dCache Setup DESY

DESY/KIT Workshop on GPFS, dCache and Tape

Christian Voß
Void of Zoom, 14th July 2022

HELMHOLTZ

DESY.

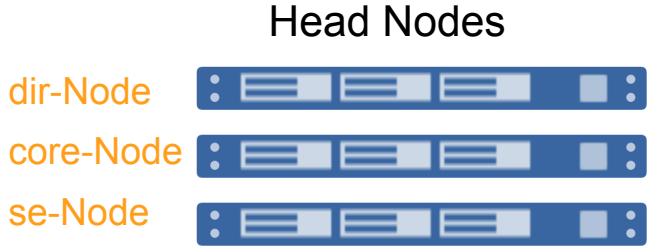
Our dCache Setup

Same setup across instances except for XFEL (dedicated DB node)

- Currently 100% bare metal deployment

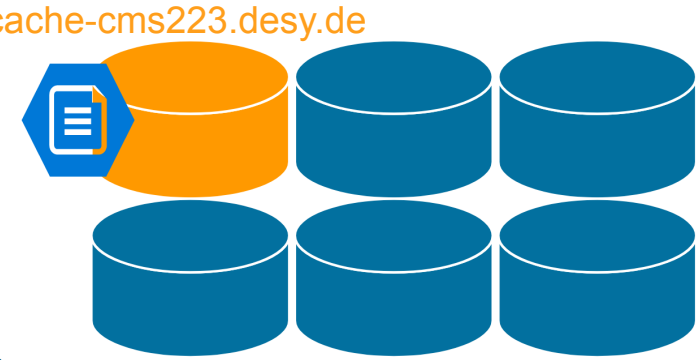
Heads

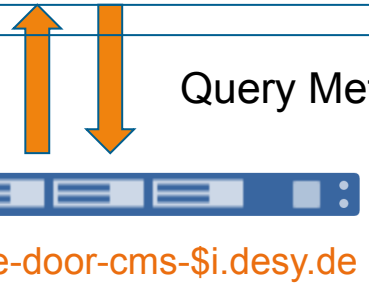
- Host databases and replicas (split Chimera/SRM)
- Run most services in HA when applicable
- Refrain from SrmManger and PoolManager
- Old school Grid-SE-endpoint



Pools

- Individual storage nodes with Raid-6 setup
- Select slices out of Raid-6 as Pools





Doors

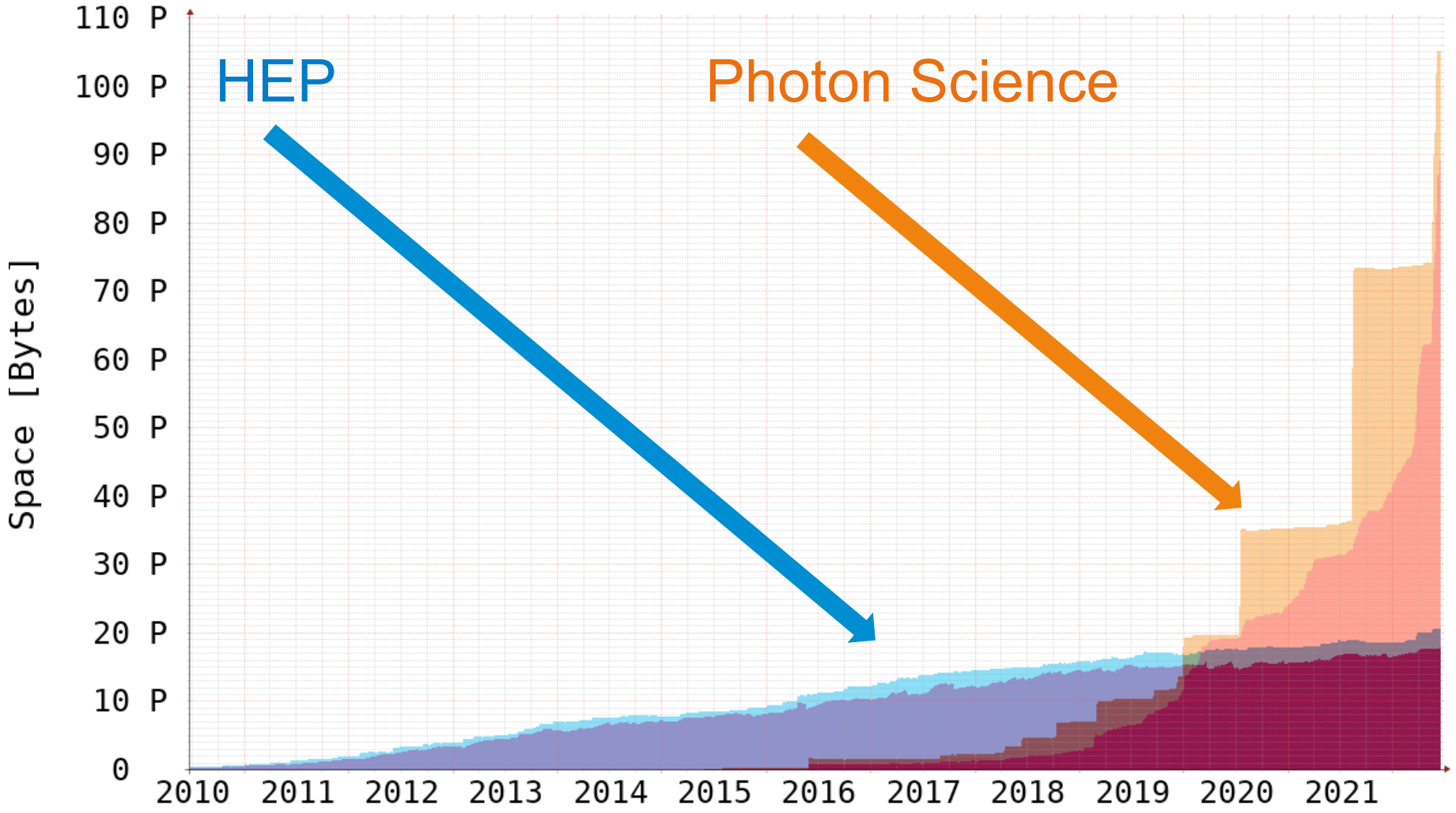
- Still on bare metal, but due to replace in 2021/22
- Plan to switch to virtualised nodes
- Select different distribution of protocols based on door role

