Funded by NSF IBM

Rethinking Tiered Storage: Talk to File Systems, Not Device Drivers

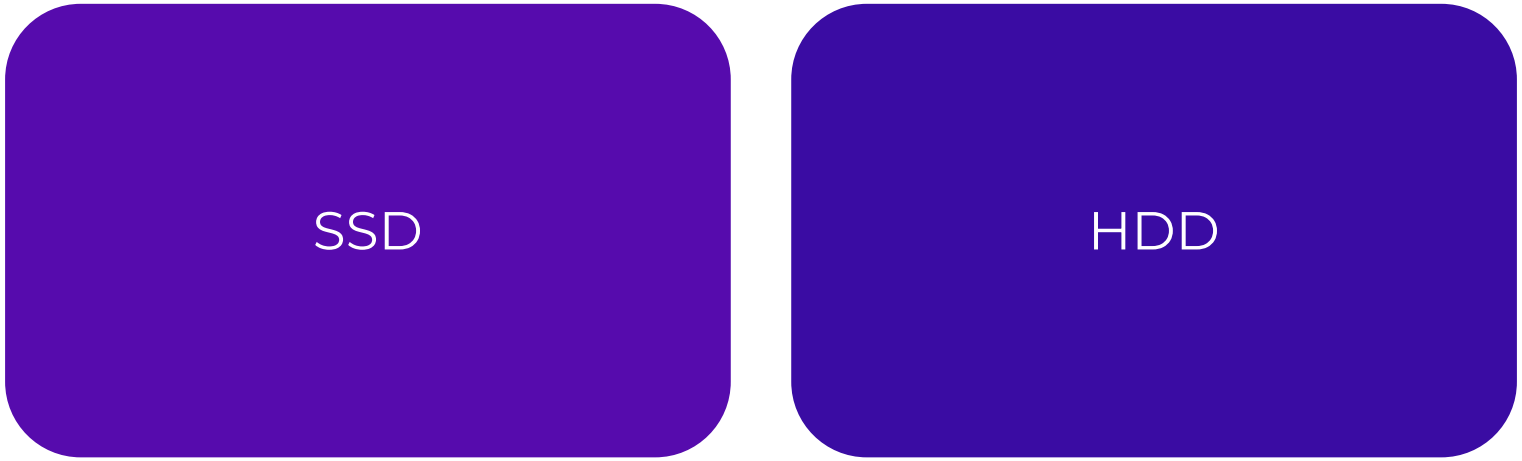
Jiyuan Zhang

Jongyul Kim, Chloe Alverti, Peizhe Liu, Weiwei Jia, and Tianyin Xu

UNIVERSITY OF ILLINOIS URBANA-CHAMPAIGN

THE UNIVERSITY OF RHODE ISLAND

Storage devices are diversifying



Storage devices are diversifying

New devices lead to heterogeneous configurations

We need to use the new devices AND the old ones.

Latest Features

Performance Advantage

Data Reliability

Backward Compatibility

New devices lead to heterogeneous configurations

Balanced Configuration

Performance Configuration

Capacity Configuration

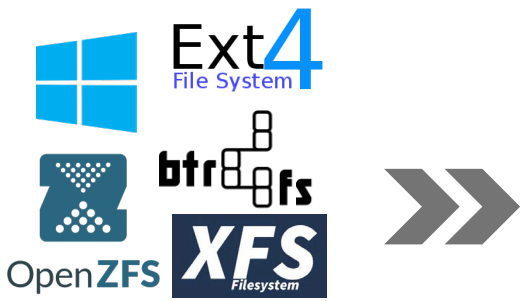
Archival Configuration

New devices lead to heterogeneous configurations

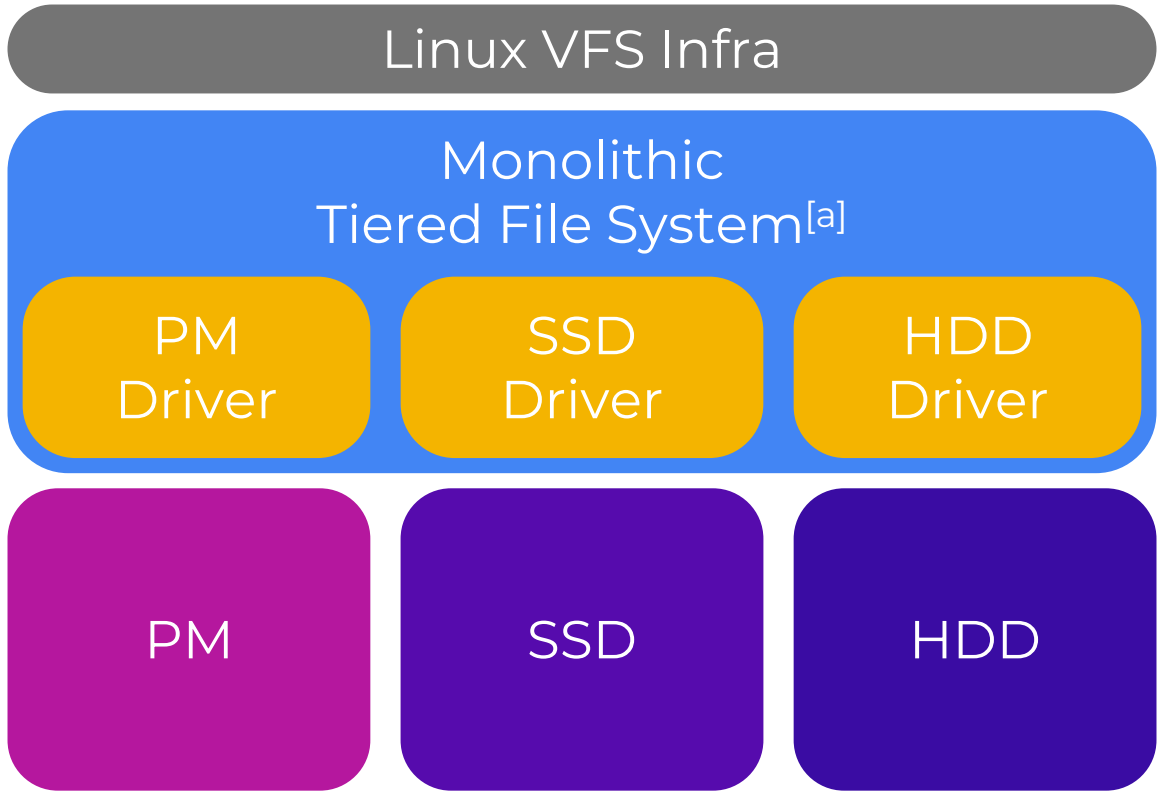
Research Question:

How to build **practical** file system
for **diverse** storage configurations?

Existing solutions are falling behind

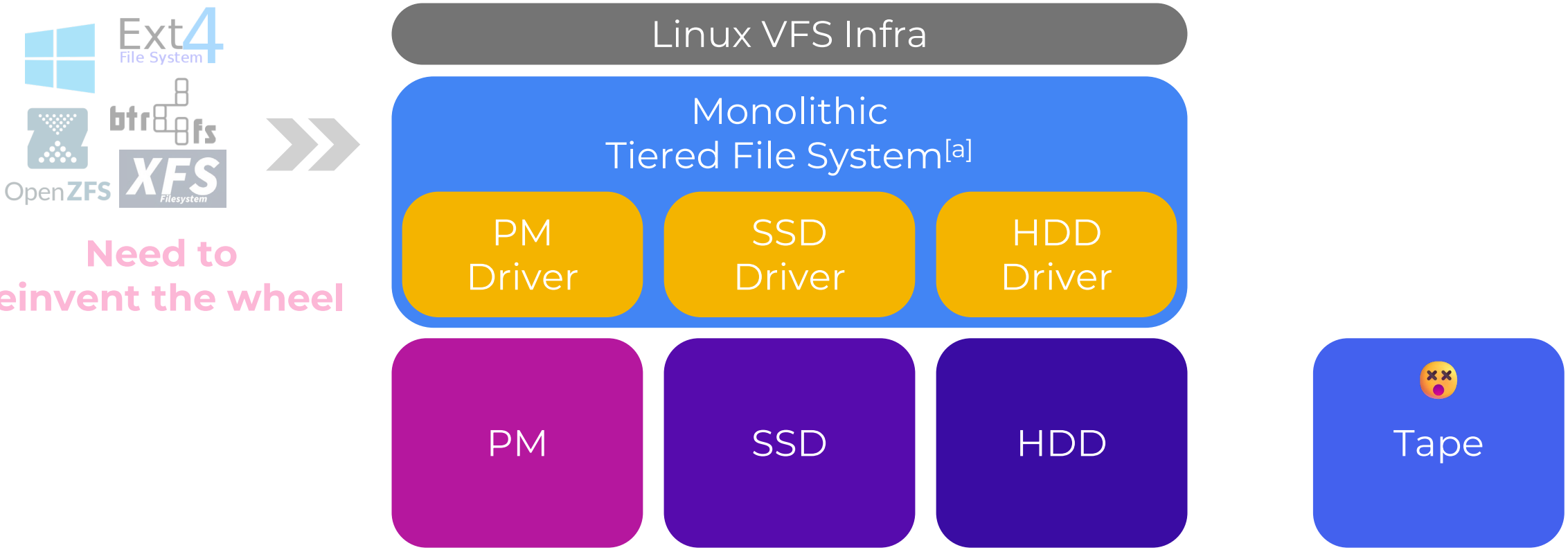


Need to reinvent the wheel



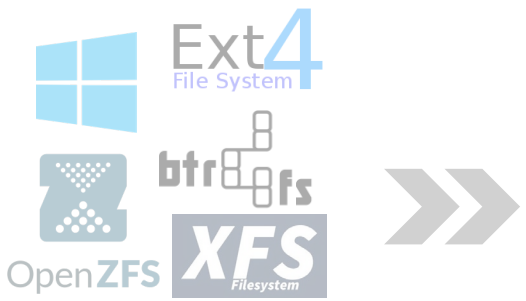
a. We observe the imperfections in the state-of-the-art monolithic tiered file system Strata

