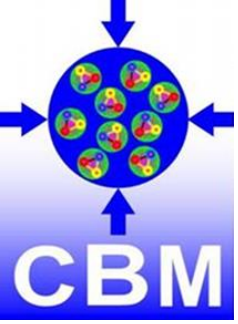


STAR-CBM Joint Workshop
PI, Heidelberg

eTOF online software status and plan: configuration and event detection

What needs to be done “online side”?

Configuration and startup

- Do only necessary configuration depending on situation
- Simplify startup procedure
- Minimize delays to fit the STAR requirements

Event building

- Build eTOF events with x12 data
- No event (STAR trigger) loss => each trigger has to generate an event object
- Long term stability to minimize reconfiguration needs

Monitoring

- Plots monitoring event builder to detect problem in data or event detection
- Should not slow down the event builder!

Unpacking (STAR side)

- New data format

Configuration and startup

2018 run feedback

1) Full configuration squeezed to less than 2 minutes
2) Full configuration applied on each “restart”
3) Access to trigger reconfiguration from the STAR side prepared for the shift crew in second half of the 2018 run

2019 status and plans

1) Full configuration slower due to new elements in the readout chain:
   a. Jitter cleaner which needs to be locked
   b. GBTx between the FEE and the FPGA need its own configuration
2) Priority is stability & reliability over speed for now
3) Startup scripts will be prepared to
   - Run the configuration of the sectors in parallel (x12 speedup)
   - Allow three levels of re-configuration:
     a. Full => E.g if full power loss or eTOF computer stuck, up to a few minutes
     b. GBTx + FEE + Time synchronization => E.g. if FEE - LV loss, < 2 min
     c. FEE + Time synchronization only => standard for e.g. data run change, < 30-40 s

