For INTT offline software

Takashi Hachiya

Nara Women’s University

Introduction

Status:

Hardware development is going to be completed soon.

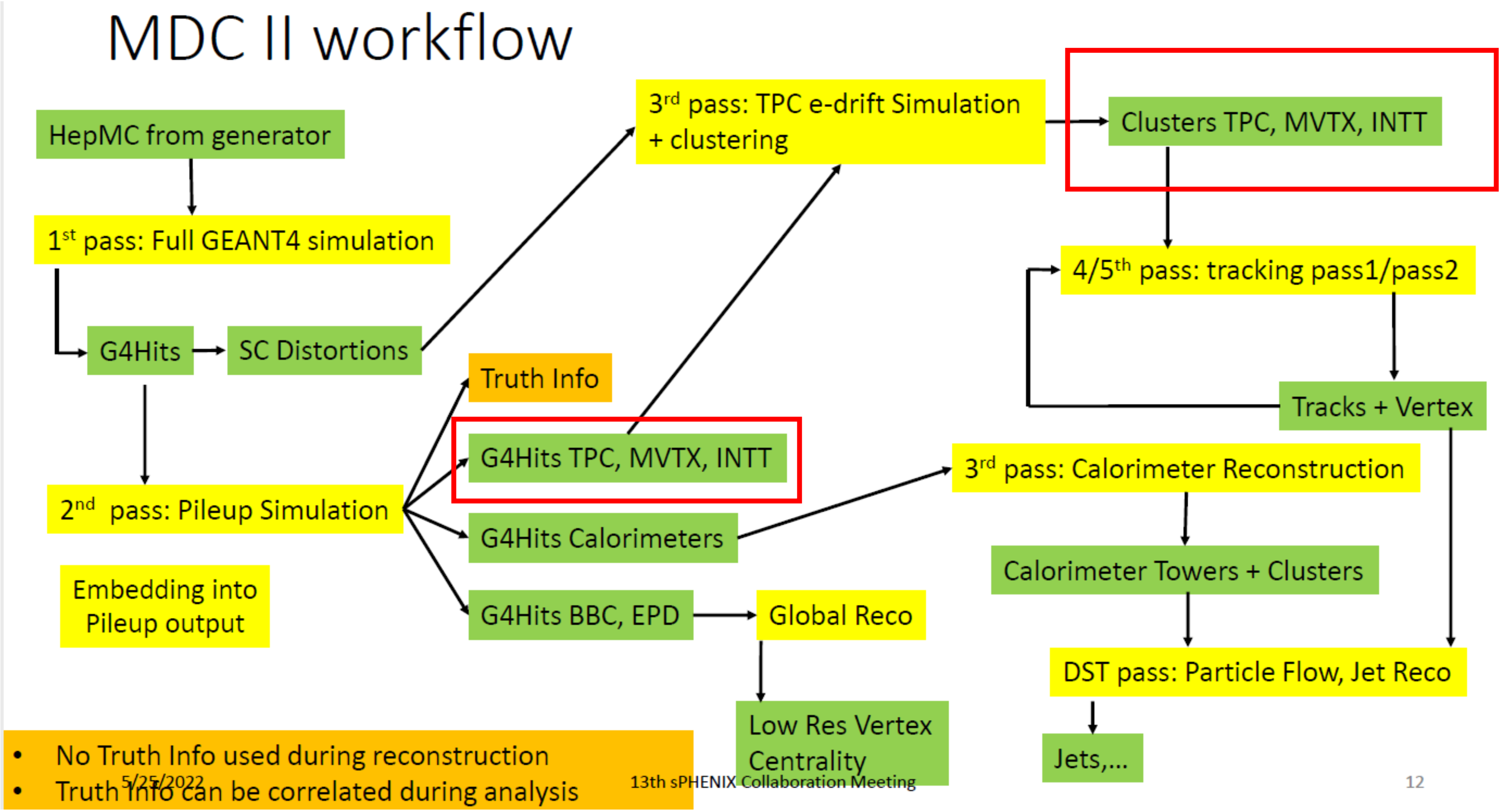
We will be moving to the offline software on sPHENIX framework

To start for the software development, We need to know

- What codes we are responsible to develop in the offline code.

