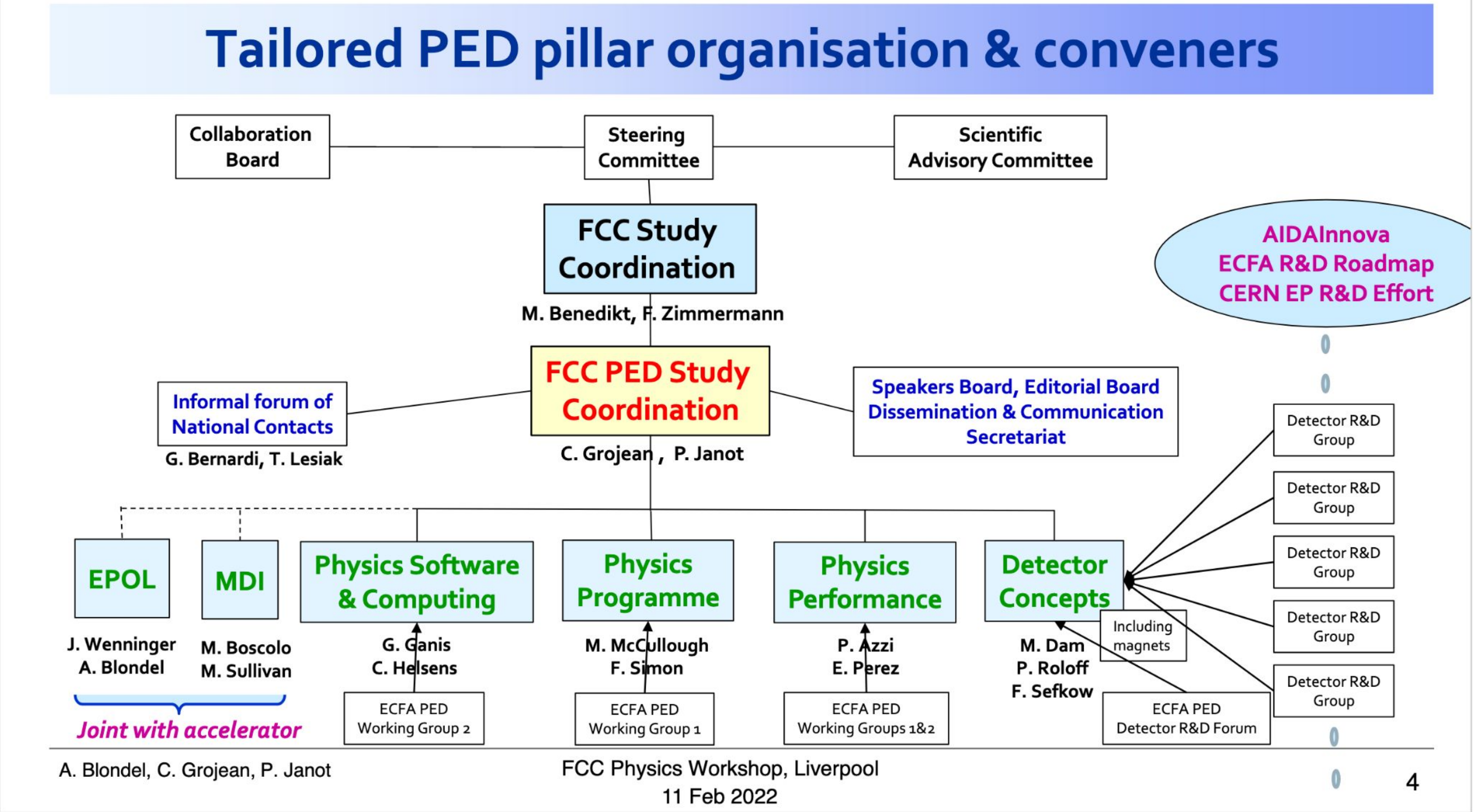
Overview of FCCSW and Key4HEP

5th FCC Physics workshop, Liverpool → Live-rpool → Vir-lepool

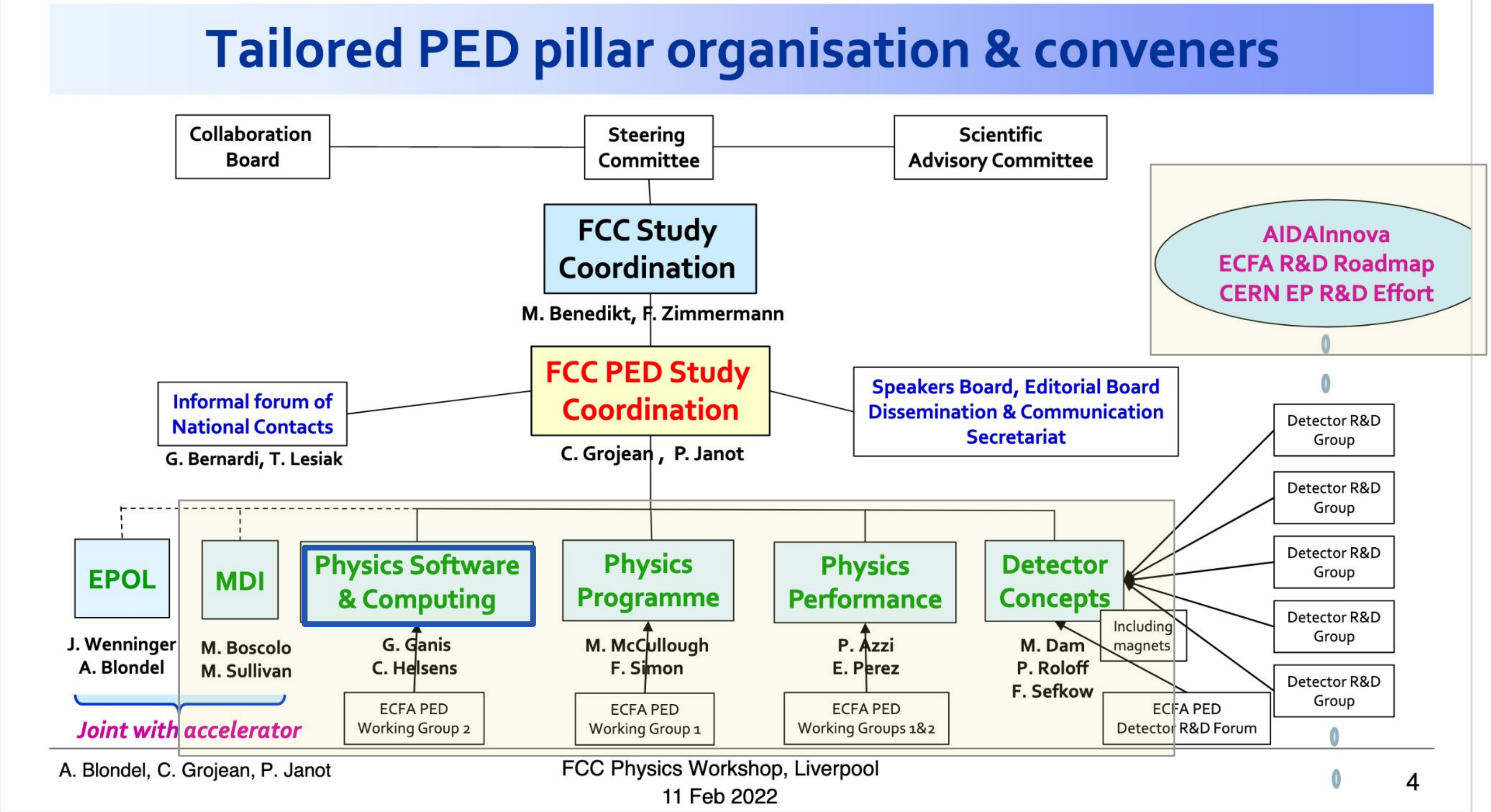
Feb 11, 2022

C Helsens
CERN-EP → KIT

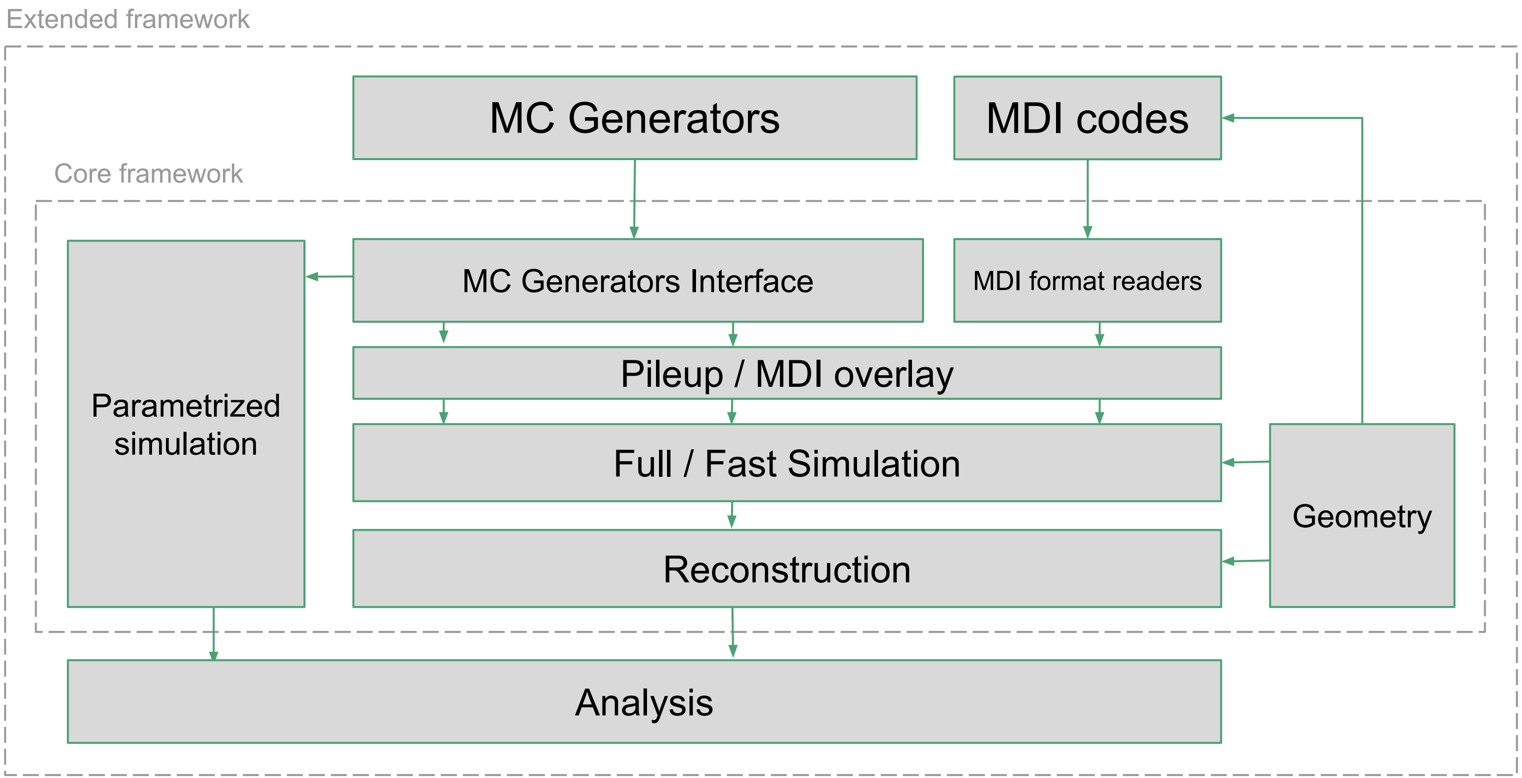
Role and relation with other groups



Role and relation with other groups



Typical workflows to support



Software Challenges

Some challenges for the next phase of the studies could be identified

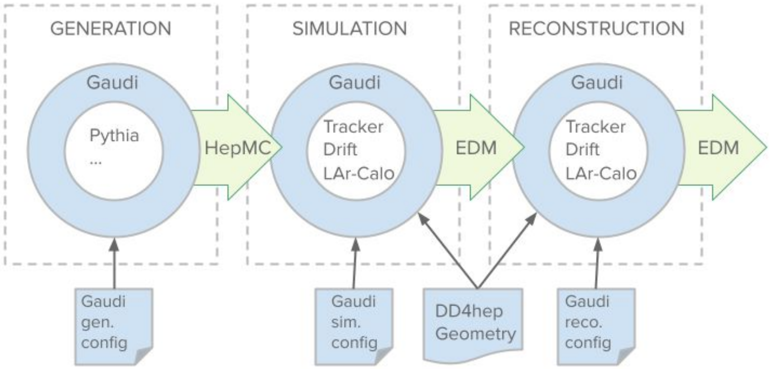
