Development of CEPC Software and Computing

Weidong LI, IHEP/CAS

representing CEPC software and computing team

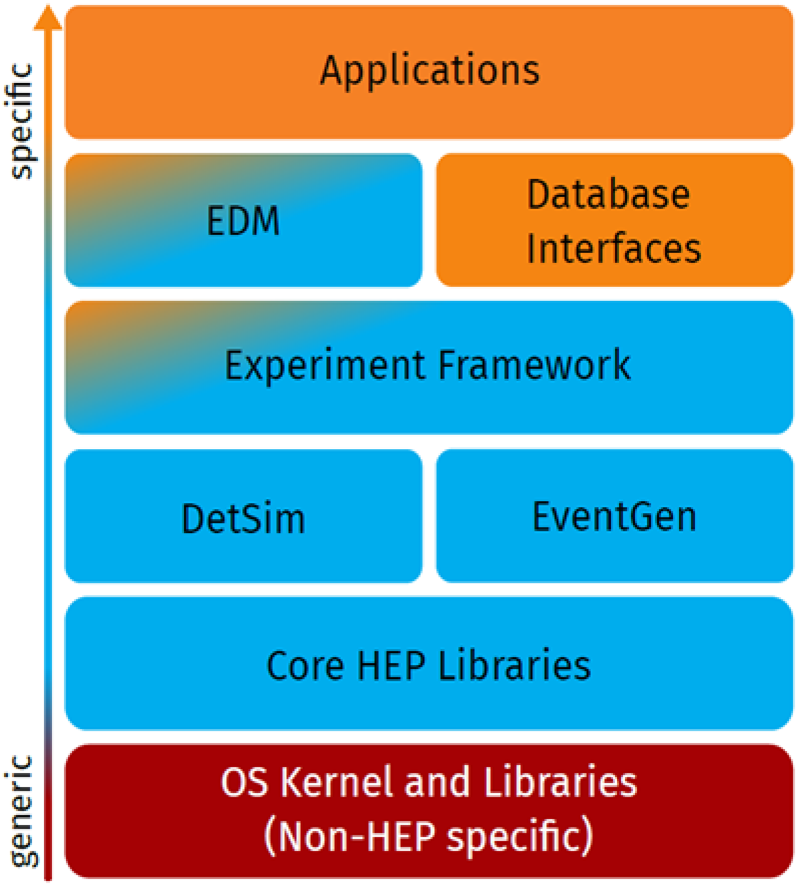
7th CEPC IAC meeting

2nd November, 2021

Contents

- Introduction
- Core software
- Simulation framework
- Detector algorithms
- CEPC computing
- Future plan
- Summary

Introduction

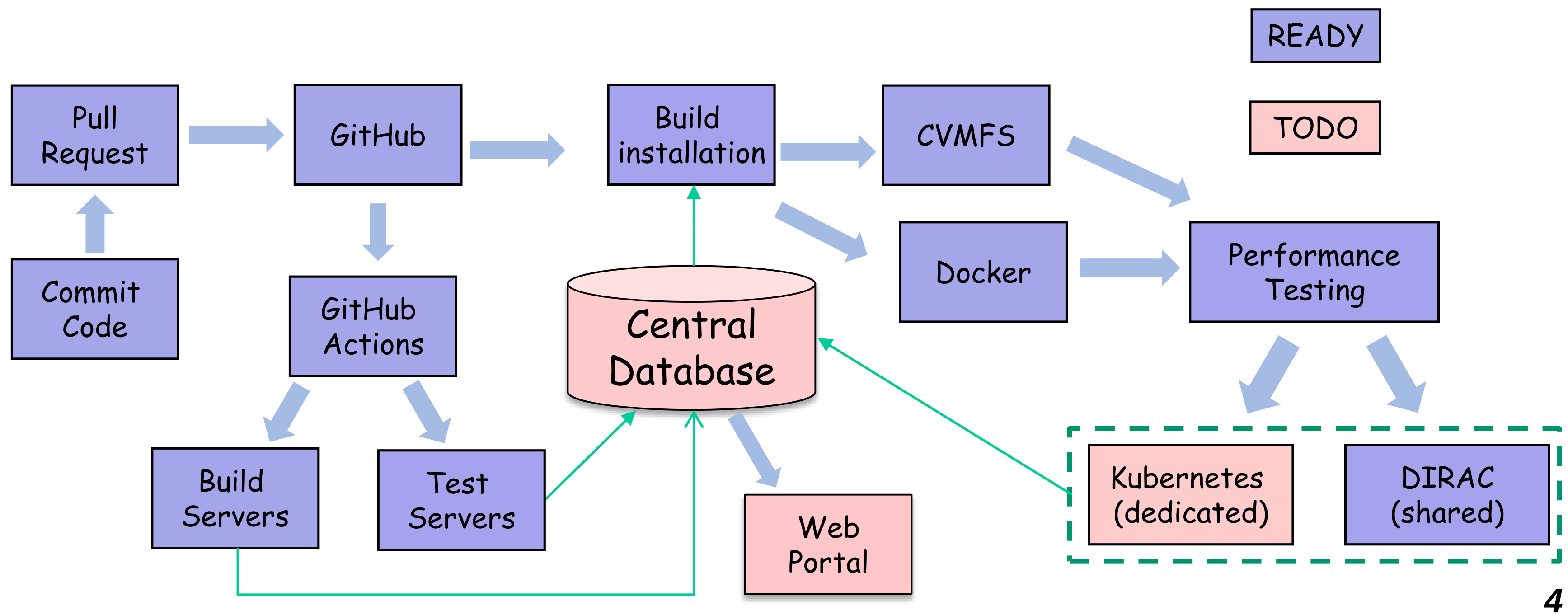


From Thomas Madlener, Epiphany Conference 2021

- The Key4hep is being developed to provide a common software stack for CEPC, CLIC, FCC and ILC experiments:
  - Application layer of modules/algorithms/processors performing physics task
  - Data access and representation layer
  - Experiment core orchestration layer
    - (Marlin, Gaudi, CMSSW, …)
  - Specific components reused by many experiments
    - (DD4hep, Delphes, Pythia, …)
  - Core HEP libraries
    - (ROOT, Geant4, CLHEP, …)
  - Commonly used tools and libraries
    - (Python, CMake, boost, …)
- IHEP and SDU are involved in Key4hep development as non-EU members.
- The CEPC software (CEPCSW) will be fully integrated with the Key4hep to share software with other future experiments.

