Key4hep and its application to the CEPCSW

Tao Lin

(on behalf of CEPC core software group)

IHEP

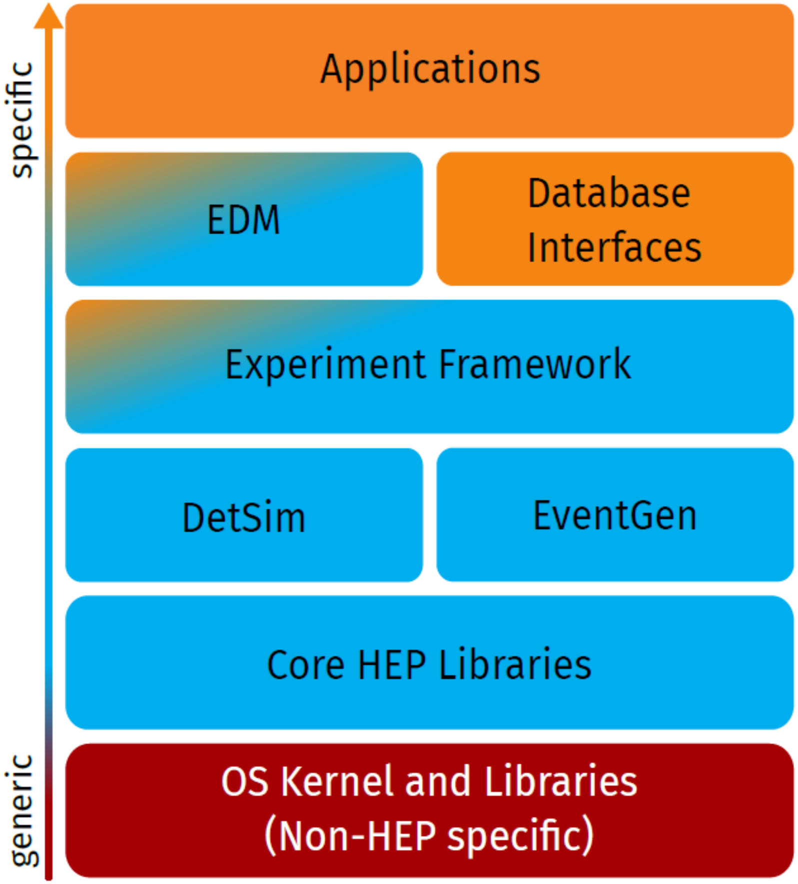
CEPC Physics and Detector Plenary Meeting

13 Jan 2021

Outline

- Overview of Key4hep project
- Status of Key4hep
- The application to CEPCSW
- Summary & Plan

HEP Software Stack



From Thomas Madlener, Epiphany Conference 2021

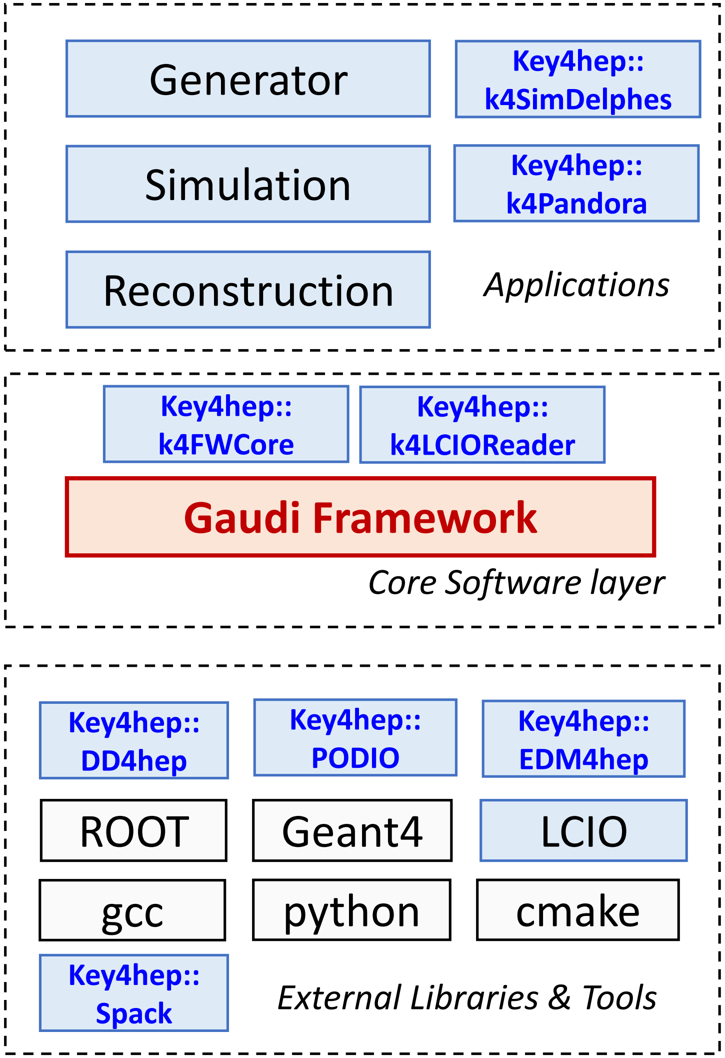
- Application layer of modules / algorithms / processors performing physics task
  - (PandoraPFA, FastJet, ACTS, ...)
- Data access and representation layer including Event Data Model
- Experiment core orchestration layer
  - (Marlin, Gaudi, CMSSW, ...)
- Specific components reused by many experiments
  - (DD4hep, Delphes, Pythia, ...)
- Commonly used HEP core libraries
  - (ROOT, Geant4, CLHEP, ...)
- Commonly used tools and libraries
  - (Python, CMake, boost, ...)

Overview of Key4hep (I)

- Motivation
  - Future detector studies rely on well maintained software to properly study possible detector concepts and their physics reach and limitations
  - Aim for a low maintenance common stack for future collider projects with ready to use “plug-ins” to develop detector concepts
- Future Collider Software Workshop (Bologna, June 2019)
  - CEPC, FCC, CLIC, ILC, SCTF
  - => Common software stack (Key4hep)
  - A turnkey system the share as many components as possible.
  - Re-use existing tools as much as possible.
  - Easy to use.
- Identified as important project in European Horizon 2020 and CERN EP R&D initiative.

