Mass production with ILCDIRAC
GEAR Extension: 3D Volume Tree

S. Poss[1] A. Münnich[1]

[1]CERN, Switzerland

19/10/2010 / IWLC, Geneva

Part I

Introduction

Aim

- Give an overview of the CERN LCD group activities in software development

Content

- ILCDIRAC: GRID solution for ILC
- GEAR extension

Part II

Mass production with ILCDIRAC

S. Poss

ILCDIRAC

ILCDIRAC is an implementation of DIRAC

- DIRAC is the GRID solution for the LHCb community, used for mass production of MC simulations events and real data reconstruction and analysis
- Designed to simplify user access to the GRID and overcome GRID middleware deficiencies
- Based on pilot job principles

Mass production with ILCDIRAC (1)

Aim:

- Provide production system for the CDR physics studies

Constrains:

- Minimal user effort to :
  1. generate;
  2. simulate;
  3. reconstruct

  the data needed
- Register the data produced by each step and keep tracks of links
- Must work for both detector concepts (ILD and SiD)

Mass production with ILCDIRAC (2)

Idea:

- Submit automatically jobs based on request

Request:

- Channel;
- Number of events / luminosity;
- Input files

Implementation

Principles:

- Divide in independant steps that can be ran one after the other

Steps:

- Whizard: generate the events according to desired channel and number of events. Produces STDHEP files
- Mokka / SLIC: simulate the detector behavior using given detector description. Produces single SLCIO file
- Marlin / LCSIM: reconstruct the events. Produces several SLCIO files
- Uploading of data: stores the data in at least 2 places (CERN and CC-IN2P3), logs are at CERN
- Register in FileCatalog: set metadata flags for each file

