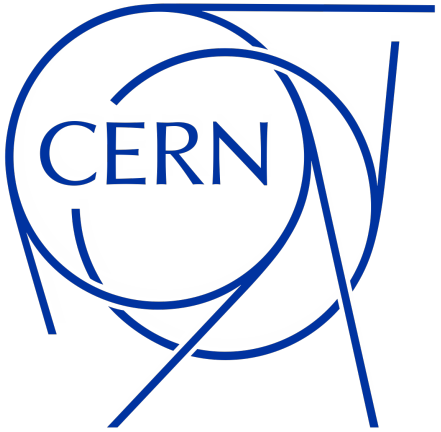


Key4hep current status and recent updates

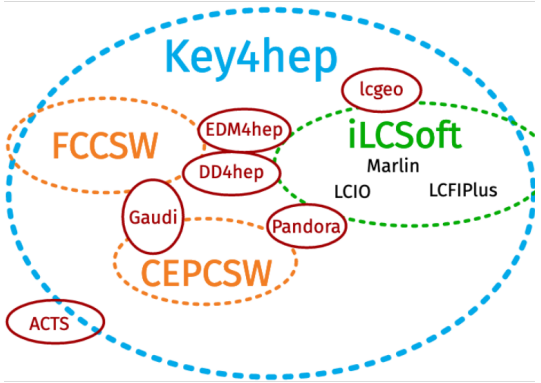
EP R&D Software Meeting

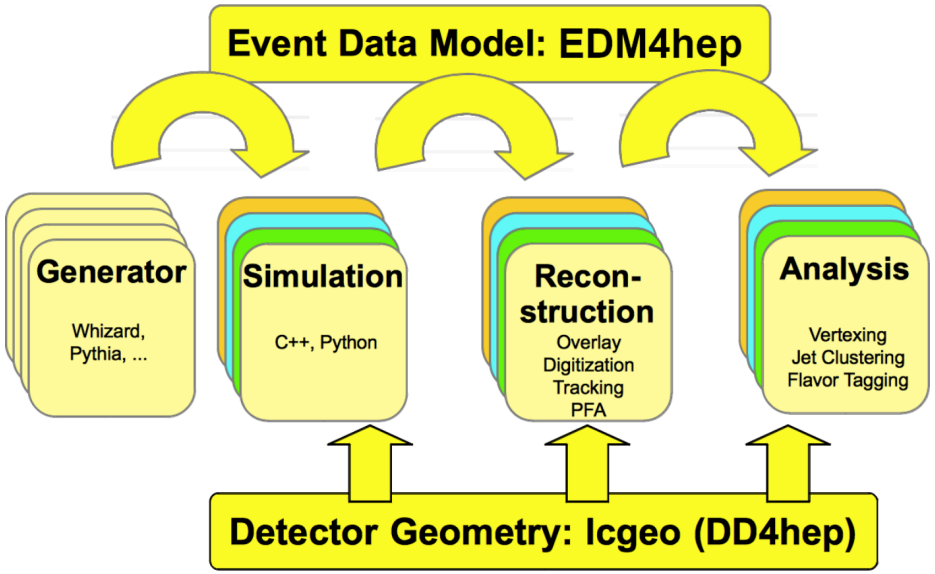
Swathi Sasikumar

19 June 2024

Key4hep

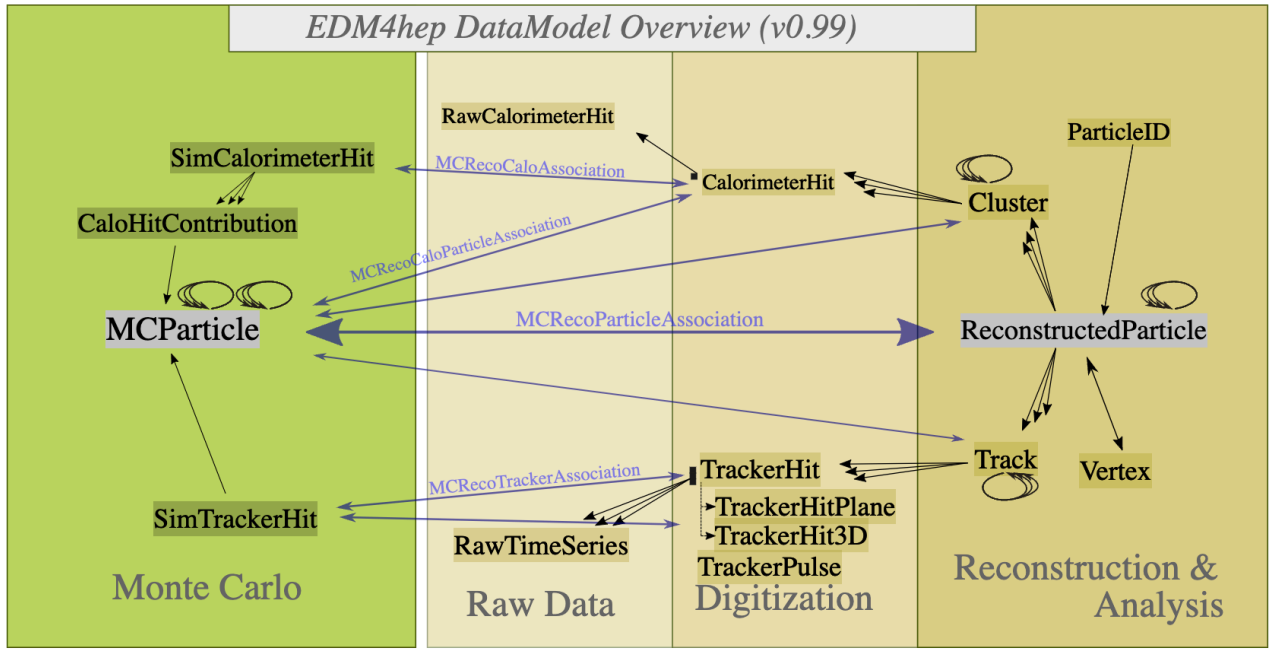
- A common turnkey software for future colliders
- Share components to reduce maintenance and development cost and allow everyone to benefit from its improvements
- Complete data processing framework from generation to data analysis
- Community with people from different future experiments: FCC, ILC, CLIC, CEPC, C³, EIC, Muon Collider etc
- Regular biweekly meetings with the stakeholders





