Uniform Grid Storage Access

Scientific Data management Research Group
Computational Research Division
Lawrence Berkeley National Laboratory

Contact: Alex Sim <ASim@lbl.gov>

Super Computing 2008, Nov. 17-20, 2008
Austin, TX, USA

Abstract

Large scale Grid computing requires dynamic storage allocation and management of large number of files. However, storage systems vary from a single disk to complex mass storage systems. A standard middleware specification, called Storage Resource Management (SRM) has been developed over the last seven years. It provides the functionality for dynamic storage reservation and management of the files in Grid spaces and manages file movement between these spaces. This demo will show the interoperability of different SRM implementations around the world based on the latest SRM specification. It will show the ability to put, get, and copy files between any of these storages systems using the SRM interfaces. In particular, we will demonstrate the ability of analysis program getting and putting files into a variety of remote storage systems using uniform SRM calls. Such analysis programs only need the SRM client to interact with any SRM-based or GridFTP-based servers. Many of these SRM-frontend systems are now used in large Grid projects, including the High Energy Physics Worldwide LHC Computing Grid (WLCG) Project, Open Science Grid (OSG) and the Earth System Grid (ESG) project.

What is SRM?

- **Storage Resource Managers (SRMs) are middleware components**
  - whose function is to provide dynamic space allocation and file management on shared storage components on the Grid
  - Different implementations for underlying storage systems based on the SRM specification
- **SRMs in the data grid**
  - 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

Motivation & Requirements (1)

- Grid architecture needs to include reservation & scheduling of:
  - Compute resources
  - Storage resources
  - Network resources
- Storage Resource Managers (SRMs) role in the data grid architecture
  - Shared storage resource allocation & scheduling
  - Specially important for data intensive applications
  - Often files are archived on a mass storage system (MSS)
  - Wide area networks – need to minimize transfers by file sharing
  - Scaling: large collaborations (100's of nodes, 1000's of clients) – opportunities for file sharing
  - File replication and caching may be used
  - Need to support non-blocking (asynchronous) requests

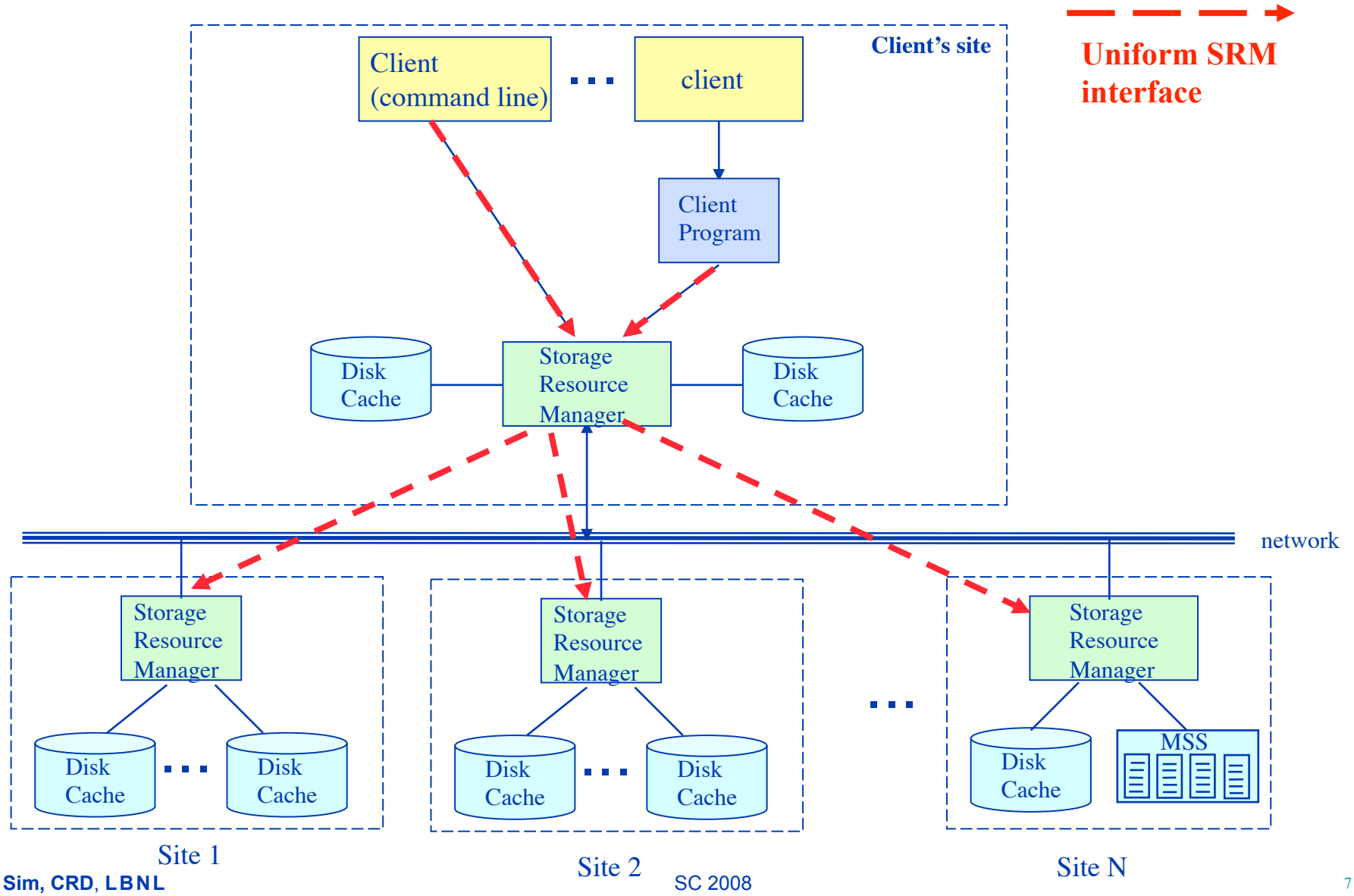
Motivation & Requirements (2)

