Applications of SNiPER Software Framework

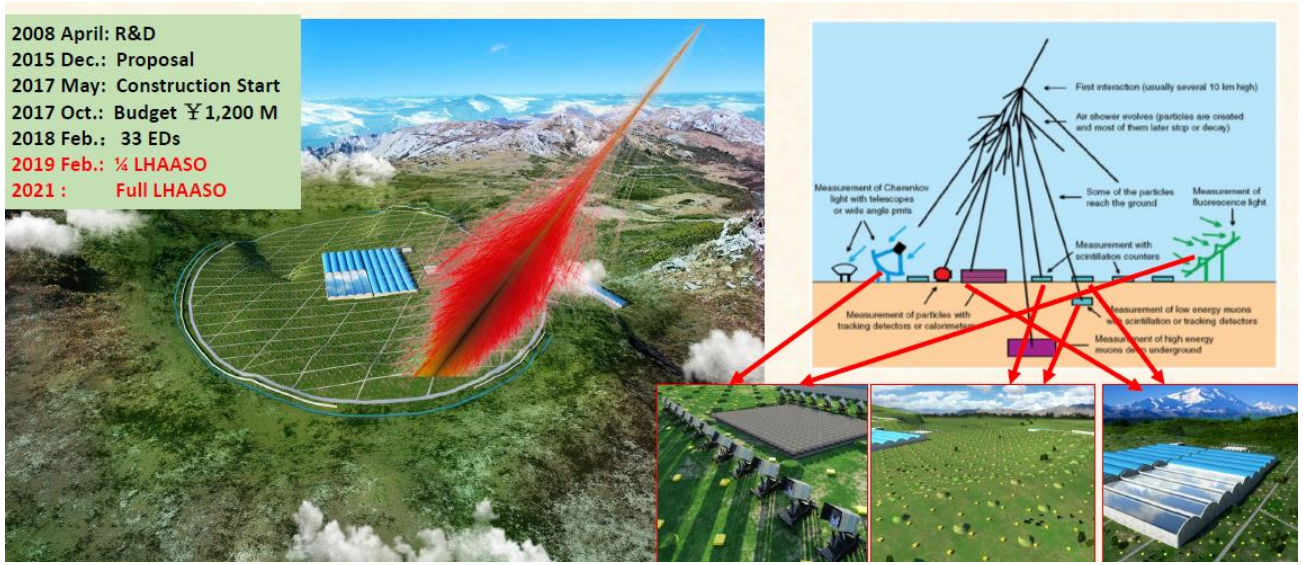
Wenhao Huang, Xingtao Huang
For LHAASO/STCF Software Team
Shandong University

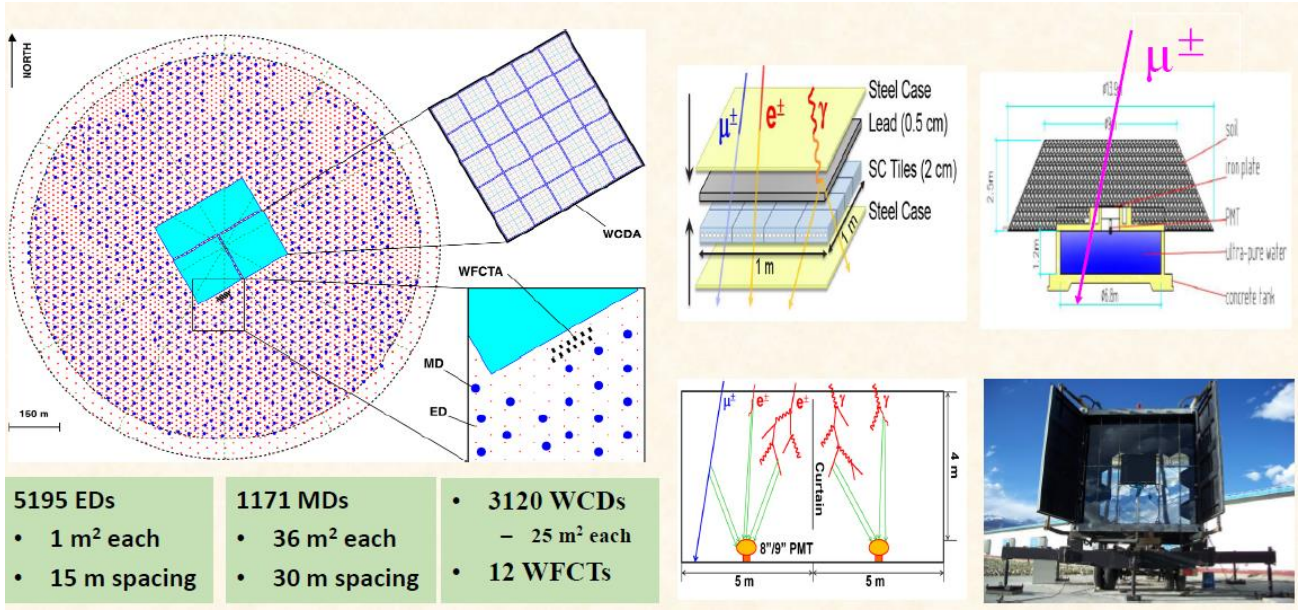
30, May, 2019
Nanjing

Outline

- Requirements
- Architecture Design
- SNiPER Framework
- Current Status
- Optimized Event Data Model
- Detector Geometry Description
- Experiment-specified function
- Summary

THE LARGE HIGH ALTITUDE AIR SHOWER OBSERVATORY

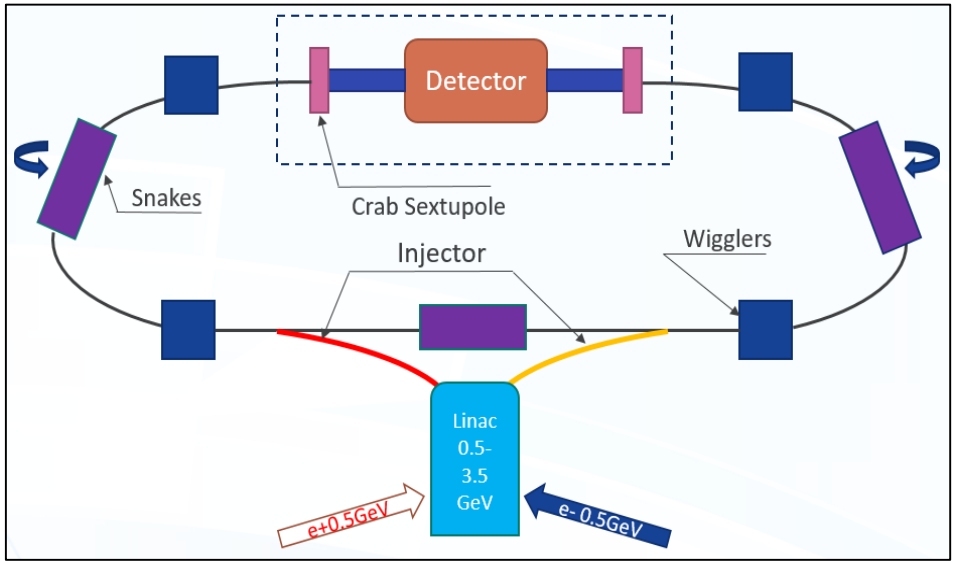




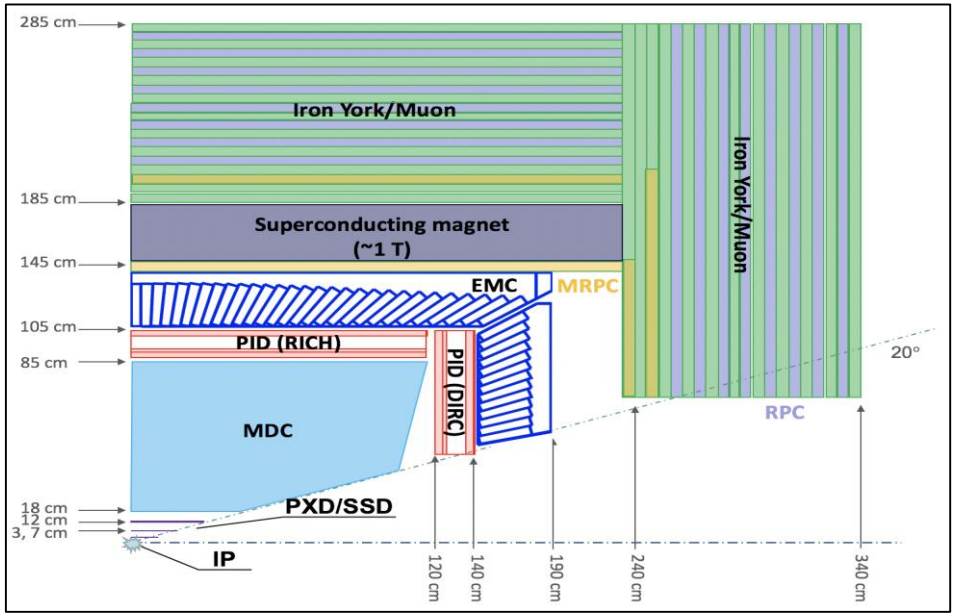
- Detector
  - KM2A
  - WCDA
  - WFCTA
- Event size: larger than Argo

