

SamGrid Integration of SRMs

Robert D. Kennedy
and Andrew Baranovski
(Fermilab Computing Division)

for the Fermilab SamGrid project

CHEP04, Interlaken, Switzerland
27 September 2004
v1.1

Big Picture

- ◆ SamGrid is the Fermilab Run II Data Handling framework
- ◆ In production use at D0 since 1999, SamGrid will soon be in production at CDF, prototype-level testing at MINOS.

- ◆ SamGrid has been extended to the Grid. At CHEP04:
 - ➔ Job and Information Management – Talk 038
 - ➔ Monitoring and Information Services – Poster 451
 - ➔ Meta-data Services – Talk 500
 - ➔ Integration of SRMs – Talk 460 (this talk)
- ◆ Major “non-Grid” developments too – Talk 462 (db server), Posters 468 and 113 (CDF deployment, testing), ...

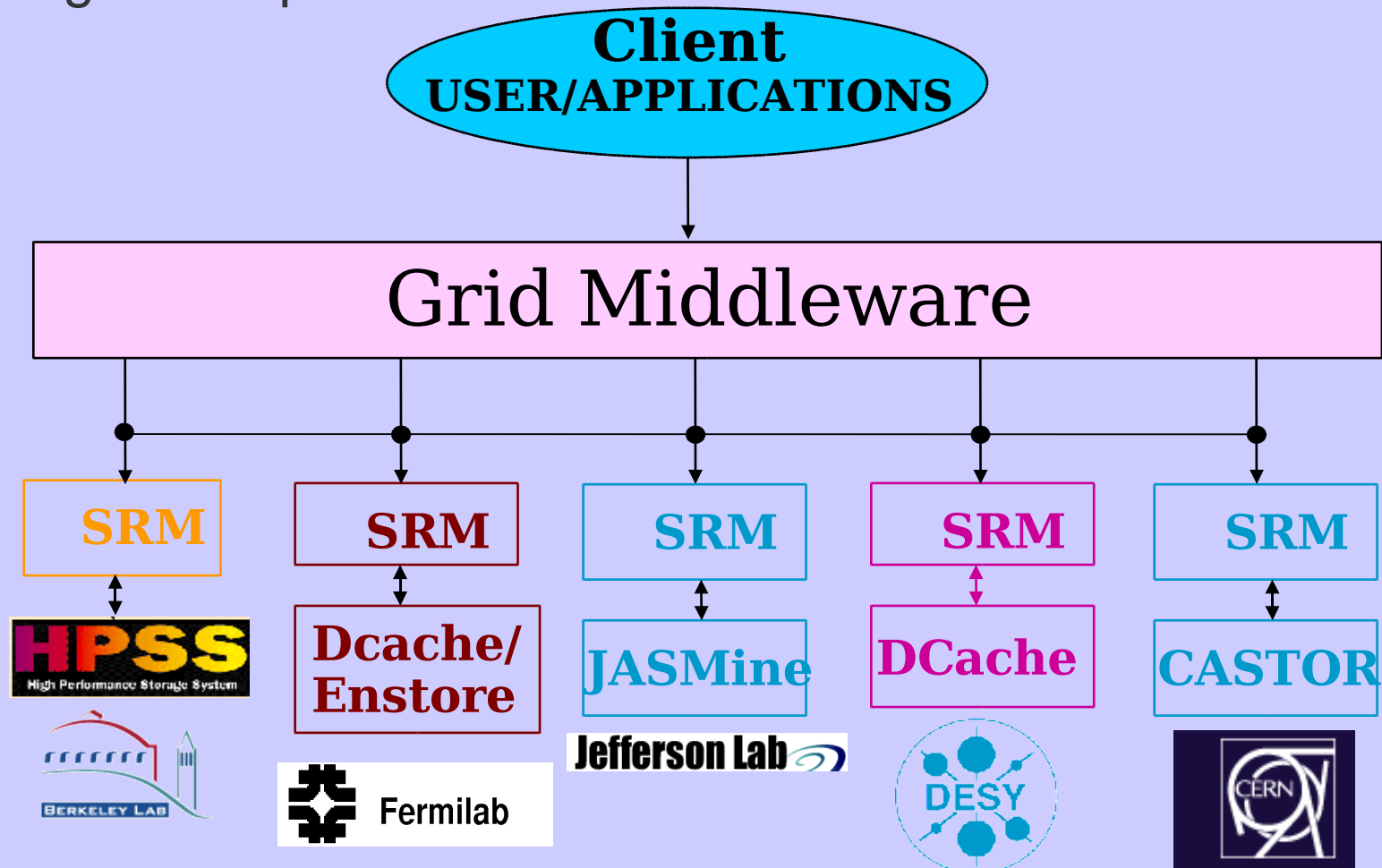
- ◆ A large team of contributors to SamGrid to acknowledge...