We had a meeting with offline software group (tracking group) to know the current status of the offline code

- I would like to share it
  - Flow of the sPHENIX reconstruction, especially tracking. (INTT cluster is a part of track object (or seed)



MDC II workflow

HepMC from generator

1st pass: Full GEANT4 simulation

G4Hits

SC Distortions

2nd pass: Pileup Simulation

Embedding into Pileup output

Truth Info

G4Hits TPC, MVTX, INTT

G4Hits Calorimeters

G4Hits BBC, EPD

Global Reco

Low Res Vertex Centrality

3rd pass: TPC e-drift Simulation + clustering

Clusters TPC, MVTX, INTT

4/5th pass: tracking pass1/pass2

Tracks + Vertex

3rd pass: Calorimeter Reconstruction

Calorimeter Towers + Clusters

DST pass: Particle Flow, Jet Reco

Jets,…

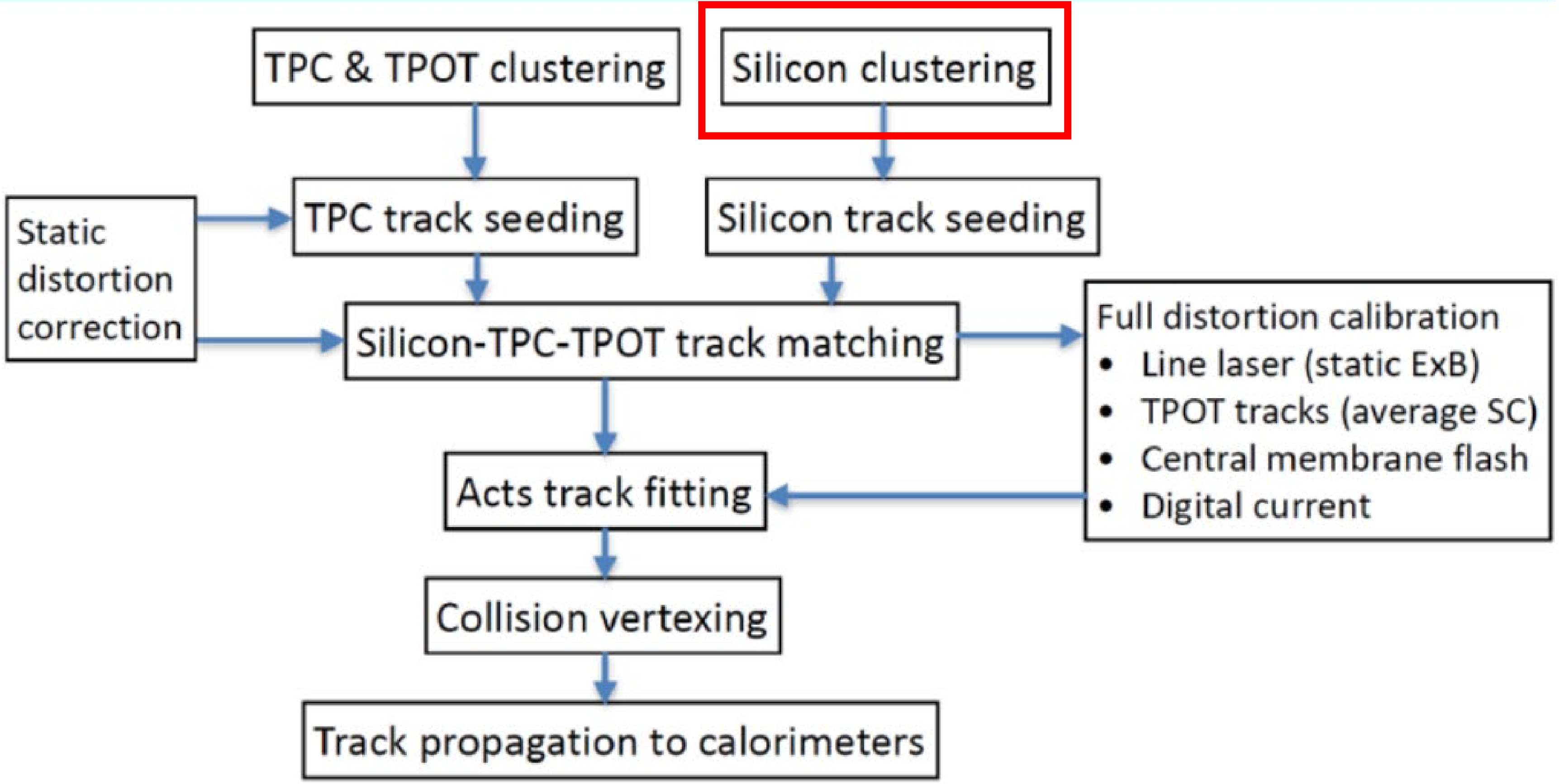
- No Truth Info used during reconstruction
- Truth Info can be correlated during analysis

