

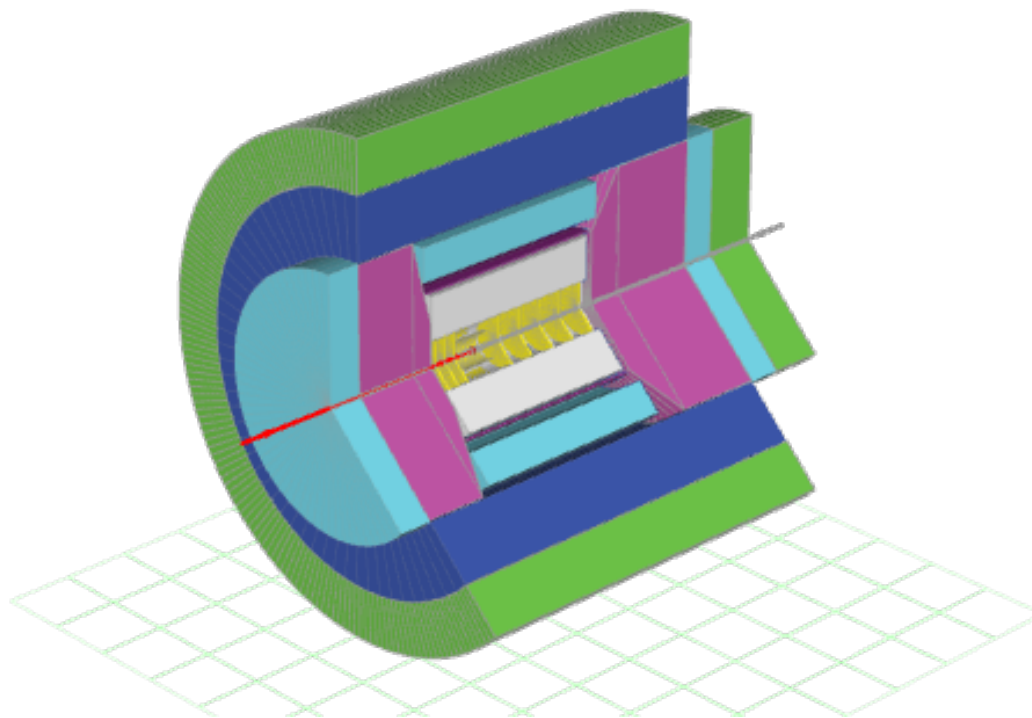


EICROOT

Simulations Software Framework for EIC

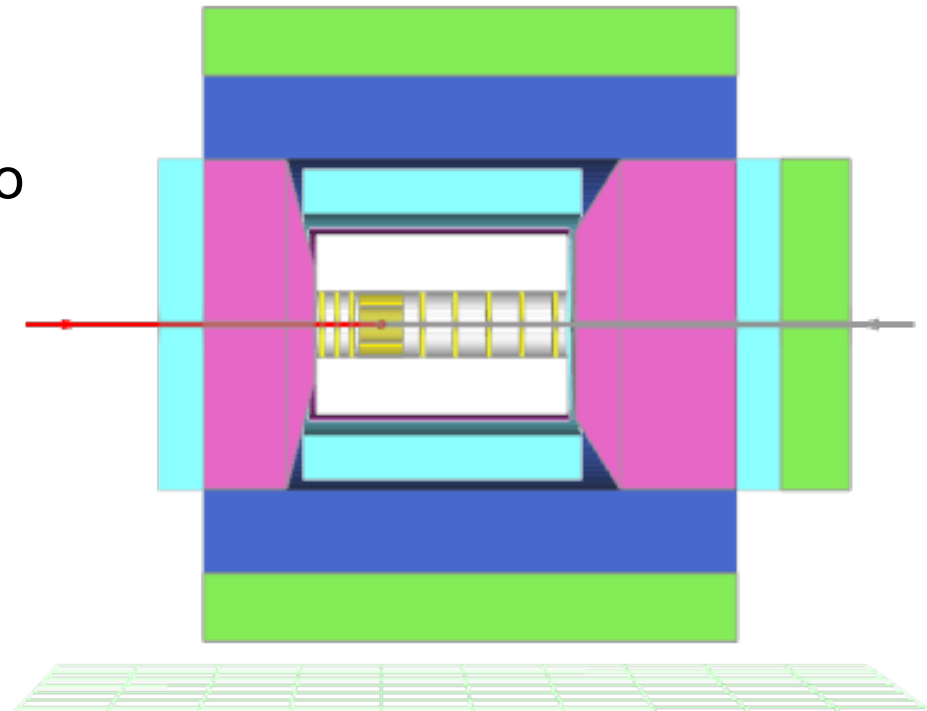
Yulia Zulkarneeva

BNL, Sep 2011



Outline

- **Intro: Current EIC Sim.Software**
 - Overview and Path Forward;
- **HEP Simulations Frameworks**
 - Why do we need HEP framework;
 - Historical perspective: from 1960s to present time;
 - VMC and VGM
- **EICRoot Framework**
 - Initial Implementation:
 - EIC Geometry + Hits;
 - Event Display;
 - Fast Simulation;
 - Event Generator Data Import;
 - Radiation Length Scan;
- **Summary and Outlook**



Current EIC Simulations State, as reported

(based on Matt's talk, mostly..)

✓ **Event generators :**

- Pythia6, SARTRE, Milou, PEPSI, gmc_trans, RAPGAP 3, HIJET, DJANGO, MC@NLO, LEPTO, DPMJET III, CASCADE;
- Mostly fortran, some have C/C++ wrappers;
- ASCII output format, unified (HEPEVT);
- ASCII → ROOT conversion using macros;

✓ **Fast Simulation :**

- Implemented using momentum/energy smearing technique;
- Does it involve realistic detector parameters characterization?

✓ **Detector geometry description implemented independently** within :

- Geant 3 / STARSIM – ready to use, but docs are limited;
- Geant 4 – status is not clear..
- Fluka – ready to use..

✓ **To-Do** : lots of things to do..

Current state summary

OK, current simulations state works great for basic estimates and fast simulations. **What's next?** Need to fix something first :

- **unified interface for transport models and detector geometries.** It is required for 1:1 comparison of detector concepts using G3/G4 (+Fluka, if licensed) transport codes;
- **unified event generator interface** to feed data into simulation transport engine(s);
- **pluggable/extensible magnetic field management** – forward tracking will require special care!
- **Input/Output management** – IO management should be centralized to avoid costly checks for IO compatibility and code duplication;
- validation of the detector R&D concepts requires more than fast simulation, so **need to decide on some framework today;**

Is it hard to create desired configuration manually, from hundreds of independent HEP packages?



Very hard: fully-featured simulations codes typically consist of few million lines of code. Need something else..

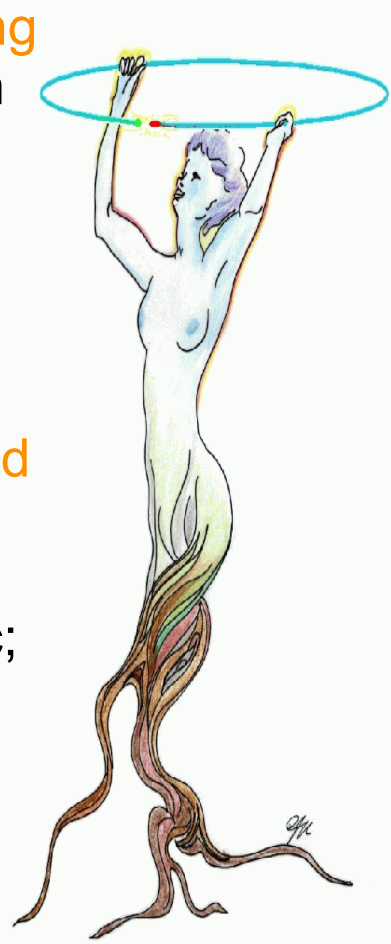
Next Step: Use HEP Framework

What is “HEP Framework” and why should we use one of those?

- **Core idea:** framework is essentially a **unified package, providing generic mechanisms to deal with most commonly used tasks** in HEP. For example, well-known ROOT framework introduced unified IO, generic data analysis procedures, basic detector visualization techniques;
- **Improved productivity:** framework **allows to focus on physics deliverables** while reusing pre-tested software components. Do not submerge into low-level details, **use pre-built and well-tested code** for common tasks. Allows physicists to **concentrate on detector performance details**, avoiding purely software engineering issues like storage, retrieval, code organization etc;

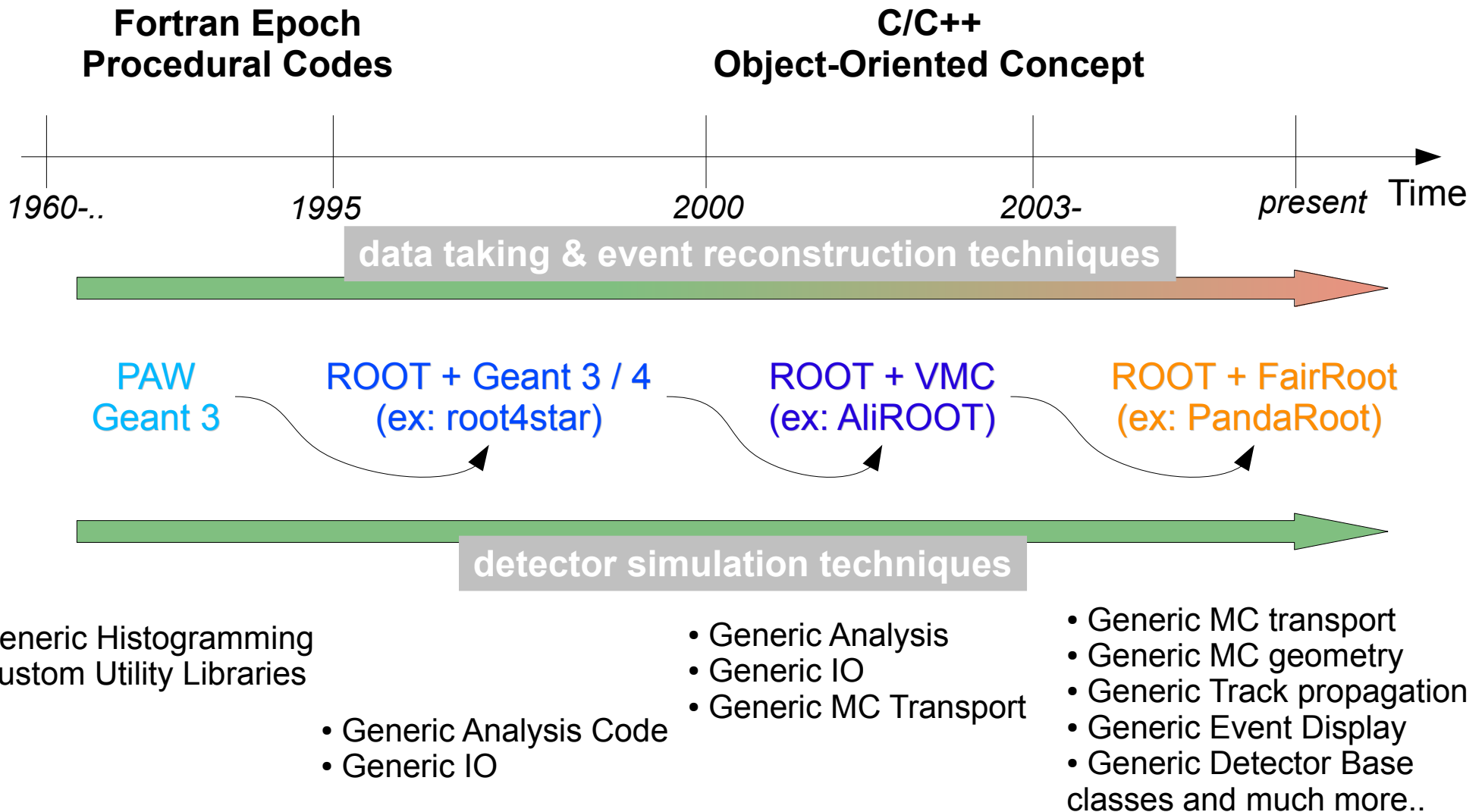
Possible drawbacks:

- **Learning curve:** new framework may mean steep learning curve, especially if this framework is really new and shiny;
- **Framework defines the flow of control:** if your problem is really unique, it could be hard to implement in the constrained model;



HEP Software: Historical Overview

Brief overview of the evolution of physics simulations and analysis software



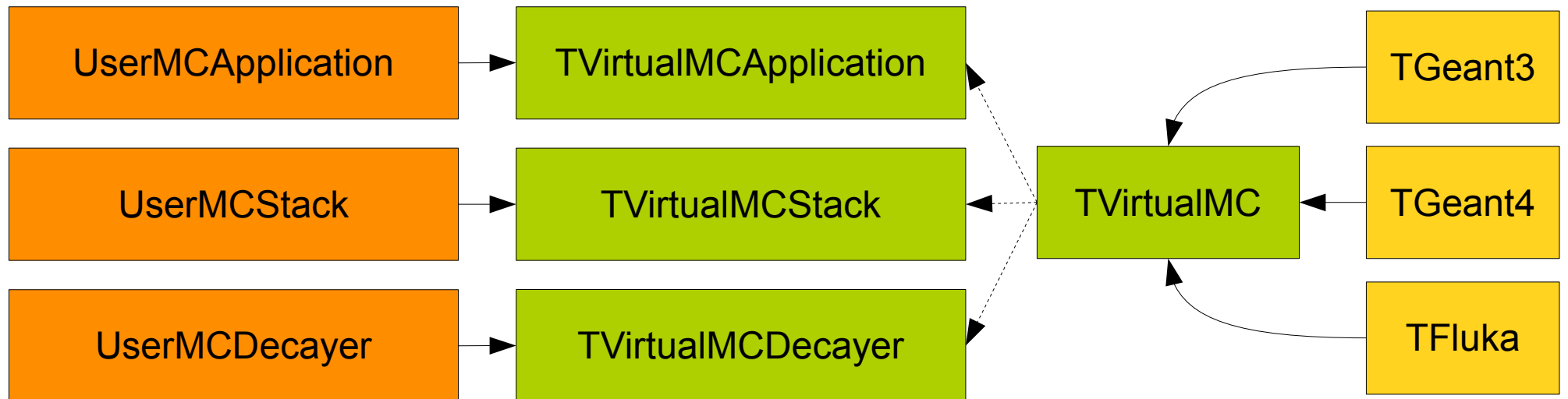
Huge improvement in HEP computing, thanks to Computer Science progress..

VMC: Virtual Monte Carlo

The **Virtual Monte Carlo** (VMC) allows to run different simulation Monte Carlo without changing the user code and therefore the input and output format as well as the geometry and detector response definition.

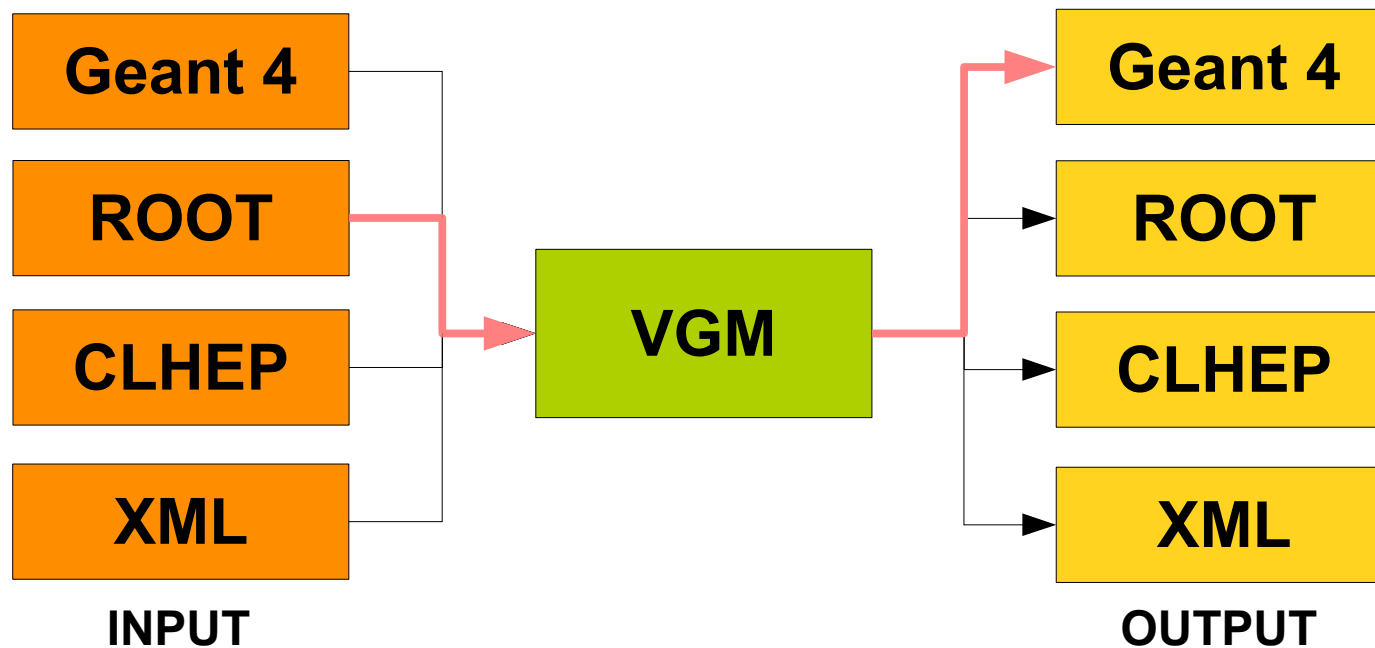
The core of the VMC is the category of classes `vmc` in `Root`. It provides a **set of interfaces which completely decouple the dependencies between the user code and the concrete Monte Carlo**:

TvirtualMC	Interface to the concrete Monte Carlo program
TVirtualMCApplication	Interface to the user's Monte Carlo application
TVirtualMCStack	Interface to the particle stack
TVirtualMCDecayer	Interface to the external decayer



VGM: Virtual Geometry Model

The Virtual Geometry Model (VGM) has been developed as a **generalization of the existing convertors** roottog4, g4toxml provided within Geant4 VMC, when new directions: g4toroot, roottoxml were asked by users. Instead of adding other bilateral converters and multiplying the implementations, **the abstract layer to geometry has been defined** and the geometry models have been "mapped" to this generalized scheme.



the import function of the VGM factory makes it possible to use VGM directly with native geometries (Geant4, TGeo). The export functions provide conversion into other native geometries or the XML format.

What does FairRoot deliver:

- **Key Features:**

- Detector base classes that handle initialization, geometry construction, hit processing (stepping action), etc. **Geant 3, Geant 4, Fluka** (if allowed by Fluka author) – all using **centralized geometry description**. Geant 4 + Native G4 navigation and Geant 3 + Root navigation are pre-configured defaults.
- **IO Manager** based on ROOT TFolder and TTree (TChain);
- **Geometry Readers:** ASCII, ROOT, CAD2ROOT;
- **Radiation length manager;**
- **Generic track propagation** based on Geane;
- **Generic event display** based on EVE and Geane;
- **Fast simulation base services** based on VMC and ROOT TTasks;
- CUDA support;

- **Support and Maintenance**

- FairRoot has **core team of 4 developers** and more than 70 registered developers from the different experiments;
- QA system, Ticket system, Wiki, discussion forum are actively used;

- **User Base:**

- Four large collaborations: CBM, Panda, R3B, MPD. Coordination of efforts with ALICE and ROOT dev teams;

Proposal: EICRoot package

TO BE SUPPORTED
BY EIC S&C GROUP

EICRoot

Geometry

HCAL

EMCAL

RICH

etc..

