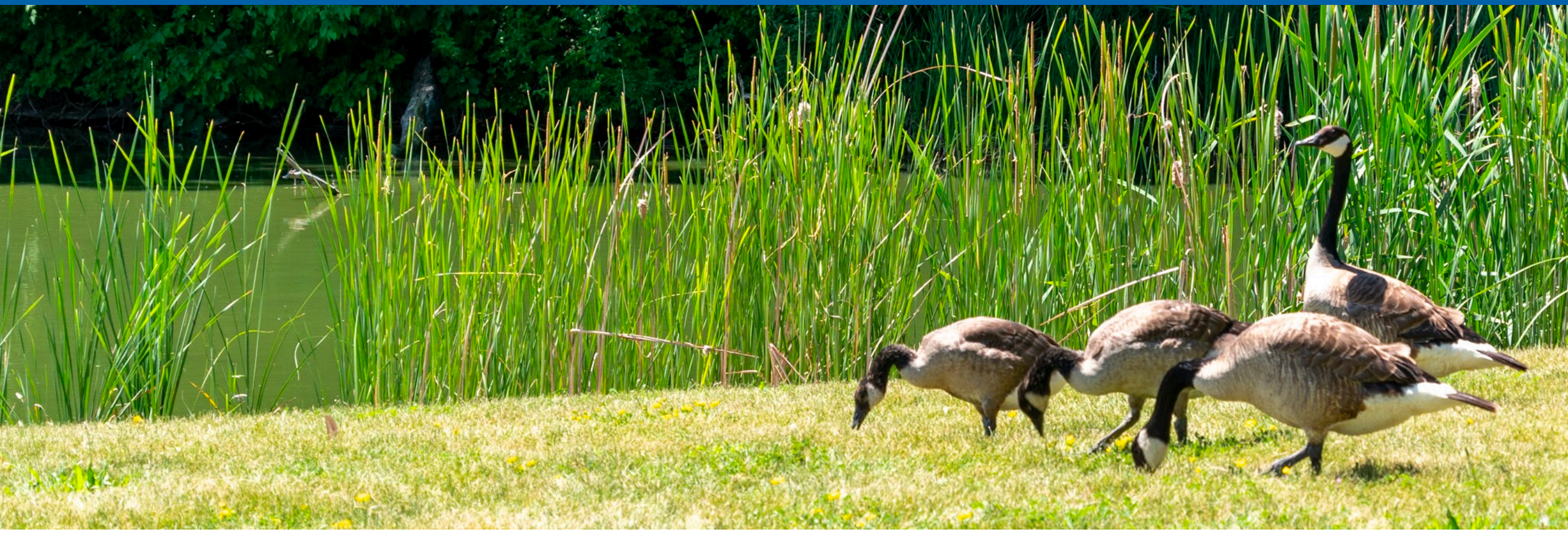


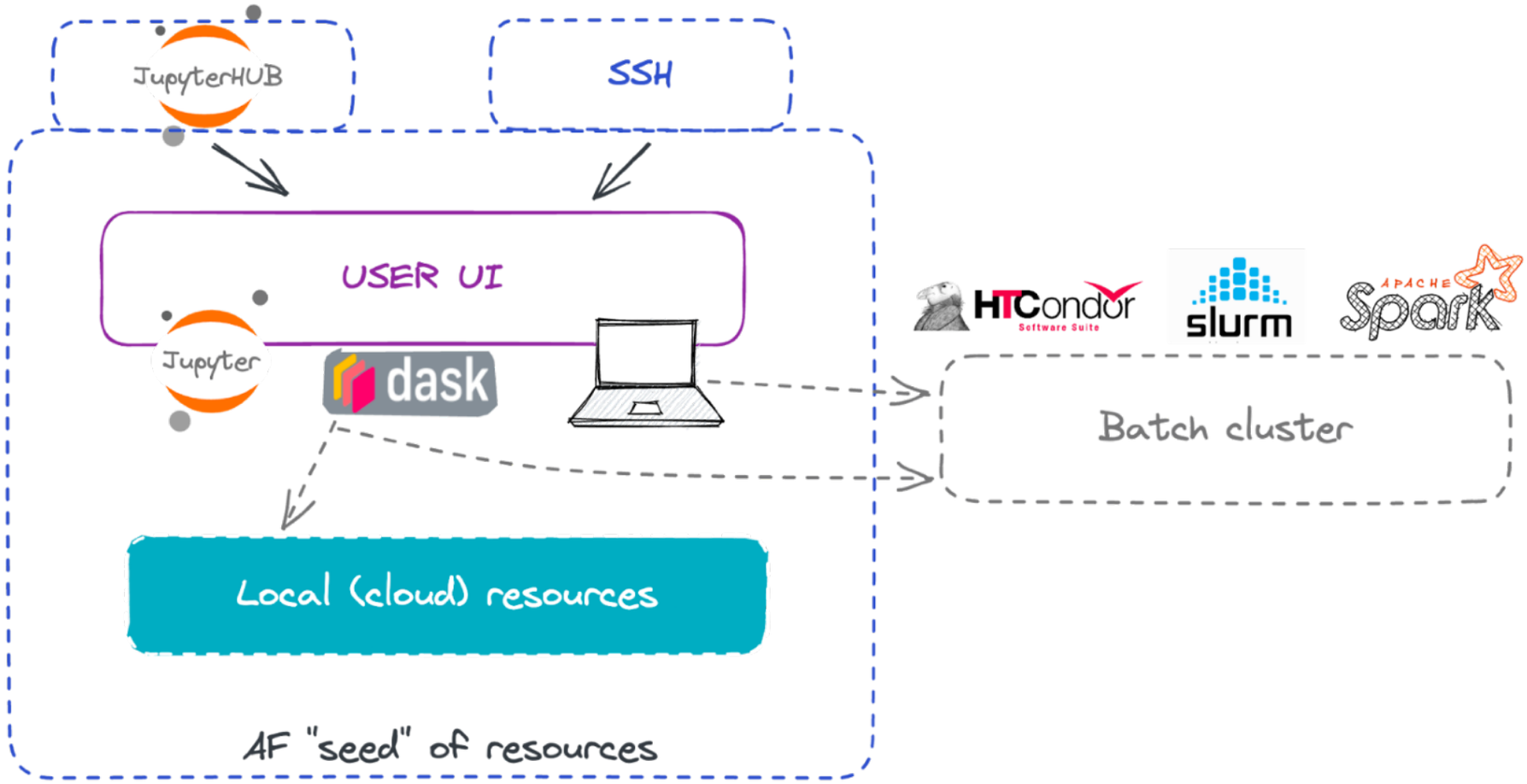
Image credit: Marguerite Tonjes

Storage for Analysis Facilities

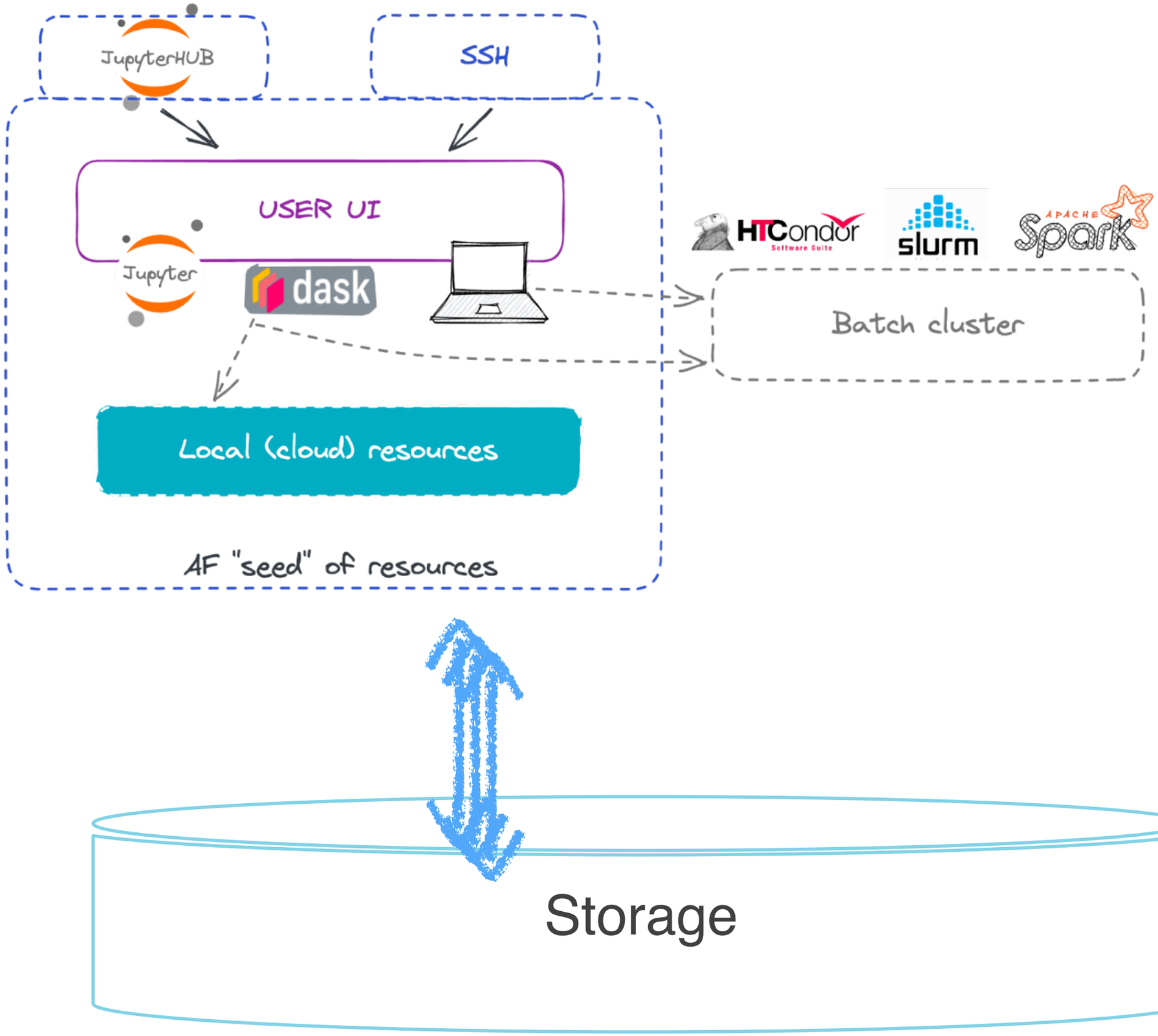
What would change if we tossed out POSIX?

