Introduction to CEPCSW and the simulation framework

Tao Lin

lintao@ihep.ac.cn

IHEP

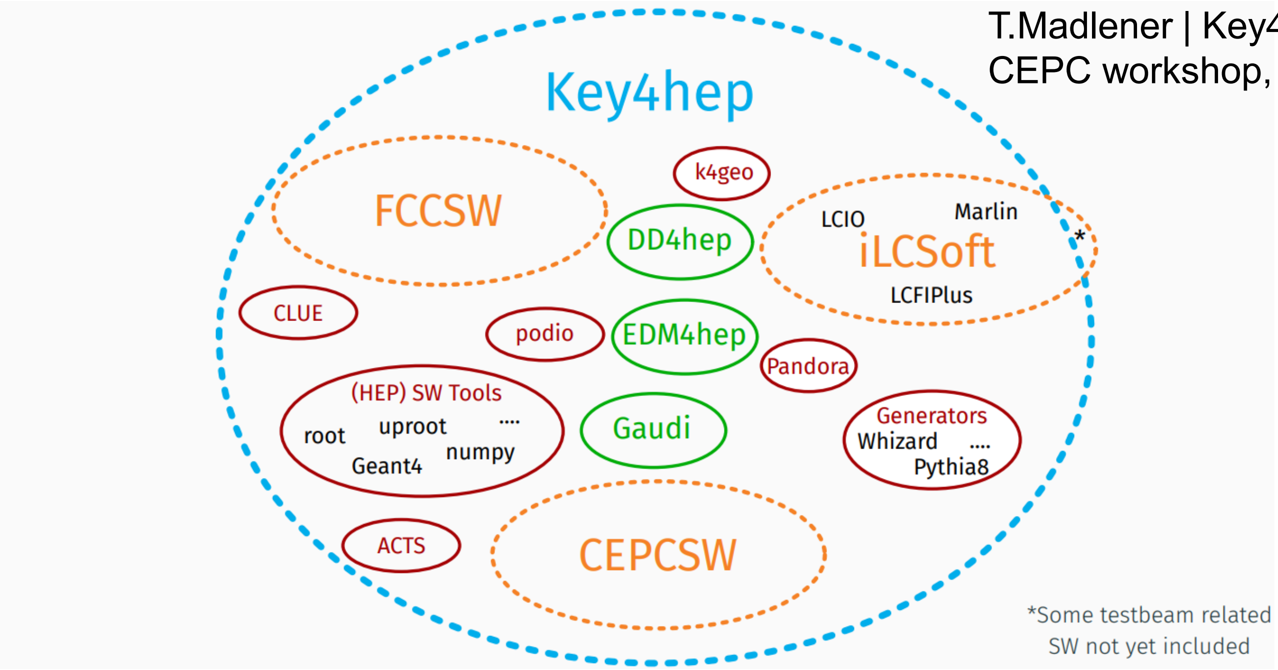
8th March 2024

Outline

- Introduction
- Core software
  - Gaudi: the underlying framework
  - EDM4hep: the event data model
  - k4FWCore: EDM management
  - DD4hep: the detector description
- Simulation framework
- Hands-on

Introduction (1)

- New CEPC software (CEPCSW) prototype was proposed at the Oxford workshop in April 2019.
- The consensus among CEPC, CLIC, FCC, ILC and other future experiments was reached at the Bologna workshop in June, 2019.
  - Develop a Common Turnkey Software Stack (Key4hep) for future collider experiments

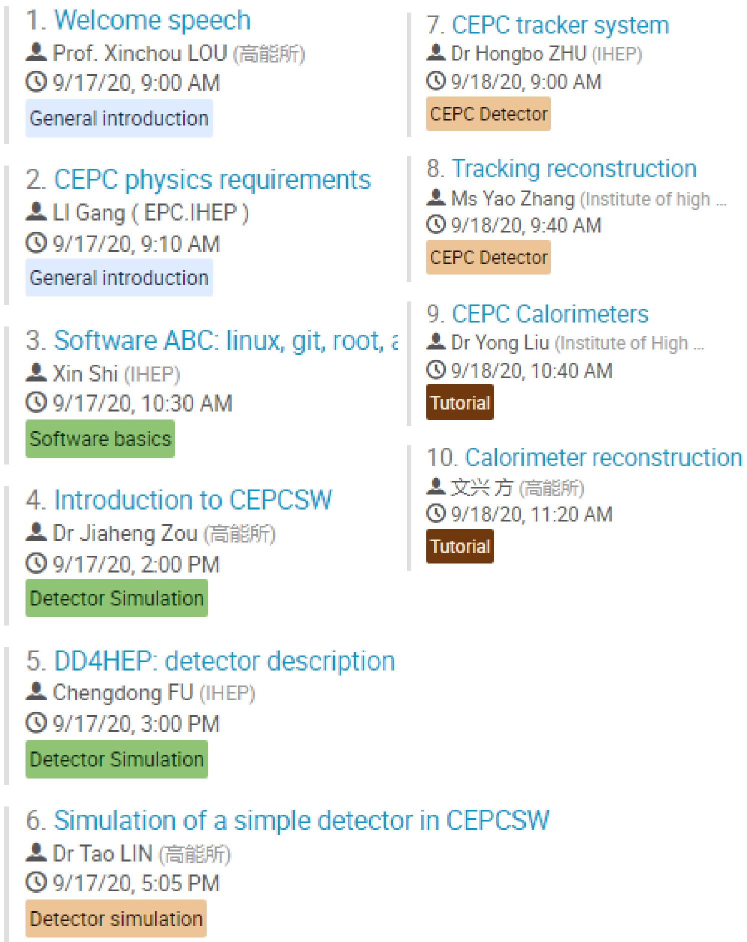


T.Madlener | Key4hep & EDM4hep
CEPC workshop, Edinburgh

*Some testbeam related
SW not yet included

Introduction (2)

- As the first application based on Key4hep, following development activities had been initiated.
  - Development of k4LCIOReader: LCIO to EDM4hep converter
  - Migration of reconstruction algorithms
  - Development of simulation framework in CEPCSW
  - Validation between iLCSoft and CEPCSW.
- CEPCSW Tutorial and detector study, IHEP, 17-18 Sept 2020
  - https://indico.ihep.ac.cn/event/12341/
- CEPCSW Training (27-29 Dec 23)
  - https://docs.qq.com/doc/DWXNkbGZDaWt
    XlM



1. Welcome speech
Prof. Xinchou LOU (高能所)
9/17/20, 9:00 AM
General introduction

2. CEPC physics requirements
LI Gang (EPC.IHEP)
9/17/20, 9:10 AM
General introduction

3. Software ABC: linux, git, root, a
Xin Shi (IHEP)
9/17/20, 10:30 AM
Software basics

4. Introduction to CEPCSW
Dr Jiaheng Zou (高能所)
9/17/20, 2:00 PM
Detector Simulation

5. DD4HEP: detector description
Chengdong FU (IHEP)
9/17/20, 3:00 PM
Detector Simulation

6. Simulation of a simple detector in CEPCSW
Dr Tao LIN (高能所)
9/17/20, 5:05 PM
Detector simulation

7. CEPC tracker system
Dr Hongbo ZHU (IHEP)
9/18/20, 9:00 AM
CEPC Detector

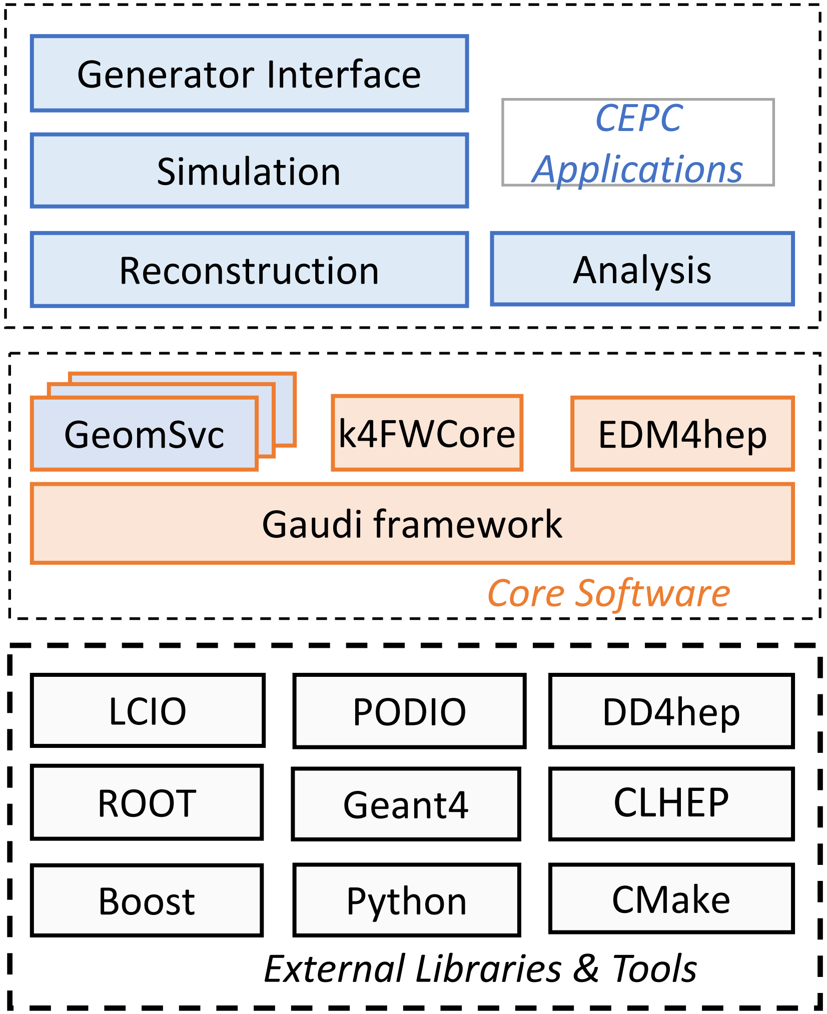
8. Tracking reconstruction
Ms Yao Zhang (Institute of high ...
9/18/20, 9:40 AM
CEPC Detector

