Improved HLT Framework for Belle II experiment

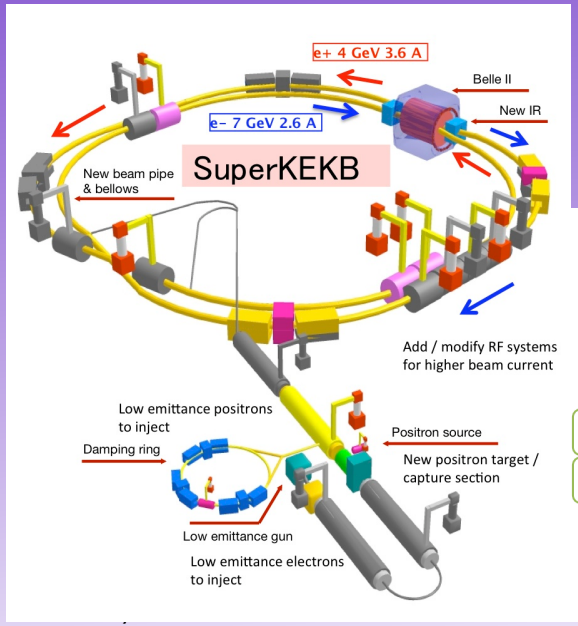
Ryosuke Itoh (KEK), Anselm Baur (DESY), Seokhee Park(KEK)
and
M.Nakao, S.Yamada, S.Y.Suzuki,
T.Kunigo, D.Levit
(KEK Belle II DAQ team)

CHEP2023, Norfolk, May 9th, 2023

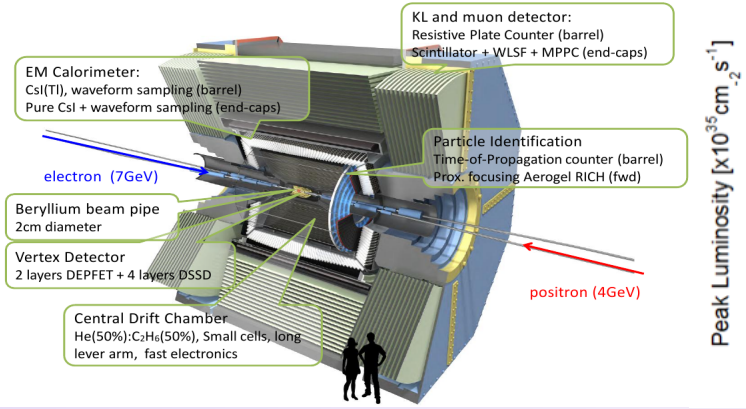
The Belle II experiment

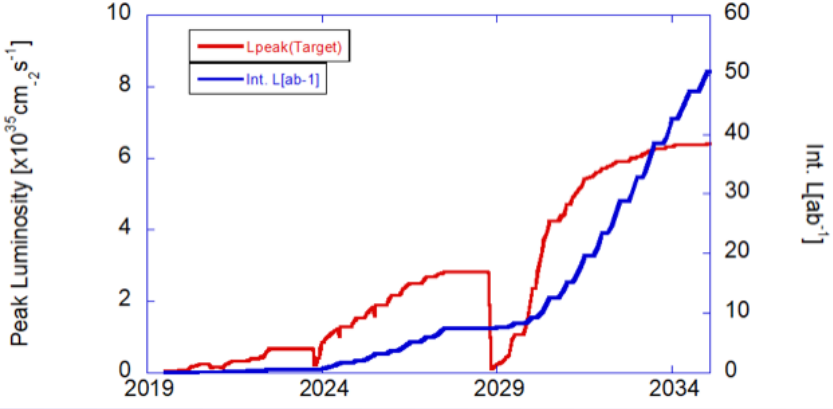
- The Belle II experiment is a B-factory experiment at KEK in Japan aiming at the discovery of New Phyisics in B meson decays.

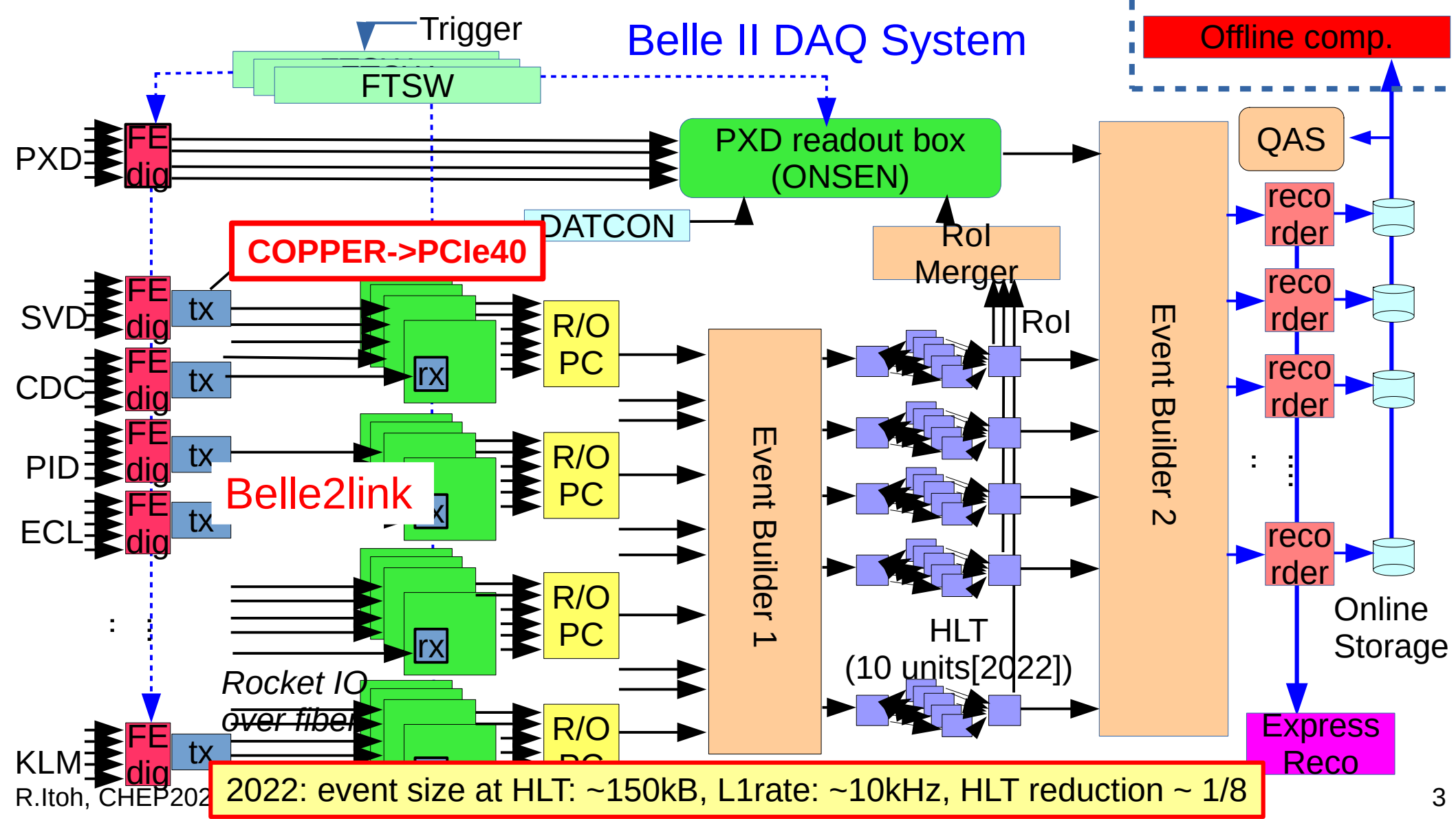
- The SuperKEKB accelerator is designed to achieve the world's highest luminosity of L = $6 \times 10^{35} cm^{-2} sec^{-1}$ and to accumulate integrated luminosity of $50 ab^{-1}$ which corresponds to a 10^{11} B meson sample.



- Now we are in the 1.5 year long shutdown period for the installation of full Pixel Detector.
- The data taking will resume in Dec.2023.



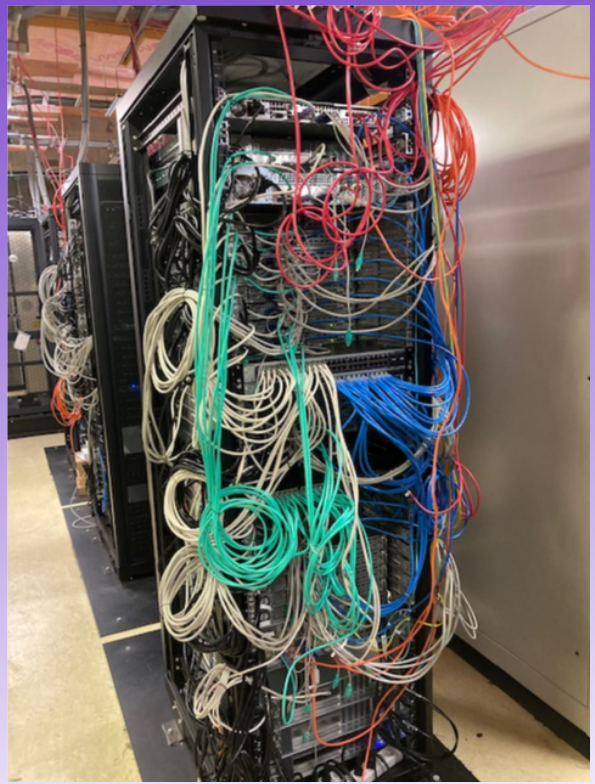



Belle II DAQ System
Trigger
FTSW
Offline comp.
PXD
FE dig
PXD readout box (ONSEN)
DATCON
COPPER->PCIe40
RoI Merger
QAS
SVD
CDC
PID
ECL
tx
rx
R/O PC
Belle2link
RoI
Event Builder 1
Event Builder 2
reco rder
HLT
(10 units[2022])
Online Storage
Rocket IO over fiber
KLM
Express Reco
2022: event size at HLT: ~150kB, L1rate: ~10kHz, HLT reduction ~ 1/8

