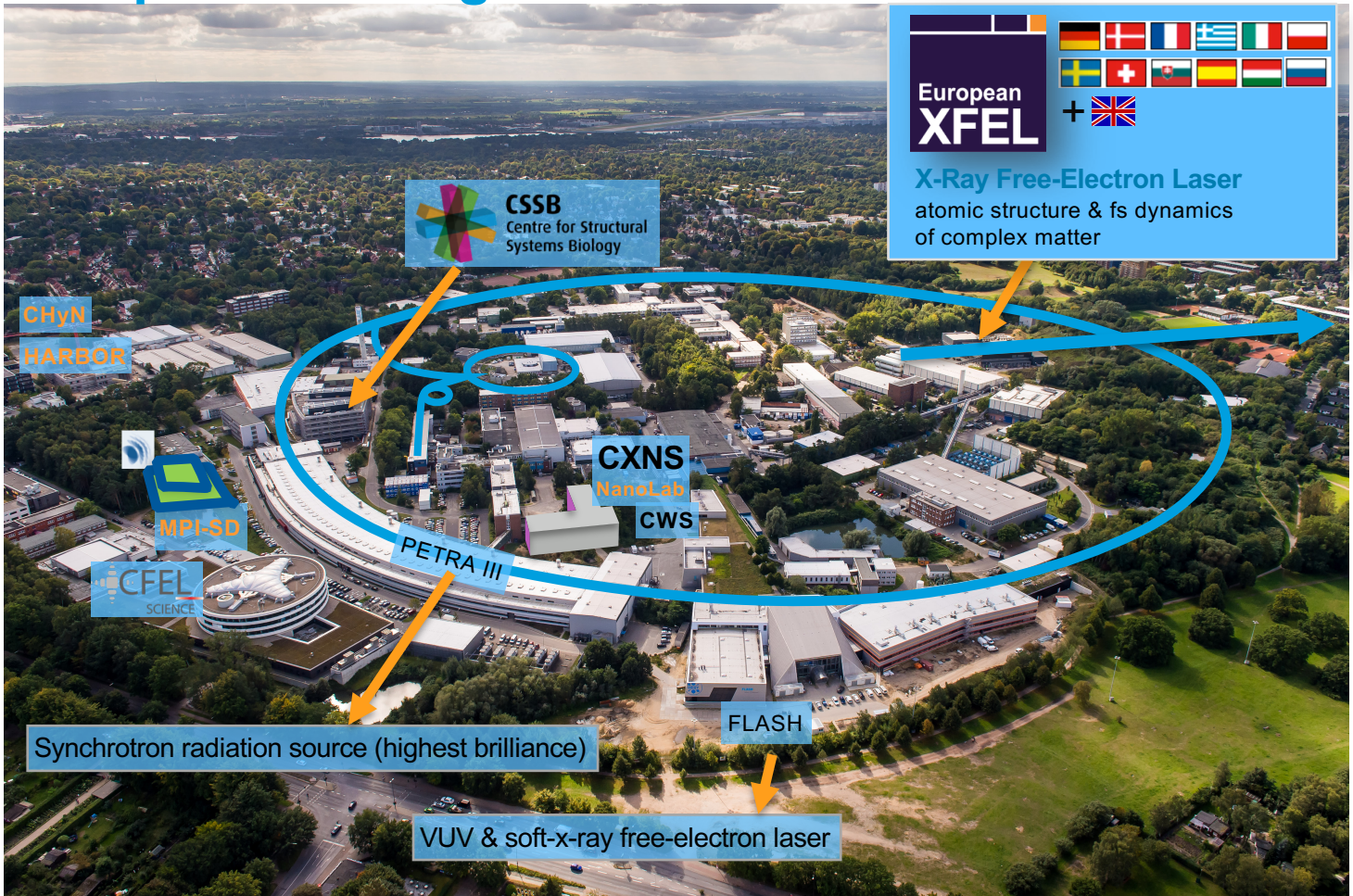
Online computing for new generation photon science experiments

