



# An Introduction to FCC Software

---

Tutorials for Snowmass'21

Sep 22, 2020  
CERN-EP



# Context

# Software Requirements to support FCC studies



- 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



Current software met basic requirements at a sufficient level to support CDRs Physics studies

Next level of detail required for the Technical Design Report, with a special focus on FCC-ee

# Adopted approach

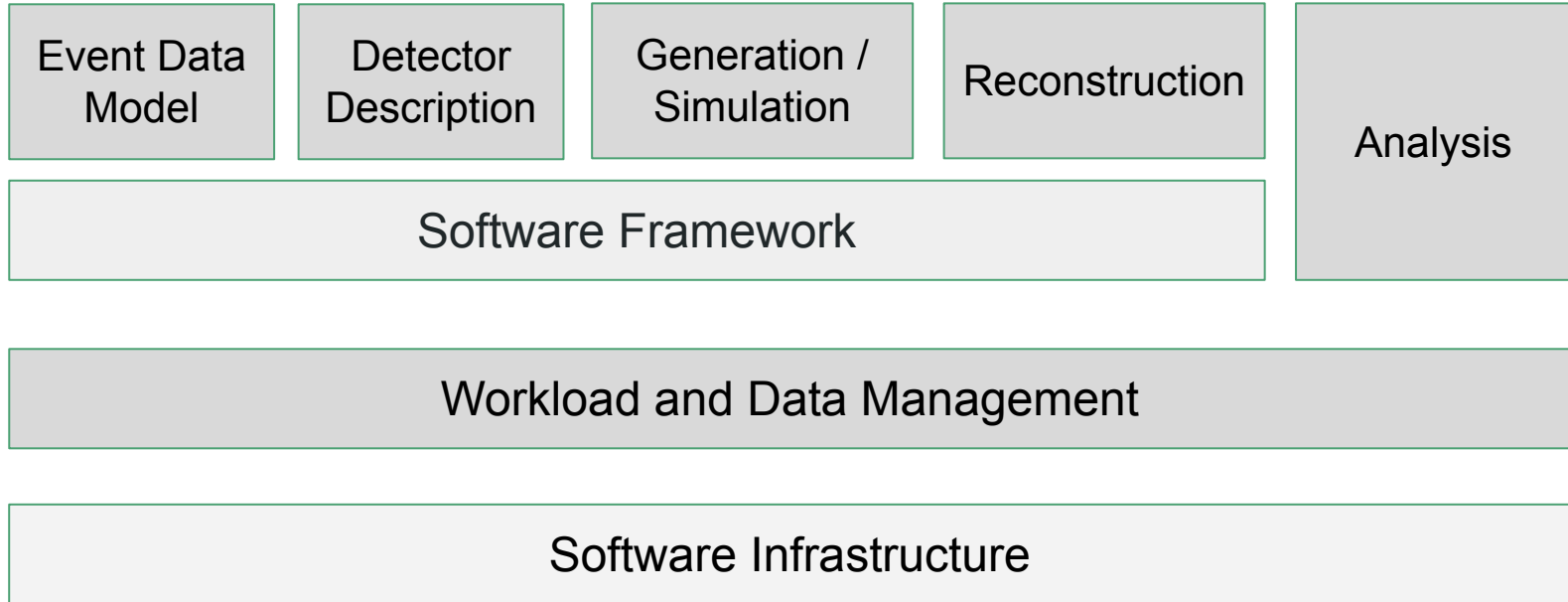


- Strategy
  - Adapt existing solutions from LHC
  - Look at ongoing common projects (AIDA)
  - Invest in streamlining of the event data model
- Future: towards a common software for future experiments
  - Discussed at [Bologna workshop, June 2019](#), [Hong Kong, 17 January 2020](#)
    - Present: LHC, ILC, CLIC, FCC, CEPC, SCTF, HSF
  - Follow-up by [regular weekly meetings](#), Tuesday 9am, CET
    - [EDM4hep](#): common event data model, v0.2 available
    - [Key4hep](#): common turnkey software stack, v 0.1 available