- 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
- 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 do that 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
- Now, suppose you want to run distributed jobs on the Grid
  - Need to allocate remote spaces
  - Need to move (stream) files to remote sites
  - Need to manage file outputs and their movement to destination site(s)

Client and Peer-to-Peer Uniform Interface



Storage Resource Managers: Main concepts

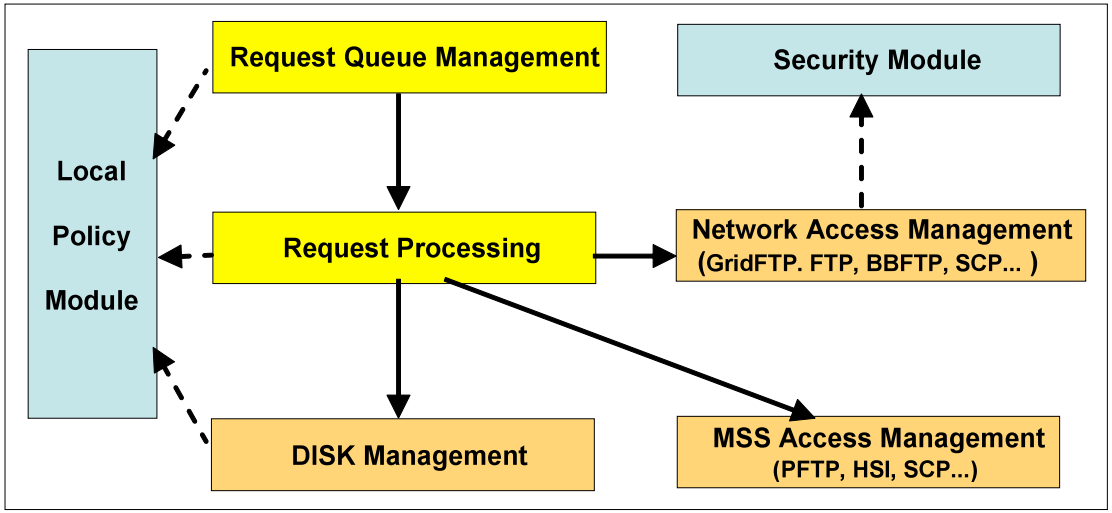
- Non-interference with local policies
- Advance space reservations
- Dynamic space management
- Pinning file in spaces
- Support abstract concept of a file name: Site URL
- Temporary assignment of file names for transfer: Transfer URL
- Directory Management and ACLs
- Transfer protocol negotiation
-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

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.**

Berkeley Storage Manager (BeStMan) LBNL

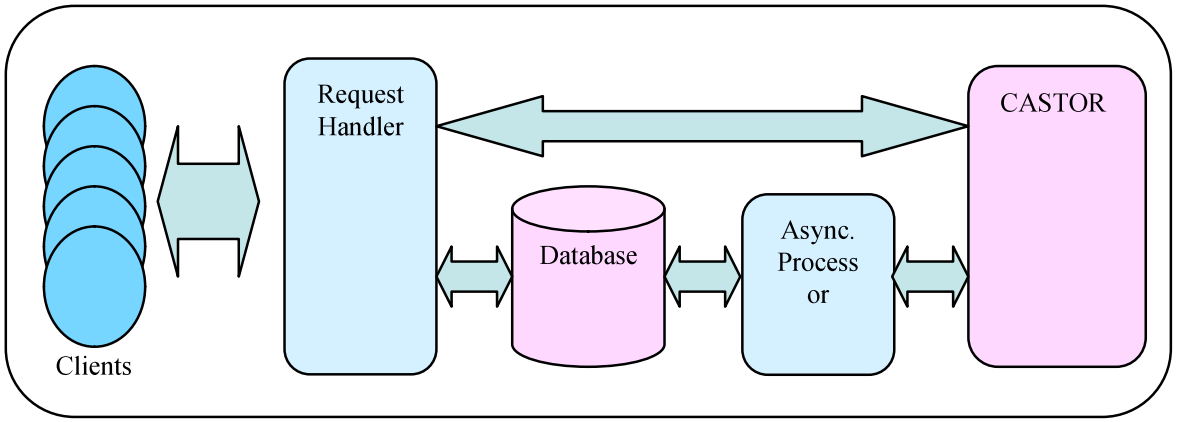
- Java implementation
- Designed to work with unix-based disk systems
- As well as MSS to stage/ archive from/to its own disk (currently HPSS)
- Adaptable to other file systems and storages (e.g. NCAR MSS, Hadoop, Lustre, Xrootd)
- Uses in-memory database (BerkeleyDB)



- Multiple transfer protocols
- Space reservation
- Directory management (no ACLs)
- Can copy files from/to remote SRMs
- Can copy entire directory robustly
  - 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



