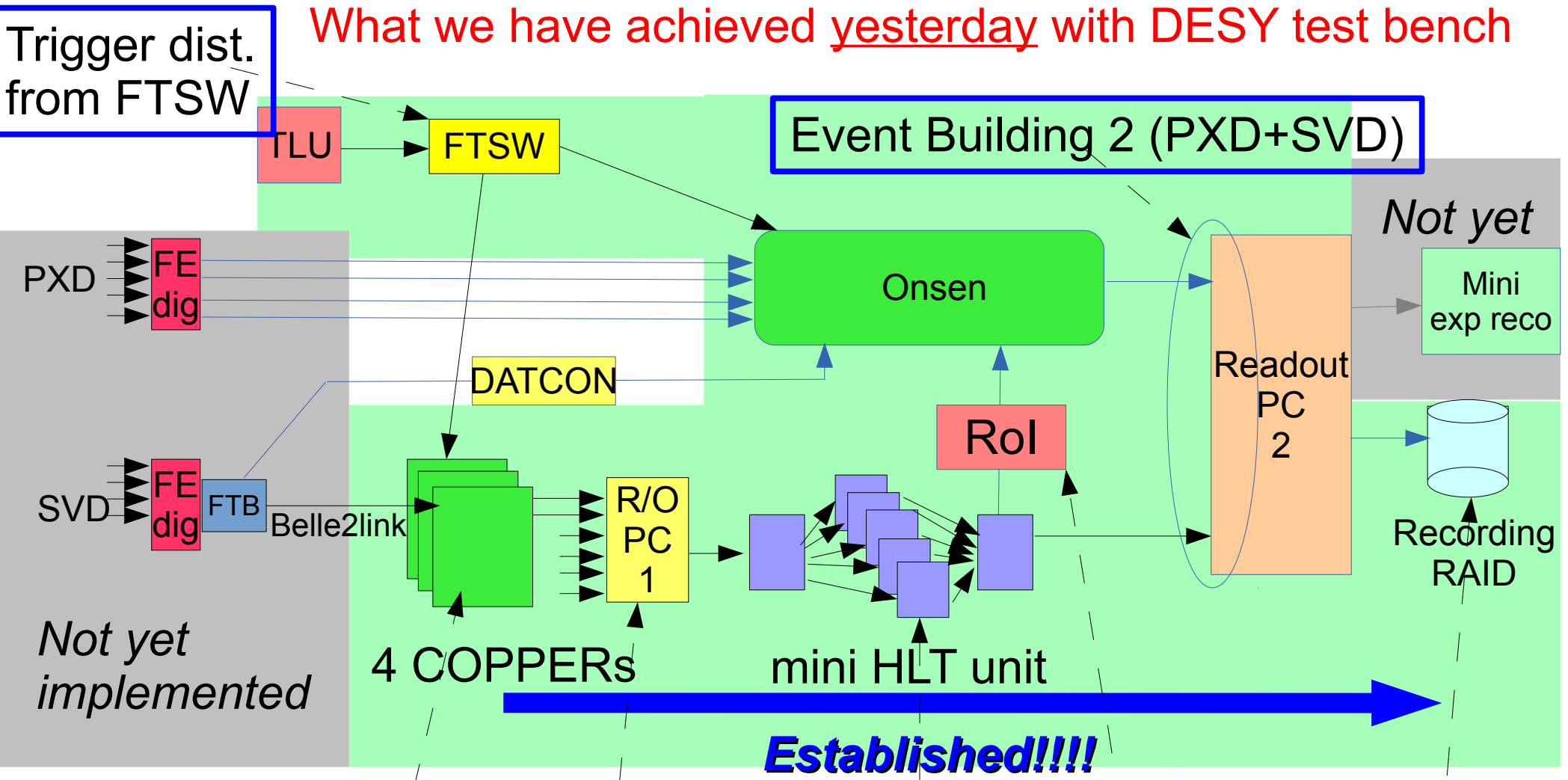
Issues in HLT and DQM for VXD Telescope Test