9. CEPC Calorimeters
Dr Yong Liu (Institute of High ...
9/18/20, 10:40 AM
Tutorial

10. Calorimeter reconstruction
文兴方 (高能所)
9/18/20, 11:20 AM
Tutorial

CEPCSW

- OS/Compiler/External Libraries
  - CentOS 7 / AlmaLinux 9
  - gcc/g++ 11 (C++ 17 standard)
  - Based on CERN LCG 103 / Key4hep
  - Additional libraries are deployed at IHEP
    - /cvmfs/cepcsw.ihep.ac.cn/prototype/releases/externals/103.0.2
- Core software
  - Gaudi: underlying framework
  - EDM4hep: the event data model
  - k4FWCore: data management
  - DD4hep: detector description
- CEPC Applications



https://github.com/cepc/CEPCSW
https://code.ihep.ac.cn/cepc/CEPCSW

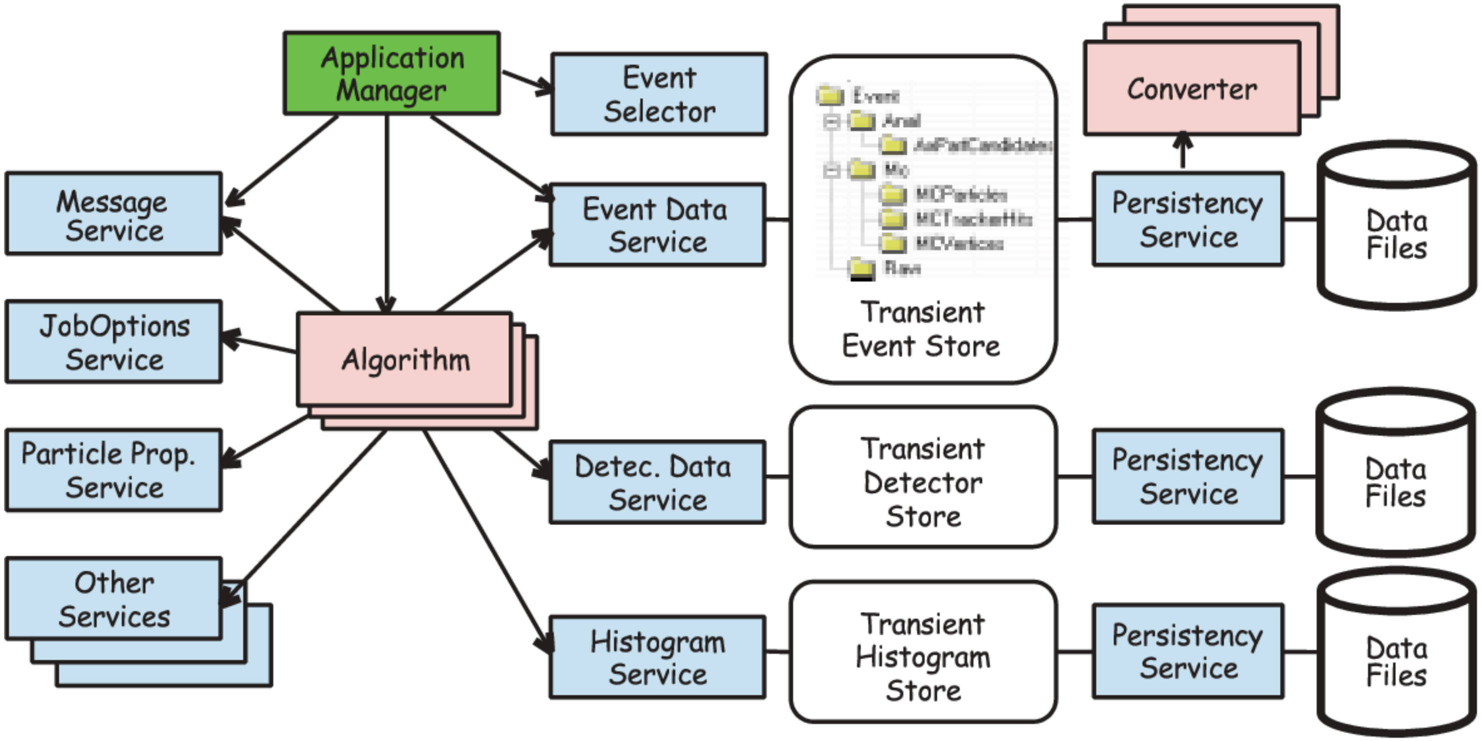
Packages in CEPCSW

- Detector concepts
  - CDR (baseline design)
  - The 4th concept
- MC Generators
  - Multiple formats for signals: HepMC, HepEvt, StdHep, LCIO
  - GuineaPig++ for MDI
  - Particle Gun
- Simulation
  - G4 based simulation framework
  - Fast simulation models, such as ML based dE/dx simulation
  - Digitization algorithms for silicon, calo, drift chamber
- Reconstruction
  - Marlin based tracking algorithms for silicon detector
  - Genfit based tracking algorithms for drift chamber
  - Pandora based PFA
- Analysis tools
  - Arbor based analysis algorithm.
  - RDataFrame based analysis framework.
- Examples and docs
  - Usage of EDM4hep, Identifier, etc.

50 packages in total

Gaudi framework

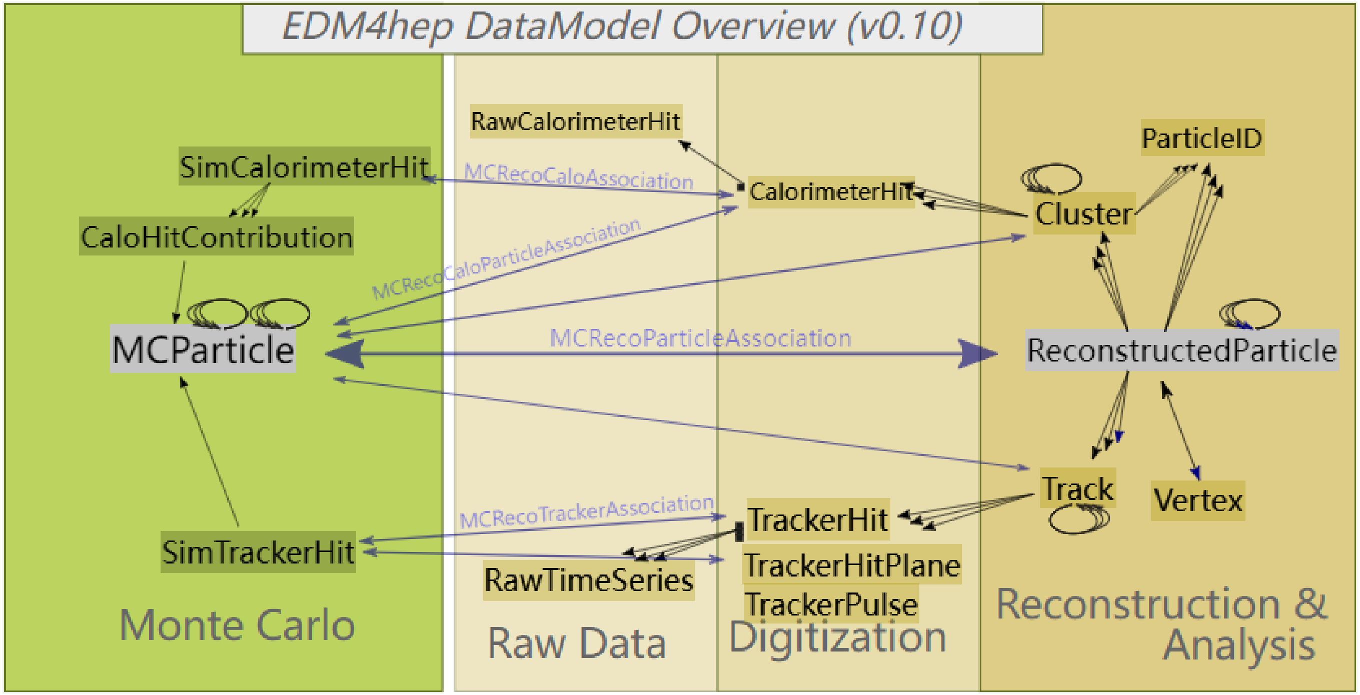
- Key components
  - Algorithm: concrete calculation to the event during event loop
  - Service: Common functionalities that can be invoked by other components



https://gitlab.cern.ch/gaudi/Gaudi

EDM4hep: Event Data Model

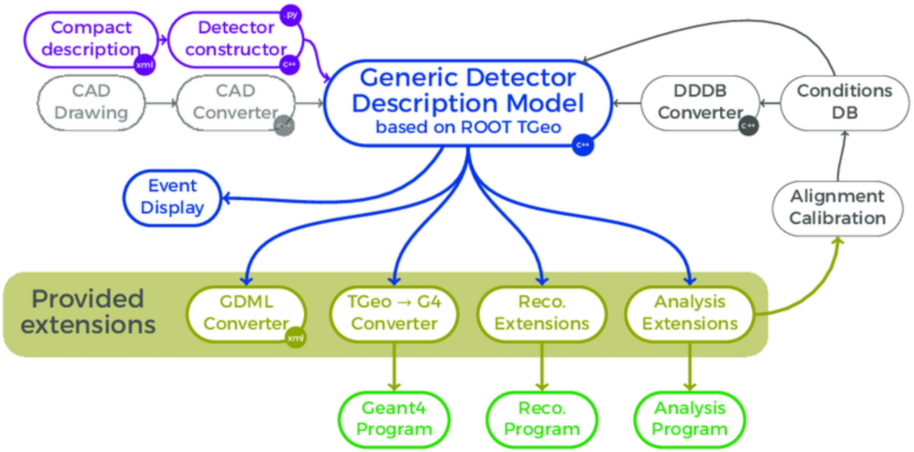
- EDM4hep is the common event data model (EDM) being developed for the future experiments like CEPC, CLIC, FCC, ILC, etc.
  - describing event objects created at different data processing stages and also reflecting the relationship between them.
- The code is generated by toolkit PODIO from a yaml file.

