# GPFS in data taking and analysis for new light source (X-Ray) experiments @DESY

**formerly known as *Large Data Ingest Architecture***

Martin Gasthuber, Stefan Dietrich, Marco Strutz, Manuela Kuhn,
Uwe Ensslin, Steve Aplin, Volker Guelzow / **DESY**
Ulf Troppens, Olaf Weiser / **IBM**

GPFS-UG Austin, November 15

HELMHOLTZ ASSOCIATION

DESY

# DESY – where & what & why

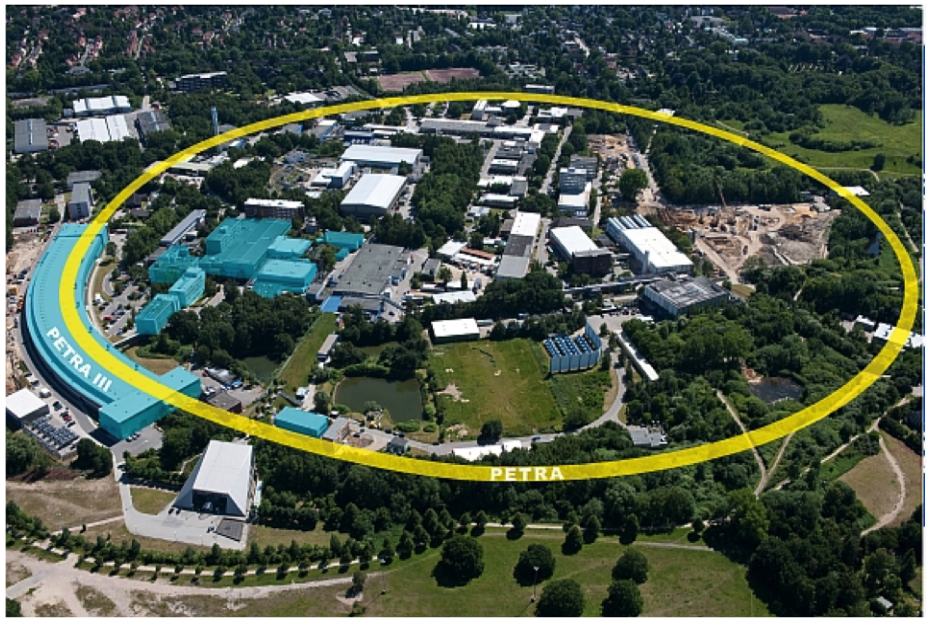
- founded December 1959
- research topics
  - Accelerator Science (>5 already built)
  - Particle Physics (HEP)

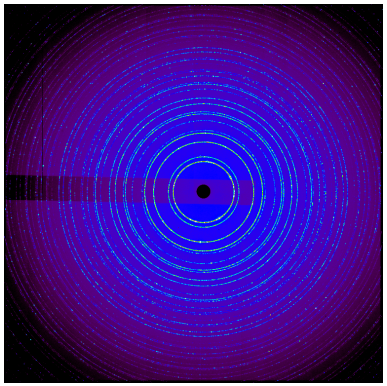
    i.e. Higgs@LHC, Gluon@Petra
  - Photon Science (physics with X-Rays)

    i.e. X-ray crystallography,
    looking at viruses to van Gogh
    and fuel cells
  - Astro Particle Physics
- 2 Sites
  - Hamburg/ Germany (main site)
  - Zeuthen / Germany (close to Berlin)
- ~2300 employes, ~3000 guest scientists annually

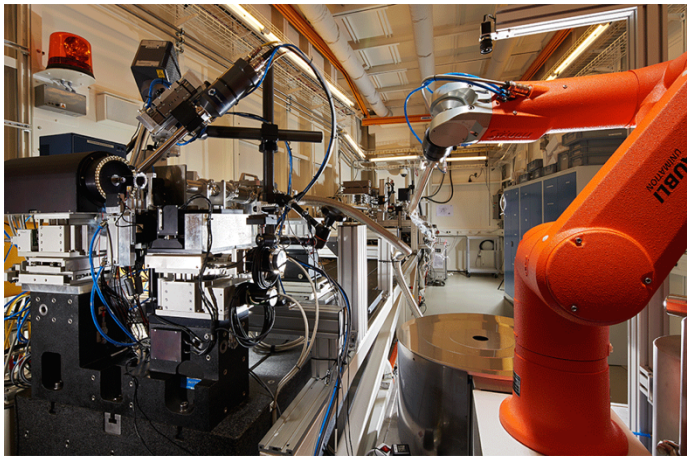
# PETRA III

- storage ring accelerator – 2.3 km circumference
- X-ray radiation source
- since 2009: 14 beamlines in operation
- 2016: 10 additional beamlines (extension) start operations

