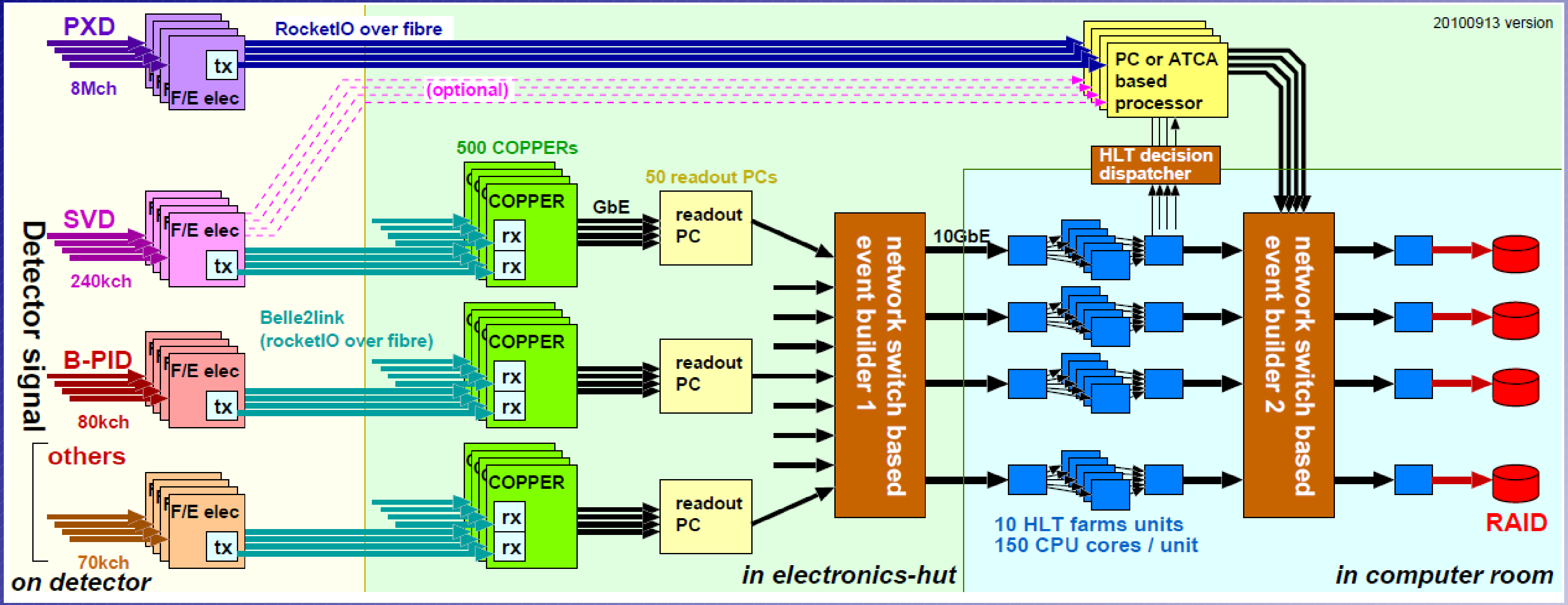
PXD<->DAQ interface

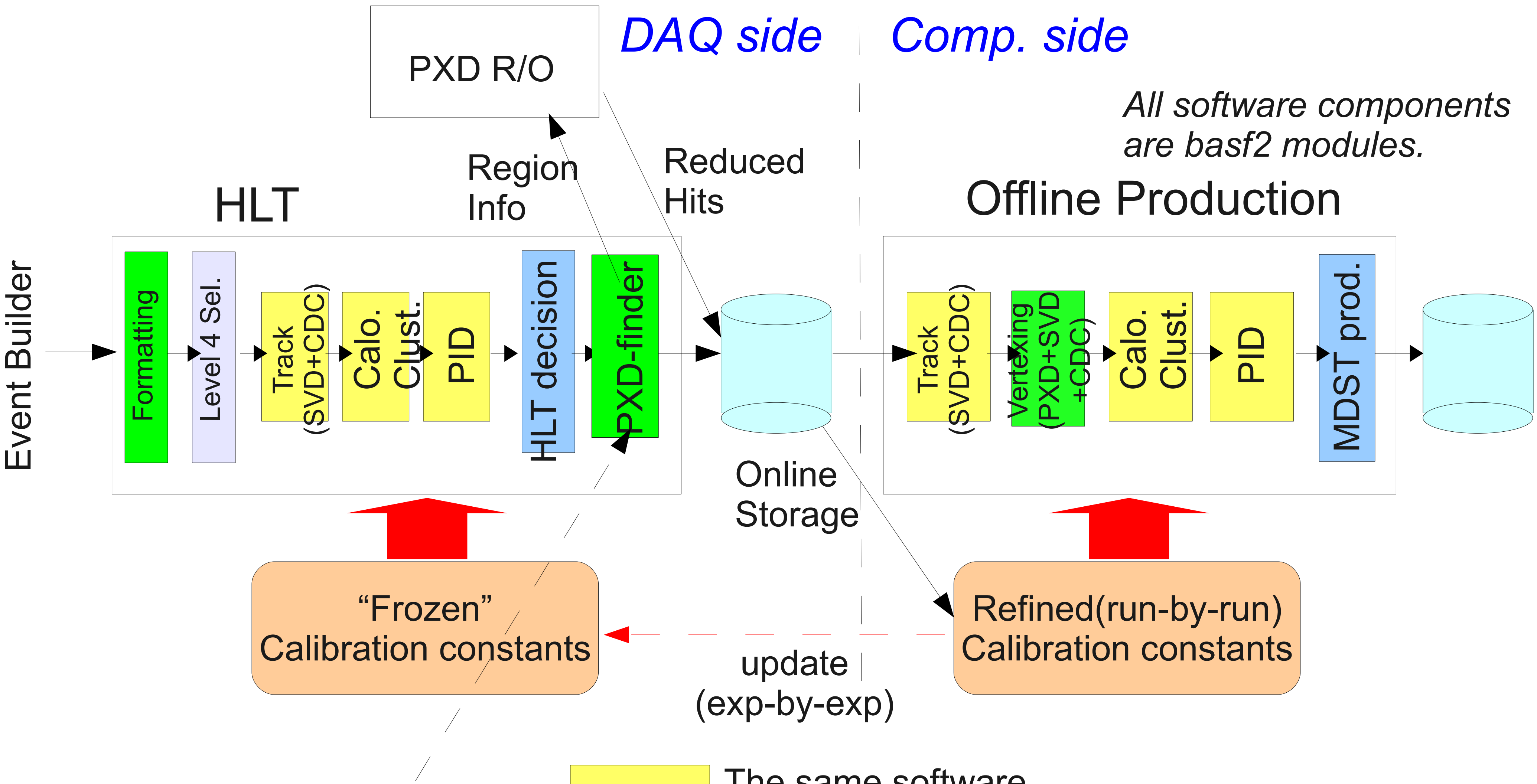
R.Itoh, KEK

Integration of Pixel Detector (PXD) readout



* Track parameters are obtained in High Level Trigger and sent to PXD readout processor.
