Interfacing IDEA with Key4hep/Edm4hep

Lia Lavezzi
for the working group

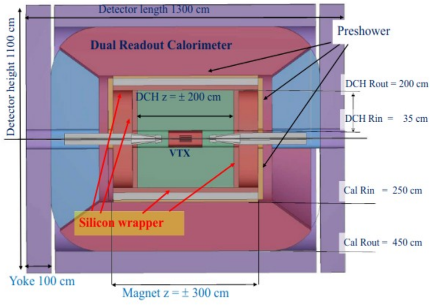
University of Torino

RD_FCC Collaboration Meeting, 15–16/12/2021

IDEA detector simulation

standalone

simulation + reconstruction



The IDEA <u>the detector</u>

the code has been implemented as a standalone software

Key4Hep

full integration

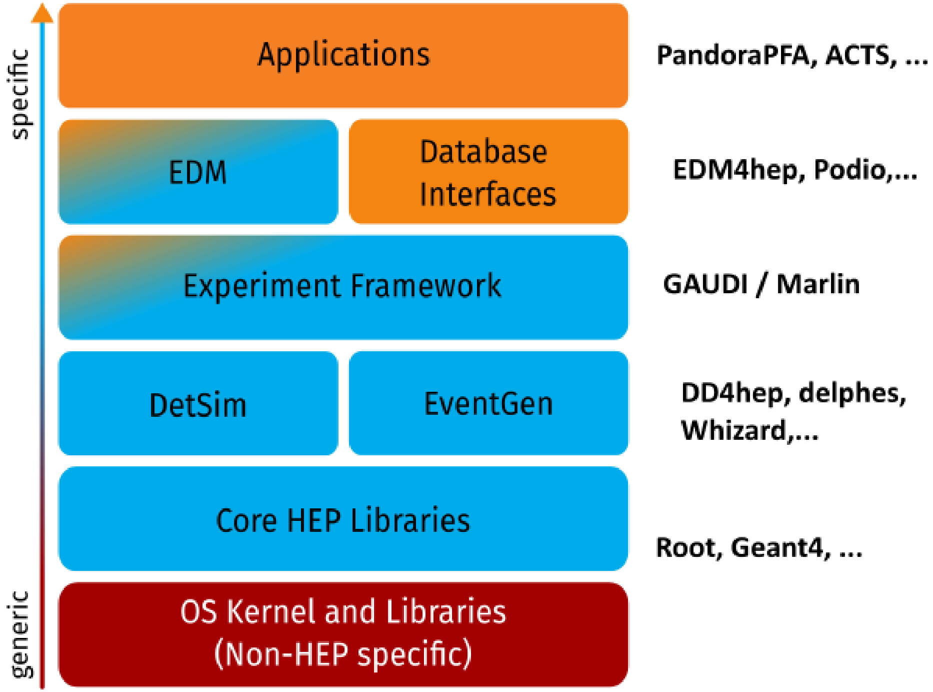
Key4hep <u>the software framework</u>

the bigger FA projects decided to share the efforts to have a platform, customizable & w/ common tools

IDEA needs to move into Key4hep

Key4hep

A turnkey software for Future Colliders from the necessity of **many projects**, ILC, CLIC, FCC, CEPC, to have a complete software stack for **physics studies**, support all the cases (hh,*ee*,*e*h) and **detector studies**, support all the detector concepts.



specific

generic

Applications — PandoraPFA, ACTS, ...

EDM — Database Interfaces — EDM4hep, Podio,...

Experiment Framework — GAUDI / Marlin

DetSim — EventGen — DD4hep, delphes, Whizard,...

Core HEP Libraries — Root, Geant4, ...

OS Kernel and Libraries (Non-HEP specific)

Requirements from users

- Easy to install
- Easy to use
- Documented

Requirements from developers

- Portable to different architectures
- GPU, FPGA, cloud
- Parallel computing, multi–threading

