Modelling signal digitization for trackers

R. Farinelli

on behalf of the
IDEA collaboration

Outline

1. Digitization and software framework
2. Ionization, charge collection, electronics
3. Status for the IDEA detector concept
4. Feedback and future plans

Digitization in a nutshell

Digitization: a way to do reproduce what a detector does… by software

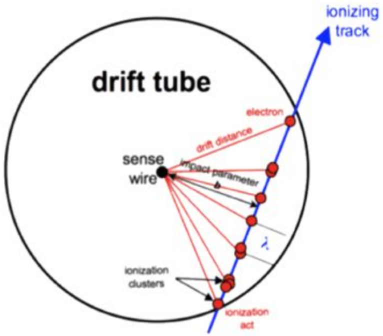
It is the last step of the **simulation** and/or the first step of the **reconstruction**

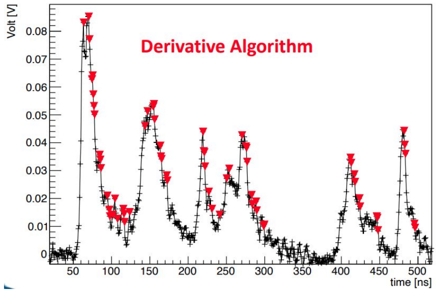
A detector interacting with particles generates, with its electronics, a digital information elaborated by a software. The same process can be reproduced by a more or less accurate simulation.

Implementation of this simulation can range from very simple and fast approaches to very detailed techniques:

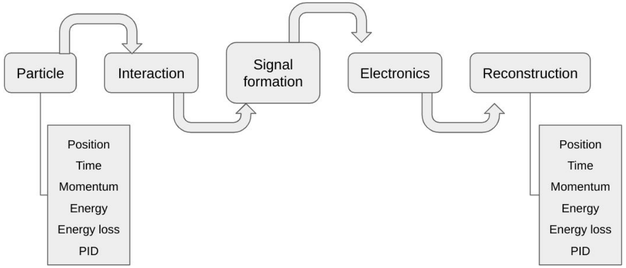
> i.e. simulation of the electron drift in a gaseous detector can evaluate it considering a constant drift velocity or a function of gas mixture, electromagnetic field, etc …
>
> i.e. simulation of a silicon tracker with a smearing of the deposited energy, time and position

Interplay between **software** development and **detector** expertise is mandatory!





Digi:	645
Det ID:	295
Energy:	452 ADC
Time:	123 ns
Nclus:	34



Digitization aims and applications

The are two main applications of a detailed simulation:

- Simulation for detector R&D
    - detector optimization -> i.e. drift field,
    - detector layout -> i.e. segmentation pitch
    - new reconstruction techniques -> i.e. cluster counting
    - electronics development -> i.e. integration time optimization

- Simulation of the detector in an experiment
    - generate a suitable output for each detector
    - performance reconstruction -> i.e. m.i.p. or soft pion
    - define the data model and the needs of each detector
    - development of the reconstruction algorithm in the software framework

Tools overview in the SW framework

Geant4:

Geant4 is a toolkit for the simulation of the passage of particles through matter.

Garfield++:

a toolkit for a detailed simulation of particle detectors that use gas and semiconductors as sensitive medium.

Neural Network:

connects a given input to the output of interest

Delphes:

provides the response of an experiment in a parametrized way and computes full covariance matrix with a smearing of the reconstructed parameters

KEY4hep:

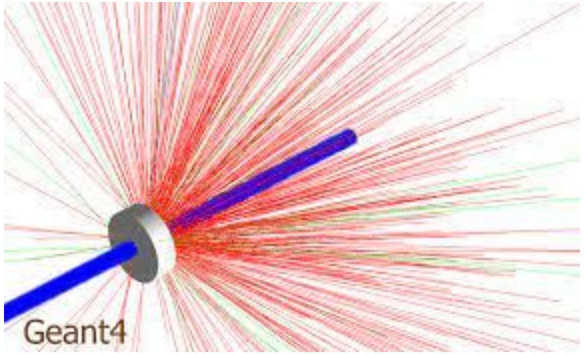
software framework with a set of common software packages for different detector concepts

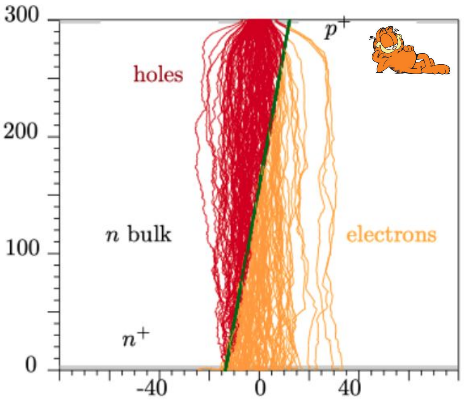
DD4hep:

provides a consistent detector description to simulation, reconstruction and analysis applications from a single source

EDM4hep:

the event data model of Key4hep





Tools overview in the SW framework

Geant4:
Geant4 is a toolkit for the simulation of the passage of particles through matter.

Garfield++:
a toolkit for a detailed simulation of particle detectors that use gas and semiconductors as sensitive medium.

