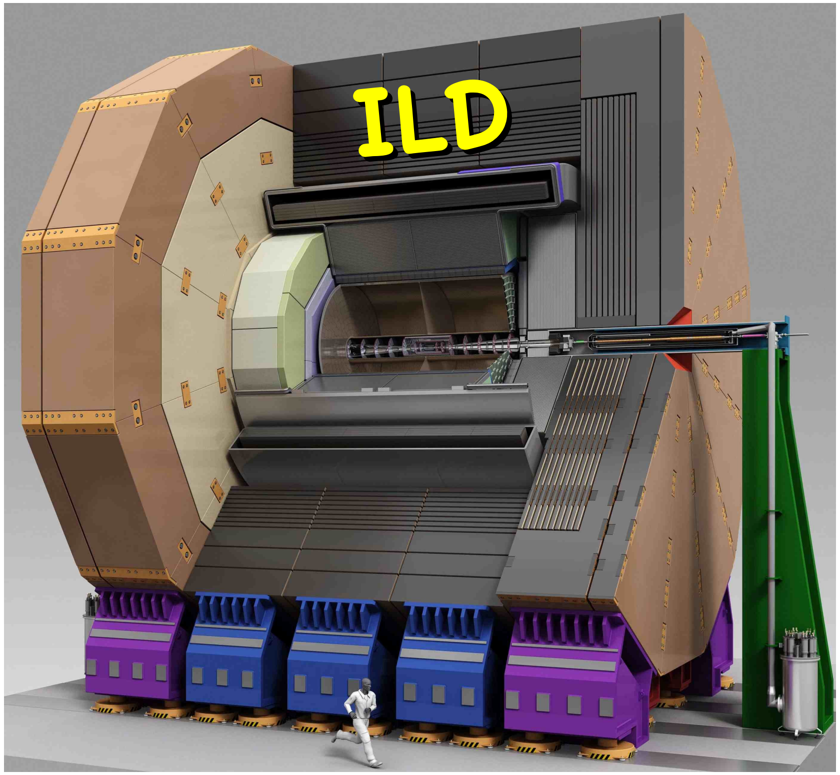
iLCSoft – Status and Plans

Frank Gaede, DESY
KILC 2012
Daegu, Korea, Apr 23-27, 2012

Outline

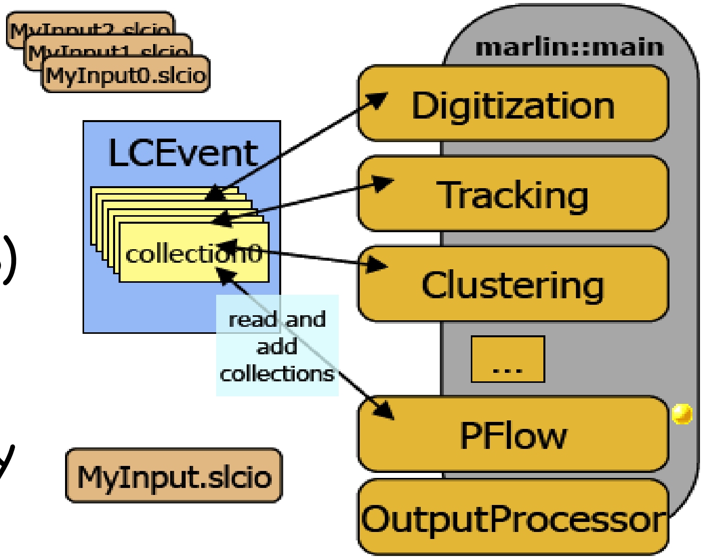


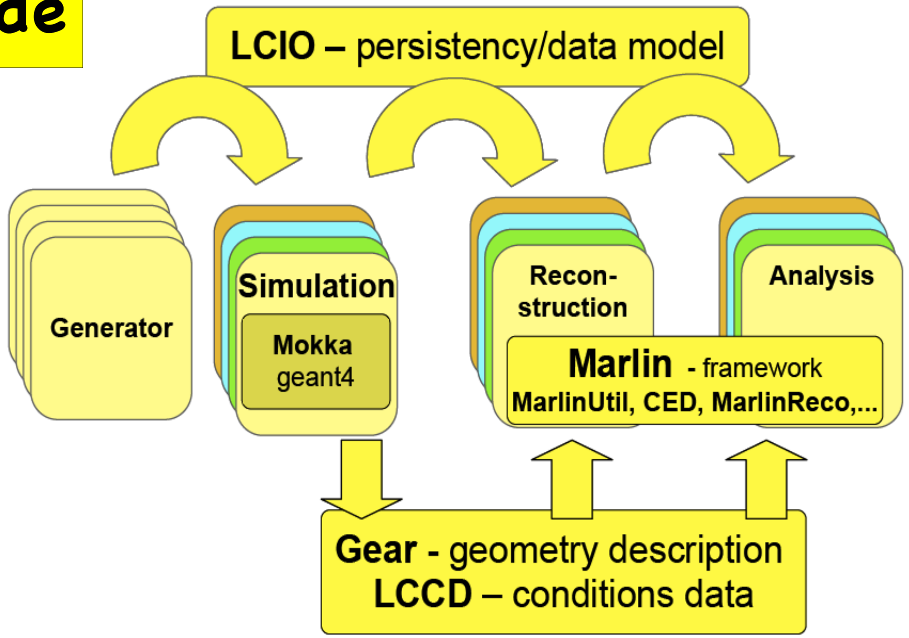
- brief overview of iLCSoft
- recent developments
  - core tools
  - Mokka simulation
  - new tracking
  - release v01-13-05
- report from 'Linear Collider Software Meeting 2012'
- Summary & Outlook

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

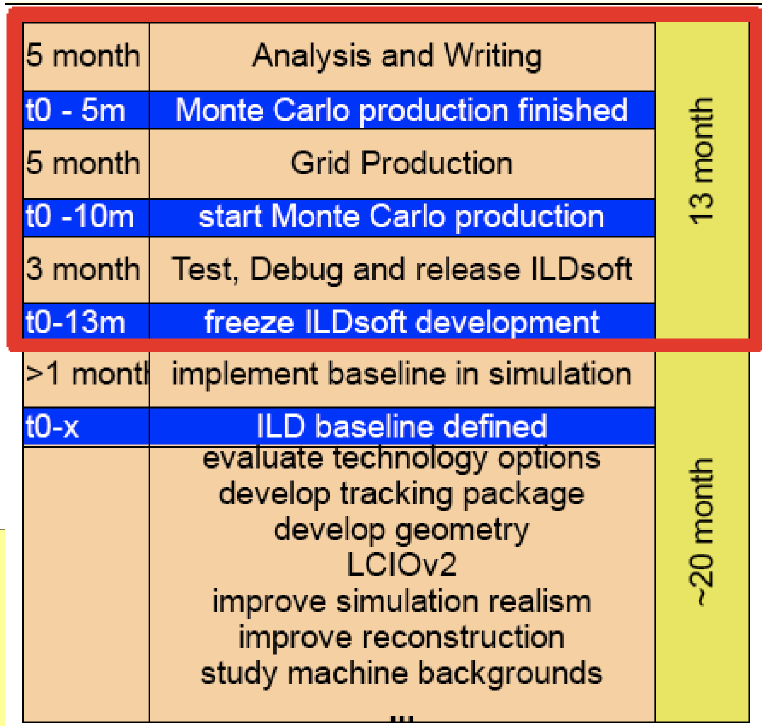




- complete framework used in Monte Carlo & 'real experiments':
  - **ILD detector concept** studies
  - **Calice** calo testbeam
  - **LC-TPC** testbeam
  - EUDET - **Pixel Telescope**
- **synergies between testbeam and global detector optimization**

timeline for iLCSoft development

- timeline for iLCSoft development in last 2-3 years was mainly driven by the requirements for the ILD-DBD



5 month	Analysis and Writing	13 month
t0 - 5m	Monte Carlo production finished	
5 month	Grid Production	
t0 -10m	start Monte Carlo production	
3 month	Test, Debug and release ILDsoft	
t0-13m	freeze ILDsoft development	
>1 mont	implement baseline in simulation	~20 month
t0-x	ILD baseline defined	
	evaluate technology options develop tracking package develop geometry LCIOv2 improve simulation realism improve reconstruction study machine backgrounds ...	

