



CMS FileMover: one click data

Valentin Kuznetsov
Cornell University
&

Brian Bockelman
University of Nebraska-Lincoln





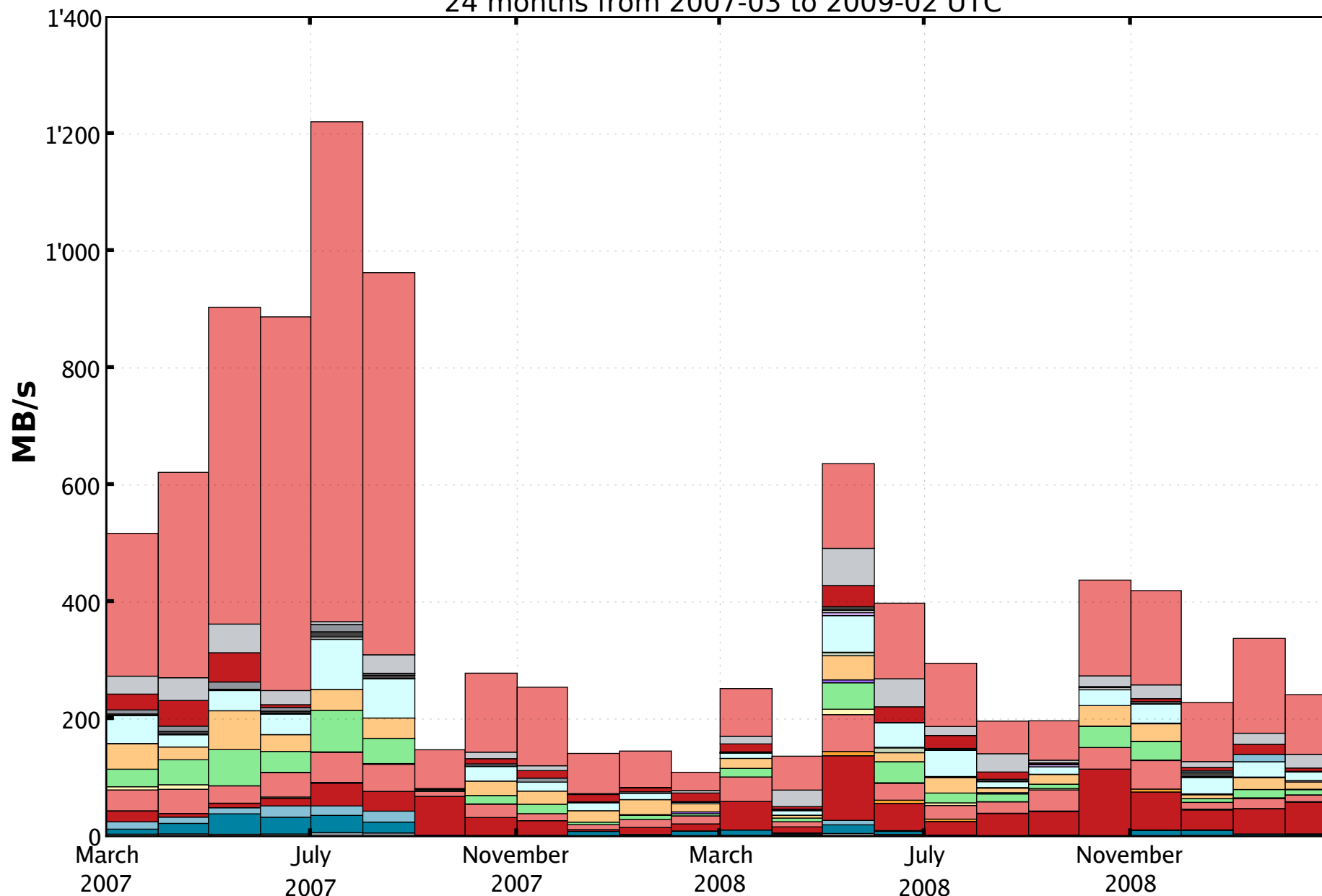
CMS data model

- CMS uses distributed dataset model, job comes to data
- CMS uses root I/O, all data written in ROOT format
- CMS uses central data placement and file transfer system called PhEDEx (Physics Experiment Data Export)
- Users can make requests to transfer data to their site of choice. Once request is approved it translated into transfer subscriptions. Subscriptions are checked periodically and new files placed into transfer automatically.

