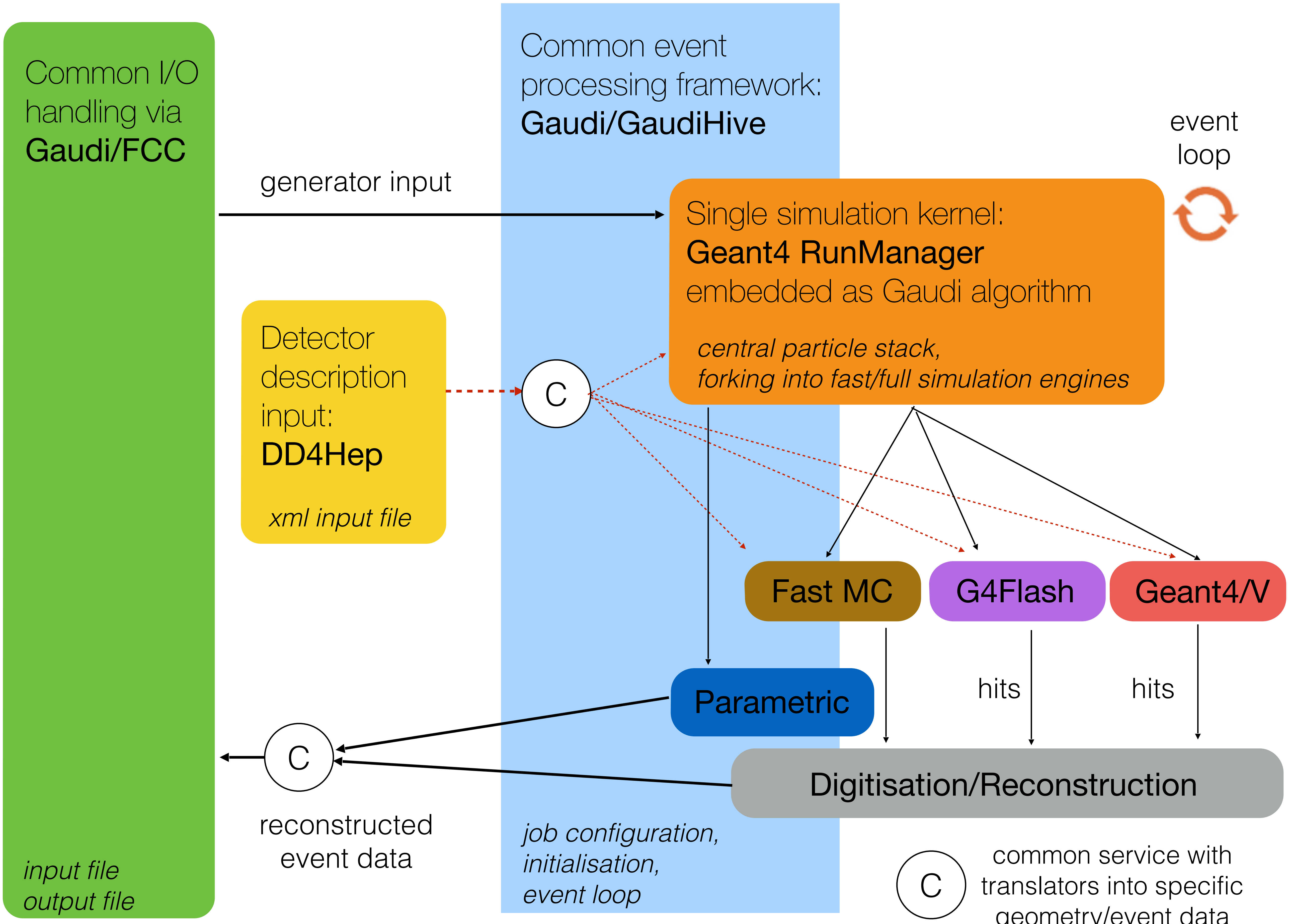
The Tracker description and Interface to Gaudi

Experiences and first Results

J. Hrdinka, A.Salzburger



Common I/O handling via
Gaudi/FCC
input file
output file
Common event processing framework:
Gaudi/GaudiHive
job configuration,
initialisation,
event loop
generator input
event loop
Single simulation kernel:
Geant4 RunManager
embedded as Gaudi algorithm
central particle stack,
forking into fast/full simulation engines
Detector description input:
DD4Hep
xml input file
C
Fast MC
G4Flash
Geant4/V
hits
hits
Parametric
Digitisation/Reconstruction
C
reconstructed event data
C
common service with translators into specific geometry/event data

