Central Detector Integration software suite

(aka EIC Toy Model)

Alexander Kiselev

EICUG Tracking WG Meeting June 25 2020

An attempt to connect some of the dots

Escalate & fun4all; migration process

Tracker, PID & Calorimetry detectors in GEANT

1-st & 2-d IR

EIC detector & greenfield solenoid design

Physics simulations & engineering design

Ideal detectors & services / support

$|\eta|<4.5$ & reality

Space available for detectors & IR vacuum chamber

- One can easily identify a number of places with a lack of sync at this early stage
- Some of them can seemingly be addressed in a more or less consistent way

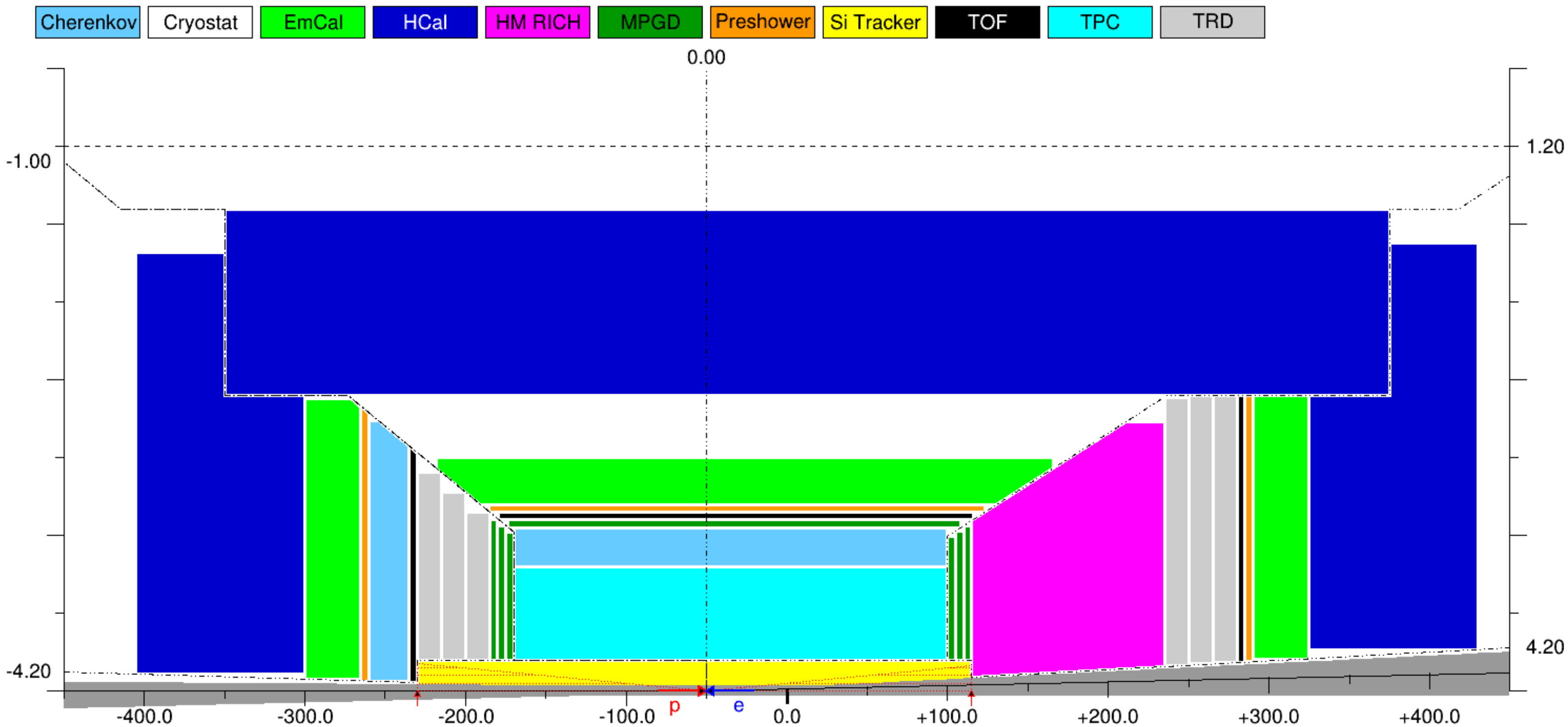
EIC Toy Model: overview

- **A tool to model & generate EIC Central Detector "templates" in a way:**
  - the new geometries (models) can be generated "quickly" …
  - … and represented instantly in a WYSIWYG fashion
  - the sub-detector "container objects" are guaranteed to not overlap either with each other or with the IR vacuum chamber elements
  - technically they can be imported in GEANT frameworks in a consistent way and used as wrappers to the "real" sub-detectors
  - they can be exported in a CAD format to be used in the engineering design of the detector support structures and / or laying out services

- **Repository:** https://github.com/eic/EicToyModel
  - a README file
  - example ROOT scripts
  - a standalone GEANT example
  - detailed API description
  - *Currently neither g4e nor fun4all examples available*

