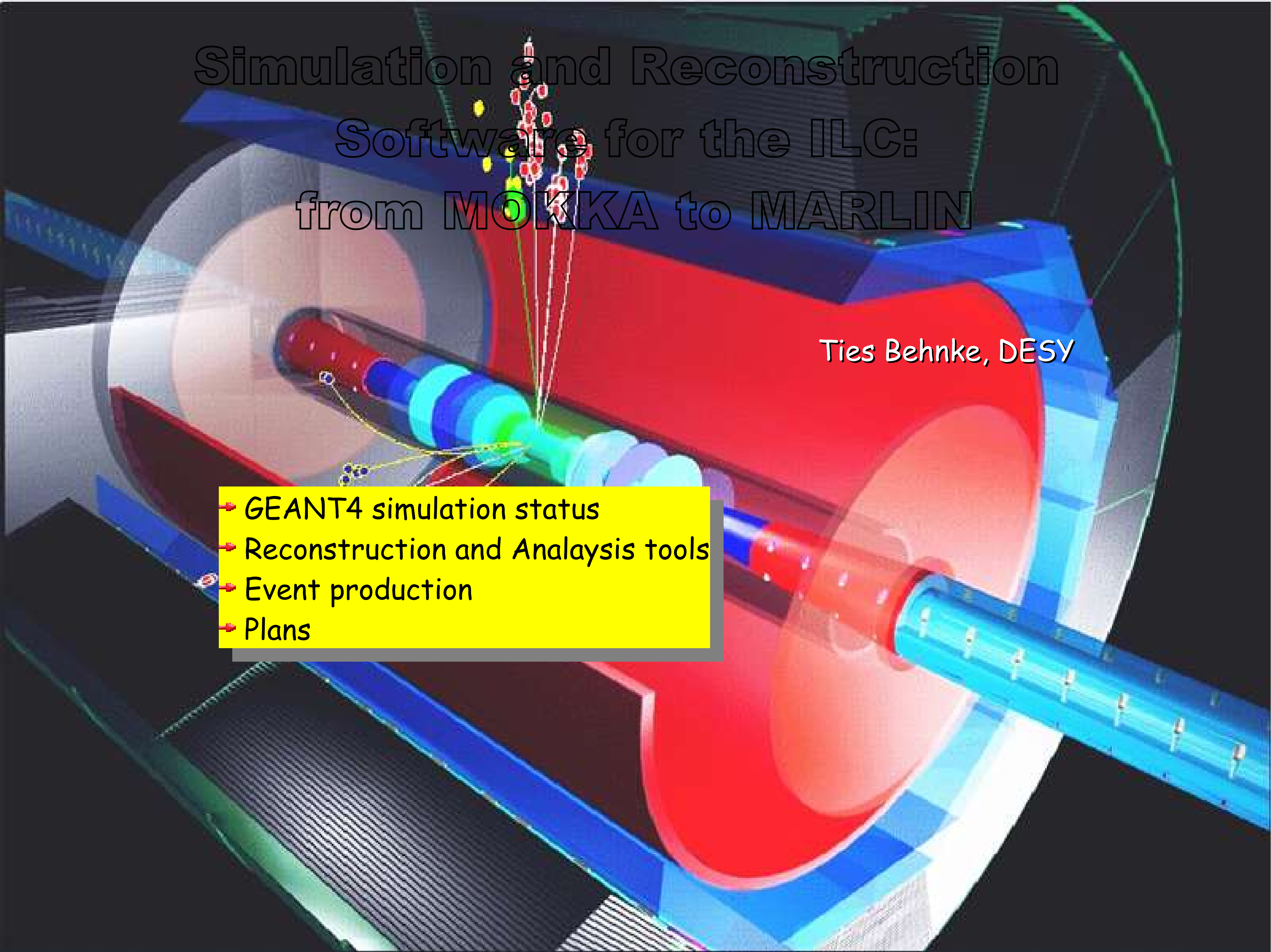
Simulation and Reconstruction Software for the ILC: from MOKKA to MARLIN



Ties Behnke, DESY

- GEANT4 simulation status
- Reconstruction and Analysis tools
- Event production
- Plans

The Software

Software developed within ECFA study framework

- adopted widely within LDC
- however not confined to LDC: goal is concept independent software

Boundary conditions:

- data model and consitency: based on LCIO
- core language C++
- minimize dependence on external packages
- modular approach
- „keep it simple“

Software Packages

	Description	Detector	Language	IO-Format	Region
Simdet	fast Monte Carlo	TeslaTDR	Fortran	StdHep/LCIO	EU
SGV	fast Monte Carlo	simple Geometry, flexible	Fortran	None (LCIO)	EU
Lelaps	fast Monte Carlo	SiD, flexible	C++	SIO, LCIO	US
Mokka	full simulation – Geant4	TeslaTDR, LDC, flexible	C++	ASCI, LCIO	EU
Brahms-Sim	Geant3 – full simulation	TeslaTDR	Fortran	LCIO	EU
SLIC	full simulation – Geant4	SiD, flexible	C++	LCIO	US
LCDG4	full simulation – Geant4	SiD, flexible	C++	SIO, LCIO	US
Jupiter	full simulation – Geant4	JLD (GDL)	C++	Root (LCIO)	AS
Brahms-Reco	reconstruction framework (most complete)	TeslaTDR	Fortran	LCIO	EU
Marlin	reconstruction and analysis application framework	Flexible	C++	LCIO	EU
hep.lcd	reconstruction framework	SiD (flexible)	Java	SIO	US
org.lcsim	reconstruction framework (under development)	SiD (flexible)	Java	LCIO	US
Jupiter-Satelite	reconstruction and analysis	JLD (GDL)	C++	Root	AS
LCCD	Conditions Data Toolkit	All	C++	MySQL, LCIC	EU
GEAR	Geometry description	Flexible	C++ (Java?)	XML	EU
LCIO	Persistency and datamodel	All	Java, C++, Fortran	-	AS,EU,US
JAS3/WIRED	Analysis Tool / Event Display	All	Java	xml,stdhep, heprep,LCIO,	US,EU

Software Packages

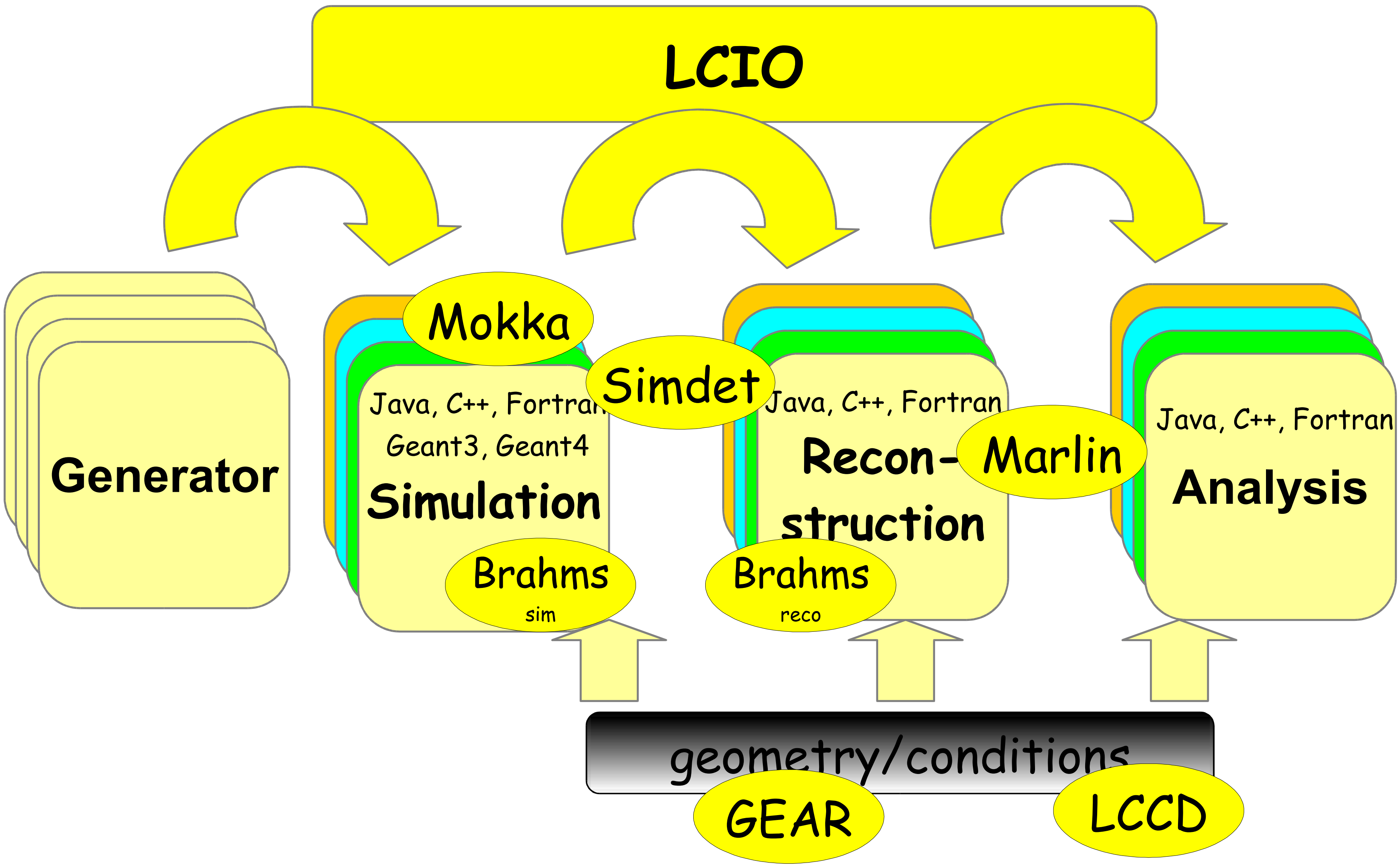
	Description	Detector	Language	IO-Format	Region
Simdet	fast Monte Carlo	TeslaTDR	Fortran	StdHep/LCIO	EU
SGV	fast Monte Carlo	simple Geometry, flexible	Fortran	None (LCIO)	EU
Lelaps	fast Monte Carlo	SiD, flexible	C++	LCIO	US
Mokka	full simulation – Geant4	TeslaTDR, LDC, flexible		LCIO	EU
Brahms-Sim	Geant3 – full simulation	TeslaTDR		O	EU
SLIC	full simulation – Geant4	SiD			US
LCDG4	full simulation – Geant4			O	US
Jupiter	full simulation – Ge				AS
Brahms-Reco	reconstructi (r				EU
Marlin	r			O	EU
h				SIO	US
org.			Java	LCIO	US
Jupiter-		DL)	C++	Root	AS
LCC		All	C++	MySQL, LCIO	EU
GEAR		Flexible	C++ (Java?)	XML	EU
LCIO	d datamodel	All	Java, C++, Fortran	-	AS,EU,US
JAS3/WIRED	alysis Tool / Event Display	All	Java	xml,stdhep, heprep,LCIO,	US,EU

clear need to do some further standardisation:

too much duplication and wasted personpower

Software

Software architechture



Simulation tools: Simdet, Brahms

- **SIMDET**

writes LCIO

  - parameterized fast Monte Carlo (f77)
  - tracks + cov. matrix and clusters
  - hard coded geometry: TESLA TDR Detector

