High Level Trigger (HLT) and PXD data flow in Belle II DAQ

R. Itoh, KEK

Outline

1. Why is HLT needed for PXD readout?
2. HLT structure
3. Requirements to HLT-PXD interface
4. Other considerations in PXD data flow

1. Why is HLT needed for PXD readout?

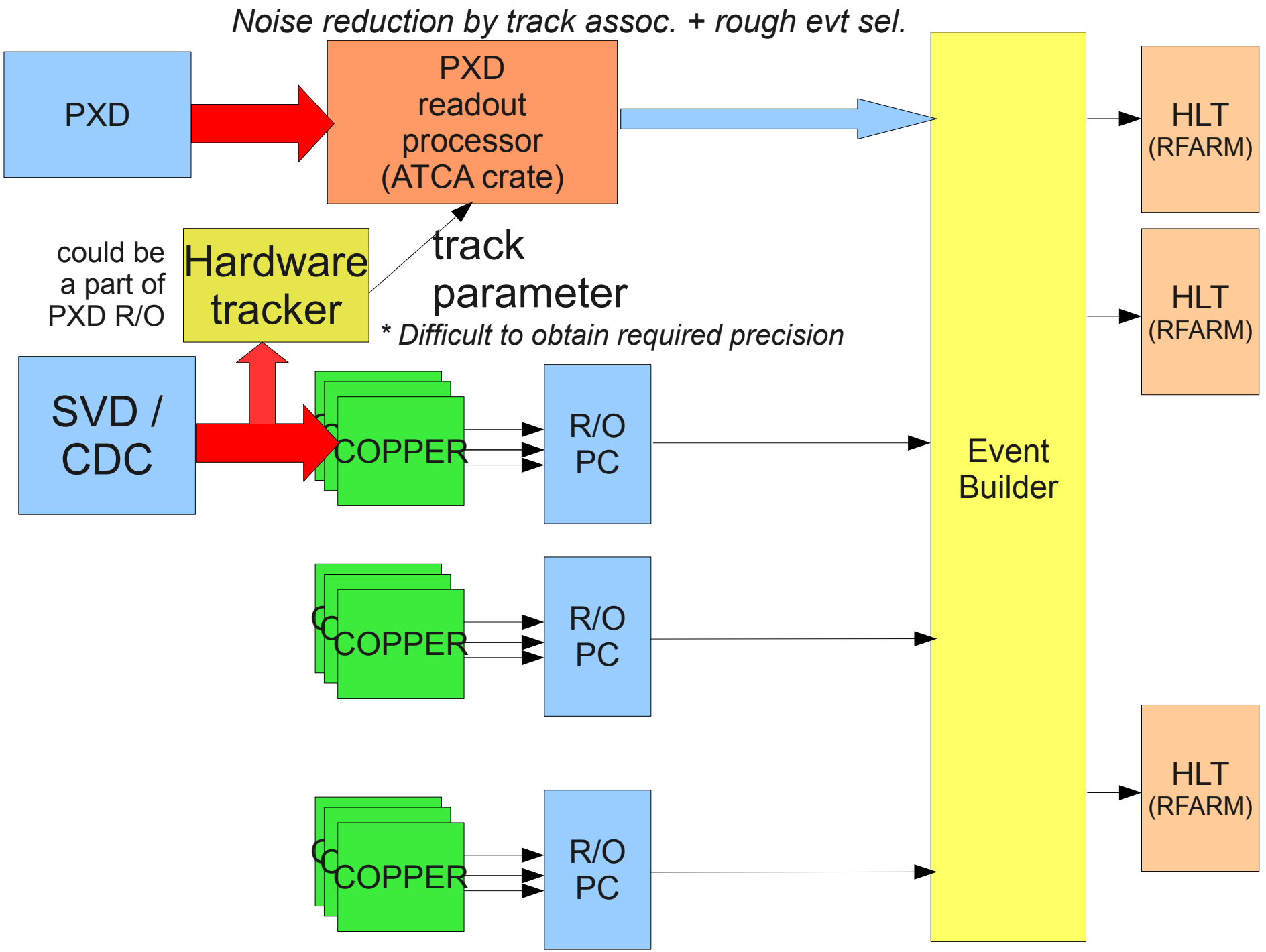
- PXD data size is enormous (1MB/ev.) resulting in a huge data flow (20GB/sec@20kHz!!!!).
  * COPPERs are apparently not suitable for its readout.
  * We cannot manage such a huge data flow, anyway.

- We need to think about
  1) How to reduce the event size to manageable level, and
  2) How to reduce the actual rate of data transfer.

- Event size reduction can be done by hit-track association.
  -> Send only the hits around the identified tracks.
  Expected reduction factor: 1/2 - 1/10

