Status of the software

5th FCC Physics Workshop

Feb 10, 2022
G Ganis
CERN-EP

S&C goals for the workshop

- Advertise status of things and available workflows
- Address technical aspects of detector concept implementation

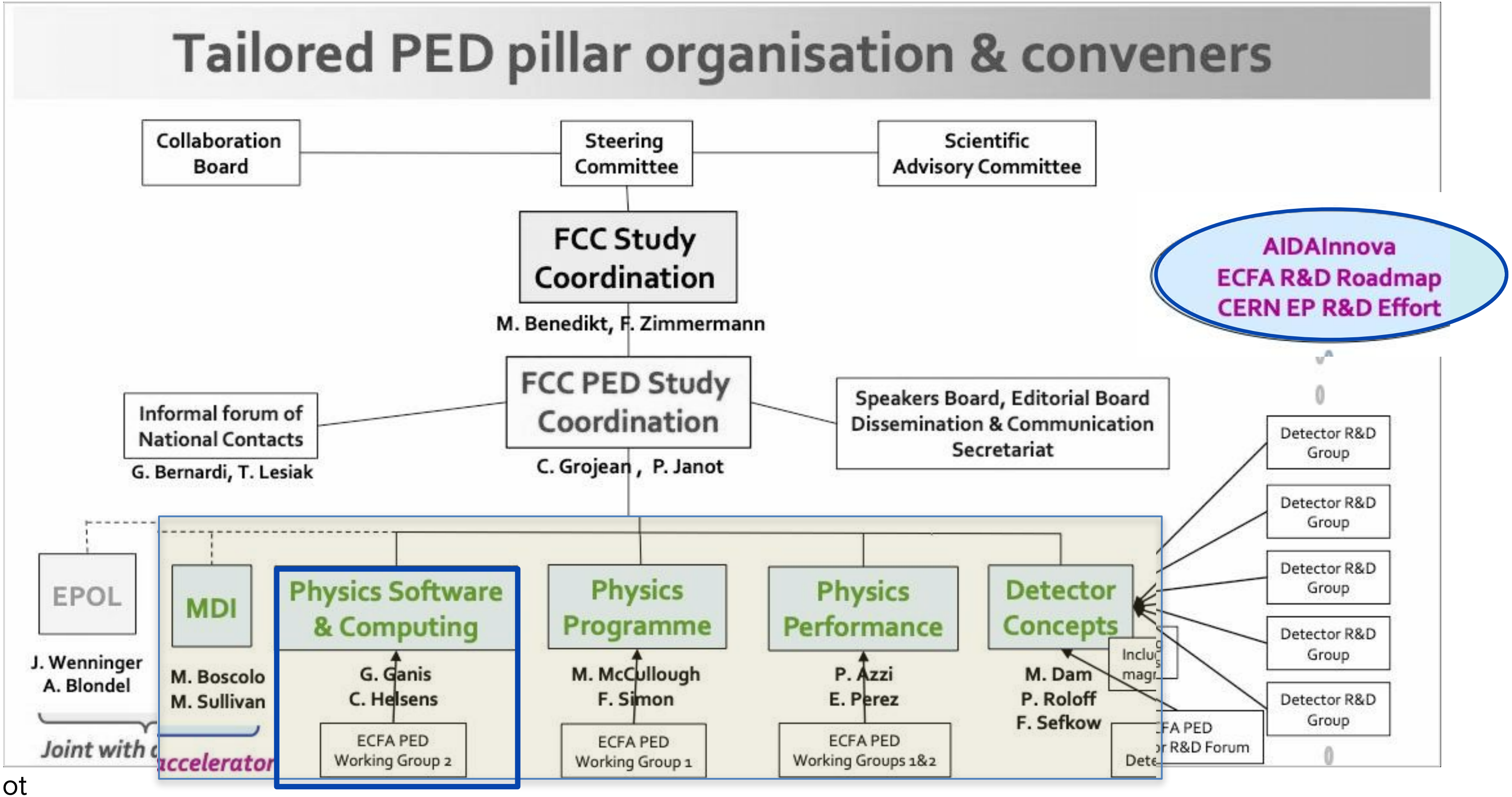
Today

- Strengthen connections with all the other branches in PED
    - In particular with the new-born Detector Concepts group
- Engage the community towards full simulation and reconstruction

