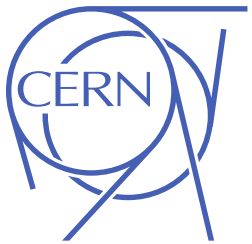
FriDAQ data generator framework

Martin Zemko

DAQFEET workshop

10th February 2021



Outline

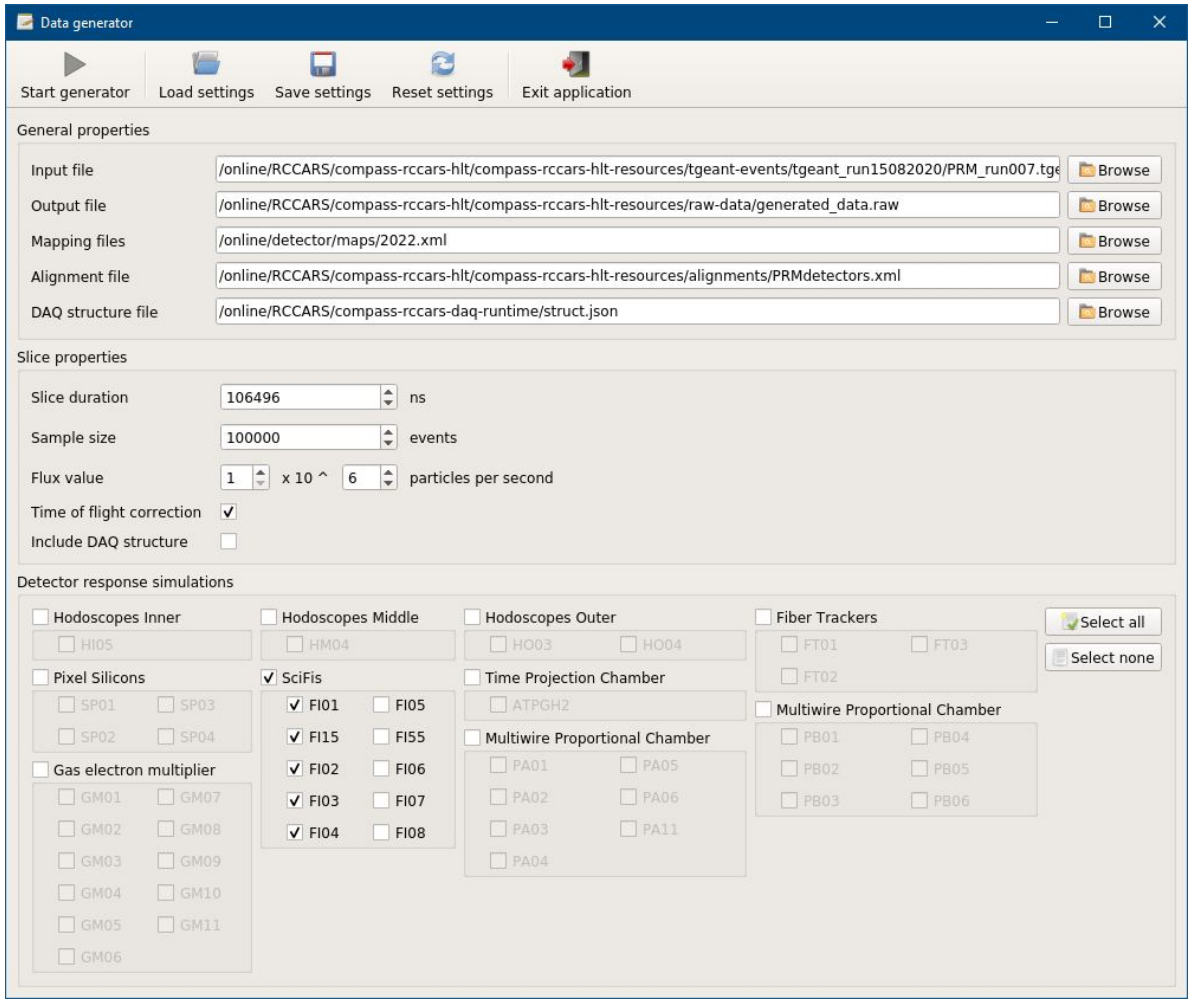
Introduction

- With respect to the agile development of the high level trigger, we need some ways how to test the software and verify its functionality
- Therefore, we need some raw data for tests
- Without an available detector setup, we can generate some physics data using Monte Carlo simulations (TGeant) and similar tools
- In particular for this purpose, the Data generator has been developed

Description of the Data generator

