

Introduction to Software Tools

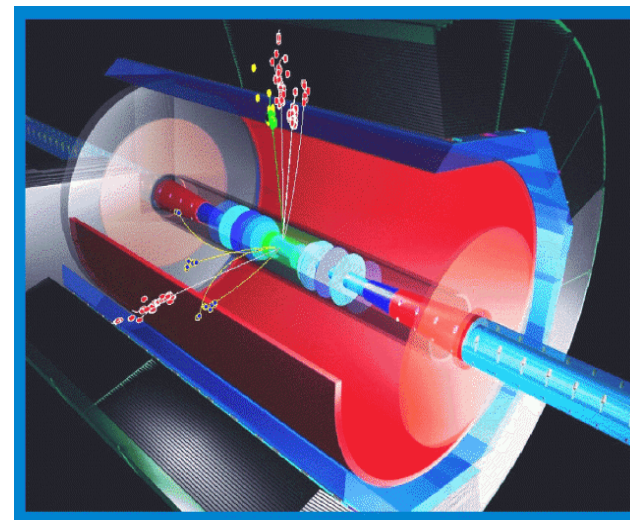
ILC-LDC Simulation and Reconstruction

Frank Gaede
DESY

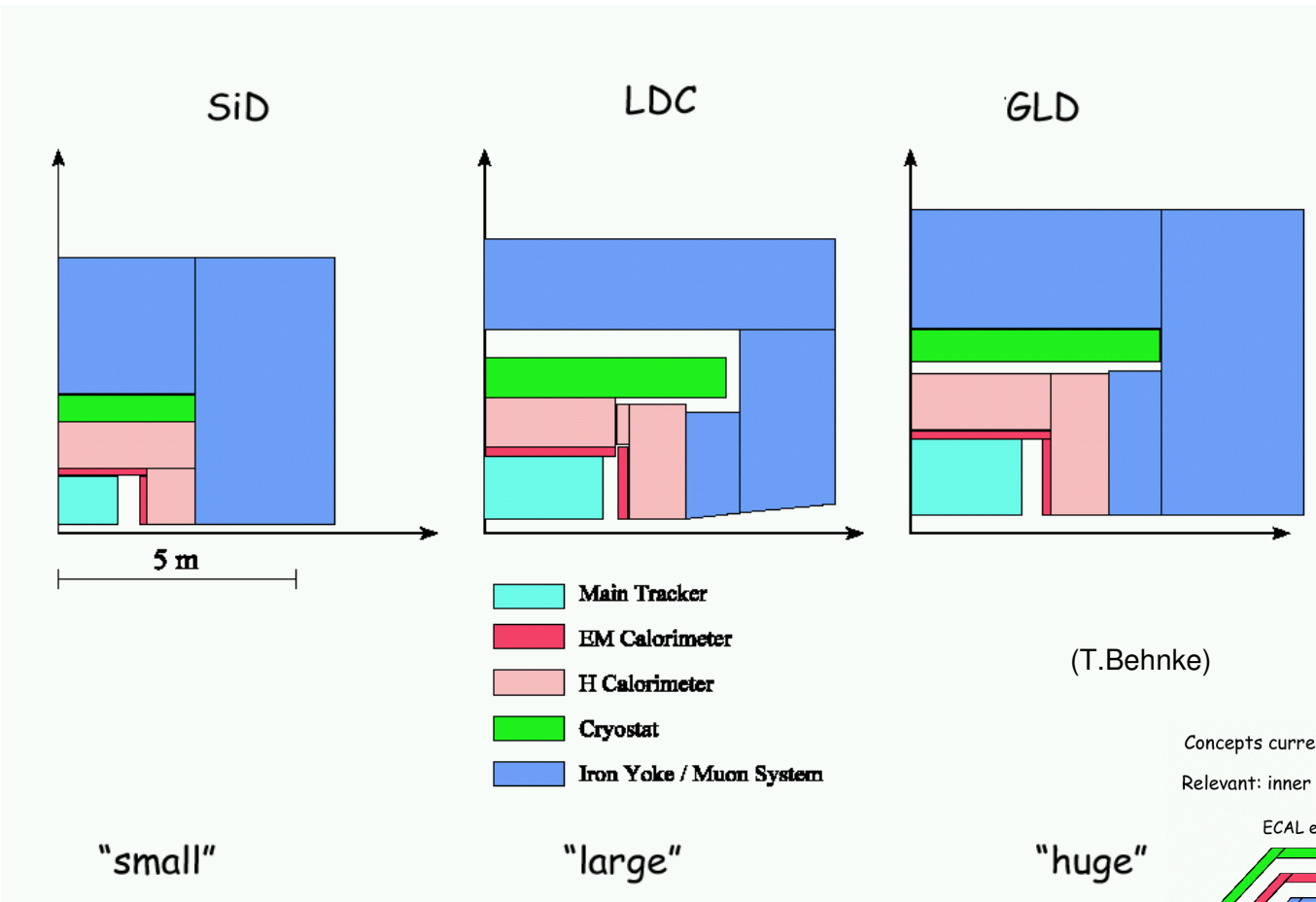
ECFA ILC Workshop, Vienna
November 14-17, 2005

Outline

- Introduction - overview international software
- Central tools for LDC study
 - **LCIO** – data model & persistency
 - **Simdet** – fast simulation
 - **Brahms** – geant3 full simulation and reconstruction
 - **Mokka** – geant4 full simulation
 - **Marlin** – C++ reconstruction framework
 - **LCCD** - conditions data toolkit
 - **GEAR** – geometry description
- Summary & Outlook

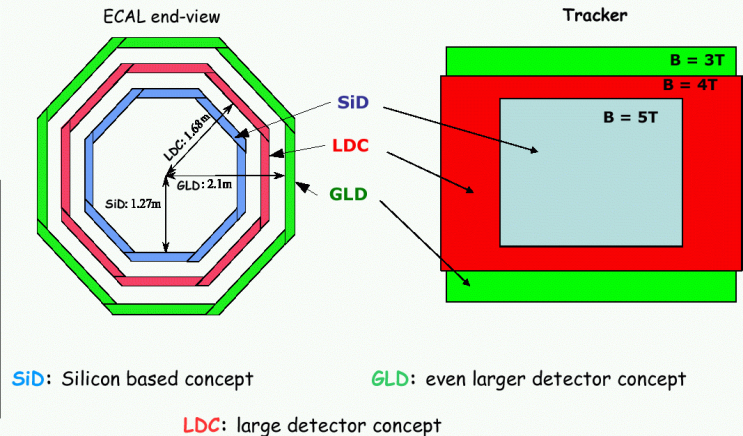


Detector Concept Study



three interregional detector concept studies ongoing

Concepts currently studies differ mainly in **SIZE** and **aspect ratio**
 Relevant: inner radius of ECAL: defines the overall scale



Need (common?) **Simulation** and **Reconstruction** software to study detector concepts' performance !

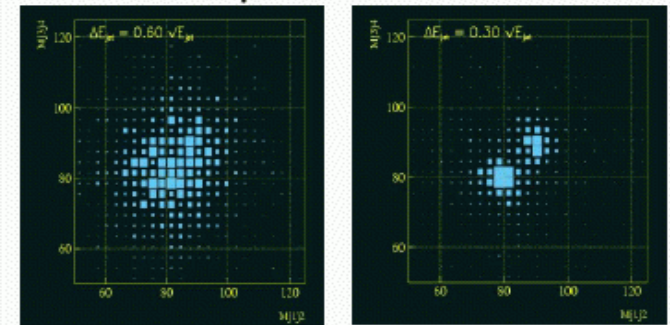
Reconstruction @ the ILC

- 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)

WW-ZZ separation



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

$$\sigma_{E_{jet}}^2 = \epsilon_{trk}^2 \sum_i E_{trk,i}^4 + \epsilon_{ECal}^2 E_{ECal}^2 + \epsilon_{HCal}^2 E_{HCal}^2 + \sigma_{confusion}^2$$
$$\epsilon_{trk} = \delta(1/p) \approx 5 \cdot 10^{-5}, \quad \epsilon_{ECal} = \frac{\delta E}{\sqrt{E}} \approx 0.1, \quad \epsilon_{HCal} \approx 0.5$$

