



EUDET Software Summary

Parallel & Common Software Issues

Frank Gaede
DESY

EUDET Annual Meeting
LLR-Paris October 8-10, 2007

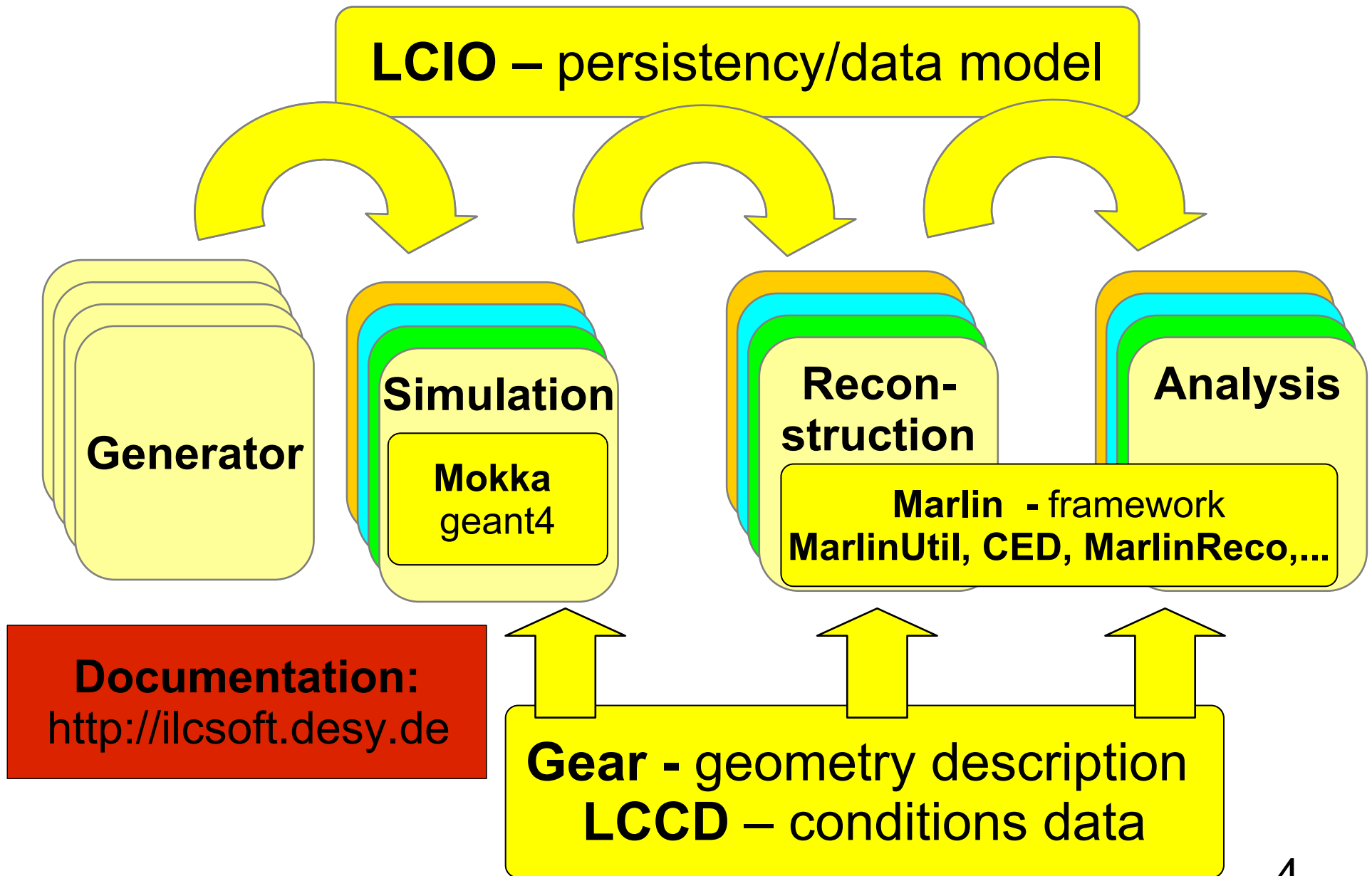
Outline

- software framework
 - status and overview
 - new build&install tools
- JRA1 – EUTelescope
- JRA2 – MarlinTPC
- JRA3 – Calice-Soft
- event display discussion

