

Mokka, main guidelines and future

P. Mora de Freitas
Laboratoire Leprince-Ringuet
Ecole polytechnique - France
Linear collider Workshop 2004, Paris

The first main guideline:

Let's compare detector designs in a common framework !

- It means, to simulate different detector models with :
 - The same physics
 - The same simulation framework (Geant4)
 - The same input data files
 - The same output data format (to be able to apply almost the same reconstruction and analyze programs)

The second main guideline:

Keep it simple, stand alone as possible and available for all !

- It means,
 - to just rely on the Geant4 framework functionalities and standard building tools (gmake)
 - To write out just ASCII files
 - To provide an open wide access via the Mokka WEB page

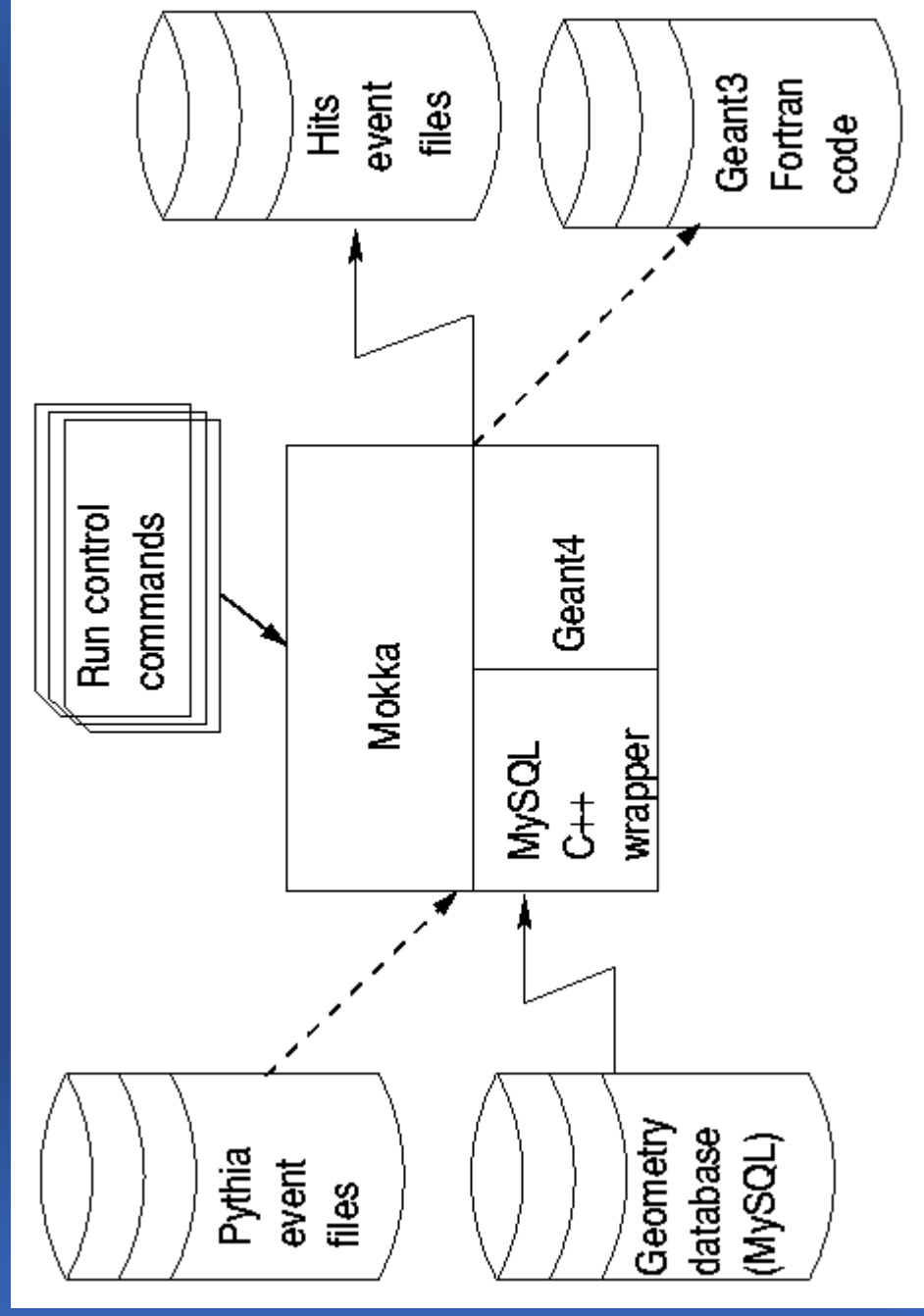
The third main guideline:

Keep trace !

