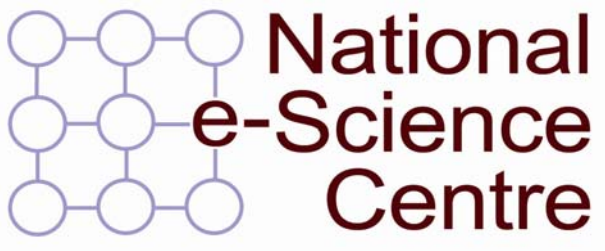
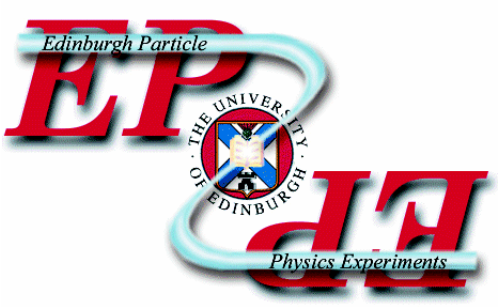
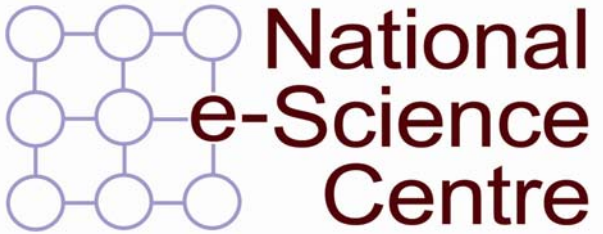


Storage Resource Managers

Functionality and Integration



Storage Management

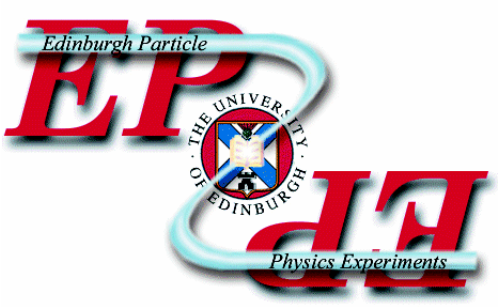


What is storage management?

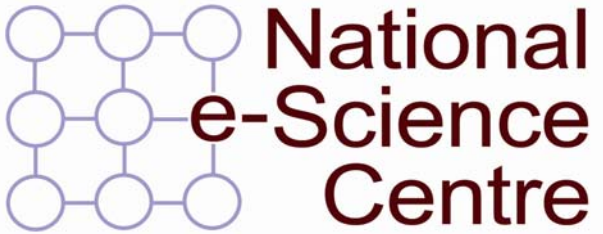
Why is storage management important?

How can we solve these problems on the Grid?

Where do I fit in?



Storage Management



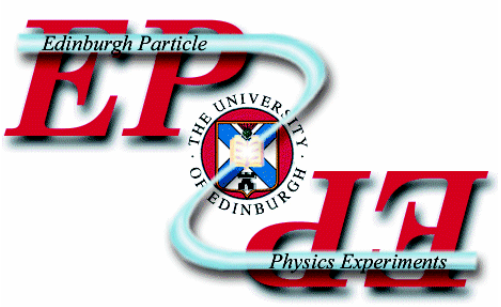
‘Grid Vision’ to bring distributed and disparate

compute

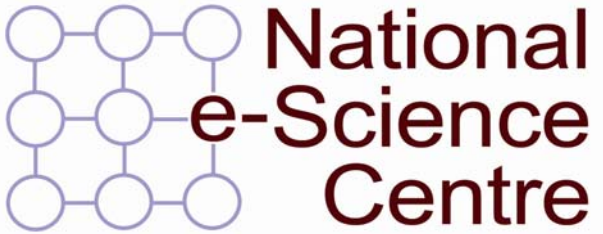
storage

network

resources together to give user impression their job in running on local system.



Storage Management



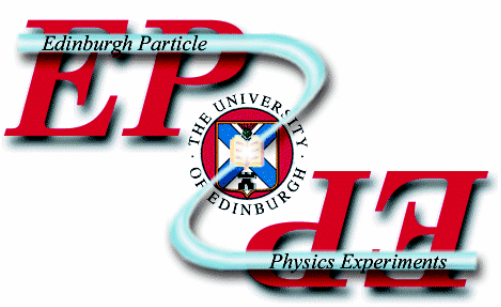
‘Grid Vision’ to bring distributed and disparate

compute

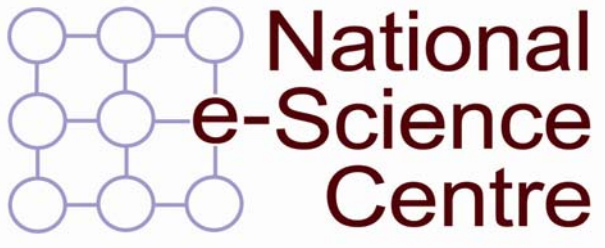
storage

network

resources together to give user impression their job in running on local system.



