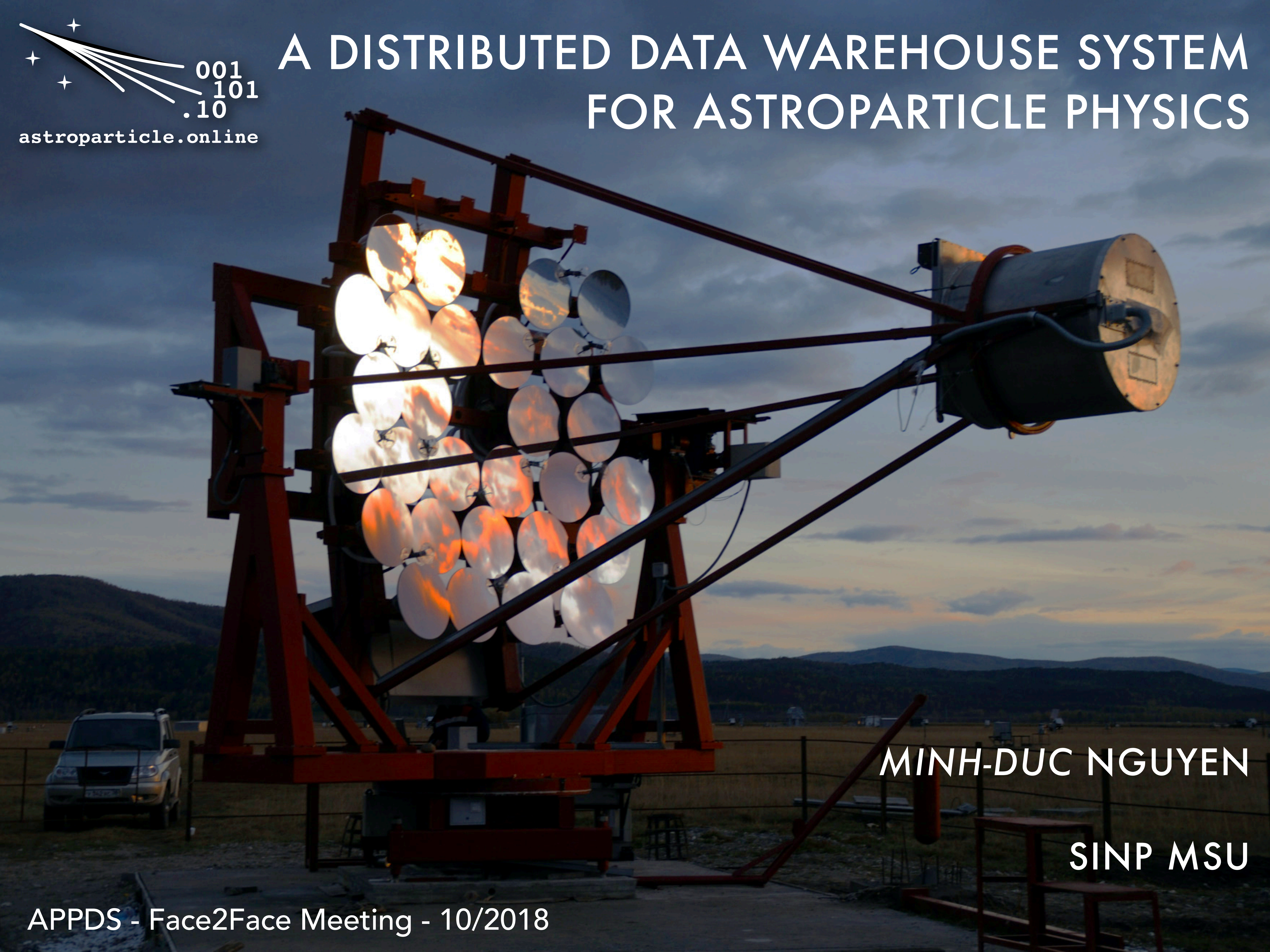


001
101
.10
astroparticle.online

A DISTRIBUTED DATA WAREHOUSE SYSTEM FOR ASTROPARTICLE PHYSICS

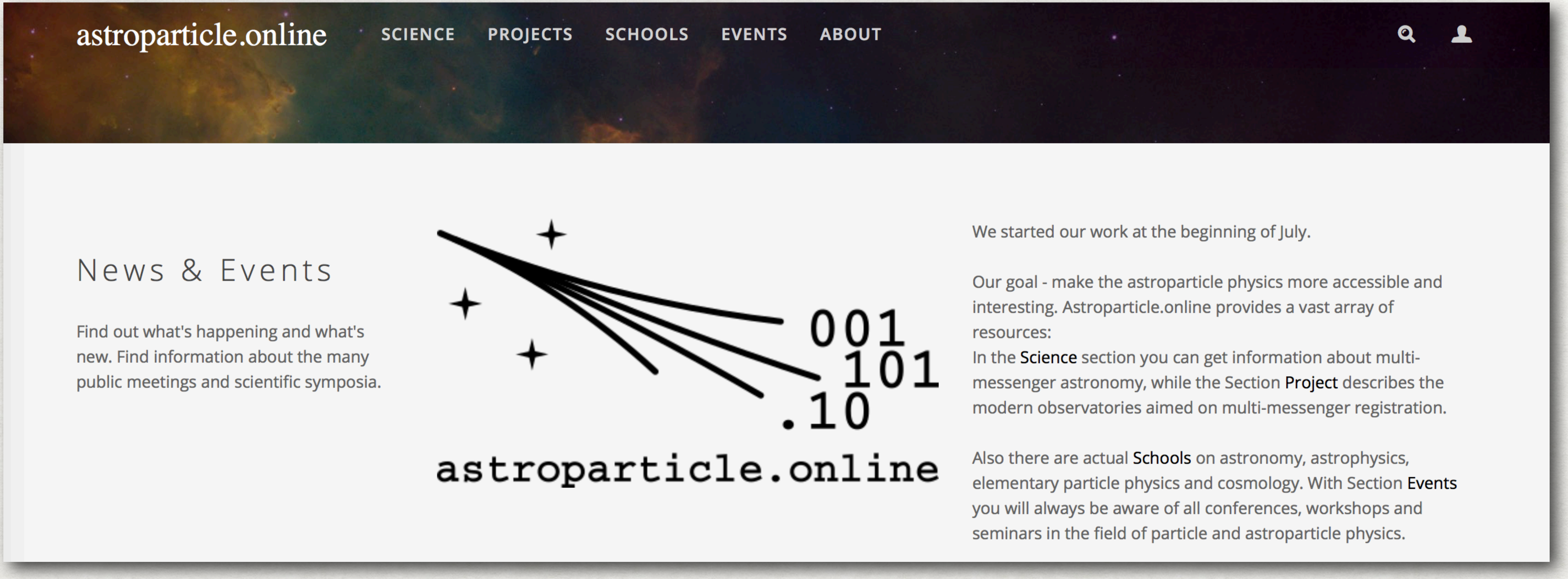
MINH-DUC NGUYEN

SINP MSU

APPDS - Face2Face Meeting - 10/2018

ASTROPARTICLE.ONLINE

- Karlsruhe-Russian Astroparticle Data Life Cycle Initiative
- Supported by RSF and Helmholtz
- Participants: SINP MSU, ISU, ISDCT SB RAS, KIT



News & Events

Find out what's happening and what's new. Find information about the many public meetings and scientific symposia.

We started our work at the beginning of July.

Our goal - make the astroparticle physics more accessible and interesting. Astroparticle.online provides a vast array of resources:
In the Science section you can get information about multi-messenger astronomy, while the Section Project describes the modern observatories aimed on multi-messenger registration.

Also there are actual Schools on astronomy, astrophysics, elementary particle physics and cosmology. With Section Events you will always be aware of all conferences, workshops and seminars in the field of particle and astroparticle physics.

