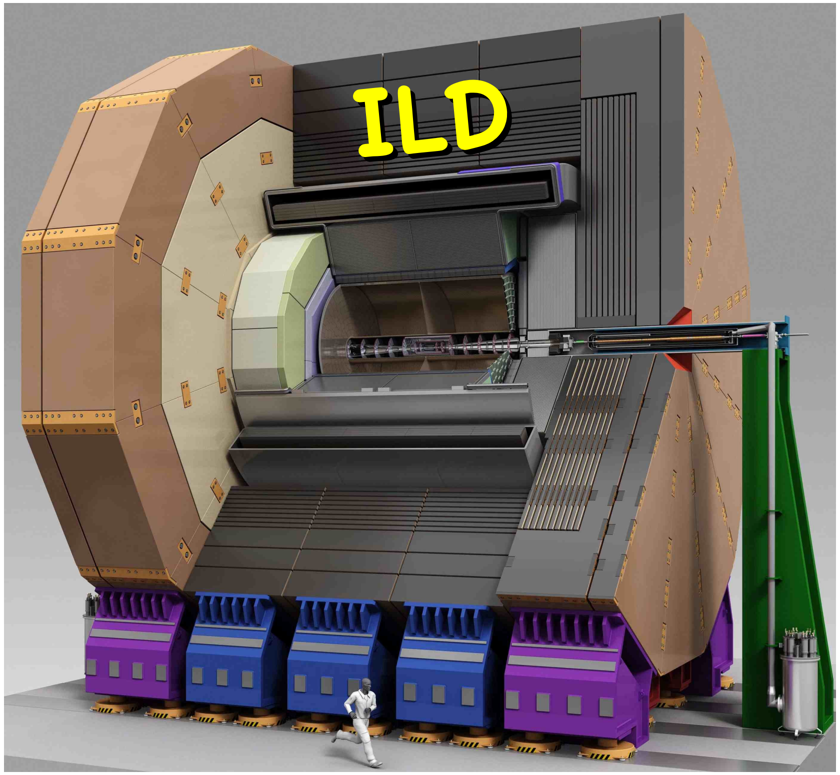
Status and plans for the iLCSoft framework

Frank Gaede, DESY
JSPS ILC Kickoff Meeting
Sendai, September 13, 2011

Outline

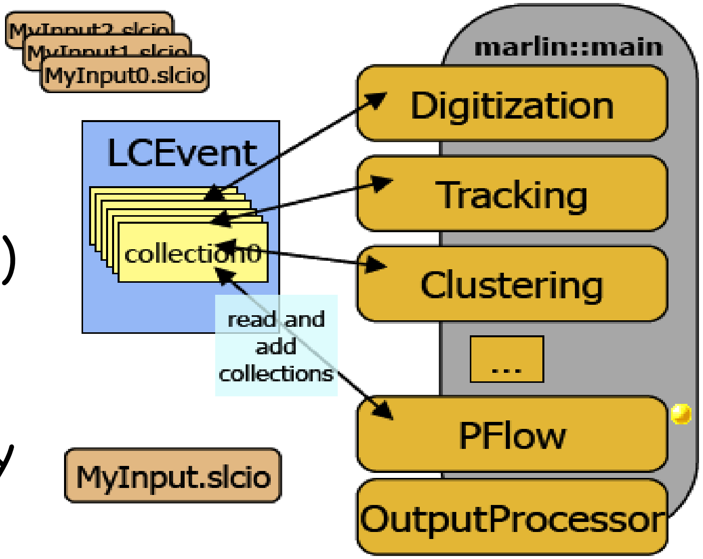
- overview of iLCSoft
  - the ILD software chain: core, sim, reco,...
- preparing for the DBD - recent developments
  - core tools
  - simulation (ILD_01)
  - reconstruction tools
- AIDA WP2
- Summary & Outlook

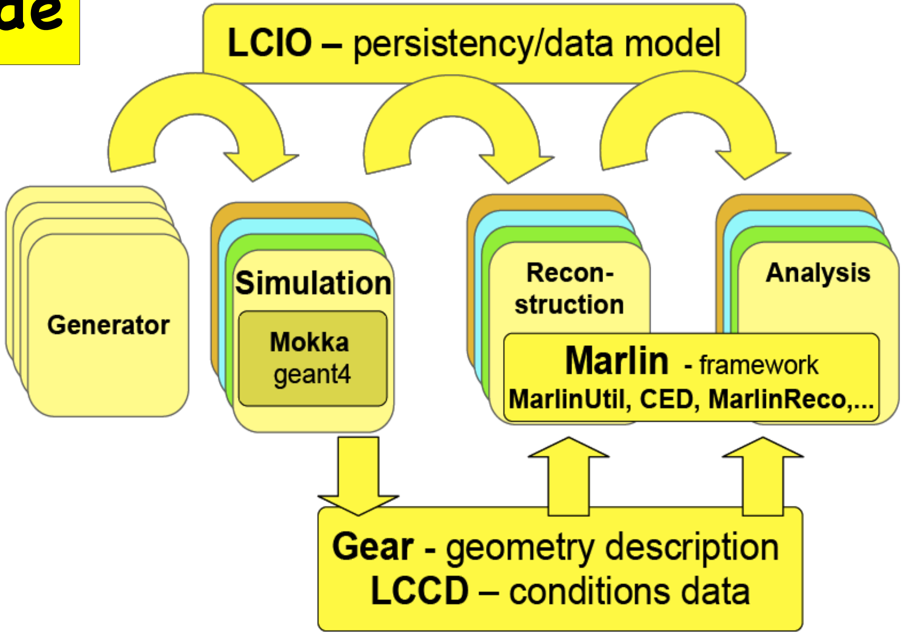


iLCSoft framework - Overview

http://ilcsoft.desy.de

- **Mokka** (LLR)
  - geant4 simulation application
- **LCIO** (DESY/SLAC)
  - international standard for persistency format / event data model
- **Marlin**
  - core application framework for reconstruction & data analysis
- **GEAR** geometry package f. reconstruction
- **LCCD**
  - conditions
  - data toolkit (DB)
- **CED**
  - 3d event display