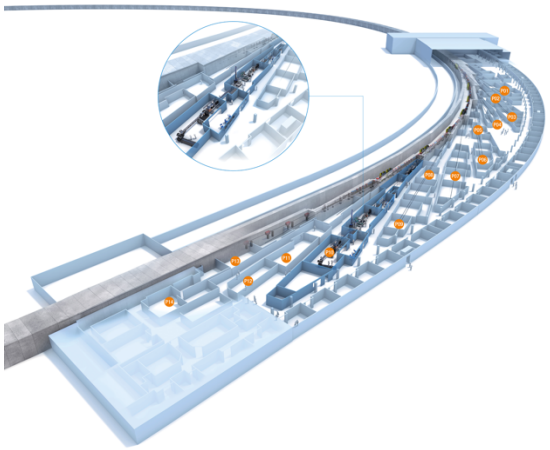


Sample raw file

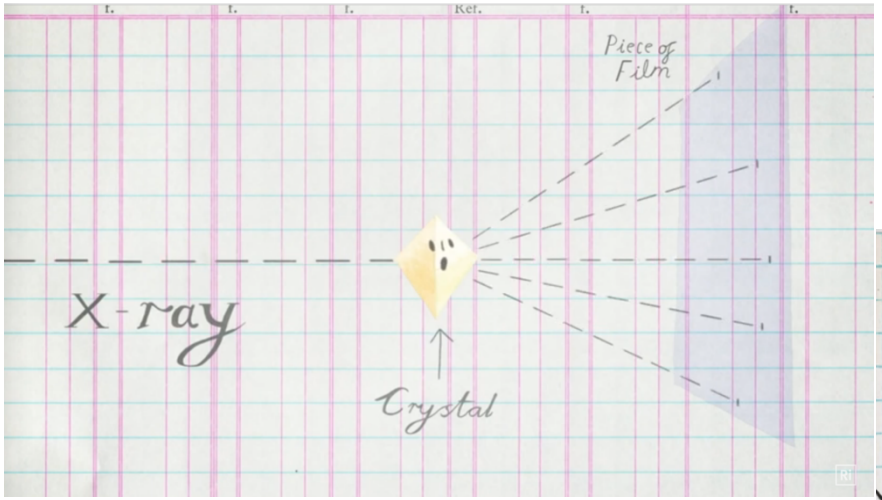


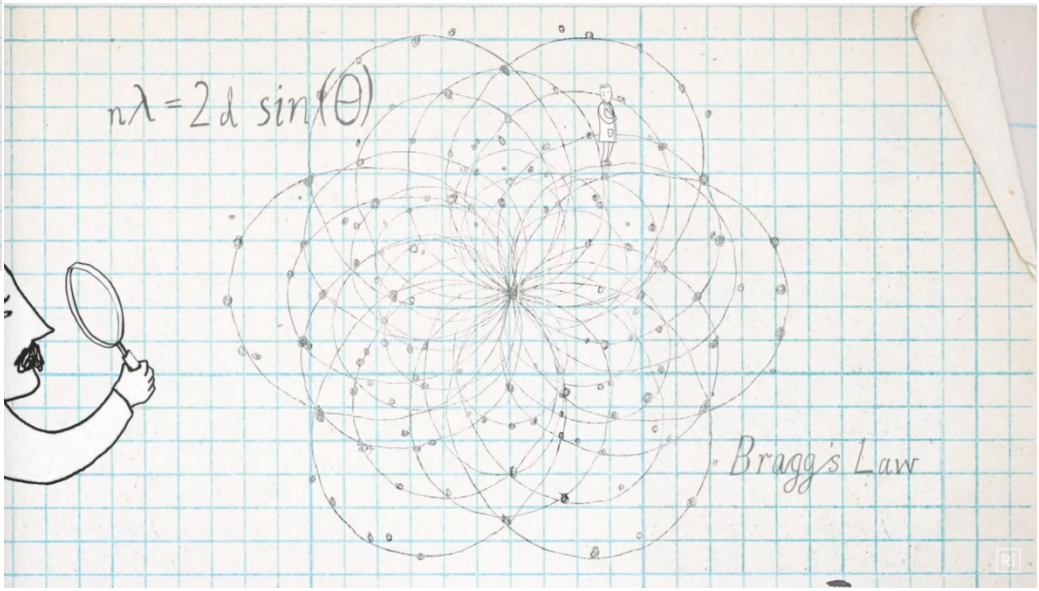
Beamline P11
Bio-Imaging and diffraction



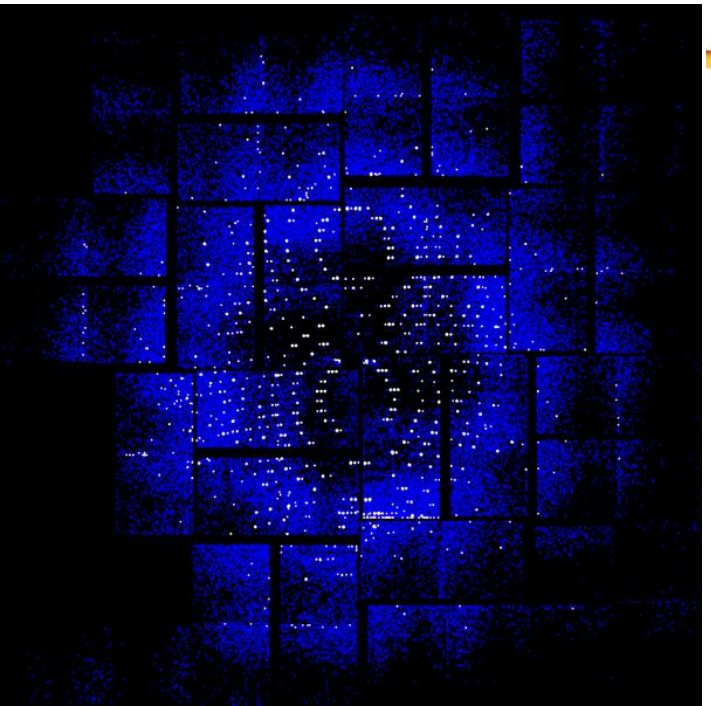
# why building a new online storage and analysis system

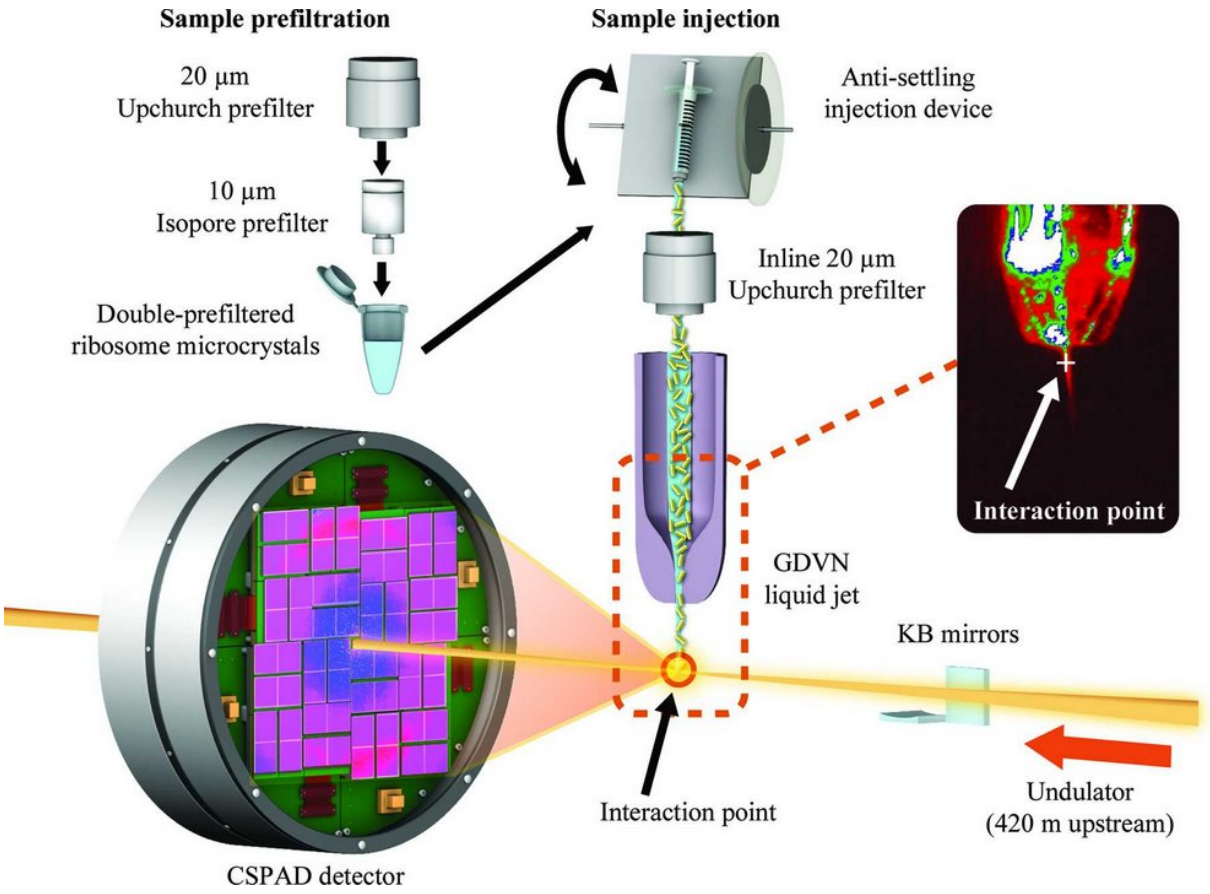
- massive increase in terms of #files and #bytes (orders of magnitude)
- dramatic increase in data complexity ⟹ more IO & CPU
  - from crystal bound targets to liquid bound targets



