The Development and Prospects of the Offline Software Framework

Han Miao[1,2] Tianyu Xing[1,2] Jascha Grabowsky[3] Jinlin Fu[1]

1-University of Chinese Academy of Sciences

2-Institute of High Energy Physics

3-Bonn University

3rd workshop on electromagnetic dipole moments of unstable particles
Orsay, France
December 12th, 2023

Content

01

Introduction

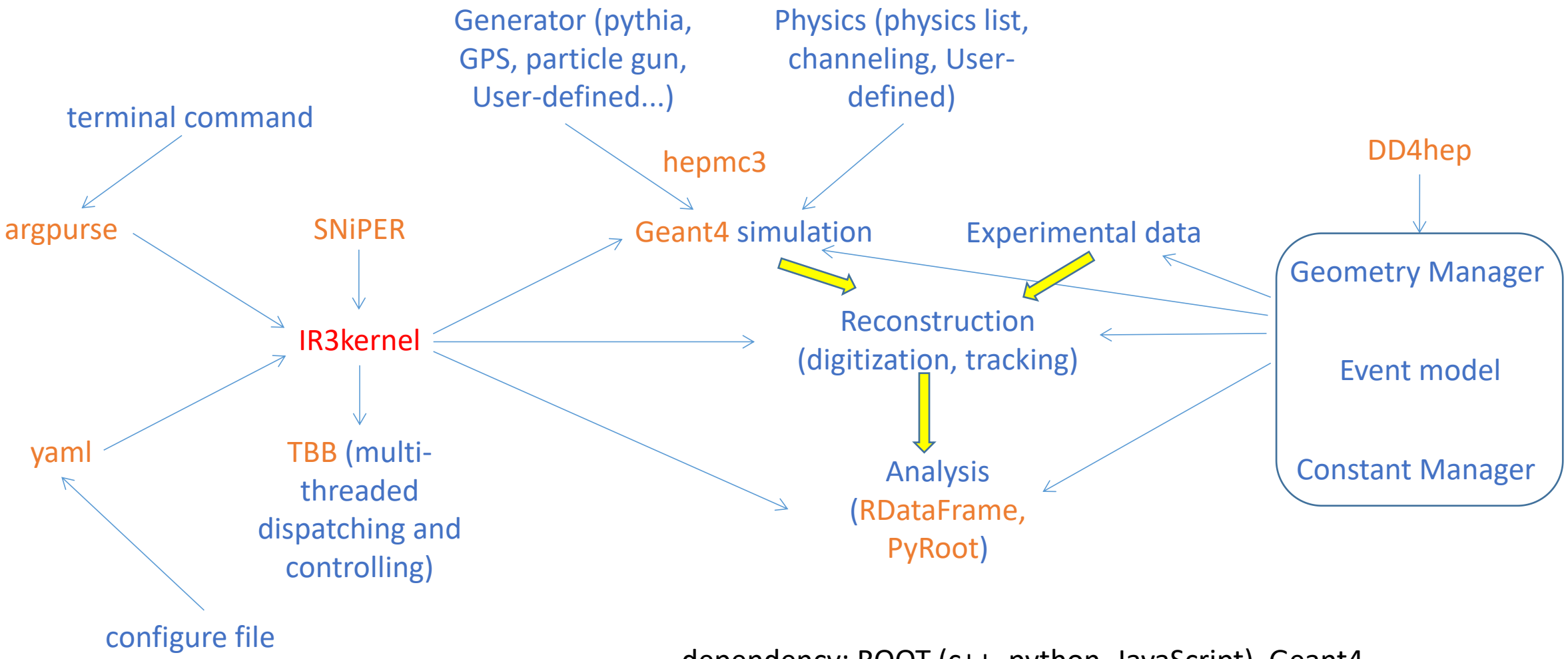
Introduction

- Offline software framework is essential and necessary
- A light-weighted, flexible and modern framework is what we need
  - **Light-weighted:** easy to deploy, less dependencies...
  - **Flexible:** easy to add modules for new features, flexible event model...
  - **Modern:** Fast running, parallel computing...
- IR3ana is designed, preliminarily realized and continuously developed