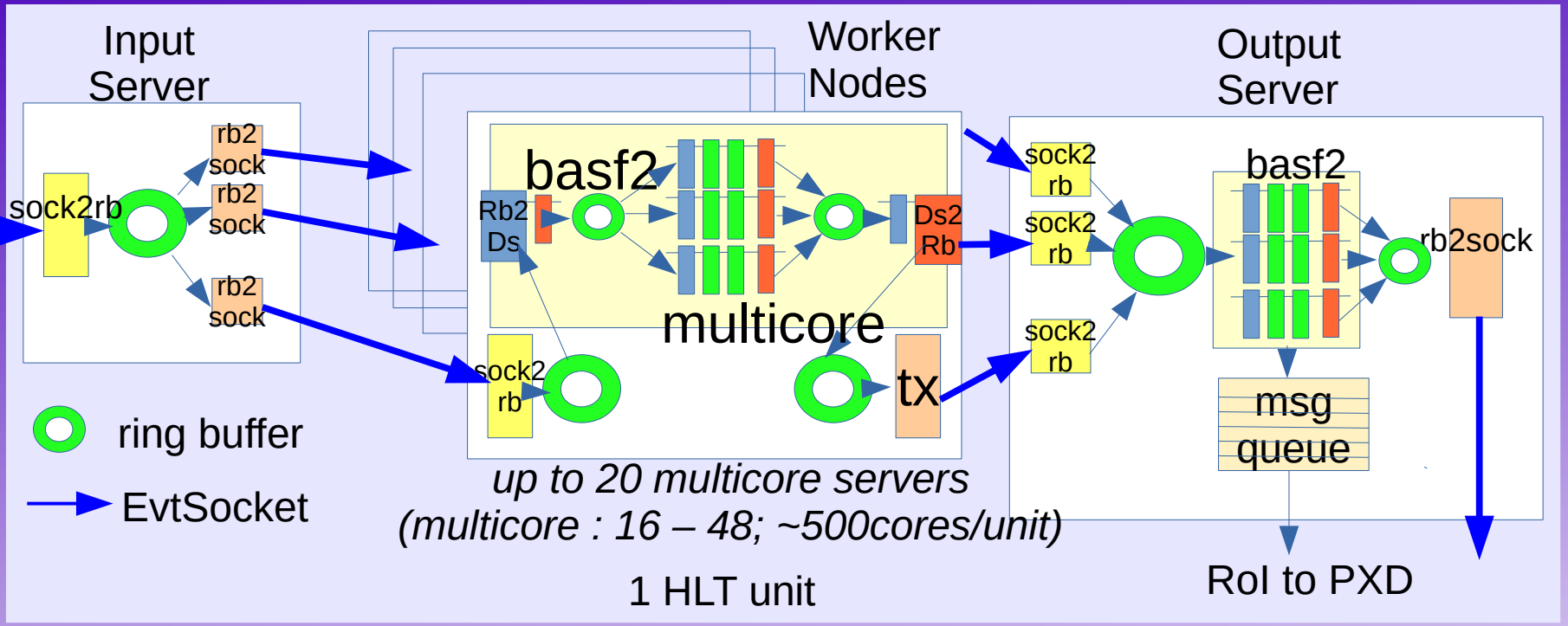
HLT hardware

- 10 units in operation (4800 cores)
  + 3 more units under preparation (-> 6400 cores)

- All assembled by ourselves



Original HLT framework before 2020



- “basf2” is the same Belle II offline processing framework which has the “module-and-path” architecture featuring the “multi-process” event-by-event parallel processing capability.
- Parallel processing is implemented using the home-grown Ring Buffer based on Linux IPC (SharedMem+Semaphore).

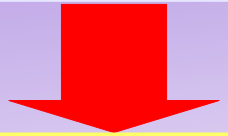
Problems in RingBuffer HLT and Transition to ZMQ-HLT

- Unexpected “Locking problem” of IPC (shared mem/semaphore) used in the ring buffer.

- Observation of cases that IPC is not removed properly at run stop/system abort.

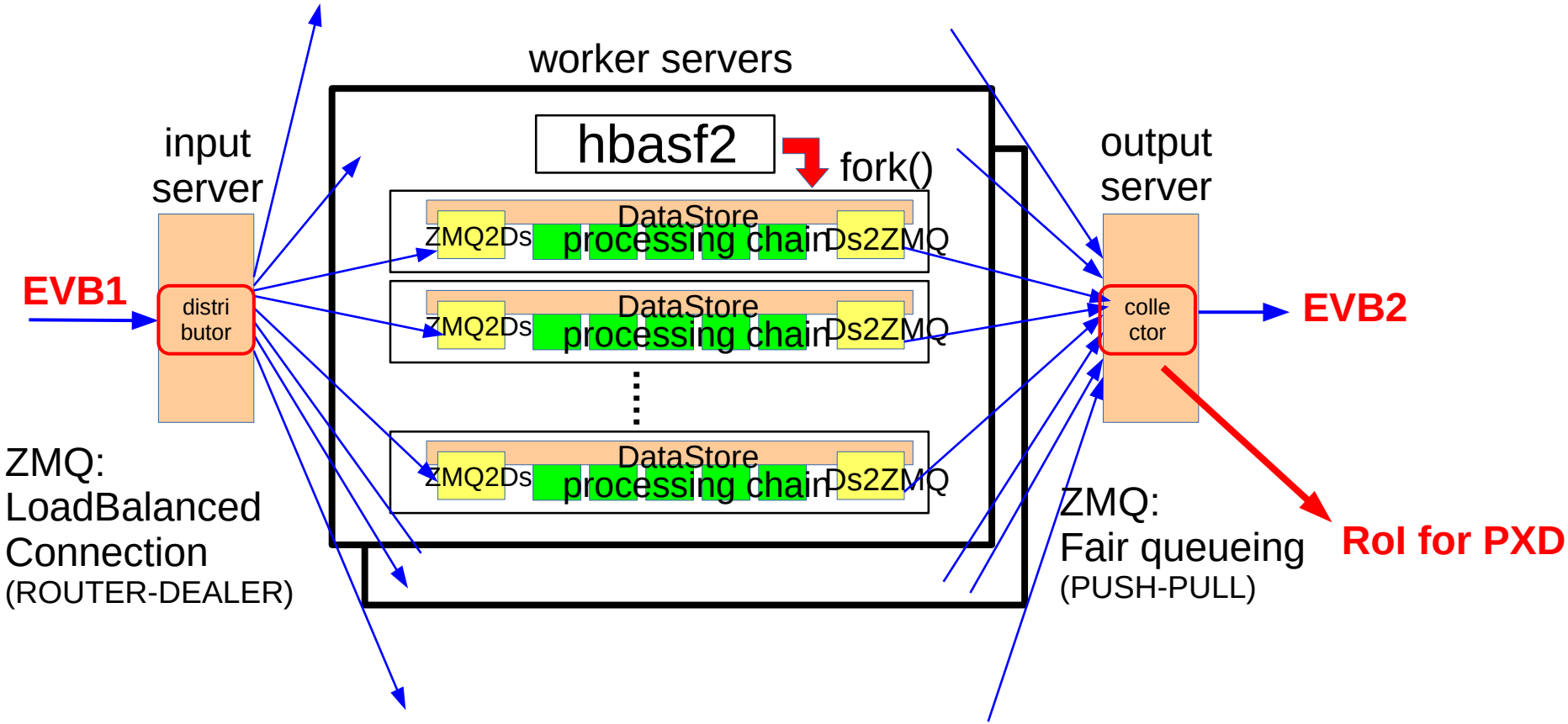
- RoI extraction for PXD requires the object destreaming which requires high CPU consumption in the output server.

- Long initialization time at run start caused by geometry initialization.



New framework with a “lock-free” data transport : ZeroMQ

First ZeroMQ based HLT framework (from 2020)



New core framework “hbasf2” was introduced replacing basf2

Problems in the first ZMQ-HLT framework

- When one of the event process dies in the middle of processing (ex. seg fault), the event is lost and the process is not restarted.
  -> * Source of “missing events/RoIs”
  -> *Possible systematics in the output events*
  * Processing power is lost and not recovered.

- “hbasf2” cannot be invoked offline as a stand alone application with the parallel processing turned on. This makes the offline debugging of HLT processing difficult for rare troubles.
  <- Original basf2 could do this, but it is based on the IPC ring buffer.

- Too many ZMQ connections from each worker process to single input or output server (~1000!).

Further Improvement

- Keep the framework outside workers unchanged (ZMQ-HLT).

