

CTA: CERN (CASTOR) Tape Archive Rationale and Status

Germán Cancio, Daniele Kruse, Eric Cano, Steven Murray

A talk in two parts

What is CTA - Presented by Eric Cano

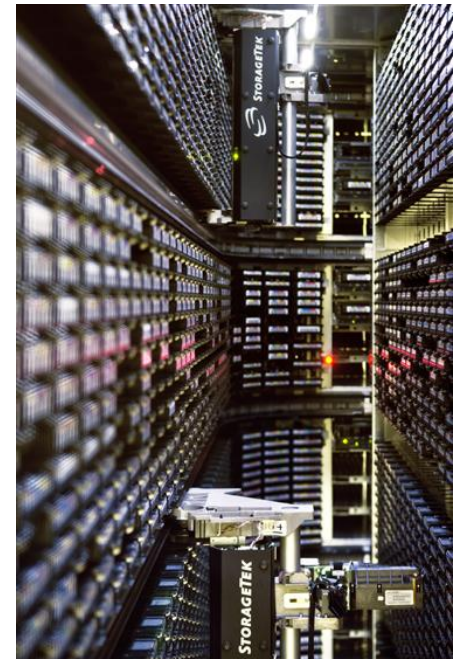
- Requirements of a tape storage system
- Architecture of CTA (CERN/CASTOR Tape Archive)

Rationale and status - Presented by Steven Murray

- Rationale
- The prototype / proof of concept
- What's next

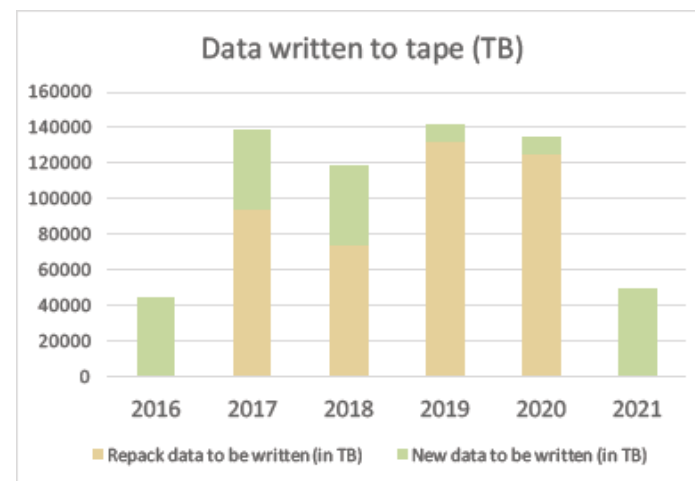
Main characteristics of tape

- It's cheap, safe, power efficient
- Media is the biggest cost driver
- Tapes can be reformatted to higher density when newer drives come out
- High throughput
 - 360MB/s per drive today, 1GB/s already in the roadmaps
- High capacity
 - 7-10TB per tape, 220TB demonstrated in labs
 - About 8 hours to fill or read a complete tape
- High latency
 - Tape mount and unmount take ~30s/1min+ each
 - Full tape seek ~30s
 - This does not improve with new equipment
- Sequential access
- Low concurrency
 - Limited number of tape drives in the library
- Locality and format constraints
 - Drive only mount tapes from their own library and right format



Our infrastructure and forecast

- 2 vendors
- 7 libraries (~70k slots)
- ~30k tape (7-10TB each)
- 80 drives (250-360MB/s)