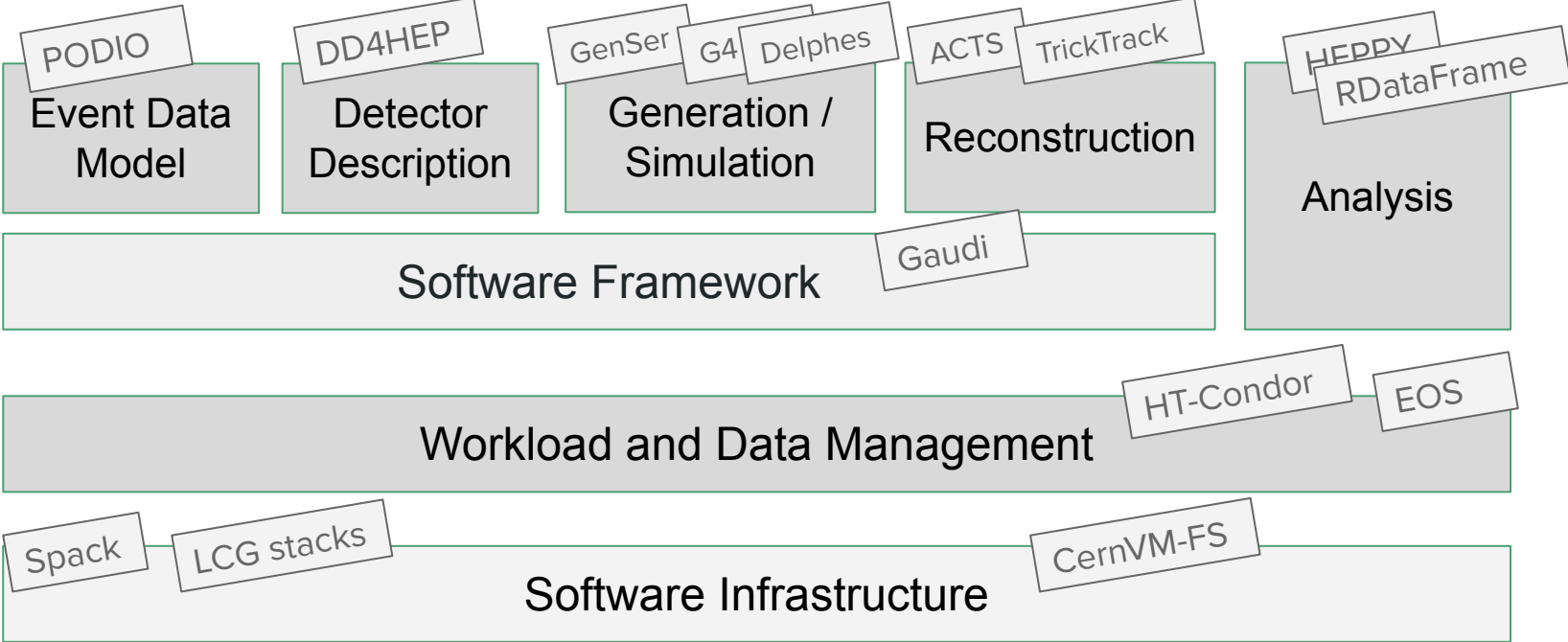


# FCC Software Components

# Components overview



# Components overview







# Event Data Model

# Event Data Model (fccedm)



- Current FCC-EDM

- Event/Run: EventInfo
- MC truth: MCParticle, GenVertex, GenJet
- Tracker: Track (PositionedTrackHit, TrackCluster, TrackState)
- Calorimeter: CaloCluster (PositionedCaloHit)
- Associations: ParticleMCParticleAssociation, DigiTrackHitAssociation, CaloHitAssociation, CaloHitMCParticleAssociation
- High-Level objects: TaggedParticle, Vertex (WeightedTrack), TaggedJet, ResolvedJet, MET

- Tuned on the needs of FCC-hh

- High-level objects of LHC inspiration

- TaggedParticle contains cross association between tracks and calo objects

# Event Data Model Evolution: EDM4HEP



- EDM4HEP: attempt to have a common EDM for FCC, LC, CEPC, OpenData,...
- Task force meeting every 2 weeks since February 2020 (started mid 2019)
  - Representatives from all future collider projects and LHC
- Started from comparing/merging FCC-EDM and LCIO (used by LC and CEPC)
  - Git repository available at [EDM4hep](#)
- Based on PODIO high-level EDM generator
  - As FCC-EDM
- Current status: v00-02 tagged on 21 Aug 2020
  - Complete EDM used for comparisons with LCIO and first studies with Delphes



# Software Framework

# Software Framework: Gaudi-based



- Framework toolkit to provide required interfaces and services to build HEP experiment frameworks
  - Opensource project and 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, Gaudi is adopted also by ATLAS
  - Actively developed to face LHC Run 3 and Run 4 challenges (high PU)

More in detail later ...