Tasks

Digitizers

Hit Producers

Track Finding and Fitting

+

FairRoot Framework

IO Manager

Event Display

Track
propagation

Application

Event Generators

Tasks

Detector base

Magnetic Field

Libraries

ROOT

VMC

Geant3

Geant4

GSL

Monalisa

BOOST

GLPK

VGM

ALREADY EXISTS
NO NEED TO SUPPORT

We propose to create **EICRoot** utilizing full power of ROOT and FairRoot

EICRoot: towards LoI goals



Initial TODO List :

- ✓ Install and configure base software stack (simple);
- ✓ Describe detector geometry in any of the supported formats;
- ✓ Create basic “Hit Producer” implementations for each detector;
- ✓ Enable magnetic field (constant/solenoid B field at first, detailed map later);
- Import event information in HEPEVT/EIC format (pythia6, RAPGAP, etc), **or write your own event generator based on Framework interface..**
- ✓ Run Full Simulation using G3/G4, and process output, all in native ROOT format;
- ✓ Run Fast Simulation, to produce smeared output based on detector configurations;

Extended TODO :

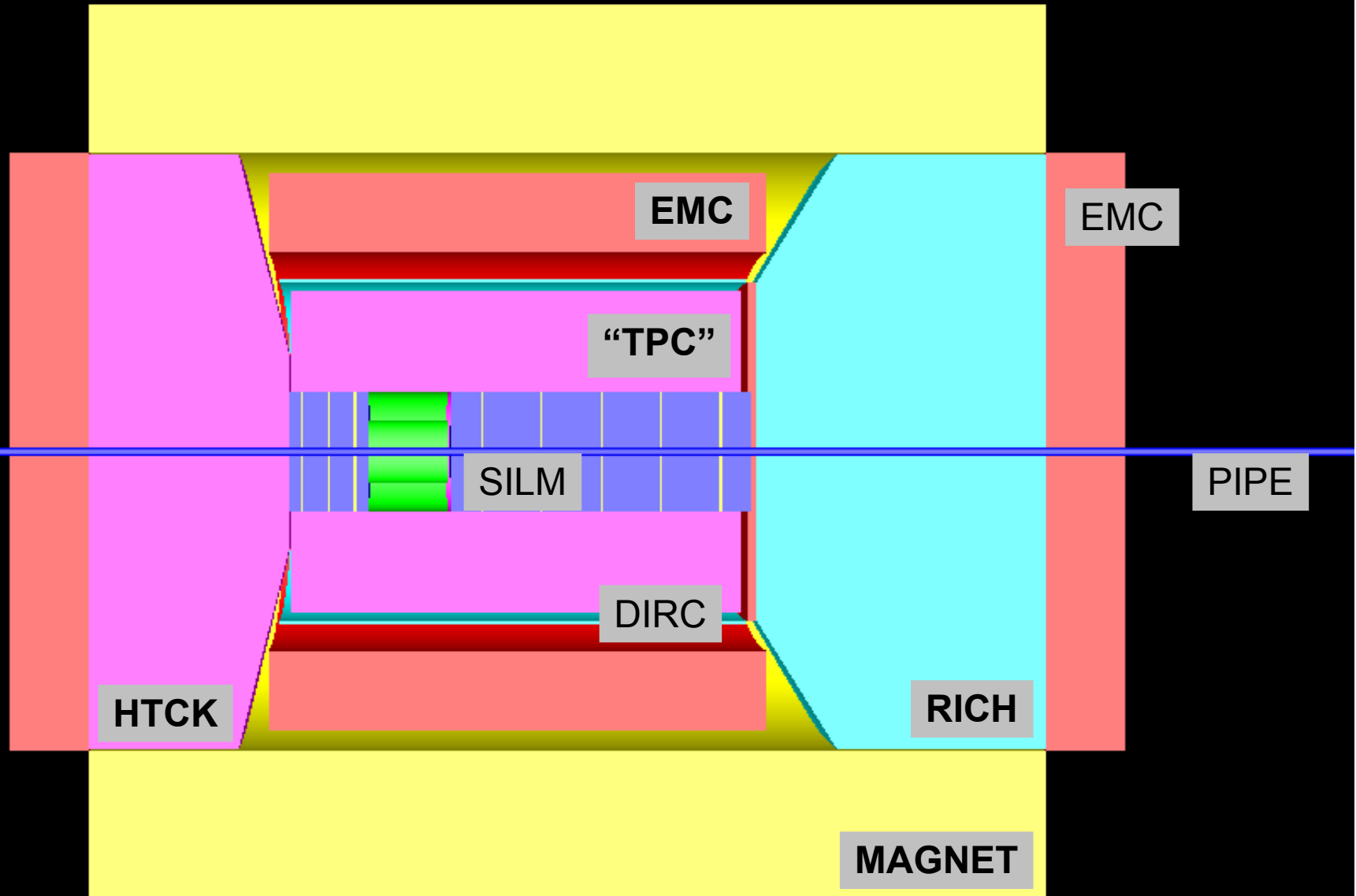
- **Check material budget of various tracker implementations;**
- **Study and compare magnetic field options;**
- Review and verify tracking position reconstruction codes;

Instant Benefits :

- Use Geant 3 and Geant 4 as transport packages, keeping single version of detector geometry (Fluka is also possible, but Fluka with VMC support requires special blessing from author);
- Detector geometry could be easily imported and exported to many formats;

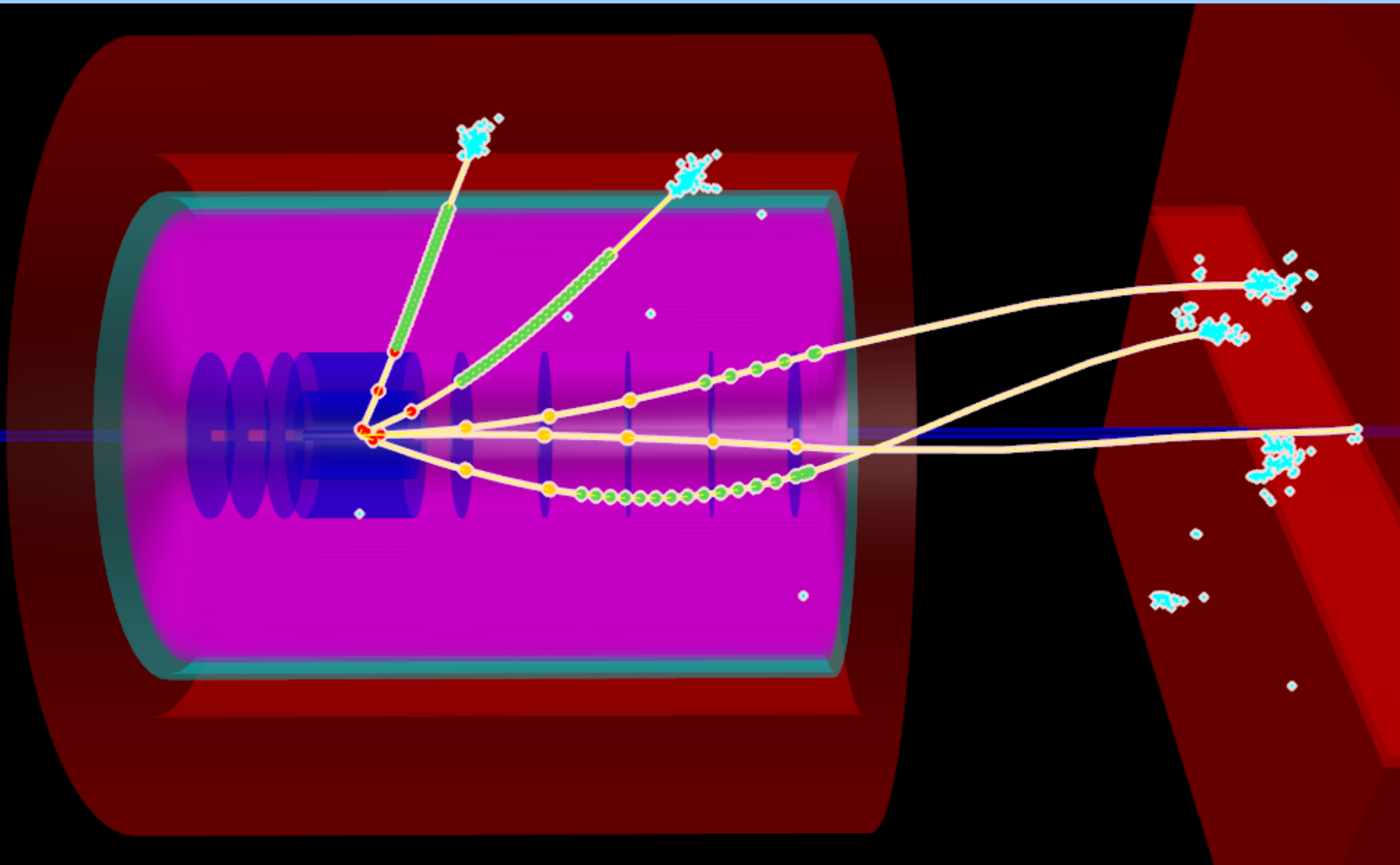
Implemented EIC Geometry Overview

3D render (raytrace) of the implemented EIC geometry (based on pdf table)



random ROOT colors – no color coding

MC Hits display: five e^- generated hits in **SIT** + **TRS** + **TPC** + **EMC**