- this talk: main activities:
  - **improved/adopted core tools**
    - LCIOv2, GEAR, CED,...
  - **improved realism of the simulation**
    - include gaps, imperfection and services
  - **complete re-write of tracking code !**
    - old code unmaintainable and cannot easily cope with high bg
  - adaption of reconstruction algorithms (PFA, Flavor tag) to new technology options (SDHcal, FPCCD,...) [not in this talk]

LCIO v2

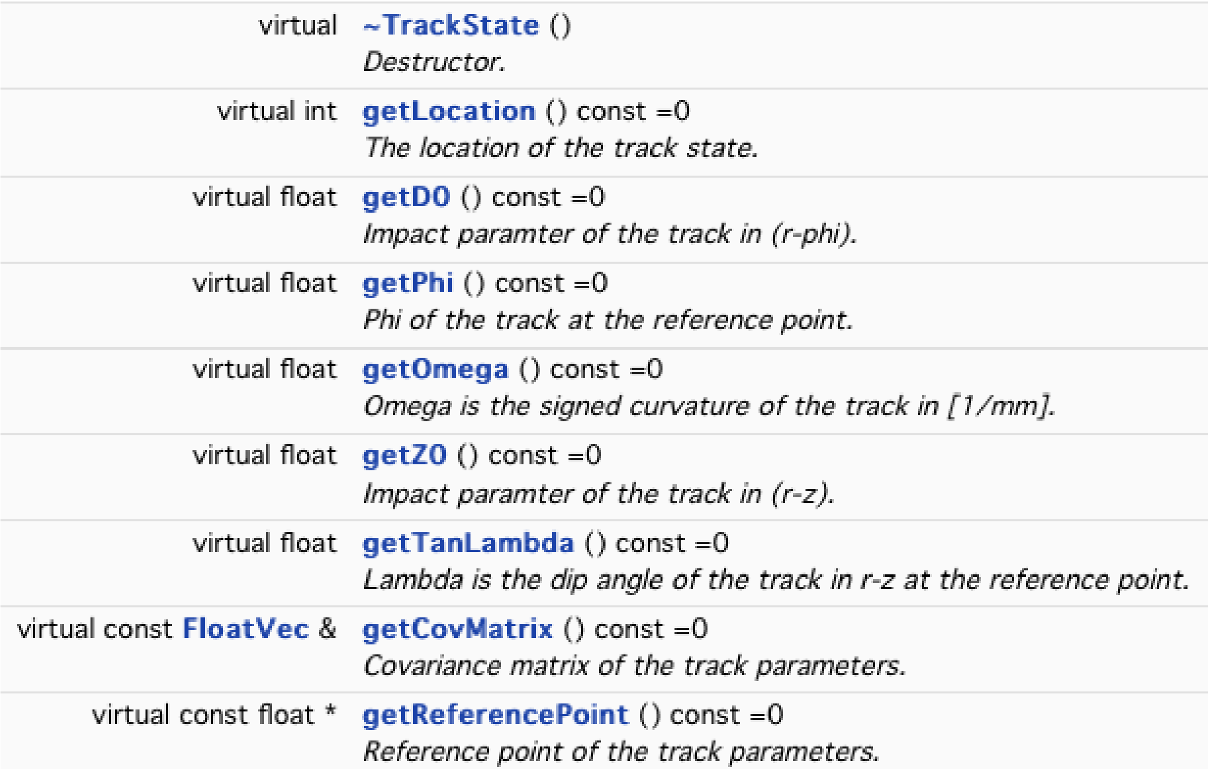
- LCIO v2 had been planned for some time - goal: improve LCIO in backward compatible way
- main new features:
  - **direct access to events**
  - **simplified use of LCIO with ROOT**
  - **improved the event data model**
- due to lack of man power needed to de-scope from original plans - postponed:
  - splitting events over files
  - partial reading of events
- v02-00 was released Sep 2011

  current: v02-01-01

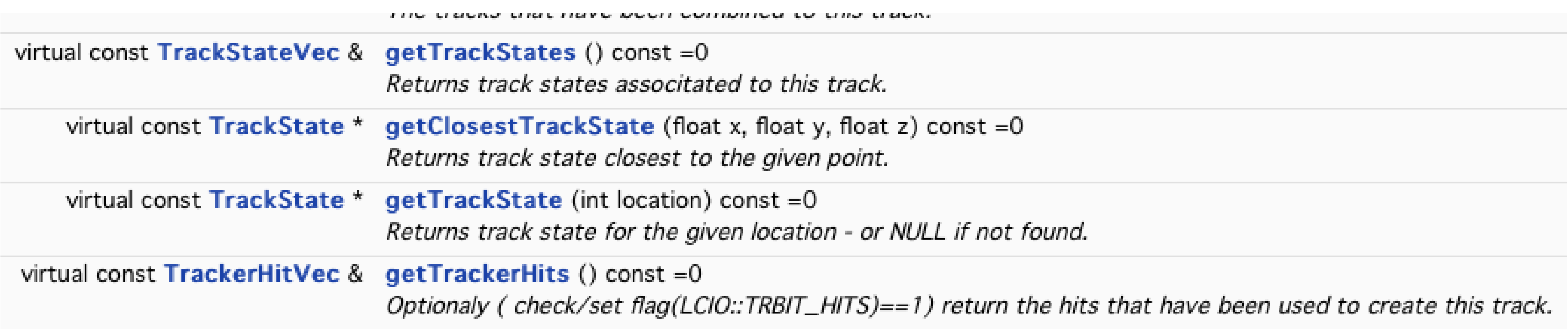
- EDM - API extensions
  - SimCalorimeterHit::getStepPosition(int i)
  - LCReader::getNumberOfEvents()
  - Cluster::getEnergyError()
  - float[3] MCParticle::getSpin()
  - int[2] MCParticle::getColorFlow()
  - int (Sim)TrackerHit::getCellID0()
  - int (Sim)TrackerHit::getCellID1()

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.*



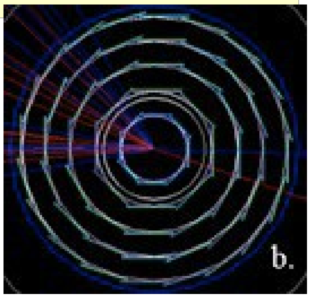
The tracks that have been combined to this track.

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

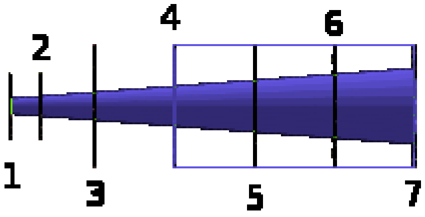
- need new tracker hit classes to properly describe 1d and 2d measurements (pixels/TPC and **strips**)
- **TrackerHitPlanar**
  - x, y, z - 'space point'
  - u(theta, phi) , v(theta, phi) – measurement directions (spanning vectors in the plane)
  - du, dv - measurement errors
  - -> to be used for 1d 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)

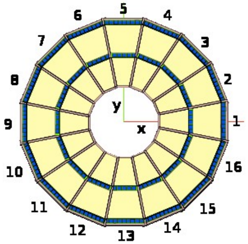
GEAR - new developments



- added SIT and SET parameters - similar to VXD
  - describe (silicon) planar wafers along z-axis with phi-symmetry in placement and support material
- added new FTDParameters (J.Duarte)
  - describe (silicon) disk detectors made from petals

  both needed to describe the now much more realistic and detailed Si-tracking simulation