Tomorrow

- Present current status of the updated S&C group mandate

Role and relation with other groups



Tailored PED pillar organisation & conveners

Collaboration Board

Steering Committee

Scientific Advisory Committee

FCC Study Coordination

M. Benedikt, F. Zimmermann

AIDAInnova
ECFA R&D Roadmap
CERN EP R&D Effort

Informal forum of National Contacts

G. Bernardi, T. Lesiak

FCC PED Study Coordination

C. Grojean , P. Janot

Speakers Board, Editorial Board
Dissemination & Communication
Secretariat

Detector R&D Group

Detector R&D Group

Detector R&D Group

Detector R&D Group

Detector R&D Group

EPOL

J. Wenninger
A. Blondel

MDI

M. Boscolo
M. Sullivan

Joint with acceleratorPhysics Software & Computing

G. Ganis
C. Helsens

ECFA PED Working Group 2

Physics Programme

M. McCullough
F. Simon

ECFA PED Working Group 1

Physics Performance

P. Azzi
E. Perez

ECFA PED Working Groups 1&2

Detector Concepts

M. Dam
P. Roloff
F. Sefkow

Inclu
magr

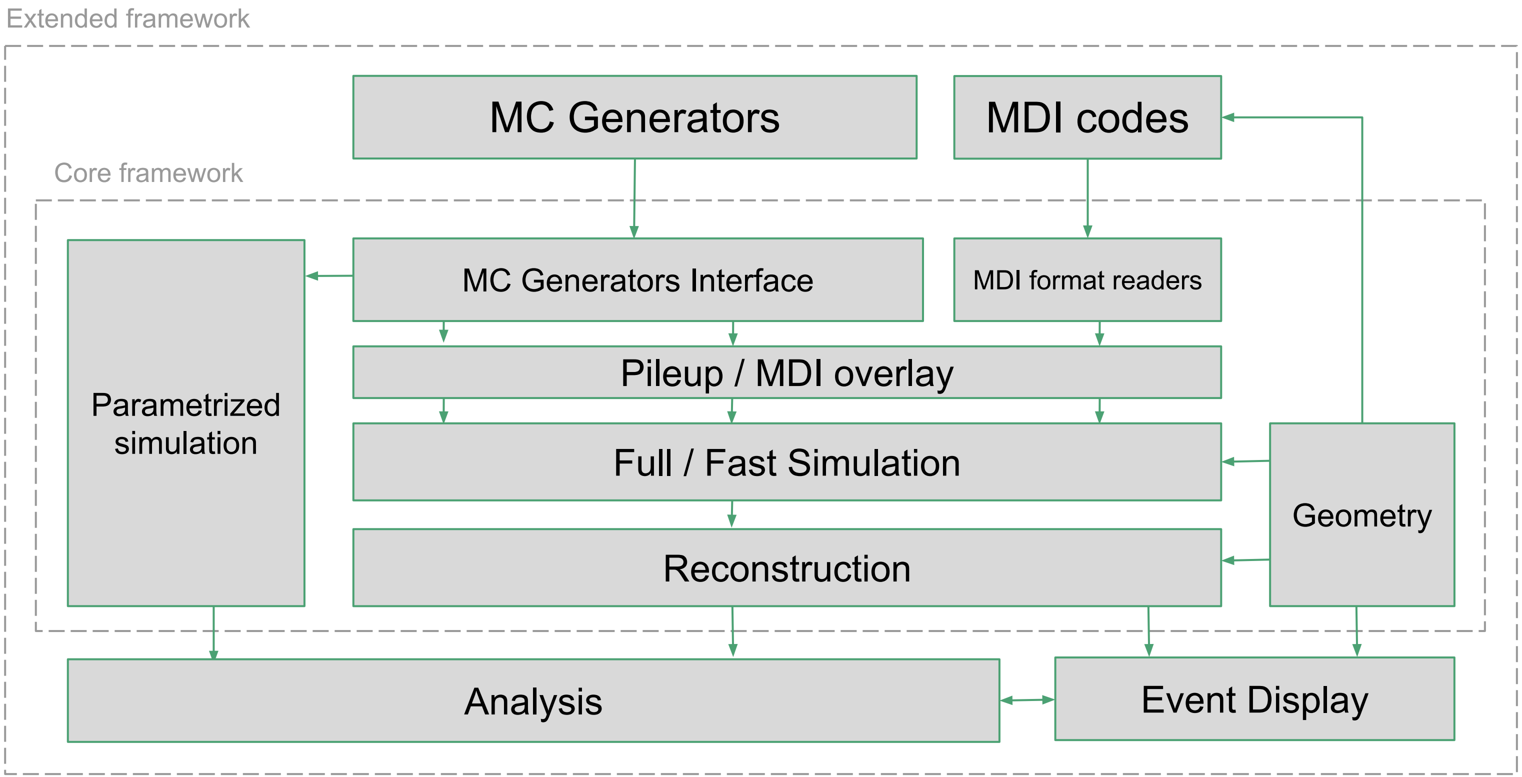
FA PED
r R&D Forum

Dete

Relation with R&D activities

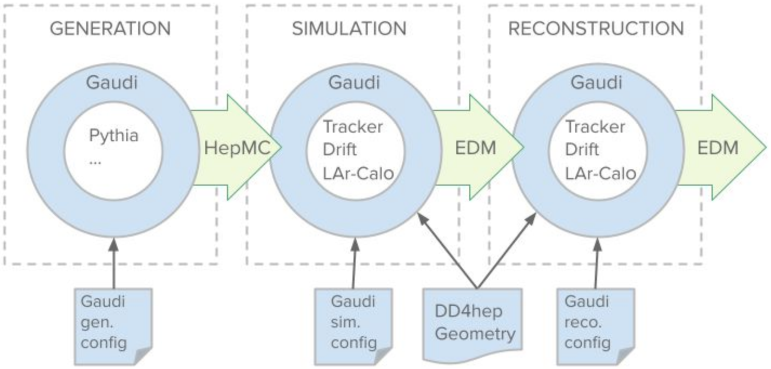
- CERN EP software R&D
  - All carried-on task/activities are connected to FCC needs
  - Key4hep is crucial, but all the other carried-on activities are connected
    - E.g. (ML-based) fast simulation of calorimeters, RNtuple-related analysis improvements
- AIDA Innova
  - Very similar palette of software R&D activities
  - Could potentially also profit of some person power for specific tasks
- ECFA R&D
  - Connection with WG2 (Physics Analysis Methods)
    - Generators, Simulation, Reconstruction, Algorithms & Tools, Software Ecosystem

Typical workflows to support



FCCSW approach

- Started in 2014, focus on FCCee after CDR (2019)



- Adopted Strategy
  - Adapt existing solutions from LHC (Gaudi, …)
  - Look at ongoing common R&D projects (AIDA)
  - Invest in streamlining of event data model

- 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

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 for

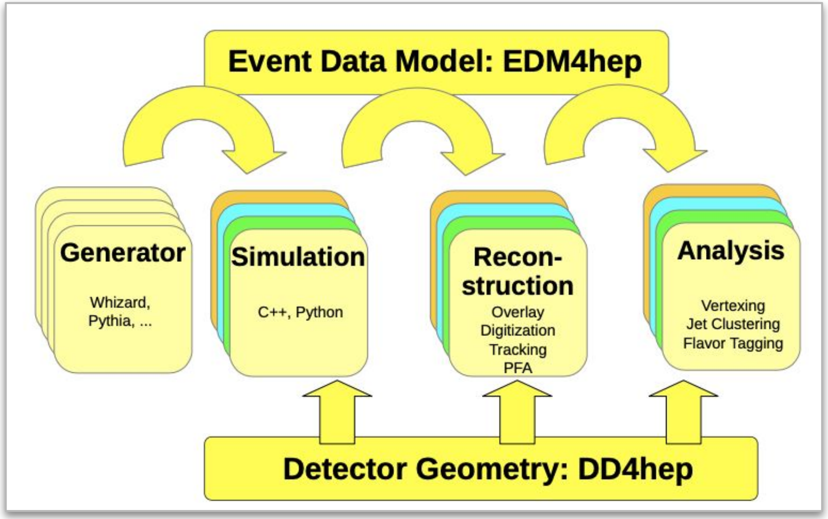
- Generation, simulation, reconstruction, analysis
- Build, package, test, deploy, run