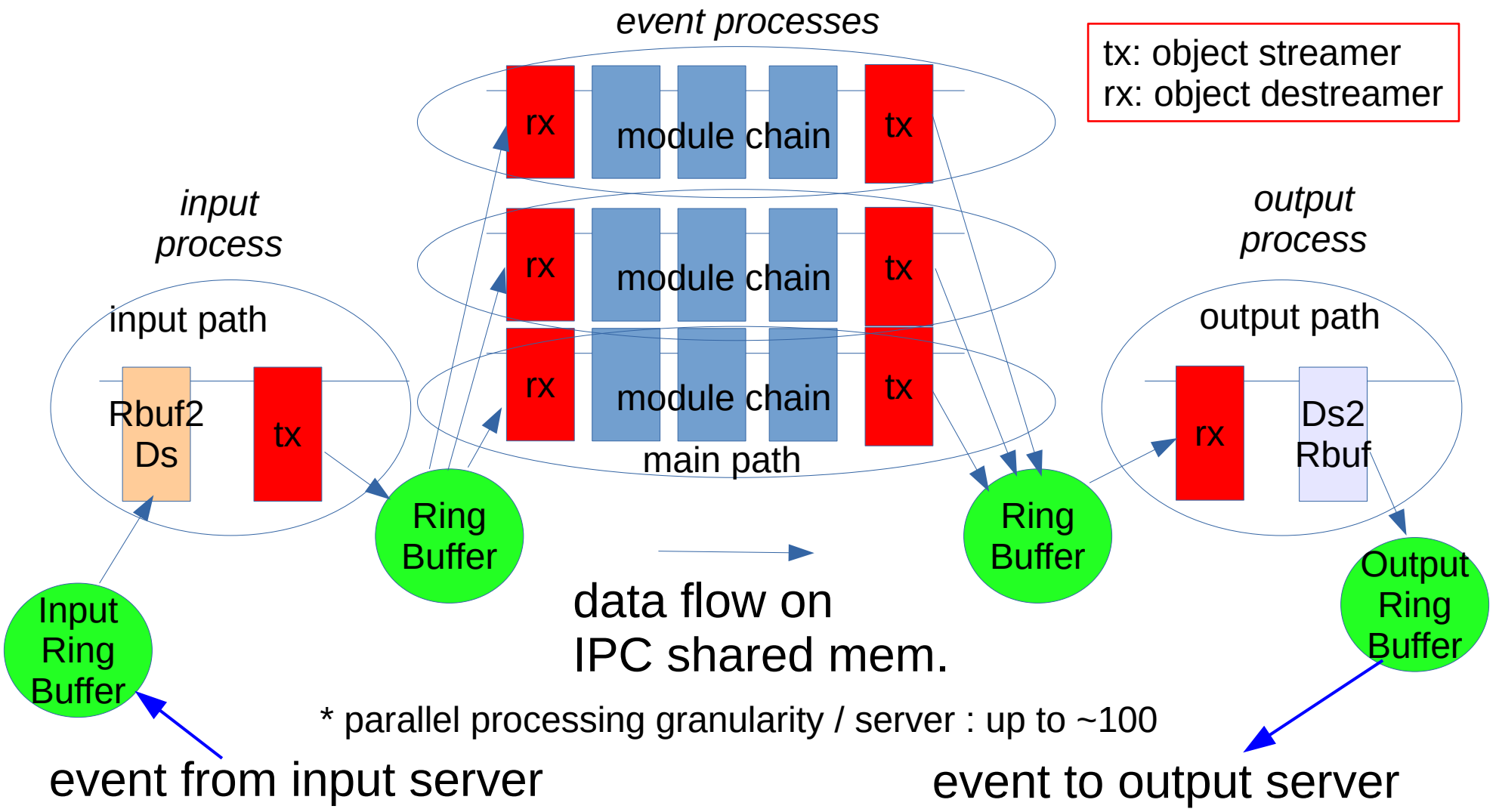
- Replace the framework inside a worker (hbasf2) with the improved original basf2 parallel processing framework.

- Replace IPC RingBuffer in basf2 with ZeroMQ transport.

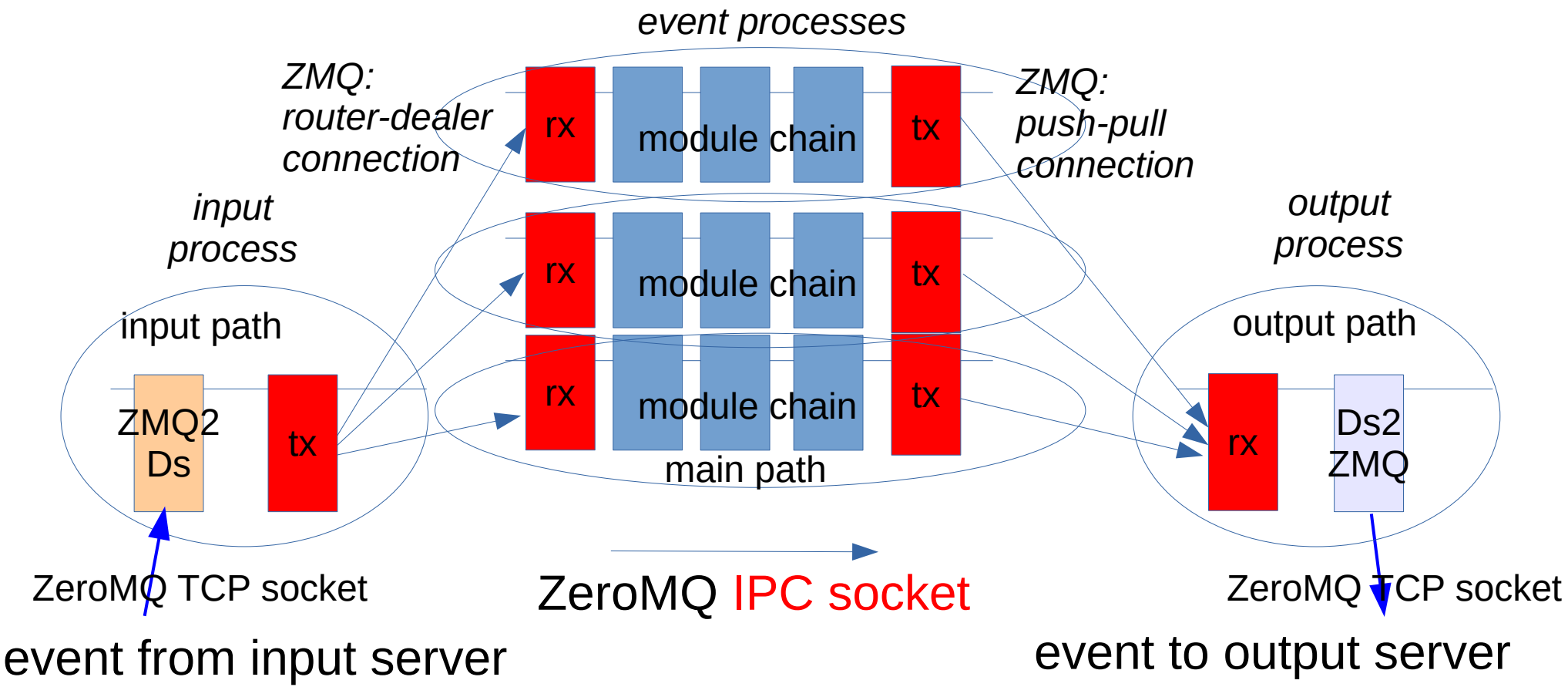
- At the beginning of each event process, the event data is copied to a buffer, and it is removed when processed successfully.

- If the process dies, basf2 mother process moves the faulty event to the output process (with a bad-event tag) and restart a new event process.

Parallel processing in original basf2 (on a worker)



Parallel processing in improved basf2 = **zmq-basf2**



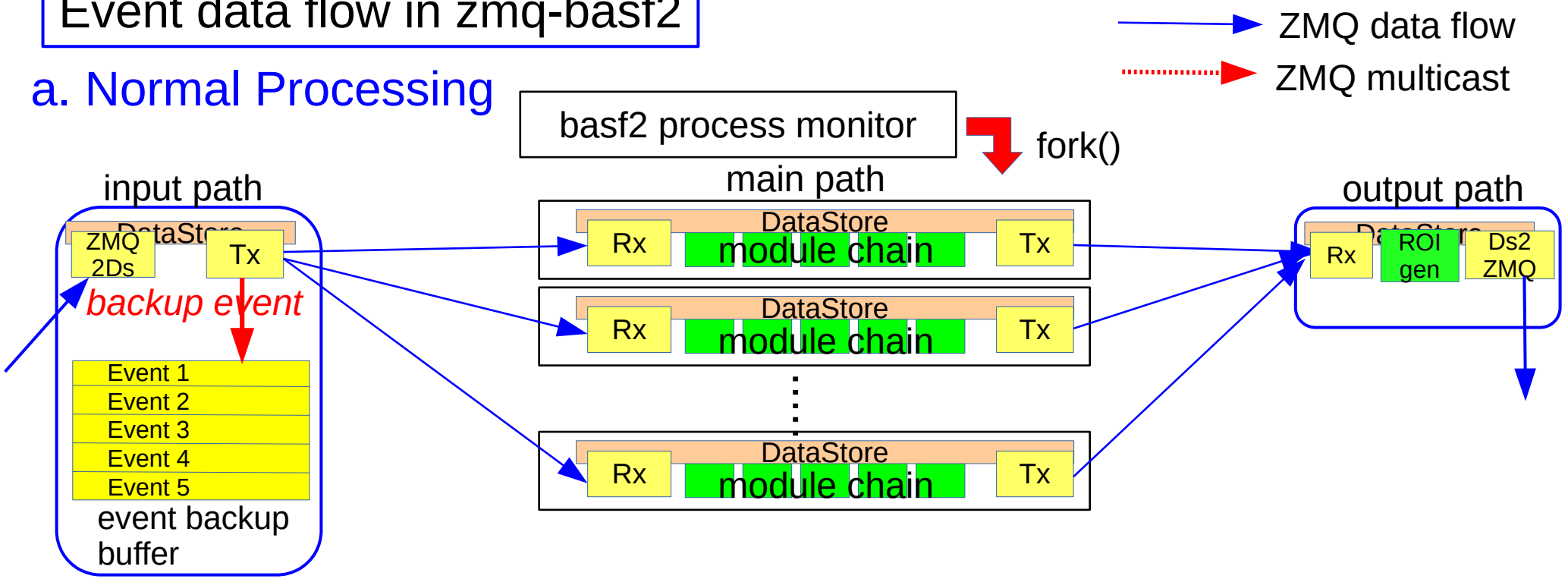
- Can be executed offline utilizing parallel processing on multi-core CPU just by replacing input/output modules with (ROOT) file I/O.