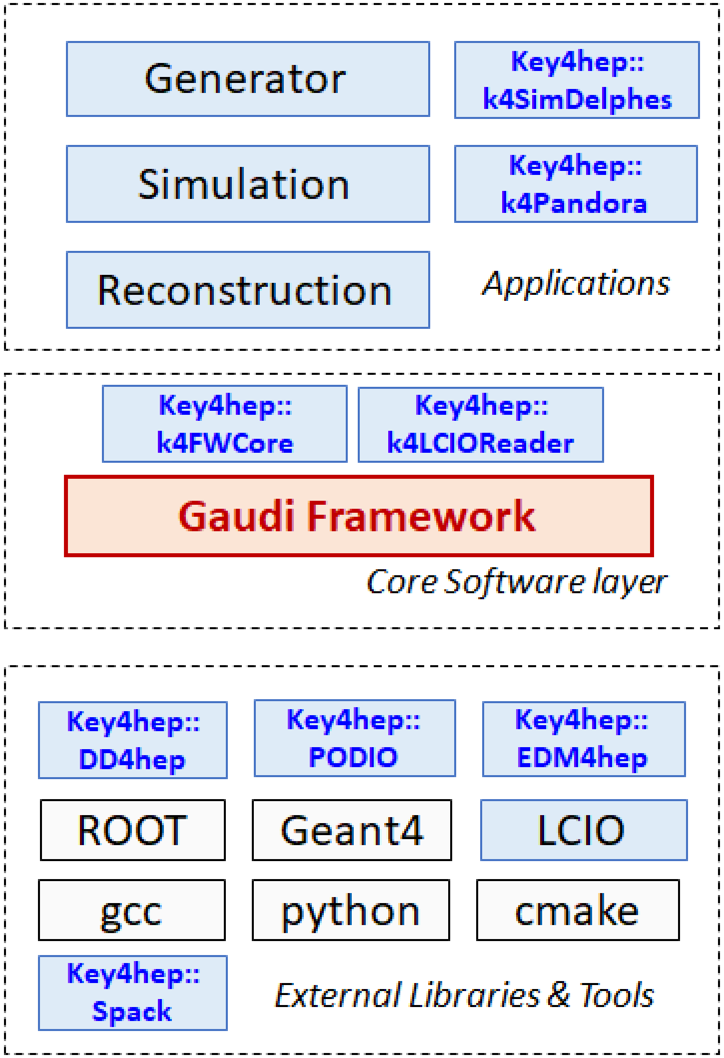
Development environment

- C++ 17 and Python 3 are the main programing languages and operation system used is CentOS 7
- GitHub is chosen as the source code repository and the pull-request mechanism is used to synchronize all developers' work.
- Automated validation system is being developed to run testing and validation automatically at different levels.



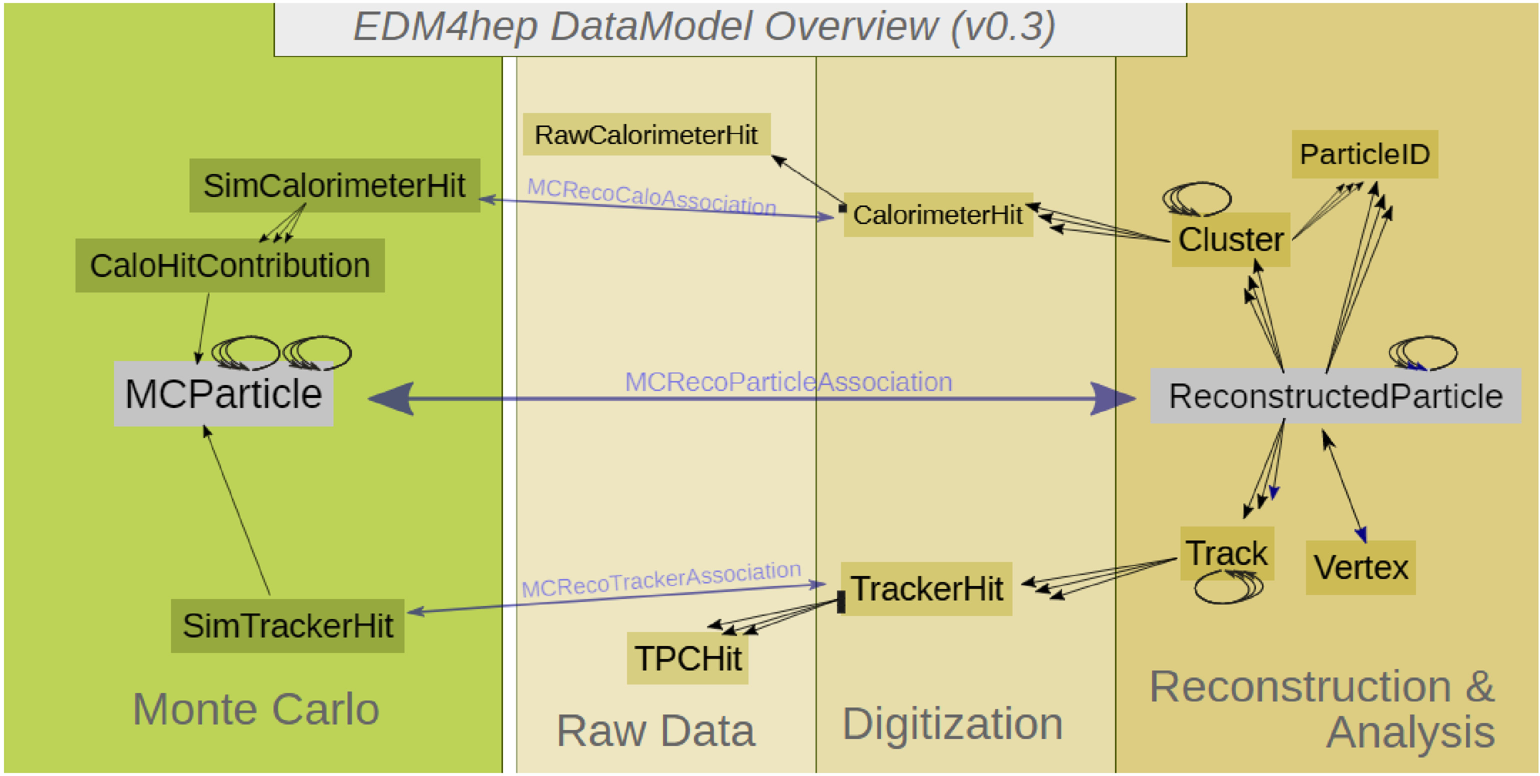
CEPCSW core software

- CEPCSW software structure
  - Core software
  - Applications: simulation, reconstruction and analysis
  - External libraries
- Core software
  - Gaudi/Gaudi Hive: defines interfaces to all software components and controls their execution.
  - CEPC-specific framework software: generator, Geant4 simulation, beam background mixing, fast simulation, machine learning interface, etc.
  - EDM4hep: generic event data model for HEP experiments
  - K4FWCore: manages the event data
  - DD4hep: geometry description



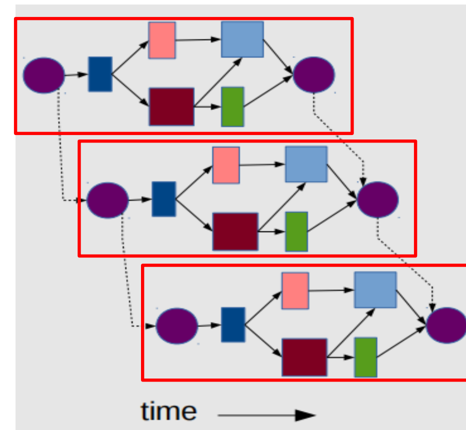
Event Data Model

- Adopted EDM4hep as the official EDM for CEPCSW
- k4LCIOReader was developed to convert the ILC format data to EDM4hep objects on fly
- Extension of the current EDM4hep to accommodate the needs from dN/dx studies of the drift chamber is in progress

