMan) LBNL

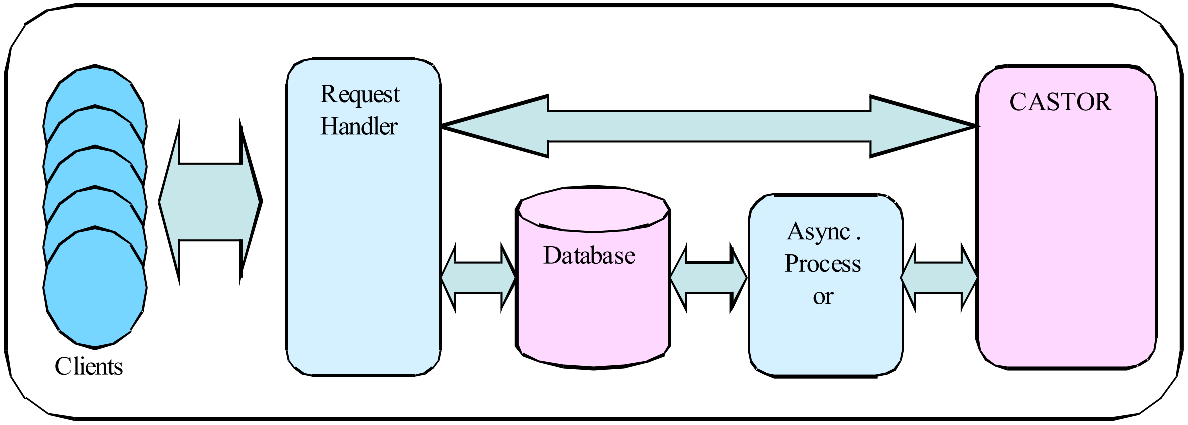
- Java implementation
- Designed to work with unix-based disk systems
- As well as MSS to stage/archive from/to its own disk (currently HPSS)
- Adaptable to other file systems and storages (e.g. NCAR MSS, VU L-Store, TTU Lustre, NERSC GFS)
- Uses in-memory database (BerkeleyDB)


Request Queue Management
Security Module
Local Policy Module
Request Processing
Network Access Management (GridFTP . FTP, BBFTP, SCP...)
DISK Management
MSS Access Management (PFTP, HSI, SCP...)

- Multiple transfer protocols
- Space reservation
- Directory management (no ACLs)
- Can copy files from/to remote SRMs or GridFTP Servers
- Can copy entire directory recursively
  - Large scale data movement of thousands of files
  - Recovers from transient failures (e.g. MSS maintenance, network down)

- Local Policy
  - Fair request processing
  - File replacement in disk
  - Garbage collection

Castor-SRM
CERN and Rutherford Appleton Laboratory

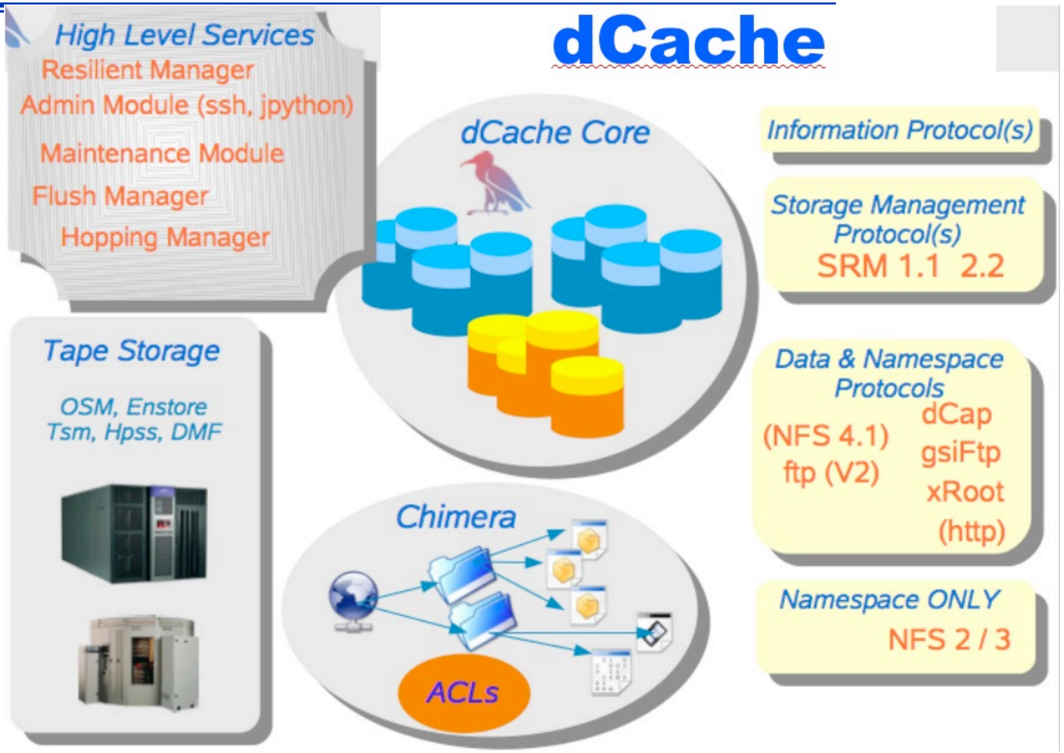


- **CASTOR is the HSM in production at CERN**
- **Support for multiple tape robots**
  - Support for Disk-only storage recently added
- **Designed to meet Large Hadron Collider Computing requirements**
  - Maximize throughput from clients to tape (e.g. LHC experiments data taking)

- **C++ Implementation**
- **Reuse of CASTOR software infrastructure**
  - Derived SRM specific classes
- **Configurable number of thread pools for both front- and back-ends**
- **ORACLE centric**
- **Front and back ends can be distributed on multiple hosts**