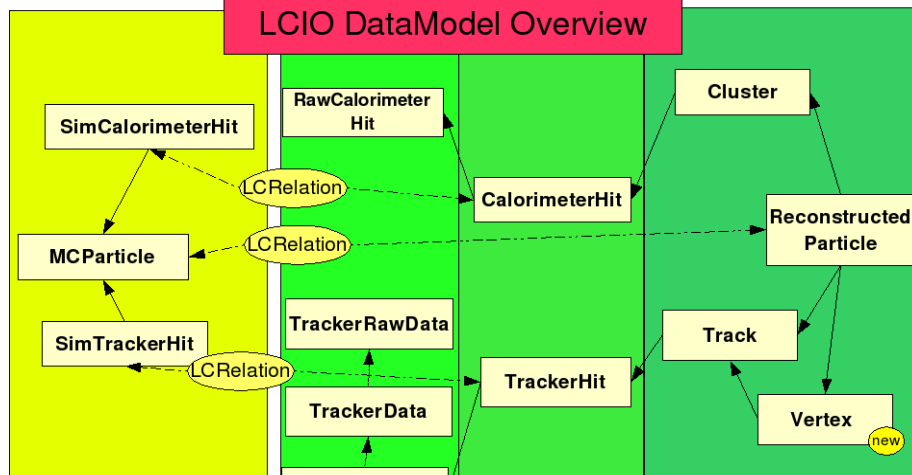
tasks COMP + ANALYS

- COMP: setup of a high performance dedicated **computer cluster** for the common data analysis and simulation using **grid technology**
- **done – grid resources are there and used !**
- ANALYS: development of a **common data analysis and simulation infrastructure**
 - development of a **software framework** for **simulation, analysis and comparison of test beam experiments**
 - -> have “version 1.0” of framework after 18 months
 - **embedded into existing GRID infrastructure**
- **strategy**
 - the test beam software effort is tightly integrated with the **overall common ILC/LDC software effort !**
 - benefit from synergies where possible
 - **same for grid: integrate with common ILC grid activities**

EUDET/LDC SW-framework



LCIO: persistency & event data model



- DESY SLAC joined project
- Java, C++ and f77 API
- now standard for
- ILC persistency & datamodel
- used in
 - all detector concept studies
 - test beam activities

