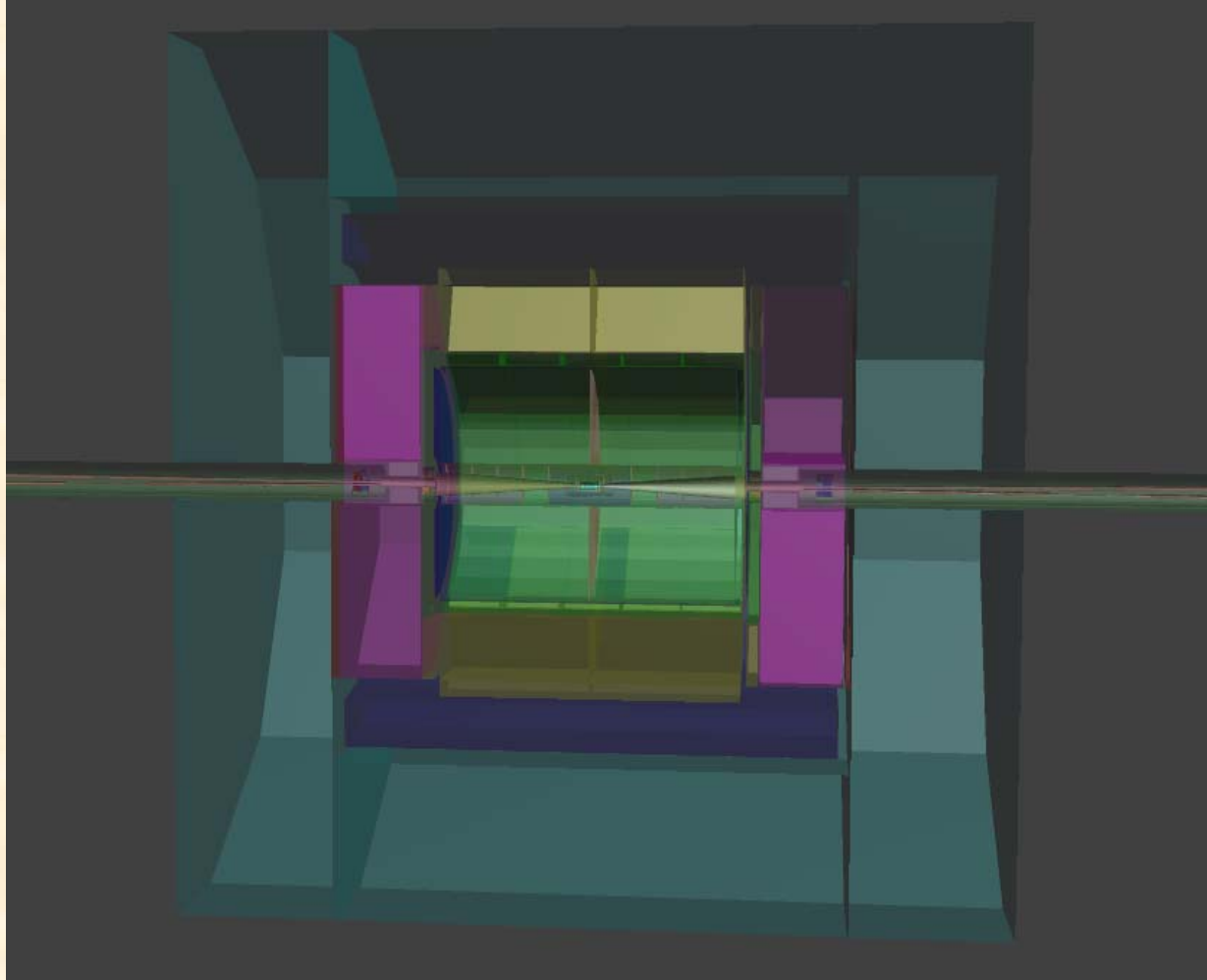


Mokka - Overview and Plans



ILD Software Meeting 2010
January 27th – Ecole polytechnique, Palaiseau
Paulo Mora de Freitas - L.L.R. – Ecole polytechnique

Mokka - Overview

- Stable : any big revolution or reengineering at Kernel level last year
- Features for detector optimization
 - Detector editing on the fly
 - Scaling geometry on the fly
- Features for detector studies
 - Users are able to plug user code at begin/end of: run, event, tracking and stepping (Plug-in user actions)

Mokka - plans

- Plans to improve:
 - Geometry: dumping automatically the model parameters into GEAR file
 - Documentation: dumping automatically for each new model in DB a detailed set of pdf 3D describing each detector device
 - Performance: testing a simple fork strategy
 - all projects started 😊, any finished 😞
 - ASAP

Physics Lists -1

- LOI studies have used LCPhys created by Dennis Wright (SLAC), which says that:
 - “- The linear collider physics list I provided some time ago (LCPhys) is in fact no longer maintained. In its place I have been recommending the Geant4 reference physics list QGSP_BERT_HP.” (January 2010)
- Calice people have used LHEP, QGSP_BERT or Q6SC_CHIPS depending on the prototype and/or the test beam year

Physics Lists -2

- A choice has to be done before starting MC mass production (before end 2011?)
- Ideally a common choice for all detector concepts, to insure consistency
- Proposal:
 - To keep QGSP_BERT_HP as the Mokka default physics list and to advice people to use it;
 - Topic to be discussed by the LC community:
LCWS? 2010 or 2011?