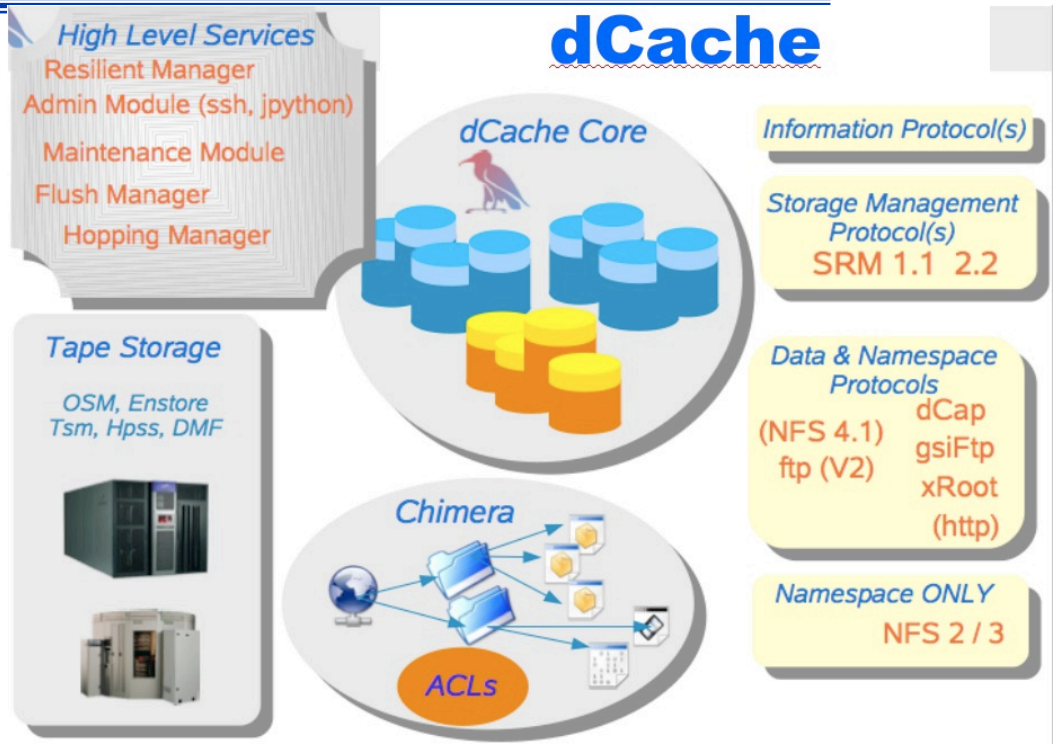
- **CASTOR is the HSM in production at CERN**
  - 21 PB on tape, 5 PB on disk, 100M+ files
- **Support for any TapeN-DiskM storage class**
- **Designed to meet Large Hadron Collider Computing requirements**
  - Maximize throughput from clients to tape (e.g. LHC experiments data taking)
- **Also deployed at ASGC, CNAF, RAL**

- **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**

Slide courtesy: Jan van Eldik
Giuseppe Lo Presti
Shaun De Witt

dCache-SRM
FNAL, DESY, NDGF

- Strict name space and data storage separation
- Automatic file replication on 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 IO 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

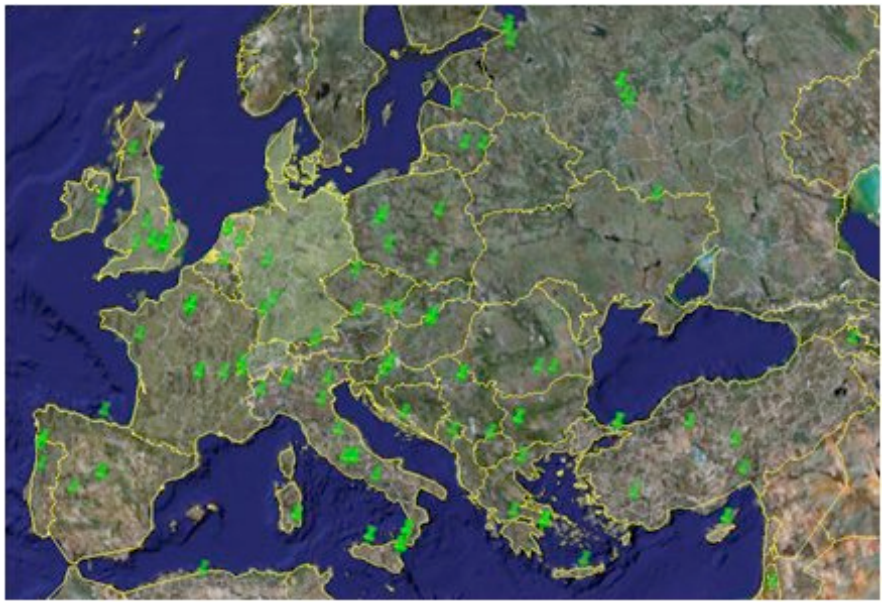


- SRM 1.1 and SRM 2.2
- Dynamic Space Management
- Request queuing and scheduling
- Load balancing
- Robust replication using SrmCopy functionality via SRM, (gsi)FTP and http protocols

