Global Data Access Architecture

Brian Bockelman

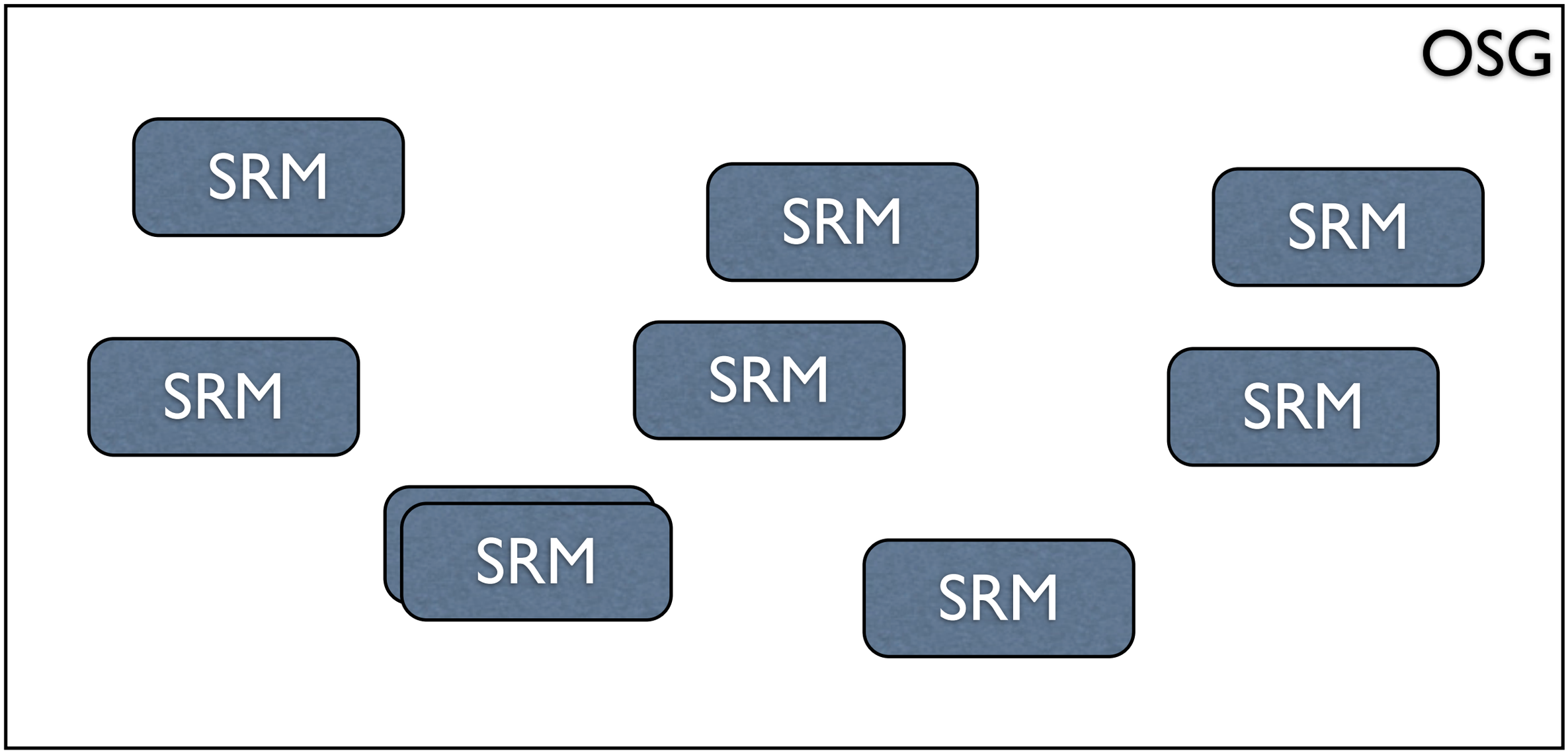
OSG Storage Management

- Today, OSG sites provide storage management via SRM:
  - Space allocation and management (extent varies between sites and software).
  - Namespace metadata operations.
  - File movement.
- All exposed via a a webservices endpoint.

