FCC Software for Physics and Detector Studies

XXVII Cracow EPIPHANY Conference on Future of particle physics

Jan 10, 2021

G Ganis, C Helsens
CERN-EP

Context

The FCC Software (FCCSW)

FCCSW is still largely based on what was used for the CDR

- Good modular structure based on the Gaudi framework (LHCb, ATLAS)
    - Base for the Key4hep common project
- Provide support for all the required functionality
    - Event Data Model (EDM), Generators, Geometry, Fast/Full simulation, Reconstruction, ...
- Current main limitations are in the implemented functionality
    - Available generators, in particular for FCC-ee
    - Palette of detector concepts with parametrized description ➔ Quality of the description
    - Palette of detectors with detailed geometry description ➔ Digitisation of their signal
    - Reconstruction algorithms

FCC Software evolution

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, ...

Follow-up in Hong Kong, 17 January 2020 (Present: ILC, CLIC, FCC, CEPC)

- Agreed to
  - set-up regular weekly meetings, a GitHub repository, documentation,
  - deployment area on CVMFS, ...
  - Get quickly first version of EDM4hep and Key4hep available

Today status: {EDM4hep v0.2.1, k4FWCore v0.1.1} being integrated in FCCSW

FCC software goals for the CDR++

Support for more detailed studies, in particular for e+e-, focusing on

- Completeness
  - State-of-Art generators, MDI support, reconstruction / analysis algorithms, ...
- Flexible detector description
  - Easy switch/ replace sub-detectors, change dimensions / layout, ...
- Easy-of-use
  - Low usability thresholds and fast / easy learning curve
- Adequate computing support and CPU / storage resources
- Extensive documentation and regular training

Foster development and use in

- Physics studies, Detector optimization, Machine-Detector Interface

Foster / support substantial participation for FCC institutes worldwide

Ensure that SW is part-and-parcels of the Turnkey Software Stack

Experimental challenges for FCC-ee (on software)

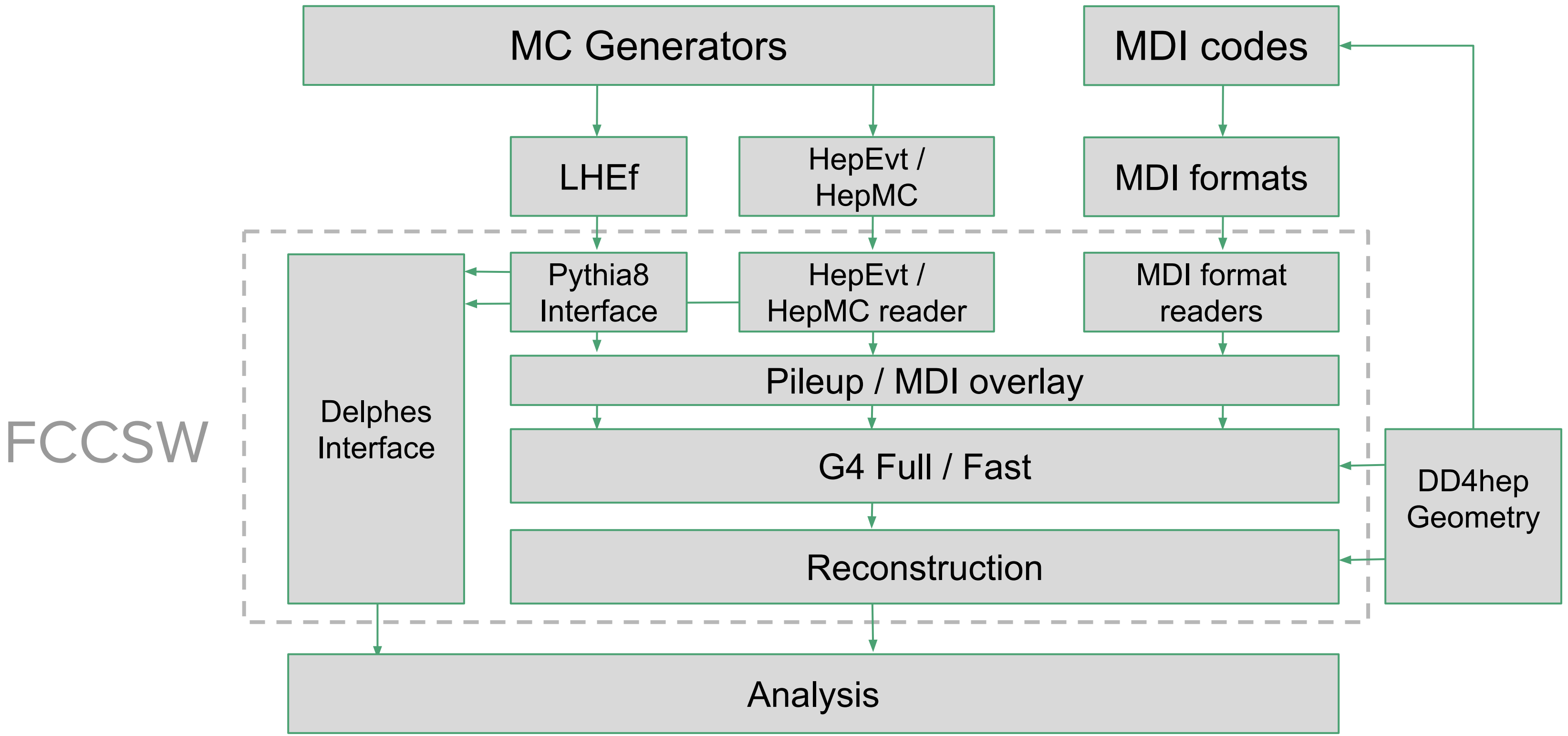
Ref: A Blondel @ FCC Physics on March 30th and case studies

