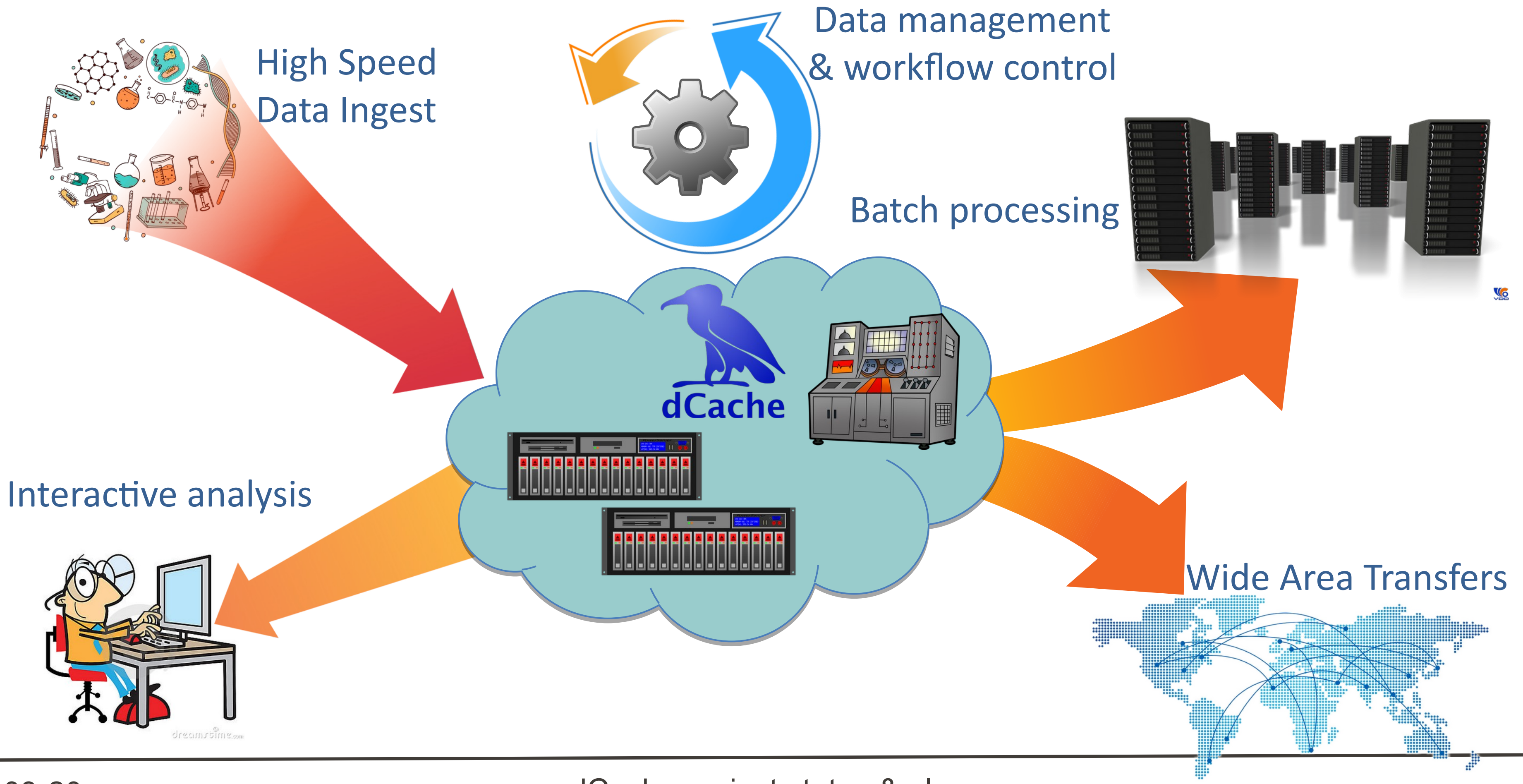
dCache Status and Plans

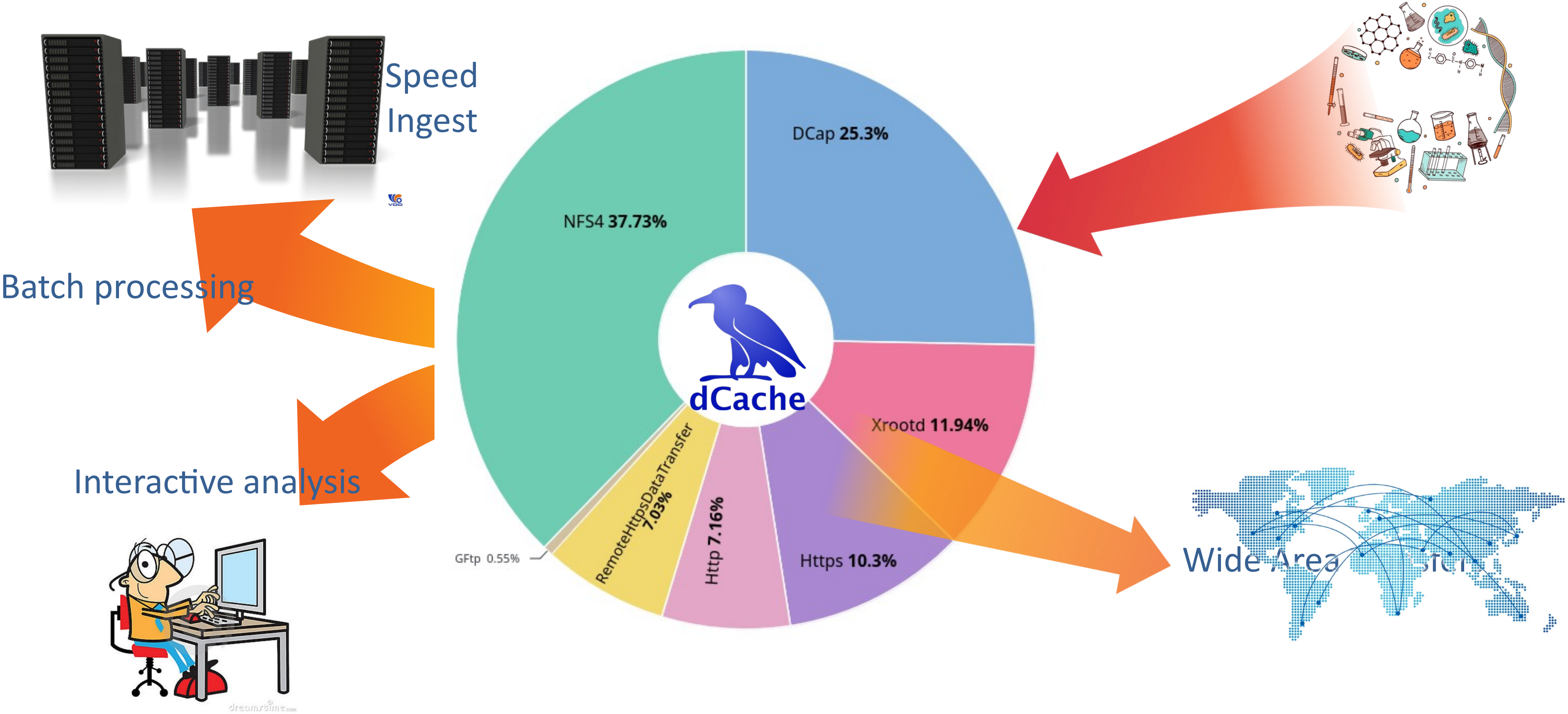
Workshop on ATLAS Computing and Software Activities at BNL - Navigating Distributed Computing, Storage, Compute, and Beyond