- **Brahms**

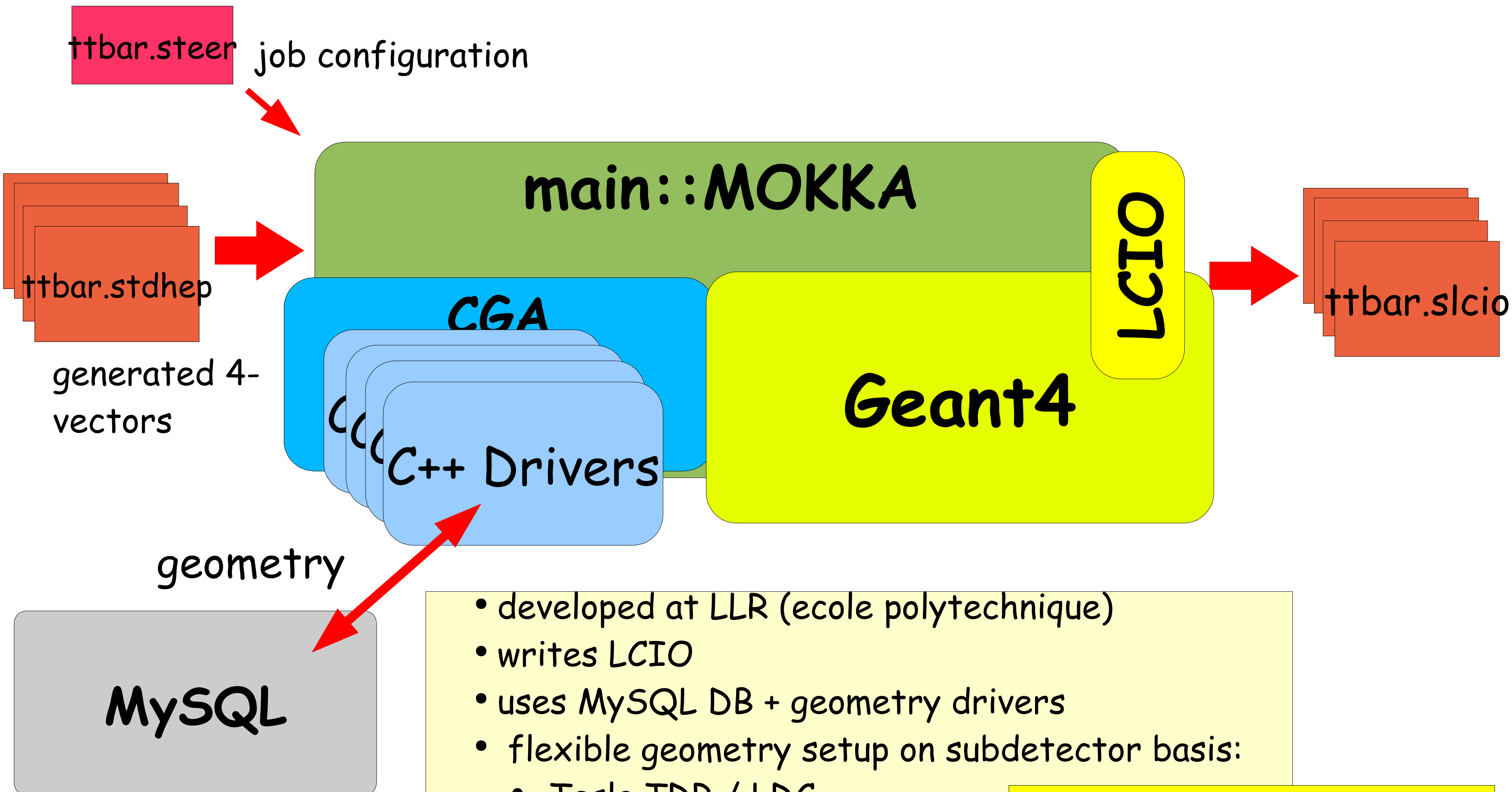
reads + writes LCIO

  - GEANT3 full simulation (f77)
  - hard coded geometry: TESLA TDR Detector
  - full standalone reconstruction part (pflow)
    - tracking based on LEP reconstruction code

for download (cvs web interface)
and more information:
http://www-zeuthen.desy.de/linear_collider

or http://ilcsoft.desy.de software portal

Mokka overview



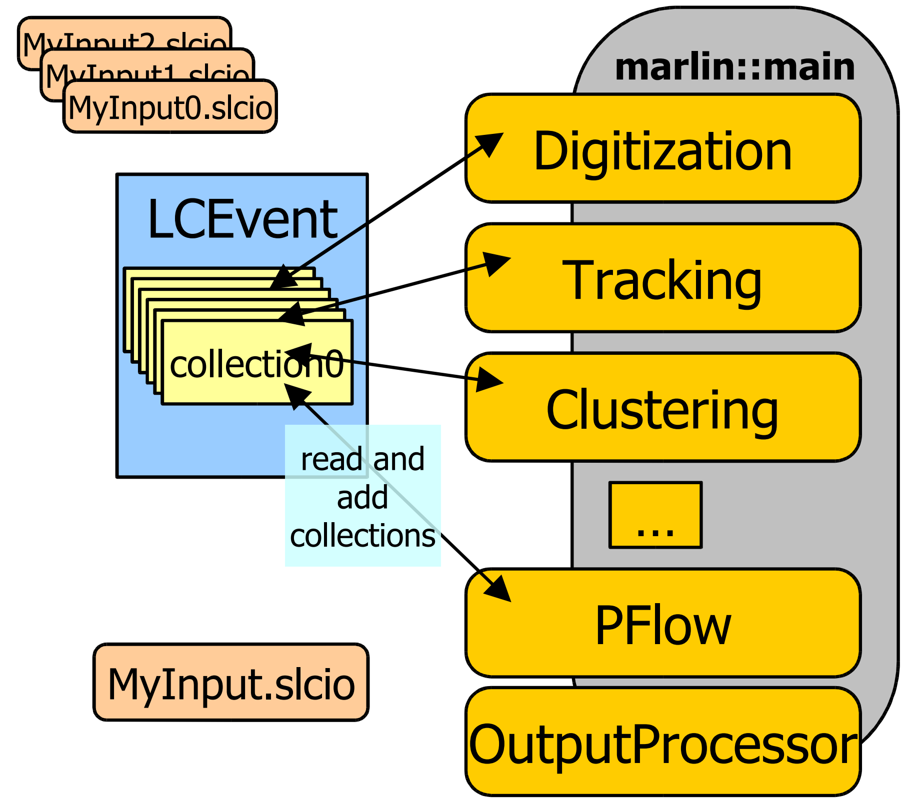
- developed at LLR (ecole polytechnique)
- writes LCIO
- uses MySQL DB + geometry drivers
- flexible geometry setup on subdetector basis:
  - Tesla TDR / LDC
  - SiD
  - Calice testbeam prototypes
- used in European/ LDC study

- new feature:
  scalable detector models
  -> detector optimization !!

Marlin

Modular**A**nalysis & **R**econstruction for the **L I N**ear Collider

- modular C++ **application framework** for the analysis and reconstruction of LCIO data
- uses LCIO as transient data model
- software modules called Processors
- provides main program !
- provides simple user steering:
  - program flow (active processors)
  - user defined variables
    - per processor and global
  - input/output files



Marlin – example XML steering

```
- <marlin>
  - <execute>
      <processor name="MyAIDAProcessor"/>
      <processor name="MyEventSelection"/>
    - <if condition="MyEventSelection">
        <group name="Tracking"/>
        <processor name="MyClustering"/>
        <processor name="MyPFlow"/>
        <processor name="MyLCIOOutputProcessor"/>
      </if>
    </execute>
  - <global>
      <parameter name="LCIOInputFiles"> simjob.slcio </parameter>
      <parameter name="MaxRecordNumber" value="5001"/>
      <parameter name="SupressCheck" value="false"/>
    </global>
  - <processor name="MyLCIOOutputProcessor" type="LCIOOutputProcessor">
      <parameter name="LCIOOutputFile" type="string">outputfile.slcio </parameter>
      <parameter name="LCIOWriteMode" type="string">WRITE_NEW</parameter>
    </processor>
  - <group name="Tracking">
      <parameter name="NTPCLayers" value="200"/>
      <processor name="MyTrackfinder" type="Trackfinder"/>
    - <processor name="MyTrackfitter" type="Trackfitter">
        <parameter name="Algorithm" value="DAF"/>
      </processor>
    </group>
   <!-- ... -->
  </marlin>
```

- ActiveProcessors replaced by <execute>…</execute> section
- Reconstruct only events that pass the event selection

- Parameters defined as content of <parameter/> tag or as its value attribute

- Processors can be enclosed by <group/> tag
- Parameters in <group/> joined by all processors

Gear

GEometry **A**PI for **R**econstruction


```
- <gear>
  - <!--
     Example XML file for GEAR describing the LDC detect
    -->
  - <detectors>
    - <detector id="0" name="TPCTest" geartype="TPCParameters" type="UNKNOWN" insideTrackingVolume="yes">
        <maxDriftLength value="2500."/>
        <driftVelocity value=""/>
        <readoutFrequency value="10"/>
        <PadRowLayout2D type="FixedPadSizeDiskLayout" rMin="386
        maxRow="200" padGap="0.0"/>
        <parameter name="tpcRPhiResMax" type="double"> 0.16 </pa
        <parameter name="tpcZRes" type="double"> 1.0 </parameter
        <parameter name="tpcPixRP" type="double"> 1.0 </paramete
        <parameter name="tpcPixZ" type="double"> 1.4 </parameter
        <parameter name="tpcIonPotential" type="double"> 0.0000000
      </detector>
    - <detector name="EcalBarrel" geartype="CalorimeterParameters"
        <layout type="Barrel" symmetry="8" phi0="0.0"/>
        <dimensions inner_r="1698.85" outer_z="2750.0"/>
        <layer repeat="30" thickness="3.9" absorberThickness="2.5"
        <layer repeat="10" thickness="6.7" absorberThickness="5.3"
      </detector>
    - <detector name="EcalEndcap" geartype="CalorimeterParameters
        <layout type="Endcap" symmetry="2" phi0="0.0"/>
        <dimensions inner_r="320.0" outer_r="1882.85" inner_z="28
        <layer repeat="30" thickness="3.9" absorberThickness="2.5"
        <layer repeat="10" thickness="6.7" absorberThickness="5.3"
      </detector>
    </detectors>
</gear>
```


compatible with US compact format=

- well defined geometry definition for reconstruction that
  - is flexible w.r.t different detector concepts
  - has high level information needed for reconstruction
  - provides access to material properties - planned
- abstract interface (a la LCIO)
- concrete implementation based on XML files
  - and Mokka-CGA – planned

GEAR Interface

isolate the geometry information in the reconstruction etc from its source:

it does not matter how the information is obtained

(at the moment CGA is used -> GEANT4
replacement through more light-weight system is desirable)

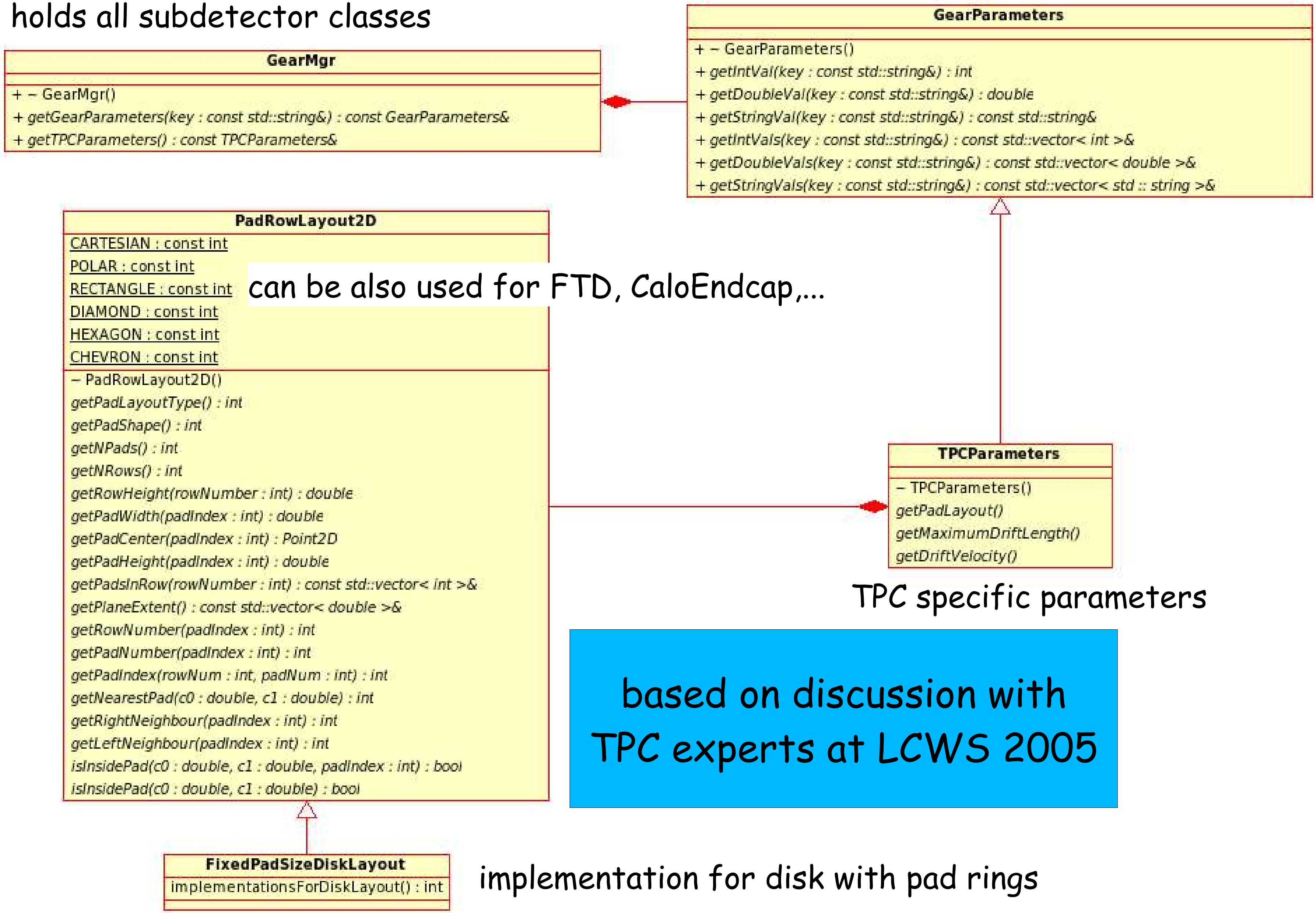
„meta information“: provide through xml file
(modelled after the input files used for SLIC and friends)

GEAR is an abstract interfact which should make the exchange of code simpler and detector independent

GEAR – TPC description

holds all subdetector classes

named parameters for additional attributes



can be also used for FTD, CaloEndcap,...