The vision of today

- Each site is operated as an independent "depot" of data.
- You, the user/VO have to explicitly manage and track your data in order to effectively use it. That is, you must write software to manage:
  - Initiating transfers, checking validity, interfacing with job submission, location lookup, manage usage, and more.

OSG Storage Mgmt



It is the VO's job to implement software to "wrangle" all these endpoints into a coherent system.

Data Management

- The sum of all these parts is your data management layer.
  - These are costly to implement.
    - Effectively, only LHC and LIGO can afford to write their own elaborate systems (for the first round of funding, at least!).
    - Lots of R&D to build "generic data management solutions" - which often become quite customize when put into place.

Let's examine the options!

DM with iRODS

- There is a new project, iRODS, which has evolved a rules-based language for expressing policies.
  - Extremely flexible, extremely scalable in terms of number of files kept. Great client tools.
- How one would integrate iRODS with OSG is unclear, but looks very appealing.
- I'll defer to later presentations on this topic.

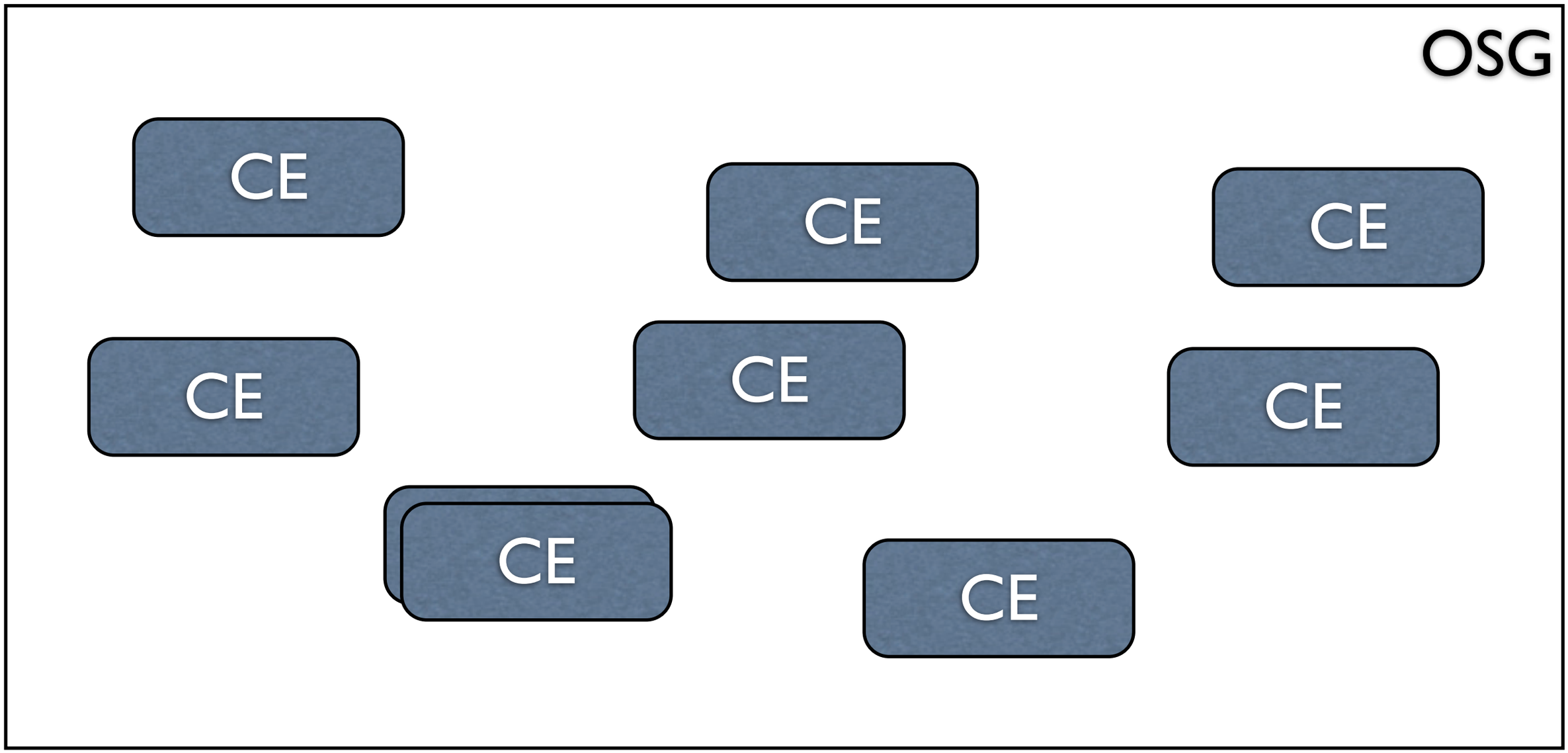
Evaluation: Very promising, but...
interfacing iRODS with OSG is embryonic.

