Status of BWEC EMC digitization in PandaRoot

Guang Zhao (zhaog@ihep.ac.cn)[1], Oliver Noll[2]

Panda Collaboration Meeting 19/3

5/11/2019 @ GSI

[1] Institute of High Energy Physics (Beijing)

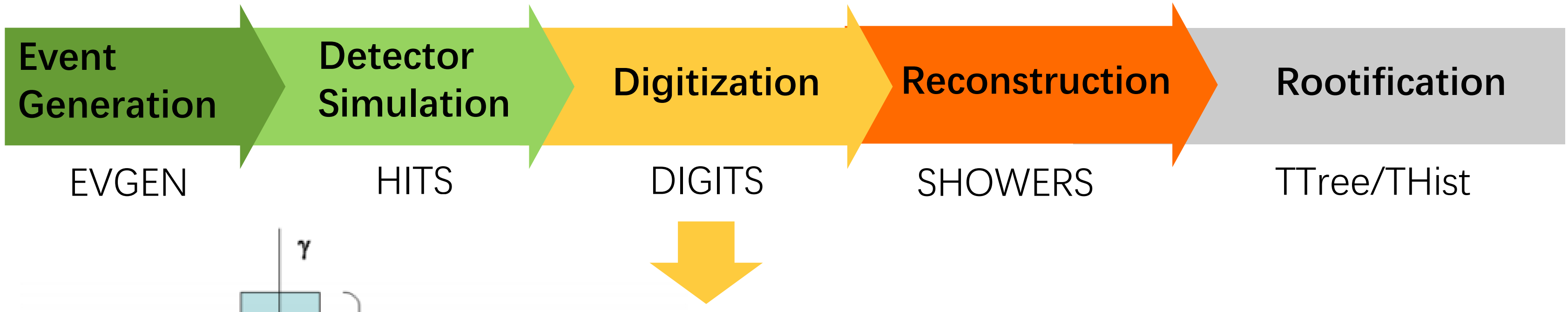
[2] Helmholtz-Institute Mainz

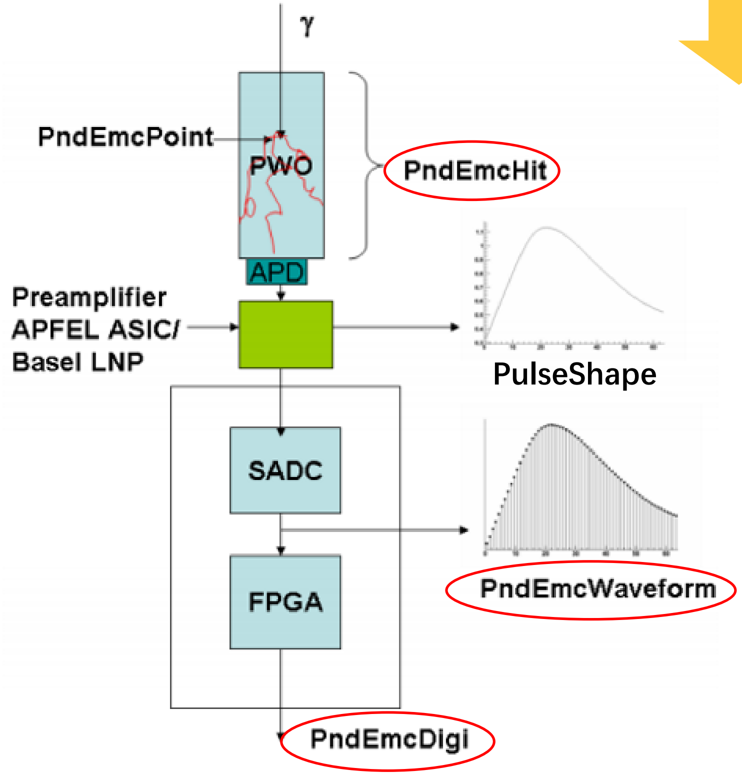
Outline

- **Digitization process update**

- **Code structure update plan**

Digitization process in PandaRoot





Signal Generator (SG)

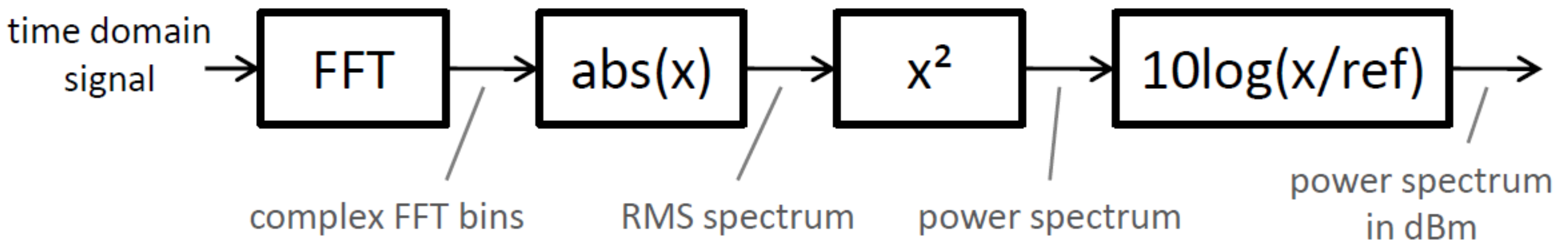
- Creating analog waveforms
- Adding noises and digitizing (UPDATE)
- Be able to produce Pile-up waveforms

