

BIC Simulation Flow

1. GEANT4 Simulation (edm4hep::SimCalorimeterHit)

- Simulated calorimeter hit – edm4hep::SimCalorimeterHit
- Storing hits per fiber in 1cm z-slices for each grid cell
- Each hit has contributions that could come from separate particles

2. Regrouping (edm4hep::SimCalorimeterHit)

- edm4hep::SimCalorimeterHit storing one hit per MCParticle for each grid cell
- One hit contribution per z-slice only: Each hit contribution sums together all contributions for a z-slice belonging to the main MCParticle. This also encodes a time series.

3. Attenuation (edm4hep::SimCalorimeterHit)

- Same structure as Regrouping, but attenuated to the end of the detector
- Will store 2 structure corresponding to the positive and negative ends of the detector

```
#----- CaloHitContribution
edm4hep::CaloHitContribution:
  Description: "Monte Carlo contribution to SimCalorimeterHit"
  Author: "F.Gaede, DESY"
  Members:
    - int32_t PDG //PDG code of the sho
    - float energy //energy in [GeV] of the
    - float time //time in [ns] of this co
    - edm4hep::Vector3f stepPosition //position of this energy
  OneToOneRelations:
    - edm4hep::MCParticle particle //primary MCParticle that

#----- SimCalorimeterHit
edm4hep::SimCalorimeterHit:
  Description: "Simulated calorimeter hit"
  Author: "F.Gaede, DESY"
  Members:
    - uint64_t cellID //ID of the sensor that created this
    - float energy //energy of the hit in [GeV]
    - edm4hep::Vector3f position //position of the hit in w
  OneToManyRelations:
    - edm4hep::CaloHitContribution contributions //Monte Carlo
```

BIC Simulation Flow (continued)

1. **GEANT4 Simulation** (edm4hep::SimCalorimeterHit)
2. **Regrouping** (edm4hep::SimCalorimeterHit)
3. **Attenuation** (edm4hep::SimCalorimeterHit)

Option 1

4. **Conversion to pulse**
(edm4eic::SimCalorimeterPulse)
 - One pulse per MCParticle
 - SiPM waveform library
5. **Combine pulses**
(edm4eic::SimCalorimeterPulse)
6. **Fold in noise pulses here?**
7. **Digi:**
edm4eic::SimCalorimeterPulse
→ edm4eic::RawHGCROCHit

Option 2

4. **Don't use pulses**
(edm4eic::SimCalorimeterHit)
5. **Combine hits**
(edm4eic::SimCalorimeterHit)
6. **Fold in noise hits**
 - Harder to do as we'd have to store it as CaloHitContribution
7. **Digi:**
edm4eic::SimCalorimeterHit
→ edm4eic::RawHGCROCHit
 - This algorithm becomes heavy as it needs to do all pulse shapes + noising + combining

SimCalorimeterPulse initial proposal

```

244  ## =====
245  ## Calorimetry
246  ## =====
247  + edm4eic::SimCalorimeterPulse:
248  +   Description: "Simulated calorimeter pulse for single-particle waveforms prior to digi."
249  +   Author: "S. Joosten"
250  +   Members:
251  +       - uint64_t      cellID      // ID of the readout cell for this pulse.
252  +       - float         energy      // Total energy for this pulse in [GeV].
253  +       - float         position    // Position of the hit in world coordinates [mm].
254  +       - float         time        // Start time for the pulse in [ns].
255  +       - float         interval    // Time interval between amplitude values [ns].
256  +   VectorMembers:
257  +       - float         amplitude   // Pulse amplitude in [GeV], sum of amplitude values equals total energy
258  +   OneToOneRelations:
259  +       - edm4hep::MCParticle MCParticle // MCParticle that caused the pulse
260  +   OneToManyRelations:
261  +       - edm4hep::SimCalorimeterHit hits // SimCalorimeterHits used to create this pulse
262  +

```

■ Modifications to consider:

<https://github.com/eic/EDM4eic/pull/106>

- OneToMany to SimCalorimeterPulse?
- OneToMany for MCParticle to allow for combining pulses?
- Or can we do without this structure?

SimCalorimeterPulse current proposal

```
## =====
## Calorimetry
## =====

edm4eic::SimCalorimeterPulse:
  Description: "Simulated calorimeter pulse prior to digitization."
  Author: "D. Anderson, S. Joosten"
  Members:
    - uint64_t      cellID      // ID of the readout cell for this pulse.
    - float         energy      // Total energy for this pulse in [GeV].
    - float         position    // Position of the hit in world coordinates [mm].
    - float         time        // Start time for the pulse in [ns].
    - float         interval    // Time interval between amplitude values [ns].
  VectorMembers:
    - float         amplitude    // Pulse amplitude in [GeV], sum of amplitude values equals total energy
  OneToManyRelations:
    - edm4hep::SimCalorimeterHit hits // SimCalorimeterHits used to create this pulse
    - edm4hep::SimCalorimeterPulse pulses // SimCalorimeterPulses used to create this pulse
    - edm4hep::MCParticle MCParticle // MCParticle that caused the pulse
```