- Requirements on detector understanding O(1-2) better than LEP
  - Need to simulate a lot of (reliable) data
- Priority is to have **as soon as possible**
  - Flexible **full** simulation and reconstruction for case and detector design studies
    - b-tagging, reconstruction and vertex geometry, tracking, PID etc.
  - Flexible **fast** simulation to support case studies
    - And also generator-level studies, or brain activity
- Independent of Snowmass
  - But Snowmass may be instrumental to foster activities and provide synergies
- Possible computing efficiency issue (in particular @ Z)
  - Interplay fast / full simulation may be required to mitigate

What can we do for FCC(-ee) today?

A few showcases ...

Typical workflows



Monte Carlo (MC) generators

MC 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

MC 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

- General purpose complements such as Tauola, EvtGen, … available
  - Getting experience on how to use them with flavour and tau physics use cases
  - Help to validate the data model (differences in truth navigation)

- Integration of LEP generators
  - KKMCee completed
  - BHLUMI in progress
  - Similar work will be needed for MCSANC, BabaYaga, ...

Delphes @ FCCSW

Delphes: parametrization of a detector concept

What's Delphes

- *A framework for fast simulation of a generic collider experiment*
  - C++ framework providing a fast multipurpose detector response simulation
  - Includes a tracking system, embedded into a magnetic field, calorimeters and a muon system
    - Effect of magnetic field, granularity of calorimeters, sub-detector resolutions
  - Interfaced to standard file formats (e.g. Les Houches Event File or HepMC)
  - Also outputs observables such as isolated leptons, missing energy and collection of jets
- Delphes output in EDM4hep
  - Key4hep provides executables interfaced to Delphes producing EDM4hep output
    - E.g. DelphesPythia8_EDM4HEP

FCC detector concept palette for Delphes

- Validated and used for CDR
  - FCC-hh baseline, HL-HELHC baseline
- IDEA, CLICDet and others available for FCC-ee
  - Starting to be extensively used, but need further validation
- Delphes version in use includes TrackCovariance, dEdx, ParticleDensity
  - Enable simulation vertexing, b-tagging, …
  - Help developing/understanding algorithms