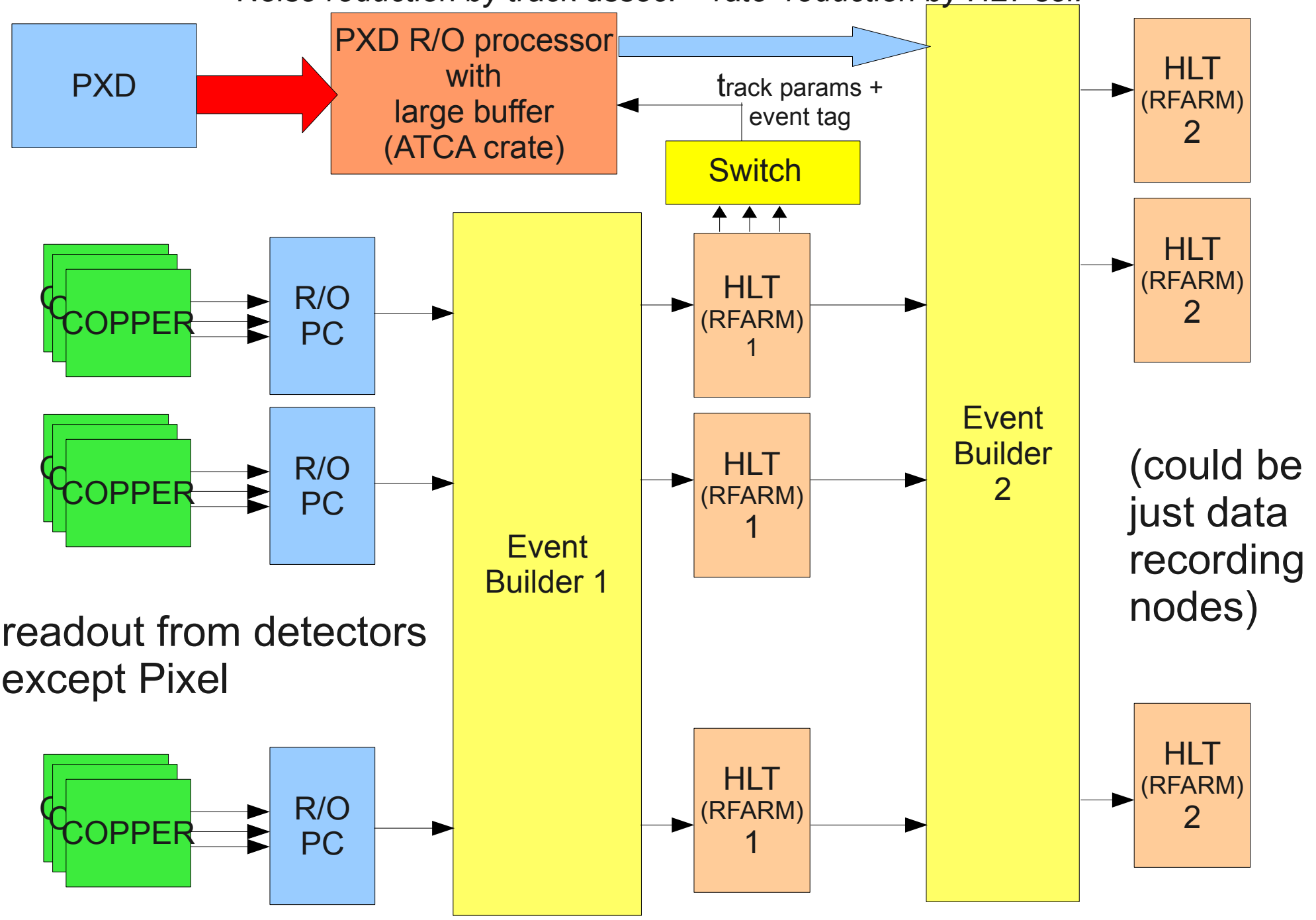
- Rate reduction can be done by pre-selection (Level 2/3-like or HLT)
  Expected reduction factor: 1/2 - 1/10

PXD Integration: Option 1



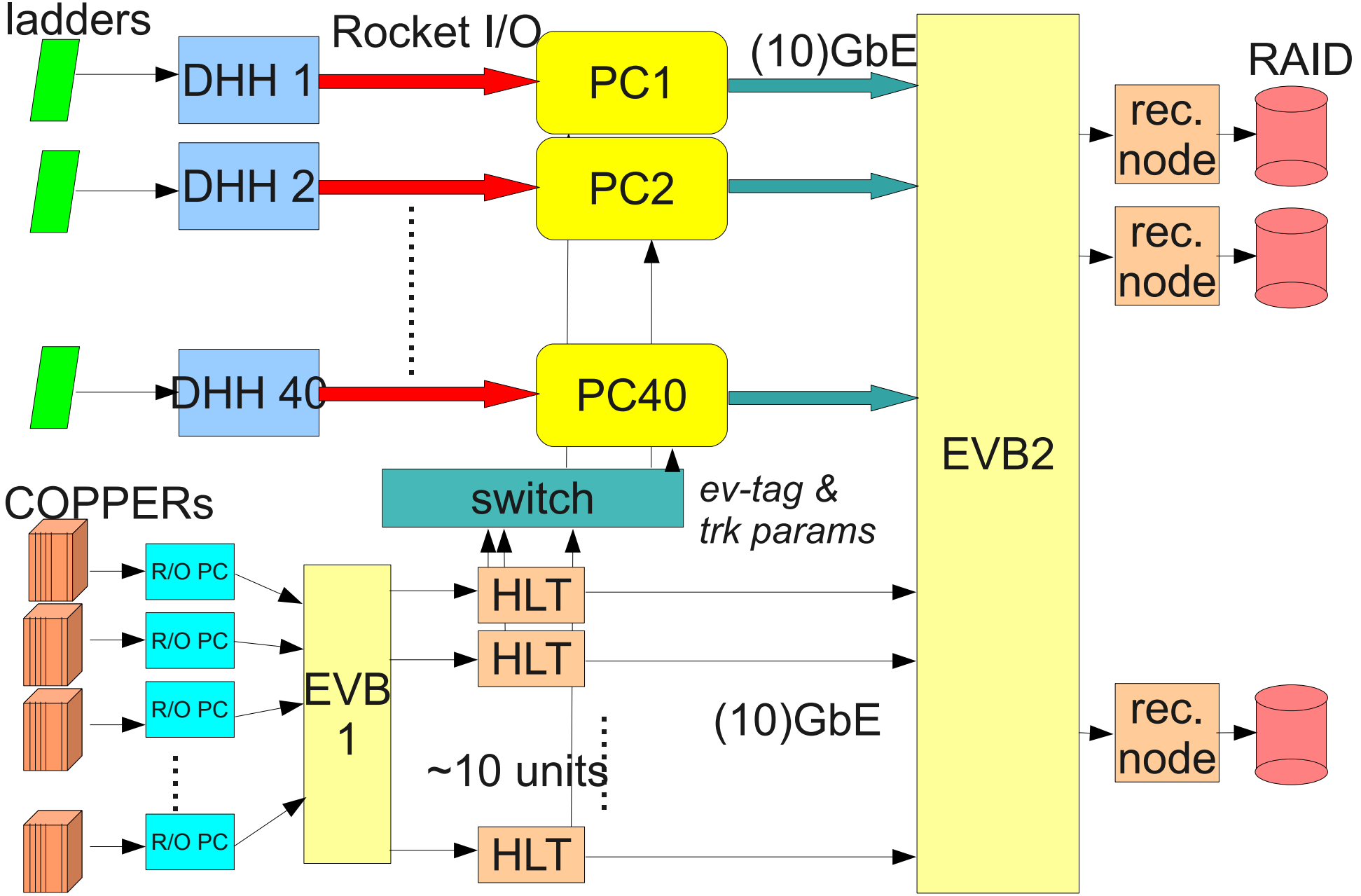
PXD Integration: Option 2

Noise reduction by track assoc. + rate reduction by HLT sel.



