LHCb-UK students' meeting

Overview of Monte Carlo simulation(s) in LHCb

G.Corti

CERN, 13 October 2009

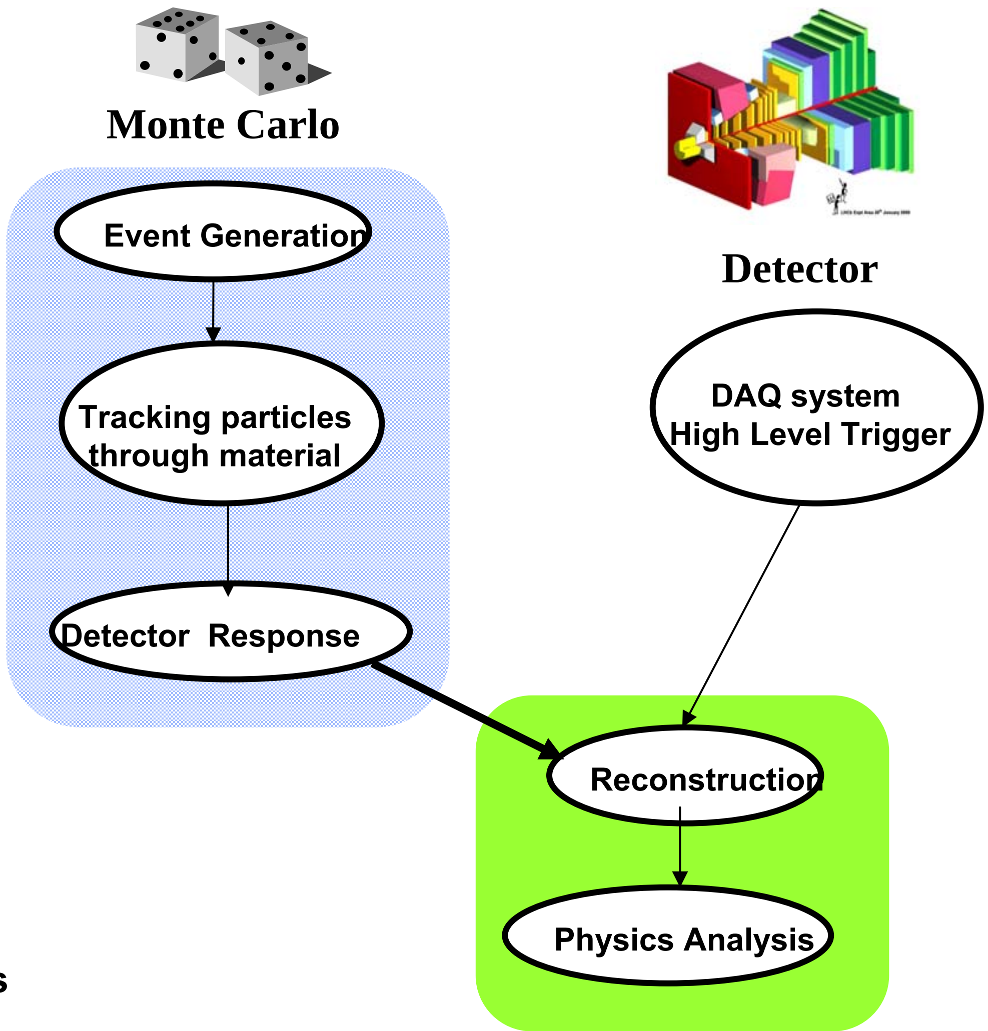
Outline

- **Introduction**
  - **Why and how Monte Carlo simulations are used in High Energy Experiments**
- **Walk through the various aspects covered by MC simulations in the LHCb experiment software**
  - **Generation**
  - **Tracking**
  - **Detector response**
- **How does it all fit together in LHCb**
- **Statistics and CPU**

Introduction

- In High Energy Experiments when elementary particles collides in accelerators (for example)
  - unstable particles are created, these particles decay quickly.
  - it is necessary to reconstruct an "image" of the event through measurements by complex detectors comprising many subdetectors.

- The Monte Carlo simulation role is to mimic what happens in the spectrometers to understand experimental conditions and performance.
  - Monte Carlo data are processed as real data in Reconstruction and Physics Analysis
    - BUT we know the "truth"
  - Comparing the simulation with what is measured in reality we can interpret the results



Why to use Monte Carlo simulations

- **Want to generate events in as much detail as Mother Nature**
  - get average and fluctuations right
  - make random choices, ~ as in nature
  - an event with n particles involves O(10n) random choices
    - multiple variables: flavour, mass, momentum, spin, production vertex, lifetime,...
  - At LHC in 4π: ~ 100 charged and ~ 200 neutral (+ intermediate stages)
  - ➔ several thousand choices

- **This applies also to the transport code through the spectrometer and the detectors response**
  - want to "track" the particles in the geometrical setup and have them interact with the matter
    - energy loss, multiple scattering, magnetic field
  - want to simulate the detection processes and response of a given detector
    - ionization, scintillation, light
  - the interaction events are stochastic and so is the transport process

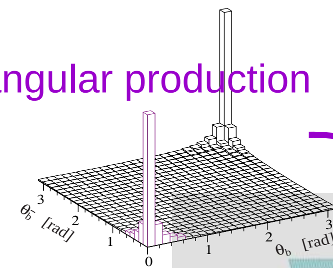
- **A problem well suited for Monte Carlo method simulations**
  - computational algorithms relying on repeated random sampling to compute their results

- **In fact a Monte Carlo simulation in a High Energy Experiment is a collection of different Monte Carlo, each specialized in a given domain, working together to provide the whole picture**

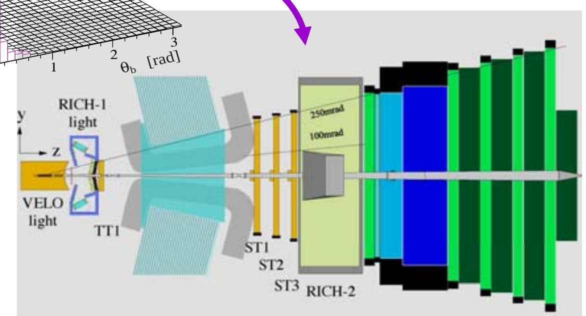
How are MC simulations used ?

- **Detailed simulations are part of HEP physics**
  - **Simulations are present from the beginning of an experiment**
    - **Simple estimates needed for making detector design choices**
    - **Develop reconstruction and analysis programs**
    - **Evaluate physics reach**
  - **We build them up over time**
    - **Adding/removing details as we go along**
  - **We use them in many different ways**
    - **Detector performance studies**
    - **Providing efficiency, purity values for analysis**
    - **Looking for unexpected effects, backgrounds**
    - **When theory is non well known compare to various models and accounts for different detector "acceptance"**

