

RICH Simulation experience from LHCb

EICUG Software seminar



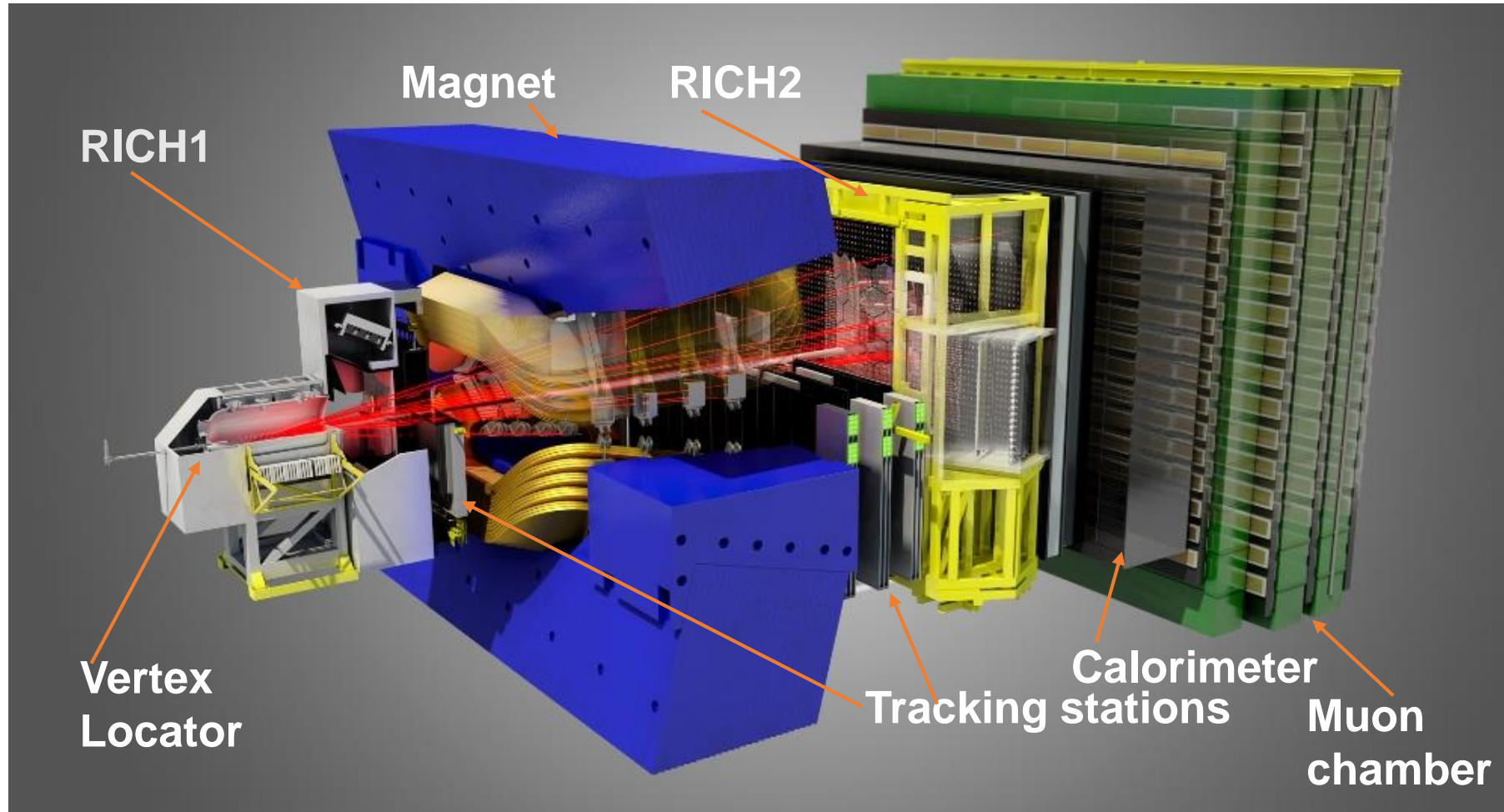
From LHCb RICH software group

Sajan Easo
27-05-2020

Preamble

- Description of LHCb and its RICH system
 - *RICH for RUN1 and RUN2 (2010-2018)*
 - *RICH for RUN3 (2021 onwards)*
- Software framework and simulations
 - *Gaudi software framework and the applications*
 - *GEANT4 in LHCb*
- **RICH Detector description and physics processes**
 - *LHCb specific processes*
 - *Validation using prototypes in testbeams*
- **Comparison with real data from LHCb**
- **Design for further upgrades**

The LHCb Experiment



From 2015:

- pp : $\sqrt{s} = 13$ TeV
- RICH1:
C₄F₁₀ < ~ 60 GeV/c
Photodetectors:
Top + Bottom
- RICH2:
CF₄: < ~ 100 GeV/c
Photodetectors:
Left+ Right

Before 2015:

- pp : $\sqrt{s} = 8$ TeV
Aerogel < 10 GeV/c

Before 2012:

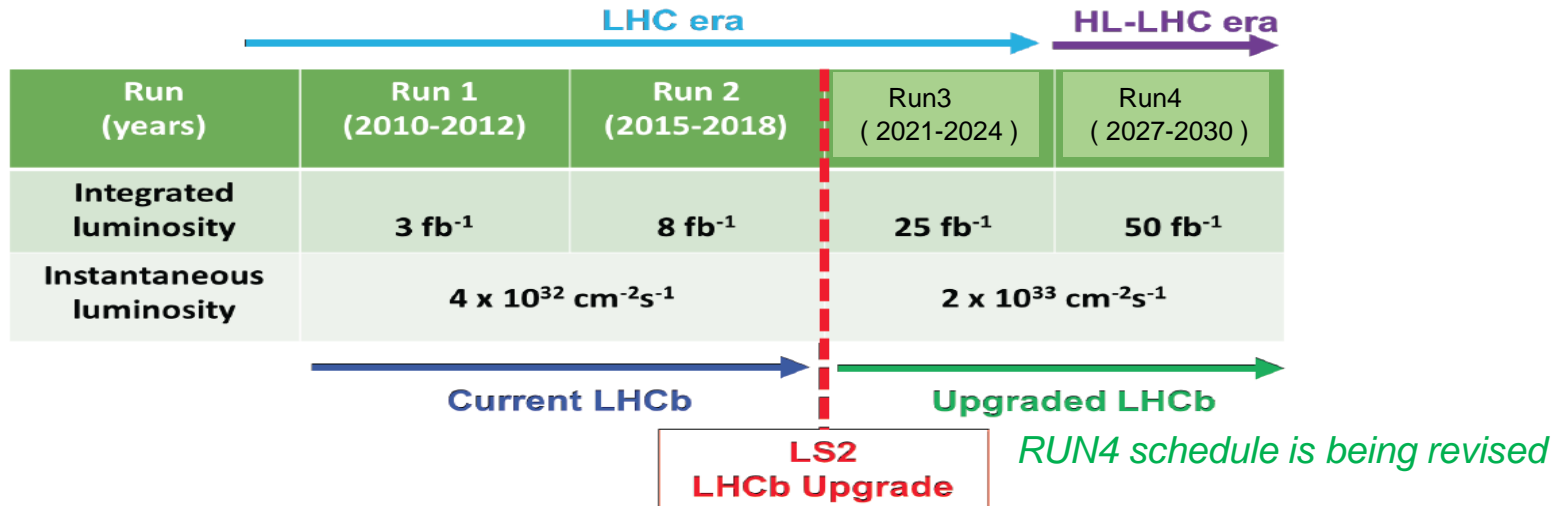
- pp : $\sqrt{s} = 7$ TeV

$2 < \eta < 5$, Forward spectrometer.

Overall acceptance ~ $10 \rightarrow 300$ mrad, Momentum range : 2-100 GeV/c

LHCb collaboration has over 1400 members from 85 institutes in 19 countries, including USA.

LHCb Timeline



T. Ypsilantis



J. Seguinot



LHCb Design and installation:

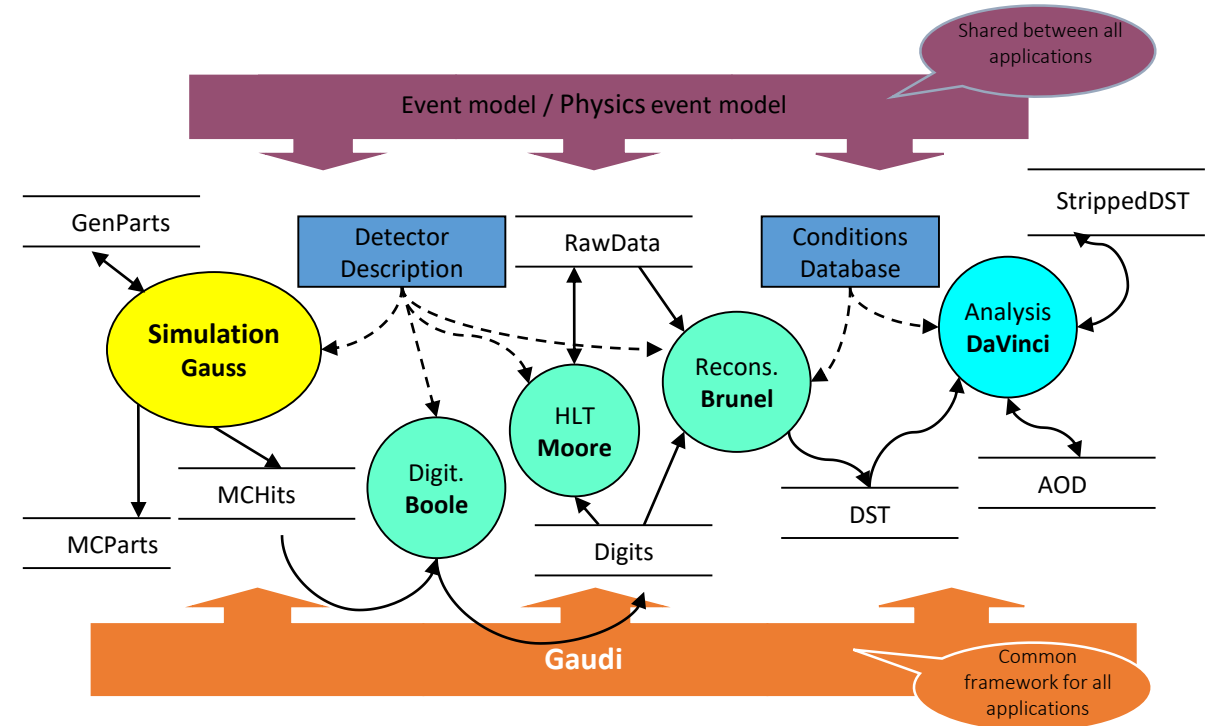
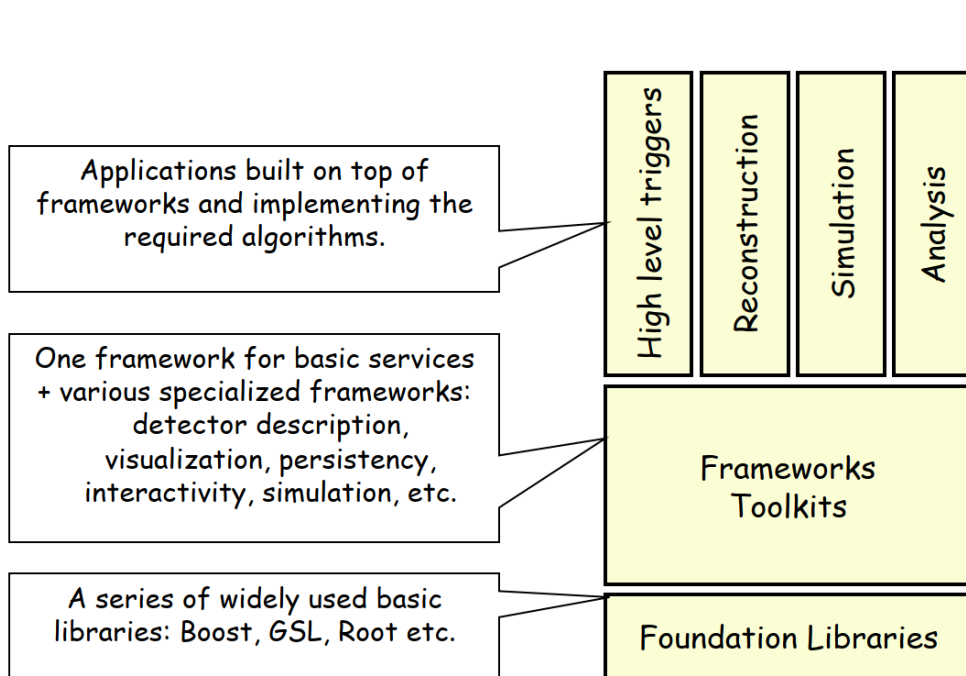
RUN1, 2	Mid 1990's - 2010
RUN3	2010 – 2020
RUN4, 5	2015- ~ 2026

