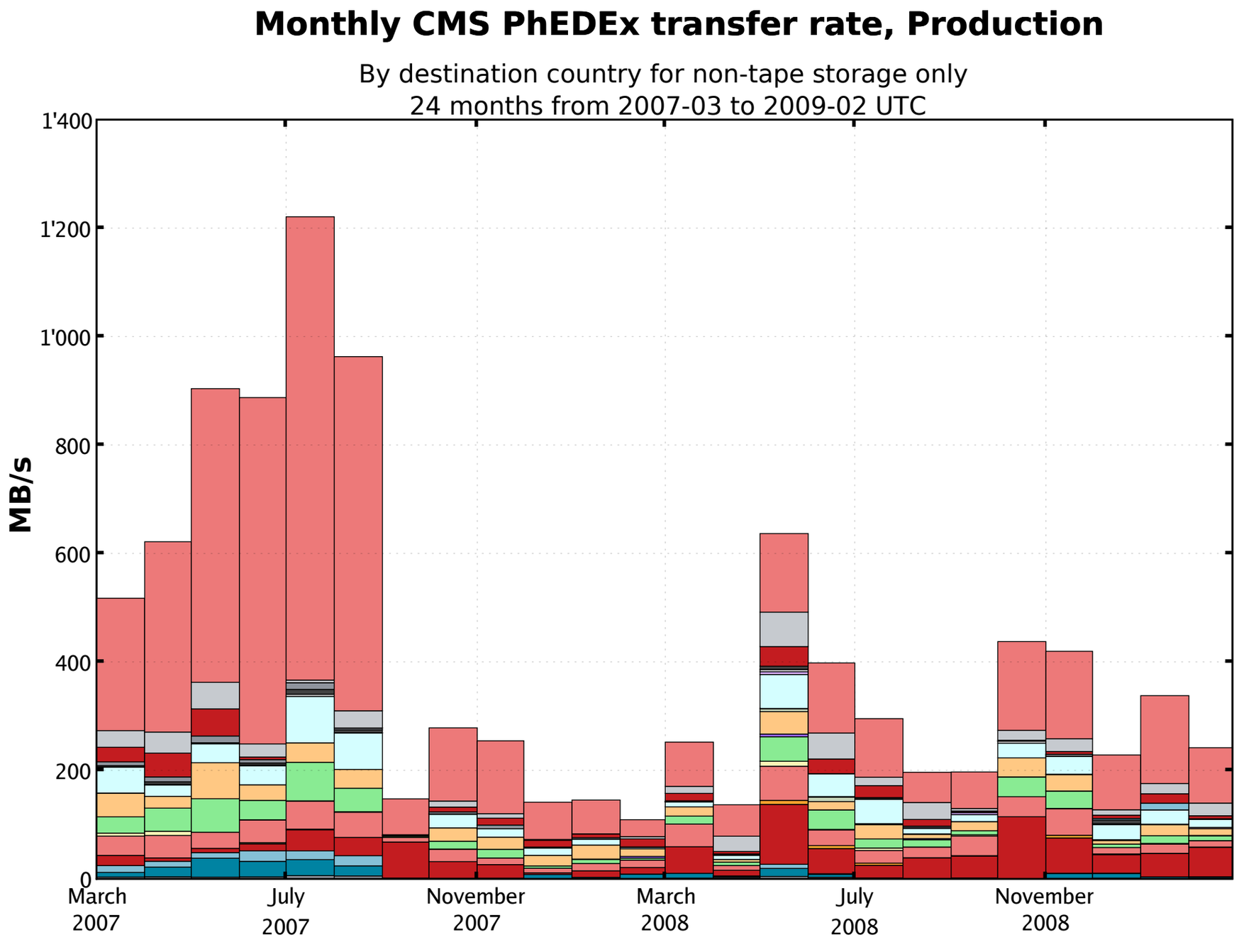
CMS FileMover: one click data

Valentin Kuznetsov
Cornell University
&
Brian Bockelman
University of Nebraska-Lincoln

