What software for EW and Higgs Factories

FCC Week 2021

June 28, 2021

C Helsens, G Ganis
CERN-EP

Role of software in future EW and Higgs factories

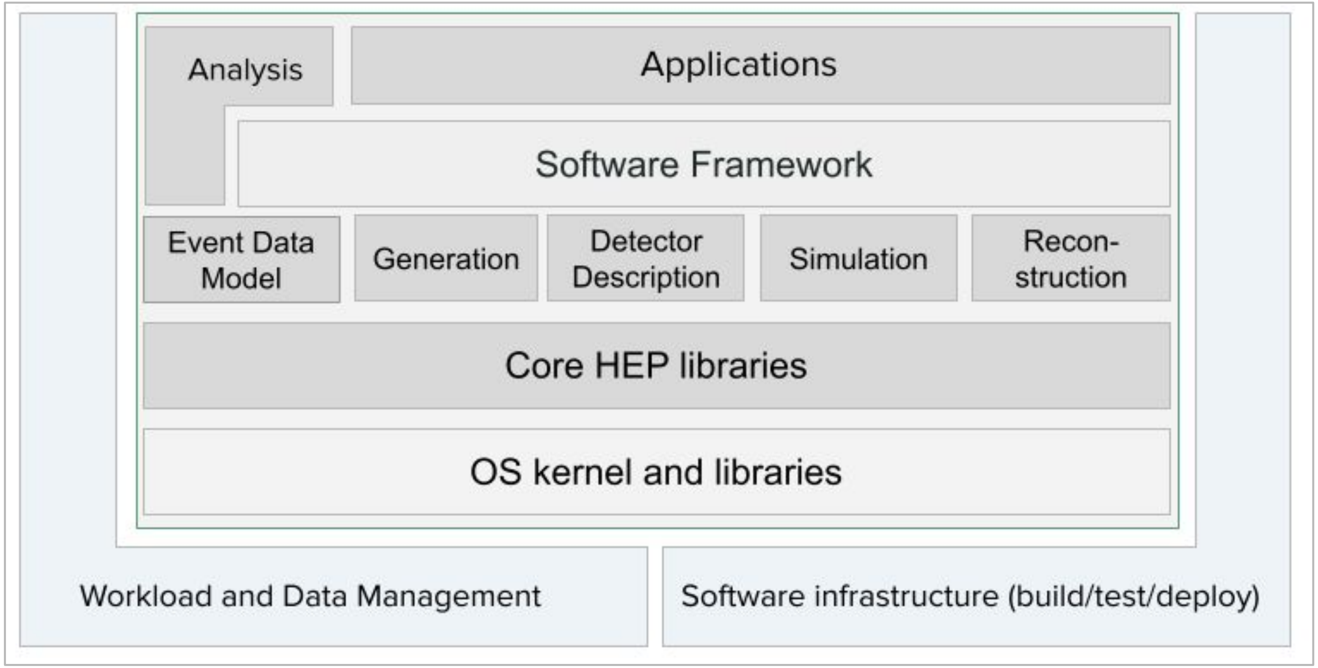
- Software is essential during any phase of an HEP project
  - No quality CDR/TDR/FSR without a robust and flexible framework
- Future EW/Higgs factories have very similar software needs
  - Support for physics and detector studies
    - Flexible detector description: open to evolution
      - Easy switch / replace sub-detectors, change dimensions, layout, …
    - Completeness: include all major aspects
      - Generation, {Parametrized, fast, full} simulation, reconstruction, analysis, MDI support, …
    - Ease of use: low usability threshold and fast learning curve
      - Extensive documentation, regular training
  - Software tools to manage computing
    - Tools to facilitate effective access to CPU / storage resources