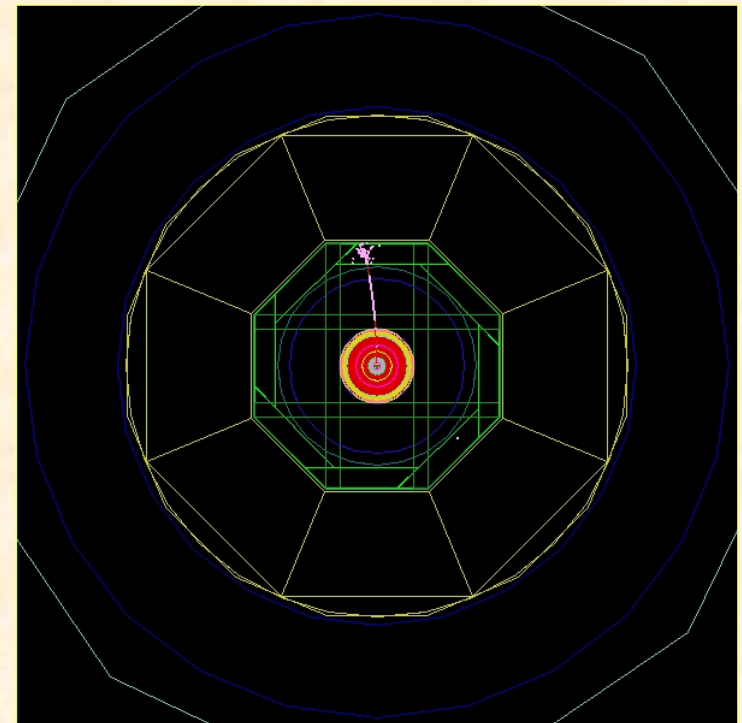
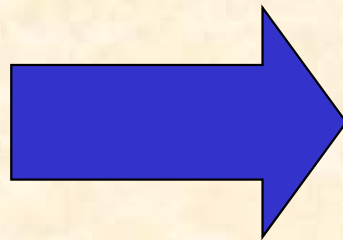
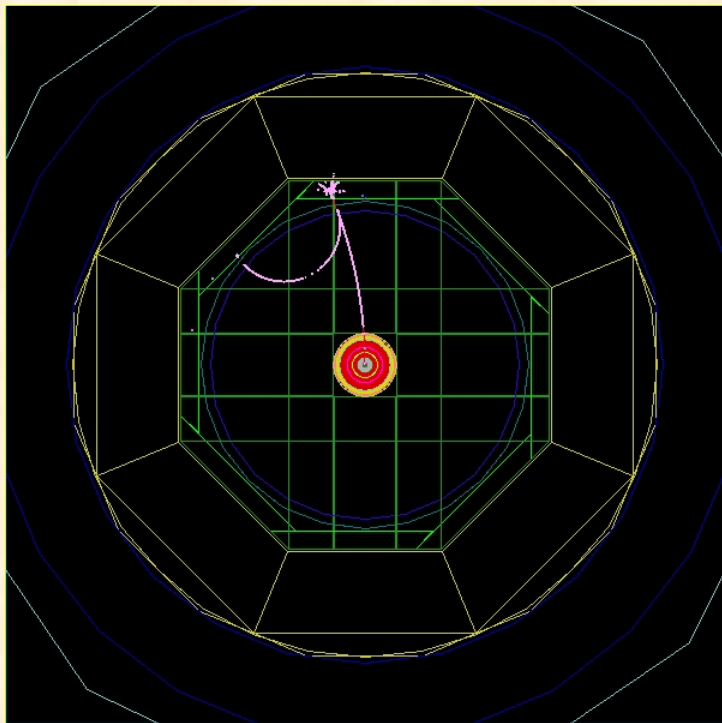
Conclusions

- Mokka *Since 1999* is stable and has been useful for ILD and Calice studies
- For the next two years:
 - Developments can / has to be done to improve Mokka as the simulation tool for ILD
 - Users requirements are welcome
- Physics list is almost a LC issue, Mokka can just implement the community choice

