A Software Framework for FCC studies: Status and Plans

FCC France 2019, Paris-LPNHE

Nov 14, 2019

C Helsens, G Ganis
(FCC Software coordinators)
CERN-EP

Basic Software Requirements

- Conceptual detector design studies
  - Flexibility
  - Ideal detector descriptions
  - Open to evolution
- Broad range of event complexity
  - e^+e^- *vs* pp *vs* ep
- Need to support physics and detector studies
  - Parameterised, fast and full simulation
- Aim to de-duplicate efforts
  - One software stack to support all the cases,
    all detector concepts and future (proto-)collaborations

Basic Software Requirements

We have achieved most of this with the publication of the FCC Conceptual Design Reports

We need to continue the work to produce Technical Detector Concepts For FCC-ee to become a reality

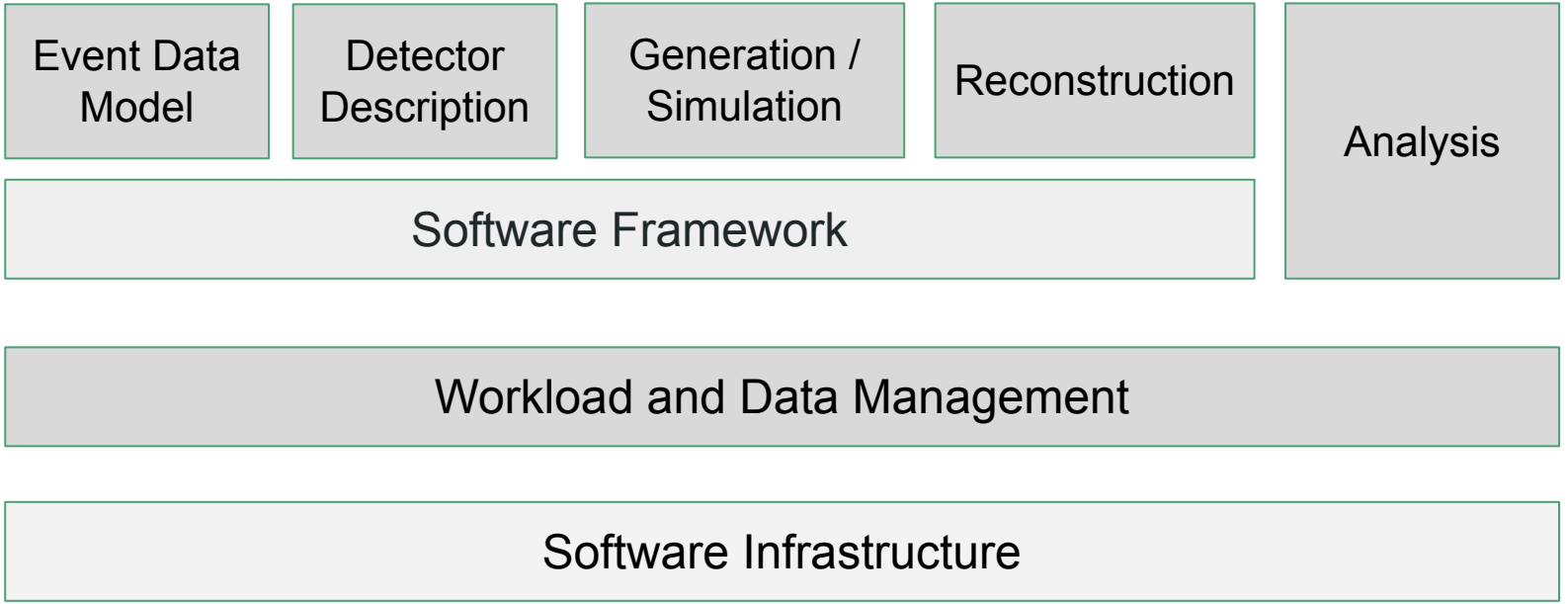
FCC Software Outlook

- Adopted Strategy
  - Adapt existing solutions from LHC
  - Look at ongoing common projects (AIDA)
  - Invest in streamlining of event data model

