Putting Ceph at the Heart of CASTOR

Sébastien Ponce
sebastien.ponce@cern.ch

CERN

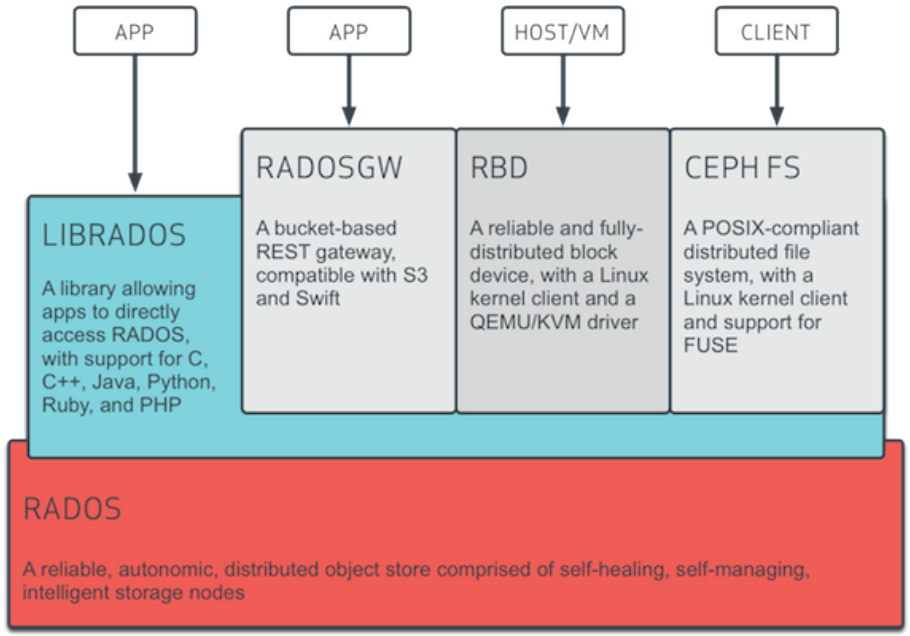
May 19th 2014

Outline

Introducing Ceph

What does ceph provide ?

- LibRados
  - Object store
  - No striping
- RadosGW
  - S3 compatible
- RBD
  - Block device
- Ceph FS
  - Filesystem thus kernel code)
  - Beta state



What do we need ?

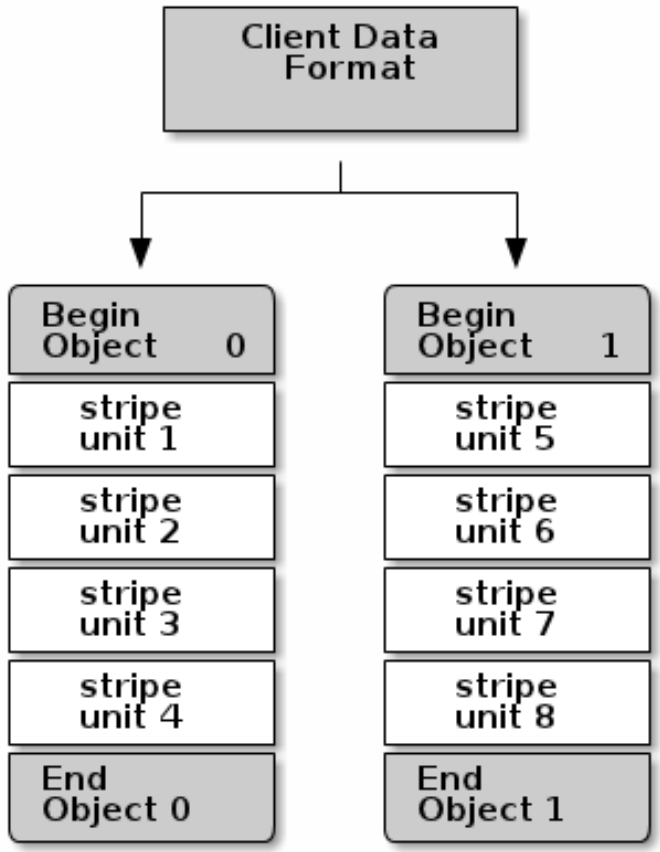
- Storing files
  - Any size (objects should be small)
  - Having external attributes
- (tunable) reliability
  - Number of replicas
  - Erasure coding
- Performance
  - Striping of files
  - Scalability
- Easiness of operation
  - Rebalancing, draining, easy management

Best candidate is librados

- Has most things we need :
  - Scalability
  - External attributes
  - Number of replicas tunable per object
  - Erasure coding is under work
  - Rebalancing, draining, easy management
- Is missing two points only :
  - Striping
  - Support for big files
    - Objects should be rather small (1-100 MB max)

