Status and plans of SW implementation of the FCC-ee calorimeters including PF reconstruction

3rd FCC Physics Week 2020

Jan 16, 2020

C Helsens, V Volkl
CERN-EP
And on behalf of the remaining people in the
FCC-hh Calorimeter group

Introduction

- Detector design studies
  - Flexibility
  - Ideal detector descriptions
  - Open to evolution
- Broad range of event complexity
  - e^+e^- *vs* pp *vs* ep
- Need to support physics and detector studies
  - Parameterised, fast and full simulation
- Aim to de-duplicate efforts
  - One software stack to support all the cases,
    all detector concepts and future (proto-)collaborations

FCCSW: packages and tools for FCC studies

Current software met basic requirements at a sufficient level to support CDRs Physics studies

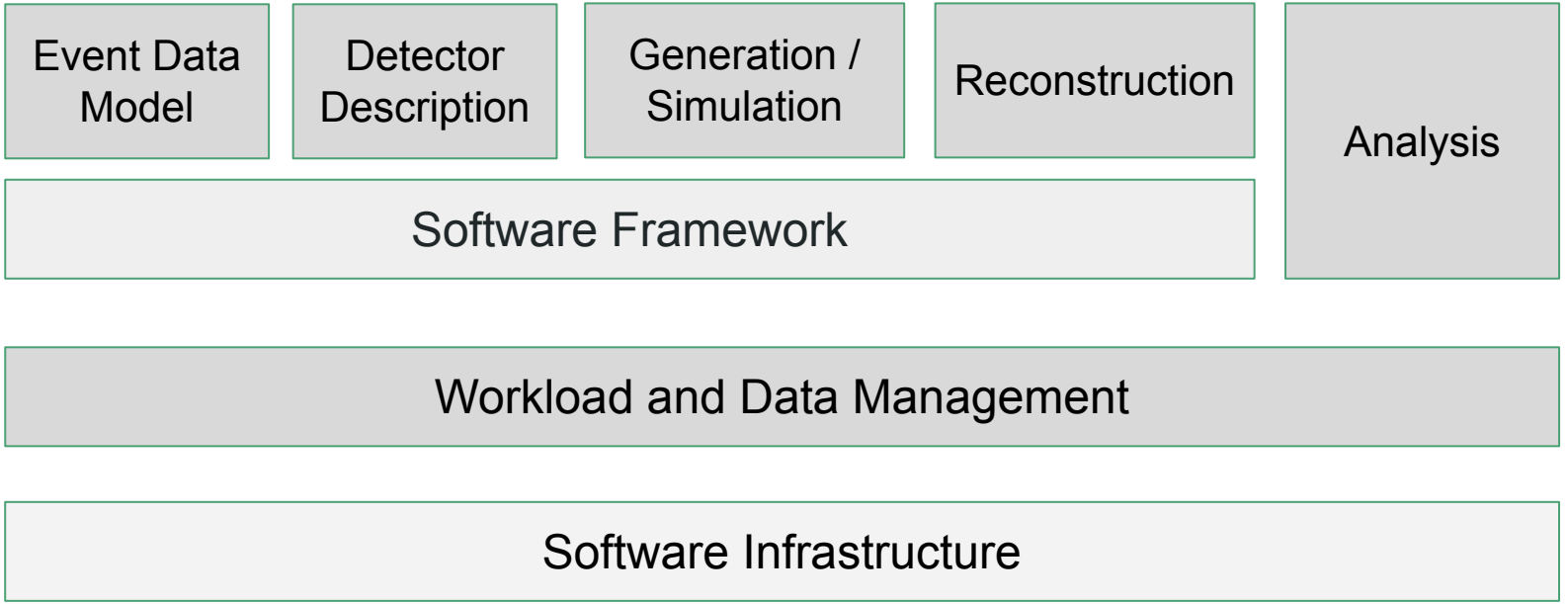
Next level required for the Technical Design Report, with a special focus on FCC-ee is much more detailed studies

FCC Software Outlook

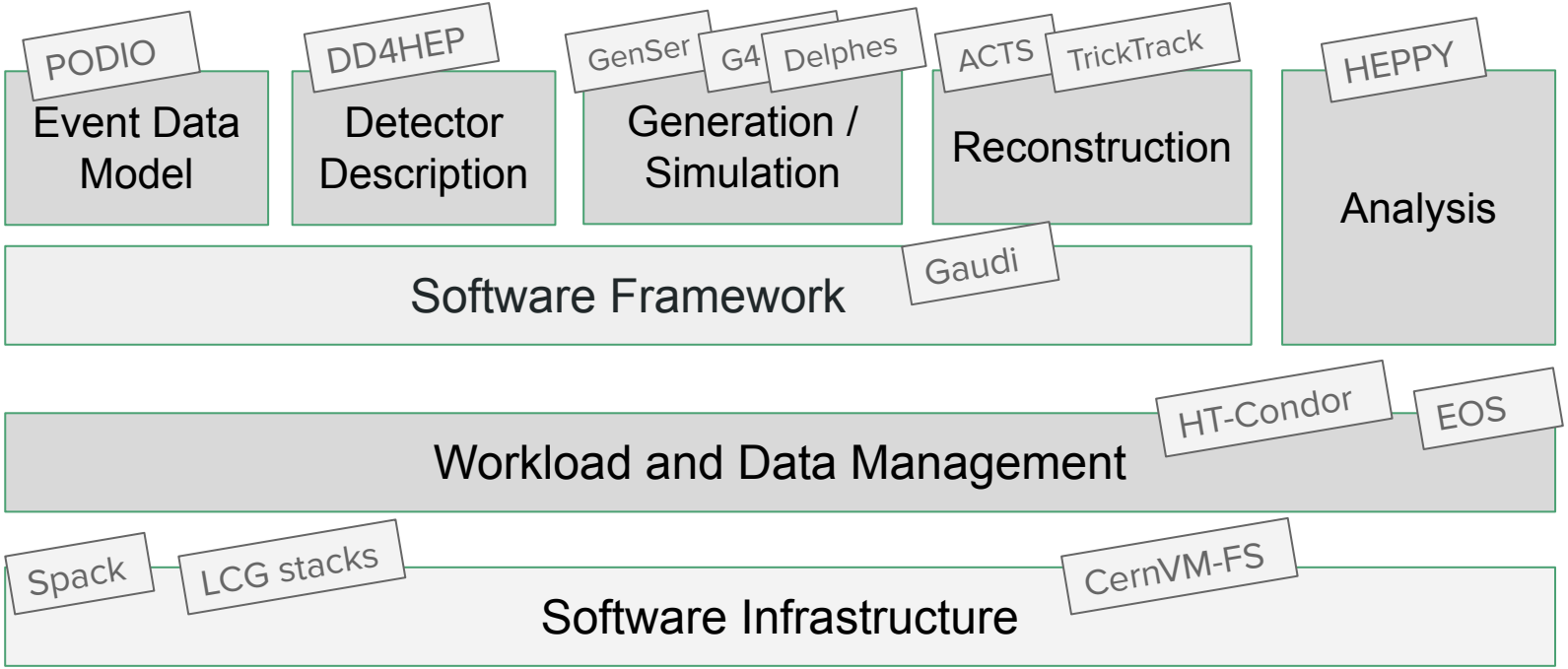
- Adopted Strategy
  - Adapt existing solutions from LHC
  - Look at ongoing common projects (AIDA)
  - Invest in streamlining of event data model

- Future: 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, ...