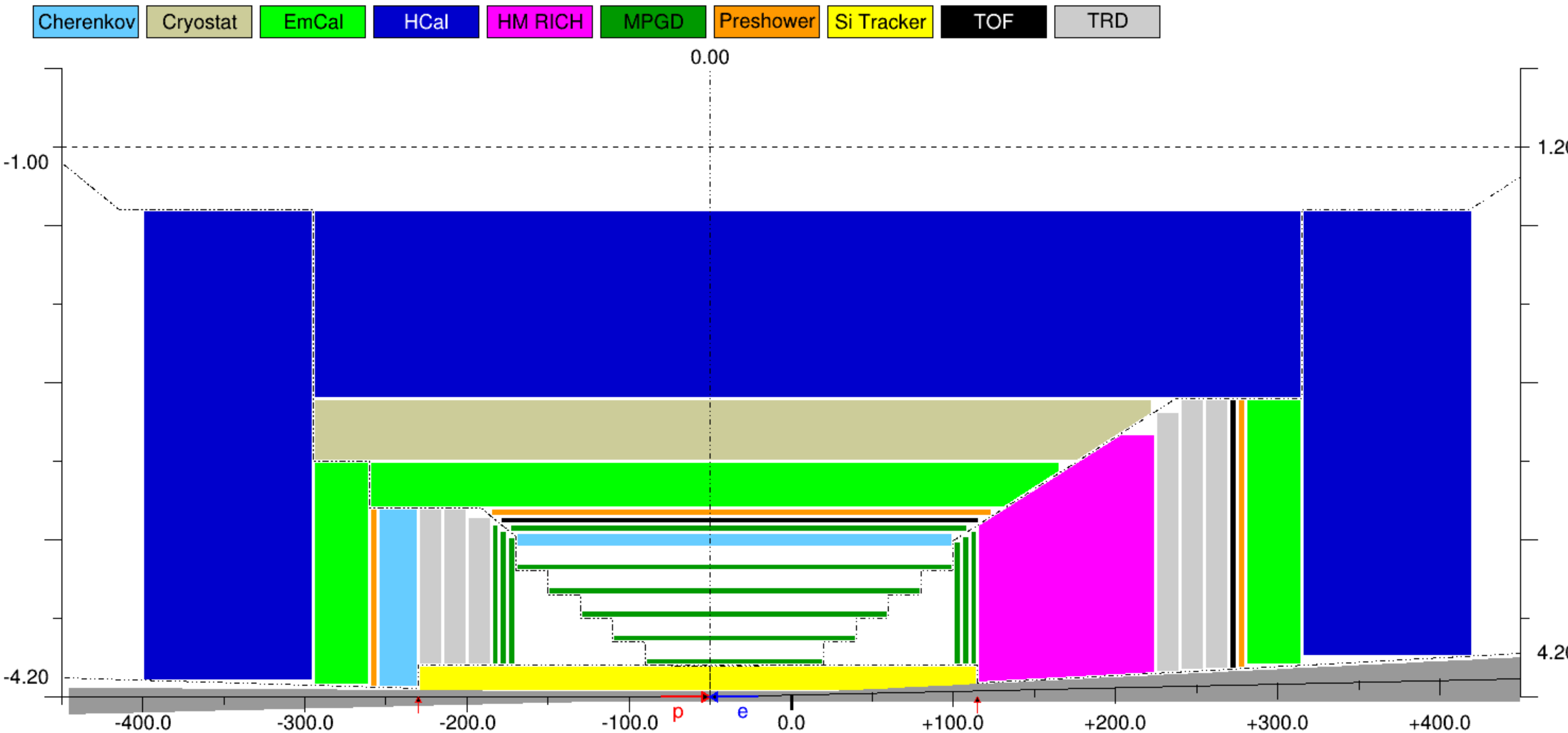
Suggested workflow

Create a model → **Save it as a .root file** → **Import in GEANT**



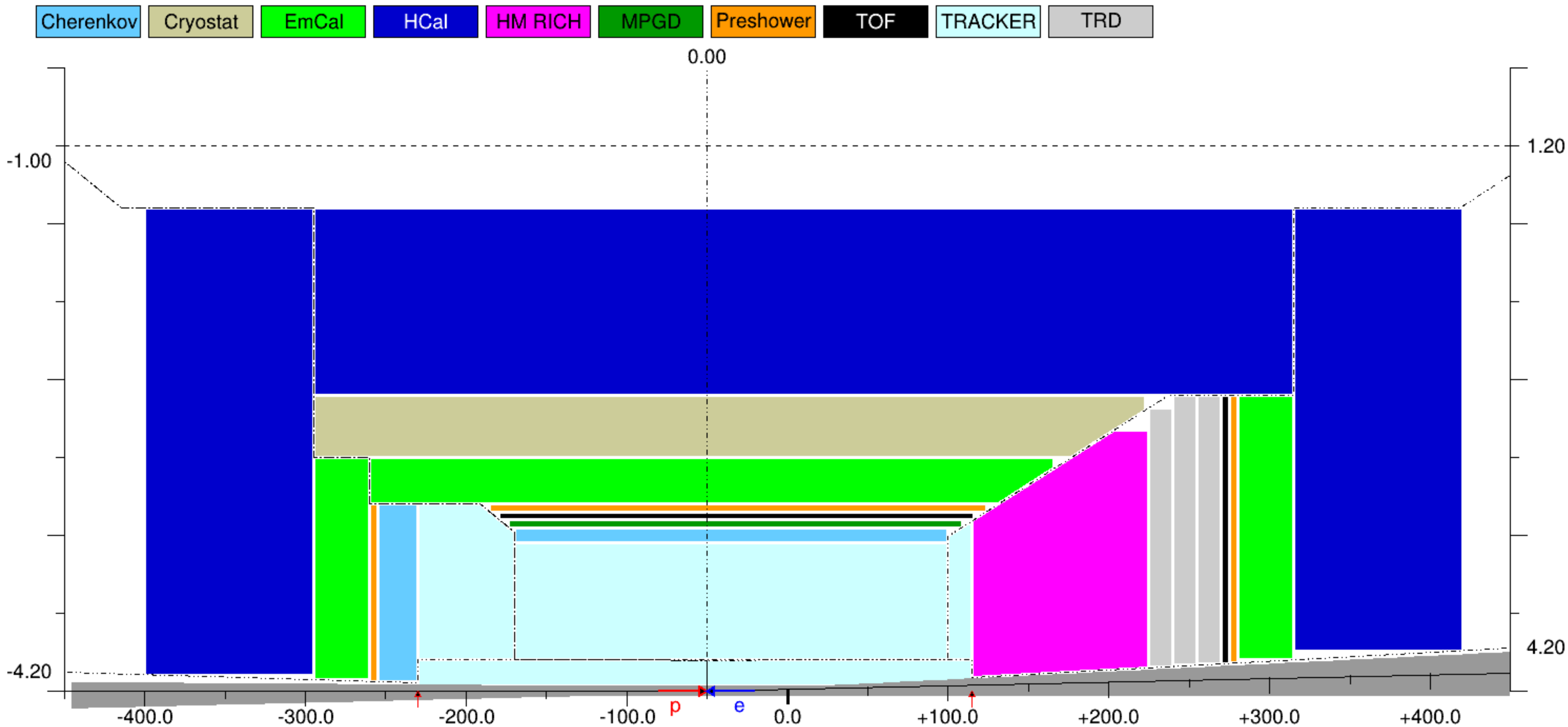
- Minimal overhead to create a 2D scheme like this (ROOT scripting)
- Model can be saved, distributed and re-imported as a .root file
- GEANT application: import .root file and **create volumes on the fly**
  - Alternatively: export and import GDML file(s) *(not yet implemented)*