- Data generator is a processing tool that can generate raw data files from Monte Carlo events
- It takes the Monte Carlo events as an input, simulates detector and frontend card responses, and produces data stream in the new DAQ format
- At the output, we obtain raw data chunks containing Monte Carlo events encoded in the new data format
- It provides a graphical user interface as well as a command line interface

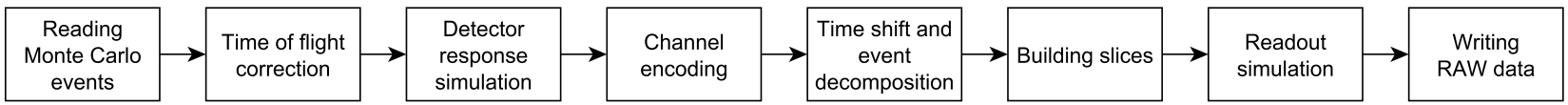
Data generator GUI



Simulation chain

Inside of the data generator, data are processed in several steps:

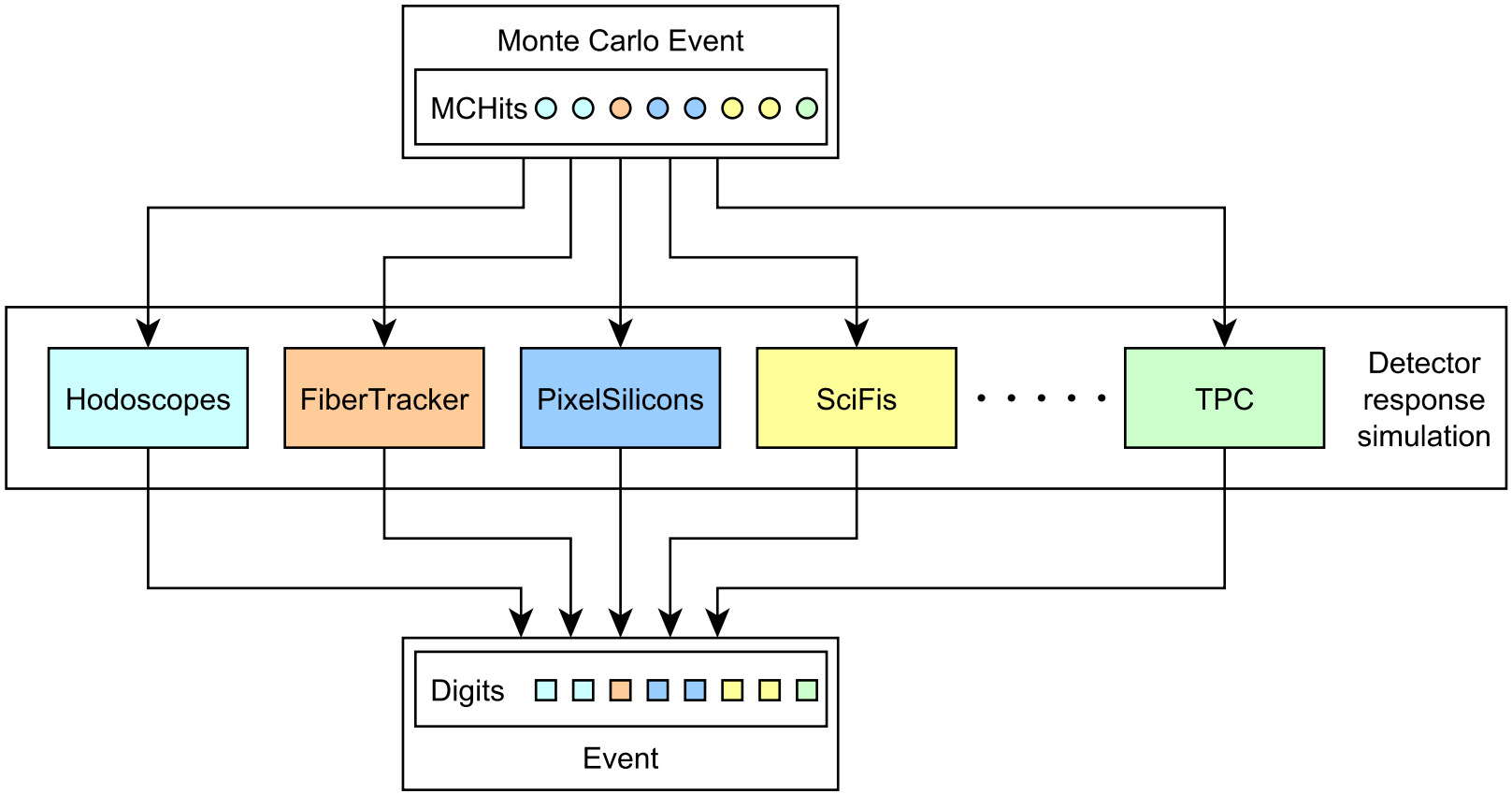
1. **Reading input files** – Monte Carlo events, mapping files, etc.
2. **Correction of time of flight** – if enabled
3. **Detector response simulation** – simulates specific responses of detectors to passing particles
4. **Event decomposition** – events are shifted in order to comply with the beam intensity (flux value) and broken into digits
5. **Slice building** – digits are incorporated into slices
6. **Readout simulation** – digits are converted into specific data words of frontend electronics
7. **Serialization** – slices and images are streamed into a data file