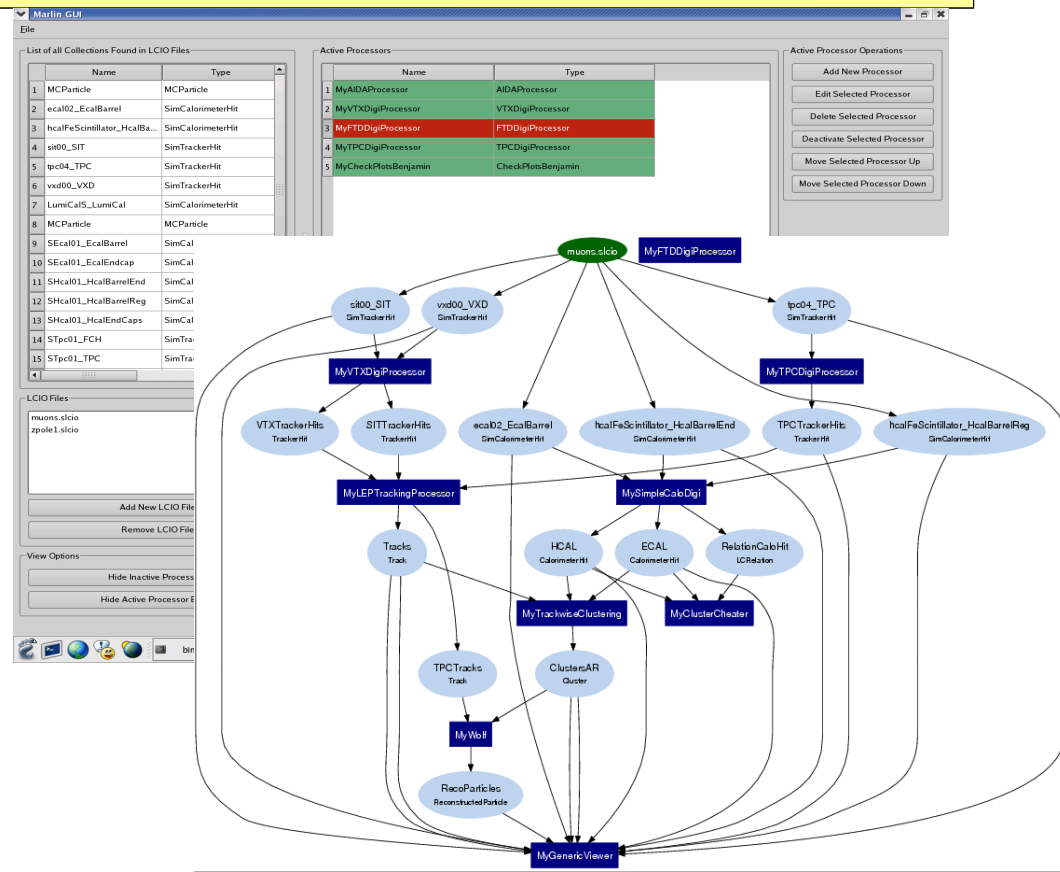
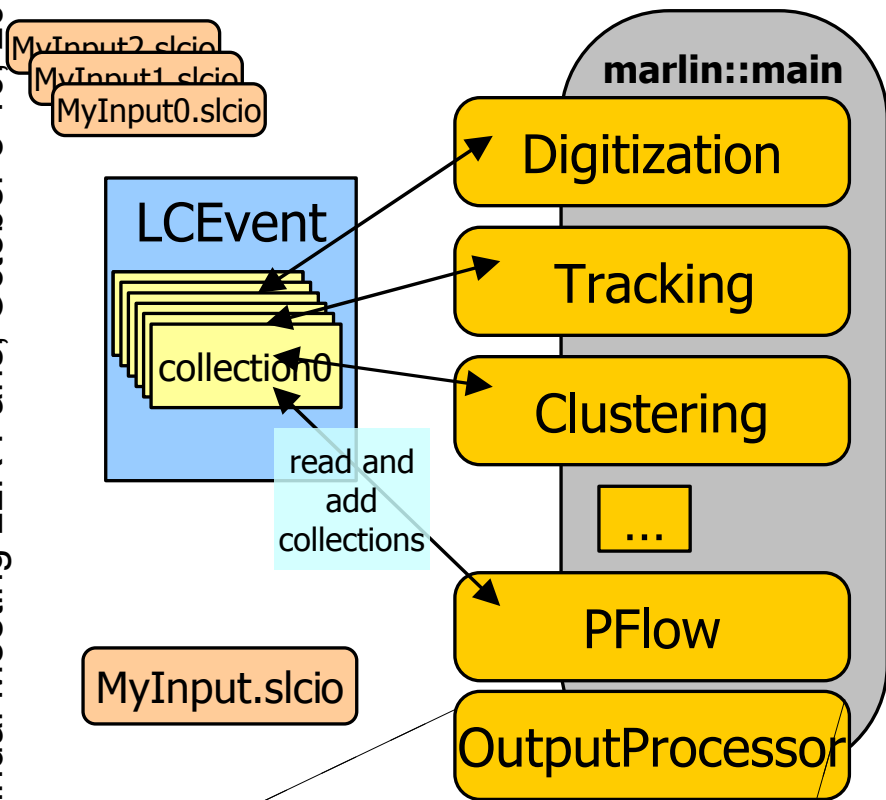
LCIO runtime extensions (C++)

- long pending user request:
 - attach user objects to LCOBJECTS
 - fast and easy creation of links (relations) between various LCOBJECT subtypes, eg. TrackerHits and Track
- features
 - extension of the object with arbitrary (even non-LCOBJECT) classes
 - extension of single objects or vectors, lists of objects
 - optionally ownership is taken for extension objects (memory management)
 - bidirectional relations between LCOBJECTS
 - one to one
 - one to many
 - many to many

to be used in reconstruction and analysis algorithms
- no persistency

- user requests:
- direct access
 - user defined objects
 - improved performance

Marlin – core application framework



```

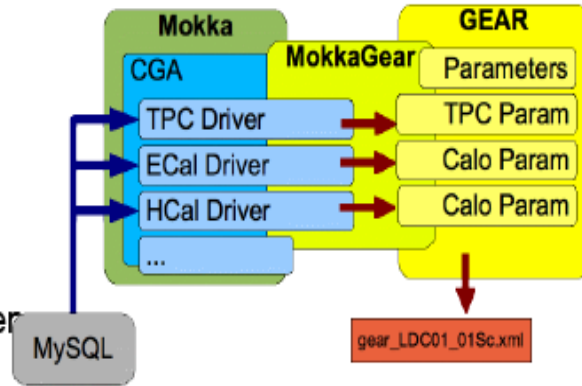
marlin::Processor
init()
processRunHeader(LCRunHeader* run)
processEvent( LCEvent* evt)
check( LCEvent* evt)
end()
    
```

