CTA: CERN Tape Archive
Overview and architecture

Vlado Bahyl, Germán Cancio, Eric Cano, Michael Davis, Cristina Moraru, Steven Murray

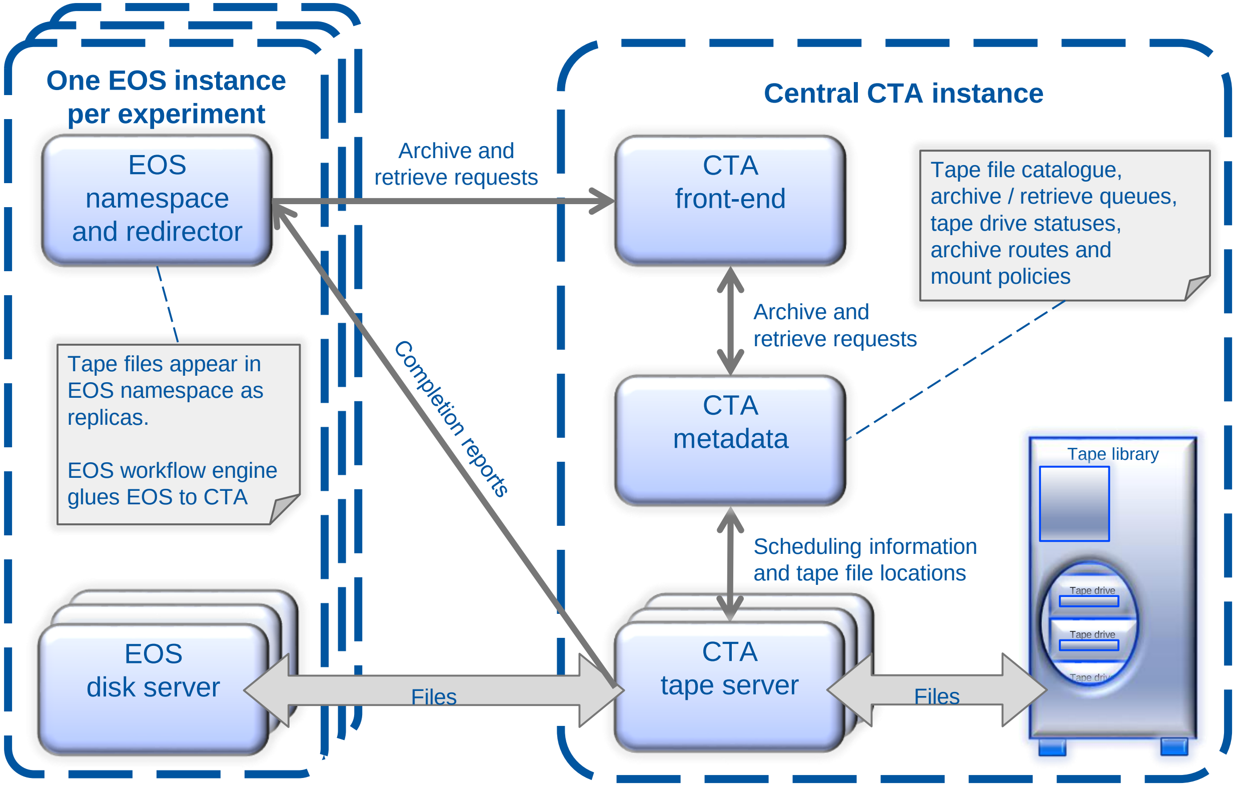
Overview

- What is CTA
- Why CTA
- When is CTA
- Architecture details
- Software architecture
- Summary

What is CTA – 1

CTA is:

- **Natural evolution of CASTOR**
- **A tape backend for EOS**
- **A preemptive tape drive scheduler**
- **A clean separation between disk and tape**


One EOS instance per experiment
EOS namespace and redirector
Tape files appear in EOS namespace as replicas.
EOS workflow engine glues EOS to CTA
EOS disk server
Archive and retrieve requests
Completion reports
Files
Central CTA instance
CTA front-end
Tape file catalogue, archive / retrieve queues, tape drive statuses, archive routes and mount policies
Archive and retrieve requests
CTA metadata
Scheduling information and tape file locations
CTA tape server
Files
Tape library
Tape drive
Tape drive
Tape drive

