



# **FAIR Simulation&Analysis Framework FairRoot**

Mohammad Al-Turany (IT-GSI)

Denis Bertini (IT-GSI)

Florian Uhlig (IT-GSI)

Ilse König (Hades-GSI)

# Overview

- Motivations
- FairRoot Features
  - Geometry Interface
  - Runtime Database and Parameter Handling
  - Integrated Track follower (Geane)
- FAIR experiments design studies
  - CBM
  - PANDA
  - HADES
- Summary

# Fair@Gsi

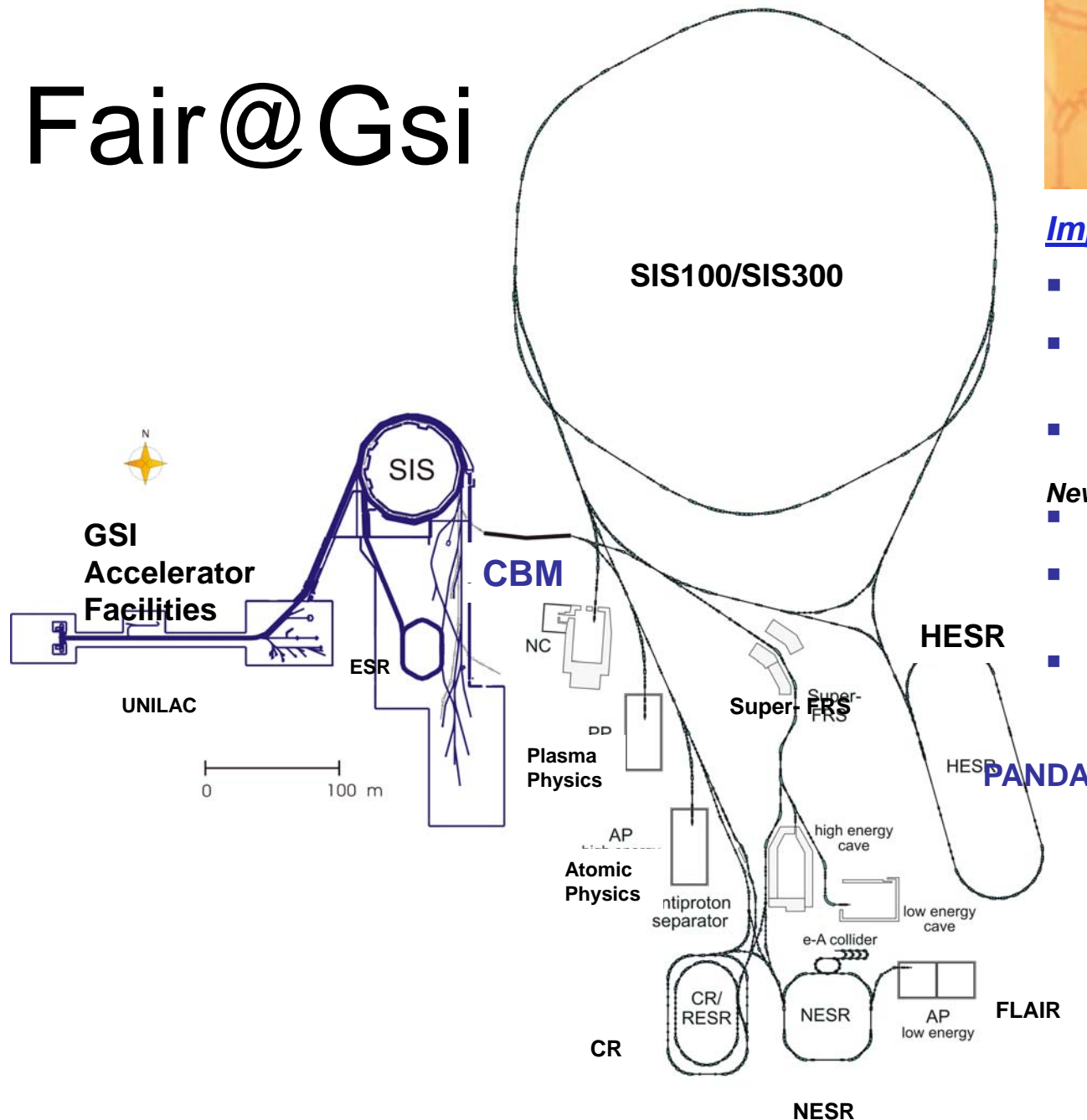


## Improvements:

- Primary beam intensity **100 – 1000 x**
- Secondary beam intensity for radioactive nucl.: up to **10 000x**
- Ion energy: **20 x**

## New features:

- **fast pulsed *supraconducting* Magnet**
- „**cooled**“ Antiproton beam up to 15 GeV
- Specific: intense „**cooled**“ beam radioactive ,exotic nucleus



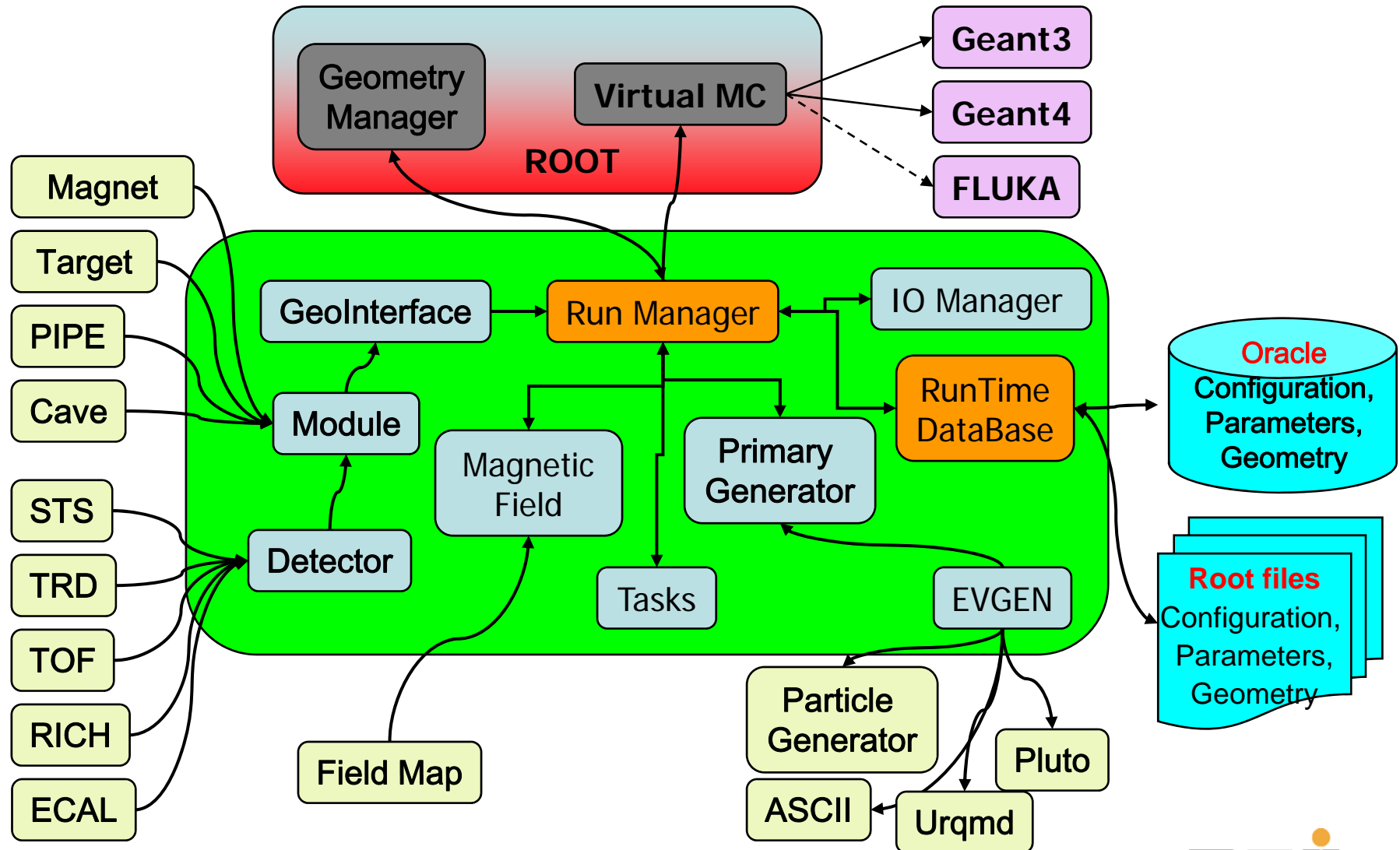
# Motivations

- **Which simulation engine to choose?**
  - Need to move to modern and maintained MC: **GEANT4**
  - Need for
    - Working fast ! ( LOI, TDR deadlines ... )
    - Making reliable simulation
  - Usually: better knowledge of “old” MC’s: **GEANT3, FLUKA ...**
  - A cross-check of simulation results between different MC is needed
    - Better understanding of GEANT4 ( intrinsic cuts / physics list ...)
    - Preparing for full simulation
- ➔
- **Use of VMC (Virtual Monte Carlo )** : an interface between MCs
  - With the same code, the user can switch between different MCs

