

WLGC Storage (Management) Interfaces Working Group

Wahid Bhimji

(Some slides are stolen from
Markus Schulz's presentation to
WLCG MB on 19 June

Apologies to those who have seen
some of this before)

Motivation: From WLCG Storage/Data TEG

- ◉ SRM (v2.2) API is **near ubiquitous** at present
- ◉ There are **issues with relying on SRM** for management:
 - Not all storage is manageable through SRM – particularly storage implementations outside the HEP niche do not integrate SRM.
 - Not all Storage Element implementations provide complete implementation of SRM spec. Very little of SRM v2.2 spec. is used.
 - Performance concerns – e.g. overhead for transfers of small files.
- ◉ The TEG identified a subset of functionality of SRM that is required for the LHC experiments and what of that functionality was provided by alternatives.
 - For disk-only systems alternatives are feasible...
 - “Allowing” other interfaces could add to simplification, flexibility or performance.
 - But this is **not** about pushing sites, developers or experiments to abandon SRM; nor about designing a top down alternative
 - Where it works well and people are happy – that’s fine.

Motivation: TEG Recommendations

- ◉ **Maintain SRM at archive sites**
- ◉ **Experiments, middleware experts and sites should agree on alternatives to be considered for testing and deployment, targeting not the full SRM functionality but the subset detailed here, determined by its actual usage.**
 - ... recommend a small working group be formed, reporting to the GDB, to evaluate alternatives as they emerge...
- ◉ **Develop future interfaces:**
 - ... different approaches to integrate cloud-based storage resources need to be investigated ...

Mandate (as presented by Markus to WLCG MB: 19 June)

- Based on the TEG report specify for disk only (and archive and hierarchical systems)* the minimal required functionality based on the work in the Storage TEG.
- **Evaluate alternatives** as they emerge, **call for the need of tests** whenever interesting, and recommend those shown to be **interoperable, scalable and supportable.** **
- Ensure that these alternatives can be supported by **FTS and lcg_utils** to allow interoperability **

•(I prefer not to cover archive sites
– see later slides)

** Text as in TEG Report

Mandate

As presented to MB:

- The WG should **not** attempt to design a replacement interface to SRM
- The WG should bring together and coordinate existing and emerging activities
 - such as the support of “gridFTP-only” sites

Added by me:

- Focus initially on non-archive use

Composition of WG

Should be a **small** WG:

Not excluding anyone but need to remain focussed pragmatic – will report to GDB for wider discussion

- Experiment representation:

- Esp. those with experience in “non-SRM” storage usage
- Simone Campana for ATLAS
- Brian Bockelman for CMS
- Philippe Charpentier for LHCb

- Site representation:

- especially those with non SRM storage
 - OSG sites (Brian), CERN (Dirk) ... Also me ..

- Middleware and Tool Developers:

- FTS etc. Markus
- Xrootd (link to other WG) (Brian) , WebDav (Paul Millar),

Timeline

○ Start:

- 1st full meeting of WG in Sept. (next week hopefully)
- Material in this talk:
 - Review of activities on ATLAS / CMS from 1-1 discussions
 - Distillation of tables and other material from TEG report
- Oct pre-GDB – f2f discussion with other storage WGs,

○ Duration (as presented by Markus to MB):

- Since the WG monitors and coordinates emerging activities it isn't a short term WG
- Report every 3 months on the progress of monitored activities and suggest additional evaluations and tests
- After 12 months propose a roadmap

Table of used functions from TEG report

	<i>Is this feature used by ...</i>				Tier	SRM function ²
	<i>Atlas</i>	<i>CMS</i>	<i>LHCb</i>	<i>FTS only</i>		
Transfer Management						
Upload / download a complete file	Yes	Yes	Yes	No	All	srmPrepareToPut/Get/Put/GetDone
Manage transfers.	Yes	Yes	Yes	Yes	T1/2	srmAbort/Suspend/ResumeRequest
Balance over multiple transfer servers.	Yes	Yes	Yes	Yes	T1/2	srmPrepareToGet ³
Manage third-party copy	Yes	Yes	Yes	Yes ^b	T1/2	
Negotiating a transport protocol	No	No	No			srmGetTransferProtocols
Namespace Interaction						
Querying information about a file (stat)	No	No	Yes ¹	Yes ^b	T1/2	srmLs
Upload data integrity information (chksums)	No	No	No	No	T1/2	
Check integrity information	Yes	Yes	Yes	Yes		srmLs
Creating/Deleting data and directories	Yes	Yes	Yes ¹	Yes ⁷	All	srmMkdir srmRmdir srmRm srmMv
Changing ownership, perms and ACLs	No	No	No	No	-	srmSet/Check/GetPermission
Storage Capacity Management						
Query used capacity (like df)	Yes	No	Yes	No	T1/2	srmGetSpaceMetaData/Tokens
Create/remove reservations; assign characteristics	No	No	No	No	-	srmReserve/Update/ReleaseSpace
Targeting uploads to specific reservation	Yes	Yes	Yes	No	T1/2	srmPrepareToPut
Moving files between reservations	No	No	Yes	No	T1/2	srmChangeSpaceForFiles
Server Identification						
Test service availability and information	Yes	Yes	No	No		srmPing