Overview of Key4hep (II)

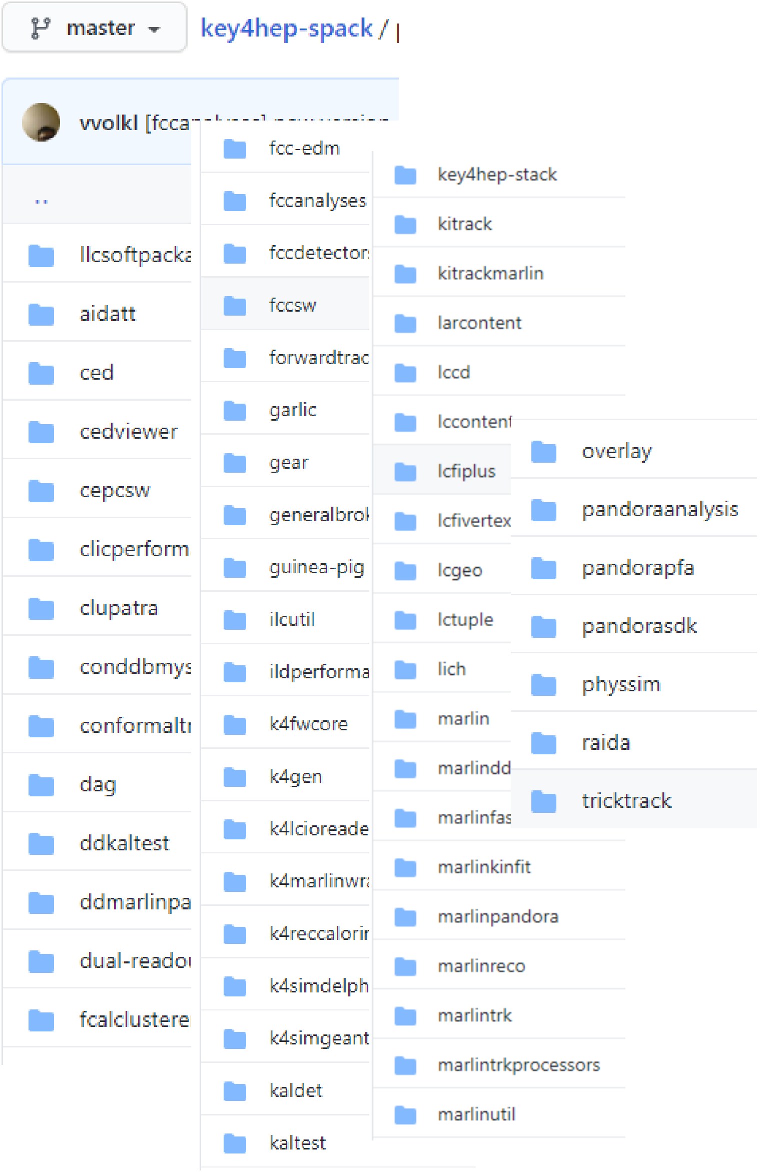
- Core components in Key4hep
  - Spack: manage the full software stack.
  - Gaudi framework: defines interfaces of all the software components and controls the event loop.
  - PODIO: toolkit to generate EDM
  - EDM4hep: generic event data model.
  - FWCore: manages the event data.
  - GeomSvc: DD4hep-based geometry management service.
- Software infrastructure
  - Documentation (Guide, Doxygen…)
  - Modern CMake
  - Automated builds and continuous integration (CI)
  - Distributed via CVMFS
  - Regular release



Key4hep-spack: Modern Software Stack Building

- Spack is a package manager
  - Originally developed by HPC community, independent of operating system.
  - Build all packages from source.
  - Handle the dependencies of all the packages.
  - Dealing with multiple configurations of the same package
    - Version, compilers, dependencies…
  - Several versions of the same package can coexist.
- Status
  - Starting in the beginning of 2020
  - Key contributions from ILC/CLIC, FCC, CEPC, …
  - Spack based installation is already deployed in CVMFS

https://Key4hep.github.io/Key4hep-doc/



master

key4hep-spack /

vvolkl [fcca

..

llcsoftpacka
aidatt
ced
cedviewer
cepcsw
clicperform
clupatra
conddbmys
conformaltı
dag
ddkaltest
ddmarlinpa
dual-readoı
fcalclustere

fcc-edm
fccanalyses
fccdetector
fccsw
forwardtrac
garlic
gear
generalbrok
guinea-pig
ilcutil
ildperforma
k4fwcore
k4gen
k4lcioreade
k4marlinwra
k4reccalorir
k4simdelph
k4simgeant
kaldet
kaltest

key4hep-stack
kitrack
kitrackmarlin
larcontent
lccd
lccontent
lcfiplus
lcfivertex
lcgeo
lctuple
lich
marlin
marlindd
marlinfas
marlinkinfit
marlinpandora
marlinreco
marlintrk
marlintrkprocessors
marlinutil

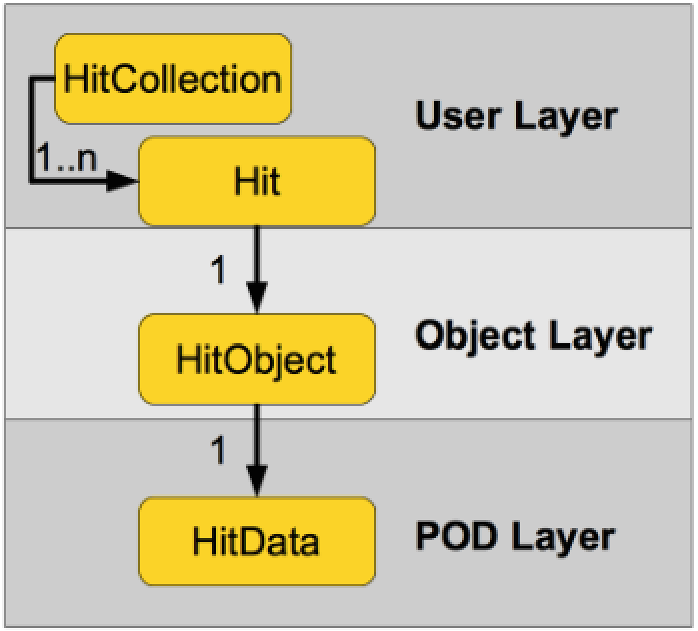
overlay
pandoraanalysis
pandorapfa
pandorasdk
physsim
raida
tricktrack

PODIO: an Event-Data Model toolkit

- Generate C++ code automatically from YAML files.
  - Support analysis in ROOT and Python.

