Software for detector concept developments

3rd FCC-France / Higgs & ElectroWeak Factory Workshop, Annecy

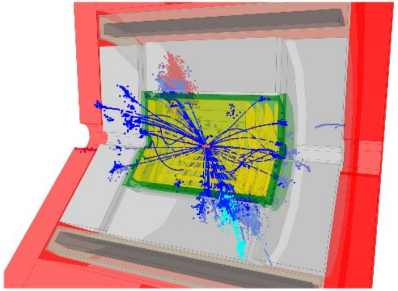
Nov 30, 2021

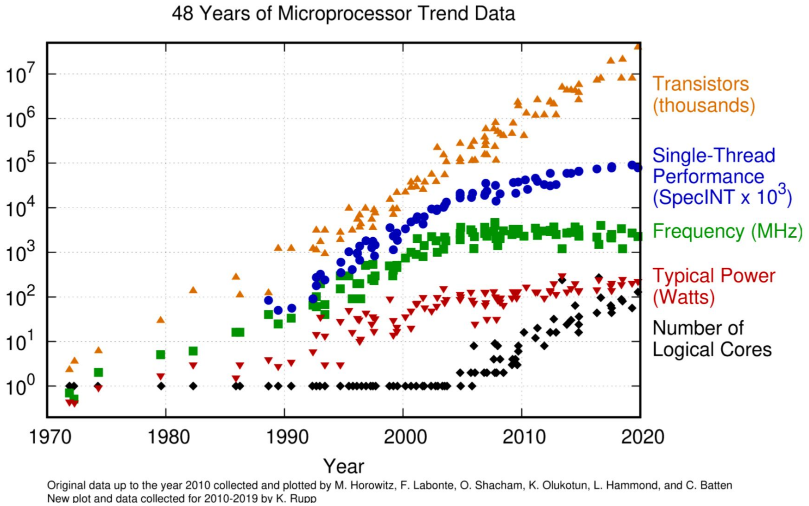
Valentin Volkl, Clement Helsens, Gerardo Ganis
CERN

This work benefited from support by the CERN Strategic R&D Programme on Technologies for Future Experiments (https://cds.cern.ch/record/2649646/, CERN-OPEN-2018-006).

Software Challenges for (Future) Detector Developments

- Complex workflows
  - MC Simulations: Event Generation, Particle Propagation; Backgrounds, Digitization, Reconstruction, Analyses ...
- Performance
  - Need distributed computing infrastructure
  - And parallel programming to use evolving hardware efficiently
- Advantage: No “real-world” problems like Alignment and Conditions
  - ... but need to design for it!





Recap of FCCSW (Full-Sim Related Components, up to v0.16)

- 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

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)

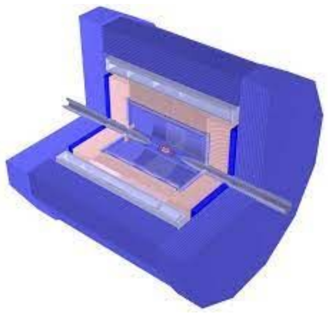
Status of Experiment Software

key4hep-stack/2021-10-29 comprises

...in a consistent stack!