dCache Disk Capacity over Time

Exponential Increase Rate due to XFEL

Start of Eu. XFEL

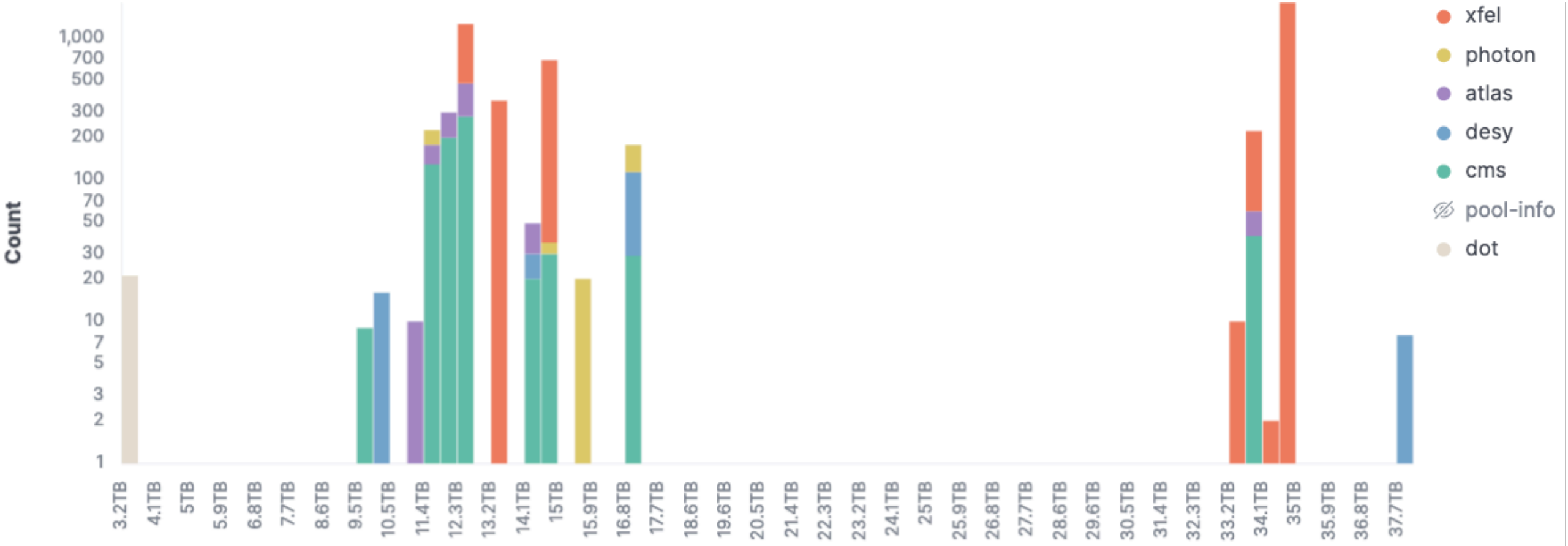
- Observe almost exponential increase in capacity
- Install 20-30PiB in January
- Further increase by 20 PiB in process
- We several issues with scale