DD4hep - the geometry input

- **full detector description**
  including geometry, material and readout segmentation
- **simple description**
  xml-file + corresponding constructor in c++
- **Root based geometry**
  TGeo package
- **Integration in Geant4**
  translation in Geant4 geometry already provided
- **Integration in Gaudi**
  to access both the DD4hep and the Geant4 geometry within the framework "**GeoSvc"** was introduced

  ➔ loads geometry from the xml-file and invokes detector construction in DD4hep

  ➔ invokes translation to Geant4

  ➔ provides the detector geometries to algorithms

Reconstruction Geometry

- **simplified geometry**
  for reconstruction purposes and fast simulation

- **based on LHC experience**
  ATLAS reconstruction geometry as guideline, enable translation from DD4hep
  =>construction of TGeoVolumes possible
  =>using DD4hep readout segmentation

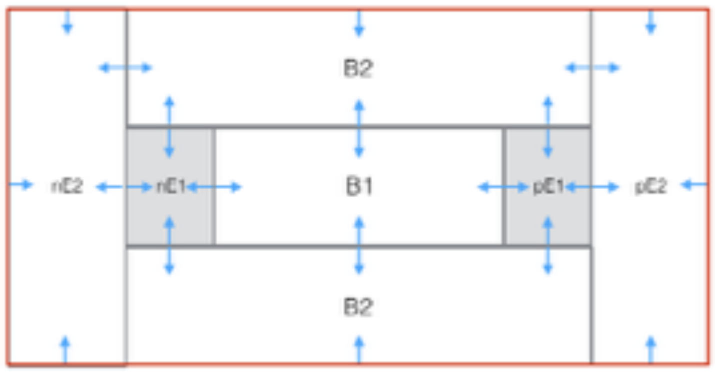
- **Surface description**
  essential for tracking, reconstruction and fast simulation

- **enabling navigation**

  ➔ layers containing the modules (surfaces) pointing to next/previous layer

  ➔ volumes containing this layers, bordered by boundary surfaces pointing to next/previous volume

  ➔ container volumes composed of volumes
  => tree structure



Simplified Reconstruction Geometry

- **Simplification**

One Module (consisting of different components)

=

one Surface in the reconstruction geometry

- **Approximation of the material**
(once per module type, described in a 2D grid):

$$\frac{t}{x_0} = \sum_{i=1}^{n} \frac{t_i}{x_i}$$

$$\frac{t}{\Lambda_0} = \sum_{i=1}^{n} \frac{t_i}{\Lambda_i}$$

$$sensper = \frac{\sum_{i=1}^{n} sensV_i}{\sum_{i=1}^{n} V_i}$$

$$\rho = \frac{\sum_{i=1}^{n} t_i \rho_i}{\sum_{i=1}^{n} t_i}$$

$$A = \frac{\sum_{i=1}^{n} \rho_i A_i}{\sum_{i=1}^{n} \rho_i}$$

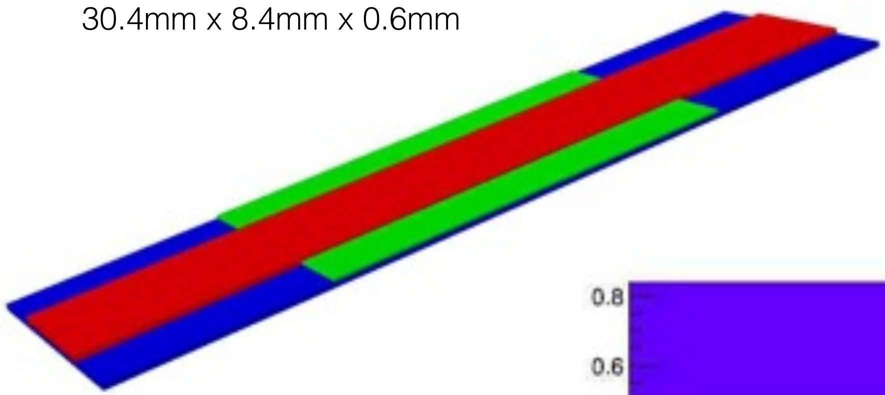
$$Z = \frac{\sum_{i=1}^{n} \rho_i Z_i}{\sum_{i=1}^{n} \rho_i}$$

t...thickness, x_0...radiation length, Λ_0...interaction length, sensper...sensitive percentage, V...Volume, ρ...density, A...mass number, Z...atomic number