PXD integration: Option 3 (or 2': variation of Option 2)

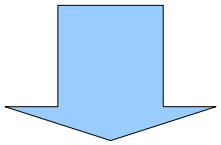
* Fully PC based solution replacing ATCA boards.



- HLT plays a major role in option 2 and 3.

- The track parameters are calculated using the offline tracking software with full SVD+CDC data (i.e. Martin Heck's tracking framework).
  <- The same offline reconstruction code is supposed to run on HLT.
  => Very precise hit-track association
  -> Possible to narrow association region as small as possible
  Reduction factor of 1/10 is in scope.

- The reduction factor of HLT is expected to be ~ 1/10.
  <- estimation based on Belle's RFARM (HLT)



Reduction factor of 1/100 can be expected!

Event reduction at HLT

Experience at Belle

- Two level reduction

a) "Level 4" selection

* Cut in event vertex obtained using fast tracking
* Cut in total energy sum of calorimeter

- Reduction rate is dependent on the beam condition

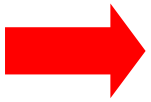
Typical reduction factor ~ 50% (2006 beam condition)

-> Will be moved to CDC 3D trigger (Hardware) in Belle II

b) "Physics skim"

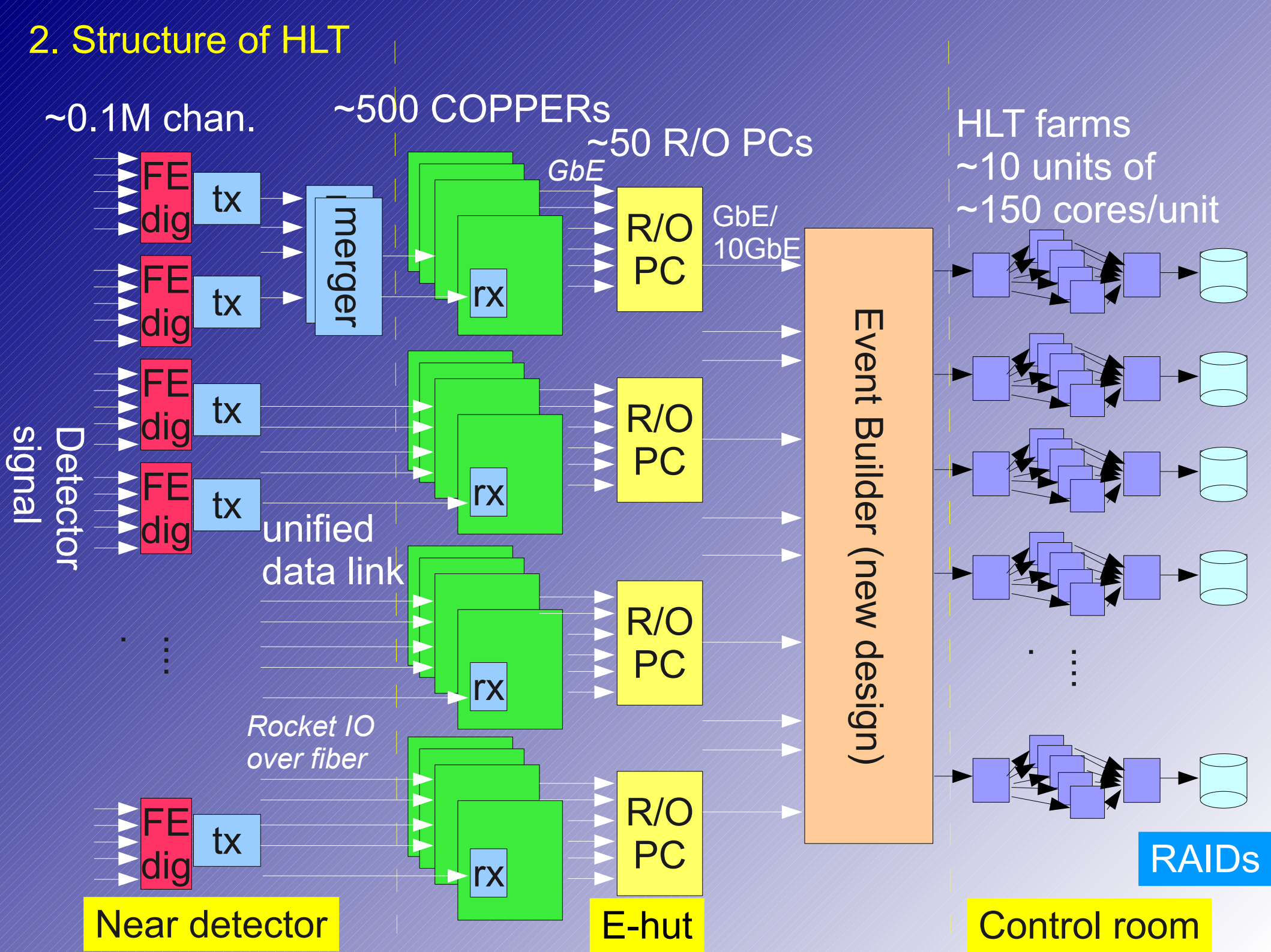
* Physics level event selection using full reconstruction results.
* Almost 100% of physics analysis use so-called "hadronBJ" and "low multiplicity skims + some scaled monitor events.

