GEAR

LCIO

MARLIN

Geometry TOOLKIT

First Proposal for Extensions

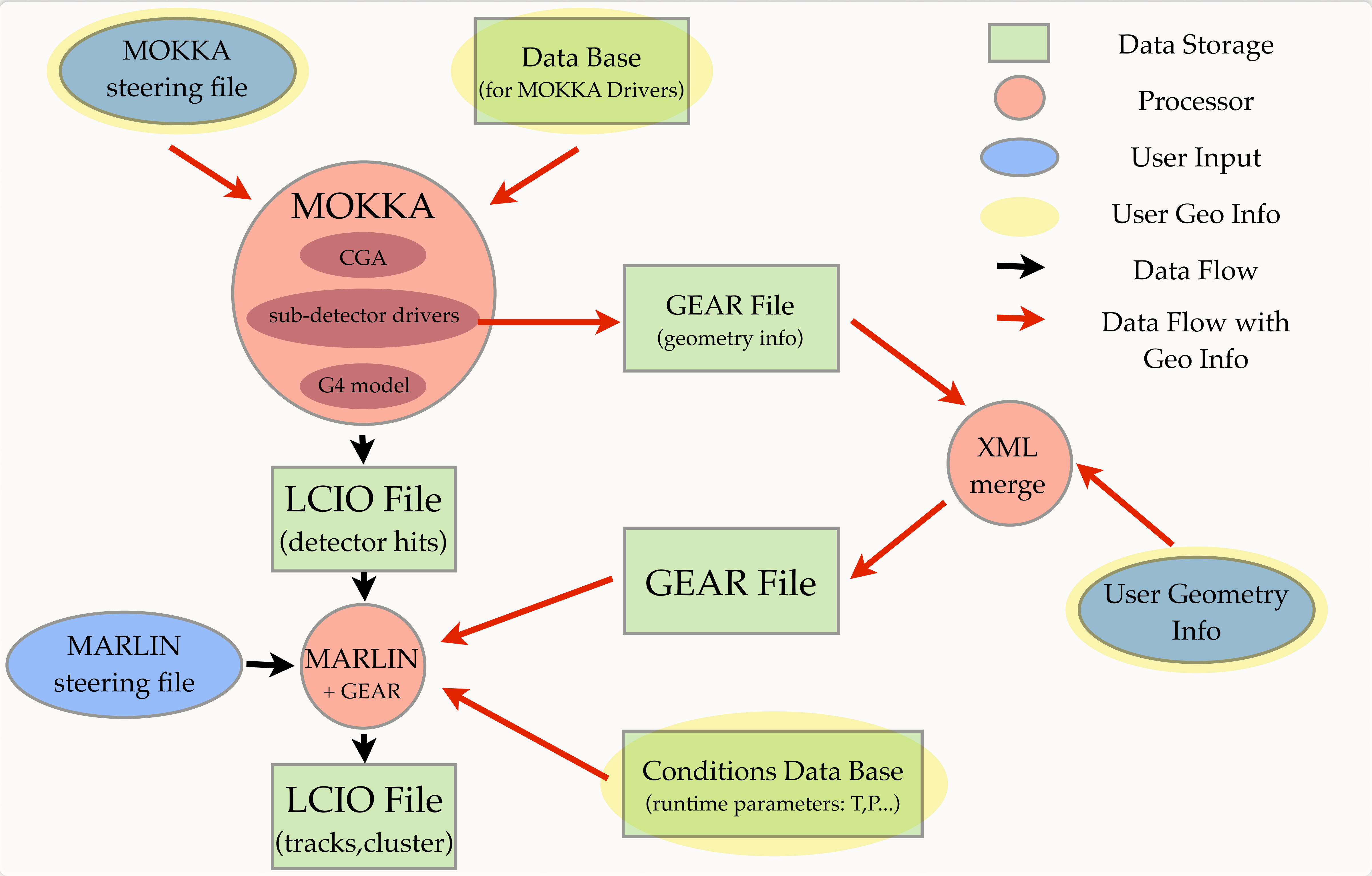
Astrid Muennich (CERN), ILD Software Meeting 27.1.10

MOKKA/
GEANT4

GDML

ROOT

SOFTWARE OVERVIEW



MOKKA steering file

Data Base (for MOKKA Drivers)

MOKKA

CGA

sub-detector drivers

G4 model

GEAR File (geometry info)

XML merge

User Geometry Info

GEAR File

LCIO File (detector hits)

MARLIN steering file

MARLIN + GEAR

Conditions Data Base (runtime parameters: T,P...)