- Separated event
- Hybrid event

Super Tau-Charm Facility



- e+ e- Collider
- E_{cm} = 2-7 GeV
- L = 0.5-1×10^{35} $cm^{-2}s^{-1}$ @4 GeV
- Event Rates: ~100 X BESIII



- General purpose Detector
  - PXD
  - MDC
  - PID
  - EMC
  - MUD
- Event size: larger than BESIII

Requirements of LHAASO/STCF offline software

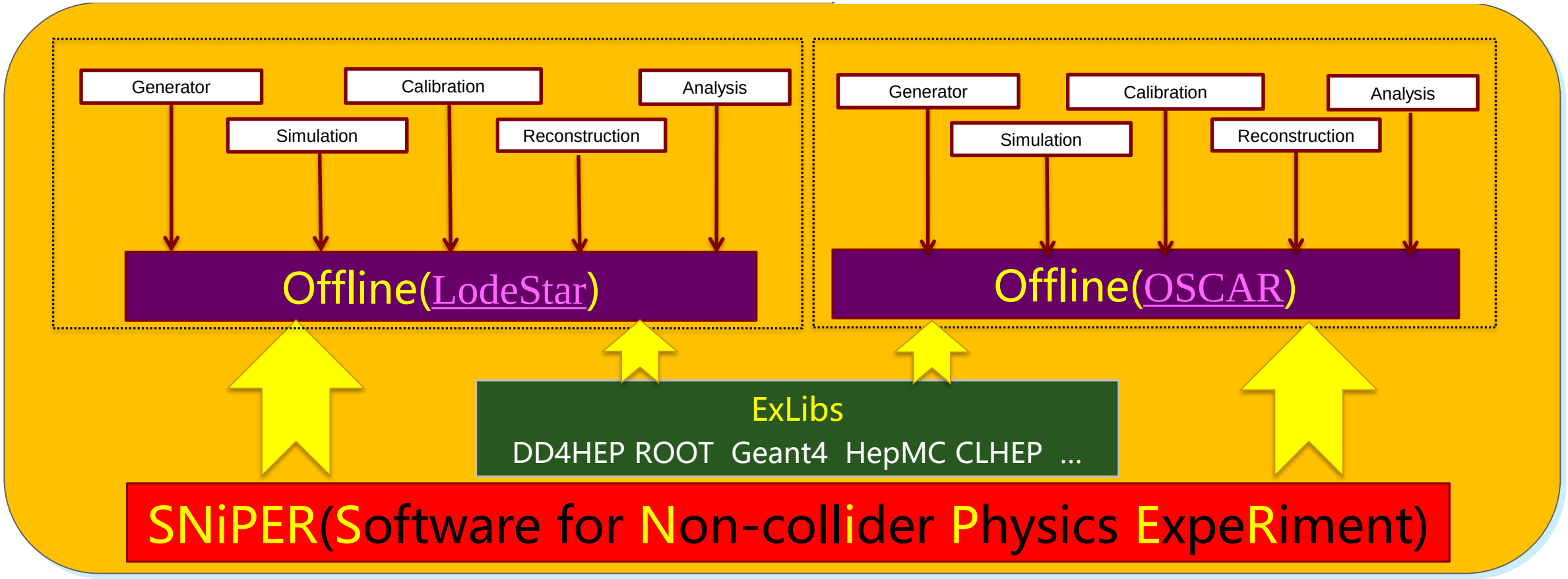
- Procedure and requirements of LHAASO are similar with Argo
- Very large event size and data volume
  - ⇨ Several PB per year
  - ⇨ Millions of sub particles in one event
- Hybrid/separated event analysis requirements
- Need faster and more powerful offline software system.

- Procedure and requirements of STCF are similar with BESIII
- Many of BESIII software could be reused in STCF
- But event rates and event size much larger than BESIII
- Higher Data IO performance required
- Need faster and more powerful offline software system than BOSS, The BESIII Offline Software System.

Architecture Design

LodeStar: LHAASO Offline Data Processing Software Framwork

OSCAR: Offline Software of Super Tau-Charm Facility

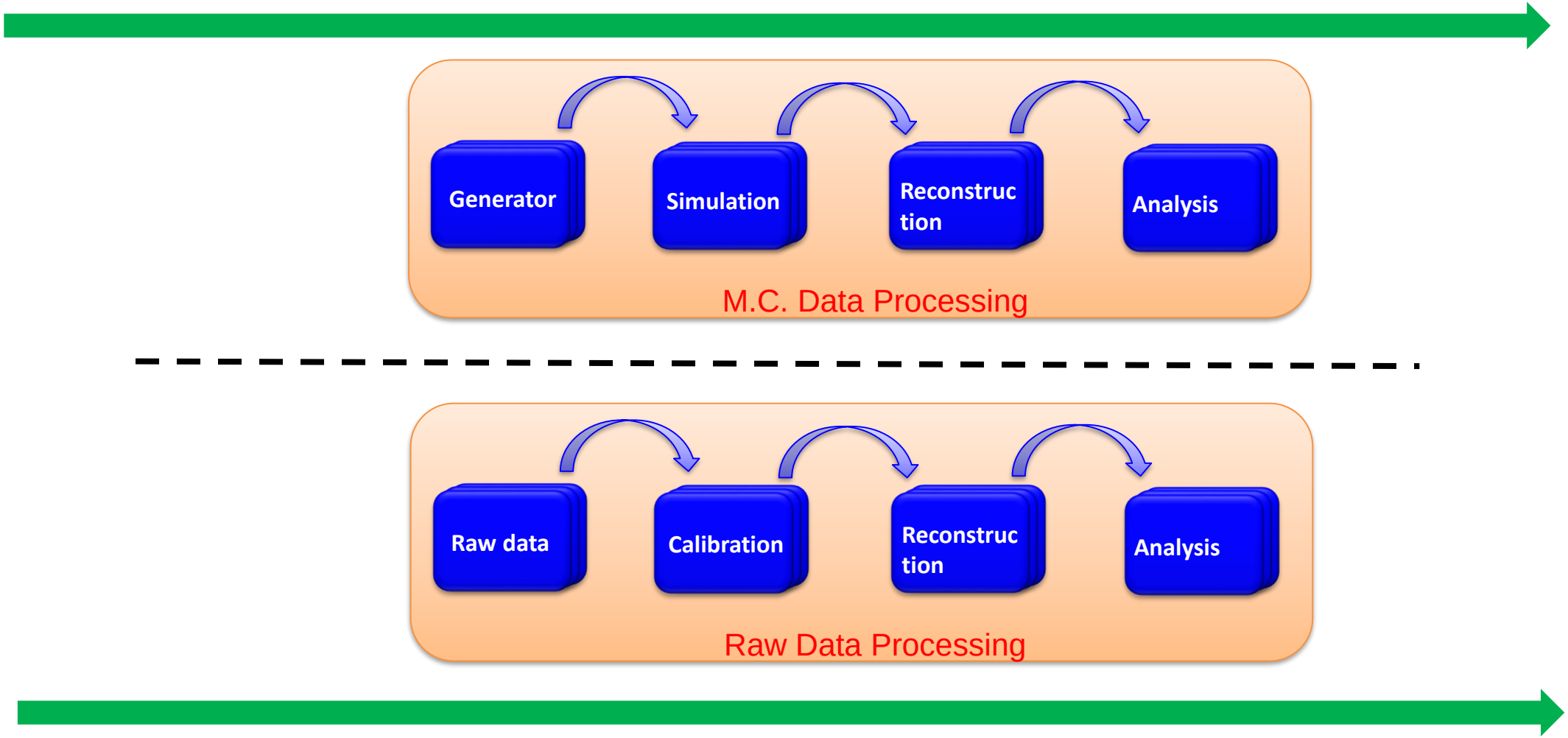


- ExLibs: include frequently used third-party software and tools
- SNiPER: a new framework to provide core functionalities and common services.
- Offline: all software specific to STCF/LHAASO, including Generator, Simulation, Calibration, Reconstruction and Analysis.