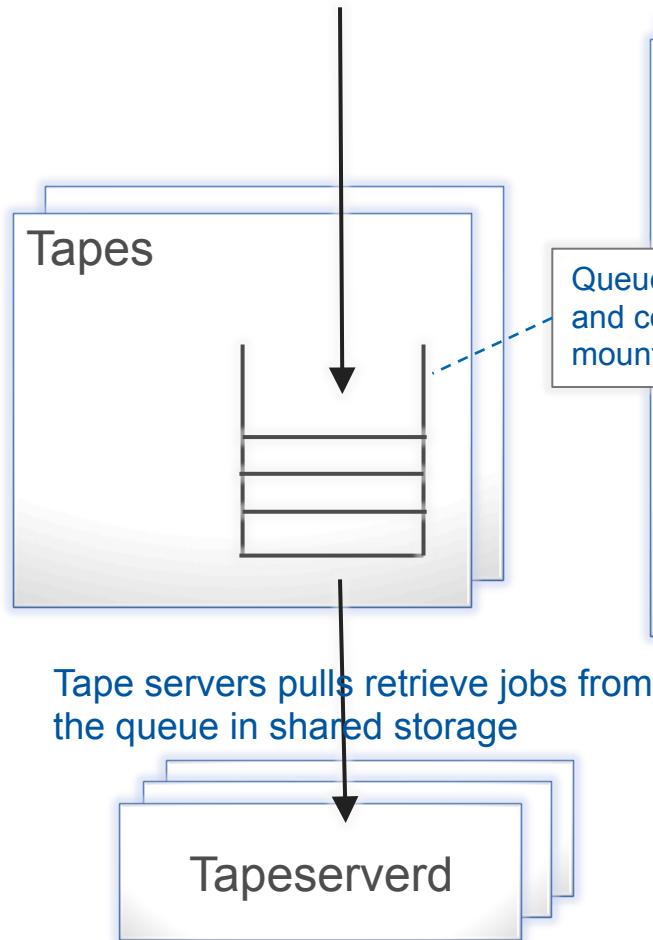
- Expected load for the coming years
 - 2016 last cool year (40PB to write, as much to verify, plus user reads)
 - From 2017 on write 120-140PB/year (multiply by 2-3 for reads)
 - Repack and verification are high volume, low priority tasks
 - Should yield drives to user activity

Requirements for tape efficiency

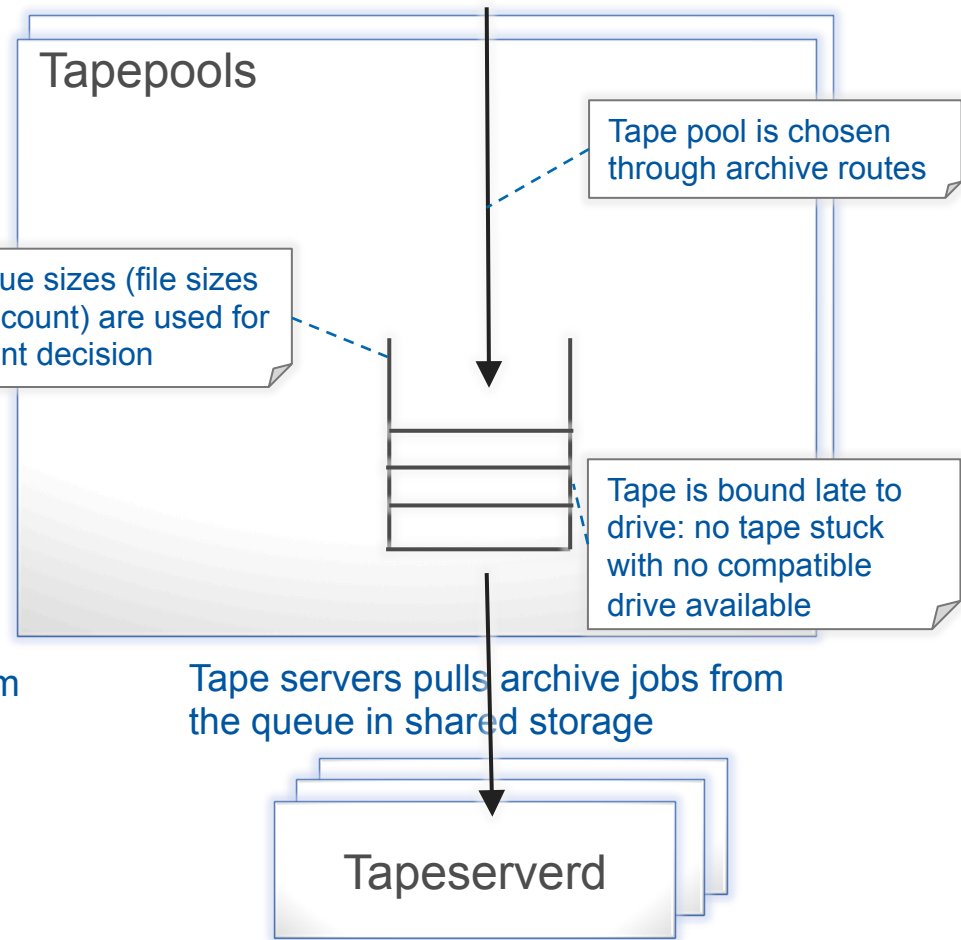
- Mounts should be made worthwhile
 - 1 mount operation \Leftrightarrow 20GB of bandwidth
 - Queue file transfers until we have enough data to transfer
- Single step mount decision for tape-drive couple
- Drives should run full speed
 - Buffering in memory (12-64GB/drive) absorbs glitches
 - Disk system should achieve proper average
- Drives should run all the time
 - Repack and verification should fill all idle drive time
 - ... but also yield to experiment activity
- Data has to be repacked regularly
 - Allowing re formatting of media
- Data needs to be verified
 - Detection of problem tapes