02

Overall Design

Structure

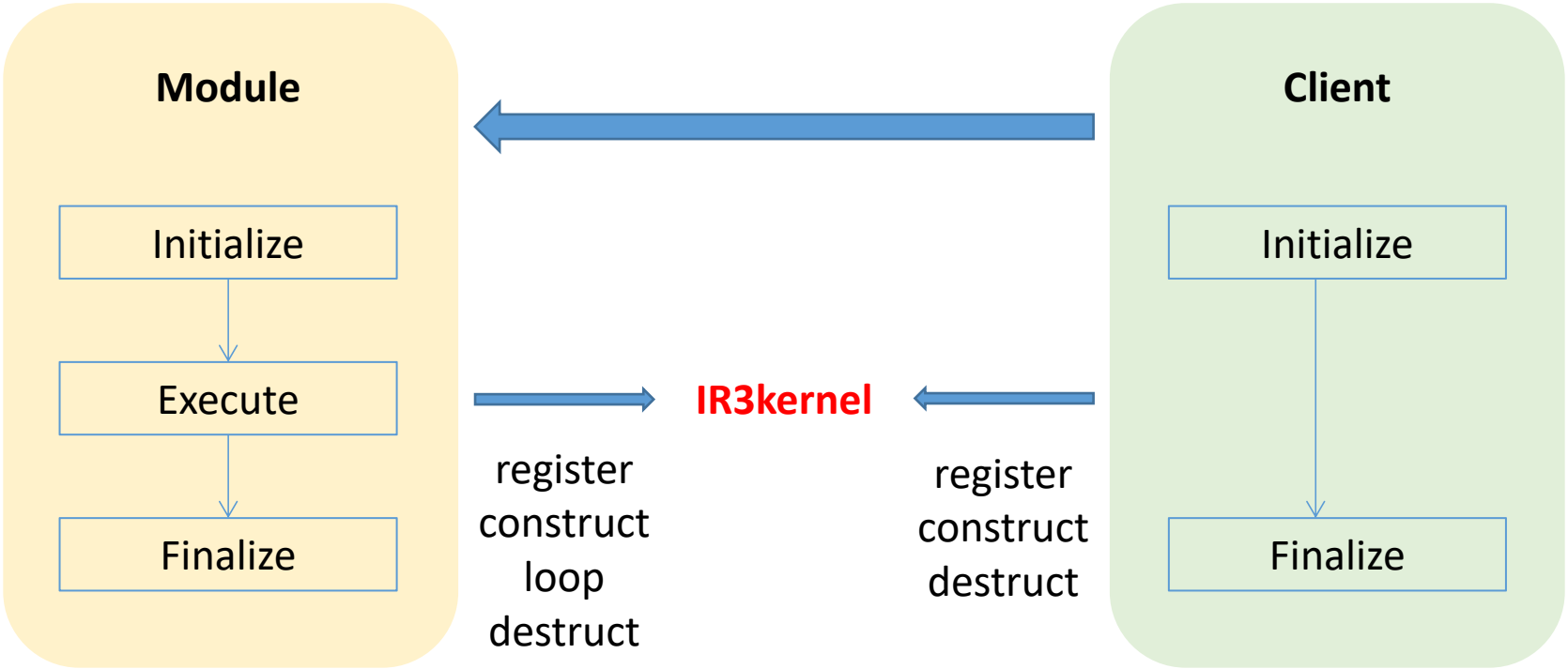


dependency: ROOT (c++, python, JavaScript), Geant4, DD4hep, PODIO, argpurse, yaml, SNiPER, Eigen3...