M. Gasthuber, S. Yakubov, S. Dietrich, J. Hannapel, C. Patzke - DESY/Hamburg

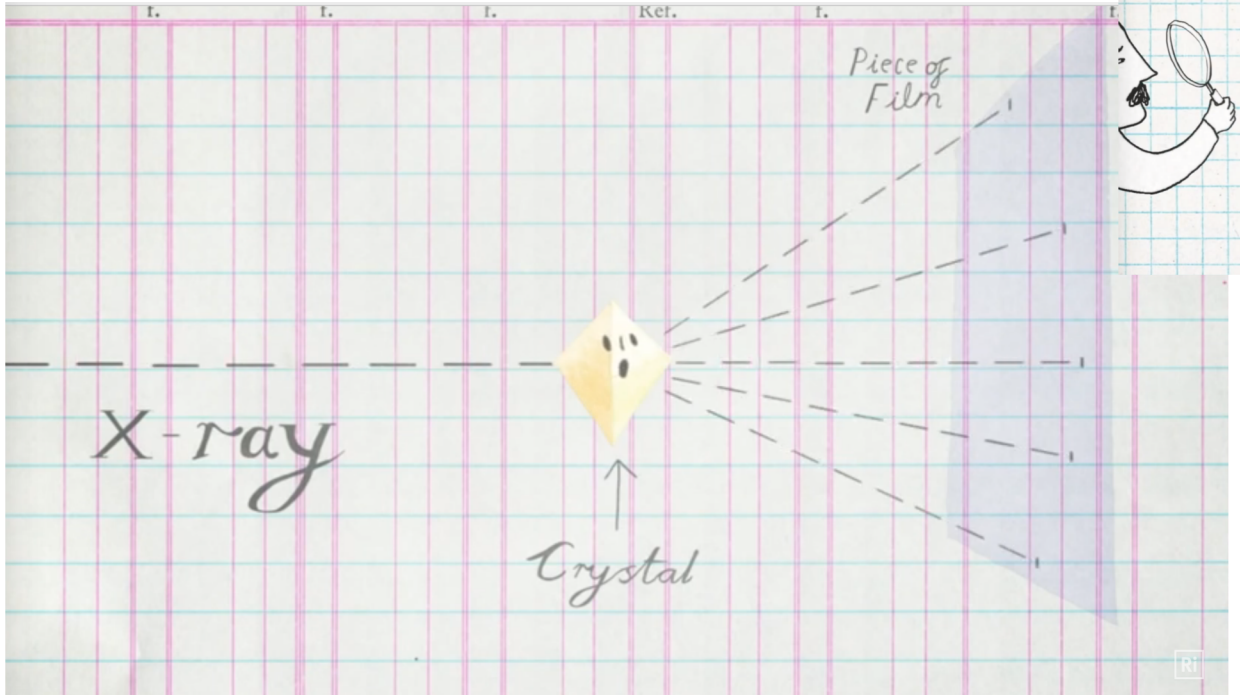
DESY Campus Hamburg – much more communities

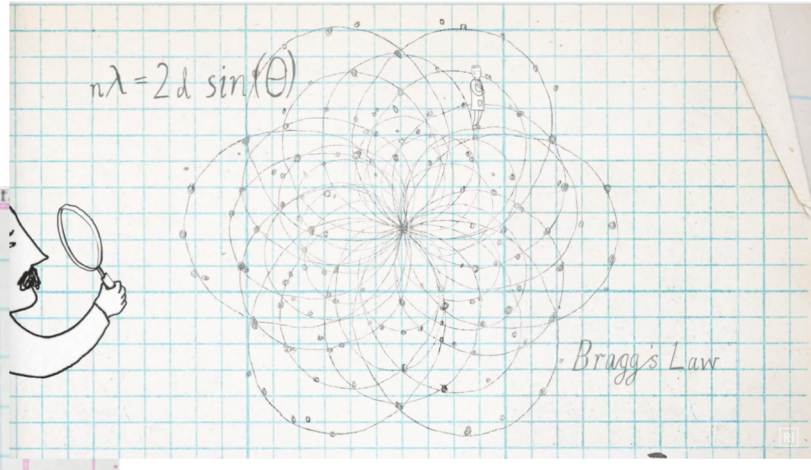


why online

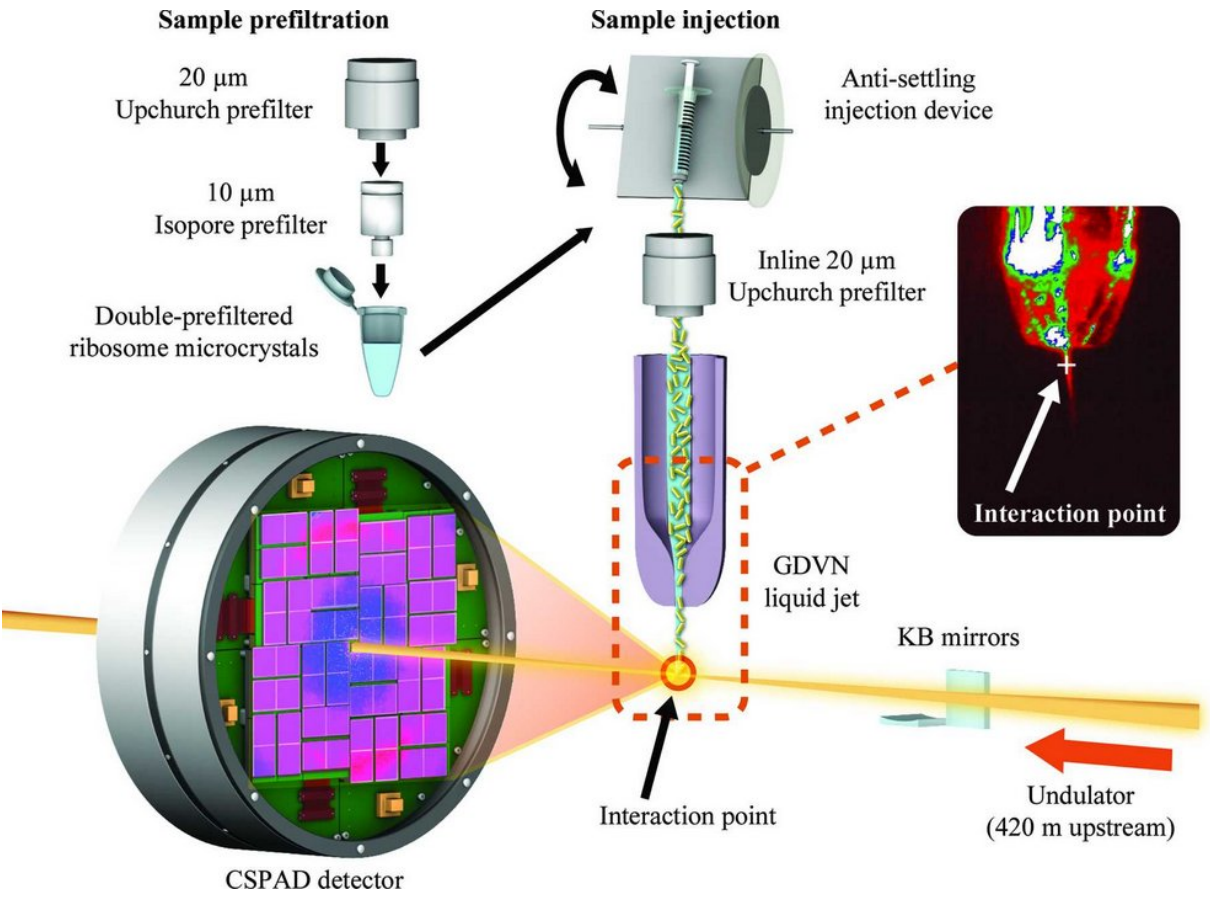
- upgraded / new accelerators – lumi equivalent (data volume driving) increases
- accelerated detector development – bigger, faster
- experimental setup…

