

Software & Computing for PED studies

Overview of the FCC-ee Software Chain

1st US FCC Workshop, BNL, Upton, NY, USA

Apr 24, 2023 G Ganis, CERN-EP

This project is supported from the European Union's Horizon 2020 research and innovation programme under grant agreement No 951754.



Outline



- Introduction
 - FCC Feasibility Study, Key4hep initiative, FCC S&C role within FCC PED
- Workflows to support
- Key4hep
 - Framework, event data model, geometry description
- Components
 - Generators, MDI
 - Parametrized simulation
 - Geometry
 - Full simulation, reconstruction
 - Analysis
 - Infrastructure for build/test/deploy and run
- Opportunities to contribute

Outline



- Introduction
 - FCC Feasibility Study, Key4hep initiative, FCC S&C role within FCC PED
- Workflows to support
- Key4hep
 - Framework, event data model, geometry description
- Components
 - Generators, MDI
 - Parametrized simulation M Selvaggi, 11h30
 - Geometry

 - Analysis
 - Infrastructure for build/test/deploy and run
- Opportunities to contribute

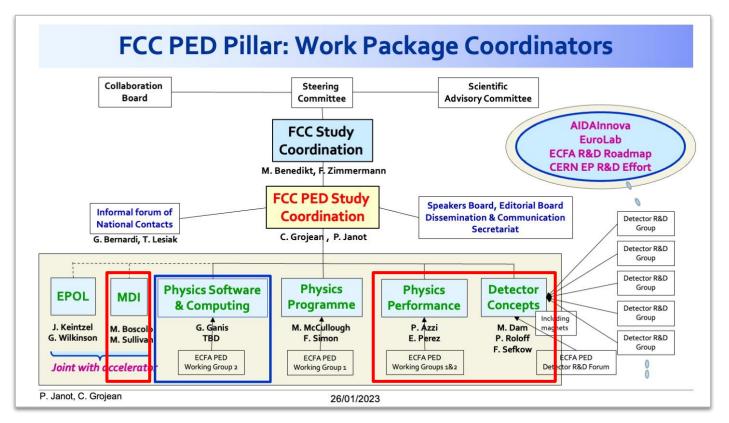
Introduction



- FCC Feasibility Study (2021-2025)
 - One of the main recommendations of the 2020 Update of the European Strategy for Particle Physics (M Benedikt)
 - Mostly about the machine, but also Physics, Experiment, Detectors or PED
 - PED studies require a solid software ecosystem
- FCC initial phase (2014 input to ESPPU 2019)
 - Developed FCCSW based on Gaudi, DD4hep, Geant4, Delphes; mostly used for FCC-hh
- Key4hep (2019- ...)
 - Initiative for a common software ecosystem future HEP projects
 - Joined by FCC from the beginning, way to mitigate lack of workforce
 - Reinforced engagement when Feasibility Study started
 - Certified in FCC S&C reviewed mandate

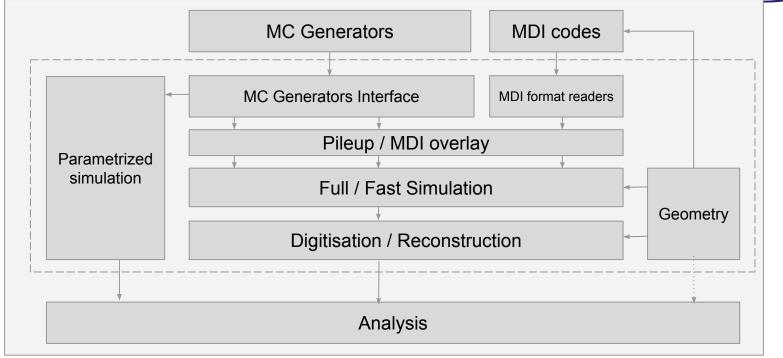
S&C relation with other groups





Workflows to support





Software Infrastructure (Repositories, Build/Test/Deploy)

Key4hep, the common software vision



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

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

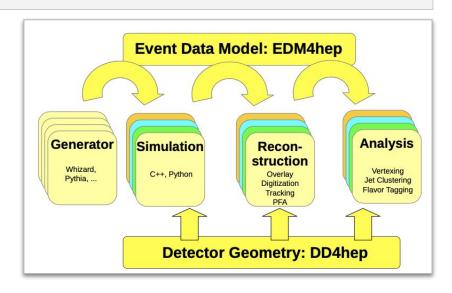
Initial Common Core ingredients

- 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

Community project

- Contributions from CLIC, ILC, FCC, CEPC and EIC
- Unifying communities, synergetic enterprise, possible testbed for common solutions

