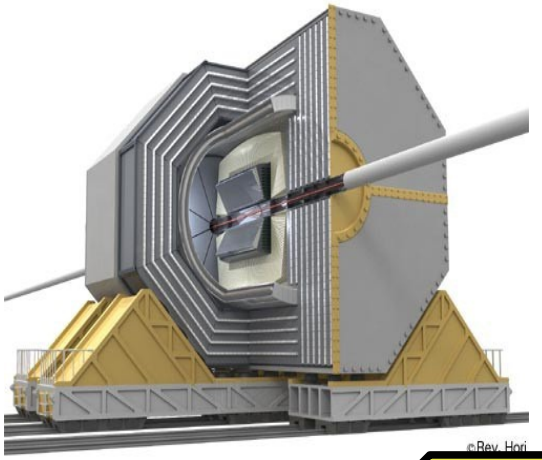
Mokka, Marlin and friends
Status of the ILD_LDC software framework

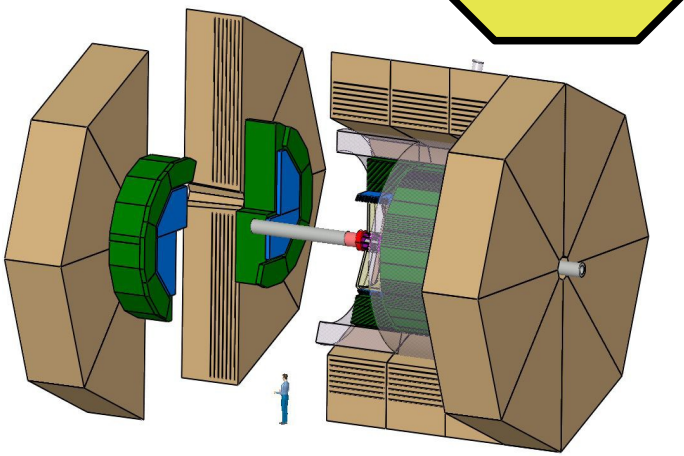
Frank Gaede
DESY
TILC08, Sendai, Japan
3-6 march 2008

Outline

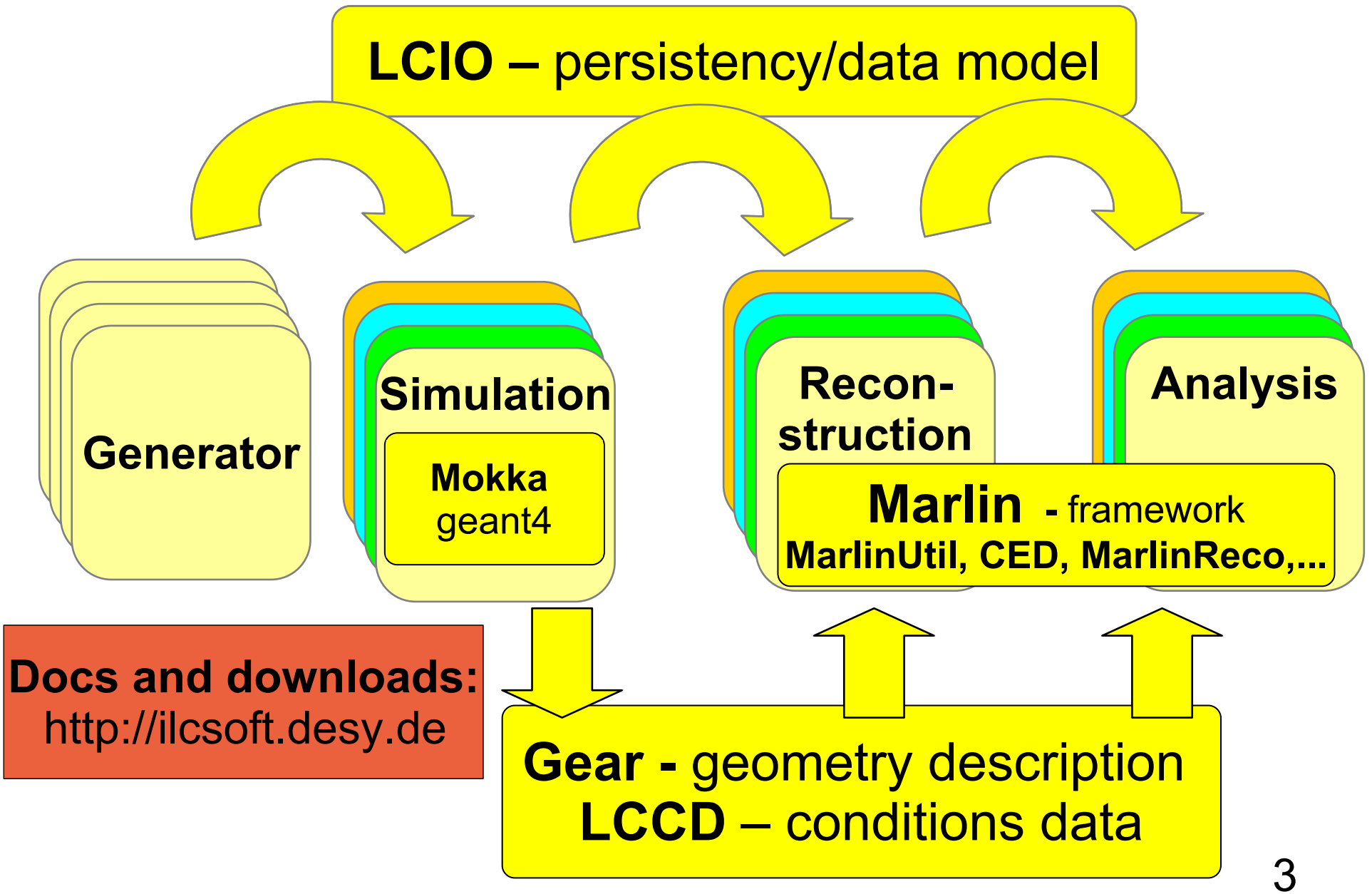
- introduction/overview
- Mokka simulation
- LCIO
- Marlin framework
- full reconstruction
- sw-installation
- grid
- summary