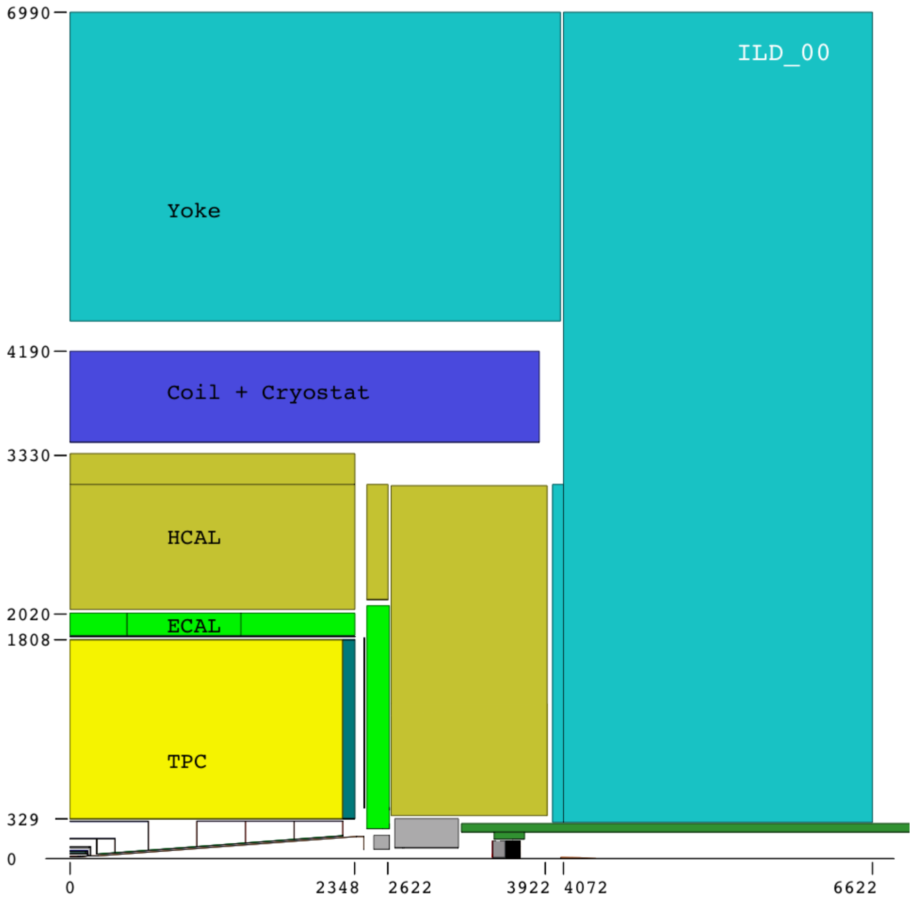


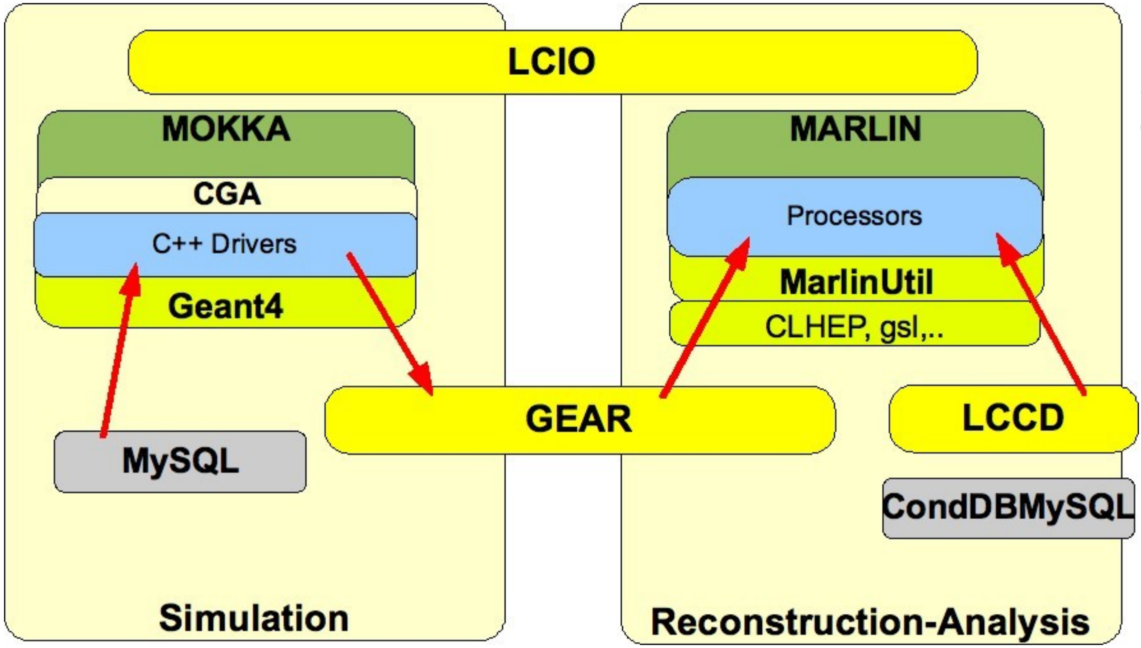


- complete framework used in Monte Carlo & 'real experiments':
  - **ILD detector concept** studies
  - **Calice** calo testbeam
  - **LC-TPC** testbeam
  - EUDET - **Pixel Telescope**
- **synergies between testbeam and global detector optimization**

Mokka Simulation ILD

- defined 'ILD simulation reference model' for LOI mass production (ILD_00)
- engineering level of detail for most subdetectors:
  - support structures
  - cracks
- CLIC uses modified detectot model ILD_CLIC for their CDR !
- goal: further improve realism of ILD model for **DBD (ILD_01)**





Mokka writes out GEAR xml files with complete geometry and material parameters that are need for reconstruction and analysis

Digitization & Reconstruction in Marlin

- VXD, SIT, FTD, SET, ETD
  - smearing of 3D space points according to resolutions from R&D groups
  - treatment of strips under development
- TPC hits
  - smearing of 3D space points - taking into account drift distance, polar and azimuthal angle of track
  - parameterization from TPC R&D groups
- ECal, HCal, LCal, Bcal, LHCal, Muon Calo hits
  - calibration (single particle resolution)
- Tracking
  - standalone tracking in Silicon detectors and TPC - **MarlinReco-FullLDCTracking**
  - Kalman filtering: code from LEP (f77)
  - new development started: **MarlinTrk** ...
- Particle Flow Algorithm
  - **MarlinPandora/PandoraPFANew**
- JetFinder
  - Durham jet finder (run for 2-6 jets)
- Flavour Tagging
  - **LCFIVertex** package: ZVTop, ZVRes + Neural Network Fl.Tag
  - **LCFIPlus** under development
- DST Maker
  - ReconstructedParticles, Jets, Tracks and Clusters (25k/evt)

afs reference installations

- provide reference installations in afs for usage from anywhere on ScientificLinux and compatible platforms:

/afs/desy.de/project/ilcsoft/sw/_OS_**/v01-11**

_OS_:	**i386_gcc34_sl4**	# i386 CPU, 32 bit, gcc3.4, SL4 and compatible
	i386_gcc41_sl5	# i386 CPU, 32 bit, gcc4.1, SL5 and compatible
	x86_64_gcc41_sl5	# i686 CPU, 64 bit, gcc4.1, SL5 and compatible

  - you can directly run from these installations, .eg:

```
source /afs/desy.de/project/ilcsoft/sw/x86_64_gcc41_sl5/v01-11/init.sh
Marlin mysteer.xml
```

  - you can link your own libraries against these
- plan to have other OSs in the future (as needed)

Note: older releases (<v01-09) at
/afs/desy.de/ilcsoft/

preparing iLCSoft for the DBD

- for the DBD we need:

- improved realism in the simulation
- **complete re-write of tracking code !**
  - old code unmaintainable and cannot easily cope with high backgrounds
- adaption of reconstruction algorithms (PFA, Flavor tag) to new technology options (SDHcal, FPCCD,...) [not in this talk]
- improved/adopted core tools

- currently finalizing next iLCSoft release: v01-12
  - LCIO v2
  - improved Mokka simulation
  - improved/extended Gear
  - new tracking package MarlinTrk
- plan to release next week - before LCWS2011