LCIO File (tracks,cluster)

🟩 Data Storage

● Processor

⬭ User Input

⬭ User Geo Info

→ Data Flow

→ Data Flow with Geo Info

PROBLEMS

- No central place for geometry information:

→ Geometry can be changed in various places (inconsistencies)

→ Changing geometry requires changes in C++ code, XML, DB..

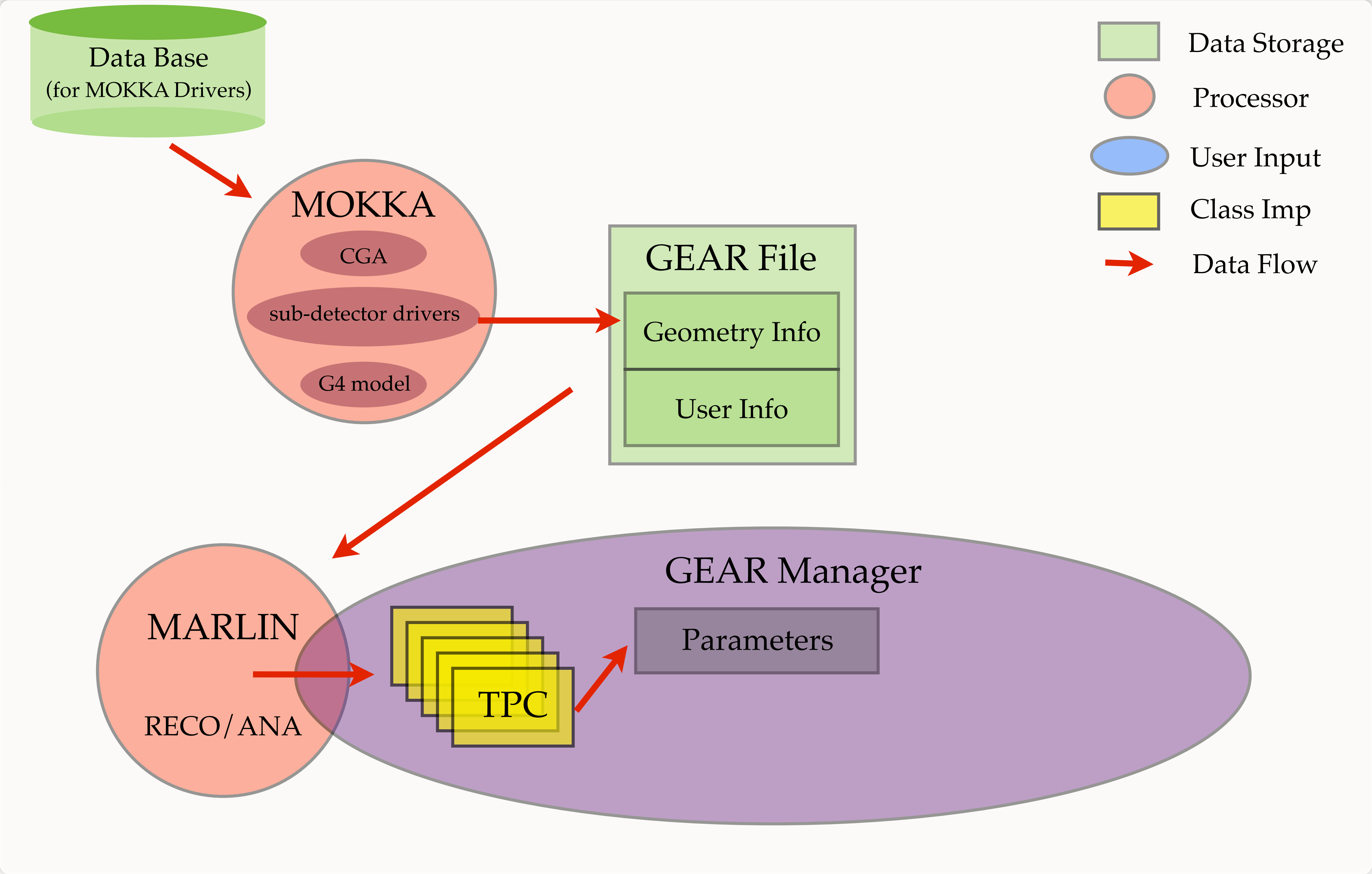
- No representation of geometrical volumes for complex questions, e.g. radiation length, misalignment, no information for coordinate transformation (global ↔ local) etc...