Important…?



Grid is dynamic collection of resources across many administrative domains.

Many Grid applications are data as well as compute intensive.

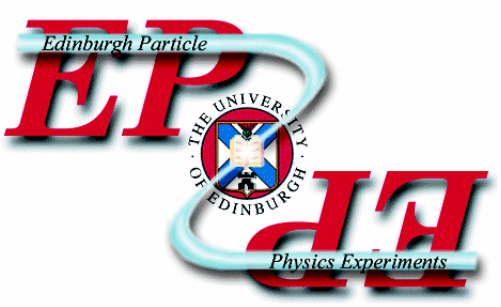
No problems if ALL clients have static space allocation on ALL administrative domains for ALL their requirements till the end of time…(?!)

Solution…

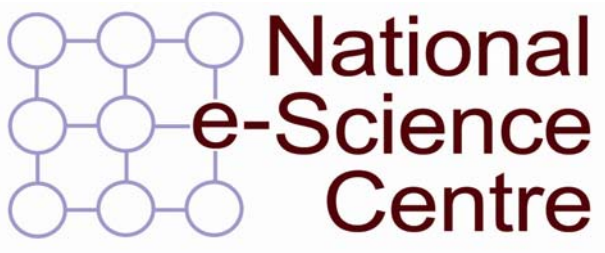
‘Storage Management Working Group’ formed in 2001 to address these issues.

Later became GGF ‘Grid Storage Management Working Group’

Aim: produce a standard interface to storage devices to allow dynamic management of resources.



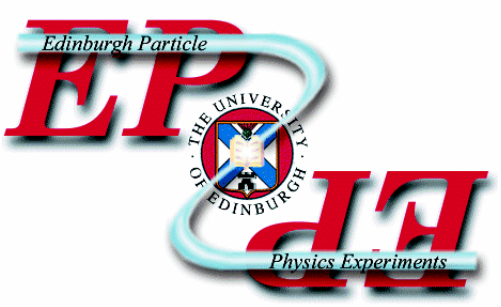
Solution…



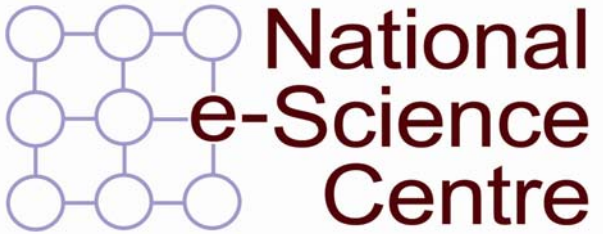
‘Storage Management Working Group’ formed in 2001 to address these issues.

Later became GGF ‘Grid Storage Management Working Group’

Aim: produce a standard interface to storage devices to allow dynamic management of resources.

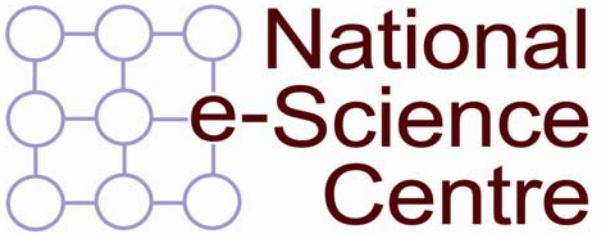


Solution…



Storage Resource Managers

Definition



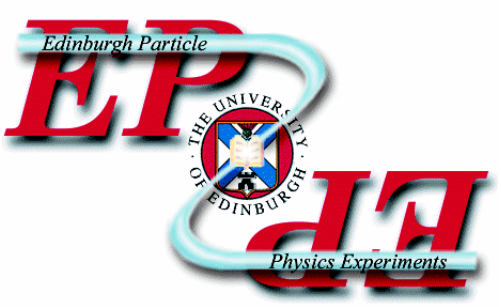
Storage Resource Manager (SRM):

Middleware component whose function is to provide dynamic

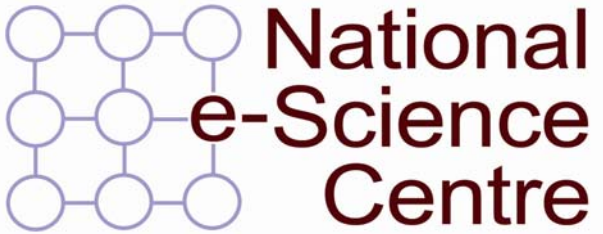
space allocation AND

file management

on shared storage components on the grid.



Definition



Storage Resource Manager (SRM):

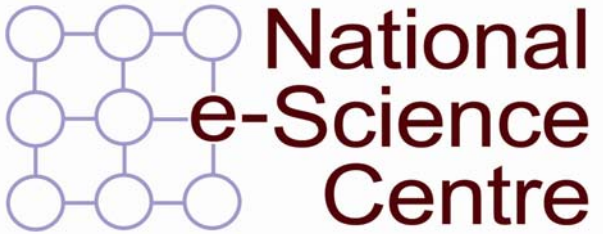
Middleware component whose function is to provide dynamic

space allocation AND

file management

on shared storage components on the grid.