Event Building: feed back and options

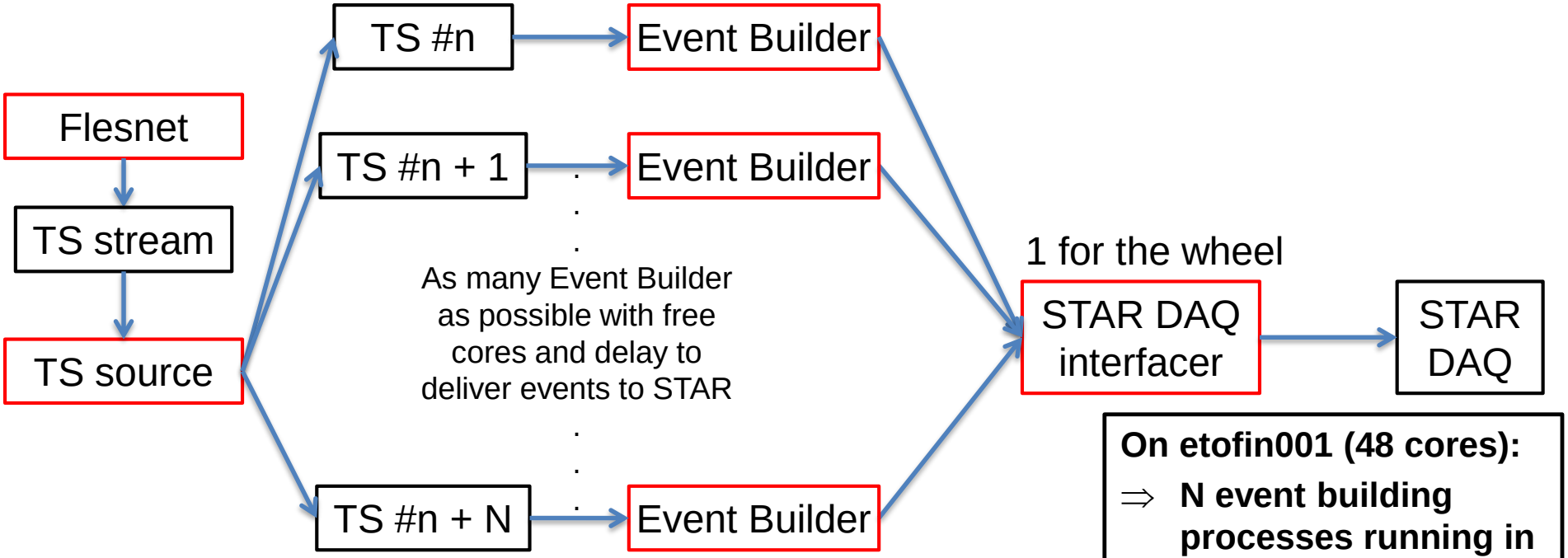
2018 run feedback

1) Event building method used worked quite well
   a. Unpack data and separate eTOF hits from STAR triggers in separate buffers
   b. Scan trigger buffer and check all readout board saw a trigger => good event
   c. Scan data buffer and extract data of each readout board within coincidence window
2) Coincidence window was large (2 us) while coincidence peak few 100’ns wide
3) Some events were lost due to processing speed at 2 levels
   a. CBM DAQ side: data drop when event builder too much behind source
   b. CBM to STAR: event lost when eTOF data sent to STAR too late
   c. Both linked to same origins: slowdown due to monitor plots saving to disk (+++), long time sorting of data when data spike or increased noise
4) Maximal run duration ~12h due to unsupported counter cycle

2019 run options

1) Need parallel processing of data: x12 data source, 48 cores on etofin001
2) Parallelization matching the HW segmentation: per sector => less time sorting, more complicated to implement
3) Parallelization matching data segmentation: building per TS => simpler

Event Building: chosen solution



On etofin001 (48 cores):

- ⇒ **N event building processes running in parallel, 1TS for each**
- ⇒ **1 interface to STAR**
- ⇒ **~6 Flesnet processes**

Event builder with parallel data blocks processing

1) Event building with full wheel and 1 TS
2) Inside process, event detection and building MS per MS
3) Tuning param.: N processes O(32-40), MS size O(1-10 ms), MS/TS O(10-100)
4) Tuning balance: avoid TS loss when data spike and deliver events in time to STAR

Principle tested by Norbert with cosmics unpacker => ~15x speedup for 32 cores

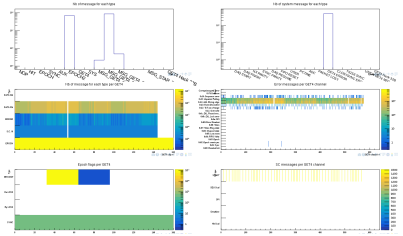
Monitoring

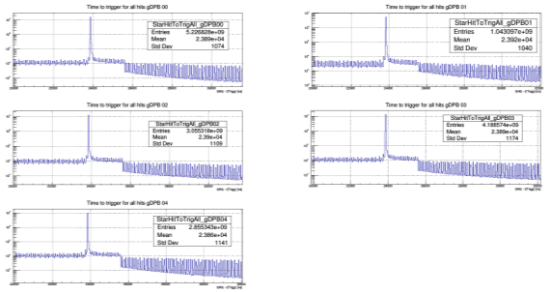
2018 run feedback

1) Histograms saving in event builder lead to data loss
2) The “shift monitor” was not easily accessible, had too many plots and was therefore never really used outside of experts
3) Still histograms helped a lot in forensics when the eTOF readout failed