Jobs queues

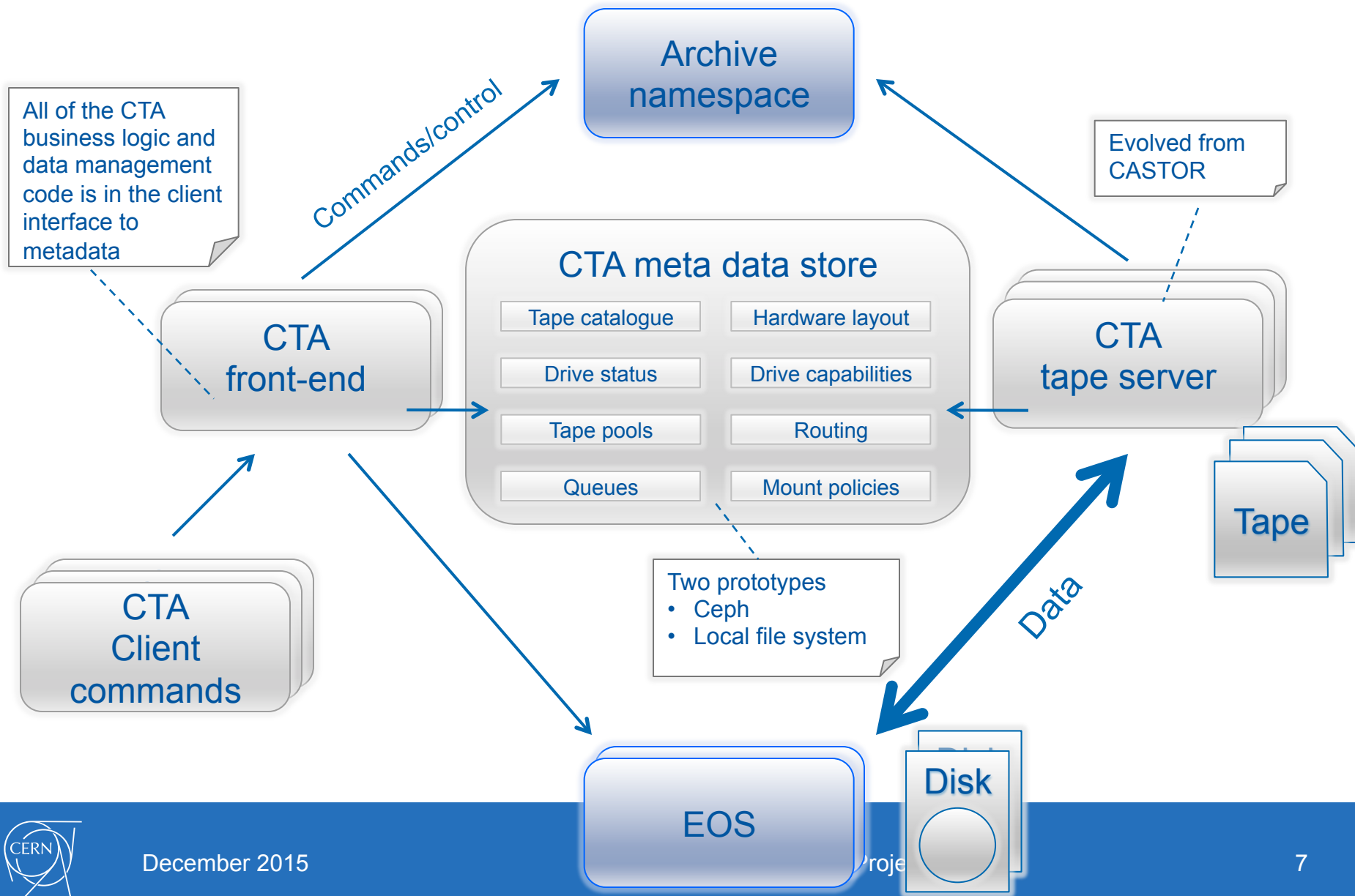
Retrieve jobs queued via front ends



Archive jobs queued via front ends



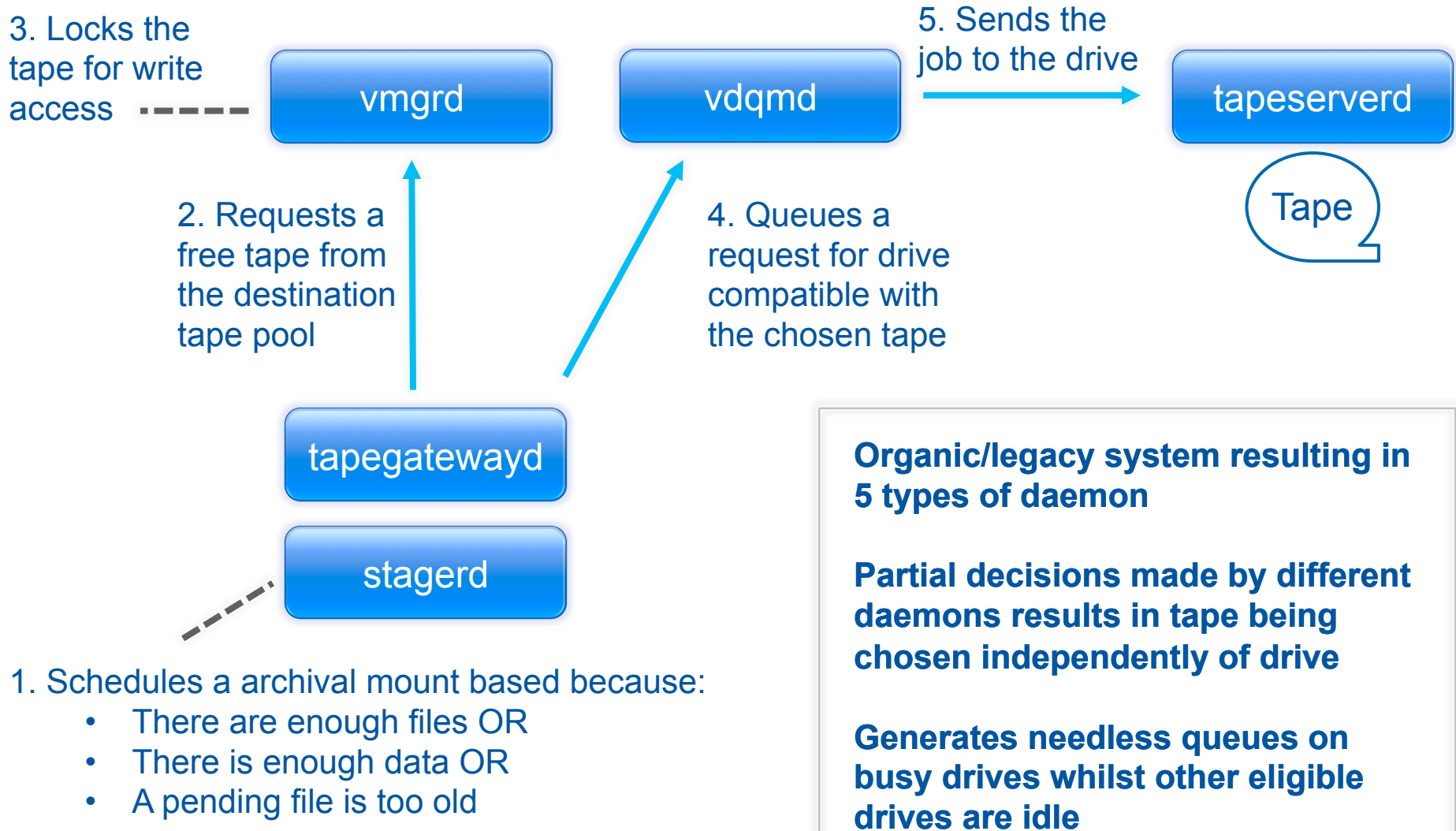
Global architecture



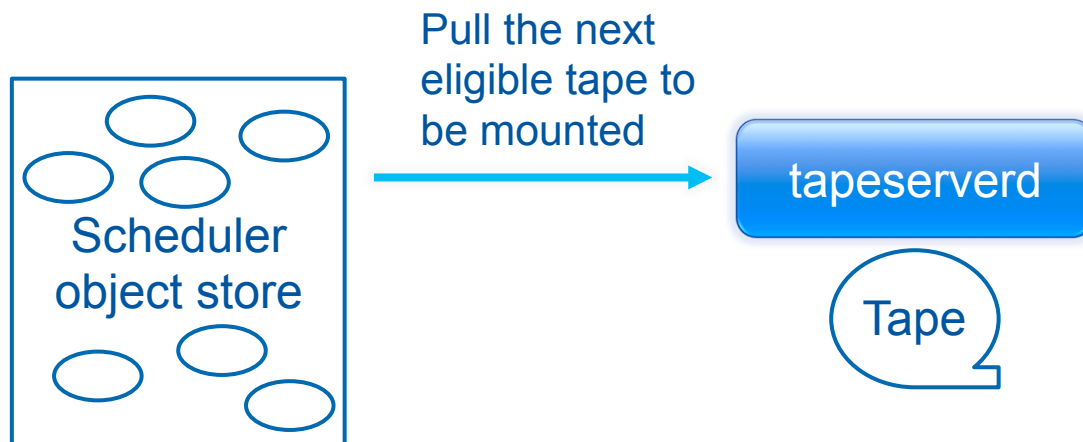
Rationale for CTA

- Less daemons
 - **CASTOR**
 - **5 types of daemon**
 - stagerd + tapegatewayd + vmgrd + vdqmd + tapeserverd
 - **CTA**
 - **2 types of daemon**
 - xrootd (front end) + tapeserverd
- Tapes and drives can be scheduled simultaneously

Scheduling a CASTOR archival



Scheduling a CTA archival



Single step “pull” scheduling using only a single type of daemon

Tape and drive scheduled simultaneously

Workload is naturally load balanced across eligible drives

Rationale for CTA

Simpler solution for a simpler problem

- No need to support random access
 - Tape access has evolved from random access to bulk archival and retrieval
- Centralized disk scheduling no longer needed
 - Distributed striped file systems now exist