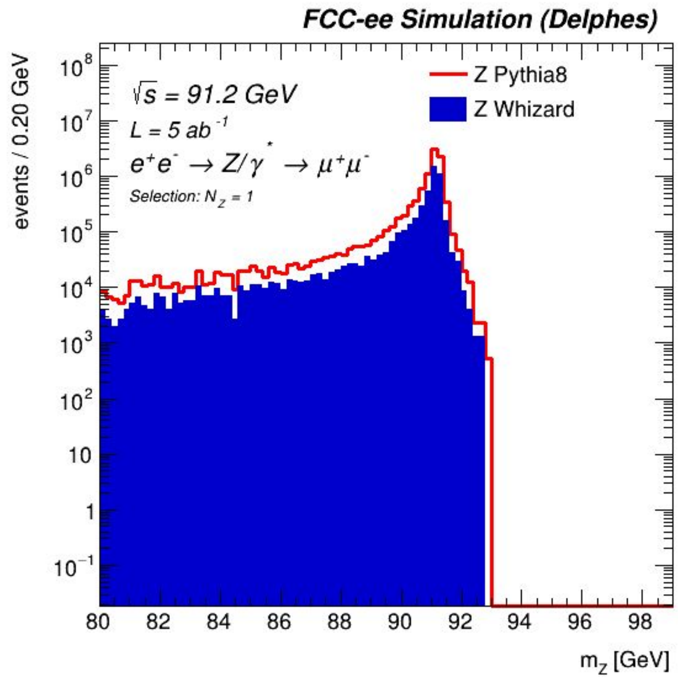
Possible contributions: testing, validation, fine tuning of existing cards; scripts or tool to easy variate relevant dimensions
Experience needed: familiarity with Delphes, Gaudi, simulation

Showcase 1: Comparing event generators

Showcase 1: event generator comparison

Example of first step of typical study

- Example of use of generators in FCCSW
- Understand meaning and effects of generator settings
- Understand differences between generators simulating the same process
- Example of use analysis tools



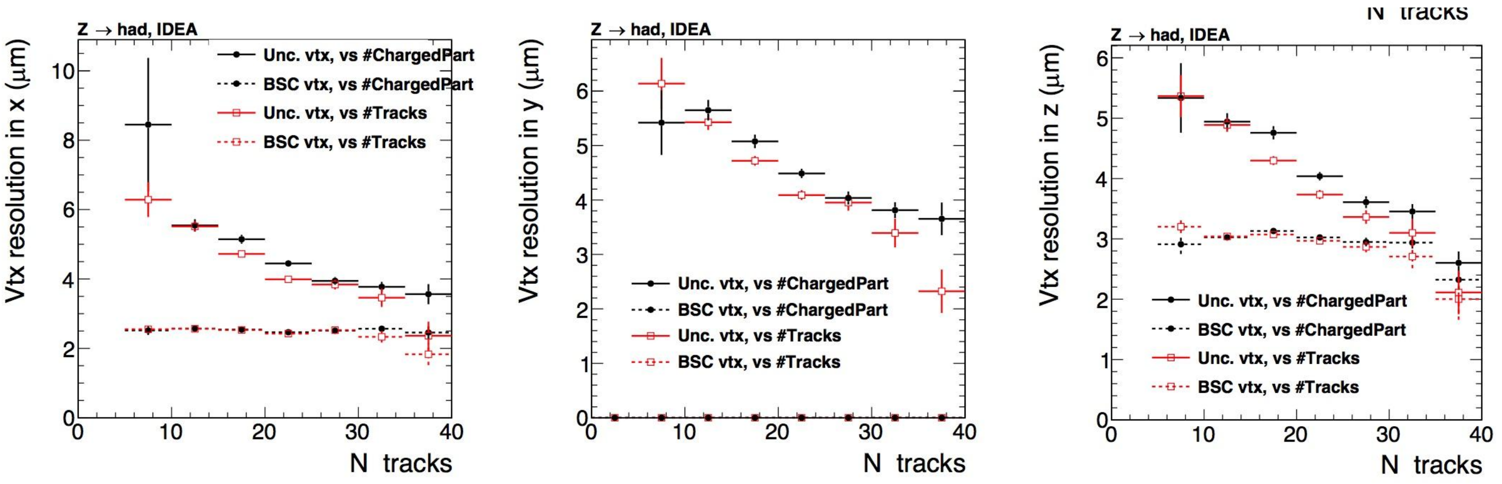
Showcase 2: Vertexing

Showcase 2: Vertexing/Flavour tagging

Example of vertexing and flavour tagging using ILC software

- Produce events in EDM4Hep format with the Delphes IDEA track covariance
- Convert the EDM4Hep outputs to LCIO with a script
- Run primary and displaced vertexing with LCFIPlus
- Run CLIC flavour tagging performance

