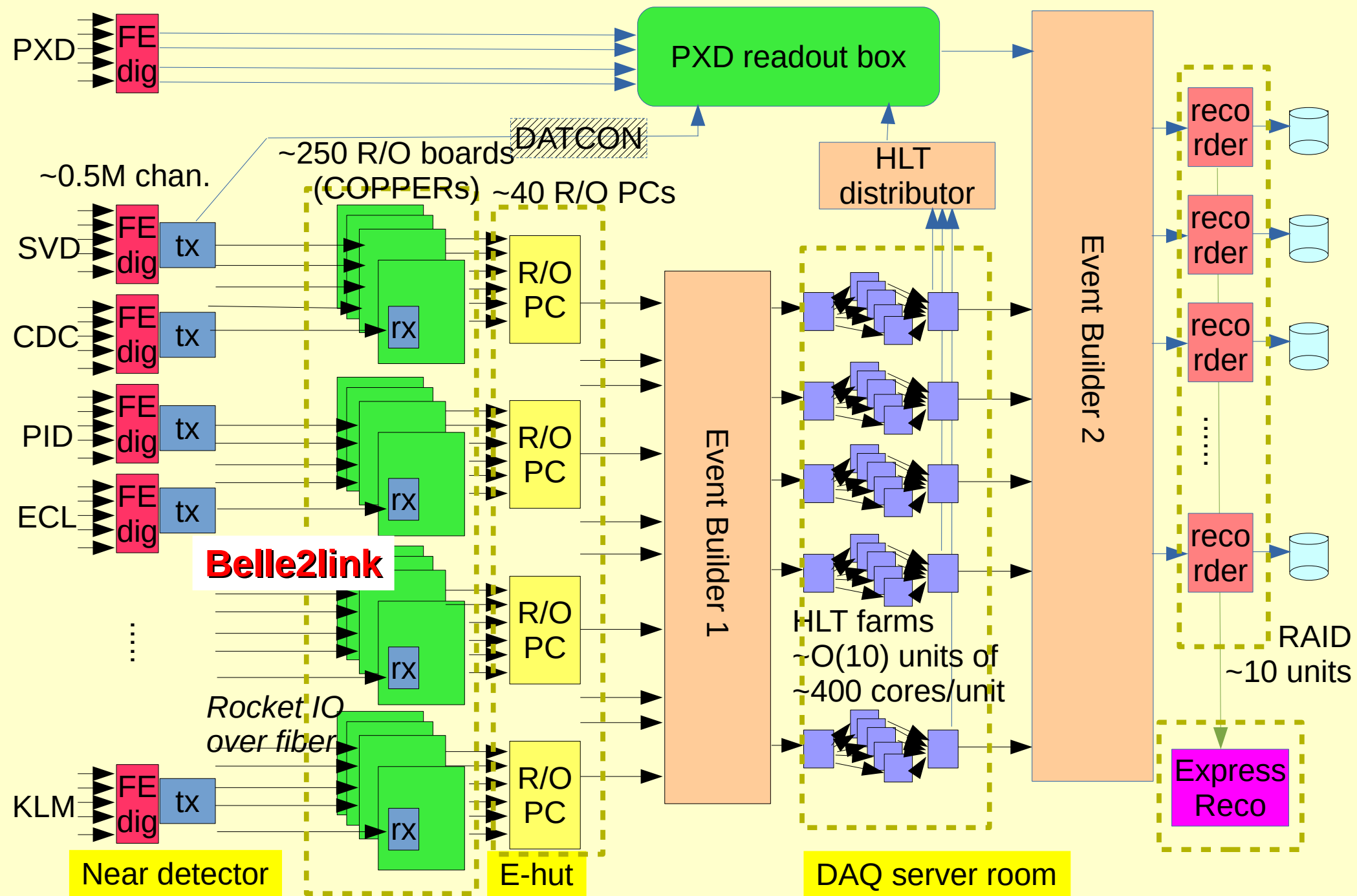


Status of Backend DAQ : Discussion Summary

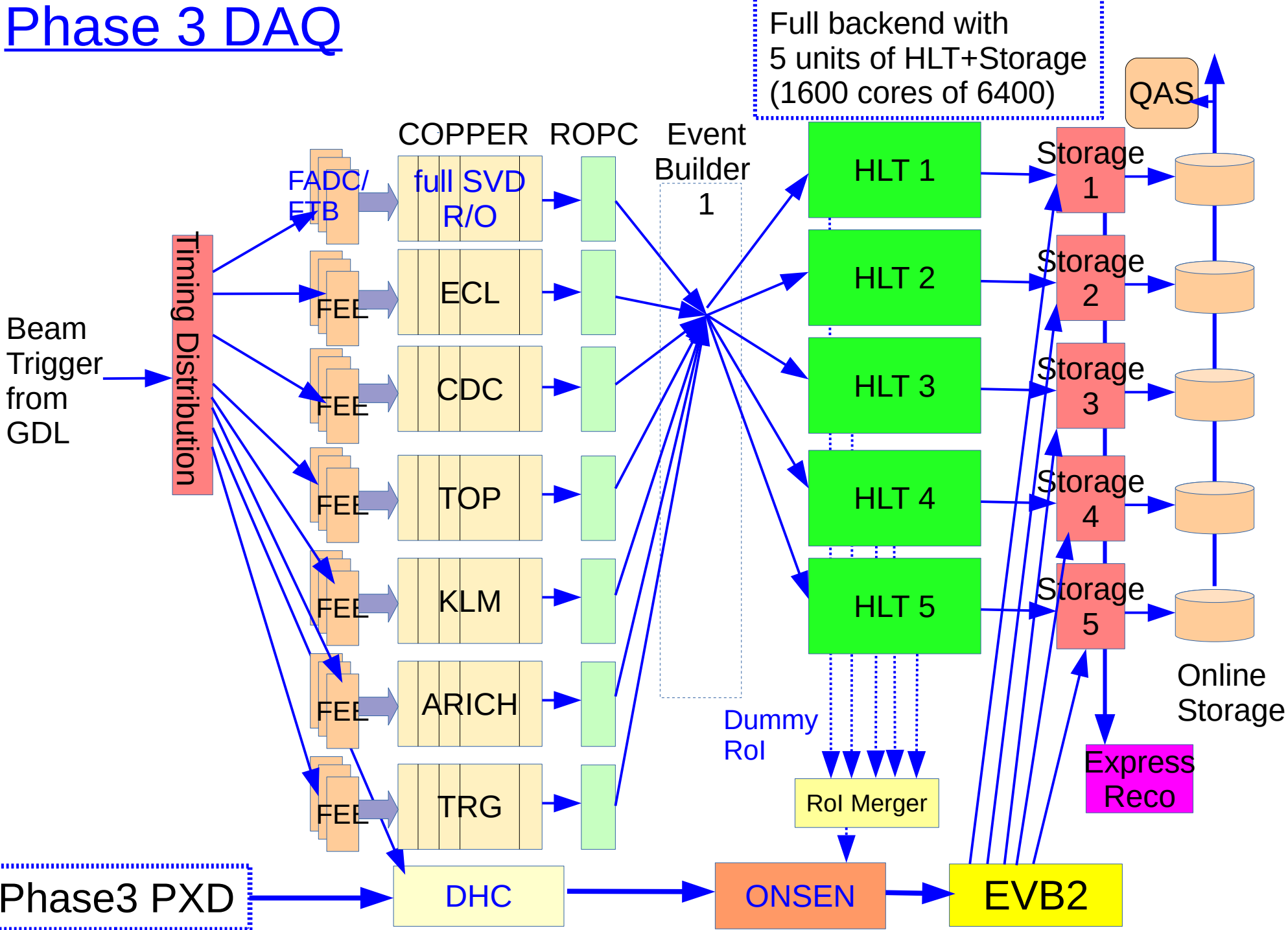
R.Itoh, KEK

Data Flow in Belle II DAQ

“basf2” on Linux CPUs



Phase 3 DAO



Event Builder

No change in phase 3

- No critical problem in phase3 runs
- After summer shutdown, small changes
 - fix for debug of unexpected death of EB2
 - Bandwidth upgrade on EB1
 - HLT06..HLT09

5 HLT units -> 9 HLT units

- Hardware
 - Just purchase necessary 10G optics (SR+LR)
- Software (daq_slc pull request 151)
 - Related items: eb1txd, eb2tx
 - eb1rx, eb2rx don't care # of HLT
 - eb2tx has static-sized array to hold parameters and statistics for each HLT.
 - **Careless extending them will cause SEGV**
 - we met it last year, but just reduce the size to save time
 - In this time, I'm debugging by valgrind

HLT

HLT DURING OPERATION

Since beginning of Phase 3, many crucial bug fixes have been implemented thanks to the help of many people

- stable signal handling
- faster stopping time
- many DQM plot iterations, fixes, improvements
- configuration file management
- totally new steering file setup
- inclusion in the zabbix setup

Additional to that, we have validated and implemented multiple new basf2 releases and many database updates

HLT operation is much more stable in the recent times

STABLE SIGNAL HANDLING

- Stopping of run and aborting of framework is done by sending Linux signals to processes
 - Very important: processes need to go down cleanly
 - Check that processes are down
 - Close ring buffers correctly
 - Typical flow: main loop
 - instead of just aborting, break out of main loop
 - make sure to call destructors
 - e.g. `daq_slc`: Guards
- Important: blocking pipes are always evil
 - e.g. seen processes stuck in termination, because logging can not be flushed
 - also writing out to `nfs` can be a problem

PRESENT

- Filtering turned on
 - Did first studies with HLT filtering turned on
 - Large trigger performance studies on data before that
 - No show-stoppers seen
 - Expected data reduction approx. factor 8-9
 - Open discussion item: What to do with dismissed events?
- CVMFS
 - Installed already on all HLT machines, quick analysis server, ExpressReco and monitored by zabbix
 - Used for `daq_s1c`, `basf2` and conditions db
 - Also want EB software :-)
 - Still not using same `daq_s1c` as the rest. Some discussion items .
 - see Oskar's talk

WHY A **NEW** DATA TRANSPORTATION SCHEMA?