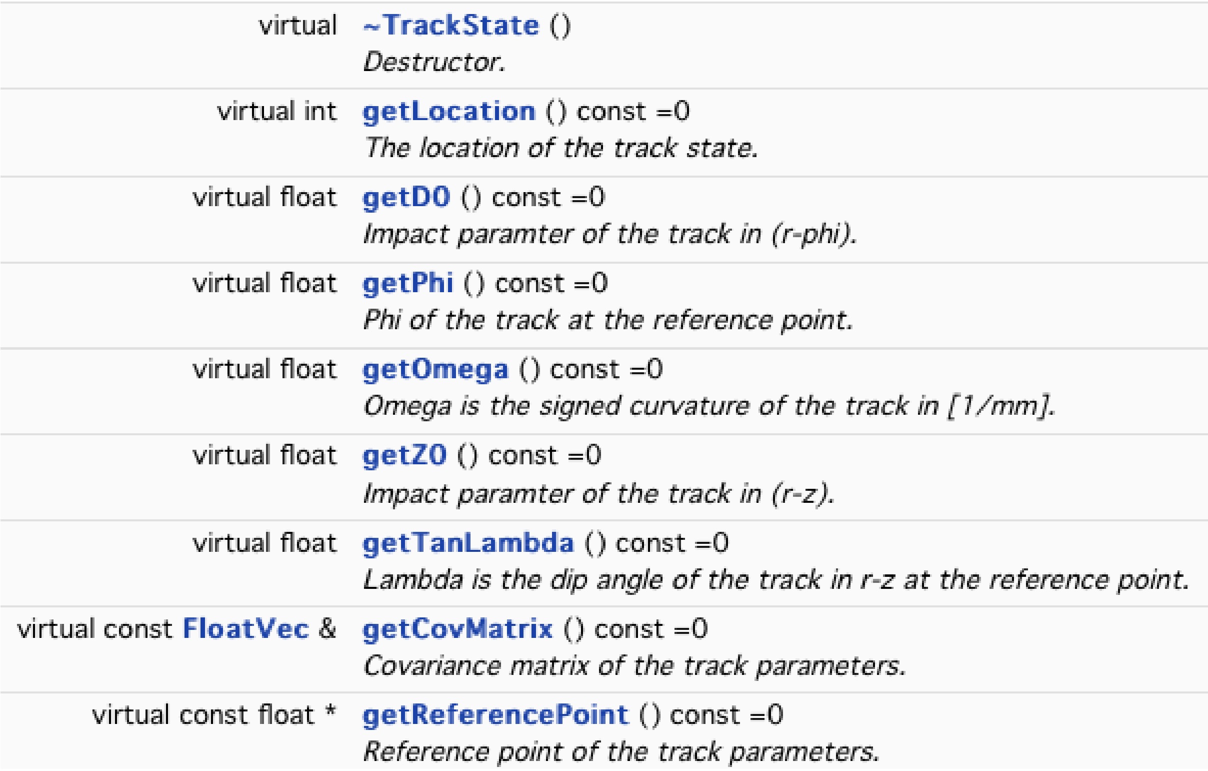
LCIO v2

- LCIO v2 planned for some time – goal: improve LCIO in backward compatible way
- main new features:
  - **direct access to events**
  - **simplified use of LCIO with ROOT**
  - **improved the event data model**
- due to lack of man power needed to de-scope from original plans - postponed:
  - splitting events over files
  - partial reading of events
- v02-00 to be released very soon !

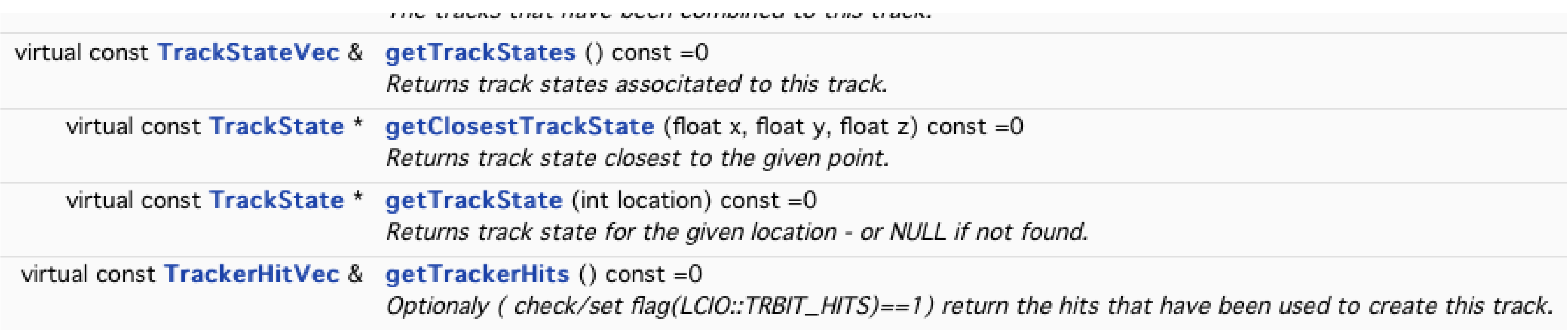
- EDM - API extensions
  - SimCalorimeterHit::getStepPosition(int i)
  - LCReader::getNumberOfEvents()
  - Cluster::getEnergyError()
  - float[3] MCParticle::getSpin()
  - int[2] MCParticle::getColorFlow()
  - int (Sim)TrackerHit::getCellID0()
  - int (Sim)TrackerHit::getCellID1()

LCIO v2 Track & Trackstates

- lcio Track now has **multiple TrackStates**
- will store four canonical TSs:
  - AtIP, AtFirstHit, AtLastHit, AtCalo
- TS returned either by
  - identifier
  - or closest to given point
- mostly backward compatible



virtual	**~TrackState** () *Destructor.*
virtual int	**getLocation** () const =0 *The location of the track state.*
virtual float	**getD0** () const =0 *Impact paramter of the track in (r-phi).*
virtual float	**getPhi** () const =0 *Phi of the track at the reference point.*
virtual float	**getOmega** () const =0 *Omega is the signed curvature of the track in [1/mm].*
virtual float	**getZ0** () const =0 *Impact paramter of the track in (r-z).*
virtual float	**getTanLambda** () const =0 *Lambda is the dip angle of the track in r-z at the reference point.*
virtual const **FloatVec** &	**getCovMatrix** () const =0 *Covariance matrix of the track parameters.*
virtual const float *	**getReferencePoint** () const =0 *Reference point of the track parameters.*



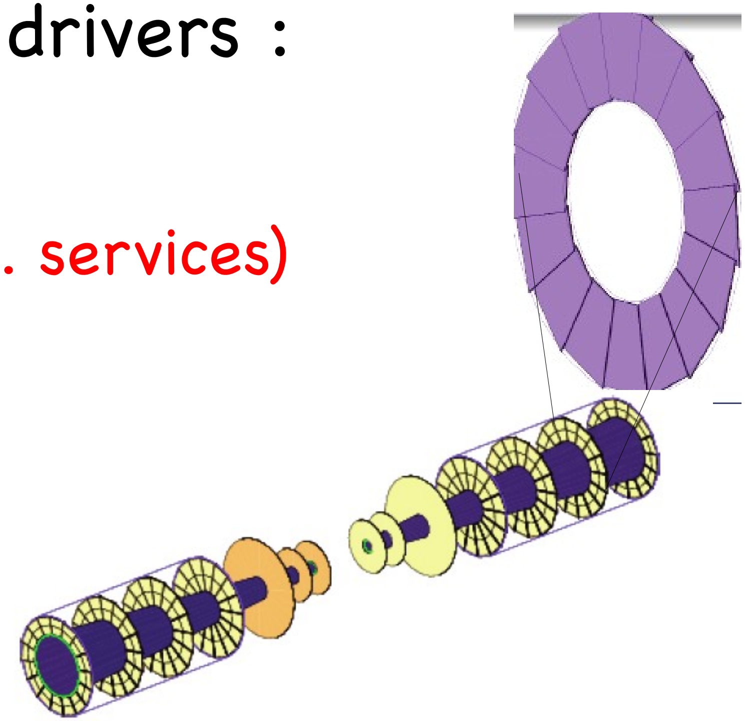
	The tracks that have been combined to this track.
virtual const **TrackStateVec** &	**getTrackStates** () const =0 *Returns track states associtated to this track.*
virtual const **TrackState** *	**getClosestTrackState** (float x, float y, float z) const =0 *Returns track state closest to the given point.*
virtual const **TrackState** *	**getTrackState** (int location) const =0 *Returns track state for the given location - or NULL if not found.*
virtual const **TrackerHitVec** &	**getTrackerHits** () const =0 *Optionaly (check/set flag(LCIO::TRBIT_HITS)==1) return the hits that have been used to create this track.*