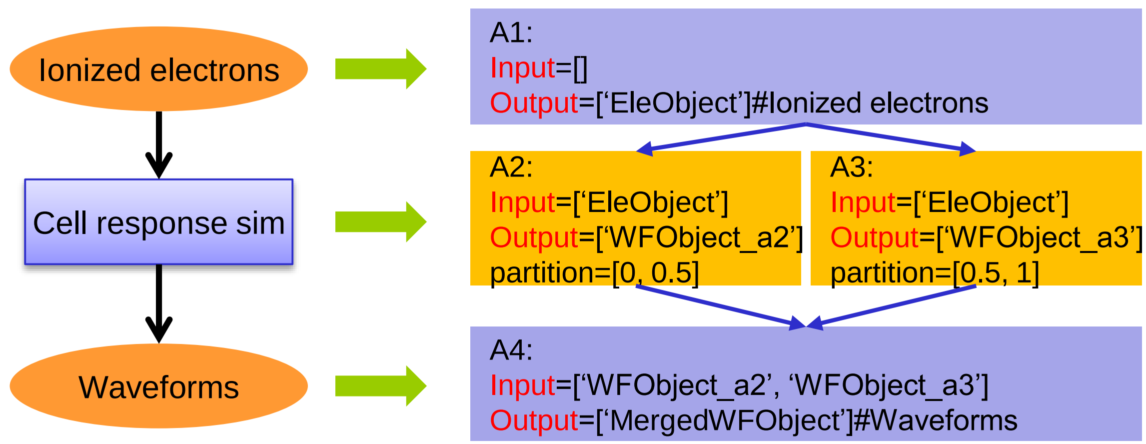


Multi-threading with Gaudi Hive

- Gaudi Hive is a Gaudi extension supporting multi-threading and concurrent computing
- Multiple algorithms and events can be executed simultaneously using the data flow driven mechanism
  - Algorithms declare their data dependencies
  - Scheduler automatically executes Algorithms as the data becomes available



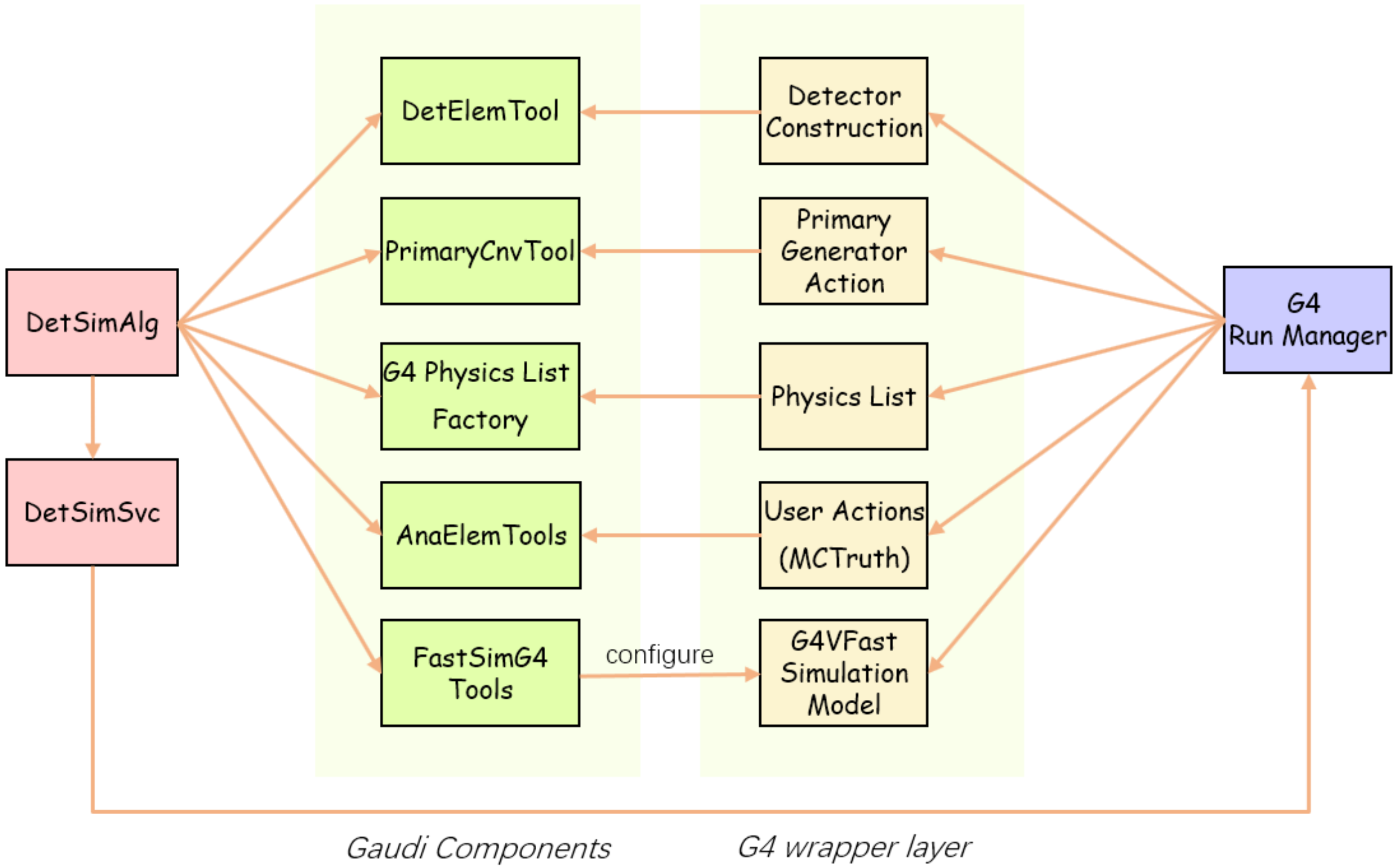
- The multi-threaded simulation of the detector response in drift chamber was developed as the first attempt to use Gaudi Hive.
- The multi-threaded simulation works well and it was reported at the Key4hep meeting.



https://indico.cern.ch/event/1076542/#4-gaudihive-in-cepc-driftchamb

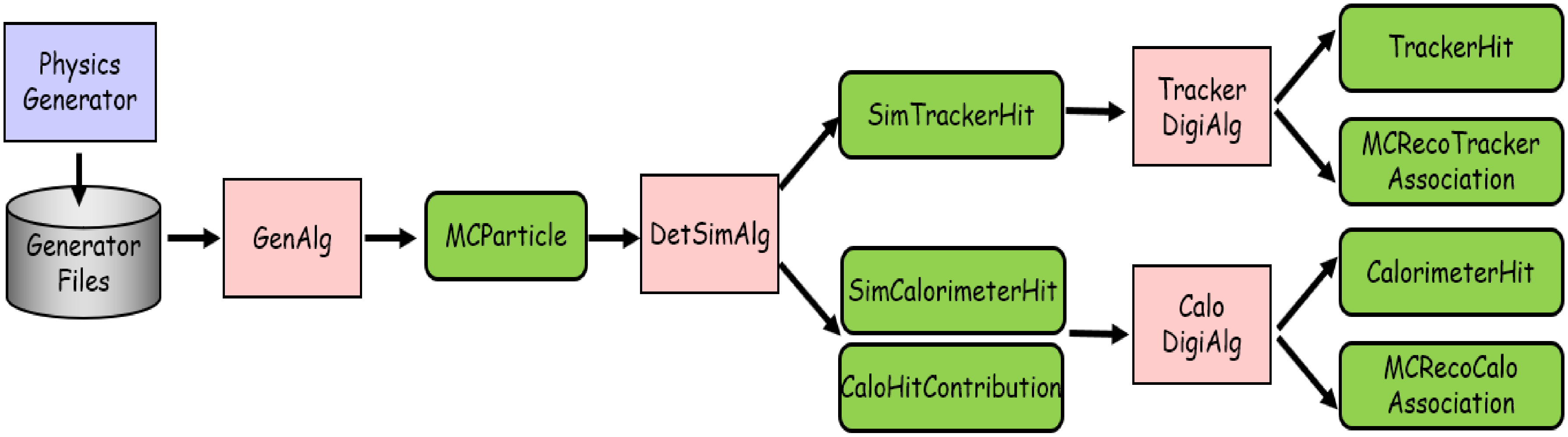
Simulation framework (1)