03

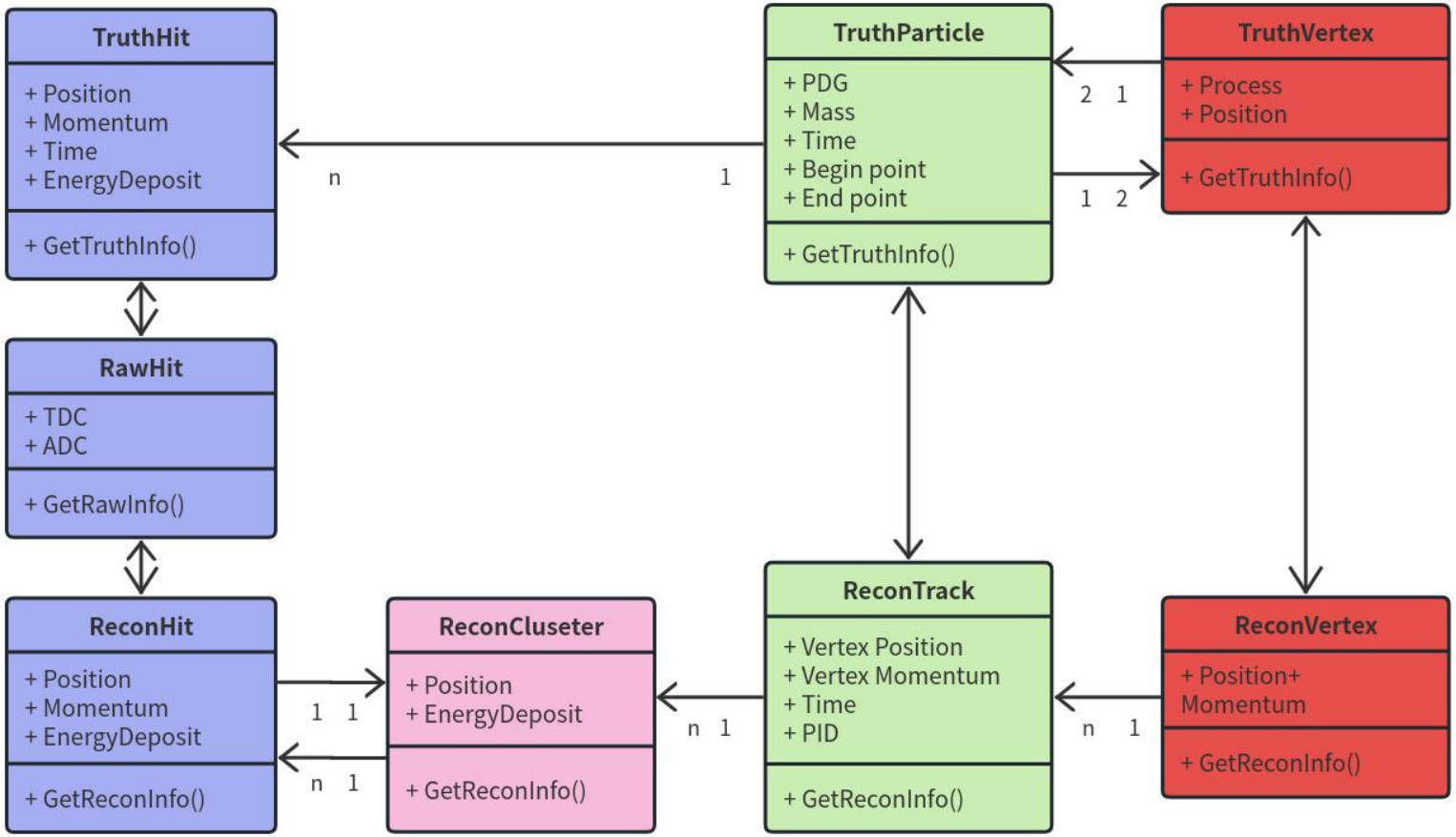
Sub-modules

Kernel



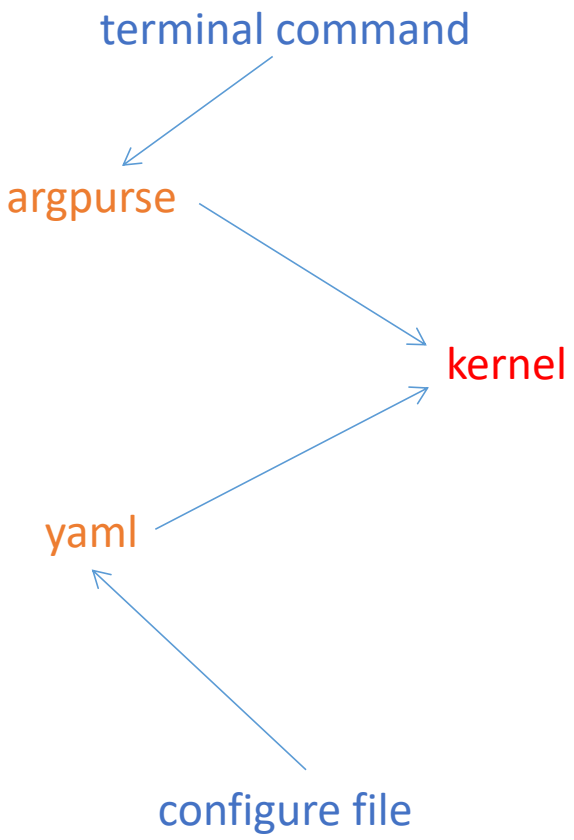
- The whole processing of data is composed of modules and clients
- Kernel will construct the only instance for the registered modules and clients
- Modules will be looped
- Relevant modules will be combined into packages to handle the sub-loop of several special modules (calibration, alignment and reconstrction) (in progress)
- Parallel computing is under consideration (in progress)