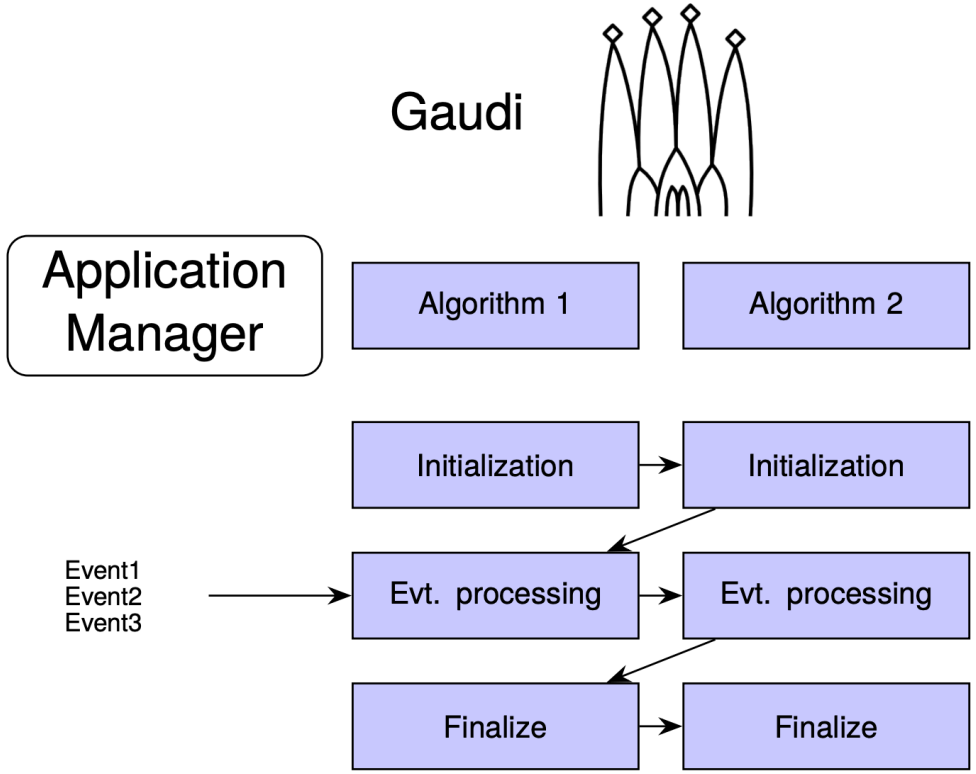
The Key4hep Event Data Model: EDM4hep

- Data model used in Key4hep, common language that all components must speak
- Goals: be generic and address the needs of the experiment
- Evolves through consensus among all stakeholders
- Generated with Podio from a text file
  - Podio (plain old data IO) is a toolkit for the creation of EDMs like EDM4hep
  - Schema evolution, along with other improvements



The Key4hep Framework

- Gaudi based core framework:
  - k4Gen for integration with generators
  - k4FWCore provides interface between EDM4hep and Gaudi
  - k4MarlinWrapper to call any Marlin (linear collider) processor
  - .....