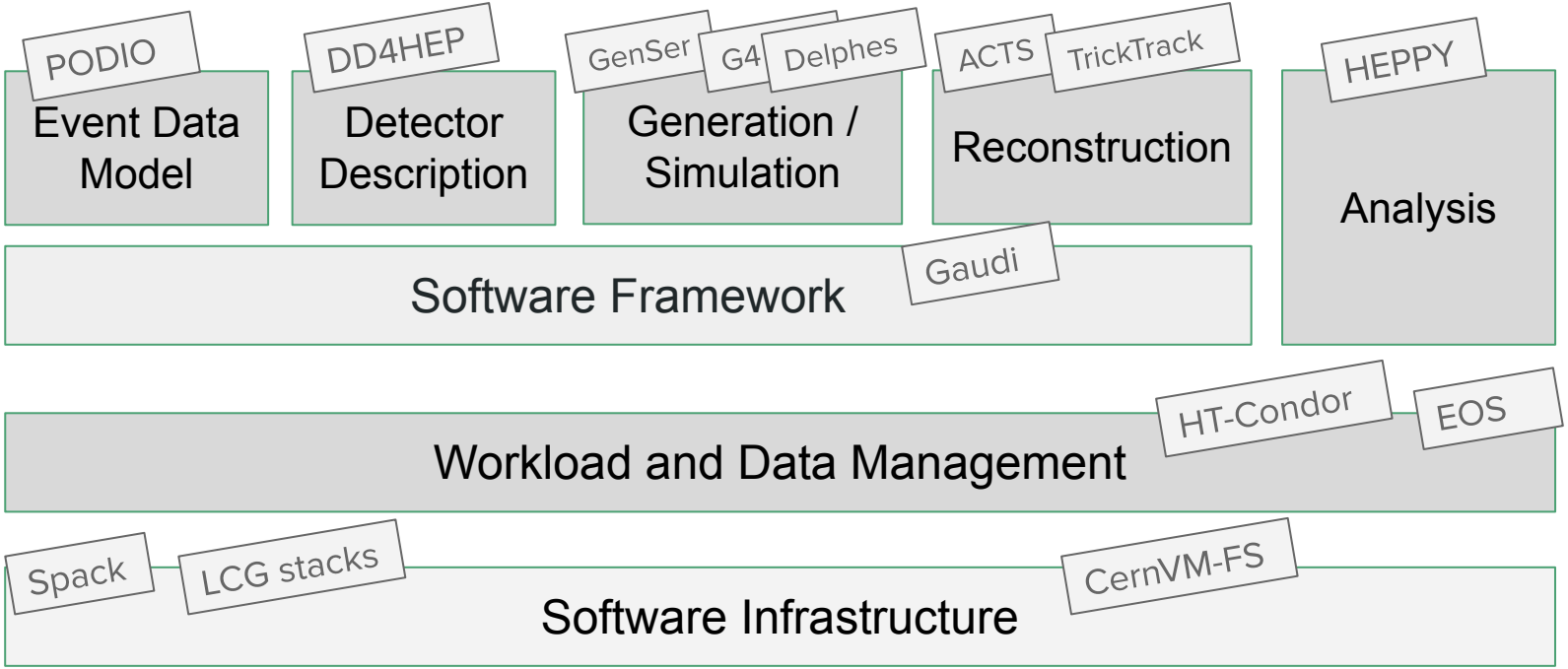
Next Slides

- Future: towards a common software for future experiments
  - Bologna workshop, June 2019
    - Present: LHC, ILC, CLIC, FCC, CEPC, SCTF, HSF
  - Agreed to:
    - Investigate the possibility to have a common event data model (EDM4hep)
    - Contribute to the development of a Common Turnkey Software Stack (Key4hep)
      - One framework (Gaudi best candidate), DD4hep, EDM4hep, Geant4, ROOT, ...

