Simulation Framework in CEPCSW

Tao Lin (IHEP)
representing CEPC software team

2024 CEPC Mechanics Workshop, Luoyang, Henan
22-24 August 2024

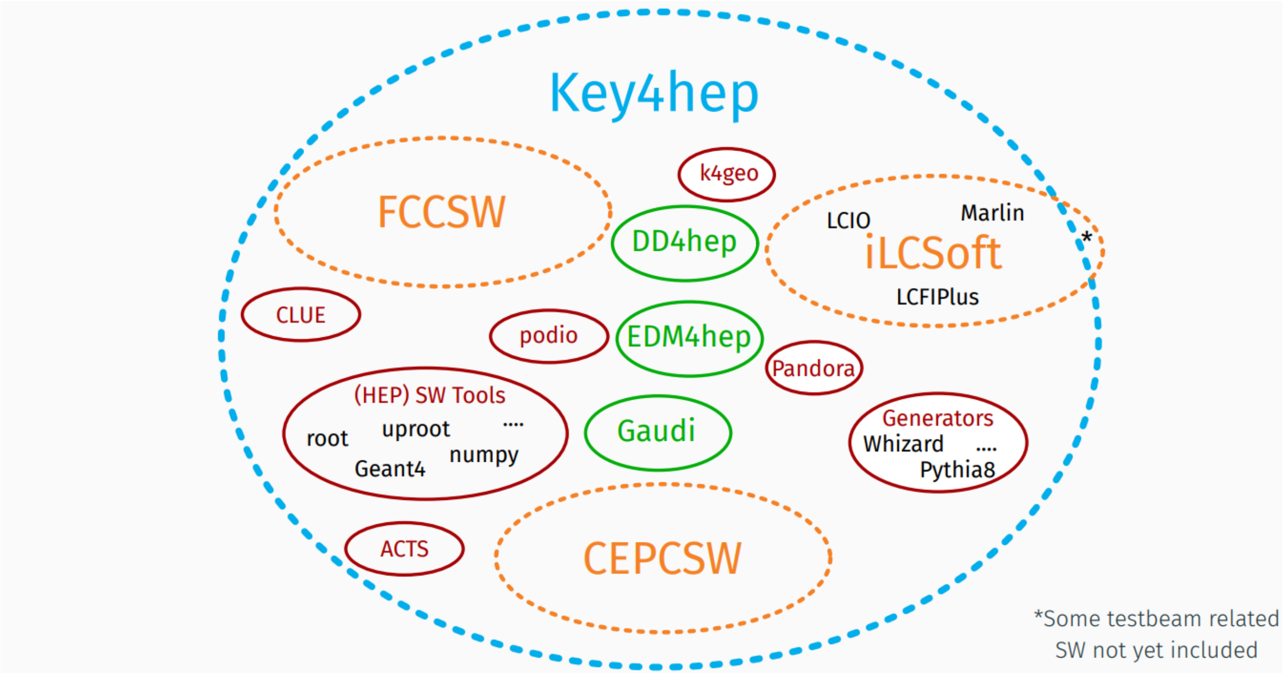
Outline

- Introduction to CEPCSW
- The simulation framework in CEPCSW
  - Physics generator interface
  - Detector description
  - Geant4 based simulation
  - Background mixing
  - Fast simulation
- R&D in the simulation framework
- Summary

Introduction

- The development of CEPC software started with the iLCSoft
  - Developed CEPC components for simulation and reconstruction
  - Generated M.C. data for detector design and physics potential studies
  - Particularly, CEPC CDR studies done with the iLCSoft
- 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
  - Maximize the sharing of software components among different experiments

T.Madlener | Key4hep & EDM4hep
CEPC workshop, Edinburgh

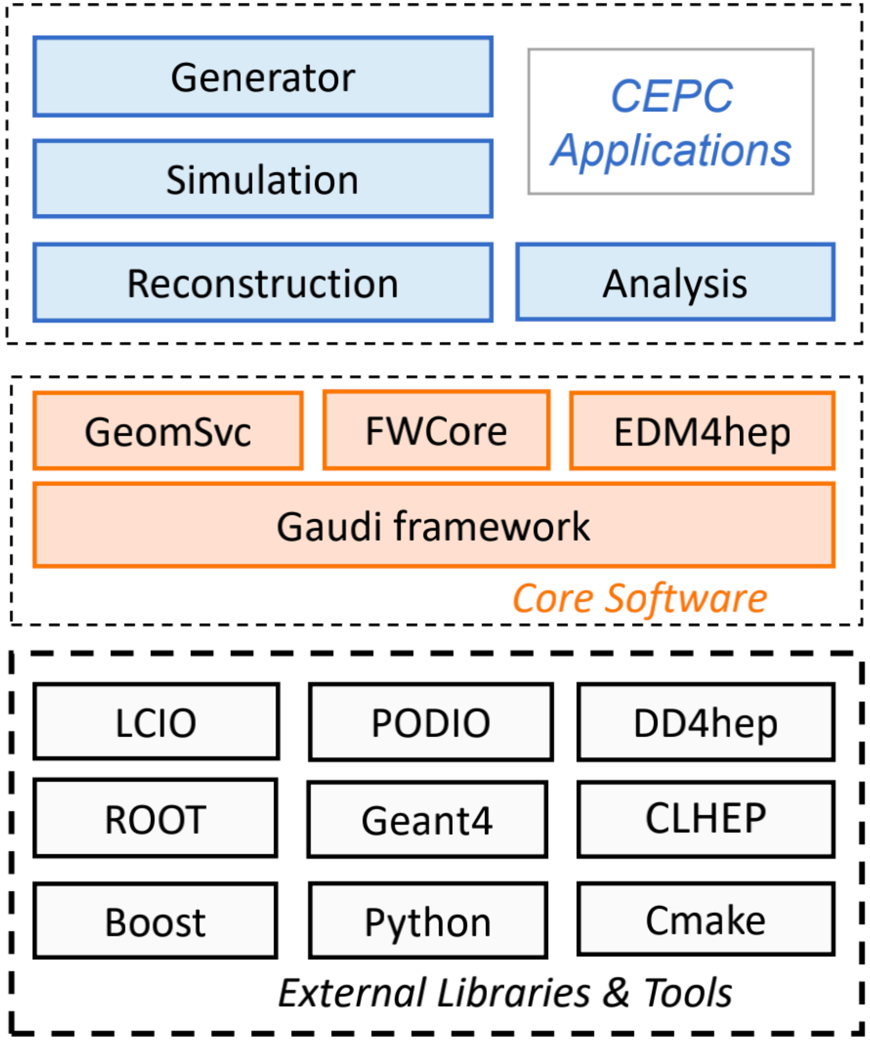


Key4hep project: https://github.com/key4hep
CEPCSW is the first application based on Key4hep.

Architecture of CEPCSW

- CEPCSW is organized as a multi-layer structure
  - Applications: simulation, reconstruction and analysis
  - Core software
  - External libraries

