

Grid Storage Management Working Group

Session 1

The Basic Storage Resource Management Functional Interface

Arie Shoshani
(co-authors: srm-wg)

Lawrence Berkeley National Laboratory

<http://sdm.lbl.gov/srm-wg>

Intellectual Property Statement

The GGF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the GGF Secretariat.

The GGF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this recommendation. Please address the information to the GGF Executive Director.

Full Copyright Notice

Copyright (C) Global Grid Forum (2003). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the GGF or other organizations, except as needed for the purpose of developing Grid Recommendations in which case the procedures for copyrights defined in the GGF Document process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the GGF or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE GLOBAL GRID FORUM DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE."

Storage Resource Management Working Group

EDG-WP2: Peter Kunszt, Erwin Laure,
Heinz Stockinger, Kurt Stockinger

EDG-WP5: Jean-Philippe Baud, Jens Jensen,
Emil Knezo, Stefano Occhetti,
Owen Synge, Olof Barring

Jefferson Lab: Bryan Hess, Andy Kowalski, Chip Watson

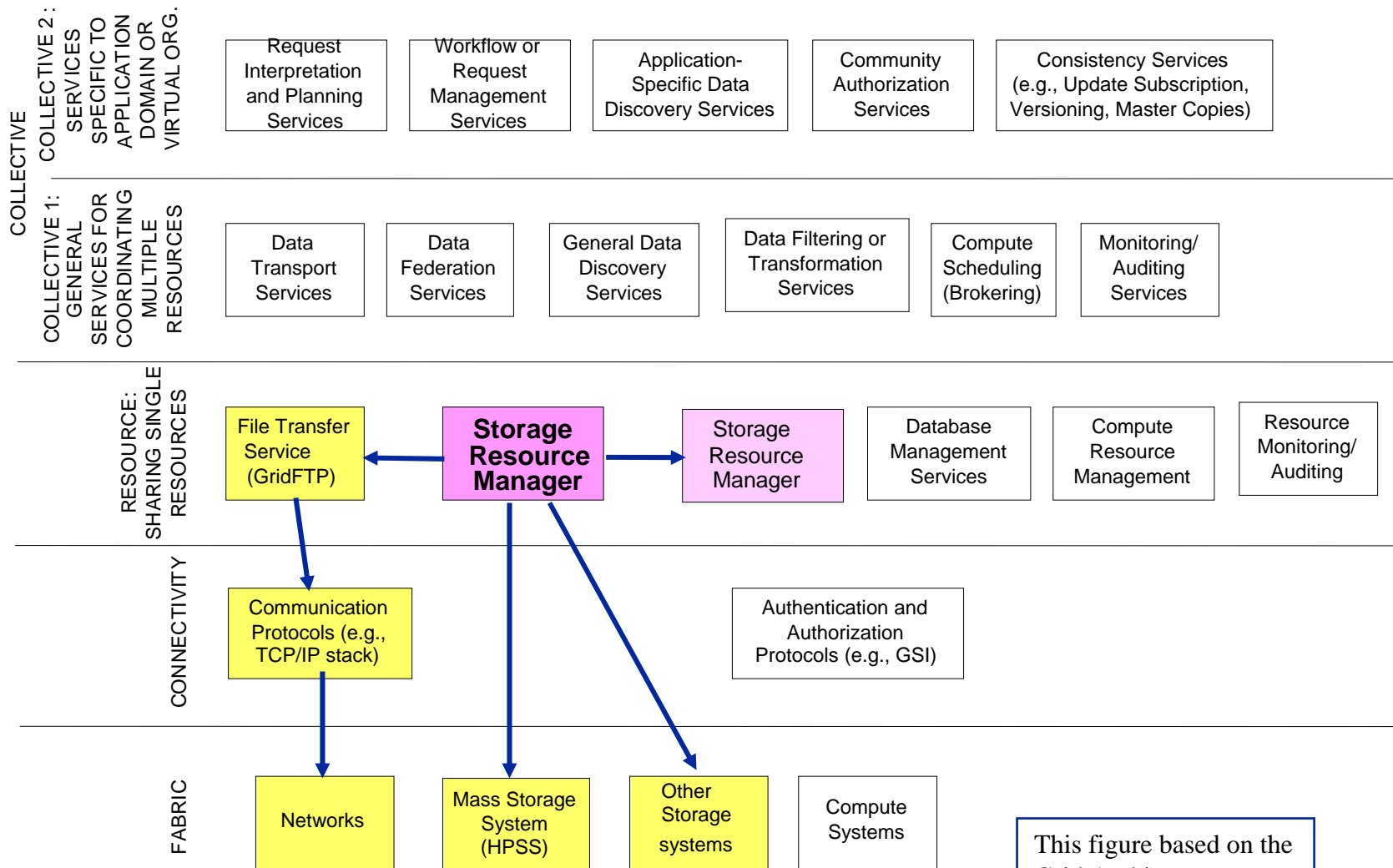
Fermilab: Don Petravick, Timur Perelmutov, Rich Wellner