Poor Man's Data Mgmt

- The "trivial data management" solution:
  - Keep all input files at one site.
  - Write all output files at the same site.
  - Assume decent network bandwidth and scalable site.
- This can be done with Condor File Transfer or SRM.
- Done today by several VOs. Very easy.
- Assumes the VO owns at least one "beefy" storage site.
- Scales up to around 100MB input/output per job.

Evaluation: Simple, cheap, reliable. "Doable" today
Limited in terms of scalability and distributed.

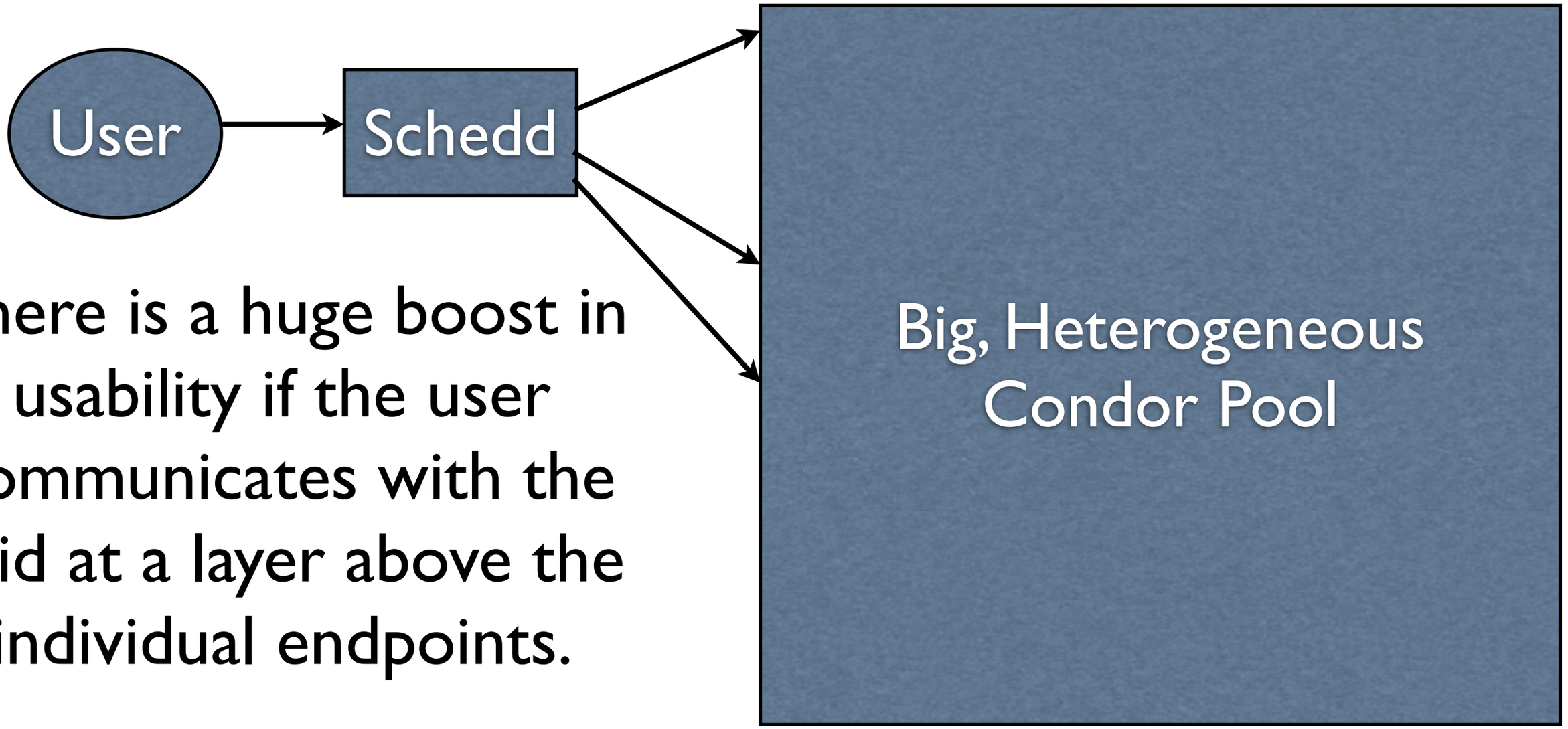
Compare to Jobs...



