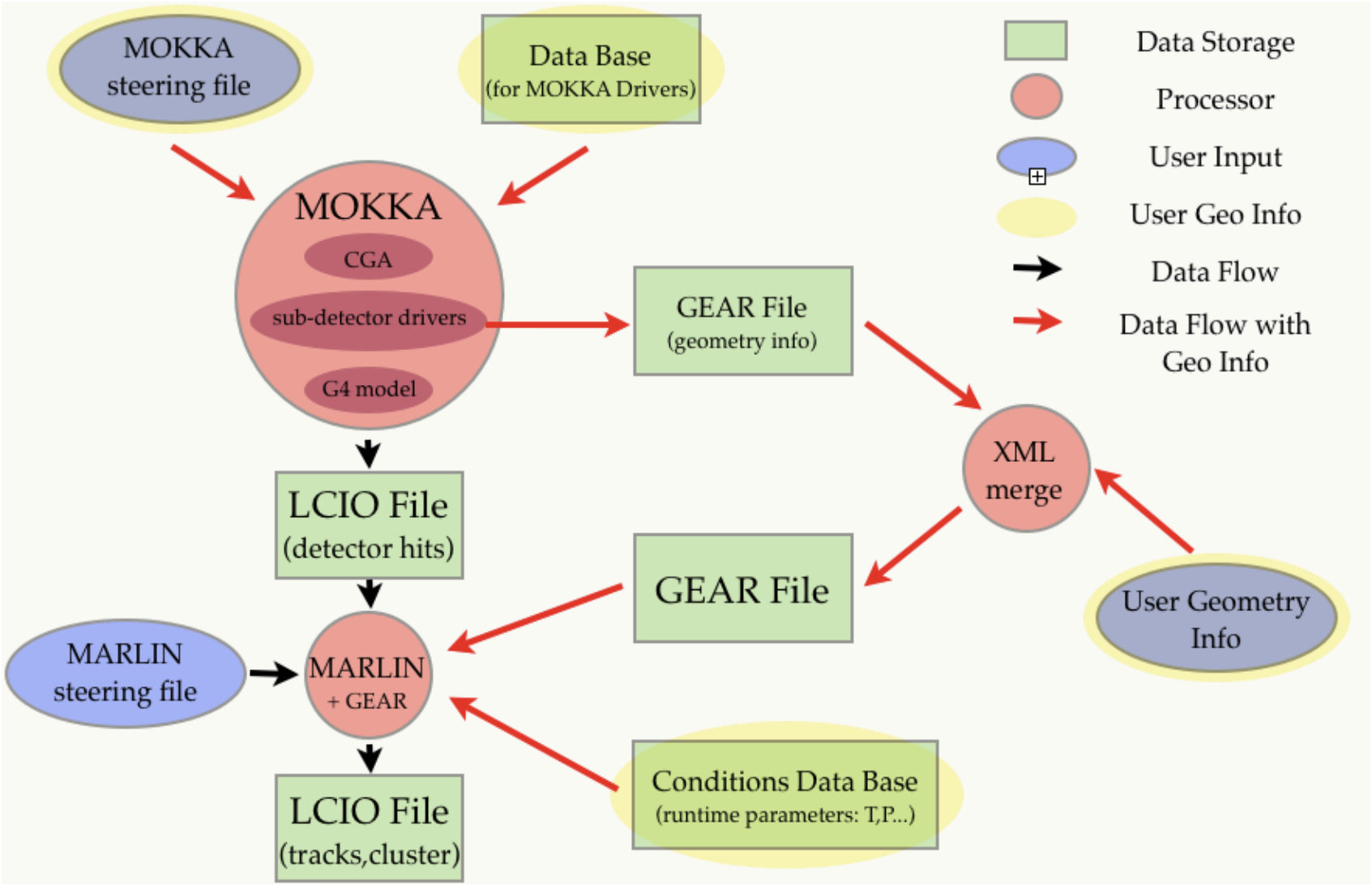
# GEAR: Extension based on ROOT TGeo

A. Muennich

CERN

Linear Collider Software Meeting, 5 July 2010
DESY

# Software Overview: Geometry Information



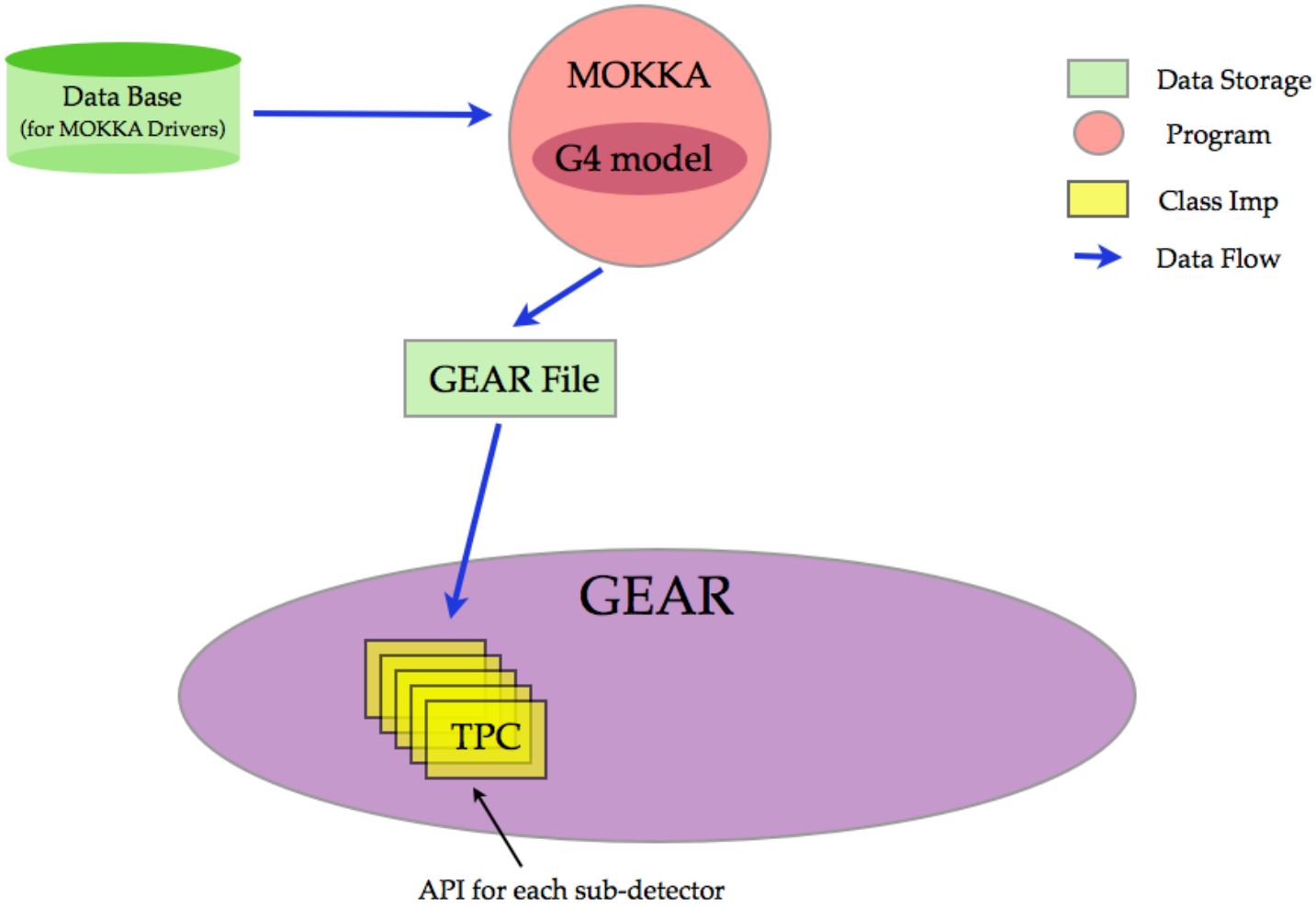
# Issues

- No central place for geometry information:
  - Geometry can be changed in various places (inconsistencies)
  - Changing geometry requires changes in C++ code, XML, DB..
- No representation of geometrical volumes for complex questions, e. g. radiation length, misalignment, no information for coordinate transformation (global $\leftrightarrow$ local) etc...
- No easy visualization at later stage (after MOKKA)
- GEAR interface needs extension, too many user-parameters written from MOKKA
- Software maintenance aspects: e. g. MOKKA code contaminated with GEAR