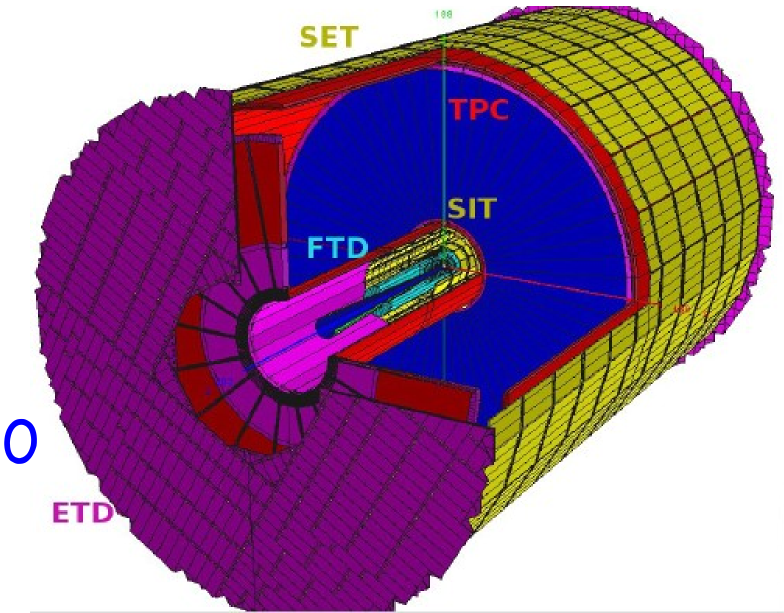
LCIOv2: 1d and 2d TrackerHits

- need new tracker hit classes to properly describe 1d and 2d measurements (pixels/TPC and **strips**)
- **TrackerHitPlanar**
  - x, y, z - 'space point'
  - u(theta, phi) , v(theta, phi) – measurement directions (spanning vectors in the plane)
  - du, dv - measurement errors
  - -> to be used for 1d and 2d (dv is strip length in 1d case)
- **TrackerHitCylindrical**
  - x, y, z - 'space point'
  - R, Xc, Yc – cylinder parameters (parallel to z)
  - dphi, dz - measurement errors
  - -> to be used for 1d and 2d
- these also implement the TrackerHit interface (x,y,z, cov) for backward compatibility and code reusability (eg in event display)

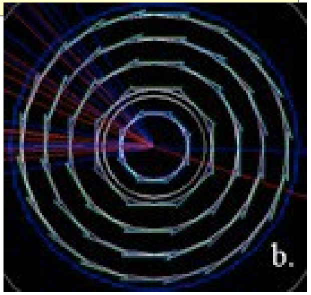
new Mokka models – ILD_01_pre02

- major rewrite of some sub detector drivers :
  - SIT, SET, ETD - FTD - Muon
  - increased level of detail and realism (incl. services)
- TPC
  - added endcap services (cooling)
- new ECal driver:
  - mixing of Scintillator and Si layers
- improved aHcal driver:
  - included electronics & services
- new SDHcal driver
  - w/ Videau geometry: ILD_01_SDH_pre00
- overall services for TPC, Ecal, Hcal

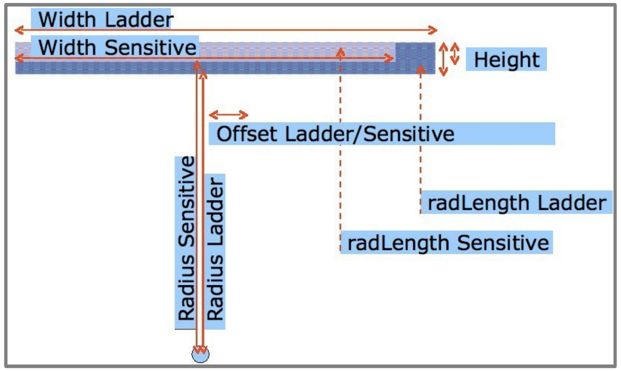




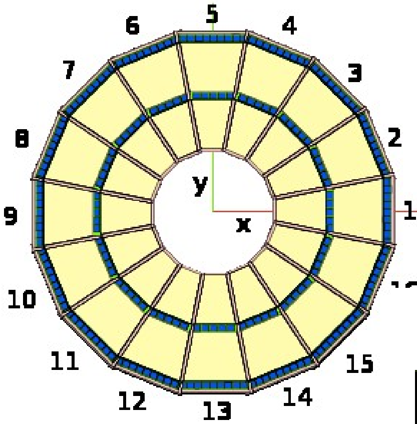
GEAR – new features

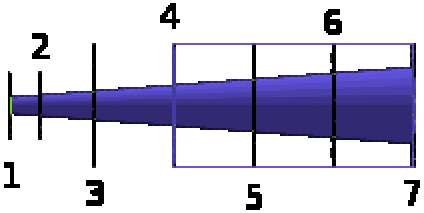


- added SIT and SET parameters – similar to VXD
  - describe (silicon) planar wafers along z-axis with phi-symmetry in placement and support material
  - renamed VXDParameters and VXDLayerLayout to ZPlanarParameters and ZPlanarLayerLayout
  - should be backward compatible through typedefs...



- added new FTDParameters and FTDLayerLayout (J.Duarte)
  - describe (silicon) disk detectors
  - made from petals
  - allow for tilting of petals (discouraged) or
  - staggering in z (preferred)

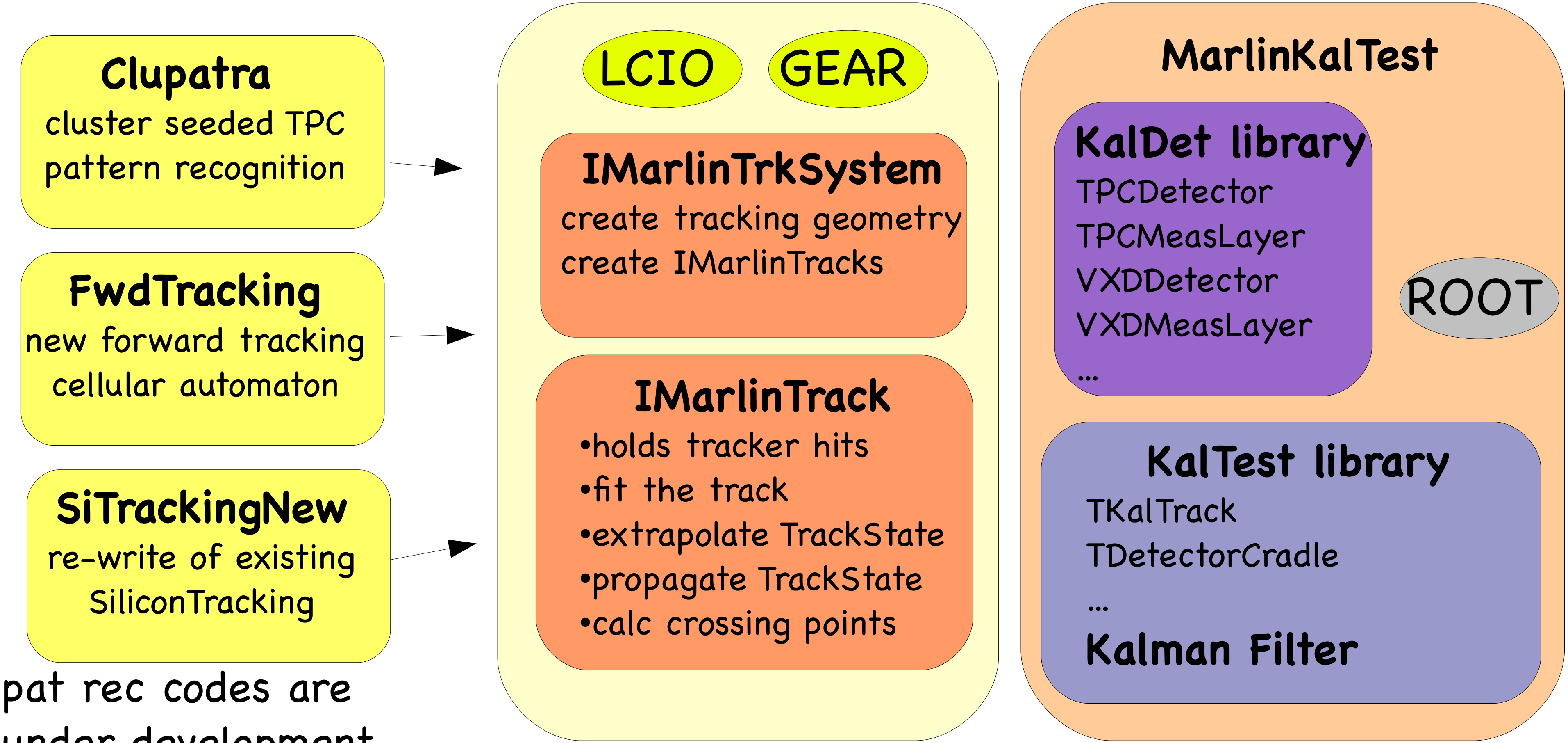




- **both are needed for the new ILD tracking code**

MarlinTrk

- need common framework for developing new tracking code (TPC, Silicon, Fwd)
- would like to have loose coupling between patrec and fitting
- defined abstract interface IMarlinTrk and implement using KalTest/KalDet
  - other fitters might follow (GenFit, ...)
- serves as tests case for writing a generic tracking package in AIDA