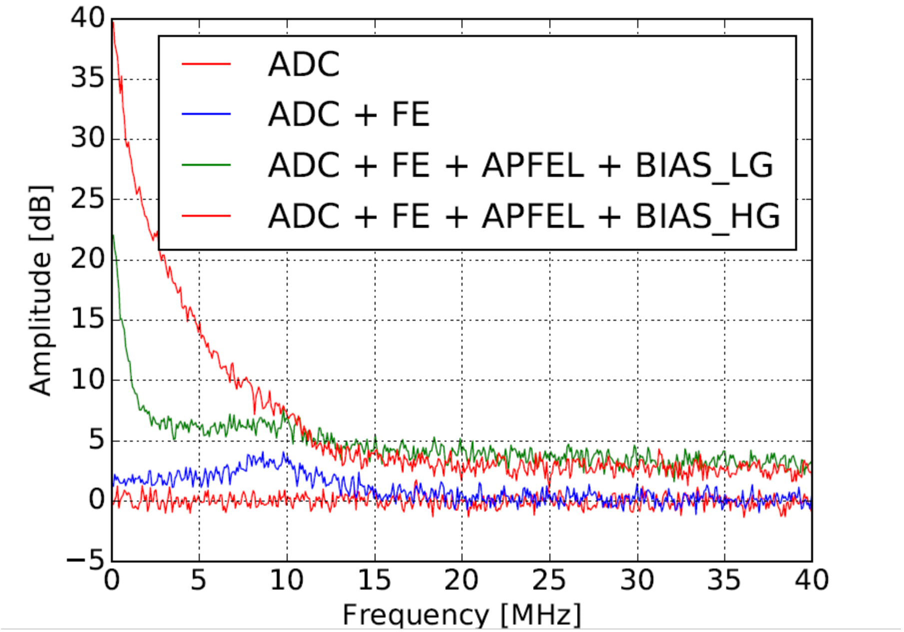
Feature Extraction (FE)

- Hit detection
- Amplitude/time extraction (UPDATE)
- Pile-up recovery

Noise

FFT analysis for the noises





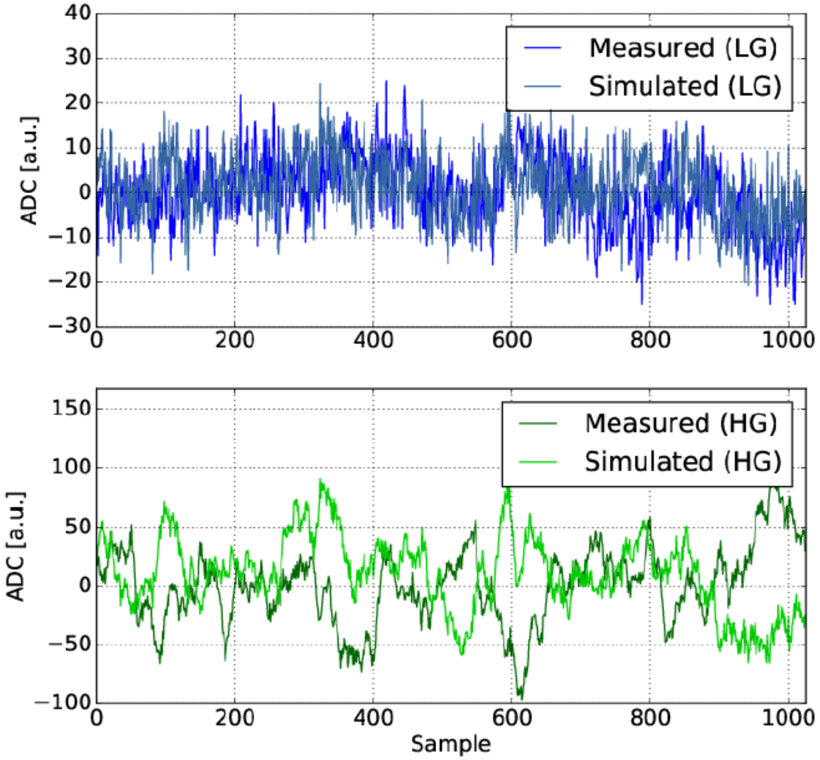
- Noise components
  - Biased APD, APFEL preamplifier at low/high gain
  - Open ADC entrance
  - Front-end electronics transmission
- FTT analysis of the noises
- IFFT of the power spectrum to obtain time-domain noises

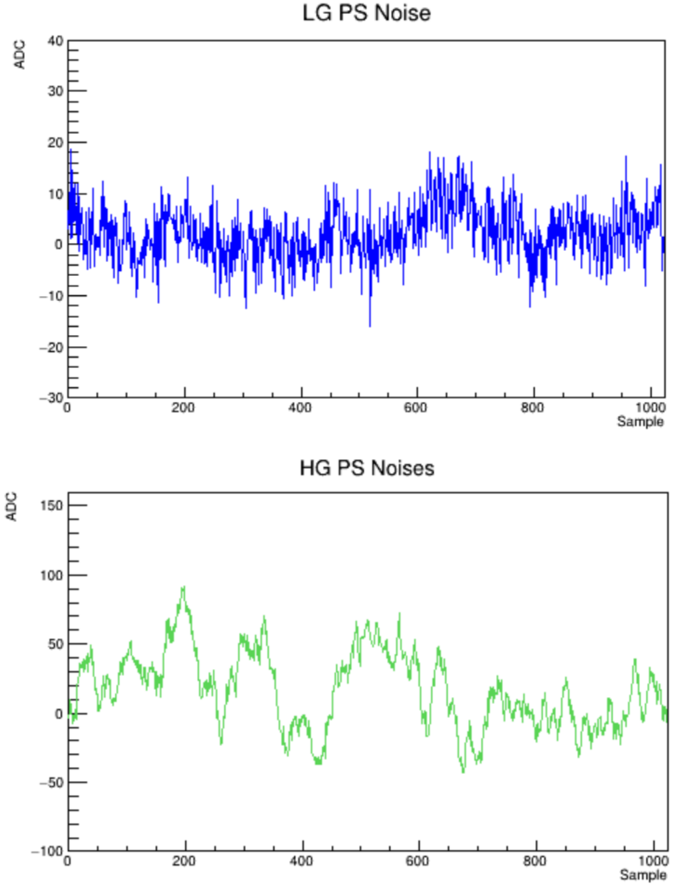
Noise (II)

