dCache ⟺ CTA integration

17th *International dCache workshop*

What is DESY (as storage provider)

	Service
EuXFEL, Petra-III, ILC, Accelerator R&D, ...	Primary data site (Tier-0). Provides online, nearline and archival storage.
Belle-II, ...	Provides online and near-line storage (Tier-1).
Atlas, CMS, LHCb	Online only (Tier-2).
H1, Hermes, Hera-B, Zeus , ...	Provides online and archival storage (Long-term preservation).
User and Services	Classic Backup for disaster recovery

Tape – Pros and Cons.

- Low cost 7.5€ 1TB
- Low power consumption
- High capacity (20 TB, LTO9)
- High IO bandwidth 300 MB/s
- Air-gap: users can't delete or modify data on tapes
- High durability 15-30 years

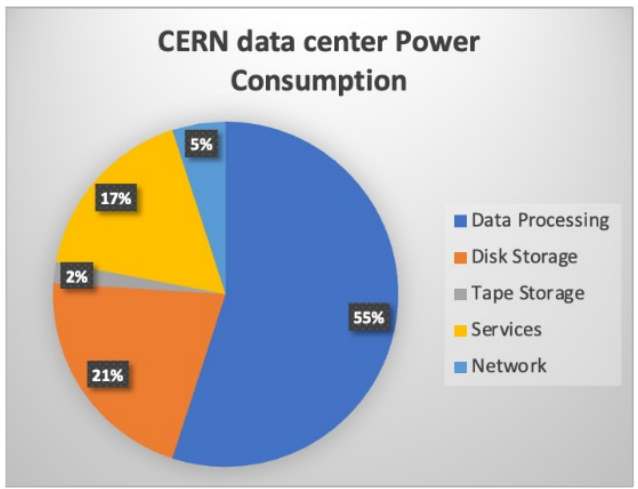
- High latency until first byte ~ 90s!
- Only one IO stream stream at any point of time

WLCG data centers power consumption

The pie chart shows the breakdown of the power consumption at the CERN data center

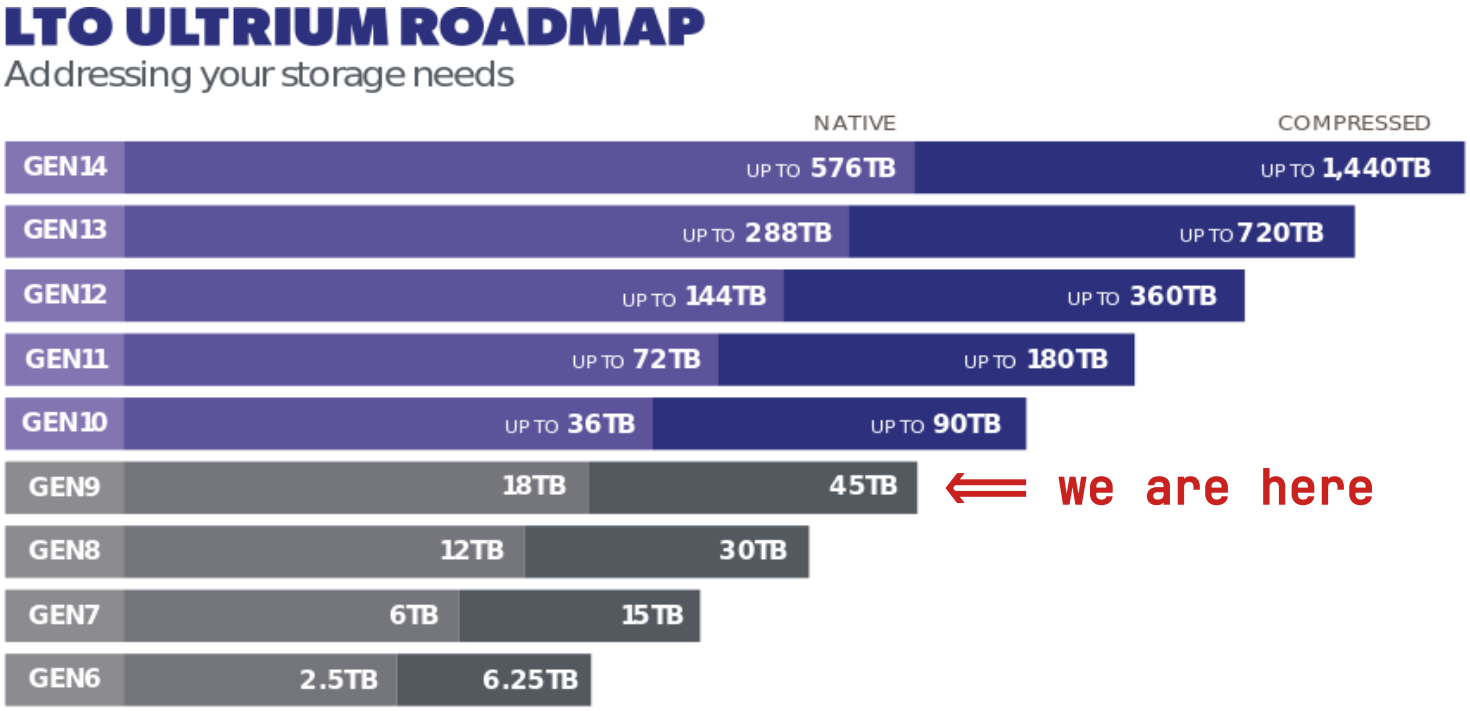
Most of the power is consumed for data processing (CPUs). Large part of the "services" are in fact CPUs

In this study we will focus on the energy needs for CPUs



Shameless stolen from Simone Campana

Tape Technologies



LTO – Linear Tape Open

The IBM drives expected to provide a comparable (better) numbers. The official information is not available yet.

Multiple Faces of Tape

At Tier-0

- High data ingest rate
- Multiple parallel streams
- High durability, multiple copies on different media
- Long-term nearline access
- Small file handling

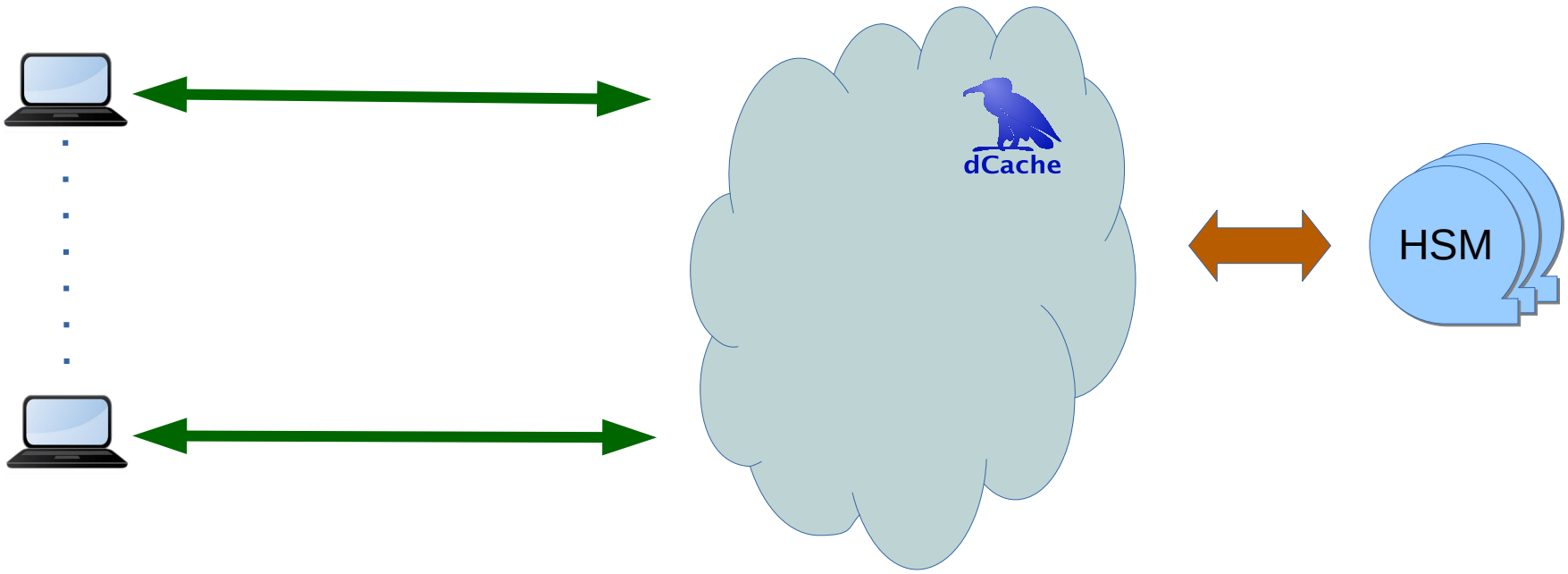
At analysis facility

- Automatic data migration
- Bulk recall on periodic basis
- Long-term nearline access
- Recall prioritization

Data Archive

- Manual data migration
- Long-term preservation
- Automatic technology migration
- Self-healing

dCache+HSM Tandem



All access to scientific data on tapes goes exclusively through dCache!

