# Overview and Status of the LDC Software

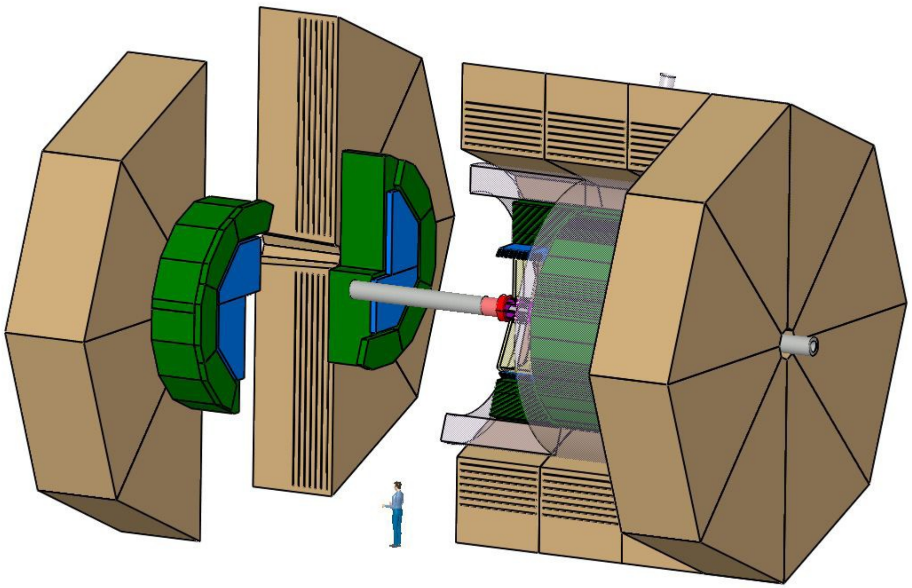
## Are we ready for physics ?

Frank Gaede
DESY

LDC UK Meeting, Cambridge
September 21, 2007

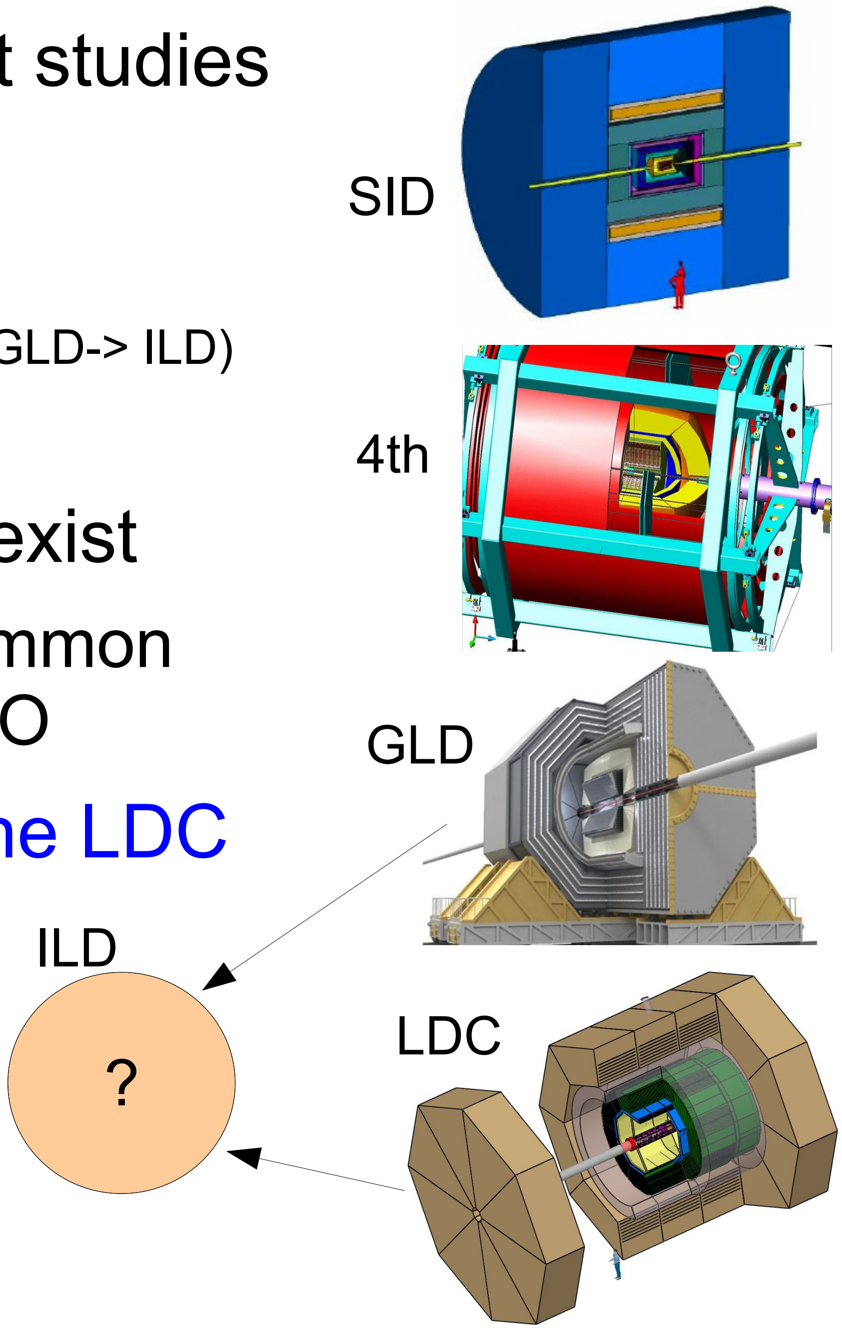
# Outline

- overview core tools
  - LCIO, Mokka, Marlin, Gear, (LCCD)
  - status and new features
  - installation, build tools
- putting it all together
  - reconstruction algorithms
  - physics studies
- grid
- summary

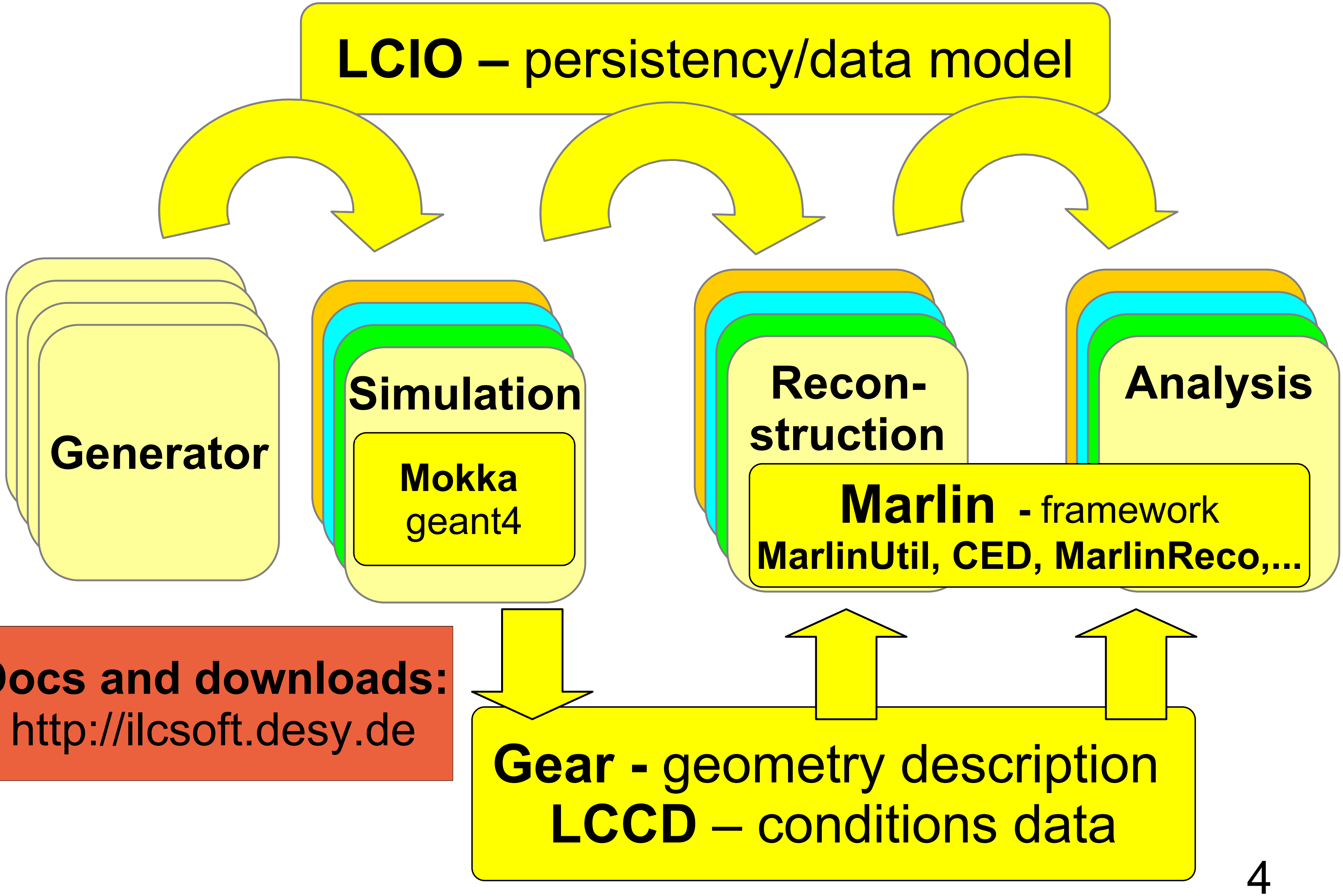


# detector concepts

- 4 international detector concept studies for the ILC ongoing
  - DCRs written this year
  - 3 **LsOI** planned for 2008 (joined LDC/GLD-> ILD)
  - 2 EDRs planned for 2010
- 4 independent sw frameworks exist
  - some interoperability through common event data model/ file format LCIO
- this talk about "Marlin et al" - the LDC concept's framework