FCC Software Current Ingredients



FCC Software Current Ingredients

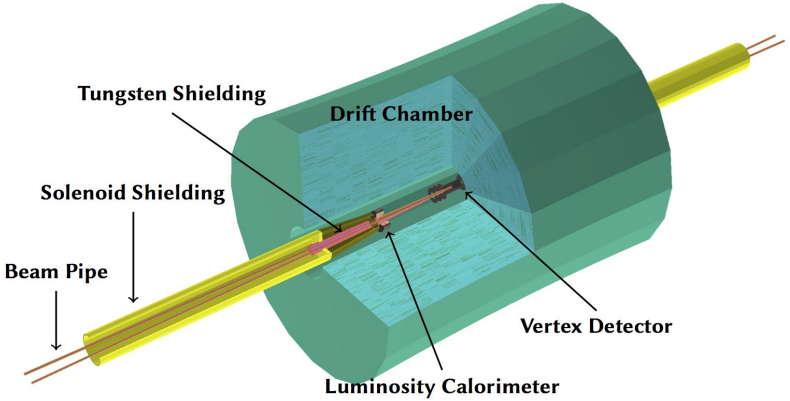


Event Data Model

- Current FCC-EDM
  - Event/Run: EventInfo
  - MC truth: MCParticle, GenVertex, GenJet
  - Tracker: Track (PositionedTrackHit, TrackCluster, TrackState)
  - Calorimeter: CaloCluster (PositionedCaloHit)
  - Associations: ParticleMCParticleAssociation, DigiTrackHitAssociation, CaloHitAssociation, CaloHitMCParticleAssociation
  - High-Level objects: TaggedParticle, Vertex (WeightedTrack), TaggedJet, ResolvedJet, MET
- Tuned on the needs of FCC-hh
  - High-level objects of LHC inspiration
- TaggedParticle contains cross association between tracks and calo objects

The Fundations

- Common detector description

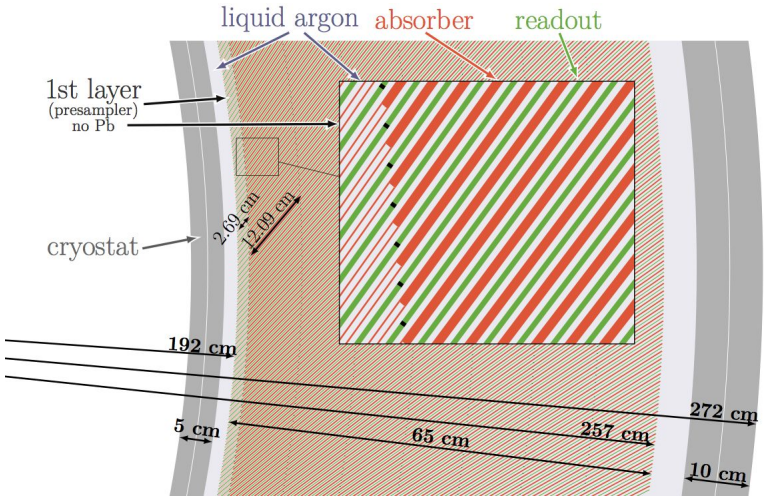


- Common Data Model

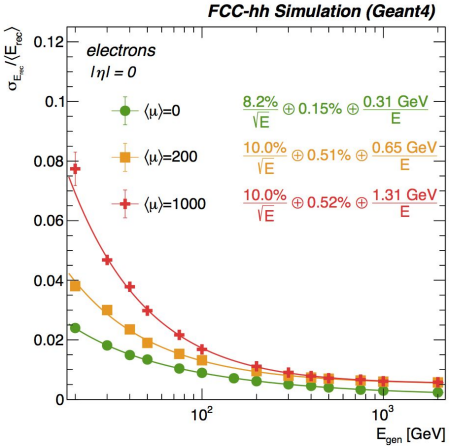
```
fcc::Track:
    Members:
     - float chi2 // from track fit
     - unsigned ndf //
     - unsigned bits // stores flags
    OneToManyRelations :
     - fcc::PositionedTrackHit hits
     - fcc::TrackState states
```

Calorimeter Reconstruction

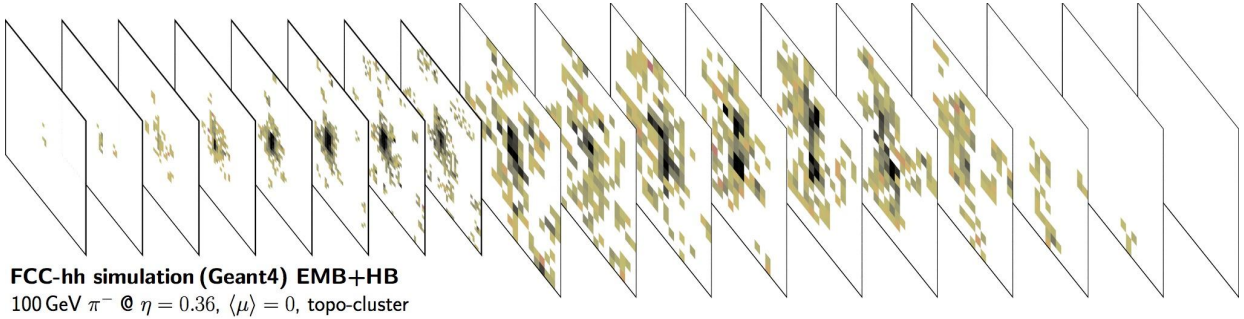
- Example: ATLAS-inspired LAr ECal for FCC-hh
  - Being studied for FCC-ee
  - More details in Martin Aleksa's talk this afternoon

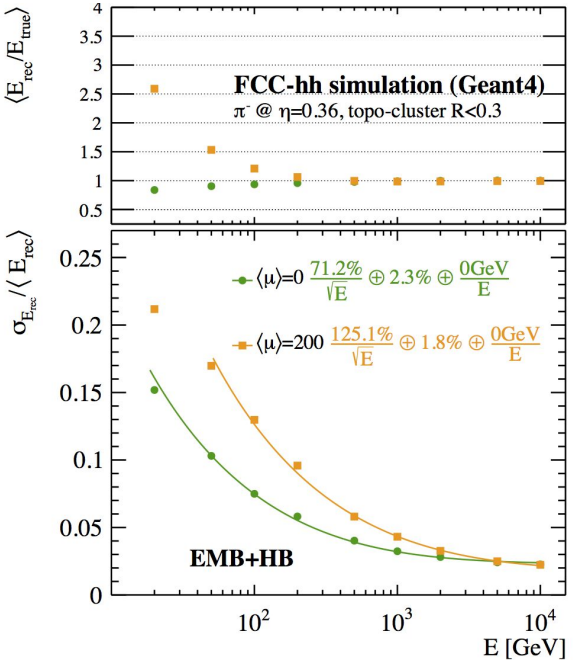


- Reconstruction code in FCCSW
  - Sliding window
  - Topo-Clusters
  - Noise/Pileup