LHCb-RICH software development:

~ 2000	First version of Gaudi Framework in C++, started to be used in LHCb. I started on GEANT4 for LHCb-RICH
2002	First version of GEANT4 simulation for RICH and Calorimeter in LHCb
2004	DC04: First production RUN of LHCb simulations using GEANT4
2010	Arrival of LHCb real data and comparison with simulations. This continued until the end of RUN2.
2011	I started on the simulations to design RICH upgrade for RUN3. This led to PID upgrade TDR in 2013
2015	RICH reconstruction became part of HLT (online) processing.

*T. Ypsilantis, J. Seguinot et al.
 were part of LHCb-RICH
 Nucl. Inst. Meth. A (1994) 30-51
 “Theory of ring imaging Cherenkov counters”*

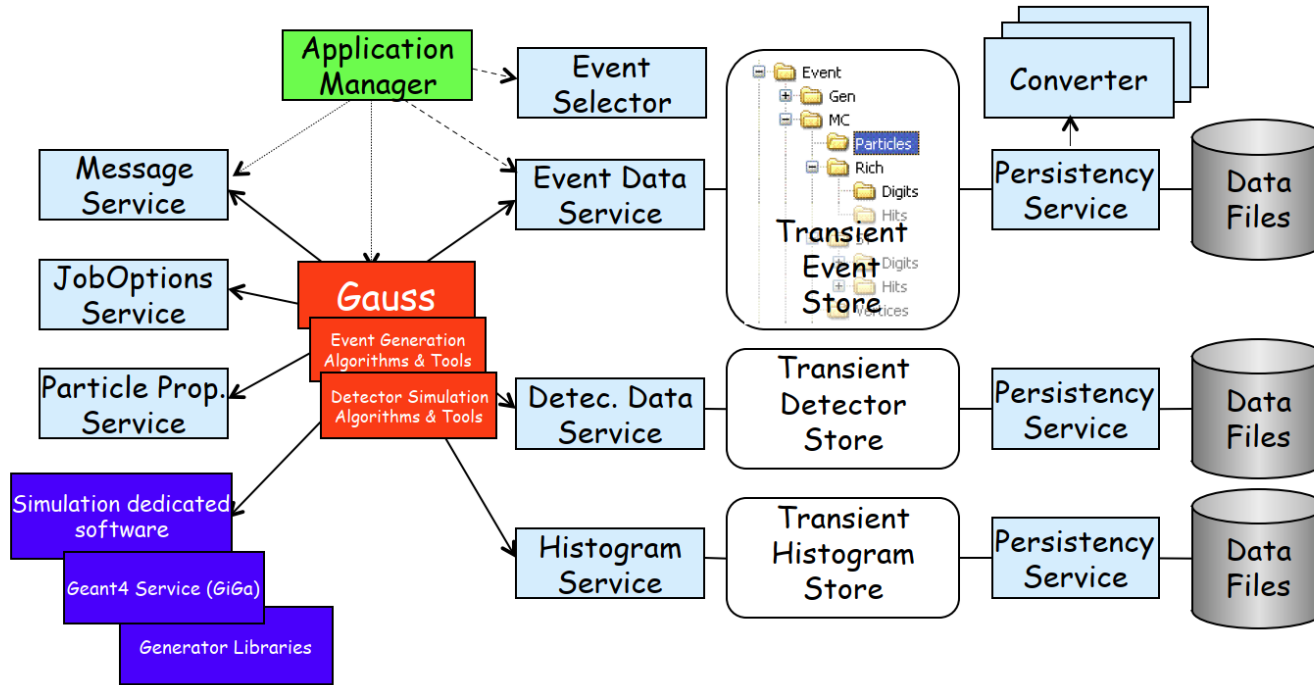
LHCb software framework



- Gaudi was originally developed in LHCb. It is now a general framework. It is also used by other experiments, including ATLAS, Fermi space telescope and Daya Bay.
- It is an open source project and is released by CERN LCG.
- It has user guides, tutorials and other documentation at https://gaudi-framework.readthedocs.io/en/latest/user_guide.html

LHCb software framework

Gaudi Object Diagram



Application software : Set of Gaudi Algorithms and Tools

GAUSS : Event generation and detector simulation

BOOLE: Readout response and digitization of hits

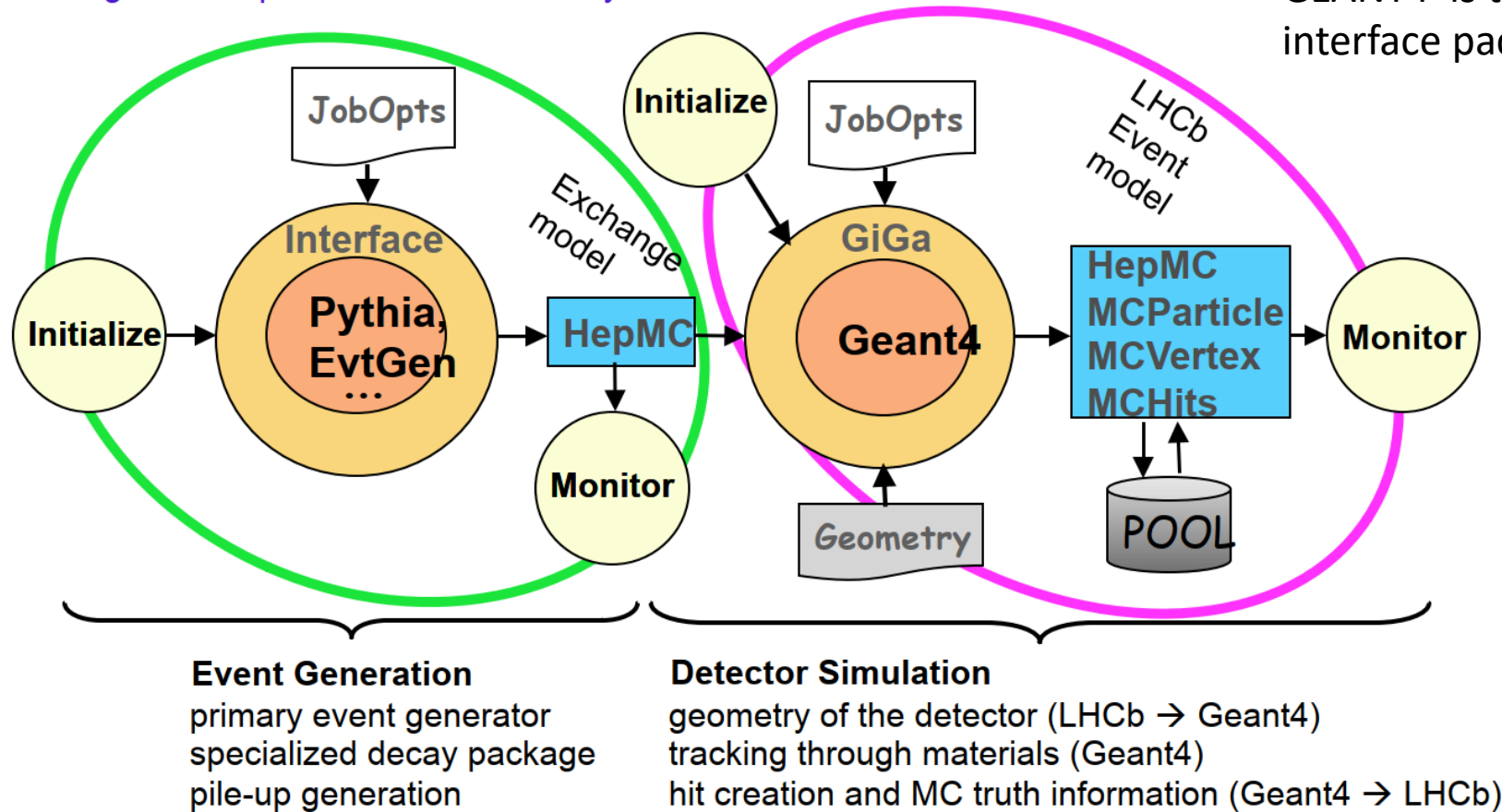
- Separation between “data” and “algorithms”
- Separation between “transient” and “persistent” representations of data
- Well defined component “interfaces”

- Simulation code encapsulated in specific places (*Algorithms, Tools*)

