Critical features for WLCG

- Result of WLCG Baseline Services Working Group
  - http://cern.ch/lcg/PEB/BS

- Originally planned to be implemented by WLCG Service Challenge 4
  - Delayed until autumn 2006

- Features from version 1.1 + critical subset of version 2.1

(Nick Brook, SC3 planning meeting – June ’05)

  - File types
  - Space reservation
  - Permission functions
  - Directory functions
  - Data transfer control functions
  - Relative paths
  - Query supported protocols

File types

- Volatile
  - Temporary and sharable copy of an MSS resident file
  - If not pinned it can be removed by the garbage collector as space is needed (typically according to LRU policy)
- Durable
  - File can only be removed if the system has copied it to an archive
- Permanent
  - System cannot remove file
- Users can always explicitly delete files
- The experiments only want to store files as permanent
  - Even scratch files → will be explicitly removed by experiment

Space reservation

- v1.1
  - Space reservation done on file-by-file basis
  - User does not know in advance if SE will be able to store all files in multi-file request
- v2.1
  - Allows for a user to reserve space
    - But can 100 GB be used by a single 100 GB file or by 100 files of 1 GB each?
    - MSS space vs. disk cache space
  - Reservation has a lifetime
  - “PrepareToGet(Put)” requests fail if not enough space
- v3.0
  - Allows for “streaming”
    - When space is exhausted requests wait until space is released
  - Not needed for SC4
- What about quotas?
  - Strong interest from LHC VOs, but not yet accepted as task for SRM

Permission functions

- v2.1 allows for POSIX-like ACLs
  - Can be associated per directory and per file
  - Parent directory ACLs inherited by default
  - Can no longer let a simple UNIX file system deal with all the permissions
    - Need file system with ACLs or ACL-aware permission manager in SRM
      - May conflict with legacy applications

- LHC VOs desire storage system to respect permissions based on VOMS roles and groups
  - Currently only supported by DPM

- File ownership by individual users not needed in SC4
  - Systems shall distinguish production managers from unprivileged users
    - Write access to precious directories, dedicated stager pools
    - Supported by all implementations

Directory functions

- Create/remove directories

- Delete files
  - v1.1 only has an “advisory” delete
    - Interpreted differently by different implementations
      - Complicates applications like the File Transfer Service

- Rename directories or files (on the same SE)

- List files and directories
  - Output will be truncated to implementation-dependent maximum size
    - Full (recursive) listing could tie up or complicate server (and client)
      - May return huge result
      - Could return chunks with cookies → server would need to be stateful
    - It is advisable to avoid very large directories

- No need for “mv” between SEs

Data transfer control functions

- StageIn, stageOut type functionality
  - prepareToGet, prepareToPut

- (a way for) Pinning and unpinning files
  - Avoid untimely cleanup by garbage collector
  - Pin has a lifetime, but can be renewed by client
    - Avoid dependence on client to clean up

- Monitor status of request
  - How many files ready
  - How many files in progress
  - How many files left to process
- Suspend/resume request
  - Not needed for SC4
- Abort request

Relative paths

- Everything should be defined with respect to the VO base directory

- Example:

  srm://srm.cern.ch/castor/cern.ch/grid/lhcb/DC04/prod0705/0705_123.dst

- SE defined by protocol and hostname

- VO base directory is the storage root for the VO
  - Advertized in information system, but unnecessary detail
    - Clutters catalog entries
    - SRM could insert VO base path automatically
      - Available in dCache

- VO namespace below base directory

Query supported protocols

- List of transfer protocols per SE available from information system
  - Workaround, complicates client
  - SRM knows what it supports, can inform client
- Client always sends SRM a list of acceptable protocols
  - gsiftp, (gsi)dcap, rfio, xrootd, root, ...
  - SRM returns TURL with protocol applicable to site
- Query not needed for SC4

More coordination items

- SRM compatibility tests
  - Test suite of Jiri Mencak (RAL)
  - Test suite for GGF-GIN by LBNL
  - Test suite of Gilbert Grosdidier (LCG)
  - ...
  - Which one(s) will do the job for WLCG?

- Clients need to keep supporting v1.1
  - First try v2.x?
- Some implementations need v2.x to be on separate port
  - 8444 standard?

- xrootd integration
- rfio incompatibility
- Quotas for user files
- ...