Scalla As a Full-Fledged LHC Grid SE

Wei Yang, SLAC
Andrew Hanushevsky, SLAC
Alex Sims, LBNL
Fabrizio Furano, CERN

SLAC National Accelerator Laboratory
Stanford University

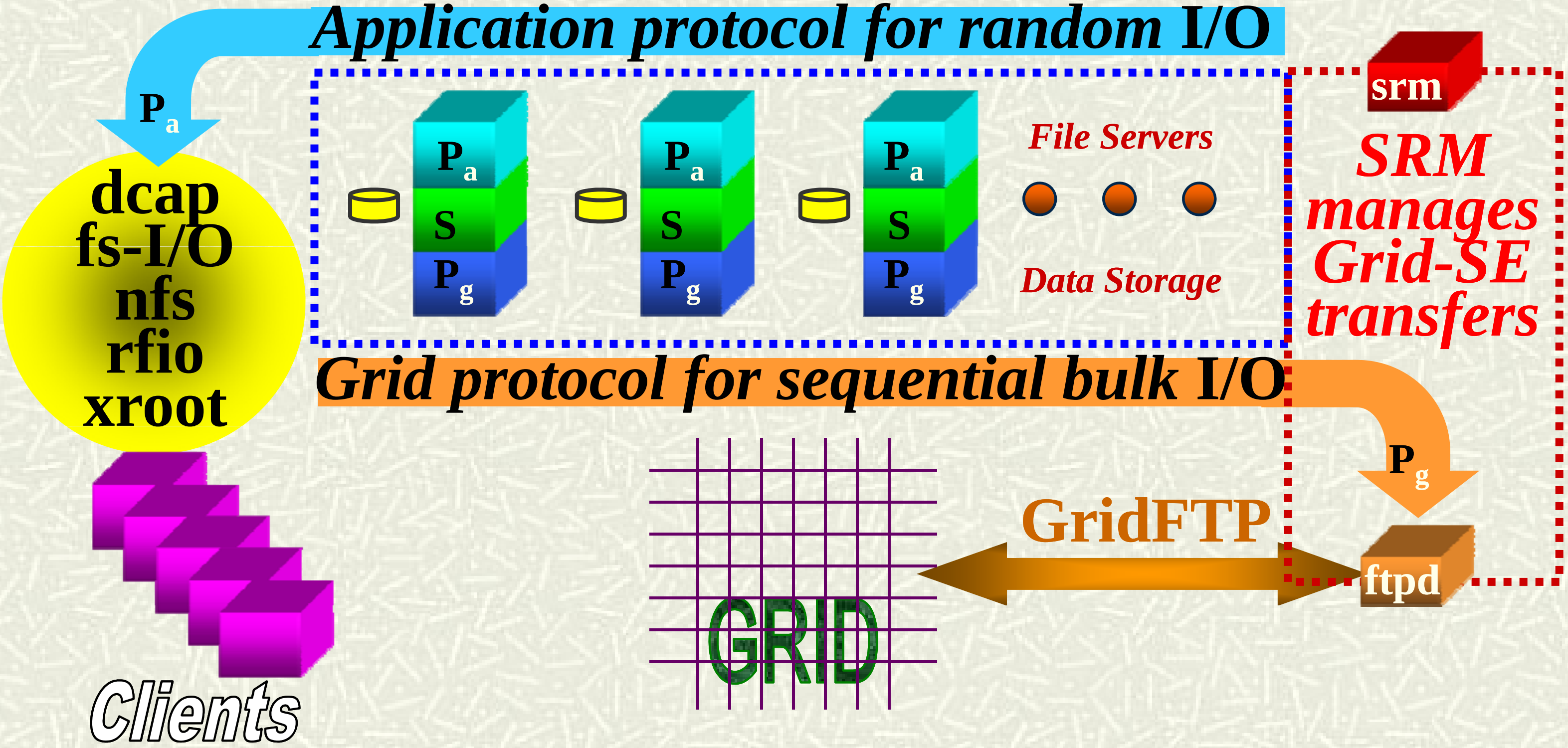
24-March-09

CHEP

Outline

- The canonical Storage Element
- Scalla/xrootd integration with SE components
  - **GridFTP**
    - Cluster I/O
  - BeStMan SRM
    - Name space issues
    - Static Space Tokens
- Conclusions
- Future Directions
- Acknowledgements