Event Model



- Event model is generated using PODIO via a yaml file
- 3 levels: truth, raw, reconstruction
- 4 types: hit, cluster, track, vertex
- Connections are established between different levels of the same type and different types of the same level

Parameter Grasp



- Two types of parameters: from command and from configure file
- Parameters from command is handled using **argpurse** package: cannot be add, remove or modified by users (verbose level, number of threads …)
- Parameters from configure file is handled using **yaml** file by **IR3ParaManager** (parameters for the modules and clients …)

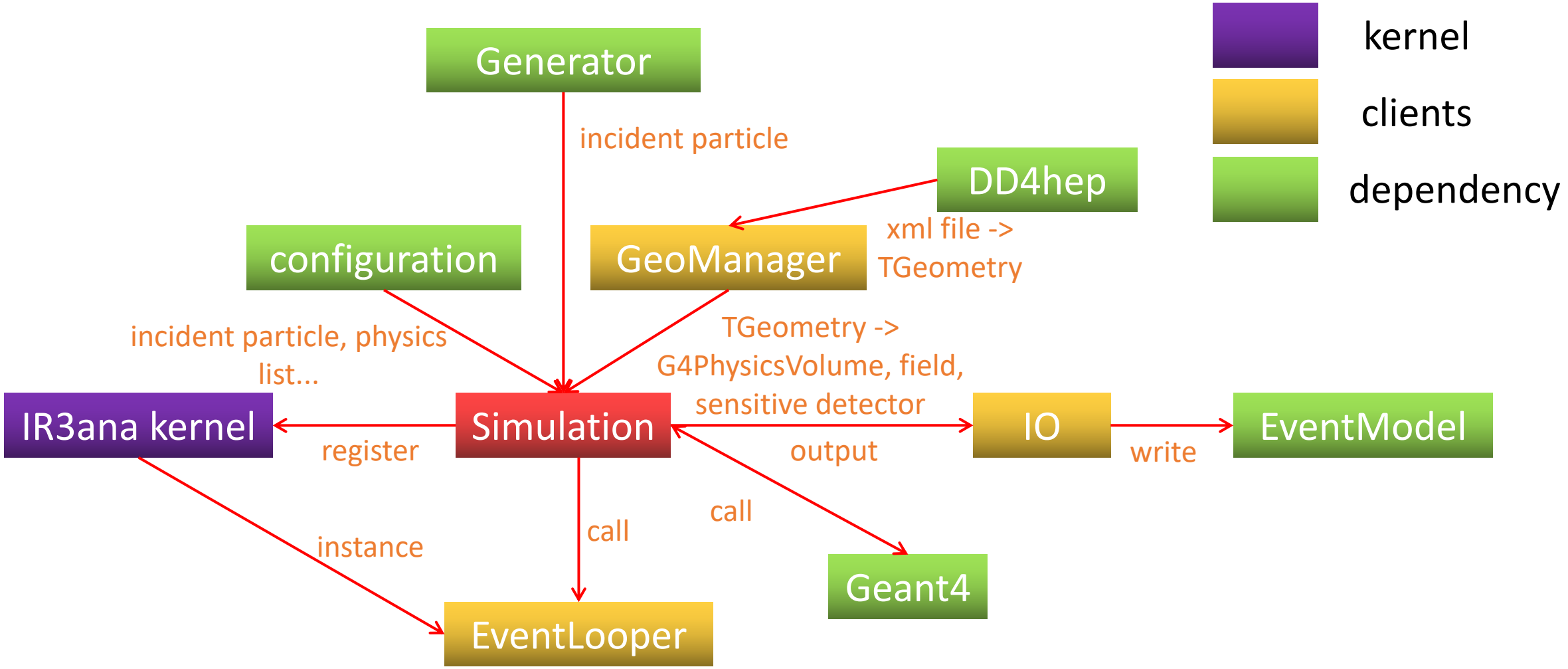
Geometry Manager

- Geometry is described in `xml` files and read by `DD4hep`
- Can also be read from or write into a `ROOT` file by IR3GeoManager
- Provide all the necessary information about the geometry used in simulation, reconstruction, analysis ...
- A toolkit to manage the versions of geometry is being planned and will be implemented in future (in progress)
- Provide the magnetic field in the whole space (uniform, map) (in progress)