Showcase 2: Vertexing/Flavour tagging



Primary vertex
$Z \to qq$ (q=u,d,s)
$\theta > 20°$

With Beam Spot Constraint

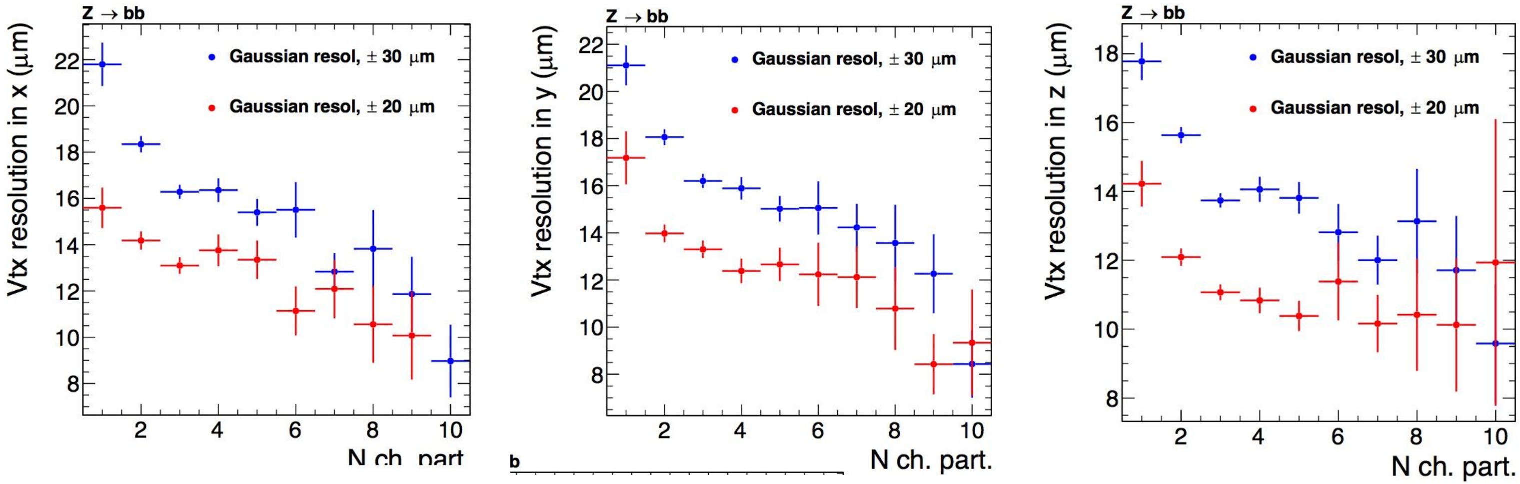
- Resolution ~independent on N_{Tracks}
- Tiny resolution in σ(y)~0.2nm
- σ(x)~2μm and σ(y)~3μm

