Key4HEP: SW Framework for Future Accelerator Experiments

In the context of FCC Calorimetry

FCC Noble Liquid Calorimetry Meeting

May 05, 2020

Valentin Volkl
CERN

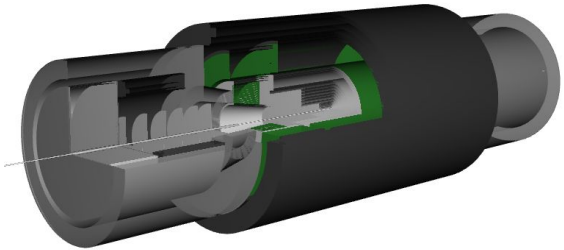
Outline

- Current Status of FCC Software
- Introduction to Key4HEP / EDM4HEP
- Overview of iLCSoft Reconstruction
- Other Plans

Recap of FCCSW / Calorimetry Related Components

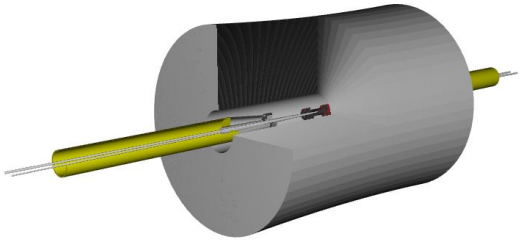
- Integrations for Fast and Full Simulations
- Geant4:
- Versatile Generation / ParticleGun Setups
- Background Overlay Handling
- Batch and Storage Infrastructure
- No Full Reconstruction
  - Selected components for focused studies
- Sliding Window Reconstruction
- Topo-Clustering
- Track Seeding
- Unmaintained parts of the software

- **FCC-hh Baseline**



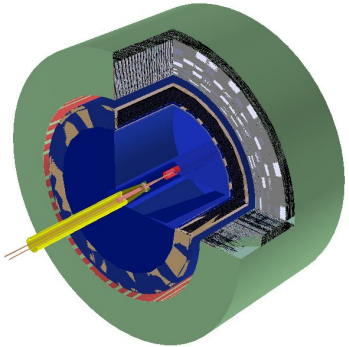
- Barrel, Endcap, Forward
- Beampipe
- Magnet Solenoid
- Shielding
- Silicon Tracker
- LAr-ECal
- Tile H-Cal
- Muon System

- **FCC-ee IDEA**



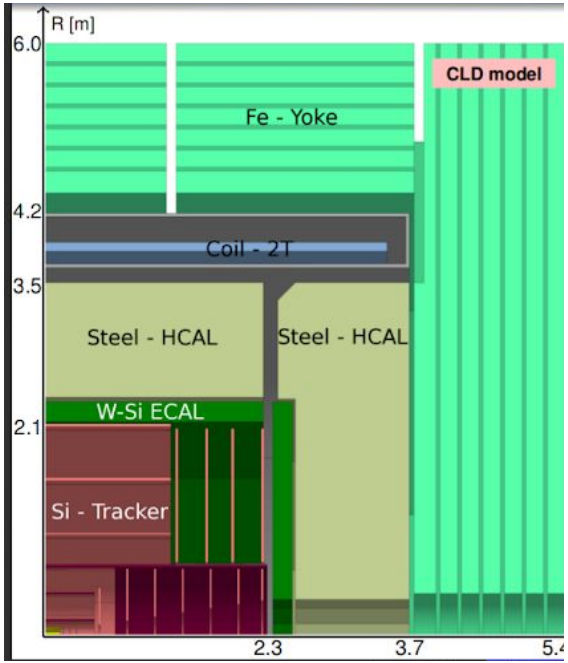
- Beampipe
- Beam Instrumentation
- LumiCal
- HOMAbsorber
- Vertex Detector
- Driftchamber
- Dual Readout Calorimeter
- Muon System

- **FCC-ee IDEA - LAr**



- Beampipe
- Beam Instrumentation
- LumiCal
- HOMAbsorber
- Vertex Detector
- Driftchamber
- Liquid Argon Calorimeter
- Muon System