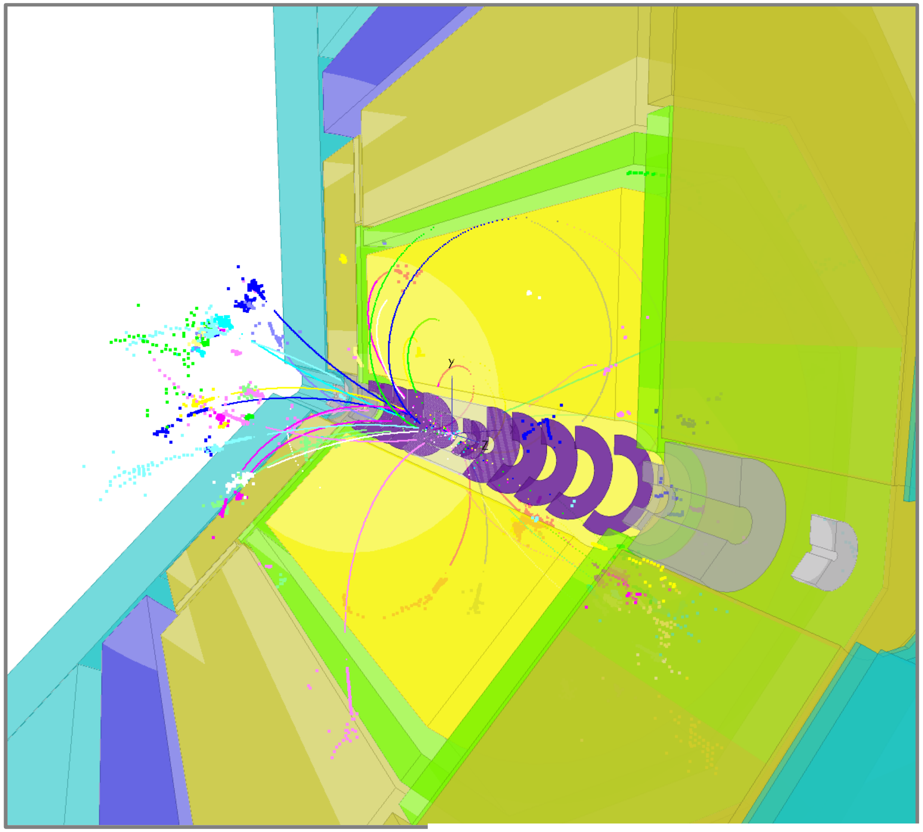


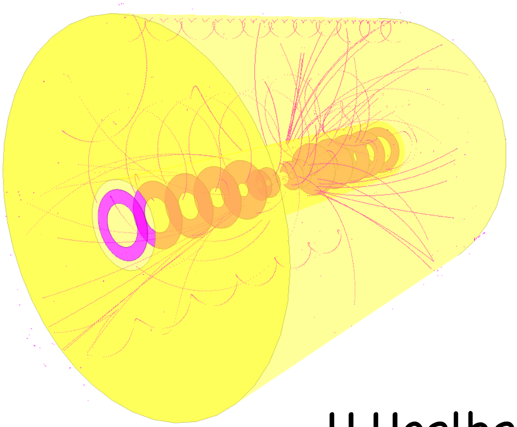


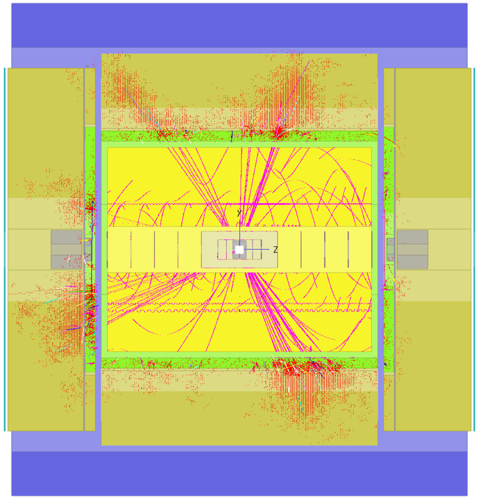
- added SimpleMaterial section in Gear parameters
  - SimpleMaterial(Name, A, Z, density, X0, Lambda)
  - need to add code to Mokka drivers to write these materials
- added MeasurementSurfaceStore (S.Aplin)
  - describes bounded measurement surfaces' coordinate systems
  - local to global transformation of tracker hit measurements

new features in CED event display

- many new features in CED, 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



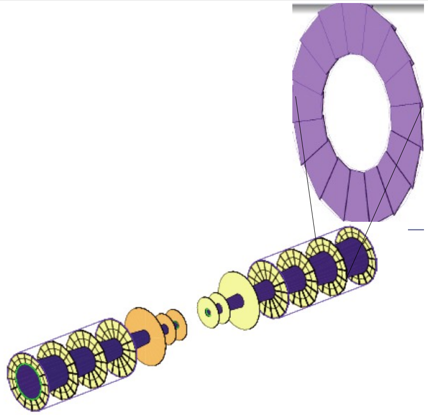


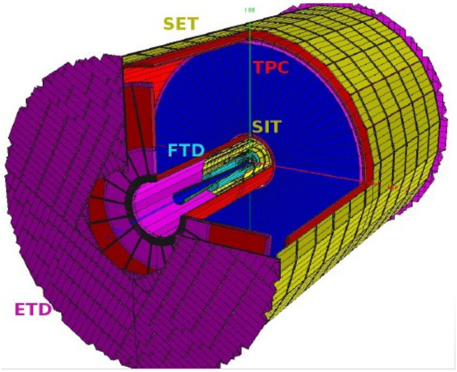


H.Hoelbe

recent developments in Mokka

- major rewrite of some sub detector drivers :
  - SIT, SET, ETD - FTD - Muon
  - increased level of detail and realism (incl. services)
- made existing drivers more realistic:
  - TPC, AHCal, Ecal, FCal,...
- new drivers (technology options):
  - SDHCal, SciEcal
- added overall services and cables
- new models for DBD:



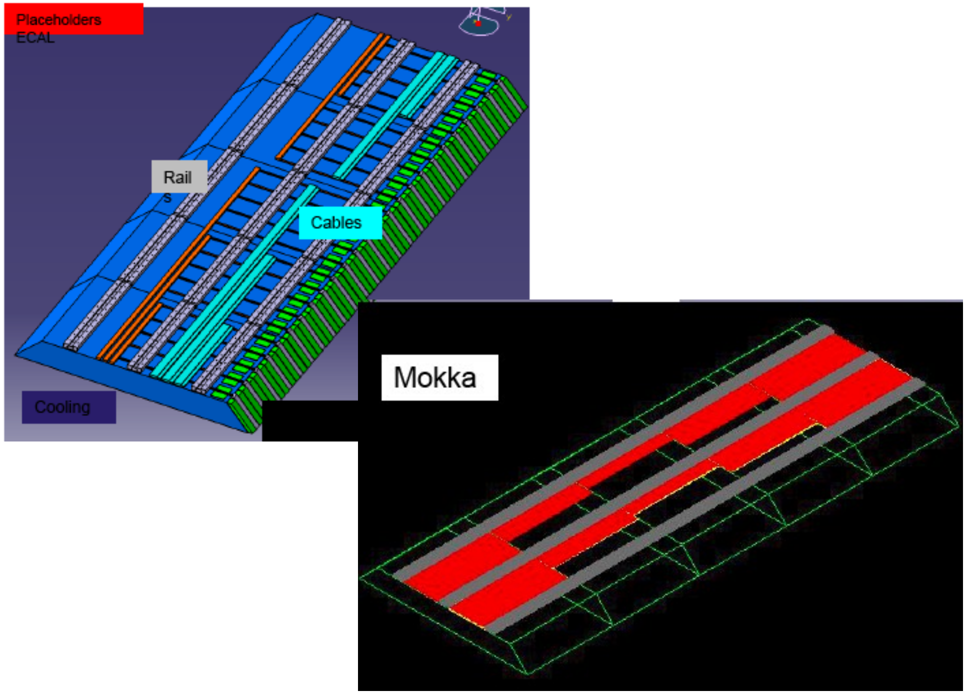


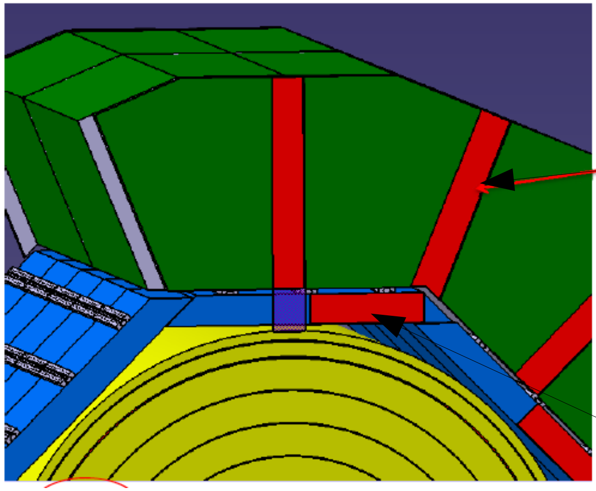
ILD_O1_v01 "ILD simulation reference Model for DBD using Analog HCal"
ILD_O2_v01 "ILD simulation reference Model for DBD using SD HCal"
ILD_O3_v01 "ILD simulation reference Model for DBD using SciW Ecal and Analog HCal"