dCache-SRM
FNAL and DESY

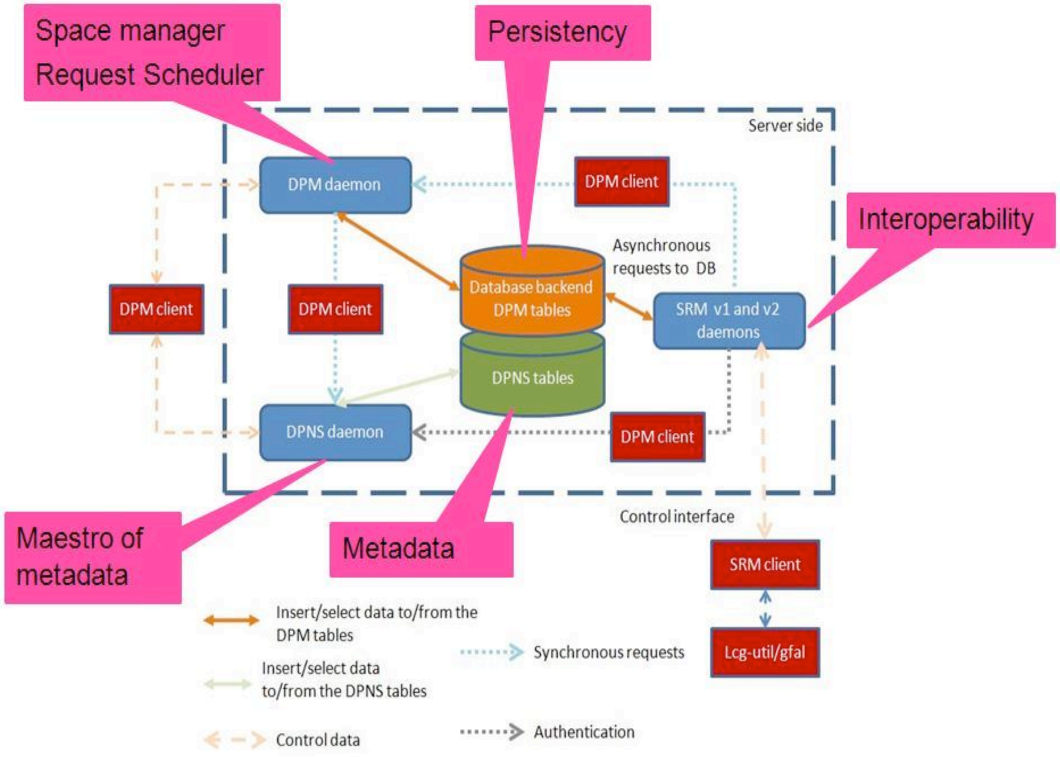
- Strict name space and data storage separation
- Automatic file replication based on access patterns
- HSM Connectivity (Enstore, OSM, TSM, HPSS, DMF)
- Automated HSM migration and restore
- Scales to Peta-byte range on 1000’s of disks
- Supported protocols:
  - (gsi/krb)FTP, (gsi/krb)dCap, xRoot, NFS 2/3
- Separate I/O queues per protocol
- Resilient dataset management
- Command line and graphical admin interface
- Variety of Authorization mechanisms including VOMS
- Deployed in a large number of institutions worldwide



- Support SRM 1.1 and SRM 2.2
- Dynamic Space Management
- Request queuing and scheduling
- Load balancing
- Robust replication using srmCopy functionality via SRM, (gsi)FTP and http protocols

Disk Pool Manager (DPM) CERN

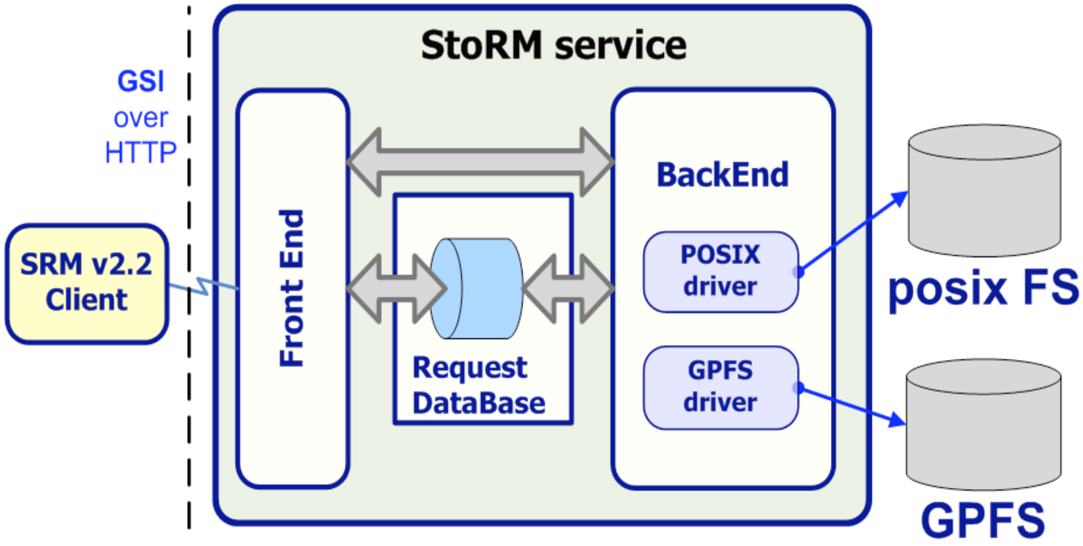
- **Provide a reliable, secure and robust storage system**
- **Manages storage on disks only**
- **Security**
  - GSI for authentication
  - VOMS for authorization
  - Standard POSIX permissions + ACLs based on user's DN and VOMS roles
- **Virtual ids**
  - Accounts created on the fly
- **Full SRMv2.2 implementation**
- **Standard disk pool manager capabilities**
  - Garbage collector
  - Replication of hot files
- **Transfer protocols**
  - GridFTP (v1 and v2)
  - Secure RFIO
  - https
  - Xroot
- **Works on Linux 32/64 bits machines**
- **Direct data transfer from/to disk server (no bottleneck)**
- **Support DICOM backend**
  - Requirement from Biomed VO
  - Storage of encrypted files in DPM on the fly + local decryption
  - Use of GFAL/srm to get TURLs and decrypt the file



- **Supported database backends**
  - MySQL
  - Oracle
- **High availability**
  - All servers can be load balanced (except the DPM one)
  - Resilient: all states are kept in the DB at all times

Storage Resource Manager (StoRM)
INFN/CNAF - ICTP/EGRID

- It's designed to leverage the advantages of high performing parallel file systems in Grid.
- Different file systems supported through a driver mechanism:
  - generic POSIX FS
  - GPFS
  - Lustre
  - XFS
- It provides the capability to perform local and secure access to storage resources (file:// access protocol + ACLs on data).

