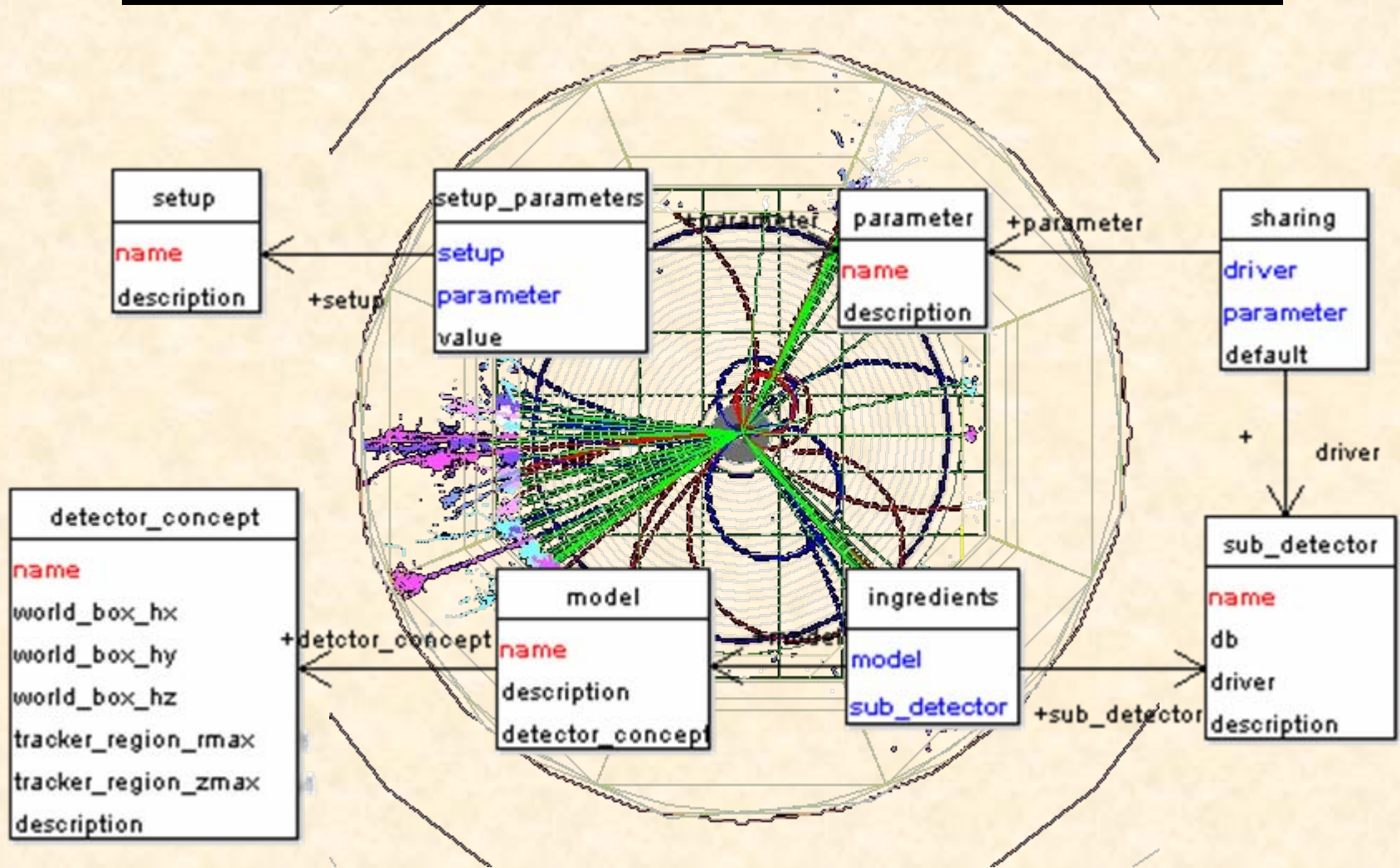
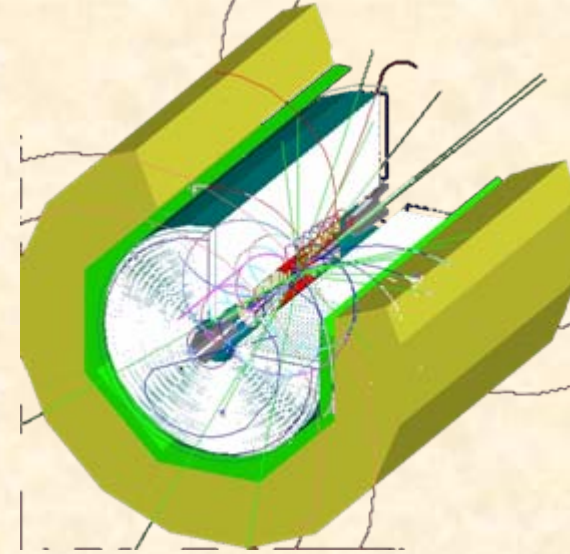