- first part ILD_OX - read "ILD - Option X", refers to the choice of sub-detector technology options of the model
- second part _vXX refers to the software release version that describes this option for ILD

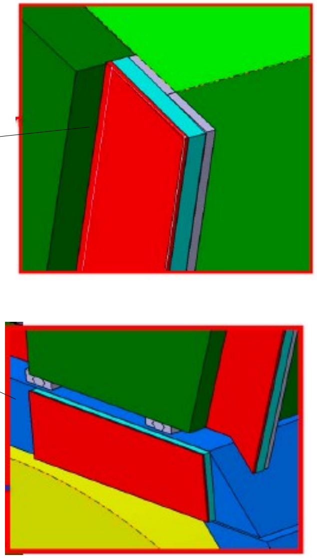
increased realism in ILD_OX models

- added cabling and services for TPC, ECal & Hcal (C.Clerc, G.Musat)
- including inner detector services as defined by R&D groups



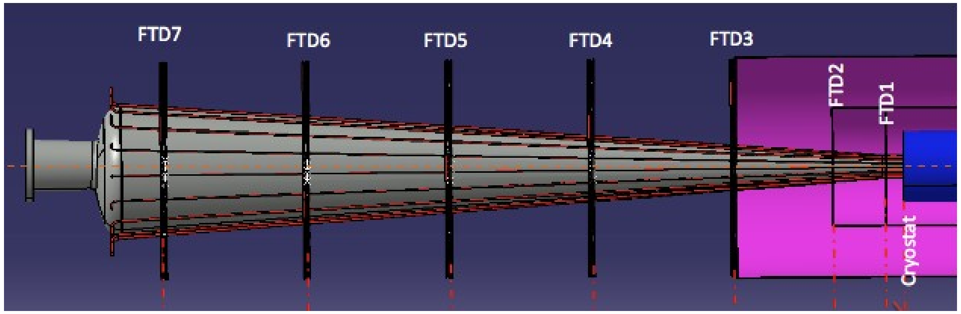


cooling pipes



electronics

big step forward in increasing realism of ILD detector simulation !



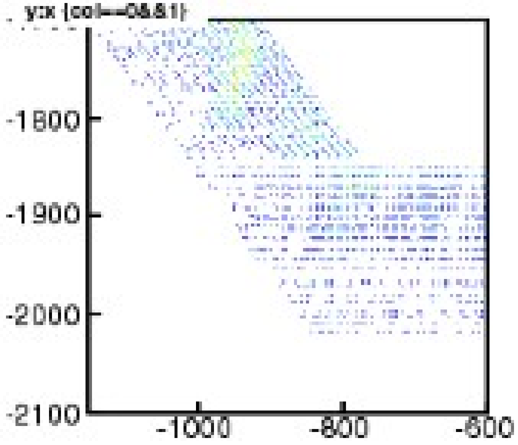
power supply cables

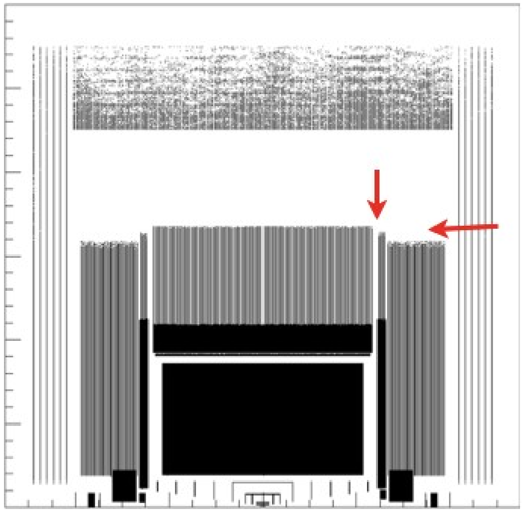
validation of Mokka ILD model(s)

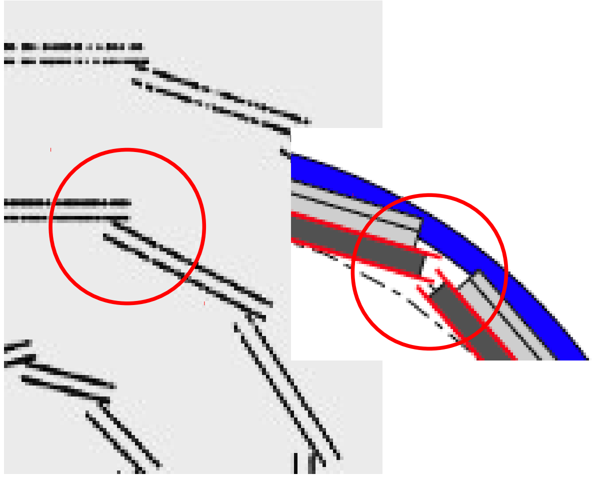
- started validation process with volunteers nominated by the R&D groups
  - checking: overlaps, consistency w/ engineering model, hit production,..

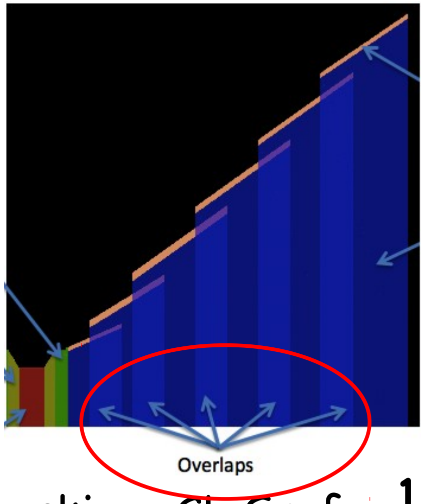
detector	person	status
VXD	G.Voutsinas	ongoing
SIT/SET	K.Androsov	to be done
FTD	J.Duarte	to be done
TPC	S.Aplin	done
ECal	D.Jeans	done
AHCal	Sh.Lu	done
SDHcal	G.Grenier	done
FCal	A.Rosca, B.Pawlik	ongoing
Muon	A.Saveliev	ongoing

start MC production, once all sub detectors are 'approved'









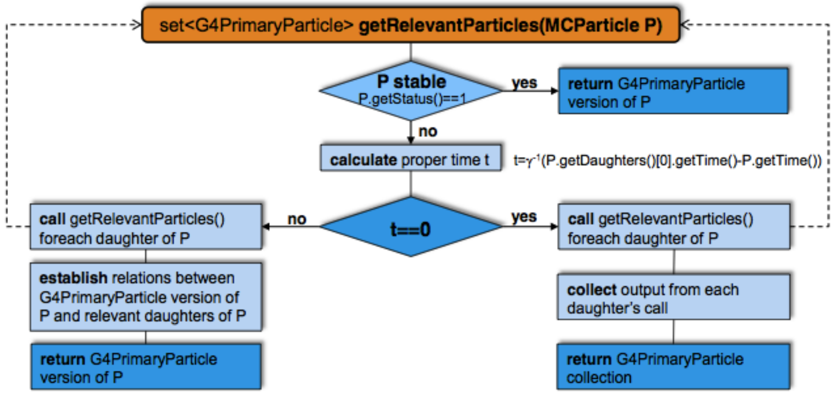
overall overlap checking: Ch.Grefe

new treatment of stdhep particles

- changed treatment of heavy particle decays in Mokka:
  - use decay and lifetime from generator file (was: lifetime from geant4)
  - added extra particle decay table for 'exotic' particles (SUSY)