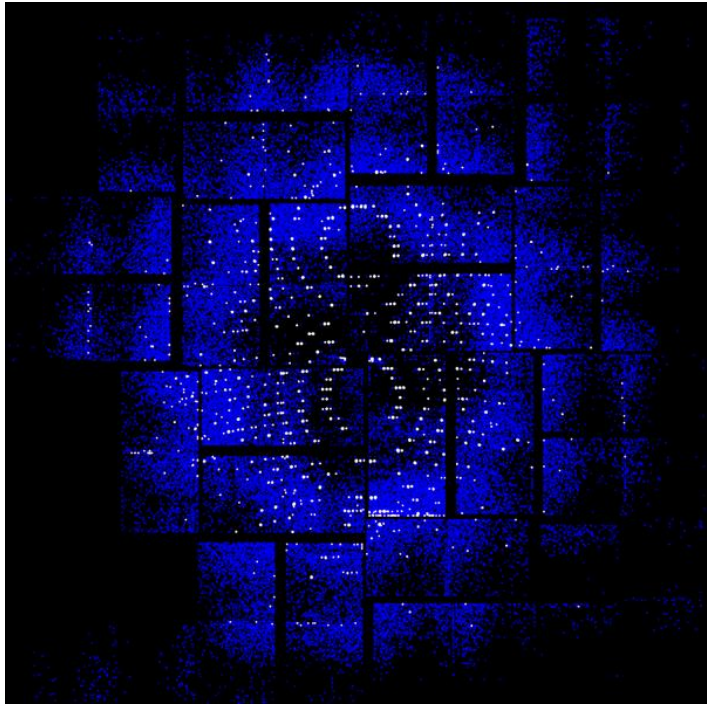
from … (oversimplified ;-)





to …


Sample prefiltration
20 µm
Upchurch prefilter
10 µm
Isopore prefilter
Double-prefiltered
ribosome microcrystals
Sample injection
Anti-settling
injection device
Inline 20 µm
Upchurch prefilter
Interaction point
GDVN
liquid jet
KB mirrors
Interaction point
Undulator
(420 m upstream)
CSPAD detector