CMS data model

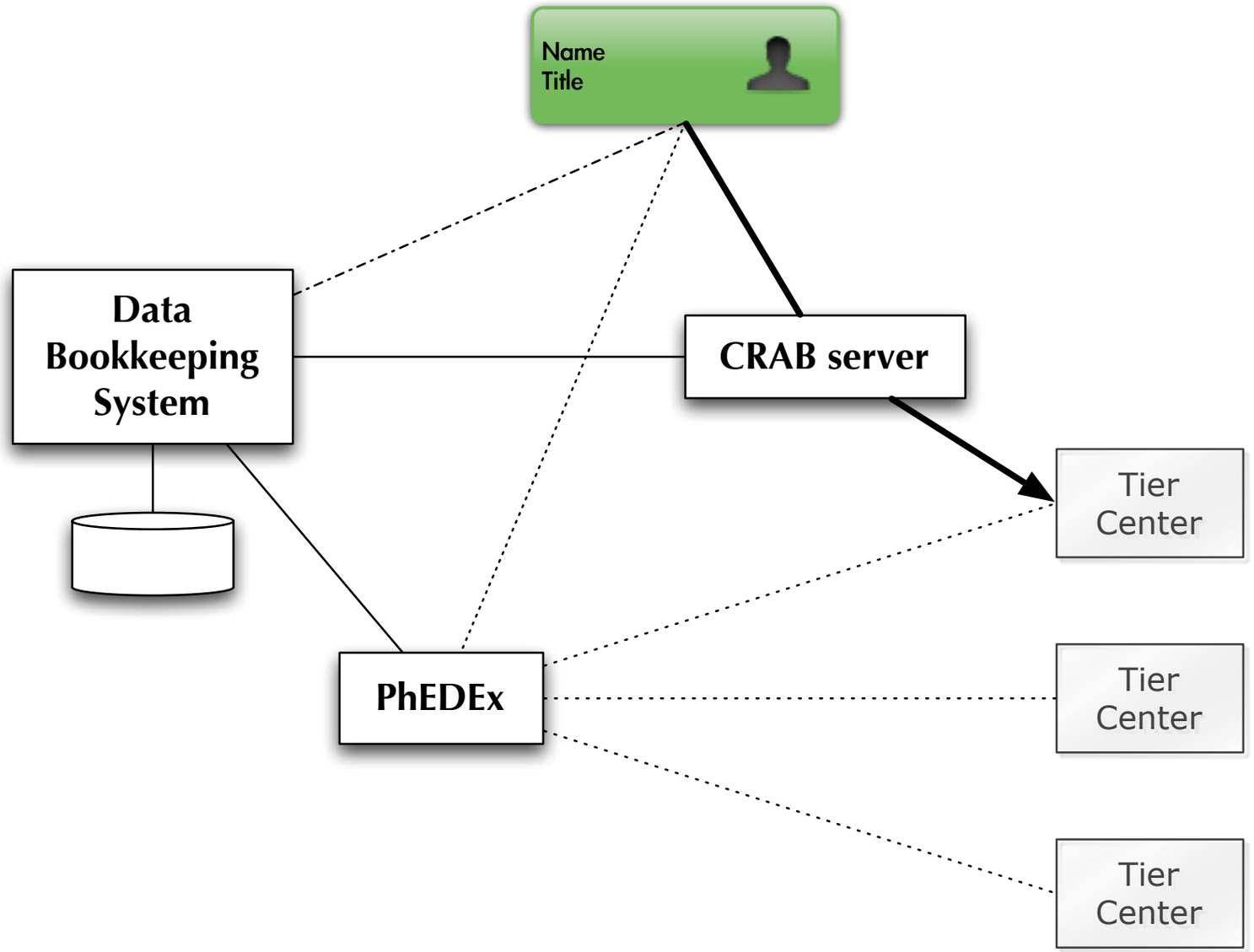
- CMS uses distributed dataset model, job comes to data
- CMS uses root I/O, all data written in ROOT format
- CMS uses central data placement and file transfer system called PhEDEx (Physics Experiment Data Export)
  - Users can make requests to transfer data to their site of choice. Once request is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