R.Itoh, KEK

4th PXD/SVD workshop, DESY, Oct.23

What we have achieved yesterday with DESY test bench



- Dummy data generation on B2link receivers
- basf2 based readout

- event building 0
- basf2 processing

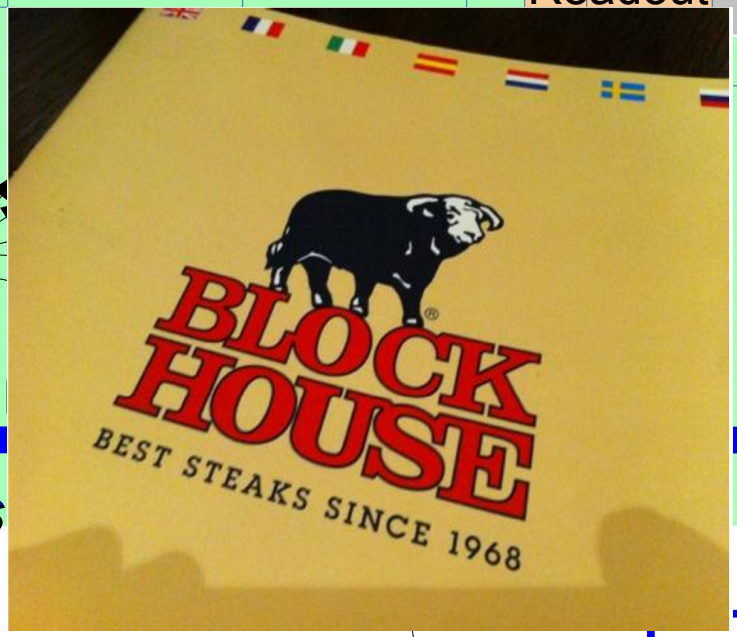
RoI transfer

- dummy RoI generation on 8 cores * 3 nodes
- parallel processing with basf2 / DQM histos

Data Recording in ROOT file with basf2

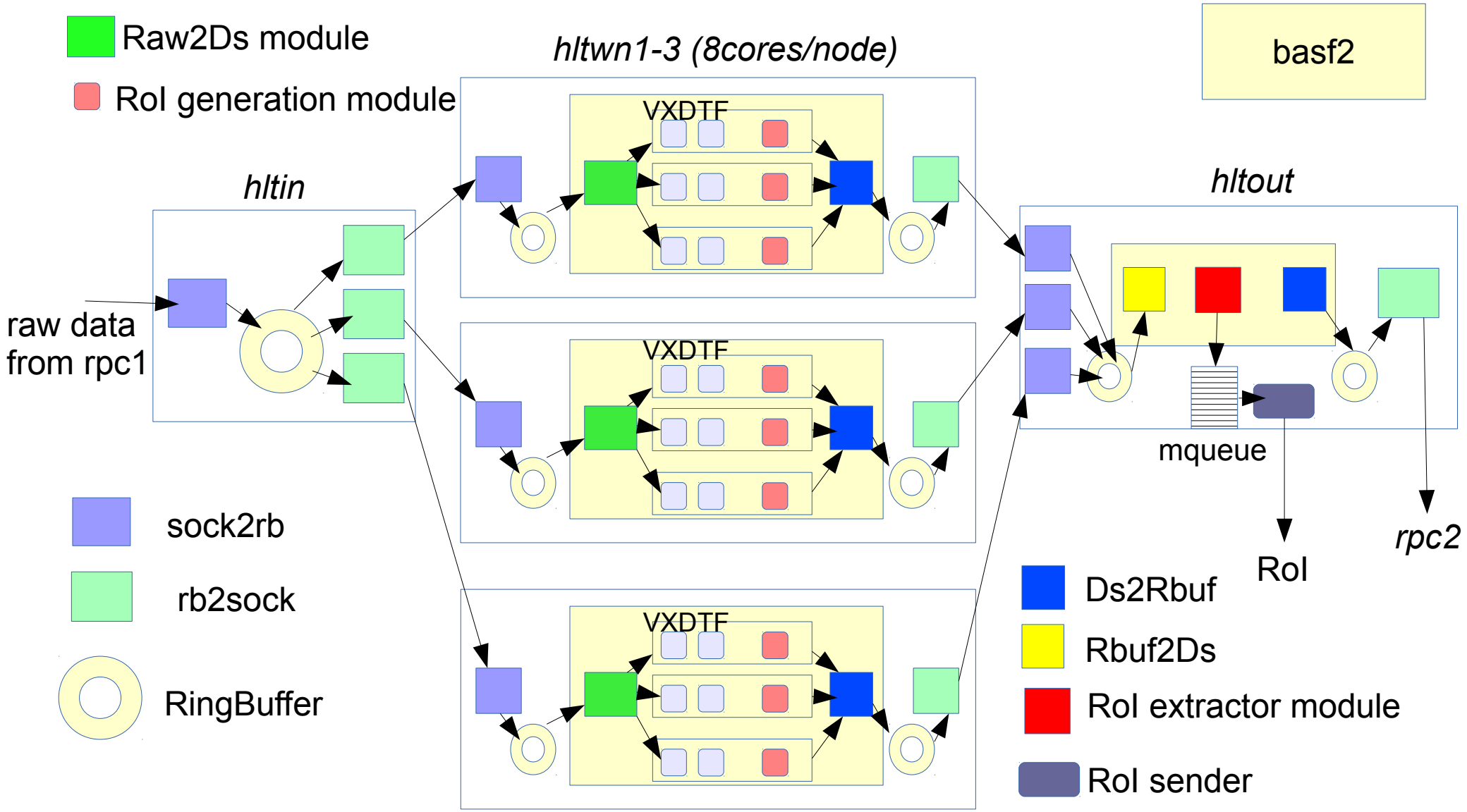
What we have achieved yesterday with DESY test bench