TPC specific parameters

implementation for disk with pad rings

based on discussion with
TPC experts at LCWS 2005

GEAR – material properties

GearDistanceProperties

```
– GearDistanceProperties()
getMaterialNames(p0 : const Point3D&, p1 : const Point3D&) : const std::vector< std :: string >&
getMaterialThicknesses(p0 : const Point3D&, p1 : const Point3D&) : const std::vector< double >&
getNRadlen(p0 : const Point3D&, p1 : const Point3D&) : double
getNIntlen(p0 : const Point3D&, p1 : const Point3D&) : double
getBdL(pos : const Point3D&) : double
getEdL(pos : const Point3D&) : double
```

integrated along path:
- straight line or
- true path in B-Field ?

GearPointProperties

```
– GearPointProperties()
getCellID(pos : const Point3D&) : int
getMaterialName(pos : const Point3D&) : const std::string&
getDensity(pos : const Point3D&) : double
getTemperature(pos : const Point3D&) : double
getPressure(pos : const Point3D&) : double
getRadlen(pos : const Point3D&) : double
getIntlen(pos : const Point3D&) : double
getLocalPosition(pos : const Point3D&) : Point3D
getB(pos : const Point3D&) : double
getE(pos : const Point3D&) : double
getListOfLogicalVolumes(pos : const Point3D&) : std::vector< std :: string >
getListOfPhysicalVolumes(pos : const Point3D&) : std::vector< std :: string >
getRegion(pos : const Point3D&) : std::string
isTracker(pos : const Point3D&) : bool
isCalorimeter(pos : const Point3D&) : bool
```

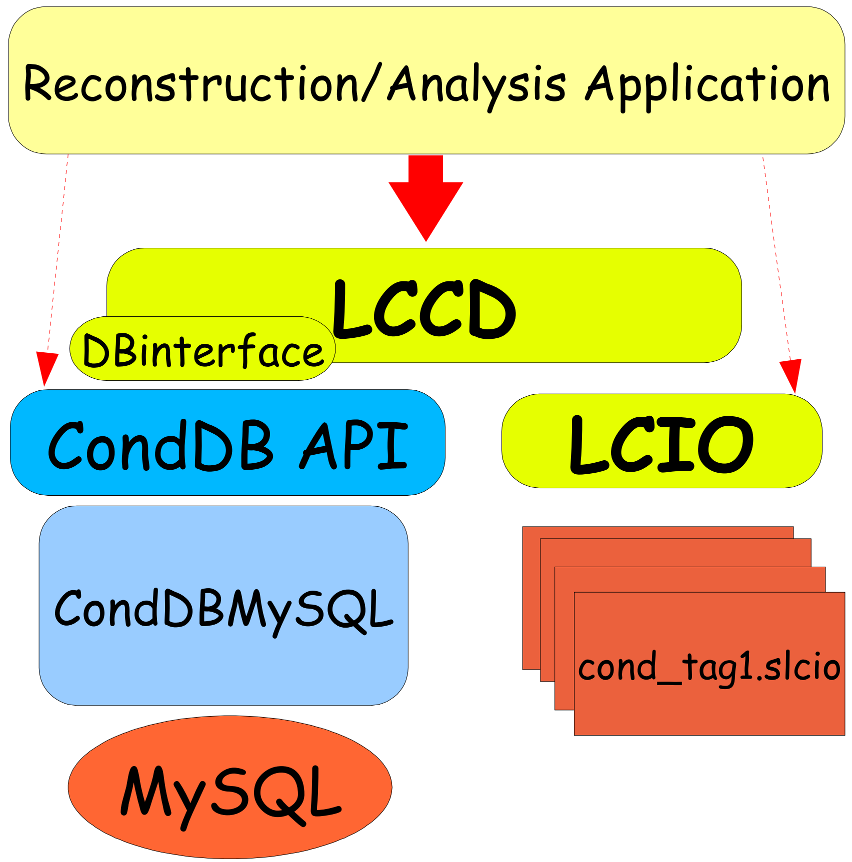
properties at point from geant4 (CGA)

based on discussions at Argonne Simulation Meeting 2004

LCCD

Linear **C**ollider **C**onditions **D**ata Toolkit

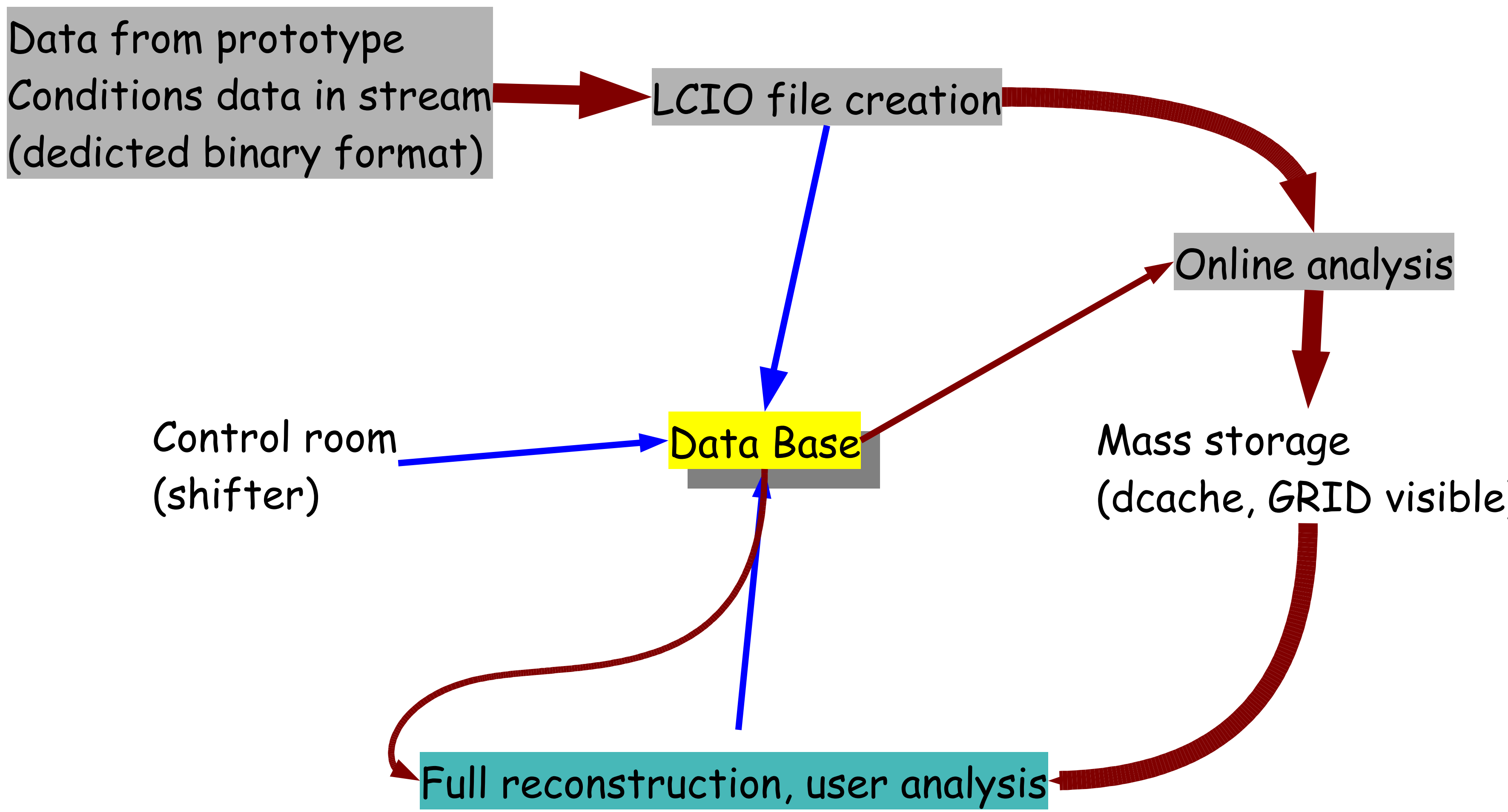
- Reading conditions data
  - from conditions database
  - from simple LCIO file
  - from LCIO data stream
  - from dedicated LCIO-DB file
- **Writing conditions data**
  - tag conditions data
  - Browse the conditions database
    - vertically (all versions for timestamp)
    - horizontally (all versions for tag)



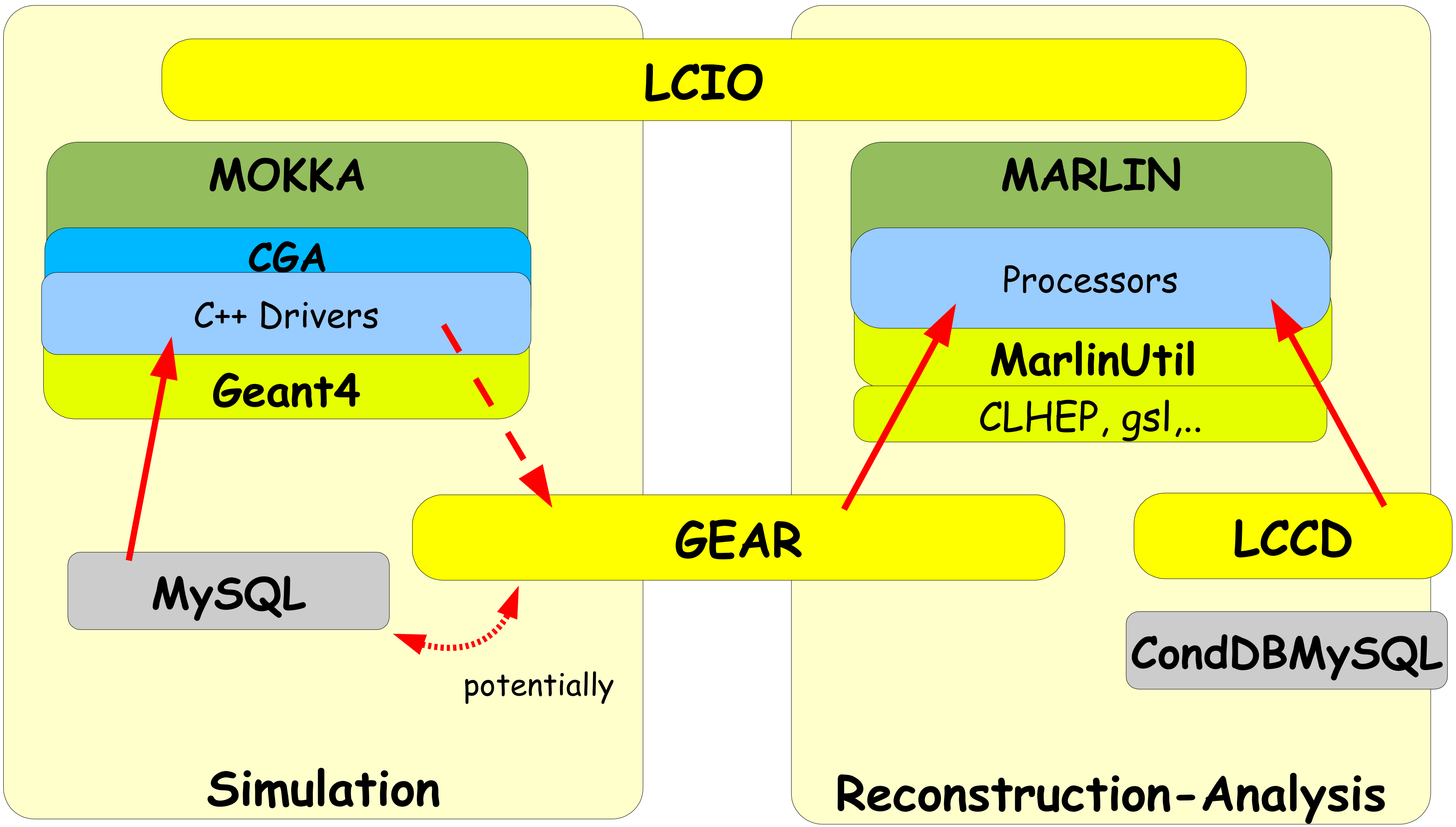
LCCD is used by Calice for the conditions data of the ongoing testbeam studies

LCCD usage

Example: CALICE test beam experiment



Software Framework



MarlinReco

- Marlin serves as a **framework** for the distributed development of reconstruction algorithms
  - provides a well defined modularity
- MarlinReco is a **toolkit** which aims at providing reconstruction algorithms for detector concept studies
  - (almost) complete set of standard reconstruction (pflow)
  - cheaters for cross checks (and replacements)
  - all processors can seamlessly be combined together with other reconstruction code or plugged into your analysis
    - e.g. different clustering algorithms
    - e.g. different track finding codes