Full support by ECFA, AIDA, CERN EP R&D



Kick-off meetings <u>Bologna</u> (6/2019), <u>Hong Kong</u> (1/2020) <u>Weekly working meetings</u> Deliverables already used in large scale production

Key4hep initial choices



Framework

- Gaudi sounded as a sort of obliged choice
 - Flexible, used at LHC, hence expected to cope with increasing needs
 - Experiment-free (already adopted by several projects, included FCC-hh for CDR-2018)

Event Data Model

- Generate with PODIO, AIDA project
 - Decouple definition (YAML files) from back-end, which can be optimised
 - Already used by FCC-hh, prototyped for LC

Geometry

- Use **DD4hep**, another AIDA project
 - Meta-framework with description in XML, interface particle transport framework (Geant4, ...)
 - Provides single source of geometry for all usages
 - Focus on detector aggregation, relevant in this phase of the project

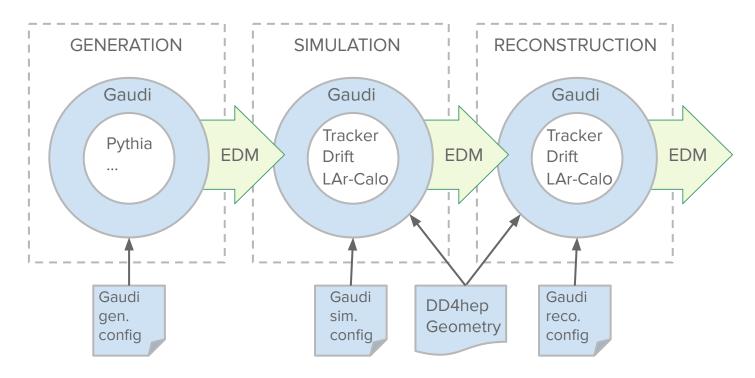
Software Framework: Gaudi-based



- 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 (v36r10).

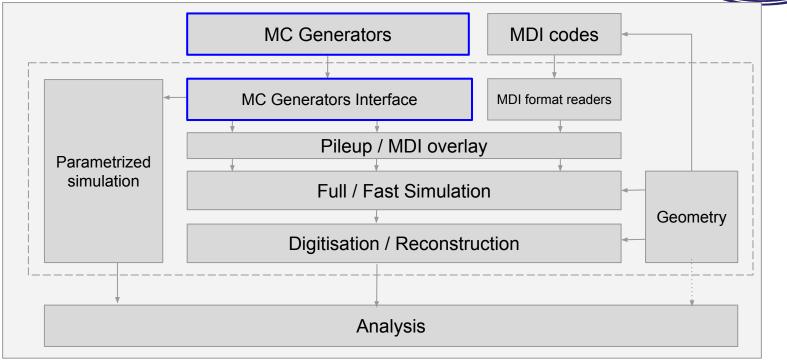
Gaudi and Key4hep





Workflows to support





Software Infrastructure (Repositories, Build/Test/Deploy)

Monte Carlo Generators @ key4hep



- All relevant generators available, including LEP ones (still state-of-art in come cases)
 - Latest version of KKMC (v5) available w/ several improvements (<u>S. Jadach, FCCW 2022</u>)
- Repository of ready-to-use codes
 - (Software-wise) sanitized: build, tested (with provide test suite)
- Interfaces to key4hep processing chain (k4Gen)
 - Readers for relevant data formats
 - Set of auxiliary tools related to MC particles (filters, smearing, particle guns, ...)
 - Possible additions:
 - Tools for crossing-angle, event mixing, event overlap
 - Generic treatment of Beam Energy Spread (BES),
 packet length, ... for generators not providing them

Demo this afternoon

Connections w/

- Physics Programme
- Physics Performance
- Phenomenologists
- Key4hep
- ECFA

List of generators currently available in key4hep



Generators

```
babayaga*1
                  baurmc<sup>†</sup>
                                         bhlumi*†
                                                         crmc<sup>†</sup>
                                                                           evtgen
                                                                                               genie<sup>†</sup>
gosam<sup>†</sup>
                  guinea-pig*1
                                         herwig3
                                                         herwigpp<sup>†</sup>
                                                                           kkmcee*
                                                                                               madgraph5amc
photos
                  pythia6<sup>†</sup>
                                         pythia8
                                                         sherpa
                                                                           starlight<sup>†</sup>
                                                                                               superchic<sup>†</sup>
tauola†
                                         whizard
                  vbfnlo
```