- Marlin fully functional since 2005
- improved user/developer convenience:
 - MarlinGUI, flow charts, log-messages
 - plugins and new build mechanism
 - event overlay

geometry and conditions

GEometry API for R Reconstruction

- 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

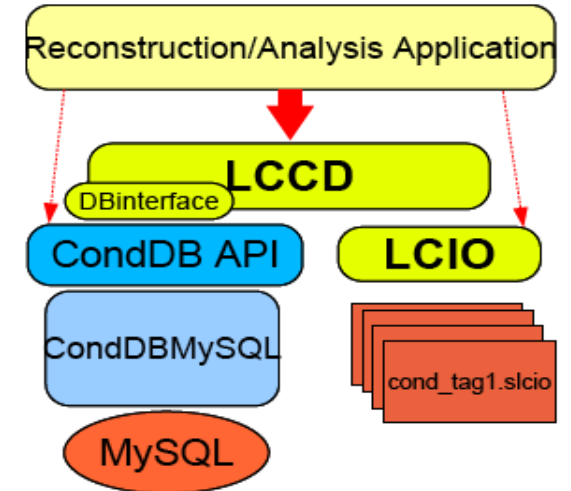


MokkaGear

- enforce only one source of geometry: the simulation program creates the geometry xml files used in reconstruction

(recently improved by K.Harder et al)

L Linear Collider Conditions Data Toolkit



LCCD is used for the conditions data of the ongoing ILC testbeam studies

software framework's base functionality complete:
-> milestone "version 1.0" after 18 month reached !

build & install tools I

Software Installation and Releases

Jan Engels
DESY

```
### RAIDA.cfg #####  
# configuration file for installing RAIDA  
# into "/data/ilcsoft/RAIDA/v01-03"  
#####  
ilcsoft = ILCSoft("/data/ilcsoft")  
  
ilcsoft.useCMake = True  
  
# install RAIDA v01-03  
ilcsoft.install( RAIDA( "v01-03" ) )  
  
# example for setting cmake build variables  
ilcsoft.module( "RAIDA" ).envcmake["BUILD_  
  
# link ROOT  
ilcsoft.link( ROOT( "/afs/desy.de/group/it/ilcso  
  
# CMake Modules  
ilcsoft.install( CMakeModules( "v01-02" ) )  
  
# End of configuration file
```

- start ilcinstall script
- go to lunch
- run ilcsoftware

3-10, 2007

ILCInstall

- Python script for installing LDC Software framework
 - LCIO, GEAR, LCCD, Marlin + modules ...
- Support for external packages
 - CLHEP, GSL, CERNLIB, QT, ...
- Automatic checking of **dependencies**
- Install whole framework without user-intervention
- **Fully configurable** (via configuration file)
 - Versions
 - Download related stuff
 - **Use/Link/Install** packages
 - Dynamic assignment of environment/cmake variables

ILCInstall:

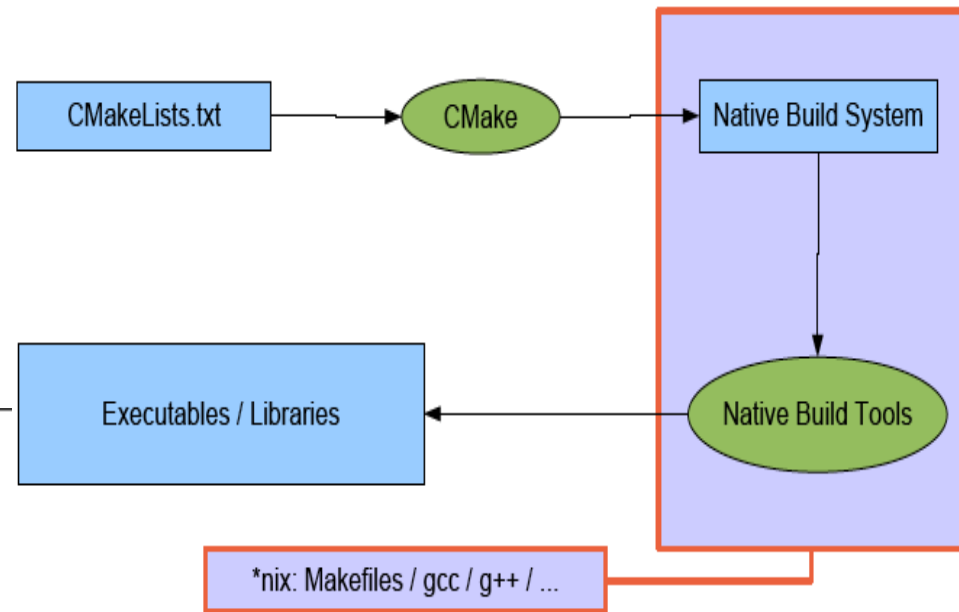
- Current version: **v01-02-01**
- Currently Supported modules:
 - LCIO, GEAR, LCCD, RAIDA, Marlin, MarlinUtil, MarlinReco, CED, CEDViewer, PandoraPFA, LCFIVertex, **SiliconDigi**, **Overlay**, **Eutelescope**, **Mokka**, **CLHEP/HepPDT**, GSL, CERNLIB, CondDBMySQL, **QT**, CMake, CmakeModules, **ROOT**, **Geant4**, **AIDAJNI**, **JAIDA**, Java, MySQL