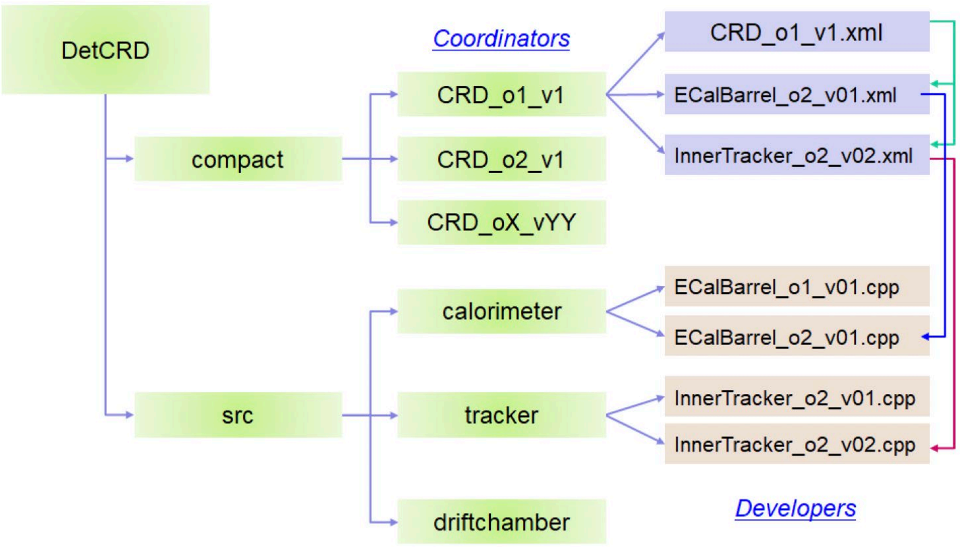


https://github.com/key4hep/EDM4hep

DD4hep: Detector Description

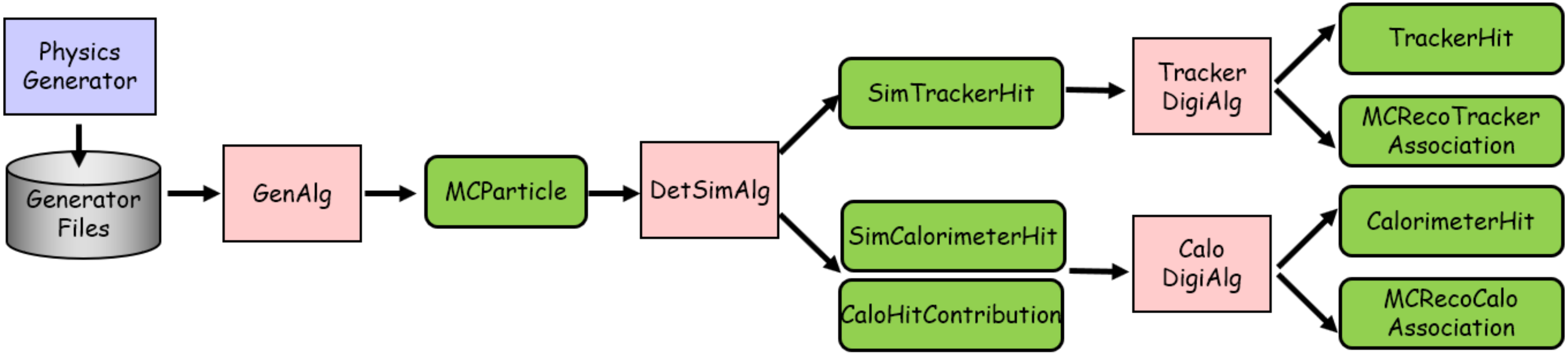
- DD4hep was adopted to provide a full detector description, which was generated from a single source
- Different detector design options are managed in the Git repository and a simulation job can be easily configured in runtime
- The non-uniform magnetic field was also implemented in CEPCSW





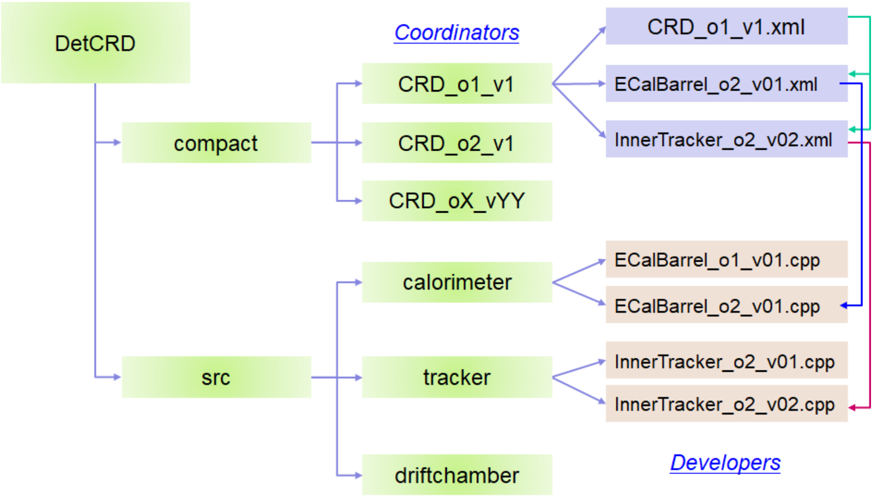
Simulation framework in CEPCSW (1)

- Complete simulation chain with EDM4hep
  - Physics generator
    - MCParticle
  - Detector Simulation based on Geant4
    - MCParticle (with secondaries), SimTrackerHit, SimCalorimeterHits
  - Digitization
    - TrackerHit, CalorimeterHit

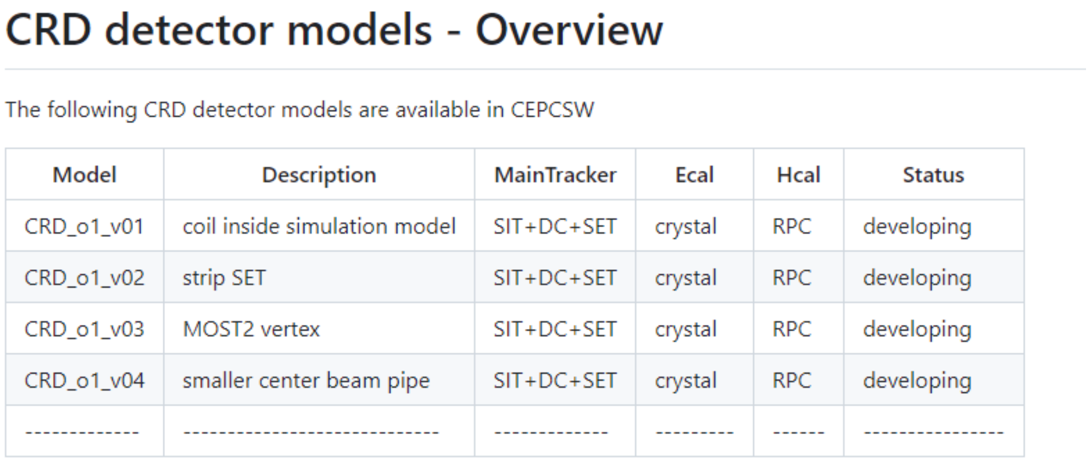


Simulation framework in CEPCSW (2)

- Geometry management with DD4hep
  - Consists of C++ constructors and XML based compact files
  - https://github.com/cepc/CEPCSW/tree/master/Detector



For the CRD detector models, see README by Chengdong:
https://github.com/cepc/CEPCSW/tree/master/Detector/DetCRD/compact