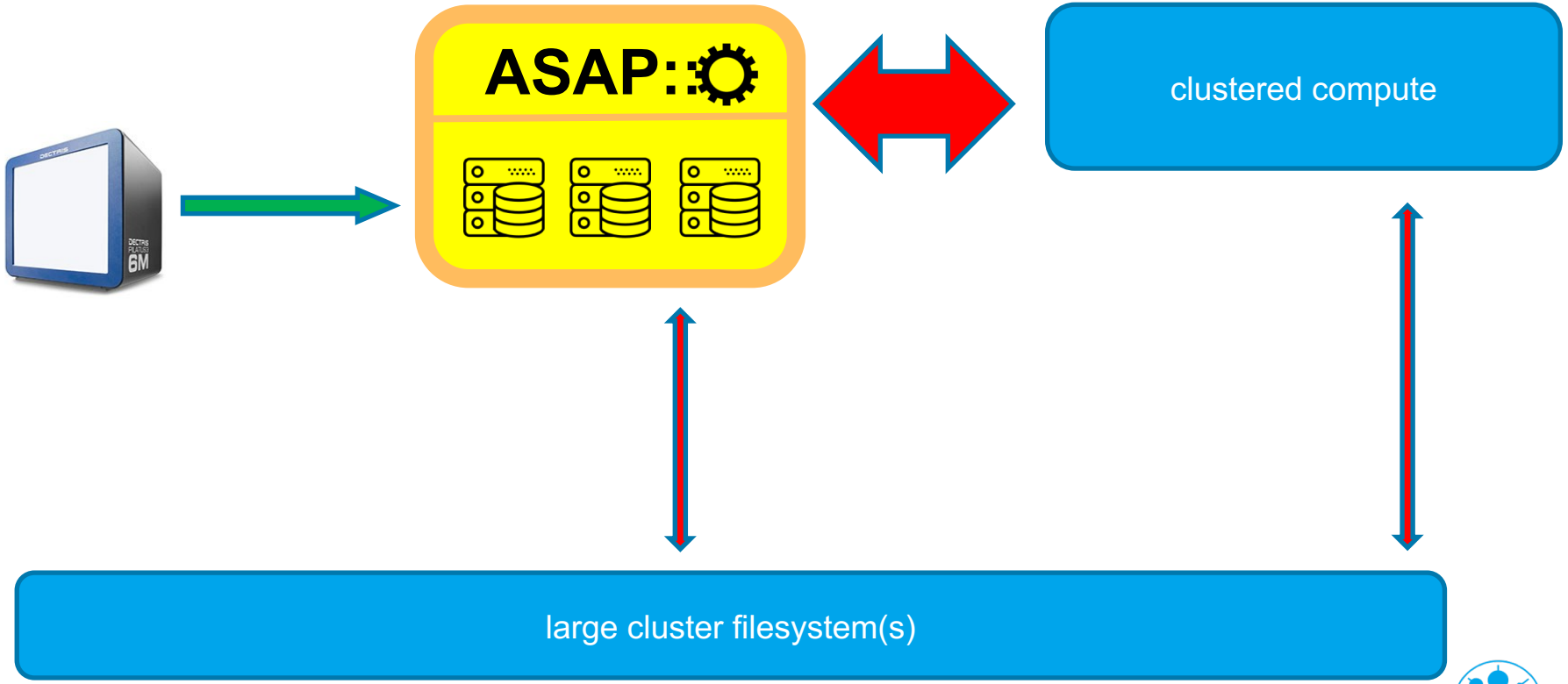


transition / not shown data archive and export

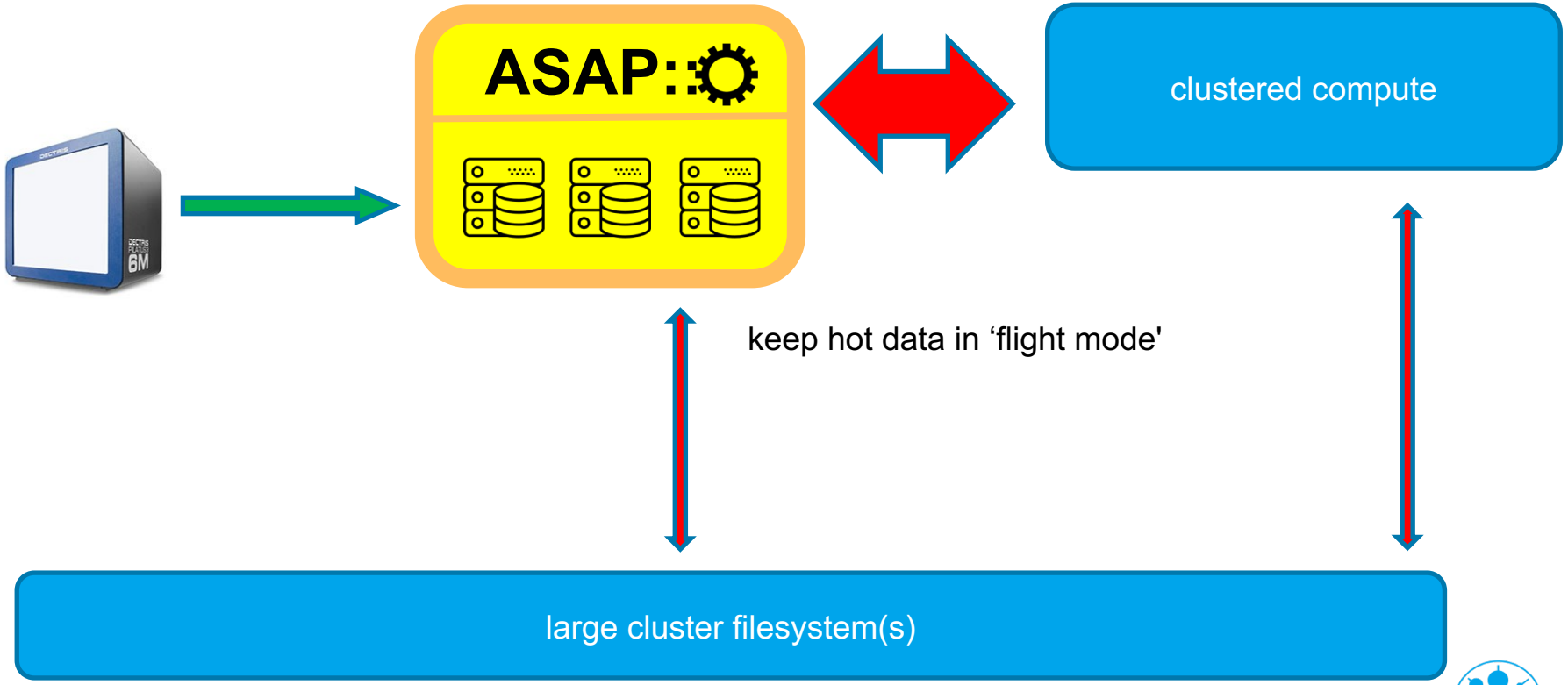
clustered compute

large cluster filesystem(s)

transition / not shown data archive and export



transition / not shown data archive and export



challenges

not just technical

- technical
  - KHz detectors (100KHz in development) – 4-8 GB/sec… and growing
  - multi detector experiments – event builder ?
  - new type/complexity of experiments – exploding demands for storing and computing – not only offline
  - beside detector all is shared – compute, storage, network
  - commercial detectors – limited (mostly none) access, fixed data egress options (i.e. HTTP get)
  - legacy apps for analysis (i.e. tied to POSIX access)
  - from ‘my-ana.py’ to MPI based distributed analysis code – all you can think of
  - Windows and very old Linux versions to be integrated – mostly at detector server level