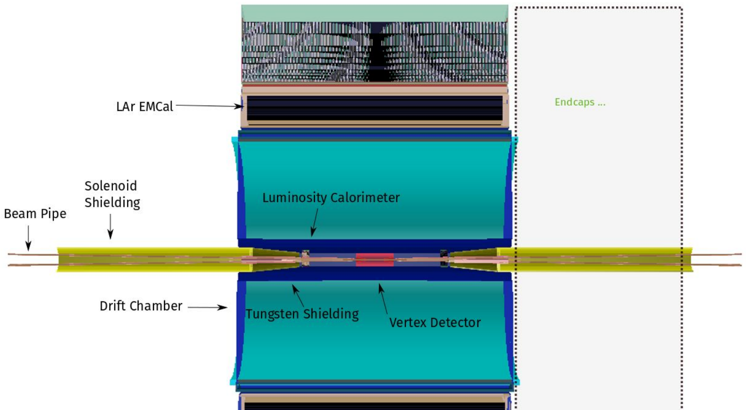
CEPCSW

v0.2.2



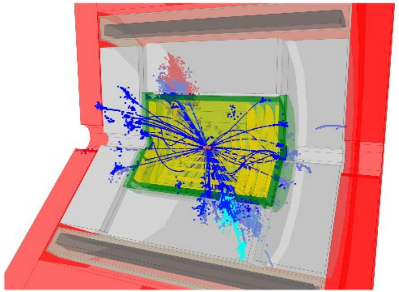
FCCSW

v1.0pre06

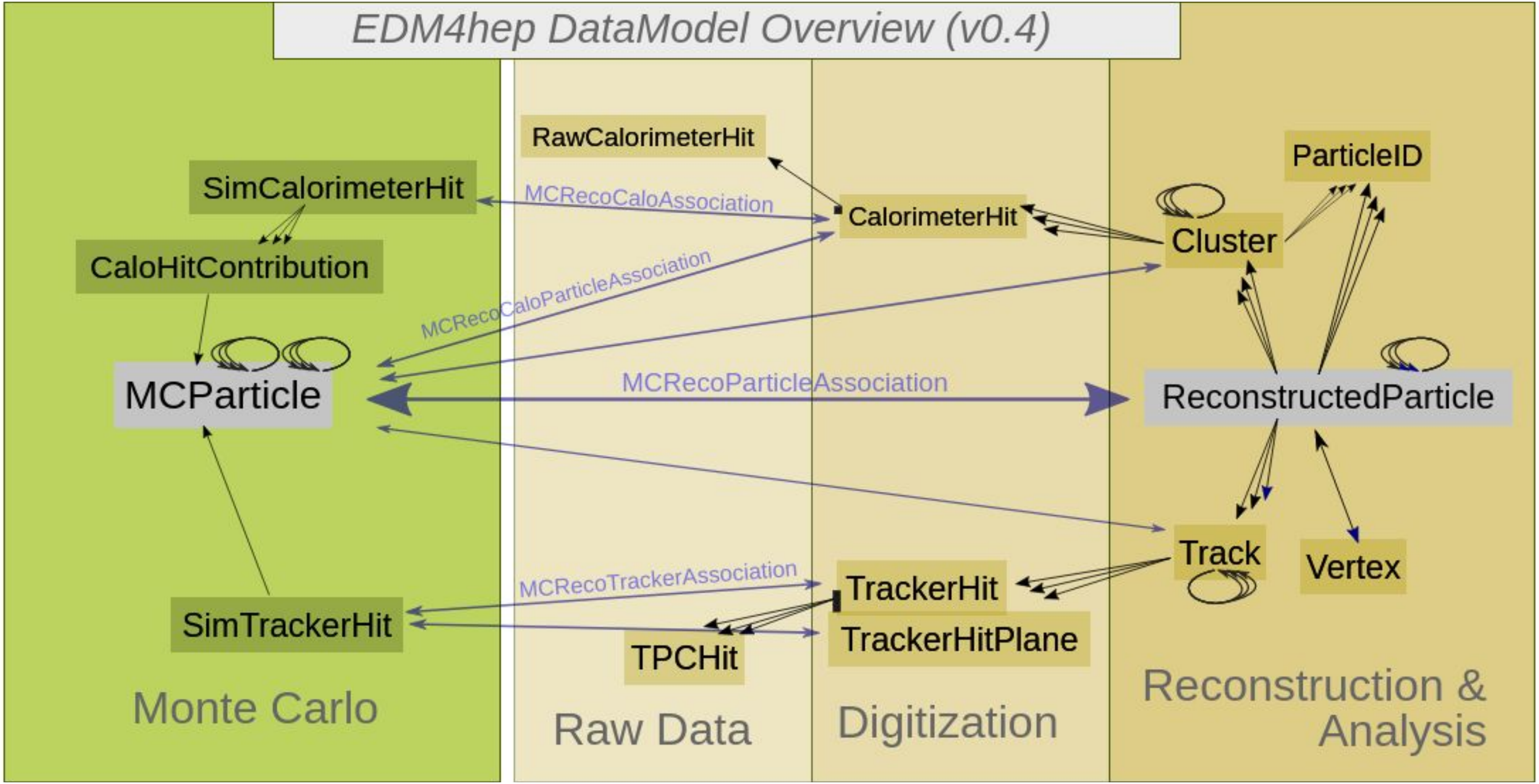


CLIC/ILCSoft

v02-02-03

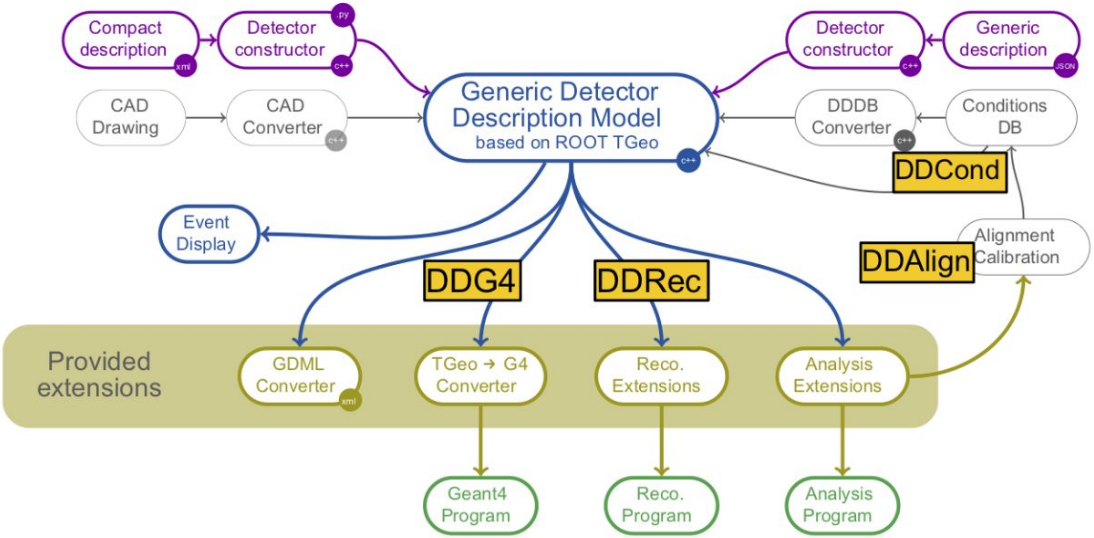


Code Reference under https://cern.ch/edm4hep



DD4hep

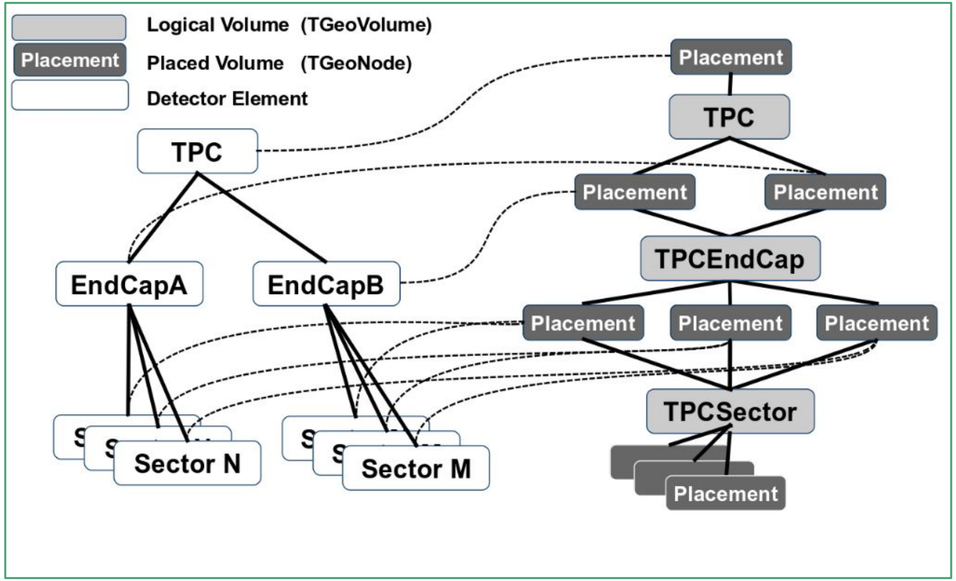
- DD4hep uses **ROOT TGeo** as Geometry implementation
- Geometry description in
  - compact *xml-files* and *C++ drivers*
  - other (generic) input sources
- output formats/interfaces
  - Geant4, GDML, *easily extendib*
- various interfaces (views) on geometry
  - DDRec, DDEve, DDAlign