The Canonical Storage Element



Bottom Up View

Distinguishing SE Components

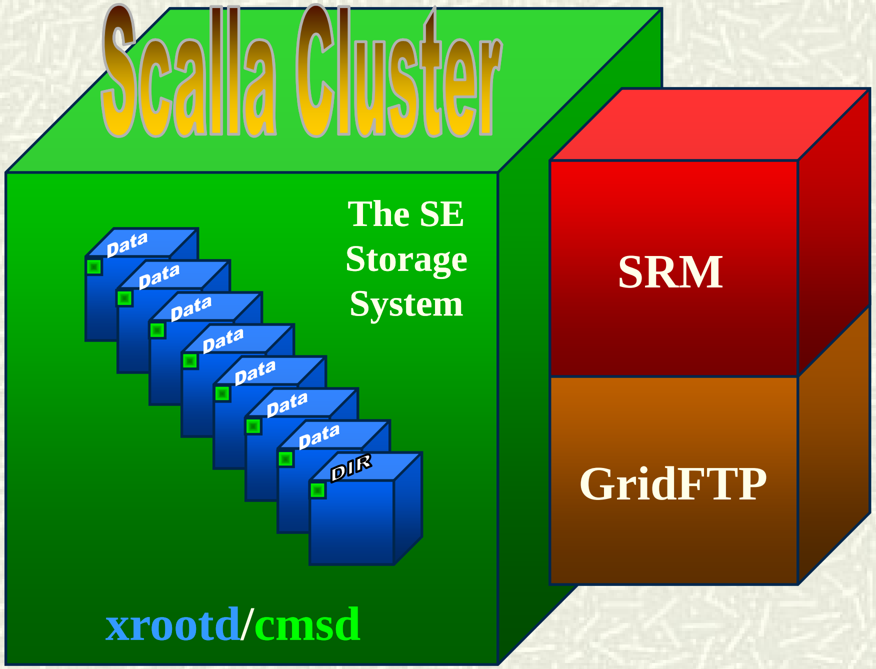
- **SRM** (Storage Resource Manager v2+)
  - Only two *independent** version available
    - Storage Resource Manager (StoRM)
      - http://storm.forge.cnaf.infn.it/
    - Berkeley Storage Manager (BeStMan)
      - http://datagrid.lbl.gov/bestman/
    - Both are Java based and implement SRM v2.2
- **GridFTP**
  - Only one de facto version available
    - Globus GridFTP
      - http://www.globus.org/grid_software/data/gridftp.php

*Castor, dCache, DPM, Jasmine, L-Store, LBNL/DRM/HRM, and SRB SRM's are tightly integrated with the underlying system.

Which SRM?

- We went with BeStMan
  - LBNL developers practically next door
  - Needed integration assistance
  - Address file get/put performance issues
- LBNL team developed BeStMan-Gateway
  - Implementation of WLCG token specification
  - Stripped down SRM for increased throughput
    - Sustained performance ~ 7 gets/sec & ~ 5.6 puts/sec
    - Original BeStMan 1 ~ 1.5 gets/sec & 0.5 ~ 1 puts/sec
  - Perhaps the fastest SRM available today