Existing solutions are falling behind

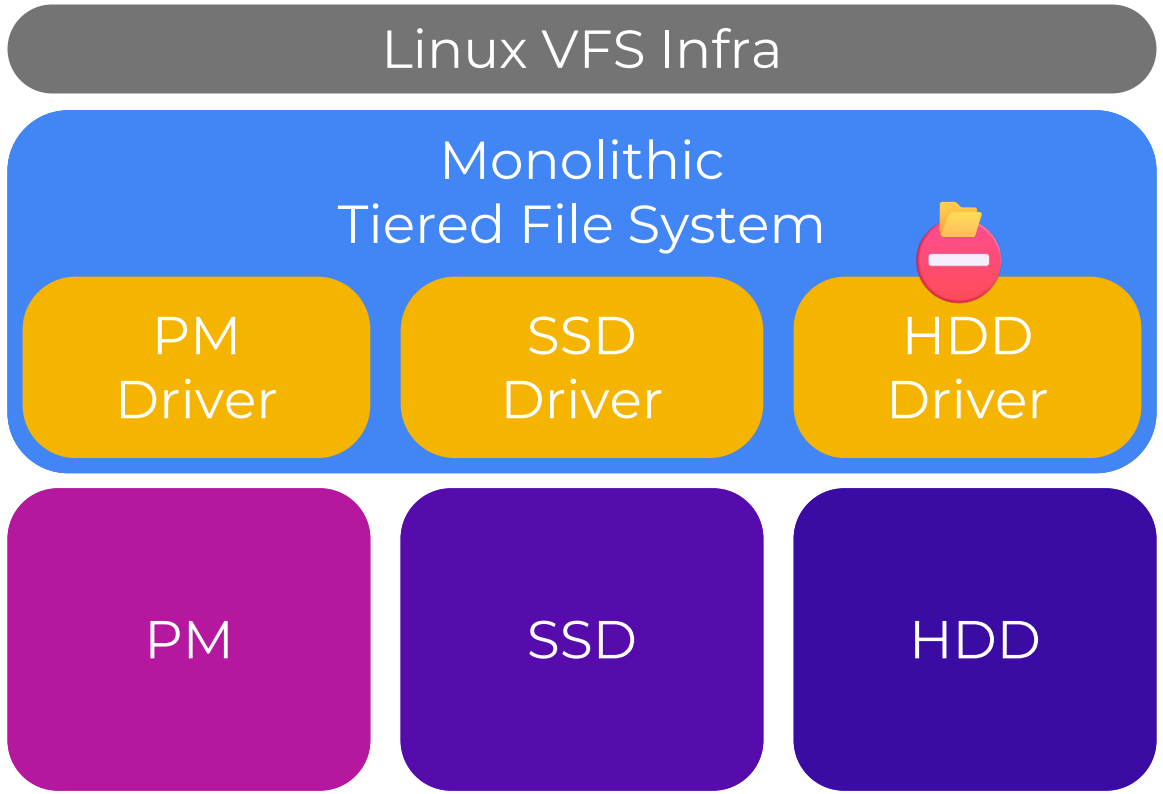


a. We observe the imperfections in the state-of-the-art monolithic tiered file system Strata

… and it introduces practical shortcomings



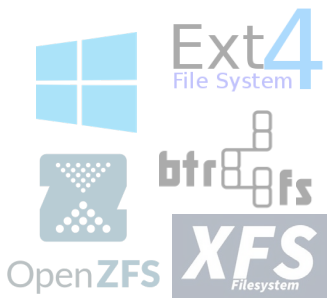
Need to reinvent the wheel



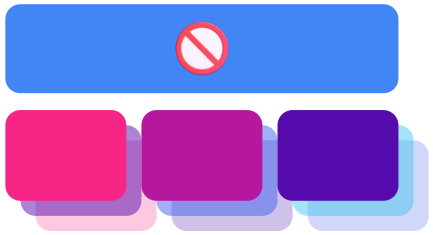
Lack of metadata compatibility

Hard to extend supported devices

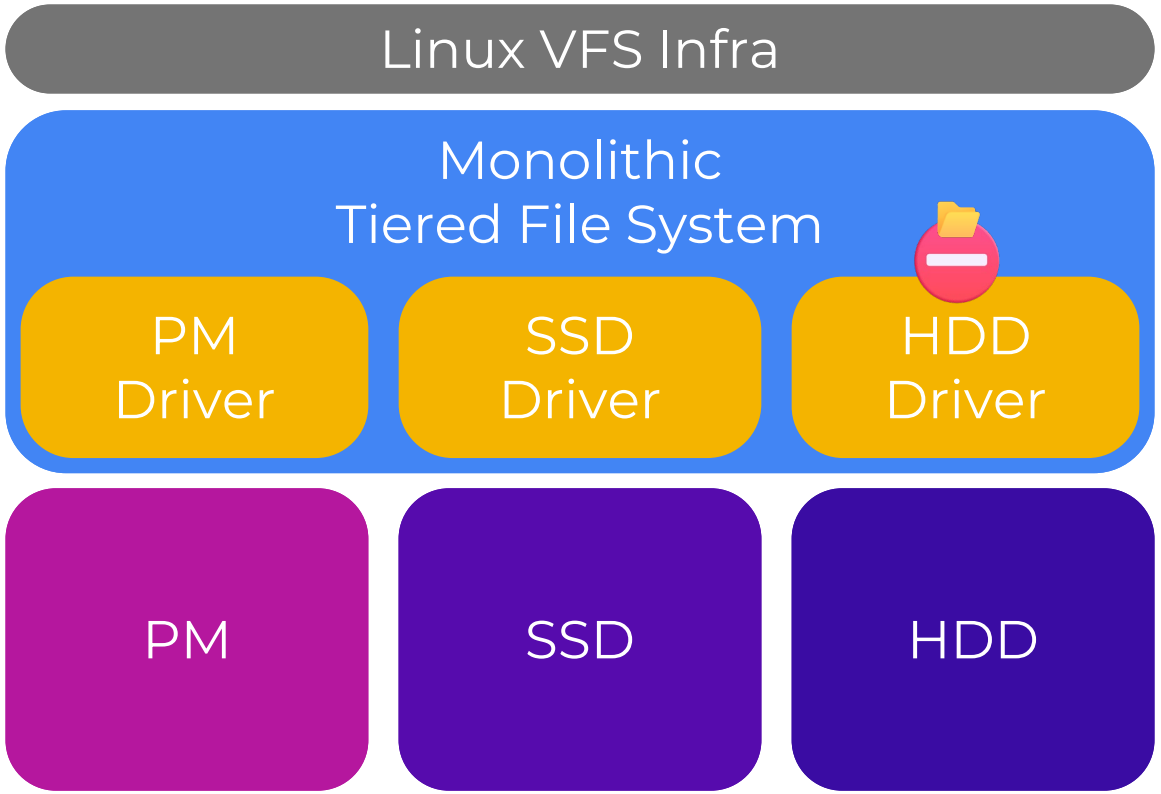
… and it introduces practical shortcomings