The Key4hep Stack

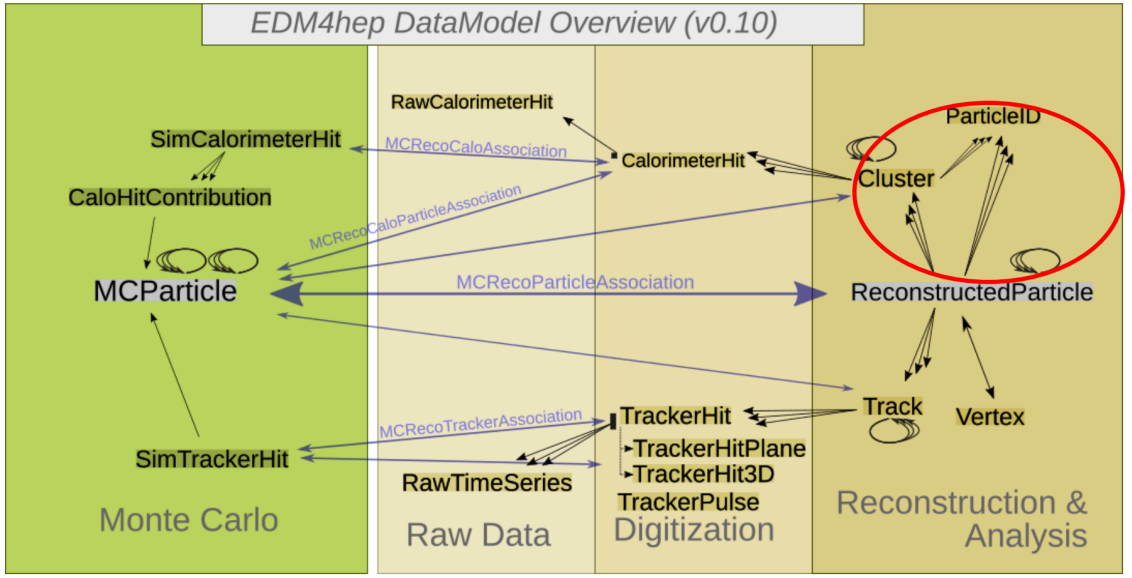
- Software provided in *stacks* deployed on cvmfs
- More than 500 packages built with Spack
- Releases in `/cvmfs/sw.hsf.org` with tagged versions of the packages
- Nightly builds in `/cvmfs/sw-nightlies.hsf.org` with the latest version of the Key4hep packages and other packages
- Easy setup with cvmfs:

```
source /cvmfs/sw.hsf.org/key4hep/releases/setup.sh              # Latest release
source /cvmfs/sw-nightlies.hsf.org/key4hep/releases/setup.sh    # Latest nightly
```

- Questions, problems, complaints and anything else related to the packages happens mostly https://github.com/key4hep/key4hep-spack

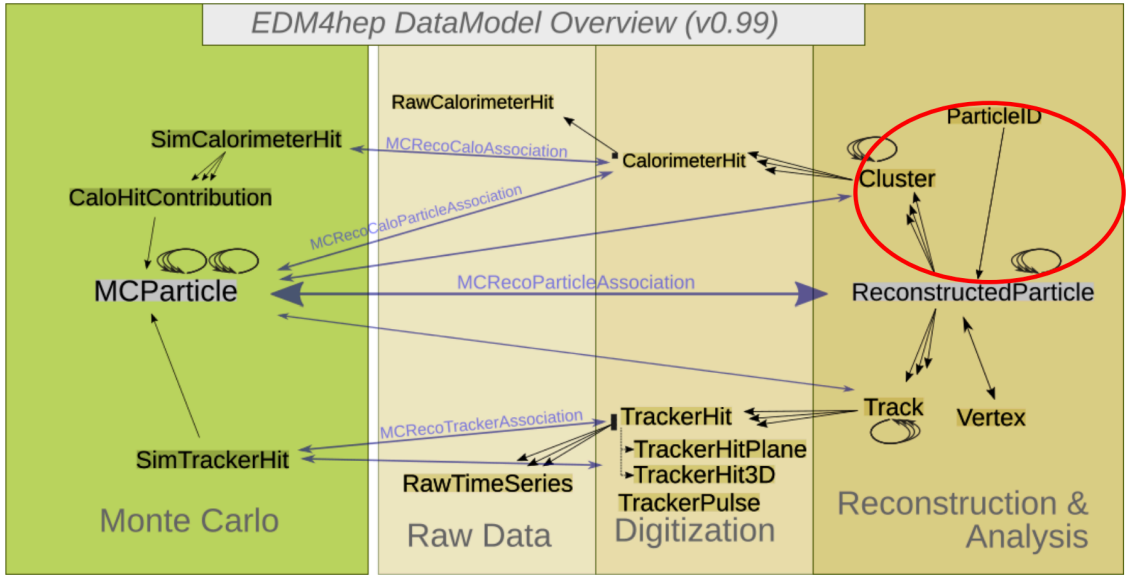
New developments in EDM4hep: Reversal of PIDs