Clupatra
cluster seeded TPC
pattern recognition

FwdTracking
new forward tracking
cellular automaton

SiTrackingNew
re-write of existing
SiliconTracking

LCIO

GEAR

IMarlinTrkSystem
create tracking geometry
create IMarlinTracks

IMarlinTrack
•holds tracker hits
•fit the track
•extrapolate TrackState
•propagate TrackState
•calc crossing points

MarlinKalTest

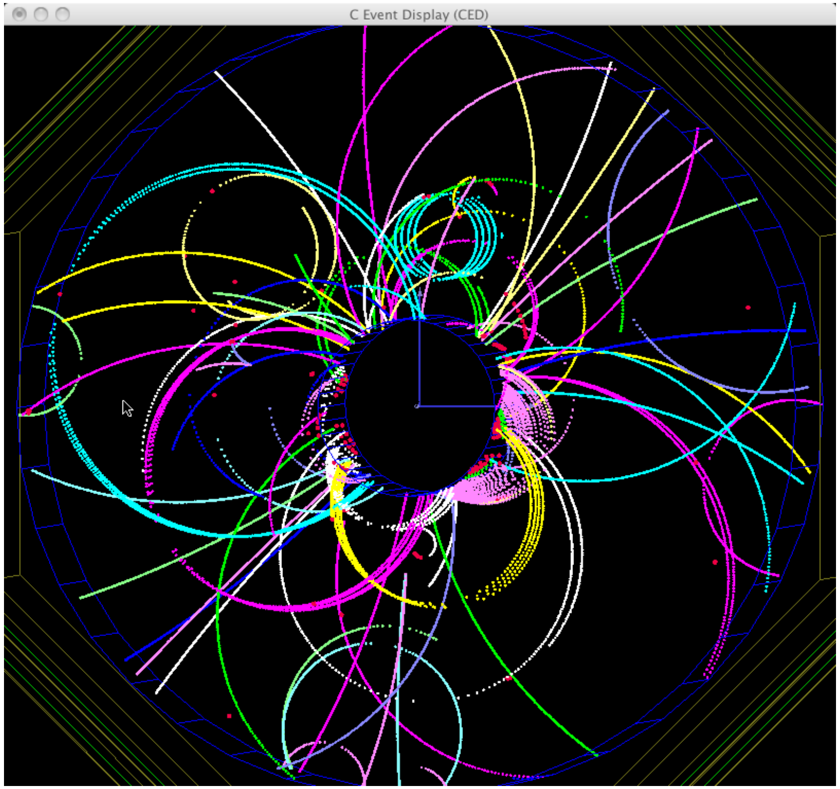
KalDet library
TPCDetector
TPCMeasLayer
VXDDetector
VXDMeasLayer
...

ROOT

KalTest library
TKalTrack
TDetectorCradle
...
Kalman Filter

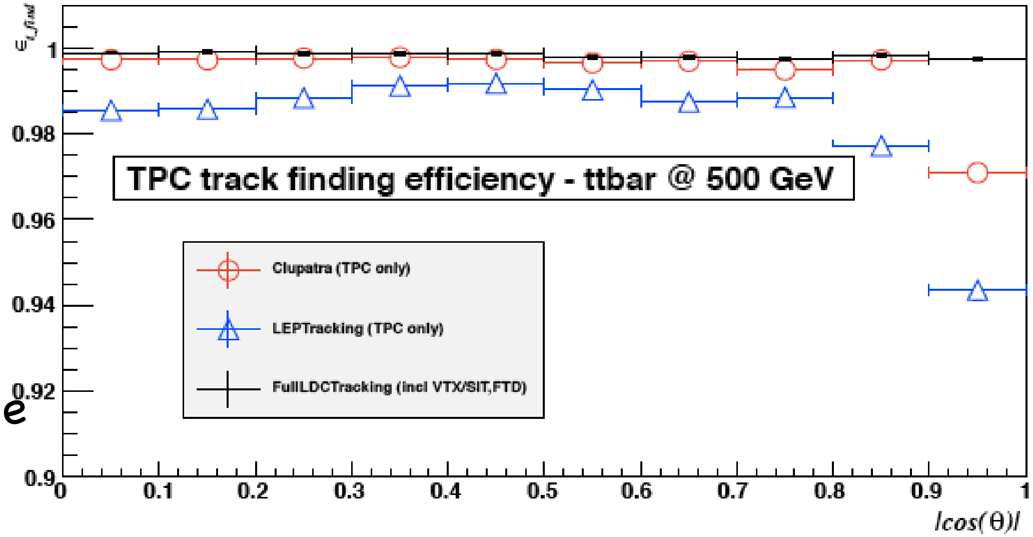
pat rec codes are
under development

Clupatra: topological TPC pat-rec



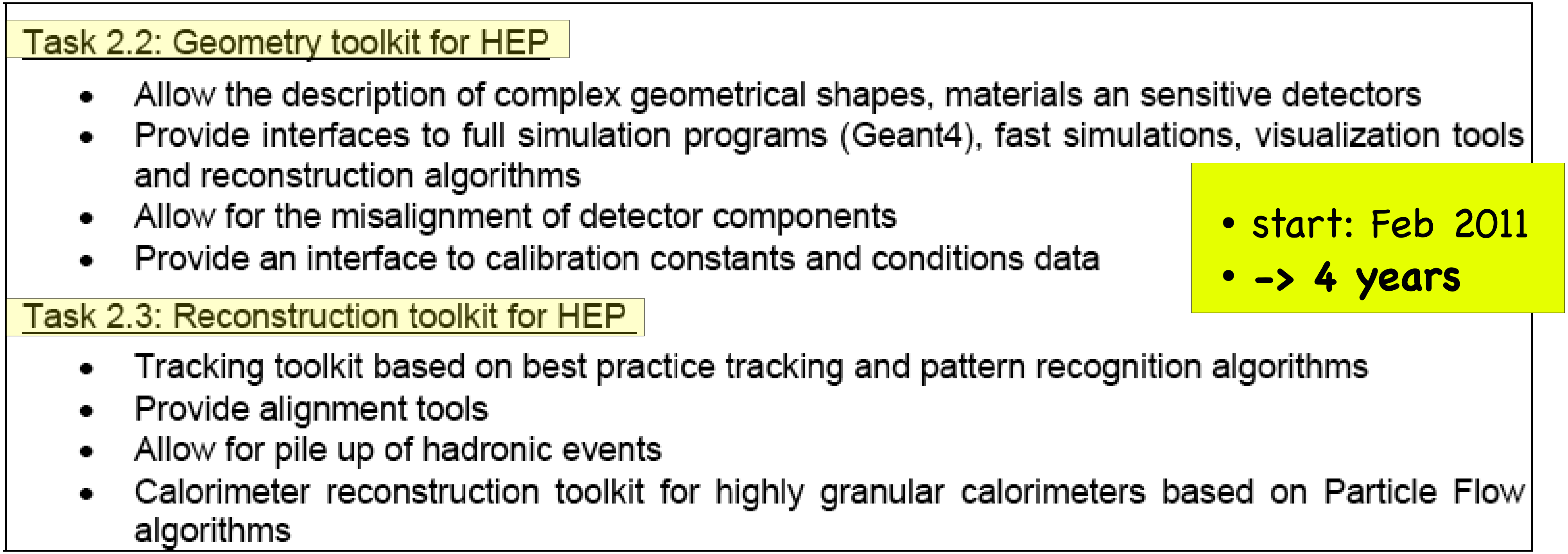
- use **NN-clustering in full TPC**
  - merge hits that have dist<3cm
- in merged clusters (duplicate pad rows) cluster in **pad row ranges** (15 rows) - ***outside inwards*** to find clean track stubs
- ***extend clean stubs with Kalman fitter***
  - pick up matching hits fwd & bwd if delta(chi2) < 35.
  - update track state
- ***force leftover clusters into one, two or three tracks*** (depending on pad row multiplicity)
- ***merge curler segments:***
  - delta(R), delta(xc,yc) and delta(tanL) < 10%

