DQM

Current Status, Technical Issues and Plans

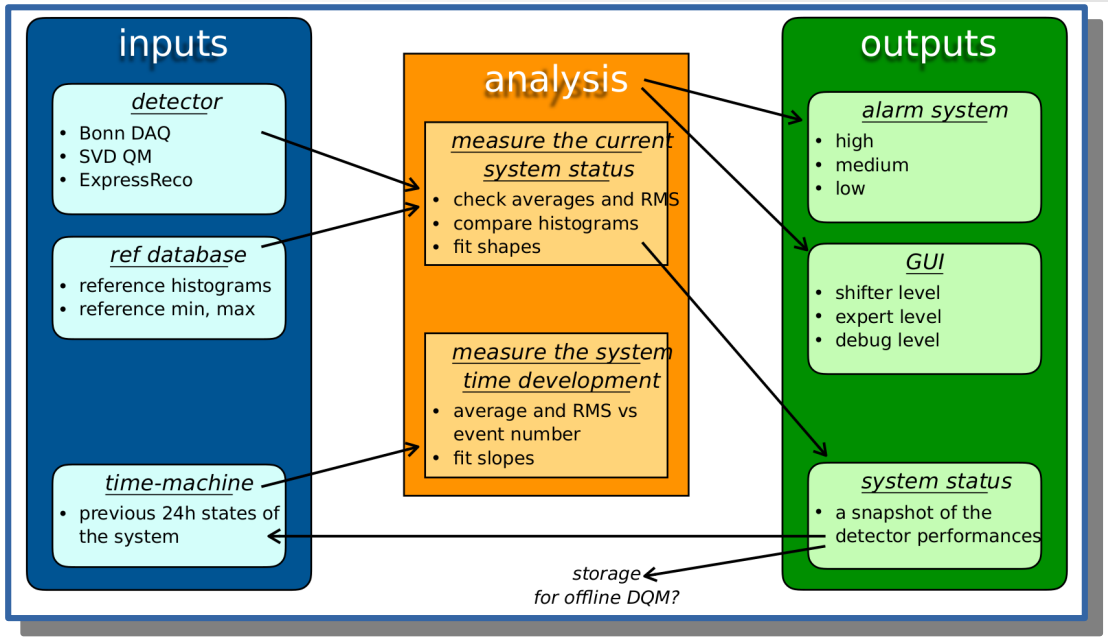
21th DEPFET workshop, Ringberg, 30.5.2017

B. Spruck for Mainz group

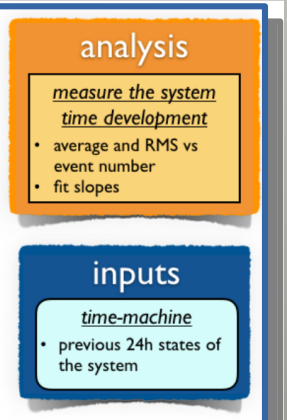
Content

- Reminder
- Status TB 2017
  - basf2 → GUI
- Wishlists
- Limits
  - Amount, Size, Frequency
- Solutions(?), Feasibility
  - Sources
  - Servers
  - GUI
- Summary

Reminder – Talk and Discussion @ B2GM



The Data Quality Analysis (2)

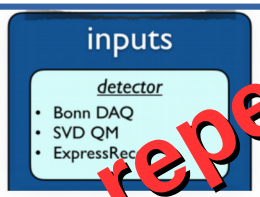


- A system that is properly working may be drifting towards a not working or problematic state:
  - stable (and properly working)
  - in development
- EPICS has the capability to provide the type of information needed in order to understand if the system is stable or not
- But, it might not be preferable to do complex fits in EPICS
- We would like a basf2 module to perform this analysis with ROOT library and other tools

Remarks about DQ Analysis:

- if possible, we would like to use the basf2 framework to perform the analysis, we need at least the ROOT libraries and tools.
- is there a dedicated machine on which it is foreseen to run the analysis?
- how do we collect data from ExpressReco, BonnDAQ and SVD QM on this machine and merge them in basf2?

Detector Observables



- PXD and SVD detector experts have provided a list of observables a the needed granularity (see backup slides for details)
- These are the observables that the expert shifter should look at to debug a problem on the detector

SVD (from SVD QM)

- Raw noise
- Noise
- Noise programmed
- Pedestal
- Pedestal programmed
- Hit Time
- S/N
- S/N peak value
- Occupancy
- MP occupancy
- Cluster size
- Average cluster size
- Hit maps
- APV25Error
- FIR filter quality
- APV data
- Event size
- APV trigger buffer occupancy

1870 1D histograms

PXD (from Bonn DAQ)

- Seed charge
- Cluster charge
- Cluster size
- Common mode
- Number of pixels fired per frame
- Occupancy
- Hit maps
- Noise

440 1D histograms (22 per layer)
200 2D histograms (10 per layer)

- First estimate of the amount of histograms we will have to deal with
  - ExpressReco histograms are not included in the list (PXDDigits, PXDCluster, SVDDigits, SVDClusters, pattern recognition, track fit, ROI finding, ...)

Question Time

- How much storage and CPU is available on ExpressReco?
- Is there a dedicated machine on which it is foreseen to run the DQ analysis?
- Where do we merge data from BonnDAQ, SVD QM and ExpressReco?
- Can we use EPICS as references database?
- Where do we store the system status for the time-development analysis and offline DQM?
- How do we interface basf2, EPICS, NSM2 and CSS?
- Offline DQM?

Some answers

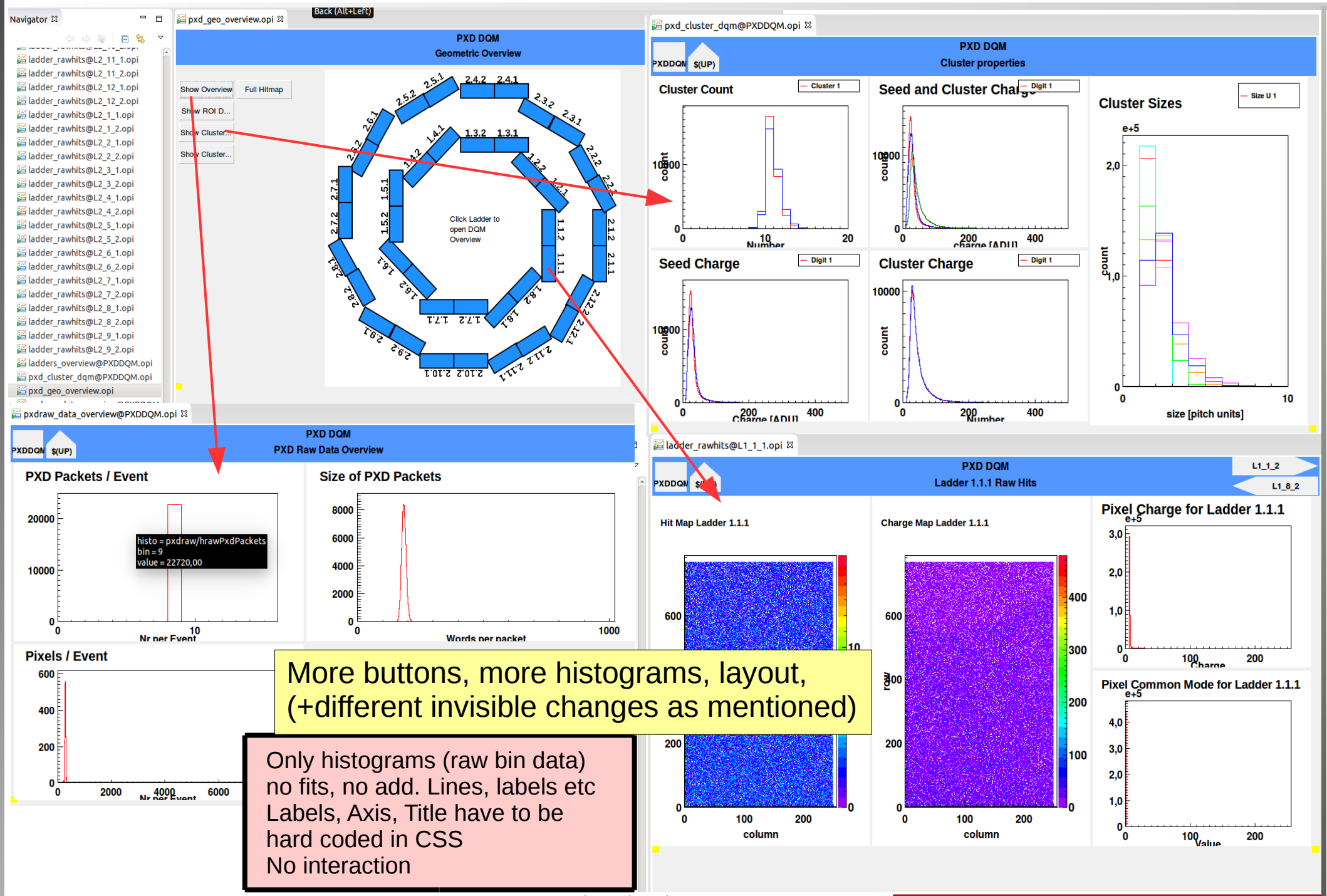
Giulia Casarosa 26th B2GM ~ DQM 13

...will try not to repeat too many things

Status TB 2017

- No technical "big" difference to TB 2016
  - 2→ 4 sensors (no big change)
  - More/other histograms
  - VXD level: PXD + SVD layers → VXD layers and correlation
    - → added to "VXD" GUI panels
  - More PXD specific histograms
  - Included some tracking histograms as well (which one?)
- Problems:
  - GUI did not show any- or everything
    - basf2 ExpReco (and HLT) scripts were changed during TB
    - Histograms and their names changed before and during TB, typo in histogram names → ongoing changes in GUI, was not updated on all accounts/computers
    - DAQ histogram server crashed silently
- Feedback?

