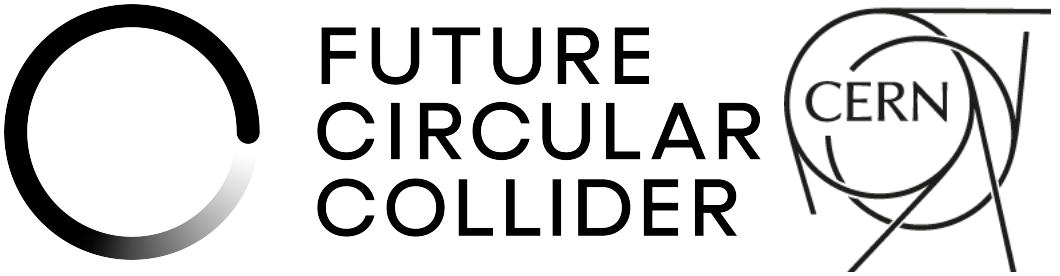
IDEA Detector

Status of the Full Simulation in Key4hep

Brieuc Francois (CERN) for the FCCSW team

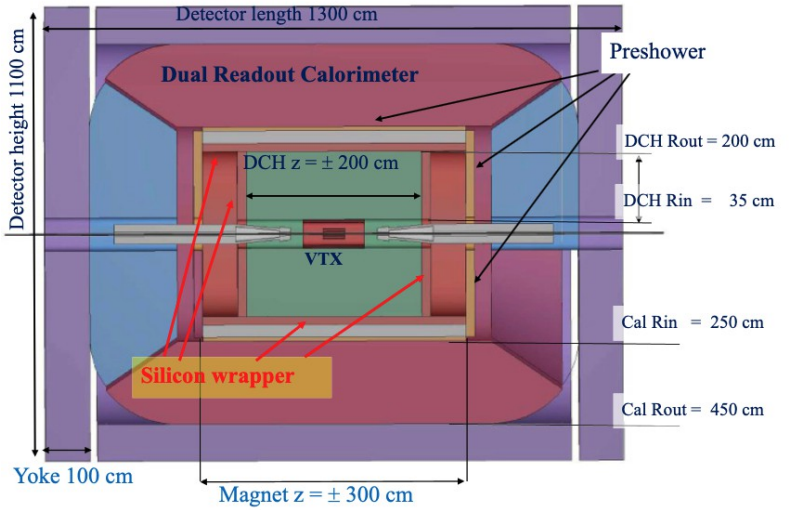
FCC General Software Meeting

May 30th, 2023



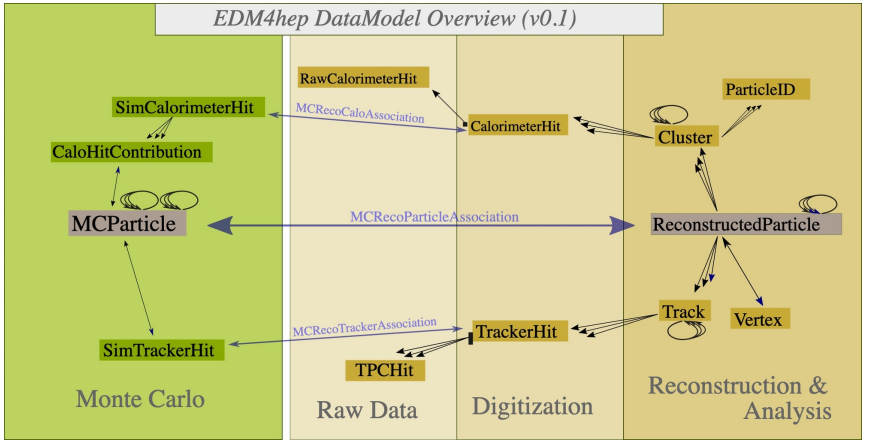
Introduction

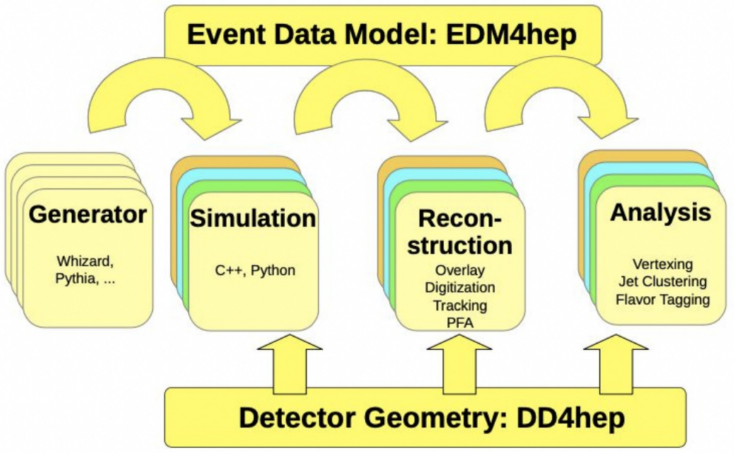
- IDEA: Innovative Detector for Electron-positron Accelerator
  - Detector concept proposed for FCC-ee (and CEPC)
  - Main features
    - Light vertex detector (DMAPS)
      - Low material budget beneficial for vertex position resolution
    - Light gaseous tracker (2 - 5 % of X_0) + silicon wrapper
      - Large volume needed due to small magnetic field (~2 T, beam emittance)
      - Good PID capabilities thanks to cluster counting
    - Ultra-thin solenoid inside calorimeter
      - Low cost, low material budget needed for particle flow performance
    - Pre-shower and dual readout calorimeter
      - 30 - 40 % / √E jet energy resolution ($H \rightarrow ZZ^{\pi} \rightarrow 4j$ and $H \rightarrow W\,W^{\pi} \rightarrow 4j$ discrimination)
    - μ-RWELL in return yoke
  - Alternative option under study: add a dual readout segmented crystal ECAL



Motivation

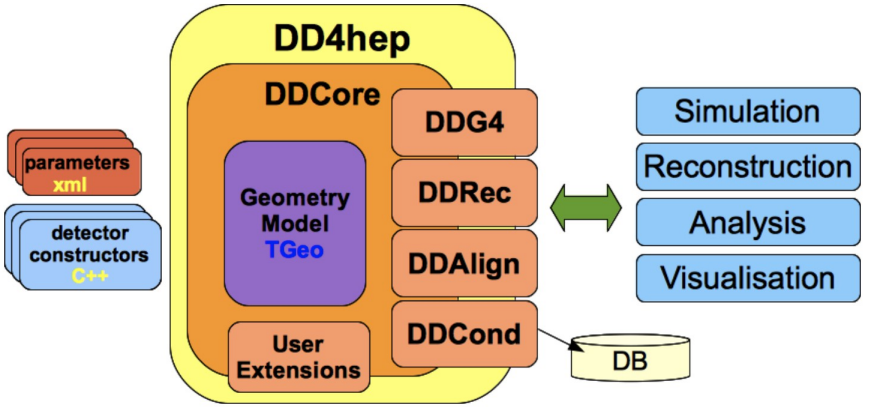
- Some of the IDEA sub-detectors were implemented by various teams in a stand-alone fashion, with heterogenous reconstruction frameworks
  - E.g. drift chamber, crystals
- In order to perform full sim studies with the whole detector concept, its sub-components have to be integrated in a common framework
  - This will in addition allow us to use IDEA sub-detectors in other detector concepts
- The future collider community agreed on using Key4hep
  - Key components already available there (e.g. detailed beam pipe description)
  - Very natural choice for the framework hosting IDEA
  - edm4hep dataformat, Gaudi orchestration, detector description based on DD4hep





DD4hep

- DD4hep: generic detector description supporting the full life cycle of the experiment
  - Conceptualization, optimization, construction and operations
- Complete description
  - Geometry, readout, alignment, calibration, …
- DD4hep uses ROOT TGeo as geometry implementation
  - Output format/interfaces: Geant4, GDML, easily extensible
- From the user perspective
  - C++ for generic geometry structure construction
  - XML configuration for detector parameters
  - Can be naively seen as an extra layer facilitating the interactions with Geant4 and extending it
  - A lot of examples available: link
  - Documentation: User's manual, doxygen

