

# ilcsoft

## introduction



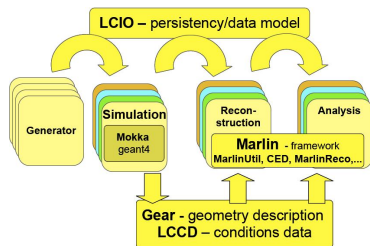
Christoph Rosemann

DESY

EUTelescope Workshop  
DESY Hamburg , March 25 2013

## Core

- Mokka - Geant4 simulation
- LCIO - EDM/persistency
- Marlin - application framework
- Gear - geometry toolkit
- LCCD - conditions data
- CED - event display



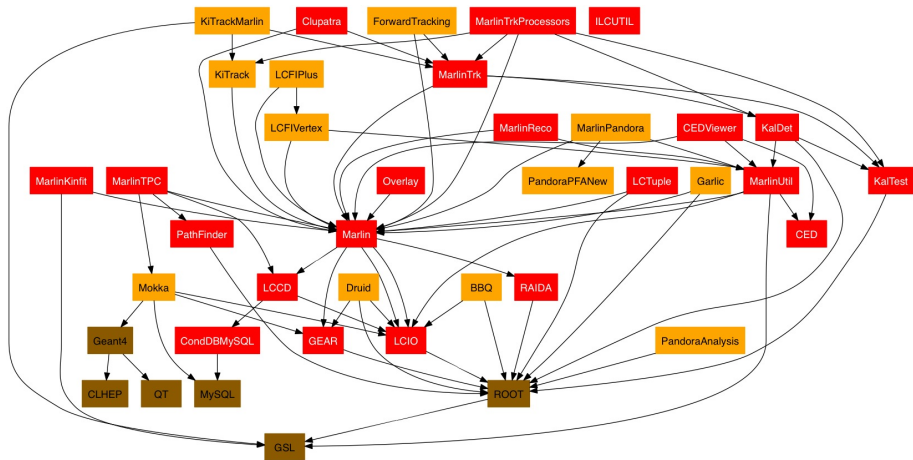
## Reconstruction packages

- MarlinReco
- MarlinTrk, Clupatra, FwdTracking,...
- MarlinKinFit
- LCFIPlus, LCFIVertex
- + many more

## Framework in use for:

- ILD simulation and reconstruction
- Testbeam projects/real data: calice, lctpc, **EUTelescope**

# The ilcsoft dependency graph



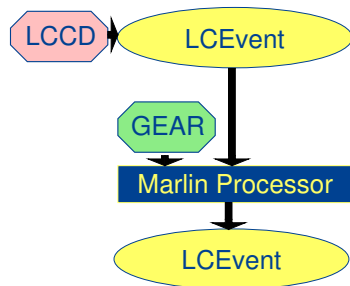
- 41 packages (General Broken Lines missing in graph)
- 6 external

# The framework

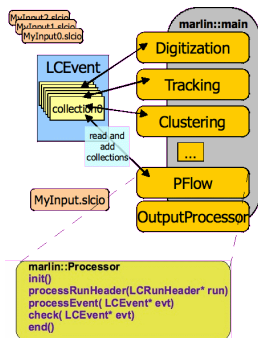
The core is the C++ application framework MARLIN:

## Modular Analysis and Reconstruction for the Linear Collider

- Interacting blocks:
  - ▶ Event data model LCIO  
*Linear Collider Input Output*
  - ▶ Geometry description GEAR  
*Geometry API for Reconstruction*
  - ▶ Extension for non-event data LCCD  
*Linear Collider Conditions Data*
  - ▶ Processor code
- Linear, sequential processing



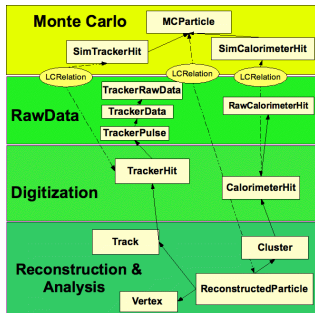
- Modular C++ application framework
- **LCIO** as transient data model
- XML configuration with local/global parameters
- Self documenting (steering parameters and defaults)
- For an example try running:  
Marlin -x | less
- Consistency check event data flow