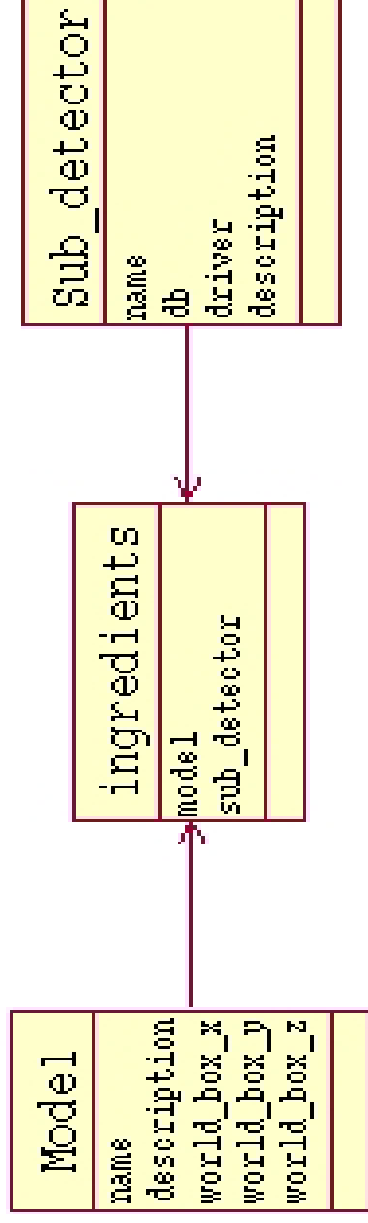
- It means,
 - Tagged releases (almost monthly)
 - Release notes for each release
 - Just one reference geometry database, indeed local copies can be useful while developing new detector models
 - Output data refers to a known detector model in the reference geometry database
 - Run log file (detector model, geant4 and Mokka release tag ids, line command parameters, etc)



