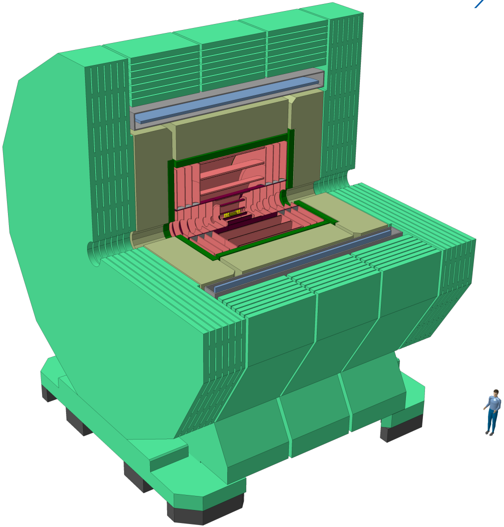
Status of CLD Software

André Sailer

CERN-EP-SFT

FCC Physics Workshop
February 11, 2022

Table of Contents

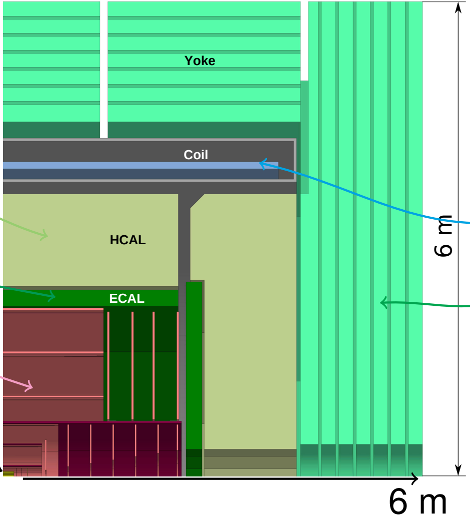


Software

- CLD Software based on Key4hep: All software available on Key4hep CVMFS `/cvmfs/sw-nightlies.hsf.org/key4hep/setup.sh`
  - DD4hep detector models
    - **FCCee_o1_v04 (lcgeo)**: detector model used for most of the performance note [1]
    - **FCCee_o2_v01 (lcgeo)**: detector model with updated beam pipe/VXD radii (10 mm)
    - **FCCee_o2_v02 (FCCDetectors)**:

Detector for FCCee

General purpose detector for Particle Flow reconstruction [1]

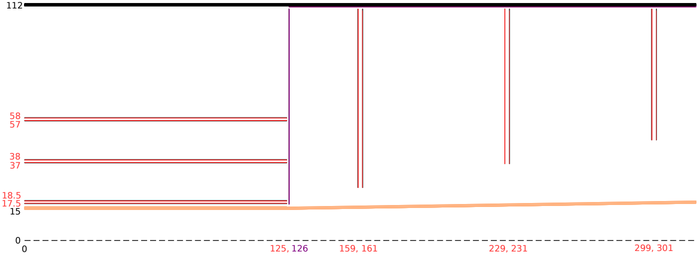


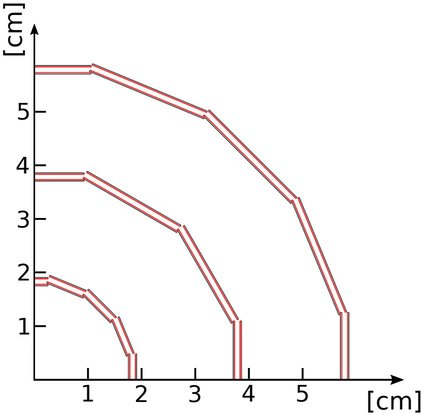
- Steel–Scintillator HCal with 3 cm cell-size
- Silicon–Tungsten ECal with 5 mm cell-size
- Silicon Tracker, mostly 50 µm pitch strips
- Vertex Detector with 25 µm pixels