“old” OPIs (from 2016 TB) → TB 2017

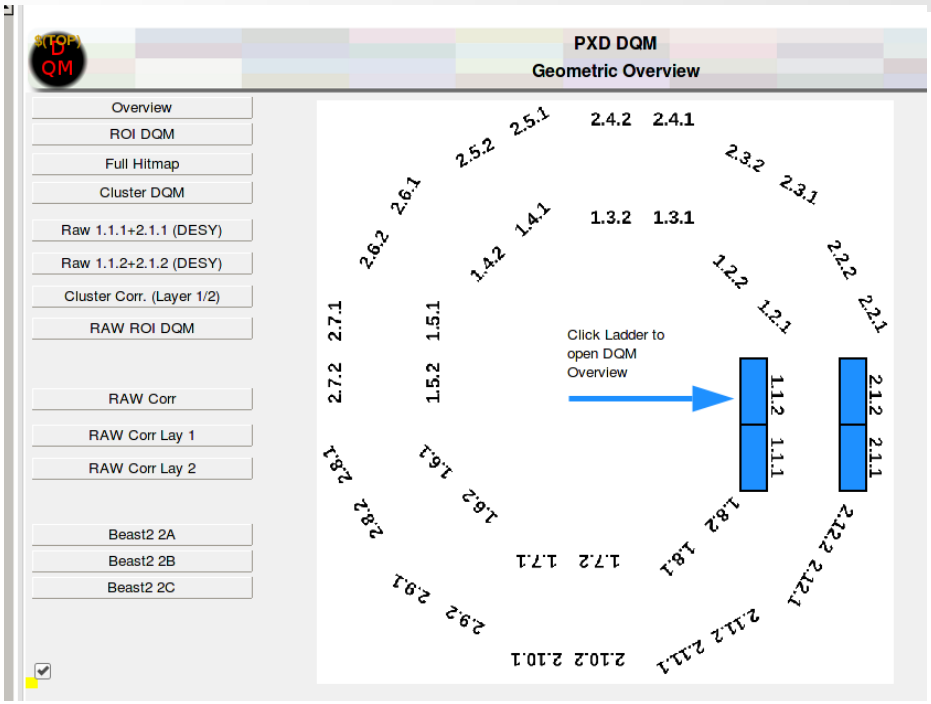


More buttons, more histograms, layout, (+different invisible changes as mentioned)

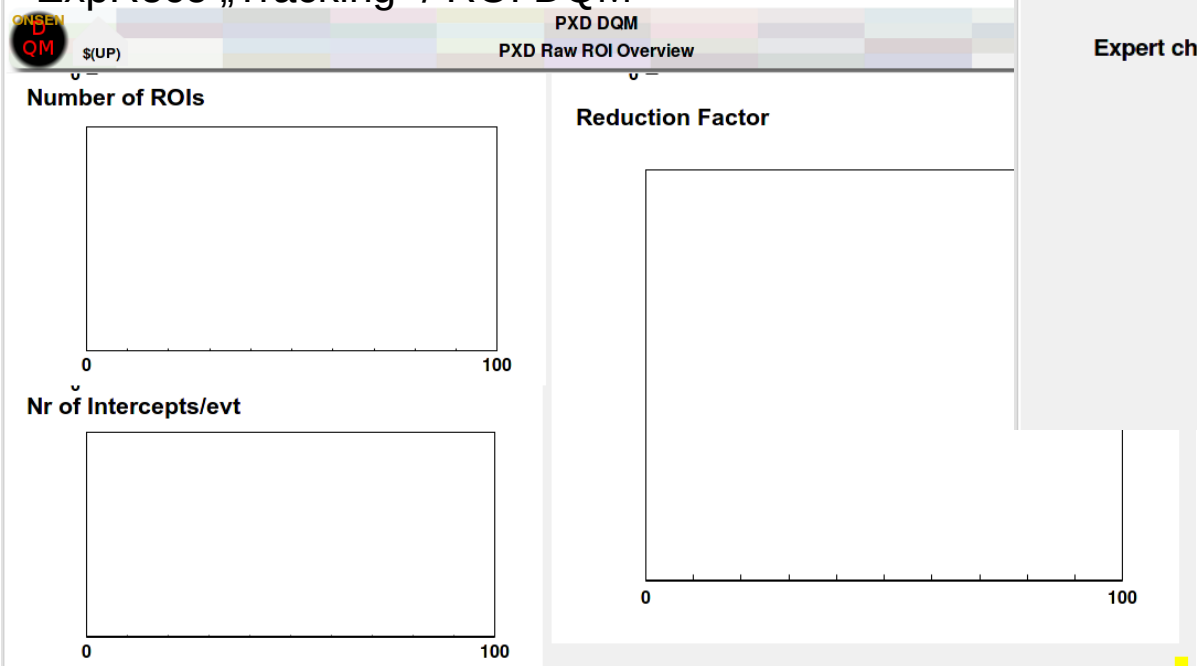
Only histograms (raw bin data)
no fits, no add. Lines, labels etc
Labels, Axis, Title have to be hard coded in CSS
No interaction

Some Selected Changes

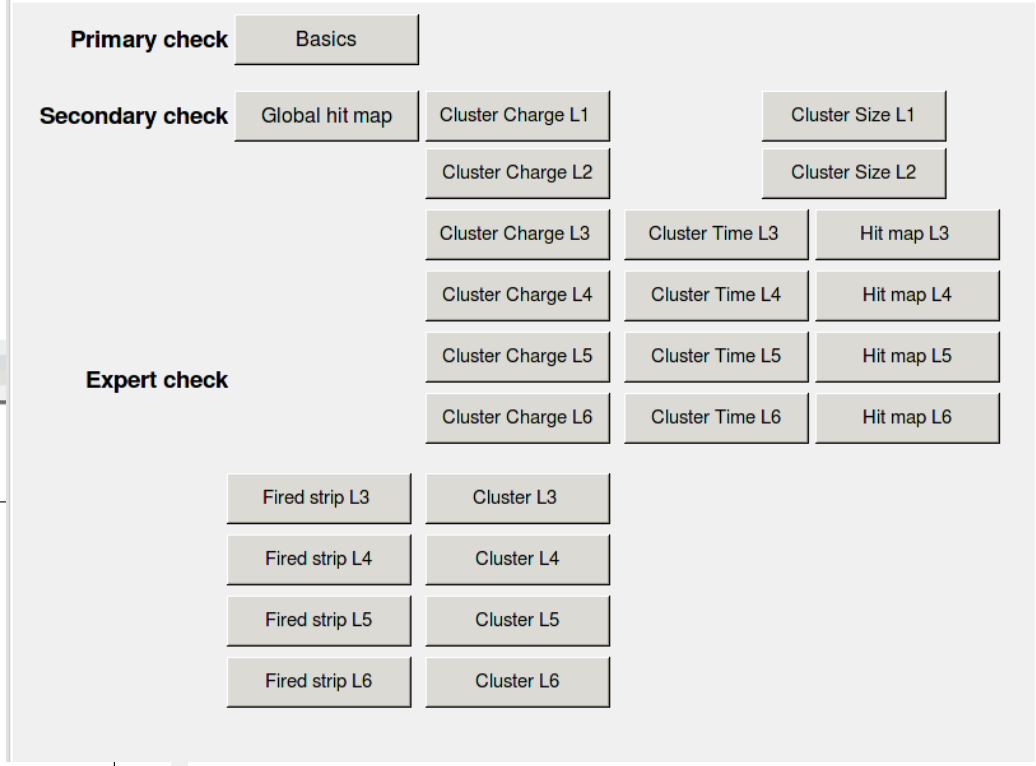
More buttons to open sub-panels



ExpReco „Tracking“ / ROI DQM



Added Layer 1+2 (PXD) to VXD/SVD Panel



Requested DQM Histograms

- "High" Level
  - Meaningful coordinates for shifter: theta/phi
  - Occupancy, VXD layer Correlations, cluster charge, etc
  - Already implemented in DQM module(s) (→ ExpReco)
- "Low" Level
  - Meaningful coordinates only for expert: sensor, switcher, DCD, local u/v
  - pixel properties (not cluster)
- "Dataflow" Level
  - No need for data analysis → SC monitoring!
  - Mostly no histograms, but trend plots
  - Example: Sensor occupancy ≙ mean data size = data rate / trigger rate
  - Compare SC and Analysis trends?
- … plus something in between
  - Trend plots/histograms depending on mean (or fit) values from other histograms

Requested Features

- High refresh rate for some of these
- Compare to reference plots, history
- Fit, shape comparison
- Histogram archive
- “freeze”
- Histograms representing fit values from (a lot of) other histograms
- “on demand” histograms (→ better on demand analysis?)