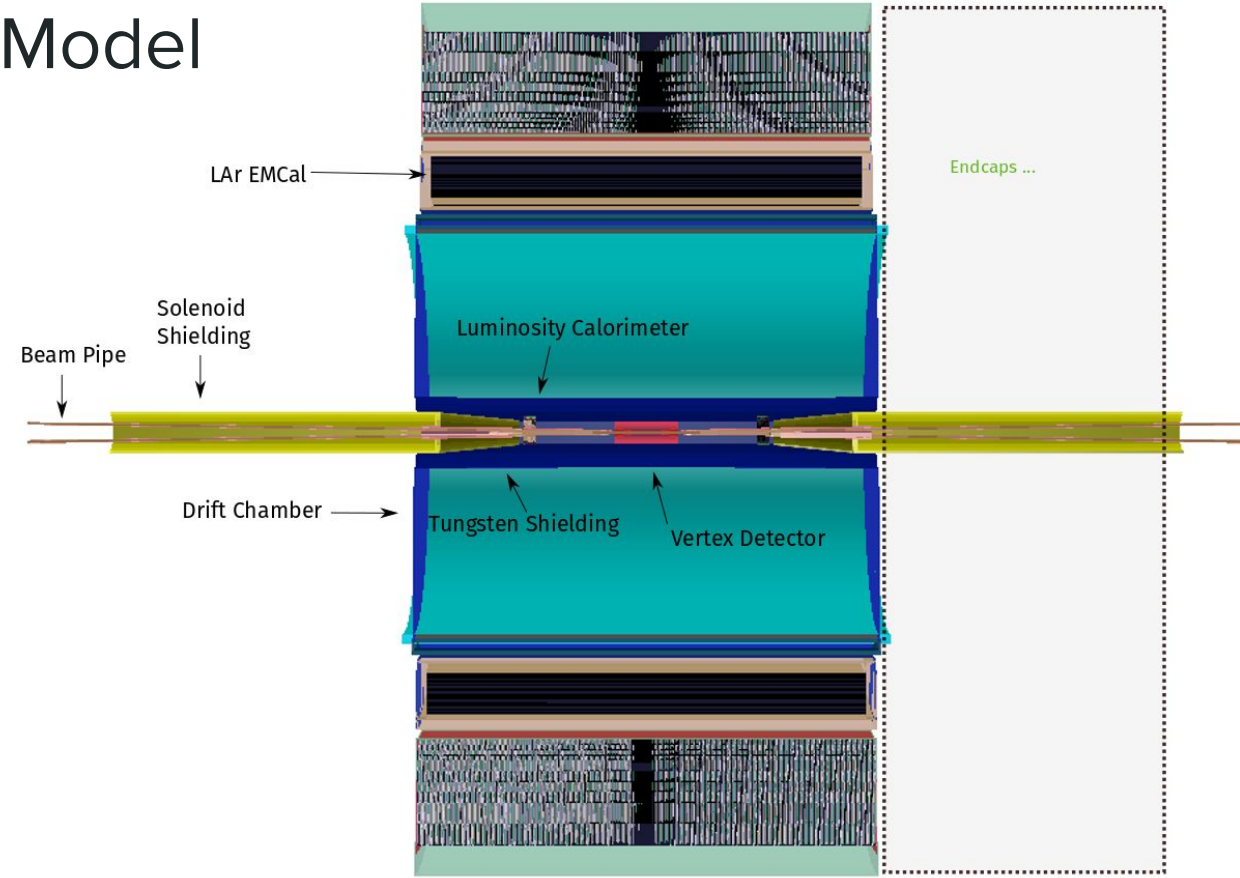
- **FCC-ee CLD**



- Not in FCCSW!
  https://github.com/iLCSoft/lcgeo

DD4hep Detector Model

- First performance results at FCC week 2019



FCCSW Recap: Beam Backgrounds

- Yorgos made an extensive effort to document his setup for Beam Background simulations
- Already used for studies of the Driftchamber (http://cds.cern.ch/record/2670936/) :
- Created a new repository for GUINEA-PIG in collaboration with CLICSW
  - https://gitlab.cern.ch/clic-software/guinea-pig
- Not clear if the calorimeter is affected, but should be studied.

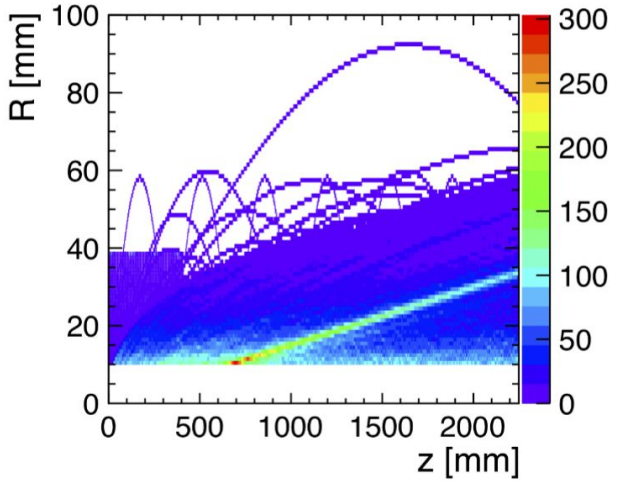
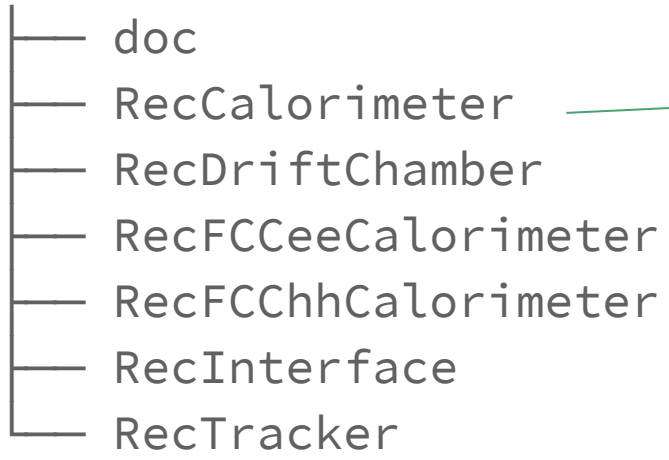


Figure 7: The trajectory of the e^+e^- pairs in a 2 T magnetic field.