* Only the hits associated with tracks in HLT-passed events are sent.

Event Reconstruction Chain at Belle II



🟨 The same software between HLT and offline

Special track finder for track-PXD hit association

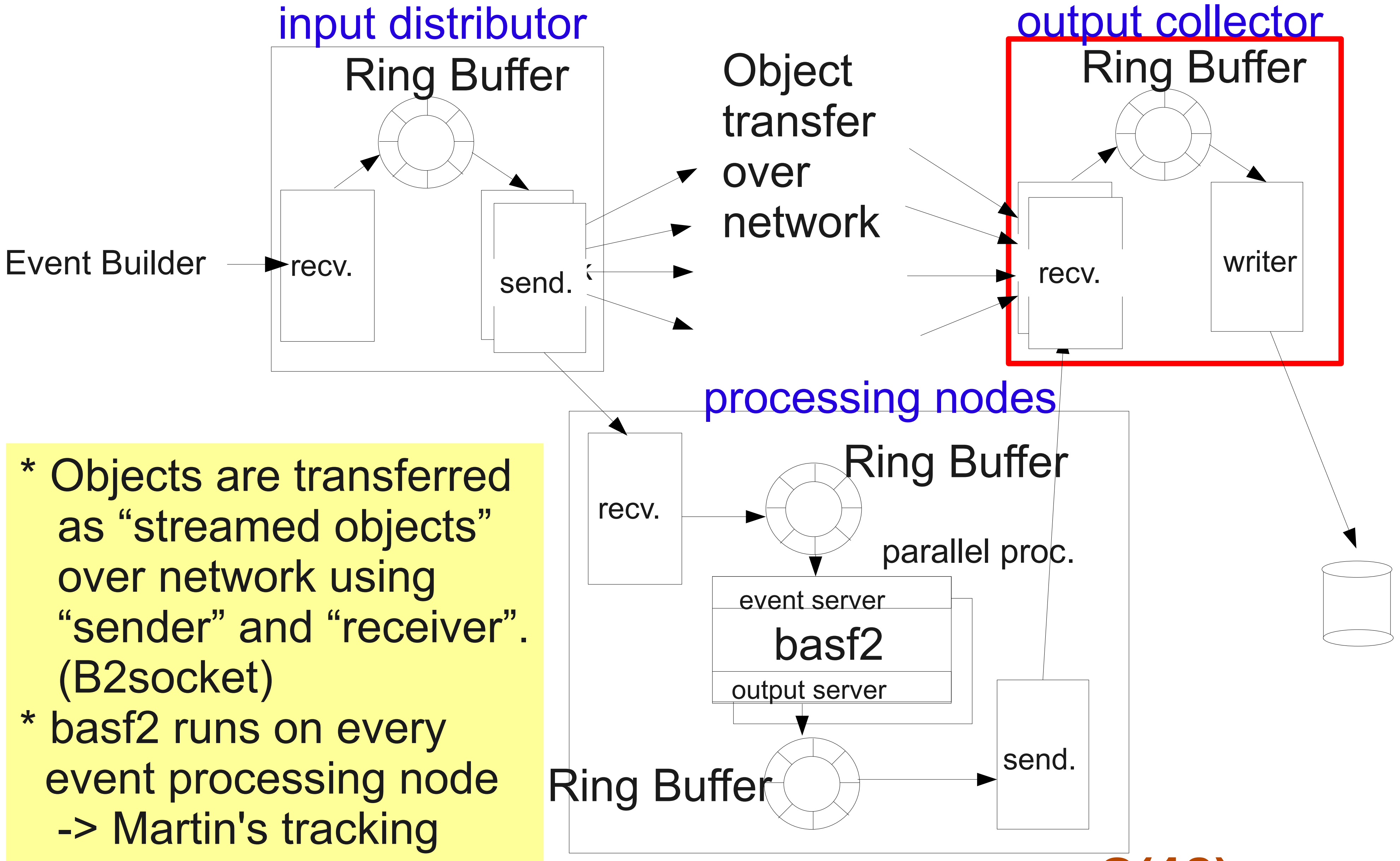
* Executed only for the HLT-taken events