- non technical
  - hundred of small groups – uneven computing skilled
  - different wording and history – as every scientific community

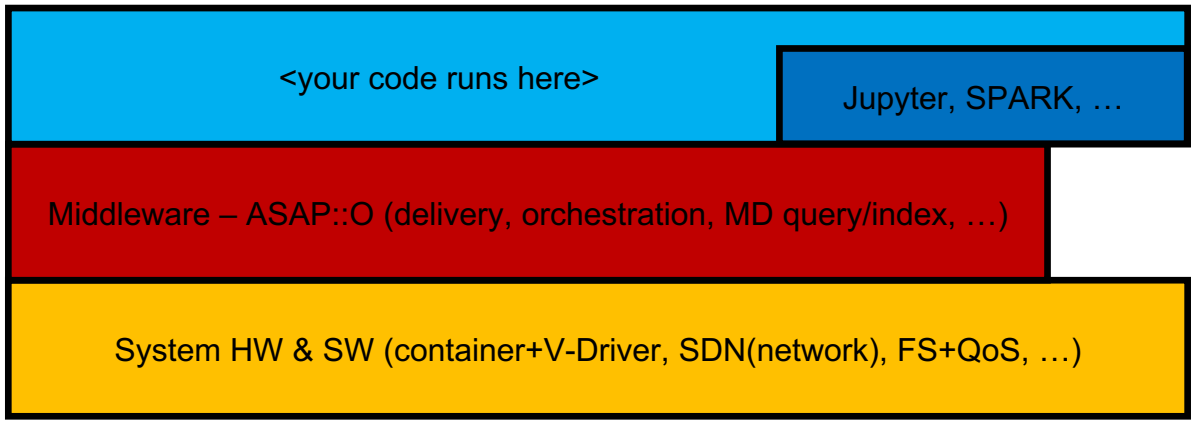
heading to … per beamline

- ~50GB/sec aggregates ingest rate of all sources (10 - 30min average) – daily averages to ~5 GB/sec
- ~1PB of ’hot’ data to work on (eventually store much less ;-)
- network - N x 200Gbs with low ‘jitter’ of latency and bandwidth
- 1k cores (CPU+GPU) with 100-300 k core-hours of compute work
- archive data for 10 years – with 1 week delay after creation (second copy asap)

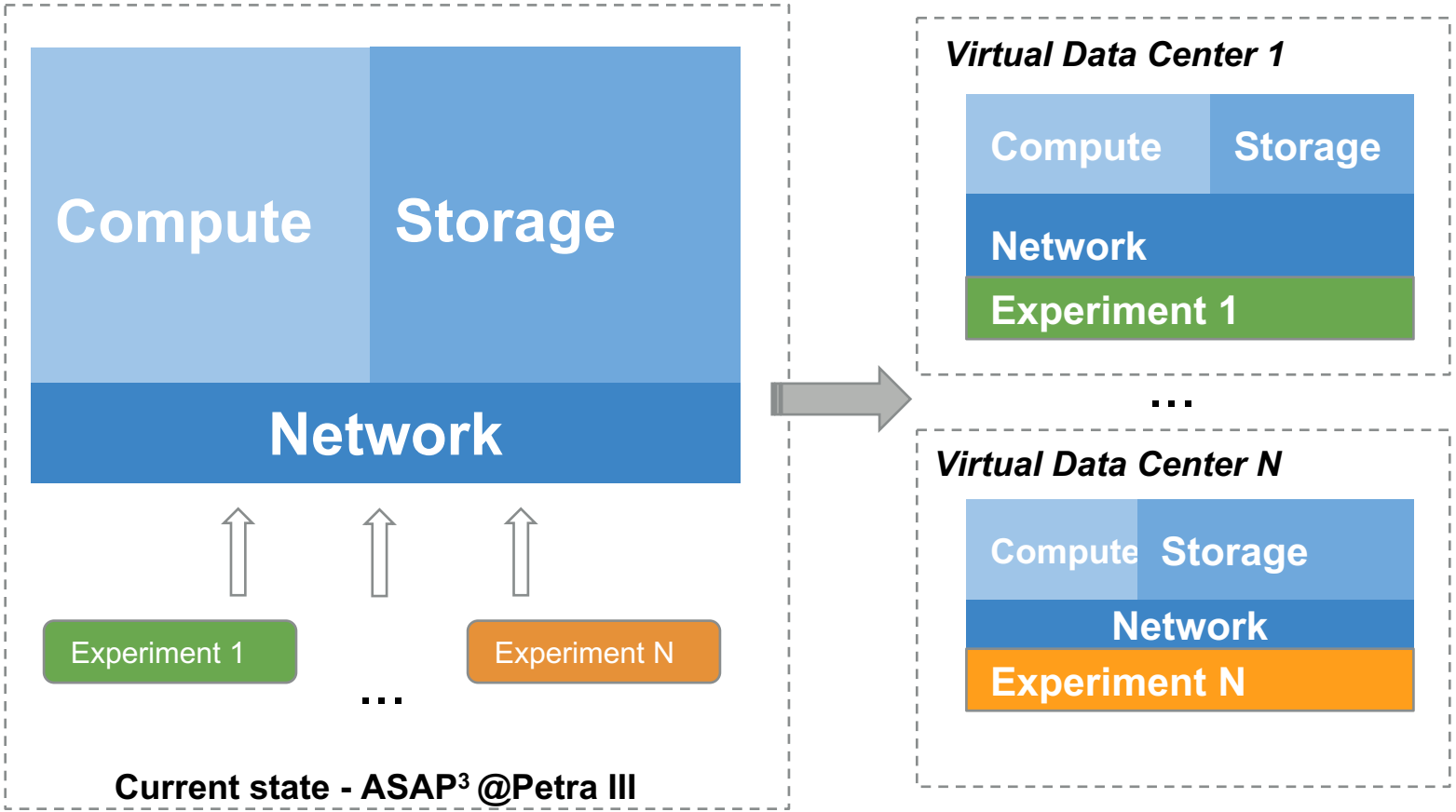
- from beamline view
  - wait 1-3 secs after ingesting a few 1000 images to be processed (generating synthetic image)
    - tune / reconfigure experiment (handcrafted)
  - …