LHCb software framework: GAUSS

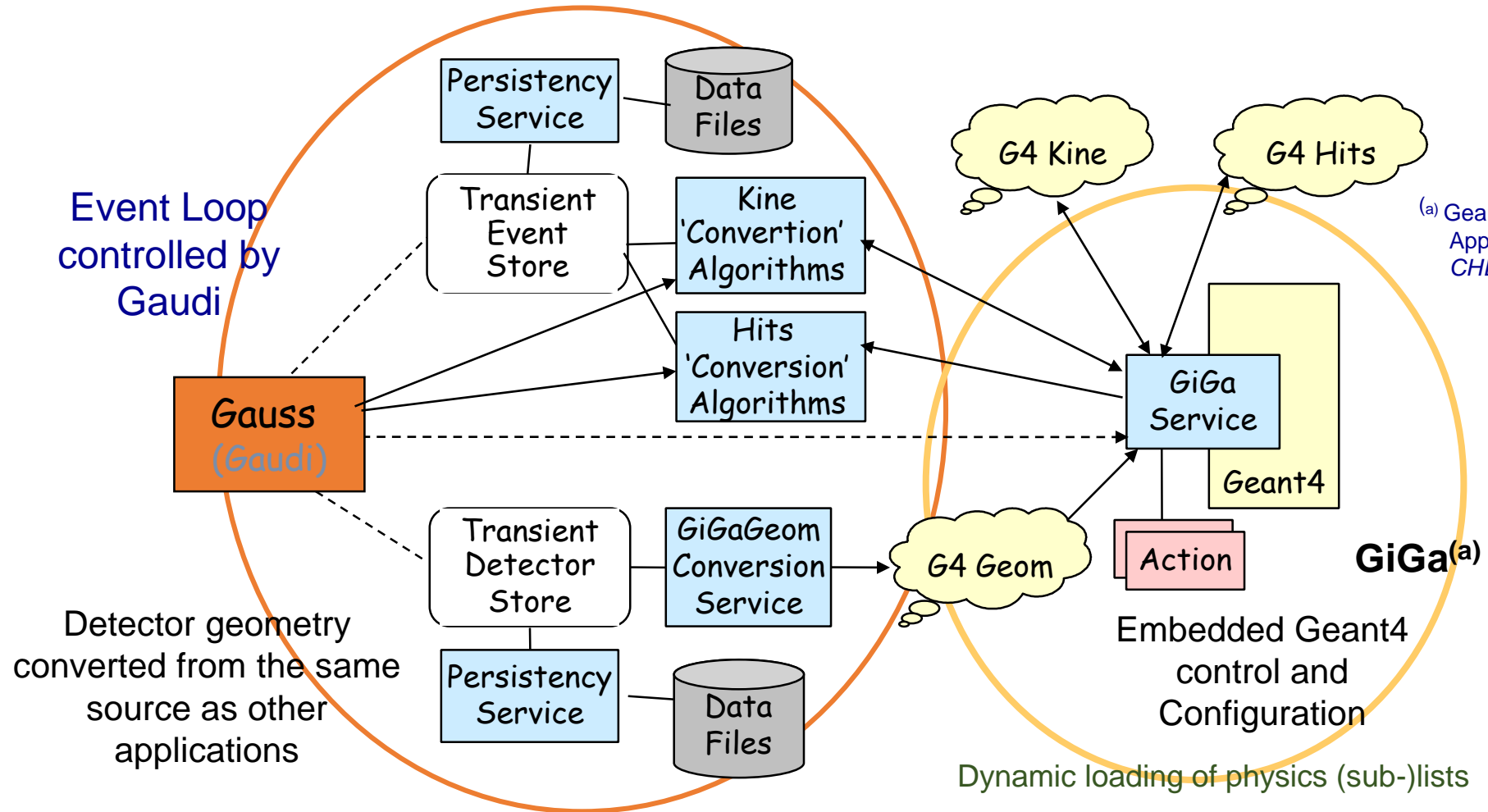
Two INDEPENDENT phases normally run in sequence
but generator phase can and is run by itself

Communication between Gauss and GEANT4 is through an interface package named GiGa



- Information from **G4Hits** created in each detector gets copied to the corresponding **MCHits**.

Encapsulation of GEANT4 in Gauss

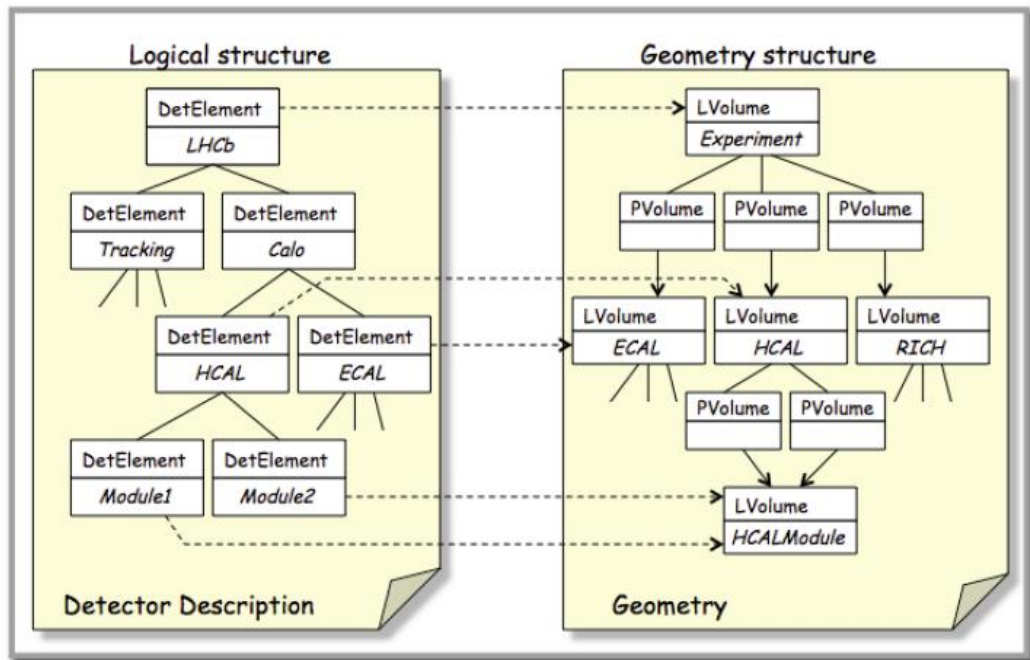


- Transformation of transient objects to/from GEANT4 is easy to configure and control in Gauss.

RICH geometry

- Persistent representation of detector data: The technology choice at the time, was to use XML language to describe the geometry
- The XML DTDs (Document Type Definition) were created for the geometry and material objects .
- Each subdetector, including the RICH used this schema to create the XML geometry database.
- The general structure of LHCb detector description was similar to that of GEANT4 geometry classes. Hence the conversion in the GiGa interface to the GEANT4 geometry and material classes was mostly straightforward.
Few exceptions included, creating detector mis-alignments in simulation, creating optical surfaces etc. Eventually conversion of these were also taken care of.
- The XML files are parsed using an 'IBM XMLparser technology' behind the scenes. The XML database is converted to an SQLITE database for Gaudi to access.
- We also had the concept of 'Detector elements' which were objects which contain various parameters regarding specific volumes. e.g. temperature of a gas, gains of channels, orientation of a plane etc.
- 'Detector elements' were defined in the database with a DTD. They enabled the corresponding C++ classes to access such information directly, whenever needed.

RICH geometry



➤ Different tools were used for geometry overlap testing:

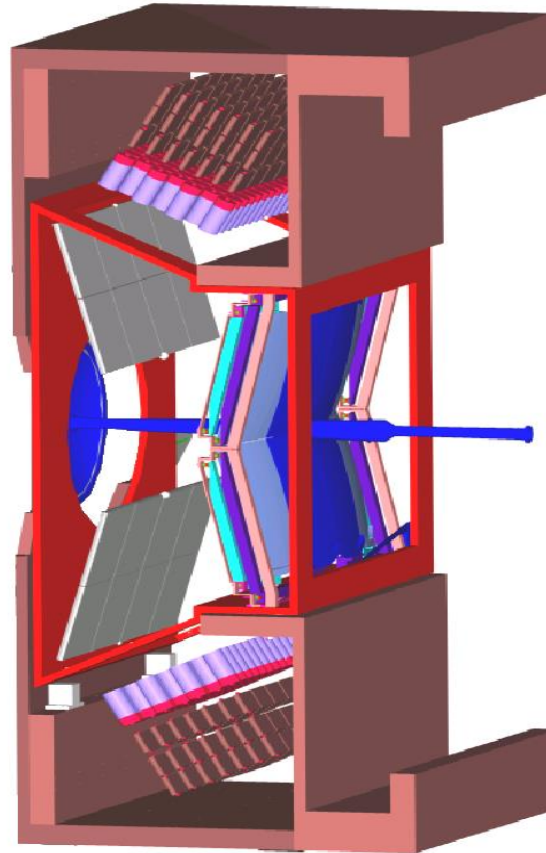
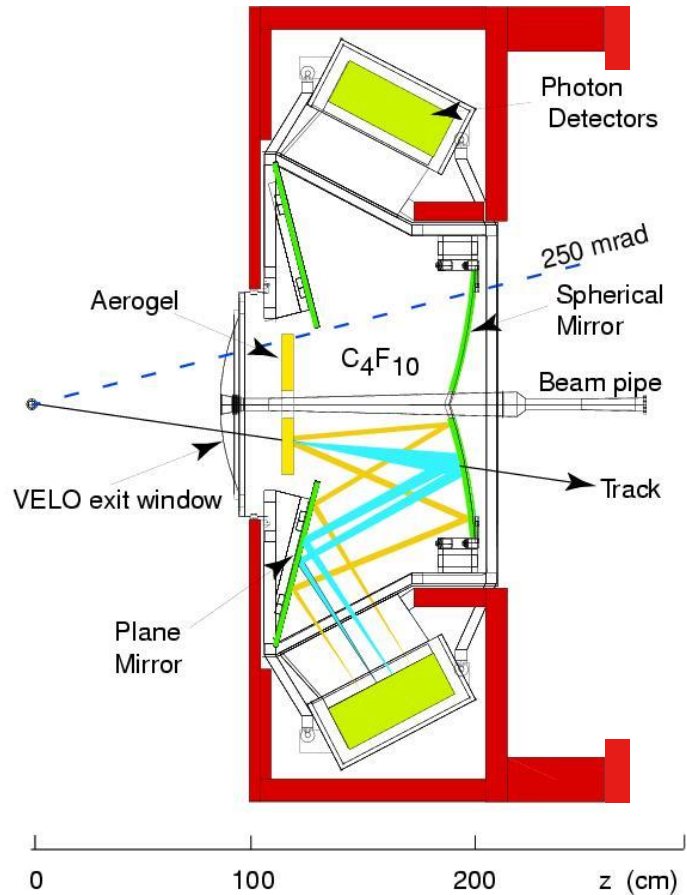
- LHCb graphics application named Panoramix
- GEANT4 with OpenGL, DAWN etc.,
- Shoot lines in different directions using Gaudi transport service and look for intersections at geometry boundaries