Tigran Mkrtchyan for the dCache collaboration

DESY. Fermilab neic NORDIC E-INFRASTRUCTURE COLLABORATION

HELMHOLTZ RESEARCH FOR GRAND CHALLENGES


Data management
& workflow control
High Speed
Data Ingest
Batch processing
dCache
Interactive analysis
Wide Area Transfers



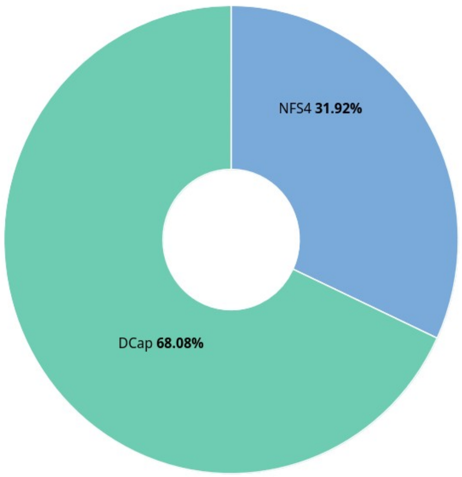
Speed Ingest

Batch processing

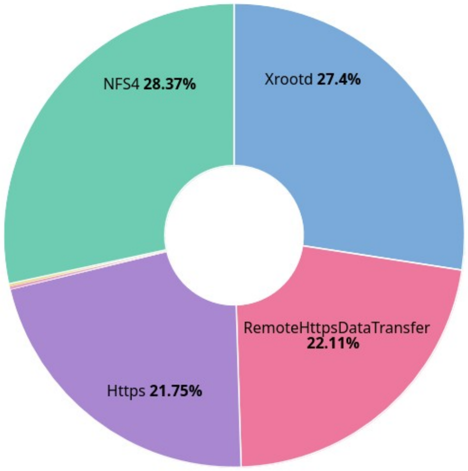
Interactive analysis

Wide Area Transfers

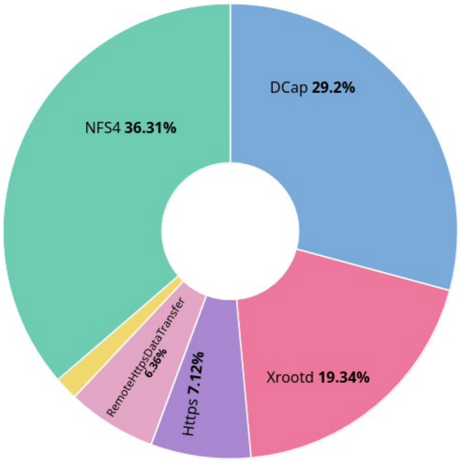
Protocols and Instances



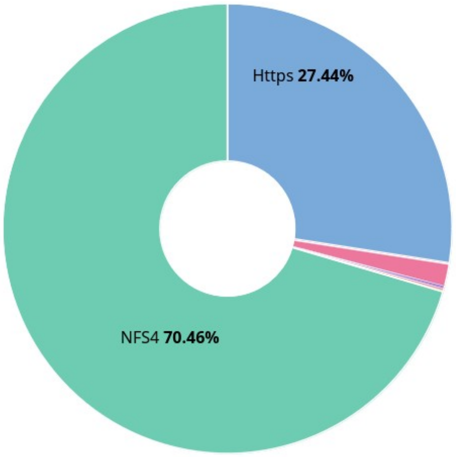
XFEL



ATLAS



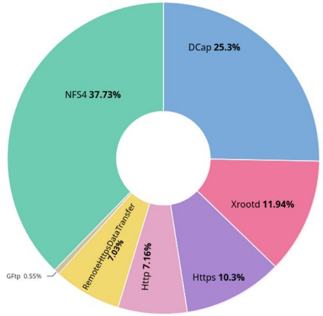
CMS

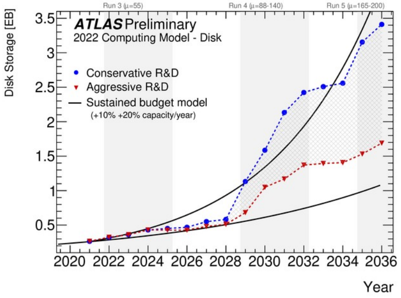


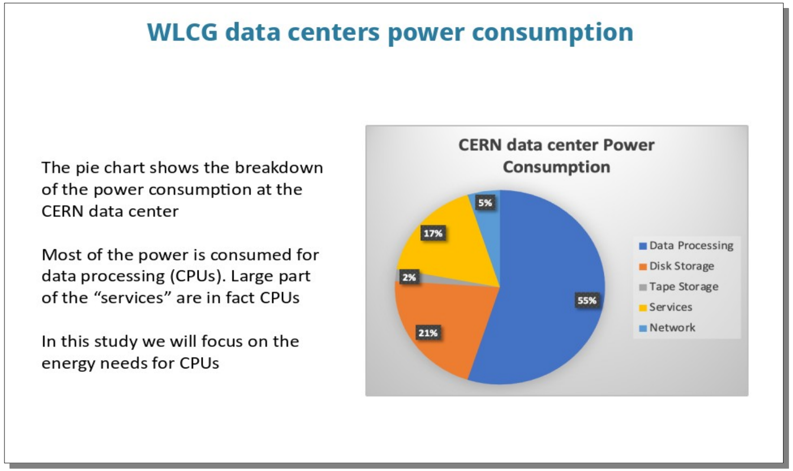
Belle-II

The Challenges

- Data is going to grow… A lot…
  - High ingest data rates
  - More movements between sites
- Shared Computing Resources
  - Analysis Facilities
  - Grid Farms
  - HPC
  - Cloud resources (CPU&Storage)
- Standard analysis tools
  - ROOT
  - Jupyter Notebooks, non-ROOT analysis
- Competing Tape Operations







WLCG data centers power consumption

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

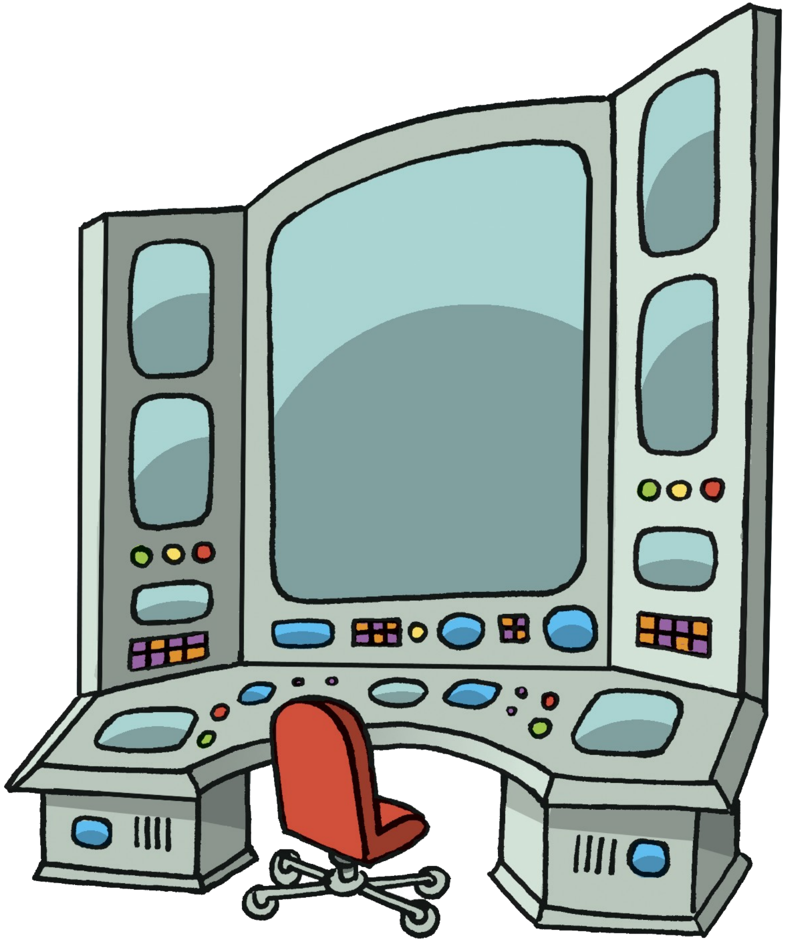
Most of the power is consumed for data processing (CPUs). Large part of the “services” are in fact CPUs