Pool Sizes

Idea: Constant Pool Size to Improve Parallel Fill Level and Improve Start-Up

- Grown from 100s of pools into 1000s of pools with the rise of XFEL and 500 pool nodes
- Tend to still keep our pools relatively small, with about a dozen per pool node
- Reasoning in the past: pool start ups, often have millions of files on quite small pools
- Observe scaling issues: especially I/O competition during write and flush, there are always rebuilds in process



DESY Specific: NFS and dCap as Pivotal Protocols

Out Local Access Patterns dominated by NFS

- Access pattern differ a lot from a regular Grid-Ses
- XrootD and WebDav to dominate

dCap

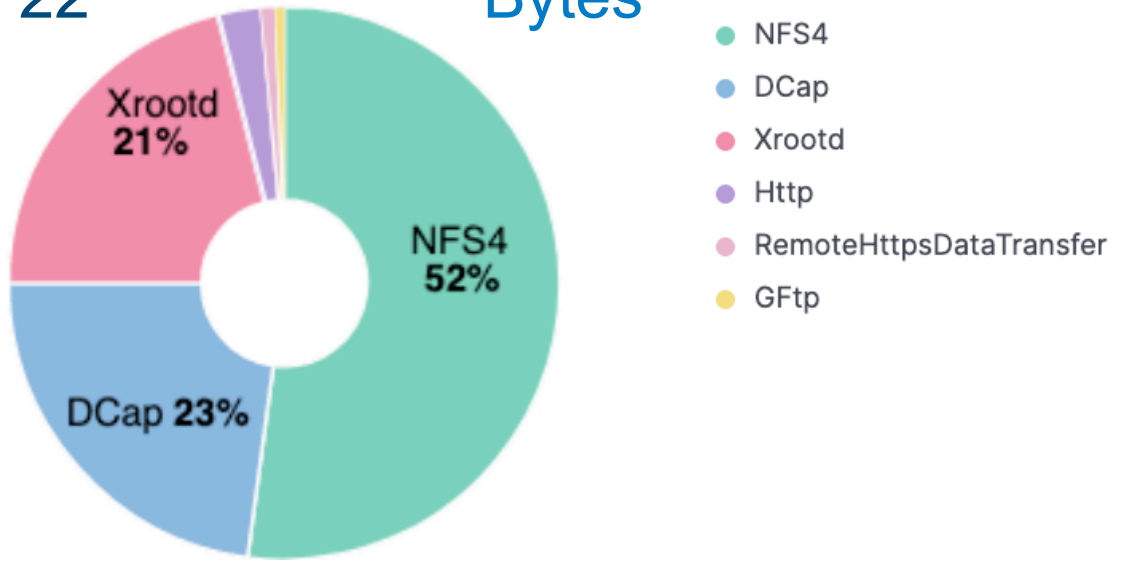
- Still in use as primary protocol for CMS
- dCap saw a revival with Photon Science
- Most efficient way to maximise throughput for XFEL
- Most efficient way to write a million PETRA III files

NFS

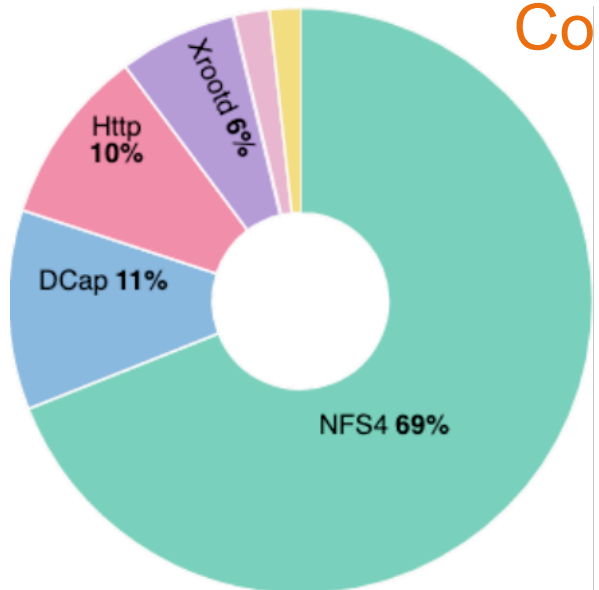
- NFS dominant protocol on NAF
  - Belle@NAF : ~100%
  - CMS@NAF : ~60%
  - ATLAS@NAF : ~25%
- Photon Science uses NFS only for HPC cluster
- Rise of new tools saw move to NFS
- Other protocols not supported, local file always available