What is CTA – 2

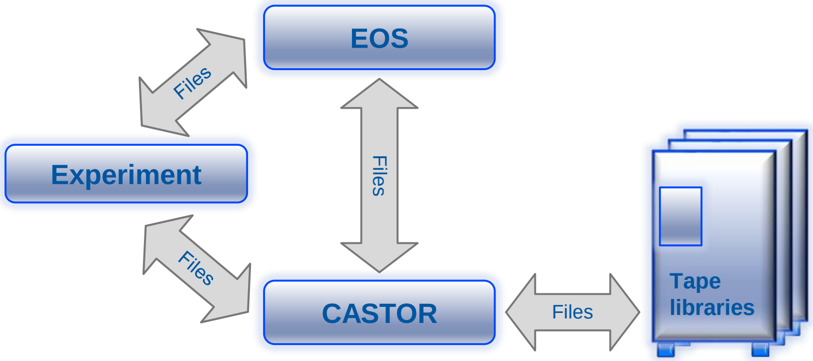
- EOS owns the name space
  - CTA provides an extra replica to disk
    - Reference to replica stored in EOS namespace
    - CTA files only have numeric Ids
- EOS calls CTA hooks on events
  - Using EOS work flow engine
  - Close after write (archive trigger)
  - Open for read with no replica on disk and prepare (retrieve trigger)
  - Delete
- EOS receives callbacks from CTA
  - Archive completion report
  - Listing of contents for reconciliation
- EOS manages the lifecycle of disk copies (garbage collection)
- CTA stores backup of EOS metadata for tape-backed files
- One CTA system serves several EOS instances