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

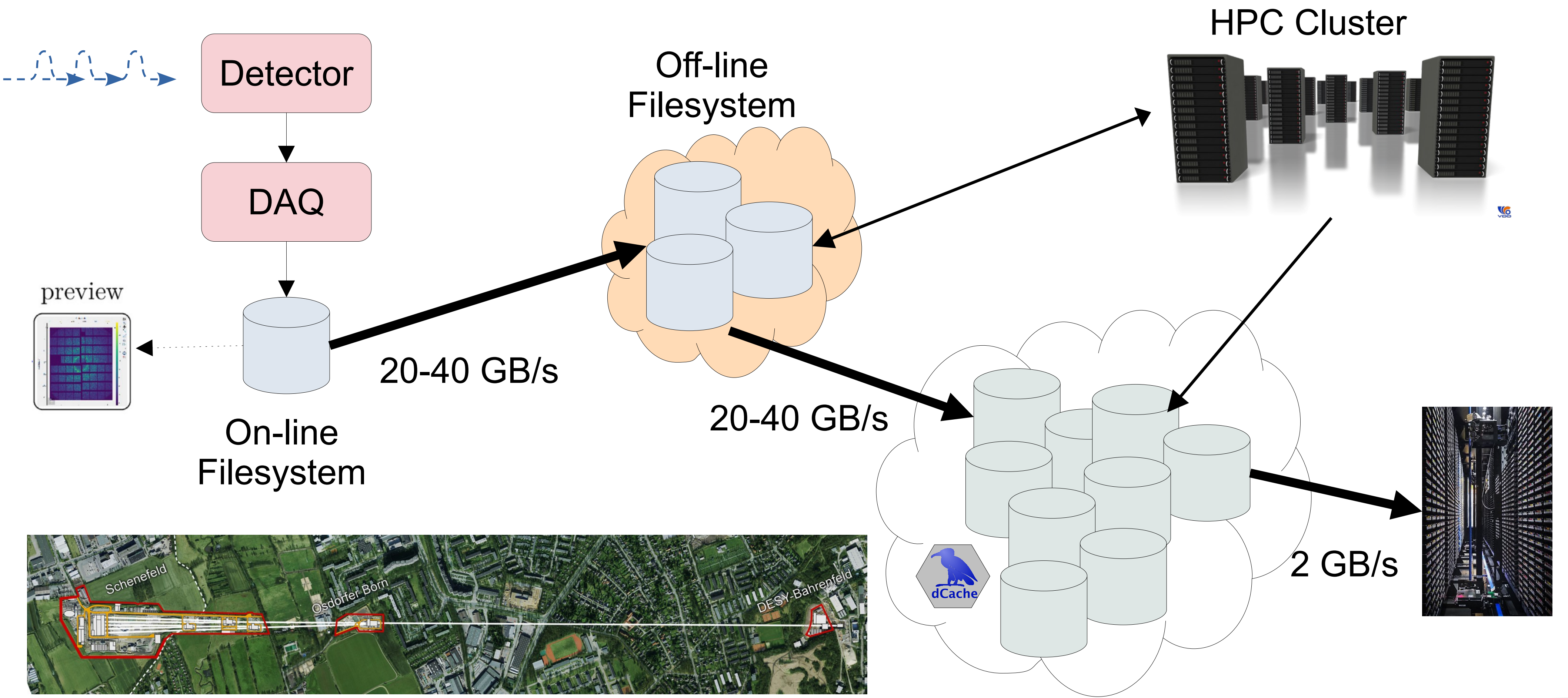
CERN data center Power Consumption
55%
21%
2%
17%
5%
Data Processing
Disk Storage
Tape Storage
Services
Network

Some (DESY) Numbers

- XFEL
  - Total capacity ~120 PB
  - ~400 physical hosts (~4000 dCache pools)
  - 20-40 GB/s ingest
- Photon
  - DB size – 2.5TB
  - ACL table 600GB
  - Directories with 3 10^6 files
  - 1.2 10^9 file system objects
  - 100K files in the flush queue
  - Two tape copies, different media type
- ATLAS
  - dir/file → 1/3
- NextCloud
  - File lifetime < 1s



XFEL Data Management



Detector

DAQ

preview

On-line Filesystem

20-40 GB/s

Off-line Filesystem

HPC Cluster

20-40 GB/s

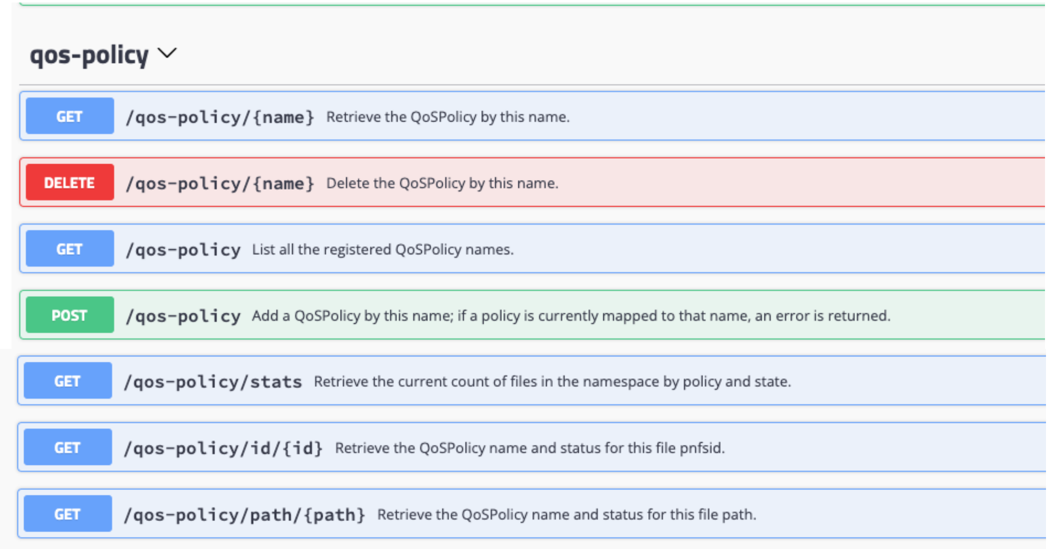
2 GB/s

QoS “Rule Engine”

- The policy contains a ordered list of QoS transitions (or media changes)
- Admins can associate a qos-policy with a file
  - New policy can be assigned to files on create
  - New “QosPolicy” directory tag
- The policy uploaded through front-end REST-API
- The policy is defied as a JSON document

QoS Policy (pseudo) Example:

```
"name": "my-policy",
"states": [
  {
    "duration": "P10D",
    "media": 2x DISK
  },
  {
    "duration": "P1M",
    "media": 1x DISK, 1x HSM
  },
  {
    "media": 2x HSM
  }
]
```



qos-policy

GET `/qos-policy/{name}` Retrieve the QoSPolicy by this name.