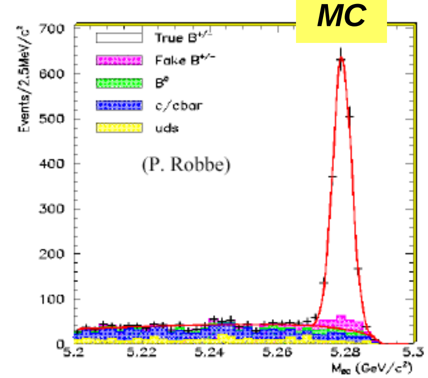
$b\bar{b}$ angular production



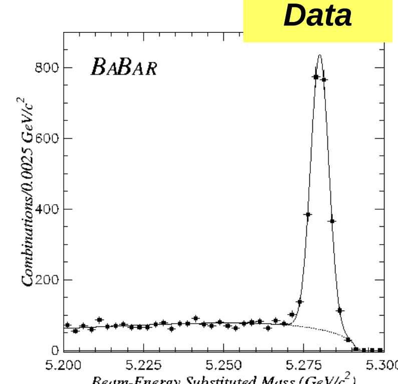
shape of the detector



MC

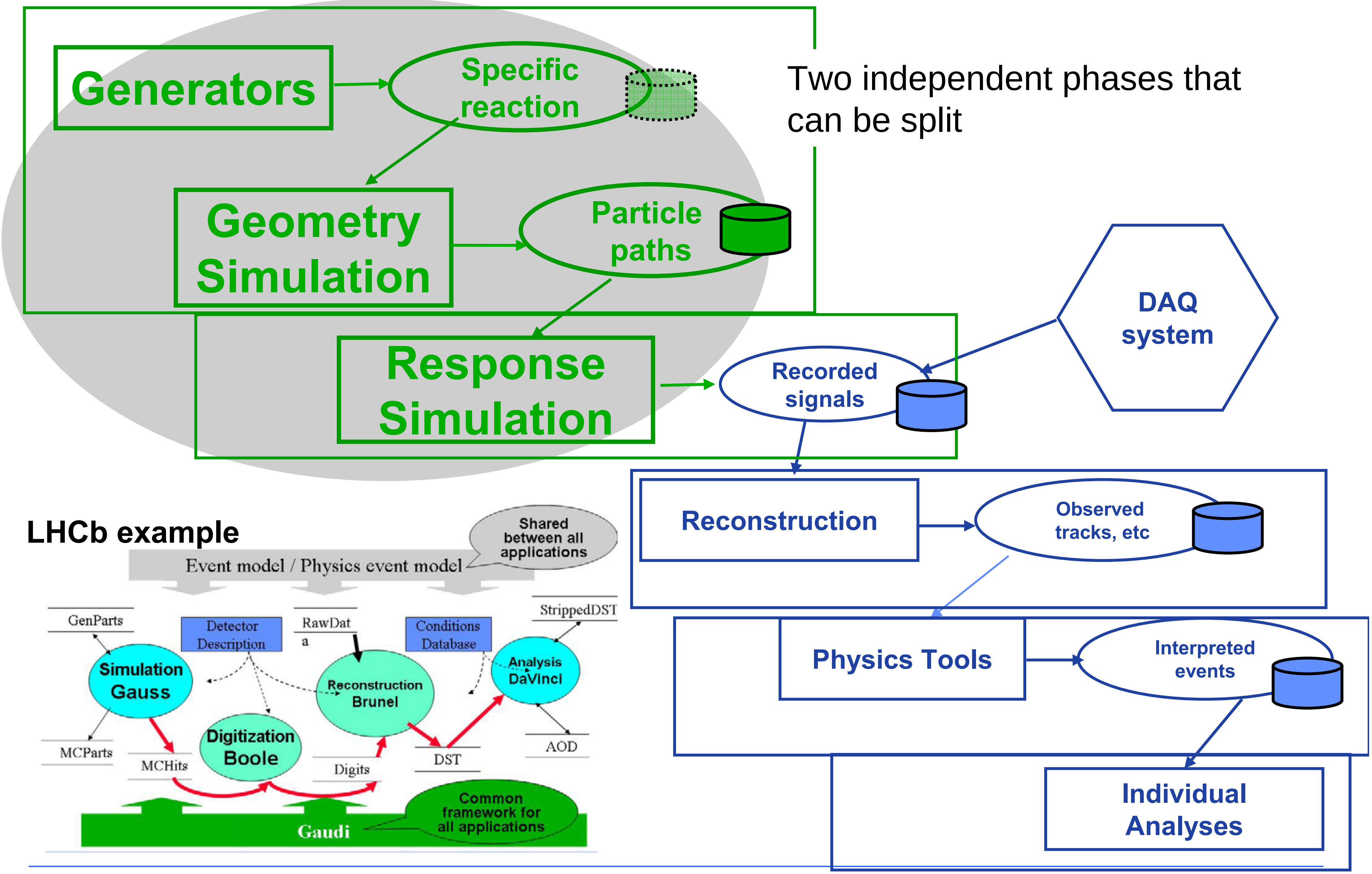


Data

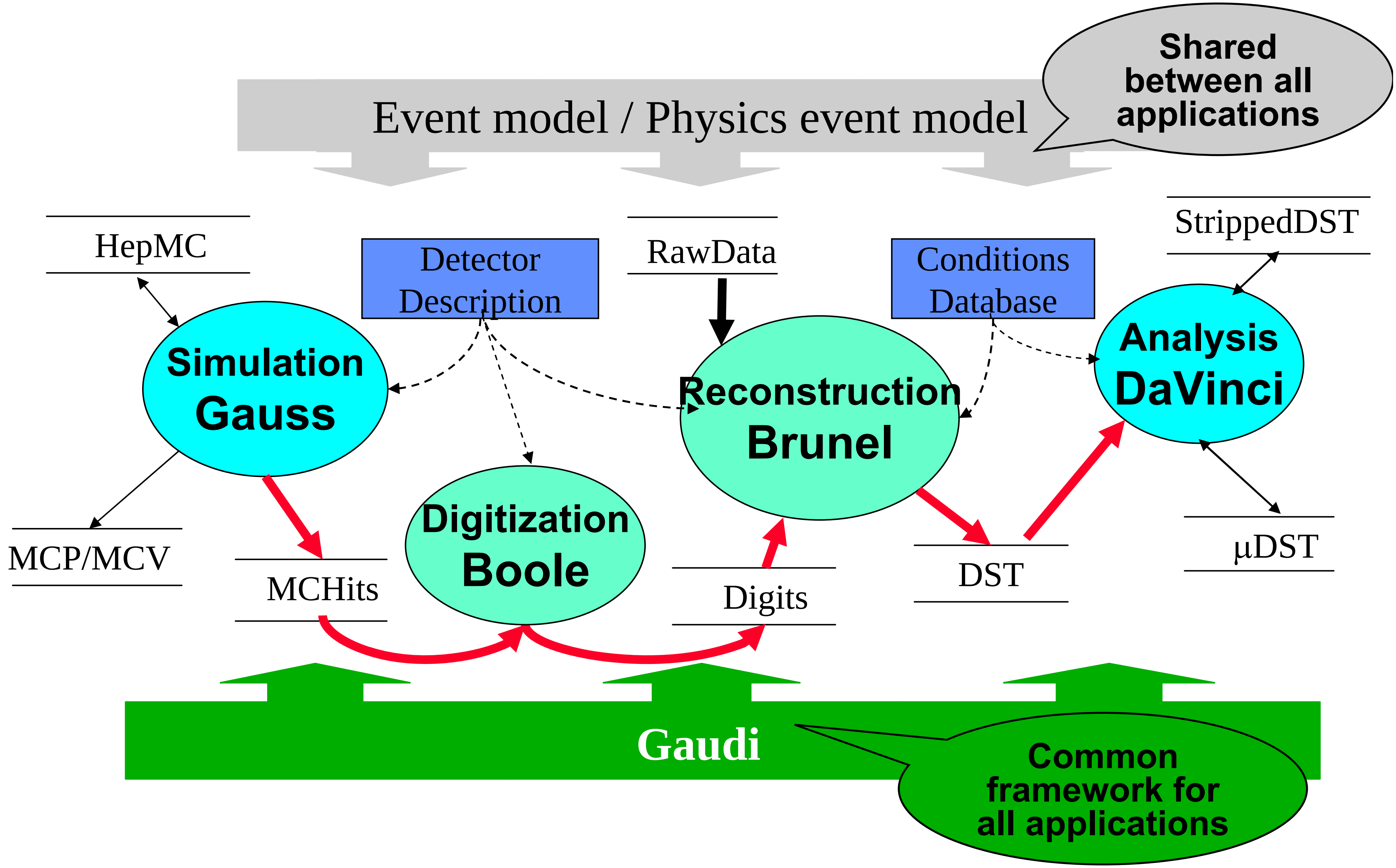


B →J/Ψ K*

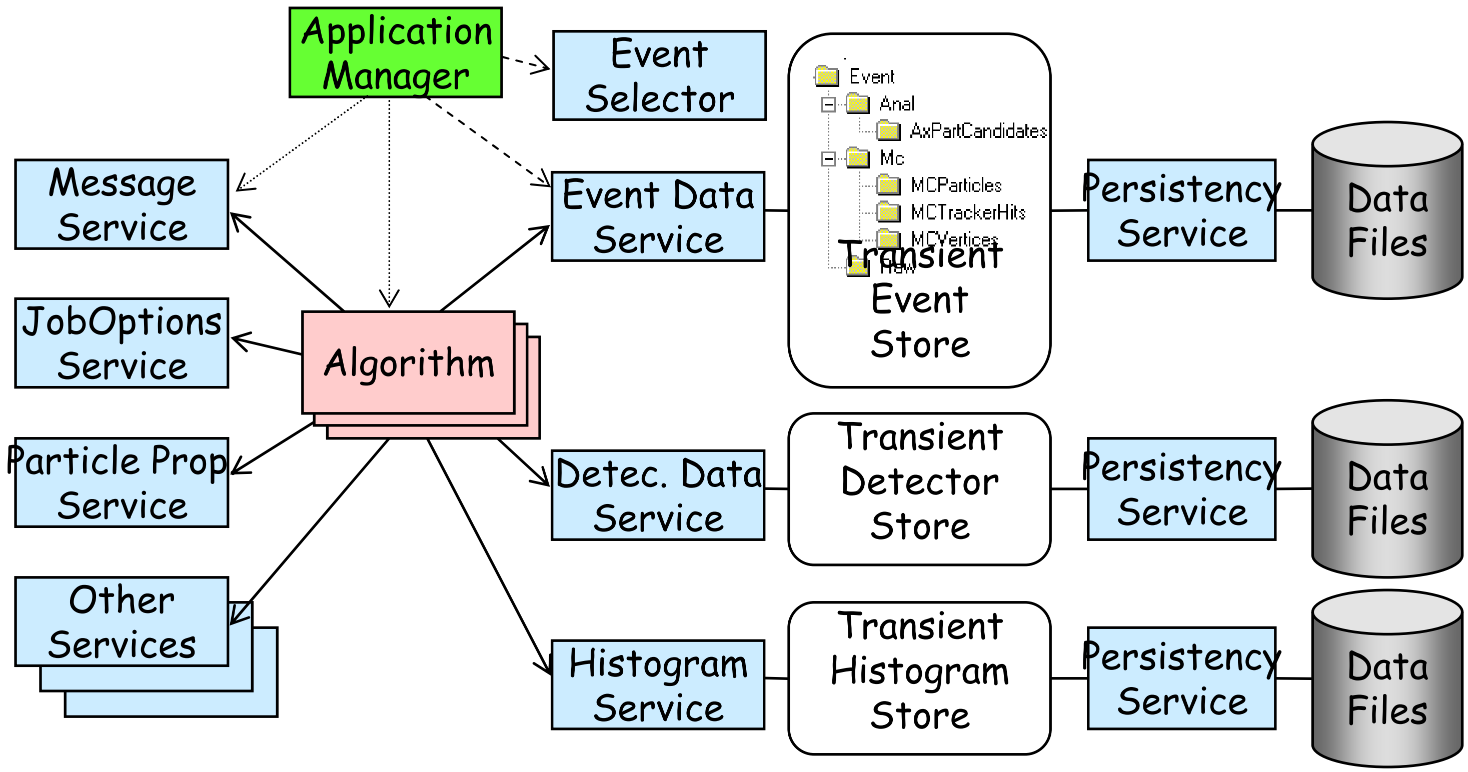
Traditional flow of simulated data and applications



Phase as encapsulated in the LHCb applications



Gauss and Boole are based on the GAUDI framework



- Separation between “data” and “algorithms”
- Separation between “transient” and “persistent” representations of data
- Simulation code encapsulated in specific places (*Algorithms*, *Tools*)
- Well defined component “interfaces”

Experiments simulation software

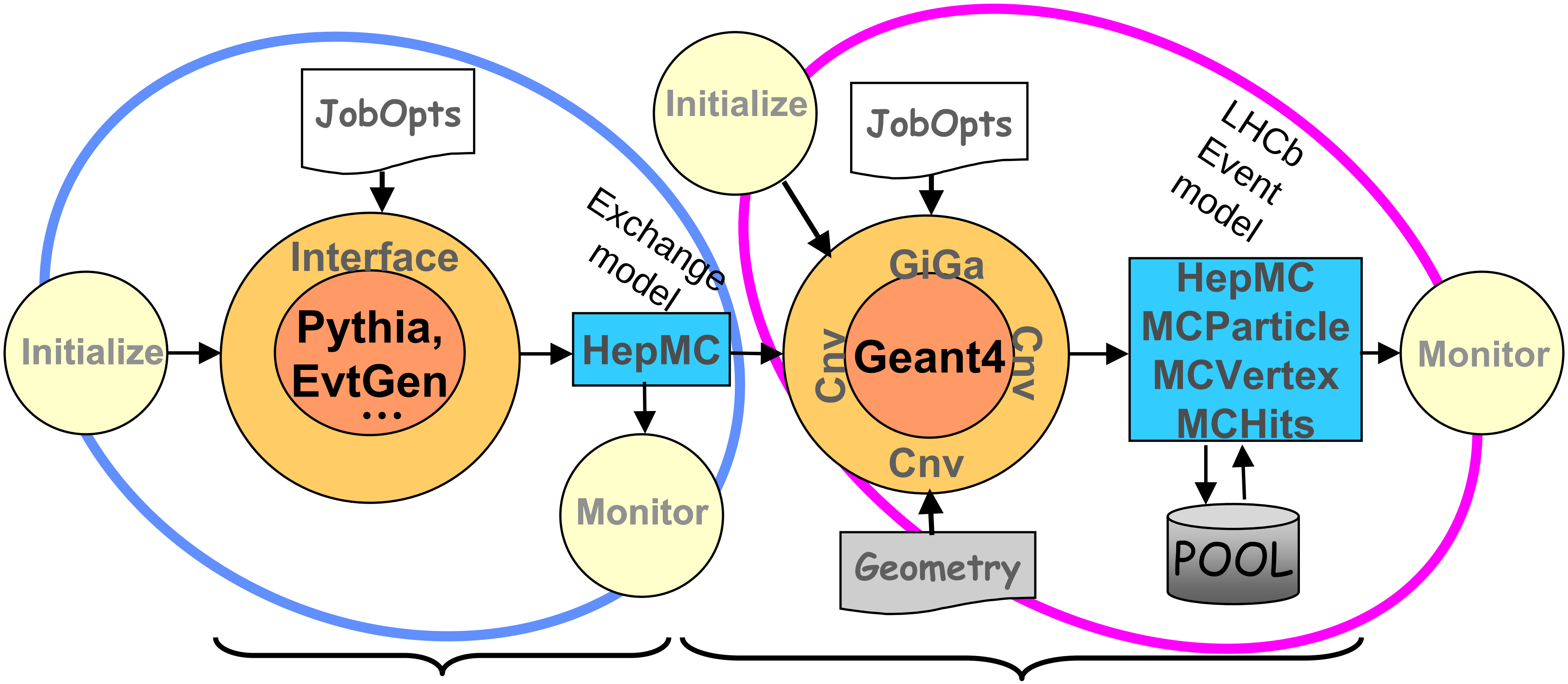
- **HEP experiments have their own software frameworks and use external packages developed in the physics community for Generators and for Transport in the detectors**
  - **Athena (ATLAS), CMSSW (CMS), Gauss (LHCb), VMC (ALICE, CBM@GSI, Minos), bbsim (BaBar), etc.**