Requirements from experiments

- Modular / Expandable / Adaptable
- Detector agnostic reconstruction tools

Key4hep

Language – C++ / python

EDM4hep – Event Data Model

GAUDI – framework

- from LHCb, also used elsewhere, *e.g.* BESIII
- Marlin wrapper

DD4hep – Detector Description

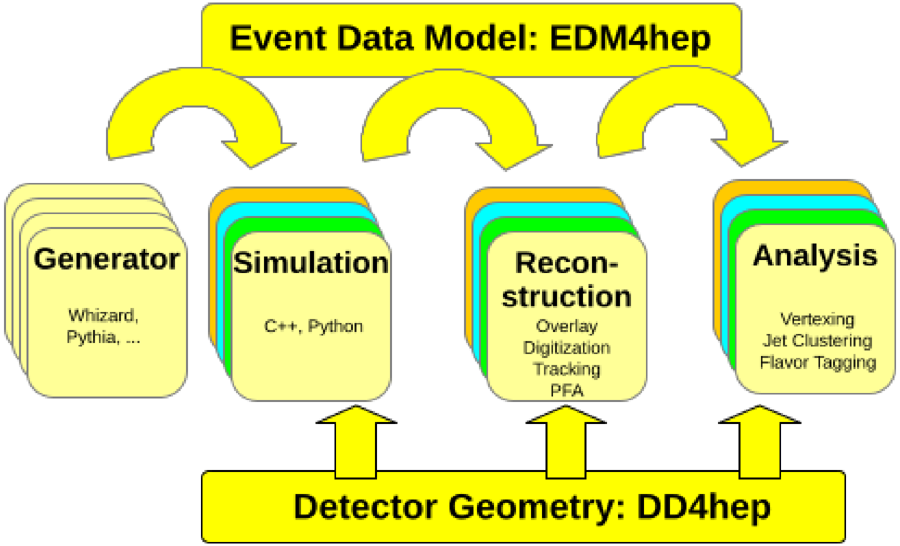
- geometry for simulation & reconstruction

Spack – package manager

- build, compilation, installation
- CernVM-FS, from LHC, w/ dedicated repository: */cvmfs/sw.hsf.org/key4hep*

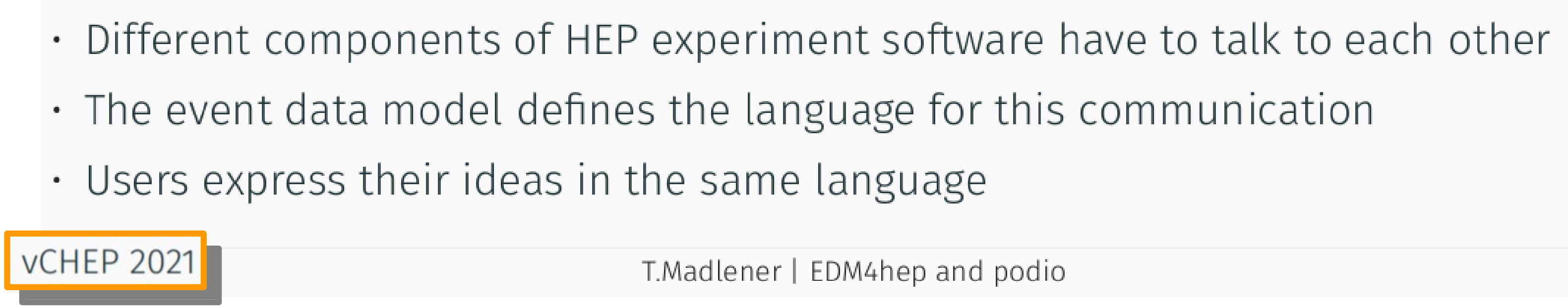
Common reconstruction tools

- ACTS (from ATLAS), PandoraPFA (from CMS), *etc.*



Key4hep GitHub Project
https://github.com/key4hep
Main documentation page
https://key4hep.github.io/key4hep-doc/

EDM4hep



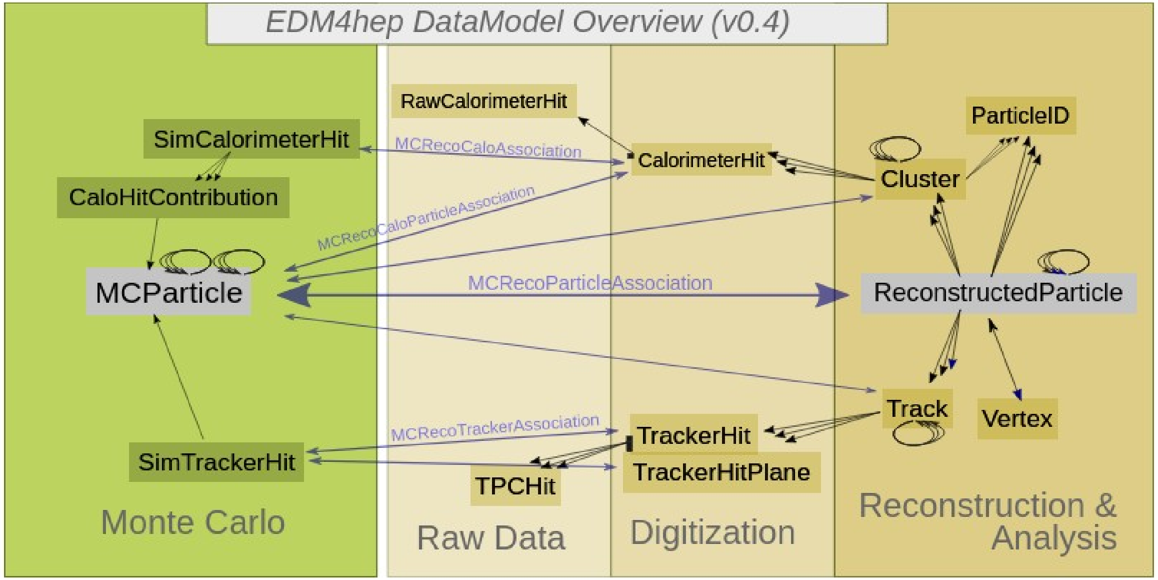
- Different components of HEP experiment software have to talk to each other
- The event data model defines the language for this communication
- Users express their ideas in the same language