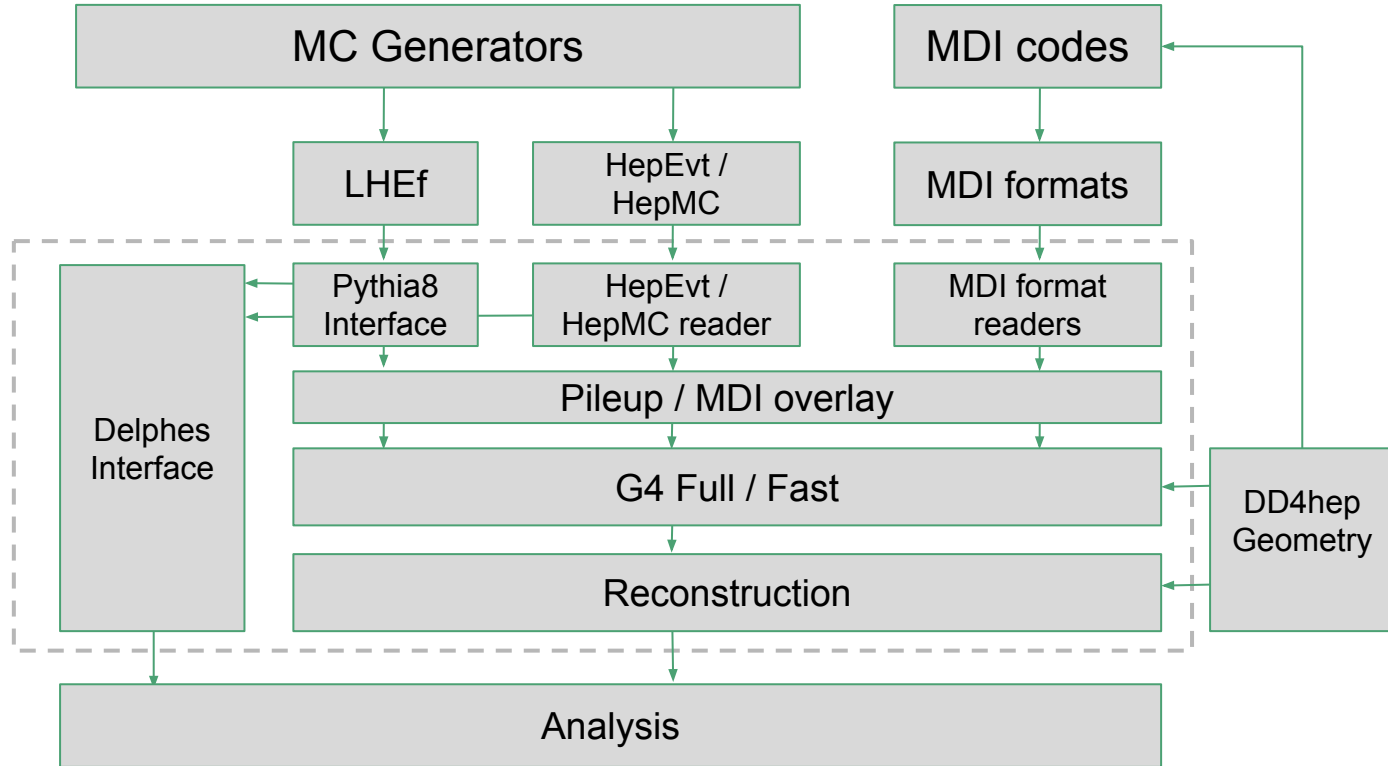


# Typical workflows

# Overview



FCCSW



# Monte Carlo Generators



- MC Generators are an essential ingredient to understand the potential of a detector
  - Need to simulate precisely enough both signal and backgrounds
- Backgrounds
  - Unwanted collisions products / signals
  - Beam-related backgrounds (SR, ...)
  - Beam unrelated backgrounds (cosmic rays, ...)
- Many programs exists to simulate the relevant processes
  - For e+e-, see, for example, [S Jadach summary](#) at the [11th FCC-ee workshop: Theory and Experiments](#)



# Monte Carlo Generators and FCCSW



- Generators repository: GenSer @ LCG software stacks
  - Generator Service hosted by EP-SFT  
*Collaboration with the authors and with the LHC experiments to prepare validated code for communities at the LHC*
  - Actively used by ATLAS, LHCb, SWAN and some SME experiments
  - Deployed via CernVM-FS
- MC generators are typically standalone codes
  - Noticeable exception is Pythia8, which provides a callable interface
- FCCSW interoperates MC generators mostly through common data formats
  - HepMC, LHEF
  - Pythia8 used to read LHEF files

# MC Generators: status and areas of work



- GenSer generators palette biased towards LHC
  - Good for FCC-hh, incomplete for FCC-ee
- General purpose generators such as Pythia8, Whizard, MadGraph5 available
  - Getting experience on how to use them effectively for FCC-ee
- Integration of KKMC and BHLUMI in GenSer in progress
  - Wrappers to produce HepMC and/or LHEF output required
- Similar work will be needed for MCSANC, BabaYaga, ...