Outline

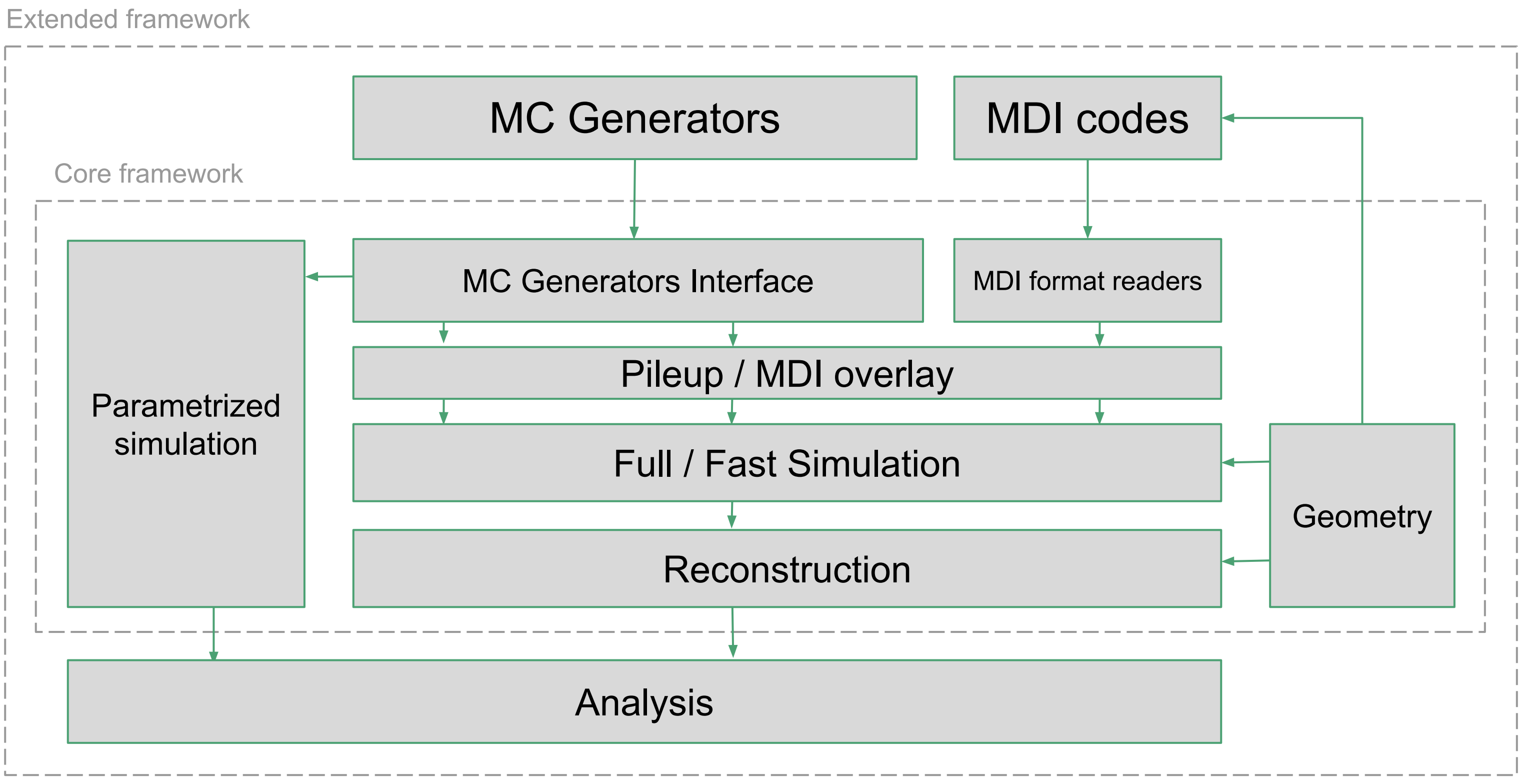
- HEP software ecosystems
  - The common software effort
- FCC-ee software potential challenges towards the next strategy update
  - Role of the LEP data

HEP Software Ecosystem

Seamless integration and optimization between various networks of devices, software and services aimed to facilitate data processing for HEP experiments



Typical workflows to support



The role of the framework

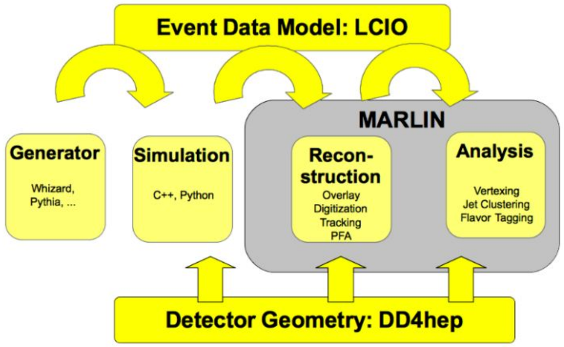
- Provide uniform view on the components
  - Common configuration, log and error reporting, plug-in management, ...
- A good framework adapts to varying landscape to always provide optimal interoperability and use of resources. Today this means:
  - Solid multi-thread support, ability to cope with heterogeneous resources in terms of
    - Different hardware (GPU, FPGA); segregation level (HPC network restrictions, ...); cloud protocols
- HEP experiments have a tendency to start from scratch
  - AliROOT, O2 (ALICE); CMSSW FWCore (CMS); Gaudi (LHCb, ATLAS, HARP, BESIII, Minerva, Daya Bay, …)
- Adopting a demonstrated solution matching the (projected) needs buys in the experience and future evolution

ILCSoft approach

F. Gaede, Bologna 2019

Brief history of iLCSoft

- in 2002 there where three LC projects in the world: Tesla, JLC and NLC
  - and about four or five different detector concepts and software frameworks
  - using C++, Java and F77 (no Python yet)
- decided to provide the basis for collaboration and common development by defining the common language, i.e. the event data model: **LCIO**
- adding **Marlin** already provided the basis for iLCSoft
- last major evolution: develop and incorporate the **DD4hep** geometry toolkit

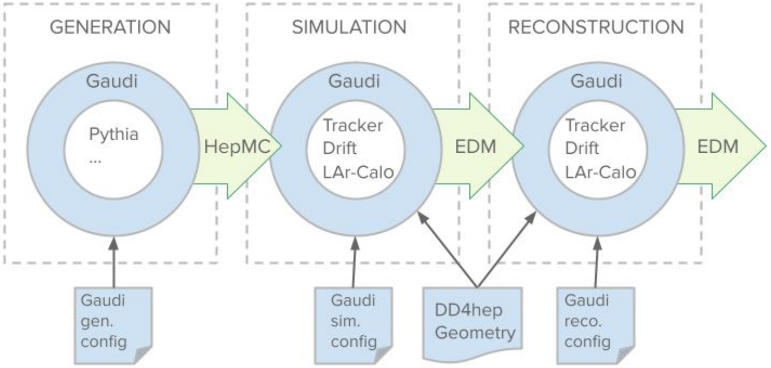


iLCsoft schema today

Design goals: keep it simple, modular, flexible, using and developing generic HEP tools whenever possible

FCCSW approach

- Started in 2014
- Driving considerations
  - One software stack to support all the cases (hh,ee,eh), all the detector concepts
  - Need to support physics and detector studies
    - Parametrised, fast and full simulation (and mixture of the three)
  - Modularity: allow for evolution
    - Component parts can be improved separately
  - Allow multi-paradigm for analysis
    - C++ and Python at the same level
- Adopted Strategy
  - Adapt existing solutions from LHC
  - Look at ongoing common R&D projects (AIDA)
  - Invest in streamlining of event data model
- Focus on FCCee after CDR (2019)



LHC as a reference?