HadronBJ	: 14.2%	
Low mult. ($\tau\tau$, 2photon)	: 9.6%	2004 experience
Monitor events (ee,$\mu\mu$...)	: ~1%	
Total	: ~25% of L4 passed events	



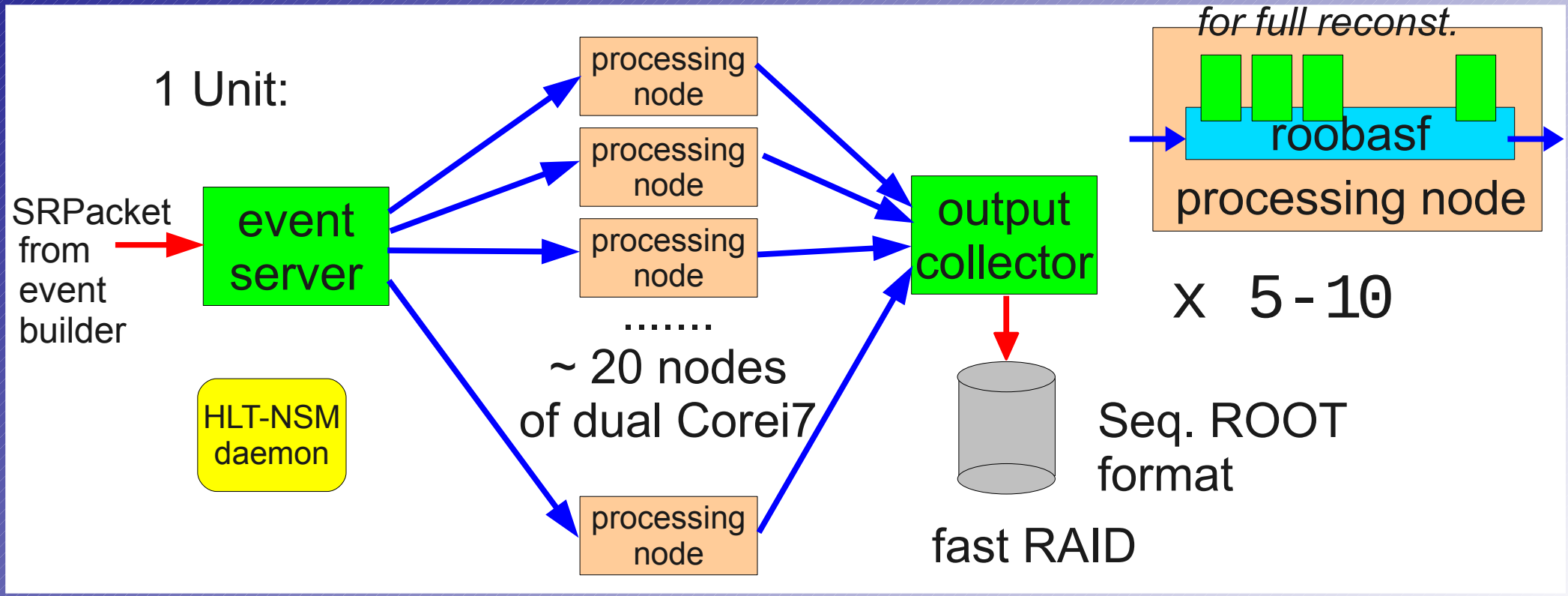
Order of 1/10 reduction at HLT is possible!

2. Structure of HLT



High Level Trigger (HLT) = RFARM@Belle

- Full event reconstruction chain identical to that in offline
- Massive parallel processing using a large number of processing nodes.
- Modularized construction to be scalable to the luminosity.
- 1 unit is supposed to process 2-3 x 10^34 luminosity.
  -> a module consists of ~20 nodes of dual Corei7(3.3GHz) servers
- ~5-10 units at t=0.



Development as a part of “roobasf” project

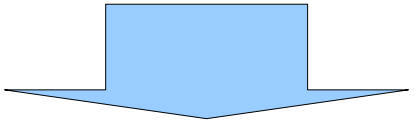
Trigger Software for HLT

- Use the event reconstruction software which is exactly the same as those used in the offline reconstruction

- The software trigger code = “Physics skim” code
  * Hadronic event selection for *B/D* physics
  * “Low multi” skim for tau physics and NP search

- Pre-selection software using fast-tracking (Level-3 like) is required to reduce CPU load on HLT (or 3D CDC HW trigger).

Estimated reduction : 1/10



* Need a close collaboration with Comp/Soft group.
  - Not only on software, but also on HLT architecture (ex. access to constant database, etc.)
  => We will have a discussion at Comp/Soft WS in June.

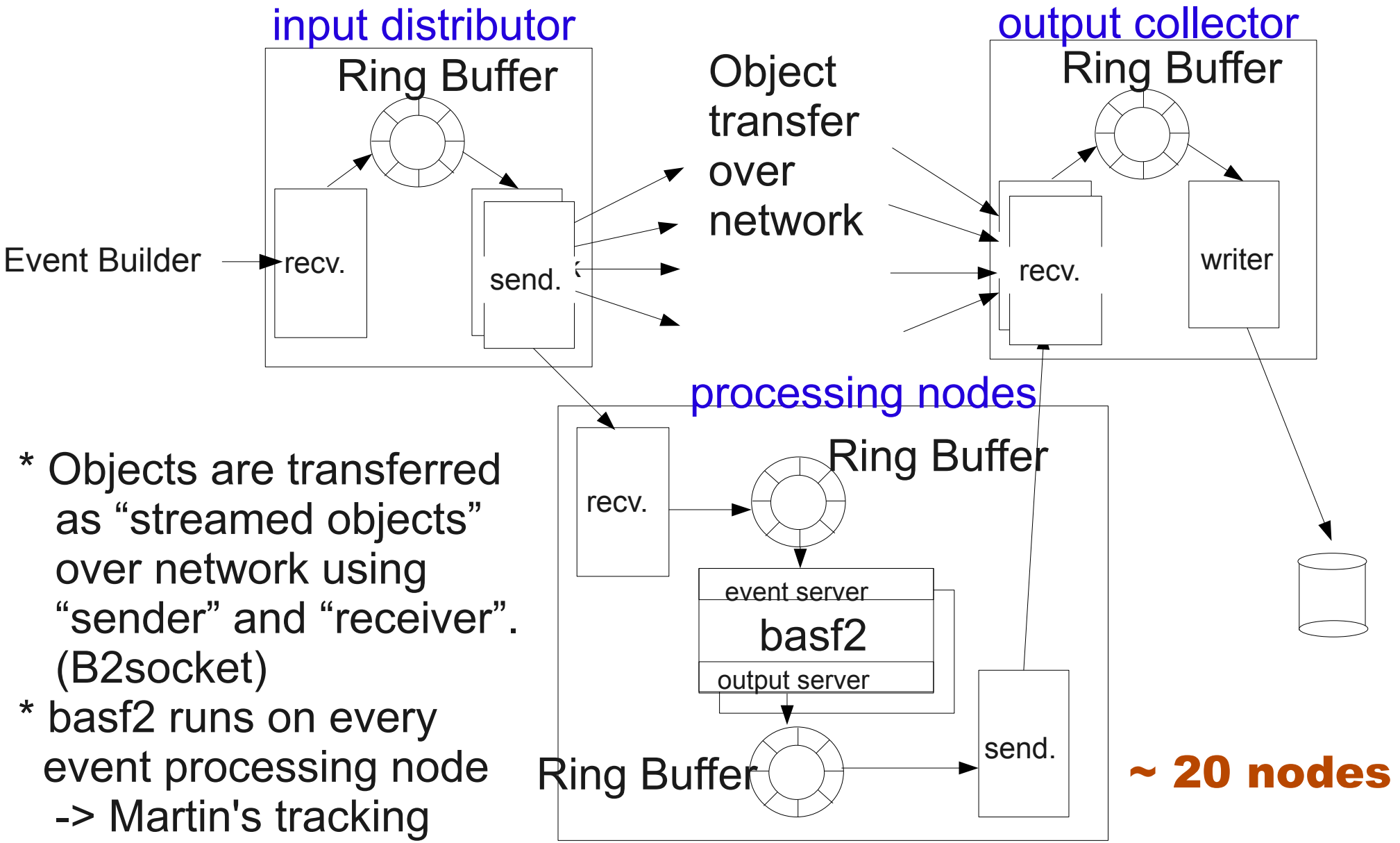
* The processing latency is a critical issue for PXD integration to feed bac reconstruted track informations to PXD readout processor.