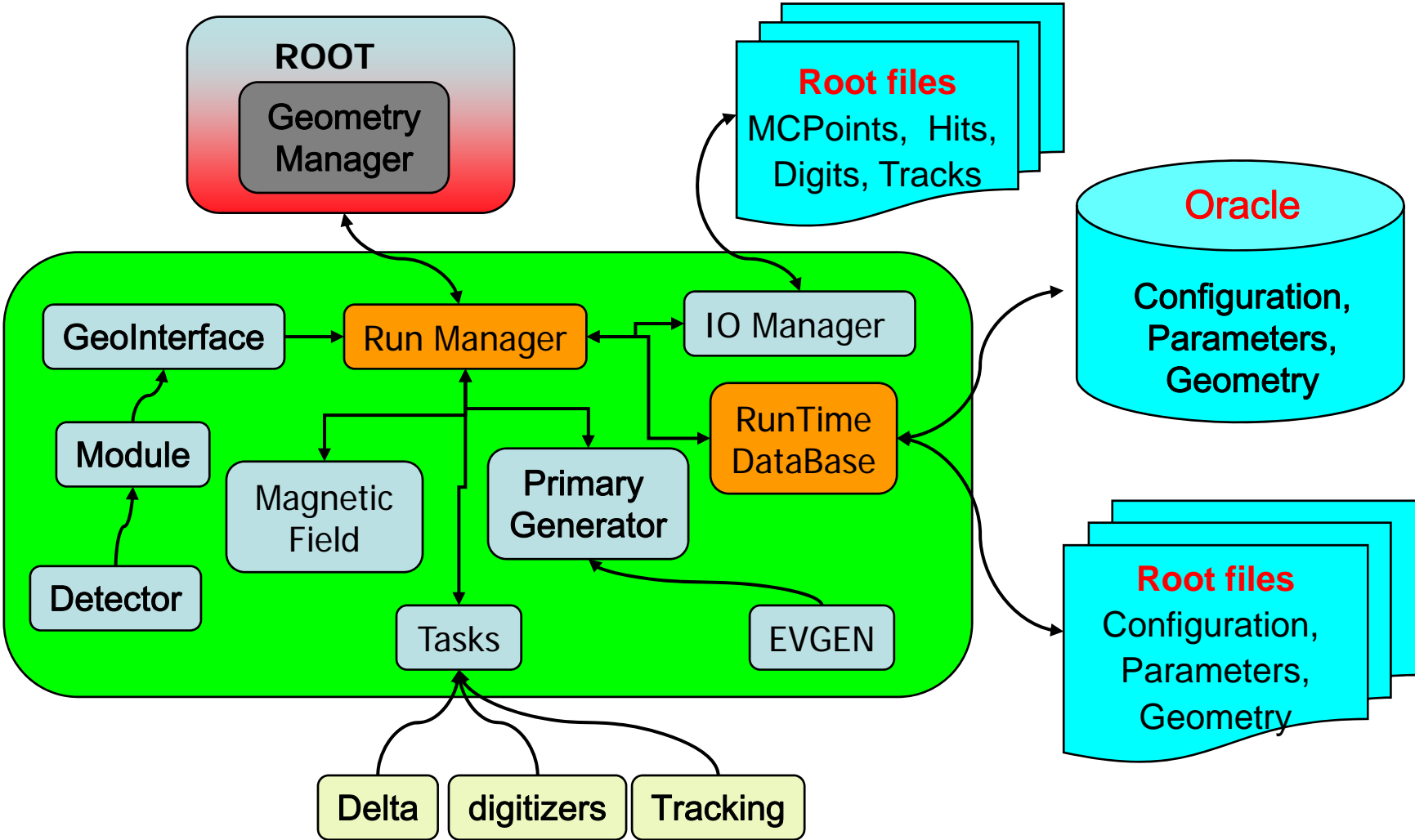
# Features

- **The same framework** can be used for Simulation and Analysis
- **Fully ROOT based:**
  - **VMC** and **VGM** for simulation
  - **IO scheme** (TChain, friend TTrees, TFolders ) for persistency
  - **TTask** to organize the analysis data flow
  - Use of **TGeoManager** for Simulation and Reconstruction
- **Completely configurable** via ROOT macros
- **Easy to maintain** (only ROOT standard services are used)
- **Use of a Geometry Interface.**
  - **G3 Native geometry**
  - **Geometry Modeller (TGeoManager)**
  - **Different geometry input format**