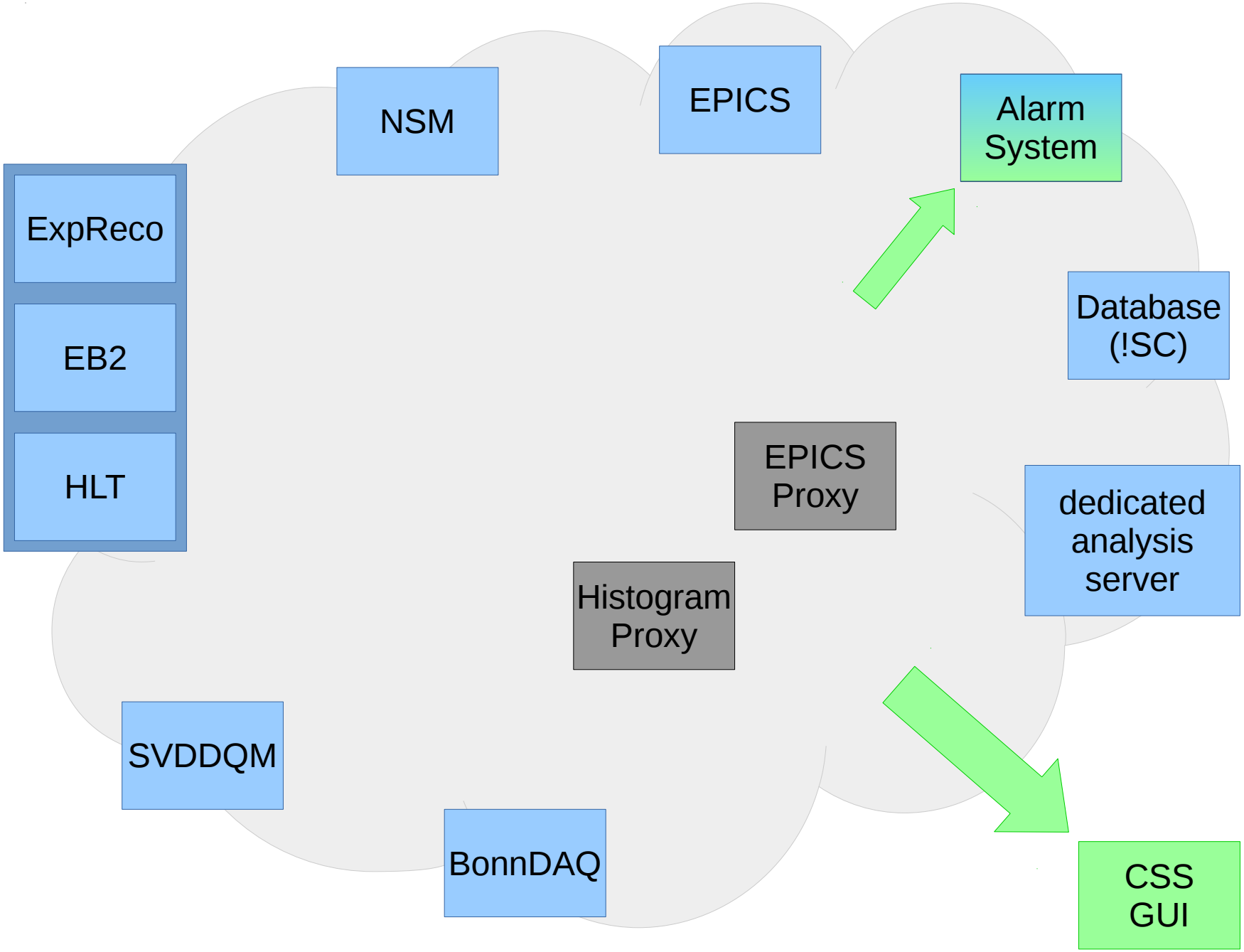
Issues

- ExpReco DQM in basf2 impose limits on
  - #histograms
  - histogram size
  - update rate
  - “on demand” histograms
  - (fit values export to EPICS)
- Actual numbers → B2GM
- Alternatives
  - BonnDAQ
  - PXD “private” DQM on separate machine (with basf2)
    - poll events from event server (rate?)

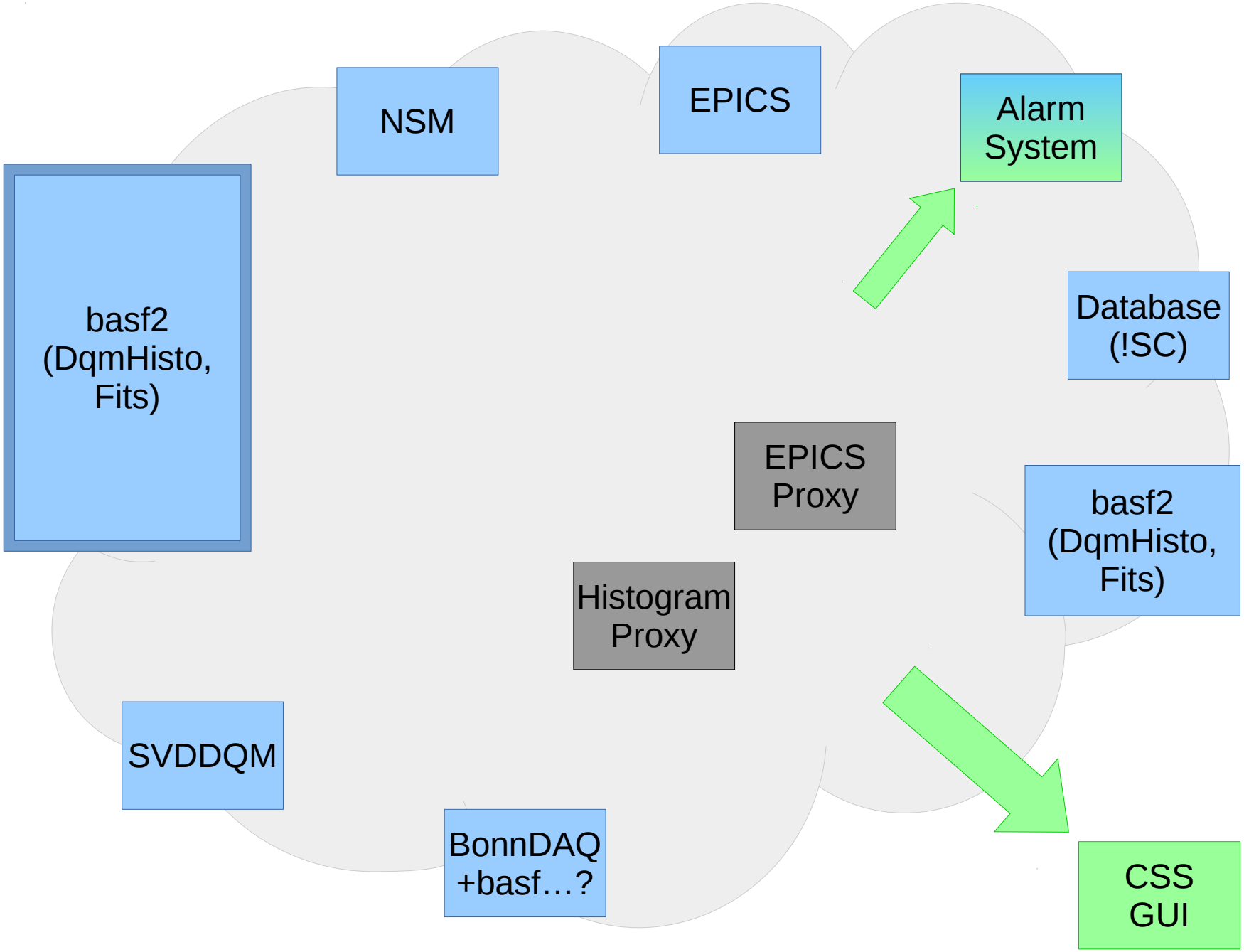
Refresh rates

- HLT will randomize events within 0.1s-3s (?) interval, does not make sense to use a higher rate on histograms from HLT.
- ExpReco adds additional delay
- Histogram server (proxy) adds delay
- JSROOT → on client (reload interval), but doesn't make sense to update more often than what analysis can provide.
- Update rates ~5s-30s seems reasonable.
- BonnDAQ gets event in correct order directly from DHH
    - Time structures may only be visible with high refresh rate
- SC trend plots depending on 'SCAN' (rate) parameter, limited by IOC
- Rate vs #events processed → statistics