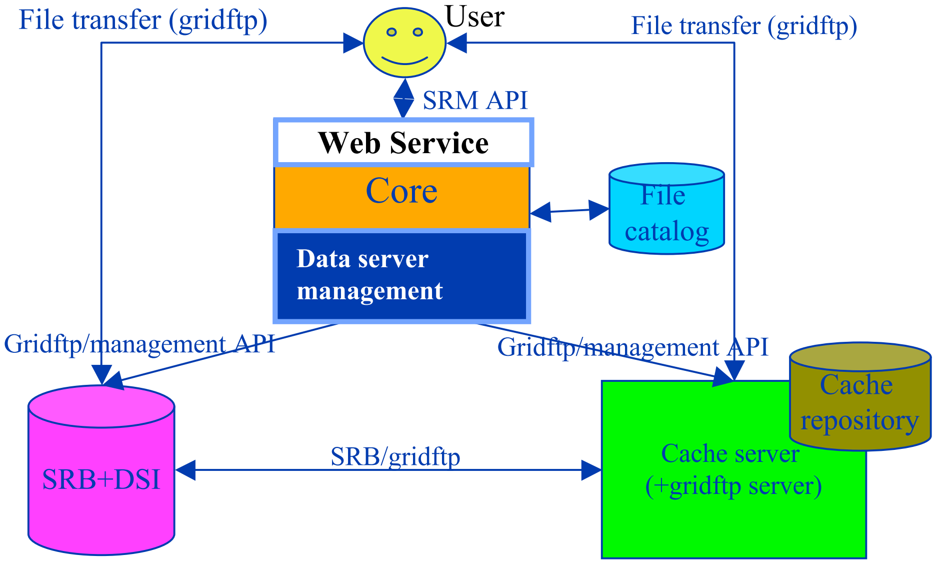


StoRM architecture:

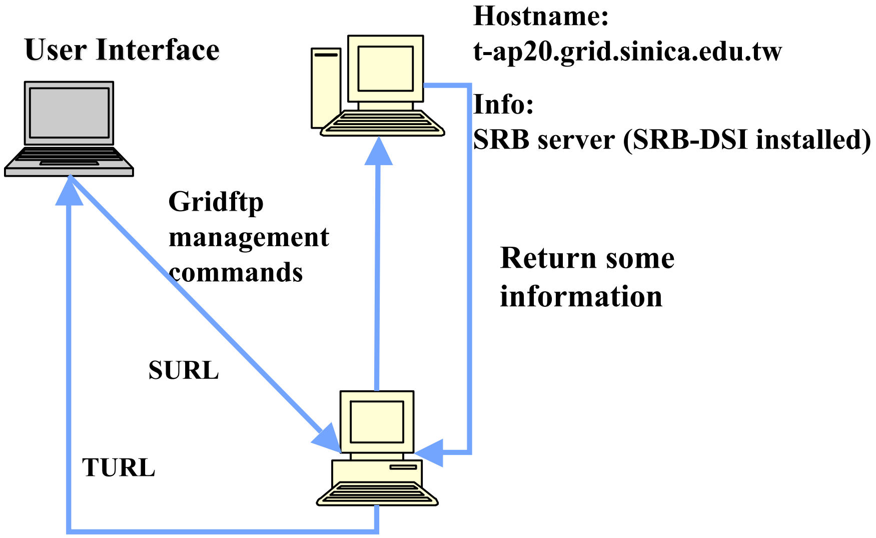
- Frontends: C/C++ based, expose the SRM interface
- Backends: Java based, execute SRM requests.
- DB: based on MySQL DBMS, stores requests data and StoRM metadata.
- Each component can be replicated and instantiated on a dedicated machine.

SRM on SRB
SINICA – TWGRID/EGEE

- SRM as a permanent archival storage system
- Finished the parts about authorizing users, web service interface and gridftp deployment, and SRB-DSI, and some functions like directory functions, permission functions, etc.
- Currently focusing on the implementation of core (data transfer functions and space management)
- Use LFC (with a simulated LFC host) to get SURL and use this SURL to connect to SRM server, then get TURL back



User
File transfer (gridftp)
File transfer (gridftp)
SRM API
Web Service
Core
Data server management
File catalog
Gridftp/management API
Gridftp/management API
SRB+DSI
SRB/gridftp
Cache repository
Cache server (+gridftp server)



User Interface

Hostname:
t-ap20.grid.sinica.edu.tw

Info:
SRB server (SRB-DSI installed)

