TDR Input for Software

- **For discussion:** C/S team interested in any additional input on TDR needs, **particularly the data model**
  - Will summarize discussion and identified input at **Feb. 21st C/S meeting**
  - A summary of what's in the data model on the calo. side is in backup

- **Above:** summary of identified data model and reconstruction needs/wants from January CM
  - c.f. this summary of the CM discussion for more details!

Identified Data Model Needs

- Improved truth-Cluster connections
- ☞ Anything else?

Identified Reconstruction Needs/Wants

- Clustering implemented in all systems
- Cluster splitting/merging
- ML Integration
- Digitization noise, noise-masking and system-specific digitization model implementations
- Better neutral identification
- Easier access to janadot output

Identified Simulation Needs/Wants

- Enhanced realism in BEMC implementation and implementation of end-of-sector box material
- Dedicated studies of HGCROC vs. waveform digitizer in BEMC
- Physics-driven performance studies for nHCal
- Update ZDC default to SiPM-on-tile
- Enhanced realism in pECal implementation

TDR Input for Software | Splitting/Merging Proposal

- **Cluster splitting/merging identified as a TDR need in reconstruction**
  - A splitting algorithm exists for Island clustering algorithm used by majority of calo.s
    - › Only used by certain calo.s (see backup)
    - › May not be appropriate for others
  - No merging algorithm exists in EICrecon, though
    - ☞ Has physics implications, e.g. in electron ID

- **Proposal:** a task force of 3 – 5 people focused on addressing this need
  - Task force charge:
    1) Implement baseline cluster merging algorithm,
    2) Identify cluster splitting needs, and
    3) Implement alternative splitting algorithm where needed
  - Possible timeline outlined to the right

Possible Timeline

- **By 02.16:**
  - › task force formed
- **By 03.01:**
  - › outline of algorithm in place,
  - › splitting needs identified
- **By 03.22:**
  - › EICrecon implementation ready
- **By 04.31:**
  - › fixes/tweaks identified and implemented

Backup | What Calo.s Have Splitting On?

Splitting: On

- EEEMCal
- nHCal
- pECal
- FHCal Insert
- Lumi. Spec.
- ZDC

Splitting: Off

- B0 ECal
- BEMC (ScFi)
- BHCal
- pECal Insert
- LFHCal

○ **Splitting algorithm for Island Clustering:**

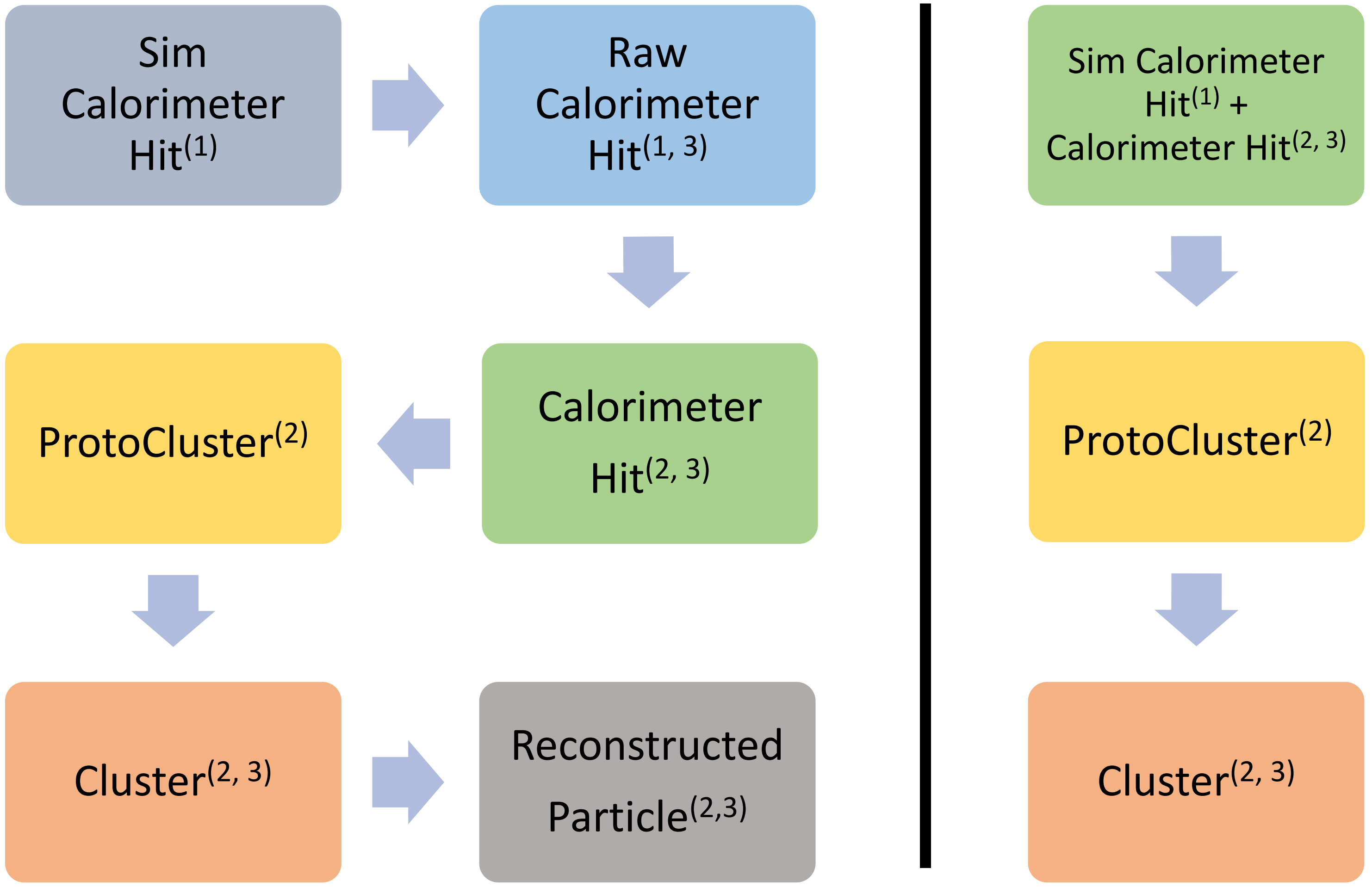
1) **For each** protocluster, **do**
   i. Identify local maxima
   ii. **For each** local maxima, **do**
      a) Calculate transverse energy profile
      b) Weight each cell in protocluster by
         - Energy
         - Distance to local maximum
         - Transverse energy profile
      c) Create a new protocluster and add all cells in old protocluster with appropriate weights
2) Turn protoclusters into clusters downstream with Center-of-Gravity method