vCHEP 2021

T.Madlener | EDM4hep and podio

- based on LCIO & FCC – edm
- different objects and their interactions
- both for leptonic & hadronic collisions
- support multi–threading

- the user is freed of any resource management duties or worries
- leverages the available computing power as efficiently as possible

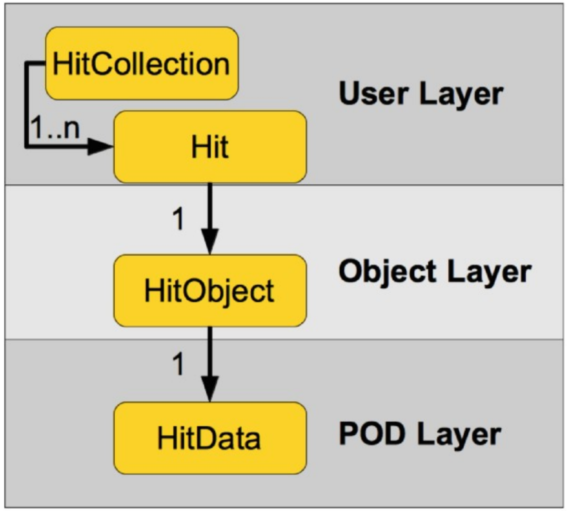


PODIO

A generic EDM toolkit

- automatic code generator – all code is automatically generated from a high level description in YAML format
  → free the users from the implementation details

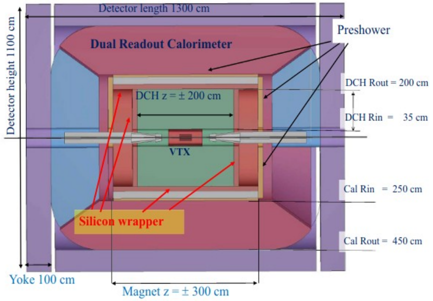
- efficient I/O & simple memory layout
- three layers:
  - (*lower*) POD – Plain Old Data. Holds arrays of the actual data structures
  - (*middle*) transient objects. Handles the relations among EDM objects and manages POD objects.
  - (*top*) comprises lightweight handles to the objects & collections.
- adaptable to backend, *e.g.* ROOT