ZMQ connection in zmq-basf2

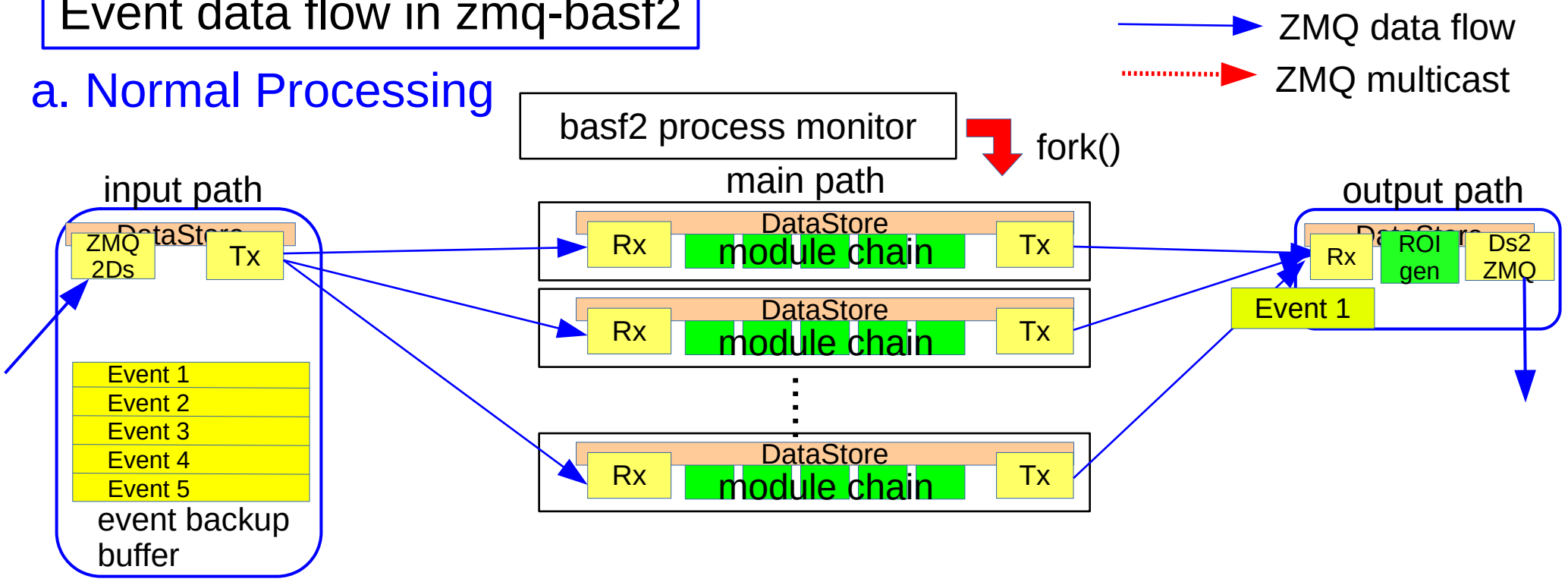
- All connections are based on the IPC socket of ZeroMQ.

- Each ZMQ client is implemented with two connections.
  1. Data socket (unidirectional). *(to replace previous IPC ring buffer)*
     - Event distribution (load balanced connection):
       1-to-N connection with ZMQ ROUTER-DEALER connection
     - Event collection (fair queued connection):
       N-to-1 connection with ZMQ PUSH-PULL connection
  2. Monitor socket (multicast)
     - Two sockets with ZMQ PUB and SUB properties
     - A set of call-backs are hooked and they are called when corresponding message is received.