Dirk Hufnagel, Nick Smith

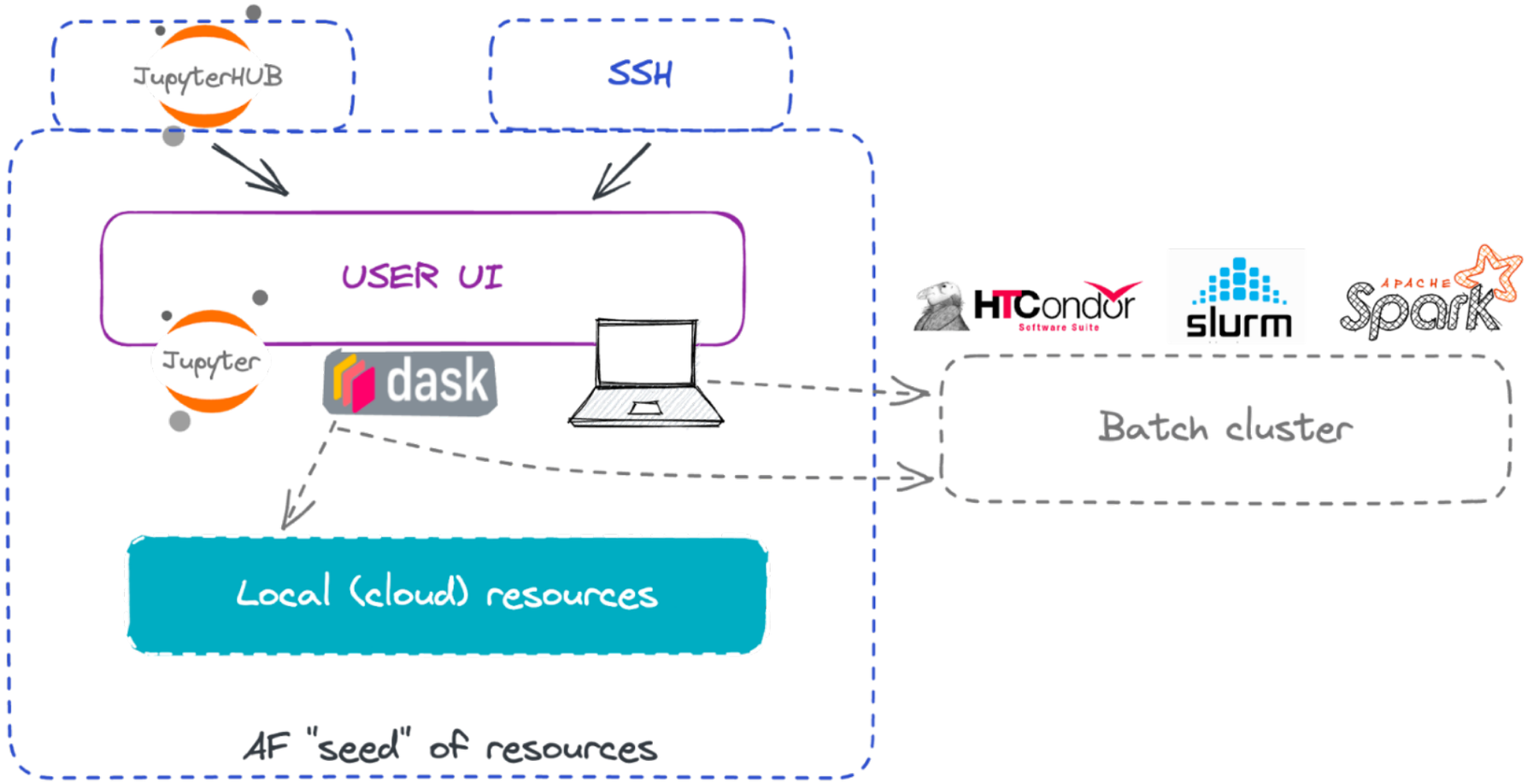
Facility view



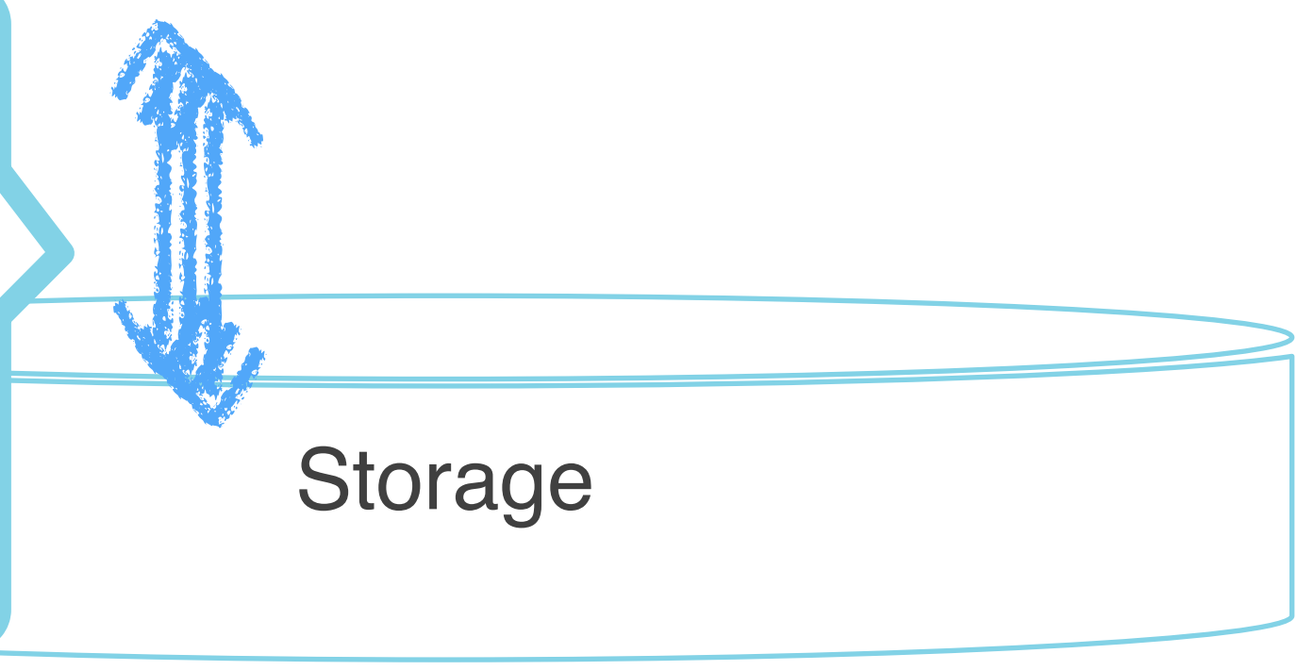
Facility view



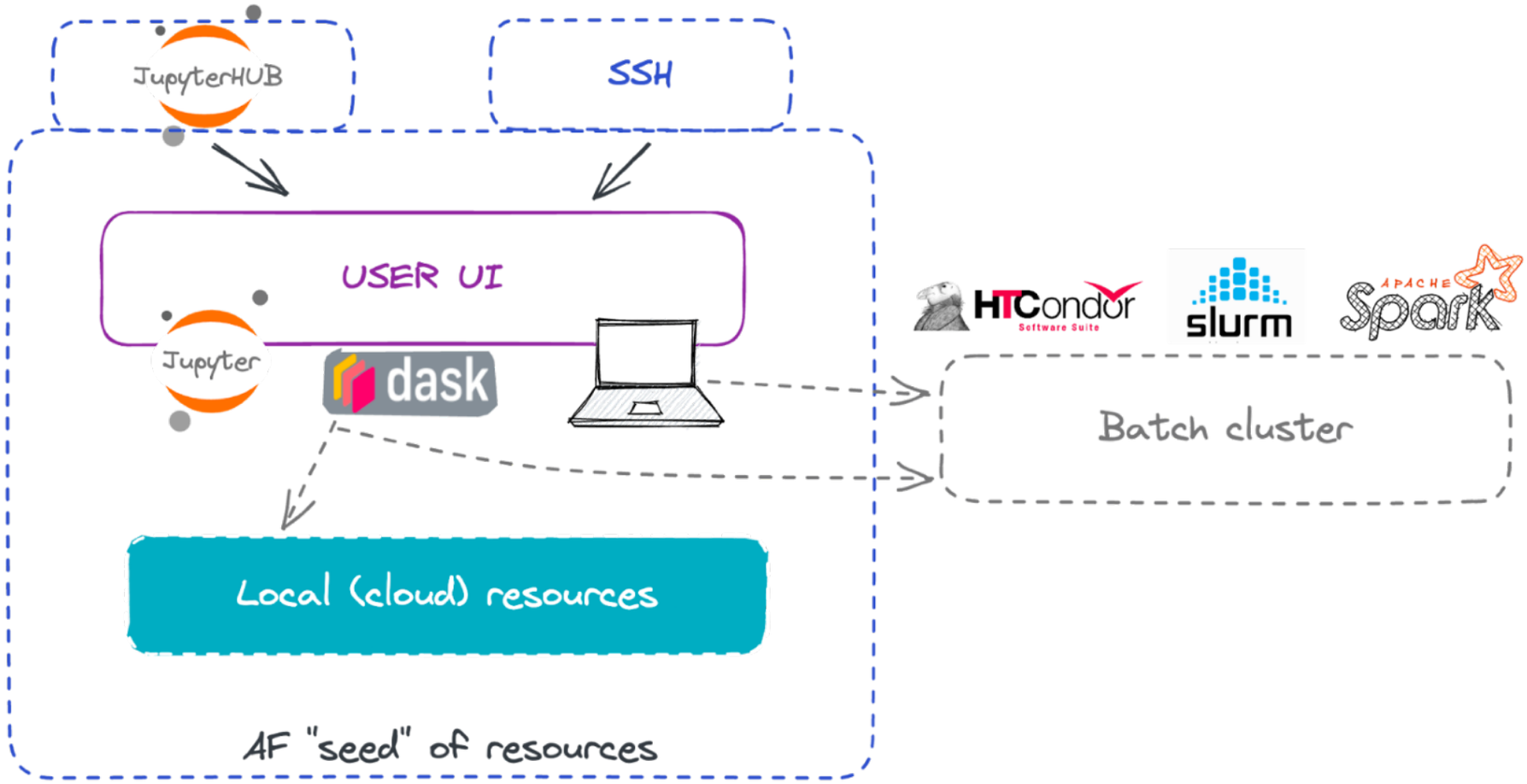
Facility view



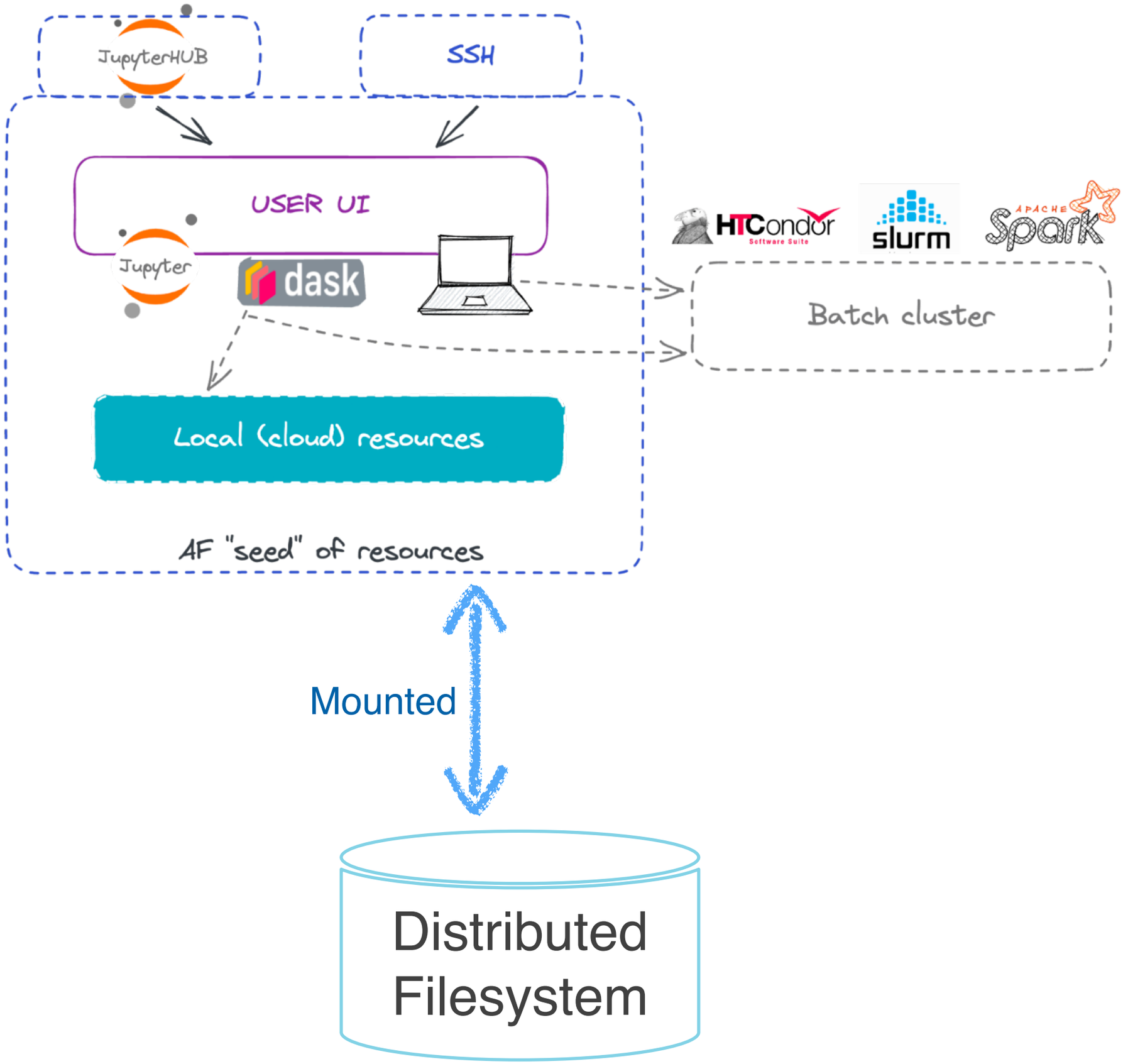
- Small data (kB-GB)
  - User code, calibration payloads, output histograms, …
- Medium data (GB-TB)
  - Intermediate datasets (skims)
- Large data (TB-PB)
  - Input datasets



