

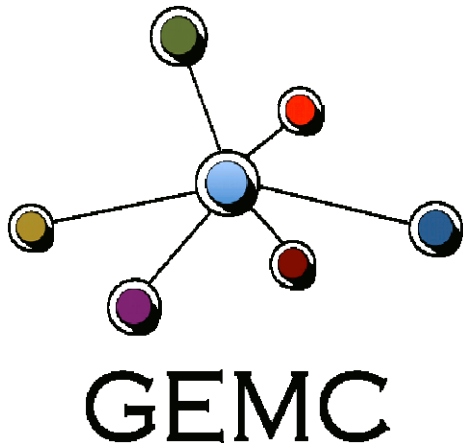
# GEMC Overview

Status, Future of GEMC

Maurizio Ungaro

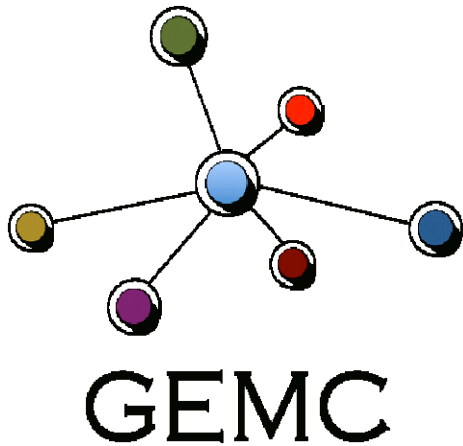
# GEMC (GEant4 MonteCarlo)

gemc is a C++ program that simulates particles through matter using the geant4 libraries.

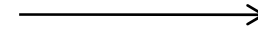


# GEMC (GEant4 MonteCarlo)

gemc is a C++ program that simulates particles through matter using the geant4 libraries.



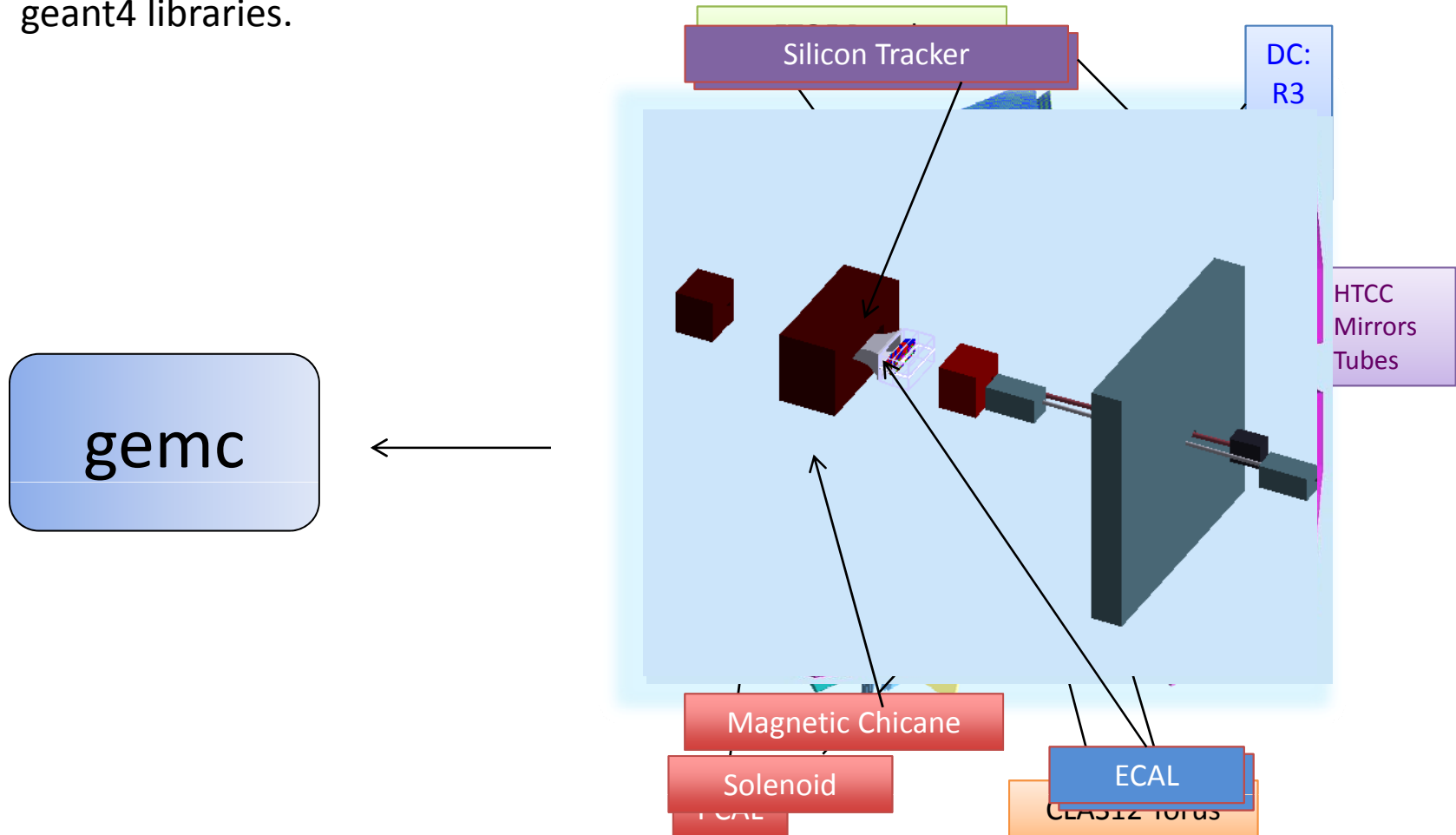
GEOMETRY,  
BANKS,  
DIGITIZATION  
DATABASE



gemc

# GEMC (GEant4 MonteCarlo)

gemc is a C++ program that simulates particles through matter using the geant4 libraries.



# GEMC (GEant4 MonteCarlo)

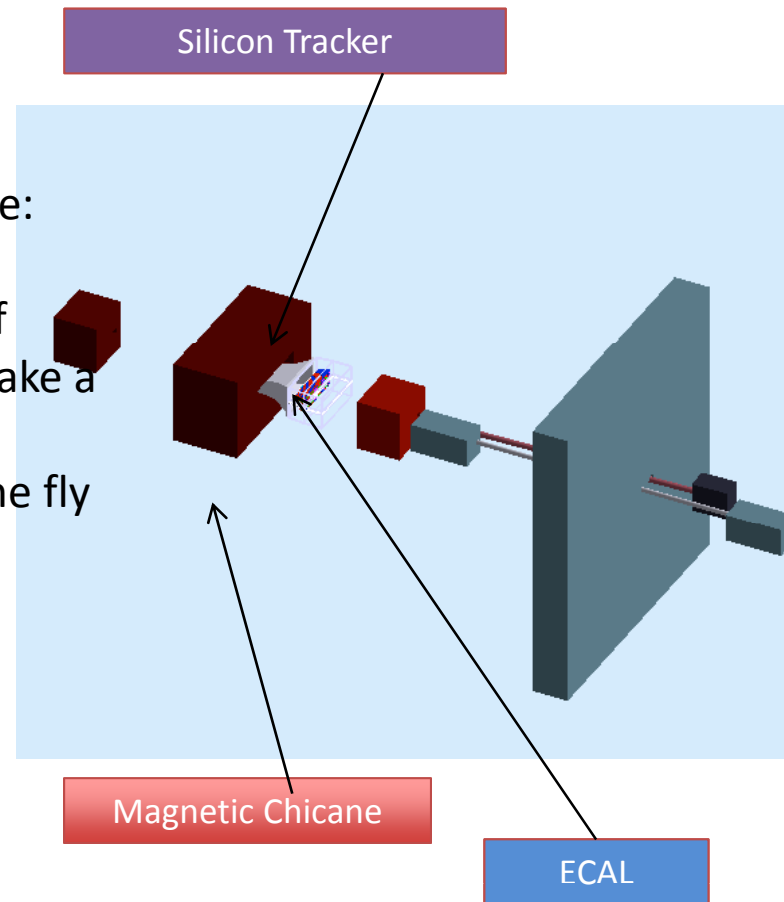
## Advantages:

### User:

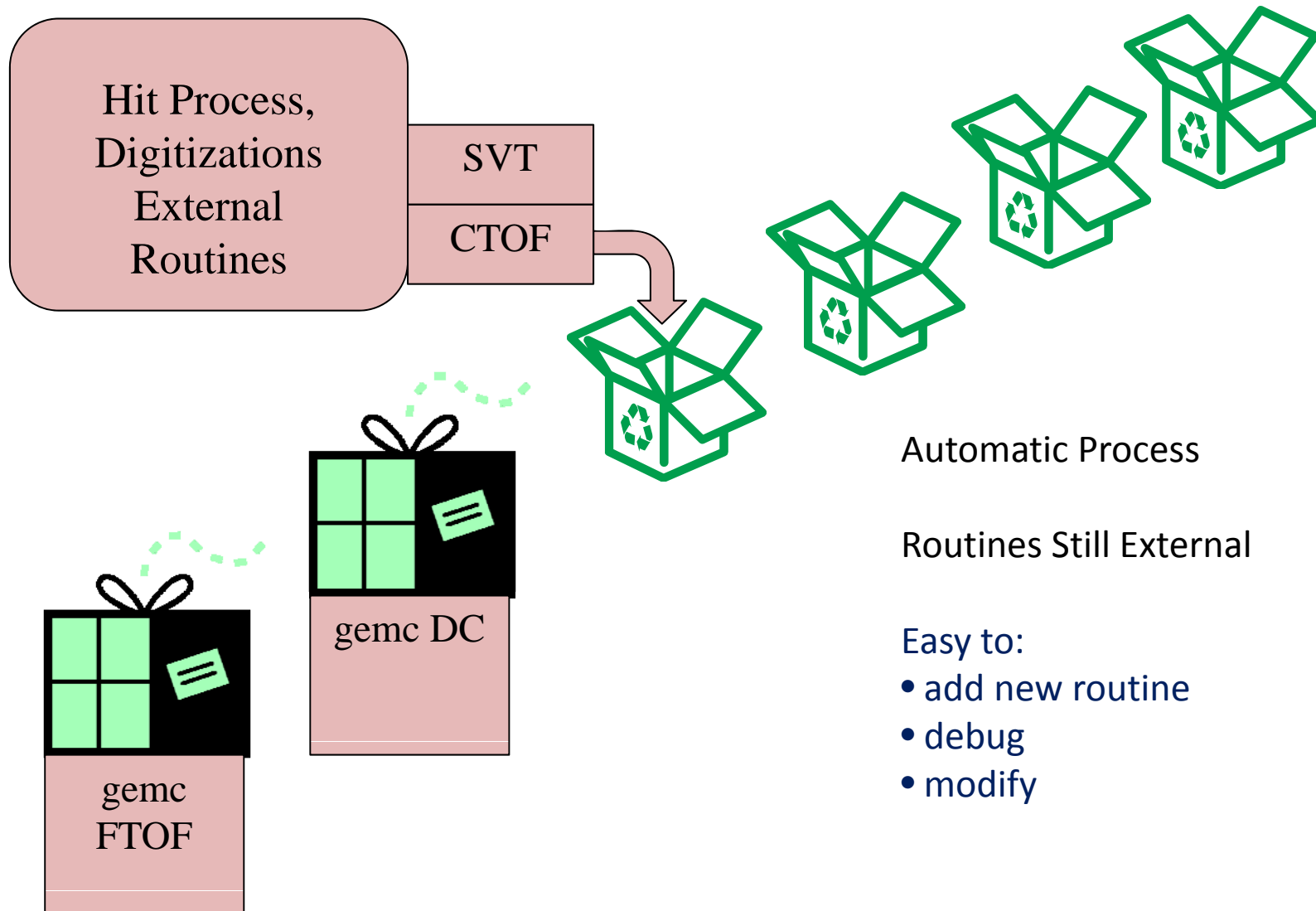
- Change parameters w/o touching the code: very fast R/D and deployment
- Abstract Objects instantiated thousands of times ensures they are “correct”: “can’t” make a mistake
- Can make “align, kinematic fit” on the fly (tilts, shifts)
- No need to know C++ or Geant4

### Development:

- Easy to debug (table → reproducibility)
- Can concentrate on “core” code.



# Factory Method for Hit Processes



# Output (factory)

TXT output:

`-OUTPUT="txt, data.txt"`

EVIO OUTPUT:

`-OUTPUT="evio, data.ev"`

# Event Generation

- 1) With gemc internal generator
- 2) LUND Format for physics events:

```
8 34406 0 0. 0. 0. 0. 0. 0. 0.
  1 0 1 2212 0 0 0.000 0.000 0.000 0.938 0.938 0.000 0.000 0.000
  2 0 1 11 0 0 0.401 -0.605 5.517 5.565 0.001 0.000 0.000 0.000
  3 0 1 211 0 0 -0.305 0.296 3.417 3.446 0.140 0.000 0.000 0.000
  4 0 1 2112 0 0 -0.379 -0.062 1.627 1.918 0.940 0.000 0.000 0.000
  5 0 1 -211 0 0 -0.015 0.114 0.098 0.205 0.140 0.000 0.000 0.000
  6 0 1 211 0 0 0.137 0.406 0.306 0.545 0.140 0.000 0.000 0.000
  7 0 1 22 0 0 0.162 -0.070 0.041 0.181 0.000 0.000 0.000 0.000
  8 0 1 22 0 0 -0.001 -0.078 -0.006 0.079 0.000 0.000 0.000 0.000
9 34407 0 0. 0. 0. 0. 0. 0. 0.
  1 0 1 2212 0 0 0.000 0.000 0.000 0.938 0.938 0.000 0.000 0.000
  2 0 1 11 0 0 0.340 0.384 2.811 2.857 0.001 0.000 0.000 0.000
  3 0 1 -211 0 0 -0.114 0.357 3.700 3.721 0.140 0.000 0.000 0.000
  4 0 1 2212 0 0 -0.002 -0.776 1.742 2.126 0.938 0.000 0.000 0.000
  5 0 1 211 0 0 0.336 -0.200 0.941 1.029 0.140 0.000 0.000 0.000
  6 0 1 22 0 0 -0.141 0.107 0.033 0.180 0.000 0.000 0.000 0.000
  7 0 1 22 0 0 -0.044 0.118 -0.067 0.143 0.000 0.000 0.000 0.000
  8 0 1 22 0 0 -0.150 0.021 0.455 0.480 0.000 0.000 0.000 0.000
  9 0 1 22 0 0 -0.225 -0.010 1.384 1.402 0.000 0.000 0.000 0.000
```

To all these “primary” particles, a luminosity beam can be added



# Luminosity Beam

- Number of Beam Particle / One single Event
- Time Window of the event
- Time Structure of the Event: “bunches” time separation

60000 electrons / event

1 event = 120 nanoseconds

Bunches every 2 nanoseconds (60 bunches, 500 electron each bunch)

Physics event generated in the middle (60 ns)

# Physics Processes Databases

## LHEP Physics Lists

The LHEP Physics lists are based on a parameterized modeling for all hadronic interactions for all particles. Based on Geisha model.