- user layer (API):
  - handles to EDM objects (e.g. **Hit**)
  - collections of EDM object handles (e.g. **HitCollection**).
- object layer
  - transient objects (e.g. **HitObject**) handling *references* to other objects and *vector members*
- POD layer
  - the actual POD data structures holding the persistent information (e.g. **HitData**)

F. Gaede, etc. , CHEP2019

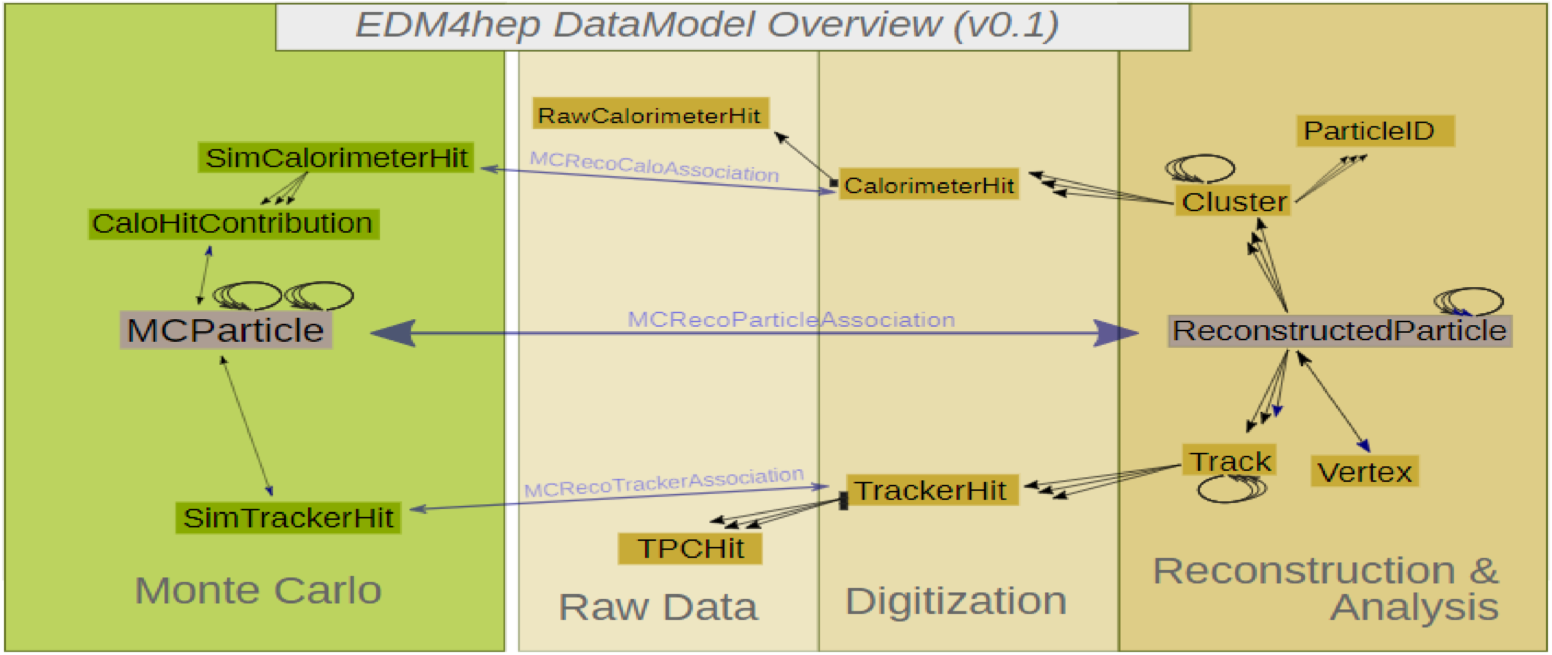


direct access to POD also possible - if needed for performance reason

https://github.com/AIDASoft/podio

EDM4hep: the official Event Data Model

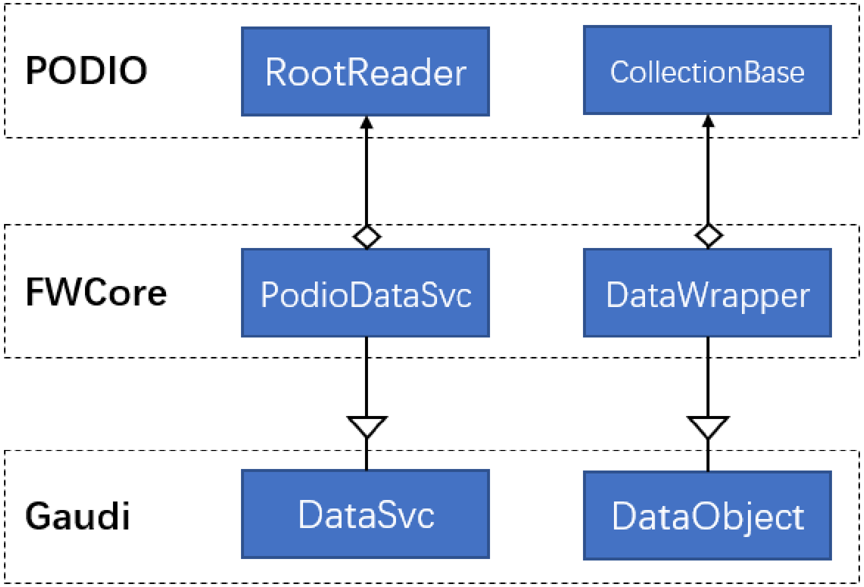
- Common EDM in Key4hep
  - The code is generated by PODIO.
  - The first version (v0.1) has been released recently



Github repository: https://github.com/Key4hep/EDM4hep

k4FWCore: integration with Gaudi

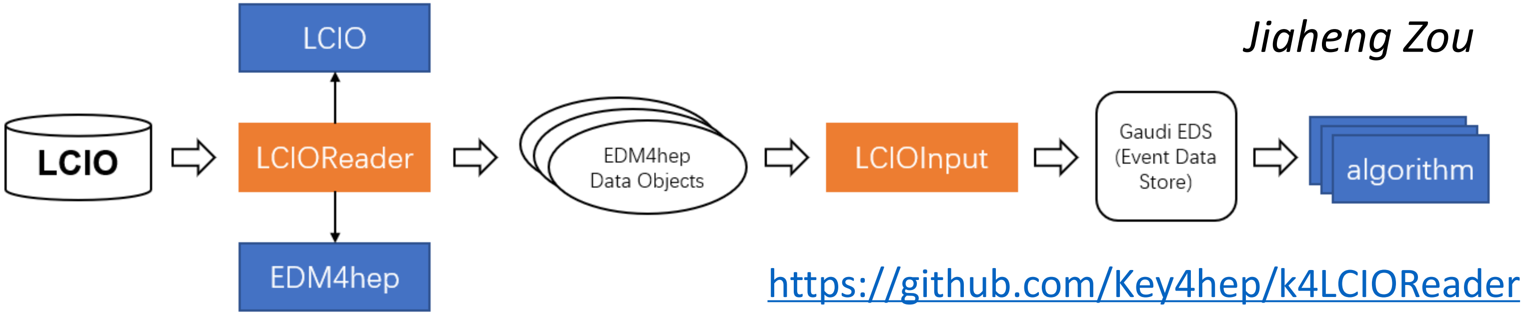
- k4FWCore is mainly used for the PODIO data handling in Gaudi
  - PodioDataSvc: a service for PODIO(in ROOT format) data I/O
  - DataWrapper: a PODIO data collection that managed in Gaudi EDS (Event Data Store)
- Status:
  - Recently switched to Gaudi v35



https://github.com/Key4hep/k4FWCore

Reading LCIO Data

- k4LCIOReader: Generate EDM4hep data collections on the fly from LCIO input files in Gaudi
  - LCIOReader: read data from LCIO format files, and convert to EDM4hep data objects in memory
  - LCIOInput: register the converted data objects in Gaudi EDS, so that other algorithms can access the data