Calorimeter Reconstruction: Topo-Clustering

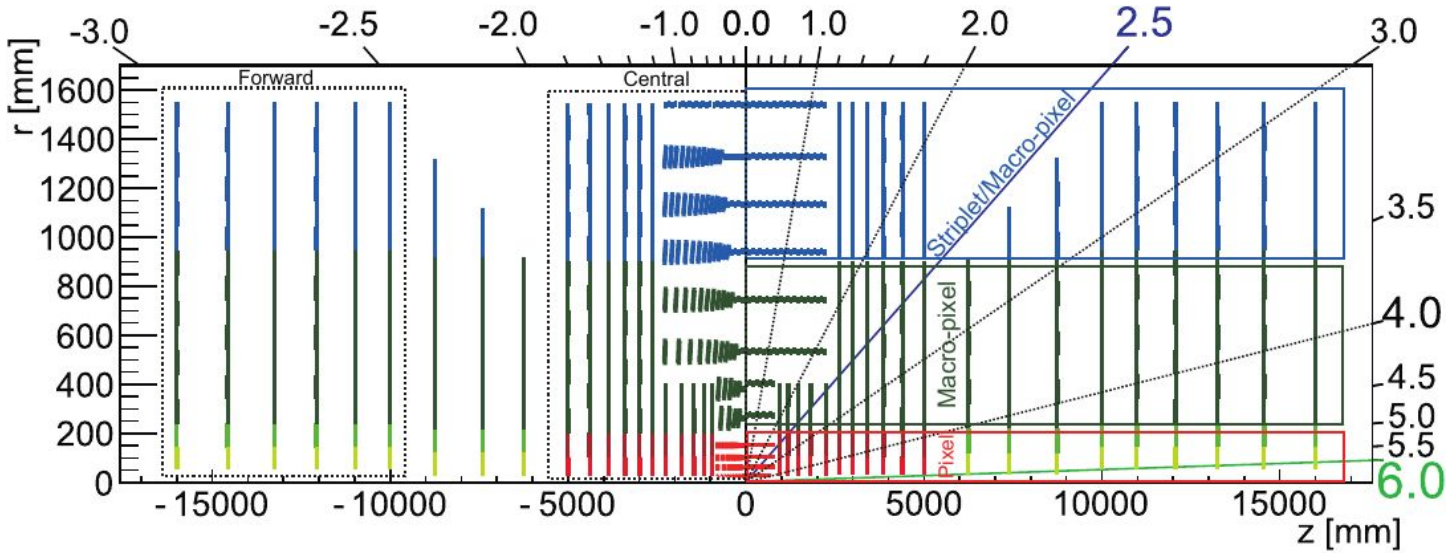




- Performance on single pions
  - Strong degradation with Pile-up
  - Topo-Cluster algorithm alone not sufficient for pile-up rejection (at FCC-hh)
  - Need more sophisticated techniques (Particle Flow)

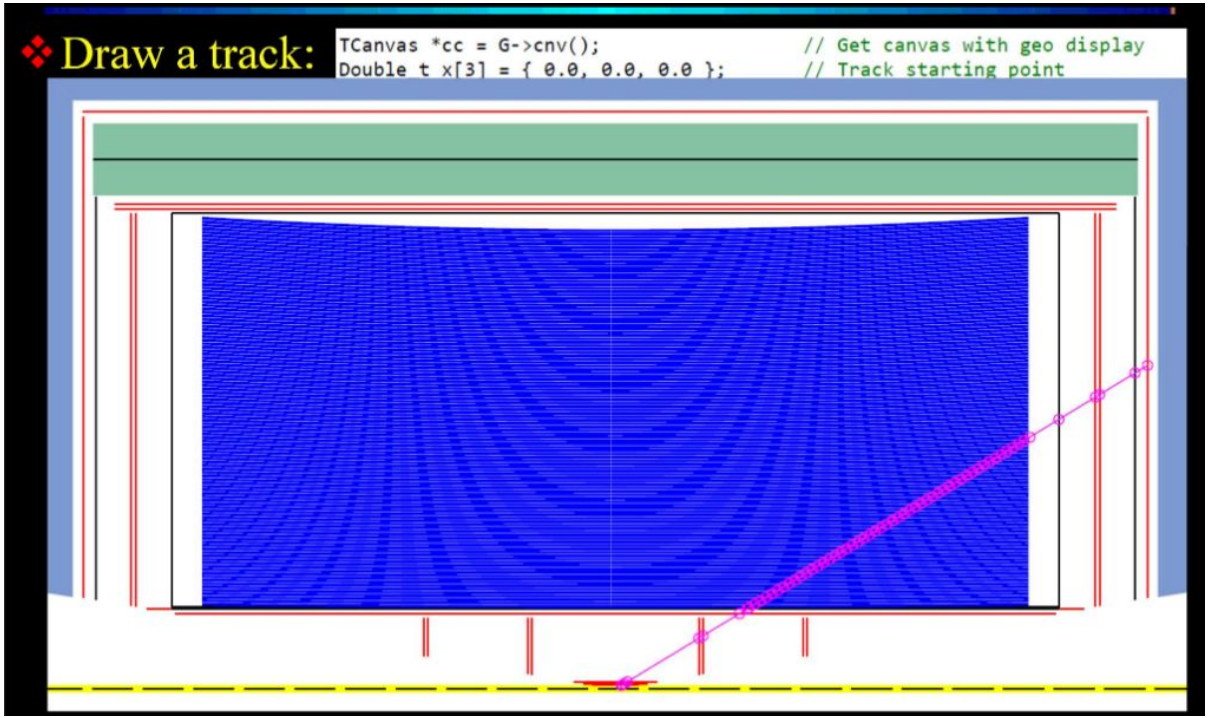
tkLayout

- Description
  - Used to study silicon detector layout for FCC-hh
  - `fcc-tkLayout` includes track analysis code and geometry export to DD4HEP



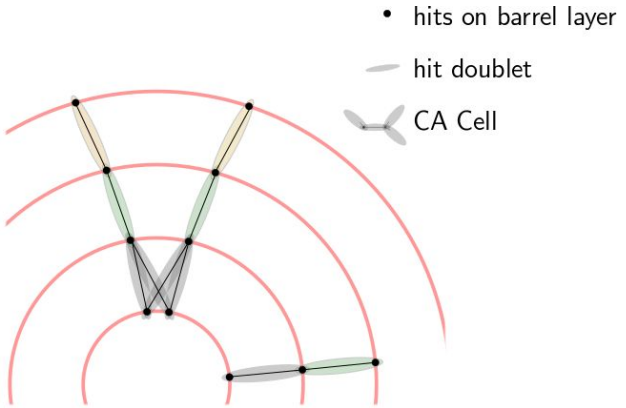
TrackSim

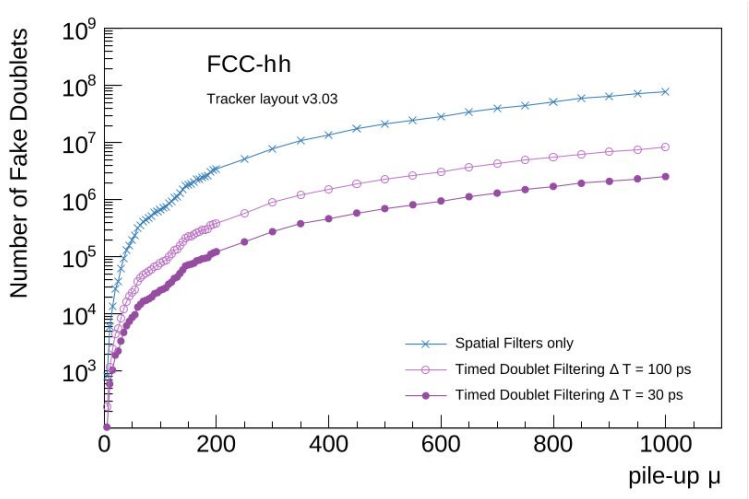
- F. Bedeschi : Root Macro for fast tracking sim in ee
  - https://www.pi.infn.it/~bedeschi/RD_FA/Software