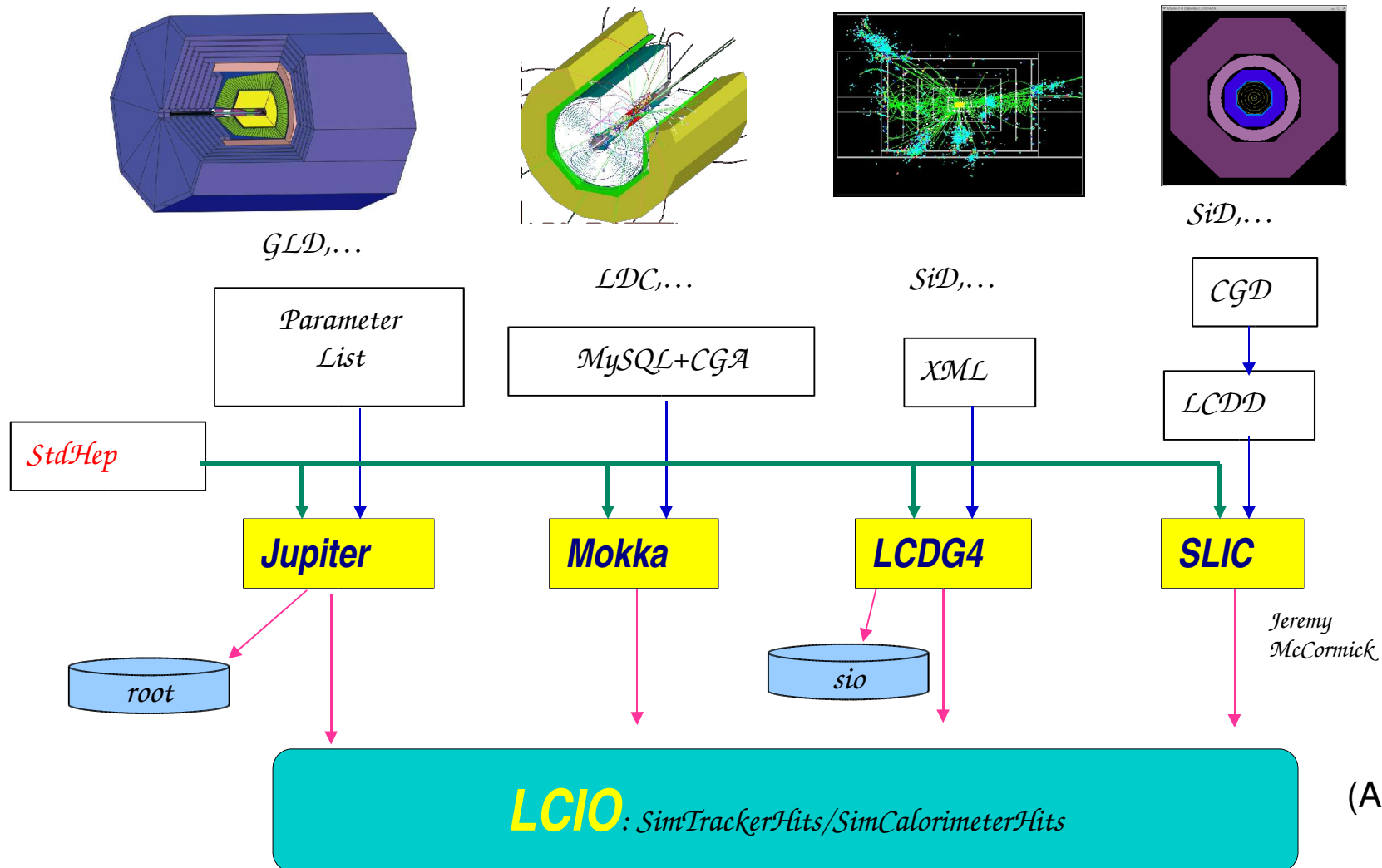
Need (common?) software framework
to develop, check and compare
different Pflow algorithms

ILC software packages

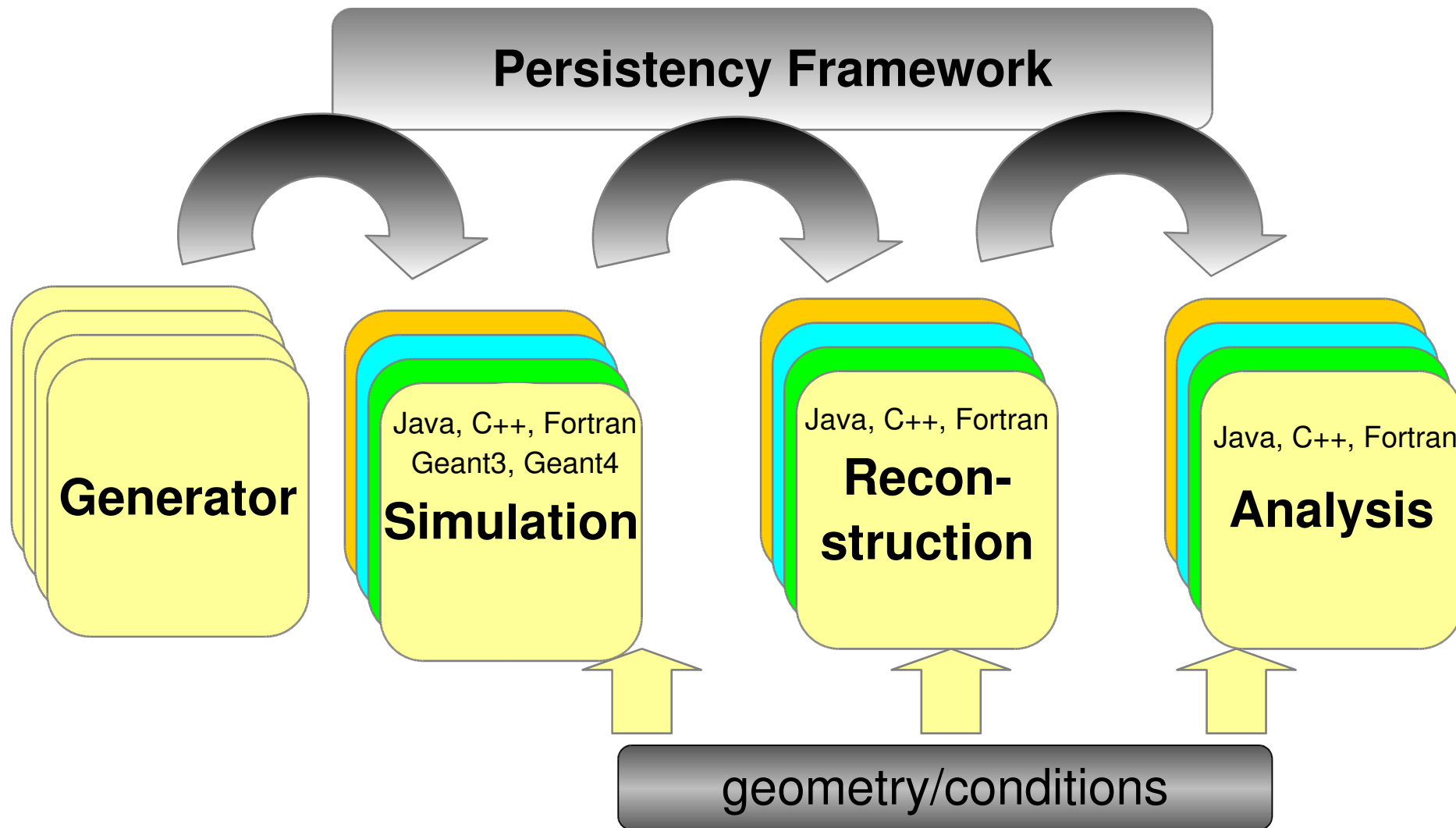
	Description	Detector	Language	IO-Format	Region
Simdet	fast Monte Carlo	TeslaTDR	Fortran	StdHep/LCIO	EU
SGV	fast Monte Carlo	simple Geometry, flexible	Fortran	None (LCIO)	EU
Lelaps	fast Monte Carlo	SiD, flexible	C++	SIO, LCIO	US
Mokka	full simulation – Geant4	TeslaTDR, LDC, flexible	C++	ASCI, LCIO	EU
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, LCIO	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

ILC Simulation Frameworks (Geant4)

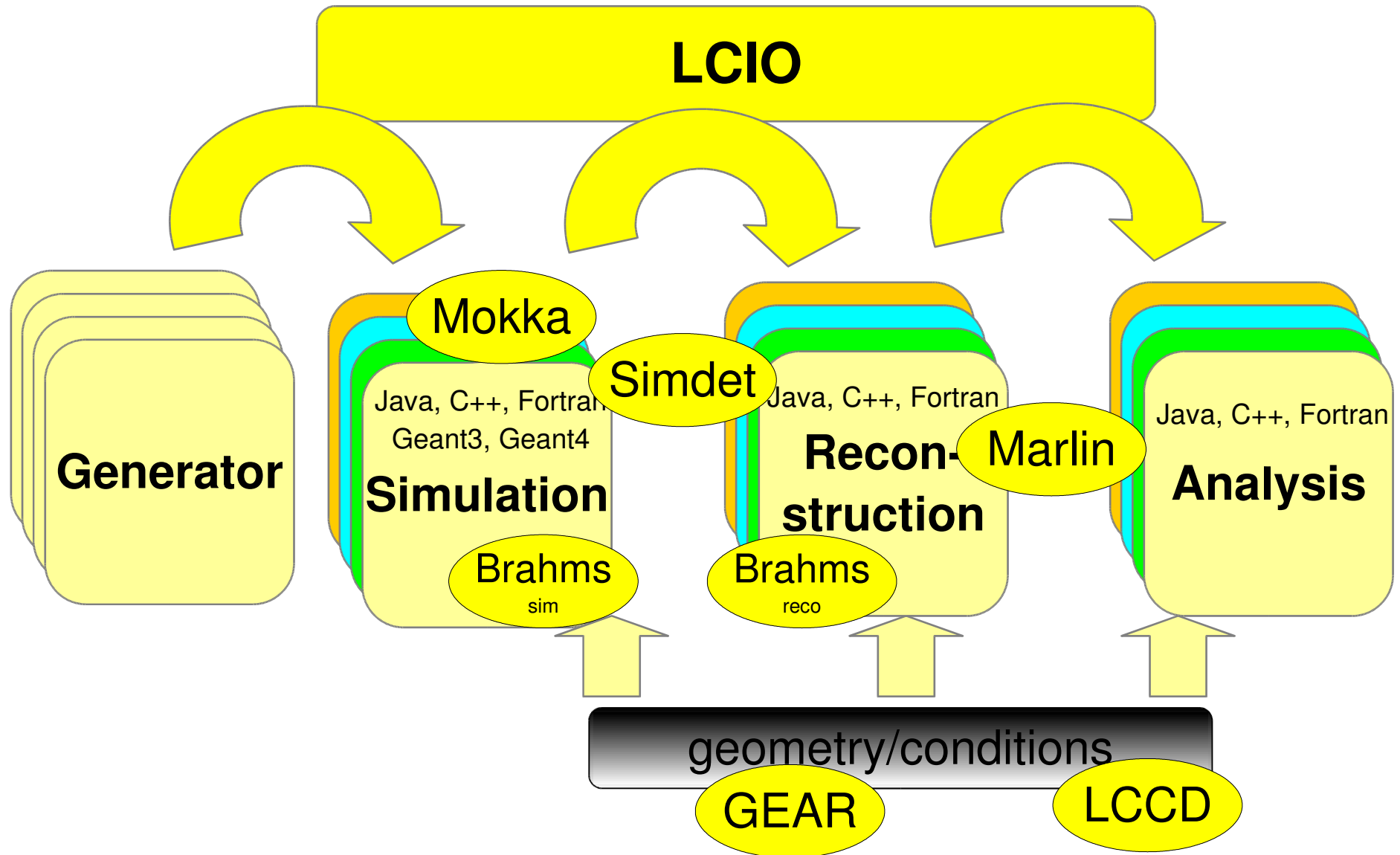
- *Geant4, StdHep and LCIO are common feature*
- *Each trying to be generic with different approach
different ways to define geometries*