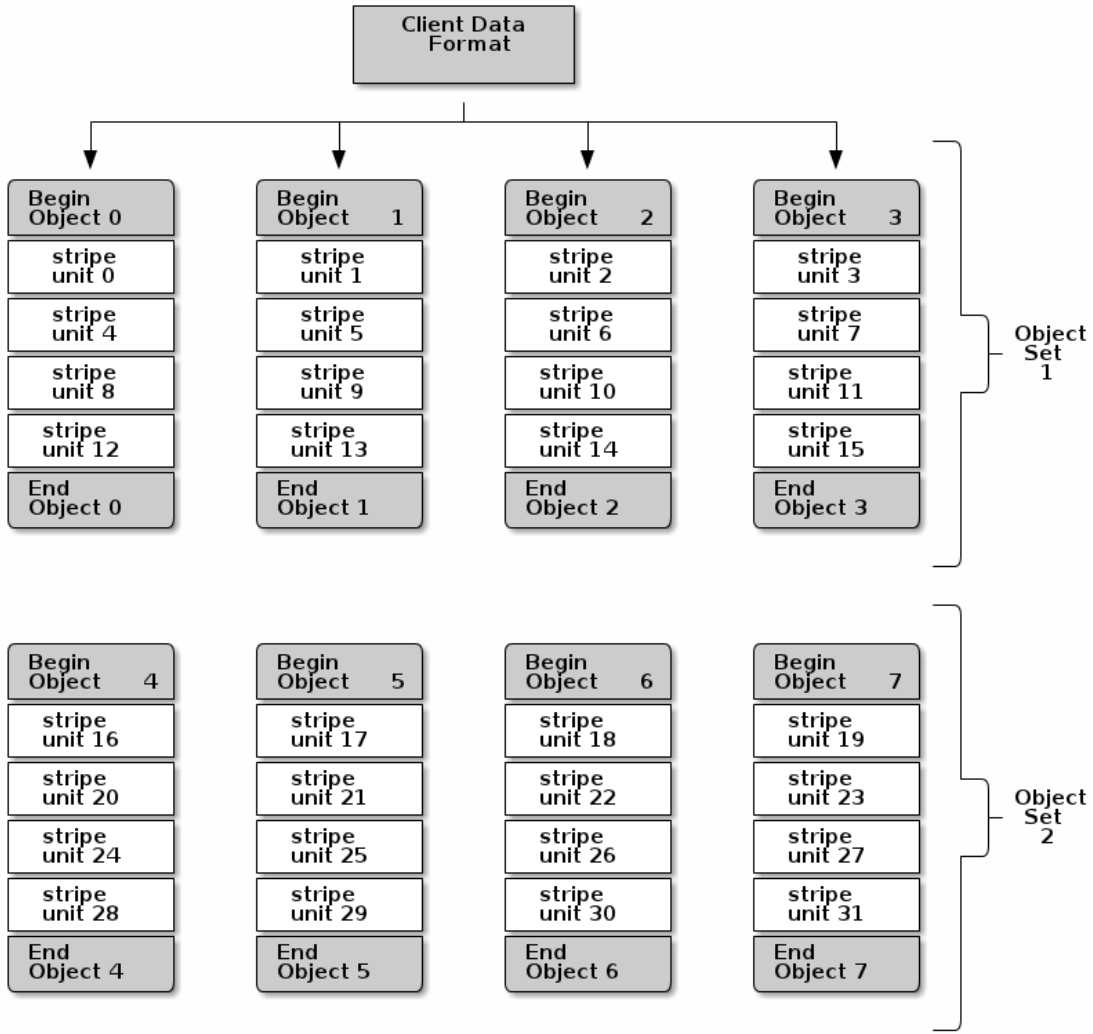
Striping can be added to ceph

- File is mapped to a set of objects
- Objects' names are <filename >_ <nb >
- Excellent performances



Big files builds on striping

- Extension of striping with the notion of object sets
- Implemented in a very generic fashion in my libradosext
- To be backported to main ceph ?



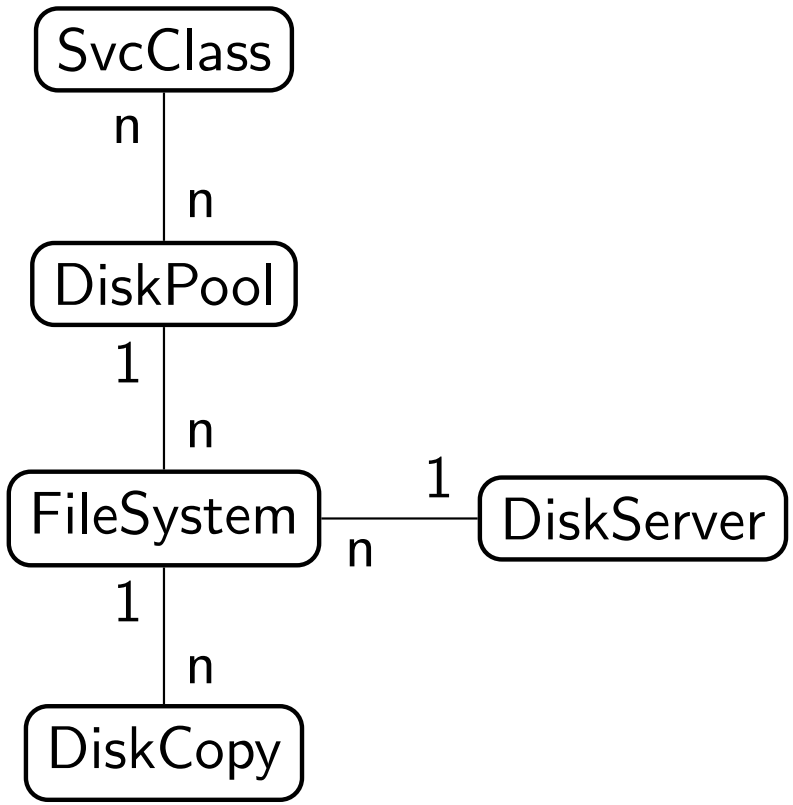
Status of striping

- striping has been implemented inside the Ceph code base
- and crontributed to the project
- ceph now comes with libradosstriper.so
- next release (giant) will expose it