REQUIREMENTS FOR THE DATA WAREHOUSE

- Multiple experiments (TAIGA, KASKADE, etc.)
- More than hundreds of terabytes of raw data at each site
- Remote access to data as local file systems
- On-demand data transfer by requests only
- Automatic real-time updates
- No change to existing site infrastructure, only add-ons

INTERACTION WITH USERS

- User asks: give me the files related to the event #E on timestamp #T with the energy range from #A to #B MeV at location #L
- The aggregation service prepares the file list from the DB and sends the response
- The client software after receiving the response makes a local mount point which contains the file list
- User works with files as if they are local
- Data chunks are delivered to the client only on system calls open(), read(), seek()

POSSIBLE SOLUTIONS

CernVM File system

APACHE hadoop

OpenAFS

CERNVM-FS

- Data are left untouched in their own file system

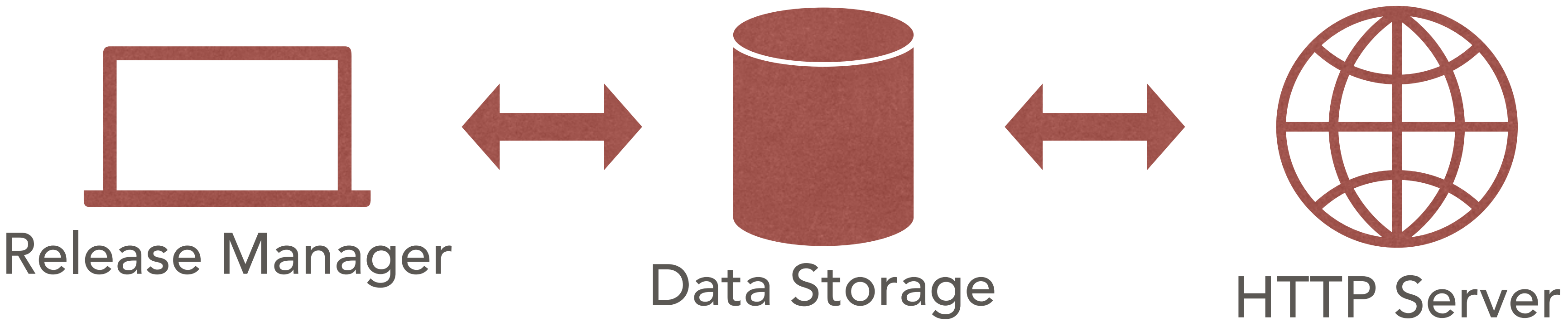
- CernVM-FS indexes the data and changes, stores only the metadata (indices, filenames, owners, modification times, checksums, locations, etc.) and the data tree structure

- CernVM-FS uses HTTP as the data transfer protocol, so there's no firewall problem