ILC software chain



ILC software tools used for LDC



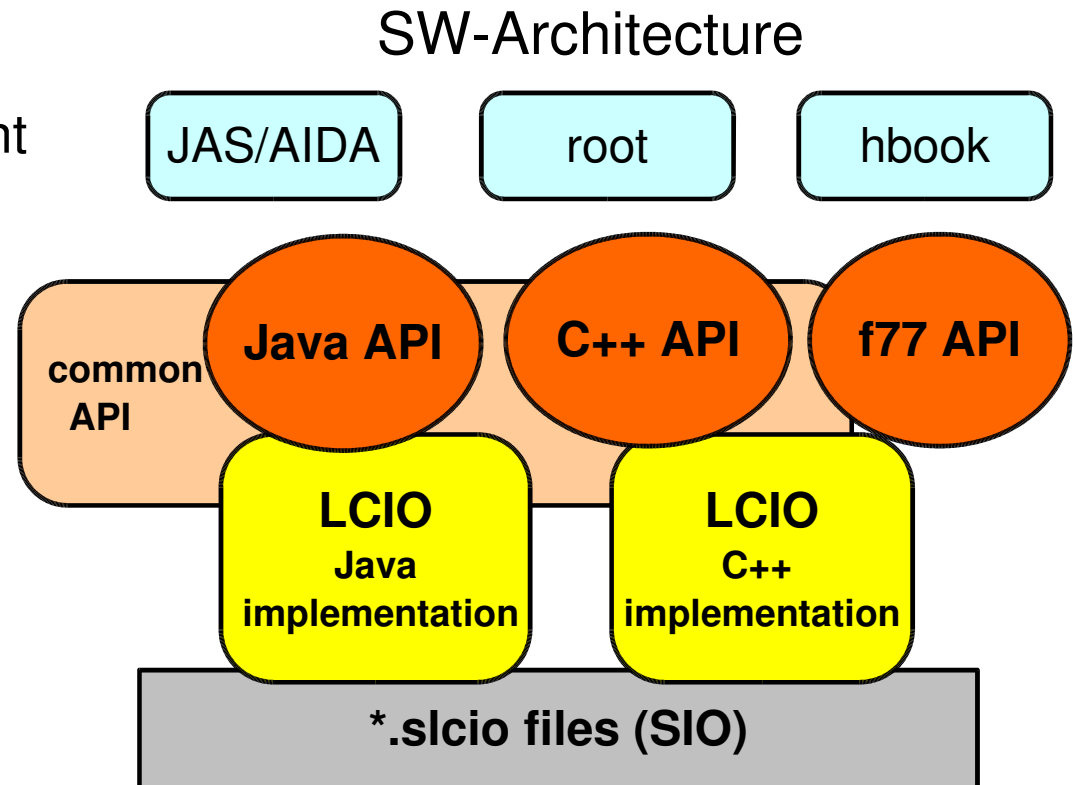
LCIO overview

- DESY and SLAC joined project:
 - provide common basis for ILC software
- Features:
 - Java, C++ and f77 (!) API
 - extensible data model for current and future simulation and testbeam studies
 - user code separated from concrete data format
 - no dependency on other frameworks