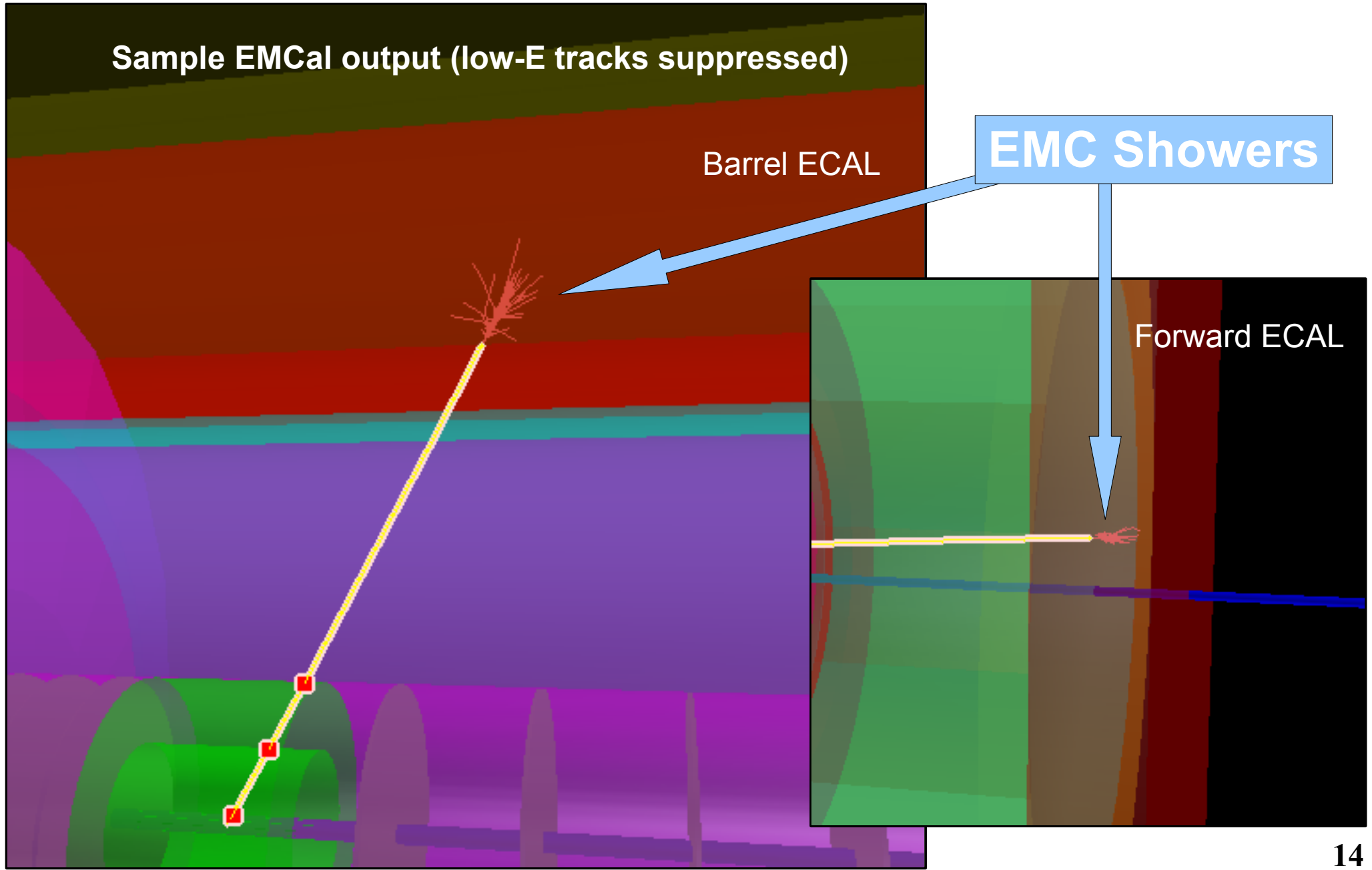
Electromagnetic Showers

Sample EMCal output (low-E tracks suppressed)

Barrel ECAL

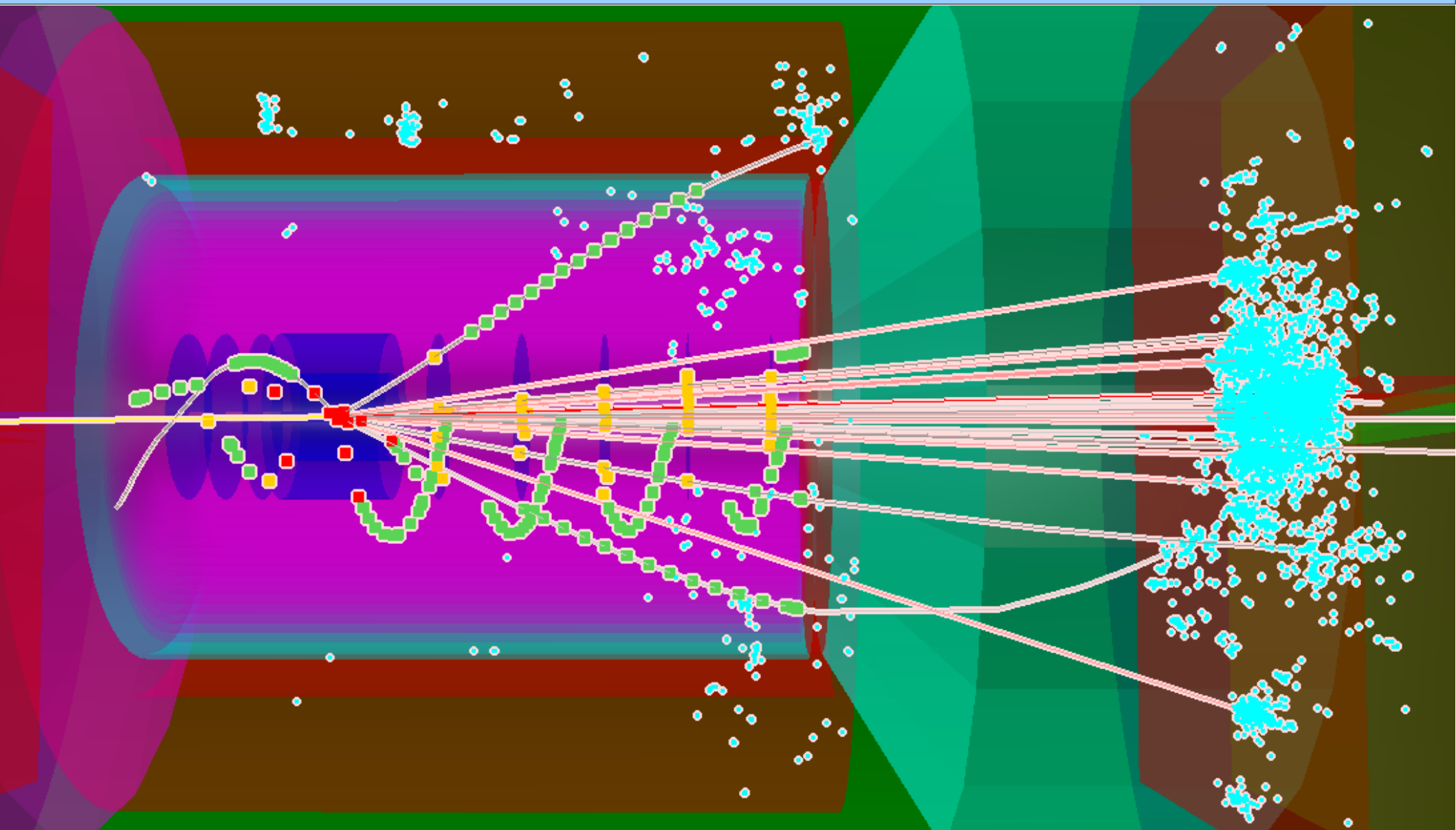
EMC Showers

Forward ECAL

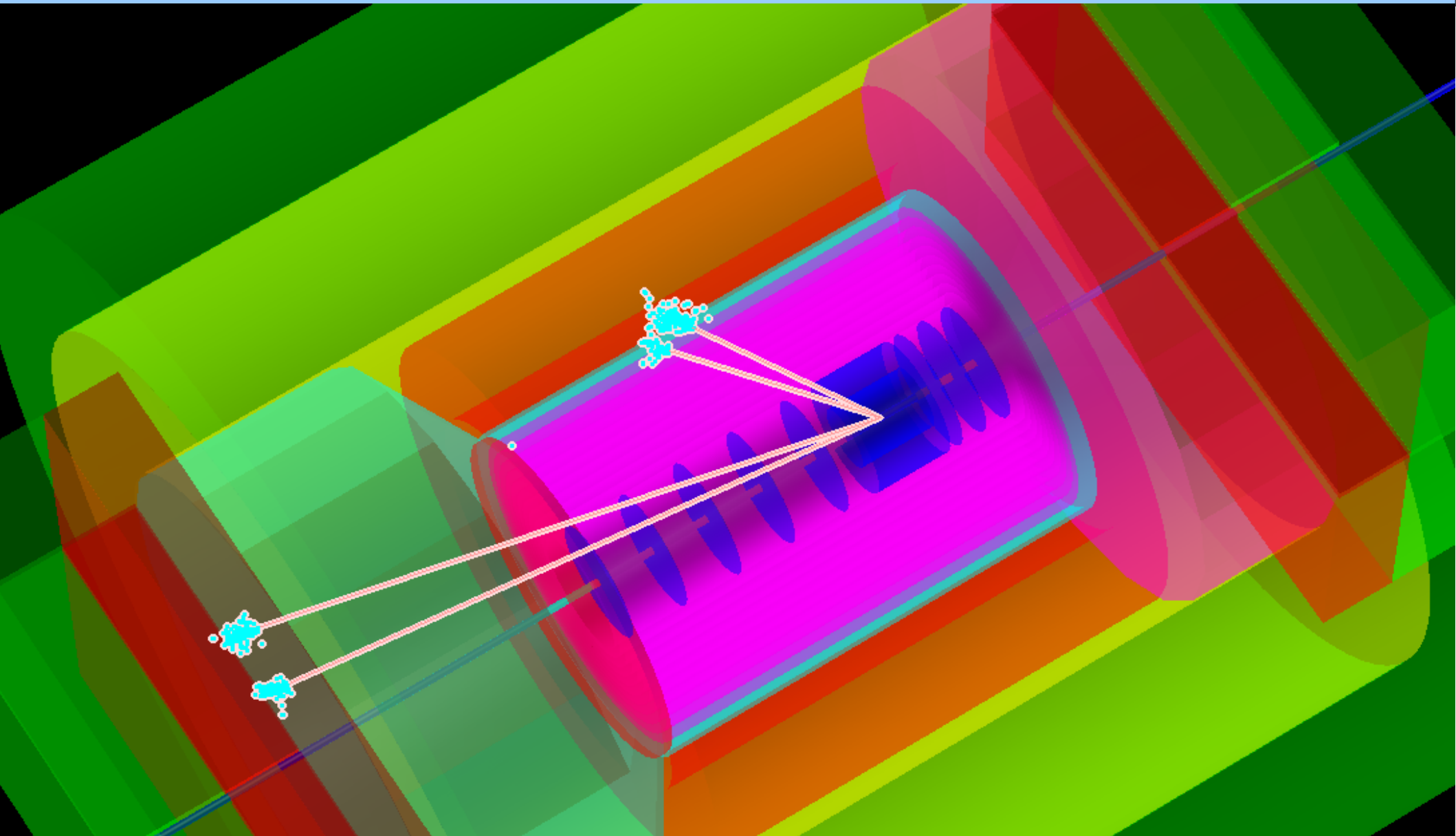


Event Generator: imported Pythia's event

HEPEVT/EIC format import: Pythia, RAPGAP, others..