* Can make use of full SVD raw data with other rawdata/HLT results

* Can be coded in C++ under Belle II software environment

already parallel in 2000 cores

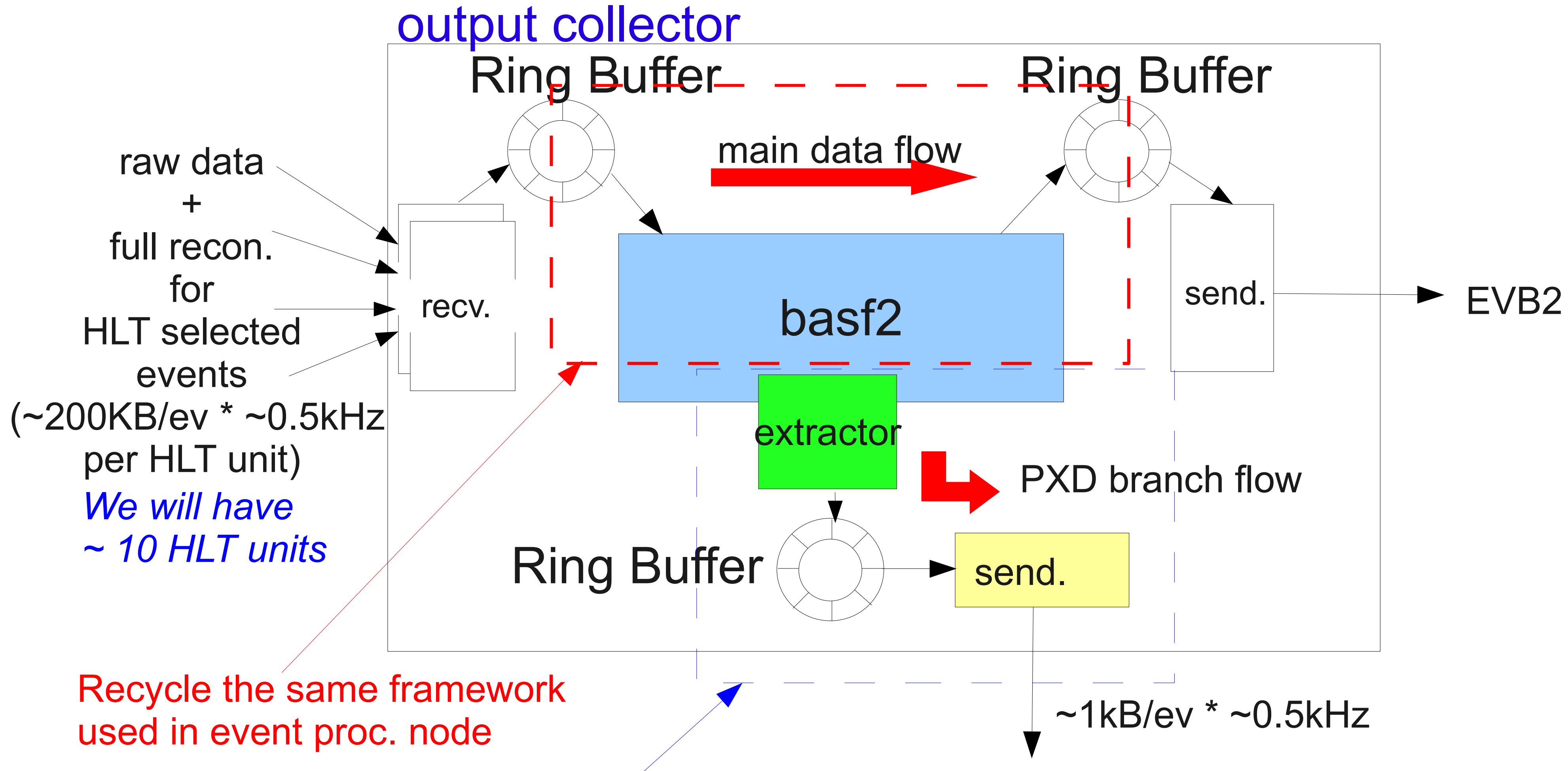
Software framework of HLT (1 unit) = pbasf2 for PC cluster



- * 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”.

~ O(10) cores x 20 nodes

Modification for PXD data stream



raw data
+
full recon.
for
HLT selected
events
(~200KB/ev * ~0.5kHz
per HLT unit)