3. Requiments to HLT-PXD interface

- Expected function of PXD readout box
  * Buffer PXD data flow as long as HLT decision latency.
    -> Assuming 2% occupancy and 30kHz trigger late
      => Buffer size = 600MB/sec * (HLT latency) for one DHH.
  * Receive event tag and track parameters (or association region) from HLT and perfom noise reduction
  * Send associated hits to 2nd level event builder.

- Additional works required for HLT
  * Software to quarry evtag/track parameters from main data flow.
  * Additional data flow for PXD splitted from the main stream.
  * Mechanism to send the data flow to PXD readout box.

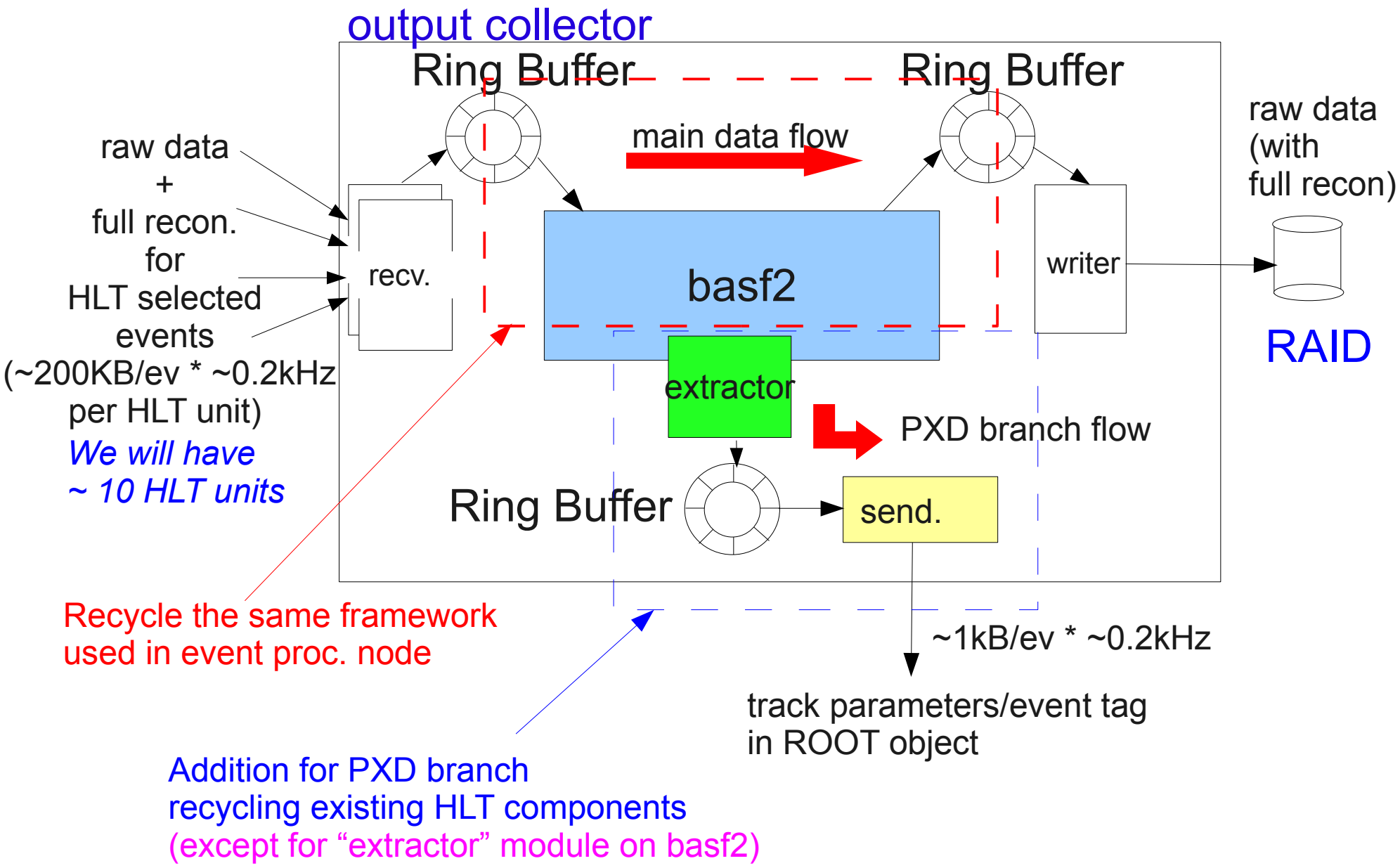
Software on HLT (1 unit) : Original design