5/25/2022 13th sPHENIX Collaboration Meeting 12

- Tracking is basically performed with TPC, INTT, MVTX
  - Calorimeter clusters are not in the tracking code

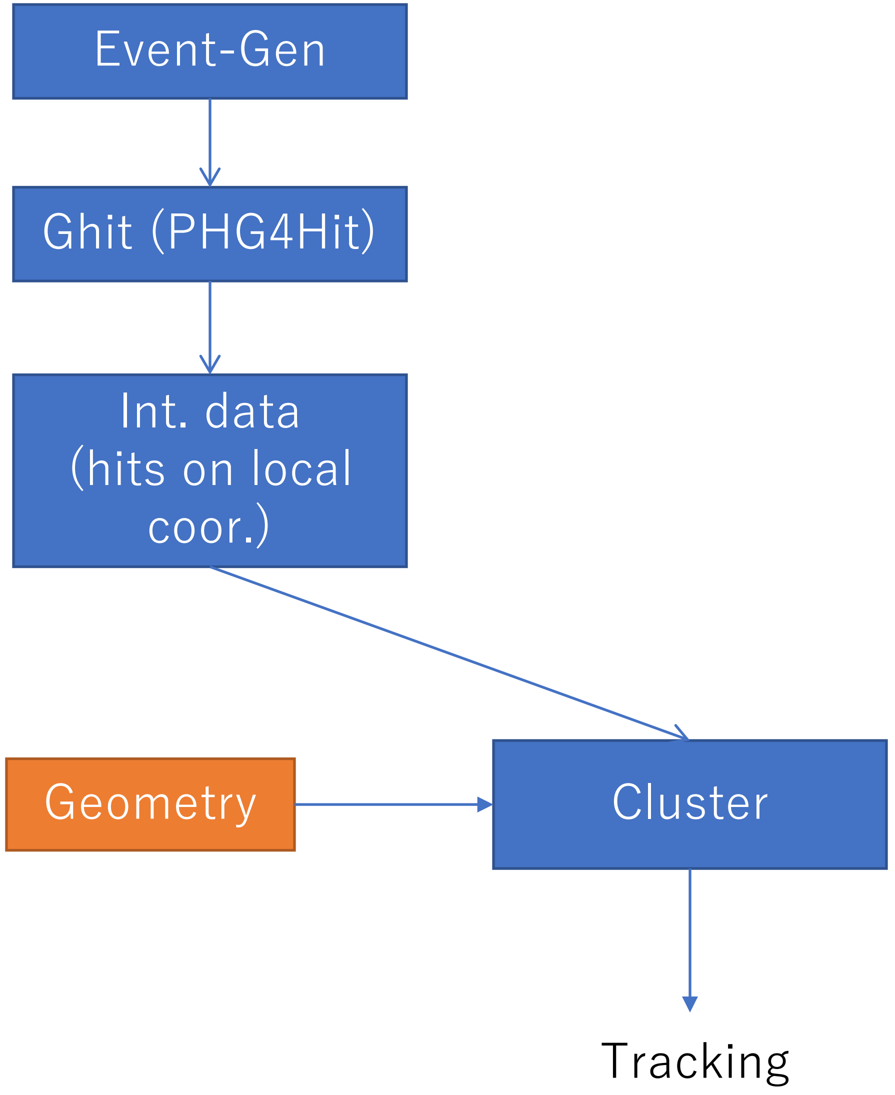
- Both the reconstructed cluster and MC truth are made (and saved) in the flow