- Earlier the reconstructed particle pointed to N particle IDs
- Motivation for change:
  - With multiple PID algorithms, the reconstructed particles are “fixed” and can’t be changed.
  - To add more PIDs the reconstructed particles need to be copied
- With the improvements, particle id points to the reconstructed particle it identifies with



New developments in EDM4hep: Reversal of PIDs

- Earlier the reconstructed particle pointed to N particle IDs
- Motivation for change:
  - With multiple PID algorithms, the reconstructed particles are “fixed” and can’t be changed.
  - To add more PIDs the reconstructed particles need to be copied
- With the improvements, particle id points to the reconstructed particle it identifies with



EDM4hep: TrackerHit Interface

- EDM4hep has different kinds of hits e.g. `TrackerHit3D` and `TrackerHitPlane`
- Tracks only pointed to the first track hit
- The changes allow the tracks to relate to an interface to any kind of the tracker hit

```
interfaces:
  edm4hep::TrackerHit:
    Description: "Tracker hit interface class"
    Author: "Thomas Madlener, DESY"
    Members:
      - uint64_t cellID
      - int32_t type
      - int32_t quality
      - float time [ns]
      - float eDep [GeV]
      - float eDepError [GeV]
      - edm4hep::Vector3d position [mm]
    Types:
      - edm4hep::TrackerHit3D
      - edm4hep::TrackerHitPlane
```

I/O Reading and Writing

- RNTuple: new format to be used instead of TTree
- Significantly less space usage than TTree and better IO throughput depending on the task
- Recently, a general Reader and Writer that work for any supported input format (including RNTuples) have been added
- Aware of the different backends and will be able to determine which one to choose for reading and can be easily selected for writing

Reading

```
auto reader = podio::makeReader("example.root");
auto frame = reader.readNextFrame(podio::Category::Events);
auto coll = frame.get("MCParticles");
```

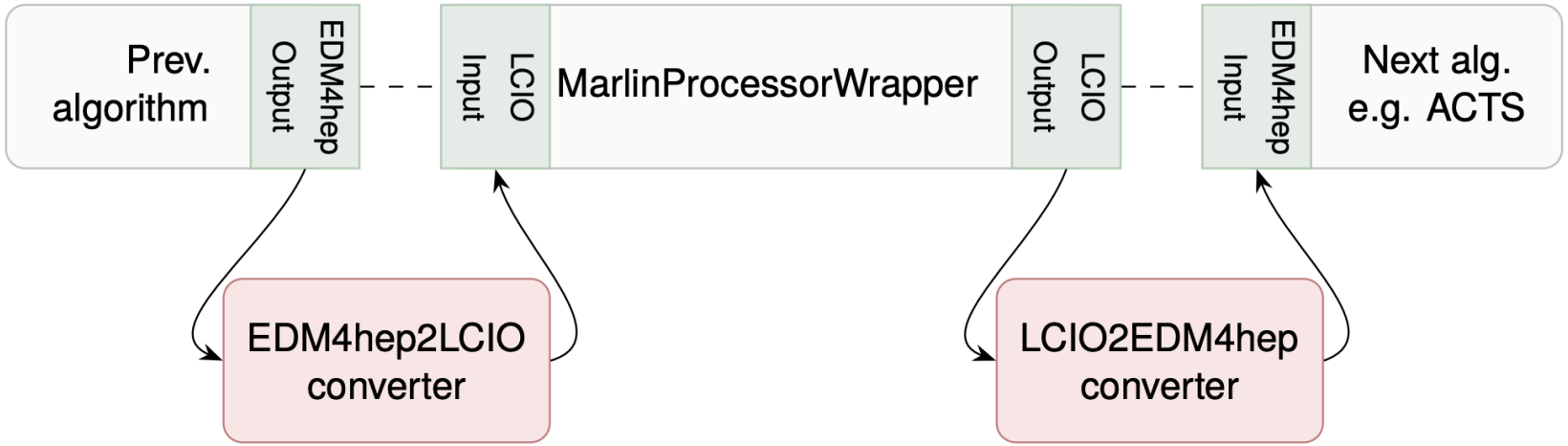
Writing

```
// Assume we have a frame called frame
auto writer = podio::makeWriter("example.root");
frameWriter.writeFrame(frame, podio::Category::Event);
```