Introducing DataPools in CASTOR

Evolution of the stager DB schema

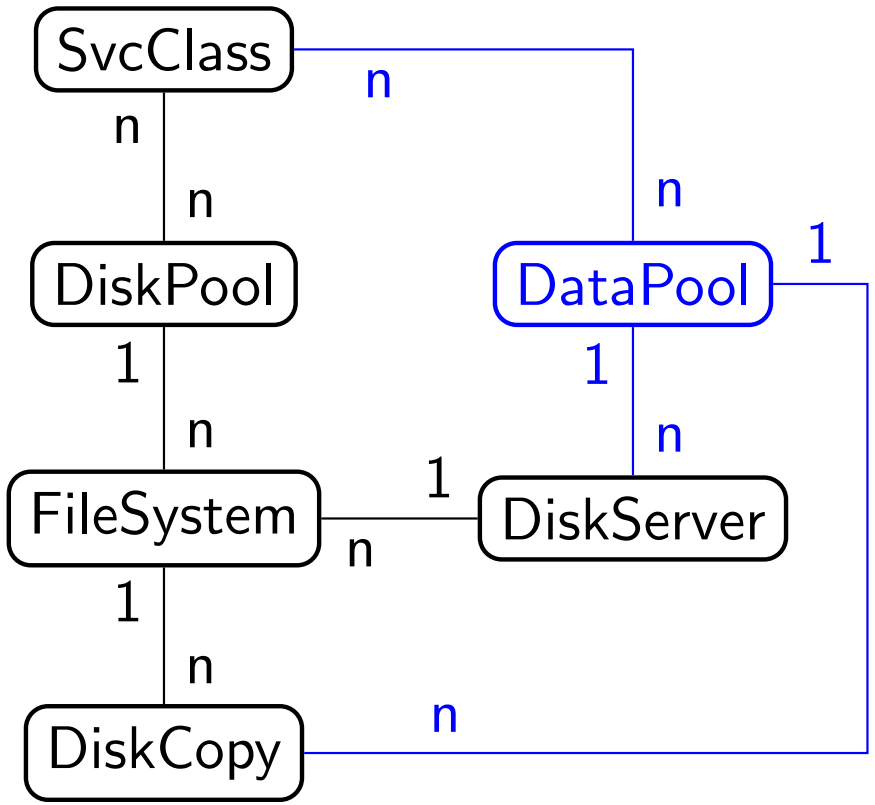
- DiskPools are a set of FileSystems
- Each FileSystem belong to a DiskPool
- DiskCopies reside in FileSystems