Integration volume granularity: tracker



- **Detector grouping is certainly possible**
  - Is it flexible enough?
  - As shown here: too detailed at this early stage?
  - https://github.com/eic/EicToyModel/blob/master/scripts/tracking.C

Integration volume granularity: tracker



- **Detector grouping is certainly possible**
  - Is it flexible enough?
  - Allocate larger volumes for PID / Tracking / Calorimetry, to start with?
  - https://github.com/eic/EicToyModel/blob/master/scripts/tracking.C

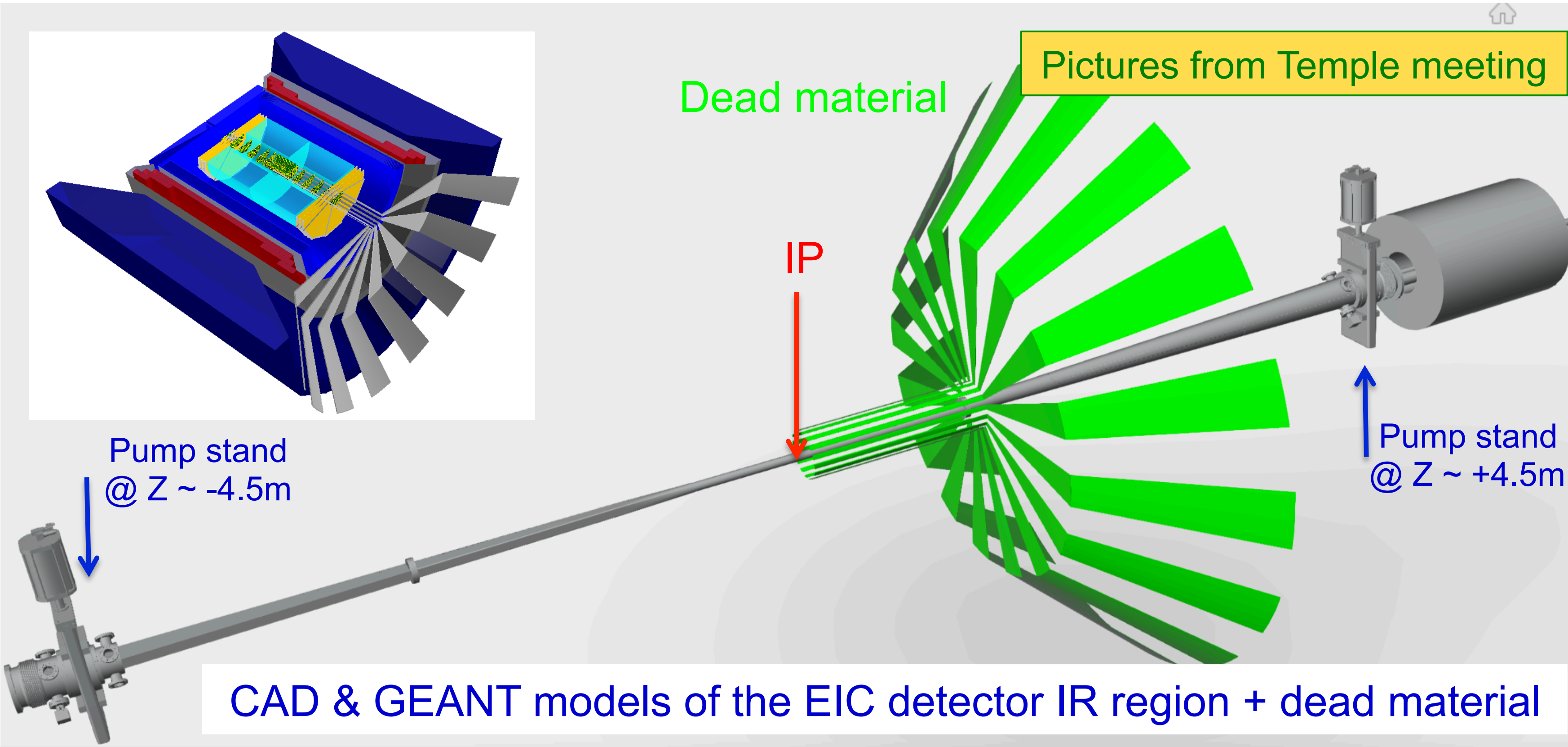
Limitations in the geometry description

- **Four pre-defined detector "stacks": vertex, barrel, and two endcaps** …
- … **in a projective configuration (defined by the η ranges)**
- **Detector volumes in the endcap stacks are placed as strictly aligned objects with flat front and rear sides, one after the other**
  - ... although stack boundaries can be shaped up creatively, if needed

- **Detector tags (like "EmCal") and respective colors are hardcoded …**
  - … though custom ones can be generated dynamically, if really needed