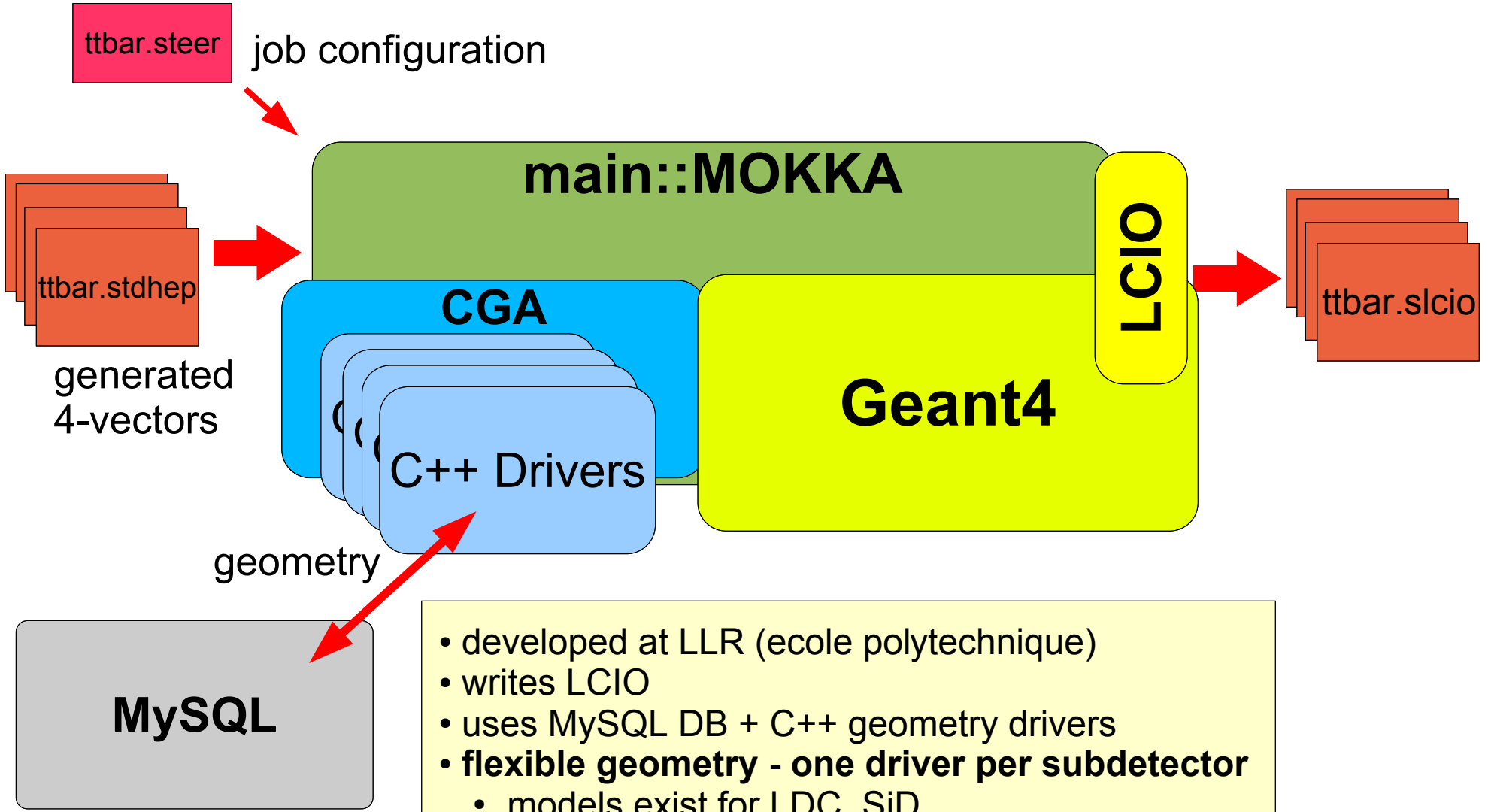
ILD



LDC sw-framework overview

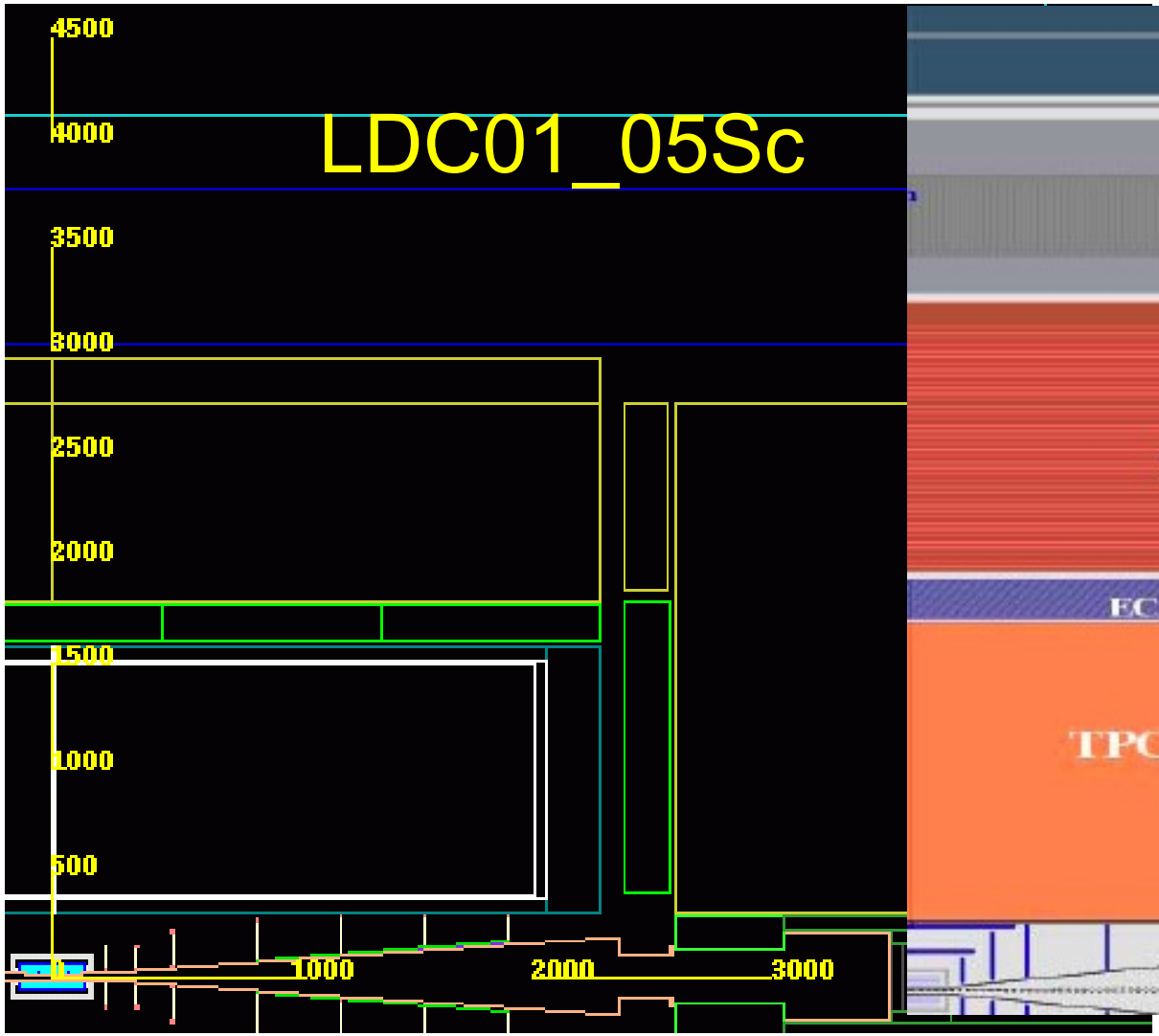


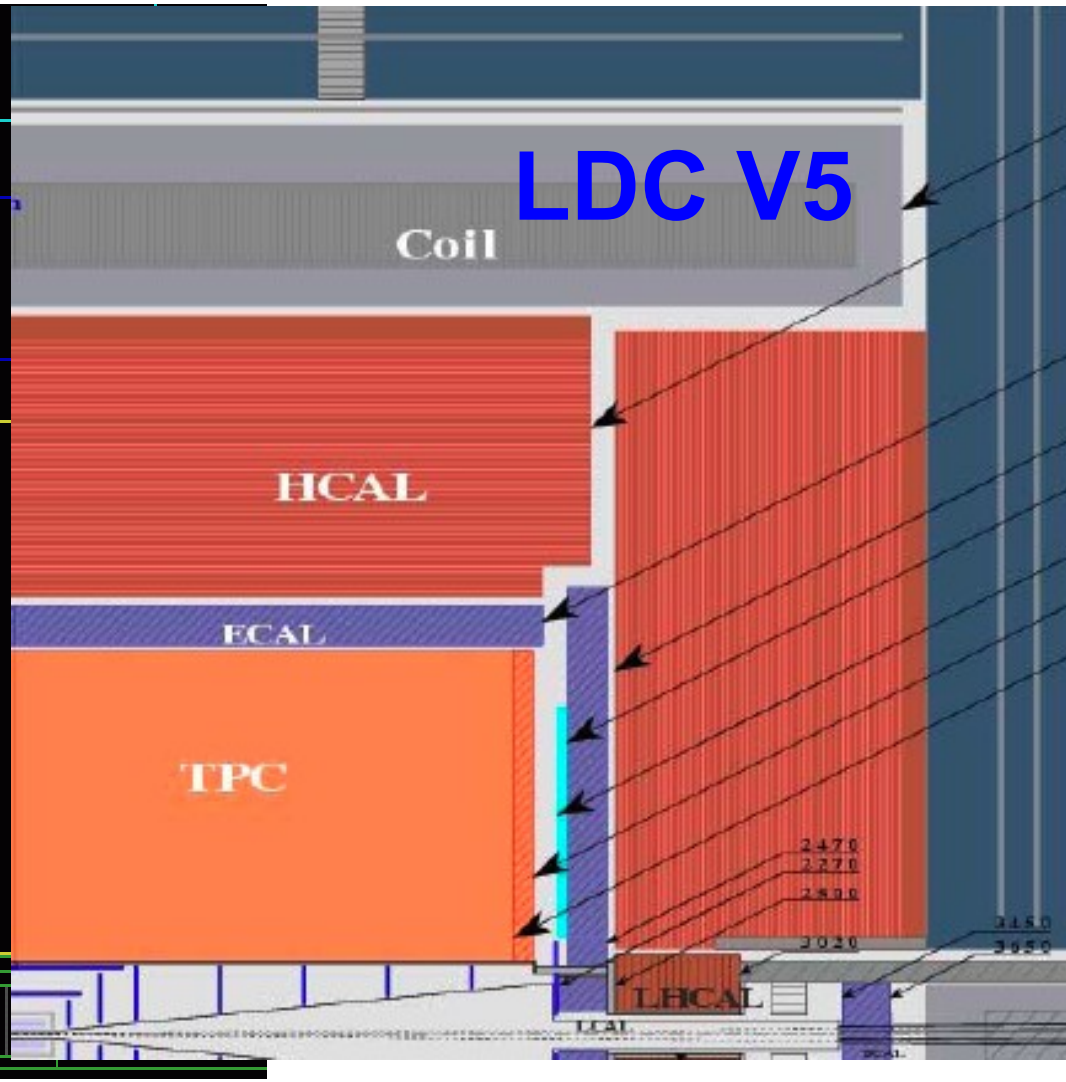
Mokka full simulation - overview



- 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 detector in Mokka

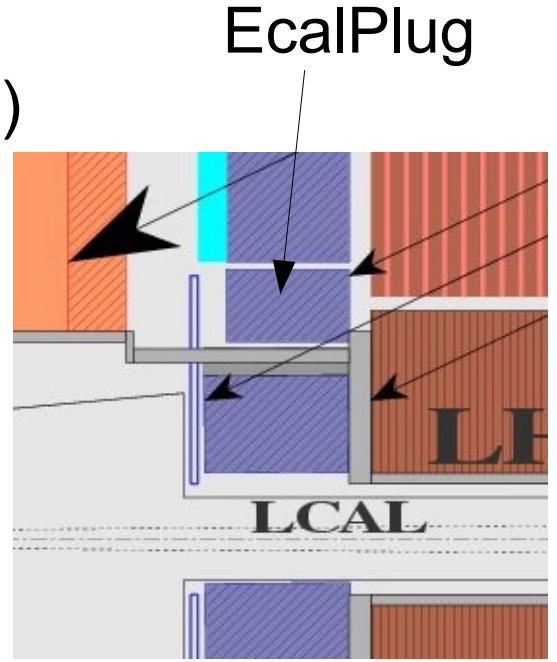


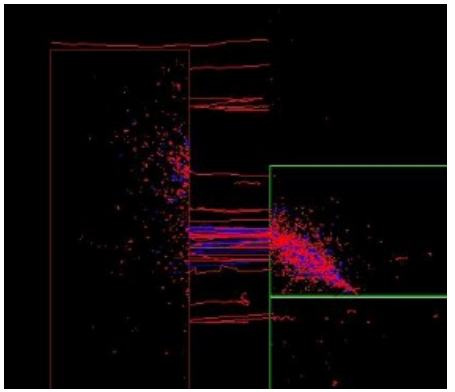


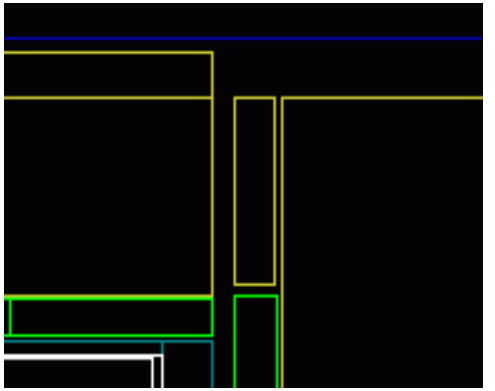
- **a detailed description of LDC exists in Mokka**
- close to proposed design of LDC V5
- lots of fixes and improvements in the last months:
  - HcalRing, Lcal, Ecal, SIT, FTD,..... (see next slide)