Mokka first architecture and relationships

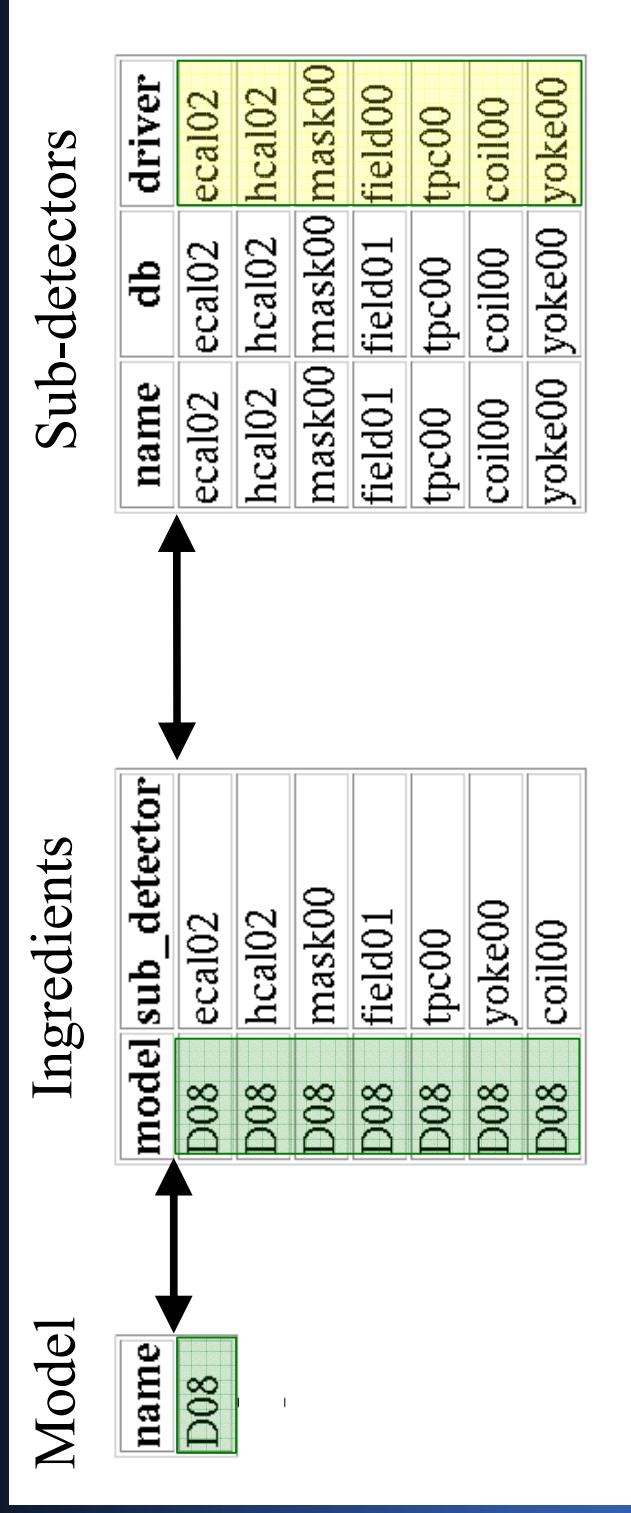


The geometry database