Frank Gaede, EU

build & install tools II

CMake:

- Cross-platform build-system manager
- Generates native build environments
 - UNIX/Linux -> Makefiles
 - Windows -> VS Projects/Workspaces
 - Apple -> Xcode



CMake:

- CmakeModules - Current version: v01-02
- Currently Supported:
 - [LCIO](#), [GEAR](#), [LCCD](#), [RAIDA](#), [Marlin](#), [Overlay](#), [MarlinUtil](#), [MarlinReco](#), [CED](#), [CEDViewer](#), [PandoraPFA](#), [LCFIVertex](#), [SiliconDigi](#), [Eutelescope](#), [CLHEP/HepPDT](#), [CERNLIB](#), [CondDBMySQL](#), [GSL](#), [ROOT](#), [Java](#), [AIDAJNI/JAIDA](#)

- build process more coherent and easier to maintain
 - > for core & package/tool developers
- build shared libraries (plugin mechanism)
- used in afs reference installations:
[/afs/desy.de/group/it/ilcsoft/v01-01](https://afs.desy.de/group/it/ilcsoft/v01-01)

JRA1 – EUTelescope I

Status of EUTelescope

Antonio Bulgheroni (INFN)

Very solid starting point...

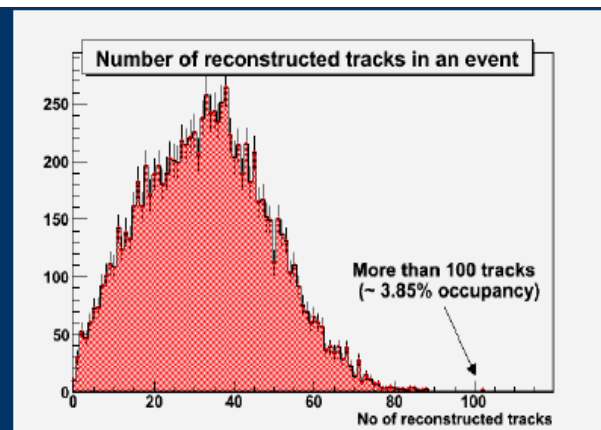
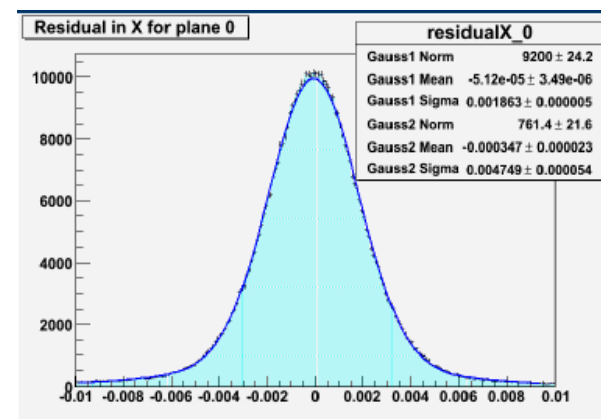
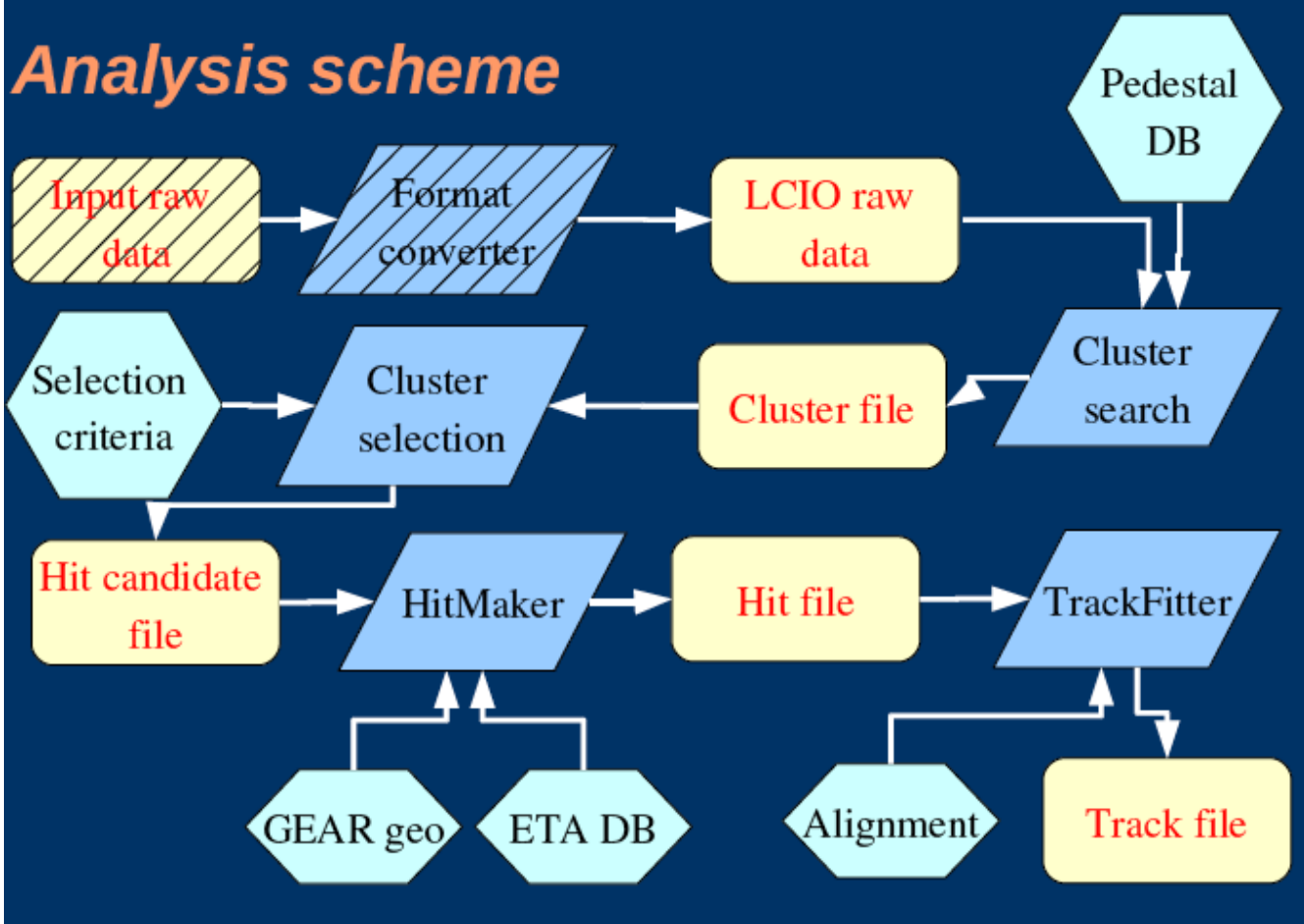
... trying not to reinvent the wheel!