Constant Manager

- Essential constants are read from files: calibration, alignment ...
- Modules and clients can get the needed constant from the constant manager (in progress)

Simulation

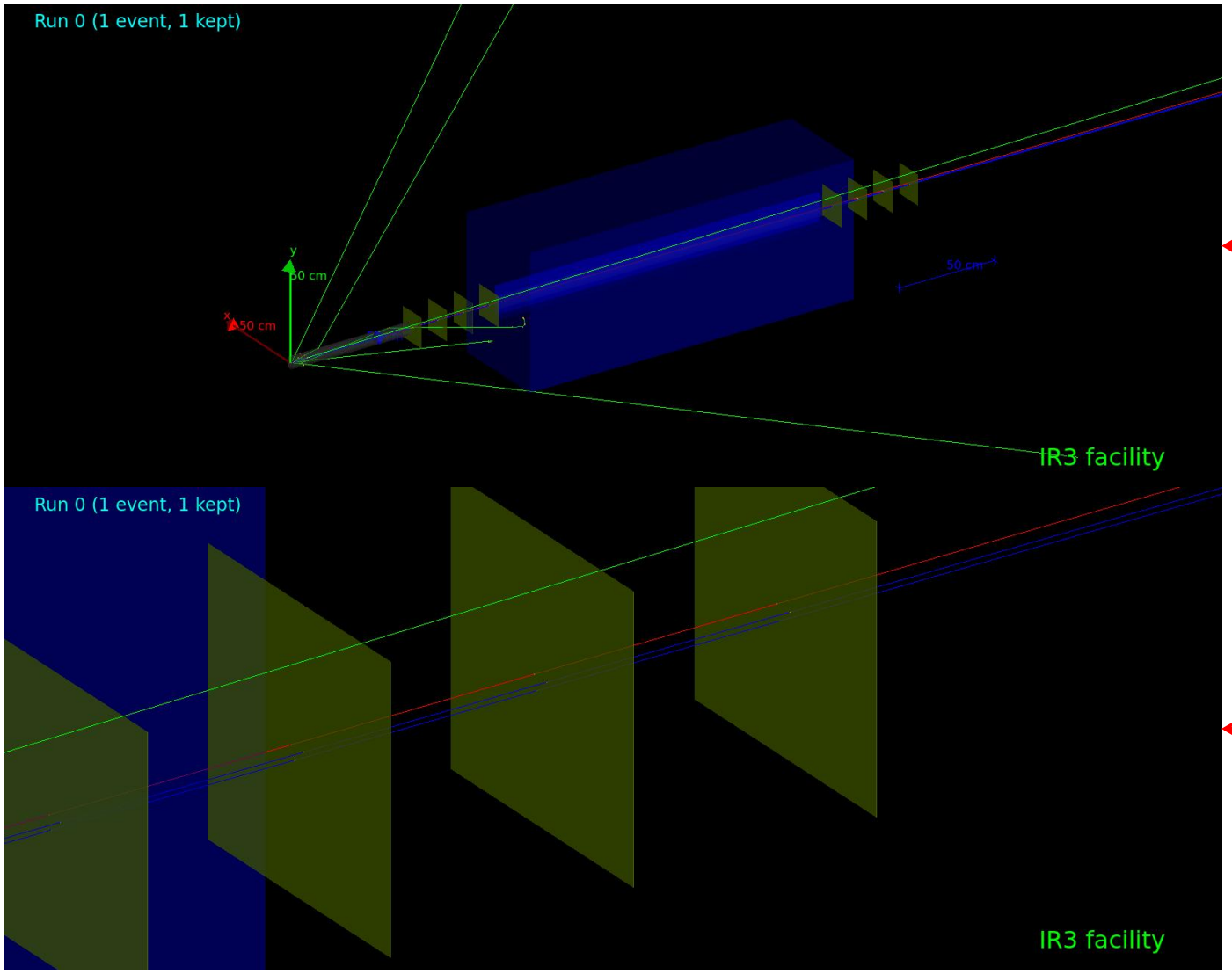

kernel
clients
dependency
Generator
incident particle
DD4hep
xml file ->
TGeometry
configuration
GeoManager
incident particle, physics
list...
TGeometry ->
G4PhysicsVolume, field,
sensitive detector
IR3ana kernel
register
Simulation
output
IO
write
EventModel
instance
call
call
Geant4
EventLooper