Simplified overview of tracking workflow


TPC & TPOT clustering
Silicon clustering
Static distortion correction
TPC track seeding
Silicon track seeding
Silicon-TPC-TPOT track matching
Full distortion calibration
• Line laser (static ExB)
• TPOT tracks (average SC)
• Central membrane flash
• Digital current
Acts track fitting
Collision vertexing
Track propagation to calorimeters

- Two independent track seeds: one is from TPC and the other is from Silicon (MVTX and INTT)
- INTT is required to produce 3D hit positions （clusters)

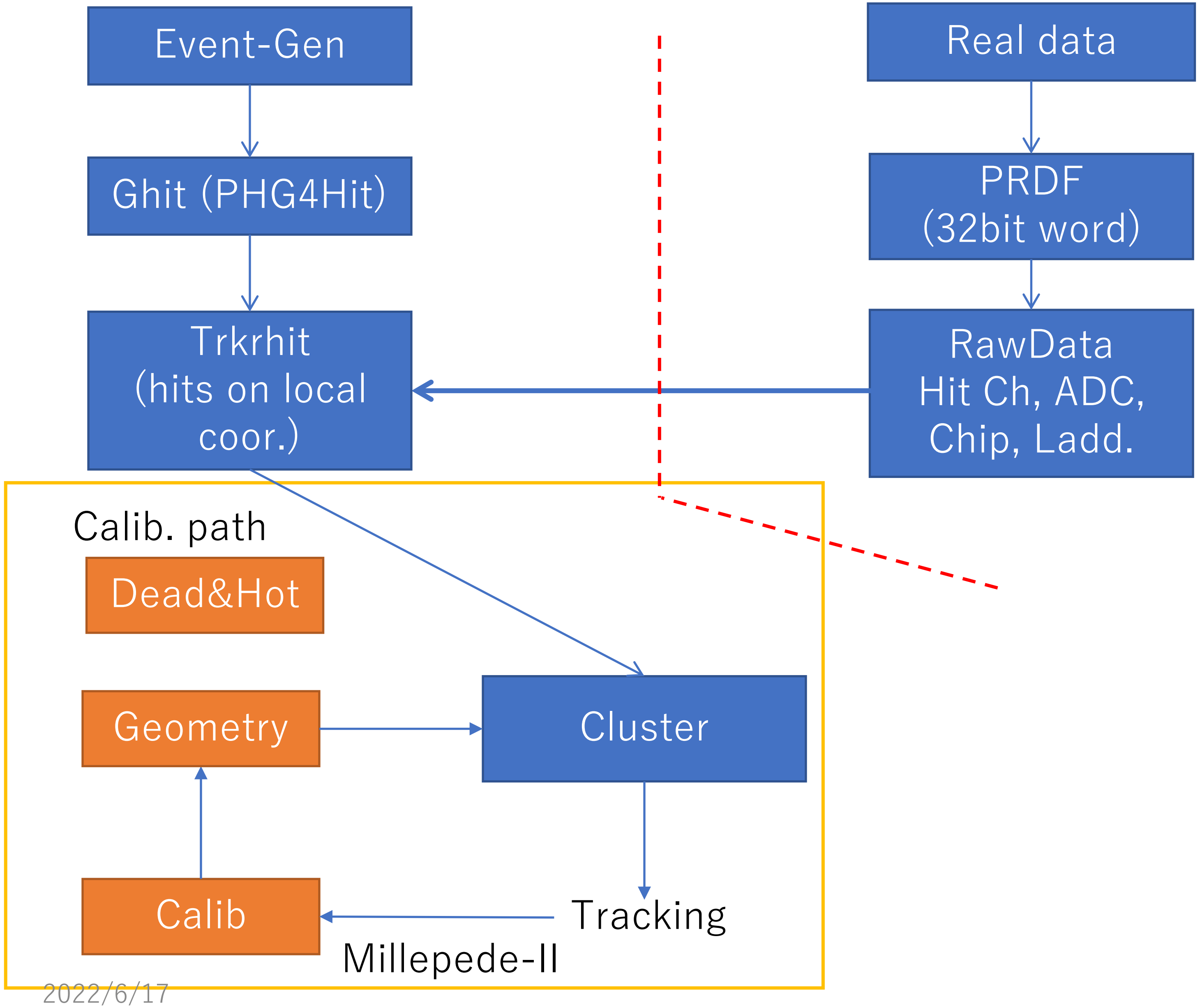
Data Flow for INTT clustering in simulation