Contributions welcome/required on interfacing and testing

Experience needed for interfacing: FORTRAN, FORTRAN&C++ interplay

For testing: ability/willingness to understand settings of a given generator

- (Beam- and) MDI- related backgrounds are source of systematics
  - Need to be controlled as precisely as possible
  - Critical aspect, for performance, is detector occupancy
    - Possibly also radiation damage
- Non-exhaustive list of programs to calculate these backgrounds
  - MDISim: *Synchtron Radiation, Single beam induced backgrounds*
  - SYNC\_BKG, SYNRAD+: *Synchtron Radiation*
  - GuineaPig++: *IP backgrounds, (In)coherent Pairs Creation,  $\gamma\gamma$  to hadrons*
  - Pythia8:  *$\gamma\gamma$  to hadrons*
  - BBBrem+SAD: *Radiative Bhabhas*
- Simplest and solid solution for integrating the MDI calculations in FCCSW is through shared data formats

- MDI code provides sets of events with the 4-vectors and vertex of the relevant particles
  - $\gamma$ 's for SR; e+e- pairs, hadrons for IP processes, ...
  - May include the interaction in the beam-pipe (as in MDISim)
- Evaluation of detector occupancy in FCCSW
  - Through **interaction** of MDI particles in the detector
  - Through **overlay** of MDI events to “signal” events for a more detailed background simulation
  - This may also be done with a **weighted mixture** of MDI processes

# MDI integration: status and areas of work



- Agreement to define shared documented data formats
  - Work not started yet
- GuineaPig++ interfaced to iLCSoft
  - May be used as source of inspiration
- Event overlay in FCCSW
  - Same technology used for pileUP
  - Need to be tested and validated

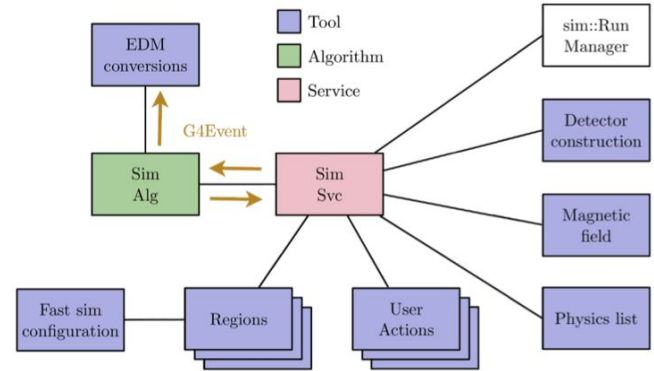
Contributions required to all the above items  
Experience required: file formats, FORTRAN, C++

# Simulation



## ● Geant4

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



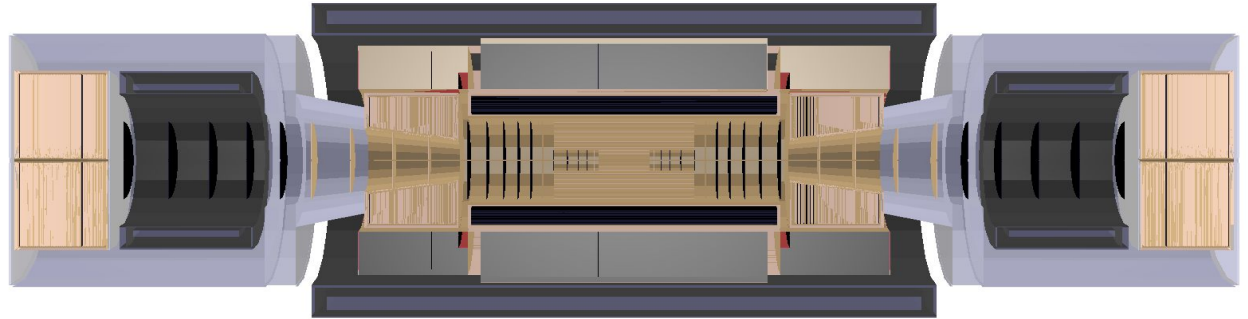
## ● Delphes

- Gaudi interface
  - FCC EDM output

# FCC detector palette in DD4hep: FCC-hh



## FCC-hh baseline



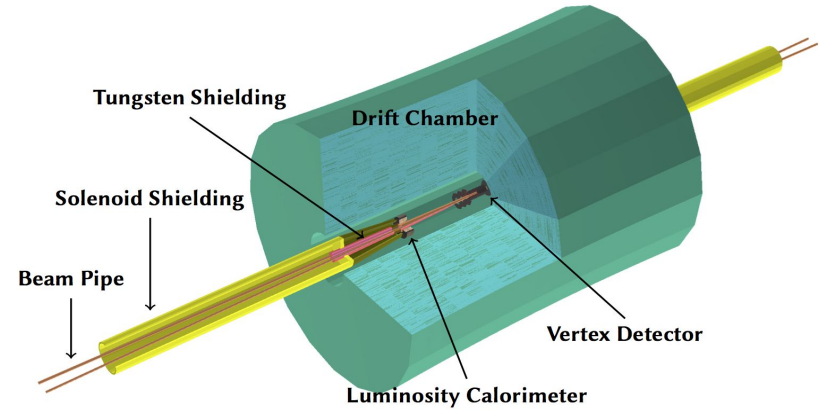
- Barrel, Endcap, Forward
- Beam Pipe, Shielding, Magnet solenoid
- Silicon Tracker
- LAr ECal, Tile HCal
- Muon System