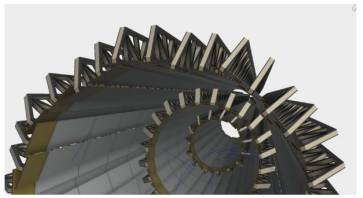


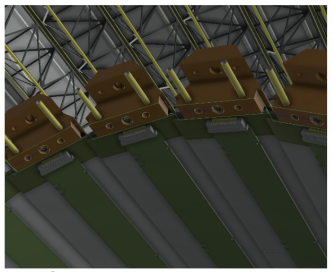
Vertex Detector

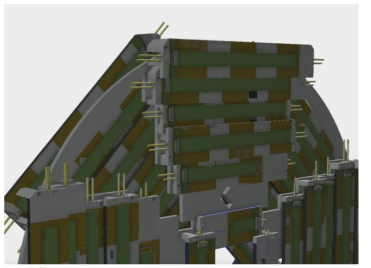
CAD Drawings
INFN

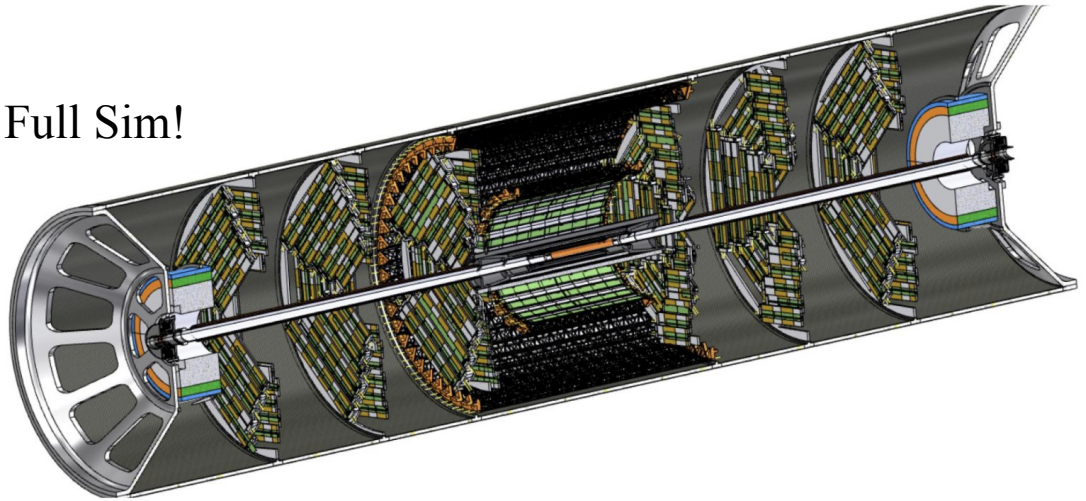
Vertex Detector

- Vertex barrel: three layers (1.37, 2.27 and 3.3 cm radius)
  - 25×25 μm^2 pixels
- Outer tracker barrel: two layers (r = 13 and 31.5 cm)
  - 150×50 μm^2 pixels
- Outer tracker endcap: three disks per side (z = 28.5, 62 and 93 cm) with 150×50 μm^2 pixels
- A detailed CAD description of supports, sensors and services is available
  - Let's get it in the Full Sim!


Vertex barrel

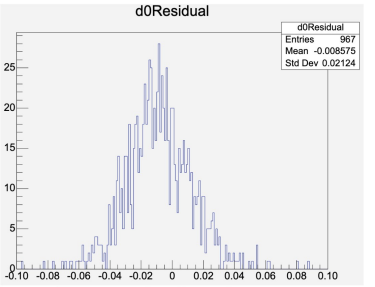

Outer tracker barrel


Outer tracker end-cap



Vertex Detector in DD4hep

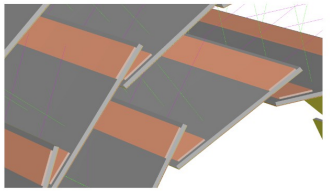
- A new detailed DD4hep description of the Vertex detector is being finalized
  - Simple sensitive plates → accurate material stack of staves, sensors and flex
    - VTX barrel support structure imported from CAD directly to DD4hep!
      - Still a few overlaps to be removed
  - WIP pull request already opened PR#273
  - First performance sanity checks performed!
    - ddsim + iLCSoft vertex reco/perf (k4MarlinWrapper)
  - Next steps: add further details (e.g. services), edm4hep native digitization
  - The silicon wrapper will be implemented based on the same detector builders
    - No engineered design available yet



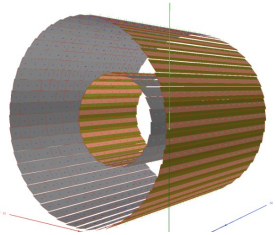
D_0 resolution in mm.

