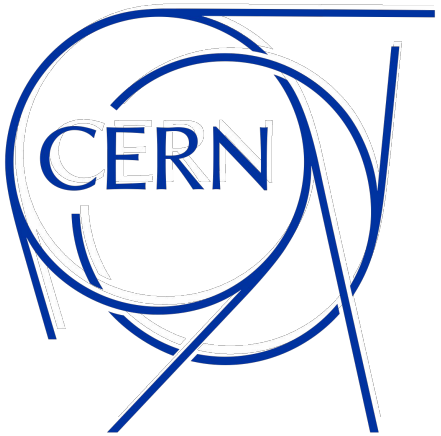


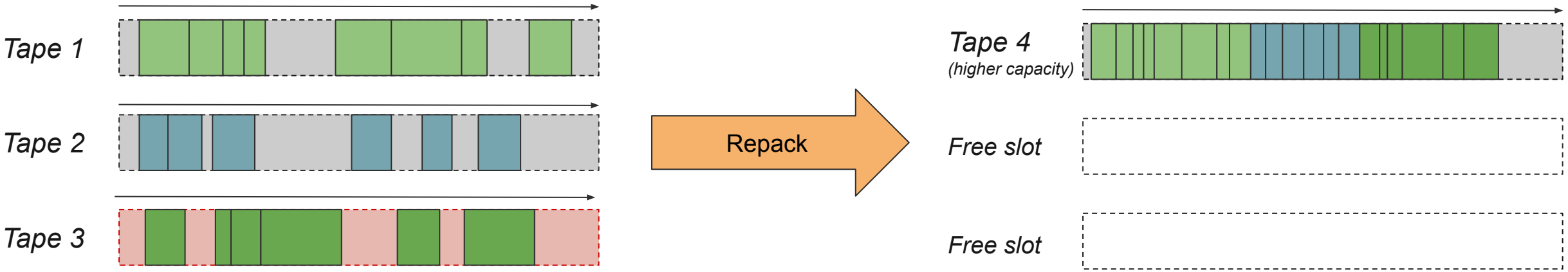
Challenges of repack in the era of the high-capacity tape cartridge

João Afonso
joao.afonso@cern.ch

22 October 2024

Use-cases for tape repack

- Moving data out of problematic tapes
- Filling gaps from deleted files
- Moving data to newer and higher density tapes
  - *Latest drivers are no longer backward compatible with previous generations of tape cartridges*
- Free up tape cartridge slots for future capacity expansion

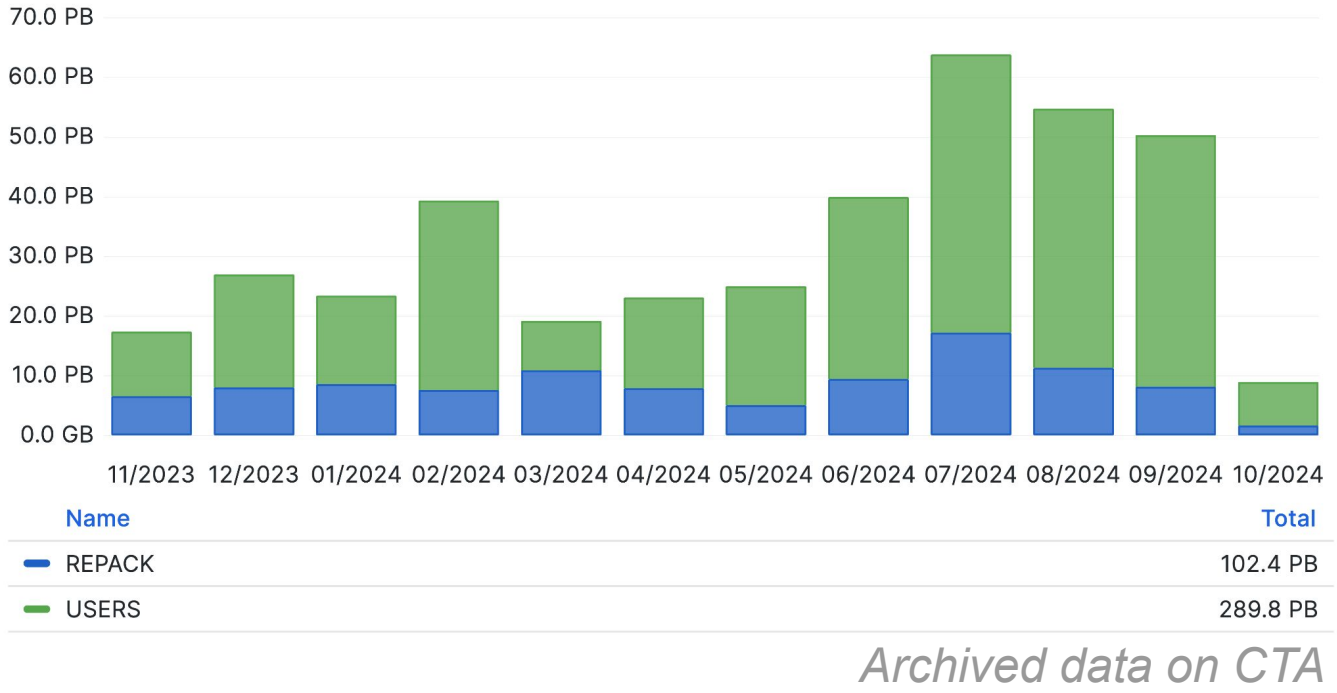


Repack is a very important and heavy procedure

- ~25% of all data written during the last year was due to repack (~100PB)
- Amount comparable to that of a large experiment

Note:

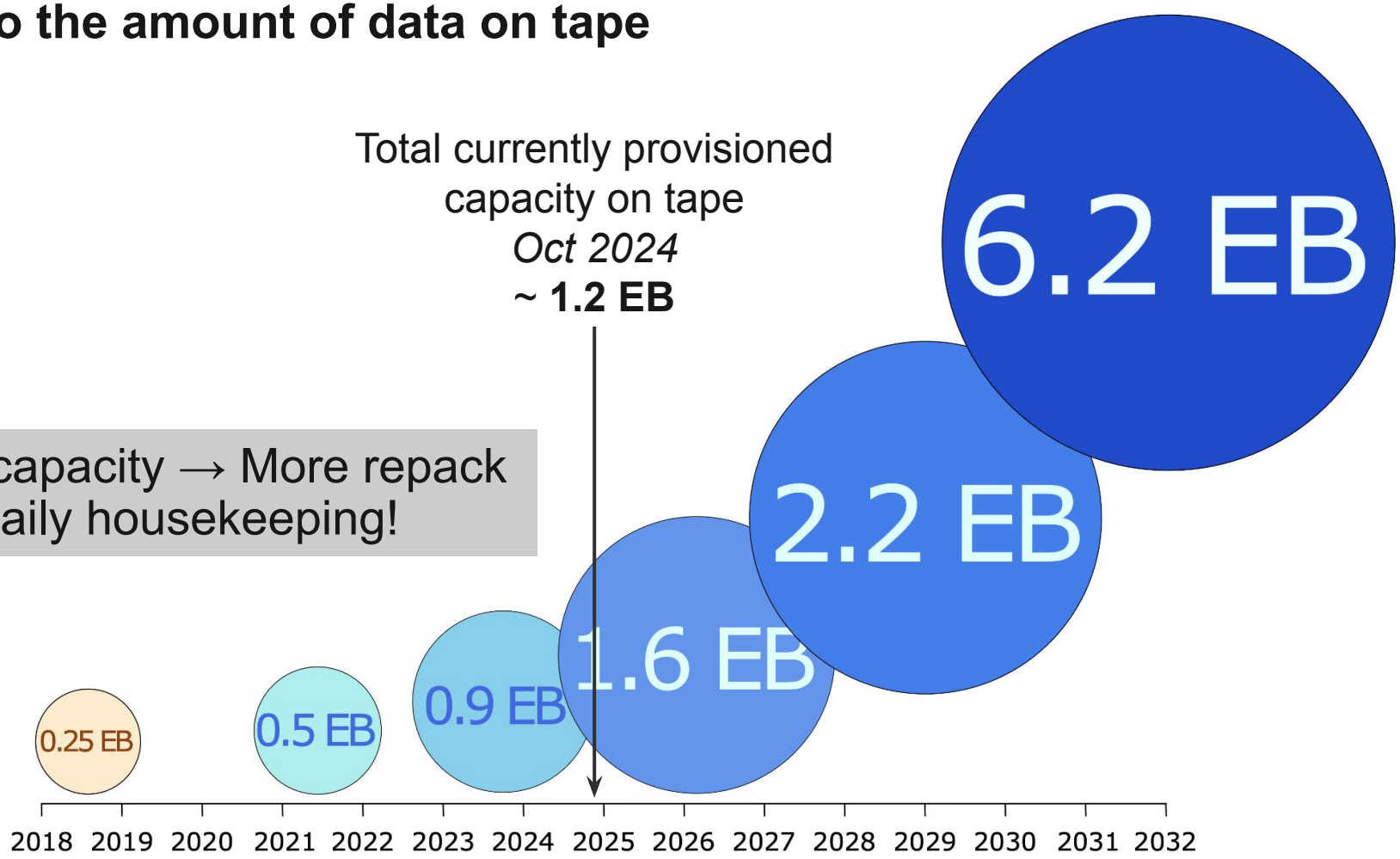
- *Repack activity is opportunistic:*
  - *It runs at lower priority, only when there are free resources!*



Archived data on CTA

And is expected to increase exponentially

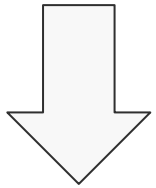
Proportionally to the amount of data on tape



Higher total tape capacity → More repack needed for daily housekeeping!

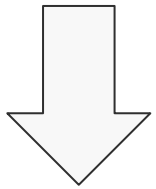
Consequences for operations

- **Tapes must be repacked much more aggressively *and smarter* than in the past.**



Main question:

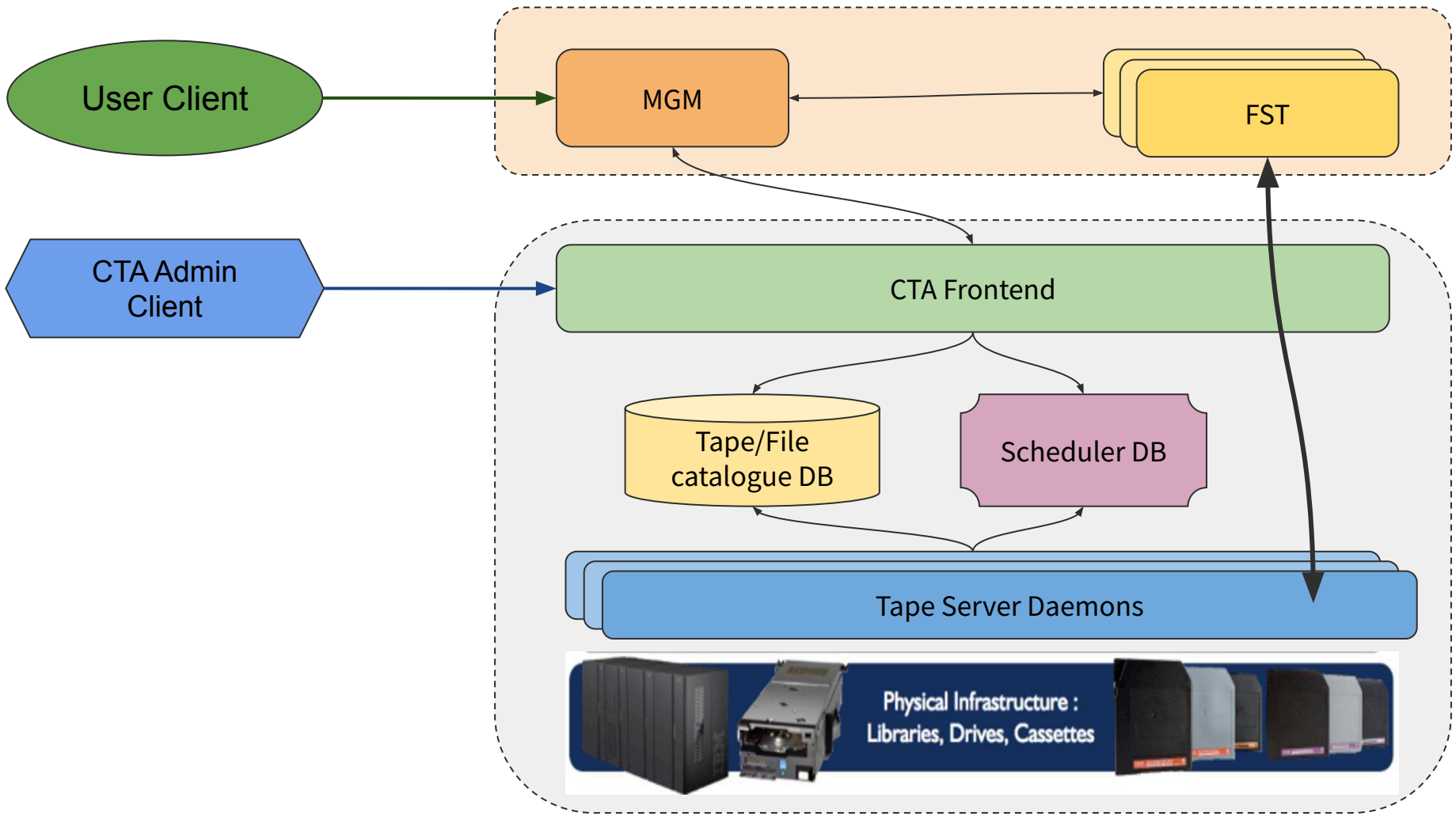
- **How to perform *repack* without affecting the *archival/retrieval* of *user/experiment* data.**



Strategy followed:

- **Decouple *repack* from the archival of *user/experiment* data.**

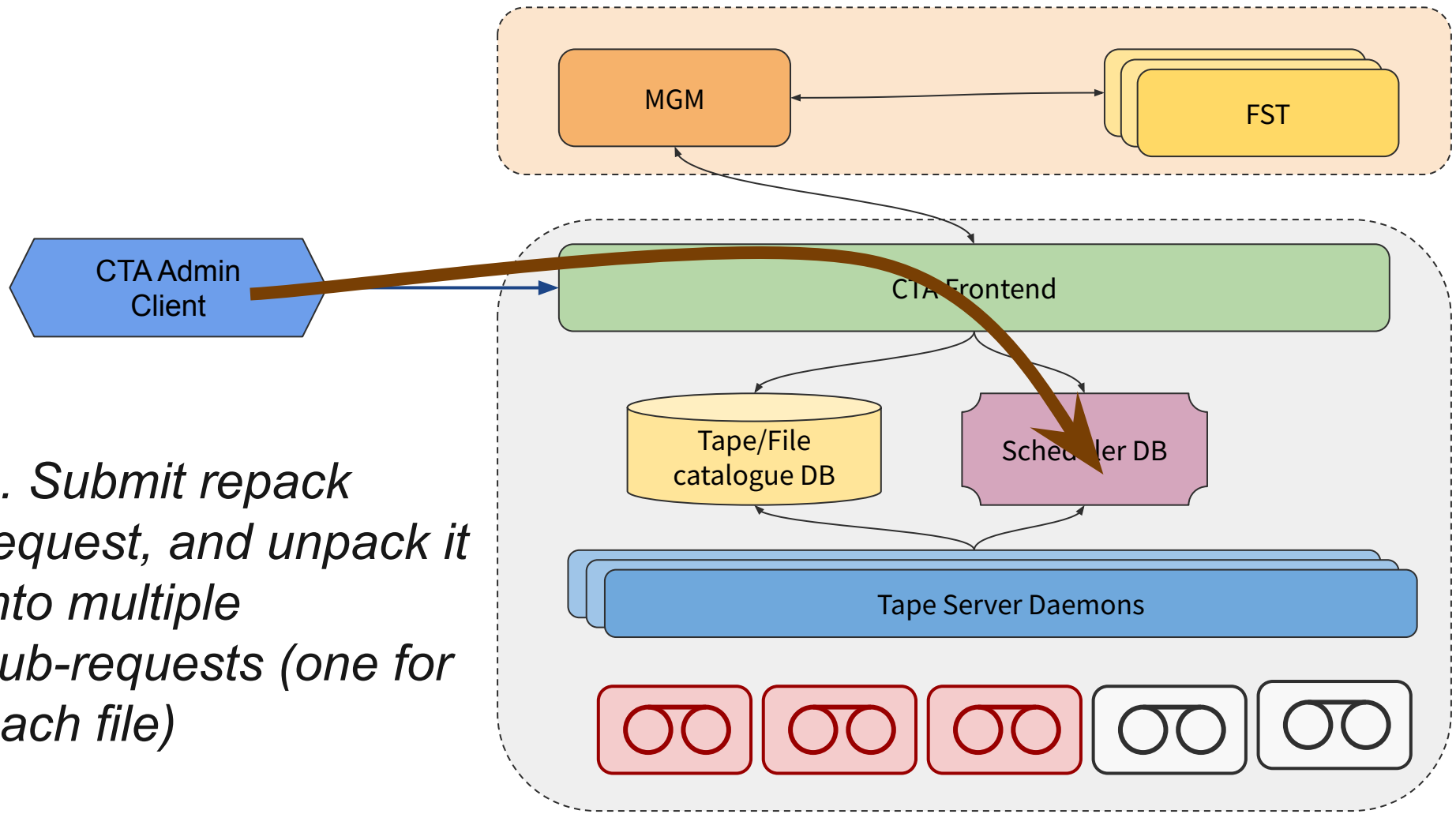
How repack works: EOS + CTA Architecture



EOS Disk Buffer

CTA Tape Backend

How repack works:

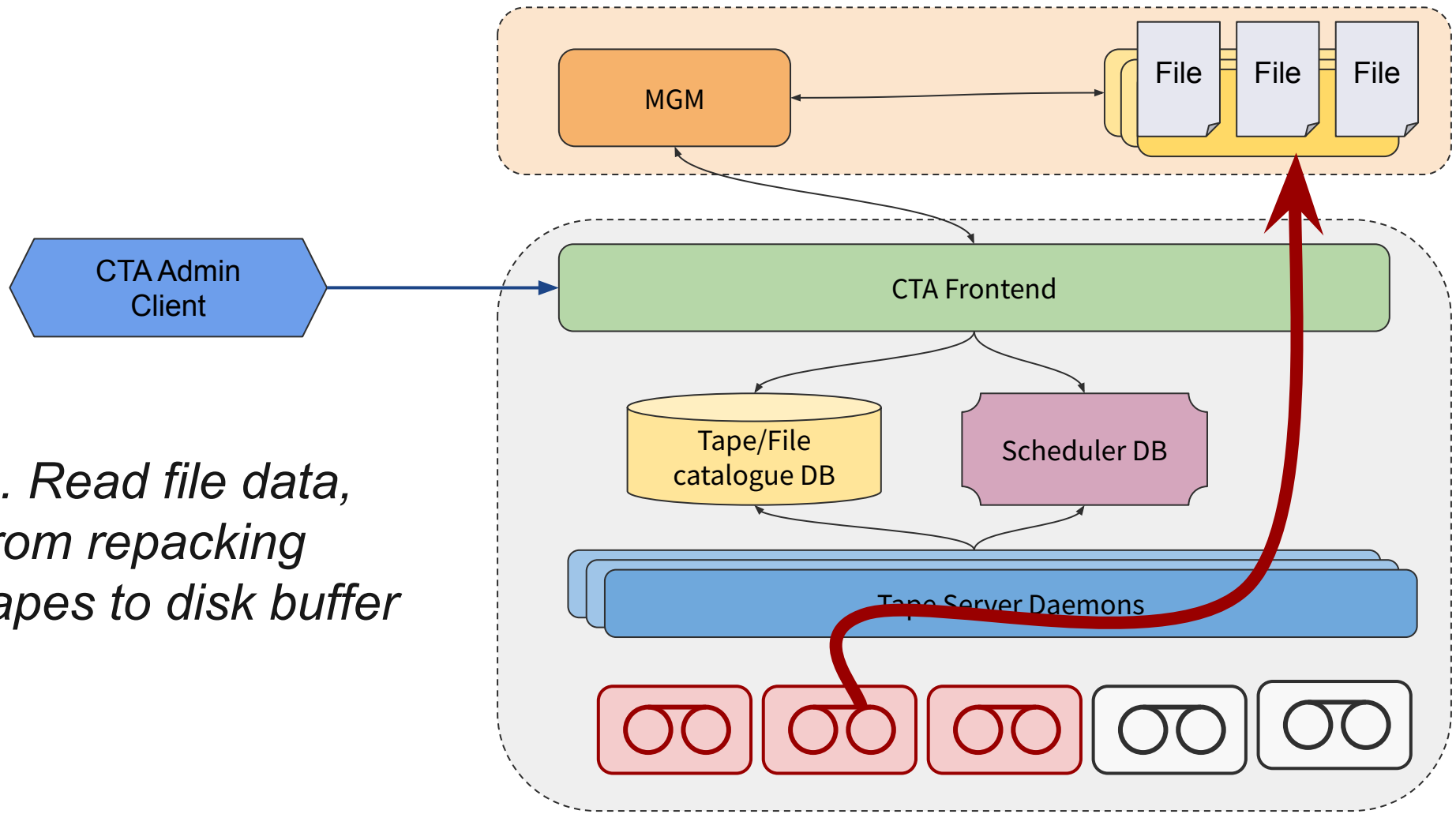


EOS Disk Buffer

CTA Tape Backend

1. Submit repack request, and unpack it into multiple sub-requests (one for each file)

How repack works:

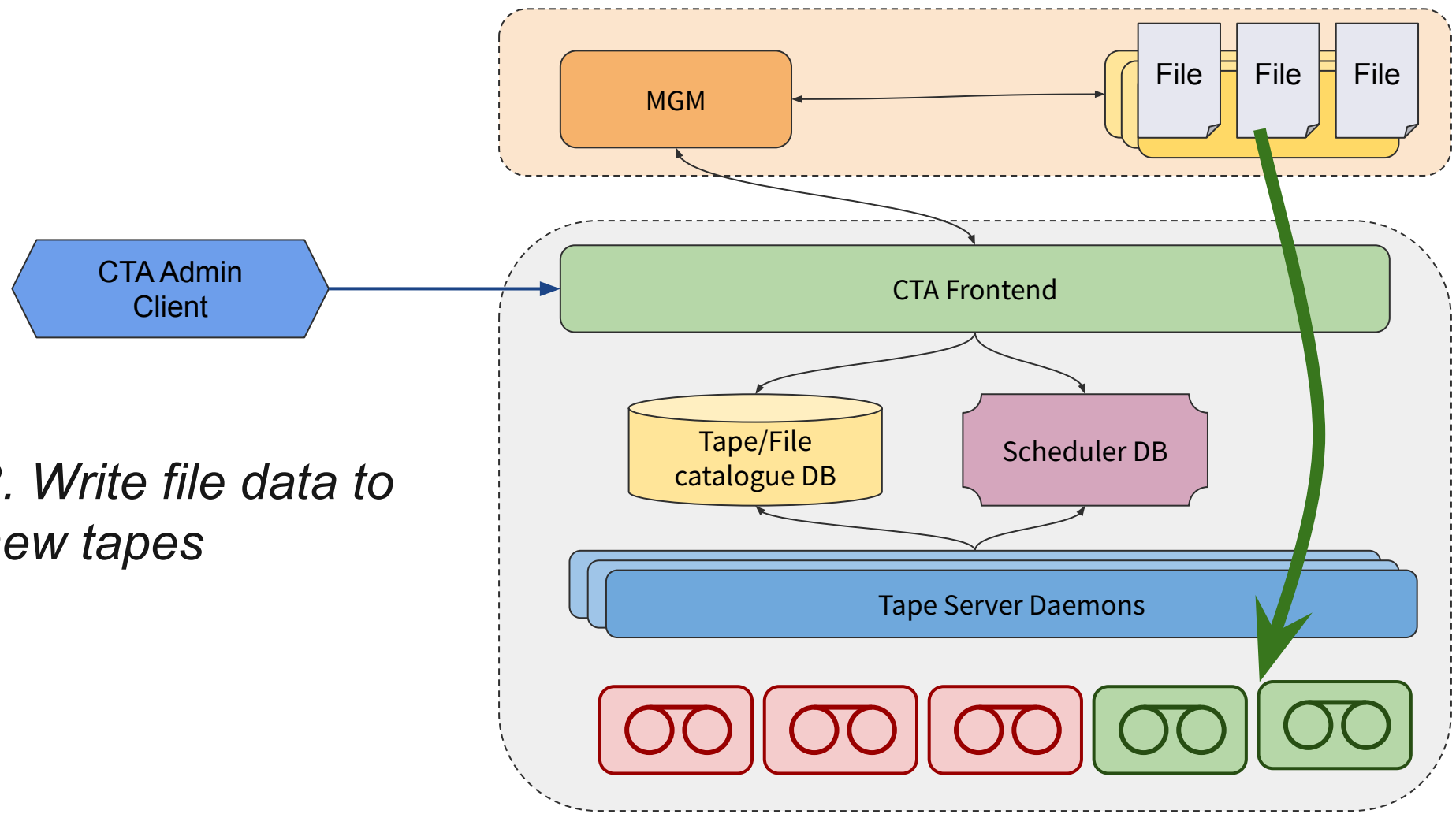


EOS Disk Buffer

CTA Tape Backend

2. Read file data, from repacking tapes to disk buffer

How repack works:

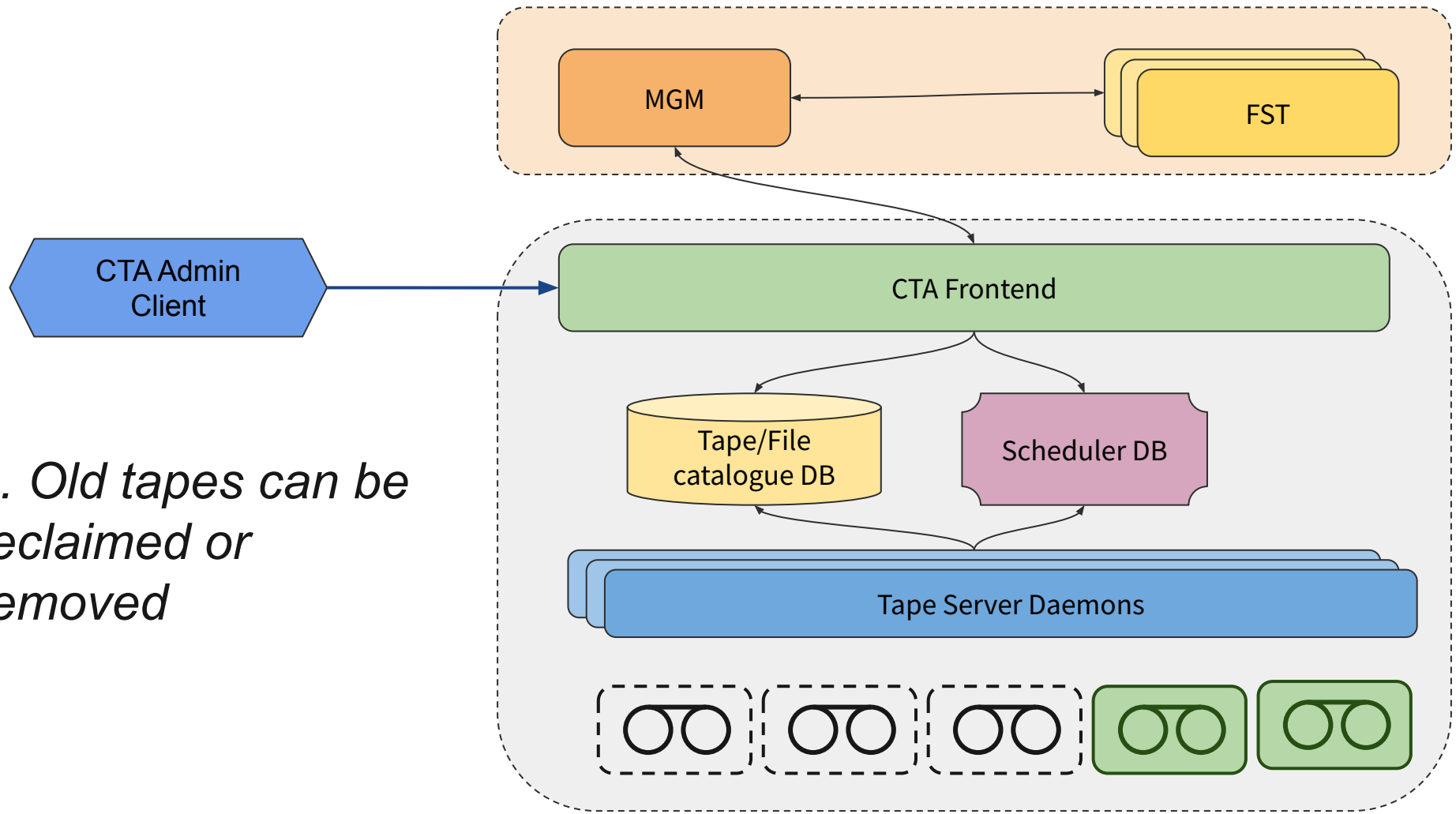


3. Write file data to new tapes

EOS Disk Buffer

CTA Tape Backend

How repack works:



EOS Disk Buffer

CTA Tape Backend

4. Old tapes can be reclaimed or removed

New features implemented for repack

1. New state for repacking tapes

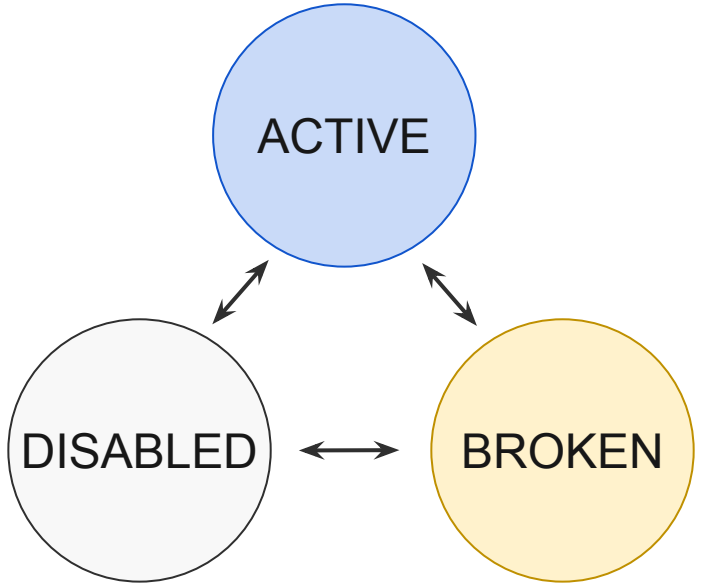
Before:

Tapes did not have a clear state for *repack*.

As a result, *user* requests could end up queued on *repacking* tapes and get mixed with *repack* sub-requests.

This resulted in unclear information passed to user:

- No notification of delays caused by *repack*
- If tape was problematic, *user* requests could stay queued indefinitely



1. New state for repacking tapes

Now:

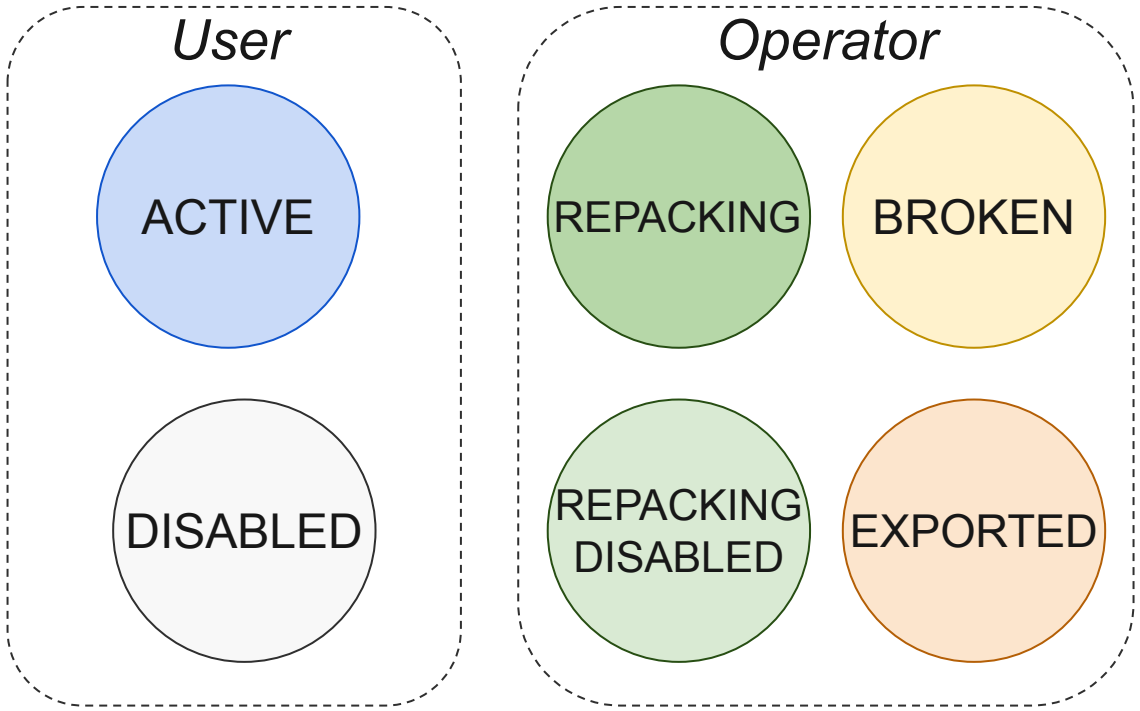
New states **specifically** for *repacking* tapes.

Clear separation between *user* and *operator* domains:

- *User* request only queued on:
  - ACTIVE, DISABLED
- *Repack* requests only queued on:
  - REPACKING, REPACKING_DISABLED

User retrieve requests are removed automatically and reported back, when moving tapes from *user* to *operator* domain:

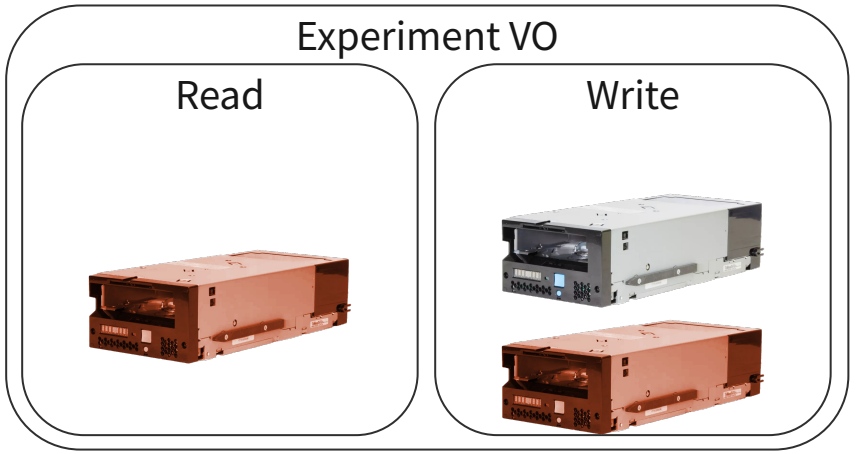
- Data will be available once *repack* is done
- Clear message passed to the *user*