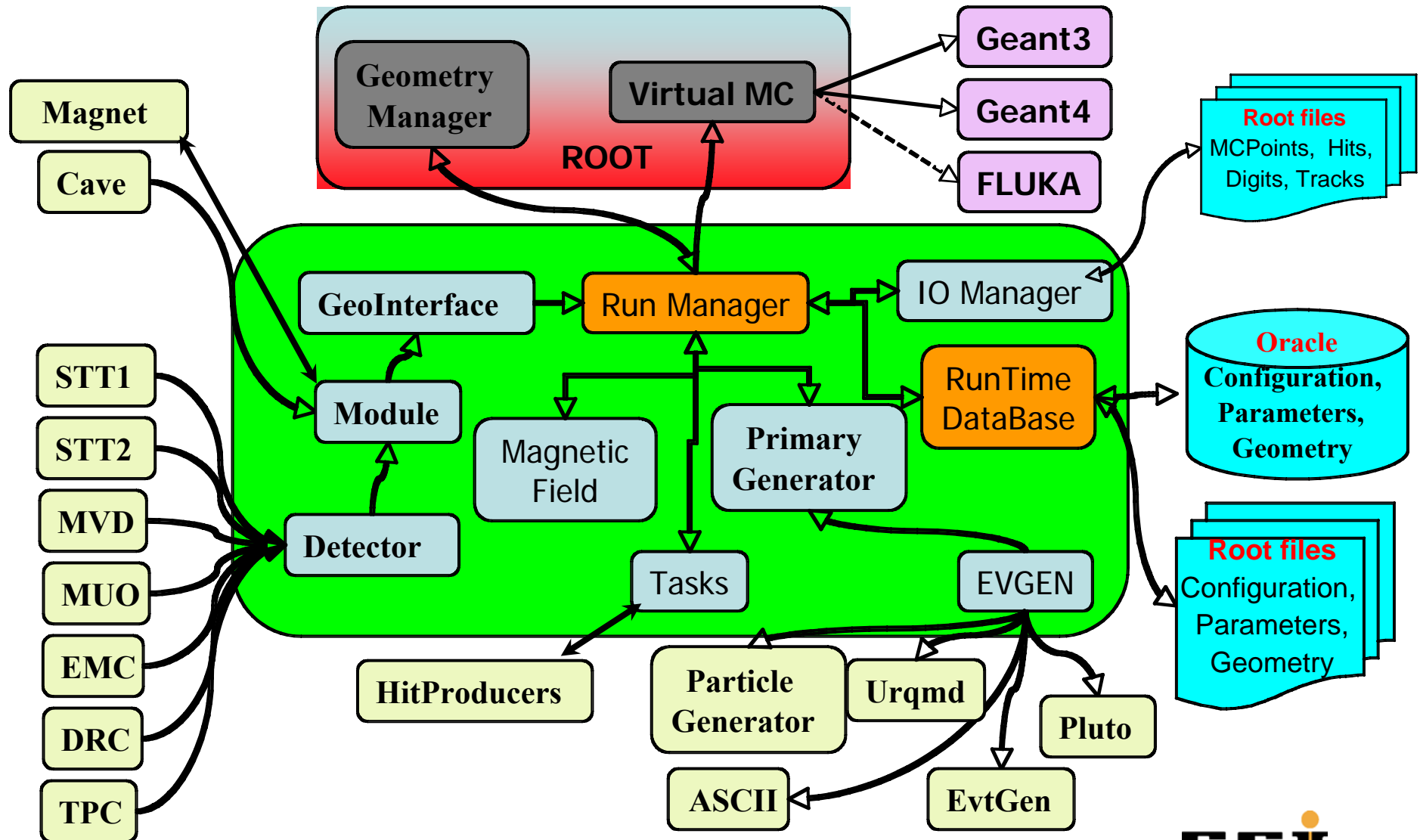
# CBM Simulation



# CBM Analysis



# Panda Simulation

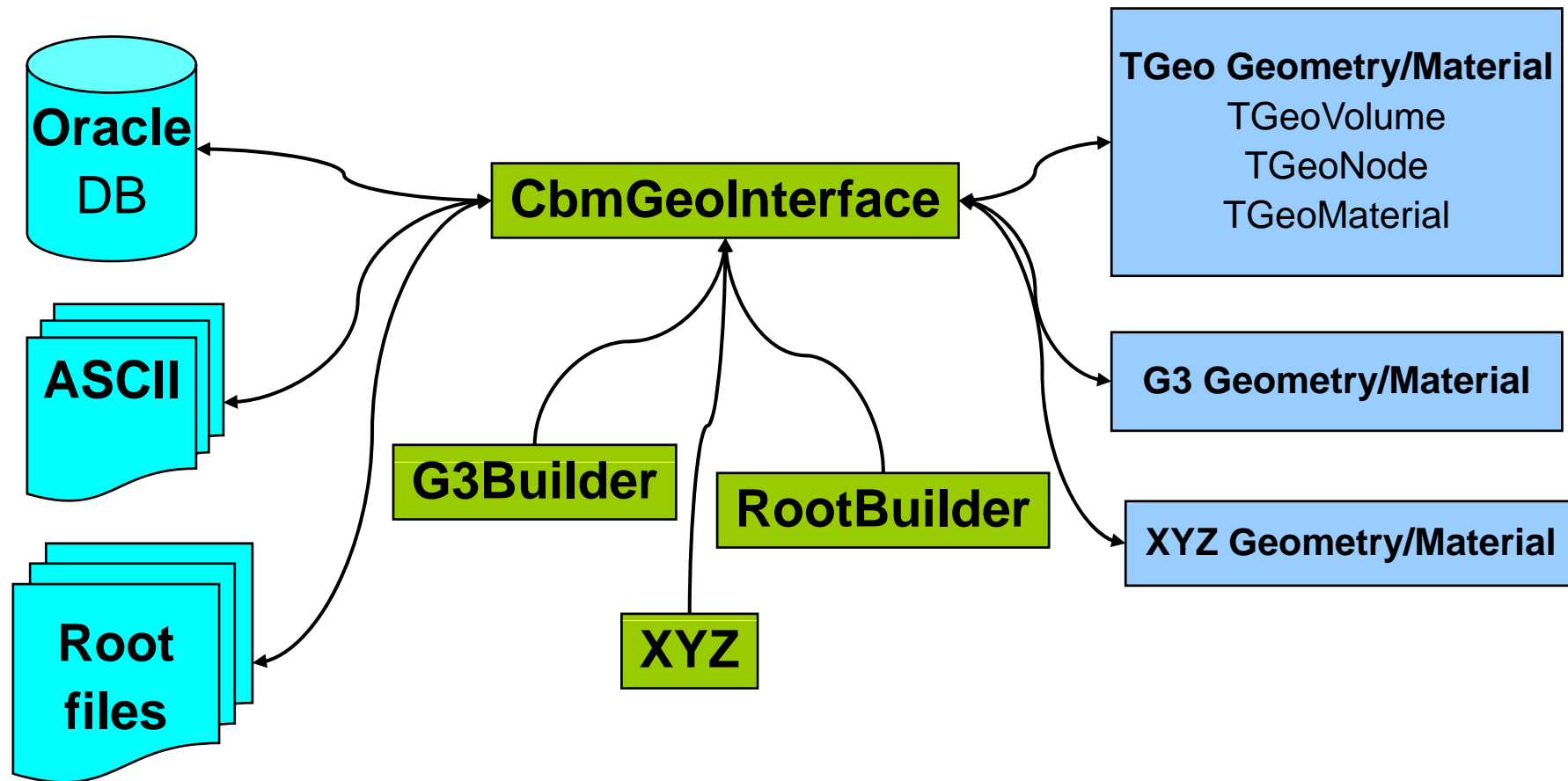




# Geometry Interface

- **Advantage:**
  - more flexibility : different inputs can be used.
  - closer to technical drawings and analysis coordinate systems
- **Oracle interface**
  - Hades geometry table design reusable

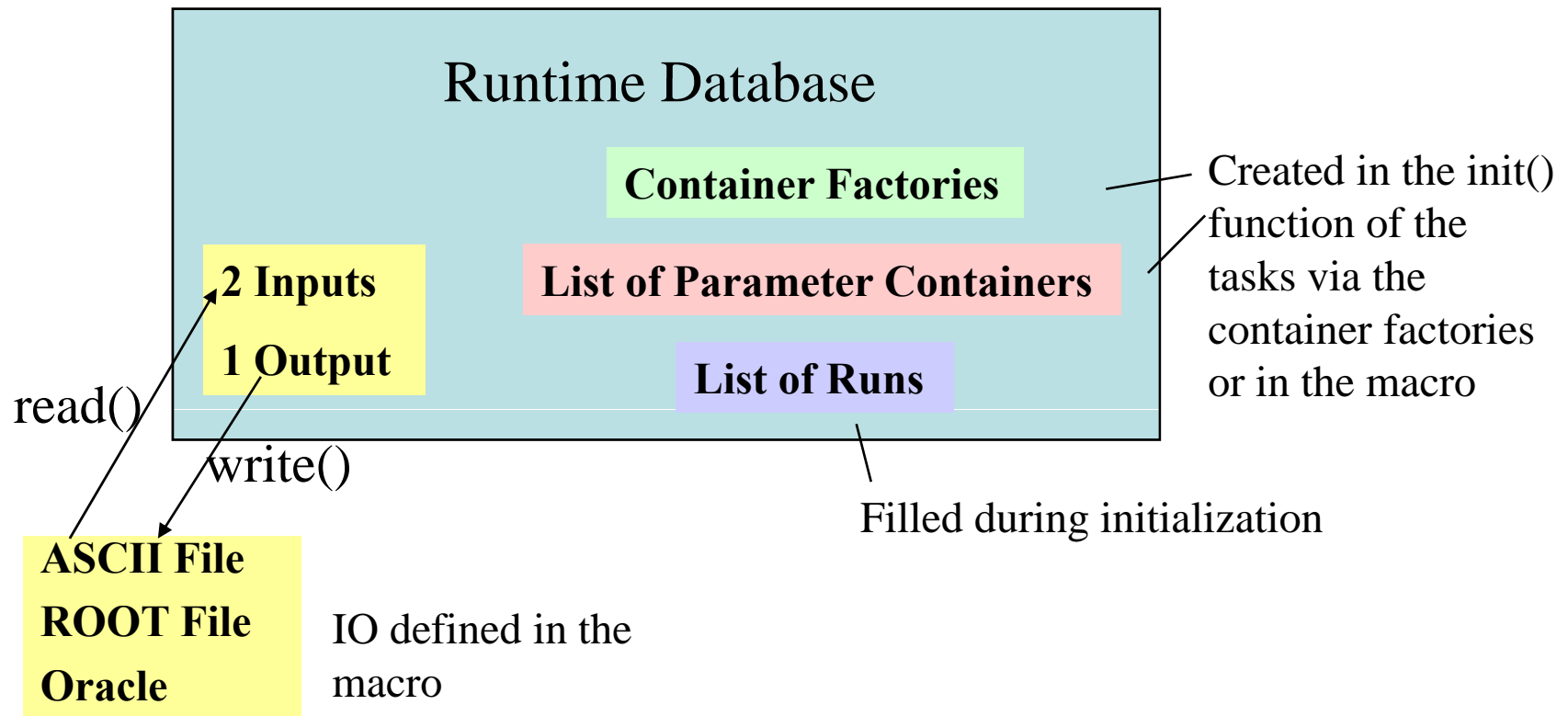
# Material & Geometry Interface



# Runtime Database

The Runtime Database is the manager class for all Parameter containers:

**Creation, Initialization, Output**



# Version management in Oracle

- For time dependent information a version management is needed which fulfills the following requirements:
  - It must be possible to get a consistent set of information for any date (e.g.the start time of a certain run).
  - To **preserve the history**, no information - even if wrong - should be overwritten without trace, which means that **only inserts** should be made, **no deletes nor updates**.
  - It must be possible to get an answer to the question: '**Which parameters were used when analyzing this run X years ago?**' (The calibration might have been optimized several times since this date. Maybe some bugs have been detected and corrected in the mean time.)

# Version management in Oracle

Time dependant entries have a time stamp (date + time with the precision of one second) in form of three columns (Format: DATE):

- **valid\_since** :First date when the entry is valid.
- **valid\_until** :Last date when the entry is still valid
- **invalid\_since** :Date when the entry is replaced by a correct entry or a better version in case of e.g. calibration parameters and therefore gets invalid.

# Example: *CbmGenericParSet*

## Base class for most parameter containers

**Advantage:** only a few lines of code to be implemented  
all I/O interfaces exist already

Allows to store various types of parameters (handled by *CbmParamList*):

int, float, double, strings,...