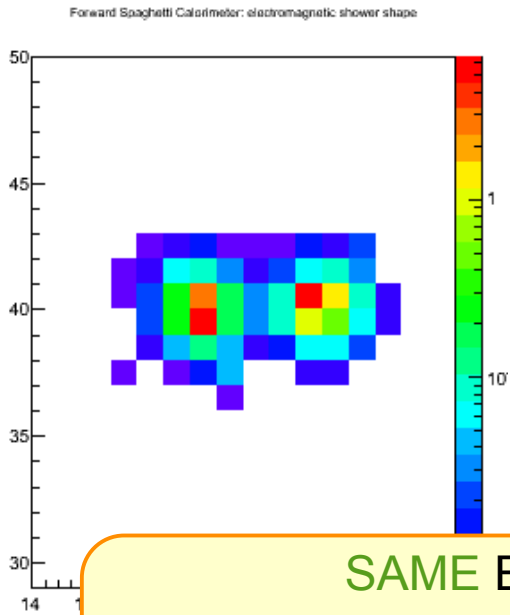
Two Pi^0 's, decayed by external decayer (Pythia)



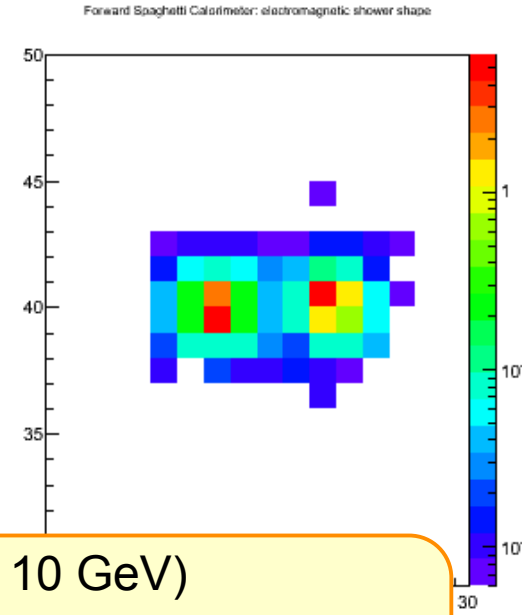
External decayer allows to force preferred branching ratios for desired particles...

EIC/eSTAR Forward Spaghetti Calorimeter: G3 vs G4

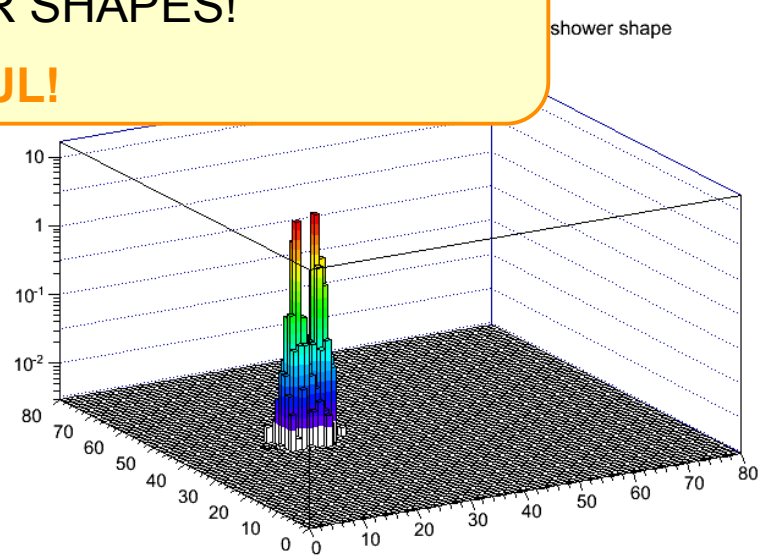
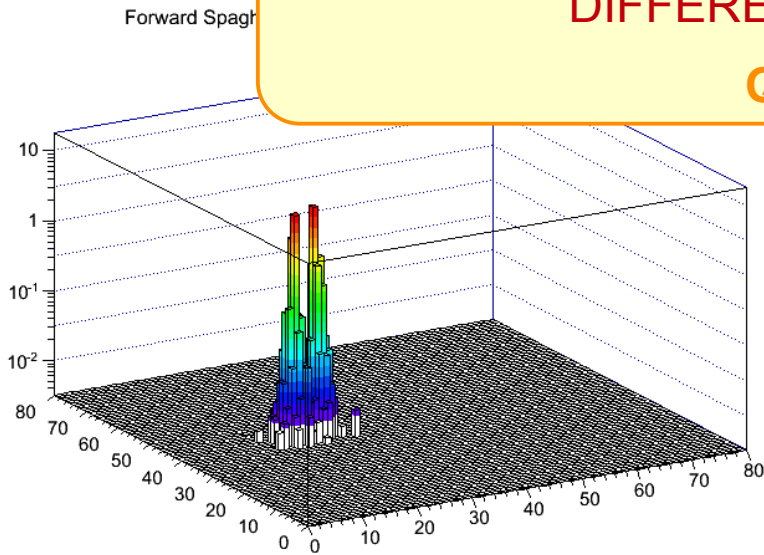
G3



G4



SAME EVENT (two gammas, 10 GeV)
SAME GEOMETRY (impl. as ROOT G3 vs NATIVE G4)
DIFFERENT SHOWER SHAPES!
QUITE USEFUL!



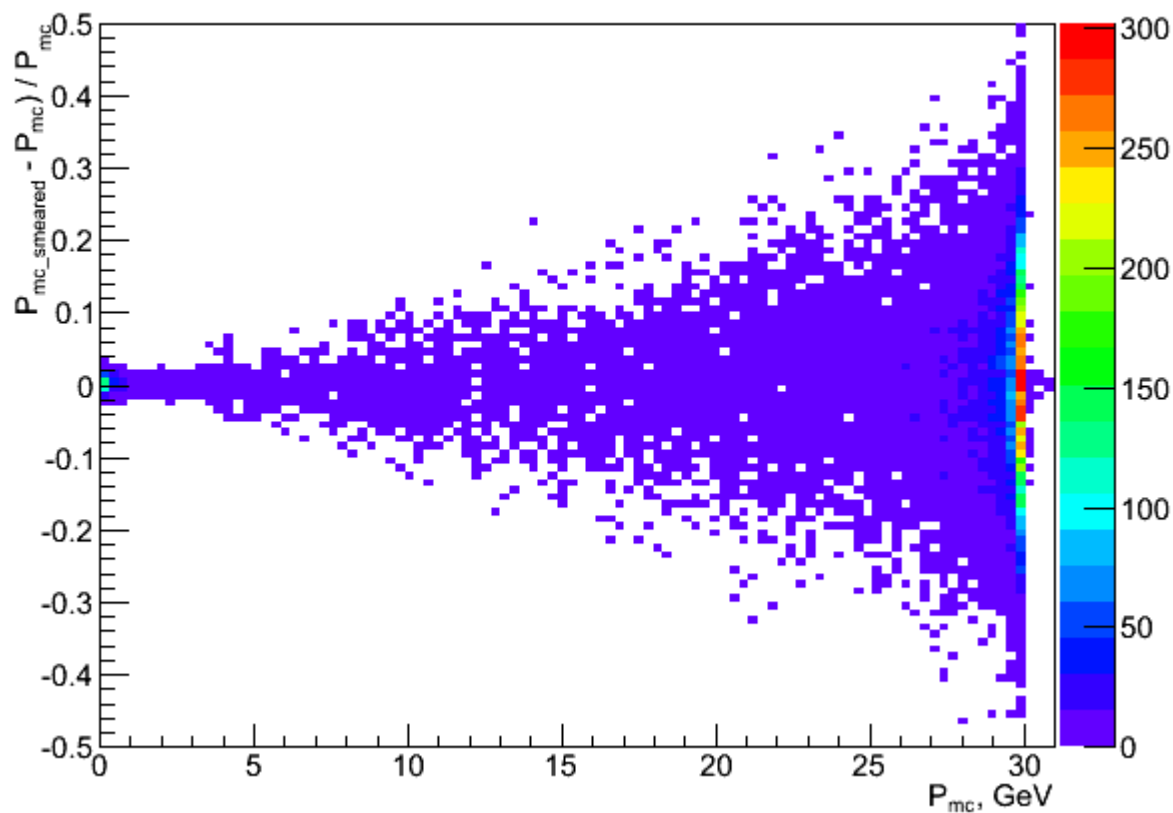
EICRoot: Fast Simulation

- **Initial version of the Fast Simulation is in place** (SIT, TRS, TPC, EMC, EEMC implemented, easy to extend);
- Fast Simulation **reads same VMC stack** to get particles, therefore **all** event generators are supported with no changes to be done to the Fast Sim codes;
- Fast Simulation may **use acceptance parametrization calculated from Full Simulation** (acceptance depends on B, P and detector params) or fast helix approximation for charged particles;
- Fast Sim and Full Sim **can run simultaneously** – easy to compare fast and full simulation results;
- Fast Sim produces additional branch in the output file, so **one can compare *Ev.Gen. data vs Fast Sim data vs Full Sim data* directly**;
- Fast Sim also works as a converter from ASCII to ROOT
- **Works fast!** Time to execute 1 event: 0.008056 s

Fast Simulation: P and Vtx smearing examples

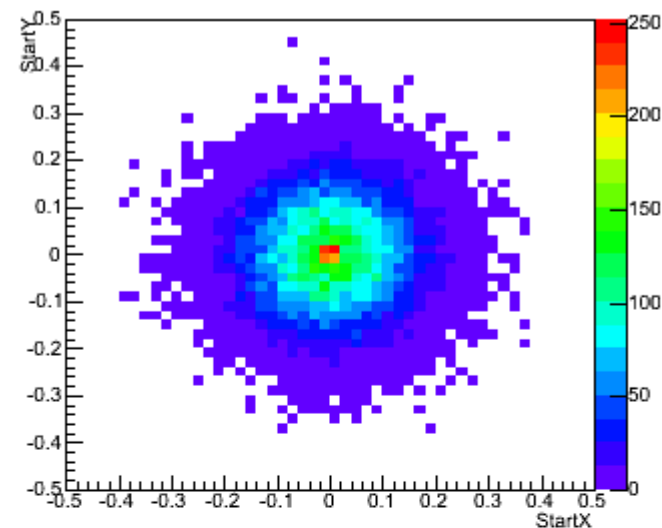
HEPEVT/EIC Pythia data (21k events), electron sample

FastSim momentum smearing (tracks from EIC Pythia)

