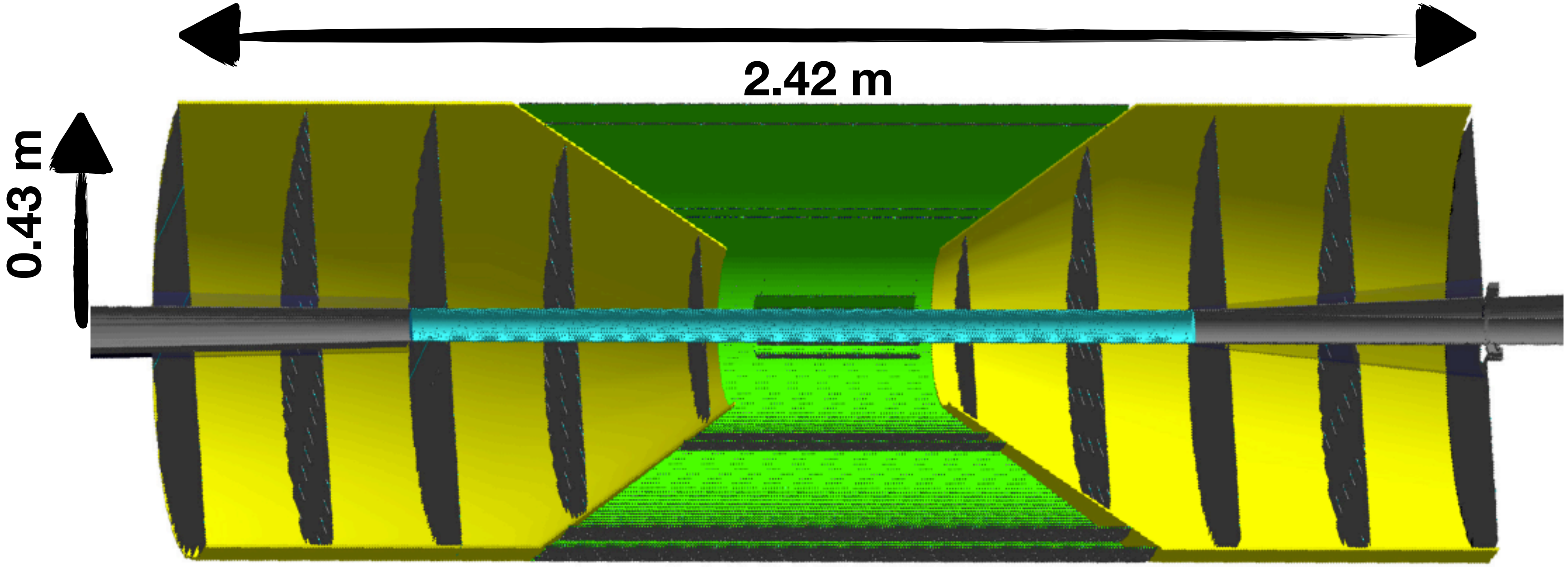
Aspects of all-silicon concept simulation in Fun4All for the YR

Rey Cruz-Torres*
(lots of help from Chris Pinkenburg and Jin Huang)