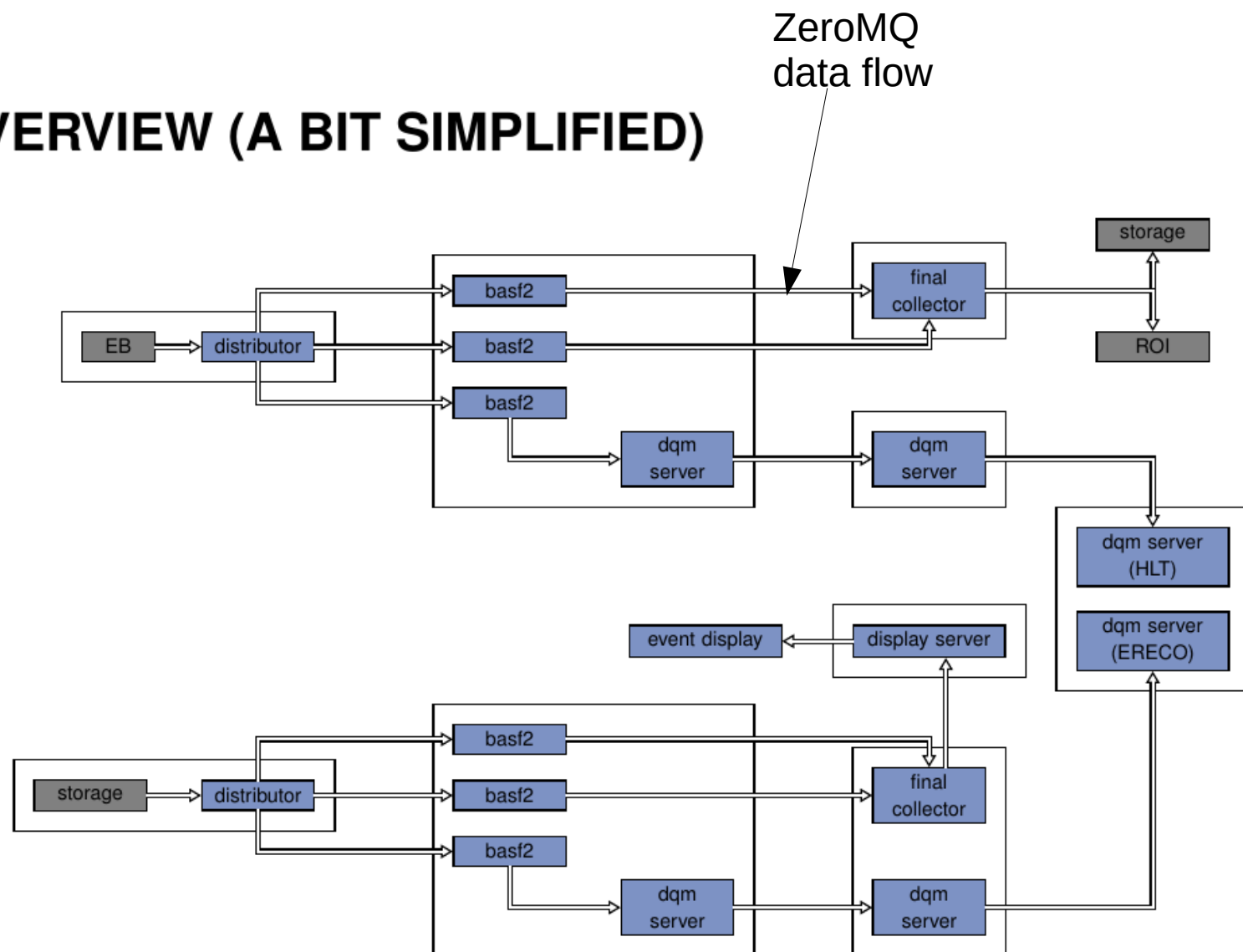
The current data transportation schema based on ring buffer works now very stable!

Still, we want some additional features:

- not possible to sent signals, e.g. run stop etc.
 - therefore: not easy to implement needed features like load geometry on startup, quick abort
- problems with residual state after abortion, sometimes cold restart needed (unpredictable)
- (sometimes) not all histograms can be stored at run end to have short stopping times
- (sometimes) events are missing (reported by PXD)

Important: It is maybe also possible to include the features we want in the current schema - however we discussed to use a more modern schema based on ØMQ.