Biased APD, APFEL preamplifier for low/high gain

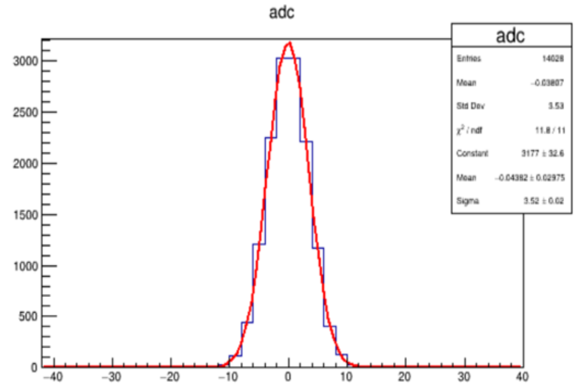
PandaRoot Sim

Measurement

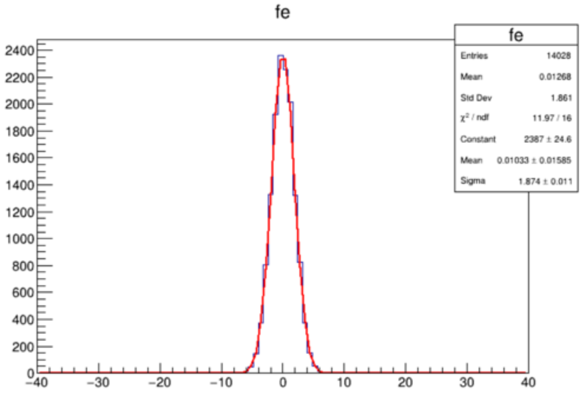




ADC & FE Transmission



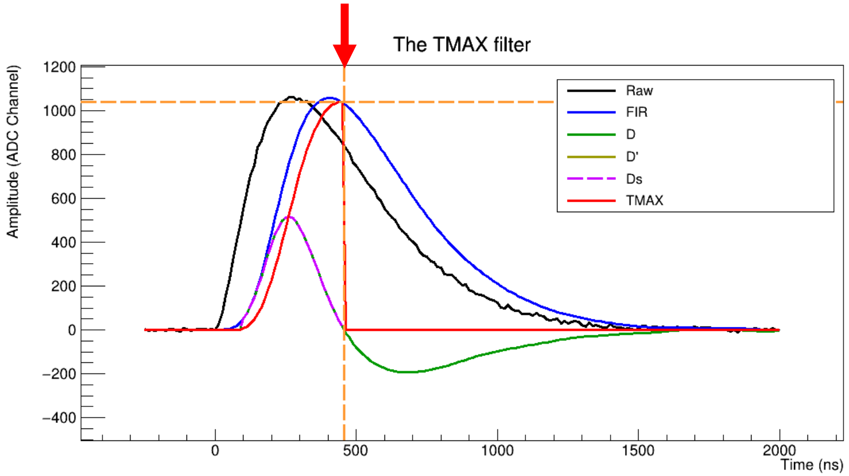
Measured value: 3.5
Simulated value: 3.52 +/- 0.02