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, lot of interest from LHC

Community project, unifying efforts

- Contributions from CLIC, ILC, FCC, CEPC



Kick-off meetings in Bologna, Hong Kong

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

FCCSW: status of things

Main goal for 2021: move to key4hep ✓

- Monte Carlo generators .... ✓≈
  - General purpose + old LEP available
  - Treatment of BES, x-angle, … still in progress
- MDI interfaces .... ≈
  - GuineaPig prototype available, start investigation of SR
- Fast simulation (Delphes) .... ✓
- Full simulation and reconstruction .... ≈
  - Single components available (machine elements, simplified vertex+DC, LAr calo, …)
  - CLD (reconstruction through iLCSoft wrappers)
  - IDEA standalone, integration scheduled
- Distributed computing .... ✓≈
  - iLCDirac instance set up (CERN, CNAF storage)
- Analysis .... ✓≈
  - FCCAnalysis (EMD4hep, ROOT::RDF)
- Documentation .... ✓≈

Monte Carlo Generators @ key4hep

- All relevant generators available
  - Including legacy LEP ones: KKMC, BHLUMI, BabaYaga, ...
- Repository of ready-to-use codes
  - (Software-wise) sanitized: build, tested (with provide test suite)
- Interfaces to key4hep processing chain (k4Gen)
  - Support for relevant data formats
  - Auxiliary tools related to MC particles (filters, boost tools, …)

Connections w/
- Physics Programme
- Physics Performance
- Phenomenologists
- Key4hep
- ECFA

Generators: relation w/ ECFA

- One of the work areas of WG2
- 1st workshop on Generators: 9-10 Nov 2021
  - Taking stock
  - Set up contacts (among authors)
  - Discussion of formats and k4Gen role (see next)
  - Identified topics for follow-up focus workshops
    - Beamstrahlung, Interfaces, Benchmarks
- 1st focus workshop on beamstrahlung: 12 Jan 2022
  - Dissection of limitations of GuineaPig and related approaches (CAIN)
  - Main uncertainties arise from missing knowledge of input beam distributions
    - Better estimates by using input files from existing tracking codes
      - Need to develop the measurement procedure to be applied on the data
  - Possible double counting of ISR to be investigated