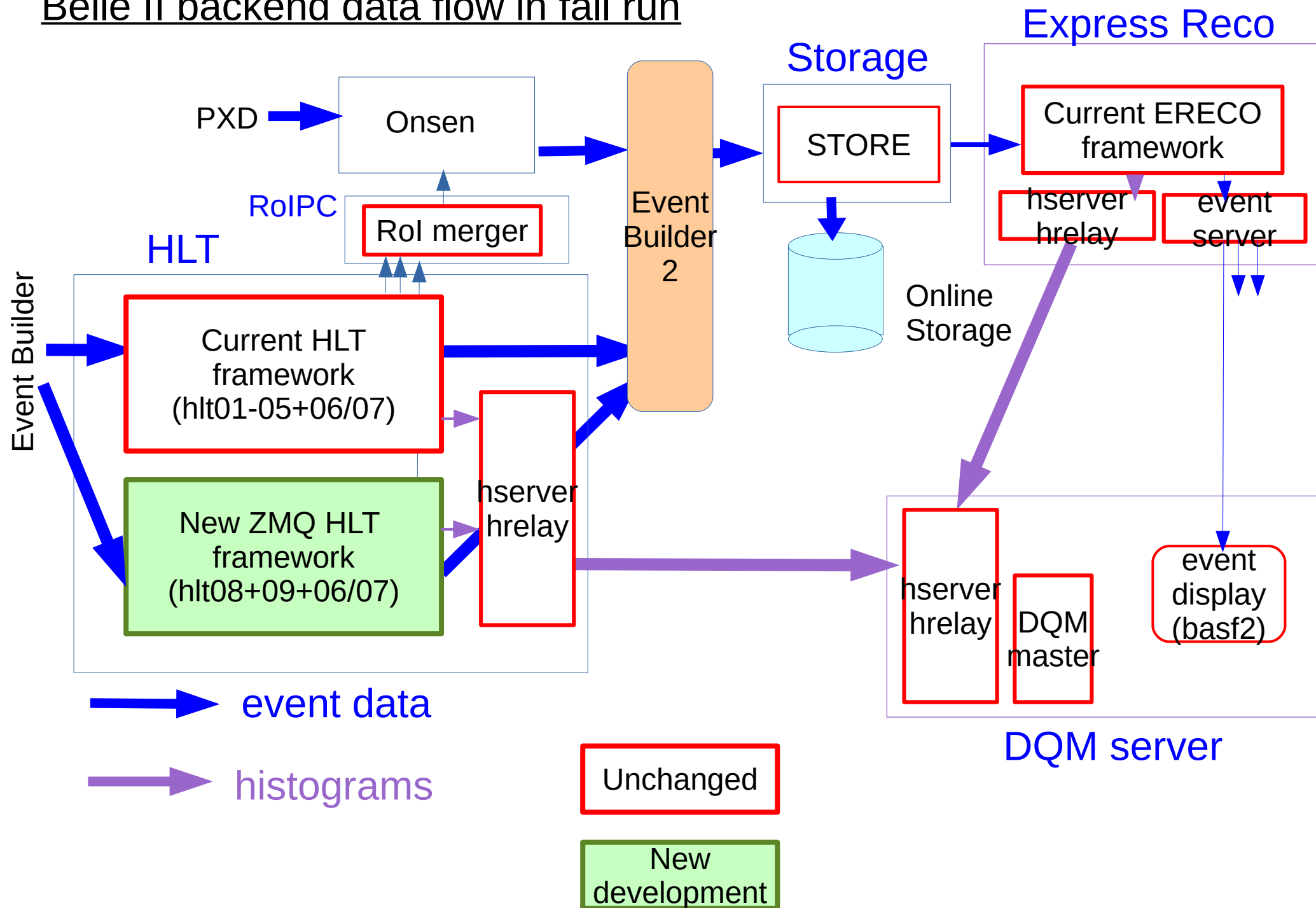
OVERVIEW (A BIT SIMPLIFIED)



CORE FUNCTIONS **WORK**

- Runs with "normal" rates (7 kHz with reconstruction and 12 kHz with passthrough on one HLT unit) very smoothly
 - These are actually really high rates for a single unit (compare: we expect 30 kHz with 20 units for normal data taking), but there is also basically no data content
 - At higher rates the storage was the limiting factor
- Many SALS and Start-Stop performed
- Longest run was approx 24h with 2 kHz
- DQM Histograms, ROI sending (also with very high rates), storage and cosmics run were tested

Belle II backend data flow in fall run



However.....

Nils is now leaving collaboration.....

N.Braun

FUTURE

- Improved HLT trigger menu and reconstruction speed
 - Work in progress, JIRA issues and discussions
 - Will be a continuous effort
- ØMQ (see other talk)
- Updated cvmfs (see Oskar's talk)
- More HLT units

Personal Note: I will not be the online coordinator for the upcoming data taking. Please ask questions to Chunhua or (starting from October) to Markus Prim.