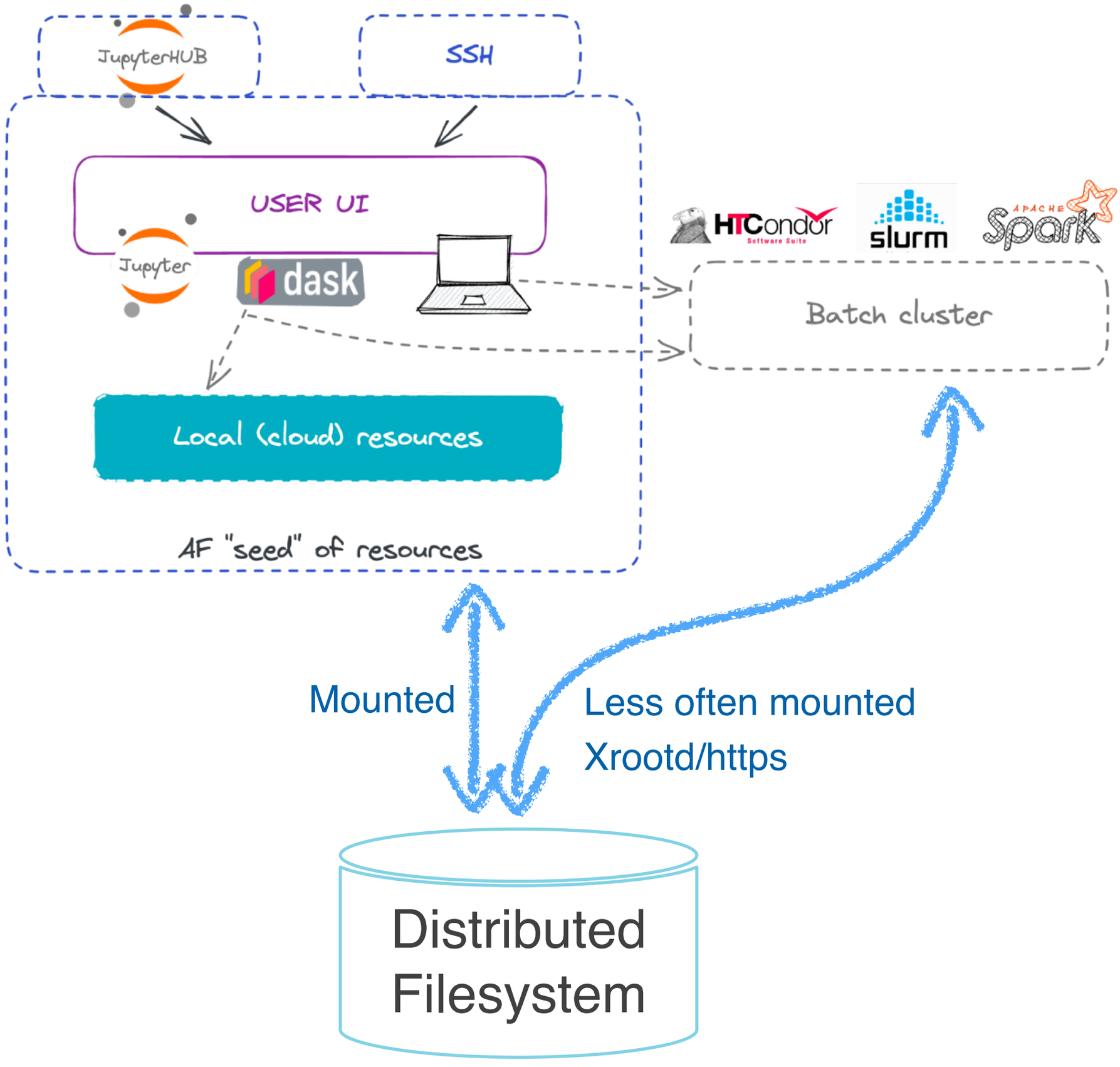
Small data


JupyterHUB
SSH
USER UI
Jupyter
dask
HTCondor
Software Suite
slurm
APACHE
Spark
Batch cluster
Local (cloud) resources
AF "seed" of resources

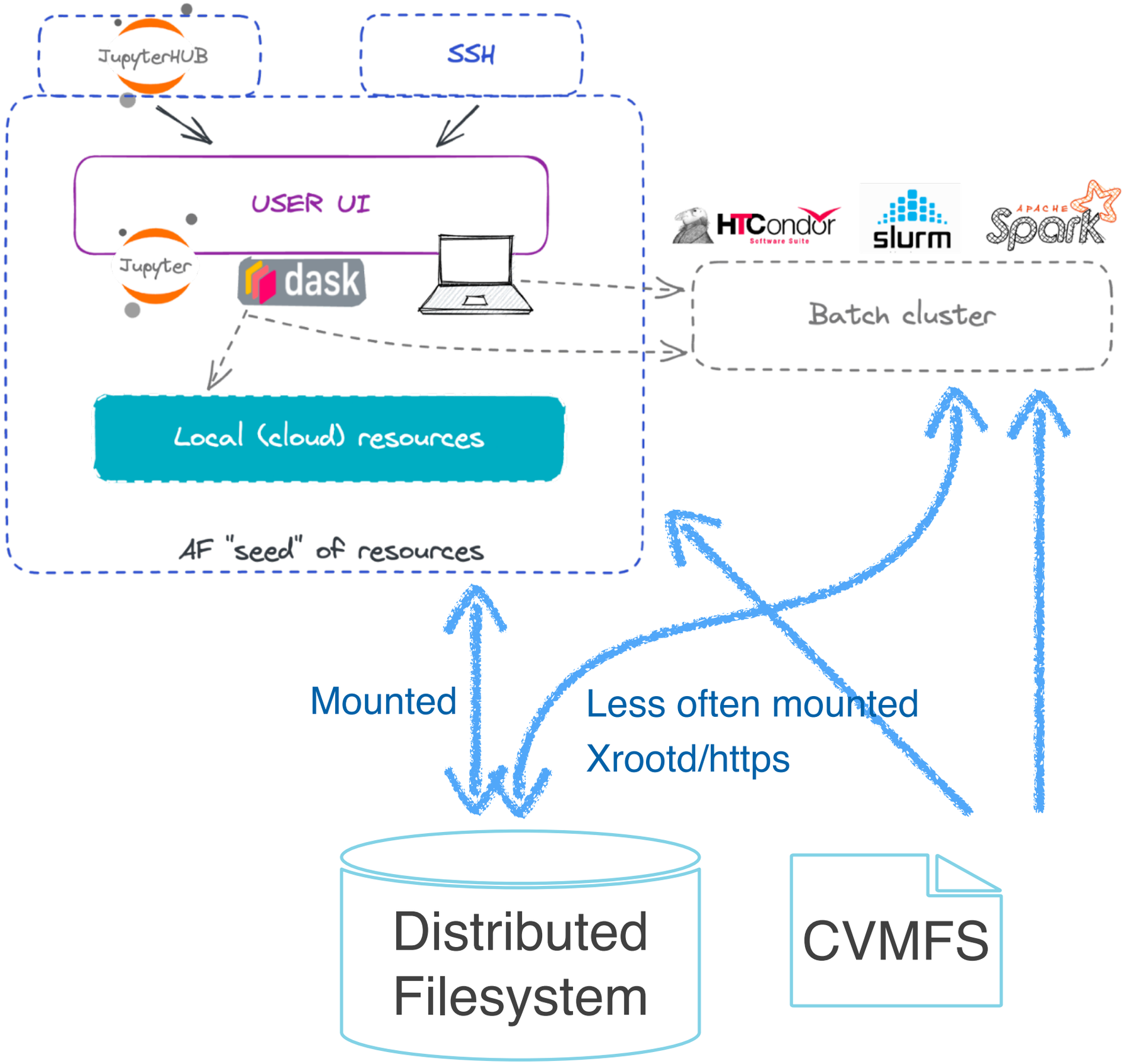
Small data


JupyterHUB
SSH
USER UI
Jupyter
dask
HTCondor
Software Suite
slurm
APACHE
Spark
Batch cluster
Local (cloud) resources
AF "seed" of resources
Mounted
Distributed
Filesystem

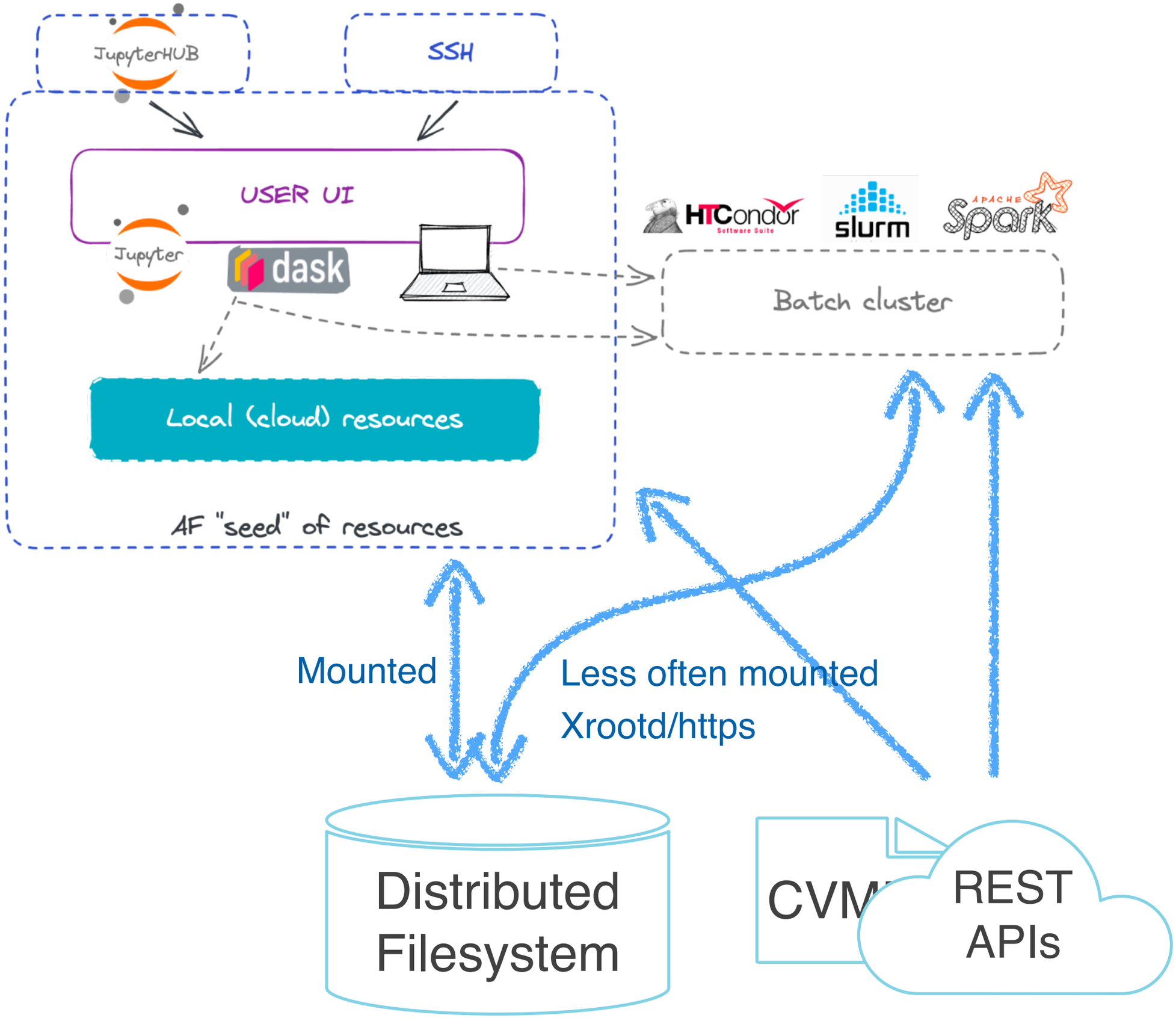
Small data


JupyterHUB
SSH
USER UI
Jupyter
dask
HTCondor
Software Suite
slurm
APACHE
Spark
Batch cluster
Local (cloud) resources
AF "seed" of resources
Mounted
Less often mounted
Xrootd/https
Distributed
Filesystem

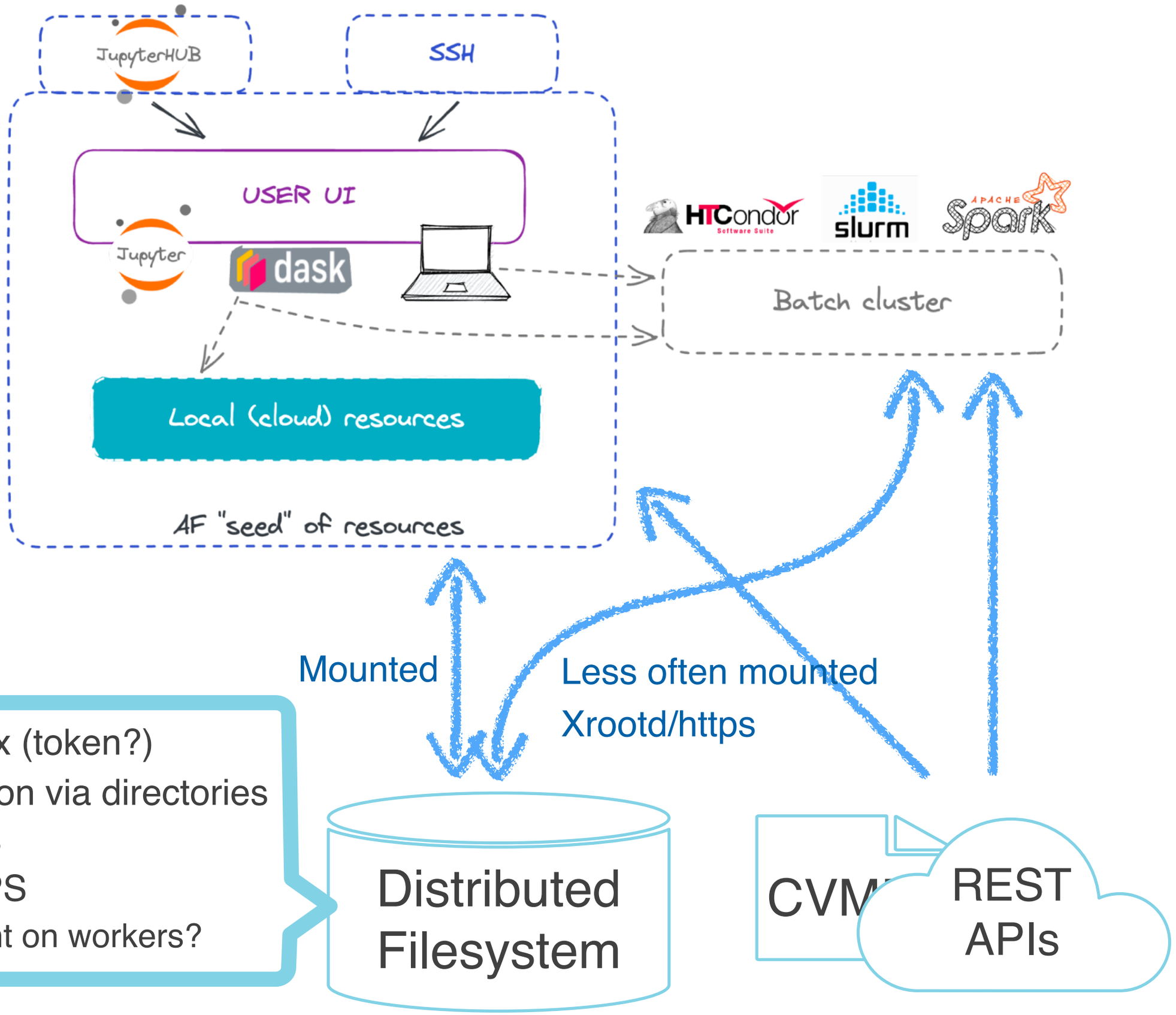
Small data


JupyterHUB
SSH
USER UI
Jupyter
dask
HTCondor
Software Suite
slurm
APACHE
Spark
Batch cluster
Local (cloud) resources
AF "seed" of resources
Mounted
Less often mounted
Xrootd/https
Distributed
Filesystem
CVMFS