 - Only really need to prioritize tape streams
 - Can concentrate on scheduling tape

Rationale for CTA

- Less wire protocols
 - Will follow the EOS approach of using XrootD
- All scheduling information in one place (hardware catalogues, queues, policies)
 - Global view
 - Easier to understand
 - Easier to improve
 - Easier to maintain
- Support preemptive scheduling
 - Throttle repack and tape verification
 - Use drives 100% of the time with little operator effort

Rationale for CTA

- Avoid duplication of disk management between CASTOR and EOS
- Preserve the knowledge and code driving the tape hardware from CASTOR
- System boundary between EOS and CTA
 - Clean separation of concerns
 - Independent EOS and CTA releases
- Simpler system to operate
 - Less daemons
 - Tape operators responsible for stager disks

CTA prototype

What was in the prototype

- End user and admin command-line tools
- Frontend server – A CTA plugin for xrootd
- Ported tapeserverd from CASTOR to CTA
- Central object store for scheduling
 - Hardware catalogues
 - Policies
 - Queues

CTA prototype

What was shown by a demo of the prototype

- Archived files from EOS to tape
- 2 tape drives were used in parallel
- Each EOS file had two tape replicas
- Retrieved the file back from tape to EOS
- Provided a user interface targeted at end users and at administrators

CTA prototype

CTA commands used during the demo

Admin commands

```
cta logicallibrary add -n IBM1JB -m "the test lib"  
cta tapepool add -n cms -p 15 -m "CMS raw"  
cta tape add -v I21748 -l IBM1JB -t cms -c 1000000000000 -m "A cta tape"  
cta tape add -v I21805 -l IBM1JB -t cms -c 1000000000000 -m "A cta tape"  
cta tape add -v I21902 -l IBM1JB -t cms -c 1000000000000 -m "A cta tape"  
cta storageclass add -n single -i 2 -c 1 -m "A single copy class"  
cta archiveroute add -s single -c 1 -t cms -m "Route to cms"  
cta mkdir /cms  
cta setstorageclass /cms single
```