- Data transfer starts only on actual reads

- Multilevel cache-proxy servers

CERNVM-FS DATA LIFECYCLE

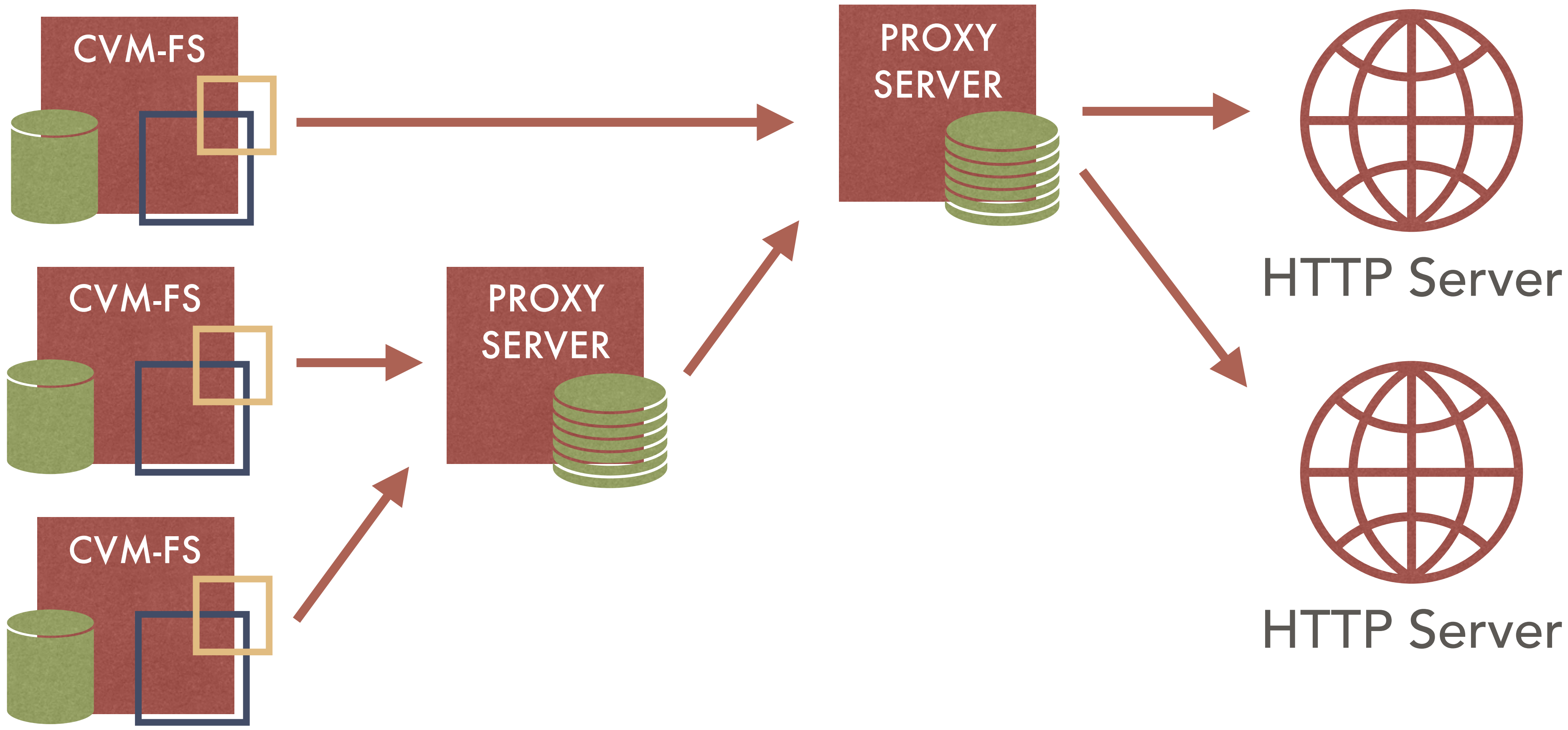


Release Manager

Data Storage

HTTP Server

CERNVM-FS DATA DISTRIBUTION



CVM-FS

CVM-FS

CVM-FS

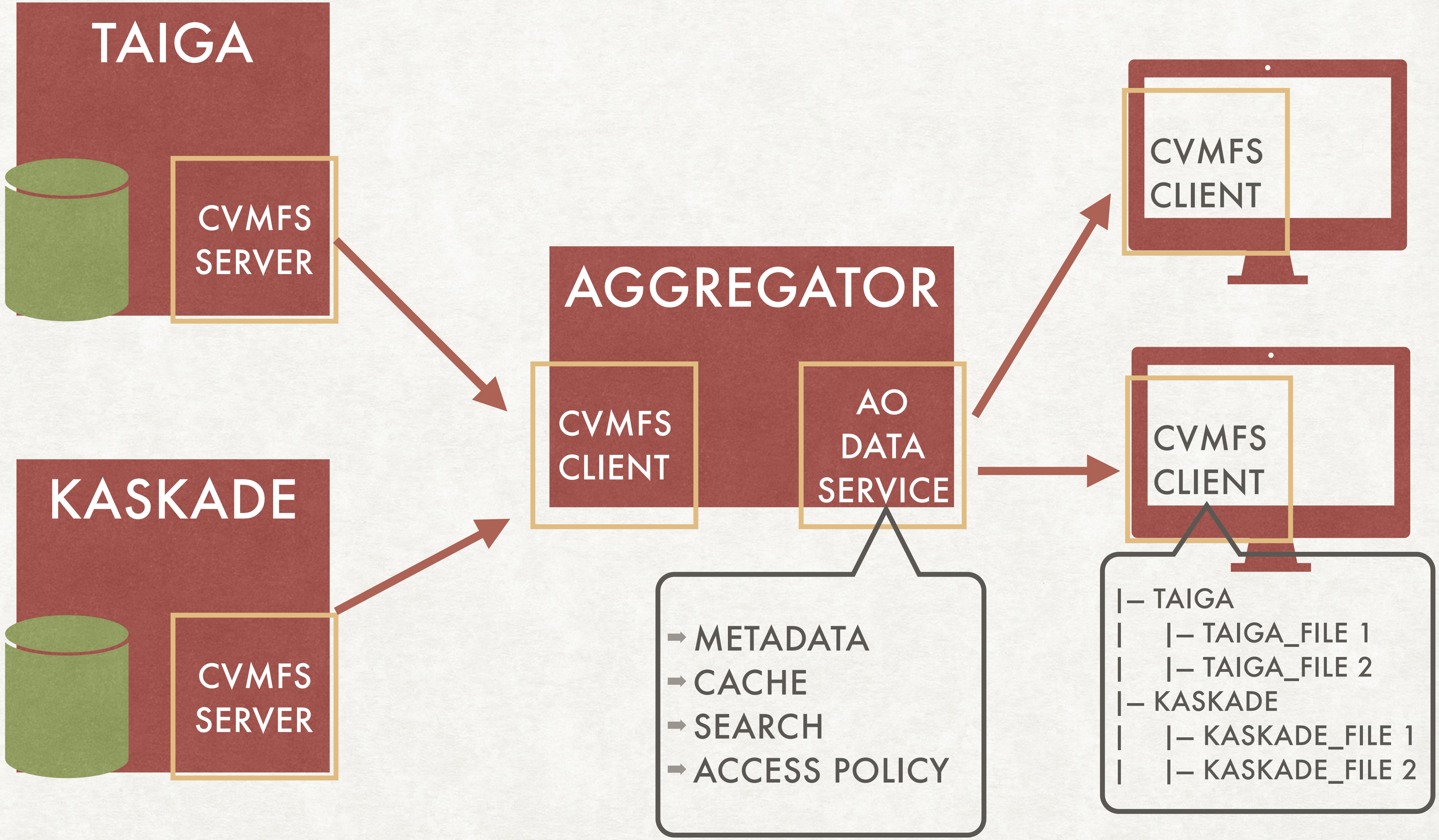
PROXY SERVER

PROXY SERVER

HTTP Server

HTTP Server

ASTROPARTICLE.ONLINE + CERNVM-FS



TAIGA

CVMFS SERVER

KASKADE

CVMFS SERVER

AGGREGATOR

CVMFS CLIENT

AO DATA SERVICE

- METADATA
- CACHE
- SEARCH
- ACCESS POLICY