Example:

LHEP\_BERT (Bertini Cascade)

## String model based physics lists

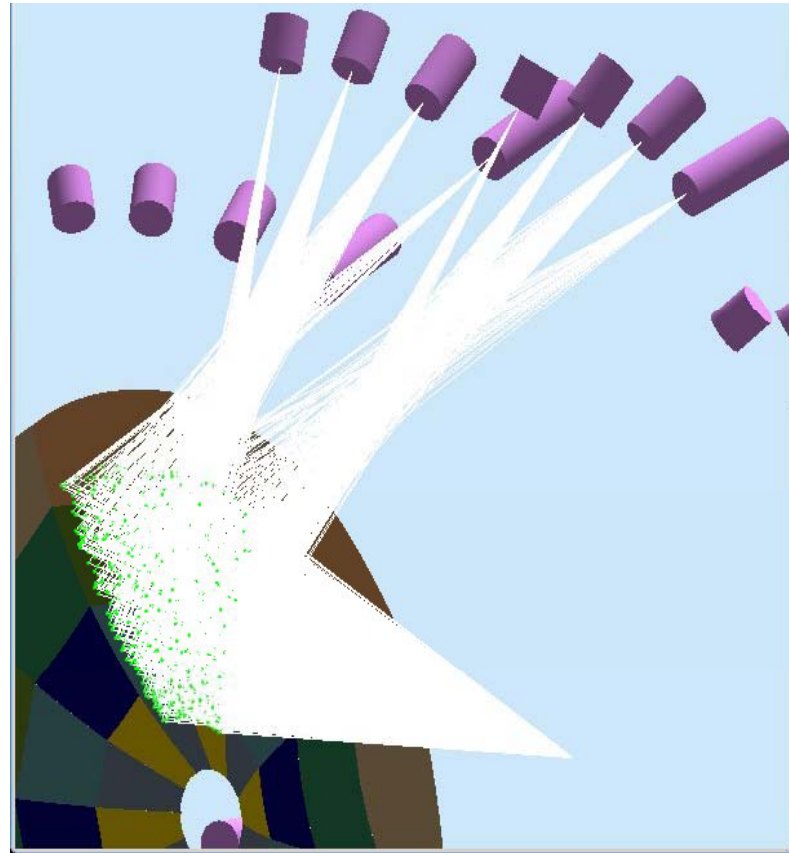
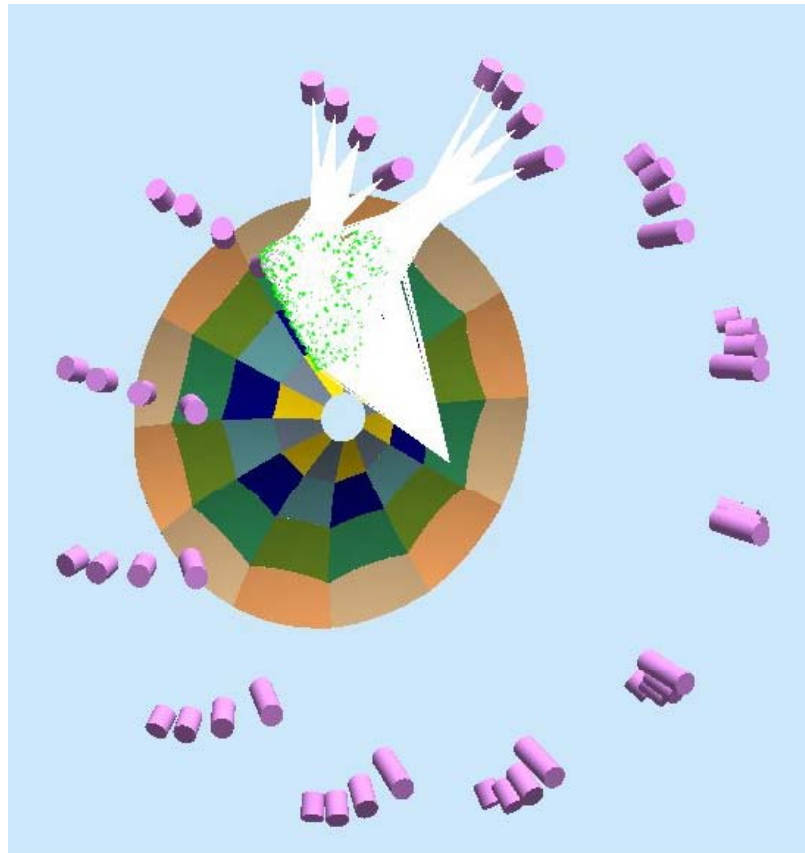
These Physics lists apply a string models for the modeling of interactions of high energy hadrons, i.e. for protons, neutrons, pions and Kaons above  $\sim(5-25)$  GeV. Examples:

QGSP: quark gluon string model

QGSC: CHIPS modeling for the nuclear de-excitation

QGSC\_BERT: Bertini cascade for primary protons, neutrons, pions and Kaons below  $\sim 10$  GeV. (recommended for CLAS12)

# Optical Processes in GEMC



# Physics Processes Databases

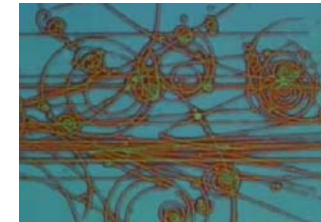
## Future: factory process

Model the physics processes in the external database:

- Can turn on/off single processes for all/individual particles (very useful for eg5 comparison and debugging in general)
- Can modify energy range of applicability of processes (e.g. Bertini cascade model, Low energy parameterization, etc)

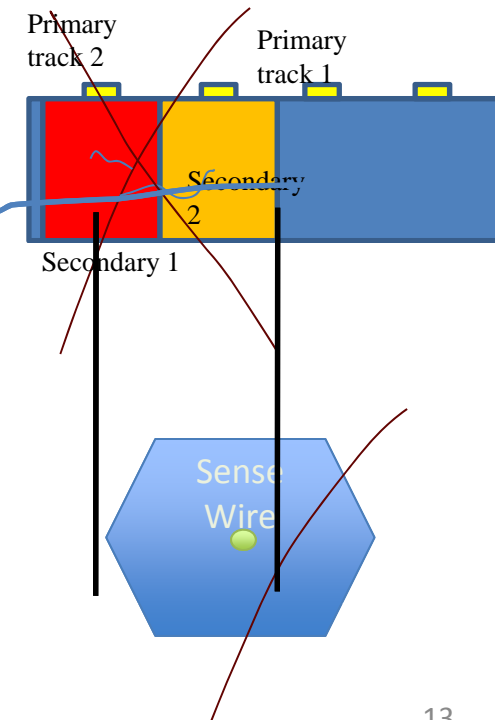
# Hit Types Databases

- 1) "FLUX" type: every track has its own hit.  
Good for counting purposes (i.e. how many protons pass through a detector, etc)



- 2) Signal Modeling: All voltage responses to the passage of particles in a time window for a particular detector will be added to a single hit

- 3) Time Window TDC: the *first signal* within the detector time window will give the TDC.