Need to reinvent the wheel



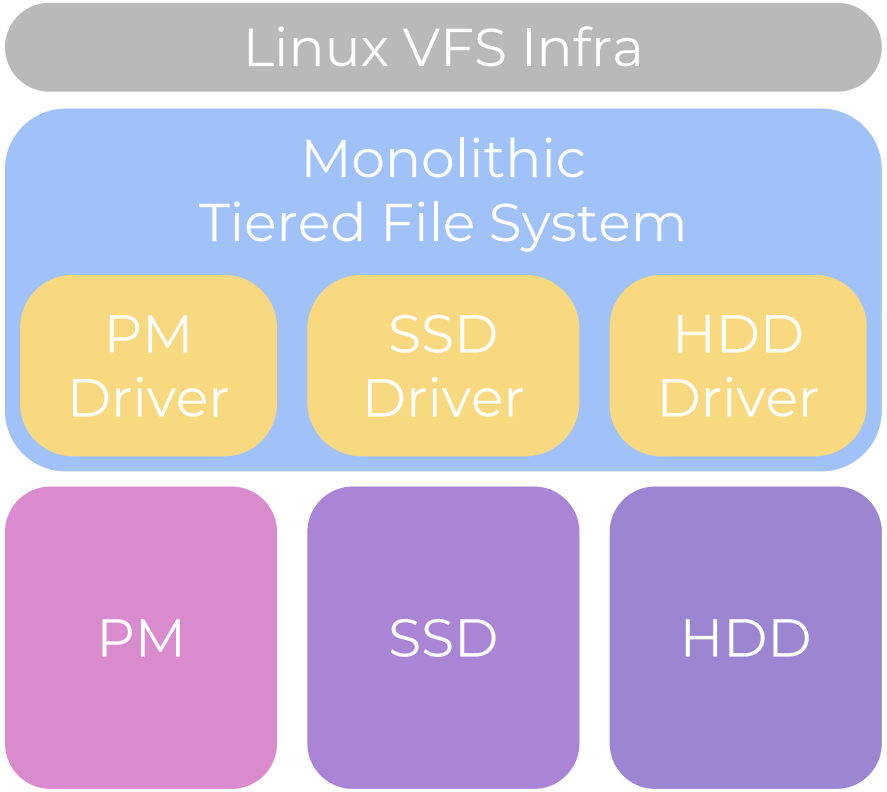
Poor reusability for other systems

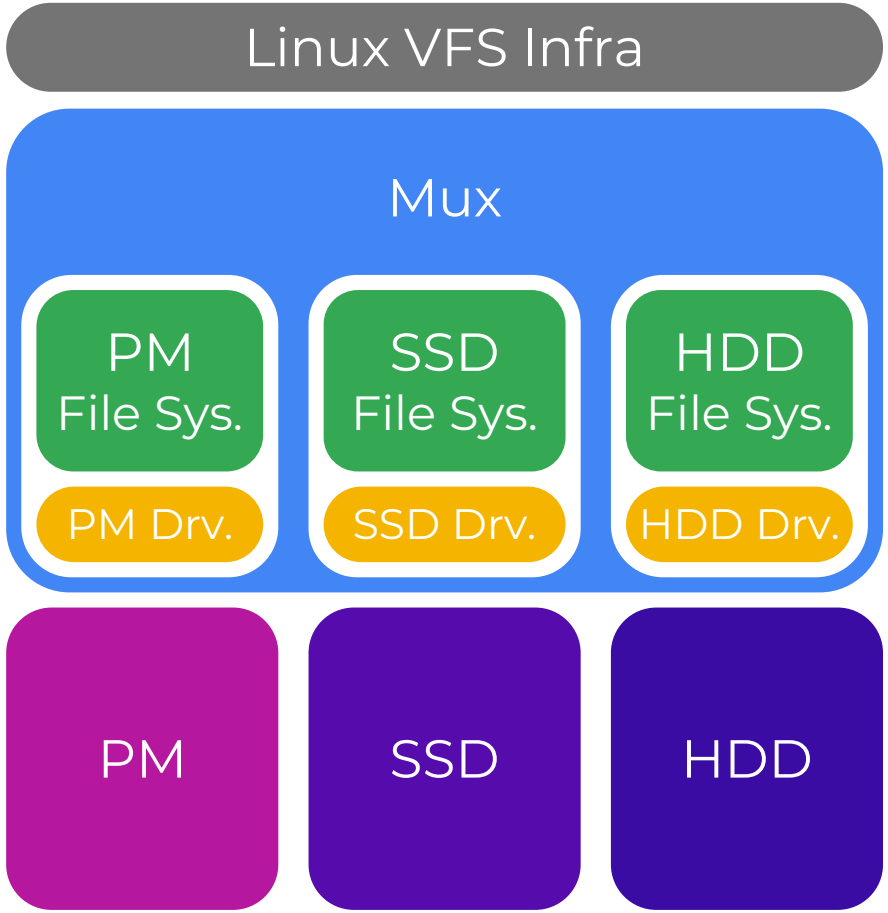


Lack of metadata compatibility

Hard to extend supported devices

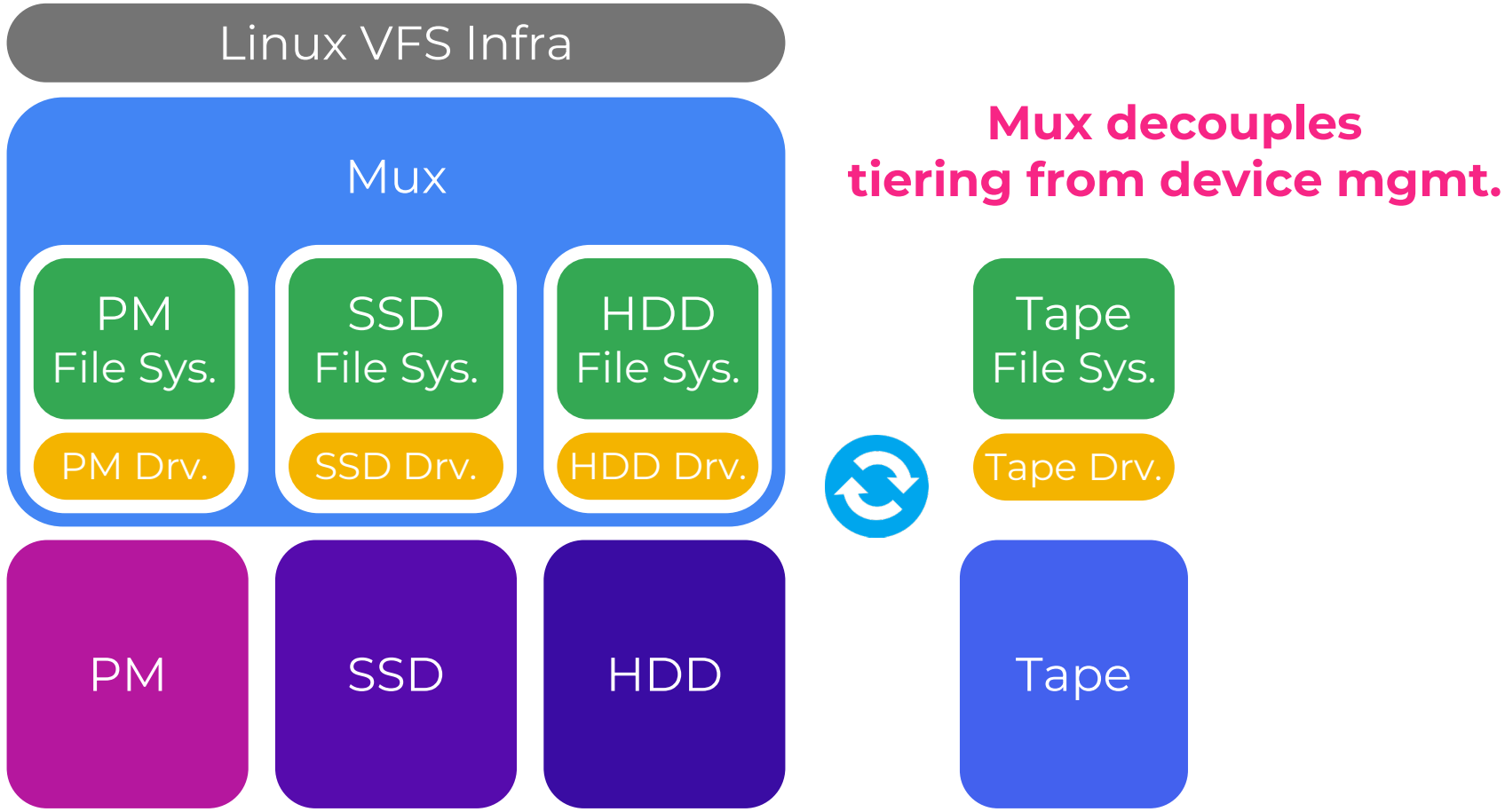
Mux: Talk to file systems, not device drivers