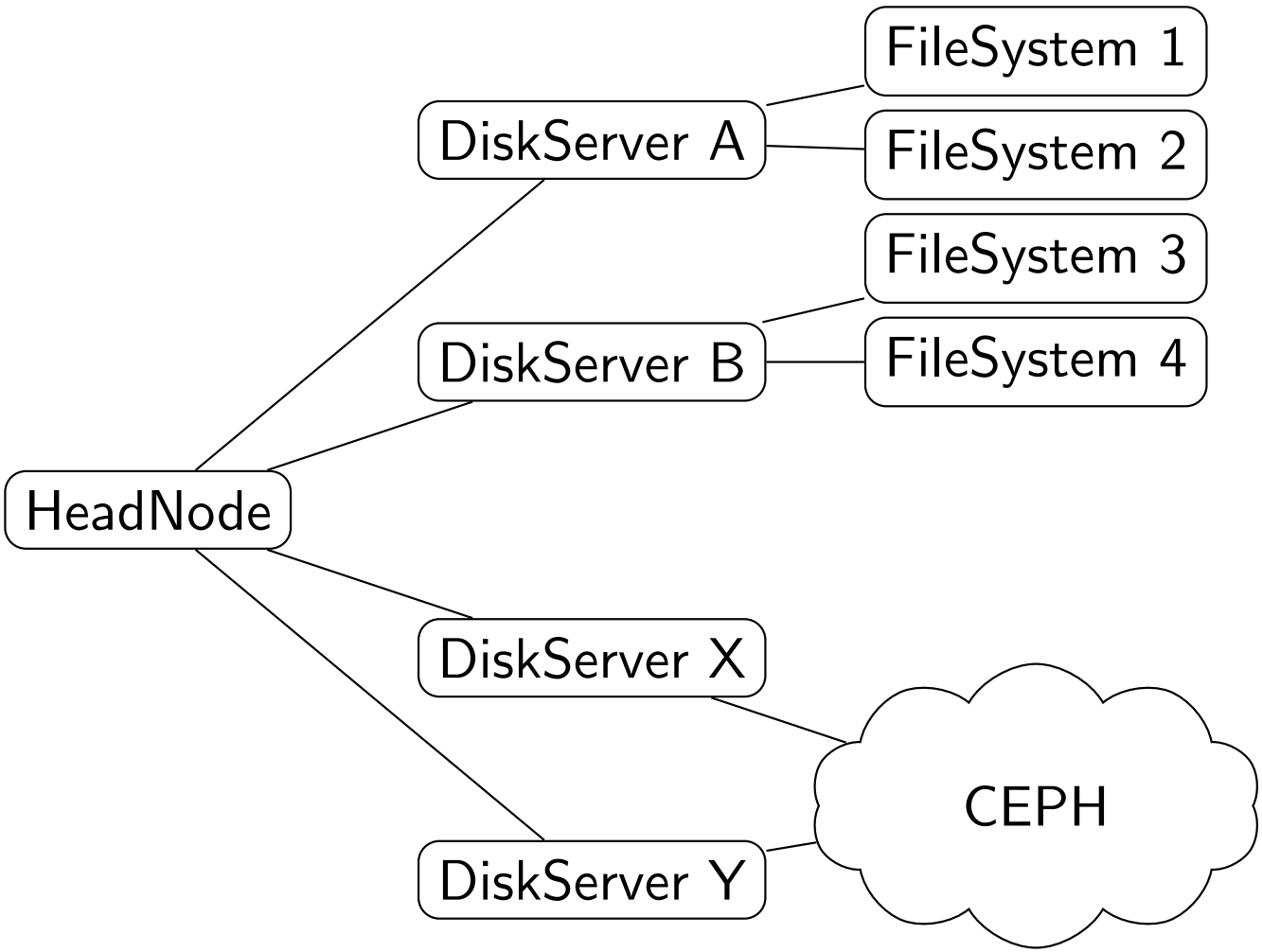
Evolution of the stager DB schema

- DiskPools are a set of FileSystems
- Each FileSystem belong to a DiskPool
- DiskCopies reside in FileSystems

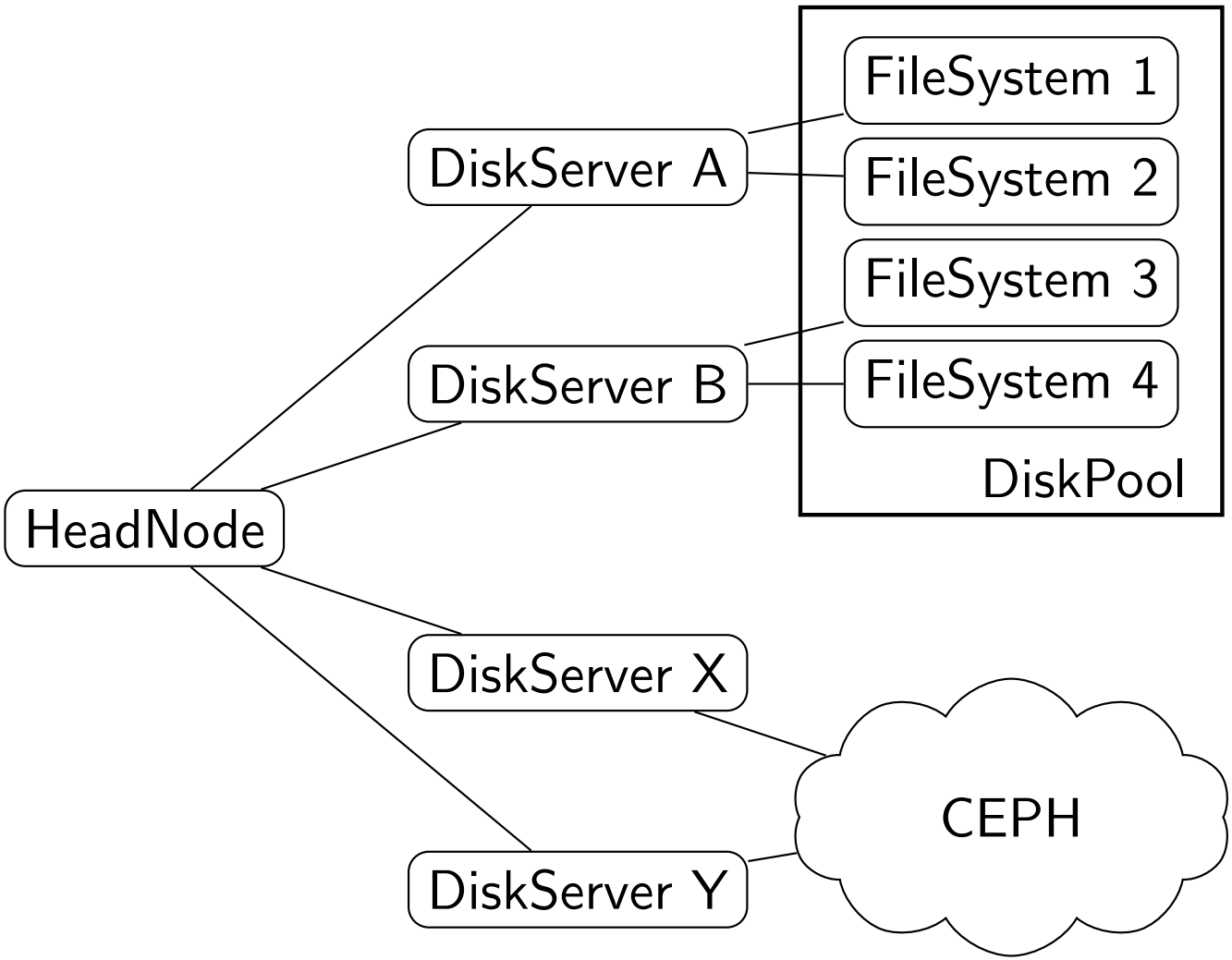
- DataPools are independent entities
- Each DiskServer serves a given DataPool
- DiskCopies reside in the DataPool