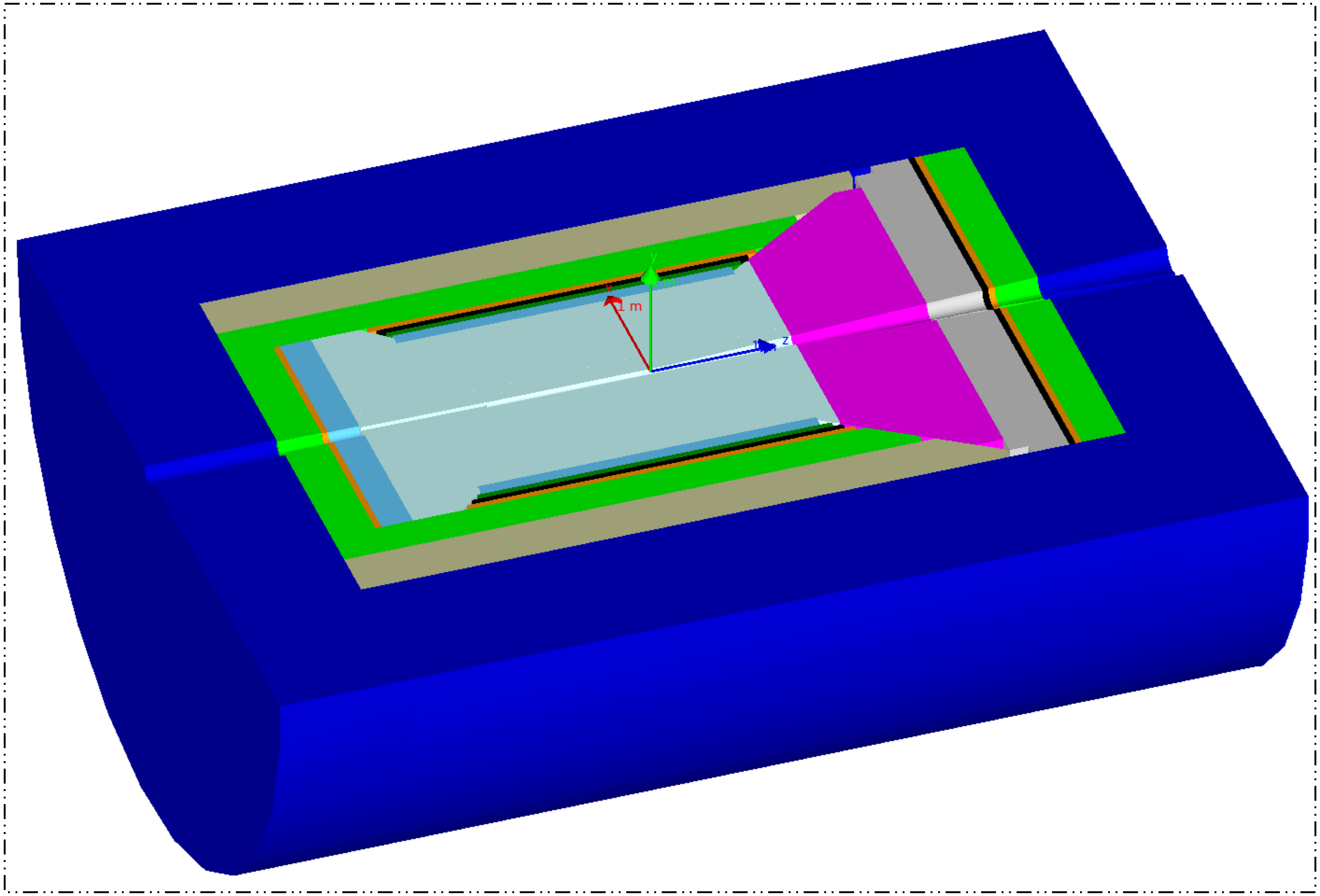
- **Exported objects are azimuthally symmetric Polycones, although …**
  - … with an asymmetric cutaway representing the IR vacuum chamber
- **Polyhedra export implemented, but can not be mixed with Polycones**
- **CAD export: presently Polycones only; no vacuum chamber cutaway**