Discussion of formats and role of k4Gen

- Have a small number of well defined I/O output(s)
  - Efficient(s) and compressed
  - Baseline: EDM4hep@{ROOT, SLCIO}, HepMCv3@ROOT (common among MCs)
- Recognized role of HepMCv3
  - Possibly provide / contribute converter to EDM4hep
  - LHEf not adapted for this
- Boundary generators / key4hep
  - Baseline: generators provide fully hadronized events
    - Key4hep starts from the generator output
    - In particular, dedicated decayers, hadronizers, should be run before, i.e, by the generator
    - HepMCv3 / EDM4hep formats preferably
  - Modules providing a direct MC to EDM4hep interface can still be accommodated
    - E.g. PythiaInterface, HerwigInterface, …
    - Possibly under the responsibility of the authors (to be discussed)

Beam and MDI-related backgrounds

- FCCee complex interaction region combined with high statistical precision requires deep level of understanding of the detector backgrounds (*)
  - Only achievable with integration in experiment software
- Several codes for modeling the processes (EPJ+ contribution)
  - Codes not always in public repositories, outputs in different, non-standard formats
- Target: supercode providing common interface to relevant codes
  - Unified/simplified access with controlled configuration and normalization
  - On going effort, example w/ GuineaPig available
- 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

(*) Active working group with experts from relevant fields (machine, physics performance, detectors, software)

Parametrized Simulation

- Current studies are mostly based on Delphes
  - Includes a tracking system, embedded into a magnetic field, calorimeters and a muon system
    - TrackCovariance, dEdx, ParticleDensity: enable realistic algorithms for vertexing, b-tagging, …
    - Effect of magnetic field, granularity of calorimeters, sub-detector resolutions
  - Interfaced to standard file formats (e.g. LHEF, HepMC)
  - Key4hep provides Delphes interfaces/executables producing EDM4hep output
  - Palette of detector concepts for e^+e^- available
- SVG (used by the LC community) being also considered in key4hep
  - Potentially more complete and faster
  - Needs some adaptation work (EDM4hep output, …)

Connections w/
- Physics Performance
- Detectors
- ECFA

Nice reviews and usage examples at
1st ECFA simulation workshop 1-2 Feb 2022

Geometry description: DD4hep

- Description of geometry and material for all relevant elements
  - Sub-detectors and related elements (supports, …)
  - Relevant MDI elements
- Single source for the several needs
  - Simulation, Reconstruction, Event Display, …
  - MDI studies
    - Including CAD interoperability
- DD4hep provides all that. And more
  - DDG4, DDRec, DDCond, …
  - AIDA project fostered by LC community and now adopted by CMS, LHCb as of Run3

Simulation

Connections w/
- Key4hep
- ECFA (1st Simulation workshop)

- Full simulation uses Geant4
- k4SimGeant4
  - Gaudi module supporting
    - User actions, regions, sensitive detectors, selective output options
  - Mixing fast and full G4 simulation possible (SimG4Full / SimG4Fast)
  - Based on LHCb approach: Gaudi is in control of the event loop
- DDSim
  - DD4hep standalone interface to Geant4
  - Used in iLCSoft
    - Lot of tools available from LC (digitization, …)
  - Currently only way to fully simulate/reconstruct CLD (also for FCCee)
- Future: k4Gaussino
  - Based on new LHCb approach, might allow easier re-use of code

Detector palette

- FCCDetectors contains what is currently available
  - Elements of the interaction region (BeamPipe, Instrumentation, HOMAbs, LumiCal)
  - CLD (CLIC detector for FCC)
  - LAr calorimenter
  - Simplified Elements for IDEA (no calo, no muon)
  - Simplified Elements for IDEA with FCChh adapted calo (LAr/Tile) (no muon)
  - FCChh baseline (+ some variants)

- Standalone implementations
  - IDEA Dual-Readout calorimeter: dual-readout
  - IDEA Drift Chamber and vertex: DriftChamberPLUSVertex

Connections w/
- Detectors
- Physics Performance

Joint session on Friday

Supporting detector concepts studies

- Adequate support for sub-detector concepts developers
  - Documentation, templates, examples, …
  - Need to provide/define envelopes where to fit in the sub-detectors
- Plug&Play support
  - Support for detector concepts creation as combination of sub-detector solutions
    - Tracking, calorimetry, muon detection, …
  - Includes overall sub-detector integration test
- Git repository with sub-detector concepts
  - With proper and efficient versioning
    - Sub-detector specific and global (for the assembly)

Supporting detector concepts studies (2)

- Visualization
  - Is what provided by DD4HEP enough (geoDisplay)?
  - Perhaps integrated with the event display
    - Phenix? See latest developments driven by LHCb
- Debugging tools
  - Checking for overlaps, radiation lengths, …