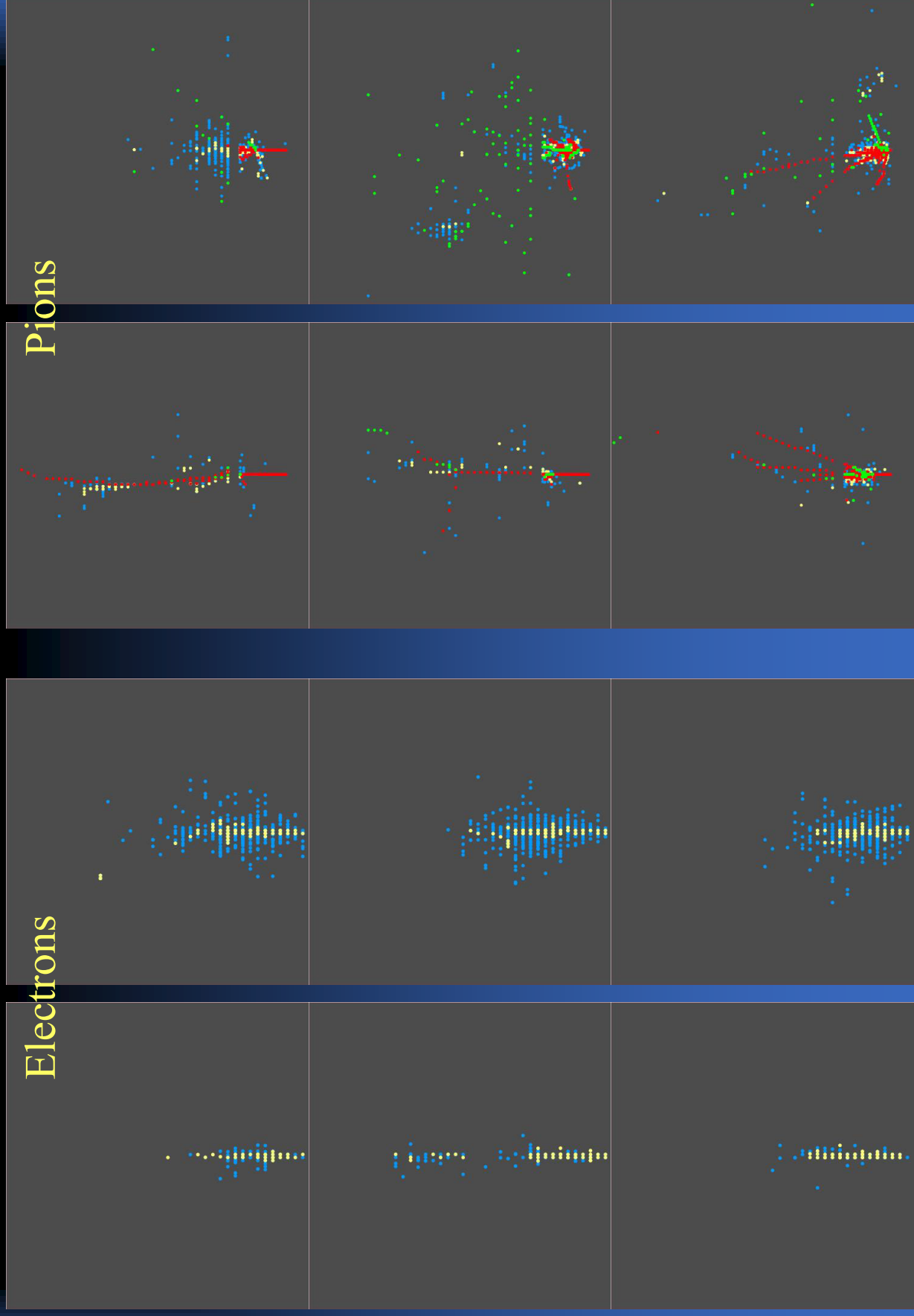


The geometry database

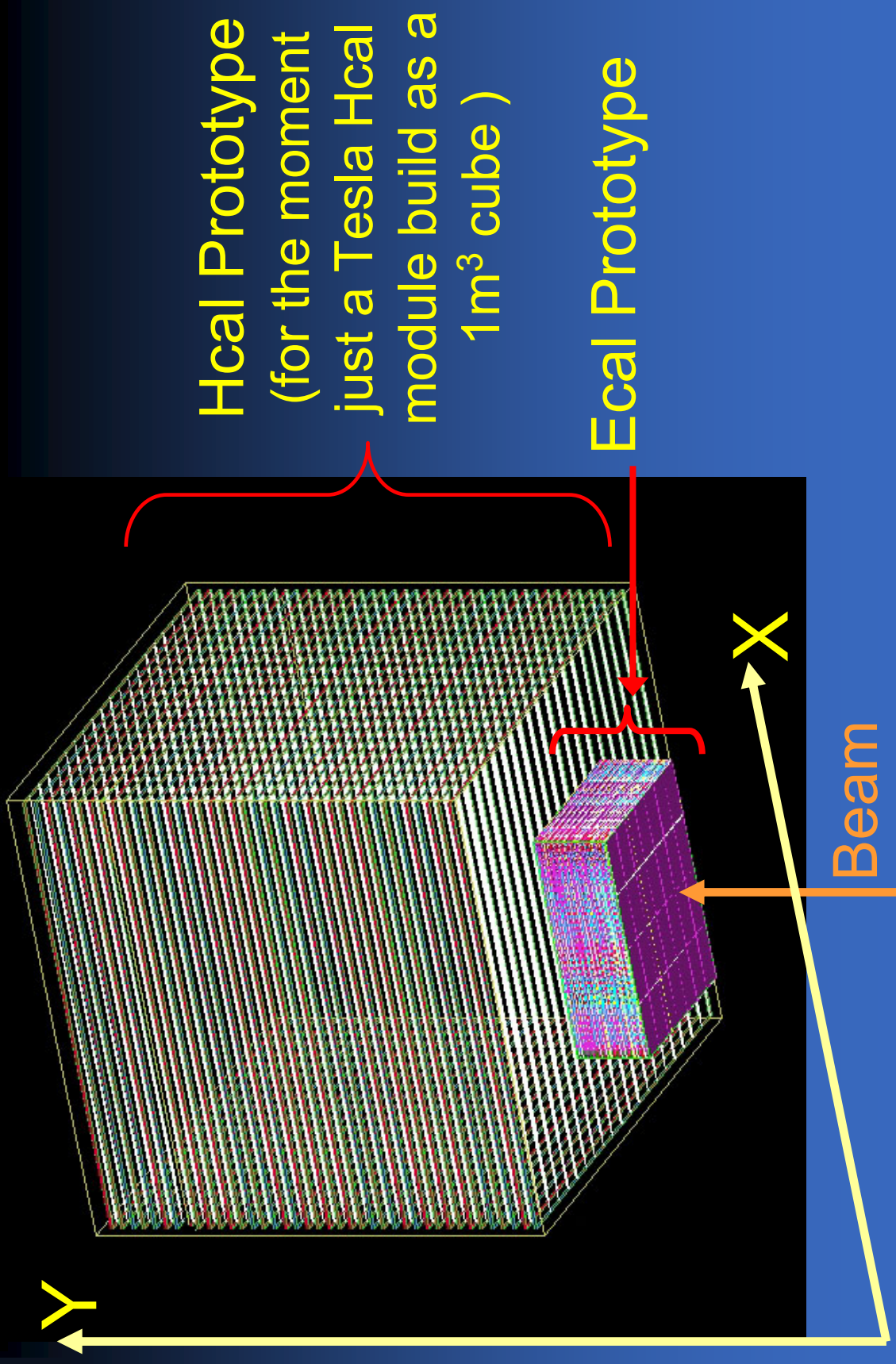
A detector model sample: “D08”