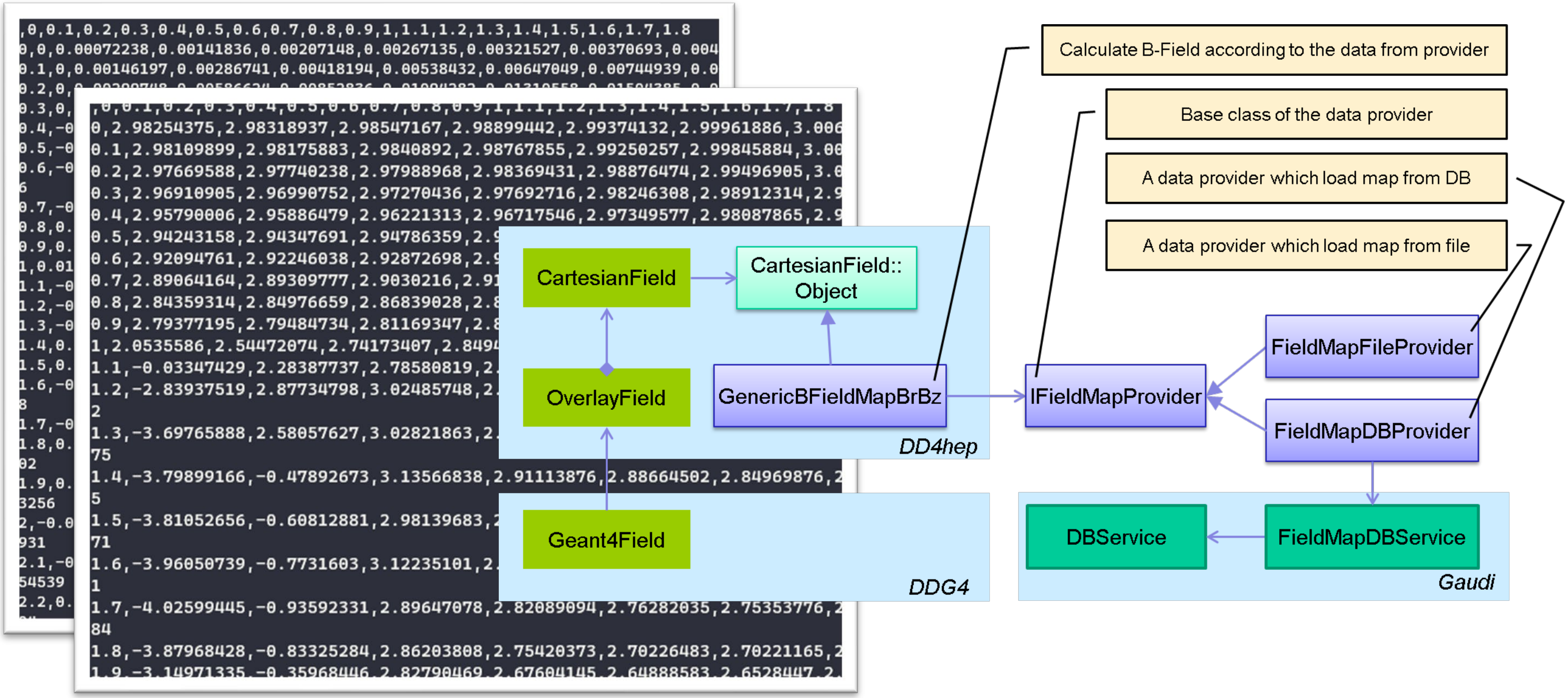
CRD detector models - Overview

The following CRD detector models are available in CEPCSW

Model	Description	MainTracker	Ecal	Hcal	Status
CRD_o1_v01	coil inside simulation model	SIT+DC+SET	crystal	RPC	developing
CRD_o1_v02	strip SET	SIT+DC+SET	crystal	RPC	developing
CRD_o1_v03	MOST2 vertex	SIT+DC+SET	crystal	RPC	developing
CRD_o1_v04	smaller center beam pipe	SIT+DC+SET	crystal	RPC	developing
-------------	-----------------------------	-------------	---------	------	----------------

Simulation framework in CEPCSW (3)

- Non-uniform magnetic fields
  - The Br/Bz csv files are provided by magnetic group.



```
,0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1,1.1,1.2,1.3,1.4,1.5,1.6,1.7,1.8
0,0,0.00072238,0.00141836,0.00207148,0.00267135,0.00321527,0.00370693,0.004
0.1,0,0.00146197,0.00286741,0.00418194,0.00538432,0.00647049,0.00744939,0.0
0.2,0,
0.3,0,
0.4,-0
0.5,-0
0.6,-0
6
0.7,-0
0.8,0.
0.9,0.
1,0.01
1.1,-0
1.2,-0
1.3,-0
1.4,0.
1.5,0.
1.6,-0
8
1.7,-0
1.8,0.
02
1.9,0.
3256
2,-0.0
931
2.1,-0
54539
2.2,0.
```

```
,0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1,1.1,1.2,1.3,1.4,1.5,1.6,1.7,1.8
0,2.98254375,2.98318937,2.98547167,2.98899442,2.99374132,2.99961886,3.006
0.1,2.98109899,2.98175883,2.9840892,2.98767855,2.99250257,2.99845884,3.00
0.2,2.97669588,2.97740238,2.97988968,2.98369431,2.98876474,2.99496905,3.0
0.3,2.96910905,2.96990752,2.97270436,2.97692716,2.98246308,2.98912314,2.9
0.4,2.95790006,2.95886479,2.96221313,2.96717546,2.97349577,2.98087865,2.9
0.5,2.94243158,2.94347691,2.94786359,2.9
0.6,2.92094761,2.92246038,2.92872698,2.9
0.7,2.89064164,2.89309777,2.9030216,2.91
0.8,2.84359314,2.84976659,2.86839028,2.8
0.9,2.79377195,2.79484734,2.81169347,2.8
1,2.0535586,2.54472074,2.74173407,2.849
1.1,-0.03347429,2.28387737,2.78580819,2.
1.2,-2.83937519,2.87734798,3.02485748,2.
2
1.3,-3.69765888,2.58057627,3.02821863,2.
75
1.4,-3.79899166,-0.47892673,3.13566838,2.91113876,2.88664502,2.84969876,2
5
1.5,-3.81052656,-0.60812881,2.98139683,2
71
1.6,-3.96050739,-0.7731603,3.12235101,2.
1
1.7,-4.02599445,-0.93592331,2.89647078,2.82089094,2.76282035,2.75353776,2
84
1.8,-3.87968428,-0.83325284,2.86203808,2.75420373,2.70226483,2.70221165,2
1.9,-3.14971335,-0.35968446,2.82790469,2.67604145,2.64888583,2.6528447,2.
```

Calculate B-Field according to the data from provider

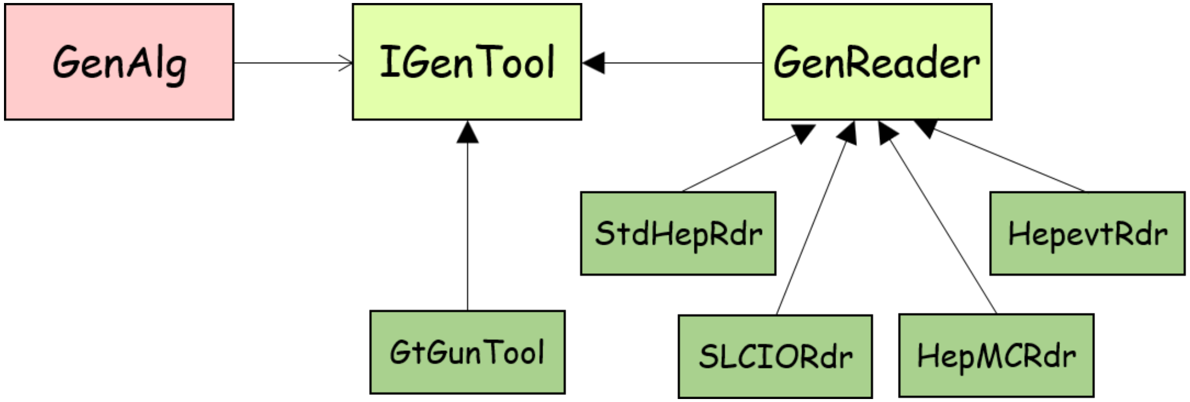
Base class of the data provider

A data provider which load map from DB

A data provider which load map from file

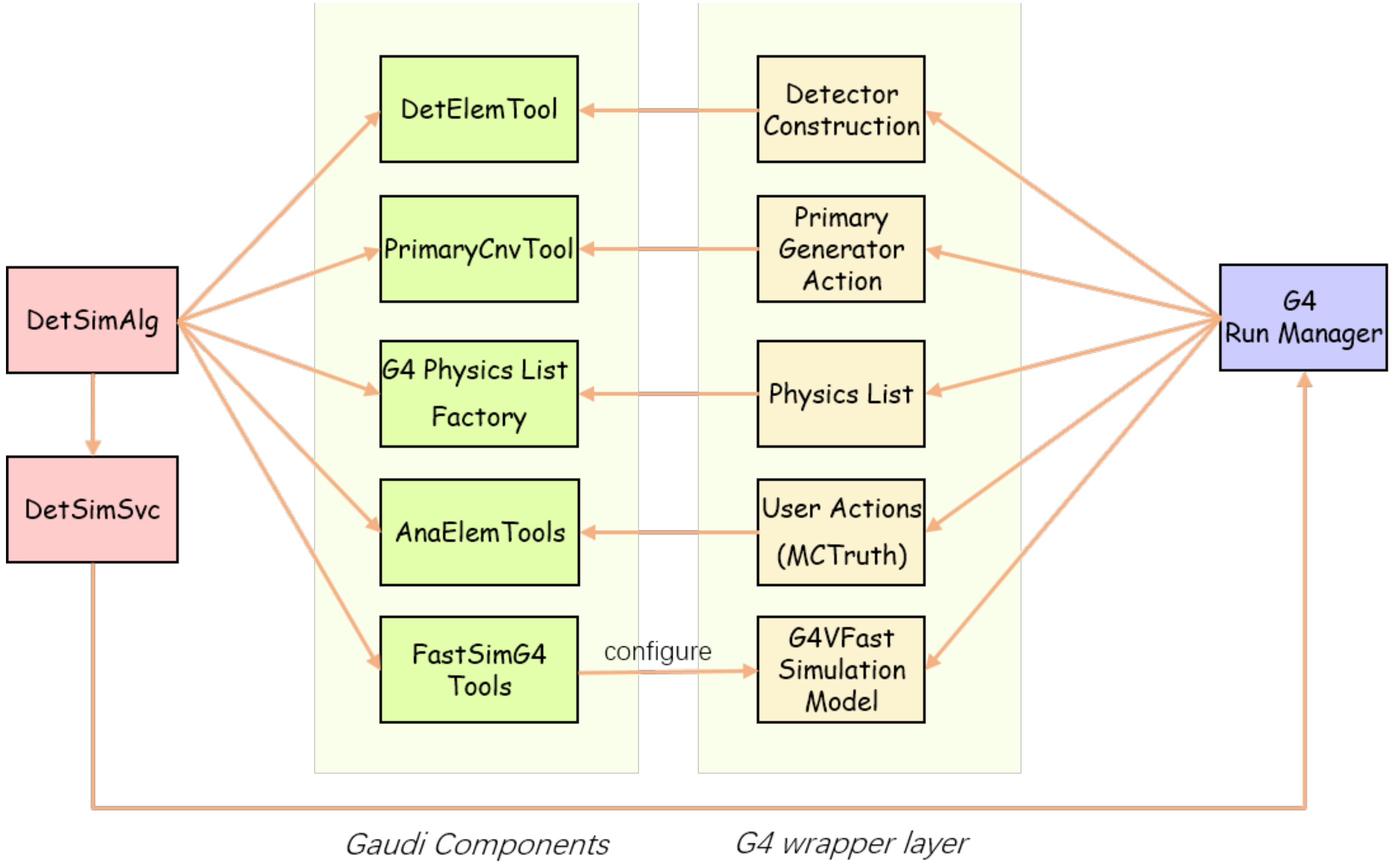
Simulation framework in CEPCSW (4)

- Physics generator interface
  - Physics generators with different formats are integrated, including StdHep, HepEvt, LCIO, HepMC formats.
  - Particle gun is supported.
  - Beam background generators, such as Guinea Pig.