CMS data transfer



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)



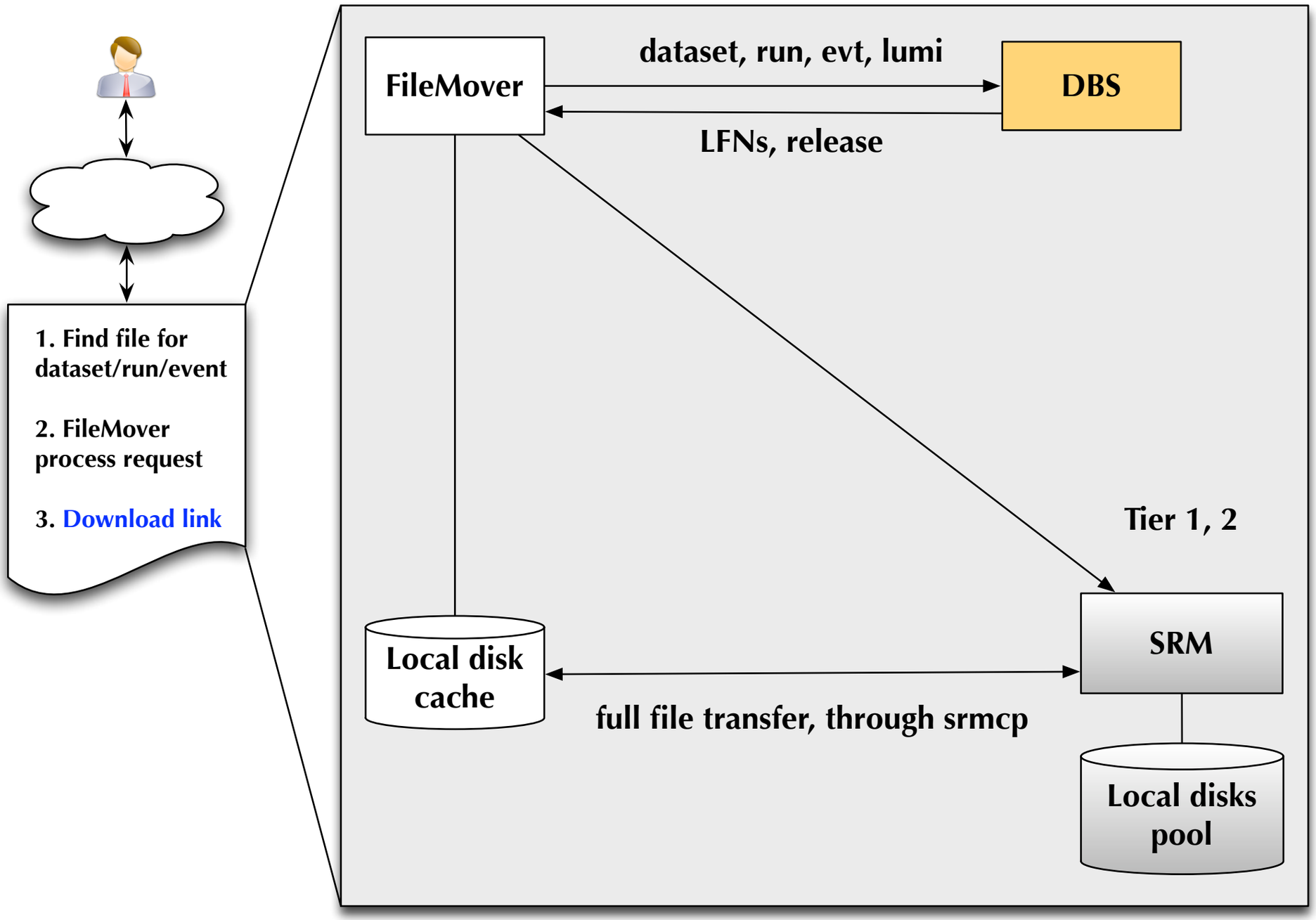
Issues