# LCIO

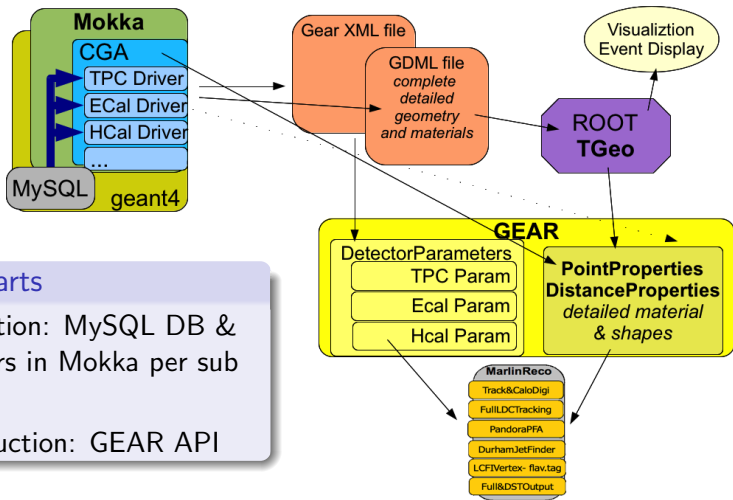
## Linear Collider Input Output

- Hierarchical **event data model**
- Implemented in C++ and Java
  - ▶ Interfaces to C and Fortran
- Common (non-ROOT) **persistency**
- Optional ROOT dictionary
- See: <http://lcio.desy.de>
- SLAC & DESY started project in 2003 (!)



- Anything you want to store must be written in LCIO (!)
- Anything you want to pass between processors is LCIO
- E.g. eudaq writes LCIO
- User defined classes: via the `LCGeneric Object`

# Detector Geometry Description in ilcsoft



## Two distinct parts

- ILD simulation: MySQL DB & C++ drivers in Mokka per sub detector
- In reconstruction: GEAR API

- Key idea: different views in reconstruction/analysis and (full) simulation
  - ▶ material voxels in space vs. sensitive detector layers (modules, sensors, cells)
- → API for special sub detector types providing the specific parameters e.g. barrel/end cap calorimeter, TPC, planar tracking detector
- → Also interface for detailed geometry & material properties:
  - ▶ material properties at any point: material, density,  $X_0$ ,  $\lambda$ ,...
  - ▶ material and field properties between two points: B field,  $X_0$ ,  $\lambda$

The model and its implementation are (slowly) changing:  
**DD4Hep**: Detector Description for High Energy Physics



# Versions and reference installations

## Current releases

- **Production v01-16-02** (December 2012)  
**Developer v01-17** – including General Broken Lines !(today)
- Supported/target platform: SL5 64bit (and SL6)

## Reference installation in afs

path: /afs/desy.de/project/ilcsoft/sw/\$OS/\$VERSION, e.g.

- /afs/desy.de/project/ilcsoft/sw/x86\_64\_gcc41\_sl5/v01-17/
- /afs/desy.de/project/ilcsoft/sw/x86\_64\_gcc44\_sl6/v01-17/

## Try to use these

- Initialize with, e.g.  
`/afs/desy.de/project/ilcsoft/sw/x86_64_gcc41_sl5/v01-17/init_ilcsoft.sh`
- Run your job  
`Marlin mySteeringFile.xml`

# Local installation

## Create your own installation with ilcinstall

- <https://svnsrv.desy.de/viewvc/ilctools/ilcinstall/tags/>
- Adjust paths and requirements in `release-versions.py`
- Run the installation script `ilcsoft-install $CONFIG`

## Two installation methods for complete install (after download)

- Start from scratch:  
`ilcsoft-install releases/$VERSION/release-scratch.cfg`
- Separate fast from slow changing content:
  - 1 `ilcsoft-install releases/$VERSION/release-base.cfg`
  - 2 `ilcsoft-install releases/$VERSION/release-ilcsoft.cfg`

You don't need the full installation!

Change config file to a minimal set

# Summary

- ilcsoft is a full, modular software framework with the key components Marlin, LCIO, GEAR and LCCD
- Used in many different context, both in parts and as a whole
- The ilcsoft team provides a world-readable reference installation in DESY afs
- The ilcinstall tool enables to create local installations

## Outlook

- The geometry description is changing towards DD4hep
- Examples can be found stepping through `ilcsoft.desy.de` and the respective sub links