<https://github.com/eic/EDM4eic/pull/106>

Raw CALOROC Hit | Summary

- [edm4eic#101](#): adds raw hit from HGC/CALOROC chip
 - Conversation now largely converged!
 - **For discussion:** possible change before merge
 - › **Change amplitudes from vectors to array + template no. of samples**
 - › Is this supported by our PODIO version?
 - › Any template examples?
 - Has implications for analyzers downstream...

```
## =====  
## Calorimetry  
## =====  
  
edm4eic::RawCALOROCHit:  
  Description: "Raw hit from a CALOROC/HGCROC chip"  
  Author: "D. Anderson, S. Joosten, N. Novitzky"  
  Members:  
    - int32_t type           // Chip type, 0 - type 1A, 1 - type 1B  
    - uint64_t cellID        // Detector specific (geometrical) cell id  
    - int32_t samplePhase    // Phase of samples in [# samples], for synchronizing across chips  
    - int32_t timeStamp      // [TDC]  
  VectorMembers:  
    - int32_t amplitude      // Iff type == 0, amplitudes of waveforms are stored here [adc counts]  
    - int32_t lowGainAmplitude // Iff type == 1, low gain amplitudes of waveforms are stored here [adc counts]  
    - int32_t highGainAmplitude // Iff type == 1, high gain amplitudes of waveforms are stored here [ac counts]  
    - int32_t timeOfArrival   // Calculated times of arrival [TDC]  
    - int32_t timeOfThreshold // Calculated times over threshold [TDC]  
  ExtraCode:  
    declaration: "  
      bool isType1A() const {return getType() == 0;}\n  
      bool isType1B() const {return getType() == 1;}\n  
    "
```

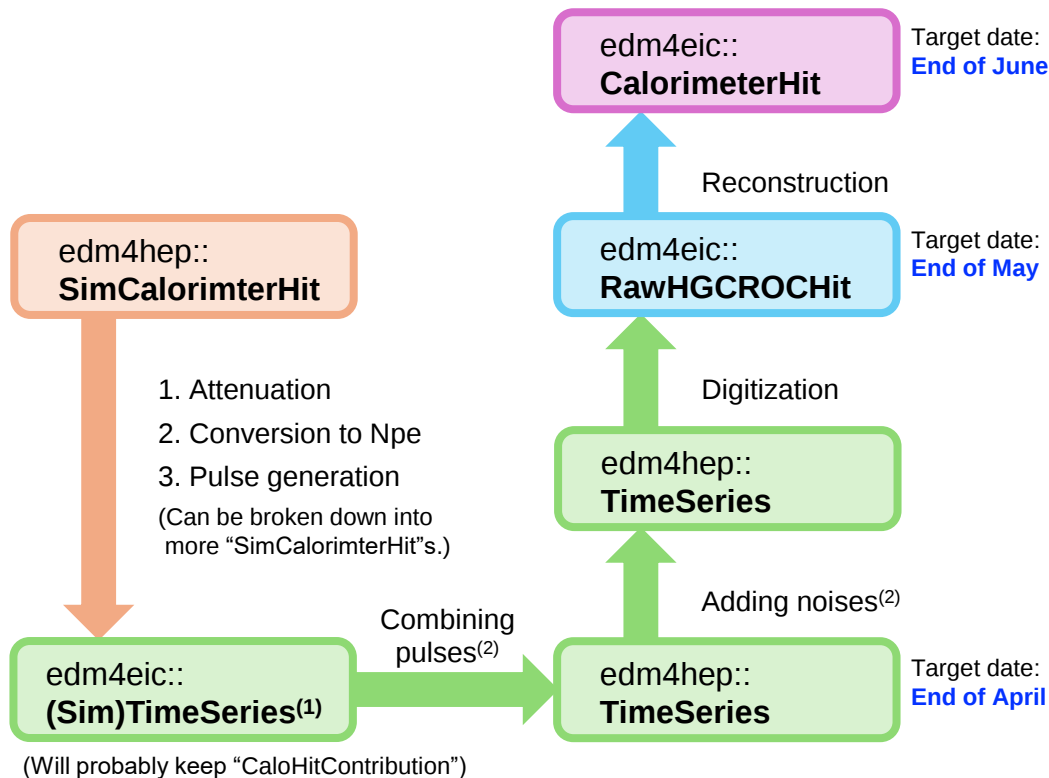
Backup | SimCalorimeterPulse vs. TimeSeries



```
## =====  
## Calorimetry  
## =====  
edm4eic::SimCalorimeterPulse:  
  Description: "Simulated calorimeter pulse prior to digitization."  
  Author: "D. Anderson, S. Joosten"  
  Members:  
    - uint64_t      cellID      // ID of the readout cell for this pulse.  
    - float         energy      // Total energy for this pulse in [GeV].  
    - float         position    // Position of the hit in world coordinates [mm].  
    - float         time        // Start time for the pulse in [ns].  
    - float         interval    // Time interval between amplitude values [ns].  
  VectorMembers:  
    - float         amplitude   // Pulse amplitude in [GeV], sum of amplitude values equals total energy  
  OneToManyRelations:  
    - edm4hep::SimCalorimeterHit hits // SimCalorimeterHits used to create this pulse  
    - edm4hep::SimCalorimeterPulse pulses // SimCalorimeterPulses used to create this pulse  
    - edm4hep::MCParticle MCParticle // MCParticle that caused the pulse
```

```
edm4hep::TimeSeries:  
  Description: "Calibrated Detector Data"  
  Author: "EDM4hep authors"  
  Members:  
    - uint64_t cellID // cell id  
    - float time [ns] // begin time  
    - float interval [ns] // interval of each sampling  
  VectorMembers:  
    - float amplitude // calibrated detector data  
  
edm4hep::RawTimeSeries:  
  Description: "Raw data of a detector readout"  
  Author: "EDM4hep authors"  
  Members:  
    - uint64_t cellID // detector specific cell id  
    - int32_t quality // quality flag for the hit  
    - float time [ns] // time of the hit  
    - float charge [fC] // integrated charge of the hit  
    - float interval [ns] // interval of each sampling  
  VectorMembers:  
    - int32_t adcCounts // raw data (32-bit) word at i
```