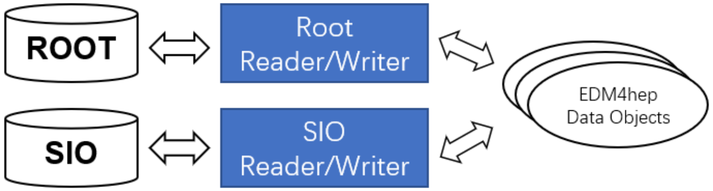


Jiaheng Zou

https://github.com/Key4hep/k4LCIOReader

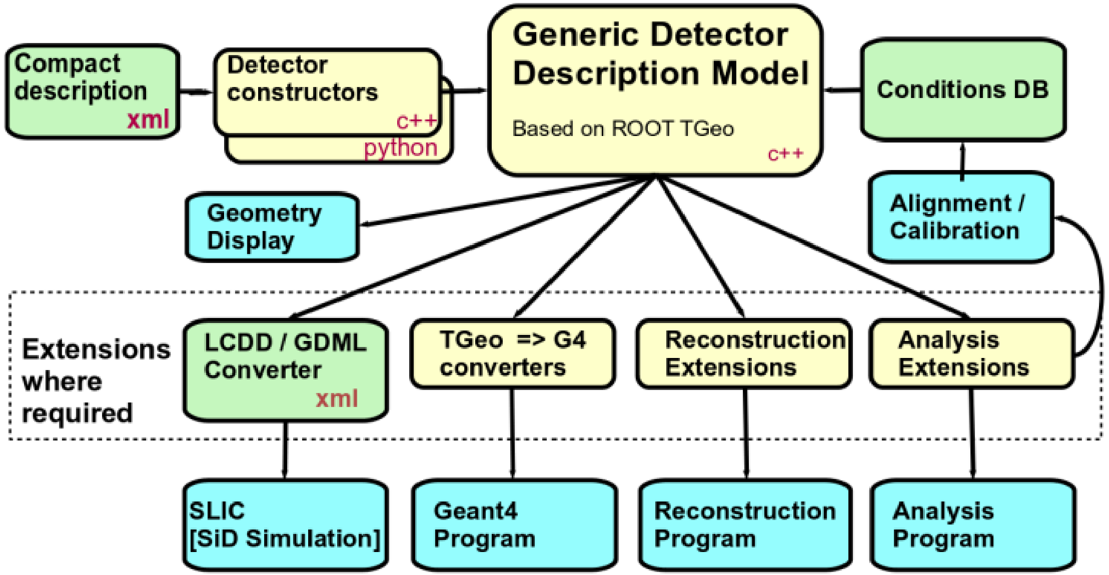
- SIO-backend in PODIO: save/read EDM4hep data objects in SIO format
  - SIO is originally a part of LCIO
  - Now a standalone project

https://github.com/iLCSoft/SIO



DD4hep: Detector Description Toolkit

- Originally developed for ILC and CLIC but with all of HEP in mind.
- A complete detector description with a single source of information
  - Geometry, materials, visualization, readout, alignment, calibration, reconstruction, …
- Covering the full life cycle of an experiment
  - Detector concepts, optimization, construction and operation



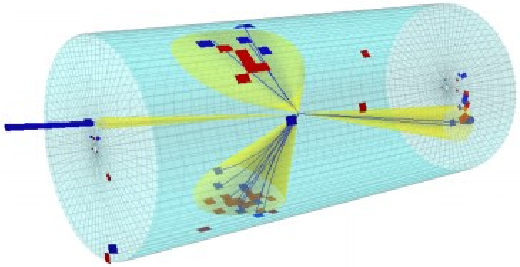
k4SimDelphes: Delphes to EDM4hep converter

Thomas Madlener & Valentin Volkl

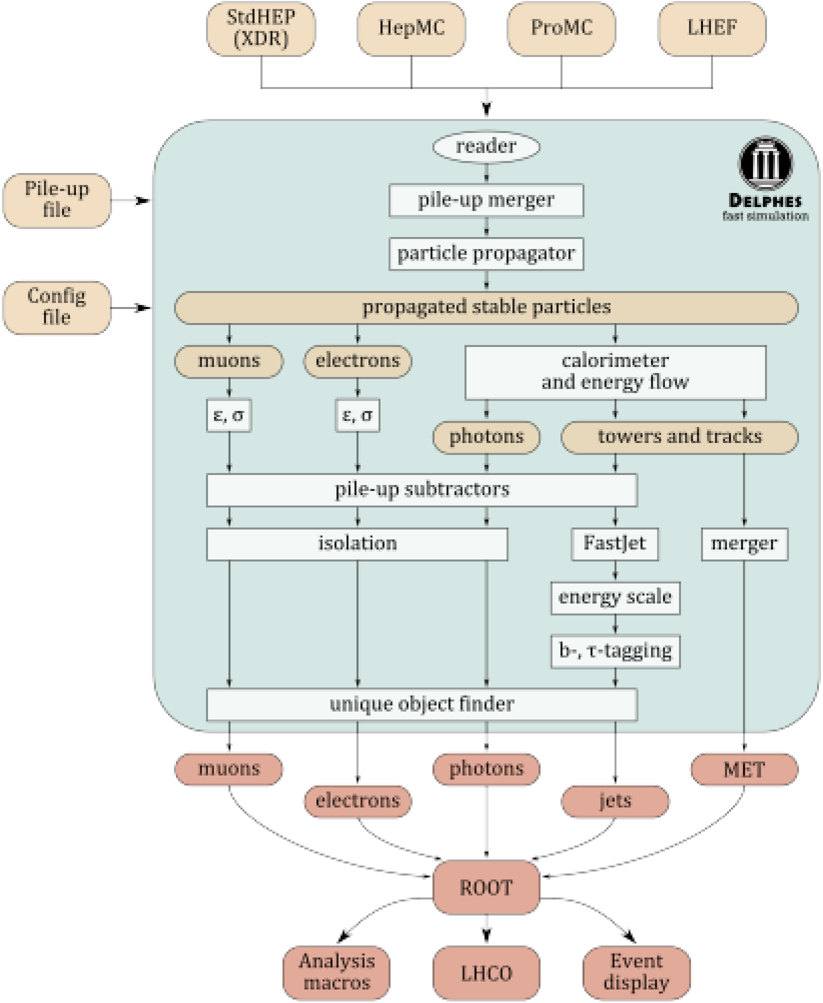
- Delphes is a fast simulation tool based on a parameterized description of the detector for phenomenological studies.