Small data

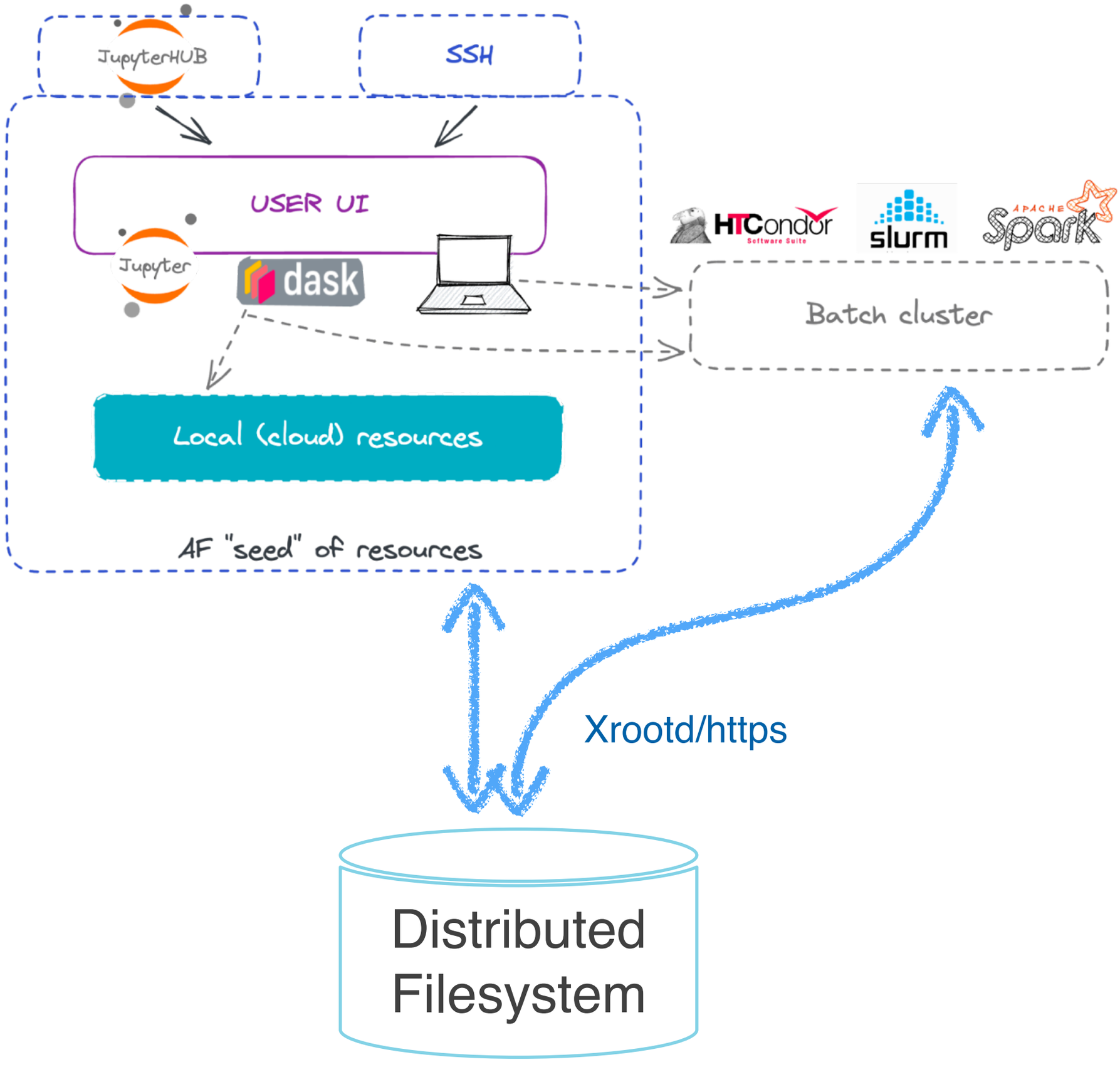

JupyterHUB
SSH
USER UI
Jupyter
dask
HTCondor
Software Suite
slurm
APACHE
Spark
Batch cluster
Local (cloud) resources
AF "seed" of resources
Mounted
Less often mounted
Xrootd/https
Distributed
Filesystem
CVM
REST
APIs

Small data

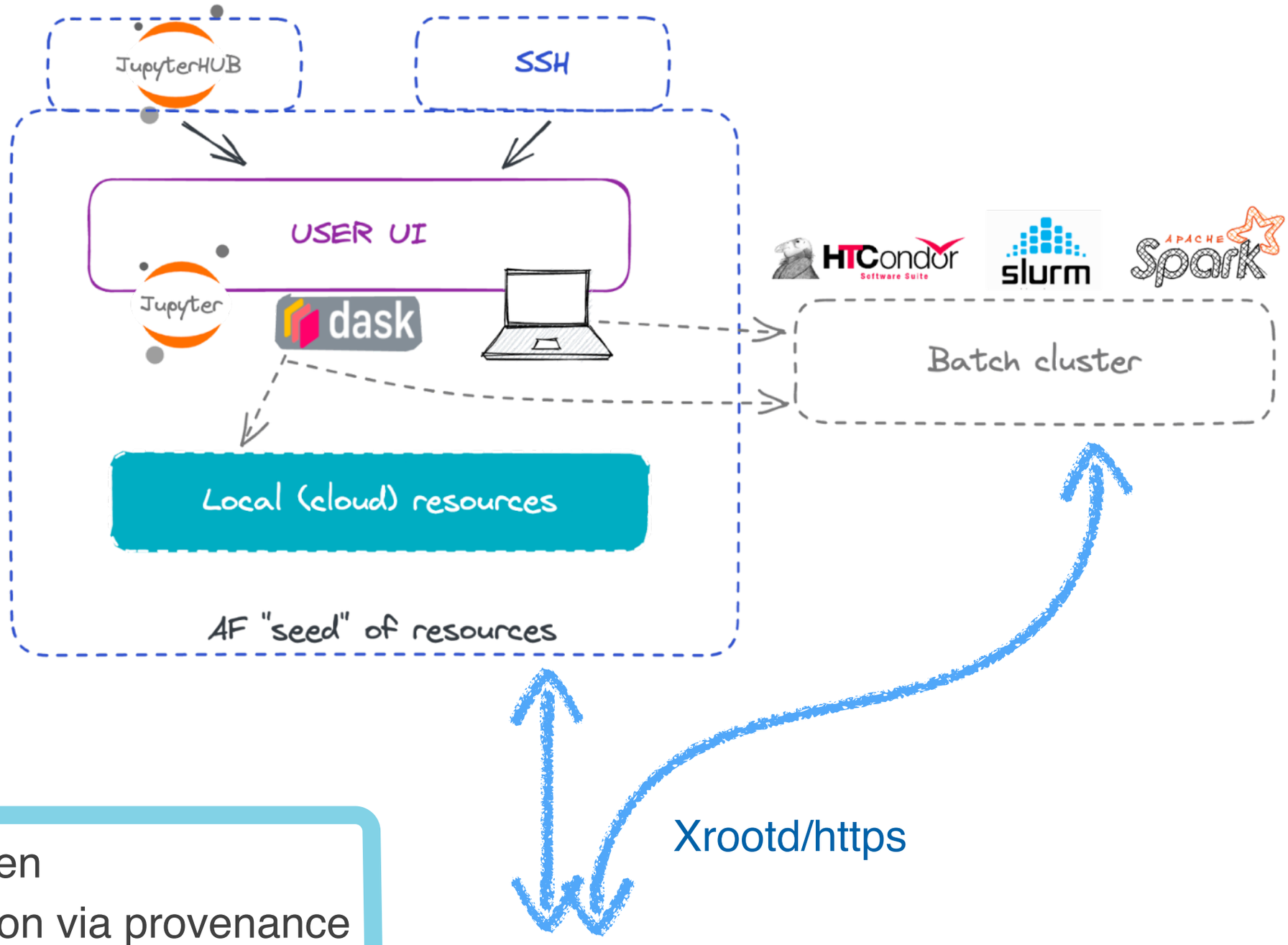


- Authorization: unix (token?)
- Logical organization via directories
  - Directory quotas
- Performance: IOPS
  - Read-only mount on workers?

Medium data


JupyterHUB
SSH
USER UI
Jupyter
dask
HTCondor
Software Suite
slurm
APACHE
Spark
Batch cluster
Local (cloud) resources
AF "seed" of resources
Xrootd/https
Distributed
Filesystem

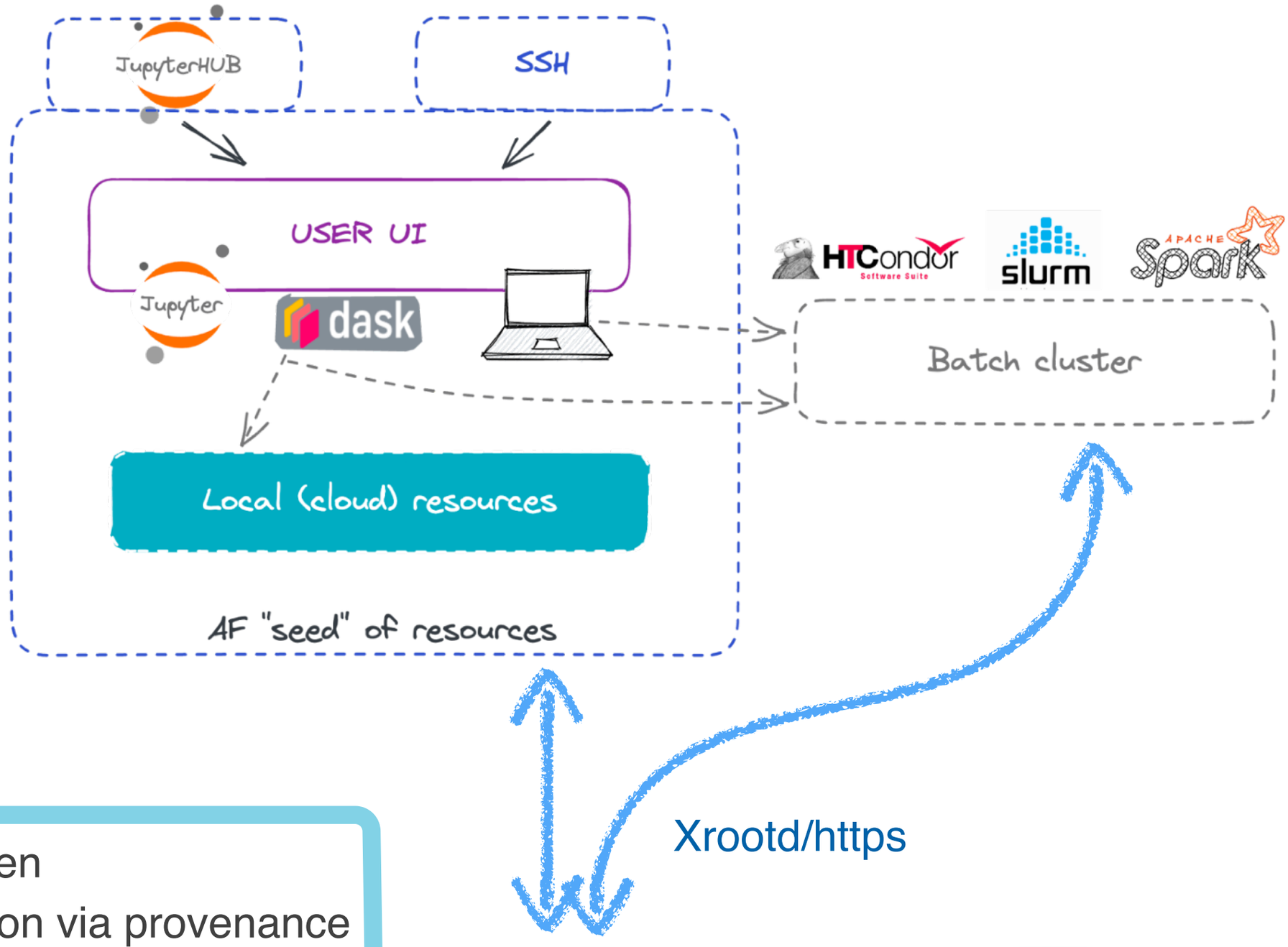
Medium data



- Authorization: token
- Logical organization via provenance
  - Derived dataset catalog?
- Performance: IOPS & Bandwidth
- Lateral movement is non-trivial
  - TPC across facilities?