CVMFS CLIENT

CVMFS CLIENT

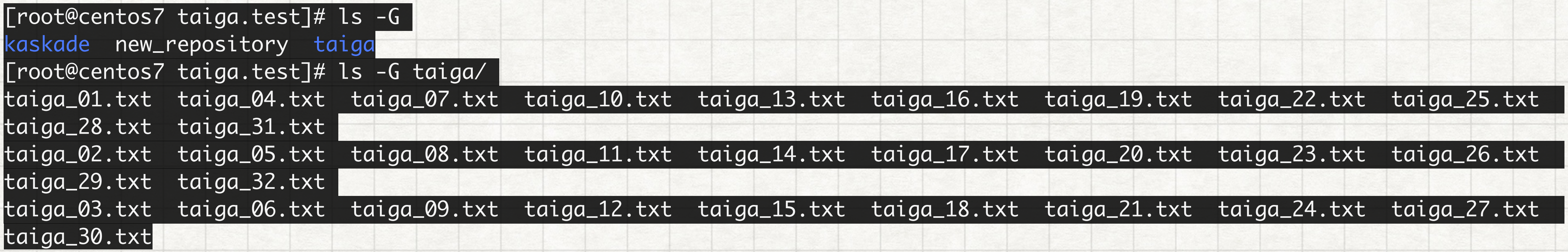
```
|– TAIGA
|     |– TAIGA_FILE 1
|     |– TAIGA_FILE 2
|– KASKADE
|     |– KASKADE_FILE 1
|     |– KASKADE_FILE 2
```

CURRENT STATUS

- ✓ Used CernVM-FS to export the existing data storage of each site as is without changing the file system
- ✓ Merged different data trees to a single one at the aggregation server level
- ✓ Discovered how to apply a filter to cvmfs to exclude a directory from user access
- Metadata search and API (in progress)