Common Interface



Grid clients want seamless access to data.

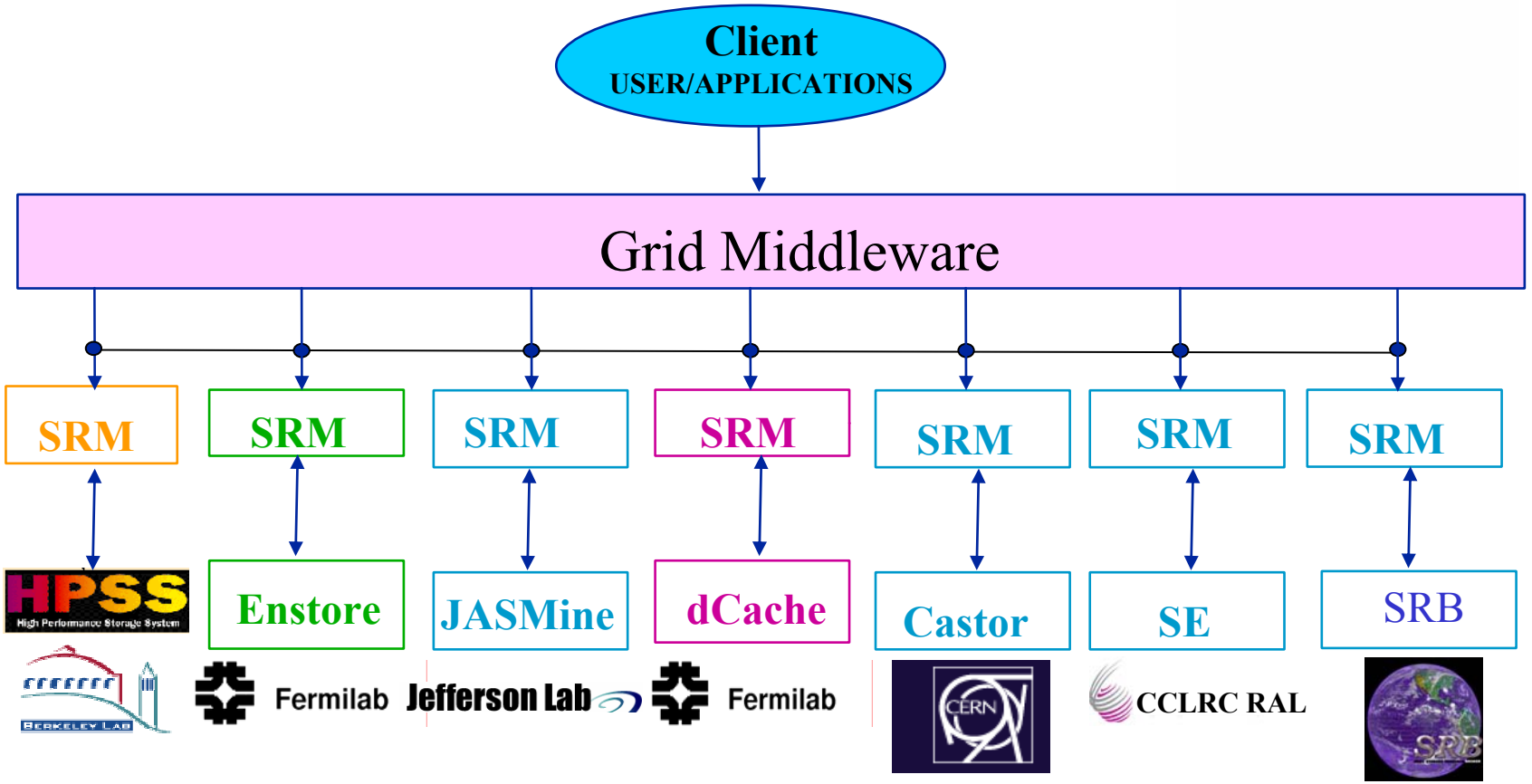
Don’t care where data stored…

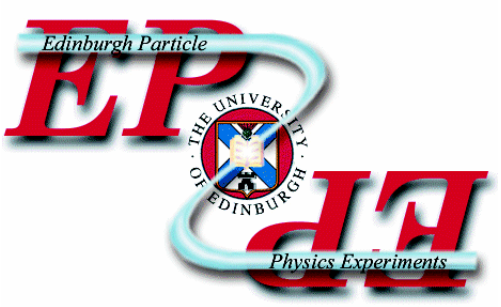
…or type of storage device.

All they want is common interface standard.