Services



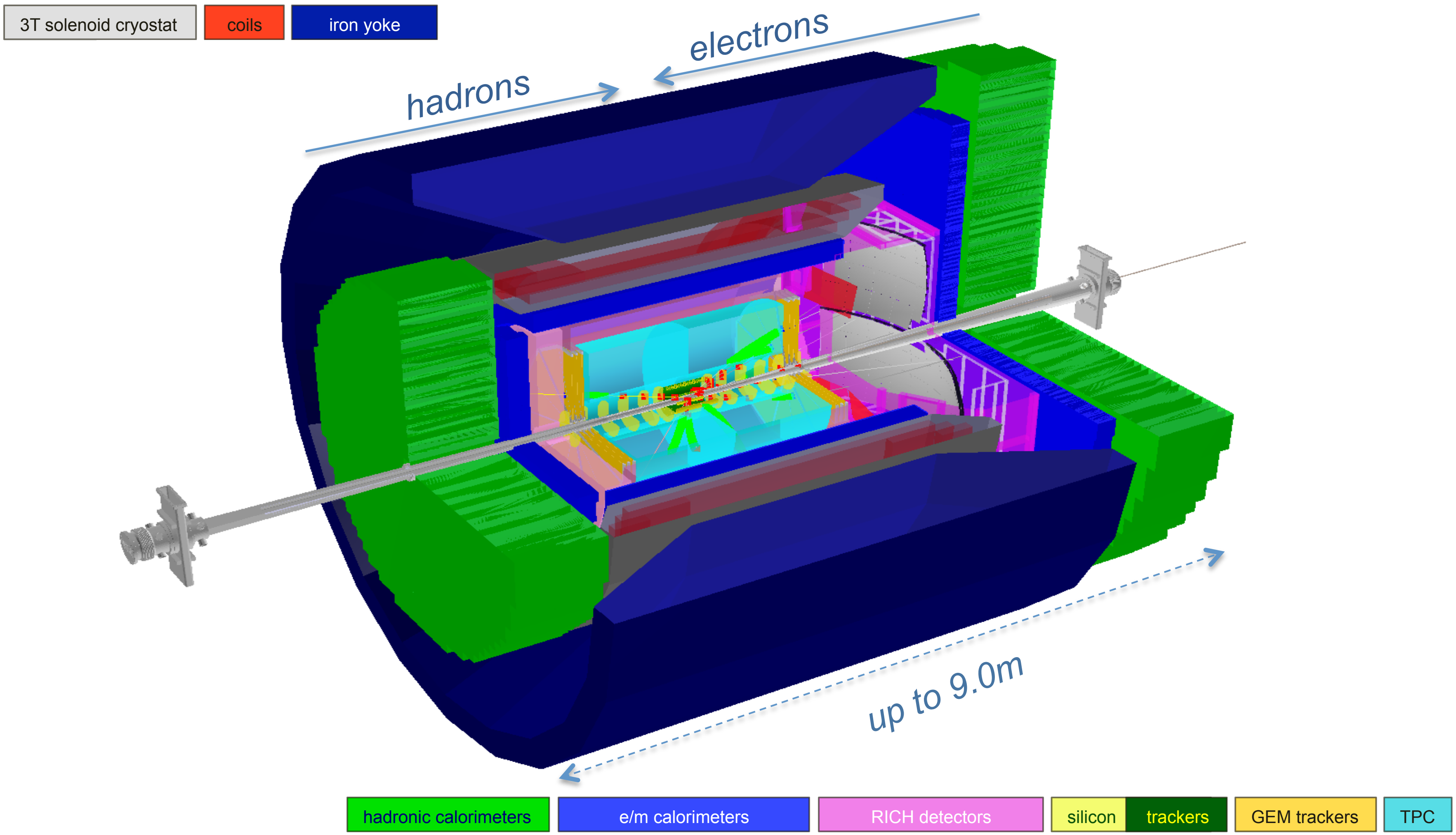
- Barrel-to-endcap cracks (support, services):
  - Obviously affect the available space for the detectors
  - Should be configurable, accumulating services from / to “inner” detectors
  - No progress since Temple; *almost the first item on the TODO list*

GEANT interface (Qt event display)



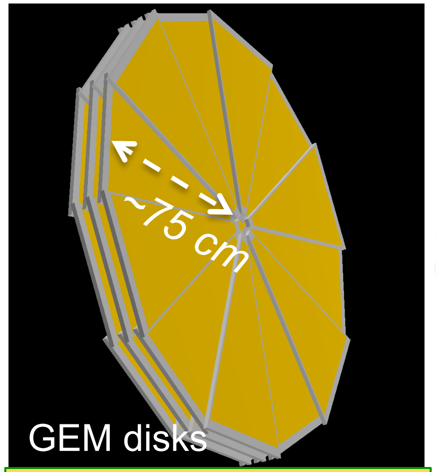
- Volumes are currently generated on the fly *(is GDML step really needed?)*
- Once imported, the layout will look the same **in all G4 applications**

Compare: BeAST EicRoot implementation

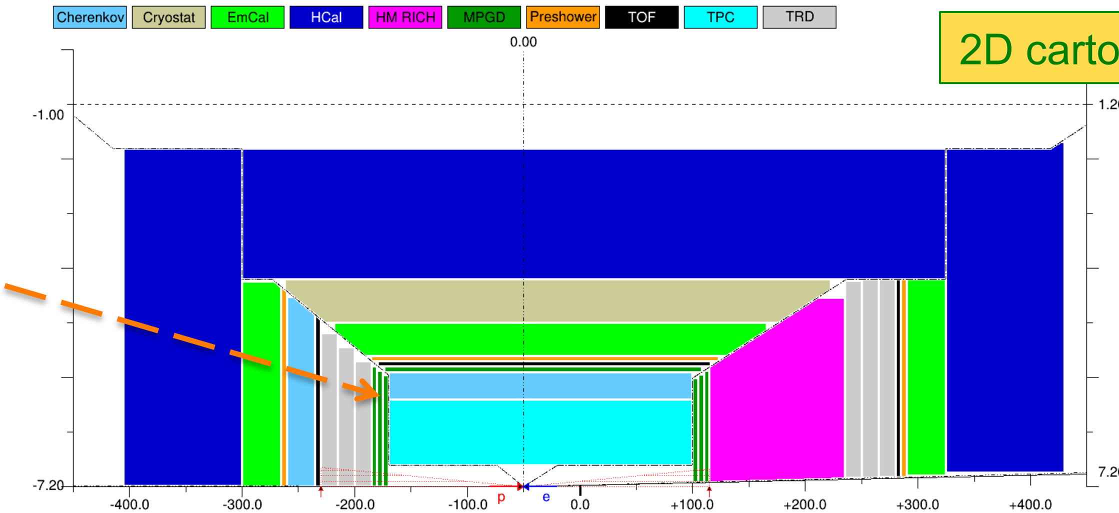


- Comment#1: some of the volumes here (PID) are also air balloons
- Comment#2: one can seemingly reuse TGeo objects in the new scheme