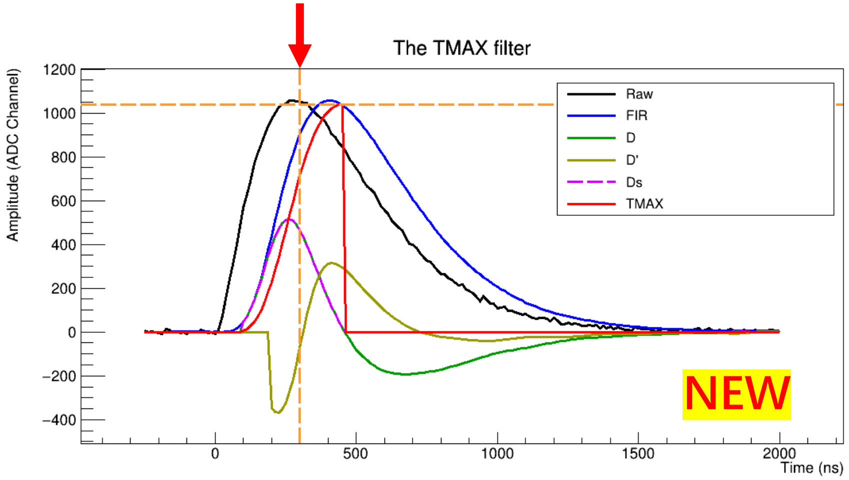


Measured value: 1.89
Simulated value: 1.874 +/- 0.011

✓ **Good agreement between simulation and measurement**

Time extraction

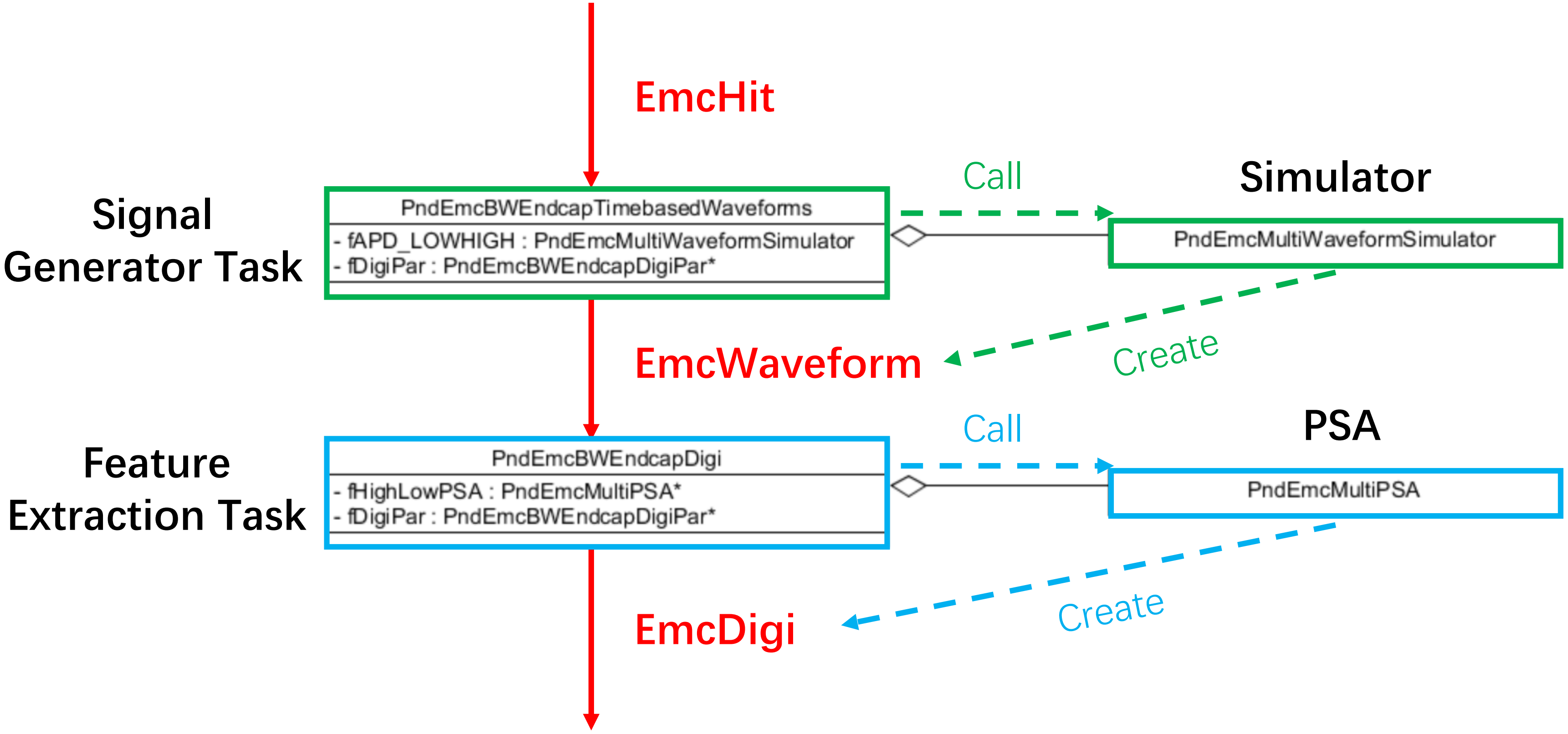




- FIR: 20-coefficient filtering
- Derivative (D) ⟶ time: zero transit of D
  - D[i] = T[i + r] – T[i]
- Second derivative (D') ⟶ time: zero transit of D' (NEW)
  - D'[i] = D[i + r] – D[i]
- TMAX
  - TMAX[i] = Σ_i(D[i]) – Θ[-D[i]] * D[i]
  - where Θ(x) =
    - 0 (x < 0) ⟶ amplitude: TMAX peak
    - 1 (otherwise)

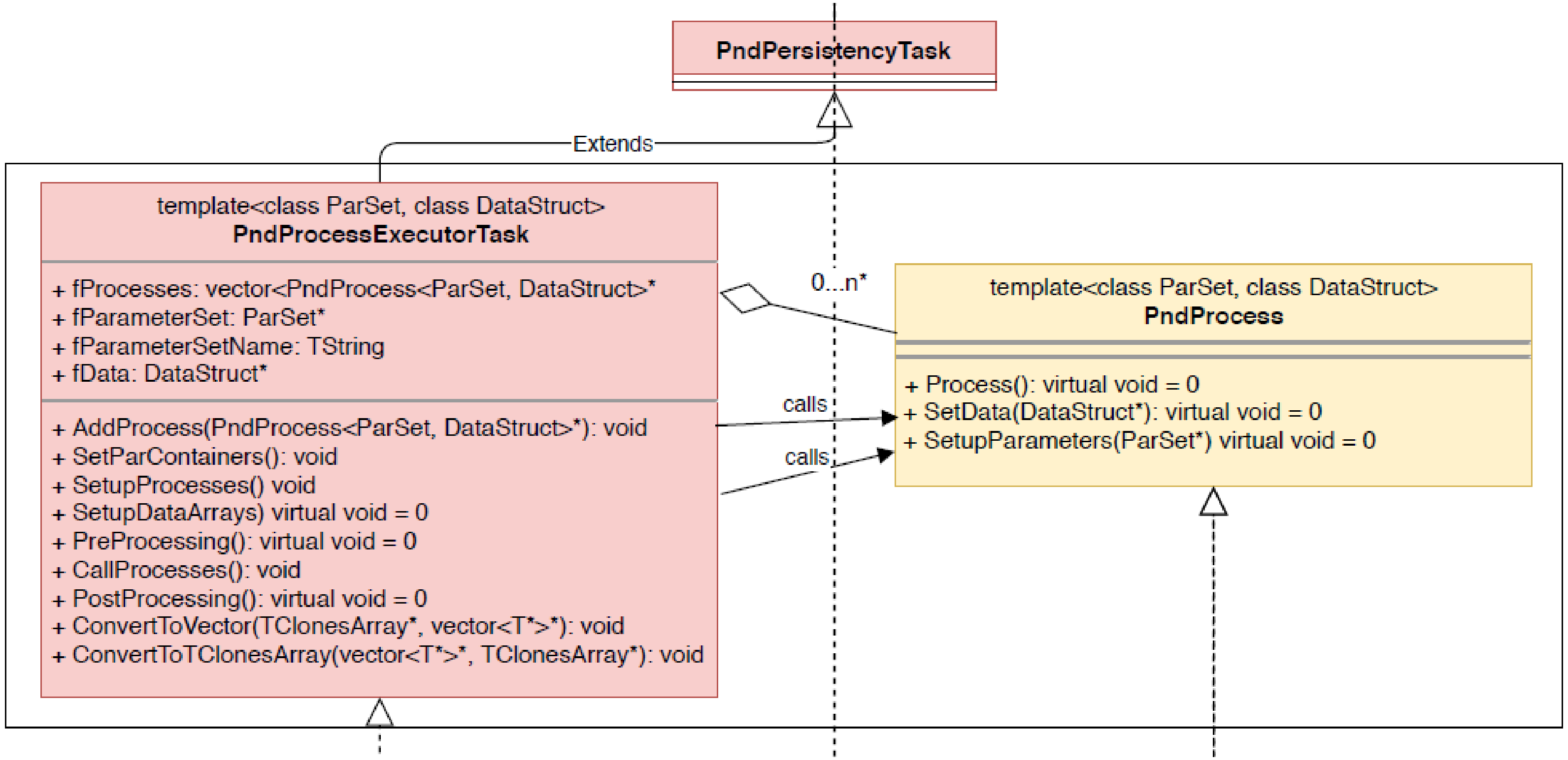
✓ Extracted times are closer to the peak time of the raw waveforms

Current BWEC digitization framework



- Two tasks for signal generator and feature extraction respectively
- Simulator for creating waveforms from hits
- Pulse Shape Analyser (PSA) for creating digis from waveforms