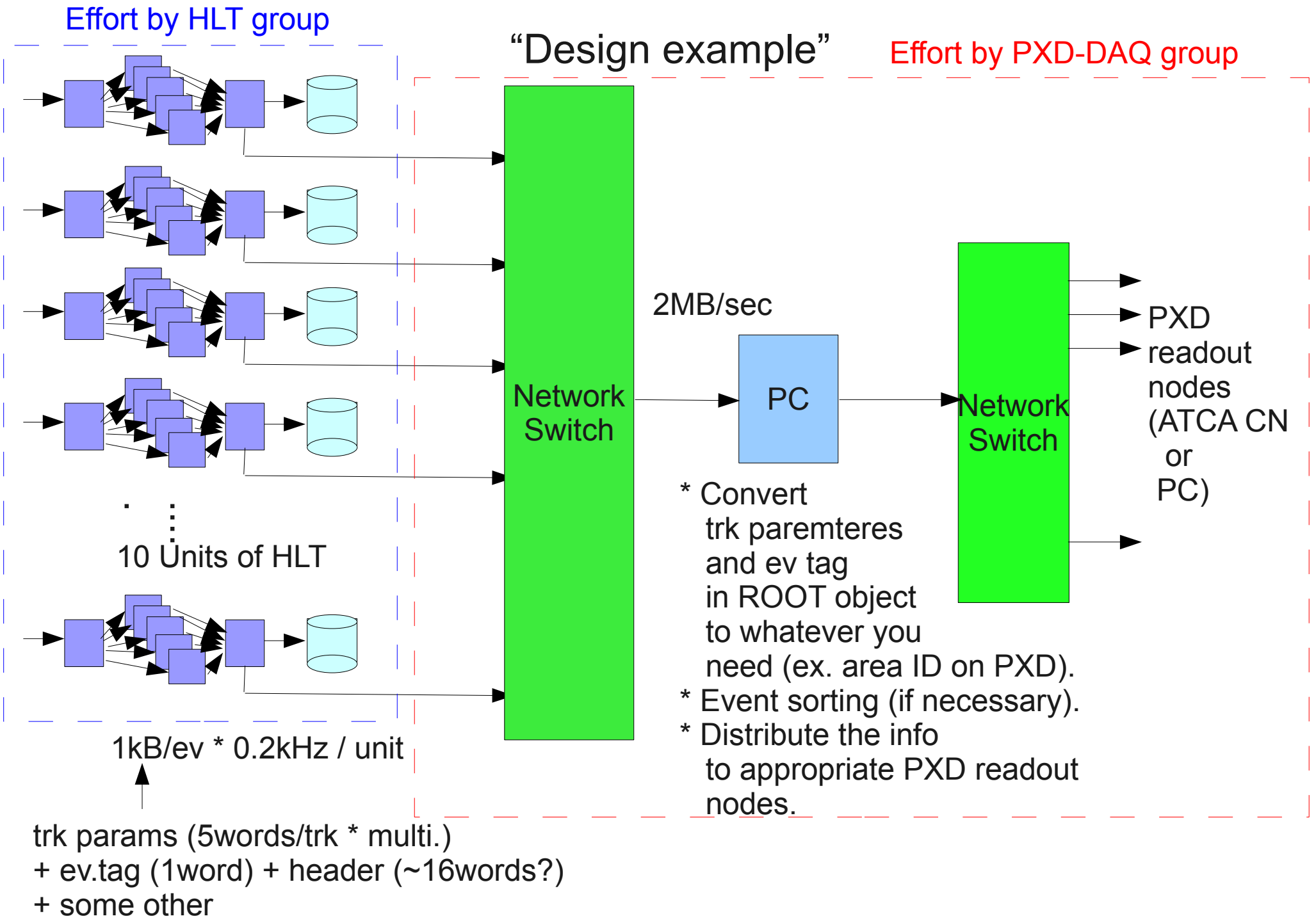


* Objects are transferred as “streamed objects” over network using “sender” and “receiver”. (B2socket)
* basf2 runs on every event processing node
  -> Martin's tracking framework runs “as is”.

~ 20 nodes

Modification for PXD data stream




Effort by HLT group
"Design example"
Effort by PXD-DAQ group
Network Switch
2MB/sec
PC
Network Switch
PXD readout nodes (ATCA CN or PC)
10 Units of HLT
* Convert trk paremteres and ev tag in ROOT object to whatever you need (ex. area ID on PXD).
* Event sorting (if necessary).
* Distribute the info to appropriate PXD readout nodes.
1kB/ev * 0.2kHz / unit
trk params (5words/trk * multi.)
+ ev.tag (1word) + header (~16words?)
+ some other

HLT latency

* The design of HLT->PXD interface heavily depends on the HLT latency, in particular, the buffering depth for PXD data flow.

* Current assumption is “5 sec. at most”.

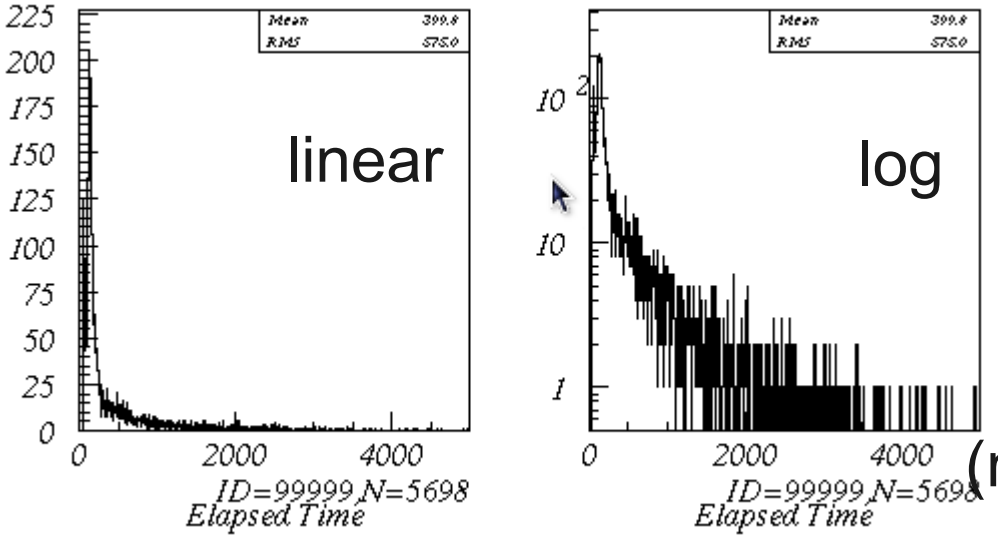
* With the assumptions of
  - Typical occupancy : 2%
  - Maximum L1 rate : 30kHz
    => Data flow per DHH = 600 MB/sec

  The buffer depth is required to be
  600MB/sec * 5 sec = 3GB per DHH.