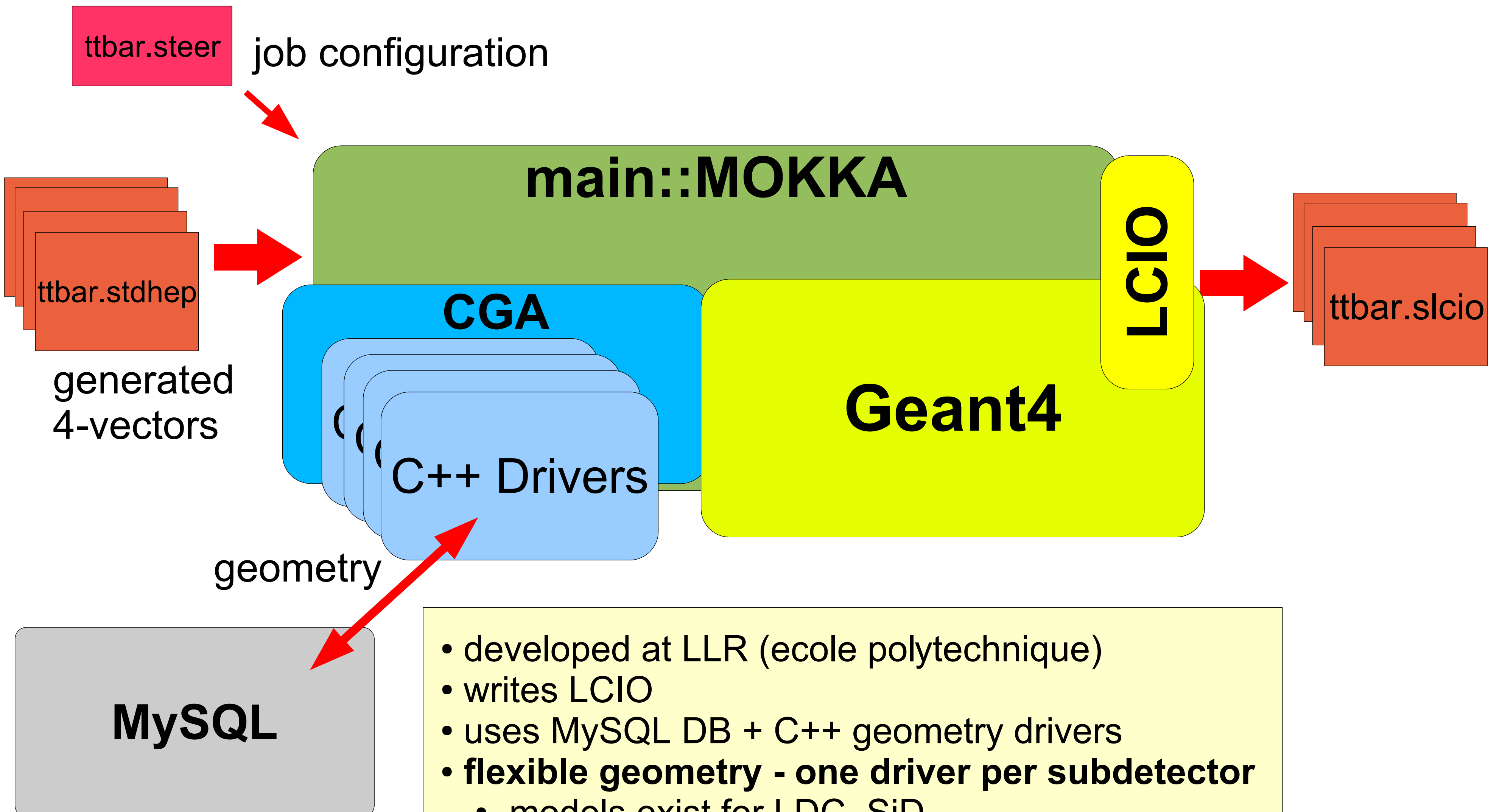


# LDC sw-framework overview



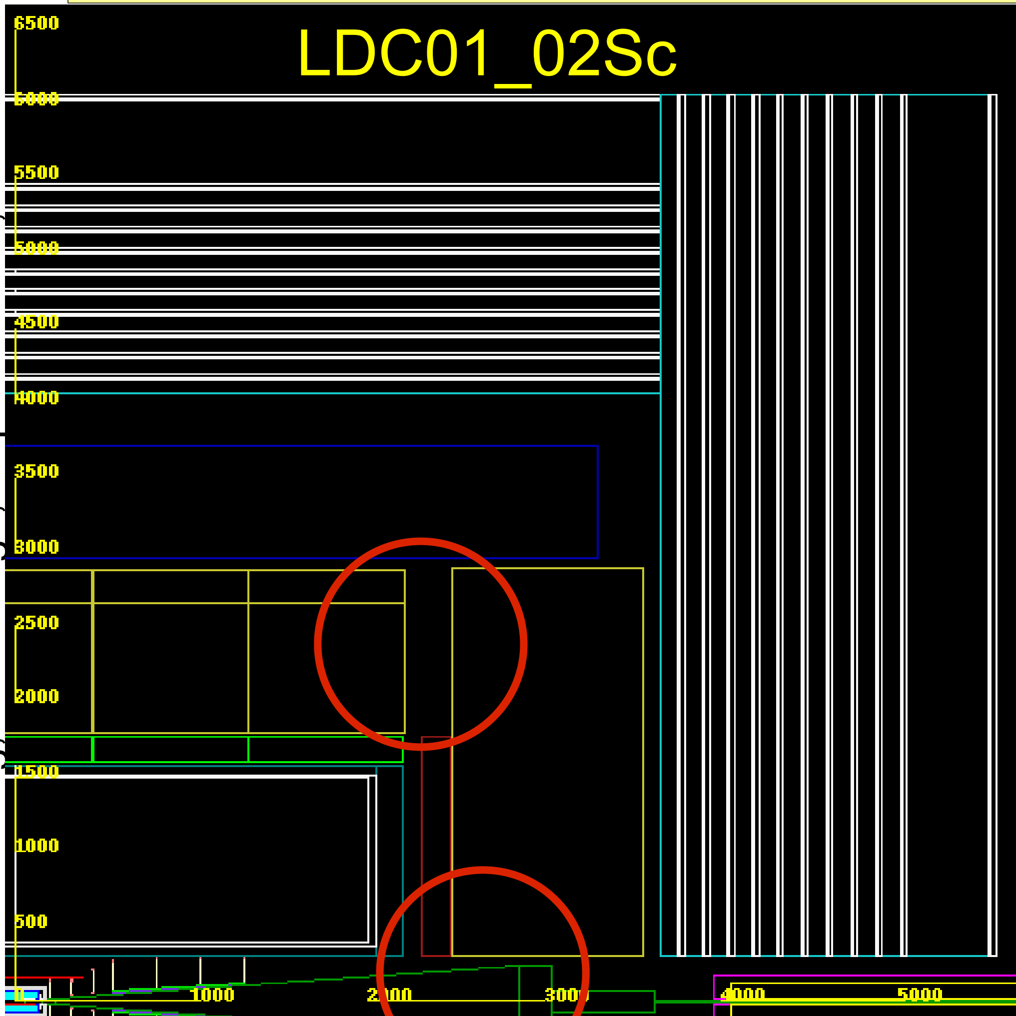

**Docs and downloads:**
http://ilcsoft.desy.de

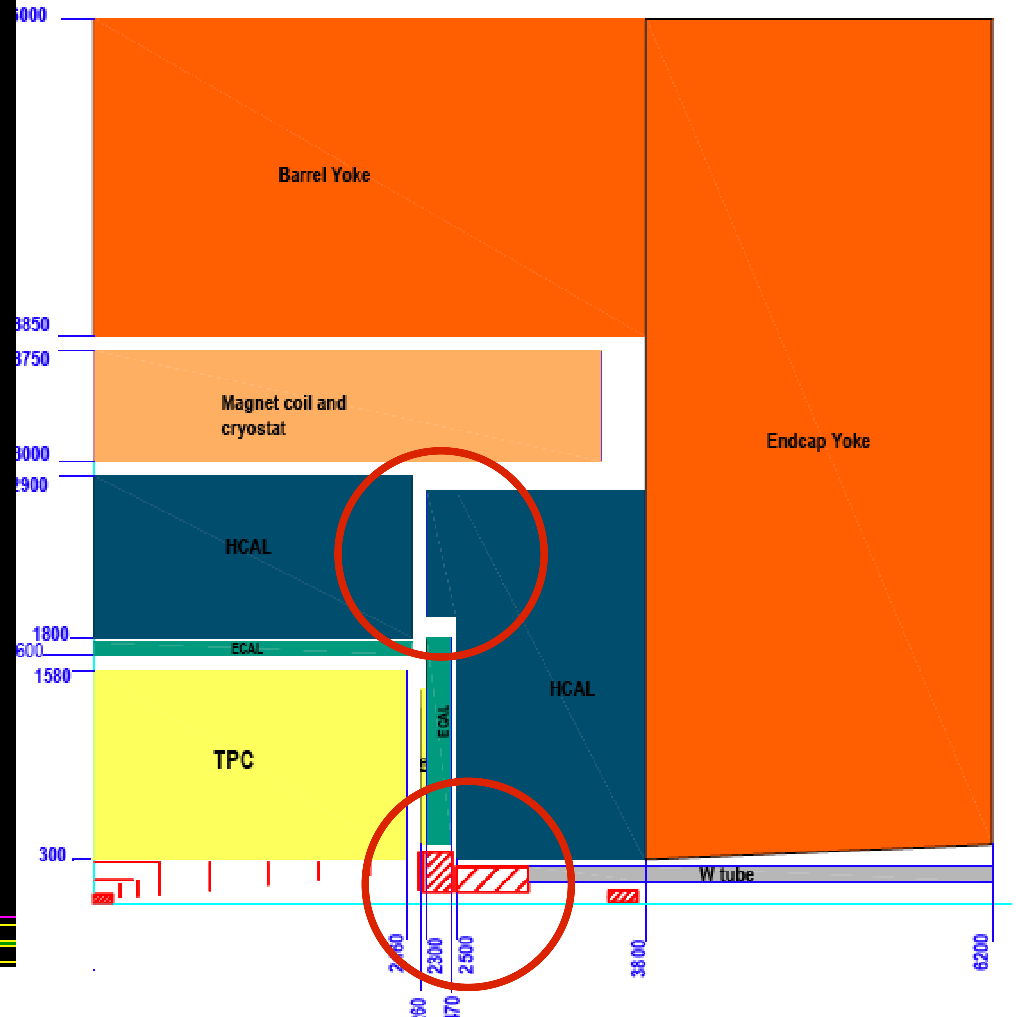
# Mokka full simulation - overview




- developed at LLR (ecole polytechnique)
- writes LCIO
- uses MySQL DB + C++ geometry drivers
- **flexible geometry - one driver per subdetector**
  - models exist for LDC, SiD
  - testbeam prototypes
    - Calice
    - EUPixelTelesescope