dCache Tape Connectivity

- Write-back / Read-through cache behavior
- Transparent for the users
- Available via all protocols (subject to authorization)
- Supports multiple HSM on a single instance
- Stores tape location as opaque data provided by HSM

Interfaces to HSM

- Execute external migration script
  - Stupid, Simple, Genius …
  - Reference implementation of driver API
- Plugable driver Java API:
  - Suitable to create efficient HSM connectivity
  - Sapphire - small file aggregation
  - ENDIT *(Efficient Northern dCache Interface to TSM)*

dCache HSM ⟺ Link

- Files belong to storage classes

```
<storage-class>@osm
```

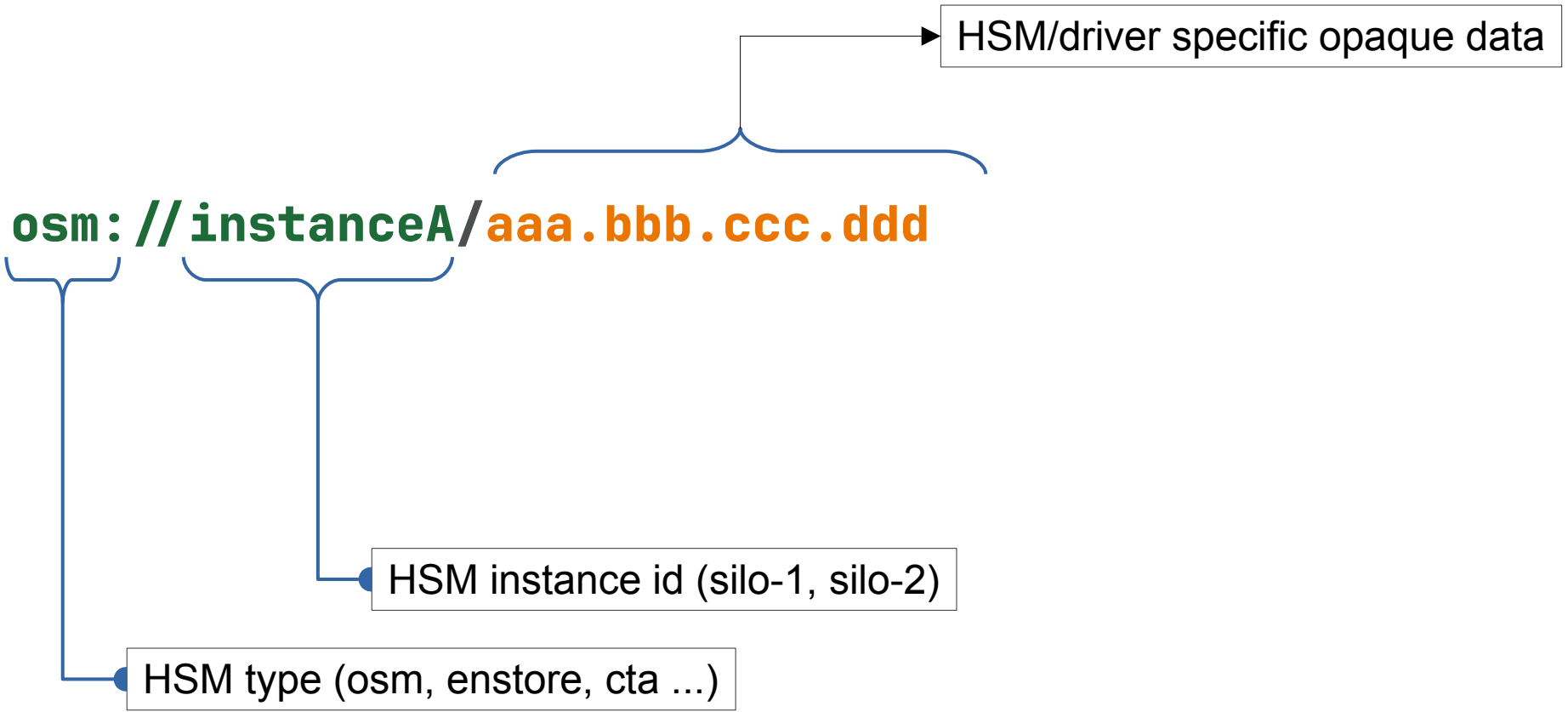
- Configure HSM connectivity on the pool

```
hsm create osm siloA script \
                    -command=hsmcp.py
```

- Tape location stored in namespace as URI

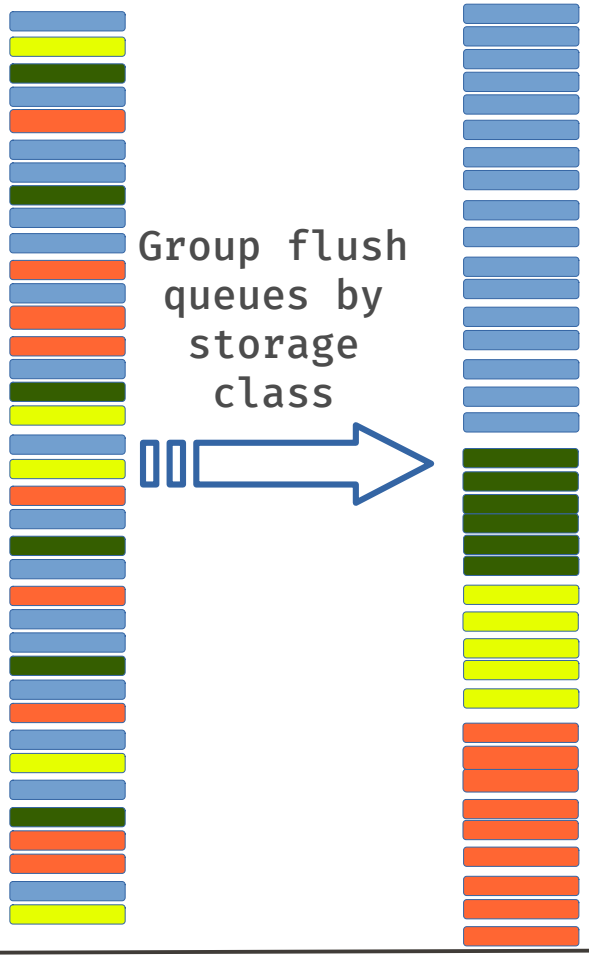
```
osm://siloA/xxxxxxxxxxxx
```

dCache HSM ⟺ Link

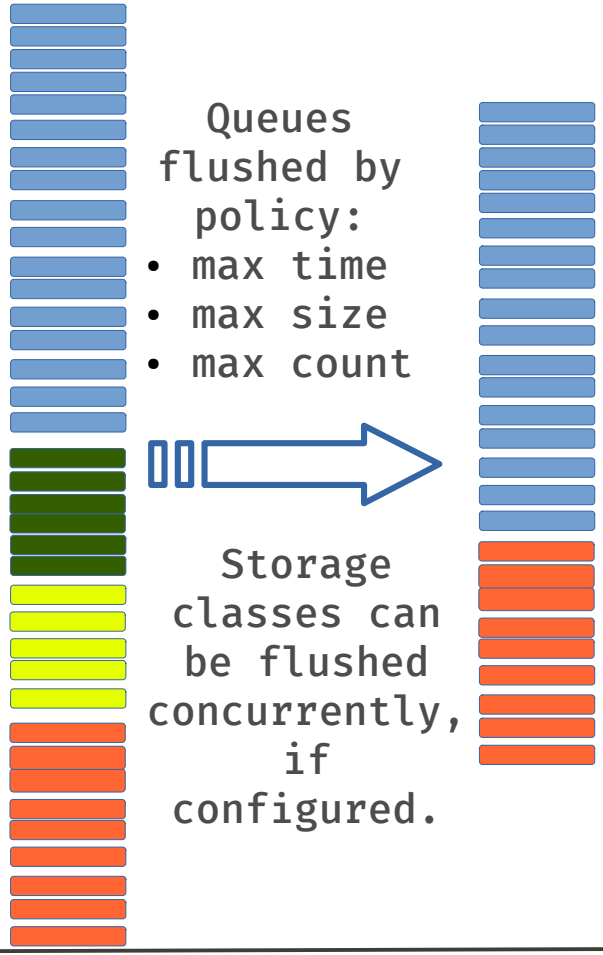


The Flush Queue

The Flush Queue



The Flush Queue

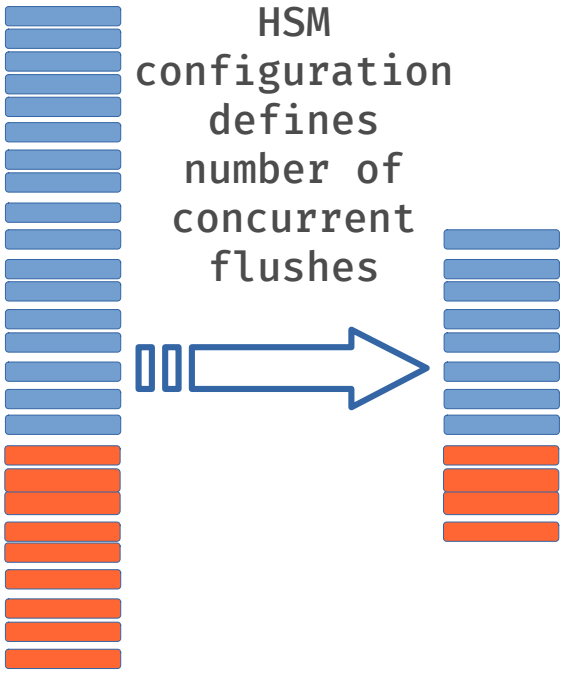


Queues flushed by policy:

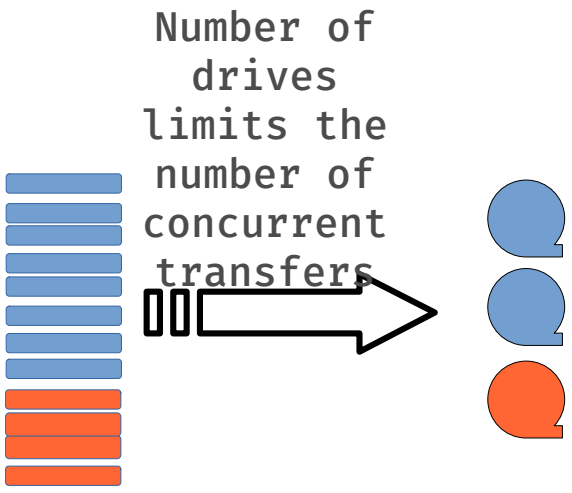
- max time
- max size
- max count

Storage classes can be flushed concurrently, if configured.

The Flush Queue



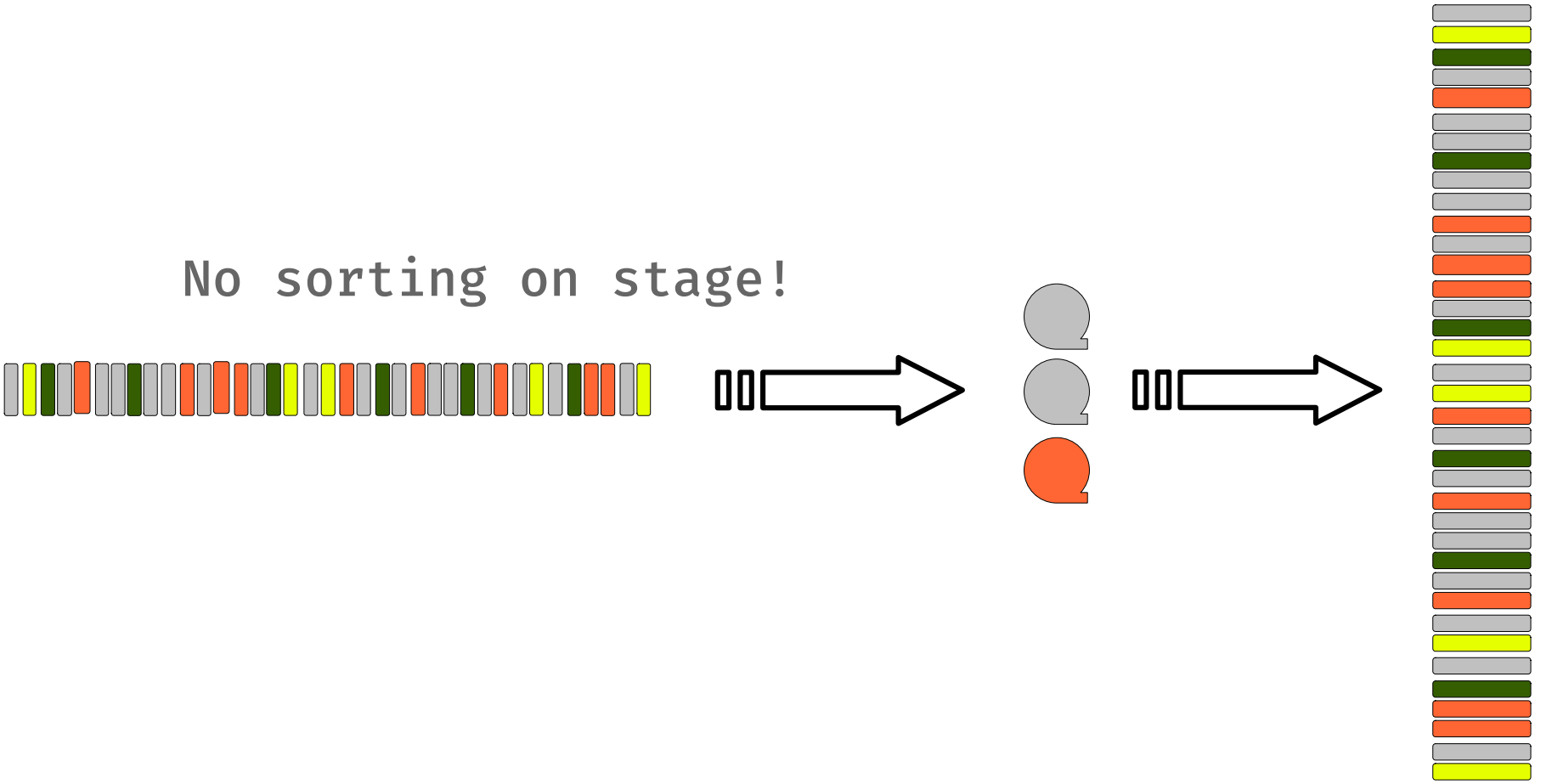
The Flush Queue



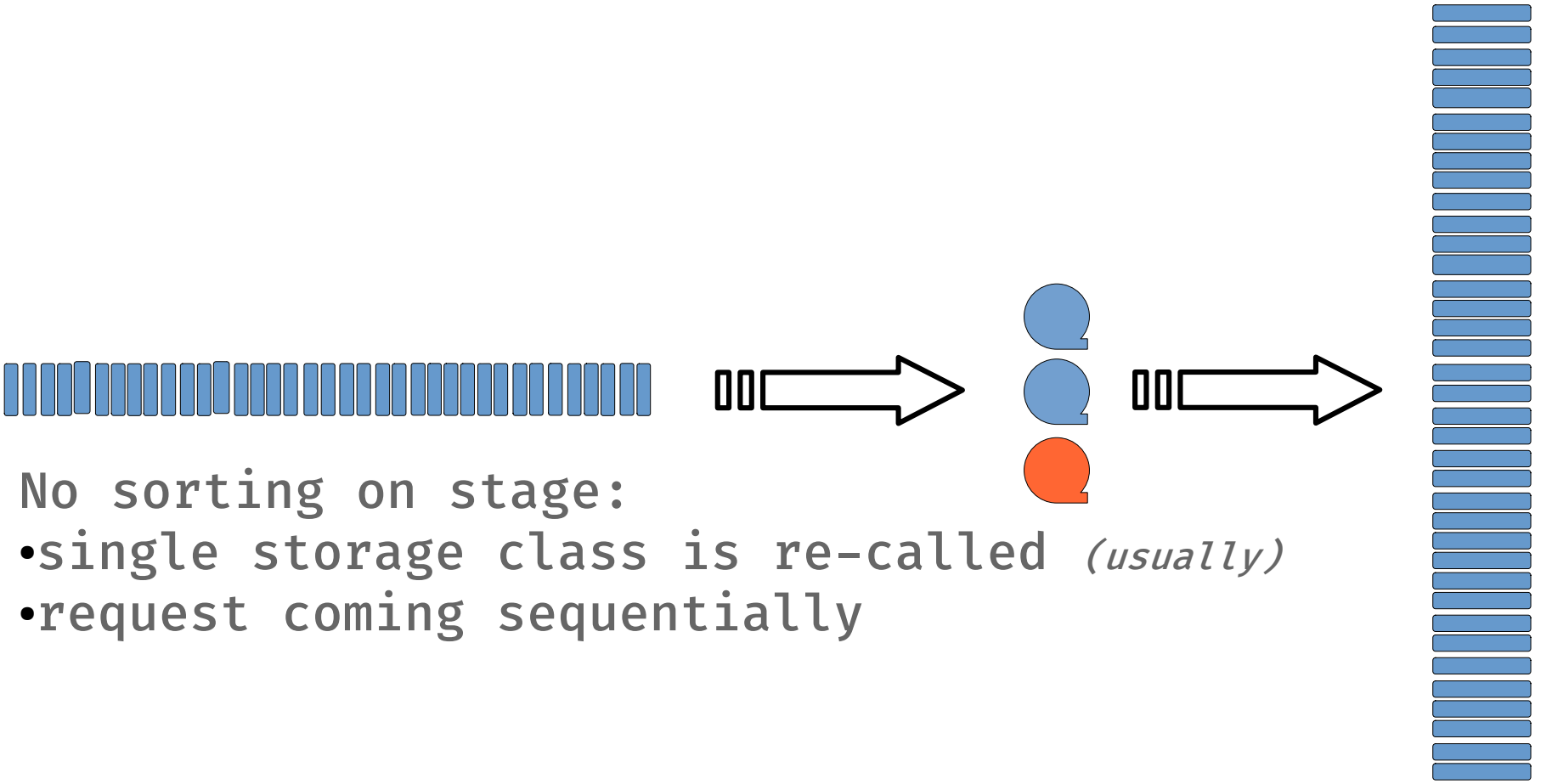
Restore

- Appropriate stage pool is selected
  - HSM type, load, space
- Space allocated on disk
  - **Never block on space allocation when tape is mounted!**
- Requests sent to tape