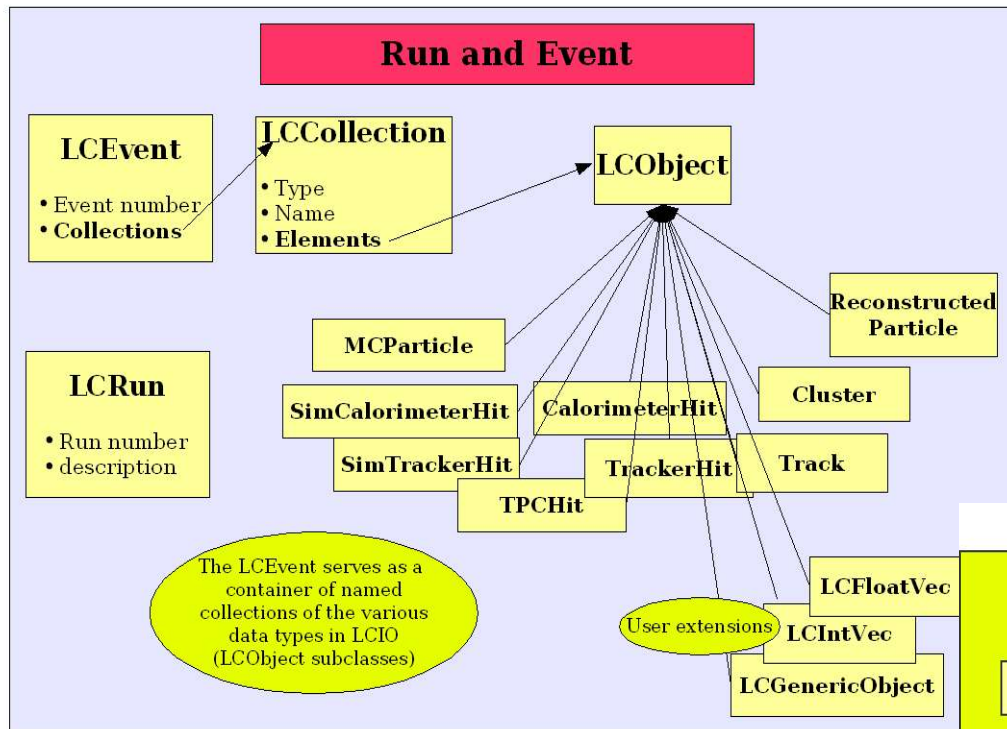
simple & lightweight

new release: **v01-06**

now de facto standard
persistency & datamodel
for ILC software

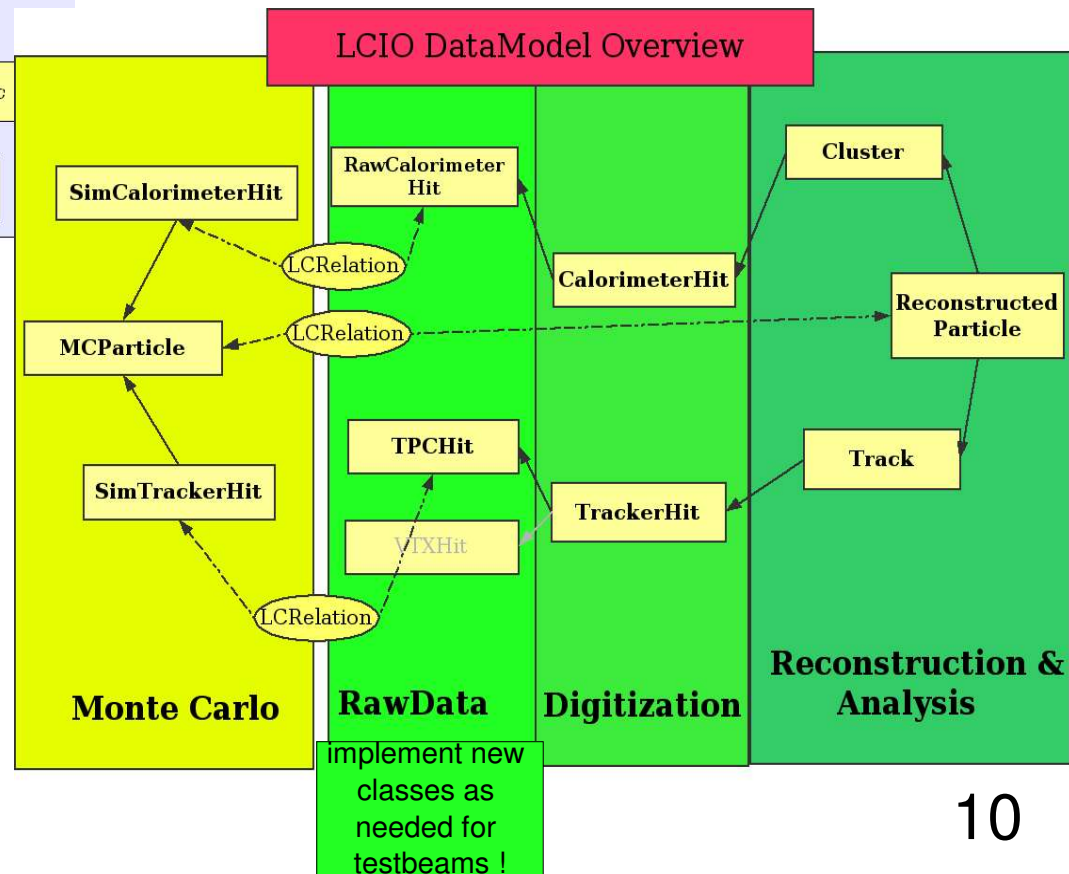


LCIO data model



event serves as container of untyped collections

hierarchy of data objects in the event



Simulation tools: Simdet, Brahms

● **SIMDET**

writes LCIO

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

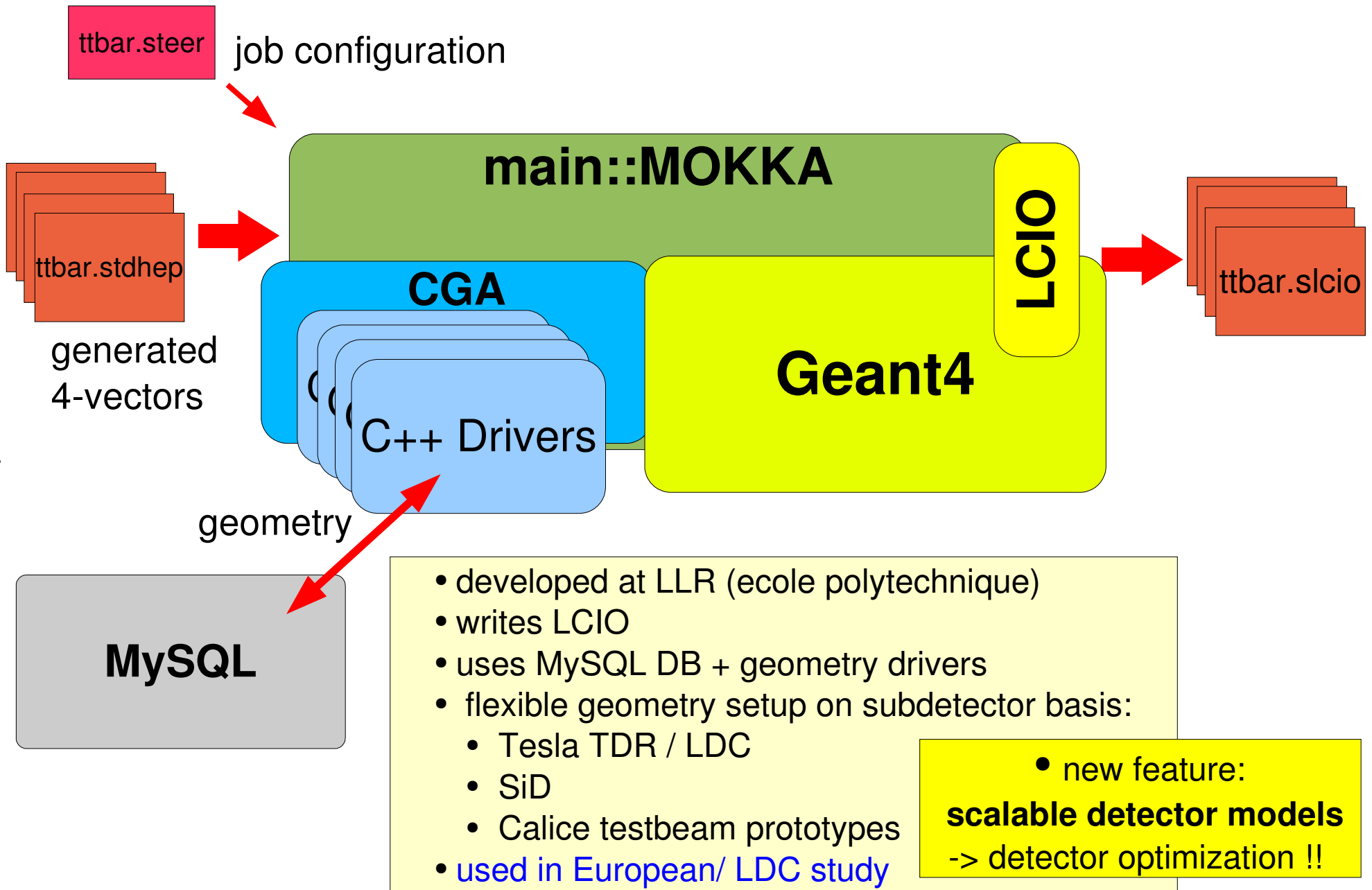
● **Brahms**

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

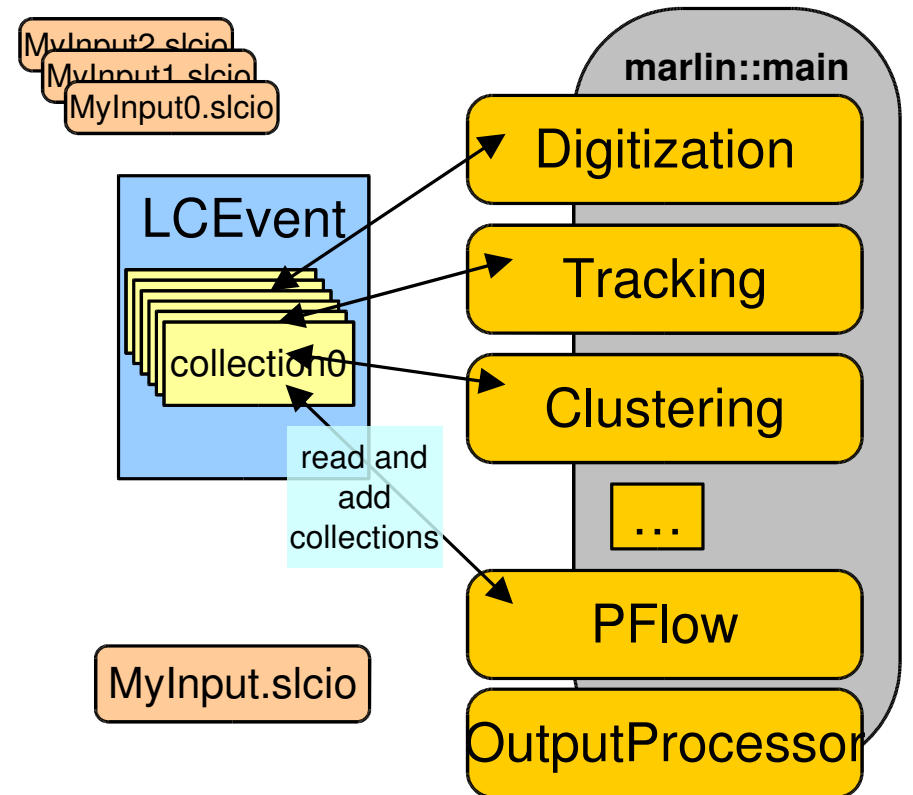
Mokka overview



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
 - **Plug&Play** of processors



Gear

GEometry API for RReconstruction

```
<gear>
  <!--
    Example XML file for GEAR describing the LDC detector
  -->
  <detectors>
    <detector id="0" name="TPCTest" geartype="TPCParameters" type="TPCParameters">
      <maxDriftLength value="2500."/>
      <driftVelocity value=""/>
      <readoutFrequency value="10"/>
      <PadRowLayout2D type="FixedPadSizeDiskLayout" rMin="386.0"
        maxRow="200" padGap="0.0"/>
      <parameter name="tpcRPhiResMax" type="double"> 0.16 </parameter>
      <parameter name="tpcZRes" type="double"> 1.0 </parameter>
      <parameter name="tpcPixRP" type="double"> 1.0 </parameter>
      <parameter name="tpcPixZ" type="double"> 1.4 </parameter>
      <parameter name="tpcIonPotential" type="double"> 0.00000003 </parameter>
    </detector>
    <detector name="EcalBarrel" geartype="CalorimeterParameters">
      <layout type="Barrel" symmetry="8" phi0="0.0"/>
      <dimensions inner_r="1698.85" outer_r="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="2820.0" outer_z="2820.0"/>
      <layer repeat="30" thickness="3.9" absorberThickness="2.5"/>
      <layer repeat="10" thickness="6.7" absorberThickness="5.3"/>
    </detector>
  </detectors>
</gear>
```