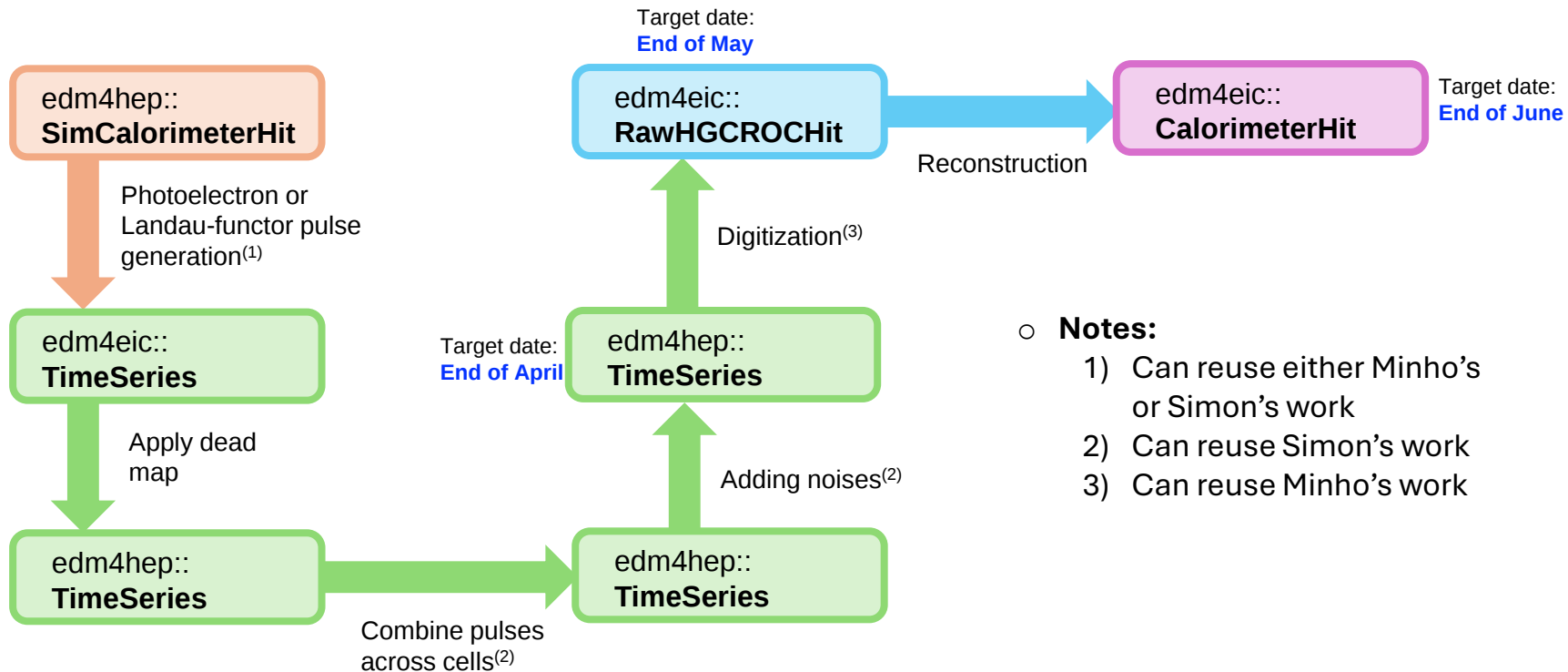
Backup | Updated BIC plan



Notes:

- 1) Possible new addition to data model, will develop proposal
- 2) Can reuse Simon's work for pulse combination, adding noise

Backup | Updated LFHCAL plan



Notes:

- 1) Can reuse either Minho's or Simon's work
- 2) Can reuse Simon's work
- 3) Can reuse Minho's work