Gridftp management commands

Return some information

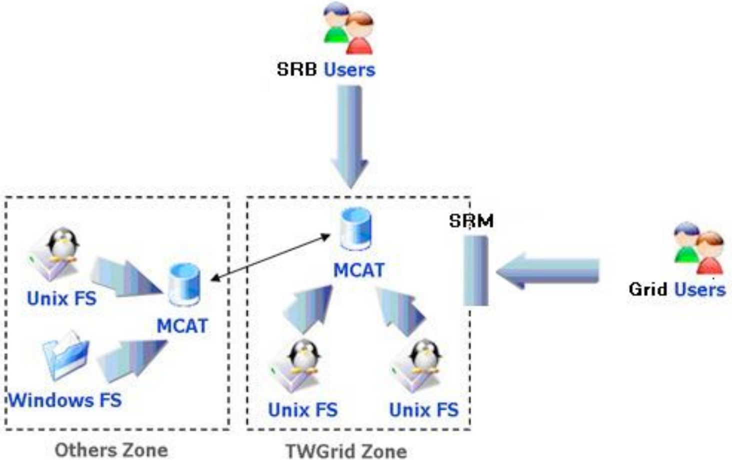
SURL

TURL

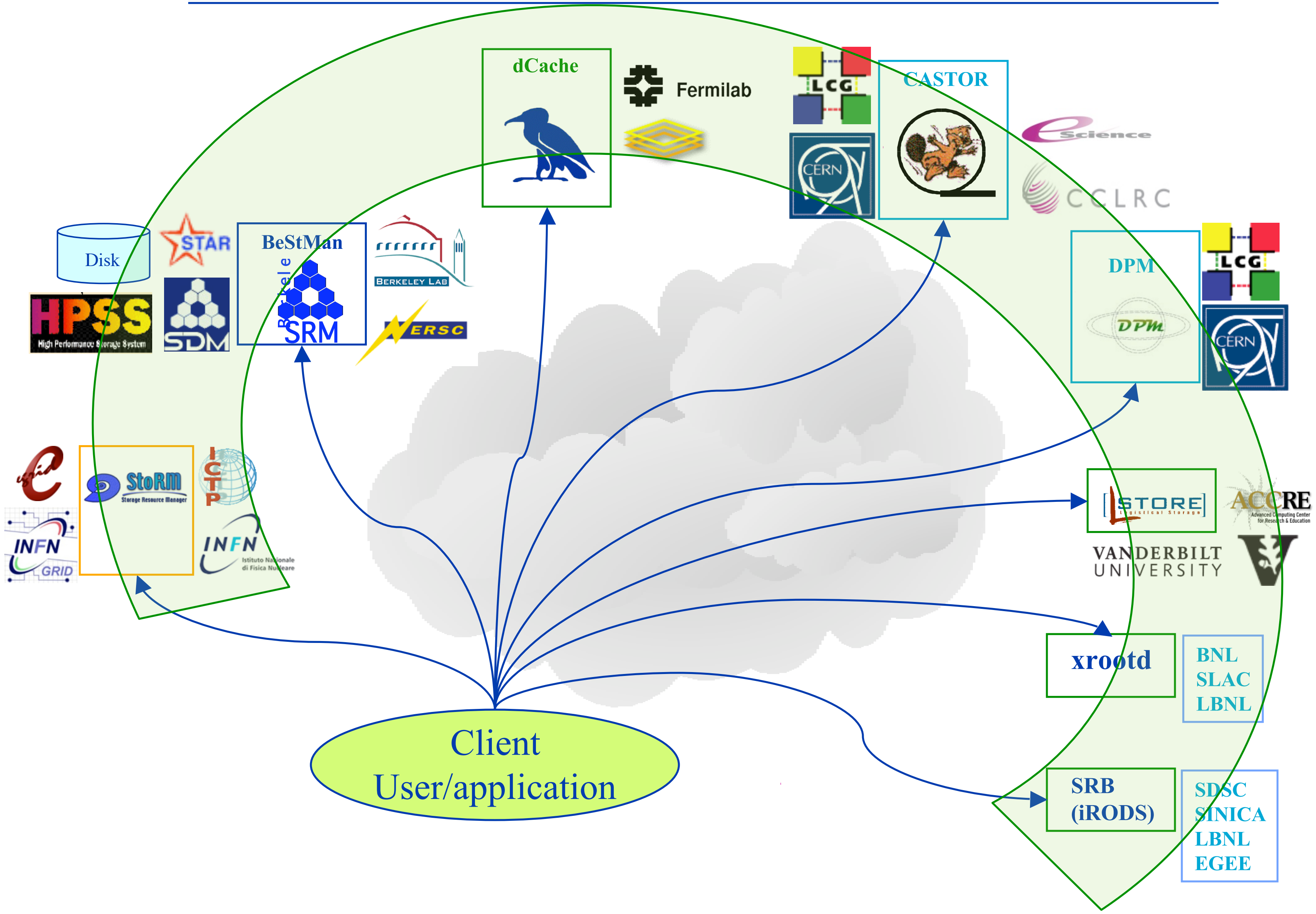
Hostname: fct01.grid.sinica.edu.tw

The end point: httpg://fct01.grid.sinica.edu.tw:8443/axis/services/srm

Info: Cache server (gridftp server) and SRM interface



Interoperability in SRM v2.2



Open Science Grid

SRMs at work

- Europe : LCG/EGEE
  - 191+ deployments, managing more than 10PB
    - 129 DPM/SRM
    - 54 dCache/SRM
    - 7 CASTOR/SRM at CERN, CNAF, PIC, RAL, SINICA
    - StoRM at ICTP/EGRID, INFN/CNAF
  - SRM layer for SRB, SINICA
- US
  - Estimated at about 30 deployments
  - OSG
    - BeStMan/SRM from LBNL
    - dCache/SRM from FNAL
  - ESG
    - DRM/SRM, HRM/SRM at LANL, LBNL, LLNL, NCAR, ORNL
  - Others
    - BeStMan/SRM adaptation on Lustre file system at Texas Tech
    - BeStMan-Xrootd adaptation at SLAC
    - JasMINE/SRM from TJNAF
    - L-Store/SRM from Vanderbilt Univ.

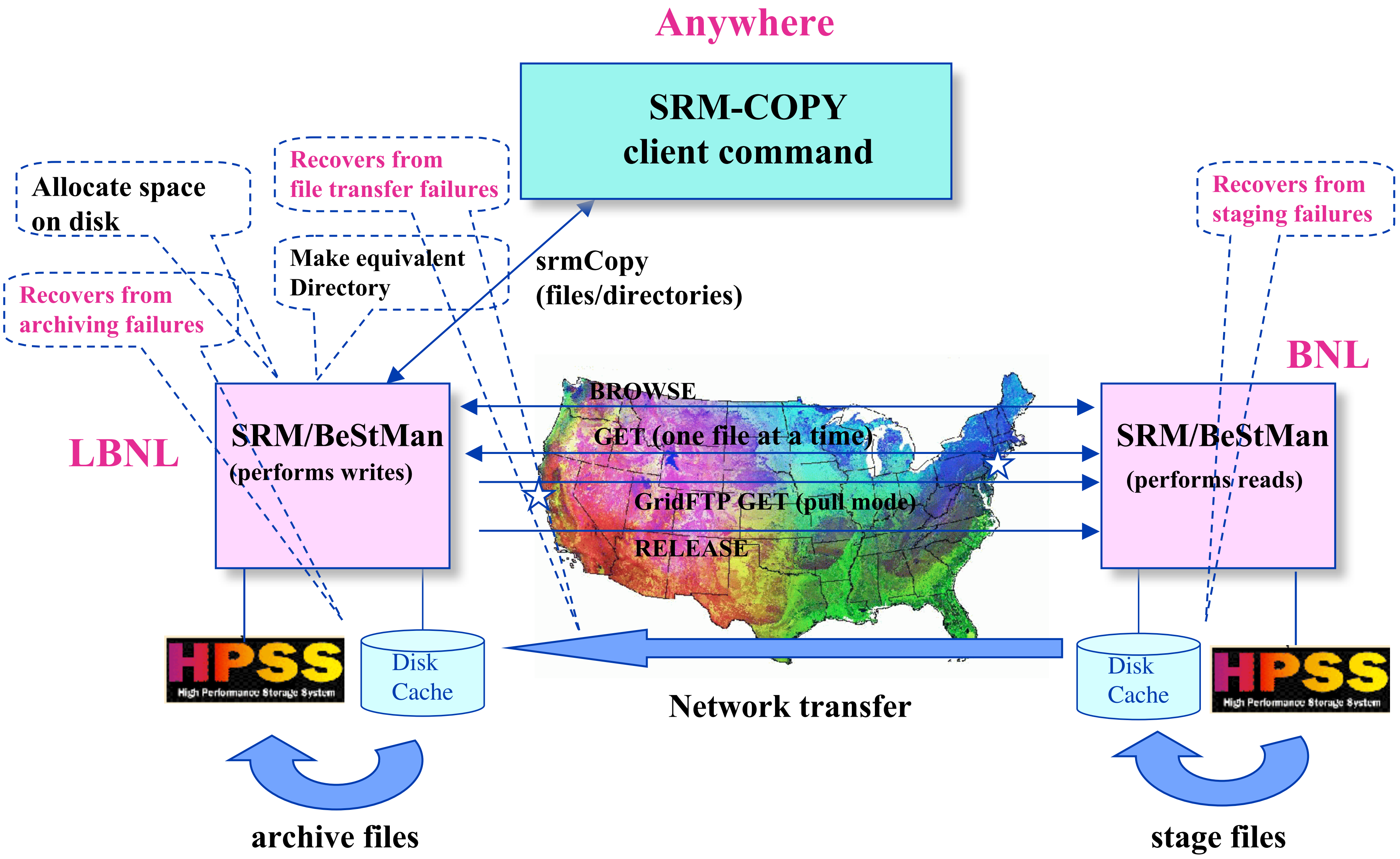
Examples of SRM usage in real production Grid projects