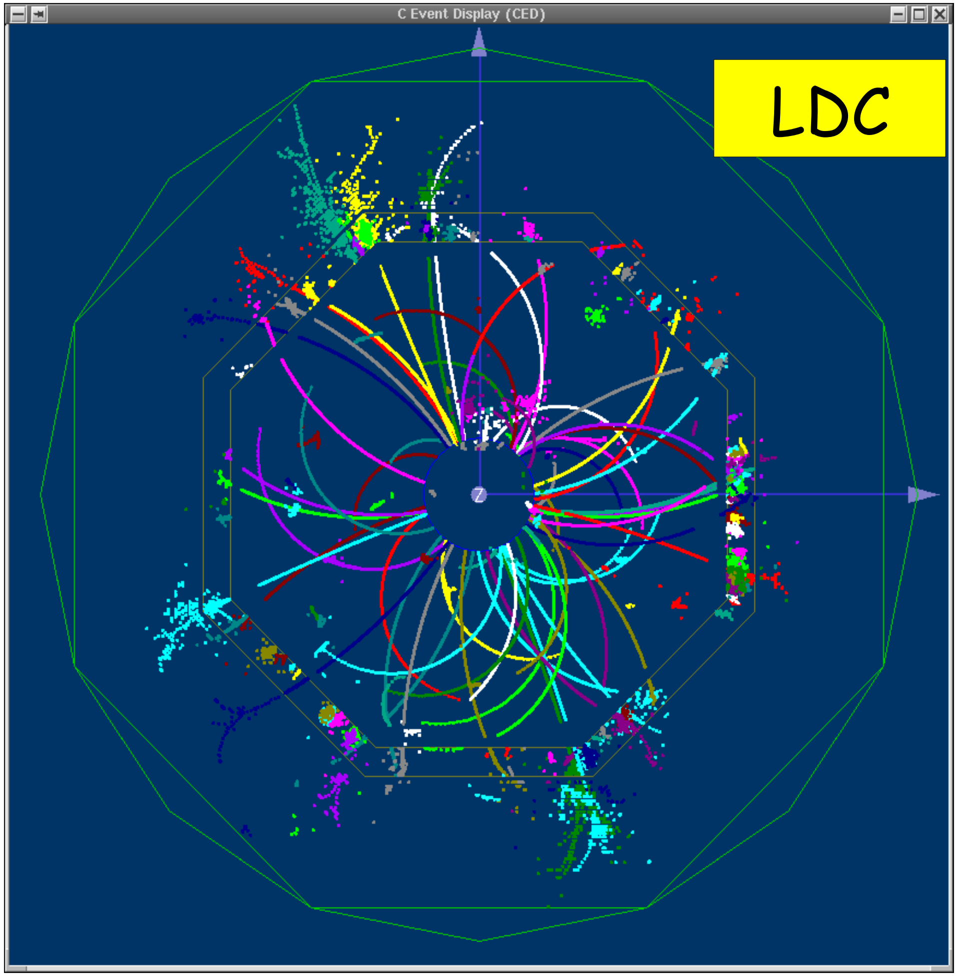
MarlinReco packages

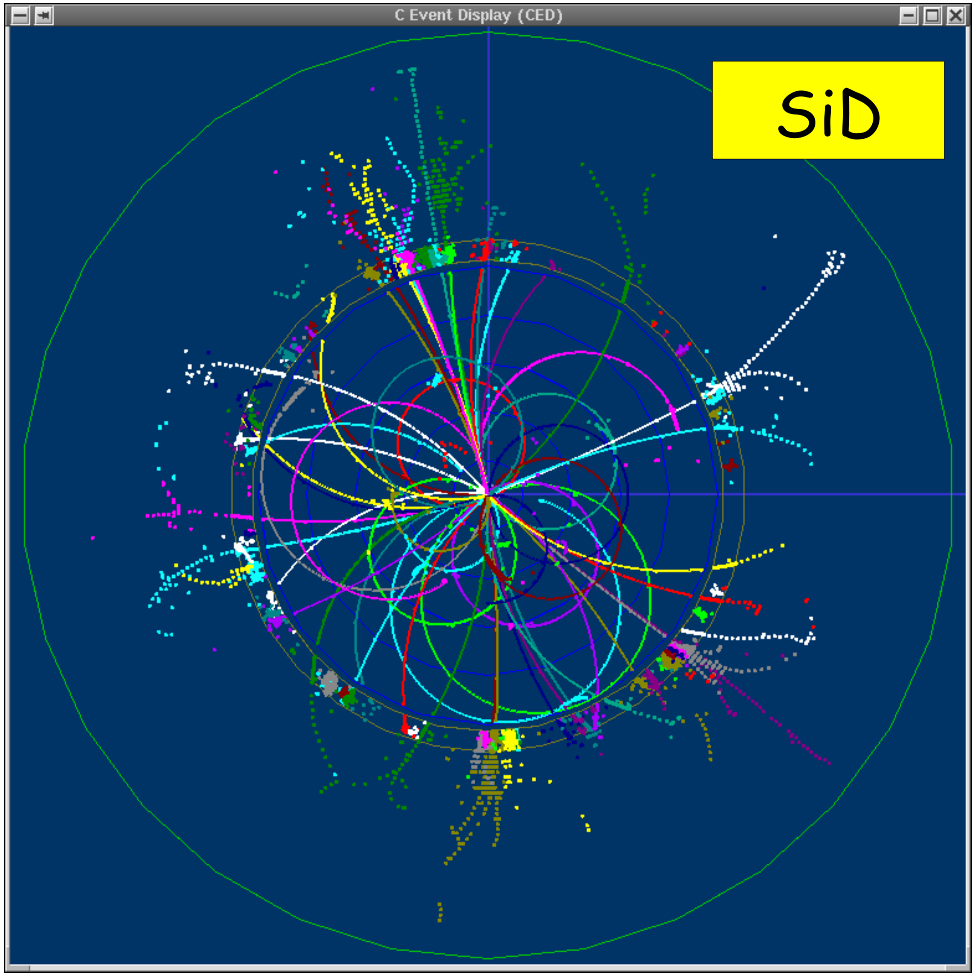
- **TrackDigi**
  - smearing in TPC
- **CaloDigi**
  - calibration, E-cut, ganging
- **Tracking**
  - central tracks in TPC+VTX
  - track cheater
- **MarlinUtil, CEDViewer**
- **Clustering**
  - trackwise clustering
  - cluster cheater
- **Pflow**
  - track-cluster match, PID
- **Analysis**
  - event shapes
  - jet finder

geometry dependence: handled through GEAR interface
apply algorithms to different detector outlines

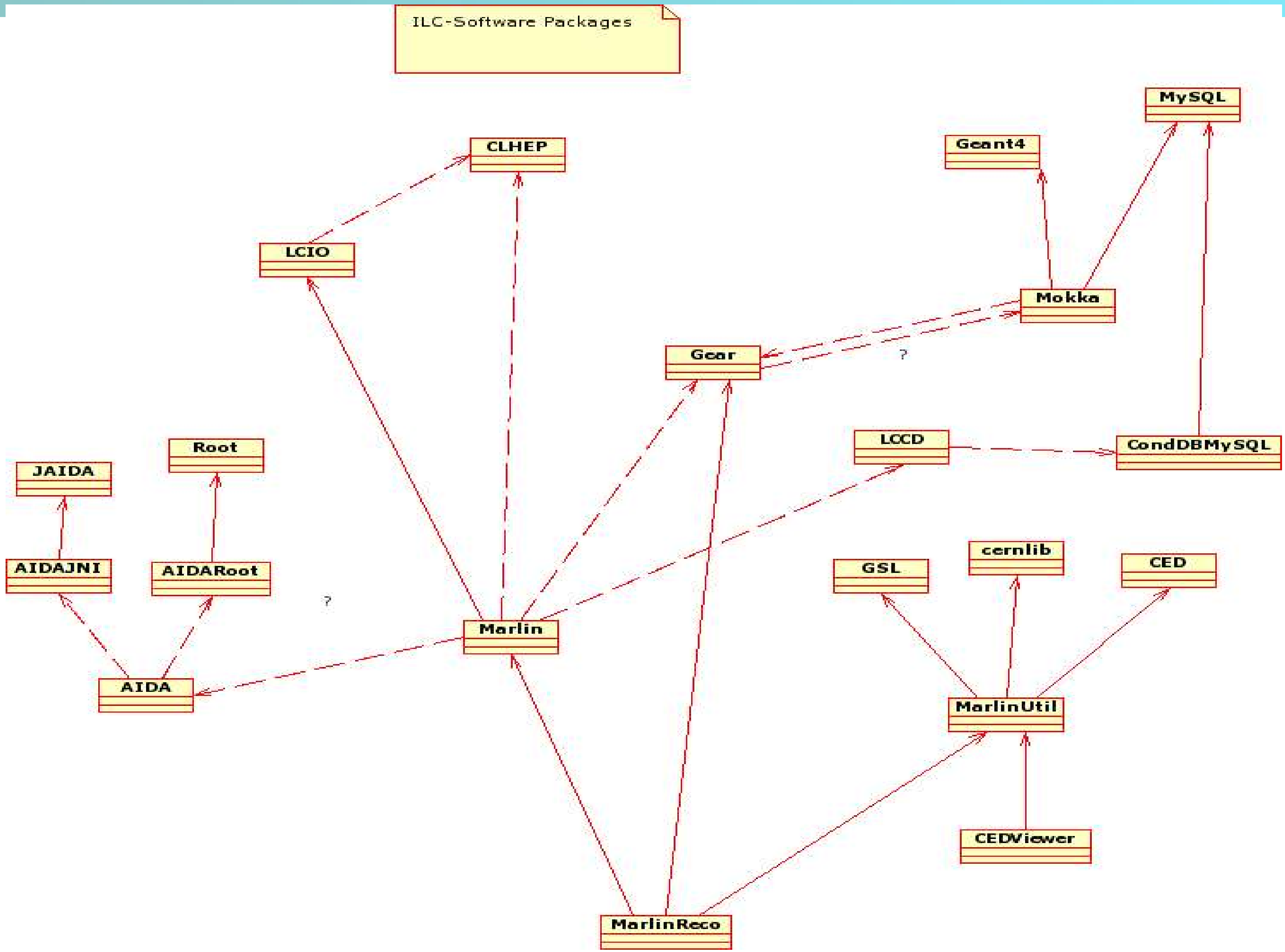
ttbar events with MarlinReco

No cheaters, only full reconstruction





software package dependencies



Tracking

- Full track finding and fitting algorithms taken from ALEPH and DELPHI optimised for TPC
  - Track finding is based on out – in search, using Circle Fit to build reference tracks
  - These are then passed to a Kalman Filter in order to take scattering within the material into account for the final fit
  - include hit pickup in inner detectors and full refit
  - Output: LCIO track collection with full covariance matrix
- Track Cheater
  - Uses MC to generate road along which hits are taken, these are then fitted with a helix hypothesis
  - Output: LCIO track collection

- missing: forward tracking
  stand-alone vertex tracking

Clustering

- Trackwise calorimeter clustering exploiting the imaging capabilities of highly granular calorimeters
    - Algorithm focuses on spatial information (no amplitude information is used at the stage of clustering), applicable to both digital and analogue calorimeters
    - Minimal dependence on detector geometry, can be used for detector optimization studies
  - Output : LCIO collection of Clusters. Each cluster is attributed with the following characteristics :
    - center-of-gravity (as position estimate)
    - vector of the main principle axis of inertia tensor (as direction estimate)
    - total energy
- MAGIC: alternative implementation of topological clustering

Event Properties and Utilities

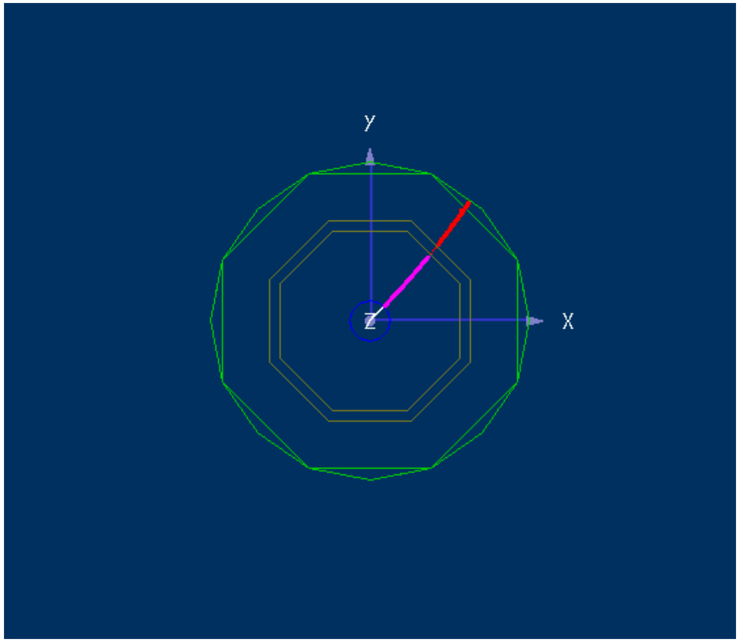
- ThrustReconstruction (T. Kraemer)
  - Tasso algorithm - calculates the principle thrust value and axis
  - Jetnet algorithm - calculates the principle thrust value and axis as well as the major and minor thrust values and axis
- Sphere (P. Krstonic)
  - Calculates the sphericity, aplanarity, C and D event parameters
- SatoruJetFinder (J. Samson)
  - A universal jetfinder module developed by Satoru Yamashita for OPAL
- Utility and Helper classes
  - These reside parallel to MarlinReco currently implemented as a separate Marlin package named MarlinUtil
    - e.g. helix fitter, clustershape (O. Wendt)