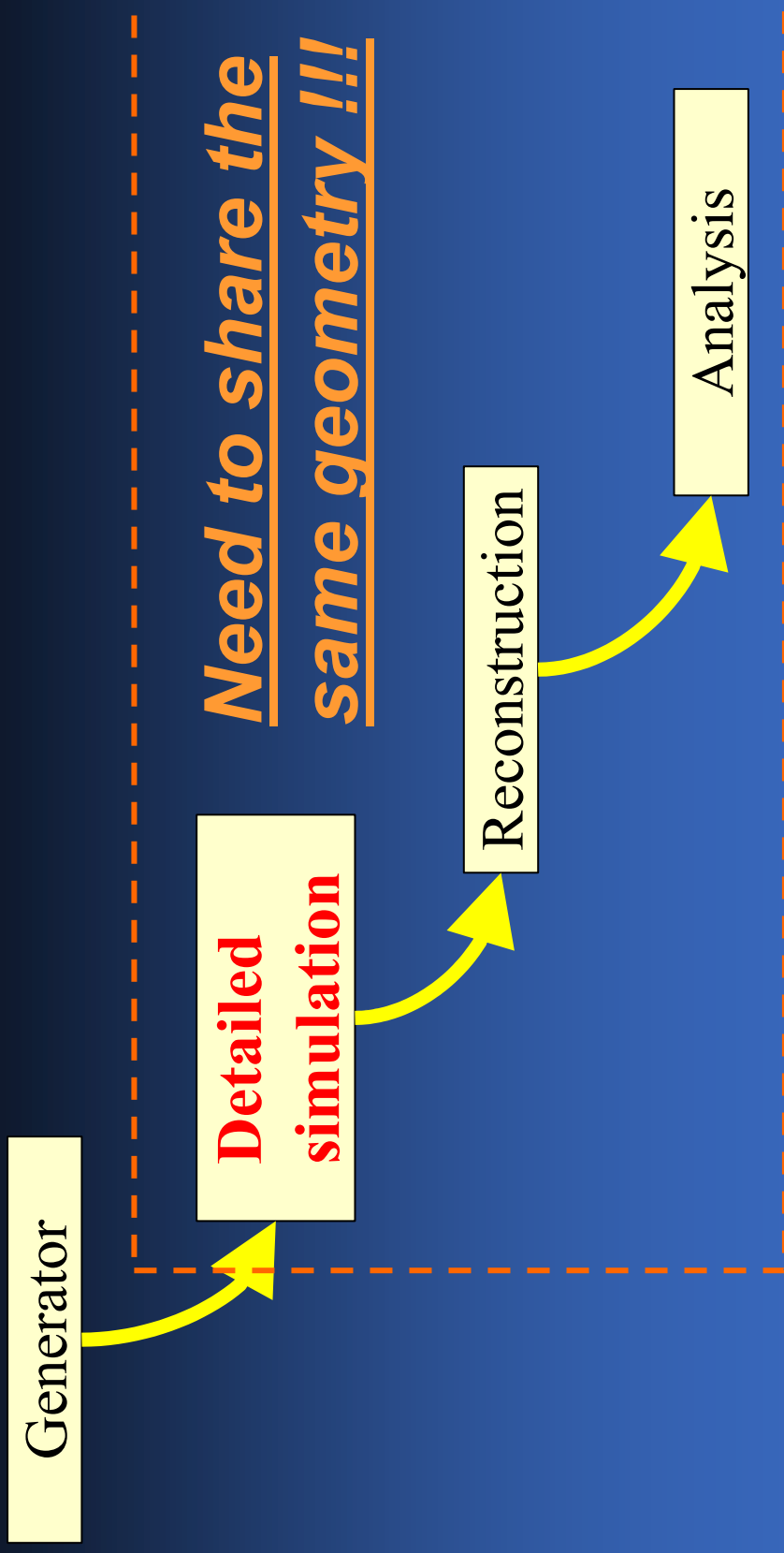
Impact of the gas in HCAL



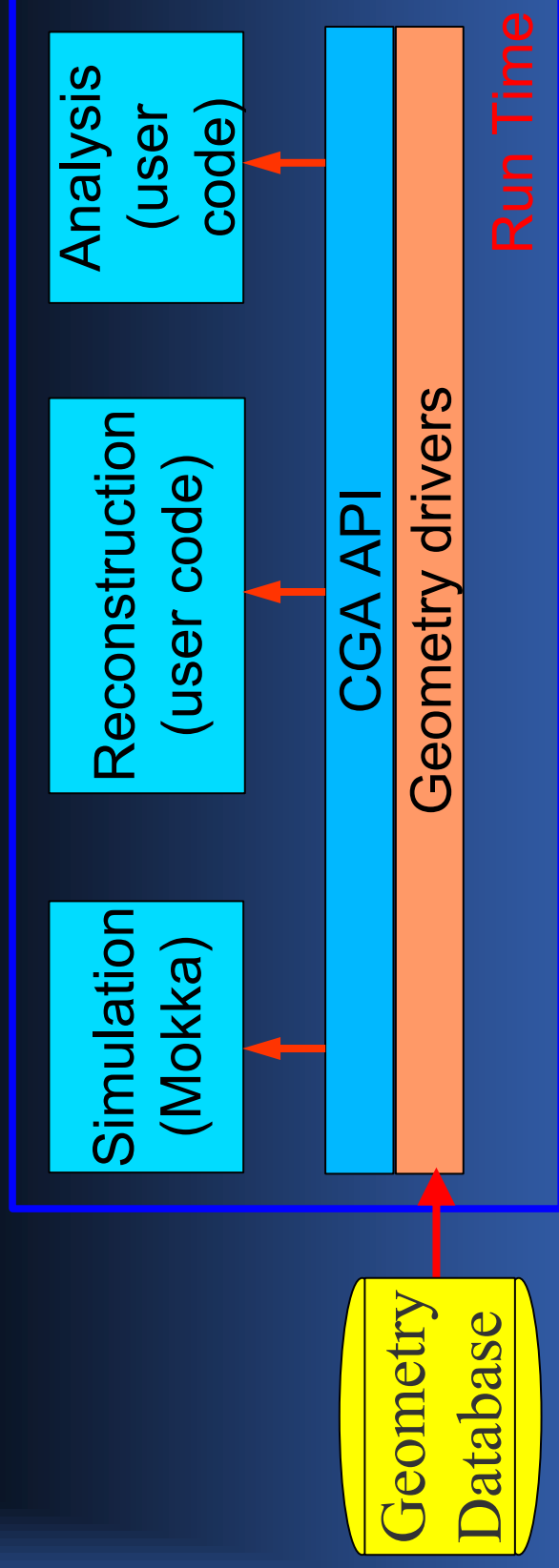
Physics validation: Calice



What we learned I: Geometry have to be shared



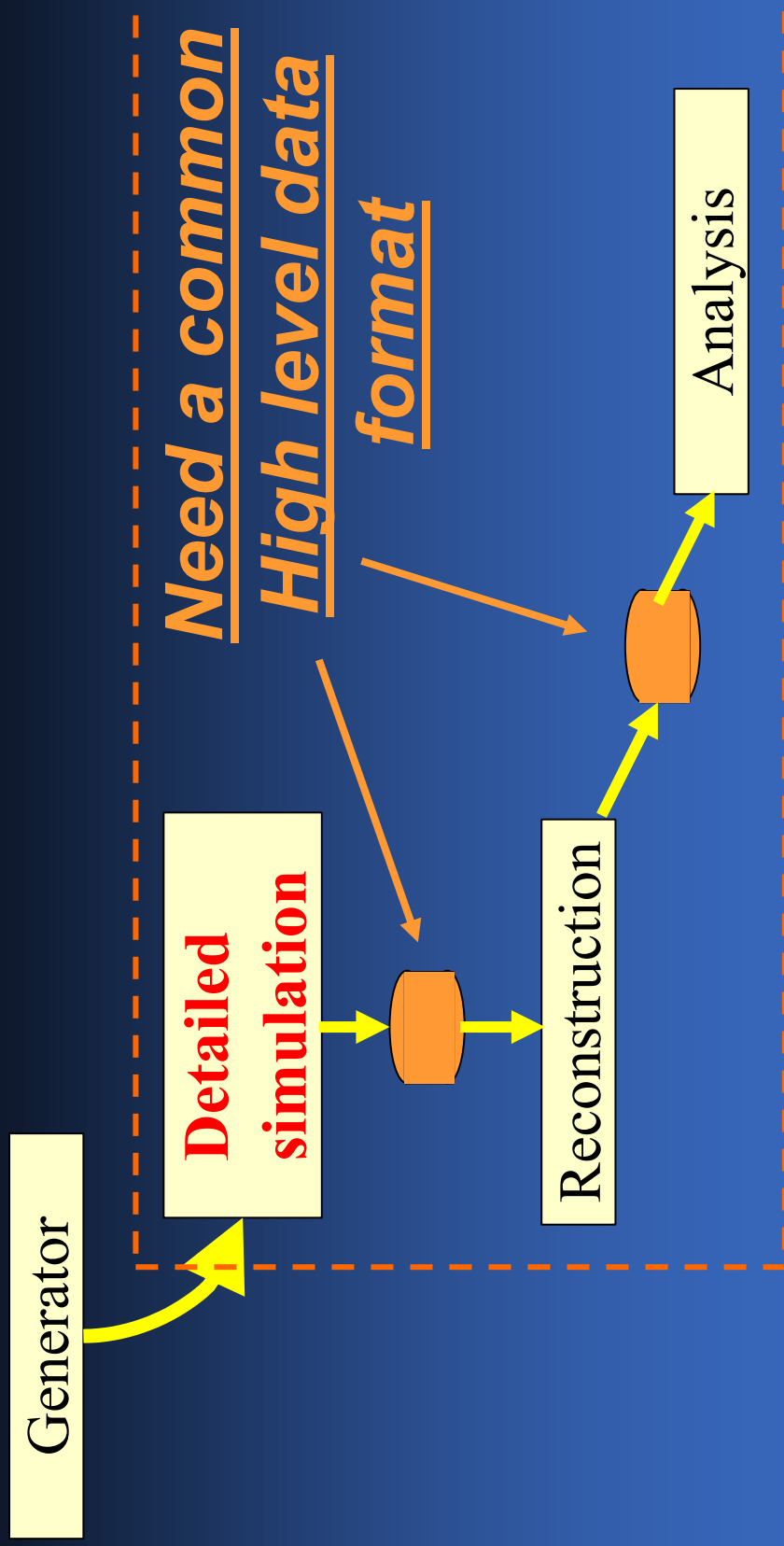