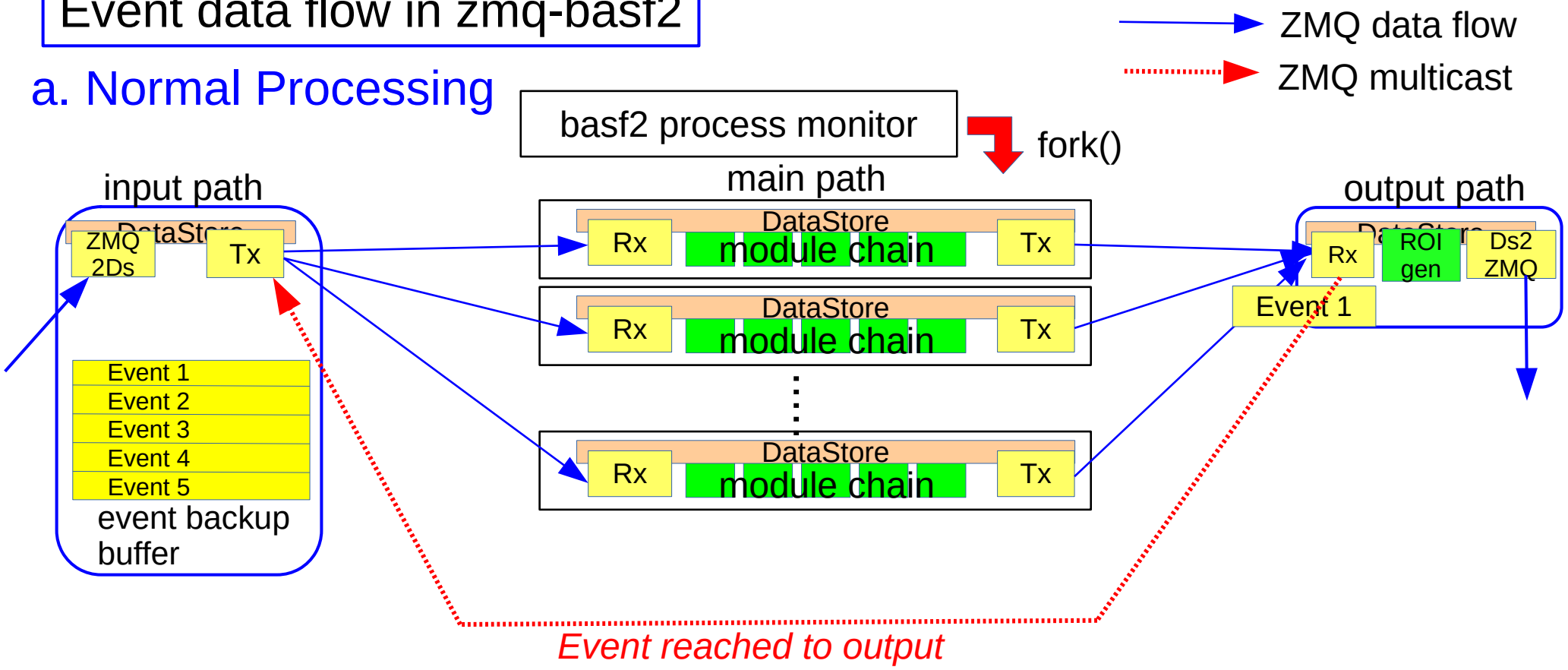
Event data flow in zmq-basf2



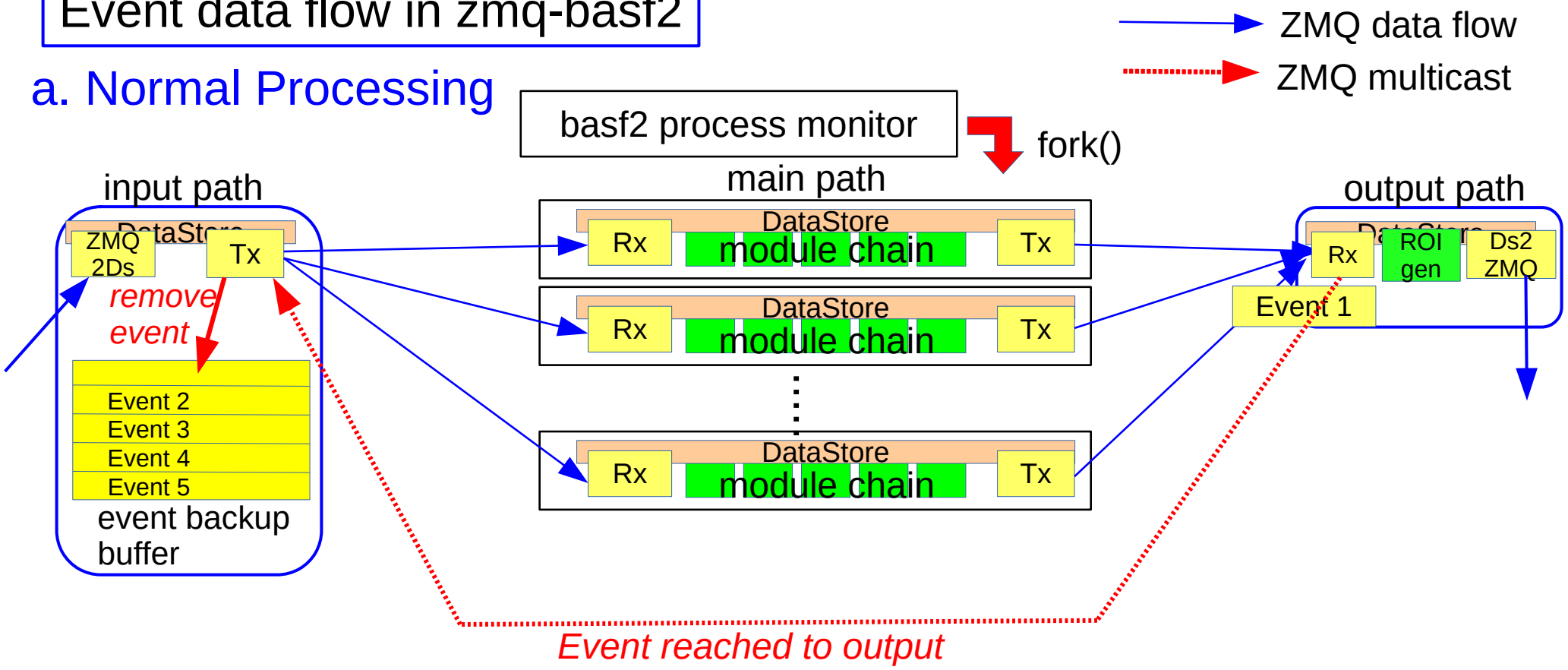
Event data flow in zmq-basf2



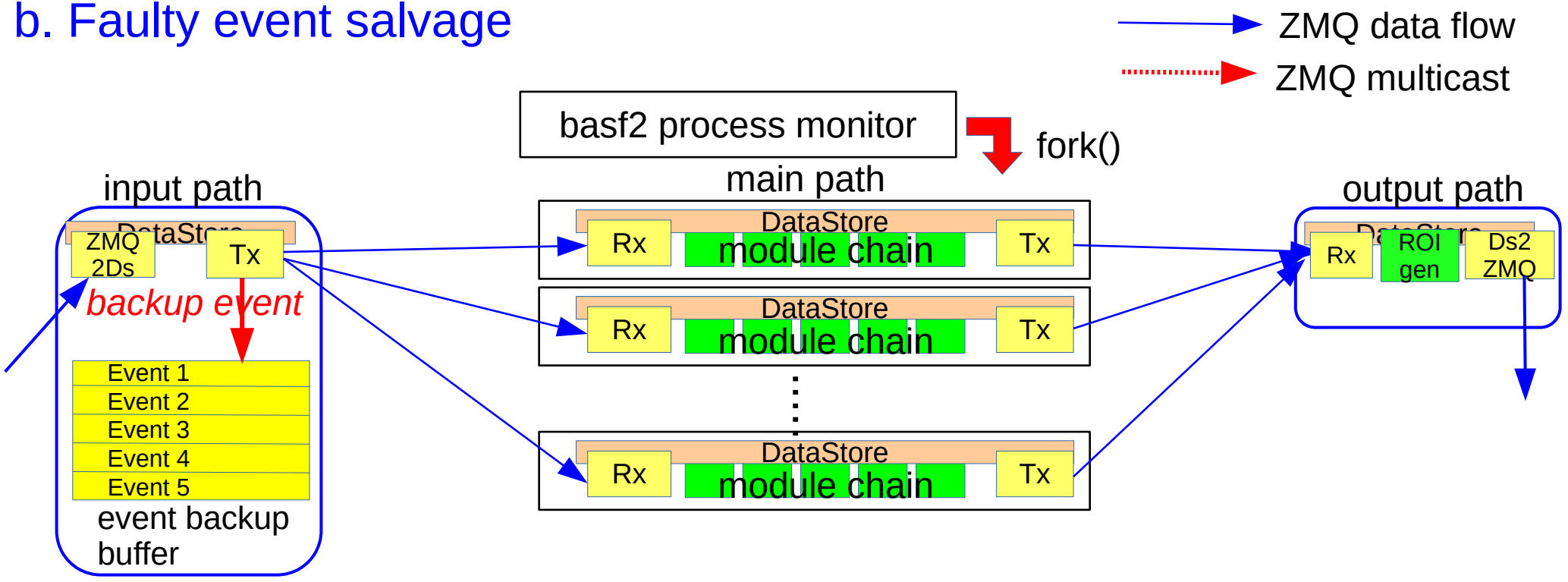
Event data flow in zmq-basf2



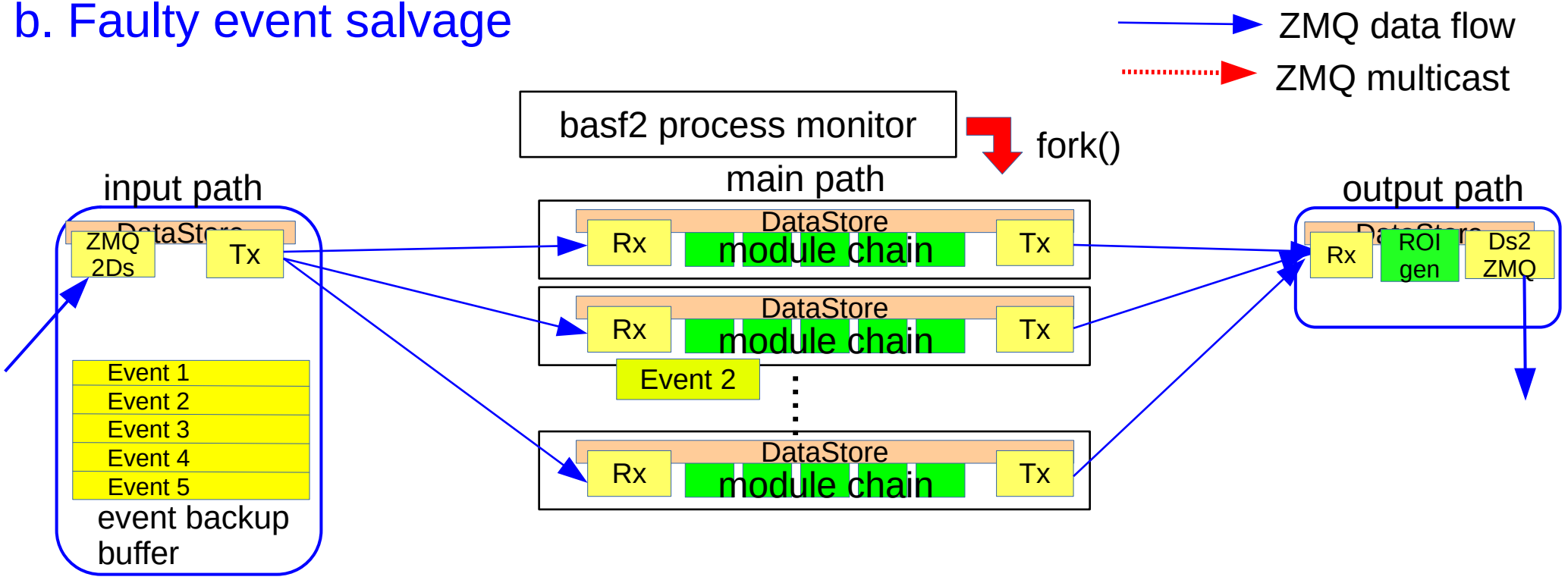
Event data flow in zmq-basf2



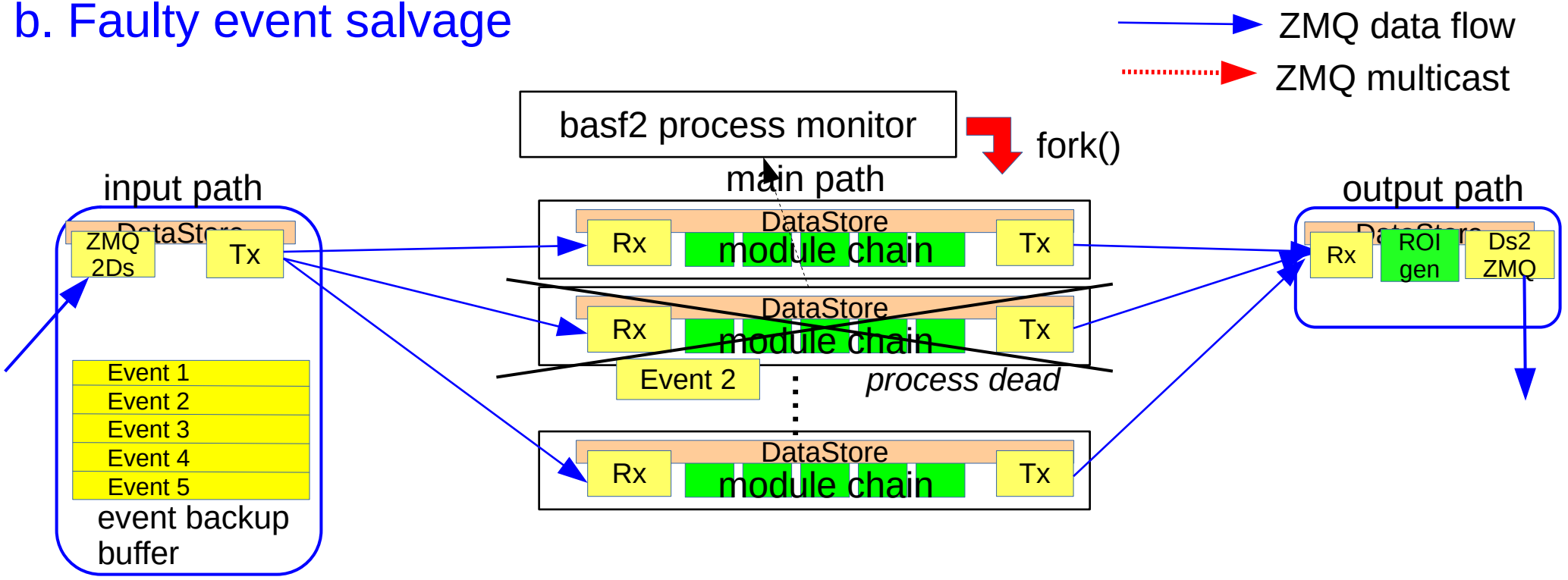
b. Faulty event salvage



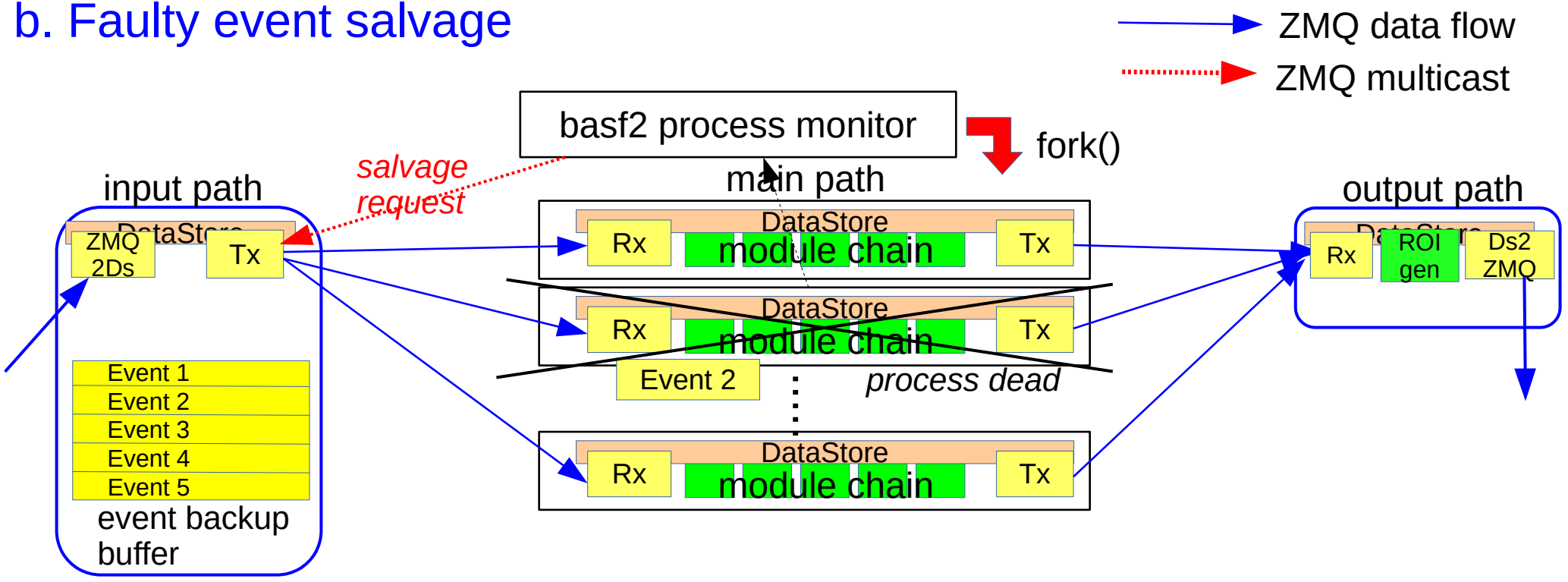
b. Faulty event salvage



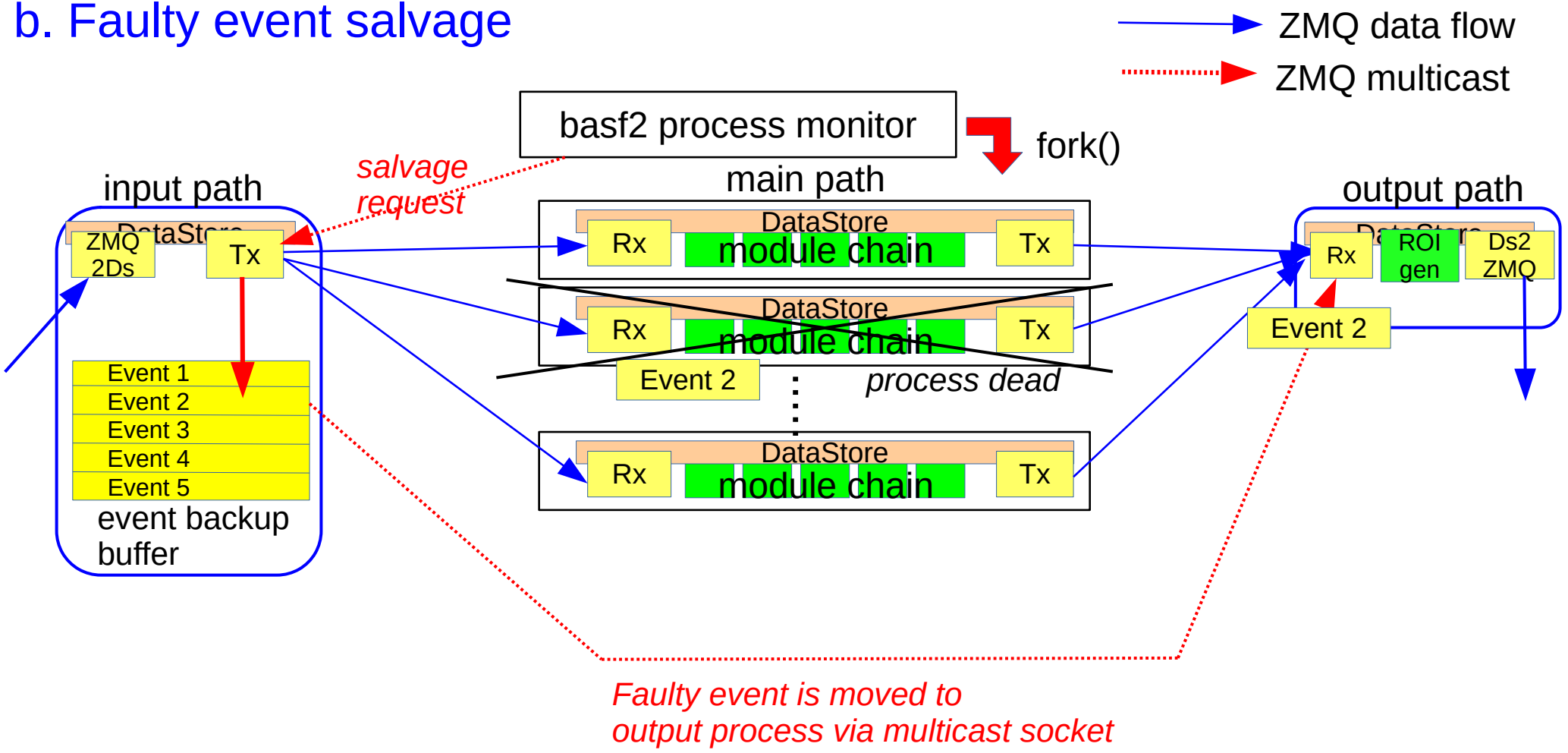
b. Faulty event salvage



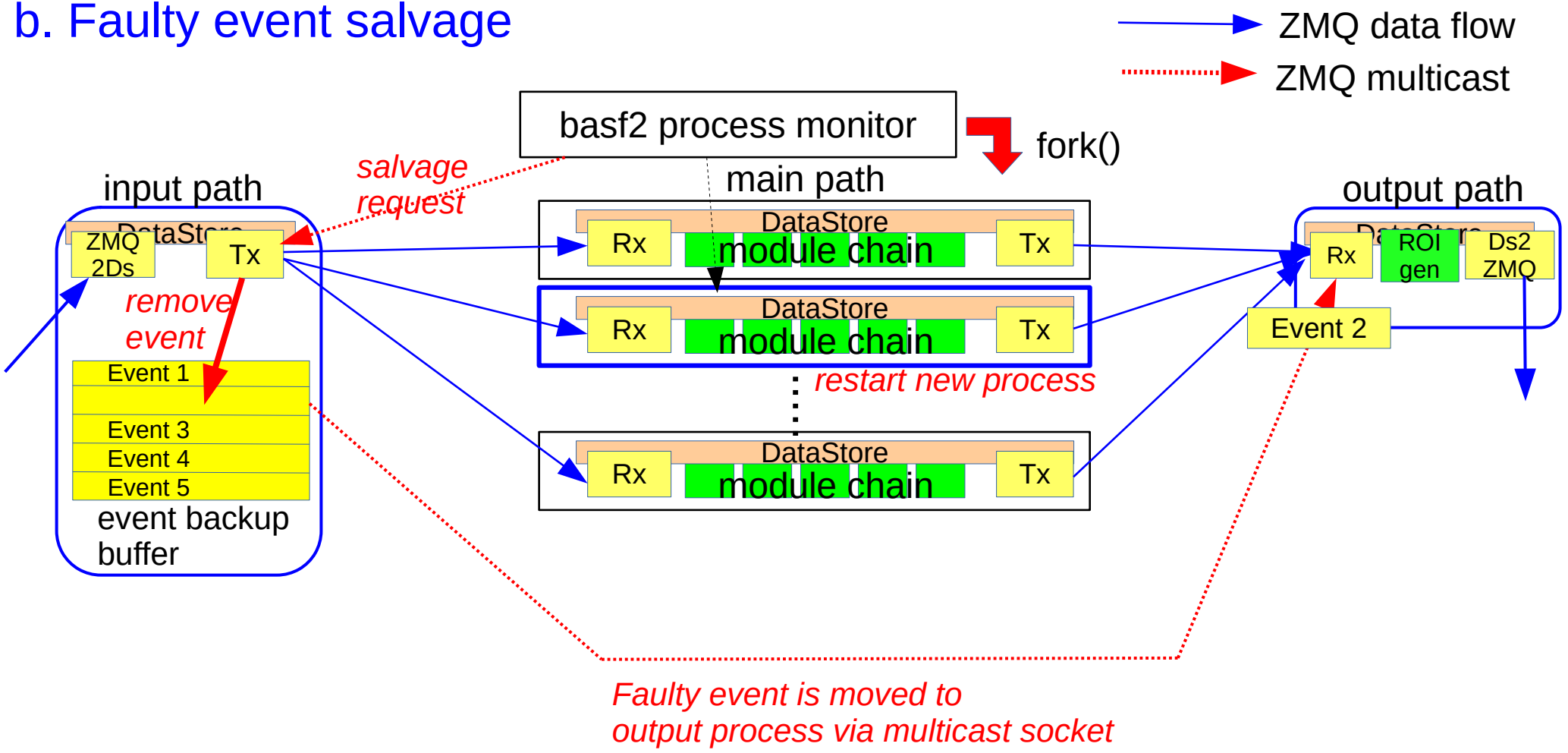
b. Faulty event salvage



b. Faulty event salvage



b. Faulty event salvage

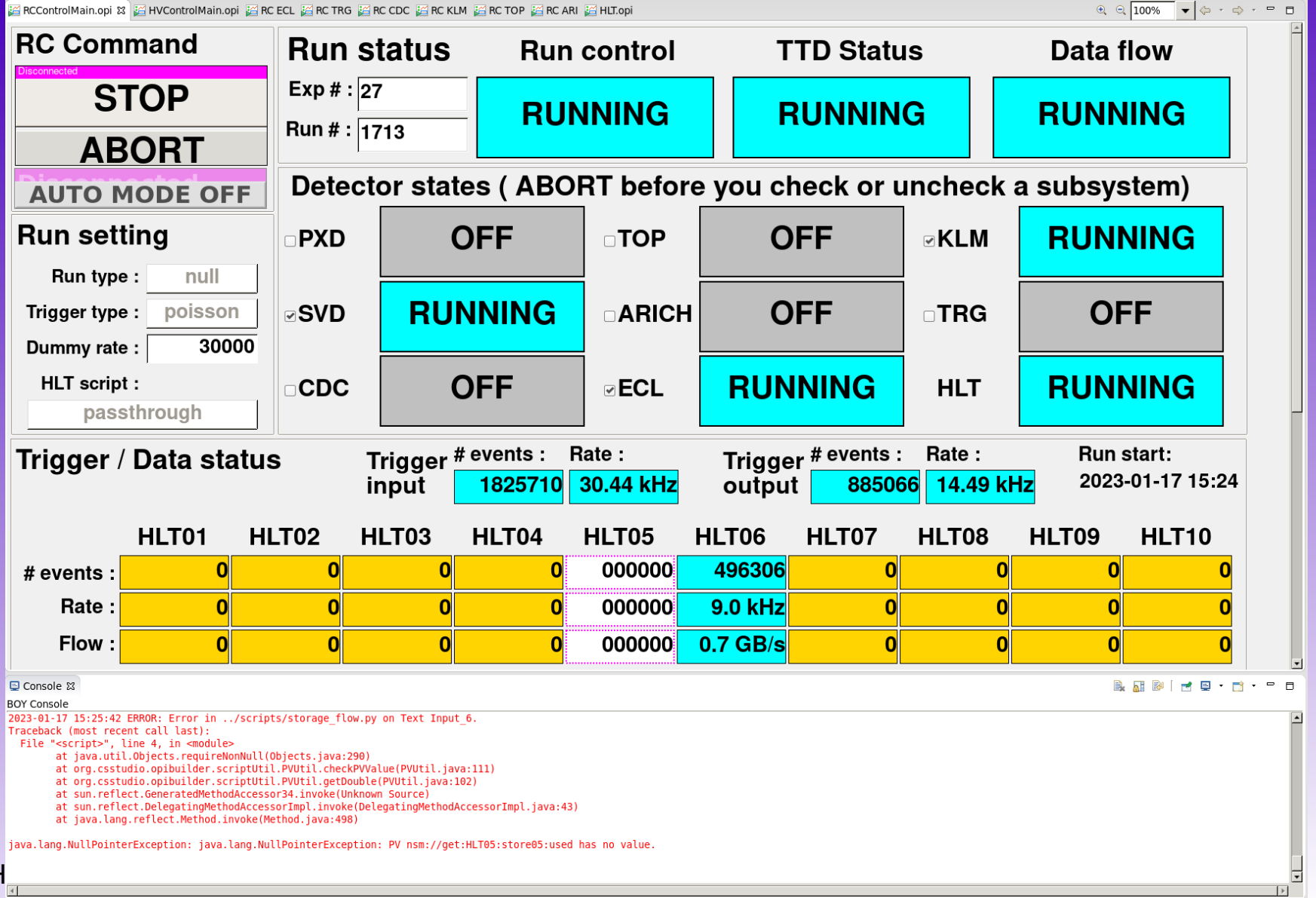


Test of new framework

- zmq-basf2 has been implemented in one of real HLT units replacing previous hbasf2 and is being tested.

- The data flow is confirmed to work stably in the real DAQ operation (at a high rate up to 10kHz/unit).

- The event recovery was tested by inserting segmentation fault in the event processing and monitoring the output record.
  * Restart of new worker process : confirmed.
  * The output of faulty event is checked by looking at the existence of HLT processing object in the recorded file.


RCControlMain.opi
HVControlMain.opi
RC ECL
RC TRG
RC CDC
RC KLM
RC TOP
RC ARI
HLT.opi
100%
RC Command
Disconnected
STOP
ABORT
AUTO MODE OFF
Run status
Run control
TTD Status
Data flow
Exp # : 27
Run # : 1713
RUNNING
RUNNING
RUNNING
Detector states (ABORT before you check or uncheck a subsystem)
PXD
OFF
TOP
OFF
KLM
RUNNING
SVD
RUNNING
ARICH
OFF
TRG
OFF
CDC
OFF
ECL
RUNNING
HLT
RUNNING
Run setting
Run type : null
Trigger type : poisson
Dummy rate : 30000
HLT script :
passthrough
Trigger / Data status
Trigger input
events : 1825710
Rate : 30.44 kHz
Trigger output
events : 885066
Rate : 14.49 kHz
Run start:
2023-01-17 15:24