* Considering the safety margin of ~50%, the buffer size should be ~5GB for a single DHH.
  -> Resulting in 5GB * 40 DHHs = 200GB in total.

Estimation of HLT latency

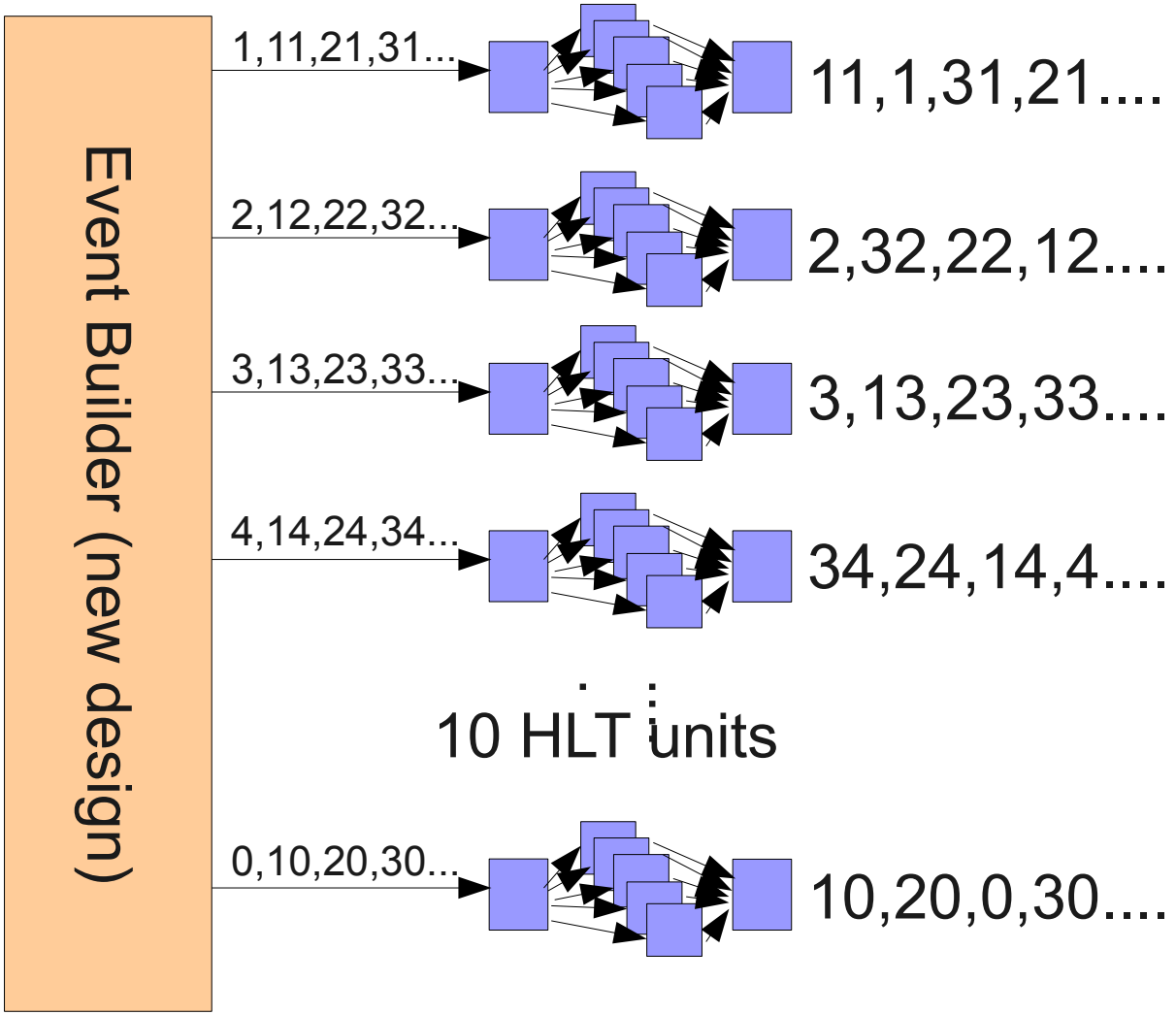
- Measurement using Belle's RFARM(=HLT) with current Belle reconstruction code.
- The processing time for full event reconstruction (incl. both full tracking + energy clustering) is measured for "L4 passed" events. (Exp.57, ~5000 events)



3.2GHz
Intel Xeon,
e-time / core.

- 5 sec. latency seems to be a reasonable assumption even though we take into the account the possibility of longer reconstruction time (~50% slower, for example.)
- 2.6 % of events takes more than 5 sec.
  -> Under investigation by Iwasaki-san
  Could be "junk events"?

Event Disordering



* HLT processing is fully event-by-event parallel.
  -> Event sequence is disorderd at the output of HLT.