Outline

- ◆ Introduction to SRM – Storage Resource Manager
- ◆ Introduction to SamGrid – terms, issues
- ◆ Integration Goals – uniform storage, abstract locations
- ◆ Initial Adaptation – first part of multi-stage process to insure stable operations while integrating new functionality
- ◆ Status and Future Work, Application example
- ◆ Summary

Storage Resource Manager

- ◆ Uniform Grid interface to heterogeneous storage
- ◆ Negotiates protocols
- ◆ Interface specification v1 is in field, v2.1 is latest
- ◆ FNAL dCache SRM Imple.



SRM Functions, Parameterization

- ◆ Abstracts basic file system operations
 - Space Management functions – `srmReserveSpace()`
 - Data Transfer functions – `srmCopy()`
 - Directory functions – `srmMkdir()`
 - Permission functions – `srmSetPermission()`
 - Status functions – `srmStatusOfCopyRequest()`

- ◆ Fermilab dCache SRM – have parameterized the internal storage device interface to allow re-use of the “Grid layer”. Greatly simplifies delivering SRMs for other storage devices. Unix file system example exists, others in the works.

- ◆ CHEP04 talk 107 “Storage Resource Manager”

SAM Introduction

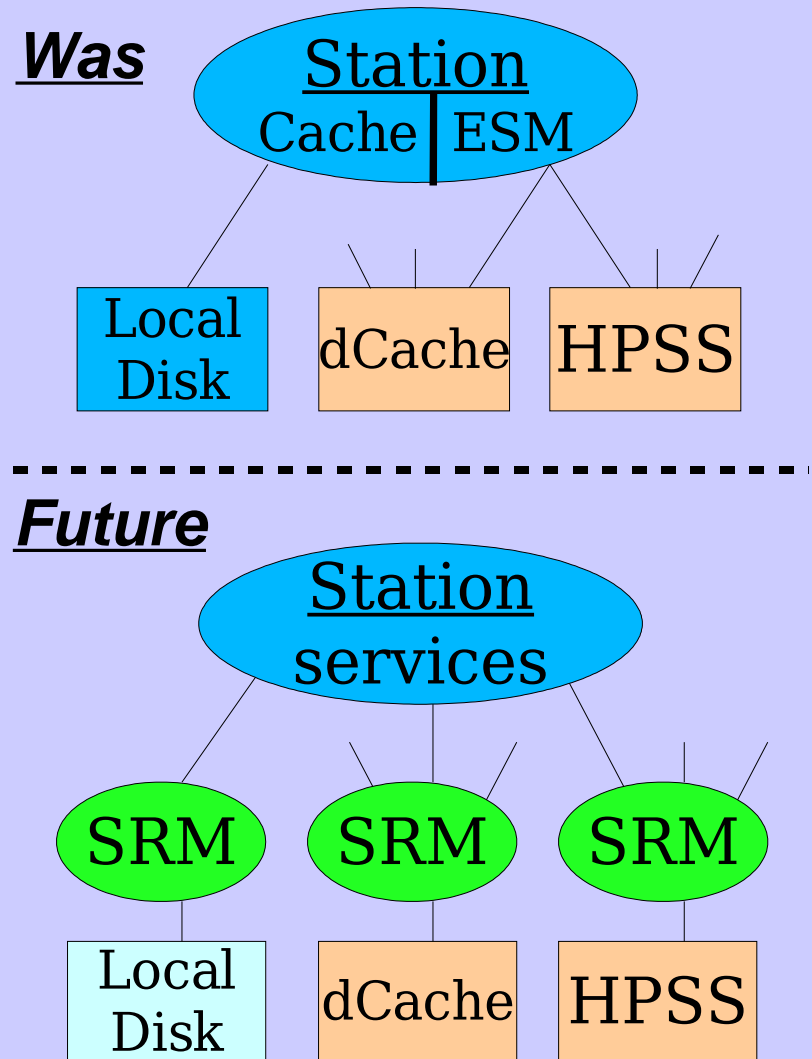
- ◆ SAM started as a DataGrid, pre-dates modern Grid design
- ◆ **Station** – resources managed together. DataGrid “node”.
- ◆ **Project** – manages file delivery for 1+ **consumers** (apps)
- ◆ **Dataset Definition** – meta-data spec. of desired data files
- ◆ **Snapshot** – the actual list of files specified by dataset
- ◆ **Stager** – agent performing file transfers to local disk cache
- ◆ Cache – a quota-enable disk cache (specific to SAM)
- ◆ Replica DB – catalog of file replica **locations**
- ◆ Project runs on a station, requests delivery of a dataset snapshot to a disk cache accessible to the interested consumers. Stagers arrange the file transfers to that cache.
- ◆ CHEP00 talk C241, “...Design and Features of SAM”