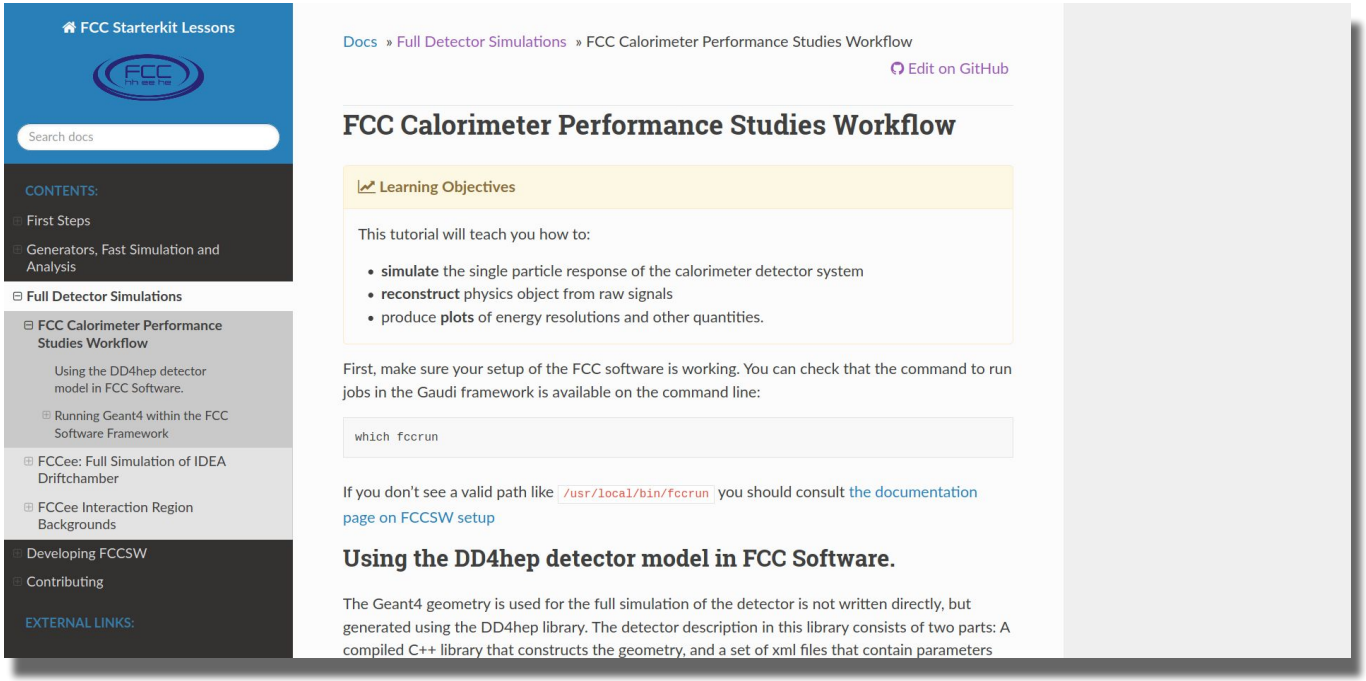
Organization of Reco Code



CalibrateCaloHitsTool
CalibrateInLayersTool
CaloTopoCluster
CaloTopoClusterInputTool
CaloTowerTool
ConeSelection
ConstNoiseTool
CorrectECalBarrelSliWinCluster
CreateCaloCells
CreateCaloClusters
CreateCaloClustersSlidingWindow
CreateEmptyCaloCellsCollection
HepMCJetClustering
JetClustering
JetHistograms
LayeredCaloTowerTool
LayerPhiEtaCaloTool
MassInv
NestedVolumesCaloTool
NoiseCaloCellsFlatTool
NoiseCaloCellsFromFileTool
PreparePileup
ReadNoiseFromFileTool
...