The Integration Task

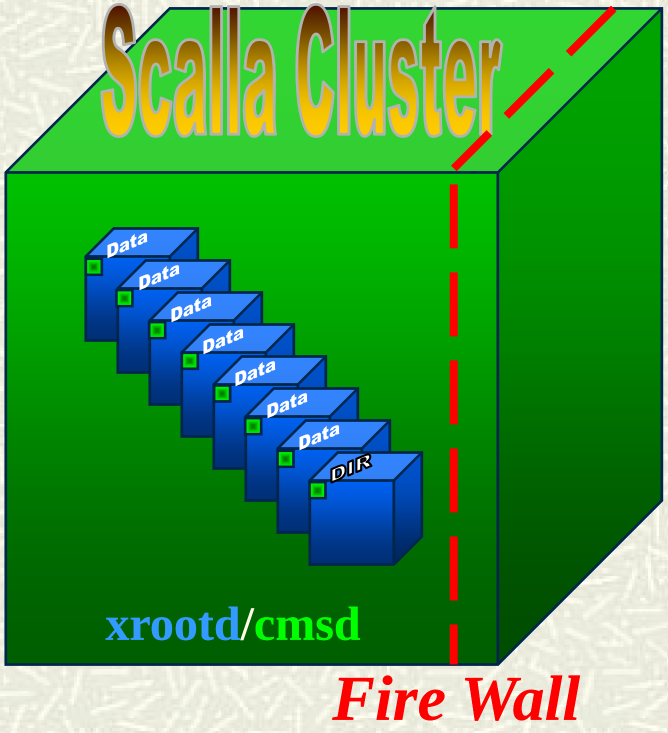


You might mistakenly think this is simple!

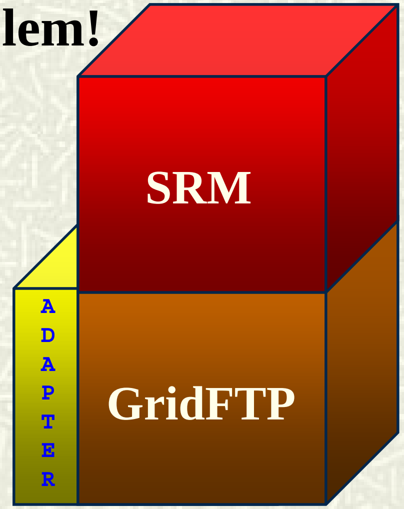
Integration Issues (why it's not simple)

- Scalla/xrootd is not inherently SRM friendly
  - SRM relies on a true file system view of the cluster
  - Scalla/xrootd was not designed to be a file system!
    - Architecture and meta-data is highly distributed
    - Performance & scalability trump full file system semantics
- The Issues . . .
  - **GridFTP** I/O access to the cluster
  - SRM's view of the cluster's name space
  - WLCG Static Space Tokens

Integration Phase I (GridFTP)