FCC Software Current Ingredients

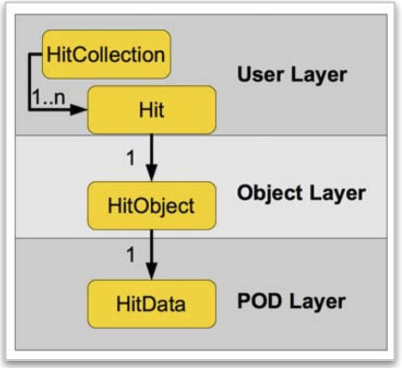


FCC Software Current Ingredients



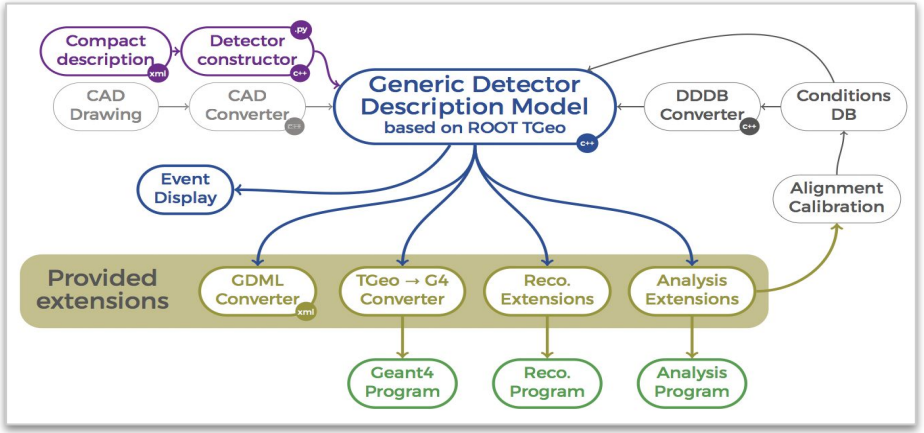
Event Data Model and PODIO

- LHC experiments / LC studies solutions suffering (partly) from
  - Overly complex data models with deep object-hierarchies
  - Unfavorable I/O performance
- **PODIO**: an EDM toolkit (AIDA2020 project)
  - Plain Old Data, automatic code generation, support for different backends
    - Keep memory model simple, enabling fast I/O and efficient vectorization
    - Consistent / homogeneous implementation, minimizes mistakes
  - High-level description in YAML format
  - Three-layers
    - User: handles objects and collections
    - Object: transient, relations between objects
    - POD: actual data structures
  - Follow-up for next AIDA being prepared
    - Schema evolution, memory and I/O optimizations



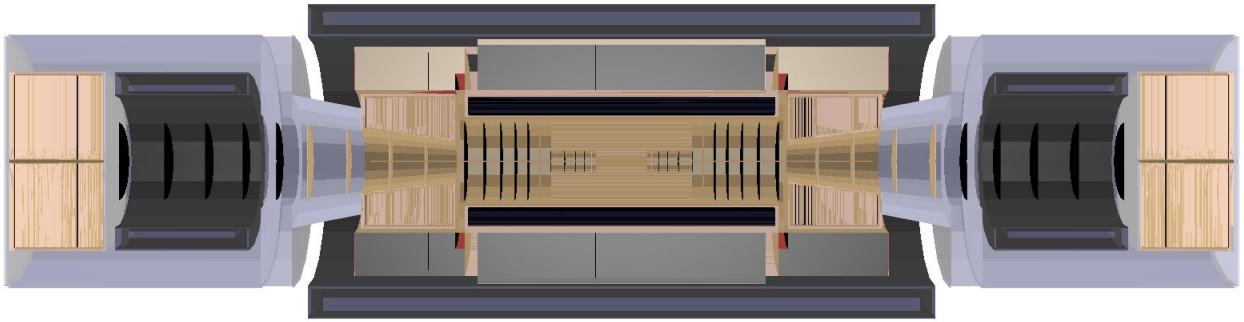
Detector Description: DD4hep

- Generic detector view appropriate to support
  - Simulation, reconstruction, analysis, …
- Design goals
  - Complete detector description
  - Single source of information
  - Support all stages of the experiment
  - Easy of use
- Part of AIDA2020
- Used by CLIC, ILC, FCC, LHCb, CMS, SCT