The Restore Queue



The Restore Queue

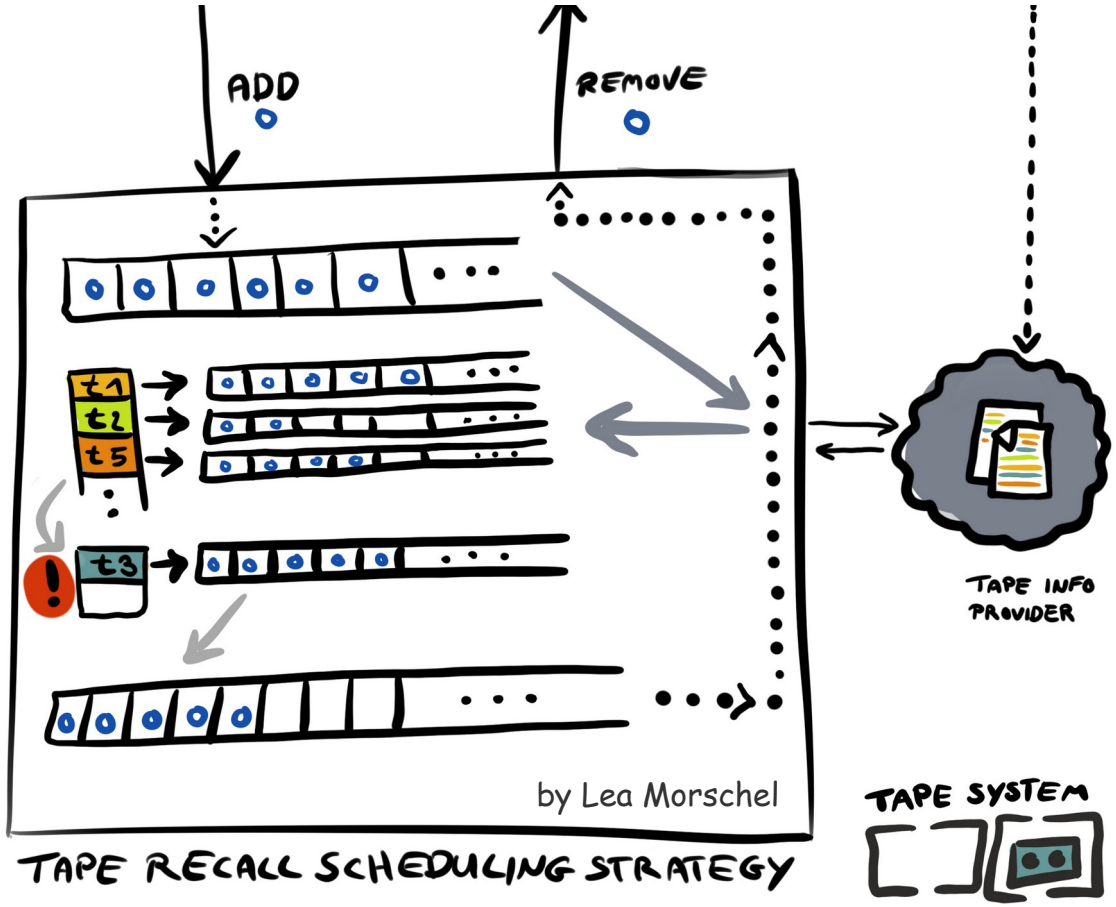


No sorting on stage:

- single storage class is re-called *(usually)*
- request coming sequentially

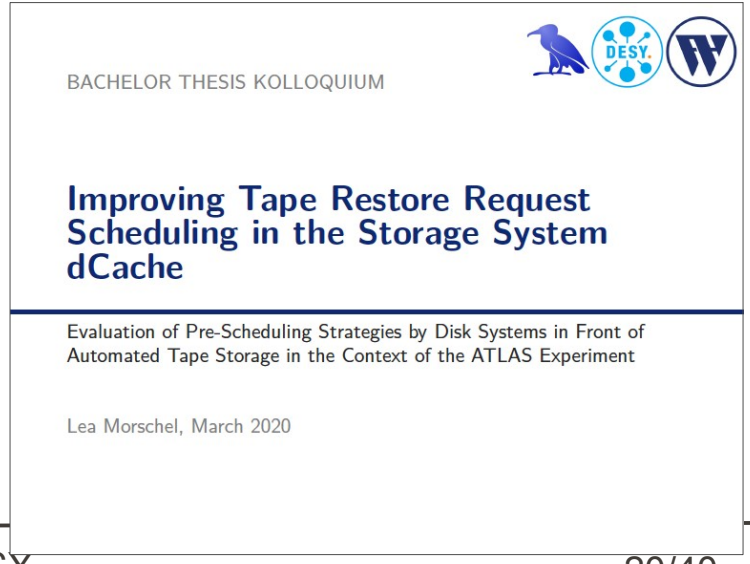
Tape recall grouping

- Group requests by tape
- Recall triggered by
  - Size
  - Max idle time
- Number of parallel recall based on number of tape drives



Morschel-Lueken Rule

Writing or Re-calling 80% of a tape in RAO* hides mount & seek overhead!



* Recommended Access Ordering

Tape Software Requirements

- Maximize tape HW efficiency
  - Integration into DESY ecosystem
  - Integration with dCache tape interface
- Stable operation for a next decade
- Should be Open-source, adopting open standards
- Wide user and technology community

Requirments (by Martin Gasthuber & Co., 2019)

- support full streaming for current and next 2 generation of drives (up to 700MB/s per drive) - assume 30-50 drives
- aggressive request aggregation and prio support - i.e. XFEL ingest @6-10GB/sec - drastically reduce number of mounts/dismounts.
- handle 10 Billion objects today - scale to x100 within 5 yrs.
- daily turnover 1+PB
- automated media migration (incl. aggregation) - 1+PB per day - x10 within next 5 years
- support Enterprise & LTO
- handling of small files
- concept to integrate metadata (i.e. SIRF format)
- reading old OSM media directly
- no HSM logic (decouple user request from tape request)
- manage 2(3) copies own distinct tape sets (+library) - allow on/off switching at any time - deferred copy creation - handle S3 endpoints (public cloud)
- automated (clever) handling of multiple copies in case of errors - incl. repair (generate a working copy again)
- integrity checks - allow (at least) 2 full checks per year - support user level checksums (in addition) - parallel checksum calculation (drive and initiator) while writing
- UI(cli,gui,api) for admin/developer and operating - integration into service monitoring
- dependencies (HW & SW) should be handled and guaranteed on (at least 5 years) contracts/agreements - sub-services (i.e. embedded DB) should be supported on open source solutions
- DESY extensions plugin - allow custom feature extensions and/or functions
- standard/open access protocol (S3, Swift, ... - NOT XRoot) - PUT/GET/REMOVE logic
- state of the art integration/use development tools (CI/CD)