# FCC detector palette in DD4hep: FCC-ee



## FCC-ee IDEA

- Beam Pipe, Beam instrumentation
- Lumical, HOM Absorber
- Vertex detector
- Drift Chamber
- Dual Readout Calorimeter
- Muon System



DR calo required support for optical properties in DD4hep available in the latest version shipped with FCCSW



# FCC detector palette in DD4hep: FCC-ee



## Possible alternatives for FCC-ee

- “IDEA” with reduced version of LAr ECal + Tile HCal
  - First DD4hep description available for testing
- CLD
  - Geometry description in DD4hep exists: <https://github.com/iLCSoft/lcgeo>
  - Requires some integration for usage in FCCSW

Contributions welcome/required on:

- DD4hep description for the IDEA DR calo, including digitization; muon system
- Enabling of CLD in FCCSW

Experience needed: familiarity/willingness to learn: DD4hep, detector geometry, Geant 4 simulation

# FCC detector palette for Delphes



- Validated
  - [FCC-hh baseline](#)
  - [HL-HELHC baseline](#)
- Available for FCC-ee (need adjustment in the steering code)
  - [IDEA](#)
  - [CLICDet](#)

Possible contributions: testing, validation, fine tuning of existing cards; scripts or tool to easy variate relevant dimensions

Experience needed: familiarity with Delphes, Gaudi, simulation

# Reconstruction



- Challenges: algorithm detector concept independent
  - Full flexibility, avoid duplication
- Tracking
  - Track seeding (Silicon tracker), Hough Transform (drift chambers)
  - Under development / investigation: ACTS integration, Conformal tracking
- Calorimeters
  - Sliding window (rectangular/ellipse), Topo-clustering
  - Under development / investigation: ML techniques

Possible contributions: vertexing, ACTS, ML, particle ID

Experience needed: familiarity with reconstruction algorithms, Gaudi, C++

# End-User Physics Analysis



- Output of previous steps are ROOT TTrees
- High-level interfaces, typically python-based, convenient
  - E.g. HEPPY, modular python framework for the analysis of collision events (CMS)
  - Flexible but (very) slow
- ROOT RDataFrame replacement of HEPPY available
  - Will be used for the hands-on
- Room for development and introduction of new technologies
  - E.g. efficient python-steered ecosystems, e.g. numpy, awkward arrays

Possible contributions: develop HEPPY replacement based on RDataFrame or other fast technology (e.g. awkward)

Experience needed: familiarity with ROOT and RDataFrame; advanced Python

More in Delphes tutorials ...



# Infrastructure, OS support, resources

# Software Infrastructure



- Typical HEP development workflow
- Deliverables
  - Full dedicated LCG stack
    - LCG\_97a\_FCC\_2, LCG\_97a\_FCC\_3
  - Full Spack managed build
- Deployment on dedicated CernVM-FS repositories
  - /cvmfs/fcc.cern.ch/
    - Also /cvmfs/fcc-nightlies.cern.ch/ for testing

FCC specific  
fccsw fcc-physics fcc-edm

LCG release  
Gaudi dd4hep ROOT ...

# OS support



- Currently CERN-centric
- Support for the default version running on lxplus
  - CentOS 7, gcc 8, gcc 9
  - Experimental support for Ubuntu 20.04 LTS, MacOSX
- VM, based on CernVM, available to recreate equivalent environment
  - Works from everywhere but speeds depends on the network

Possible contributions: provide support for other OSs (Ubuntu, MacOSX, ...)

Experience needed: familiarity with build systems, linux, ...

# CERN resources and access policy



- CERN resources are available to member of institutes having signed the [Memorandum of Understanding and its addendum](#)
- EOS areas for **data or large files**: `/eos/experiment/fcc`
  - Current quota: 400 TB
  - E-group membership: `fcc-eos-access` (and alike)
  - Dedicated areas for ee, hh, eh, helhc, users
    - Plan to deprecate 'users': each CERN user has 1 TB at `/eos/user/u/username`
    - Needs to be enabled on [Account Management](#) page
- EOS areas for **shared files**: `/eos/project/f/fccsw-web/www`
  - Also accessible also [via web](#)
- **Dedicated queue on LXBATCH**
  - AccountingGroup = "`group_u_FCC.local_gen`" (on HTCondor)
  - E-group membership: `fcc-experiments-comp`



# Areas of work summary



- MC generators interfacing
- MC generators testing
- MDI shared formats
- GuineaPig++ integration
- Overlay of MDI/signal events
- IDEA DR Calo full simulation
- IDEA Muon system full sim
- Validation of LAr Ecal for FCC-ee
- Enabling of CLD in FCCSW
- Validation/testing of Delphes cards
- Vertex reconstruction
- ACTS integration
- ML for calo reconstruction
- e, mu, tau, c, b tagging / ID
- RDataFrame based analysis
- Porting to other OSs
- ...