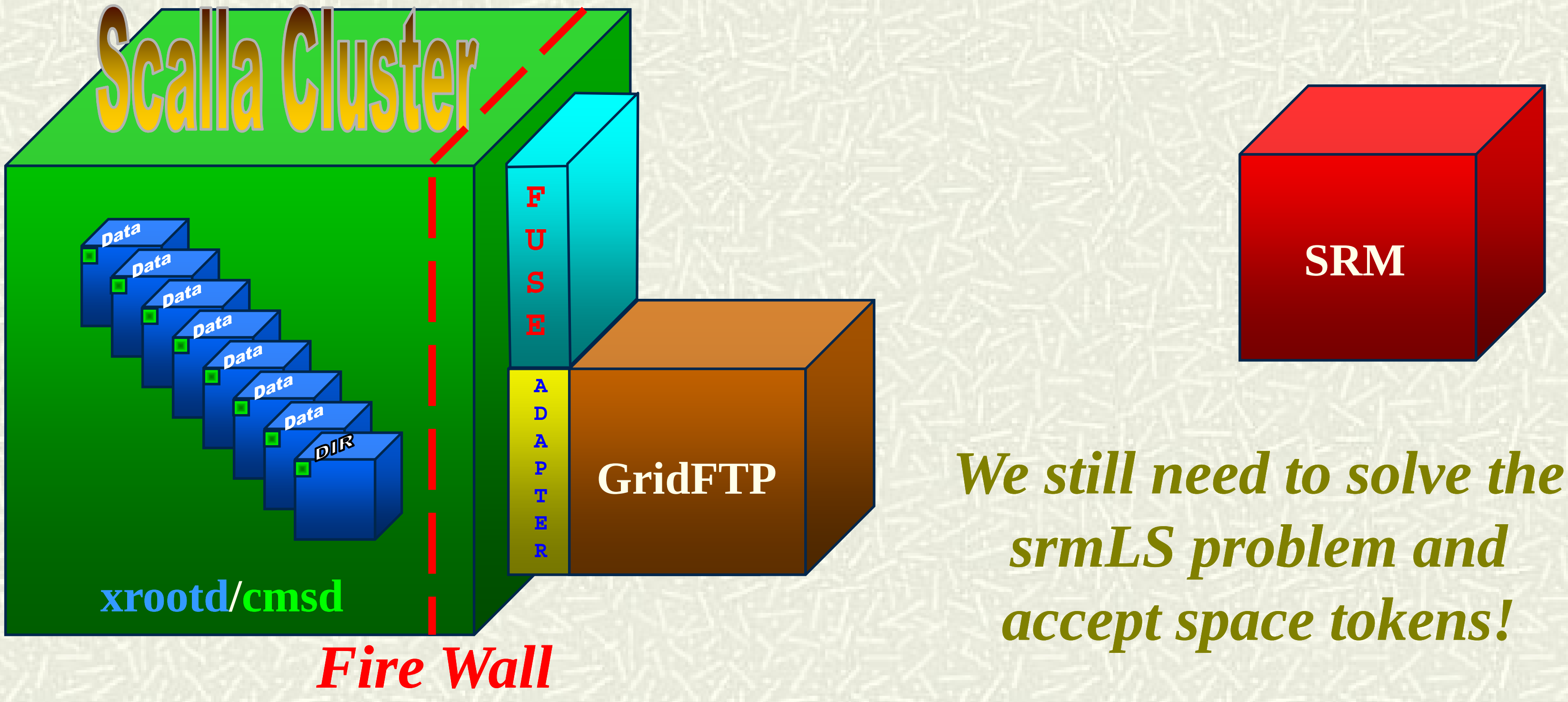


We still have an SRM problem!
Source adapters generally won’t work with Java.



Source Adapter: POSIX Preload Library for xrootd access
Provides full high-speed cluster access via POSIX calls
GridFTP positioning can be more secure!