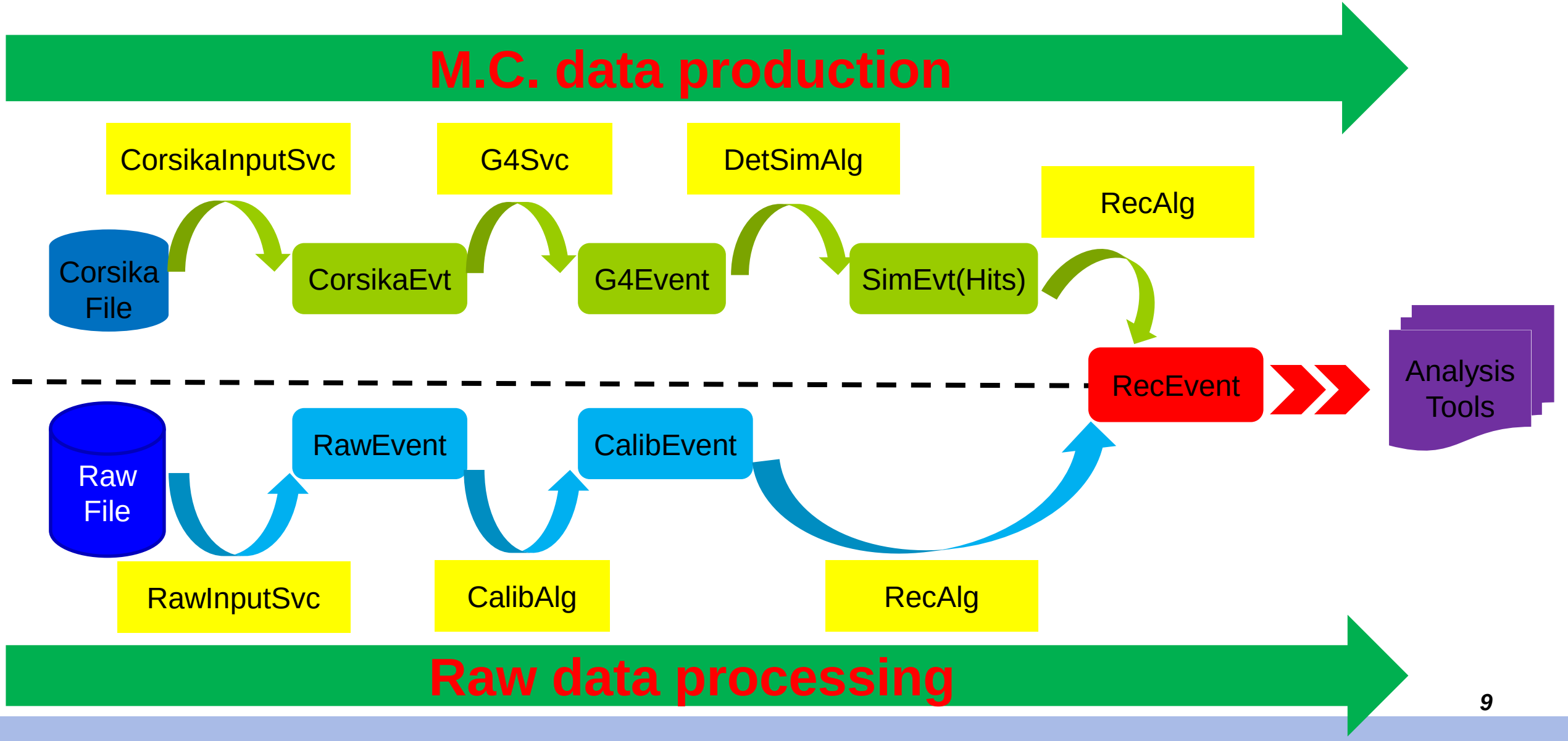
SNiPER Framework

- Developed by JUNO Collaboration since 2013
- Feature
  - ⇨ lightweight framework, less dependence on third-party software
  - ⇨ Flexible and fast execution
  - ⇨ Easy to learn and convenient to use
  - ⇨ Support multithreading
  - ⇨ Learn a lot from other software frameworks, such as Gaudi
  - ⇨ Based on the valuable experiences of DayaBay and BESIII
  - ⇨ Coding from scratch
- Current Status
  - ⇨ Successfully used for JUNO Experiments
  - ⇨ Tested with BESIII Real Data Analysis
  - ⇨ Used in LHAASO and STCF Experiments
  - ⇨ Under investigation by several other experiments, sucha as nXEO, CEPC(optional)…

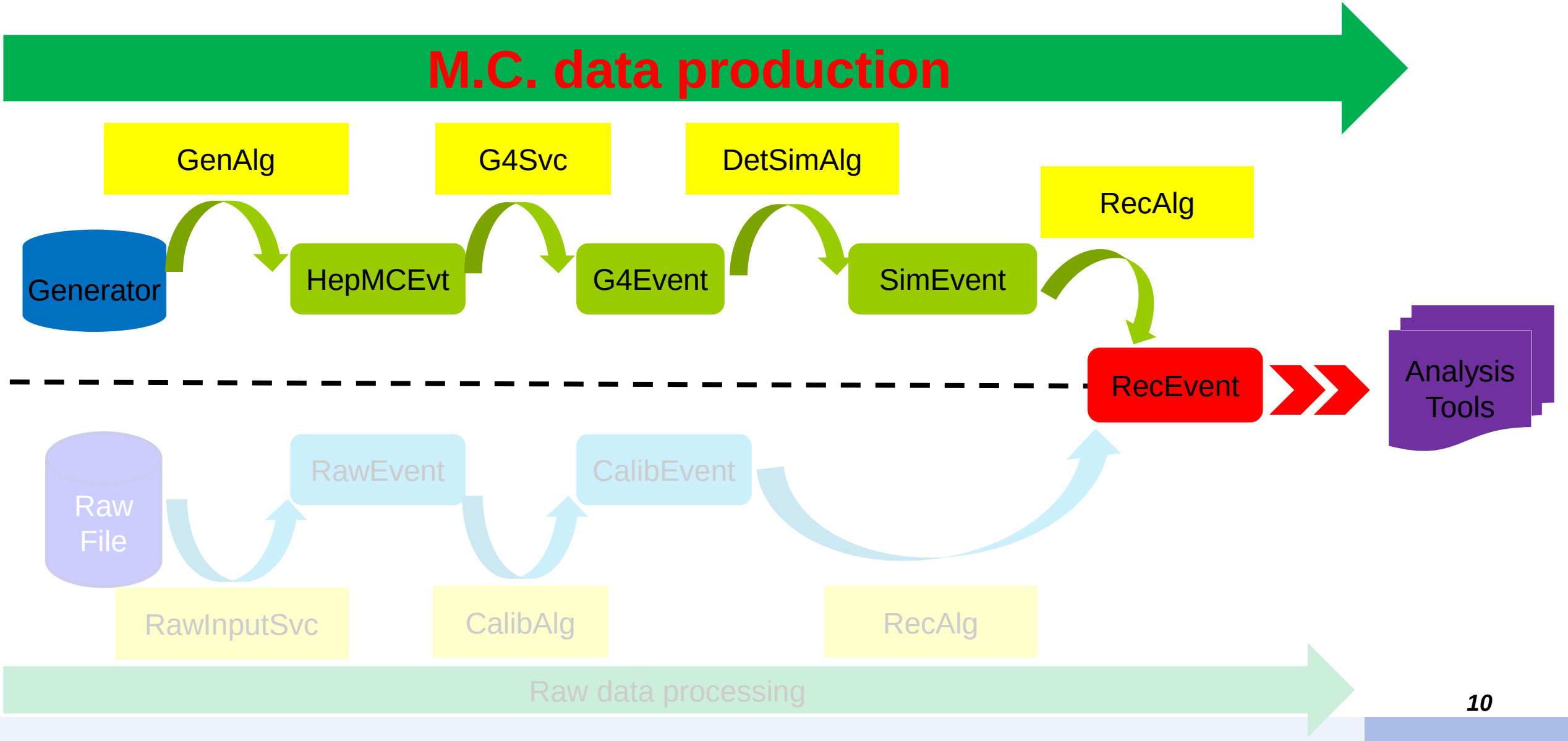
Data Flows in hep experiments



Two Data Flows in Framework(LHAASO)



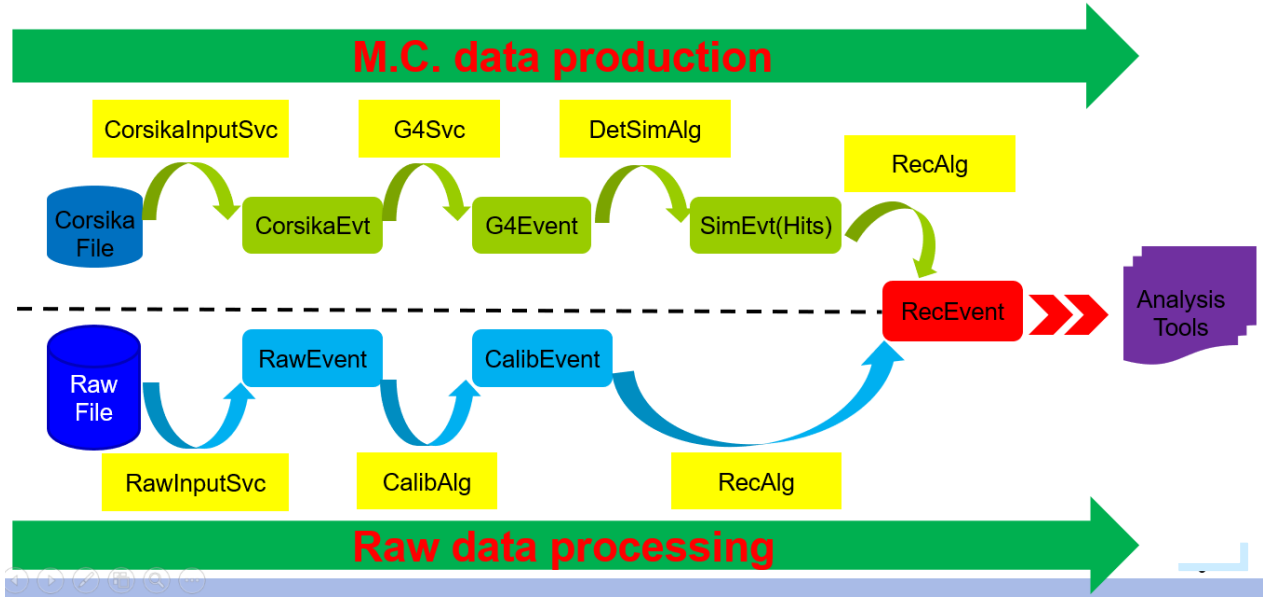
M.C. Data Flows in Framework(STCF)