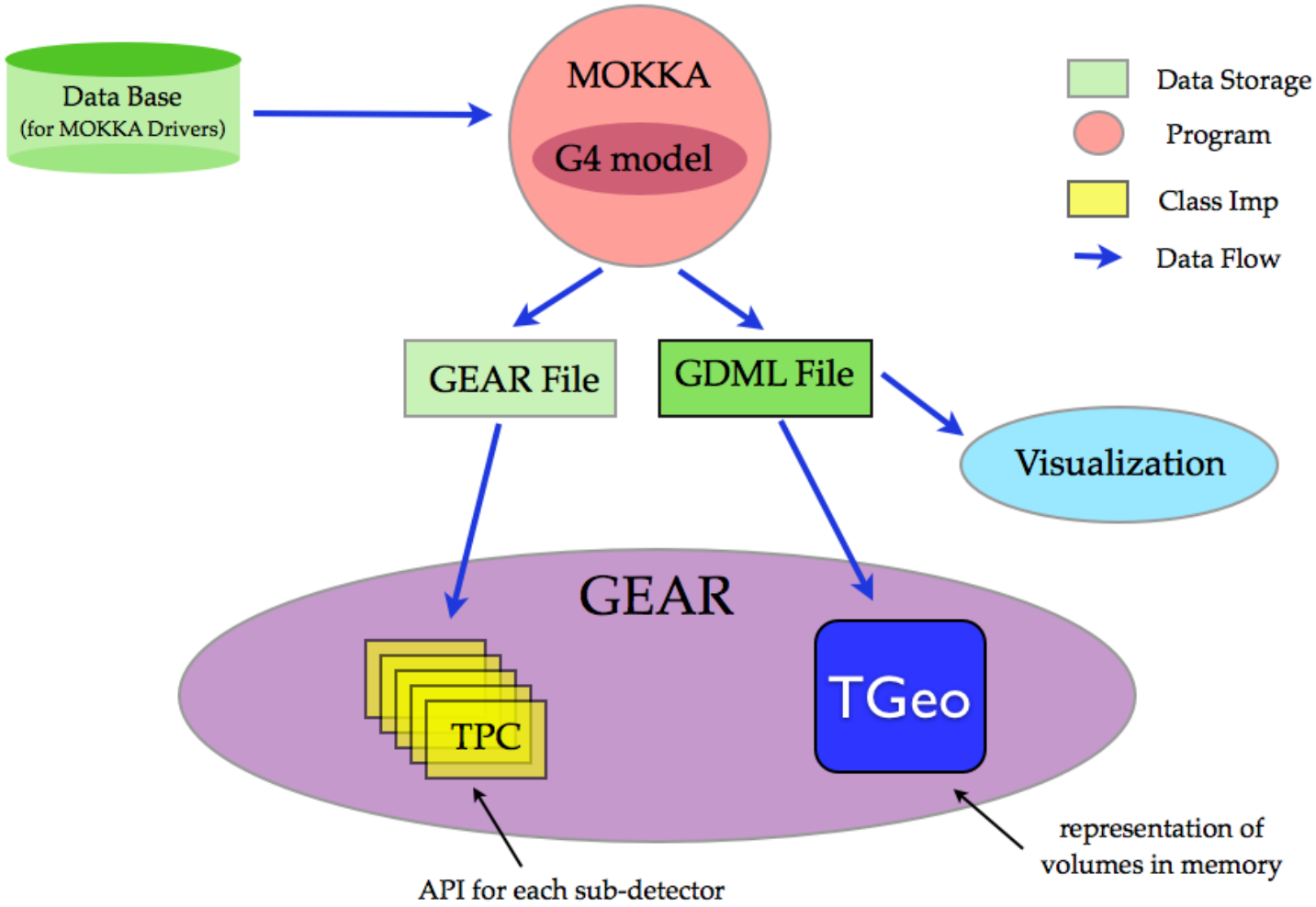
# Idea

- **Starting Point:**
  Use what we have!
- **First step:**
  Create a 3D geometry description available to reconstruction and analysis
- Information needed hidden in MOKKA:
  How to get it out and make it persistent?
  $\rightarrow$ GDML
- **Available:**
  ROOT understands GDML and creates a volume hierarchy using TGeo, very much like G4 shapes
- **Idea:**
  MOKKA dump to GDML $\rightarrow$ TGeo from GDML to service GEAR

# Geometry Information Flow



# Geometry Information Flow



# Re-implement Point and Distance Properties

## GearPointProperties & GearDistanceProperties

Abstract interface for a class that returns the (material) properties of a given point or along a given distance between two points in world coordinates