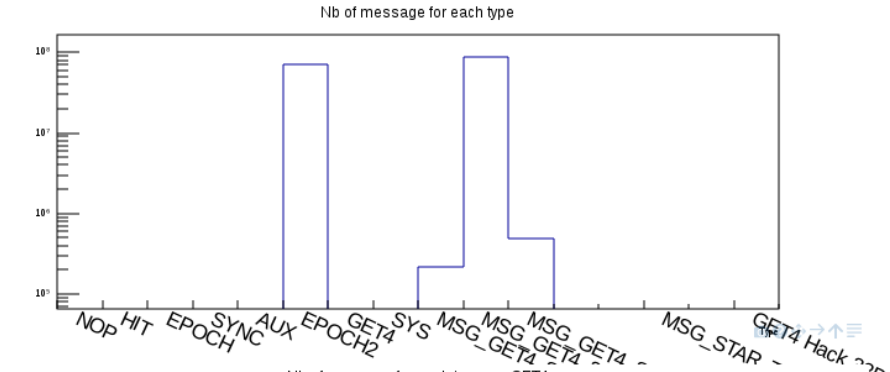
2019 plans

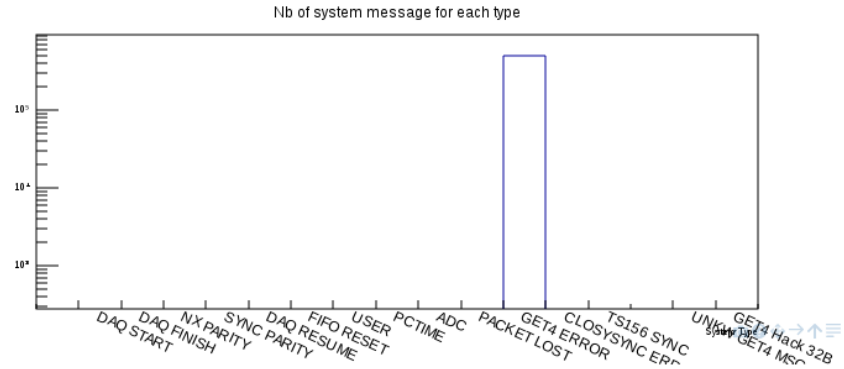
1) Minimize the amount of plots using the HD setup
   a. Should focus on information lost after event building or related to the event building itself
   b. As shown by eTOF software team in STAR, other plots can be done with the data reaching the STAR DAQ
2) Find a way to decouple the histogram saving to disk from the event building tasks (histogram manager task running in parallel or part of the TS processed in two tasks, event building and monitor)