Slide courtesy: Timur Perelmutov

Disk Pool Manager (DPM)
CERN

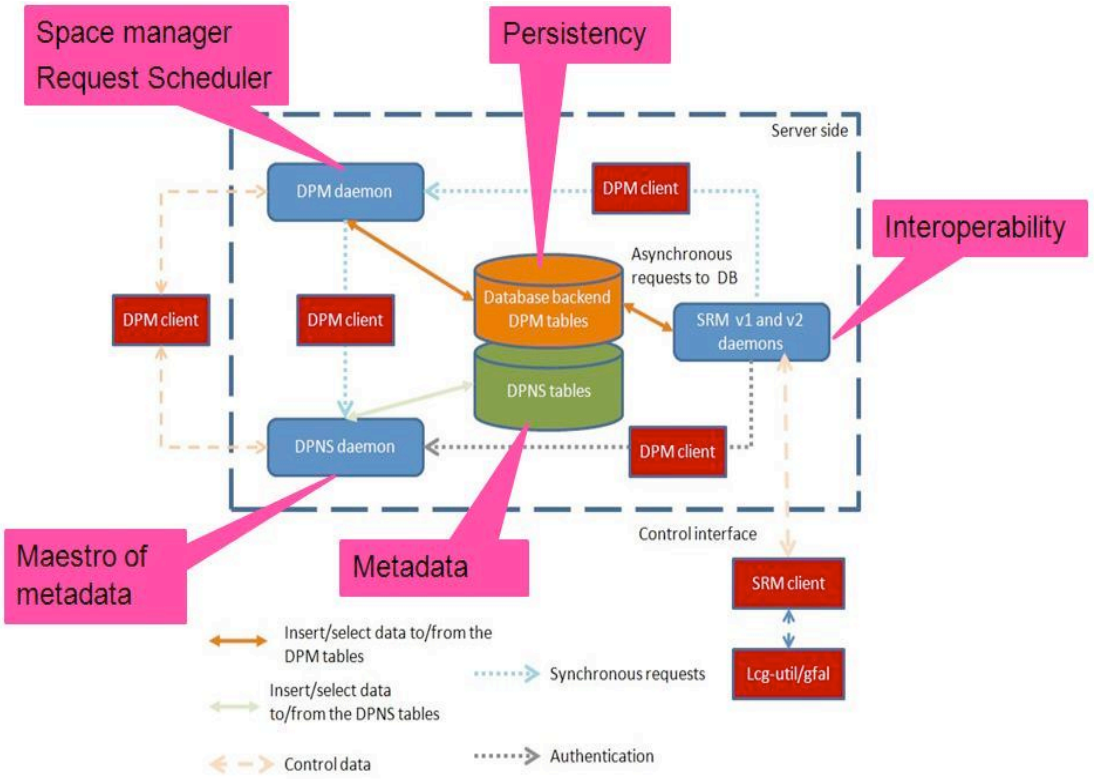
- **Objective**
  - Provide a reliable, secure and robust storage system
- **Requirements**
  - Store a few hundreds of TB
  - Easy to install and to manage
  - Scalable storage system
  - Interoperable with other SEs
- **Production**
  - 187 DPM installed
  - 224 supported VOs
  - Biggest DPM : 200 TB
  - Expanding to 400 TB



Slide courtesy: Maarten Litmaath
Flavia Donno
Akos Frohner
David Smith
Jean-Philippe Baud