EIC@IP6 tracking meeting
05/12/2021

* full disclosure: I'm far from a Fun4All expert

Integrated (Barrel+Disks) All-Silicon Tracker Concept



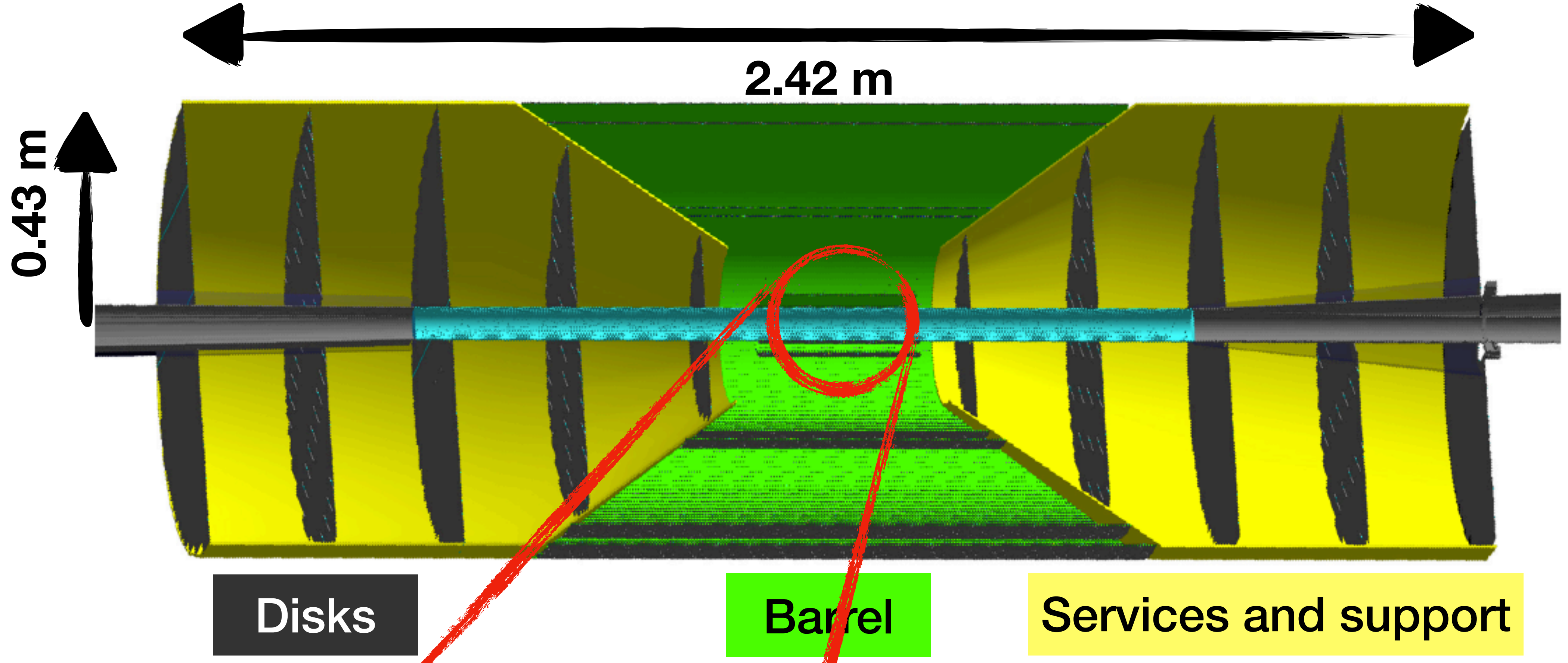
Disks

Barrel

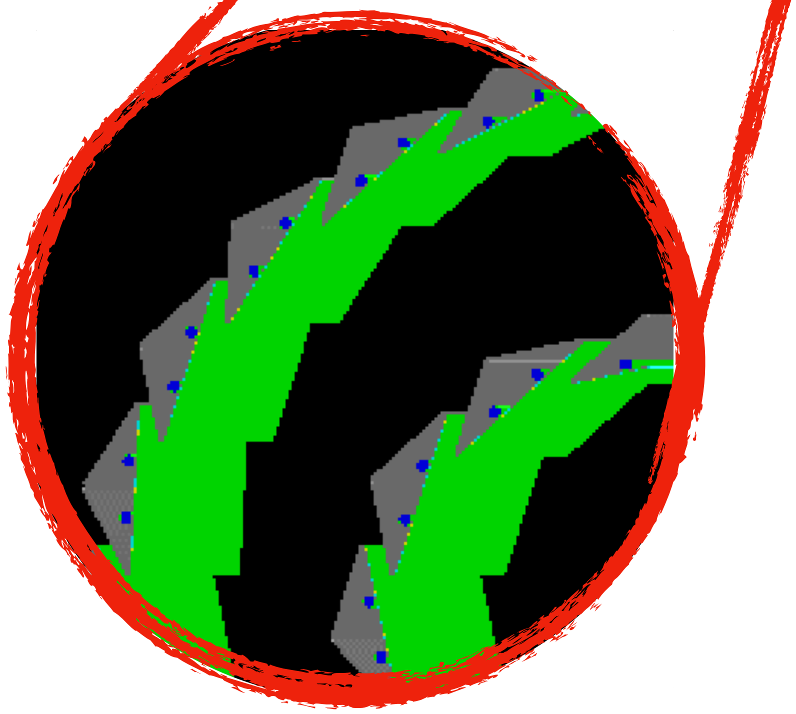
Services and support

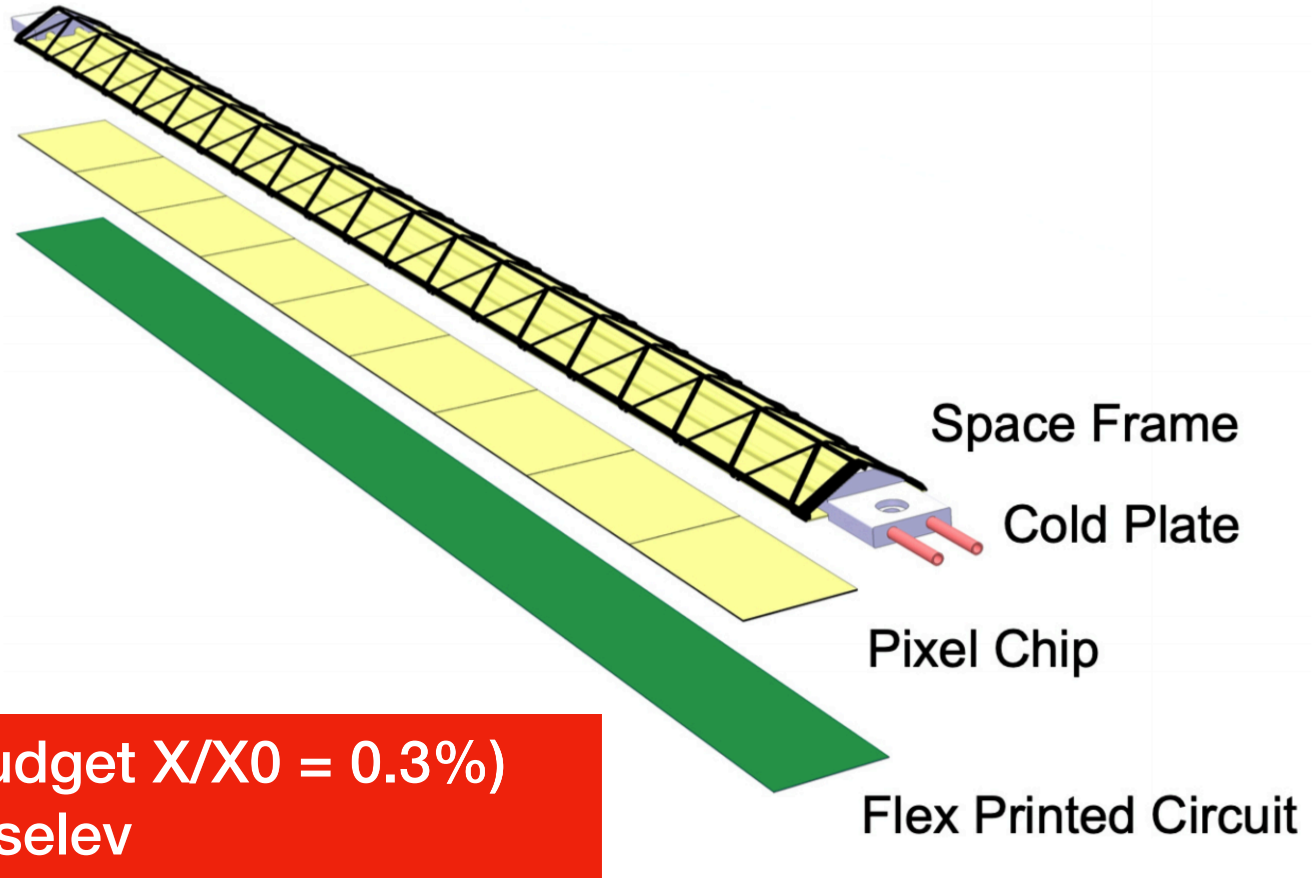
Geometry originally implemented in EICroot by E. Sichtermann and Y.S. Lai

Integrated (Barrel+Disks) All-Silicon Tracker Concept



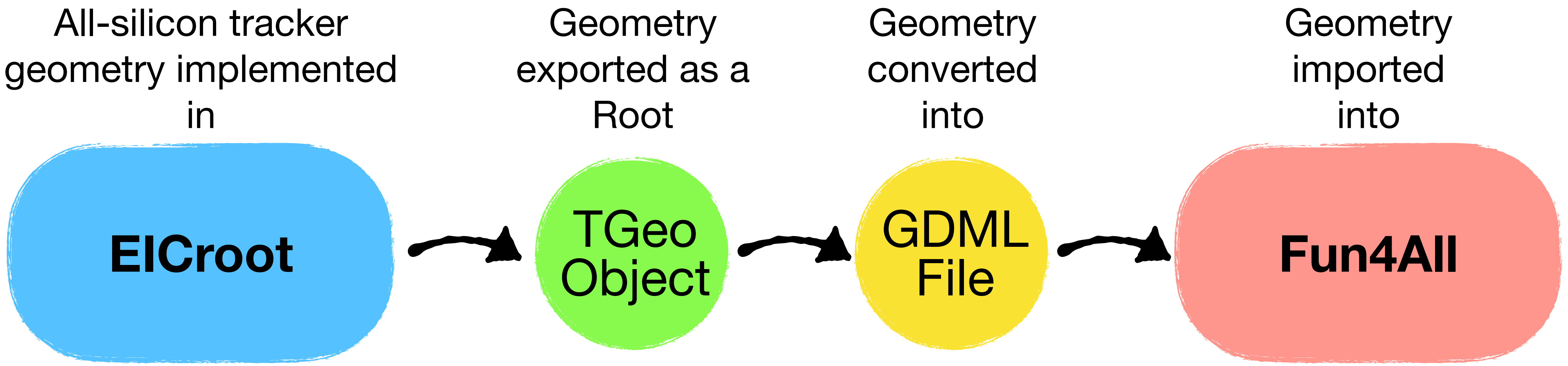
Geometry originally implemented in EICroot by E. Sichtermann and Y.S. Lai