Fermilab

Medium data



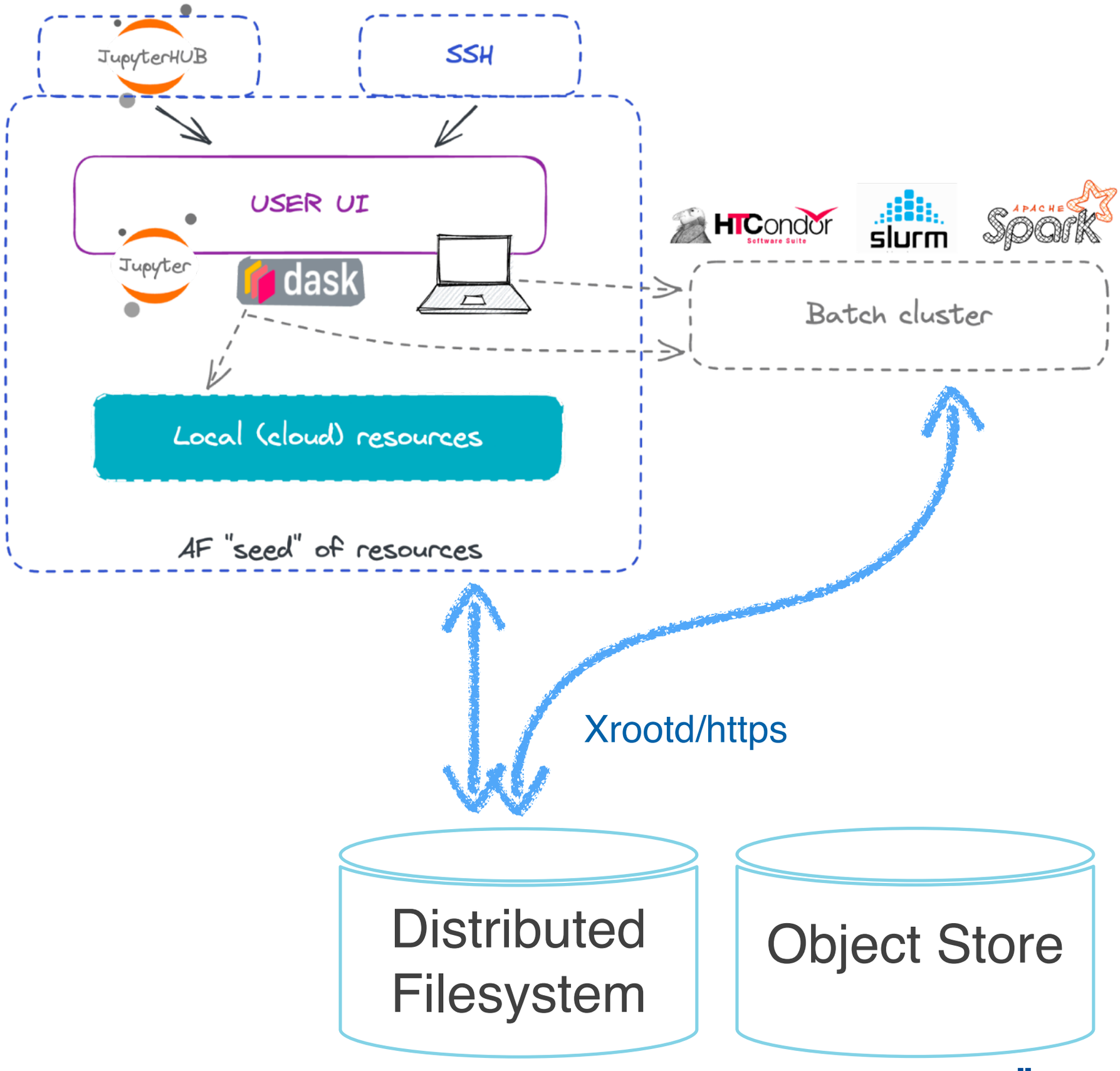
- Authorization: token
- Logical organization via provenance
  - Derived dataset catalog?
- Performance: IOPS & Bandwidth
- Lateral movement is non-trivial
  - TPC across facilities?

Distributed Filesystem

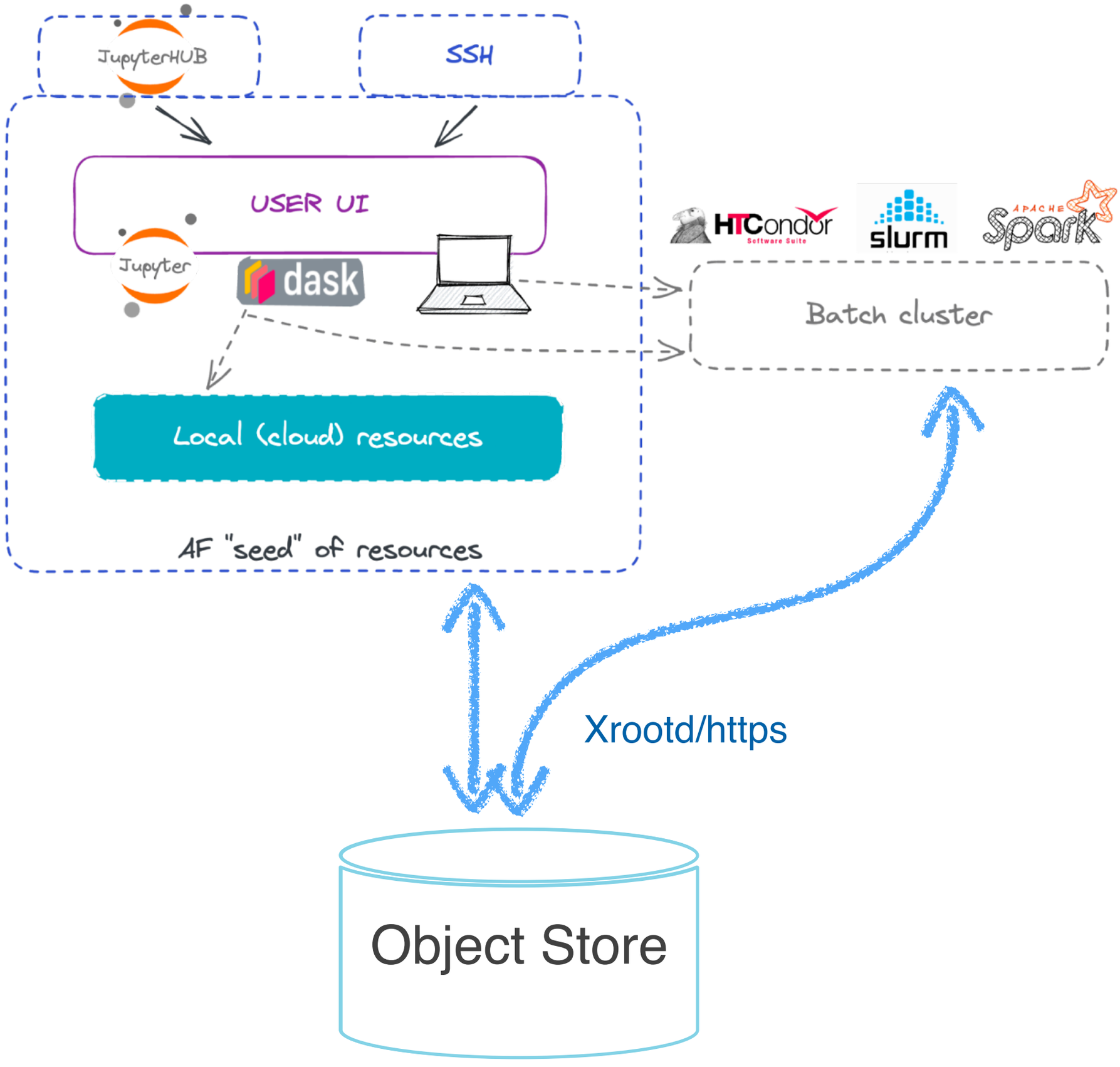
Object Store

Fermilab

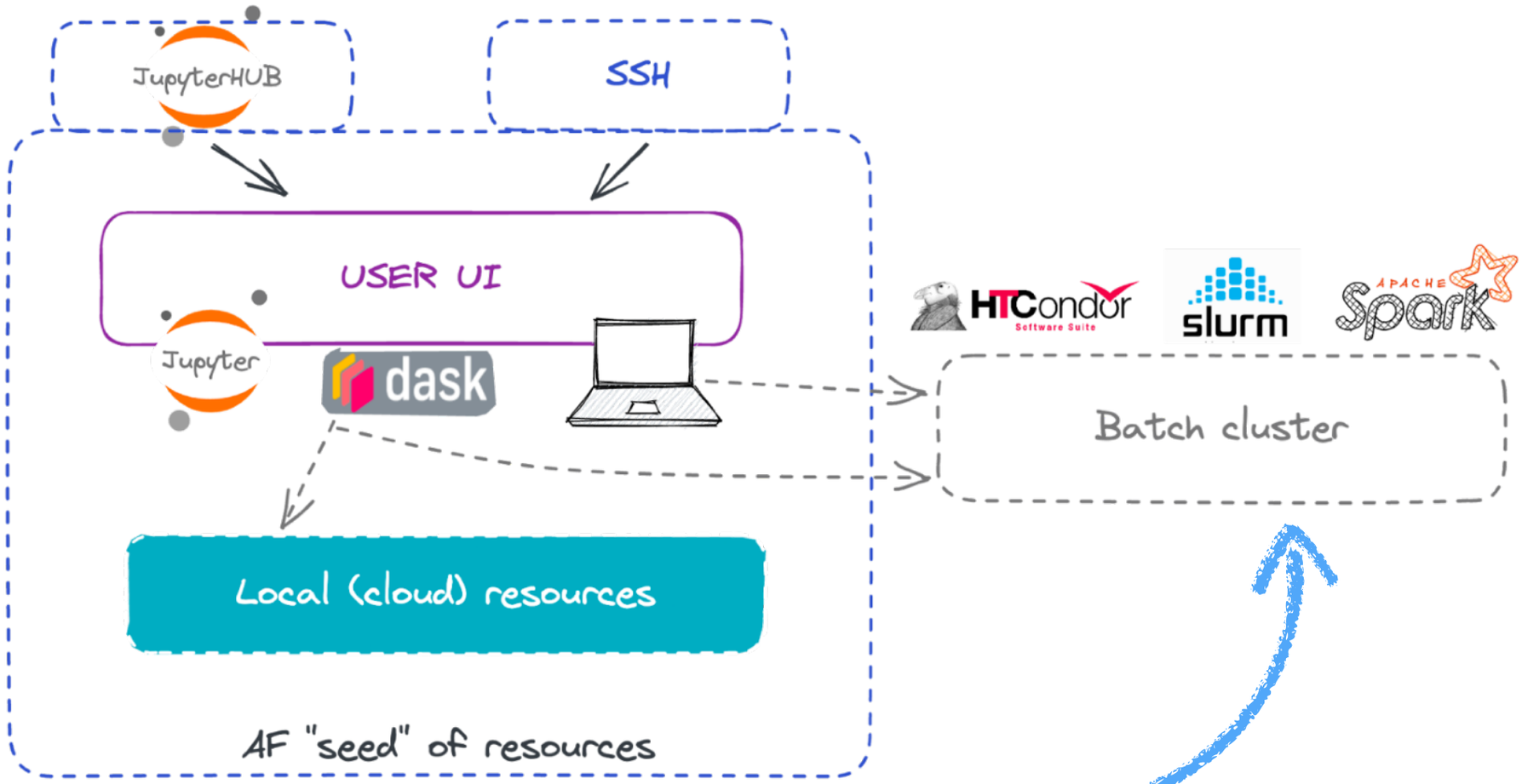
Medium data


JupyterHUB
SSH
USER UI
Jupyter
dask
HTCondor
Software Suite
slurm
APACHE
Spark
Batch cluster
Local (cloud) resources
AF "seed" of resources
Xrootd/https
Distributed
Filesystem
Object Store