LBNL: Junmin Gu , Arie Shoshani, Alex Sim

- **What are Storage Resource Managers**
- **General Analysis Scenario and the use of SRMs**
- **SRM concepts**
- **SRM functionality**
- **Example functions**
- **Features of Basic vs. Advanced SRMs
(discussion / opinions)**

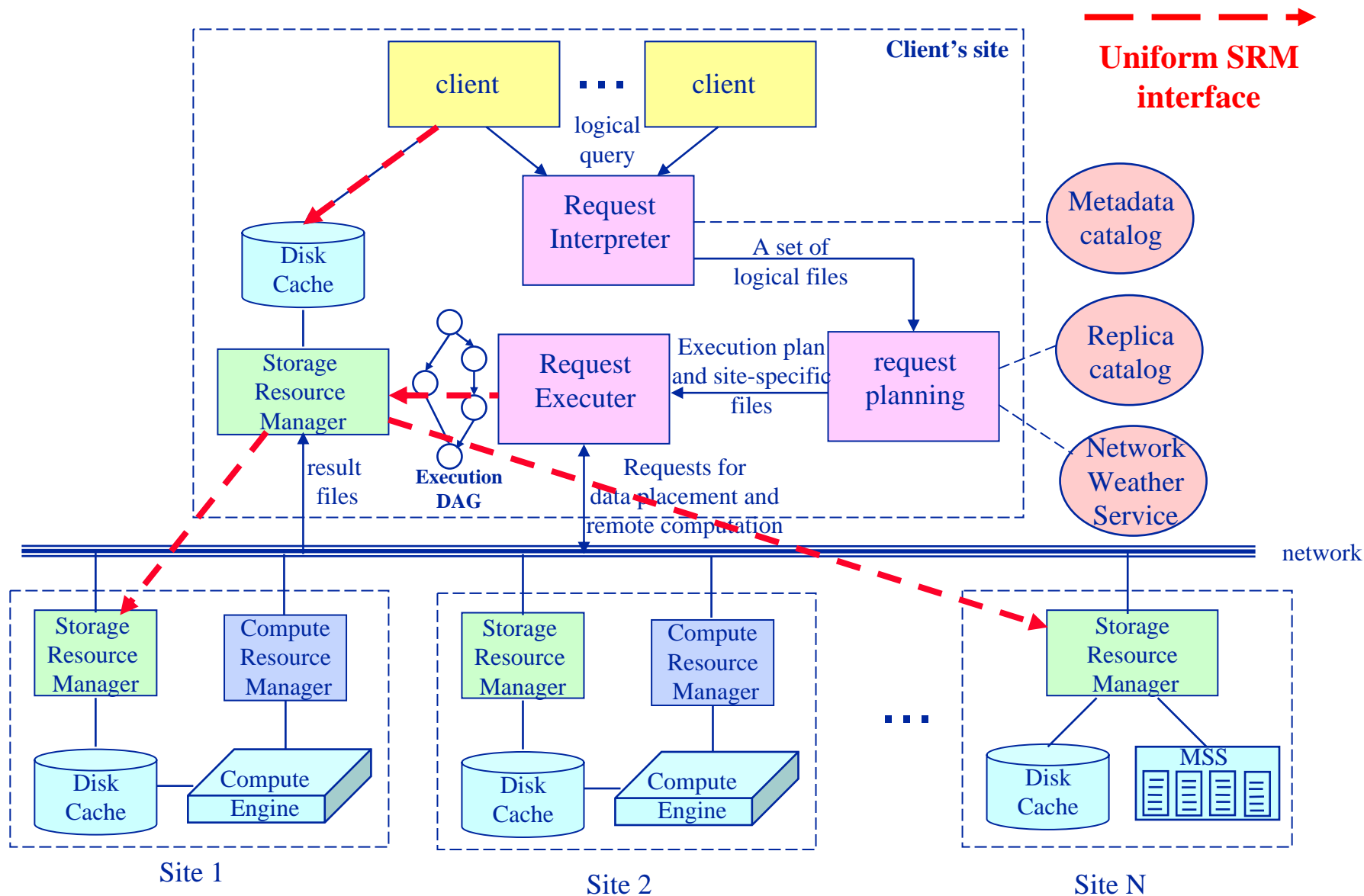
- **Grid architecture needs to include reservation & scheduling of:**
 - Compute resources
 - Storage resources
 - Network resources
- **Storage Resource Managers (SRMs) role in the data grid architecture**
 - Shared storage resource allocation & scheduling
 - Specially important for data intensive applications
 - Often files are archived on a mass storage system (MSS)
 - Wide area networks – minimize transfers
 - large scientific 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