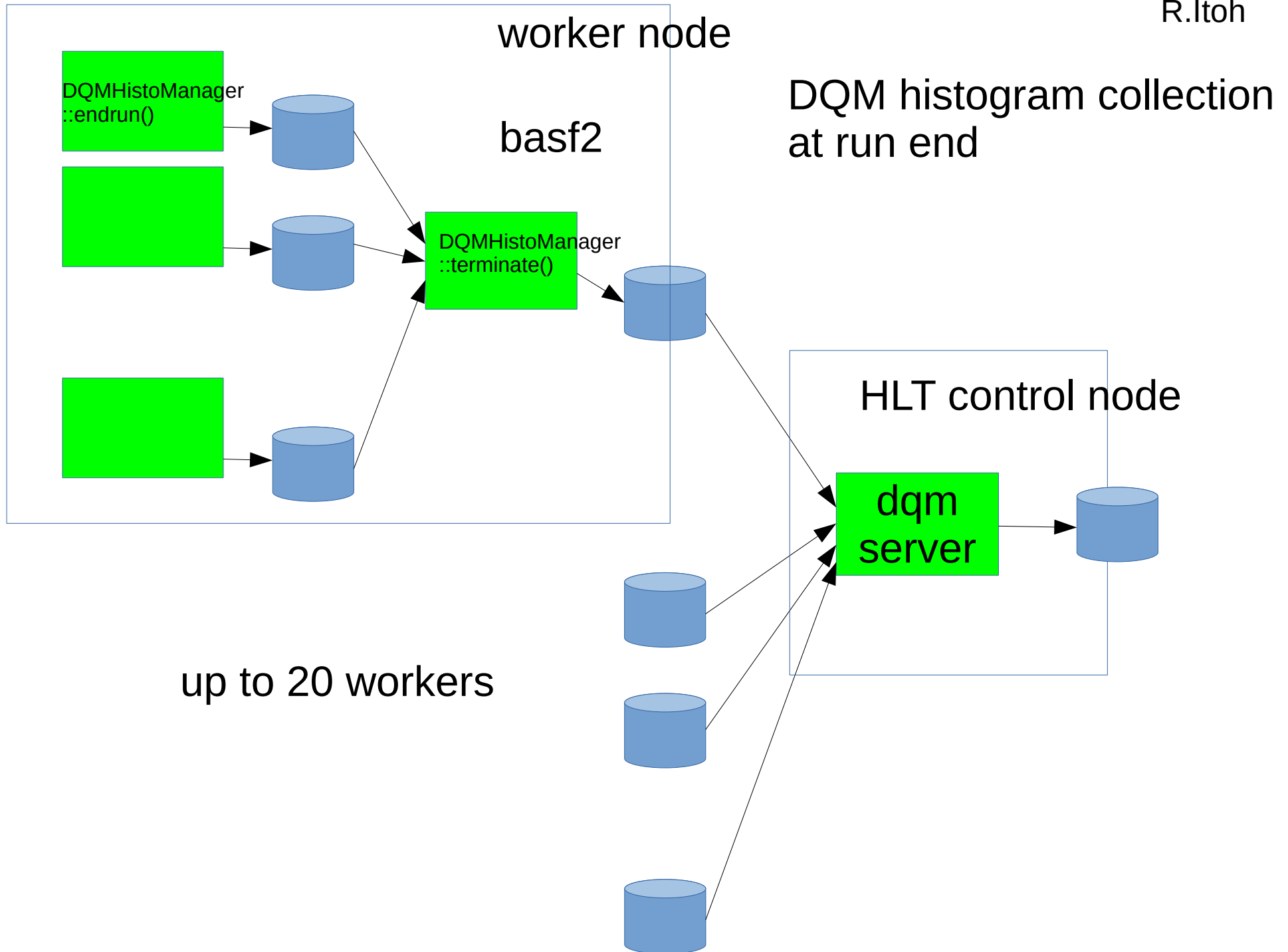
STOP AND START AGAIN

Especially at high(er) rates, starting the next run after stopping leads to a strange incident:

- few events of the last run (8-12) still reach the EB1 and are given to the HLT before the events from the new run come
- Master detective Yamagata-san found out after some longer investigation: events remain in the FIFO of the COPPERS (this time it was CDC, do not know about the others) because stop does not lead to a flush
- This breaks some of my assumptions, so I needed to change parts of the code
 - Now it is not a technical problem for HLT anymore to receive mixed-run events, but it is still a but "strange" (and has probably some second-order influence)
 - Can every system handle those additional events (I think storage dismisses them?)
 - What is our policy with those events?

DQM

- At each run end, the DQM histograms are supposed to be collected and stored in a file.
- In the previous implementation, the collection is done separately from the live DQM collection to manage not only 1D/2D histograms but also TTrees/TNTuples.
- The collection is done in 3 steps.
- When a run is stopped,
 - 1) basf2 is terminated by sending signal and the histograms accumulated in each event process are written to files corresponding to the processes in `DqmHistoManager::endrun()`.
 - 2) The histogram files are collected and added in one file per worker node in `DqmHistoManager::terminate()`.
 - 3) The control nodes collect the files on worker nodes and add them into a single file.



- This mechanism includes three file writing. The number of DQM histograms in phase 3 was around 7500 and it took a long time to go through all 3 steps.
 - > **Up to 5 minutes to stop a run!**
- Several reasons.
 - 1) The files were first placed in NFS filesystem shared by all worker nodes on a GbE network.
 - > Changed to local file system, but still ~3min or so.
 - 2) The number of histograms was reduced to 1/3. But still took a few minutes to stop.
- **Finally gave up to use this mechanism to leave DQM histograms in a file.**
- Instead, **the histograms collected by live histogram transport are stored in a file at run end. "DQMMASTER"**

- Number of histograms has been reduced down to 2800 (thanks to the effort by TRG group), and the time for HLT stop was reduced to 2min. or so. But still long.