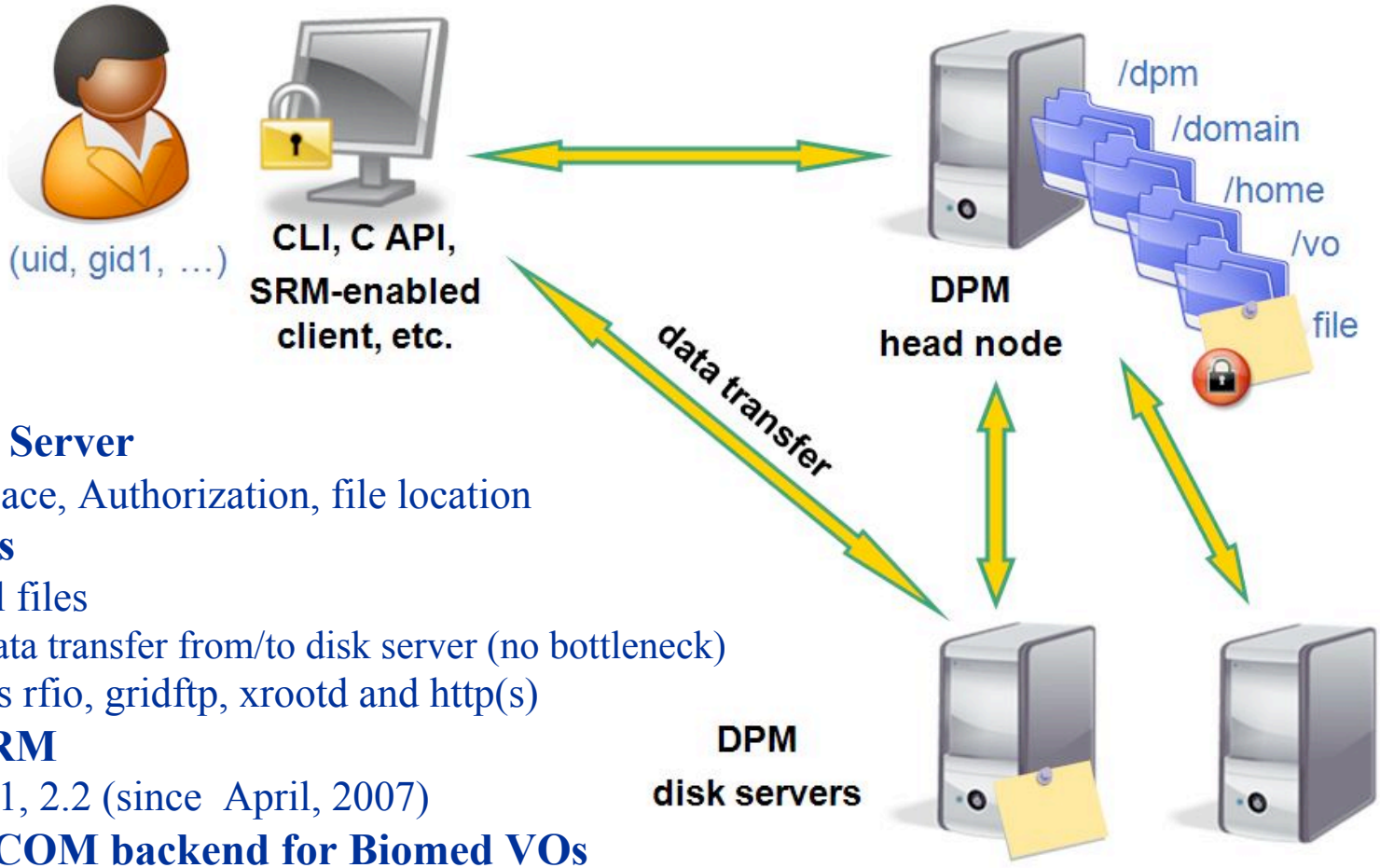
DPM: technical aspects

- **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, MacOSX and OpenSolaris 10**



- **Supported database backends**
  - MySQL, Postgres, Oracle
- **Support for IPv6**
- **High availability**
  - All services except DPM daemon can be load balanced
  - Resilient: all states are kept in the DB at all times

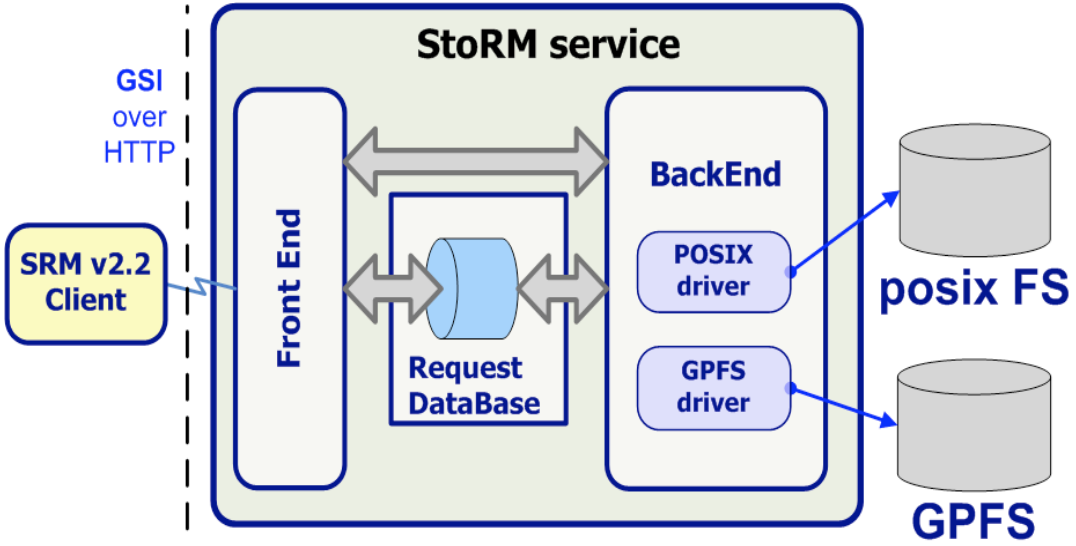
DPM: user's point of view



- **DPM Name Server**
  - ✓ Namespace, Authorization, file location
- **Disk Servers**
  - ✓ Physical files
  - ✓ Direct data transfer from/to disk server (no bottleneck)
  - ✓ Supports rfio, gridftp, xrootd and http(s)
- **Supports SRM**
  - ✓ SRM-1.1, 2.2 (since April, 2007)
- **Support DICOM backend for Biomed VOs**
  - ✓ Encryption of DICOM files on the fly + local decryption
  - ✓ Use of GFAL and Hydra to get and decrypt the file

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).



