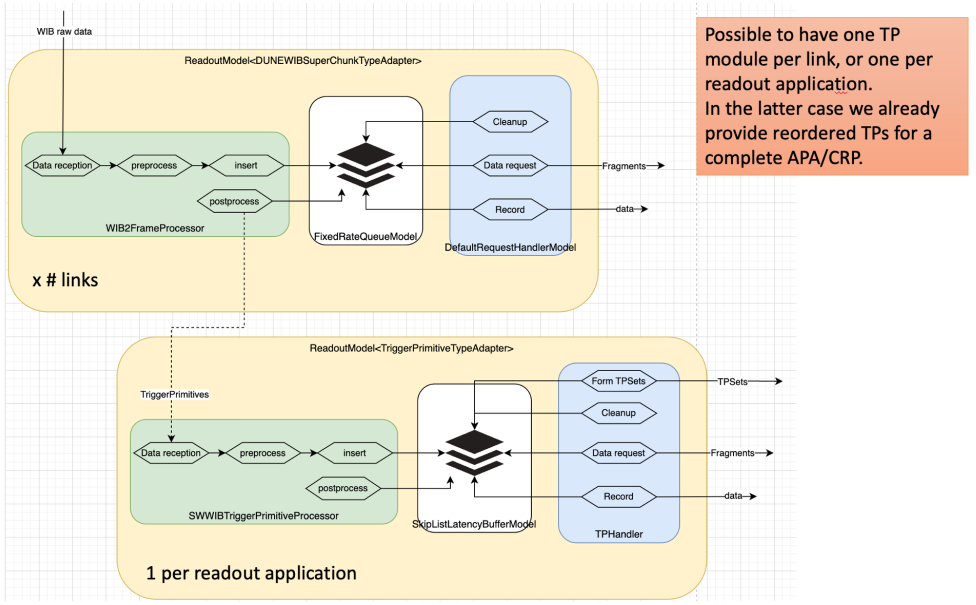
Trigger infrastructure restructuring – episode 3

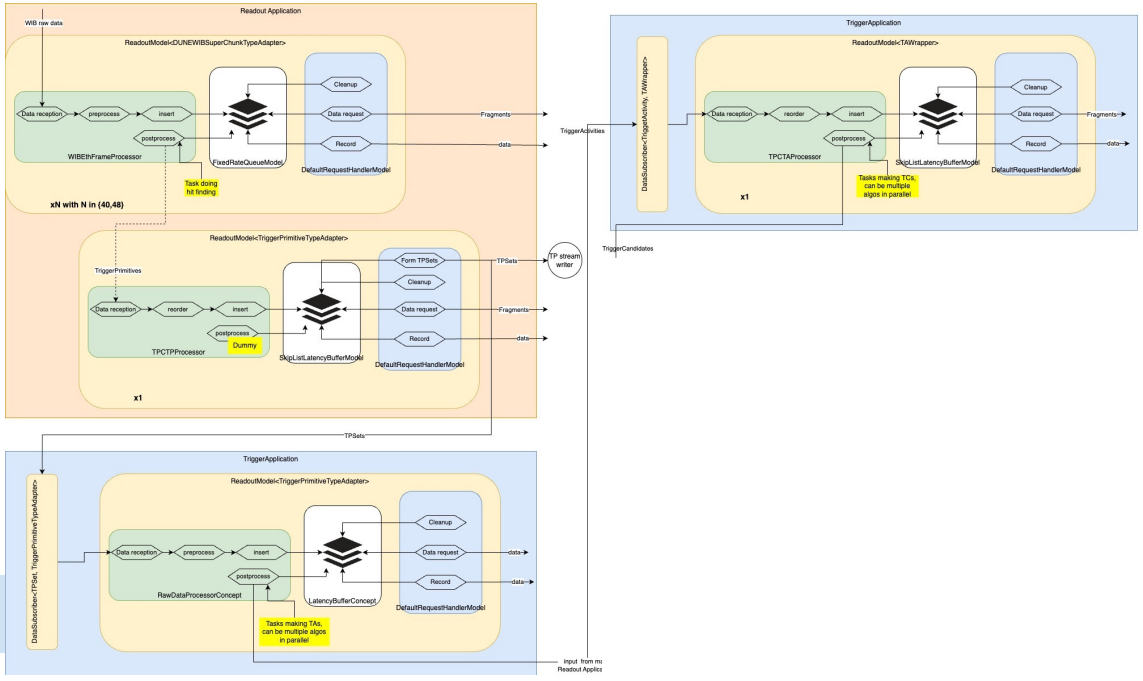
G. Lehmann Miotto, A. Sztuc

Recap from episode 1 & 2

- Last year we carried out a first restructuring of the trigger chain by grouping an reordering all the TPs from one detector unit (e.g. 1 APA), and to buffer TPs systematically in the readout
  - Episode 1: only 1 stream per TA maker



- We realised that the trigger configuration was very monolithic and several code blocks were sub-optimal.
- Alejandro improved the loading of algorithms, which allowed to eliminate a lot of code duplication
- Episode 2: we transformed the trigger modules to use the same design pattern as the readout modules, with shared code for buffering and processing (fddaq-v5.0.0)