"Generator tools"

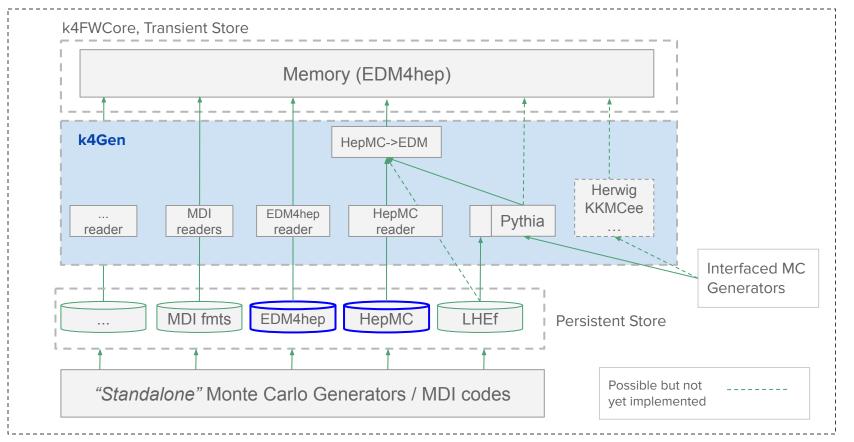
```
agile<sup>†</sup>
                      alpgen<sup>†</sup>
                                             ampt<sup>†</sup>
                                                                      apfel<sup>†</sup>
                                                                                                ccs-qcd<sup>†</sup>
                                                                                                                     chaplin<sup>†</sup>
collier<sup>†</sup>
                      cuba†
                                             dire<sup>†</sup>
                                                                      feynhiggs<sup>†</sup>
                                                                                                form<sup>†</sup>
                                                                                                                     hepmc
                      heppdt
                                             hoppet<sup>†</sup>
                                                                      hztool†
                                                                                                 lhapdf
                                                                                                                     lhapdfsets<sup>†</sup>
hepmc3
                                                                                                 qdt
                                                                                                                     qgraf<sup>†</sup>
looptools
                      openloops
                                             professor<sup>†</sup>
                                                                      prophecy4f<sup>†</sup>
recola<sup>†</sup>
                      rivet
                                             syscalc<sup>†</sup>
                                                                      thepeg
                                                                                                 unigen<sup>†</sup>
                                                                                                                     yoda
```

Currently the latest version of each package is installed in Key4hep stack

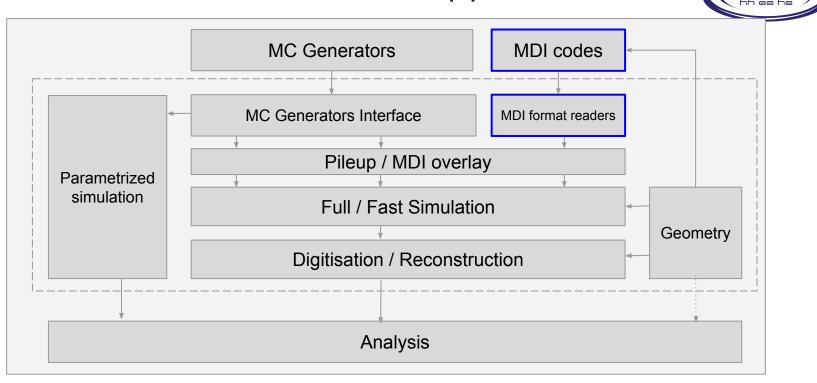
T Mandlener, Oct 2022

Managing interoperability in Gaudi





Reminder of the workflows to support



Software Infrastructure (Repositories, Build/Test/Deploy)