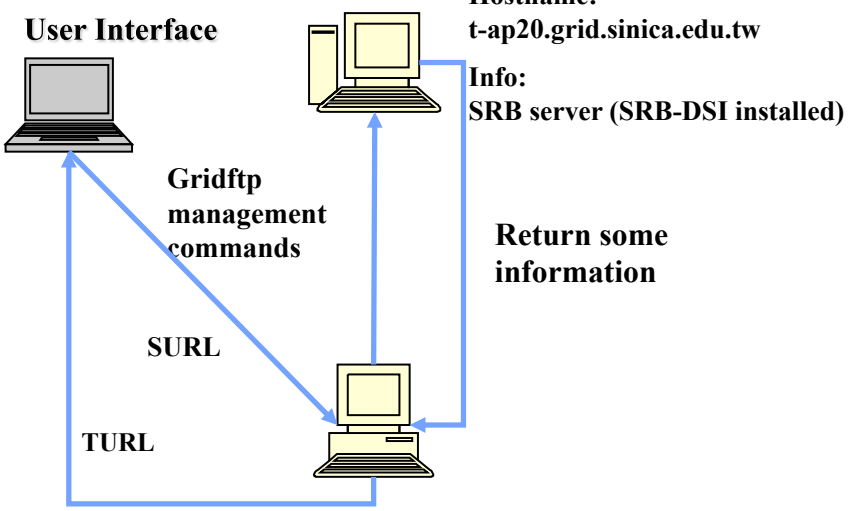
StoRM architecture:

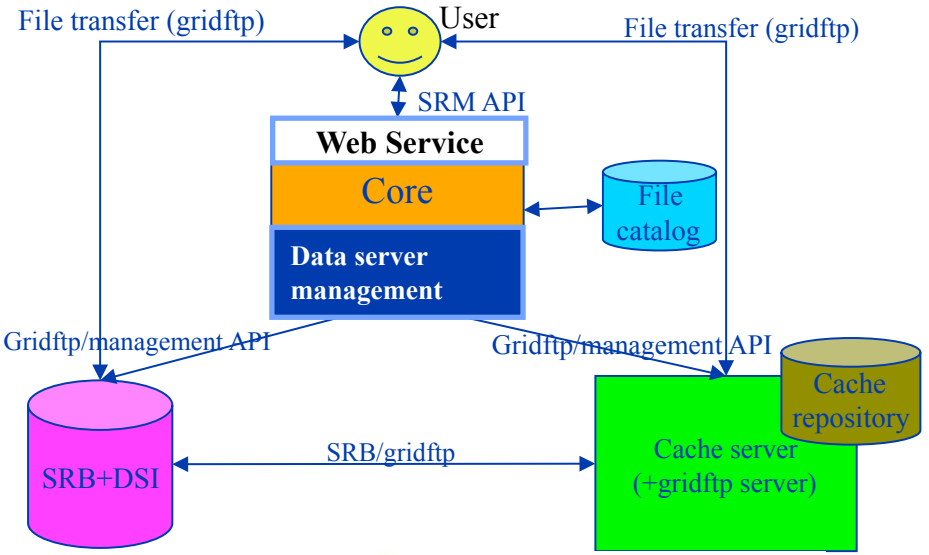
- Frontends: C/C++ based, expose the SRM interface
- Backends: Java based, execute SRM requests.
- DB: based on MySQL DBMS, stores requests data and StoRM metadata.
- Each component can be replicated and instantiated on a dedicated machine.