Current status

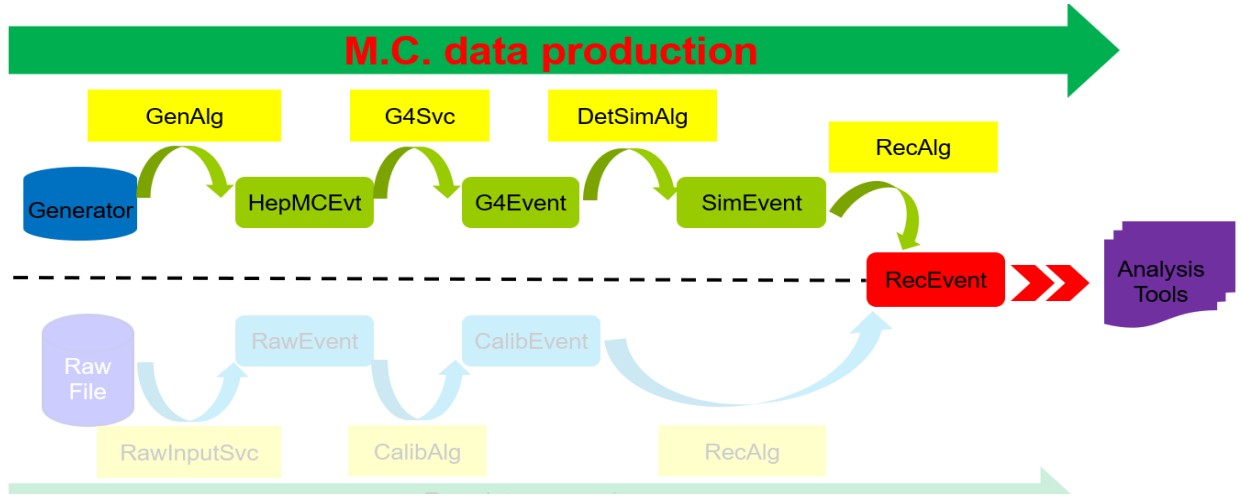
For LHAASO

- M.C. data
  - ⇨ CorsikaInput system
  - ⇨ Simulation and reconstruction
- Raw data
  - ⇨ RawInput system
  - ⇨ Calibration algorithm is under developing



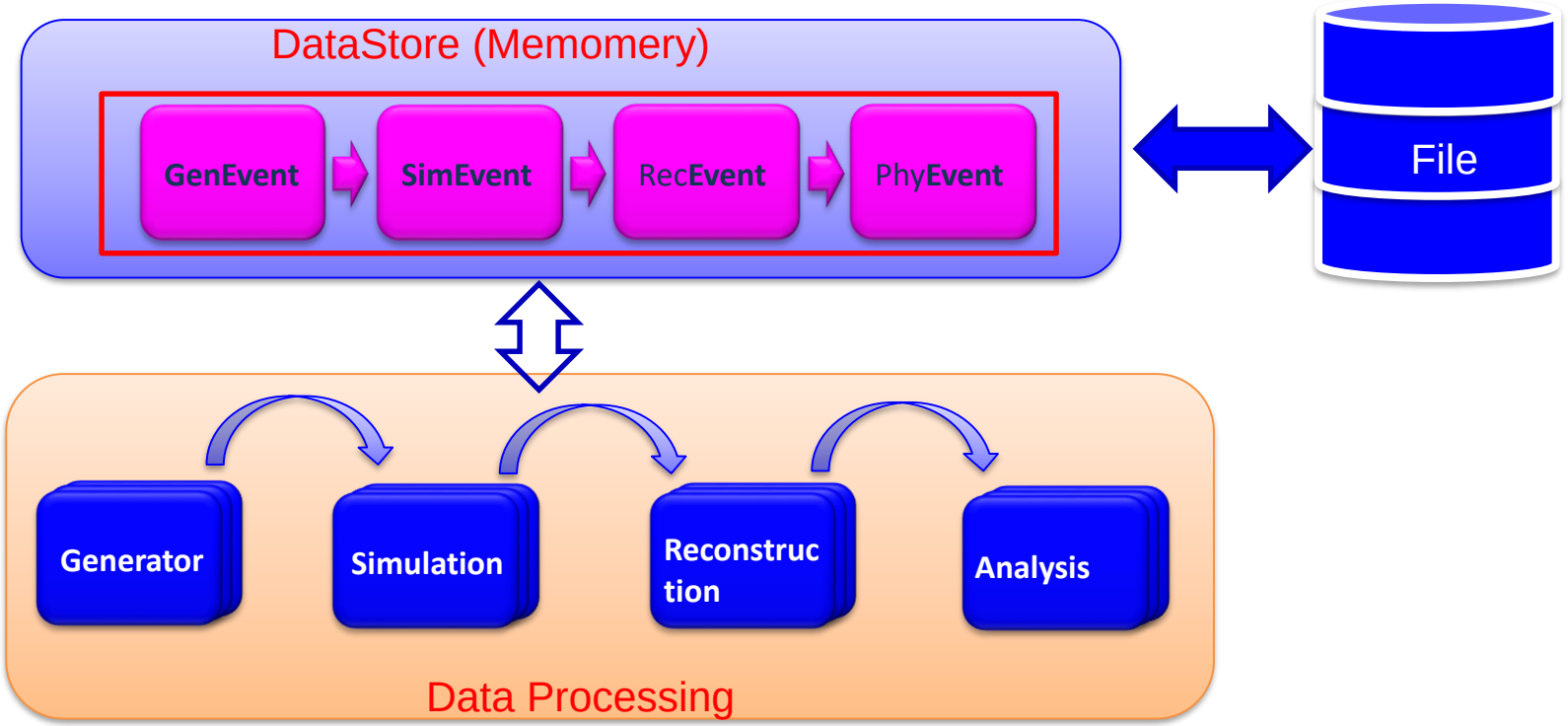
For STCF

- M.C. data
  - ⇨ Integrated generators from BESIII
    - Babayaga, KKMC, phokara
  - ⇨ Simulation
  - ⇨ Reconstruction is under developing

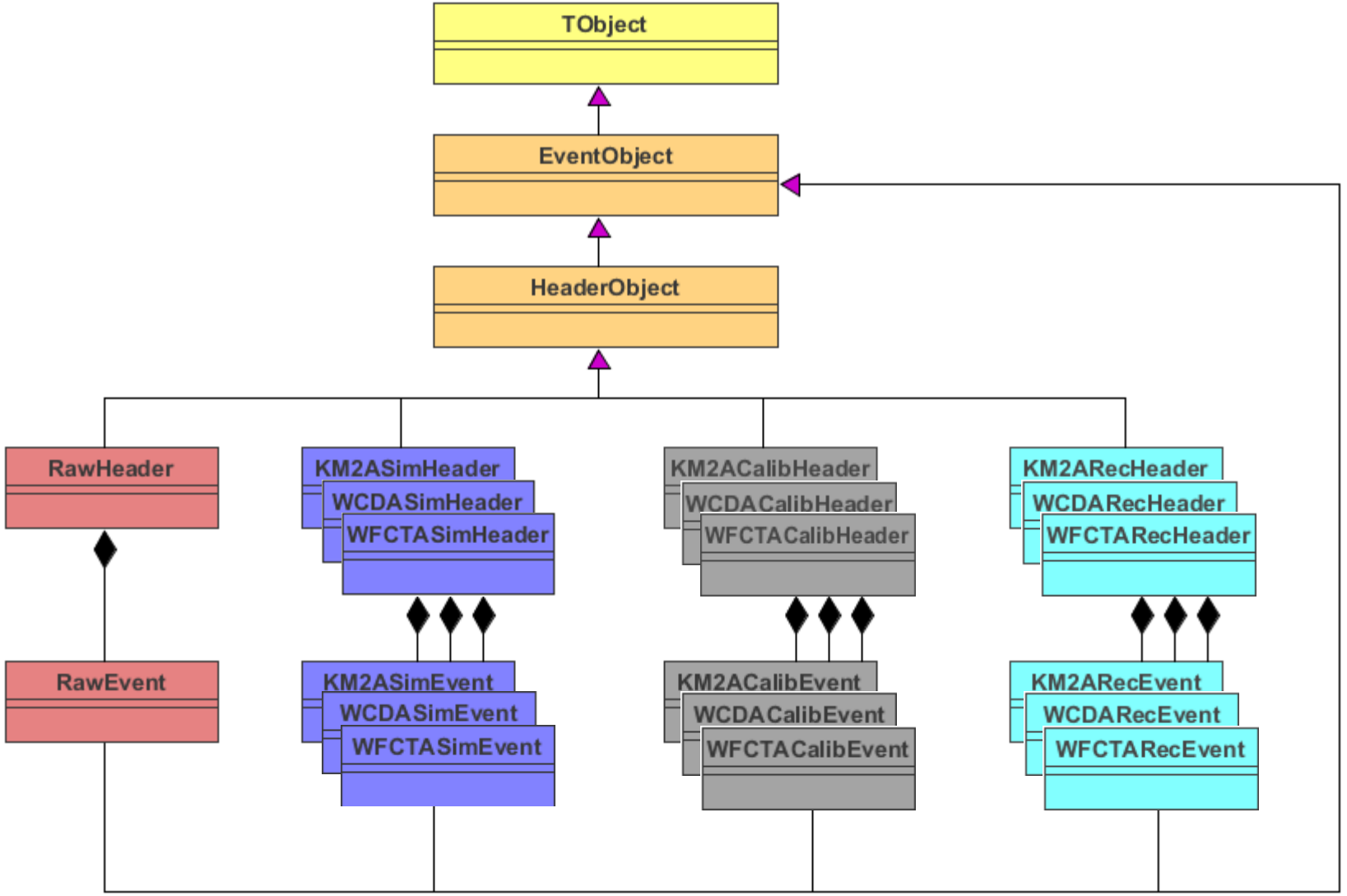


Event Data Model

- Defines the data unit to be processed in offline data processing
- In form of C++ classes
- Can be converted into persistent type and saved into disk