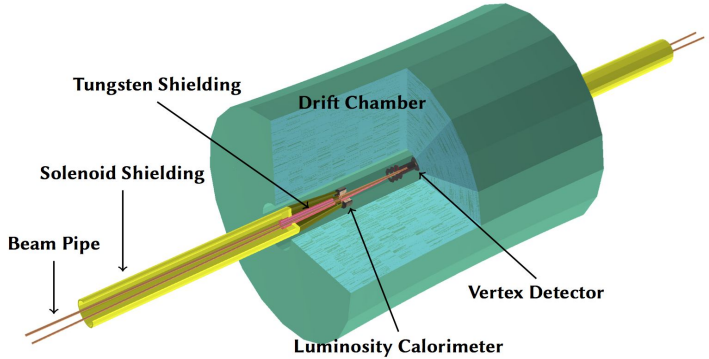


DD4HEP and FCCSW

- FCC-hh
  - Reference detector complete



- FCC-ee
  - IDEA Concept
    - Beam Pipe, instrumentation
    - Vertex Detector, Drift Chamber
    - DREAM Calorimeter (under dev)
  - LAr+Tile calorimeter (under dev)
  - CLiCDP



Software Framework: Gaudi

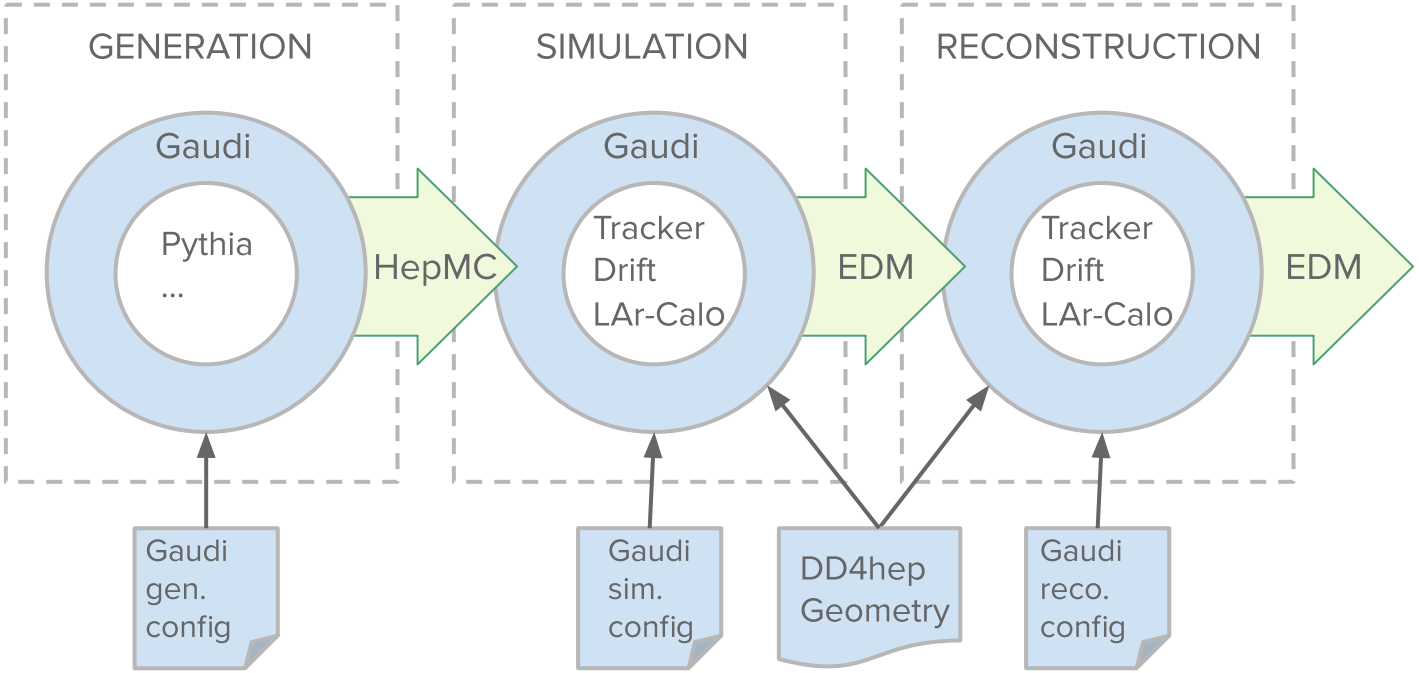
- *"The Gaudi project is a **open project for providing the necessary interfaces and services** for building HEP experiment frameworks in the domain of event data processing applications. The Gaudi framework is **experiment independent**."*