Integration Phase II (BeStMan SRM)



We still need to solve the srmLS problem and accept space tokens!

Target Adapter: File System in User Space (FUSE)

Full POSIX file system based on XrdClient called xrootdFS

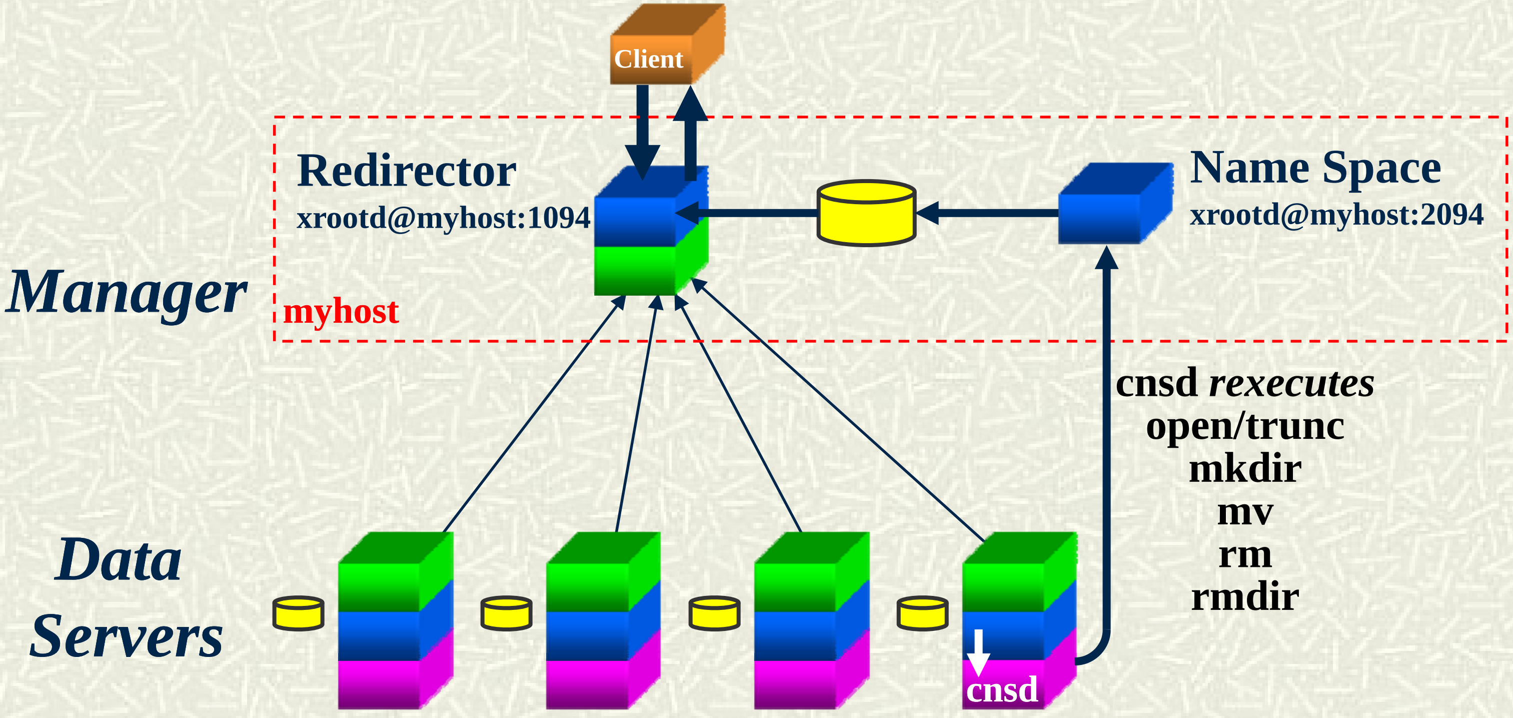
Interoperates with BeStMan **and probably StoRM**