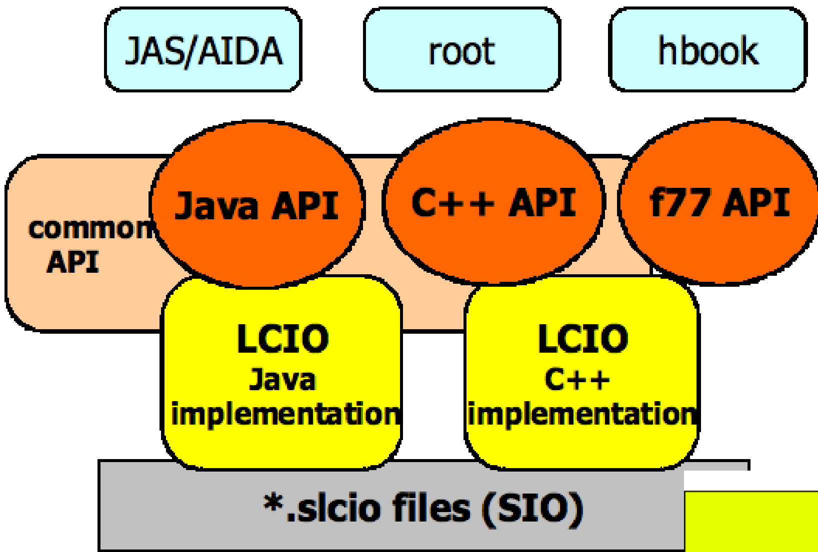
# LDC description in Mokka



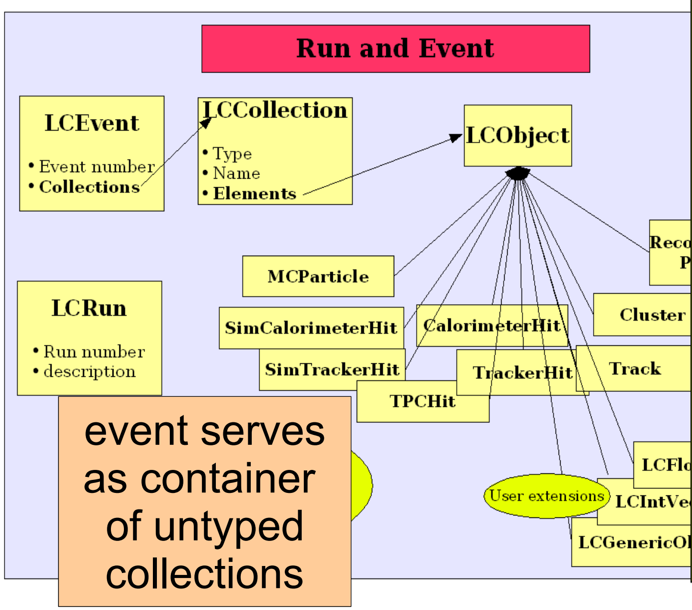



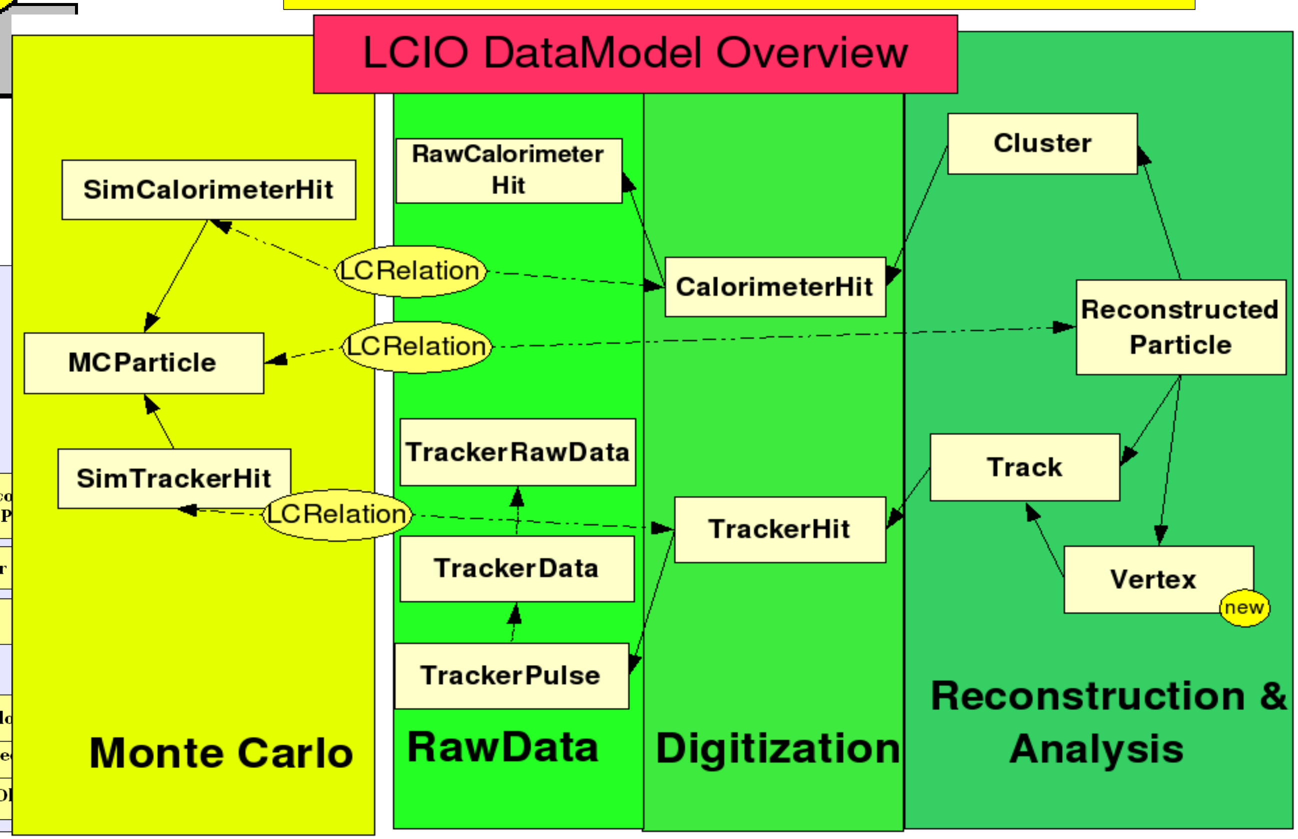

- **detailed description of LDC (DOD) exists in Mokka**
- some caveats: missing Hcal ring and forward region
- appropriate code exits – need to combine into new model
- **fixed right now for planned MonteCarlo production**

# LCIO: persistency & event data model




- DESY SLAC joined project
  (first presented at CHEP03)
- Java, C++ and f77 (!) API
- extensible data model
- now standard for
- ILC persistency & datamodel
- -> used in all detector concept
- studies



# LCIO runtime extensions (C++)

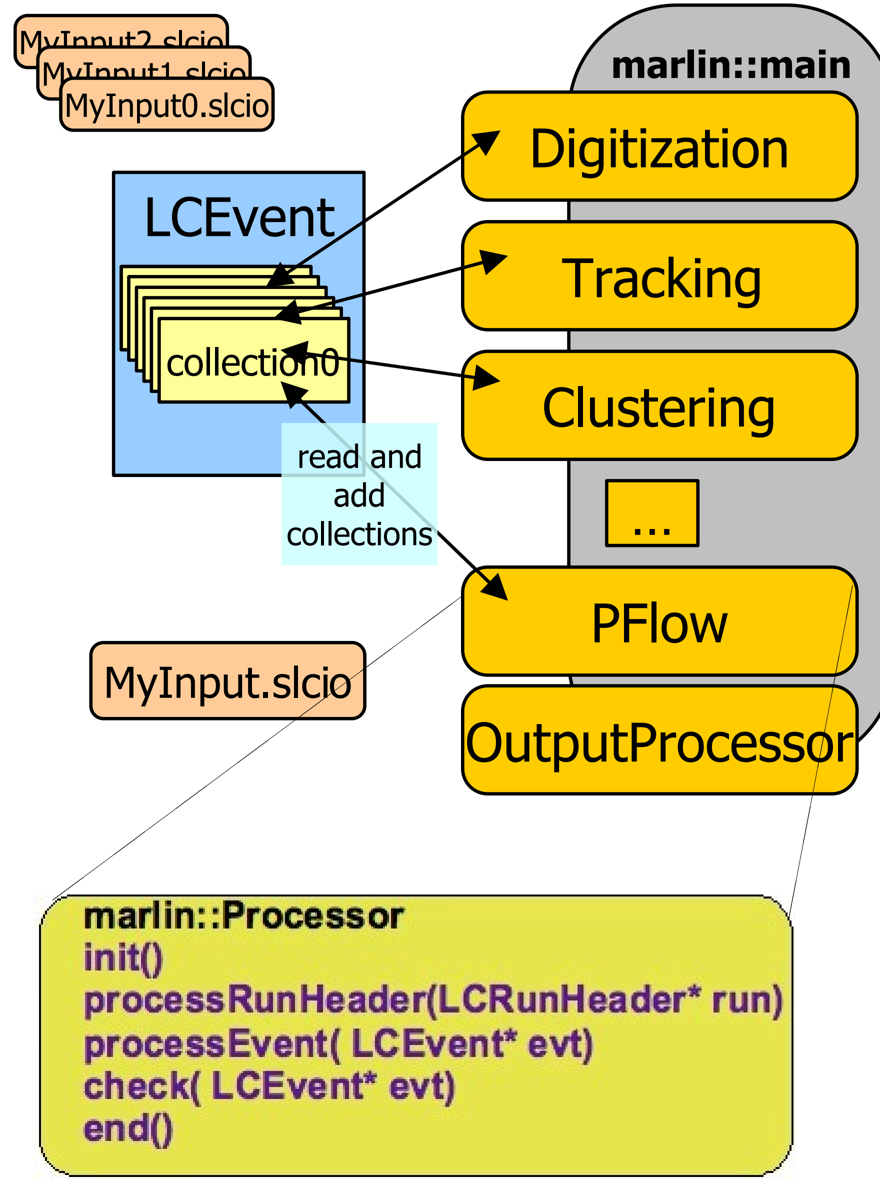
- long pending user request:
  - attach user objects to LCObjects
  - fast and easy creation of links (relations) between various LCObject subtypes, eg. TrackerHits and Track
- features
  - extension of the object with arbitrary (even non-LCObject) classes
  - bidirectional relations between LCObjects
    - one to one
    - one to many
    - many to many

extends LCIO from pure data model and persistency with capabilities needed for analysis and reconstruction

# Marlin – application framework

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

- modular C++ application framework for the analysis and reconstruction of ILC data
- **LCIO** as transient data model
- xml steering files:
  - fully configure application
  - order of modules/processors
  - parameters global + processor
- self documenting
  - parameters registered in user code
- consistency check of input/output collection types
- **Plug & Play** of modules



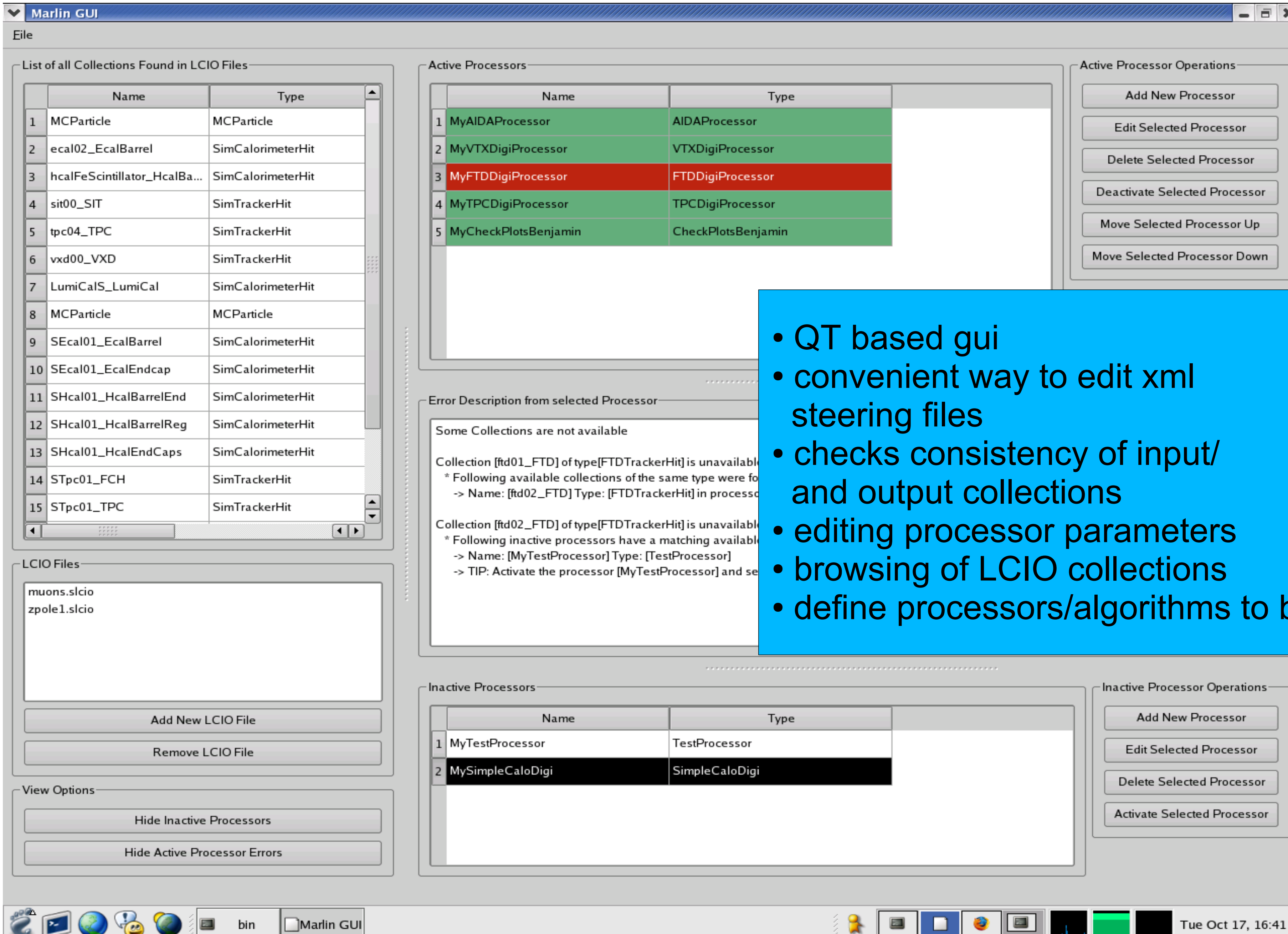
# Marlin recent developments

- Marlin fully functional since 2005
  - -> focus on increasing user, i.e. developer convenience
- introduced new build system: CMake
  - 'successor of GNU autotools' - allows easy configuration of build process and multi-platform support (Linux, MacOS, Windows)
- switched to shared libraries and support for plugins
  - users can combine their binary from installed package libraries
- MarlinGUI,
  - flow charts
- new logging mechanism:

```
streamlog_out( DEBUG ) << “ digitizing hit : “
                                         << hit->getCellID() << std::endl ;
[ DEBUG “TrackDigitizer” ]  digitizing hit : 12345678
```