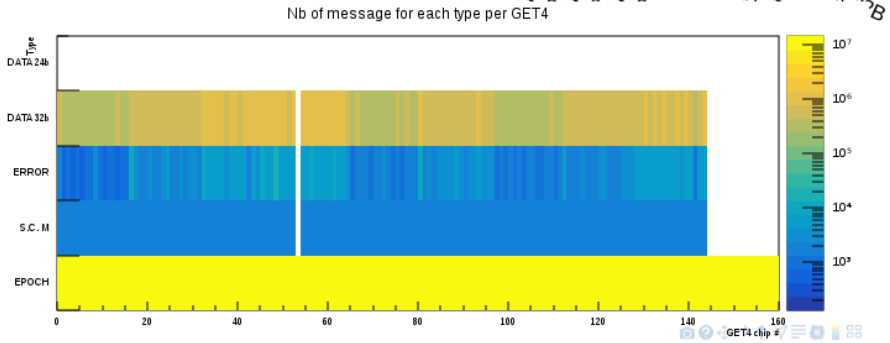


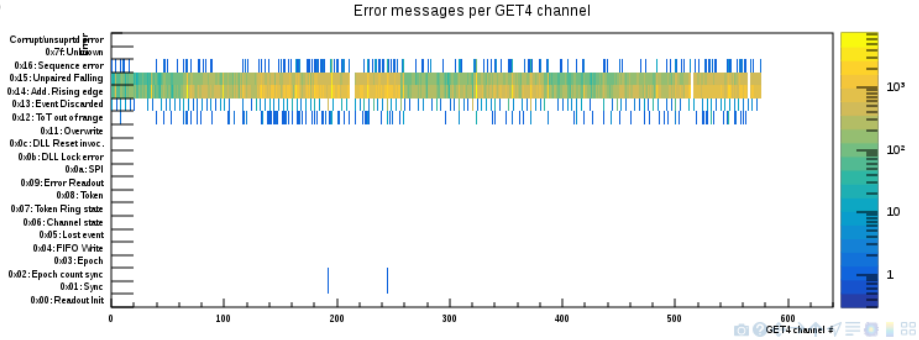


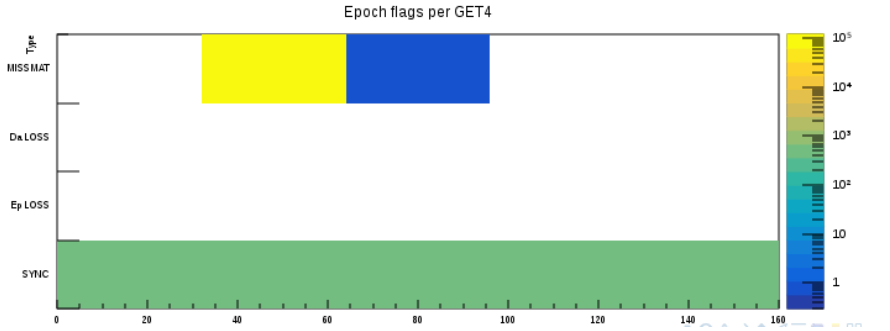
Monitoring example: Readout status

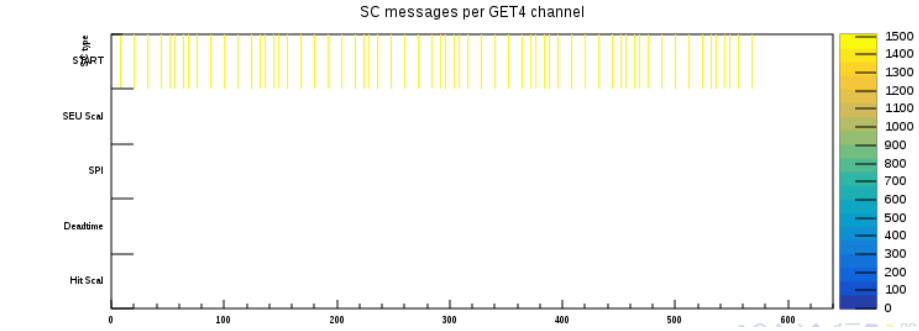




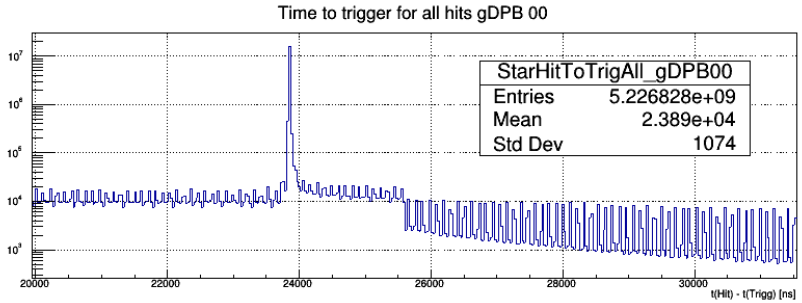


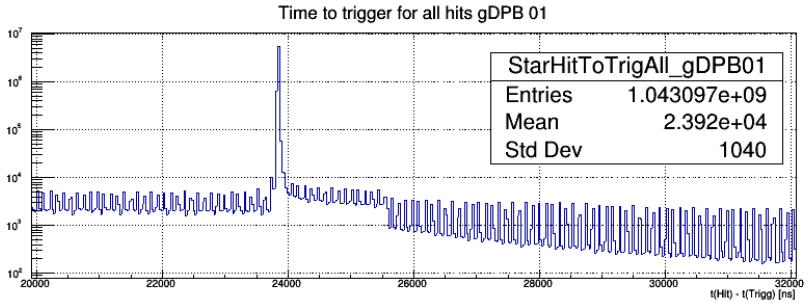


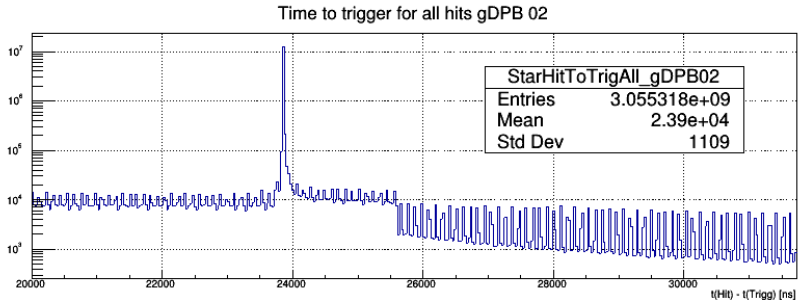


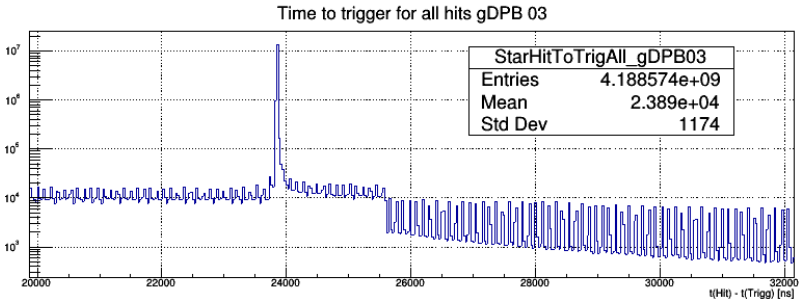


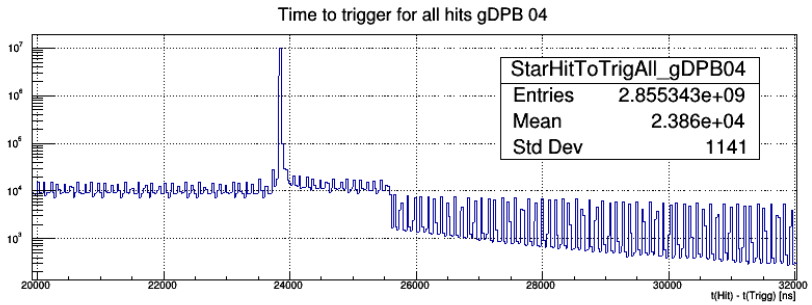
Monitoring example: Time correlation











Unpacking on STAR side

New message data format

1) Linked to increased FEE count per FPGA
2) Prepares an eventual decrease of data size by 30% by removing obsolete data fields from early CBM developments
3) Introduce a few new Error message types to help diagnostic/recovery

Description file

1) Work in progress available to all CBM members at
   a. https://redmine.cbm.gsi.de/projects/mcbm-proposal/wiki/Messages_data_format_and_microslices_content_data_format
   b. Still under modification as we test the new FW in Heidelberg and mCBM
   c. Will be added to eTOF page in STAR website as soon as stable, together with a description of the fields
2) Unpacking class available at
   a. https://redmine.cbm.gsi.de/projects/cbmroot/repository/show/trunk/fles/mcbm2018/dataformat

Other formats: event class, event header, trigger information

⇒ No changes expected, pending event builder testing in Heidelberg

STAR Requirements

What are the STAR DAQ requirements and "best practice" for the following points for a "detector not anymore under test"?

- Maximal start time for
  - Cold boot?
  - Full restart?
  - Quick restart? (Should be the 2 minutes limits we tried to obey in 2018)
- Maximal time delay for event delivery to the STAR DAQ?
  - I remember something like 1-2 minutes, linked to the trigger rate and how fast STAR runs out of trigger indices
- Event disorder? (delivering events to the STAR DAQ in a different order from the triggers)

$\Rightarrow$ Will be needed to know to which degree we have to tune the startup procedure and event building system

Difficulties and priorities

Configuration and startup

- No difficulties expected, natural derivation of mCBM and Heidelberg test-stands
- Priority will be reliability to maximize automatization
- Optimization after this

Event building

- No difficulties expected, amount of work depend on how far from STAR needs we are after the first implementation
- Priority is to extract the algorithm of the 2018 event builder to run it in parallel tasks
- Testing to be done in Heidelberg with extrapolation for more sectors

Monitoring

- Main difficulty will be finding the time to prepare it as typically pushed for later
- Lower priority

Unpacking (STAR side)

- No difficulties expected, most interfaces stay the same (hidden implementation)