- No easy visualization at later stage (after MOKKA)

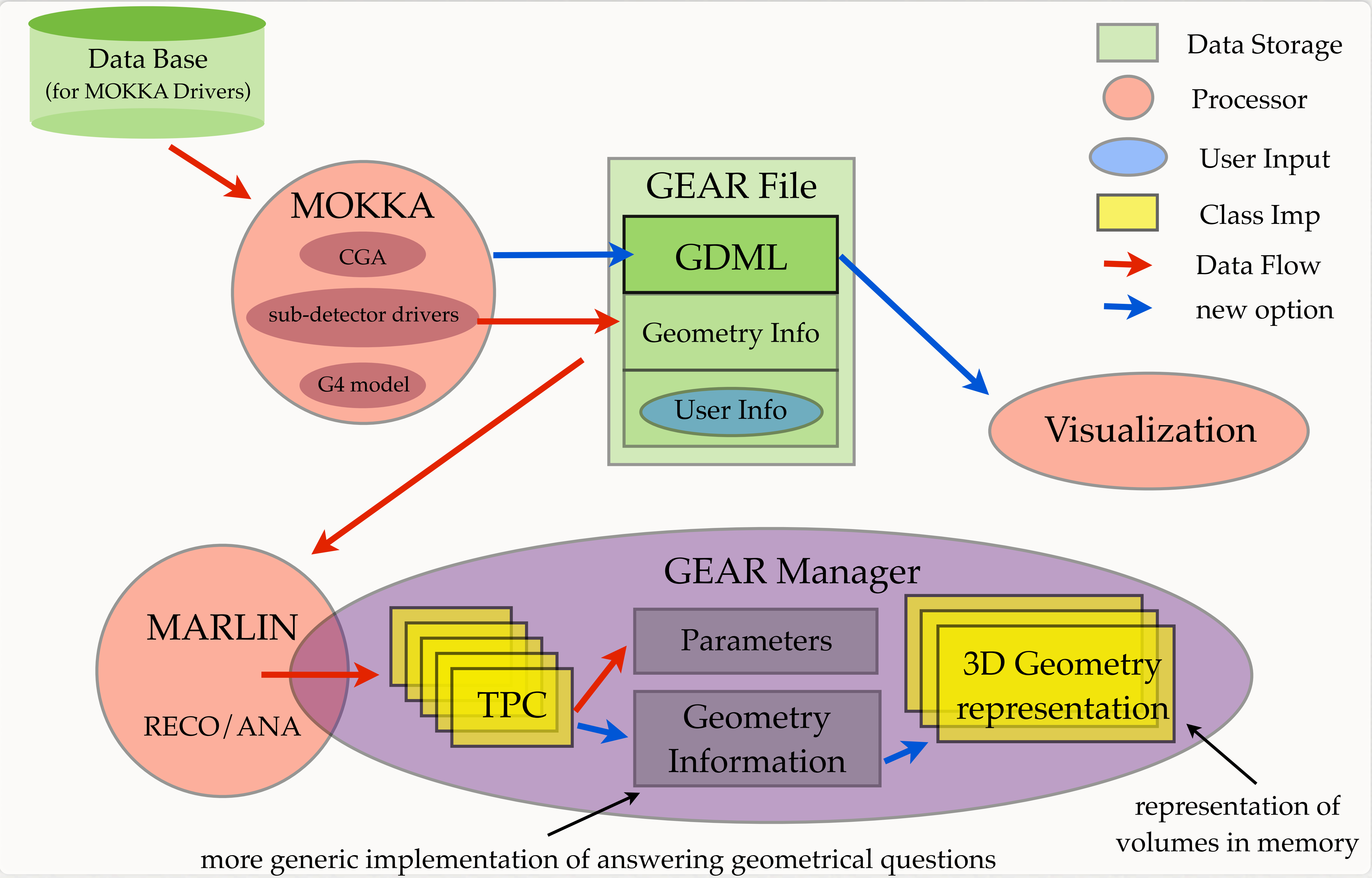
- GEAR interface needs extension, too many “userparameters” written from MOKKA