Beam and MDI-related backgrounds



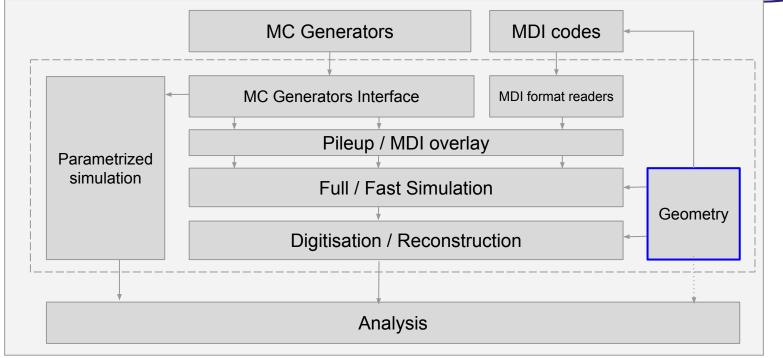
- FCC-ee particular interaction region combined with high statistical precision requires deep level of understanding of the detector backgrounds
 - Only achievable with integration of accelerator-related codes in experiment software
- Several codes for modeling the processes (<u>EPJ+ contribution</u>)
 - Not always in public repositories, outputs in different, non-standard formats
- Supercode gluing relevant codes more difficult to get than foreseen
 - Difficult access to codes, difficult to run them (exception is GuineaPig)
 - Still need code owners to produce list of particles for a given configuration
 - Injection in the framework standardised w/ the help of dedicated readers
- Consistent description of the relevant geometry elements
 - Requires interplay between detector and machine geometry formats (e.g. CAD)

Connections w/

- MDI study group, Physics Performance, Detectors

Workflows to support

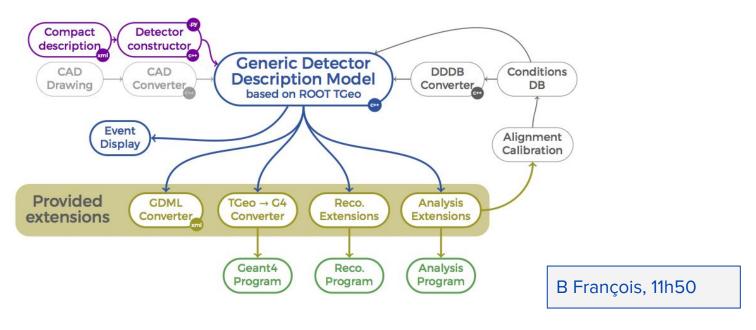




Software Infrastructure (Repositories, Build/Test/Deploy)

DD4hep - Detector Description



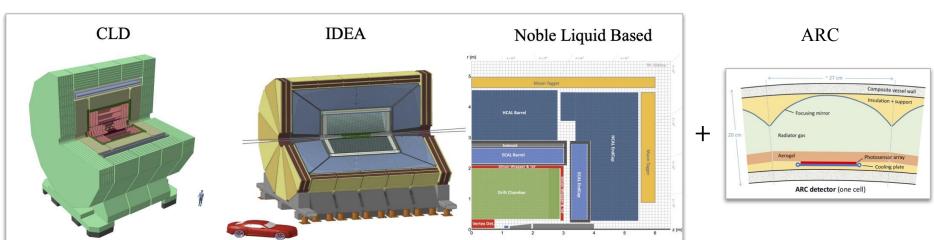


- Complete detector description: geometry, conditions, alignment, ...
- Used by CLIC, ILC, CEPC, FCC, EIC, CMS, LHCb, ...

k4geo: common repository for detector models



- DD4hep detector models spread in several project repositories
 - <u>iLCSoft/lcgeo</u>, <u>FCCDetectors</u>, <u>CEPCSW/Detector</u>
- k4geo is an attempt to reduce risk of duplications
 - E.g. CLD will only appear in one place
- Current detector concepts for FCC-ee



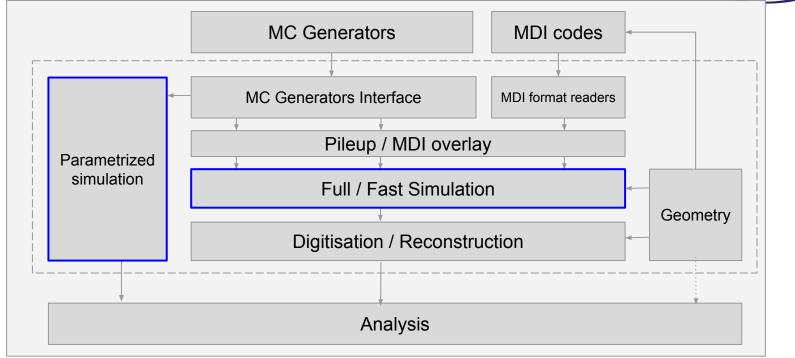
DD4hep considerations



- Flexible and powerful tool
 - Adoption by LHC, CMS and LHCb, supports the choice
- However
 - Entry threshold quite high
 - Difficult for beginners, even with Geant4 experience
 - LC community addressed this with ad hoc in-site (at CERN) induction training
 - Doable but not very sustainable for our core group
 - FCC software core is prototyping beginners tutorials w/ dedicated documentation
 - In close contact with DD4hep developers
 - Under discussion the possibility to make this as official as needed