2. Dedicated Virtual Organization (VO) for Repack

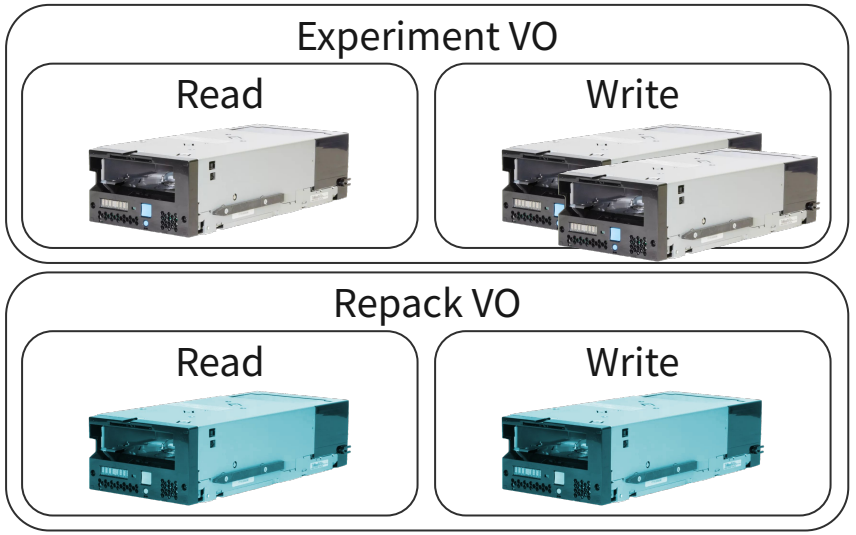
Before:

- *Repacking* tapes would consume their respective *VO* drive quota.
- This could keep most (or all) the *VO* drives occupied.
- Read/write operations from *users/experiments* risked getting starved.



Now:

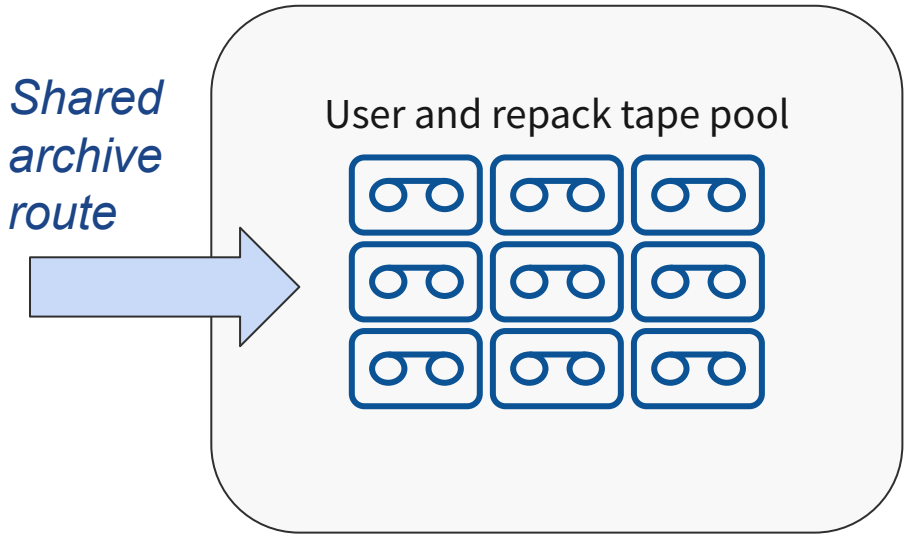
- Operators can define a *default VO* for *repack*.
- *User* driver quota will not be affected.
- Can control the number of drives used for reading/writing *repacking* data.



3. New Archive Routes for Repack

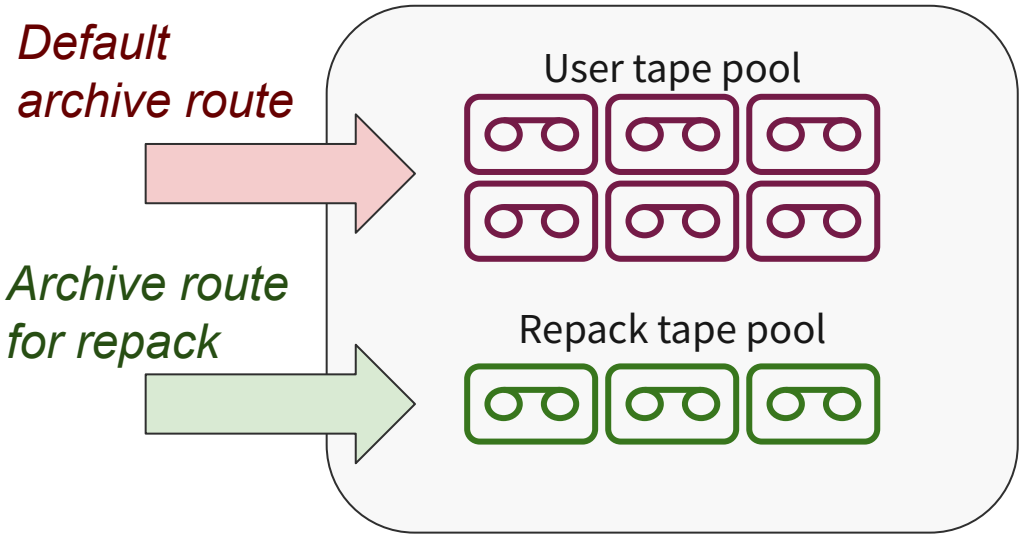
Before:

- Shared *archive routes* between *user* and *repack* archive jobs (per *storage class*):
  - **Tapes selected from same tape pool.**
- Old and new data could get mixed…



Now *(next CTA release)*:

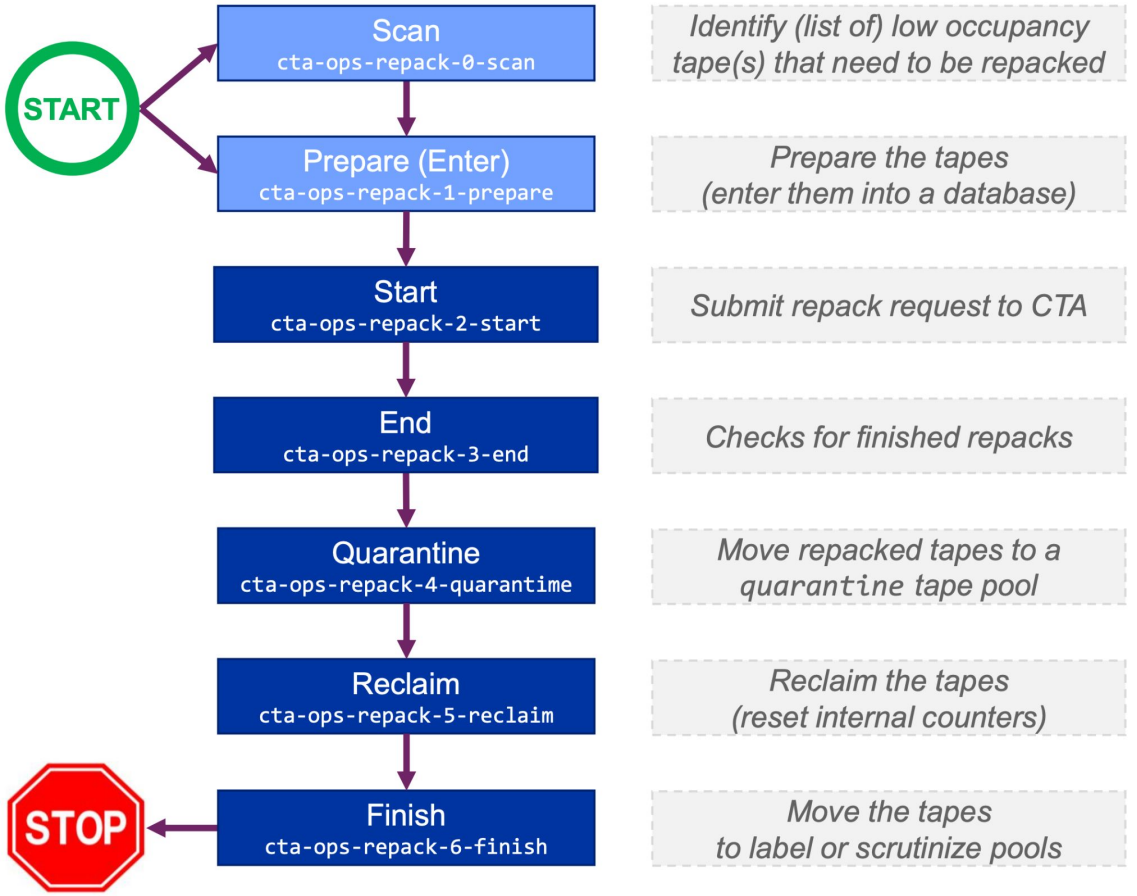
- Operators can define *repack archive routes*:
  - **Different tape pools selected.**
- Old and new data do not get mixed:
  - **Data colocation is preserved**



ATRESYS

Automated Tape REpacking SYStem

- Tool to automate orchestration of tape repacks.
- Takes advantage of new features.
- Used by CTA operations.

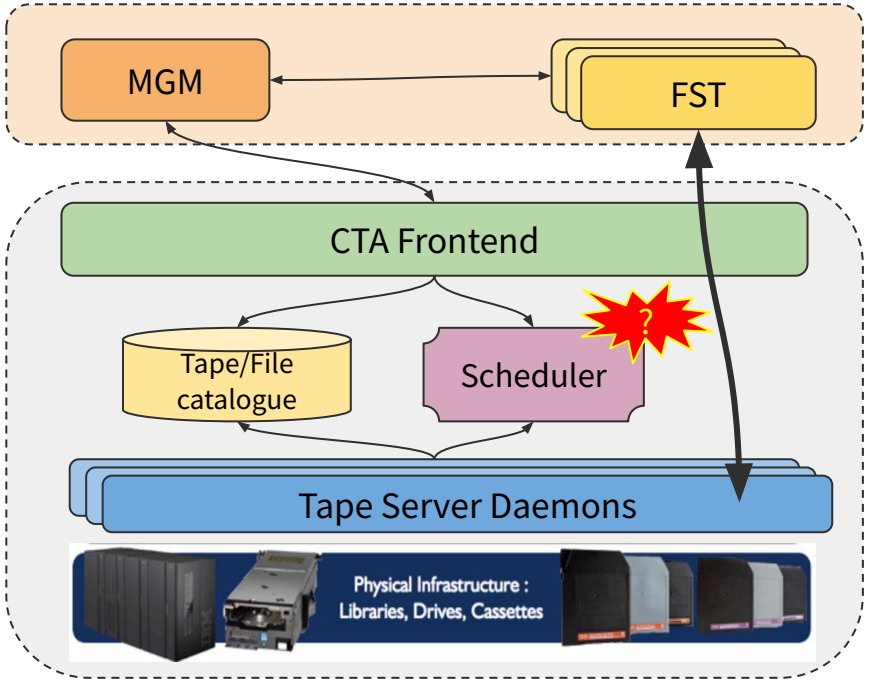


Performance challenges

We started *repacking*.

Unfortunately, when performing *repack* on large tapes we saw a significant degradation on the scheduler backend performance, which caused CTA to become unresponsive.

This was affecting not only *repack*, but also *user* requests.



Performance challenges

Main reason:

- Modern tapes can contain ***millions of files***. *Repacking* them results in queueing a very large number of *retrieve/archive sub-requests*.
- These *sub-requests* need to be tracked collectively by a *repack* request, which can grow to a very large object size in the current objectstore scheduler backend (Ceph/RADOS).
- This can result in very slow read/write/update operations.

Examples of large repack objects:

Tape	Nr files	*Repack* object size
L76199	2725278	~272 MB
I00146	2605639	~260 MB
I00837	2571847	~257 MB
I75773	2286214	~228 MB

Solution found

Two mitigation strategies were followed:

1. **Allow operators to limit the number of files to repack per tape:**

   - Keeps the pressure on the scheduler backend under control (ex: repacking 200K files only requires a 20MB object).
   - As a consequence, repacking a tape may require several iterations until complete.

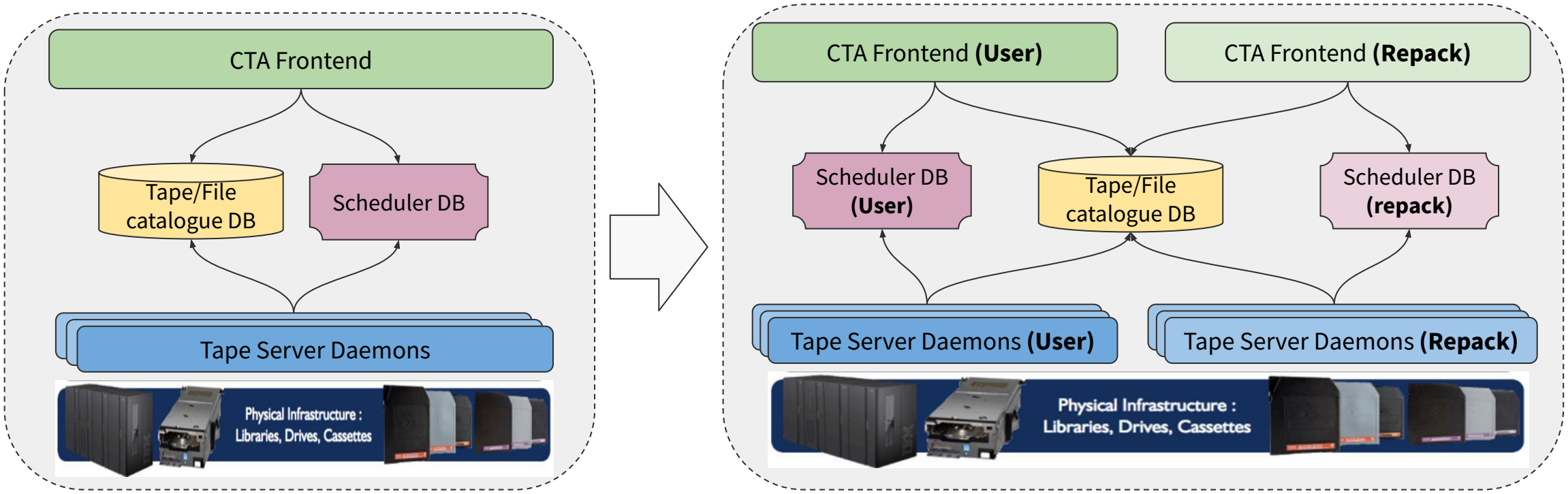
2. **Split user and repack scheduler backends *(soon in production at CERN)*:**