Our Contributions

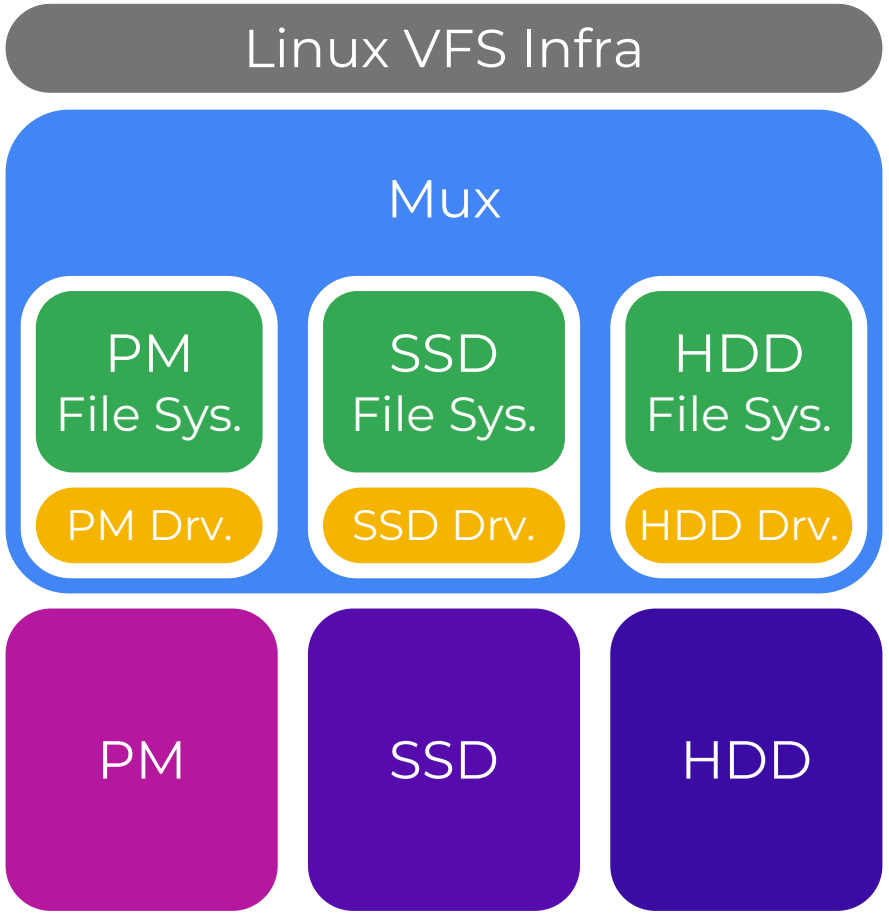
- ✅ **Flexibility**



Our Contributions

Mux can directly use latest file systems / devices

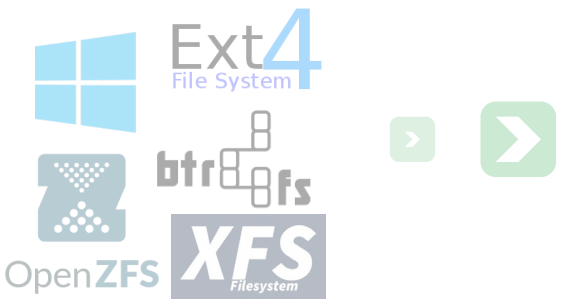
- ✅ **Flexibility**
- ✅ **Modularity**

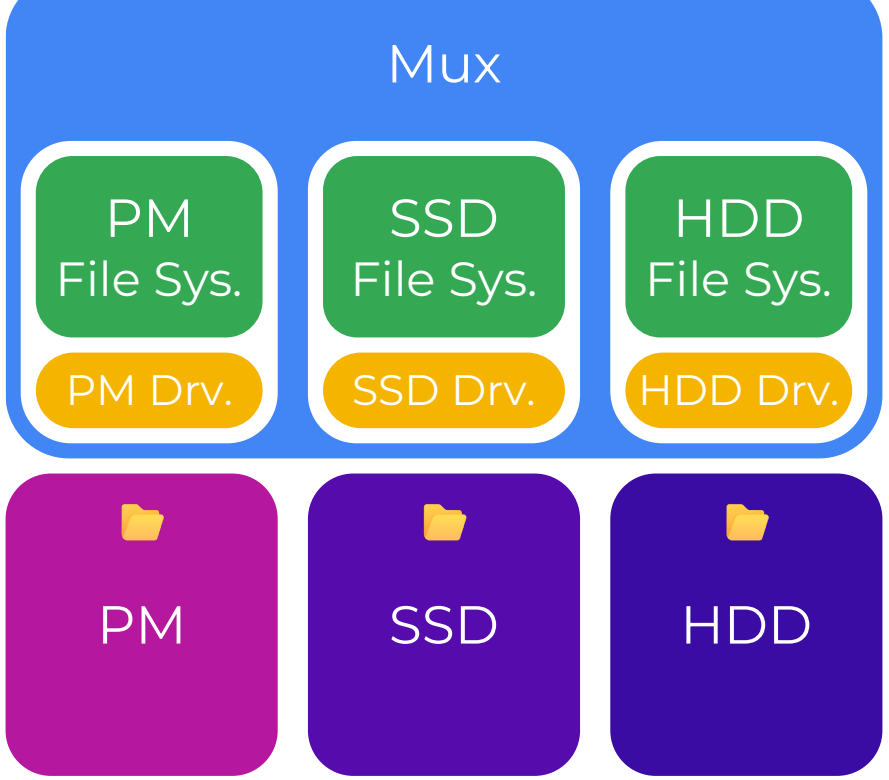


Mux decouples tiering from device mgmt.

Our Contributions

Mux can directly use latest file systems / devices



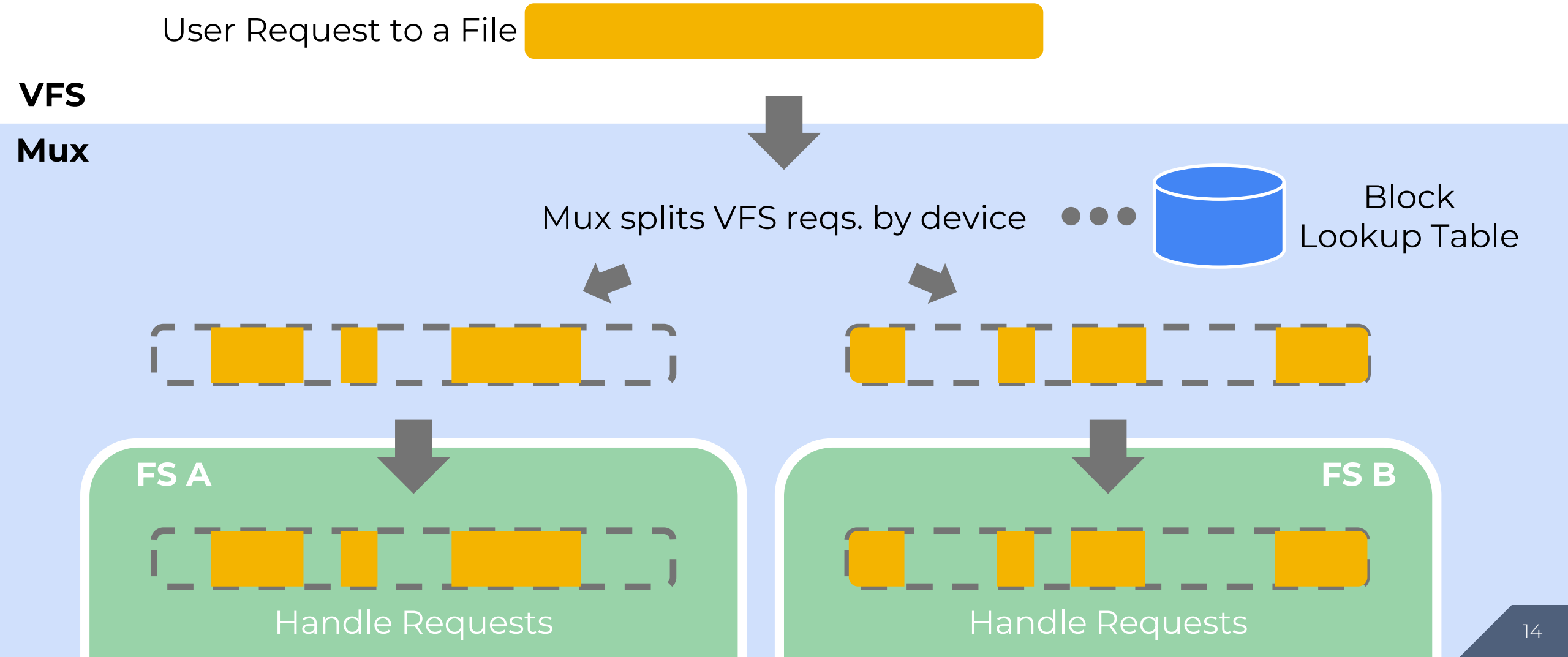


Mux decouples tiering from device mgmt.