Workflows to support

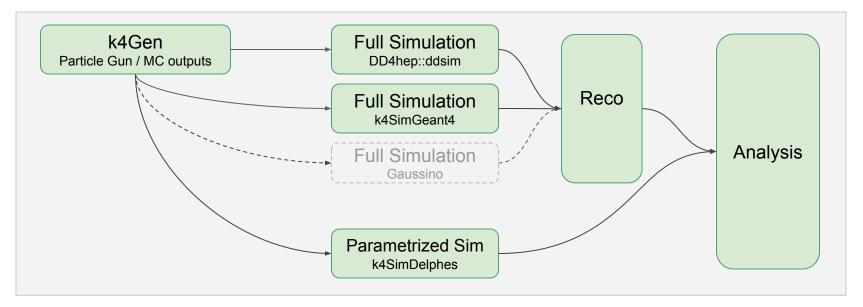




Software Infrastructure (Repositories, Build/Test/Deploy)

Integrating simulation tools

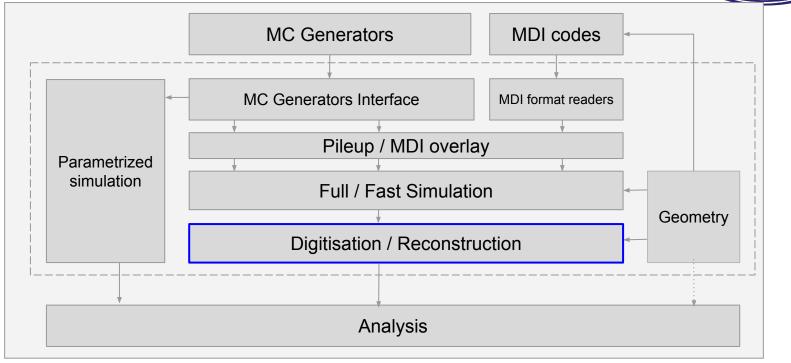




- Can be run in standalone on the MC output or as a framework
 - Particle guns available at several levels
- In all cases the following steps (reconstruction, analysis) are the same

Workflows to support



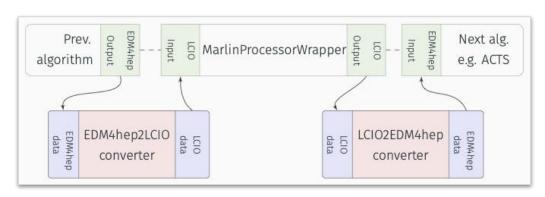


Software Infrastructure (Repositories, Build/Test/Deploy)

Reconstruction



• For CLD, CLICdp algorithms from iLCSoft available through k4MarlinWrapper



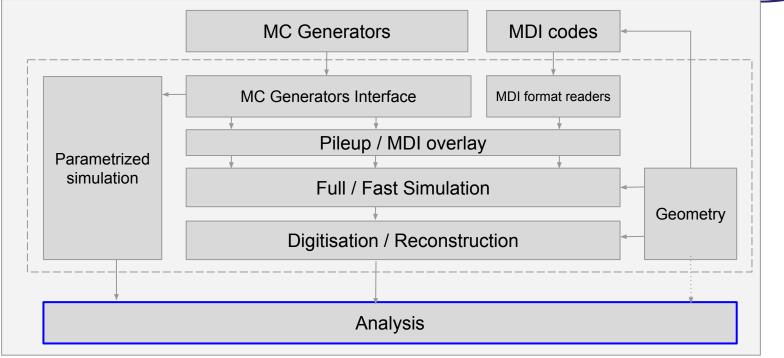
Possibly the area where there is more room for contribution

- Calo reconstruction algorithms in Gaudi
 - LAr, derived form FCC-hh LAr (see tutorial), IDEA Dual Readout
- Particle Flow: PandoraPFA available through wrapper, native implementation in the pipeline
- Other
 - CLUE, developed for CMS HGCAL, available as k4CLUE
 - ACTS, tracking in high density scenario, started by ATLAS; in the pipeline

Very useful <u>1st ECFA workshop on Reconstruction</u> 2nd coming on July 11-12 2023

Workflows to support





Software Infrastructure (Repositories, Build/Test/Deploy)