Armin Ilg

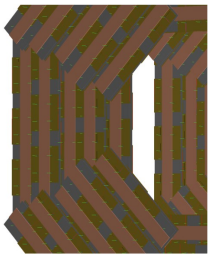
Inner barrel



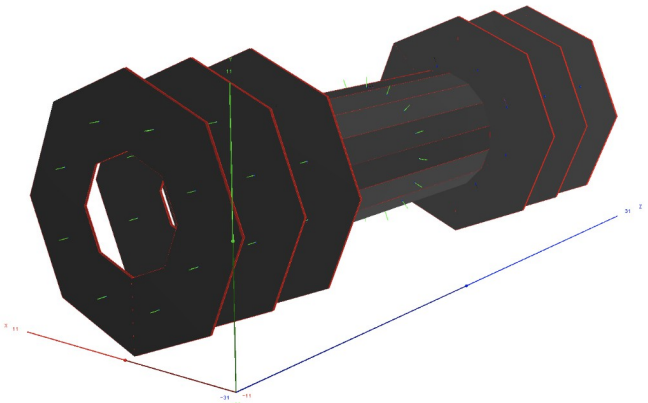
Outer barrel



Endcaps



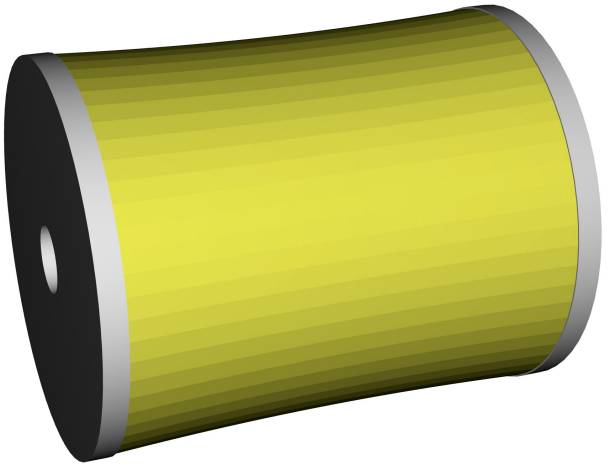
Previous version

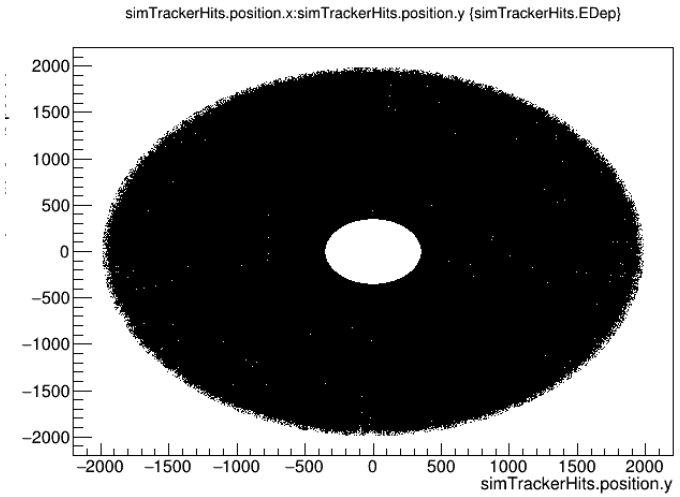


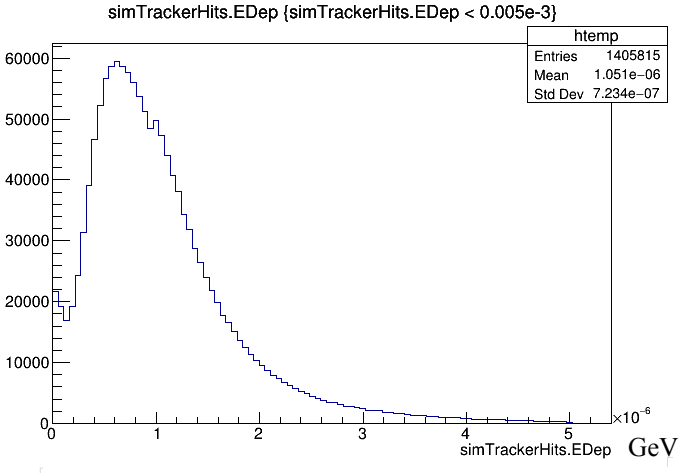
New version

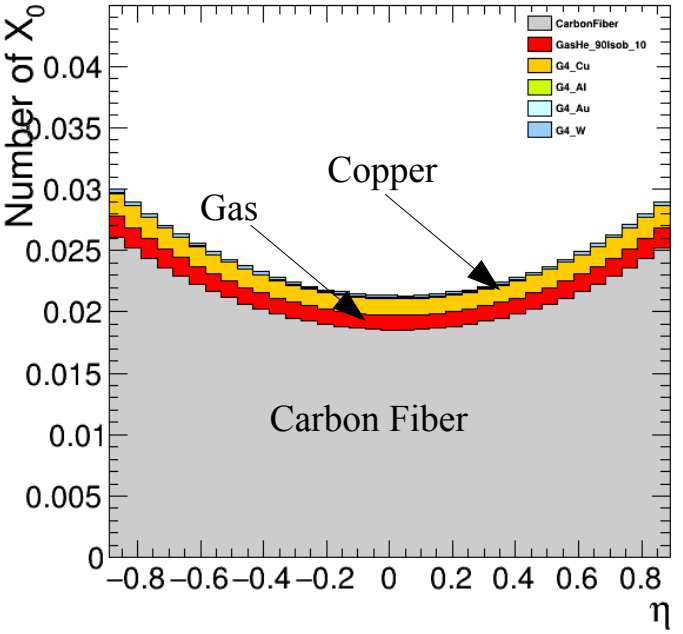
Drift Chamber in DD4hep

- IDEA DC originally implemented in plain Geant4: link
- Simplified (no wires) DD4hep version used so far: link
- New detailed DD4hep implementation available in FCCDetectors: link (Lorenzo Capriotti)
  - Carbon fibre/Cu walls, W sense wires, Al field/guard wires, Au coating, includes stereo angle, filled with GasHe_90Isob_10
  - Currently being validated and integrated in the Full Sim workflow
    - Overlap checks, material scan, diagnostic plots
  - SimHits can now be extracted! (PR to be opened soon)









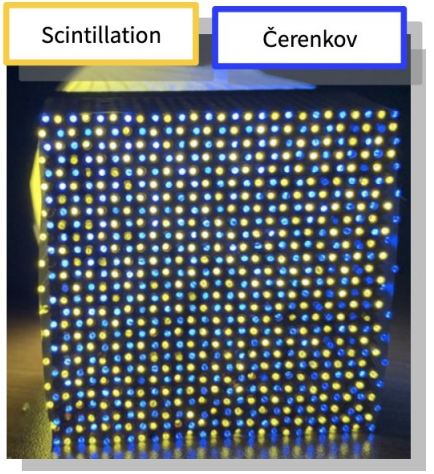
Drift Chamber Reconstruction

- Need now to implement DC reconstruction in Key4hep
  - DC segmentation (hit cellID)
  - SimHit → RecHit in edm4hep dataformat, cluster counting