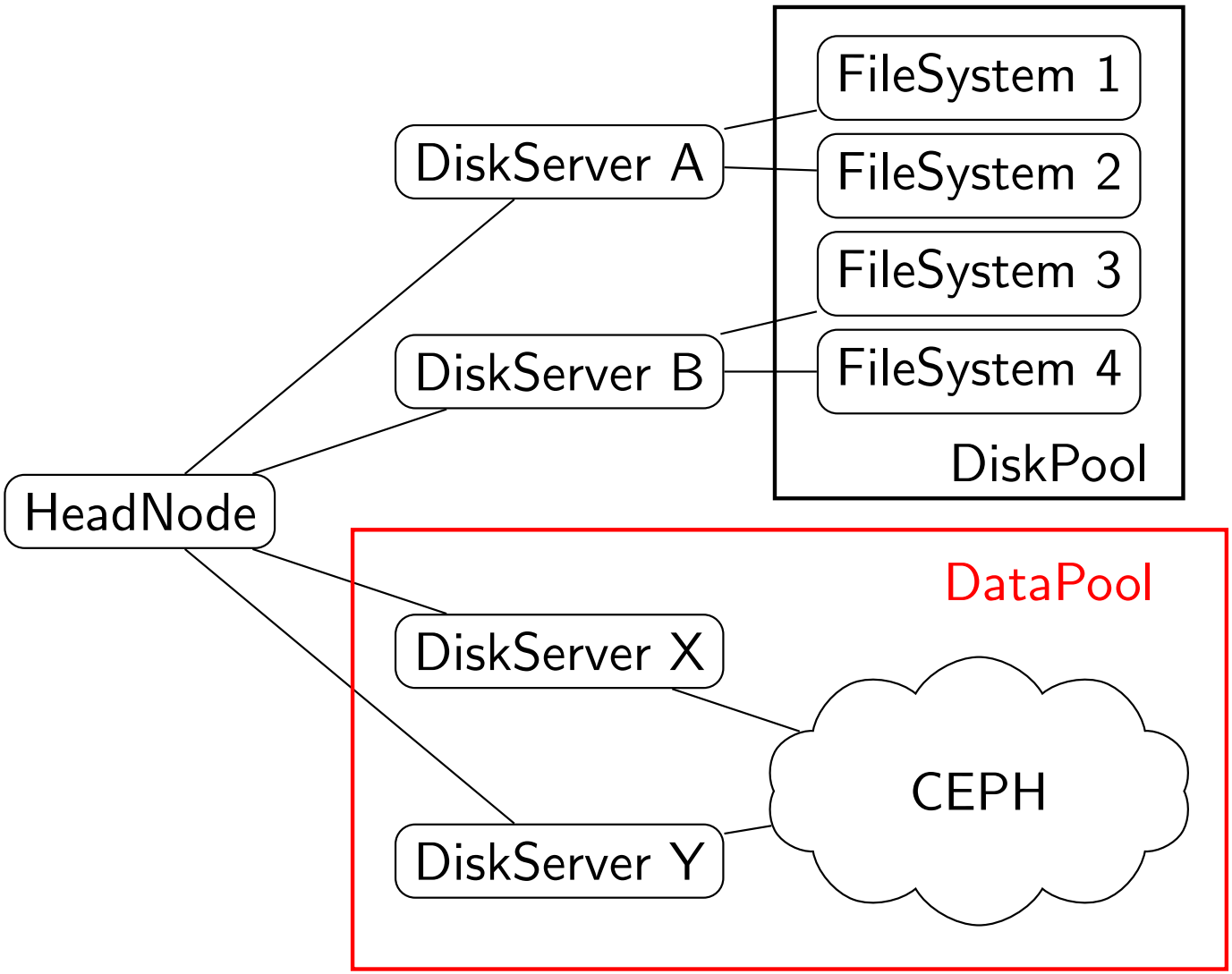
Practically



Practically



Practically



Not pecific to Ceph !

Note that the DataPool concept is generic

- So far, nothing is ceph specific
- We could as well use this with another backend
  - any object store
  - any shared filesystem

Modifying CASTOR for Ceph

What needs to be adapted to Ceph

Has to be adapted

- protocols
  - root is discontinued
  - rfiod and gridFTP need modifications
  - xrootd can use an OSS plugin
- GC
- synchronization

What needs to be adapted to Ceph

Has to be adapted

- protocols
  - root is discontinued
  - rfiod and gridFTP need modifications
  - xrootd can use an OSS plugin
- GC
- synchronization

Do no change

- request handler, stager
- PL/SQL code
- scheduling (transfermanager and diskmanager)
- tape transfers
- disk to disk copy

Adaptation of protocols

The principle

- identify POSIX calls
- replace them with an if leading to POSIX or CEPH call

Adaptation of protocols

The principle

- identify POSIX calls
- replace them with an if leading to POSIX or CEPH call

In practice for rfio/gridFTP

- a small library with generic_open/write/read/close functions
- sed of open/read/write/close with call to generic version

Adaptation of protocols

The principle

- identify POSIX calls
- replace them with an if leading to POSIX or CEPH call

In practice for rfio/gridFTP

- a small library with generic_open/write/read/close functions
- sed of open/read/write/close with call to generic version