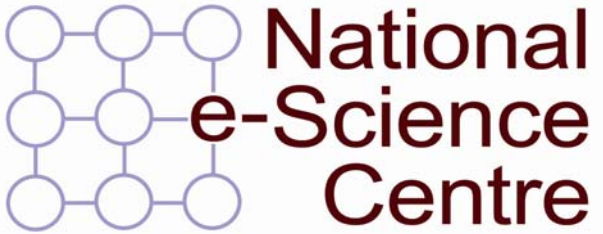
Common Interface

Courtesy of Berkeley Lab Website



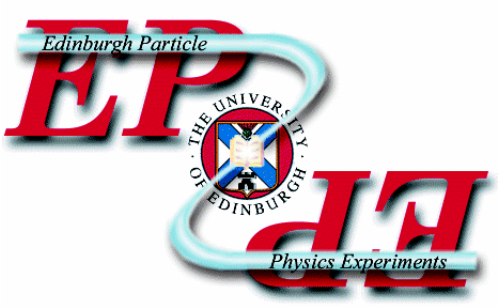


Dynamic Space Management

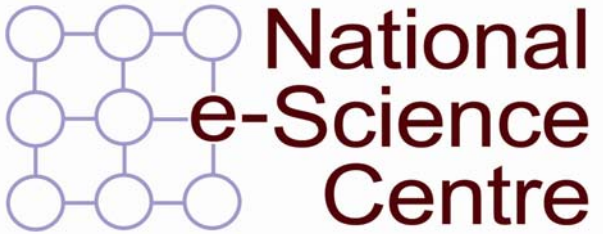


Want to:

- Optimise the efficiency of Grid schedulers and planners storage interactions.
- Space reservation in real time
  - Increase the efficiency of job allocations
  - Fewer failures due to lack of storage.
- Create space by removing in-frequently used files
  - Reduces impact of forgotten files.



Dynamic File Management

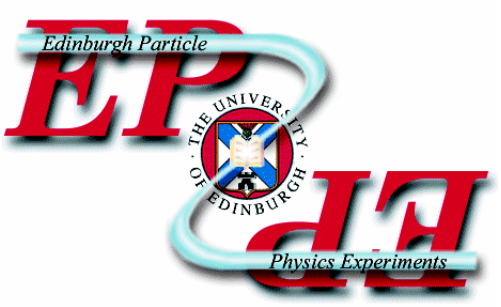


Want to:

Allow clients to copy/transfer files around storage sites on grid. Easy replication.

Store files safely temporarily and permanently.

Allow files to be shared between clients.

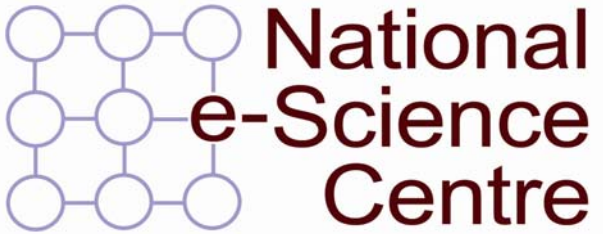


Simplify

Want to:

- Simplify process of making multi-file requests.
- Remove need for clients to submit and monitor multiple requests.
- Insulate clients from network and storage device failures.

Functionality Overview 1



Three FILE types

- Volatile, Durable, Permanent

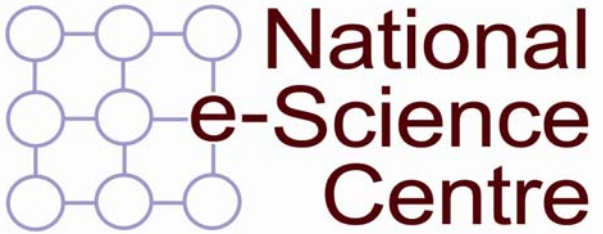
File ‘pinning’

- Garbage Collection

Three SPACE types

- Volatile, Durable, Permanent

Functionality Overview 2



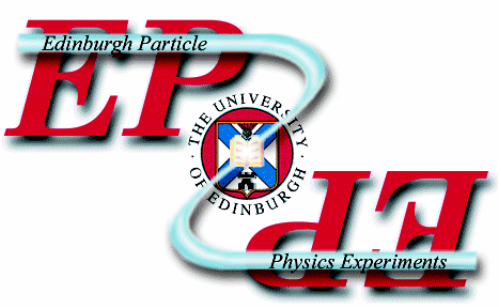
Space Reservation

- Guaranteed
- Best Effort Space

File Transfers

- Get, Put, Copy

Multi-file Requests



File Types

‘Volatile’ files are those which can be stored temporarily.