- Suport Complex workflows
  - MC Simulations: Event Generation, Particle Propagation
  - Backgrounds (from the machine), Digitization, Reconstruction, Analyses ...

- Performance
  - Distributed computing infrastructure
  - Efficient parallel programming to use evolving hardware efficiently
  - Anticipate and quickly integrate new technologies

- Advantage?
  - Not all features needed right away
  - ... gives time for proper design!

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 (Gaudi, …)
  - Look at ongoing common R&D projects (AIDA)
  - Invest in streamlining of event data model
- Focus on FCCee after CDR (2019)



The common Key4Hep 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

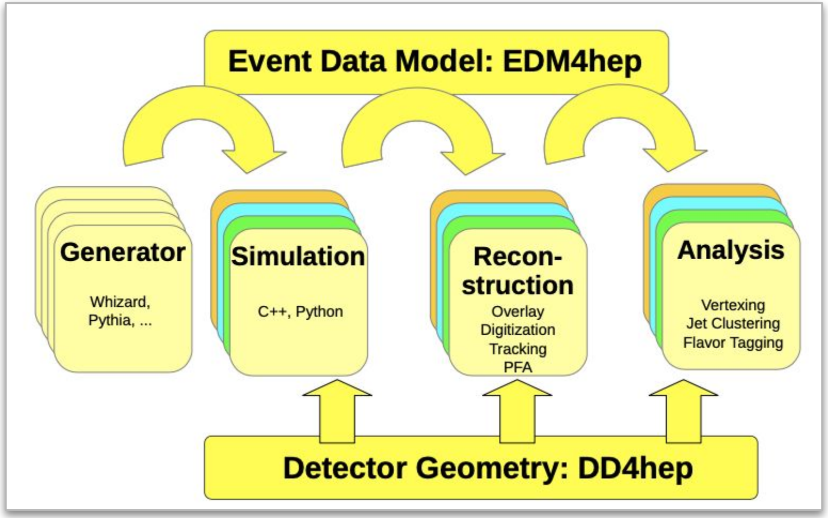
- Generation, simulation, reconstruction, analysis
- Build, package, test, deploy, run

Core Ingredients 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 Key4Hep software vision

Key4Hep Goals - take home messages

1. Key4Hep is a software stack that connects and extends packages to provide a complete data processing framework, comprising fast and full simulation, reconstruction, and analysis.

2. It should be ease of use for librarians, developers and users

3. It should provide examples, detailed documentation, templates and common best practices

Status of Experiment Software

From V. Volk @ Lepton Photons 2022

key4hep-stack/2021-10-29 comprises

...in a consistent stack!