- Superconducting Solenoid of 2 T
- Iron Yoke with RPCs for Muon ID

Vertex Detector

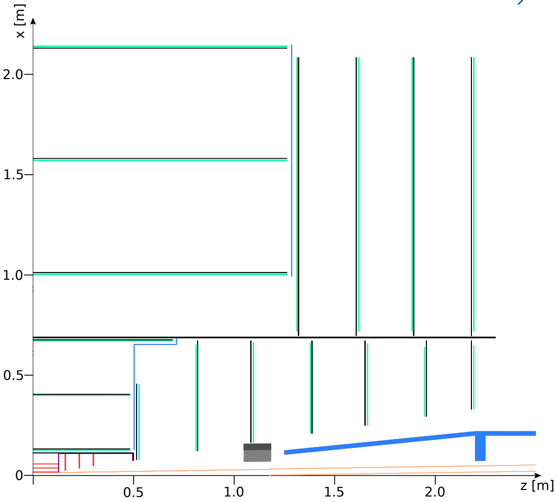
- Silicon vertex detector: precise vertex reconstruction
- Double layers (0.3%X_0 per detection layer)
- $R_{\text{in}} = 17.5$ mm





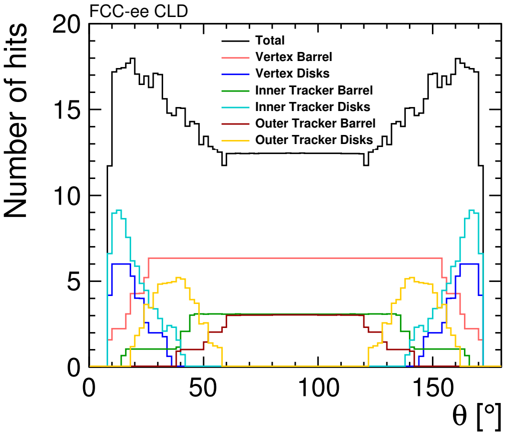
Silicon Tracking

- Inner and Outer Tracker
  - Support tube for extraction with beam-pipe assembly
- 3 short and 3 long barrel layers
- 7 inner and 4 outer endcaps
- Some studies for re-scaling were done [1]
- At least 8 hits for $\theta > 8.5^\circ$



Silicon Tracking

- Inner and Outer Tracker
  - Support tube for extraction with beam-pipe assembly
- 3 short and 3 long barrel layers
- 7 inner and 4 outer endcaps
- Some studies for re-scaling were done [1]
- **At least 8 hits for $\theta > 8.5^{\circ}$**



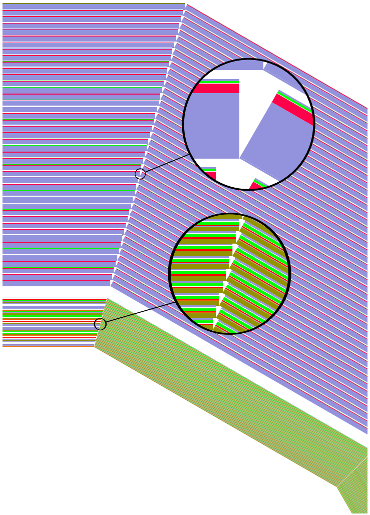
Calorimeters

ECal

- 40 layers, 1.9 mm tungsten absorber, 22 X_0,
- silicon sensors with 5×5 mm^2 granularity
- ECal optimisation studies [1, 2]

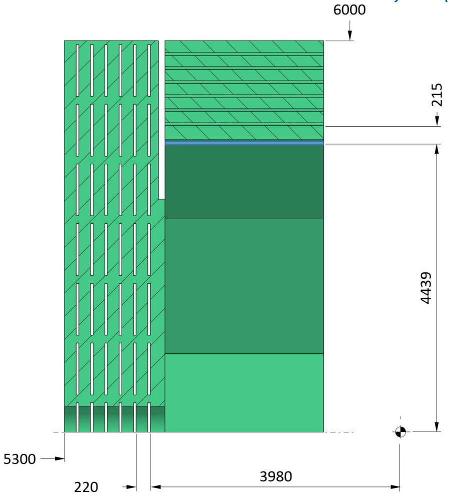
HCal

- 44 layers, 19 mm steel absorber, 5.5 (+1) λ_I
- scintillator tiles with 3×3 cm^2 granularity



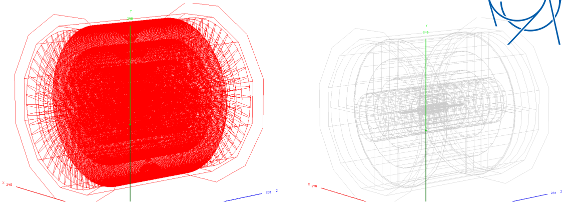
Magnet and Muon System

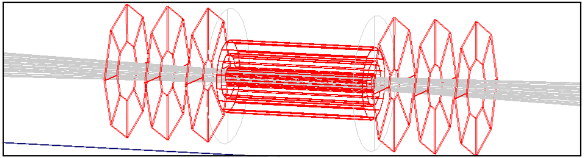
- 2 Tesla Solenoid Field
- Return yoke contains muon system with 6 equidistant layers