- **Response of the detectors is often in-"house" and requires detectors experts**
  - **tuned first with test beam data, then with measurements in the experiment**
  - **in LHCb in a separate application, Boole**

- **The Gauss applications provide**
  - **Interfaces to event generators**
  - **Interface to transport code**
  - **Event model for Generator and MC truth, and Persistency**
    - **Access to snap-shots of process to understand what happened**
  - **Histograms, messaging**
  - **Physicists in the experiments are shielded from Generators and transport code (eg. Geant4) to different degrees**
    - **different in the various experiments**
    - **different for different roles**

Structure of 'a' Gauss application

Two INDEPENDENT phases normally run in sequence in a single job



Event Generation

primary event generator
specialized decay package
pile-up generation

Detector Simulation

geometry of the detector (LHCb → Geant4)
tracking through materials (Geant4)
hit creation and MC truth information (Geant4 → LHCb)

Event generation

- Many programs available in the physics community to generate primary collisions
  - specialized for hard processes, resonance, decays, parton showers, hadronization, etc.
    - often best at given task, but not always directly usable by experiments
  - general-purpose
    - PYTHIA and PYTHIA8 (T.Sijostrand et al.)
    - HERWIG and HERWIG++ (G.Marchesini et al.)
    - ISAJET (H.Baer et al.)
    - SHERPA (F.Kraus et al.)
  - LCG Generators Service at CERN provides libraries for many of them
    - http://lcgapp.cern.ch/project/simu/generator/
- Experiments generally use one generator for massive production and make smaller data sets with others
    - Pythia/Herwig have different hadronisation mechanism (clusters as opposed to strings) for example

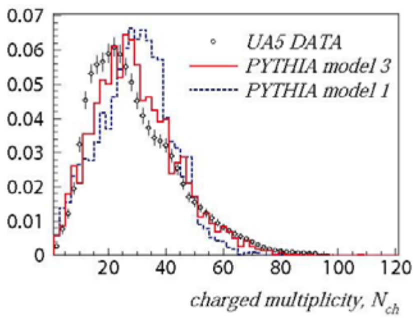
  and specialized codes when necessary
    - BaBar and LHCb use EvtGen (D.Lange and A.Ryd) for b-hadrons decays
    - ATLAS and CMS use MCatNLO (S.Frixione, B.Webber) , AlpGen (M.Mangano et al) for matrix elements to feed to general purpose
    - ALICE uses Hiijing (M.Gyulassy and X.-N. Wang) for Pb-Pb interaction

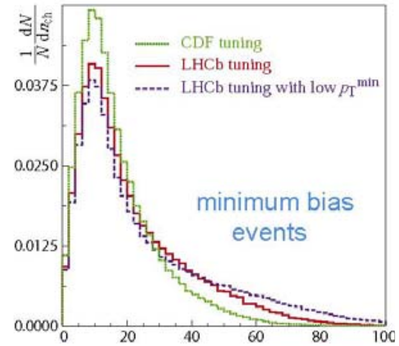
Generator phase in practice

- **In LHCb in Gauss proton-proton interactions at E_{cm}= 14 TeV are generated using Pythia 6**
  - **Pythia is the reference generator, but possible to use others**
  - **First simulate the hard processes in colliding particles, then the subsequent hadronisation**
  - **LHAPDF for Parton Density Functions**

  particles are then decayed by EvtGen (+PHOTOS and Pythia)

- **Various type of events generated**
  - **Minimum bias**
    - **includes hard QCD processes, single and double diffractive events**
  - **Signal b and bb events**
    - **obtained from minimum bias events with b or b-hadron**
  - **Cosmics and "particle guns (ie. a given particle with given kinematic)"**

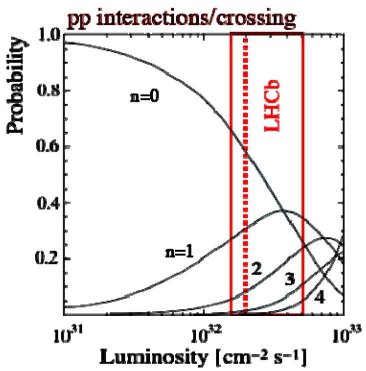




Choose Pythia setting that would agree with experiments measured particle multiplicities

- **Simulate running conditions**

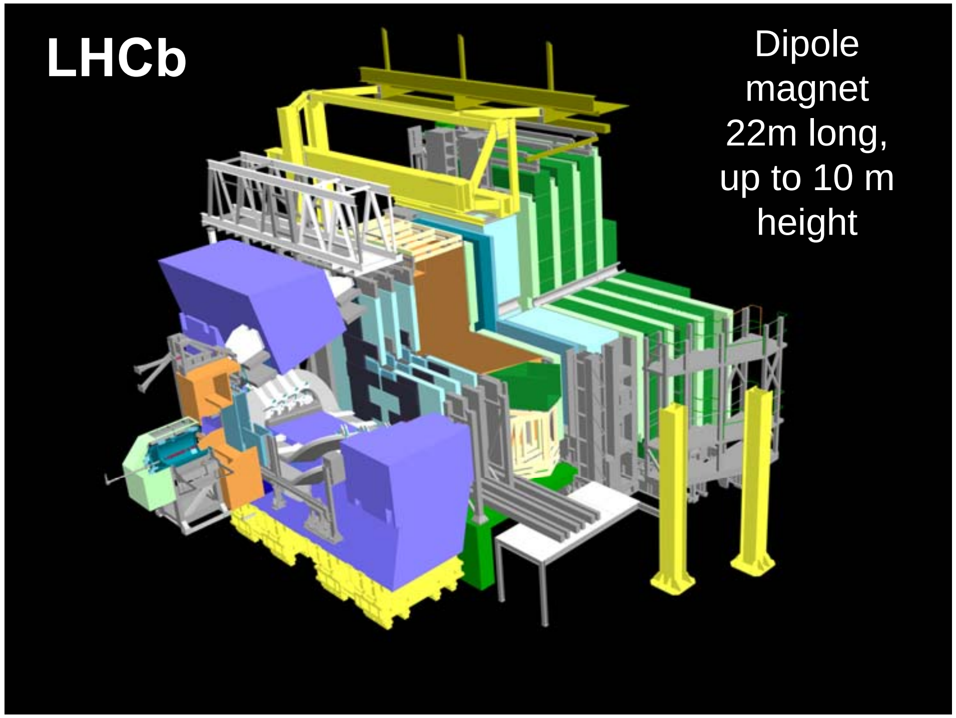
  Luminous region with smearing of primary vertex according to bunch sizes



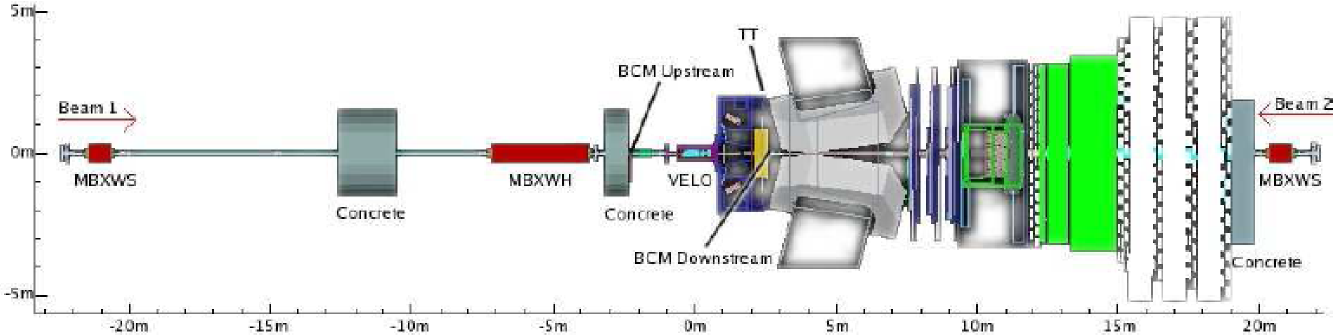
Pileup (number of pp collisions per bunch) at running luminosity

Detector simulation

Need to describe how particles traverse the experimental setup and what happens to them

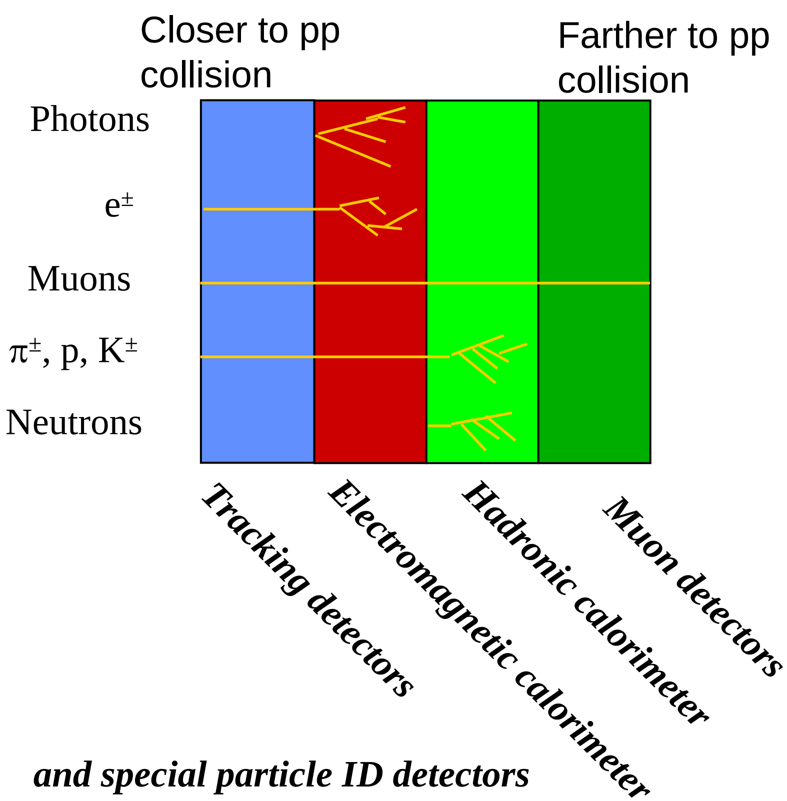


For some studies need to extend the description to tunnel close by or include the cavern