IDEA detector simulation

standalone

simulation + reconstruction



port the geometry

port the algorithms

port the the data format

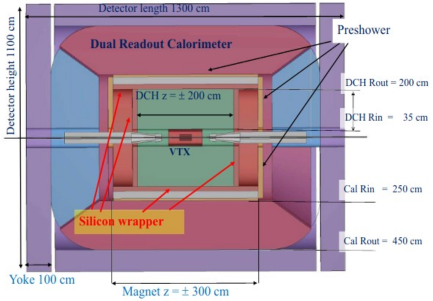
Key4Hep

full integration

IDEA detector simulation

standalone

simulation + reconstruction



port the geometry

port the algorithms

port the the data format

Key4Hep

full integration

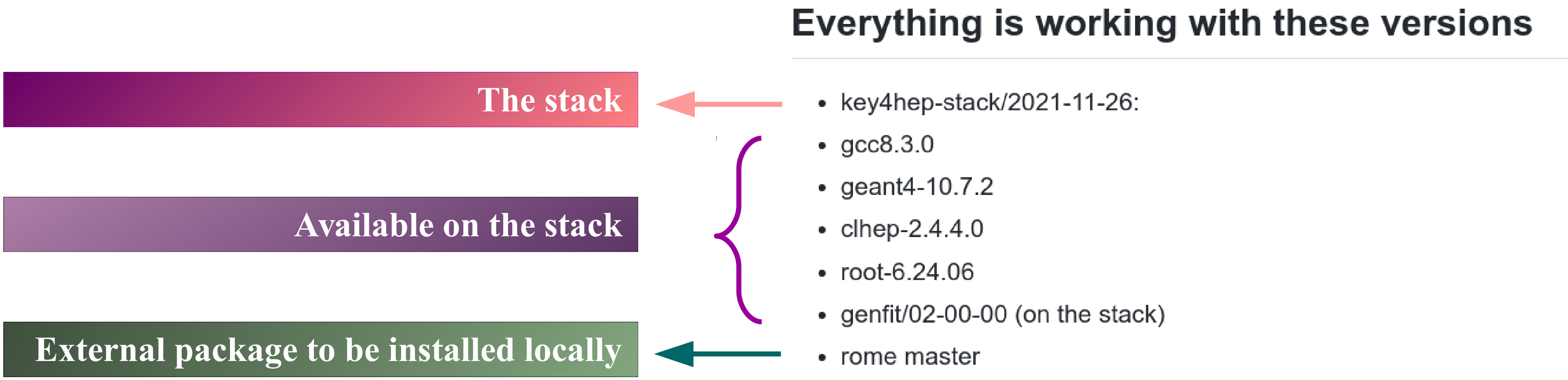
First step is to translate to EDM4Hep format

- GEANT4 Monte Carlo hits
- Standalone reconstructed tracks

Standalone code on the stack

- The standalone code was adapted for compilation on Key4hep stack (thanks to G. Tassielli)

- It works with the latest key4hep stack on CERN **lxplus** machines
  source */cvmfs/sw.hsf.org/key4hep/setup.sh*



The stack

Available on the stack

External package to be installed locally

Everything is working with these versions

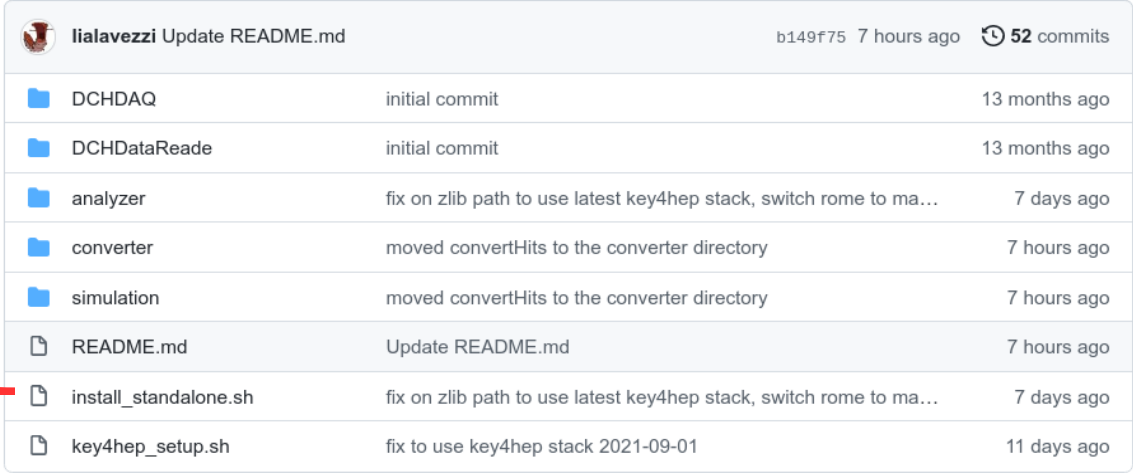
- key4hep-stack/2021-11-26:
- gcc8.3.0
- geant4-10.7.2
- clhep-2.4.4.0
- root-6.24.06
- genfit/02-00-00 (on the stack)
- rome master

- Contacted Key4hep developers to plan how to distribute the code

How to install

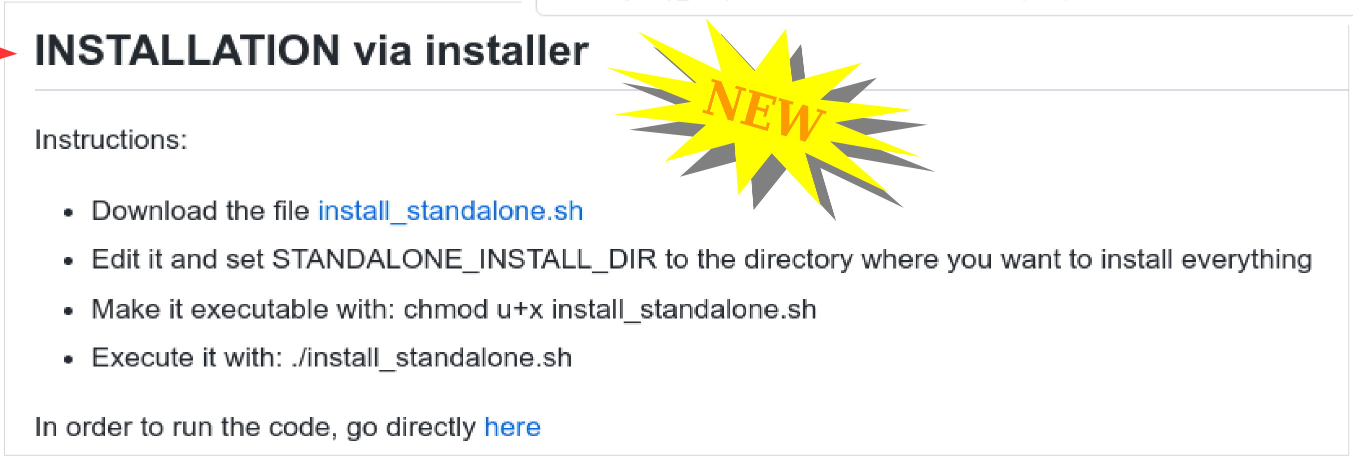
https://github.com/lialavezzi/DriftChamberPLUSVertex/tree/uptodate

