



CBM Simulation&Analysis Framework

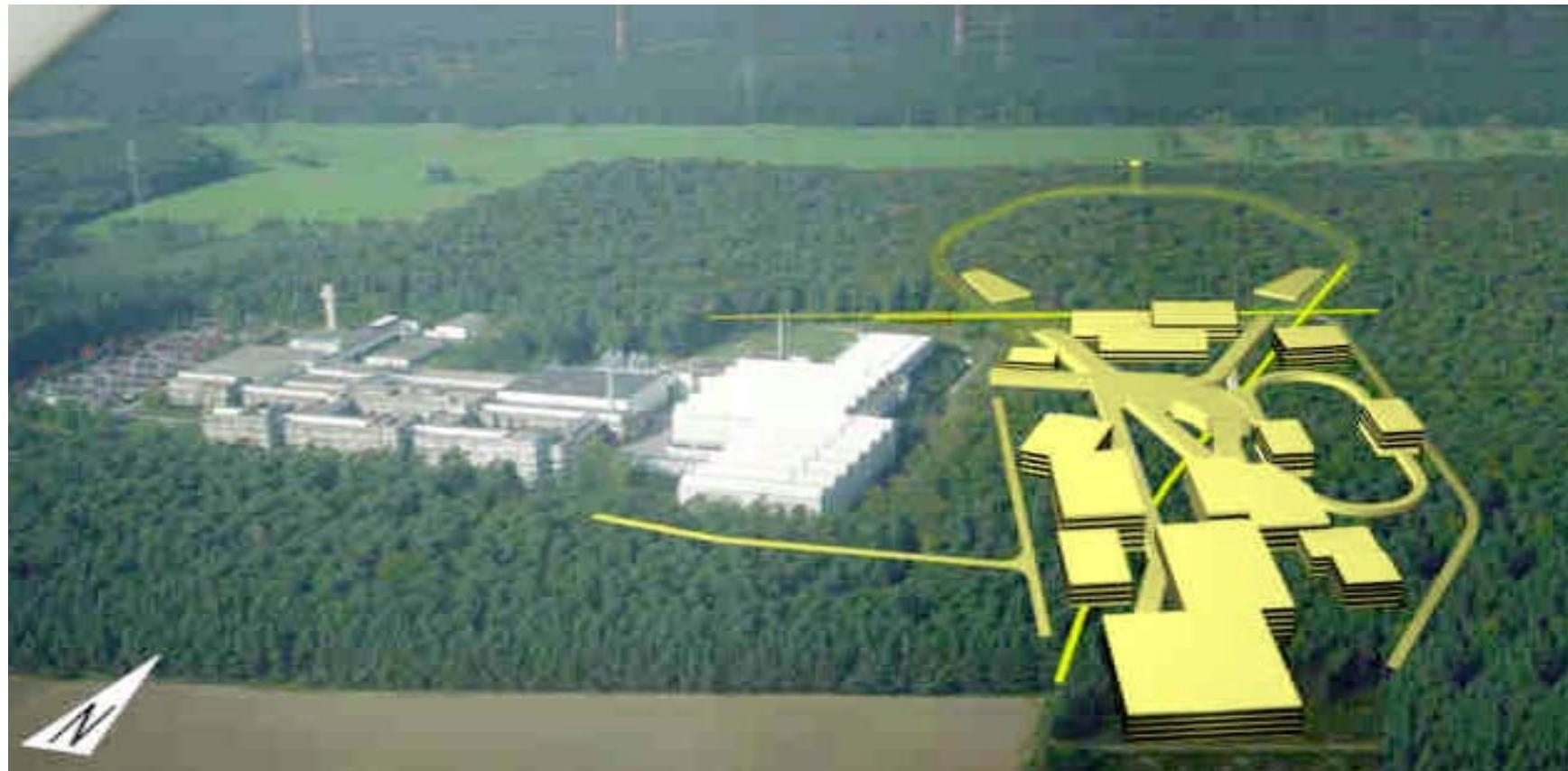
Cbmroot

Mohammad Al-Turany

Denis Bertini

Ilse König

The FAIR project

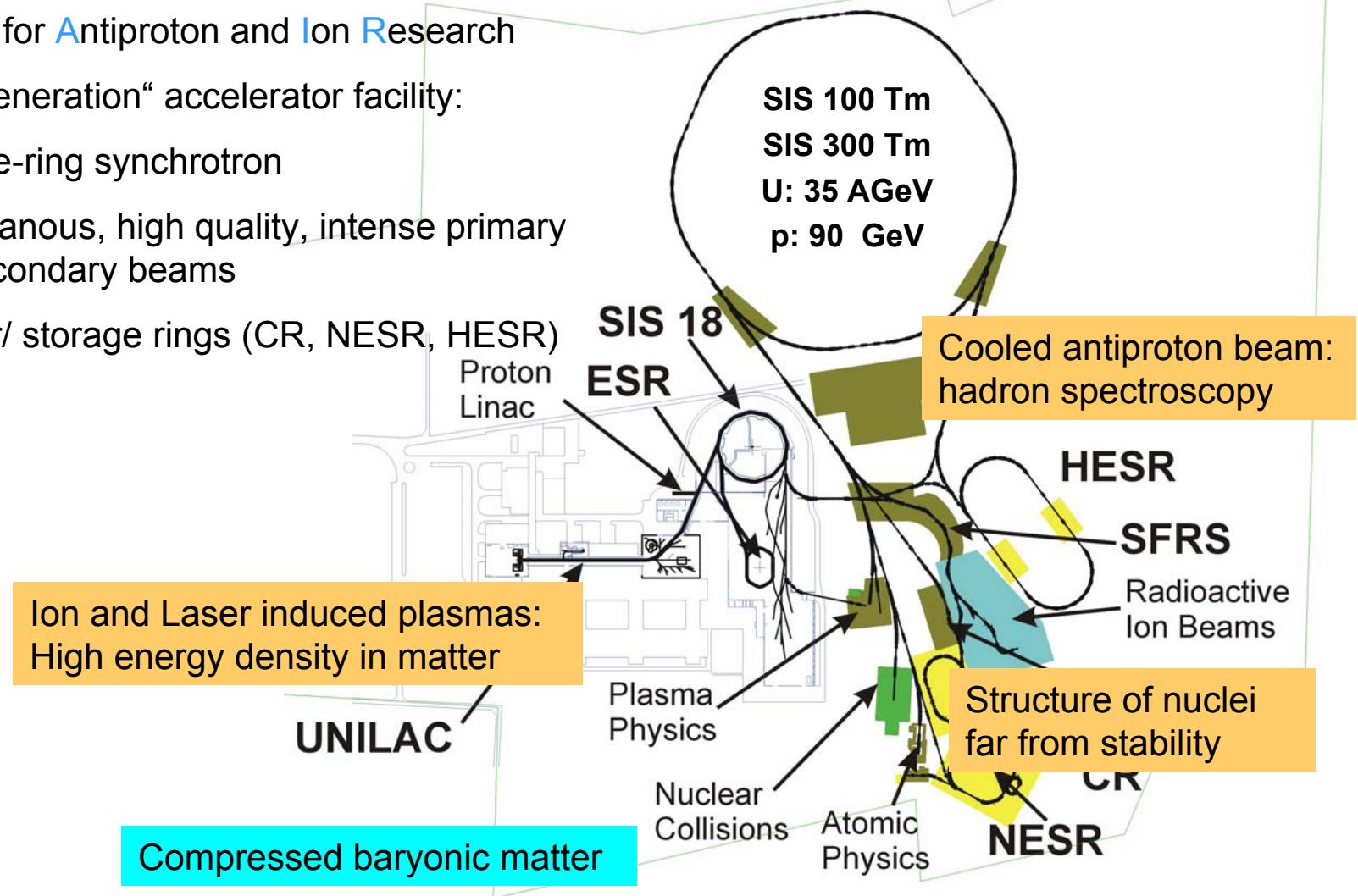


CBM@FAIR(GSI)

Facility for Antiproton and Ion Research

„next generation“ accelerator facility:

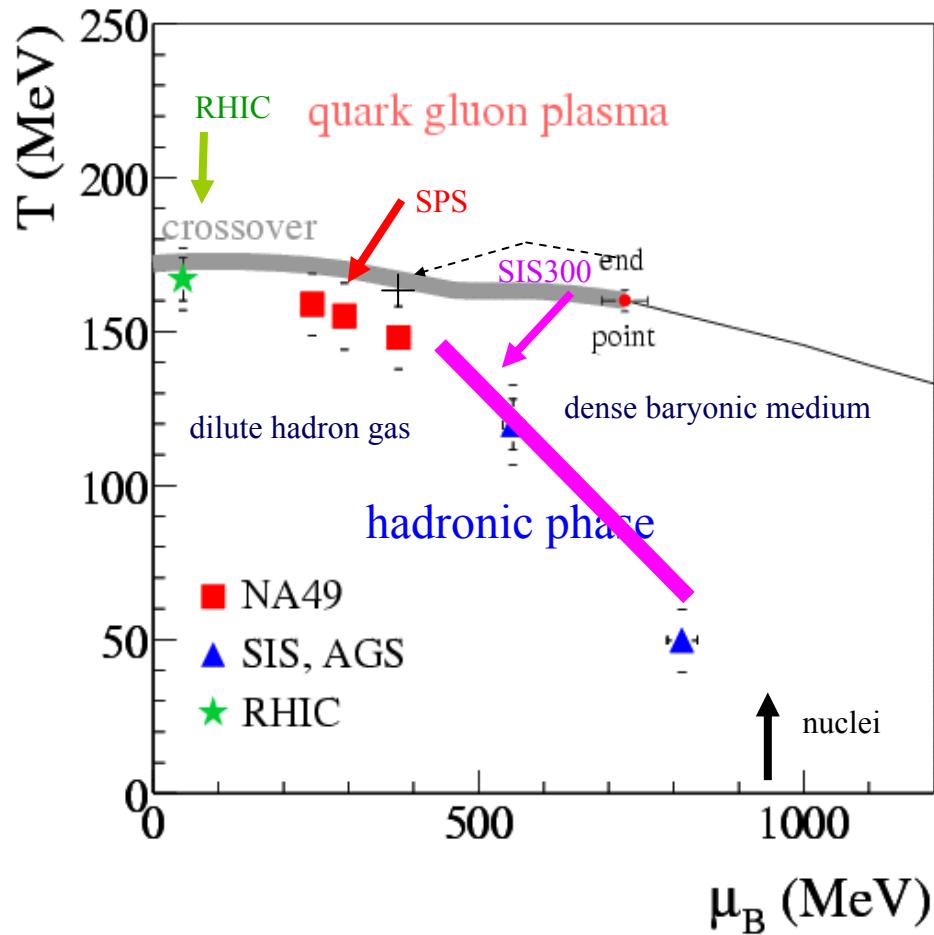
- double-ring synchrotron
- simultaneous, high quality, intense primary and secondary beams
- cooler/ storage rings (CR, NESR, HESR)