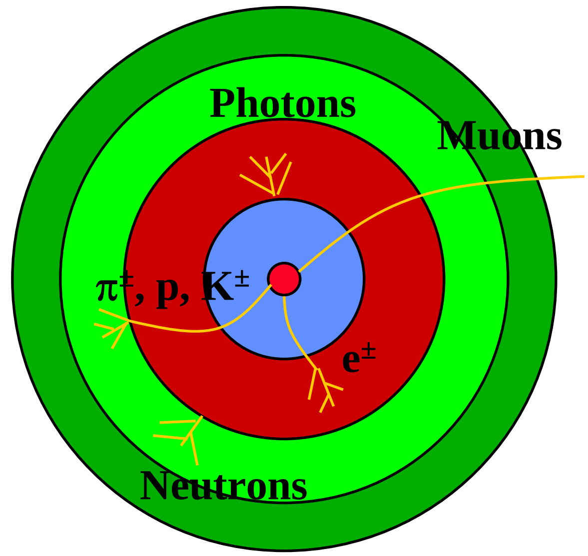


Transport through detectors

Complex detectors and a large variety of physics processes that need to be simulated



and special particle ID detectors (eg. Ring Imaging Cerenkov)



At LHC particle energies range from few MeV to TeV

What is signal for one detector can be background for another

HEP Experiments use in their framework packages developed in the physics community for transport of particles: GEANT4 and/or FLUKA

Detector simulation: transport codes

- **GEANT4 is a C++ toolkit to track particles through the detector developed in the Physics community**
  - **GEANT4 International Collaboration ~ 10 year old**
  - **used in HEP, nuclear physics, heavy ion physics, cosmic ray physics, astrophysics, space science and medical applications**
  - **GEANT4 is the successor of GEANT3, the world-standard toolkit for HEP detector simulation**
  - **http://geant4.web.cern.ch/geant4/**

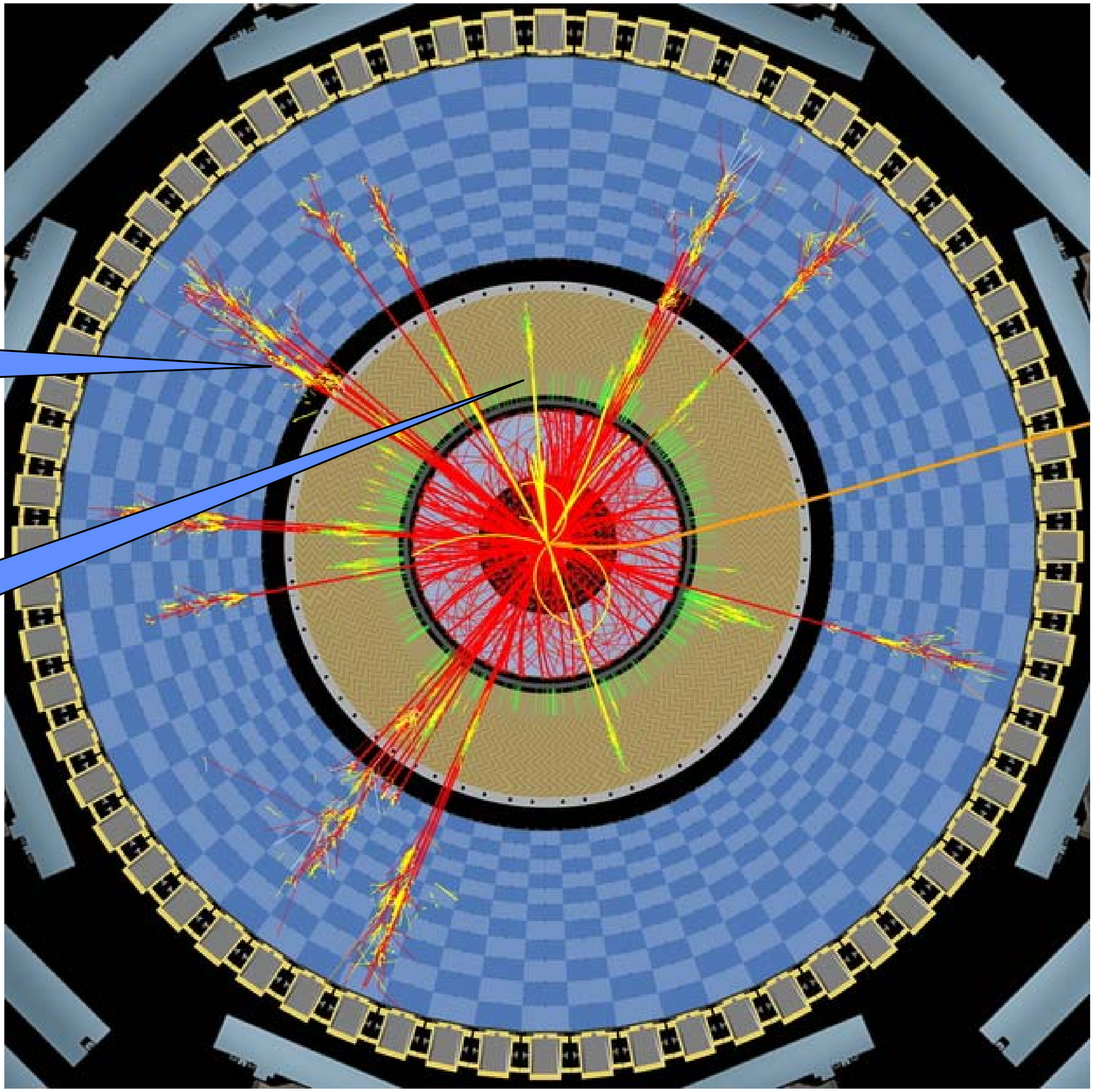
- **All LHC experiments use GEANT4 for transporting particles in the experimental setup and simulating the physics processes that can occur**
  - **Navigation in EM fields**
  - **Physics processes for a variety of particles at different energies**

- **ALICE also uses FLUKA (A.Ferrari et al)**
  - **FORTRAN-based**
  - **Couples low energy neutron transport and particle transport**
  - **LHCb uses it for prompt radiation levels and radiation protection estimates (activation on materials and ambient dose)**

Detector simulation: Geant 4

- **GEANT4 coverage of physics comes from mixture of theory-driven, parameterized, and empirical formulae. Both cross-sections and models (final state generation) can be combined in arbitrary manners as necessary.**
  - **Standard and Low energy EM processes, Hadronic processes, Optical photon processes, Decay processes, etc.**
- **Each particle is moved in steps (few microns to cm)**
  - **For each step:**
    - **The material the particle is in is found**
    - **The energy loss and directional change due to multiple scattering is applied**
    - **The effects of electric and magnetic fields on charged particles are calculated**
    - **The particle might generate photons by bremstrahlung or Cherenkov radiation**
    - **The particle might decay in flight**
    - **The particle might collide with a nucleus and generate more particles**
    - **If the particle enters or leaves a detector it is recorded**

EM and hadronic showers in material



Hadronic shower shapes

Electro-magnetic showers

In dense material number of particles created and tracked by Geant grows quickly and with that the CPU time!

Atlas event from their publicity page

GiGa

- **Gauss uses a dedicated Gaudi service to interact and communicate with Geant4**
  - **minimizes the couplings to Geant4**

- **GIGA = GEANT4 Interface for Gaudi Applications or Gaudi Interface to GEANT4 Applications**
  - **GEANT4 callable and controllable from within GAUDI environment**
  - **common detector geometry source used by other applications (reconstruction, visualisation)**
  - **use of Gaudi features as algorithms, tools, services**
  - **use of common services (ex. RandomNumberSvc, MagneticFieldSvc, DetectorDataSvc, etc.)**
  - **access to internal Geant4 event loop via GiGaRunManager**
  - **allows loading external physics lists**
  - **instantiates (using Abstract Factory pattern) different "actions" (makes them to be plugable components)**