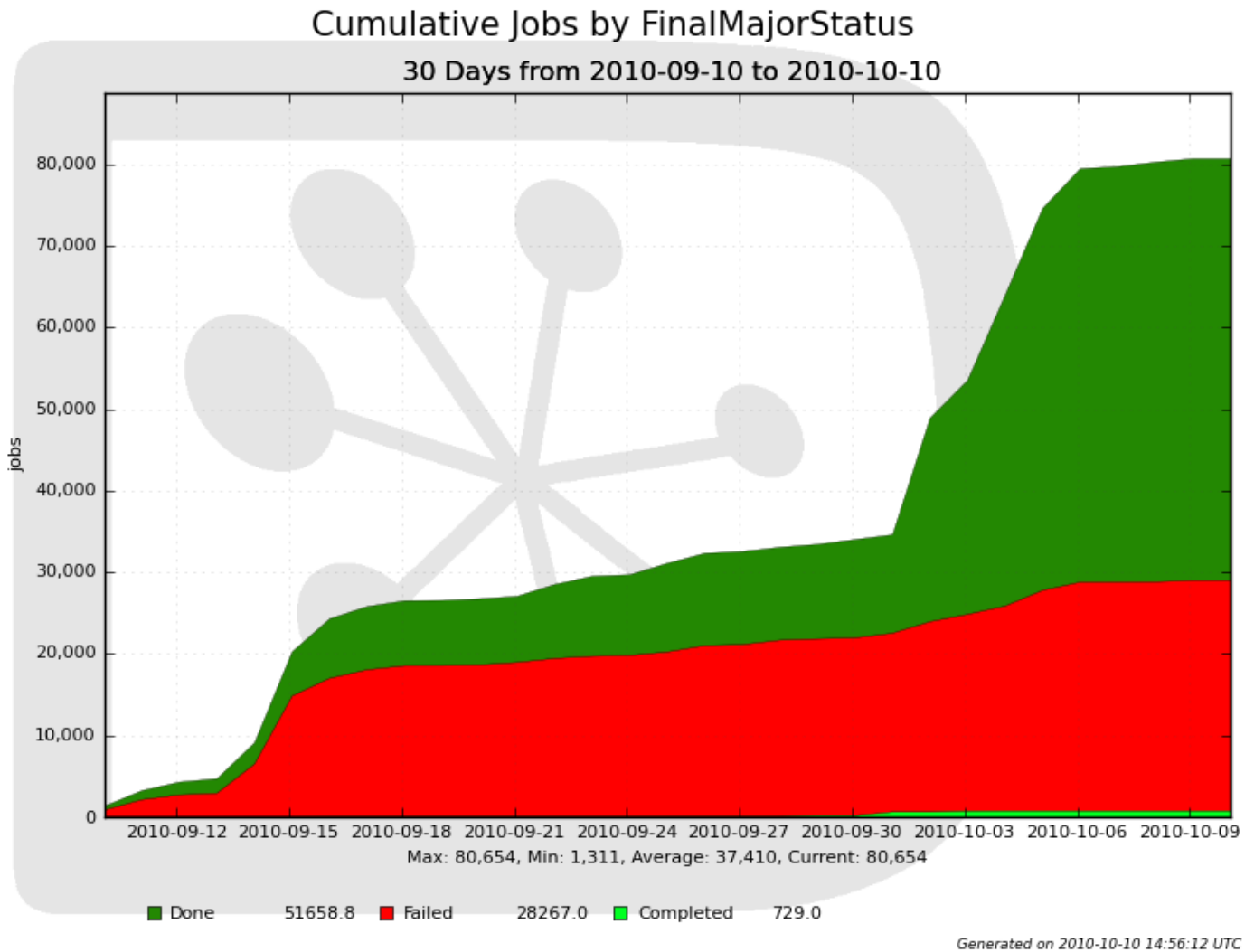
Results

Full production chain available and working !