➤ “New twist to an old story”, after 20 years:

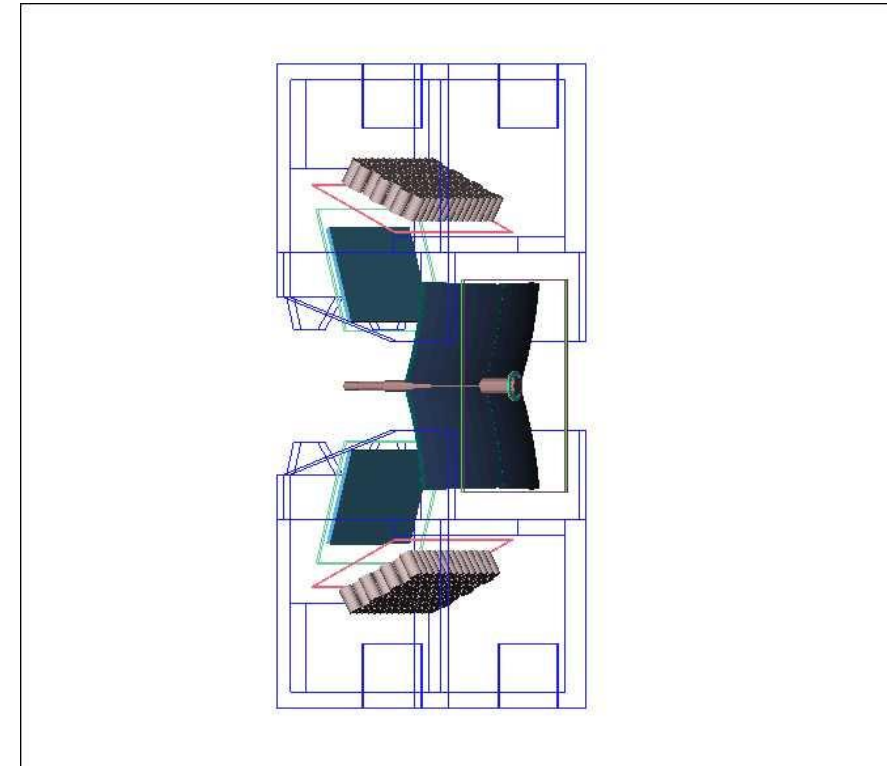
- The XML file structure for Database has some drawbacks
 - Difficult to perform ‘do loops’, for repeated operations in the database files.
 - Difficult to interface to modern ROOT graphics structures
- Solution: Move towards using DD4HEP structure in the coming years.
 - This will also involve modifications to the corresponding detector element implementations.

RICH1 pictures

RICH1 schematic figures

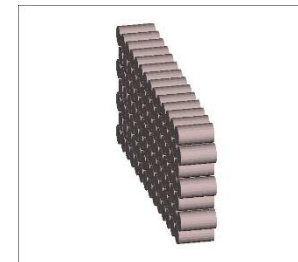


RICH1 in software : RUN2



LHCb: Panoramix graphics used here.
Some components set as invisible or as wireframe for illustration.

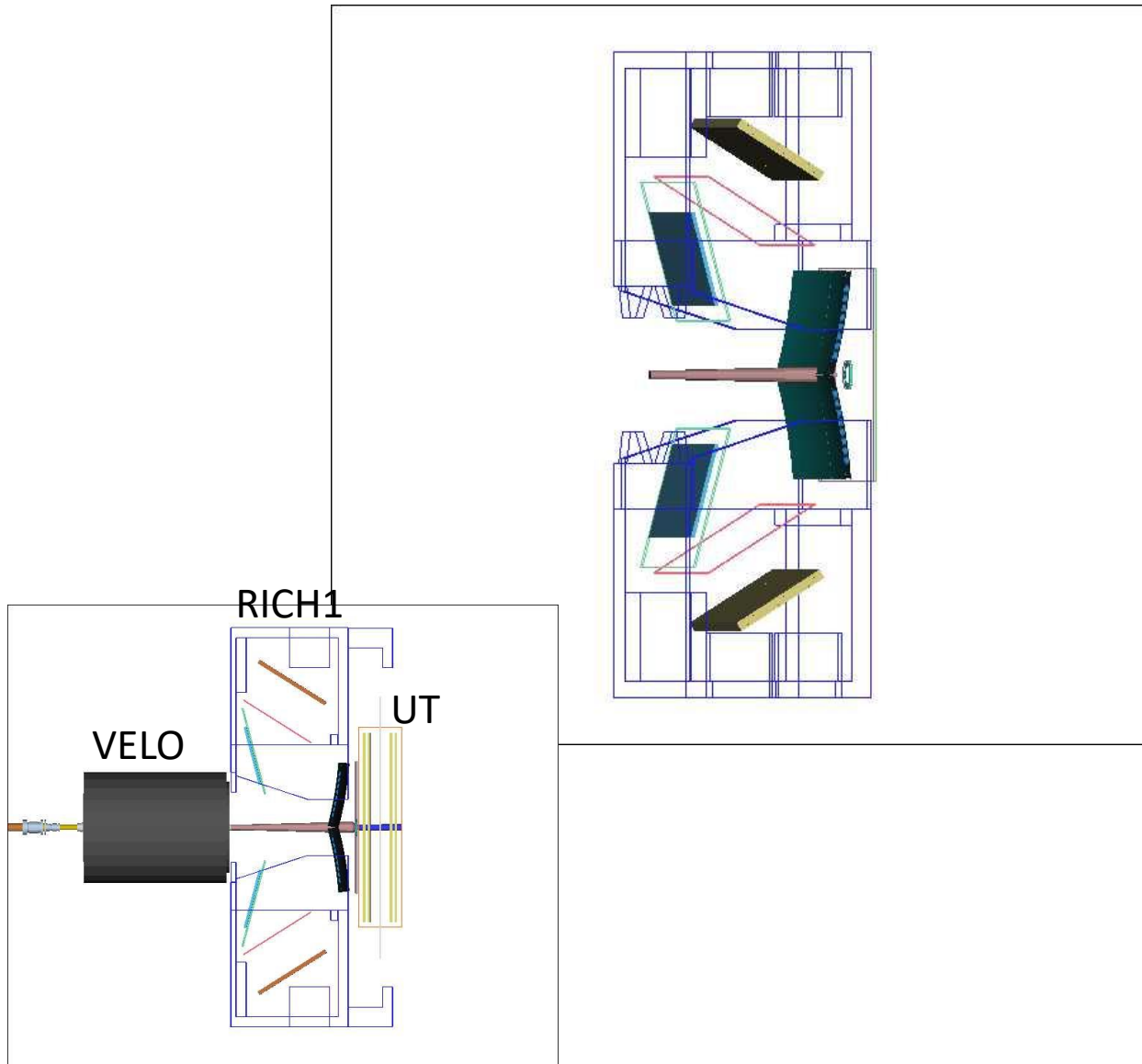
- Aerogel was removed for RUN2
- 4 m³ of C₄F₁₀ , n ~ 1.0014 , acceptance: 25-300 mrad