June ‘22

Bytes



Counts



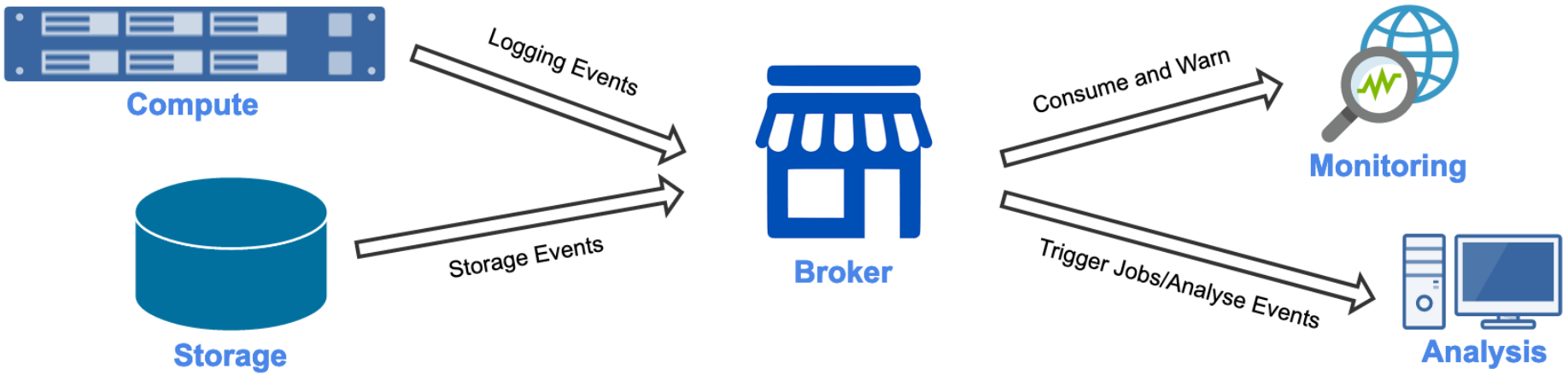
Monitoring for dCache

Using Events to Control Flow of Monitoring Data

- dCache produce large amounts of data about its operation, e.g. billing data
- Build an message stream for dCache billing and logging data based on Apache Kafka → Monitoring becomes a consumer of these messages

kafka

Message Producer – Broker – Consumer Model



> Close to how dCache works internally (remember assigning `core` Domains?)

> Categorise messages/events in different manner

- Message points to data, need interface to analyse data: Storage events
- Message contains all the data: Self-contained events

From dCache to Analysis

Connecting dCache with Monitoring Infrastructure

- Apache Kafka common IT tool → a lot of services can subscribe and consume Kafka messages