DELETE `/qos-policy/{name}` Delete the QoSPolicy by this name.

GET `/qos-policy` List all the registered QoSPolicy names.

POST `/qos-policy` Add a QoSPolicy by this name; if a policy is currently mapped to that name, an error is returned.

GET `/qos-policy/stats` Retrieve the current count of files in the namespace by policy and state.

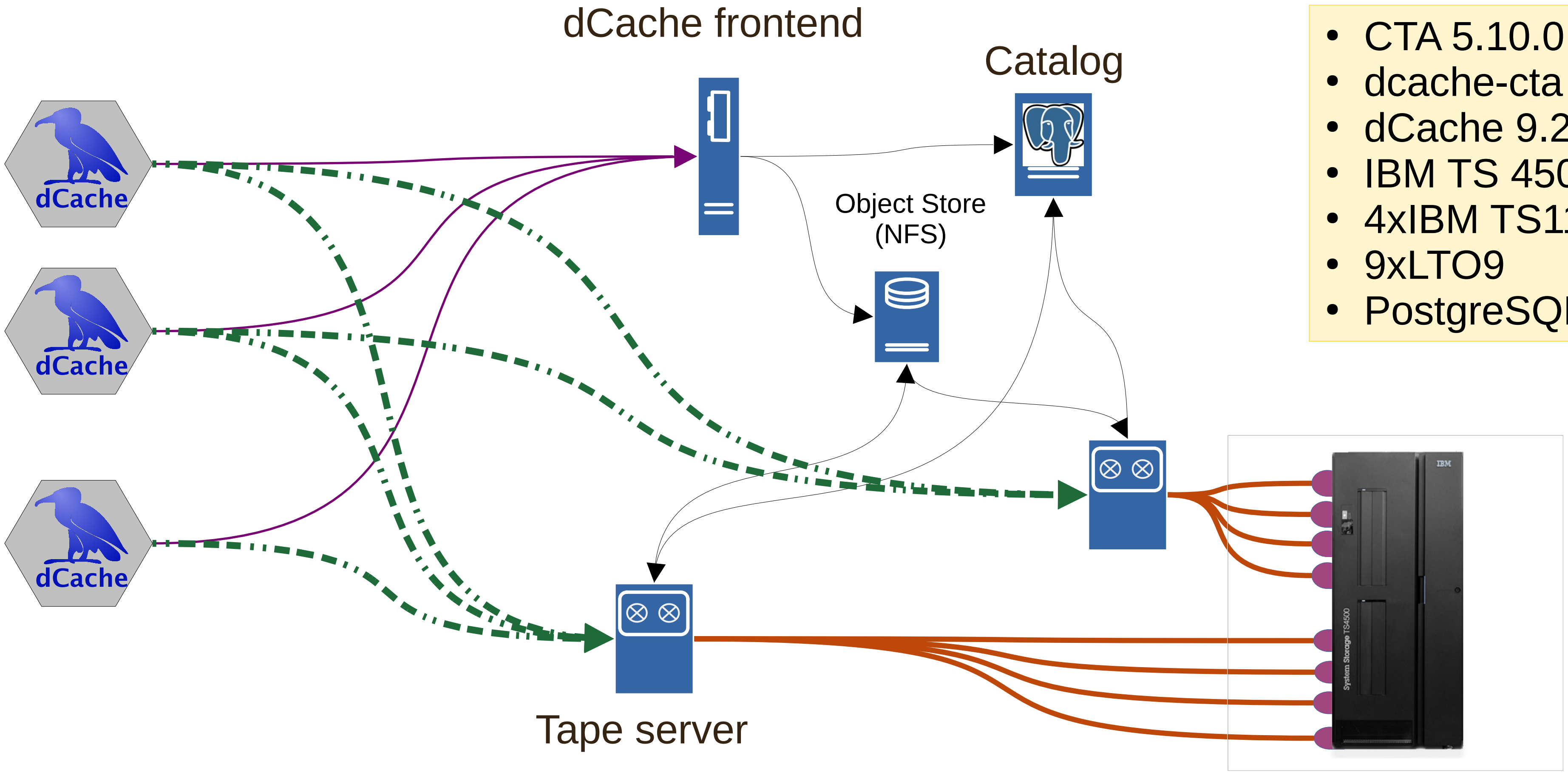
GET `/qos-policy/id/{id}` Retrieve the QoSPolicy name and status for this file pnfsid.

GET `/qos-policy/path/{path}` Retrieve the QoSPolicy name and status for this file path.

QoS Requirements

- HEP
  - Single copy (tape or disk)
- Photon Science
  - 2 tape copies, different media types (Jag+LTO)
- XFEL
  - 2 media copies (disk+tape ⇒ tape+tape)
- NextCloud
  - 2 disk copies + tape