Phase diagram of strongly interacting matter

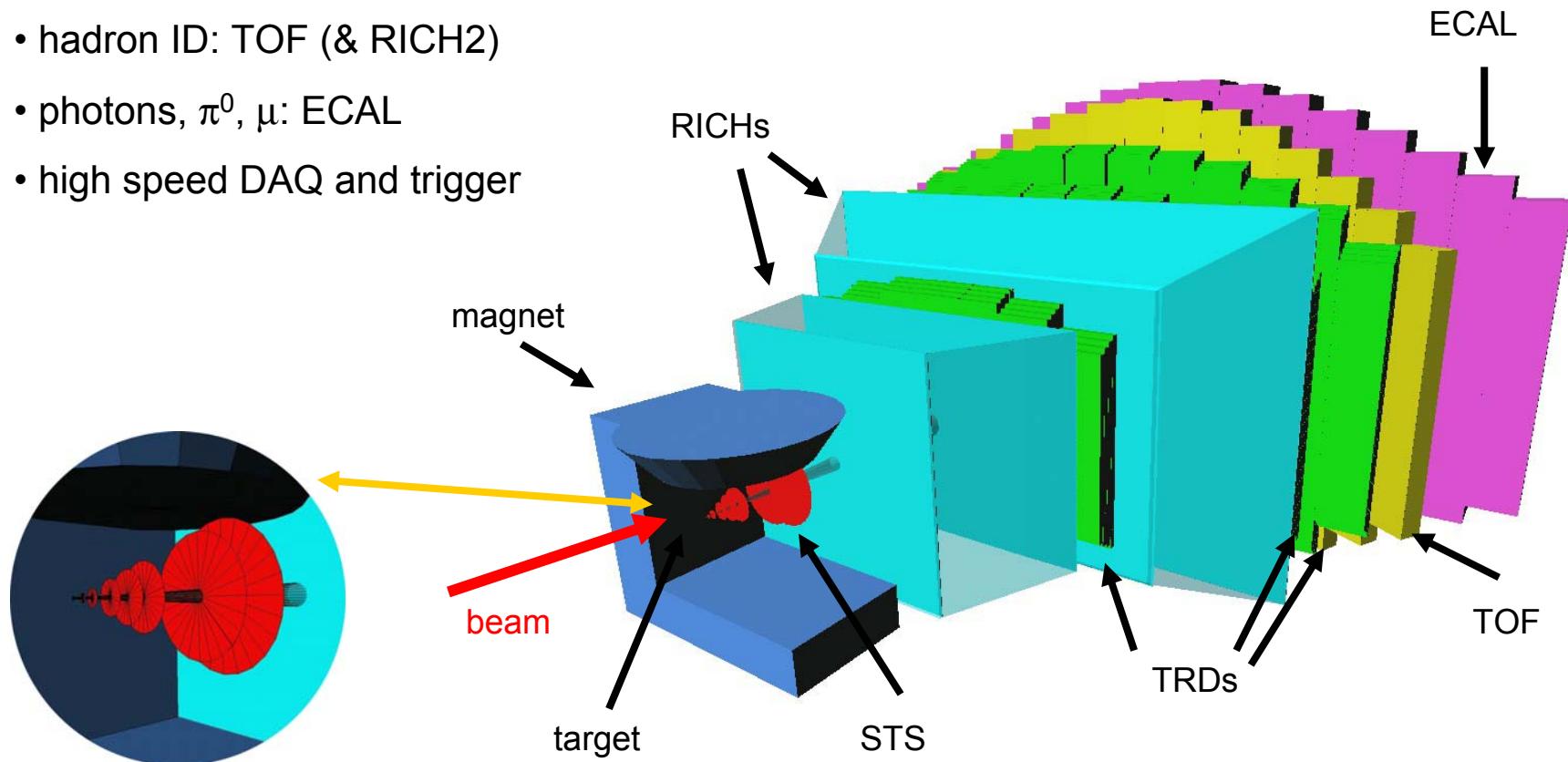
lattice QCD : Fodor / Katz, Nucl. Phys. A 715 (2003) 319

- high T, low μ_B
→ top SPS, RHIC, LHC
- low T, high μ_B
→ SIS
- Highest baryon densities
- Deconfinement ?
 - Strangeness: K, Λ , Σ , Ξ , Ω
 - Charm: J/ ψ , D
 - Flow excitation function
- In medium properties of hadrons
 - $\rho, \omega, \Phi \rightarrow e^+e^-$
- Critical point?
 - Evt/Evt fluctuations



CBM experiment

- 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, π^0 , μ : ECAL
- high speed DAQ and trigger