- Ongoing work to add this functionality to Gaudi algorithms to make changing between TTrees and RNTuples easy

LCIO Converters

- Algorithms in Key4hep in Gaudi
- Such Marlin processors can be used in Gaudi using the `MarlinProcessorWrapper`
- EDM4hep input can be used seamlessly in processors taking LCIO input and giving LCIO output
- Standalone converter lcio2edm4hep to convert files



Pandora PFA and Key4hep

- Important ingredient for performance of future Higgs factory experiments: particle flow reconstruction for optimal jet energy resolutions
- Pandora particle flow algorithm (PandoraPFA) developed to study particle flow calorimetry
  - DDMarlin Pandora is the Marlin integration to iLCSoft framework to study particle flow at high granularity CALICE calorimeters
- Goals:
  - To enable use of PandoraPFA across multiple detector models (e.g. Liquid-Argon Calorimeter), important to integrate it into Key4hep
  - Replace the DDMarlinPandora and K4MarlinWrapper combination with DDGaudiPandora
- Study of PndoraPFA conducted on Nobel Liquid Argon Calorimeter of FCC

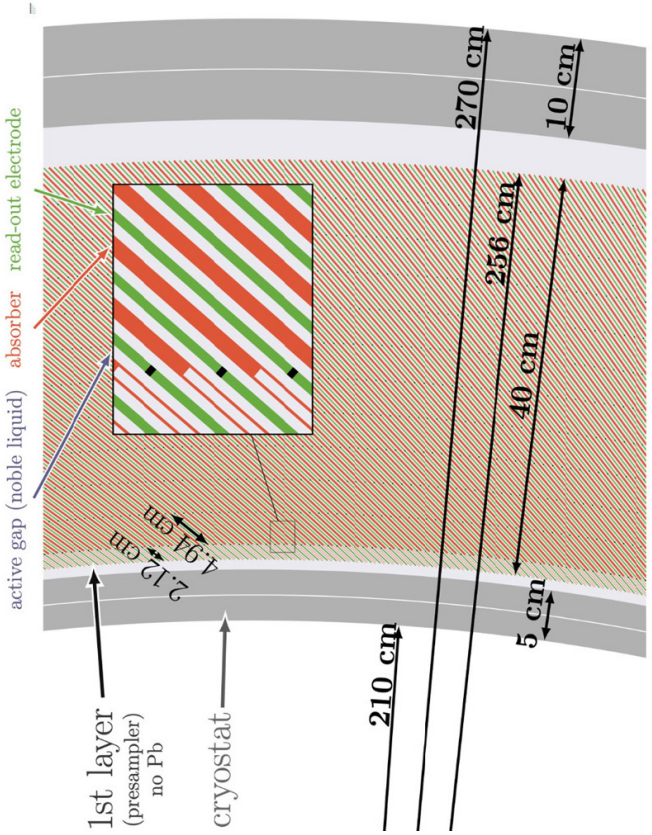
Pandora PFA and Layered Calorimeter Data

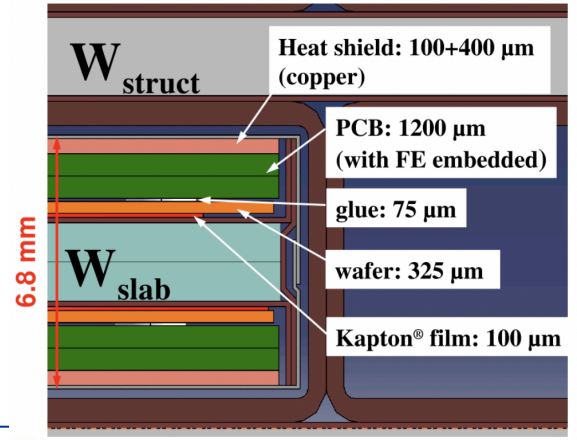
- PandoraPFA uses material properties e.g. radiation lengths and interaction lengths to determine the depth of the particle shower in the detector
- Particle flow clustering with Pandora uses the extensions attached to the detector geometries to provide the properties of the calorimeter
- The **`DD4hep::rec::LayeredCalorimeterData`** provides details like radiation length, interaction length and dimensions to the reconstruction algorithms

```
dd4hep::rec::LayeredCalorimeterData::Layer caloLayer;
caloLayer.distance = rad_first;
caloLayer.inner_nRadiationLengths   = value_of_x0/2.0;
caloLayer.inner_nInteractionLengths = value_of_lambda/2.0;
caloLayer.inner_thickness           = difference_bet_r1r2/2.0;
```