Documentation: https://hep-fcc.github.io/fcc-tutorials

- To ease barrier of entry, a detailed workflow is documented
- Full simulation, Sliding Window Reconstruction, Fit



TODO’s / Wishlist for FCCSW

- More complete workflows / scripts in FCCSW
- Human readable documents for detector model with target performance
- More issues and forum posts instead of private communication
- Review of existing code and data

The Key4HEP Project

- Future detector studies critically rely on **well-maintained software stacks** to model detector concepts and to understand a detector's limitations and physics reach
- Aim at a low-maintenance common stack for FCC, ILC/CLIC, CEPC with ready to use "plug-ins" to develop detector concepts
- Reached consensus among all communities for future colliders to develop a **common turnkey software stack** at recent Future Collider Software Workshop
- Identified as an important project in the CERN EP R&D initiative
- Regular meetings
  - https://indico.cern.ch/category/11461/
- Docpages
  - https://cern.ch/key4hep (main documentation site)
  - https://cern.ch/edm4hep (doxygen code reference)

Spack for Key4HEP

- Spack is a package manager
  - Does not replace CMake, Autotools, ...
  - Comparable to apt, yum, homebrew, ...
    - But not tied to operating system
    - And no central repository for binaries!
- Originally written for/by HPC community
  - Emphasis on dealing with **multiple configurations** of the same packages
    - Different versions, compilers, external library versions ...
    - ... may coexist on the same system
  - `Spec:` Syntax to describe package version configuration and dependencies
- Repository added with Key4HEP package recipes

```
git clone https://github.com/spack/spack.git
git clone https://github.com/key4hep/k4-spack.git
alias spack='python $PWD/spack/bin/spack'
spack repo add k4-spack
# install the meta-package for the key4hep-stack
spack install key4hep-stack
```

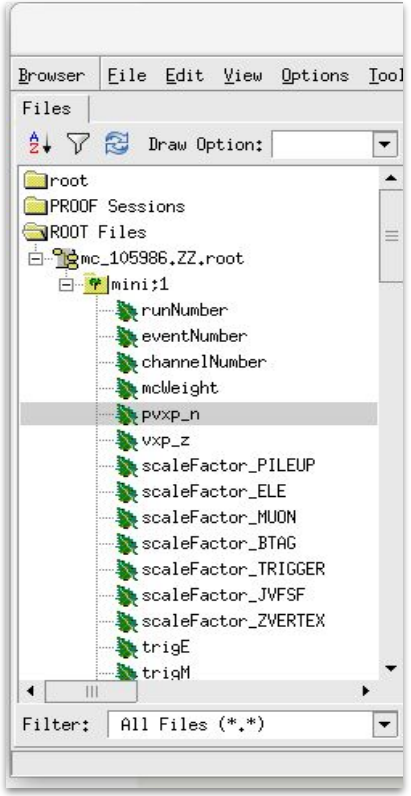
Deployment on CVMFS

```
/cvmfs/sw.hsf.org/key4hep/
|-- releases/ $LCG_version / $platform / $pkgname-$spackhash / (bin  … )
|-- views   / $K4_version  / $platform / (bin include share … init.sh)
|-- setup.sh
|-- contrib

/cvmfs/sw-nightlies.hsf.org/key4hep/
|-- nightlies/ $timestamp / $platform / $pkgname-$spackhash / (bin  … )
|-- views   / $timestamp  / $platform / (bin include share … init.sh)
|-- setup.sh
|-- contrib
```