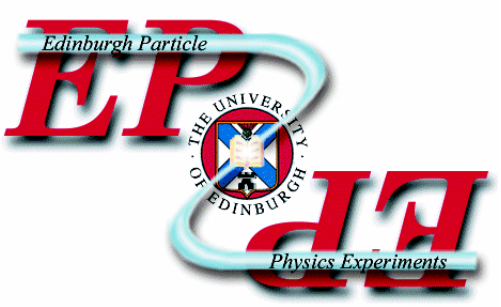
lifetime associated

‘owned’ by the SRM.

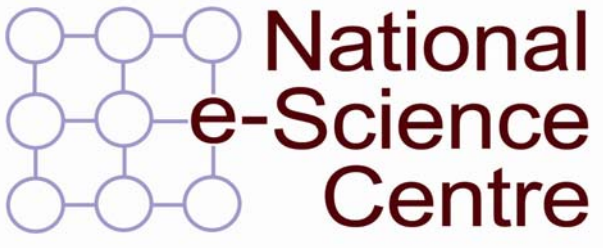
‘Permanent’ files are those which are archived.

no lifetime associated

‘owned’ by the client.

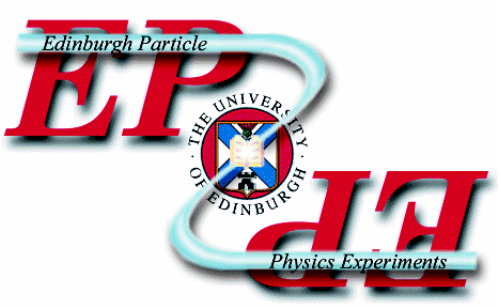


File Types

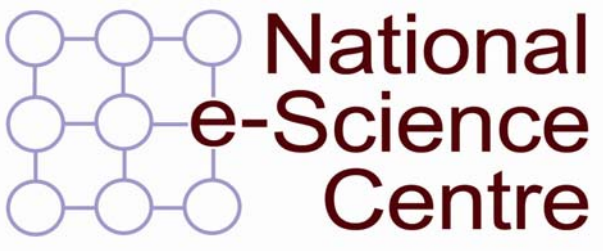


‘Durable’ files have both ‘Volatile’ and ‘Permanent’ characteristics.

- lifetime associated (like volatile).
- owned by client and can’t be removed by SRM (like permanent).
- temporary storage for real time data taking.
- guarantee files won’t be removed before archiving.



File Pinning



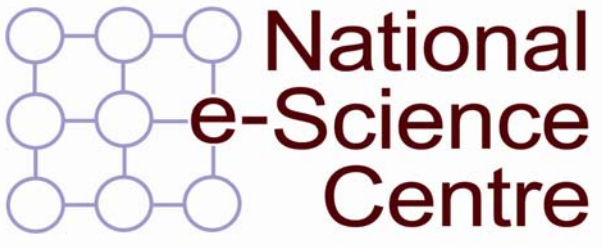
Soft guarantee a file will be present for lifetime.

For lifetime of pin the file can't be removed.

At end of pin:

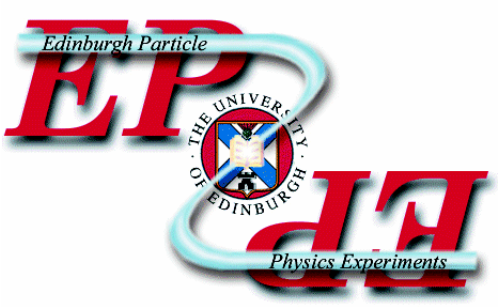
- Volatile files may be removed by the SRM.
- The owner of durable files are informed of pin expiration

‘Garbage Collection’

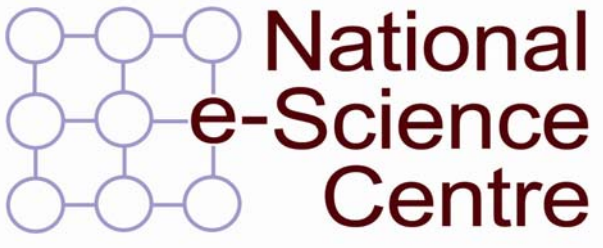


SRM create space using ‘Garbage Collection’

- Volatile Files who’s lifetimes are expired are removed.
- These are not necessarily removed when lifetime expired but when space is needed.
- Allows SRM control of disk cache.



Space Types

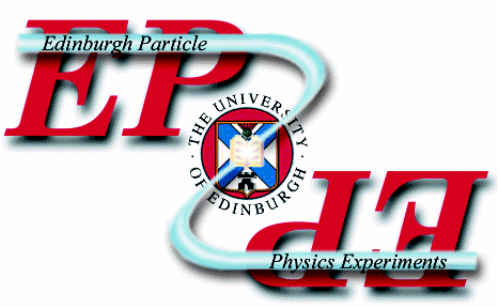


Three space types supported

Volatile, Durable and Permanent.

But, why implement both space and file types?

Can't the files of different types be stored in a homogenous space allocated to the client?