# MarlinGUI

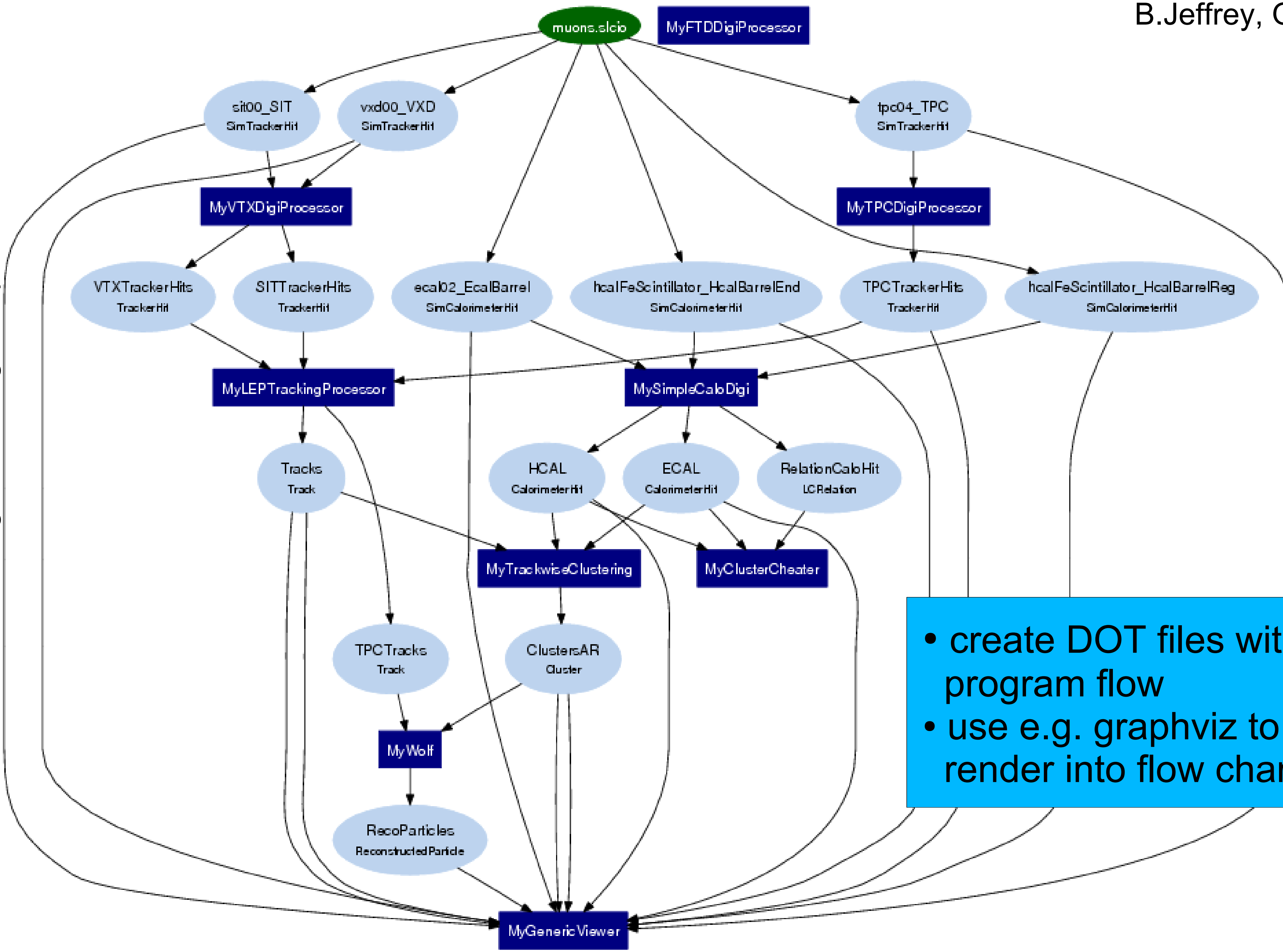
J.Engels, DESY




- QT based gui
- convenient way to edit xml steering files
- checks consistency of input/ and output collections
- editing processor parameters
- browsing of LCIO collections
- define processors/algorithms to be run

# Marlin program flow charts

B.Jeffrey, Oxford



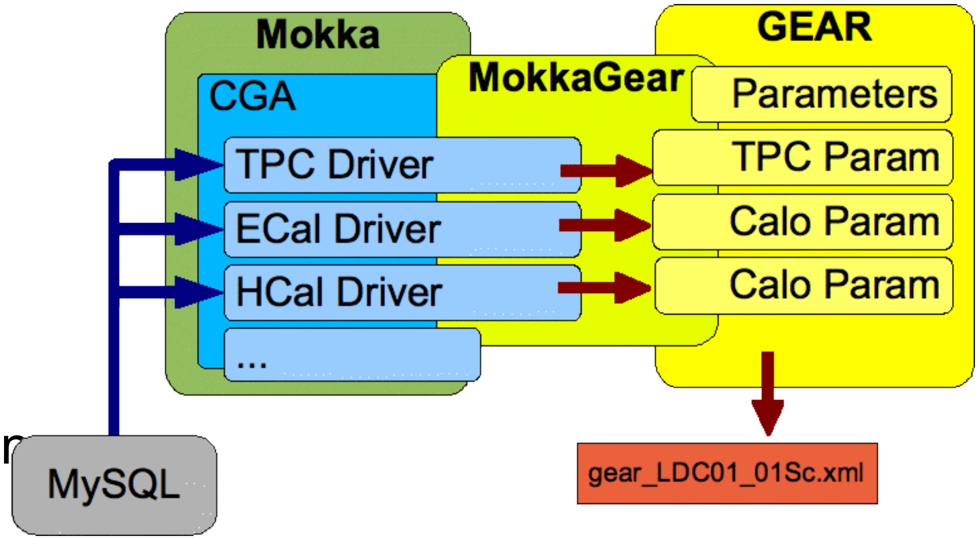

- create DOT files with program flow
- use e.g. graphviz to render into flow chart

Frank Gaede, LDC UK Meeting, Cambridge, September 21, 2007

# geometry for reconstruction

## GEometry API for Reconstruction

- high level abstract interface:
  - per subdetector type (Hcal,TPC,...) parameters/quantities for reco
    - geometry + some navigation
    - implementation uses xml files written from Mokka (simulation)
  - abstract interface for detailed geometry &materials:
    - point properties
    - path properties
    - implementation based on geant4




**MokkaGear**
- enforce only one source of geometry: the simulation program creates the geometry xml files used in reconstruction

( recently improved by K.Harder et al)

# example – GEAR API VXD

Gear: gear::VXDParameters class Reference - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

http://ilcsoft.desy.de/gear/v00-03/doc/html/classgear_1_1VXDParameters.html

simulation/geant4 LCIO Linux Conferences DESY IT Group LEO English/Ger... Google MyHome Ctime

virtual const **VXDLayerLayout** & **getVXDLayerLayout** () const=0
*The layer layout in the Vertex.*

virtual int **getVXDType** () const=0
*The type of Vertex detector: VXDParameters.CCD, VXDParameters.CMOS or VXDPara*

virtual double **getShellHalfLength** () const=0
*The half length (z) of the support shell in mm (w/o gap).*

virtual double **getShellGap** () const=0
*The length of the gap in mm (gap position at z=0).*

virtual double **getShellInnerRadius** () const=0
*The inner radius of the support shell in mm.*

virtual double **getShellOuterRadius** () const=0
*The outer radius of the support shell in mm.*

virtual double **getShellRadLength** () const=0
*The radiation length in the support shell.*

virtual bool **isPointInLadder** (**Point3D** p) const=0
*returns whether a point is inside a ladder*

virtual bool **isPointInSensitive** (**Point3D** p) const=0
*returns wheter a point is inside a sensitive volume*

virtual **Vector3D** **distanceToNearestLadder** (**Point3D** p) const=0
*returns vector from point to nearest ladder*

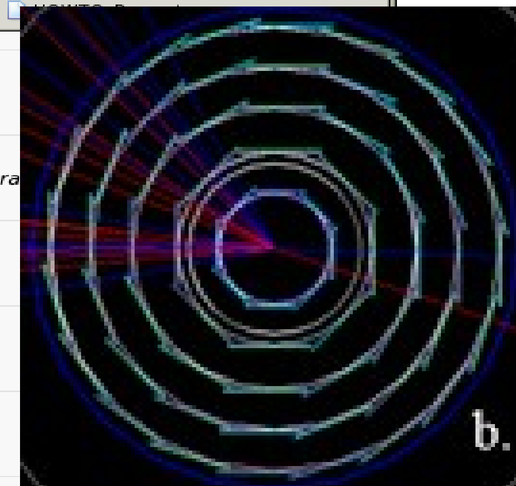
virtual **Vector3D** **distanceToNearestSensitive** (**Point3D** p) const=0
*returns vector from point to nearest sensitive volume*

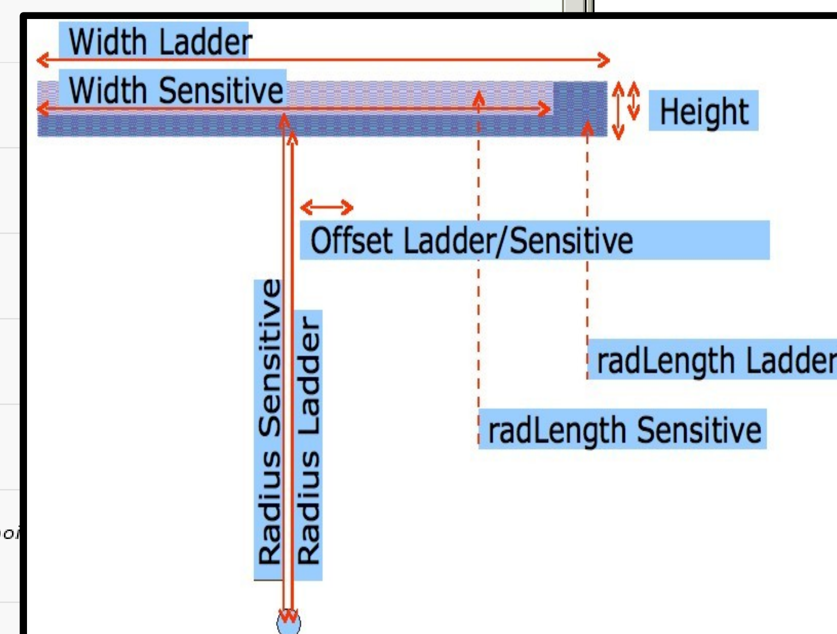
virtual **Vector3D** **intersectionLadder** (**Point3D** p, **Vector3D** v) const=0
*returns the first point where a given strainght line (parameters poi*
*is returned if no intersection can be found.*

virtual **Vector3D** **intersectionSensitive** (**Point3D** p, **Vector3D** v) const=0
*returns the first point where a given strainght line (parameters poi*
*volume (0,0,0) is returned if no intersection can be found.*