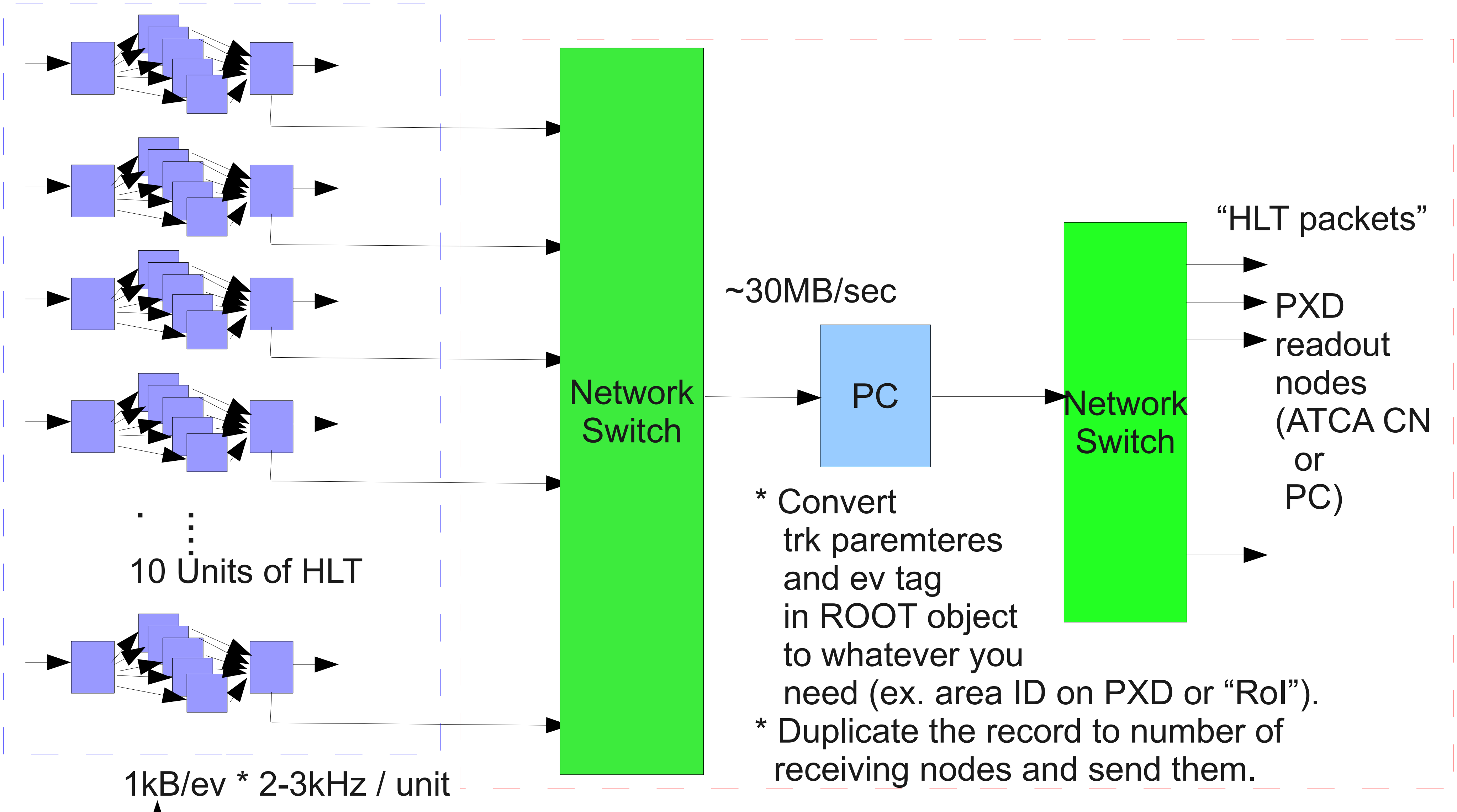
We will have
~ 10 HLT units

Recycle the same framework
used in event proc. node

HLT packet containing
- event tag
- flag of “take” or “discard”
- Region ID for hit association (for “take”)

Addition for PXD branch
recycling existing HLT components
(except for “extractor” module on basf2)

HLT-PXD interface



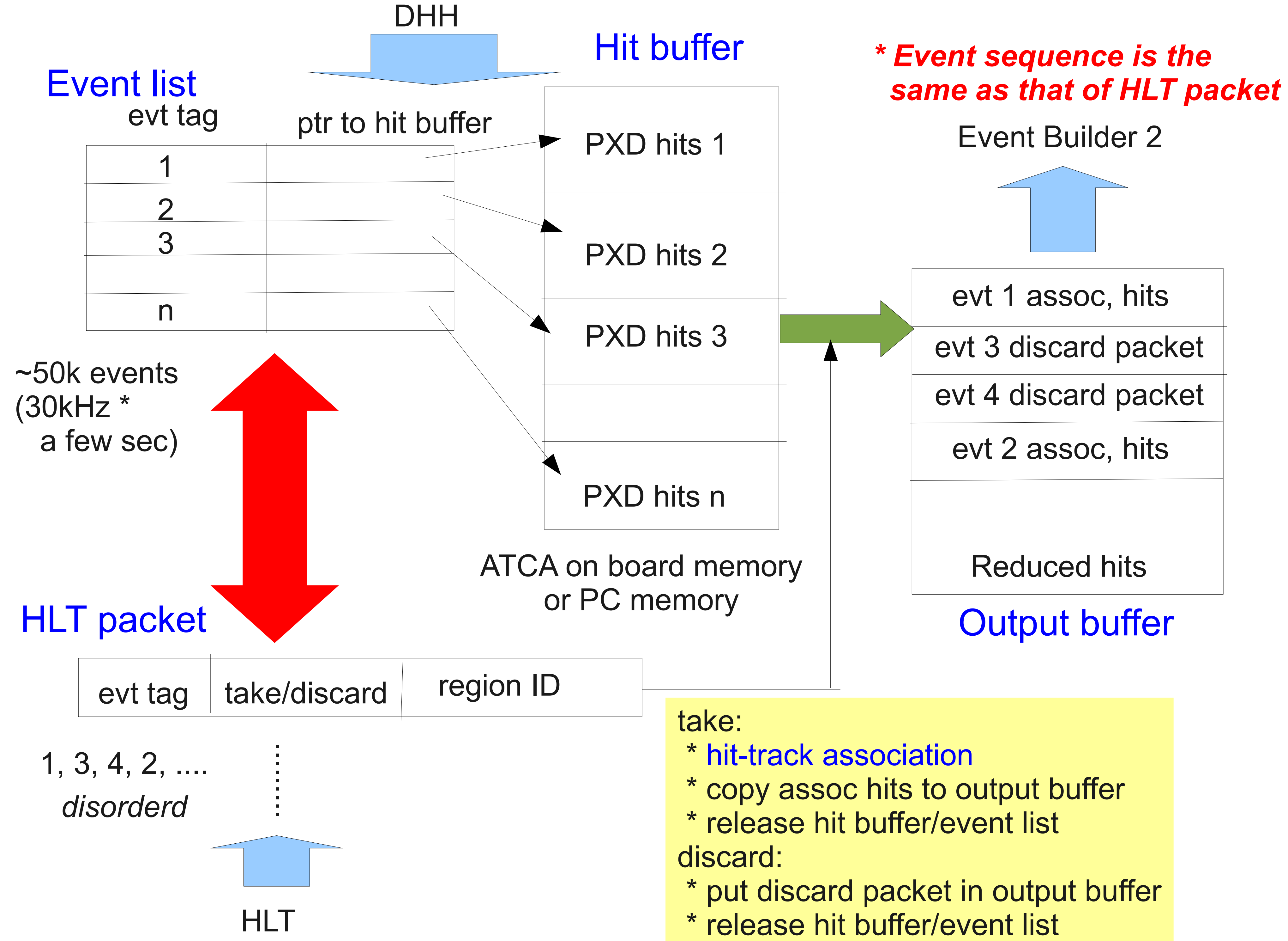
* Convert trk paremteres and ev tag in ROOT object to whatever you need (ex. area ID on PXD or "RoI").