Without Beam Spot Constraint

- Resolution in x and y are similar
- σ(x, y)≈4-5μm and σ(z)≈4μm

Showcase 2: Vertexing/Flavour tagging

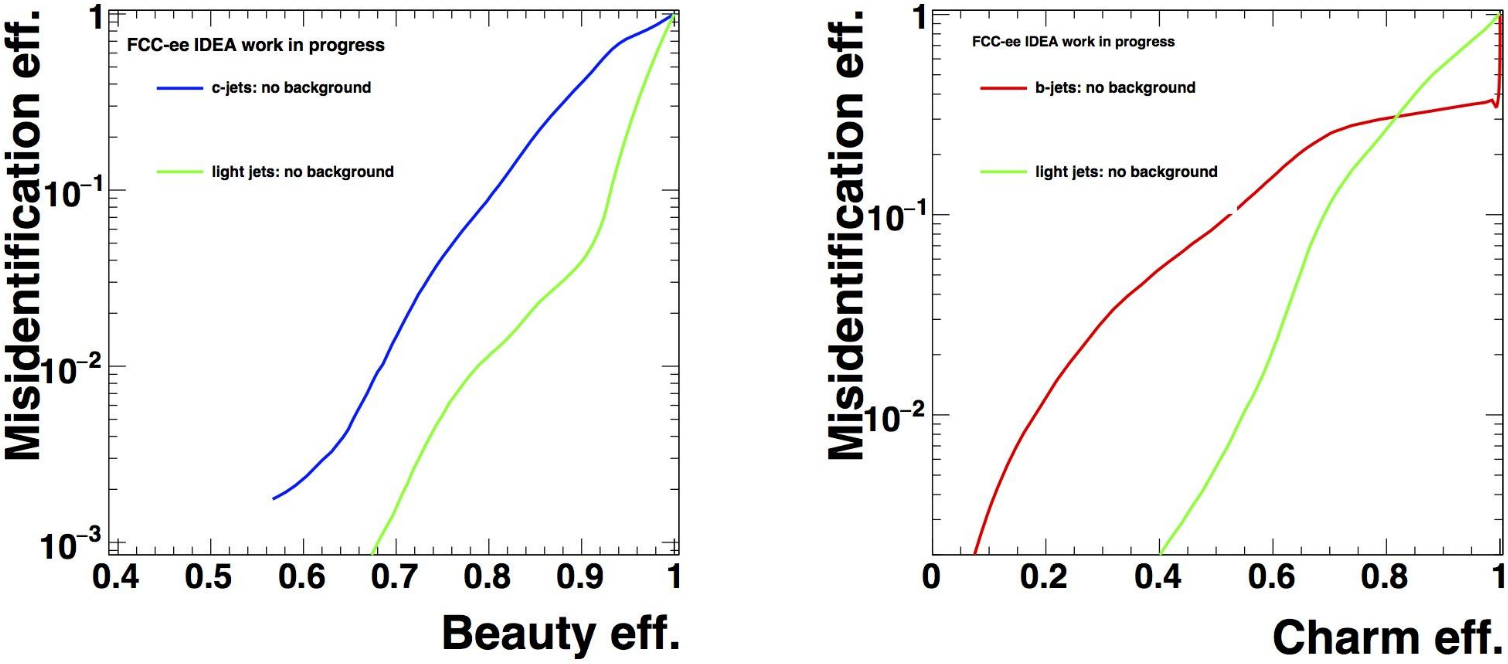
$Z \to bb$, $\theta > 20°$



Resolutions: σ(x, y)~15μm, σ(z)~12μm

Showcase 2: Vertexing/Flavour tagging

Misidentification efficiencies can be calculated as with fully simulated data



Preliminary results: need further validation especially the discriminating variables used in the BDT

Showcase 3: Flavour physics

Showcase 3: Flavour physics

- Use of EvtGen important in heavy avour studies
- Use Pythia to generate $e^+e^- \longrightarrow Z^0 \longrightarrow$ bb and to hadronise in order to make B-hadrons
- EvtGen used to decay the hadrons produced
  - DECAY.DEC used in general to decay all of the products, but
  - User can request a specific exclusive decay chain for a particle produced e.g. $B^{\mp} \longrightarrow (D^0 \longrightarrow K^{\mp}\pi^{\pm})\pi^{\mp}$
- EvtGen is now integrated in FCCSW:

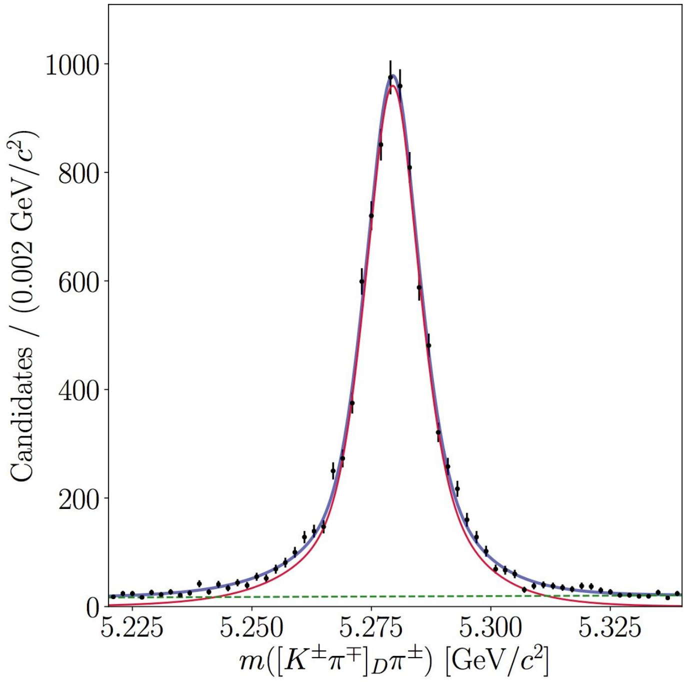
```
$ fccrun PythiaDelphes_config_IDEAtrkCov.py --Filename ee_Z_bbbar.cmd --doEvtGenDecays true \
    --UserDecayFile user_decay.dec --EvtGenDecayFile DECAY.DEC --EvtGenParticleDataFile evt.pdl -n 10000
```