# Thank you!



- Web site <https://cern.ch/fccsw>

A screenshot of the FCCSW website homepage. The top navigation bar is dark with white text for 'FCCSW', 'Home', 'Tutorials', 'Talks and Papers', 'Computing', 'FCC-hh Detector Display', and 'FCC-ee IDEA Detector Display'. The main content area has a light gray background. On the left, the text 'FCCSW' is displayed in a large, bold font, with the tagline 'Software for the Future Circular Collider.' below it. On the right, there is a large version of the FCC 'hh ee he' logo. Below the main content, there is an 'About' section on the left and an 'External links' section on the right. The 'About' section contains a paragraph of text. The 'External links' section is a table with three rows, each containing a blue link: 'FCC Documentation', 'Key4HEP Documentation', and 'FCC Mattermost Chat'.

## About

FCCSW is a set of software packages, tools, and standards to help different FCC studies work together. Common software helps to avoid duplicated effort and compare results. In addition, the software group provides infrastructure and services such as build systems, testing and continuous integration, code format guidelines, linting and static analysis, release management and software distribution and data persistency. This is possible due to the kind support of the EP-SFT group.

## External links

[FCC Documentation](#)

[Key4HEP Documentation](#)

[FCC Mattermost Chat](#)



# Backup

# Summary



- The FCC software stack has been assembled using as much as possible existing components
  - Served well the purposes of the CDRs
- Started a new phase of development to support more detailed studies in view of the TDRs, in particular for  $e^+e^-$
- Areas of work identified
- Follow closely, participate and collaborate new common activities {Key4hep, EDM4hep}