Analysis: FCCAnalysis



- Provides relevant code to interpret EDM4hep files
 - Required despite the information being stored in Plain Old Data
- Based on RDataFrame, 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
 - Possible inclusion in key4hep under discussion

Analysis configuration 4 **python** scripts to configure:

- 1. Samples to run over
- 2. Functions/algorithm to call
- Event selection
- 4. Plotting configuration

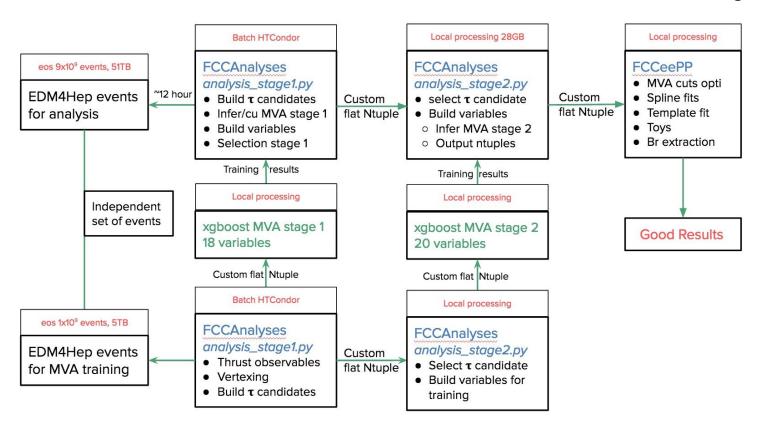
Connections w/

- Physics Performance
- Key4hep

Common utility functions, algorithm, etc... **C++ library**

Common interface code Sample database, RdataFrame, plotting **Python**

First <u>published</u> analysis with FCCAnalysis: $B_c \rightarrow \tau^+ \nu_{\tau}$

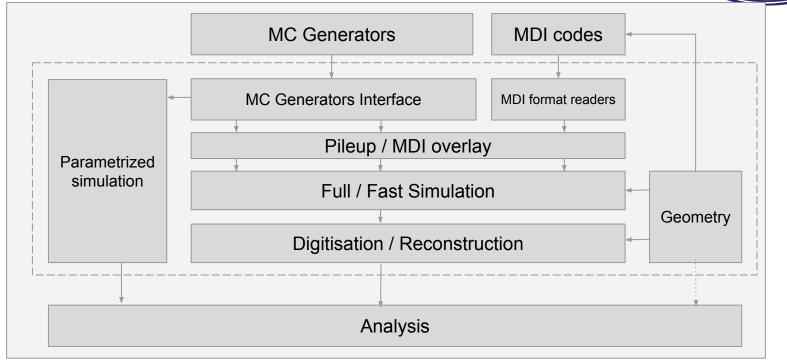


Involves:

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

Workflows to support





Software Infrastructure (Build/Test/Deploy)

Resource needs for the FSR



• The run at Z peak sets the scale

See EPJ+ contribution

- \circ ≈10¹² 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 $B_c \rightarrow \tau^{\pm} v_{\tau}$)
 - Full simulation of all components require 10³-10⁴ times more
- Techniques of overcome this limitations are required
 - E.g. interplay of full and parametrized simulation
- Need to understand
 - Role of improvements in simulation of EM calorimetry
 - New simulation accelerating techniques
 - Role of fast simulation GFlash or Machine Learning / GAN

Event Producer Workload / Data Management



- Currently still using home-made solution (HTCondor, EOS, ...)
 - Served well CERN-based productions (CDR, Spring 2021)
- Inclusion of non-CERN resources desired
 - Main requirements: central file catalogue, replication, remote access
 - Major development for the in-house system
- iLCDirac: LC community DIRAC instance
 - Workload management, file catalogue used by LHCb, Belle II, BES III, JUNO, ILC/CLIC, ...
 - Already serving another VO (CALICE)
- FCC @ iLCDirac
 - Re-activated FCC VO
 - Associated CERN FCC resources to FCC VO (HTCondor, EOS area)
 - Added steering applications of interest for FCC workflows
 - Storage organisation based on LC and LHCb experience

Connections w/

- Key4hep
- Physics Performance

FCC @ iLCDirac : adding resources



- Grid model
 - Sites can decide to add resources to the VO
 - CPU + storage, storage only

lxplus:~\$ source /cvmfs/clicdp.cern.ch/DIRAC/bashrc

- First integration of external site: CNAF (storage only)
 - Can replicate files between CERN-EOS ad CNAF
 - Common catalogue view, Remote access enabled