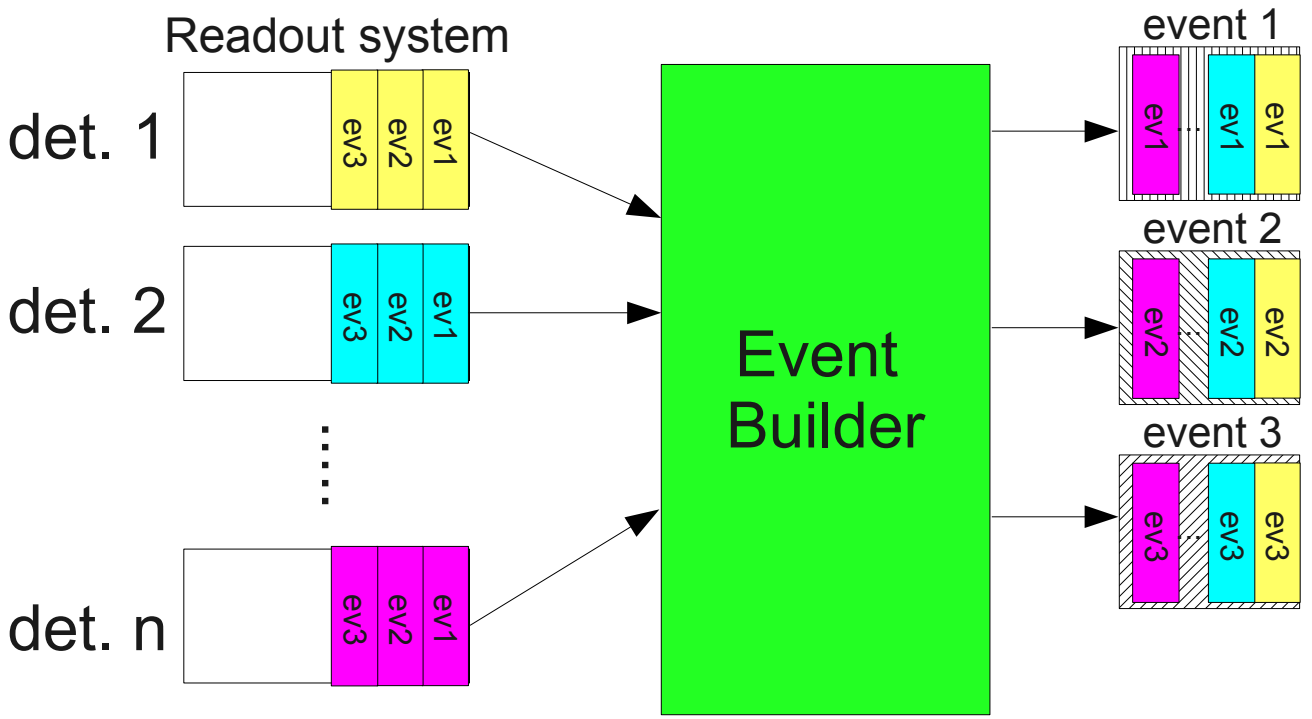
* “Sorting” might be necessary for the event matching at PXD readout.
  -> needs extra latency.

evtno = mod(evt,10)

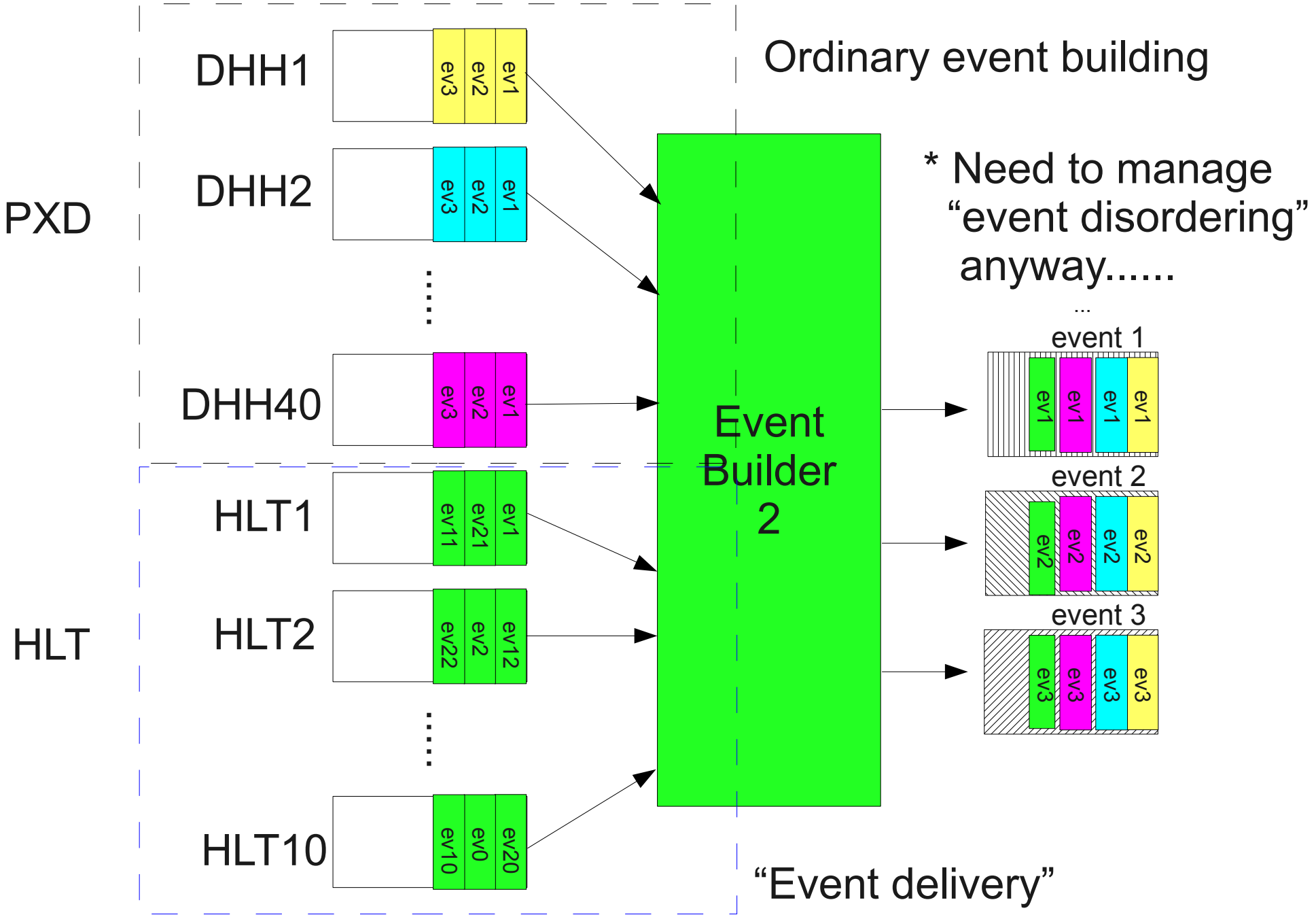
4. Other considerations on PXD data flow

- 2nd level event building is not so trivial.

Usual event building:



2nd level Event Builder



The input data format for Event Builder 2

- The data from HLTs are streamed ROOT objects so that they can be directly written to RAID through EVB2.

- If we expect the same functionality for PXD readout, the data fed int EVB2 are expected to be formatted in streamed ROOT object.
  - Option 3 (PC solution):
    - * Straight-forward. Just run ROOT-application there (even BASF2 can be used for this purpose).
  - Option 2: (ATCA CN)
    - * Data formatting is performed by FPGA code (HDL).
      -> Possible to convert to ROOT?

- If not, formatting is required to be performed on recording nodes.