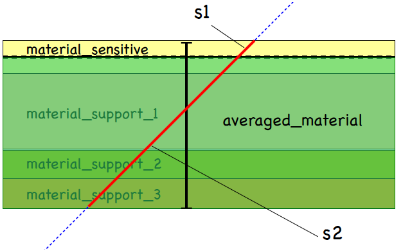


Surfaces

- The DD4hep detector model contains surfaces used for track reconstruction
- For each sensitive sensor (red) and in the support material inside the tracking region (gray, sorry), containing material information
- Around the IP for extrapolating to the IP
- Before the ECal to extrapolate tracks for cluster matching in the particle flow reconstruction





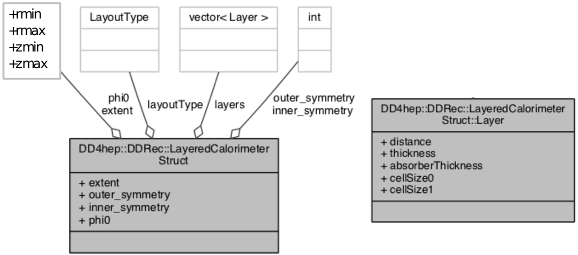


High Level Information for Reconstruction

High level view onto the detectors through DDRec `DataStructures` extensions for `DetElements`

- Constructors fill DDRec `DataStructures`
- Dimensions, positions, parameters for layers,. . .
- `DataStructures` allow to decouple detector implementation from reconstruction algorithms
- Detector drivers responsible to fill and attach appropriate `DataStructures` to sub-detectors

`DataStructures` contain sufficient information to provide geometry information to particle flow clustering via `PandoraPFA`



Data Structure	Detector Type
`ConicalSupportData`	Cones and Tubes
`LayeredCalorimeterData`	Sandwich Calorimeters
`ZPlanarData`	Planar Silicon Trackers
`ZDiskPetalsData`	Forward Silicon Trackers

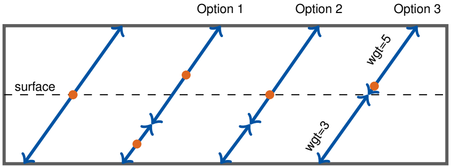
Full Simulation with DDSim

- We can run the simulation with ddsim and the CLD specific steering file
    - 30 mrad crossing angle
    - configuration of sensitive detectors, particularly Scintillator HCal (Birks' Law), and silicon tracking sensors
    - Physics list, simulation parameters (e.g., stepper, range cut)
- Configuration using `k4SimGeant4` still has to be established

```
/cvmfs/sw-nightlies.hsf.org/key4hep/setup.sh
git clone https://github.com/iLCSoft/CLICPerformance
cd CLICPerformance/fcceeConfig/
ddsim --compactFile $LCGEO/FCCee/compact/FCCee_o2_v01/FCCee_o2_v01.xml \
      --inputFiles ../Tests/yyxyev_000.stdhep \
      --numberOfEvents 3 \
      --steeringFile fcc_steer.py \
      --outputFile tops_edm4hep.root
```

Silicon Tracker Setup

- Different ways to count hits on the sensitive detectors: per step (option 1), per particle (option 2), per particle weighted by energy deposit (option 3)
- For the current track reconstruction, the hits are placed on the sensitive surfaces.
  - Surface defines local coordinate system also for error ellipses
- Using `Option 2` hits are by construction on the surface (unless they leave the sensor on the sides)
  - Energy thresholds are taken against combined deposit, simplifies the simple digitisation