Reconstruction

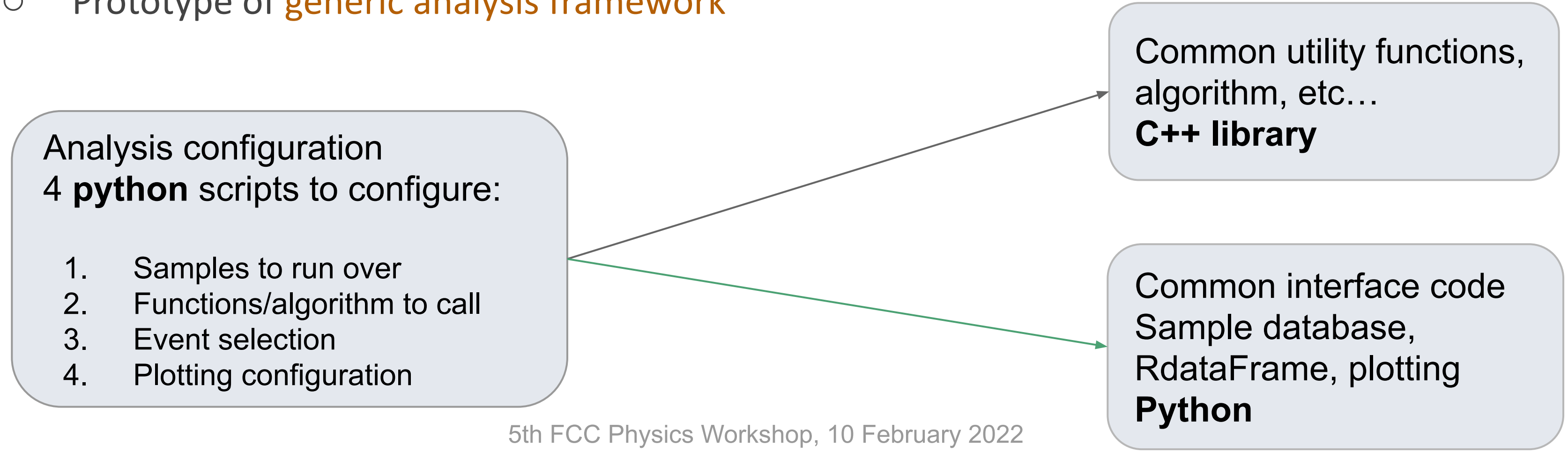
- Little specific to FCC-ee
  - Tracking and calorimetric algorithms for baseline FCC-hh
  - Full sim studies for FCCee not really started
- Lots of algorithms available for iLCSoft
  - Accessible through LCIO to/from EDM4hep on the fly converter
    - Enables initial studies and evaluations
    - Base / 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
- Ongoing framework integration of general purpose tools such as ACTS, PandoraPFA, CLUE/TICL, ...

Connections w/
- Physics Performance
- Detectors
- Key4hep
- ECFA (1st Reco Workshop May 4-5 2022)

Analysis: FCCAnalysis

- Replaces the fully pythonic HEPPY framework used for CDR
  - Good functionality but extremely slow
- Based on RDataFrame, new ROOT paradigm aimed 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