Geometry information for PandoraPFA

- DDMarlinPandora designed with high granularity CALICE sandwich calorimeters
- LAr calorimeter has a very different structure : an ensemble of different materials in a cell varying in density and homogeneity
- Density of material also varies from the inner radius to the outer radius of the barrel
- Moreover, the inclination of the segments play a role
- Challenging to calculate radiation length or interaction length for LAr





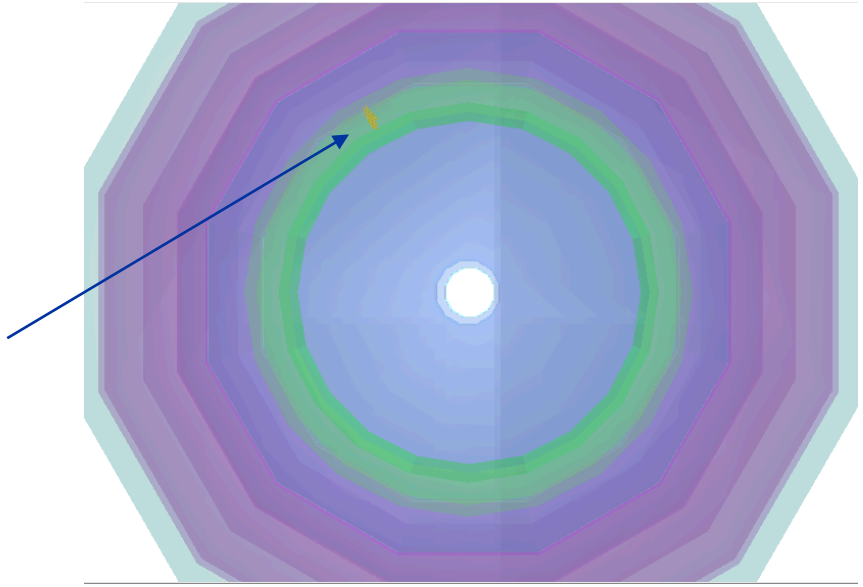
Material Manager

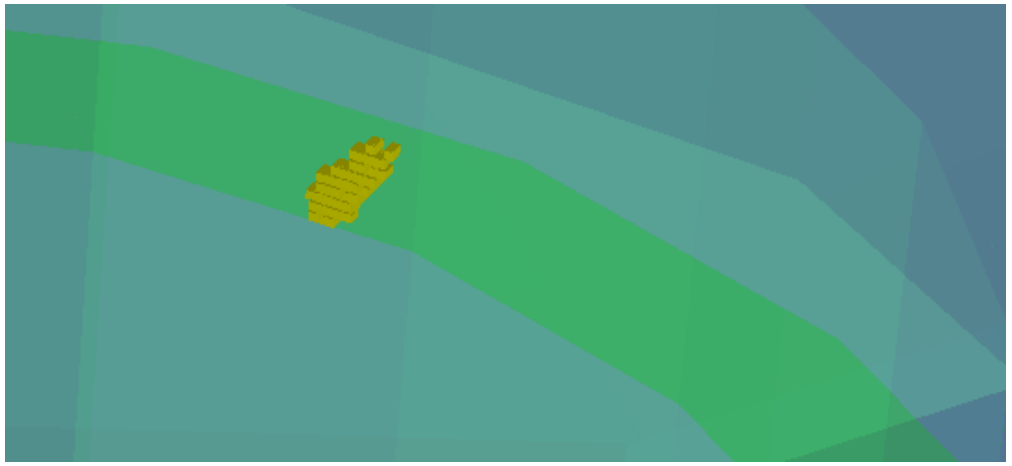
- Such information for the LAr calorimeter is obtained in a more dynamic way
- **`MaterialManager`** is a tool from DD4hep that helps extracting the necessary information between arbitrary space points
- **`MaterialManager`** returns the list of materials and their thickness along the vector
- By averaging the material between the arbitrary points material properties of the averaged material was extracted
- Crosscheck: The sum of the radiation lengths across the layers sums up to 22 X_0 as expected for the calorimeter

```
const dd4hep::rec::MaterialVec& materials = matMgr.materialsBetween(ivr1, ivr2);
auto mat = matMgr.createAveragedMaterial( materials) ;
nRadiationLengths = mat.radiationLength();
nInteractionLengths = mat.interactionLength();
double difference_bet_r1r2 = (ivr1-ivr2).r();
double value_of_x0 = layerHeight[il] / nRadiationLengths;
double value_of_lambda = layerHeight[il] / nInteractionLengths;
```