- Tracking
  - One algorithm ready to be used in Key4hep: iLCSoft MarlinTracker (CLIC/CLD)
    - Full silicon oriented
  - Several solutions will be investigated
    - ACTS: need some dataformat gymnastic and a way to ship the geometry
      - Solution implemented by EIC
    - ILD approach:
      - Track segments built separately in inner, outer and forward Si-tracking + central TPC
      - Combined and refitted in a second stage
    - Genfit: already available as a Key4hep package, only for track fitting

Dual Readout Fiber Calorimeter

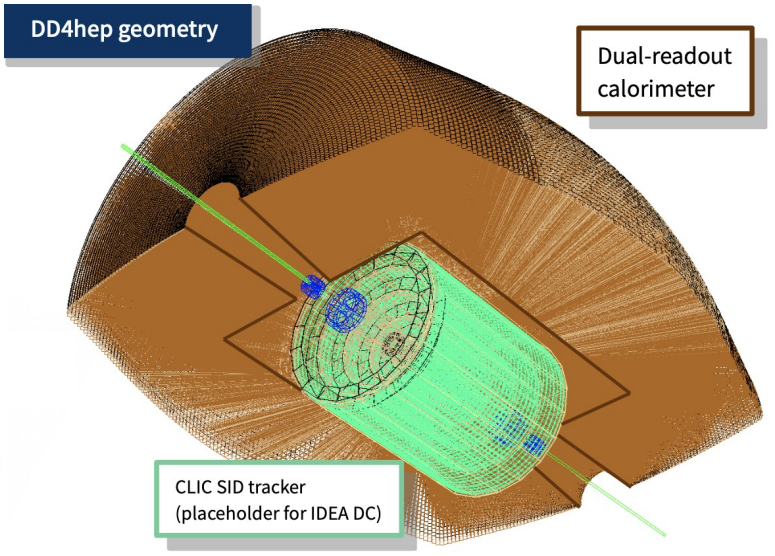
Sanghyun Ko

- Dual readout calorimeter fully available in Key4hep: HEP-FCC/dual-readout
  - Geometry, simulation, digitization, reconstruction
  - Custom segmentation (more fibers in the rear than in the front)
  - Optical physics included: link
    - Fastsim module applied to optical photons: link
  - SiPM emulation with external package: SimSiPM
- Next steps
  - Integrate geometry in the central repository
  - Further work on lowering CPU cost of simulation