Medium data

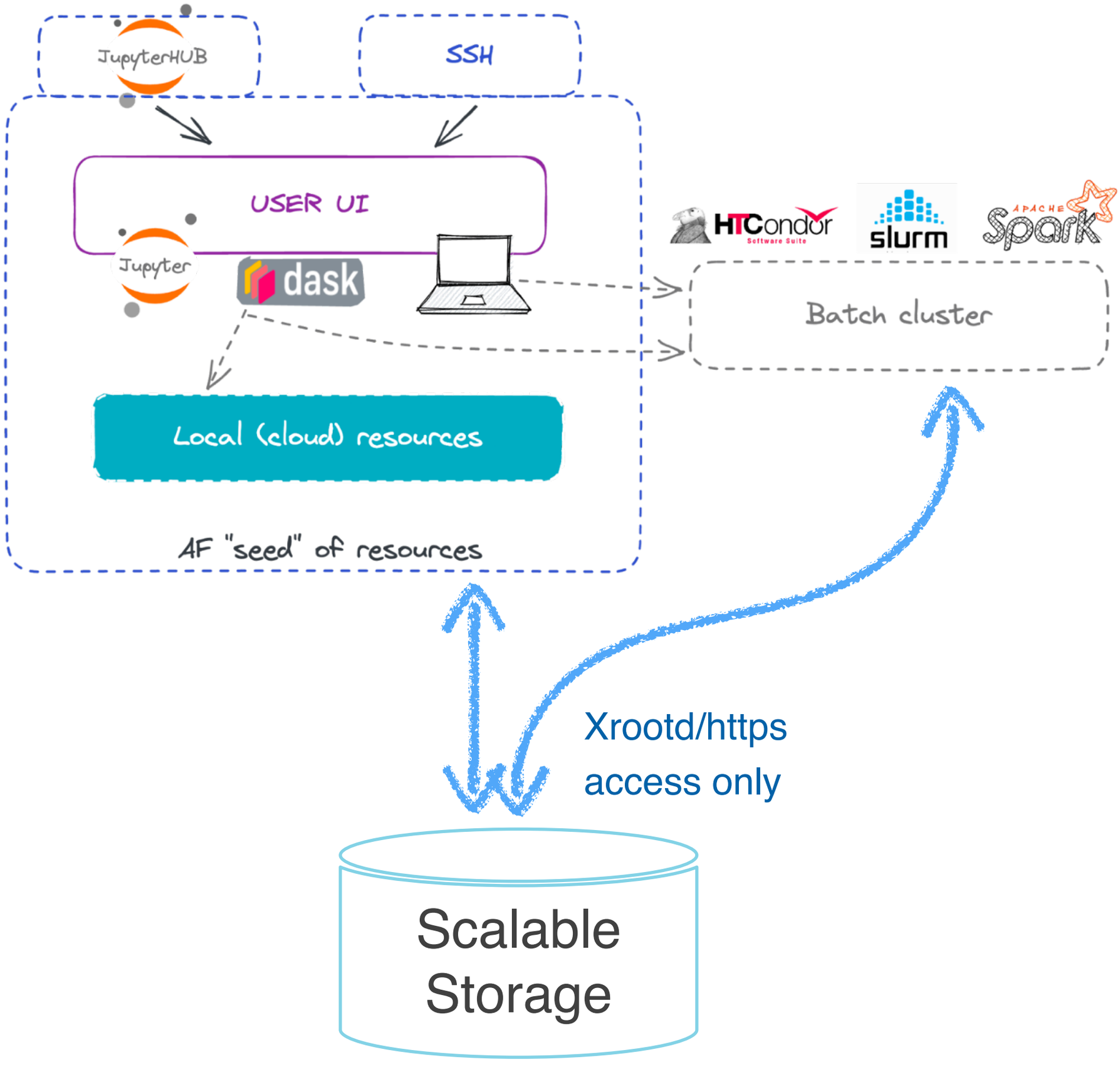

JupyterHUB
SSH
USER UI
Jupyter
dask
HTCondor
Software Suite
slurm
APACHE
Spark
Batch cluster
Local (cloud) resources
AF "seed" of resources
Xrootd/https
Object Store

Medium data

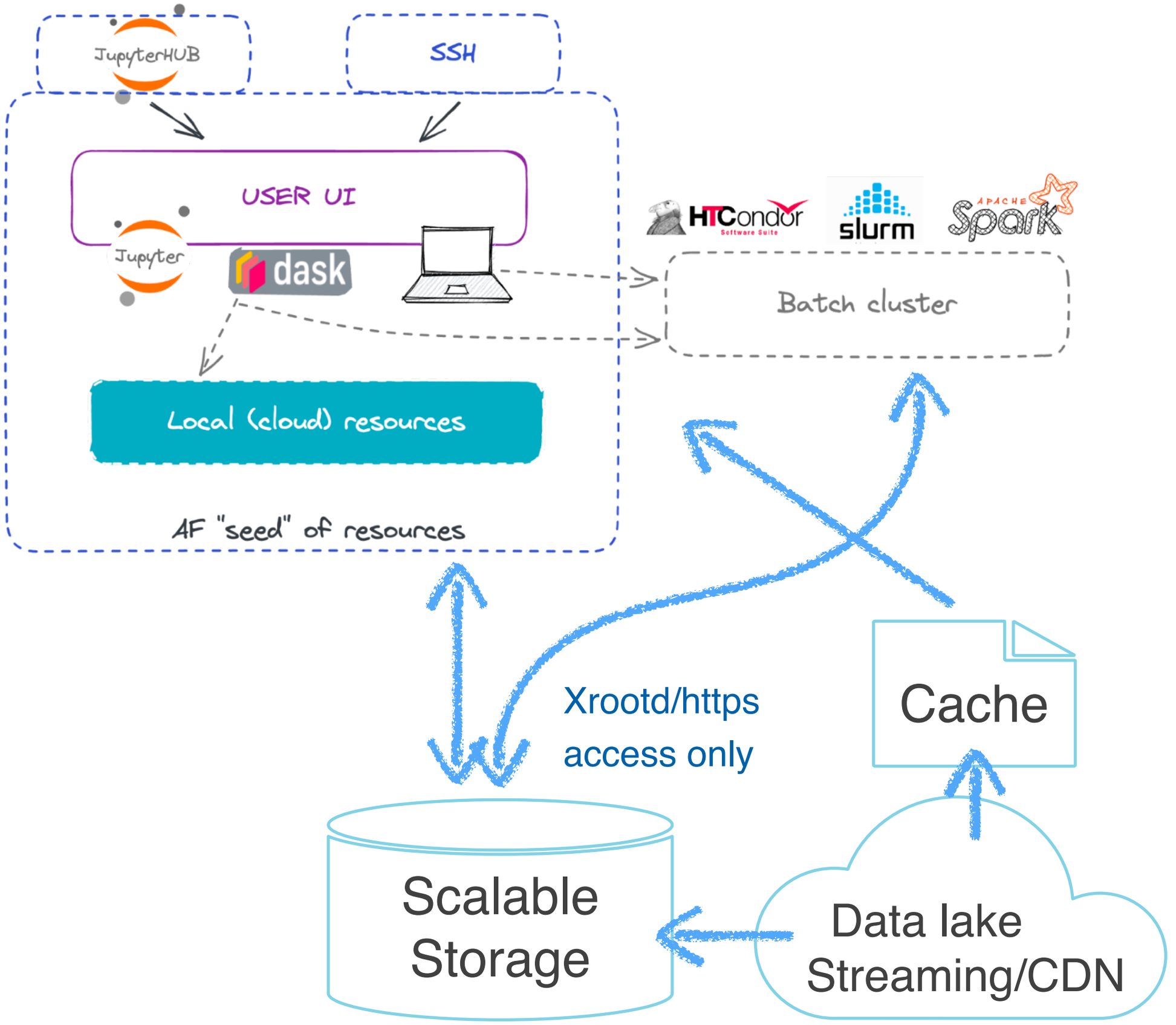


- A few pros:
  - Cloud-friendly: support industry query platforms
  - More flexible authorization & QoS
- A few cons:
  - Fighting 50y of unix knowledge
  - Existing infrastructure built on top of POSIX(-ish) base layer

Large data

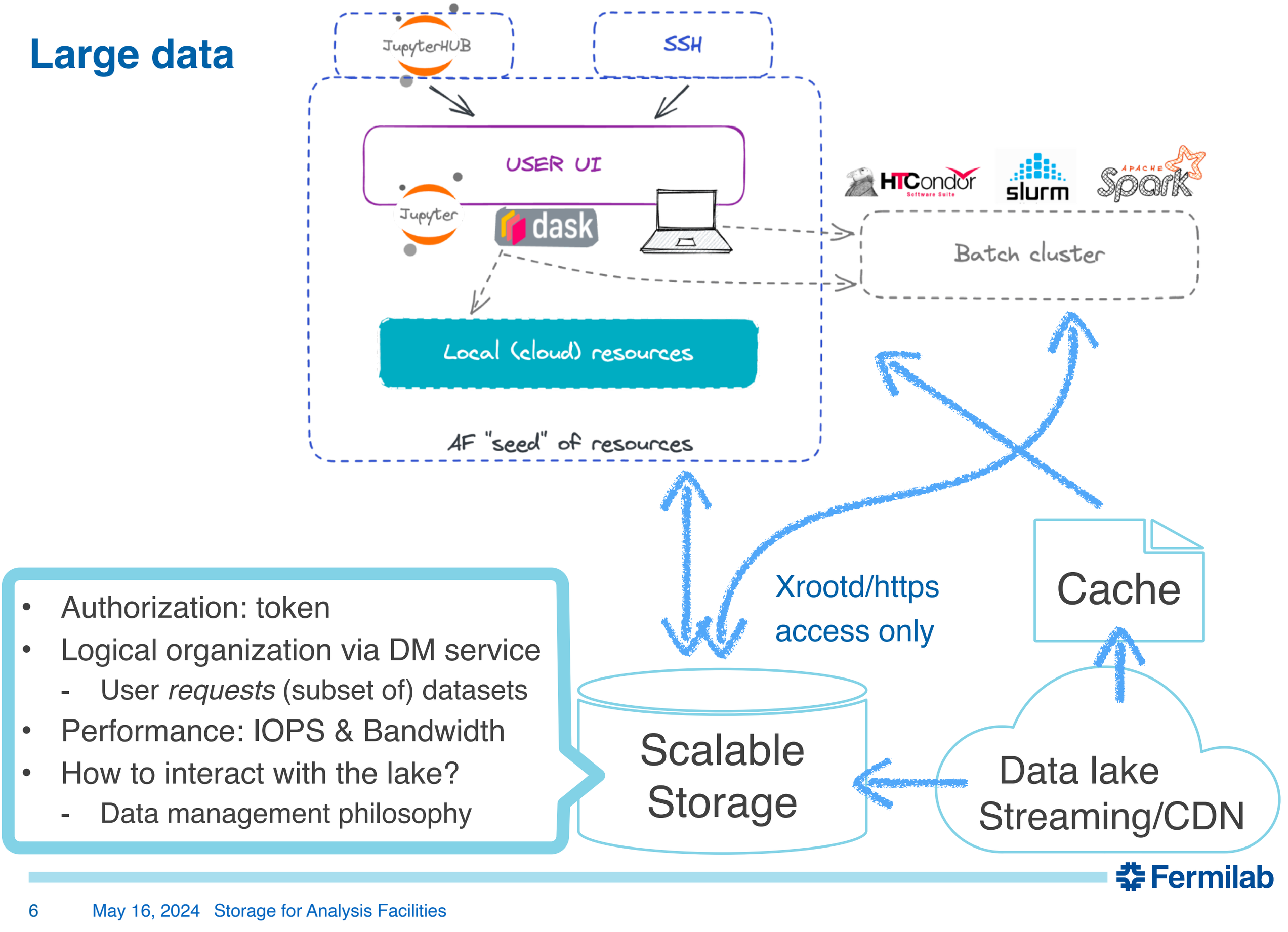

JupyterHUB
SSH
USER UI
Jupyter
dask
HTCondor
Software Suite
slurm
APACHE
Spark
Batch cluster
Local (cloud) resources
AF "seed" of resources
Xrootd/https
access only
Scalable
Storage

Large data



Fermilab

Large data

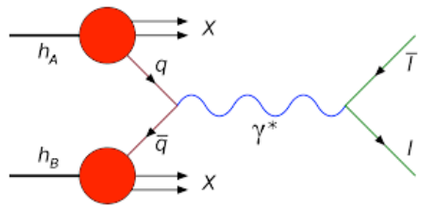


- Authorization: token
- Logical organization via DM service
  - User *requests* (subset of) datasets
- Performance: IOPS & Bandwidth
- How to interact with the lake?
  - Data management philosophy

Data management philosophy

Primary dataset

Abstract, “what kind of events.”
e.g. hard scatter process for simulation, trigger filter for data

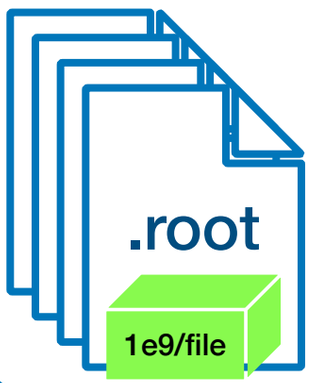


Data tiers

AOD

1e5/event

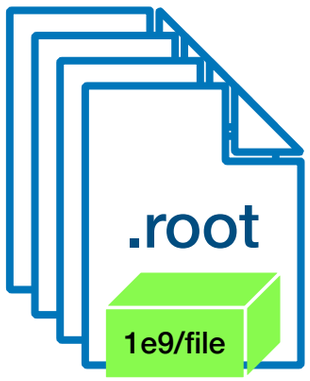
Data columns pertaining to low-level reconstruction



MiniAOD

1e4/event

Calibrated physics objects
Particle-flow candidates



. . .