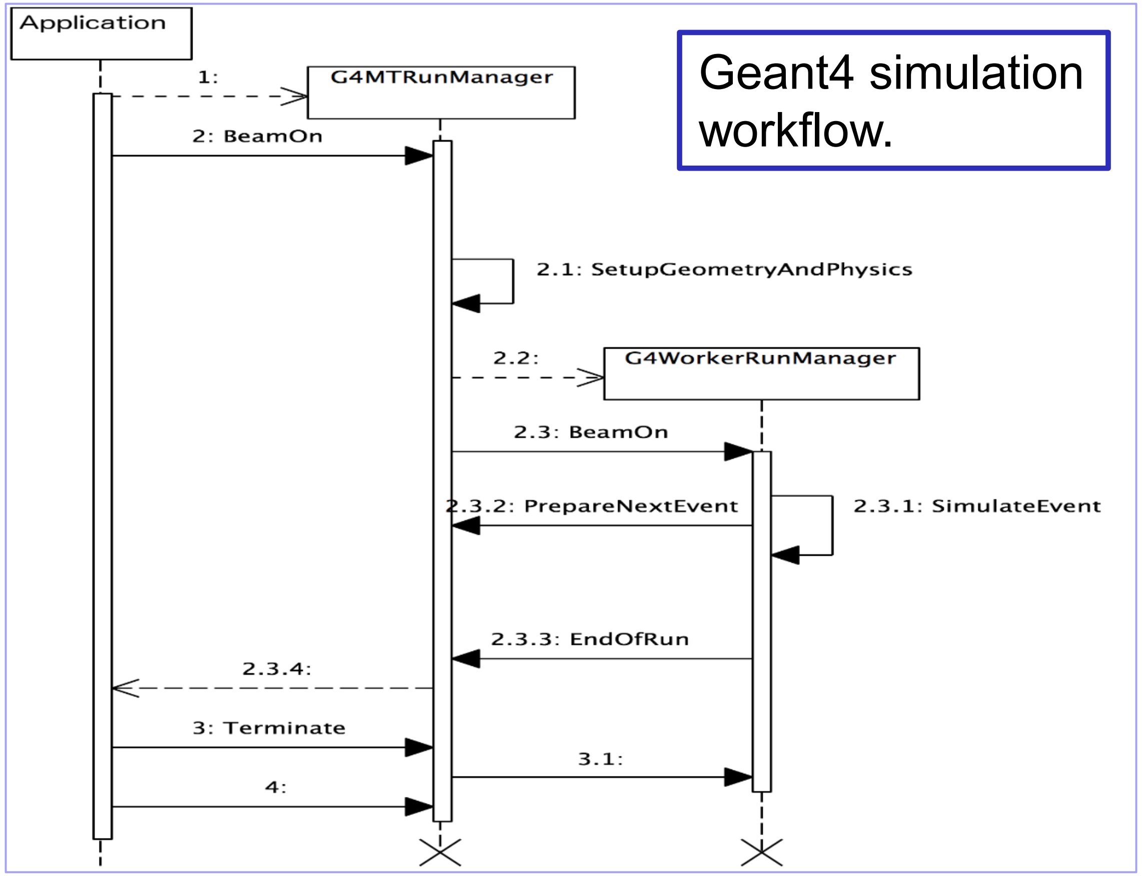
- The key components of core software include:
  - Gaudi: defines interfaces to all software components and controls their execution
  - EDM4hep: generic event data model
  - K4FWCore: manages the event data
  - DD4hep: geometry description
  - CEPC-specific framework software: generator, Geant4 simulation, beam background mixing, fast simulation, machine learning interface, etc.



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

Simulation framework in CEPCSW

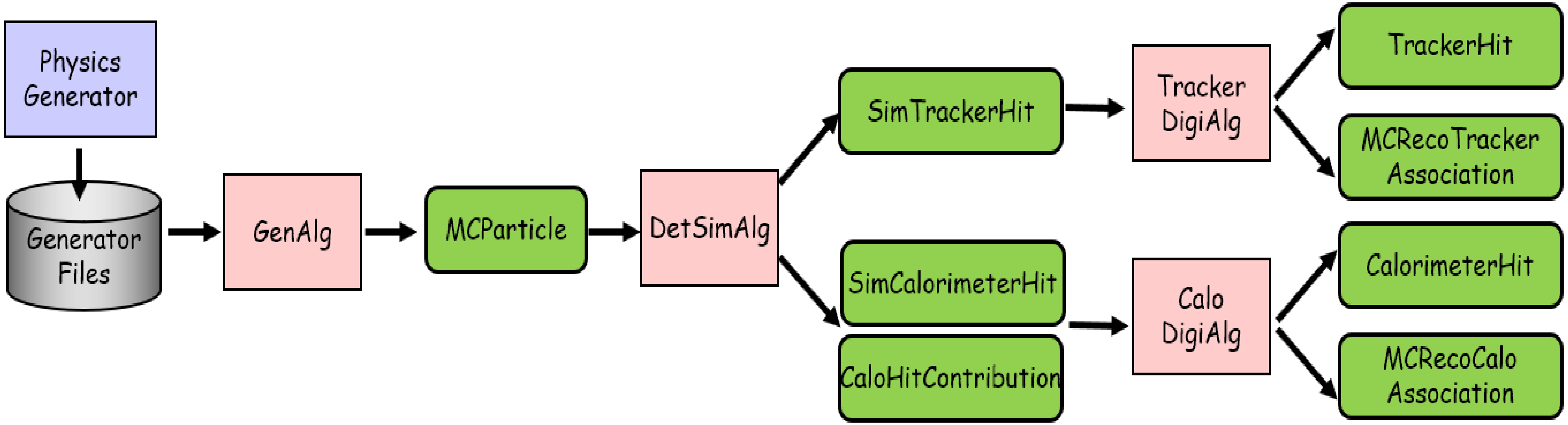
- Simulations have become critical for
  - the design of detectors
  - the development of reconstruction algorithms
- Three stages in a simulation chain
  - Physics generator: produce primary particles.
  - Detector simulation: produce hits.
  - Digitization: produce digits.
- The simulation framework provides the abilities to run the simulation chain easily.
  - Case 1: load the events from physics generators into detector simulation.
  - Case 2: control the simulation workflow.
  - Case 3: change the detector designs.
  - Case 4: support the background mixing.
  - …



For example, Geant4 implements its own run managers to control the simulation workflow.
Need to make Geant4 work with the other algorithms.

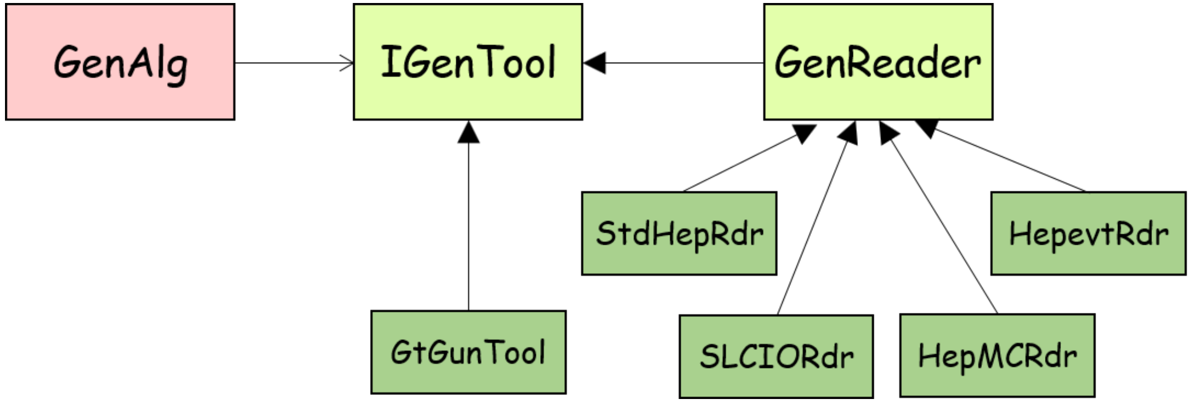
Complete simulation chain with EDM4hep

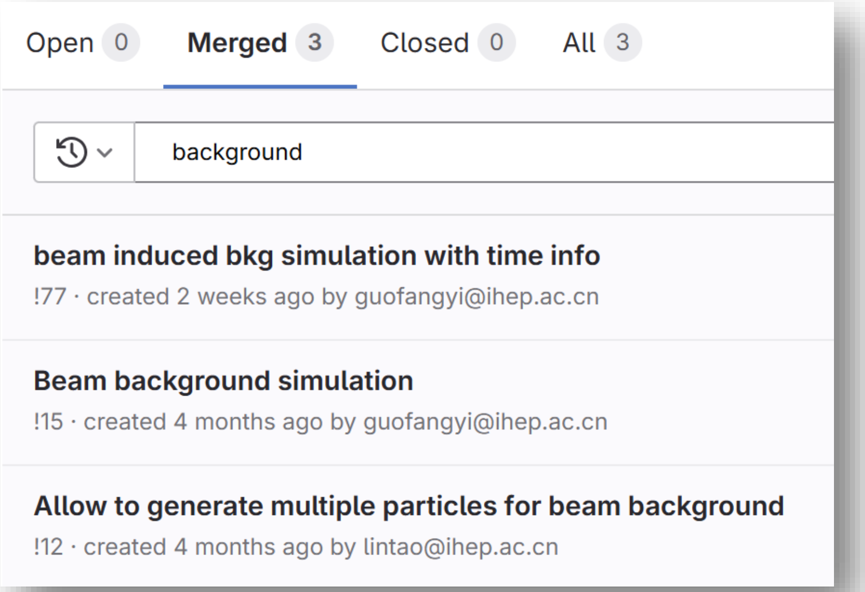
- Follow the design of Gaudi framework, several algorithms are implemented for the three stages respectively.
- Event Data Model acts as *standards* between different algorithms.
- EDM4hep:
  - Physics generator: MCParticle
  - Detector Simulation: MCParticle (with secondaries), SimTrackerHit, SimCalorimeterHits
  - Digitization: TrackerHit, CalorimeterHit



Physics generator interface

- Physics generators with different formats are integrated
  - StdHep, HepEvt, LCIO, HepMC formats.
- Particle gun is supported.
  - Multiple particles
- Beam backgrounds
  - Support to generate multiple tracks according to the rates.
  - Support the uniform time distribution in a time window.
  - MDI group also provides ROOT files, which support random access. So the start index is not always 0.