Frank Gaede, CHEP 2019

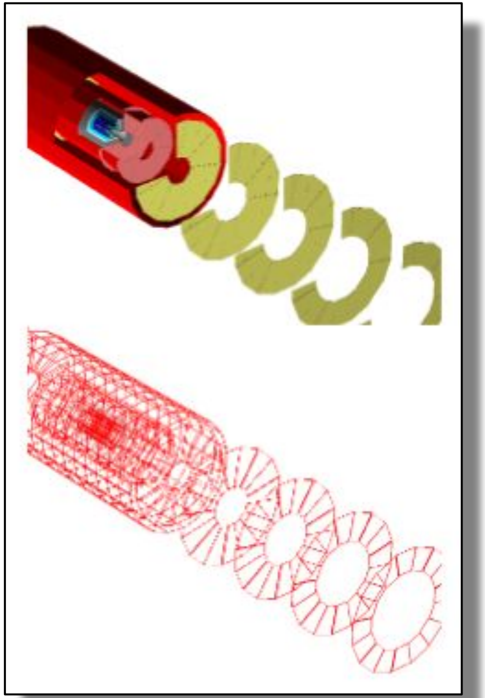
DDCore - DetElements and Geometry Trees

- Additional hierarchy of *DetElements* provides access
  - Alignment, Conditions, Readout (sensitive detectors), Visualization
  - Arbitrary *user defined objects*
- Define for every *touchable* that needs extra data
- the tree of *DetElements* provides the *high level view* into the detector geometry with subdetectors, measurement layers, etc.



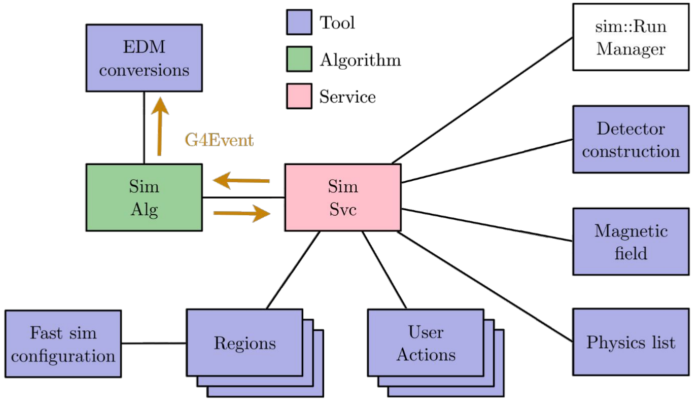
DDRec: dedicated tracking surfaces

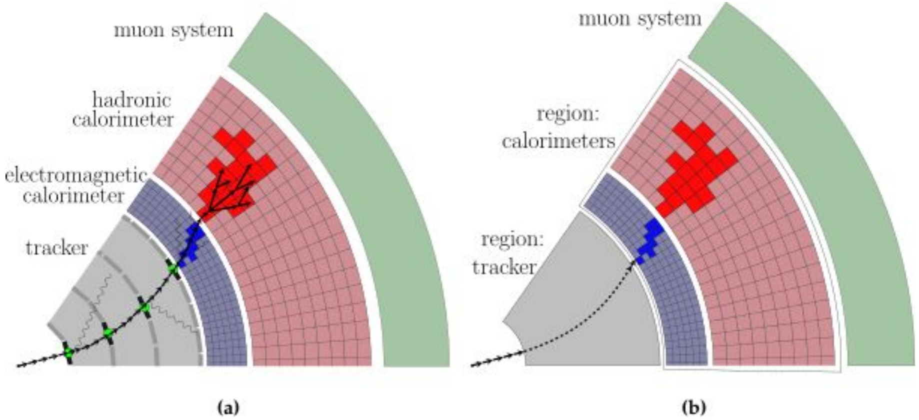
- **tracking** needs special interface to geometry
- Measurement and dead material *surfaces* (planar, cylindrical, conical)
- **surfaces** *attached to volumes* in detailed geometry model



- **u**, **v,** origin and normal
- inner and outer thicknesses and material properties
- *local to global* and *global to local* coordinate transformations:
    - **(x, y, z) ↔ (u, v)**