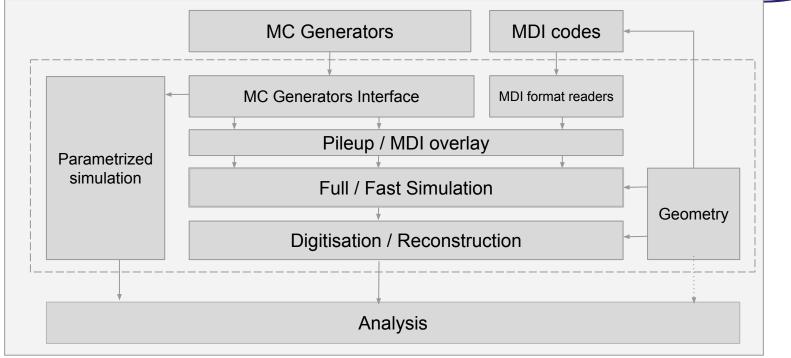
Enable sharing of public productions through optimisation of computing resources

(TFile *) 0x2ec28b0

root [1]

Workflows to support





Software Infrastructure (Repositories, Build/Test/Deploy)

Repository/build/test/deploy infrastructure



- Repositories on GitHub
 - Projects key4hep, HEP-FCC
- Managed with Spack
 - Dedicated add-on key4hep-spack
 - CI based on GitHub actions
- Deployed in dedicated CernVM-FS
 - Setup with /cvmfs/sw.hsf.org/key4hep/setup.sh
- Official build are for CentOS7, Gcc 11
 - Builds for other platforms in preparation
 - In principle works for newer OSes, such as CentOS8, AlmaLinux9 or Fedora37 because of glibc backward compatibility (less core aspects, such graphics, might still be OS specific)
- Legacy build/deploying infrastructure (extension of LCG builds)
 - Used for quick tests, but being phase-out

Some remarks



- Currently still CERN-centric
- VM, based on CernVM, available to recreate equivalent environment
 - Works from everywhere but speeds depends on the network
 - Documentation available
 - Unfortunately does not work for Apple M1

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

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



Opportunities to contribute

Based on the current status of things and needs as described on the dedicated sessions of Krakow FCC PED workshop and follow-ups

Technical tasks examples



- 1. [R] Validate iLCSoft reconstruction algorithms for FCCee
 - Validate relevant algorithms using k4MarlinWrapper (MarlinTrk, LCFIplus, ...) to understand possible performance bottlenecks requiring native porting.

 Start form CLD and extend to other detector concepts as they become available
- 2. [R] Pandora integration in Key4hep
 - Make the k4Pandora algorithm use a DD4hep geometry service and validate it with CLD in preparation for usage with the other detector concepts (IDEA, LAr, ...)
- 3. [R] ACTS integration: CLD and IDEA track reconstruction with ACTS in Key4HEP Implement ACTS interface in k4ActsTracking and test/validate it with CLD first, and extend to IDEA Drift Chamber
- 4. [S] Investigate use of Geant4 fast simulation for FCCee detector concepts
 - There are several improvements in fast simulation, using classic (parametrization) and modern (ML) techniques which might become interesting for FCCee detector concepts

[R] Reconstruction, [G] Generators interfaces, [S] Simulation, [D] Digitisation, [M] MDI

Technical tasks examples



- 5. [S] Optimise speed-up of full simulation through ddsim and k4SimGeant4

 Make sure that all options are used (e.g. multi-thread execution) for an optimal use of the available resources.
- 6. [M] Investigate ways to automatically translate CAD to DD4hep formats
 Current readers are inefficient (tesselated) and not including material information
- 7. [G] Consolidation and maintenance of k4Gen

 Make sure that tools for crossing angle, energy spread are validated and can be used for generators not implementing the functionality. Make sure that all required readers are available and functional.