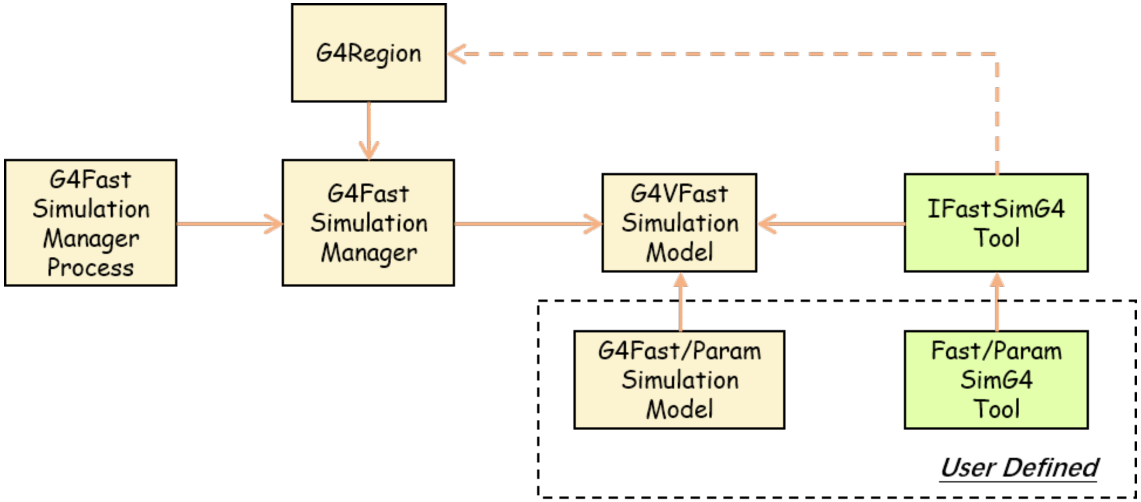
Simulation framework in CEPCSW (5)

- Integration with Geant4 and Gaudi
  - A thin layer is developed to manage corresponding Geant4 objects.

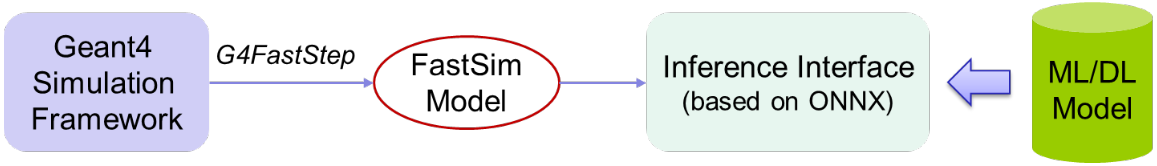


Simulation framework in CEPCSW (6)

- Integration with Fast Simulation
  - Region based: when a particle enter a region, fast simulation will be triggered by Geant4.


G4Region
G4Fast Simulation Manager Process
G4Fast Simulation Manager
G4VFast Simulation Model
IFastSimG4 Tool
G4Fast/Param Simulation Model
Fast/Param SimG4 Tool
User Defined

  - Support ML methods via ONNX inference interface.
    - Example: Fast pulse simulation (MLP) in drift chamber done by Wenxing

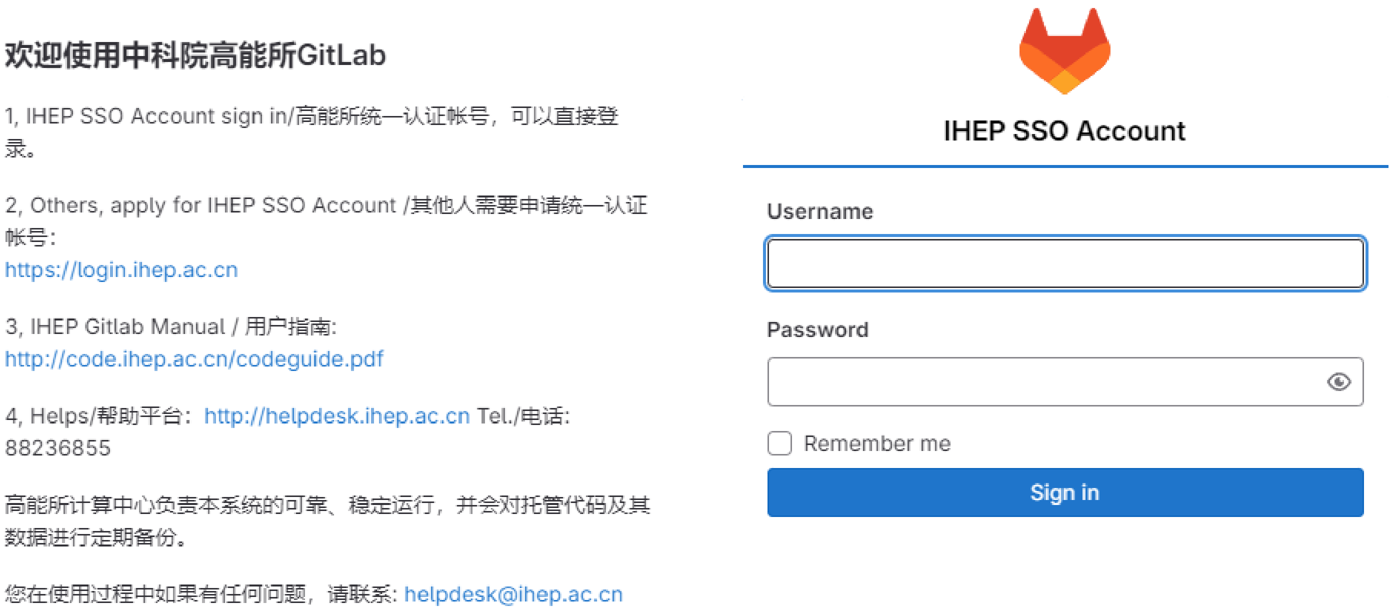

Geant4 Simulation Framework
G4FastStep
FastSim Model
Inference Interface (based on ONNX)
ML/DL Model

Hands-on

- Use GitLab at IHEP: https://code.ihep.ac.cn/cepc/CEPCSW
  - Login with your IHEP SSO account.
  - Upload your SSH public key into GitLab.
  - Fork the CEPCSW to your account.
  - Clone the source code.
- Build and test your own CEPCSW
  - Setup software environment
  - Build with CMake
  - Test the job

Login with IHEP SSO account

- https://code.ihep.ac.cn/users/sign_in



欢迎使用中科院高能所GitLab

1, IHEP SSO Account sign in/高能所统一认证帐号，可以直接登录。

2, Others, apply for IHEP SSO Account /其他人需要申请统一认证帐号：
https://login.ihep.ac.cn

3, IHEP Gitlab Manual / 用户指南:
http://code.ihep.ac.cn/codeguide.pdf

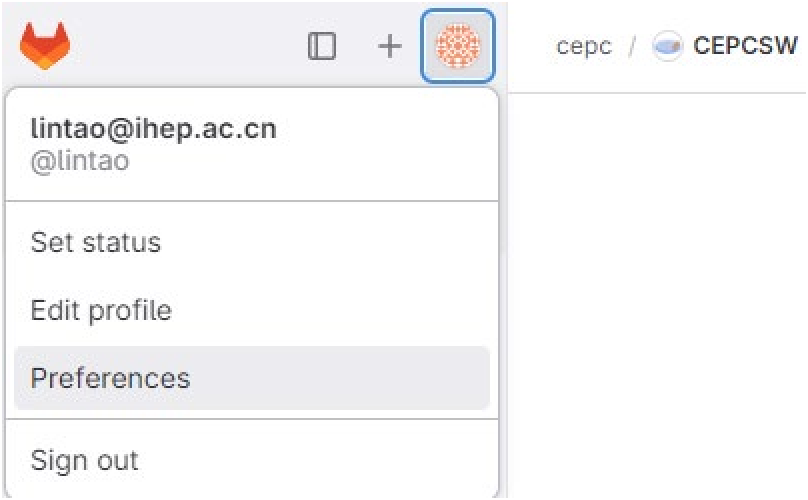
4, Helps/帮助平台：http://helpdesk.ihep.ac.cn Tel./电话: 88236855