Open Science Grid

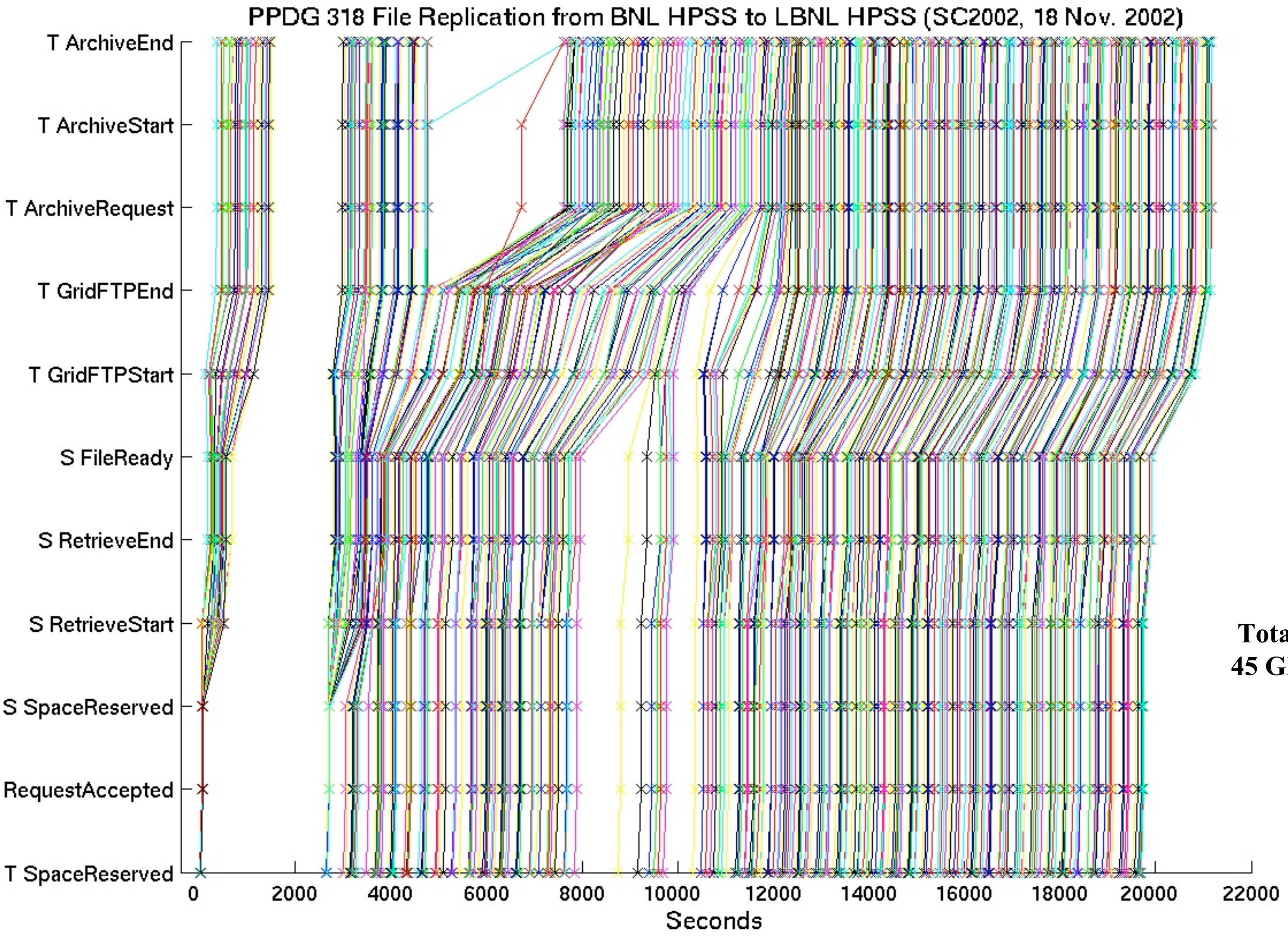
HENP STAR experiment

- Data Replication from BNL to LBNL
  - 1TB/10K files per week on average
  - In production for over 4 years
- Event processing in Grid Collector
  - Prototype uses SRMs and FastBit indexing embedded in STAR framework
- STAR analysis framework
  - Job driven data movement
    1. Use BeStMan/SRM to bring files into local disk from a remote file repository
    2. Execute jobs that access “staged in” files in local disk
    3. Job creates an output file on local disk
    4. Job uses BeStMan/SRM to moves the output file from local storage to remote archival location
    5. SRM cleans up local disk when transfer complete
    6. Can use any other SRMs implementing v2.2