More options for DDSim

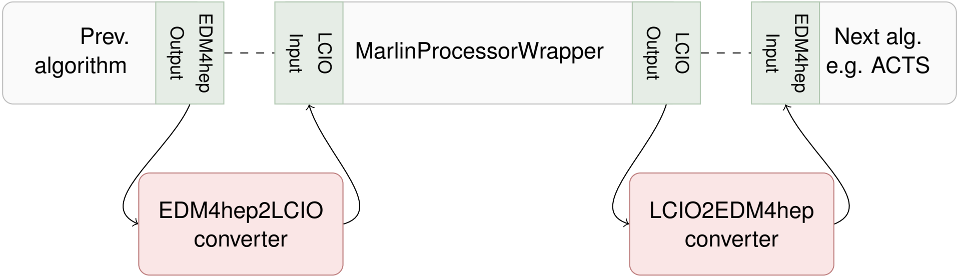
- `ddsim` python executable is part of the DD4hep release [3]
- Get steering file `ddsim --dumpSteeringFile > mySteer.py`
  - Steering file includes documentation for parameters and examples
  - The python file contains a `DD4hepSimulation` object at global scope
  - Configure simulation directly from command-line
  - Input: Particle Gun, stdhep, HepMC, slcio, GuineaPig Pairs; EDM4hep forthcoming

```python
from DDSim.DD4hepSimulation import DD4hepSimulation
from SystemOfUnits import mm, GeV, MeV, keV
SIM = DD4hepSimulation()
SIM.compactFile = ”CLIC_o3_v06.xml”
SIM.runType = ”batch”
SIM.numberOfEvents = 2
SIM.inputFile = ”electrons.HEPEvt”
SIM.part.minimalKineticEnergy = 1*MeV
SIM.filter.filters ['edep3kev'] =
dict (name=”EnergyDepositMinimumCut/3keV” ,
      parameter={”Cut” : 3.0*keV} )
```

```
$ ddsim
--action.calo                --filter.tracker             --part.keepAllParticles
--action.mapActions          -G                           --part.minimalKineticEnergy
--action.tracker             --gun.direction              --part.printEndTracking
--compactFile                --gun.energy                 --part.printStartTracking
--crossingAngleBoost         --gun.isotrop                --part.saveProcesses
--dump                       --gun.multiplicity           --physics.decays
--dumpParameter              --gun.particle               --physics.list
--dumpSteeringFile           --gun.position               --physicsList
--enableDetailedShowerMode   -h                           --physics.rangecut
--enableGun                  --help                       --printLevel
--field.delta_chord          -I                           --random.file
--field.delta_intersection   --inputFiles                 --random.luxury
--field.delta_one_step       -M                           --random.replace_gRandom
--field.eps_max              --macroFile                  --random.seed
--field.eps_min              -N                           --random.type
--field.equation             --numberOfEvents             --runType
--field.largest_step         -O                           -S
--field.min_chord_step       --outputFile                 --skipNEvents
--field.stepper              --output.inputStage          --steeringFile
--filter.calo                --output.kernel              -v
--filter.filters             --output.part                --vertexOffset
--filter.mapDetFilter        --output.random              --vertexSigma
```

Reconstruction

- Reconstruction, consisting of,
  - Background Overlay, Digitisation
  - Track Pattern Recognition (ConformalTracking [4]), track fit
  - Particle Flow Reconstruction (PandoraPFA [5])
  - Vertexing and Flavour Tagging (LCFIplus [6])
- Run with Gaudi via the k4MarlinWrapper: `k4run fccRec_e4h_input.py`
  - Input and output in EDM4hep
  - Steering file will be available **soon**