Connections w/
- Physics Performance
- Key4hep



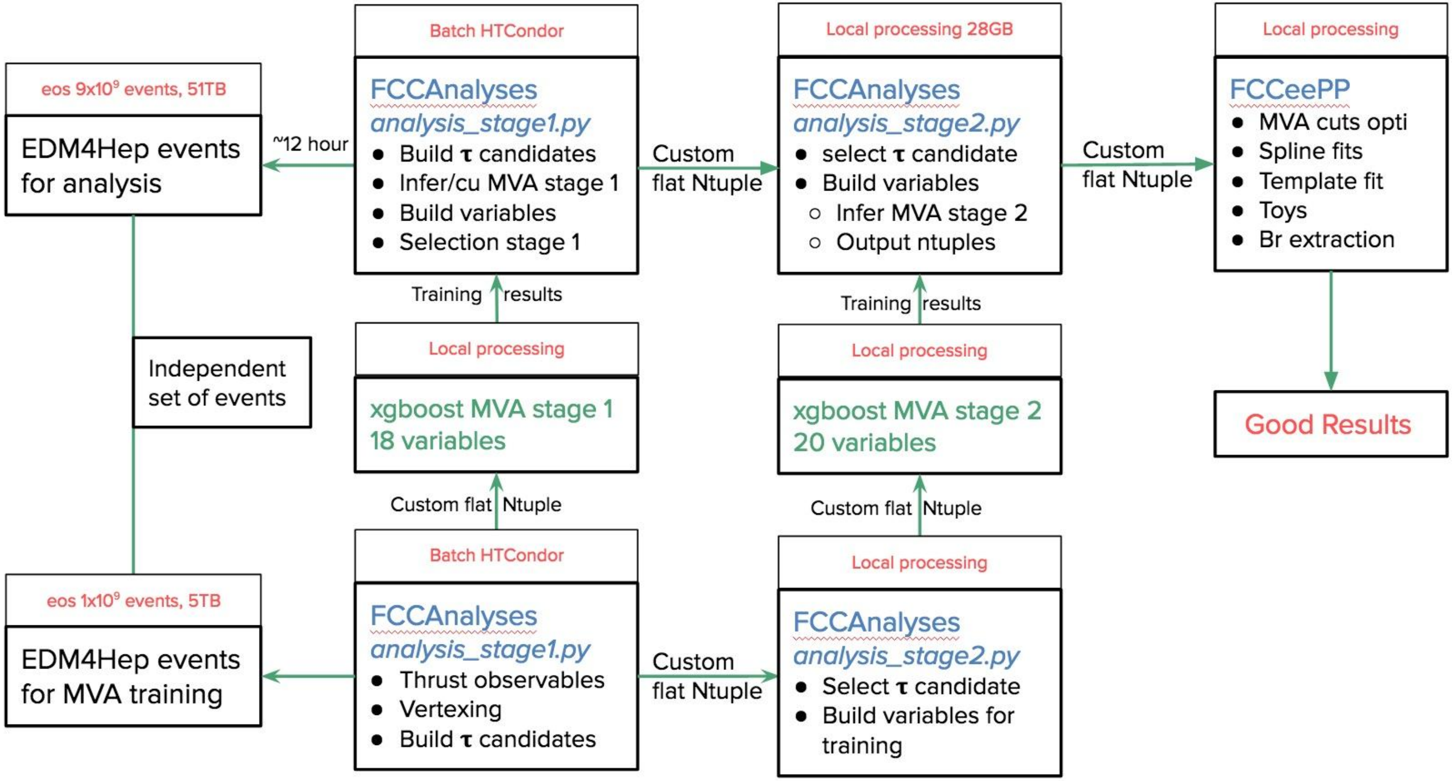
Analysis configuration
4 **python** scripts to configure:

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

Common utility functions,
algorithm, etc…
C++ library

Common interface code
Sample database,
RdataFrame, plotting
Python

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



eos 9x10^9 events, 51TB

EDM4Hep events for analysis

~12 hour

Batch HTCondor

FCCAnalyses *analysis_stage1.py*

- Build τ candidates
- Infer/cu MVA stage 1
- Build variables
- Selection stage 1

Custom flat Ntuple

Local processing 28GB

FCCAnalyses *analysis_stage2.py*

- select τ candidate
- Build variables
  - Infer MVA stage 2
  - Output ntuples

Custom flat Ntuple

Local processing

FCCeePP

- MVA cuts opti
- Spline fits
- Template fit
- Toys
- Br extraction

Good Results

Training results

Local processing

xgboost MVA stage 1
18 variables

Training results

Local processing

xgboost MVA stage 2
20 variables

Independent set of events

Custom flat Ntuple

Custom flat Ntuple

eos 1x10^9 events, 5TB

EDM4Hep events for MVA training

Batch HTCondor

FCCAnalyses *analysis_stage1.py*

- Thrust observables
- Vertexing
- Build τ candidates

Custom flat Ntuple

Local processing

FCCAnalyses *analysis_stage2.py*

- Select τ candidate
- Build variables for training

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

Resource needs for the FSR

See EPJ+ contribution

- The run at Z peak sets the scale
  - $\approx 10^{12}$ 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 $B_c \longrightarrow \boldsymbol{\tau^+ \nu_\tau}$)
  - 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
- 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

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

Thanks to
- A Sailer (CLIC, key4hep)

FCC @ iLCDirac : adding resources

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

```
lxplus:~$ source /cvmfs/clicdp.cern.ch/DIRAC/bashrc
lxplus:~$ dirac-proxy-init -g fcc_user
lxplus:~$ dirac-dms-lfn-replicas /fcc/user/g/ganis/edm4hep_test_output.root
LFN                                        StorageElement URL
=============================================================
/fcc/user/g/ganis/edm4hep_test_output.root    CNAF-DISK      davs://xfer-archive.cr.cnaf.infn.it:8443/fcc/user/g/ganis/edm4hep_test_output.root
                                              CERN-DST-EOS   gsiftp://eospublicftp.cern.ch//eos/experiment/fcc/prod/fcc/user/g/ganis/edm4hep_test_output.root
lxplus:~$ source /cvmfs/sw.hsf.org/key4hep/setup.sh
lxplus:~$ root -l
root [0] TFile *f1 = TFile::Open(“davs://xfer-archive.cr.cnaf.infn.it:8443/fcc/user/g/ganis/edm4hep_test_output.root”)
(TFile *) 0x2ec28b0
root [1]
```