- software maintenance aspects: e.g. MOKKA code ‘contaminated with GEAR’

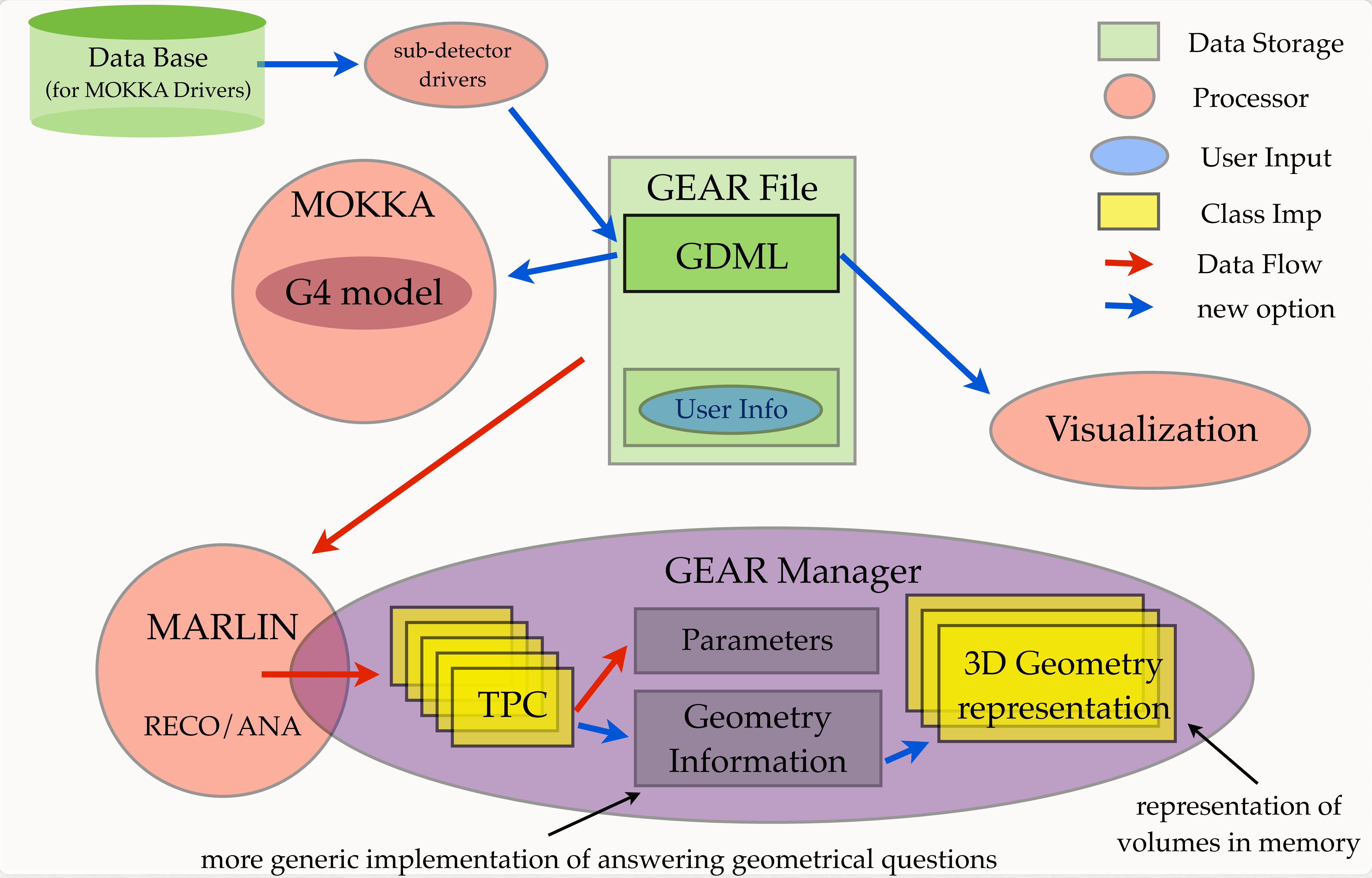
GEOMETRY INFO



GEOMETRY: EXTENSION 1



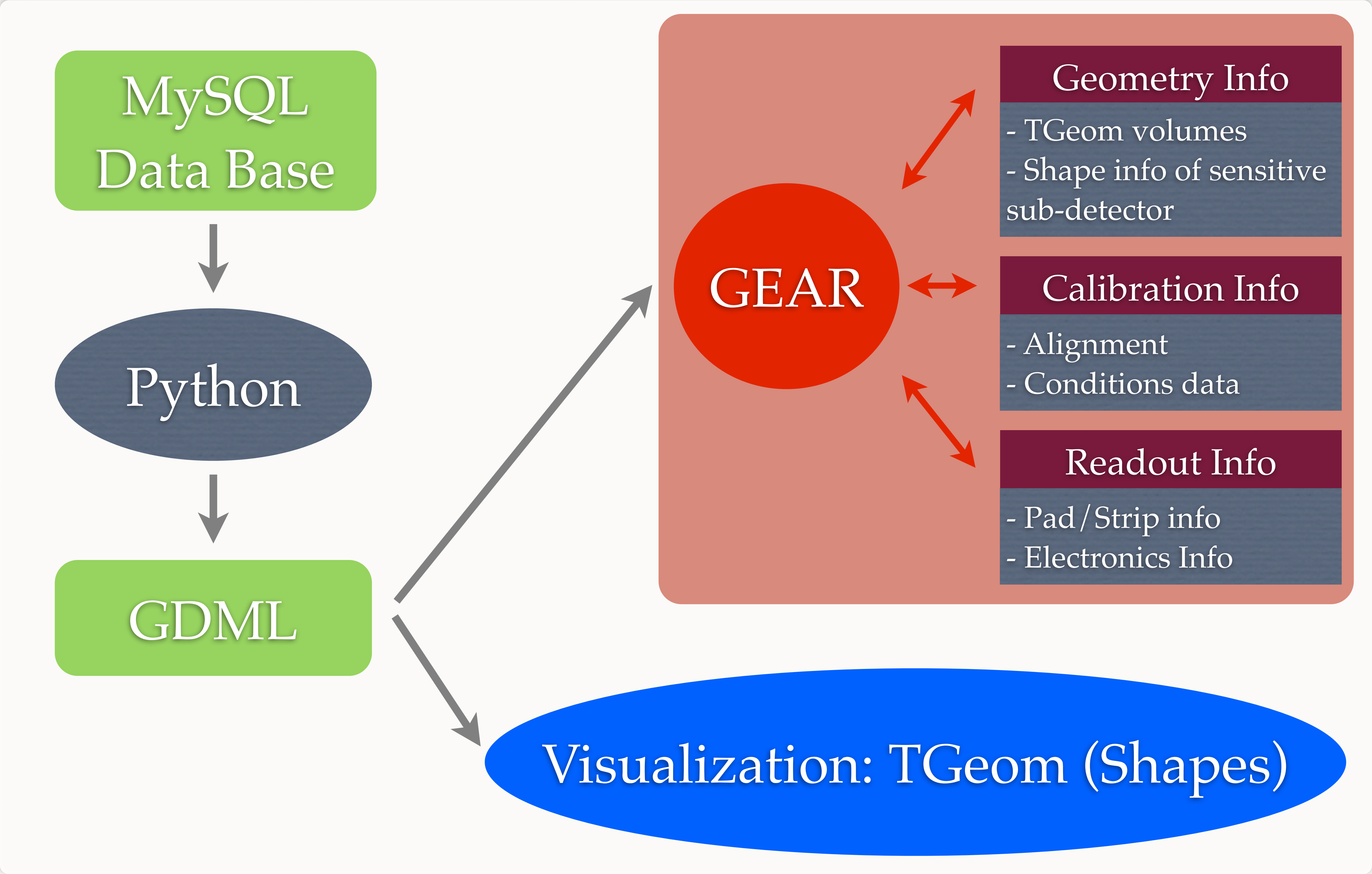
GEOMETRY: EXTENSION 2



representation of volumes in memory

more generic implementation of answering geometrical questions

PROTOTYPE CHAIN



MySQL Data Base

Python

GDML

GEAR

Geometry Info

- TGeom volumes
- Shape info of sensitive sub-detector

Calibration Info

- Alignment
- Conditions data

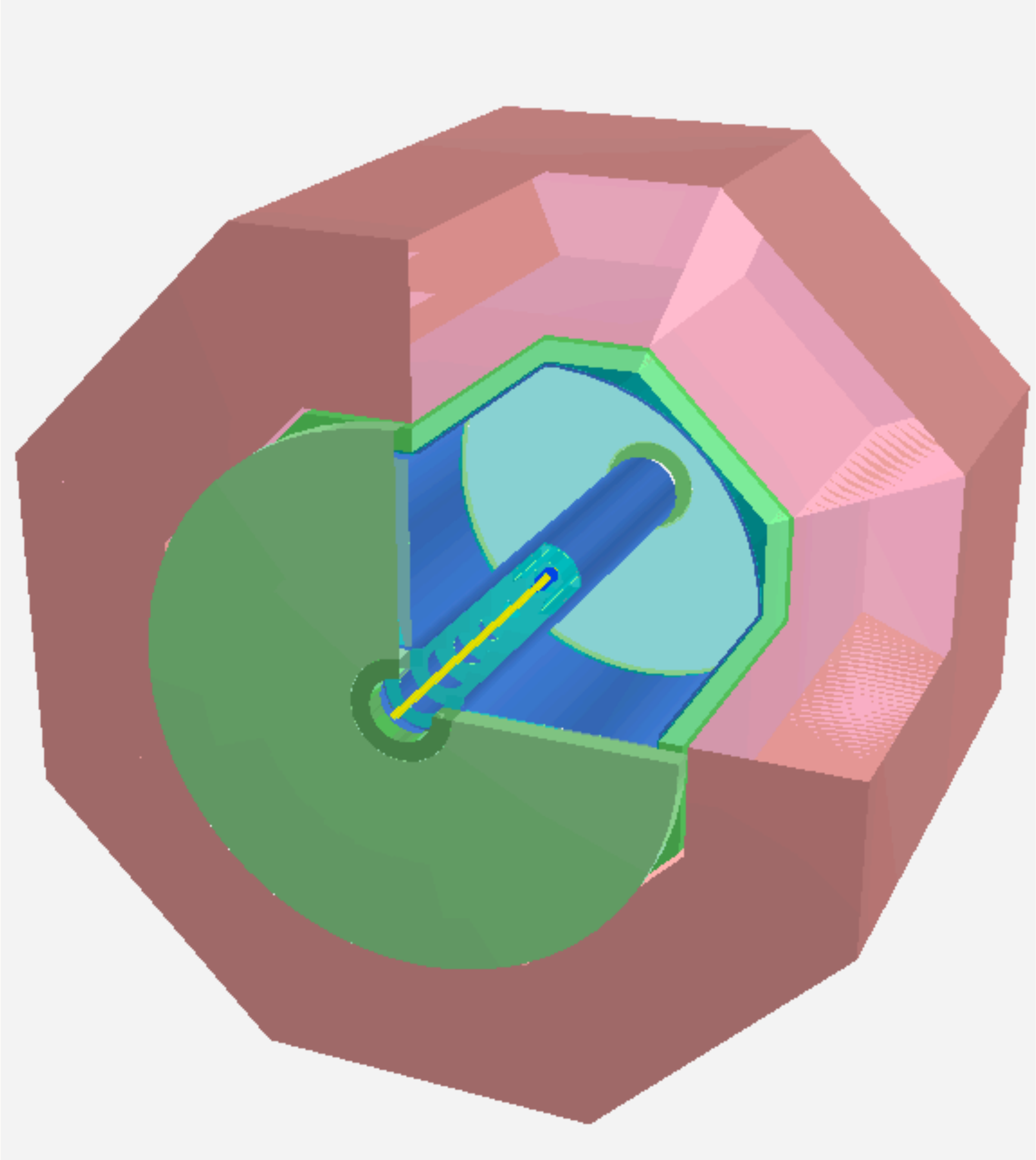
Readout Info

- Pad/Strip info
- Electronics Info

Visualization: TGeom (Shapes)

TGEOMETRY IN ROOT

- Large collection of shapes
- Navigation in volumes
- Visualization: OpenGL