Prev. algorithm

EDM4hep Output

LCIO Input

MarlinProcessorWrapper

LCIO Output

EDM4hep Input

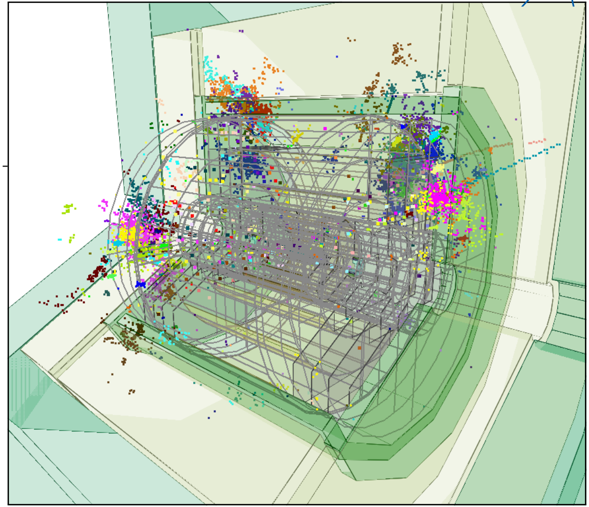
Next alg. e.g. ACTS

EDM4hep2LCIO converter

LCIO2EDM4hep converter

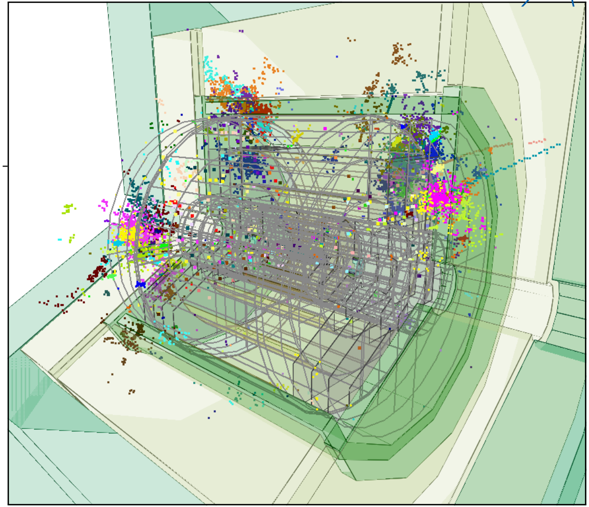
Simulated Event

-
  ```
  ced2go -d
  $LCGEO/FCCee/compact/FCCee_o2_v01/FCCee_
  -v CEDViewer sim.slcio
  ```

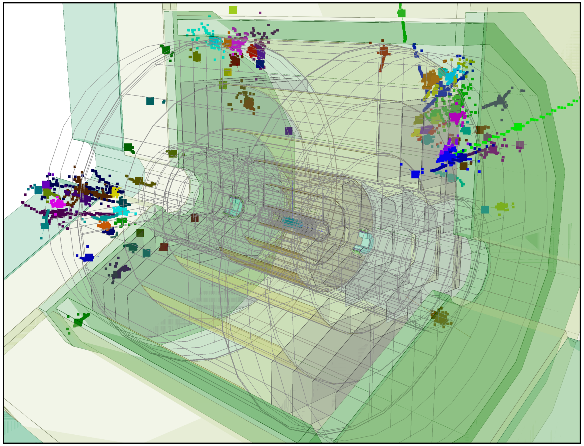


Simulated Event

- `ced2go -d $LCGEO/FCCee/compact/FCCee_o2_v01/FCCee_ -v CEDViewer sim.slcio`
- Needs simulation output in slcio format (`ddsim ... -O sim.slcio ...`)
- `ced2go` is a wrapper around Marlin running a `CEDViewer` processor, so in principle we should be able to use this event display via `k4MarlinWrapper` and `EDM4hep` as well. . .



Reconstructed Event



- After reconstruction, showing reconstructed clusters (big squares) and associated hits
- Some pruning of low energy or out of time hits occurred with respect to the previous slide as well.

Summary

- The state of the CLD software is strong
- Simulation and reconstruction for current detector model exists in Key4hep
- Need validation to be sure nothing broke since the last studies were done
- The detector model itself could benefit from an update: beam pipe, tracker layout, ECal layout, forward region

References

[1] N. Bacchetta et al. *CLD – A Detector Concept for the FCC-ee*. 2019. arXiv: `1911.12230 [physics.ins-det]`.

[2] Oleksandr Viazlo and Andre Sailer. “Efficient Iterative Calibration on the Grid using iLCDirac”. In: *EPJ Web Conf.* 245 (2020), p. 03003. DOI: `10.1051/epjconf/202024503003`. URL: `https://doi.org/10.1051/epjconf/202024503003`.

[3] M. Frank et al. “DDG4: A Simulation Framework using the DD4hep Detector Description Toolkit”. In: *J. Phys. Conf. Ser.* 664 (Apr. 2015), p. 072017. DOI: `10.1088/1742-6596/664/7/072017`.

[4] E. Brondolin et al. “Conformal tracking for all-silicon trackers at future electron–positron colliders”. In: *Nucl. Instrum. Meth.* A956 (2020), p. 163304. ISSN: 0168-9002. DOI: `https://doi.org/10.1016/j.nima.2019.163304`.

[5] J.S. Marshall and M.A. Thomson. “The Pandora Software Development Kit for Pattern Recognition”. In: *Eur.Phys.J.* C75.9 (2015), p. 439. DOI: `10.1140/epjc/s10052-015-3659-3`.

[6] T. Suehara and T. Tanabe. “LCFIPlus: A Framework for Jet Analysis in Linear Collider Studies”. In: *Nucl. Instrum. Meth.* A808 (2016), pp. 109–116. DOI: `10.1016/j.nima.2015.11.054`.