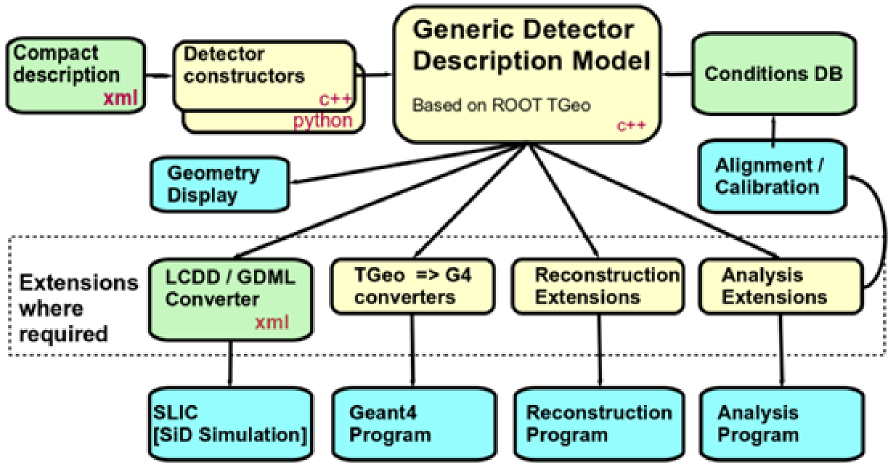


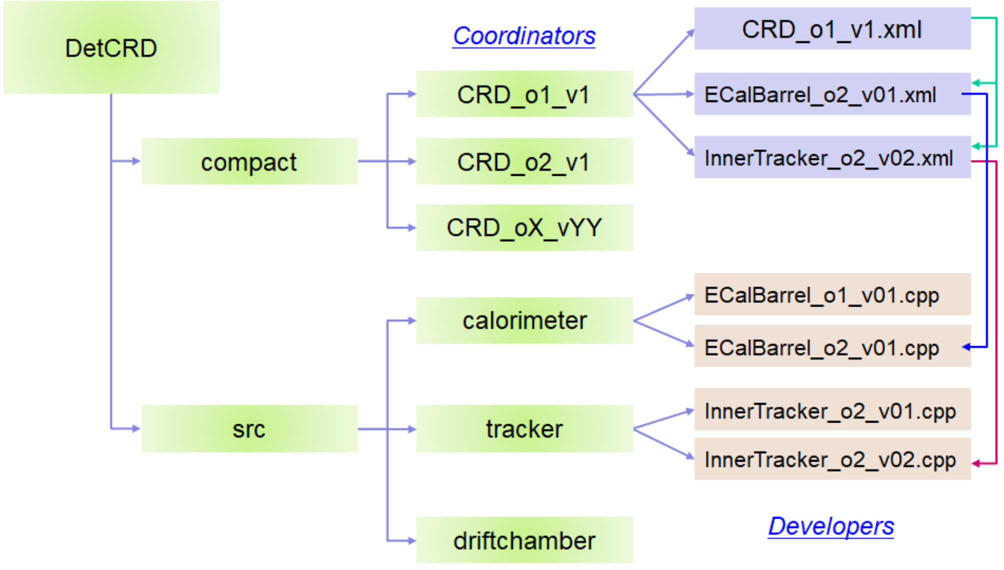
Detector Description (1)

- DD4hep is adopted to provide the full detector description with a single source of information.
  - Consists of C++ constructors and XML based compact files
- Different detector options are managed in a git repository.
  - Easy to setup detectors and compare between different options.
- Available options in CEPCSW
  - CEPCv4: baseline detector in Conceptual Design Report
  - TDR: The TDR Detector

Geometry: Chengdong Fu

Model	Description	MainTracker	Ecal	Hcal	Status
TDR_o1_v01	long barrel vertex, TPC	SIT+TPC+SET	crystal	Glass	developing
TDR_o1_v02	short barrel vertex, TPC	SIT+TPC+SET	crystal	Glass	developing
TDR_o2_v01	long barrel vertex, DC	SIT+DC +SET	crystal	Glass	developing
TDR_o2_v02	short barrel vertex, DC	SIT+DC +SET	crystal	Glass	developing