- TGDMLParse reads in GDML file and generates a volume representation
- ...



SO FAR...

- Extended GEAR:
    - Read in GDML file using TGDMLParse
    - Added functionality for TPCParameter, e.g. radius is obtained from volume tree
    - Extension of interface is fully backward compatible, changes only visible in GEAR xml file   (see next slide)
- Toy python script to emulate MOKKA drivers to generate GDML directly

OLD GEAR XML FILE

```
<gear>
  <global detectorName="D09" />
  <!--Gear XML file automatically created with GearXML::createXMLFile ....-->
  <BField type="ConstantBField" x="0.0" y="0.0" z="4.0"/>
  <detectors>
    <detector name="TPC" geartype="TPCParameters">
      <driftVelocity value="0.000000000e+00" />
      <maxDriftLength value="1.967500000e+03" />
      <readoutFrequency value="0.000000000e+00" />
      <PadRowLayout2D type="FixedPadSizeDiskLayout" rMin="3.710000000e+02" .......
      <parameter name="TPCGasProperties_RadLen" type="double" value="1.155205825e+05" />
      <parameter name="TPCGasProperties_dEdx" type="double" value="2.669216431e-07" />
      <parameter name="TPCWallProperties_RadLen" type="double" value="8.896320560e+01" />
      <parameter name="TPCWallProperties_dEdx" type="double" value="4.328948956e-04" />
      <parameter name="tpcInnerRadius" type="double" value="3.050000000e+02" />
      <parameter name="tpcInnerWallThickness" type="double" value="1.160000000e+00" />
      <parameter name="tpcIonPotential" type="double" value="3.200000000e-08" />
      <parameter name="tpcOuterRadius" type="double" value="1.580000000e+03" />
      <parameter name="tpcOuterWallThickness" type="double" value="1.510000000e+00" />
     </detector>
   </detectors>
</gear>
```

Userparameter written from MOKKA:

NO DEFINED INTERFACE !!!

NEW GEAR XML FILE

```
<gear>
   <global detectorName="D09"  />
   <!--Gear XML file automatically created with GearXML::createXMLFile ....-->
   <GDMLFile name="GDMLOutput-TPC.xml"/>
   <BField type="ConstantBField" x="0.0" y="0.0" z="4.0"/>
   <detectors>
      <detector name="TPC" geartype="TPCParameters">
         <volumeref name="TPCSensitiveLog"/>
         <PadRowLayout2D type="FixedPadSizeDiskLayout" rMin=...../>
      </detector>
   </detectors>
</gear>
```