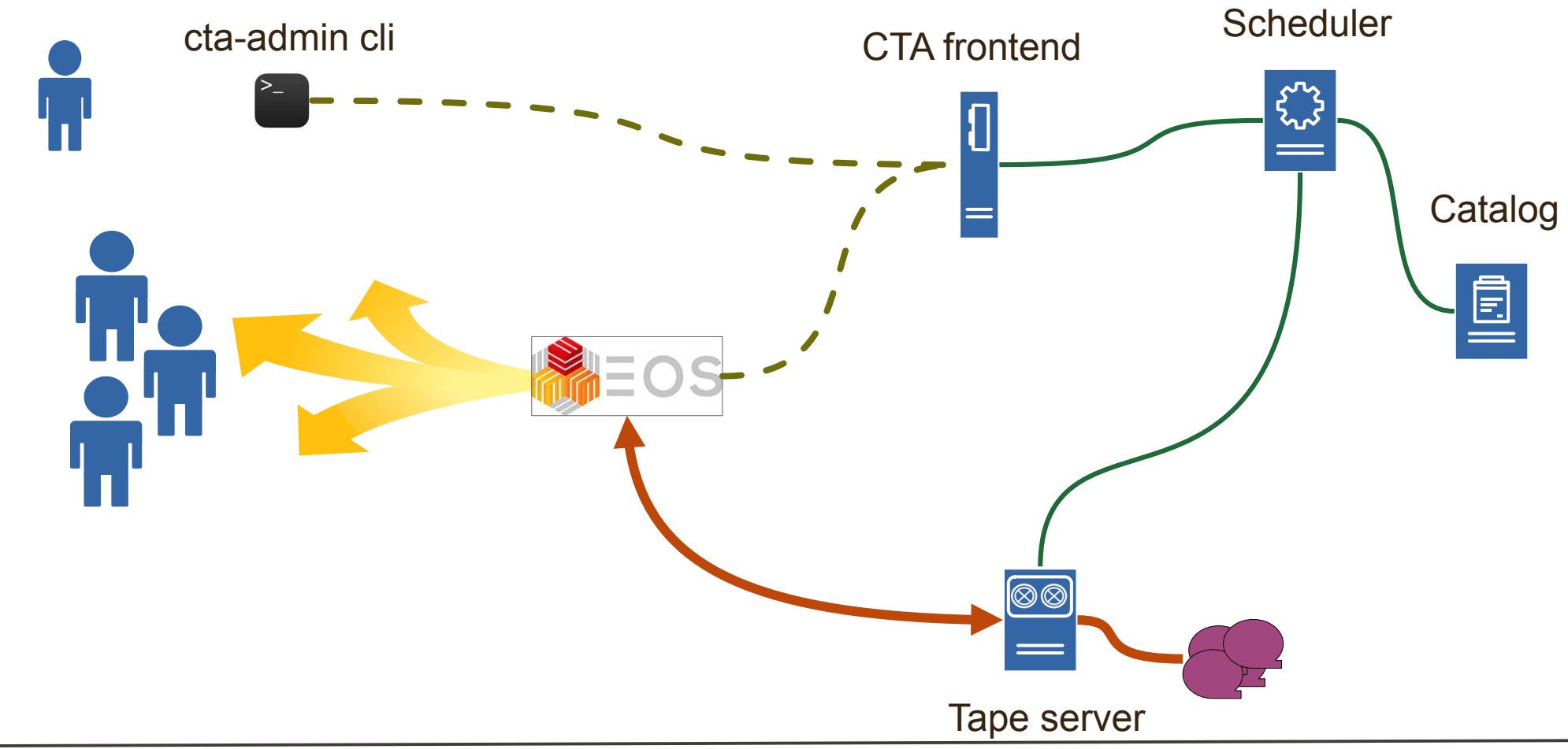
CERN Tape Archive

- Developed by CERN-IT for LHC experiments (successor of CASTOR)
  - Adopted by RAL, UK
  - Planed by Fermilab and other ENSTORE sites
- An open source alternative to commercial storage management systems.
  - Openness allows users to customize CTA and share ideas and solutions.
- Successful deployment of CTA by DESY can be seen as an example of productive collaboration between CERN-IT CTA and DESY dCache development teams.

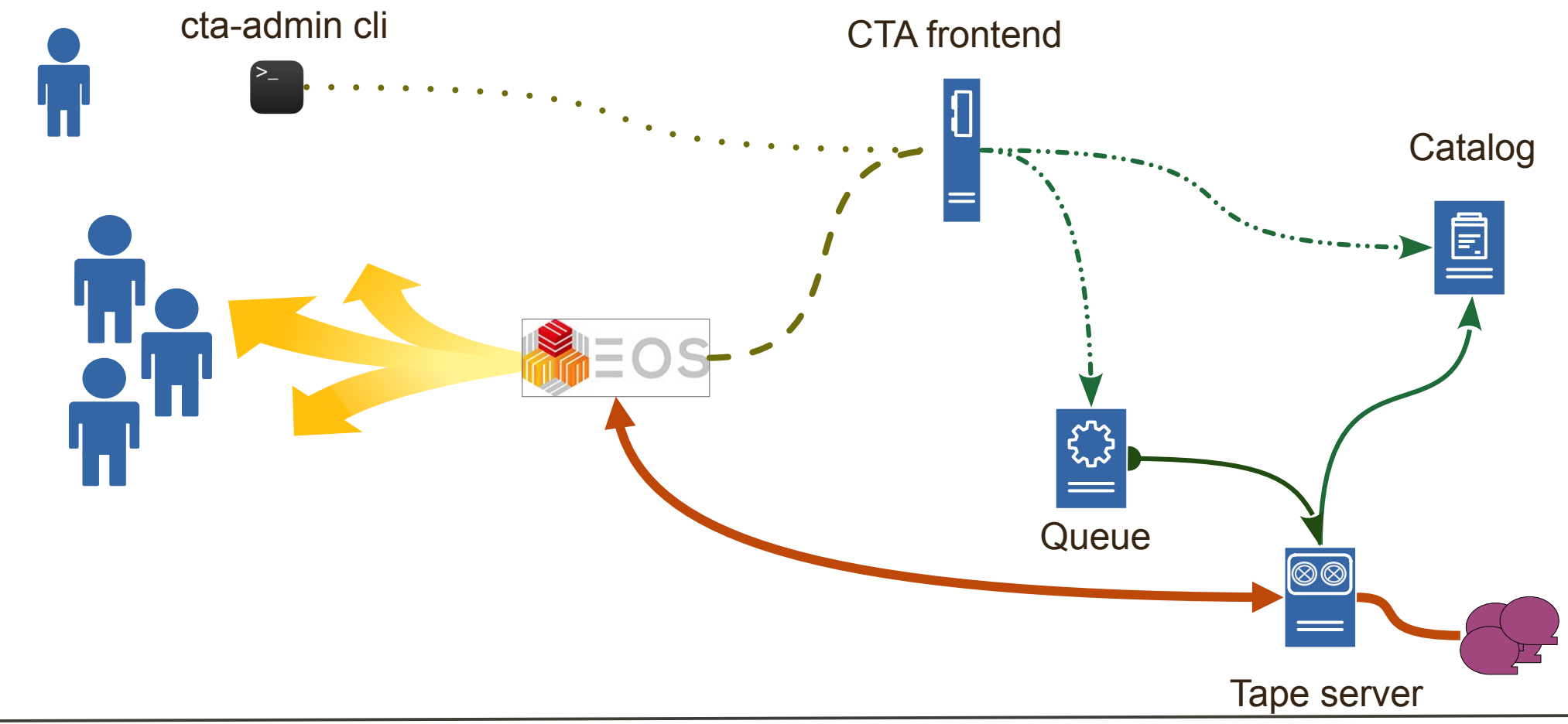
CTA Design, Basics

- cta-frontend
  - Accepts storage system requests: Archive, Retrieve, Cancel, Delete
    - Creates/Deletes entry in the Catalog
  - Creates request in the job queue (CEPH | file system | DB)
- cta-taped
  - One process per tape drive
  - Seeks for a jobs in the job queue
  - Moves data between disk ⟺ tape
  - Updates catalog
  - Notifies storage system about transfer success/failure