Simulation

- Generator:
  - Particle Gun
  - G4GeneralParticleSource
  - Pythia (in progress)
  - HepMC3 (in progress)
  - User-defined spectra
  - …

- Process:
  - Geant4 physics list
  - Channeling (in progress)
  - photoproduction (in progress)
  - …

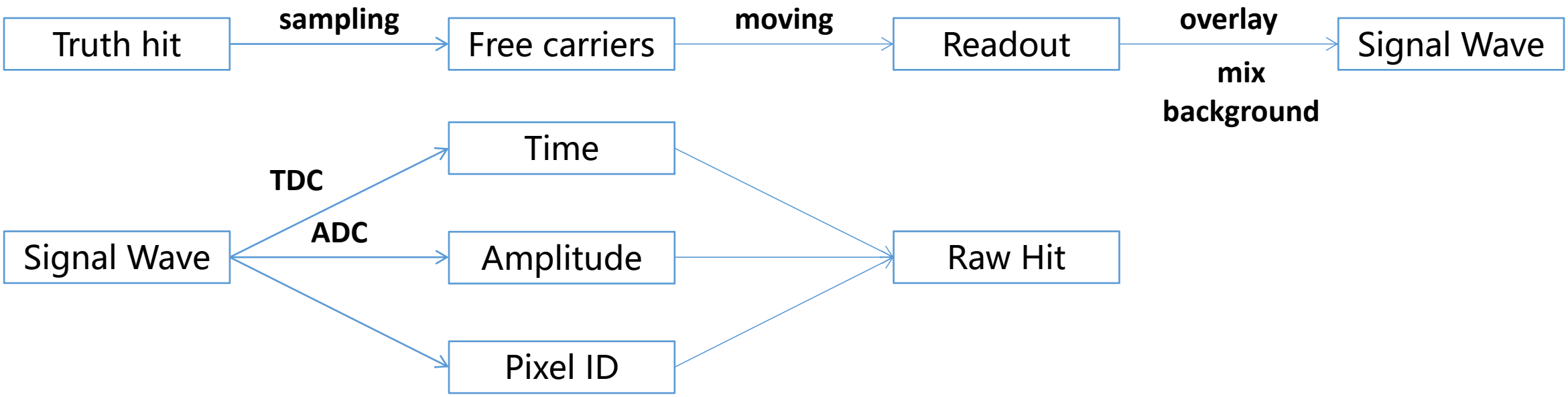
Simulation



Geometry is imported from DD4hep

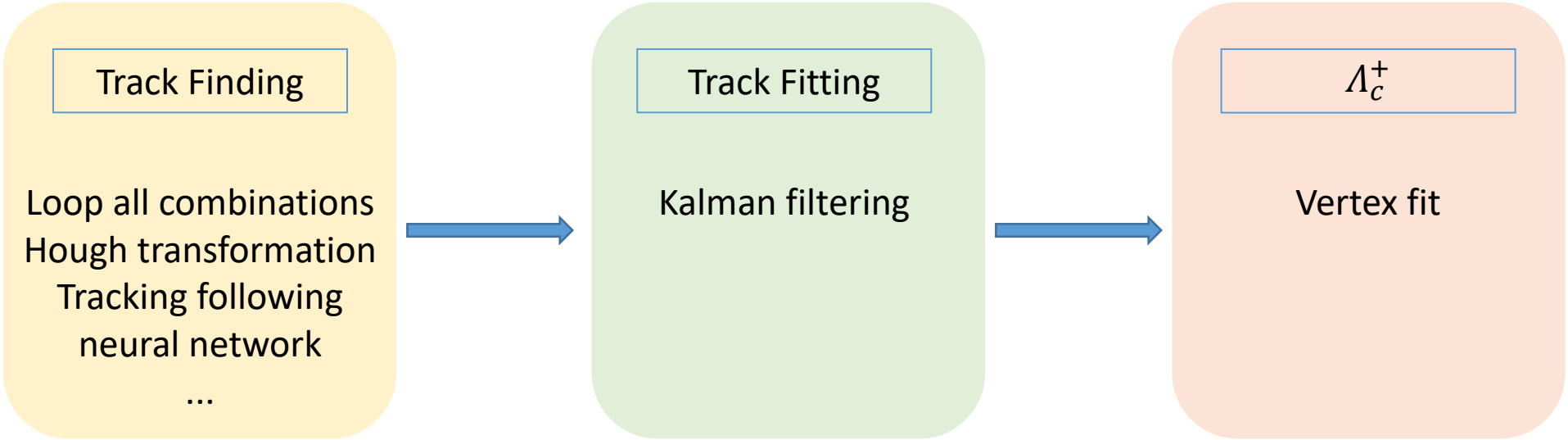
Clear trajectories of three charged particles from Λ_c^+