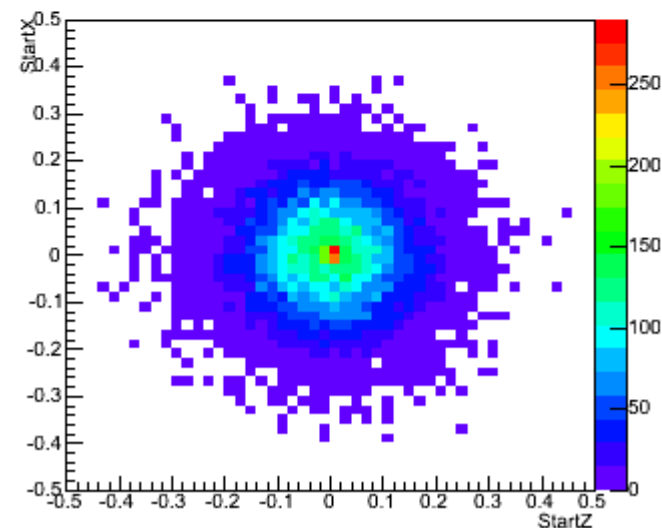


Track Momentum

FastSim vertex smearing X-Y (tracks from EIC Pythia)



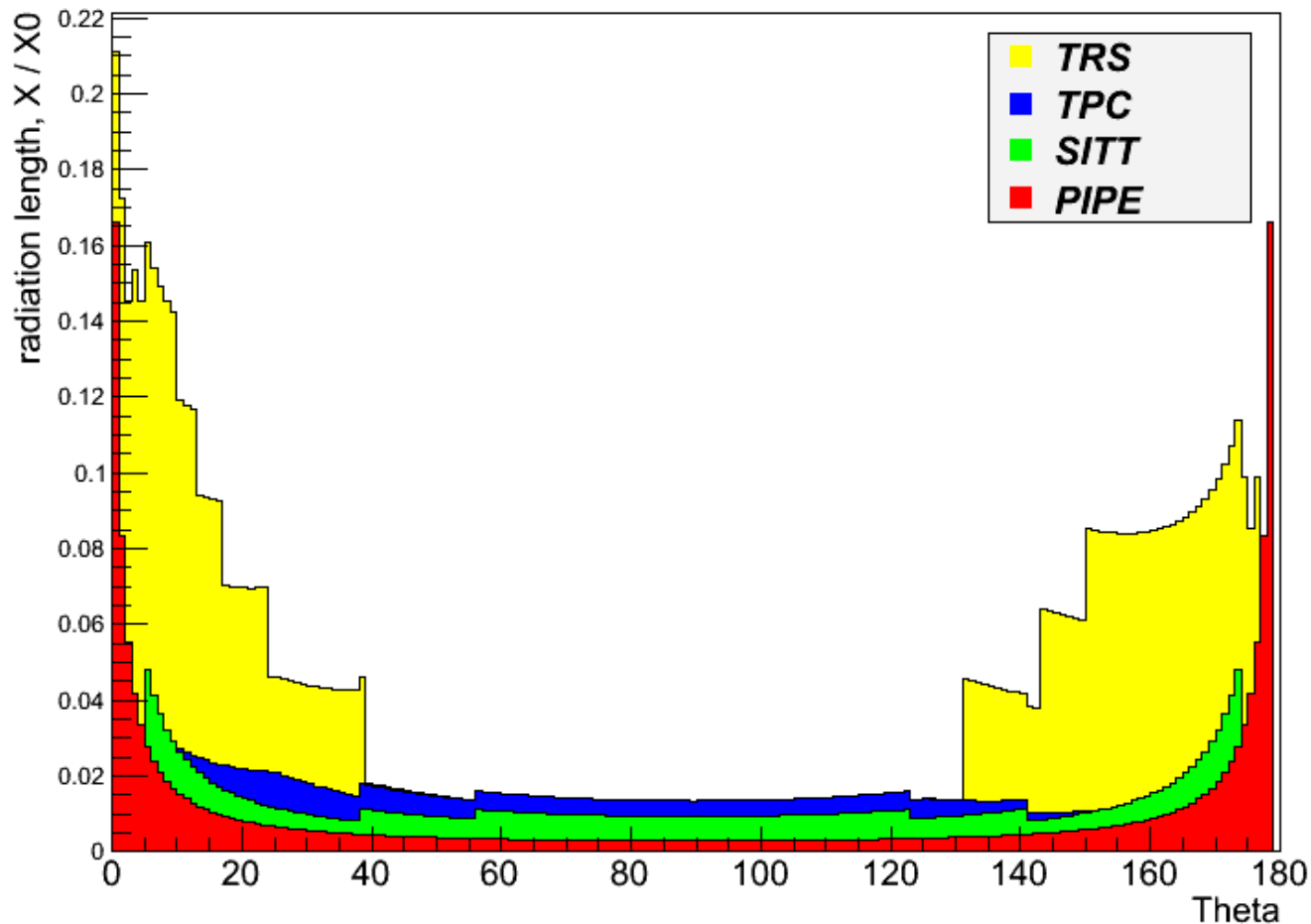
FastSim vertex smearing Z-X (tracks from EIC Pythia)



Track Start Vertex

Radiation Length Manager: tracker material scan

EIC Detector Geometry: Radiation Length Scan



Two options supported:

- ol' good "geantino"
- direct scan of ROOT TGeo

"geantino": time-consuming, but more efficient if native geometry issue is being investigated

"direct scan": exploits ROOT TGeo options directly, fast and flexible. It is a simple ROOT macro, which reads geometry and materials from MC output files.

Geometry has no dead material for now. Support/cooling/cabling will increase radlength..

EIC Magnetic Field

1. Current magnetic field is defined as const $B = 3\text{T}$ (easy to change), confined by the inner volume of EIC detector's solenoid;
2. If needed, we can try magnetic field maps from Panda (or STAR) as a first approximation..

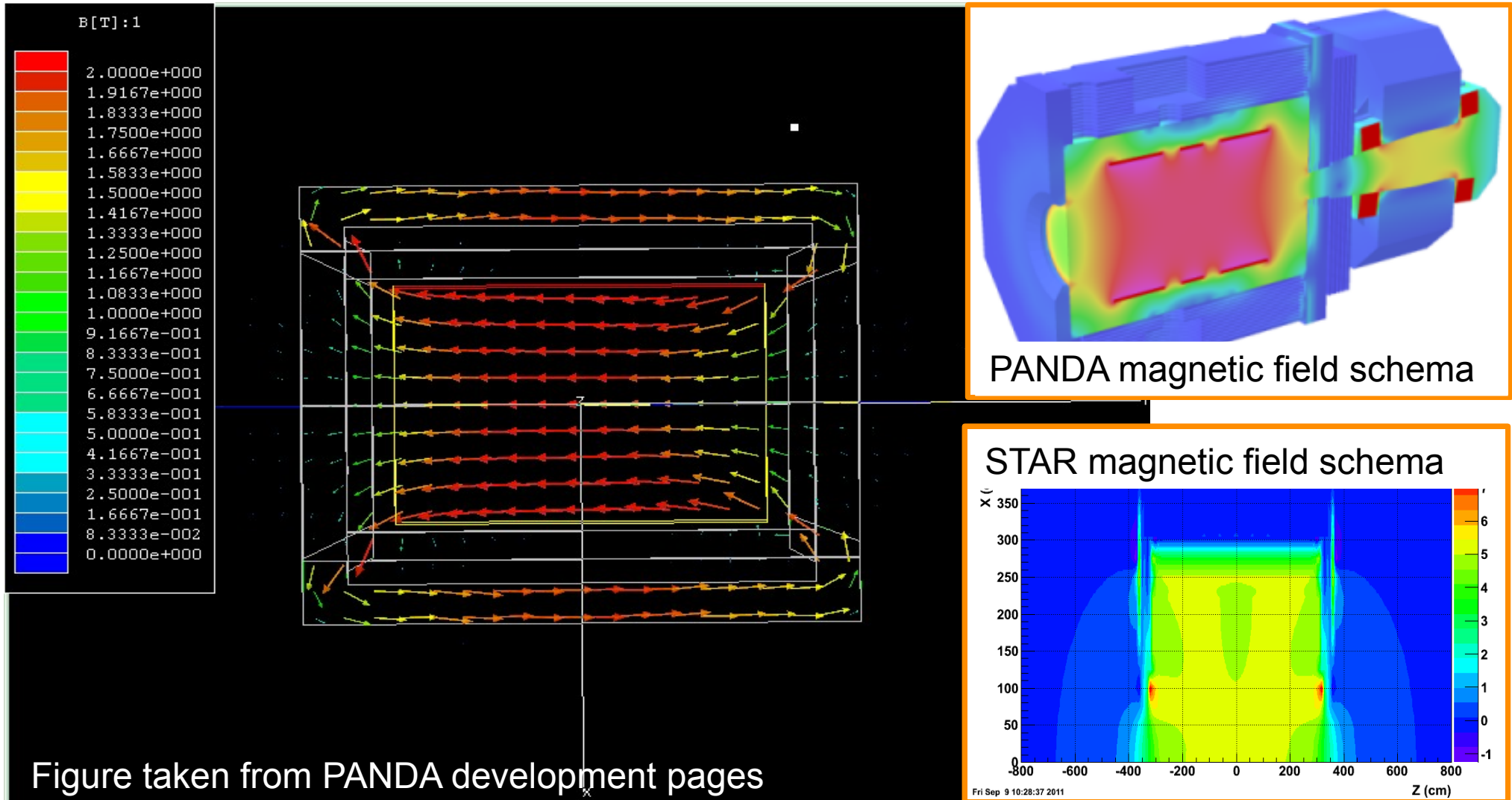


Figure taken from PANDA development pages

Summary and Outlook

- Proposal for fully featured **EIC Simulations Software Framework (EICRoot)**, based on existing FairRoot and ROOT packages, was presented.
- EICRoot has initial **implementation of EIC detector geometry**, based on public data tables, basic hits in all detectors are also implemented;
- EICRoot opens path forward to **detailed simulations and core concept verifications**, so we can make stronger case for Letter of Intent / CDR, utilizing newest software designed for HEP, using single software package.
- Well-designed underlying framework (FairRoot) allows rapid progress with simulations. Some basic checks were described in this presentation. Large user base allows **“borrowing” detector geometries from other experiments** for testing, with minimal effort.
- Distant Future: same framework supports **real data processing** chain, with no or minimal changes on user side compared to simulations. STAR event reconstruction experience allows to plan well ahead and to avoid costly pitfalls.

Thank You!