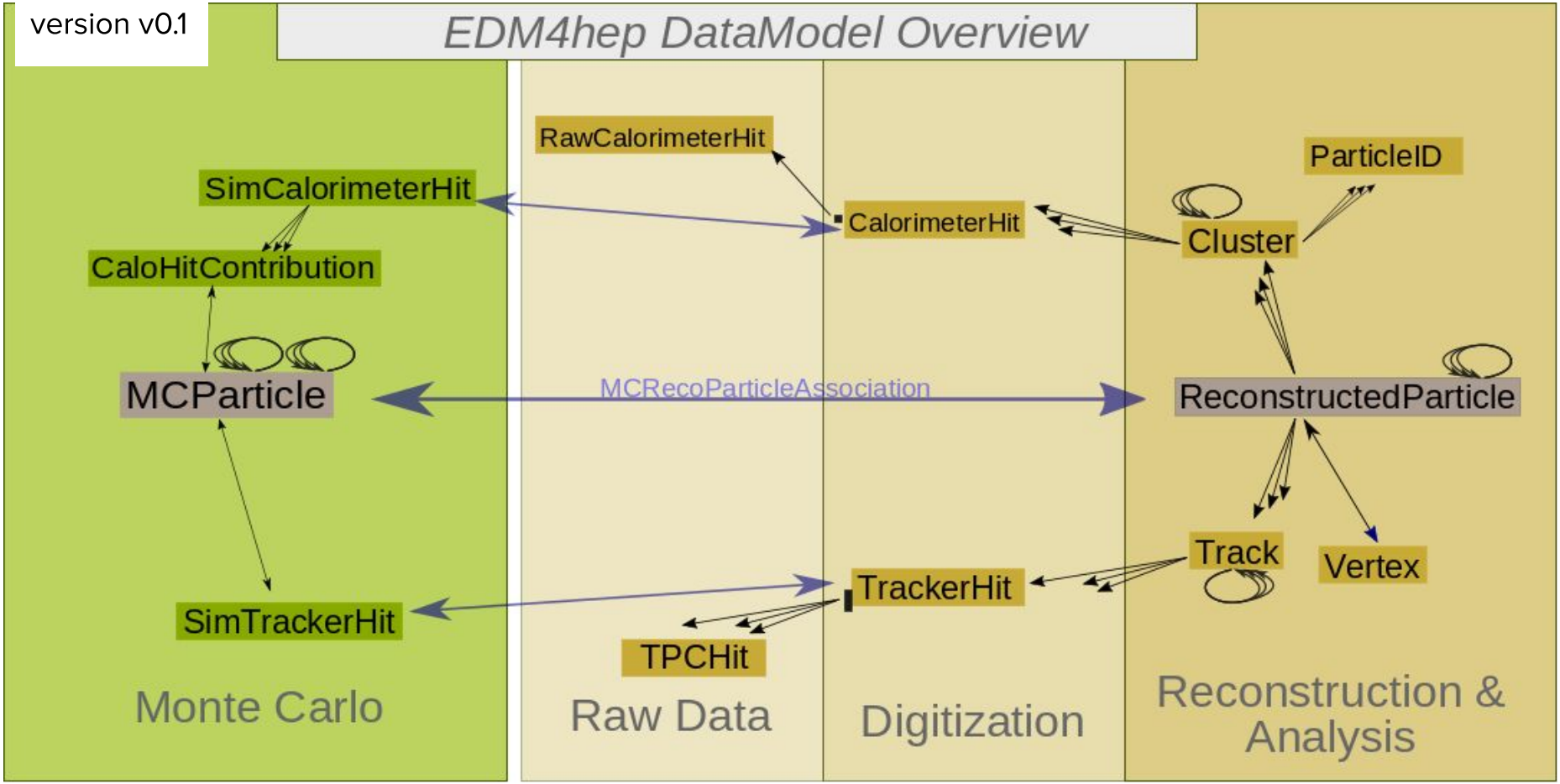
EDM4HEP - Introduction

- Event Data Model:
  - Describes structure of HEP Data:
    - definitions of **objects** and how they are **grouped**
    - **technical** implementation of persistency and processing
- Can be as simple as “Branch names in ROOT file”
  - But more sophisticated solutions can:
    - provide an **application programming interface** for HEP software
    - aid developers in writing more **efficient code**
    - enable **collaboration**