Find: VXD Find Next Find Previous Highlight Match case

Done

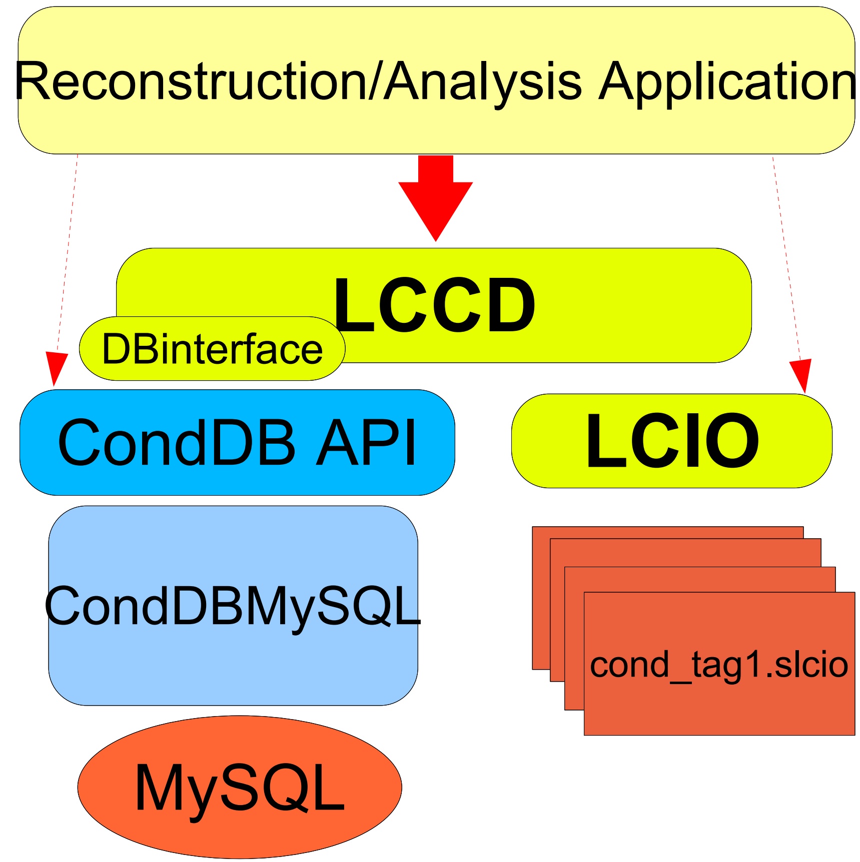


# LCCD

**L**inear **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
  - through creation of LCIO files
    - vertically (all versions for timestamp)
    - horizontally (all versions for tag)




LCCD is used for the conditions data of the ongoing ILC testbeam studies

# ilc sw-installation

- ilc software requirements and complexity has grown
  - ~30 packages with sometimes optional dependencies
- tool to make installation and build process easier:
- **ilcinstall** (python)
  - script to install all of the LDC software in one go
  - "**start script – go to lunch – run application**"
  - fully configurable:
    - versions, dependencies/build options, links to existing packages/tools, e.g. root, CLHEP,...
  - used for reference installations in afs (SL3/SL4)
  - user can link their packages against these
    - even w/o installing any software on their computer

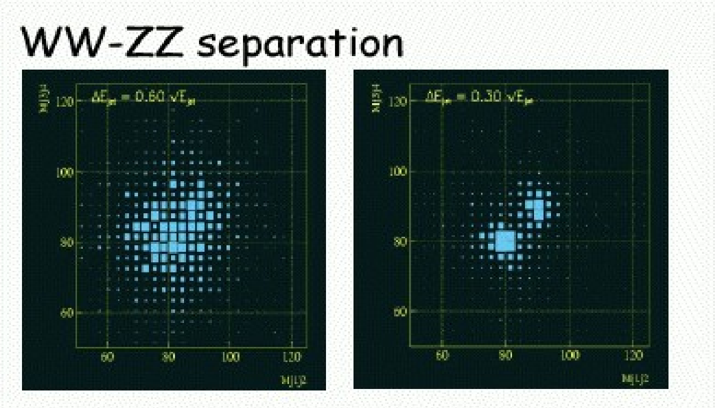
/afs/desy.de/group/it/ilcsoft/v01-01

# Applications of Marlin et al

- LDC detector optimization (MonteCarlo)
  - MarlinReco – full reconstruction suite
    - Digitization Calo,TPC, Silicon, PatternRecognition/Tracking, clustering, ParticleFlow algorithms: **Wolf**, **TrackBased**
  - PandoraPFA
    - ParticleFlow algorithm
  - LCFIVertex
    - ZVTop/ZVKin vertex finding and fitting algorithms
  - various physics analyses ...
- testbeams (Data & MonteCarlo)
  - the LDC software framework has been adopted by and improved within the EUDET project for ILC testbeam infrastructure
    - Calice - calorimeter
    - MarlinTPC – TPC tracking
    - EUTelescope – pixel telescope for silicon tracking

using the same core framework for MC/offline and testbeam/online provides synergies for both worlds

# Reconstruction @ the ILC




- general ILC detector features:
  - precision tracking
  - precision vertexing
  - high granularity in calorimeters
    - ( Ecal ~1cm, Hcal ~1-5cm)
- important: very high jet-mass resolution ~30%/sqrt(E/GeV)

## Particle Flow

- reconstruct all single particles
- use tracker for charged particles
- use Ecal for photons
- use Hcal for neutral hadrons

- dominant contribution (E<50 GeV):
  - Hcal resolution
  - **confusion term**

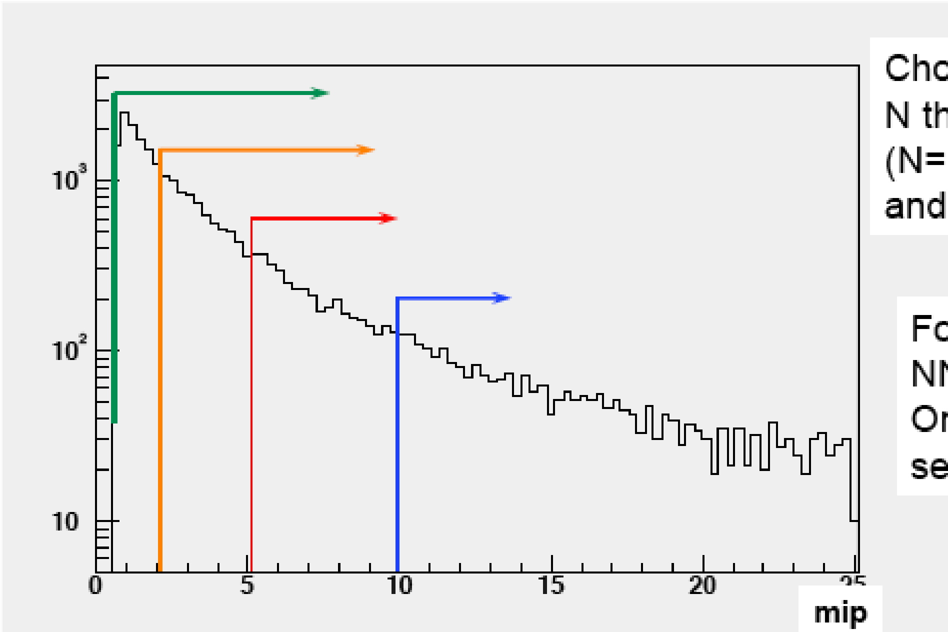
$$\sigma_{E_{jet}}^2 = \epsilon_{trk}^2 \sum_i E_{trk,i}^4 + \epsilon_{ECal}^2 E_{ECal} + \epsilon_{HCal}^2 E_{HCal} + \sigma_{confusion}^2$$

$$\epsilon_{trk} = \delta(1/p) \approx 5\cdot 10^{-5}, \quad \epsilon_{ECal} = \frac{\delta E}{\sqrt{E}} \approx 0.1, \quad \epsilon_{HCal} \approx 0.5$$

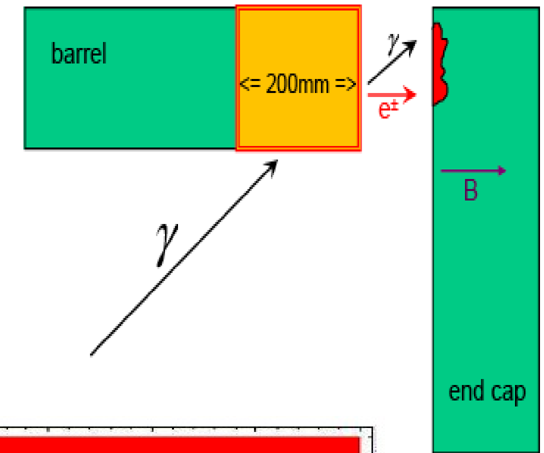
- **PFA performance determines detector resolution**
- need sophisticated algorithms for **minimal confusion**

# example: photon finding in MarlinReco

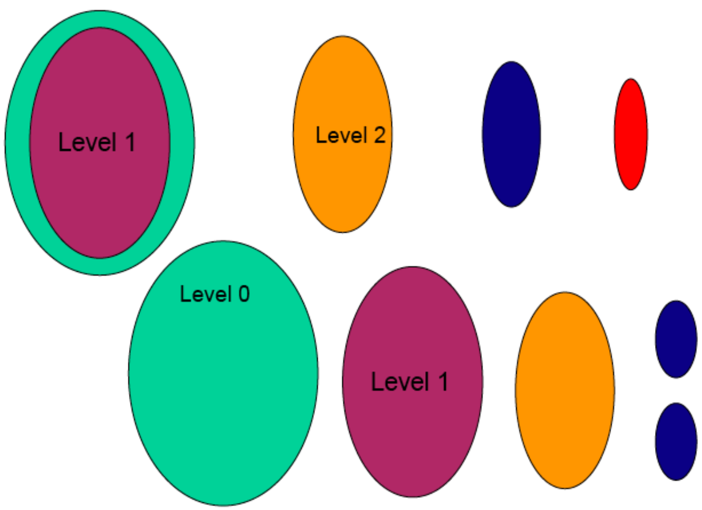
P.Krstonosic



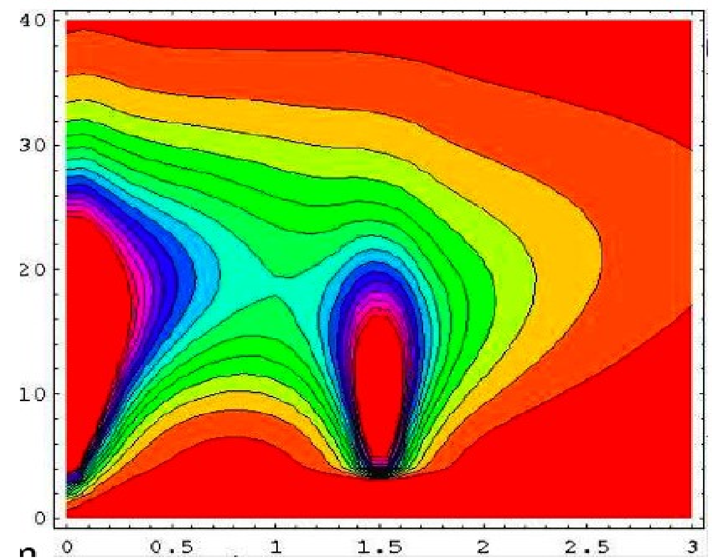

Choose
N threshold levels
(N=10 at the moment)
and get N sets of hits




For each set do a
NN clustering
Only in particular
set!!