Pandora on other detector models

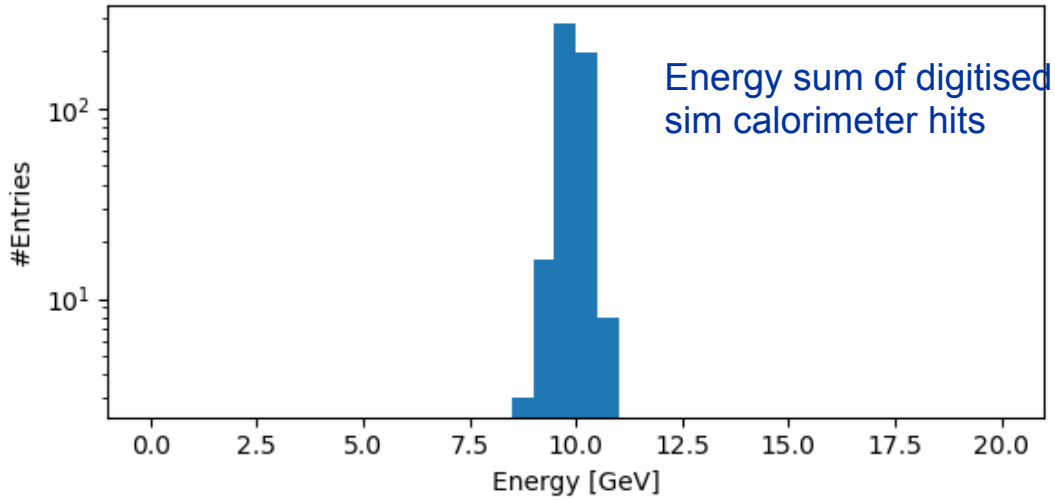
- 500 events of photons using a particle gun was simulated at an energy of 10 GeV for the CLD_LAr detector model
- By running reconstruction with all the digitized hit collections provided, PandoraPFOs could be observed for the LAr Calorimeter