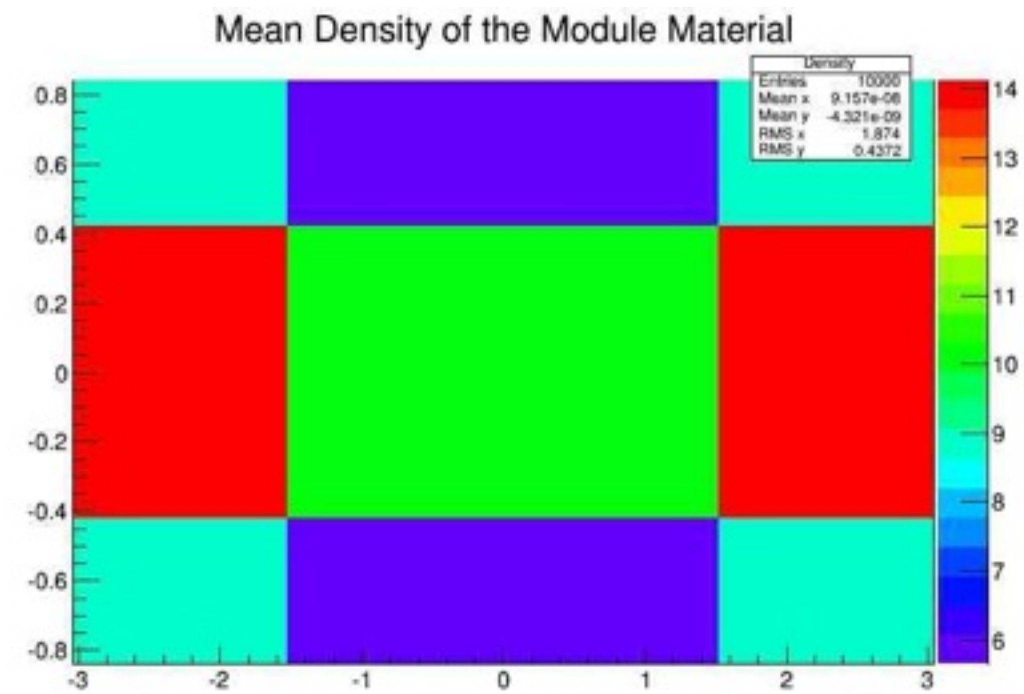
Approximation of a Module

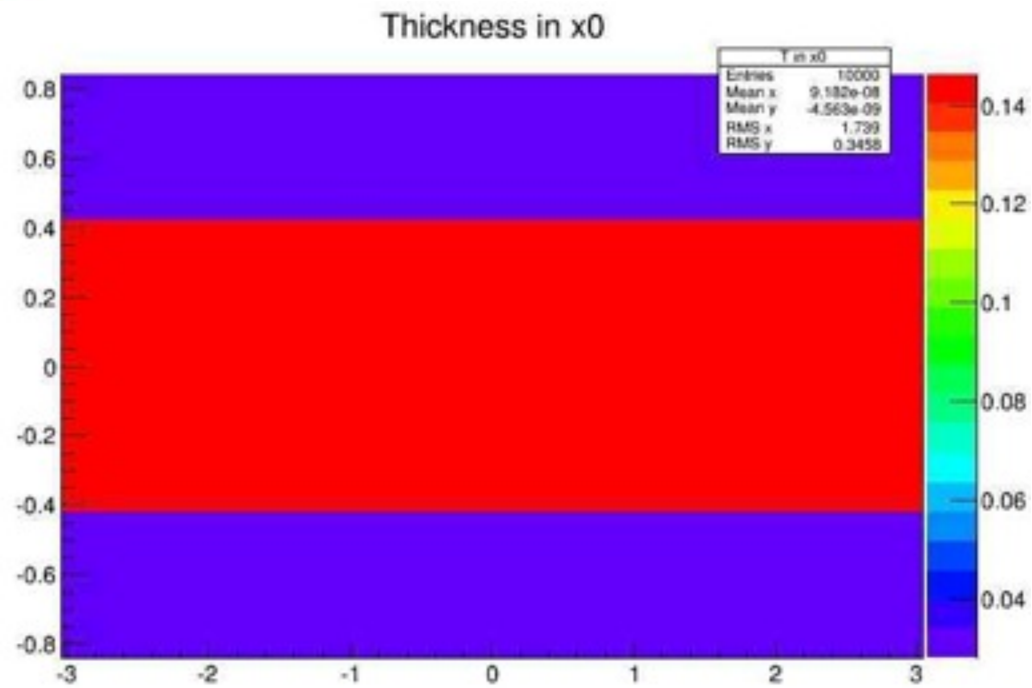
Example

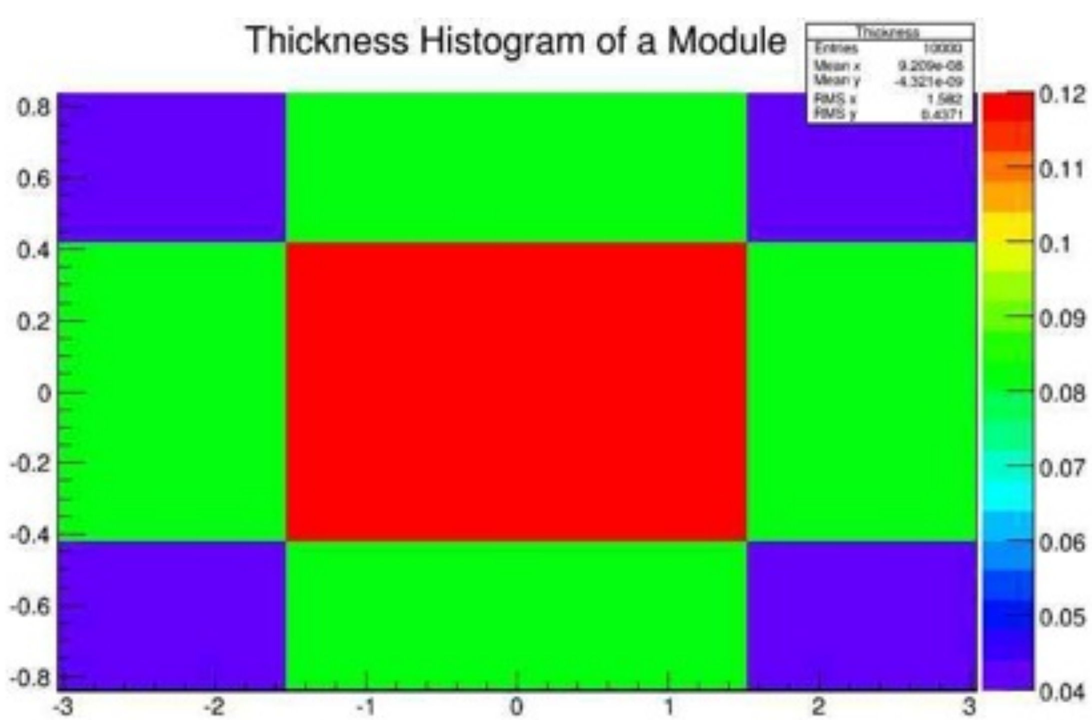
ModuleA
30.4mm x 8.4mm x 0.6mm

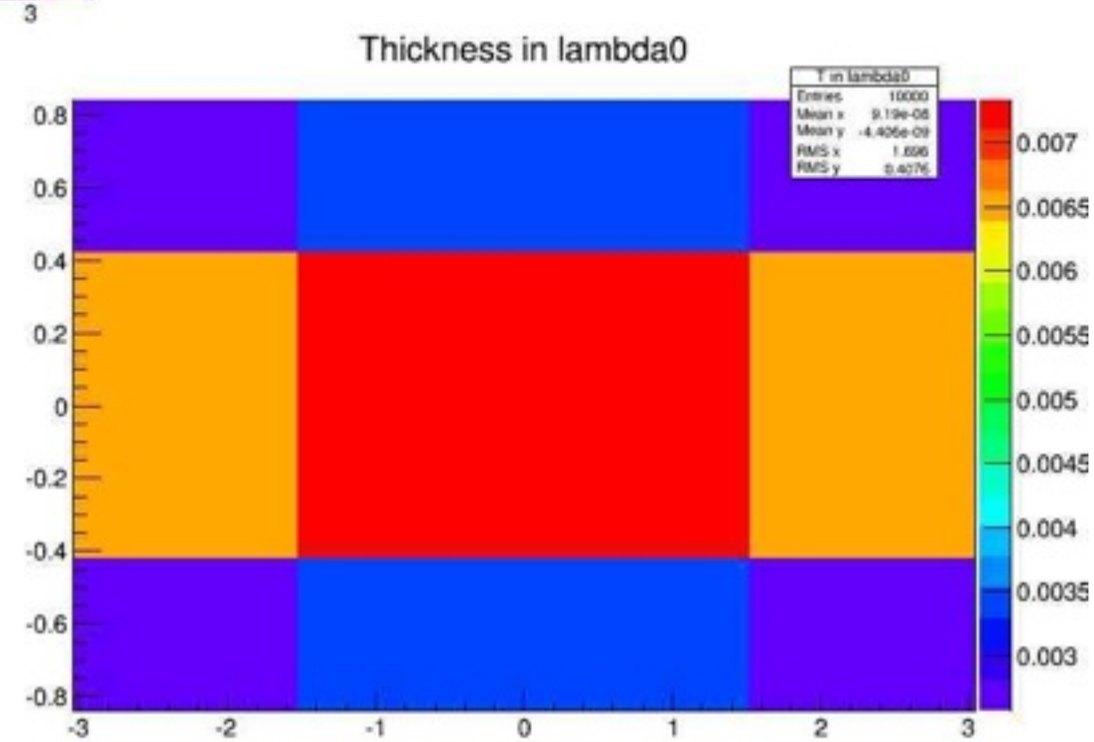


- Copper (30.4mm x 8.4mm x 0.2mm)
- Silicon (15.2mm x 8.4mm x 0.2mm)
- Tungsten (30.4mm x 4.2mm x 0.2mm)