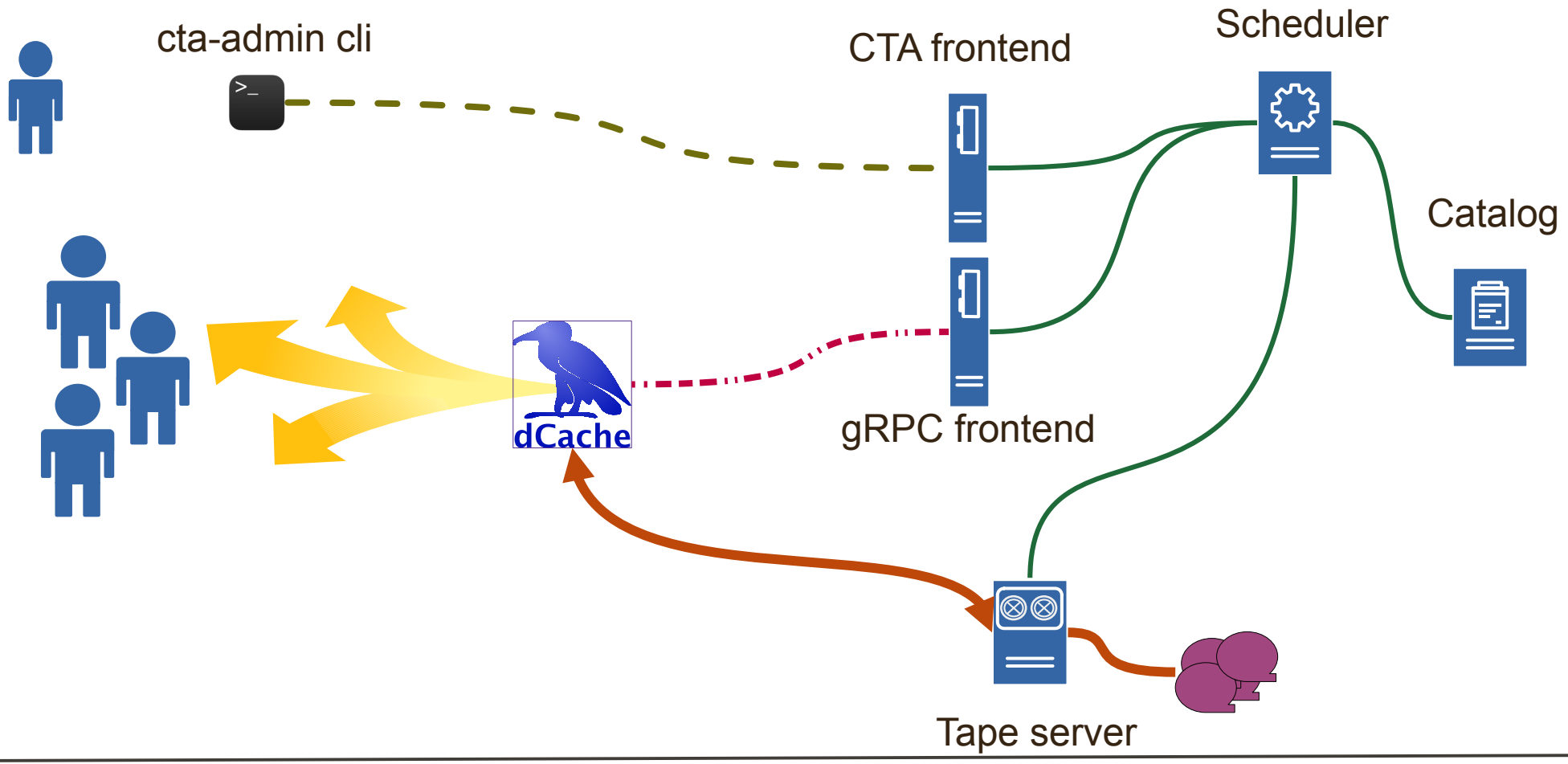
(Extremely) Simplified CTA design



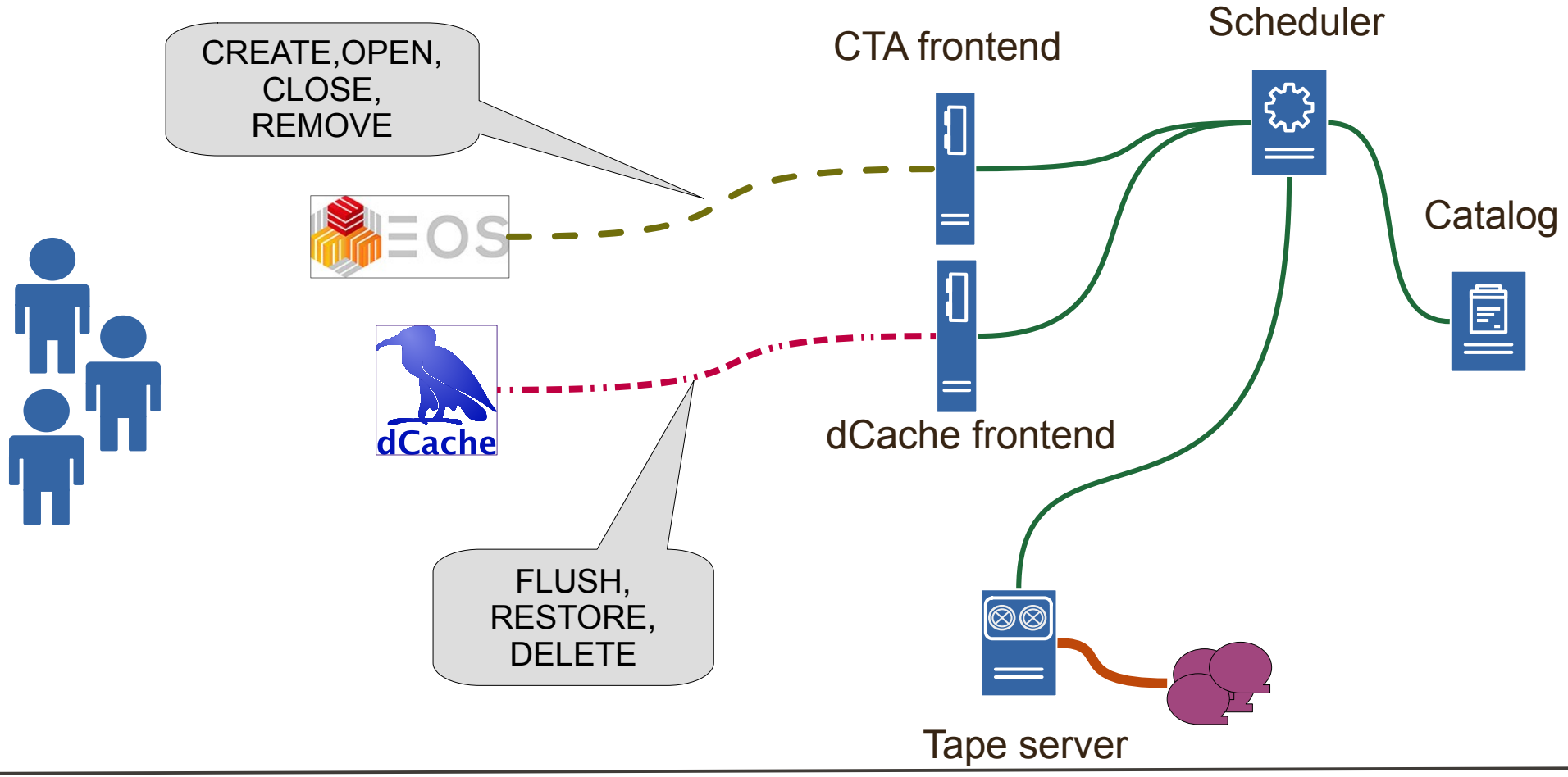
(Extremely) Simplified CTA design



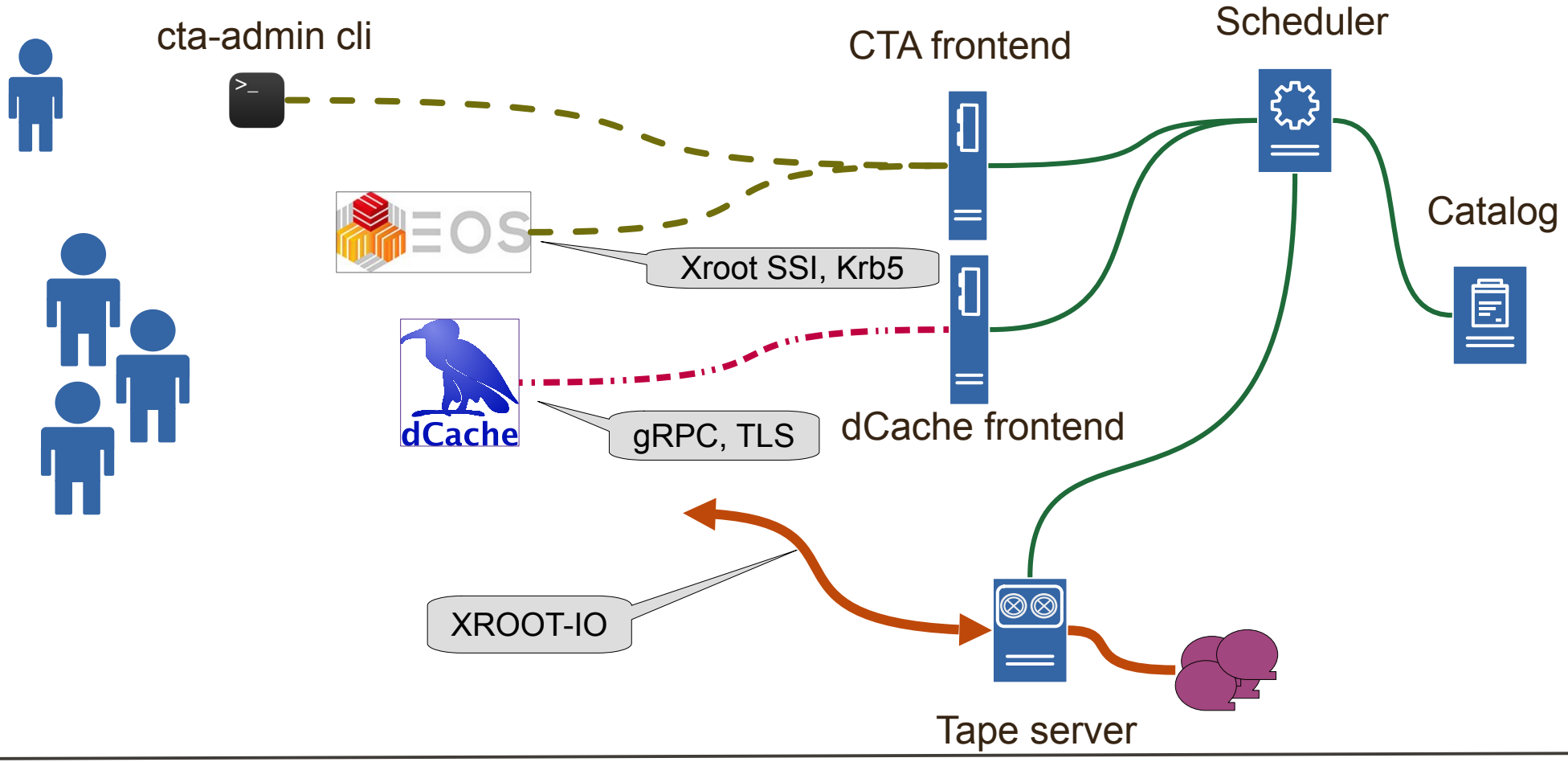
(Extremely) Simplified CTA design



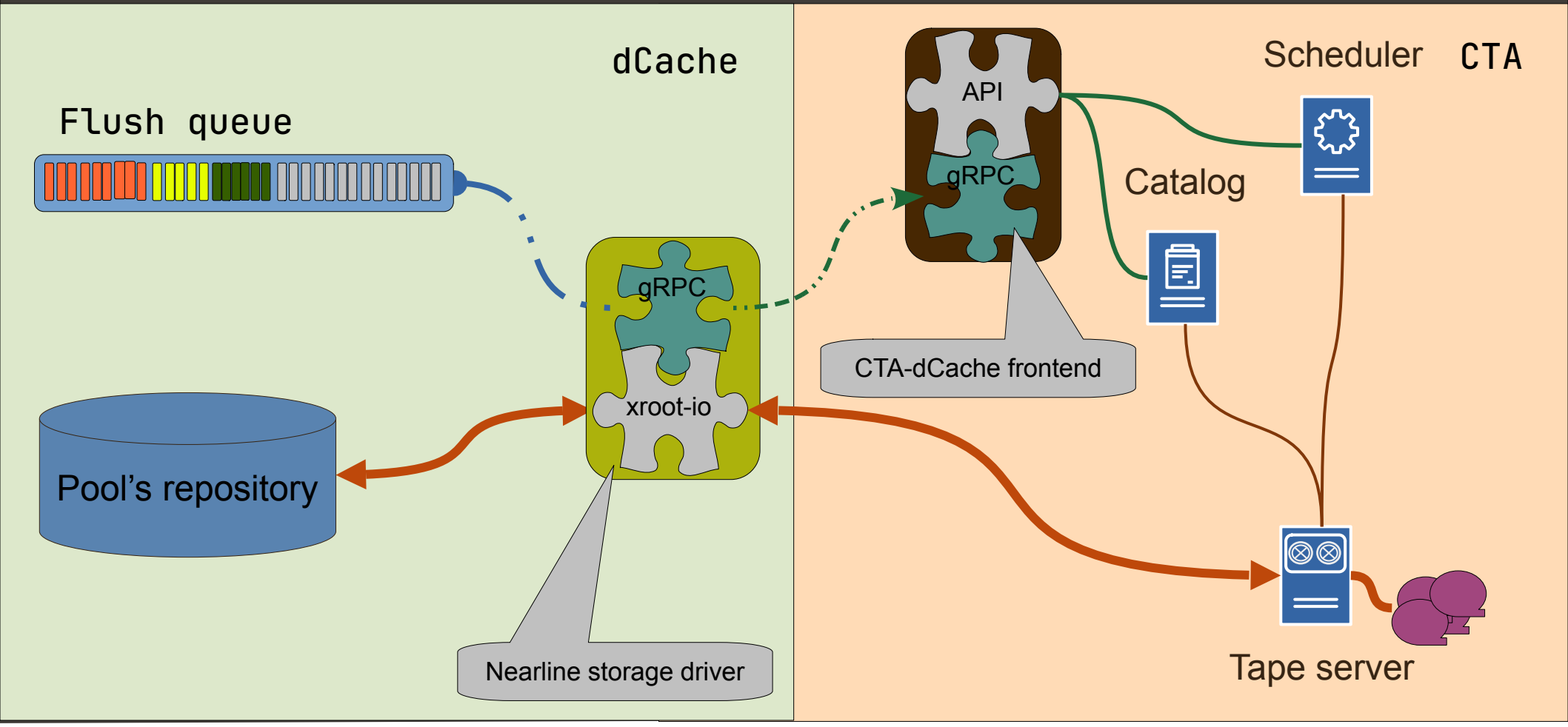
(Extremely) Simplified CTA design



(Extremely) Simplified CTA design



Nearline CTA Storage Driver