- Data processing framework designed to manage experiment workflows
  - Separate data and algorithms; well defined interfaces
  - User's code encapsulated in Algorithm's, Tool's / Interface's, Service's
  - Different persistent and transient views of data
  - C++, with Python configuration

- Originating from LHCb, adopted by ATLAS, Daya Bay, GLAST, LZ, …

- Actively developed to face LHC Run 3 and Run 4 challenges (high PU)

Gaudi and FCCSW

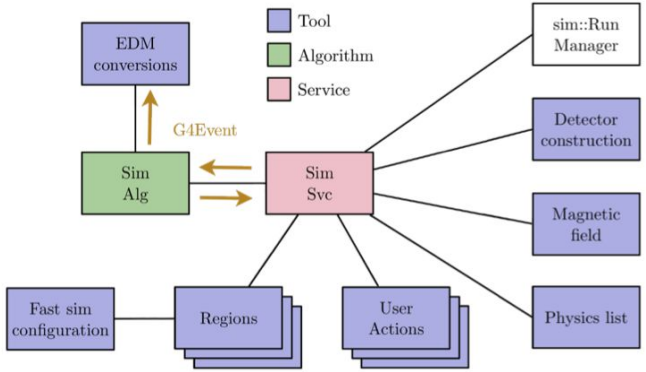


Monte Carlo Generation

- Interoperability with Monte Carlo generators at
  - Level 0 - common data formats
    - E.g. HepMC, Les Houches Event Files, MDI
  - Level 1 - callable interfaces
    - E.g. Pythia8
- Pythia8 also acts as LHEF event reader
  - Other converters available (HepMC, HepStd)
- Generator repository: GenSer @ LCG Stacks
  - Available in FCCSW
- MDI codes (MDISim, GuineaPig, ...) use level 0 interoperability
  - Adhoc converters under discussion

Simulation

- Geant4
  - Gaudi components exists to create
    - User Actions
    - Regions
    - Sensitive detectors
    - Selective output options
  - Mixing fast and full G4 simulation possible
    - SimG4Full / SimG4Fast



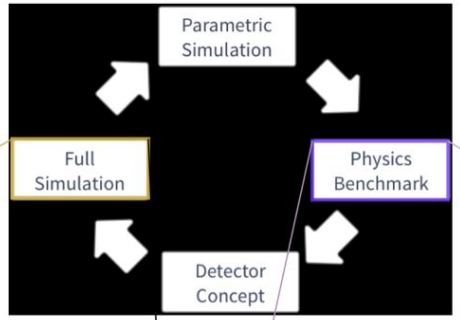
- Delphes
  - Gaudi interface
    - FCC EDM output