Single photon

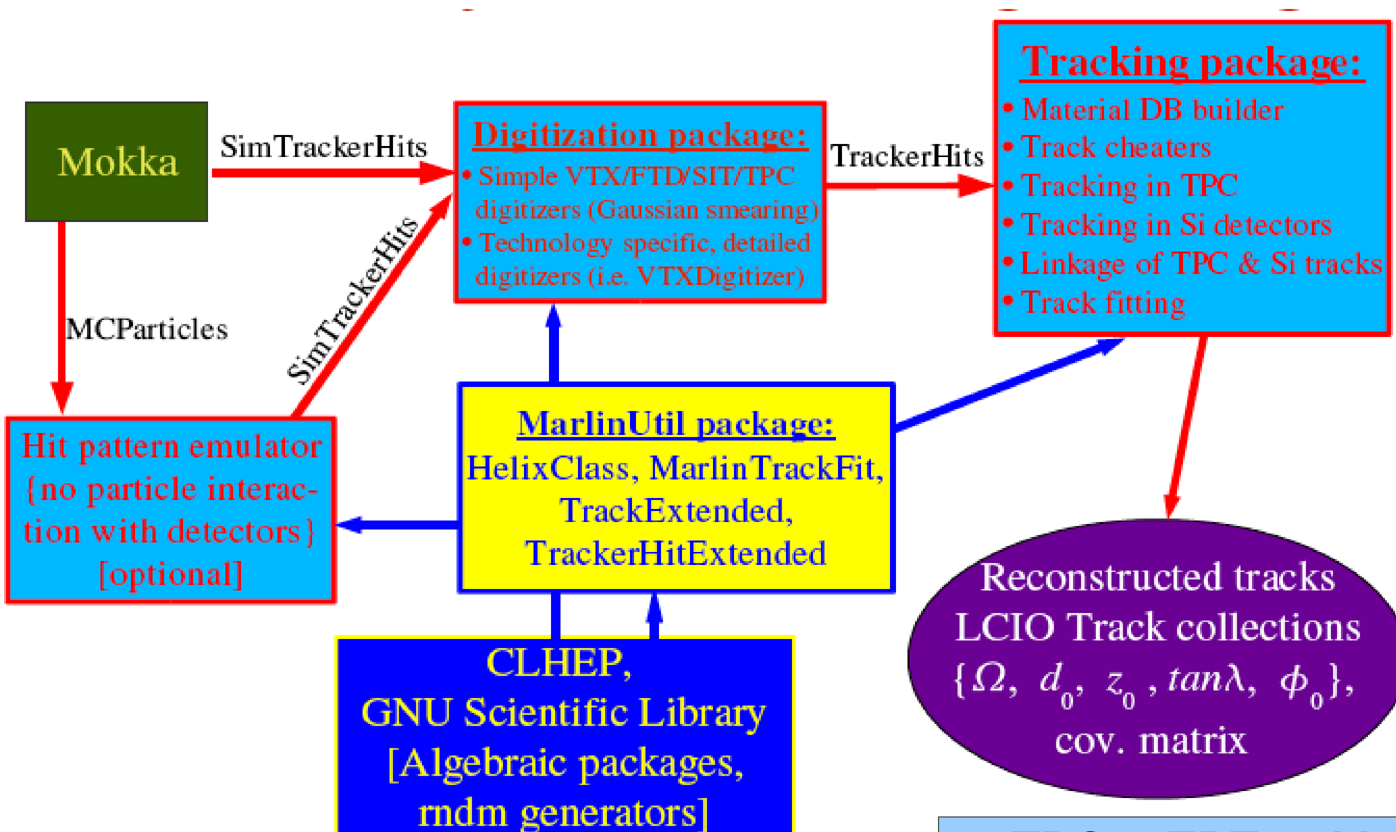


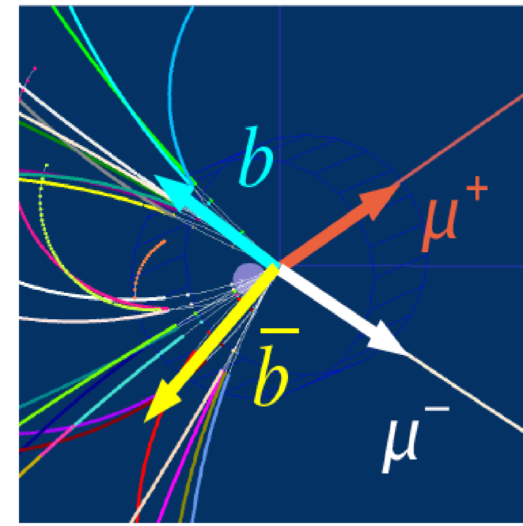

?!

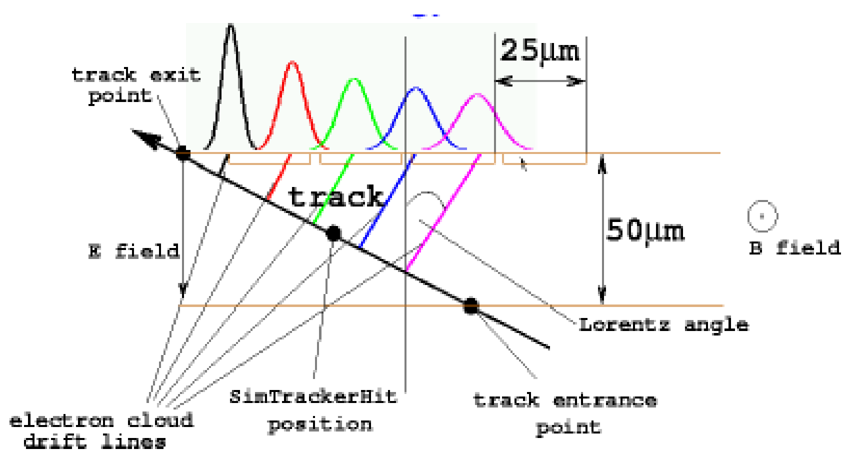
- sophisticated photon ID
- not yet incl. in PFA algorithms

# MarlinReco - FullLDCTracking

A.Raspereza (MPI)



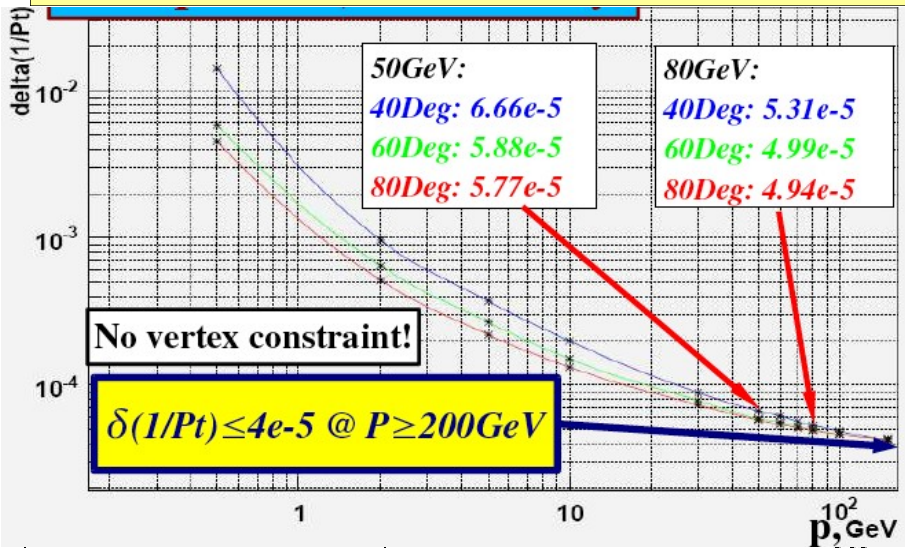



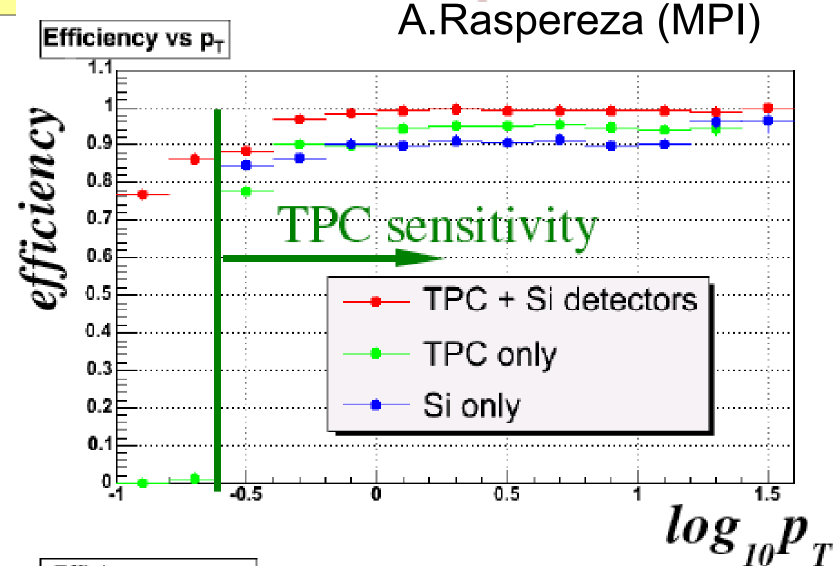



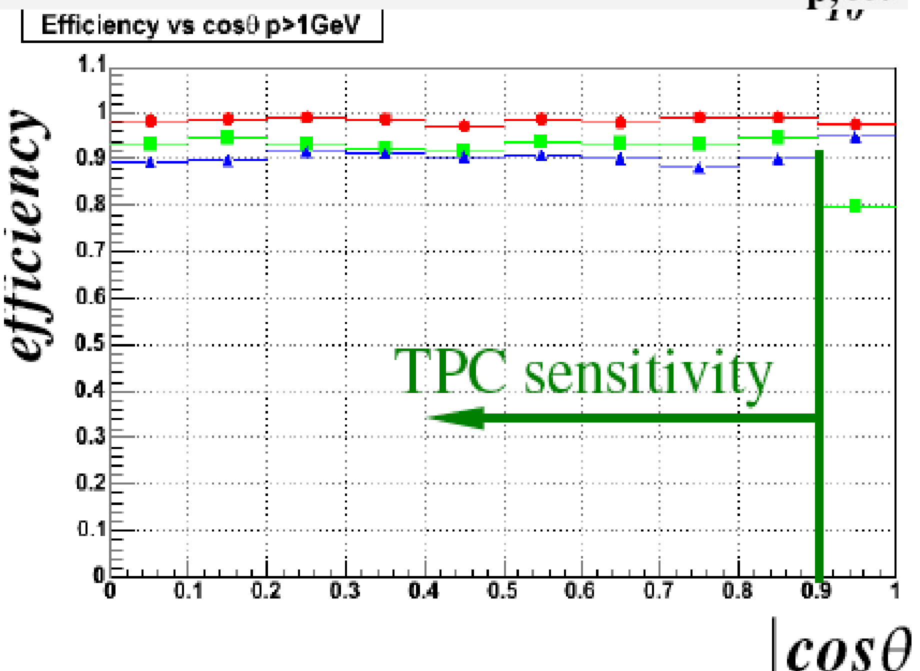

- **TPC: LEPTracking (wrapped LEP code)**
- **VXD, FTD, SIT:**
  - **detailed silicon digitization**
  - **standalone patrec + fitting**
- **LDCTracking:**
  - **combine tracks**
  - **find loopers**
  - **refit (Kalman Filter)**