K4SimGeant4 configuration

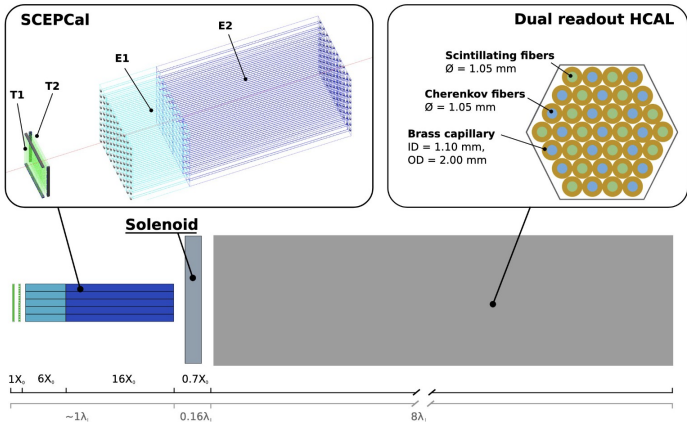
```
regionTool = SimG4FastSimOpFiberRegion("fastfiber")
opticalPhysicsTool = SimG4OpticalPhysicsList("opticalPhysics", fullphysics="SimG4FtfpBert")
physicslistTool = SimG4FastSimPhysicsList("Physics", fullphysics=opticalPhysicsTool)

from Configurables import SimG4DRcaloActions
actionTool = SimG4DRcaloActions("SimG4DRcaloActions")

# Name of the tool in GAUDI is "XX/YY" where XX is the tool class name and YY is the given name
geantservice = SimG4Svc("SimG4Svc",
  physicslist = physicslistTool,
  regions = ["SimG4FastSimOpFiberRegion/fastfiber"],
  actions = actionTool
)
```

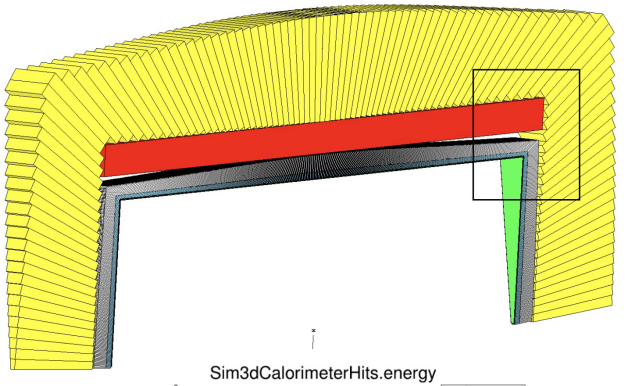


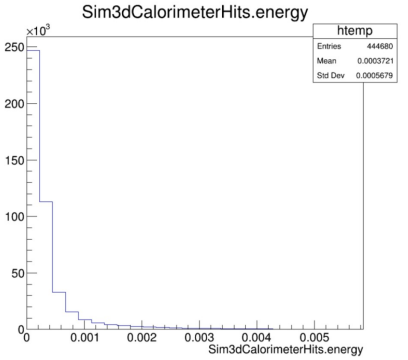
Dual Readout Crystals in Key4hep

Wonyong Chung and Marco Lucchini

- Alternative detector configuration with greatly improved **EM energy resolution**/longitudinal segmentation under study
  - Add longitudinally segmented **dual readout crystals** in front of the HCAL (before the solenoid)
  - Again, need a common framework to thoroughly assess the gain from such a configuration
- Detector description implemented in DD4hep: link
  - Great synergies with fiber dual readout!
    - Used the same Github repository as starting point
  - SimHits available
- WIP: port the code to the central dual-readout repository, digitization, reconstruction, Particle Flow (not Pandora based)







Beam Pipe

Andrea Ciarma

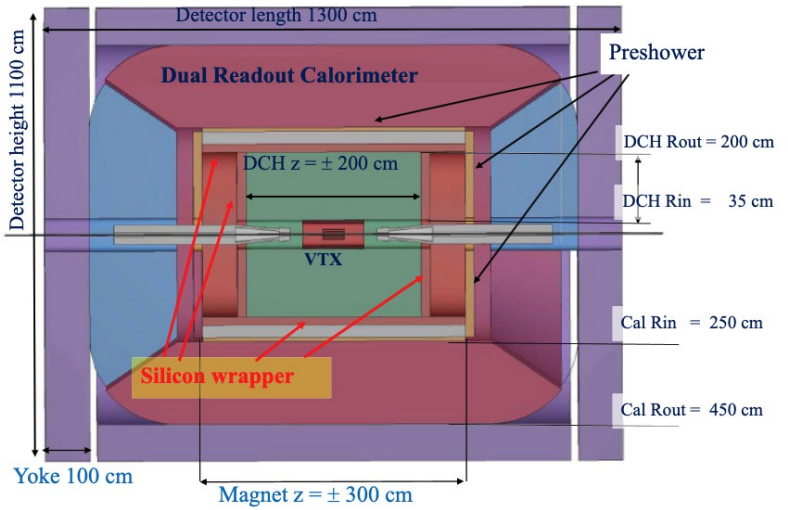
- Beam pipe is common to all detectors
- An updated version is now available in DD4hep
  - Smaller radius: starts at r = 1 cm (instead of 1.5 cm)
- Split in four files in FCCDetectors under Detector/DetFCCeeCommon/compact/
  - Beampipe_with_notch_noShield.xml
  - SRshielding.xml
  - BeamInstrumentation.xml
  - FFQuads.xml (possibility to make this part sensitive)
- LumiCal under Detector/DetFCCeeCommon/compact/LumiCal.xml
  - SiW sandwich design which is a few years old and comes from ILD
  - Some inconsistencies spotted w.r.t. current design (thanks Mogens)
    - Should be updated at some point but no manpower for that at the moment