C.G.A: the Common Geometry Access API (F77, C++, C and Java)



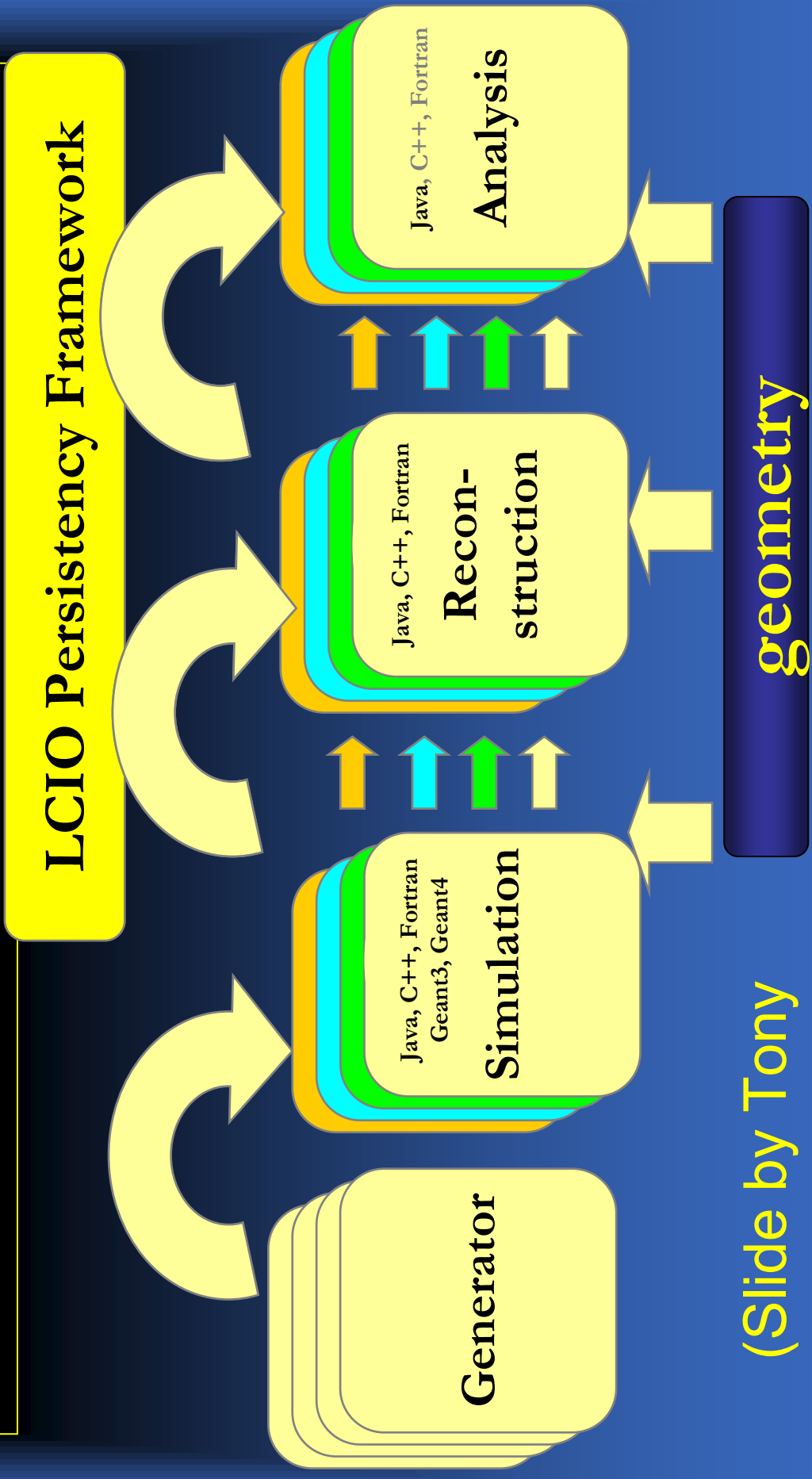
- Relies on Geant4 geometry layer (By G. Musat)
- Implements some reconstruction utilities.