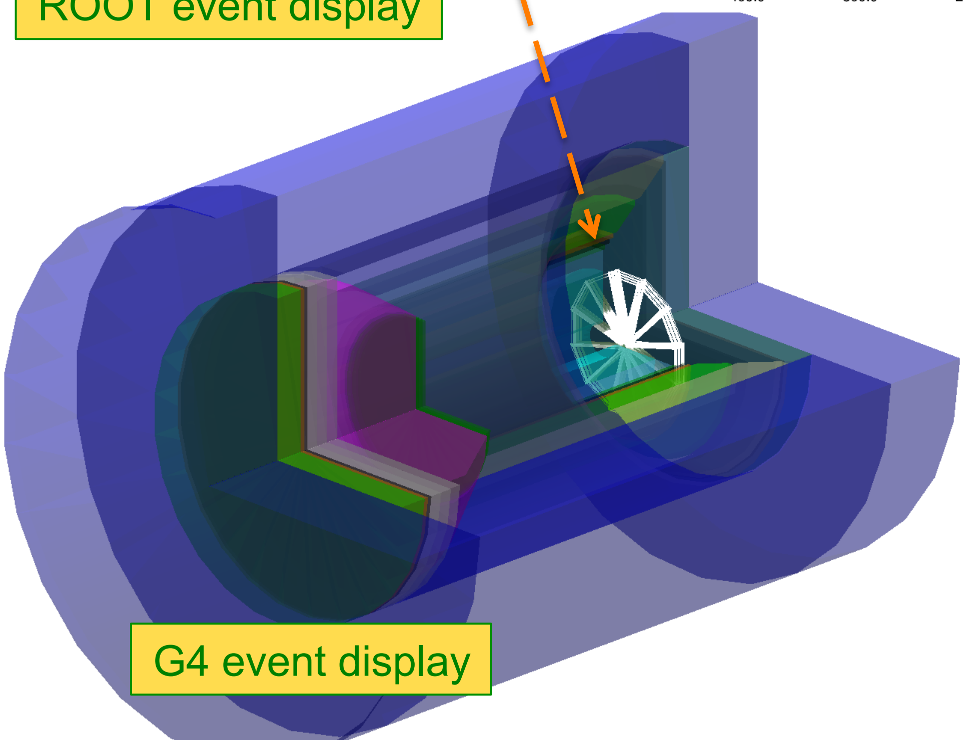
EicRoot geometry import



ROOT event display



2D cartoon



G4 event display

- Yet *experimental,* but seems to work, as expected
- Possible other candidates: MM barrel, silicon vertex, calorimetry
- Material information merging from different files may be an issue

Coding overhead in GEANT

Excerpt from a modified working calorimetry code:

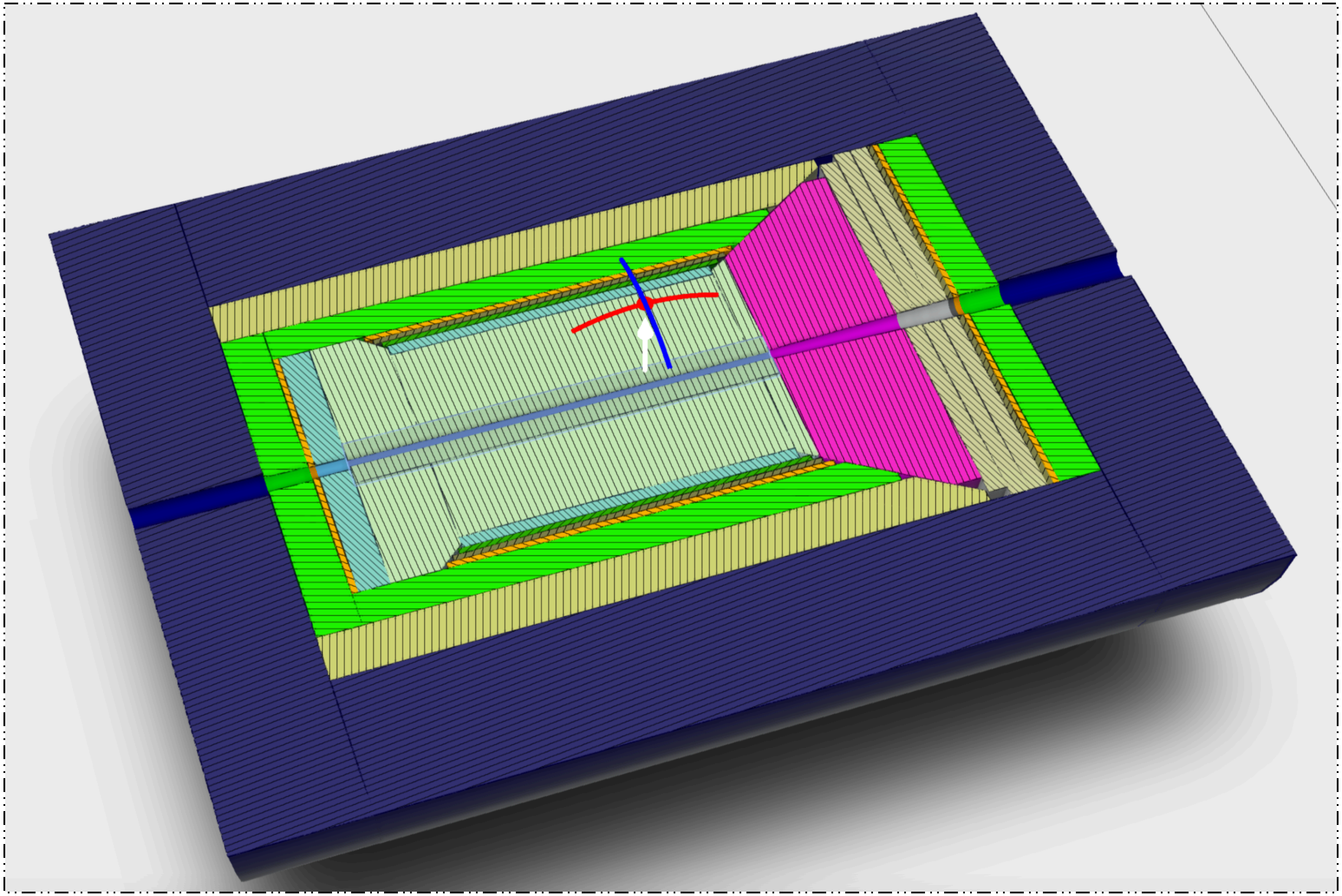
```
// Construct the integration volumes geometry, internally;
TFile fin(argv[1]);
dynamic_cast<EicToyModel *>(fin.Get("EicToyModel"));
eic->Construct();
// Populate G4 world by these volumes;
eic->PlaceG4Volumes(expHall_phys);

// Place "MyHCal" tower matrix into the integration volume bubble instead of the world;
new G4PVPlacement(0, G4ThreeVector(0, 0, zOffset), myhcal_log, "MyHCal", expHall_log,    false, 0);
auto hcal_bubble_log = eic->fwd()->get("HCal")->GetG4Volume()->GetLogicalVolume();
new G4PVPlacement(0, G4ThreeVector(0, 0,       0), myhcal_log, "MyHCal", hcal_bubble_log, false, 0);
```

This part should be taken care of by the framework

- Immediate migration is not mandatory for everybody
  - Integration bubbles can be imported into a framework one by one
- Bubble size (and location) can be polled (*not yet; coming soon*)
  - Parametric detectors can be implemented in a proper way
- If the community prefers to use GDML files instead, so be it (consistency?)