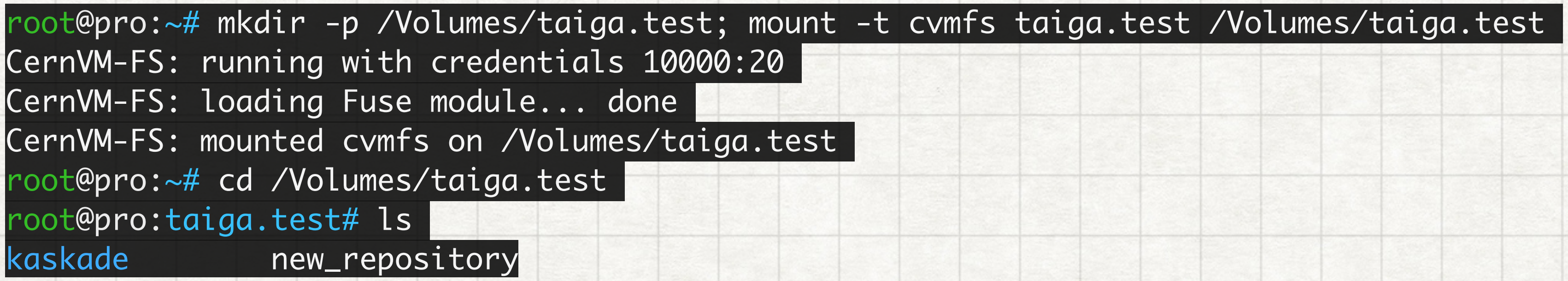
CVM-FS WITH FILTERS

THE ORIGINAL REPOSITORY



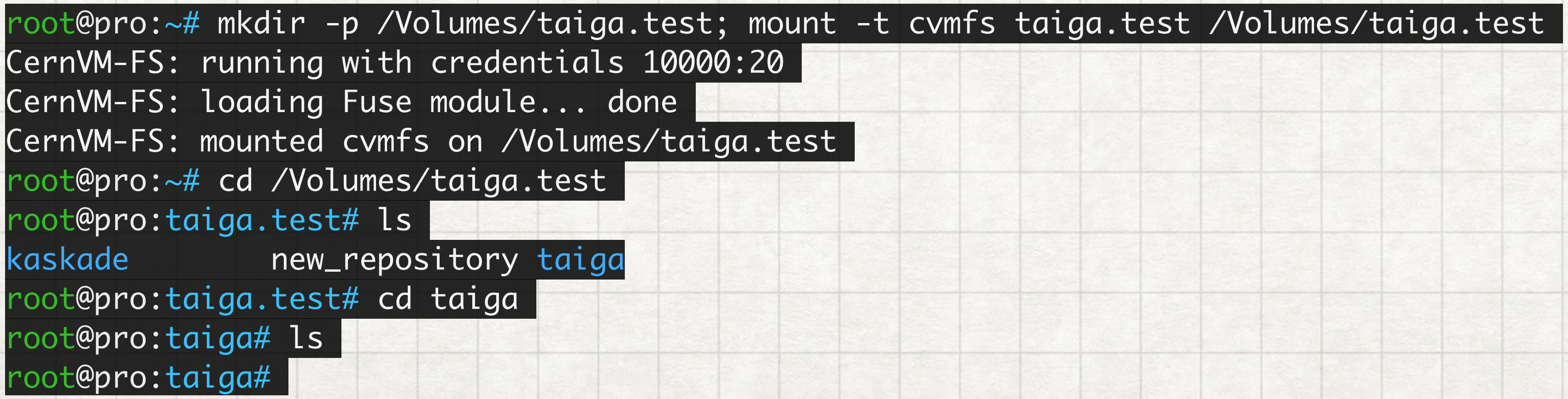
```
[root@centos7 taiga.test]# ls -G
kaskade  new_repository  taiga
[root@centos7 taiga.test]# ls -G taiga/
taiga_01.txt  taiga_04.txt  taiga_07.txt  taiga_10.txt  taiga_13.txt  taiga_16.txt  taiga_19.txt  taiga_22.txt  taiga_25.txt
taiga_28.txt  taiga_31.txt
taiga_02.txt  taiga_05.txt  taiga_08.txt  taiga_11.txt  taiga_14.txt  taiga_17.txt  taiga_20.txt  taiga_23.txt  taiga_26.txt
taiga_29.txt  taiga_32.txt
taiga_03.txt  taiga_06.txt  taiga_09.txt  taiga_12.txt  taiga_15.txt  taiga_18.txt  taiga_21.txt  taiga_24.txt  taiga_27.txt
taiga_30.txt
```

THE REPOSITORY WITH THE TAIGA DIRECTORY EXCLUDED



```
root@pro:~# mkdir -p /Volumes/taiga.test; mount -t cvmfs taiga.test /Volumes/taiga.test
CernVM-FS: running with credentials 10000:20
CernVM-FS: loading Fuse module... done
CernVM-FS: mounted cvmfs on /Volumes/taiga.test
root@pro:~# cd /Volumes/taiga.test
root@pro:taiga.test# ls
kaskade         new_repository
```

THE REPOSITORY WITH THE TAIGA'S FILES EXCLUDED



```
root@pro:~# mkdir -p /Volumes/taiga.test; mount -t cvmfs taiga.test /Volumes/taiga.test