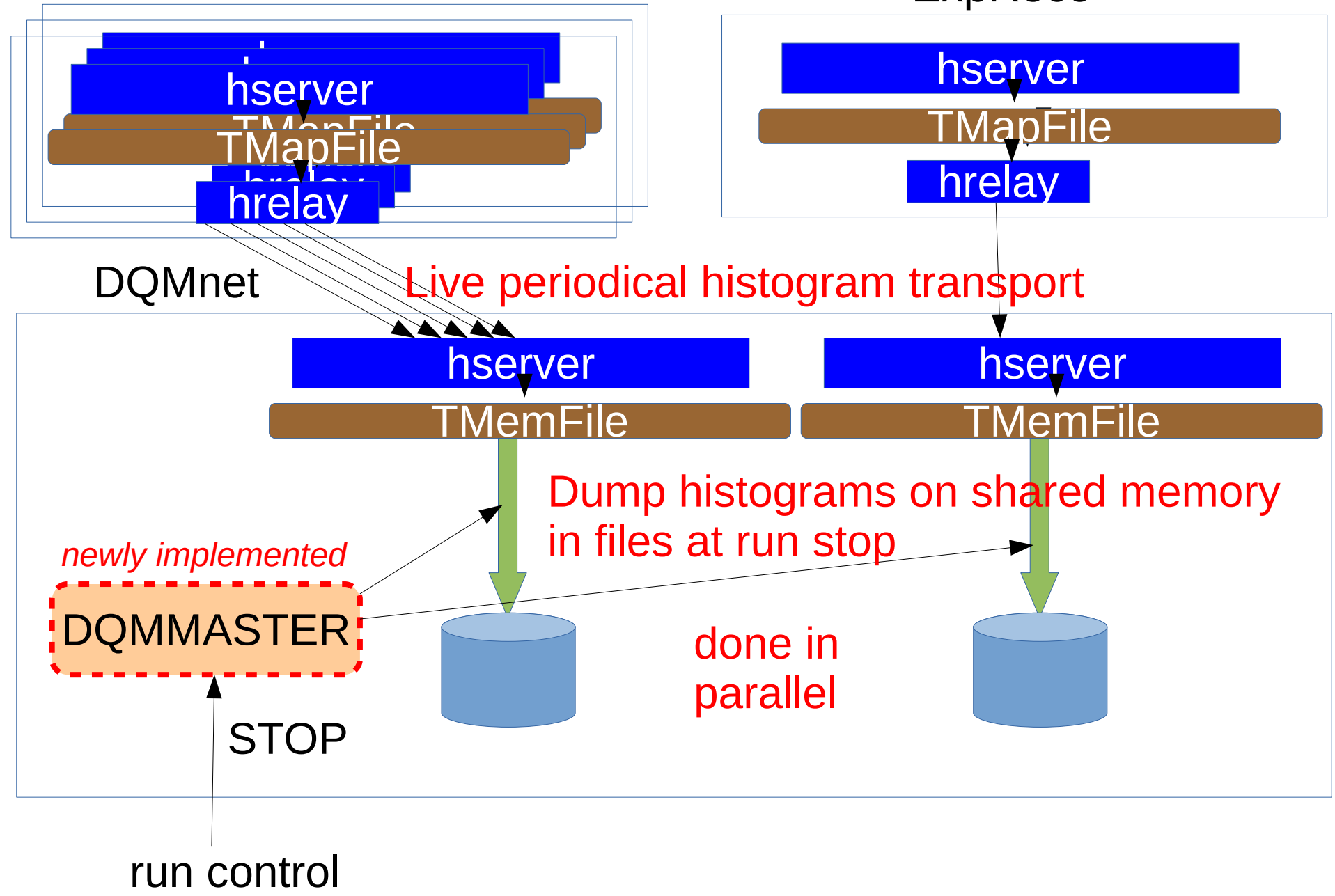
- Final solution:

- * To give up the histogram collection at each run end.
- * Instead, store the live histograms already transferred to the main node at run end.
- * For the purpose, a new NSM node called "DQMMASTER" is installed.
- * It receives the STOP signal from run control master and dump histograms on TMemFile into files.
- * It was implemented during last acc. maintenance day and now being operated.

-> The stopping time reduced to ~30 sec.

HLT01-05

ExpReco



DQM Analysis Module

- The histograms saved in the shared memory of the DQM server (dqmsrv1) can be accessed by multi processes to perform high level analyses.
- An input module (DQMHistAnalysisInput) load the histograms for following processing.
- The analysis module can do:
 - Fit the histogram and draw the curve and parameters.
 - Change the background color to indicate the current working condition and whether experts should be notified.
 - Draw additional lines and marks on plots to indicate good and bad regions.
 - Export features extracted from the plots to EPICS PVs for other programs (CS-Studio) to use.
- The analysis module also has the access to Belle II database.
- Currently most detectors have or plan to have their own analysis modules.
- TCanvas objects are generated by analysis modules to display the plots and other analysis results.