Towards an automatic translation

- **IRecoGeoSvc**
  Interface for different translations

- **ClassicalRecoGeoSvc**
  “classical” tracker types

  - barrels with corresponding end caps nested over each other, containing layers of modules, composed of components

=> allowing implementation of translation, for more specific detector

- **Access DD4hep geometry : world “DetElement”**

  ➡ walk through “detector tree” - parallel to volume tree

  ➡ access geometry + detector information

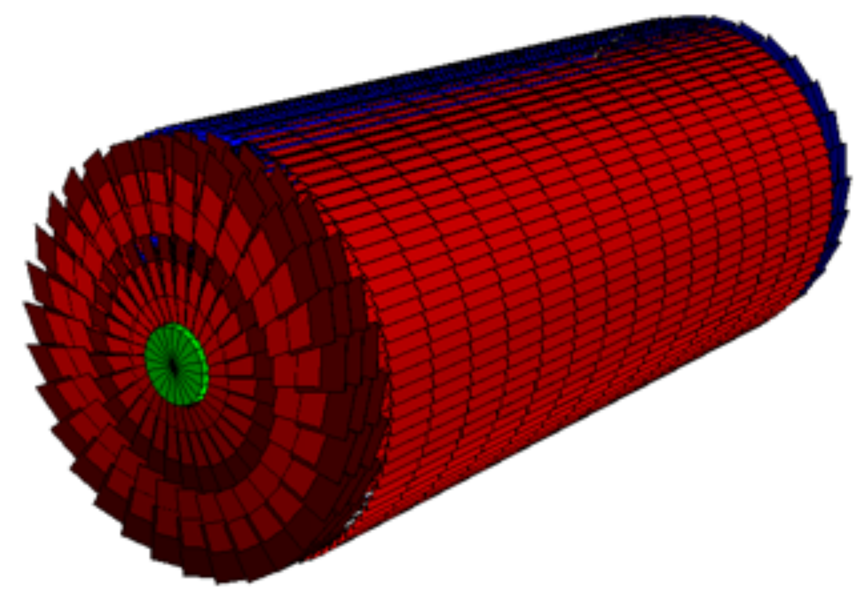
- **DD4hep Extension mechanism** of DetElement
  possibility to distinguish between the different Detector elements