Data Replication in STAR

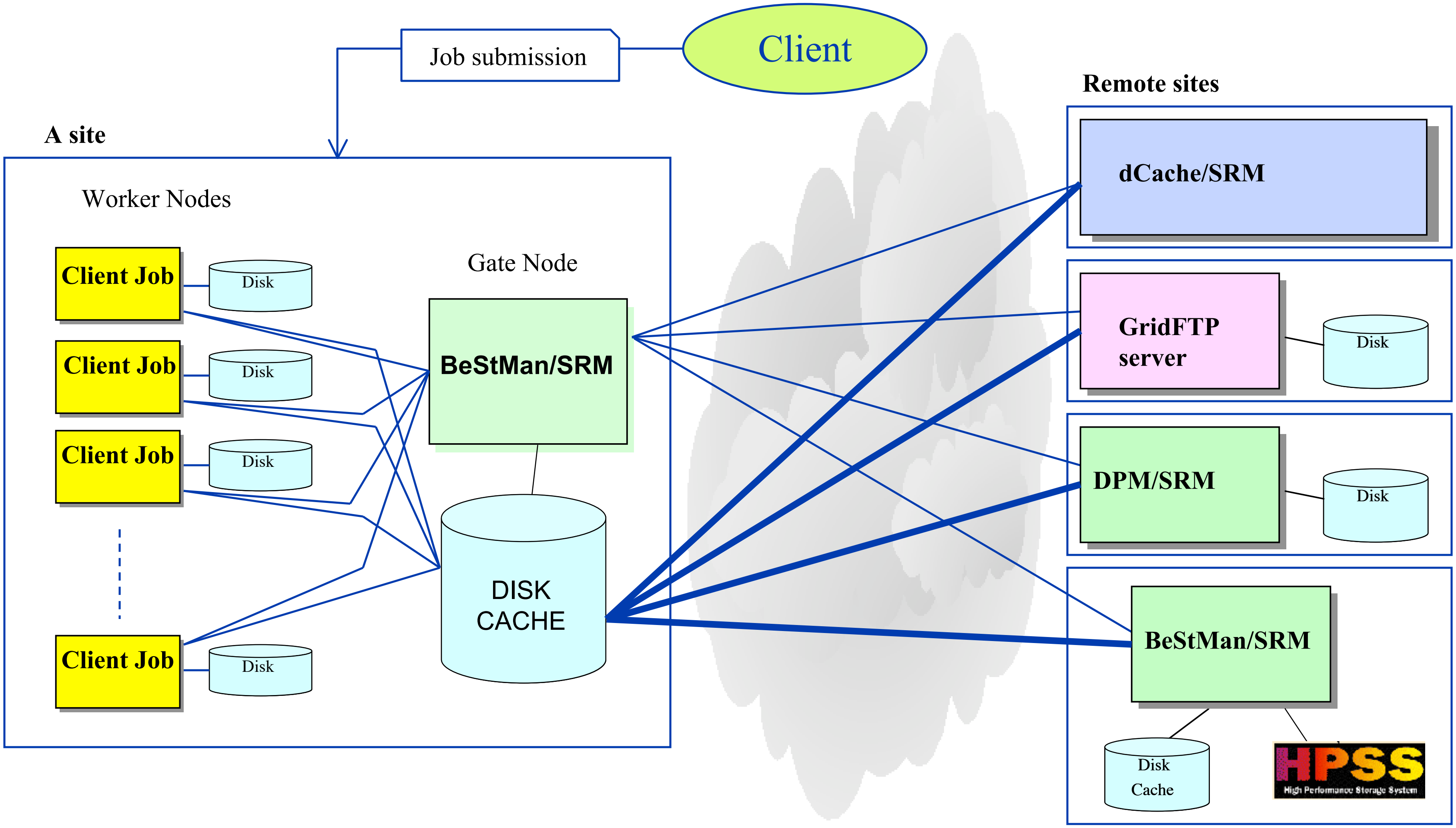


File Tracking Shows Recovery From Transient Failures



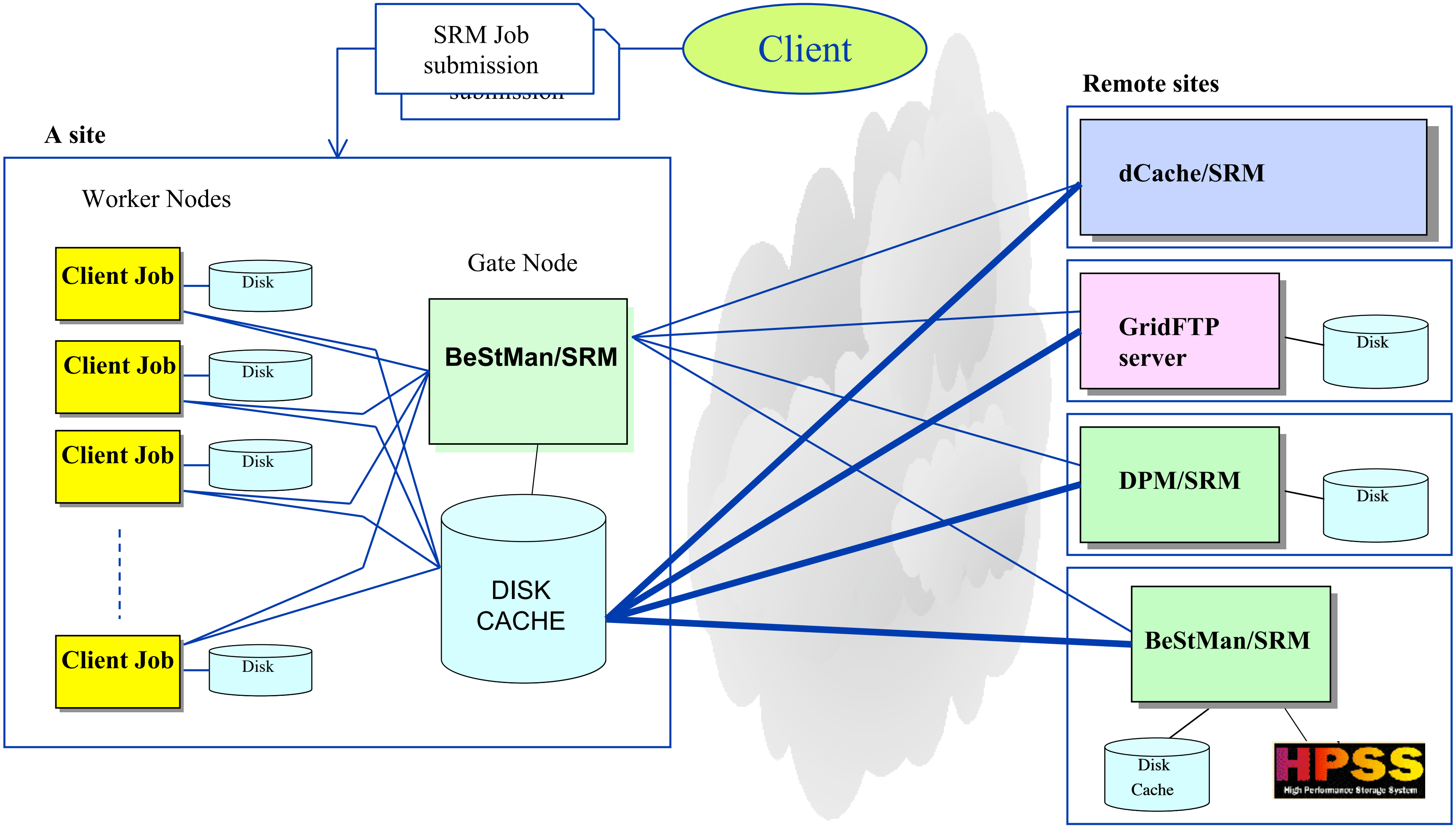
Total: 45 GBs

STAR Analysis scenario (1)