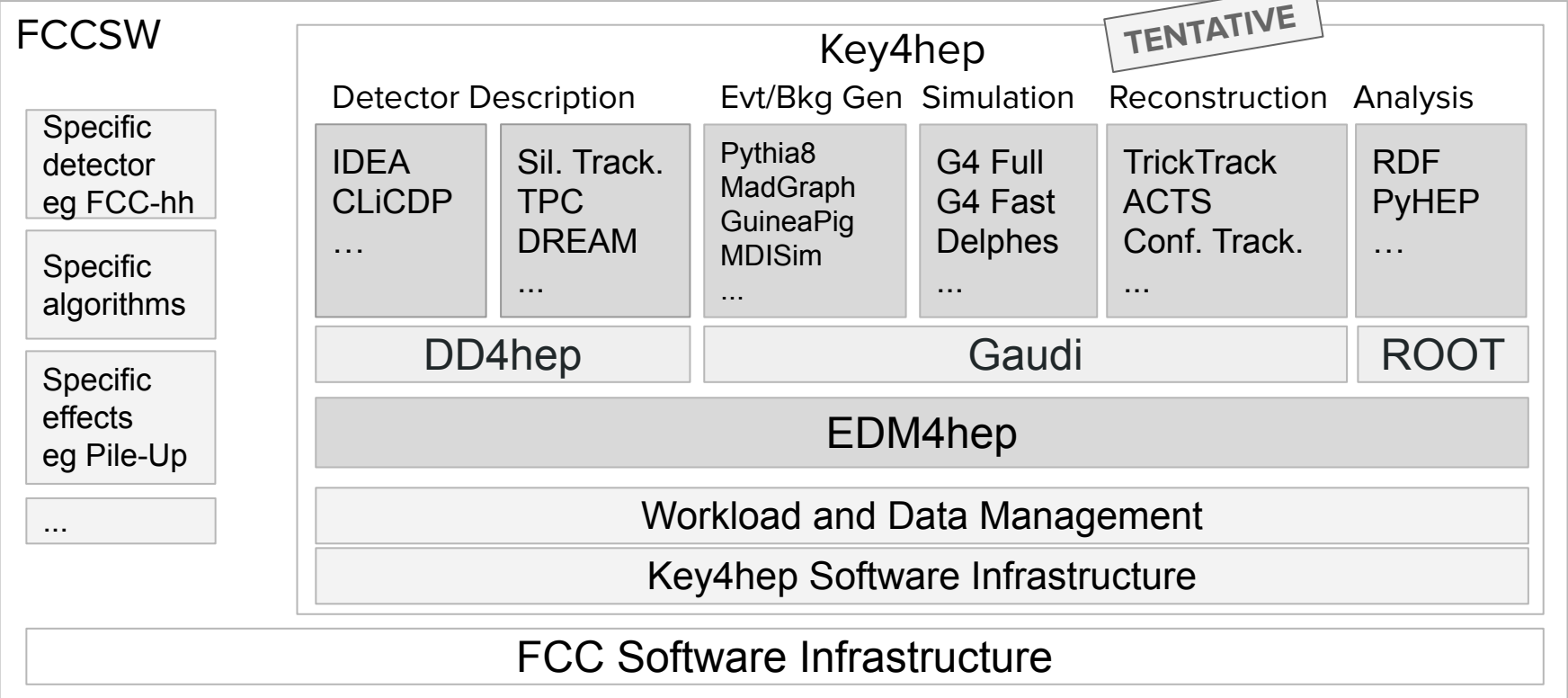
# About Key4HEP

# Key4HEP



- Depends **crucially** on EDM4HEP
  - Sets the chronological order
- **Full-time fellow just started now (Jan 2020)**
  - Another fellow working through CREMIN PLUS soon (Mar 2020)
- **Possible contribution from EU projects**
  - AIDA++, MSCA-ITN SPIRAL-NET
- **Possibly rapid development when EDM4HEP is available**
  - Key4HEP core  $\approx$  FCCSW core + EDM4HEP
  - Algorithms (FCCSW, other) adapted to EDM4HEP can then be used
- **Deliver early and often approach**

# Connection with Key4HEP



# After-CDR for FCCSW



- Current approach seems adequate to fulfill the requirements
- Contribution from FCC institutes is essential and very welcome
  - Interested people should manifest themselves and pick-up an area of work
  - Public software coordination meetings will be resumed soon (see next)
- Foster activities such as Key4HEP, which formalize and extend FCCSW approach
  - Exchange / share effort within the community



# Software Coordination Meetings



Open software coordination meeting will restart on **31 January 2020**

- Frequency: **Bi-weekly meetings with remote connection**
- Time slot: **Friday morning 9h-11h**
- Location: **40/R-B10**
- Announced on **[fcc-experiment-sw-dev](#)**

# Hands-on tutorial tomorrow!

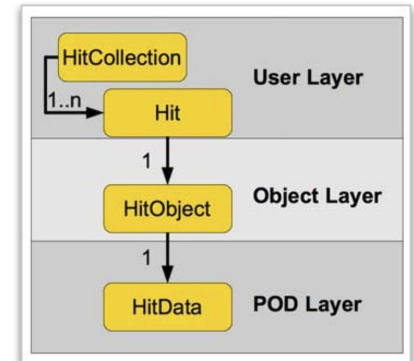


- Location: 4/3-006 (this room)
- Time slots: 9h-13h, 14h-17h
- Program: Roughly the same as in [October SW Workshop](#)
  1. Running FCCSW standalone
  2. Example of Physics Analysis (ZH)
  3. Delphes cards optimization
  4. Tracking with Drift Chamber
- Bring your own laptop!

# 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

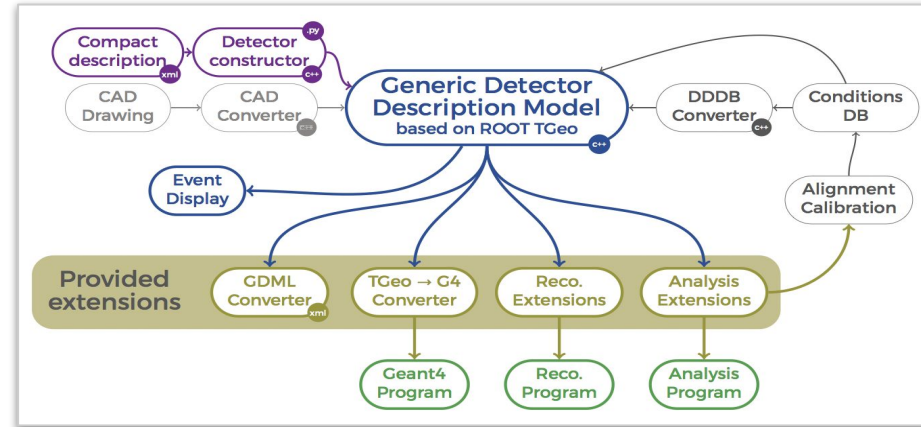
- Simulation, reconstruction, analysis, ...