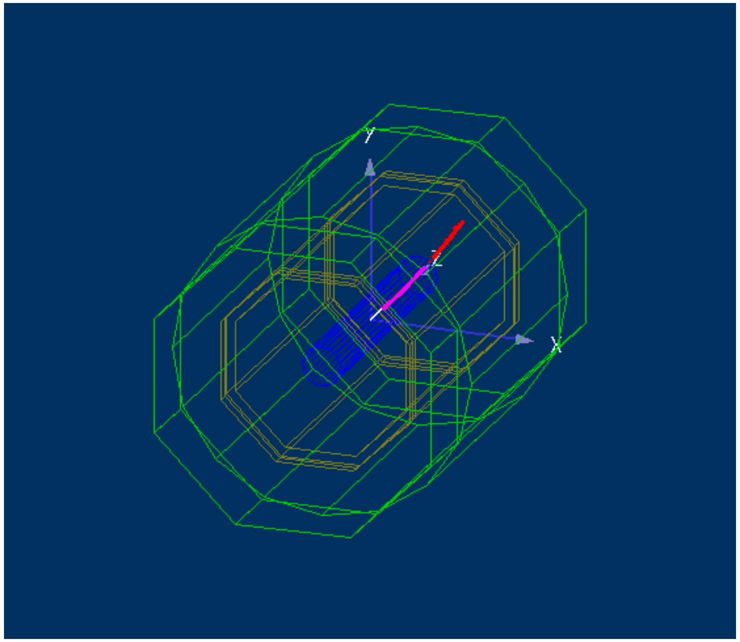
CED Viewer

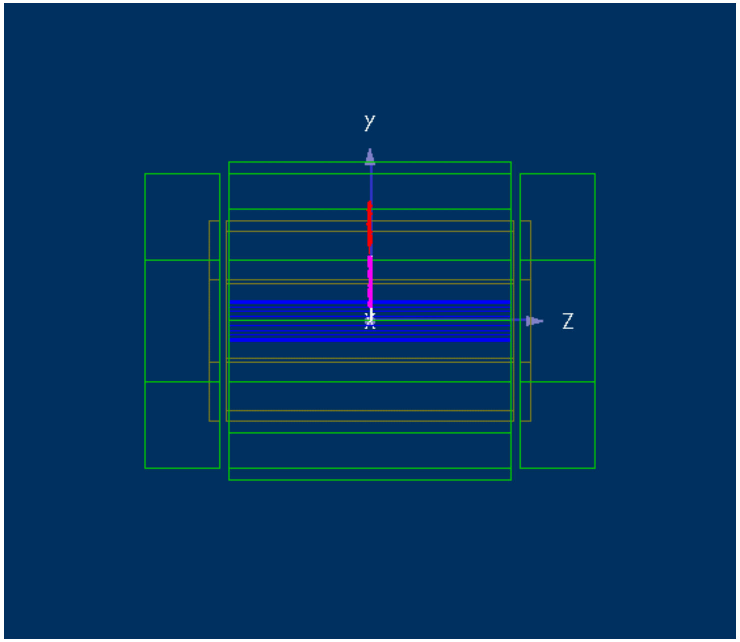
- CED developed by Alexi Zhelezov
  - Based on GLUT – OpenGL
  - Two Marlin Processors available
    - CEDViewer
    - GenericViewer
  - Displays MC objects; simulated and reconstructed hits; reconstructed track and clusters
  - Very useful in the early stages of algorithm development

- advantages:
  - very fast
  - easy to change, easy to implement custom graphics
  - but scope is somewhat limited

CEDViewer







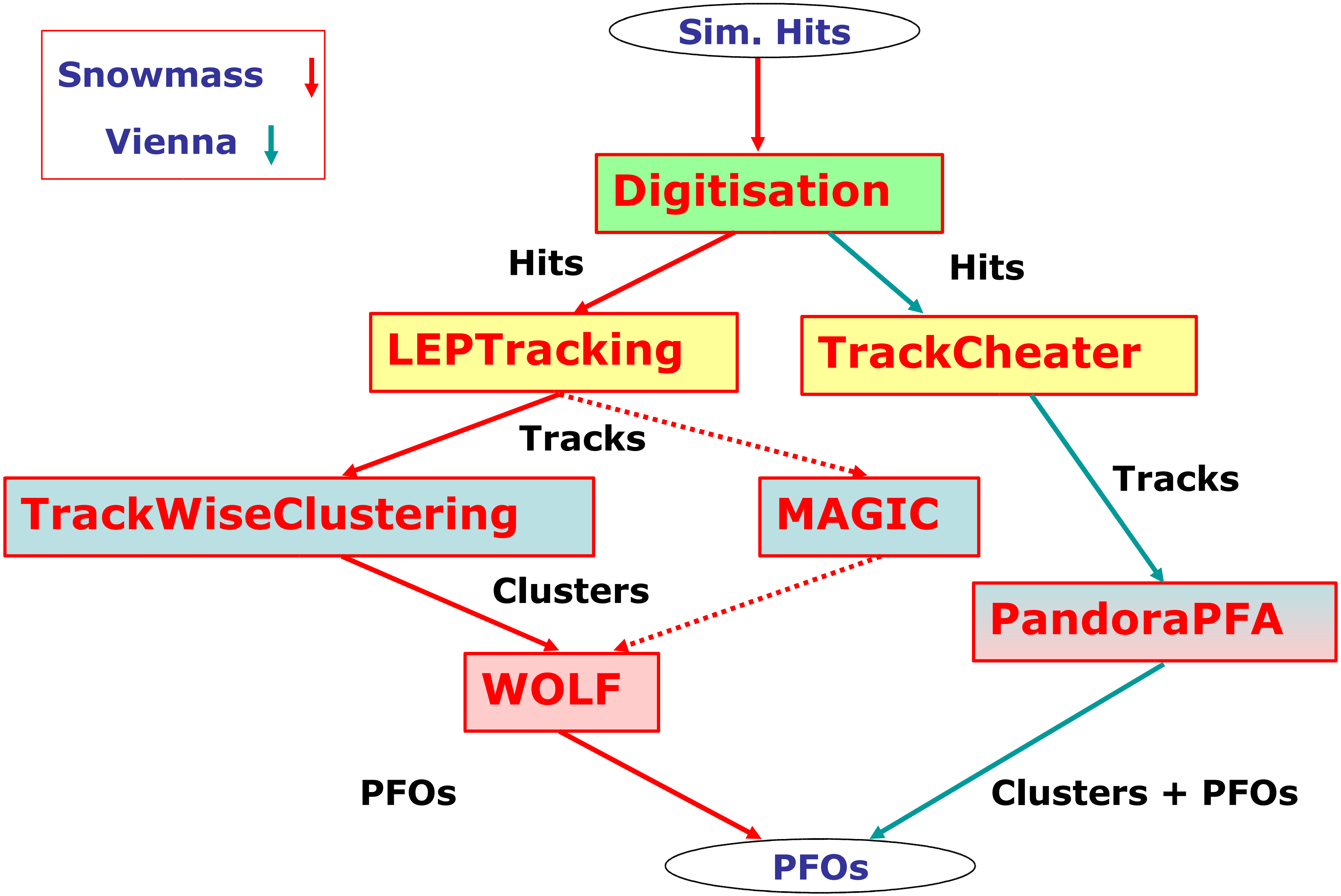
PFlow

- PFA is implemented as a dedicated processor
  - Track-cluster matching
  - Estimation of four-momenta of PF objects
    - charged objects (clusters with associated tracks) : 4-momentum is evaluated based on tracking information
    - neutral particles (calorimeter clusters with no associated tracks) : 4-momentum is evaluated using calorimeter information
  - Particle ID
    - currently based on calorimeter cluster shape analysis (fraction of energy in ECAL, longitudinal profile, transverse profile) and amplitude analysis (test of MIP hypothesis)

2 fairly complete implementations:

WOLF (A. Raspereza etal)
PandoraPFA (M. Thompson)

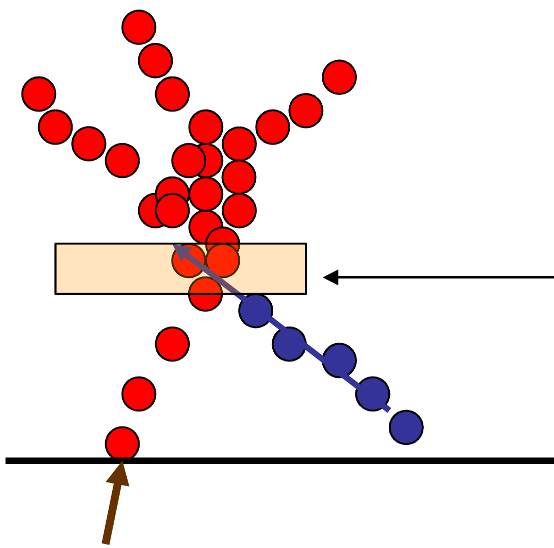
Particle Flow Algorithms in MARLIN



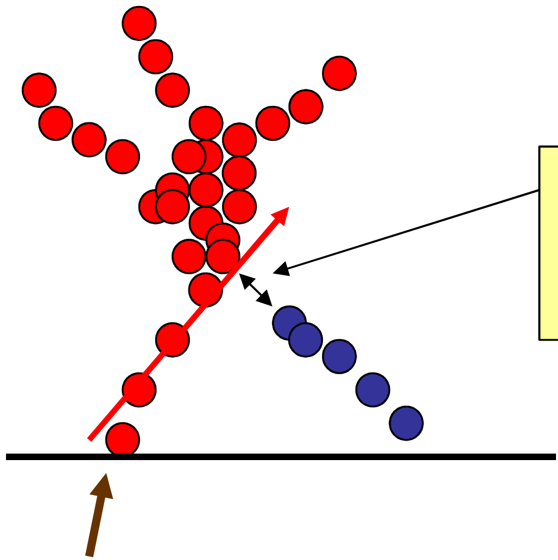
- ★ PandoraPFA/WOLF/MAGIC share many common features
- ★ Will briefly discuss some of the main points of the new Algorithm

Cluster Association : Backscatters

- Forward propagation clustering algorithm has a major drawback:
  back scattered particles form separate clusters

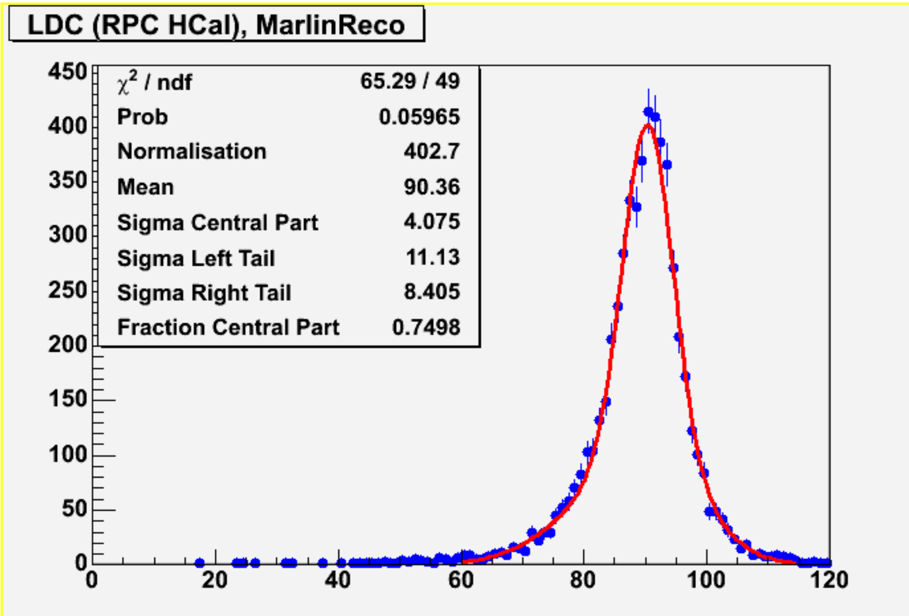


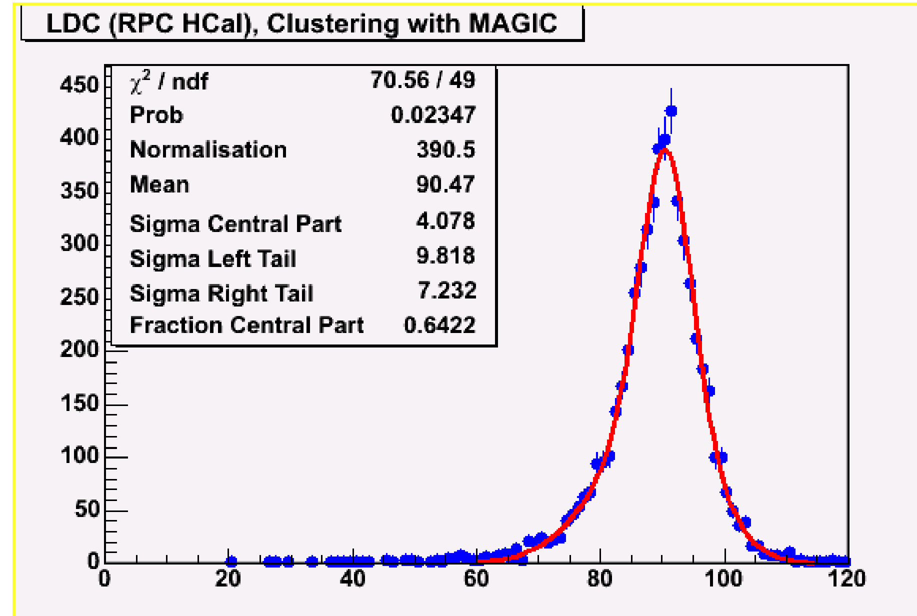
Project track-like clusters forward and check distance to shower centroids in subsequent N layers