- The detector simulation framework has been developed in CEPCSW.
  - A thin layer is developed to connect Geant4 and Gaudi.
  - The event loop is controlled by Gaudi with a customized G4RunManager.
  - The geometry conversion from DD4hep to Geant4 is done by DDG4.



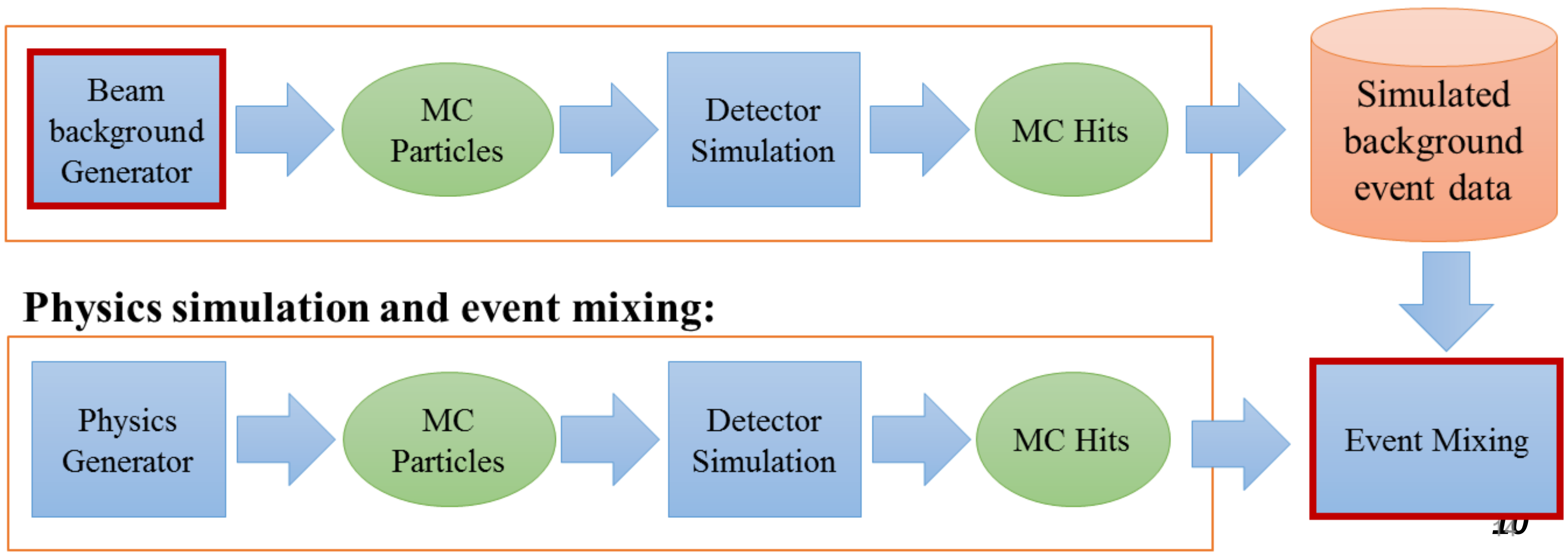
Simulation framework (2)

- The full simulation chain from physics generator to digitization is completed.
- Data objects as well as M.C. Truth information are available for detector performance studies:
  - Physics generator generates the kinematics information of primary MC particles
  - Detector simulation provides the relationship between MC hits and MC Particles
  - Digitization creates the association between the Digi objects and Hit objects



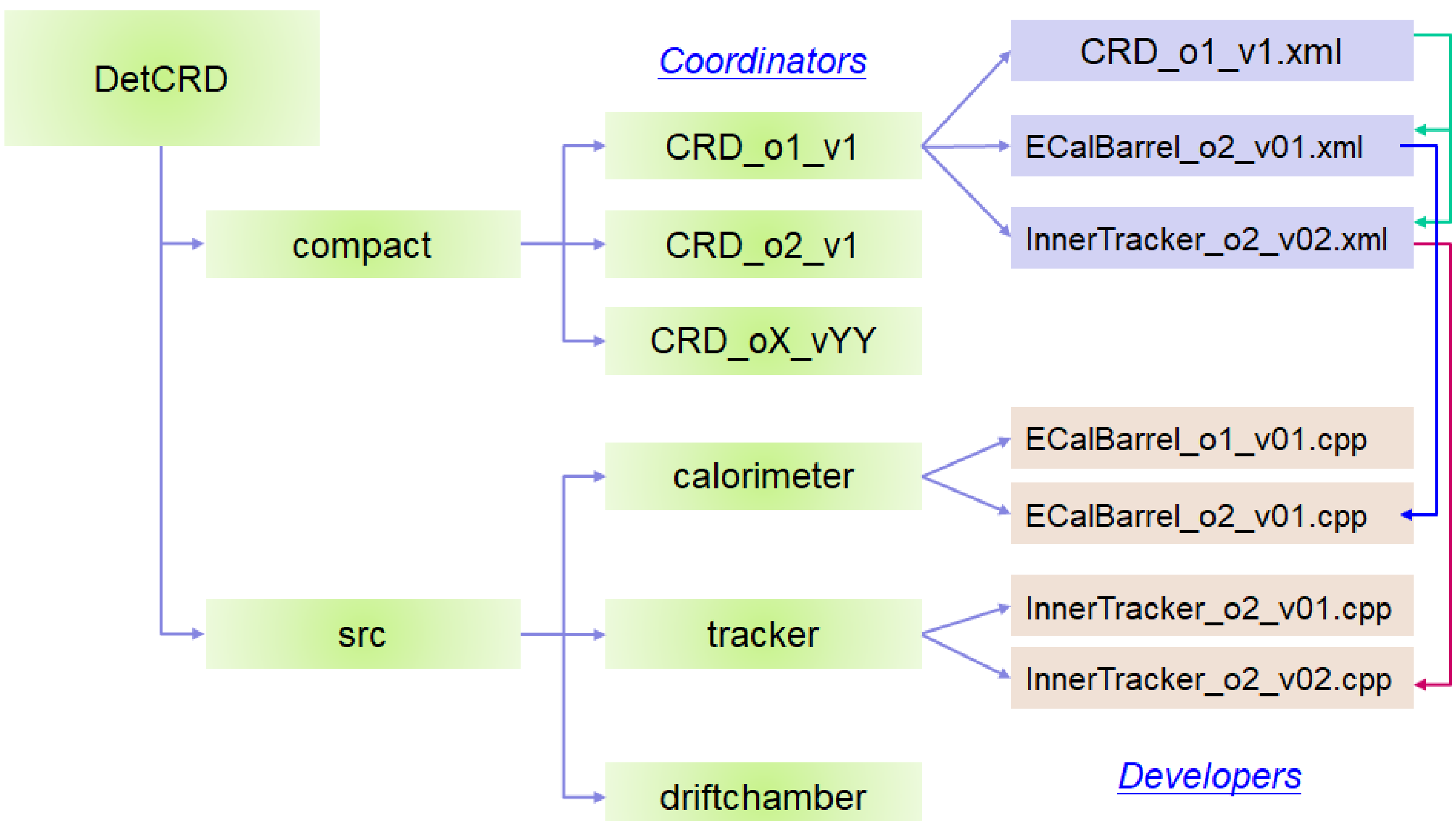
Simulation framework (3)