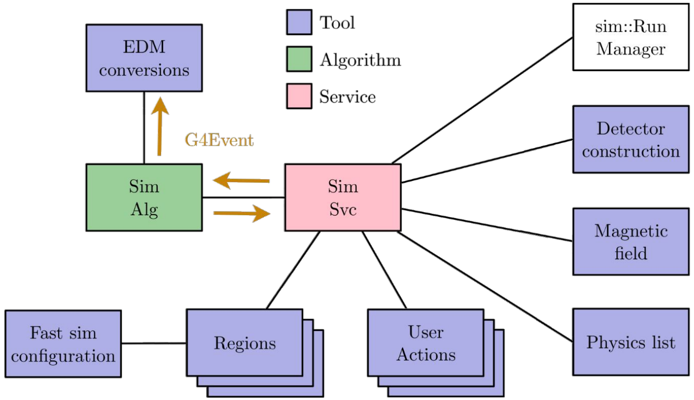
k4SimGeant4: allows mixing of Full/Fast Sim in Geant4

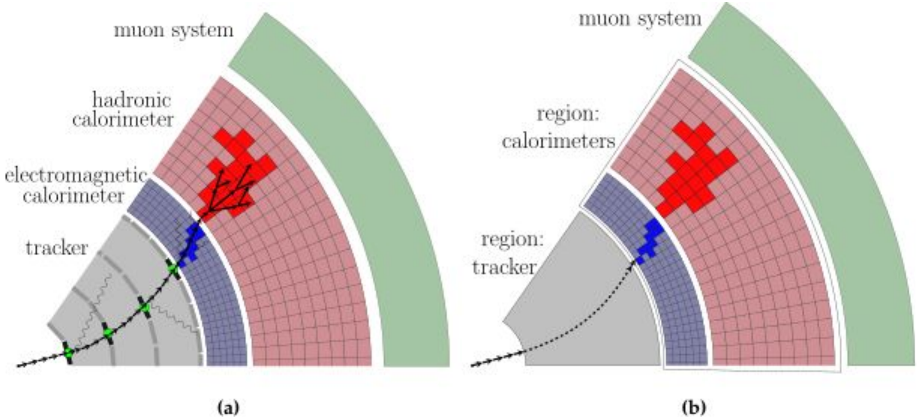




- Users can attach **parametrizations** to detector regions
- "Geant4 fast and full simulation for Future CircularCollider studies" link to CHEP '17 proceedings (Anna Zaborowska)
  - https://github.com/HEP-FCC/FCCSW/blob/master/Examples/options/geant_fastsim_tklayout.py
  - https://github.com/HEP-FCC/FCCSW/blob/master/Examples/options/geant_fastsim.py
- Ongoing work on using Machine-Learning techniques to speed up full simulation

k4SimGeant4: allows mixing of Full/Fast Sim in Geant4

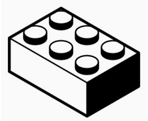
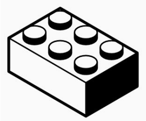
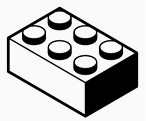




Currently consolidating different approaches to running Geant4 in a framework

- **k4SimGeant4 (evolution of FCCSW)**
- **Gaussino (LHCb)**
- **DDSim (DD4hep/iLCSoft)**

Modular approach

Updated style of “job option files” allows for easier re-use of parts of a job

```
# Geant4 algorithm
# Translates EDM to G4Event, passes the event to G4, writes out outputs via tools
from Configurables import SimG4Alg
geantsim = SimG4Alg("SimG4Alg")
from Configurables import SimG4PrimariesFromEdmTool
geantsim.eventProvider = SimG4PrimariesFromEdmTool("EdmConverter")
geantsim.eventProvider.GenParticles.Path = "GenParticles"
ApplicationMgr().TopAlg += [geantsim]
```

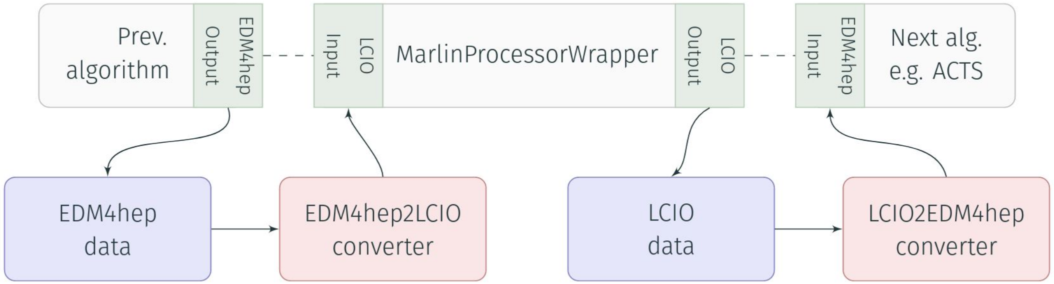
… even python-style import of configuration blocks!

```
from k4_workflow_blocks.fccsw.detector_fcc_hh_main import *
```