Made out of Alice ITS2-like staves (material budget X/X0 = 0.3%) implemented in EICroot by A. Kiselev

From Geometry Implementation to Simulations



All-silicon tracker geometry implemented in **EICroot** → Geometry exported as a Root TGeo Object → Geometry converted into GDML File → Geometry imported into **Fun4All**

GDML ≡ Geometry Description Markup Language

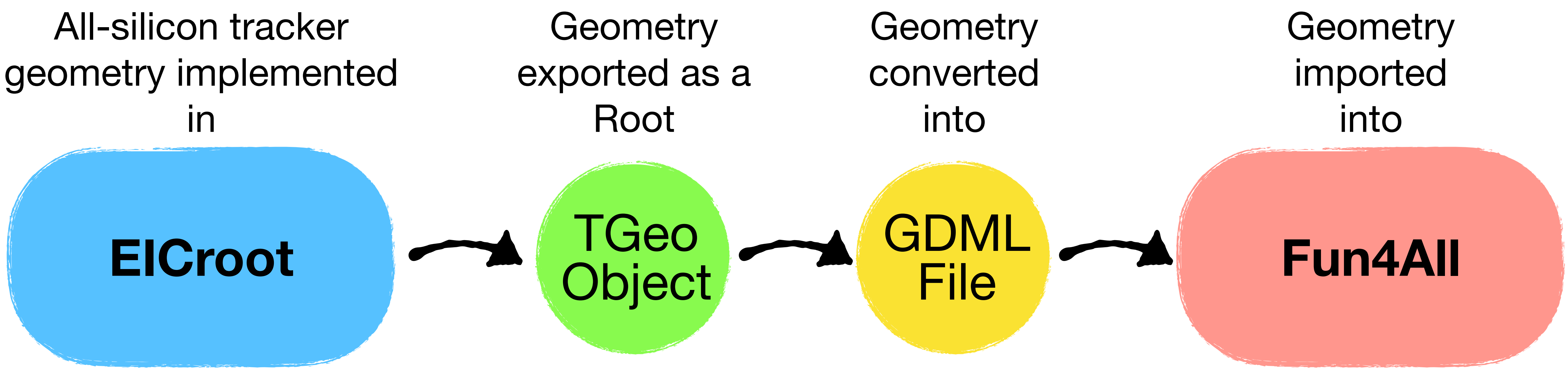
Functionality to import GDML into Fun4All implemented by C. Pinkenburg

https://github.com/reynier0611/g4lblvtx/blob/master/source/AllSiliconTrackerDetector.cc

See, e.g., function AllSiliconTrackerDetector::InsertVolumes

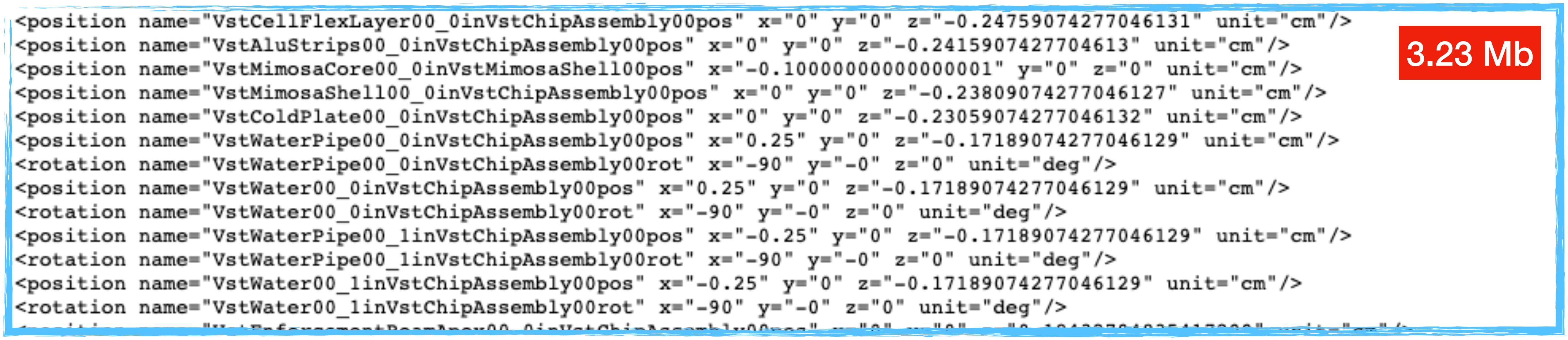
These classes require some fine tuning to work with different gdml files (e.g. some strings are hardcoded)

From Geometry Implementation to Simulations



GDML ≡ Geometry Description Markup Language

Excerpt from the All-Si tracker GDML file

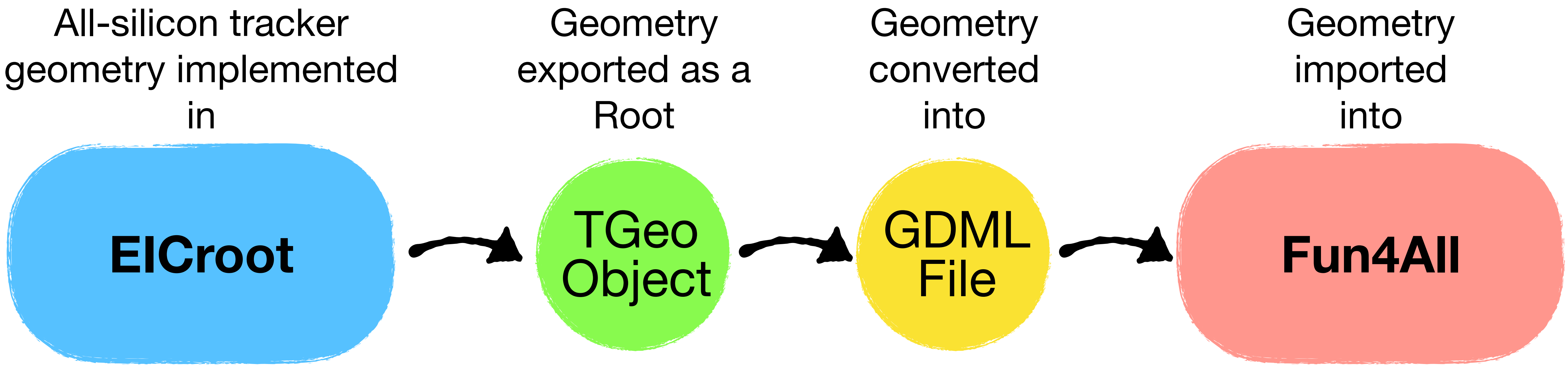


```
<position name="VstCellFlexLayer00_0inVstChipAssembly00pos" x="0" y="0" z="-0.24759074277046131" unit="cm"/>
<position name="VstAluStrips00_0inVstChipAssembly00pos" x="0" y="0" z="-0.2415907427704613" unit="cm"/>
<position name="VstMimosaCore00_0inVstMimosaShell00pos" x="-0.10000000000000001" y="0" z="0" unit="cm"/>
<position name="VstMimosaShell00_0inVstChipAssembly00pos" x="0" y="0" z="-0.23809074277046127" unit="cm"/>
<position name="VstColdPlate00_0inVstChipAssembly00pos" x="0" y="0" z="-0.23059074277046132" unit="cm"/>
<position name="VstWaterPipe00_0inVstChipAssembly00pos" x="0.25" y="0" z="-0.17189074277046129" unit="cm"/>
<rotation name="VstWaterPipe00_0inVstChipAssembly00rot" x="-90" y="-0" z="0" unit="deg"/>
<position name="VstWater00_0inVstChipAssembly00pos" x="0.25" y="0" z="-0.17189074277046129" unit="cm"/>
<rotation name="VstWater00_0inVstChipAssembly00rot" x="-90" y="-0" z="0" unit="deg"/>
<position name="VstWaterPipe00_1inVstChipAssembly00pos" x="-0.25" y="0" z="-0.17189074277046129" unit="cm"/>
<rotation name="VstWaterPipe00_1inVstChipAssembly00rot" x="-90" y="-0" z="0" unit="deg"/>
<position name="VstWater00_1inVstChipAssembly00pos" x="-0.25" y="0" z="-0.17189074277046129" unit="cm"/>
<rotation name="VstWater00_1inVstChipAssembly00rot" x="-90" y="-0" z="0" unit="deg"/>
```