- Somewhat simplified and removed those only relevant for Archive/T1
- Still probably can't read it (!) but an observation:
 - Not that much is needed – e.g. space management is only querying and not even that for CMS

Even shorter table of SRM used functions for ATLAS

And listing where there are alternatives demonstrated

Function	Alt.?	Issue	Notes
Transfer: 3 rd Party (FTS)	✓	✗	using gridFTP only in EOS
Transfer: Job in/out (lan protocol)	✓	✗	using xrootd only in EOS
Transfer: Download (DQ2-Get)	✗	✓	Currently use lcg-cp (alt plugins in rucio - dev)
Namespace: Creating Directories/Deletion	✗	✓	Deletion would need plugin for an alternative.. Dev work
Namespace: Checking Checksums	?	?	?
Space Management: Query Use	✗	✓✓	Need something
Space Management: Target Upload	✗	✓	Need something (could use namespace)
Service: Availability and Info	✗	✗	Probably Not needed

Other ATLAS perspectives

○ Cloud Storage

- Some work but early stages

○ Xrootd – federation

- Use expanding but not likely to replace FTS placement
- There is another WLCG WG on this: liaison important

○ FTS3

- Prototype exists that supports gridFTP session reuse.
- Xrootd and http are expected in next prototype. Will test.

○ Possible ATLAS perspective on (issues for) the WG

- See advantages in SRM alternatives (e.g. reducing complexity; local access performance; dq2-get overheads)
- Need to ensure interoperability – not a proliferation of interfaces
- Ensure lcg-cp etc. support interfaces (make transition easier)
- Need alternative for space used querying.
 - Namespace on top-level dirs under /atlas/ good enough?
- Manpower on ATLAS development side needs to be considered

CMS table

Function	Alt.?	Issue	Notes
Transfer: 3 rd Party (F'TS)	✓	X	gridFTP only in Nebraska (interested in xrootd or HTTP)
Transfer: Job in/out (lan protocol)	✓	X	xrootd only in Nebraska
Transfer: Download (DQ2-Get)	X	X	Not really used on CMS
Namespace: Creating Directories/Deletion	X	X	Deletion or query not needed on CMS, site approves transfers.
Space Management: Query Use	X	X	
Space Management: Target Upload	X	X	STs not required on CMS
Service: Availability and Info	X	X	Probably Not needed

CMS perspectives

- ◉ **Experience with using site with gridftp / xrootd**
 - IP Load balancing for gridftp - one giant endpoint.
 - Integrating with FTS and phedex
 - Some other advantages over srm - can take failing servers out easily - one host cert .
 - Interested in trying xrootd with FTS.
 - Need work on xrootd for 3rd party transfers - will be following that.
 - Interested in 3rd party transfers over HTTP.
- ◉ **Asking all sites to provide xrootd interface by end of year.**
- ◉ **Don't have same issues / need for space tokens monitoring or direct download:**
 - All transfers approved by the site. So they don't overflow...
 - No real direct download (lcg-cp); move output data to site then access it
 - Spacetokens - only for sites benefit
- ◉ **No real blockers** on incorporating non-SRM non-archive sites that provide gridFTP and xrootd interfaces

Conclusions

- For non-archive sites there is an opportunity for simplification of the storage management interface.
- There is a need for some oversight to evaluate/incorporate any transitions:
 - For ATLAS at least, some coordination and development needed on issues of space querying; deletion; upload to space tokens.
 - Overseeing that FTS and lcg-utils support different interface in same way (e.g. error codes) will ease transition.