$$w_{ij} = e^{-D_{ij}/\lambda} \times E_j$$

$$E_{ij} = \frac{w_{ij}}{\sum_j w_{ij}} E_j$$

Note: equations pulled from 06.06.2023 slides by Chao (see link at bottom of slide)

Backup | Calo. Data Model Overview



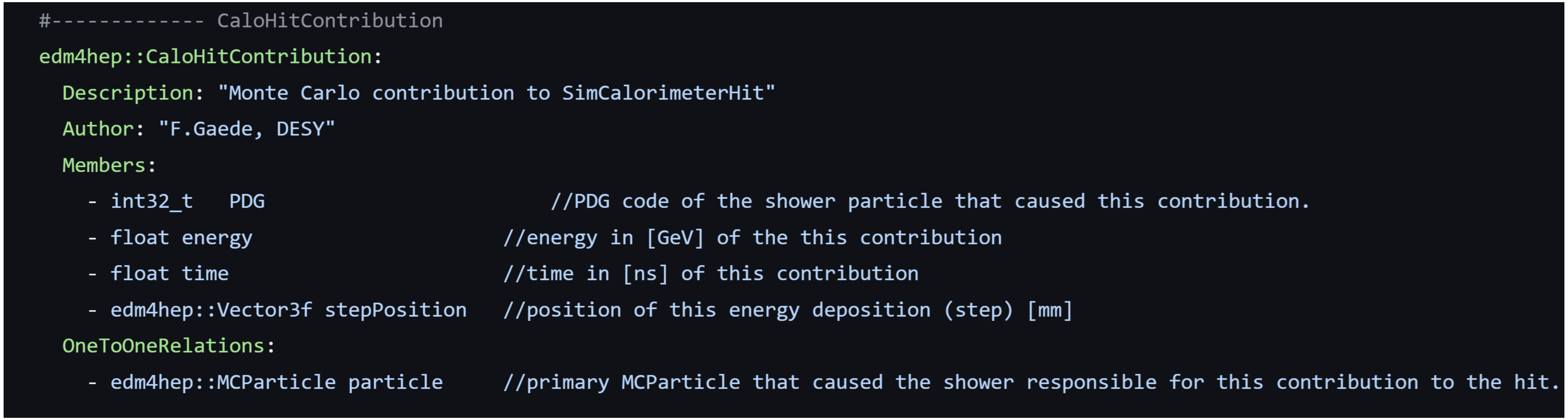
Sim-to-Reco. Cluster Circuit

Sim-to-Truth Cluster Circuit

Notes:

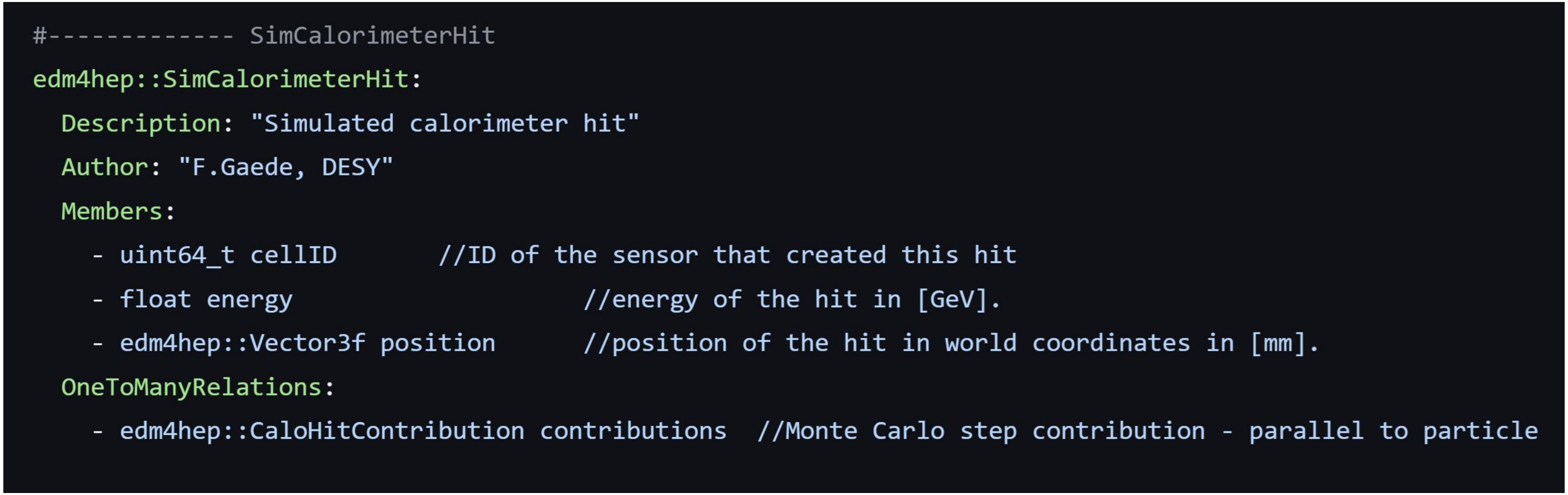
1) edm4hep::
2) edm4eic::
3) Saved to EICrecon output by default