- Requirements of running experiments, in particular LHC, in terms of software and computing, are unprecedented. Open source working solutions exist for:
  - Frameworks
  - Reconstruction techniques / algorithms
  - Machine Learning techniques development / deployment
  - Multi-Thread, heterogenous (GPU, FPGA, …) support
  - Workload and Data Management
  - Software build / packaging / test / deployment
  - Analysis tools
  - ...

New developments should focus on what is uncovered / specific to EW/Higgs factories e.g. e+e- MC, flexible geometry, specific detector technologies, reconstruction/analysis tools, ... possibly being generic and re-usable.

AIDA, AIDA2020, AIDAInnova

- Joint European effort for detector R&D
- Successful software packages (WP2@AIDA, WP3@AIDA2020)
  - Core software, Simulation
    - VecGeom: vectorized geometry
    - DD4hep: geometry description, conditions data, alignement + extensions
      - Gateway to Geant4 (DDG4), interface to reconstruction (DDrec)
    - PoDIO: EDM toolkit
  - Advanced reconstruction
    - Tracking (converged to ACTS)
    - Particle Flow (PandoraPFA)
- Approved follow-up: AIDA-Innova 2021-2025
  - Fast Simulation, Track Reconstruction (ACTS), Particle Flow (PFA), Turnkey Software Stack
  - Focus on: parallelisation, acceleration, machine learning

The common software vision

More details: F Gaede, on Wednesday

Create a software ecosystem integrating in optimal way various software components to provide a ready-to-use full-fledged data processing solution for HEP experiments

Complete set of tools for

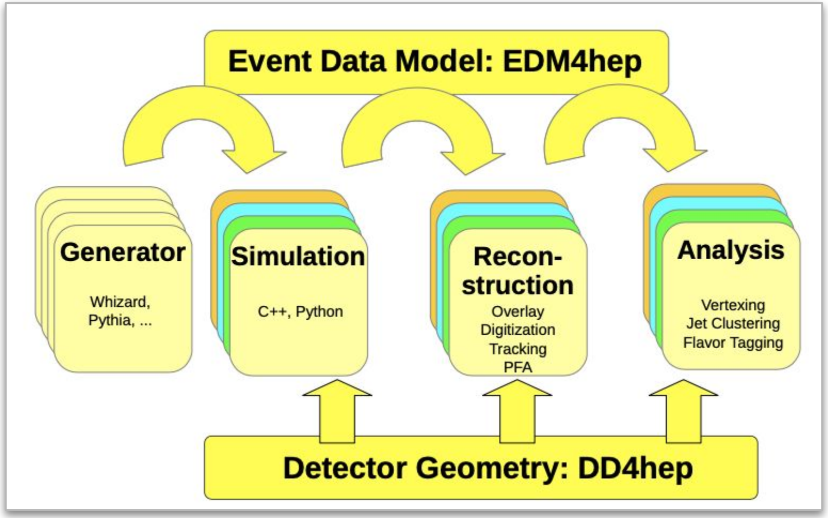
- Generation, simulation, reconstruction, analysis
- Build, package, test, deploy, run

Core Ingredients of current key4hep

- PoDIO for EDM4hep, based on LCIO and FCC-edm
- Gaudi framework, devel/used for (HL-)LHC
- DD4hep for geometry, adopted at LHC
- Spack package manager, lot of interest from LHC

Community project, unifying efforts

- Contributions from CLIC, ILC, FCC, CEPC
- And interest from STCF, muon collider, ...

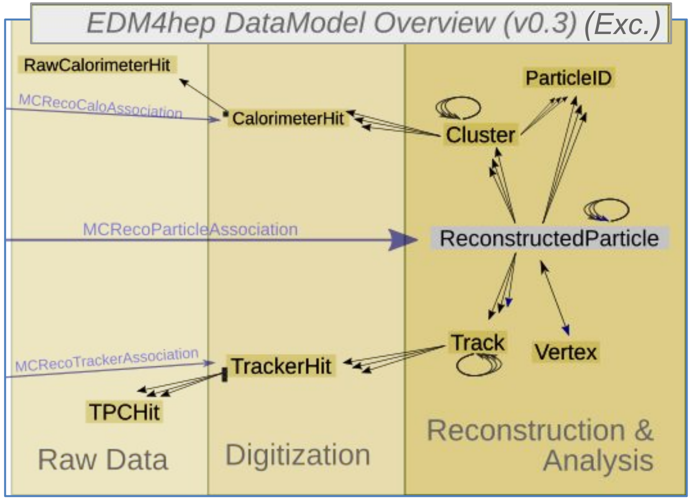


Kick-off meetings in Bologna, Hong Kong

The common event data model: the challenges

EDM provides common language for exchange among framework components

- <u>Challenge 1:</u> efficient support different collision environments (e+e-, pp, ...)
  - Positive first experiences with FCC-hh components
- <u>Challenge 2:</u> keep I/O efficient
  - PoDIO: separate definition from implementation, facilitate optimal adaptation to backend
    - POD layer designed for efficient I/O, simple memory layout
    - Flat data support (RNTuple) will provide insight
- <u>Challenge 3:</u> efficient support for schema evolution
  - Requires schema evolution in PoDIO, planned
- <u>Challenge 4:</u> efficient support for detector needs
  - Interaction w/ detector teams from the start
    - Eg. cluster counting for IDEA Drift Chamber