Mokka recent developments

- SIT, FTD
  - material & thicknesses improved (V. Saveliev)
- TPC
  - endcap materials, readout (S. Aplin)
- Lcal
  - alignment w/ Ecal & readout (B. Pawlik)
- Ecal
  - inner plug (P. Mora)
- Hcal
  - additional ring (P. Mora)

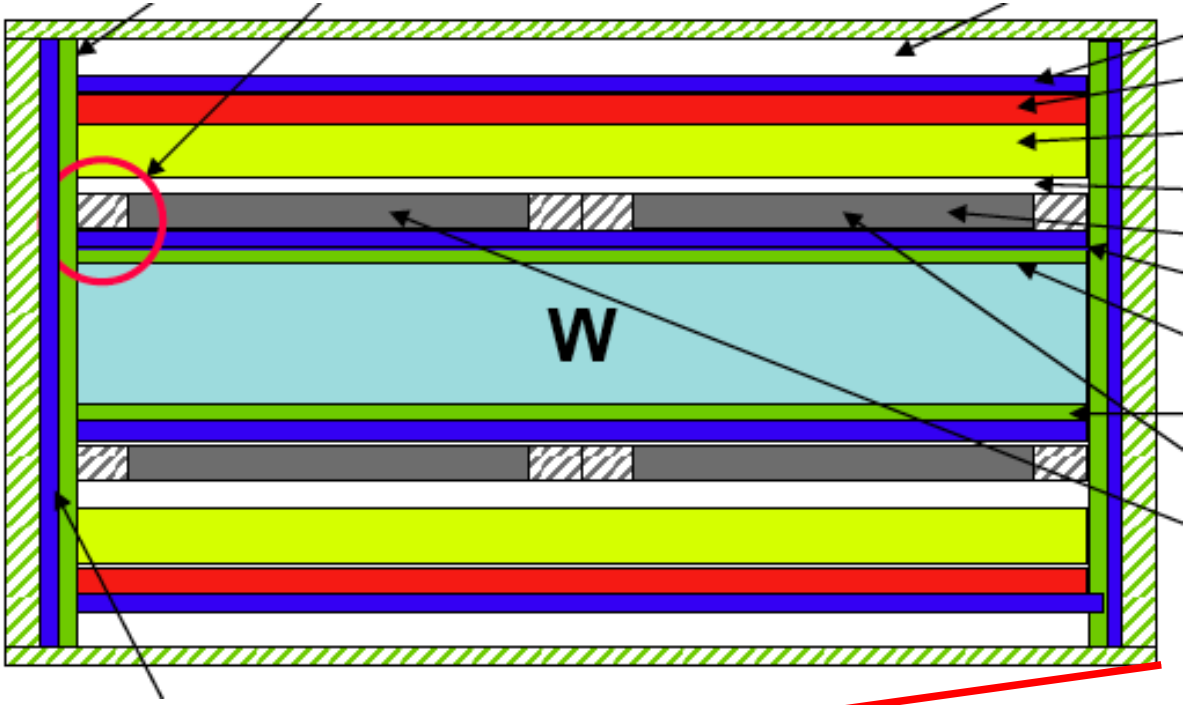


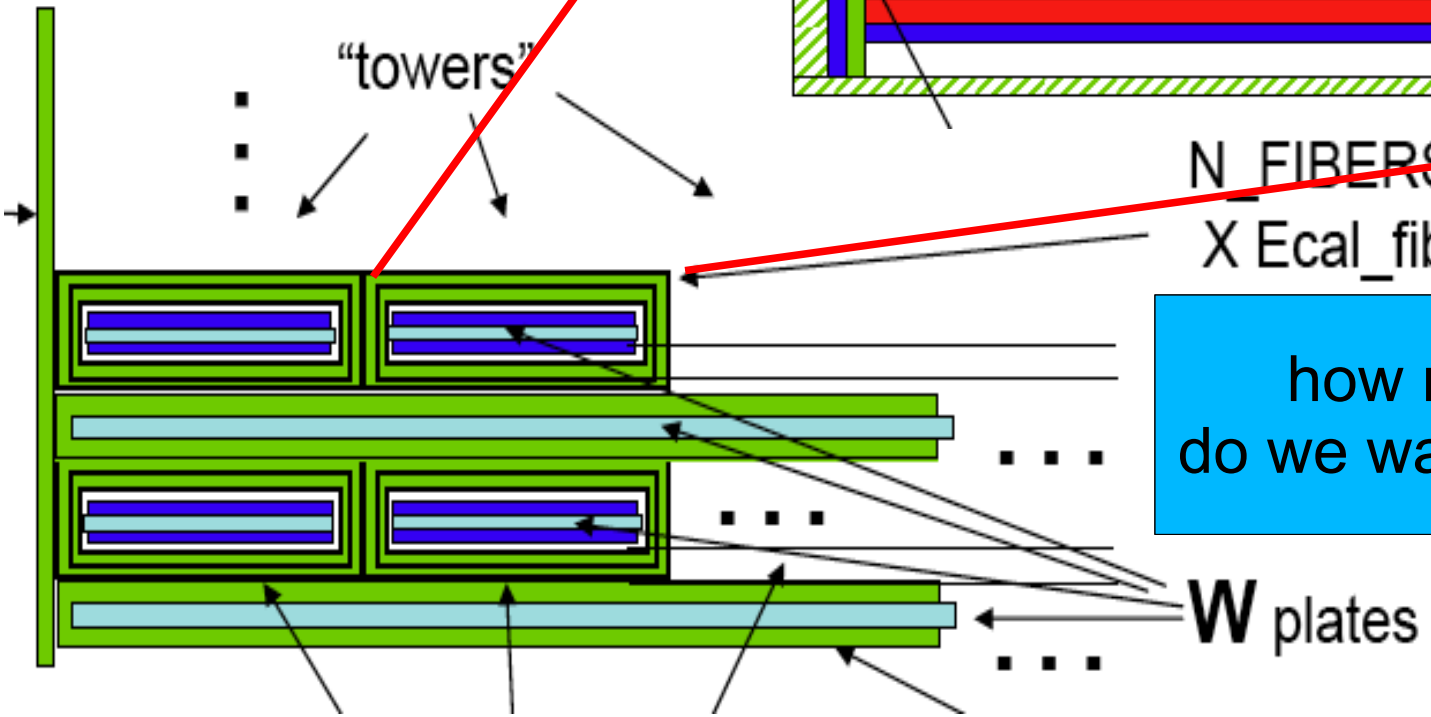




improved Ecal in Mokka (P.Mora)

- description on 'engineering level'
  -> **more realism & reduced performance**

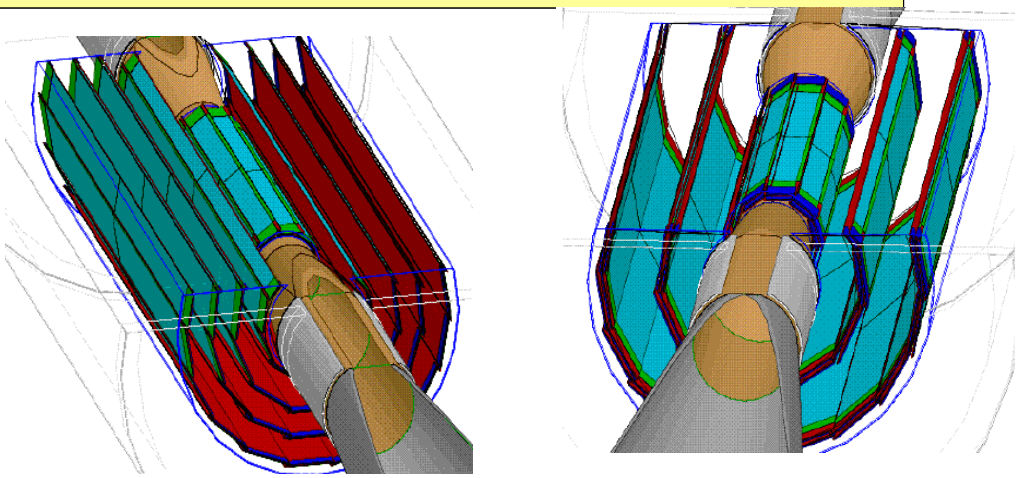




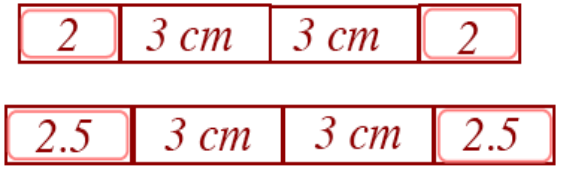
how much realism
do we want/need for LOI ?