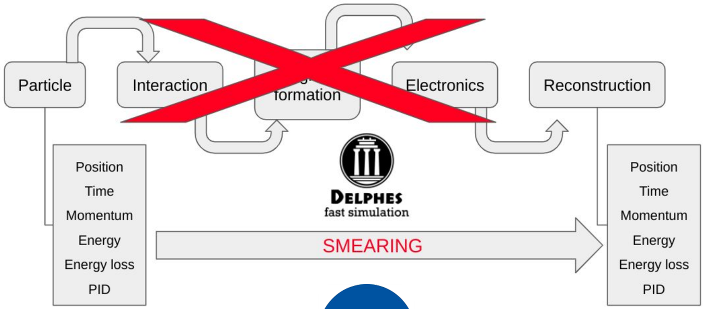
Neural Network:
connects a given input to the output of interest

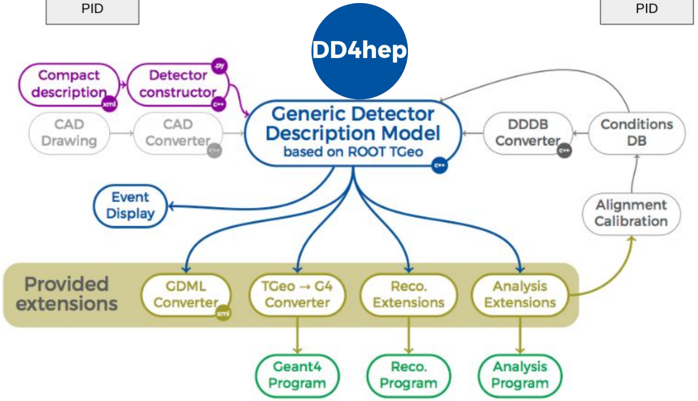
Delphes:
provides the response of an experiment in a parametrized way and computes full covariance matrix with a smearing of the reconstructed parameters

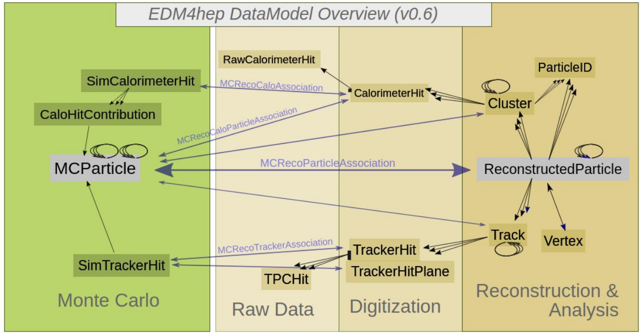
KEY4hep:
software framework with a set of common software packages for different detector concepts

DD4hep:
provides a consistent detector description to simulation, reconstruction and analysis applications from a single source

EDM4hep:
the event data model of Key4hep







Digitization in the SW framework

There are two hit types to be digitised:

- TrackerHit
- CalorimeterHit (not covered in this talk)

Input/Output are needed to:

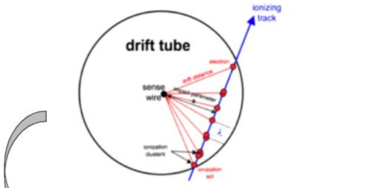
- define the data model
- develop the reconstruction algorithms
- complete the skeleton of the code

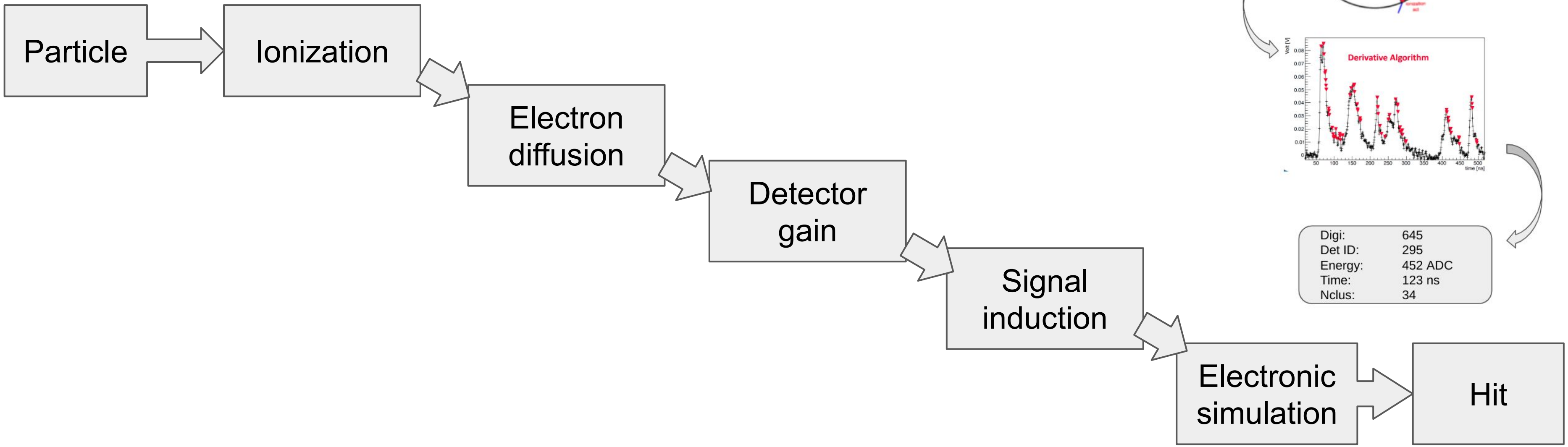
Once completed with an agile **approach** there is room to implement all the detail needed. Be aware of the simulation time and remember: optimise!

```
#----------- SimTrackerHit
edm4hep::SimTrackerHit:
  Description: "Simulated tracker hit"
  Author : "F.Gaede, DESY"
  Members:
    - uint64_t 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 resulted in this hit.
    - int32_t   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]
#------------  TrackerHit
edm4hep::TrackerHit:
  Description : "Tracker hit"
  Author : "F.Gaede, DESY"
  Members :
    - uint64_t cellID      //ID of the sensor that created this hit
    - int32_t type                      //type of raw data hit, either one of edm4hep::TPCHIT, edm4hep::SIMTRACKERHIT -
    - int32_t quality                   //quality bit flag of the hit.
    - float time                    //time of the hit.
    - float eDep                    //energy deposited on the hit [GeV].
    - float eDepError               //error measured on EDep [GeV].
    - edm4hep::Vector3d position    //hit position in [mm].
    - std::array<float,6> covMatrix  //covariance of the position (x,y,z), stored as lower triangle matrix. i.e. cov(x,x)
#----------  TPCHit
edm4hep::TPCHit:
  Description: "Time Projection Chamber Hit"
  Author : "F.Gaede, DESY"
  Members:
     - uint64_t cellID  //detector specific cell id.
     - int32_t quality               //quality flag for the hit.
     - float time                //time of the hit.
     - float charge              //integrated charge of the hit.
```

```
#------------ 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.
#------------  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.
#------------  CalorimeterHit
edm4hep::CalorimeterHit:
  Description: "Calorimeter hit"
  Author : "F.Gaede, DESY"
  Members:
    - uint64_t cellID  //detector specific (geometrical) cell id.
    - float energy              //energy of the hit in [GeV].
    - float energyError         //error of the hit energy in [GeV].
    - float time                //time of the hit in [ns].
    - edm4hep::Vector3f position //position of the hit in world coordinates.
```