SRMs supports a brokering service by invoking transfer services



This figure based on the
Grid Architecture paper
by Globus Team

General Analysis Scenario

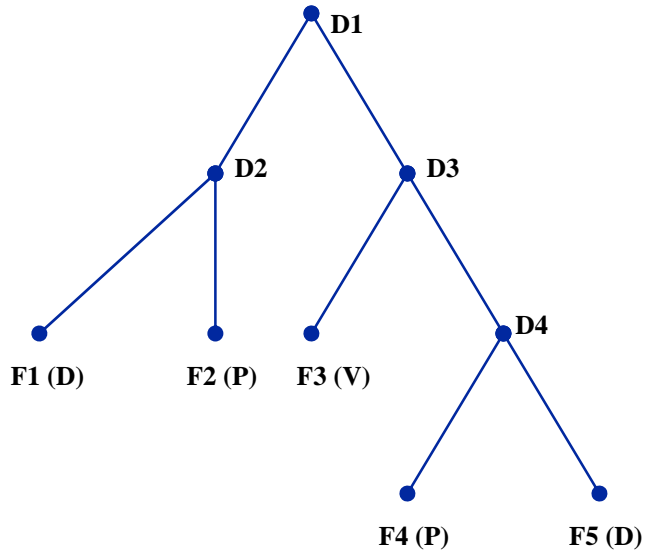


- **Volatile: temporary files with a lifetime guarantee**
 - Files are “pinned” and “released”
 - Files can be removed by SRM when released or when lifetime expires
- **Permanent**
 - No lifetime
 - Files can only be removed by creator (owner)
- **Durable: files with a lifetime that CANNOT be removed by SRM**
 - Files are “pinned” and “released”
 - Files can only be removed by creator (owner)
 - If lifetime expires – invoke administrative action (e.g. notify owner, archive and release)