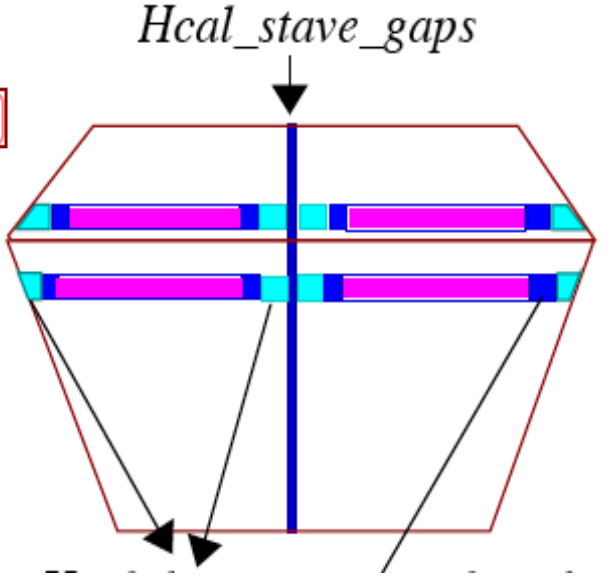
ongoing activities in Mokka

- VTX (D.Grandjean)
  - more flexible driver LDC
  - GLD like layout



- Hcal (A.Lucacci)
  - increase realism:
  - 'additional' cracks in middle of octant
  - realistic tiling





- SIT, FTD (V.Saveliev)
  - make drivers scalable

plan: freeze all development next week for LOI mass production

Mokka for ILD optimization

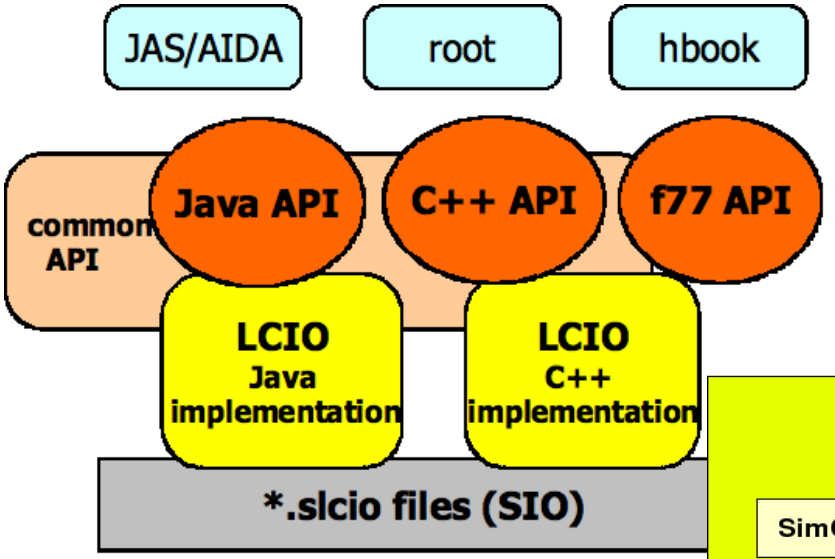
Sub-Detector	Parameter	GLD	LDC	GLD'	LDC'
TPC	R_{inner} (m)	0.45	0.30	0.45	0.30
	R_{outer} (m)	2.00	1.58	1.80	1.80
	Z_{max} (m)*	2.50	2.16	2.35	2.35
Barrel ECAL	R_{inner} (m)**	2.10	1.60	1.85	1.82
	Material	Sci/W	Si/W	Sci/W	Sci/W
Barrel HCAL	Material	Sci/W	Sci/Fe	Sci/Fe	Sci/Fe
Endcap ECAL	Z_{min} (m)***	2.80	2.30	2.55	2.55
Solenoid	B-field	3.0	4.0	3.50	3.50
VTX	Inner Layer (mm)	20	16	18	18

M. Thomson

- created two models for Monte Carlo mass production of LOI benchmark processes:
  - LDC and LDC' initially
    - add more detector variations later for optimization
    - possible due to scaling geometry drivers

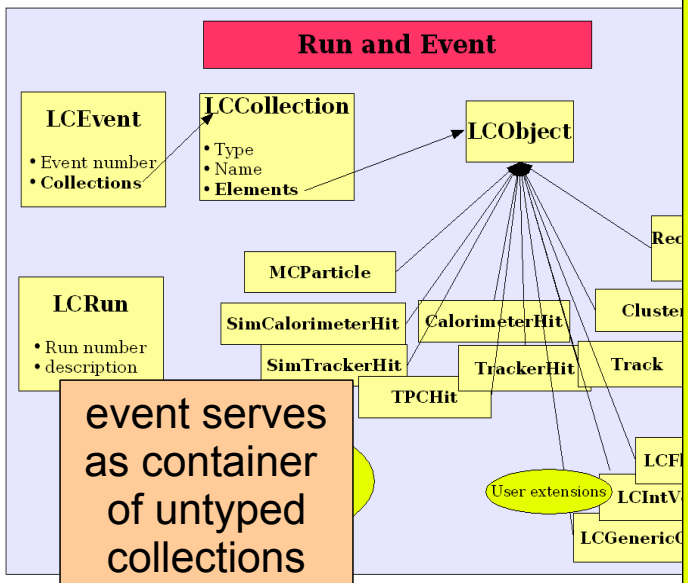
LCIO: persistency & event data model

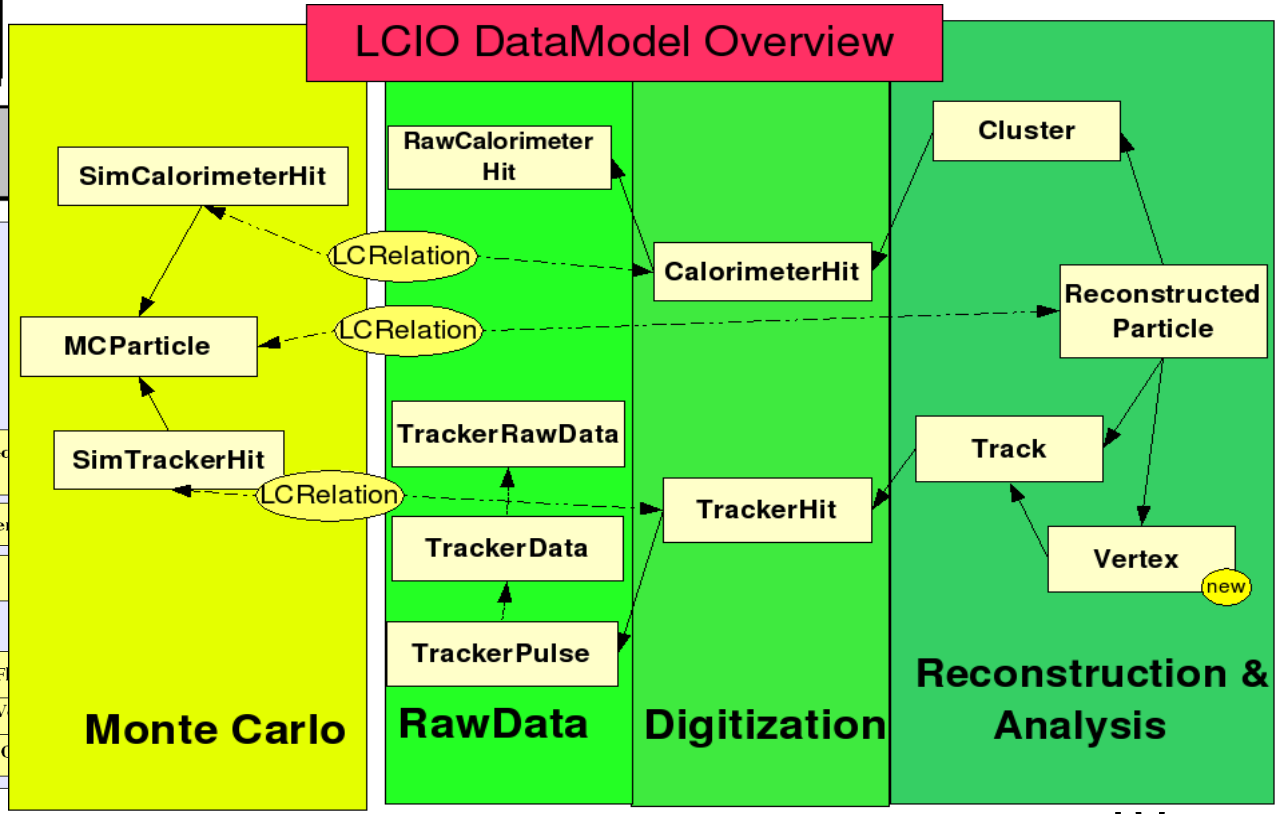
LCIO is the standard output format for LDC/GLD and SID