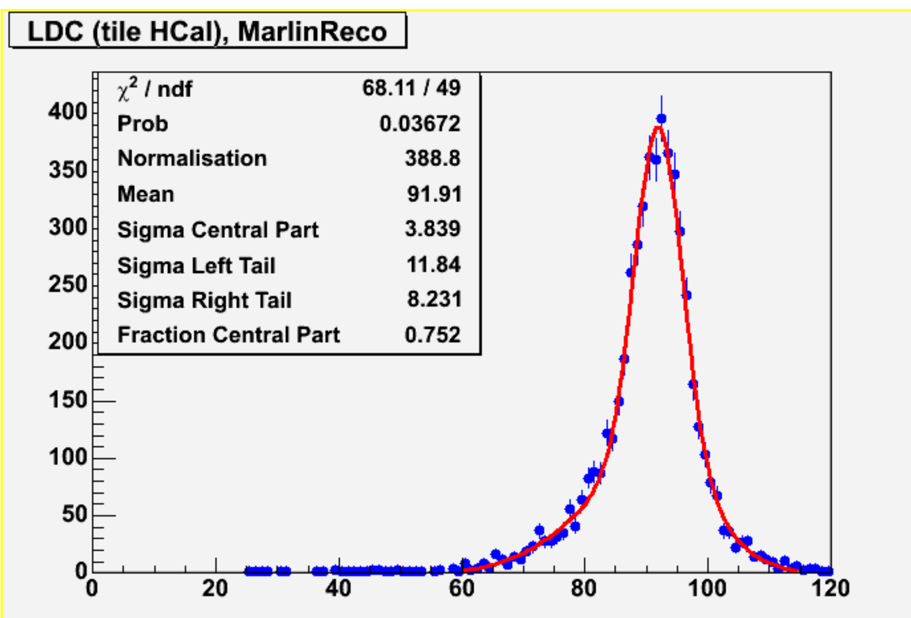


Also look for track-like segments at start of cluster and try to match to end of another cluster

Wolf Results (Z $\rightarrow$uds)







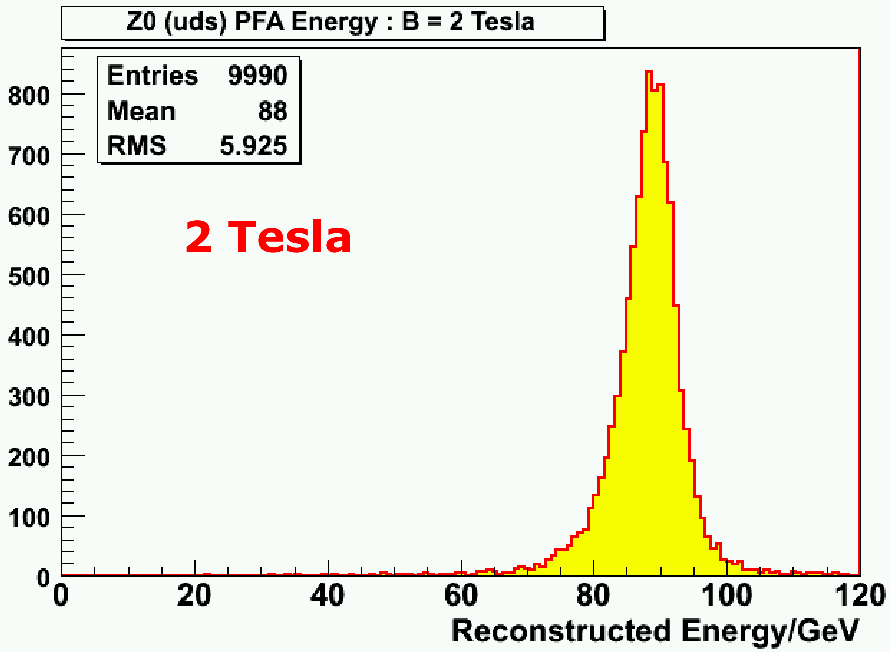
- RMS of Central 90 % of Events

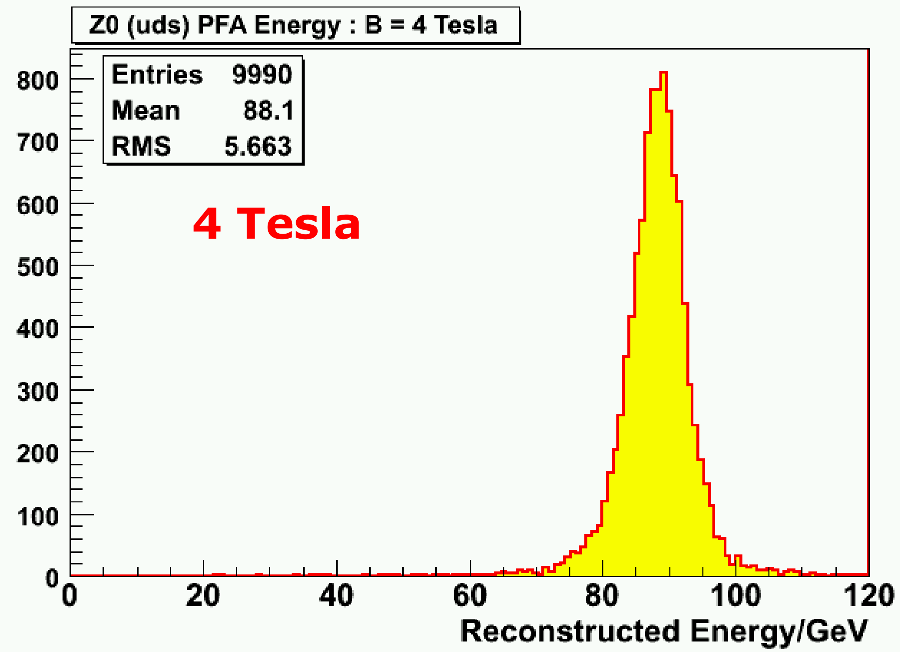
	RMS (90%)
RPC HCAL	4.3 GeV
Tile HCAL	4.1 GeV
RPC (MAGIC)	4.4 GeV

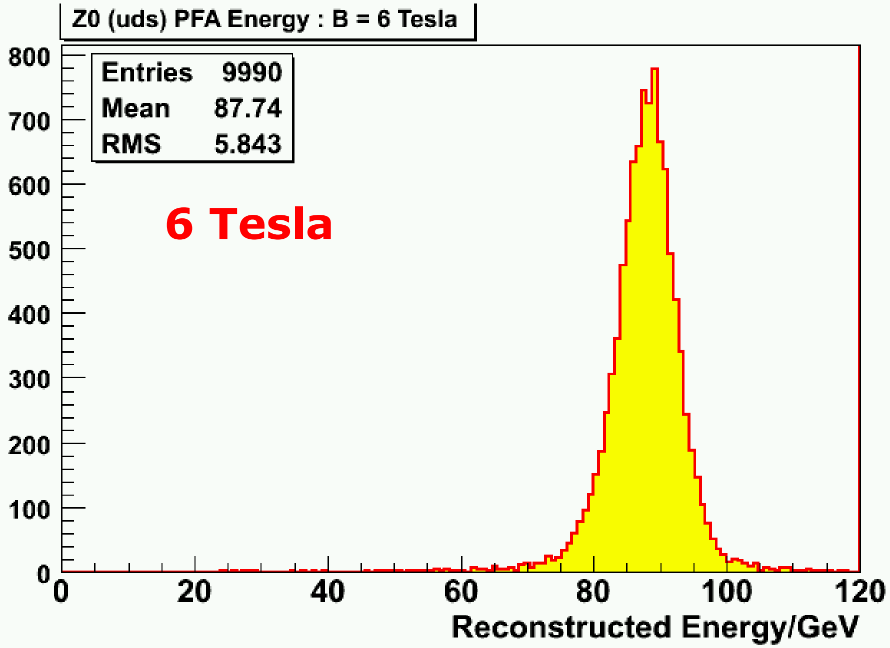
- RMS (90 %) is somewhat larger than width of fitted peak

(Results for Reco Tracks)

PandoraPFA Results (Z →uds)







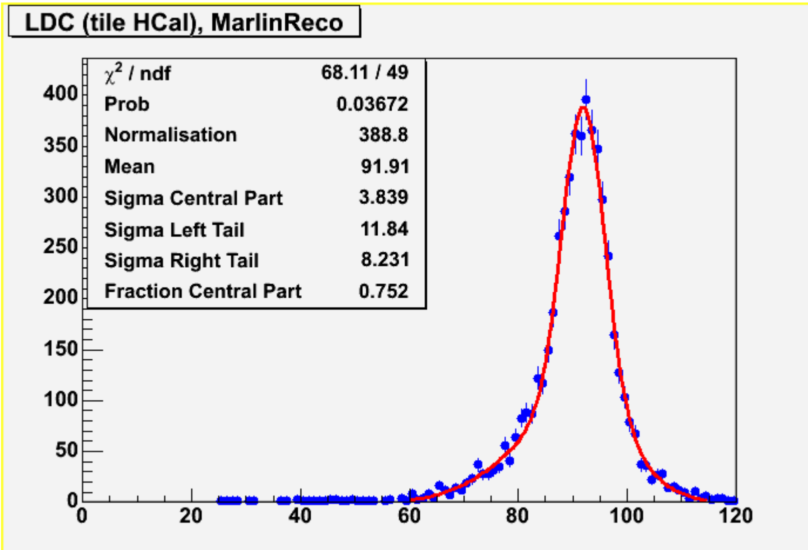
- RMS of Central 90 % of Events

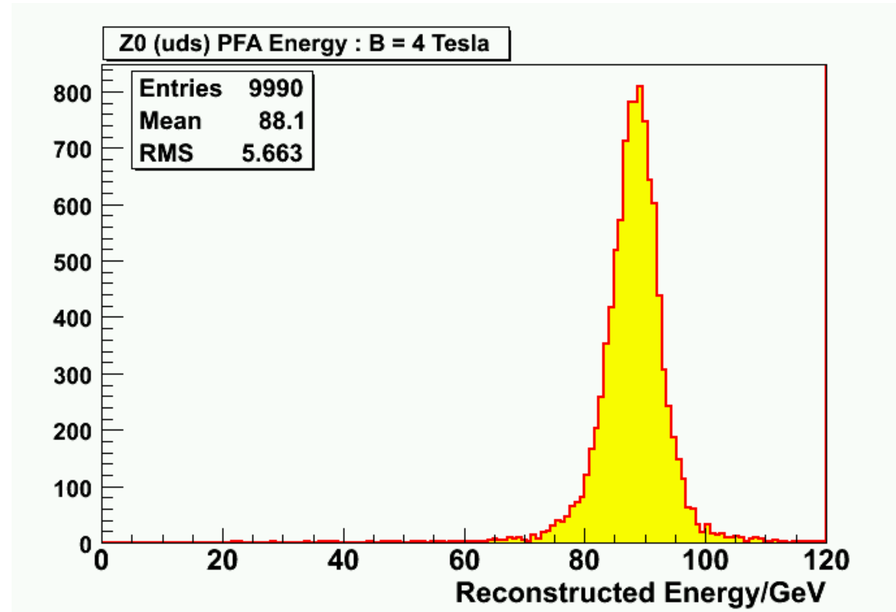
B-Field	$\sigma_E/E = \alpha\sqrt{(E/GeV)}$
2 Tesla	35.3±0.3%
4 Tesla	35.8±0.3 %
6 Tesla	37.0±0.3 %