  ➡ extended by different volume, layer, module and sensitive component classes

  ➡ handing over readout information

First Test Tracker

Building a first TestTracker:

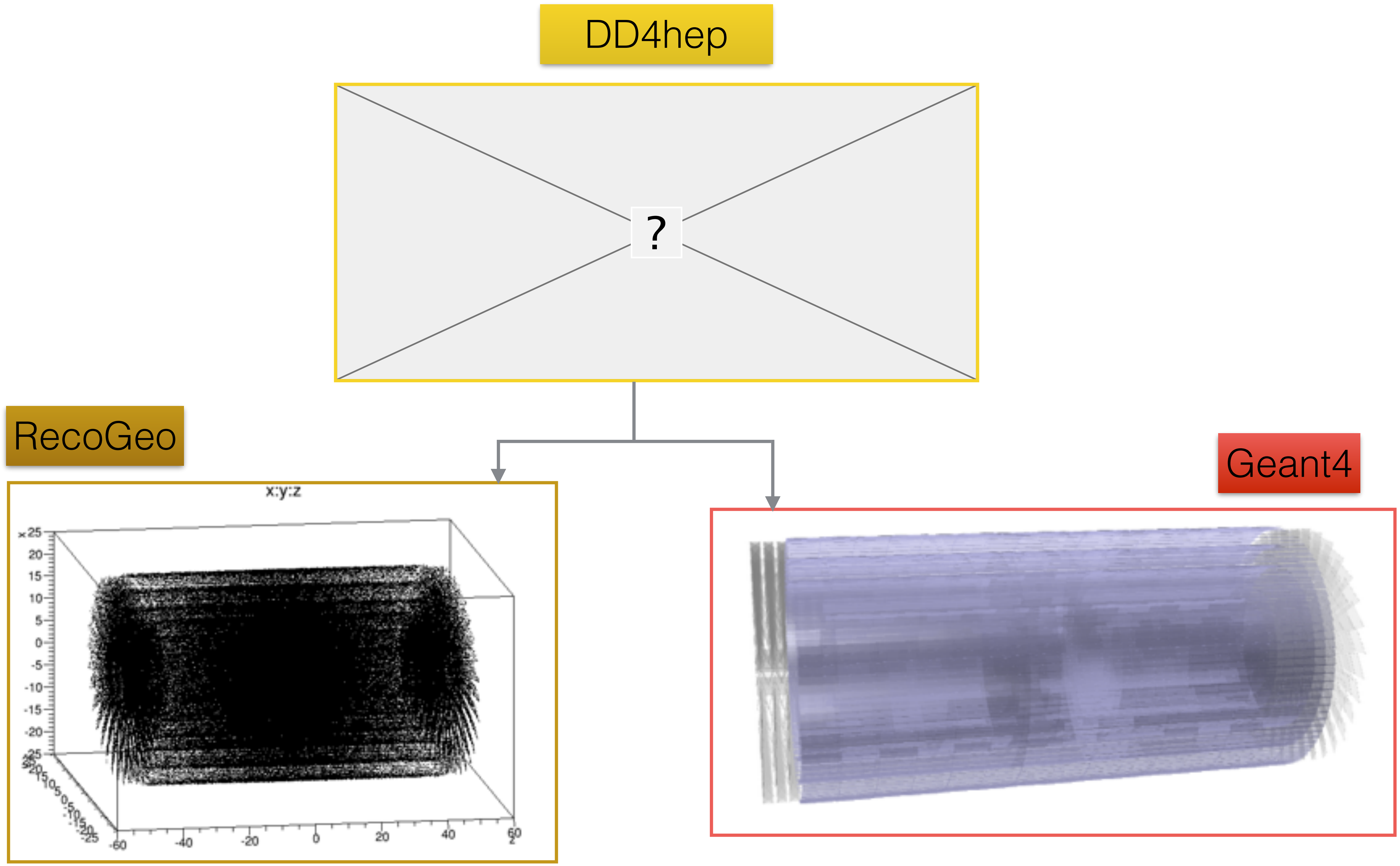


Two hierarchies:

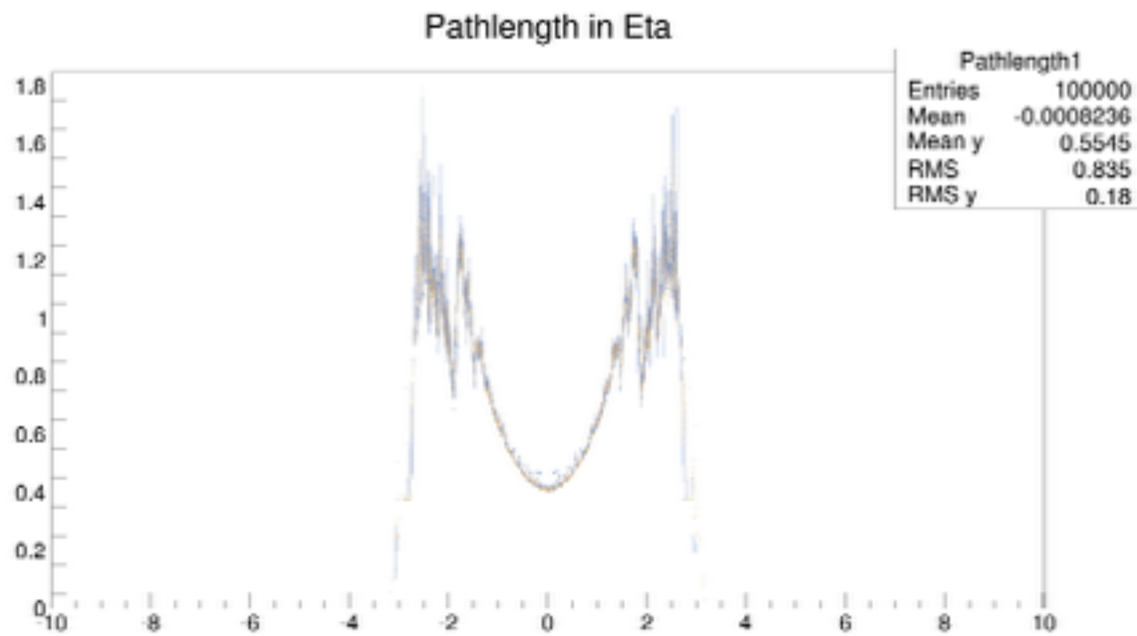
- two end caps with three disc layers each
- one barrel volume with three cylinder layers
- each layer is filled with modules
- each module is composed of three components

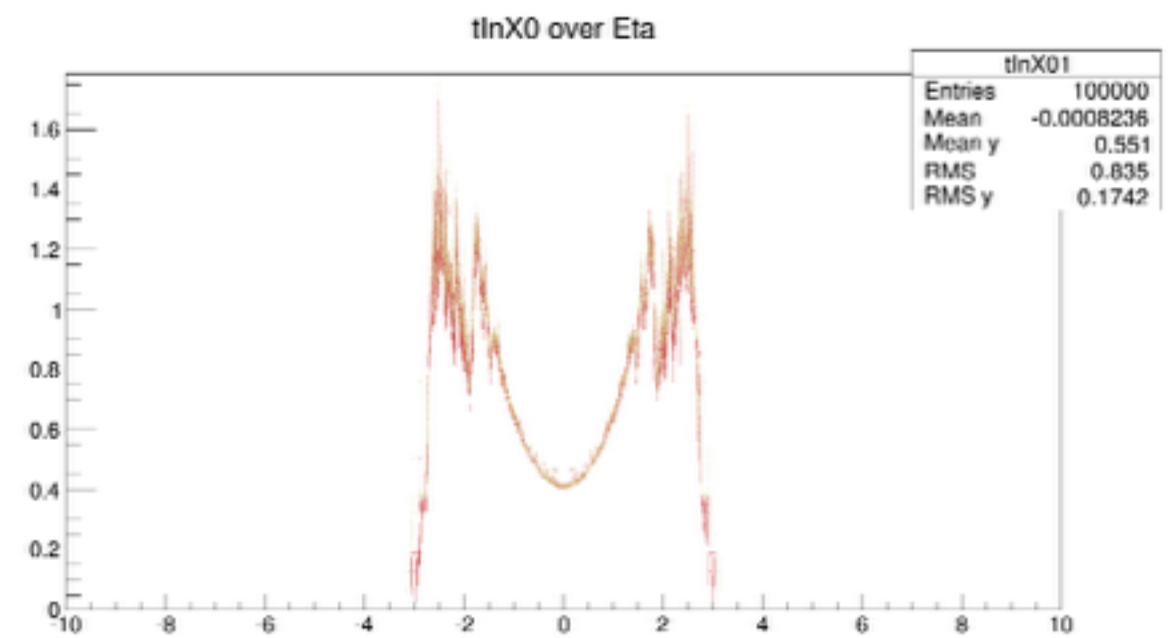
Number of hierarchies, layers, modules and components is variable

