

# Geometry - Detector Description

## Current implementations

Frank Gaede, DESY

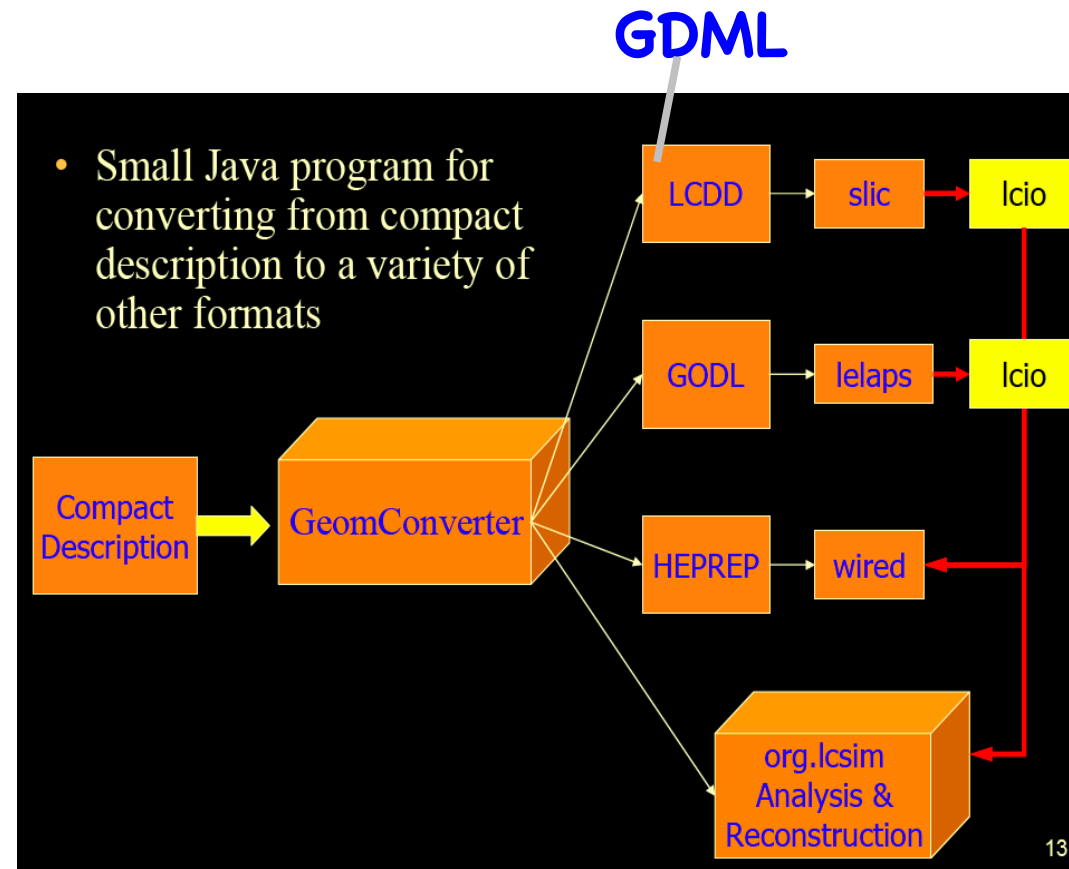
Norman Graf, SLAC

Linear Collider Software Meeting

CERN, Feb 2-3, 2012

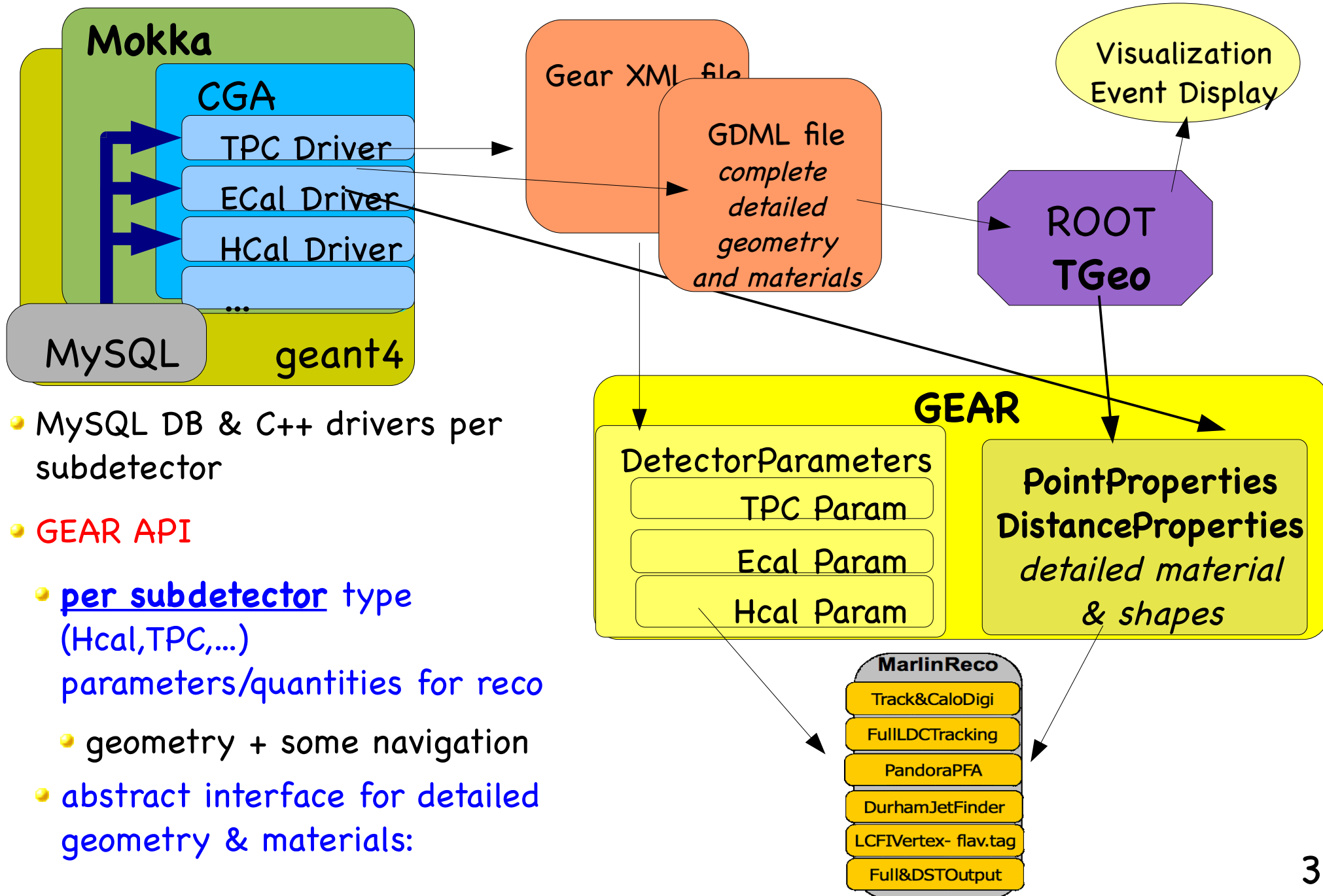
# SID geometry description

- geometry definition in xml files
- **GeomConverter** tool (java) provides various representations of geometry:
  - fast & full simulation
  - event display
  - reconstruction
- detailed geometry for geant4:
  - **LCDD** - extension of GDML
- reconstruction:
  - cellid <-> position
  - local <-> global coordinate
  - find neighbors
  - materials, shapes
  - conditions (time- dependent) data



- rather complete geometry framework for simulation and reconstruction
- bound to **Java** based system (for reco code, e.g. coordinate transforms)

# ILD geometry description



- MySQL DB & C++ drivers per subdetector
- **GEAR API**
- per subdetector type (Hcal, TPC, ...) parameters/quantities for reco
- geometry + some navigation
- abstract interface for detailed geometry & materials:

# issues with current approach

- Mokka/Gear:
  - storing geometry in MySQL-DB somewhat inflexible
  - artificial split into simulation and reconstruction geometry
  - lack of navigation functionality
    - needed for improved tracking code
  - historically and organically grown – hard to maintain
  - ...
- lcsim:
  - need for more sophisticated navigation/propagation
  - split between simulation and reconstruction implementations
  - would like more sophisticated field maps

would be useful to have common tools or common exchange formats

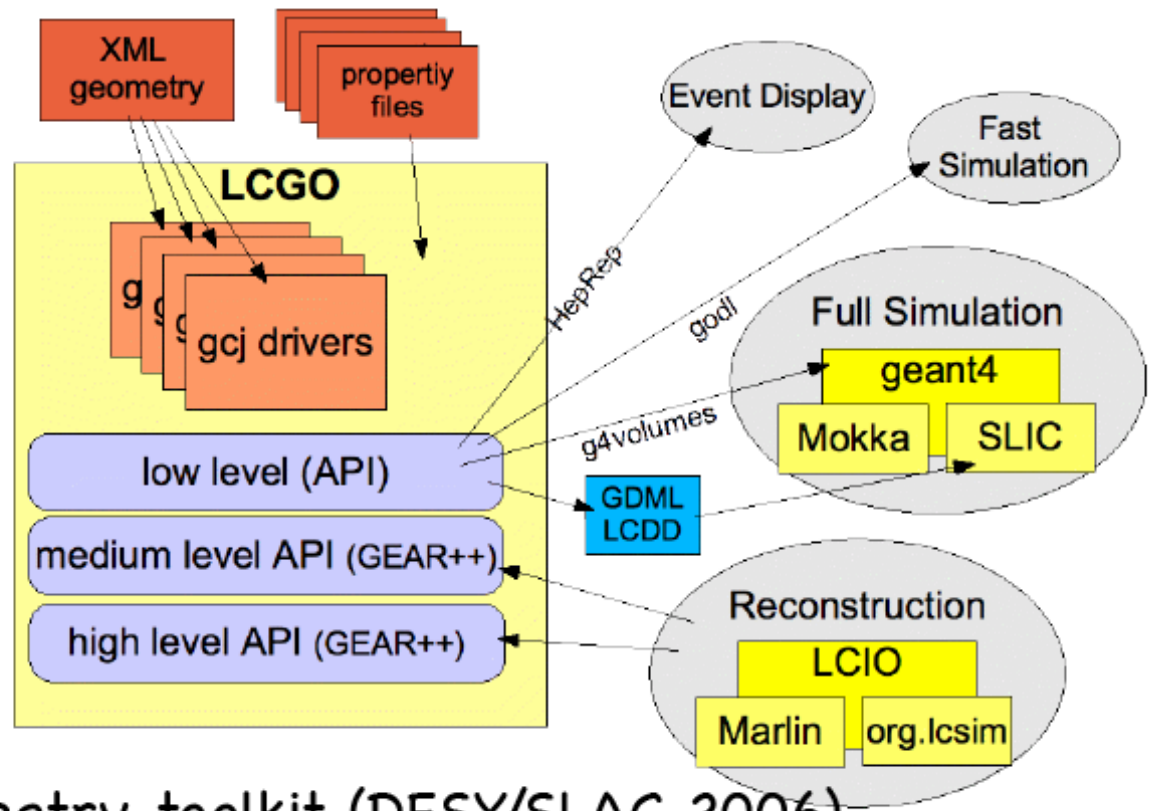
# wish list for a geometry tool

- full detailed geometry description for full simulation (geant4)
- reconstruction( pattern recognition):
  - simplified detector geometry
  - surfaces: planes, cylinders,...
  - intersection with 'next' volume
  - dE/dx
  - access to volumes
    - #layers, thickness, width,...
- material database
- field maps
- detector properties (sampling fractions)
- readout properties
  - cellId <-> position
  - cell sizes
  - neighbor cells/elements
- Vector and Matrix classes
  - ThreeVector, Point3D
  - FourVector
  - SymMatrix (covariances)

most of these features already exist in either or both of the current systems - but not in a consistent and interchangeable way

# LCGO geometry tool - a conceptual idea

slide shown @ LCSWWS 2009  
-> conceptual idea might still be useful



- LCGO - a planned geometry toolkit (DESY/SLAC 2006)
- based on geometry drivers - written in **JAVA** !
- use **gcj-compiler** to compile to binary & interface with C++
- issues with performance - 4 times slower than C++ (2007)
- -> could look into implementing a similar concept in C++
- investigate existing packages TGeo, VGeometry,...