BACKUPS

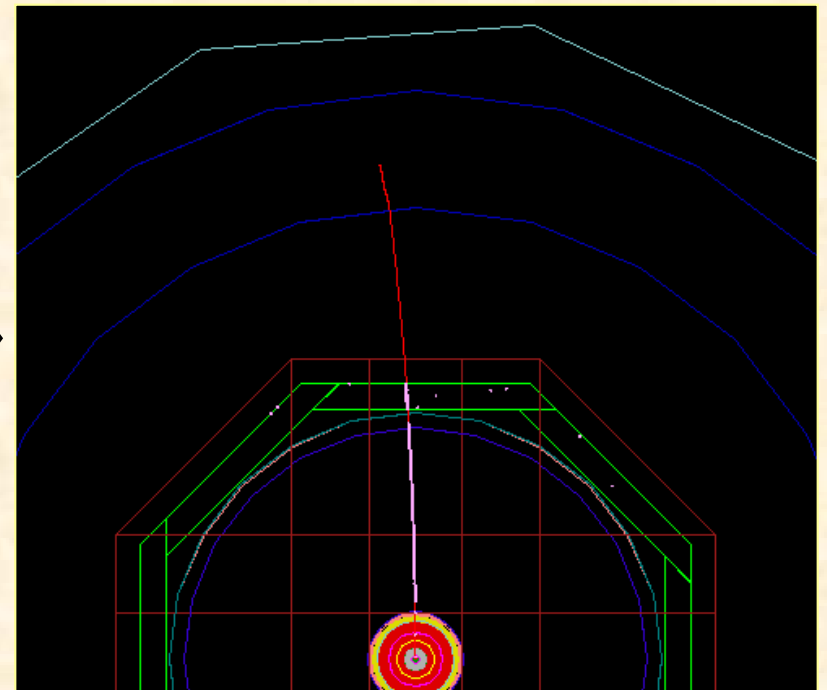
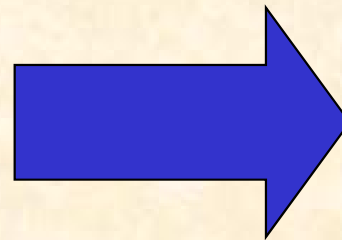
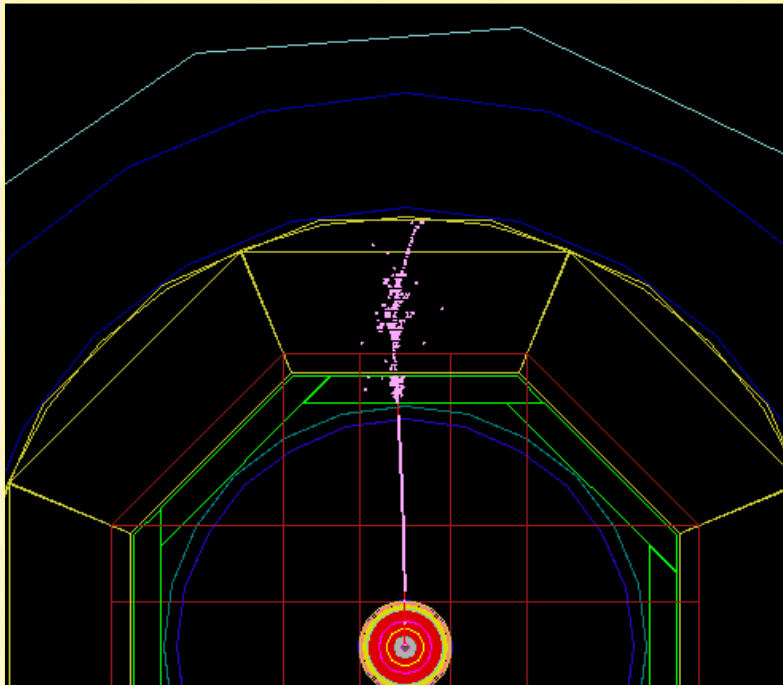
Some good features... (1)

- “Scaling”, the user is able to modify the model's main parameters at launch time, ex :
 - `/Mokka/init/globalModelParameter TPC_outer_radius 800`