standalone previous structure
+
converter/ dir
+
install_standalone.sh



lialavezzi Update README.md b149f75 7 hours ago **52** commits

DCHDAQ	initial commit	13 months ago
DCHDataReade	initial commit	13 months ago
analyzer	fix on zlib path to use latest key4hep stack, switch rome to ma…	7 days ago
converter	moved convertHits to the converter directory	7 hours ago
simulation	moved convertHits to the converter directory	7 hours ago
README.md	Update README.md	7 hours ago
install_standalone.sh	fix on zlib path to use latest key4hep stack, switch rome to ma…	7 days ago
key4hep_setup.sh	fix to use key4hep stack 2021-09-01	11 days ago



INSTALLATION via installer

NEW

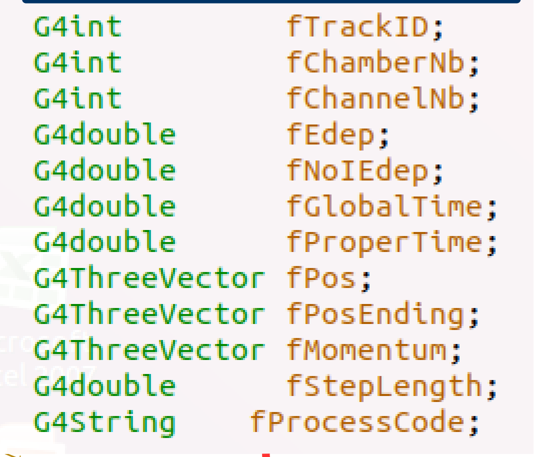
Instructions:

- Download the file install_standalone.sh
- Edit it and set STANDALONE_INSTALL_DIR to the directory where you want to install everything
- Make it executable with: chmod u+x install_standalone.sh
- Execute it with: ./install_standalone.sh