Detector response simulation

- Detector response simulation is a simulation library providing a detector-specific response to passing particles
- This module represents the most important part
- Several detectors are already implemented
- Others are still waiting for their implementations
- However, the basic framework is ready to use
- Implementation of detectors is simple and straightforward

Detector response simulation



Detector response simulation

- Curent status of the development is the following

FiberTracker	almost fully implemented
GEM	basic implementation
Hodoscopes	implemented
MWPC	basic implementation
Pixel Silicon	basic implementation
SciFi	basic implementation
TPC	almost fully implemented

- Thanks to Moritz, Martin, and Christian
- Other detectors are not implemented at all

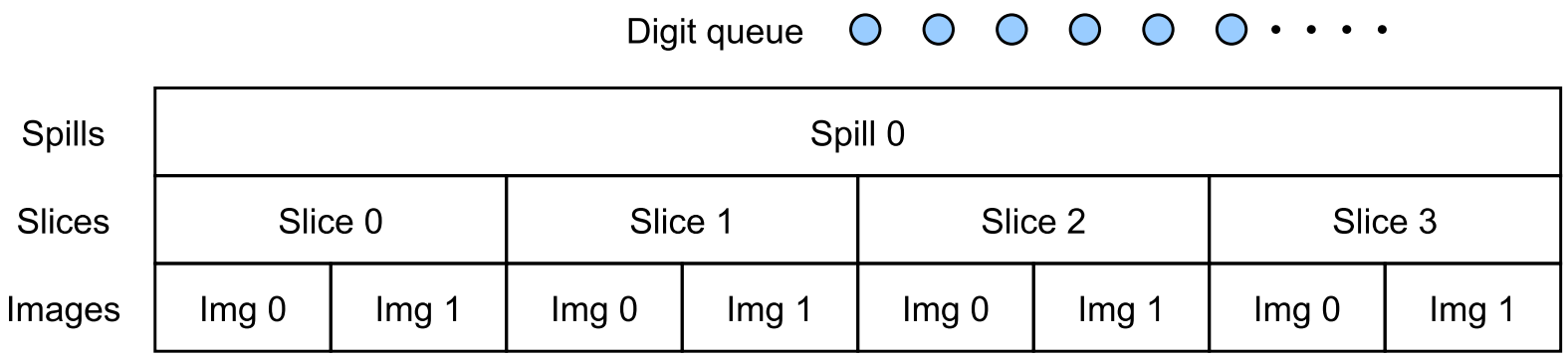
Readout simulation

- Simulates the behaviour of individual frontend cards (chips)
- Such behaviour is specific for each chip, e.g. some digits can be merged into one word, transformed into several data words, compressed, etc.
- Currently, we foresee implementations of several readout cards – TDC, SADC, ALPIDE, etc.
- This simulation generates data words in formats that are specific for these chips, i.e. converts digits into data words

ALPIDE	basic implementation
SADC	not yet implemented
TDC	basic implementation

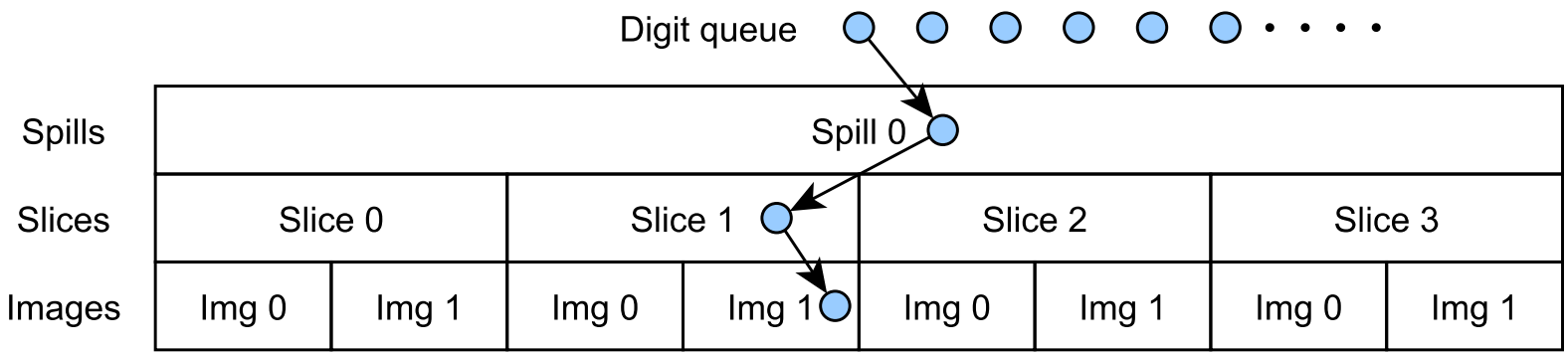
Generating slices I

- Generating slices out of digits is a multi-step procedure
- Process begins with calculations of absolute hit times of digits
- Then, digits are assigned to an appropriate spill, slice, and image
- In this example, we consider only a single level of slices
- In fact, there can be several layers of nested slices