- DD4hep is used to provide a complete detector description with a single source of information
- The non-uniform magnetic field has also been implemented in the DD4hep framework.
- More realistic simulation needs to include beam-related backgrounds
  - The current design is to mix a physics event with backgrounds at MC hit level and implementation is in progress.



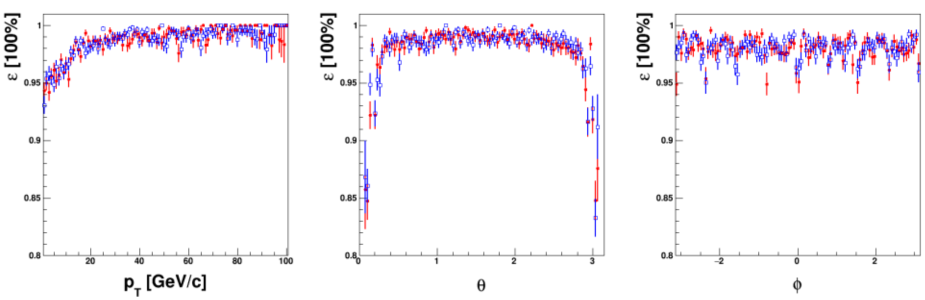
Simulation framework (4)

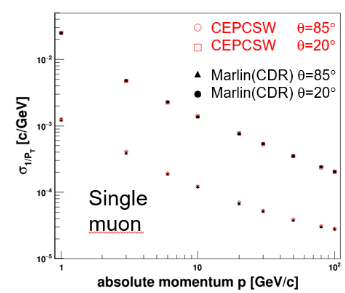
- Detector geometry management
  - A detector design option is defined by a compact file e.g. CEPC Reference Detector (CRD_o1-v1)
  - Details of sub-detectors are described by XML compact files and C++ constructors.

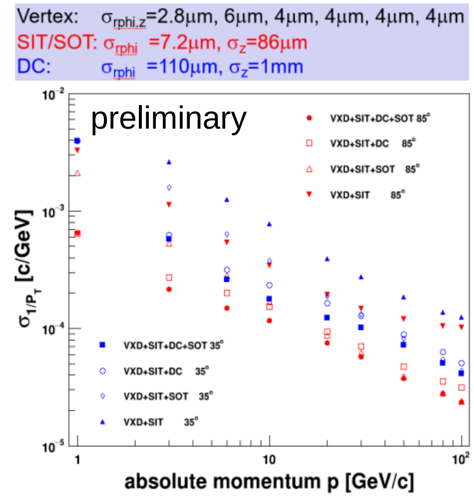


Tracking for Silicon Detector and TPC

- Migrated tracking and fitting algorithms from cepcsoft (ILDSoft based) to CEPCSW
  - Marlin→Gaudi
  - LCIO→EDM4hep
  - consistent performance
    - Marlin VS CEPCSW
    - (CEPCSW)simulated LCIO input VS generator simulating
- Fixed bugs to make running more stably
- Performance studies for the 4th conceptual detector
  - 6 pixel vertex + 4 pixel inside + drift chamber (regard as TPC hits in this fitting)+ 1 pixel outside
  - validate resolutions: similar tendency with fast estimation (<20%)
- Testing with non-uniform magnetic field
  - close resolutions with uniform field: $(\sigma_{Pt}-\sigma_{Pt,non})/\sigma_{Pt}$ ~4%@100GeV
  - momentum departure from MC truth, to correct

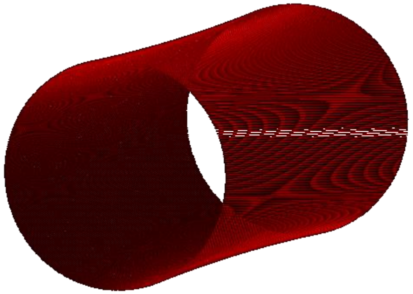




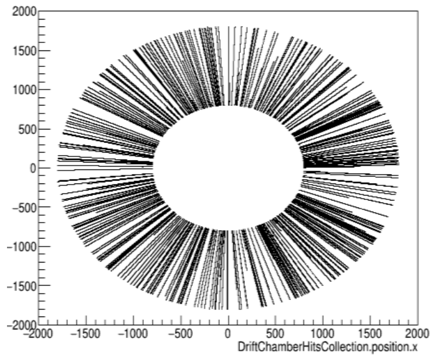


Simulation for the drift chamber

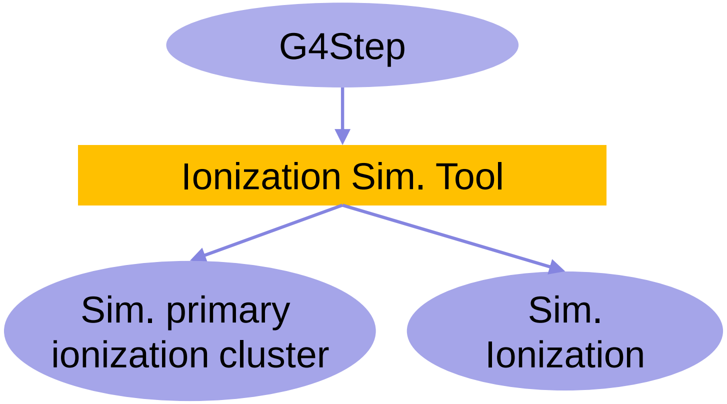
- Baseline configuration
  - Axial/stereo drift chambers with silicon layers
  - Radius **1~1.8**m, **100** layers, He:iC_4H_{10}=**90:10**
- Simple digitization
  - Constant X-T (V_{drift}=40μm/ns) and fixed spatial resolution (110 μm)
- Integration of Garfield++ with Geant4:
  - For each G4Step, Heed is used to simulate ionization process. The kinetics of G4Track will be updated according to its energy loss.
  - Tracking with Geant4 then continues and Garfield++ will take charge of simulation of the detector response in the cell.