- k4SimDelphes uses Delphes to do the simulation and reconstruction and creates output files in EDM4HEP format

- Status:
  - Work on full integration is ongoing
  - It will be adopted by CEPCSW when it is ready.



Delphes event display



Workflow chart of Delphes fast simulation

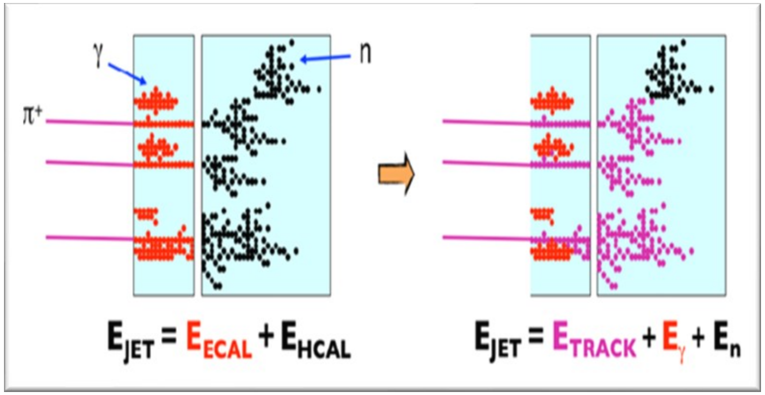
k4Pandora: Run Pandora in Gaudi

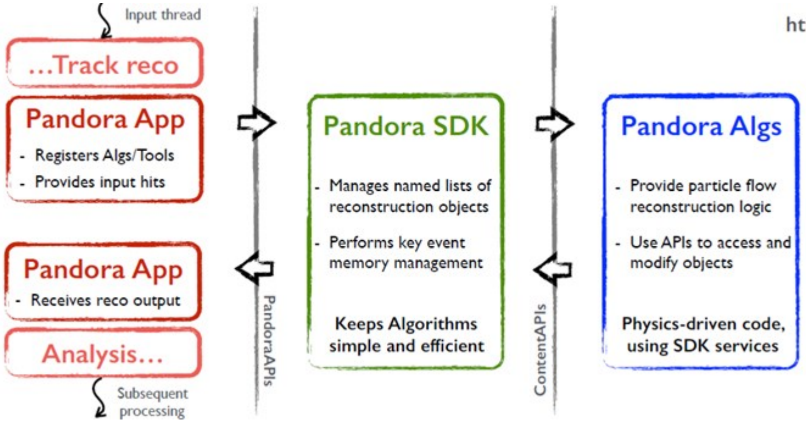
Wenxing Fang

- Using PFA algorithm to do particle reconstruction is a common choice for future collider experiments, such as ILC,FCC, and CEPC.

- Pandora is a framework designed to solve pattern recognition problem. The algorithms can be arranged flexibly.

- The k4Pandora is a Pandora App designed using EDM4HEP data as input. It supports to read detector geometry information from DD4HEP or Gear.


γ
n
π^+
$E_{JET} = E_{ECAL} + E_{HCAL}$
$E_{JET} = E_{TRACK} + E_{\gamma} + E_{n}$



Input thread

…Track reco

Pandora App
- Registers Algs/Tools
- Provides input hits

Pandora App
- Receives reco output

Analysis…

Subsequent processing

PandoraAPIs

Pandora SDK
- Manages named lists of reconstruction objects
- Performs key event memory management

Keeps Algorithms simple and efficient

ContentAPIs

Pandora Algs
- Provide particle flow reconstruction logic
- Use APIs to access and modify objects