- And also in Key4Hep

```
$ DelphesPythia8EvtGen_EDM4HEP  delphes.tcl  edmout.tcl  pythia.cmd   out.root   DECAY.DEC   evt.pdl
user.dec
```

Showcase 3: Flavour physics

- 10k $Z \longrightarrow b\bar{b}$ with exclusive Bu2D0Pi.dec
  - 20k b-quarks
- 43% of q-quarks hadronise to $B^{\pm}$
  - expect ~ 8, 600 $B^{\pm}$ in total
- Using FCCAnalyses to produce small ntuples
- Using ReconstructedParticles to build the D^0 and $B^{\pm}$ candidates
  - Some loss expected due to track cuts in Delphes
- Fit $m(D^0\pi)$ in exclusive sample
  - Exponential background included in the fit
  - Only additional cut: ±30 MeV $m(D^0)$ window
- Yield well aligned with expectation



Full simulation

What can we do for FCC-ee in Full Simulation?

IDEA

- Tracking
  - Basic digitisation producing space points
  - No integrated reconstruction algorithm available
    - Proof-of-concept of Hough Transform algorithm available as standalone script
- DR Calorimetry
  - DR geometry description in DD4hep being commissioned (see Sang Hyun Ku talk)
    - Optimization and performance improvement required
- LAr calorimetry
  - Description + digitization available
    - Exercise in tutorials
  - Code being reviewed and ported to EDM4hep

Challenges ahead

Software and Computing

- Framework transition to Key4hep
  - Completeness of EDM4hep: stressing the EDM at the moment with use cases
  - Availability of relevant components: k4Sim, k4Reco, ...
  - Algorithm availability and optimization
- Identify performance bottlenecks
  - Monte Carlo generators, in particular in corner cases (rare decays)
  - Full simulation components (DR calo …)
- Adequate and sustainable computing and storage infrastructure
  - Accessible and efficient meta-data, aggregation / sharing of simulated data files
  - Smooth access to distributed resources
  - Dataset reduction for efficient end user analysis
- ...

MDI-related software challenges

- Enabling estimation of reliable Machine-Detector effects
  - Identify needs in terms of multiturn, multiparticle tracking accelerator codes in particular with respect to
    - Beamstrahlung and beam-beam interactions
    - Machine imperfections and beam induced backgrounds relevant for the experiments
- Efficient and flexible interface of accelerator and experiments codes
  - Use of shared data formats

Overview of the areas of work

MC generators
Interfacing, testing, validating, optimization

Detector concepts
Geometry description, full simulation, validation, parametrization, optimization

Reconstruction algorithms
Tracking, vertexing, clustering, jet finding, particle identification, optimization

Analysis
State-of-Art (python) tools, ML, ...

Computing
Porting to other OSs, Distributed computing, ...

MDI
Shared formats, Identify relevant process and codes

Overview of the areas of work

MC generators
Interfacing, testing, validating, optimization

Detector concepts
Geometry description, full si
validation, parametrizati

algorithms
, clustering, jet finding,
entification, optimization

Analysis
State-of-Art (python) tools, ML, ...

s,
mputing,
...

MDI
Shared formats, Identify relevant process and codes

Nothing is over staffed, don't be shy please join!!

Summary

- Software is essential during any phase of a project
  - No CDR+/TDR without a robust software
  - Designing for long term usage provides stability and preserves knowledge
  - Essential role of documentation, training, for users and developers

- Current phase very challenging
  - Unprecedented level of precision expected at FCC-ee
  - Detector performance optimization requires optimal flexibility

- Try to get as much as possible from the community
  - Following closely, participate-to, collaborate-w/ common activities {Key4hep, EDM4hep}

- Every group should feel concerned
  - Immediate areas of contribution identified
  - Output of European Strategy gave momentum, let’s sustain it

Thank you!