Key4hep adoption plans

- ILC/CLIC
  - Keep existing software chains / samples available for on-going studies
    - Enabled by k4MarlinWrapper and k4LCIOReader components
  - Full reconstruction chain through the wrapper (and EDM4hep) conversion part of the AIDA Innova work plan milestones
    - Study overheads, eventually port algorithms to Gaudi
- FCC
  - Already Gaudi based, move FCC-edm to EDM4hep
  - Re-arrangement and modernization into components considered for migration to common project
    - Generation, simulation, reconstruction, …

More details on V Volkl, on Wednesday

A few take away messages

- Designing with long term vision and awareness of existing solutions provides stability and sustainability, and preserves / optimises use of knowledge
  - Documentation and training for both users and developers is fundamental

- HEP is ready for a leap towards a common software ecosystem
  - Common tools always existed (Cernlib, PAW, ROOT, …), new general purpose tools go further and enter deep in the running experiments systems (DD4hep, Gaudi, …)
  - Efforts funneled through the Key4hep project
  - Joining efforts and optimisation of resources

- Common R&D, such as the AIDAs, CERN EP’s, ECFA’s ... have an essential role in this

The FCC software the Feasibility Study Report for the next European Strategy Update

What we need (and, next, where we are)

- Signal and background generation
  - Support for suitable set of generators and for easy addition of new ones
  - Including interfaces to the relevant codes for MDI-related backgrounds
- Detector description, simulation and reconstruction
  - Support for flexible definition/composition of detector concepts
  - Support for {full, fast} simulation and possible interplay
  - Support for a suitable set of reconstruction algorithms and for easy addition of new ones
  - Support for parametrized simulation with same output of full/fast sim. + reconstruction
- Analysis
  - User-friendly framework working on reconstruction / parametrized simulation output
    - Including library of common tools / algorithms, event display, …
- Access to appropriate workload and data management tools
- Appropriate documentation, training, tutorials

Monte Carlo Generators

- Recent review of tools for e+e- at the ECFA kick-off (W. Kilian)
- Legacy LEP generators still state of the art at the Z and WW energies
  - “Archeological” work to recover KKMC, BHLUMI, BabaYaga, KoralWW, ...
- Main software challenges
  - Interfacing with the framework
    - Typically standalone apps w/ output in common data format, e.g. HepMC, LHEF
      - In rare cases they provide a callable interface, e.g. Pythia8
  - Availability in the shared software stacks
    - Private codes, unversioned tarballs, … version control issues
- Other requirements relevant for FCCee
  - Uniform treatment of beam parameters (beam energy spread, crossing angle)
  - For generation, possibly better achieved at framework level
    - Generate at nominal $\sqrt{s}$ and zero crossing angle apply same boost before simulation

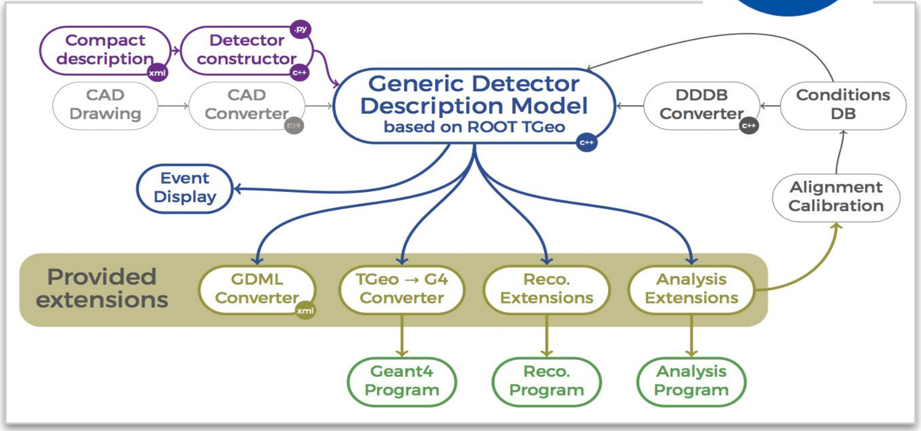
Beam and MDI-related backgrounds

- Several processes and codes, including
    - (In)coherent pair creation — GuineaPig
    - Synchtron Radiation — MDISim, SynRad, Sync_Bkg
    - Radiative bhabhas — GuineaPig, BBBrem, BHLUMI, Whizard
    - $\gamma\gamma \rightarrow$hadrons — GuineaPig + Pythia

  Codes not always in public repositories, outputs in different, non-standard formats
- Optimization of the FCCee interaction region design requires deep level of understanding of the detector backgrounds
    - Only achievable with integration in experiment software
- Framework integration to unify/simplify access of each relevant codes
    - Supercode as glue with controlled configuration and normalization; on going effort
- Consistent description of the relevant geometry elements
    - Requires interplay between detector and machine geometry formats (e.g. CAD)

Flexible detector definition - DD4hep

- Addresses the needs of HEP
  - Precision: describe the smallest element
  - Flexibility, modularity: facilitate composition of basic elements
  - Universality: single source for all needs



- De facto a standard
  - AIDA2020 project for ILC/CLIC, adopted by FCC, CEPC, Muon Collider, EIC; CMS, LHCb (run 3)
  - Geometry description with C++ detector ctors and XML (*compact*) files
    - Complex detector composed from (interchangeable) sub-detector components
- Framework integration through "geometry service"
  - Provides constructions of sensitive detectors, translation to Geant4 geometry
  - Used for simulation, reconstruction, …
- Repository of available detector implementations: FCCDetectors