Space Types

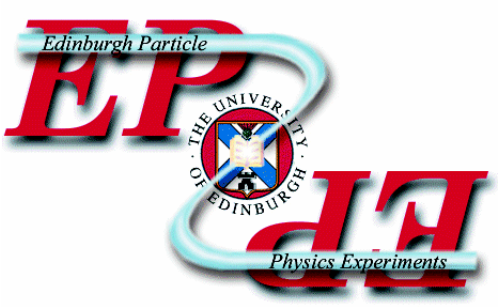
Different space types are required to support space reservation.

Grid clients reserve space of type required.

‘Volatile’ space required only temporarily.

‘Durable’ for secure temporary space.

‘Permanent’ space for archival purpose.



Space Reservation

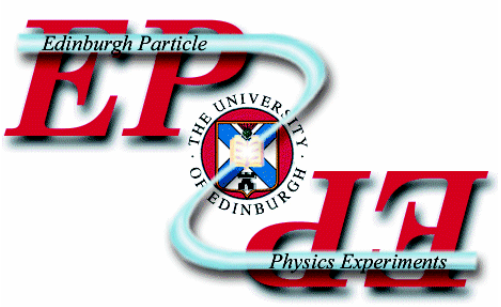
Space reservations analogous to file pins.

Guarantee the space for given lifetime.

At end of reservation:

- Volatile space reclaimed and all files deleted.
- Unused durable space reclaimed. Owner of space and files present informed of expiration.
- Permanent space has no lifetime.

Only works if file lifetimes shorter than space lifetime.



Space Reservation

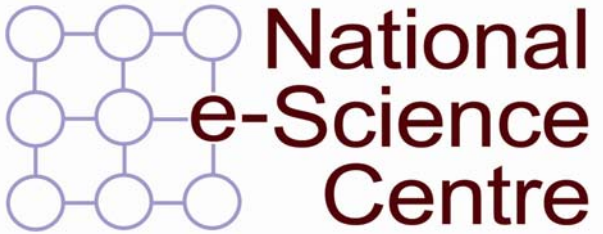
Guaranteed space in ideal world.

Not practical in shared resources of real-world.

Best-effort reservations offer alternative

- Clients provisionally offered space
- More offered as space used
- Avoids wasted unused space.

Reservation in Practise



Negotiation of space size and lifetime:

Client requests size: C-guaranteed, MaxDesired

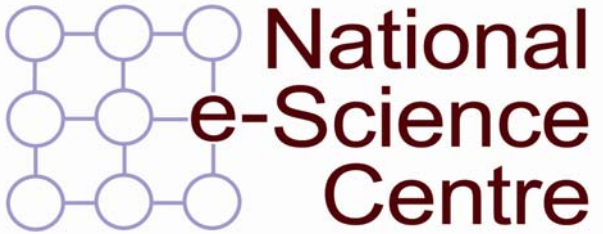
SRM returns: S-guaranteed <= C-guaranteed, best effort <= MaxDesired

Clients requests: C-lifetime

SRM returns: S-lifetime <= C-lifetime

SRM returns ‘Token’ for future management.

File Transfers



Initially only two transfer functions supported

srmGet – client gets file from SRM

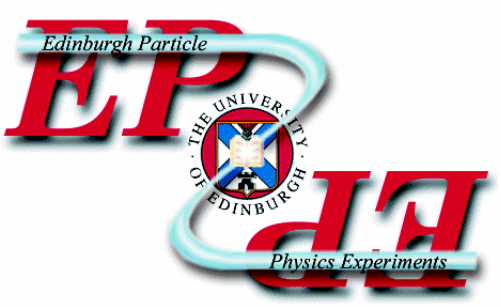
srmPut – client puts file into SRM

Later a third srmCopy function implemented

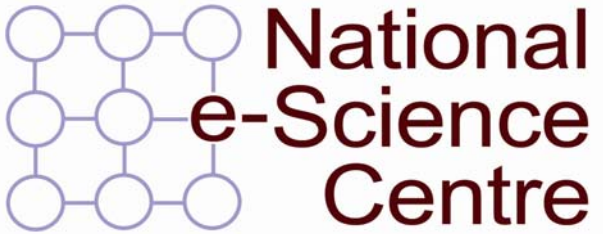
Third-party transfer between two SRMs

Subsequent SRM specifications allow hybrid

srmCopyandGet, srmPutandCopy etc…



Multi-File Requests



SRMs support single requests containing multiple files.