k4Geo

- Having all sub-detector geometries in a common place will ease inter-operability through the 'plug-and-play' approach
  - The FCC detector geometries are being moved from FCCDetectors to k4geo
    - The target is to host all the future collider detector geometries!
    - Policy (already followed by CLD)
      - Facility agnostic detector builders (C++) separated from specific implementations (xml)
        - Both documented, separately
      - Different detector options will be able to co-exists + versioning possible for book-keeping and backward compatibility (my_detector_oX_vY)
      - Full detector configuration folder has to be self-consistent to ease grid job submission
        - Requires some discipline, e.g. Vertex_o4_v05.xml must correspond to the exact same detector across folders
  - FCCDetectors will not be removed, it will be frozen and kept for legacy
- k4geo could also potentially host test-beam geometries
  - A flexible enough detector builder (C++) should allow us to easily write the xml for a small module

Next Steps

- The main missing player to build a full IDEA detector in Key4hep is the muon system
  - μ-RWELL implementation will start soon (4 months trainee starting in August)
- Next steps
  - Build a full IDEA detector DD4hep implementation based on all sub-detectors described earlier
    - Adapt detector dimensions consistently (Patrizia is collecting the needed information)
    - Implement a steering file orchestrating the simulation of all these detectors
      - E.g. dual readout technologies need optical physics
  - Implement sub-detector reconstruction (for those not having it already)
  - Start building physics objects!



Organization

- Building now a **collaboration** of people working on FCC detector full sim in Key4hep
- Advertisement: **bi-weekly working meeting on FCC detector Full Sim just started!**
  - Mondays at 11 am CEST (try to keep the meeting short, within ~ 1 hour)
    - Flexible on the frequency and time (can e.g. consider some afternoon slots for people on the other side of the Atlantic)
  - Indico page
  - Subscribe to the **FCC-PED-SoftwareAndComputing-Full-Simulation** CERN e-group to receive the announcements
  - Really a working meeting: round table to discuss ongoing developments, technical issues faced, profit from each other expertise, … Don't be shy to show unpolished plots!
  - HEP detectors can be complicated and reliably simulating them requires a deep understanding of their various components
    - Will also try to bring some detector physics expertise there when needed

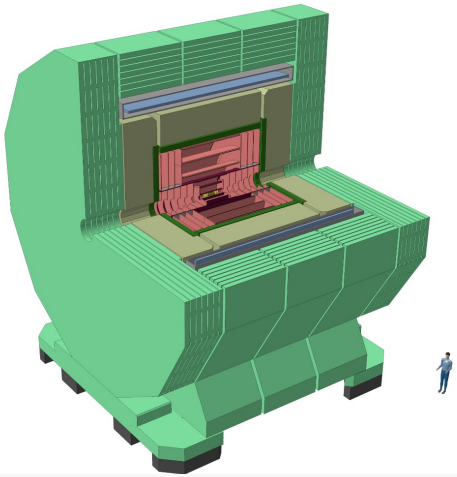
Outlook

- Many IDEA sub-detectors start being available in DD4hep/Key4hep
  - Various stages of development (Sim, Digi, Reco) and validation
- Started collecting all detector descriptions into a common GitHub repository
- FCC Detector Full Sim Meeting being started
- What is ahead of us
  - Validate and mature detector geometries and reconstruction
  - Start playing with physics objects (and Particle Flow)
  - Not a linear process: some bugs/limitations will only reveal themselves when producing 'high-level' physics objects
    - The feedback loop is important
- We must keep the team active and interacting
  - There won't be a time where we will say “we are done”
    - Operating detector are still working on their full sim, not only for Phase II