HLT01 HLT02 HLT03 HLT04 HLT05 HLT06 HLT07 HLT08 HLT09 HLT10
events : 0 0 0 0 000000 496306 0 0 0 0
Rate : 0 0 0 0 000000 9.0 kHz 0 0 0 0
Flow : 0 0 0 0 000000 0.7 GB/s 0 0 0 0
Console
BOY Console
2023-01-17 15:25:42 ERROR: Error in ../scripts/storage_flow.py on Text Input_6.
Traceback (most recent call last):
File "<script>", line 4, in <module>
at java.util.Objects.requireNonNull(Objects.java:290)
at org.csstudio.opibuilder.scriptUtil.PVUtil.checkPVValue(PVUtil.java:111)
at org.csstudio.opibuilder.scriptUtil.PVUtil.getDouble(PVUtil.java:102)
at sun.reflect.GeneratedMethodAccessor34.invoke(Unknown Source)
at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
at java.lang.reflect.Method.invoke(Method.java:498)
java.lang.NullPointerException: java.lang.NullPointerException: PV nsm://get:HLT05:store05:used has no value.

```
[INFO] ================================================================================
[INFO] Processed:   1 runs,      8/     0 events.
[INFO] ================================================================================
[INFO] DataStore collections in event 20997130
[INFO] ================================================================================
[INFO] Type                 Name                        #Entries          <Event>
[INFO] EventMetaData        EventMetaData
[INFO] ROIpayload           ROIpayload
[INFO] SoftwareTriggerResult  SoftwareTriggerResult
[INFO] SoftwareTrigger::SoftwareTriggerVariables  SoftwareTriggerVariables
[INFO] OnlineEventT0[]      OnlineEventT0s              0
[INFO] ROIid[]              ROIs                        0
[INFO] RawARICH[]           RawARICHs                   0
[INFO] RawCDC[]             RawCDCs                     0
[INFO] RawECL[]             RawECLs                     0
[INFO] RawFTSW[]            RawFTSWs                    1