- track finding efficiency better than previous algorithm (based on LEP tracking code)
- NB: no fully reconstructed tracks yet -> might loose a bit due to quality cuts
- next steps:
  - re-write using new **MarlinTrk** package



AIDA WP2 – Common Software

develop core software tools that are useful for the HEP community at large and in particular for the next big planned projects: sLHC and Linear Collider (ILC/CLIC)



Task 2.2: Geometry toolkit for HEP

- Allow the description of complex geometrical shapes, materials an sensitive detectors
- Provide interfaces to full simulation programs (Geant4), fast simulations, visualization tools and reconstruction algorithms
- Allow for the misalignment of detector components
- Provide an interface to calibration constants and conditions data

Task 2.3: Reconstruction toolkit for HEP

- Tracking toolkit based on best practice tracking and pattern recognition algorithms
- Provide alignment tools
- Allow for pile up of hadronic events
- Calorimeter reconstruction toolkit for highly granular calorimeters based on Particle Flow algorithms

- start: Feb 2011
- **-> 4 years**

general strategy:

- **integrate as much as possible with existing software framework(s) and international activities outside of the AIDA project**
- collaborate with software activities in other AIDA work packages, e.g. the alignment of silicon sensors

Summary & Outlook

- very active development in iLCSoft framework as preparation for the DBD:
  - LCIOv2, Gear extensions, new MarlinTrk and PatRec code,...
  - greatly improved realism in Mokka simulation – in particular for Si-Tracking detectors
  - new technology options: FPCCD, SDHcal
  - not covered in this talk: calorimeter reconstruction/PFA, LCFIVertex,...
- **we are in good shape – but a lot of work still to be done until DBD !**

- plan to continue to provide iLCSoft as software tool beyond the DBD for LC detector R&D

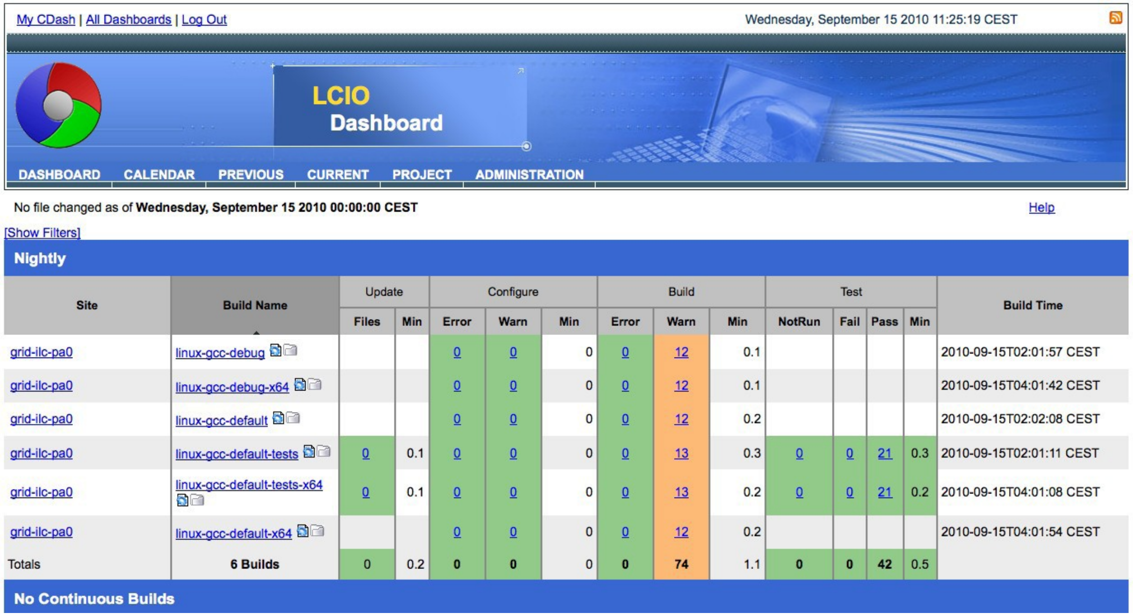
additional material

ILCTest

S.Aplin, J.Engels

- generic test system for iLCSoft:
  - unit tests
  - integration tests
  - physics test
- added some unit tests to most packages (run in Nightly Builds)
  - result browsable on dashboard
  - can be added to any iLCSoft package

```
// first line in your c++ source file
static ILCTest ilctest = ILCTest( "hello_world" );
...
ilctest.log( "hello world test" ); // a log message
...
If( x != 42 ){ ilctest.error("wrong answer!!") ; }
...
cout << ilctest.last_test_status() << endl; // prints "FAILED"
...
If( r > 3 ){ ilctest.fatal_error("this is a fatal error. program will quit now!") ; }
```



My CDash | All Dashboards | Log Out

Wednesday, September 15 2010 11:25:19 CEST

No file changed as of **Wednesday, September 15 2010 00:00:00 CEST**

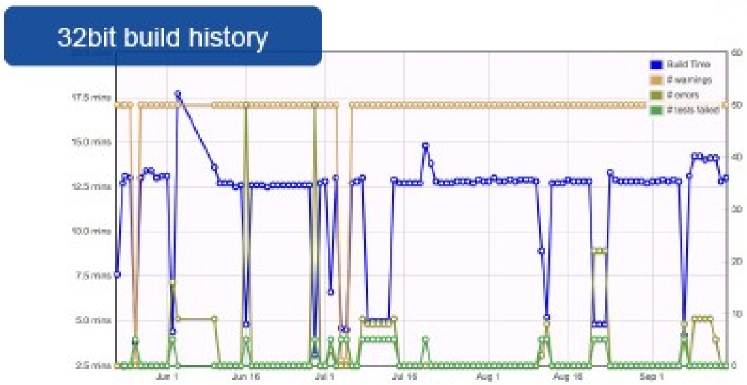
Help

[Show Filters]

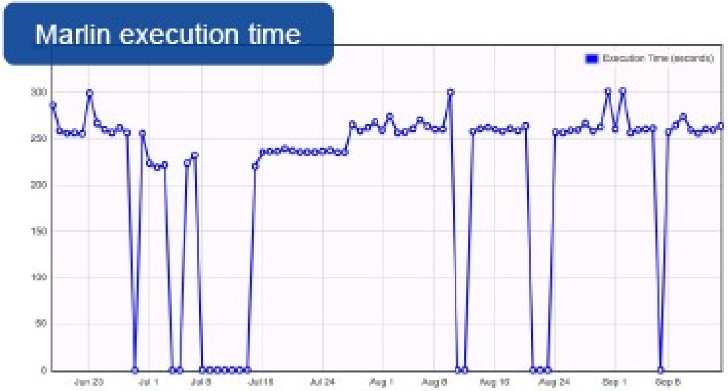
Nightly

Site	Build Name	Update		Configure			Build			Test				Build Time
		Files	Min	Error	Warn	Min	Error	Warn	Min	NotRun	Fail	Pass	Min	
grid-ilc-pa0	linux-gcc-debug			0	0	0	0	12	0.1					2010-09-15T02:01:57 CEST
grid-ilc-pa0	linux-gcc-debug-x64			0	0	0	0	12	0.1					2010-09-15T04:01:42 CEST
grid-ilc-pa0	linux-gcc-default			0	0	0	0	12	0.2					2010-09-15T02:02:08 CEST
grid-ilc-pa0	linux-gcc-default-tests	0	0.1	0	0	0	0	13	0.3	0	0	21	0.3	2010-09-15T02:01:11 CEST
grid-ilc-pa0	linux-gcc-default-tests-x64	0	0.1	0	0	0	0	13	0.2	0	0	21	0.2	2010-09-15T04:01:08 CEST
grid-ilc-pa0	linux-gcc-default-x64			0	0	0	0	12	0.2					2010-09-15T04:01:54 CEST
Totals	**6 Builds**	0	0.2	**0**	**0**	0	**0**	**74**	1.1	**0**	**0**	**42**	0.5	