levels to work on

- **system level**
  - container, storage, network – transition to ‘virtual datacenter’ shared nothing
- **middleware**
  - common services, abstract system level, landing ground for scientific analysis payload

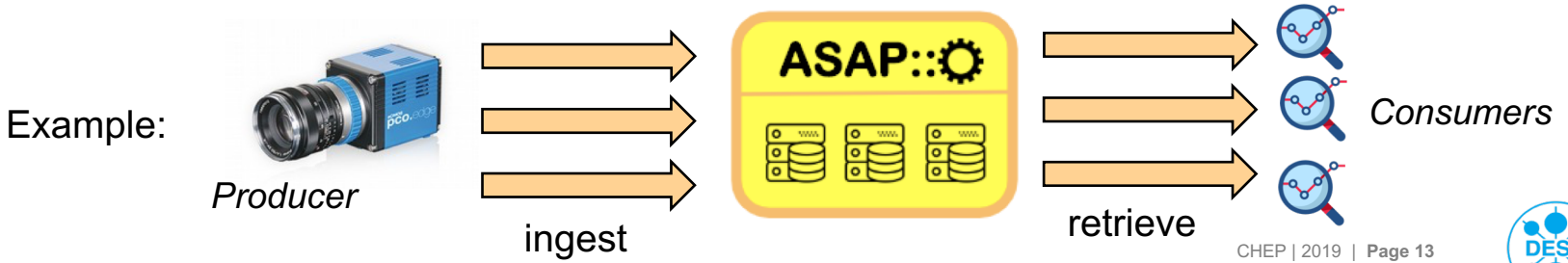


shared all to *shared nothing* / dealing with “noisy neighbor”



ASAPO (High Performance Online Data Analysis) – the middleware

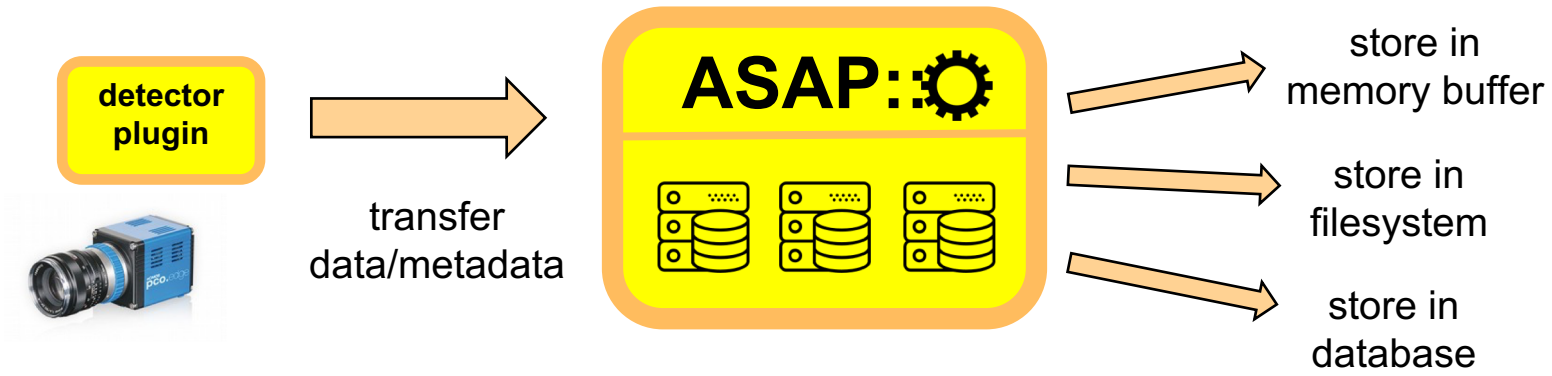
- middleware for high-performance next-generation detector data analysis
  - Provides API to inject data to the system - e.g takes care of the “first mile” between the experimental hall and the compute center (high-performance data transfer)
  - Provides API to retrieve data from the system - e.g. for data analysis synchronous (online) and asynchronous (offline) to data taking
- Basic characteristics
  - Scalable (N sources, K network links, L service nodes, M analysis nodes)
  - Highly available (services in Docker containers managed by Nomad/Consul)
  - Efficient (C++, multi-threading, RDMA, …)
  - Provides user friendly API interfaces (C/C++, Python, REST API)
  - Runs on Linux/Windows/...