- 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

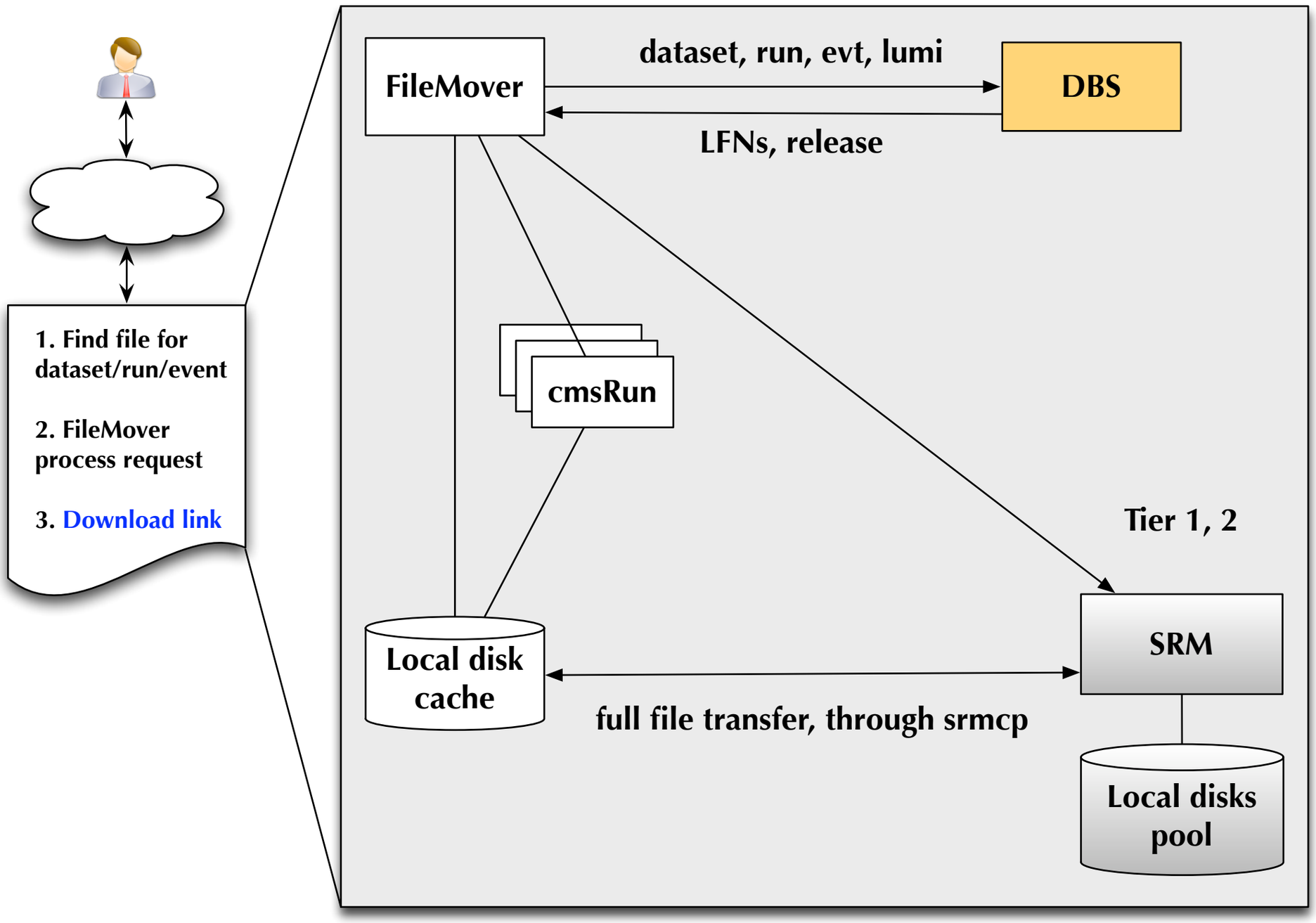
dataset path or pattern (optional) run event or event range (optional)