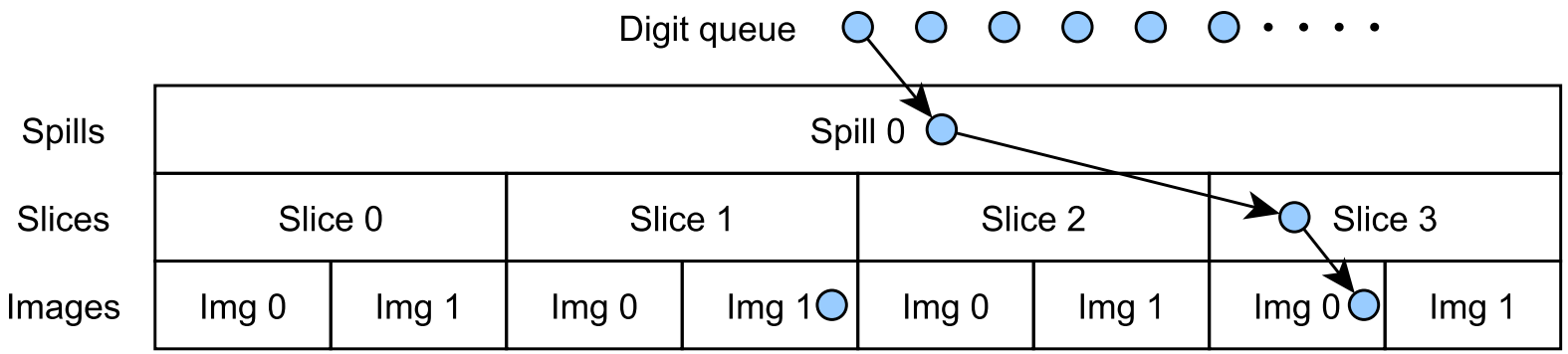
Generating slices II

- Generating of slices is based on the absolute hit times
- Procedure starts at the spill level
- Digit passes from top levels (spills) to the very bottom (images)
- At each level, the correct container on the lower level is chosen; and the digit is moved there



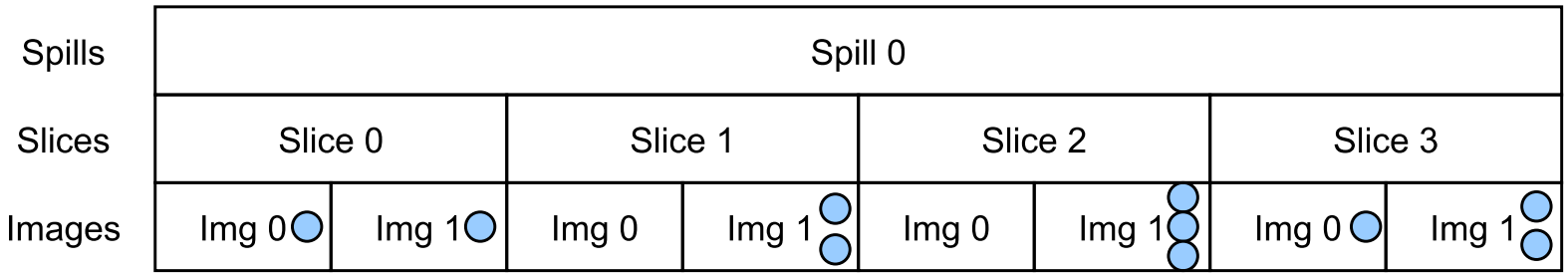
Generating slices III

- This procedure is repeated for every single digit, until there are no more digits left