# becomes – less precise, more complex, more iterative



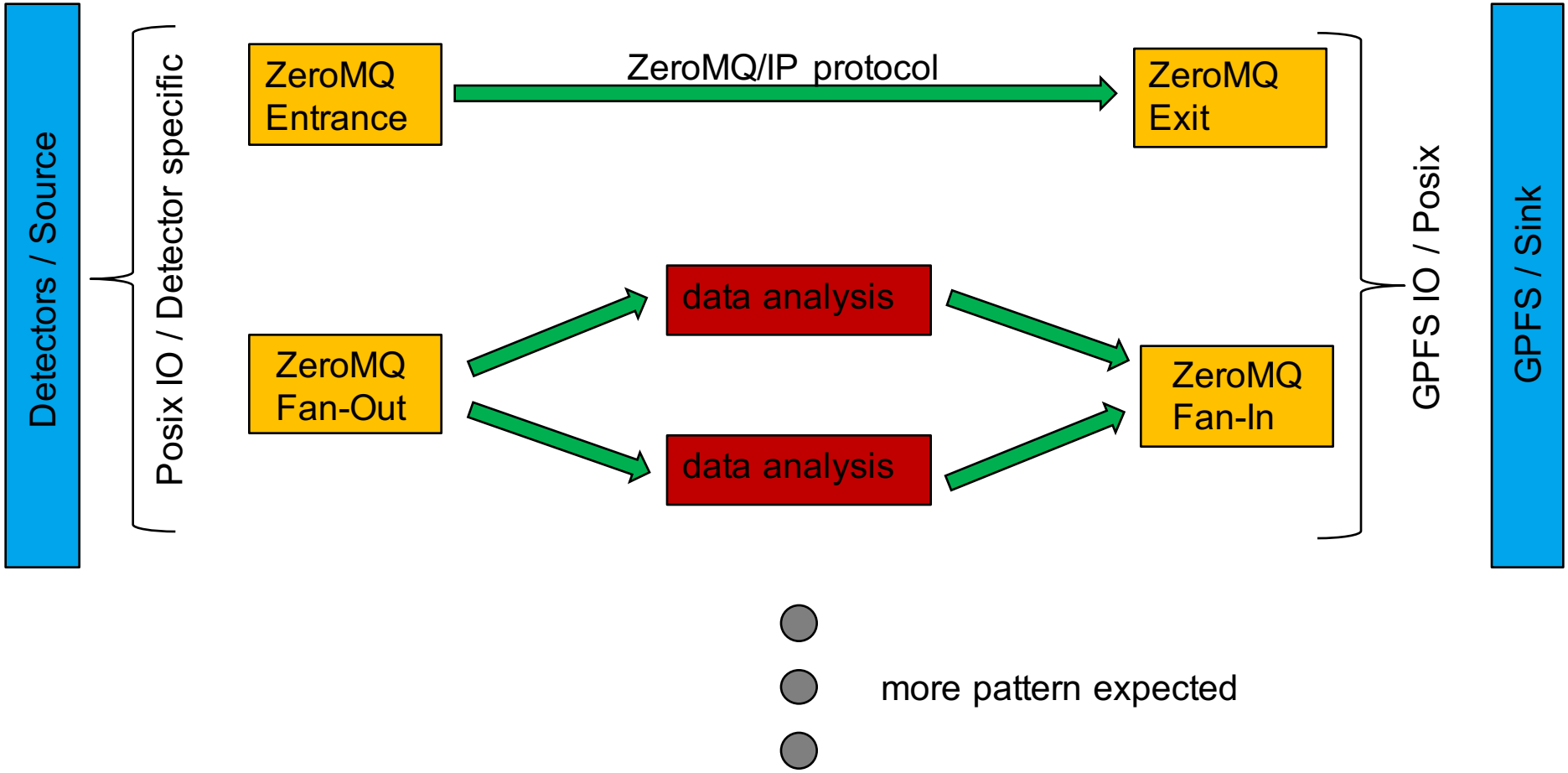


# user perspective

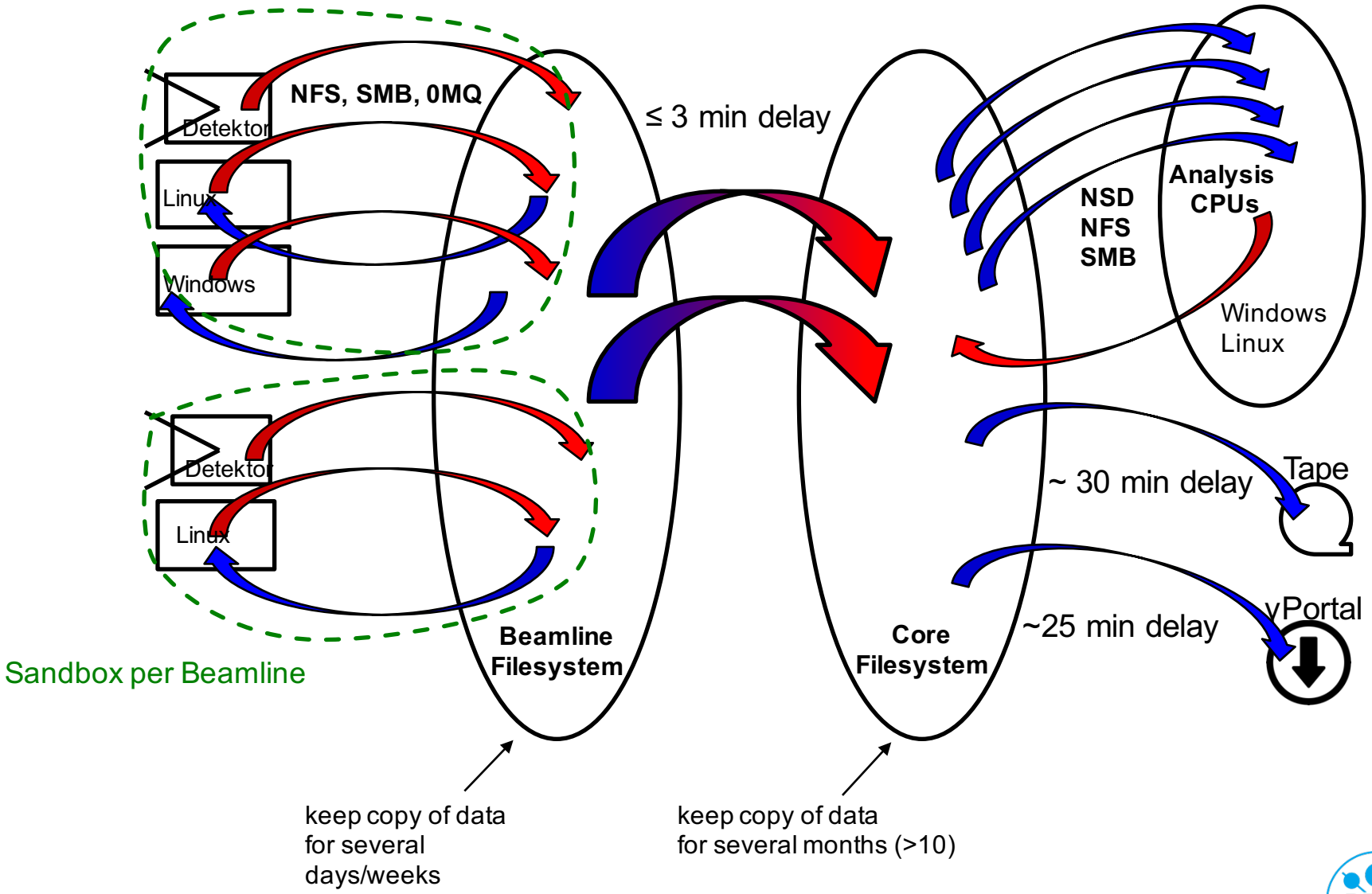
- run *startbeamtime <beamtime-id>*
  - from relational DB get [PI, participants, access rights, nodes, …]
  - create filesets, create def. dirs + ACLs, setup policy runs, quota setting, XATTR def. settings, create shares for nfs+smb, start ZeroMQ endpoints
- take data – mostly detector specific
  - mostly done through NFS & SMB today – ZeroMQ (vacuum cleaner) growing