[INFO] RawKLM[]             RawKLMs                     0
[INFO] RawPXD[]             RawPXDs                     4
[INFO] RawSVD[]             RawSVDs                     0
[INFO] RawTOP[]             RawTOPs                     0
[INFO] RawTRG[]             RawTRGs                     0
[INFO]
[INFO] --------------------------------------------------------------------------------
[INFO] Type                 Name                        #Entries      <Persistent>
[INFO] ProcessStatistics    ProcessStatistics
[INFO]
[INFO] ================================================================================
[INFO] Processed:   1 runs,      9/     0 events.
[INFO] ================================================================================
[INFO] DataStore collections in event 20997830
[INFO] ================================================================================
[INFO] Type                 Name                        #Entries          <Event>
[INFO] EventMetaData        EventMetaData
[INFO] OnlineEventT0[]      OnlineEventT0s              0
[INFO] ROIid[]              ROIs                        0
[INFO] RawARICH[]           RawARICHs                   0
[INFO] RawCDC[]             RawCDCs                     0
[INFO] RawECL[]             RawECLs                     0
[INFO] RawFTSW[]            RawFTSWs                    1
[INFO] RawKLM[]             RawKLMs                     0
[INFO] RawPXD[]             RawPXDs                     4
[INFO] RawSVD[]             RawSVDs                     0
[INFO] RawTOP[]             RawTOPs                     0
[INFO] RawTRG[]             RawTRGs                     0
[INFO]
[INFO] --------------------------------------------------------------------------------
```

Event processed normally.
SoftwareTriggerResults
is there.