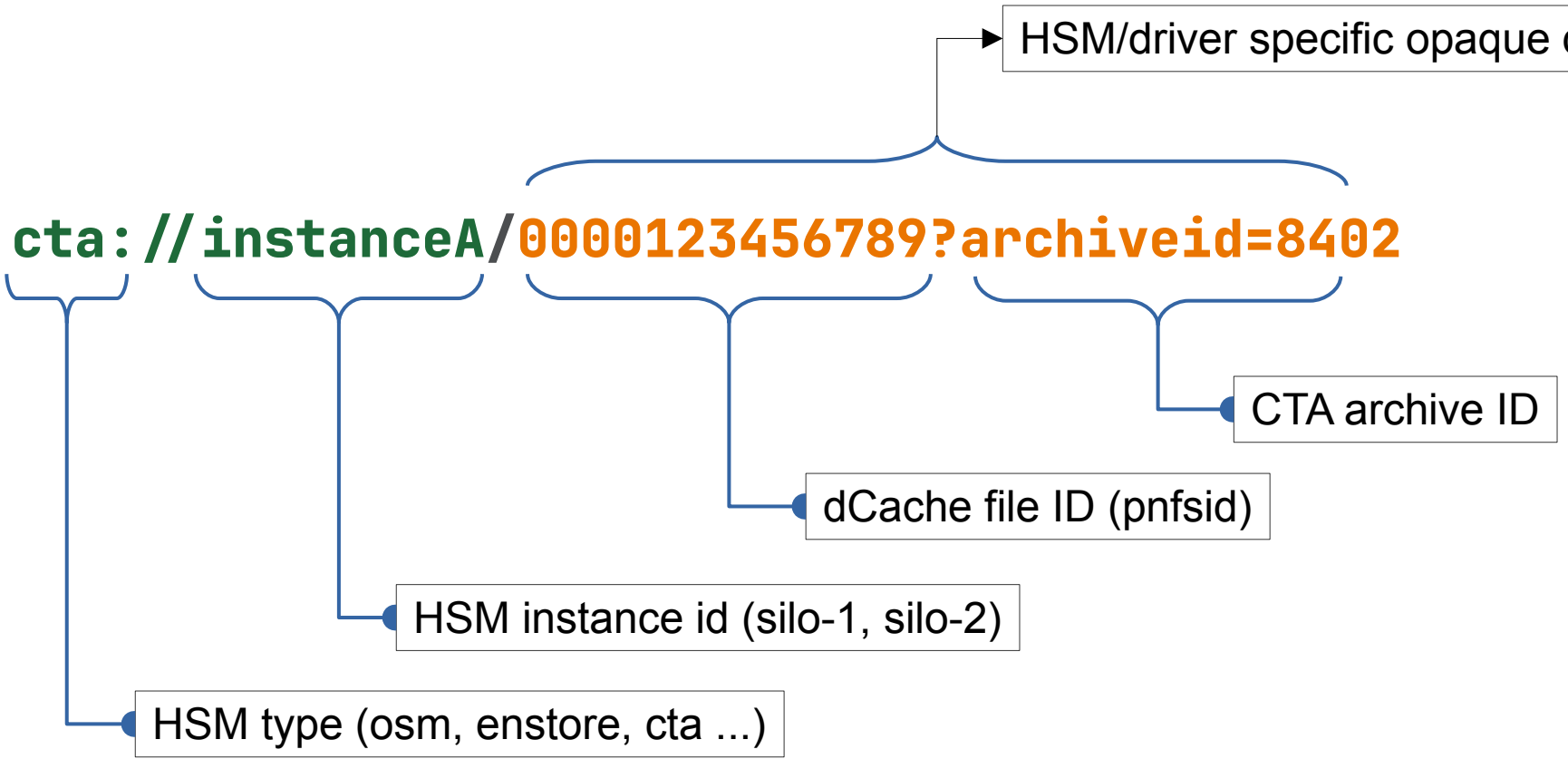
dCache HSM Interface

```
// dCache interface to tape system

public interface NearlineStorage {

    void flush(Collection<FlushRequest> requests);
    void stage(Collection<StageRequest> requests);
    void remove(Collction<RemoveRequest> requests);

    void cancel(UUID uuid);

    // driver initialization methods

    ...
}
```