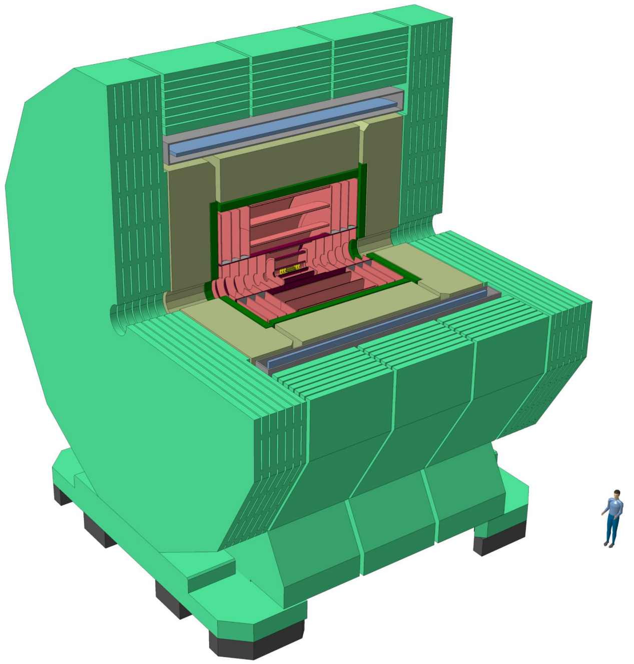
FCCee detector palette in DD4hep: CLD

Adaptation of the CLIC detector concept: arXiv 1911.12230

Nice example of CLIC/FCC collaboration

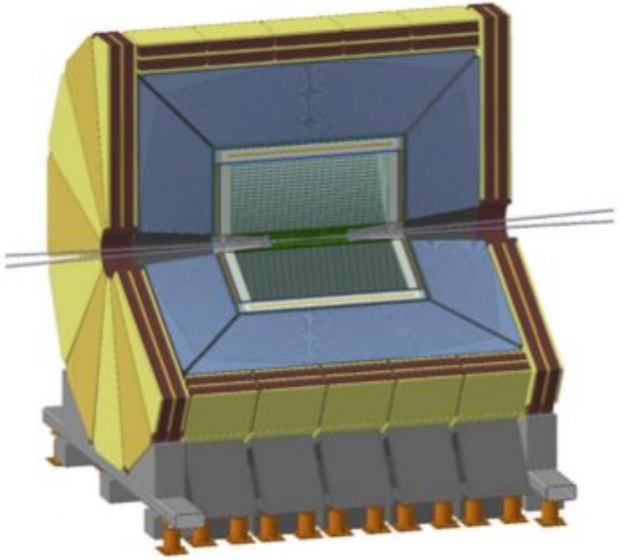
- Full silicon tracker
- High Granularity PFA calorimeter
- Solenoid outside calorimeter

- DD4hep implementations
  - Several versions in iLCSoft/LCGEO
  - Latest version in FCCDetectors



FCCee detector palette in DD4hep: IDEA

Ultra-light Drift Chamber, Dual Readout calorimeter,
Solenoid inside calorimeter



Current implementation

CLD

- Beam Pipe, Beam instrumentation
- Lumical, HOM Absorber
- Vertex detector
- Simplified Drift Chamber

Full simulation of **Vertex**, **Drift Chamber** and **Dual Readout** available in “standalone”.
Integration in FCCSW/Key4hep on the way; includes translation of geometry in DD4hep format, migration to EDM4hep and digitisation.
Work on software for **muon detectors** also started.

Completing the FCC detector palette in DD4hep

- DD4hep allows easily replacement of parts
  - Example is a reduced version of the FCChh LAr ECal + Tile HCal to be evaluated, for example, together with IDEA tracking system
    - First DD4hep description available for testing
  - Plug&Play in place replacement technology to be consolidated and streamlined
- New sub-detector concepts must be integrated in key4hep/FCCSW
  - Part of the FCC Detector Concept mandate (P.Giacomelli talk):
    - Promote the use of the common FCCSW software platform & tools, including the development of the sub-detector geometrical description, simulation, and local reconstruction;
    - Integrate sub-detectors into detector concepts: a plug-and-play technology is offered by the key4hep software framework;

Reconstruction

- Little specific to FCC-ee
  - Tracking and calorimetric algorithms for baseline FCC-hh
  - Full sim studies for FCCee not really started (using conversion to LCIO for tests)
- Lots of algorithms available for iLCSoft
  - Accessible through LCIO to/from EDM4hep on the fly converter
    - Enables initial studies and evaluations
    - Base / reference for native implementation when required
- Need to integrate algorithms attached to a given detector concept
  - E.g. IDEA Drift Chamber or Dual Readout calorimeter
- Framework integration of general purpose tools such as ACTS, PandoraPFA, CLUE/TICL, ...