Reconstruction

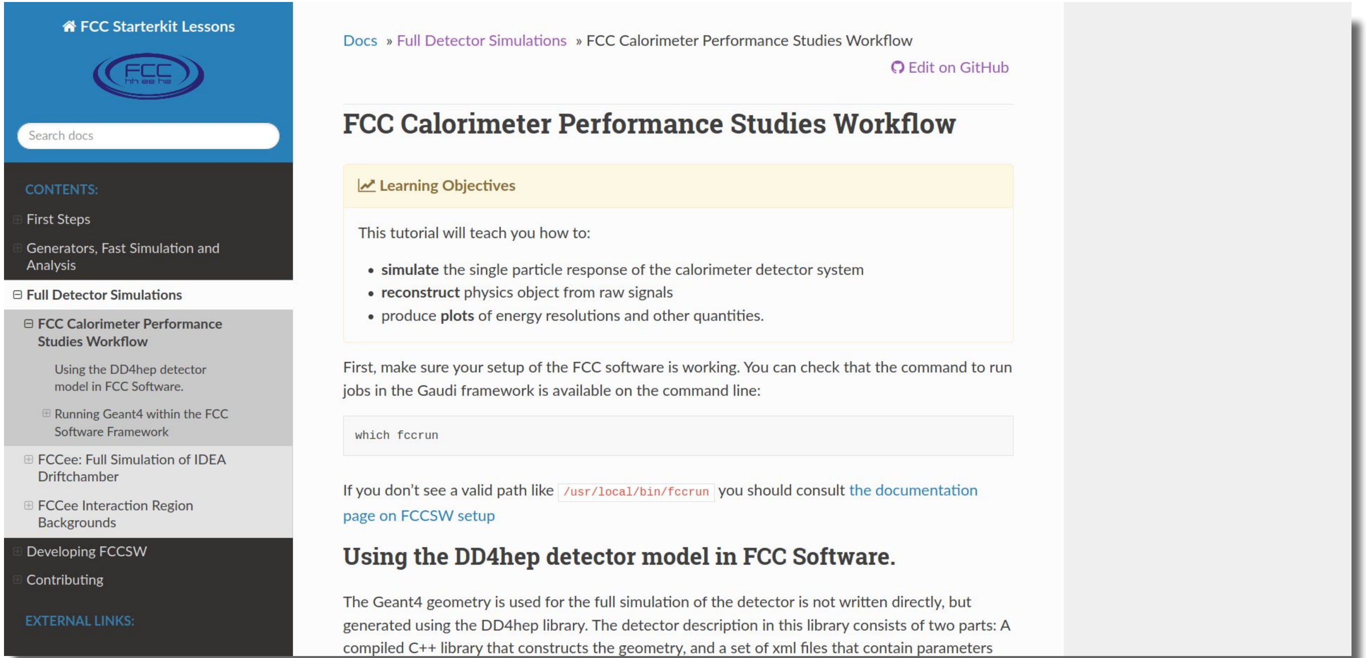
- Ongoing integration of other EP R&D work packages
  - ACTS: k4ActsTracking
  - CLUE: k4Clue
- iLCSoft Reconstruction chain (see Talk by Thomas Madlener) usable through **k4MarlinWrapper**

- In-memory on-the-fly conversion for LCIO ↔ EDM4hep
- Implemented as Gaudi Tools, can be attached to any *MarlinProcessorWrapper*
- LCIO → EDM4hep conversion achieved through k4LCIOReader
- Metadata conversion is being implemented
- Time measurements for the converters



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

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



FCC Starterkit Lessons

Search docs

CONTENTS:

- First Steps
- Generators, Fast Simulation and Analysis
- Full Detector Simulations
  - FCC Calorimeter Performance Studies Workflow
    - Using the DD4hep detector model in FCC Software.
    - Running Geant4 within the FCC Software Framework
  - FCCee: Full Simulation of IDEA Driftchamber
  - FCCee Interaction Region Backgrounds
- Developing FCCSW
- Contributing

EXTERNAL LINKS:

Docs » Full Detector Simulations » FCC Calorimeter Performance Studies Workflow

Edit on GitHub

FCC Calorimeter Performance Studies Workflow

Learning Objectives

This tutorial will teach you how to:

- **simulate** the single particle response of the calorimeter detector system
- **reconstruct** physics object from raw signals
- produce **plots** of energy resolutions and other quantities.