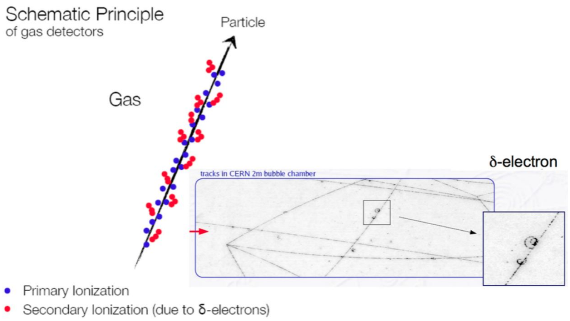
General steps in tracker digitization





Here is shown a list of processes needed to detect a particle and to generate a hit. Some of the processes are not mandatory, some other ones are not listed

Ionization



Geant4 is a pillar for HEP simulations, thanks to several years of feedback between developers and users: it reproduces the evolution of the particle's tracks and its daughters; the energy loss and its location among the path; multiple scattering and many other features.

Most of the energy loss generates primary electrons by means of ionization processes, then Geant4 is extraordinary importance to simulate the signal that a detector has to reconstruct. **This is the main driver** for the simulation in FCC software framework.

Another software is suited for the this simulation: TrackerHeed (from **Garfield++**) where an extensive implementation of many medium parameters (i.e. energy levels, excitation modes and others) have been calibrated with experimental data with unprecedented precision.

A **discrepancy** between Geant4 and Garfield++ has been observed and this could impact the detector optimization (i.e. cluster counting technique) and several solutions are under study to improve the precision of the codes and optimize the time consumption:

1. merge the two codes to improve the ionization precision; this would be time consuming in the simulation

2. the development of a code to parametrize the energy loss and the ionization (number of clusters);
this would be man-power consuming in the development (NN are under investigation to perform this task)
i.e. reproduce the poissonian process with a sampling of the relative position of the primary cluster from exponential distribution. Secondary electrons can be taken from tables.

Electron diffusion

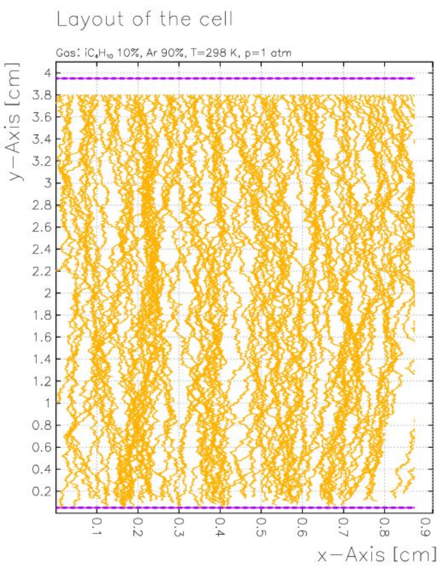
Electron diffusion simulation takes care of the primary electrons spatial (and temporal) shift during their drift to the collection point.

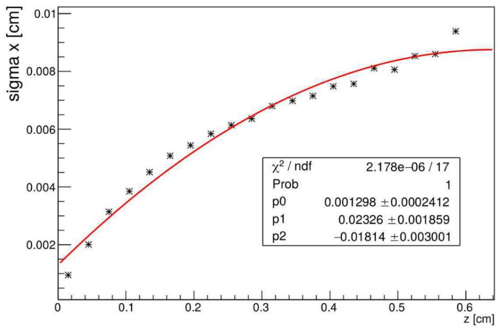
In a **silicon** detector with few hundreds microns of thickness, diffusion is almost negligible but in a **MPGD** with 5mm gap this effect smears the electron position of hundreds of microns and the time by several ns.

Garfield++ simulated the electron diffusion both in semiconductors and gas mixtures but full simulation of this effect could be very time consuming.

A parametrization of the electron diffusion can be generated with a full simulation.

To speed it up the **diffusion** can be **set to zero**





Detector gain

The simulation of a detector amplification needs complex simulations starting from Ansys for the 3D electric field description together with Garfield++ (i.e. gaseous detectors)

This simulation can be very time consuming (10^4 s) if the number of electron reaches high gain performance (10^4-10^5) but simpler approaches can be used:

- sampling from a Polya
- sampling from a full simulation detector gain spectrum

To speed it up the **detector gain** can be set to a **fixed value**

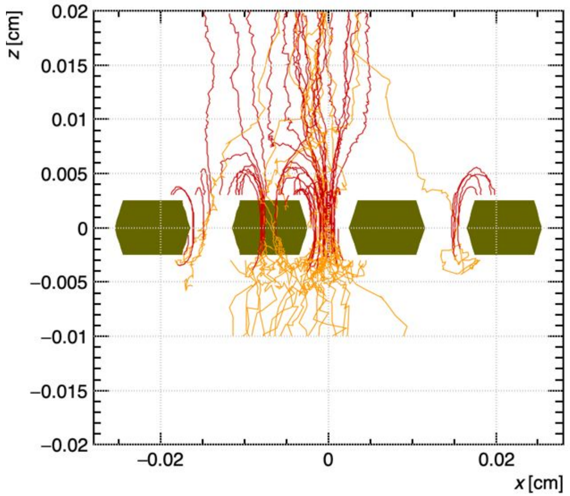
Gain fluctuations → Polya distribution

[G. Iakovidis PhD Thesis, Research and Development in Micromegas Detector for the ATLAS Upgrade]