In practice for xroot

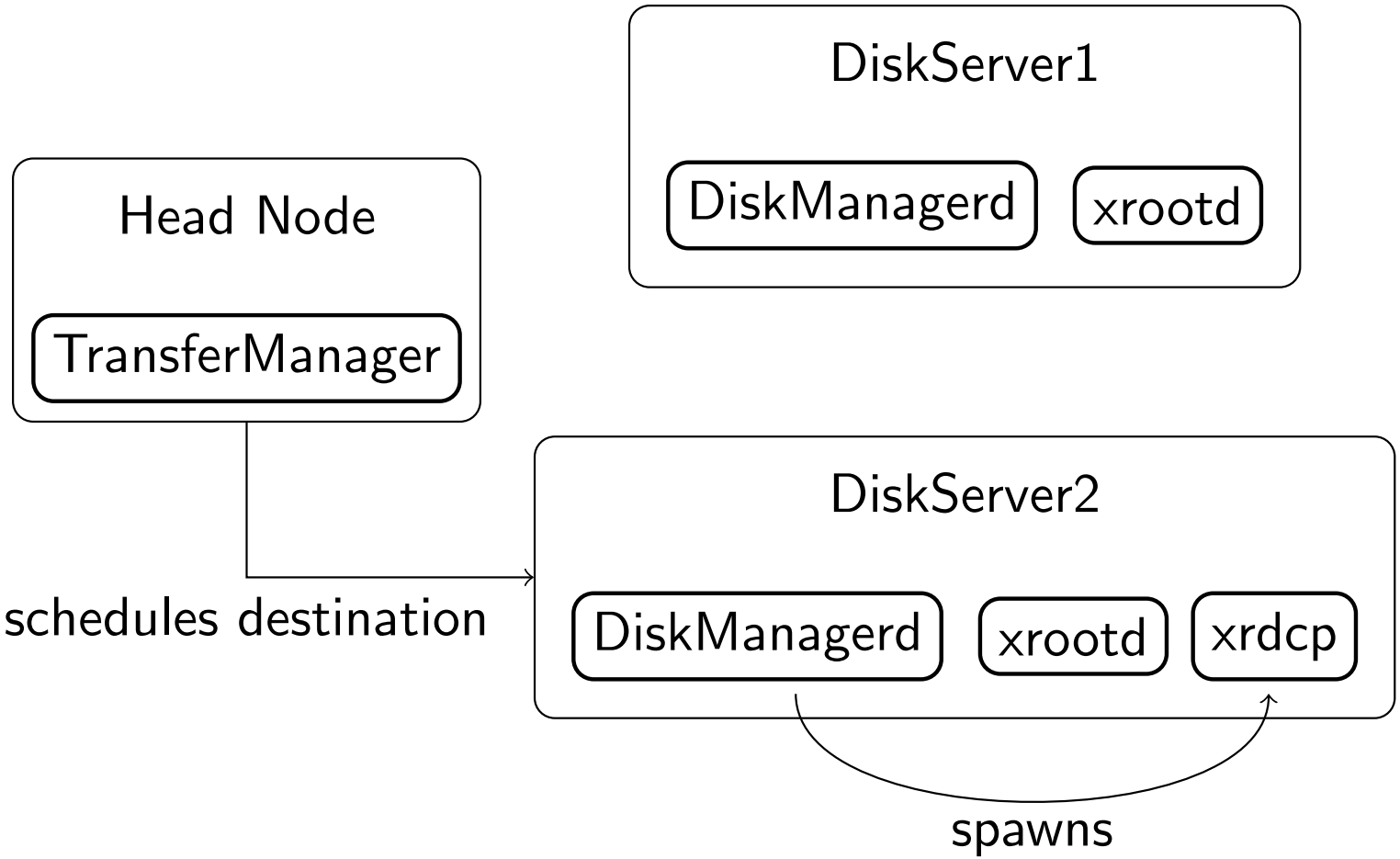
- write an OSS plugin
- basically, same code as the previous library

Disk2 disk copies with ceph

- in principle rfcp/xrdcp should be adapted and talk ceph
- but xroot can use its local daemon and its OSS plugin