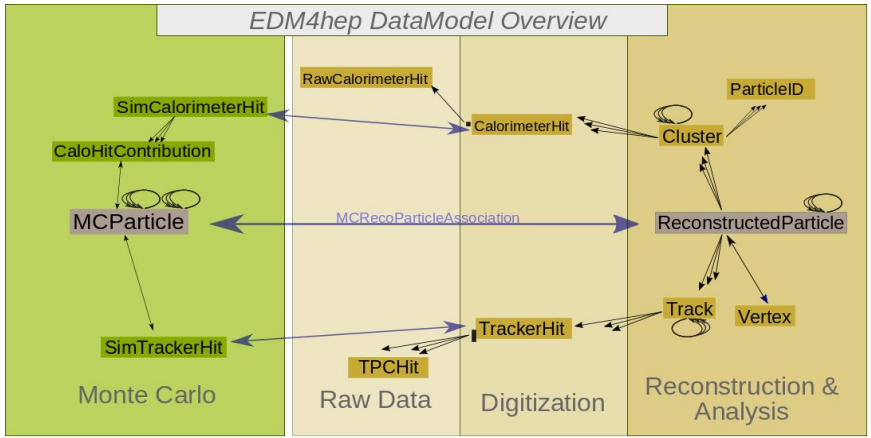
BETA
version v0.1

Code Reference under https://cern.ch/edm4hep



Plans for Transitioning

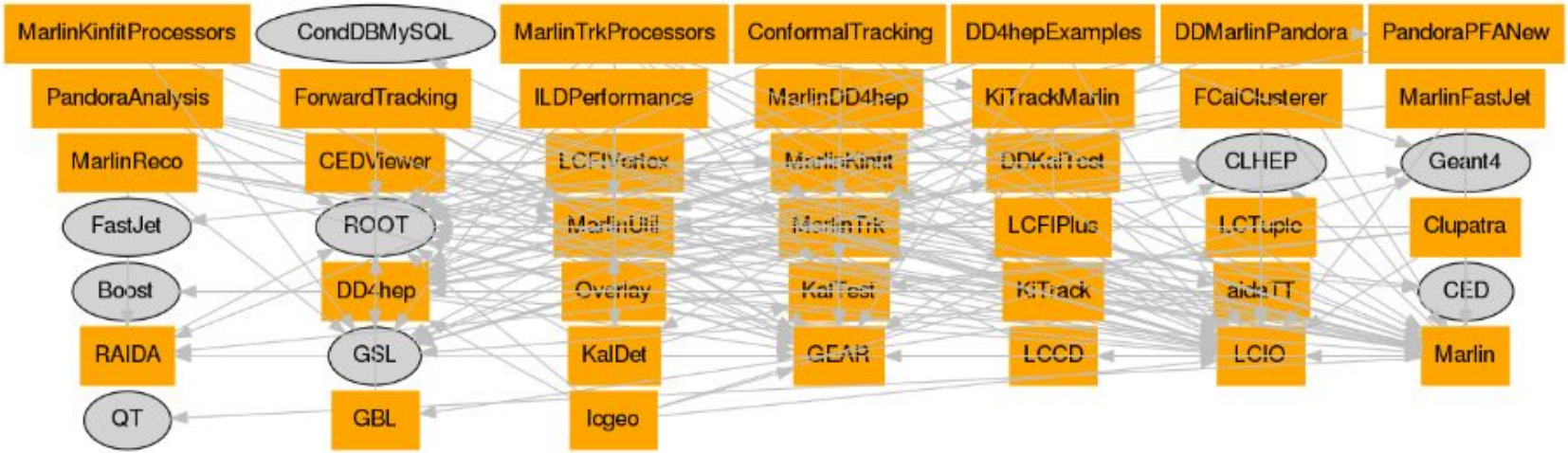
- With next release:
- Write Converters to do step-wise transition
  - (no conversions of existing data planned)
- Start with Delphes Infrastructure
- Opportunity to review code.
- Start basing FCCSW on Key4HEP Core Components.
- Check to see if common repository (lcgeo) used for Detector Models



iLCSoft Overview

From F. Gaede's Talk at the Future Collider Software Workshop (Bologna 2019)

iLCSoft packages



external packages: ROOT, Geant4, CLHEP, GSL, QT, Boost, FastJet

- iLCsoft provides the complete software chain for full simulation, reconstruction and analysis chain for lepton colliders

Reconstruction and Analysis Tools in iLCSoft

- realistic Digitizers
  - Tracker: smearing of tracker hits w/ established point resolutions
  - Calorimeter: 'calibrations' of cell energies, cross talks, light non-uniformity, ...
- Tracking Toolkit: MarlinTrk/aidaTT
  - generic track fitting and interface for pattern recognition
  - a variety of pattern recognition algorithms
- Particle Flow Algorithm: PandoraPFA
  - reconstruction of individual particles
- PID Tools
  - dE/dx, shower shapes, TOF,...
- Jet Clustering Tools
- Flavor Tagging (LCFI)
  - MVA tools
- Vertexing Tools
  - zvtop
- Monte Carlo-Truth Tools
  - Track and Cluster cheaters
  - Jet-Clustering
  - detailed MCTruth-Reconstruction Links
- ...

most of the tools are detector agnostic or at least easily adaptable to different detectors

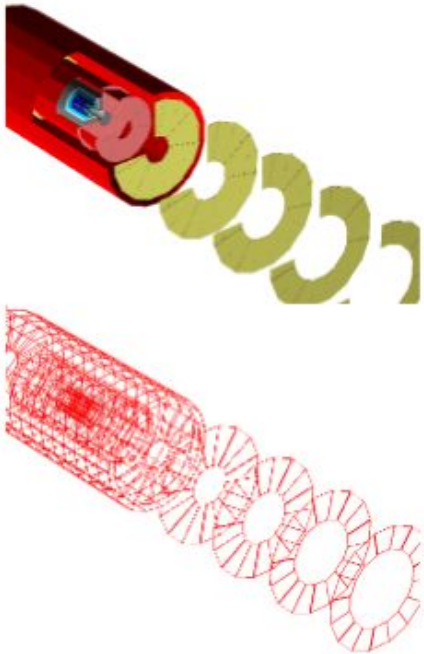
some examples on next slides ...

DDRec: dedicated tracking surfaces

- tracking needs special interface to geometry
- measurement and dead material surfaces (planar, cylindrical, conical)
- surfaces attached to volumes in detailed **DD4hep** geometry model

surfaces:

- u,v, origin and normal
- inner and outer thicknesses and material properties
- local to global and global to local coordinate transforms:
  - $(x, y, z) \leftrightarrow (u, v)$