- Other disk system should provide the same hooks and callbacks

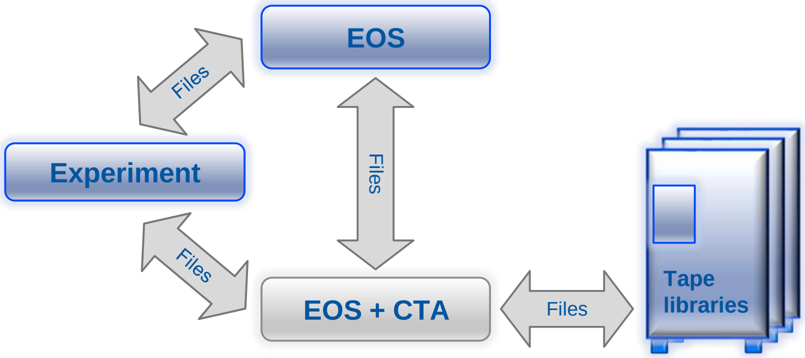
What is CTA – 3

- **EOS plus CTA is a “drop in” replacement for CASTOR**

Current deployments with CASTOR



Future deployments with EOS plus CTA



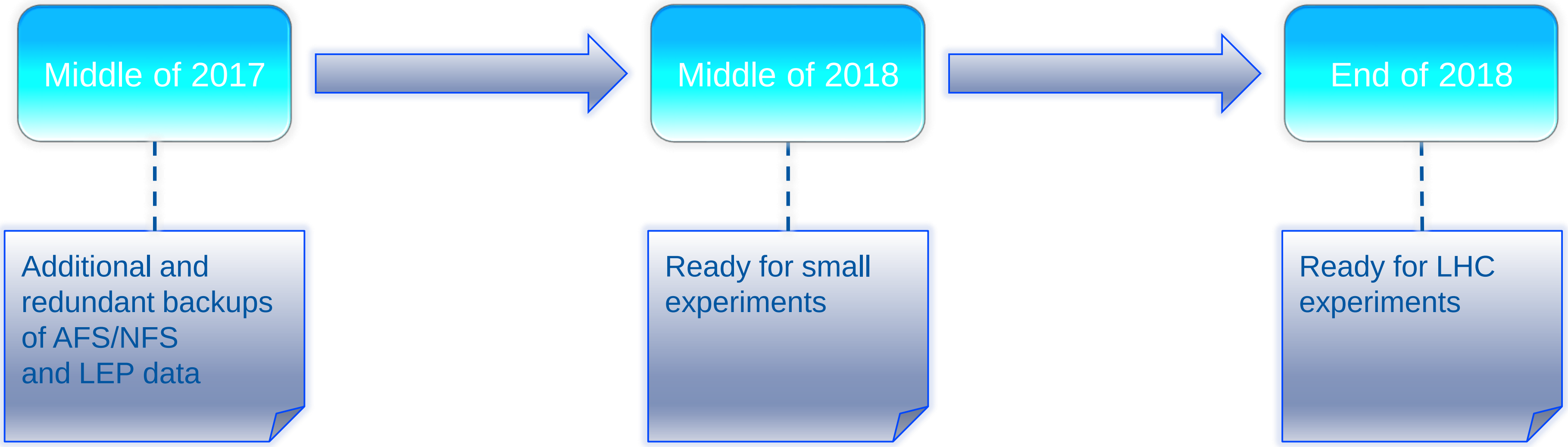
Why CTA – 1

- EOS has become the de facto disk storage for LHC physics data
- Natural evolution from CASTOR
  - Remove duplication between CASTOR disk storage and EOS
  - Thin layer on top of existing CASTOR tape server
  - Stronger and more decoupled separation between disk and tape

Why CTA – 2

- **CTA preemptive scheduler**
  - **Use drives at full speed all of the time**
  - **Single step scheduling vs the multi step scheduling of CASTOR with partial information**
- **Same tape format as CASTOR – only need to migrate metadata**
- **Full flat catalogue of all tape files can be used for disaster recovery**
- **Less networked components than CASTOR (no CUPV, VDQM or VMGR)**

When is CTA



Middle of 2017

Additional and redundant backups of AFS/NFS and LEP data

Middle of 2018

Ready for small experiments

End of 2018

Ready for LHC experiments

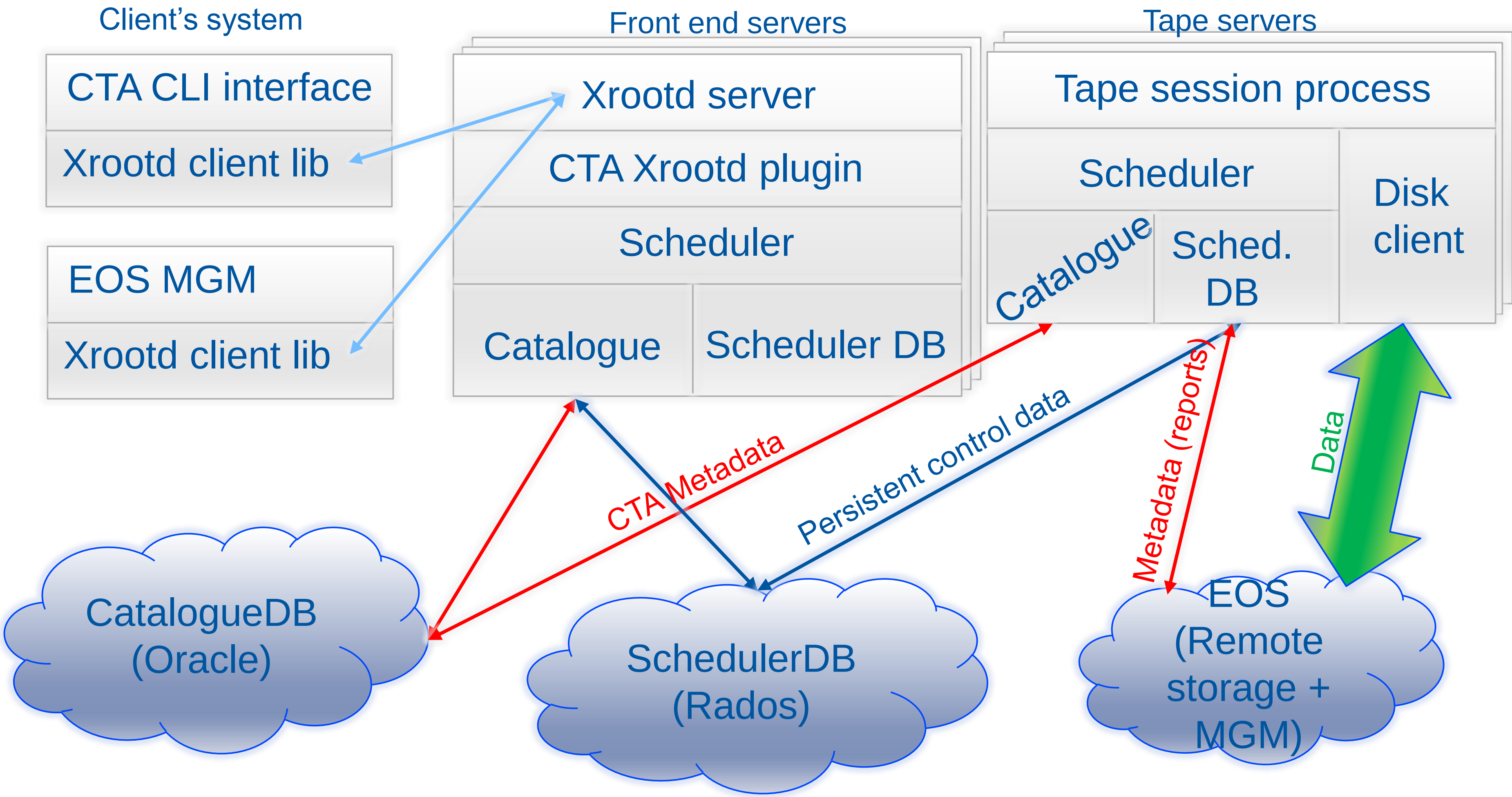
Volunteer experiments to test are more than welcome!