CEPCSW

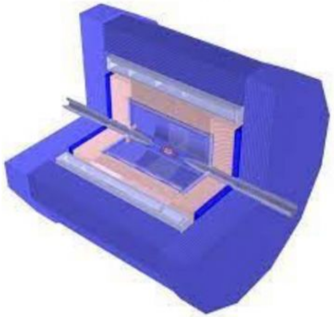
v0.2.2

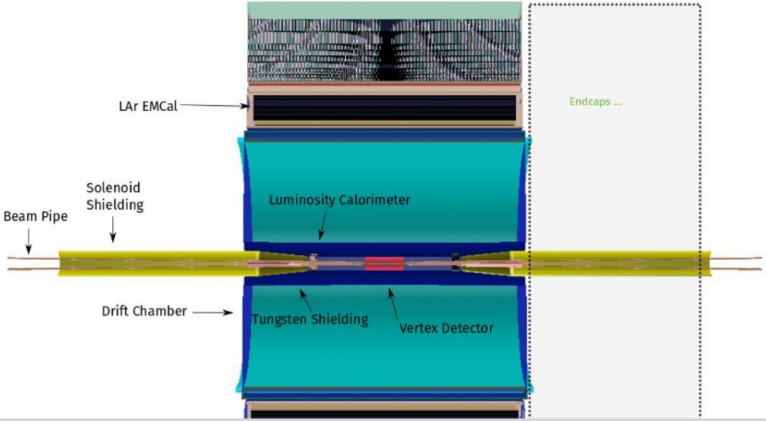
FCCSW

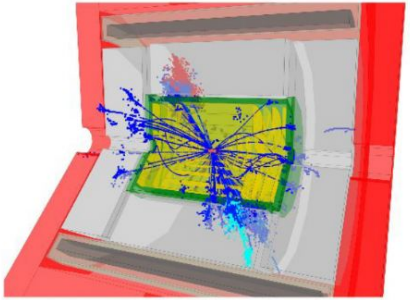
v1.0pre06

CLIC/ILCSoft

v02-02-03

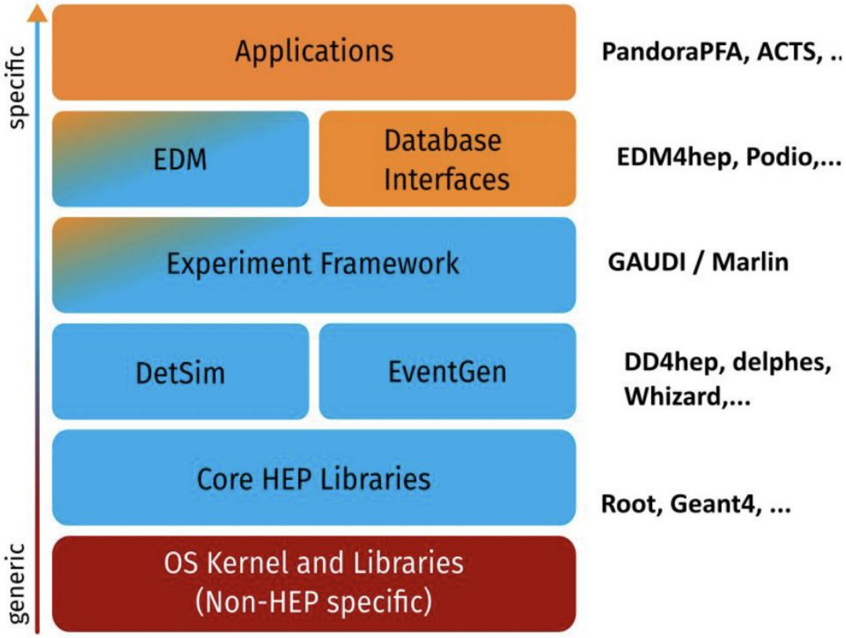






4

HEP software stack

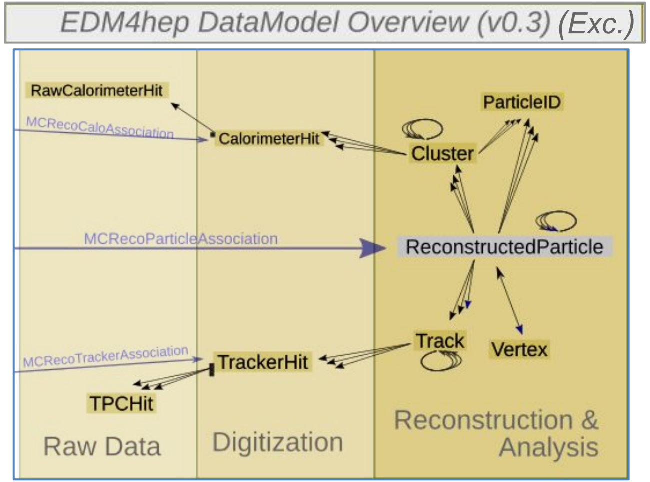


- Software tools are not living alone, they are social entities and they need to interact with their SW friends
- Compatibility between different elements doesn't come for free
  - Common standards can help a lot
- Building a consistent stack of software for an experiment is highly non-trivial
  - Benefits can be gained from using common approaches