(check backup slides for more goodies)

Backup Slides

Geometry Description example

```
//*****
fsce01#1
cave
BOX
WSciMix
12.500000 -12.500000 -450.000000
12.500000 12.500000 -450.000000
-12.500000 12.500000 -450.000000
-12.500000 -12.500000 -450.000000
12.500000 -12.500000 450.000000
12.500000 12.500000 450.000000
-12.500000 12.500000 450.000000
-12.500000 -12.500000 450.000000
-487.500000 -987.500000 7617.000000
1.000000 0.000000 0.000000 0.000000 1.000000 0.000000 0.000000 0.000000 1.000000
//*****
fsce01#2
cave
-487.500000 -962.500000 7617.000000
1.000000 0.000000 0.000000 0.000000 1.000000 0.000000 0.000000 0.000000 1.000000
//*****
fsce01#3
cave
-487.500000 -937.500000 7617.000000
1.000000 0.000000 0.000000 0.000000 1.000000 0.000000 0.000000 0.000000 1.000000
//*****
```

VMC and **VGM** for simulation:

- Running different transport MC's from the same application
- No need to customize detector response, physics configuration and cuts (partially) or input/output according the transport code
- Geometry is described once and then one can choose between different MC's and
- different navigations: e.g:
 - G3 Native geometry and navigation
 - G4 Native geometry and navigation
 - G4 Native geometry and Root navigation
 - G4 Root geometry and navigation

....

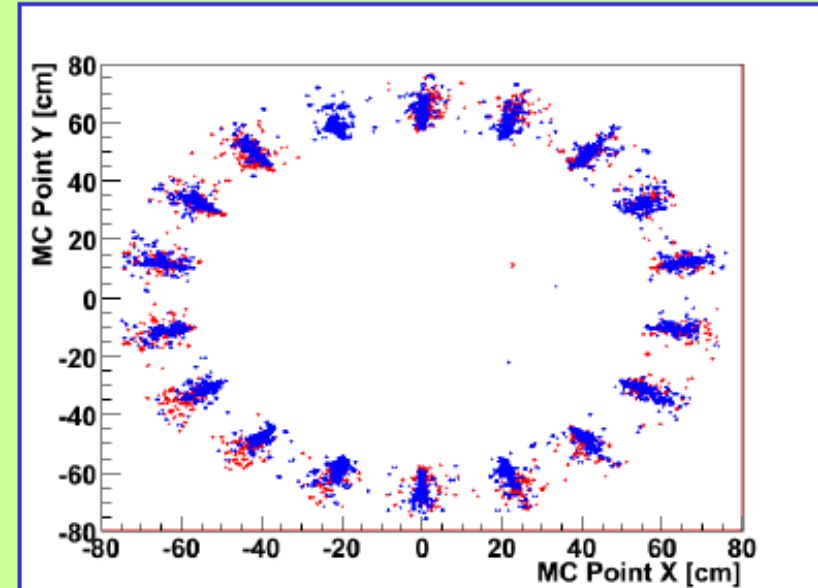
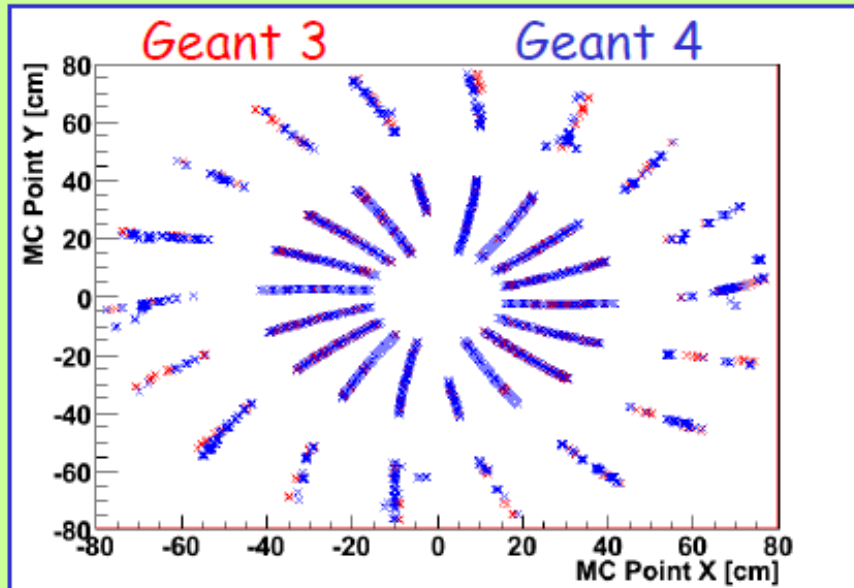
Transport Models Comparison

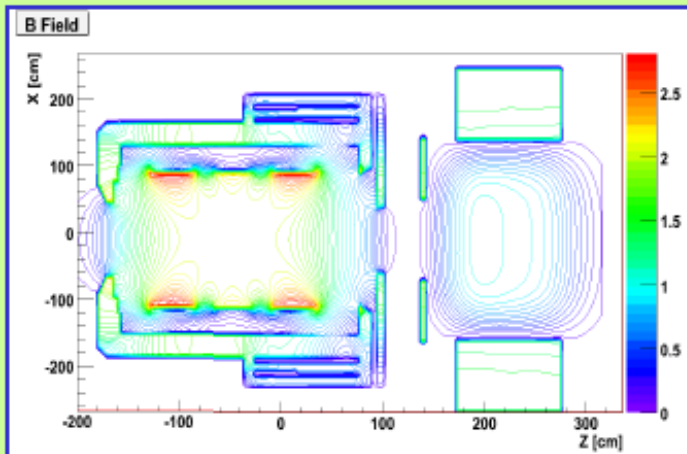


With exactly the same code
Changing only one flag

$\mu^- @ 1 \text{ GeV}/c$

$\gamma @ 1 \text{ GeV}/c$





Global Tracking

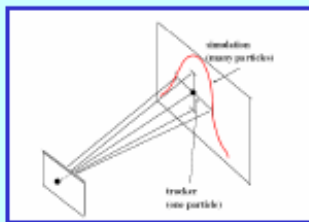
- How to have a high resolution tracking?
- How to merge different detectors?
- How to deal with field inhomogeneity?

Global tracking

Kalman Filter

genfit

Already working
for TPC



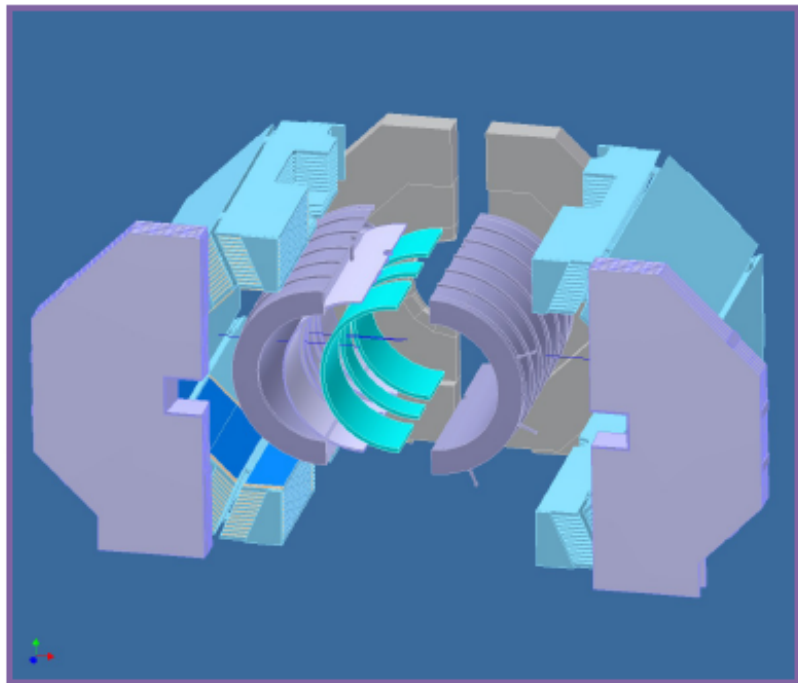
Track Follower

GEANE

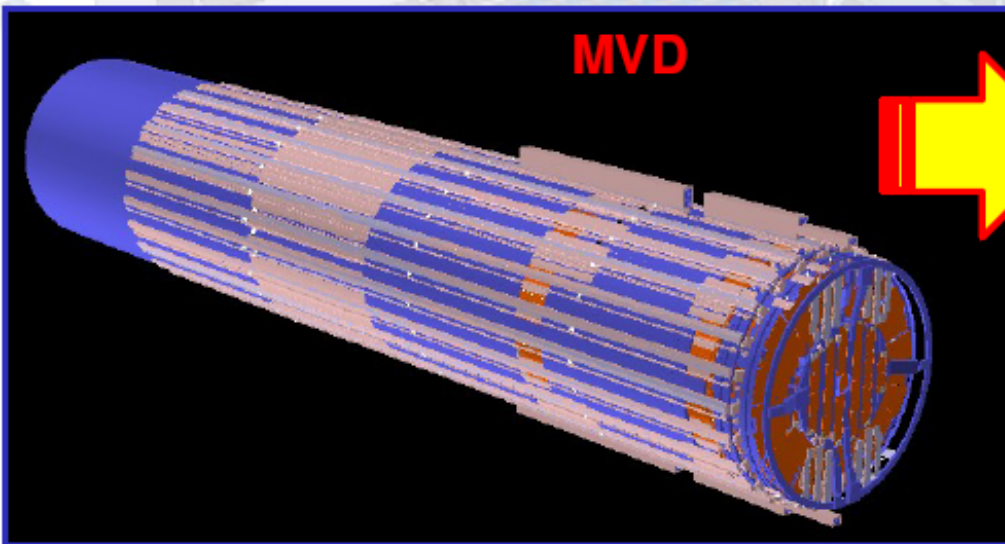
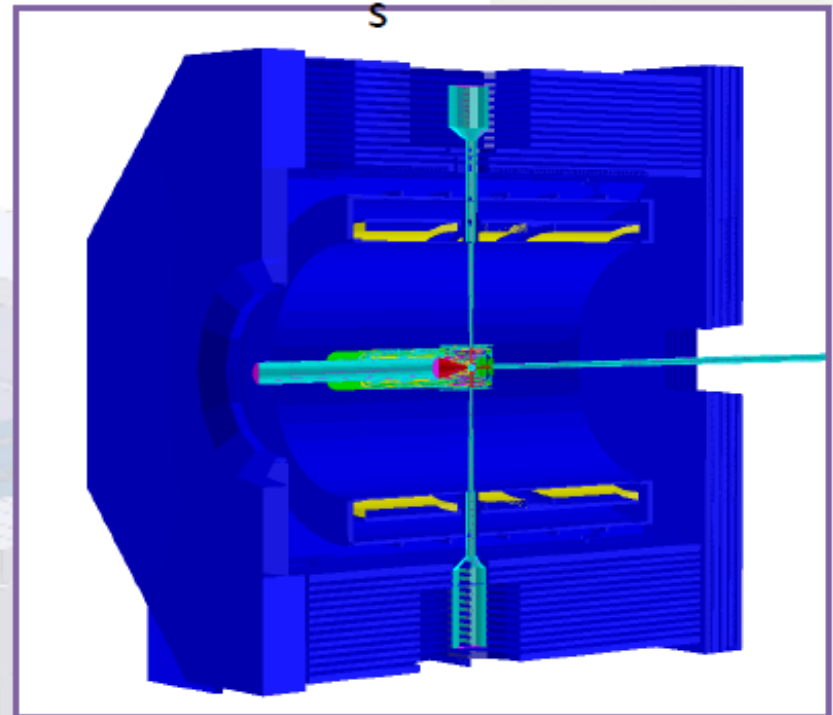
see A.Fontana
Poster session