File commands

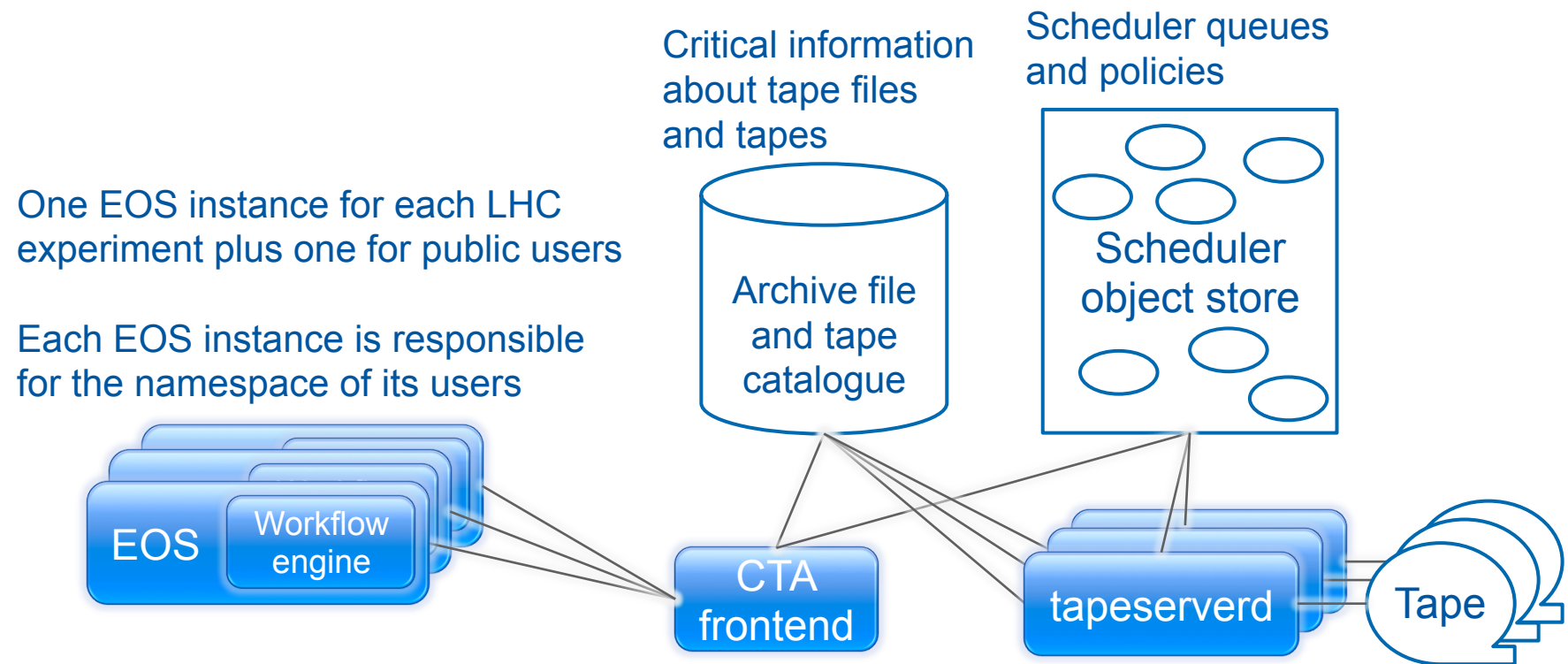
```
cta archive "eos://eos/cms/smallfile" /cms/smallfile  
cta ls /cms  
cta listpendingarchives  
sleep 60  
cta ls /cms  
  
cta retrieve /cms/smallfile "eos://eos/cms/smallfile_retrieve"  
cta listpendingretrieves
```


What's next

- Conclusions from architecture meetings
 - Several models and approaches were discussed within the section and group
 - Models ranged from
 - Putting CTA on the back of EOS
 - Through to
 - A new orchestrator in front of EOS and CTA
- Similar systems were studied and in particular the IBM Spectrum Archive Solution (GPFS)
- Will now concentrate on putting CTA on the back of EOS

What's next

EOS at the front – CTA hidden from end users



What's next

Modules to be developed

- Archive file and tape catalogue
- Production version of scheduler object store

Functionalities to be developed

- EOS workflow engine to CTA glue
- Repack
- Tape verification
- EOS to CTA reconciliation engine
- Operations scripts and procedures

What's next

Migration strategy from CASTOR to CTA

- CTA uses the same tape format as CASTOR
- No need to transfer data
- Only need to transfer metadata
- Many possibilities still under discussion
 - Transfer one experiment at a time
 - Transfer one tape pool at a time
 - Transfer one tape at a time
 - Transfer based on the namespace