What we learned II:

Data format have to be shared



Motivation

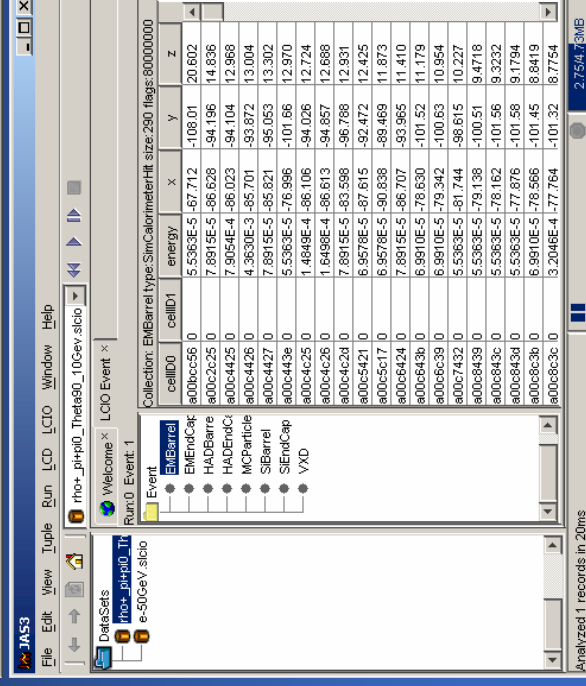
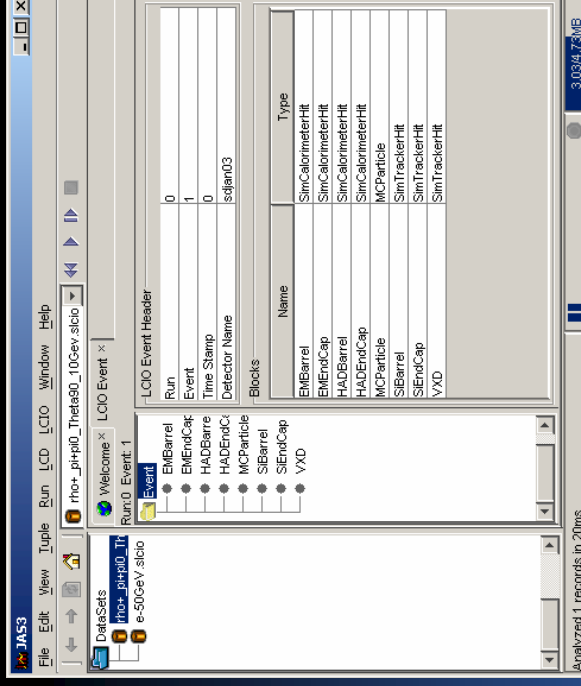


(Slide by Tony Johnson)

JAS3 + LCIO

- JAS3 Plugin exists for viewing and analyzing LCIO files
 - Allows to directly open and browse any LCIO
 - Works with any LCIO file
 - no requirement for prior knowledge of what is being read.
 - Recently updated to support LCIO 1.0
 - (in fact 0.8 or later)

(Slide by Tony Johnson)



What we learned III:

New users ask for new functionalities and use cases !