Track Seeding

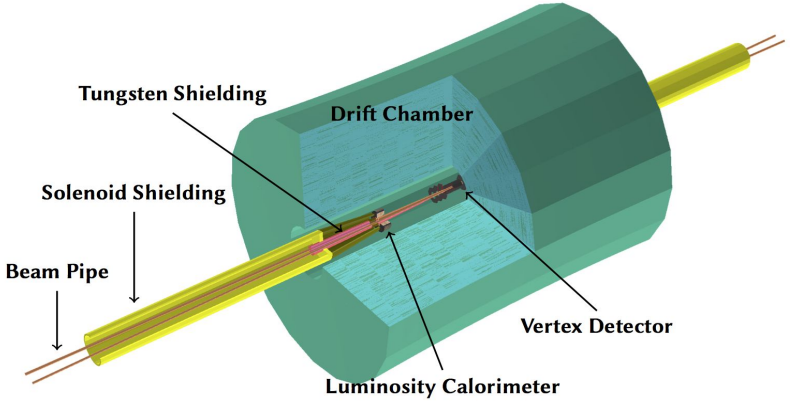
- TrickTrack
  - Standalone library with cellular-automaton code from CMS
  - Used for timed tracking studies in FCC-hh
  - Now hosted by the HEP-Software foundation

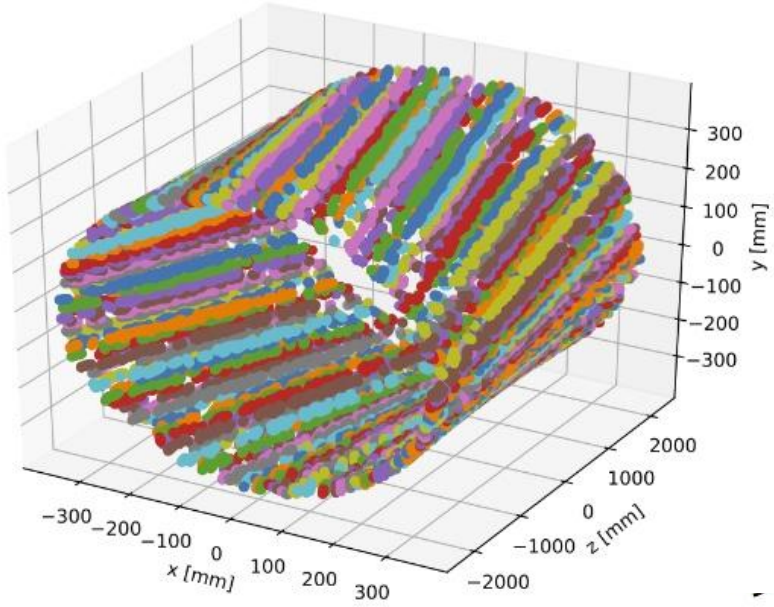




Drift Chamber Track Reconstruction

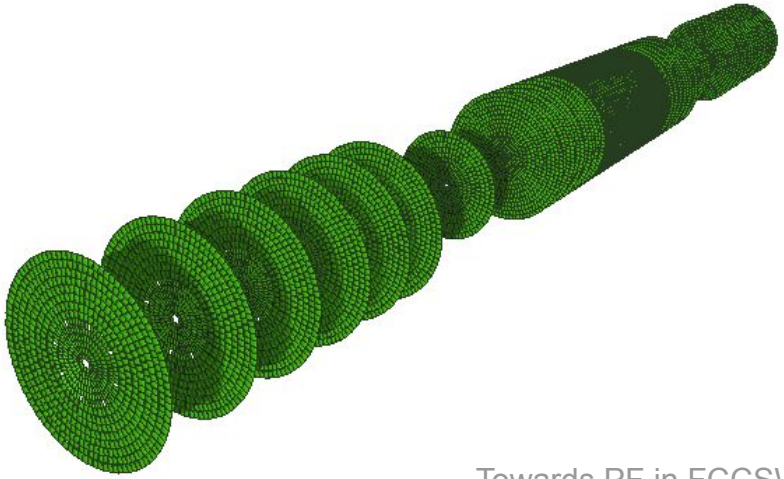
- Within Partial IDEA detector implementation in FCCSW
  - Hit reconstruction on each wire
  - Hough transform for tracking Python implementation in FCCSW
  - ILCSoft conformal tracking to be used in the future

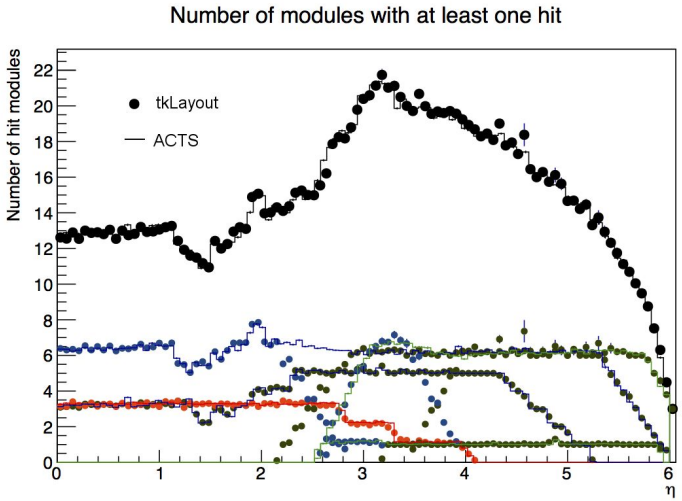




ACTS: A Common Tracking Software

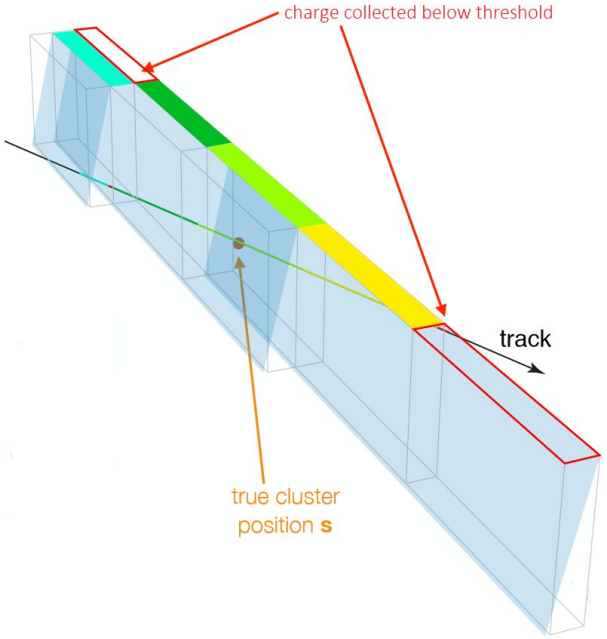
- Description
  - ACTS is integrated in FCCSW and used to building tracking geometry, digitisation and track extrapolation for FCC-hh
  - ACTS and DD4HEP now talk to each other
  - Following the future developments closely