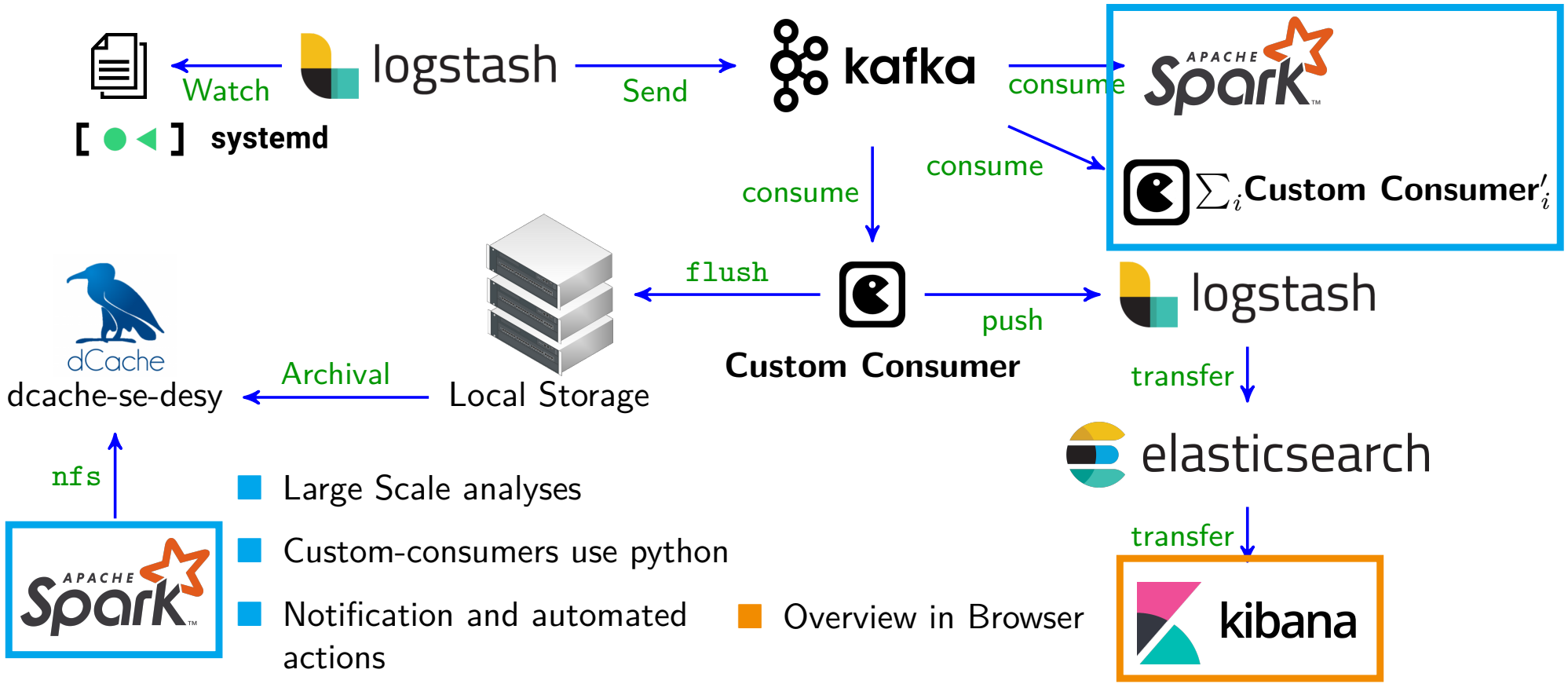
- Beats can easily push JournalD, Syslog, … into Kafka
- Logstash accepts Kafka streams → direct in ingest and visualisation with Elastic tool box (classic monitoring)

Benefit of Plugging Kafka in Between?

- Have operations triggered on Kafka messages → Listen passively rather than search actively
- Often have well known signatures in billing/logging → introduce an automatic response (alarm or fix)
- Ubiquity of Kafka gives rise to easy-to-use libraries in C++, Java or Python
- Have a number of lightweight python based consumers in operation listening for specific patterns
- Complex data analysis frameworks can consume and analyse Kafka events at large