This is how OSG computing looks to a new VO.

Compare to Jobs...

- This is the user view of the world using glideins, which have received much enthusiasm...



There is a huge boost in usability if the user communicates with the grid at a layer above the individual endpoints.

A Global Data Access Architecture

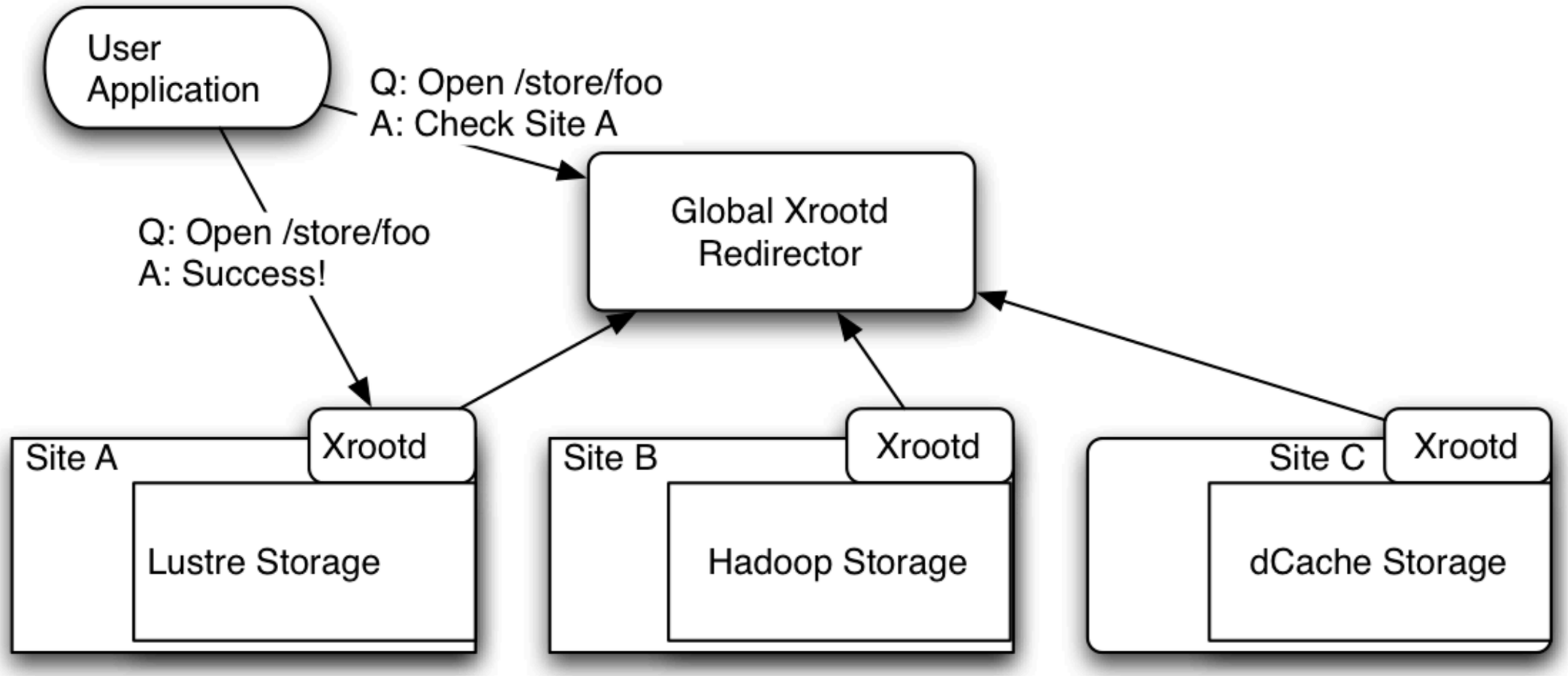
- Hopefully, this motivates the need for a "data management" layer provided by the OSG above the current storage layer.