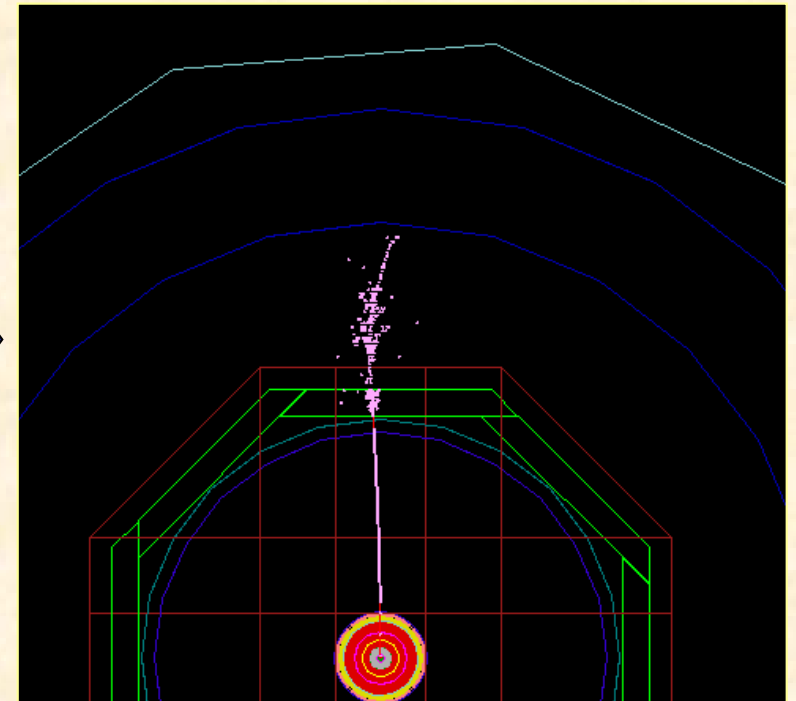
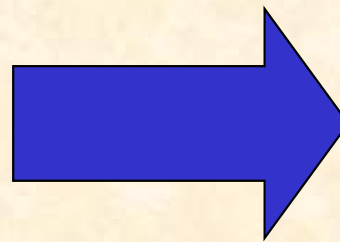
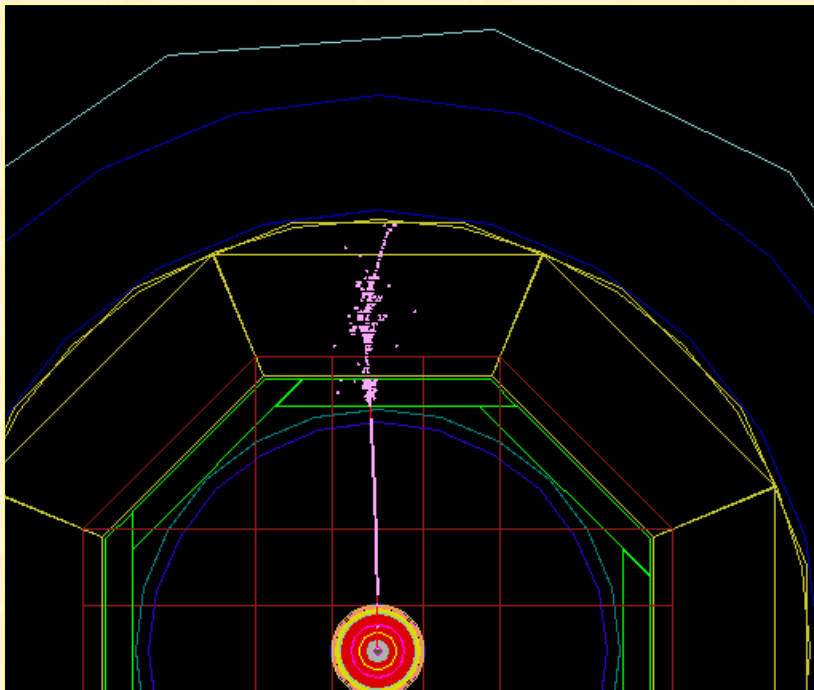
Some good features...(2)

- “Cooking”, the user is able to modify the model ingredients at launch time, ex :
 - `/Mokka/init/EditGeometry/rmSubDetector SHcal01`



Some good features...(3)

- “Visioning models”, the user is able to interactively modify the model rendering, ex :
 - Idle> /Mokka/Visu/Detector/Visibility hcal false