Open Science Grid

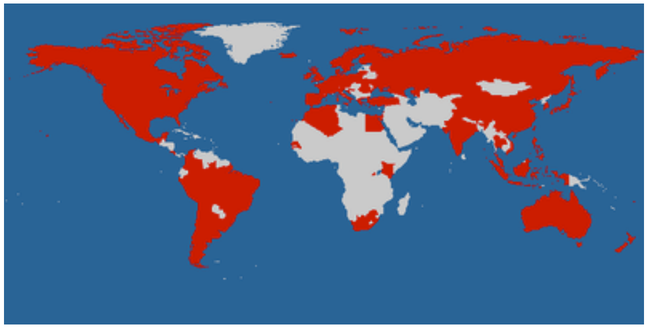
STAR Analysis scenario (2)



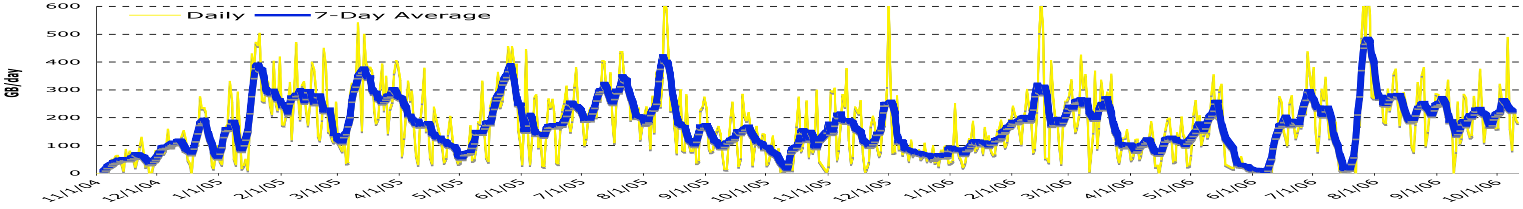
Open Science Grid

Earth System Grid

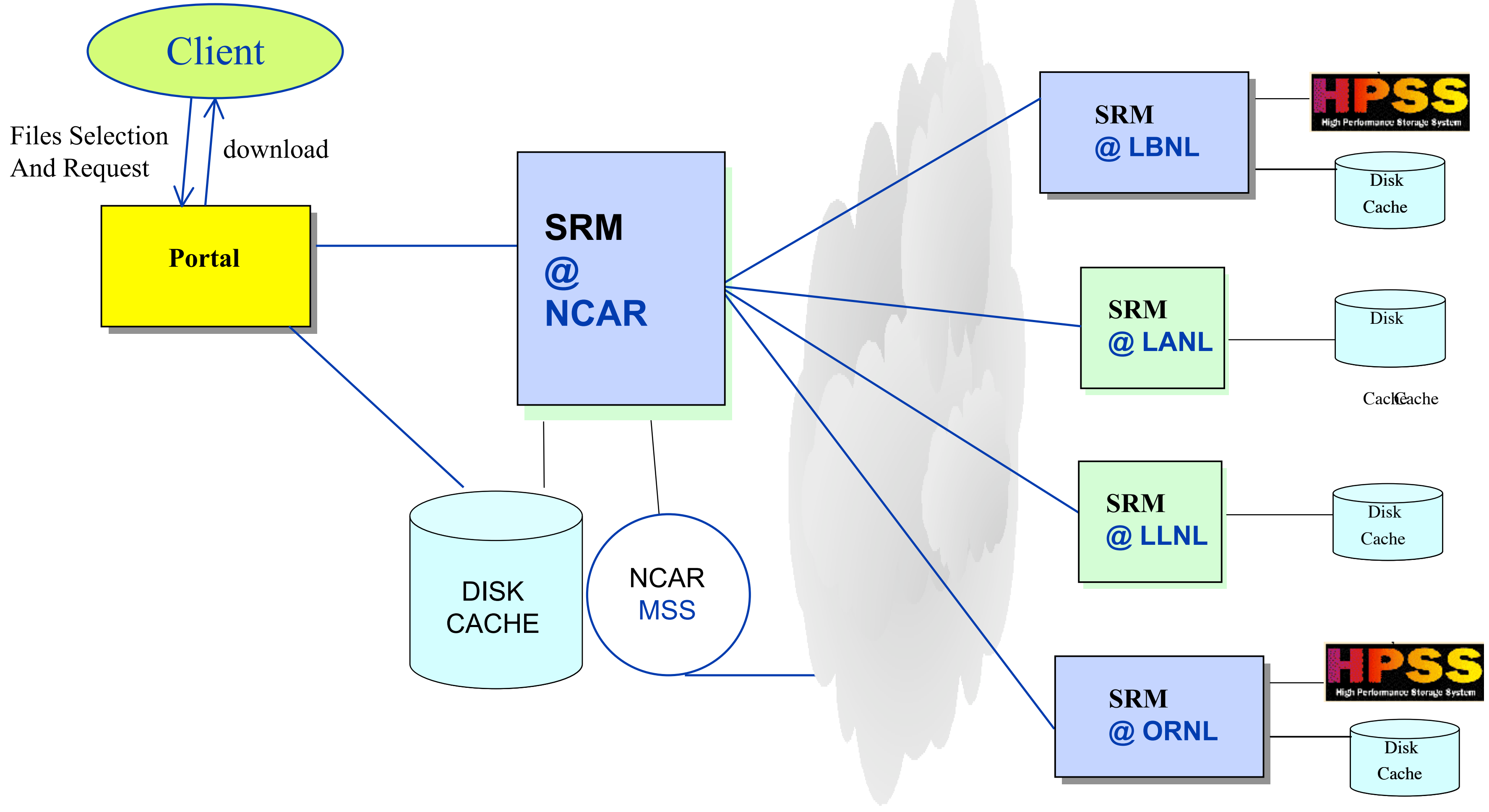
- **Main ESG portal**
  - **148.53 TB of data at four locations (NCAR, LBNL, ORNL, LANL)**
    - **965,551 files**
    - **Includes the past 7 years of joint DOE/NSF climate modeling experiments**
  - **4713 registered users from 28 countries**
    - **Downloads to date: 31TB/99,938 files**
- **IPCC AR4 ESG portal**
  - **28 TB of data at one location**
    - **68,400 files**
    - **Model data from 11 countries**
    - **Generated by a modeling campaign coordinated by the Intergovernmental Panel on Climate Change (IPCC)**
  - **818 registered analysis projects from 58 countries**
    - **Downloads to date: 123TB/543,500 files, 300 GB/day on average**