Digitization



- Get number, energy, creating time and position of carriers (electron-hole pair) by sampling typical distributions
- Calculate the readout waveshapes of carriers
- Combine the waveshapes of carriers to get the total signal shape

Reconstruction



- A specialized recontruction module has been developed by Jascha Grabowsky
- General tracking modules are being considered (in progress)
- Tracking finding module based on neural network will be implemented (in progress)
- The reconstruction of RICH and muon chamber will be implemented (in progress)

Calibration & Alignment

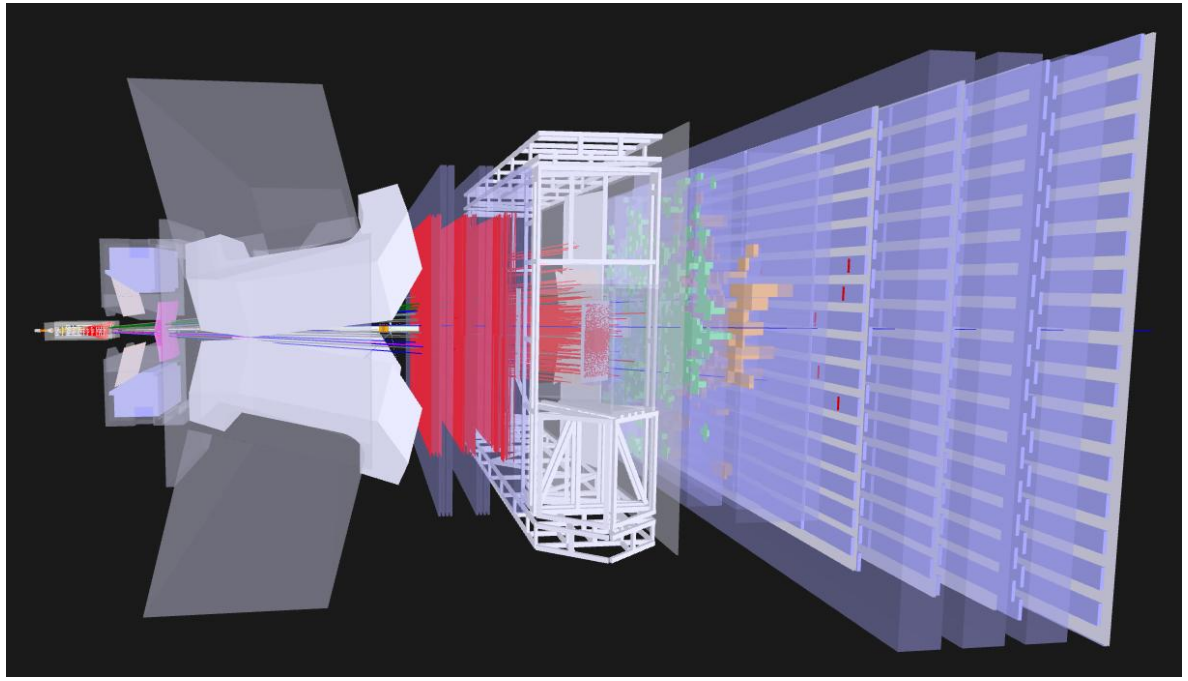
- Something not so urgent but essential and necessary for the smoothing running of the experiment
- Iterations with reconstruction procedures
- Output calibration and alignment parameters will be written in files and managed by the constant manager