- changed logic for selecting particle presented to geant4

1. Find all MC particles without parents → initial particles
2. Run foreach initial particle P getRelevantParticles(P)



set<G4PrimaryParticle> **getRelevantParticles(MCParticle P)**

P stable P.getStatus()==1

yes: **return** G4PrimaryParticle version of P

no: **calculate** proper time t

$t = \gamma^{-1}$(P.getDaughters()[0].getTime()-P.getTime())

t==0

no: **call** getRelevantParticles() foreach daughter of P

establish relations between G4PrimaryParticle version of P and relevant daughters of P

return G4PrimaryParticle version of P

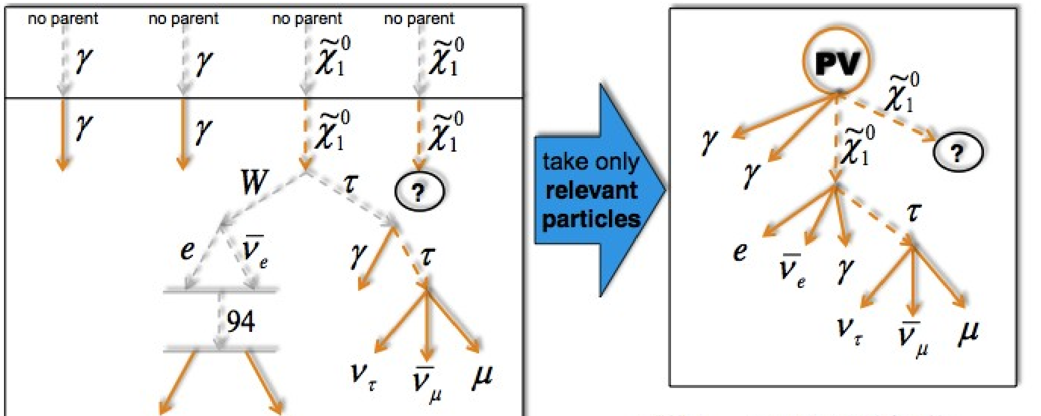
yes: **call** getRelevantParticles() foreach daughter of P

collect output from each daughter's call

return G4PrimaryParticle collection

3. Add G4PrimaryParticles returned by initial call of getRelevantParticles() to G4PrimaryVertex

→ **take MC generator life time information instead of defining "special cases"**

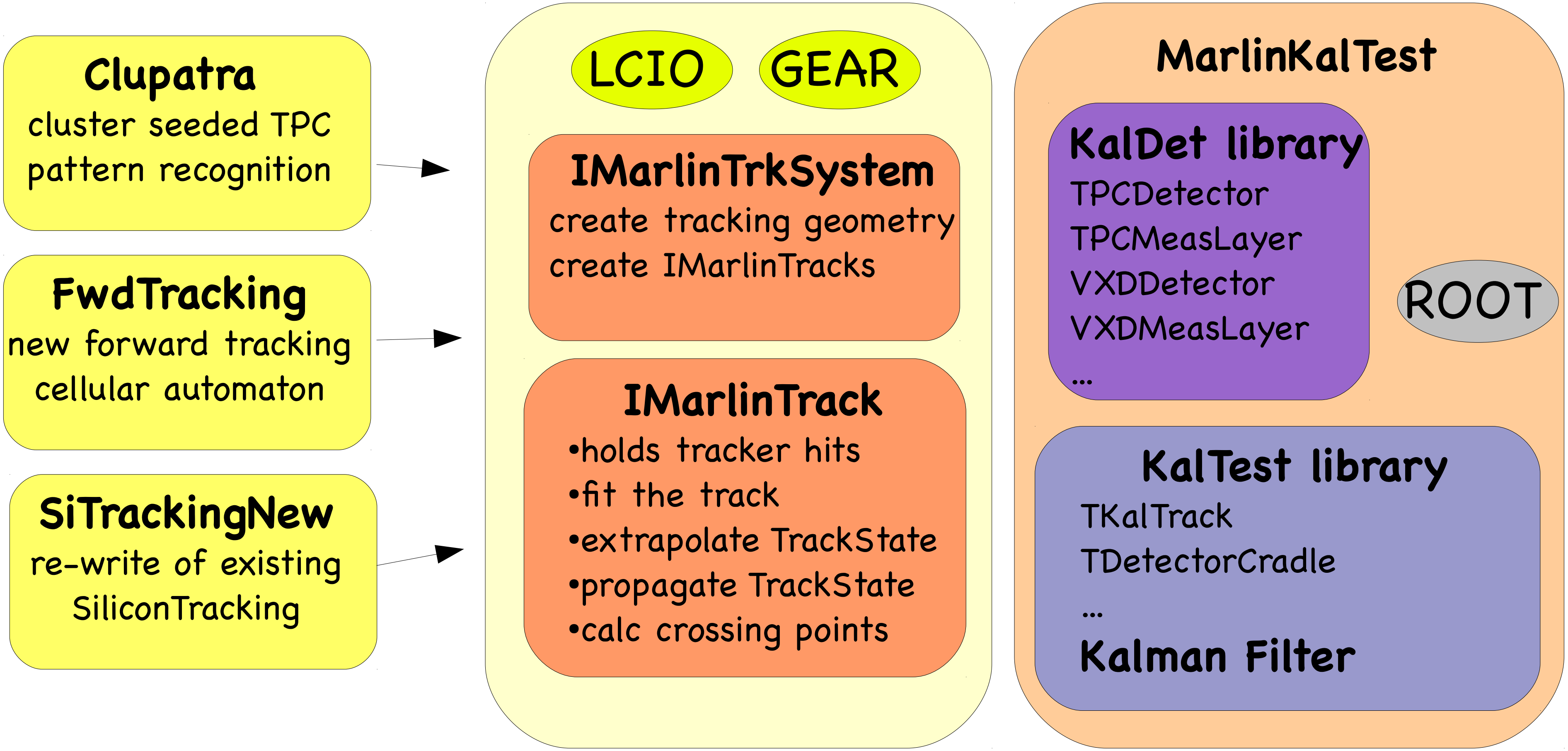


solid line:	generator status 1 → stable
dashed line:	generator status 2 → unstable
colored line:	relevant particles (stable or lifetime!=0)

- improved Lorentz boost for crossing angle (both: B.Vormwald)
  - apply to complete MCParticle list incl. Vertices

new C++ tracking: MarlinTrk

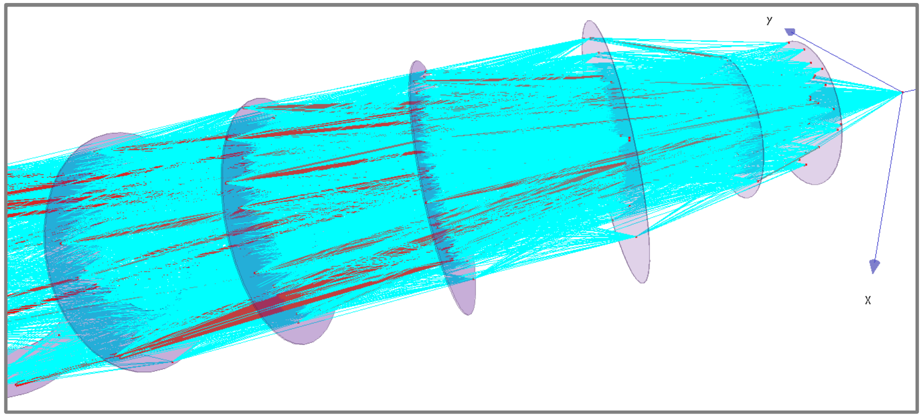
- new common API for developing tracking code (TPC, Silicon, Fwd)
- provides loose coupling between patrec and fitting
- defined abstract interface IMarlinTrk and implement using KalTest/KalDet
- currently lives in MarlinTrkProcessors

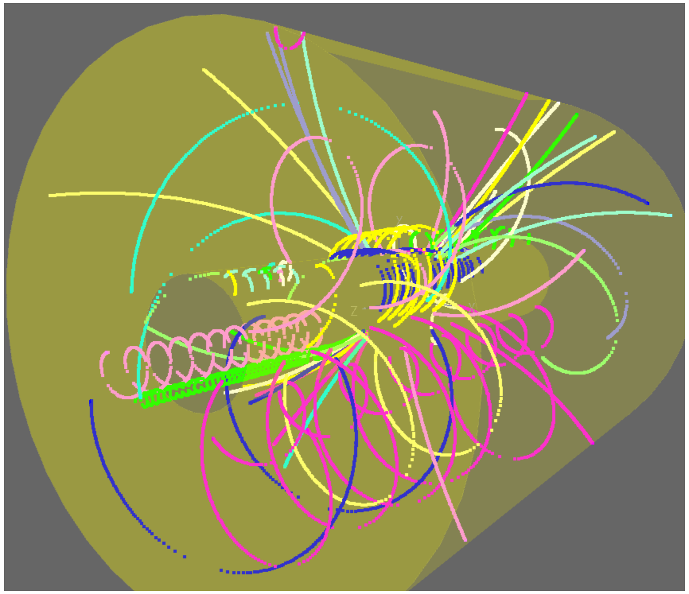


new C++ tracking: patrec activities

- ForwardTracking
  - new forward tracking patrecusing cellular automaton

    (R.Glattauer)
- Clupatra
  - new TPC patrec - recently:
    - fixed memory consumption
    - cleaned up code & algorithm
    - use new IMarlinTrk/MarlinKalTest
- MarlinTrkProcessors
  - rewrite of 'old' SiTracking and FullLDCTracking using MarlinTrk/MarlinKalTest
  - recently added proper 1D strip measurements





see dedicated talk on Wednesday

many more new developments ...