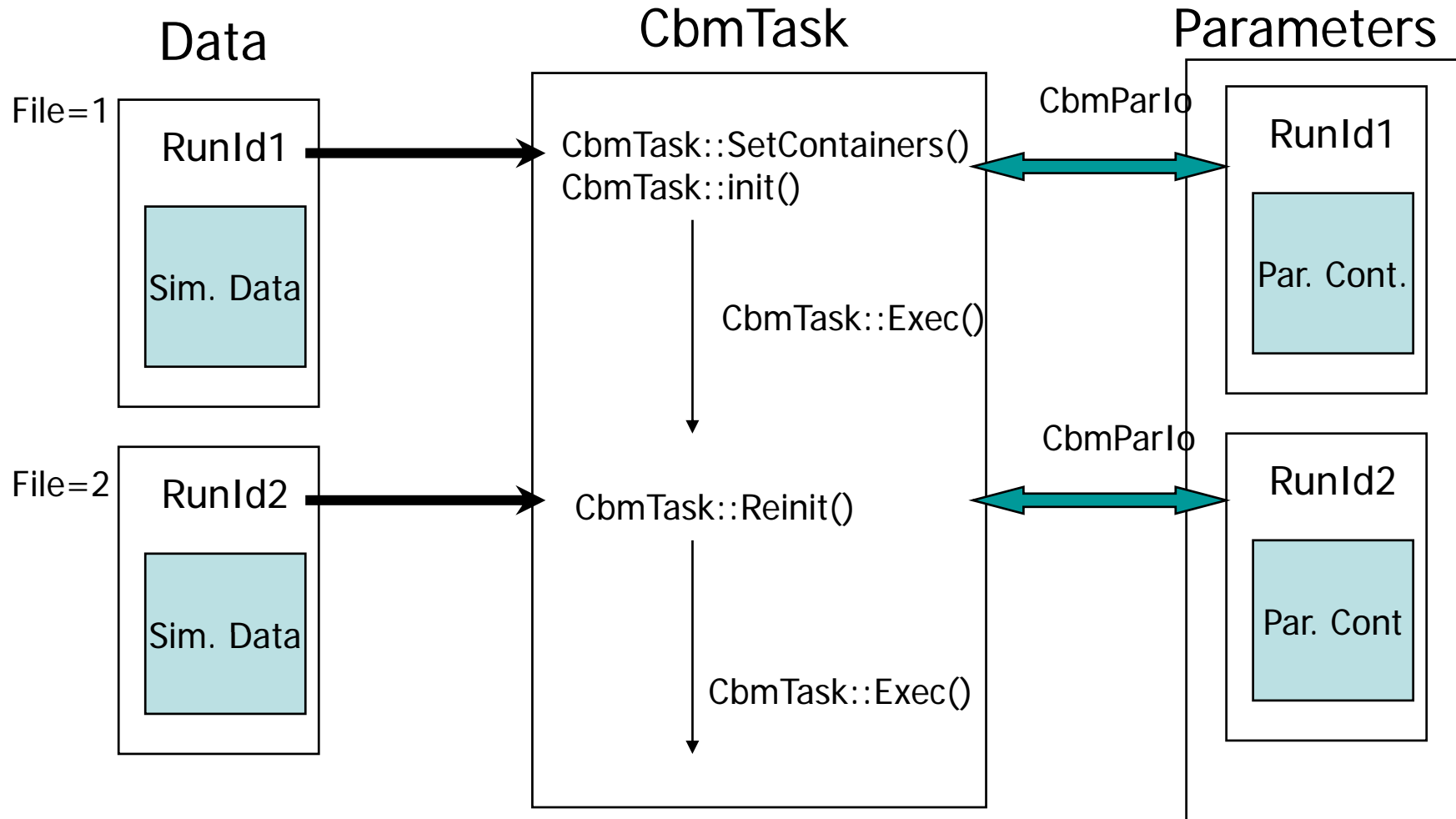
arrays

TObjects (classes, histograms, ...)

in derived class  
implemented

```
class CbmParGenericSet : public CbmParSet {  
public:  
    CbmParGenericSet(const char* name,const char* title,const char*  
        context)  
        : CbmParSet(name,title,context) {}  
    virtual ~CbmParGenericSet() {}  
    virtual Bool_t init(CbmParlo*);  
    virtual Int_t write(CbmParlo*);  
    virtual void putParams(CbmParamList*)=0;  
    virtual Bool_t getParams(CbmParamList*)=0;  
    virtual void printParams();  
};
```

# Initialisation scheme (Analysis)



# Combined Chain& Friend

- **Make use of our CbmRootManager (IO):**
  - When the **TChain** switch to new file:
    - Clear the global list of friends
    - Add the correct next friend to the list
    - Update the corresponding pointers (TTree, Friend Tree ..)
- **Combining Chain with Friend:**

```
{ //...
```

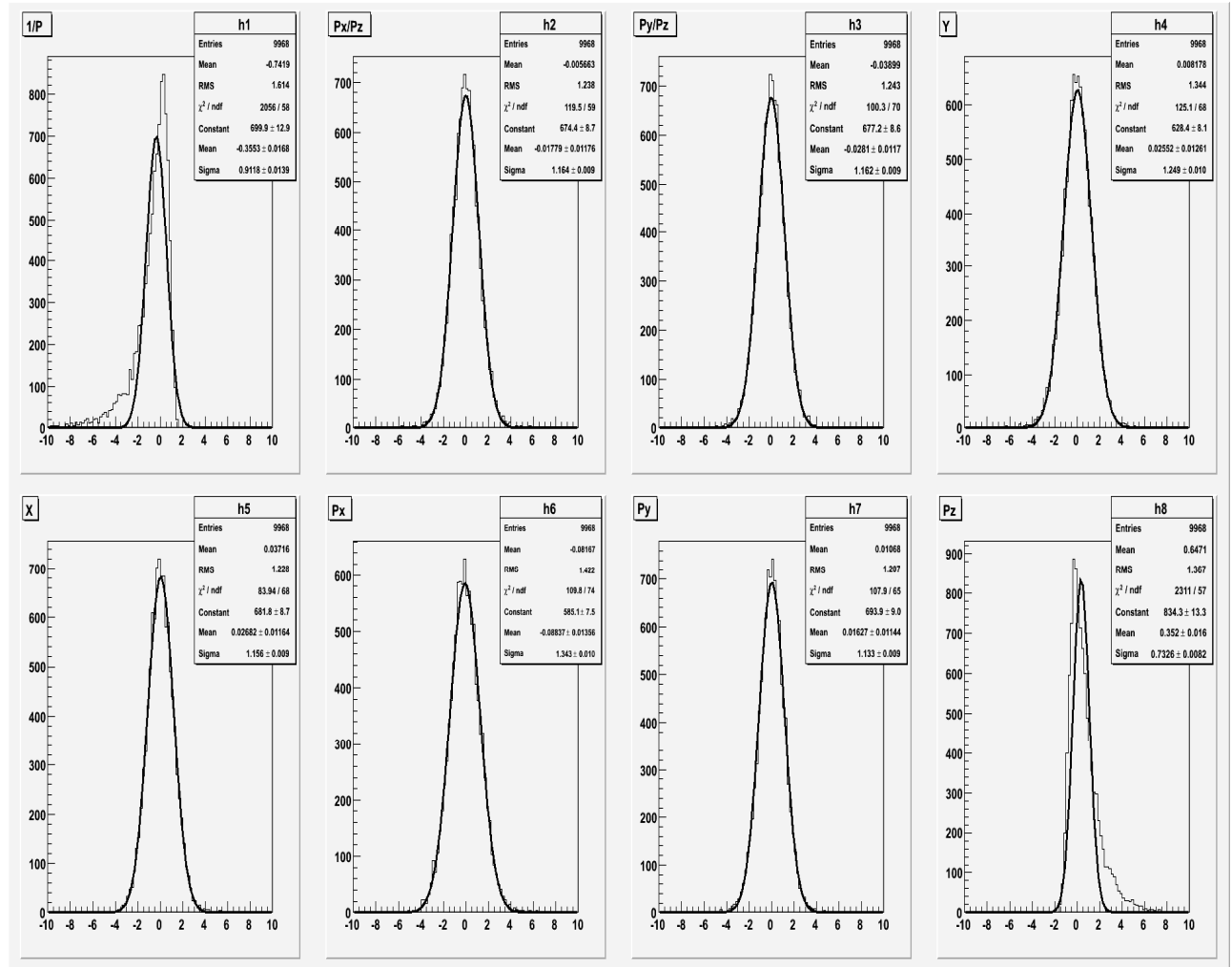
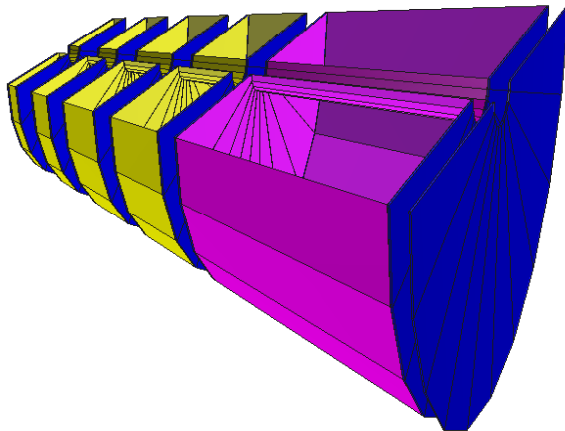
```
CbmRun *fRun = new CbmRun();  
fRun->SetInputFile("rich_hit1.root");  
fRun->AddFriend("Rguidance1.root");  
fRun->AddFile("rich_hit2.root");  
fRun->AddFriend("Rguidance2.root");  
//...  
fRun->Init();  
fRun->Run();  
... }
```