3.23 Mb

Modifying the geometry once it has been exported into GDML is not practical

From Geometry Implementation to Simulations



All-silicon tracker geometry implemented in

EICroot

Geometry exported as a Root

TGeo Object

Geometry converted into

GDML File

Geometry imported into

Fun4All

GDML ≡ Geometry Description Markup Language

Alternative

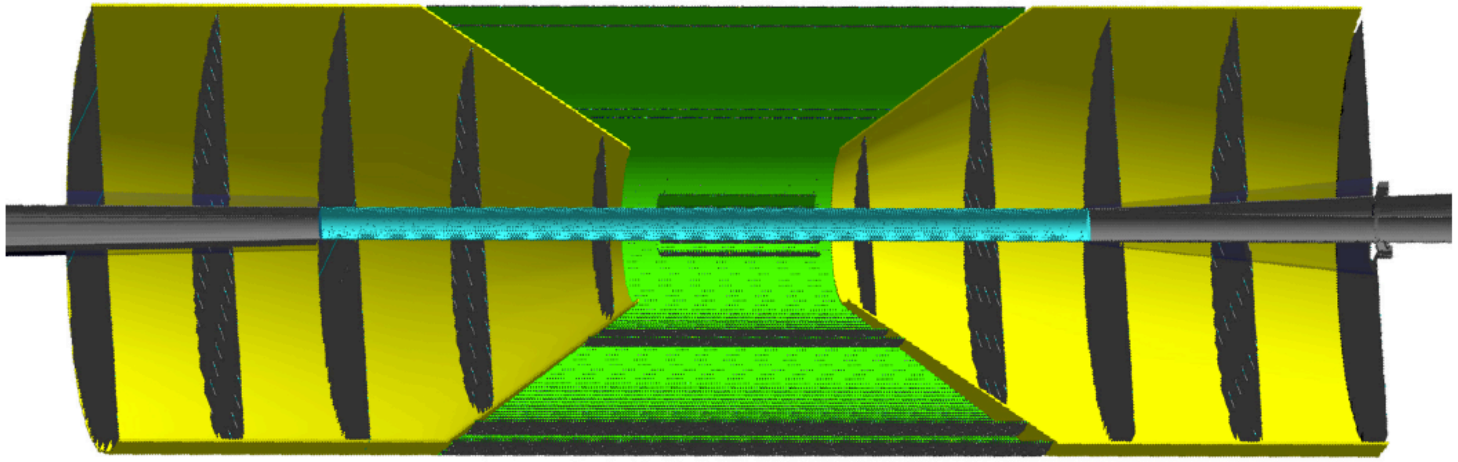
Directly implement geometry in Fun4All

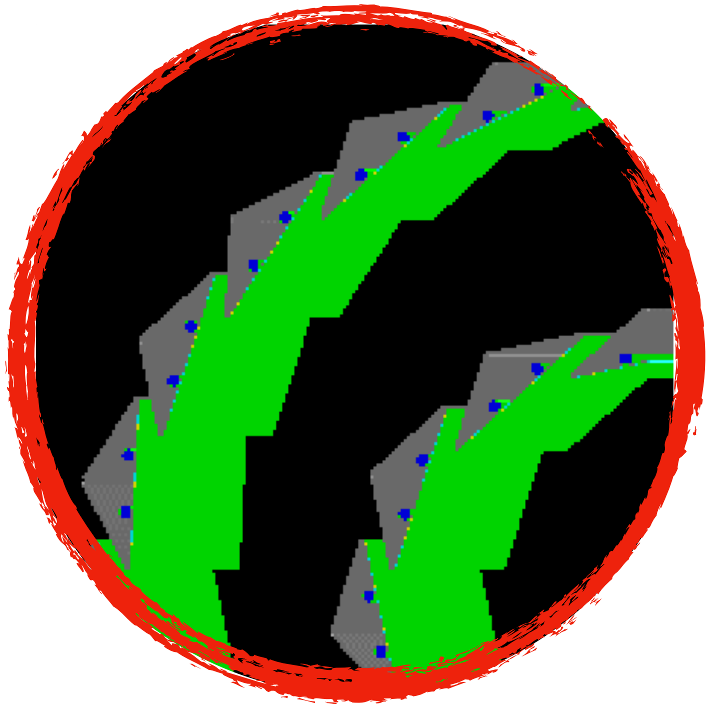
Implementing realistic stave-based geometry: see [presentation by A. Kiselev]