Request Reset

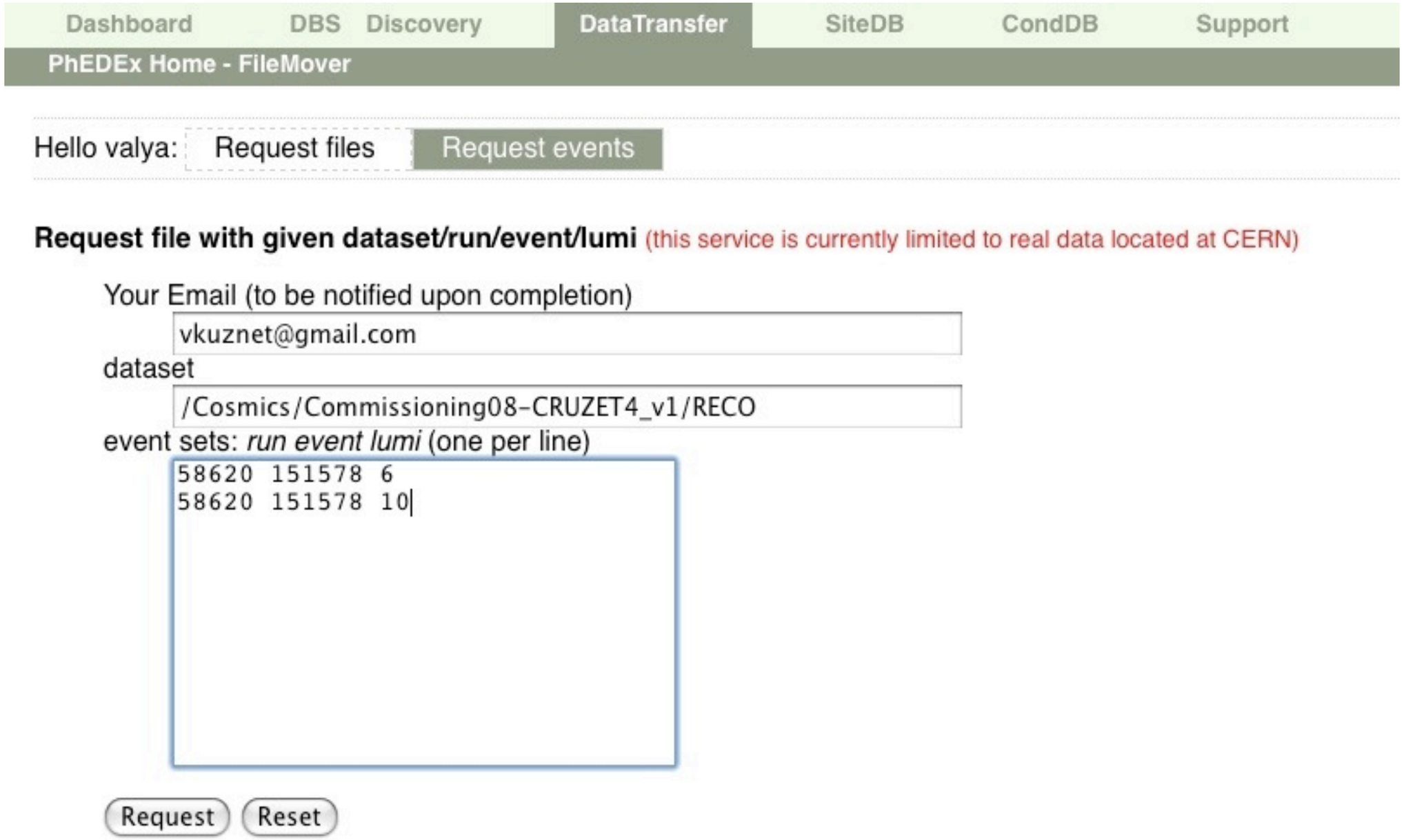
/store/data/CRUZET4_v1/Cosmics/RECO/CRZT210_V1_multiCosmicMuon_v1/0001/5CCDE2D7-1E73-DD11-ACF8-001A92971B0C.root	Download (8.5MB) I Remove
/store/data/CRUZET3/Cosmics/RAW/v1/000/050/832/186585EC-024D-DD11-B747-000423D94AA8.root	Download (5.2MB) I Remove
/store/data/CRUZET3/Cosmics/RAW/v1/000/050/796/4E1D3610-E64C-DD11-8629-001D09F251FE.root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