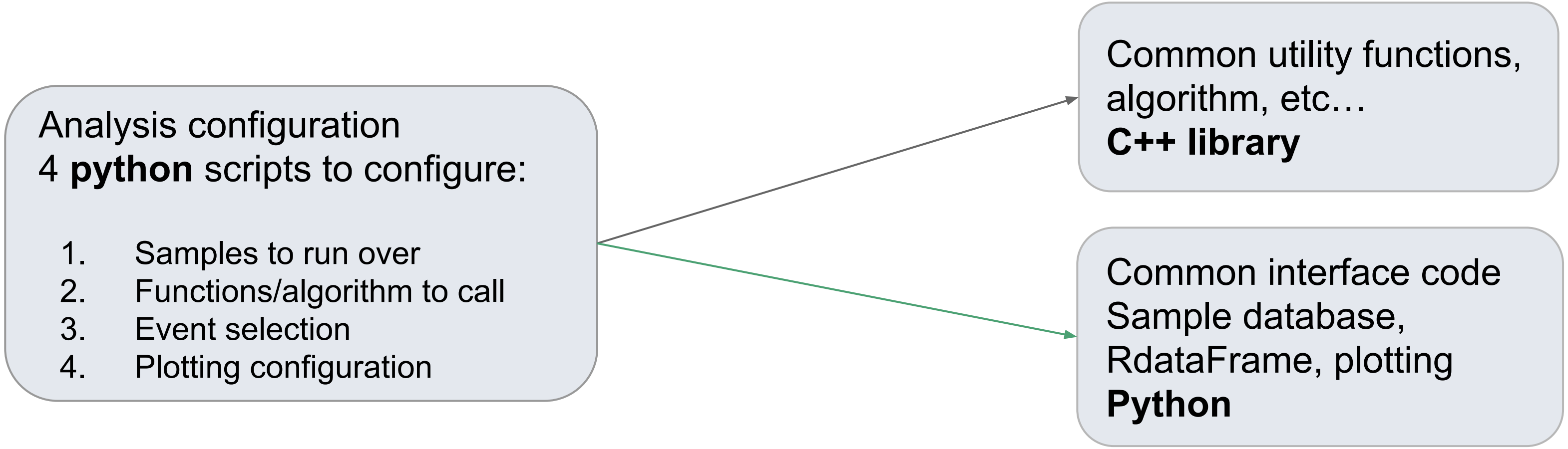
Parametrized Simulation: Delphes

- *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
    - TrackCovariance, dEdx, ParticleDensity
      - Enable realistic algorithms for vertexing, b-tagging, …
    - Also outputs observables such as isolated leptons, missing energy and collection of jets
  - Interfaced to standard file formats (e.g. Les Houches Event File or HepMC)
- Output in EDM4hep
  - Key4hep provides interfaces to Delphes producing EDM4hep output
    - Standalone executables, e.g. DelphesPythia8_EDM4HEP; Framework integration
- Palette of detector concepts for e+e- available
  - E.g. IDEA extensively used for current FCCee studies

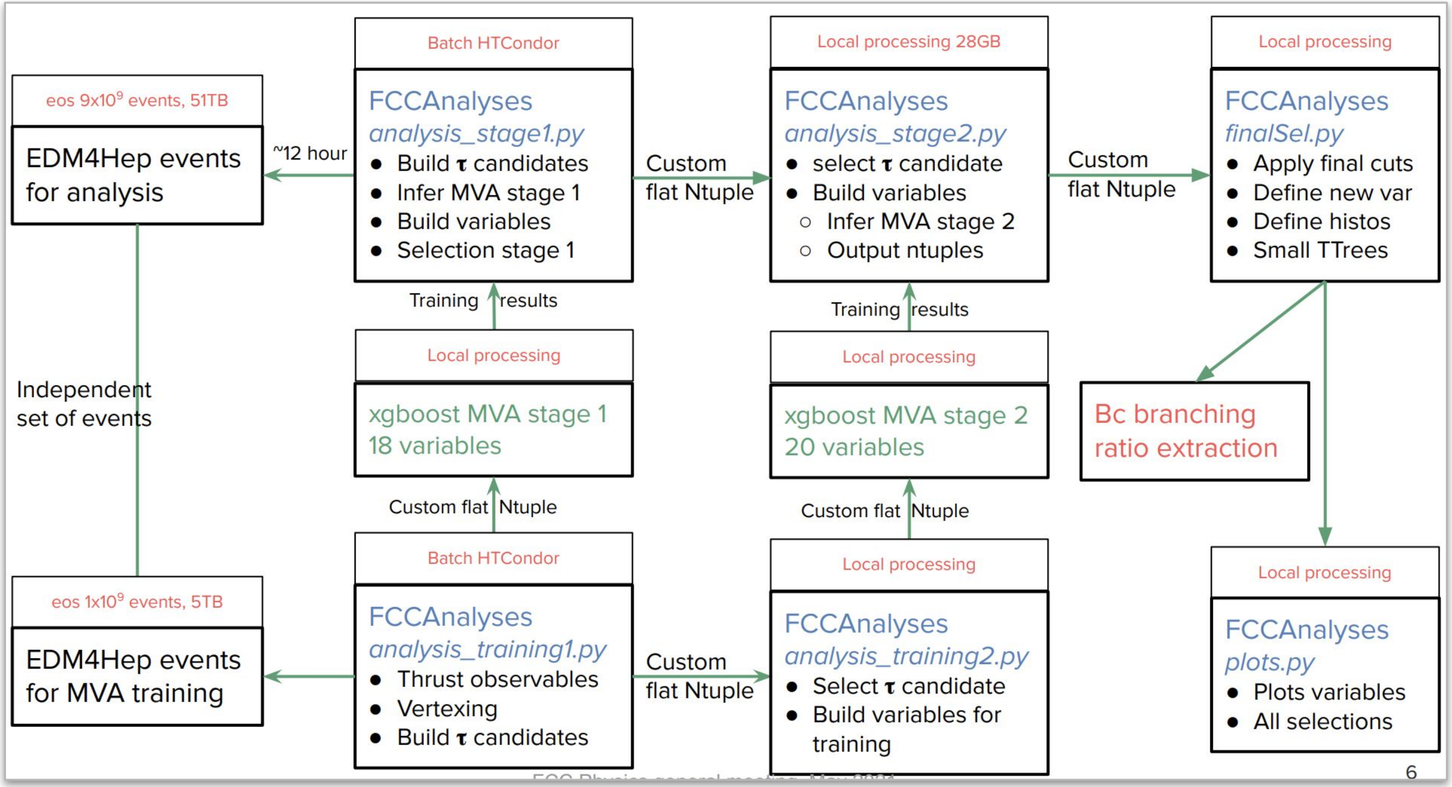
Analysis framework: FCCAnalysis

- Based on RDataFrame, new ROOT paradigm developed for (HL-)LHC
  - Python framework with C++ backend
  - Bridges the gap with LHC involved people
- Runs on EDM4hep, non FCC specific
  - Prototype of generic analysis framework

More details: C Helsens, on Wednesday



First published analysis with FCCAnalysis: $B_c \to \tau^+ \nu_\tau$



Involves:
- Distributed processing
- Storage
- ML Training
- Fitting
- Local processing
- ...