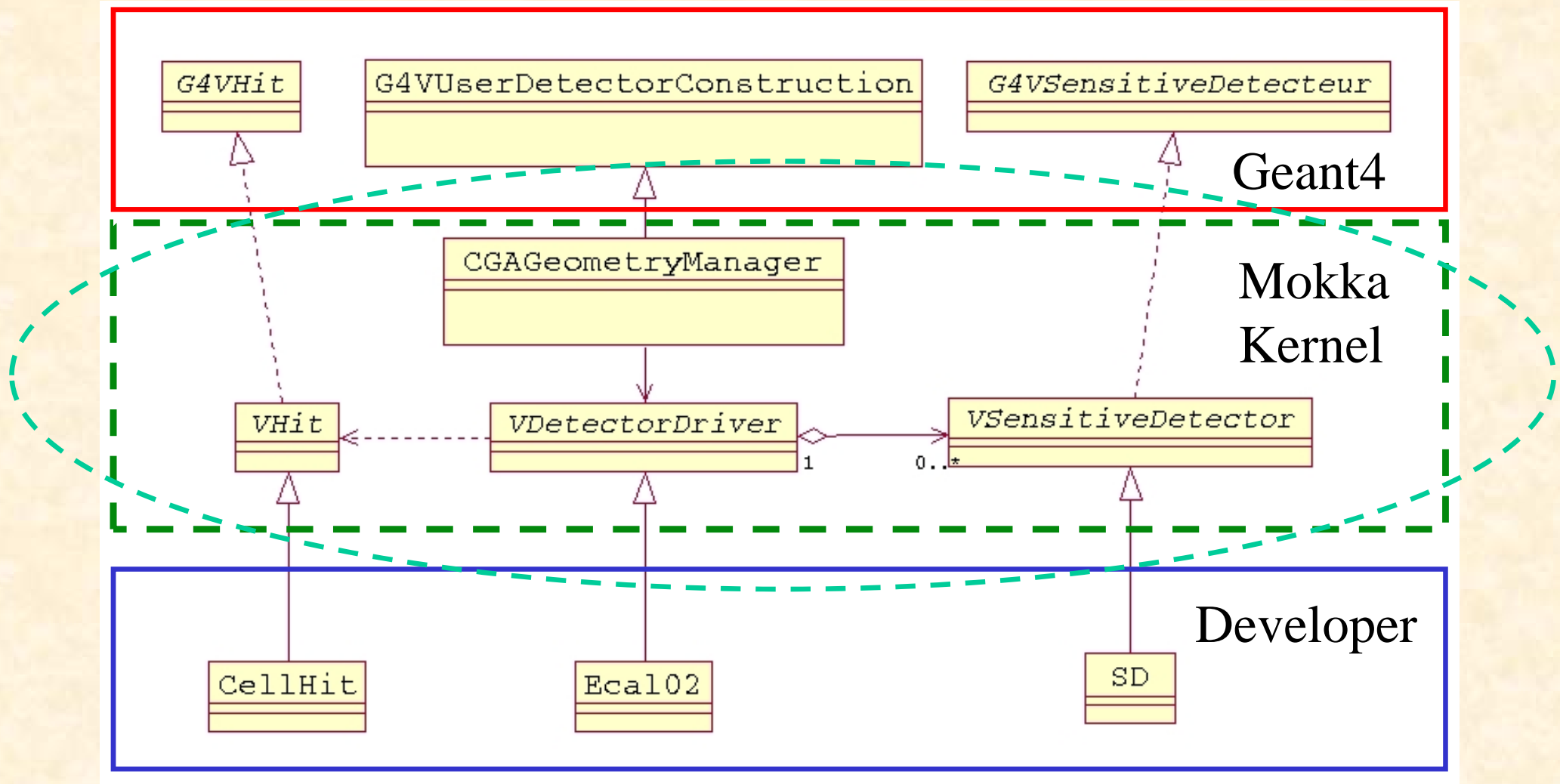


Some good features...(4)

- Plug-in user actions, the possibility to define several run time « user actions » via plug-ins:
 - virtual void *BeginOfRunAction* (const G4Run *)
 - virtual void *EndOfRunAction* (const G4Run *)
 - virtual void *BeginOfEventAction* (const G4Event *)
 - virtual void *EndOfEventAction* (const G4Event *)
 - virtual void *PreUserTrackingAction* (const G4Track *)
 - virtual void *PostUserTrackingAction* (const G4Track *)
 - virtual void *UserSteppingAction* (const G4Step *)

The strategy:

- Improving the Mokka Kernel:

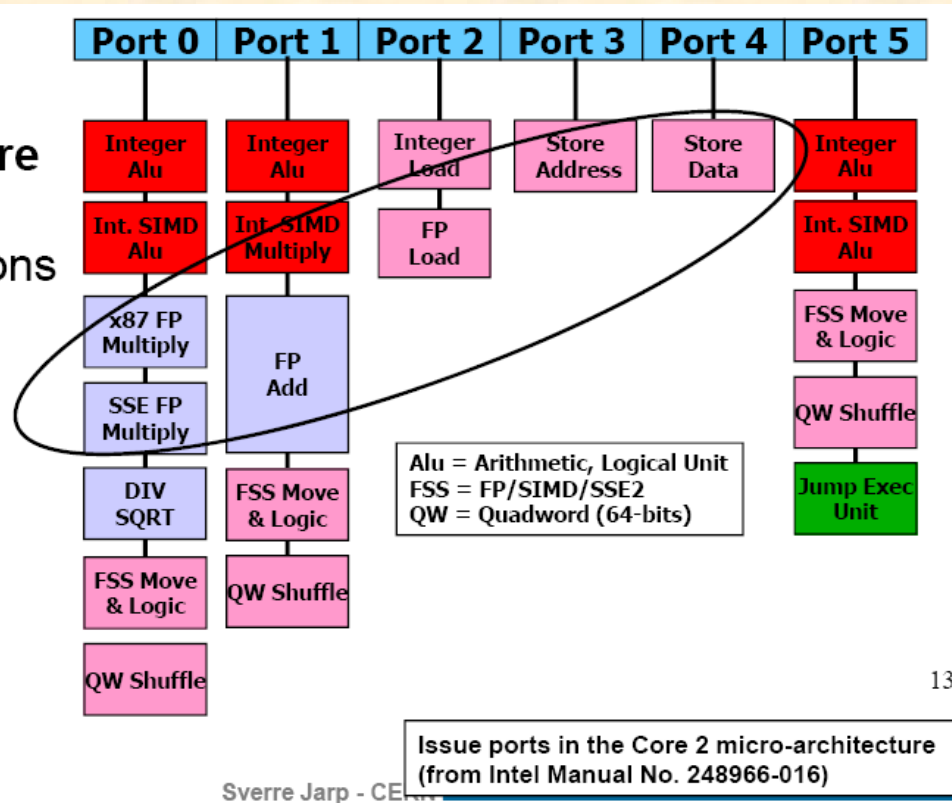


Improving performance (1)

- Optimizing CPU use:

(thanks to Sverre Jarp / Cern)

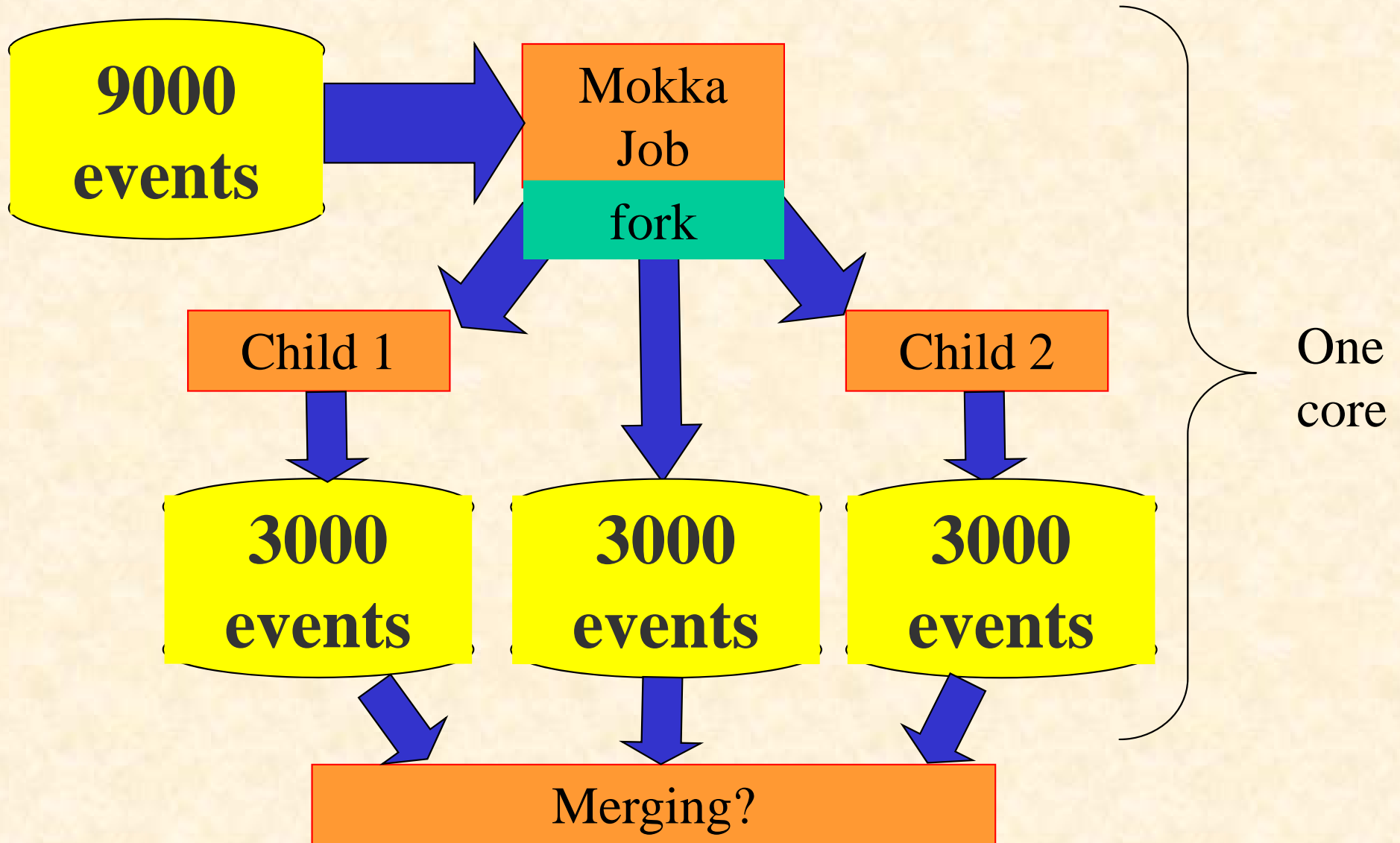
- Intel's Core microarchitecture can handle:
 - Four instructions in parallel:
 - Every cycle
 - Data width of 128 bits



“Like having a Ferrari, but using only...
 the first gear.”