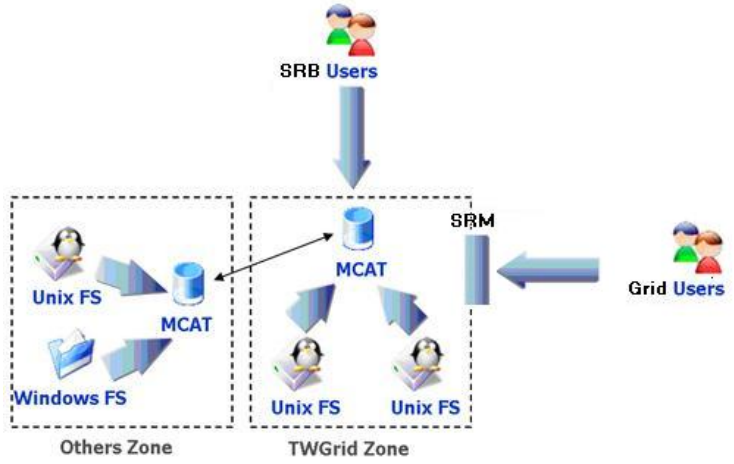
Slide courtesy: Luca Magnoni

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

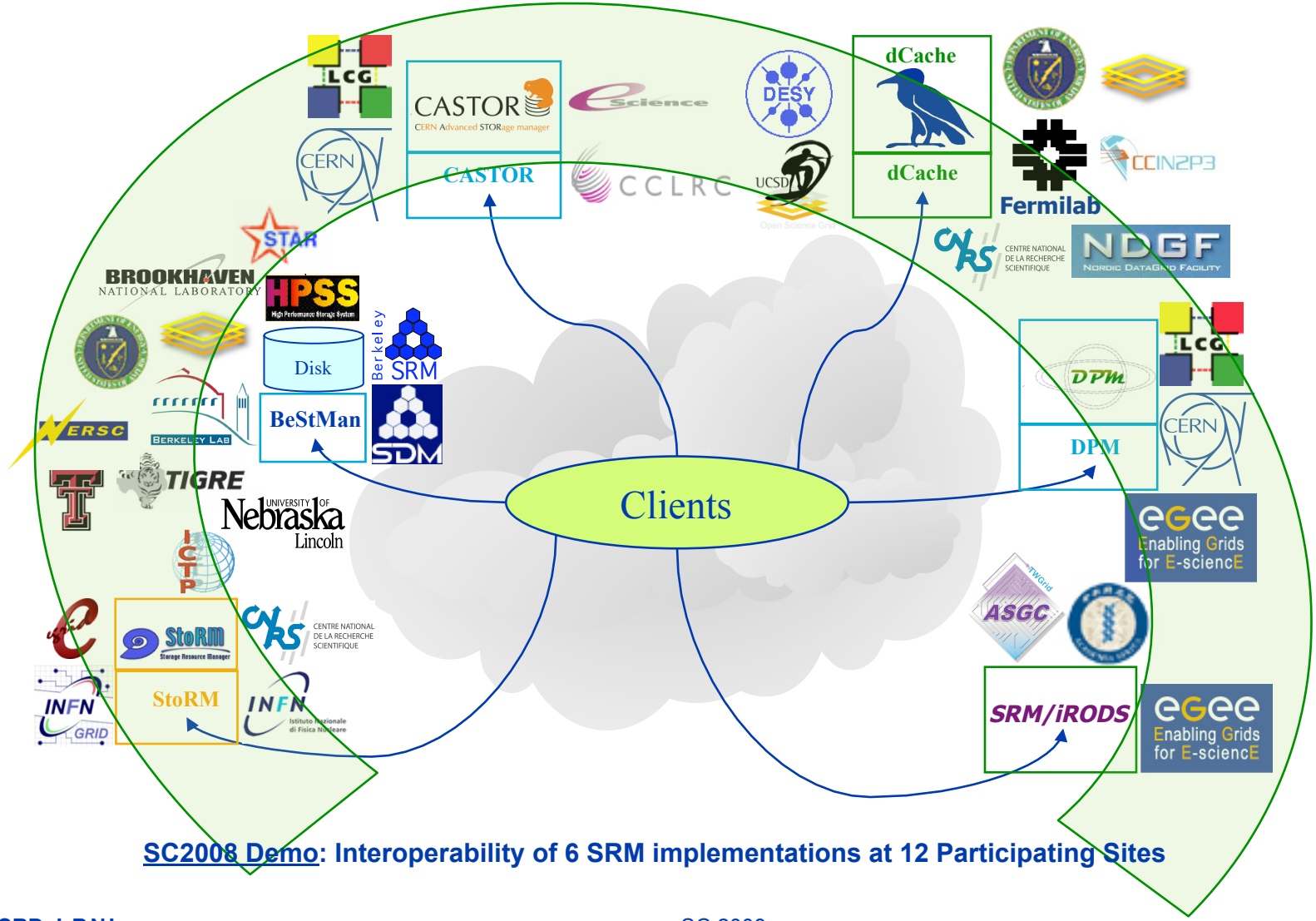






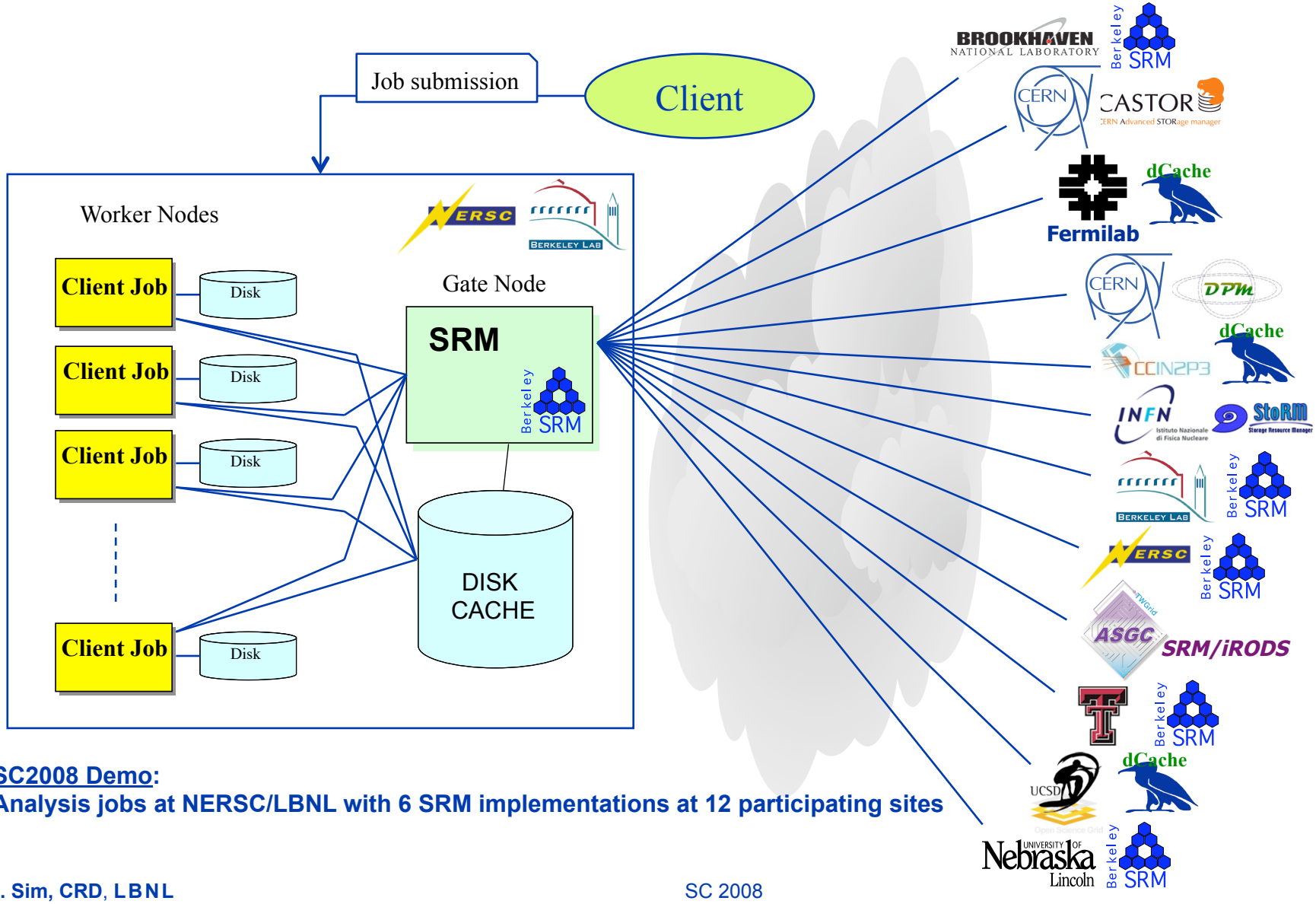
Slide courtesy: Fu-Ming Tsai
Wei-Lung Ueng