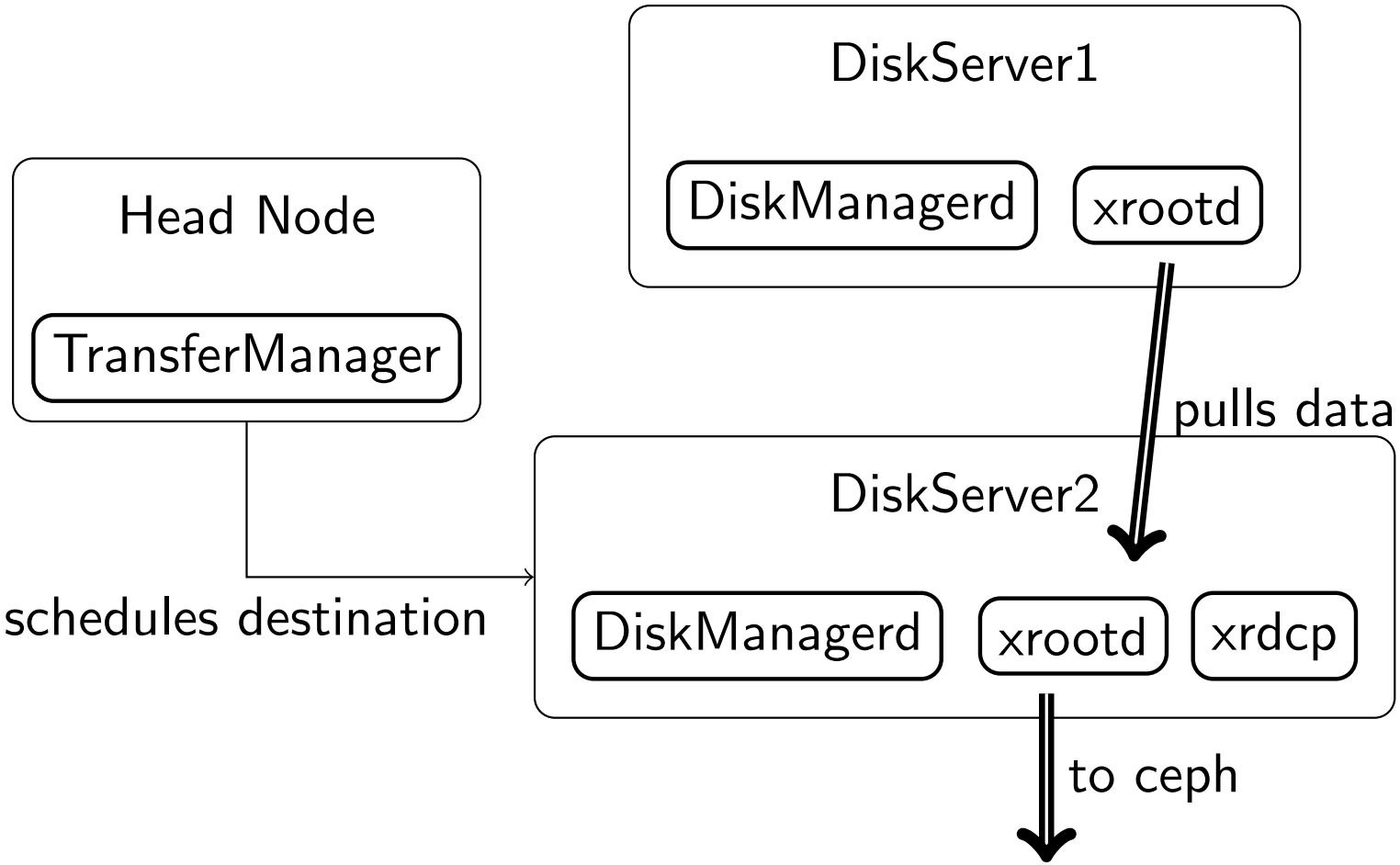
Disk2 disk copies with ceph

- in principle rfcp/xrdcp should be adapted and talk ceph
- but xroot can use its local daemon and its OSS plugin



Disk2 disk copies with ceph

- in principle rfcp/xrdcp should be adapted and talk ceph
- but xroot can use its local daemon and its OSS plugin

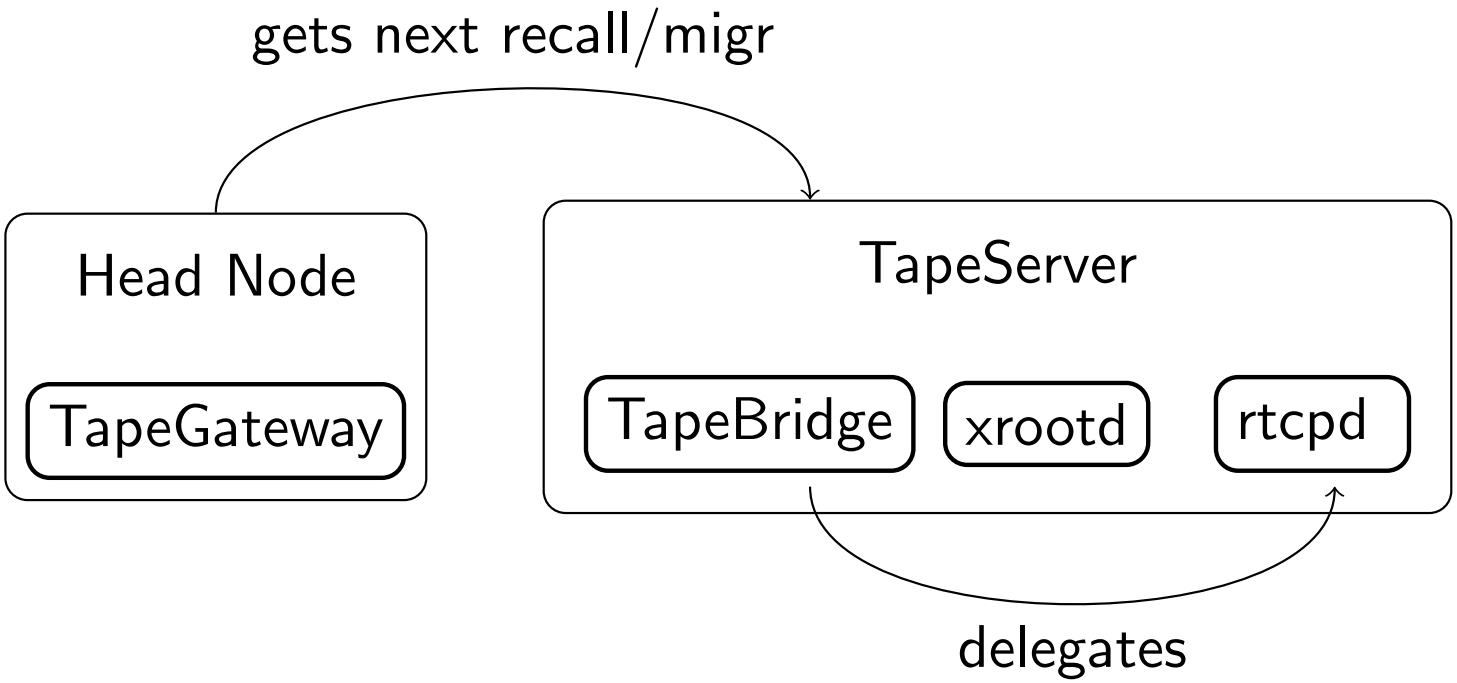


Tape transfers with ceph

- Same situation as for disk to disk copies
- and we run an xroot daemon on the tapeservers