In order to run the code, go directly here

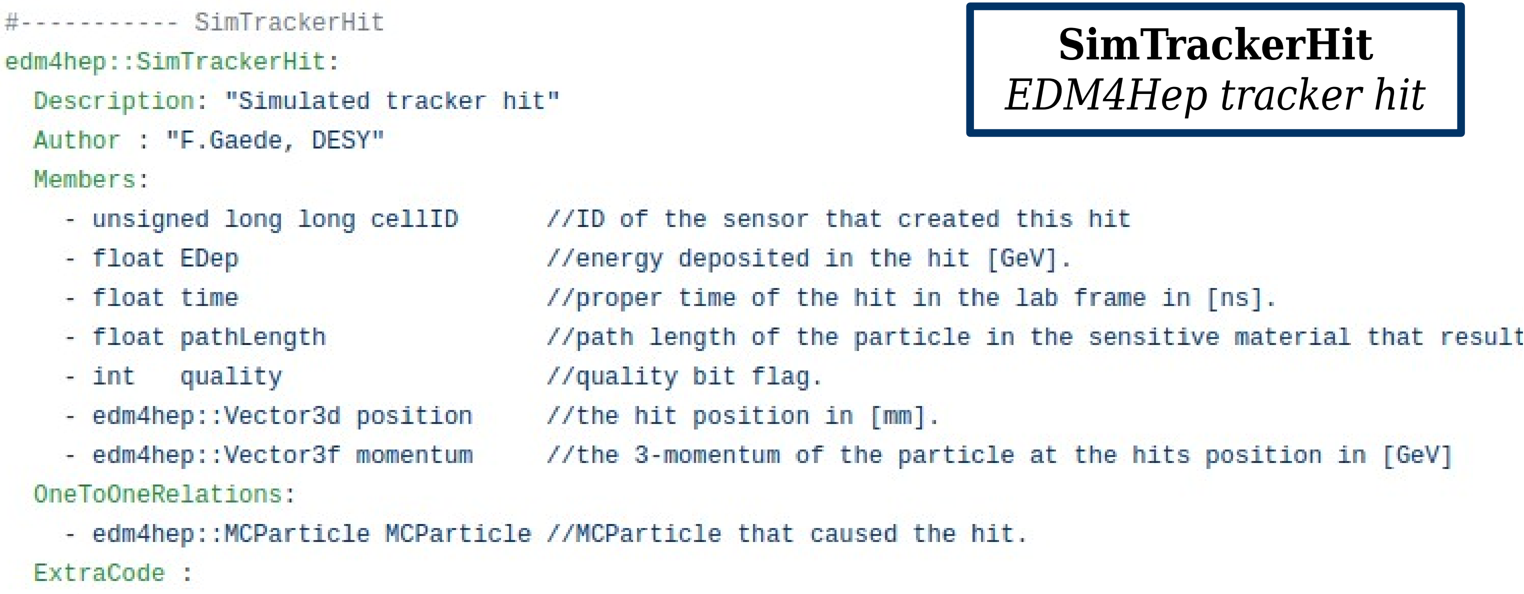
MC hit conversion

GMCG4TrackerHit
original GEANT4 *hit*



```
G4int          fTrackID;
G4int          fChamberNb;
G4int          fChannelNb;
G4double       fEdep;
G4double       fNoIEdep;
G4double       fGlobalTime;
G4double       fProperTime;
G4ThreeVector  fPos;
G4ThreeVector  fPosEnding;
G4ThreeVector  fMomentum;
G4double       fStepLength;
G4String     fProcessCode;
```

SimTrackerHit
EDM4Hep tracker hit



```
#----------- SimTrackerHit
edm4hep::SimTrackerHit:
  Description: "Simulated tracker hit"
  Author : "F.Gaede, DESY"
  Members:
    - unsigned long long cellID      //ID of the sensor that created this hit
    - float EDep                     //energy deposited in the hit [GeV].
    - float time                     //proper time of the hit in the lab frame in [ns].
    - float pathLength               //path length of the particle in the sensitive material that result
    - int   quality                  //quality bit flag.
    - edm4hep::Vector3d position     //the hit position in [mm].
    - edm4hep::Vector3f momentum     //the 3-momentum of the particle at the hits position in [GeV]
  OneToOneRelations:
    - edm4hep::MCParticle MCParticle //MCParticle that caused the hit.
  ExtraCode :
```

- The class **convertHits** translates GEANT4 hits to EDM4hep tracker hits
- **The EDM4hep hit is at the moment the SimTrackerHit** defined in:
  https://github.com/key4hep/EDM4hep/blob/master/edm4hep.yaml
- Discussed with EDM4hep people:
  if needed more information can be accommodated, but for now it should be fine

MC hit conversion

Here I will present only the tracker hits: **silicon vertex tracker**, **drift chamber**, **pre-shower**