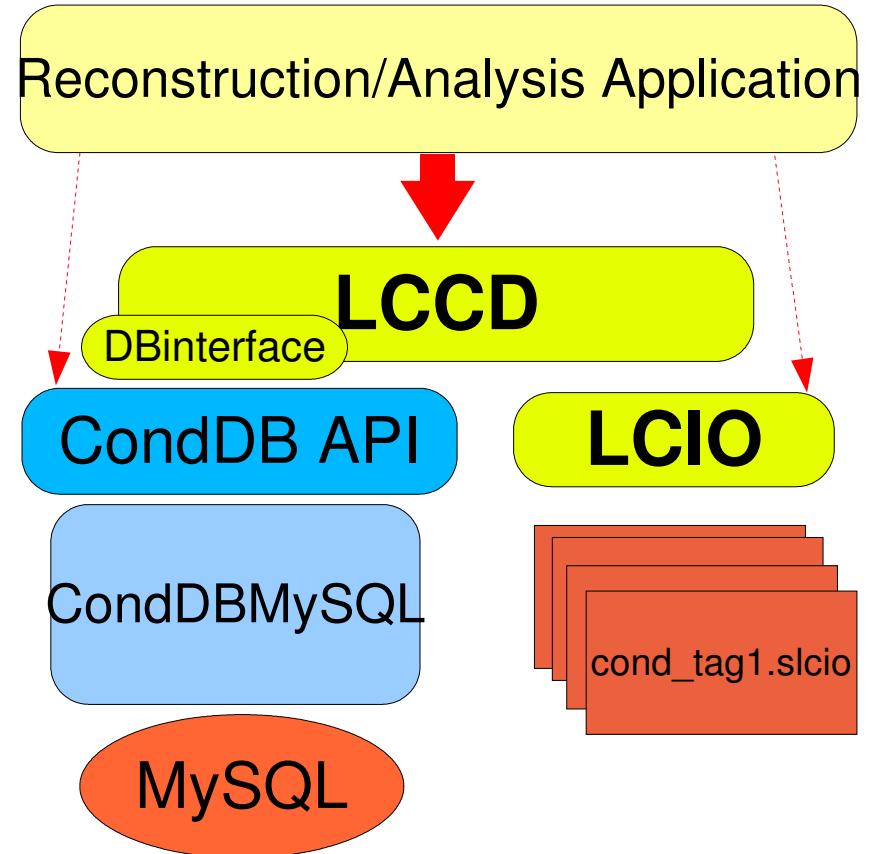
compatible with US – compact format

- 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

LCCD

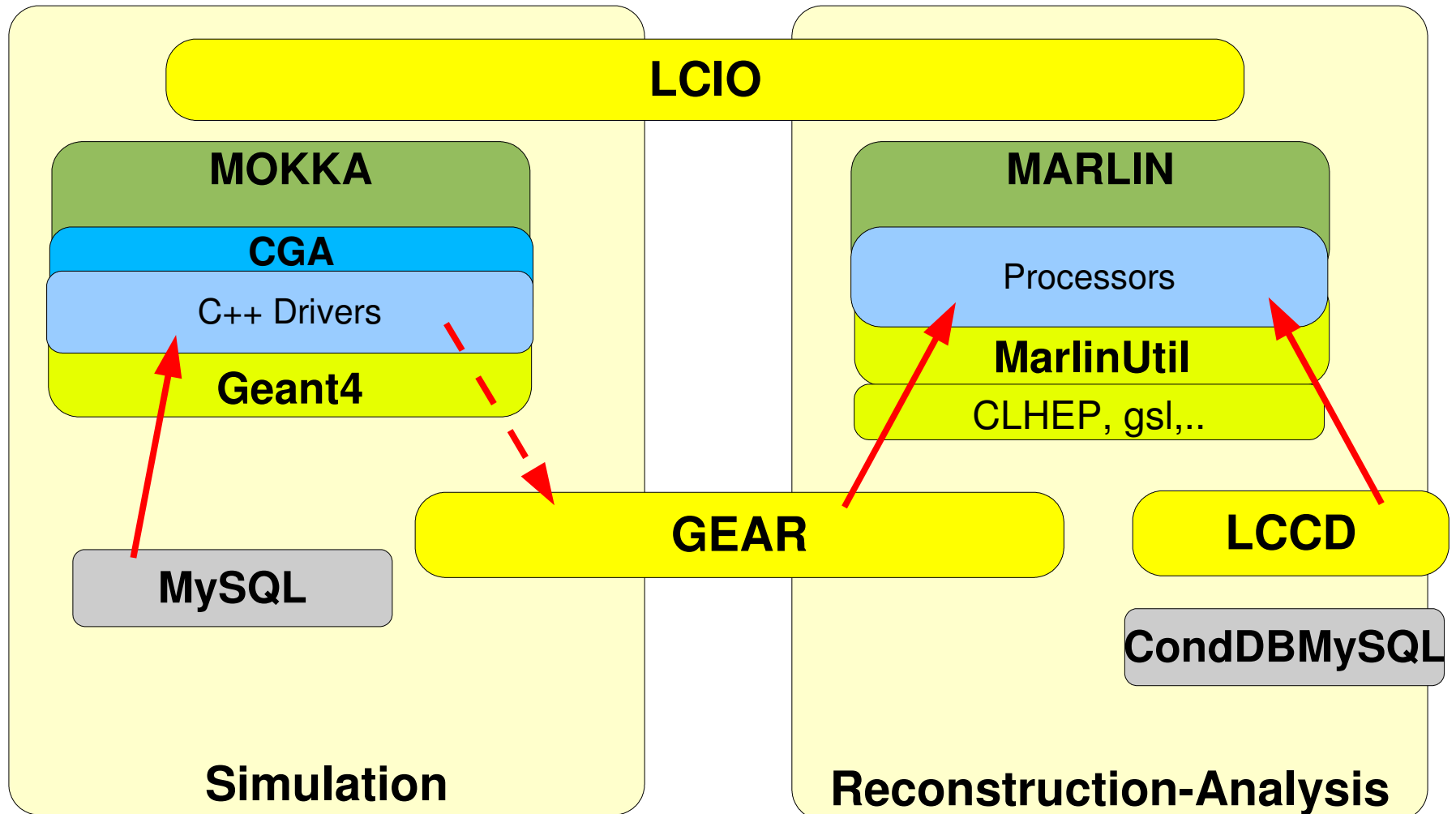
Linear **C**ollider **C**onditions **D**ata Toolkit

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



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

LDC simulation 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. together with MAGIC-clustering

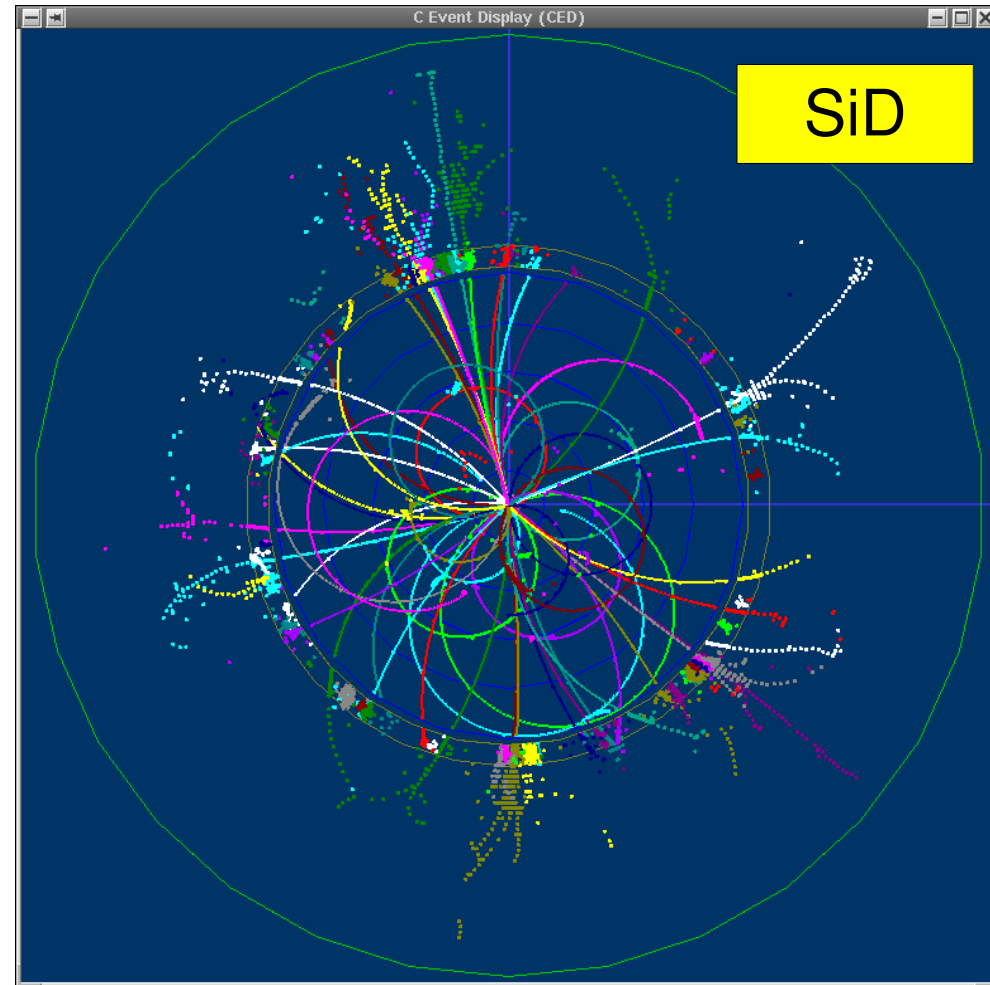
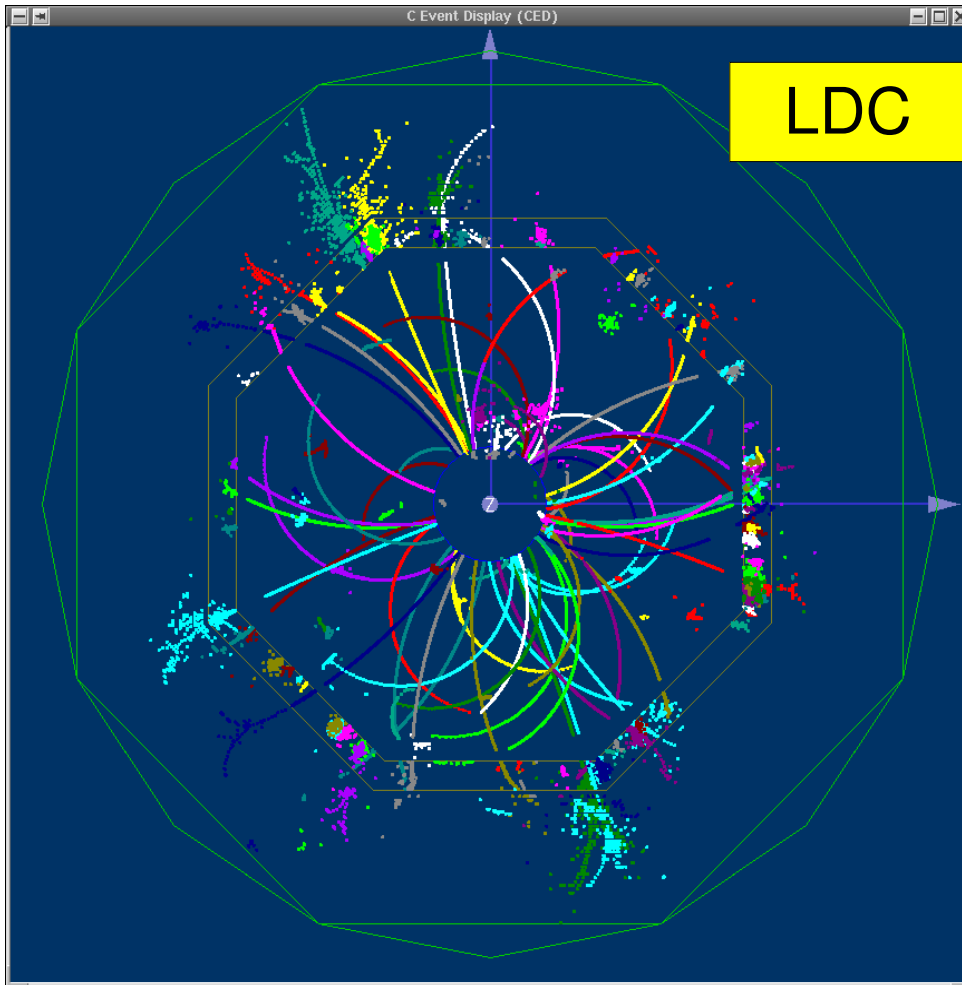
MarlinReco packages