SAM Issues

- ◆ SAM works. With experience, some issues recognized:
- ◆ **Diversification of “cache” options**
 - Originally, files consumed from local disk cache. Other storage valid for locations, files copied to disk cache.
 - Interest in alternatives – NFS, HPSS, dCache, AFS,
 - “External Storage Mechanism” developed, works, but...
 - Code and configuration specific to each storage system.
- ◆ **(Replica) Location abstraction**
 - Locations of files in disk cache contain the node name and pathname of the file in cache. Not robust to equipment failures, directory re-organization, etc.
 - Can be more than one location per physical file (dCache dcap doors) for same protocol. Can be confusing.

Uniform Storage Interfaces

- ◆ **Uniform storage interface** – eliminate specialized code
- ◆ **Centralized configuration** of shared storage services
- ◆ **Constraints:** (some of many)
 - maintain production system
 - must support other dev too
 - old and new stations must be able to work together for smooth deployment
- ◆ **Challenge:** Indistinct interfaces in station components

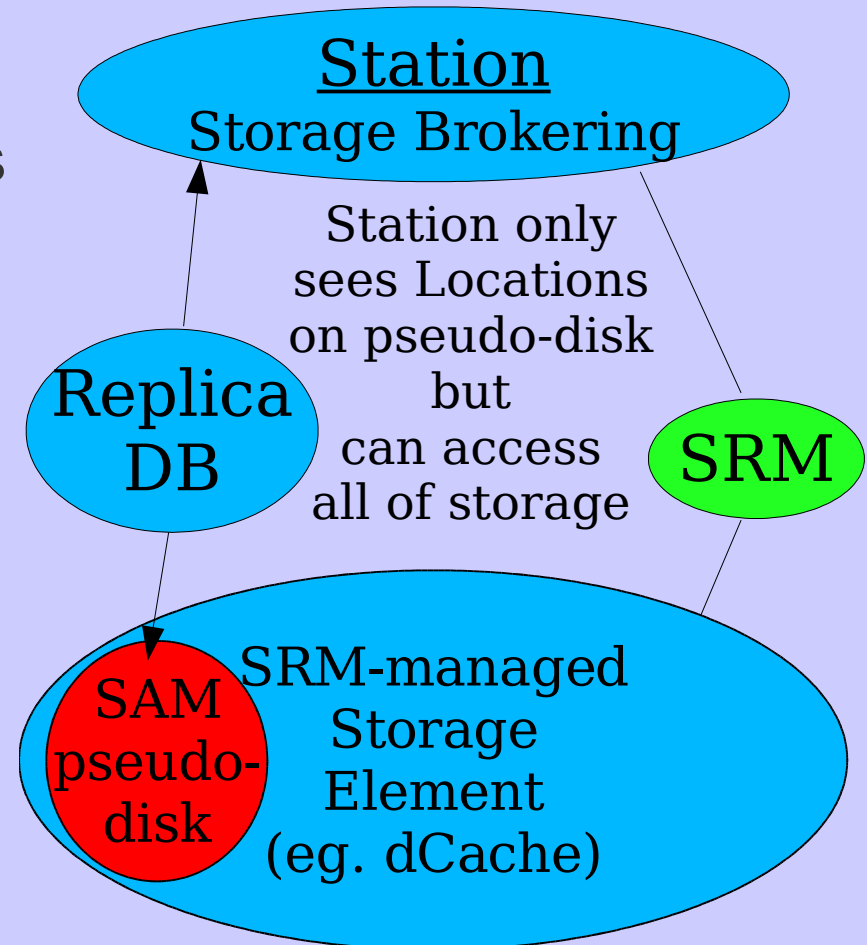