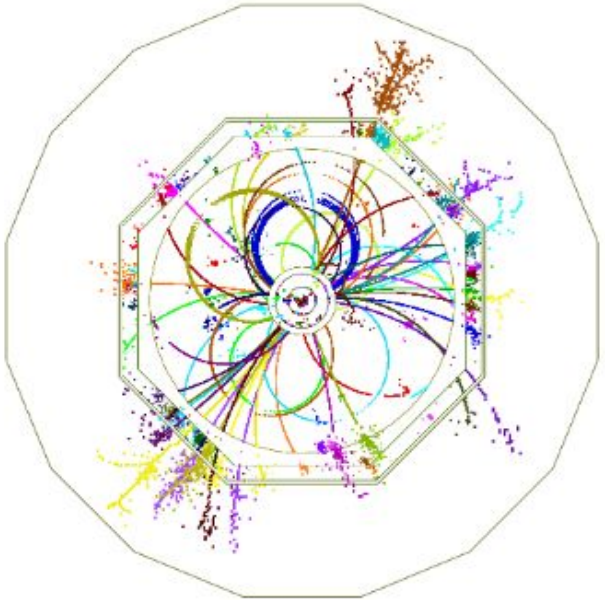


Particle Flow reconstruction at linear colliders

- linear collider detectors are optimized for PFA:
  - high granular calorimeters
  - high hermiticity
  - excellent tracking efficiency

PFA

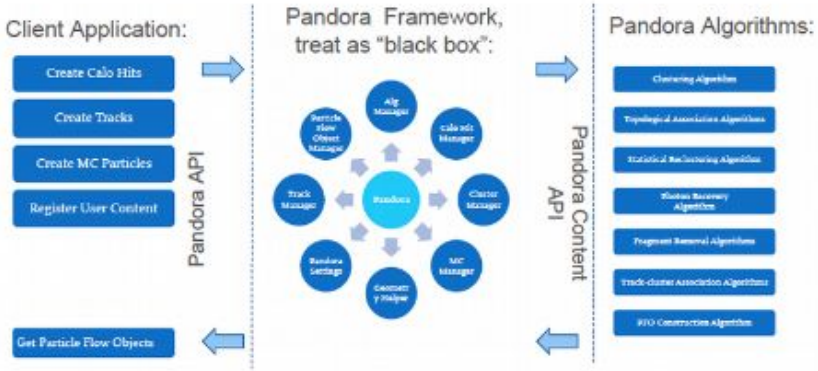
- reconstruct every single particle
- use tracks for all charged particles
- use Ecal for photons
- use Hcal for neutral hadrons



ttbar @ 500 GeV in ILD

PandoraPFA

- generic framework for pattern recognition in calorimeters
  - originally developed for ILC and CLIC (U.Cambridge)
  - glue code for Marlin: **DDMarlinPandora**
- state of the art particle flow algorithm for highly granular calorimeters
- **AIDA2020** project
  - application to LAr-TPC at neutrino experiments
  - application to HL-LHC



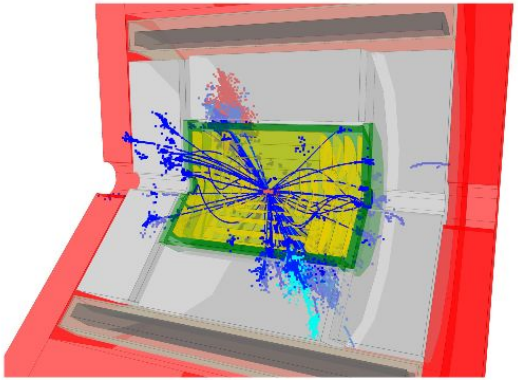
J.Marshall

the PandoraSDK provides a nice blue print for how to handle cross-experiment software tools and algorithms

CLIC Reco Evolution: Adiabatic Changes

[Talk by A. Sailer at CHEP 2019]

- Full CLIC reconstruction implemented in iLCSoft
- While transitioning to KEY4HEP, need to be able to keep running the CLIC reconstruction
- Switch components one by one, validate changes
  - Geometry provided by DD4HEP, no changes needed
  - Move framework from Marlin to Gaudi: wrap existing processors
  - Move from LCIO to EDM4HEP
  - Replace wrapped processors with native Gaudi algorithms
- Incidentally will make iLCSoft functionality available to other users of the stack



CHEP'19, Nov. 5, 2019 A. Sailer et al.: Towards a Turnkey Software Stack for HEP Experiments 11/16

ACTS

- Talk by A. Salzburger and M. Kiehn at [last EP R&D Discussion]
- IHEP interested in contributing Key4HEP Framework integration

A testbed Open Data Detector

DD4Hep based tracking detector

- Supports Geant4 simulation and Acts Fatras simulation

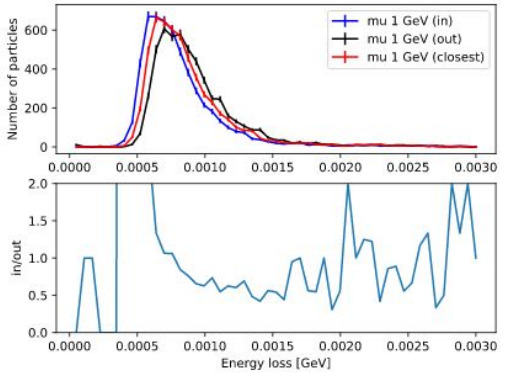
  Adaption of TrackML detector

  Fully integrated into acts examples

  - supports TrackML data format (csv writer infrastructure)