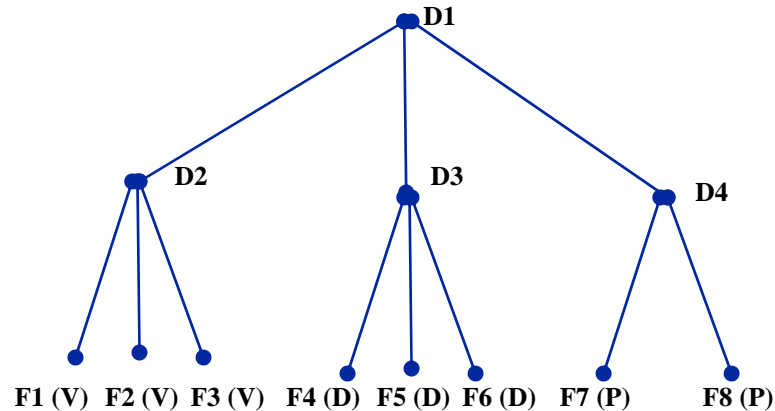
- **Types**
 - **Volatile**
 - Space can be reclaimed by SRM when lifetime expires
 - **durable**
 - Space can be reclaimed by SRM only if it does NOT contain files
 - Can choose to archive files and release space
 - **Permanent**
 - Space can only be released by owner or administrator
- **Assignment of files to spaces**
 - Files can only be assigned to spaces of the same type
- **Spaces can be reserved**
 - No limit on number of spaces
 - Space reference handle is returned to client
 - Total space of each type are subject to SRM and/or VO policies
- **Default spaces**
 - Files can be put into SRM spaces without explicit reservation
 - Defaults are not visible to client
- **Compacting space**
 - Release all unused space – space that has no files or files whose lifetime expired

- **Usual unix semantics**
 - srmLs, srmMkdir, srmMv, srmRm, srmRmdir
- **A single directory for all file type**
 - No directories for each type
 - File assignment to types is virtual
 - File can be placed in SRM-managed directories by maintaining mapping to client's directory
- **Access control services**
 - Support owner/group/world permission
 - Can only be assigned by owner
 - When file requested by user, SRM should check permission with source site

Examples of Directory Structures (user defined)



(1) Mixed file types



(2) By file type

- **Supported function: ChangeFileType**
- **Advantage of (1): no need to move files when file types are changed**

- **Negotiation**

- Client asks for space: guaranteed, MaxDesired
- SRM return: guaranteed \leq client-requested, best effort \leq MaxDesired

- **Type of space**

- Can be specified
- Subject to limits per client (SRM or VO policies)
- Default: volatile

- **Lifetime**

- Negotiated: C-lifetime requested
- SRM return: S-lifetime \leq C-lifetime

- **Reference handle**

- SRM returns space reference handle
- User can provide: srmSpaceTokenDescription to recover handles