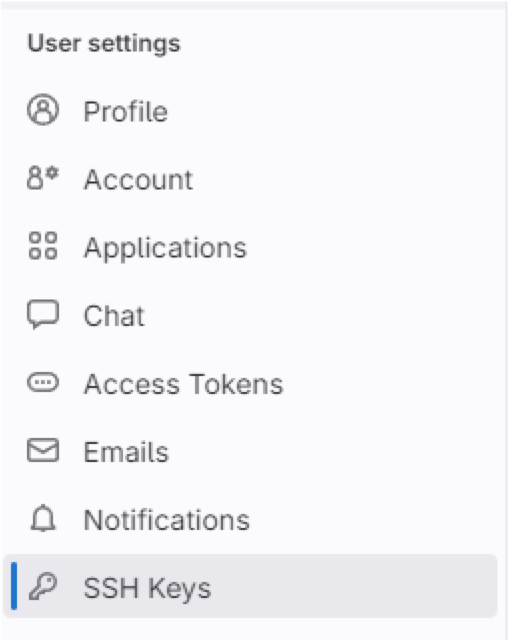
高能所计算中心负责本系统的可靠、稳定运行，并会对托管代码及其数据进行定期备份。

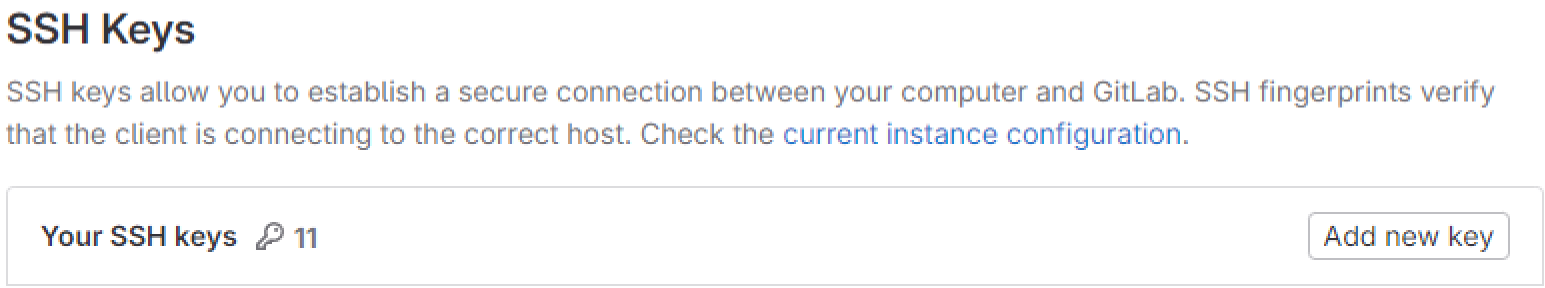
您在使用过程中如果有任何问题，请联系：helpdesk@ihep.ac.cn

Upload SSH public key into GitLab.

- SSH key could be generated using ssh-keygen





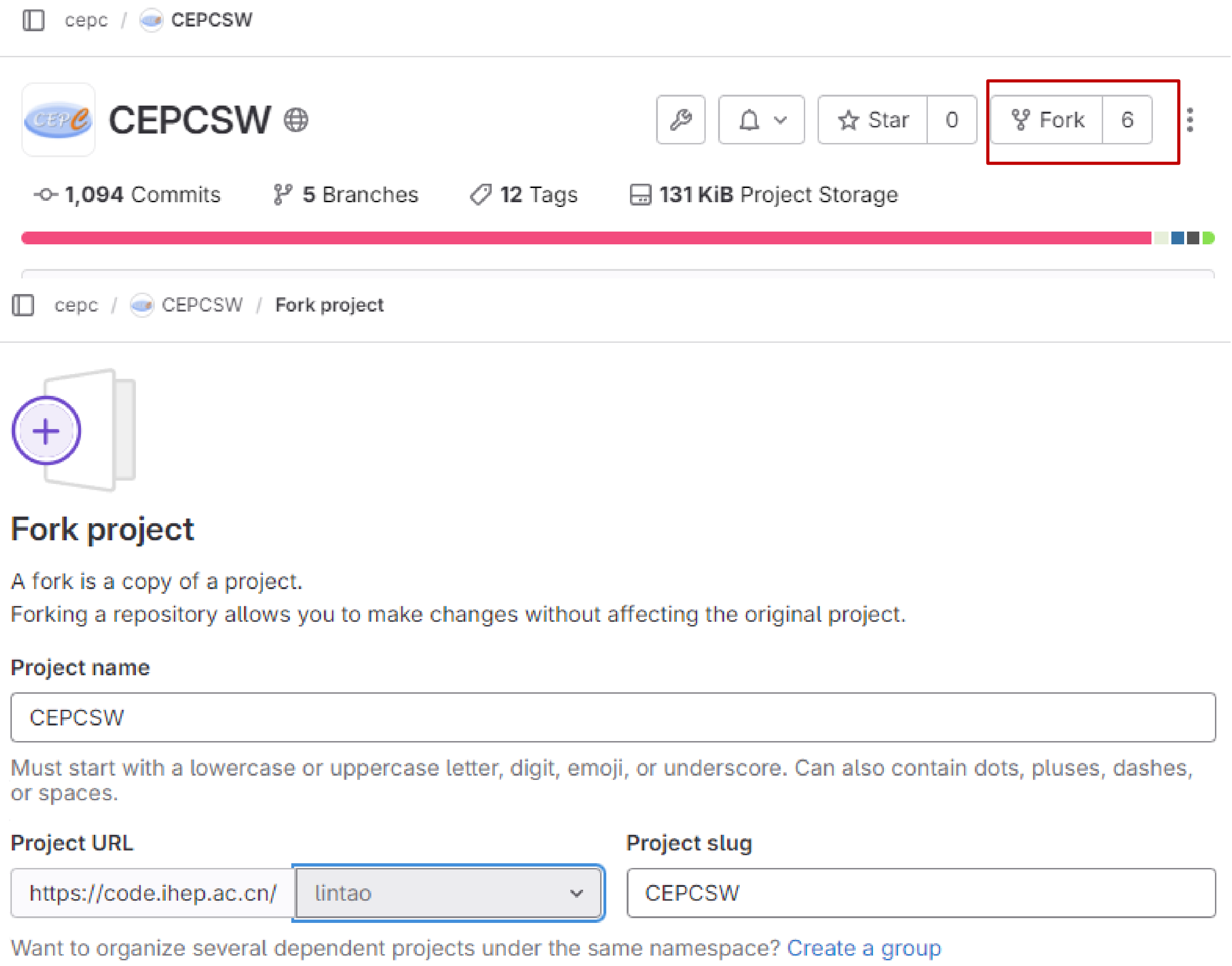


SSH Keys

SSH keys allow you to establish a secure connection between your computer and GitLab. SSH fingerprints verify that the client is connecting to the correct host. Check the current instance configuration.

Fork the CEPCSW

- Fork the source code to your account.



Fork project

A fork is a copy of a project.
Forking a repository allows you to make changes without affecting the original project.

Project name

CEPCSW

Must start with a lowercase or uppercase letter, digit, emoji, or underscore. Can also contain dots, pluses, dashes, or spaces.

Project URL

https://code.ihep.ac.cn/ lintao

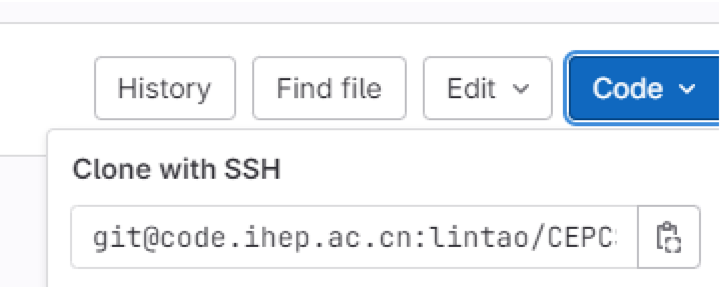
Project slug

CEPCSW

Want to organize several dependent projects under the same namespace? Create a group

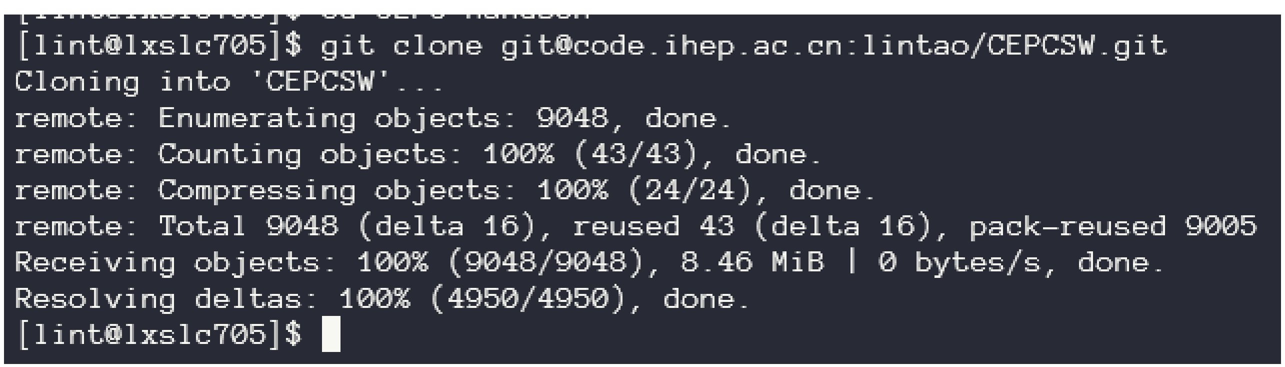
Clone the source code

- After copying the URL, use "git clone" to get the source code.