CAD interface (3D model in Autodesk viewer)



- Obviously looks identical to the GEANT picture
  - Services and support structure engineering design can start off the same configuration as used in GEANT

Magnetic field map interface

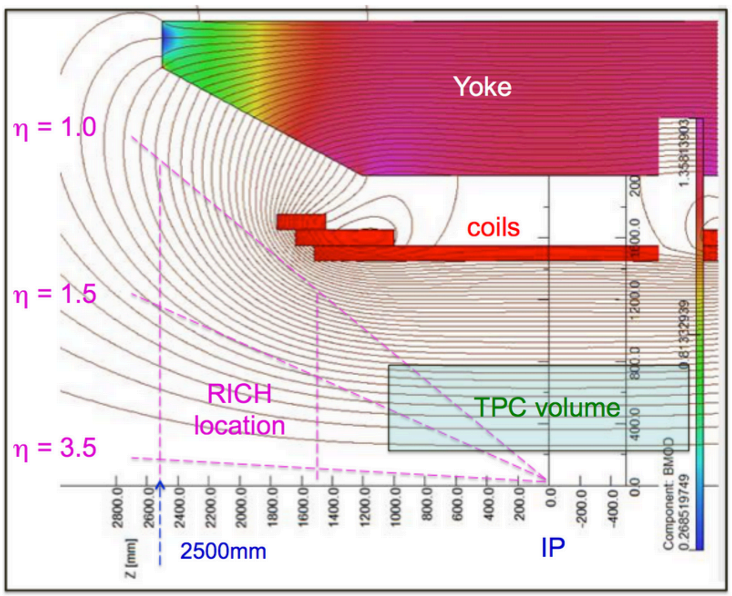
https://eic.github.io/software/beast_magnetic_field.html

Computing | Software | Working Group | Information | About

BeAST solenoid magnetic field map

The repository contains an ASCII file with the field map, a C++ class to handle it and a GDML model

https://github.com/eic/BeastMagneticField



Open solenoid design (no field clamps)

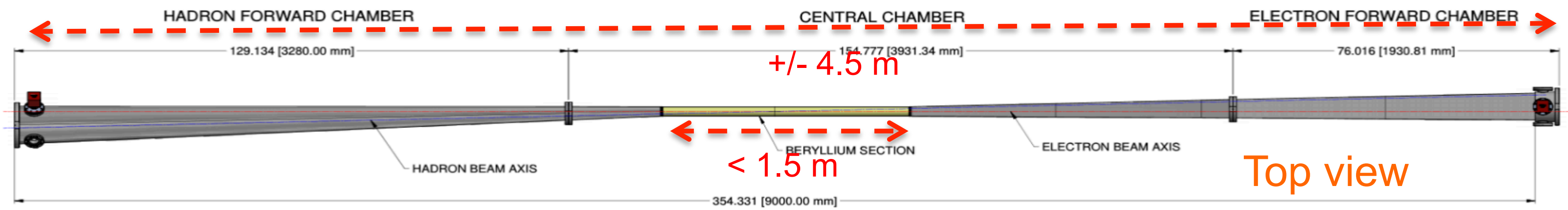
Homogeneous (less than than 4% variation) 3T central field in the TPC volume

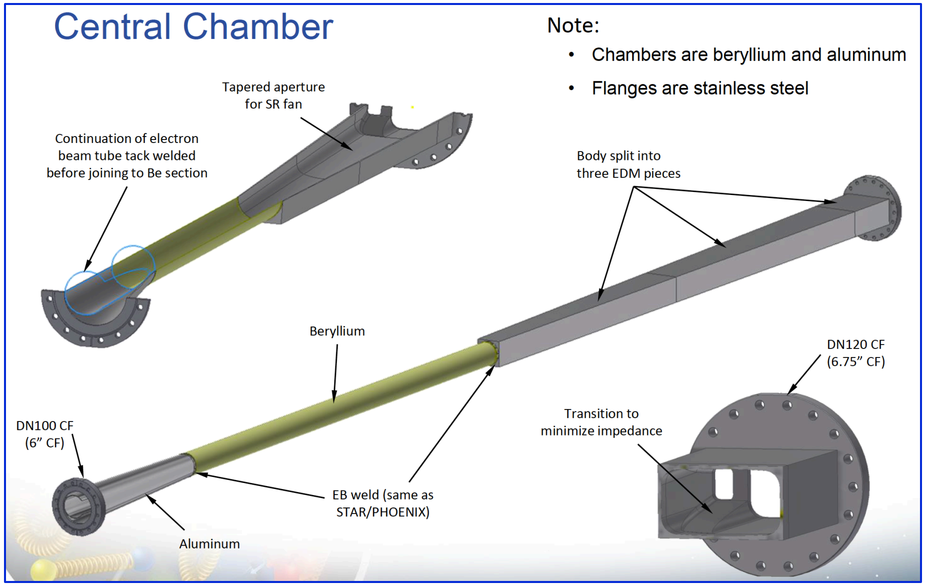
Fringe field is tuned in order and minimize charge particle bending in the forward gaseous RICH volume (less than 1mrad RMS for 10 GeV/c particles up to 25 degree polar angles)