Backup | edm4hep::SimCaloHitContritbution



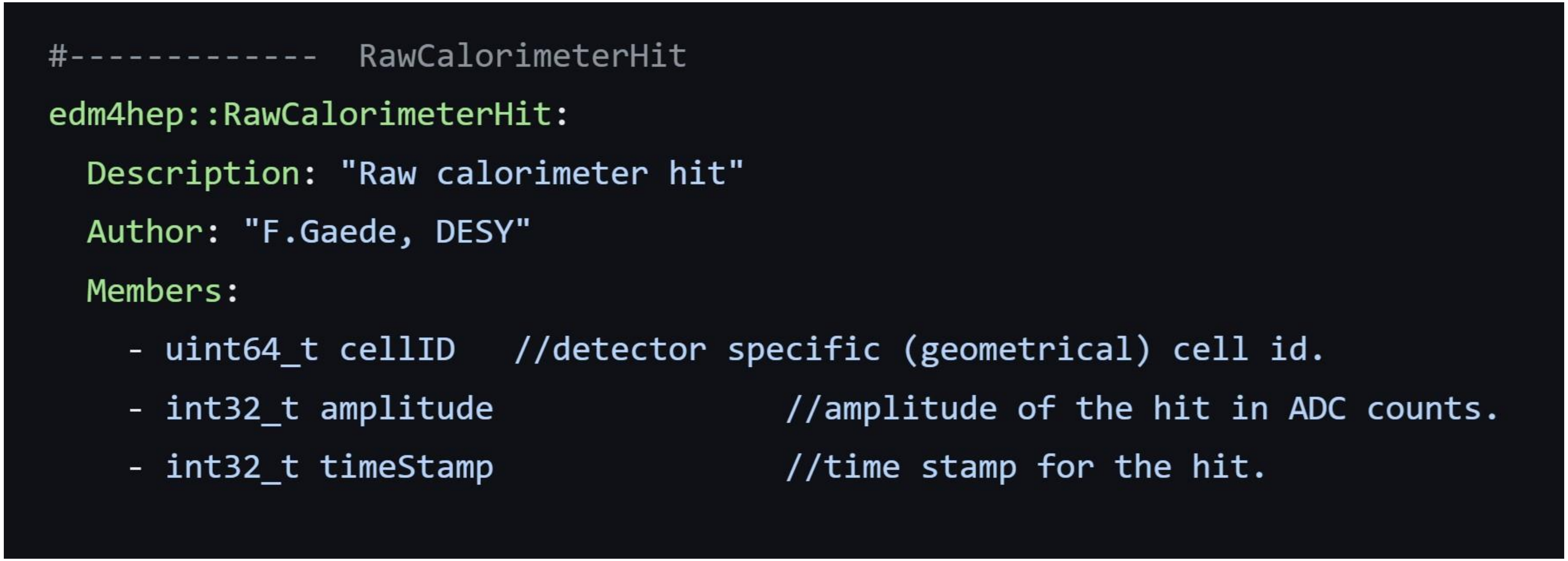
```
#------------- CaloHitContribution
edm4hep::CaloHitContribution:
  Description: "Monte Carlo contribution to SimCalorimeterHit"
  Author: "F.Gaede, DESY"
  Members:
    - int32_t   PDG                        //PDG code of the shower particle that caused this contribution.
    - float energy                     //energy in [GeV] of the this contribution
    - float time                       //time in [ns] of this contribution
    - edm4hep::Vector3f stepPosition   //position of this energy deposition (step) [mm]
  OneToOneRelations:
    - edm4hep::MCParticle particle     //primary MCParticle that caused the shower responsible for this contribution to the hit.
```