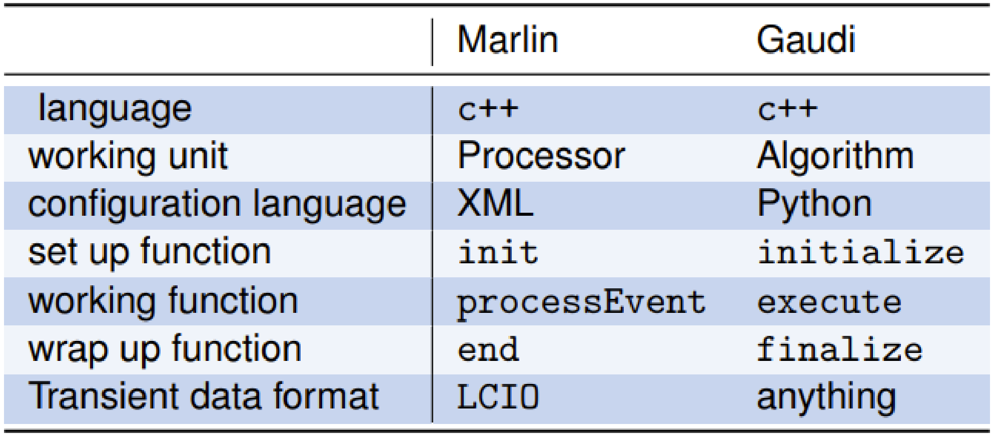
Physics-driven code, using SDK services

https://www.github.com/
PandoraPFA

k4MarlinWrapper: Run Marlin Processor in Gaudi

Andre Sailer

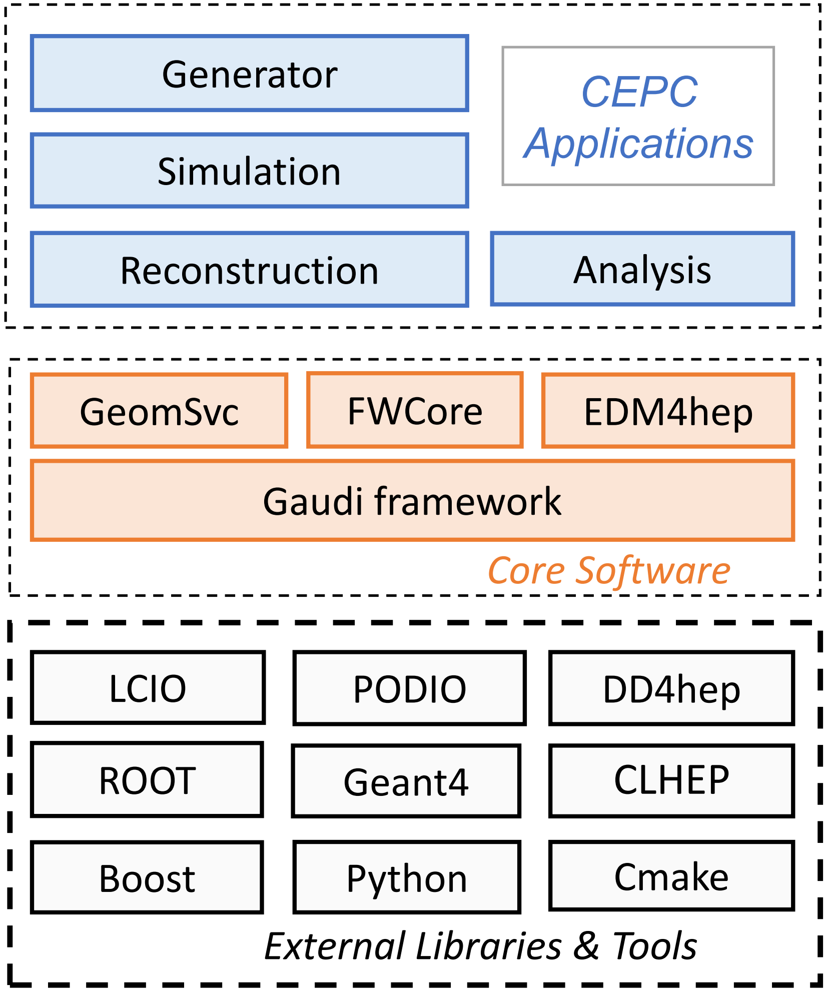
- The software of CLIC is also under migration from iLCSoft to Key4hep.
- A generic wrapper to execute the Marlin processors.
  - Use one Gaudi algorithm to run any Marlin processors.
  - Get LCIO event from Gaudi Data Store and call the marlin processors.
- Very minimal changes needed in Marlin
  - Make marlin::Processor::setParameters/setName public



	Marlin	Gaudi
language	c++	c++
working unit	Processor	Algorithm
configuration language	XML	Python
set up function	init	initialize
working function	processEvent	execute
wrap up function	end	finalize
Transient data format	LCIO	anything

CEPCSW: the first application of Key4hep

- Architecture of CEPCSW
  - external libraries
  - core software
  - CEPC applications for simulation, reconstruction and analysis.
- Core software
  - Gaudi framework: defines interfaces of all the software components and controls the event loop.
  - EDM4hep: generic event data model.
  - FWCore: manages the event data.
  - GeomSvc: DD4hep-based geometry management service.
- CEPCSW is already included in Key4hep software stack.



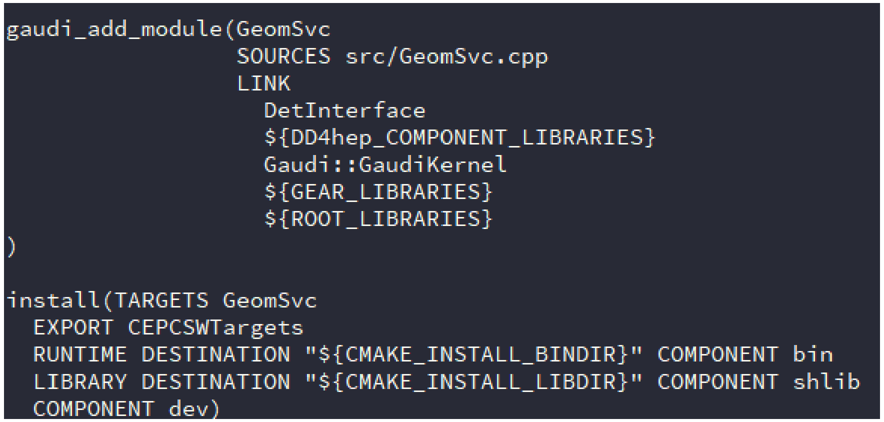
https://github.com/cepc/CEPCSW

Xingtao Huang: https://indico.ihep.ac.cn/event/11444/session/12/contribution/169/material/slides/0.pdf

CEPCSW: towards to Gaudi v35

Tao Lin

- Gaudi v35 introduces major changes including
  - Modern CMake support. A lot of complicated CMake Macros are removed from Gaudi.
  - Deprecated APIs are removed.
- Status in CEPCSW:
  - The migration of all the 45 packages from Gaudi v34 to v35 is done.
  - Optimization of the CMake.
    - The configuration time is reduced from 5 min to ~40 seconds.
    - Support Make and Ninja build tools.



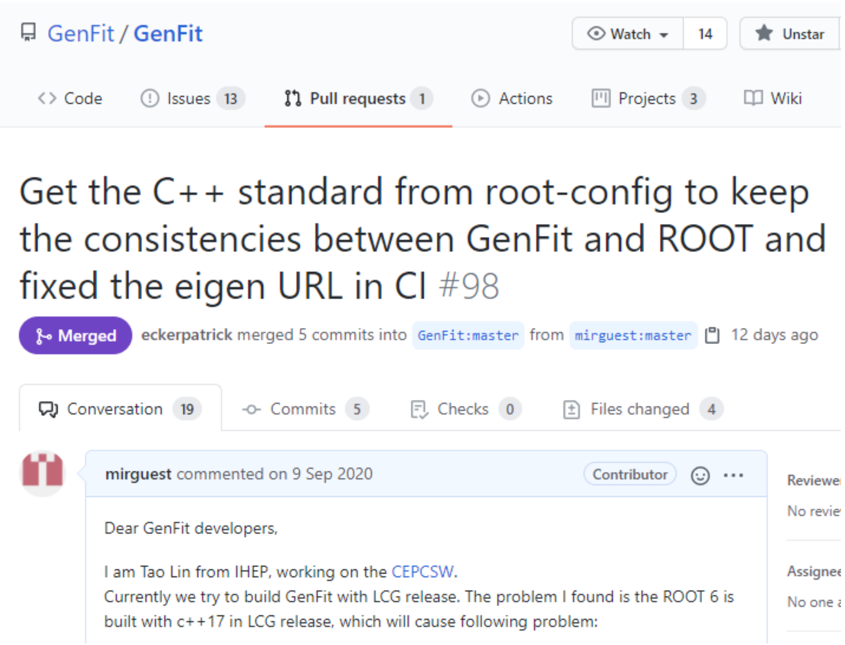
```
gaudi_add_module(GeomSvc
                 SOURCES src/GeomSvc.cpp
                 LINK
                   DetInterface
                   ${DD4hep_COMPONENT_LIBRARIES}
                   Gaudi::GaudiKernel
                   ${GEAR_LIBRARIES}
                   ${ROOT_LIBRARIES}
)

install(TARGETS GeomSvc
  EXPORT CEPCSWTargets
  RUNTIME DESTINATION "${CMAKE_INSTALL_BINDIR}" COMPONENT bin
  LIBRARY DESTINATION "${CMAKE_INSTALL_LIBDIR}" COMPONENT shlib
  COMPONENT dev)
```

- In the modern cmake, only need to declare the dependencies on target.
- The include directories and link libraries are resolved automatically.

https://github.com/cepc/CEPCSW/pull/120

Add a new external library to the Key4hep stack

- Motivation *Tao Lin*
  - Genfit is used in the reconstruction of Drift Chamber.
  - Adding a new external library which is not included in spack will break the compilation of Key4hep.
- Cooperated with Genfit (DONE), spack (DONE) and Key4hep (WIP)
  - Waiting for a new release of Genfit.