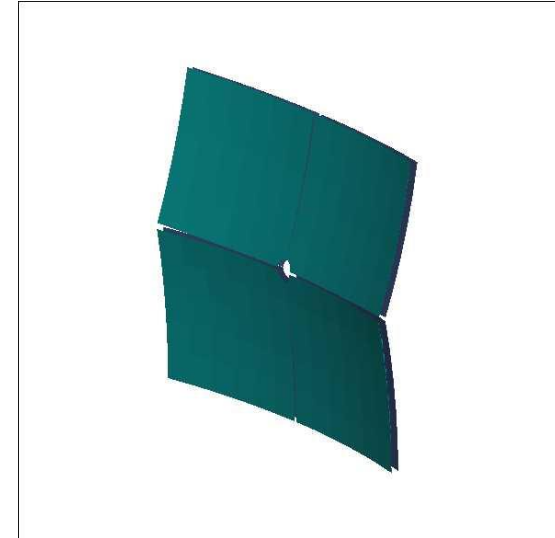
Array of HPDs

RICH geometry in software

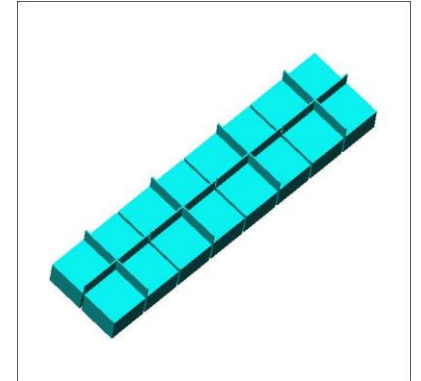
RICH1 in software : RUN3



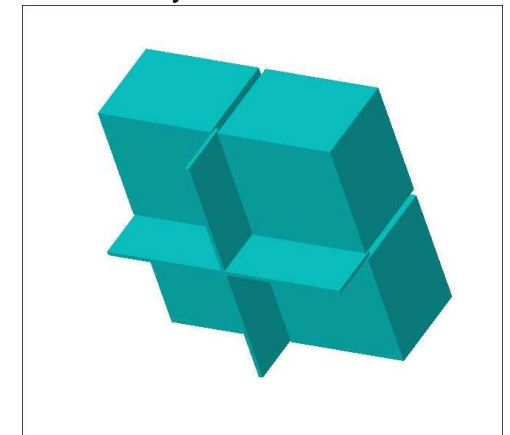
RICH1: spherical mirrors



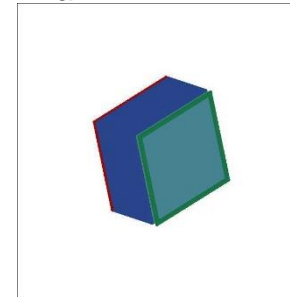
RICH1 MaPMT-Module



Elementary Cell with 4 MaPMTs



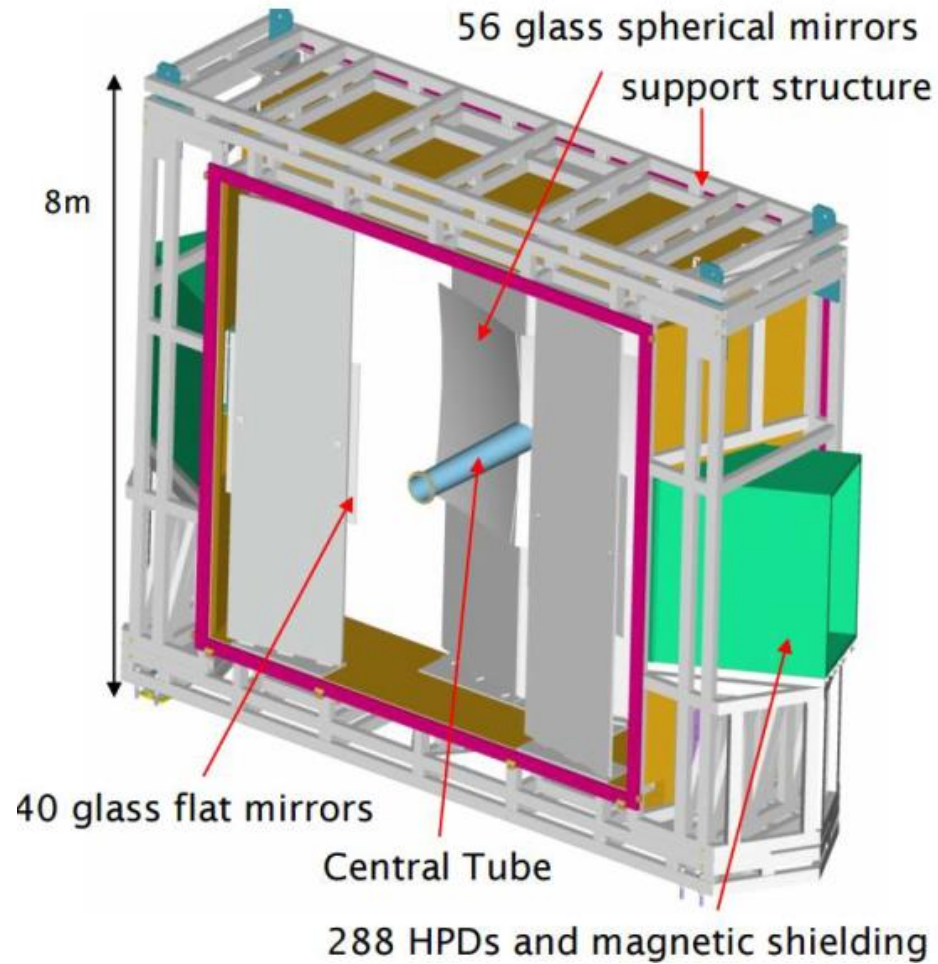
MaPMT



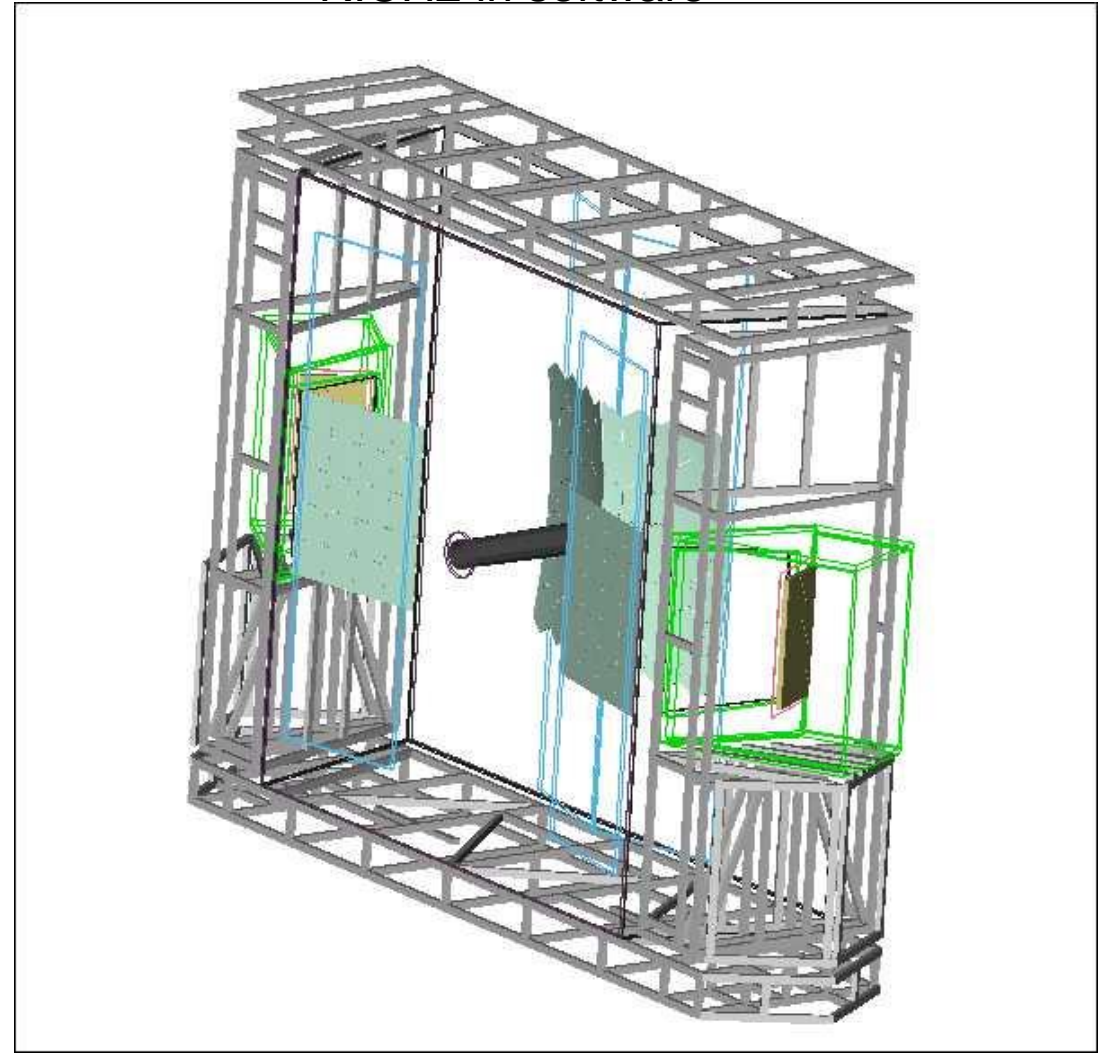
LHCb: Panoramix graphics used here.
Some components are set as invisible or as wireframe in graphics, for illustration.

RICH geometry in software

RICH2 schematic figure

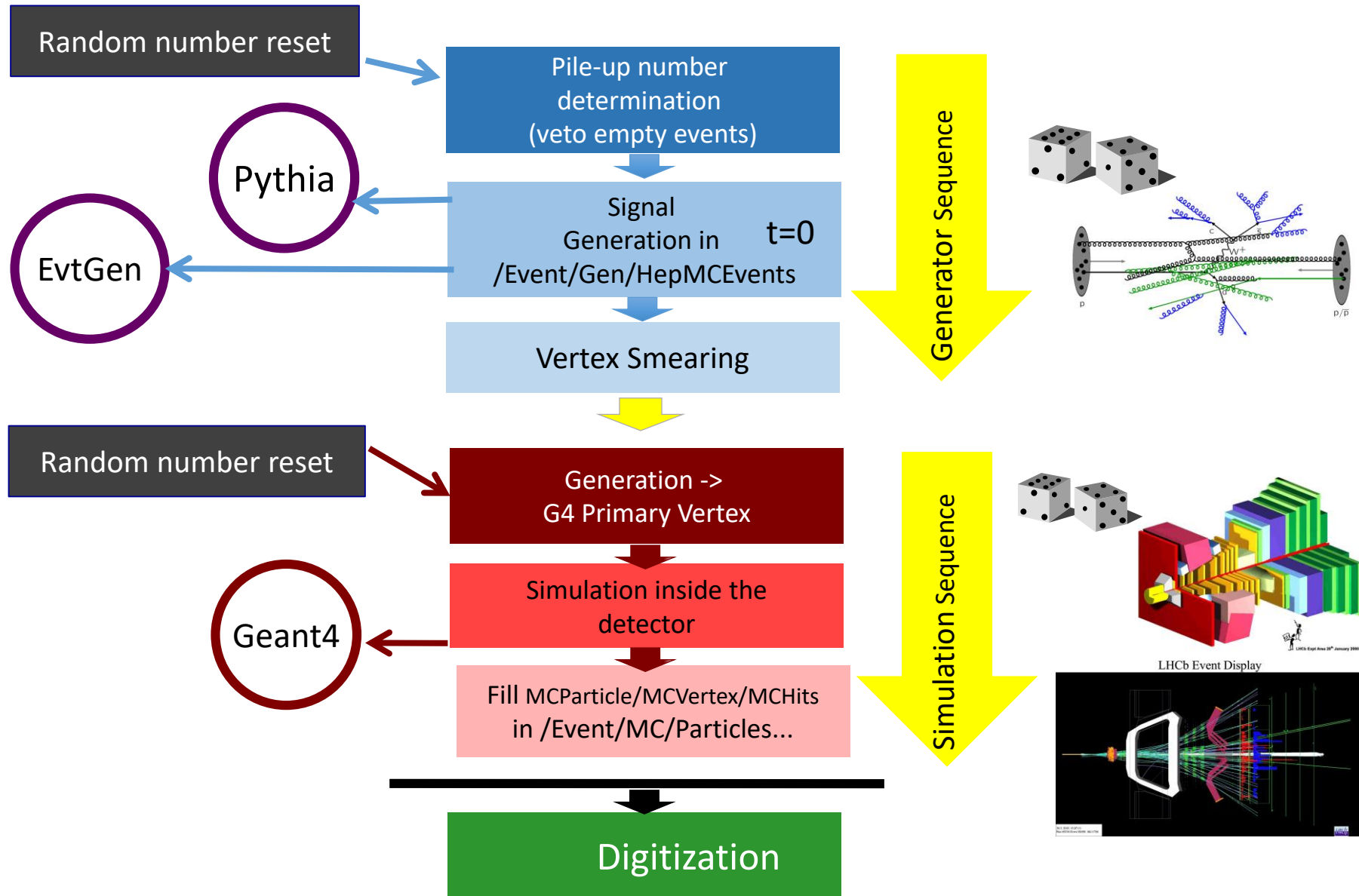


RICH2 in software



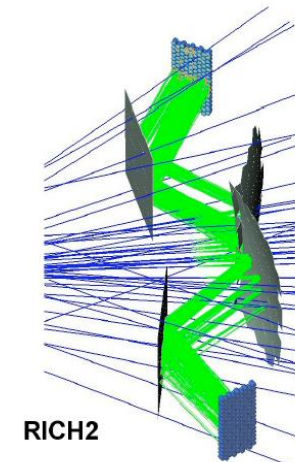
100 m³ CF₄ , n ~ 1.0005, acceptance: 15-120 mrad

Simulation sequence



RICH Simulation specific actions

- For creating the sensitive detector where hits are created, one just sets an attribute for the corresponding volume in the XML database. This is then automatically transferred to the GEANT4 through the various interfaces.
- For the general LHCb simulation, we did not make use of the ROGeometry in GEANT4 since the simulation of the readout was done in the Boole application separately. But it was used in our test beam applications.
- Once the structure is in place, the RICH specific software included some GEANT4 user actions activated through the GiGa interface for optimizing the CPU time and for monitoring. The monitoring of performance was done extensively.
- We also had some RICH specific physics processes.