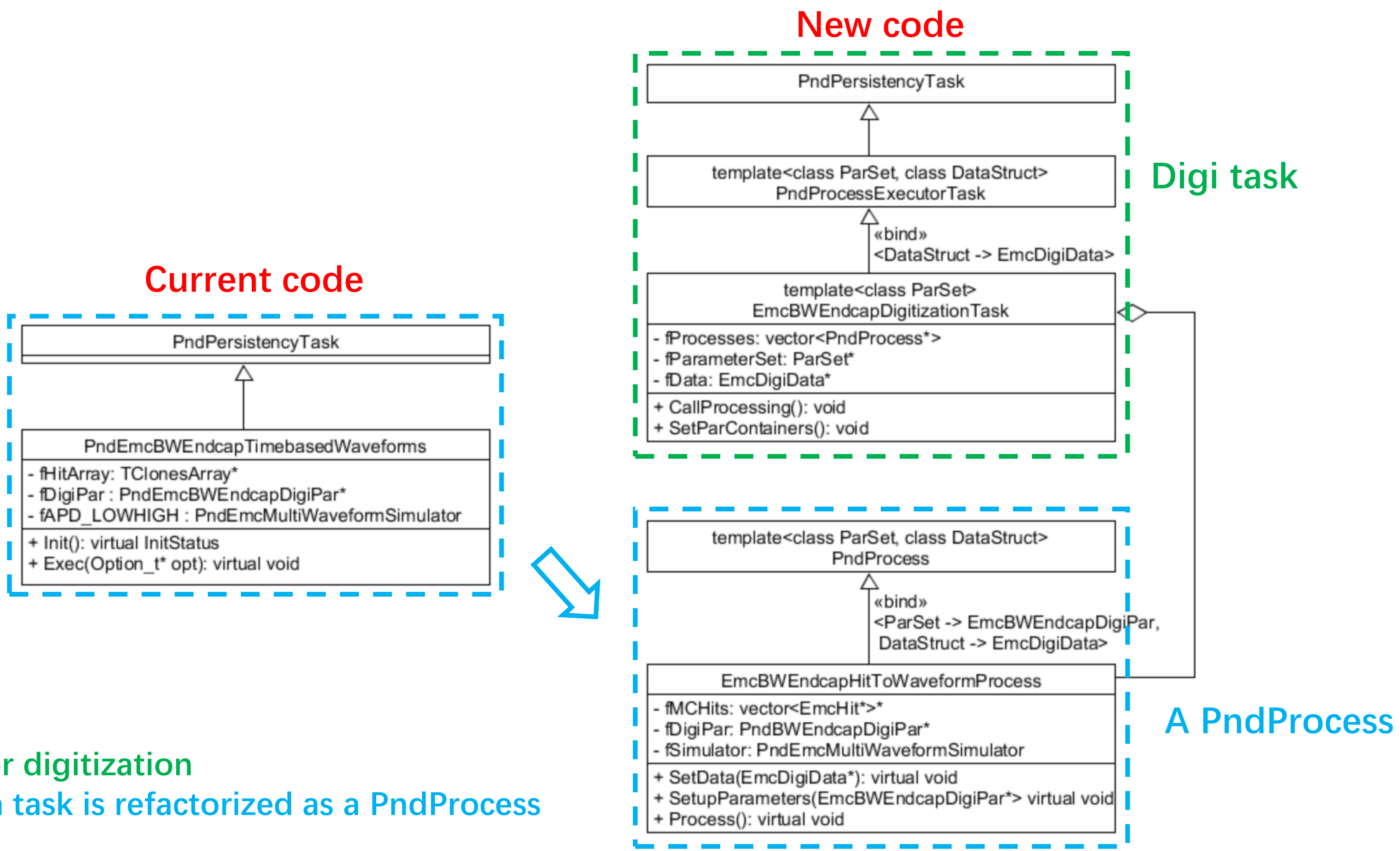
The new "ExecutorTask" framework



- A new OO design
- Tasks that do similar jobs are encapsulated into a single "PndProcessExecutorTask" (e.g. there should be a single EmcDigiTask)
- The encapsulated tasks now become several "PndProcess"s

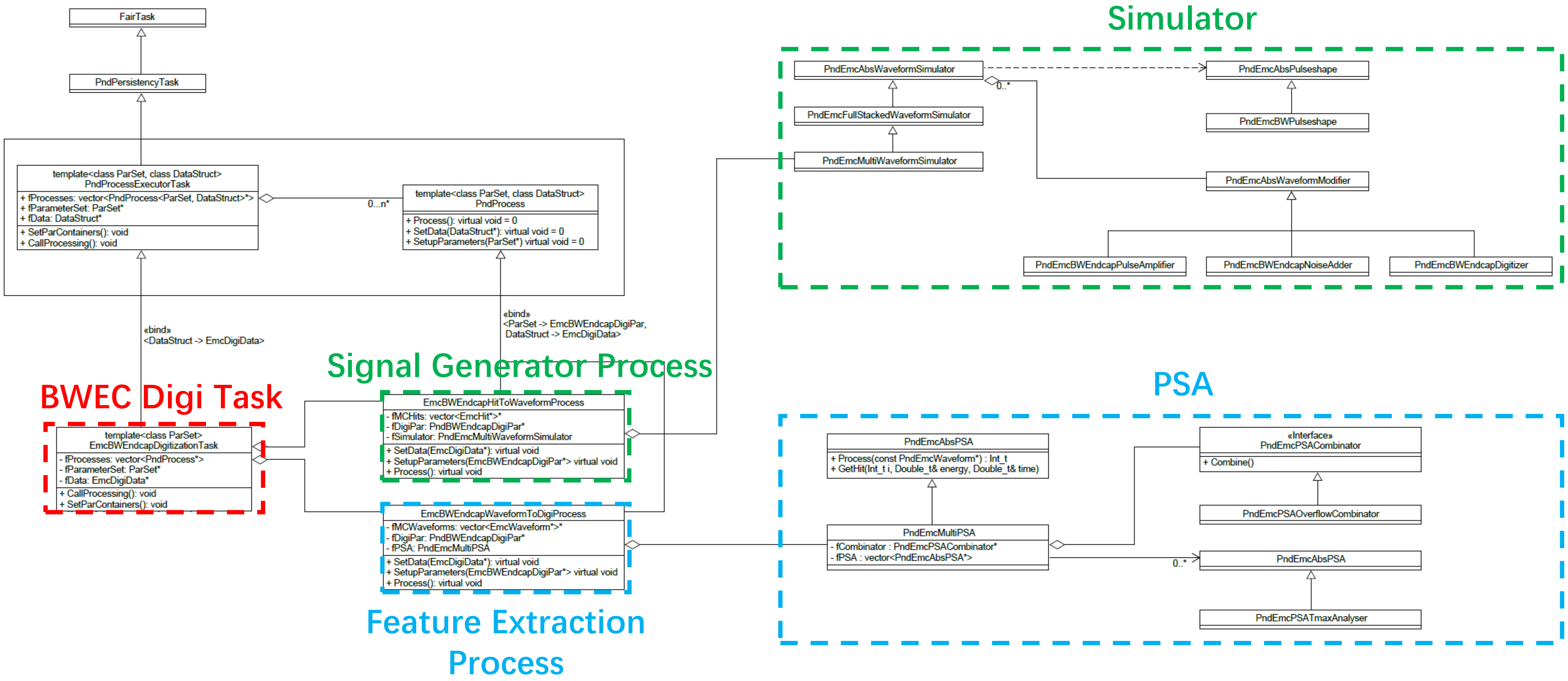
From Ben Salisbury
See his talk for details