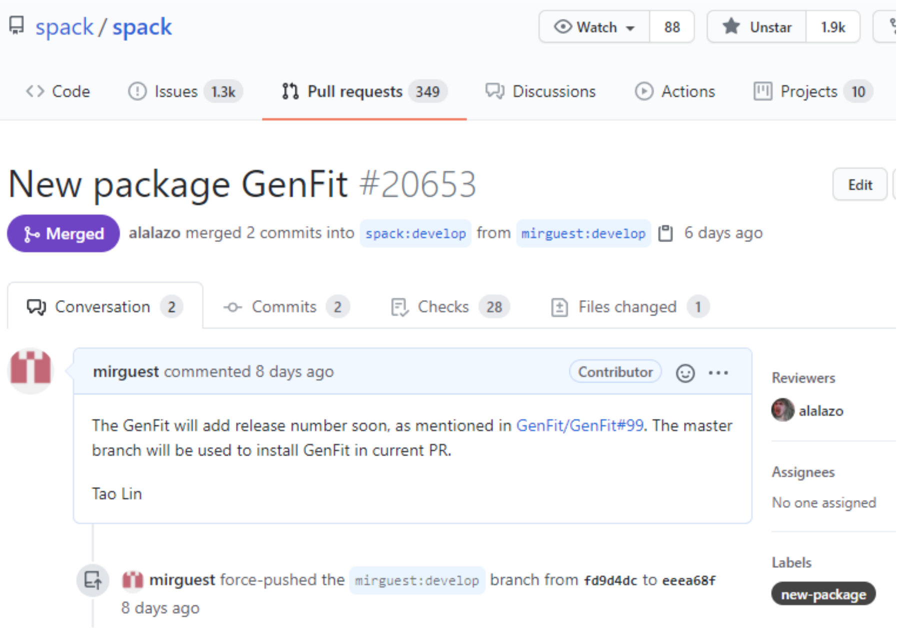
Get the C++ standard from root-config to keep the consistencies between GenFit and ROOT and fixed the eigen URL in CI #98

eckerpatrick merged 5 commits into `GenFit:master` from `mirguest:master` 12 days ago

mirguest commented on 9 Sep 2020

Dear GenFit developers,

I am Tao Lin from IHEP, working on the CEPCSW.
Currently we try to build GenFit with LCG release. The problem I found is the ROOT 6 is built with c++17 in LCG release, which will cause following problem:



New package GenFit #20653

alalazo merged 2 commits into `spack:develop` from `mirguest:develop` 6 days ago

mirguest commented 8 days ago

The GenFit will add release number soon, as mentioned in GenFit/GenFit#99. The master branch will be used to install GenFit in current PR.

Tao Lin

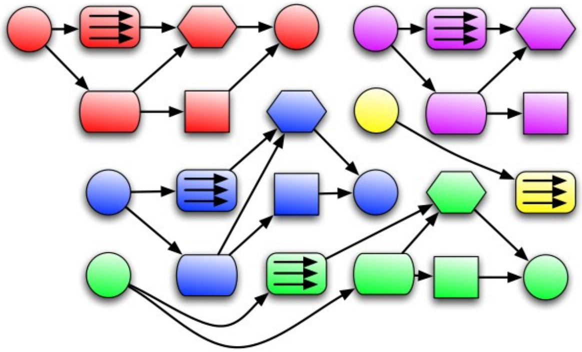
mirguest force-pushed the `mirguest:develop` branch from `fd9d4dc` to `eeea68f` 8 days ago

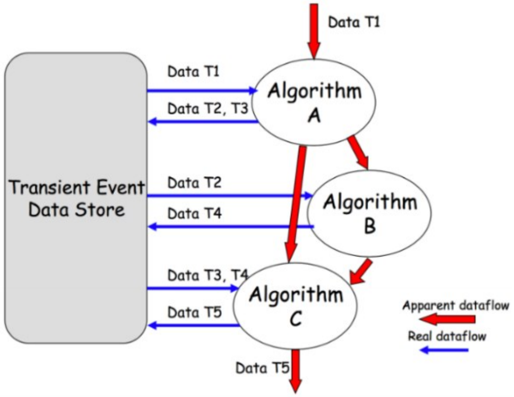
Multi-threading in CEPCSW

Wenxing Fang & Jiaheng Zou

- In order to speed up the simulation and reconstruction in CEPCSW, the multi-threading solution is under investigation.

- GaudiHive: the multi-threaded Gaudi
  - Both event level and algorithm level
- Current status
  - Integration of the Garfield++ in algorithm level parallelism has been done.
- Next to do
  - Reading and writing EDM4HEP data using GaudiHive.
  - Try multi-threading in event level for Pandora reconstruction.