Additional material

CLD Full Sim Status

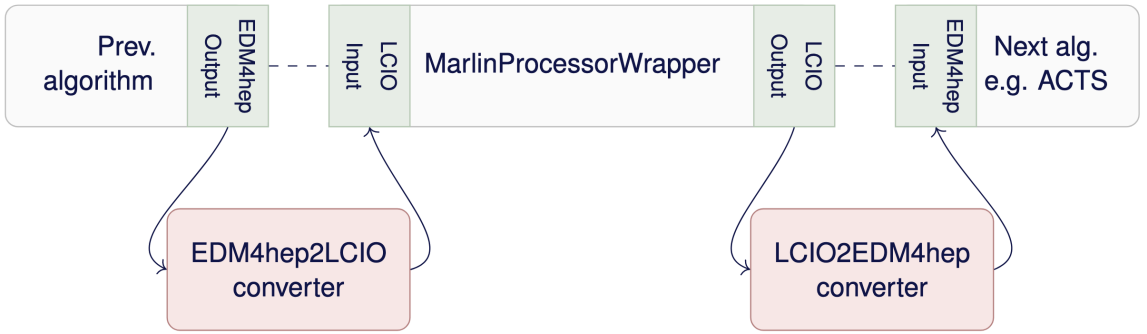
- **All CLD sub-detectors implemented in DD4hep**
  - Several configurations envisaged
- Full simulation + reconstruction workflow available!
  - Simulation through *ddsim*
  - Reconstruction through *Marlin*
    - Background overlay, digitization, conformalTracking, ParticleFlow (PandoraPFA), vertexing and flavor tagging
    - Inherited from ILD/CLICdet



```
ddsim --compactFile FCCee_o1_v05/FCCee_o1_v05.xml \
      --enableGun \
      --gun.distribution uniform \
      --gun.energy "10*GeV" \
      --gun.particle mu- \
      --numberOfEvents 100 \
      --outputFile Step2_edm4hep.root
```

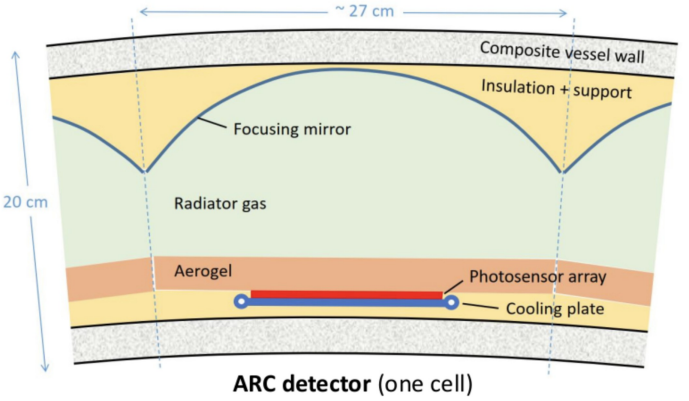
Link to tutorial

- *Marlin* reconstruction based on LCIO data format but can be **integrated in EDM4hep Gaudi based workflows** through the *MarlinWrappers* + data format translation
  - Example of steering file

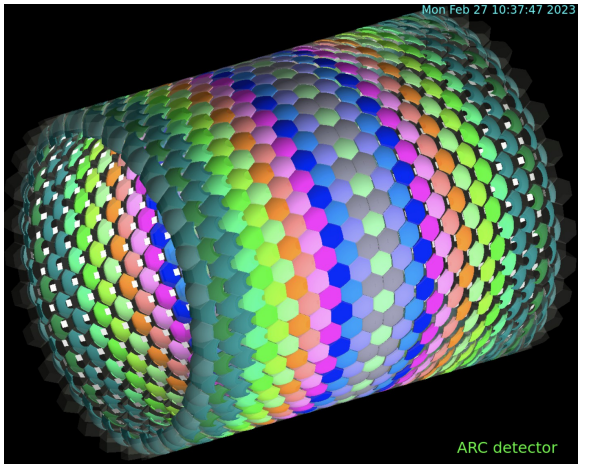


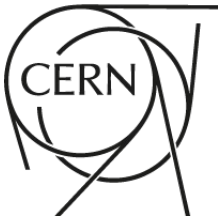
PID Detectors

- Detector layouts are not frozen!
  - Exploring further sub-detector technologies
- Particle ID detectors can complement/replace dE/dx or dN/dx
  - Technology more mature then at the LEP time (DELPHI)
    - LHCb RICH
- Accurate and comprehensive estimation of what it brings needs full sim
  - Photon yield/collection, additional material budget
  - Quite difficult to implement
- Array of RICH Cells (ARC) implemented in DD4hep
- Readout and reconstruction will start soon



ARC detector (one cell)




CERN