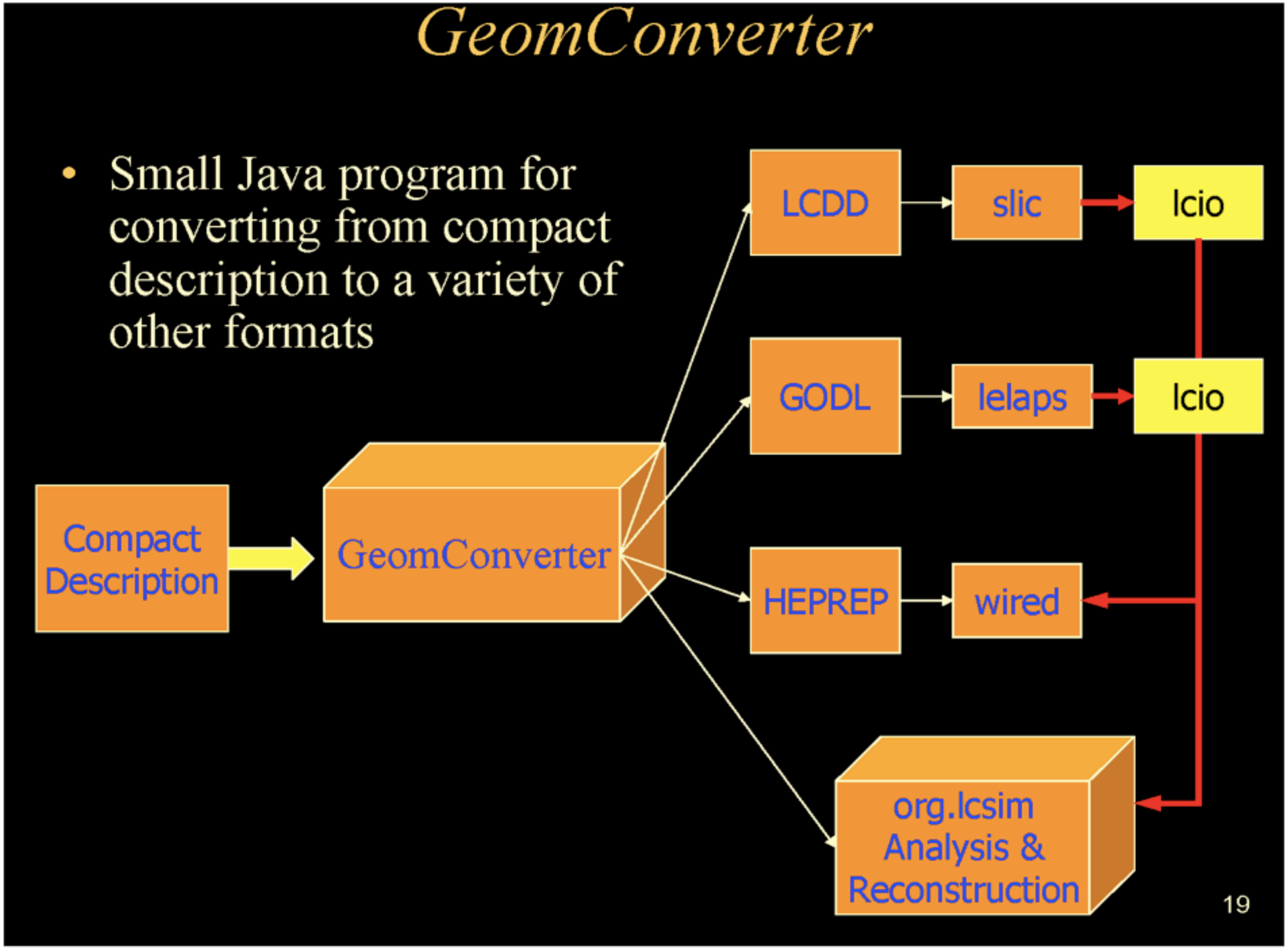
new

- GDML file takes care of all geometry and material information
- Volume name defines the name of the sub-detector in the GDML tree

MANY OPEN QUESTIONS

- Abstraction level from GDML to GEAR ?
- How powerful should GEAR be ?
- Changes to MOKKA: How much and when ?
- Need easy way for user to write geometry input
- Different levels of geometry details needed in various stages of the framework
- And many more ....

SIMILAR TO SLIC



GeomConverter

- Small Java program for converting from compact description to a variety of other formats

Compact Description

GeomConverter

LCDD

slic

lcio

GODL

lelaps

lcio

HEPREP

wired

org.lcsim Analysis & Reconstruction

19

SLAC: Jeremy McCormick, Norman Graf, Ron Cassell, Tony Johnson