Backup | edm4hep::SimCalorimeterHit



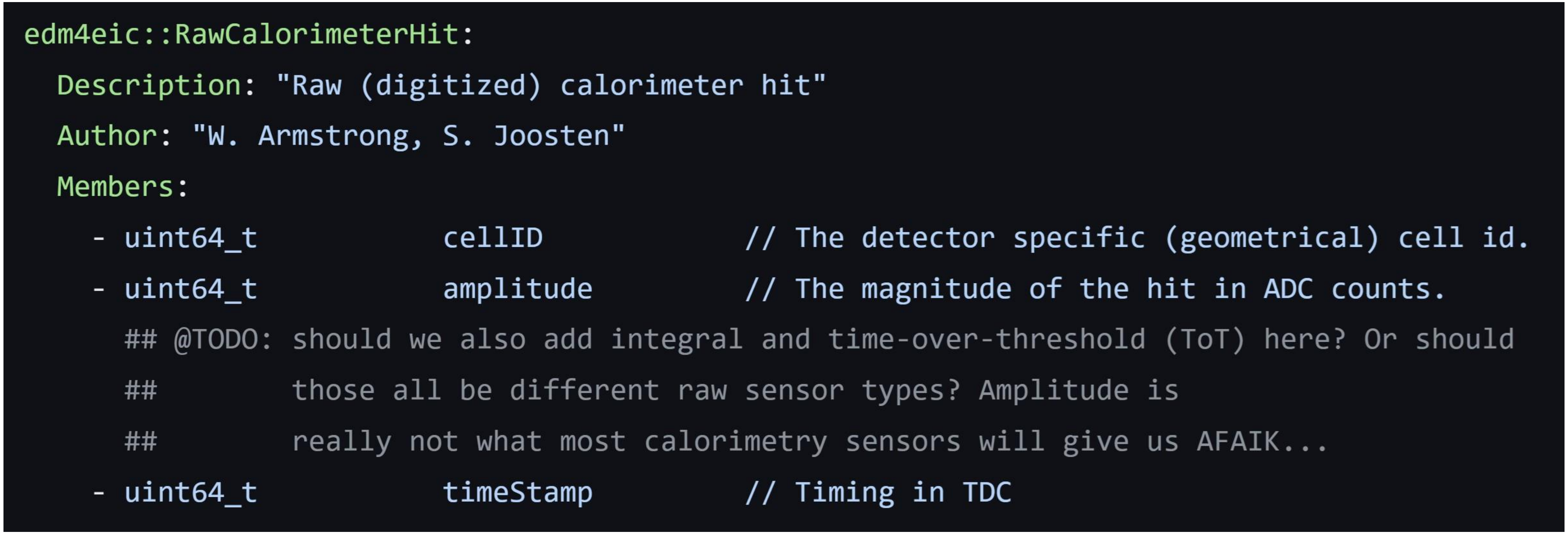
```
#------------- SimCalorimeterHit
edm4hep::SimCalorimeterHit:
  Description: "Simulated calorimeter hit"
  Author: "F.Gaede, DESY"
  Members:
    - uint64_t cellID       //ID of the sensor that created this hit
    - float energy                    //energy of the hit in [GeV].
    - edm4hep::Vector3f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

Backup | edm4hep::RawCalorimeterHit



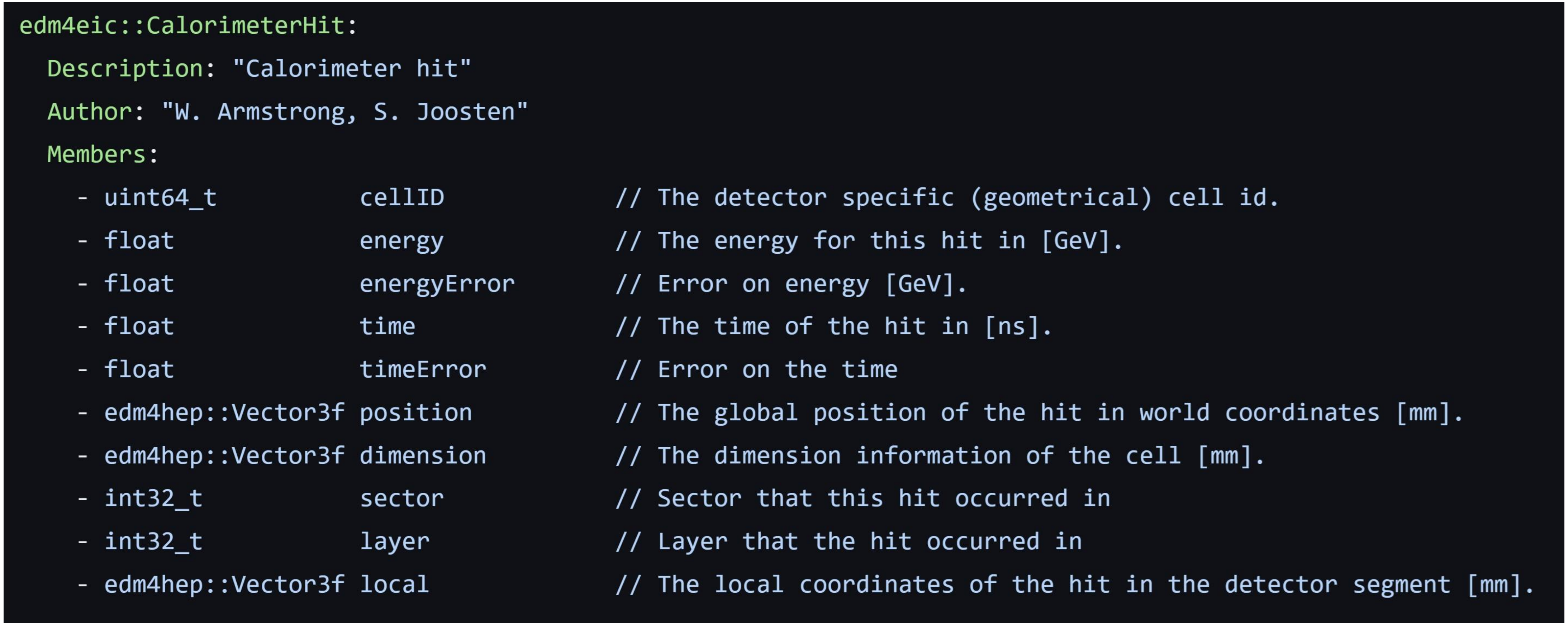
```
#-------------  RawCalorimeterHit
edm4hep::RawCalorimeterHit:
  Description: "Raw calorimeter hit"
  Author: "F.Gaede, DESY"
  Members:
    - uint64_t cellID   //detector specific (geometrical) cell id.
    - int32_t amplitude               //amplitude of the hit in ADC counts.
    - int32_t timeStamp               //time stamp for the hit.
```

Backup | edm4eic::RawCalorimeterHit



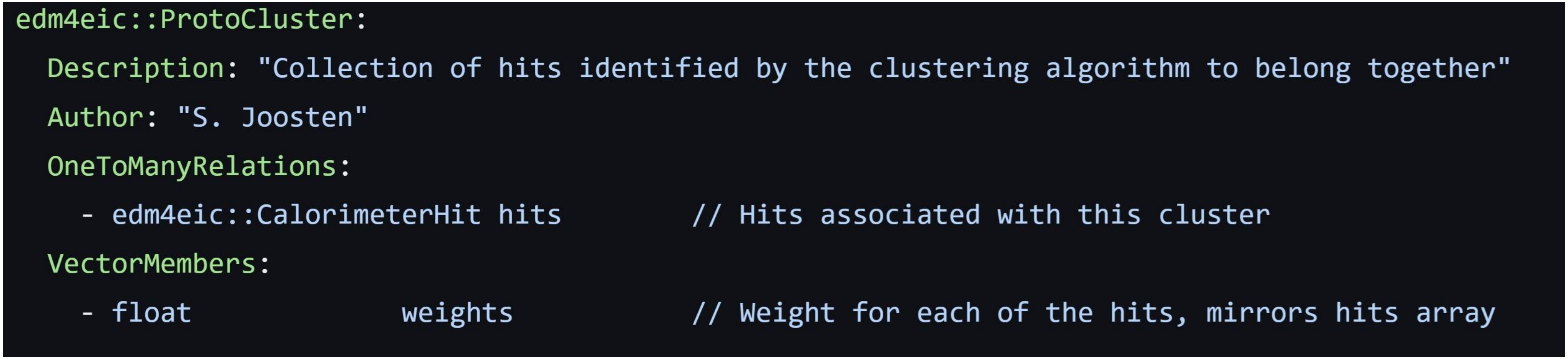
```
edm4eic::RawCalorimeterHit:
  Description: "Raw (digitized) calorimeter hit"
  Author: "W. Armstrong, S. Joosten"
  Members:
    - uint64_t           cellID            // The detector specific (geometrical) cell id.
    - uint64_t           amplitude         // The magnitude of the hit in ADC counts.
      ## @TODO: should we also add integral and time-over-threshold (ToT) here? Or should