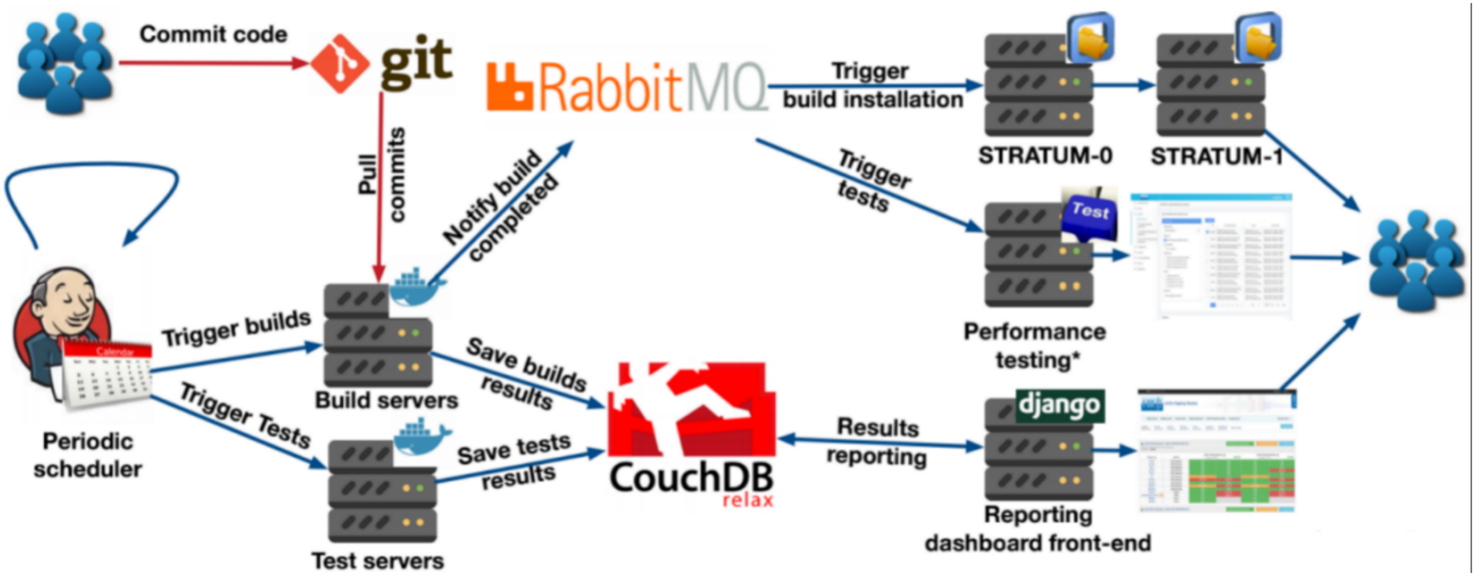




Dataflow in Gaudi

Test and Validation framework (I)

- Based on the LHCb validation system:
  - A nightly build system based on Jenkins
  - A rich set of GitLab CI tests
  - Standard CI tests for simple tests, e. g. unit tests and simple analysis
  - Customized CI tests for complicated tests, e. g. complex analysis

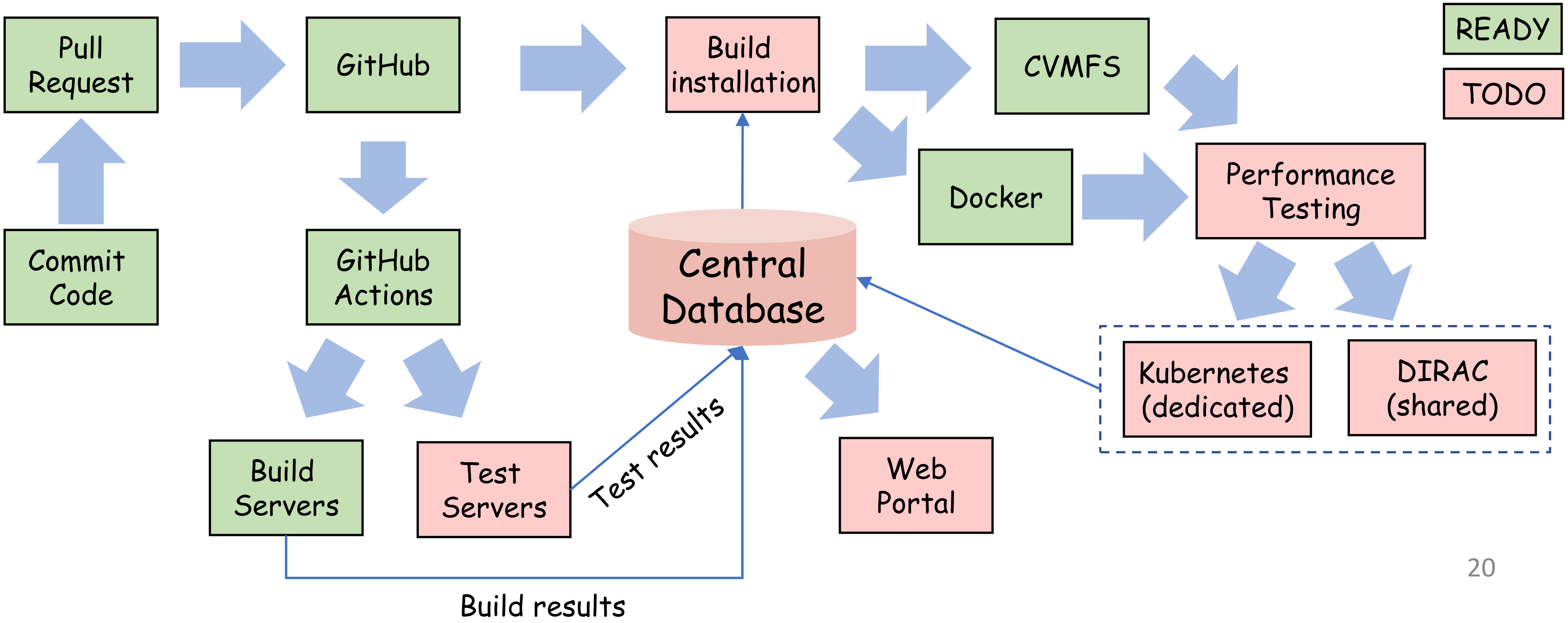


Chris Burr’s talk: https://indico.ihep.ac.cn/event/11444/session/12/contribution/173/material/slides/0.pdf

Test and Validation framework (II)

Teng Li & Tao Lin

- Validation system proposed for Key4hep
  - Based on the Github Action system
- Support wide range of features
  - Customized test runners, Nightly build and unit test, Performance test, Dashboard based on central database (CouchDB), Messaging components for long running tests



Test and Validation framework (III)

- Work in progress
  - Build self-hosted runners
    - Jenkins, Kubernetes and DIRAC
  - Build central database and messaging components
  - Customize test container image
    - Support tests on multiple platforms
    - Support CVMFS
  - Unify tools for unit test and performance test
  - Enable automatic software release
    - Auto building/registering container images
    - Auto CVMFS deployment
- Timeline
  - Prototype in 2021
  - Fully functioning in 2022

Summary & Plan

- Key4hep is the common software stack for future experiments.
    - Reusing the existing libraries and tools
    - Gaudi is one of the baseline frameworks
    - EDM4hep being developed as the common Event Data Model
- Since the meeting in Bologna in June 2019, Key4hep project becomes very active and lots of progress has been made.
- CEPCSW is fully integrated with the Key4hep by
    - Adopting EDM4hep, FWCore as well as Gaudi
    - Implementing k4LCIOReader and k4Pandora
- A short-term plan of China group
    - Multi-threading testing of EDM4hep with GaudiHive
    - An automatic testing and validation framework for Key4hep