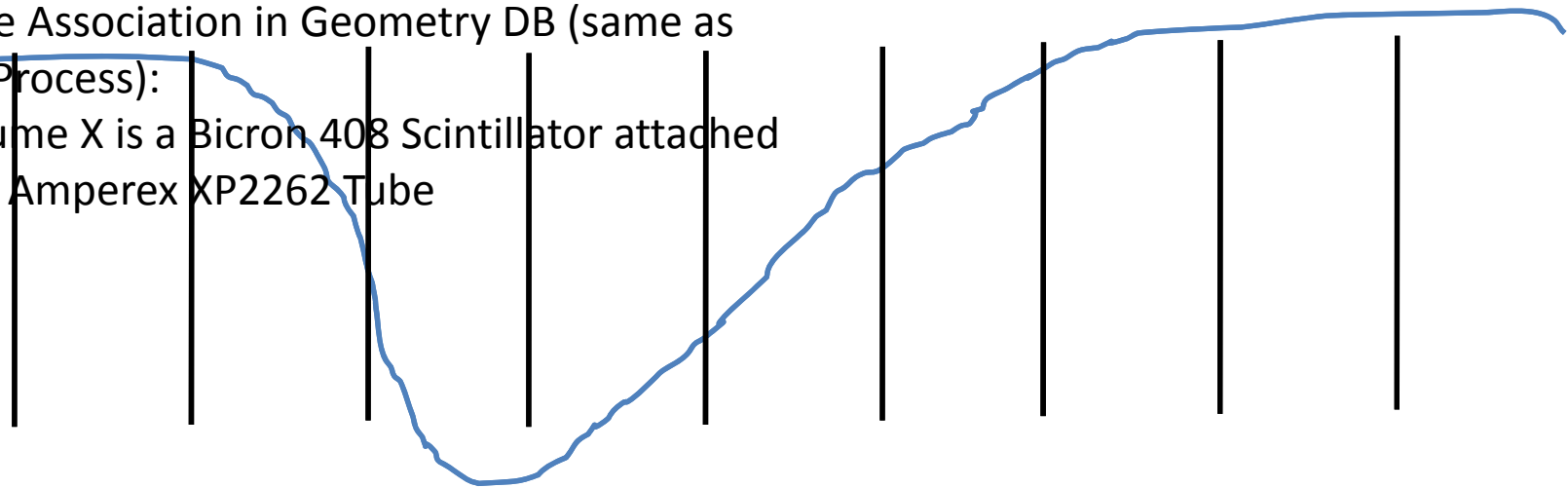


# Hit Types Databases

Signal Modeling Parameterization:

Each Scintillator, Tube, etc, modeled by a few parameters in DB

Tube Association in Geometry DB (same as Hit Process):  
Volume X is a Bicron 408 Scintillator attached to a Amperex XP2262 Tube



# Geometry Database

**Extension to GDML input  
(output is supported thanks to Maurik)  
Hall-D extension?  
LCSIM extension?**

# Materials Database

## **Future: Factory Method**

Materials will be defined in DB  
GDML Extension

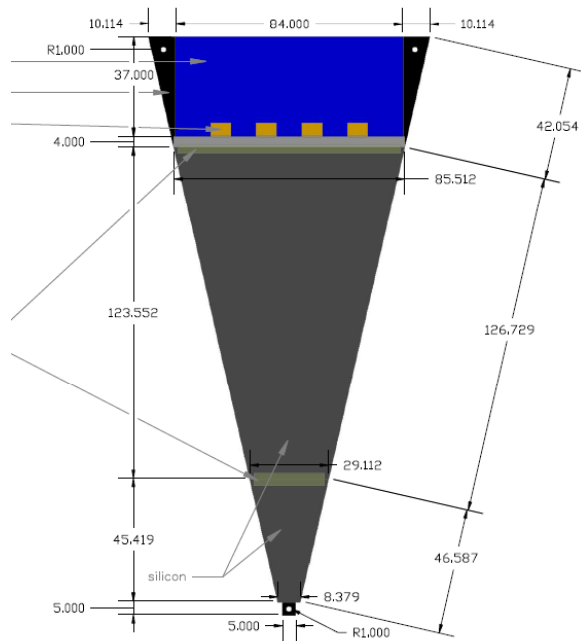
(last hurdle to complete the code-writing independence)