      ##        those all be different raw sensor types? Amplitude is
      ##        really not what most calorimetry sensors will give us AFAIK...
    - uint64_t           timeStamp         // Timing in TDC
```

Backup | edm4eic::CalorimeterHit



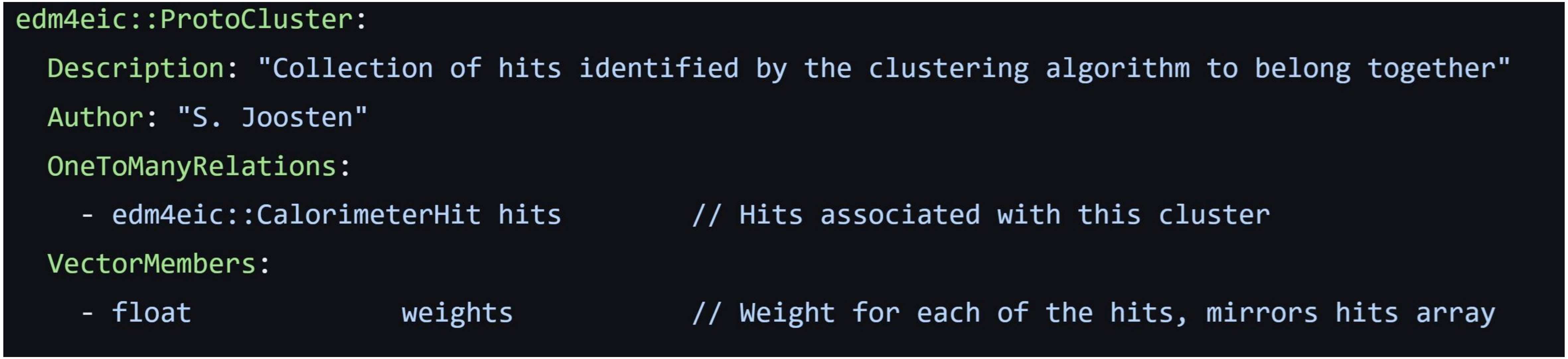
```
edm4eic::CalorimeterHit:
  Description: "Calorimeter hit"
  Author: "W. Armstrong, S. Joosten"
  Members:
    - uint64_t          cellID            // The detector specific (geometrical) cell id.
    - float             energy            // The energy for this hit in [GeV].
    - float             energyError       // Error on energy [GeV].
    - float             time              // The time of the hit in [ns].
    - float             timeError         // Error on the time
    - edm4hep::Vector3f position          // The global position of the hit in world coordinates [mm].
    - edm4hep::Vector3f dimension         // The dimension information of the cell [mm].
    - int32_t           sector            // Sector that this hit occurred in
    - int32_t           layer             // Layer that the hit occurred in
    - edm4hep::Vector3f local             // The local coordinates of the hit in the detector segment [mm].
```

Backup | edm4eic::Protocluster



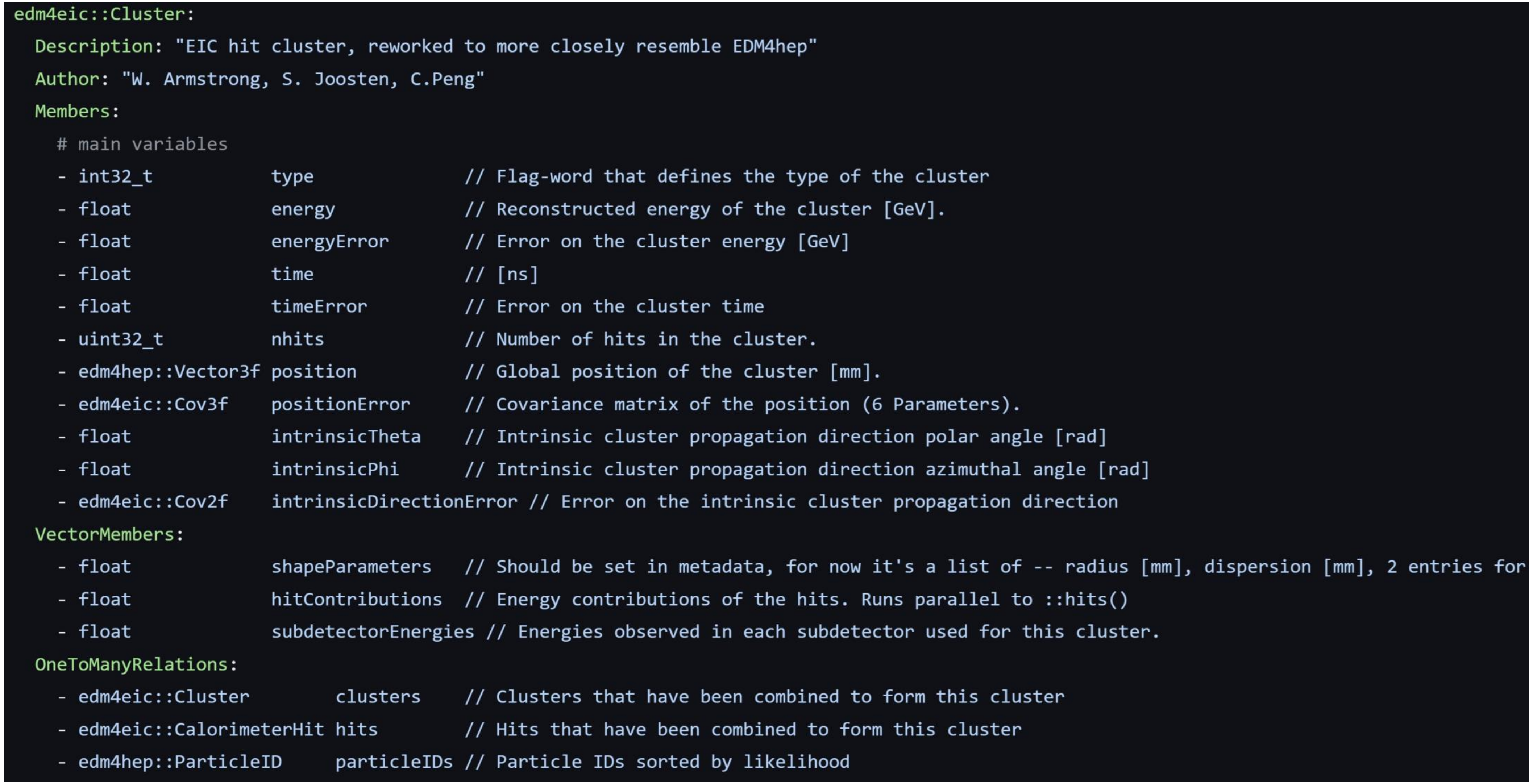
```
edm4eic::ProtoCluster:
  Description: "Collection of hits identified by the clustering algorithm to belong together"
  Author: "S. Joosten"
  OneToManyRelations:
    - edm4eic::CalorimeterHit hits        // Hits associated with this cluster
  VectorMembers:
    - float             weights           // Weight for each of the hits, mirrors hits array
```

Backup | edm4eic::Protocluster



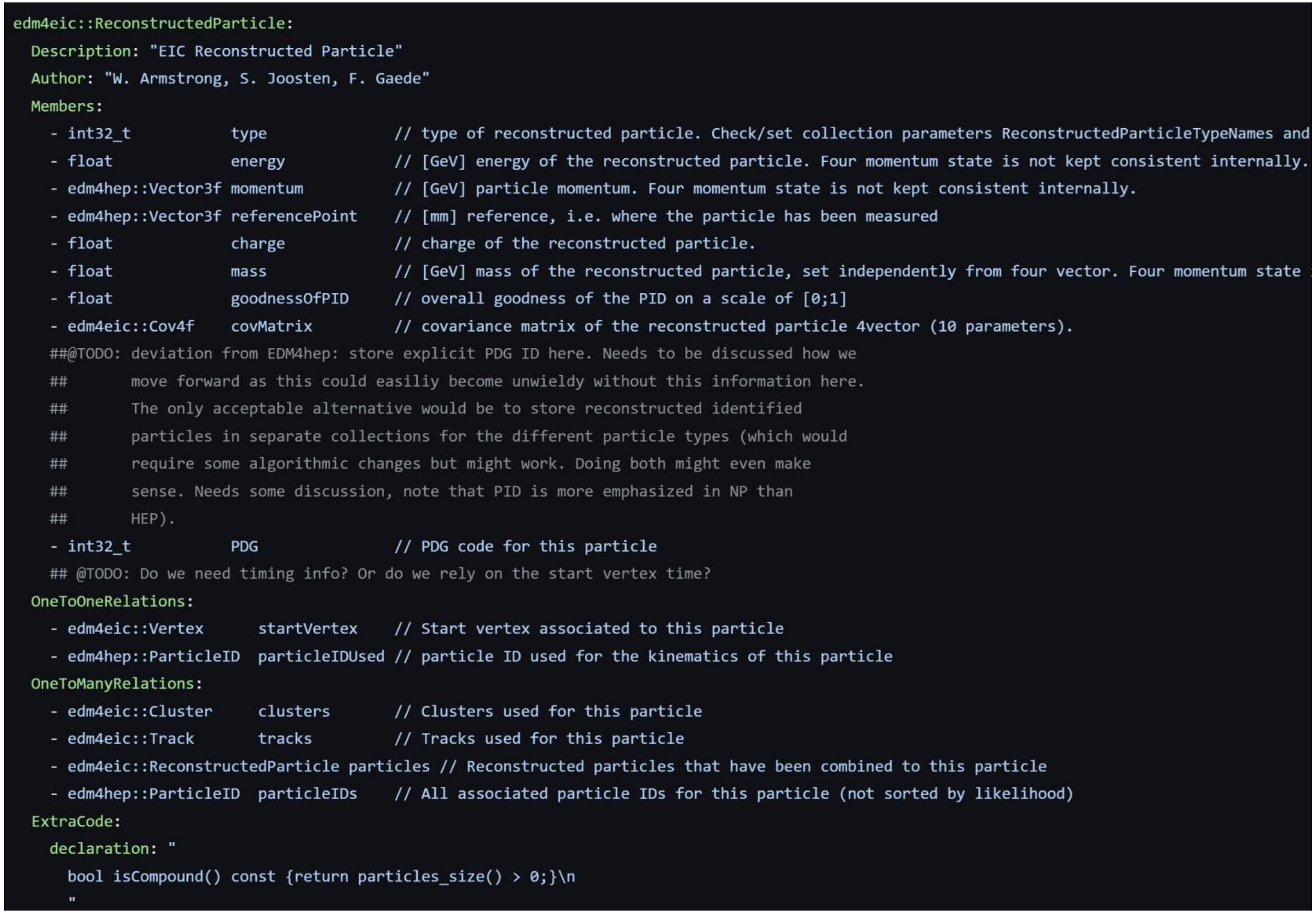
```
edm4eic::ProtoCluster:
  Description: "Collection of hits identified by the clustering algorithm to belong together"
  Author: "S. Joosten"
  OneToManyRelations:
    - edm4eic::CalorimeterHit hits        // Hits associated with this cluster
  VectorMembers:
    - float             weights           // Weight for each of the hits, mirrors hits array
```

Backup | edm4eic::Cluster



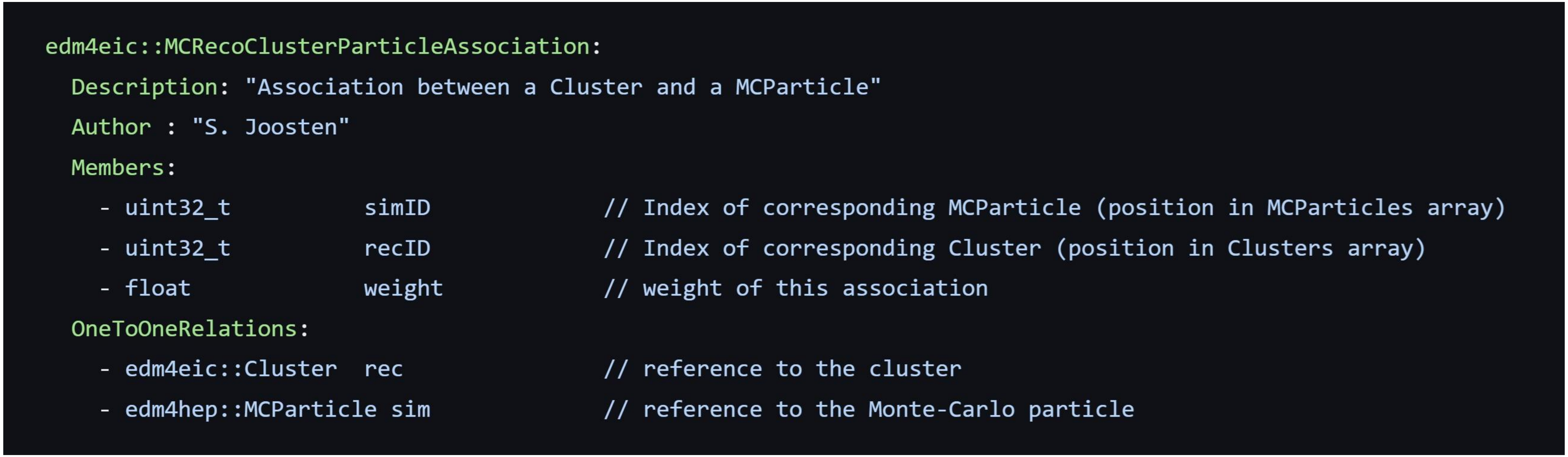
```
edm4eic::Cluster:
  Description: "EIC hit cluster, reworked to more closely resemble EDM4hep"
  Author: "W. Armstrong, S. Joosten, C.Peng"
  Members:
    # main variables
    - int32_t           type              // Flag-word that defines the type of the cluster
    - float             energy            // Reconstructed energy of the cluster [GeV].
    - float             energyError       // Error on the cluster energy [GeV]
    - float             time              // [ns]
    - float             timeError         // Error on the cluster time
    - uint32_t          nhits             // Number of hits in the cluster.
    - edm4hep::Vector3f position          // Global position of the cluster [mm].
    - edm4eic::Cov3f    positionError     // Covariance matrix of the position (6 Parameters).
    - float             intrinsicTheta    // Intrinsic cluster propagation direction polar angle [rad]
    - float             intrinsicPhi      // Intrinsic cluster propagation direction azimuthal angle [rad]
    - edm4eic::Cov2f    intrinsicDirectionError // Error on the intrinsic cluster propagation direction
  VectorMembers:
    - float             shapeParameters   // Should be set in metadata, for now it's a list of -- radius [mm], dispersion [mm], 2 entries for
    - float             hitContributions  // Energy contributions of the hits. Runs parallel to ::hits()
    - float             subdetectorEnergies // Energies observed in each subdetector used for this cluster.
  OneToManyRelations:
    - edm4eic::Cluster        clusters    // Clusters that have been combined to form this cluster
    - edm4eic::CalorimeterHit hits        // Hits that have been combined to form this cluster
    - edm4hep::ParticleID     particleIDs // Particle IDs sorted by likelihood
```

Backup | edm4eic::ReconstructedParticle