CMS data transfer

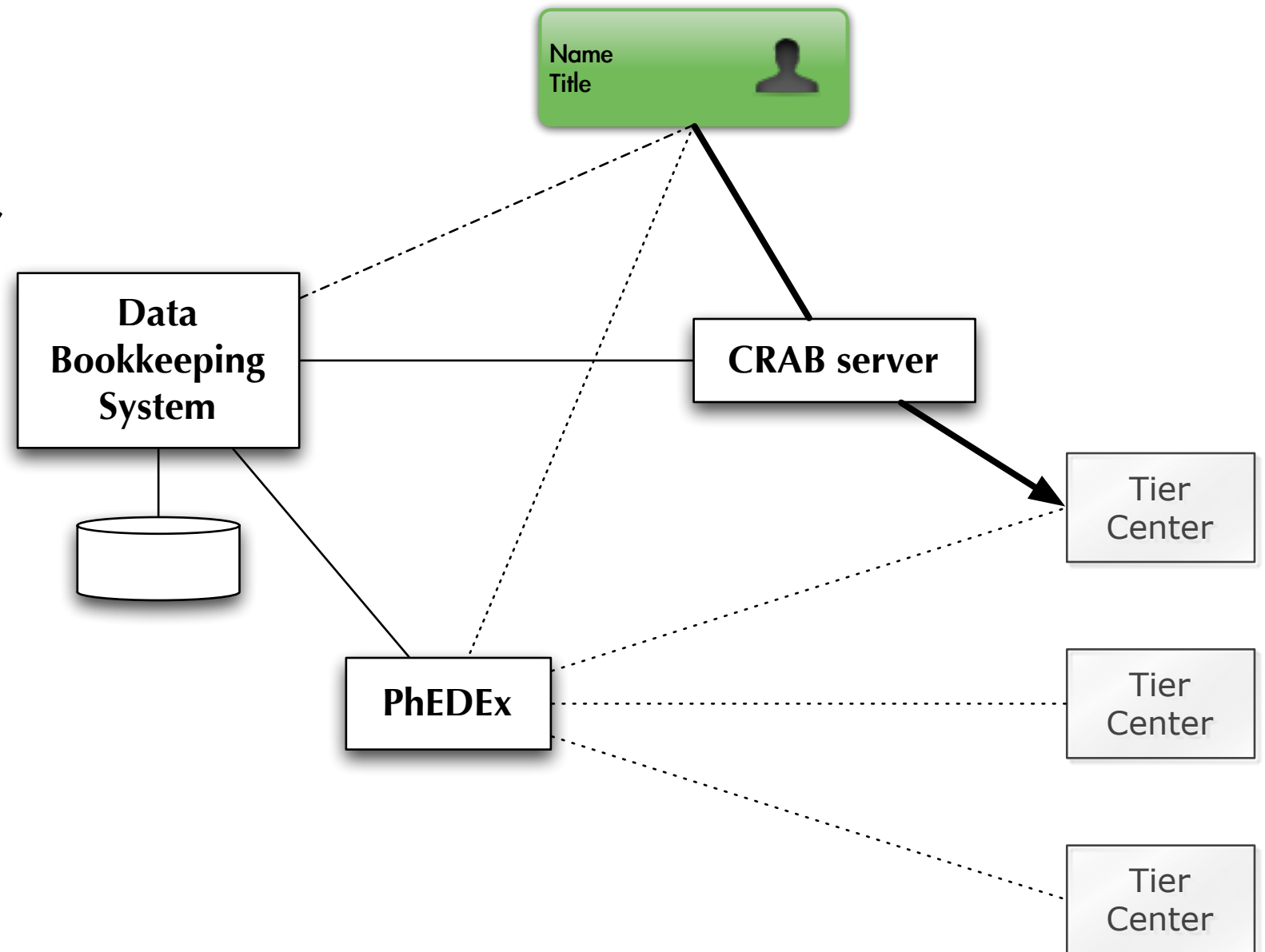
Monthly CMS PhEDEx transfer rate, Production