Implementing simplified geometry

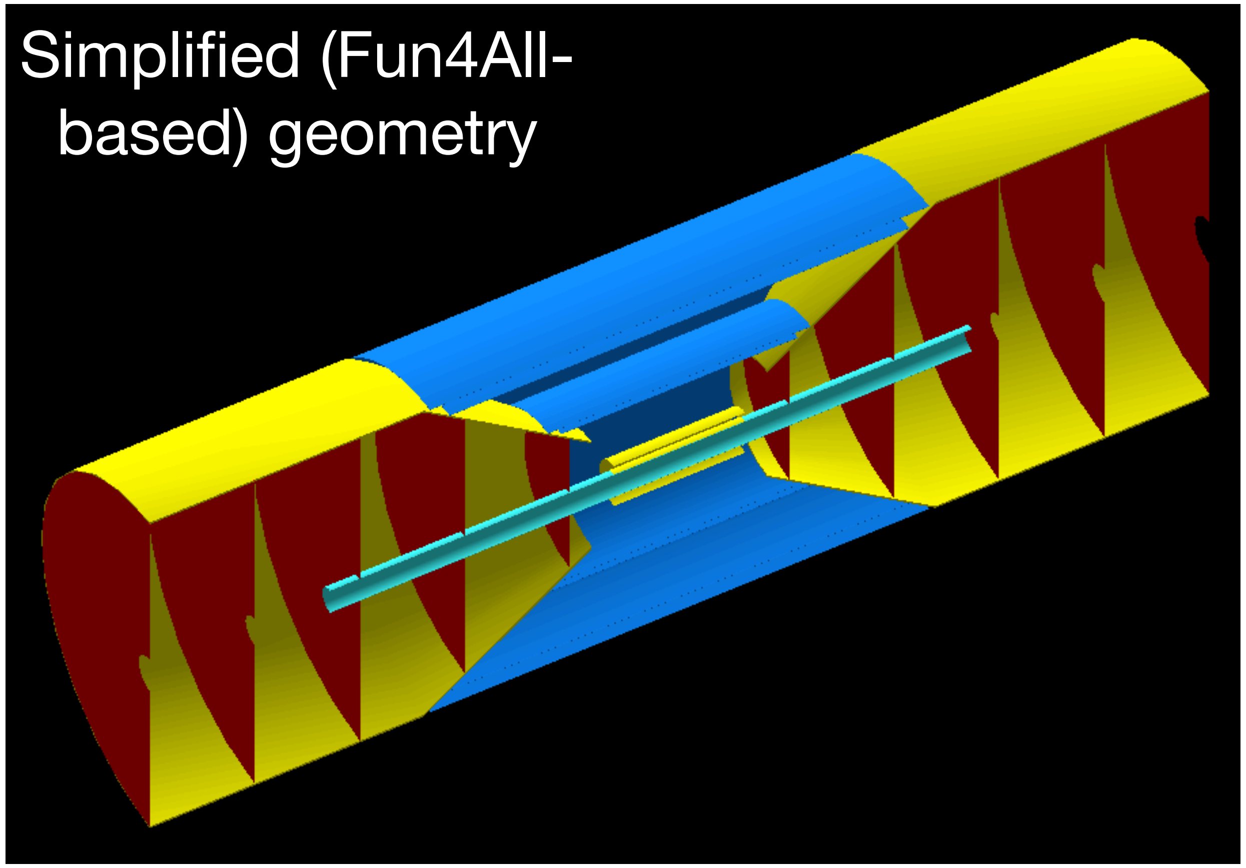
Simplified all-si tracker geometry

Realistic stave-based geometry





Simplified (Fun4All-based) geometry

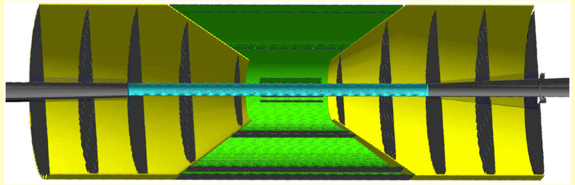


Identical detector layouts (i.e. number and placement of layers, equivalent average material budget, …)

Allows for quick geometry modifications, e.g.:
- adding or removing layers
- changing layer coordinates
- altering material budget

Besides providing ease of geometry modifications, it also avoids issues with other parameters such as #staves per layer, stave rotations, …

How to load these geometries

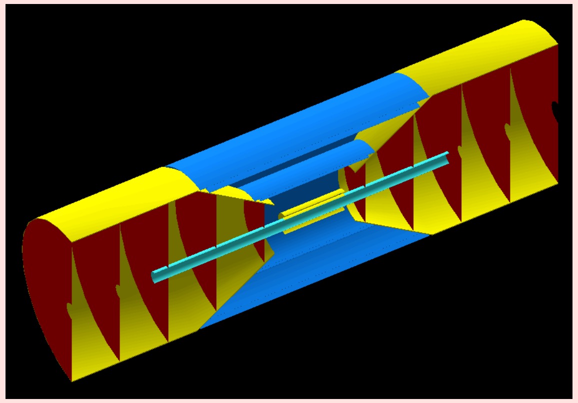


Example of code using stave-based geometry:

https://github.com/reynier0611/g4lblvtx/blob/master/macros/Fun4All_G4_FastMom.C#L169

This macro uses functions from the G4_AllSi.C macro. Specifically, the geometry is loaded in the "load_AllSi_geom" function

https://github.com/reynier0611/g4lblvtx/blob/master/macros/G4_AllSi.C#L13

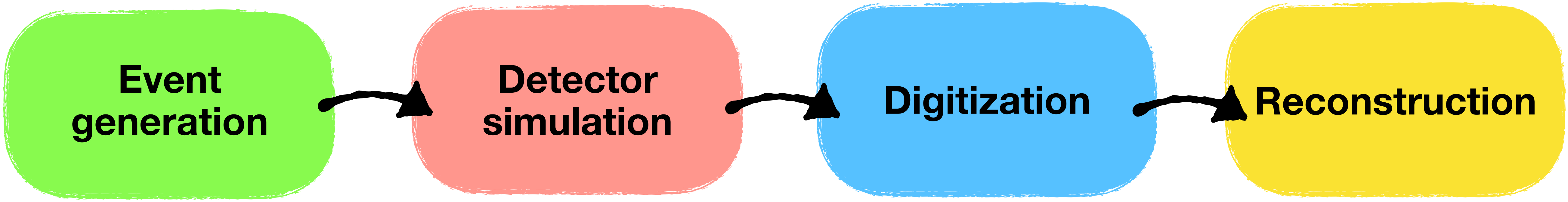


Example of code using simplified geometry:

https://github.com/reynier0611/g4lblvtx/blob/master/macros/auxiliary_studies/simplified_geometry/Fun4All_G4_simplified_v2.C#L110