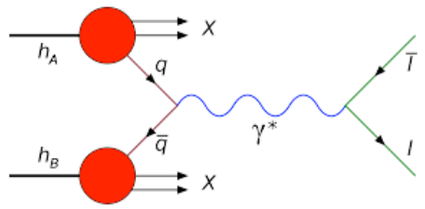
Similar layout for xAOD / PHYSLITE

Data volume
order of magnitude
[bytes]

Data management philosophy

Primary dataset

Abstract, “what kind of events.”
e.g. hard scatter process for simulation, trigger filter for data

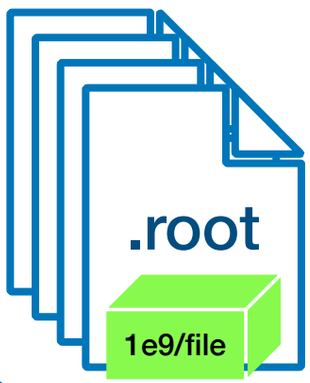


Data tiers

AOD

1e5/event

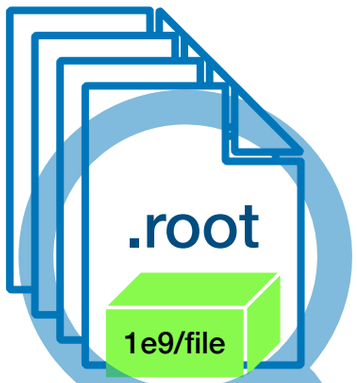
Data columns pertaining to low-level reconstruction



MiniAOD

1e4/event

Calibrated physics objects
Particle-flow candidates



. . .

Similar layout for xAOD / PHYSLITE

Data volume
order of magnitude
[bytes]

Data management ph

Primary

Abstract, “what kind
e.g. hard scatter proc

Data t iers

AOD

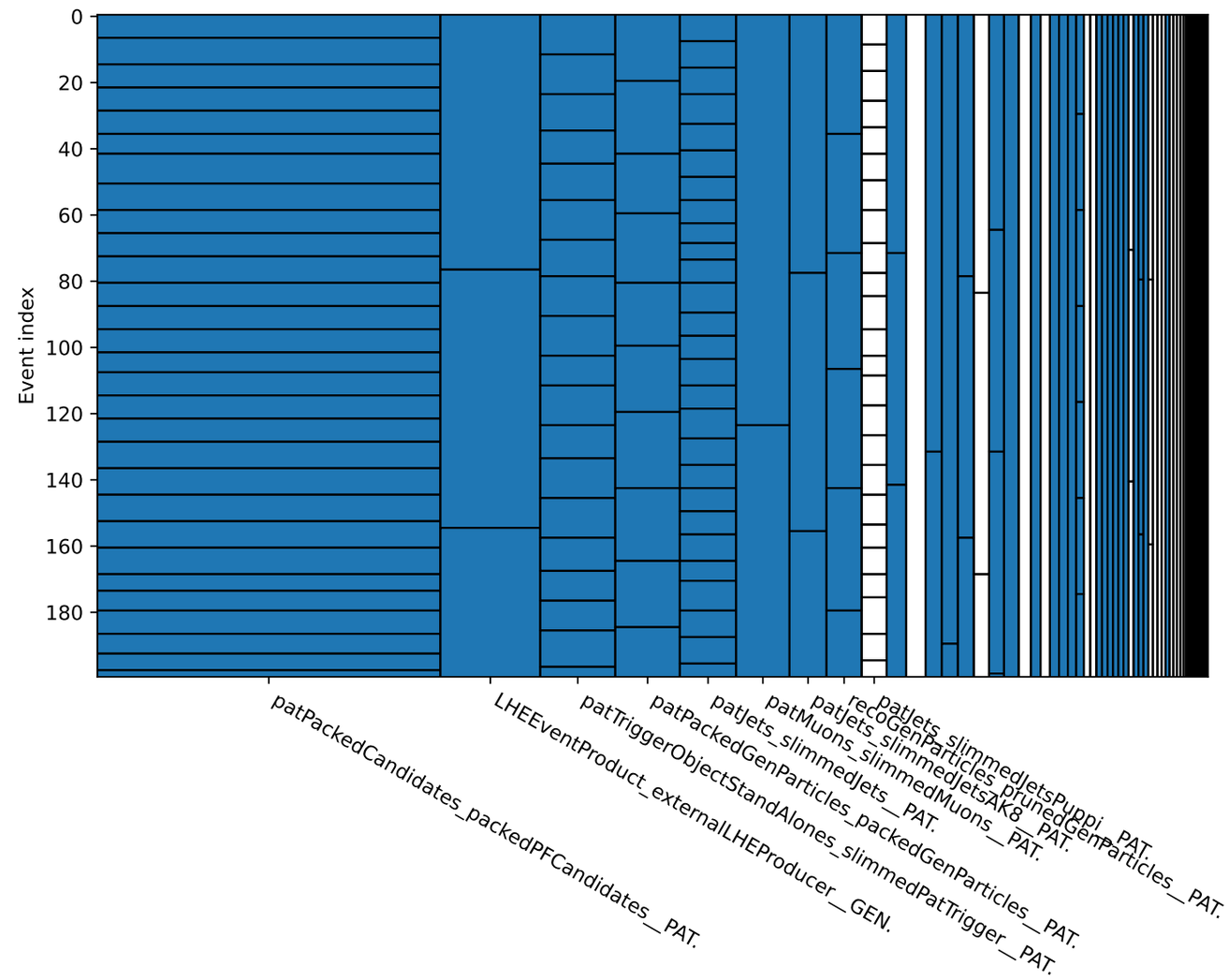
Data columns pe
low-level reconst

.root

1e9/file

Accessed

Not accessed



lume
gnitude
s]

Similar layout for xAOD / PHYSLITE

Fermilab

Data management ph

Primary

Abstract, “what kind
e.g. hard scatter proc

Data t iers

AOD

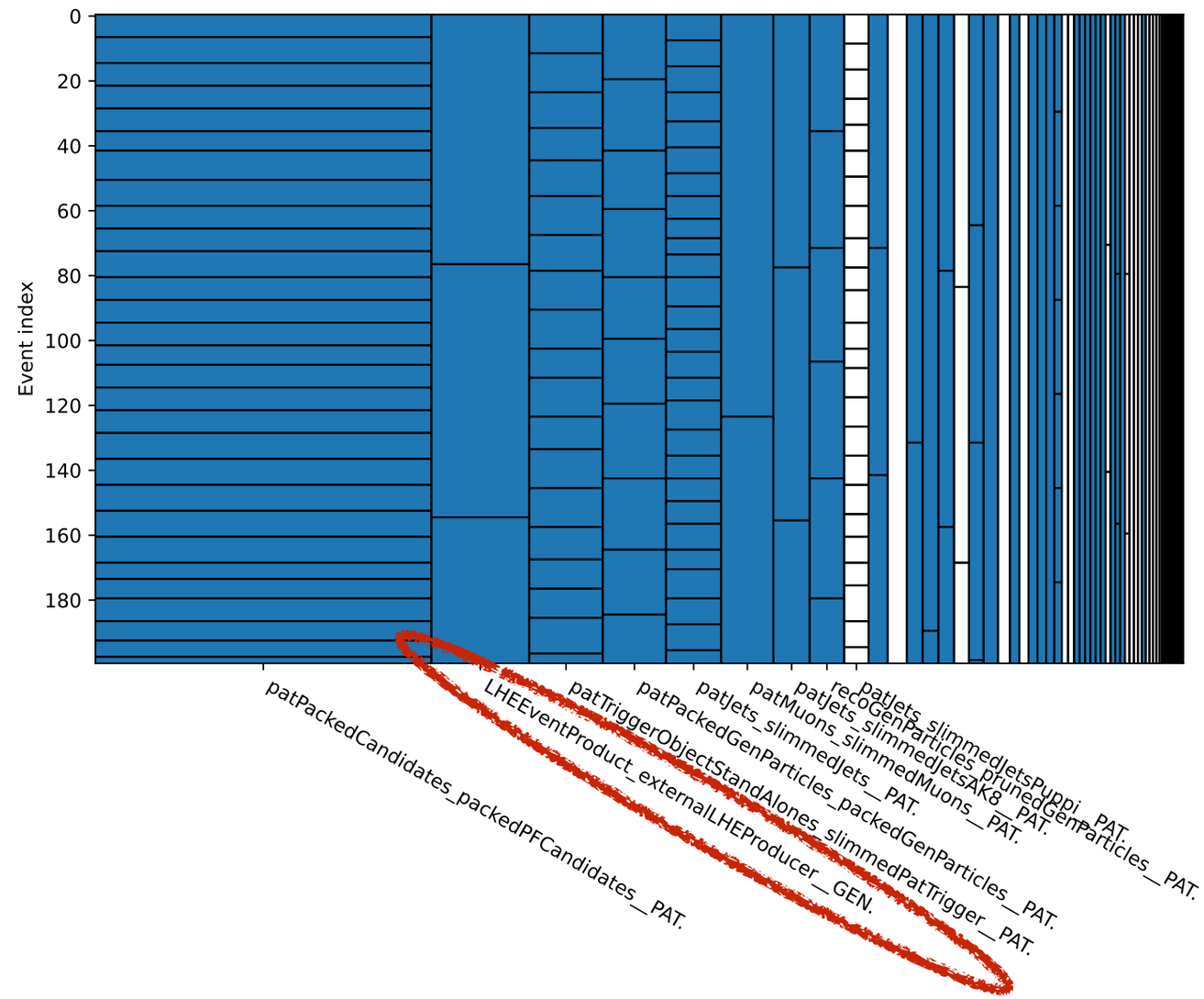
Data columns pe
low-level reconst

.root

1e9/file

Accessed

Not accessed



Copied forward each data tier

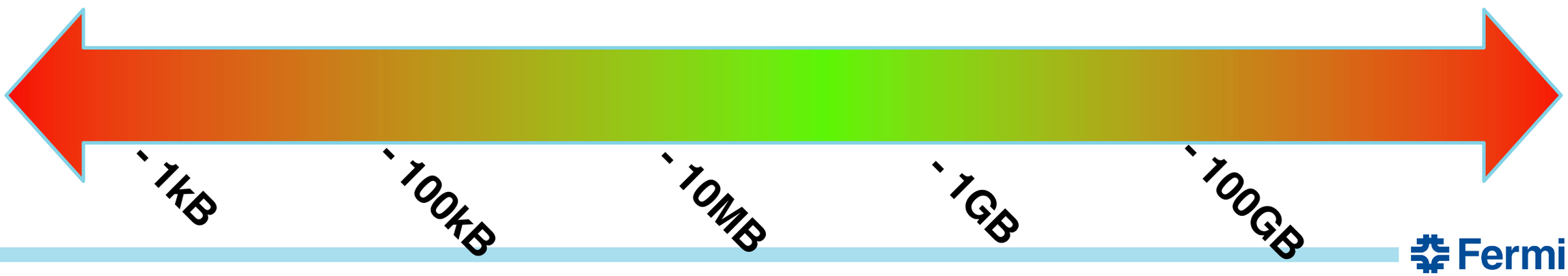
Similar layout for xAOD / PHYSLITE

lume
gnitude
s]

Fermilab

Can we align unit of data access to unit of physics content?

- Dataset = list of 2-4 GB files, totaling 10GB-1PB. Why?

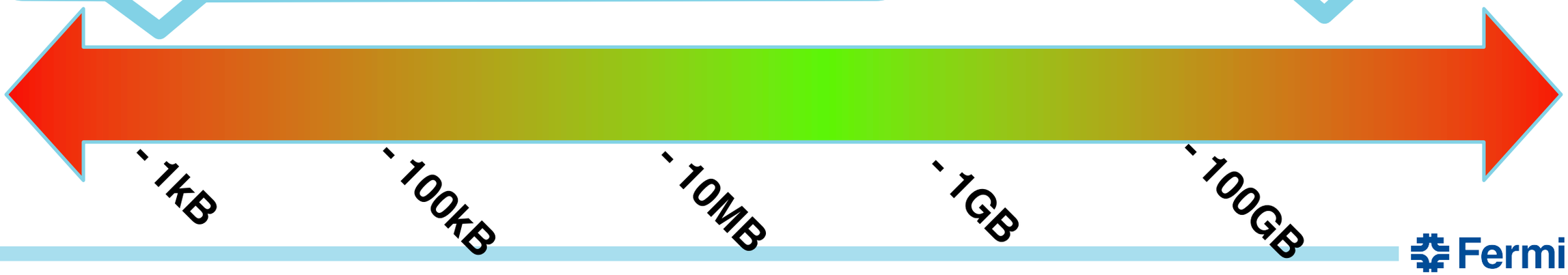


Can we align unit of data access to unit of physics content?

- Dataset = list of 2-4 GB files, totaling 10GB-1PB. Why?

- Lower limits:
  - IOPS!!
  - Erasure-code block size ~ 4-16kB
  - Catalog/filesystem overhead ~ 10MB-1GB?

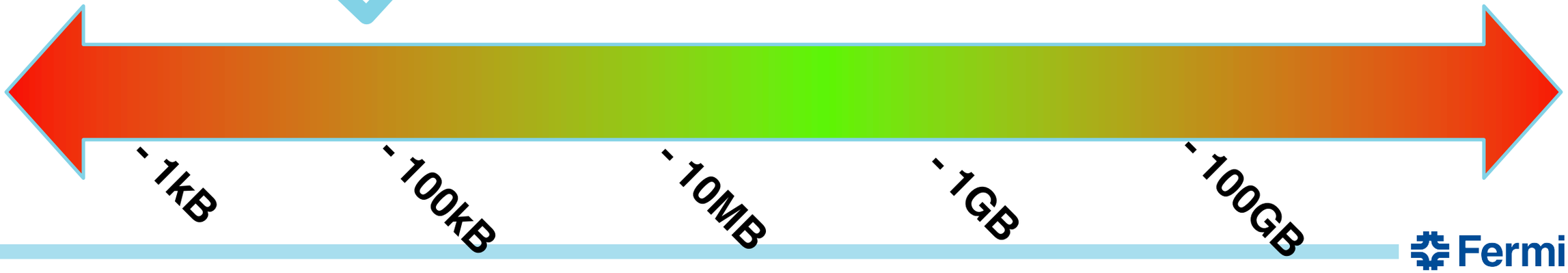
- Upper limits:
  - Third party copy timeout ~ 20 GB
  - Tape cartridge ~ 10 TB



Can we align unit of data access to unit of physics content?