- For the rest of this talk, I'll be talking about a global data access architecture I'm putting together for CMS.
  - And how it might apply to the OSG as a whole.

CMS Xrootd Demonstrator

- We are working putting together an xrootd-based testbed that will expose a read-only interface to the CMS data systems.
- Based on a tree-like hierarchy of redirectors.
  - This makes the entire CMS global namespace available if you contact the top of the tree.

Federating Sites



Each site exports the global namespace, and translates the file open requests to the local namespace.

Elapsed time is often around 100ms.

Demo

- (Network permitting)

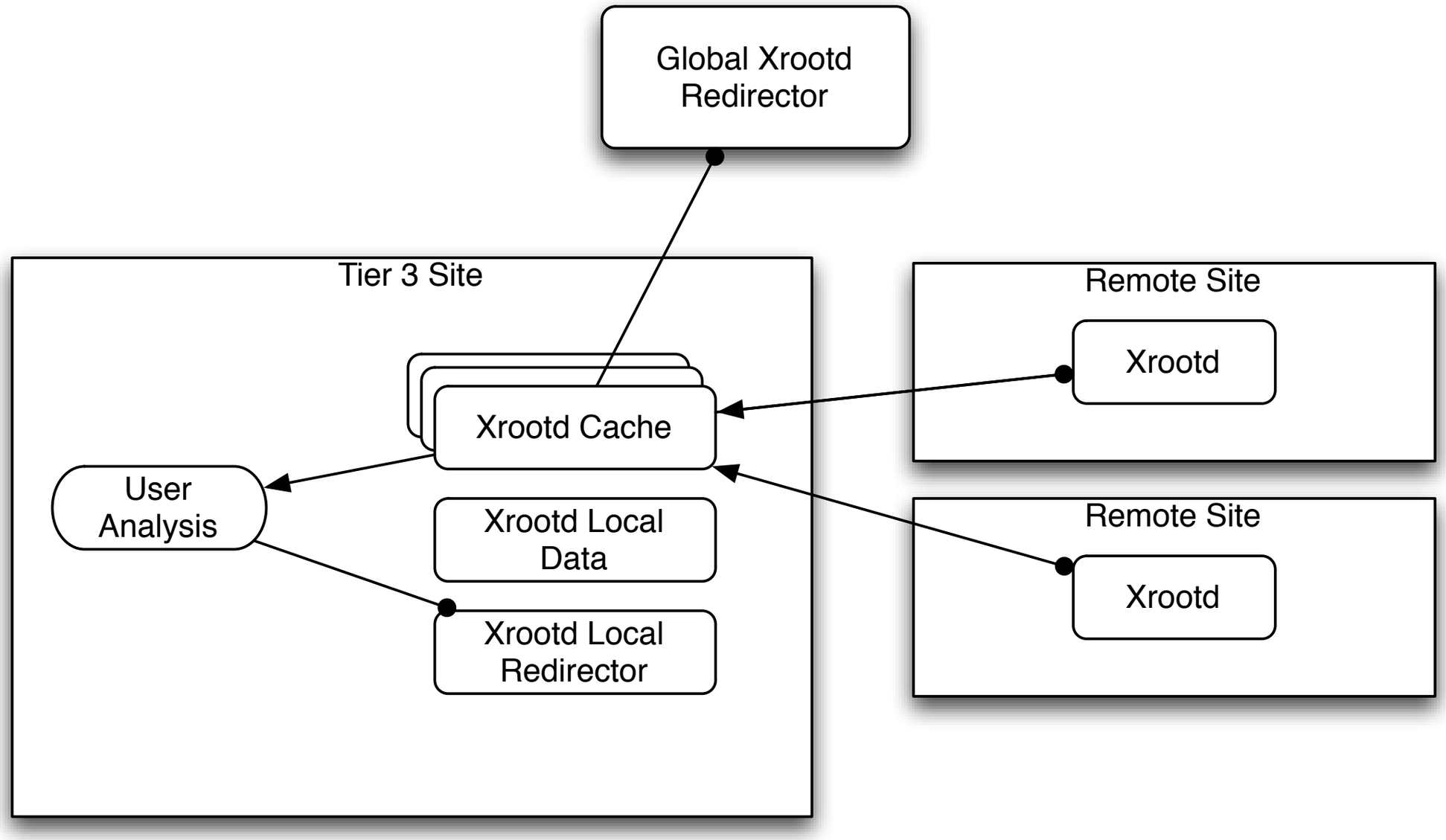
Federating Xrootd

- All data is accessed via a single global namespace (the CMS namespace).
  - No need to know location info.
- The system performs site selection.
  - Or you can use the bittorrent-like mode and download from all sites - this auto-tunes to select the best server.

Caching

- Xrootd can additionally act as a cache and bring the complete file locally.
- In this case, the client will talk to a local redirector which will decide whether the file is local and download it from the global federation if not.
  - Once cached locally, the cache can be reused (both by local users and in the global architecture)!