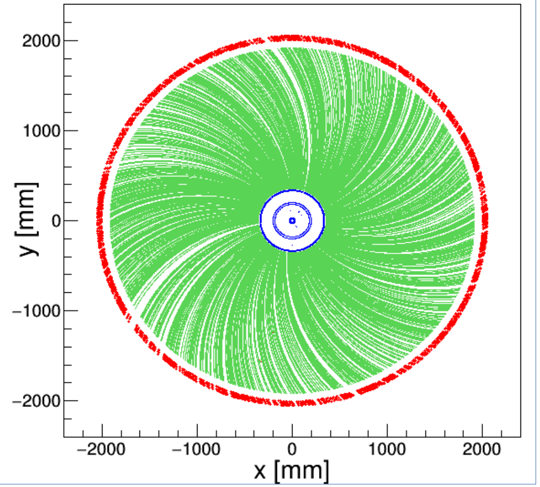
Example of simulation

particle

- 1090 events
- 1 muon/event
- theta in [88.5, 90.5] deg
- energy = 1 GeV

geometry

- Beam pipe
- SVX
- DCH
- PSHW
- magnetic field = 2.0 T



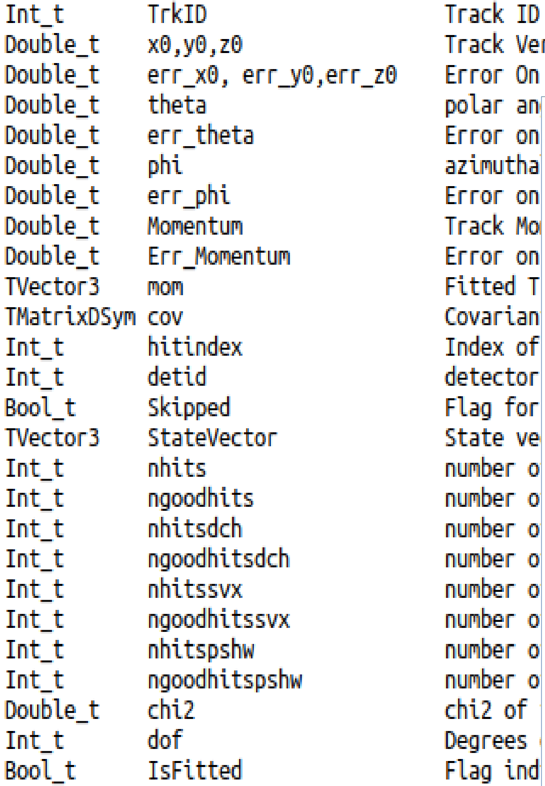
- I. Vivarelli / L. Pezzotti ported **DR calo hits** to EDM format
- W. Elmetenawee is adding them to the standalone code
  → we will need to merge all the pieces of code
- I. Garzia is adding the geometry of the **muon chambers**

Hits are not enough for inserting IDEA in the analysis.
Also reconstruction is standalone **for now** → need reconstructed tracks in EDM4hep!

Reconstructed track conversion

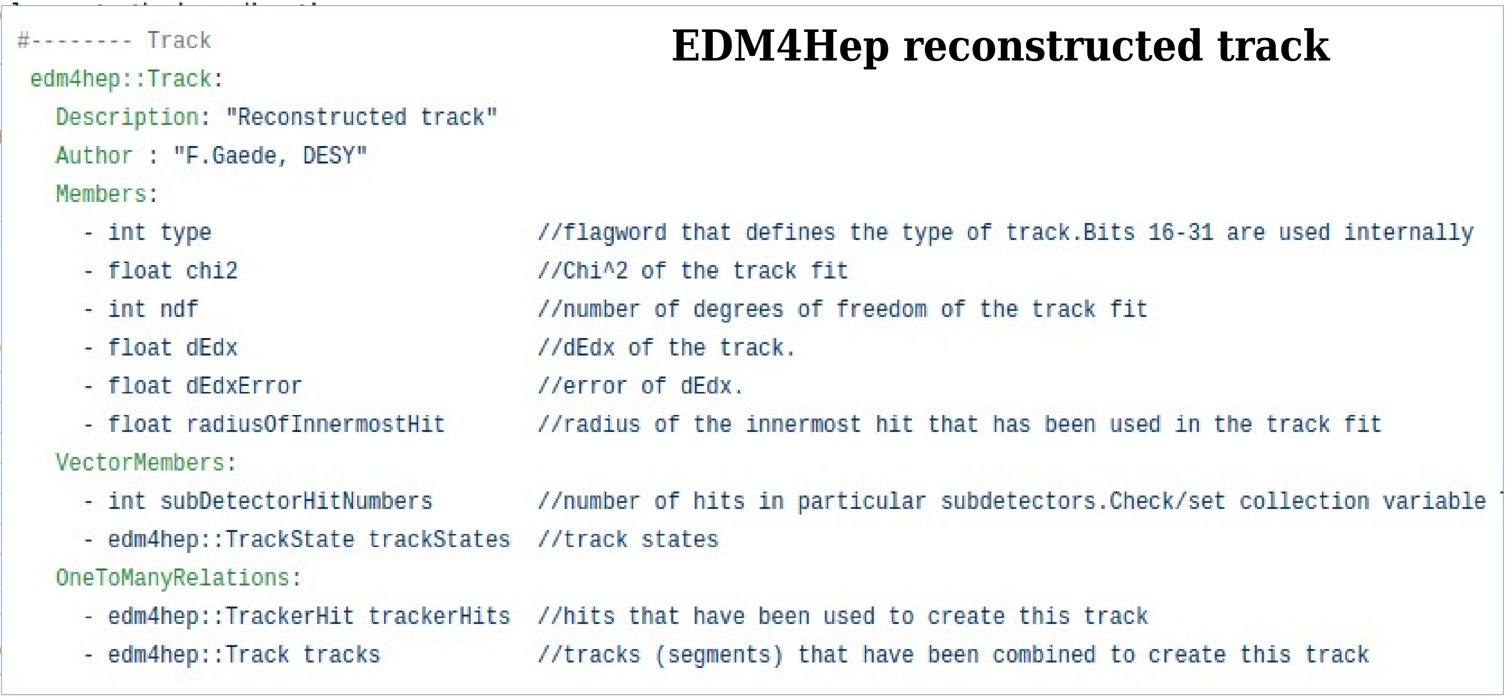
The class **convertTracks** translates standalone reconstructed tracks to EDM4Hep model