- ... not covered in this talk:
- many additional small developments in core tools:
  - command line parameters and improved logging in Marlin
  - ILDConfig utility to encode/decode TrackerHits cellIDs
  - fixed bug in LCIO w/ LCSplitWriter
  - Cone surfaces in KalTest/KalDet
- some new packages:
  - new BCal reconstruction in MarlinReco
  - SpacePointBuilder: combine 1D strip hits to "3D points"
- major improvements in:
  - PandoraPFA
  - LCFIPlus (-> dedicated talk on Wednesday)

=> see dedicated talks in ILD Analysis&Software Meetings

ILCSoft release v01-13-05

gear	v01-02
ilcutil	v00-03
lccd	v01-02
lcio	v02-01-01
CED	v01-05
Marlin	v01-02-01
RAIDA	v01-06-02
CEDViewer	v01-05
Clupatra	v00-05-01
CondDBMySQL	ILC-0-9-5
Druid	1.8
FastJet	2.4.2
FastJetClustering	v00-02
ForwardTracking	v01-02
Garlic	v2.0.4
KalDet	v01-07
KalTest	v01-05
LCFIPlus	v00-03
LCFIVertex	v00-06-01
MarlinFastJet	v00-01

MarlinKinfit	v00-01-01
MarlinPandora	v00-07
MarlinReco	v01-01-01
MarlinTPC	v00-09-01
MarlinTrk	v01-05
MarlinTrkProcessors	v01-04
MarlinUtil	v01-05-01
Mokka	mokka-07-07-p07
Overlay	v00-11-01
PandoraAnalysis	v00-03
PandoraPFANew	v00-08
pathfinder	v00-01-01
CLHEP	2.1.1.0
QT	4.7.4
cernlib	2006
geant4	9.5.p01
gsl	1.14
java	1.6.0
mysql	5.0.45
root	5.28.00f

core tools
application packages
external packages

afs reference installations

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

```
/afs/desy.de/project/ilcsoft/sw/_OS_/v01-13-05

_OS_: 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:

```
. /afs/desy.de/project/ilcsoft/sw/x86_64_gcc41_sl5/v01-13-05/init_ilcsoft.sh

Marlin mysteer.xml
```

- you can link your own libraries against these
- plan to have other OSs in the future (as requested !?)
- you can use ilcinstall tool for your own installation

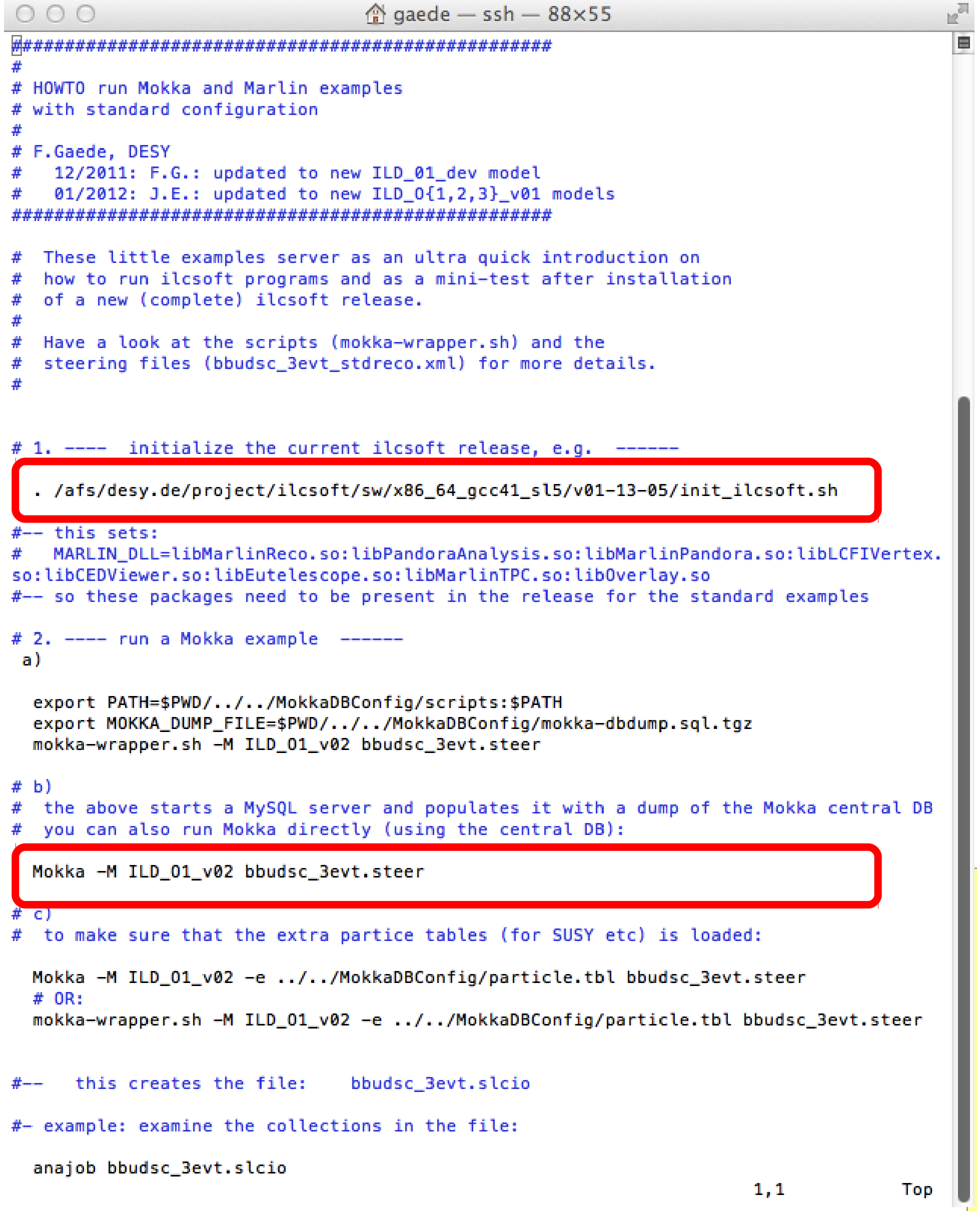
-> https://svnsrv.desy.de/viewvc/ilctools/ilcinstall/tags/v01-13-05/

new package ILDConfig

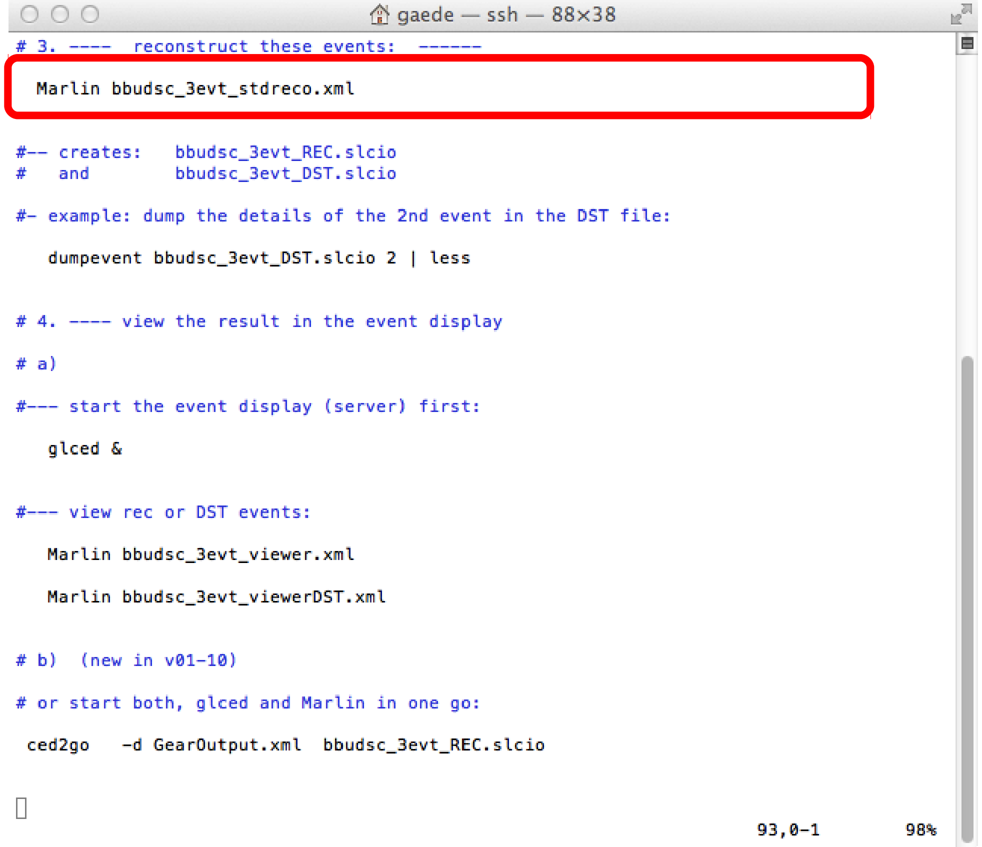
- **ILDConfig** - configuration and steering files:
  - combination of MokkaDBConfig, StandardConfig, LCFI_MokkaBasedNets:
  - **Mokka steering**
  - **Mokka models (DB dump)**
  - **Marlin steering (stdreco)**
  - **flavor tag NNs**
- release independent of iLCSoft in order to decouple configuration changes/releases from code releases
- version names start as corresponding iLCSoft release:
- current version ILDConfig **v01-13-05** to be followed by e.g:
  - v01-13-05-p01
  - v01-13-05-p02_special
  - v01-13-05-p03
- version tag will be used in filenames of centrally produced data to uniquely identify software and configuration !

Frank Gaede, KILC12, Daegu, Korea, Sep 23-27, 2012

ILD standard simulation/reconstruction