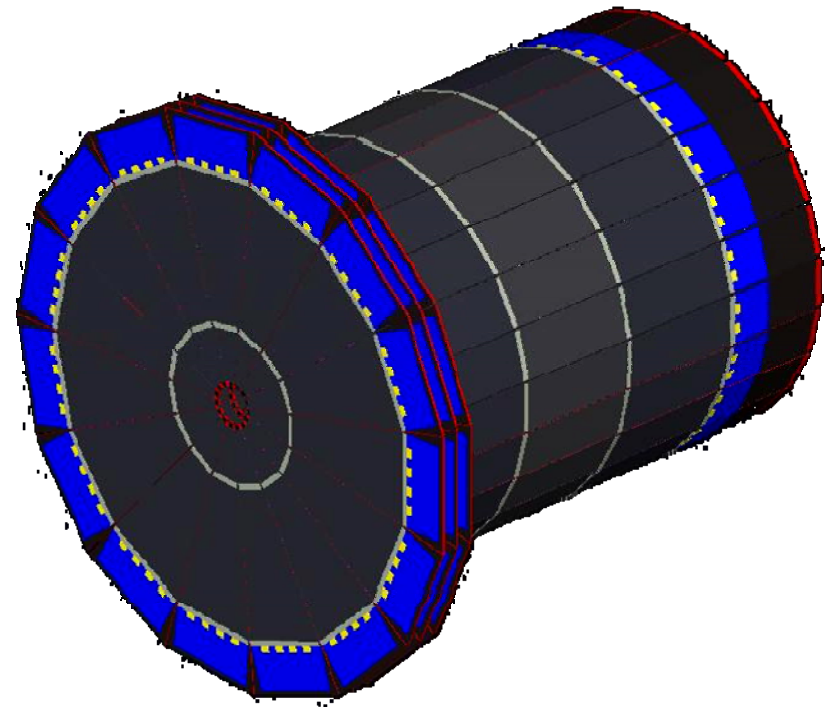
# Geometry Overview

# Geometry: FST Implementation

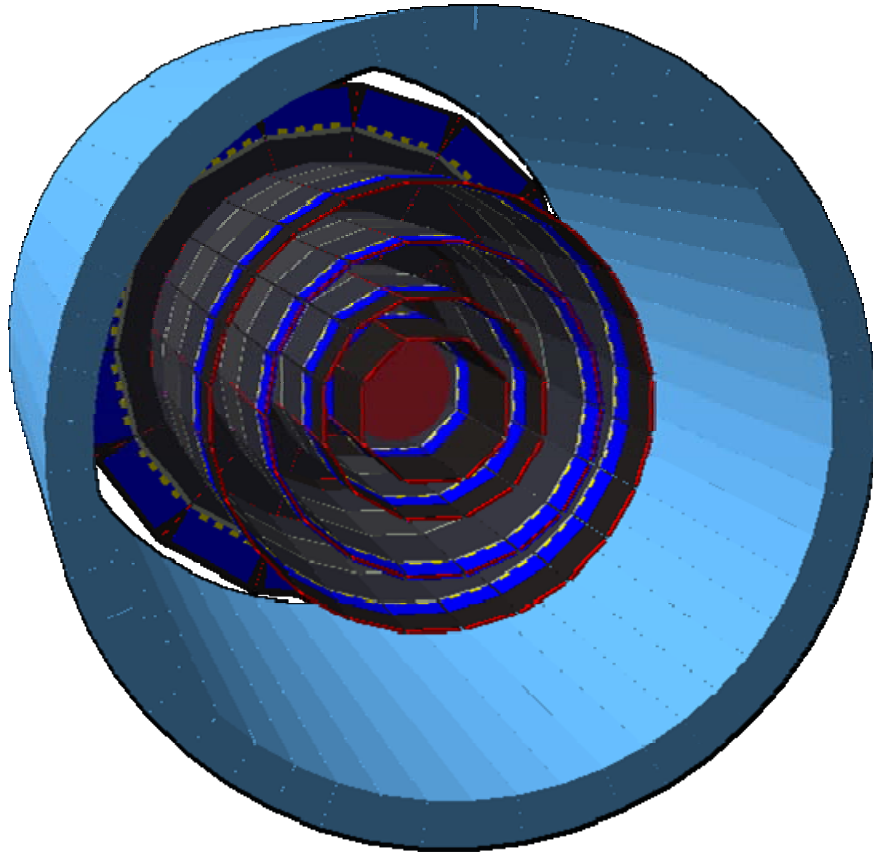
Engineering Design



Geant4 Implementation



# Geometry: CTOF

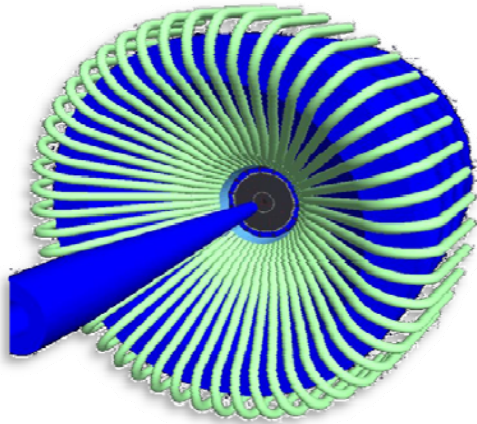


50 Trapezoidal Paddles

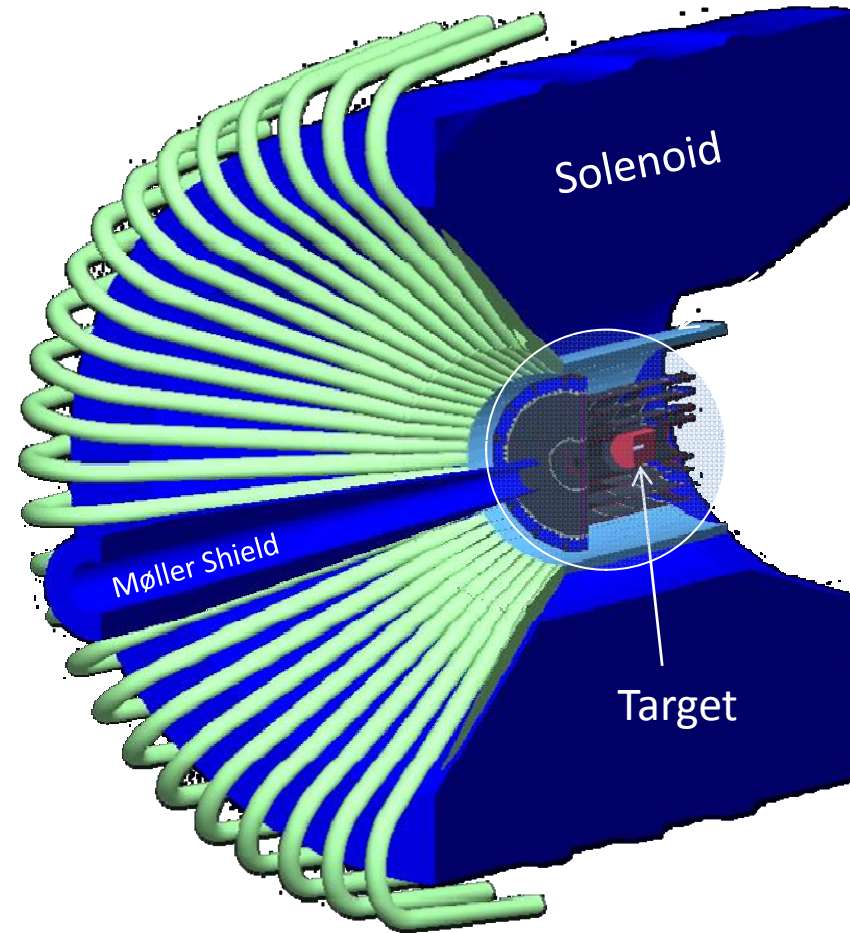
Thickness: 1.24''

Geometry Parameters:  
Same as designers

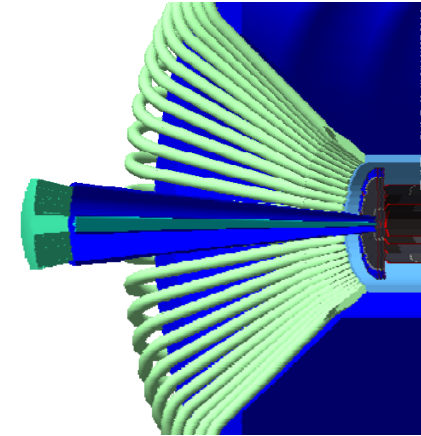
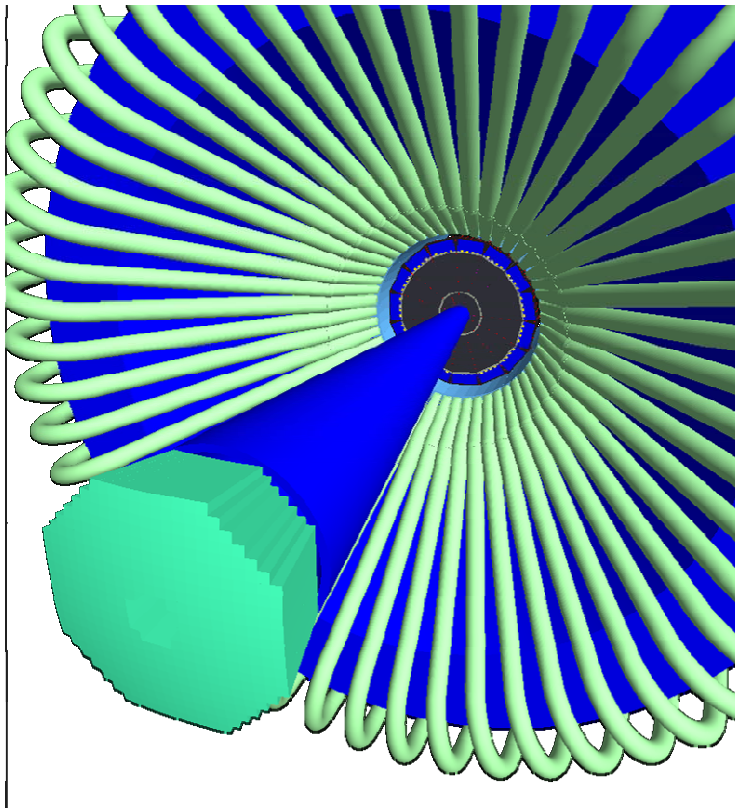
# Geometry: Central Detector