We succeeded to achieve the primary goal of our 2 weeks stay in DESY!!!!!



Many thanks to Alan, Torben, Carsten, and others for setting up the test bench!

Close up of HLT for telescope test



Current HLT performance

- Processing performance was measured in the test bench by receiving COPPER data stream.

- With all data transfer from RPC1 to RPC2 : ~6kHz.
  * Not limited by data transfer bandwidth (although almost reaching at 1GbE maximum (~120MB/sec), nor CPU load.
  * One point bottleneck : DataStore streaming at hltout.
    <- it turned out that current DataStore streaming is unexpectedly CPU-consuming.....
    => should be tuned.

- Removing the DataStore streaming at hltout : ~15kHz.
  <-> Design performance : 10kHz/unit.
    * *We will have 5 units at t=0, and then the units will be added.*
    * *With software trigger turned on, the rate at HLT output node becomes 1/3 of L1 rate.*
      => current performance of 6kHz/unit is still OK.

[Issues in HLT]

- Need to establish the data processing chain on HLT consisting of
  * Raw data unpacking :
    unpack “RawCOPPER” object into SVD hits/clusters
  * SVD only tracking with raw data.
  * RoI generation (at HLT worker nodes)
  * RoI extraction (at HLT output node)

- All of them are supposed to be basf2 modules.

- We would like to clarify the responsibility and timeline of each of them.
  As far as I know, responsibilities are shared by
  * Raw data unpacking : ?????
  * SVD only tracking (VXDTF?) : Jakob
  * RoI codes : Euginio and Giulia -> seems to be ready!

- My worry : heavy MC dependence of existing SVDDigitizer and VXDTF which I noticed when looking at codes.
  => Really usable for test beam?

- Strategy to test SVD only tracking on our test bench w/o beam is still unclear.