Encapsulation of Geant4 in Gauss via GiGa

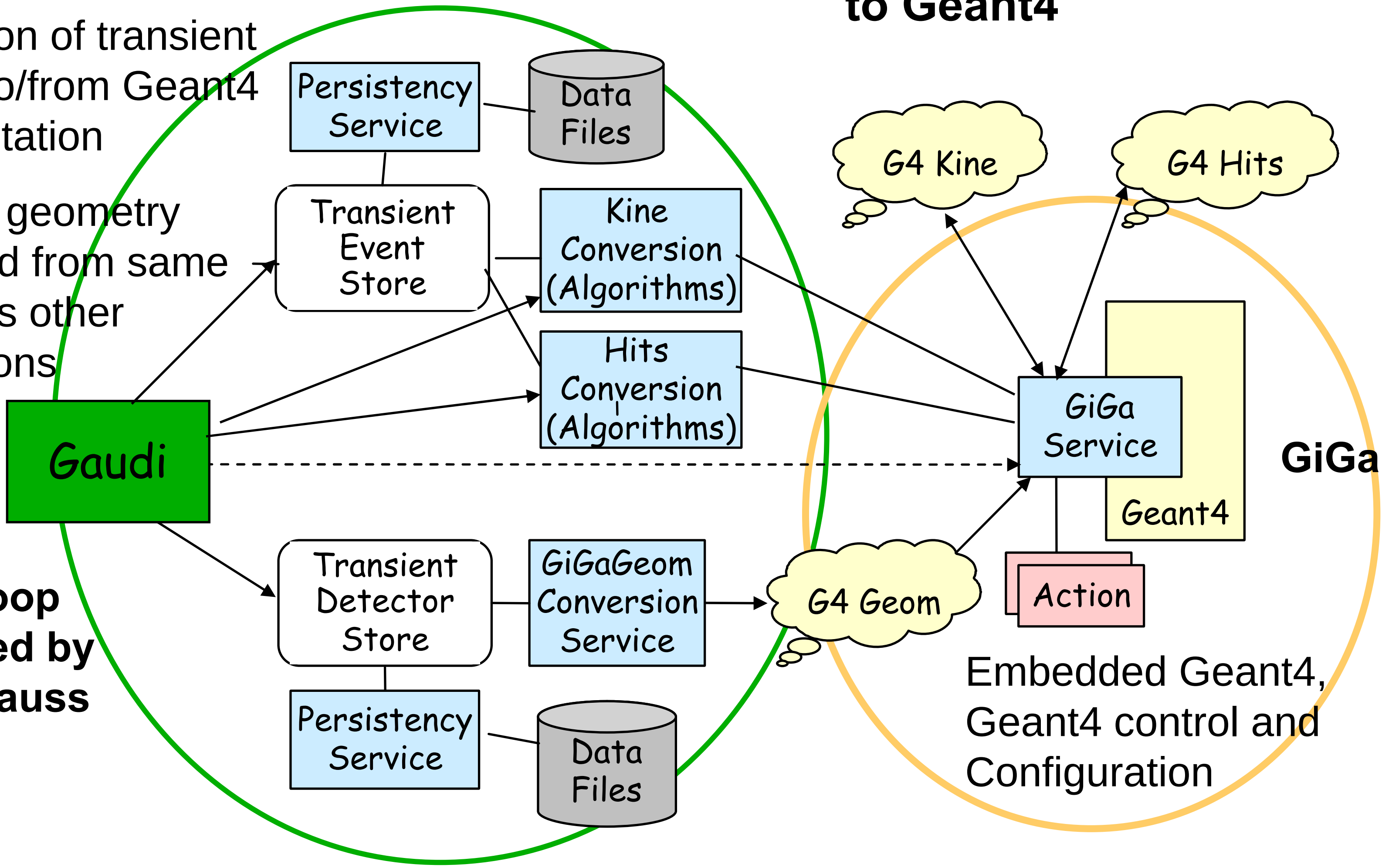
GiGa

Convertion of transient objects to/from Geant4 representation

Detector geometry converted from same source as other applications

Event Loop controlled by Gaudi/Gauss

Single point of connection to Geant4



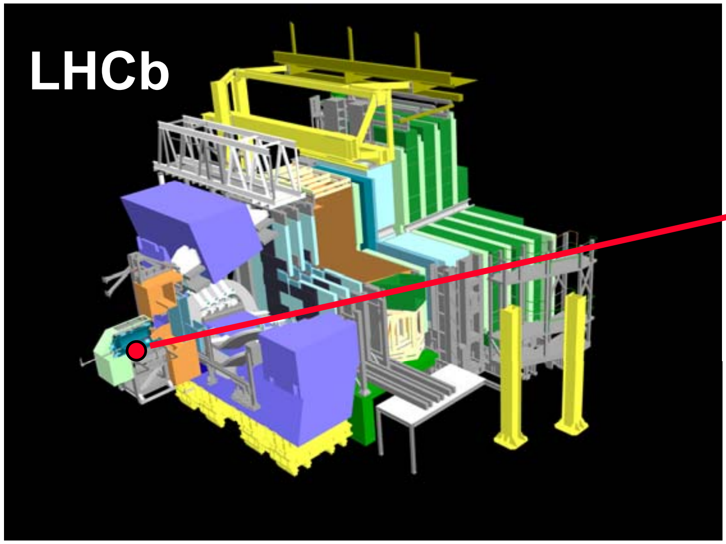
Embedded Geant4, Geant4 control and Configuration

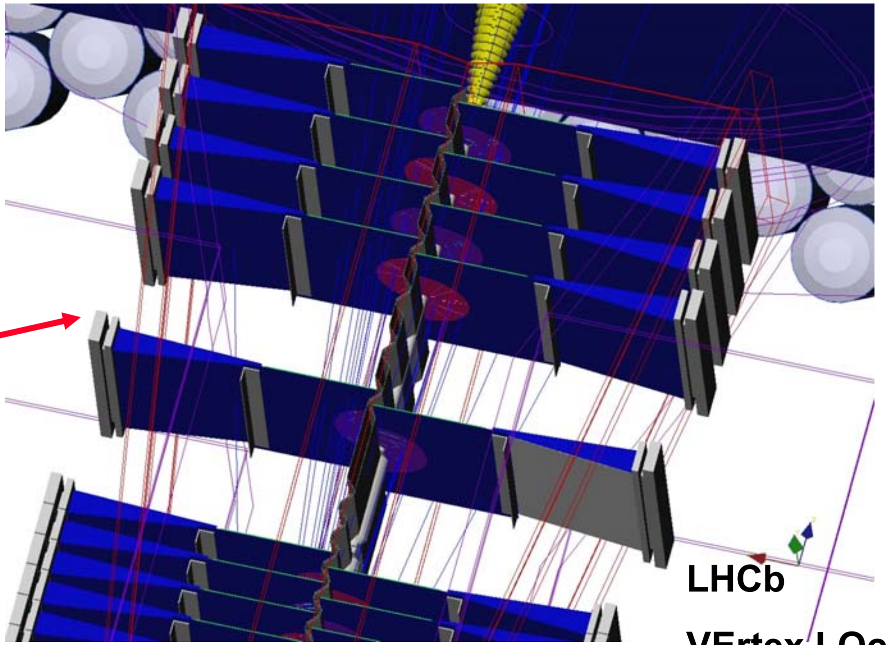
Geometry modeling

Need accurate modeling to have the most accurate results but more volumes, more memory, more CPU time

A wide variety of dimensions in an HEP experiment

Choose depending on relevance to physics study



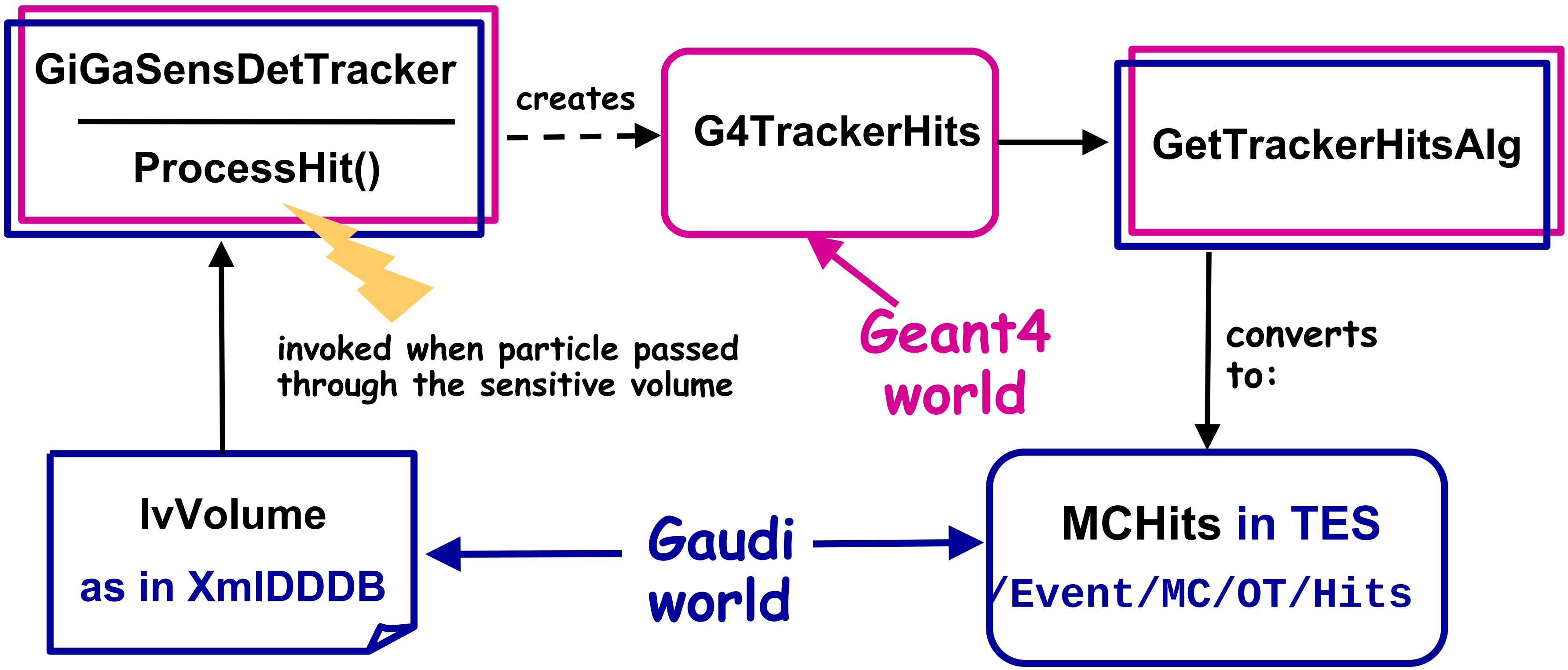


LHCb

VErtex LOcator

For trackers detailed description of all active and passive components; material budget

Detector simulation




```
<logvol material “Silicon”
        name="lvLadder1"
        sensdet="GiGaSensDetTracker/TTSDet">
```

GetTrackerHits configuration for TT

```
GetTTHits.CollectionName = “TTDSet/Hits”;
GetTTHits.MCHitsLocation = “/Event/MC/TT/Hits”;
GetTTHits.MCHitsLocation = “/dd/Structure/LHCb/BeforeMagnetRegion/TT”;
```

Control of Detector simulation

- **Uses a special GiGa component:**

```
geo = GiGaInputStream( "Geo", ExecuteOnce = True,
                       ConversionSvcName = "GiGaGeo",
                       DataProviderSvcName = "DetectorDataSvc")
```

giving it the list of detectors to simulate

```
geo.StreamItems += {"/dd/Structure/LHCb/BeforeMagnetRegion/Velo"};
geo.StreamItems += {"/dd/Structure/LHCb/BeforeMagnetRegion/Velo2Rich1"};
geo.StreamItems += {"/dd/Structure/LHCb/BeforeMagnetRegion/Rich1"};
geo.StreamItems += {"/dd/Geometry/BeforeMagnetRegion/Rich1/Rich1Surfaces"};
...
geo.StreamItems += {"/dd/Structure/LHCb/DownstreamRegion/Hcal"};
geo.StreamItems += {"/dd/Structure/LHCb/DownstreamRegion/Muon"};
```

can give a different list: switch off some detectors, add some new, a test beam setup

  - **Just need the sensdet Keyword in the XML Detector Description**

- **In future version Gauss() will provide a control do it is in a simpler way**

```
Gauss().DetectorGeo = {"VELO":['Velo','PuVeto'], "TT":['TT'], "IT":['IT'],
                       "OT":['OT'], "RICH":['Rich1','Rich2'], "CALO":[],
                       "MUON":[], "MAGNET": True }
```