- Gain as much as possible from past experience and already available and tested software tools:
 - **Single sensor analysis** → sucimaPix (INFN)
 - **Eta function correction** → MAF (IPHC - there are several other things we can and should borrow)
 - **Track fitting** → Analytical track fitting (A. F. Zarnecki taking care of MS) and straight line fitting (T. Klimkovich)
 - **Simple alignment** → Minuit based (copied and improved from other codes)
 - **Framework** → ILC Core software = Marlin + LCIO + GEAR + (R)AIDA + CED (+ LCCD).

JRA1 started to adopt common framework early this year !

JRA1 – EUTelescope II

Analysis scheme



CERN large multiplicity data taken two weeks ago

It's WORKING

- EUTelescope has been successfully used during the last three test beam periods.

It's on the GRID

- Nearly all the analysis steps are performed on the GRID reducing the processing time by at least one order of magnitude

JRA1 – EUTelescope III

Future plans

- Remove the format conversion step... getting LCIO files directly from the DAQ
 - So far used the native data format only for debug. Now time has come to test the LCIO output.
- Include Millepede in MarlinUtil package and use it within EUTelescope (P. Roloff)
- Taking part on the development of a common event display for EUDET.

Conclusion

- EUTelescope is working.
- The results obtained from the three data taking periods are proving that it is well behaving...
- A lot of work has been done and the milestone foreseen for the end of 2008 will be easily achieved.