Track from standalone reconstruction



```
Int_t        TrkID                  Track ID
Double_t     x0,y0,z0               Track Vertex x,y,z coord
Double_t     err_x0, err_y0,err_z0  Error On the Track Vertex x,y,z coord
Double_t     theta                  polar an
Double_t     err_theta              Error on
Double_t     phi                    azimutha
Double_t     err_phi                Error on
Double_t     Momentum               Track Mo
Double_t     Err_Momentum           Error on
TVector3     mom                    Fitted T
TMatrixDSym  cov                    Covarian
Int_t        hitindex               Index of
Int_t        detid                  detector
Bool_t       Skipped                Flag for
TVector3     StateVector            State ve
Int_t        nhits                  number o
Int_t        ngoodhits              number o
Int_t        nhitsdch               number o
Int_t        ngoodhitsdch           number o
Int_t        nhitssvx               number o
Int_t        ngoodhitssvx           number o
Int_t        nhitspshw              number o
Int_t        ngoodhitspshw          number o
Double_t     chi2                   chi2 of
Int_t        dof                    Degrees
Bool_t       IsFitted               Flag ind
```

EDM4Hep reconstructed track



```
#-------- Track
 edm4hep::Track:
   Description: "Reconstructed track"
   Author : "F.Gaede, DESY"
   Members:
     - int type                          //flagword that defines the type of track.Bits 16-31 are used internally
     - float chi2                        //Chi^2 of the track fit
     - int ndf                           //number of degrees of freedom of the track fit
     - float dEdx                        //dEdx of the track.
     - float dEdxError                   //error of dEdx.
     - float radiusOfInnermostHit        //radius of the innermost hit that has been used in the track fit
   VectorMembers:
     - int subDetectorHitNumbers         //number of hits in particular subdetectors.Check/set collection variable
     - edm4hep::TrackState trackStates   //track states
   OneToManyRelations:
     - edm4hep::TrackerHit trackerHits   //hits that have been used to create this track
     - edm4hep::Track tracks             //tracks (segments) that have been combined to create this track
```

Will need a place to accommodate the cluster counting variable dN/dx

Reconstructed track conversion

Track from standalone reconstruction

- track parameters @ poca to the beamline
- (x, y, z, p_x, p_y, p_z) state vector @ poca
- relative covariance matrix

EDM4hep reconstructed track

- track parameters @ poca to the beamline
- $(k, \varphi, \varrho, cotg(\theta), z_0)$ state vector @ poca
- relative covariance matrix

$$J = \frac{\partial(k, \varphi, \rho, ctg(\theta), z_0)}{\partial(x, y, z, p_x, p_y, p_z)}$$

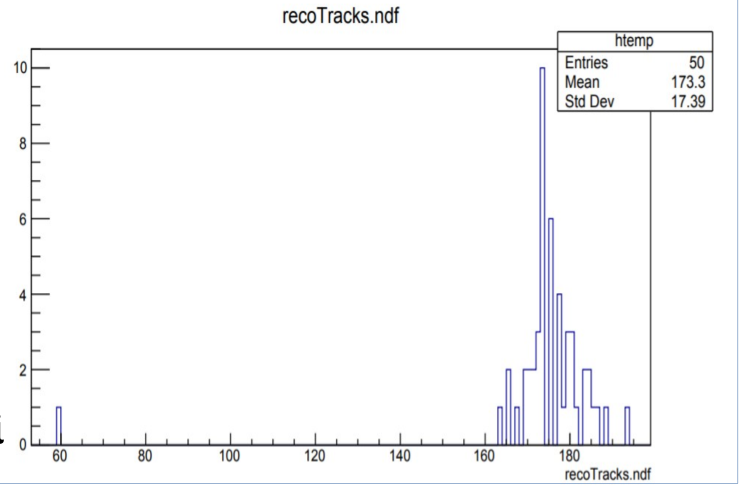
Jacobian for the conversion

Example of simulation → reconstruction

particle

- 50 events
- 1 particle/event
- energy = 1 GeV

from **P. Azzi**



Conclusions

- The code is available on *https://github.com/lialavezzi/DriftChamberPLUSVertex/tree/uptodate* and it is installable on **lxplus** machines with a **script**

- It does the following, with IDEA detector:
  - standalone simulation, with SVX, DCH, PSHW (DRCALO and MUC *in progress*)
  - standalone track reconstruction
  - hit / track conversion to EDM4hep

Usable as interface between IDEA detector and FCC analysis tools for now

→ *later the geometry and reconstruction will be implemented in the framework*

Ongoing

- Under test in order to be used soon
- Soon (I hope) will be made available to everyone
- Soon (I hope) merged with the latest developments from DRCALO & MUC0

thank you for the attention