LCIO recent developments (v01-09):

- preparation for LOI MC-mass production
- improved handling of stdhep generator files
  - bug fixes, stdhep reader
- added event weight to LCEvent
- added experimental support for direct access (C++)
- added runtime extensions (C++)

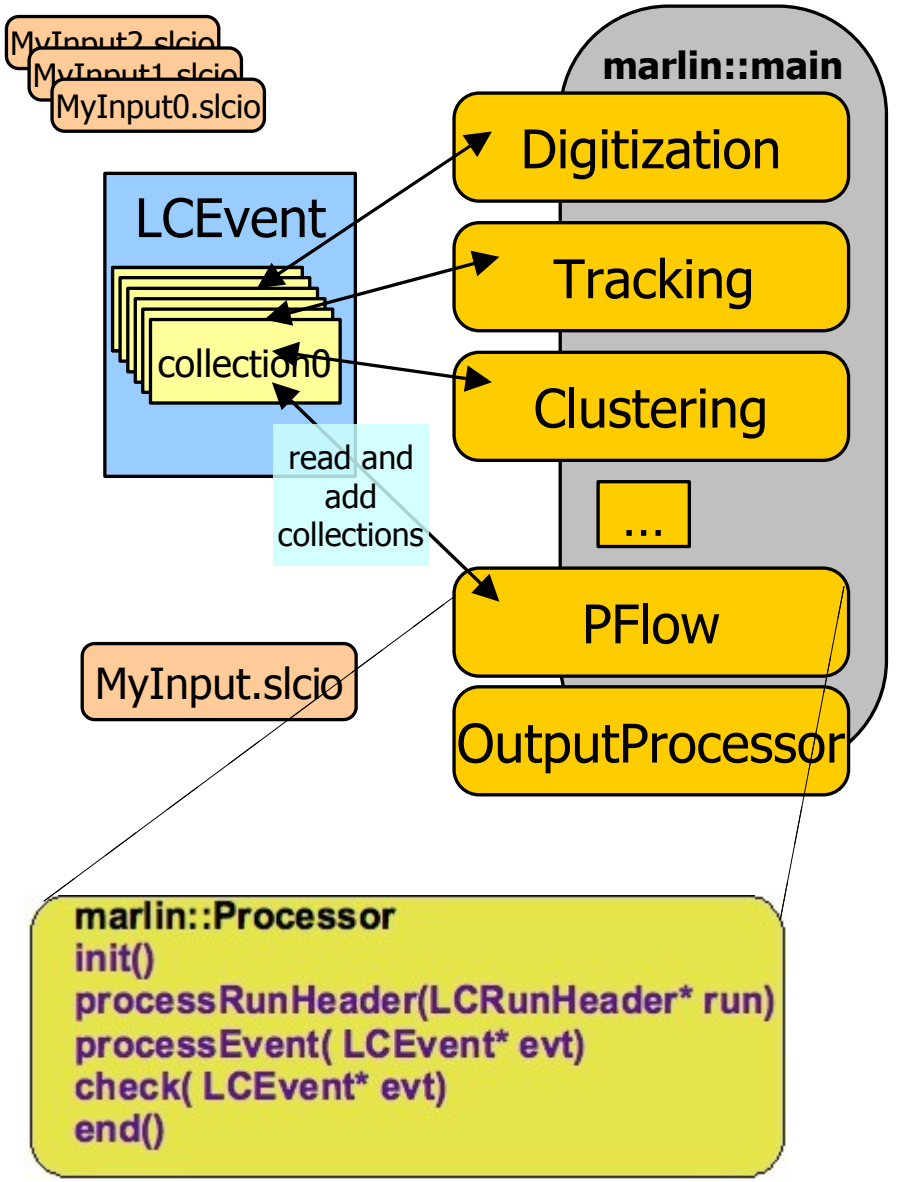




Marlin – application framework

Modular**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

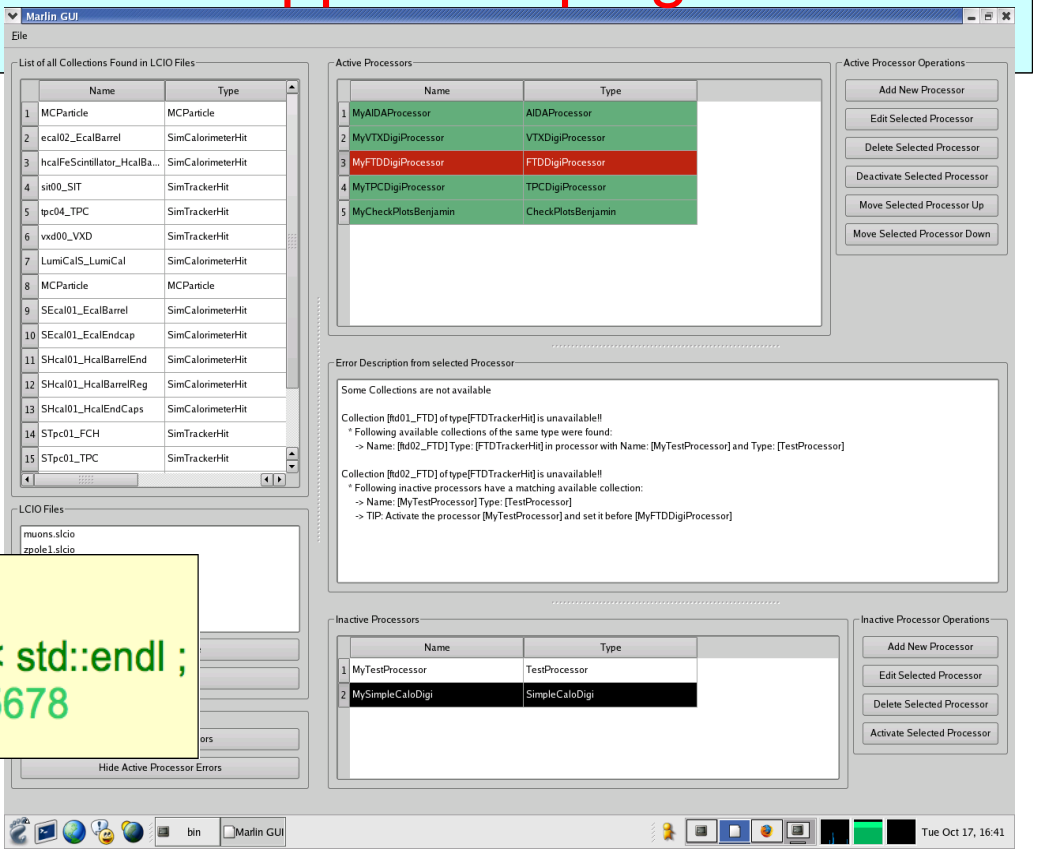


Marlin recent developments

- introduced new build system: CMake
  - ' easy configuration of build process and multi-platform support (Linux, MacOS, Windows)
  - also used for all other core tools
  - -> CMAKE will be the only supported build tool for Marlin et al.
- switched to shared libraries and support for plugins

- MarlinGUI,
- flow charts
- new logging mechanism

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



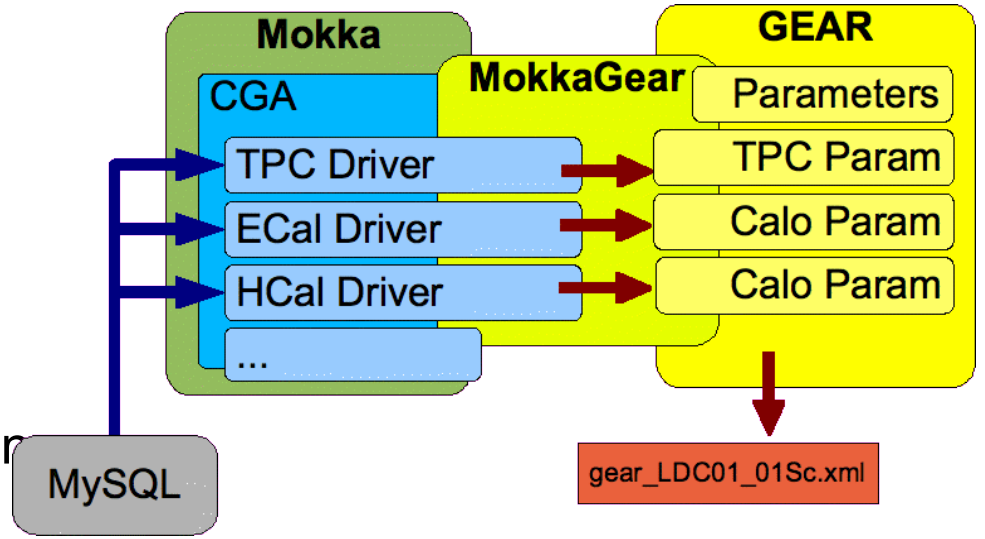
Marlin supporting packages

- the Marlin framework is completed by additional packages for
- description of detector geometry: **GEAR**
- conditions data: **LCCD**
- utility software , math libraries, ...
  - MarlinUtil – utility library
  - CED - event display
  - RAIDA – root AIDA implementation
  - CLHEP, gsl, cernlib,....