Caching Architecture



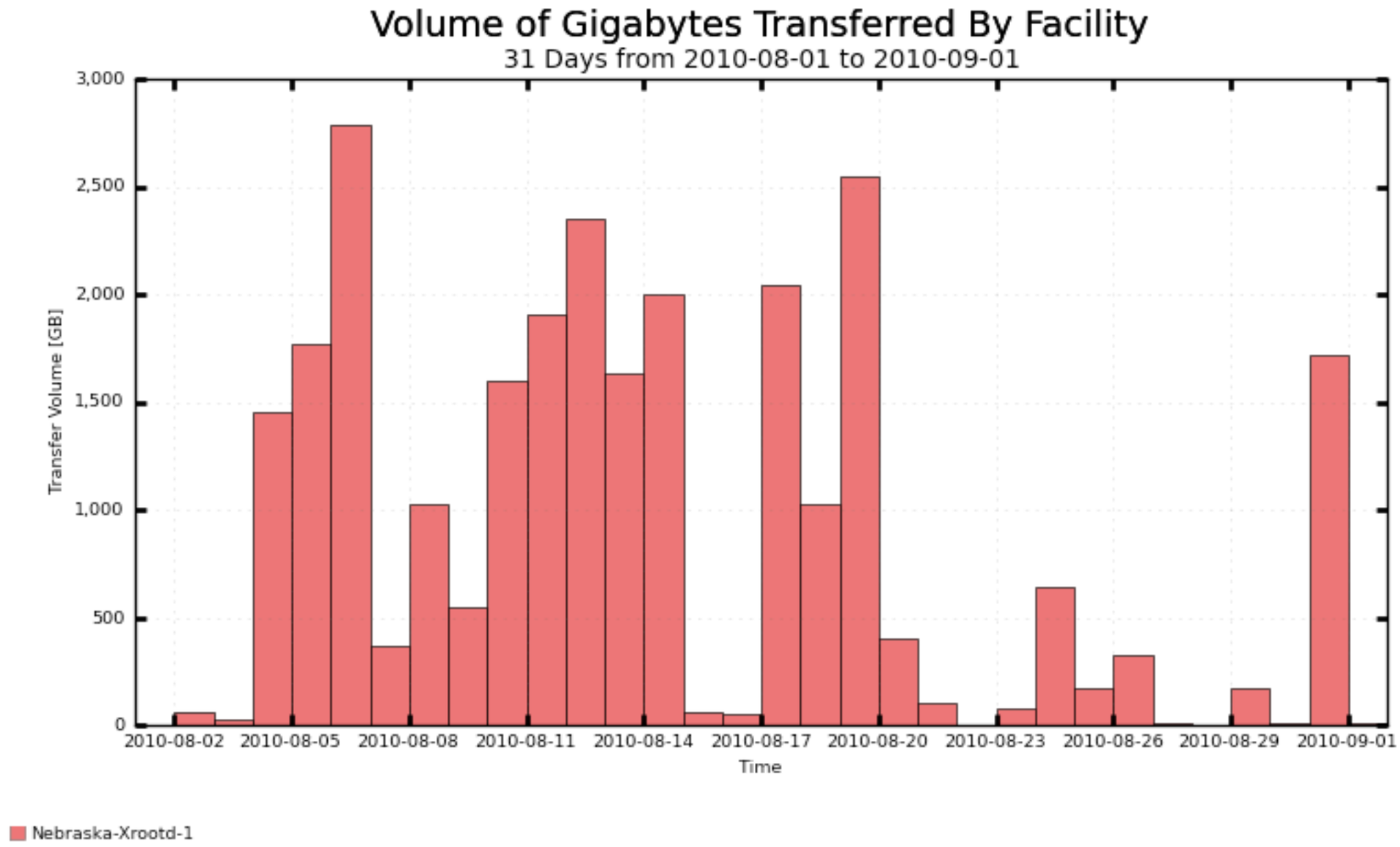
Caching Downloads

- The caching architecture can be combined with the bittorrent mode of xrdcp to optimize the performance of downloads.
- Errors are only propagated if all sources error out.

Monitoring Xrootd

- Xrootd has built-in monitoring. It sends out summary UDP packets for all transfer activities.
  - Sent to MonALISA for realtime monitoring.
  - Sent to Gratia for long-term accounting.

Monitoring Xrootd



Issues

- Namespace consistency is assumed.
- Unsure about data integrity issues.
- Authorization issues when redirecting.
- Does not solve data archival / metadata issues.
- Caching approaches have drawbacks thoroughly discussed by computer scientists.

Xrootd on the OSG?

- It would be possible to reuse the xrootd global architecture for OSG.
  - This project shows its ability to run “in production”
- I believe this provides a clever solution to the global namespace and data management problem.

Extensions Needed

- There are a few extensions needed before this can be useful to the OSG in-general:
  - Generic handling of VO namespaces.
    - Current "translation layer" implemented at sites.
  - Tackling the "write problem".
  - Interfacing with job submission and monitoring systems.

For More Info...

- I try to keep everything documented on the CERN twiki:
  - https://twiki.cern.ch/twiki/bin/view/Main/CmsXrootdArchitecture