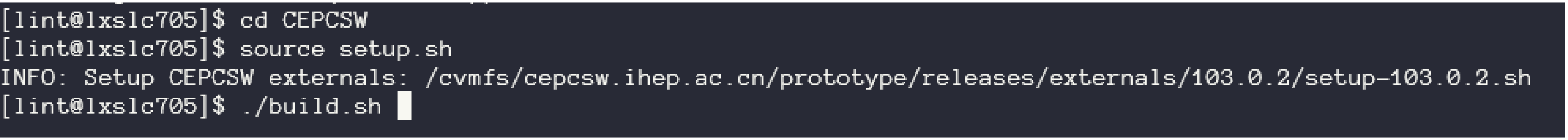
Clone with SSH

git@code.ihep.ac.cn:lintao/CEPC

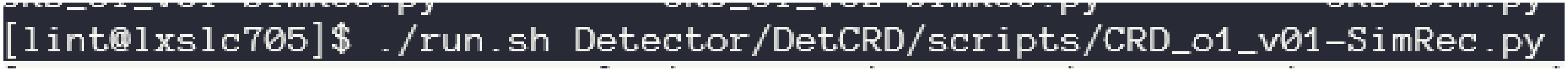


```
[lint@lxslc705]$ git clone git@code.ihep.ac.cn:lintao/CEPCSW.git
Cloning into 'CEPCSW'...
remote: Enumerating objects: 9048, done.
remote: Counting objects: 100% (43/43), done.
remote: Compressing objects: 100% (24/24), done.
remote: Total 9048 (delta 16), reused 43 (delta 16), pack-reused 9005
Receiving objects: 100% (9048/9048), 8.46 MiB | 0 bytes/s, done.
Resolving deltas: 100% (4950/4950), done.
[lint@lxslc705]$
```

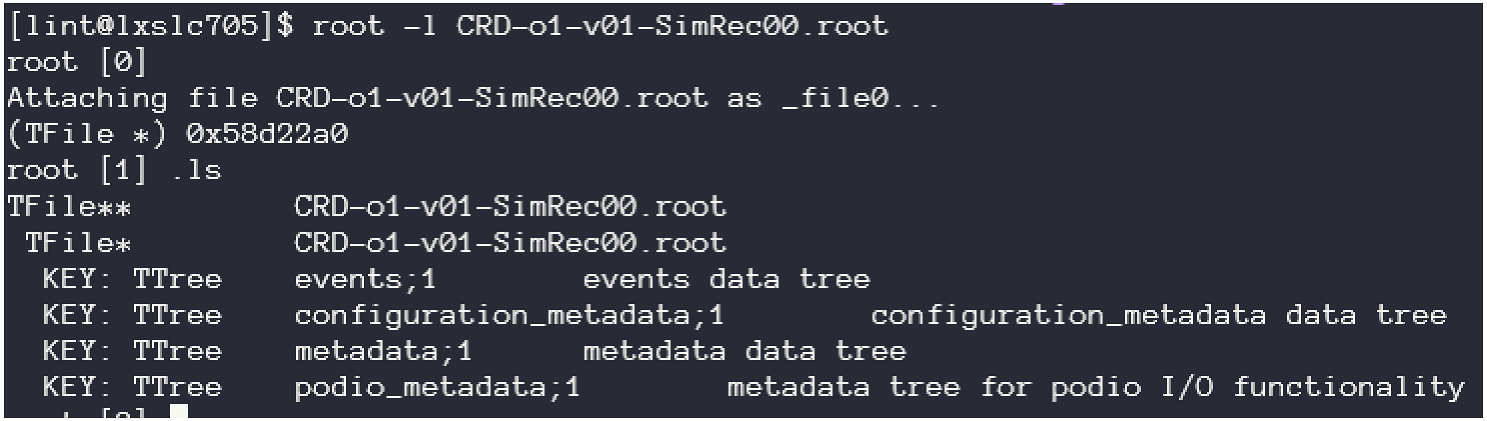
Build and test CEPCSW



```
[lint@lxslc705]$ cd CEPCSW
[lint@lxslc705]$ source setup.sh
INFO: Setup CEPCSW externals: /cvmfs/cepcsw.ihep.ac.cn/prototype/releases/externals/103.0.2/setup-103.0.2.sh
[lint@lxslc705]$ ./build.sh
```



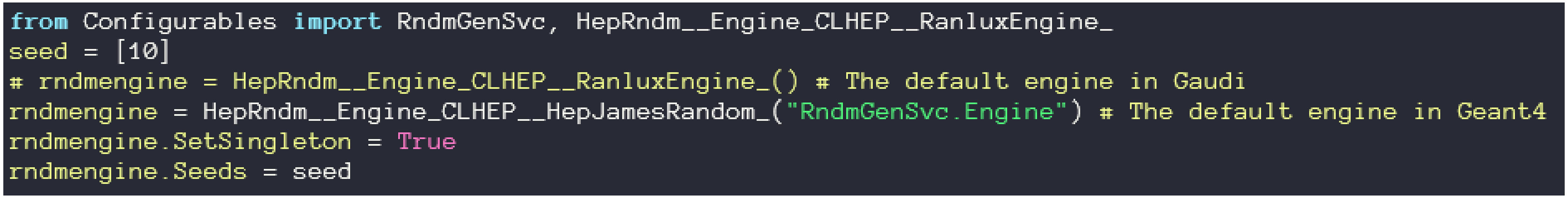
```
[lint@lxslc705]$ ./run.sh Detector/DetCRD/scripts/CRD_o1_v01-SimRec.py
```



```
[lint@lxslc705]$ root -l CRD-o1-v01-SimRec00.root
root [0]
Attaching file CRD-o1-v01-SimRec00.root as _file0...
(TFile *) 0x58d22a0
root [1] .ls
TFile**         CRD-o1-v01-SimRec00.root
 TFile*         CRD-o1-v01-SimRec00.root
  KEY: TTree    events;1        events data tree
  KEY: TTree    configuration_metadata;1        configuration_metadata data tree
  KEY: TTree    metadata;1      metadata data tree
  KEY: TTree    podio_metadata;1        metadata tree for podio I/O functionality
```

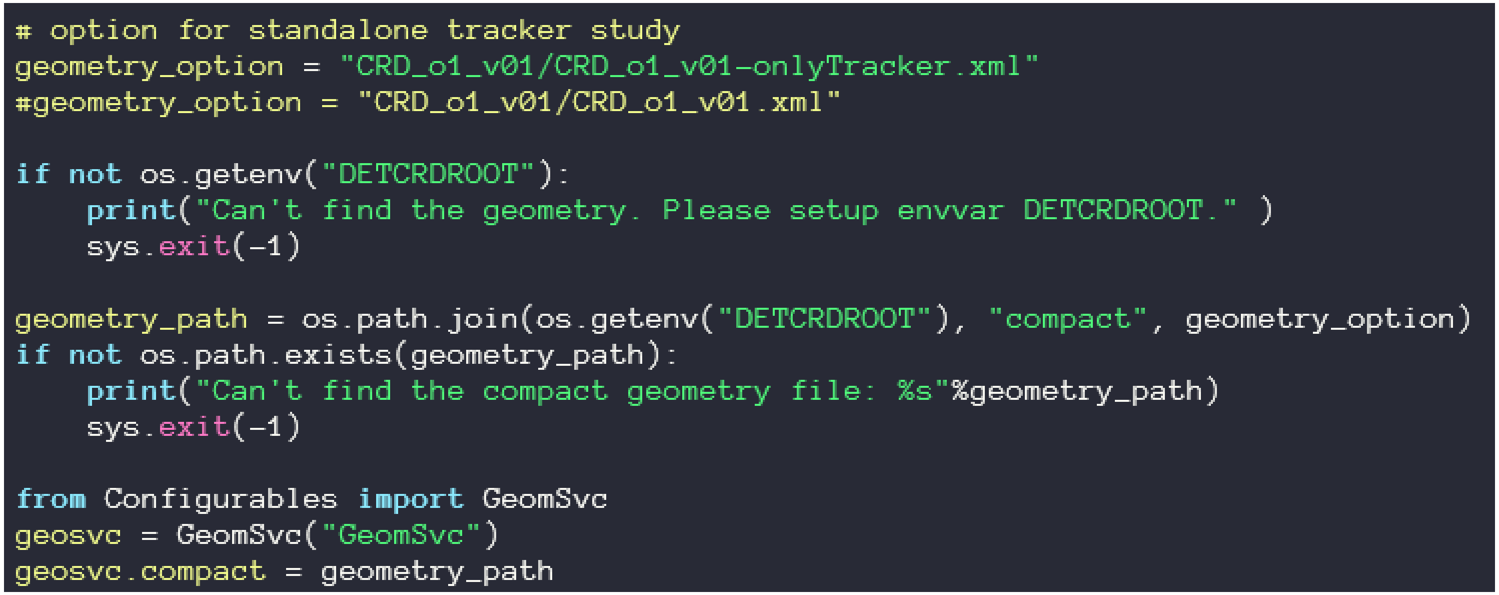
Job option (Gaudi steering file)

- Example: Detector/DetCRD/scripts/CRD_o1_v01-SimRec.py
- Seed of random number



```
from Configurables import RndmGenSvc, HepRndm__Engine_CLHEP__RanluxEngine_
seed = [10]
# rndmengine = HepRndm__Engine_CLHEP__RanluxEngine_() # The default engine in Gaudi
rndmengine = HepRndm__Engine_CLHEP__HepJamesRandom_("RndmGenSvc.Engine") # The default engine in Geant4
rndmengine.SetSingleton = True
rndmengine.Seeds = seed
```