Deployment at DESY

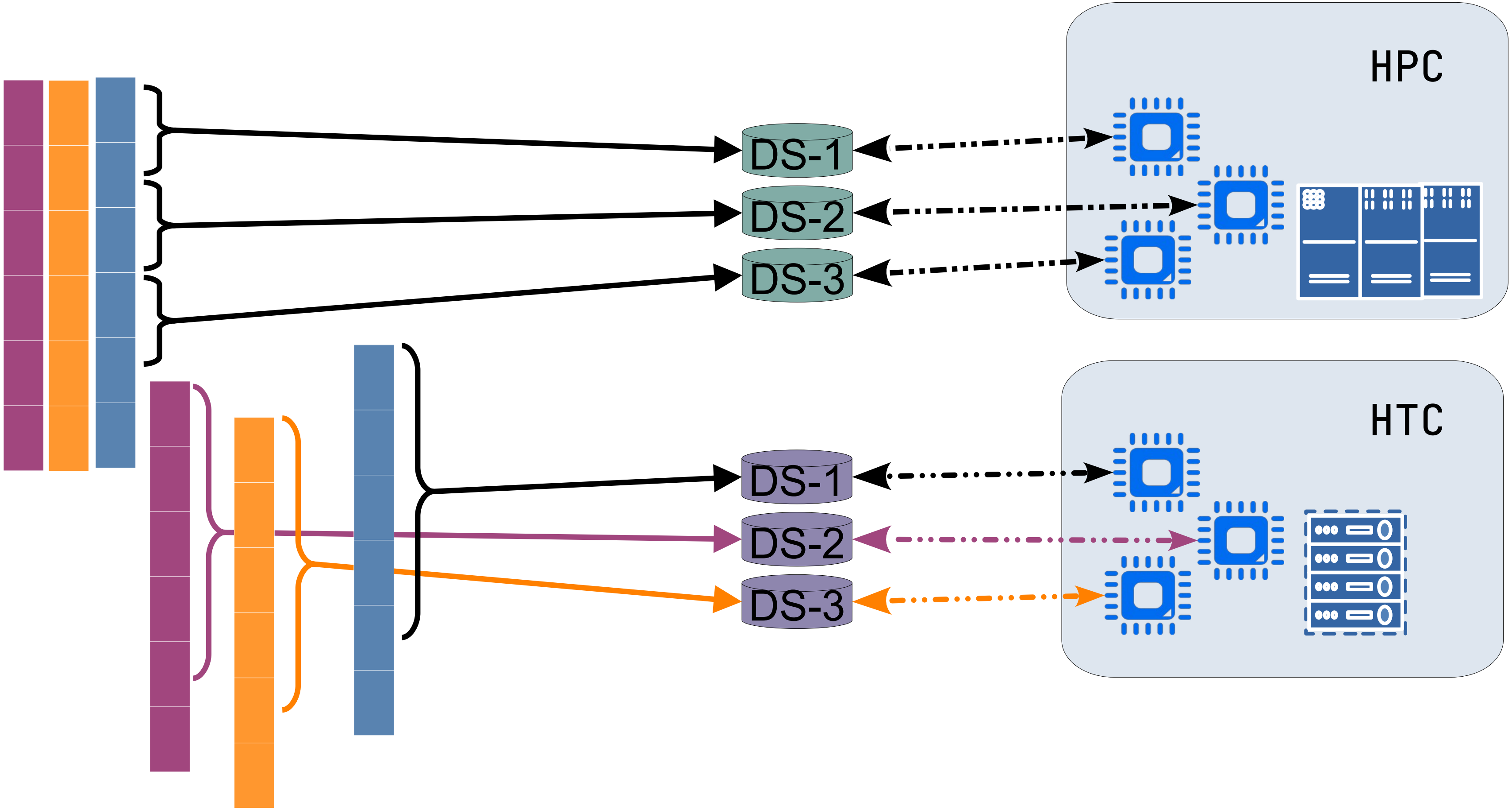


- CTA 5.10.0
- dcache-cta 0.11.0
- dCache 9.2.6
- IBM TS 4500
- 4xIBM TS1160
- 9xLTO9
- PostgreSQL 14.1

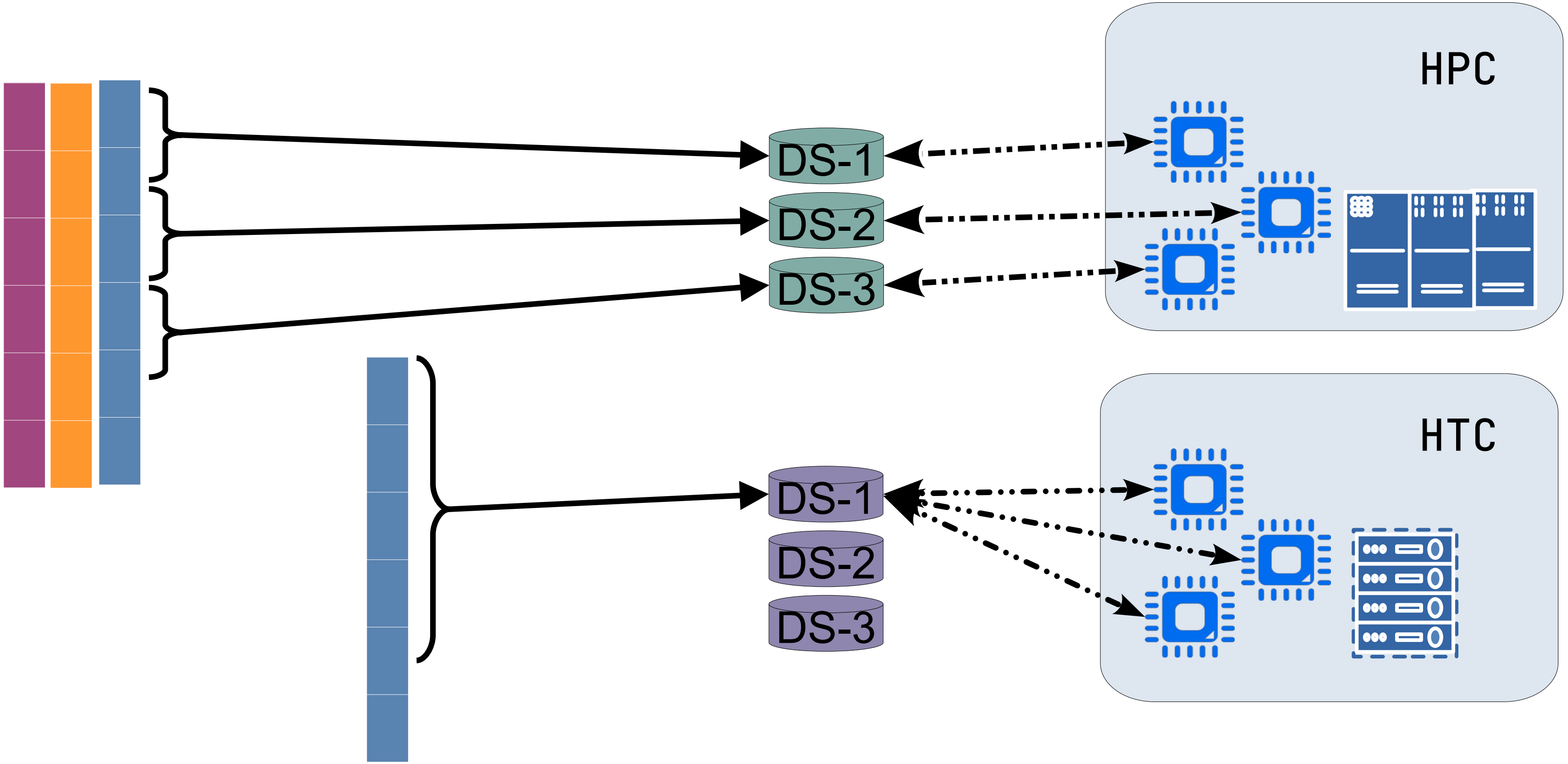
dCache+CTA Status

- Seamless integration with dCache is merged into upstream CTA code at CERN
  - Starting CTA release {4,5}.7.12
- The existing ENSTORE/OSM tape format is supported for READ
  - The ENSTORE/OSM tape catalog conversion procedures are successfully tested at DESY, Fermilab, PIC.
- dCache+CTA is deployed at DESY for all experiments
  - ~2PB/week (3.4 GB/s, 9 drives)
- dCache+CTA deployment replicate to by other HEP sites
  - Fermilab and PIC Barcelona have successfully replicated our setup (currently dCache + ENSTORE).
  - RAL in UK plans to migrate to PostgreSQL from ORACLE based on our experience

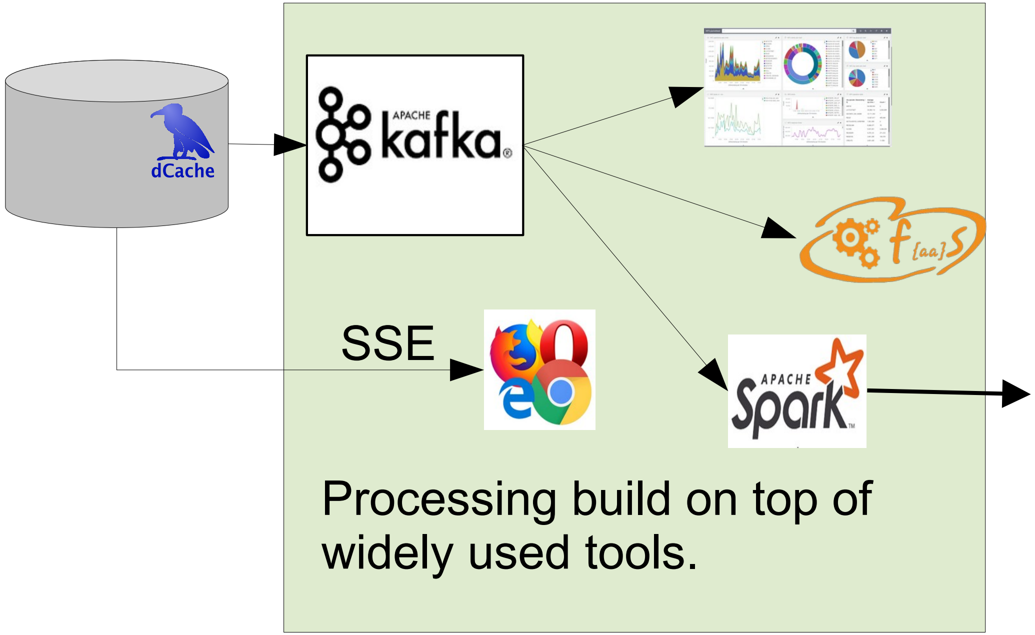
HPC vs. HTC (IO)