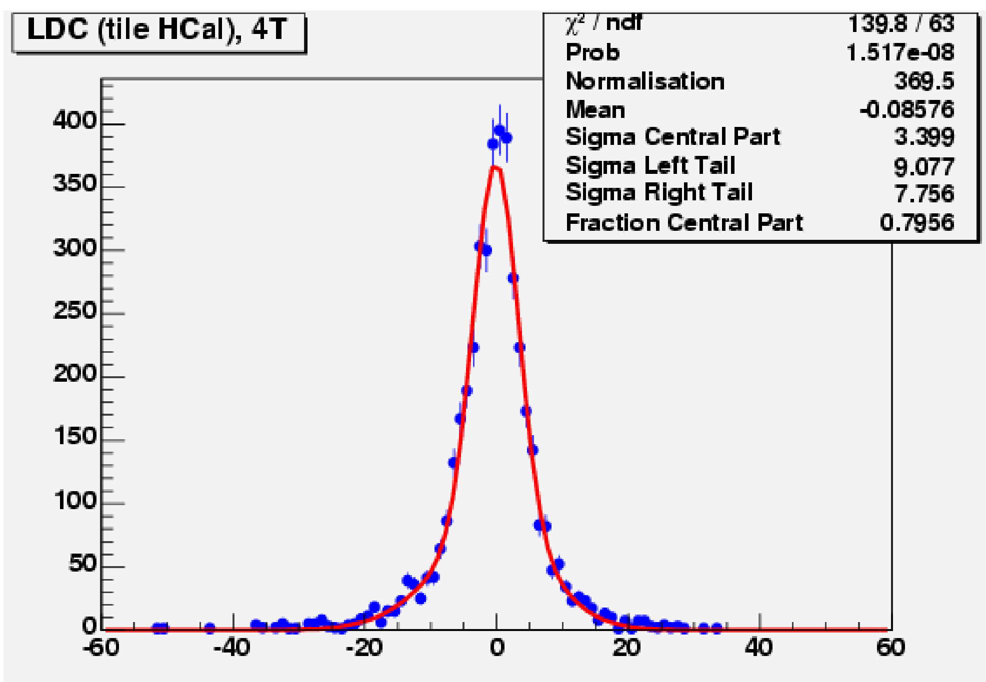
- only weakly depends on B

(Results for Cheated Tracks)

PFLOW results







The newest results:

improvement based on proper energy calibration of calorimeter
(V. Morgunov)

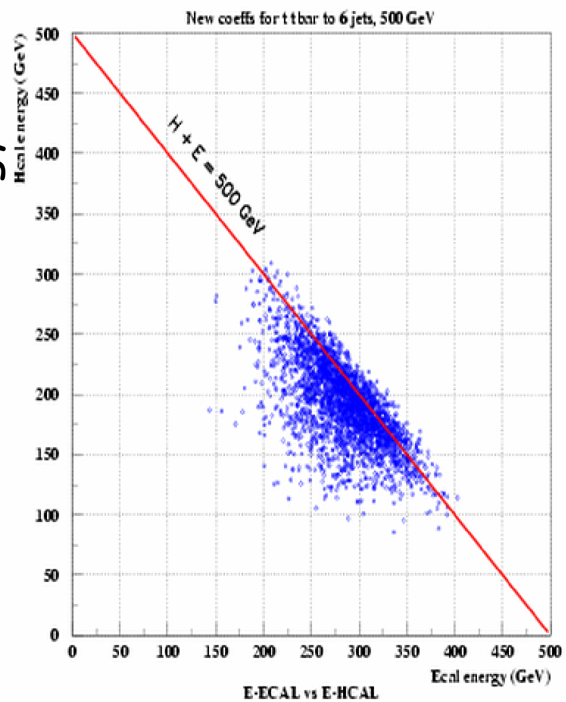
37%, with real tracks etc

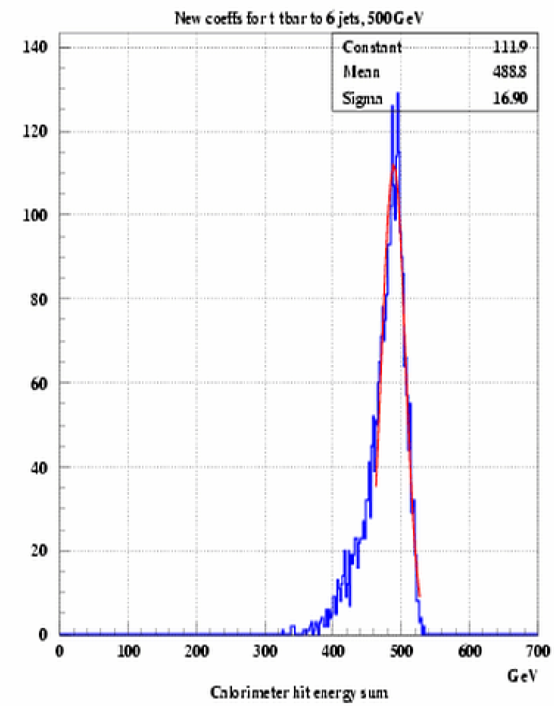
PFLOW progress

No particle flow, utilise only hermeticity of detector and good calorimeter

measure total deposited energy in calo

HCAL energy





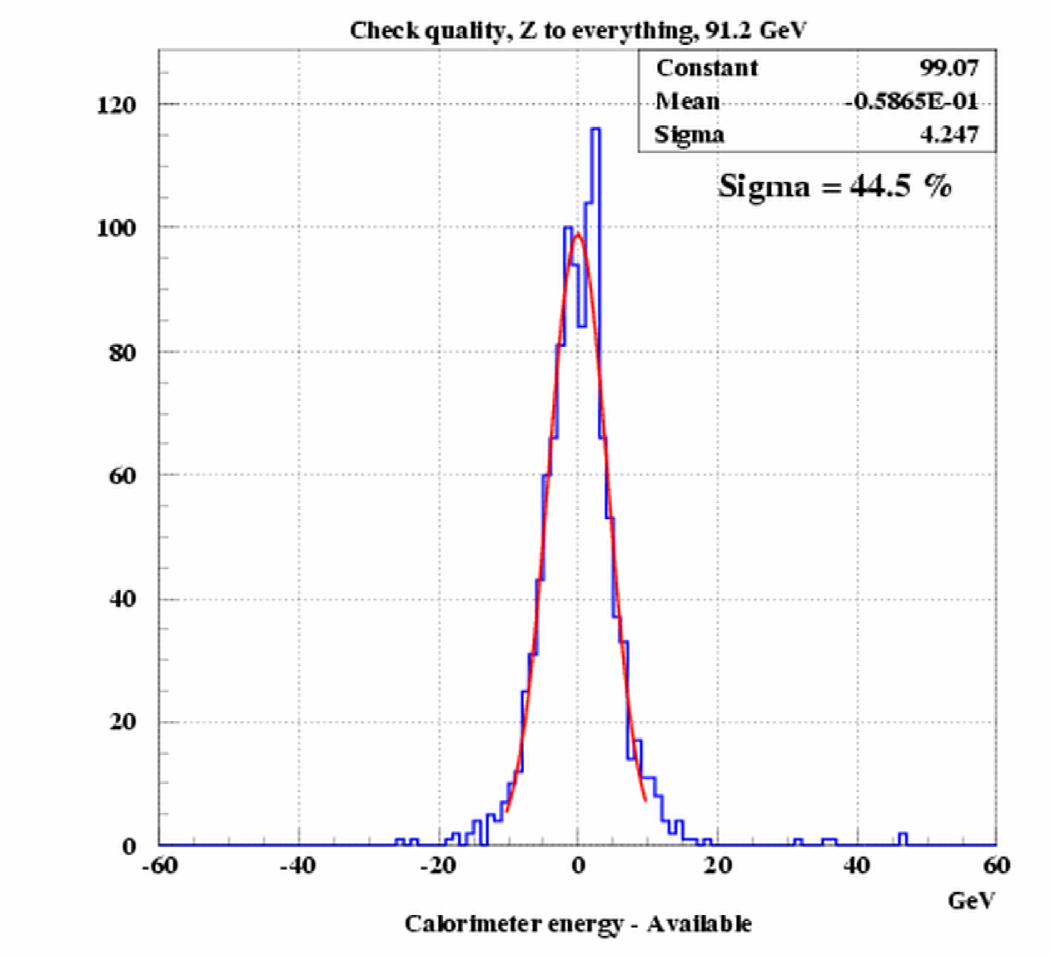
ECAL energy

use known total energy to derive calibration constants for the calorimeter

Do calibration of average response, assume same flavour composition of jets

V. Morgunov

The Z0 starting point

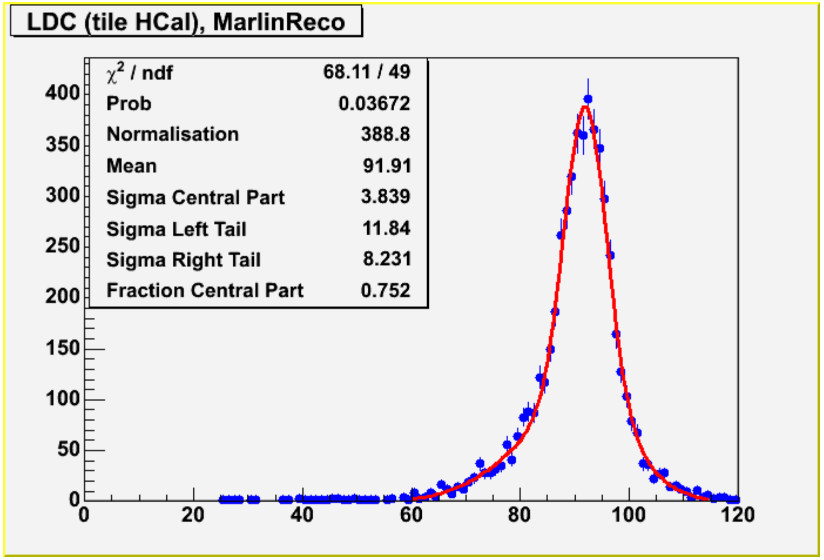


The reference reaction:
Z hadronics decays

available:
tree – neutrino - leakage

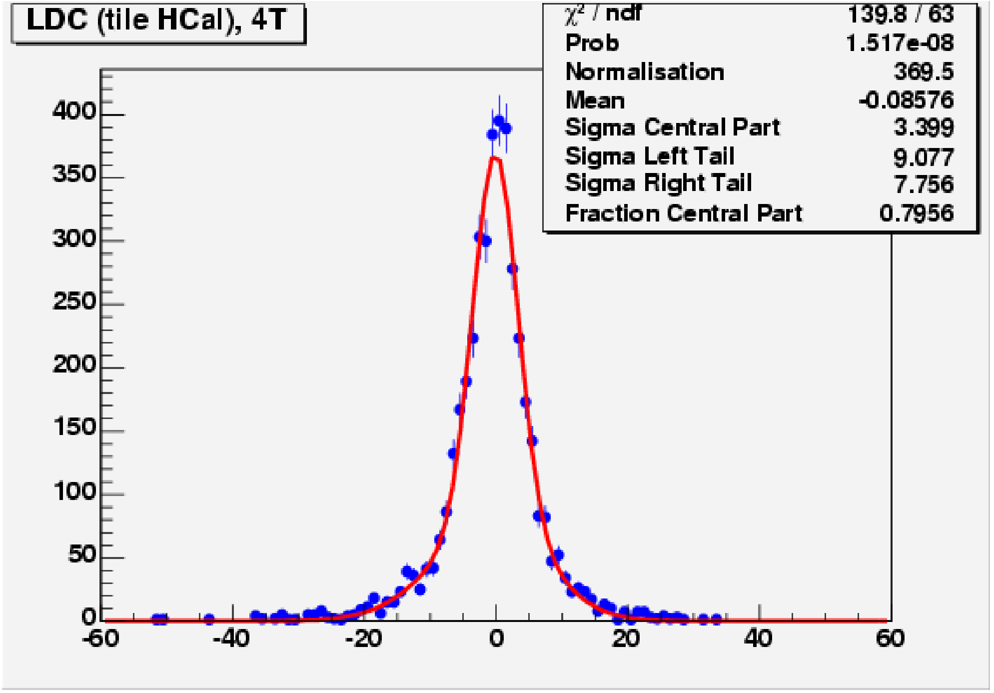
Resolution WITHOUT using particle flow:
44.5%
is possible

Latest Results



3.7 GeV

Based on calibration with single particles (muons)



3.4 GeV

Based on calibration with Jets

Particle Flow Plans

- continue development of WOLF and PANDORAPFA

    expect releases of software in January / February for general use

- need to improve the tracking software significantly!

- work on second generation algorithm (SNARK inspired and improved) ongoing

want to use some PFLOW in the LDC optimisation before outline document!