No Continuous Builds



32bit build history



Marlin execution time

a ROOT dictionary for LCIO

- LCIO now comes with a ROOT dictionary for all LCIO classes (optional) - with this one can:
  (allready in v01-12-01)
  - use LCIO classes in ROOT macros
  - write simple ROOT trees, e.g. std::vector<MCParticleImpl*>
  - use TTreeDraw for quick interactive analysis of LCObjects:

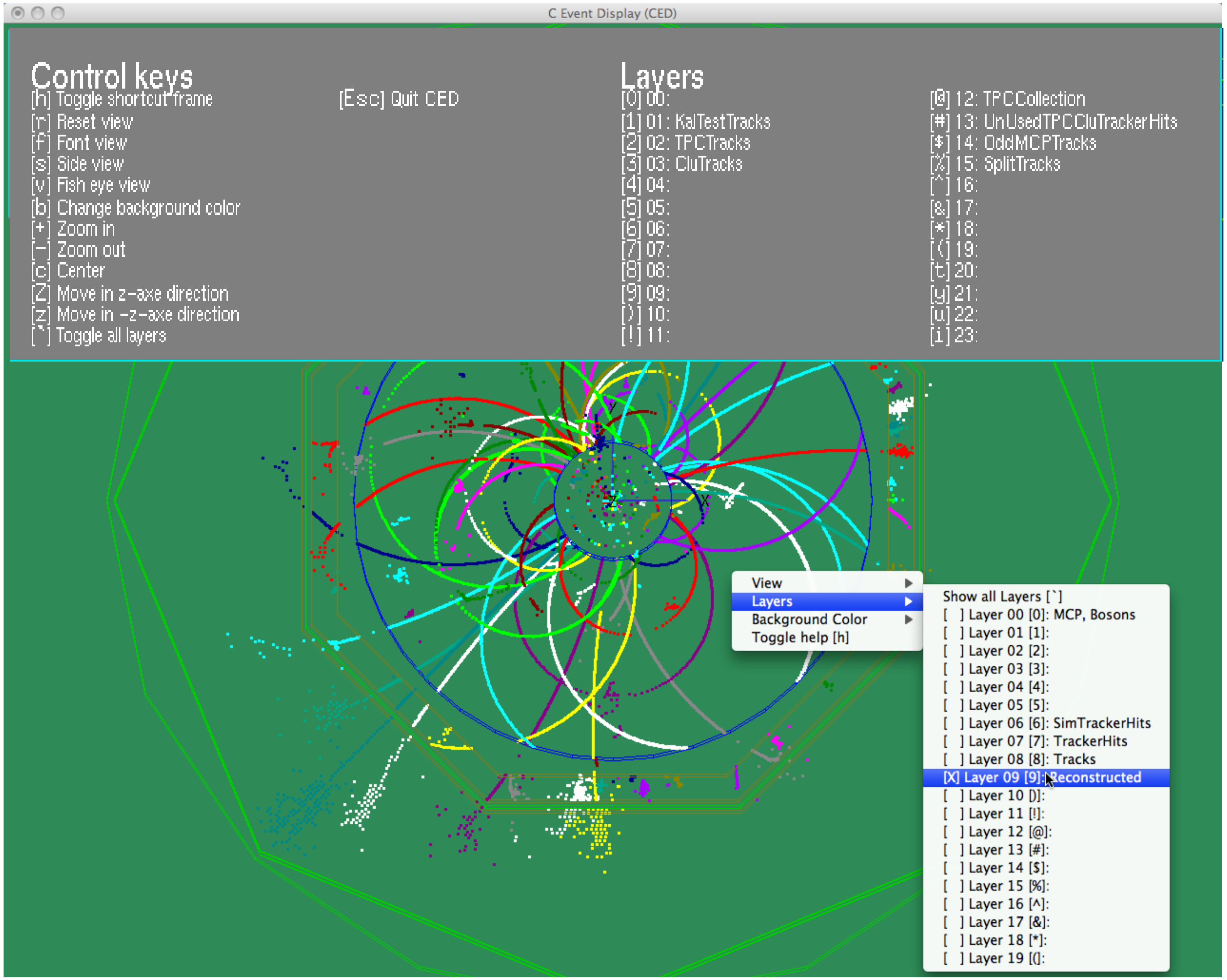
```
//---gamma conversions:
TCut isPhoton("MCParticlesSkimmed.getPDG()==22" ) ;
LCIO->Draw("MCParticlesSkimmed._endpoint[][0]:
           MCParticlesSkimmed._endpoint[][1]",isPhoton ) ;
```

  - write complete LCIO events in one ROOT branch
  - see: $LCIO/examples/cpp/rootDict/README for details & help

- -> we are interested in feedback from the users if this provides already the requested features

improved CED event display - I

- added help menu
  - toggled with 'h'
  - shows all keys
  - shows all 'collections'
- added mouse menu
  - toggle single visualization layers
  - choose bg colors
  - views
  - zoom
- commands to add layer description and picking also for user code



C Event Display (CED)

Control keys
[h] Toggle shortcut frame
[r] Reset view
[f] Font view
[s] Side view
[v] Fish eye view
[b] Change background color
[+] Zoom in
[-] Zoom out
[c] Center
[Z] Move in z-axe direction
[z] Move in -z-axe direction
[`] Toggle all layers

[Esc] Quit CED

Layers
[0] 00:
[1] 01: KalTestTracks
[2] 02: TPCTracks
[3] 03: CluTracks
[4] 04:
[5] 05:
[6] 06:
[7] 07:
[8] 08:
[9] 09:
[)] 10:
[!] 11:

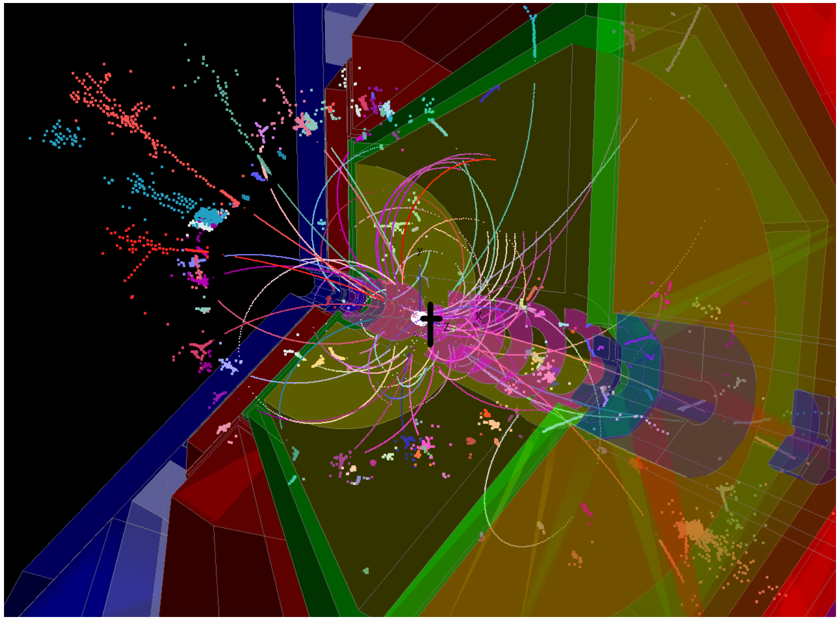
[@] 12: TPCCollection
[#] 13: UnUsedTPCCluTrackerHits
[$] 14: OddMCPTracks
[%] 15: SplitTracks
[^] 16:
[&] 17:
[*] 18:
[(] 19:
[t] 20:
[y] 21:
[u] 22:
[i] 23:

- new python script to start CED & CEDViewer in on go: ced2go.py

(H.Hoelbe)

improved CED event display - II

- many new features in CED (H.Hoelbe) (and CEDViewer, MarlinCED) :
  - added a New View with
    - 3d transparent surfaces
    - cut open detector
  - save display settings
  - turn on/off detector components
  - new projections:
    - r-phi ("F")
    - r-z   ("S")
  - toggle view of axes
  - ...
  - detailed User Manual