Code refactor (signal generator as an example)



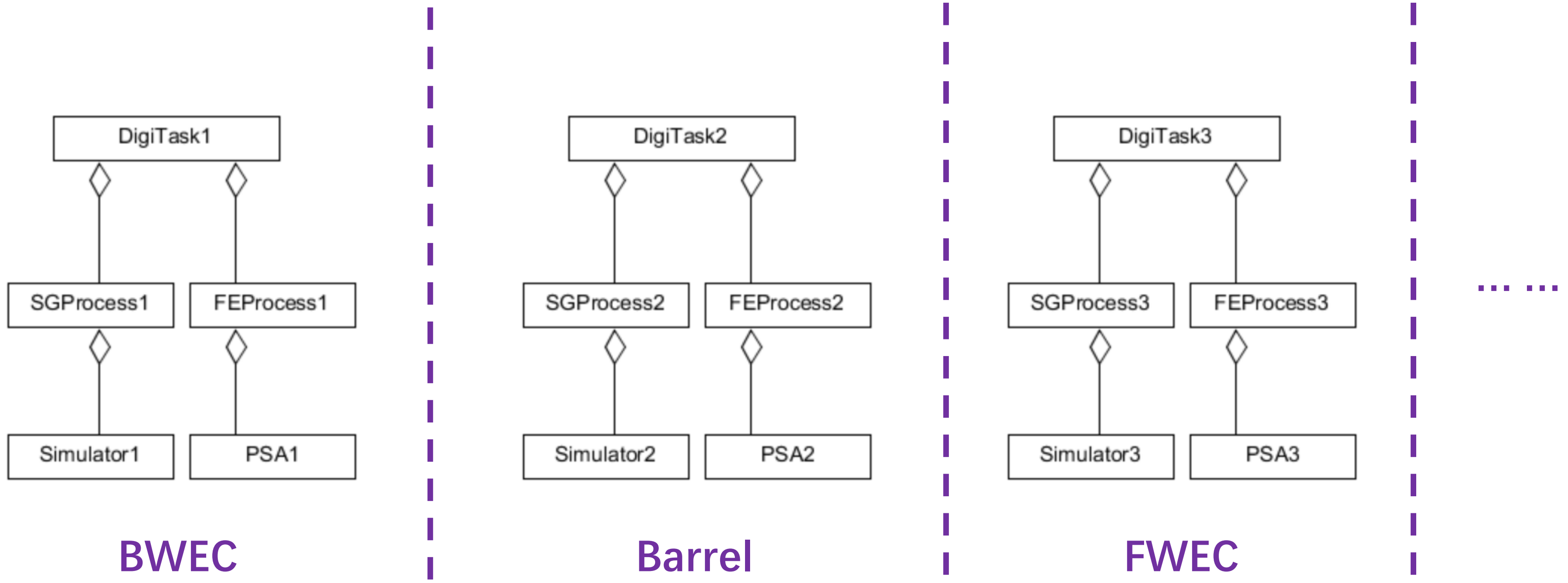
- ✓ Create a new task for digitization
- ✓ Previous digitization task is refactorized as a PndProcess