Summary

- Plans for 2021 mostly achieved
    - Adopted key4hep and consolidated Delphes-based workflow
- Next is to fully engage in full simulation and reconstruction
    - Strengthen connections with Physics Performance and Detector Concepts groups
- Fruitful connection established with ECFA and R&D projects
- Planning of resources required for FSR
    - Interplay full / parametrized simulation
    - Framework to add national contributions enabled

Software solidity requires the contribution of skilled users

- The community should feel engaged in supporting/fostering the effort

Useful pointers

Project repositories

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

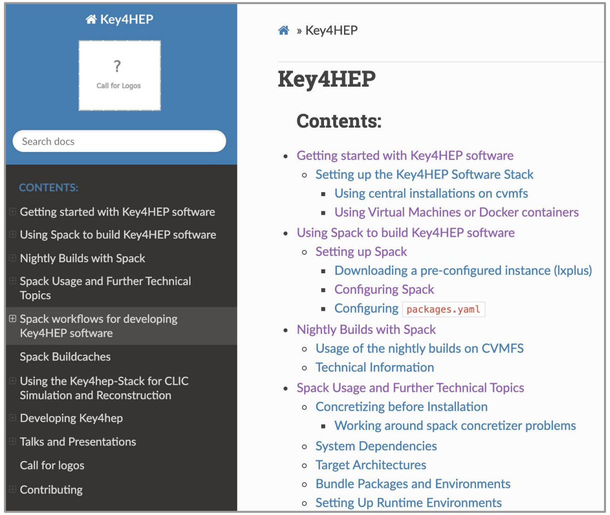
Forum: https://fccsw-forum.web.cern.ch/

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, …



Key4HEP

?

Call for Logos

Search docs

CONTENTS:

- Getting started with Key4HEP software
- Using Spack to build Key4HEP software
- Nightly Builds with Spack
- Spack Usage and Further Technical Topics
- Spack workflows for developing Key4HEP software
- Spack Buildcaches
- Using the Key4hep-Stack for CLIC Simulation and Reconstruction
- Developing Key4hep
- Talks and Presentations
- Call for logos
- Contributing

» Key4HEP

Key4HEP

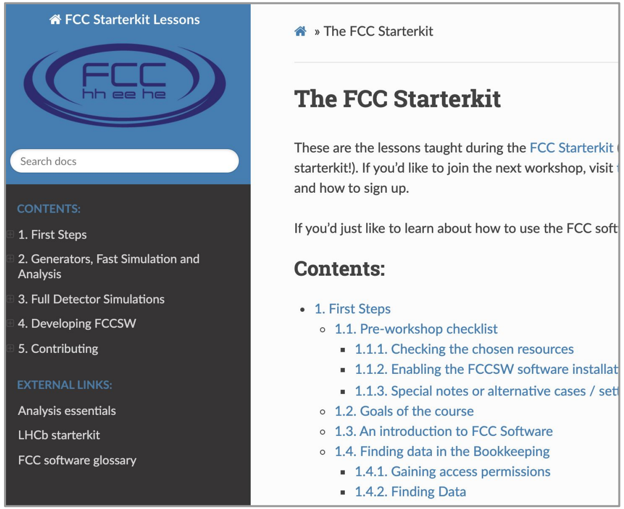
Contents:

- Getting started with Key4HEP software
  - Setting up the Key4HEP Software Stack
    - Using central installations on cvmfs
    - Using Virtual Machines or Docker containers
- Using Spack to build Key4HEP software
  - Setting up Spack
    - Downloading a pre-configured instance (lxplus)
    - Configuring Spack
    - Configuring `packages.yaml`
- Nightly Builds with Spack
  - Usage of the nightly builds on CVMFS
  - Technical Information
- Spack Usage and Further Technical Topics
  - Concretizing before Installation
    - Working around spack concretizer problems
  - System Dependencies
  - Target Architectures
  - Bundle Packages and Environments
  - Setting Up Runtime Environments

Key4hep GitHub Project

Main documentation page

Doxygen software documentation



FCC Starterkit Lessons

Search docs

CONTENTS:

- 1. First Steps
- 2. Generators, Fast Simulation and Analysis
- 3. Full Detector Simulations
- 4. Developing FCCSW
- 5. Contributing

EXTERNAL LINKS:

- Analysis essentials
- LHCb starterkit
- FCC software glossary

» The FCC Starterkit

The FCC Starterkit

These are the lessons taught during the FCC Starterkit
starterkit!). If you'd like to join the next workshop, visit
and how to sign up.