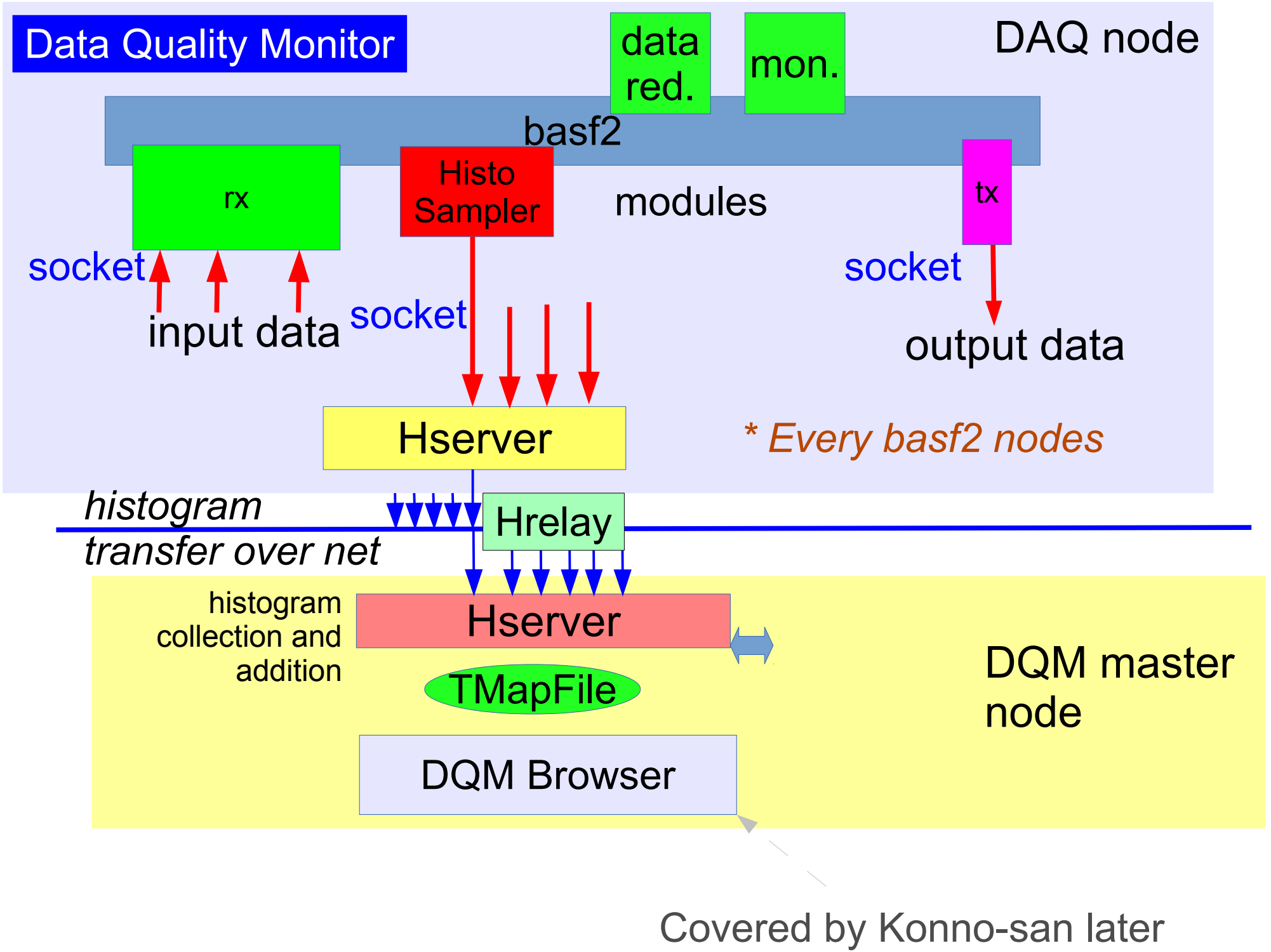
  * Need to estimate the realistic performance of full processing chain on the test bench HLT
    -> tolerable up to 8 kHz?

  * For the test, I need a working basf2 script with “simulated” raw data of SVD.
    -> When are they available?
    or, *can we connect real SVD to test bench well before the beam test and test the tracking by cosmic?*

- HLT operation in test beam
  * No event selection is supposed to be performed. All the events are recorded w/o any “software trigger” -> Do you agree?