  - non present (on site) experimentors can start accessing the data remoteley

    Portal (http), FTP, Globus-Online (also used for remote data ingest)
- run *stopbeamtime <beamtime-id>*
  - verify all files copied
  - unlink fileset (beamline fs), 'unshare' exports, stop ZeroMQ endpoint
  - update XATTR to reflect state
- users continue data analysis on core fileystem & remote access
  - GPFS native, NFS, SMB

# ZeroMQ usage



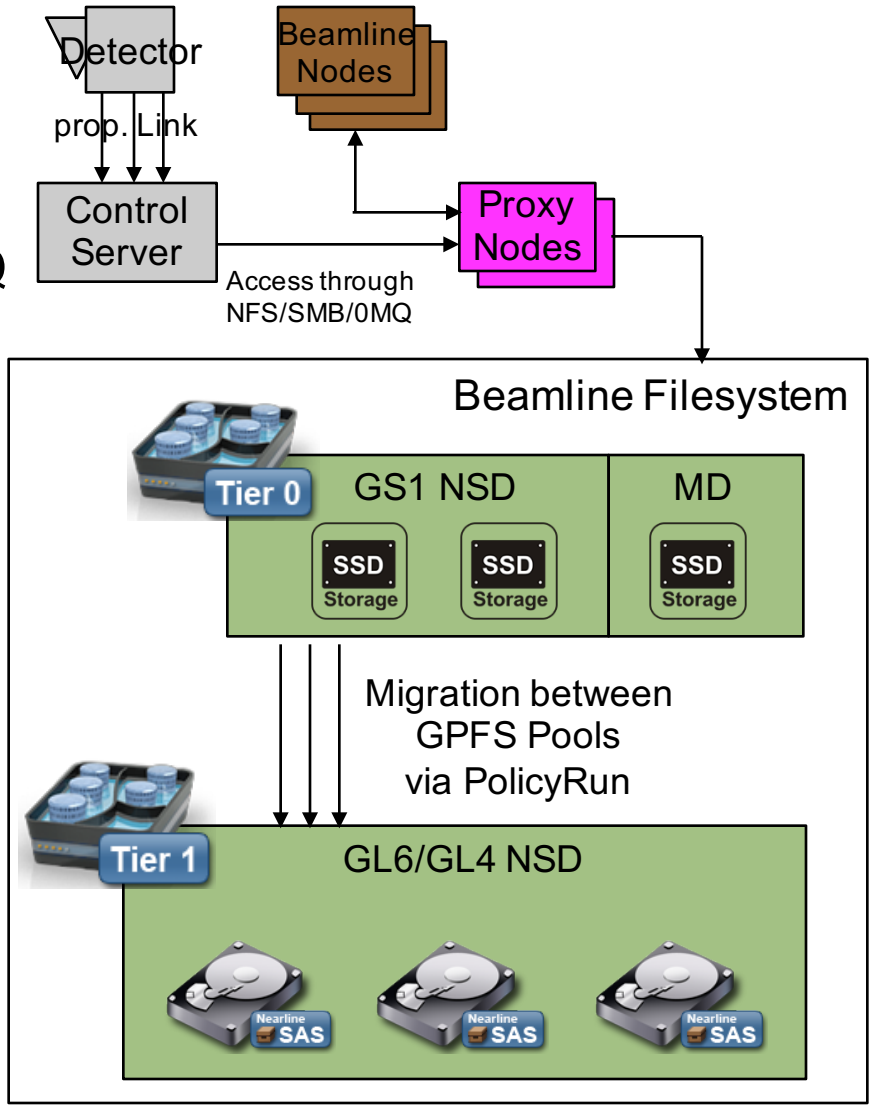
# from the cradle to the grave – the overall dataflow