By destination country for non-tape storage only
24 months from 2007-03 to 2009-02 UTC



Typical user scenario

1. Find data in DBS
2. Request data transfer
3. Write job submission script
4. Submit job to GRID
5. Collect results
(transfer to my site)





Issues

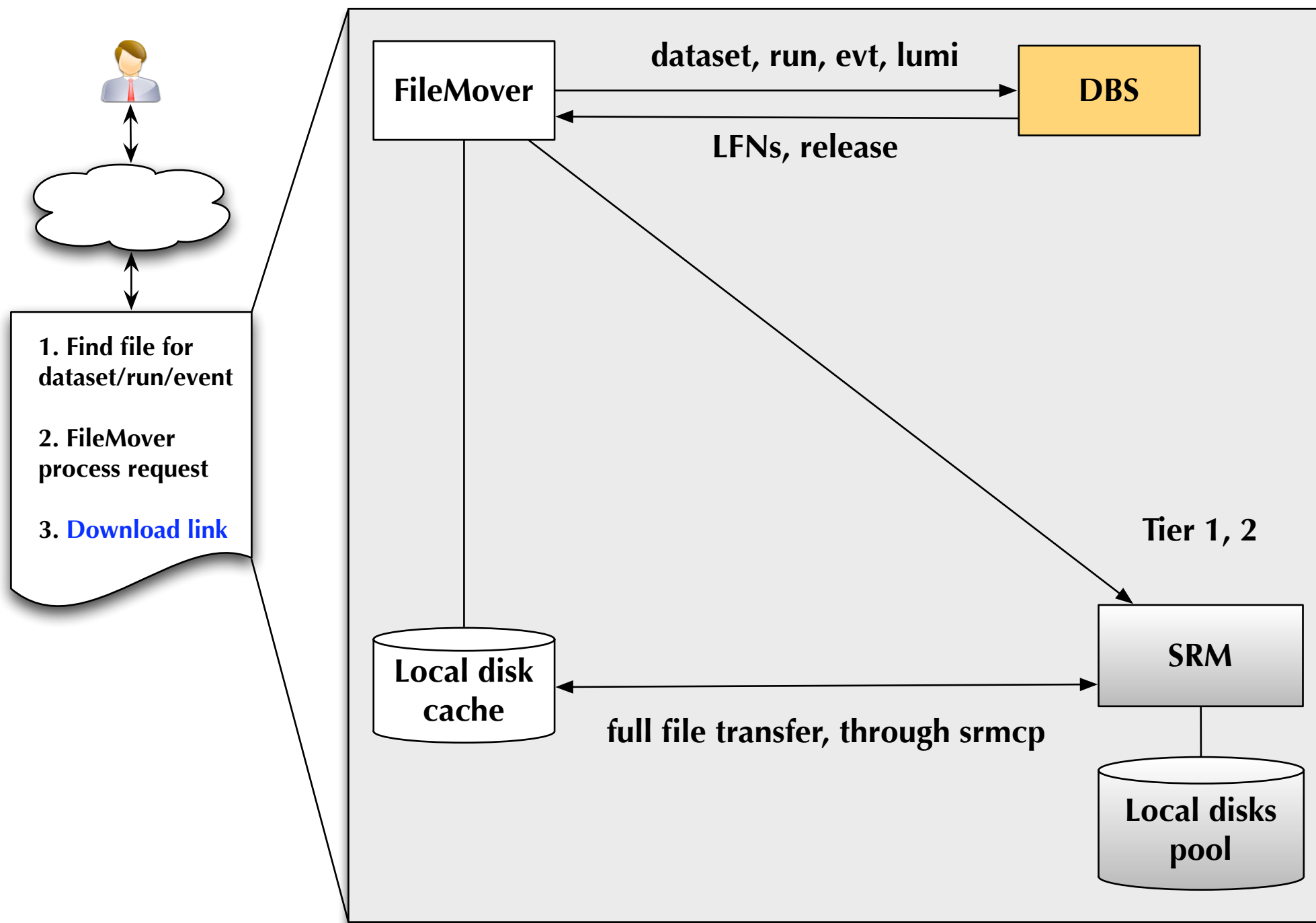
- Force users to learn underlying data management structure
- Users do their data bookkeeping along with data management tools
- Users unable to run interactive jobs, e. g. how you can run event display over the grid?
- Job failure rate is still quite high
- Users want to search and access data quickly (at the same time)
- Users want their data at local file storage (disk)