geometry for reconstruction

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

- 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

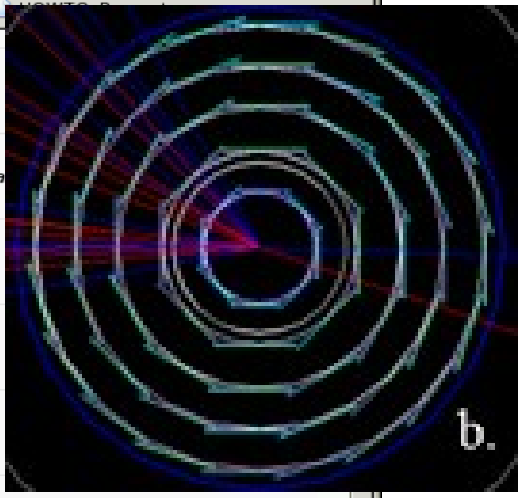
virtual **Vector3D** **distanceToNearestSensitive** (**Point3D** p) const=0
returns vector from point to nearest sensitive volume

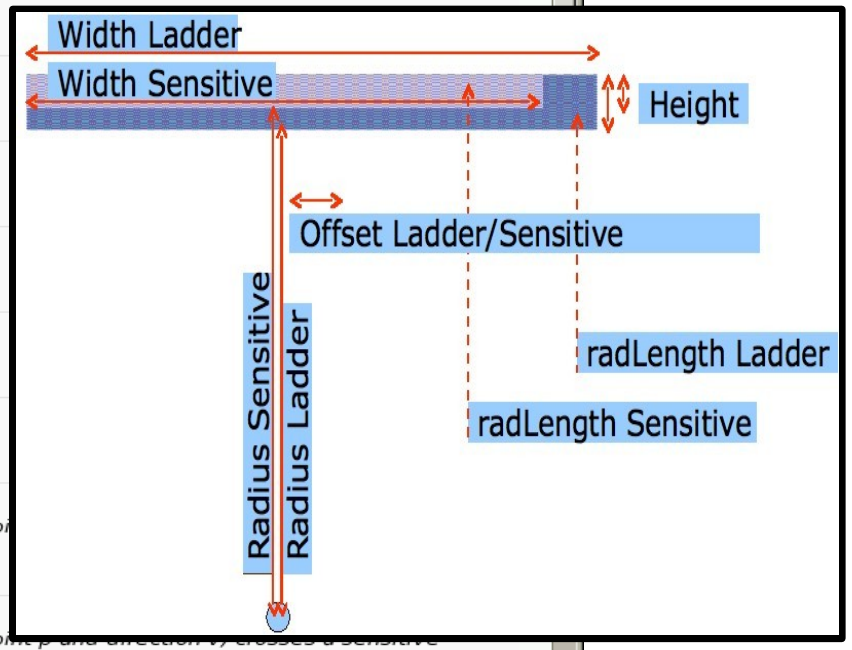
virtual **Vector3D** **intersectionLadder** (**Point3D** p, **Vector3D** v) const=0
returns the first point where a given strainght line (parameters po
is returned if no intersection can be found.

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

Find: VXD Find Next Find Previous Highlight Match case

Done

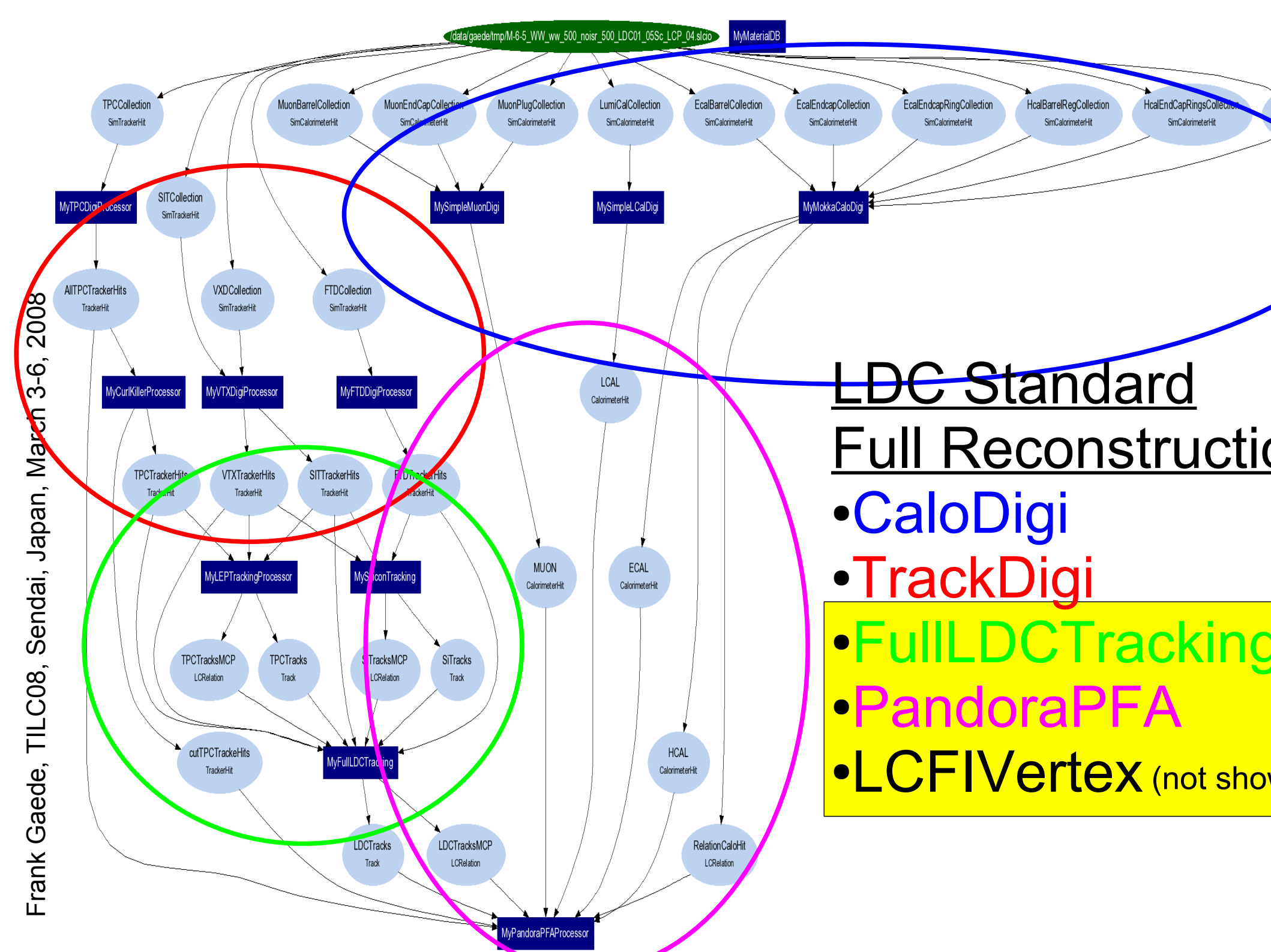




Applications of Marlin et al

- 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)
  - Calice - calorimeter
  - MarlinTPC – TPC tracking
  - EUTelescope – pixel telescope for silicon tracking

using the same core framework for MC/offline and testbeam/online facilitates exchange of knowledge !