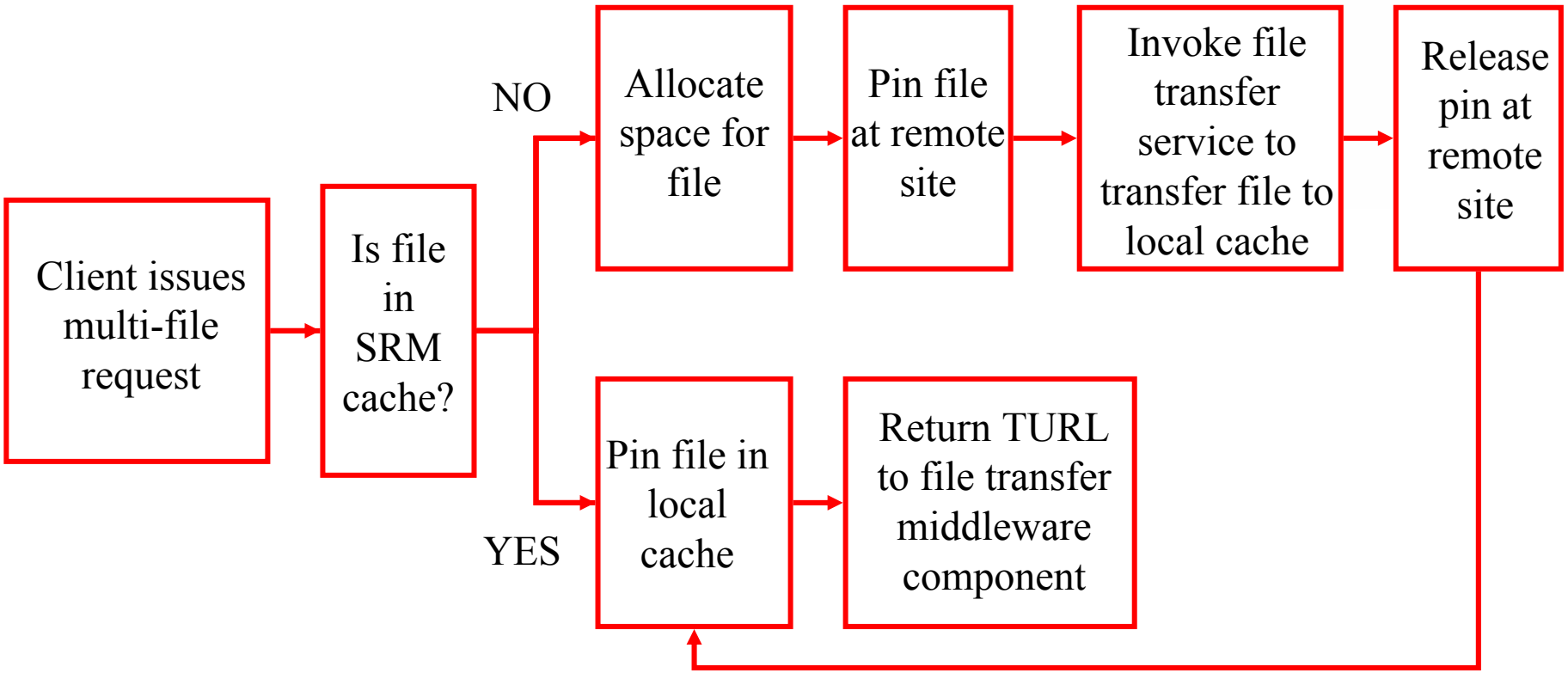
Shields clients from multitude of Gets/Puts/Copies.

Allows SRM to order files for efficiency.

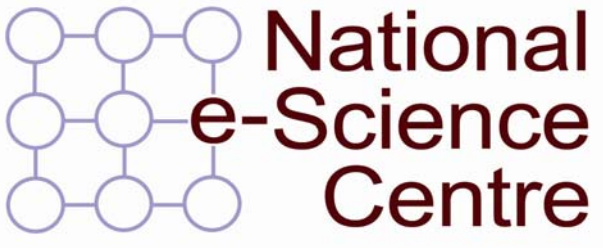
SRM monitors transfers.

Can restart transfers which fail.

Example workflow



Protocol Negotiation



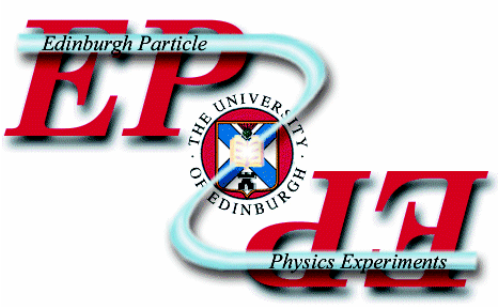
Allow transfer protocol negotiation because:

- No one transfer protocol available to all clients
- Support future protocols

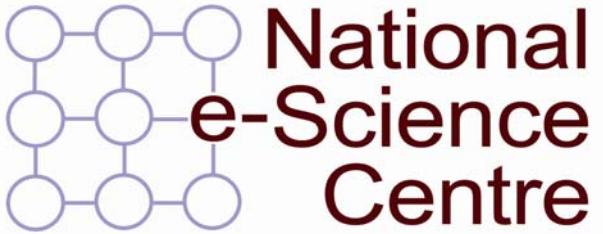
Client provides an ordered list

- e.g. bbftp, gridftp, ftp

SRM returns highest possible protocol it supports



Summary of Functionality

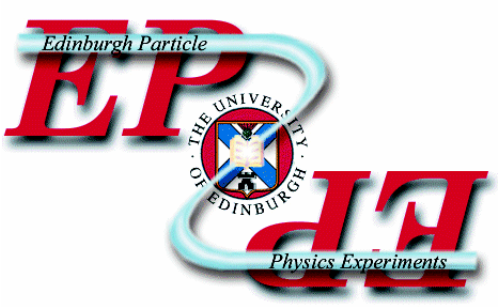


Space reservation

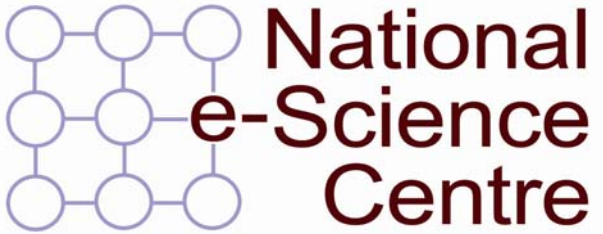
- Negotiate and assign space to users
- Manage lifetime of spaces
- Release and compact space

File management

- Pin files in storage when requested till they are released
- Manage lifetime of files
- Manage action when pins expire (depends on file types)



Summary of Functionality



File Transfers

Put files into and retrieve files from SRM

Get files from remote locations

Third-party copying of files

Simplify client's task (multi-file requests)

Implementation

Current SRM specification implements functionality with:

Data Transfer Functions

srmPrepareToGet
srmPrepareToPut
srmCopy
srmRemoveFiles
srmReleaseFiles
srmPutDone
srmExtendFileLifeTime

Space Management Functions

srmReserveSpace
srmReleaseSpace
srmUpdateSpace
srmCompactSpace
srmGetSpaceMetaData
srmChangeFileStorageType
srmGetSpaceToken

Implementation

Current SRM specification implements functionality with:

Status functions

- srmStatusOfGetRequest
- srmStatusOfPutRequest
- srmStatusOfCopyRequest
- srmGetRequestSummary
- srmGetRequestID

Abort/resume

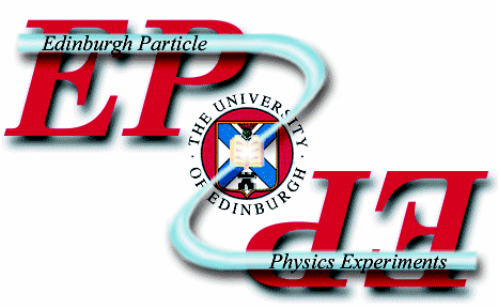
- srmAbortRequest
- srmAbortFiles
- srmSuspendRequest
- srmResumeRequest

Directory Functions

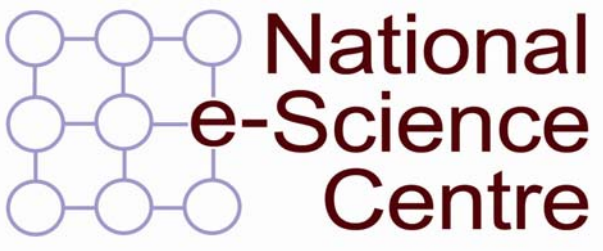
- srmMkdir
- srmRmdir
- srmRm
- srmLs
- srmMv

Permission Functions

- srmSetPermission
- srmReassignToUser
- srmCheckPermission

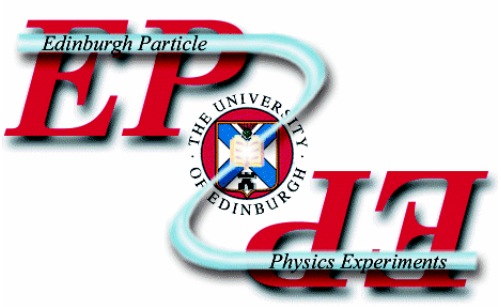


Where do I fit in?

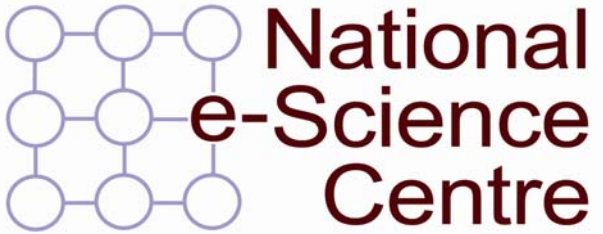


Stuart (hopefully) talked a bit about DIRAC within the LHCb experiment at LHC…

I will start working in at CERN integrating LHCb's DIRAC agent with SRM to allow dynamic access to data.


Edinburgh Particle
THE UNIVERSITY OF EDINBURGH
Physics Experiments

Questions…


National
e-Science
Centre

?