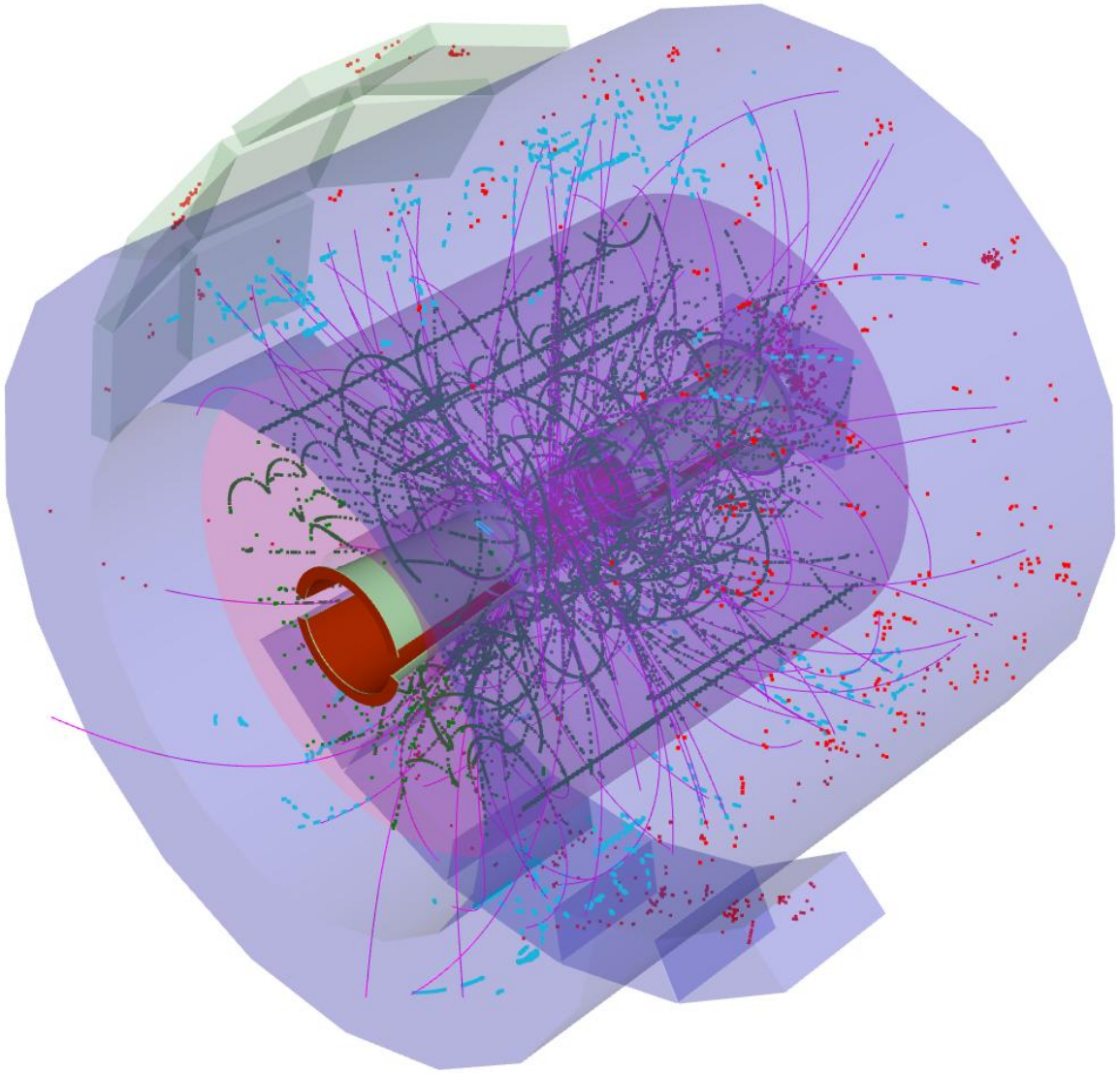
Particle Identification

- A RICH detector is being considered as the possible method for PID
- The feasibility is being studied
- A preliminary PID algorithm has been implemented
- More complete packages will be developed

Event Display

- 4 possible techniques are begin considered:
  - ✓ ROOT browser
  - ✓ JavascriptROOT: https://root.cern.ch/js/
  - ✓ Unity
  - ✓ Pheonix: https://github.com/HSF/phoenix





04

Summary and Next Work

Summary and Next Work

- The preliminary version of IR3ana has been developed and used with incorperating necessary features
- Totally 5 versions has been released
- Far from complete and need more contributions from anyone who are interested
- A usable framework will be provided before the IR3 test and future experiment

Thank you !

Han Miao[1,2] Tianyu Xing[1,2] Jascha Grabowsky[3] Jinlin Fu[1]

1-University of Chinese Academy of Sciences

2-Institute of High Energy Physics

3-Bonn University