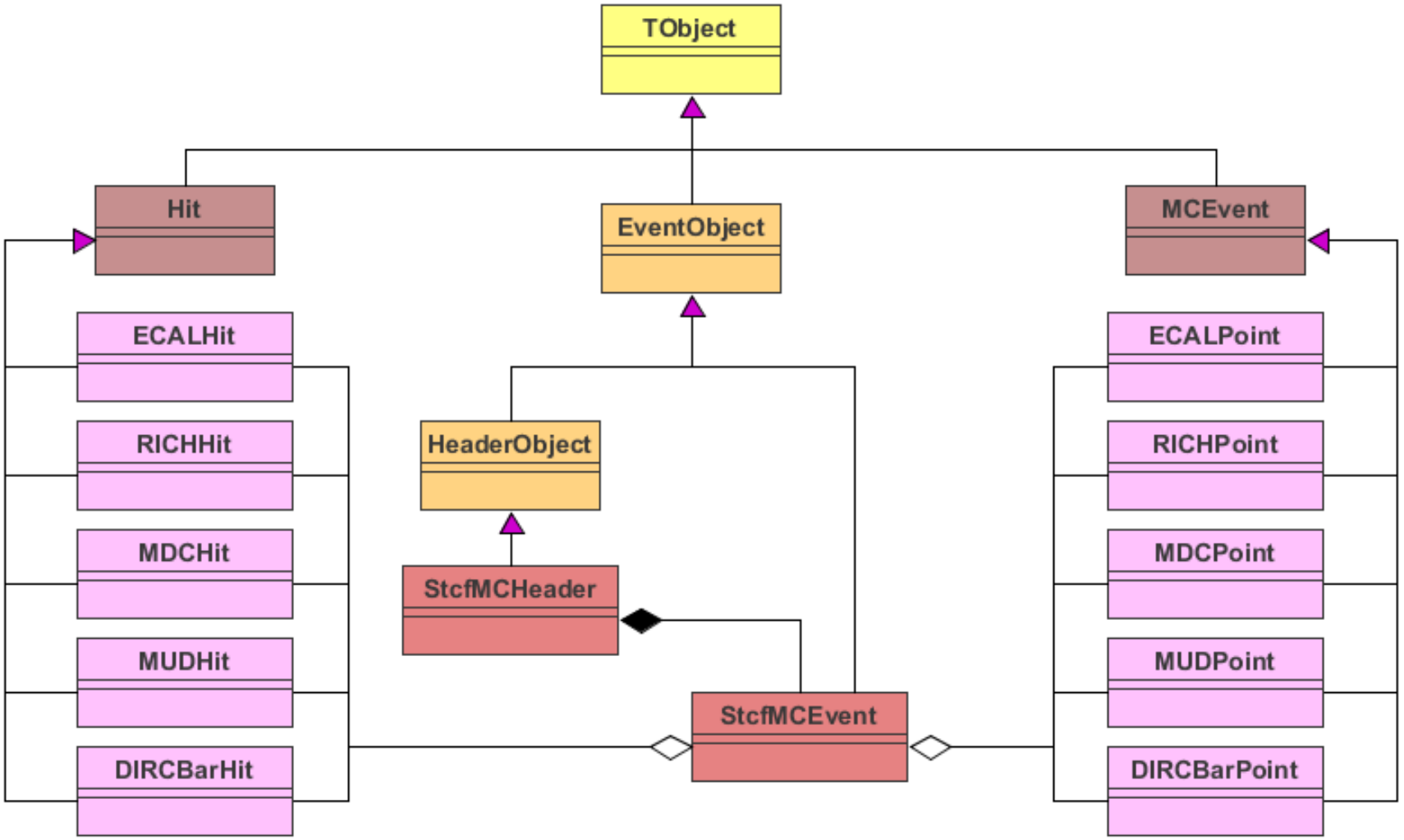


Optimized Event Data Model(LHAASO)



- Each process defines their Event Objects
- Each subdetector defines their Event Objects
- All EventObject inherits from TObject.

Optimized Event Data Model(STCF)



- Each subdetector defines their Event Objects
- All EventObject inherits from TObject.