Stereo layer of drift chamber



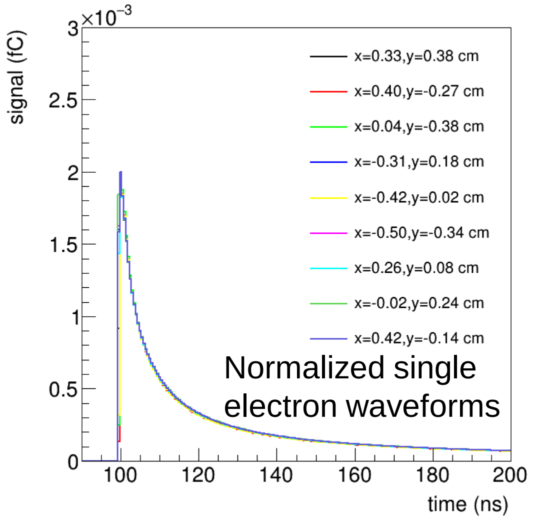
Hitmap of MC hits in DC

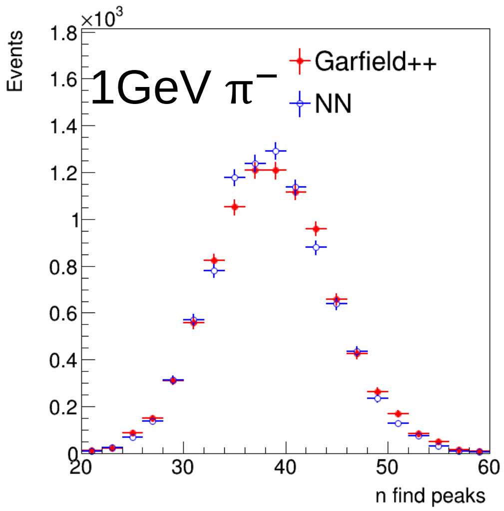


Integrating Garfield++ and Geant4 at G4Step

Fast waveform simulation

- Extremely time consuming to use Garfield++ to simulate
  - Drift of ions and electrons, amplification via electron avalanches and final signal generation
- Studies show that the waveform shape of each ionized electron in Garfield++ is similar. Main difference is the beginning time and amplitude
- Using machine learning technique to learn the distributions of beginning time and amplitude for each ionized electron
  - Training sample is produced by Garfield++

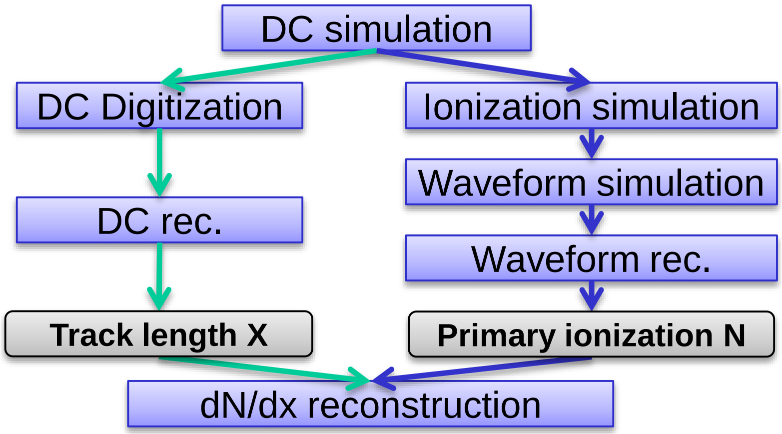


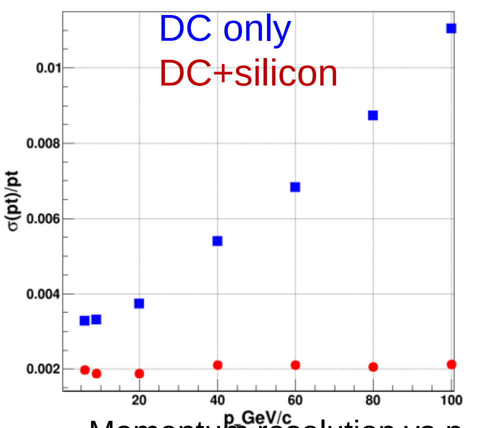


- ❑ Good agreement between NN and Garfield++
- ❑ ~200 times speed up

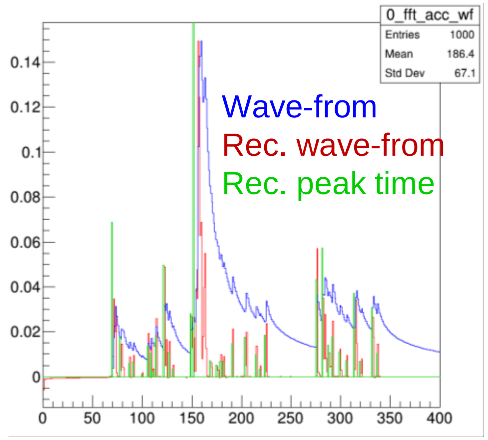
Reconstruction for the drift chamber

- Track finding
  - Truth tracking: track finding using MC truth information
  - Traditional and machine learning based tracking have been planned
- Track fitting
  - Genfit-based kalman filter was used to handle material effects and non-uniformity of B field correction and produce track parameters
  - Fitting to space points/wire/pixel/strip on a track has been tested and reasonable performance was obtained.
- dN/dx reconstruction
  - Waveform reconstruction algorithm with Fourier transform method was imported from JUNO





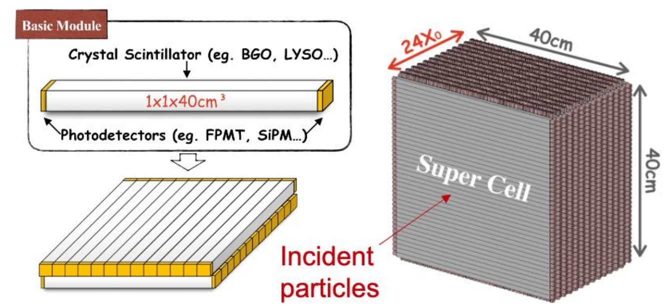
Momentum resolution vs p_T



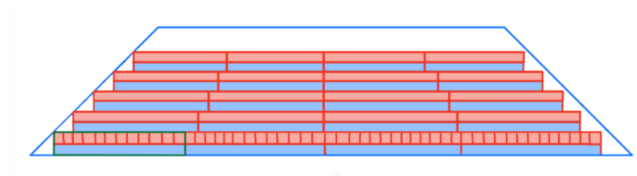
Waveform reconstruction

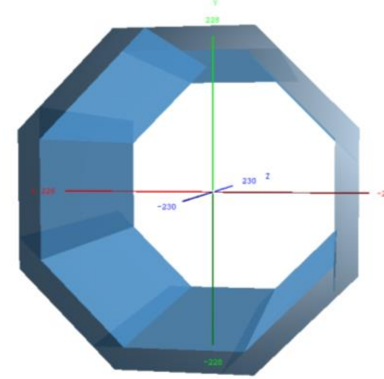
ECAL Simulation and Digitization

- Crystal ECAL
  - Homogeneous BGO crystal
  - Size: $1 \times 1 \times \sim 40\ \mathrm{cm}^3$, double-sided readout.
  - Time measurement at two ends for position along the bar.
  - Crossed arrangement in adjacent layers.
  - Full detector: $R = 1.8m, L = 4.6m, H = 28cm$, 8 same trapezoidal staves.