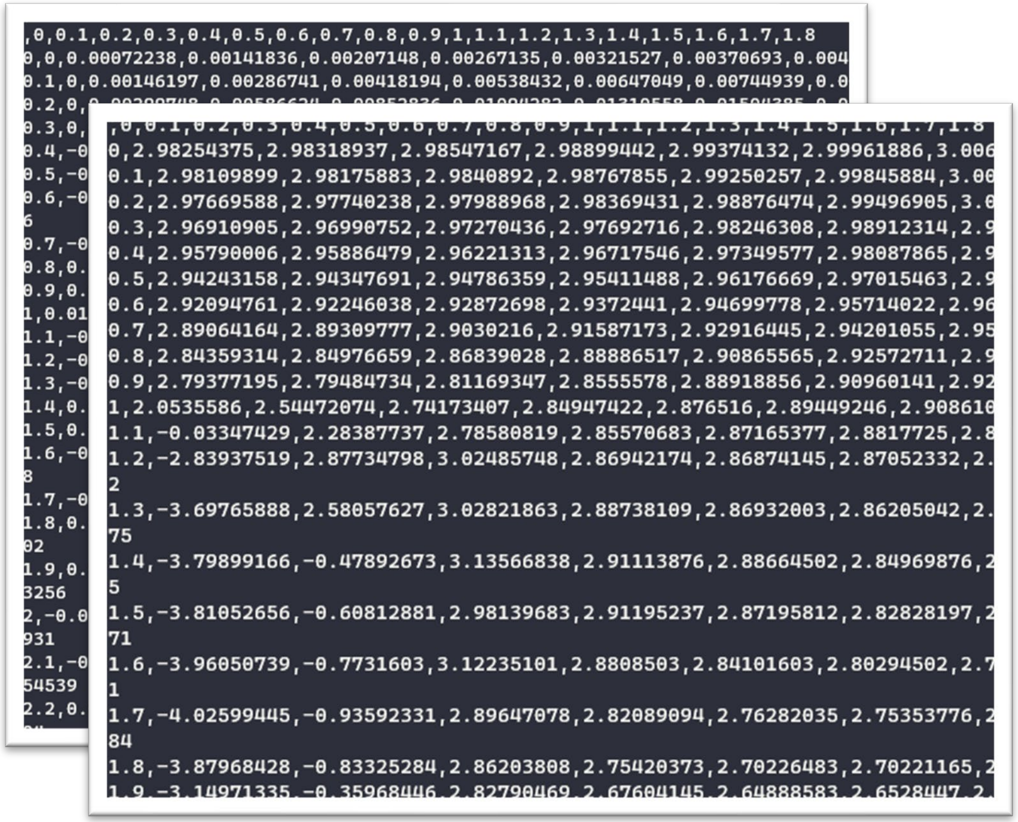




Schematic view of detector management

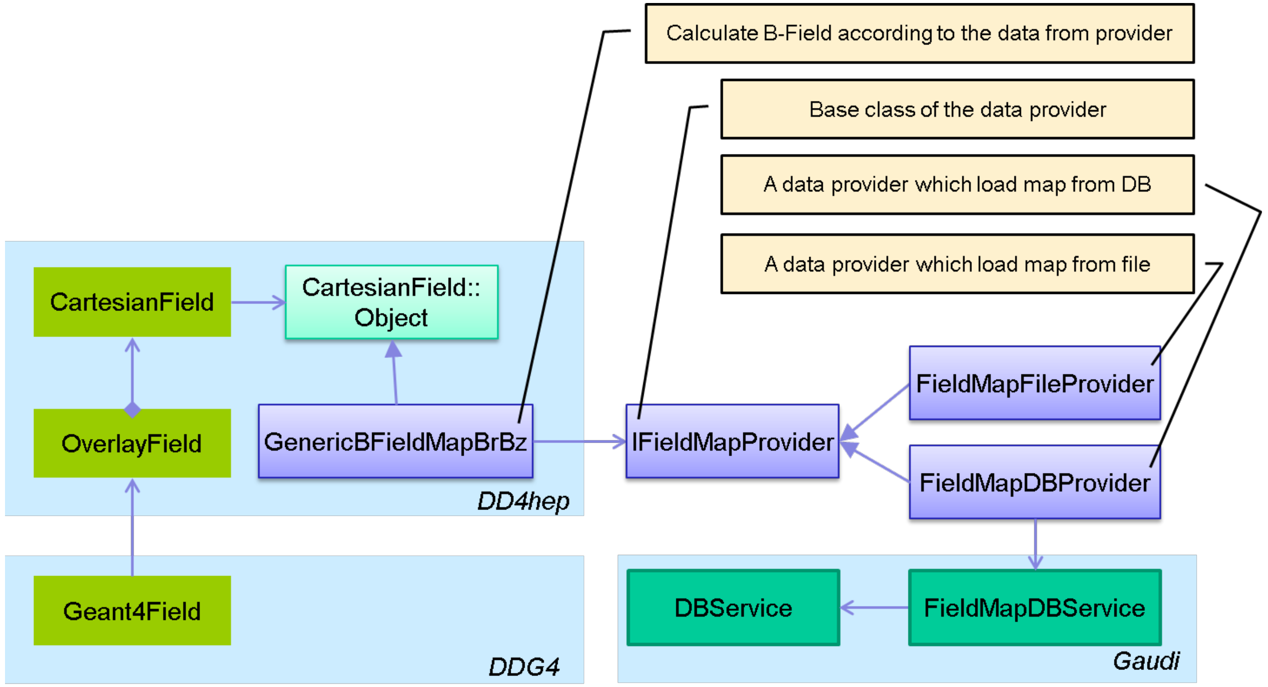
Detector Description (2)

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



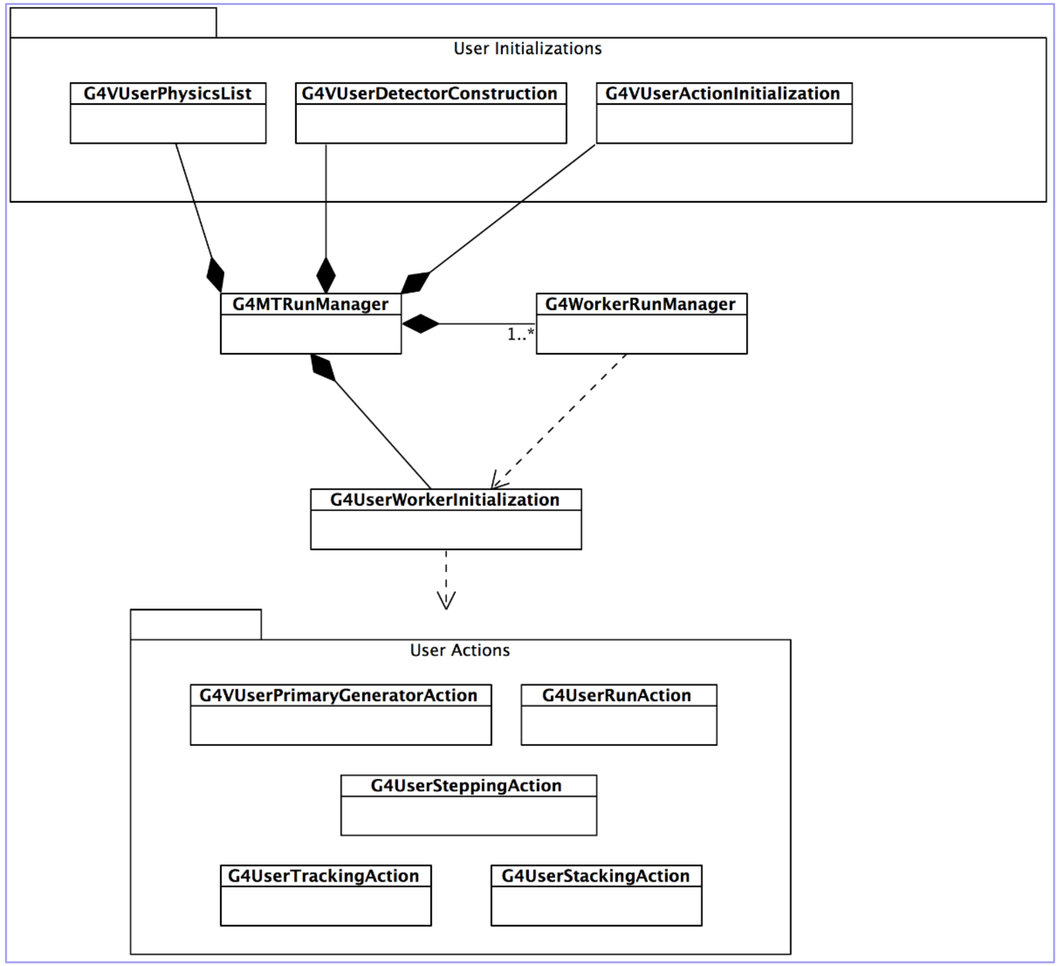
```
,0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1,1.1,1.2,1.3,1.4,1.5,1.6,1.7,1.8
0,0,0.00072238,0.00141836,0.00207148,0.00267135,0.00321527,0.00370693,0.004
0.1,0,0.00146197,0.00286741,0.00418194,0.00538432,0.00647049,0.00744939,0.0
```