The common event data model: the challenges

EDM provides common language for exchange among framework components

- Challenge 1: efficient support different collision environments (e^+e^-, pp, ...)
  - Positive first experiences with FCC-hh components
- Challenge 2: 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
- Challenge 3: efficient support for schema evolution
  - Requires schema evolution in PoDIO, planned
- Challenge 4: efficient support for detector needs
  - Interaction w/ detector teams from the start
    - Eg. cluster counting for IDEA Drift Chamber



FCCSW @ key4hep adoption process

- Already Gaudi based
  - Move FCC-edm to EDM4hep
  - Re-arrangement and modernization
  - Some components considered for migration to key4hep
    - Generation, simulation, reconstruction, …

- Migration status (as of end of last year in EPJ+ Essay Eur. Phys. J. Plus 137 no. 1, (2022) 149)

old FCCSW (version $\leq$ 0.16)	Key4hep	new FCCSW	status	migration
FWCore	k4FWCore		done	yes
Sim/SimDelphesInterface	k4SimDelphes		done	yes
Generation		k4Gen	done	under evaluation
Sim		k4SimGeant4	done	under evaluation
Reconstruction/Rec[...]Calorimeter		k4RecCalorimeter	done	under evaluation
Reconstruction/RecDriftChamber		to be determined		
Detector		FCCDetectors	done	no, FCC specific
	to be determined	dual–readout		under evaluation

Geometry description: DD4hep

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

- Standalone implementations
  - IDEA Dual-Readout calorimeter: dual-readout
  - IDEA Drift Chamber and muon chambers not started

Connections w/
- Detectors
- Physics Performance

Completing the detector palette in DD4hep

- DD4hep allows easily replacement of parts
  - Example is a reduced version of the FCC-hh 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:
    - 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

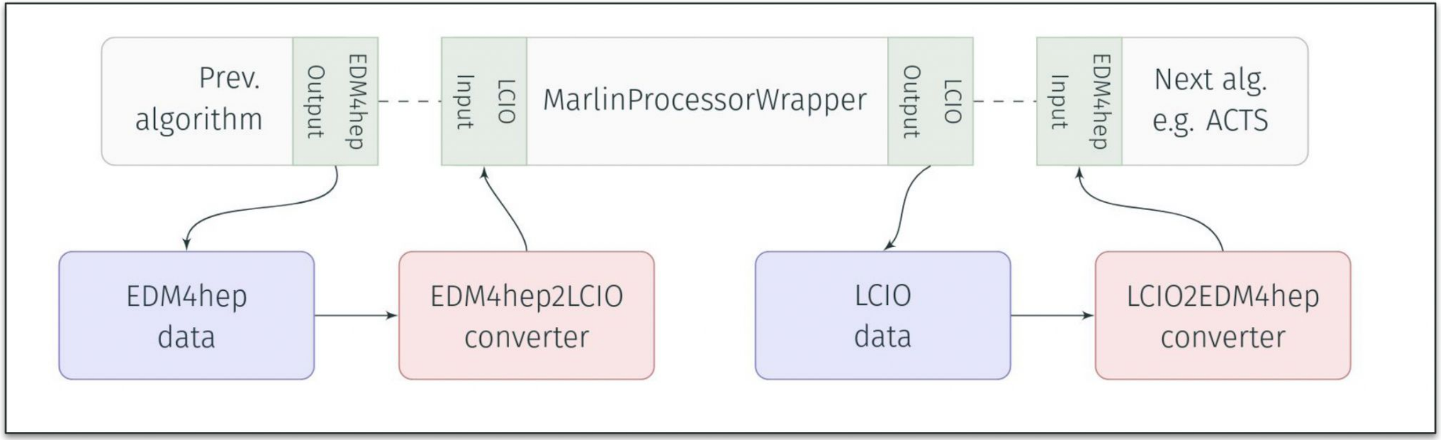
- Little specific to FCC-ee
  - Tracking and calorimetric algorithms for baseline FCC-hh
  - Full sim studies for FCC-ee are starting
- Lots of algorithms available for iLCSoft
  - Accessible through LCIO to/from EDM4hep on the fly converter
    - Enables initial studies and evaluations
    - Base / 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, ...