Abstract Locations

- ◆ **Abstract Locations** – avoid direct hardware references. SAM caches run on desktop and CPU farm IDE disks too.
- ◆ **Unique Locations per physical file** – avoid confusion
- ◆ **Constraints:** (some of many)
 - ➔ support old and new locations for smooth deployment
 - ➔ subtleties in data management decisions and locations
- ◆ **Old-style** – `cachehost.fnal.gov/hardware-directory`
- ◆ **New** – `srm://srmhost.fnal.gov:8843/namespace-filepath`
- ◆ **TURL** – `protocol://storagehost.fnal.gov/transfer-path.dat`
 - ➔ New: can change physical location inside SRM and still be accessibility to consumers. NFS mounts, disk fails.
 - ➔ SRM provides location “indirection” in a sense.

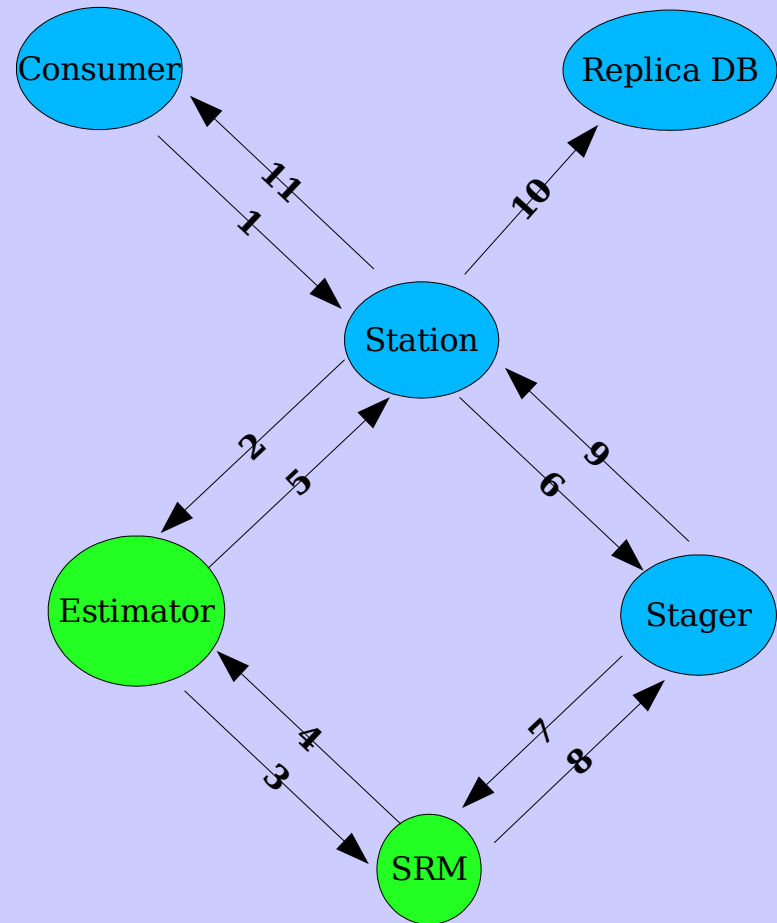
Initial SRM Integration - Overview

- ◆ Initial integration stage – re-interpret, re-use existing functionality, prove concepts
- ◆ Configure SRM managed storage elements as if they were station consumption nodes with a disk per access method of some “size”.
- ◆ Pseudo-station disks, nodes describe physical cache element accessible by consumers with common data access requirements.