  Current ongoing activity to fully validate detector and fast sim (M. Kiehn & S. Sevova)

Release should be in O(1 months),
Interest to wrap e.g. FCC-hh calorimeter
And a simplified muon system around



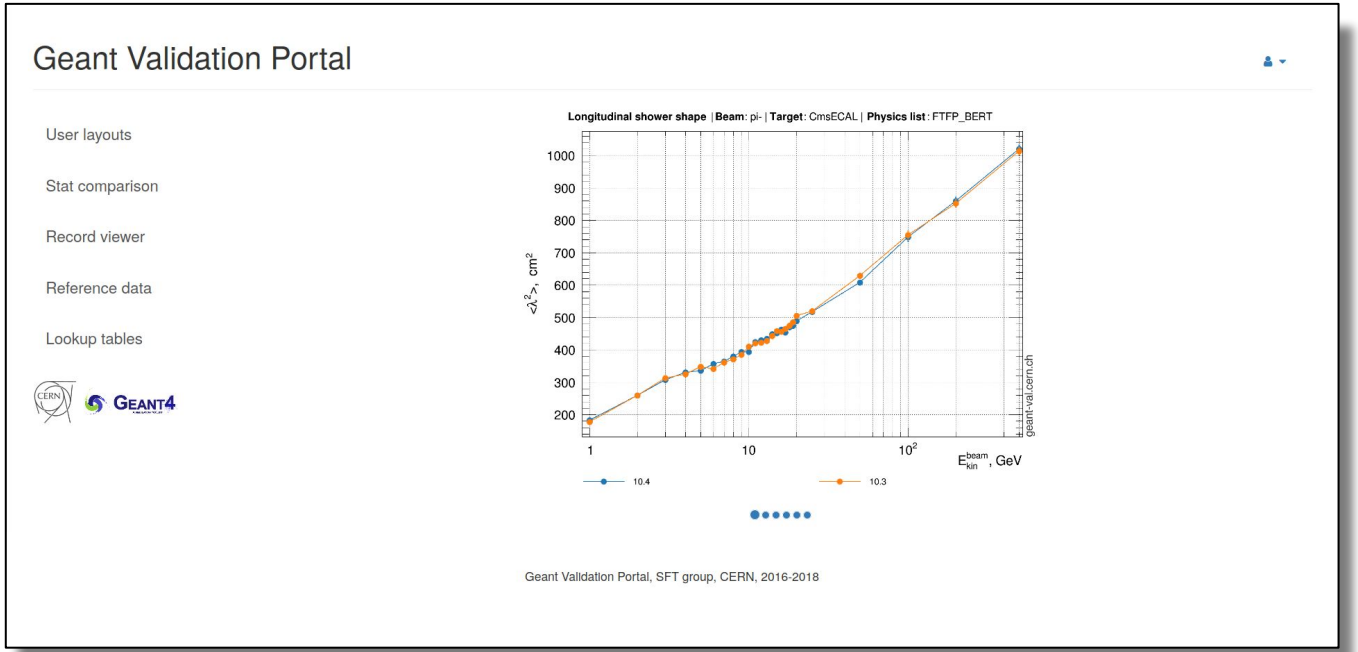
29

Performance Plots and Comparison

Need for common package to evaluate detector performance

[iLCSoft/ILDPerformance: Package to evalute the Performance of the ILD detector simulation]

Investigate tooling of [geant-val.cern.ch]



Conclusions

- Currently transitioning the current Software, FCCSW, to a common framework and datamodel (Key4HEP / EDM4HEP)
- With Key4HEP there is a roadmap for to complete Reconstruction with ParticleFlow
- Initial effort on Build infrastructure and Datamodel mostly complete
- Ready to start work on Detector Models and Full Simulation

Organization in FCCSW/Detector

```
DetCommon
├── compact
│   ├── Box.xml
│   ├── EcalAir.xml
│   ├── elements.xml
│   ├── HcalAir.xml
│   ├── materials.xml
│   ├── MuonSystemAir.xml
│   └── TrackerAir.xml
```

```
DetFCChhBaseline1/
├── CMakeLists.txt
└── compact
    ├── FCChh_BeamTube.xml
    ├── FCChh_DectDimensions.xml
    ├── FCChh_DectEmptyMaster.xml
    ├── FCChh_DectMasterMaterial.xml
    ├── FCChh_DectMaster.xml
    ├── FCChh_ECalAir.xml
    ├── FCChh_MuonAir.xml
    ├── FCChh_Shielding.xml
    ├── FCChh_Solenoids.xml
    ├── FCChh_TrackerAir.xml
    └── FCChh_Visualisation.xml
```