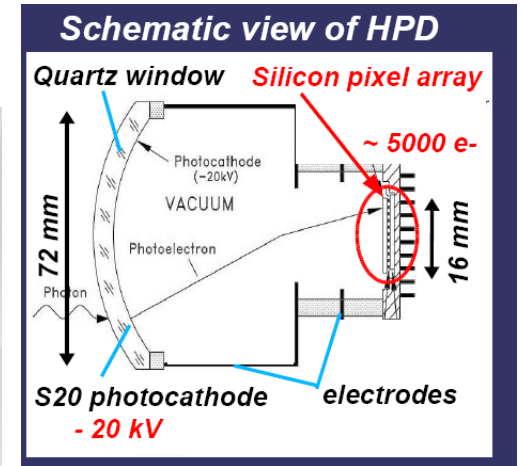


RICH physics processes

- For electromagnetic and hadronic processes, LHCb uses the standard physics lists from GEANT4. The RICH system in LHCb is part of that.
- For Cherenkov photon production and transport the following processes are important:
G4Cerenkov, G4Scintillation, G4OpBoundaryProcess, G4OpRayleigh
- For LHCb-RICH use case, we implemented some modifications for these processes
Examples:
 - Keep track of the Cherenkov angle and Scintillation angle at photon production, for tests during the reconstruction. This required attaching 'user info' to photon tracks.
 - Avoid 'almost infinite loops' from total internal reflection of Cherenkov photons, for example in a rectangular quartz plate.
 - Avoid the possibility of 'almost infinite' number of Rayleigh scattering steps.
 - Perform CPU time optimization, by having different wavelength ranges for Cherenkov production and transport. For example, when there are multiple radiators, one may restrict the wavelength range of photons produced in a radiator for reducing chromatic error. But one may allow the photons produced elsewhere with a wavelength outside this range, to travel through that radiator.

RICH physics processes

- For RUN1 and RUN2 we used HPDs (Hybrid photo diode). The photoelectrons are produced based on the tables of QE versus photon energy. The standard GEANT4 process did not use such an input. Hence we created a special process for this.



- Similarly, for the energy loss of photoelectron at the silicon anode, we created a special process.
- Further details on the RICH simulation from LHCb:
“Simulation of LHCb RICH detectors using GEANT4”, S.Easo et.al., IEEE Trans. Nucl. Sci. 52 (2005) 1665.
- Verification of physics processes:
 - The RICH physics processes were used in the RICH test beam simulation program.
 - The results from simulation were compared with the results from analyzing the real data from different test beams where the RICH prototypes were tested.
 - These verifications were done in terms of Cherenkov angle resolutions and photon yield.
 - This procedure helped to validate the physics processes well before the LHCb data arrived.
 - In the old days, a version of this test beam program was available as a ‘GEANT4 example’ for some years.
Later this was replaced by other examples for optical processes.

Examples of Testbeam results

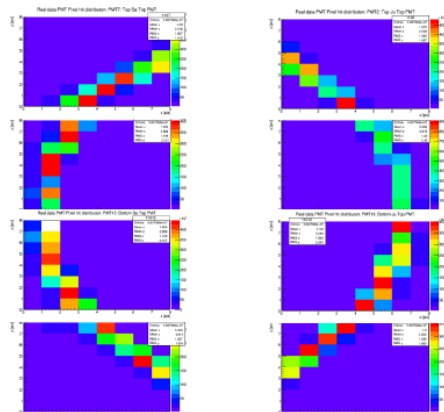
- Various aspects of the system were verified in different testbeams which used different geometries.
Three of them are listed here:

(a) “Performance of the LHCb RICH photo-detectors and readout in a system test using charged particles from a 25 ns-structured beam”, M.adinolfi et.al. Nucl. Inst. Meth. A 603 (2009) 287-293

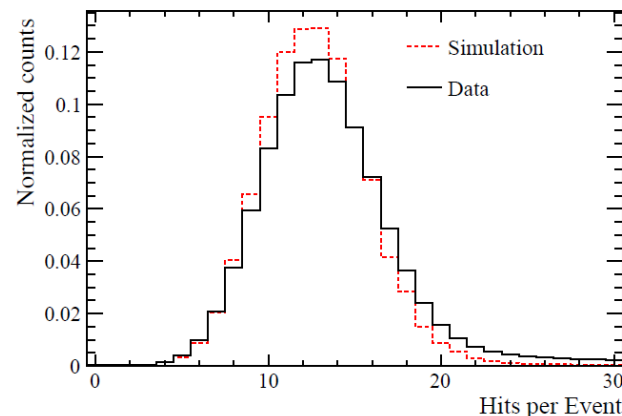
(b) “Performance of the LHCb RICH photodetectors in a charged particle beam”, M.Adinolfi et.al.,Nucl. Inst. Meth. A 574 (2007) 39.

(c) “Test of the photon detection system for the LHCb RICH Upgrade in a charged particle beam”
M.K. Baszczyk et.al., JINST 12, P010112 (2017)

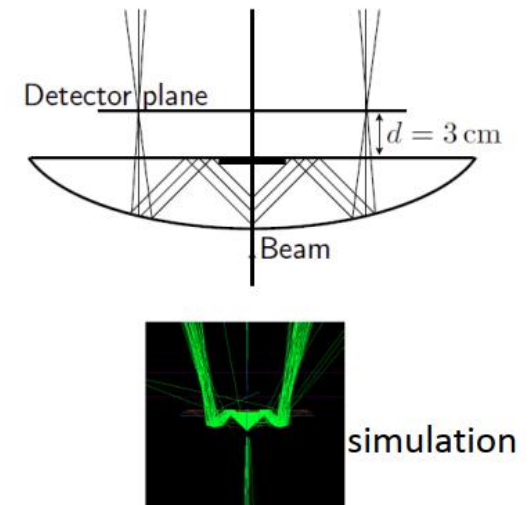
Test beam in 2015



Cherenkov ring in 2015 test beam data



Number of hits per track
in 2015 test beam



LHCb-RICH performance from simulations: RUN2 and RUN3

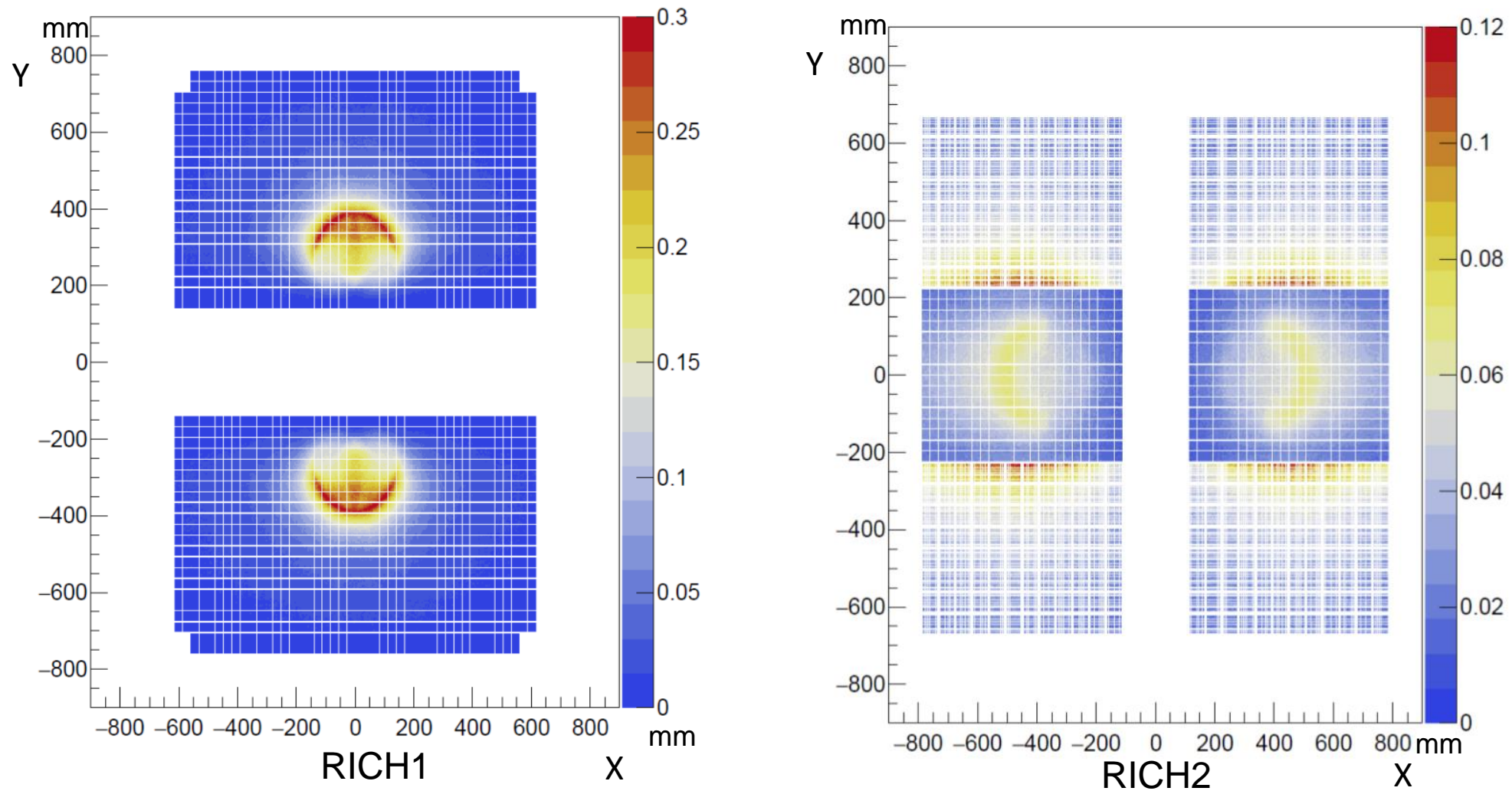
Examples of Cherenkov Angle Resolutions and yields

Resolution (in mrad)	RICH1-2015 HPD, C ₄ F ₁₀	RICH1-upgrade MaPMT, C ₄ F ₁₀	RICH2-2015 HPD, CF ₄	RICH2-upgrade MaPMT, CF ₄
Chromatic	0.84	0.58	0.48	0.31
Pixel	0.60, PSF=0.86	0.44	0.19 PSF=0.29	0.19
Emission point	0.76	0.37	0.27	0.27
Overall	1.60	0.78	0.65	0.45
Overall+Track	1.65	0.88	0.76	0.60
Yield	32	42	24	22

- Yield: Mean Number of RICH hits per saturated track (Beta ~1).