Individual steps available to users jobs, and already massively used

Please find more info at
https://twiki.cern.ch/twiki/bin/view/CLIC/DiracUsage

Results

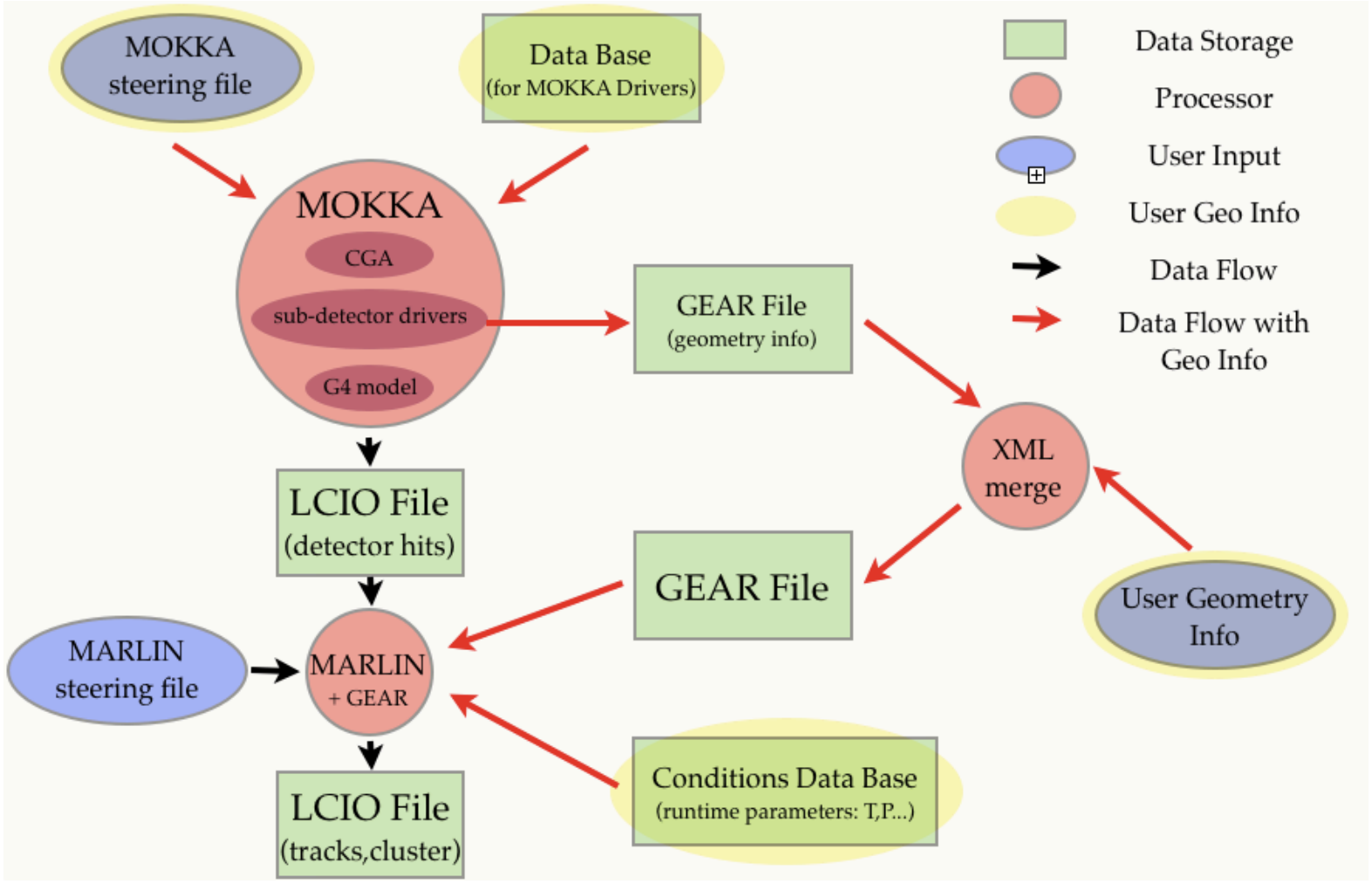

Cumulative Jobs by FinalMajorStatus
30 Days from 2010-09-10 to 2010-10-10
jobs
80,000
70,000
60,000
50,000
40,000
30,000
20,000
10,000
0
2010-09-12
2010-09-15
2010-09-18
2010-09-21
2010-09-24
2010-09-27
2010-09-30
2010-10-03
2010-10-06
2010-10-09
Max: 80,654, Min: 1,311, Average: 37,410, Current: 80,654
Done 51658.8
Failed 28267.0
Completed 729.0
Generated on 2010-10-10 14:56:12 UTC