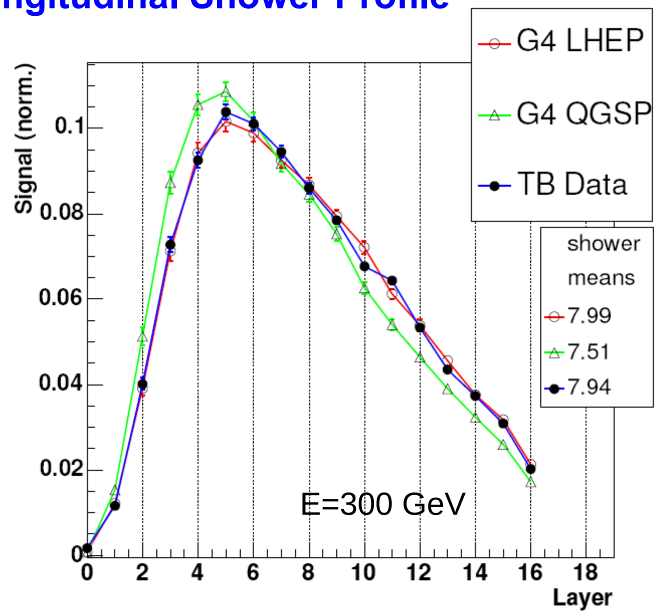
Physics processes and cuts

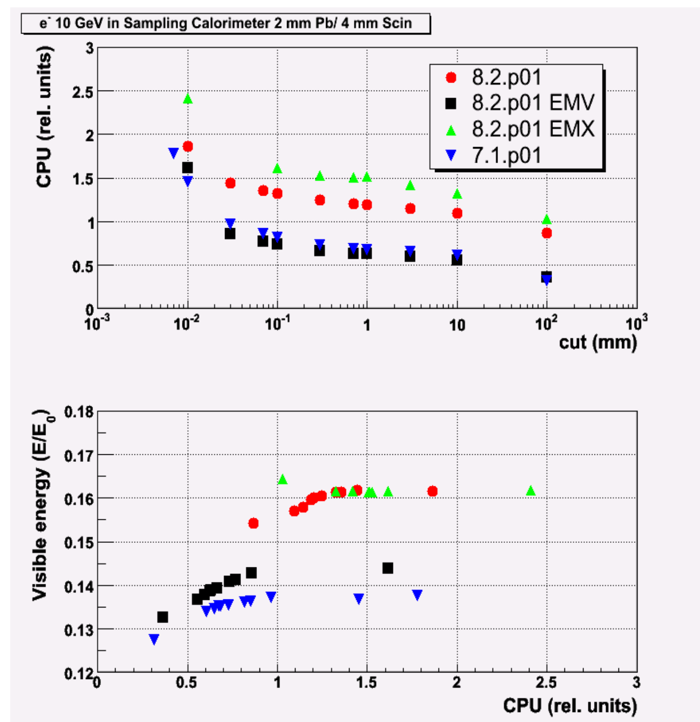
Need to describe physics processes as accurate as needed by resolution of detectors

Choose appropriate physics models and set cuts: more accuracy → more CPU

Timing Performance vs Range Cut

CMS: Pion HCAL
Longitudinal Shower Profile





V. Ivanchenko, Geant4

In LHCb full simulation QGSP takes ~ 10% longer that LHEP

Physics processes and lists

- **Geant4 has a big variety of processes that can be combined as necessary in Physics Lists:**
  - **crucial part of the whole simulation program**
  - **most of the processes are already implemented in Geant4**
  - **some specific processes needed implementation for LHCb**
    - **for RICH: photoelectric process (creation of photoelectrons in HPDs), energy loss in the silicon of HPDs**
- **GiGa modular physics lists**
  - **allows dynamic loading (at run time) of particular physics “sublists”**
- **Geant4 provides pre-packaged Physics Lists with suggestions for use**
  - **we base our run time physics list on Geant4 existing one**
  - **the default one we used so far for production, LHEP, has some known issues**

Configuration of Physics lists to use

- Defined as list of options that can be changed

```
GiGa.ModularPL.PhysicsConstructors +=
  { "GiGaExtPhysics<GeneralPhysics>/GeneralPhysics" };
GiGa.ModularPL.PhysicsConstructors +=
  { "GiGaExtPhysics<EMPhysics>/EMPhysics"           };
GiGa.ModularPL.PhysicsConstructors +=
  { "GiGaExtPhysics<MuonPhysics>/MuonPhysics"       };
GiGa.ModularPL.PhysicsConstructors +=
  { "GiGaExtPhysics<HadronPhysicsLHEP>/LHEPPhysics" };
GiGa.ModularPL.PhysicsConstructors +=
  { "GiGaExtPhysics<IonPhysics>/IonPhysics"         };
GiGa.ModularPL.PhysicsConstructors +=
  { "GiGaPhysConstructorOp"                         };
GiGa.ModularPL.PhysicsConstructors +=
  { "GiGaPhysConstructorHpd"                        };
```

LHCb RICH processes

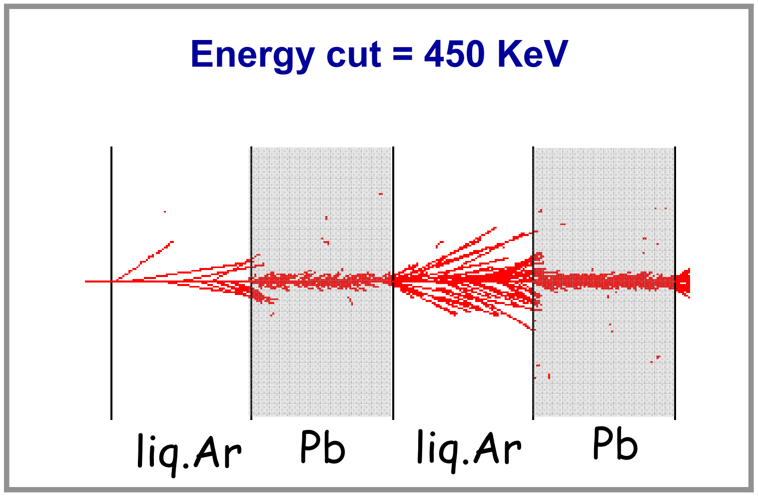
```
GiGa.ModularPL.PhysicsConstructors -=
  { "GiGaExtPhysics<HadronPhysicsLHEP>/LHEPF
GiGa.ModularPL.PhysicsConstructors +=
  { "GiGaExtPhysics<HadronPhysicsQGSP_BERT_HP>/QGSP_BERT_HPPhysics" };
```

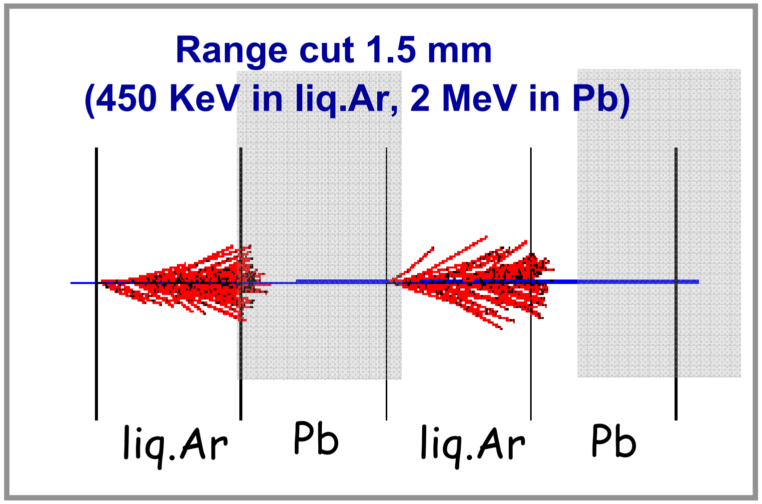
in old MuonLowEnergy.opts set a different hadronic physics list with neutrons to thermal energy

in the process of passing the controls to Gauss().PhysList = “LHEP”

Geant4 production thresholds

- **Geant4 has production thresholds for EM processes**
  - **Specify range (which is converted to energy for each material) at which continuous loss begins, track primary down to zero range**
  - **Create secondaries only above specified range, or add to continuous loss of primary for secondaries of less energetic than travelling the required range in the current material**





G4 Range cuts applied:

10 mm for gamma's

5 mm for e+

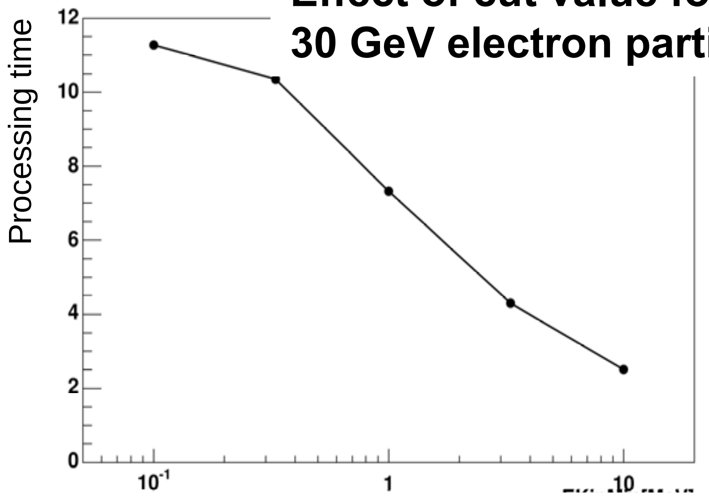
10 km for e- to have delta-rays off in trackers but affect dE/dx (see later): will change post-MC09

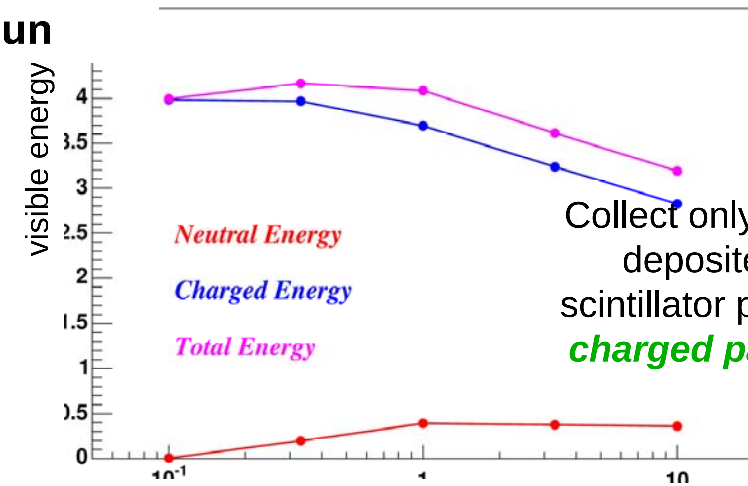
LHCb tracking cuts

- **Introduce tracking cuts on E_{kin} of particles of all type (special Gauss stepping actions)**
  - **Track particle until cut-off energy is reached, stop it at that point**

```
GiGa.RunSeq.Members  += { "TrCutsRunAction/TrCuts"};
```

Effect of cut value for ECAL with 30 GeV electron particle gun





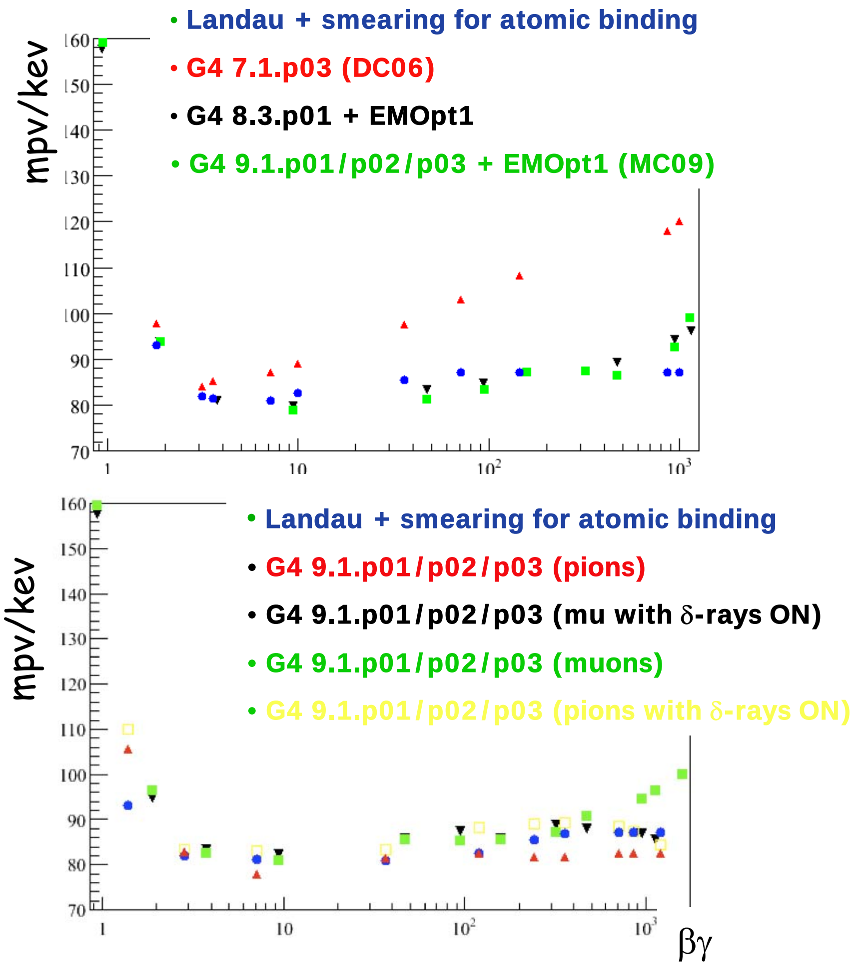
Collect only energy deposited in scintillator plates by ***charged particles***

Energy threshold below which the particle is not tracked.