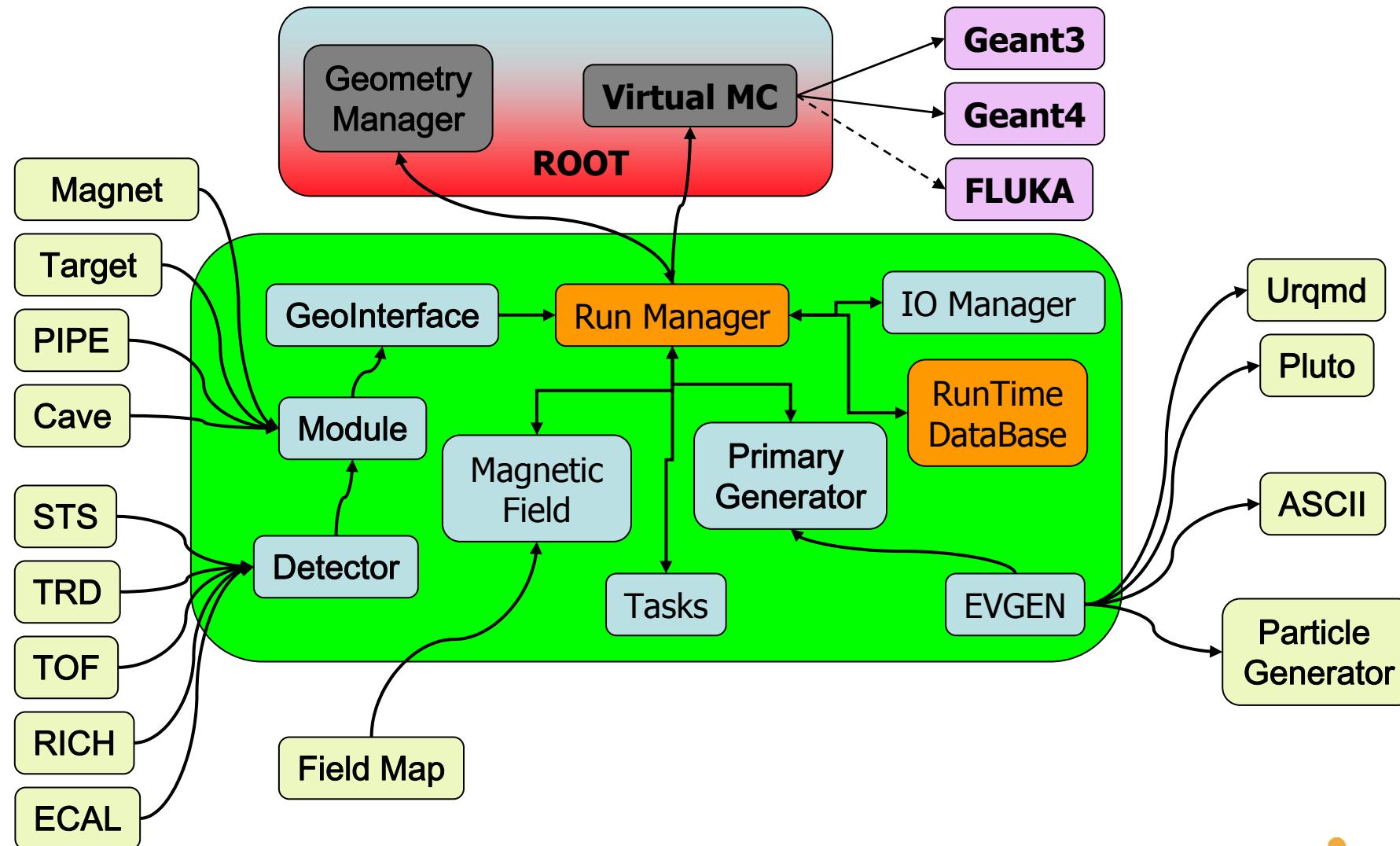




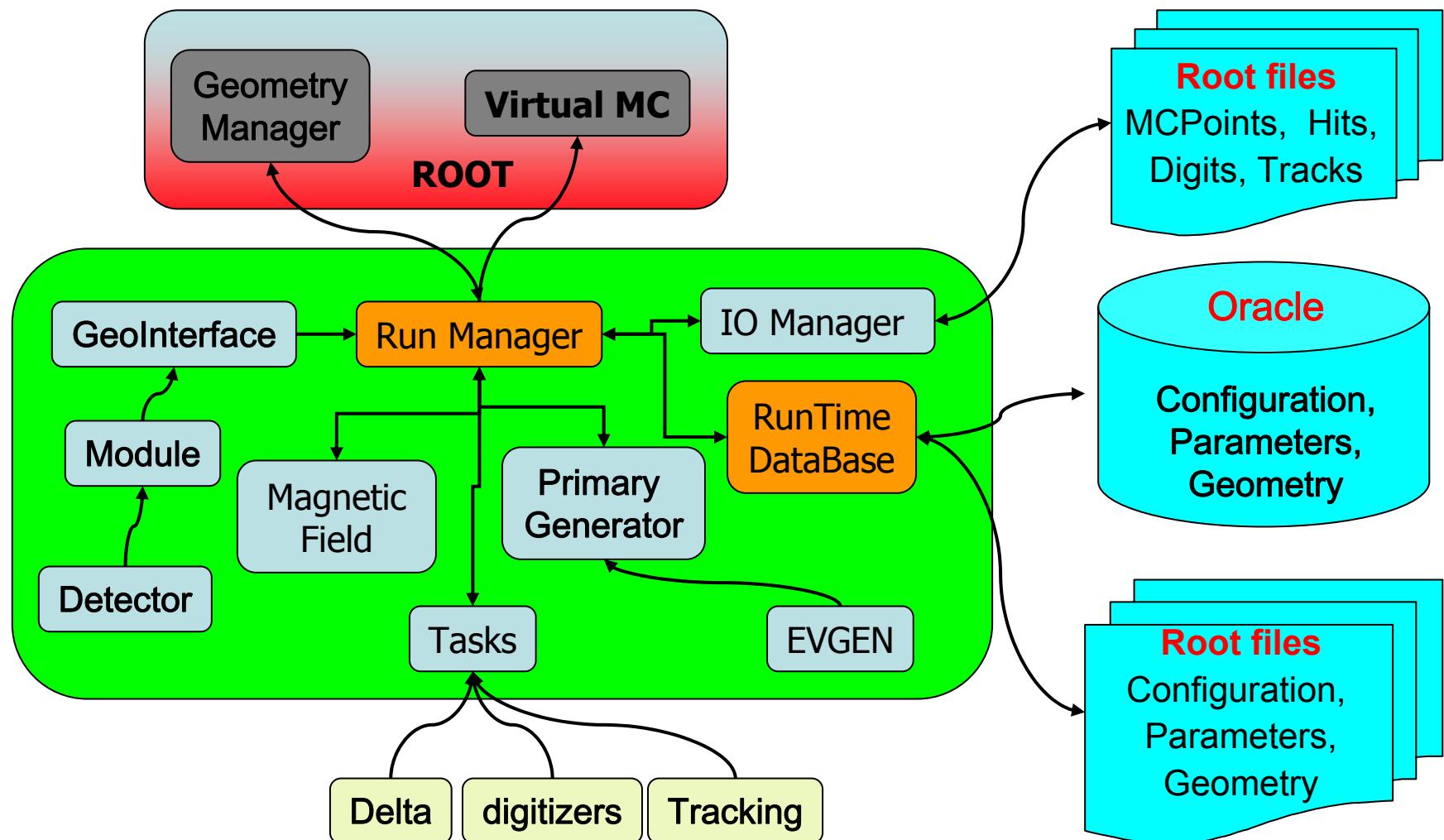
CBM Framework: Features

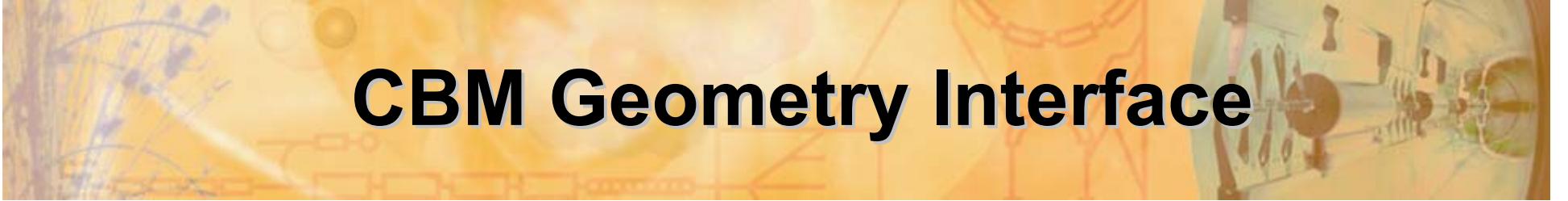
- **The same framework** can be used for Simulation and Analysis
- **Fully ROOT based:**
 - **VMC** for simulation
 - **Geometry Modeller** (TGeoManager) for geometry definition
 - **IO scheme** (TChain, friend TTrees, TFolders) for persistency
 - **TTask** to organize analysis data flow
- **Completely configurable** via ROOT macros
- **Easy to maintain** (only ROOT standard services are used)
- **Reuse of HADES Geometry Interface.**
- **Reuse of HADES Parameter containers.**
- **Event merging** before and/or after transport