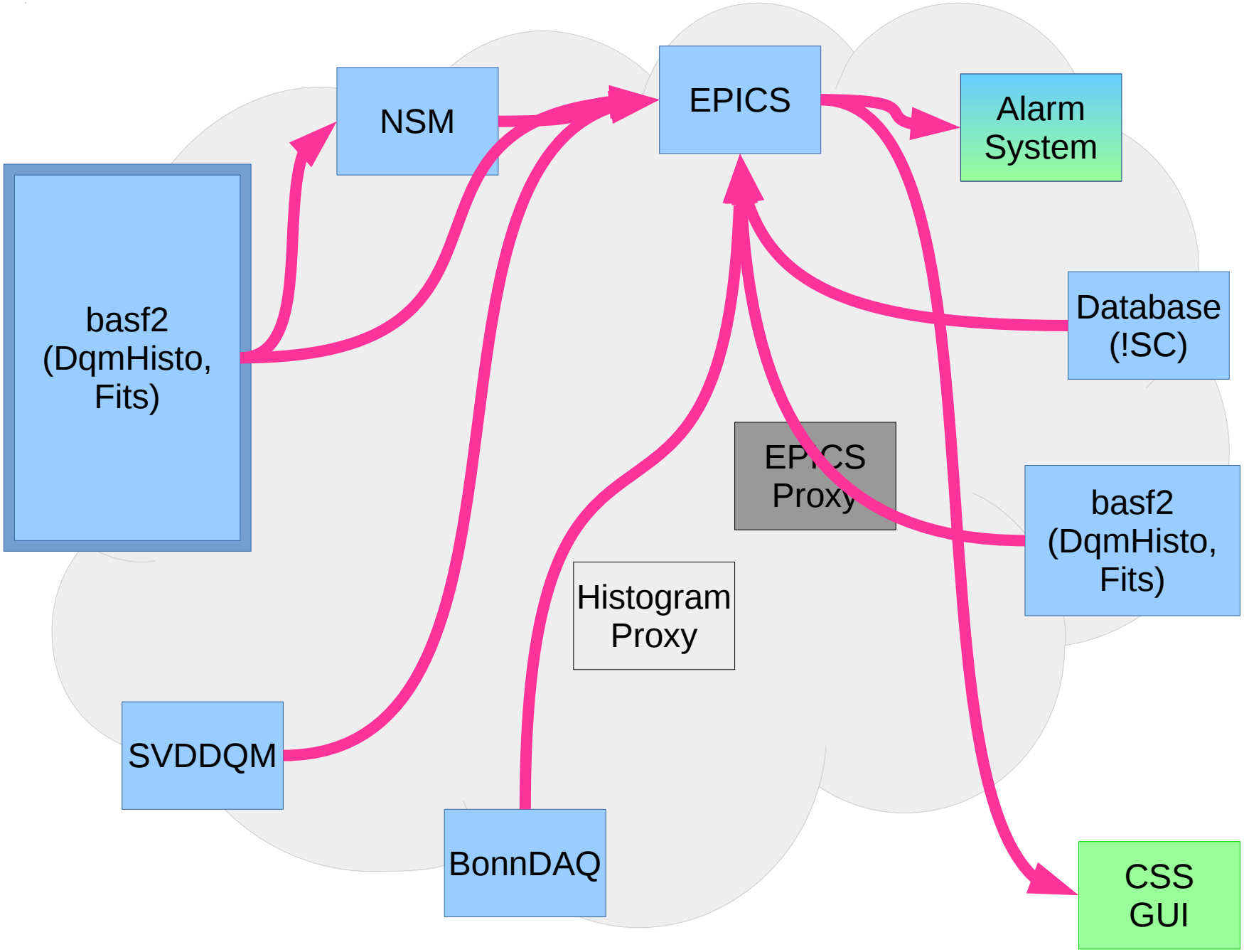
DQM – Involved System



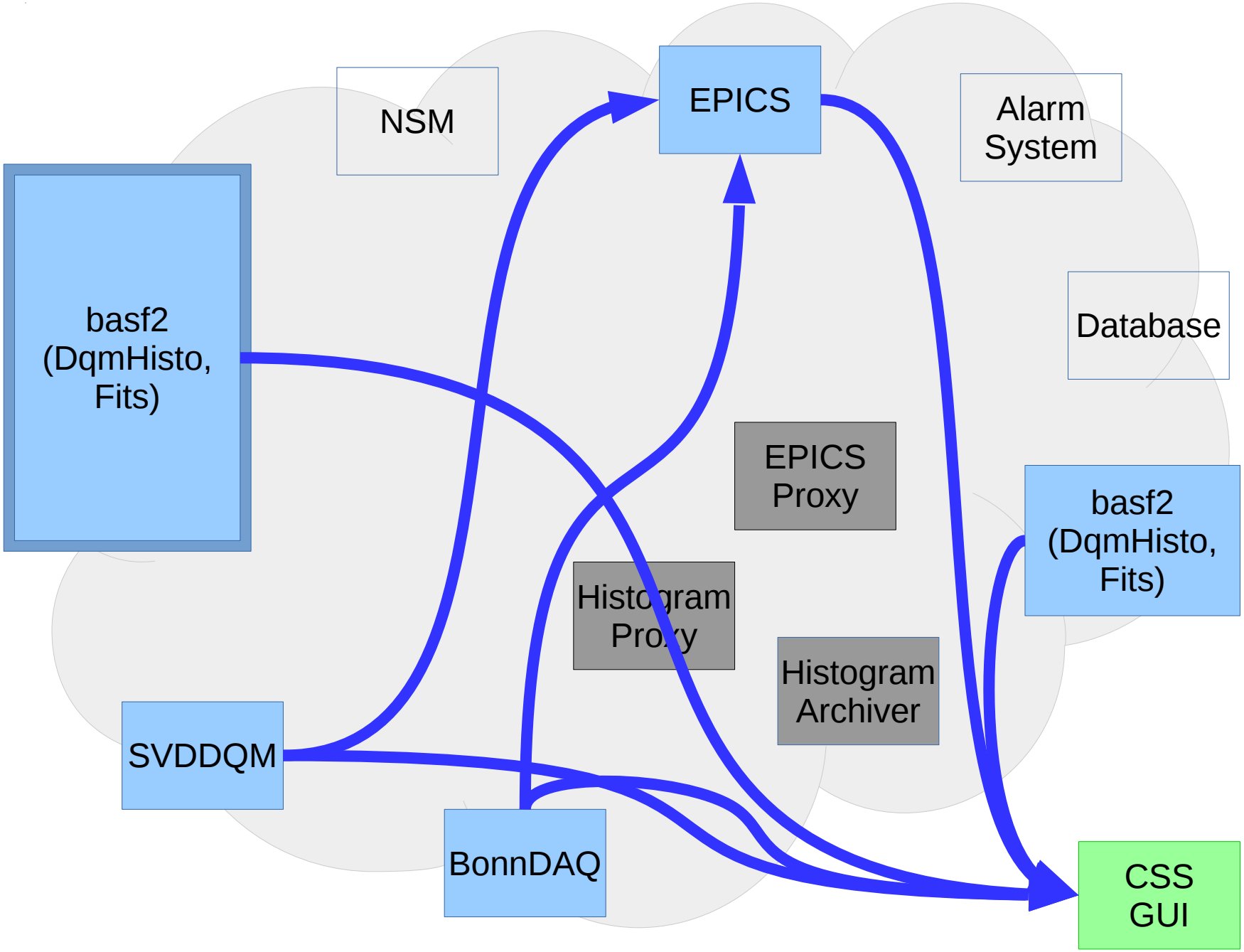
DQM – Involved System



Values (PVs)


NSM
EPICS
Alarm System
basf2 (DqmHisto, Fits)
Database (!SC)
EPICS Proxy
basf2 (DqmHisto, Fits)
Histogram Proxy
SVDDQM
BonnDAQ
CSS GUI

Histograms


NSM
EPICS
Alarm System
basf2 (DqmHisto, Fits)
Database
EPICS Proxy
basf2 (DqmHisto, Fits)
Histogram Proxy
Histogram Archiver
SVDDQM
BonnDAQ
CSS GUI

Sources

- HLT
  - ROIs (extrapolations to PXD)
  - SVD and others, no PXD
  - Latency = ~0.1s + histogram/proxy update intervall
  - very strong limits on histograms
- ExpReco
  - PXD (selected data in ROIs)
    - signal/background biased, DQM mainly after clustering
  - SVD, VXD, <u>correlations</u>
  - Tracking, residuals, ROI efficiency
  - Latency = ExpReco O(s) + histogram/proxy update intervall
  - serves all detectors, strong limits on histograms

Sources

- BonnDAQ
  - lowest level raw data, unselected
  - no correlations
  - Latency = „none“ + histogram/proxy update intervall
  - Highest update frequency (in evts/s)?
- SVDDQM
  - lowest level raw data, bypassing DAQ chain
  - no correlations
  - Latency = „none“ + histogram/proxy update intervall
- „separate analysis chain“
  - → get full events from ExpReco, like event display, but delayed!
  - very limited rate (in evts/s), but full control over histograms and update rate
  - Could do “any” analysis