Geometry implemented in lines 110 - 189

Adding geometry to Kalman Filter



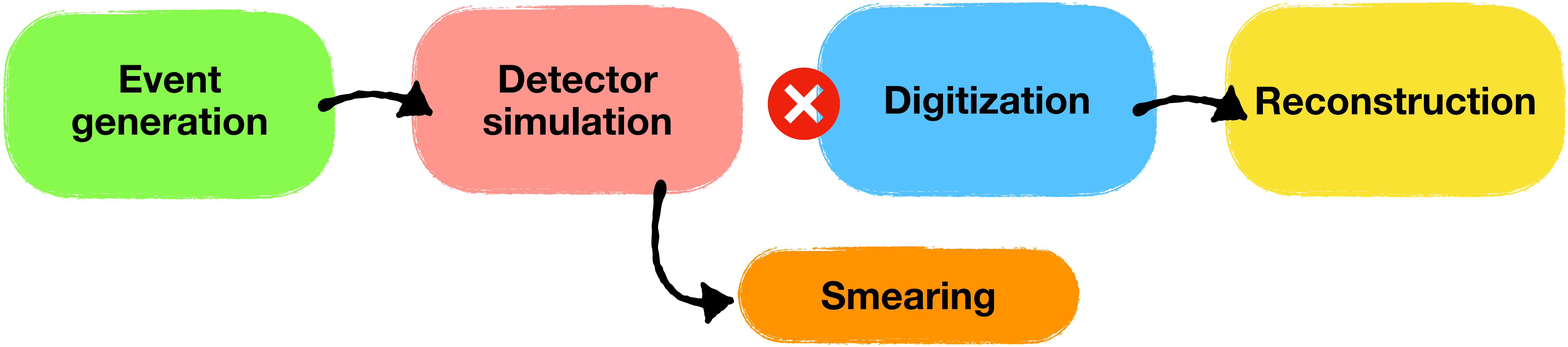
Event generation

Detector simulation

Digitization

Reconstruction

Adding geometry to Kalman Filter



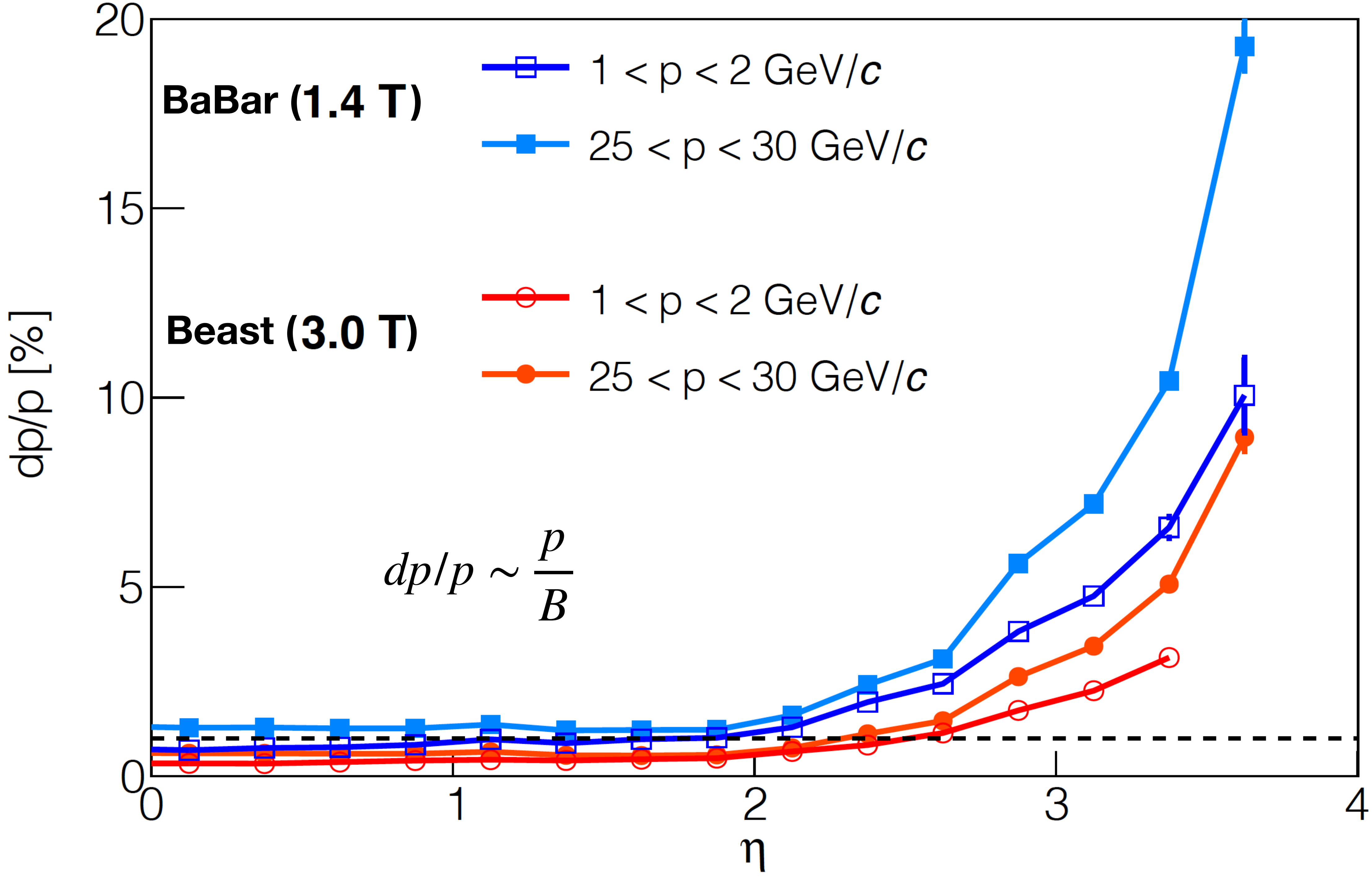
```
PHG4TrackFastSim *kalman

kalman->add_phg4hits(
    std::string& phg4hitsNames (“G4Hit…”)
    detector type phg4dettype (PHG4TrackFastSim:: Cylinder, Vertical_Plane)
    radial-resolution* [cm] (this parameter is not used in barrel-like geometry)
    azimuthal (arc-length)* resolution [cm]
    longitudinal (z) resolution* [cm] (this parameter is not used in disk-like geometry)
    fractional hit-finding efficiency (1)
    hit noise (0)
);
```