CBM Simulation



CBM Analysis

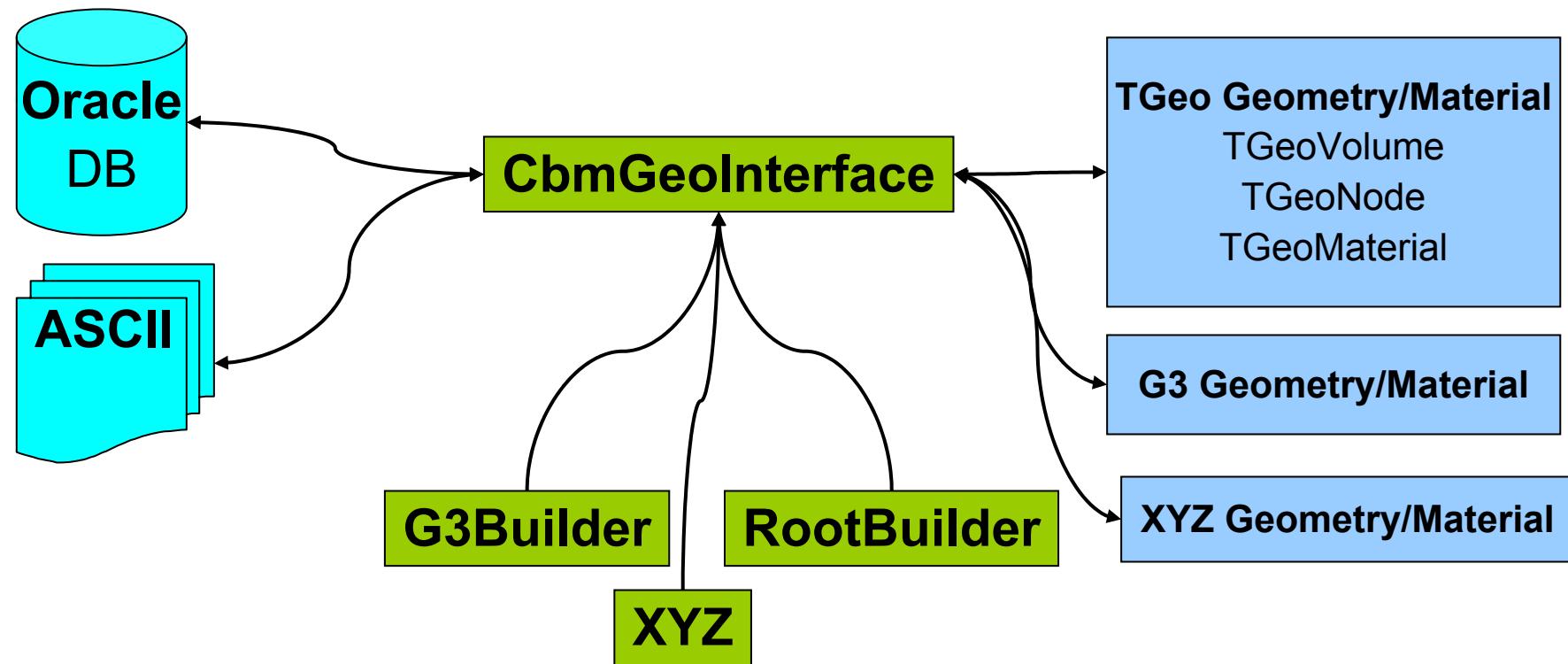




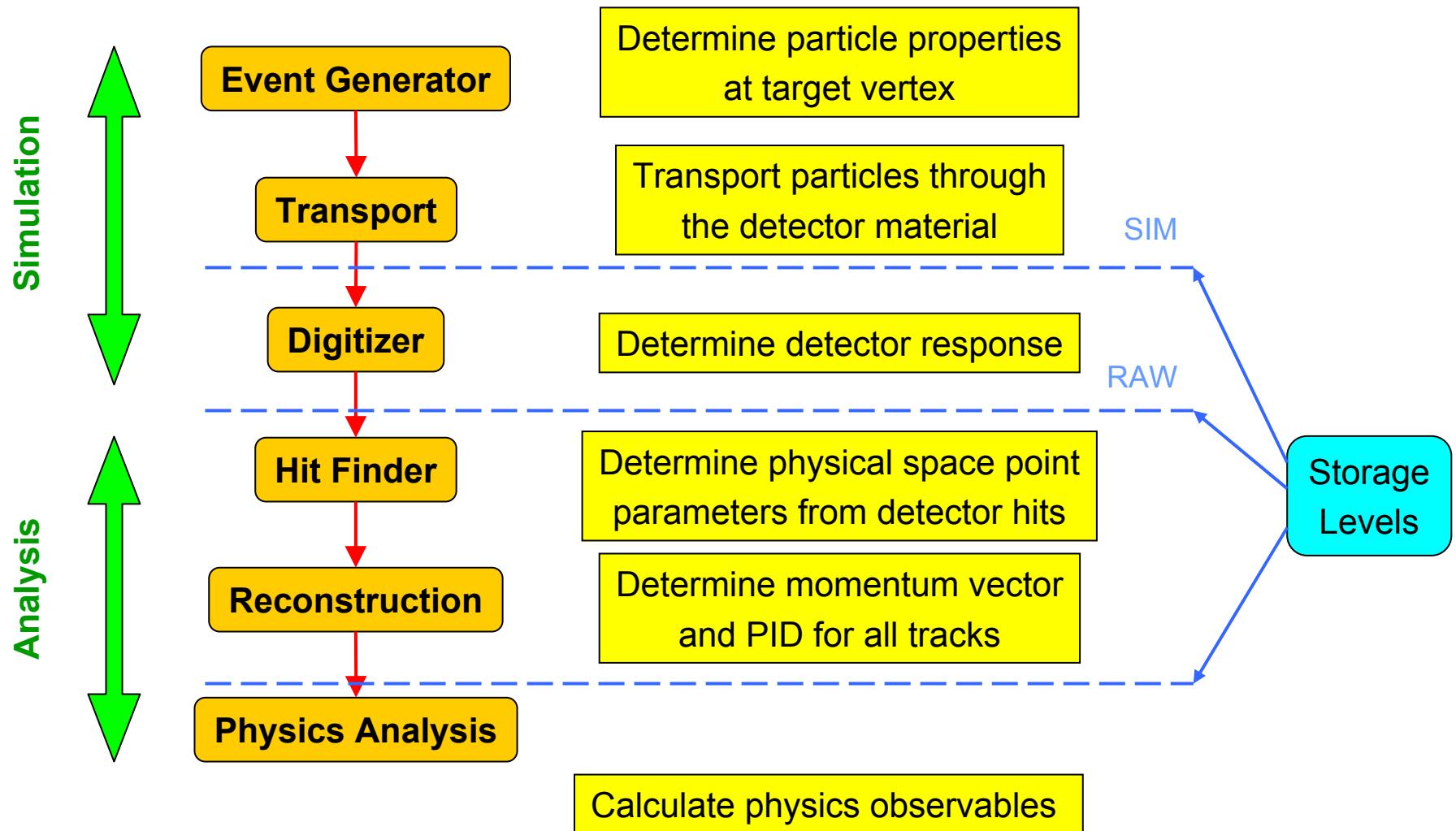
CBM Geometry Interface

- Use the copy mechanism
 - reduce the size of ASCII files
 - Reduce the number of Volumes in Geant
 - Improve Geant tracking performance
- Oracle interface
 - Hades geometry table design reusable
- Advantage:
 - more flexibility : different inputs can be used.
 - closer to technical drawings and analysis coordinate systems