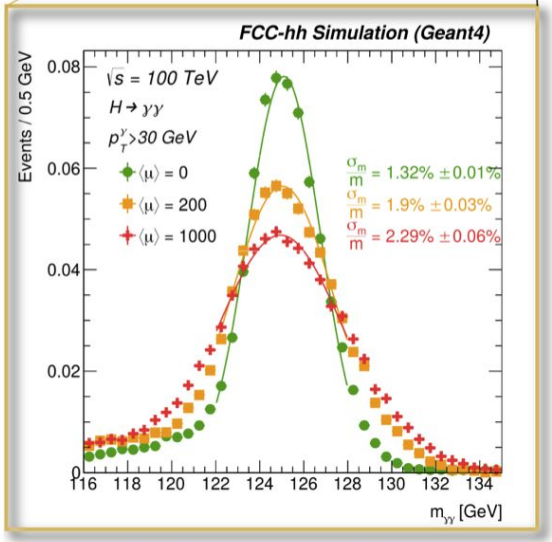
Fast / Full Simulation Interplay

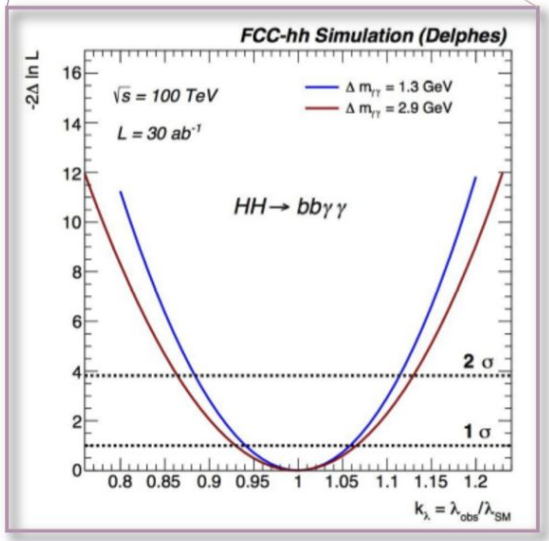
Example:

Higgs self-coupling

@ FCC-hh







Reconstruction

- Challenges: algorithm detector concept independent
  - Full flexibility, avoid duplication
- Tracking
  - Track seeding (TrickTrack)
  - Hough Transform for drift chambers
  - Under implementation / investigation
    - ACTS integration
    - Conformal tracking
- Calorimeters
  - Sliding window (rectangular/ellipse)
  - Topo-clustering
  - Deep learning

Considerations for Physics Analyses

- HEPPY: High Energy Physics with PYthon
  - Modular python framework for the analysis of collision events
  - Developed and still used for CMS

- In FCCSW HEPPY is used to
  - Process EDM events, apply-preselection, produce a flat and light ROOT ntuple
  - Analyse the ROOT ntuple
    - Not the only code used for this purpose

- Flexible but slow
  - Plan to move to a C++-based analysis framework, e.g. RDataFrame

Software Infrastructure

- **Typical HEP development workflow**

- Deliverables
  - FCCSW
  - Externals: FCCSW specific dependencies
  - Based on LCG releases

FCCSW - Main package

FCC externals
fcc-edm fcc-physics tricktrack
heppy podio ...

LCG release
Gaudi dd4hep ROOT ...

- Deployment on dedicated CernVM-FS repositories
  /cvmfs/fcc.cern.ch/, /cvmfs/fcc-nightlies.cern.ch/

- Builds (nightlies, releases) managed by **Spack**
  - Good feedback to HSF packaging WG