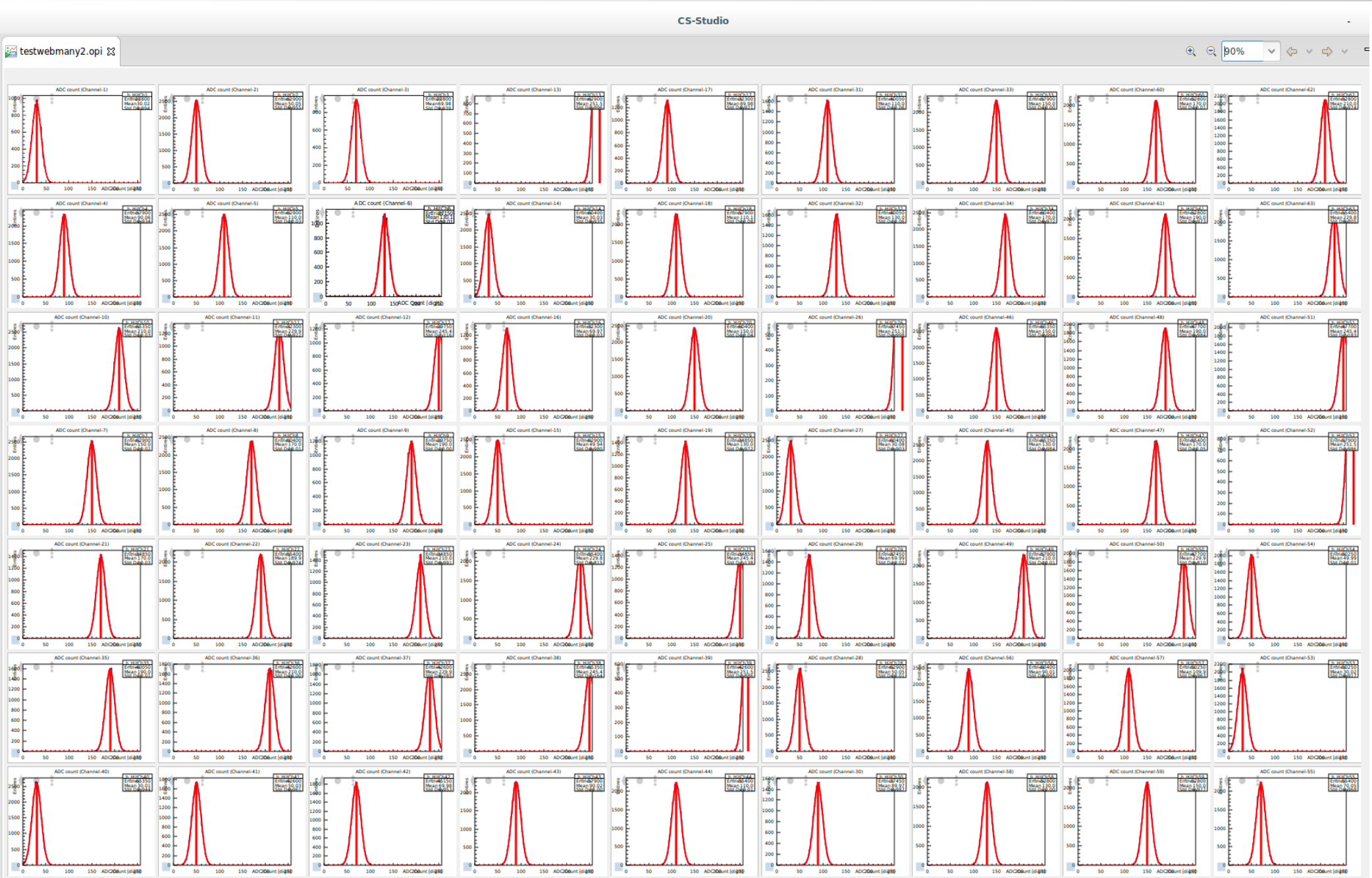
Sources

- SlowControl (DAQ Monitoring)
  - DHE: Data/Trigger rate → occupancy per sensor
    - (maybe even per DHP)
  - ONSEN: Data Out/In → reduction factor
    - (per 5 sensors load balanced)
  - Latency = SCAN intervall

Display in CSS

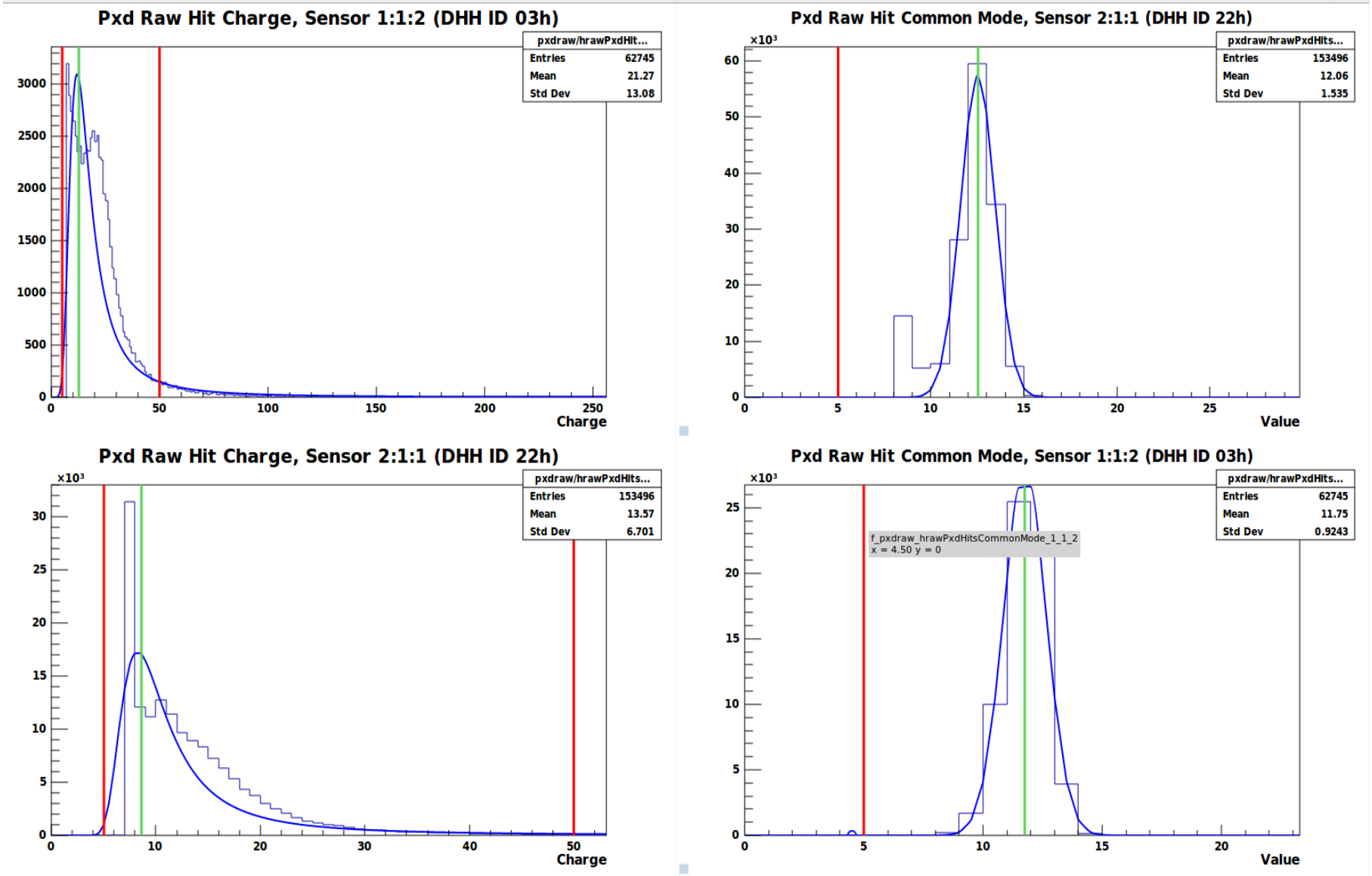
- Until TB 2017, we used DAQ home-brewed solution to display histograms in CSS, using some histogram proxy server
  - Supports only „raw“ histograms, no graphs, fit functions etc
- JSROOT (as part of root) has much better support for any kind of root objects. Using root's internal http server.
  - Display complete histogram+fit+values/labels/guidance lines etc
- Plugs seamless into existing histogram and fitting server scheme
  - One additional server task needed.
  - Full advantage only if fitting and drawing is done in DQM modules
- DAQ crew is considering it. → B2GM discussion
- Roll-out at PERSY? Needs updates of software on ExpReco (…)
  - Run on other machine as workaround?