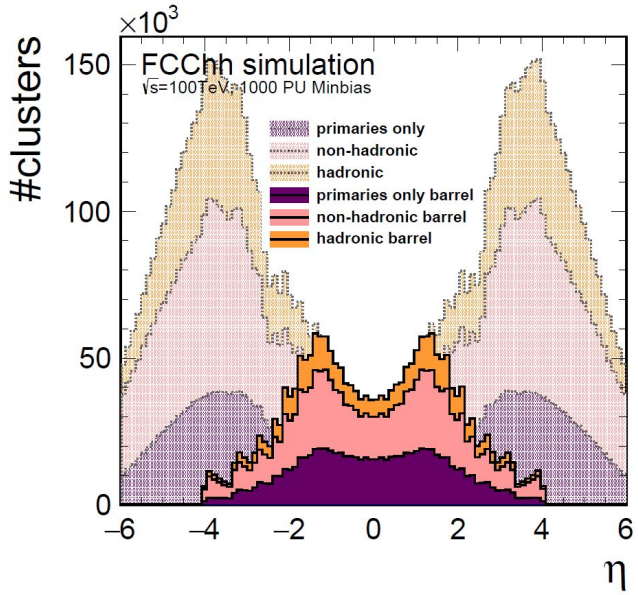
Digitization using ACTS

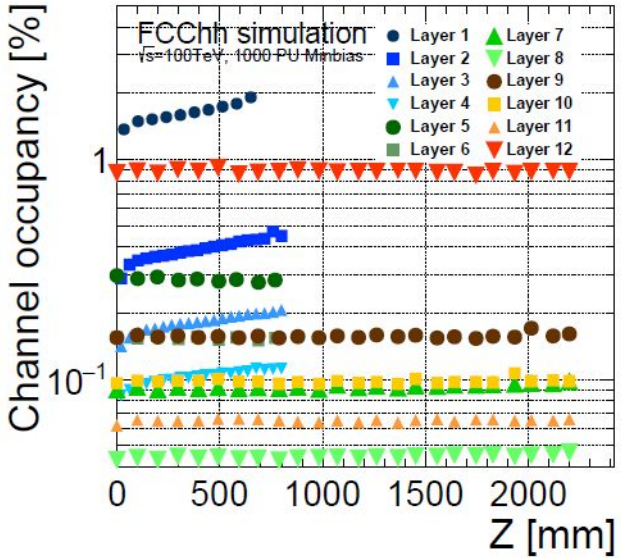
- Translate Hit into a measurement
  - Currently purely geometric (technologies not know yet for the detectors)
  - Followed by clustering
  - DD4HEP developers want to include some kind of generic digitisation (to be followed up)



Digitization: FCC-hh performance

- Tracker performance studies @ <μ>1000
  - In total ~10^7 clusters, ~$3x10^7$ activated pixels
  - 2-3 PB/s @ first trigger level (assuming binary readout @40MHz)

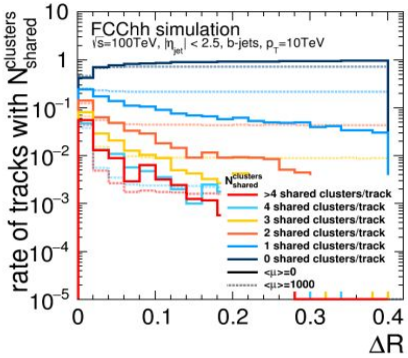


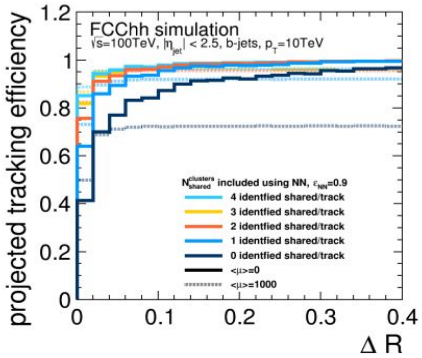


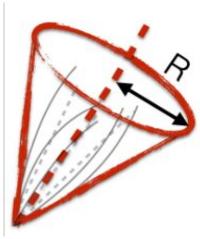
FCC-hh jet Reconstruction Studies

Tracker:

Rate of shared clusters along a track as a function of the distance from the Jet axis

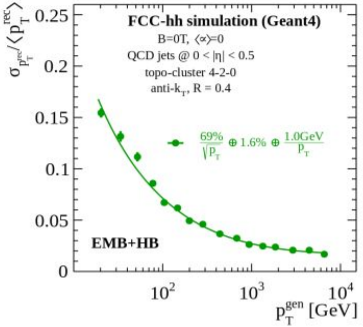




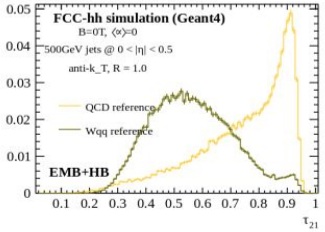


- Close by tracks produce single (thus shared) clusters
- Decreased tracking efficiency in jet core
- DNN needed for shared cluster id
- Otherwise 40% efficiency at jet core

Calorimeters:



Jet energy resolution of the barrel calorimeters standalone



Jet-substructure studies for collimated high pT jets started … see talk by C. Neubuser

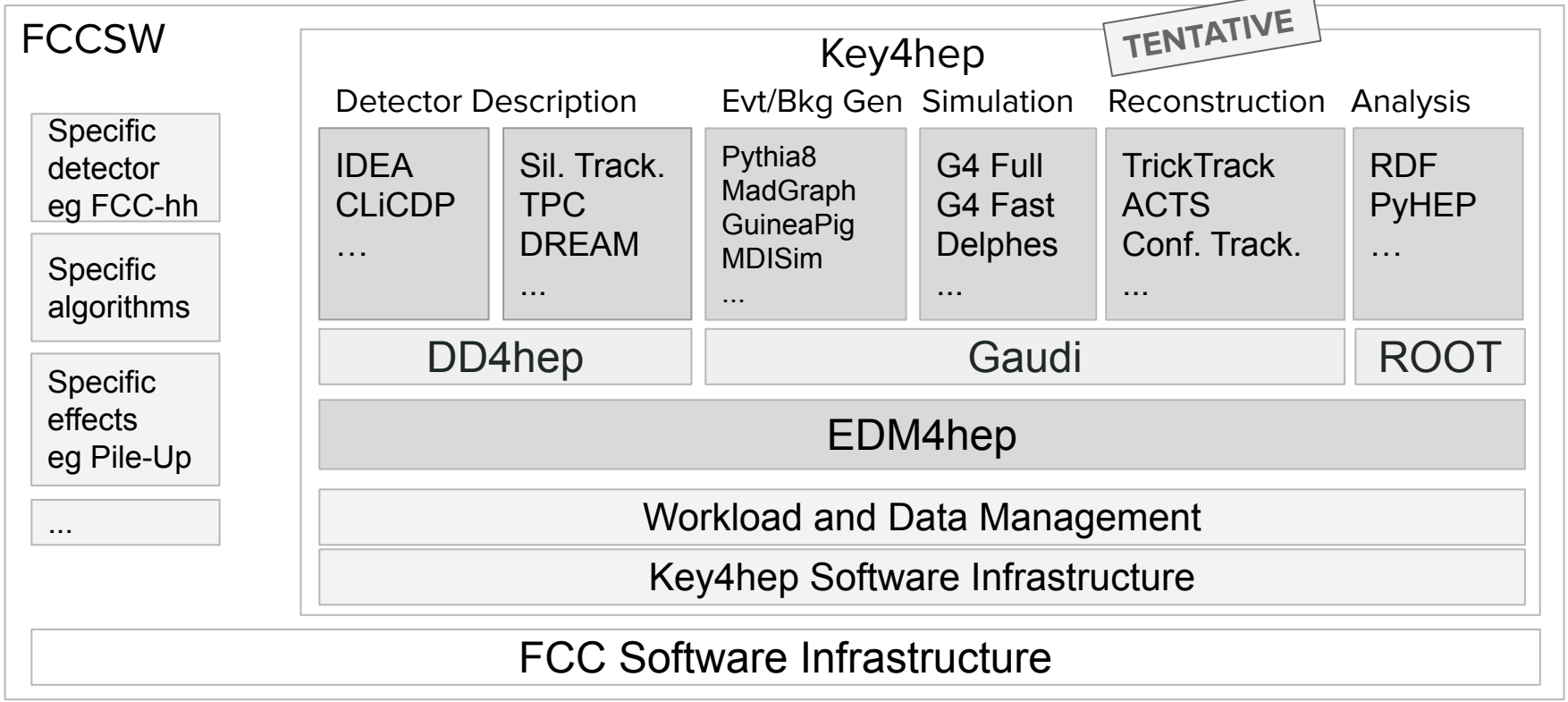
About Key4HEP

Key4HEP

- Depends crucially on EDM4HEP
  - Sets the chronological order
- Full-time fellows starting now (Jan 2020, Feb 2020)
- Possible contribution form AIDA++