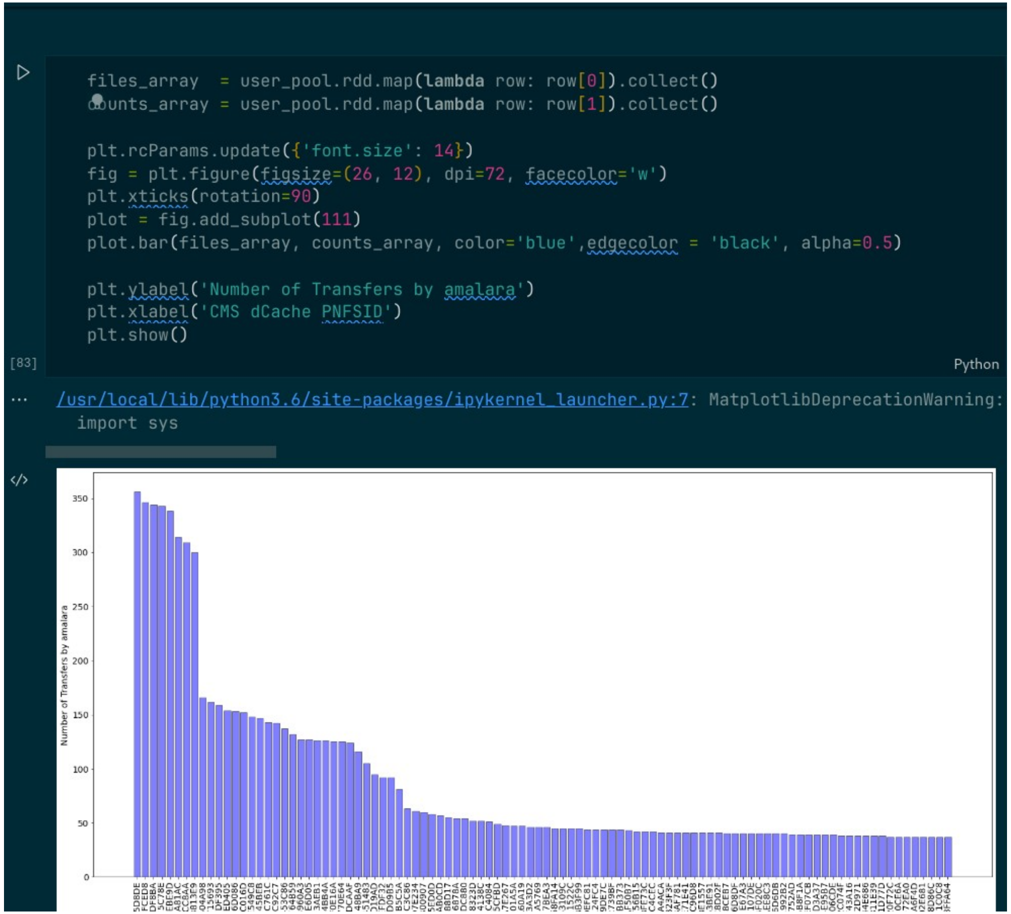
HPC vs. HTC (IO)



Big-Data Tools for Log Processing



Processing build on top of widely used tools.



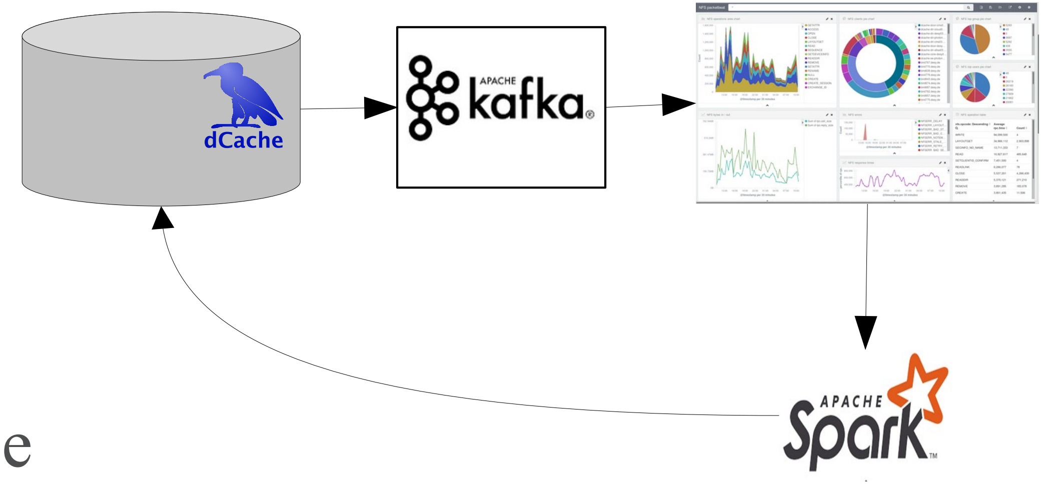
```
files_array  = user_pool.rdd.map(lambda row: row[0]).collect()
counts_array = user_pool.rdd.map(lambda row: row[1]).collect()

plt.rcParams.update({'font.size': 14})
fig = plt.figure(figsize=(26, 12), dpi=72, facecolor='w')
plt.xticks(rotation=90)
plot = fig.add_subplot(111)
plot.bar(files_array, counts_array, color='blue',edgecolor = 'black', alpha=0.5)

plt.ylabel('Number of Transfers by amalara')
plt.xlabel('CMS dCache PNFSID')
plt.show()
```

```
/usr/local/lib/python3.6/site-packages/ipykernel_launcher.py:7: MatplotlibDeprecationWarning:
  import sys
```