ASAPO - Ingest (detector data)

> “First mile” details

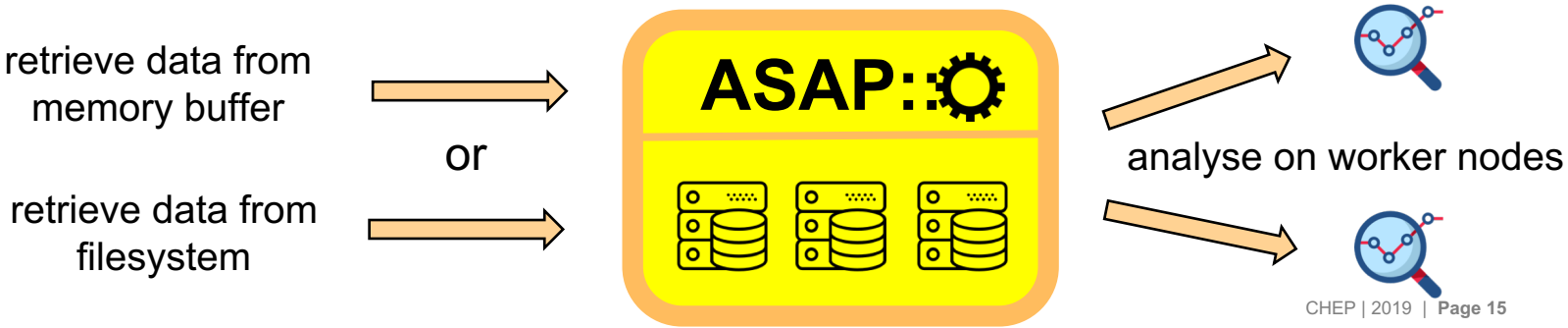
- Detector plugin (folder monitoring, ZeroMQ, HTTP, ...) or via HiDRA
- Data transfer (Ethernet, RDMA, …) – observing i.e. FairMQ, Mercury,…
- Metadata can be added (global per beamtime, local per image). Can be used to work with multiset images.
- Transfer acknowledged as completed after data was received, stored in a filesystem and recorded in a database (MongoDB)
- Retransfer in case of failure



ASAPO - Retrieval

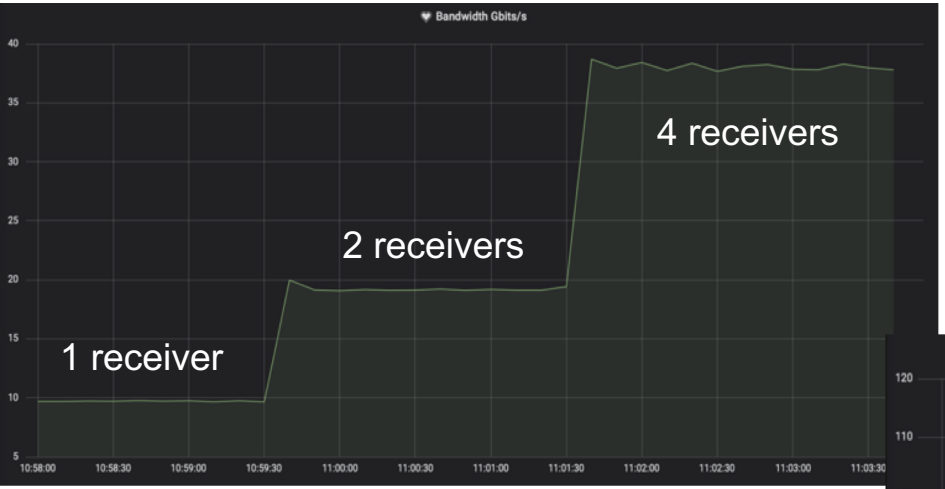
> Data analysis functionality

- Parallel data processing (GetNextImage() can be called from multiple nodes in parallel). Blocks until new image is there
- Online visualization (GetLastImage() – always returns last image).
- Same code for online/offline. If data is not in buffer, it'll be read from filesystem
- Random access (GetImage(id=111))
- Get data based on metadata (create a query, get all data that match this query or pass it to GetXX)
- Working with multiset images – GetXX returns set of images instead of a single one



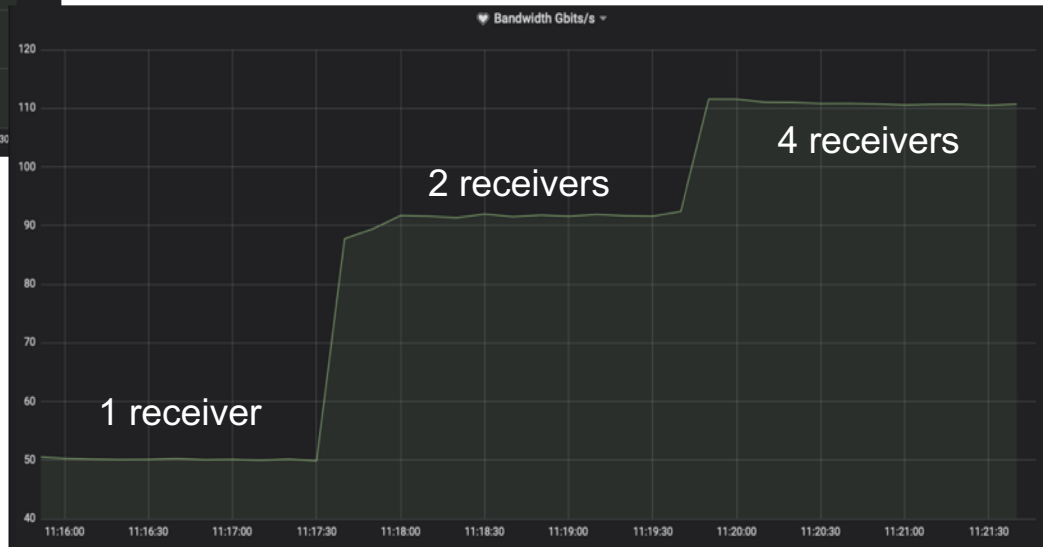
ASAPO - Performance/Scalability tests – Ethernet/IB IP