# Beamline Filesystem

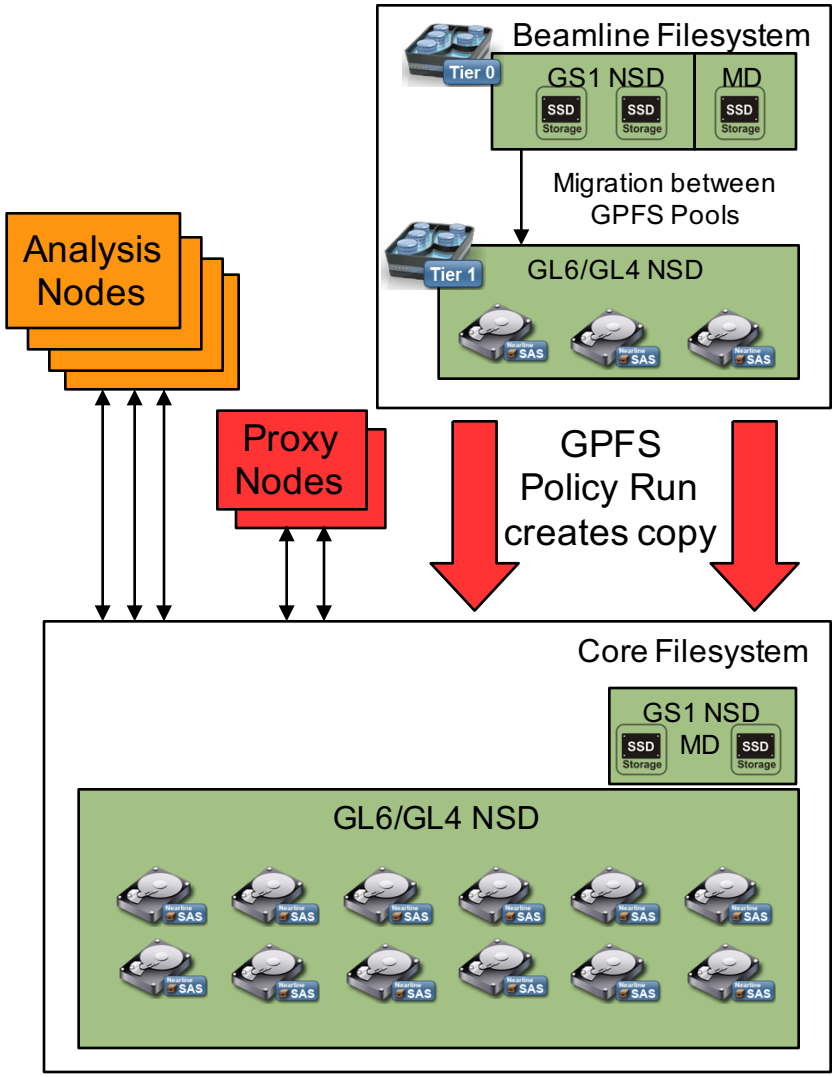
- wild-west" area for beamline
- only host based authentication, no ACLs
- access through NFSv3, SMB or ZeroMQ
- optimized for performance
  - 1 MiB filesystem blocksize
  - pre-optimized NFSv3: ~60 MB/s
  - NFSv3: ~600 MB/s
  - SMB: ~300-600 MB/s

  } after tuning !
  - 'carve out' low capacity & large # of spindles
- Tiered storage
  - Tier 0: SSDs from GS1 (~10 TB)
  - migration after short period of time
  - Tier 1: ~80 TB capacity at GL4/6