CSS/JSROOT Performance



63 instances (out of 2000 fitted histograms on server

Example with TB Data



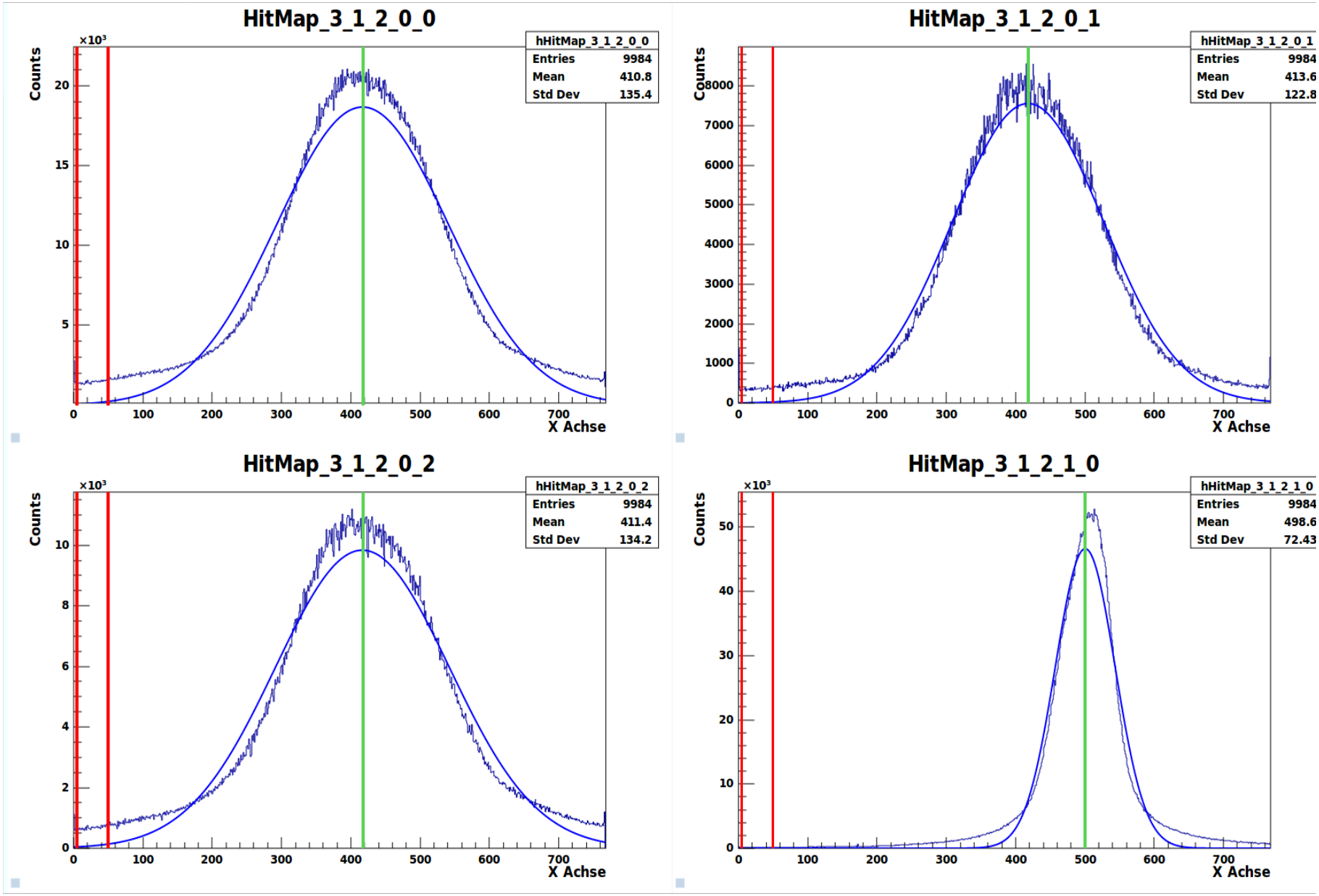
(blue/green are fit result, red are „guide“ lines (=limits)

Combining

- Can we import from BonnDAQ and SVDDQM to the histogram analysis chain of basf2?
  - → Yes
  - EPICS PV → ROOT Histogram importer module

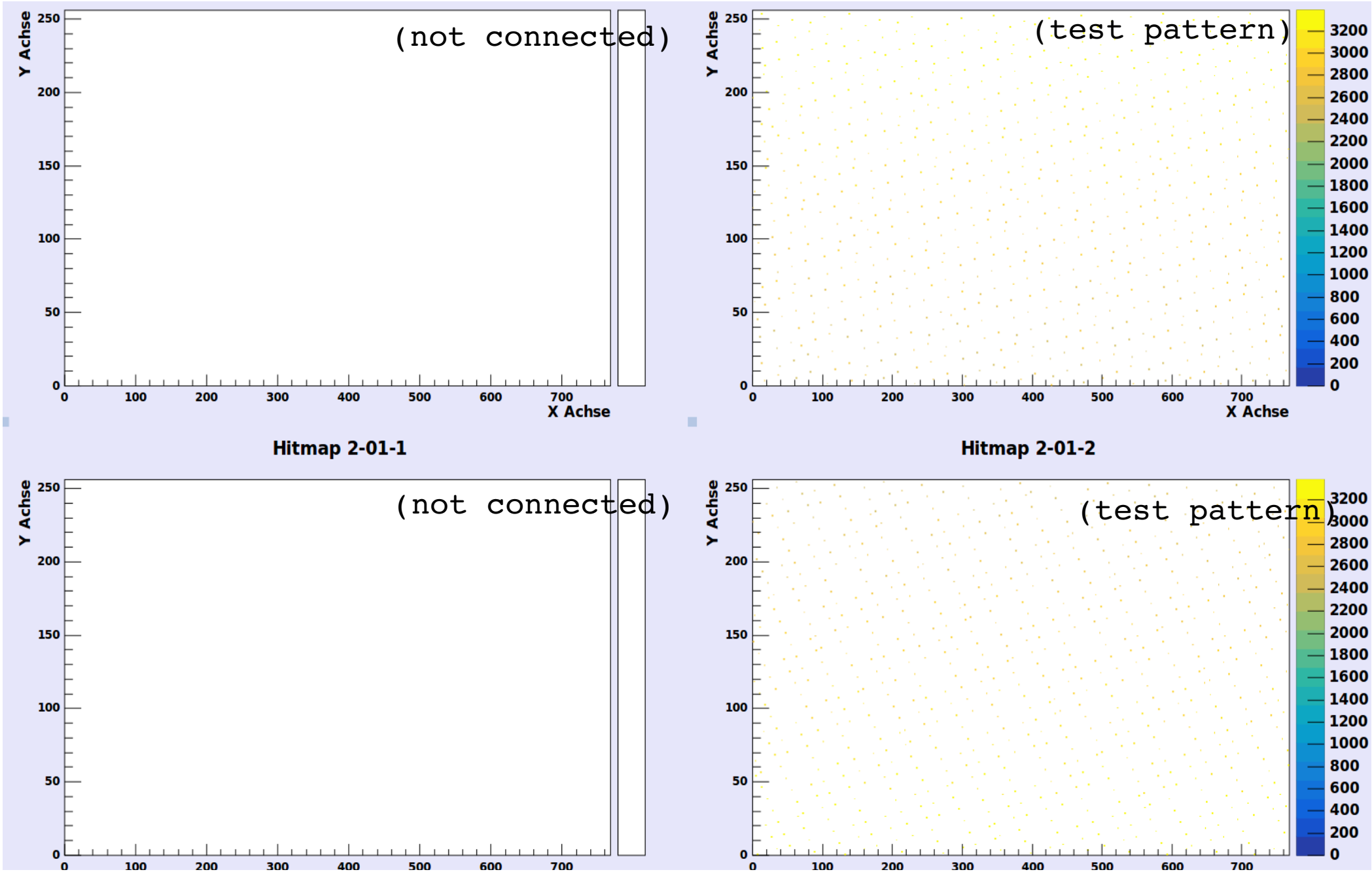
Import Histograms from EPICS PVs (SVDDQM)

Replay of data file from last TB setup. Importing 1800 histograms from EPICS to basf2.
(Quite useless to fit a gauss to here, just to prove the principle)



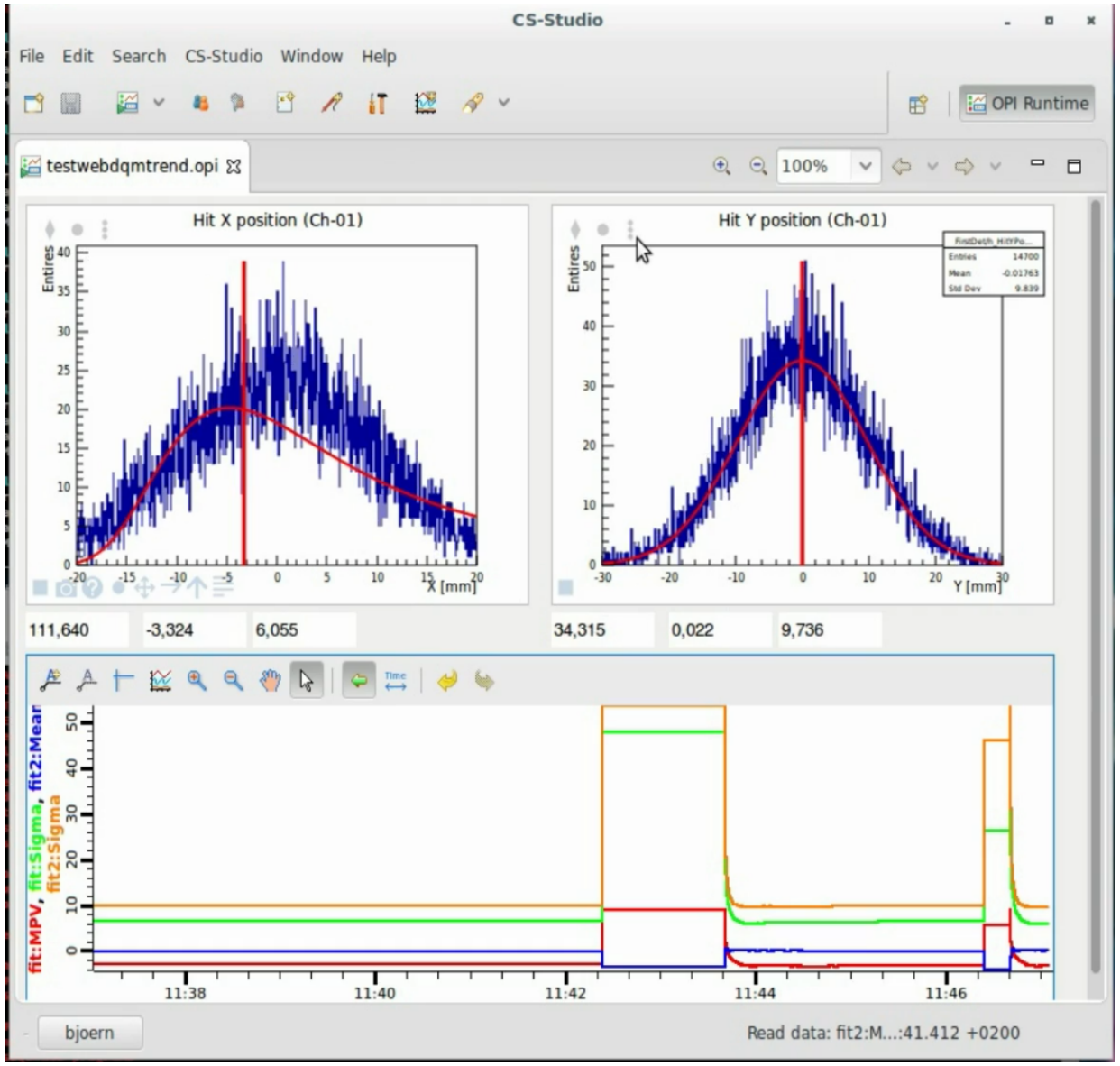
Import Histograms from EPICS PVs (BonnDAQ)