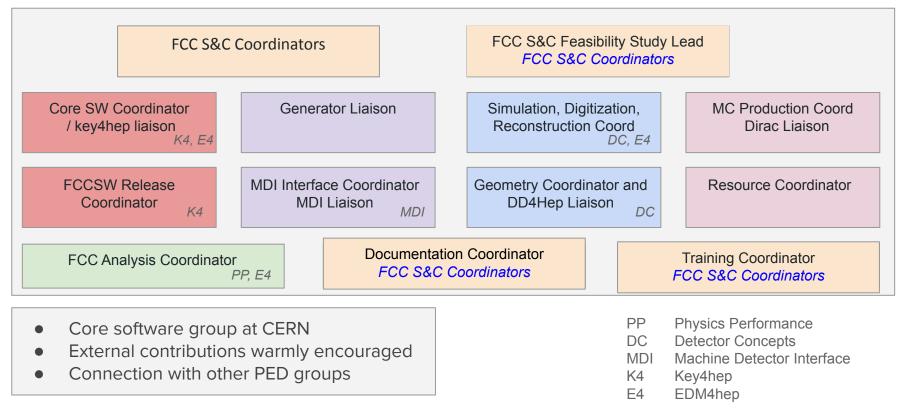
 Make sure that tools for mixing and overlapping events are available and functional.
- 8. [C] Make software builds available on more platforms
 ARM and MacOsX builds would be particular appreciated by end users

. . .

[R] Reconstruction, [G] Generators interfaces, [S] Simulation, [D] Digitisation, [M] MDI, [C] Core

FCC S&C revised structure





Coordination opportunities



1. Monte Carlo Production coordinator and DIRAC liaison

Coordinate all facets of Monte Carlo production including sample prioritization. Works with generator experts, resource managers, and others to ensure that needed samples are produced in a timely way [...]. Assists users with MC sample availability issues and their interactions with DIRAC

2. Resource coordinator

Responsible for identifying and coordinating compute and storage resources for `FCC`. Works with resource providers, including CERN, WLCG grid sites involved in `FCC` research, and `HPC` centers to enable `FCC` Monte Carlo production and analysis needs. [...]

3. Analysis Tools coordinator

Establish and maintain infrastructure and best practices for user analysis. Assist user community with on-boarding and encourage integration of user-developed features into the infrastructure.

4. Generators Liaison

Works with physics group and generator authors to integrate and update generator software needed for FCC studies. Fosters use of the agreed exchange formats for passing results between generators and from generators to detector simulation applications. [...]

Coordination opportunities (cnt'd)



5. Geometry coordinator and DD4hep liaison

Assists detectors experts with sub-detector geometry integration; works with detector concert experts to orchestrate sub-detectors in full detector geometries. Act as primary contact with DD4hep developers for reporting an prioritizing FCC issues and development priorities.

6. MDI Interface Coordinator and MDI Liaison

Oversee integration of programs for beam-background simulation. Work with experts to integrate background simulations using established data-exchange interfaces.

7. Simulation, Digitization and Reconstruction application coordinator

Orchestrate application components for simulation and digitization. Identify missing or incomplete algorithm components. Ensure that developments are integrated into applications in a timely fashion. Application configuration should continue to evolve using a flexible and scalable approach. Some aspects will be detector concept or geometry driven.

Summary



Horizon for the current FCC S&C efforts is the Feasibility Study Report (2025)

- Studies should be done by ~ end of 2024, software 'final' by ~ end of 2023
 - Tight timescale does not allow for revolutions
- Joining Key4hep helped in sharing experience and optimise workforce needs

Current efforts focused on

- Getting detectors concepts in full simulation
- Have a minimal complete set of reconstruction tools
- Getting adequate computing resources

Core workforce (CERN + external institutes) should guarantee minimal service

 Room for contributions at both technical, especially for reconstruction, and coordination level

Useful pointers



Project repositories

- GitHub: https://github.com/key4hep
- CernVM-FS: /cvmfs/sw.hsf.org, /cvmfs/fcc.cern.ch

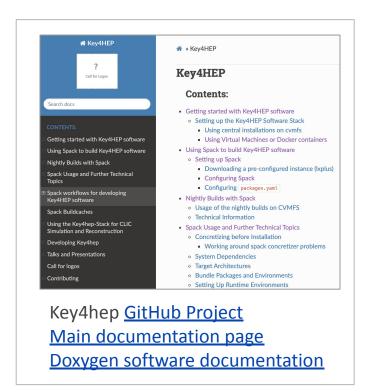
Forum: https://fccsw-forum.web.cern.ch/

Existing documentation: https://hep-fcc.github.io/fcc-tutorials/index.html

EPJ+ Software & Computing contributions (Part IV)

- Accelerator-related codes and their interplay with the experiment's software
- Online computing challenges: detector and readout requirements
- Offline Computing resources for FCC-ee and related challenges
- Key4hep, a framework for future HEP experiments and its use in FCC

Documentation, tutorials, ...





Demo/tutorials this afternoon in this room