Frank Gaede, TILC08, Sendai, Japan, March 3-6, 2008

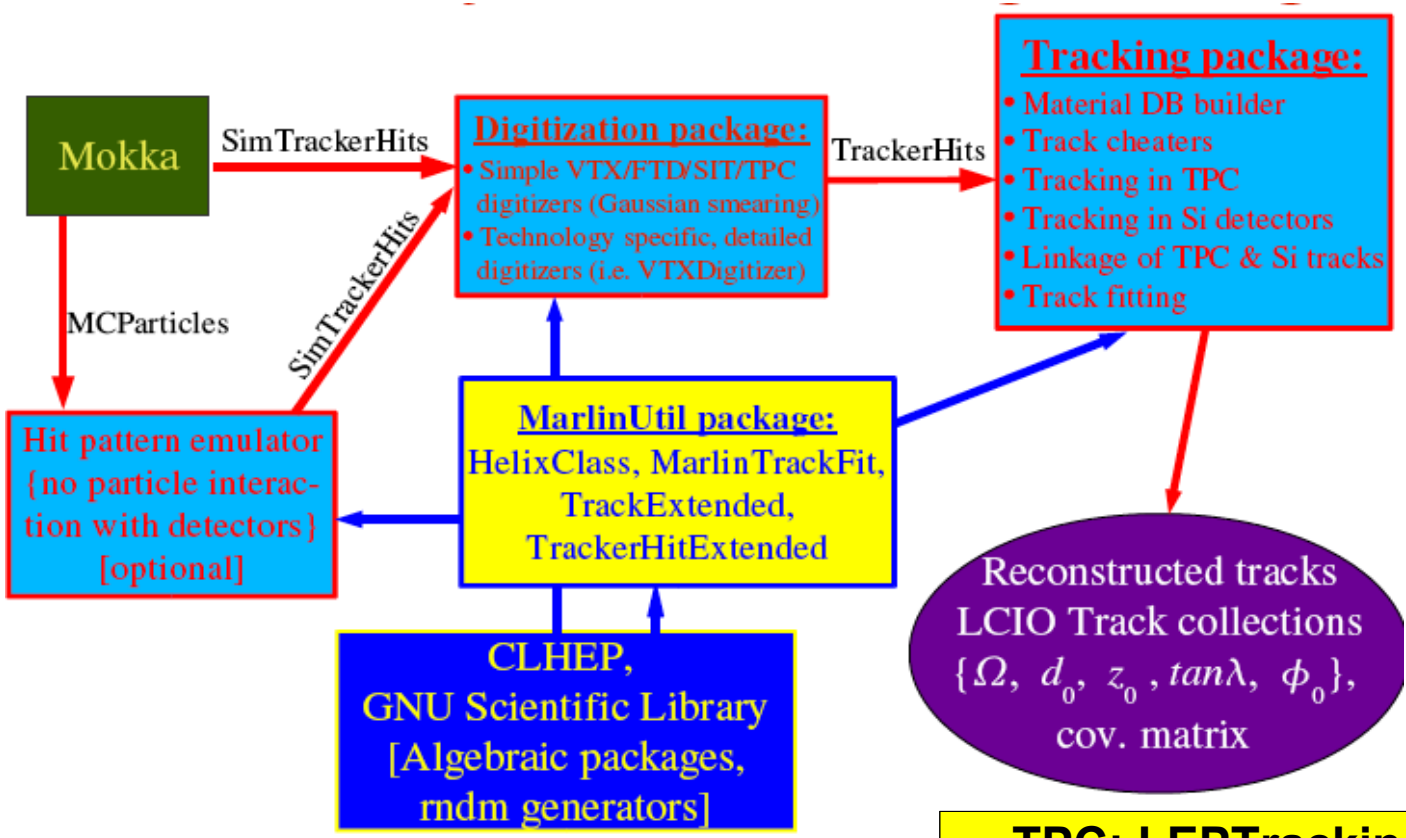
LDC Standard

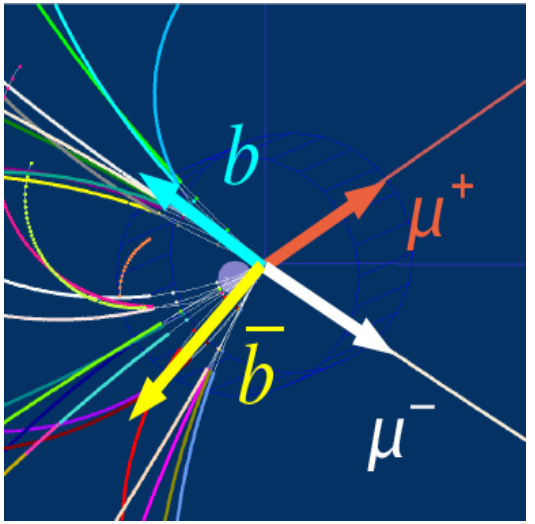
Full Reconstructio

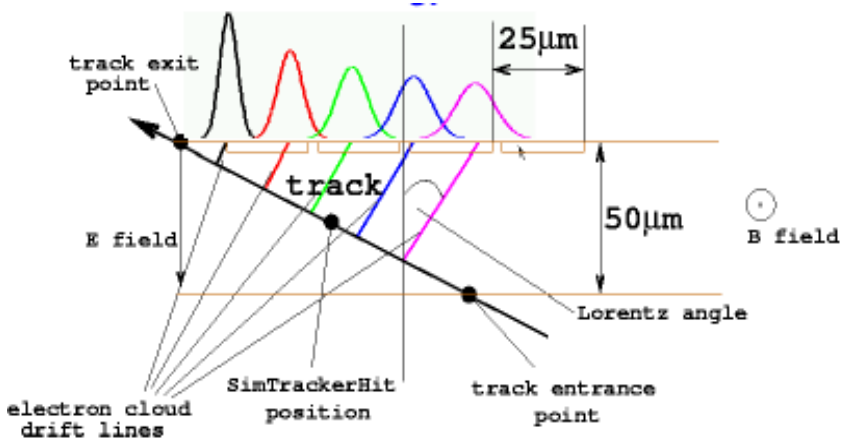
- CaloDigi
- TrackDigi
- FullLDCTracking
- PandoraPFA
- LCFIVertex (not sho

MarlinReco - FullLDCTracking

A.Raspereza (MPI)

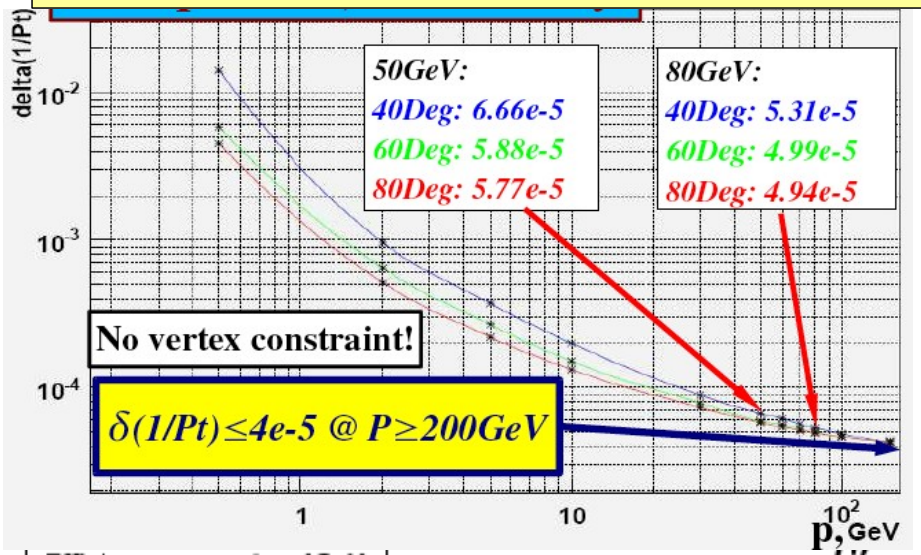




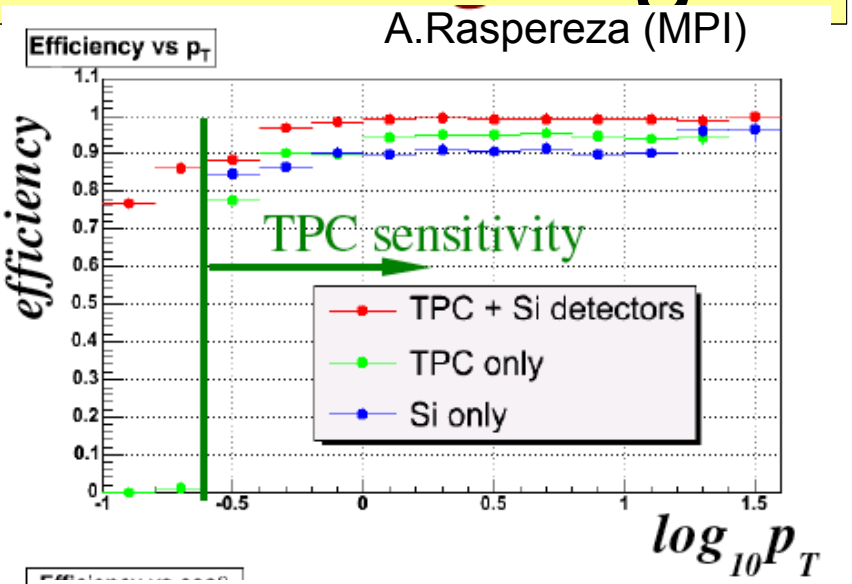


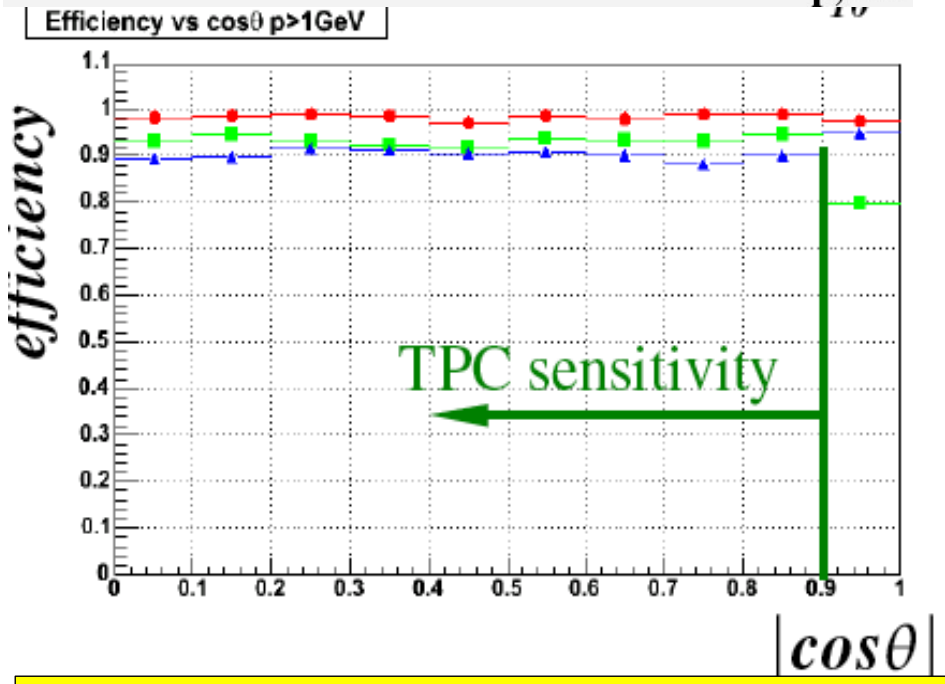
- **TPC: LEPTracking (wrapped LEP code)**
- **VXD, FTD, SIT:**
  - **detailed silicon digitization**
  - **standalone patrec + fitting**
- **LDCTracking:**
  - **combine tracks**
  - **find loopers**
  - **refit (Kalman Filter)**