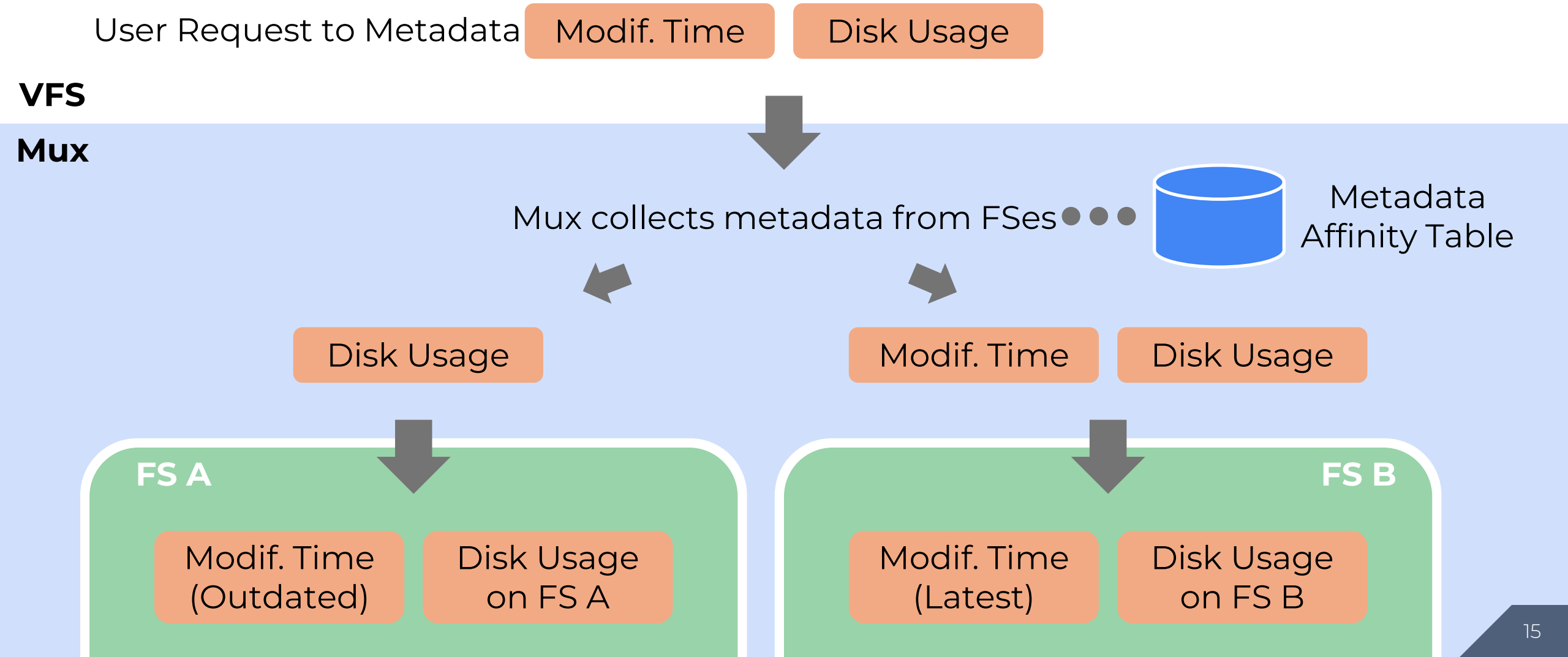
- ✅ **Flexibility**
- ✅ **Modularity**
- ✅ **Compatibility**