CTA and CASTOR use the same tape format.
Only metadata will need to be migrated.
No files will need to be copied between tapes.

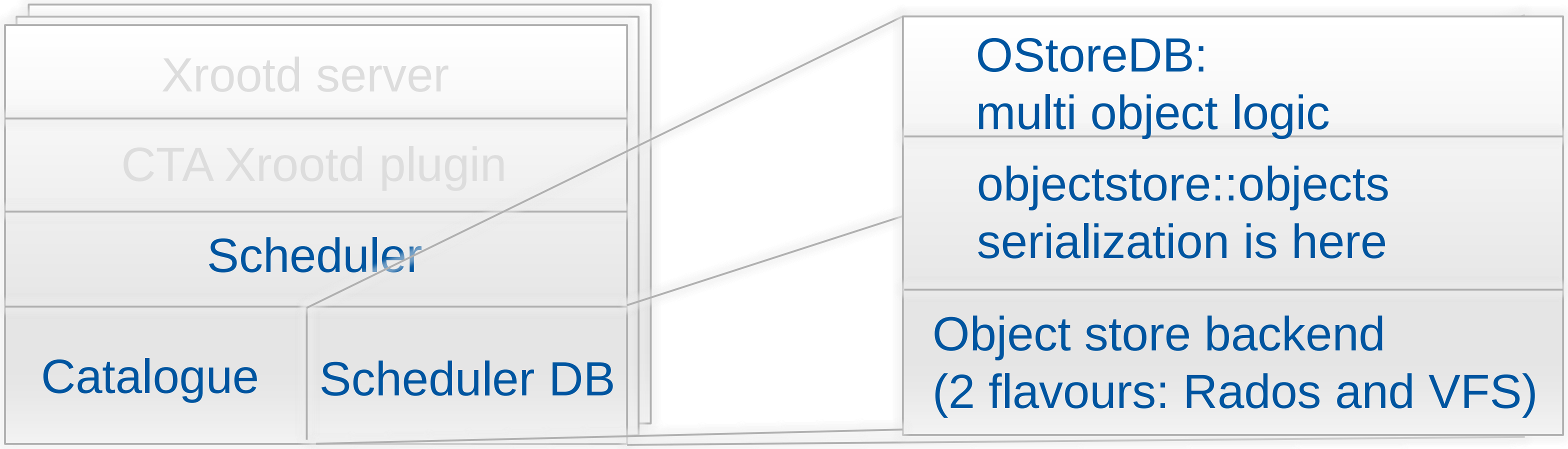
Architecture details

- Shared storage concept
  - Only 2 daemons: front end for CLI & disk system interface (xrootd based) & cta-taped.
- New queueing system
  - Based on Ceph (Rados)
  - Only for transient data
  - Each queue (per tape/tapepool) is an independent object
    - Avoids a single huge queue table
  - Allows storage of rich objects
- Separate file catalogue
  - Based on usual relational DB
  - For persistent data
  - Can be replaced by another implementation
- cta-taped an adapted tapeserverd from CASTOR
  - Multiple data transfer protocols already present (xroot (2 flavors), local file, Rados striper, (rfio in CASTOR))
  - URL based selection, file-by-file granularity

Software architecture



CTA scheduler DB & Catalogue



- OStoreDB is an implementation of the Scheduler DB interface
- Object store based
  - Can leverage rados already running in our group (and its scaling)
  - Can be implemented over a simple filesystem (testing, small scale system)
  - Relies on lock, full reads and writes, and delete
  - Serialization with Google protocol buffers
  - Multithreading and asynchronous IO to achieve performance
- Catalogue has a similar layout
  - See Steve's presentation