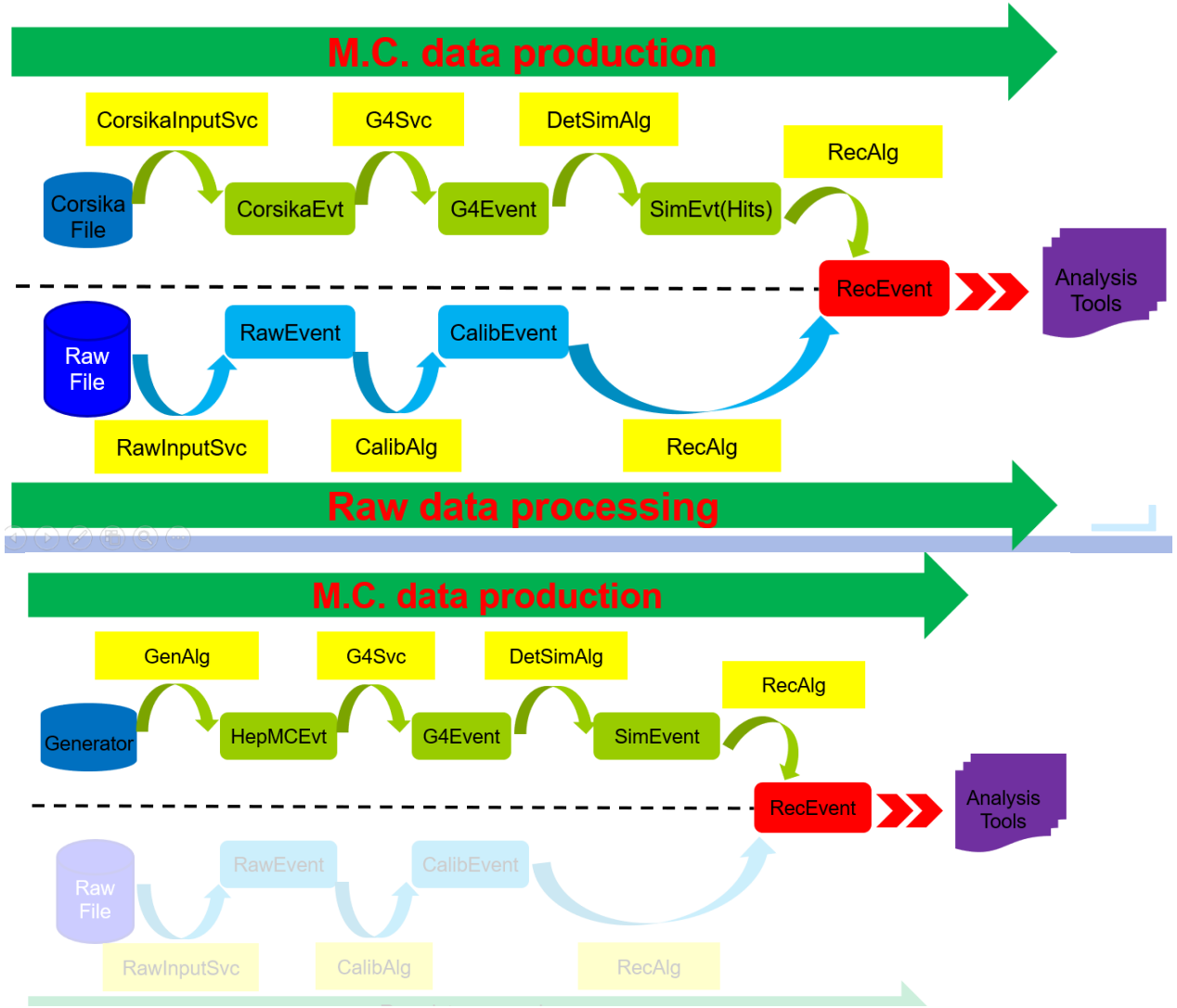
Data Input/Output System

For Corsika file and raw file in disk

- Two types of IO Systems
  - CorsikaInput system
  - RawInput system

For EDM inherit from EventObject

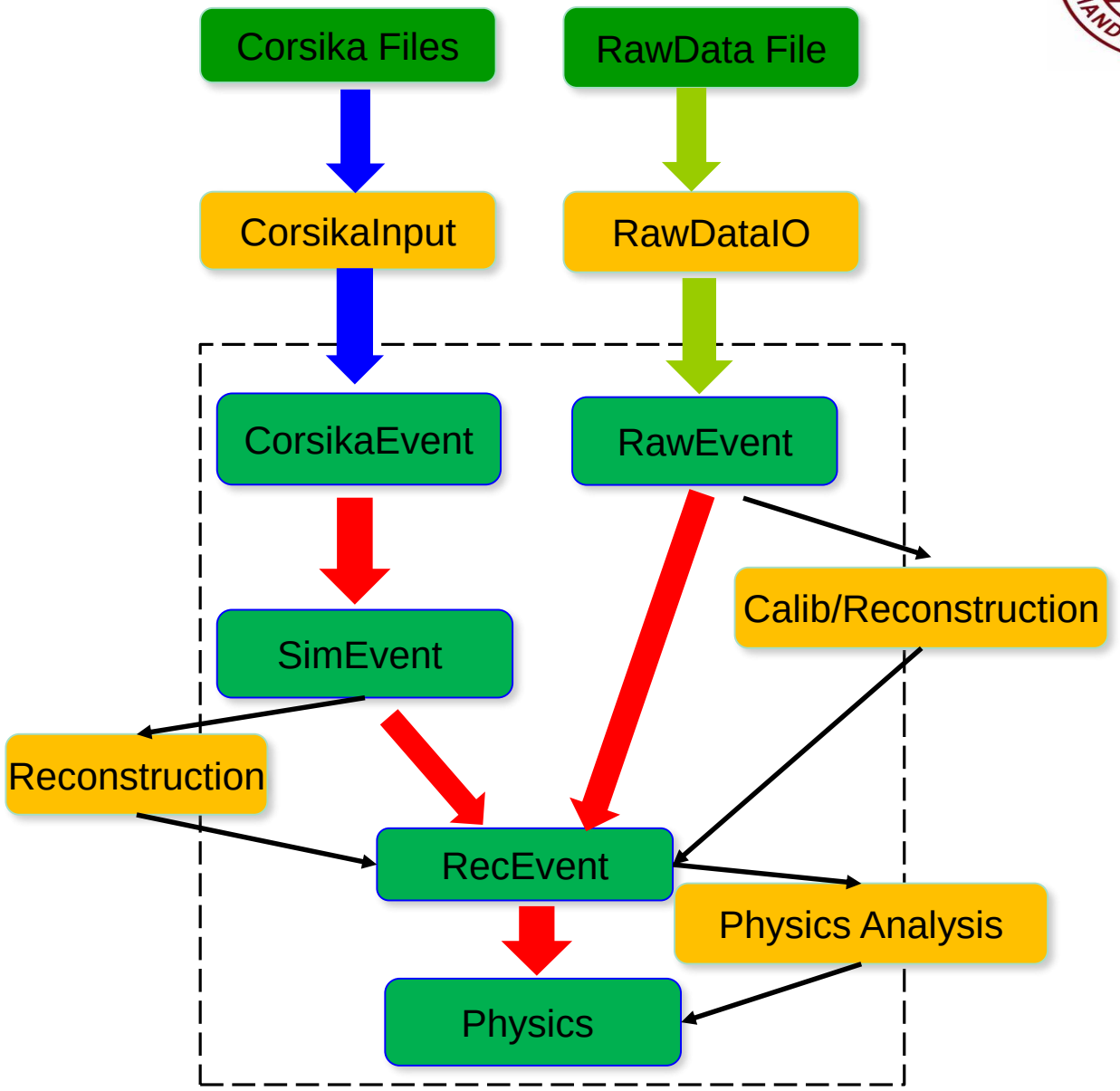
- Unified RootIO system



Data Input/Output System

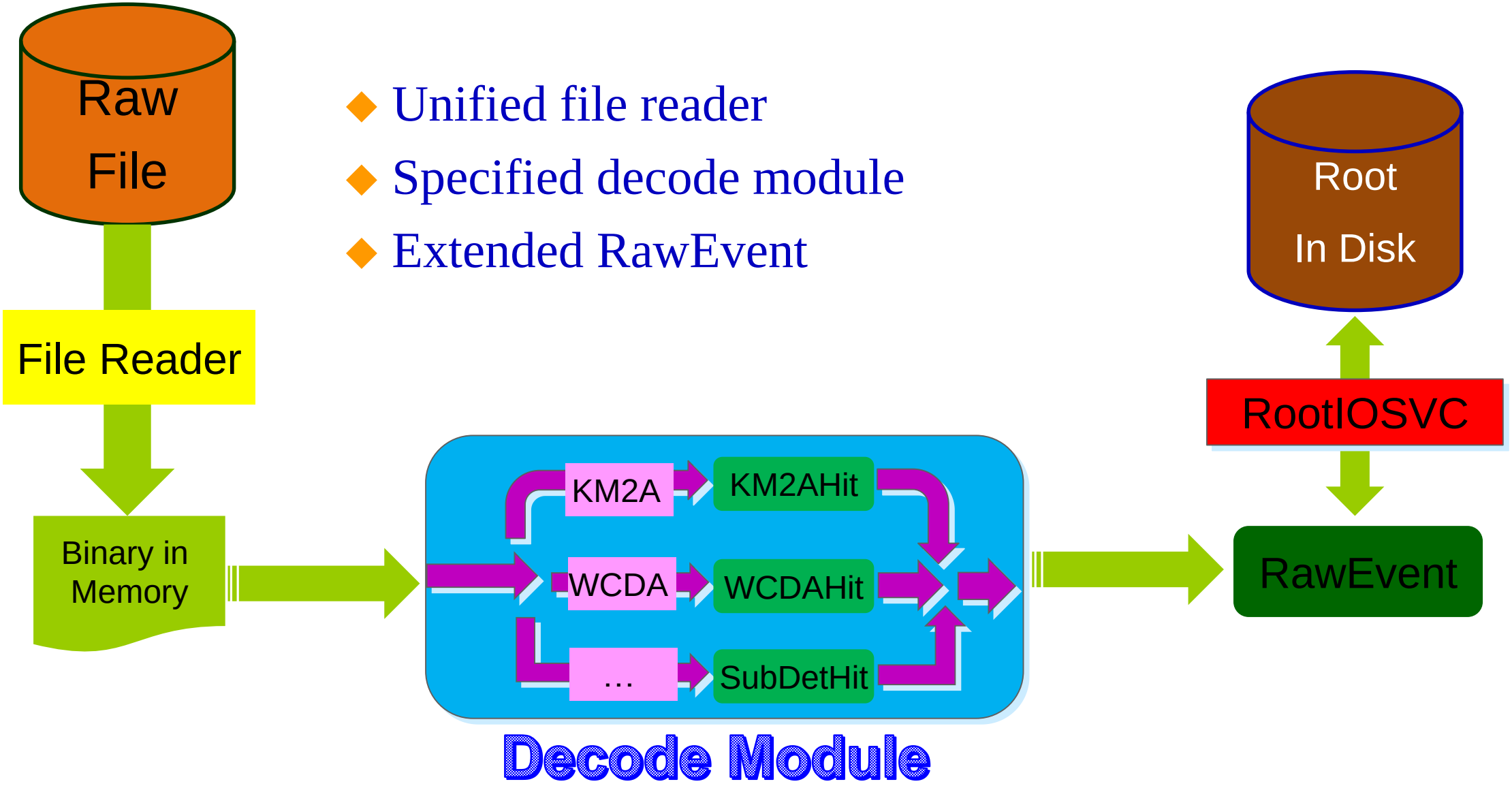
- Two types of IO Systems
- Responsible for reading/ writing event data from/to files.
- Currently support:
  - Corsika Files
  - Root Files
  - Raw Data Files

  (KM2A data supported)