# *Linear Collider Detector Concepts with Mokka/Geant4: present and future*



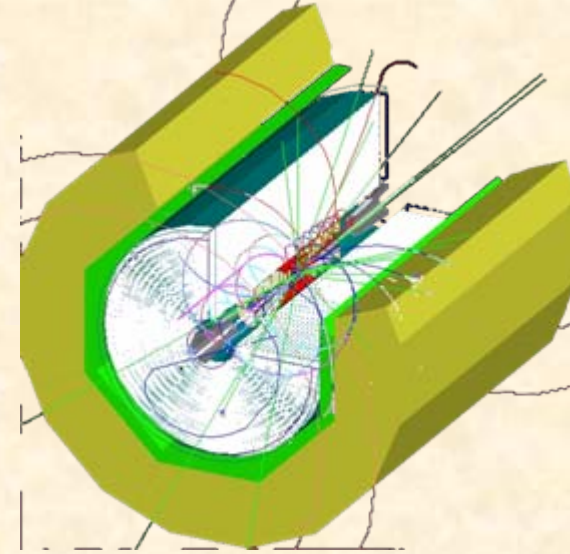
# What is MOKKA ?

- A Geant4 full simulation for the Linear Collider detectors in use since Dec'99
- Extensively used for the TESLA T.D.R. calorimeter energy flow studies.
- Able to simulate several detector and prototype models thanks to its Geometry Database architecture:
  - To share the same Geant4 Physics and Kernel (and bugs...)
  - To check against real data from prototypes (as the Calice one)
- Open for a wide collaborative use and development.
- Currently developed by L.L.R., DESY, NIU...
- Currently in use in several laboratories around the world.



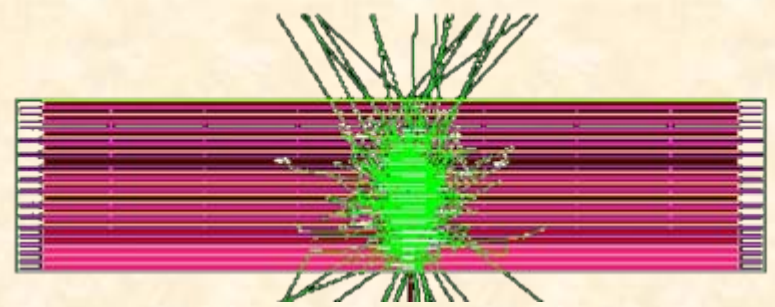
# Mokka Kernel

- Almost all features are up to date
  - ✓ particle  $\leftarrow$  hit assignment:
    - ✓ 100% compliant with the user requirements
    - ✓ Implemented as a run time user choice
  - ✓ Simulator status word:
    - ✓ 100% compliant with the LCIO requirements
  - ✓ Input ASCII and binary stdhep files
  - ✓ LCIO compliant output file format
  - ✓ Physics lists:
    - ✓ Implemented as a run time user choice
    - ✓ SLAC “Linear Collider Physics List” also available
- To be foreseen
  - Event files X LCIO particle list collection: LCIO files will be read as input event files (coming soon)





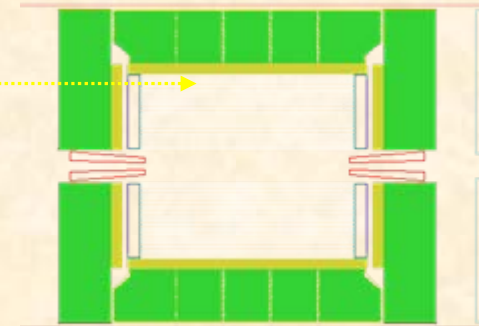
# Detector models status



- ✓ Test beam is quite complete with the Ecal, Hcal prototypes and the catcher
- ✓ A new driver for the TPC improves the geometry of the Tesla TDR TPC.
- ✓ The SI/W Ecal maintained by L.L.R.