- Possibly rapid development when EDM4HEP is available
  - Key4HEP core ≈ FCCSW core + EDM4HEP
  - Algorithms (FCCSW, other) adapted to EDM4HEP can then be used
- Deliver early and often approach

Connection with Key4HEP



After-CDR for FCCSW

- Current approach seems adequate to fulfill the requirements
- Contribution from FCC institutes is essential and very welcome
  - Interested people should manifest themselves and pick-up an area of work
  - Public software coordination meetings will be resumed soon (see next)
- Foster activities such as Key4HEP, which formalize and extend FCCSW approach
  - Exchange / share effort within the community

Next Steps: Calo-Tracker matching

- Uniformise the tracking in FCCSW
- Validate the calorimeter implementations and reconstructions
- Start with simple matching studies of single particles
- Define benchmarks for assessing the performances
- Single photon/π^0 rejection
- Particle identification
- Define a common platform to compare the performances of different detector implementations
- Once FCCSW and ILCSoft talks to each other through PODIO/LCIO wrapper, could in principle use the ILCSoft tools in FCCSW/key4HEP (PandoraPF)

Conclusions

- Reconstruction code in FCCSW has been driven by studies for the CDR
- Particle flow is essential to achieve the targeted performances in many detector scenarios
- No small task, collaborations is essential

- Current approach seems adequate to fulfill the requirements
- Contribution from FCC institutes is essential and very welcome
  - Interested people should manifest themselves and pick-up an area of work
  - Public software coordination meetings will be resumed soon
- Foster activities such as Key4HEP, which formalize and extend FCCSW approach
  - Exchange / share effort within the community

Areas of work summary

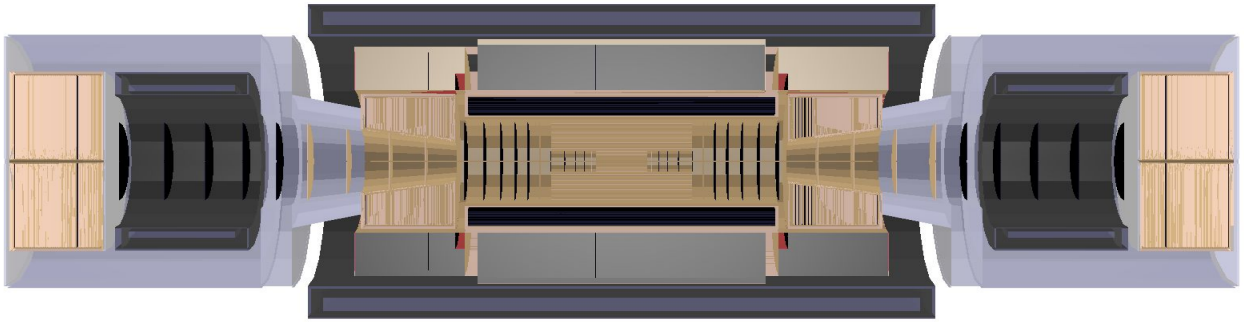
- MC generators interfacing
- MC generators testing
- MDI shared formats
- GuineaPig++ integration
- Overlay of MDI/signal events
- IDEA DR Calo full simulation
- IDEA Muon system full sim
- Validation of LAr Ecal for FCC-ee
- Enabling of CLD in FCCSW
- Validation/testing of Delphes cards

- Vertex reconstruction
- ACTS integration
- ML for calo reconstruction
- e, mu, tau, c, b tagging / ID
- RDataFrame based analysis
- Porting to other OSs
- ...

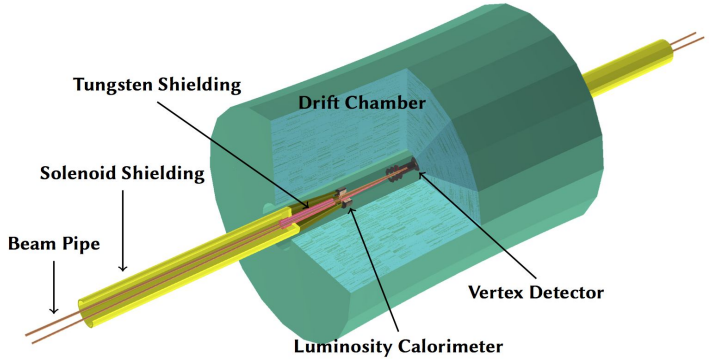
Backup

DD4HEP and FCCSW

- FCC-hh
  - Reference detector complete

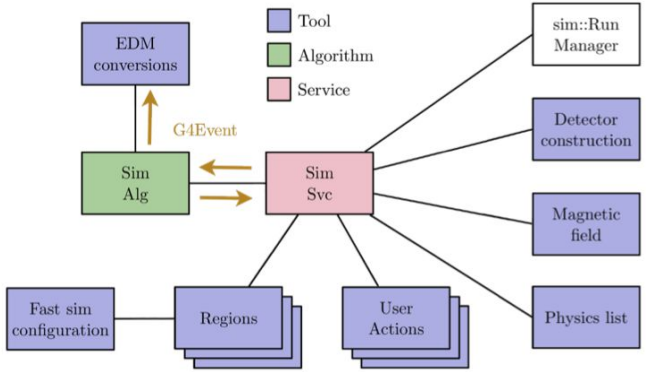


- FCC-ee
  - IDEA Concept
    - Beam Pipe, instrumentation
    - Vertex Detector, Drift Chamber
    - DREAM Calorimeter (under dev)
  - LAr+Tile calorimeter (under dev)
  - CLiCDP



Simulation

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



- Delphes
  - Gaudi interface
    - FCC EDM output

Reconstruction

- Challenges: algorithm detector concept independent
  - Full flexibility, avoid duplication
- Tracking
  - Track seeding (TrickTrack)
  - Hough Transform for drift chambers
  - Under implementation / investigation
    - ACTS integration
    - Conformal tracking
- Calorimeters
  - Sliding window (rectangular/ellipse)
  - Topo-clustering
  - Deep learning

Considerations for Physics Analyses

- HEPPY: High Energy Physics with PYthon
  - Modular python framework for the analysis of collision events
  - Developed and still used for CMS

- In FCCSW HEPPY is used to
  - Process EDM events, apply-preselection, produce a flat and light ROOT ntuple
  - Analyse the ROOT ntuple
    - Not the only code used for this purpose

- Flexible but slow
  - Plan to move to a C++-based analysis framework, e.g. RDataFrame

Software Infrastructure

- **Typical HEP development workflow**

- Deliverables
  - FCCSW
  - Externals: FCCSW specific dependencies
  - Based on LCG releases

FCCSW - Main package

FCC externals
fcc-edm fcc-physics tricktrack
heppy podio ...