The srmLS Problem & Solution

- The SRM needs full view of the *complete* name space
  - SRM simply assumes a central name space exists
  - Scalla/xrootd distributes the name space across *all* servers
    - There is no central name space whatsoever!
- Solution: create a “central” *shadow* name space
  - Shadow name space ≡ ∑ cluster name space
    - Uses existing xrootd mechanisms + cnsd daemons (i.e., no database)
- This satisfies srmLS requirements
  - Easily accessed via FUSE

The Composite Name Space (cnsd)

opendir() refers to the directory structure maintained by xrootd:2094 (*full* cluster name space)

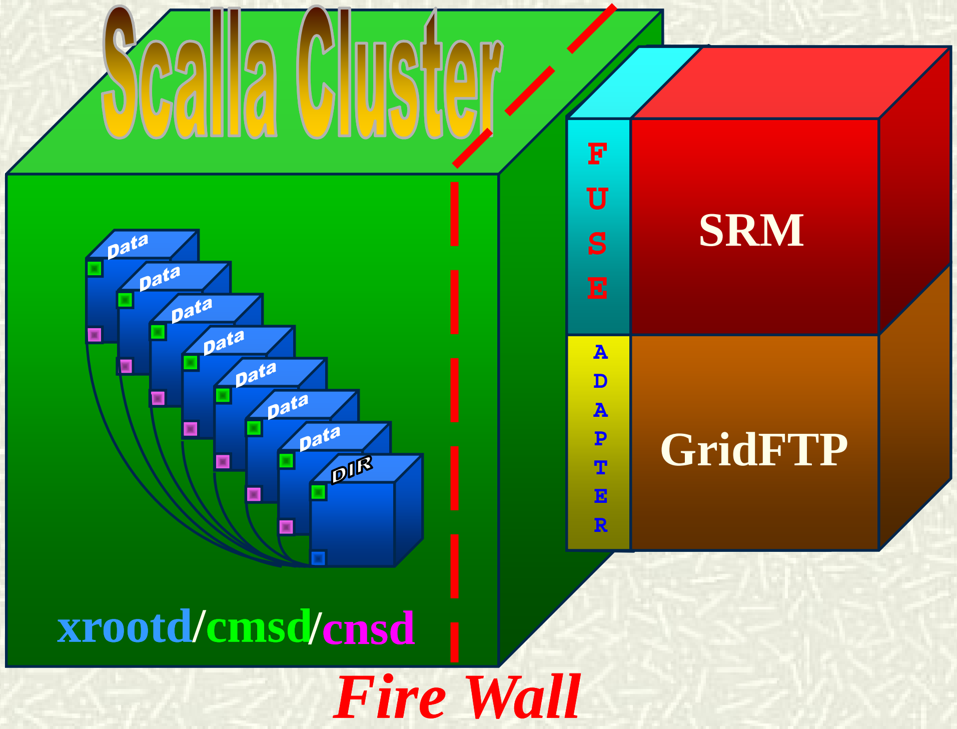


ofs.notify closew, create, mkdir, mv, rm, rmdir |/opt/xrootd/bin/cnsd

Composite Name Space Actions

- All name space actions sent to designated xrootd's
  - Local xrootd's use an external local process named cnsd
  - cnsd name space operations done in the background
    - Neither penalizes nor serializes the data server
- Designated xrootd's maintain composite name space
  - Typically, these run on the redirector nodes
- Distributed name space can now be concentrated
  - No external database needed
  - Small disk footprint
  - Well known locations for find complete name space

The 10,000 Meter View



A cnsd runs on each data server node communicating to an extra xrootd server running on the redirector node

SRM Static Space Tokens

- Encapsulate fixed space characteristics
  - Type of space
    - E.g., Permanence, performance, etc.
  - Implies a specific quota
- Using an arbitrary pre-defined name
  - E.g., atlasdatadisk, atlasmcdisk, atlasuserdisk, etc.
- Typically used to create new files
  - Think of it as a space profile
- Space tokens required by “some” LHC experiments
  - E.g. Atlas

Static Space Token (SST) Paradigm

- Static space tokens map well to disk partitions
  - A set of partitions define a set of space attributes
    - Performance, quota, etc.
  - Since an SST defines a set of space attributes
  - Then partitions and SST’s are interchangeable
- Why do we care?
  - Because partitions are natively supported by xrootd