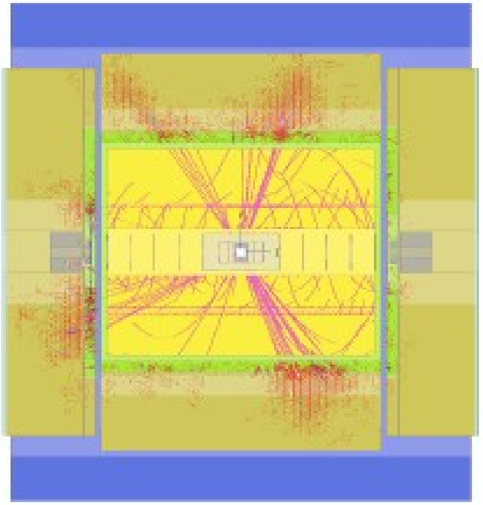


Figure 2: *Side view projection*

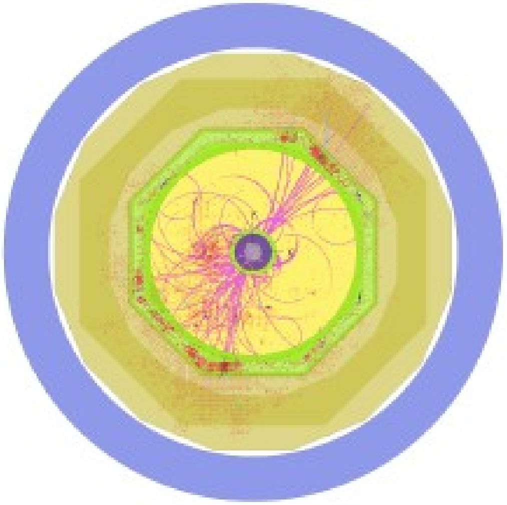
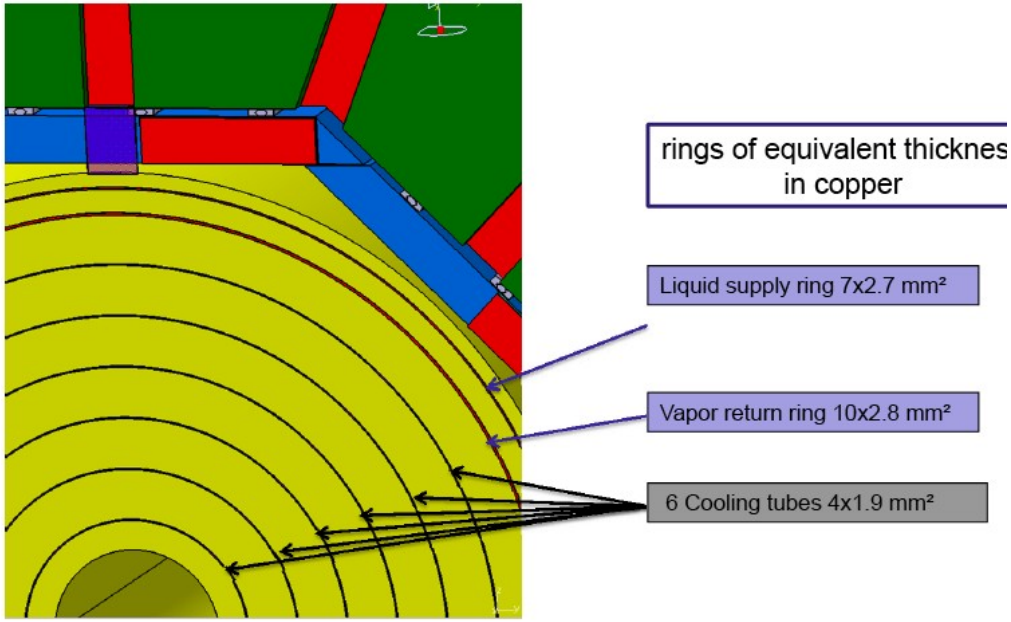


Figure 3: *Front view projection*

new Mokka release - towards ILD_01

- added cabling and services for TPC, ECal & Hcal (C.Clerc, G.Musat)
- still missing: inner detector services (to be defined by R&D groups)

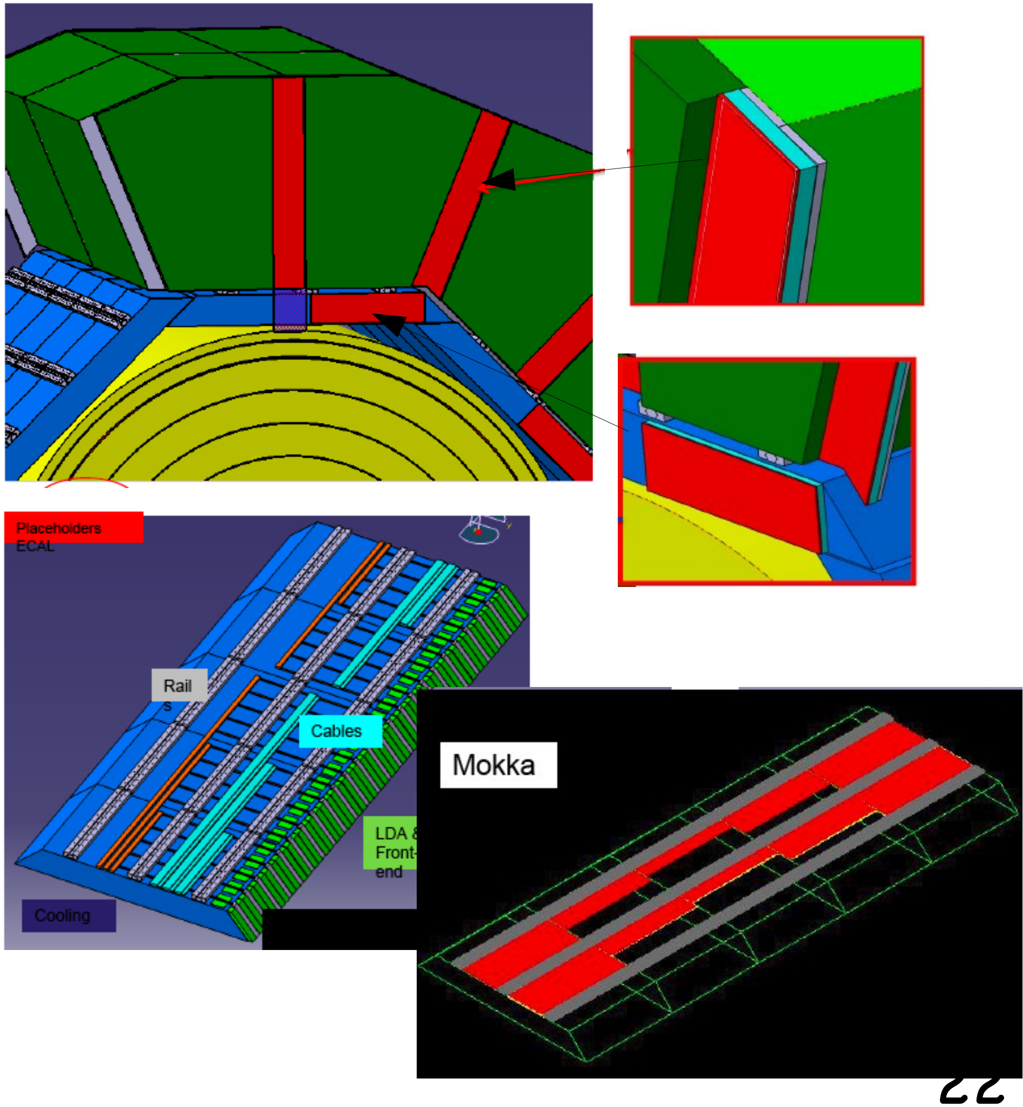


rings of equivalent thicknes in copper

Liquid supply ring 7x2.7 mm²

Vapor return ring 10x2.8 mm²

6 Cooling tubes 4x1.9 mm²



big step forward in increasing realism of ILD detector simulation !