- **Possible also to set cuts per region**
  - **For example we turn on delta rays in RICH1 Aerogel (set in Simulation.xml)**

dE/dx on thin layers and choice of cuts

Compare dE/dx in silicon from simulation with particle guns (μ of fixed E) to simple model describing data



Need to adapt to changes in Geant4

in G4 7.1.p03 the dE/dx intrinsic width was too small (atomic binding correction was missing) → smearing applied in digitization phase

in G4 9.1 the atomic binding is simulated (width is close to expectation) → correction no longer needed.

still problem in vertex detector (220μm Si) simulation: Landau width too wide w.r.t. test beam data

- better agreement with δ-rays on
  - still problem at high $\beta\gamma$ for muons

Physics tuning: Aerogel Test Beams

Validation of different RICH-specific process (Cherenkov, Rayleigh scattering, etc) using test beam data:

- **specific processes developed for LHCb simulation:**
  - **Photoelectric process: creation of photoelectrons in HPDs.**
  - **Energy loss: in the silicon of HPDs. (speed vs precision)**

Results from 2001 and 2003 Test beam data using Geant4 simulation

- **different aerogel tiles**
- **various pixel HPDs**

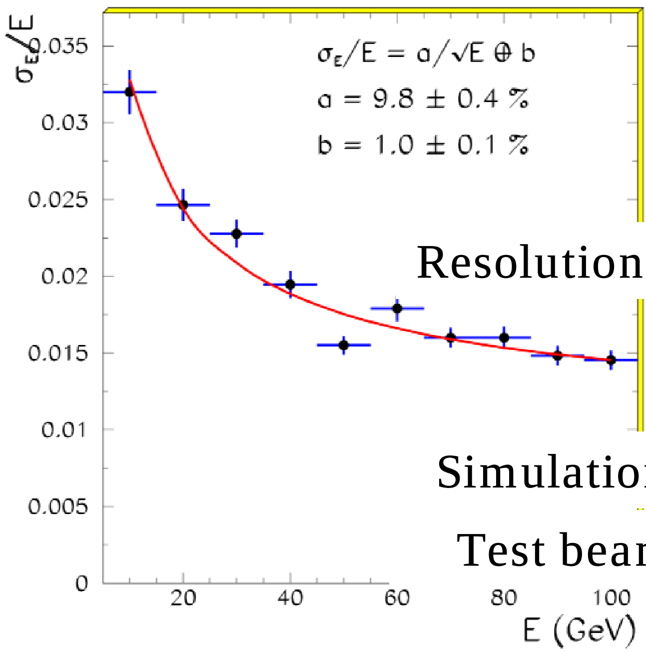
Example: NIM A 519 (2004) 493-507

Novosibirsk	No Filter	Filter D263	
4 cm	9.7 ± 1.0 11.5 ± 1.2	6.3 ± 0.7 7.4 ± 0.8	Data MC
8 cm	12.2 ± 1.3 14.7 ± 1.6	9.4 ± 1.0 10.1 ± 1.1	

Physics tuning: ECAL test beam

Simulation of ECAL (Pb-Scintillator Shashlik) with Geant4 tuned to test beam data

- Geant4 processes + Birk's saturation law
- + timing - share energy deposition in two consecutive bins as from test beam measurements
- + non uniformities – more detailed studies in progress with this year test beam data (preliminary results from last year showed that parameters reproducing muon test beam data did not reproduce electrons)



Resolution, electrons perpendicular to outer module

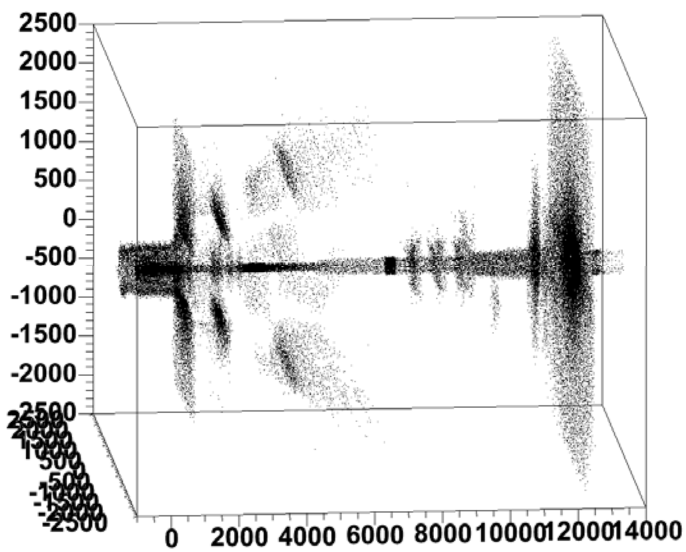
Simulation $\sigma_E/E = (\ 9.8 \pm 0.4\ )\%/E^{0.5} \oplus (\ 1.0 \pm 0.1)\ \%$

Test beam $\sigma_E/E = (\ 9.4 \pm 0.4\ )\%/E^{0.5} \oplus (\ 0.83 \pm 0.02\ )\ \%$

Understanding the simulation as a whole

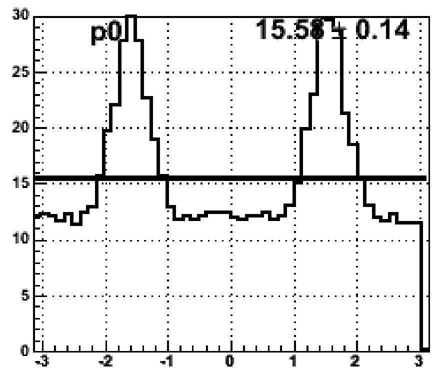
- **To understand in depth the simulation, the information about the history of what happened in the detector is very important (trajectories and processes originating them)**