- **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

most MarlinReco processors (algorithms) are geometry independent
→ they can be applied to other detector concepts (via Gear file)

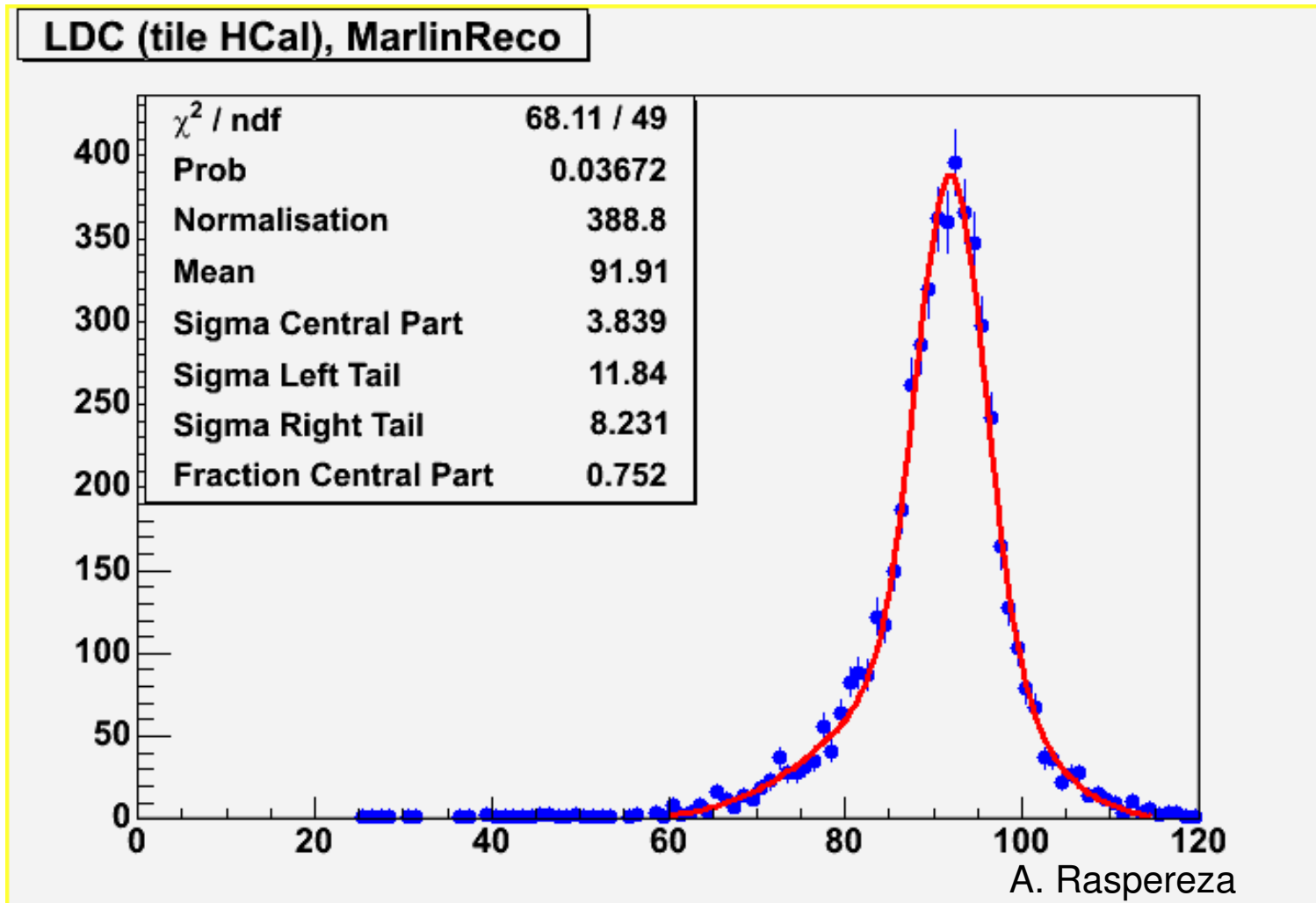
ttbar events with MarlinReco

No cheaters, only full reconstruction



Initial results

- $e^+e^- \rightarrow Z \rightarrow qq\bar{q}$ using full reconstruction



“How can I create my own Zpole plot ?”

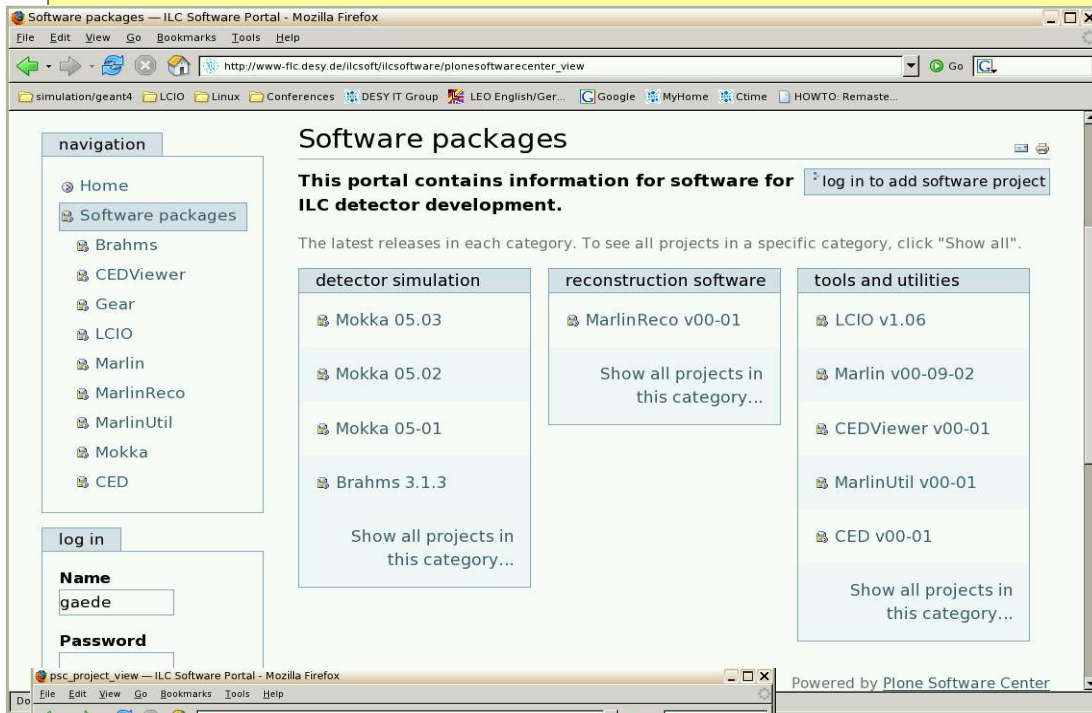
“Download, install and run the software tools !”

“Where should I start?”

“At the software portal !”

ILC software portal

Frank Gaede, ECFA ILC Workshop, Vienna, Nov 14-17, 2005



Software packages — ILC Software Portal - Mozilla Firefox

http://www-flc.desy.de/ilcsoft/ilcsoftware/plonessoftwarecenter_view

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

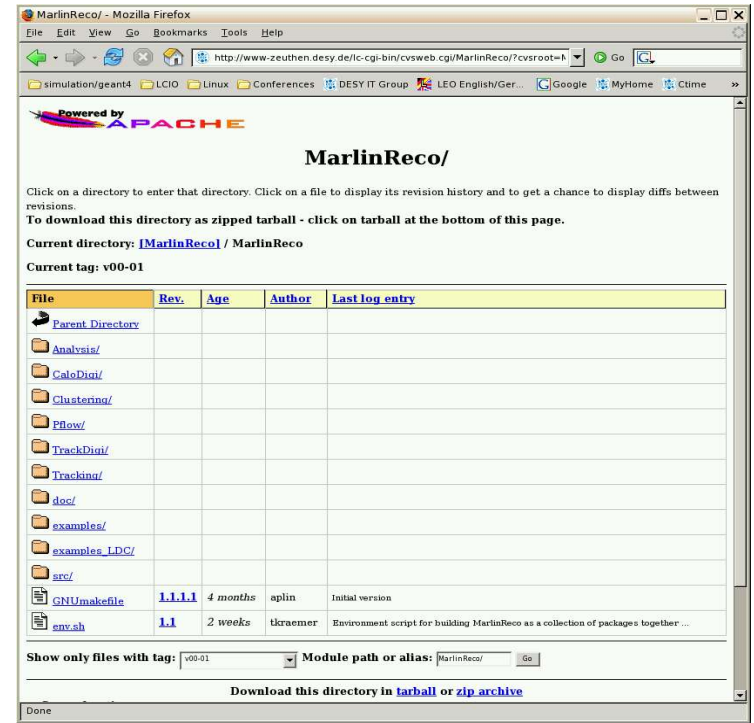
Software packages

This portal contains information for software for ILC detector development. [log in to add software project](#)

The latest releases in each category. To see all projects in a specific category, click "Show all".

detector simulation	reconstruction software	tools and utilities
Mokka 05.03	MarlinReco v00-01	LCIO v1.06
Mokka 05.02	Show all projects in this category...	Marlin v00-09-02
Mokka 05-01		CEDViewer v00-01
Brahm3 3.1.3		MarlinUtil v00-01
Show all projects in this category...		CED v00-01
		Show all projects in this category...