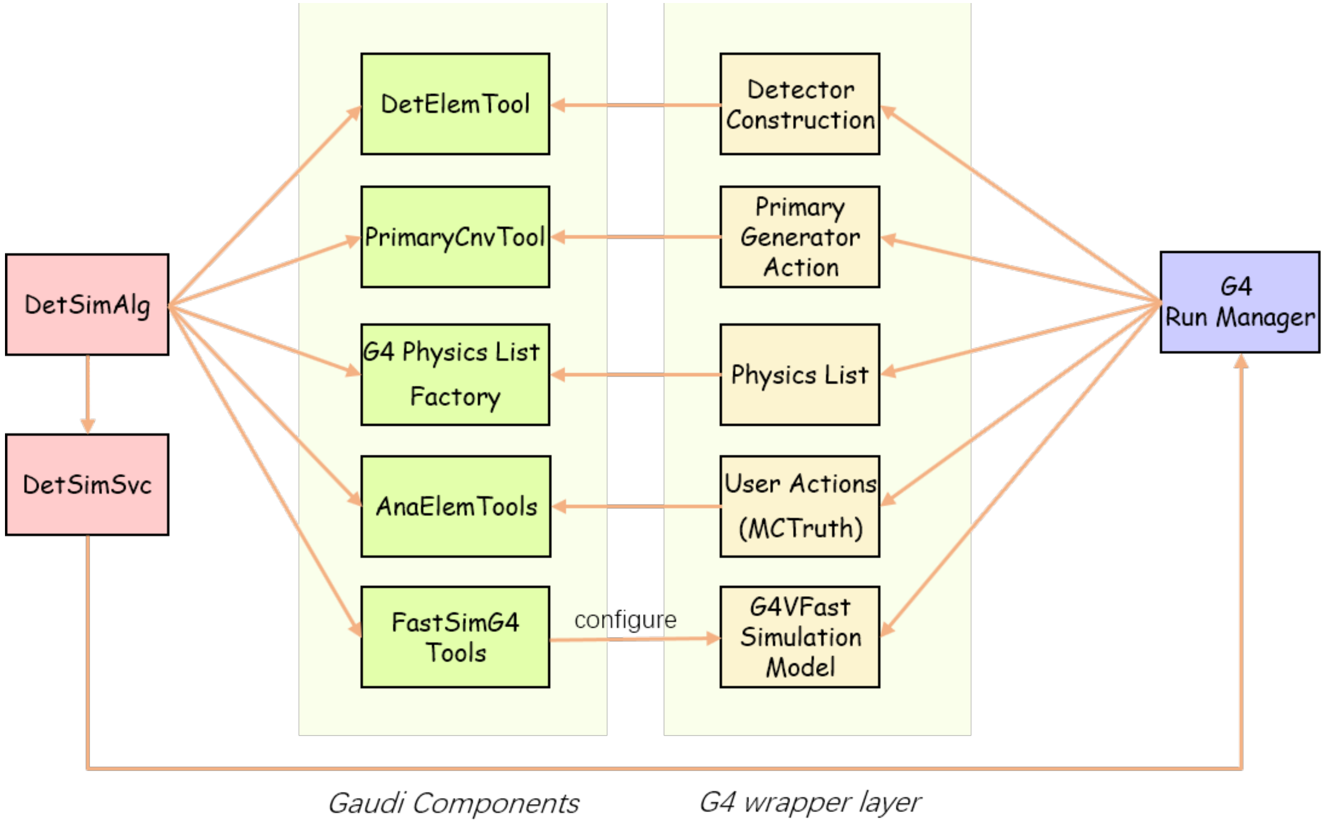
```
,0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1,1.1,1.2,1.3,1.4,1.5,1.6,1.7,1.8
0,2.98254375,2.98318937,2.98547167,2.98899442,2.99374132,2.99961886,3.006
0.1,2.98109899,2.98175883,2.9840892,2.98767855,2.99250257,2.99845884,3.00
0.2,2.97669588,2.97740238,2.97988968,2.98369431,2.98876474,2.99496905,3.0
0.3,2.96910905,2.96990752,2.97270436,2.97692716,2.98246308,2.98912314,2.9
0.4,2.95790006,2.95886479,2.96221313,2.96717546,2.97349577,2.98087865,2.9
0.5,2.94243158,2.94347691,2.94786359,2.95411488,2.96176669,2.97015463,2.9
0.6,2.92094761,2.92246038,2.92872698,2.9372441,2.94699778,2.95714022,2.96
0.7,2.89064164,2.89309777,2.9030216,2.91587173,2.92916445,2.94201055,2.95
0.8,2.84359314,2.84976659,2.86839028,2.88886517,2.90865565,2.92572711,2.9
0.9,2.79377195,2.79484734,2.81169347,2.8555578,2.88918856,2.90960141,2.92
1,2.0535586,2.54472074,2.74173407,2.84947422,2.876516,2.89449246,2.908610
1.1,-0.03347429,2.28387737,2.78580819,2.85570683,2.87165377,2.8817725,2.8
1.2,-2.83937519,2.87734798,3.02485748,2.86942174,2.86874145,2.87052332,2.
2
1.3,-3.69765888,2.58057627,3.02821863,2.88738109,2.86932003,2.86205042,2.
75
1.4,-3.79899166,-0.47892673,3.13566838,2.91113876,2.88664502,2.84969876,2
5
1.5,-3.81052656,-0.60812881,2.98139683,2.91195237,2.87195812,2.82828197,2
71
1.6,-3.96050739,-0.7731603,3.12235101,2.8808503,2.84101603,2.80294502,2.7
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.
```



Geant4 based simulation

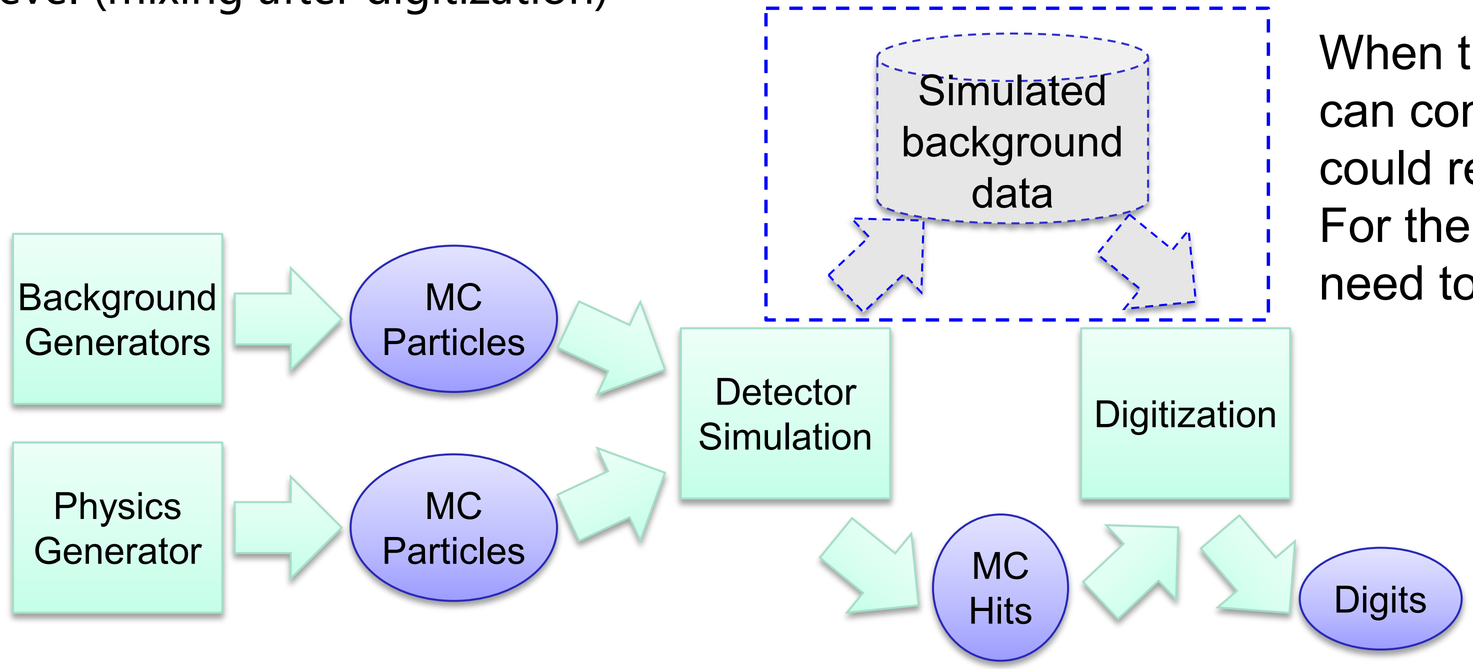
- Additional layers are implemented between Gaudi and Geant4.
  - Take controls of initialization and event loop.





Background mixing

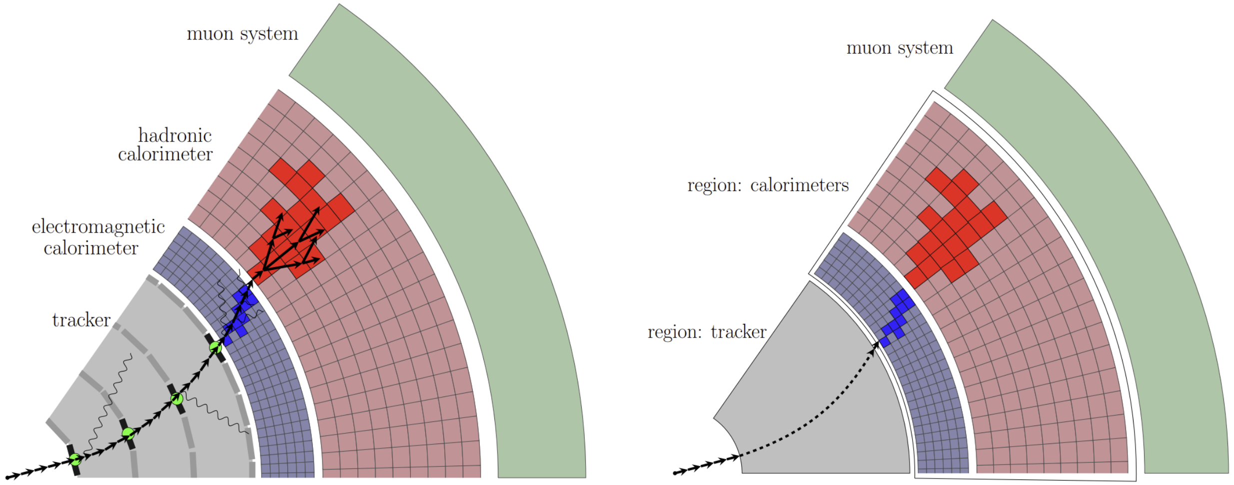
- There are several different ways to mix the backgrounds.
  - ✓ Primary particle level (mixing before detector simulation)
  - Hit level (mixing after detector simulation)
  - Digit level (mixing after digitization)
- Need balances in CPU, memory and I/O.
- Currently, the MDI group uses the first way.
  - It is close to reality, but time consuming.
  - The events need to be simulated again every time.