FileMover service

- The FileMover project was born to address users demand for quick, interactive data access
- Idea to hide complexity of underlying data management structure and provide intuitive interface, i.e. browse-click-download
- It consists of
 - Request file web-interface
 - Pick Event web-interface
 - CmsFS
- Located at <https://cmsweb.cern.ch/filemover/>

file request architecture





Request a file

Dashboard DBS Discovery DataTransfer SiteDB CondDB Support valya » logout

PhEDEx Home - FileMover

Hello valya: Request files Request events

Request file via LFN

Request Reset

Request file via dataset/run/event

dataset path or pattern (optional) run event or event range (optional)

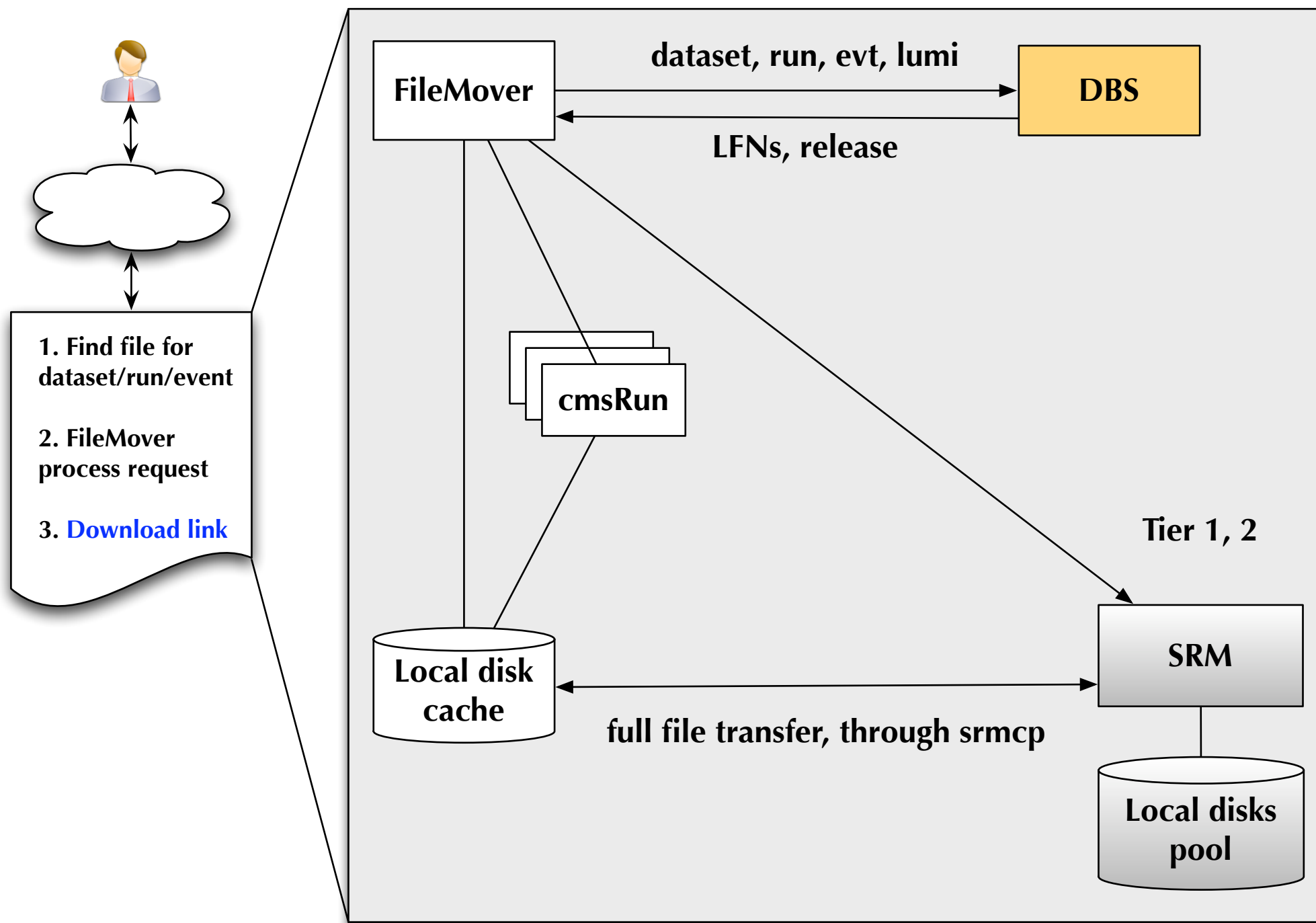
-

Request Reset

/store/data/CRUZET4_v1/Cosmics/RECO/CRZT210_V1_multiCosmicMuon_v1/0001/5CCDE2D7-1E73-DD11-ACF8-001A92971B0C.root	Download (8.5MB) Remove
/store/data/CRUZET3/Cosmics/RAW/v1/000/050/832/186585EC-024D-DD11-B747-000423D94AA8.root	Download (5.2MB) Remove
/store/data/CRUZET3/Cosmics/RAW/v1/000/050/796/4E1D3610-E64C-DD11-8629-001D09F251FE.root	Download (5.4MB) Remove
/cmssw/root_files/valya/12311867299.root	Download (2.7MB) Remove

- Allow to look-up data via file name, dataset/run/event
- Restrict for N simultaneous downloads, M requests/day

pick event architecture





Pick event

Dashboard DBS Discovery **DataTransfer** SiteDB CondDB Support
PhEDEx Home - FileMover

Hello valya: Request files **Request events**

Request file with given dataset/run/event/lumi (this service is currently limited to real data located at CERN)

Your Email (to be notified upon completion)

vkuznet@gmail.com

dataset

/Cosmics/Commissioning08-CRUZET4_v1/RECO

event sets: *run event lumi* (one per line)

```
58620 151578 6  
58620 151578 10|
```

Request Reset

Find events in desired sample upon user requests



FileMover: file/event interfaces

- Use local disk cache (1TB)
- Authenticate users, delegate request, fetch srmcp
 - use thread pool model, keep files in cache, share them among users
 - all complexity among data-services are hidden from users (e.g. DBS requests, site-lookup, etc.)
- Keep users updating with status via AJAX (e. g. you downloaded 10% of data)
- Once job is completed provide **Download** link and send Email notification
- Demo: <http://www.youtube.com/watch?v=XC00QIBRcIU>



Status/progress via AJAX

The screenshot shows the CMS File Server interface with the following components:

- Navigation:** Dashboard, DBS, Discovery, DataTransfer, SiteDB, CondDB, Support.
- User:** Hello valya: Request files Request events
- Request file via LFN:**
 - Input: /store/data/CRUZET4_v1/Cosmics/RECO/CRZT210_V1_multiCosmicMuon_v1/0001/5CCDE2D7-1E73-DD11-ACF8-001A92971B0C.root
 - Buttons: Request, Reset
- Request file via dataset/run/event:**
 - Fields: dataset path or pattern (optional), run, event or event range (optional)
 - Buttons: Request, Reset
- Requested file:** /store/data/CRUZET4_v1/Cosmics/RECO/CRZT210_V1_multiCosmicMuon_v1/0001/5CCDE2D7-1E73-DD11-ACF8-001A92971B0C.root has been placed into the transfer queue
- Transfer Progress Table:**

File Path	Progress
/store/data/CRUZET4_v1/Cosmics/RECO/CRZT210_V1_multiCosmicMuon_v1/0001/5CCDE2D7-1E73-DD11-ACF8-001A92971B0C.root	Data moving; 22.0%, 2.0 MB complete. Cancel
/store/data/CRUZET3/Cosmics/RAW/v1/000/050/832/186585EC-024D-DD11-B747-000423D94AA8.root	Download (5.2MB) Remove
/store/data/CRUZET3/Cosmics/RAW/v1/000/050/796/4E1D3610-E64C-DD11-8629-001D09F251FE.root	Download (5.4MB) Remove
- Terminal Output (Left Panel):**

```
Request successfully completed for
dataset : /Cosmics/Commissioni
eventset:
58620 151578 10
58620 151578 6

Number of events: 2
File download (3.0MB) | remove

Used release: CMSSW_2_1_17
Used config: show | hide
Output of edmFileUtil (file content): show

( 2 events, 3132014 bytes )

Printing FileIndex contents. 1
and Events stored in the root 1

Run      Lumi
58620    6
58620    6
58620    6
58620    6

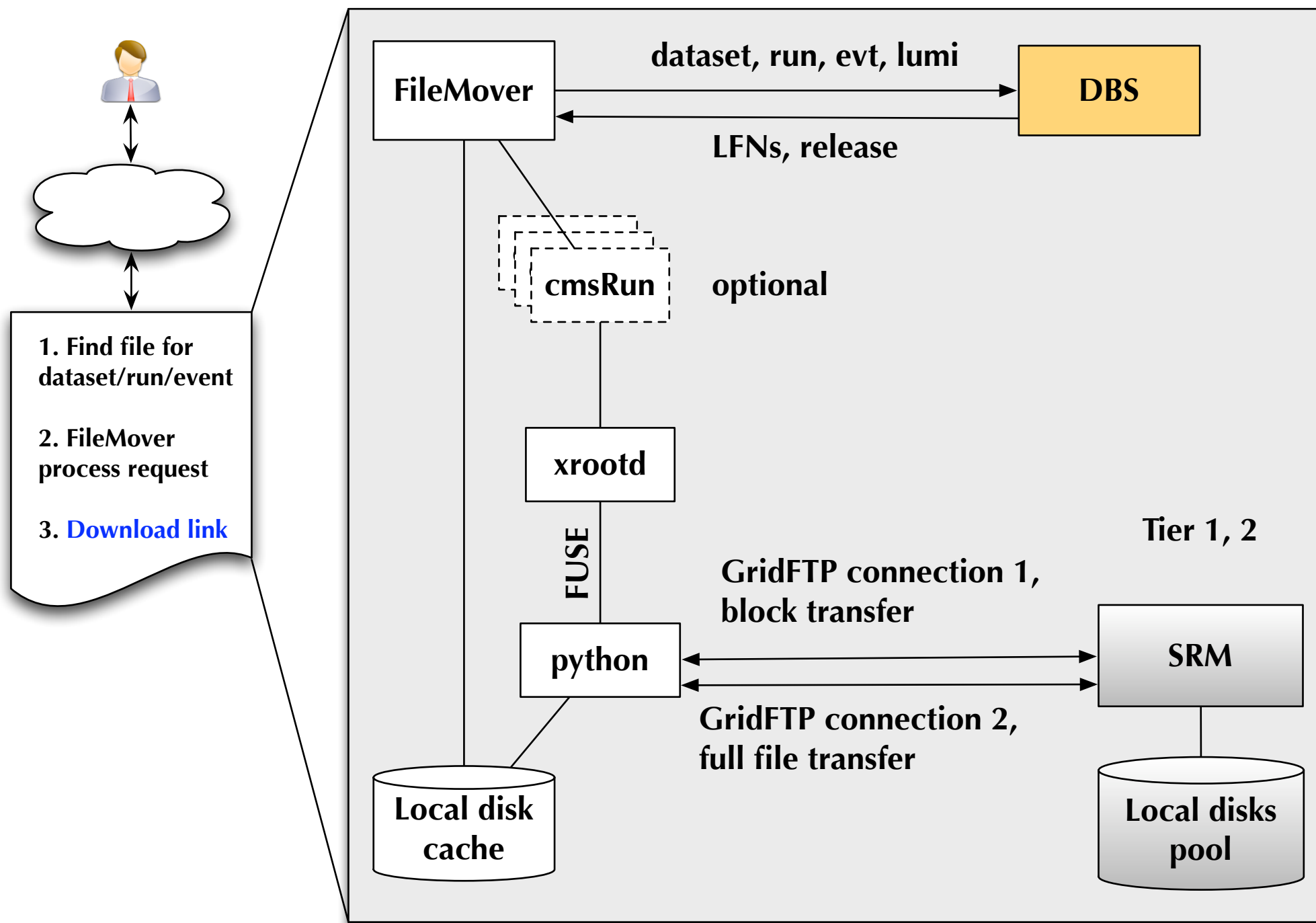
FileFormatVersion = 10. This v
Events are sorted such that fas
Events are sorted such that fas
(Note that other factors can p
```