Mux supports tiering over existing storage in production

Dispatch user requests, not block I/Os



Track metadata location, not its content

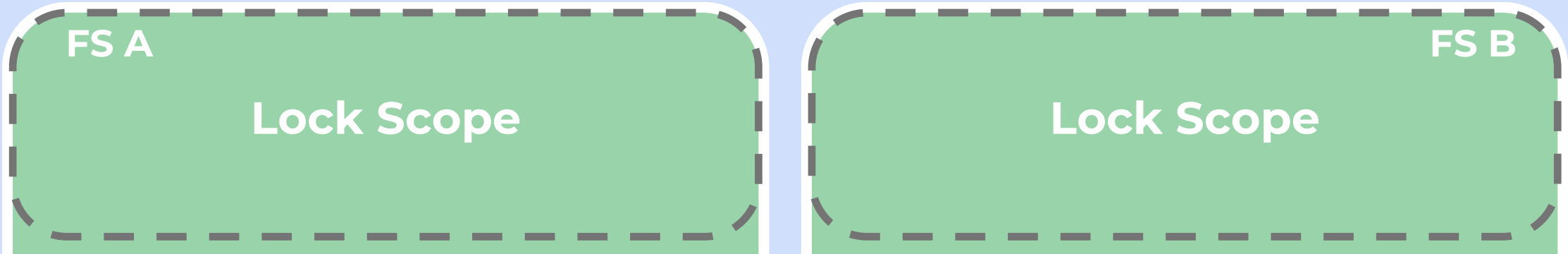


Migrate with versioning, not locking

VFS

Mux

File systems' lock scope
only applies to themselves internally

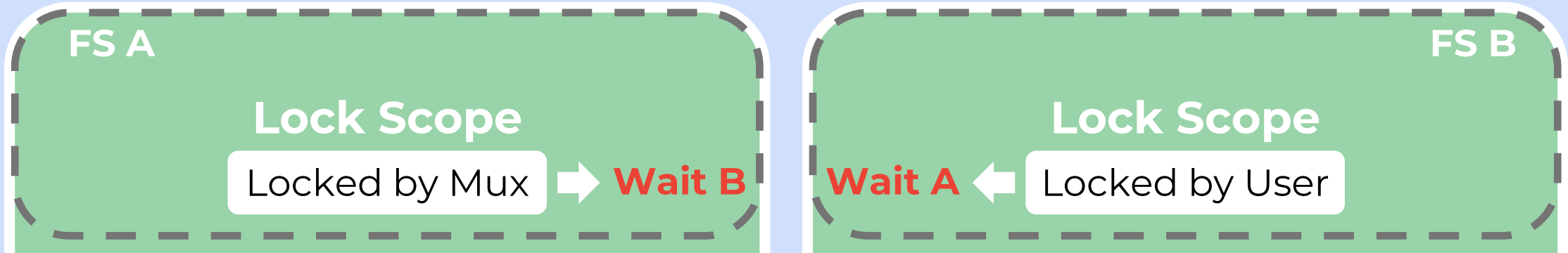


Migrate with versioning, not locking

VFS

Mux

Mux cannot utilize file system locks
due to potential deadlock with user locks

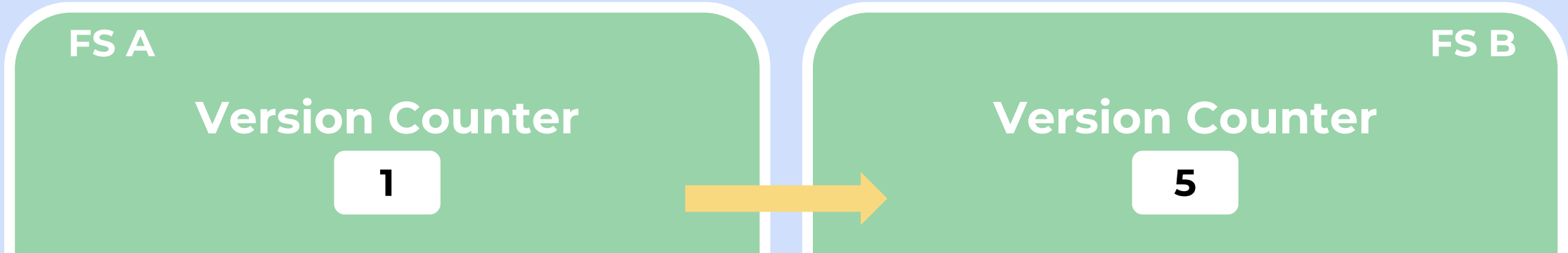


Migrate with versioning, not locking

VFS

Mux

Mux uses optimistic concurrency control
for both correctness and performance

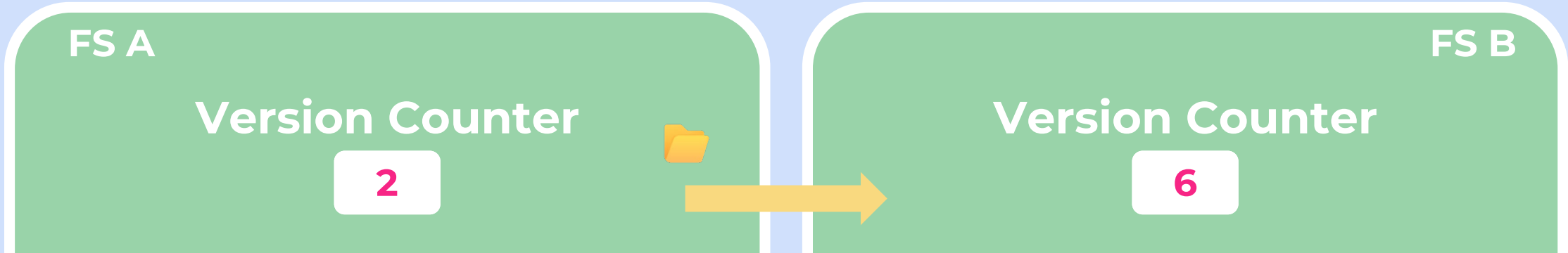


Migrate with versioning, not locking

VFS

Mux

Mux uses optimistic concurrency control
for both correctness and performance

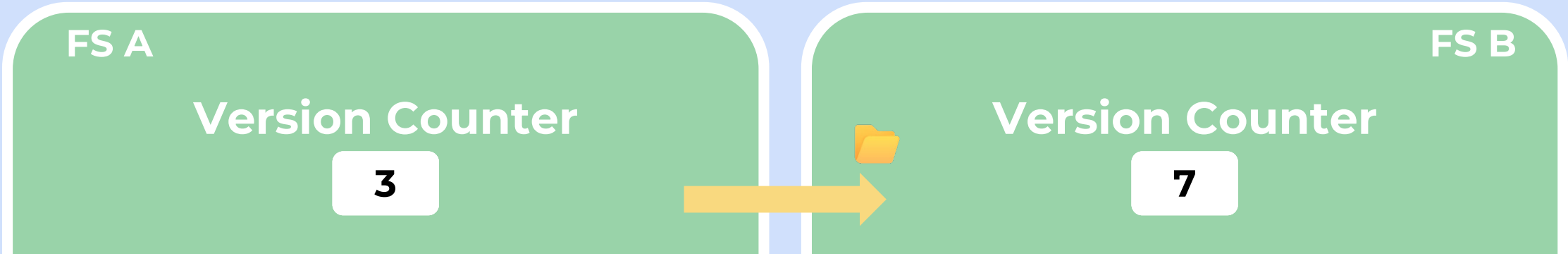


Migrate with versioning, not locking

VFS

Mux

Mux uses optimistic concurrency control
for both correctness and performance

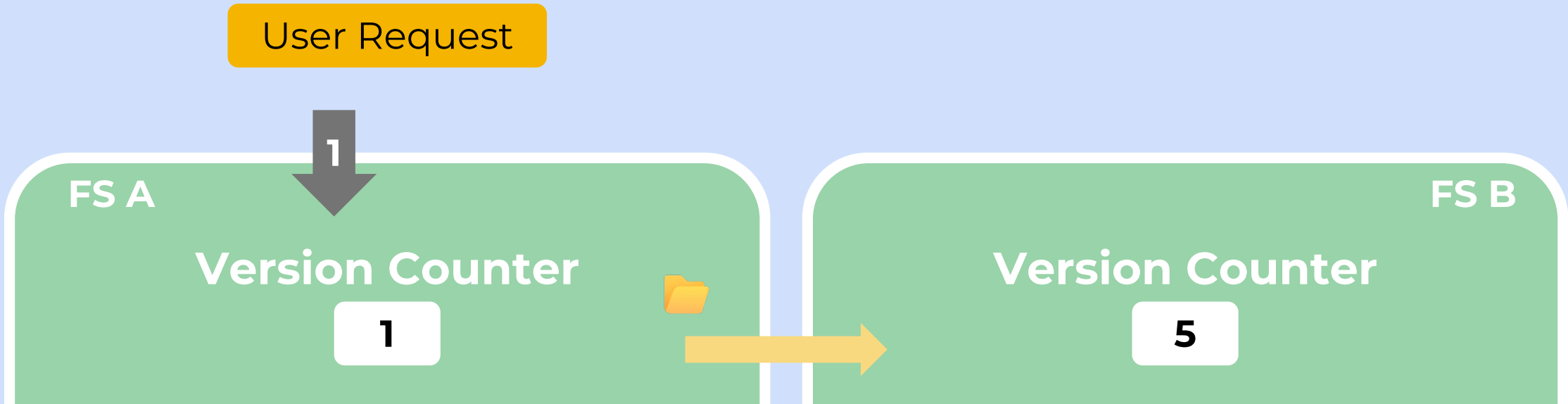


Migrate with versioning, not locking

VFS

Mux

Mux uses optimistic concurrency control
for both correctness and performance

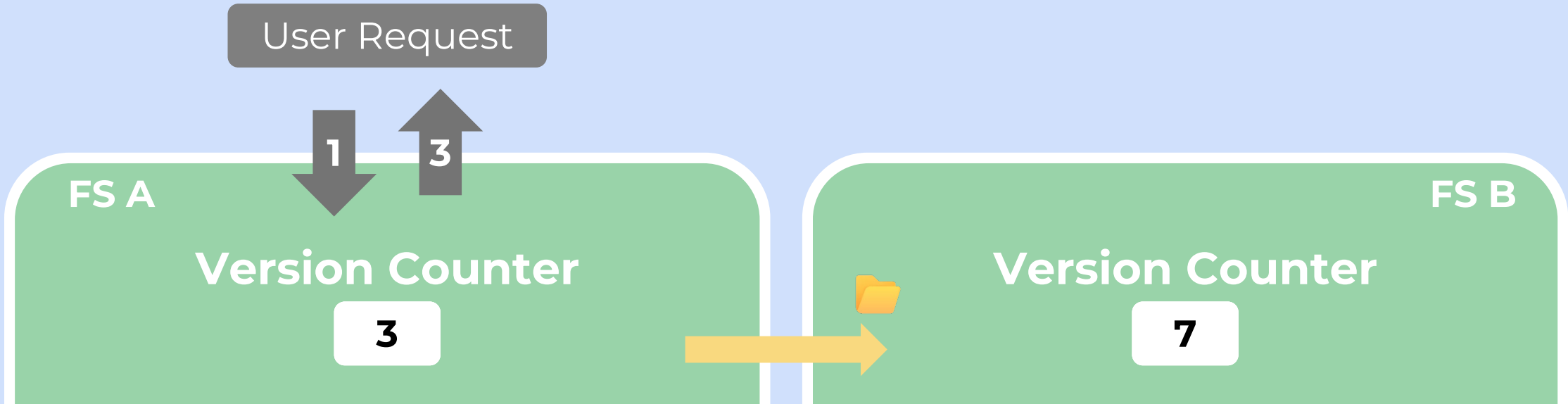