```
########################################################
#
# HOWTO run Mokka and Marlin examples
# with standard configuration
#
# F.Gaede, DESY
#   12/2011: F.G.: updated to new ILD_01_dev model
#   01/2012: J.E.: updated to new ILD_0{1,2,3}_v01 models
########################################################

#  These little examples server as an ultra quick introduction on
#  how to run ilcsoft programs and as a mini-test after installation
#  of a new (complete) ilcsoft release.
#
#  Have a look at the scripts (mokka-wrapper.sh) and the
#  steering files (bbudsc_3evt_stdreco.xml) for more details.
#


# 1. ----  initialize the current ilcsoft release, e.g.  ------

 . /afs/desy.de/project/ilcsoft/sw/x86_64_gcc41_sl5/v01-13-05/init_ilcsoft.sh

#-- this sets:
#   MARLIN_DLL=libMarlinReco.so:libPandoraAnalysis.so:libMarlinPandora.so:libLCFIVertex.
so:libCEDViewer.so:libEutelescope.so:libMarlinTPC.so:libOverlay.so
#-- so these packages need to be present in the release for the standard examples

# 2. ---- run a Mokka example  ------
 a)

  export PATH=$PWD/../../MokkaDBConfig/scripts:$PATH
  export MOKKA_DUMP_FILE=$PWD/../../MokkaDBConfig/mokka-dbdump.sql.tgz
  mokka-wrapper.sh -M ILD_01_v02 bbudsc_3evt.steer

# b)
#  the above starts a MySQL server and populates it with a dump of the Mokka central DB
#  you can also run Mokka directly (using the central DB):

  Mokka -M ILD_01_v02 bbudsc_3evt.steer

# c)
#  to make sure that the extra partice tables (for SUSY etc) is loaded:

  Mokka -M ILD_01_v02 -e ../../MokkaDBConfig/particle.tbl bbudsc_3evt.steer
  # OR:
  mokka-wrapper.sh -M ILD_01_v02 -e ../../MokkaDBConfig/particle.tbl bbudsc_3evt.steer


#--   this creates the file:    bbudsc_3evt.slcio

#- example: examine the collections in the file:

  anajob bbudsc_3evt.slcio
                                                                 1,1           Top
```



```
# 3. ----  reconstruct these events:  ------

  Marlin bbudsc_3evt_stdreco.xml


#-- creates:   bbudsc_3evt_REC.slcio
#   and        bbudsc_3evt_DST.slcio

#- example: dump the details of the 2nd event in the DST file:

   dumpevent bbudsc_3evt_DST.slcio 2 | less


# 4. ---- view the result in the event display

# a)

#--- start the event display (server) first:

   glced &


#--- view rec or DST events:

   Marlin bbudsc_3evt_viewer.xml

   Marlin bbudsc_3evt_viewerDST.xml


# b)  (new in v01-10)

# or start both, glced and Marlin in one go:

 ced2go   -d GearOutput.xml  bbudsc_3evt_REC.slcio


                                                                 93,0-1        98%
```

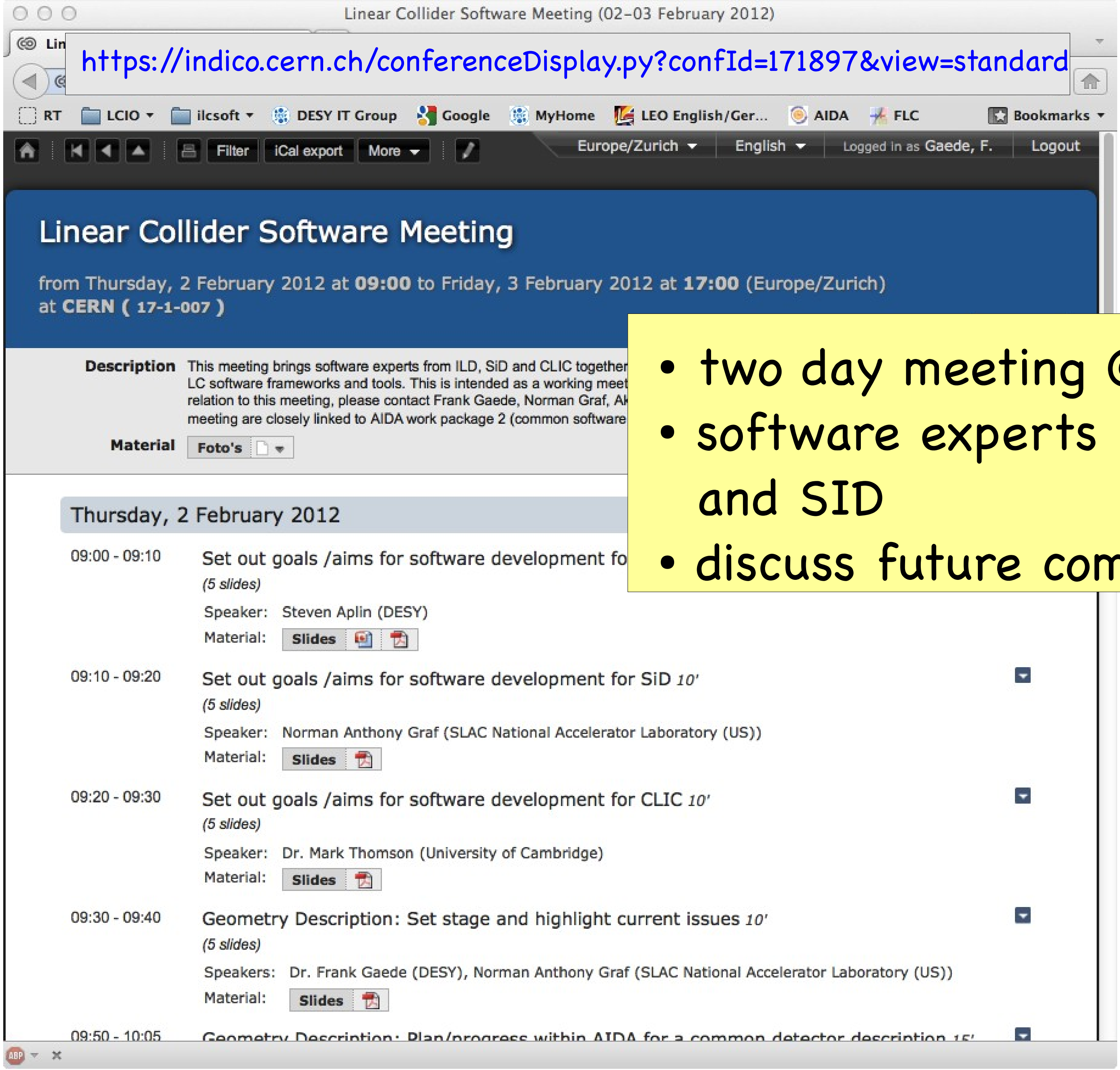
- StandardConfig/current sub package with current steering files for ILD
- defines **canonical ILD simulation and reconstruction**
- **README is "shortest introduction to running iLCSoft for ILD"**

PART I : Summary & Outlook

- very active development in iLCSoft framework driven by preparation for the ILD DBD:
  - LCIOv2, Gear extensions, new MarlinTrk and PatRec code,...
  - greatly improved realism in Mokka simulation - in particular for Si-Tracking detectors - currently validated
  - new technology options: FPCCD, SDHcal
  - not covered in this talk: PFA & LCFIVertex,...
- **we are in quite good shape - but some work still to be done until DBD !**
  - hope to start simulation for DBD soon...

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

PART2: LC Software Meeting



https://indico.cern.ch/conferenceDisplay.py?confId=171897&view=standard

Linear Collider Software Meeting

from Thursday, 2 February 2012 at **09:00** to Friday, 3 February 2012 at **17:00** (Europe/Zurich)
at **CERN (17-1-007)**

Description This meeting brings software experts from ILD, SiD and CLIC together
LC software frameworks and tools. This is intended as a working meet
relation to this meeting, please contact Frank Gaede, Norman Graf, Ak
meeting are closely linked to AIDA work package 2 (common software

Material Foto's

Thursday, 2 February 2012

09:00 - 09:10 Set out goals /aims for software development fo
(5 slides)
Speaker: Steven Aplin (DESY)
Material: **Slides**

09:10 - 09:20 Set out goals /aims for software development for SiD *10'*
(5 slides)
Speaker: Norman Anthony Graf (SLAC National Accelerator Laboratory (US))
Material: **Slides**

09:20 - 09:30 Set out goals /aims for software development for CLIC *10'*
(5 slides)
Speaker: Dr. Mark Thomson (University of Cambridge)
Material: **Slides**

09:30 - 09:40 Geometry Description: Set stage and highlight current issues *10'*
(5 slides)
Speakers: Dr. Frank Gaede (DESY), Norman Anthony Graf (SLAC National Accelerator Laboratory (US))
Material: **Slides**

09:50 - 10:05 Geometry Description: Plan/progress within AIDA for a common detector description *15'*

- two day meeting @ CERN in Feb.'12
- software experts from CLIC, ILD and SID
- discuss future common developments

Linear Collider Software Meeting Close Out

Frank Gaede (DESY), Norman Graf (SLAC), Akiya Miyamoto (KEK), Mark Thomson (U.Cambridge)

CERN, Feb 2-3, 2012

common simulation

- general consensus to work towards a common simulation application
  - build on the ongoing work for detector description and geometry (AIDA WP2)
- setup a working group to work towards that goal
- should start quite soon
  - this summer when DBD software work reduces
- define a geometry API for reconstruction, e.g. Gear

PFA

- need to work on SDHcal and DHcal reconstruction
- develop clustering algorithms in pandora

LCIO

- no immediate action items identified
- Whizard will provide LCIO MCParticle files in the future

Common production

- no immediate action items identified
- already very good collaboration and splitting of the work load by Generator group and SCTG
- analysis groups need to make requests for number of (bg) events they need
    - backed up by 4-vector (fastsim) study

Tracking

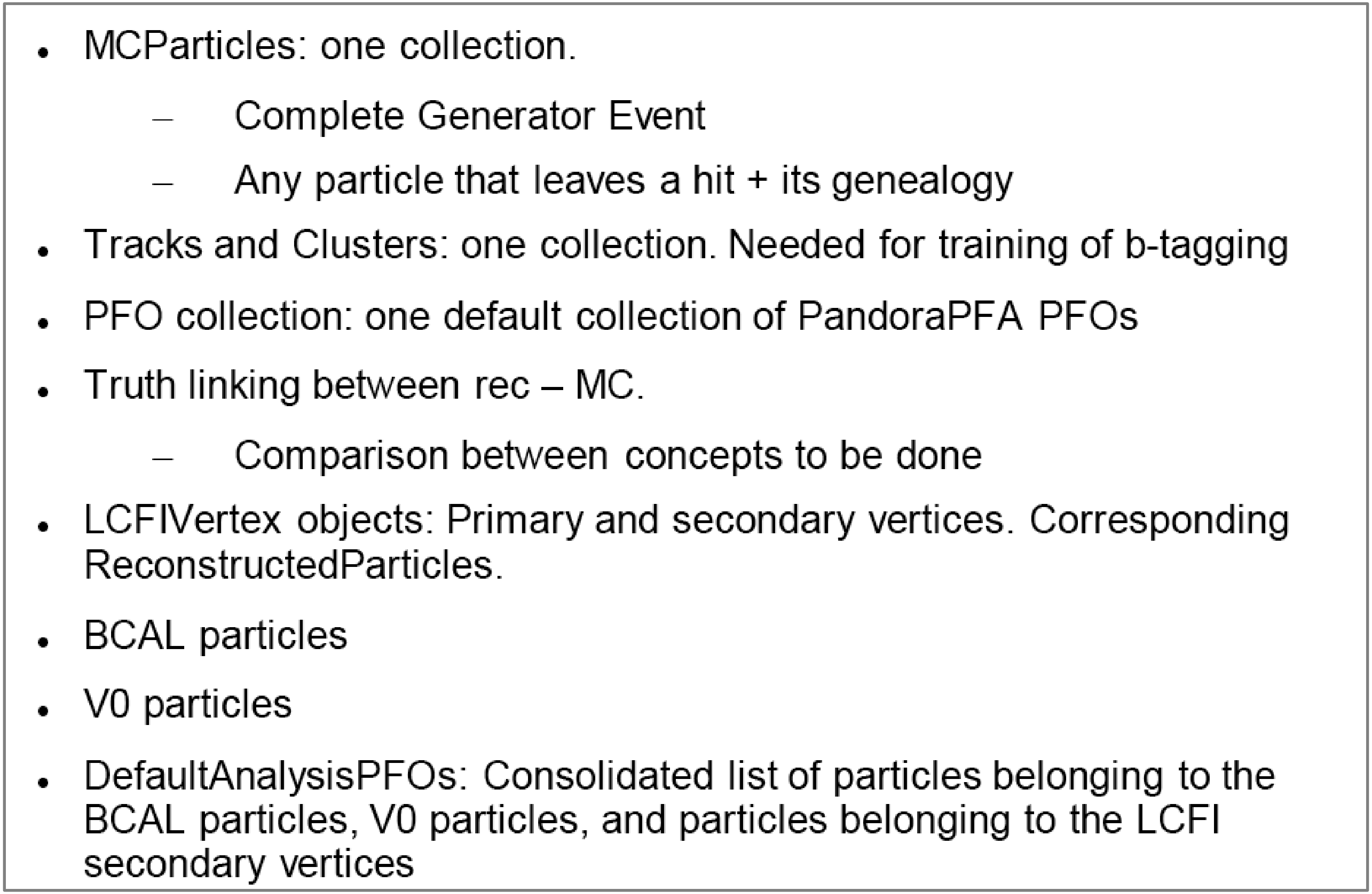
- general consensus to work towards a common track reconstruction package in C++
- in context of AIDA WP2
- implementation of FTF and TRF like algorithms for Si-Tracking

LCFIPlus

- lots of progress with vertexing and flavor tagging
- some minor issues to be addressed
- e.g. singleton pattern for data model, documentation

Common DST Format

- reached consensus on collections on DST:



- MCParticles: one collection.
  - Complete Generator Event
  - Any particle that leaves a hit + its genealogy
- Tracks and Clusters: one collection. Needed for training of b-tagging
- PFO collection: one default collection of PandoraPFA PFOs
- Truth linking between rec – MC.
  - Comparison between concepts to be done
- LCFIVertex objects: Primary and secondary vertices. Corresponding ReconstructedParticles.
- BCAL particles
- V0 particles
- DefaultAnalysisPFOs: Consolidated list of particles belonging to the BCAL particles, V0 particles, and particles belonging to the LCFI secondary vertices