```
edm4eic::ReconstructedParticle:
  Description: "EIC Reconstructed Particle"
  Author: "W. Armstrong, S. Joosten, F. Gaede"
  Members:
    - int32_t           type             // type of reconstructed particle. Check/set collection parameters ReconstructedParticleTypeNames and
    - float             energy           // [GeV] energy of the reconstructed particle. Four momentum state is not kept consistent internally.
    - edm4hep::Vector3f momentum         // [GeV] particle momentum. Four momentum state is not kept consistent internally.
    - edm4hep::Vector3f referencePoint   // [mm] reference, i.e. where the particle has been measured
    - float             charge           // charge of the reconstructed particle.
    - float             mass             // [GeV] mass of the reconstructed particle, set independently from four vector. Four momentum state
    - float             goodnessOfPID    // overall goodness of the PID on a scale of [0;1]
    - edm4eic::Cov4f    covMatrix        // covariance matrix of the reconstructed particle 4vector (10 parameters).
    ##@TODO: deviation from EDM4hep: store explicit PDG ID here. Needs to be discussed how we
    ##       move forward as this could easiliy become unwieldy without this information here.
    ##       The only acceptable alternative would be to store reconstructed identified
    ##       particles in separate collections for the different particle types (which would
    ##       require some algorithmic changes but might work. Doing both might even make
    ##       sense. Needs some discussion, note that PID is more emphasized in NP than
    ##       HEP).
    - int32_t           PDG              // PDG code for this particle
    ## @TODO: Do we need timing info? Or do we rely on the start vertex time?
  OneToOneRelations:
    - edm4eic::Vertex      startVertex    // Start vertex associated to this particle
    - edm4hep::ParticleID  particleIDUsed // particle ID used for the kinematics of this particle
  OneToManyRelations:
    - edm4eic::Cluster     clusters       // Clusters used for this particle
    - edm4eic::Track       tracks         // Tracks used for this particle
    - edm4eic::ReconstructedParticle particles // Reconstructed particles that have been combined to this particle
    - edm4hep::ParticleID  particleIDs    // All associated particle IDs for this particle (not sorted by likelihood)
  ExtraCode:
    declaration: "
      bool isCompound() const {return particles_size() > 0;}\n
      "
```

Backup | edm4eic::MCRecoClusterParticleAssoc.



```
edm4eic::MCRecoClusterParticleAssociation:
  Description: "Association between a Cluster and a MCParticle"
  Author : "S. Joosten"
  Members:
    - uint32_t          simID            // Index of corresponding MCParticle (position in MCParticles array)
    - uint32_t          recID            // Index of corresponding Cluster (position in Clusters array)
    - float             weight           // weight of this association
  OneToOneRelations:
    - edm4eic::Cluster  rec              // reference to the cluster
    - edm4hep::MCParticle sim            // reference to the Monte-Carlo particle
```

- **Note:** associates cluster to particle associated with highest energy cell