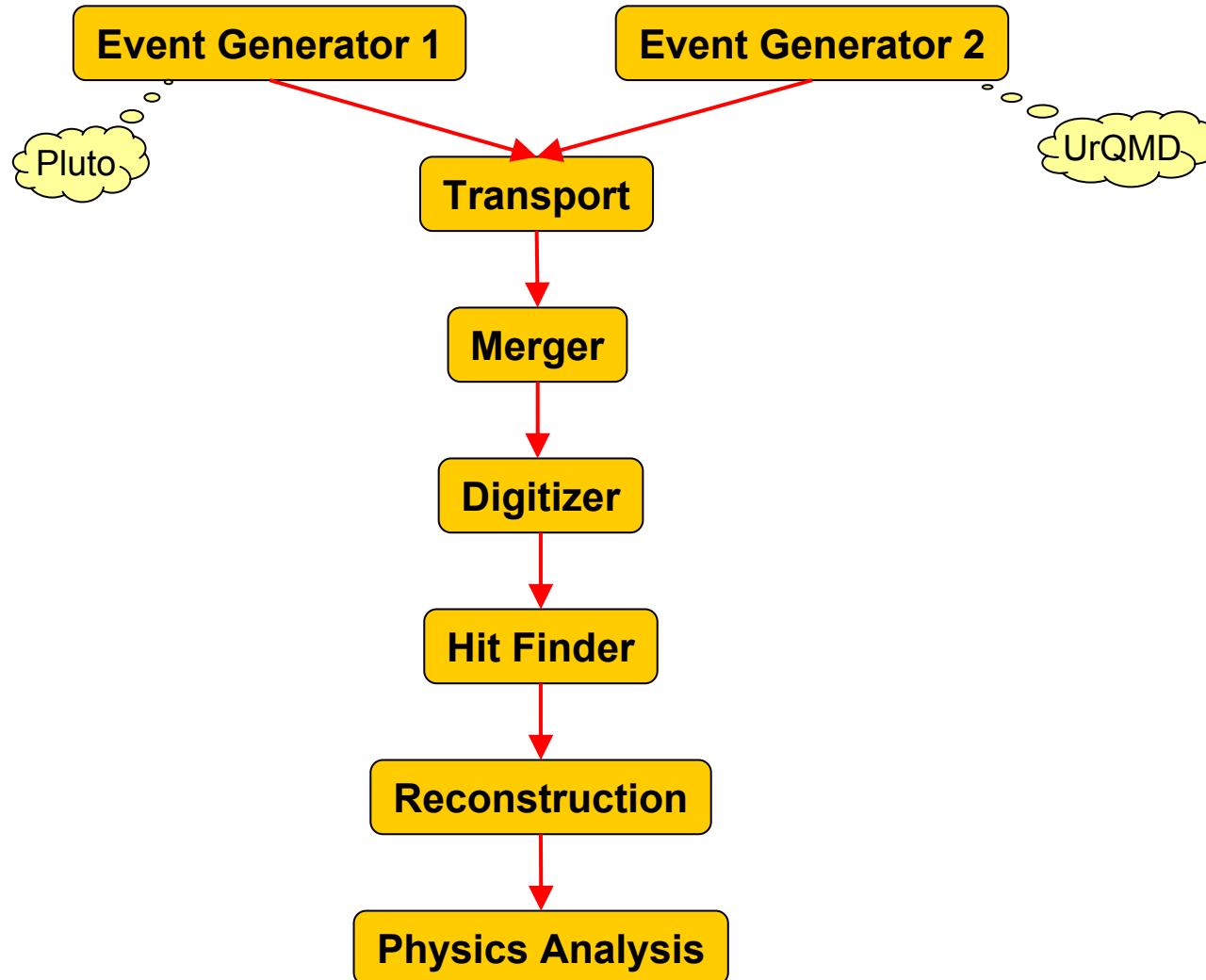
Material & Geometry Interface



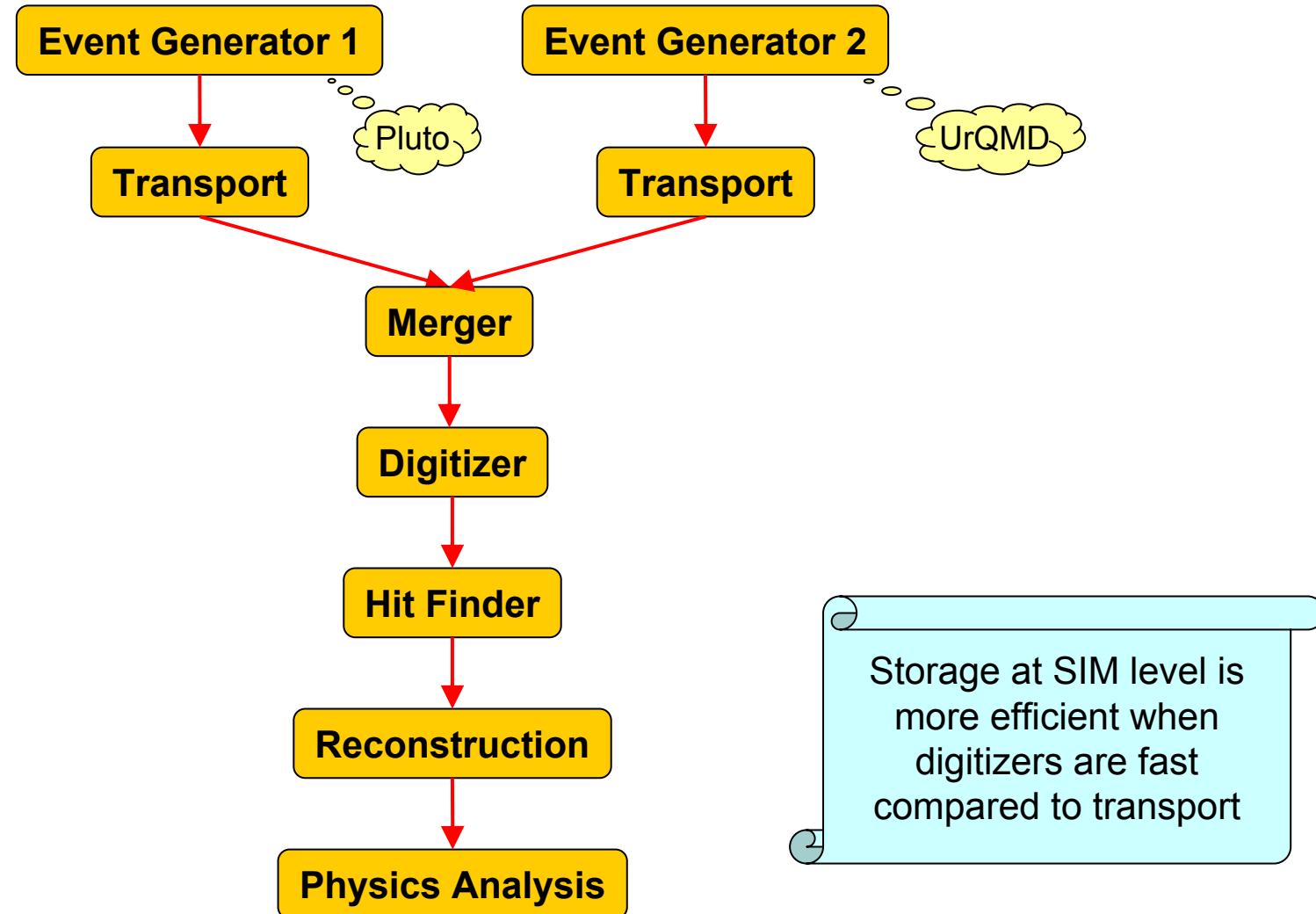
Full Simulation-Analysis Chain



Event merging before transport

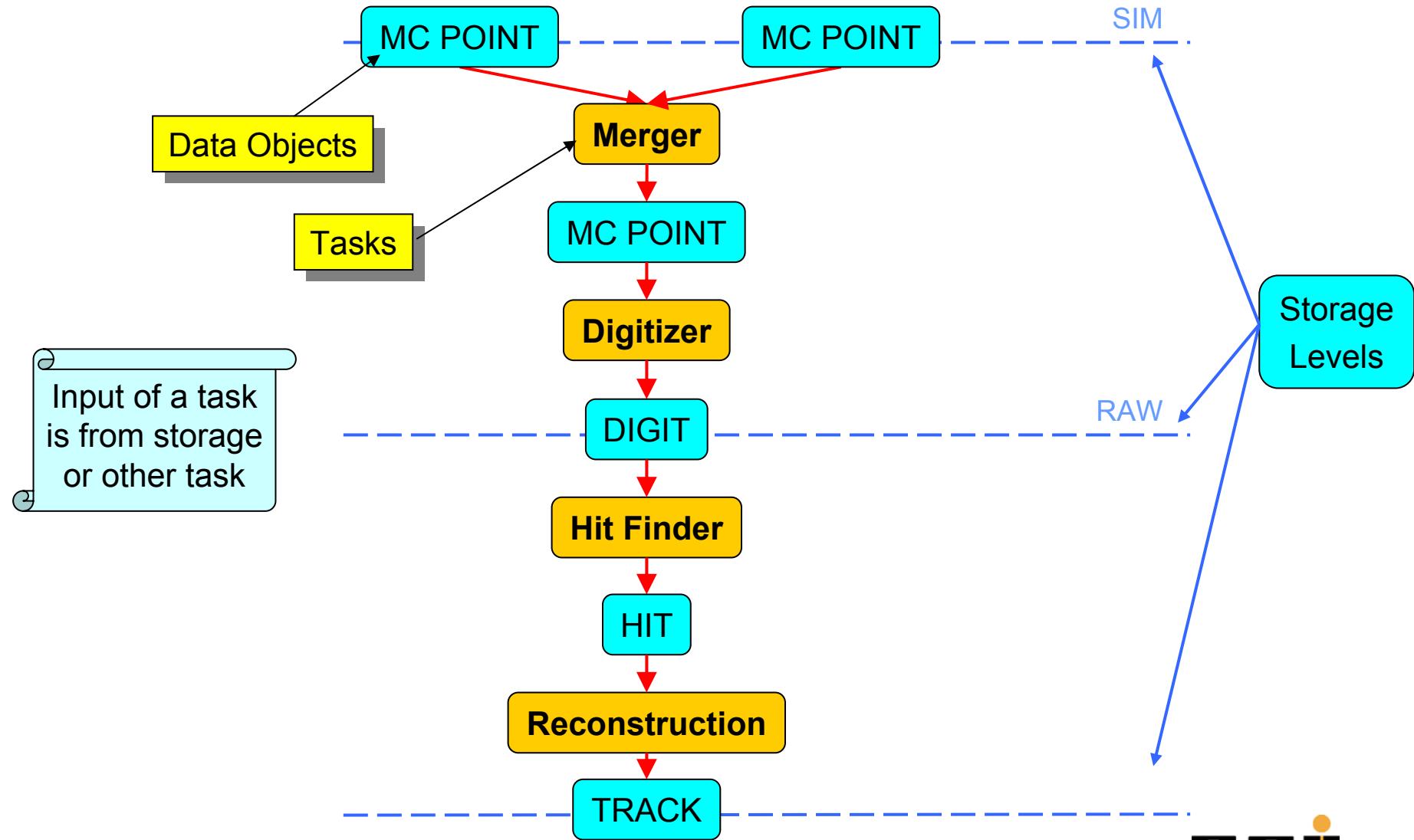


Event merging after transport



Storage at SIM level is
more efficient when
digitizers are fast
compared to transport

Tasks and interfaces



Simulation Macro – loading Libs

```
// Load basic libraries
```

```
gROOT->LoadMacro("$VMCWORKDIR/gconfig/basiclibs.C");  
basiclibs();
```

```
// Load Cbmroot libraries
```

```
gSystem->Load("libCbm");  
gSystem->Load("libPassive");  
gSystem->Load("libGen");  
gSystem->Load("libSts");  
gSystem->Load("libTrd");  
gSystem->Load("libTof");  
gSystem->Load("libRich");
```

Simulation Macro

```
//create the Run Class  
CBMRun *fRun = new CBMRun();  
  
// set the MC version used  
fRun->SetName("TGeant4"); //for G3 use "TGeant3"  
  
// chose an output file name  
fRun->SetOutputFile("test.root");
```

Simulation Macro- Create Modules

```
CbmModule *Cave= new CbmCave("WORLD");
Cave->SetGeometryFileName("PASSIVE/CAVE", "v03a");
fRun->AddModule(Cave);
CbmModule *Target= new CbmTarget("Target");
Target->SetGeometryFileName("PASSIVE/TARGET", "v03a");
fRun->AddModule(Target);

CbmModule *Pipe= new CbmPIPE("PIPE");
Pipe->SetGeometryFileName("PASSIVE/PIPE", "v03a");
fRun->AddModule(Pipe);

CbmModule *Magnet= new CbmMagnet("MAGNET");
Magnet->SetGeometryFileName("PASSIVE/MAGNET", "v03a");
fRun->AddModule(Magnet);
```

Simulation Macro- Create Detectors

```
CbmDetector *STS= new CbmSts("STS", kTRUE);  
STS->SetGeometryFileName("STS/STS", "v03c");  
fRun->AddModule(STS);
```

```
CbmDetector *TOF= new CbmTof("TOF", kTRUE );  
TOF->SetGeometryFileName("TOF/TOF", "v03_v10");  
fRun->AddModule(TOF);
```

```
CbmDetector *TRD= new CbmTRD("TRD",kFALSE );  
TRD->SetGeometryFileName("TRD/TRD", "v04b_9" );  
fRun->AddModule(TRD);
```

Simulation Macro-Event Generators

```
CbmPrimaryGenerator *priGen= new CbmPrimaryGenerator();  
fRun->SetGenerator(priGen);
```

```
CbmUrqmdGenerator *fGen1= new CbmUrqmdGenerator("00-03fm.100ev.f14");
```

```
CbmPlutoGenerator *fGen2= new CbmPlutoGenerator("jpsi.root");
```

```
CbmParticleGenerator *fGen3= new CbmParticleGenerator();
```

```
fRun->AddGenerator(fGen1);  
fRun->AddGenerator(fGen2);  
fRun->AddGenerator(fGen3);
```

Simulation Macro-Magnetic Field

```
// setting a field map
CbmFieldMap *fMagField= new CbmFieldMap("FIELD.v03b.map");

// setting a constant field
CbmConstField *fMagField=new CbmConstField();
fMagField->SetFieldXYZ(0, 30 ,0 );      // values are in kG

// MinX=-75, MinY=-40,MinZ=-12 ,MaxX=75, MaxY=40 ,MaxZ=124 );

fMagField->SetFieldRegions(-74, -39 ,-22 , 74, 39 , 160 ); // values are in cm
```

```
fRun->SetField(fMagField);
```