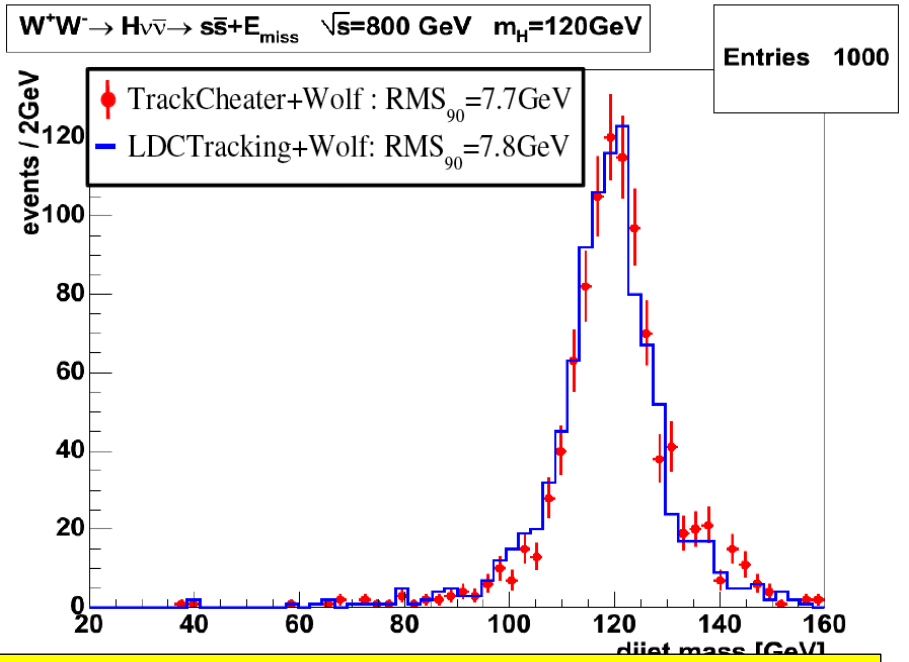
MarlinReco - FullLDCTracking



A.Raspereza (MPI)



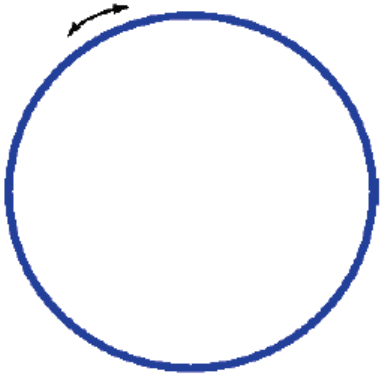




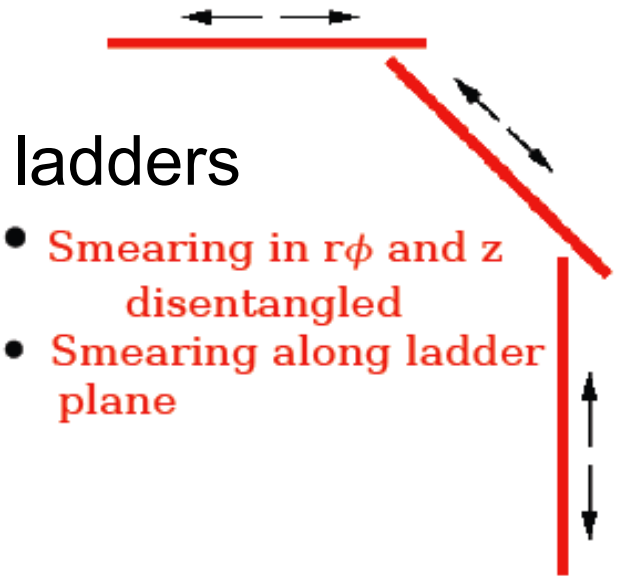
- can now **use real tracking code** and PFA for detector optimization !
- improved recently: coherent Bfield desc., silicon ladders, fit options,...

MarlinReco- recent developments

- VTX – Digitization (C.Lynch)
  - ensure 'r-phi' smearing is done on ladders



vs.



- TPC – Digitization (S.Aplin)
  - improved parameterization of smearing based on input from LC-TPC R&D group:
  - Δr-ø = sqrt(a + b sin(phi)^2 + c/sin(theta)*L_drift)
  - Δz = sqrt(A + B * L_drift)

Status of Marlin based Full Reco

- **new release ilcsoft v01-03 (xmas-07)**
  - all core tools + MarlinReco, PandoraPFA, ...
  - many improvements:
    - FullLDCTracking, event overlay, LCIO direct access, gear description, improved performance (PFA),...
- now put together a 'standard reconstruction' to be used for physics analyses
  - the tools are essentially there
  - -> need proper configuration & calibration
  - need lot of testing by experts
  - already some issues identified...

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/v01-03-02 New

Marlin et al on the grid

- need to use Grid computing for Monte Carlo mass production
  - virtual organizations 'ilc' and 'calice' supported by a growing number of grid sites (~20 sites worldwide)
- need to develop infrastructure to make use of the grid resources:
  - job submission scripts
  - file catalogue (data base)
  - grid software installations
  - -> ongoing at DESY (see talk in ILD-meeting on Thursday)

grid sites with ilcsoft v01-03-02

CE	SW-VER	SW-OS	DATE	TIME	SAM	JOB	TAGGED	HIST-LOGS
cclcgceli02.in2p3.fr	v01-03-02	sl4	2008-02-29	18-54-12	OK	OK	VO-ilc-ilcsoft-v01-03-01-sl4 VO-ilc-ilcsoft-v01-03-02-sl4	History
cclcgceli03.in2p3.fr	v01-03-02	sl4	2008-02-29	18-54-12	OK	OK	VO-ilc-ilcsoft-v01-03-01-sl4 VO-ilc-ilcsoft-v01-03-02-sl4	History
dg10.cc.kek.jp	v01-03-02	sl3	2008-03-03	12-38-40	OK	OK	VO-ilc-ilcsoft-v01-03-02-sl3	History
grid-ce3.desy.de	v01-03-02	sl4	2008-02-29	18-54-12	OK	OK	VO-ilc-ilcsoft-v01-03-01-sl4 VO-ilc-ilcsoft-v01-03-02-sl4	History
hepinx206.pp.rl.ac.uk	v01-03-02	sl4	2008-02-29	18-54-12	OK	OK	VO-ilc-ilcsoft-v01-03-01-sl4 VO-ilc-ilcsoft-v01-03-02-sl4	History
lcg-ce0.ifh.de	v01-03-02	sl4	2008-02-29	18-54-12	OK	OK	VO-ilc-ilcsoft-v01-03-01-sl4 VO-ilc-ilcsoft-v01-03-02-sl4	History
lcg-ce1.ifh.de	v01-03-02	sl4	2008-02-29	18-54-12	OK	OK	VO-ilc-ilcsoft-v01-03-01-sl4 VO-ilc-ilcsoft-v01-03-02-sl4	History
lcgce02.gridpp.rl.ac.uk	v01-03-02	sl4	2008-02-29	18-54-12	OK	OK	VO-ilc-ilcsoft-v01-03-01-sl4 VO-ilc-ilcsoft-v01-03-02-sl4	History
node07.datagrid.cea.fr	v01-03-02	sl4	2008-02-29	18-54-12	OK	OK	VO-ilc-ilcsoft-v01-03-01-sl4 VO-ilc-ilcsoft-v01-03-02-sl4	History
t2ce03.physics.ox.ac.uk	v01-03-02	sl4	2008-02-29	18-54-12	OK	OK	VO-ilc-ilcsoft-v01-03-01-sl4 VO-ilc-ilcsoft-v01-03-02-sl4	History

http://ilcsoft.desy.de/grid/results/User.html

Initial MC production

- plan: have central production of LDC/LDC' MC files
- need to test the full machinery:
  - simulation (Mokka)
  - grid tools
  - Marlin reconstruction code
- start now with some simple events:
  - singles O(10k) of g, e, mu+-, pi+-, K_l, K_s
  - O(10k) Z->uds @ 90,250,500 GeV
  - O(10k) ZH @ 250, 500 GeV
  - both for LDC and LDCPrime
- -> use to put together, configure and test a standard reconstruction

Summary

- LDC (ILD) has a mature and powerful software framework based on Mokka, LCIO, Marlin
- tools for full reconstruction are now there:
  - high performance full Tracking algorithms
  - various PFA algorithms
  - PandoraPFA
  - LCFIVertex
  - **will use full reconstruction for detector optimization and physics studies for LOI**

Outlook

- need to put everything together and create 'standard reconstruction'
- test and validate
- start 'massive' MonteCarlo production on the grid