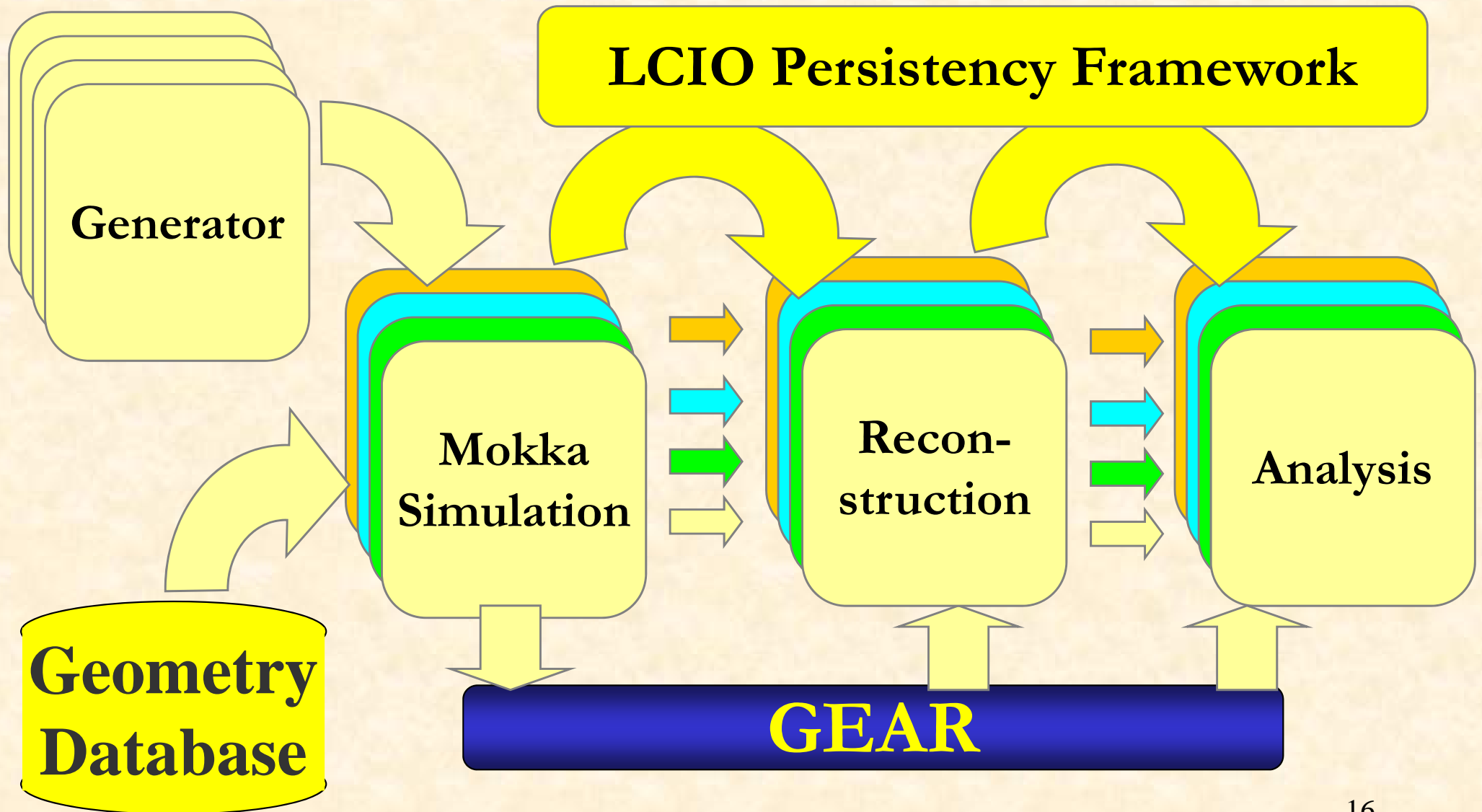
Improving performance (1)



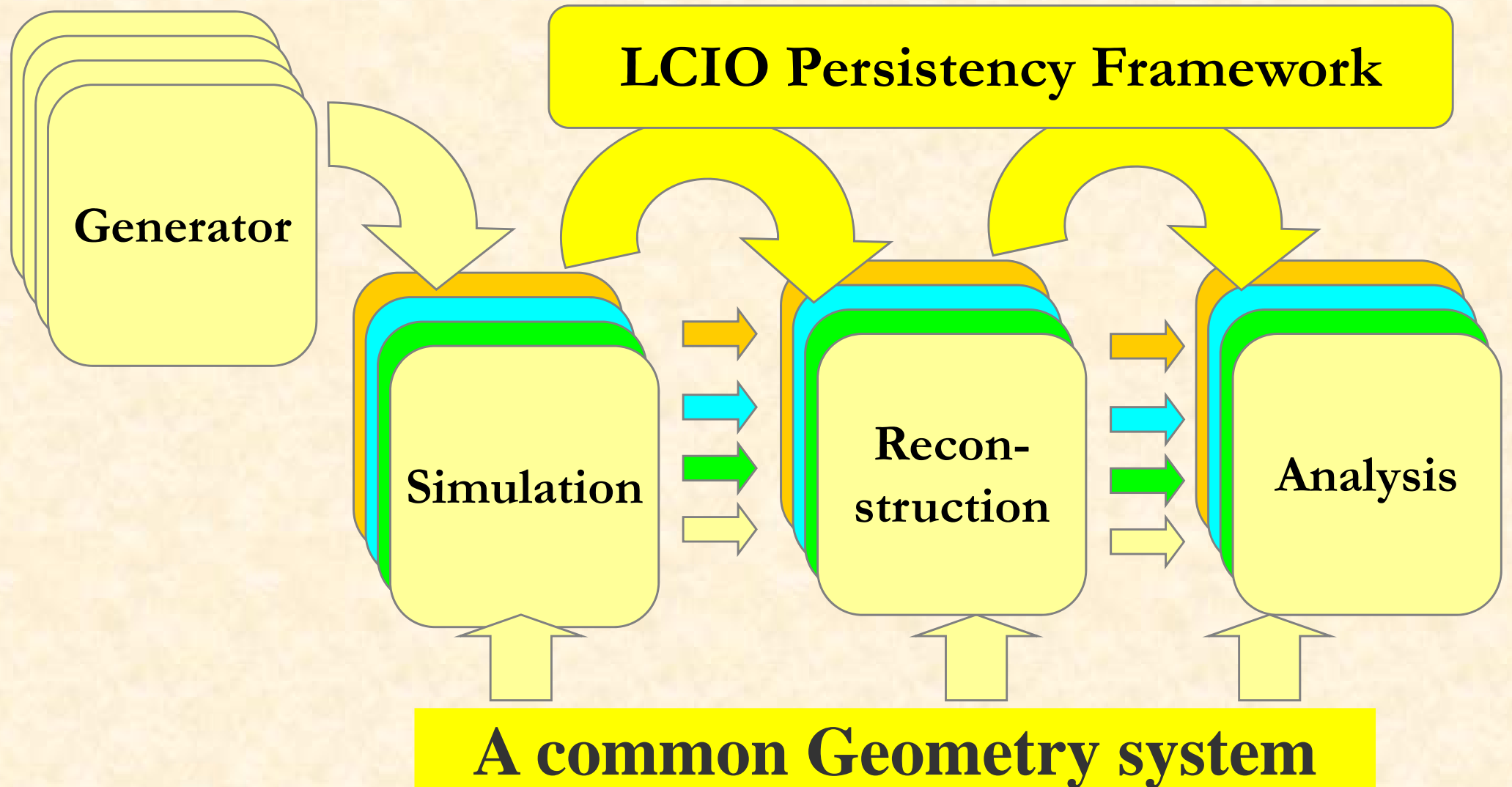
Improving performance (2)