# Geane Integration in FairRoot

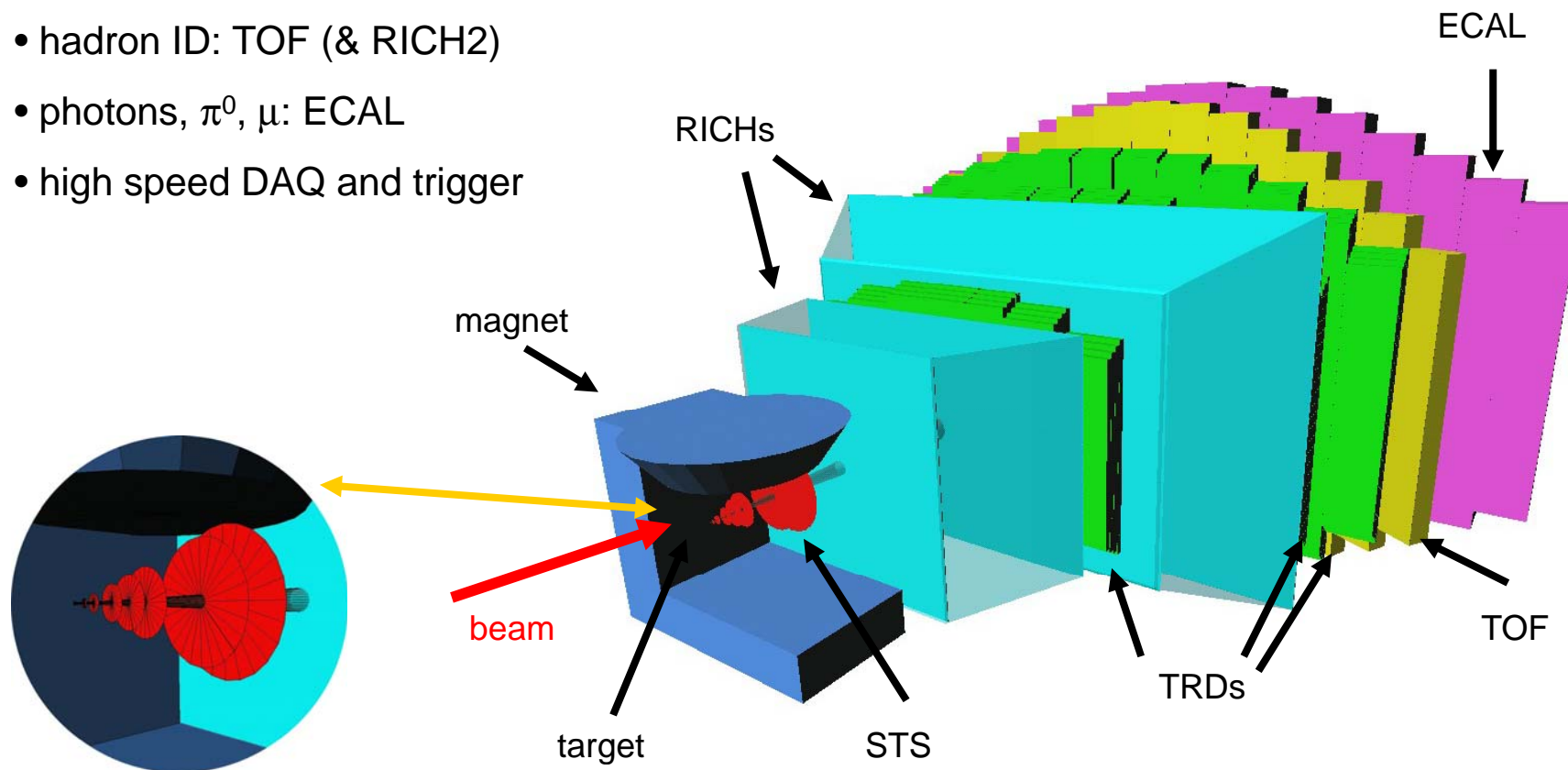
- The integration into the VMC (TGeant3) is done
- In FairRoot:
  - Geane can be used in the analysis or from macro
  - Propagation to
    - Length
    - Plane
    - Volume (Enter or Exit point)
  - CbmPoints and/or CbmTrackParam can be used as input for propagation
  - See talk of Andrea Fontana

# Muon Absorber in CBM

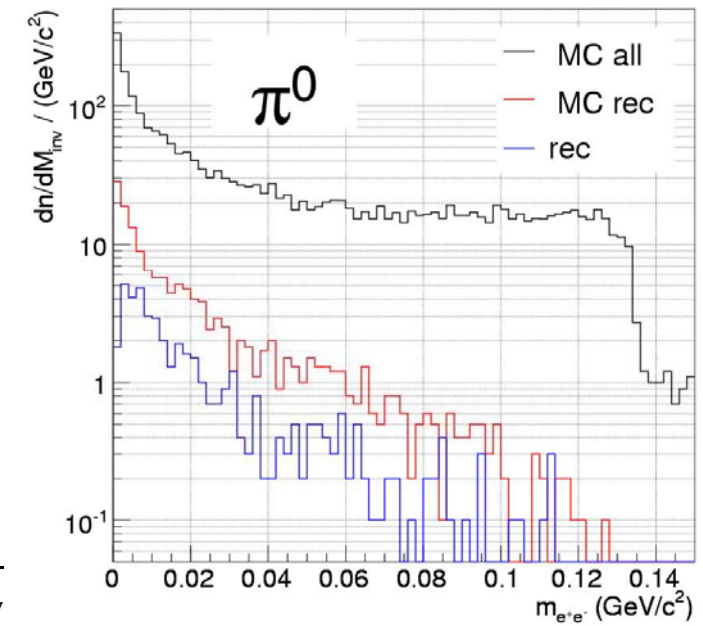
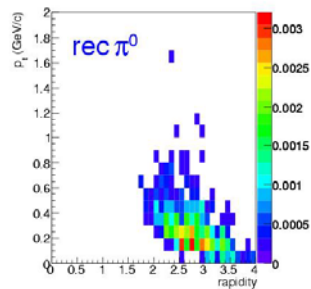
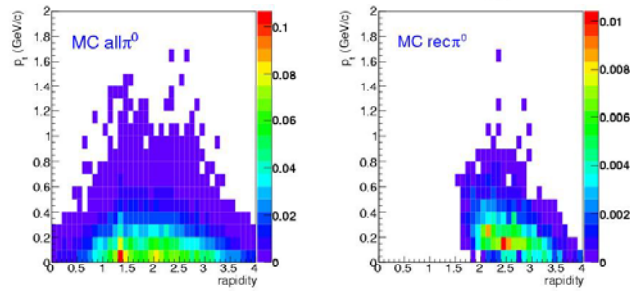
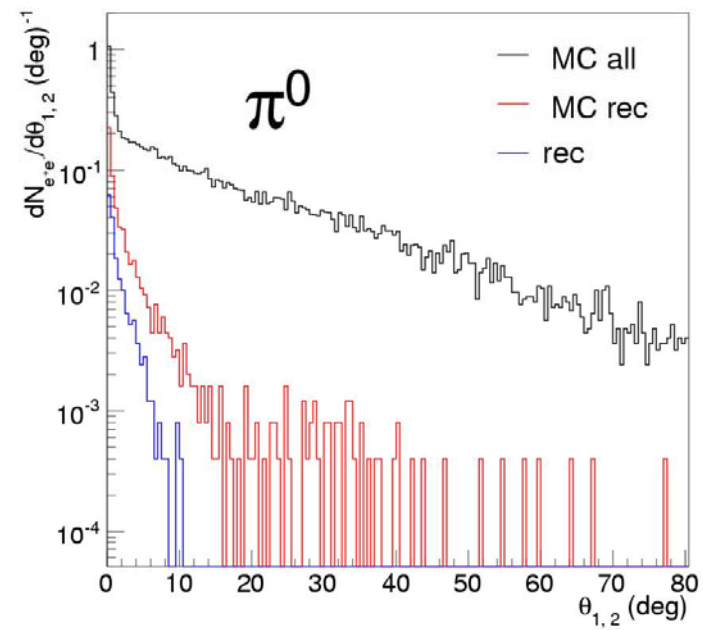
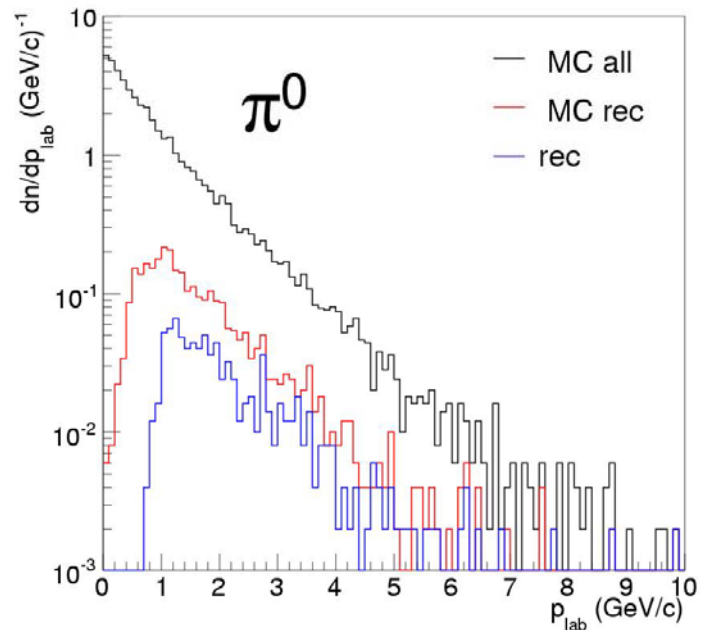


# CBM experiment@GSI

- tracking, vertex reconstruction: radiation hard silicon pixel/strip detectors (STS) in a magnetic dipole field
- electron ID: RICH1 & TRD (& ECAL)  $\rightarrow \pi$  suppression  $\geq 10^4$
- hadron ID: TOF (& RICH2)
- photons,  $\pi^0$ ,  $\mu$ : ECAL
- high speed DAQ and trigger