*10 μm pixel -> resolution of $(10\ \mu\mathrm{m})/\sqrt{12}$

Momentum resolution vs. pseudorapidity

π^-, 10 μm pixel, $X/X_0 = 0.3\%$

Full (Geant4) simulations

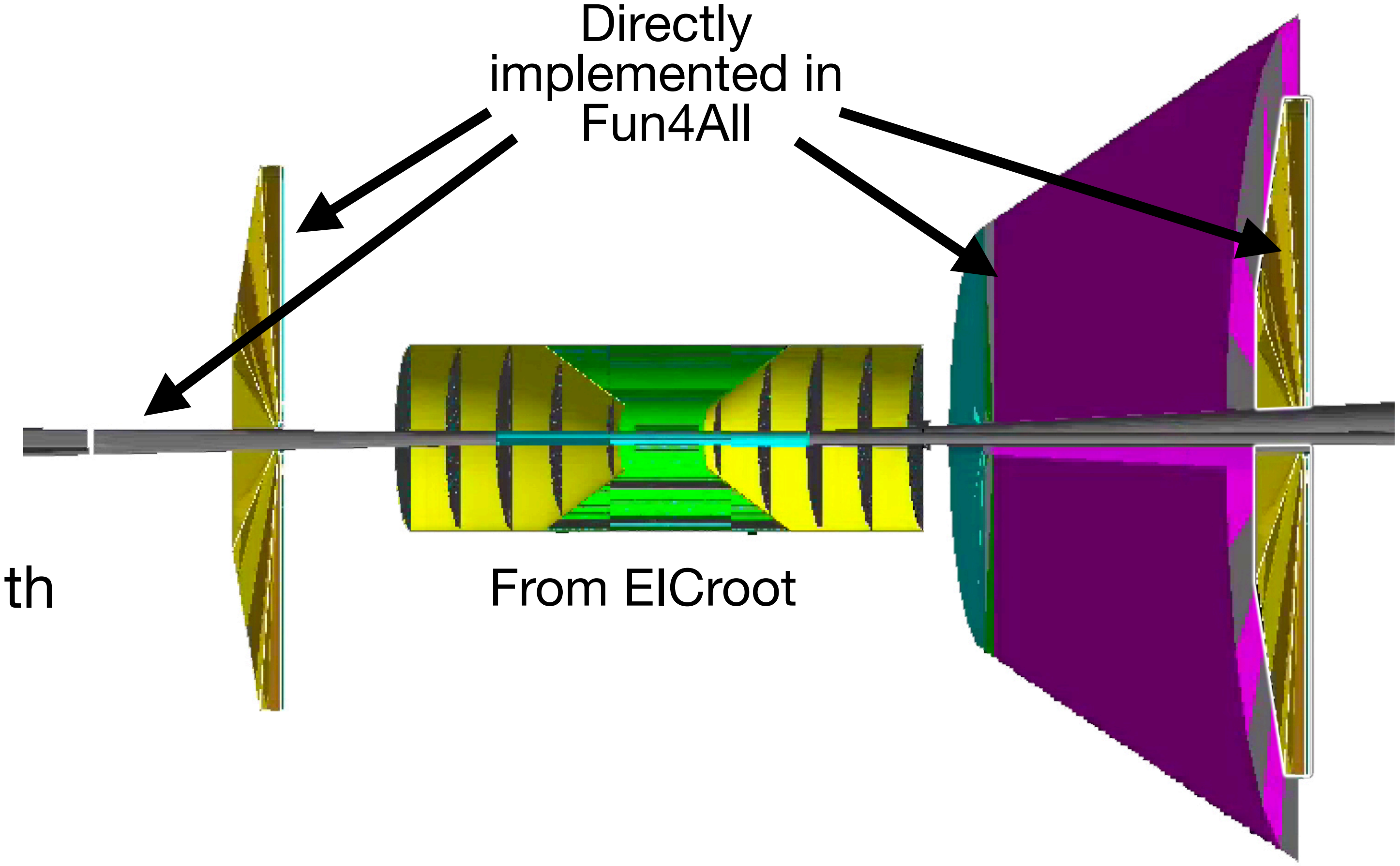


η	0	0.5	1	1.5	2	3	4	5
θ [deg]	90	62	40	25	15	5.7	2.1	0.8

Complementing All-Si tracker with other detectors

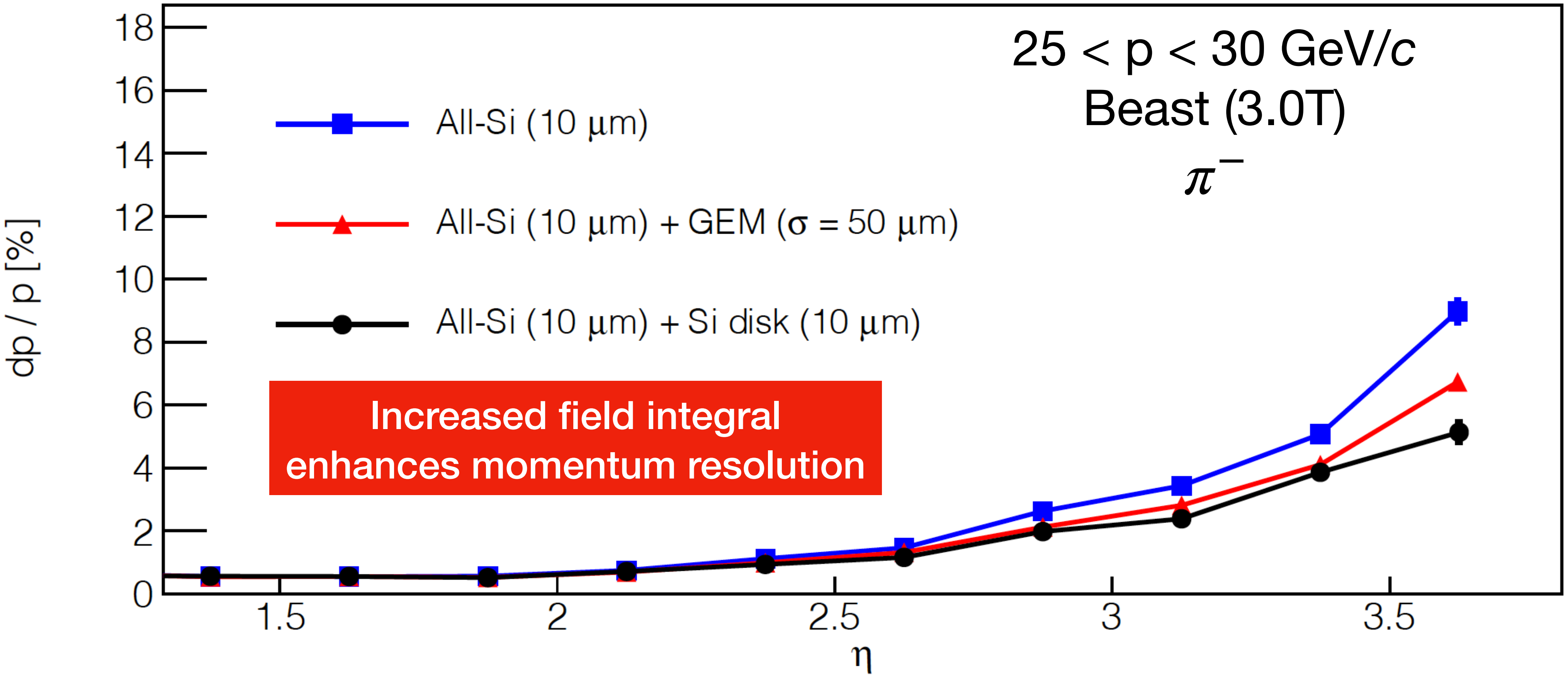
YR studies included:

- Momentum, angular (@vertex and @PID), DCA, … resolution characterization
- Geometry optimization
- Crossing-angle impact on dp/p
- Impact of complementing all-si tracker with far-away tracking stations