   - *Optional configuration.*
   - *User archive/retrieve* requests can no longer be impacted by heavy *repack* operations.

A long-term solution (partitioning the *repack* request data) will be provided by the future *PostgreSQL Scheduler DB*:

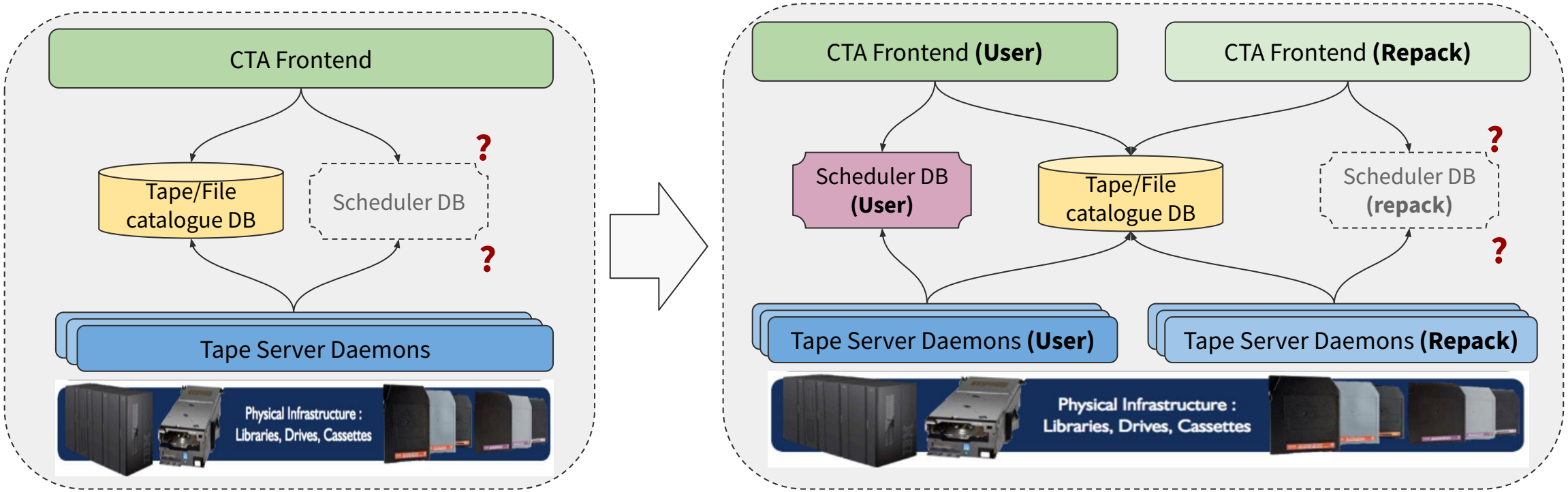
- Check *“Evolution of the CERN Tape Archive scheduling system”*, by Jaroslav Guenther (CHEP 2024).

Split user and repack scheduler backends



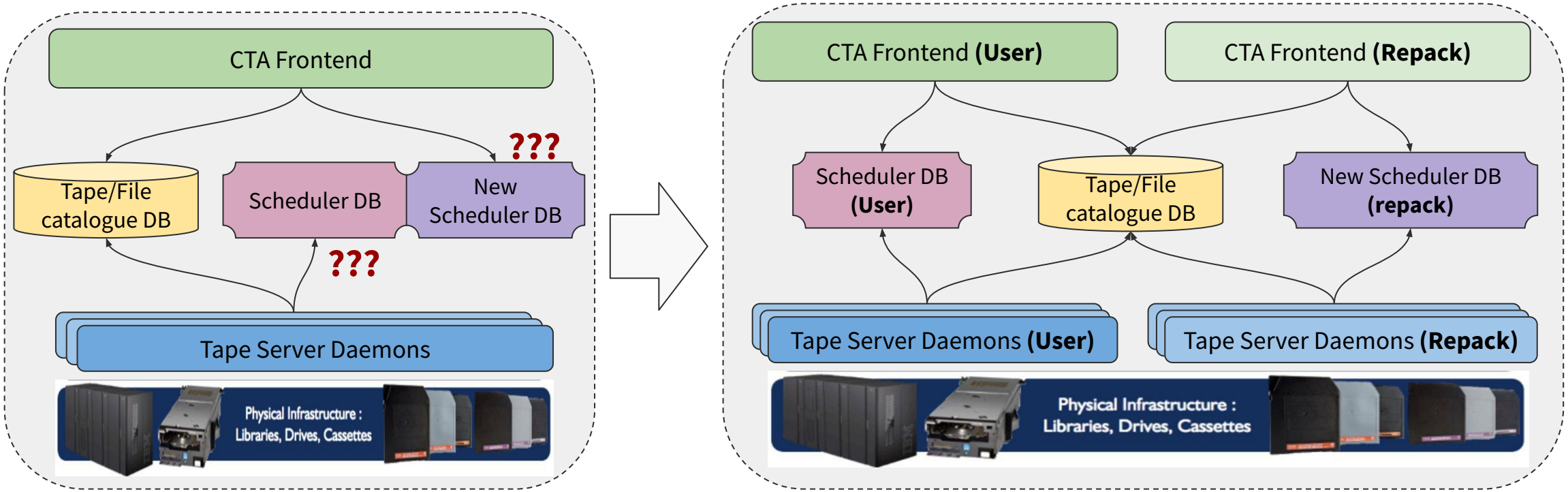
Split user and repack scheduler backends

User path will still work if the repack backend fails



Split user and repack scheduler backends

We can test the new PostgreSQL Scheduler DB with *repack*, without affecting *user* request scheduling



Conclusion

- At CERN, *repacking* has become more and more critical for tape operations.
- Therefore, to keep the system healthy, it became important to decouple *repack* from the *archival/retrieval* of *user* data.
- To improve *repack* operations, new features were added to CTA:
  - New tape state for *repack*.
  - New *virtual organization (VO)* for *repack.*
  - New *archive routes* for repack.
- Finally, to shield the *archival/retrieval user* requests from *repack* operations, we allow the scheduler backend to be split in two.

home.cern