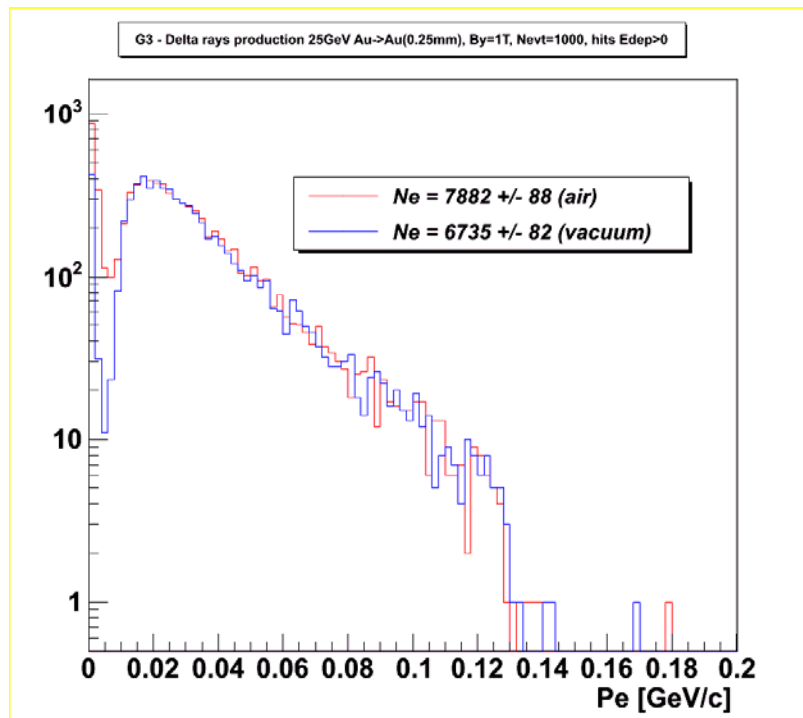


# Analysis: $\pi^0 \rightarrow \gamma e^+ e^-$

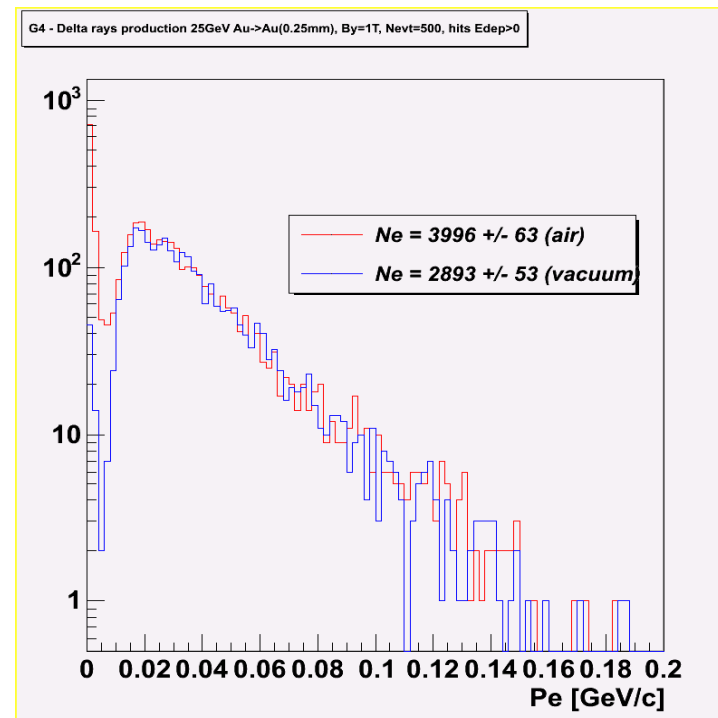


# CBM:KO electrons production in the Au target (setup in the air or in vacuum $B=1T$ )

Geant3



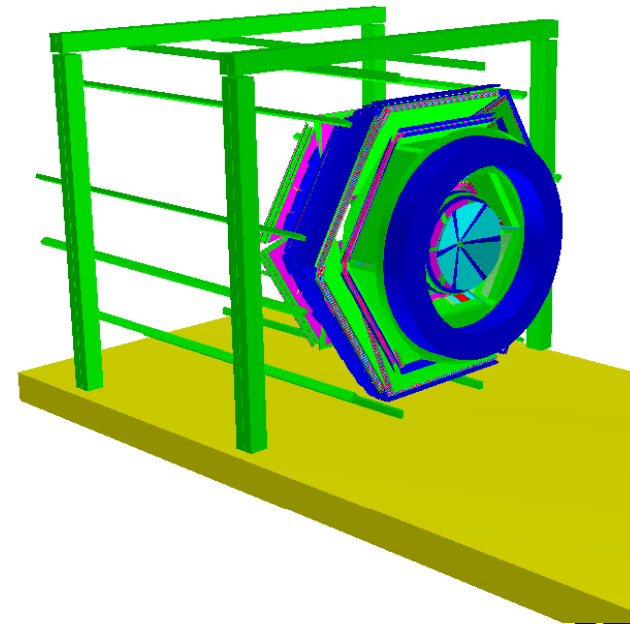
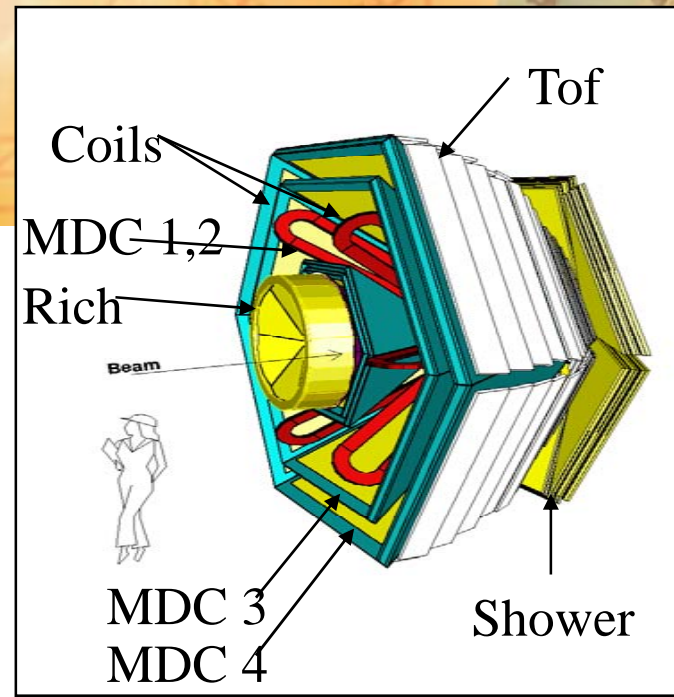
Geant4



7.9 e/HI @  $B=1T$

# Hades@GSI

- **Goal:** Study of in-medium modifications ( $\rho$ ,  $\omega$ ,  $\phi$ ) properties
  - Produced in A+A, p+A,  $\pi$ +A collisions
  - Di-electrons are used as probes:  $V \rightarrow e^+e^-$
- Hexagonal symmetry around the beam axis
- Geometrical acceptance of 40%
- Invariant mass resolution of 1%
- Operates at reaction rates up to  $10^6$  /s
  - $\Rightarrow$  0.5 - 1 Tbyte/year
- ~ 70.000 readout channels



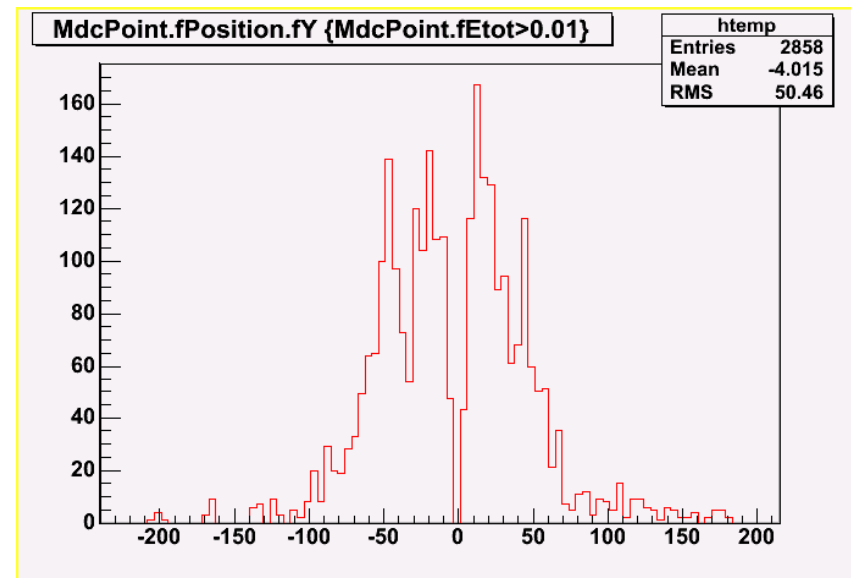
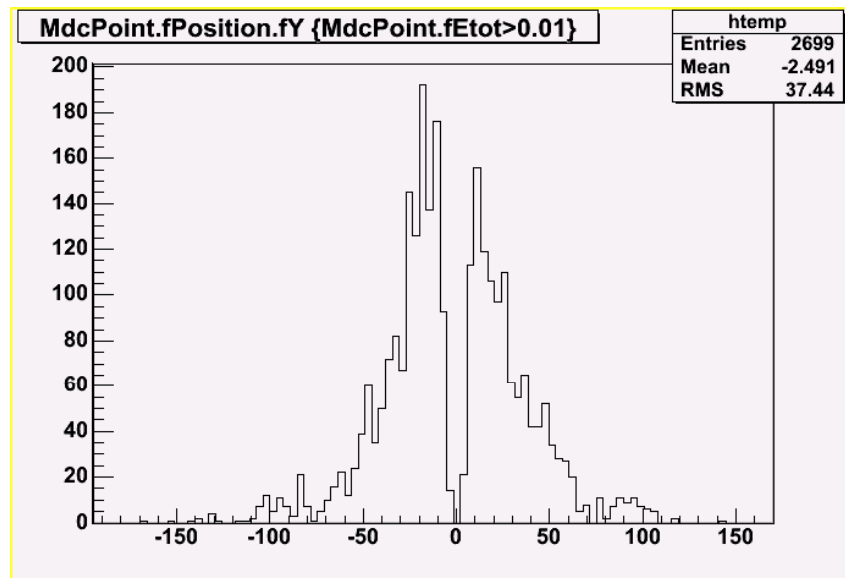
# Hades example