Generating slices IV

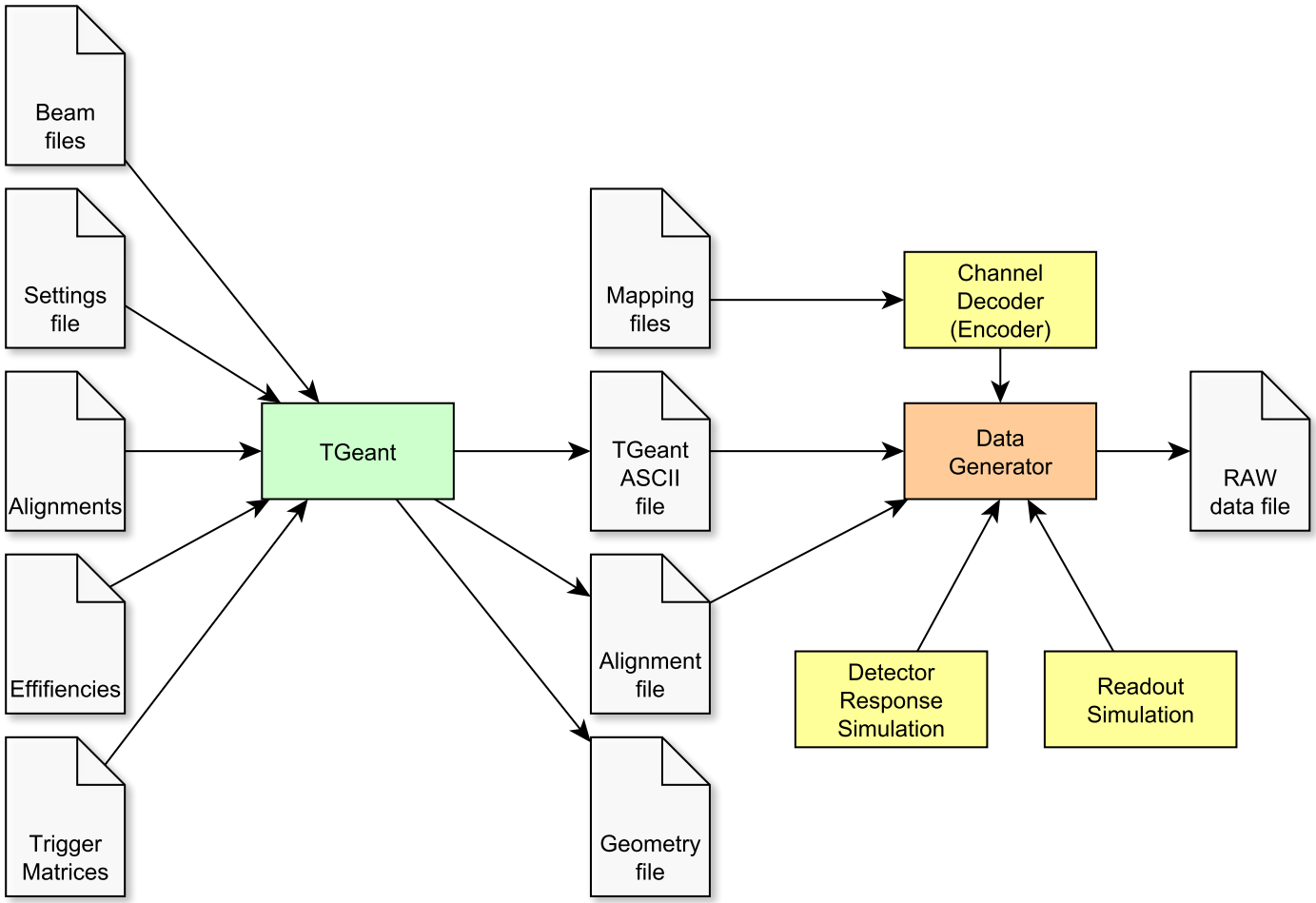
- At the end, we obtain a multilevel structure of containers ready for serialization and streaming
- Empty slices are preserved in the output file
- On the other hand, empty images are omitted



Integration with the TGeant

- Preferably, the data generator can be perceived as the following step (after the TGeant) in the simulation chain
- Together, they can create a full simulation of the detector setup
- Data generator accepts outputs of the TGeant, and it can process them further into raw chunks
- Unfortunately, there is no backtrace to the original Monte Carlo event
- We are still looking into how to include back-pointers to the Monte Carlo data

Integration with the TGeant



Conclusion

- Data generator is a useful tool for generating raw data before the actual data taking
- It can produce trustworthy data that can be triggered by the HLT and reconstructed
- Generator framework is ready to use and can already generate useful data
- Detector response simulation library describes and simulates the behaviour of various detectors; unfortunately, support of many detectors is still missing
- Similarly to the HLT, source codes can be found in the GitLab repository or in the /online folder

Thank you for your attention