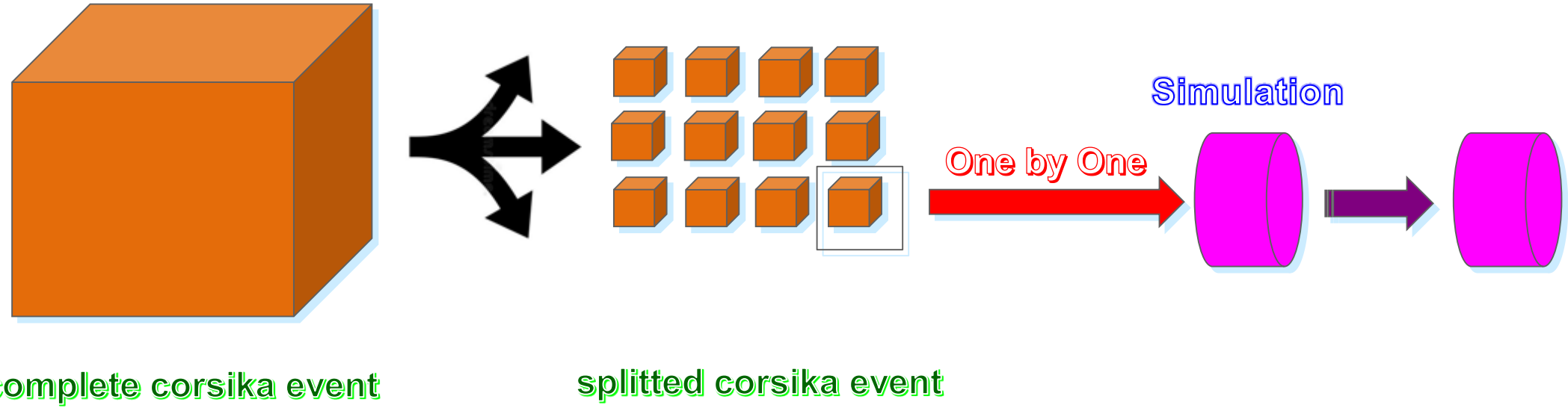
Raw Data Input Module

- Unified file reader
- Specified decode module
- Extended RawEvent



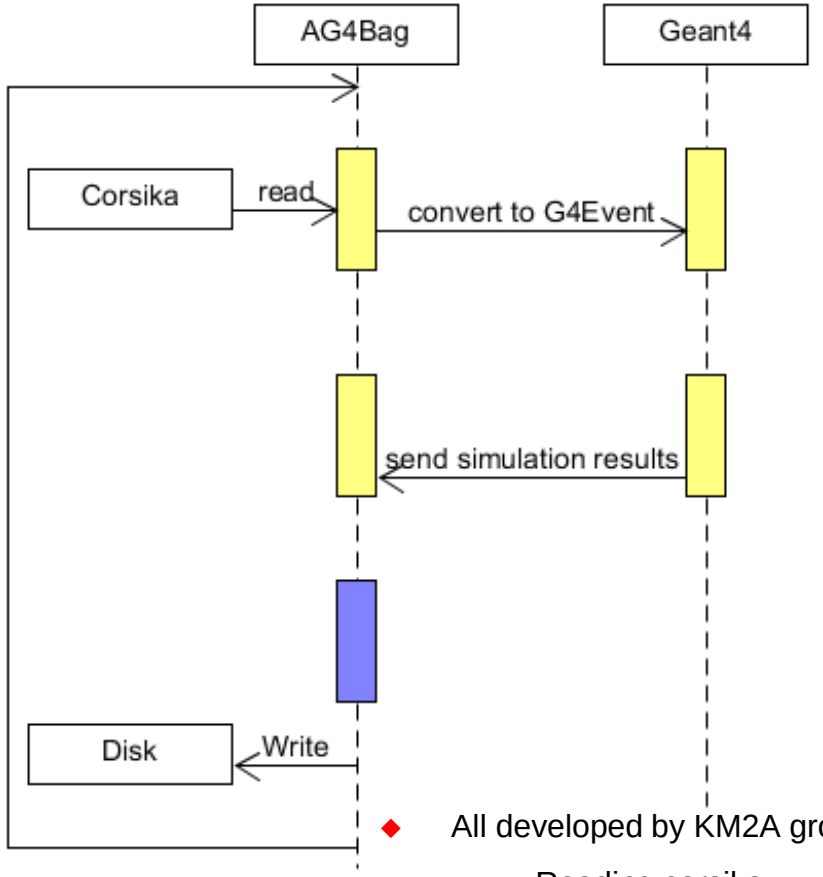
CorsikaInput: A way to deal with large event

- Corsika event Splitting

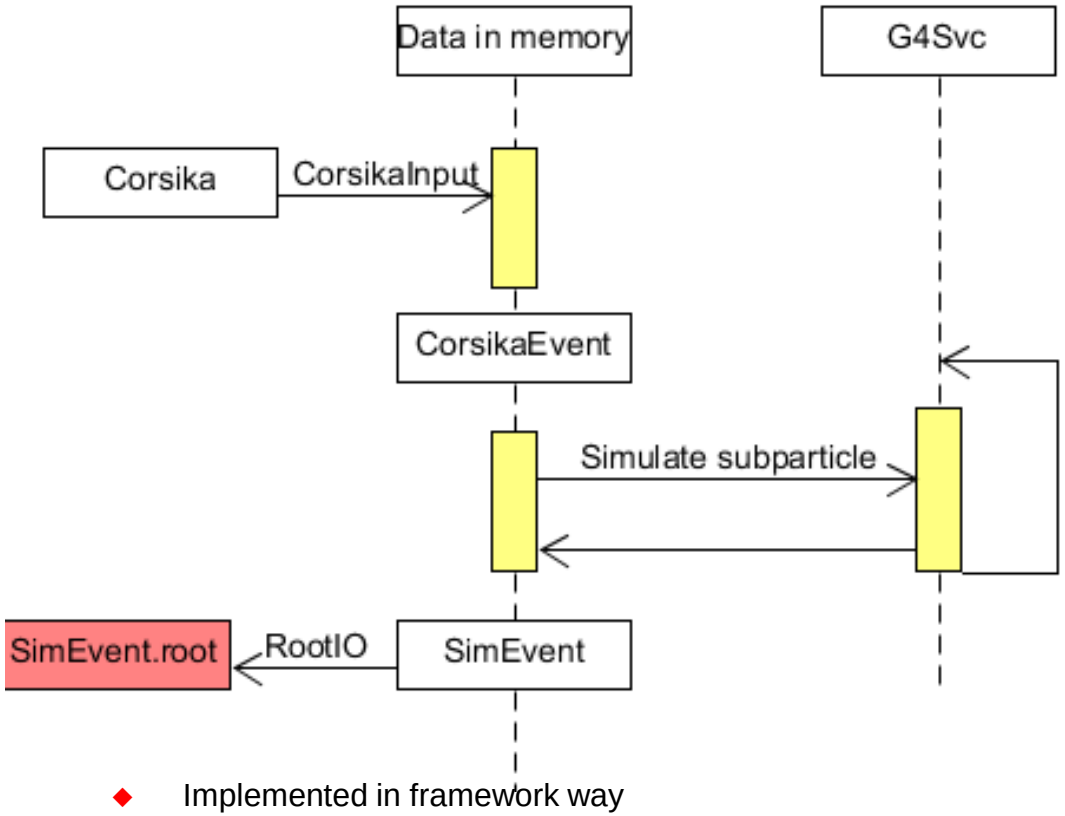


- Split corsika event is the unit of corsika data proceeding

Optimizations in KM2A Simulation



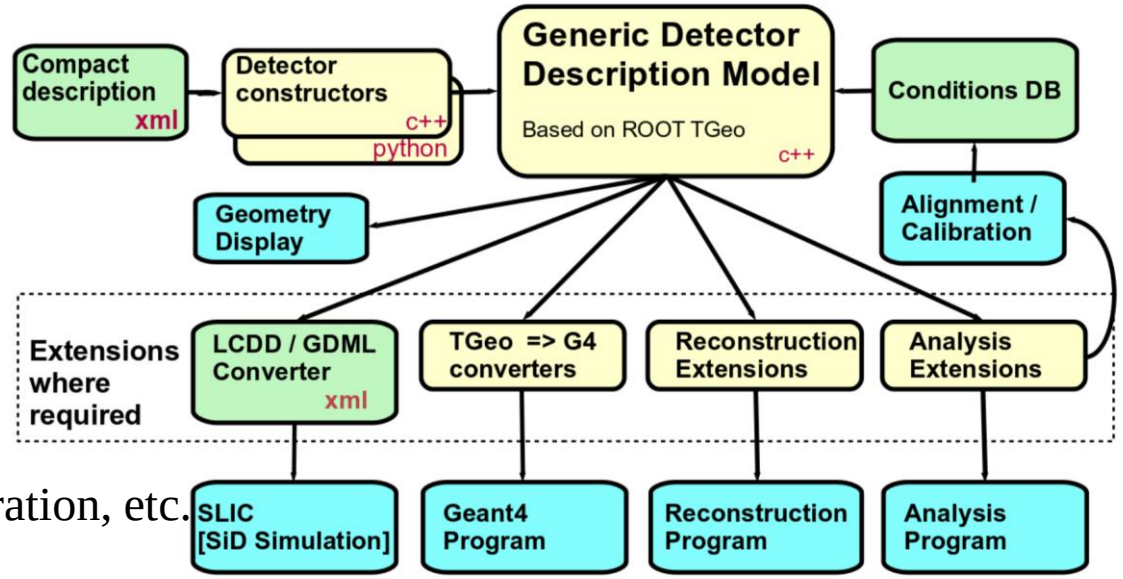
- All developed by KM2A group
  - Reading corsika
  - Event loop
  - Writing into file
- Codes are used theirself



- Implemented in framework way
  - Corsika input
  - G4Svc
  - RootIO
- Codes can be used in other applications

DD4hep

- A general detector (geometry) description toolkit for HEP
- Developed in AIDA and AIDA2020
- Used by ILC, CLIC, FCC
- Under Evaluation by EIC, CMS and LHCb for upgrade
- Key functionalities
  - ⇨ Full Detector Description
    - Includes geometry, materials, visualization, readout, calibration, etc.
  - ⇨ Full Experiment life cycle
    - From concept development, detector optimization, construction to operation.
  - ⇨ Consistent Description with single source
    - for simulation, reconstruction, analysis, etc.
  - ⇨ Ease of Use
- Ouput formats/interface
  - ⇨ GDML
  - ⇨ Geant4 geometry



STCF Geometry

- Vertex Detector
- Solenoid Coil
- MDC
- Ring Imaging Cherenkov Counter
- Electromagnetic Calorimeter
- MU Detector

XML(DD4hep)