- **Need to simulate heavy system at High energy**
  - **Need external stack for Geant3: internal stack capacity reached)**
  - **Check data with geant4**
- **Easy reuse of FairRoot framework services**
- **Interesting case !**
  - Gives us the opportunity to tune Geant4 (cuts/physics list ...)  
and compare with real Data !
  - Realistic test of the framework

# HADES: MDC Hit distribution

Geant3

Geant4



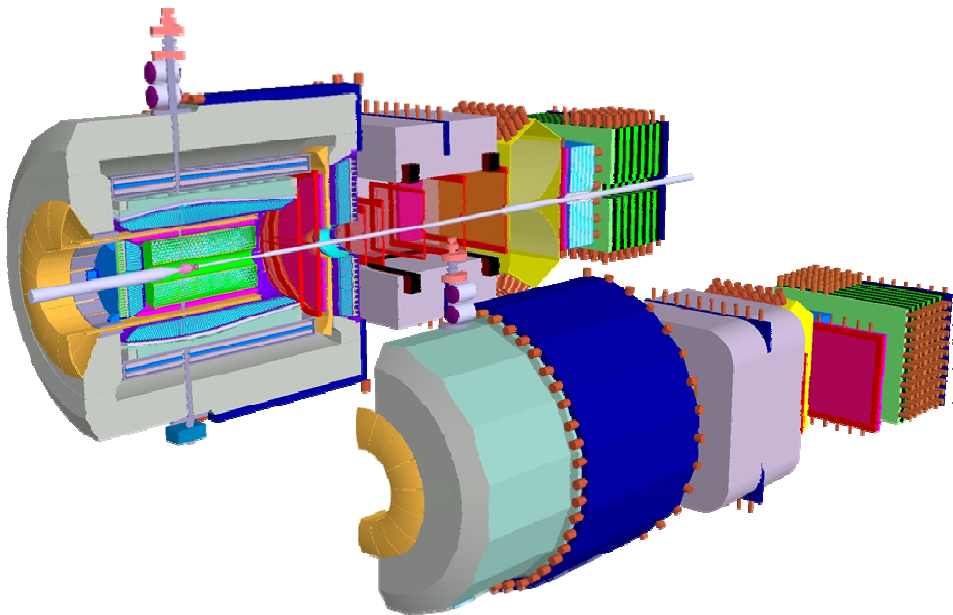
Cross-check with Hades data to be done !



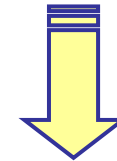
# The Panda experiment

Multi purpose detector at FAIR

$\bar{p}p, \bar{p}A$  collisions  
 $1.5 \Rightarrow 15 \text{ GeV}/c$  ( $\bar{p}$  momentum)