CernVM-FS: running with credentials 10000:20
CernVM-FS: loading Fuse module... done
CernVM-FS: mounted cvmfs on /Volumes/taiga.test
root@pro:~# cd /Volumes/taiga.test
root@pro:taiga.test# ls
kaskade         new_repository taiga
root@pro:taiga.test# cd taiga
root@pro:taiga# ls
root@pro:taiga#
```

CVM-FS FILTER IMPLEMENTATION

- ✓ Injected the filter into SQLite3 queries using the Catalog API
- ✓ The filter injection must be added into the CVM-FS Client not CVM-FS Server

- Issue #1: Currently, changing the filter requires recompilation
- Issue #2: When the global repository grows the filter size grows
- Issue #3: How to identify the users?
- Searching for uncertainty in more than 2000 files of cvm-fs source code is tough :) but doable :)

FUTURE WORK

- Sub-tree export (build a CVM-FS middleware module to do SQL injection with filters based on users' requests)
- Data access policy and API (RESTful API or GraphQL?)
- Metadata indexing and parameterised search (RDBMS (PostgreSQL) or NoSQL (column-based or row-based)?)
- HDFS-prototype and AFS-prototype
- Benchmark

“

THANK YOU!

— *Minh-Duc Nguyen <conqueror@dec1.sinp.msu.ru>*

”