SRM+Cloud session

Status quo: SRM

- SRM protocol
  - **Problems**:
    - Not all implementations support all functionality.
    - Not all storage supports SRM.
    - Only a fraction of SRM protocol is actually used.
  - Current **in production** and **works**, but **seen as high maintenance**.
- Cloud:
  - Defined by a **service**, ***not*** by **standards.**
    - Amazon S3 service defines a de facto standard,
    - CDMI is a standardisation effort (from SNIA), but does it have traction?
  - Often with **limited functionality** (e.g., lack of hierarchical namespace),
    (but it it good enough?)
  - **Microsoft SkyDrive** is somewhat unique for using a standard: WebDAV
- WebDAV
  - It's a **standard**. Lots of **clients, covering all platforms** (most are somewhat buggy, though). Might do everything needed; support in SEs coming with EMI-2.

Discussion at F2F

- Interesting that (in general) people **do not distinguish** between:
  - Bad: because that's just **how we use the software**,
  - Bad: because of the **software implementation**,
  - Bad: because of limitations of **the protocols**.
- Whole swathes of SRM functionality that is **not being used**:
  - Decoupling space-reservation from namespace,
  - Dynamic space-reservations,
  - ...

F2F discussion points

- Cloud vs SRM:
  - **Unclear** if Cloud provides **any benefit**, at protocol level.
  - Can we work with the **limitations** of Cloud-like protocols?
  - Should consider USA's **push toward virtualisation**

    Sites may provide commodity access to storage via Cloud APIs
- Question: if an experiment were given access to x Petabytes for 1—2 months, could use it?

  Answer: just now, almost certainly no.
- Is WebDAV an option:
  - What is missing, if anything? How do we find out what's missing?

F2F discussion points

- Need to separate SRM into **core functionality blocks**
  - This allows identification of **which parts are in use**,
  - Needed as a **framework** to make above questions tractable.
  - We allows us to consider protocols as **partial replacements** for SRM, but keeping SRM for the rest.
- Started identifying core functionality during the meeting:
  - Identified four core areas:

    Transfer management, Interacting with namespace, Aggregated space querying, Storage management.
  - Work is on-going
  - Here's the initial results ...

SRM initial breakdown

- Transfer management:
  - (GET / PUT) – operating on complete files; Ability to cancel an upload,
  - Negotiate direct access using another protocol,
  - Resource provisioning (uploading useful data),
  - Load balancing; back-pressure (SE tells client to slow down),
  - 3rd party copy
- Namespace interaction
  - Querying (stat), manipulation (rm, rmdir), data integrity (checksums)
- Aggregated space querying
  - Equivalent to POSIX: df / fstatfs
- Storage Management:
  - Bring online,
  - Pinning a file,
  - Cancelling a pin?

Moving forward

- Unresolved questions:
  - What do experiments **actually use** from SRM?
- Tasks:
  - Continue breaking SRM into components:
    - Identify actual usage by functional part,
    - Provides a framework for comparing different protocols,
    - Allows us to consider a protocol as a partial SRM replacement
      - The protocol replaces part of the SRM functionality, but SRM provides the remainder
  - Use this breakdown to compare alternatives
    - Analyse Cloud APIs and WebDAV
    - Identify their **limitations**,
    - Is adopting them **worth the cost** (in effort, diverted from other activity)?