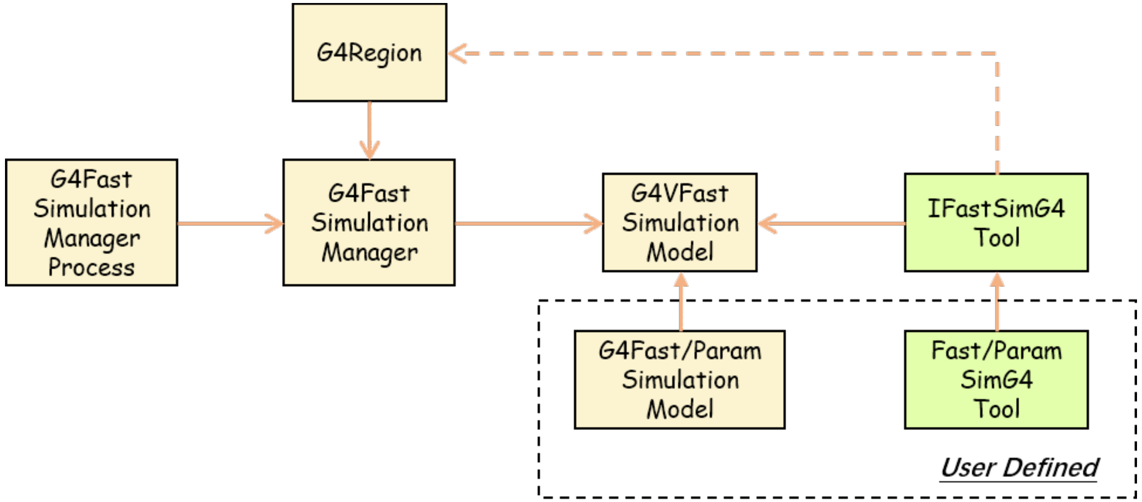
When the detector design is fixed, we can consider to the second option. It could reduce a lot of CPU time.
For the hit level mixing, the data files need to be created before digitization.

Fast simulation (1)

- Geant4 simulation could be time consuming. The idea of fast simulation is to replace the consuming part with a fast algorithm, such as parameterization method.
- Region based fast simulation is adopted
  - When a particle enter a region, fast simulation will be triggered by Geant4.

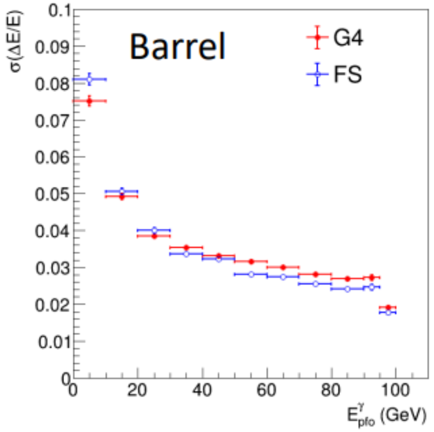


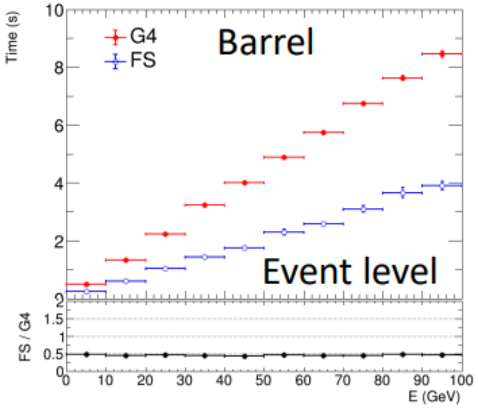
A Zaborowska 2017 J. Phys.: Conf. Ser. 898 042053



Fast simulation (2)

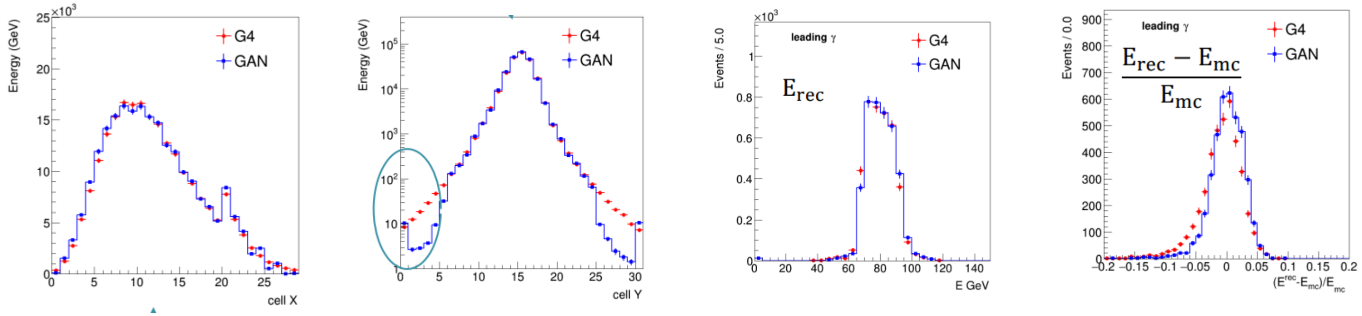
- Traditional method: shower parameterization, frozen shower, …



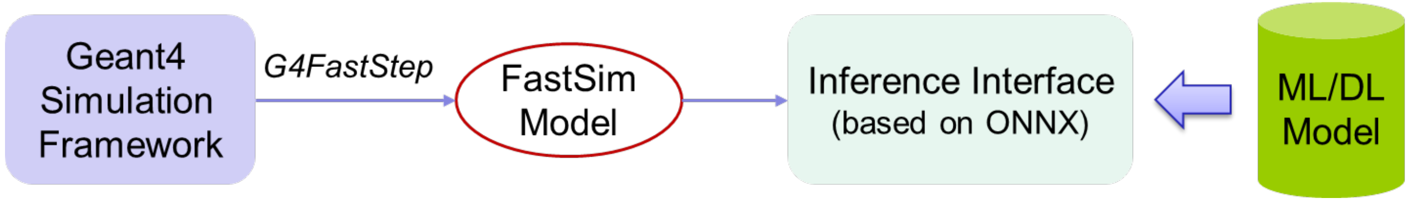


- ML-based method: GAN, VAE, NF, Diffusion, …
  - In general the agreement between GAN and Geant4 is good, while some distributions still need further improvement

Wenxing Fang



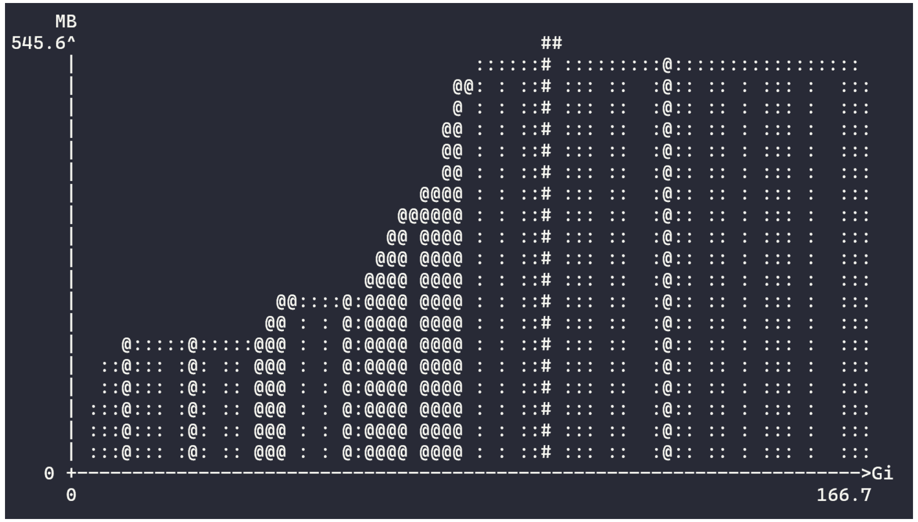
- Support ML methods via ONNX inference interface.
  - Fast pulse simulation in the drift chamber provided as an example (MLP)



R&D in simulation framework

- Challenges in the current simulation framework
  - Due to the more precise geometries and physics, more memory will be used.
  - An efficient solution is using the multi-threaded techniques, which requires the multi-threading design.