* Duplicate the record to number of receiving nodes and send them.

"HLT packets"
Passed events:
evtag + track parameters
Discarded events:
evtag + "discarded" flag

- Event sequence of HLT packets are disordered because of parallel processing.
  => Could be a problem in further processing (track-hit matching, 2nd level event building)

Processing framework in PXD R/O box



DHH

Event list

evt tag	ptr to hit buffer
1	
2	
3	
n	

Hit buffer

- PXD hits 1
- PXD hits 2
- PXD hits 3
- PXD hits n

ATCA on board memory or PC memory

*** Event sequence is the same as that of HLT packet***

Event Builder 2

Output buffer

- evt 1 assoc, hits
- evt 3 discard packet
- evt 4 discard packet
- evt 2 assoc, hits
- Reduced hits

~50k events (30kHz * a few sec)

HLT packet

evt tag	take/discard	region ID

1, 3, 4, 2, ...

disorderd

HLT

take:

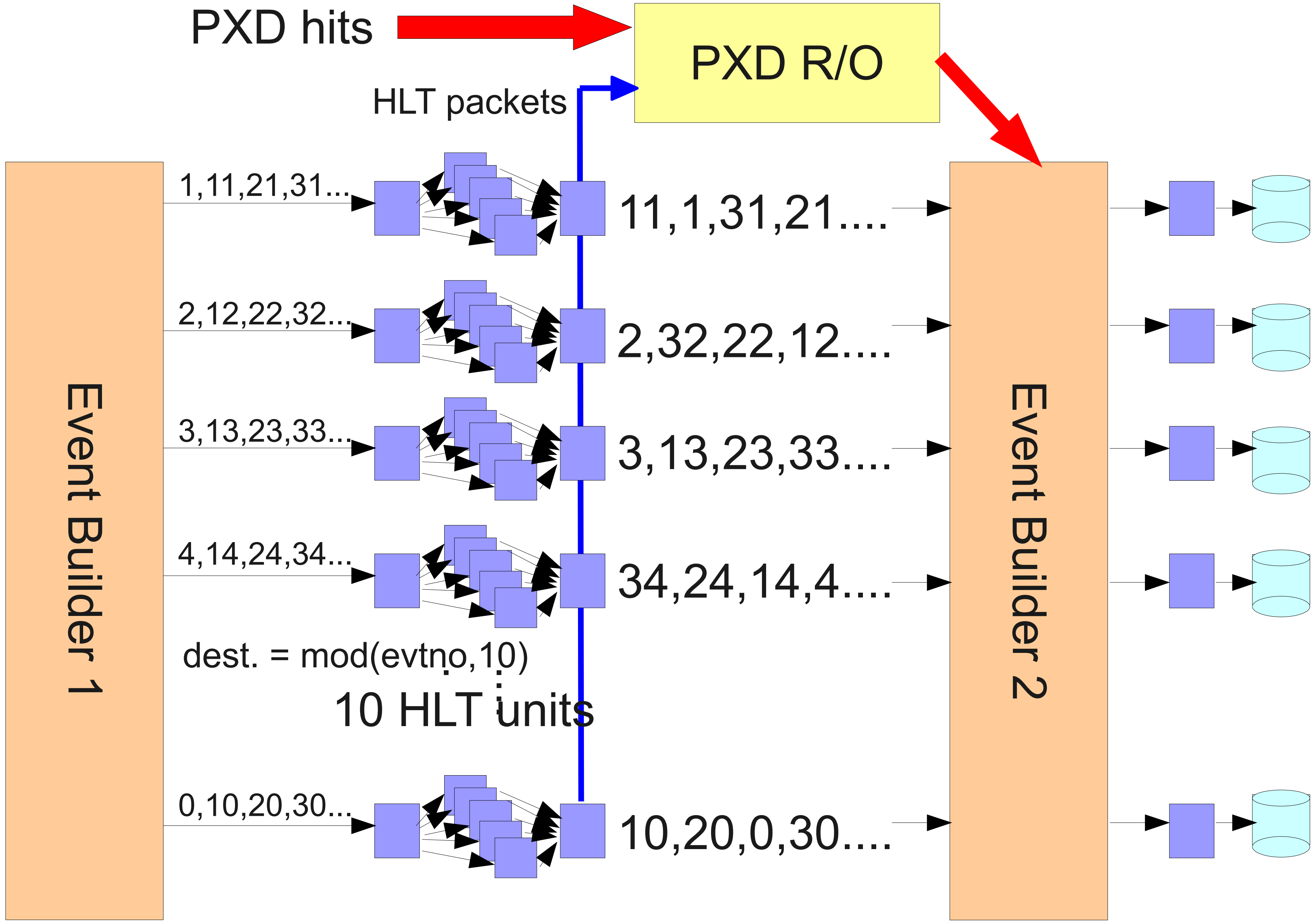
- * hit-track association
- * copy assoc hits to output buffer
- * release hit buffer/event list

discard:

- * put discard packet in output buffer
- * release hit buffer/event list

Idea on 2nd level event building (EVB2)