JRA2 – MarlinTPC I

Martin Killenberg¹

Jason Abernathy², Klaus Dehmelt³, Ralf Diener³, Jim Hunt⁴,

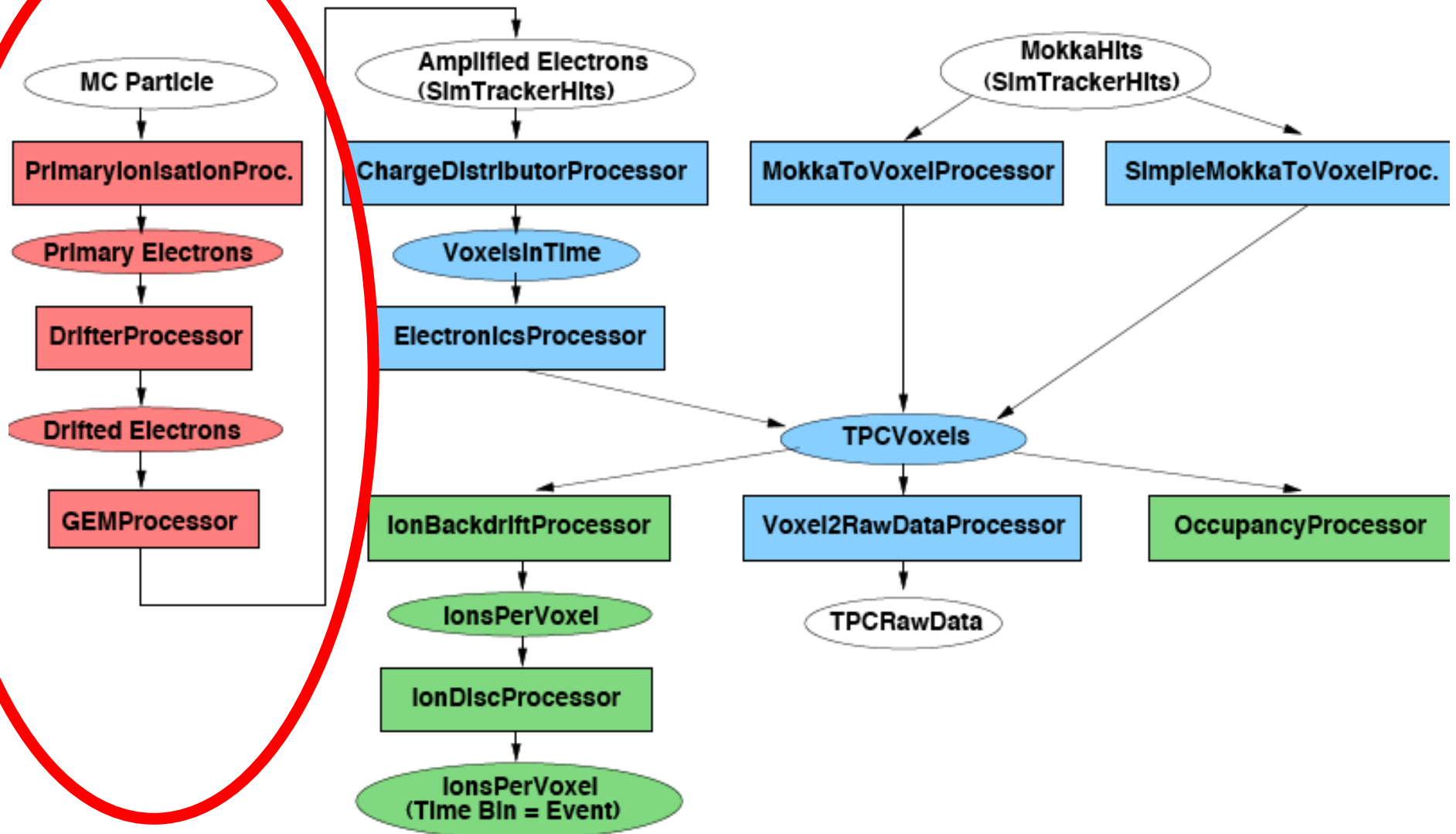
Matthias Enno Janssen³, Thorsten Krautscheid¹, Astrid Münnich⁵,

Martin Ummenhofer¹, Adrian Vogel³, Peter Wienemann¹ and Simone Zimmermann¹

- MarlinTPC is independent of the specific detector:
 - Works for prototypes and ILC detectors
(every TPC that can be described with GEAR)
 - Works for Micromegas, GEMs and anode wires
 - Works for pads and pixel based readout
 - Independent of electronics: TDCs, ADCs ...
- Take advantage of the well defined LCIO and Marlin interfaces
 - Highly modular
 - Good reusability of code
 - Only little additional code to adapt to specific detector