SVT  
CTOF  
Light Guides  
Solenoid  
Møller Shield

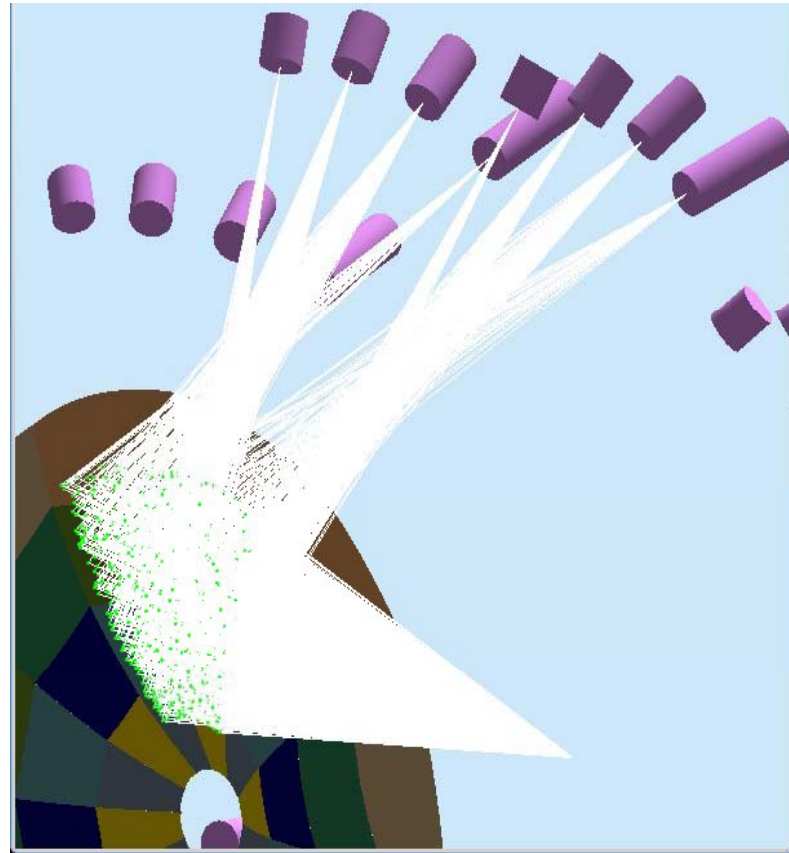
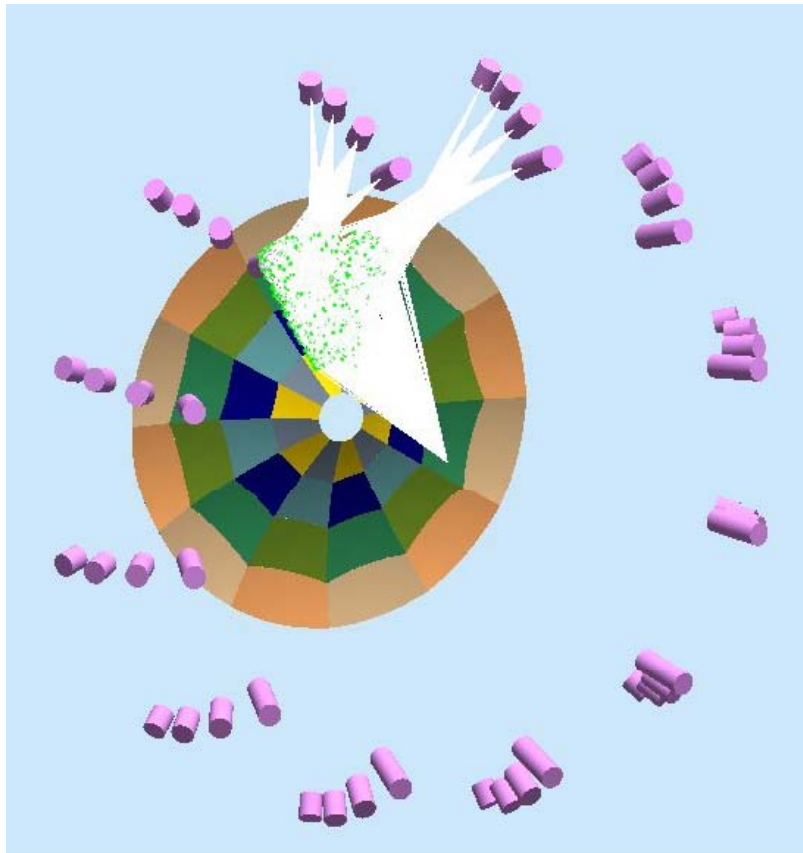


# Geometry: FT Configuration

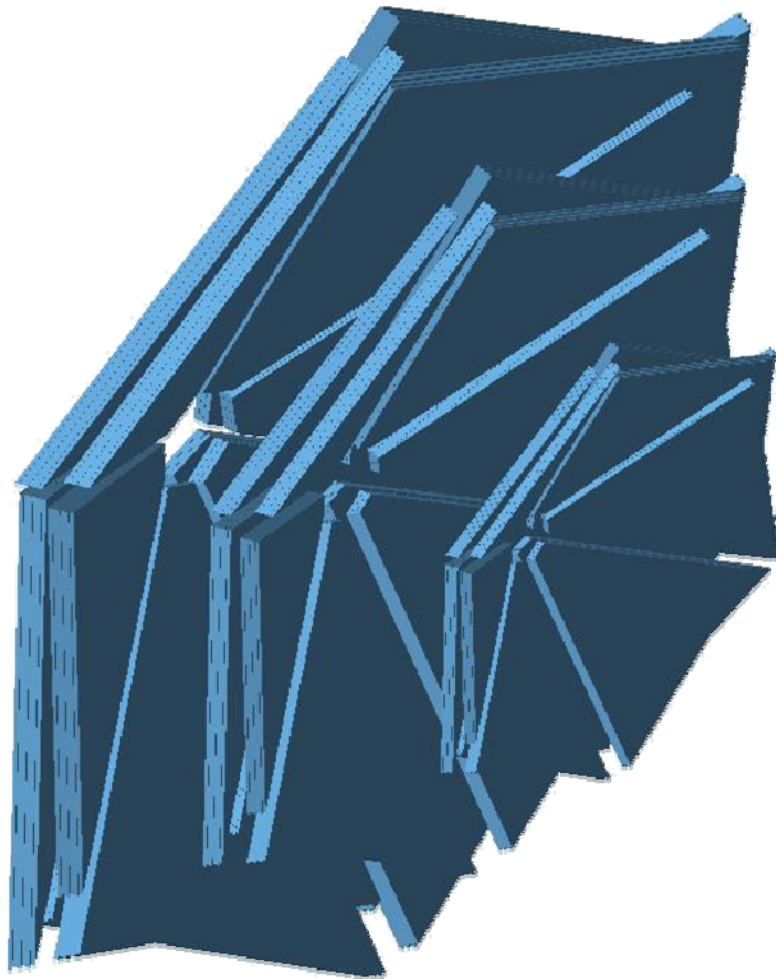


1.5 and 5.5 deg  
shields  
allow DVCS  
gammas into IC  
(optimization in progress)

# HTCC in GEMC



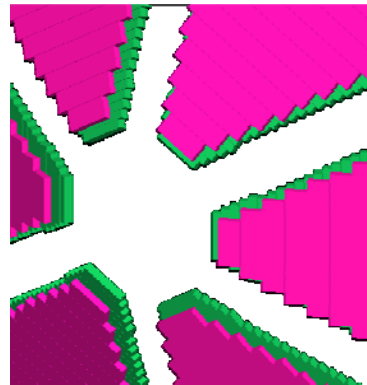
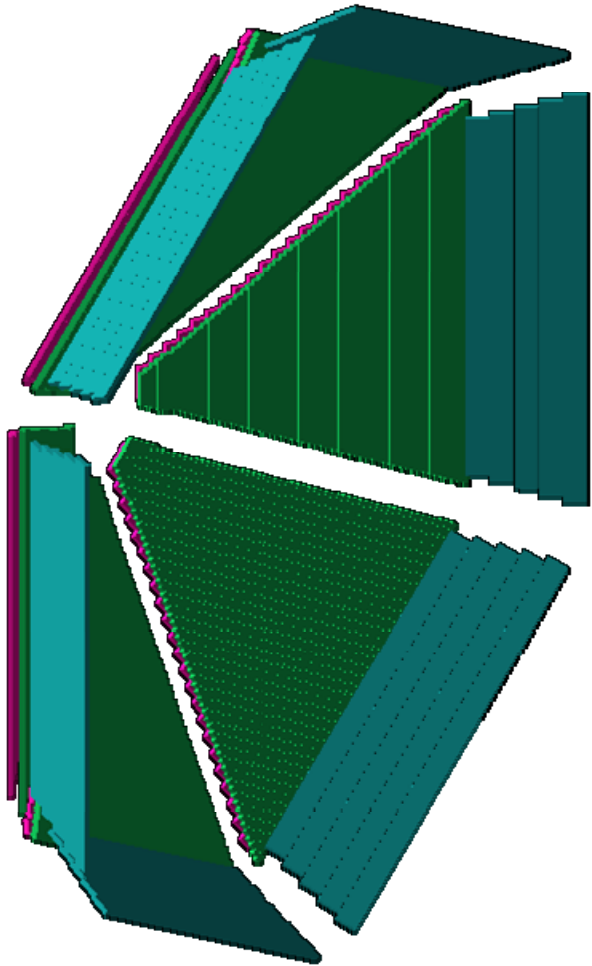
# Geometry: DC