**BUT**

- A lot to do for final detector studies:
  - ? all the others inner devices
  - ? Detailed Scintillator and RPC HCal models
  - ? The several detector concepts

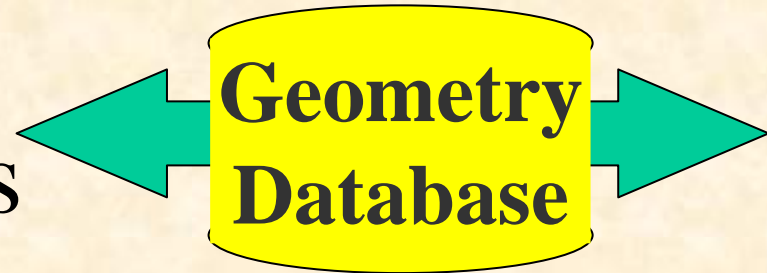


A big challenge to have done with the detector studies  
in a few years !!!

# Mokka perspectives to help to face this challenge: **Geometry Database** !

- Database architecture extensions:
  - “Detector drivers should be able to share global parameters describing the **collider parameters**, the **detector concepts** and **special setups**.”
  - “Detector drivers should be able to scale the detector devices to agree with the global parameters”

# Database extensions: implementation status



## ✓ Shared global parameters :

- ✓ Implemented and available in the LCWS05 Mokka major release (tag mokka-04-00)
- ✓ 100% backward compatible

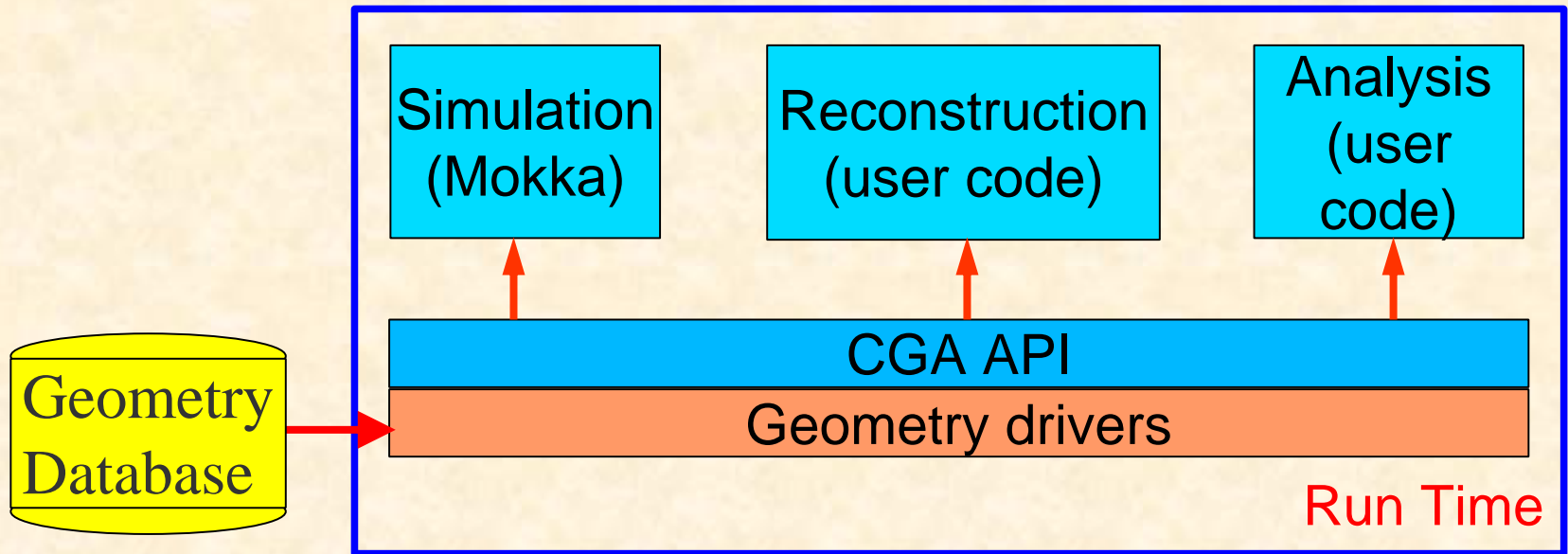
**BUT**

- ✓ Database local copies HAVE TO upload the new models01 database from our Mokka central database on pollin1.in2p3.fr

## ■ Detector drivers able to scale :

- Coming soon: the “**Super detector drivers**”, a super layer which relies on the debugged Mokka drivers.

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



- ✓ Relies on Geant4 geometry layer
- ✓ Implements some reconstruction utilities.

# C.G.A. implementation status

- ✓ 100 % implemented for C, C++, Java and F77 as a standard Mokka feature (since April 2004)
- Coming soon:
  - The CGA Marlin processor, to make available on the fly the detector geometry and the CGA reconstruction utilities for all Marlin processors in the same job.