C. Helsens, FCC Physics General Meeting, 31 May 2021, arXiv 2105.13330, D. Hill's talk Tuesday, github

A few considerations on the resources needed

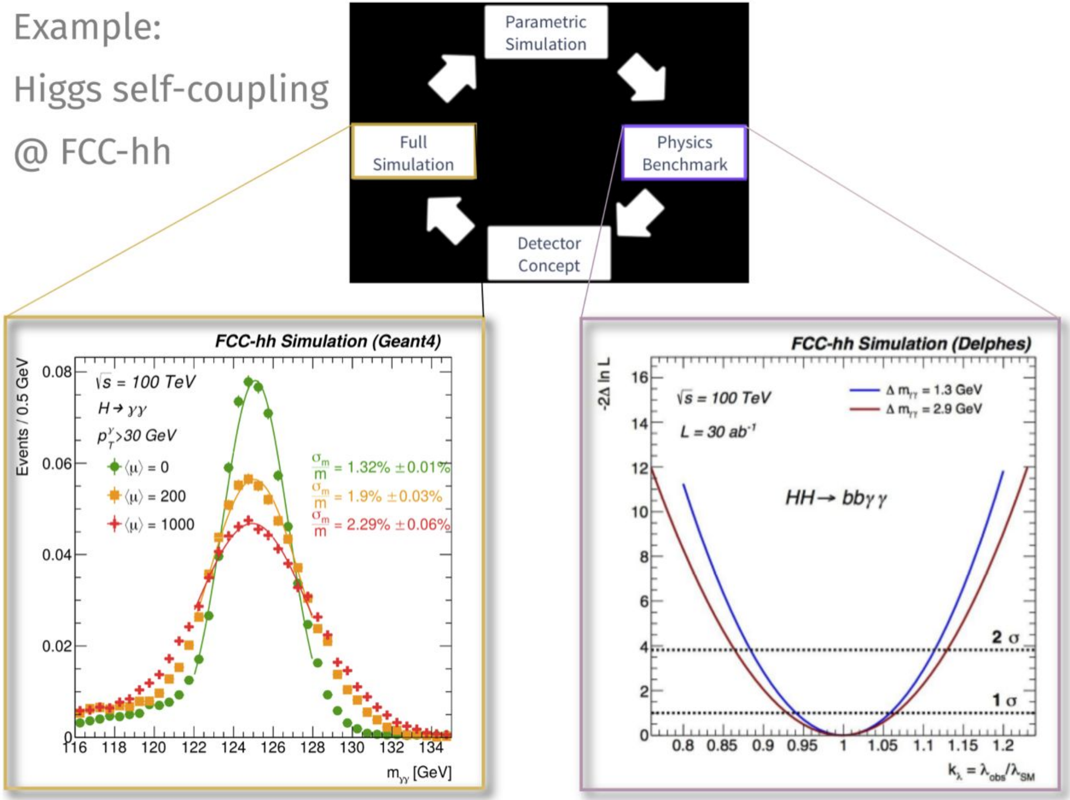
- The run at Z peak sets the scale
  - ≈10^{12} evts, 3-6 EByte storage, 10 MHS06 CPU (≈ current ATLAS yearly needs)
- These numbers are similar to the ones expected for (HL-)LHC
  - Do not expect issues for operations in 2040 and beyond
- For the FSR the situation is different
  - Analysis at Delphes level are possible (see $B_c \longrightarrow \tau^+ \nu_\tau$)
  - Full simulation of all components require 10^3-10^4 times more
- Techniques of overcome this limitations are required
  - E.g. interplay of full and parametrized simulation (see next)
- Planned community improvements in fast simulation very welcome
  - Possible improvements of the parametrized simulation treatment of critical parts such as calorimetric object could also be envisaged / investigated
    - E.g. based on improvements of fast simulation à la Geant4/GFlash or Machine Learning / GAN

CDR example of fast / full simulation interplay

Example:

Higgs self-coupling

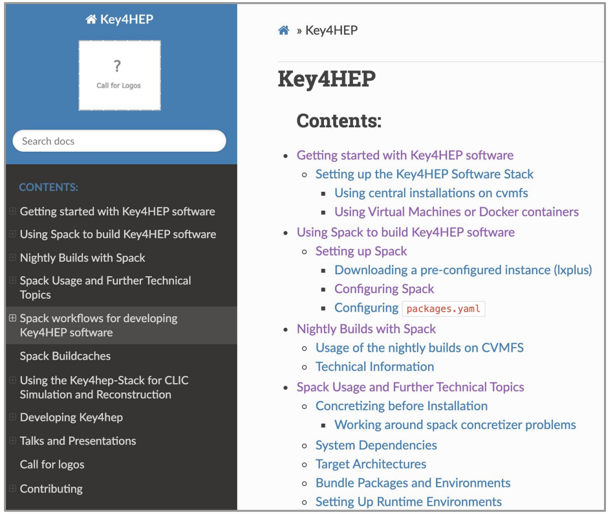
@ FCC-hh



Software infrastructure

- Workload and Data Management: DIRAC
  - Started from LHCb, used by Belle II, BES III, JUNO, … ILC/CLIC
  - LC community instance, iLCDirac, serves also CALICE and now FCC
    - Good example of re-use of generic solution
  - Provides native support for data management and interface with RUCIO
- Build / packaging / testing / deploying
  - Lots of open tools and experience available in the community
  - Using HSF recommended solutions
    - Spack for build/packaging, CernVM-FS for deployment; GitLab CI, Jenkins for continuous integration, ...