Simulation Macro- Run Simulation

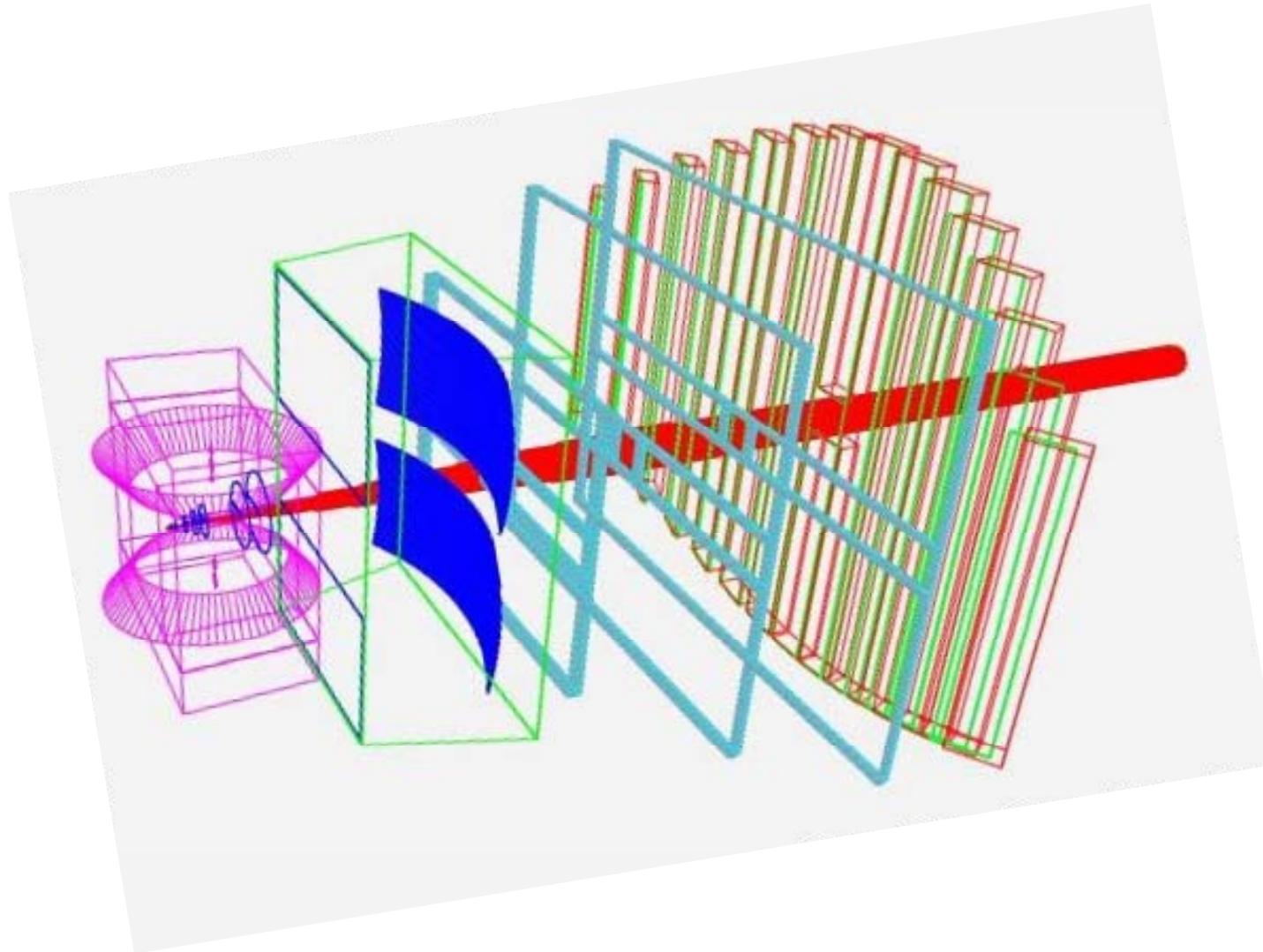
```
fRun->Init(); // Initialize the simulation
```

Simulation:

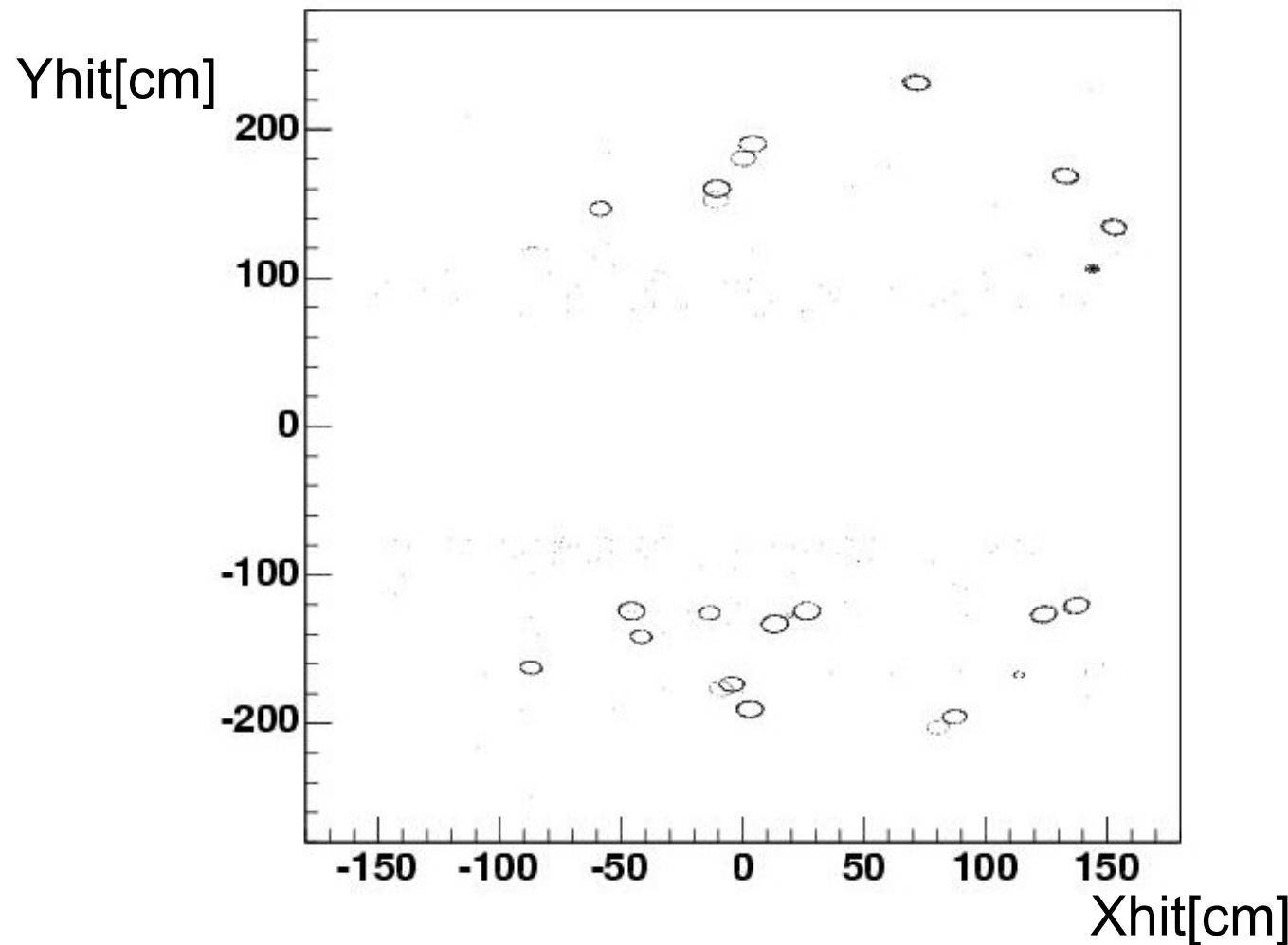
1. Initialize the VMC (Simulation)
2. Initialize Tasks (if they are used in Simulation)

```
fRun->Run(NoOfEvent); //Run the Simulation
```

CBM Detector Geometry



Example: Rich Detector



CbmTask

- Tasks can be organized into a hierarchy and displayed in the browser.
- The **CbmTask** class is the base class from which the tasks are derived.
- To give task functionality, you need to subclass the **CbmTask** class and override:
 - `Init();` //Initialization
 - `Exec(Option_t * option);`

Analysis Task- Init example

```
void CbmITrack::Init()
{
    // Get a pointer to the ROOT Manager ( data store )
    CbmRootManager *fManager =CbmRootManager::Instance();
    // Get the relevant data for the Task
    // activate in IO the corresponding TTree branch
    fListSTSpts=(TClonesArray *)fManager->ActivateBranch("STSPoint");

    //Create a new branch in the output file for the results
    fHitCollection = new TClonesArray("CBMSTSDoubleHit");
    fManager->Register("STSDoubleHit","STS", fHitCollection);
}
```

Analysis Macro

```
{
```

```
    gROOT->LoadMacro("$VMCWORKDIR/gconfig/basiclibs.C");
```

```
    basiclibs();
```

```
    gSystem->Load("libCbm");
```

```
    gSystem->Load("libITrack");
```

```
    CBMRun *fRun= new CBMRun();
```

```
    fRun->SetInputFile("/d/STS_AuAu25Gev_Urqmd.root");
```

```
    fRun->SetOutputFile("trackOutput.root");
```

```
    CBMITrack *tr= new CBMITrack("Tracking Algorithm");
```

```
    fRun->AddTask(tr);
```

```
    fRun->Init();
```

```
    fRun->Run();
```

```
}
```

Chaining root files

- Chaining mechanism supported in the Analysis:

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

```
    CbmRun *fRun = new CbmRun();
    fRun->SetInputFile("myfile1.root");
    fRun->AddFile("myfile2.root");
    fRun->AddFile("myfile3.root");
```

```
    // loop over all entries
    fRun->Run();
```

```
}
```

Track Visualization

- In the Simulation macro add:

.....

```
fRun->SetStoreTraj(kTRUE);  
fRun->Init();
```

```
// Set cuts for storing the trajectories
```