- Flow of INTT simulation in sPHENIX GEANT and reconstruction code
    - Geometry object is used to convert the hit in the local coordinate (2D) to the global coordinate (3D)
    - Geometry object is implemented in the tracking framework (ACTS)

- Similar flow should be prepared for the real data

Data Flow for INTT clustering in real data



Online part

Decoder/Unpacker (iValue())
for INTT

- In online framework
- Data format is not known , ask online experts

After the discussion, we agreed to make “RawData” module and interface to “Trkrhit” by INTT group.

Questions and answers

Q1: Rawhit and Cluster objects are defined and made?

- Yes for Cluster and Trkrhits but no for Rawhit

Q2: Geometry object is defined and made?

- Yes. It is implemented in ACTS framework. The code is available in the github.

Q3: What is an interface to Millepede-II?

- No. Millepede alignment is not started yet. It is just an idea

Q4: Dead/Hot object is defined and made?

- Yes, we need to prepare 0th version of the map once the barrel assembly is settled

Q5: What is an interface to the database?

- Generic module is available, we need to develop for INTT calibration

Q6: How is the ADC value kept?

- Yes, the cluster has ADC variable, we need to think about what we store .

Q7: How is the timing (beam clock) used to resolve the pile-up?

- Yes. Beam crossing value is stored for the hit list (cluster list) event by event
- It looks the beam crossing value is different with the BCO but the (UNIX) time divided by 106ns

Q8: How are the documents managed

- They said the code is updating so quick. No document is prepared and the code is the doc for now
  - They pointed the list of coded related to INTT. I should go though these codes.
- We should write the doc by ourself for us and others