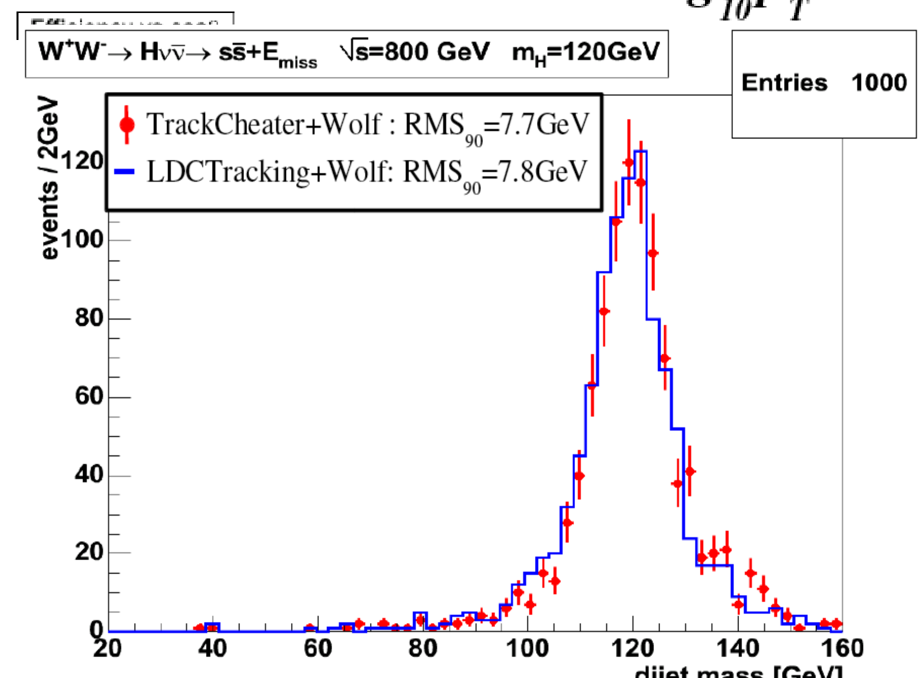
# MarlinReco - FullLDCTracking

A.Raspereza (MPI)



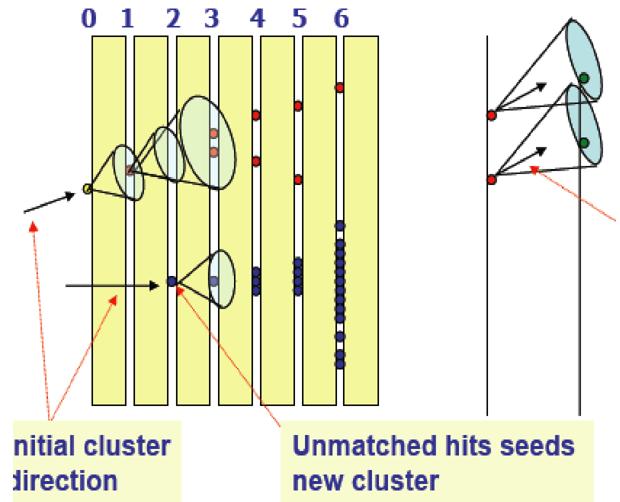

- can now **use real tracking code** and PFA for detector optimization !
- improved recently: coherent Bfield desc., various fit options,...

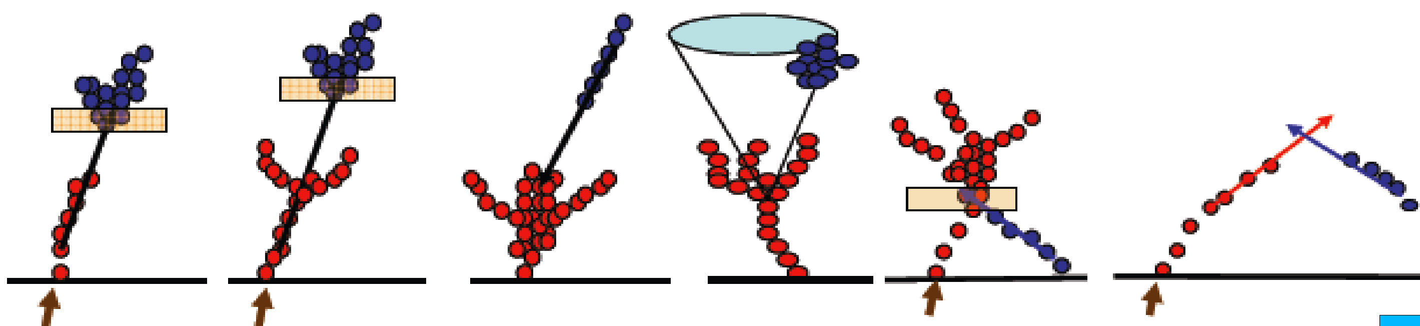
# PandoraPFA

M.Thomson

i. Preparation (MIP hit ID, isolation, tracking)
ii. Loose clustering in ECAL and HCAL
iii. Topological linking of clearly associated clusters
iv. Courser grouping of clusters
v. Iterative reclustering
vi. Photon Recovery (NEW)
vii. Fragment Removal (NEW)
viii. Formation of final Particle Flow Objects (reconstructed particles) – not very sophisticated

Order inter-changable



If track momentum and cluster energy inconsistent : RECLUSTER

e.g.


Pandora is the most sophisticated and best performing PFA to date

# PandoraPFA performance

M.Thomson

rms90

| $E_{JET}$ | $\sigma_E/E = \alpha/\sqrt{(E/GeV)}$ $\lvert\cos\theta\rvert<0.7$ | $\sigma_E/E$ |
|---|---|---|
| 45 GeV | 0.295 | 4.4 % |
| 100 GeV | 0.305 | 3.0 % |
| 180 GeV | 0.418 | 3.1 % |
| 250 GeV | 0.534 | 3.3 % |

[TrackCheater used]

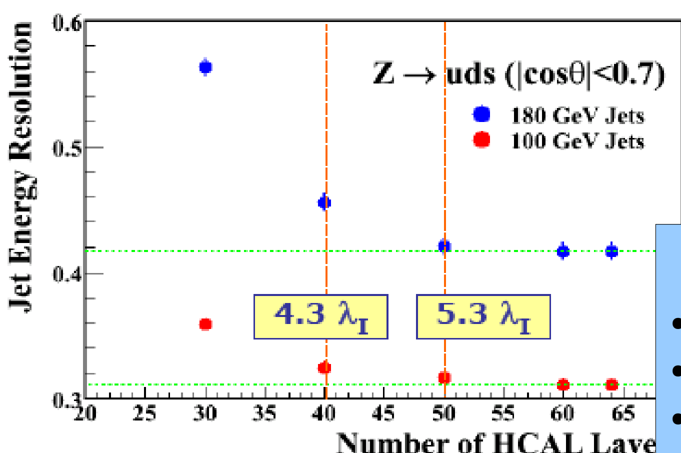
0.35 at LCWS06

**For jet energies < 100 GeV ILC "goal" reached !!!**

★For a Gauge boson mass resolution of order $\Gamma_{W/Z}$

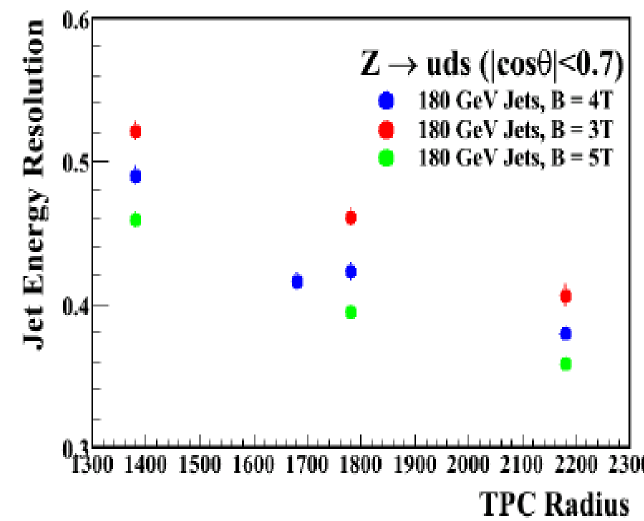
| $E_{jj}$/GeV | $\alpha(E_j)$ | $\sigma_{Ej}/E_j$ |
|---|---|---|
| 91 | < 26 % | 3.8 % |
| 200 | < 38 % | 3.8 % |
| 360 | < 51 % | 3.8 % |
| 500 | < 60 % | 3.8 % |

'proof of concept' for PFA @ILC
-> use for detector optimization

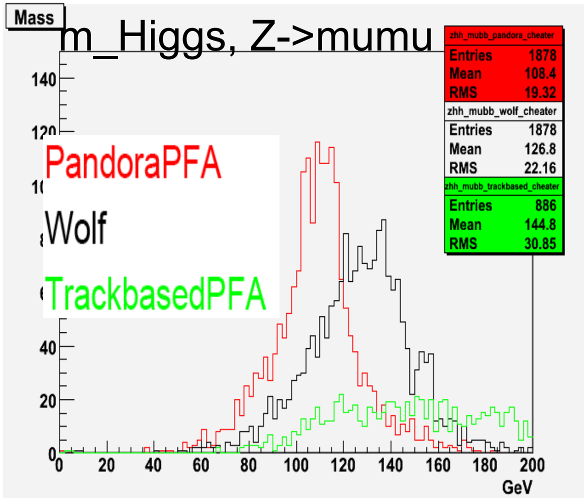


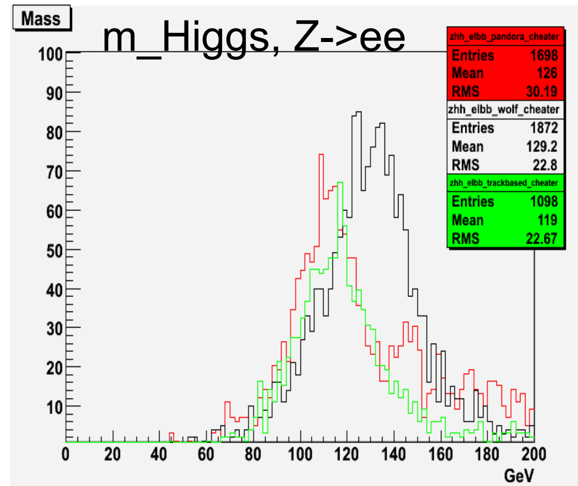

**• PFA improves with:**
- thicker Hcal
- larger Tracking radius
- higher Bfield

- can use PFA for cost conscious optimization

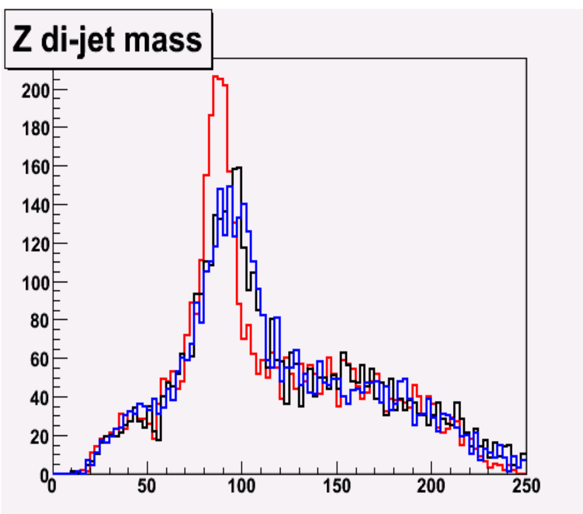


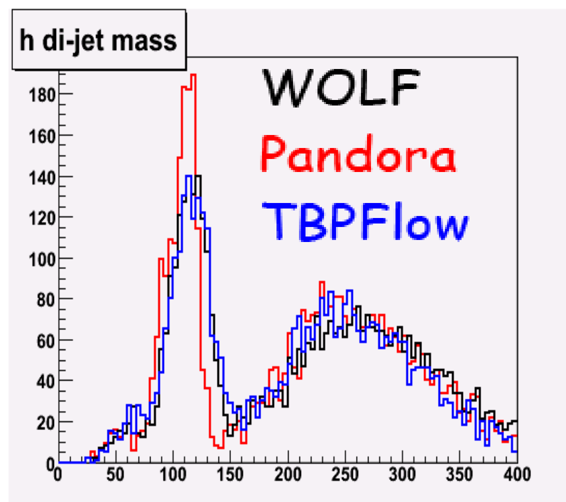
# physics with different PFAs




Michele Faucci Gianelli
ZHH analysis




**SW- tools:**
- Mokka
- MarlinReco/MarlinUtil
- TrackCheater/LDCTracking
- ***PandoraPFA***
- ***Wolf***
- ***TrackBasedPFA***