If you'd just like to learn about how to use the FCC soft

Contents:

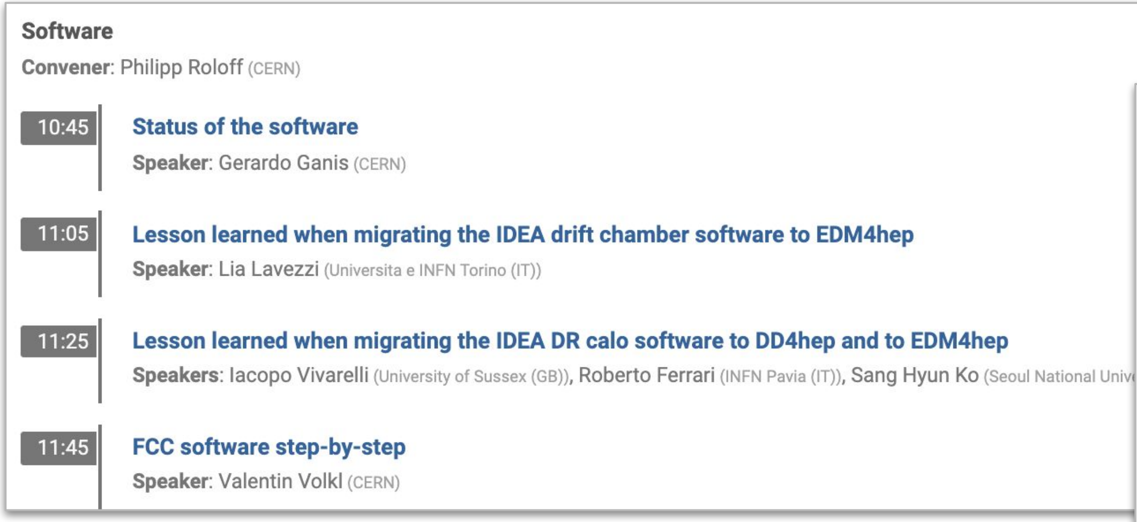
- 1. First Steps
  - 1.1. Pre-workshop checklist
    - 1.1.1. Checking the chosen resources
    - 1.1.2. Enabling the FCCSW software installat
    - 1.1.3. Special notes or alternative cases / set
  - 1.2. Goals of the course
  - 1.3. An introduction to FCC Software
  - 1.4. Finding data in the Bookkeeping
    - 1.4.1. Gaining access permissions
    - 1.4.2. Finding Data

FCCSW GitHub Project

Main documentation page

Workshop software sessions

Today



Software

Convener: Philipp Roloff (CERN)

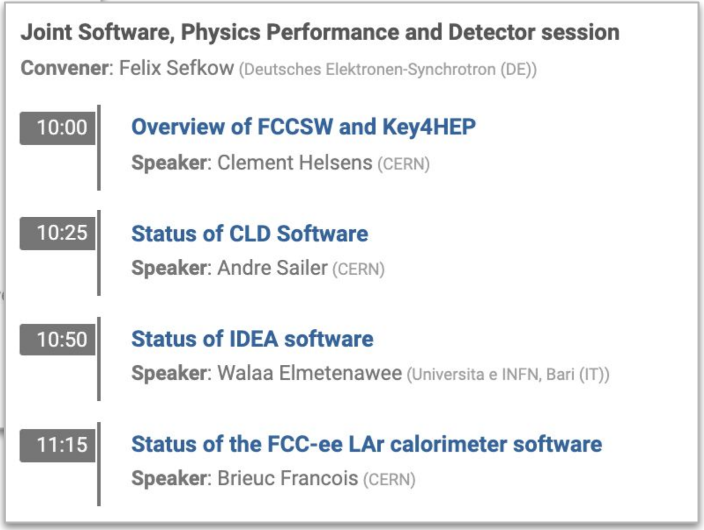
10:45 **Status of the software**
Speaker: Gerardo Ganis (CERN)

11:05 **Lesson learned when migrating the IDEA drift chamber software to EDM4hep**
Speaker: Lia Lavezzi (Universita e INFN Torino (IT))

11:25 **Lesson learned when migrating the IDEA DR calo software to DD4hep and to EDM4hep**
Speakers: Iacopo Vivarelli (University of Sussex (GB)), Roberto Ferrari (INFN Pavia (IT)), Sang Hyun Ko (Seoul National Univ

11:45 **FCC software step-by-step**
Speaker: Valentin Volkl (CERN)

Tomorrow



Joint Software, Physics Performance and Detector session

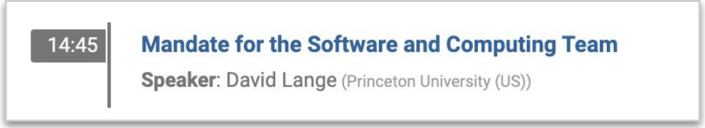
Convener: Felix Sefkow (Deutsches Elektronen-Synchrotron (DE))

10:00 **Overview of FCCSW and Key4HEP**
Speaker: Clement Helsens (CERN)

10:25 **Status of CLD Software**
Speaker: Andre Sailer (CERN)

10:50 **Status of IDEA software**
Speaker: Walaa Elmetenawee (Universita e INFN, Bari (IT))

11:15 **Status of the FCC-ee LAr calorimeter software**
Speaker: Brieuc Francois (CERN)



14:45 **Mandate for the Software and Computing Team**
Speaker: David Lange (Princeton University (US))