Directly implemented in Fun4All

From EICroot

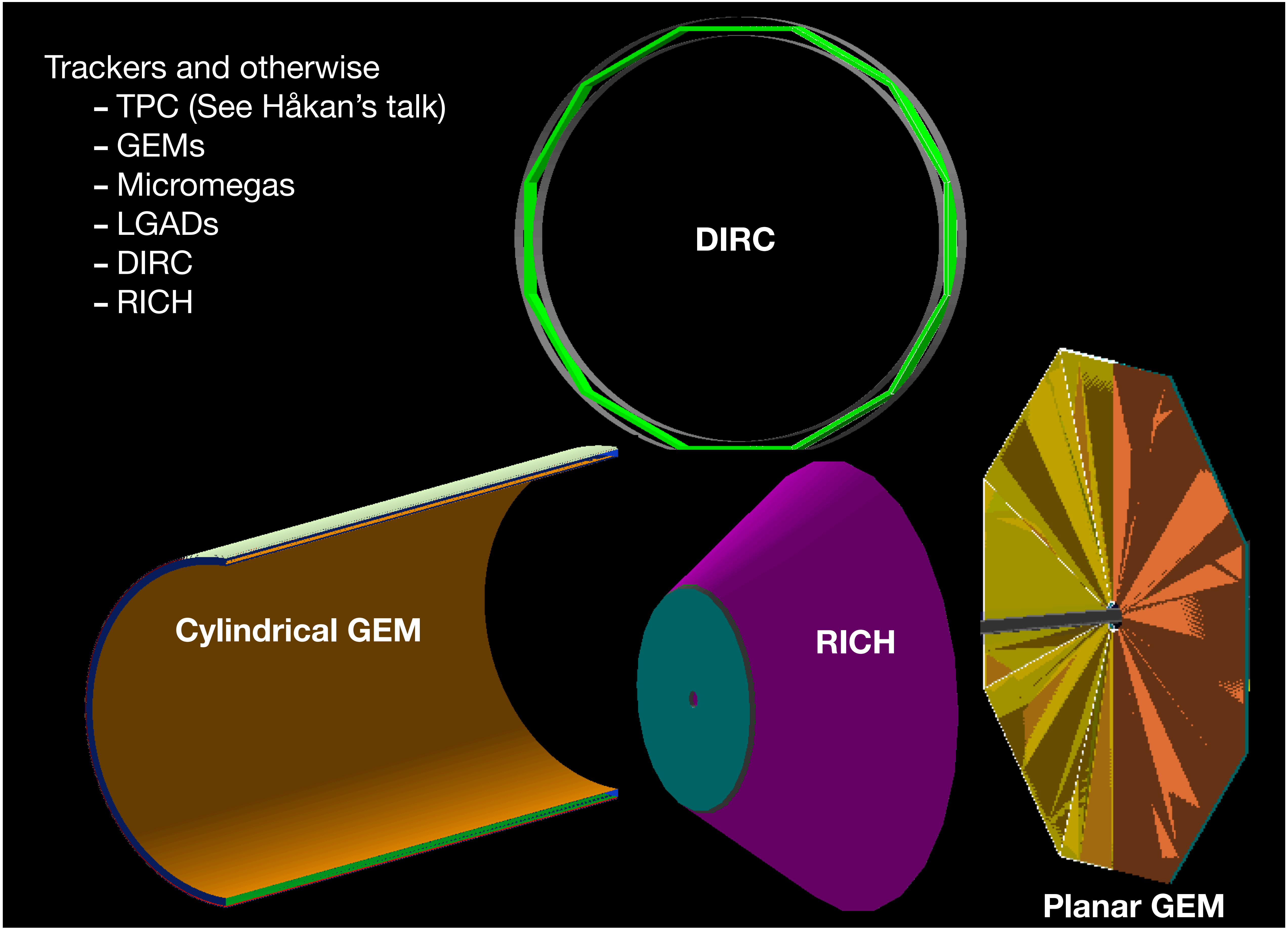


Increased field integral enhances momentum resolution

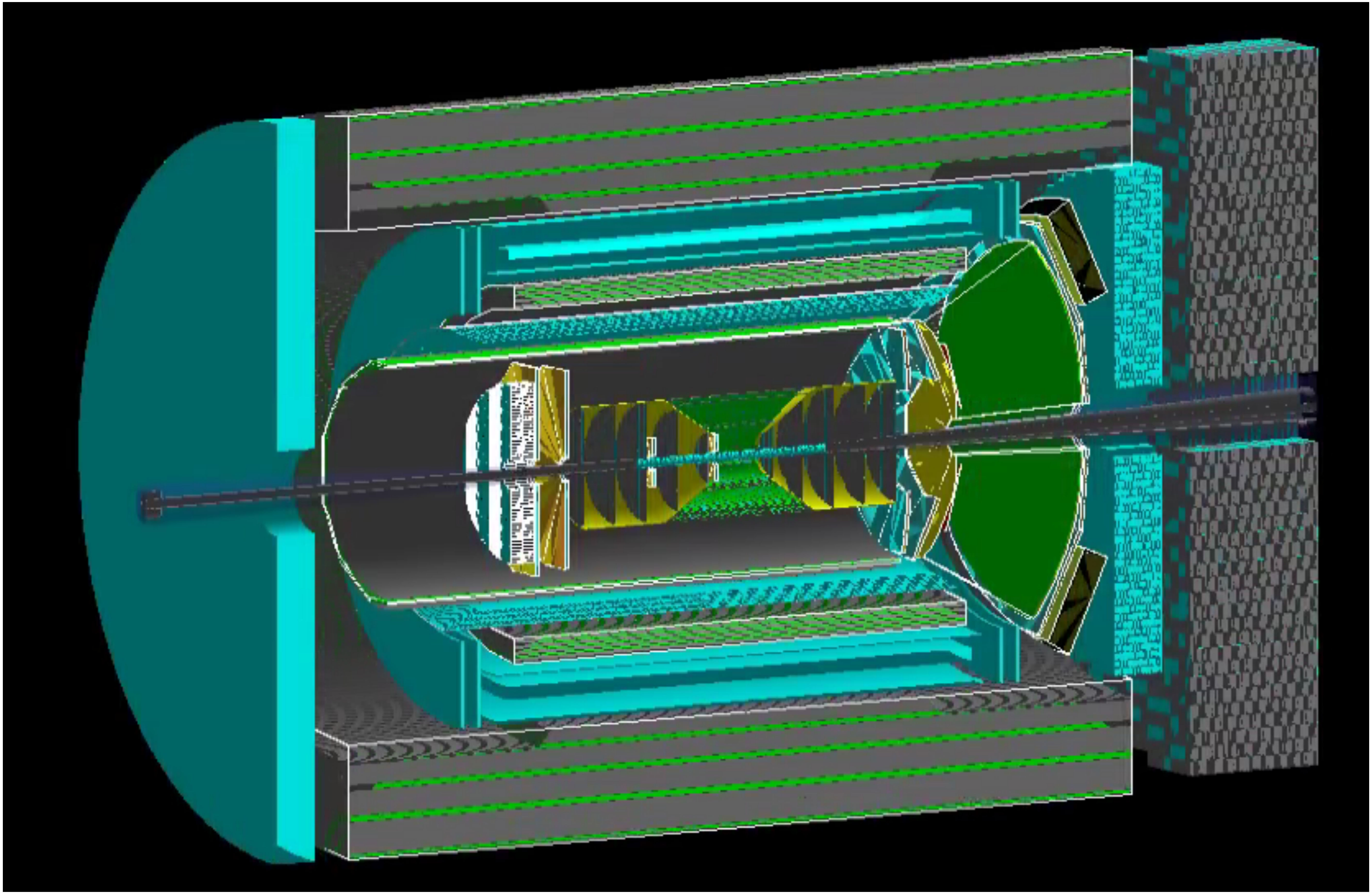
Plethora of implemented detectors

Trackers and otherwise

- TPC (See Håkan’s talk)
- GEMs
- Micromegas
- LGADs
- DIRC
- RICH



Fun4All Modularity



Summary

- Fun4All is a flexible, modular, full-simulation framework
- It allows importing geometries implemented elsewhere
- It was extensively used during the YR exercise
- Very quickly evolving

Standalone all-silicon tracker simulations:

https://github.com/eic/g4lblvtx

https://github.com/reynier0611/g4lblvtx

Resources:

Doxygen: https://eic-detector.github.io/doxygen/

Mattermost: https://chat.sdcc.bnl.gov/eic