Use of the same geometry for simulation and track following

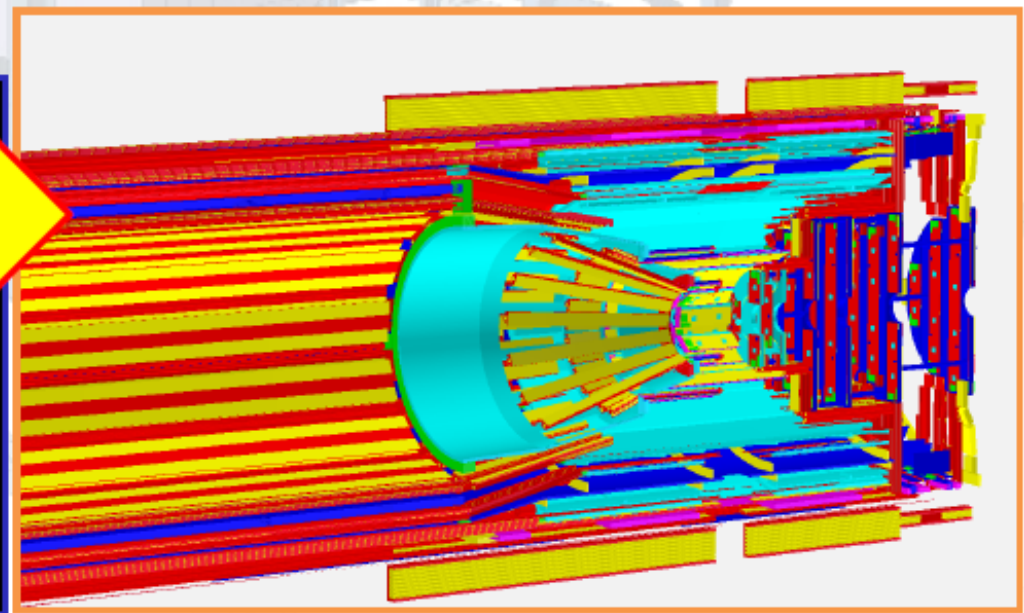
!



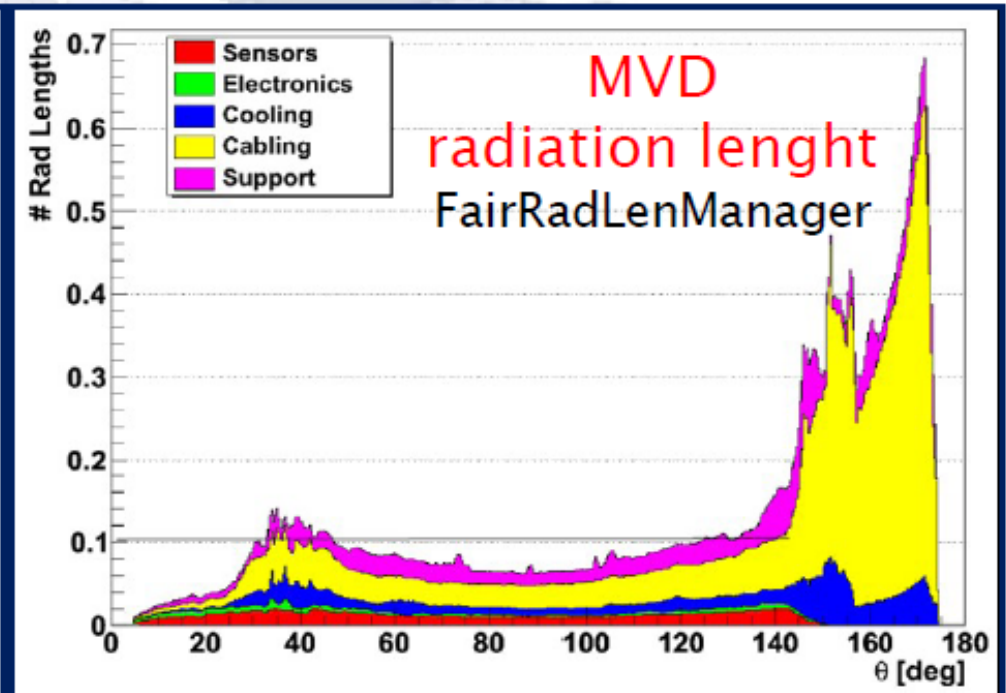
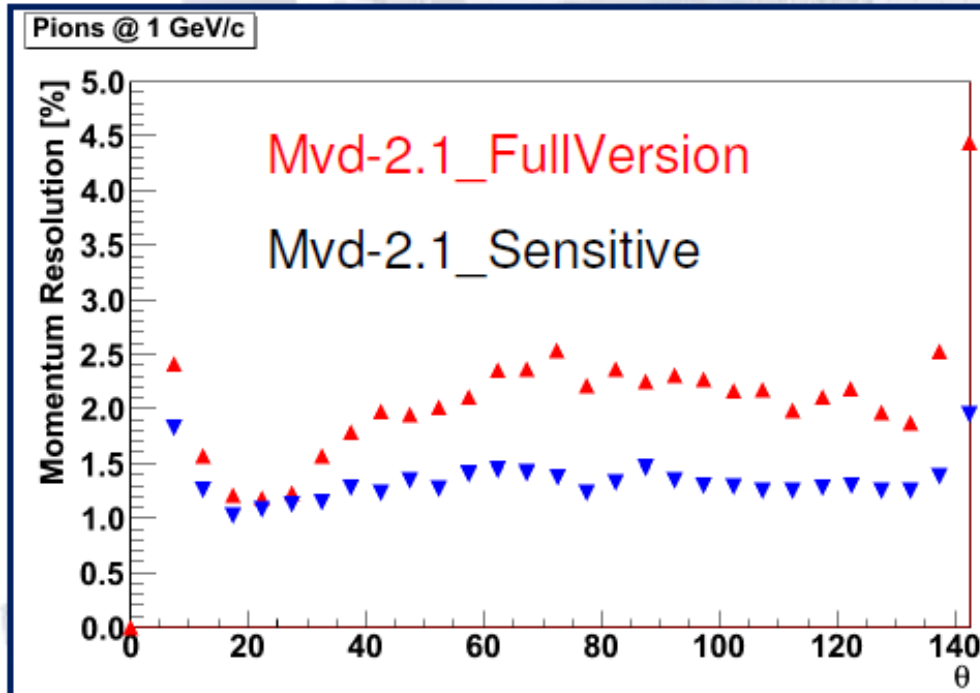
Solenoid
Yoke



MVD



Tracking: Performances (TPC + MVD + GEM)

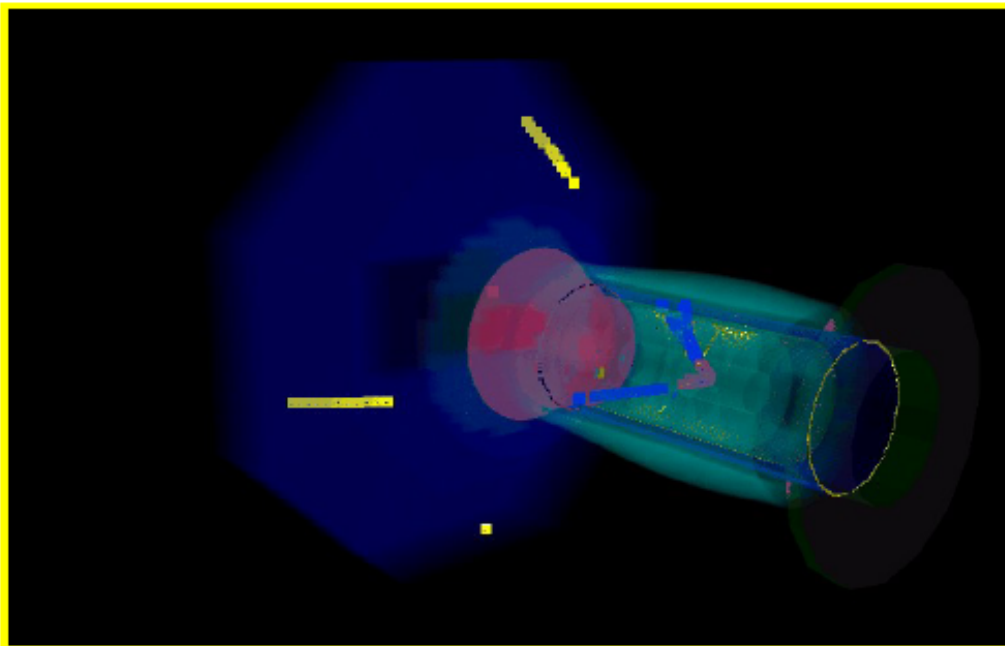


resolution $\times 2$ due to passive materials

full geometry description
very important



CadToRoot converter



Event Display \Rightarrow TEve

$X(3872) \rightarrow J/\psi \pi^+ \pi^-$

$J/\psi \rightarrow \mu^+ \mu^-$

$J/\psi \rightarrow e^+ e^-$