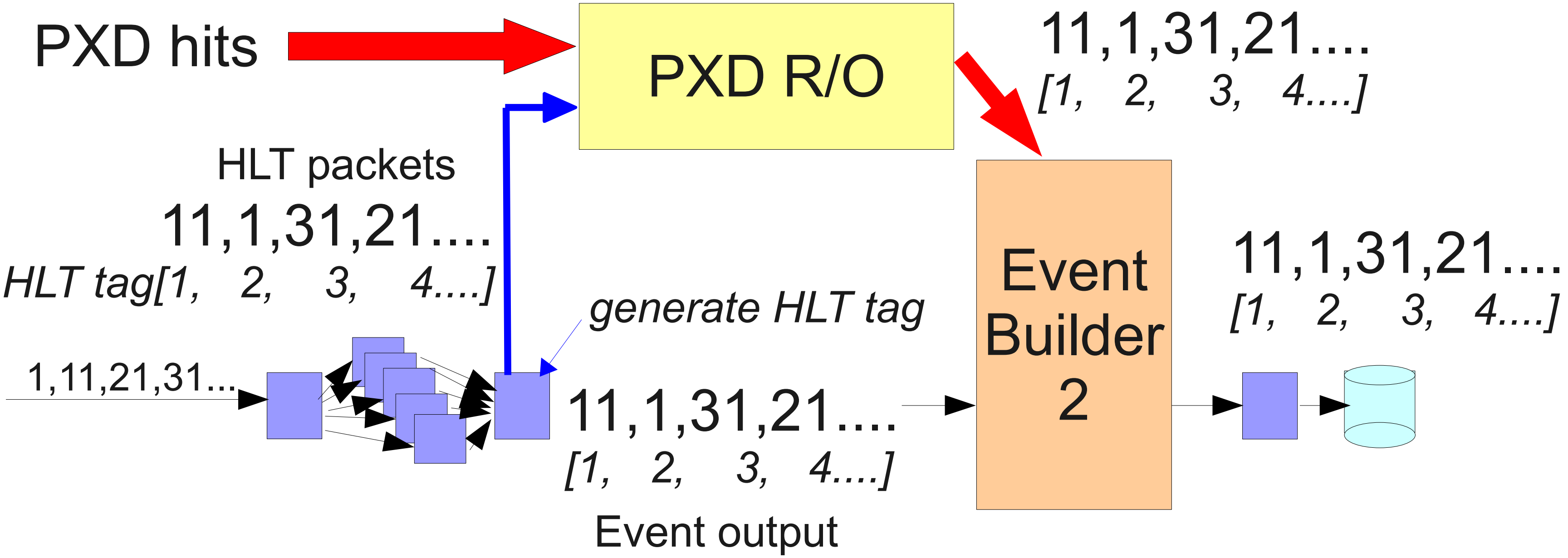
- Event sequence from HLT is disordered.
- Do we need to "sort" events for the 2nd level event building?