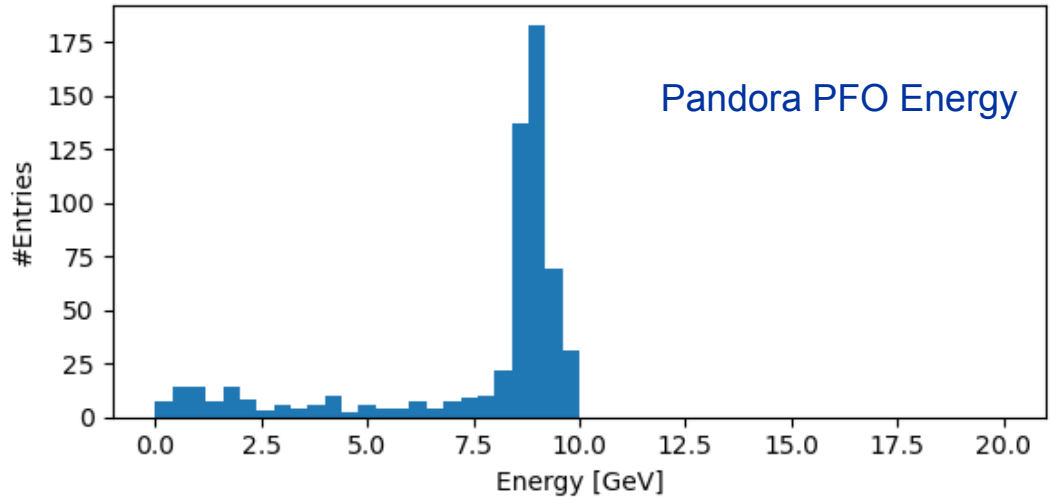




Energy of Pandora PFO

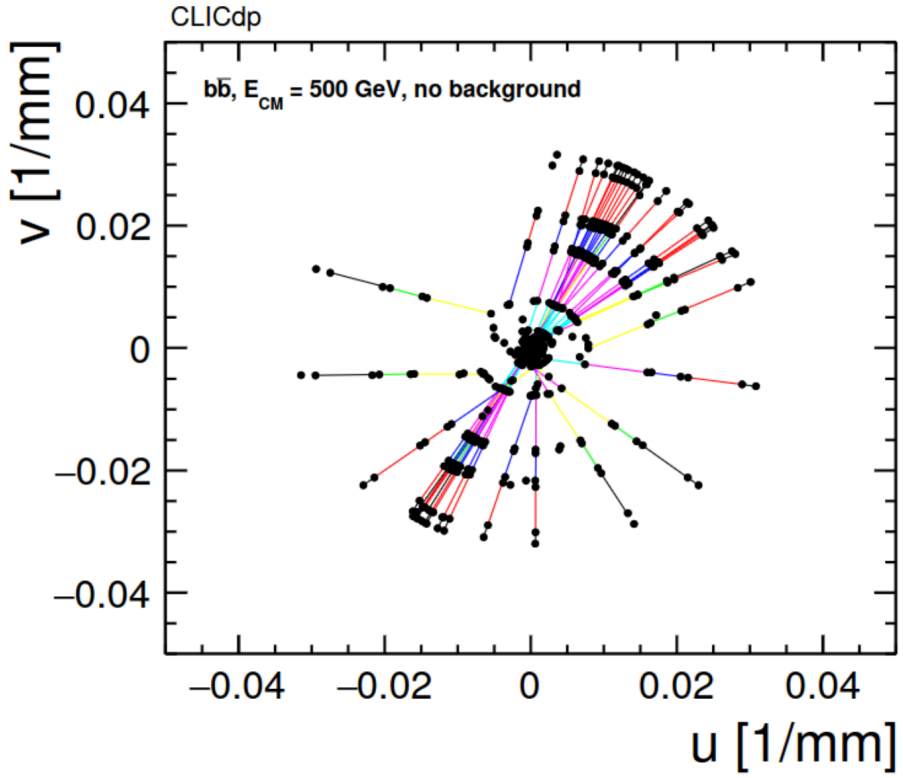
- The sum of the energies of the digitised sim calorimeter hits peaks nicely at 10 GeV as expected
- The energy of the pandora PFO obtained seen in the second figure mostly peaked at 9 GeV and has a tail
- The correction factor for photon energies needs to be adapted to the LAr calorimeter from CLD
- With the corrections even better results expected: work in progress





ACTS in Key4hep

- L. Reichenbach is working on integrating ACTS (A common tracking software) into the Key4hep framework
- Goal: Create all ‘true’ tracks
- Take all the reconstructed hits of a `MCParticle` and fit them with the ACTS Kalman filter
- Checks on all the relevant parts made
- Status: first track fits achieved!
- Missing: conversion from ACTS to EDM4hep tracks
- ACTS integration into Key4hep is progressing: https://github.com/Zehvogel/k4ActsTracking



Integration to Gaudi Algorithms

- Support for Gaudi::Functional: Functional algorithms do not have an internal state and are suitable to run multithreaded
- New service added, IOSvc that allows one to easily switch to multithreading
- To make use of these features, existing algorithms have to move to Gaudi::Functional and use IOSvc
- While porting the algorithms, should be possible to give arbitrary lists of collection to these algorithms
- Previously tried to implement using `std::map`

```
using retType = std::tuple<
    std::map<std::string, edm4hep::CalorimeterHitCollection>,
    std::map<std::string, edm4hep::MCRecoCaloAssociationCollection>>;
```

- However, `std::map` sorts keys alphabetically making it impractical for complex algorithms with multiple inputs and outputs e.g. DDCaloDigi or Overlay

Key4hep Validation: Simulation and Reconstruction

- Validation of the algorithms, either newly developed or ported from other places is very important
- Regular check of simulation and reconstruction chain performed with the latest key4hep nightlies by J M Carceller
- Plots of the relevant quantities are made and compared to the reference samples
- Plots are deployed to WebEOS
- https://key4hep-validation.web.cern.ch/
- Work in progress, no documentation yet

Summary

- Lots of progress in Key4hep in different areas
- Integration of novel and existing methods to Key4hep framework
- Progress on integration of ACTS into Key4hep framework
- Pandora PFOs could be observed for LAr Calorimeter study
- More to come, expect more integrations and native algorithms in Key4hep framework, bug fixes and quality of life improvements

home.cern