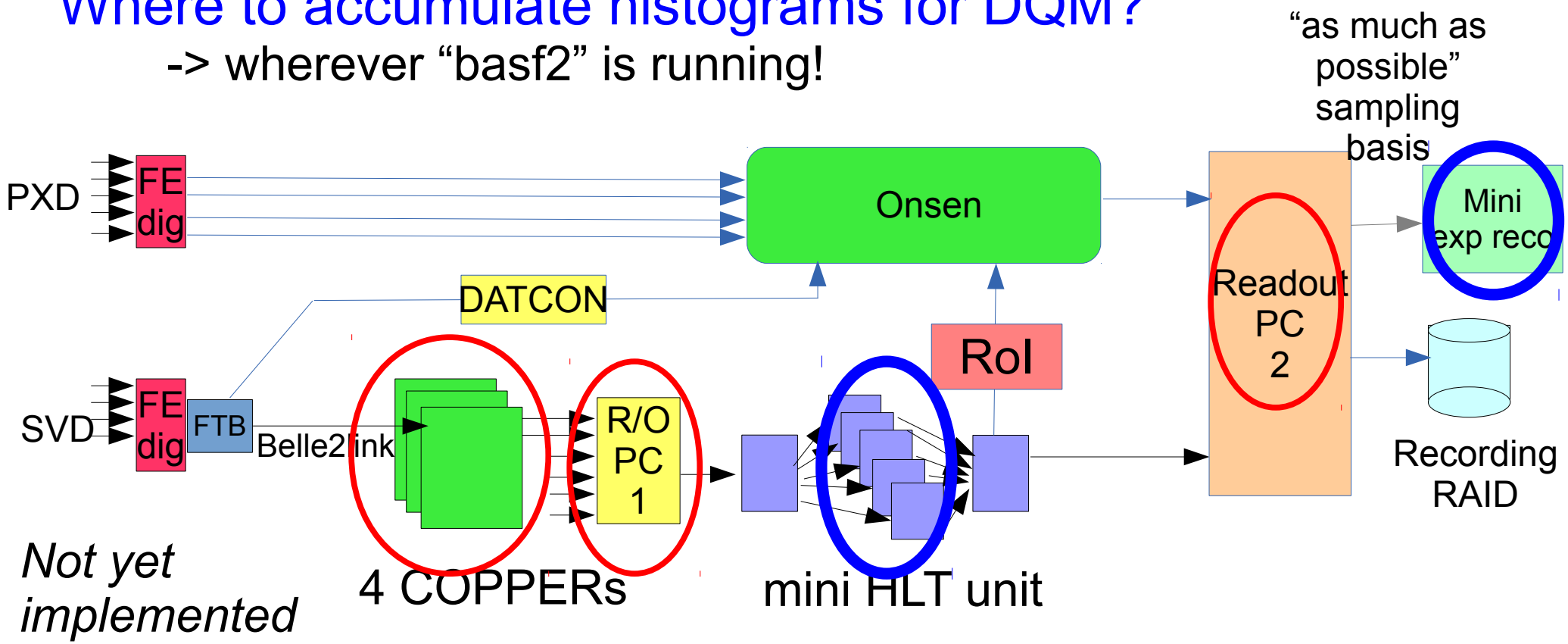
We need to draw a realistic time-line to prepare the HLT processing chain on the test bench.

We have only 2 months before the beam test !!!


Data Quality Monitor
DAQ node
data red.
mon.
basf2
rx
Histo Sampler
modules
tx
socket
input data
socket
socket
output data
Hserver
* Every basf2 nodes
histogram
transfer over net
Hrelay
histogram collection and addition
Hserver
TMapFile
DQM Browser
DQM master node
Covered by Konno-san later

Where to accumulate histograms for DQM?

-> wherever “basf2” is running!



For detector monitoring, the DQM codes (and also rawdata unpacking codes) by each detector group are supposed to run on

* HLT
* mini-express Reco

only. The basf2 codes on other nodes are limited for DAQ debugging purpose only. (Otherwise, they may deteriorate DAQ perfo

How to write DQM code?

- DQM codes are supposed to be provided as basf2 modules, or you can embed histogram accumulation in your data processing codes.

- To accumulate histograms for DQM, you need to follow a convention to define histograms:
  * Standard ROOT histograms can be used (only 1D and 2D histograms are allowed for real-time monitoring).
  * Inherit from "HistoModule" instead of standard "Module" class in your Module definition header.
  * Do not open your own TFile.
  * All histograms are supposed to be defined in a specific function Module::defineHisto().

- An example can be found in Belle2 library:
  daq/dqm/modules/src/MonitorData.cc
  include/MonitorData.h

- Script to test your DQM code:

```
input = register_module('RootInput')
histoman = register_module ('HistoManager')
histoman.param("histoFileName", "your_histo_file")
yourdqm = register_module('YourDQM')
main = path_create()
main.add_module(input)
main.add_module(histoman)
.....
main.add_module(yourdqm)
....
process(main)
```

module to manage histograms

must be placed right after input module

- Your histograms are all saved in the specified histogram file.

- Multiple DQM modules can be placed.

- If you use parallel processing mode of basf2 (HLT processing), the histograms accumulated by many processes in parallel are automatically added in the specified file at the end.

[Issues in DQM]

Who provides what? and Time-line?

- SVD:
  * Preparation is somewhat straight-forward.
  * All the codes are supposed to run on HLT.
  * DQM can be embedded in SVDClusterizer, VXDTF and RoI generation codes -> supposed to be taken care by corresponding persons.

- PXD:
  * Supposed to run on expressReco node.
  * Simple monitoring DQM (hit map, # of words.....)
    -> don't we need DQM for direct output of Onsen?
    -> If yes, who and how?
  * Monitor of full tracking with PXD+SVD combined
    -> code availability?
  * Responsibility : Peter Kvasnicka ?
  * basf2 framework on expressReco -> DAQ group (in a few weeks)