Initial SRM Integration - Sequence

- 1) start project
- 2) get priority
- 3) srm get meta-data
- 4) return meta-data
- 5) return priority
- 6) request transfer
- 7) srm get/copy
- 8) srm get/copy done
- 9) transfer request complete
- 10) register new SRM replica
- 11) file is available



Status and Future Work

◆ Project Status

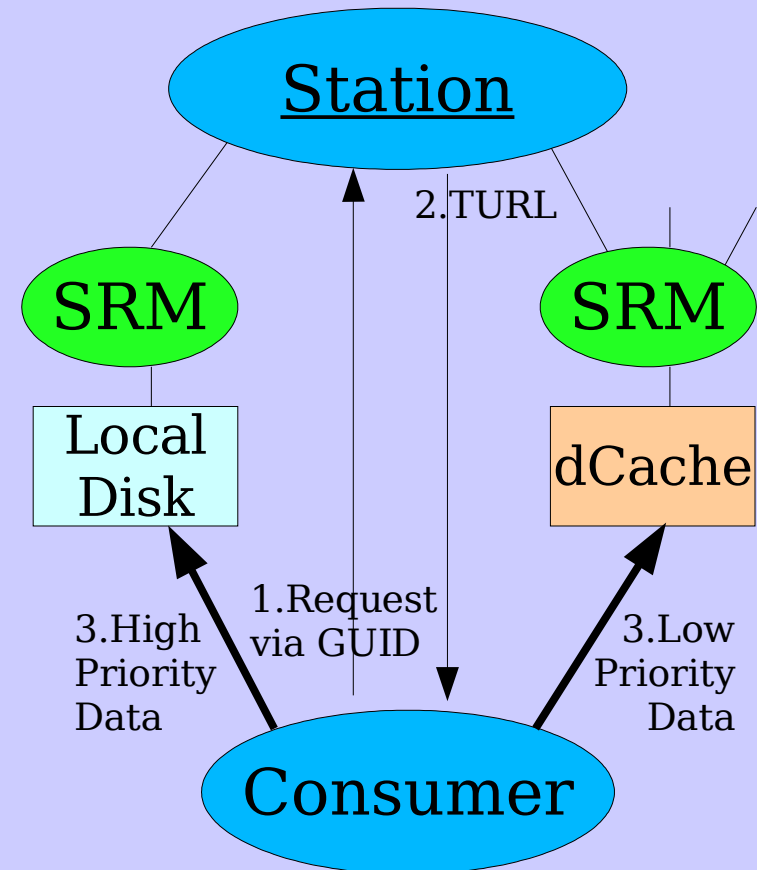
- prototype working now with dCache gridftp transfers
- ready to try other storage, transfer protocols (dcap)
- small mismatch in SAM, SRM error handling approach

◆ Future Work

- modularize station components into distinct services
- re-implement SAM cache with SRM interface
- fully adopt modern GRID GUID/SURL/TURL model, integrating transfer protocol into data mgmt decisions.
- move to symmetric data stage and store services
- lower priority – full adaptation to web services. SAM's CORBA infrastructure works well enough for now.

Application – Multi-tier Caching

- ◆ User desktop analysis – optimize access to high priority data by storing locally, but read low priority data from large, central dCache.
- ◆ Station routing configurable by dataset, SRM locations
- ◆ High priority data copied to local disk, then that TURL given to the consumer
- ◆ Low priority data left in dCache, and TURL given.



Summary

- ◆ SamGrid in the process of integrating the use of SRMs as a uniform storage interface to numerous storage systems
- ◆ Project is ambitious – must consider that SamGrid is:
 - 24 x 7 x 365 production system
 - used by running experiments for data logging
 - undergoing important (unrelated) parallel development
 - already has code specific to different storage devices whose idiosyncrasies must be taken into account.
- ◆ Initial integration tests successful, working to expand this to other protocols, storage systems. Onward, onward 8^).