Online Display

- After the processing of the analysis modules, the TCanvas objects generated are transferred by an output module (DQMHistAnalysisOutputRelay) to a canvas server.
- A canvas server is a root macro which uses THttpServer to serve the TCanvas objects. It can be accessed by web browser and other programs in the DAQNET.
 - To access from inside the KEK network, a ssh tunnel on bdaq is necessary.
- On the client side, the JSROOT library is used to render the plots and provide some interactive functionalities like zooming and set log scale display.
- The online display is shown on the monitoring screens in the Belle II control room in Tsukuba Hall B3.
- For control room shifters, there is a page for them which contains selected plots from all detectors to make it easy for non-experts to check their running conditions. More detailed plots can be checked by detector experts in other pages.

Mirror Site

- One mirror site on DESY has been prepared for people outside of KEK:
 - <https://dqm.belle2.org>
 - Log in with Belle II DESY credential.
- A program running on the DQM server (dqmsrv1) takes the snapshot of the canvas server every 5 minutes and transfers it to the mirror site.
- The canvas server running on the mirror site reads the snapshot file (root file) and serves the canvases for public access.
- Access speed is a bit slow from KEK, but much faster from Europe.
- The mirror site also hosts the histograms from past runs and offline plots for sub-detectors (see next page).

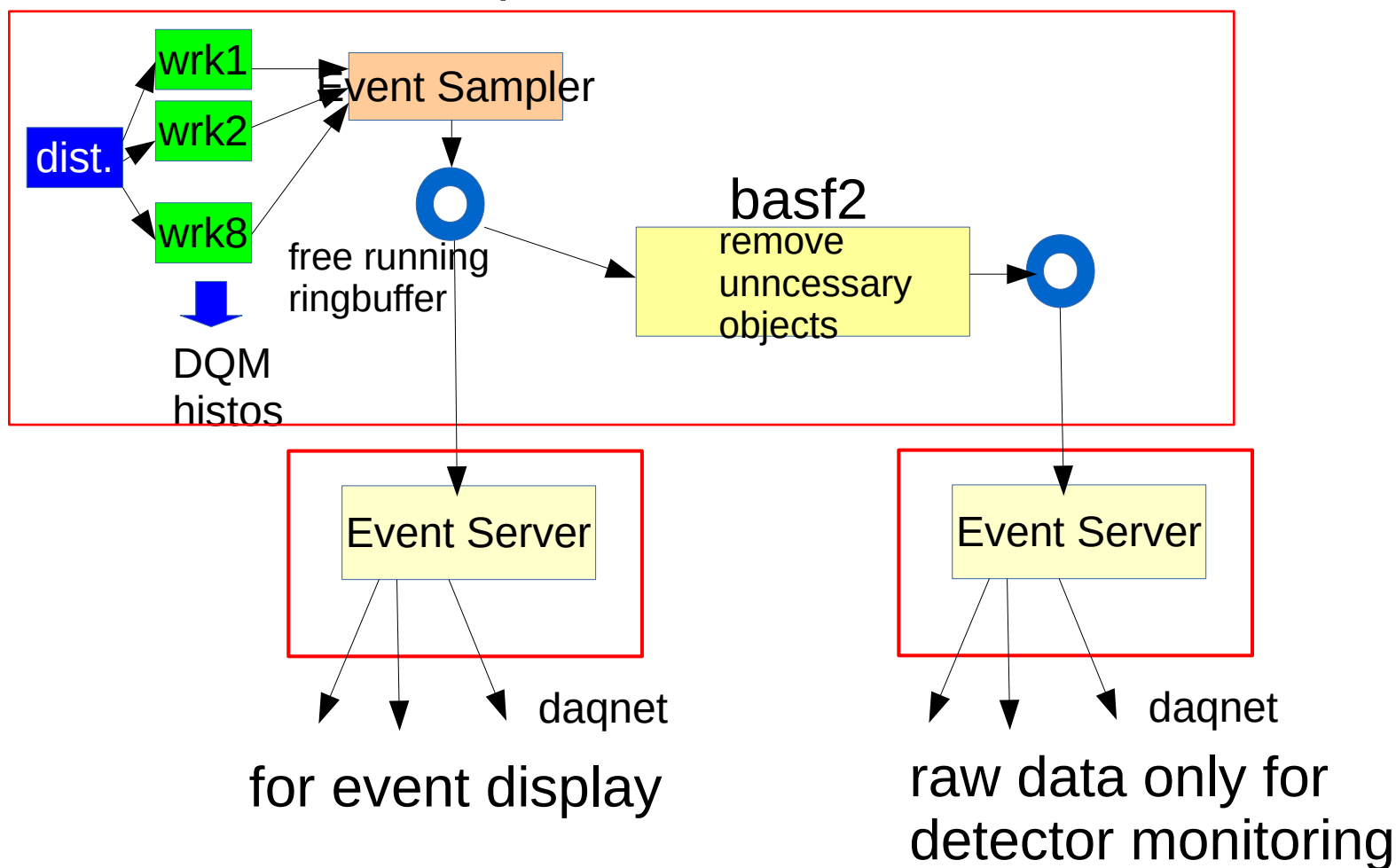
DQM for Past Runs

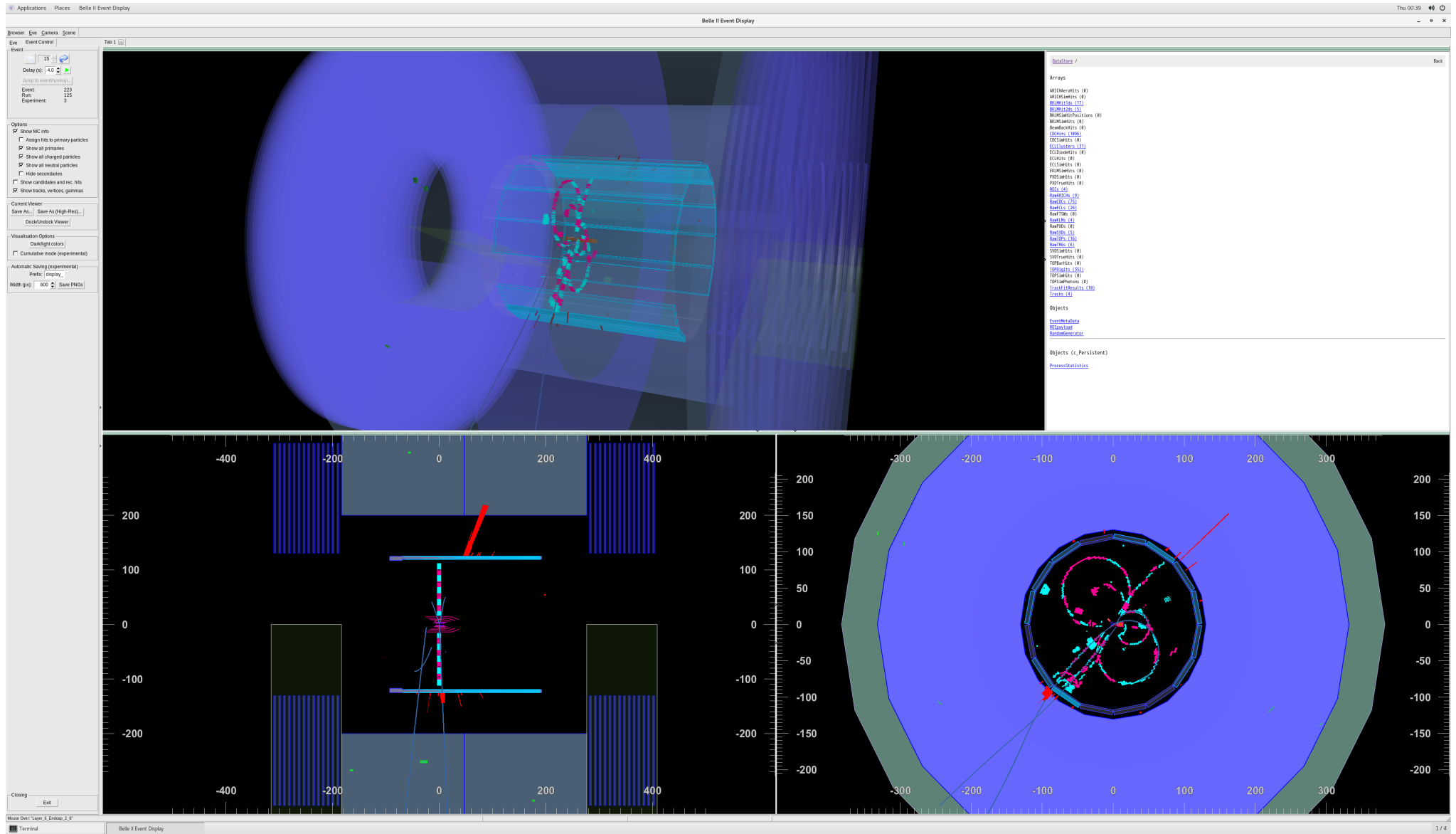
- The DQM root files are transferred to the mirror site for offline checking:
 - https://dqm.belle2.org/past_runs/index_hlt_exp8.htm
 - Only physics, cosmic and beam runs are transferred.
 - The root files are reprocessed offline to generate plots from analysis modules.
 - These files on mirror sites can be downloaded, checked with JSROOT browser or displayed with the same layout as in online display.
- Synchronization interval: ~30 minutes.
- Reference display is also enabled for the past runs.
- Sub-detectors can also use this mirror site to store their own plots for offline checking.

Event feeding to other apps

- Sampled events could be useful for the detailed debugging of detector.
- Reconstructed objects are not necessary for the debugging.
 - >Removing them from sampled event stream is preferred.
- An additional event server is being prepared for the purpose and will be tested soon.
(Simple test was already performed.)
- I will announce it when ready with a simple description on Confluence.

Express Reco





- Update so that event display can start in continuous running mode without pressing start button is desired.

No Summary of Summary