Event Production

Software is GRID enabled

use EGEE grid software for job submission and control
ILC is accepted virtual organisation in Europe in several centers

map of potential
sites which can host
ILC VO
within EGEE

collaboration with
Nordu-GRID
and GRID3 ongoing

First mass production

Before Christmas:

major new releases for all central software packages
much improved stability
numerous small and large bug fixes

During Christmas holidays:

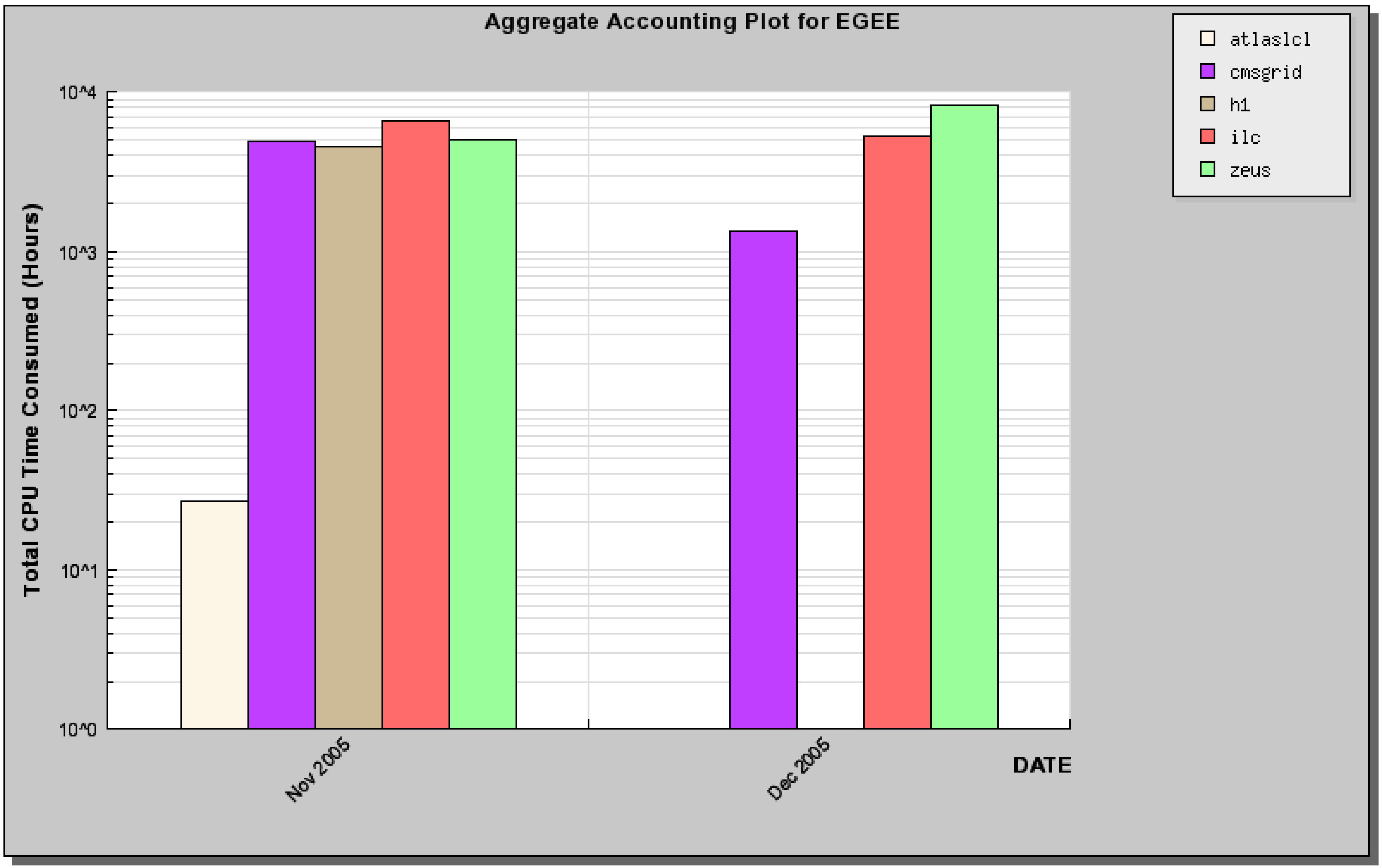
operated GRID extensivly and produced several 10000 of fully simulated events

ttbar events
WW events — three different geometries for LDC
Z events

Data are available on the GRID (see ilcsoft.desy.de)

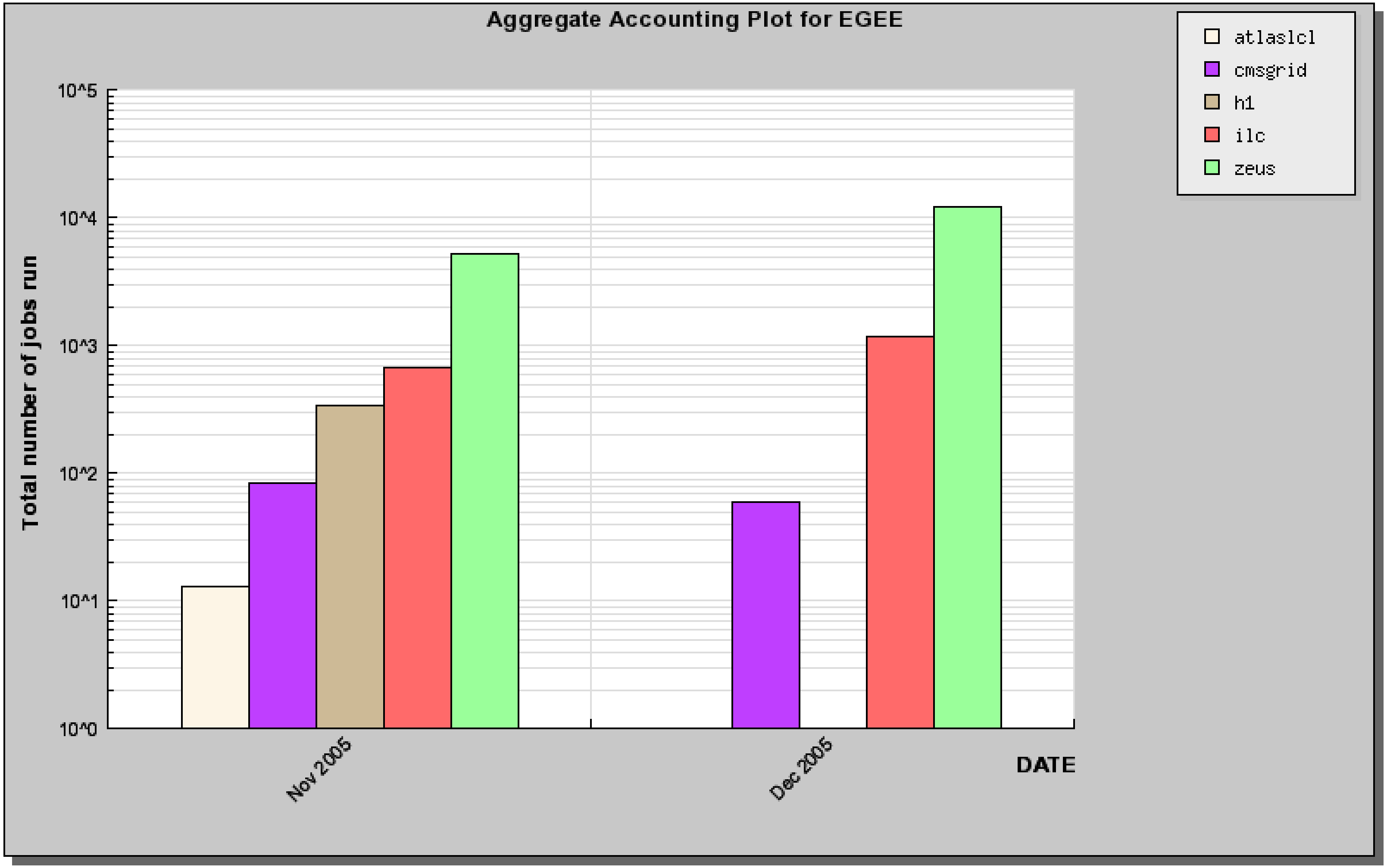
GRID statistics for ILC

plot of CPU time consumed during Nov – Dec for ILC on EGEE



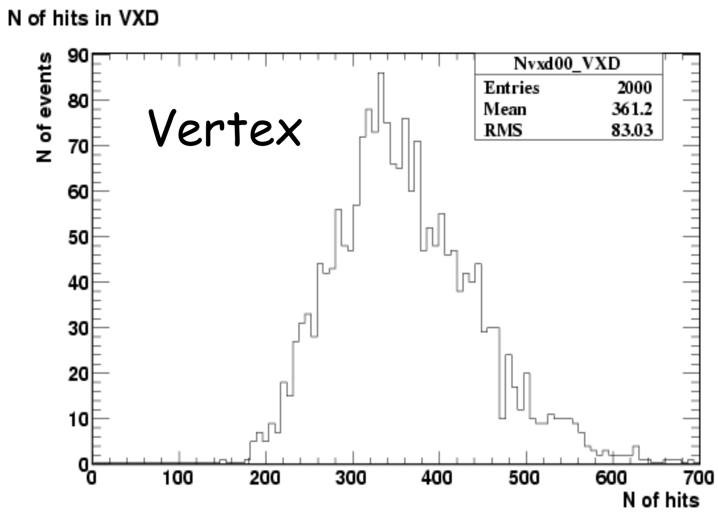
GRID statistics

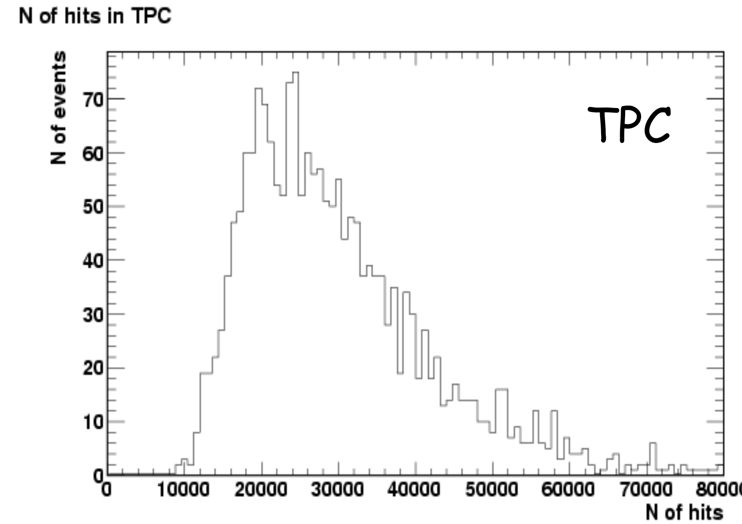
Number of jobs processed on GRID

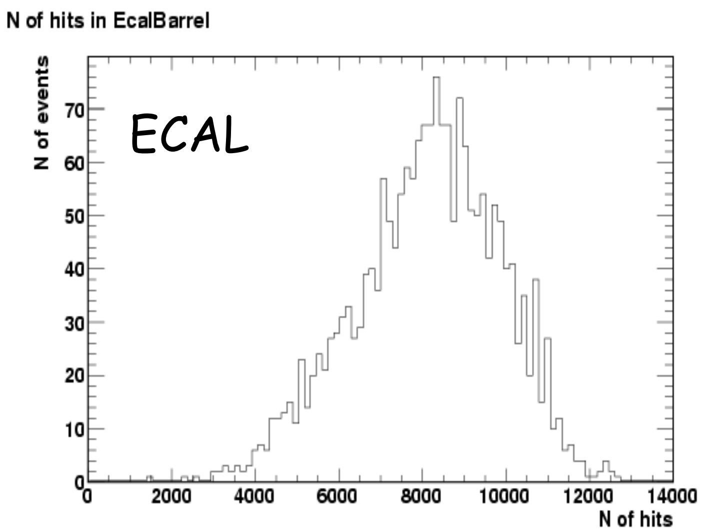


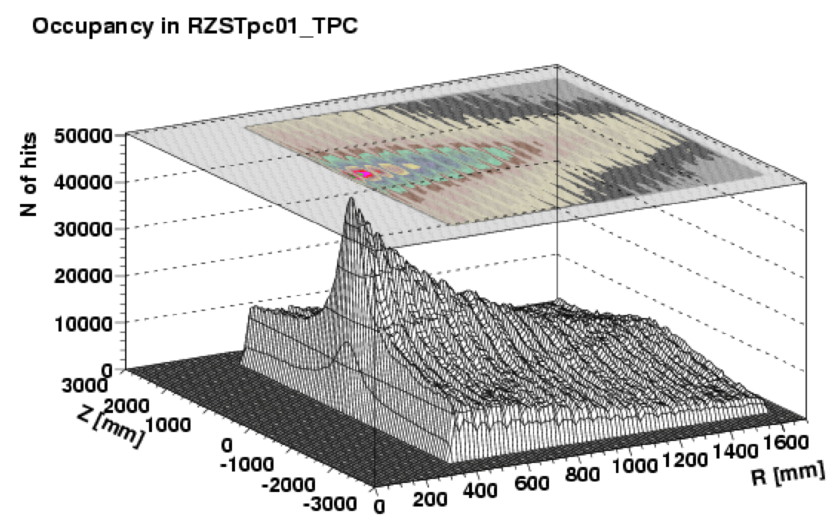
during Christmas: run approx. 1000 jobs (1000 events each)

Some data quality checks









output of data quality check processor under MARLIN (to be released) O. Wendt

GRID for the ILC

current ILC hosts with significant resources:

DESY

Freiburg

RAL

Zeuthen

significant CPU power is available for ILC work (often lower priority though)

ILC VO is administered through DESY and Karlsruhe Tier 1 center

Plans:

we plan to utilise the GRID system heavily for both

production (computing GRID)

data storage (storage GRID)

Summary

MOKKA MARLIN based software framework is getting usuable

A number of reconstruction utilities exist

The user base is expanding, and people start to contribute code

I still hope for a better integration and sharing of software between the efforts/ regions / concepts

Next software workshop in Europe (focus on reconstruction).

April 4-6 in Cambridge, UK

http://www.hep.phy.cam.ac.uk/~thomson/meetings/ilcsoft/

Outlook

This software is available through

http://ilcsoft.desy.de

The site is evolving, not all packages are equally well supported or documented

Manpower resources are very small:

I think we need to improve the collaboration and exchange of software etc between the regions/ concepts/ ...