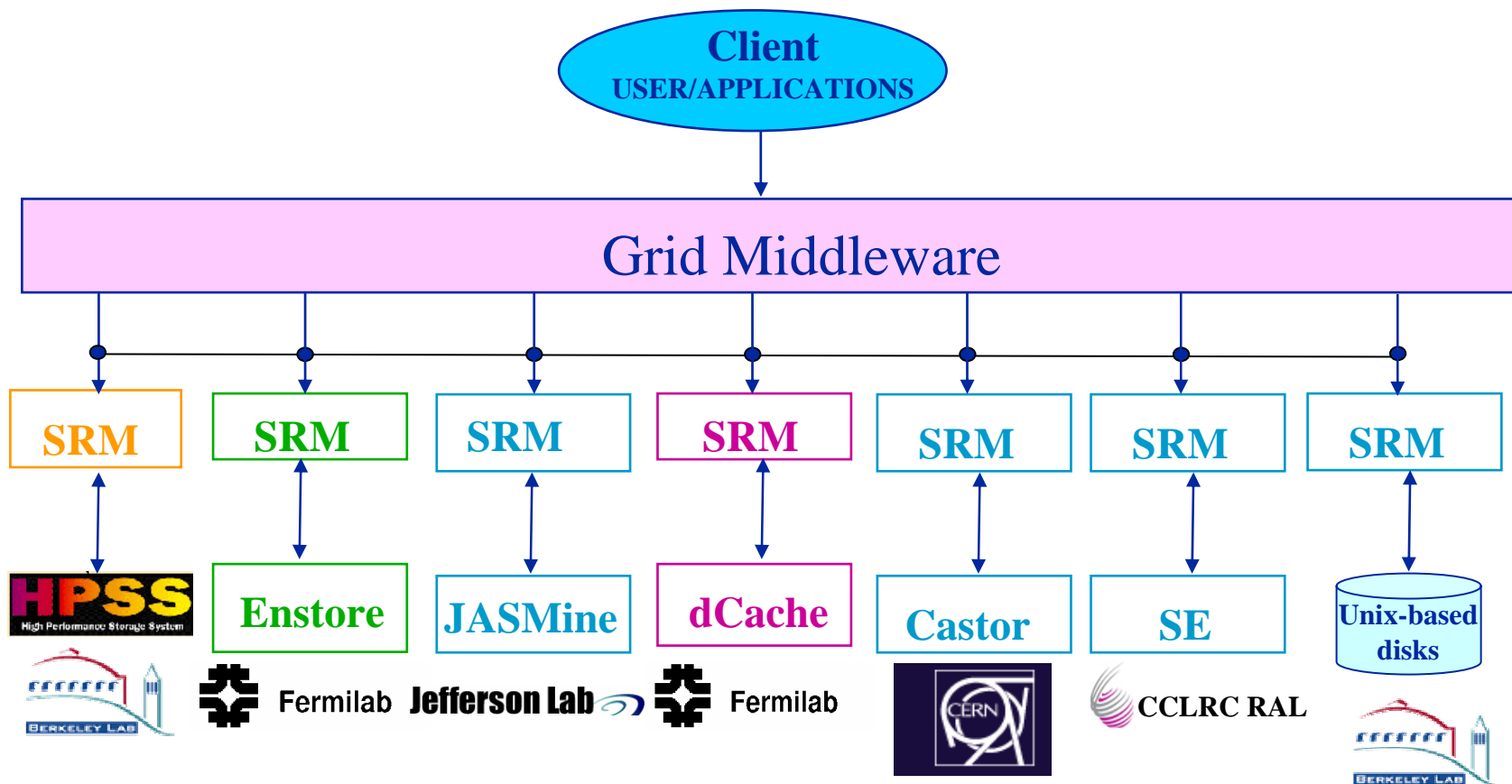
- **Negotiation**
 - Client provides an ordered list
 - SRM return: highest possible protocol it supports
- **Example**
 - Protocols list: bbftp, gridftp, ftp
 - SRM returns: gridftp
- **Advantages**
 - Easy to introduce new protocols
 - User controls which protocol to use
 - Default – SRM policy choice
- **How it is returned?**
 - The protocol of the Transfer URL (TURL)
 - Example: bbftp://dm.slac.edu/temp/run11/File678.txt

- **Can srmRequestToGet multiple files**
 - Required: Files URLs
 - Optional: space file type, space handle, Protocol list
 - Optional: total retry time
- **Provide: Site URL (SURL)**
 - URL known externally – e.g. in Rep Catalogs
 - e.g. srm://sleepy.lbl.gov:4000/tmp/foo-123
- **Get back: transfer URL (TURL)**
 - Path can be different that in SURL – SRM internal mapping
 - Protocol chosen by SRM
 - e.g. gridftp://dm.lbl.gov:4000/home /level1/foo-123
- **Managing request queue**
 - Allocate space according to policy, system load, etc.
 - Bring in as many files as possible
 - Provide information on each file brought in or pinned
 - Bring additional files as soon as files are released
 - Support file streaming