LCG release
Gaudi dd4hep ROOT ...

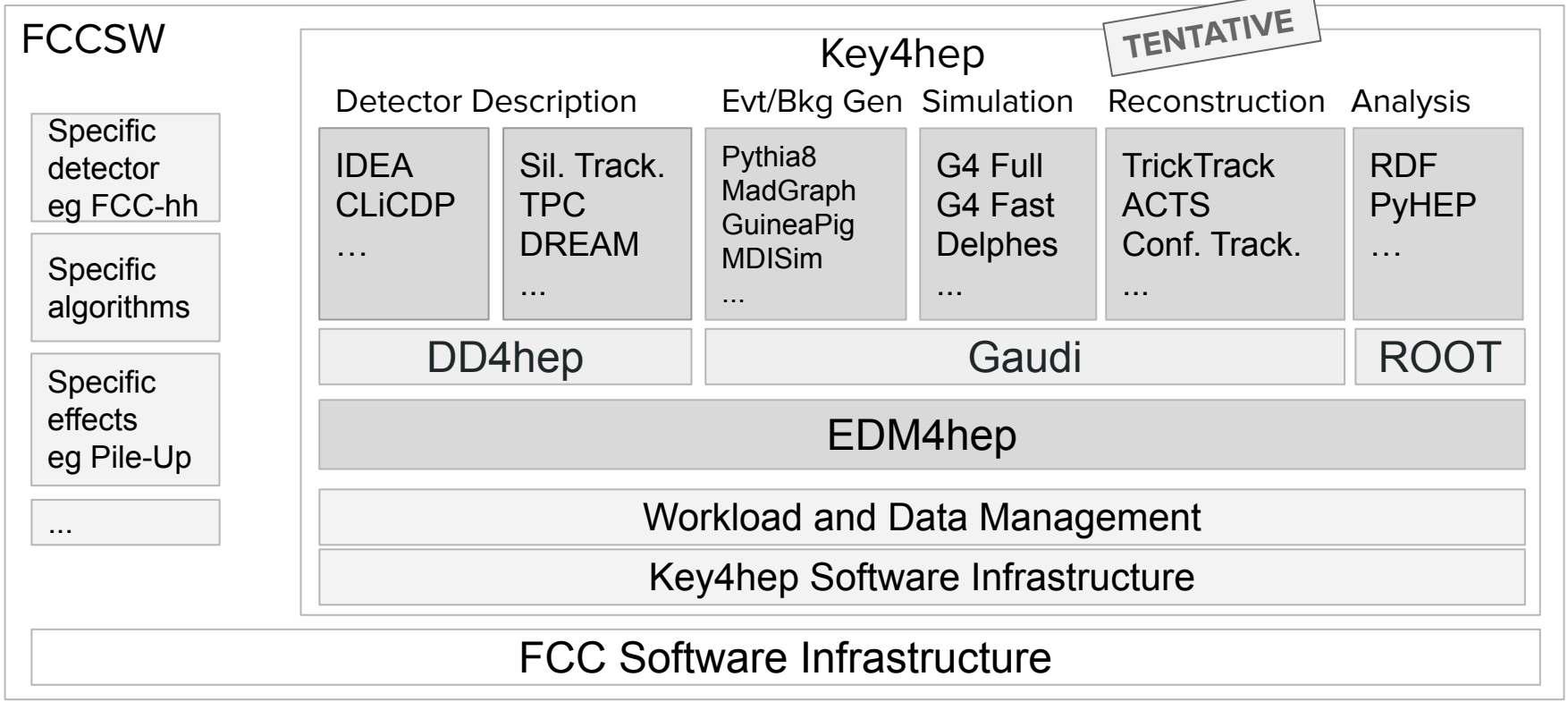
- Deployment on dedicated CernVM-FS repositories
  /cvmfs/fcc.cern.ch/, /cvmfs/fcc-nightlies.cern.ch/

- Builds (nightlies, releases) managed by **Spack**
  - Good feedback to HSF packaging WG

OS support and Resources

- Currently CERN-centric
- Support for the default version running on lxplus
  - CentOS 7, gcc 8
- EOS for shared files and data files
  - Shared: /eos/project/f/fccsw-web/www (also accessible also via web)
  - Data: /eos/experiment/fcc
- VM, based on CernVM, available to recreate equivalent environment
  - Works from everywhere but speeds depends on the network
- Access via notebooks (SWAN) also possible
  - Being used for tutorials

Connection with Key4HEP



How to contribute

- We are designing the software for future detectors and your help is welcome
- Proto-detector concepts have been made
  - Full/Fast/Parameterized simulations, Pen and paper detectors
- We need to finalised 2 to 4 of them towards detector technical studies in full simulation with full reconstruction (core software development at CERN)
  - Finalise/start geometry descriptions
  - Implement vertexing, flavour tagging, identification algorithms....
  - Assess the performances from common benchmarks
  - Start with interfacing generator
- Detailed tasks are being written and will be accessible soon

About the after-CDR for FCCSW

- Requirements:
  - Support more detailed studies, in particular for e^+e^-, focusing on
    - *Completeness*: state-of-Art generators, MDI, reconstruction / analysis algorithms, …
    - *Flexible detector description*: easy switch / replace sub-detectors, change dimensions / layout

- **Current approach** seems **adequate** to fulfill the requirements

- Foster and support
  - Participation from FCC institutes worldwide
  - Activities such as Key4HEP, which formalize and extend FCCSW approach

Software Coordination Meetings

Open software coordination meeting will restart on 31 January 2020

- Frequency: Bi-weekly meetings with remote connection
- Time slot: Friday morning 9h-11h
- Location: 40/R-B10
- Announced on fcc-experiment-sw-dev

Summary

- The FCC software stack
  - Assembled using as much as possible existing components
  - Served well the purposes of the CDRs

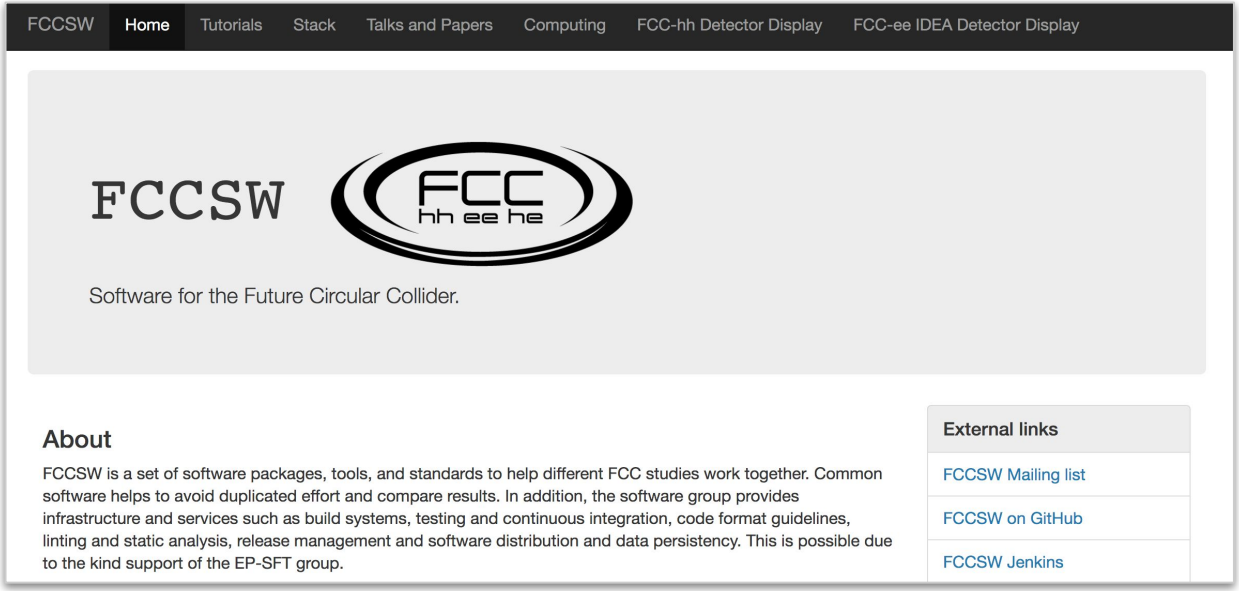
- Started a new phase to further develop to support more detailed studies, in particular for e^+e^-

- Areas of work identified

- Follow closely, participate and collaborate new common activities {Key4hep, EDM4hep}

Thank you!