Technical and administrative needs

eTOF DAQ computer status: etofin001

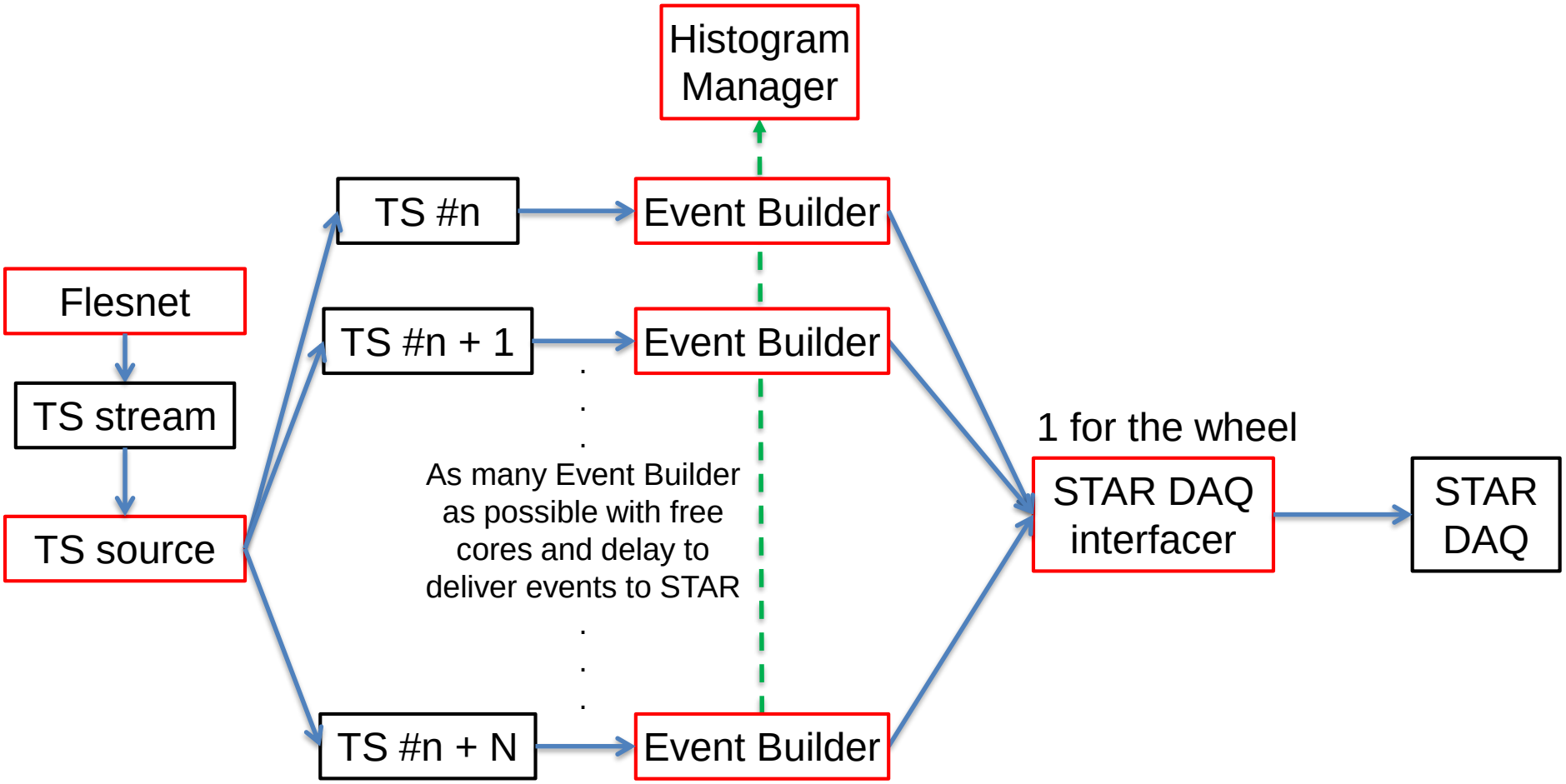
- In 2018 run, hidden behind STAR DAQ network and (to my knowledge) never declared to BNL IT
- Made the update of the eTOF software complicated after the German team left BNL
  - ⇒ No internet connection
  - ⇒ No access to our versioning systems
  - ⇒ Software fixes applied by copies through the DAQ network and text file as “log”, with a few near misses of over-written eTOF specific modifications
  - ⇒ No OS updates or new packages installation
- Mixed mission: this node is used for configuration, DAQ and some slow control
- <u>Questions for the BNL experts</u>
  - Should/must we make the etofin001 node official at BNL?
  - If yes, in which network should it live? (my best guess is the STAR DAQ network)
  - Could it get some (eventually restricted) connection to the outside world? An svn and/or git access, both way, would already help a lot
  - If not, is there a versioning system that could be access from it and from Germany?