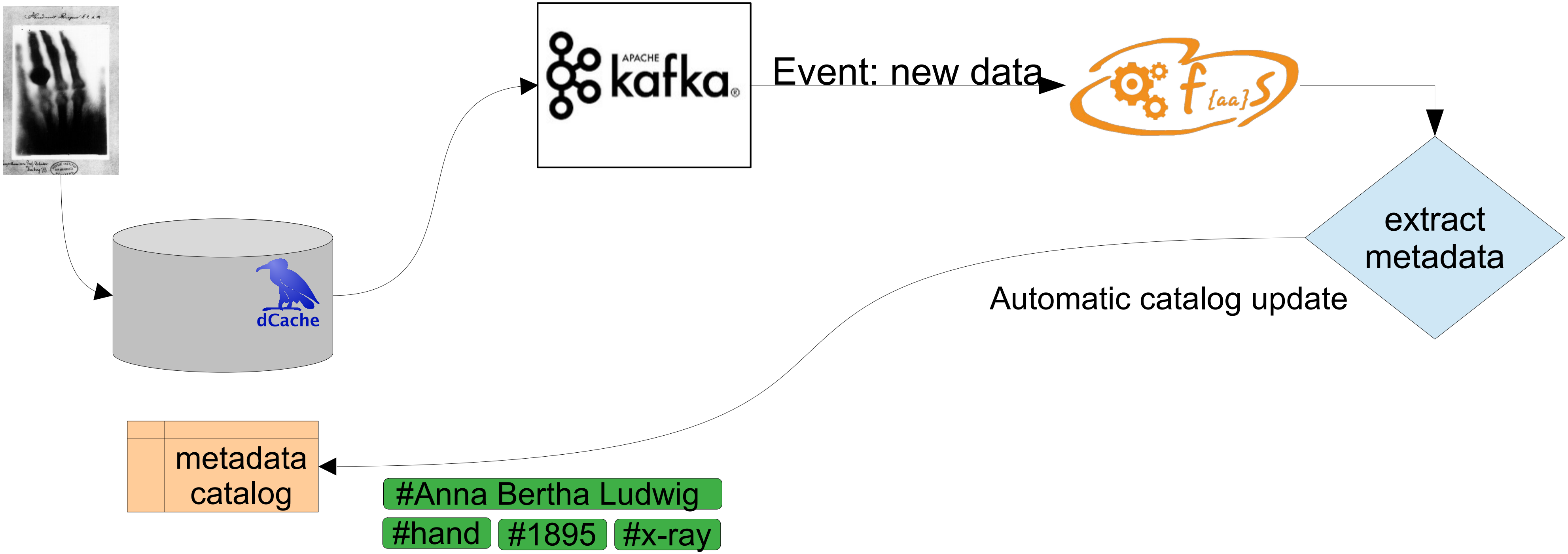
dCache ops @ DESY

Self-Adaptive dCache

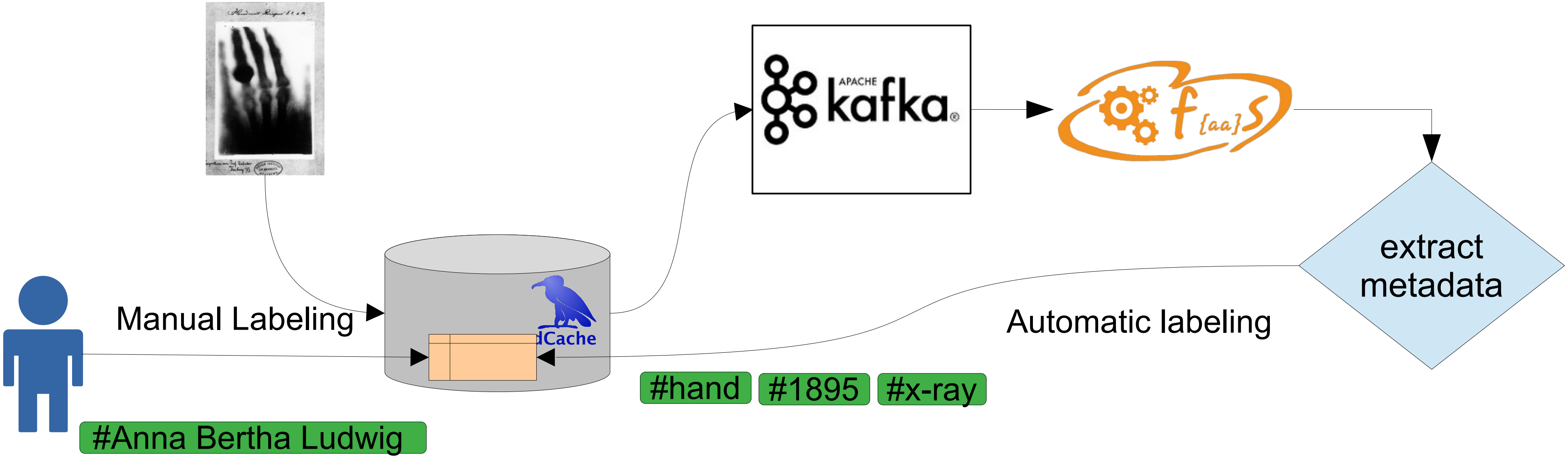
- Joint project with Hamburg University on Applied Science
  - MAPE-Loop
    - Automation of large deployments
    - Push-back batch system
    - Hotspot detection and re-balance
    - Self-healing load optimization
    - *Your imagination



Automatic Metadata Population



Metadata Population



User Metadata Handling

- User metadata important (again)
  - Data labeling/classification
- Can be populated by storage events
  - Some automation is required

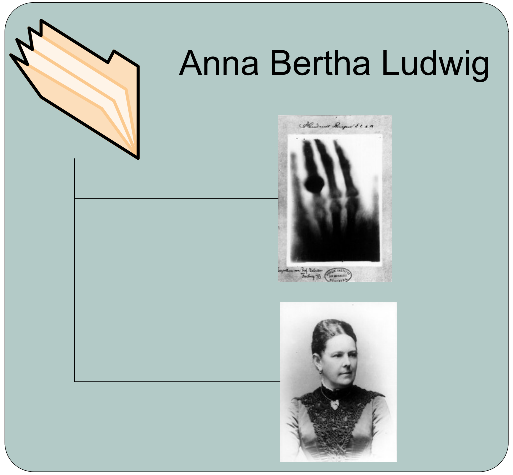
Hand mit Ringen
eigenthum von Prof. Zehnder
Freiburg i/B
PHYSIK. INSTITUT DER UNIVERSITÄT WÜRZBURG

#Anna Bertha Ludwig

#hand #1895 #x-ray

User Metadata/Labeling in dCache

- Extended attributes
  - Exposed via NFS, WebDAV, REST
- Label-based virtual **read-only** directories (WIP)
  - List all files with a given label
- dCache rules applies
  - Visible through all protocols
  - Respect file/dir permissions