Migrate with versioning, not locking

VFS

Mux

Mux uses optimistic concurrency control
for both correctness and performance

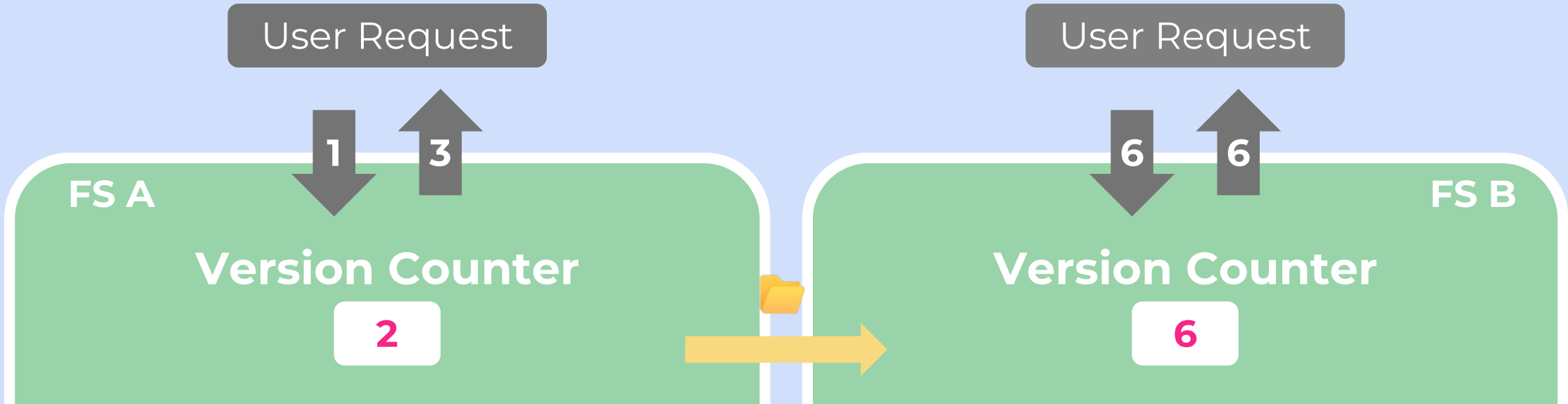


Migrate with versioning, not locking

VFS

Mux

Mux uses optimistic concurrency control
for both correctness and performance



Cache on DAX devices, not only DRAM

DRAM Memory

VFS

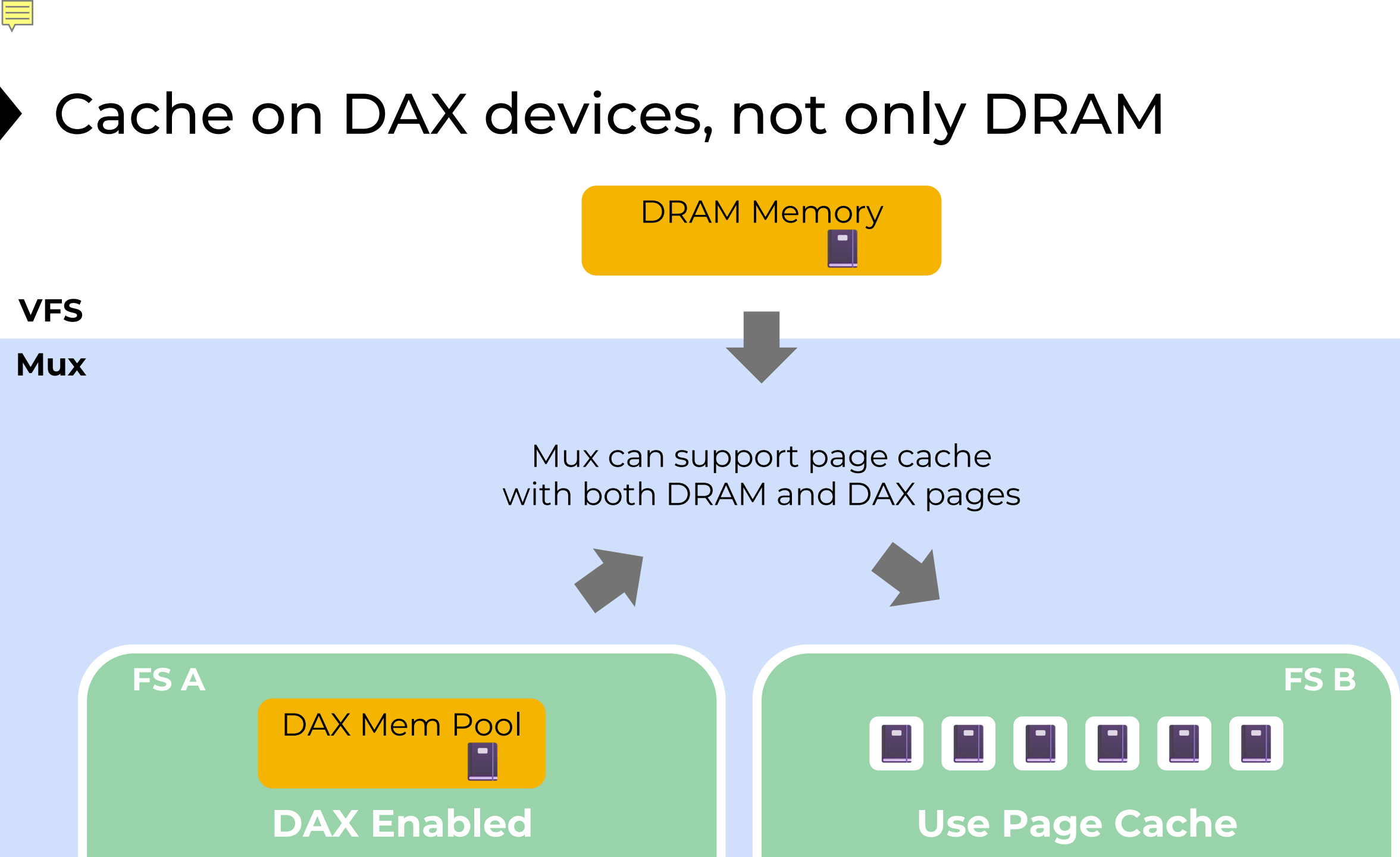
Mux

FS A

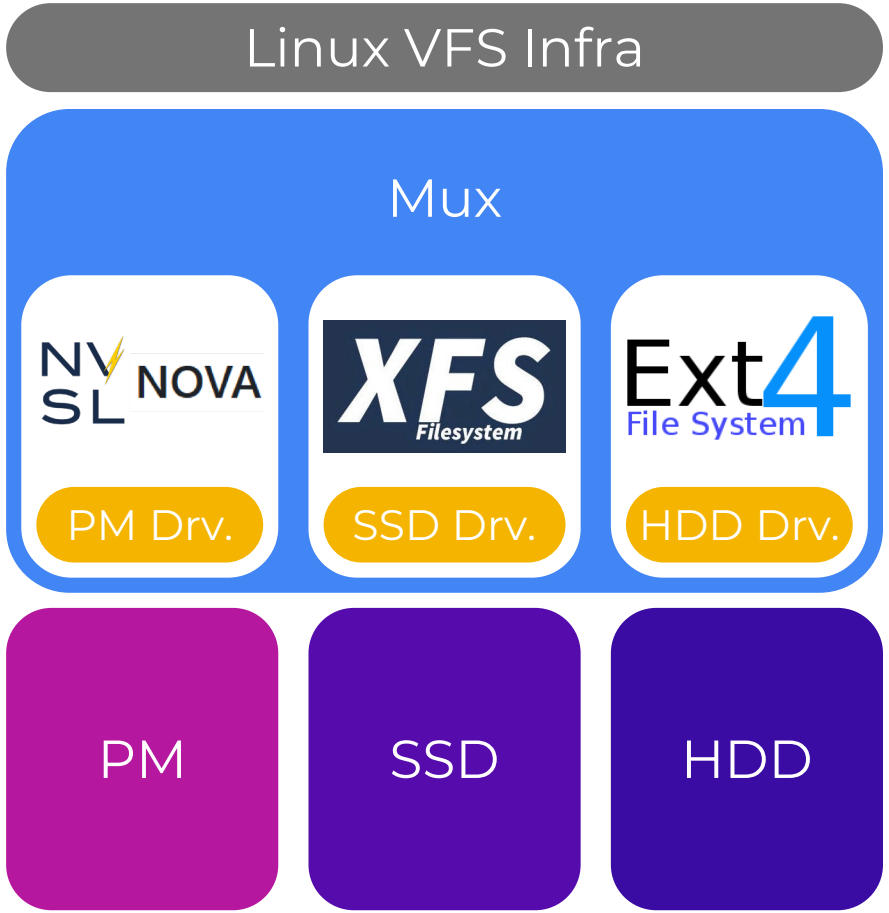
FS B

Use Page Cache

Cache on DAX devices, not only DRAM

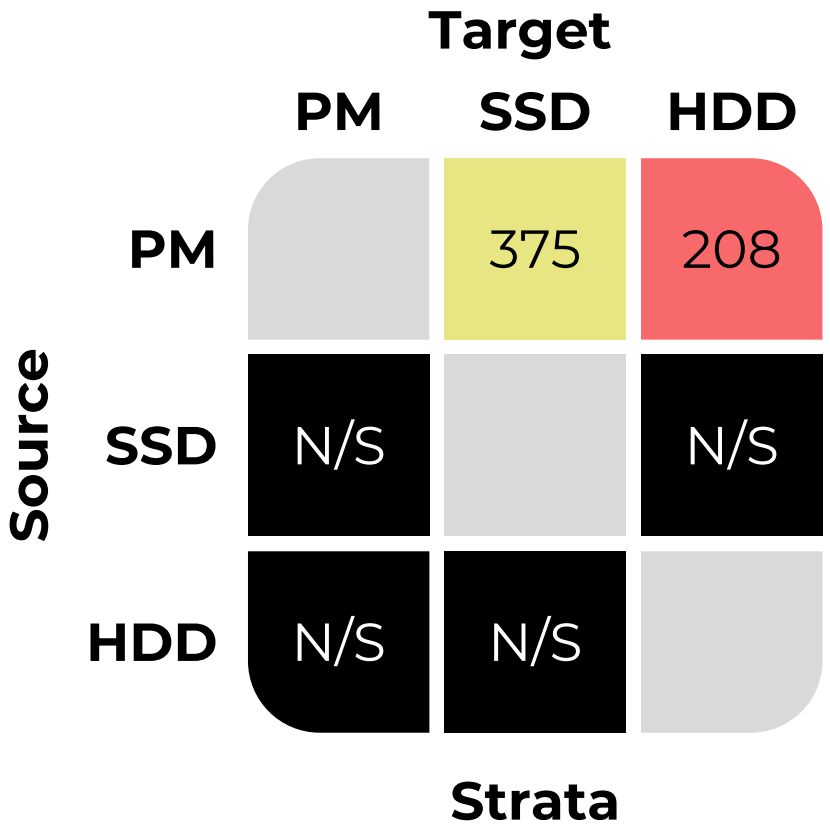


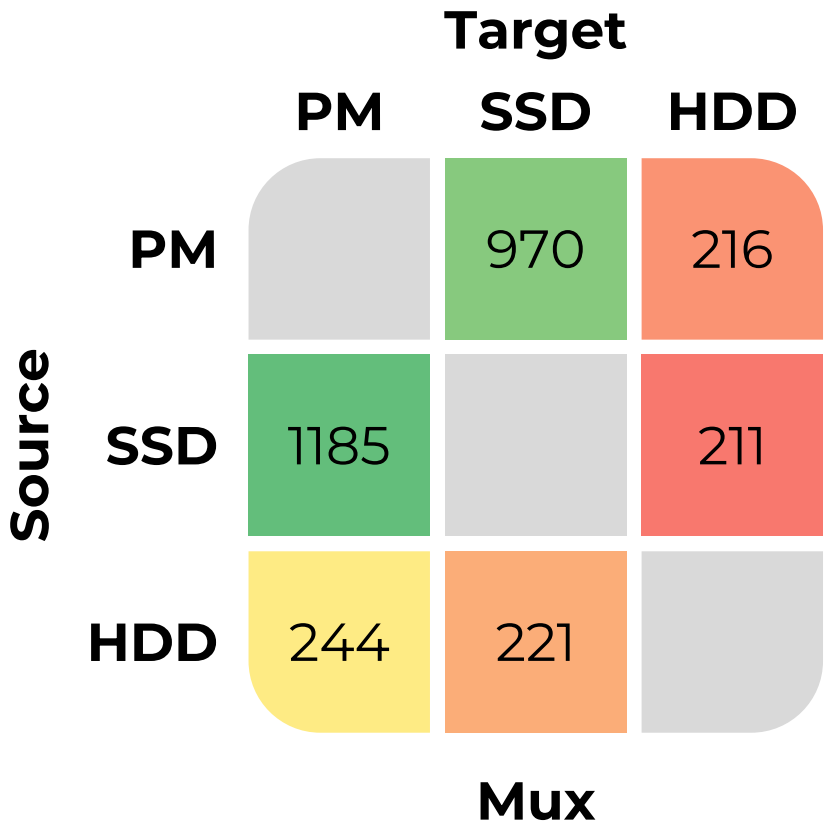
Evaluating a prototype of Mux



- **Extensibility** = 5 LoC / added device
- **Env** = Ubuntu 20.04 on Linux 5.15
- **Tool** = microbench from Strata
- **PM** = Intel Optane PM 200
- **SSD** = Intel Optane SSD DC P4800X
- **HDD** = Seagate Exos X18

Mux brings more tiering possibilities



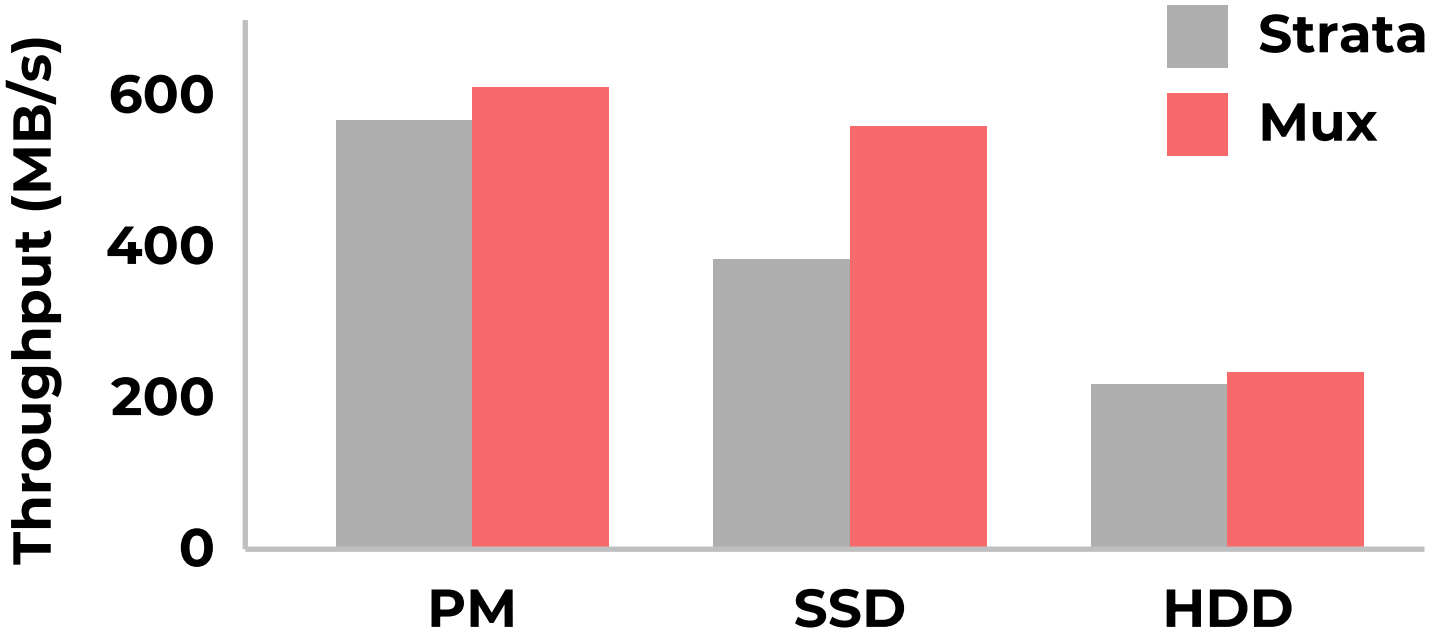


Mux supports all migration directions

Mux enables 2.59x faster migration via FS optimizations

Unit: MB/s
N/S: Not Supported

Mux makes high-performance achievable



Mux shows 1.46x higher throughput for device I/O

Mux adds a worst-case read latency overhead of 6.6% to 87.3% compared to non-tiered FSes