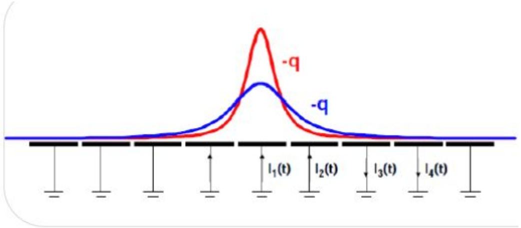
$$P(G) = C_0 \frac{(1+\theta)^{1+\theta}}{\Gamma(1+\theta)} \left(\frac{G}{\overline{G}}\right)^{\theta} \exp\left[-(1+\theta)\frac{G}{\overline{G}}\right]$$

$\overline{G}$ = intrinsic gain mean value

θ → connected to variance



Signal induction

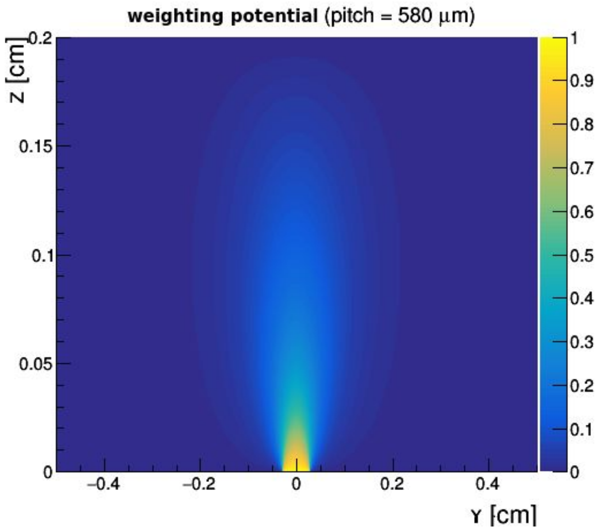


The electrons motion induces a continuous time-dependent current until the electrons arrive to the anode

$$i_{ind}(t) = e \cdot \overline{v}_{drift}(t) \cdot \overline{E}_w(t)$$

The instantaneous current induced is given by the **Shockley-Ramo theorem**.

If the readout is sufficiently large, all the charge *e* is induced otherwise only a fraction of it.



This topic is under study in environments where the radiation rate is so high to generate significant distortion of the electric field in the induction region.

To speed it up the energy/charge measured on the readout corresponds to entire energy loss in the detector and the signal is induced to the closest pixel/pad/wire/strip.

Electronics simulation

It is important to know how the electronics measures the induced signal.

Electronics may significantly affect the detector output

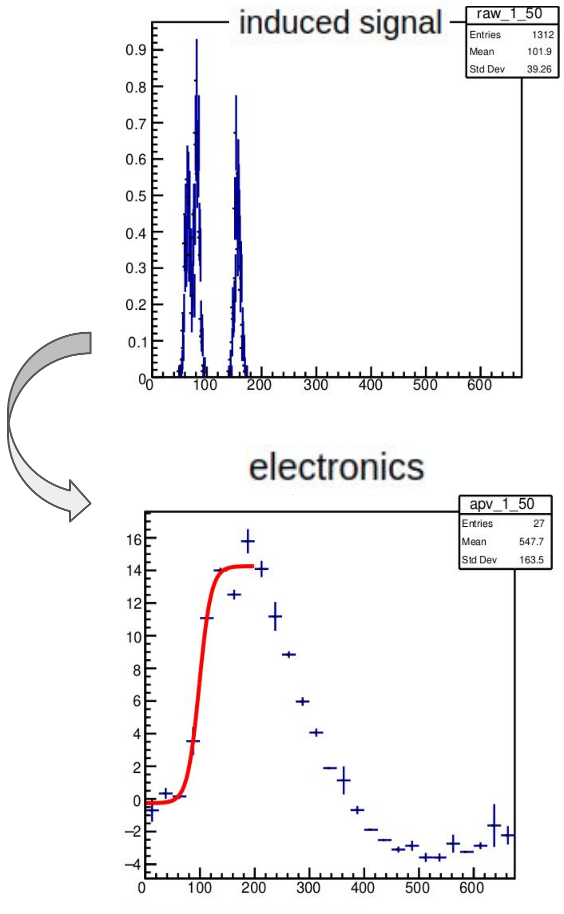
A proper simulation of the **electronics** can be performed with a **convolution** between the **induced signal** and the transfer function:

- a preamplifier–shaper stage (CR–RC) with a given peaking time
- transfer function from the simulated circuit (i.e. Spice)

Electronic **noise** is superimposed to the signal and it can be sampled from real data.

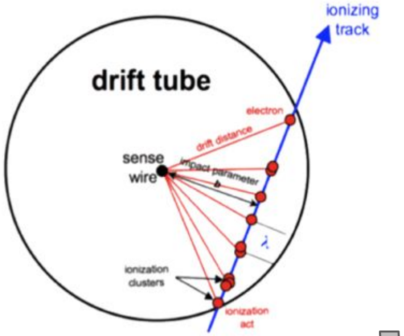
The electronics plays a central role on the energy and time measurement. It is also important to apply a threshold cut on the energy collected from each readout channel or simulated hit.