Interoperability in SRM



SC2008 Demo: Interoperability of 6 SRM implementations at 12 Participating Sites

SRMs Facilitate Analysis Jobs



SC2008 Demo:
Analysis jobs at NERSC/LBNL with 6 SRM implementations at 12 participating sites

SRMs at work

- Europe/Asia/Canada/South America/Australia/Afraca : LCG/EGEE
  - 250+ deployments, managing more than 10PB (as of 11/11/2008)
    - 172 DPM
    - 57 dCache at 45 sites
    - 6 CASTOR at CERN, CNAF, RAL, SINICA, CIEMAT (Madrid), IFIC (Valencia)
    - 22 StoRM (17 Italy, 1 Greece, 1 UK, 1 Portugal, 2 Spain)
  - SRM layer for SRB, SINICA
- US
  - Estimated at about 50 deployments (as of 11/11/2008)
  - OSG
    - dCache from FNAL
    - BeStMan from LBNL
  - ESG
    - BeStMan at LANL, LBNL, LLNL, NCAR, ORNL
  - Others
    - JasMINE from TJNAF
    - BeStMan adaptation on Lustre file system at Texas Tech Univ.
    - BeStMan adaptation on Hadoop file system at Univ. of Nebraska

Acknowledgements : SC08 demo contributors

In alphabetical order

- **BeStMan**
  - BNL/STAR : Jerome Lauret, Wayne Betts
  - LBNL : Vijaya Natarajan, Junmin Gu, Arie Shoshani, Alex Sim
  - NERSC : Shreyas Cholia, Eric Hjort, Doug Olson, Jeff Porter, Andrew Rose, Iwona Sakrejda, Jay Srinivasan
  - TTU : Alan Sill
  - UNL : Brian Bockelman, Research Computing Facility at UNL
- **CASTOR**
  - CERN : Olof Barring, Miguel Coelho, Flavia Donno, Jan van Eldik, Akos Frohner, Rosa Maria Garcia Rioja, Giuseppe Lo Presti, Gavin McCance, Steve Murray, Sebastien Ponce, Ignacio Reguero, Giulia Taurelli, Dennis Waldron
  - RAL : Shaun De Witt
- **dCache**
  - CERN : Flavia Donno
  - DESY : Bjoern Boettscher, Patrick Fuhrmann, Iryna Koslova, David Melkumyan, Paul Millar, Tigran Mkrtchyan, Martin Radicke, Owen Synge, German HGF Support Team, Open Science Grid
  - FNAL : Andrew Baranovski, Matt Crawford, Ted Hesselroth, Alex Kulyavtsev, Tanya Levshina, Dmitry Litvintsev, Alexander Moibenko, Gene Oleynik, Timur Perelmutov, Vladimir Podstavkov, Neha Sharma
  - gridPP : Greig Cowan
  - IN2P3 : Jonathan Schaeffer, Lionel Schwarz
  - Quattor : Stijn De Weirdt
  - NDGF : Gerd Behrmann
  - UCSD : James Letts, Terrence Martin, Abhishek Singh Rana, Frank Wuerthwein
- **DPM**
  - CERN : Lana Abadie, Jean-Philippe Baud, Akos Frohner, Sophie Lemaitre , Maarten Litmaath , Remi Mollon, David Smith
  - LAL-Orsay : Gilbert Grosdidier
- **StoRM**
  - CNAF : Luca Dell'Agnello, Luca Magnoni, Elisabetta Ronchieri , Riccardo Zappi
  - LHCb : Vincenzo Vagnoni
- **SRM/iRODS-SRB**
  - SINICA : HsinYen Chen, Ethan Lin, Simon Lin, FuMing Tsai, WeiLong Ueng, Eric Yen
- **SRB/GridFTP**
  - RAL : Jens Jensen (RAL), Matt Hodges (RAL), Derek Ross (RAL)
  - Daresbury lab : Roger Downing

Summary and Current Status

- Storage Resource Management – essential for Grid
  - OGF Standard GFD.129 (Apr. 15, 2008)
- Multiple implementations interoperate
  - Permit special purpose implementations for unique products
  - Permits interchanging one SRM product 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
    - Structural biology, medicine

Documents and Support

- SRM Collaboration and SRM Specifications
  - http://sdm.lbl.gov/srm-wg
  - 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

- Other info : srm@lbl.gov