Seg-faulted event.
Only RawData are there.

Summary and Plan

- A new core framework for the Belle II HLT is developed for the recovery from the unexpected processing faults.

- It is implemented by replacing the old IPC based ring buffer with the ZeroMQ IPC socket in the original basf2 framework.

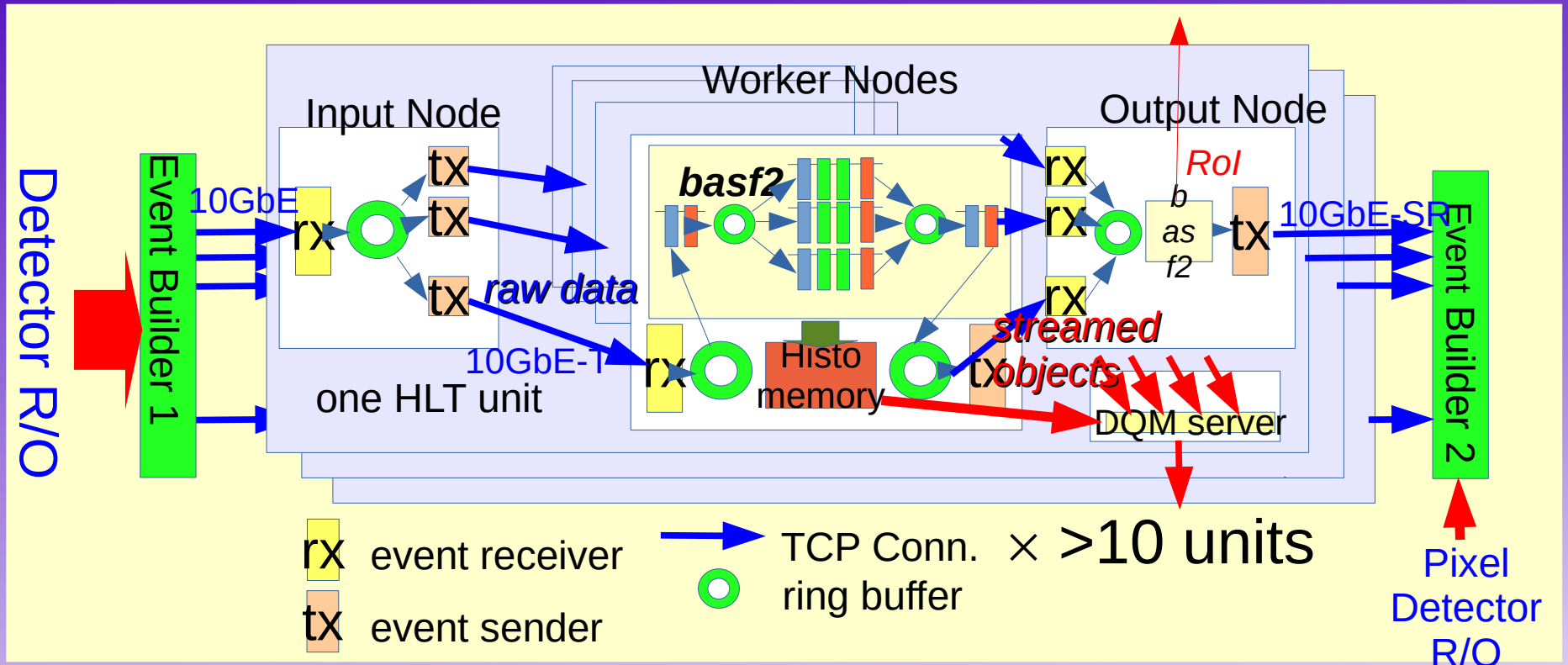
- The event recovery is implemented using the ZeroMQ multicast.

- It is integrated with the external ZeroMQ-based HLT data transport and being tested in the actual DAQ system.
  -> Basically working stably in a single HLT unit.

- Test with multiple HLT units is scheduled. When succeeded, the framework will be used in coming beam run from Dec.2023.

Backup Slides

Belle II High Level Trigger (HLT) : Original Design



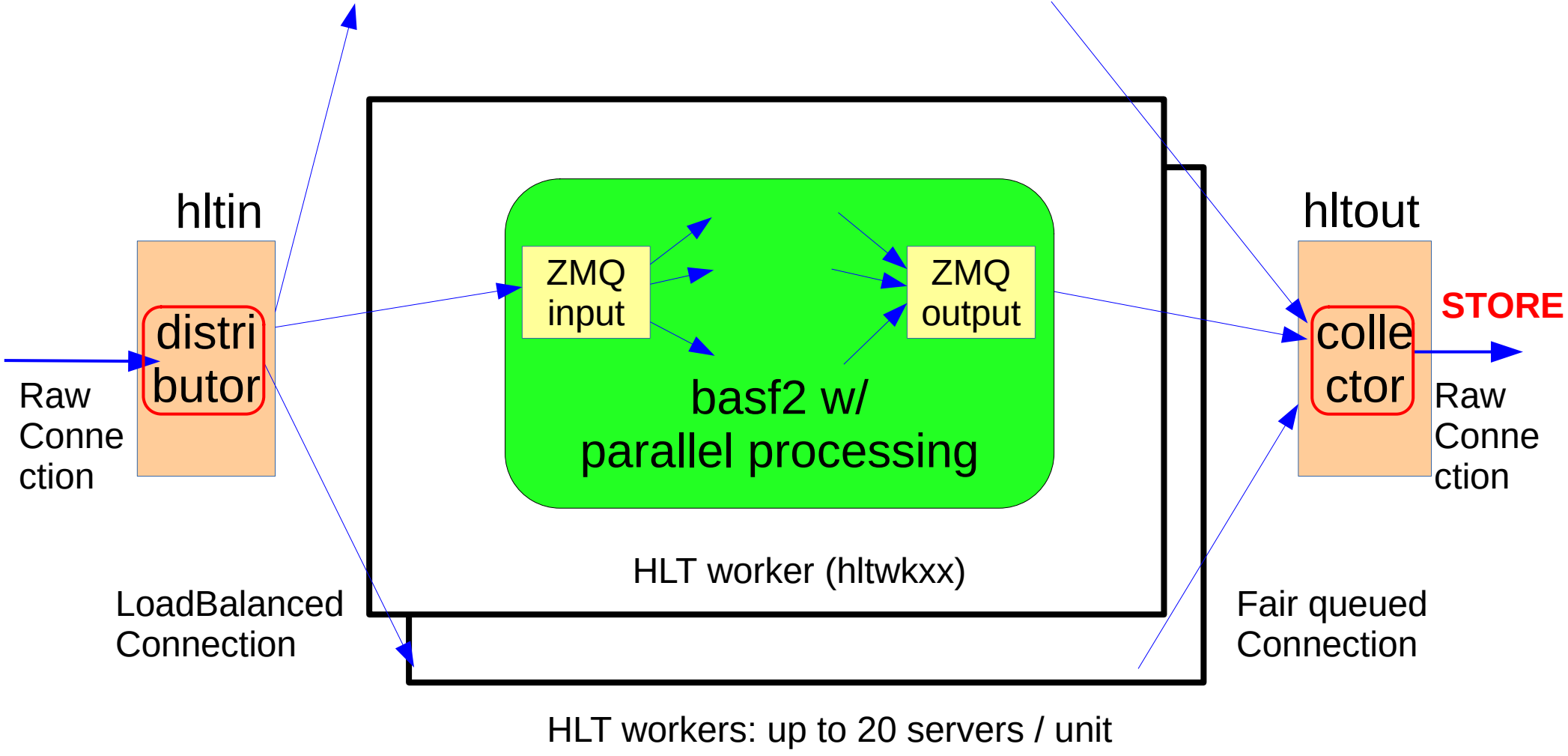
ZeroMQ

- An open-source package for the general message passing.
  -> Strong community support. Standard in HEP community.

- The usage resembles to that of the standard UNIX socket, but it has various functions.

- It supports N-to-1 and 1-to-N connection with a variety of connection style including load-balanced pipeline.

https://zeromq.org/


hltin
distri
butor
Raw
Conne
ction
ZMQ
input
ZMQ
output
basf2 w/
parallel processing
HLT worker (hltwkxx)
LoadBalanced
Connection
hltout
colle
ctor
STORE
Raw
Conne
ction
Fair queued
Connection
HLT workers: up to 20 servers / unit