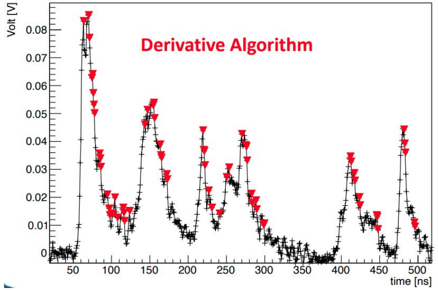
To speed it up the **deposited energy** is converted to the **readout-energy** and there the **threshold is applied**.



Full digitization examples

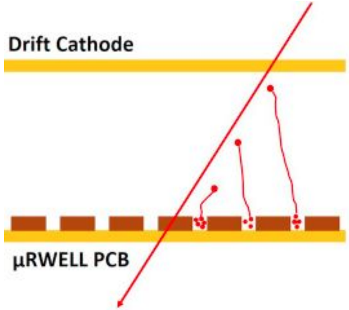
Drift chamber

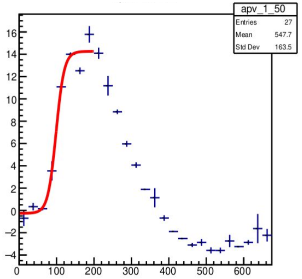




Optimize the cluster counting

Pre-shower & Muon system





Optimize the segmentation

Digitization optimization

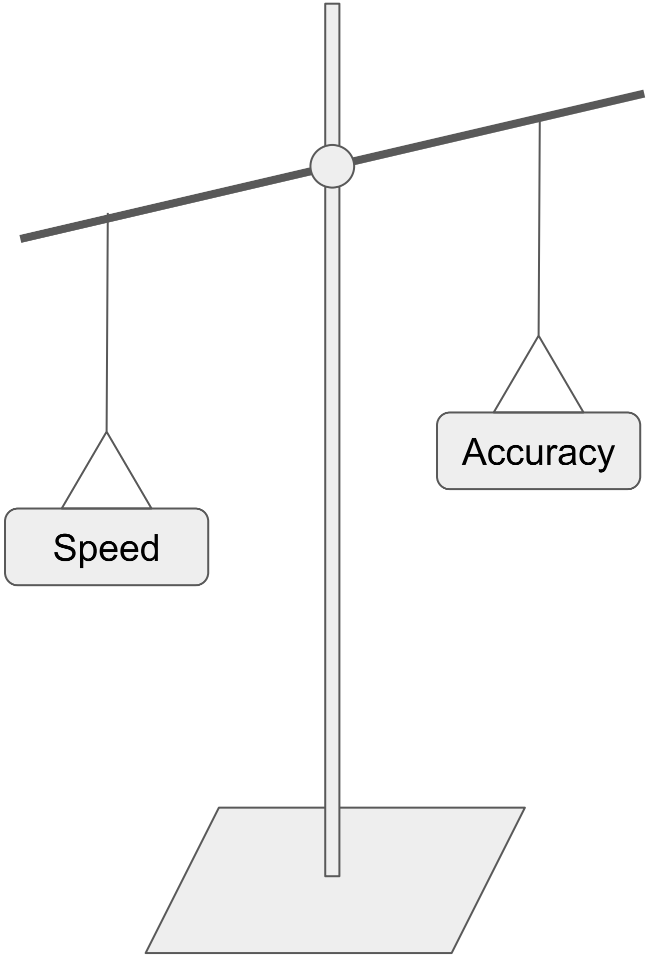
To be or not to be, **speed or accuracy**?

It depends on the experiment's needs; the balance between the two should be taken into account during the detector design.

Optimization is applied to a **full simulation** where the **time-consuming** simulation steps are removed or **parametrized** (i.e. calorimeter shower or signal formation in silicon tracker).

Shortcuts are selected by the detector experts. Interplay between hardware and software is mandatory.

A things to consider is the **person-power cost**!



IDEA digitization summary

```
#-------------  TrackerHit
edm4hep::TrackerHit:
  Description : "Tracker hit"
  Author : "F.Gaede, DESY"
  Members :
    - uint64_t cellID       //ID of the sensor that created this hit
    - int32_t type                     //type of raw data hit, either o
    - int32_t quality                  //quality bit flag of the hit.
    - float time                   //time of the hit.
    - float eDep                   //energy deposited on the hit [GeV].
    - float eDepError              //error measured on EDep [GeV].
    - edm4hep::Vector3d position     //hit position in [mm].
    - std::array<float,6> covMatrix  //covariance of the position (x,y,z),
```

The full IDEA detector is implemented in a stand-alone Geant4 framework at it is fully interfaced to Key4hep.

Now the community is porting the code to DD4hep:

vertex, drift chamber and dual readout calorimeter have been already developed.

The **speeded-up steps** for the trackers are the follow:

1. Geant4 + DD4hep simulated the energy loss in the sensitive volume of each detector

IDEA digitization summary

```
#------------  TrackerHit
edm4hep::TrackerHit:
  Description : "Tracker hit"
  Author : "F.Gaede, DESY"
  Members :
    - uint64_t cellID      //ID of the sensor that created this hit
    - int32_t type                    //type of raw data hit, either or
    - int32_t quality                 //quality bit flag of the hit.
    - float time                  //time of the hit.
    - float eDep                  //energy deposited on the hit [GeV].
    - float eDepError             //error measured on EDep [GeV].
    - edm4hep::Vector3d position    //hit position in [mm].
    - std::array<float,6> covMatrix  //covariance of the position (x,y,z),
```