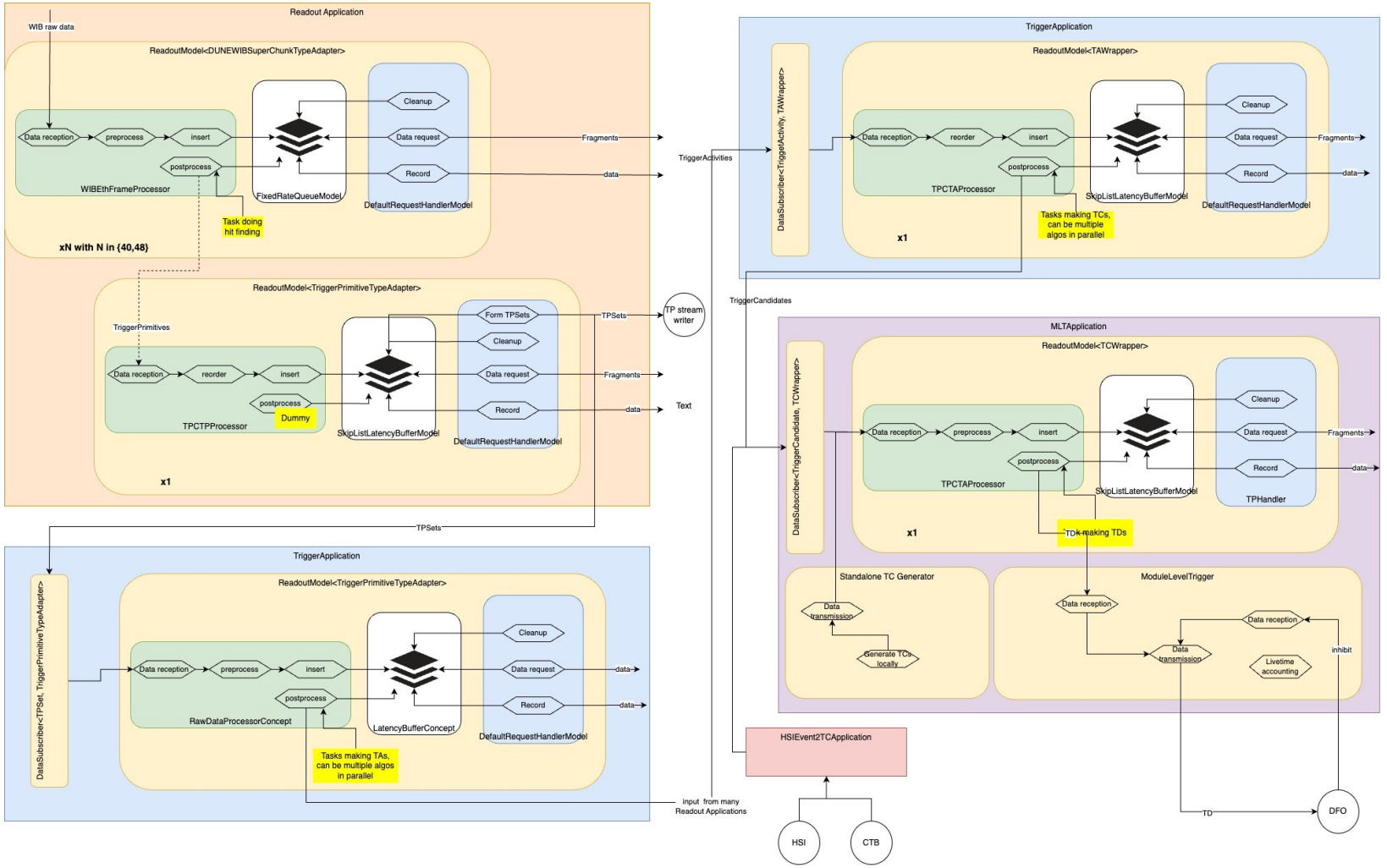


Episode 3- the MLT

- The MLT is not well separated from the other parts of the trigger chain
  - Processing of TCs (flitering, merging, …)
  - Forming of Trigger Decisions and their time/geographical windows
  - Interaction with the DFO and run control to hold/release triggering
  - Livetime accounting
- Mixed with it are
  - The generation of local TCs
  - The buffering of TCs

Episode 3 – the MLT

- For some aspects the MLT application can be similar to the other trigger applications
- We propose to make a first split like this
  - Receive and buffer TCs (and respond to data requests)
  - Generate TCs locally (e.g. random or custom generator)
  - Form TDs according to merging, grouping and bitmasks settings
  - Account for dead-time and send TDs only when they should be sent (running, not paused, DFO not inhibited, …)



Is this the end of refactoring?

- From a trigger software infrastructure point of view we think that now blocks are sufficiently modular with clear scope
- There are aspects getting closer to the trigger function proper that we should start thinking about, e.g.:
  - How can livetime accounting be done correctly?
  - Who should decide which parts of the detector and for which time intervals should be part of the Trigger Record?
    - This is now all done in the MLT, but to me this should be part of the TAs/TCs; the MLT may have rules for forcing enlarged windows, but it is the TCs that know what they triggered on

Let’s discuss