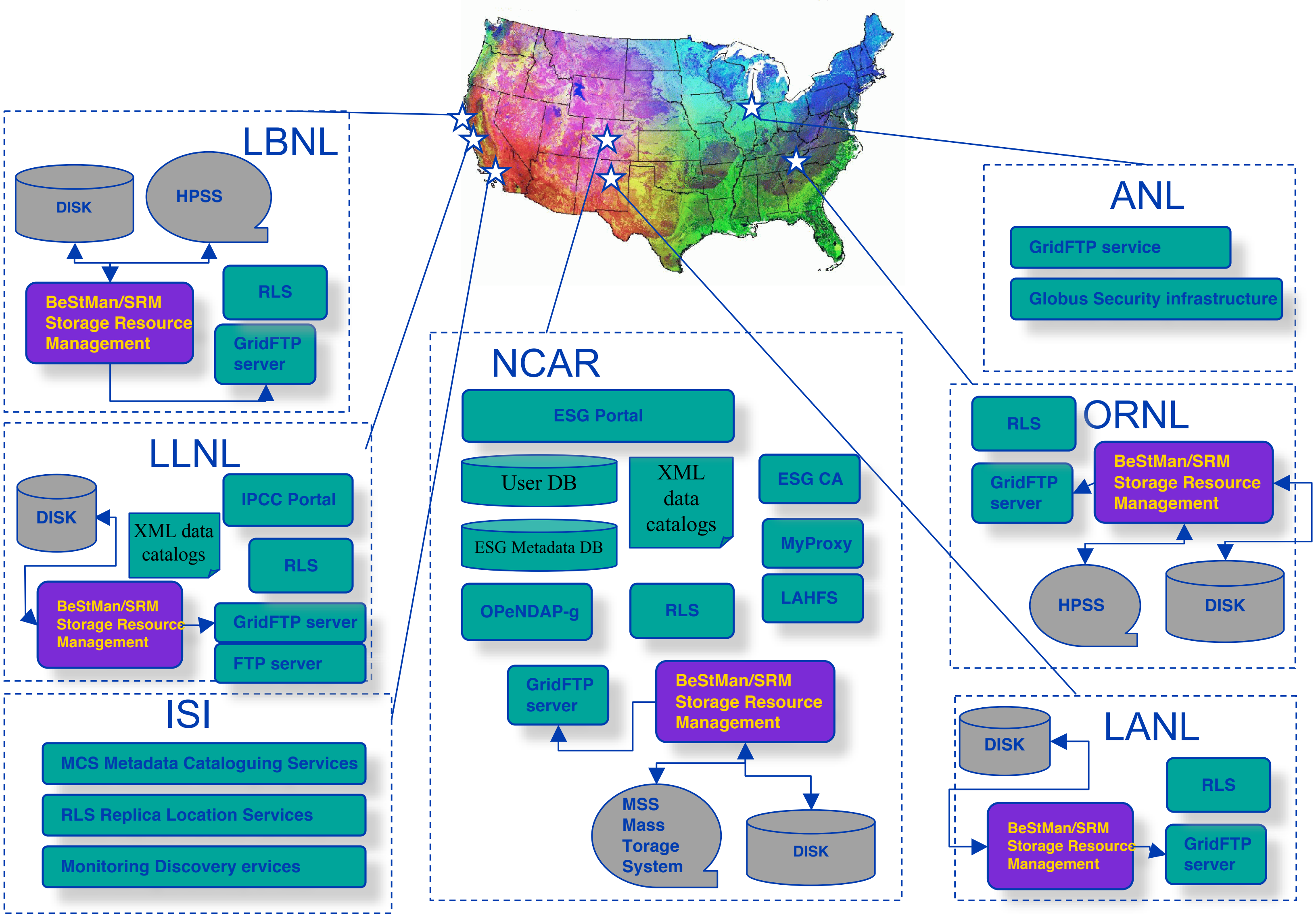
Courtesy: http://www.earthsystemgrid.org



SRMs in ESG


Client
Files Selection
And Request
download
Portal
SRM
@
NCAR
DISK
CACHE
NCAR
MSS
SRM
@ LBNL
HPSS
High Performance Storage System
Disk
Cache
SRM
@ LANL
Disk
CacheCache
SRM
@ LLNL
Disk
Cache
SRM
@ ORNL
HPSS
High Performance Storage System
Disk
Cache

SRM works in concert with other Grid components in ESG



LBNL

DISK

HPSS

BeStMan/SRM Storage Resource Management

RLS

GridFTP server

LLNL

DISK

IPCC Portal

XML data catalogs

RLS

BeStMan/SRM Storage Resource Management

GridFTP server

FTP server

ISI

MCS Metadata Cataloguing Services

RLS Replica Location Services

Monitoring Discovery ervices

NCAR

ESG Portal

User DB

XML data catalogs

ESG CA

ESG Metadata DB

MyProxy

OPeNDAP-g

RLS

LAHFS

GridFTP server

BeStMan/SRM Storage Resource Management

MSS Mass Torage System

DISK

ANL

GridFTP service

Globus Security infrastructure

ORNL

RLS

GridFTP server

BeStMan/SRM Storage Resource Management

HPSS

DISK

LANL

DISK

RLS

BeStMan/SRM Storage Resource Management

GridFTP server

Summary

Summary and Current Status

- Storage Resource Management – essential for Grid
- Multiple implementations interoperate
  - Permits special purpose implementations for unique storage
  - Permits interchanging one SRM implementation by another
- Multiple SRM implementations exist and are in production use
  - Particle Physics Data Grids
    - WLCG, EGEE, OSG, …
  - Earth System Grid
  - More coming …
    - Combustion, Fusion applications
    - Medicine

Open Science Grid

Documents and Support

- SRM Collaboration and SRM Specifications
  - http://sdm.lbl.gov/srm-wg
  - OGF mailing list : gsm-wg@ogf.org
  - SRM developer's mailing list: srm-devel@fnal.gov

- BeStMan (Berkeley Storage Manager): http://datagrid.lbl.gov/bestman
- CASTOR (CERN Advanced STORage manager): http://www.cern.ch/castor
- dCache: http://www.dcache.org
- DPM (Disk Pool Manager): https://twiki.cern.ch/twiki/bin/view/LCG/DpmInformation
- StoRM (Storage Resource Manager): http://storm.forge.cnaf.infn.it
- SRM-SRB: http://lists.grid.sinica.edu.tw/apwiki/SRM-SRB
- SRB: http://www.sdsc.edu/srb
- BeStMan-XrootD: http://wt2.slac.stanford.edu/xrootdfs/bestman-xrootd.html

- Other support info : srm@lbl.gov

Credits

Alex Sim <asim@lbl.gov>

Arie Shoshani <ashoshani@lbl.gov>