Replay of data file from last TB setup.



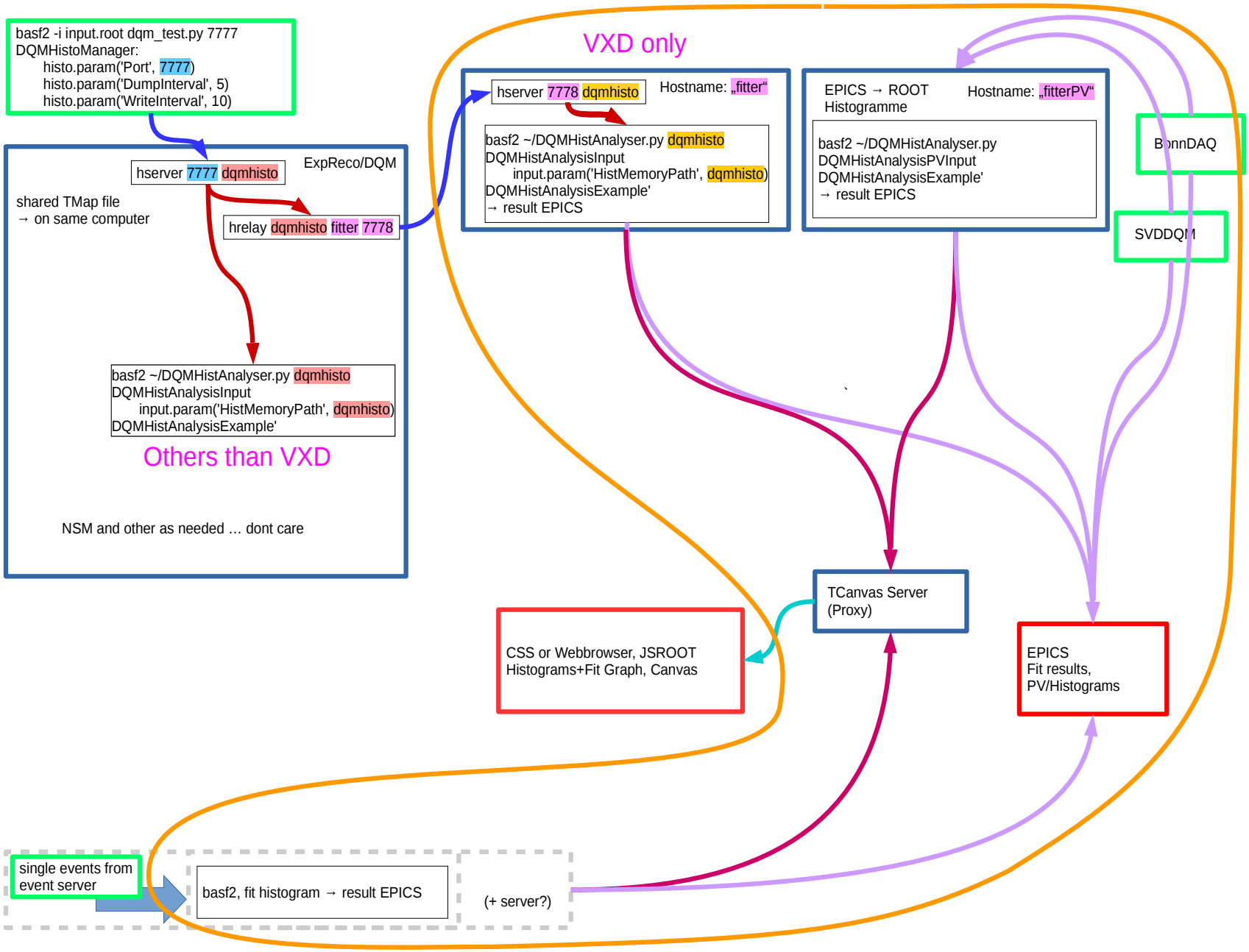
Getting Fit Results → EPICS

- Exporting fit result directly into EPICS PVs (without intermediate NSM step)
- (same as with NSM)
- History of PVs



Simplified Flow – Suggestion

Possible data flow if VXD stays independent (reasons: EPICS, Bonndaq etc)



History, Histogram Archiver

- Easy way to access reference histograms (from last hour/day/week).
- Needs: (just brainstorming)
  - Dedicated Server?
  - Format: Images (png), PV, root, JSON, … ?
  - Access: From CSS?
    - → Link to web server which serves images („simple“)
  - Backup to KEKCC? (is it needed?)
  - Join forces with basf2 software verification/regression testing suite?
  - How about other detectors?
- But: Each extra system needs administration!

More Things

- Some histograms require to reverse the mapping to get DCD/Switcher for actual pixel
  - border problem for clusters
- Limits from (EPICS) database ↔ ExpReco
  - EPICS not supported on ExpReco
  - Fit values in EPICS: only by NSM and gateway
  - → convince people in charge OR use independent machine for fits.
- "Fit" technically the same as compare with reference histogram
  - "Fit result" in EPICS → Good/bad flag for Alarm System
- How to access to reference histograms from analysis chain?

Analysis Example

- Histograms depending on DCD/Switcher combination
  - 40*6*4=960 histograms per type
  - Fit + fill 40 new 2d histogram with fit results
  - → Check with new histogram/canvas server → O.k.
  - (sorry, no screen shot)
- Guess: Not on ExpReco (at least not for 40 sensors)
- Using mean instead of fit → reduce to 80 histograms

Summary / Discussion

- B2GM discussions → JSROOT? ExpReco limits? EPICS?
  - JSROOT – Better display of fits, values …
- Separate analysis server will allow for a more or bigger histograms (but lower event rate)
- basf2 modules prepared for Simulation and test beams
  - Lot of the requested histograms already there
  - → Report of Peter Kodys
- Import BonnDAQ/SVDDQM PV histograms into basf2 demonstrated
  - → Discuss for which this is needed
- Demonstrated that complex histogram analysis can be done in histogram analysis modules in basf2
  - Creation of new histograms has not been foreseen at this step
- How to add to PERSY setup? Depends on KEK support.

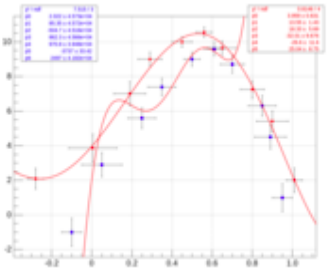
Backup

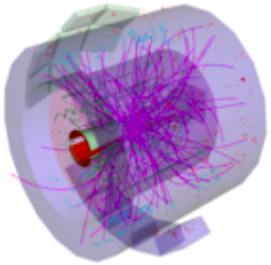
Alternate Solution: JSROOT

- Part of ROOT.

  http://jsroot.gsi.de/

  - Server and Javascript client are part of ROOT. Active development (S. Linev @ GSI)
- Can serve and display (any?) ROOT object
  - Even event display!
  - Auto update with configurable interval





jsroot, event display

- Needs web browser with Javascript support.
  - Luckily, this is a standard widget for CSS!
  - → technically, there is no show stopper
- Adding to basf2 DQM Histogram Module chain: four (!!!) code lines.
- Performance?
  - Is running one instance per widget a problem? No!