- ECAL geometry was implemented in CEPCSW for Geant4 simulation





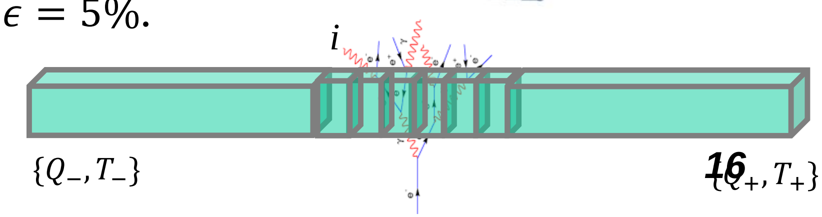
- Simple digitization for one long crystal bar
  - Contribution from G4step i :

$$Q_{\pm}^{i} = E_0 \cdot e^{-\frac{\frac{L}{2} \pm z_i}{L_{Att}}}, \qquad T_{\pm}^{i} = T_0 + Gaus(z_{\pm}^{i}/v,\ \sigma_T).$$

  - Full crystal bar:

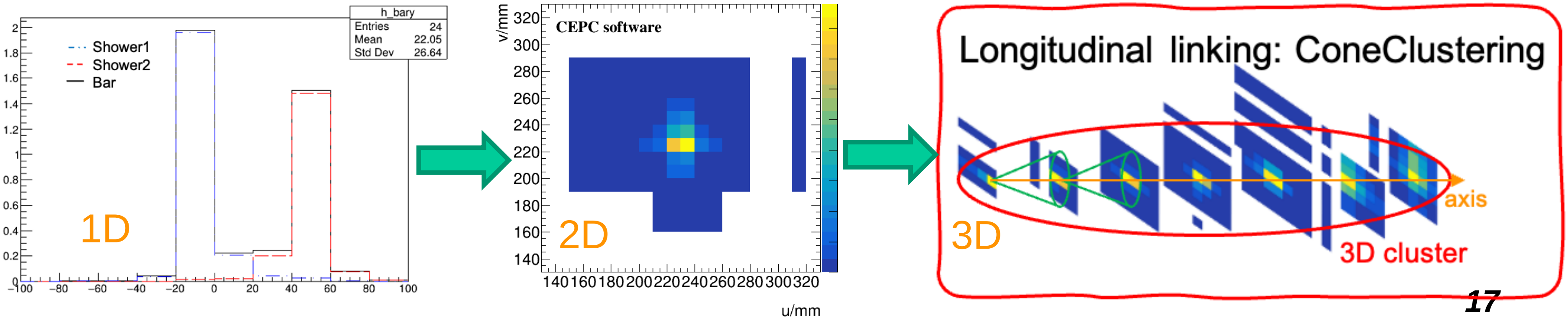
$$Q_{\pm} = \sum_{step} Q_{\pm}^{i}, \qquad T_{\pm} = T_{\pm}^{k} \mid \left(\sum_{i=1}^{k} Q_{\pm}^{i} > \epsilon Q_{\pm}^{tot} \right), \qquad \epsilon = 5\%.$$

Simplified condition: $L_{Att} = \infty$, so $Q_{\pm} = E_{tot}$.



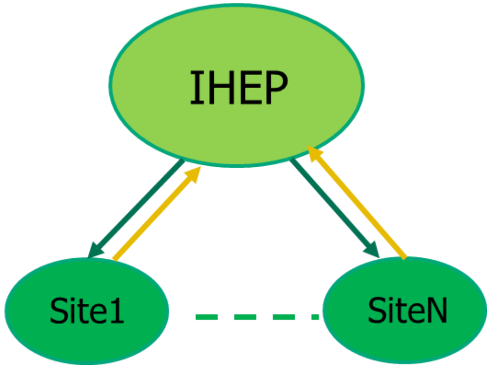
ECAL Reconstruction

- Reconstruction Iteration 0: roughly reconstruct clusters to remove hits not attributing to any cluster
- Reconstruction Iteration 1:
  - 1 dimension: clustering and energy splitting
  - 2 dimension: ghost hit removal in adjacent layers
  - 3 dimension:
    - Cone clustering longitudinally
    - Cluster ID: MIP (minimum ionizing particle) / EM / Hadronic showers
    - Merging clusters



CEPC computing: computing model

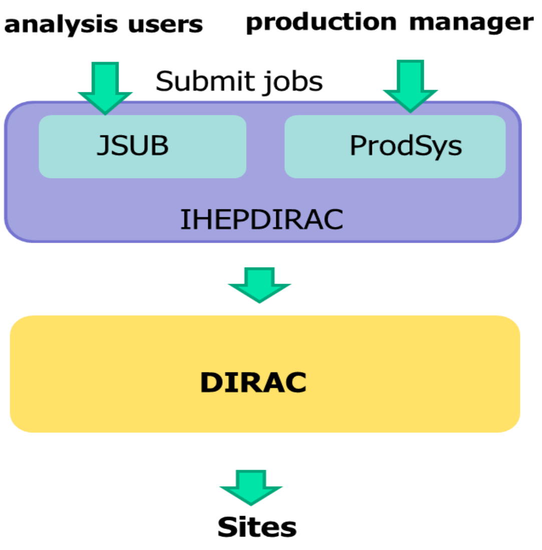
- The CEPC distributed computing system has been built using DIRAC
  - Six sites from UK and other China universities owning ~3000 CPU cores, ~3PB disk
  - 500 dedicated cores will be available at IHEP soon
  - Proved to work well with various types of computing resource including Grid, Cluster, Cloud, Commercial Cloud
- Applying a simple computing model
  - IHEP as the central site holding central storage
  - Remote sites only provide CPUs for MC production
  - Data flow
    - Input data of a job locates at IHEP
    - Output of the job will be transferred back to IHEP

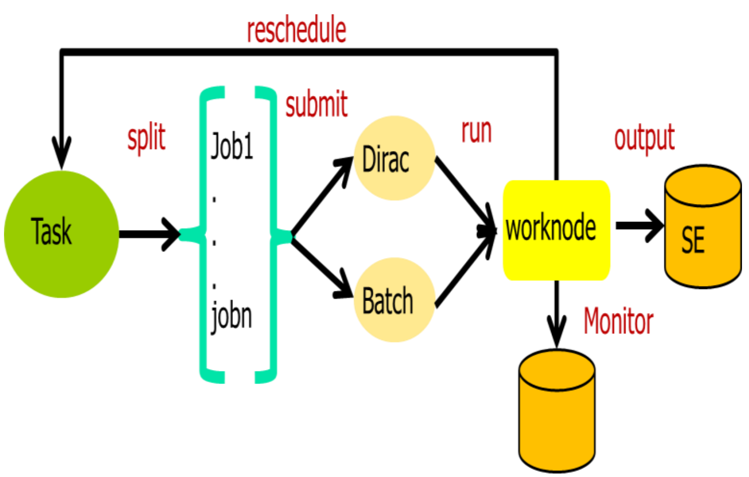


Site Name	CPU Cores
Grid.IHEP.cn	500
CLOUD.IHEPCLOUD.cn	100
GRID.QMUL.uk	1600
CLUSTER.IPAS.tw	500
CLUSTER.SJTU.cn	100
GRID.LANCASTER.uk	300
Total (Active)	**~3000**