FileMover: CmsFS

- Both file and pick-event interfaces have one downside, latency in file fetching
 - we look-up your data in DBS
 - we request your data from T2,T1 centers
 - download file to local cache and give you the download link
- How to avoid these limitations, e. g. event streaming
- Solution: CmsFS, file system which will allow to standard file operations over files located on remote sites, e.g. read, seek, close, etc. (POSIX I/O).

CmsFS architecture





FileMover CmsFS, cont'd

Benefits

- start downloading the file immediately
- once first event is read, you can access subsequent ones very fast, suitable event-display use case
- support POSIX I/O operations, easy to use in applications, cmsRun, event-display, etc.

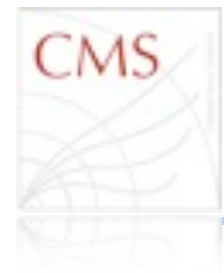
Limitations

- Initial time to access first event in a file is quite large, a few minutes. Time depends on “slowness” of remote site connection, event structure, FileMover cache.



FileMover CmsFS: status

- Base interface is ready, but work still in progress
 - simple and sufficient authentication schema
 - deployment & packaging
 - testing, testing, testing
- One global xrootd server for CMS usage
- Primary use case: allow CMS FireWorks event display to access events without extra dependencies and access LFNs in ROOT
- ✓ `cmsShow root://user@hostname//cmsfs/lfn/LFN_NAME`
- ✓ `TNetFile::Open("root://user@hostname//cmsfs/lfn/LFN_NAME");`



CmsFS usage

- Proof of concept and prototype already exists, but
- We must pay attention to scalability of the service
 - preliminary studies shown it can sustain up to 100 users, but realistic analysis is required
- We should not replace central data transfer system (PhEDEx)
 - keep well-defined policies and rules to prohibit users from nasty behavior
- We may replicate service upon further analysis (FNAL, CERN, etc. data centers)



Summary

- FileMover Service in production for several month
- code written in python with java srmcp client
- web interface based on CherryPy/Cheetah python frameworks + AJAX, it runs behind apache
- Proxy delegation run by CERN operator once a month
- CPU idle, < 1MB/s of network traffic on average
- More then 200 users use it
- Almost 300 files in local cache
- A few times seen orphan request due to unresponsive SE