Ref: “*Overview of LHCb RICH Upgrade*”,
S.Easo, Nucl.Inst.Meth. A 876 (2017) 160-163

RICH data from simulation

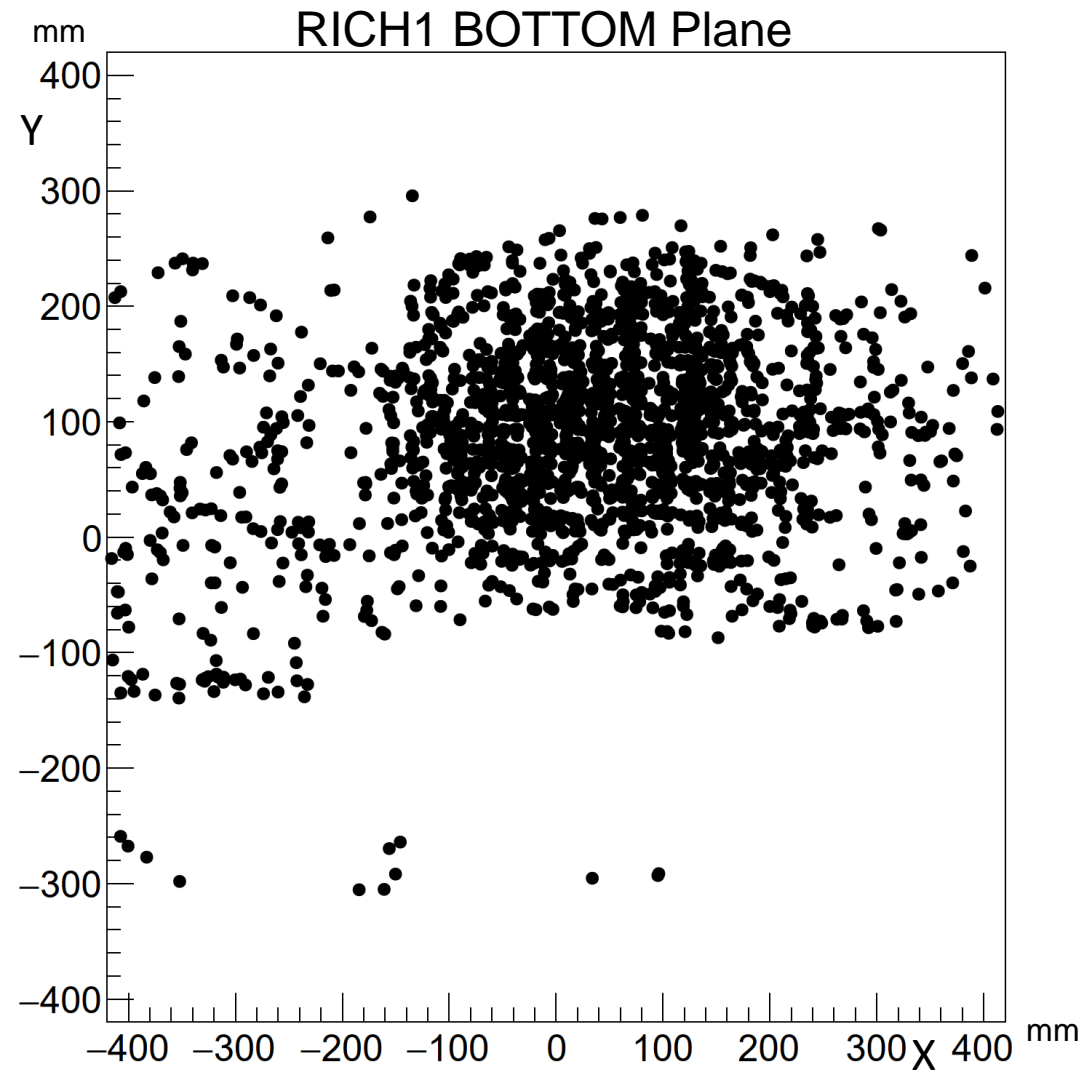
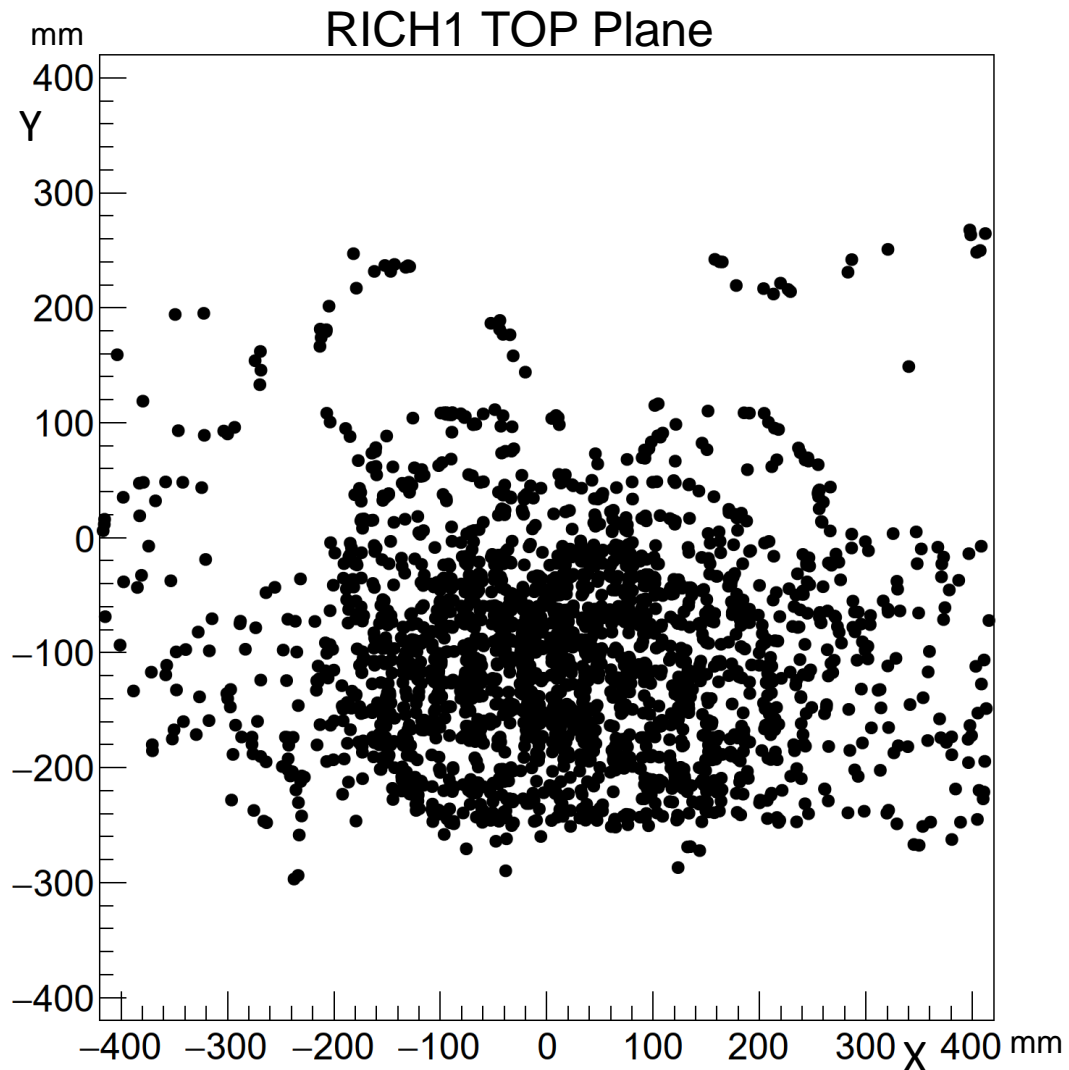


Y vs X of the hit coordinates on the detector planes, from simulations.

This is normalized to the number of events.

- B-meson signal events, RUN3 simulations

RICH data from simulation



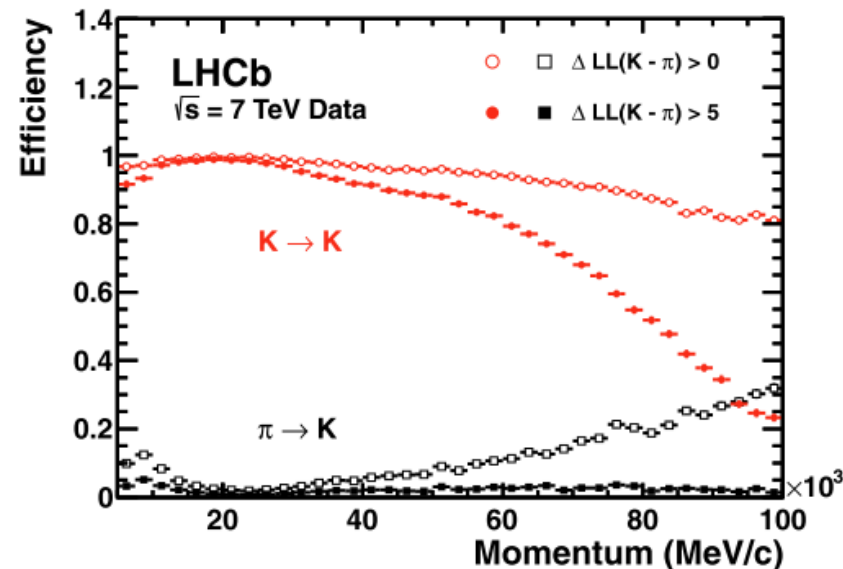
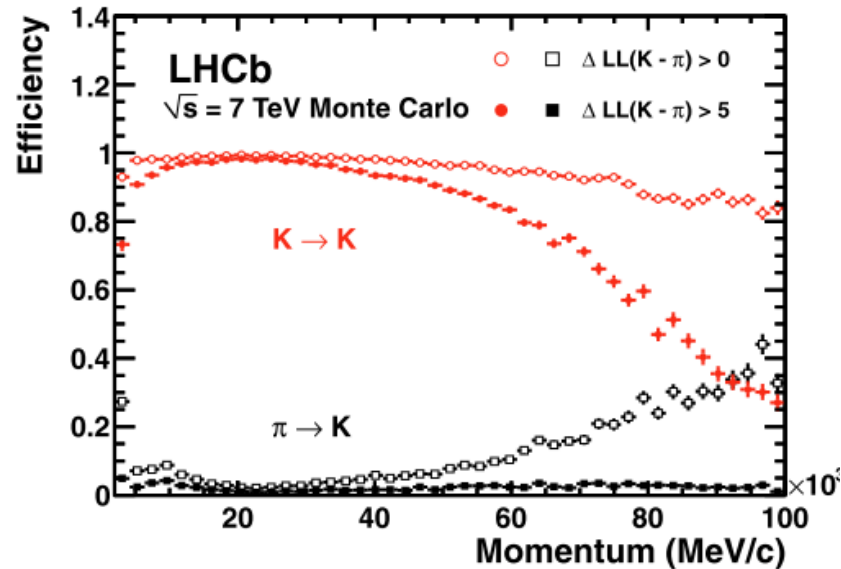
- Y vs. X of the hit coordinates in the central region of the detector plane in a single event in RICH1.
- B-meson signal events, RUN3 simulations

- *Number of tracks ~ 215*
- *Number of RICH1 hits ~ 5300*

RICH data reconstruction

- The reconstruction program is same for simulated and real data.
- It is implemented within the Brunel application of the Gaudi software framework.
- LHCb uses a log-likelihood algorithm for particle identification.