CEPC computing: workload management

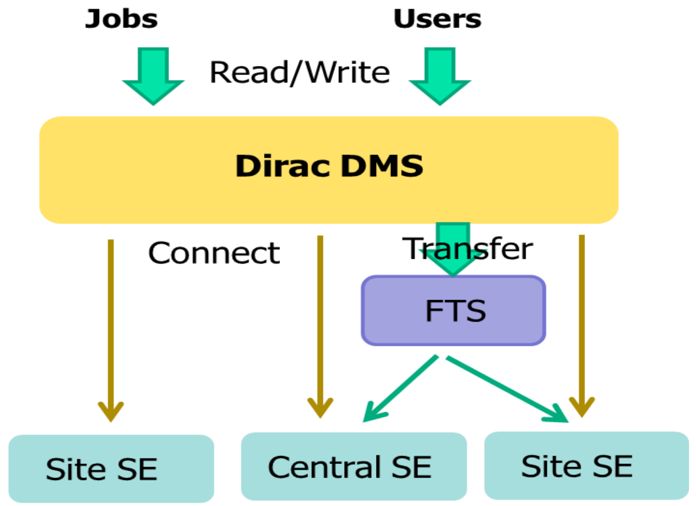
- Managing job submission and work flow
- DIRAC
  - Provide a middle layer between jobs and resources to hide complexity from users
- JSUB (developed)
  - Massive job submission frontend was developed for data analysis users
- ProdSys (being developed)
  - Be used to submit and manage production tasks for the data production group
- Both JSUB and ProdSys can take care of job lifecycles in an automatic way

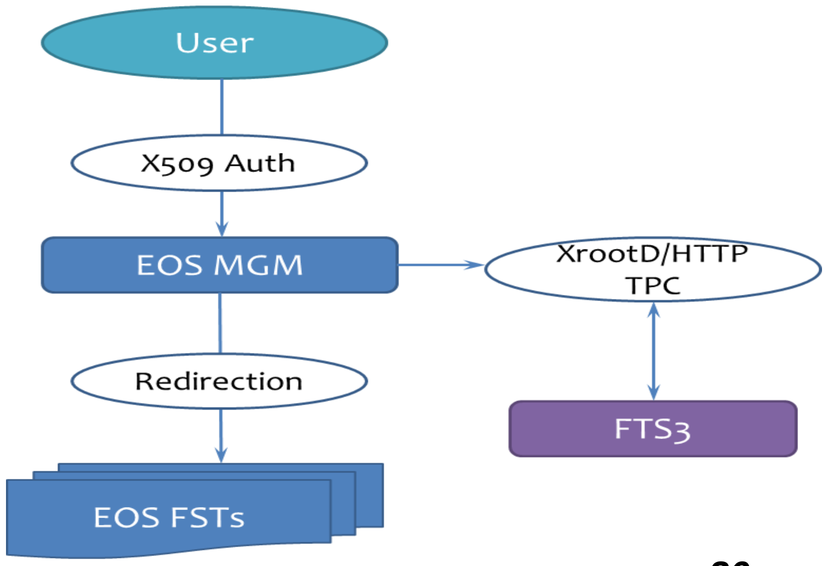




CEPC computing: data management

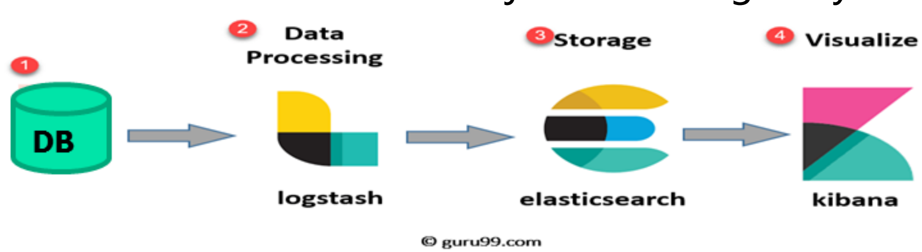
- Managing data placement and data flow globally, and providing interface for accessing data
- DIRAC Data Management System
  - File Catalogue: global view of data
  - Meta Catalogue: dataset management
- FTS (File Transfer System)
  - Manage file movements
  - fts3 server in IHEP: https://fts3.ihep.ac.cn
- Storage Element (SE)
  - Lustre as its backend now
  - EOS for newly purchased hardware

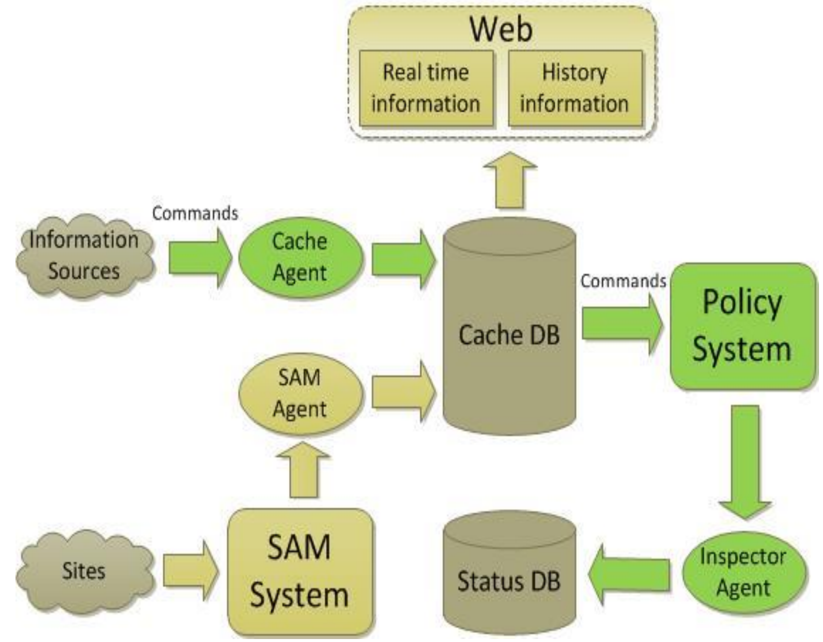




CEPC computing: monitoring

- Regular site and service status need to be checked to achieve high availability and reliability
- Monitoring dashboard was set up using Logstash + ES + Kibana
  - Give a view of sites and services status
- Site monitoring system was implemented in two ways to obtain site status
  - Active: send out standard CEPC jobs and check results periodically
  - Passive: collect user job status regularly





Plan for next year

- Core software
  - Moving towards multi-threading based on the Intel TBB (Threading Building Blocks)
  - Providing user-friendly interfaces to machine learning libraries like TensorFlow and PyTorch
  - Development of data analysis software using ROOT RDataFrame
  - Deployment of the automated validation platform to support continuous integration
- Simulation software
  - Updating geometry information according to the latest detector designs
  - Adding beam-related backgrounds
  - Providing more realistic simulation of digitization process
- Reconstruction software
  - Performance optimization of tracking algorithms in silicon and TPC trackers
  - Development of new pattern recognition algorithm for the drift chamber
  - Improving the performance of 3D cluster identification in the long crystal bar ECAL
  - Optimization of ArborPFA to improve PID performance for charged particles in the final state
- Computing
  - The data production prototype will be built to facilitate massive Monte Carlo production

Summary

- Significant progress has been made since the CEPC workshop in Shanghai last year.
- The work plan will be adjusted according to the IAC’s recommendations as well as feedbacks from the oncoming CEPC workshop.
- Both software and computing need more people’s involvements in the future development.

Thank You！

谢谢