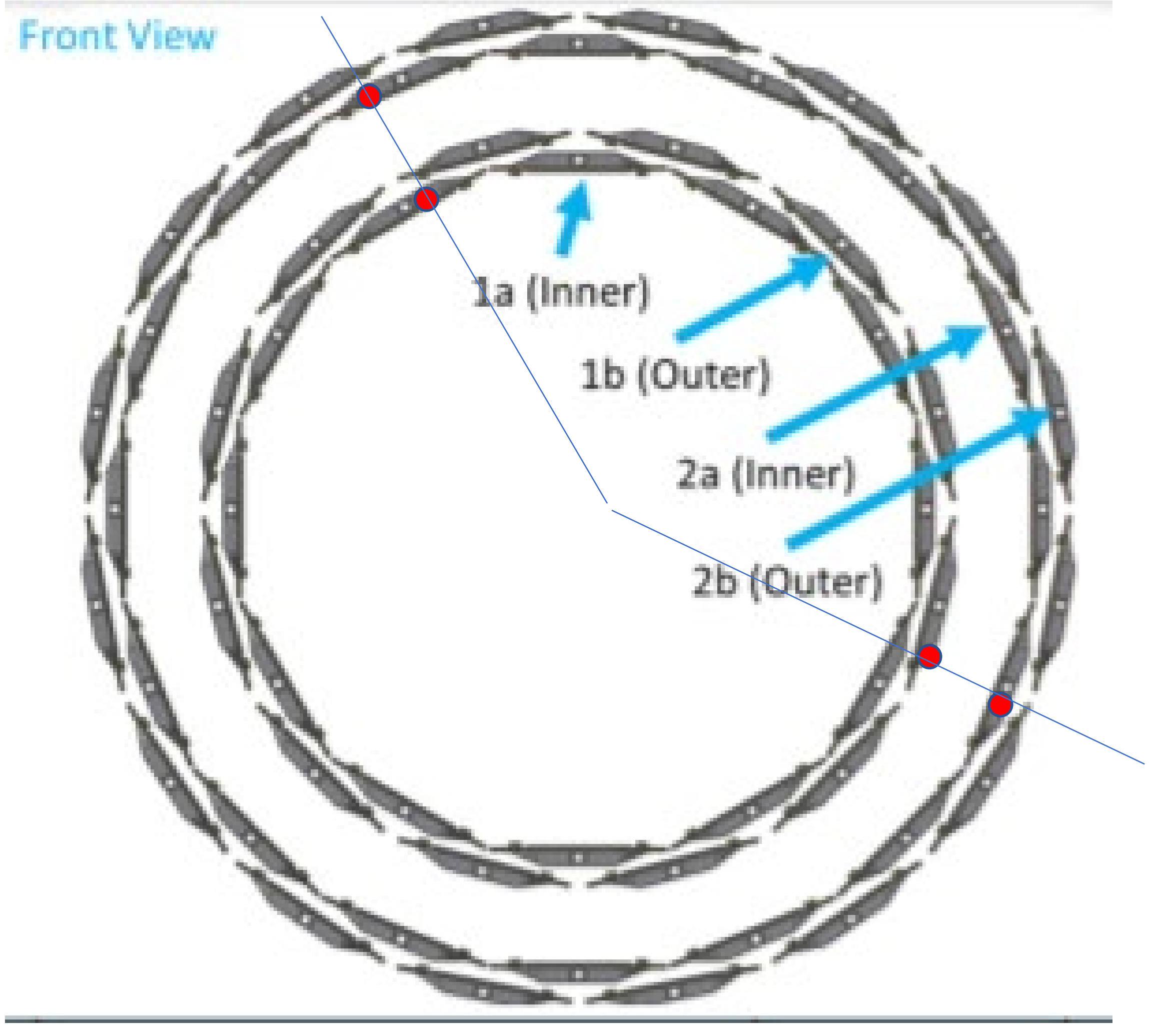
Summary

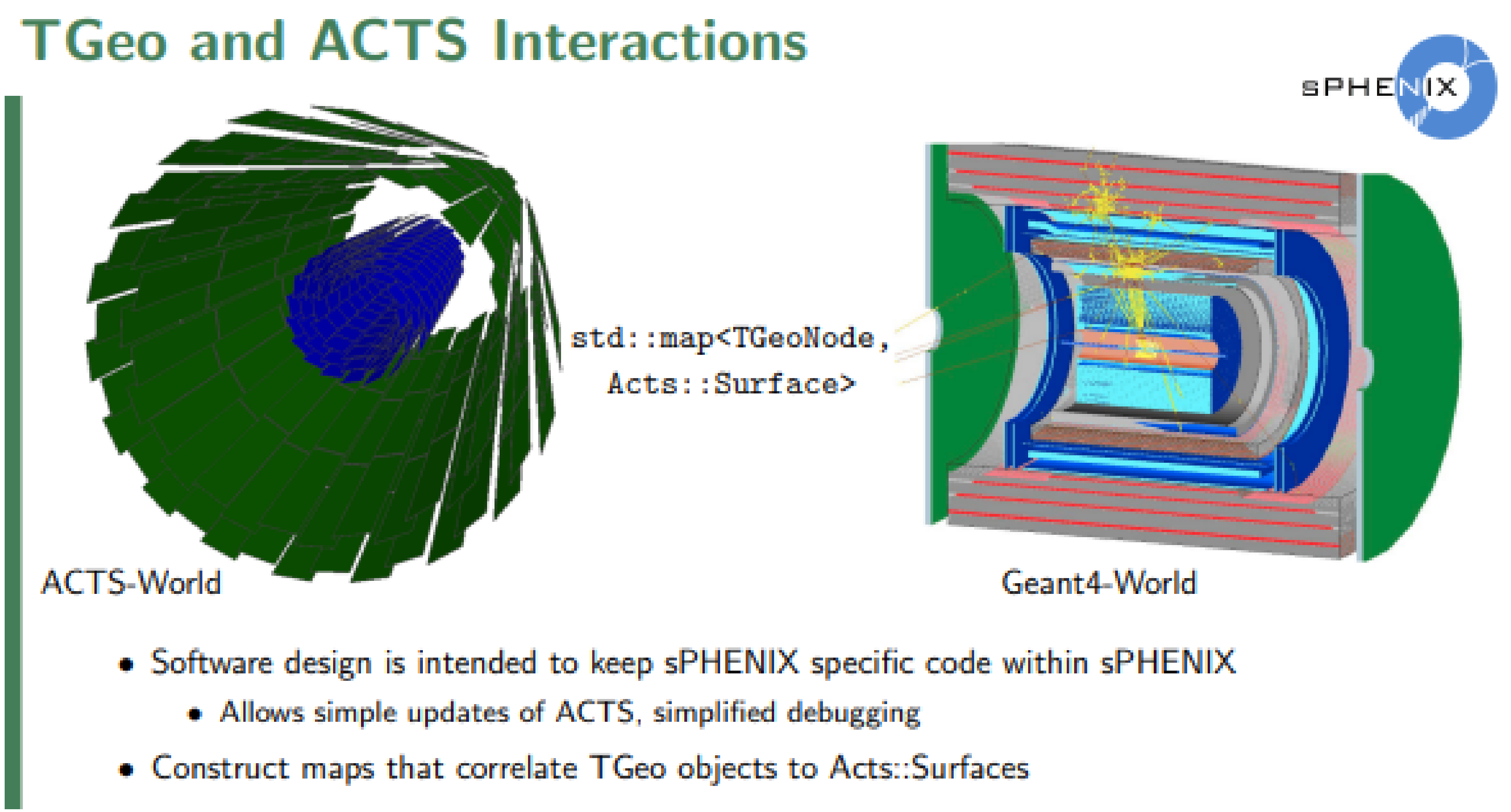
- We learned what the status of the offline code is and what we need to develop.
- Nest step, I will go though the code to know what the actual modules are and how they work.
  - I will report the structure of the offline code in the next meeting or soon