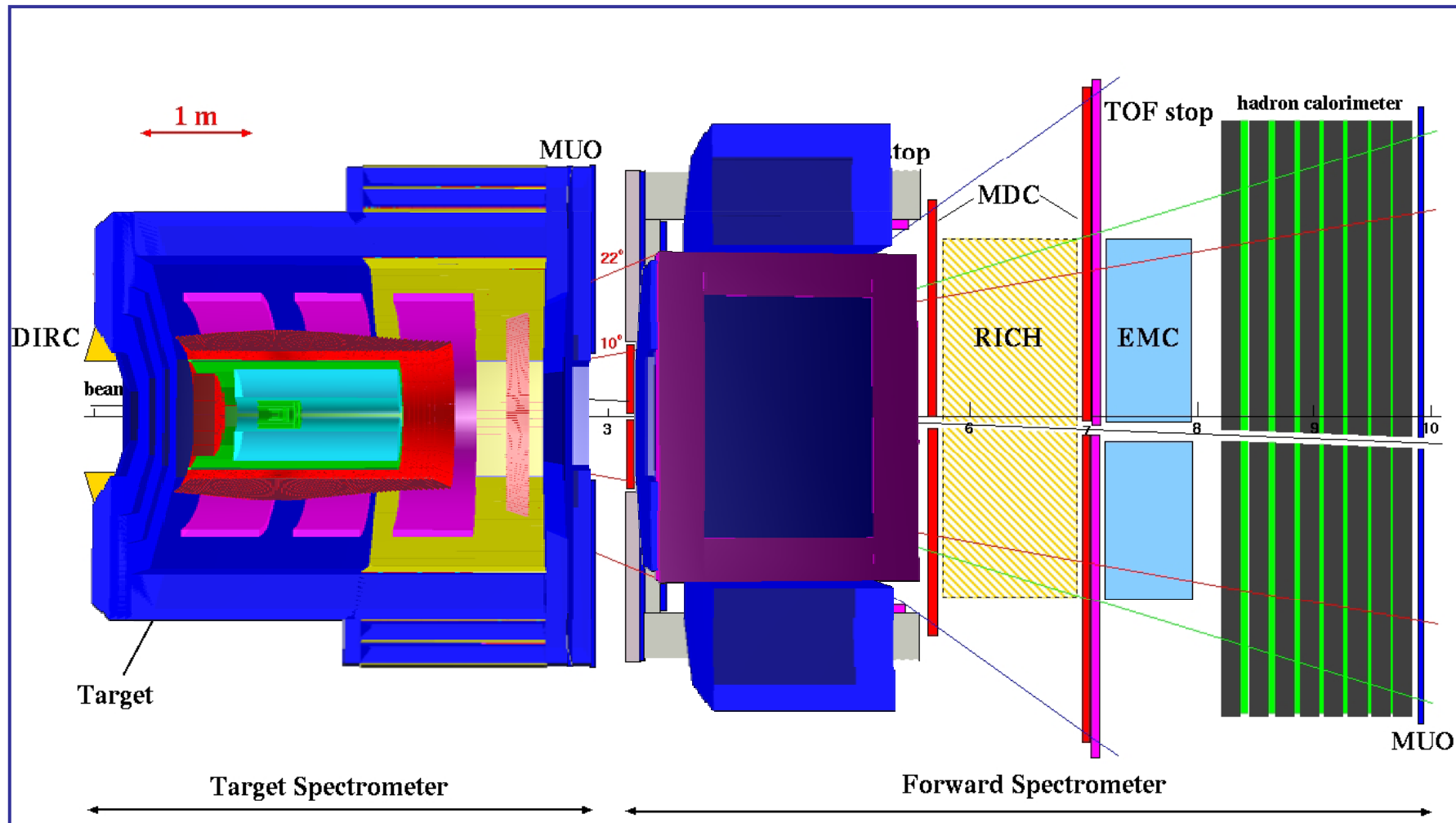


Physics program

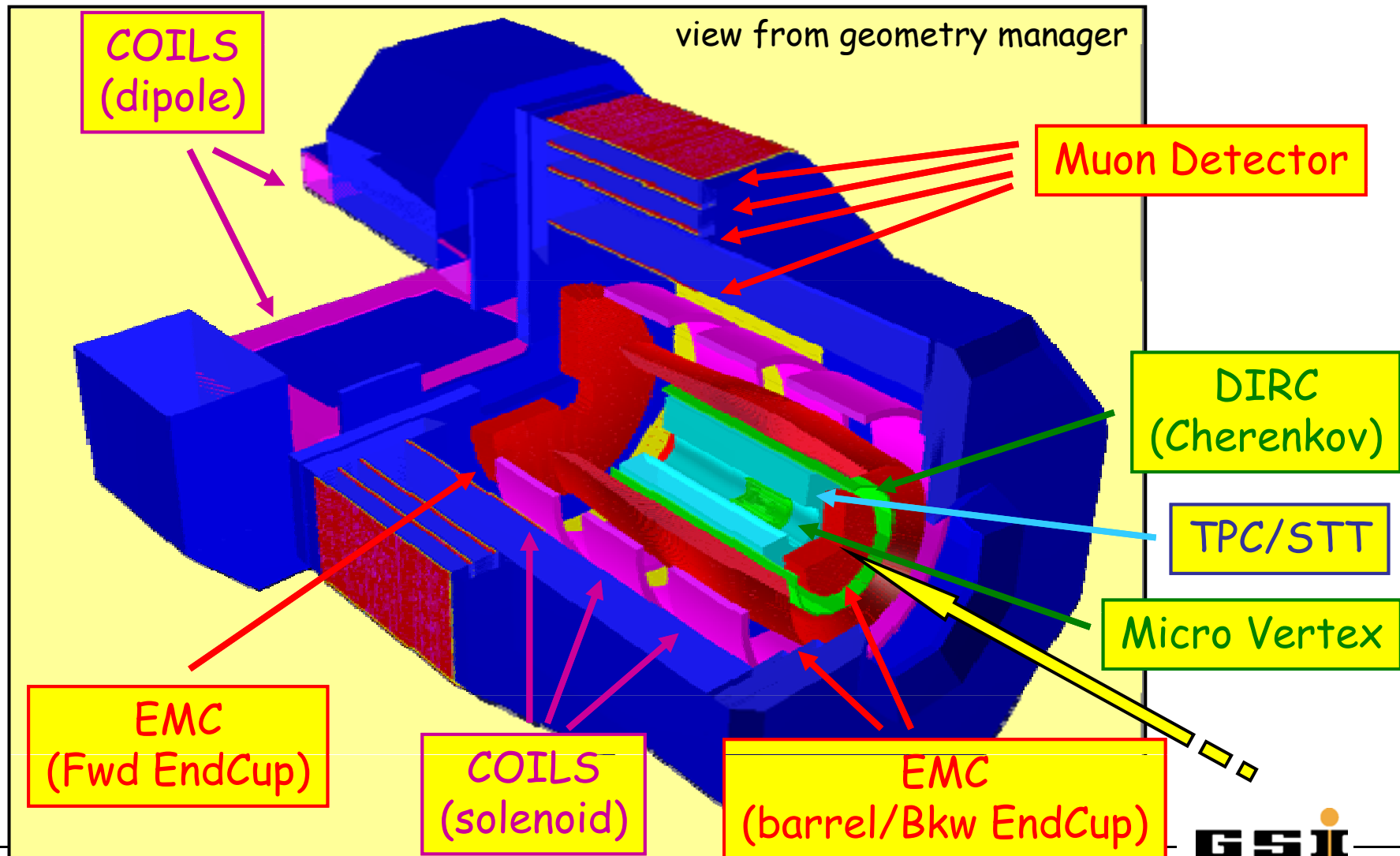


- Charmonium ( $\bar{c}c$ ) spectroscopy
- Open charm spectroscopy
- Search for gluonic excitations (hybrids - glueballs)
- Charmed hadrons in nuclei
- Single and double Hypernuclei
- Other options (EFF, GPD, ...)

# PANDA Detector implementation: proposed geometry



# Detector implementation: state of art

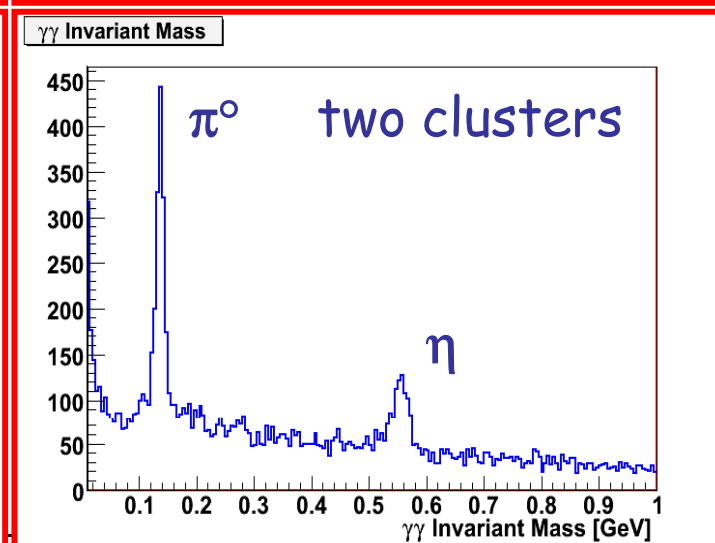
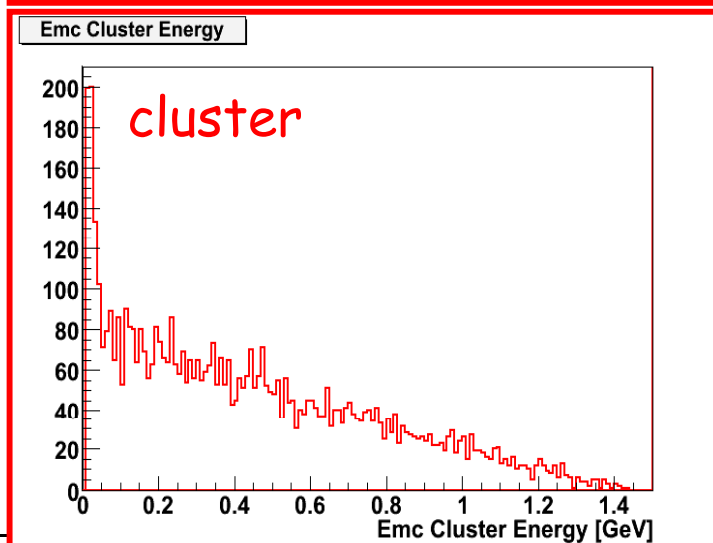
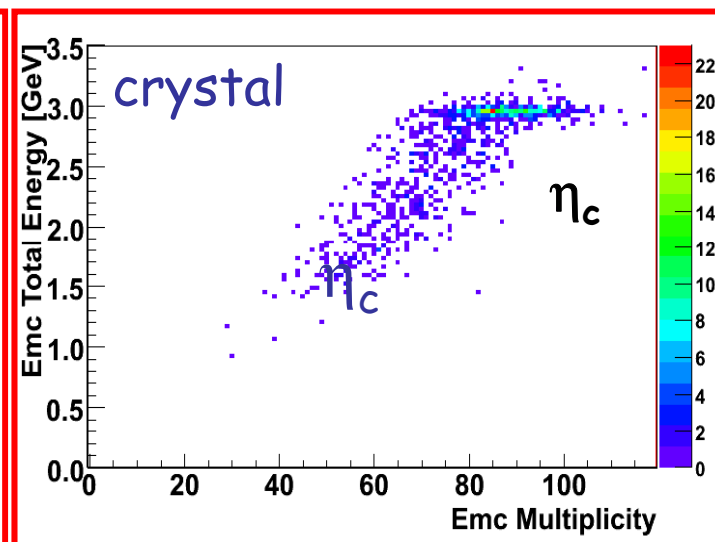
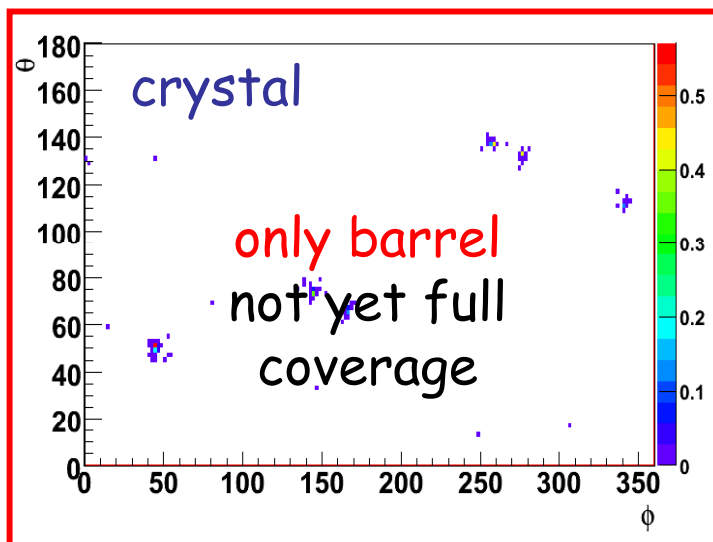
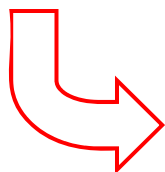


# Reconstruction example: $\eta_c$ in EMC barrel

$\eta_c$  at rest  
in lab frame

$$\eta_c \rightarrow \pi^0 \pi^0 \eta$$

$$\pi^0, \eta \rightarrow \gamma\gamma$$



Clusterization  
Jan Zhong  
Dima Melnichuk

# Summary

- A VMC based framework for FAIR has been implemented
  - First released in March 2004
  - Used for FAIR experiments design studies
- Ongoing work
  - Check of Geant4 VMC interface
  - Check of TFluka VMC interface
  - Use more intensively the TGeoManager services for the reconstruction

# Availability

- Tested on
  - Red Hat 9.0 (gcc 3.2.2 and icc 8.1)
  - Suse 9.0 (gcc 3.3.1)
  - Suse 10.1 (gcc 4.1.0)
  - Debian (gcc 3.2.3)
  - Fedora Core 2 (gcc 3.3.3)
  - Fedora Core 4 (gcc 4.0 )
  - Fedora Core 5 (gcc 4.1.0)
  - Gentoo (gcc 4.1.0)
  - SL 3-4
  - 64 bit architectures
- Binaries are also available for these platforms