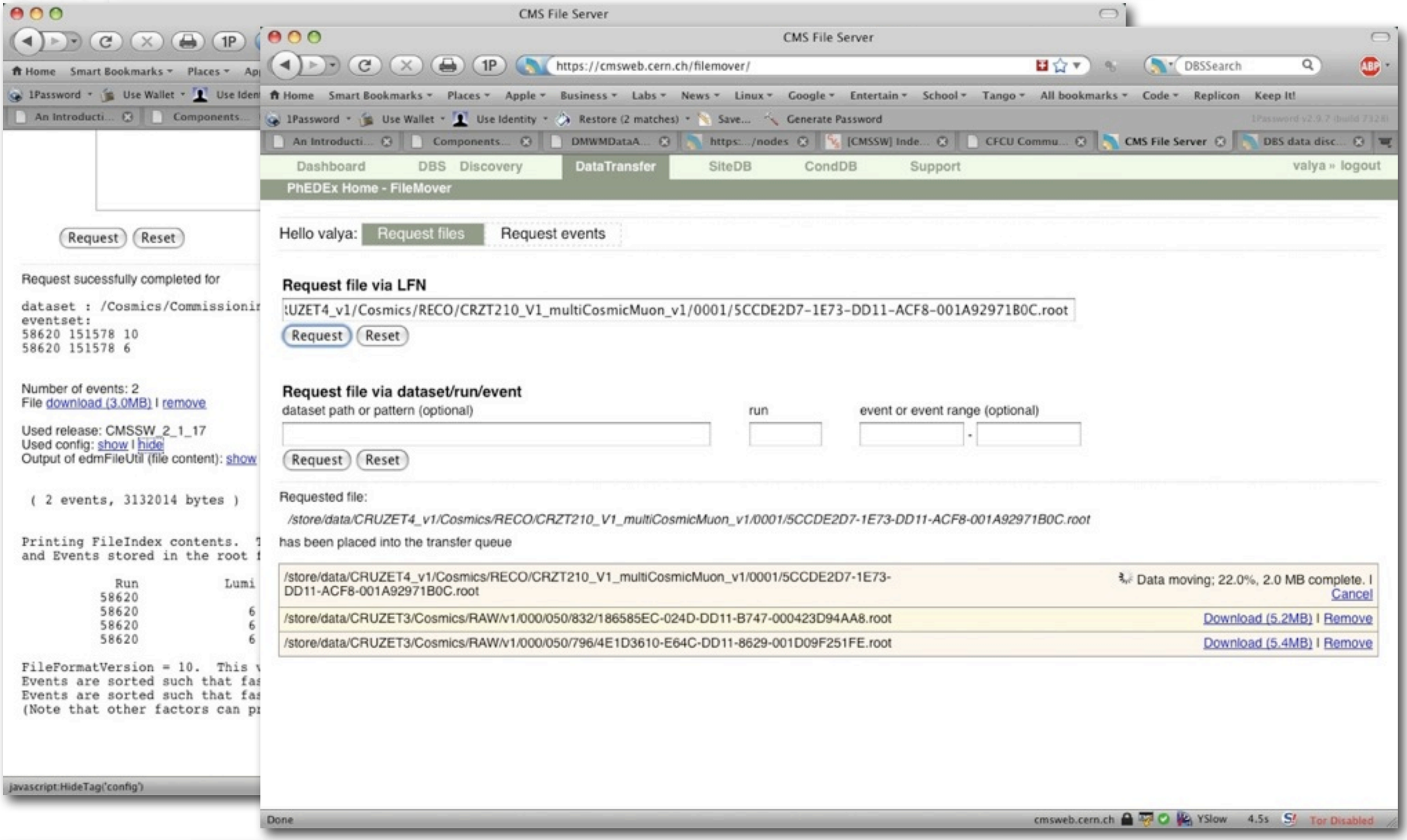


Find events in desired sample upon user requests

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=XCOOQ1BRcIU

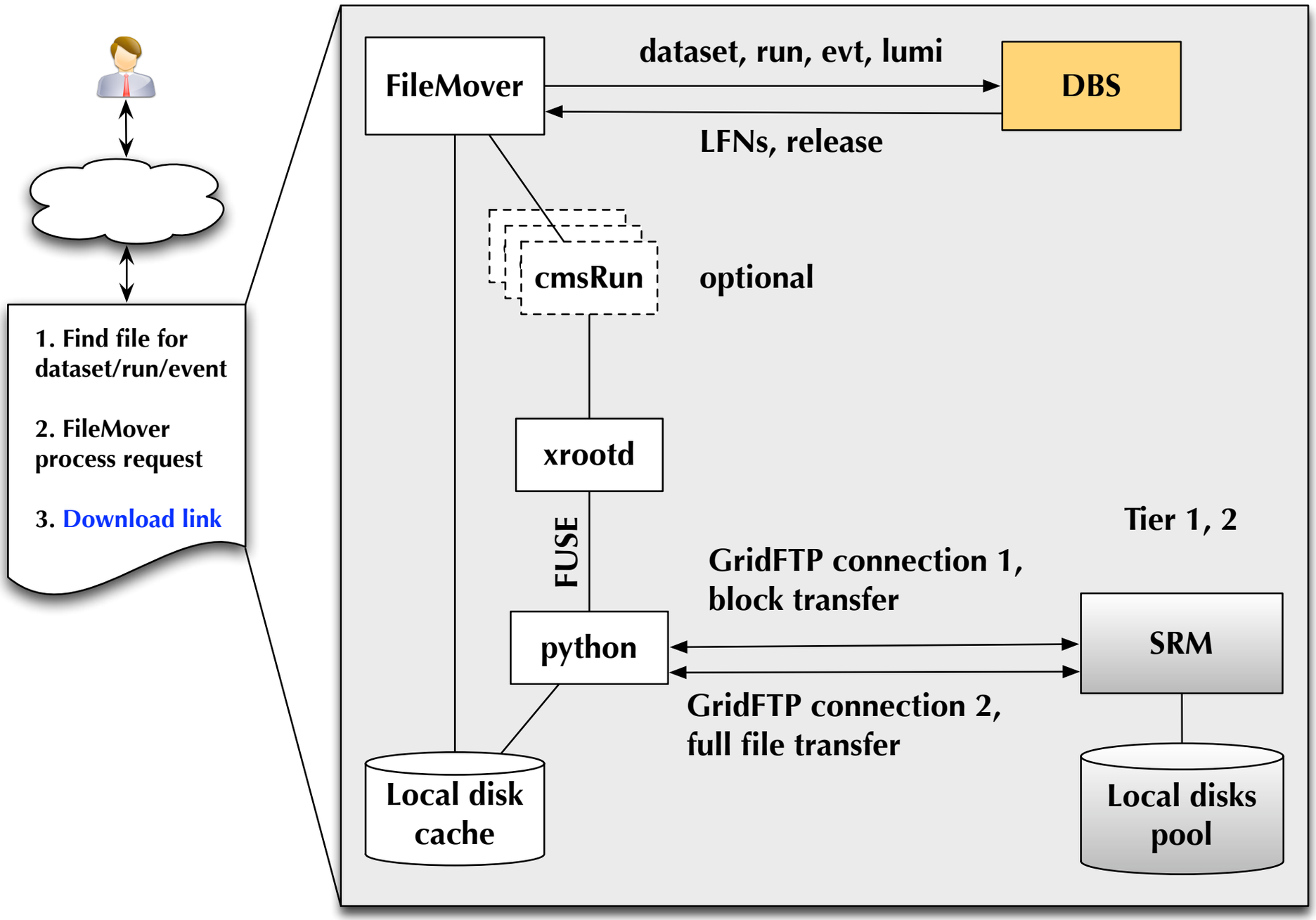
Status/progress via AJAX



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/lfns/LFN_NAME

✓ TNetFile::Open("root://user@hostname//cmsfs/lfns/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