JSROOT Summary

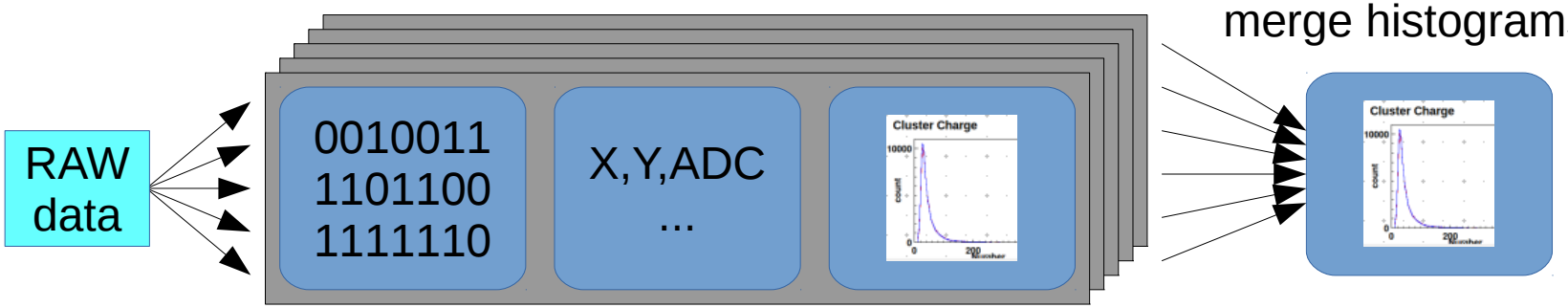
- Replacing histogram widget technically possible
  - more features (plot fits, limits, etc)
  - part of ROOT (→support)
- Adds to established DQM scheme
  - Decouple Fitting/Plotting from serving histograms/canvases to client
  - Possibility for proxy and "load balancing"
- Could run stand alone or as part of basf2
- Import histograms from BonnDAQ/SVDDQM: PV → ROOT histogram (but thats another topic...)
- To check
  - Performance & Scalability
    - serving several clients, many/large histograms, fitting large number of histograms, large number of fitting modules
  - Memory leak (long term running)

PXD DQM in basf2

- High level DQM provided by Prague group
  - 17*layers = 680 hists
  - Incl. 2d histos with pixel binning
  - Binning configurable
- → too much

ExpReco: Histogram Filling and Summing

distribute event wise to n threads on m machines



unpack .. reconstruct … fill

merge histograms every x events

ONE x for all histograms.
Defines update speed
(but not in „seconds“).

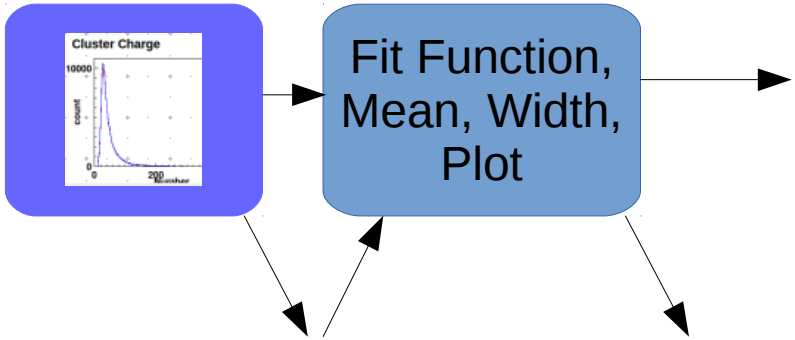
no actual merge; but still histograms are send out only every x events

single thread

unpack .. reconstruct … fill

ONE x for all histograms. Defines update speed (but not in „seconds“).

on merge .. export and/or fit



fit result to EPICS/ Alarm System

(to other server)

to histo proxy

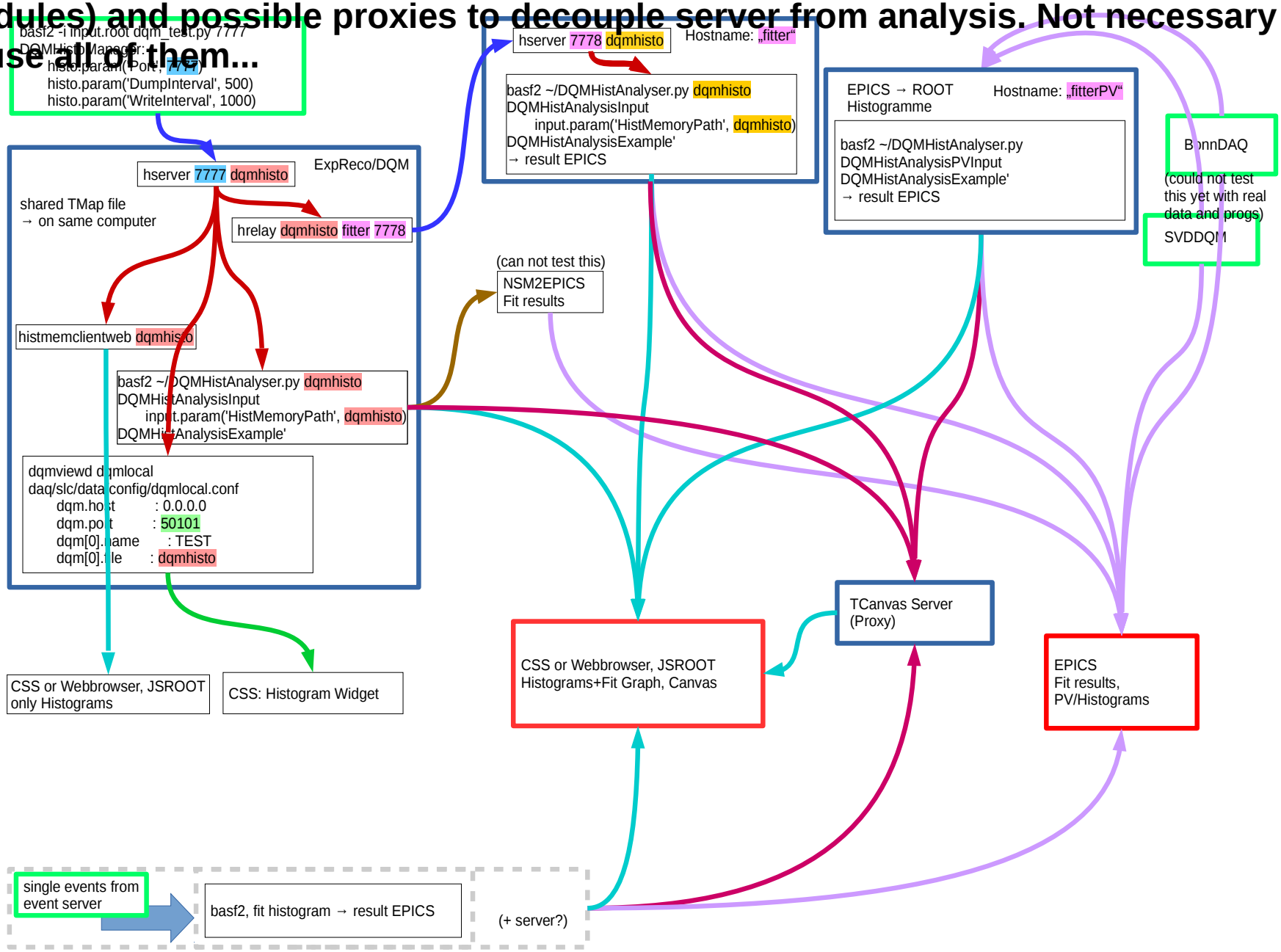
All variables have to be defined in EPICS. Alarm limits should be stored in/read from the data base

DESY Experiences

- Histogram update ~100MB every 30s was too much for the (not optimized) setup → performance on final setup has to be verified

Possible Connections and Histogram Data Flow (some Redundant)

These are just possibilities where to get what data (using available! code and modules) and possible proxies to decouple server from analysis. Not necessary to use all of them...



DQM – Tech Details

- Histograms from within basf2 (Express Reco)
- How and where to show histograms → Belle 2 common
  - Automatic analysis and report back to operator? → Belle 2 common
- Where: CSS
- How: CSS widget and daq/basf2 modules/programs. Main code by Itoh-san and Konno-san
- What is doing technically: Take histograms from basf2 and serve them by Ethernet to CSS clients.
  - (Analysing histograms is a different but related story)
- Thus → if we want to monitor something, we have to unpack it and store it in basf2, make histograms within basf2
  - Unpack data → data objects → plot data
  - DQM code written (be)for DESY TB 2014: f.e. PXDDQMModule, PXDRawDQMModule, PXDROIDQMModule

Issues

- Performance unclear for
  - large histograms (esp 2d)
  - many histograms
  - many clients
- All histograms share a single fixed update interval
- One way communication
  - no way to dynamically switch histograms on/off
- No possibility to show lines, fits, comparison graphs
- Changes require basf2 recompile and exp reco restart
  - Expert needed during TB → no updates

- Improvement needed: → DAQ group

DQM

- Data Quality Monitoring
  - NOT status/health of ASICs, links, etc → EPICS
  - NOT online O(seconds) but close to O(minutes)
- DQM runs on Express Reco and Offline.
  - NOT on the event builder and not on the HLT (PXD data enters afterwards)
- Raw DQM (pixel based, (hardware) cluster based)
  - data rates, mean event size, fired pixels, plain distributions
- Cluster DQM (after clusterizer)
- Overlap with tracking and ROI calculating DQM
  - (efficiency of ROI selection etc)