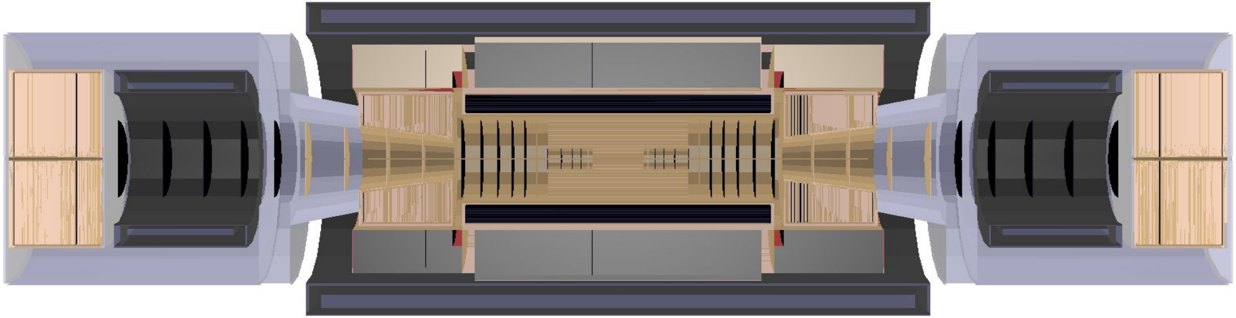
- Web site https://cern.ch/fccsw

- Tutorials / Starter Kit: ReadTheDocs

Backup

FCC detector palette in DD4hep: FCC-hh

FCC-hh CDR baseline

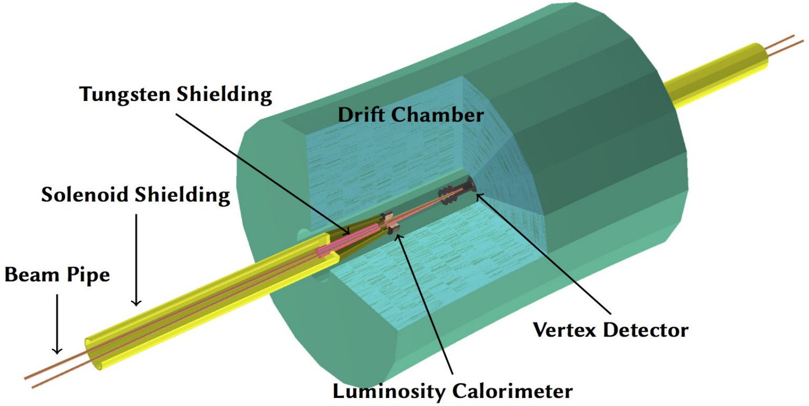


- Barrel, Endcap, Forward
- Beam Pipe, Shielding, Magnet solenoid
- Silicon Tracker
- LAr ECal, Tile HCal
- Muon System

FCC detector palette in DD4hep: FCC-ee

FCC-ee IDEA

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



DR calo full simulation available in "standalone". Integration in FCCSW/Key4hep requires:

- Translation of geometry in DD4hep format
  - Requires support for optical properties, available in DD4hep since 11/2019
- Integration of digitisation

FCC detector palette in DD4hep: FCC-ee

Possible alternatives for FCC-ee

- “IDEA” tracker 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 integration in FCCSW (digitisation modules exists in iLCSoft)

Contributions welcome/required on:
- IDEA: cross-check/complete existing stuff or provide (DR calo, muon) DD4hep descriptions and digitization
- Enabling of CLD in FCCSW: digitisation, ...

Experience needed: familiarity/willingness to learn: DD4hep, detector geometry, Geant4

Reconstruction

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

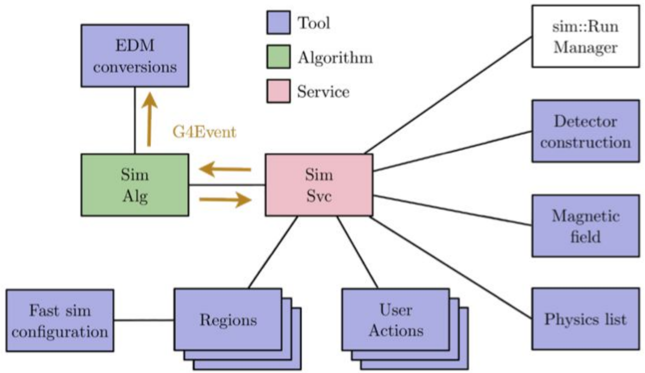
Possible contributions: tracking, vertexing, ACTS, ML, particle ID