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

k4MarlinWrapper: Run ILCSoft through Gaudi

- Run Marlin processors through Gaudi Algorithms, without changes to Marlin
- Enhanced Marlin XML to Gaudi Python steering files converter
  - Optional execution markers
  - Constants parsing: propagated through the steering file
- Input and output support for LCIO and EDM4hep

From V. Volk @ Lepton Photons 2022



Towards supporting detector concepts studies (1)

- 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
  - Careful design when it comes to reconstruction of complex object, for example when there is a need to combine tracks and calorimeter clusters, generic vs specific
- Git repository with sub-detector concepts
  - With proper and efficient versioning
    - Sub-detector specific and global (for the assembly)

Towards 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, would be interesting collaboration for the long term
- Debugging tools
  - Checking for overlaps, radiation lengths, …
- Efficient computing and sufficient disk space
  - Should not be a limitation when testing ideas
  - Quick processing time from idea to result is a very important aspect

Some more (personal) thoughts

- For the next phase of the study, most of the contributors will most likely not be full time working on FCC
    - They will probably be sharing time with a running experiment where the data format, and procedures are well defined and tested
    - How can we make sure that their contributions, with limited time, can lead to outstanding results and avoid the frustration of slow progress?

- Connected to the previous point we need that the new generations of physicist become interested about detailed simulation (that we need to achieve outstanding performance) and not only about fancy machine learning (both could have significant overlap)

- We need to keep simulation expertise on the long run!!

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

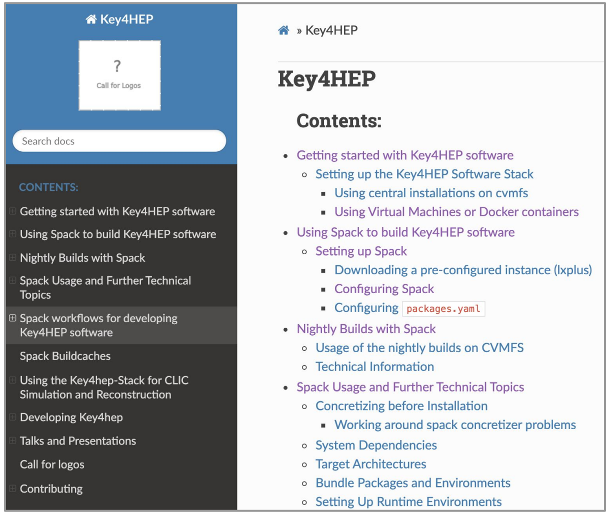
Software solidity requires the contribution of skilled users

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

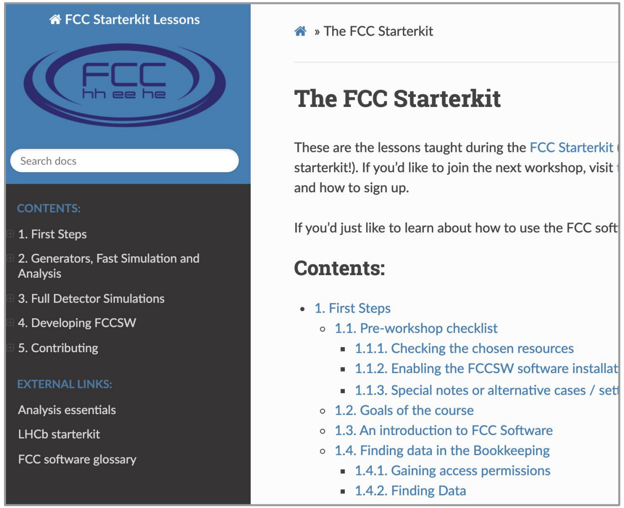
Useful pointers

- Project repositories
    - GitHub: https://github.com/HEP-FCC
    - CernVM-FS: /cvmfs/fcc.cern.ch
- Forum
    - https://fccsw-forum.web.cern.ch/
- Existing documentation
    - https://hep-fcc.github.io/fcc-tutorials/index.html

Documentation, tutorials, …



Key4hep GitHub Project
Main documentation page
Doxygen software documentation



FCCSW GitHub Project
Main documentation page