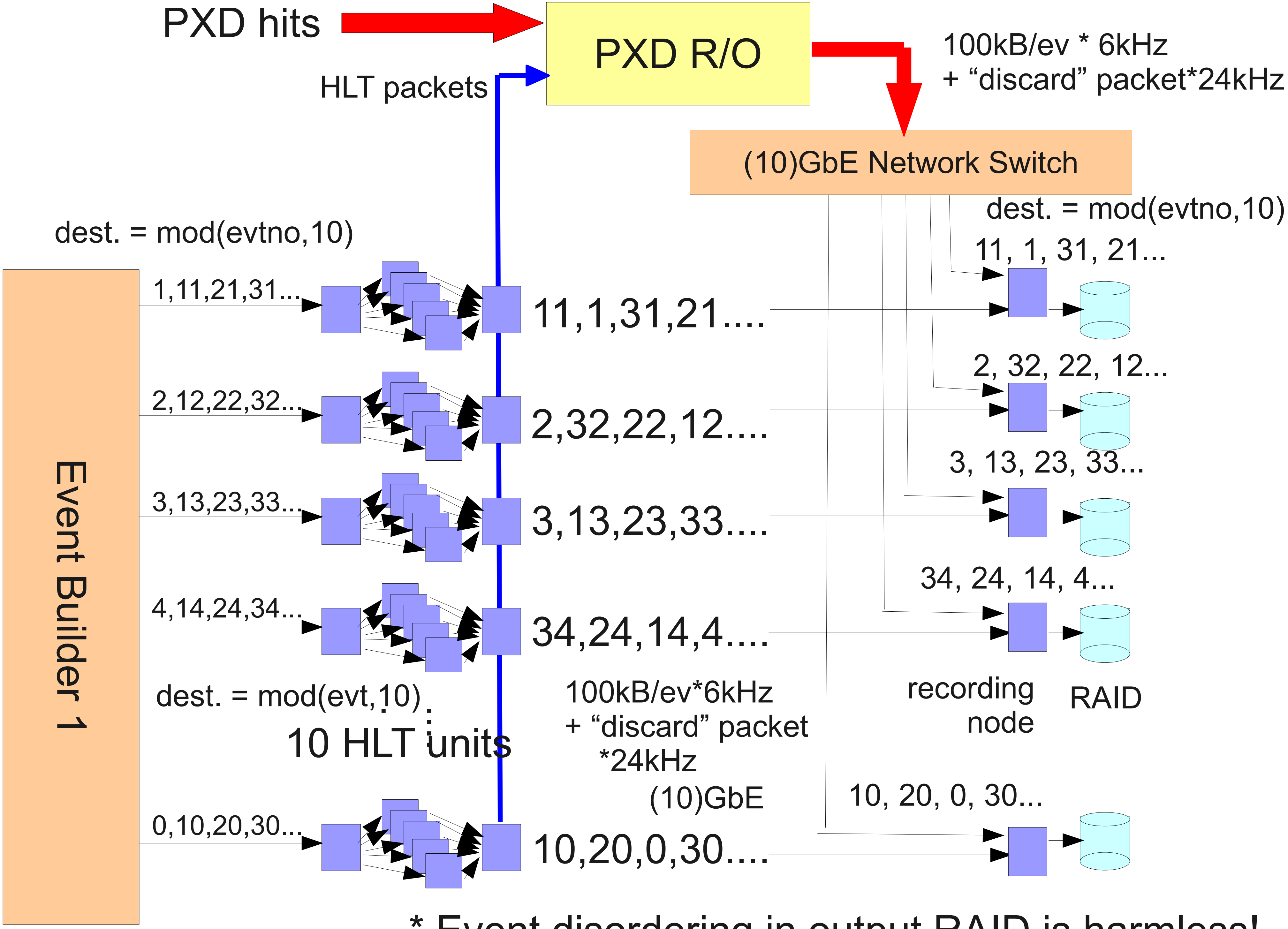


- HLT sends the event output to EVB2 and HLT packets to PXD R/O in the same order, although event tag is disordered.
  - -> Idea: place new "HLT tag" both in "event output" and "HLT packet" and build event using the tag at EVB2.

- Processing on PXD R/O does not break the ordering of HLT packets.

→ No sorting is required although the event tag is disordered in recording RAID.





PXD hits

PXD R/O

HLT packets

100kB/ev * 6kHz
+ “discard” packet*24kHz

(10)GbE Network Switch

dest. = mod(evtno,10)

dest. = mod(evtno,10)

Event Builder 1

1,11,21,31...

2,12,22,32...

3,13,23,33...

4,14,24,34...

0,10,20,30...

11,1,31,21....

2,32,22,12....

3,13,23,33....

34,24,14,4....

10,20,0,30....

11, 1, 31, 21...

2, 32, 22, 12...

3, 13, 23, 33...

34, 24, 14, 4...

10, 20, 0, 30...

dest. = mod(evt,10)

10 HLT units

100kB/ev*6kHz
+ “discard” packet
*24kHz

(10)GbE

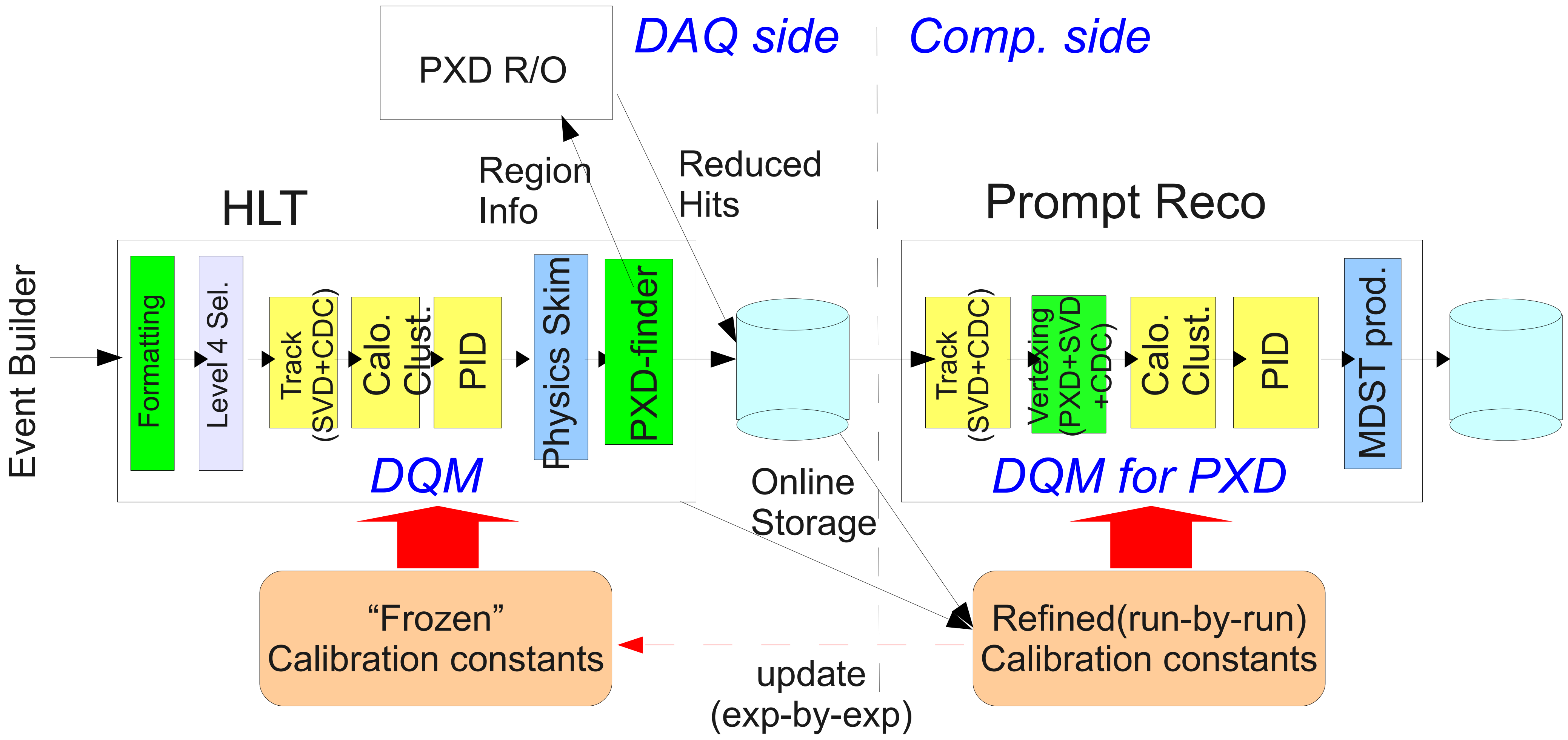
recording
node

RAID

* Event disordering in output RAID is harmless!

DQM for PXD by prompt reco.