Code structure for BWEC digitization



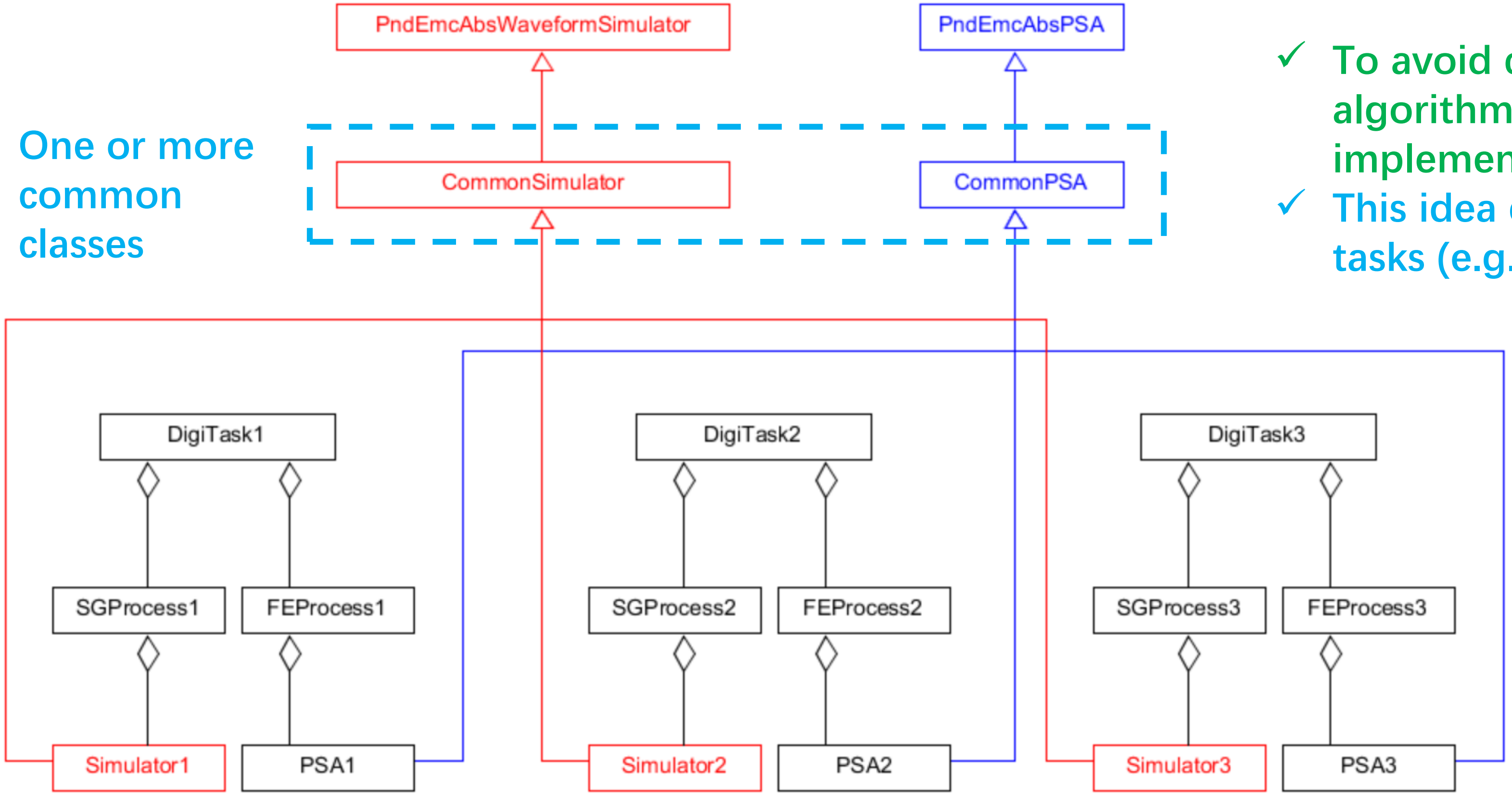
Digitization for different EMC modules

- ✓ In the new framework, for EMC, simulation are separated for EMC modules. Therefore, like BWEC, we should have several more “DigiTasks”
- ✓ For each task, it handles the data objects from its own sub-detector
- ✓ The simplified diagram are shown below (The “class names” and “structures” are also simplified for demonstration)



Class for common functionality among modules

One or more common classes



- ✓ To avoid code duplication, common algorithms between modules are implemented in one or more base class
- ✓ This idea can also be applied to other tasks (e.g. clustering task)

…… ……

Summary and outlook

- **Summary**
  - Done some minor updates for the BWEC EMC digitization
    - Noise normalization
    - Time extraction
  - Proposed an updated digitization framework based on the “ExecutorTask” framework

- **Plan**
  - Keep optimizing the bwec digitization algorithm. For now, FFT is slow. Try to find a better noise modeling method.
  - Will implement the two-APD version digitization algorithm (can reduce noise rate).
  - Will check in the code with the old framework. Plan to migrate to the new framework in collaboration with Ben from Bonn University.

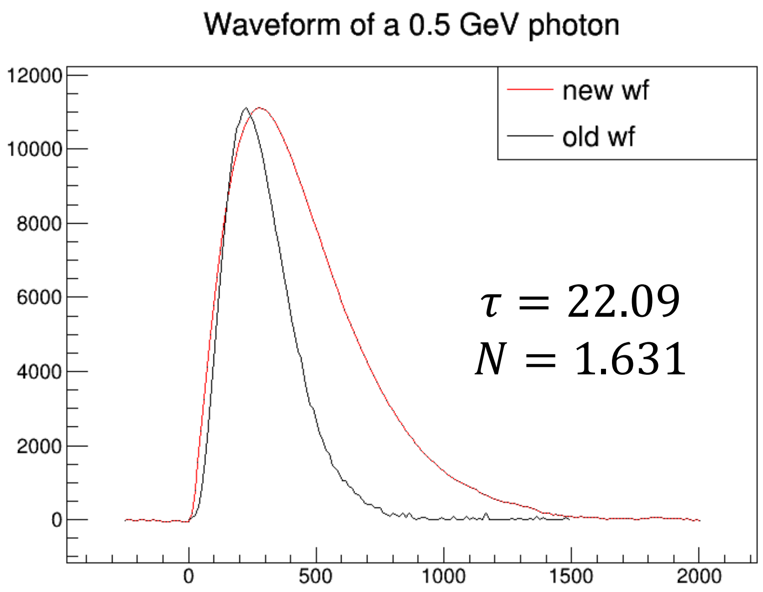
Backups

Pulse shape

$$f(x) = -A \cdot e^{\frac{-N(x-\delta)}{\tau}} \cdot \left(\frac{x-\delta}{\tau}\right)^N \tag{2.1}$$

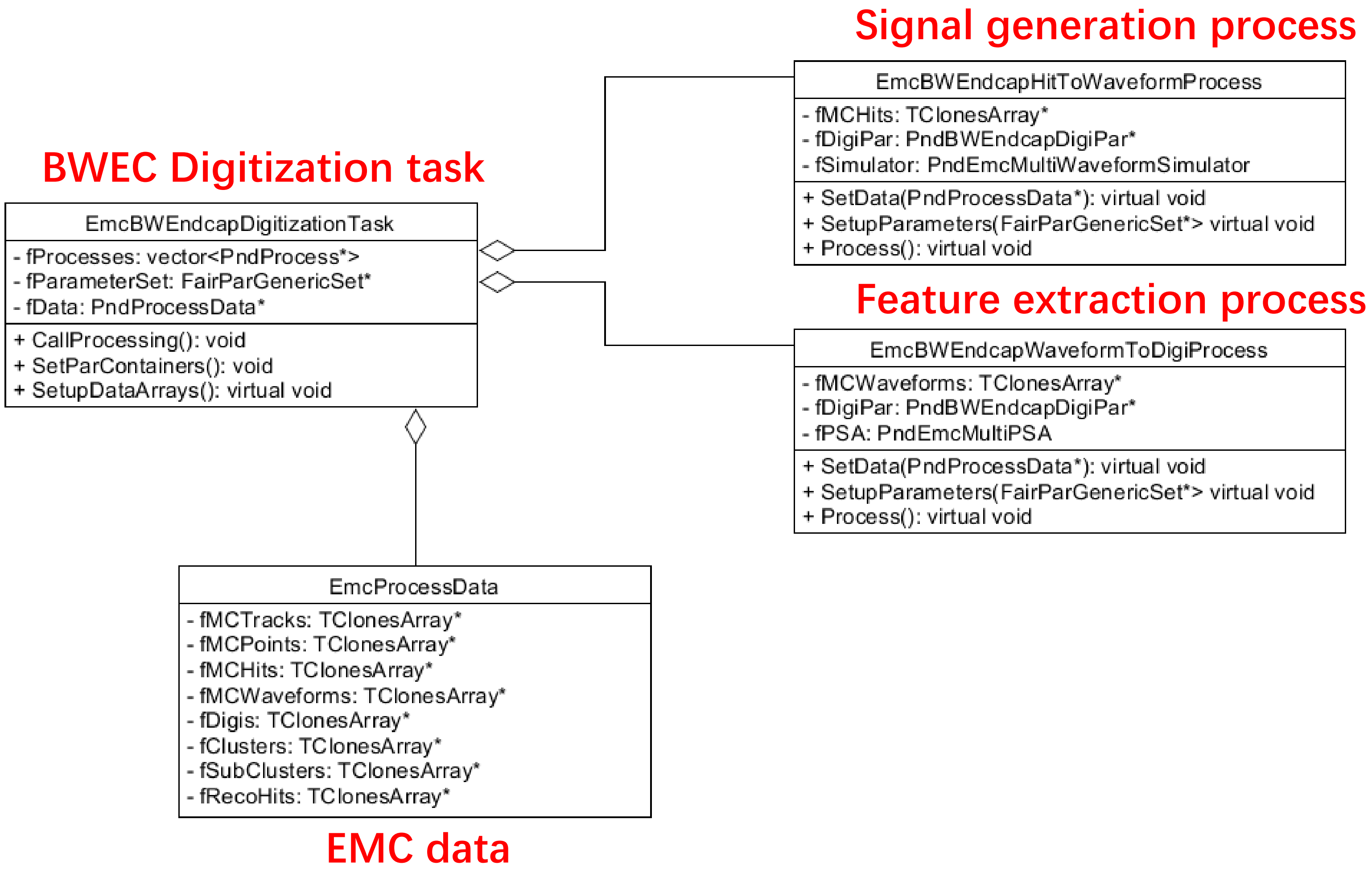
Whereby τ is describing the decay behavior. N has an impact on the rising and decay ratio. δ shifts the pulse in time. A is proportional to the pulse hight H:

$$A = H \cdot e^N \tag{2.2}$$

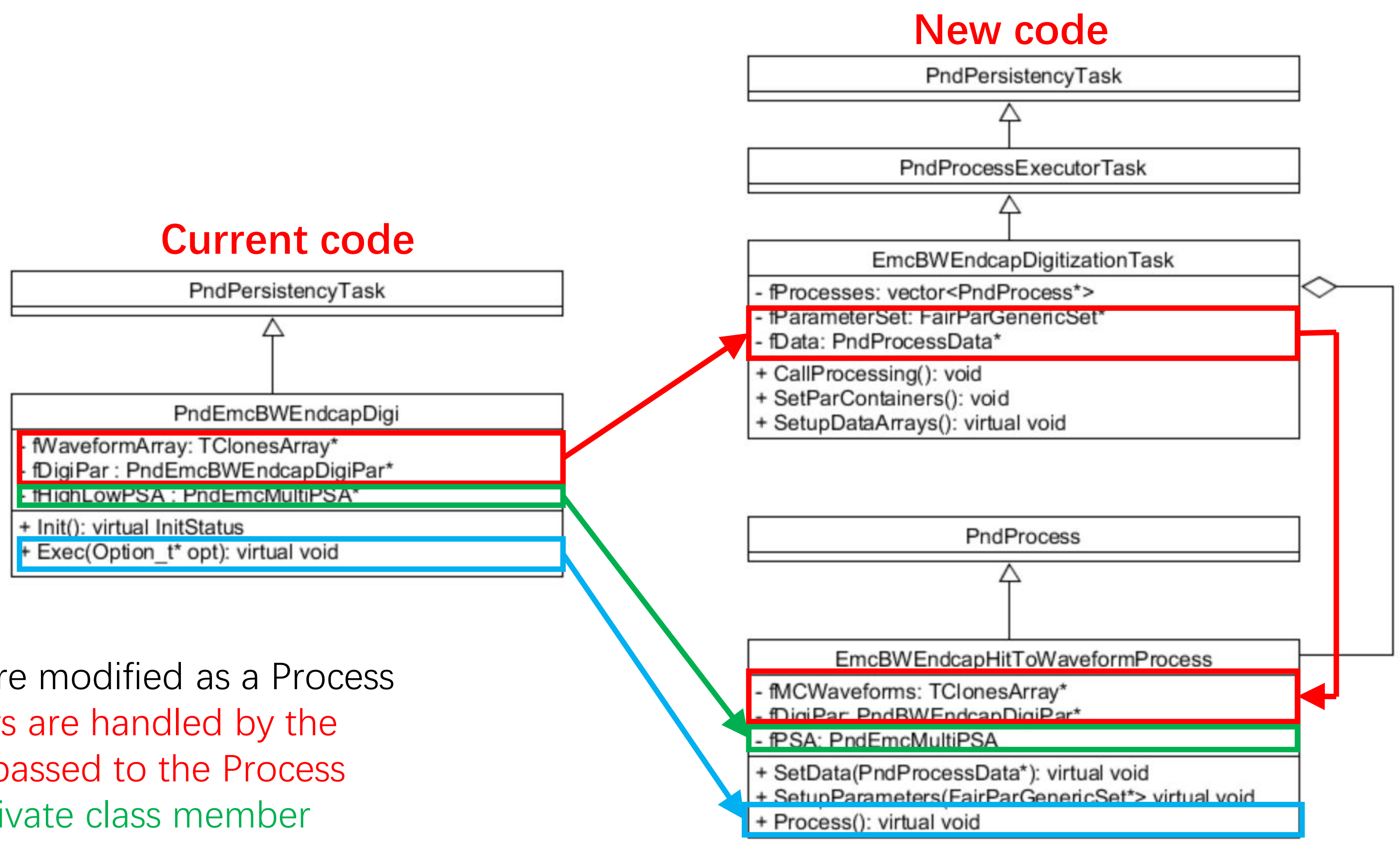


- APFEL ASIC pulse digitized by the SADC
- Two gains: HG/LG = 10.5
- Full pulse width: ~1700 ns
- Rising time: ~300 ns

An implementation for the BWEC EMC



Refactor of the feature extraction code



- ✓ The previous Task are modified as a Process
- ✓ Data and parameters are handled by the ExecutorTask, then passed to the Process
- ✓ PSA remains as a private class member
- ✓ Content of Exec() moves to Process()