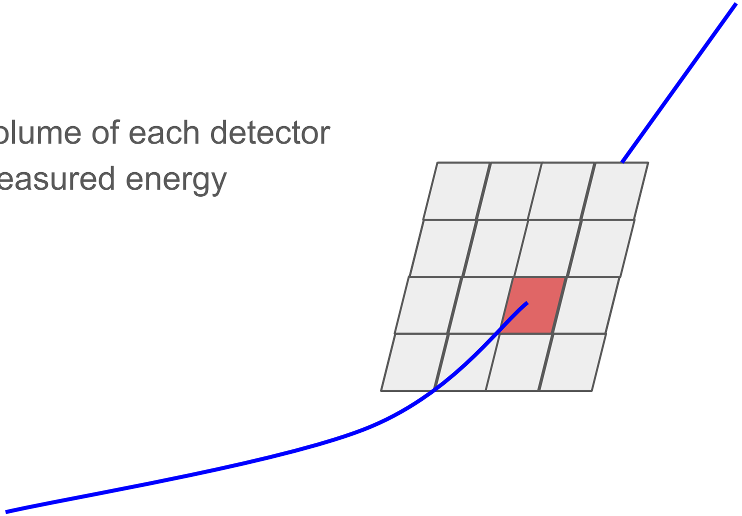
The full IDEA detector is implemented in a stand-alone Geant4 framework at it is fully interfaced to Key4hep.

Now the community is porting the code to DD4hep:

vertex, drift chamber and dual readout calorimeter have been already developed.

The **speeded-up steps** for the trackers are the follow:

1. Geant4 + DD4hep simulated the energy loss in the sensitive volume of each detector
2. the closest pixel, wire, pad or strip to the track reads out the measured energy



IDEA digitization summary

```
#------------  TrackerHit
edm4hep::TrackerHit:
  Description : "Tracker hit"
  Author : "F.Gaede, DESY"
  Members :
    - uint64_t cellID      //ID of the sensor that created this hit
    - int32_t type                     //type of raw data hit, either or
    - int32_t quality                  //quality bit flag of the hit.
    - float time                   //time of the hit.
    - float eDep                   //energy deposited on the hit [GeV].
    - float eDepError              //error measured on EDep [GeV].
    - edm4hep::Vector3d position     //hit position in [mm].
    - std::array<float,6> covMatrix  //covariance of the position (x,y,z),
```

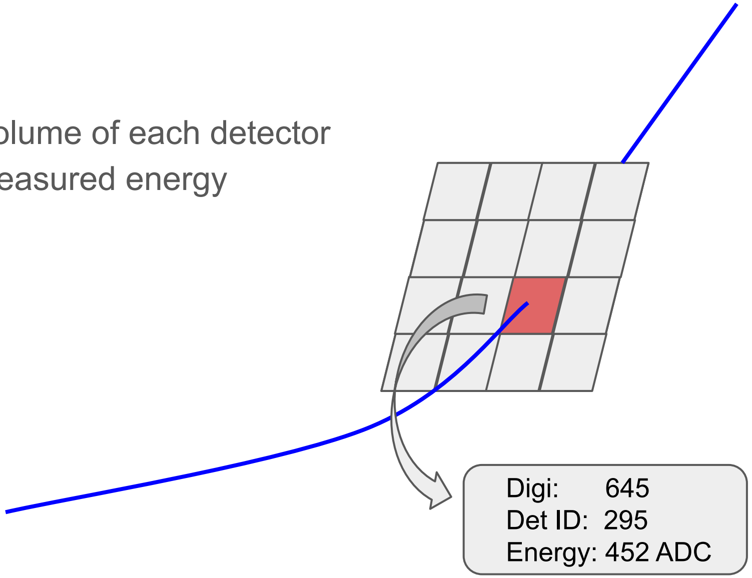
The full IDEA detector is implemented in a stand-alone Geant4 framework at it is fully interfaced to Key4hep.

Now the community is porting the code to DD4hep:

vertex, drift chamber and dual readout calorimeter have been already developed.

The **speeded-up steps** for the trackers are the follow:

1. Geant4 + DD4hep simulated the energy loss in the sensitive volume of each detector
2. the closest pixel, wire, pad or strip to the track reads out the measured energy
3. all the deposited energy is converted to measured energy



IDEA digitization summary

```
#------------  TrackerHit
edm4hep::TrackerHit:
  Description : "Tracker hit"
  Author : "F.Gaede, DESY"
  Members :
    - uint64_t cellID      //ID of the sensor that created this hit
    - int32_t type                     //type of raw data hit, either o
    - int32_t quality                  //quality bit flag of the hit.
    - float time                   //time of the hit.
    - float eDep                   //energy deposited on the hit [GeV].
    - float eDepError              //error measured on EDep [GeV].
    - edm4hep::Vector3d position     //hit position in [mm].
    - std::array<float,6> covMatrix  //covariance of the position (x,y,z),
```

The full IDEA detector is implemented in a stand-alone Geant4 framework at it is fully interfaced to Key4hep.

Now the community is porting the code to DD4hep:
vertex, drift chamber and dual readout calorimeter have been already developed.

The **speeded-up steps** for the trackers are the follow:

1. Geant4 + DD4hep simulated the energy loss in the sensitive volume of each detector
2. the closest pixel, wire, pad or strip to the track reads out the measured energy
3. all the deposited energy is converted to measured energy
4. no signal diffusion, no detector amplification, no electronics (they are too time demanding!)

IDEA digitization summary

```
#-------------  TrackerHit
edm4hep::TrackerHit:
  Description : "Tracker hit"
  Author : "F.Gaede, DESY"
  Members :
    - uint64_t cellID      //ID of the sensor that created this hit
    - int32_t type                      //type of raw data hit, either o
    - int32_t quality                   //quality bit flag of the hit.
    - float time                    //time of the hit.
    - float eDep                    //energy deposited on the hit [GeV].
    - float eDepError               //error measured on EDep [GeV].
    - edm4hep::Vector3d position    //hit position in [mm].
    - std::array<float,6> covMatrix  //covariance of the position (x,y,z),
```

The full IDEA detector is implemented in a stand-alone Geant4 framework at it is fully interfaced to Key4hep.

Now the community is porting the code to DD4hep:

vertex, drift chamber and dual readout calorimeter have been already developed.