Part III

GEAR Extension

A. Münnich

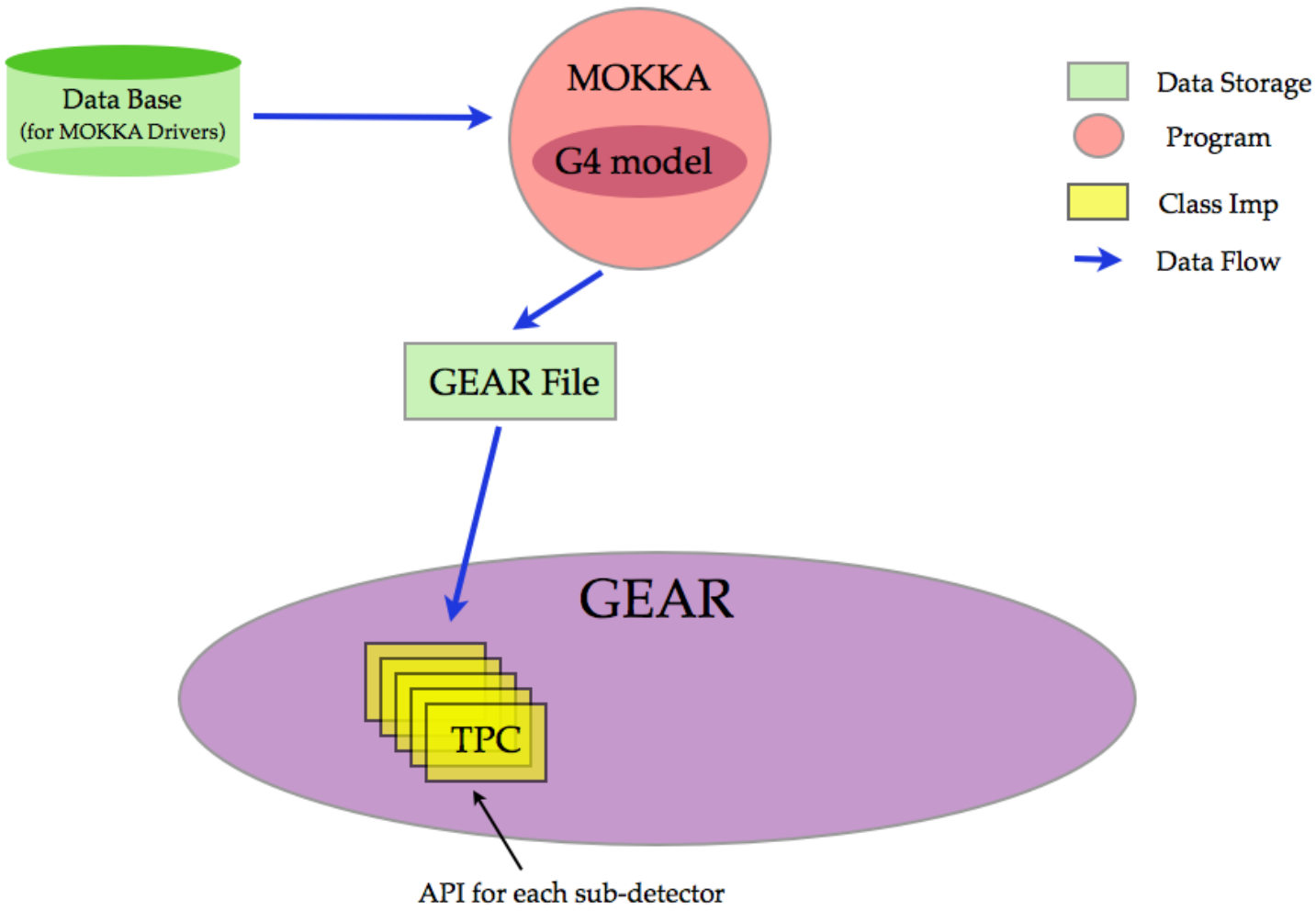
Current status of geometry description



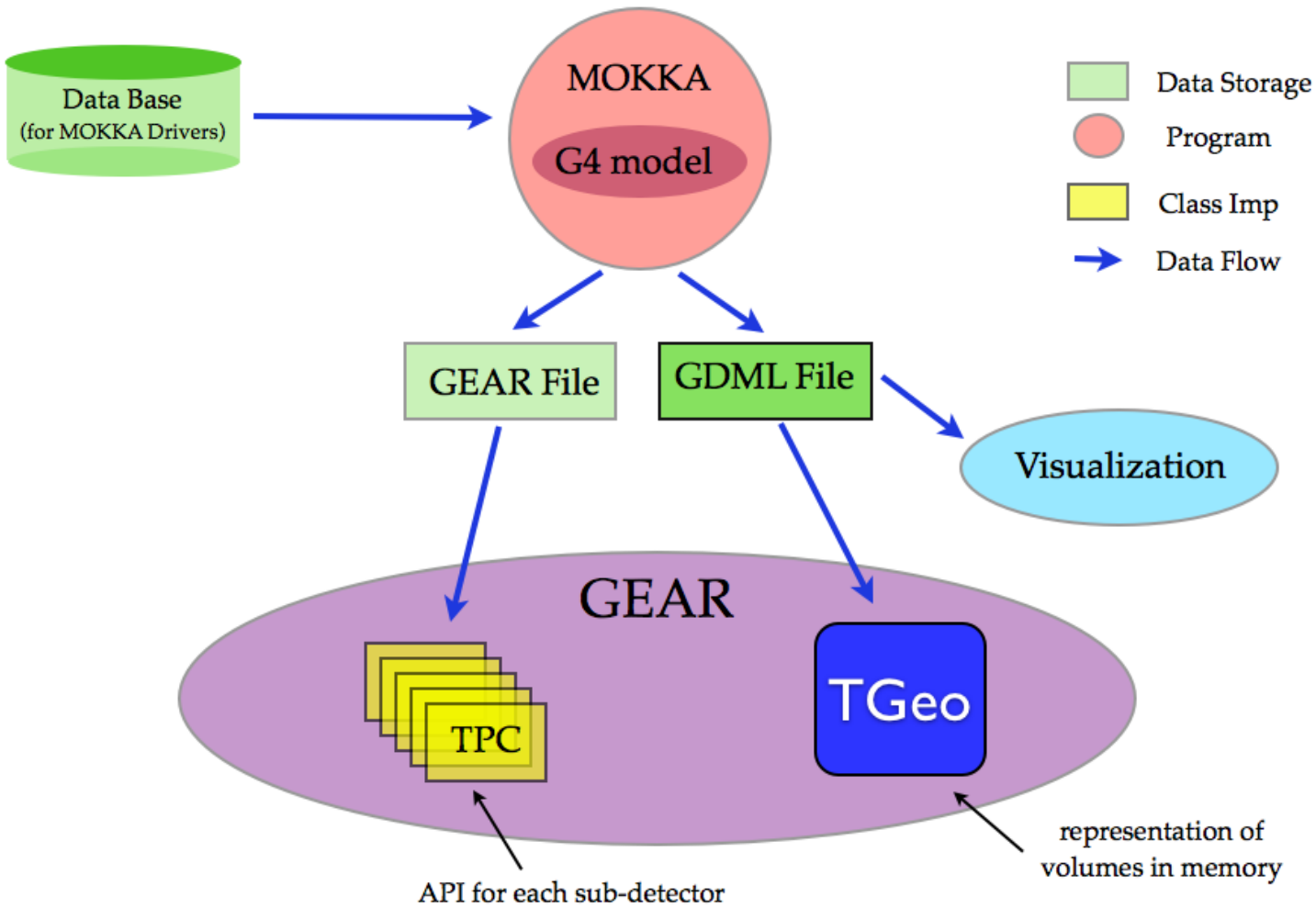
Motivation for new Geometry System

- 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

Geometry Information Flow



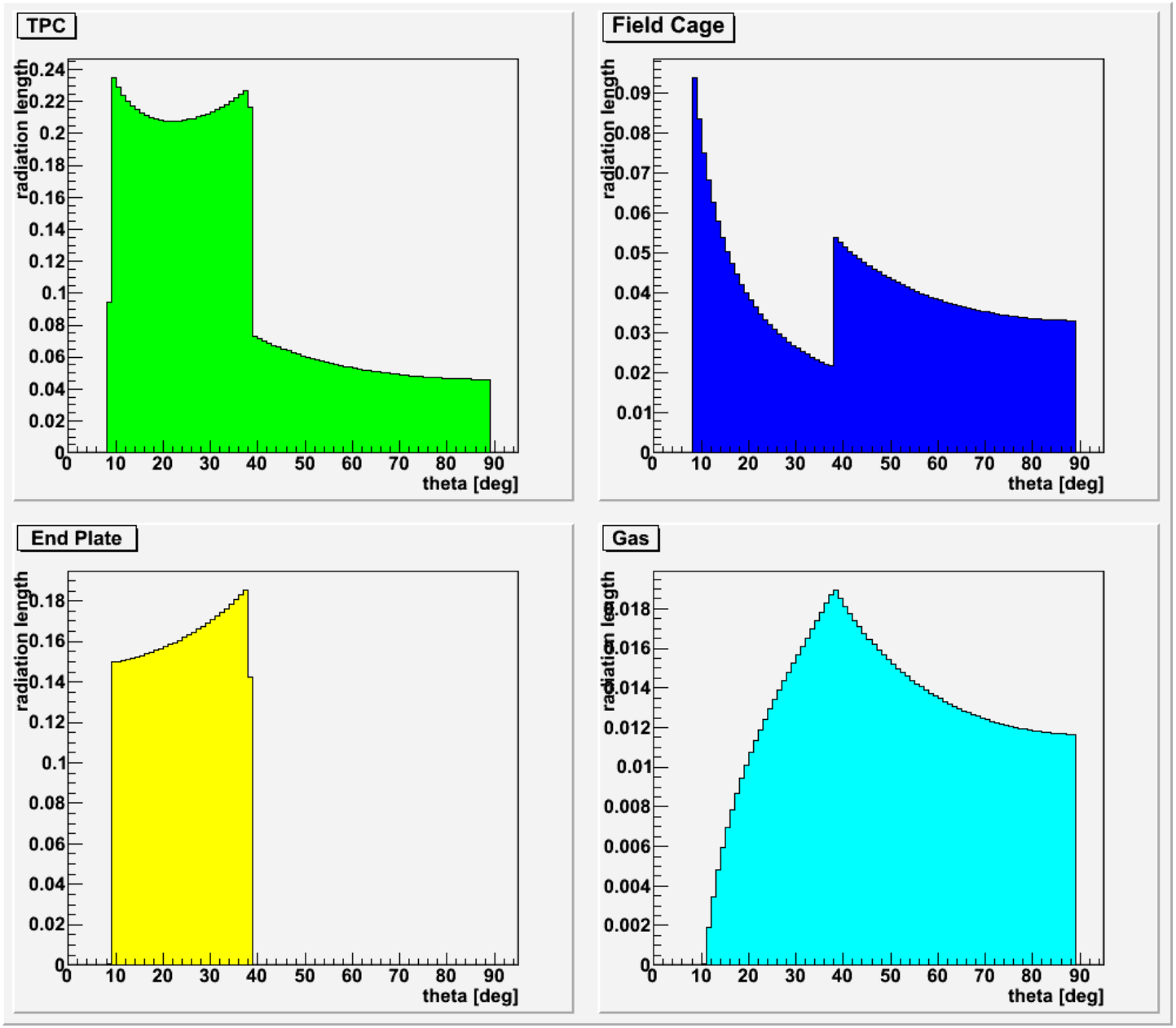
Geometry Information Flow



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



Summary

Allows a better geometry handling and description, plus gives access to information not easily available before

Users need to get latest version of ilc-soft to benefit

One extra line in GEAR xml file:

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

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

Part IV

Conclusion

Conclusion

The group exclusive activities in software development are:

- ILCDIRAC: provides user friendly access to grid resources
- GEAR extension: better geometry interface

Many other activities are not described here: detector model implementation, sofware benchmarking, Pandora development, etc.