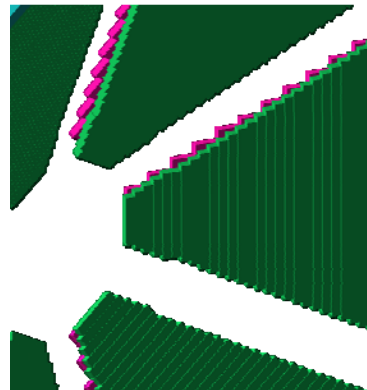
3 Regions  
2 SuperLayers per Region  
6 Layers per SuperLayer

Geometry Parameters:  
same as designers

# Geometry: FTOF



Panel 1a:  
23 paddles



Panel 1B:  
58 paddles

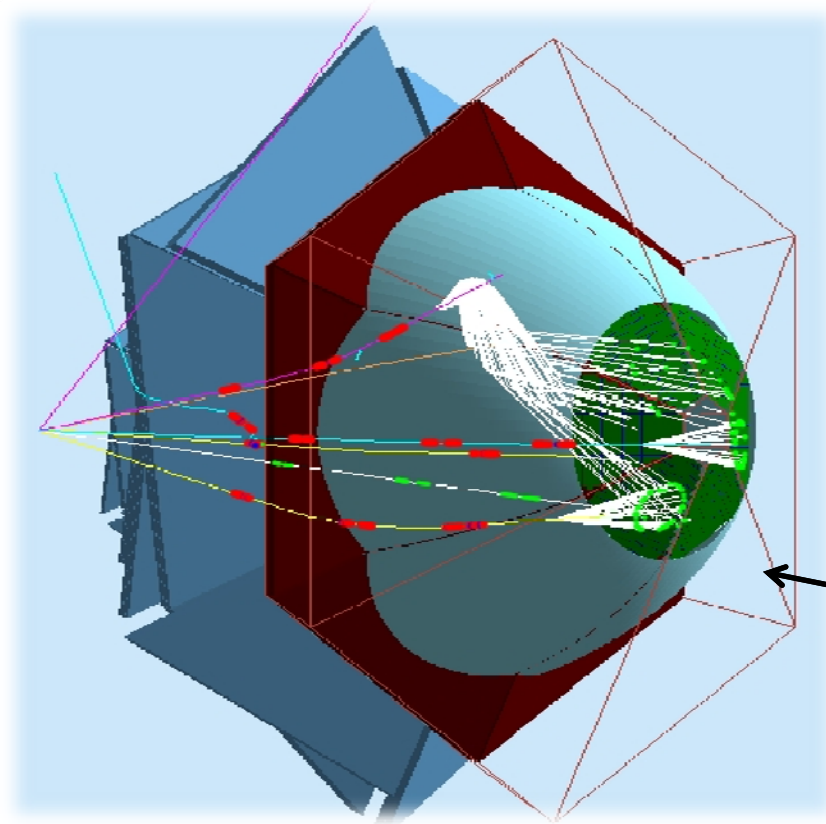
Panel 2B:  
5 paddles



# RICH for CLAS12

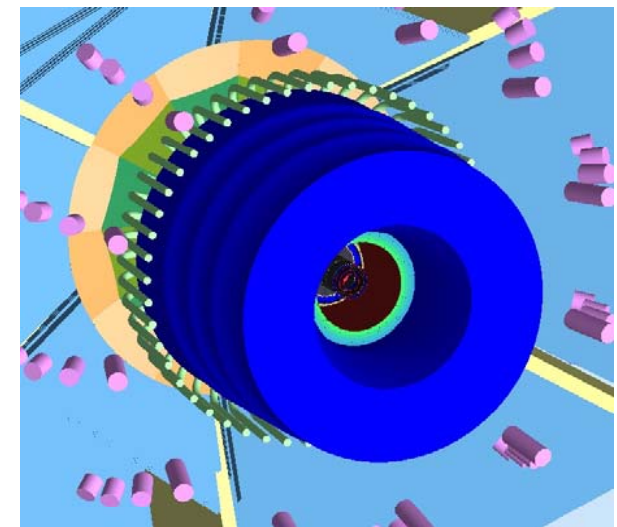
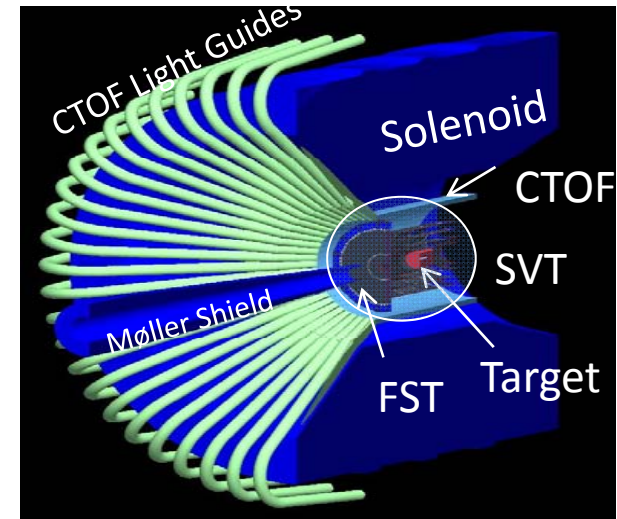
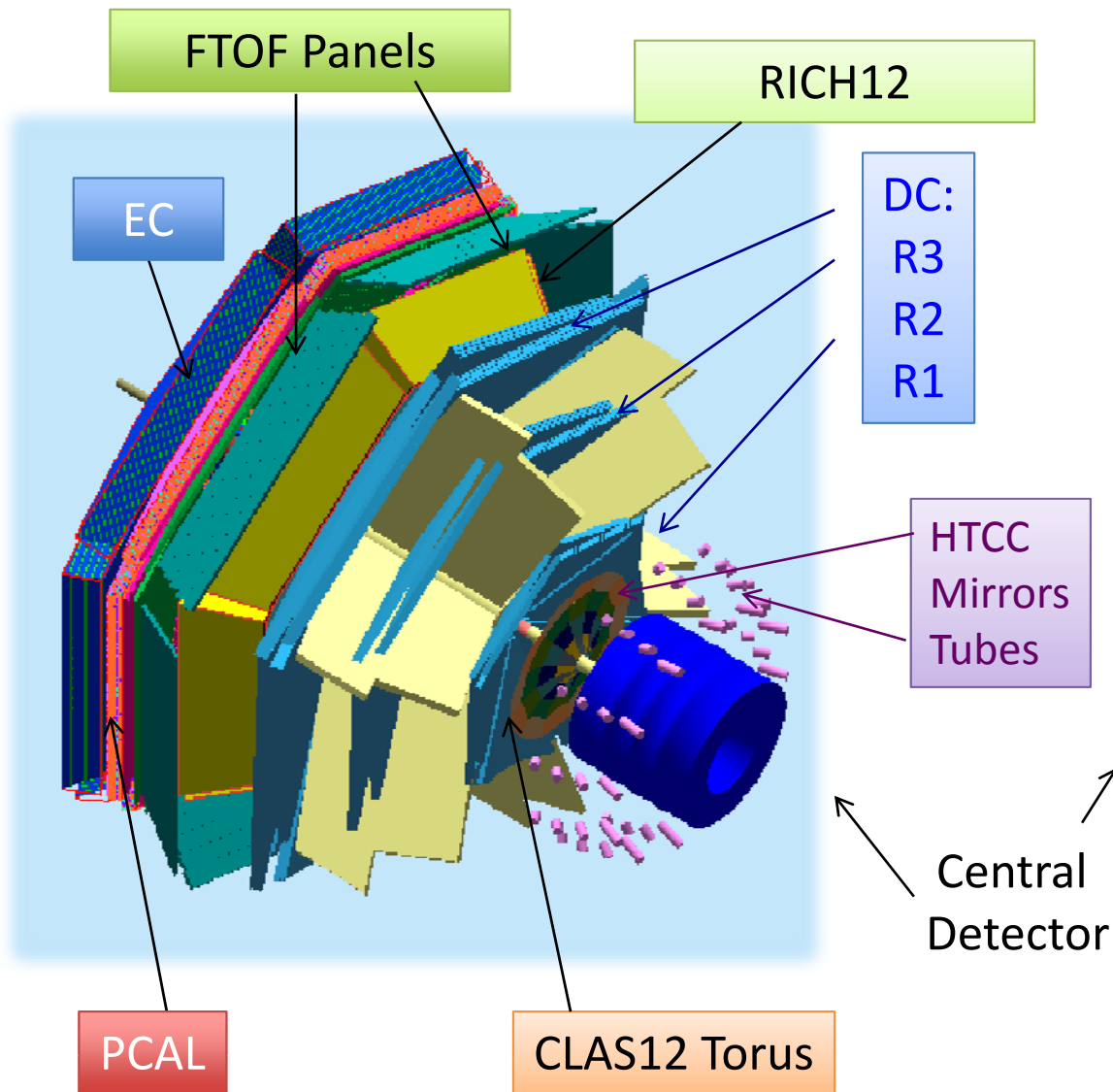
## RICH Detector