INTT Event Display

- We plan to make an INTT event display for QA and sanity check

- Offline code uses ACTS framework for tracking.
  - We would like to follow it
  - We would like to play with ACTS





TGeo and ACTS Interactions

- Software design is intended to keep sPHENIX specific code within sPHENIX
  - Allows simple updates of ACTS, simplified debugging
- Construct maps that correlate TGeo objects to Acts::Surfaces

- Acts is a software project being developed by ATLAS/Belle2/LHCb (and other) collaborators
- Intended to be an experiment independent set of track reconstruction tools
- Performant and flexible algorithms for track reconstruction
- See also Xiaocong Ai's and Bastian Schlag's talks at CTD2020

- ACTS contains a ROOT TGeo plugin
- Like most experiments, sPHENIX has a full Geant4 description
- Provide ACTS with (already created) `TGeoManager` object to build tracking detectors
- ACTS creates Surfaces that correspond to TGeo objects in the G4 description

INTT module in GEANT4 macro

```
// Initialize the selected subsystems
 G4Init();  // if (Enable::INTT) InttInit(); // set parameters to BlackHole parameters

 //---------------------
 // GEANT4 Detector description
 if (!Input::READHITS) G4Setup();    //  if (Enable::INTT) radius = Intt(g4Reco, radius);
 //                                          //   PHG4InttSubsystem* sitrack = new PHG4InttSubsystem(“INTT”, vpair);
 //------------------
 // Detector Division
 if (Enable::INTT_CELL) Intt_Cells();
 //  if (G4INTT::InttDeadMapOption != G4INTT::kInttNoDeadMap)
 //                 PHG4InttDeadMapLoader* deadMapINTT = new PHG4InttDeadMapLoader(“INTT”);
 //
 //  PHG4InttHitReco* reco = new PHG4InttHitReco();         //  // new storage containers
 //  PHG4InttDigitizer* digiintt = new PHG4InttDigitizer(); //  // new containers

 //--------------
 // SVTX tracking
if(Enable::TRACKING_TRACK) TrackingInit();
if(Enable::INTT_CLUSTER)   Intt_Clustering();   //  InttClusterizer* inttclusterizer = new InttClusterizer(“InttClusterizer”,
                                                //    G4MVTX::n_maps_layer, G4MVTX::n_maps_layer + G4INTT::n_intt_layer - 1);
if(Enable::TRACKING_TRACK) Tracking_Reco();
```