Memory usage (serial version)



Two different approaches for Gaudi and Geant4.

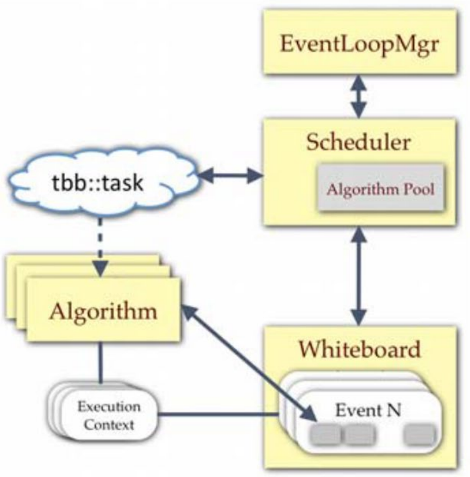


Figure 1. General structure of the GaudiHive framework

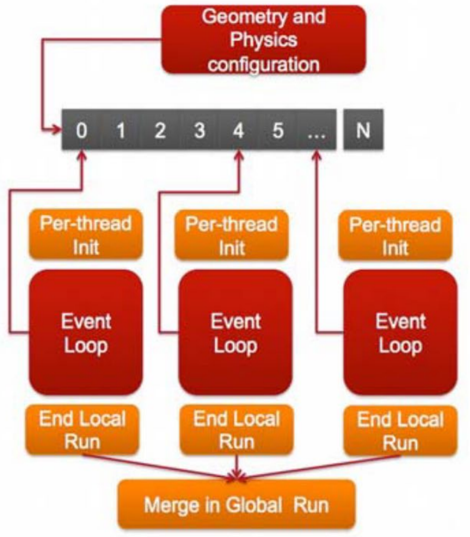


Figure 2. A sketch of the structure of a Geant4-MT simulation

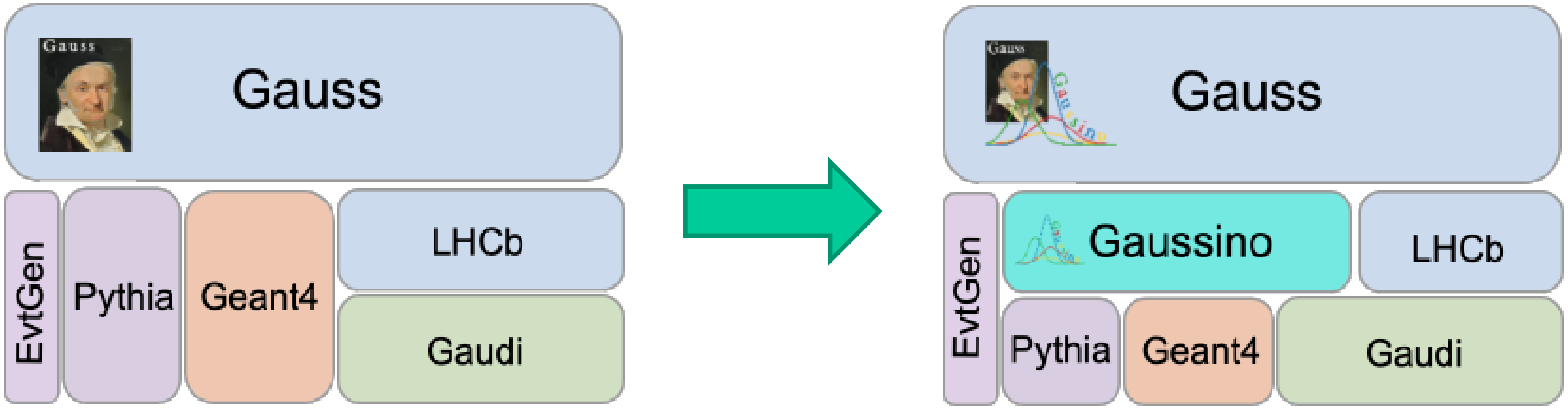
Simulation setup:
- Detector: TDR_o1_v01
- Physics list: QGSP_BERT
- Generation: single muons
- N events: 100

The RSS memory is about **950MB at initialization stage**.
The figure shows the **heap memory usage**.

A Di Simone and on behalf of the ATLAS Collaboration 2017 J. Phys.: Conf. Ser. 898 042010

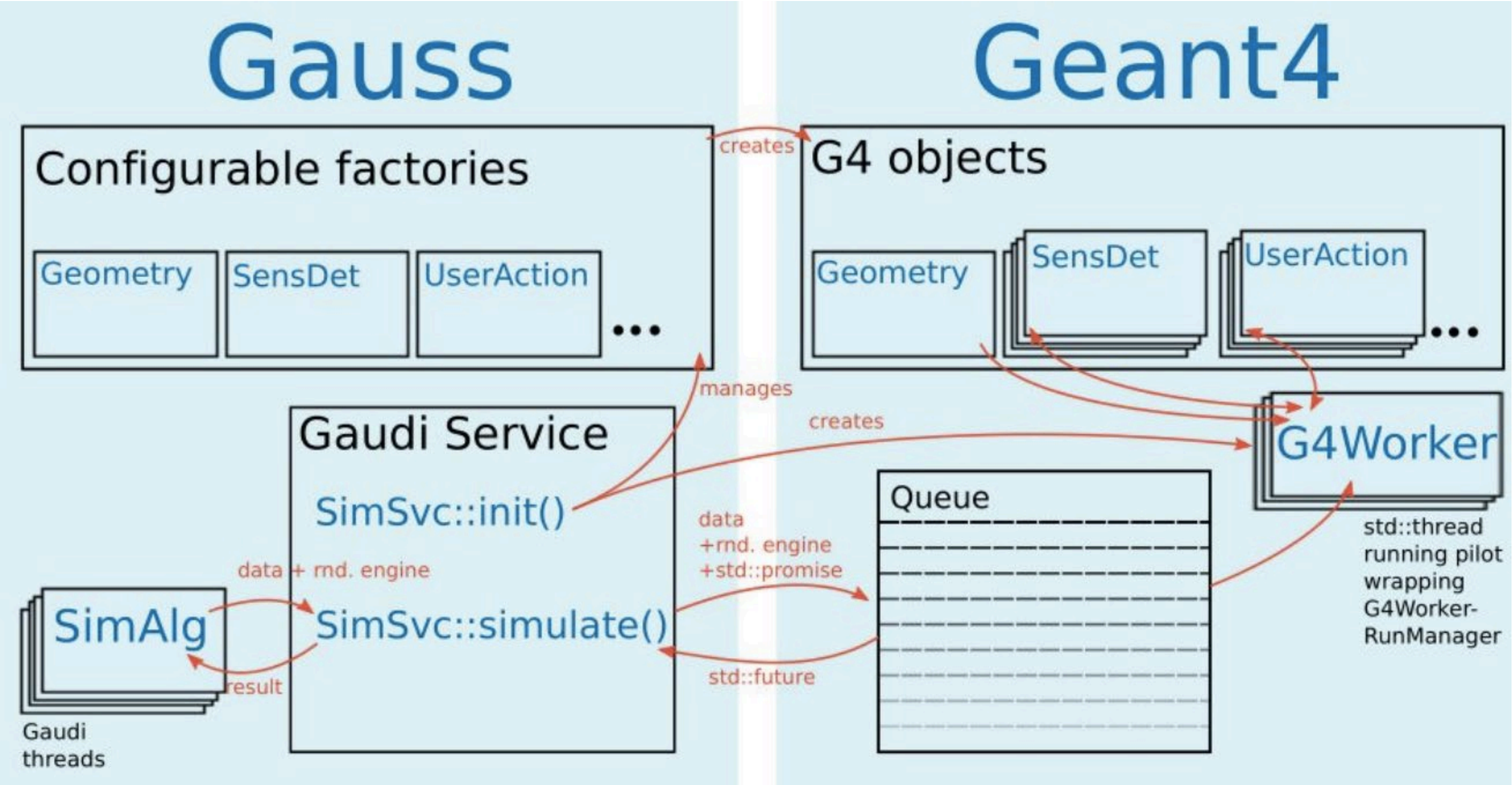
Gaussino: simulation framework from LHCb

- Question: Develop a new one or adopt an existing one?
- Evolution of the simulation framework from LHCb
  - The underlying framework is moving to Gaudi Functional and Gaudi Hive
  - Better support for multi-threading, machine learning, fast simulation methods
  - Gauss-on-Gaussino is a new version of LHCb simulation framework
- Gaussino is being added to Key4hep by extracting experiment-independent parts from Gauss.