Where particles originate in the detector

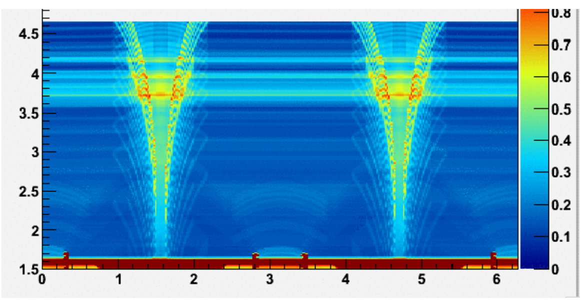


Radiation length as seen by photon convertion

After Vertex Locator



and match it with what seen in data
and what put in simulation



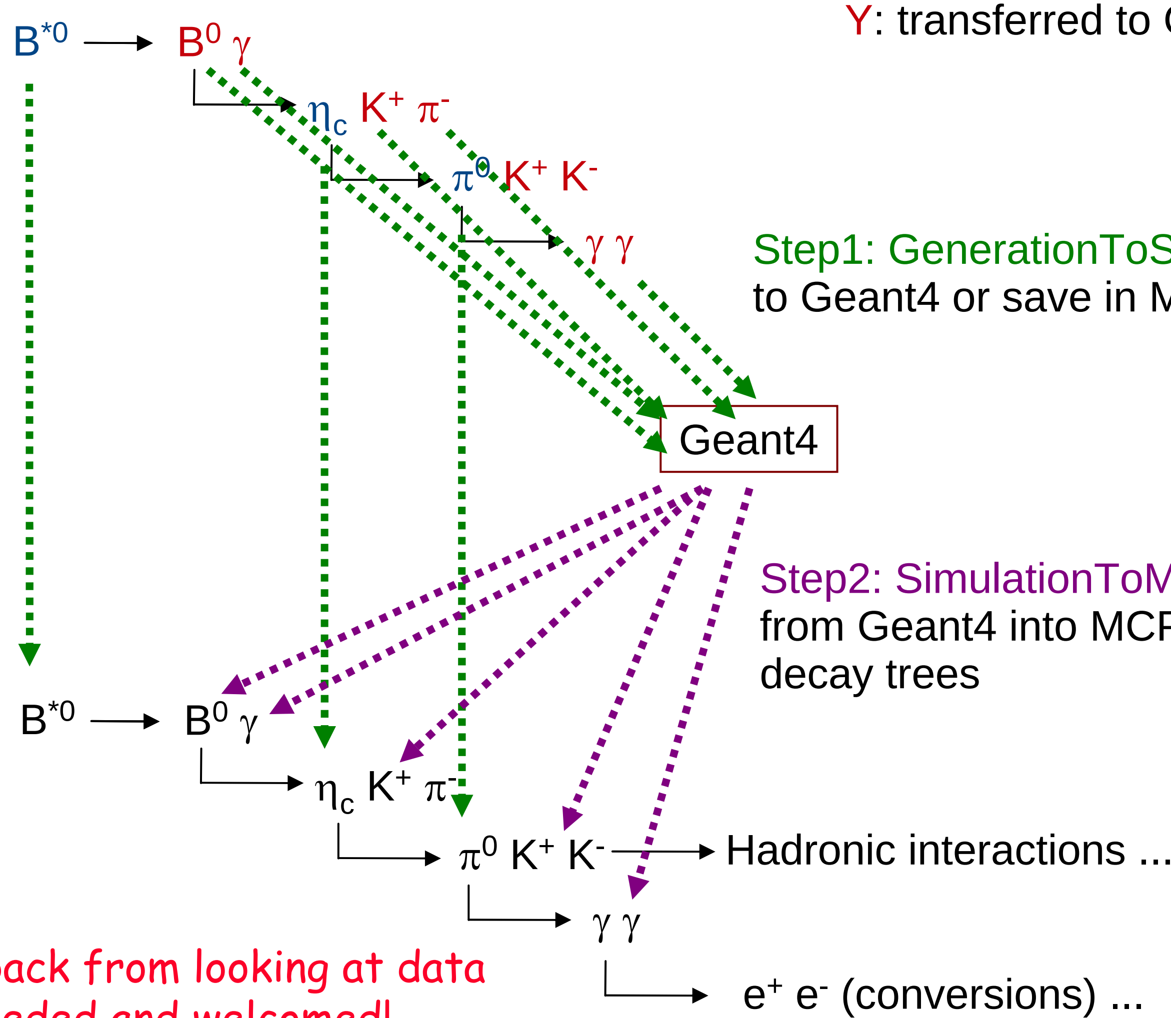
available from Gauss web

The Monte Carlo truth

- Generator has its own event record with all history from proton beams to decay products
- MC truth filled with both generator information and results of tracking through detector at the end of a Geant4 event
  - Dedicated LHCb event data classes: MCParticles and MCVertices
    - Relationship between them holds the tree structure
    - An MCParticle has one originVertex() and more than one endVertices() (Bremsstrahlung)
    - An MCVertex has a mother particle (but for primay vertex) and product particles
    - We keep originating process identifier in MCVertex
  - Link from other classes (MCHits for example, but also possibility to associate them to reconstructed Tracks, physics Particles)
  - These classes are written out by the simulation and accessed in a variety of ways in successive processing
- Generator to Geant4 and MC truth
  - Pass only particles from generator to Geant4 which will interact with detector, that is to say particles with non-zero travel lenght.
  - All other particles are saved directly in MCParticle container, and the decay chains are restored at the end of the processing by Geant4.
    - Allows to save memory and time when creating particles not used.
  - All particles are now saved in MCTruth (hadrons, excited states, ...) except all special particles (quarks, strings, ...)

Example of how particles are given to G4

X: saved to MCTruth directly
Y: transferred to Geant4



Step1: GenerationToSimulation, transfer to Geant4 or save in MCParticle

Step2: SimulationToMCTruth, transfer from Geant4 into MCParticle and rebuild decay trees

Feed-back from looking at data needed and welcomed!

The Monte Carlo truth

- Necessary to decide a priori what to store in Geant4
  - cannot store all particles in the showers in calorimeter!!!
  - internally use HepMC structure and retrieve originating process
  - set of criteria can be changed in job options
- The set of conditions for production is the following:

```
GiGa.TrackSeq.PostTrack.StoreAll             = false;
GiGa.TrackSeq.PostTrack.StorePrimaries       = true;
GiGa.TrackSeq.PostTrack.StoreForcedDecays    = true;
GiGa.TrackSeq.PostTrack.StoreMarkedTracks    = true;
// The following only for z<12280mm (i.e.PRS/SPD)
GiGa.TrackSeq.PostTrack.StoreByOwnEnergy      = true;
GiGa.TrackSeq.PostTrack.OwnEnergyThreshold = 100.0 * MeV;
GiGa.TrackSeq.PostTrack.StoreByChildProcess   = true;
GiGa.TrackSeq.PostTrack.StoredChildProcesses = {"RichG4Cerenkov"};
GiGa.TrackSeq.PostTrack.StoreByOwnProcess    = true;
GiGa.TrackSeq.PostTrack.StoredOwnProcesses = {“Decay”}
```

pp collision

seen by detectors

with E > threshold

producing Cerenkov

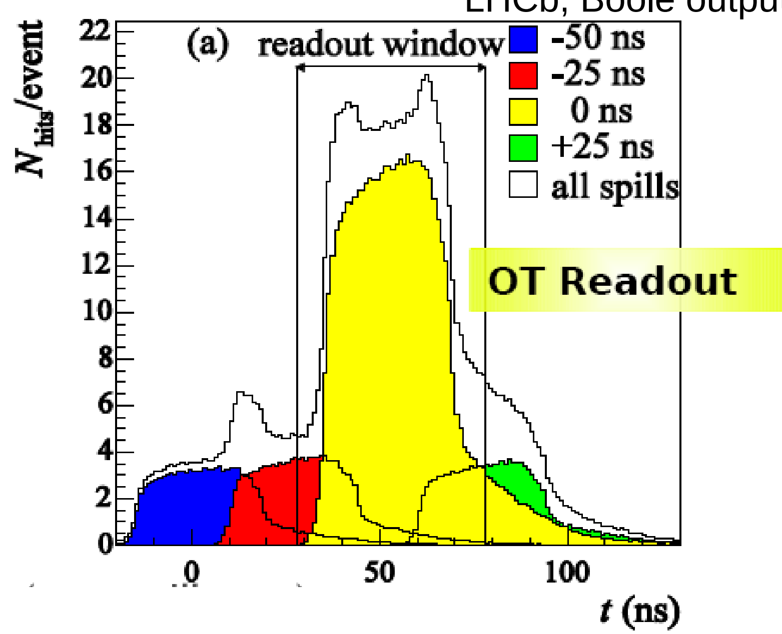
from decay in detector

Simulation of Detector response (aka digitization)

- **Simulation of detector response transforming hits in sensitive detectors to produce digitized data and provide them in DAQ-like format is provided by Boole (separate application)**
  - **Convenience**
  - **Flexibility**
- **Each detector has its own detailed detector response simulation and imperfections**
  - **Detections efficiencies and resolutions are adjusted according to test beam data**
  - **Electronic noise and cross-talk for each specific detector are added**
  - **Time information is correctly taken into account by all detectors**

- **Handling of spill-over effects (data from adjacent beam crossings)**
  - **A detector is sensitive to previous or subsequent bunch crossings depending on its electronics**

LHCb, Boole output



In LHCb last proceessing of Boole is the L0 harwdare trigger emulation

Statistics and CPU time

- **Statistics required of the same order of real events**
  - **for LHC feasible for signal (O(10)-O(10^6)/year), even many times more.**
  - **while impossible for all pp collisions (O(10^7)-O(10^8)/sec) → months of running on the GRID to generate few seconds generic b events of LHC(b) data taking**
- **Simulations are the most CPU time consuming applications of HEP experiments**

  in LHCb the simulation takes 50 + 1 KSI2k sec/event (Gauss + Boole)

  while the reconstruction takes 2.5 KSI2k sec/event
- **Transport through the detectors is the longest**

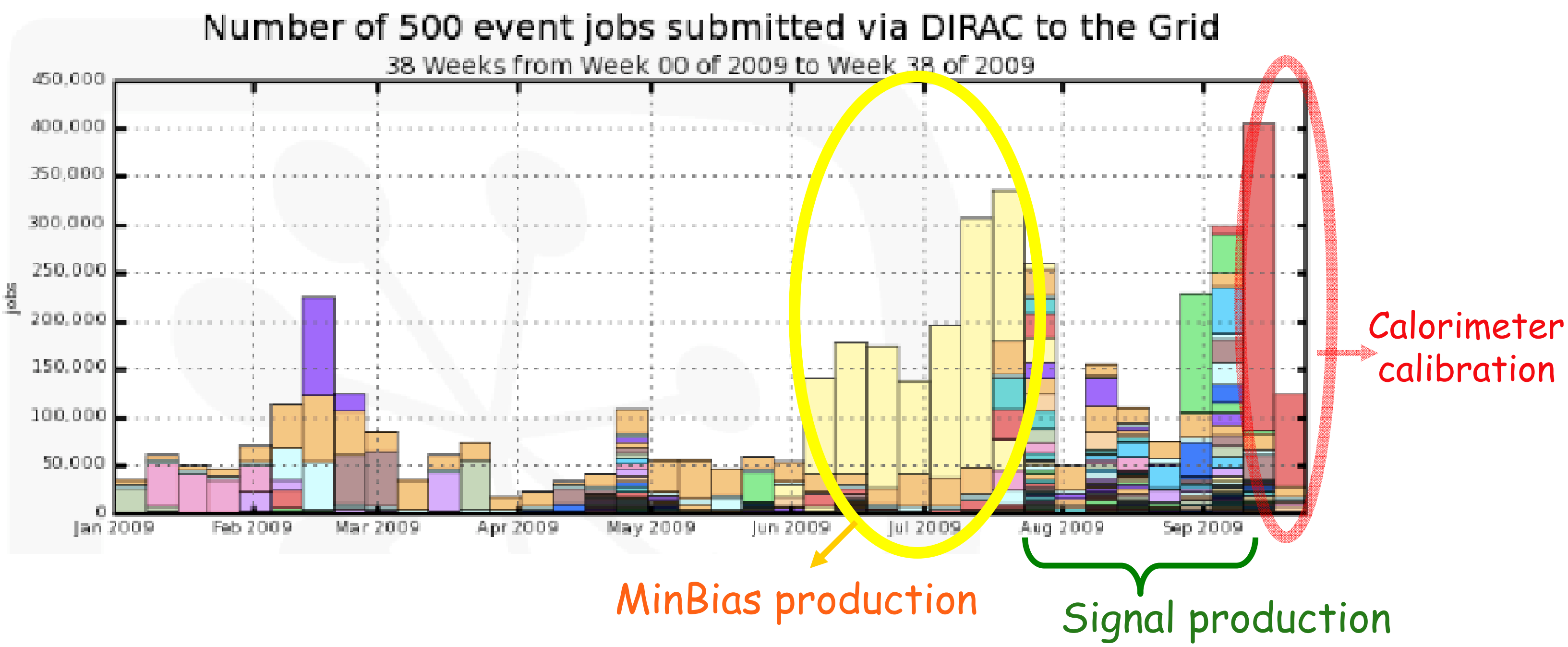
  in LHCb: generator phase for most of the events types is 1/100th of simulation phase
- **The CPU time depends on the complexity of the primary events**

LHCb:	**minimum bias**	**20 KSI2k sec/event**
	generic b events	**50 KSI2k sec/event**
ALICE:	**pp collisions**	**39 KSI2k sec/event**
	Pb-Pb collisions	**7000 KSI2k sec/event with tricks to save time**

- **Require a production system and GRID resources to simulate events**
  - **in LHC experiment computing model Tier2 will be used to produce them and Tier1 to store the output data**
  - **ensure non-overlap of samples with control of random number generators seeds**

MC09 production: few numbers

10^9 MinBias events, few hundreds Millions of signal events (b,c,Z,Higgs...) produced in MC09



- 3M jobs run
- 45000 jobs/day

Simplify simulation by crossing levels

Generators → Specific reaction → parameterized simulation → Interpreted events → Individual Analyses

eg. Atlfast (ATLAS)

- **Advantages:**
  - Fast and flexible for “what if” analysis studies
  - Retains flexibility to choose generators
- **Disadvantages**
  - Often not sufficiently realistic
  - Only certain information, tools available
- **Can use similar techniques at other levels**

Summary

- Events studied in High Energy Physics experiments at more and more complex. With LHC very high energy will be reached.
- Simulations of such events are an integral part of an experiment in all of its phases from design to physics analysis
- Simulations applications are a complex system of dedicated software developed by the physics community and by the experiments
- Considerable computing resources are (and will be) necessary: the GRID is already extensively used for this