For the time being CERN centric, but distributed model is considered

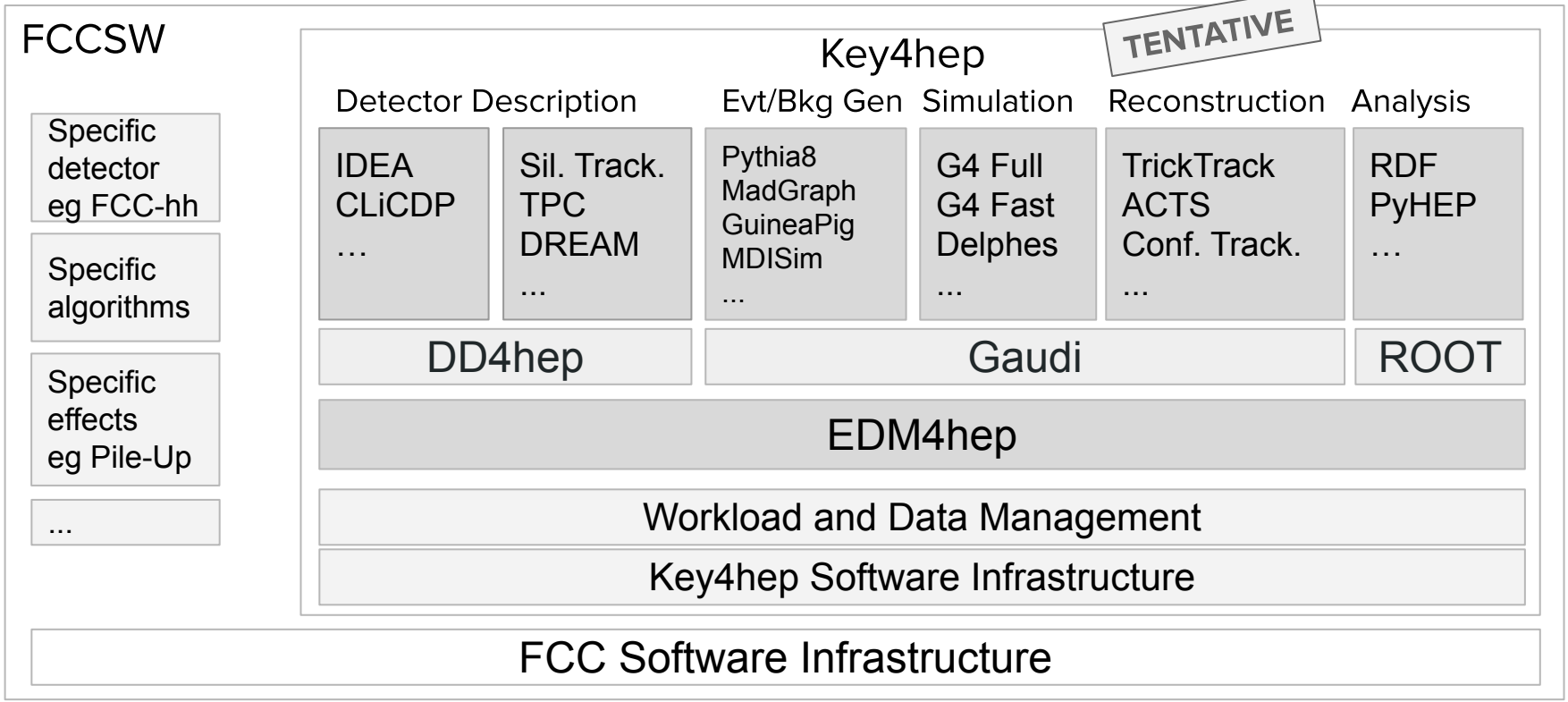
About the after-CDR for FCCSW

- Requirements:
  - Support more detailed studies, in particular for e^+e^-, focusing on
    - *Completeness*: state-of-Art generators, MDI, reconstruction / analysis algorithms, …
    - *Flexible detector description*: easy switch / replace sub-detectors, change dimensions / layout

- **Current approach** seems **adequate** to fulfill the requirements

- Foster and support
  - Participation from FCC institutes worldwide
  - Activities such as Key4HEP, which formalize and extend FCCSW approach

Connection with Key4HEP



How to contribute

- We are designing the software for future detectors and your help is welcome
- Proto-detector concepts have been made
  - Full/Fast/Parameterized simulations, Pen and paper detectors
- We need to finalised 2 to 4 of them towards detector technical studies in full simulation with full reconstruction (core software development at CERN)
  - Finalise/start geometry descriptions
  - Implement vertexing, flavour tagging, identification algorithms….
  - Assess the performances from common benchmarks
  - Start with interfacing generator
- Detailed tasks are being written and will be accessible soon