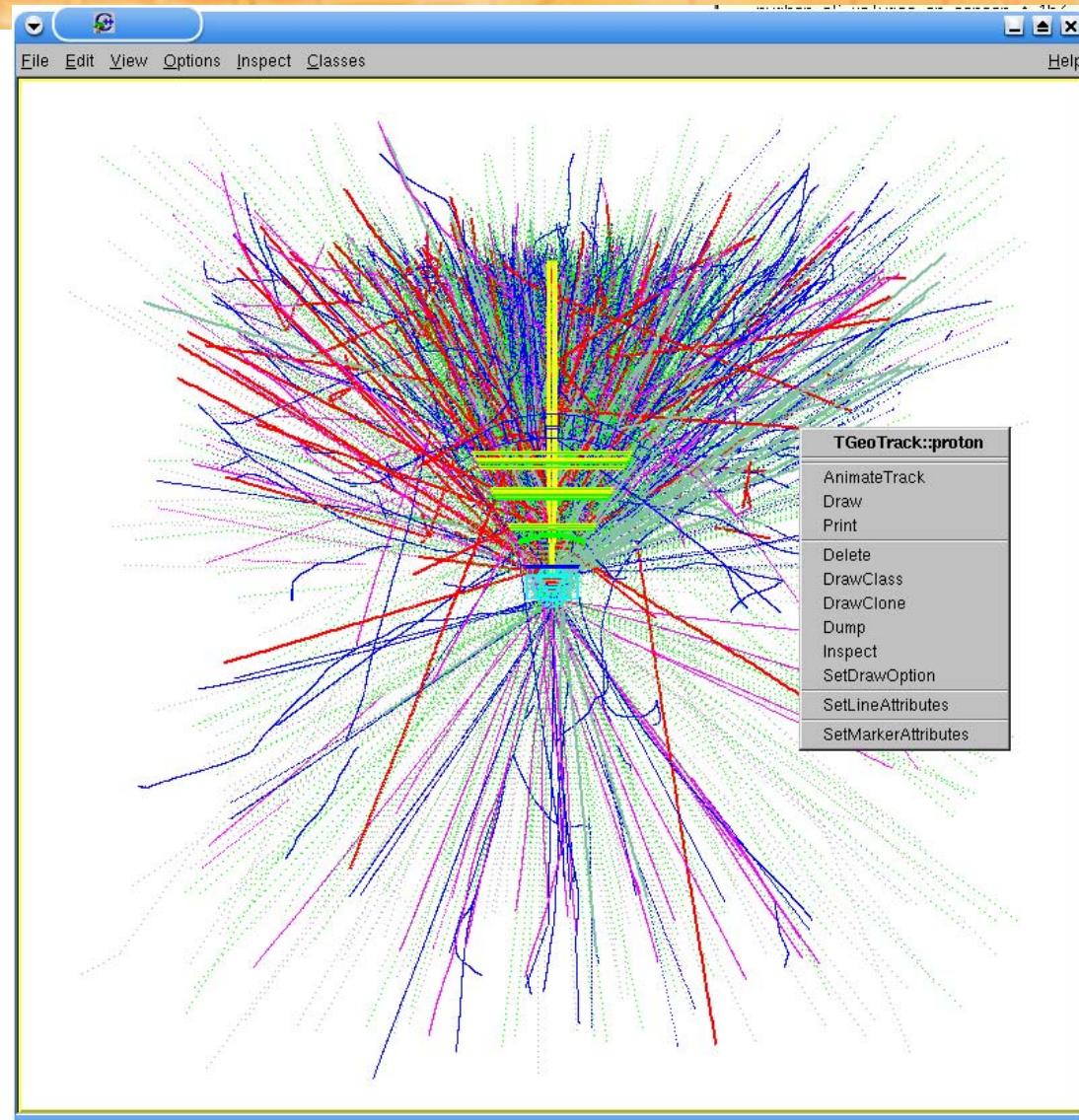
```
CbmTrajFilter* trajFilter = CbmTrajFilter::Instance();  
trajFilter->SetStepSizeCut(1); // 1 cm  
trajFilter->SetEnergyCut(0., 1.04); // 0 < Etot < 1.04 GeV  
trajFilter->SetStorePrimaries(kFALSE);
```

.....

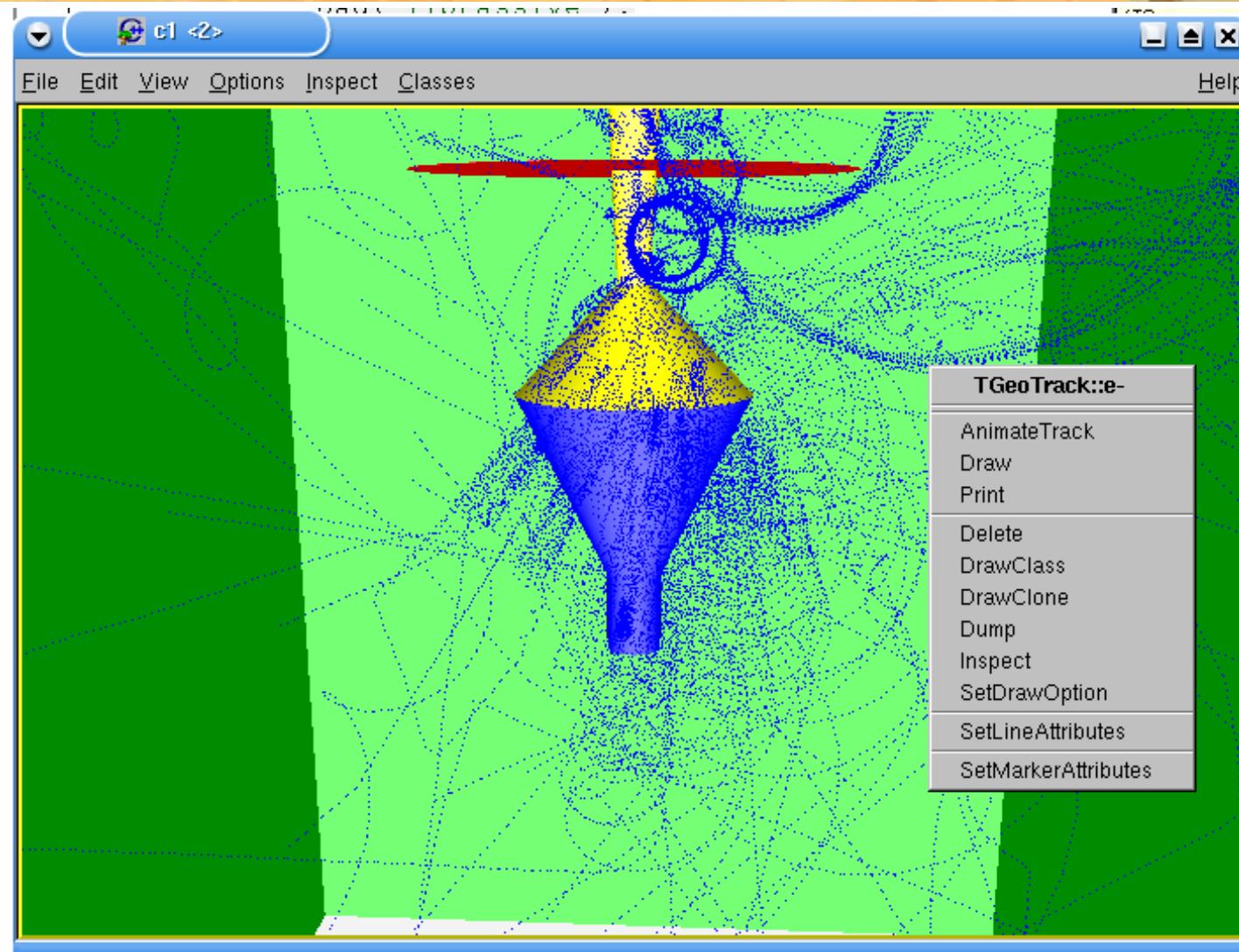
Example: Visualization macro

```
{  
gROOT->LoadMacro("$VMCWORKDIR/gconfig/basiclibs.C");  
basiclibs();  
gSystem->Load("libCbm");  
.....  
  
TFile* file = new TFile("test.root");  
TGeoManager *geoMan = (TGeoManager*) file->Get("CBMGeom");  
TCanvas* c1 = new TCanvas("c1", "", 100, 100, 800, 800);  
c1->SetFillColor(10);  
geoMan->DrawTracks("same/Nneutron");  
geoMan->SetVisLevel(3);  
geoMan->GetMasterVolume()->Draw("same");  
}
```