- Switching fast/detailed simulation in Mokka :
 - It's a standard Geant4 feature
 - It can take the control on a detector region, thanks to the “ghost volumes”
 - Can be generic (ex: Gflash for e-/e+ showers)
 - Could generate hits or reco-like objects directly written into the lcio output file
 - Has to be implemented by specialists, but should be driven by end users depending on their needs
 - Could help providing quick answers, depending on the studies being done

Sharing geometry today



Sharing geometry in the future?



While waiting for the future...

- The model parameters for each sub detector are there :

Base de données models03 - table parameters

requête SQL

		name	description	default_value
Modifier	Effacer	Ecal_Barrel_halfZ	The half Z size of Ecal barrel. It's a master parameter for LDCxx_02yy.	2206.25

[Insérer un nouvel enregistrement](#)

...at the geometry data base

While waiting for the future...

– But also there :

...

Building sub_detector SEcal03, geometry db VOID, driver SEcal03:

A scalable LDC Ecal driver without database, just parameters.

Current parameters for the SEcal03 detector :

- Ecal_Alveolus_Air_Gap = 0.5

- Ecal_Barrel_halfZ = 2350

- Ecal_EC_Ring_gap = 10

...

... at the Geometry Manager level, at run time.

While waiting for the future...

– And a few ones are also there :

...

```
<detector name="TPC" geartype="TPCParameters">
```

...

```
<parameter name="tpcInnerRadius" type="double" value="3.290000000e+02" />
```

```
<parameter name="tpcOuterRadius" type="double" value="1.808000000e+03" />
```

...

...at the GearOutput.xml file.

(But depending on the detector driver code)



Proposal to improve the geometry sharing (while waiting for the future):

- Exporting the model parameters for each sub detector into the GearOutput.xml file should be:
 - A Mokka Kernel responsibility
 - Done automatically for all parameters & for all sub detectors
- Reconstruction / analysis developers will be able to:
 - Access to all model parameters really used by simulation
 - Reliable information and for free (it's automatic)
 - Introduce in the Mokka DB specific reconstruction parameters, providing defaults in an elegant way (as already done for the Hcal_virtual_cell_size parameter)

Improving Documentation

- Doing it automatically, with Mokka & scripts.
For example,
 - Gave a new model:
 - To create automatically 3D* pdf files, one per sub detector and fully detailed for deep inspection
 - To put it together the parameter list, description and values per sub detector in the Mokka Web page which describes the new model
 - Gave a new Mokka release:
 - Automatically indexing the release notes per subject and adding it to the Mokka Web site, in a indexed / searchable Web page

(*) concerning pdf 3D, many thanks to Norman Graf!

And don't forget the user's requests...

- New or better detector studies ask for new features or improvements. For example:
 - “Low energy particles can stay a long time looping inside the field”. So we should improve the event time structure:
 - to deal correctly with the detector response. Now probably it includes hits that are no more read by the readout system
 - to provide a way to implement and study correctly events with pile-ups (end of the previous event)
 - “Frozen models could be built in stand alone (to avoid DB accesses), helping people using grid”
 - (see MokkaDB tool)