- Web site https://cern.ch/fccsw
- Hands-on tutorials https://indico.cern.ch/event/839794/



Towards the TDR (1)

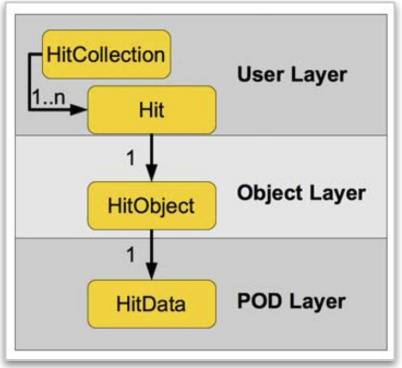
- Framework recontextualization
  - Identification of missing, underdeveloped parts
- Identification of Monte Carlo generators needed for physics studies
  - …
- State of Art MDI
  - …
- Provide ways to easy integrate new detector solutions proposed by the detector concept group
  - …
- Integrate digitization
  - …

Towards the TDR (2)

- Integrate reconstruction algorithms
  - …
- Integrate tools for physisc analysis
  - …
- Provide software satcks easy to use
  - …
- Ensure adequate computing resources
  - …
- Provide documentation
  - …

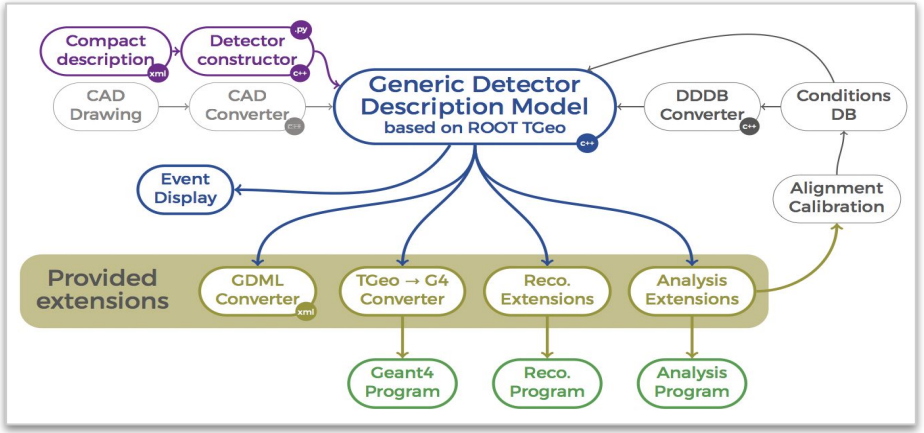
Event Data Model and PODIO

- LHC experiments / LC studies solutions suffering (partly) from
  - Overly complex data models with deep object-hierarchies
  - Unfavorable I/O performance
- **PODIO**: an EDM toolkit (AIDA2020 project)
  - Plain Old Data, automatic code generation, support for different backends
    - Keep memory model simple, enabling fast I/O and efficient vectorization
    - Consistent / homogeneous implementation, minimizes mistakes
  - High-level description in YAML format
  - Three-layers
    - User: handles objects and collections
    - Object: transient, relations between objects
    - POD: actual data structures
  - Follow-up for next AIDA being prepared
    - Schema evolution, memory and I/O optimizations



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



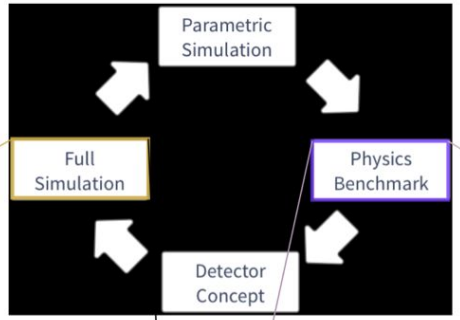
- Part of AIDA2020
- Used by CLIC, ILC, FCC, LHCb, CMS, SCT

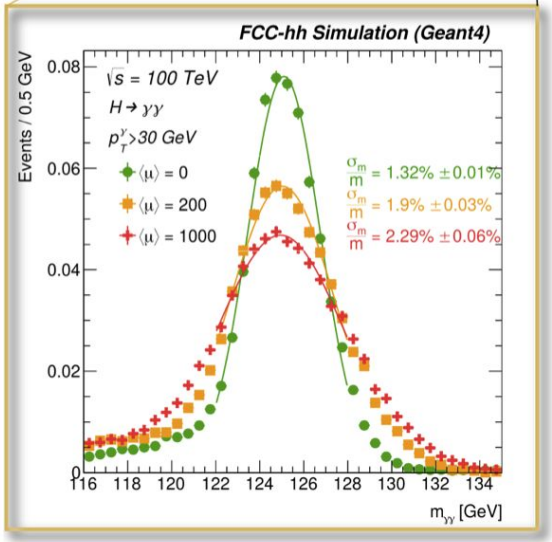
Fast / Full Simulation Interplay

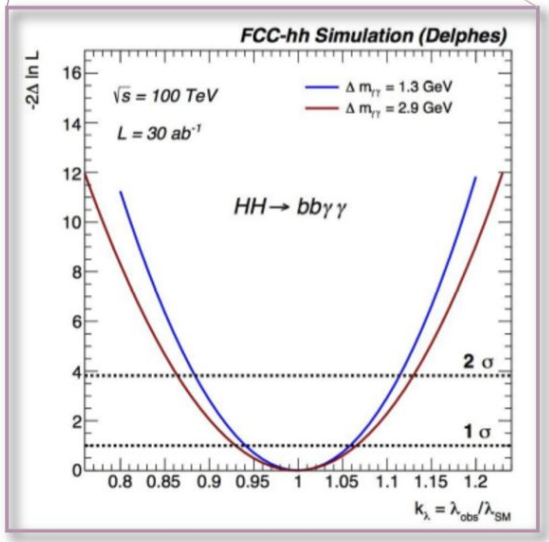
Example:

Higgs self-coupling

@ FCC-hh

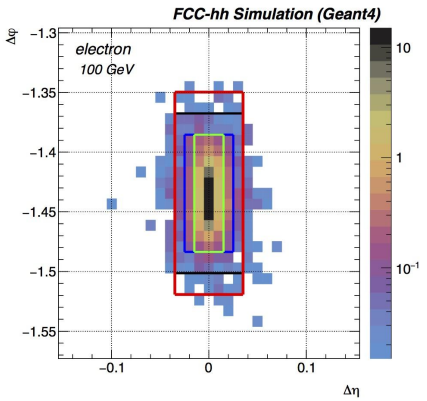






Calo Reconstruction Example

- Single 100 GeV e- reconstructed by sliding window



- Single 100 GeV pion in 8+10 layers of the E+HCal reconstructed by topo-cluster