SLAC
NATIONAL ACCELERATOR LABORATORY

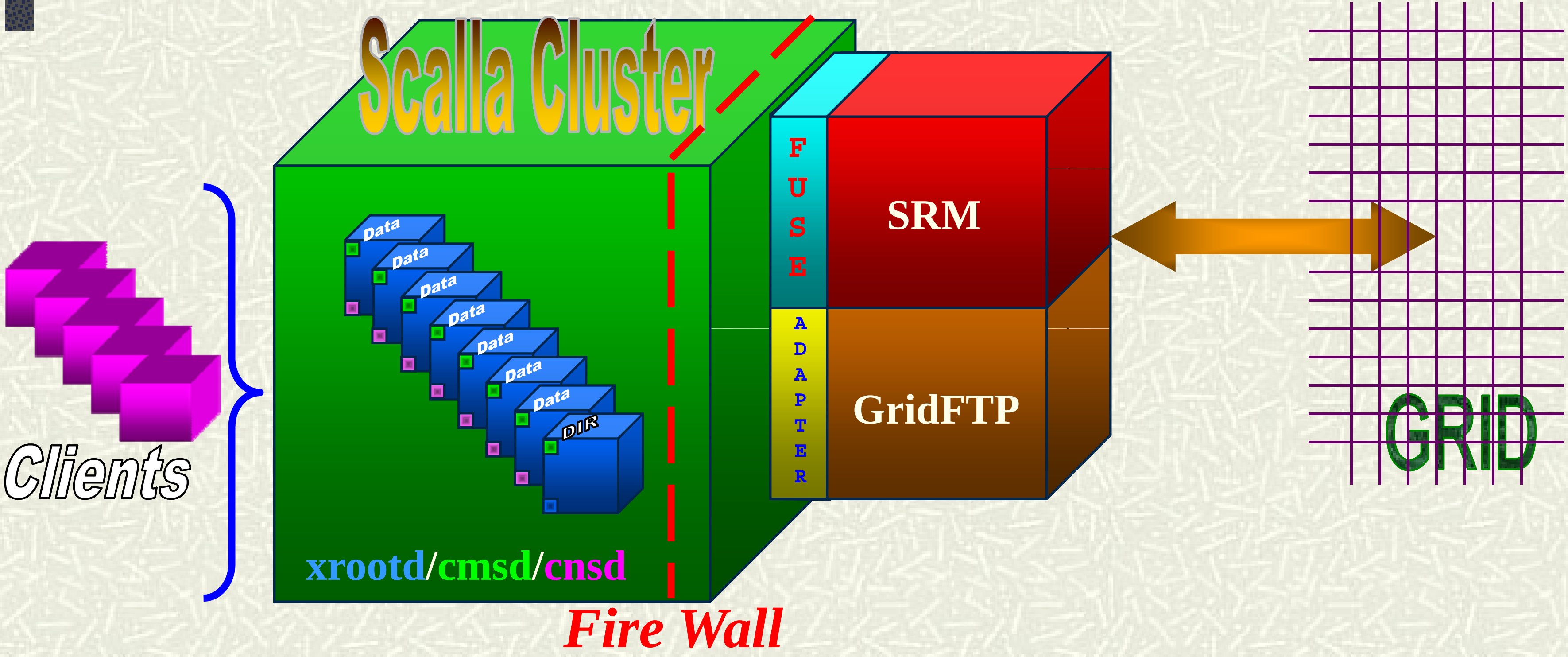
Supporting Static Space Tokens

- We leverage xrootd's built-in partition manager
  - Just map space tokens on a set of named partitions
    - xrootd supports real *and* virtual partitions
      - Automatically tracks usage by named partition
      - Allows for quota management (real → hard & virtual → soft quota)
- Since Partitions ⇔ SRM Space Tokens
  - Usage is also automatically tracked by space token
- getxattr() returns token & usage information
  - Available through FUSE and POSIX Preload Library
    - See Linux & MacOS man pages

Integration Recap

- GridFTP
  - Using POSIX preload library (source adapter)
- SRM (BeStMan)
  - Cluster access using FUSE (target adapter)
  - srmLS support
    - Using distributed cnsd's + central xrootd processes
  - Static space token support
    - Using the built-in xrootd partition manager

The Scalla/xrootd SE



But wait!

Can’t we replace the source adapter with the target adapter
Why not use FUSE for the complete suite?

Because Simpler May Be Slower

- Currently, FUSE I/O performance is limited
  - Always enforces a 4k transfer block size
  - Solutions?
    - Wait until corrected in a post 2.6 Linux kernel
    - Use the next SLAC xrootdFS release
      - Improved I/O via smart read-ahead and buffering
    - Use Andreas Peters', CERN xrootdFS
      - Fixes applied to significantly increase transfer speed
    - Just use the Posix Preload Library with **GridFTP**
      - You will get the best possible performance

Conclusions

- Scalla/xrootd is a solid base for an SE
  - Works well; is easy to install and configure
    - Successfully deployed at many sites
  - Optimal for most Tier 2 and Tier 3 installations
    - Distributed as part of the OSG VDT
- FUSE provides a solution to many problems
  - But, performance limits constrain its use

Future Directions

- More simplicity!
  - Integrating the cnsd into cmsd
    - Reduces configuration issues
  - Pre-linking the extended open file system (ofs)
    - Less configuration options
- Tutorial-like guides!
  - Apparent need as we deploy at smaller sites

Acknowledgements

- Software Contributors
  - Alice: Derek Feichtinger
  - CERN: Fabrizio Furano , Andreas Peters
  - Fermi: Tony Johnson (Java)
  - Root: Gerri Ganis, Beterand Bellenet, Fons Rademakers
  - STAR/BNL: Pavel Jackl
  - SLAC: Jacek Becla, Tofigh Azemoon, Wilko Kroeger
  - LBNL: Alex Sim, Junmin Gu, Vijaya Natarajan (BeStMan team)
- Operational Collaborators
  - BNL, FZK, IN2P3, RAL, UVIC, UTA
- Partial Funding
  - US Department of Energy
    - Contract DE-AC02-76SF00515 with Stanford University