Summary

- First version of the configuration and startup tools in test in Heidelberg and mCBM setups
- Example of the CBMROOT tasks architecture for the event builder tested with unpacker of Heidelberg cosmic setup, now need to be applied to the event builder with STAR triggers
- Monitoring tasks update on hold until most helpful plots identified with HD/mCBM setups
- New data format for STAR side software will be made available soon, as the changes needed by the FW development are decreasing

- No big difficulties expected, but some testing/development still pending
- Priorities are first stability, then matching STAR DAQ minimal requirement, then performance optimization if possible

Thank you for your attention!

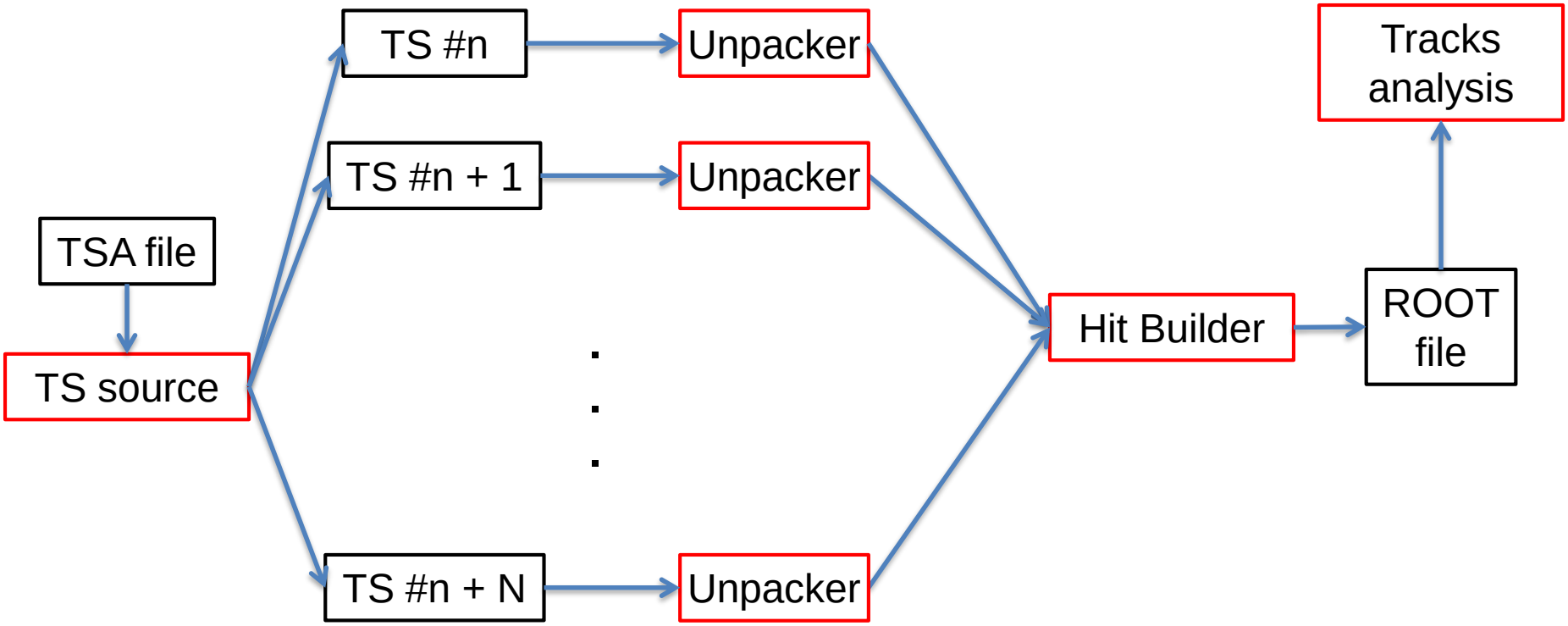


Event Builder = unpack, find trigger and select data

STAR DAQ interface = gather events from the event builders and send them to the STAR DAQ (in order if necessary)

Histogram manager = takes care of monitor plots creation, sharing and saving to disk

Event Building example: HD cosmics



Unpacker = unpack to “Digi”, time sort them and build events by selection time intervals with multiplicity above threshold
Hit Builder = Build Hits from L/R matches and clusters, apply calibration
Cosmics analysis = Use geometry, build tracks, … => See plots shown by Ingo!