STCF_pre.xml

Combined XML

Combine in G4

STCF.gdml

- Detector of Internally Reflected Cherenkov light

Geant4

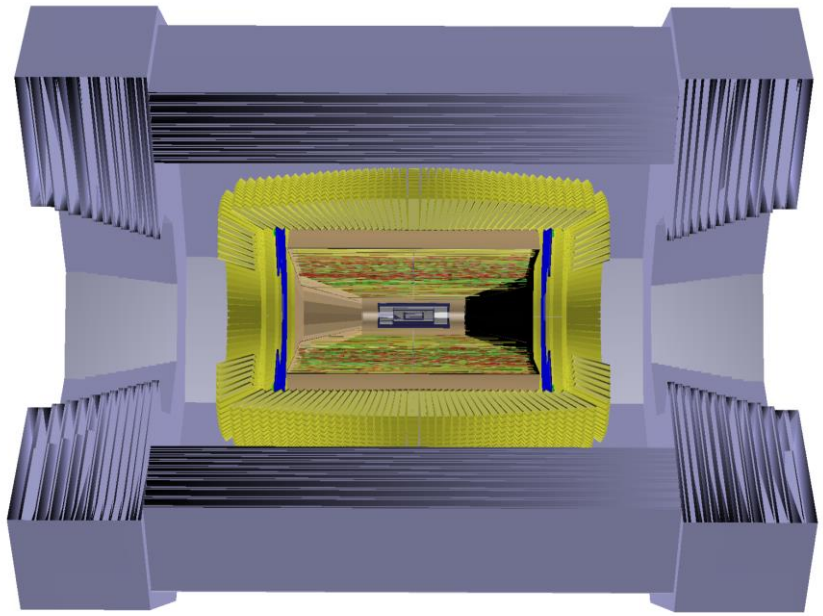
Visualization of STCF Detector with DD4hep

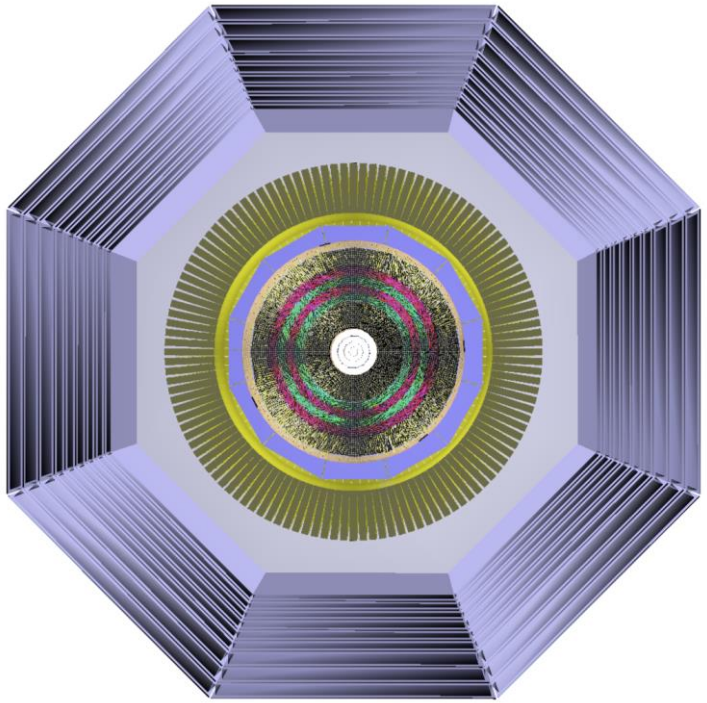
```
<detectors>
    <comment>Trackers</comment>

     <detector name="AirTube" type="AirTube" vis="VXDVis" id="42" insideTrackingVolume="true">
      <dimensions rmin="10.*mm" rmax="11.*mm" zhalf="6.250000000e+01*mm"/>
    </detector>
...

 </detectors>
```

From Dong Liu





STCF Geometry

- Vertex Detector
- Solenoid Coil
- MDC
- Ring Imaging Cherenkov Counter
- Electromagnetic Calorimeter
- MU Detector

XML(DD4hep)

STCF_pre.xml

Combined XML

Combine in G4

- Detector of Internally Reflected Cherenkov light

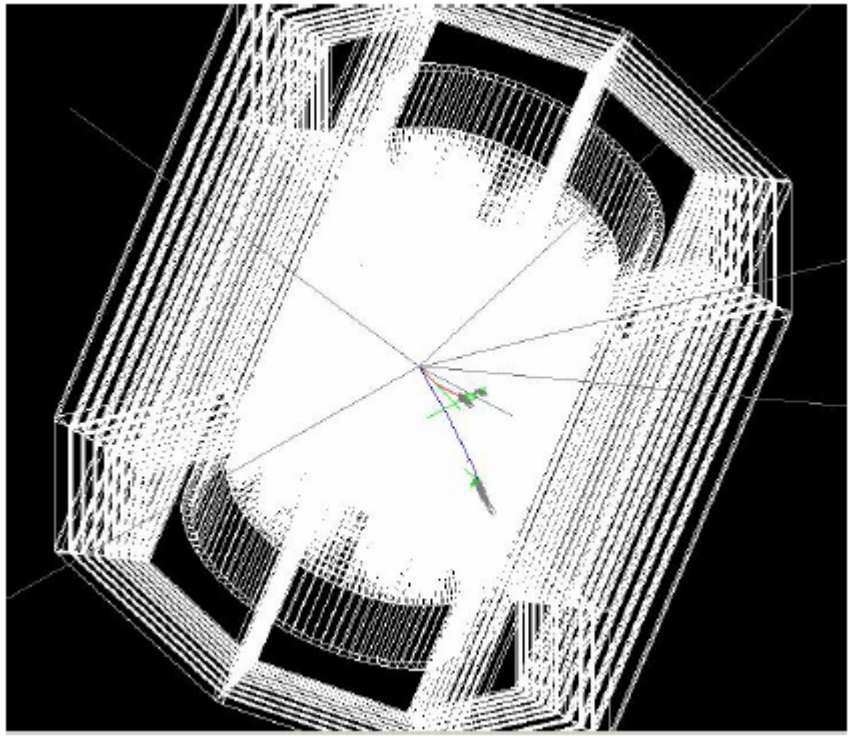
Geant4

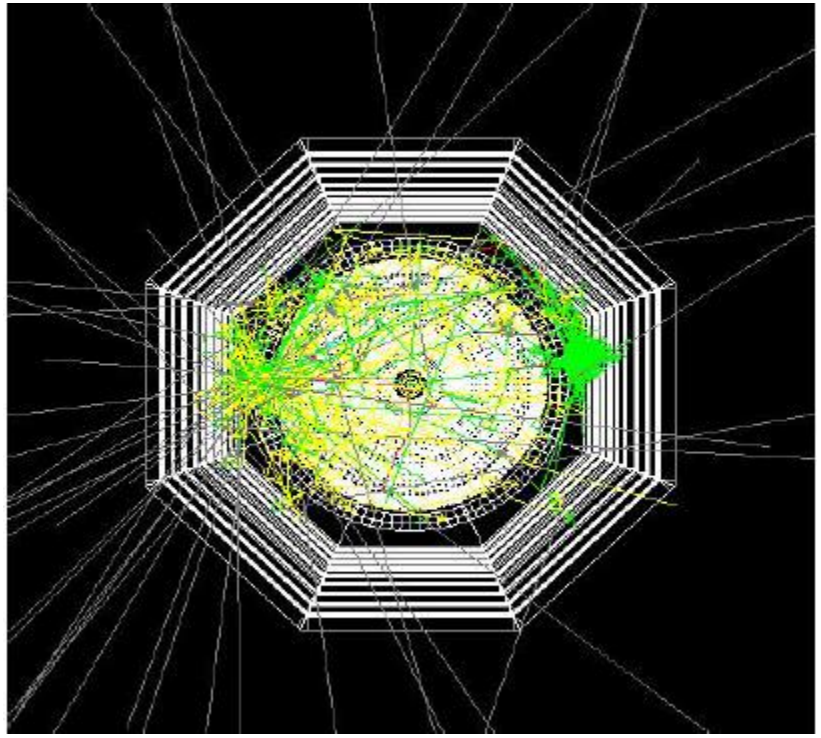
STCF.gdml

Geant4

ROOT

STCF Geometry(gdml in Geant4)





Documentation, Release and SVN

- For LodeStar
  - ⇨ The latest working version of LodeStar has been installed in iHep computing nodes
    - /afs/ihep.ac.cn/soft/LHAASO/LodeStar-SLC6/Pre-Release/L19-Pre1_v1r1
  - ⇨ Documentation
    - LodeStar User Guide
      - http://202.38.128.85/docdb/DocumentDatabase
  - ⇨ SVN repository
    - http://svn.lhaaso.ihep.ac.cn/LodeStar/
- For OSCAR
  - ⇨ The latest working version of OSCAR has been installed in USTC computing nodes
    - stcf01.ustc.edu.cn
  - ⇨ Documentation
    - OSCAR User Guide
      - http://cicpi.ustc.edu.cn/indico/getFile.py/access?contribId=1&resId=0&materialId=slides&confId=1610
  - ⇨ SVN repository
    - http://202.141.163.202/svn/oscar/

Summary

- SNiPER has been adopted in LHAASO and STCF
  - ⇨ LodeStar
    - M.C. Data
    - Raw Data
  - ⇨ OSCAR
    - M.C. Data
- Detector Simulation
  - ⇨ KM2A, WCDA and WFCTA have been integrated with LodeStar
  - ⇨ Whole detector simulation has been implemented in OSCAR
- Reconstruction
  - ⇨ KM2A, WCDA and WFCTA have been integrated with LodeStar
  - ⇨ Reconstruction algorithm is developing in OSCAR
- The whole chain from Raw data to physics analysis has been setup in LodeStar