Track Visualization

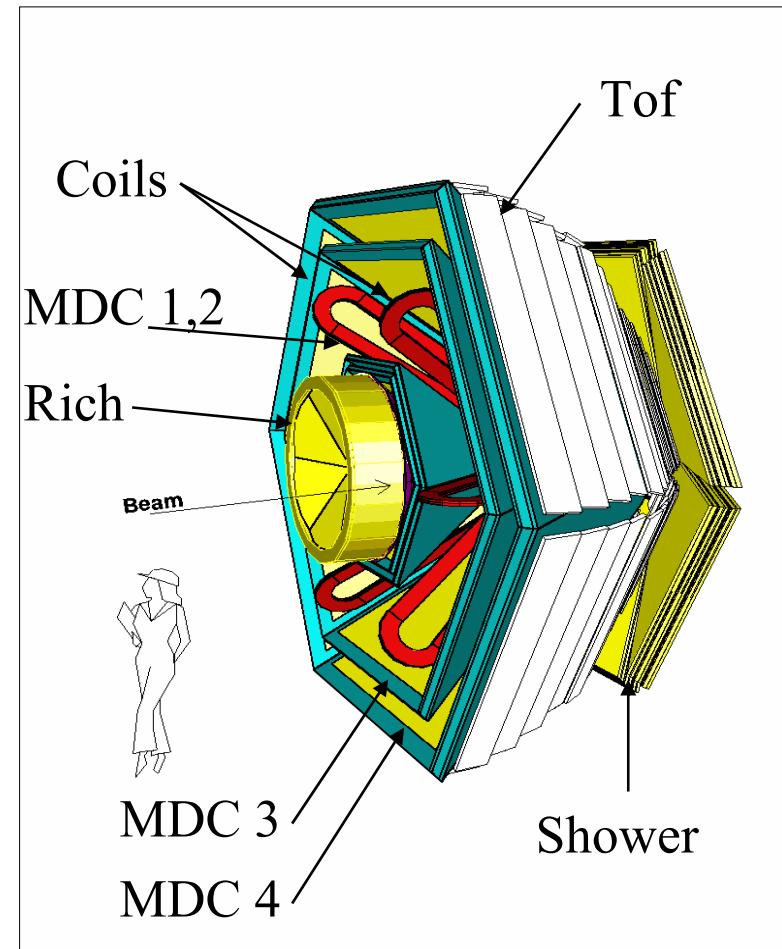


Track Visualization



Hades@GSI

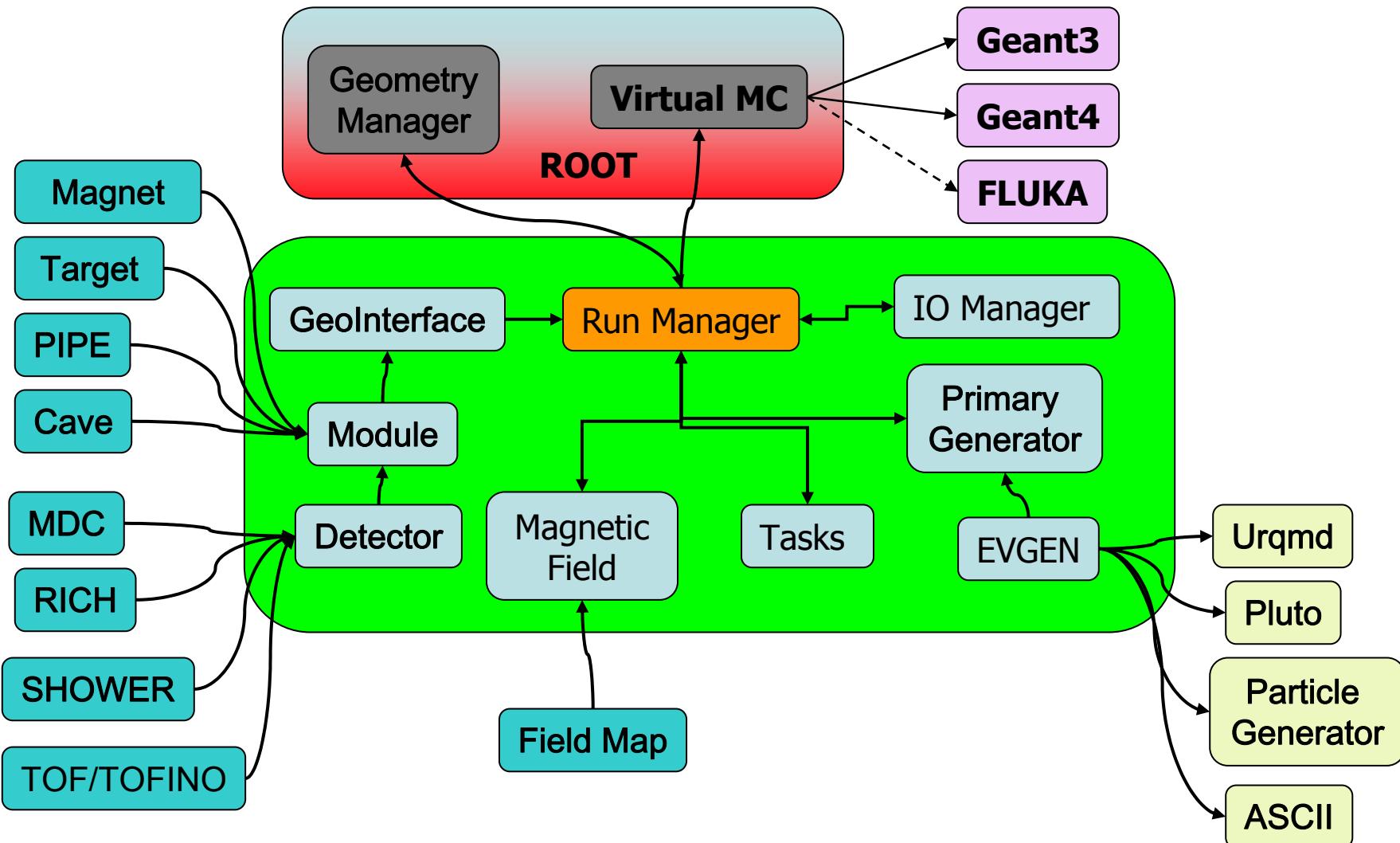
- **Goal:** Study of in-medium modifications (ρ , ω , ϕ) properties
 - Produced in A+A, p+A, π +A collisions
 - Di-electrons are used as probes: $V \rightarrow e^+e^-$
- Hexagonal symmetry around the beam axis
- Geometrical acceptance of 40%
- Invariant mass resolution of 1%
- Operates at reaction rates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



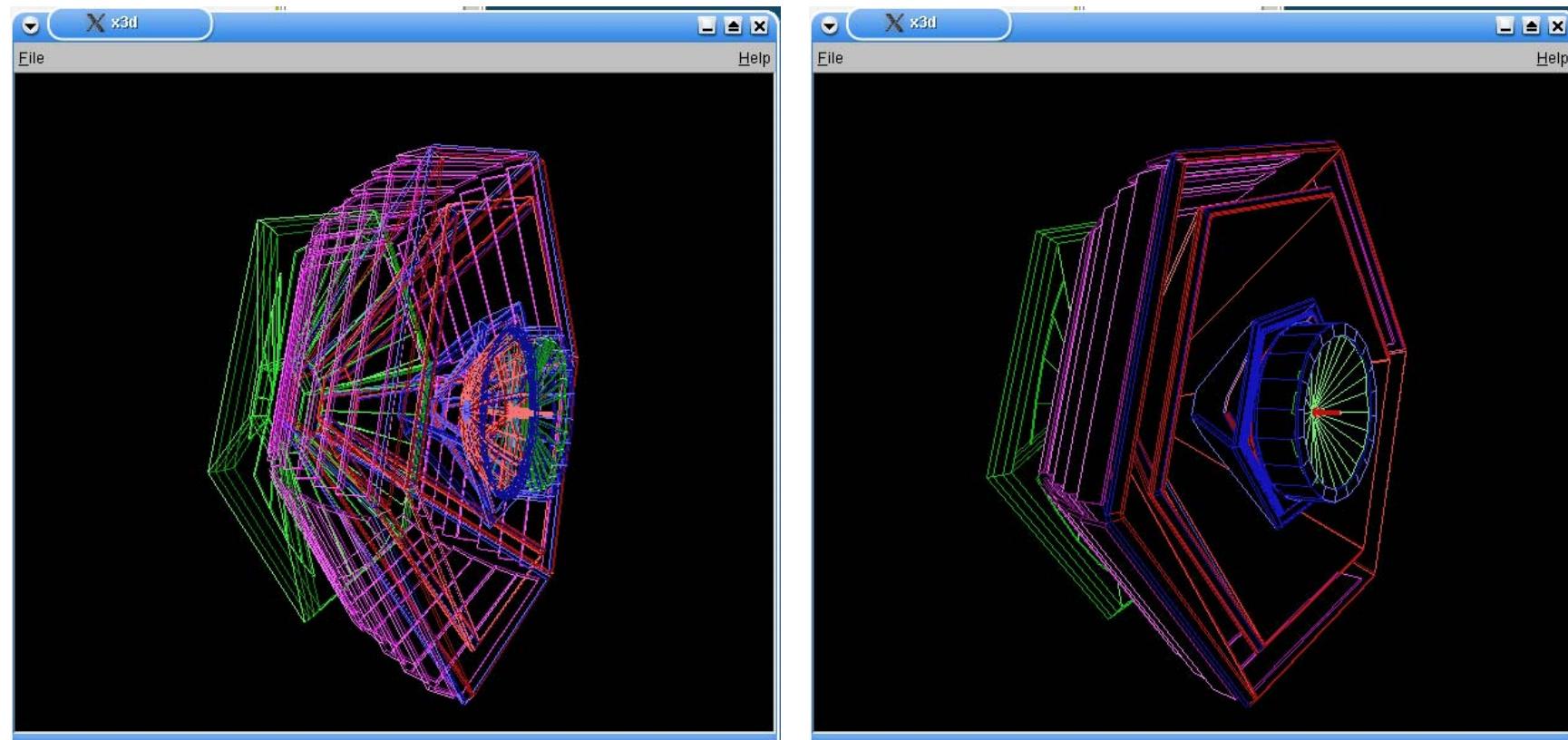
Hades Simulation using Cbmroot, why?

- **Need to simulate heavy system at High energy**
 - Need external stack for Geant3: internal stack capacity reached
 - Check data with geant4
- **Easy to use CBM framework services**
- **The Only efforts:**
 - Definition of Detector MC point container
 - Field map reader
 - Conversion from Lab. MC point definition to points defined in the local ref. Frame of the sensitive volume
 - Modification of some particles physical characteristics:
 - Use of TVirtualMC::Gspart()

Hades Simulation



Hades Simulation





Summary

- A VMC based framework for CBM has been implemented
 - First released in March 2004
 - Work on digitizers and full tracking is going on.
- Oct 04 release of Cbmroot was used to produce data for the CBM technical report
 - Packages (ROOT 4.01/02 , GEANT3/GEANT4.6.2)
- Tested on
 - Red Hat 9.0 (gcc 3.2.2 and icc 8.1)
 - Suse 9.0 (gcc 3.3.1)
 - Debian (gcc 3.2.3)
 - Fedora Core 2 (gcc 3.3.3)
- Binaries are also available for these platforms

Summary

- Hades spectrometer has been fully integrated
 - Gives us the opportunity to tune Geant4 (cuts/physics list ...) and compare with real Data !
 - Realistic test of the framework.
- FLUKA simulation will be also possible with no efforts !
- CBM Convention rules will be forced via CodeWizard very soon
- HADES Parameter containers are implemented and ready to use
- The Hades Oracle interface is already available and will be integrated within the next months



Problems!

VMC Configuration:

- Now: g3config.C and g4config.C + g4config.in
- Need to define a unique configuration VMCconfig.C
- Need to add in TVirtualMC a set of virtual functions to define:
 - Processes
 - Physics cuts
- in G4VMC: user defined physics list and cuts need to be supported



Problems!

VMC Geometry definition:

- VGM should be part of the VMC (replace the existing converters)
- VGM should support replicas and copies.
- Handling the optical properties of material should be done in the geometry and not from the TVirtualMC