Powered by Plone Software Center



MarlinReco/ - Mozilla Firefox

http://www-zeuthen.desy.de/ilc-cgi-bin/cvsweb.cgi/MarlinReco/?cvsroot=lc

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

Powered by **APACHE**

MarlinReco/

Click on a directory to enter that directory. Click on a file to display its revision history and to get a chance to display diffs between revisions.

To download this directory as zipped tarball - click on tarball at the bottom of this page.

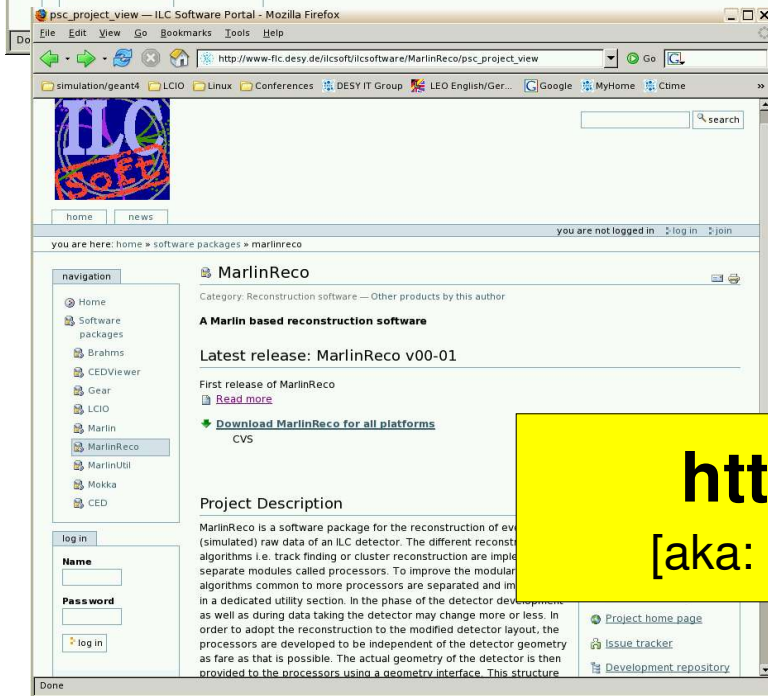
Current directory: [MarlinReco/](#) / MarlinReco

Current tag: v00-01

File	Rev.	Age	Author	Last log entry
Parent Directory				
Analysis/				
CaloDist/				
Clustering/				
Pflow/				
TrackDist/				
Tracking/				
doc/				
examples/				
examples_LDC/				
src/				
GNUmakefile	1.1.1.1	4 months	aplin	Initial version
env.sh	1.1	2 weeks	ukraemer	Environment script for building MarlinReco as a collection of packages together...

Show only files with tag: v00-01 Module path or alias: [MarlinReco/](#) [Go](#)

[Download this directory in tarball or zip archive](#)



pssc_project_view — ILC Software Portal - Mozilla Firefox

http://www-flc.desy.de/ilcsoft/ilcsoftware/MarlinReco/pssc_project_view

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

MarlinReco

Category: Reconstruction software — Other products by this author

A Marlin based reconstruction software

Latest release: MarlinReco v00-01

First release of MarlinReco [Read more](#)

[Download MarlinReco for all platforms](#)

CVS

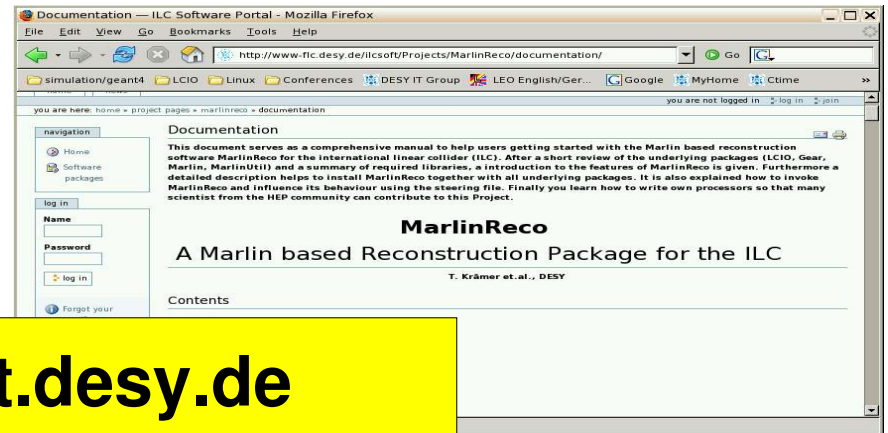
Project Description

MarlinReco is a software package for the reconstruction of event (simulated) raw data of an ILC detector. The different reconstruction algorithms i.e. track finding or cluster reconstruction are implemented as separate modules called processors. To improve the modular architecture, the processors are separated and implemented in a dedicated utility section. In the phase of the detector development as well as during data taking the detector may change more or less. In order to adopt the reconstruction to the modified detector layout, the processors are developed to be independent of the detector geometry as far as that is possible. The actual geometry of the detector is then provided to the processors using a geometry interface. This structure

[Project home page](#)

[Issue tracker](#)

[Development repository](#)



Documentation — ILC Software Portal - Mozilla Firefox

http://www-flc.desy.de/ilcsoft/Projects/MarlinReco/documentation/

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

Documentation

This document serves as a comprehensive manual to help users getting started with the Marlin based reconstruction software MarlinReco for the international linear collider (ILC). After a short review of the underlying packages (LCIO, Gear, Marlin, MarlinUtil) and a summary of required libraries, an introduction to the features of MarlinReco is given. Furthermore a detailed description helps to install MarlinReco together with all underlying packages. It is also explained how to invoke MarlinReco and influence its behaviour using the steering file. Finally you learn how to write own processors so that many scientist from the HEP community can contribute to this Project.

MarlinReco

A Marlin based Reconstruction Package for the ILC

T. Krämer et al., DESY

Contents

<http://ilcsoft.desy.de>
[aka: <http://www-flc.desy.de/ilcsoft>]

LDC simulation “mass production”

- detector optimization – vary
 - B,R_TPC,L_TPC,...
- need considerable number of events with detector parameter variations for benchmark reactions
- @ DESY:
 - started to produce these files on the **grid** for VO ILC
 - resolved some issues with Mokka and Geant4
 - will provide database with available data files
 - use **grid tools** to distribute the data !

- provide the simulated data that's needed
- exercise the software & computing infrastructure

International Linear Collider MC Production Navigation Bar

[Search Database](#) [Browse Database](#)

Search Database

Run Number:	<input type="text"/>
Date of Production [yyyy-mm-dd]:	<input type="text"/>
Process:	<input type="text"/>
Event Generator:	<input type="text"/>
Simulation:	<input type="text"/>
Detector Model:	<input type="text" value="LDC00Sc"/>
B Field [T]:	<input type="text"/>
Center of Mass Energy [GeV]:	<input type="text"/>

Oliver Wendt
Last modified: Wed Nov 09 11:22:33 MEST 2005

International Linear Collider MC Production Navigation Bar

[Search Database](#) [Browse Database](#)

MC data-files matching your query:

Run Number	Event Generator	Simulation	Detector Model	B Field [T]	Center of Mass Energy [GeV]
zpole_noisr_LDC00Sc_6.0T_r1690_I2730_LCPhys_5	Pythia 6.321	Mokka 5.03pre	LDC00Sc	6	91.2
zpole_noisr_LDC00Sc_6.0T_r1690_I2730_LCPhys_4	Pythia 6.321	Mokka 5.03pre	LDC00Sc	6	91.2
zpole_noisr_LDC00Sc_6.0T_r1690_I2730_LCPhys_3	Pythia 6.321	Mokka 5.03pre	LDC00Sc	6	91.2
zpole_noisr_LDC00Sc_6.0T_r1690_I2730_LCPhys_2	Pythia 6.321	Mokka 5.03pre	LDC00Sc	6	91.2
zpole_noisr_LDC00Sc_6.0T_r1690_I2730_LCPhys_1	Pythia 6.321	Mokka 5.03pre	LDC00Sc	6	91.2
zpole_noisr_LDC00Sc_4.0T_r1690_I2730_LCPhys_5	Pythia 6.321	Mokka 5.03pre	LDC00Sc	4	91.2
zpole_noisr_LDC00Sc_4.0T_r1690_I2730_LCPhys_4	Pythia 6.321	Mokka 5.03pre	LDC00Sc	4	91.2
zpole_noisr_LDC00Sc_4.0T_r1690_I2730_LCPhys_3	Pythia 6.321	Mokka 5.03pre	LDC00Sc	4	91.2
zpole_noisr_LDC00Sc_4.0T_r1690_I2730_LCPhys_2	Pythia 6.321	Mokka 5.03pre	LDC00Sc	4	91.2
zpole_noisr_LDC00Sc_4.0T_r1690_I2730_LCPhys_1	Pythia 6.321	Mokka 5.03pre	LDC00Sc	4	91.2
zpole_noisr_LDC00Sc_2.0T_r1690_I2730_LCPhys_5	Pythia 6.321	Mokka 5.03pre	LDC00Sc	2	91.2
zpole_noisr_LDC00Sc_2.0T_r1690_I2730_LCPhys_4	Pythia 6.321	Mokka 5.03pre	LDC00Sc	2	91.2
zpole_noisr_LDC00Sc_2.0T_r1690_I2730_LCPhys_3	Pythia 6.321	Mokka 5.03pre	LDC00Sc	2	91.2
zpole_noisr_LDC00Sc_2.0T_r1690_I2730_LCPhys_2	Pythia 6.321	Mokka 5.03pre	LDC00Sc	2	91.2
zpole_noisr_LDC00Sc_2.0T_r1690_I2730_LCPhys_1	Pythia 6.321	Mokka 5.03pre	LDC00Sc	2	91.2
Mokka 5.03pre			LDC00Sc	4	500
Mokka 5.03pre			LDC00Sc	4	500
Mokka 5.03pre			LDC00Sc	4	500
Mokka 5.03pre			LDC00Sc	4	500