The **speeded-up steps** for the trackers are the follow:

1. Geant4 + DD4hep simulated the energy loss in the sensitive volume of each detector
2. the closest pixel, wire, pad or strip to the track reads out the measured energy
3. all the deposited energy is converted to measured energy
4. no signal diffusion, no detector amplification, no electronics (they are too time demanding!)
5. threshold is applied to the measured energy

IDEA digitization summary

```
#------------  TrackerHit
edm4hep::TrackerHit:
  Description : "Tracker hit"
  Author : "F.Gaede, DESY"
  Members :
    - uint64_t cellID      //ID of the sensor that created this hit
    - int32_t type                   //type of raw data hit, either o
    - int32_t quality                //quality bit flag of the hit.
    - float time                  //time of the hit.
    - float eDep                  //energy deposited on the hit [GeV].
    - float eDepError             //error measured on EDep [GeV].
    - edm4hep::Vector3d position  //hit position in [mm].
    - std::array<float,6> covMatrix  //covariance of the position (x,y,z),
```

The full IDEA detector is implemented in a stand-alone Geant4 framework at it is fully interfaced to Key4hep.

Now the community is porting the code to DD4hep:

vertex, drift chamber and dual readout calorimeter have been already developed.

The **speeded-up steps** for the trackers are the follow:

1. Geant4 + DD4hep simulated the energy loss in the sensitive volume of each detector
2. the closest pixel, wire, pad or strip to the track reads out the measured energy
3. all the deposited energy is converted to measured energy
4. no signal diffusion, no detector amplification, no electronics (they are too time demanding!)
5. threshold is applied to the measured energy
6. charge and time are associated to the TrackerHit

IDEA digitization summary

```
#-------------  TrackerHit
edm4hep::TrackerHit:
  Description : "Tracker hit"
  Author : "F.Gaede, DESY"
  Members :
    - uint64_t cellID      //ID of the sensor that created this hit
    - int32_t type                     //type of raw data hit, either o
    - int32_t quality                  //quality bit flag of the hit.
    - float time                   //time of the hit.
    - float eDep                   //energy deposited on the hit [GeV].
    - float eDepError              //error measured on EDep [GeV].
    - edm4hep::Vector3d position     //hit position in [mm].
    - std::array<float,6> covMatrix  //covariance of the position (x,y,z),
```

The full IDEA detector is implemented in a stand-alone Geant4 framework at it is fully interfaced to Key4hep.

Now the community is porting the code to DD4hep:

vertex, drift chamber and dual readout calorimeter have been already developed.

The **speeded-up steps** for the trackers are the follow:

1. Geant4 + DD4hep simulated the energy loss in the sensitive volume of each detector
2. the closest pixel, wire, pad or strip to the track reads out the measured energy
3. all the deposited energy is converted to measured energy
4. no signal diffusion, no detector amplification, no electronics (they are too time demanding!)
5. threshold is applied to the measured energy
6. charge and time are associated to the TrackerHit
7. position measurement differs detector by detector
    a. silicon detectors and MPGD detectors have a digital readout (approx) and they return their own position
    b. drift chamber measures the impact parameters with a fix smearing on the reconstructed position

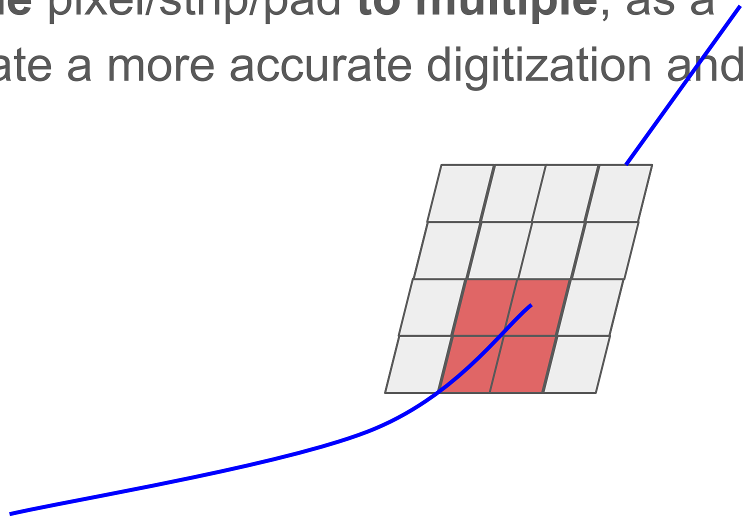
IDEA digitization future plan

First of all, **complete the DD4hep IDEA description** to allow the FCC-ee community to do performance studies of the IDEA detector concept

IDEA digitization future plan

First of all, **complete the DD4hep IDEA description** to allow the FCC-ee community to do performance studies of the IDEA detector concept

Extend the readout signal formation **from single** pixel/strip/pad **to multiple**, as a function of the impinging angle. This will generate a more accurate digitization and will allow to develop **clustering algorithms**



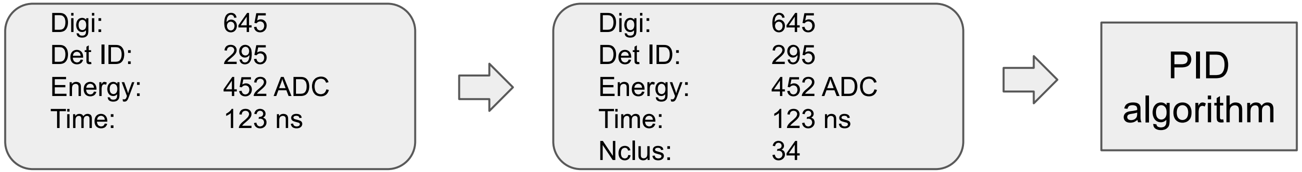
IDEA digitization future plan

First of all, **complete the DD4hep IDEA description** to allow the FCC-ee community to do performance studies of the IDEA detector concept