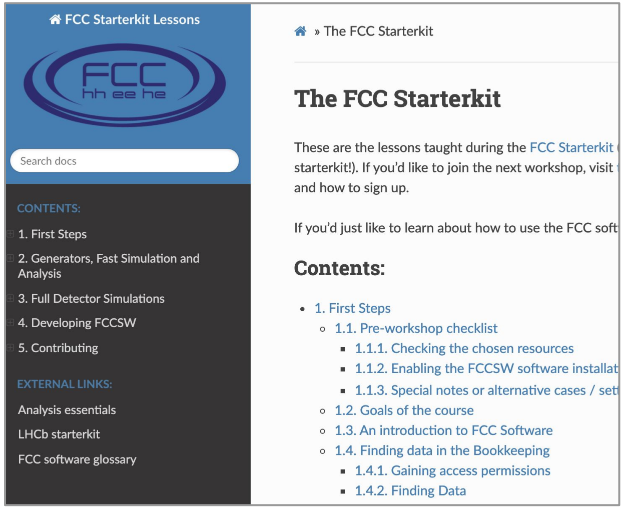
Documentation, tutorials, …



Key4hep GitHub Project

Main documentation page

Doxygen software documentation



FCCSW GitHub Project

Main documentation page

The LEP data

What can be still done with LEP data? Some examples

- Validation and Tuning of new version of hadronization tools
  - E.g. Pythia8, Herwig7, Sherpa … require genuine e+e- data sets (W. Kilian@ECFA-kickoff)
- Optimization / generalization of analysis techniques applied only partially
  - E.g. event rotation technique for acceptance determination for Z→hadrons
    - 2-3 order of magnitude more powerful statistically than straight MC
    - Can be generalized to leptons; improve LEP results
- New analysis
  - Test new ideas and/or tools coming from future experiments studies
    - E.g. Machine learning techniques for Heavy Flavour observables, tau physics
      - Tuning of algorithms and possibly improved LEP measurements (or even new, e.g. mass)
  - Confirmation/verification of (HL-)LHC discoveries / 3-sigma excess
    - Complementing w/ channels difficult at LHC (e.g. those w/ taus)
  - Complete the exploitation of LEP detectors, e.g. in searching for feebly interacting particles
    - Optimized tools may allow to include channels difficult for LHC, e.g. those w/ taus

Can we still use LEP data today? How?

- Yes: ALEPH, DELPHI, OPAL active in the DPHEP collaboration
  - L3 ntuples around for specific analysis cases
- Need approval from collaboration and access to the *preserved* software chain
  - And typically a member of the collaboration on board
  - Not uniform, requires VM or container for SLC6 (ALEPH, OPAL); DELPHI can run on lxplus
  - Requires experiment expertise
- Typical approach
  - Interesting observables extracted to text files and converted into ROOT TTrees
  - It works and detaches from experiment specifics
    - ROOT is de-facto a cross-experiment standard
  - But lacks generality
- Can this be generalized?

Why LEP data @ EDM4hep?

- Key4hep / EDM4hep: framework with longer perspective than a single experiment
  - Not just *another data format*, but one that might become a standard
    - For data structures and file format
      - ROOT (compatible) default today, technology evolve as needed
- Requires “migration”, which may be a pain
  - But return-on-investment may be large
- Is it feasible?
  - Idea is not to re-code but do minimal modifications to the preserved experiment software to
    - Automatically extract data from native format to EDM4hep
    - Interface new(er) Monte Carlo for new productions using *preserved* software virtual machines
  - First inspection of the ALEPH case promising; need to investigate DELPHI and OPAL

What LEP data @ EDM4hep could bring to FCCee?

- A great value!
  - Allow the same algorithms and techniques to be applied on simulated and real data
  - Give more realistic evaluations of the techniques under study
  - Provide a powerful training basis to the youngsters
  - Opportunity for future (EW/Higgs factories) students for thesis / publications with real data
    - Essential to get a job in HEP!
  - *The best Monte Carlo is the data!* (J.F. Grivaz, 1989)
- Several LEP *experts* involved in EW/Higgs factories studies
  - Unique opportunity
    - For future factories to get access to still useful data
    - For LEP data to get better preservation perspectives
- Dedicated DPHEP task force at work to investigate the details and estimate the needs in terms of resources

Concluding remarks

- Current direction in software is towards strengthening the common efforts
  - HEP is ready for a leap towards a common software ecosystem
  - Common R&D (AIDAs, CERN EP's, ECFA's ...) have an essential role in this
  - Future collider community engaged around Key4hep and FCC is part and parcel of this
  - Documentation and training for both users and developers is essential

- FCC software requirements towards the next strategy update defined
  - Main challenge is to get realistic performance estimations with limited resources
  - The (FCC) community should all feel concerned and get involved
    - Effort needed to provide dedicated workforce and preserve current expertise
    - The best effort approach may not be sufficient

- The LEP data can still have a role for future EW and Higgs factories
  - Effort should be made to integrated them in the common software effort

Thank you!