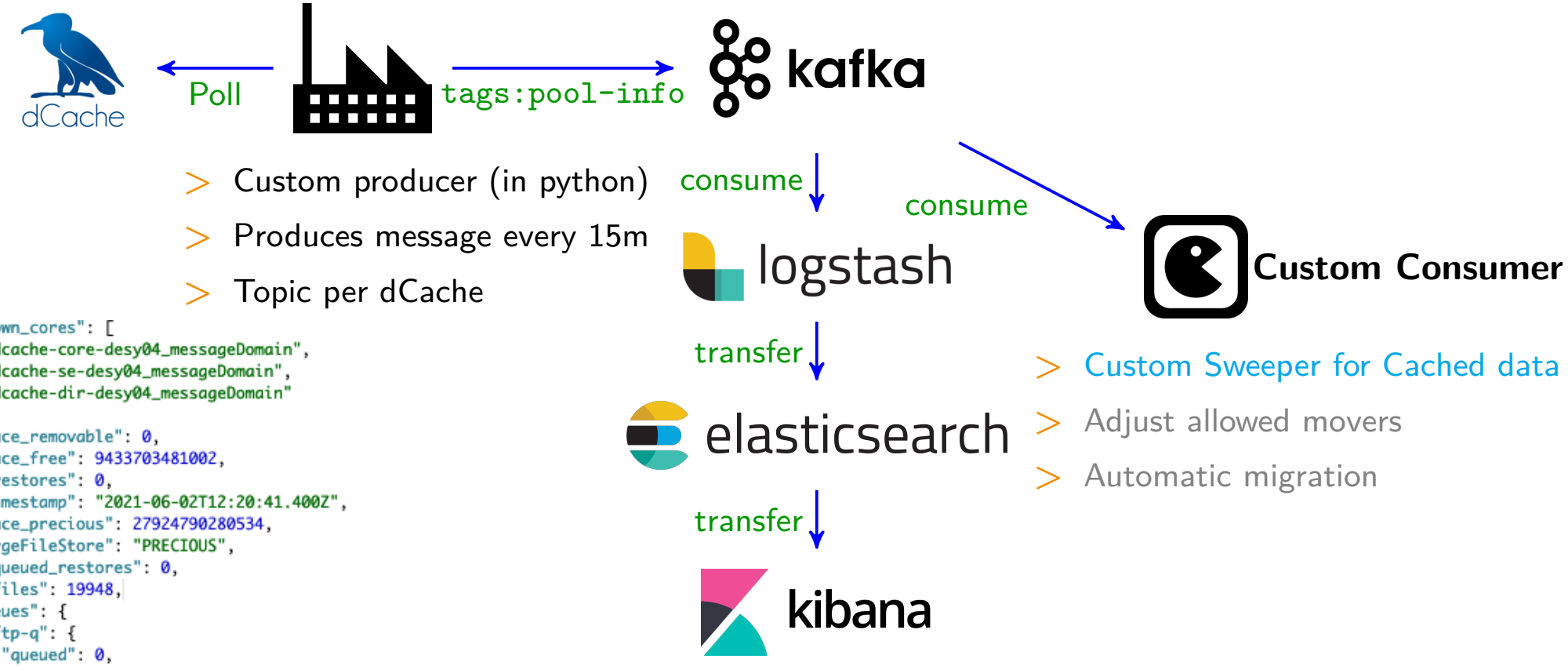
Full dCache Monitoring Workflow

Example for Loggings Stream



Example of Custom Data Stream

Collect Data for Each Pool Every 15min




```
"known_cores": [
  "dcache-core-desy04_messageDomain",
  "dcache-se-desy04_messageDomain",
  "dcache-dir-desy04_messageDomain"
],
"space_removable": 0,
"space_free": 9433703481002,
"n_restores": 0,
"@timestamp": "2021-06-02T12:20:41.400Z",
"space_precious": 27924790280534,
"LargeFileStore": "PRECIOUS",
"n_queued_restores": 0,
"n_files": 19948,
"queues": {
  "ftp-q": {
    "queued": 0,
    "active": 0
  },
```


- Pool plot shown earlier based on this stream
- Also archived on dCache → allows forensics on which pool belonged where in the past

Recent Development: Propose to use QoS for Customers

Allow Better Control on Data Locality

- General XFEL policy (copy on disk and tape) not strictly enforced in the beginning
- Started with classic setup: files written as precious with NEARLINE state afterwards
- Setting PnfsManager default accordingly

- Often found files without disk copy due to being cached
- Introduced a pin based workflow (initial pin lasting a year)
- XFEL admins missed to reapply pins → triggered disappearance of disk copies and restages
- Asked us to pin data for 10years (a.k.a. infinity) on writing to not bother with it themselves
- Idea: unpin when capacity became critical
- Observed issues with pins → not tracked errors on pins/repins: trigger major restore campaign

Sounds like a use case for QoS shown during dCache workshops?

- Files are supposed to be on `disk+tape` until changed to `tape`
- So rather than unpin, XFEL would use a QoS transition → matching exactly their desired data policy
- Added bonus: delegates SRM to second line since REST-API can be used
- Photon Science unfamiliar with Grid infrastructure, certificates a tedious affair

Current Issues with dCache

Possible Links to GPFS

- Raid-6 pool setup start to turn into a liability:
  - Increasingly large number of storage nodes (starting to fill up our data centre)
  - Increased disk size lead to considerable rebuild times
  - Number of pools causes side effects (e.g. Info-system rendered broken)
  - Performance issues when exposed to excessive user access

- Look at alternatives: GPFS or CEPH
- GPFS:
  - Lots of experience but in different mode than would be suited for dCache
  - Maybe use a similar model as KIT (no native raid)
- CEPH:
  - Open source solution
  - Use CEPHFS as foundation of pools
  - Significant lack of experience and men-power at DESY: research project
- Big solution to manage the 100 PiB of XFEL dCache storage

Tape at DESY

With dCache as Main Customer

- Use TSM to store general purpose backups
- dCache main customer with ~100PiB of data

- Switch to Tigran’s Talk for details on tape

Thank you