Multithreading in Gaussino

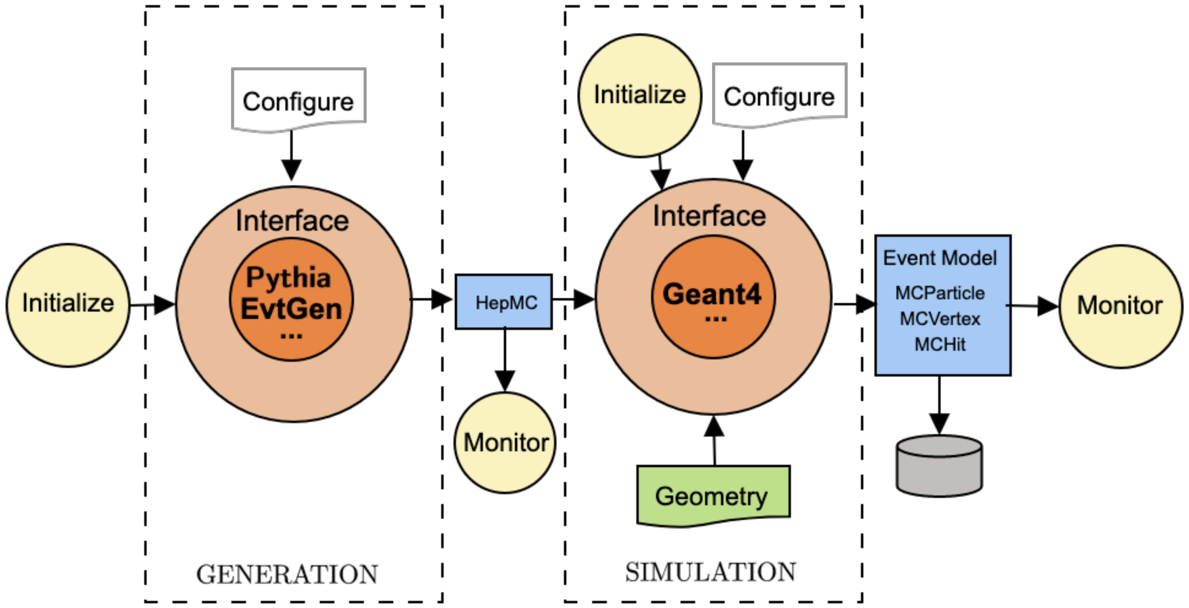
- A thread-safe queue is used to communicate between Gaudi and Geant4



Sim/GiGaMTCore/include/GiGaMTCoreRun/GiGaWorkerPilot.h

Core components of Gaussino

- Gaussino is a thread-safe simulation framework based on Gaudi Functional and provides interfaces to Pythia and Geant4.
- Modular design
  - Gaudi Functional Algorithms
  - Gaudi Tools
- Four components
  - Generation of events
  - The detector simulation
  - Geometry service
  - Monitoring & output
- Easy to configure by customizing the algorithms, services and tools

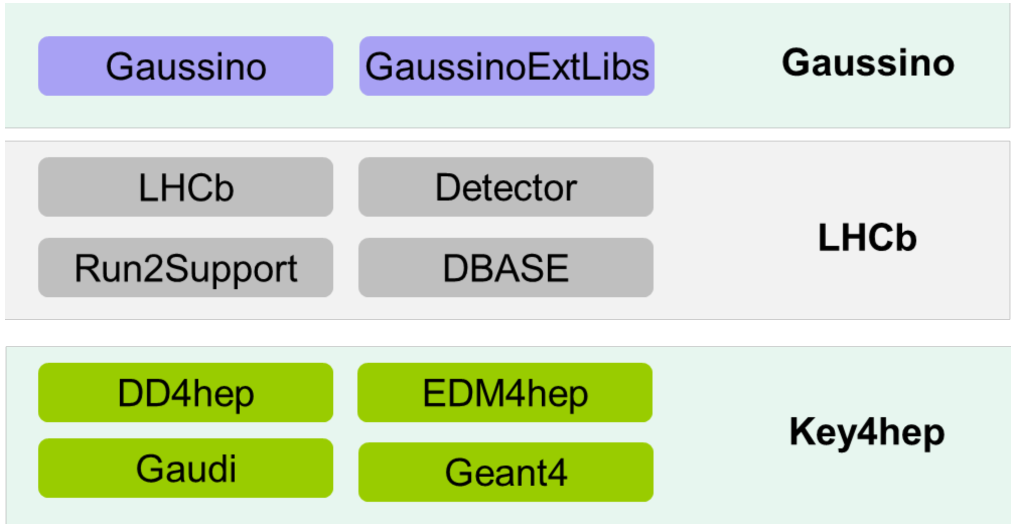


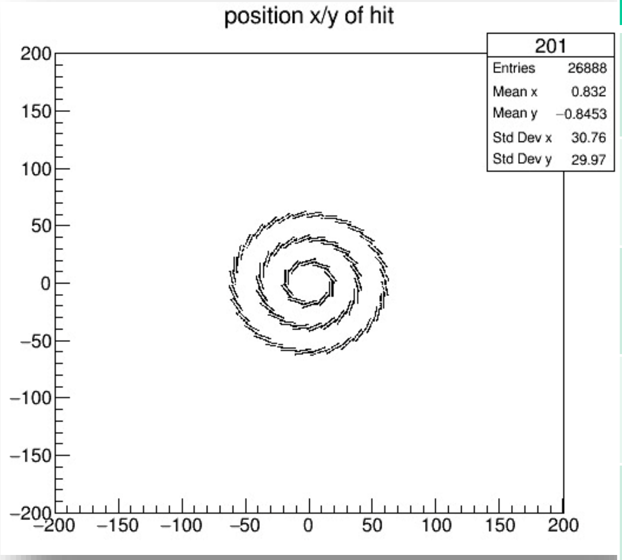
- Generation: `Generation` and `ParticleGun`
  - The input is **LHCb GenHeader**
  - The output is **HepMC GenEvent**
- Detector simulation: `GiGaAlg`
  - The input is **HepMC GenEvent**
  - The output is **G4Event** and **MC truths**

CEPC-on-Gaussino prototype

- Development of CEPC-on-Gaussino was planned with the following three steps
  - ✓ **Using the original version having the dependency on the LHCb software**
  - ✓ **Creating a modified version with less LHCb dependency**
  - ☐ Directly using the Key4hep version without LHCb dependency (not available at the moment).

- It consists of three parts
  - Adapt the Event Data Model from LHCb.
  - Use DD4hep as the geometry service.
  - Implement the Geant4 sensitive detector and hit objects.





Project	Git repo
LHCb	https://gitlab.cern.ch/talin/LHCb/-/tree/cepc-on-gaussino?ref_type=heads
gaussino extlibs	https://gitlab.cern.ch/talin/gaussinoextlibs/-/tree/cepc-on-gaussino?ref_type=heads
Gaussino	https://gitlab.cern.ch/talin/Gaussino/-/tree/cepc-on-gaussino?ref_type=heads
CEPCSW	https://gitlab.cern.ch/talin/CEPCSW/-/tree/cepc-on-gaussino?ref_type=heads
Installation script	https://gitlab.cern.ch/talin/build-cepc-on-gaussino

Summary

- The simulation framework in CEPCSW has been developed to support detector design and algorithm development.
  - The background mixing part still needs improvement.

- Regarding the R&D of simulation framework, we focus on the application of new technologies.
  - CEPC-on-Gaussino prototype works for the vertex detector.
  - Simulation of other sub-detectors will be added.

Thank you for your attention!