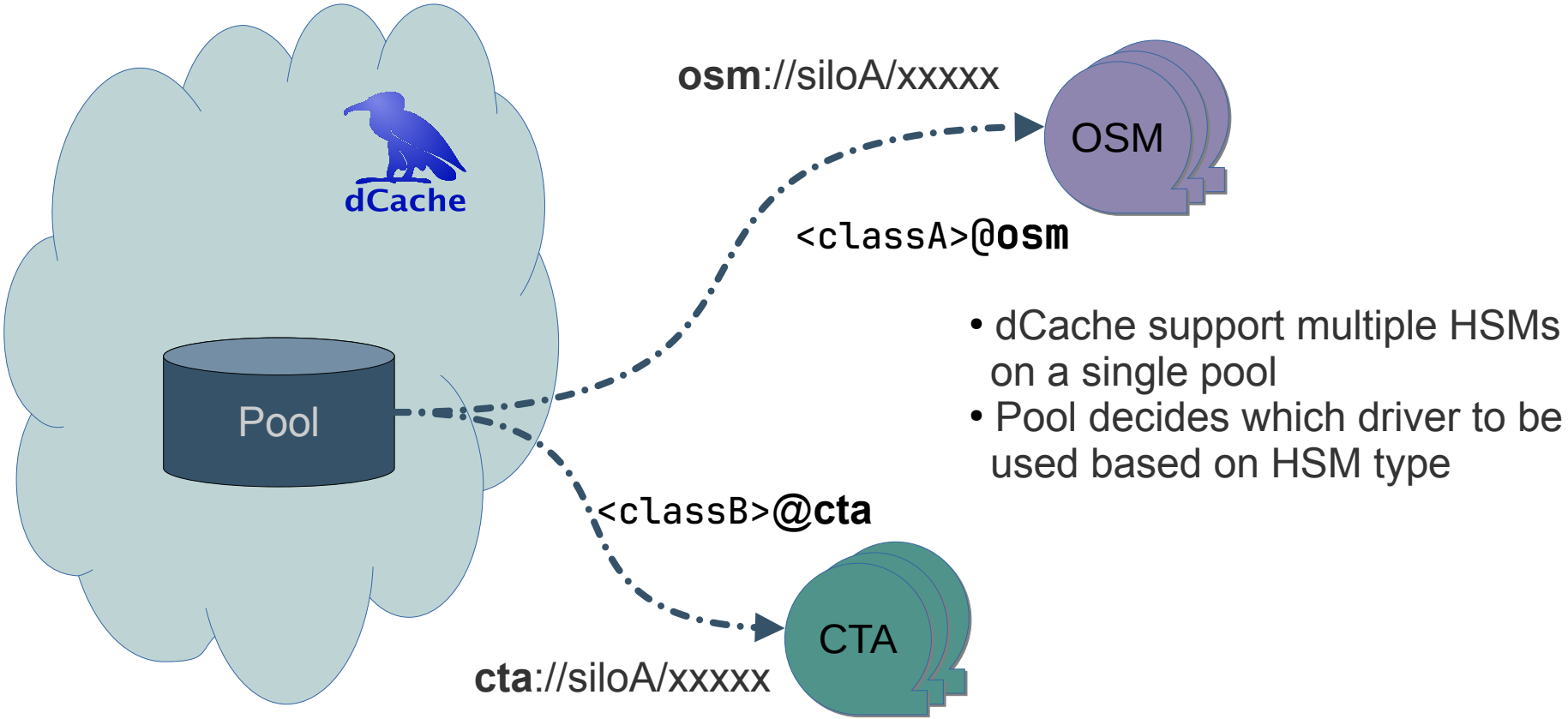
dCache CTA gRPC

```
// gRPC definition of dcache-cta interface

service CtaRpc {
  rpc Version (google.protobuf.Empty) returns (cta.admin.Version) {}


  rpc Create (CreateRequest) returns (CreateResponse) {}
  rpc Archive (ArchiveRequest) returns (ArchiveResponse) {}
  rpc Retrieve (RetrieveRequest) returns (RetrieveResponse) {}
  rpc Delete (DeleteRequest) returns (google.protobuf.Empty) {}
  rpc CancelRetrieve (CancelRequest) returns (google.protobuf.Empty) {}

}
```

dCache HSM ⟺ Link



Multi HSM Deployment

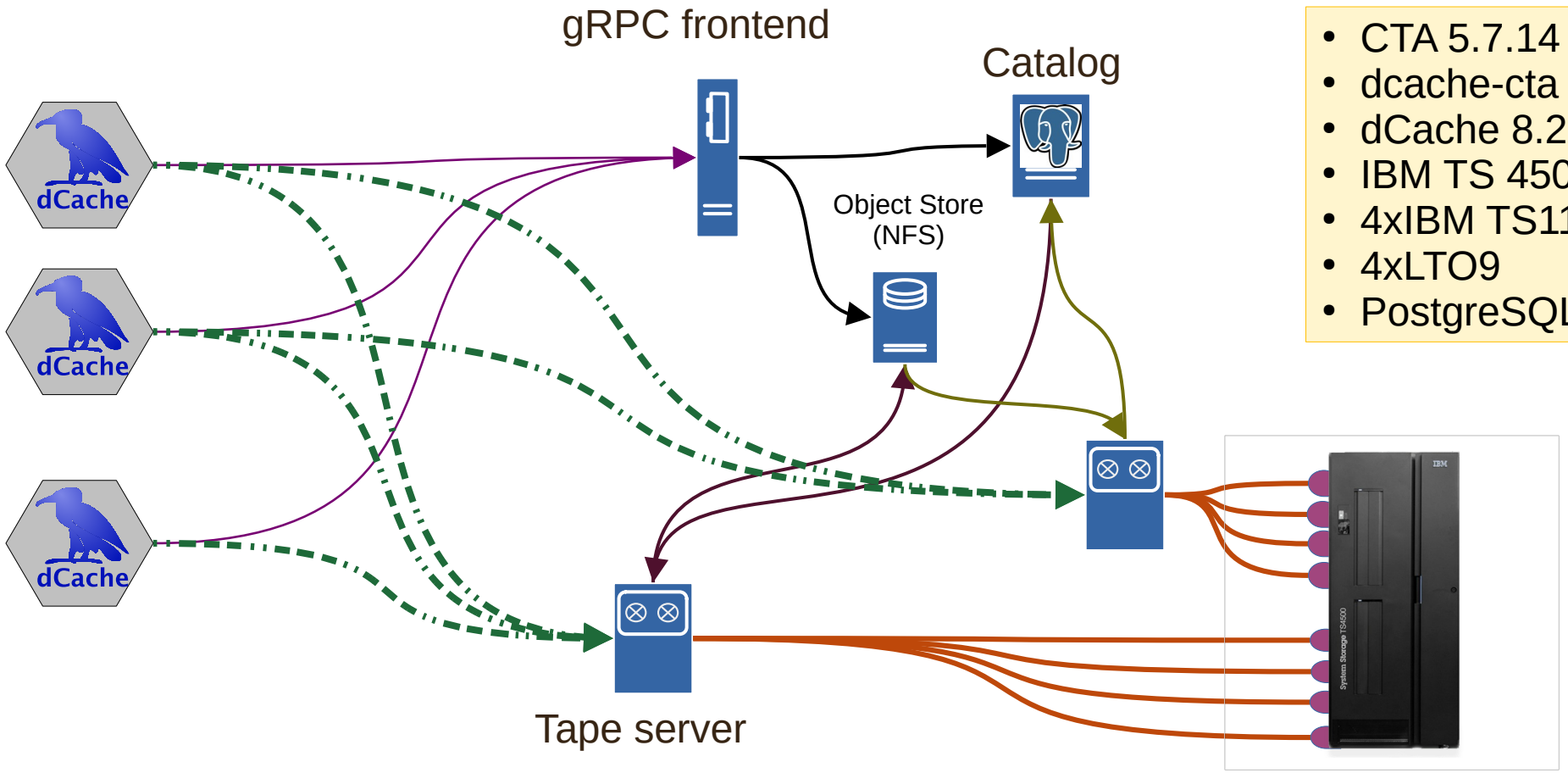


- dCache support multiple HSMs on a single pool
- Pool decides which driver to be used based on HSM type

dCache Config

```
[dcache-head] (dcache-xfel399-09)> hsm ls
cta(cta):dcache-cta     // <class>@cta
    cta-instance-name   production
    cta-frontend-addr   tpm103.desy.de:17017
    ...
osm(osm):script         // <class>@osm
    command   /usr/share/dcache/lib/hsmcp.py
    ...
```

Deployment at DESY



- CTA 5.7.14
- dcache-cta 0.8.0
- dCache 8.2.7
- IBM TS 4500
- 4xIBM TS1160
- 4xLTO9
- PostgreSQL 14.1

dCache ⟺ CTA Integration

dCache (⩾ 7.2)

- Nearline driver to add
- Can run in parallel with other HSMs
- Pre-scheduling on pools should be disabled/reduced
- File path, uid, gid not preserved

CTA (⩾ {5,4}.7.12)

- Additional *cta-frontend-grpc* service (packaged as own rpm)
- Limited to dCache required minimal functionality
  - *Not dCache specific*
  - *cta-frontend still needed for admin commands*

dCache+CTA Status

- Seamless integration with dCache is merged into upstream CTA code at CERN
  - The latest official CERN release 5.7.14 is deployed at DESY.
  - The proposed dCache interface is under adoption by EOS.
- The existing OSM tape format is supported for READ
  - The code changes are adopted by Fermilab data management team for ENSTORE tape format.
  - The OSM tape catalog conversion procedure is ready and exercised multiple times. Final migration expected by Q1 2023 (a.k.a now).

- Our deployment replicate to by other HEP sites
  - PIC Barcelona have successfully replicated our setup (currently dCache + ENSTORE).
  - Fermilab is planning in Q2 2023 (currently dCache + ENSTORE).
  - RAL in UK plans to migrate to PostgreSQL from ORACLE based on our experience

Next Steps

Work in progress

- Handling re-submits
  - dCache restarts and retries re-submits requests to CTA
- New scheduler
  - Order requests by creation time
- gRPC front-end for admin commands
  - Replace legacy xrootd based communication with widely adopted standard
- CI tests covering CTA integration with dCache
  - Tests done at CERN k8s cluster. dCache tests are missing.
- CTA in container
  - Easy deploy deployment for testing

Questions?