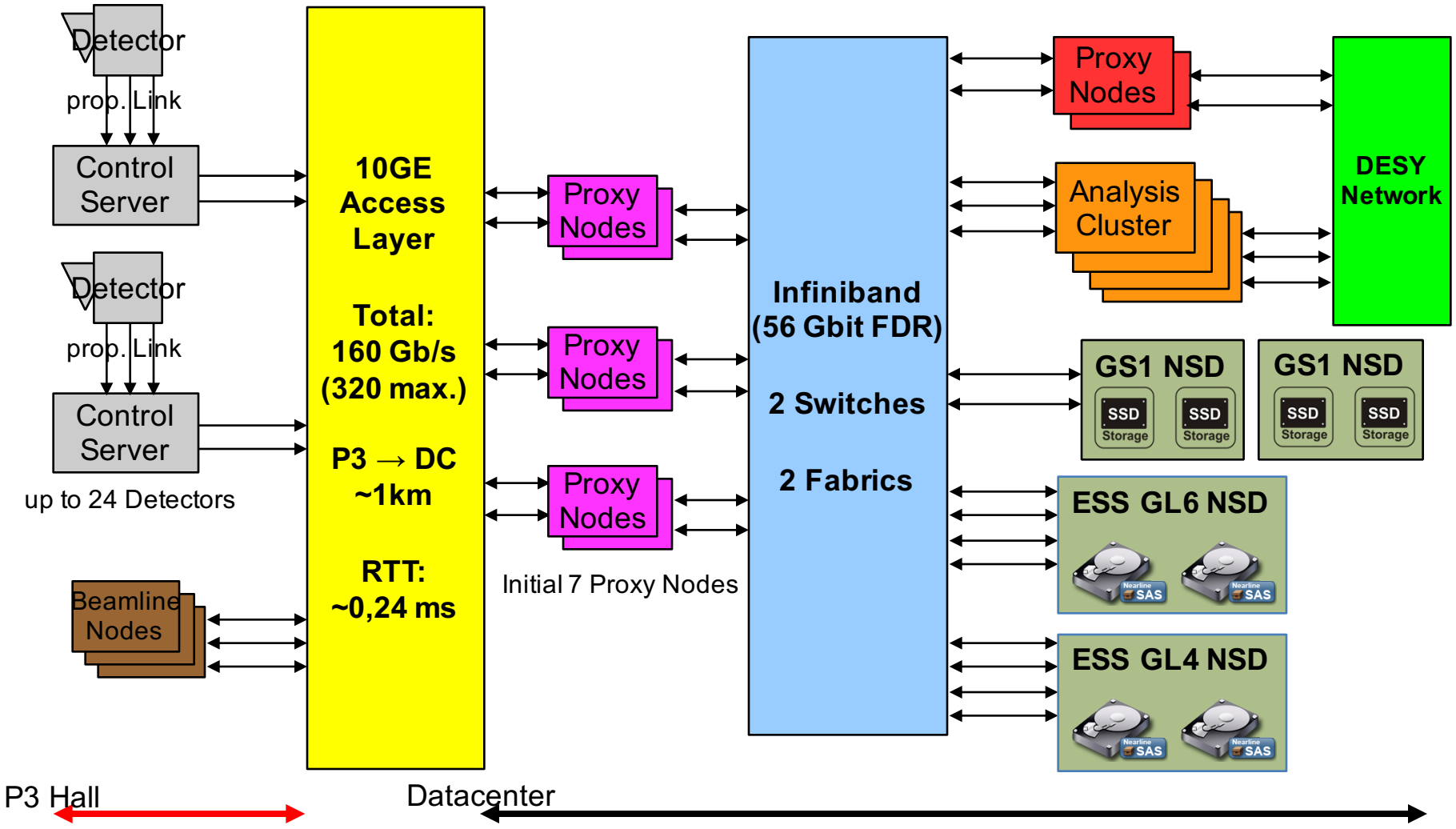


# Core Filesystem

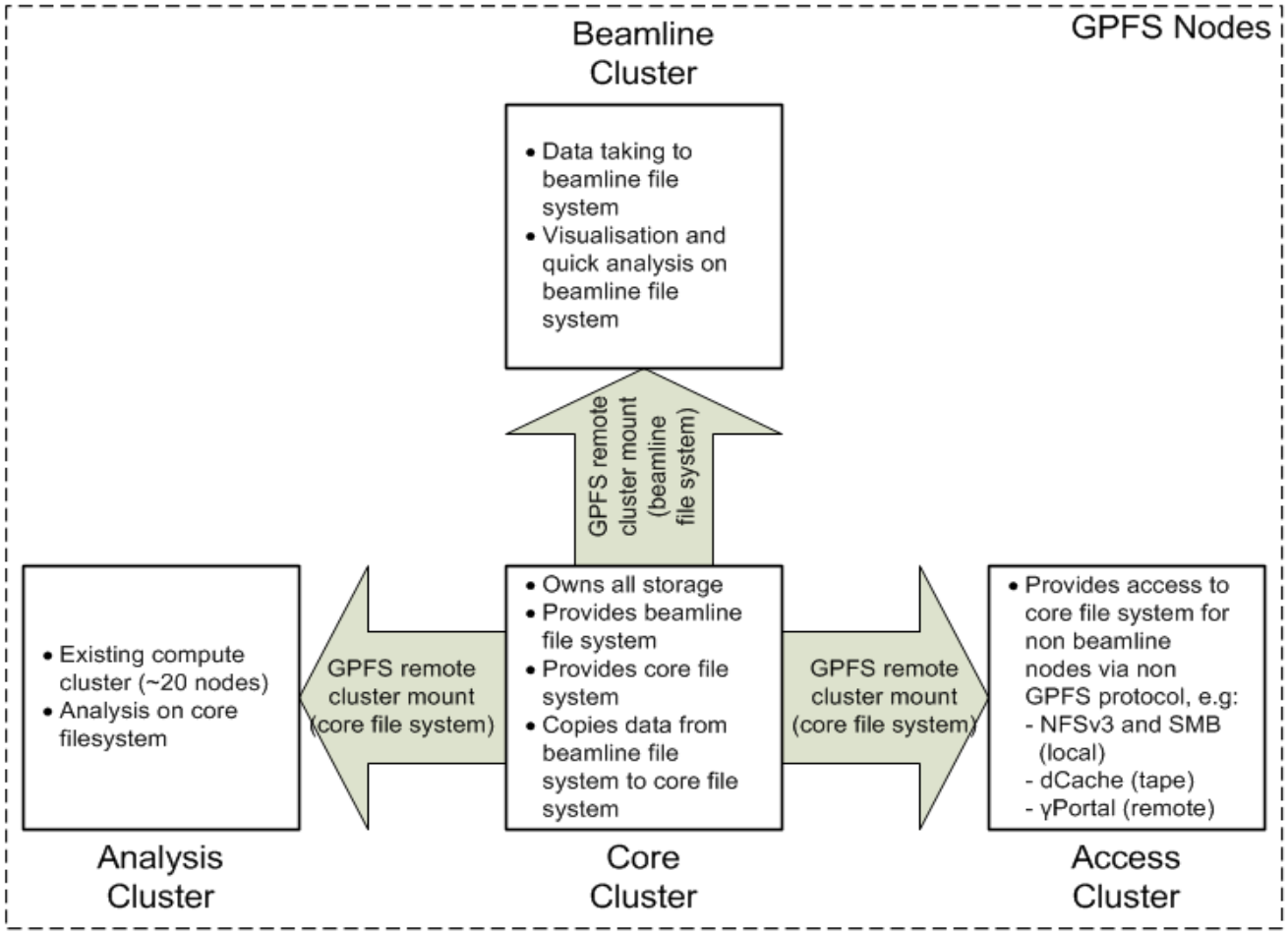
- "settled world"
- full user authentication
- NFSv4 ACLs (only)
- access through NFSv3, SMB or native GPFS (dominant)
- GPFS policy runs copy data
  - Beamline → Core Filesystem
  - Single UID/GID (many to one)
  - ACLs inherited from '.'
  - raw data set to immutable
- 8 MiB filesystem blocksize
- fileset per beamtime



# boxes linked together



# structured view




**Key Components**

- 4x GPFS clusters
  - Looks like one cluster from the user point of view
  - Provides separation for admin purposes
- 2x GPFS file system
  - Provides isolation for data taking (beamline file system) and long-term analysis (core file system)
- 2x IBM ESS GS1
  - Metadata core file system
  - Metadata beamline file system
  - Tier0 beamline file system
  - Having two ESS improves high availability for data taking
- 2x IBM ESS GL4/6
  - Tier1 beamline file system
  - Data core filesystem
- Additional nodes in access cluster and beamline cluster
- Analysis cluster already exists
- All nodes run GPFS
- All nodes are connected via InfiniBand (FDR)