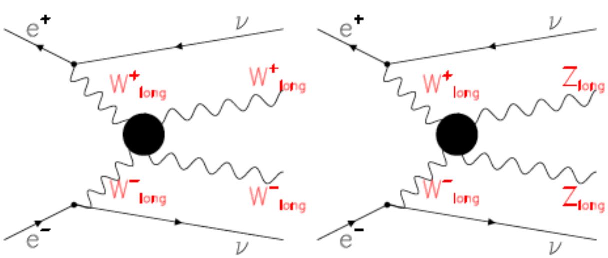
Katarzyna Wichmann
Higgstrahlung analysis

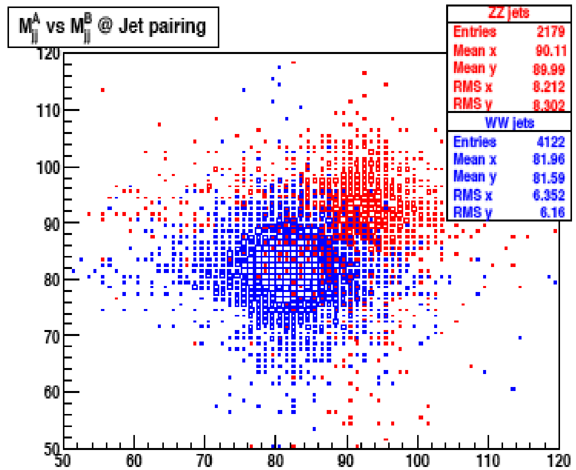
- modular frameworks allow comparison of different (PF)Algorithms
- can use multiple algorithms for cross check of detector optimization
- so far Pandora is best

# WW scattering with full sim-reco

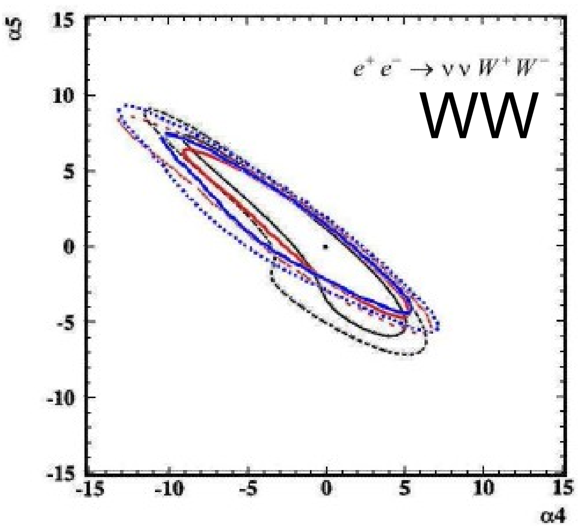
**David Ward and Wenbiao Yan**

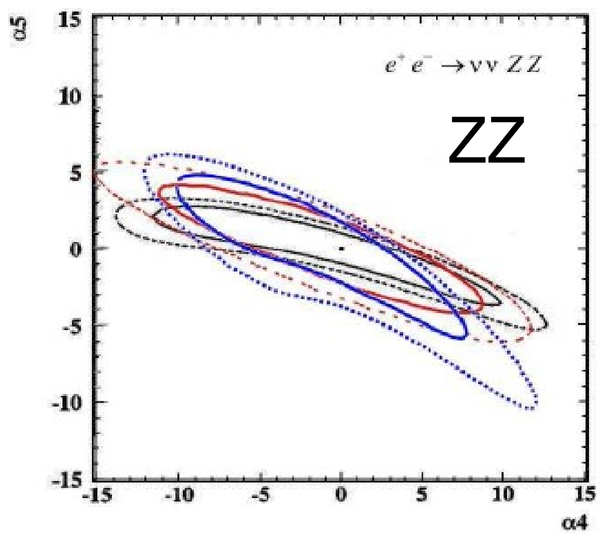
UNIVERSITY OF CAMBRIDGE




**SW- tools:**
- Mokka LDC00Sc
- MarlinReco/MarlinUtil
- TrackCheater
- PandoraPFA




- **Possible improvements**
  - **b-tag** $\to t\bar{t}$ **events**
  - **lepton identification** $\to e\nu WZ$

- full sim & reco
- Tesla fast simulation

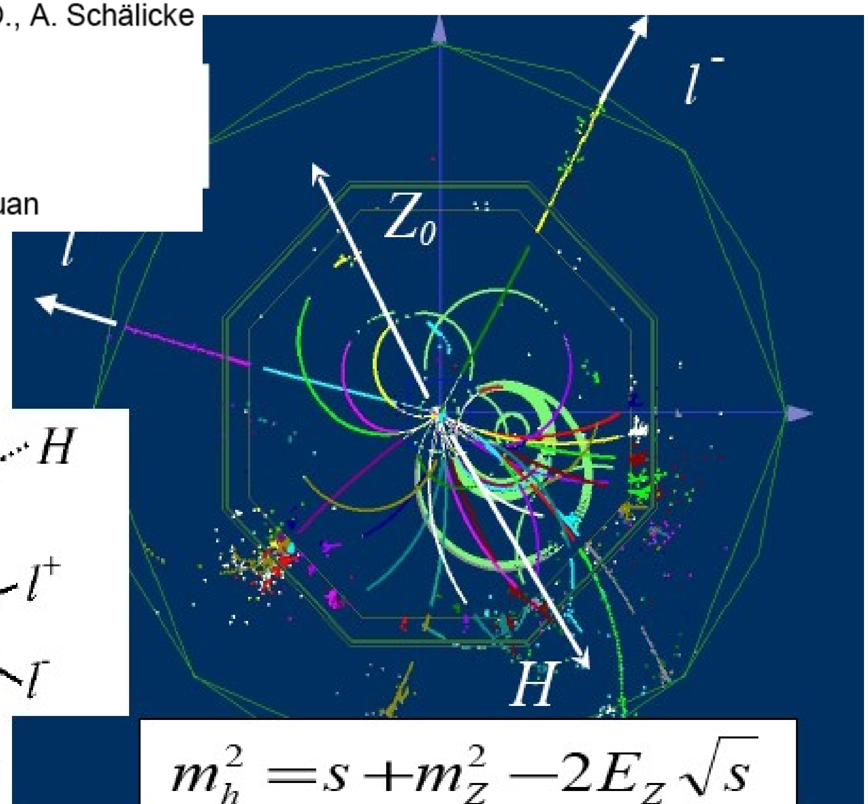
full simualtion and reconstruction tools start to have maturity
to be used for validation of results from fast simulation

# Higgs-Recoils analysis

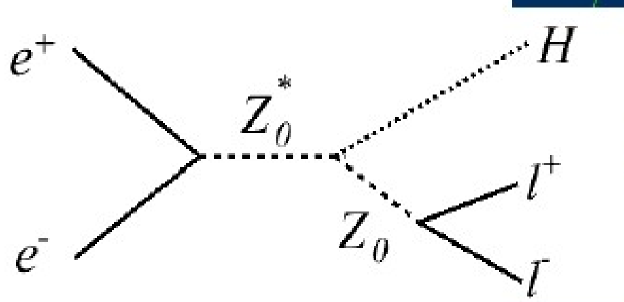
DESY (Zeuthen): W. Lohmann, M.O., A. Schälicke
(Hamburg): K. Wichmann

MPI (München): A. Raspereza

LAL (Orsay): H. Li, R. Pöschl, M. Ruan




- Higgs-Strahlung-Process:




- Higgs-Recoil-Mass

$$m_h^2 = s + m_Z^2 - 2E_Z\sqrt{s}$$

- Coupling Strength (model independent)

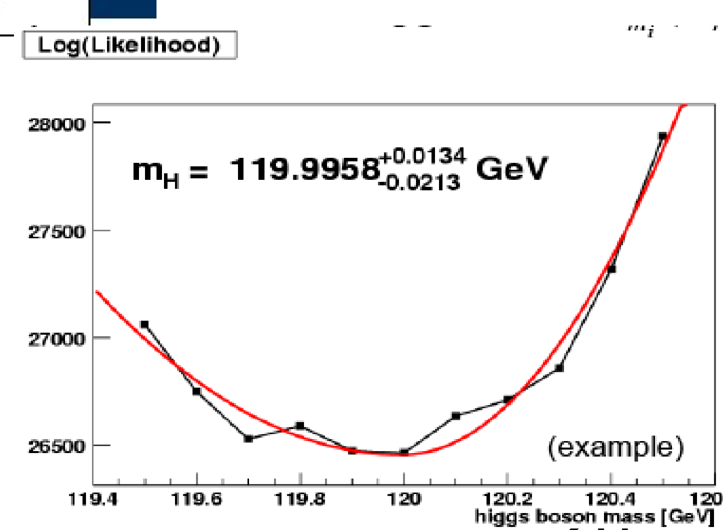
$$g_{ZZH}^2 \propto \sigma = N / L\varepsilon$$

Wolf / Pandora ...

MOID / PFOID

HiggsRecoil

Analysis with ROOT




- particle ID code based on Log-Likelihood method developed
- can be used in other Marlin bases analysis

# ILC and the grid

- Grid computing is the strategic technology for future HEP computing
- significant computing resources will be available in the grid only
- virtual organizations 'ilc' and 'calice' are in place and supported by a growing number of grid sites
  - ~20 sites worldwide
  - all of UK Tier-2 sites
  - calice is using the grid for massive data processing and storage
  - for the rest of ilc only a few power users
  - -> need a coherent approach to run ilc software on the grid
  - to make it a useful tool for everyone in the community

# LDC/ILD and the grid

- some tools for job submission and meta data handling exist
  - already used for MC production (rather rudimentary)
  - xmas05 MC production
- plan: use LHC expertise and tools to improve these
  - Glasgow and Edinburgh groups expressed interest in contributing to this effort (contact with DESY group)
  - use tools like Ganga and Dirac,....
- DESY group about to set up software distribution system for VO ILC
  - ilcsoft releases: Marlin et al, Mokka, ....
  - aim: have full ilcsoft release on all grid workernnodes

# grid Monte Carlo production

- DESY detector optimization group now looking into 'mass production' of LDC Monte Carlo
  - make 500 fb-1 of SM generator 4-vector data files available on the grid (1.7 TB, produced at SLAC)
  - debug and test job submission, data catalogue,..
  - fix issues in Mokka detector description
  - discuss physics benchmark for detector optimization
- input from ILD community needed
  - which channels ?
  - how much data ?
  - how many / which detector models ?

- ideally this should be a real community effort
so that all groups can benefit from the data sample
for their optimization study

# Summary & Outlook

- LDC (ILD) has a mature and easy to use software framework
  - important tools developed in this framework:
    - Vertexing and flavor tagging
    - high performance full Tracking algorithms
    - various PFA algorithms
  - PandoraPFA demonstrates that “PFA works” @ ILC !
  - **use full reconstruction for detector optimization and physics studies !**

**Outlook**

• need to put everything together and create 'standard reconstruction'
• agree on benchmarks for LOI detector optimization
• start 'massive' MonteCarlo production on the grid

**A: yes, the software is ready for physics !**
**It's a good time to get involved in ILD !**