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

Simulation

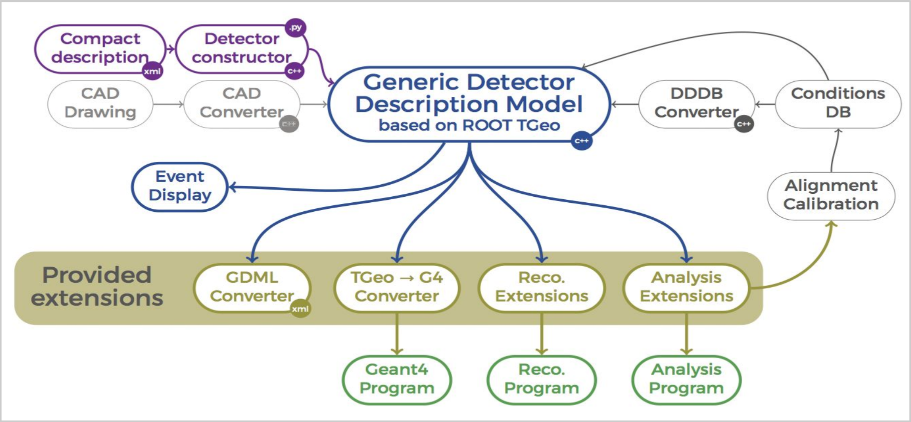
- Delphes (parametrized)
  - Gaudi interface
    - FCC EDM output

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



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



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)

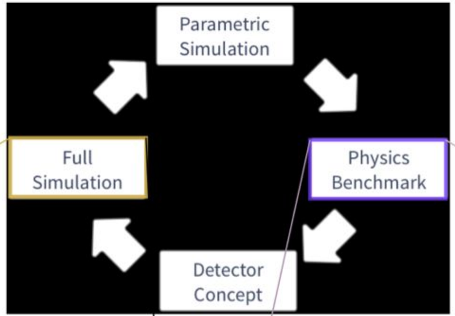
- Using the latest Gaudi version (v32r2).

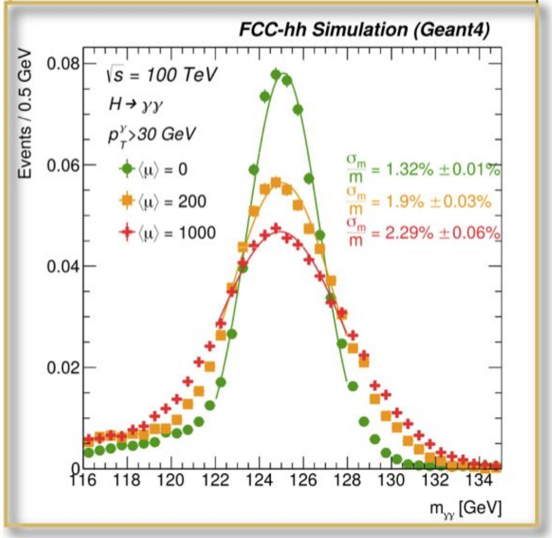
Fast / Full Simulation Interplay

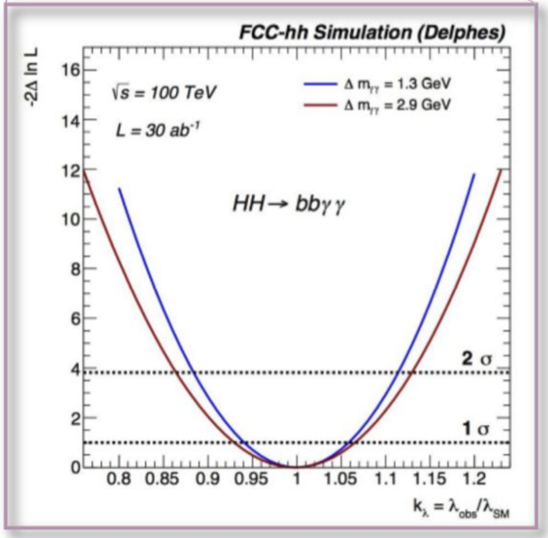
Example:

Higgs self-coupling

@ FCC-hh







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