- Design goals

- Complete detector description
- Single source of information
- Support all stages of the experiment
- Easy of use

- Part of AIDA2020

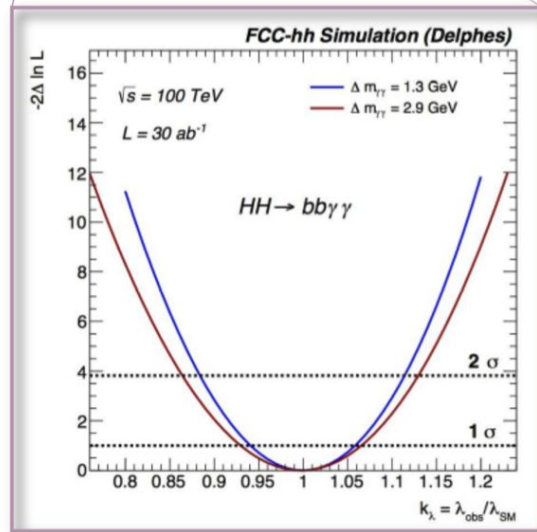
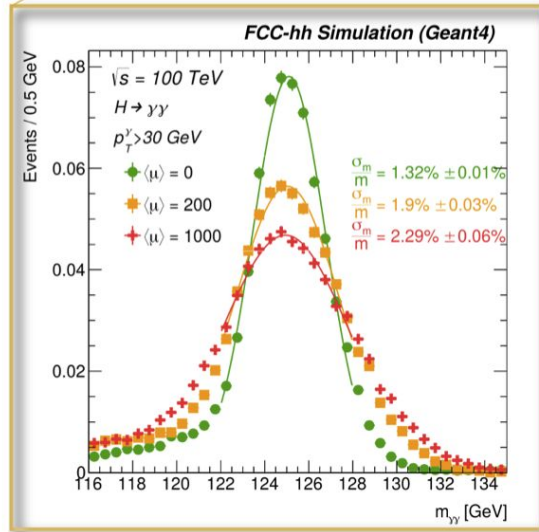
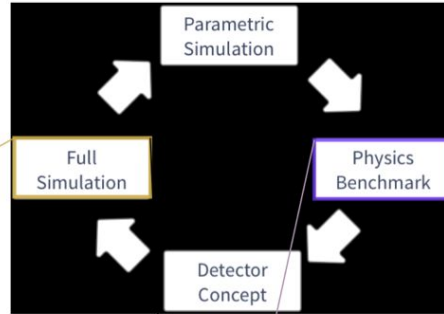
- Used by CLIC, ILC, FCC, LHCb, CMS, SCT



# Fast / Full Simulation Interplay



Example:  
Higgs self-coupling  
@ FCC-hh





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

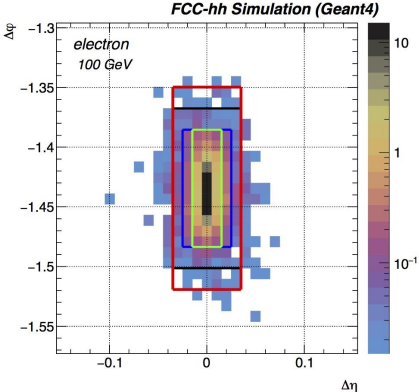
```
$ fccrun \                                     # Generic Gaudi app
  ./Examples/options/geant_fullsim_fccee_pgun.py \   # job definition
  --energyMin=10 --energyMax=10 --particleName="mu-" \ # parameters
  --n=1000 --filename=fccee_idea_mu.root
```

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

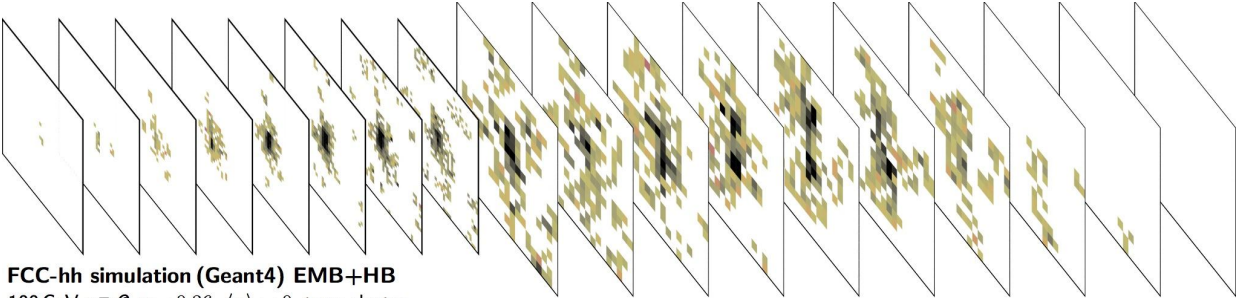
# Calo Reconstruction Example



- Single 100 GeV e- reconstructed by sliding window



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



**FCC-hh simulation (Geant4) EMB+HB**  
100 GeV  $\pi^-$  @  $\eta = 0.36$ ,  $\langle \mu \rangle = 0$ , topo-cluster