- Geometry



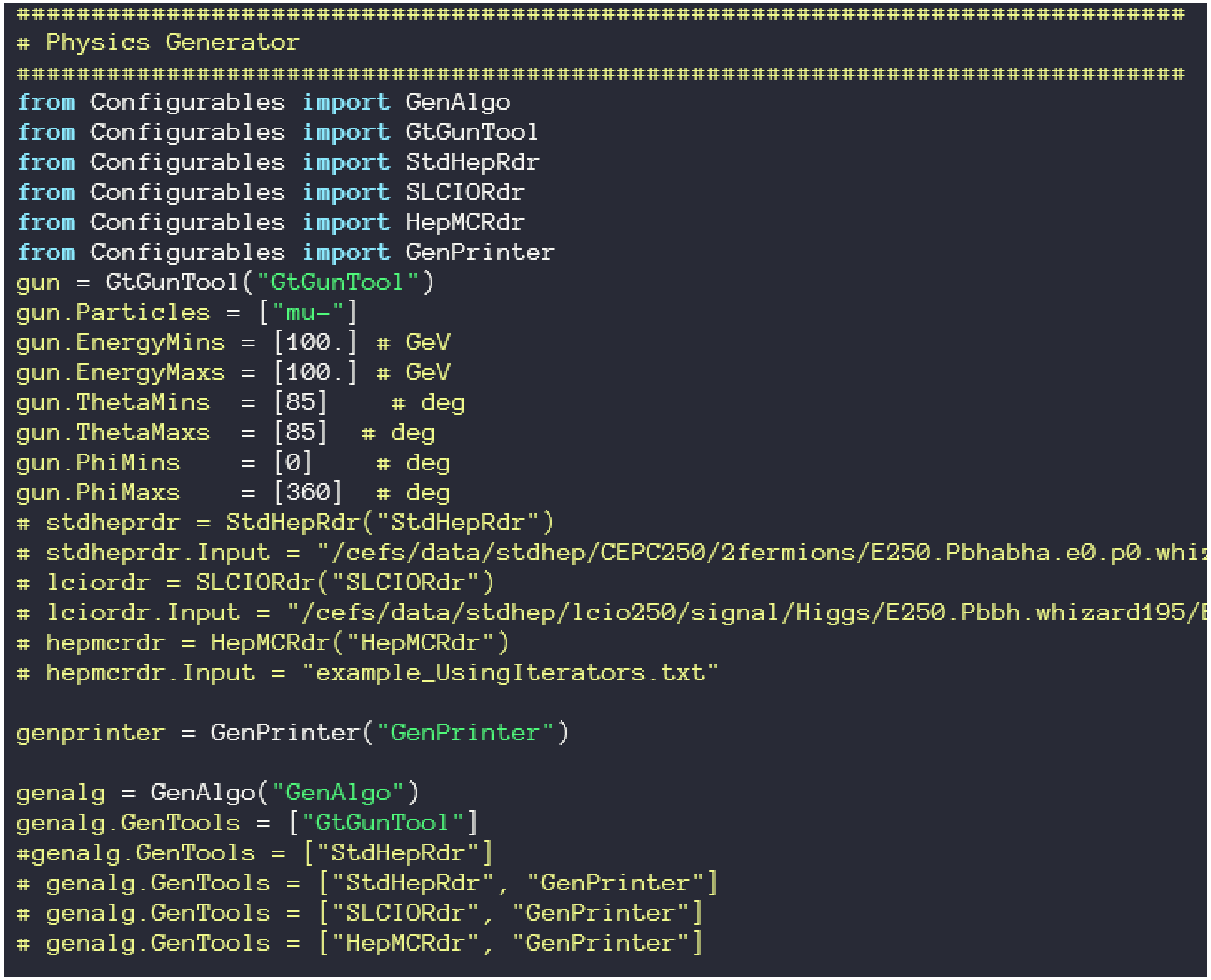
```
# option for standalone tracker study
geometry_option = "CRD_o1_v01/CRD_o1_v01-onlyTracker.xml"
#geometry_option = "CRD_o1_v01/CRD_o1_v01.xml"

if not os.getenv("DETCRDROOT"):
    print("Can't find the geometry. Please setup envvar DETCRDROOT." )
    sys.exit(-1)

geometry_path = os.path.join(os.getenv("DETCRDROOT"), "compact", geometry_option)
if not os.path.exists(geometry_path):
    print("Can't find the compact geometry file: %s"%geometry_path)
    sys.exit(-1)

from Configurables import GeomSvc
geosvc = GeomSvc("GeomSvc")
geosvc.compact = geometry_path
```

❖ Generators: particle gun, StdHep, HepMC



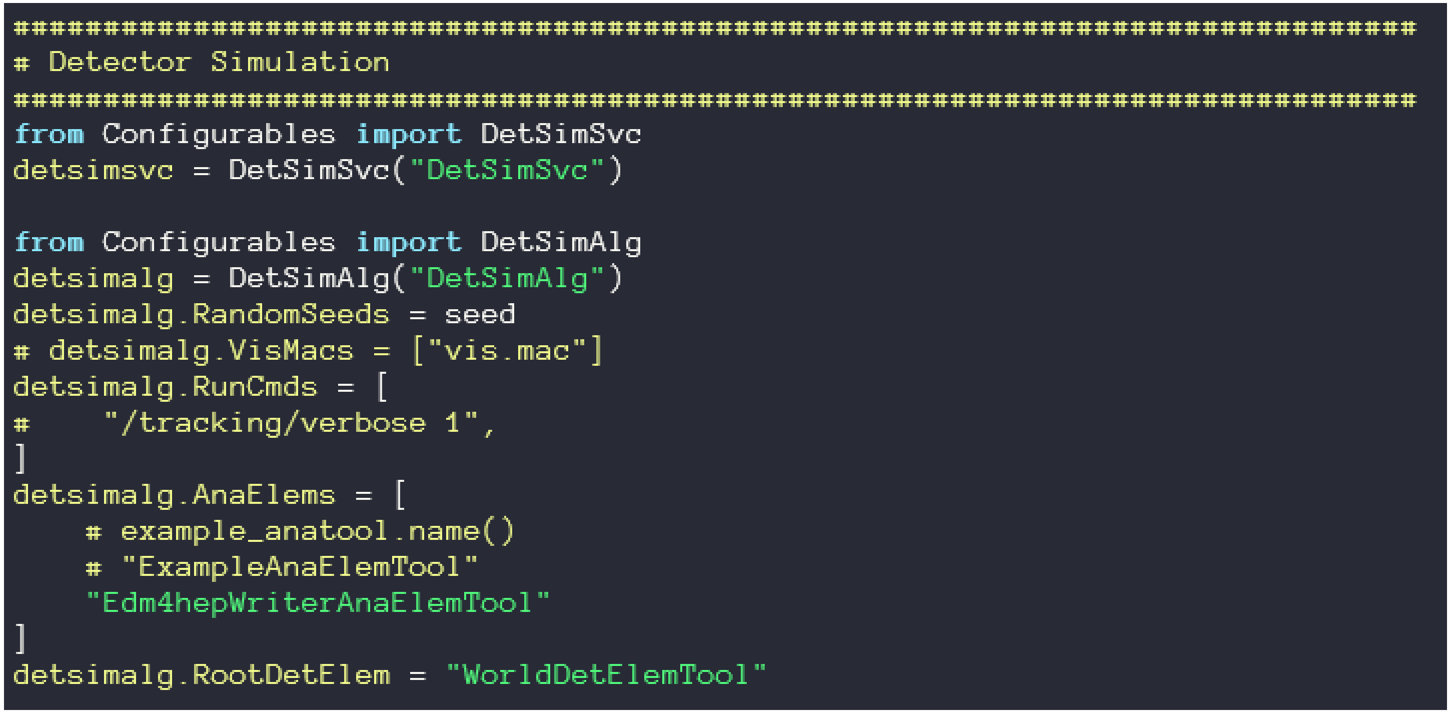
```
###############################################################################
# Physics Generator
###############################################################################
from Configurables import GenAlgo
from Configurables import GtGunTool
from Configurables import StdHepRdr
from Configurables import SLCIORdr
from Configurables import HepMCRdr
from Configurables import GenPrinter
gun = GtGunTool("GtGunTool")
gun.Particles = ["mu-"]
gun.EnergyMins = [100.] # GeV
gun.EnergyMaxs = [100.] # GeV
gun.ThetaMins  = [85]    # deg
gun.ThetaMaxs  = [85]  # deg
gun.PhiMins    = [0]    # deg
gun.PhiMaxs    = [360]  # deg
# stdheprdr = StdHepRdr("StdHepRdr")
# stdheprdr.Input = "/cefs/data/stdhep/CEPC250/2fermions/E250.Pbhabha.e0.p0.whi
# lciordr = SLCIORdr("SLCIORdr")
# lciordr.Input = "/cefs/data/stdhep/lcio250/signal/Higgs/E250.Pbbh.whizard195/E
# hepmcrdr = HepMCRdr("HepMCRdr")
# hepmcrdr.Input = "example_UsingIterators.txt"

genprinter = GenPrinter("GenPrinter")

genalg = GenAlgo("GenAlgo")
genalg.GenTools = ["GtGunTool"]
#genalg.GenTools = ["StdHepRdr"]
# genalg.GenTools = ["StdHepRdr", "GenPrinter"]
# genalg.GenTools = ["SLCIORdr", "GenPrinter"]
# genalg.GenTools = ["HepMCRdr", "GenPrinter"]
```

❖ Detector simulation

- The RunCmds are the Geant4 macros. Useful to debug.



```
##############################################################################
# Detector Simulation
##############################################################################
from Configurables import DetSimSvc
detsimsvc = DetSimSvc("DetSimSvc")

from Configurables import DetSimAlg
detsimalg = DetSimAlg("DetSimAlg")
detsimalg.RandomSeeds = seed
# detsimalg.VisMacs = ["vis.mac"]
detsimalg.RunCmds = [
#    "/tracking/verbose 1",
]
detsimalg.AnaElems = [
    # example_anatool.name()
    # "ExampleAnaElemTool"
    "Edm4hepWriterAnaElemTool"
]
detsimalg.RootDetElem = "WorldDetElemTool"
```