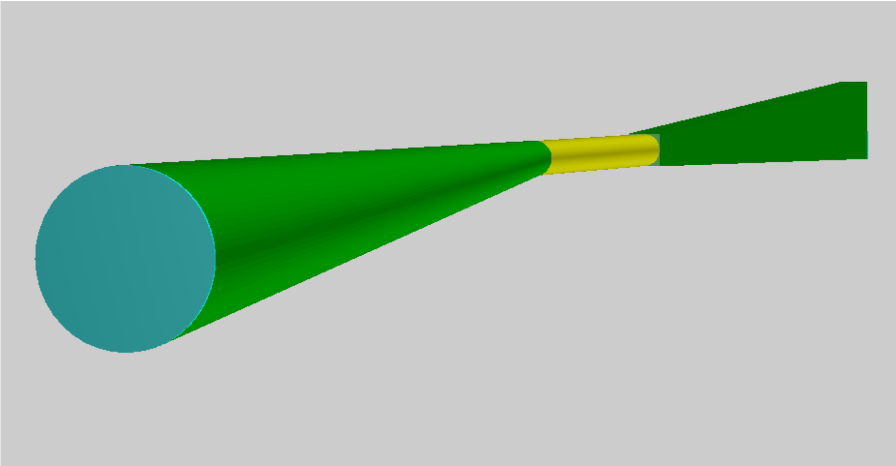
Field map originally produced by a collection of Open Source tools (Elmer, Netgen, ROOT)

- Currently only BeAST field map import implemented
- Interface is forward compatible with the greenfield solenoid maps (?)

IR vacuum chamber description



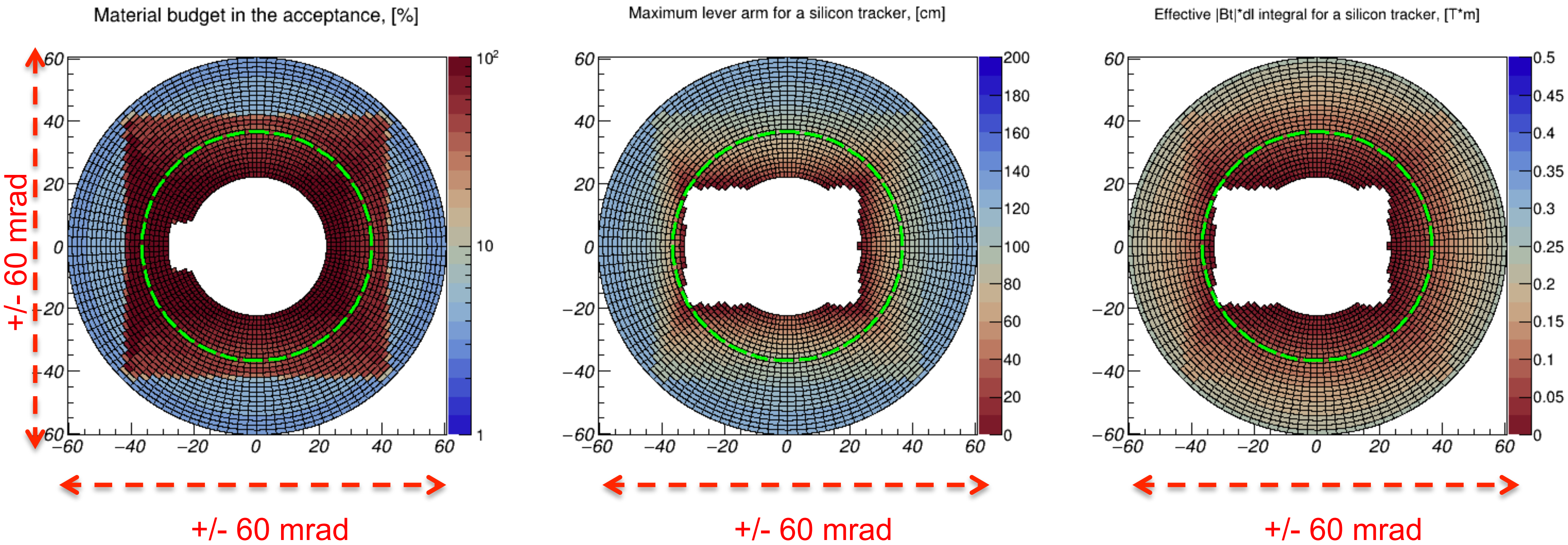




CAD drawing and ROOT TGeo implementation

- Coded in TGeo, exportable as GDML
- Kind of parametric *(suitable for the 2-d IR description)*
- *Only the essential part (the outer shell in particular) is implemented*

B*dl integral and material scan evaluation



- Material budget: direct use of the vacuum chamber TGeo implementation
- Estimate of the maximum lever arm available for the silicon tracker:
  - Account for the vacuum chamber shape: consider a 3D point where a particle with a given $\{\theta,\phi\}$ would exit the vacuum chamber (starting point) …
  - … and account for the configurable markers, indicating at which max distance from the IP the last silicon tracker station can be installed (end point)
- B_T*dl integral estimate: same idea + BeastMagneticField interface
- Primary vertex smearing implemented (this part is trivial of course)

Documentation

https://github.com/eic/EicToyModel/

README.md

Perhaps almost too detailed already

EIC Toy Model (ETM)

EicToyModel (ETM) is a C++ ROOT-based software suite for EIC Central Detector configuration purposes.

Content:

- Introduction
- Other features
- Model limitations
- Pre-requisites
- Downloading
- Compiling
- Running
- GEANT interface
- CAD interface
- Magnetic field interface
- Vacuum chamber material and B*dl integral scans
- ROOT macros options

Introduction

The main objective of this software was providing the EIC community users with a tool, which would allow them to easily (whatever easy means) create possible EIC detector configuration *templates* (namely, the self-consistent collections of 3D sub-detector intergation volumes) and make use of them in the GEANT simulation environment.