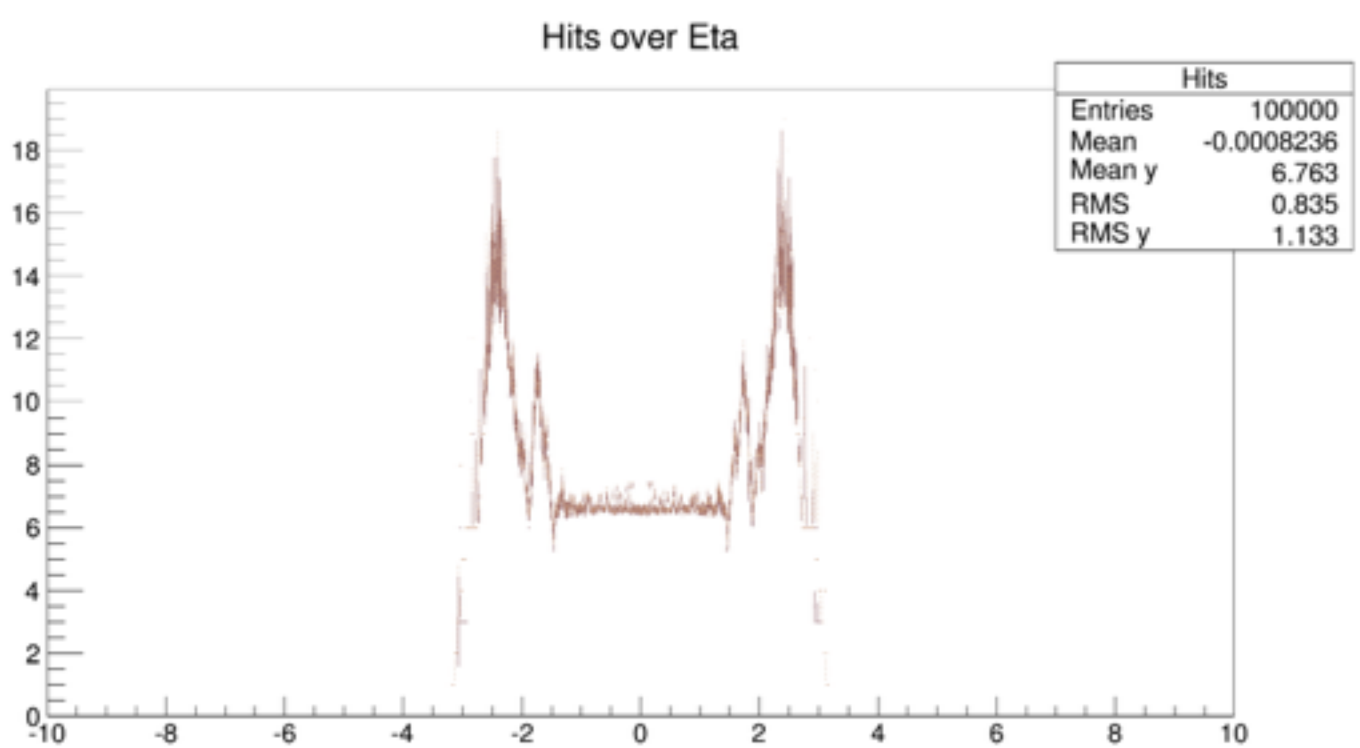
Comparing geometries



Geant4 vs. RecoGeo







Conclusion & Next steps

- We are able to build a first test tracker and provide it in the Geant4 and the reconstruction geometry
- Implementation of magnetic field transport & track fitting from ATLAS code
- Proof of principle
    - both full and fast simulation can be invoked from one common source
    - create tracks from truth particles via full simulation and fast simulation
    - input for parametric simulation
    - compare output