Summary

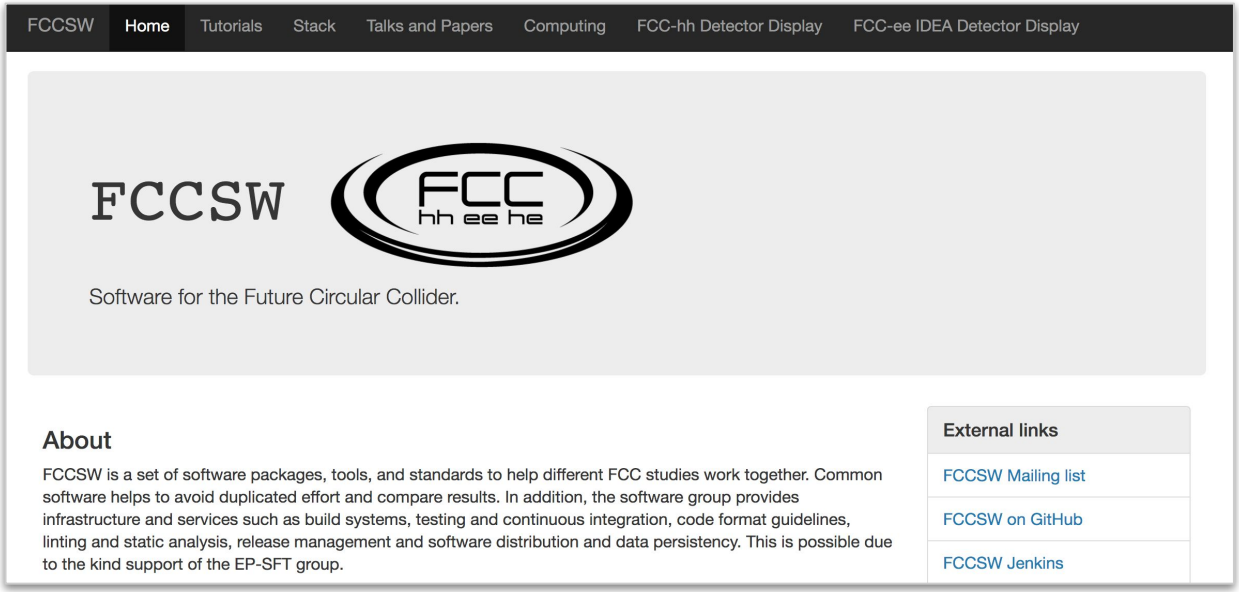
- The FCC software stack
  - Assembled using as much as possible existing components
  - Served well the purposes of the CDRs

- Started a new phase to further develop to support more detailed studies, in particular for e^+e^-

- Follow closely, participate and collaborate new common activities {Key4hep, EDM4hep}

Thank you!

- Web site https://cern.ch/fccsw
- Hands-on tutorials https://indico.cern.ch/event/839794/



Backup

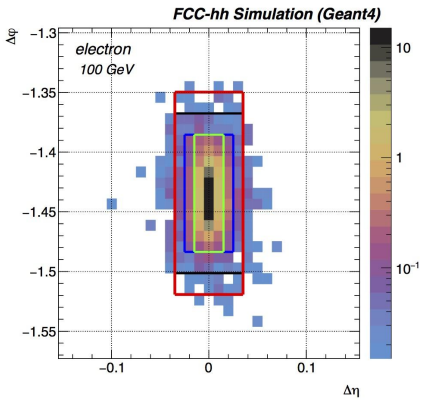
Gaudi and FCCSW

- Python scripts to configure tools and algorithms and define their processing order

```
$ fccrun \                                                  # Generic Gaudi app
      ./Examples/options/geant_fullsim_fccee_pgun.py \      # job definition
      --energyMin=10  --energyMax=10 --particleName=”mu-” \ # parameters
      --n=1000 --filename=fccee_idea_mu.root
```

- Each algorithm defines its parameters which can be overwritten of the command line

Calo Reconstruction Example



- Single 100 GeV e- reconstructed by sliding window

- Single 100 GeV pion in 8+10 layers of the E+HCal reconstructed by topo-cluster