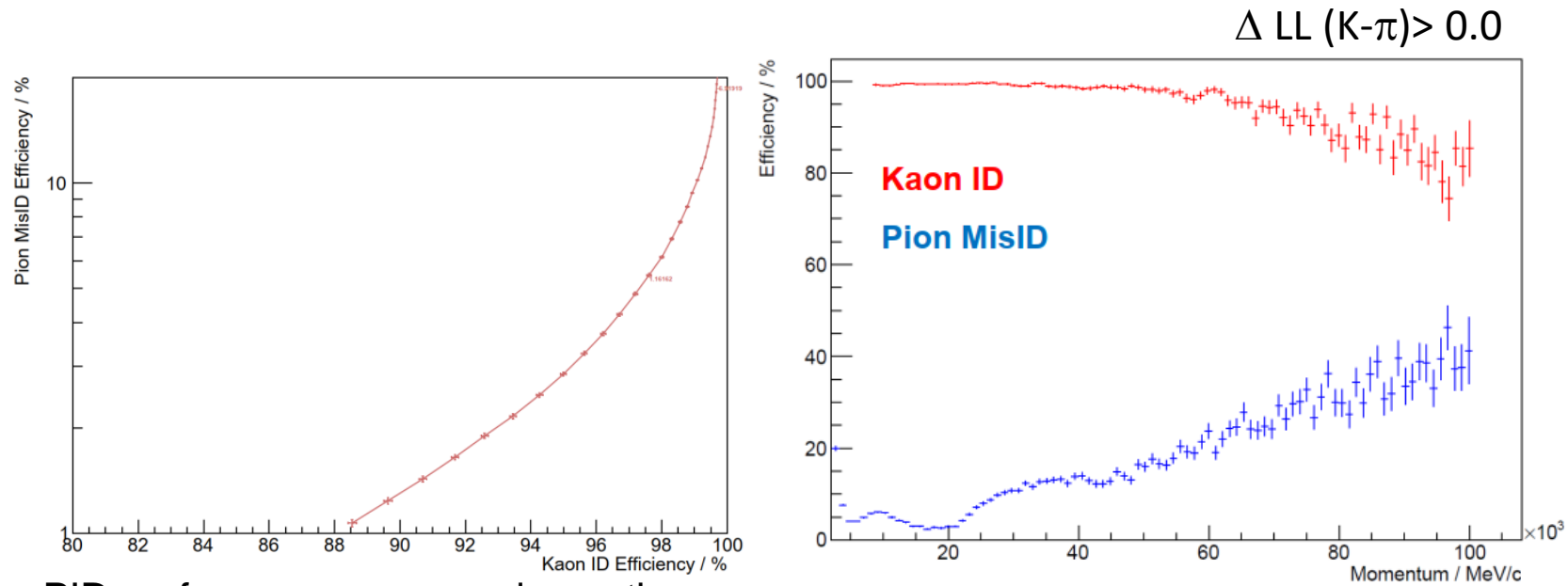
Ref: “*RICH pattern recognition for LHCb*”, R.Forty: Nucl. Inst. Meth. A 433 (1999) 257-261



- Example of PID (particle identification) performances from simulation and real data, from 2011 configuration in RUN1.
Ref: “*Performance of LHCb RICH detector at the LHC*”, M.Adinolfi et.al., Eur. Phys. J.C. (2013) 73 : 2431

RICH data reconstruction

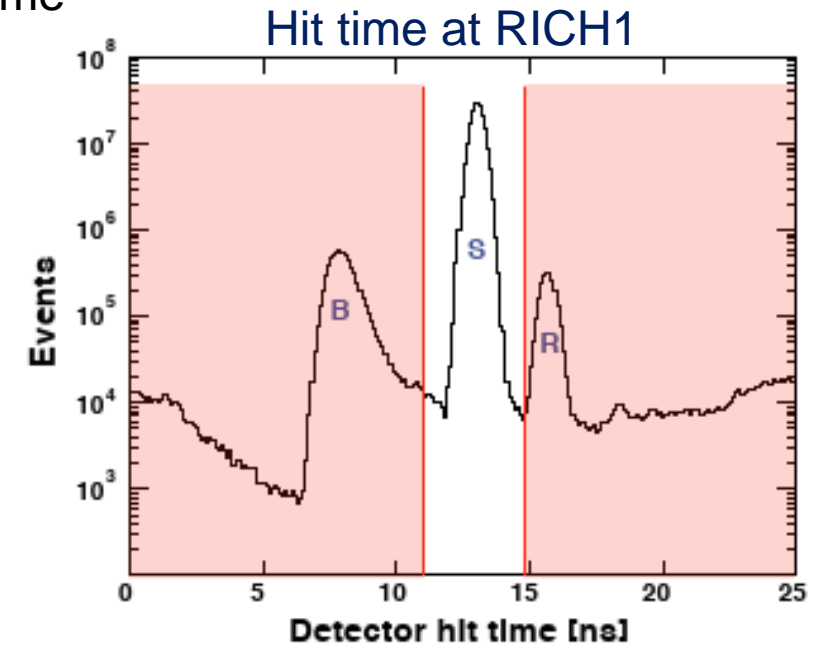
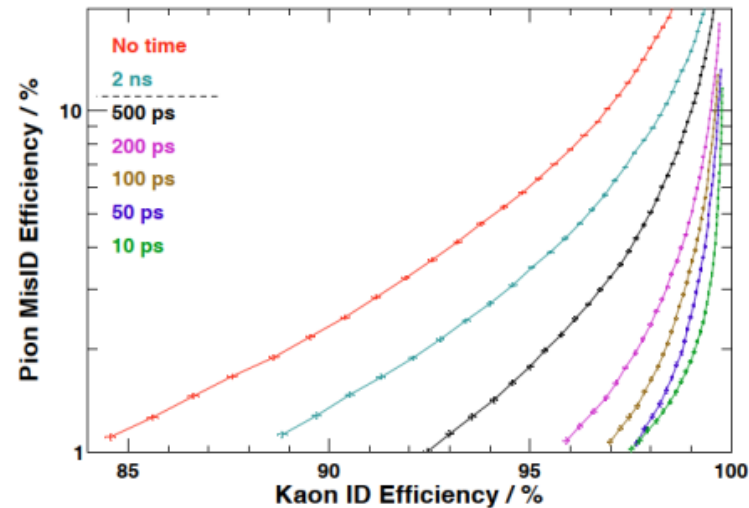
- An example of PID performance expected for RUN3.



PID performance averaged over the momentum range 3-100 GeV/c
Each point corresponds to a different cut on $\Delta LL (K-\pi)$

RICH Simulation: Using fast timing

- In order to reduce the backgrounds and improve the PID, using the time gating of signals in hardware and software is useful.
- For the designing the RICH for RUN3 and beyond, we are using the time information of the hits, created in GEANT4.
- We also simulate the processing times needed by the photon detectors TTS (Transit Time Spread) and the readout.
- An example of PID improvement with timing:



Prospects beyond the horizon



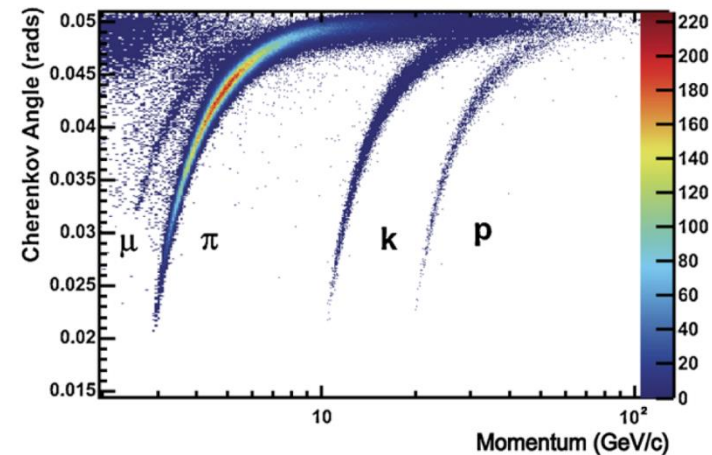
- Updates for using DD4HEP structure for geometry description
- Ray tracing of photons in simulation:
 - The LHCb simulations are run over billions of events in production runs. Hence we try to optimize the CPU time wherever possible.
 - A major part of the CPU time taken for the RICH is from the transport of the optical photons in GEANT4.
 - Recently, the Daya Bay collaboration has successfully used GPUs to use 'OptiX' for the ray tracing of Cherenkov photons and achieved large gains in CPU time.
 - They interfaced 'OptiX' to GEANT4. Here the Cherenkov photon production and transport is done by 'OptiX' and the hits created are copied back into the GEANT4 structure. We are also in the process of moving towards using this. GEANT4 is also looking into this.
- In the distant future, there is also GEANTV, which aims to have vectorized workflows for improving CPU time.

SUMMARY

- The Gaudi framework is used for the software in LHCb, including the simulations of the RICH system.
- The detector simulation is performed using GEANT4.
- During the RICH simulation, Cherenkov photons are produced and transported through the optical geometry.
- The RICH simulation has couple of custom made physics processes and various measures are taken to optimize the CPU time.
- Simulation studies are underway for using the 'RICH hit time' to improve the PID performance in the future.
- There are also other developments foreseen for RICH simulation, making use of the modern GPUs.

SUMMARY

- I wish to acknowledge the support from my colleagues in LHCb and especially from colleagues in the LHCb-RICH group.
- I also received lot of useful help from our colleagues in the GEANT4 collaboration, especially in the early days.
- I hope this has some useful information for RICH simulation studies at the EIC detector.
- Wish you all the success in that project 😊



*EPJC (2013) ,73:2431
From RICH1 gas radiator*