Mux adds a worst-case write throughput overhead of 1.6% to 3.5% compared to non-tiered FSes

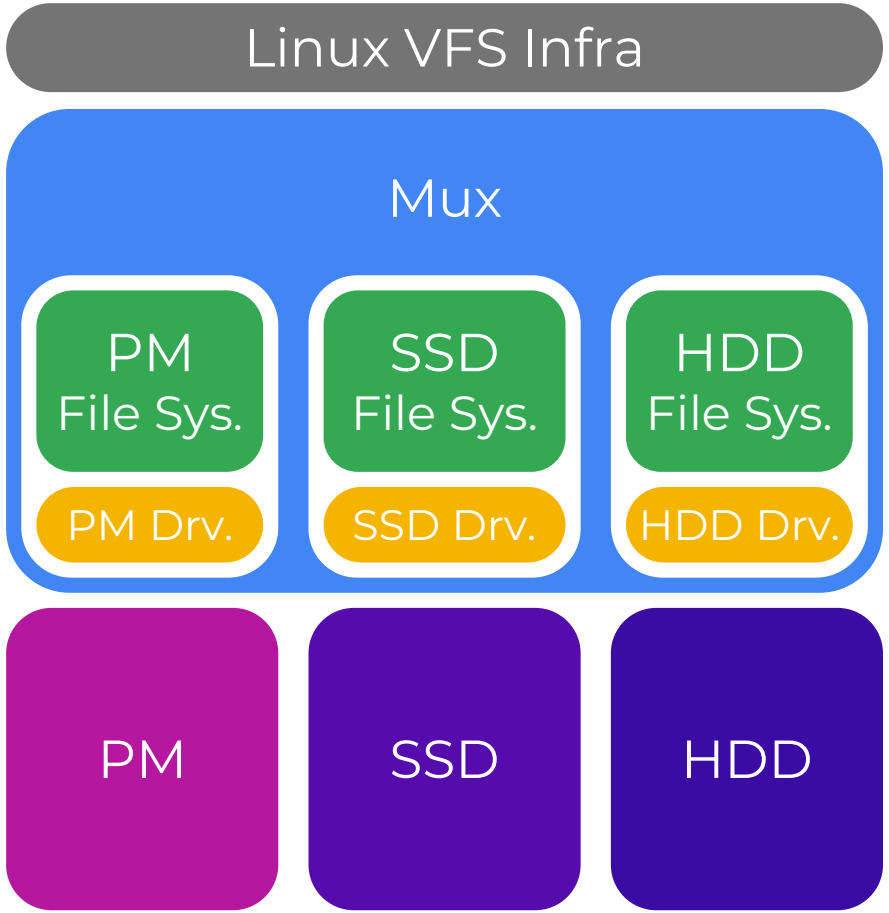
Future work: Mux can be further improved

- **Metadata** could be better handled to reduce overhead, e.g., read latency.
- **Crash consistency** can be enhanced beyond the lowest of all member file systems.
- **Feature imparity** among file systems could be preserved for better compatibility.

Future work: Mux enables new storage research

- **Scheduling** of file I/O could now be done at VFS level with Mux, rather than block level.
- **Configuration** of file systems for a given workload or a given set of storage devices can now be studied with Mux.
- **Distributed Mux** can be designed to perform tiering among an array of machines over network.

Conclusion



- **Rethinking Tiered Storage: Talk to File Systems, Not Device Drivers**
- **Mux** is a new tiered file system that **accesses devices through file systems**, rather than device drivers.
- **Extensibility** as a first-class principle to decouple tiering from device mgmt.
- **1.46x** higher throughput for device I/O; **2.59x** faster data migration over Strata.