User Attribute Support

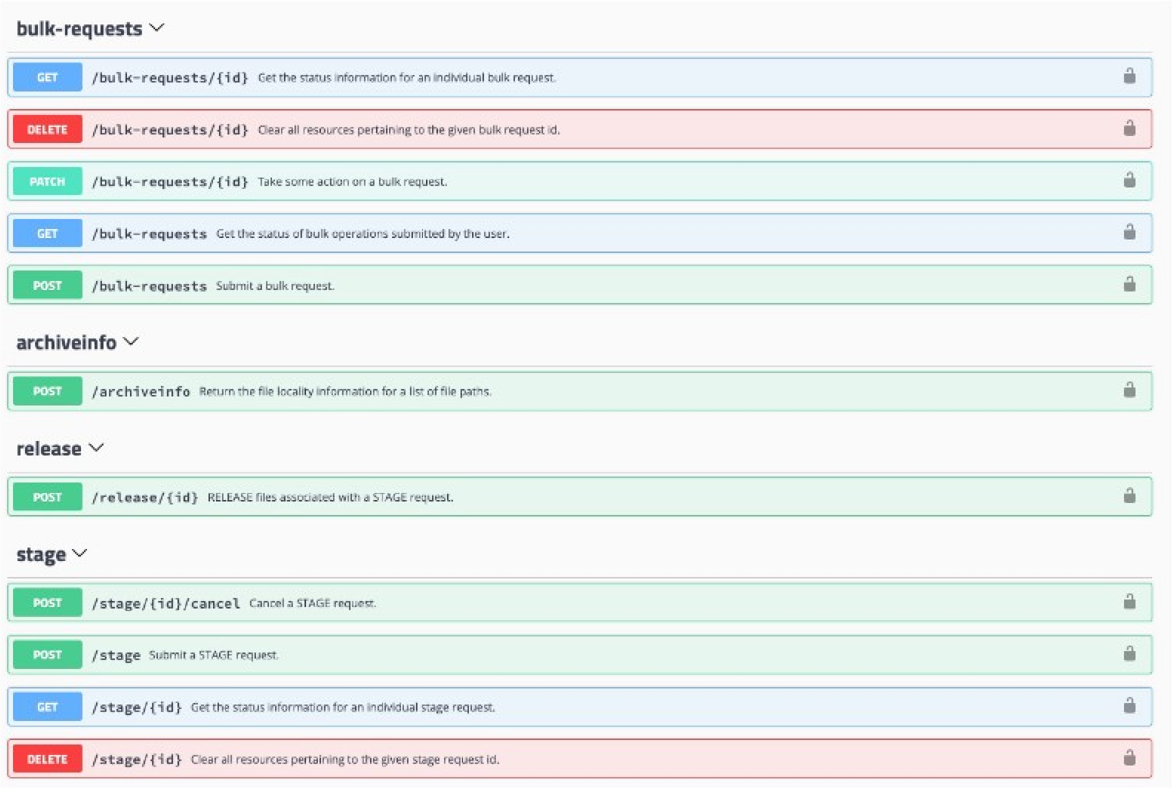
- HTTP(s)
  - As query option on upload
  - Those attributes are available to the flush process!
- ~~POSIX~~ xattrs
  - {get/set}fattr over NFS
  - Exposes directory tags
- File *tagging/labeling*

SRM Still Here ... *but not too long*

- Two main gaps to fill
  - Space allocation
  - Tape operation
- Two alternatives to replace
  - User and Group based Quota system
  - WLCG tape recall API

Tape rest API

https://example.org:3880/api/v1



bulk-requests

- GET /bulk-requests/{id} Get the status information for an individual bulk request.
- DELETE /bulk-requests/{id} Clear all resources pertaining to the given bulk request id.
- PATCH /bulk-requests/{id} Take some action on a bulk request.
- GET /bulk-requests Get the status of bulk operations submitted by the user.
- POST /bulk-requests Submit a bulk request.

archiveinfo

- POST /archiveinfo Return the file locality information for a list of file paths.

release

- POST /release/{id} RELEASE files associated with a STAGE request.

stage

- POST /stage/{id}/cancel Cancel a STAGE request.
- POST /stage Submit a STAGE request.
- GET /stage/{id} Get the status information for an individual stage request.
- DELETE /stage/{id} Clear all resources pertaining to the given stage request id.

dCache bulk API

WLCG Tape API

Tape REST-API v1 (like SRM, but different)

STAGE

- Request to stage many files at once

CANCEL

- Cancel bulk request

DELETE

- Cancel bulk request + clear history/status

EVICT

- unpin cached copy

PIN

- Pin cached copies with a lifetime

FILEINFO

- Request status many files at once (locality, checksum)

FTS

EOS

dCache

StoRM

User/Group Quotas

- **`Quota ≠ Space reservation`**
- Lazy, based on periodic scans
  - Users might overrun
  - Removed space not reclaimed immediately
- Global per file system
  - No quota per directories
- Respects Files Retention policy
  - Separate for 'disk' and 'tape' files
- Available since 7.2, enabled by default since 8.2

Non Functional Developments

- Documented release/test process
- Shareable build pipelines
  - Can be replicated at sites
- Transparent release process
- K8S based deployment
- Code will stay on Github

K8S Based Testing

- Sites can reproduce our release process
- dCache containers available at docker hub
- Helm carts to deploy dCache with three commands

```
$ helm install dcache-db bitnami/postgresql
$ helm install cells bitnami/zookeeper
$ helm --set image.tag=9.2.0 my-tier-2 dcache/dcache
```

Technical Directions

- Scaleout
  - Namespace
  - Number of pools (SW/HW)
- BULK operations
- Token-based Authentication
- Better *Analysis Facility* support
  - POSIX access and compliance
  - HPC workload support (DDoS protection)
- QoS
- Tape integration

Call to Action

- You can contribute with ...
  - Code
  - Configuration
  - Testing
  - HW setup
  - Knowledge
- You can make dCache visible with …
  - Sharing your use case
  - Demonstrate dCache use in various events

18th International dCache Workshop June 6-7,DESY-Hamburg

More info:

https://dcache.org

To steal and contribute:

https://github.com/dCache/dcache

Help and support:

supportꝑdcache.org, user-forumꝑdcache.org

Developers:

devꝑdcache.org

Scientific Data Challenges

Ingest

- High data ingest rate
- Multiple parallel streams
- High durability
- Effective handling of large number of files

Analysis

- High CPU efficiency
- Chaotic access
- Standard access protocols
- Access control
- Local user management

Sharing & Exchange

- 3rd party copy
- Effective WAN Access
- In-flight data protection
- Identity federation
- Access control

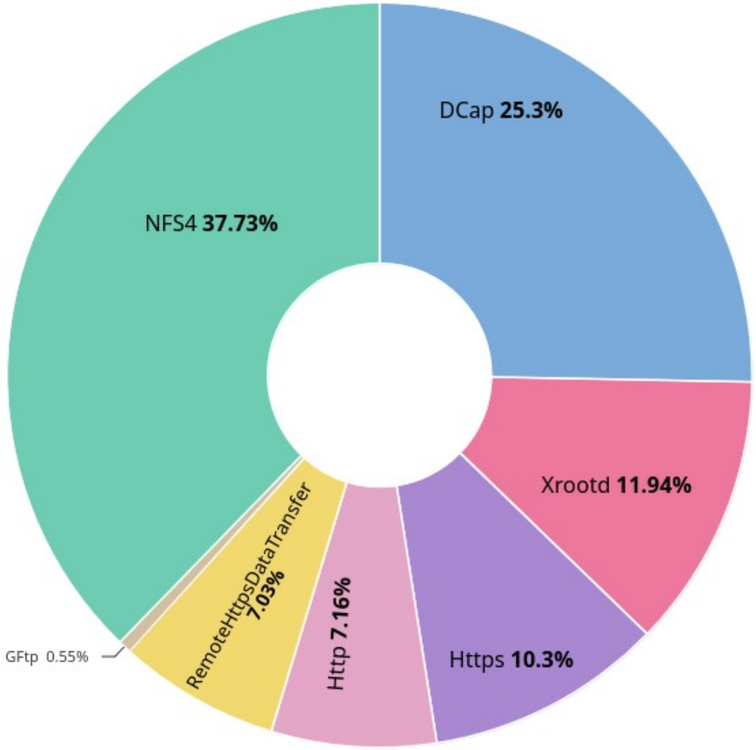
Long Term Preservation

- High Reliability
- Self-healing
- Automatic technology migration
- Persistent identifier

Prominent Changes

- QoS & BULK Service
- TPC improvements
- NFSv4.1/pNFS improvements
- XROOT evolution (TLS, tokens, TPC, proxy-IO)
- Namespace performance improvements
- HSM connectivity

All Protocols and Instances


DCap 25.3%
NFS4 37.73%
Xrootd 11.94%
Https 10.3%
Http 7.16%
RemoteHttpsDataTransfer 7.03%
GFtp 0.55%