- **Space reservation**
 - Negotiate and assign space to users
 - Manage “lifetime” of spaces
 - Release and compact space
- **File management**
 - Assign space for putting files into SRM
 - Pin files in storage when requested till they are released
 - Manage “lifetime” of files
 - Manage action when pins expire (depends on file types)
- **Get files from remote locations when necessary**
 - Purpose: to simplify client’s task
 - srmCopy: in “pull” and “push” modes

- **Space management policies and file sharing**
 - Policies on what should reside on a storage resource at any one time
 - Policies on what to evict when space is needed
 - Share files to avoid getting them from remote locations
- **Manage multi-file requests**
 - Queues file requests, pre-stage when possible
- **Status functions**
 - Files: lifetime remaining, what's available locally
 - Requests: what files are available (needed in lieu of callbacks)
 - Request summary: for progress report
 - Space metadata: space in use, space available, lifetime
- **Provide grid access to/from mass storage systems**
 - HPSS (LBNL, ORNL, BNL), Enstore (Fermi), JasMINE (Jlab), Castor (CERN), MSS (NCAR), SE (RAL) ...

Uniformity of Interface → Compatibility of SRMs



File Movement

srm(PrepareTo)Get
srm(PrepareTo)Put
srmCopy

Lifetime management

srmReleaseFiles
srmPutDone
srmExtendFileLifeTime

Terminate/resume

srmAbortRequest
srmAbortFile
srmSuspendRequest
srmResumeRequest

Space management

srmReserveSpace
srmReleaseSpace
srmUpdateSpace
srmCompactSpace

FileType management

srmChangeFileType

Status/metadata

srmGetRequestStatus
srmGetFileStatus
srmGetRequestSummary
srmGetRequestID
srmGetFilesMetaData
srmGetSpaceMetaData

srmPrepareToGet

In:	TUserID	userID,
	TGetFileRequest[]	<u>arrayOfFileRequest,</u>
	string[]	arrayOfTransferProtocols,
	string	userRequestDescription,
	TStorageSystemInfo	storageSystemInfo,
	TLifeTimeInSeconds	TotalRetryTime
Out:	TRequestToken	<u>requestToken,</u>
	TReturnStatus	<u>returnStatus,</u>
	TGetRequestFileStatus[]	arrayOfFileStatus

srmReserveSpace

In:	TUserID	userID,
	TSpaceType	<u>typeOfSpace,</u>
	String	userSpaceTokenDescription,
	TSizeInBytes	sizeOfTotalSpaceDesired,
	TSizeInBytes	sizeOfGuaranteedSpaceDesired,
	TLifeTimeInSeconds	lifetimeOfSpaceToReserve,
	TStorageSystemInfo	storageSystemInfo
Out:	TSpaceType	typeOfReservedSpace,
	TSizeInBytes	sizeOfTotalReservedSpace,
	TSizeInBytes	sizeOfGuaranteedReservedSpace,
	TLifeTimeInSeconds	lifetimeOfReservedSpace,
	TSpaceToken,	referenceHandleOfReservedSpace,
	TReturnStatus	<u>returnStatus</u>

Features in Basic vs. Advanced SRM

	BASIC	ADVANCED
• File movement		
• PrepareToGet	yes	yes
• PrepareToPut	yes	yes
• Copy	no	yes
• Request capabilities		
• Multi-file Streaming	yes	yes
• Trans. Prot. Negotiation	yes	yes
• File lifetime negotiation	no	yes
• File types		
• Volatile	yes	yes
• Permanent	yes (for MSS)	yes
• durable	no	yes

Features in Basic vs. Advanced SRM

	BASIC	ADVANCED
• Space reservations		
• Space-time negotiation	no	yes
• Space types	no	yes
• Remote access		
• gridFTP	yes?	yes
• Other SRMs	yes?	yes
• User-specified Directory		
• Volatile	no	yes
• Permanent	yes	yes
• Durable	no	yes
• Terminate/suspend		
• Abort file	yes	yes
• Abort request	yes	yes
• Suspend/resume request	no	yes