> Multiple senders, dynamically changed number of receivers, no write to disk



Ethernet / IP

InfiniBand / IPoIB

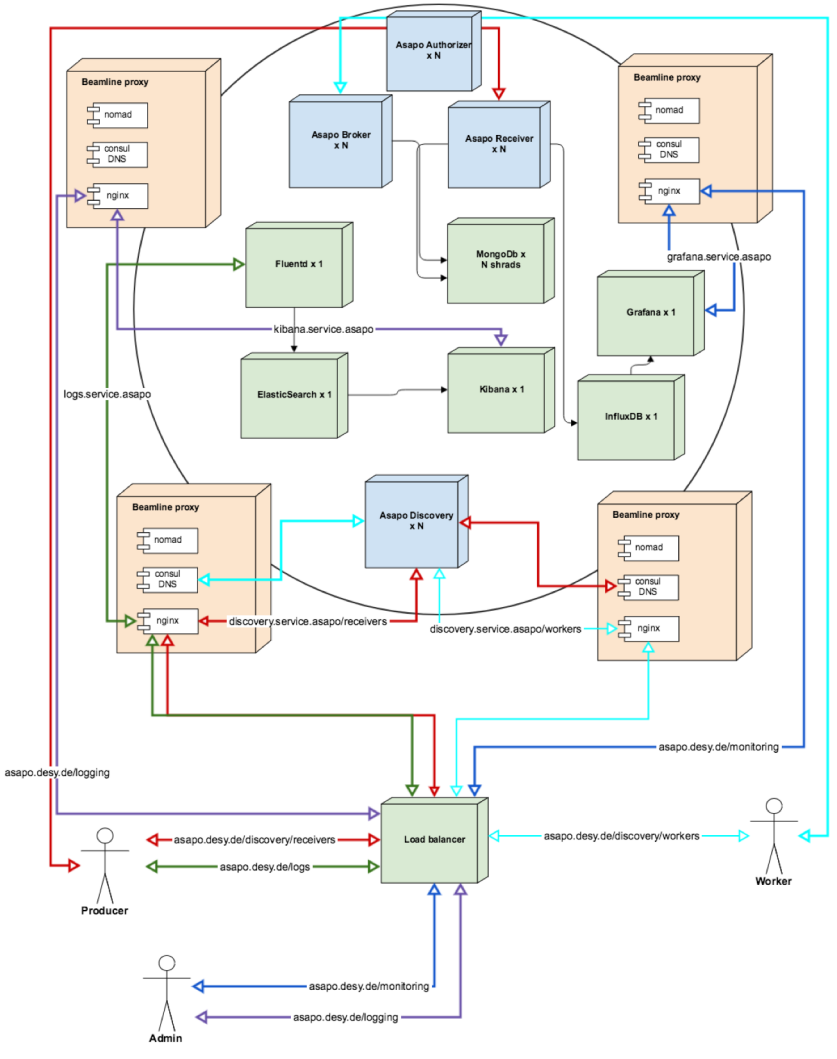


initial RDMA test (IB) saturated 4xEDR links
with just a few % of cpu costs – promising !

core status & architecture

- current setup based on clustered shared FS, MongoDB and memory buffers (cache) – all distributed
- MongoDB – 7K insert and 12K retrieves / sec for single instance – OK for today

- looking for
  - fast KV Store based on fast Flash/PM
    relax demands on cluster-FS
  - simple/fast query (integrated in KV Store) (similar to DAQDB presented on Tuesday)



ASAP::O – A Crystallography Experiment with PILATUS 6M Detector

Detector generates a new image – saves a file to RAM disk

ASAP::O monitors RAM disk and detects the new file

ASAP::O sends the file to the data center

```
In [2]: %matplotlib inline

In [3]: import asapo_worker
        import numpy as np
        import matplotlib.pyplot as plt
        import tempfile
        import cbf

        broker, err = asapo_worker.create_server_broker("asapo-server:8400",

In [4]: data, meta, err = broker.get_last(meta_only=False)
        plt.imshow(data, cmap='gnuplot', vmax=500)
        plt.show()
```

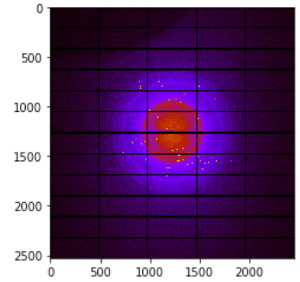


Image appears in a Jupyter Notebook

PETRA III Experiment Hall

A web browser somewhere in the world

summary

- first beamline adopted – will start within next months
  - low demands on performance
- on the list
  - provenance – system generated/maintained
  - CWL integration
  - CI/CD
  - … unknown unknowns

DESY