First, make sure your setup of the FCC software is working. You can check that the command to run jobs in the Gaudi framework is available on the command line:

```
which fccrun
```

If you don't see a valid path like `/usr/local/bin/fccrun` you should consult the documentation page on FCCSW setup

Using the DD4hep detector model in FCC Software.

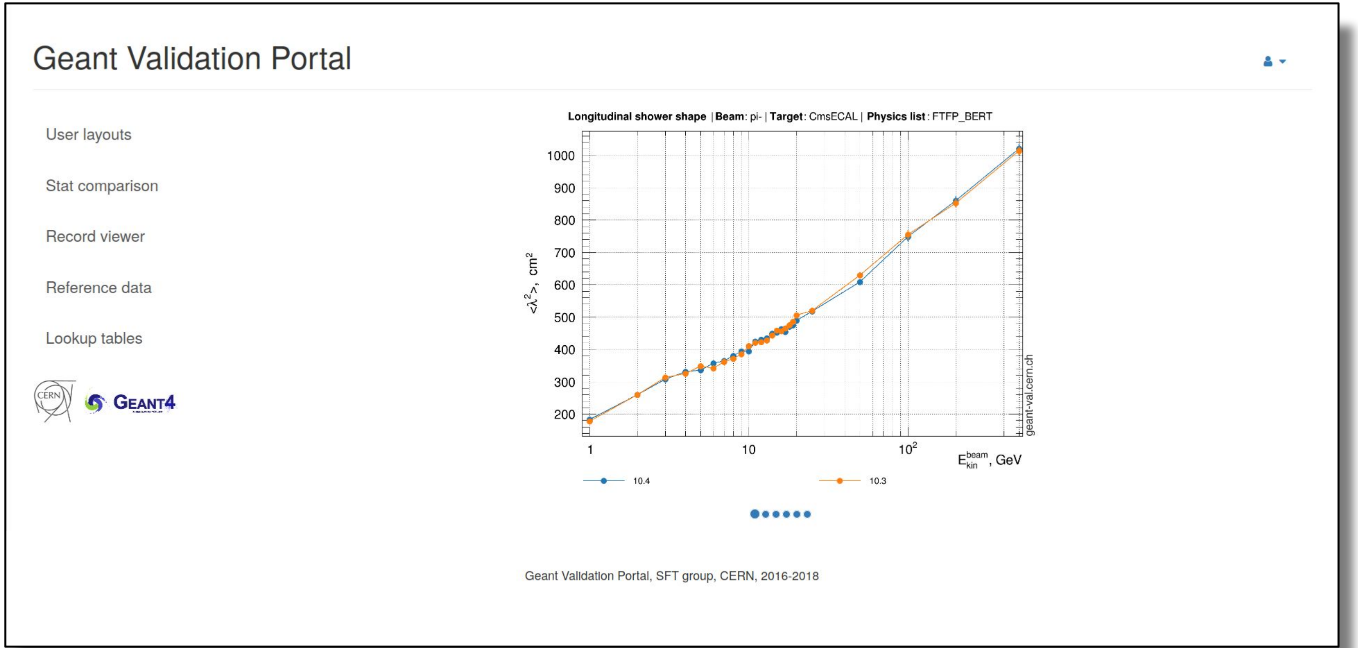
The Geant4 geometry is used for the full simulation of the detector is not written directly, but generated using the DD4hep library. The detector description in this library consists of two parts: A compiled C++ library that constructs the geometry, and a set of xml files that contain parameters

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

- Detector simulations are complex
  - performance, flexibility: full / fast simulations, validation
- Collaboration is crucial
  - Experiment-independent software libraries
  - Key4hep
- Need new developments as well as maintenance for past efforts!