Extend the readout signal formation **from single** pixel/strip/pad **to multiple**, as a function of the impinging angle. This will generate a more accurate digitization and will allow to develop **clustering algorithms**

Update the TrackerHit data model to allow the cluster counting technique:

“**Ncluster**” variable to be implemented



IDEA digitization future plan

First of all, **complete the DD4hep IDEA description** to allow the FCC-ee community to do performance studies of the IDEA detector concept

Extend the readout signal formation **from single** pixel/strip/pad **to multiple**, as a function of the impinging angle. This will generate a more accurate digitization and will allow to develop **clustering algorithms**

Update the TrackerHit data model to allow the cluster counting technique:

    “**Ncluster**” variable to be implemented

Neural Network are considered to parametrize and simplify the digitization

    i.e. path length in the sensitive volume and the number of cluster

Thank You

This project is supported from the European Union's Horizon 2020 research and innovation programme under grants agreements
No 951754 (H2020 H2020-INFRADEV-2018-2020 FCCIS) and
No 101004761 (H2020-INFRAINNOV-2019-2020 Aidainnova)

Literature and link

André Sailer - Simulation: Going Fast, Full and Fully Fast
https://indico.desy.de/event/33640/contributions/125939/attachments/77621/100455/221006_sailer_simulation.pdf

Erica Brondolin - Reconstruction: Imaging the Events
https://indico.desy.de/event/33640/contributions/125940/attachments/77620/100422/20221005_RecoInHiggsFactories_ECFA_final.pdf

Github - key4hep/EDM4hep/edm4hep.yaml
https://github.com/key4hep/EDM4hep/blob/master/edm4hep.yaml

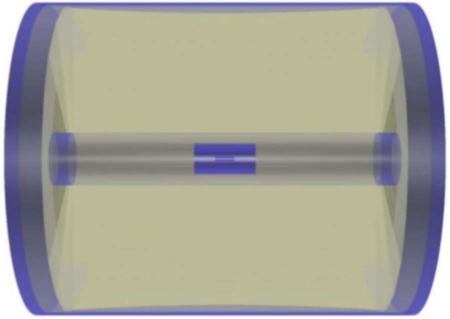
Federica Cuna - Simulation of particle identification with the cluster counting technique
https://www.researchgate.net/publication/351658318_Simulation_of_particle_identification_with_the_cluster_counting_technique

Lia Lavezzi - PARSIFAL: a toolkit for triple-GEM parametrized simulation
https://www.researchgate.net/publication/341310679_PARSIFAL_a_toolkit_for_triple-GEM_parametrized_simulation

Riccardo Farinelli - A parametric simulation of the µ-RWELL detector
https://indico.cern.ch/event/1138814/contributions/4918486/attachments/2462640/4222374/A%20parametric%20simulation%20of%20the%20%C2%B5-RWELL%20detector.pdf

GEANT4: tracking system

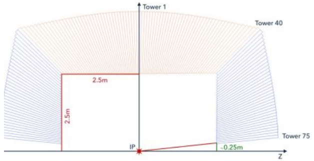
- Si vertex, Si wrapper and preshower are simulated as simple layers
- Drift chamber simulated good level of geometry details, including wires and detailed description of the endcaps
- 56448 cells (≈1.2 cm) for a total of 343968 wires

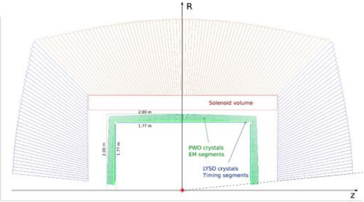


New VTX detector and MDI ?
F. Palla + Armin

GEANT4: Dual Readout Calorimeter Status + Crystal Option

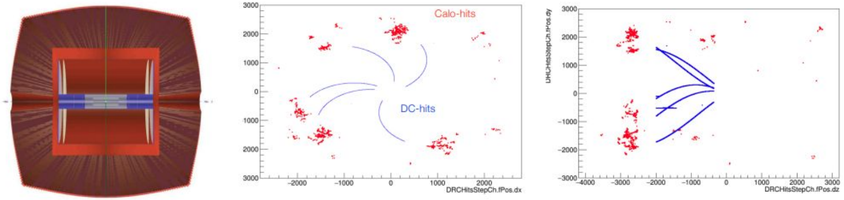
- **Fully projective fiber calorimeter description ported from GEANT4 to DD4HEP geometry** (with Sussex and South Korea)
- **Recently proposed, a dual-readout crystal em calorimeter integrated** in the existing Geant4 Calorimeter application for performance studies
  - Goal: maintain the key capability to correct for fluctuations of the electromagnetic fraction in hadronic showers while boosting the energy resolution for photons and electrons to about 3-4%/√E





GEANT4: tracking system + DR CALO

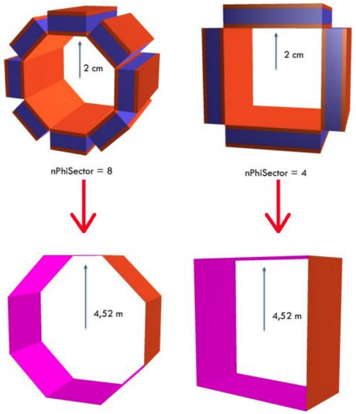
- **Standalone GEANT4 implementation of IDEA** Tracker+Calo



- **Code to translate the output (hits & Tracks) of the standalone G4 simulation of the IDEA detector in the new EDM4HEP format.** This allows the communities of FCC-ee and CEPC to profit of the IDEA FullSimulation response for performance studies
  - Having reconstructed tracks from FullSim available in the events in the EDM4HEP format is crucial for further development of many other tools (Particle Flow, particle ID, LLP etc…)

GEANT4: preshower and muon counters

Material stratification implemented



See slides by I. Garzia