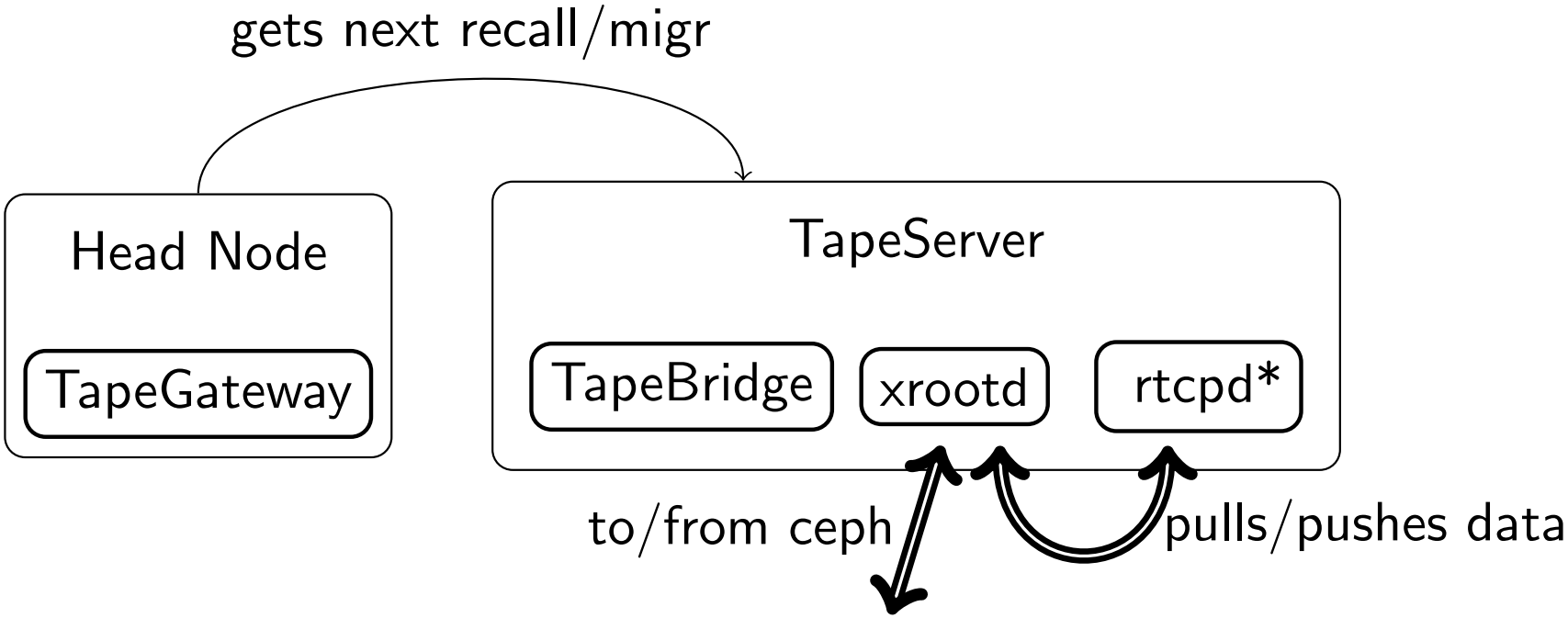
Tape transfers with ceph

- Same situation as for disk to disk copies
- and we run an xroot daemon on the tapeservers



Tape transfers with ceph

- Same situation as for disk to disk copies
- and we run an xroot daemon on the tapeservers



Adaptation of GC and synchronization

GC

- same approach as for protocols
- replace POSIX calls (here rm/stat) with generic version

Synchronization

- need to adapt file listing
- and then same approach as GC/protocols

Adaptation of GC and synchronization

GC

- same approach as for protocols
- replace POSIX calls (here rm/stat) with generic version

Synchronization

- need to adapt file listing
- and then same approach as GC/protocols

Where should they run for DataPools ?

- on any node declaring the pool in the DiskManager/DataPool
- these nodes will fight for GC and sync of their common pool

Current status and deployment

Status right now

- ✓ Striping
  - ✓ Implementation
  - ✓ Merge into Ceph master
    - But will only be released in giant
- ✓ DataPool introduction
  - ✓ DB schema adaptation
  - ✓ PL/SQL changes
  - ✓ Admin tools modifications
- × Protocols
  - ✓ rfio
  - × gridFTP
  - × xroot OSS plugin
- ✓ GC and synchronization to be done

Deployment

Certitudes

- 2.1.15 will have xroot for internal transfers and ceph enabled
- But LHC production should start without DataPools/ceph
- Major repack campaign should finish without DataPools/ceph
- External institutes should not play with it and real data

Deployment

Certitudes

- 2.1.15 will have xroot for internal transfers and ceph enabled
- But LHC production should start without DataPools/ceph
- Major repack campaign should finish without DataPools/ceph
- External institutes should not play with it and real data

Proposals

- Heavy testing to be done on ITDC/pps for the stability/deployment of CEPH itself
- Then we could test it on non critical repacks
- Before a test production pool can be created (2015)
  - "use at your own risk SLA"