Rossi Patrizia, Osipenko  
Mikhail, Nathan Harrison,  
Pappalardo Luciano, Cisbani  
Evaristo, Contalbrigo Marco,  
Alaoui Ahmed, Maurizio  
Ungaro, De Leo Raffaele,  
Hafidi Kawtar

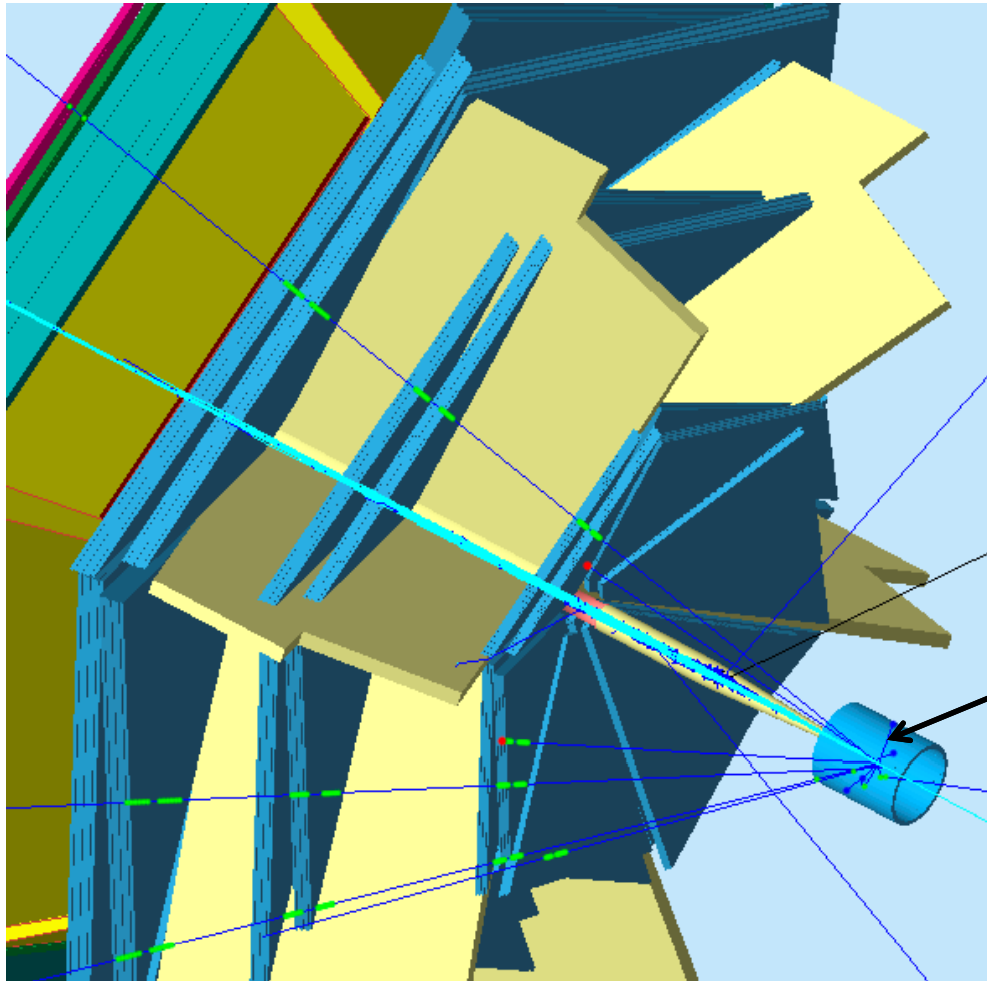


Pions, kaons producing rings  
directly on PMTs, or reflected by  
mirrors

# CLAS12 Geometry



# Background Studies



$10^{35}$  Luminosity  
5 cm LH2 Target  
 $4.7 \cdot 10^{11}$  e-/s

120K e- (250ns Window)

2K e-

# Magnetic Fields

Definitions in DB

Association to volume by name

Map based

Currently:

Torus

Solenoid

Transverse Polarized Target

Dipoles (Frascati Magnets, Pair Spectrometer)

# Summary: gemc.jlab.org

**GEMC**

Home Detectors Documentation Gallery Support Blog Web Interface

**Latest News**

**7/29/2011:**  
GDML output support completed.

**7/25/2011:**  
Particles that do not release energy are not saved by default. Use RECORD\_PASSBY to record them

**7/19/2011:**  
Added **Signal** interface: displays the single step informations for each hit, and graphs of energy deposited.

[News Archive](#)

**GEant4 Monte-Carlo**

GEMC is a framework based on [Geant4 Libraries](#) to simulate the passage of particles through matters.

The idea behind GEMC is to store parameters such as the geometry in an external database so that no numbers is hardcoded in the software.

**The Database**

gemc supports *mysql* as the external database.

**GEMC**

**GEant4 Monte-Carlo**

EVIO input/output  
MYSQL Database  
C++ Objects and Factories  
QT4 Graphical Interface  
CLHEP Library

Getting Started with GEMC

CLAS12 Detectors

Main Page Data Structures Files

**GEMC Documentation**

**GEMC**

**Overview**

gemc (**GEant4 MonteCarlo**) is a simulation software based on [Geant4](#).

The simulation parameters are external to the software:  
Geometry, Materials, Fields, Banks definitions are stored in an external database.  
The Factory Method is used for the Hit Processes/Digitization Routines and for the Input/Output formats.

**The Database**

gemc supports *mysql* as the external database.  
The informations stored in the database determine:

- The Geometry.
- Sensitive Detectors definitions (including Thresholds, Time Window, Production Cut).
- The Hit Process Factories.
- The Bank Output Format.
- The Magnetic Fields definitions.

**Software Requirement:**

- geant4 (simulation libraries)
- clhep (random generators, physics vectors, geometry and linear algebra libraries)
- qt4 (graphic libraries)
- mysql
- scon (build system)

**Platforms Supported:**