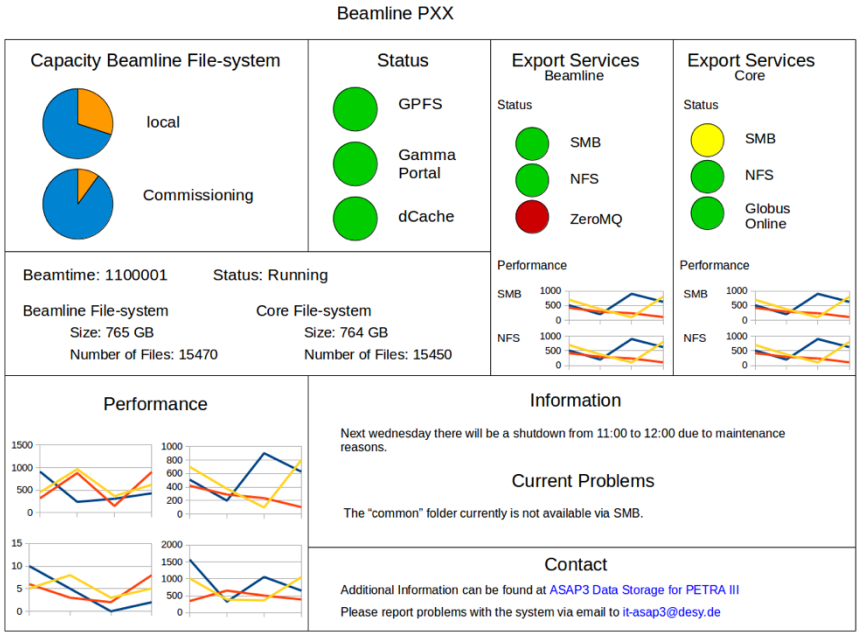
# monitoring

- data tapped from ZIMon Collector -> ElasticSearch
- ESS GUI (used supplementary to the others)
- three areas to cover
  - dashboards (view only)

    target for beamline, operating, public areas
    data from ElasticSearch presented with Kibana
  - expert

    ElasticSearch data accessed through Kibana
    – hunt for covered correlations
    logfile analysis (Kibana)
  - operating alert system

    Nagios/Icinga and RZ-Monitor (home grown) checks



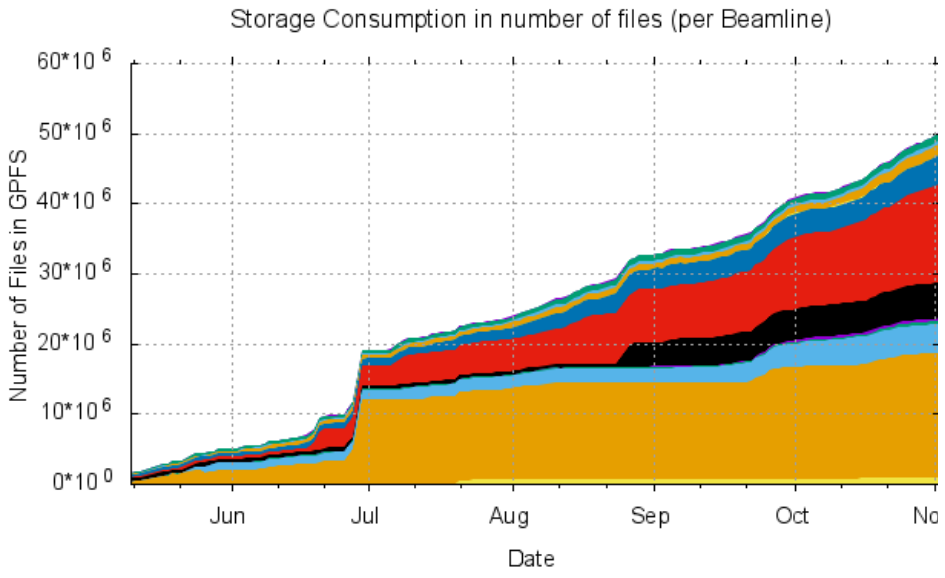
# primary ‘civil duty’ - tuning, tuning, tuning…

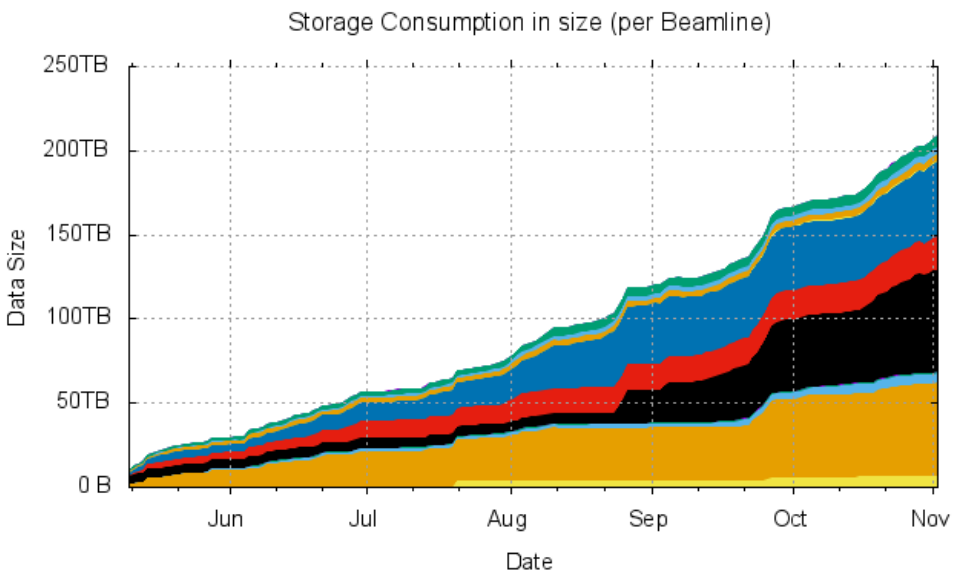
- in several tuning sessions (multiple days…), on the invest side – we got impressive speed-ups
- getting the right expert on hand (for days) is the key – know-how is there
- changes with the biggest impact are:
  - GPFS: blocksize (1M, 8M), prefetch-aggressiveness (nfs-prefetch-strategy), logbuffer-size (journal), pagepool (prefetch-percentage)
  - NFS: r/w size (max)
- continues to be an important permanent topic !
- finding the real ‘protein resource’ @IBM is possible but not always easy