🟨 The same software between HLT and offline



* Idea is, to run offline reconstruction instead of raw data copy right after a run is completed.

* DQM modules can be inserted in Prompt Reco script.

* Since full event reconstruction with PXD is supposed to be done in prompt reco, detailed monitoring of PXD data quality even at physics level becomes possible.

Discussion

- HLT is supposed to provide HLT trigger decision with track information for PXD readout box in any cases of PXD R/O options (ATCA(baseline and challenge), and PC).

- PXD-specialized tracking/RoI generation can be done on HLT using full SVD raw data + other detector info + HLT procesing results. -> very versatile. Standard C++ coding under Belle II env.

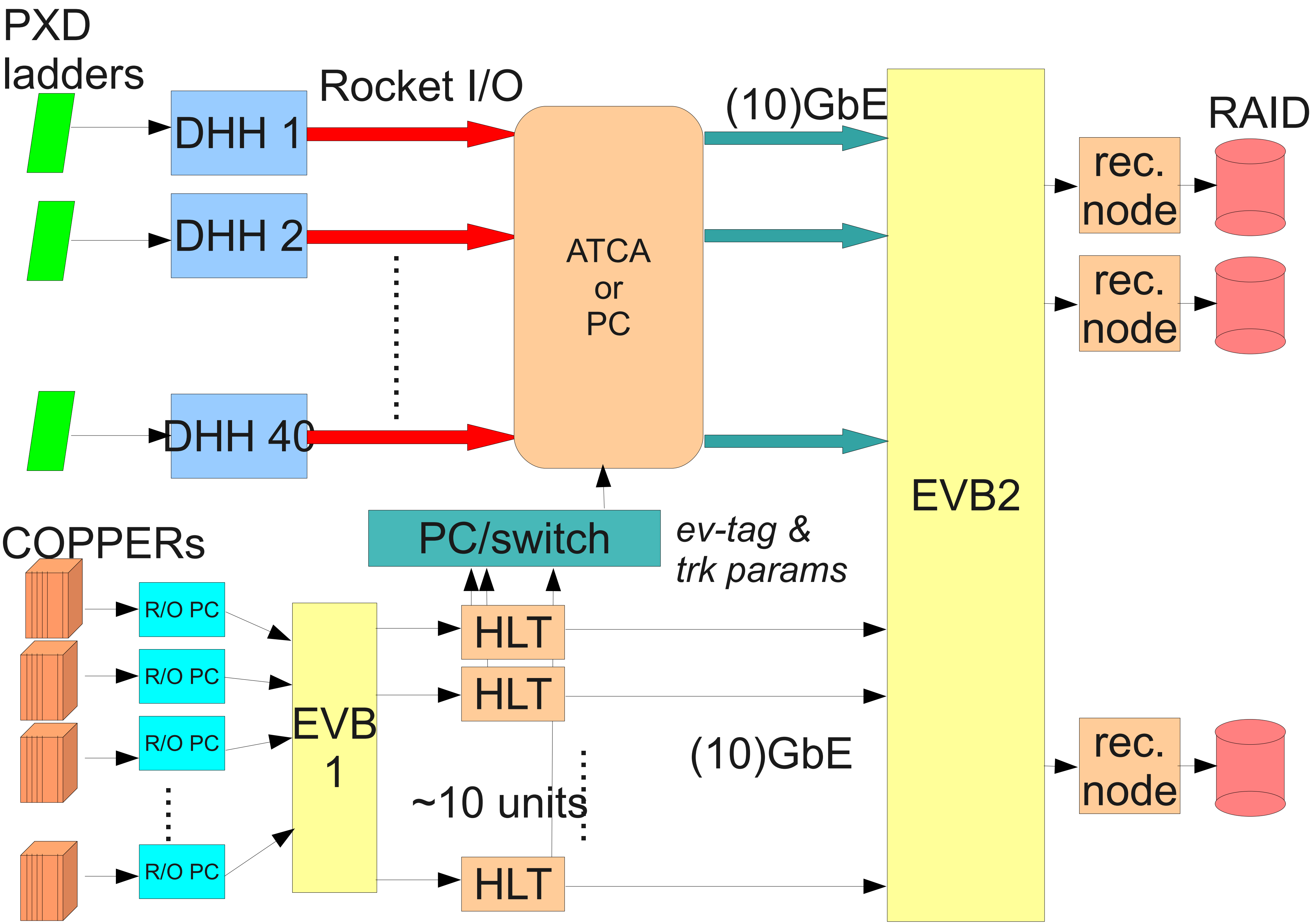
- The “event disordering problem” can be solved by considering the algorithm of event matching with HLT packets in PXD readout box. -> Is it OK?

- Regardless of having ATCA based tracking or not, HLT decision is fed into PXD readout box and it is supposed that all PXD data matched with HLT tag are sent to 2nd level event builder.

- How do we implement DQM for PXD?
  * Is it possible to implement the function in ATCA board?
  * Full event monitoring after 2nd level event building is difficult.
  * Prompt Reco is required for detailed monitoring.

Backup Slides

PXD integration



Do we really need 4GB/DHH for data buffering?

- Average HLT latency for an event is expected to be less than 1 sec as shown in previous meeting.

- Thanks to parallel processing, the HLT packets for events with shorter processing latency come faster. Not necessary to wait for the 5 sec latency for one particular problematic event.

- The PXD hit buffer is supposed to be released immediately after the hit-track association (if the algorithm shown in previous slide is used).

We don't have to buffer every event for 5sec.

* It seems the required buffer size is much smaller than 30kHz*5sec*600MB/sec.

<- This size is necessary only when 5 sec latency is consumed by all of ~2000 cores in HLT at one time. ==> very rare!

* We need a simple MC study to estimate the actual size.

* Sophisticated memory management scheme is required on FPGA in case of Option 12, anyway.