$\Downarrow$

Needs access to geometry and material info

## So far:

CGA from MOKKA is used to re-instantiate full geometry from data base

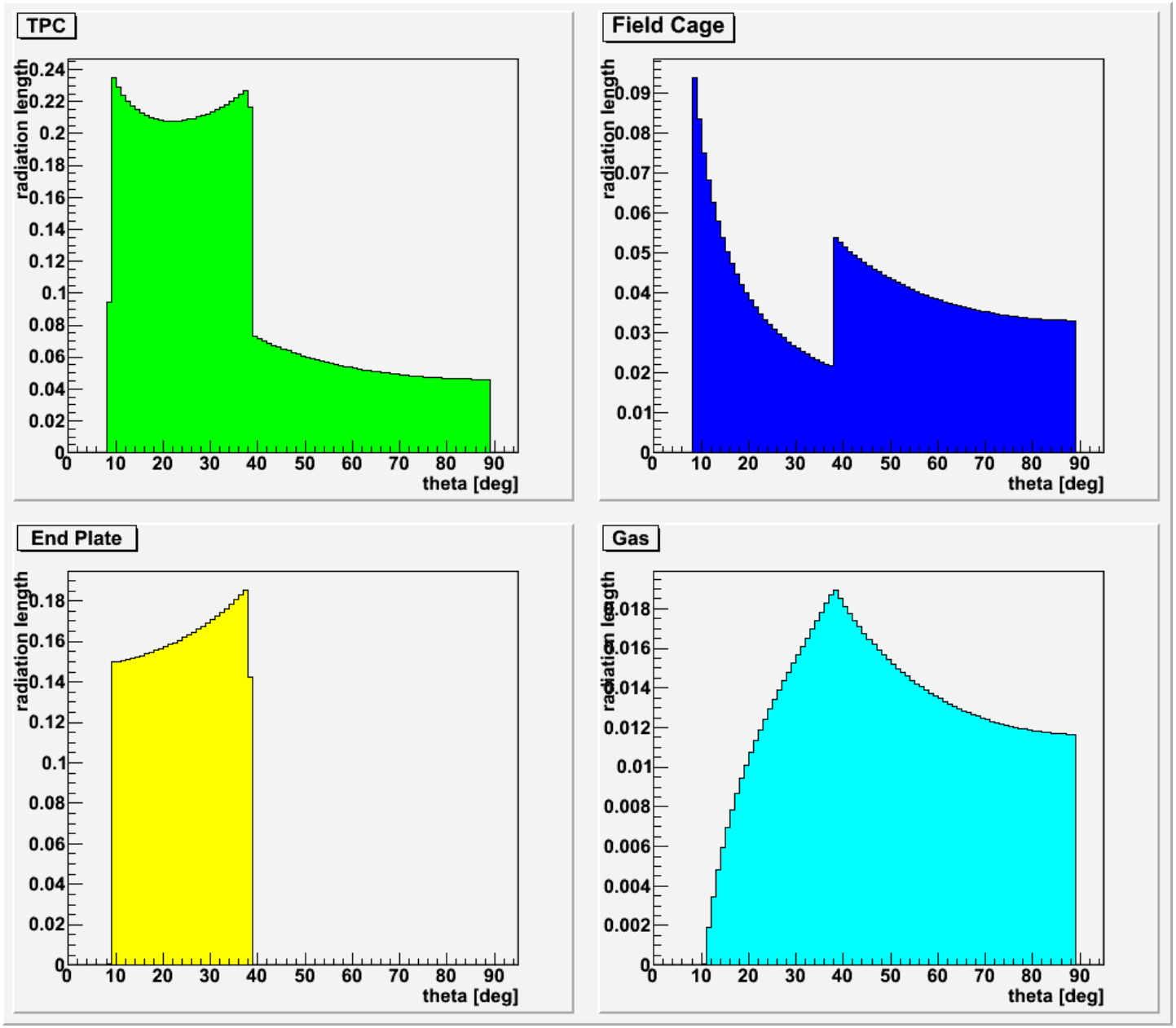
## New:

Within GEAR itself use GDML file to instantiate full geometry in TGeo

# Application

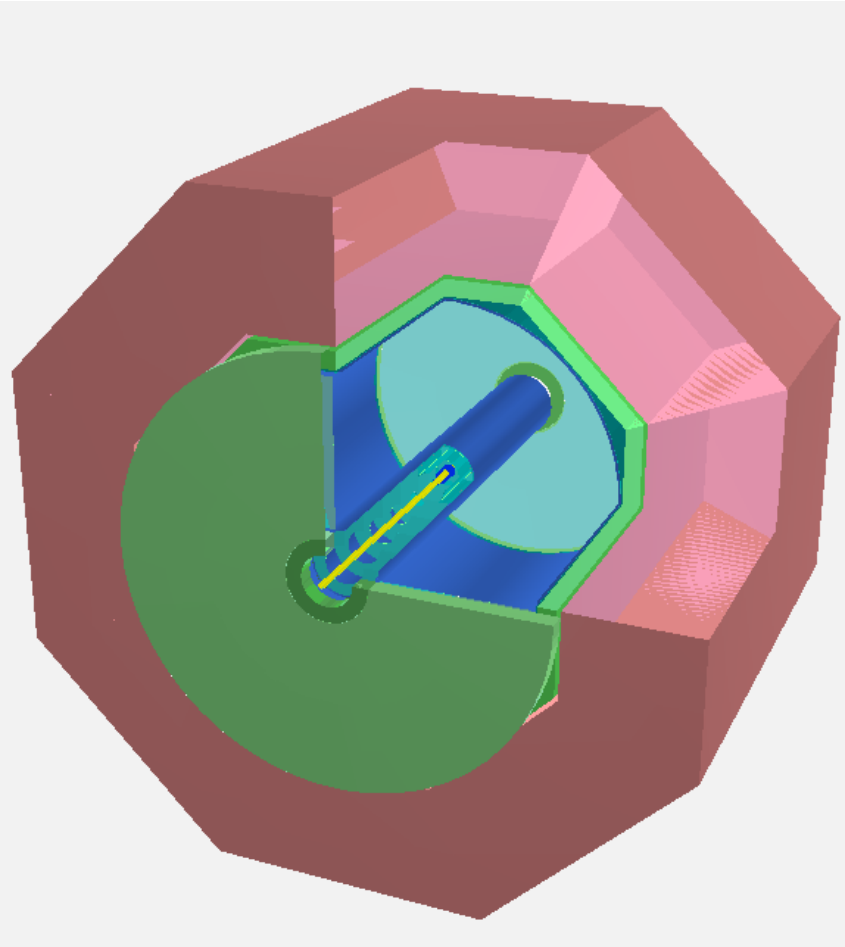
- Tracking code: Material budget between two points
- Misalignment: shift volumes or coordinates based on global to local coordinate transformation
- Access to detailed geometry info that goes beyond abstraction level in GEAR API (but requires knowledge of volume hierarchy or some convention to browse efficiently)
- Local to global coordinate transformations
- Material budget for full detector
- ...

# Example: Material Budget TPC using GEAR



# Visualization

- TGDMLParse reads GDML and generates volume hierarchy
- Large collection of shapes
- Navigation in volumes
- Visualization: OpenGL
- Direct access to TEve for event displays (like DRUID)
- In memory, or writable to file



# Changes

## Usage:

One extra line in GEAR xml file:

```
<GDMLFile name="World.gdml">
```

## Added

geartgeo:

- TGeoGearDistanceProperties
- TGeoGearPointProperties
- TGeoGeometryInitializer

## Changed:

GearXML.cc to read in GDML file and instantiate geometry using TGeoGeometryInitializer

Everything else stays the same, changes are transparent to user!

# Requirements

During development extensions were needed or bugs were found and fixed in

- ROOT: TGeoMaterial, TGeoVolume, TGDMLParse
- Geant4: GDMLParser

Therefore newest versions are needed of both

- ROOT Version 5.27/03 (svn trunk)
- Geant4 release 9.4-beta (scheduled for end of June)

GEAR with TGeo available from svn DESY:
https://svnsrv.desy.de/public/gear/gear/branches/gear_tgeo_v00

# Outlook

- Access to volume representation only through GearDistanceProperties and GearPointProperties
  $\Rightarrow$ Possible extension into detector parameters
  $\rightarrow$ needs well organized volume hierarchy or at least naming convention of volumes to be able to run somewhat automatic
- General extension of GEAR for missing detector description, e.g. forward tracking $\rightarrow$ needs input from reconstruction algorithm requirements
- Extension to service new clients, e.g. more advanced tracking code or misalignment studies
- Goal: single source of geometry information

# What's Next?

## New extensions require

- Input from user community to define sub-detectors
- Interest to use and test new features

## Generic Geometry Description?

- What should it provide?
- Different abstraction levels?
- Clients (reconstruction, analysis etc.) that actually use and need something beyond GEAR
- Time scale?
- Who is interested?
- Who will actually contribute?