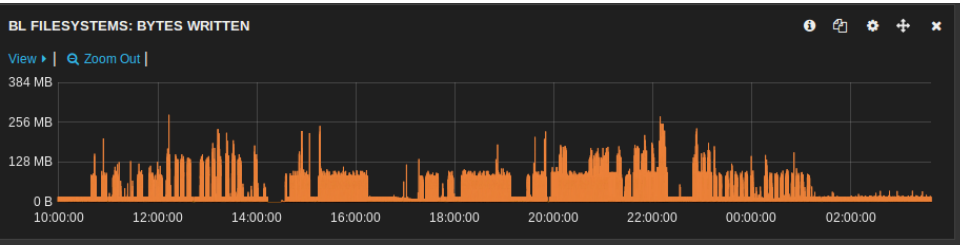
# things we hope(d) would work, but…

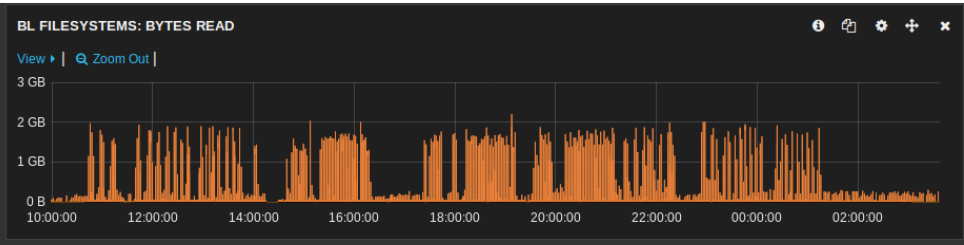
- current architecture result of process during last months
- detectors as native GPFS clients
  - old operating systems (RHEL 4, Suse 10 etc.)
  - inhomogeneous network for GPFS: InfiniBand and 10G Ethernet
- Windows as native GPFS client
  - more or less working, but source of pain
- Active file management (AFM) as copy process
  - no control during file transfer
  - not supported with native GPFS Windows client
  - cache behavior not optimal for this use case
- self-made SSD burst buffer
  - SSDs died very fast (majority), others by far too expensive (NVMe devices) but survived !
- remote cluster UID-mapping

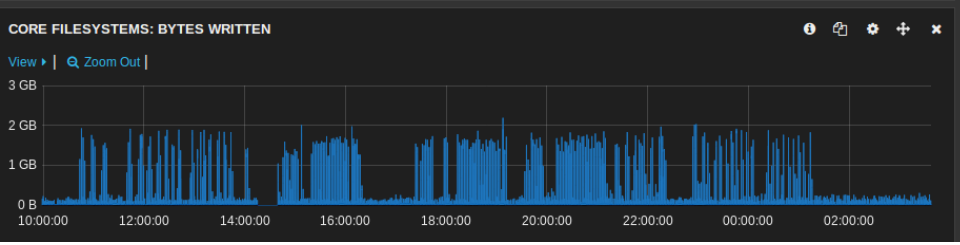
# usage

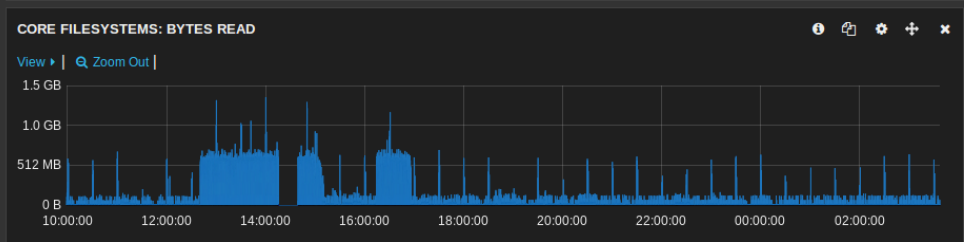

Storage Consumption in number of files (per Beamline)
Number of Files in GPFS
60*10 6
50*10 6
40*10 6
30*10 6
20*10 6
10*10 6
0*10 0
Jun
Jul
Aug
Sep
Oct
No
Date


Storage Consumption in size (per Beamline)
Data Size
250TB
200TB
150TB
100TB
50TB
0 B
Jun
Jul
Aug
Sep
Oct
Nov
Date


BL FILESYSTEMS: BYTES WRITTEN
View
Zoom Out
384 MB
256 MB
128 MB
0 B
10:00:00
12:00:00
14:00:00
16:00:00
18:00:00
20:00:00
22:00:00
00:00:00
02:00:00

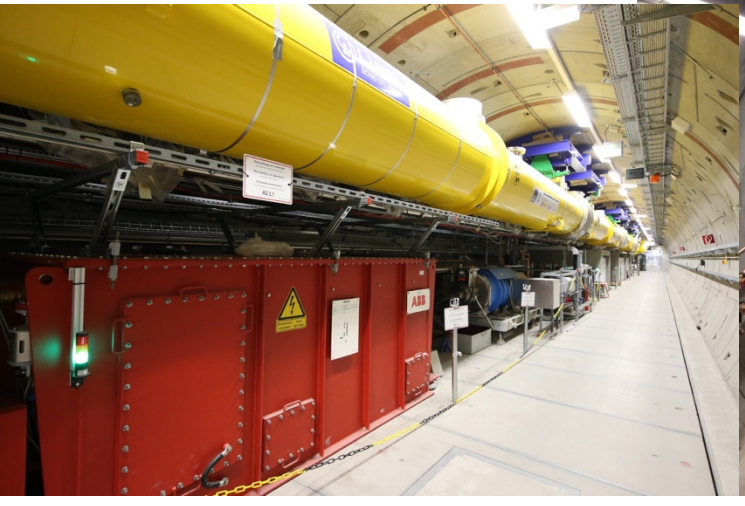

BL FILESYSTEMS: BYTES READ
View
Zoom Out
3 GB
2 GB
1 GB
0 B
10:00:00
12:00:00
14:00:00
16:00:00
18:00:00
20:00:00
22:00:00
00:00:00
02:00:00


CORE FILESYSTEMS: BYTES WRITTEN
View
Zoom Out
3 GB
2 GB
1 GB
0 B
10:00:00
12:00:00
14:00:00
16:00:00
18:00:00
20:00:00
22:00:00
00:00:00
02:00:00


CORE FILESYSTEMS: BYTES READ
View
Zoom Out
1.5 GB
1.0 GB
512 MB
0 B
10:00:00
12:00:00
14:00:00
16:00:00
18:00:00
20:00:00
22:00:00
00:00:00
02:00:00

# preparing for

- next gen. detectors @PetraIII – 6GB/sec (expect 4-6 next year)
- next 'customer' - XFEL (http://www.xfel.eu) (free electron laser)
  - start in 2016/2017
  - blueprint
  - larger - 100 PB per year
  - faster - (10+10+30GB/sec) (initial three experiments)



# summary

- key points for going with GPFS as the core component
  - creator knows business – well-hung technology – storage is conservative
  - fast & reliable & scalable (to a larger extend ;-) – GNR usage ‘a must’
  - XATTR, NFSv4 ACL (no more POSIC ACLs !)
  - federated (concurrent) access by NFS, SMB and native GPFS protocol
  - policy-runs – drives all the automatic data flow (ILC)
  - distinguished data management operations @large scale
  - integration into (our weird) environments/infrastructure (auth, dir services…)
  - future stuff – burst buffer, CES (incl. Ganesha), QOS, … continuing

    expect even higher bandwidth with Ganesha (~800MB/sec single thread)
    Ganesha running NFSv4.1 (including pNFS ;-) – our long-term agenda point