- Big thanks to Frank Gaede (DESY) for a set of new user facilities for Mokka:
 - steering files
 - init commands
 - environment variables
 - Plugin interface for Geant4 user actions, mainly for “on line” analyzes

But what we really learned: *We have to work together !*

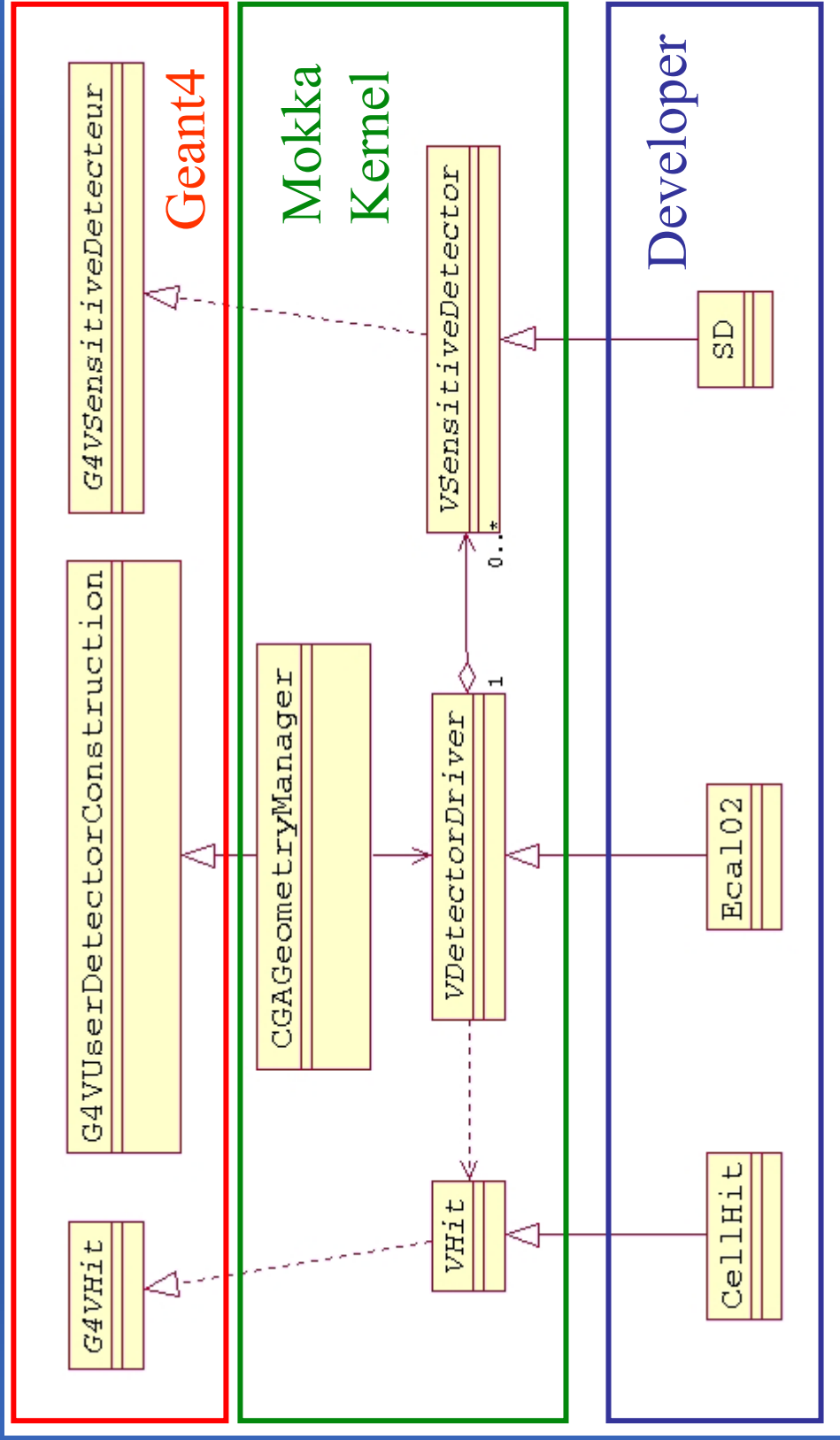
- Big improvement since the LClO interface
- The LCWS04 Mokka release includes very good contributions also from
 - Jeremy McCormick (NICADD)
 - Frank Gaede (DESY)

Future

- We have to improve working together:
 - by an informal developers committee and its organization around the CVS repository management, code standards, Geant4-LC users interface, etc.
 - to bring adequate level of detector descriptions to this common simulation tool.

 **Job for detector specialists**

Mokka's kernel framework



Who is who for the next model improvements in Mokka?

- Ecal:
 - W/Si calorimeter and its prototype:
detailed model by Ecole polytechnique
 - Other options: ...
- Hcal:
 - Digital and its prototype:
 - RPC: generic model by Ecole polytechnique, detailed model by ...
 - GEM: detailed model by Nicadd (Jeremy McCormick)
 - Analog (scintillator, etc): generic model by Ecole polytechnique, detailed model by ...

Who is who for the next model improvements in Mokka?

- Inner tracking devices:
 - Generic model by Ecole polytechnique (directly translated from Brahms), detailed model by ...
 - The SET: LPNHE – Paris + ...
- Forward detectors (masks, LAT, ...):
 - Generic model by Ecole polytechnique (directly translated from Brahms), detailed model by ...
- Muon chamber: ...
- Beam line: ...

Conclusions:

- 1) *“Improve framework” but
“Keep things simple”.*
- 2) *“Let’s compare detector designs in a
common framework”, to do this
“We need adequate level of detector
descriptions”.*

 *Let’s work together !!!*