- Dataset = list of 2-4 GB files, totaling 10GB-1PB. Why?

- TBasket / Page sizes ~ 10-100kB
  - This is physics-relevant
  - One float column for O(100k) events
  - One ragged column for O(10k) events
- Motivation for byte-range xcache

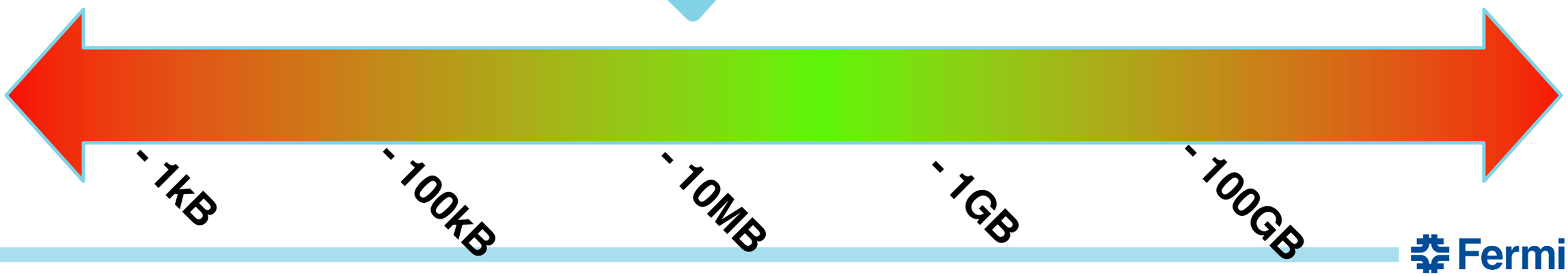


Fermilab

Can we align unit of data access to unit of physics content?

- Dataset = list of 2-4 GB files, totaling 10GB-1PB. Why?

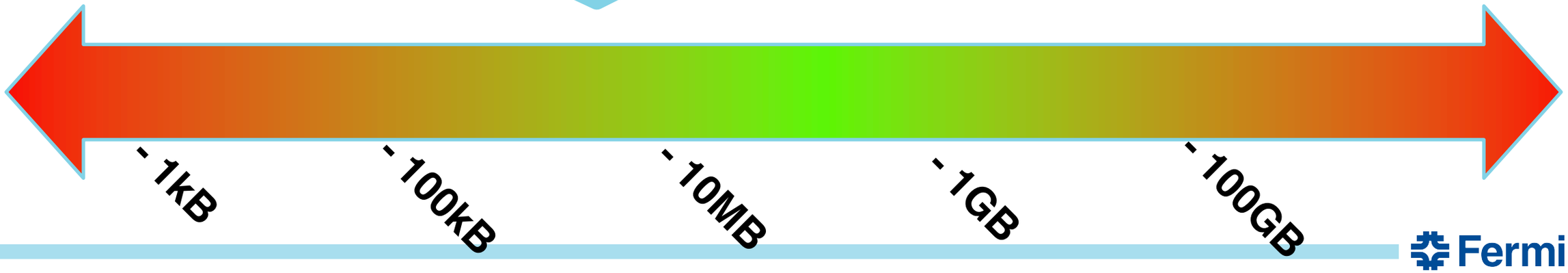
- Cluster size ~ 10MB
  - All columns pertaining to same group of events
- Good target for read-ahead buffer size
- Do we want to cluster *all* columns though?
  - Typical analysis accesses 10-50%
  - How will column joins be performed?



Can we align unit of data access to unit of physics content?

- Dataset = list of 2-4 GB files, totaling 10GB-1PB. Why?

- Sweet spot for access ~ 1MB
  - Few ragged columns for O(10k) events?
  - Many columns for O(1k) events?
  - Do we want small # events per unit?
- Whole-unit cache ➜ off-shelf solutions
- Catalog challenge: need indirection



Fermilab

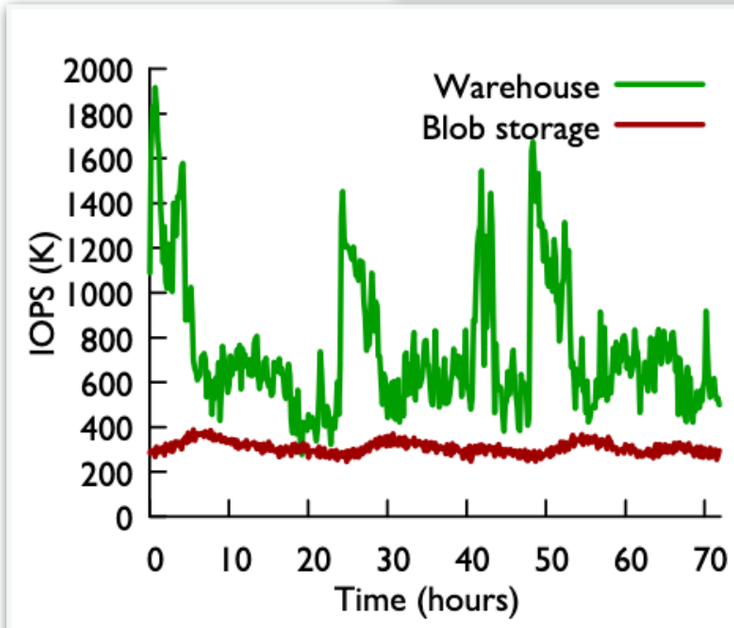
Aspirational

- Facebook Tectonic FS: one disk cluster per datacenter, two basic workloads:
  - Blob storage: pictures/videos
    - Steady-state IOPS, random access
  - Warehouse: engagement data (clicks/likes)
    - Bursty, more sequential access
- Potential analog:
  - Blob storage: pileup mixing in generation
  - Warehouse: analysis queries
- Many spindles!
  - Load-balance➜performance
  - Scalability via indirection
    - 3 (!) metadata queries /access

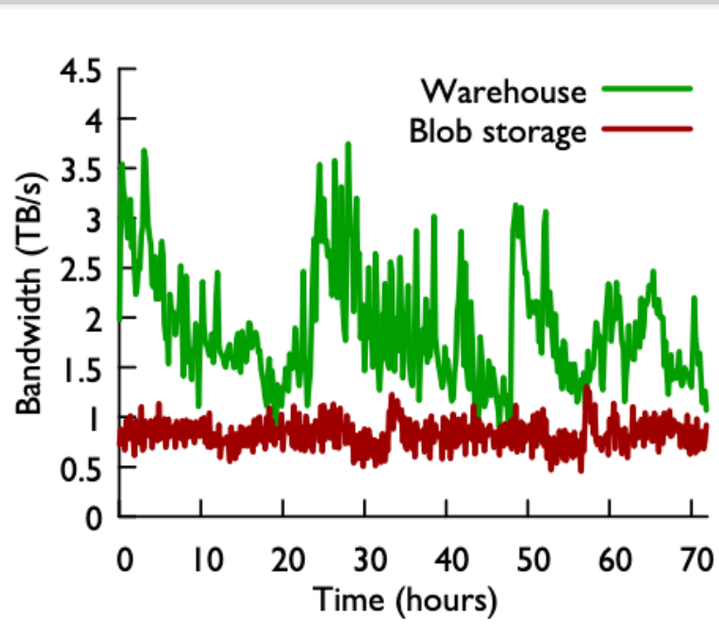
- https://www.usenix.org/system/files/fast21-pan.pdf

Capacity	Used bytes	Files	Blocks	Storage Nodes
1590 PB	1250 PB	10.7 B	15 B	4208

Table 2: Statistics from a multitenant Tectonic production cluster. File and block counts are in billions.



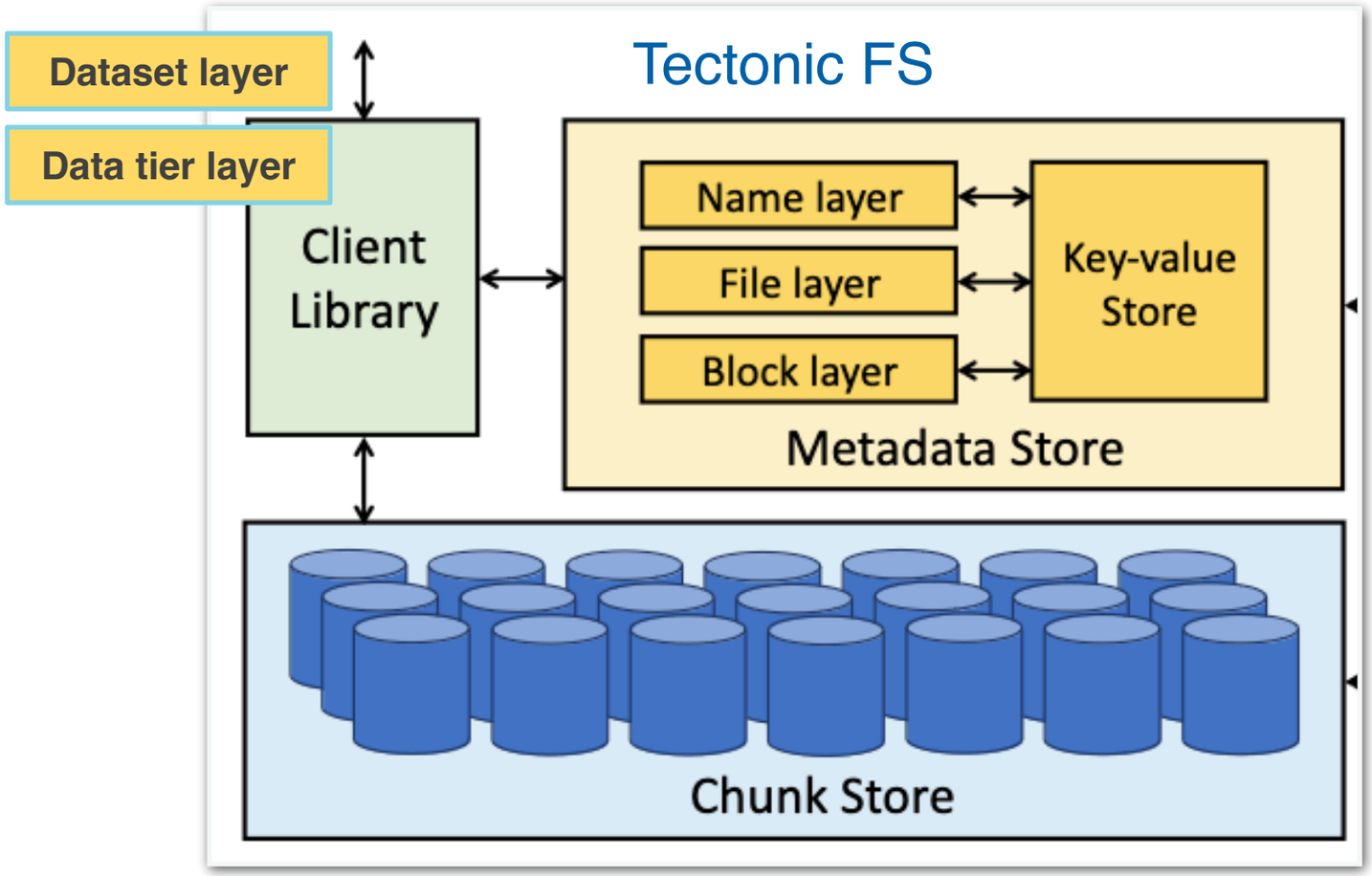
(a) Aggregate cluster IOPS



(b) Aggregate cluster bandwidth

Fermilab

Higher levels of indirection



For intermediate data, (ab)using POSIX filesystem as an implicit data catalog.

Bring Rucio to intermediate data? Does Rucio have sufficient indirection layers?

How do we enable a “facility grid” (cross-facility namespace for intermediate data)

Conclude with AF Whitepaper open questions

On data "locality" at facilities
Should analysts expect the data, particularly reduced formats, to be local to any facility they wish to use (thus providing low latency access)? Often analysts work with derived datasets (with extra cuts, derived variables), does the same apply?

On POSIX file access?
Is POSIX required? Maybe just interacting with an object store via e..g xrootd / https / analysis software is sufficient? How much work is required to fully support object stores? Users like filesystem-like semantics, but what part of POSIX is really needed and can we decouple mass storage access from more interactive, smaller scale activities?

On user interaction with Distributed Data Management systems
Can all Distributed Data Management queries be hidden from the user and is this desirable? Should users expect that this is managed for them?

On provenance of intermediate data products
Which intermediate files need to be promoted from local to global storage so that users can run at different sites and how is this declared by the user?

On common file sharing services
Do we need a common file sharing service to help users share their files? In which case can any existing services fulfill this purpose?