Summary & Outlook

- a fairly complete OO-software framework exists for the LDC study based on Mokka, Marlin, LCIO, LCCD and GEAR
- can already be used for detector concept study !
- -> started Monte Carlo production on the grid

details @ software portal:
<http://ilcsoft.desy.de/>

To Do:

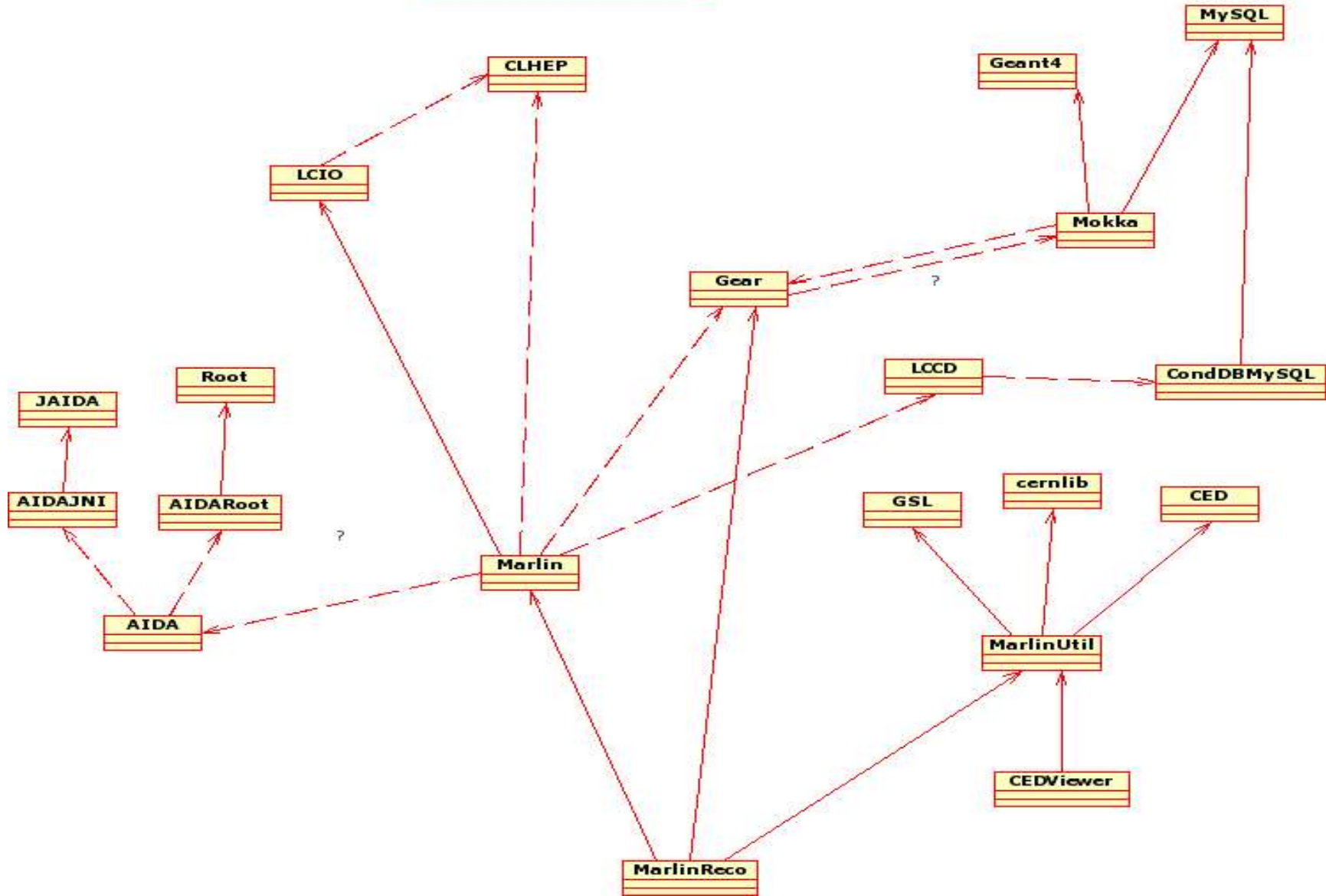
- investigate interoperability with other frameworks (ongoing !)
- apply software to other concepts (ongoing !)
- combine with other software developments (Magic, DigiSim,...)
- improve software ...

Please consider using common software tools
for your ILC study and provide feedback
and contribute to the effort !

Backup slides ...

software package dependencies

ILC-Software Packages



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

Marlin features

- core processors

- **AIDAProcessor**

- for easy creation of histograms, clouds, ntuples

- **OutputProcessor**

- writes current event or subset thereof

- **MyProcessor**

- simple example – serves as template for user code

- **ConditionsProcessor** **New**

- read conditions transparently with LCCD

- **DataSourceProcessor** **New**

- read non LCIO input, e.g. StdHepReader

- **SimpleFastMCProcessor** **New**

- fast smearing Monte Carlo
- needs testing

- fully configurable through steering files:

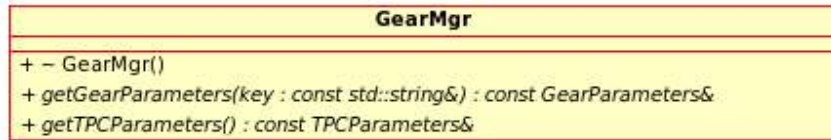
- program flow
- input parameters
- processor based and global

- self-documenting:

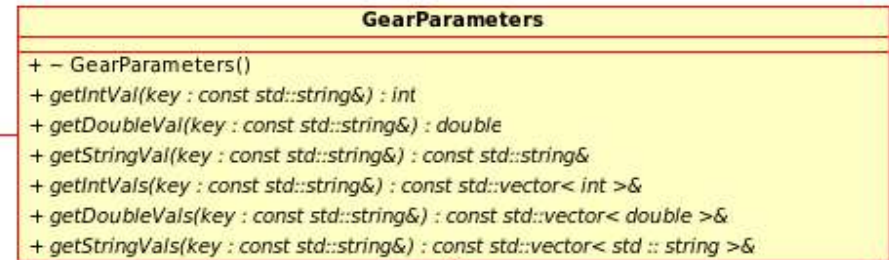
- MyApplication -l
will print all available processors with their parameters and example/default values

GEAR – TPC description

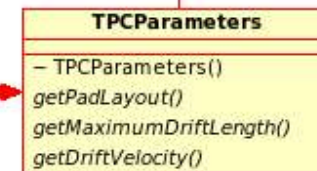
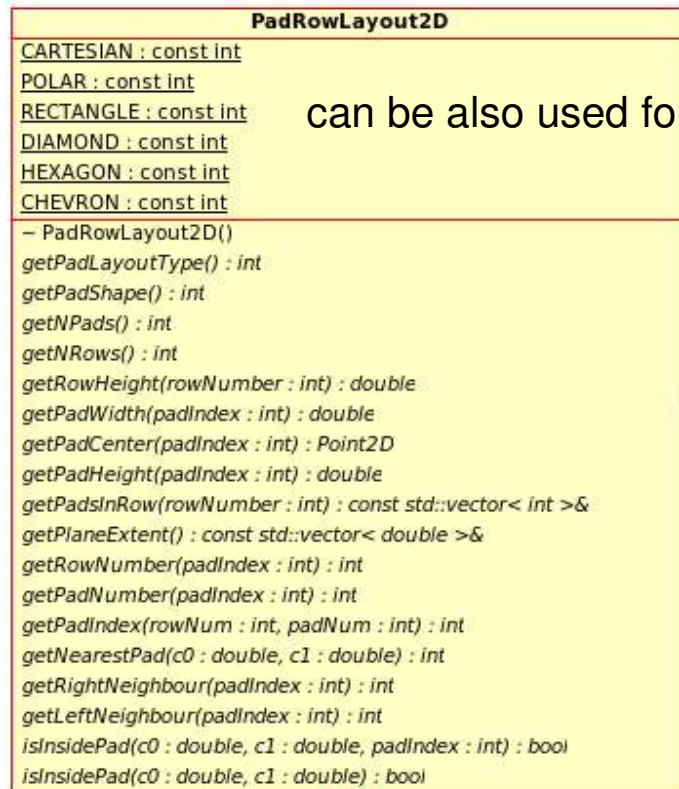
holds all subdetector classes



named parameters for additional attributes

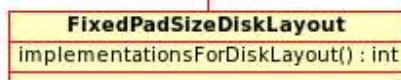


can be also used for FTD, CaloEndcap,...



TPC specific parameters

based on discussion with
TPC experts at LCWS 2005



implementation for disk with pad rings

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

Marlin Processor

- provides main **user callbacks**
- has **own set of input parameters**
 - int, float, string (single and arrays)
 - parameter description
- naturally modularizes the application
- **order of processors is defined via steering file:**
 - easy to exchange one or several modules w/o recompiling
 - can run the same processor with different parameter set in one job
- **processor task can be as simple as creating one histogram or as complex as track finding and fitting in the central tracker**

```
marlin::Processor  
init()  
processRunHeader(LCRunHeader* run)  
processEvent( LCEvent* evt)  
check( LCEvent* evt)  
end()
```

```
UserProcessor  
processEvent( LCEvent* evt){  
    // your code goes here...  
}
```



LCIO status [v01-05]

- changed return values of E, P, m to double for MCParticle and ReconstructedParticle
 - stored are still floats
 - requires some trivial changes (float->double) in code where indicated by the compiler !
- new template UTIL::LCTypedVector<T> for creating std::vector<LCObjectType> from LCCollections
 - allows to use iterators and STL algorithms, e.g. std::for_each()
- added first implementations of generic tracker raw data classes (TPC, VTX, SiliconStrip,...)
 - TrackerRawData, TrackerData, TrackerPulse
- files are downward compatible with LCIO 1.4
- bug fixes
- see \$LCIO/doc/versions.readme for more