JRA2 – MarlinTPC II

Simulation and Digitisation

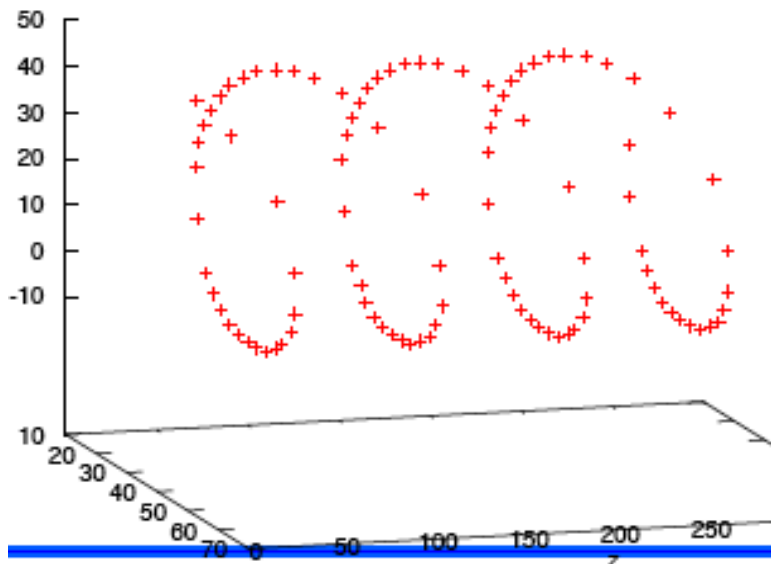
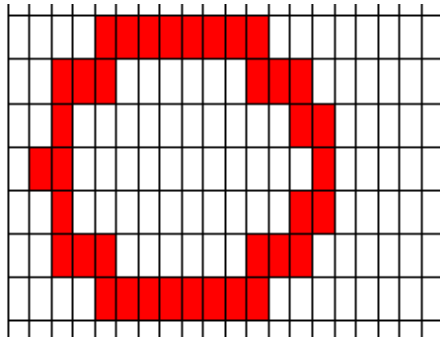


JRA2 – MarlinTPC III

Topological Hit and Track Finder

Conclusions

- Independent of trajectory, no track hypothesis
- Works in 3D

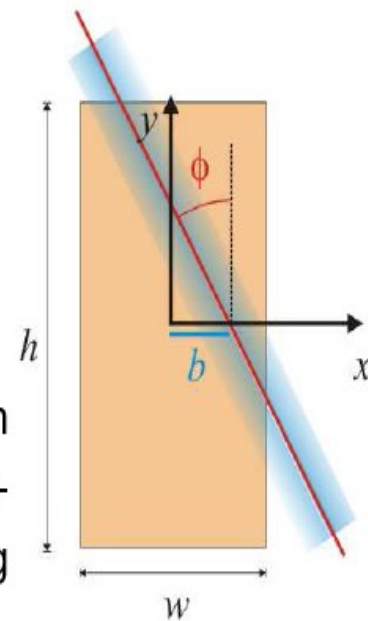


- Detailed simulation provides realistic data
- Digitisation incl. pileup is under development
- Reconstruction chain is complete now, provides basic functionality

TrackFitterLikelihoodProcessor



- The pad response can only be calculated correctly if angle of track wrt. pad row is known.
- This cannot be done on hit basis
- ⇒ Do it globally for the whole track
- Calculate likelihood of charge distribution on a single pad row for given track parameters, assuming Gaussian distribution along the track
- Sum up $\log(\text{likelihood})$ on all pad rows to get global likelihood
- Maximise the $\log(\text{likelihood})$ by varying the track parameters



JRA3 – Calice software I

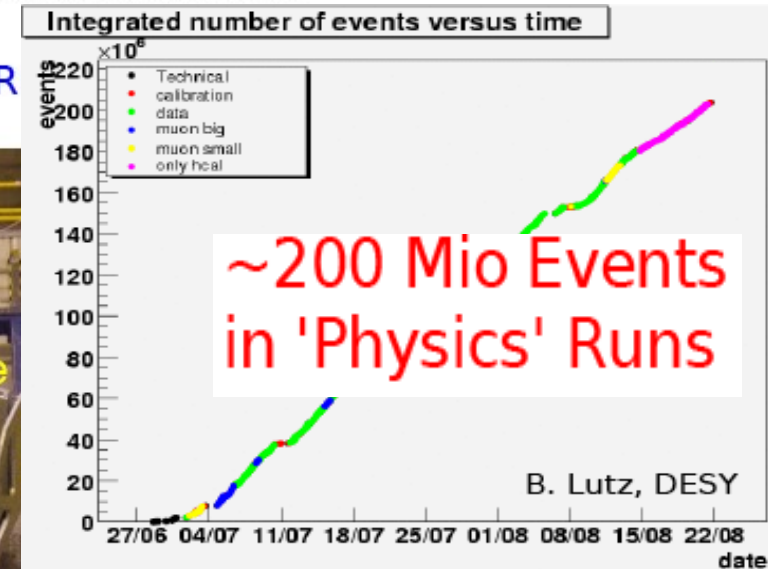
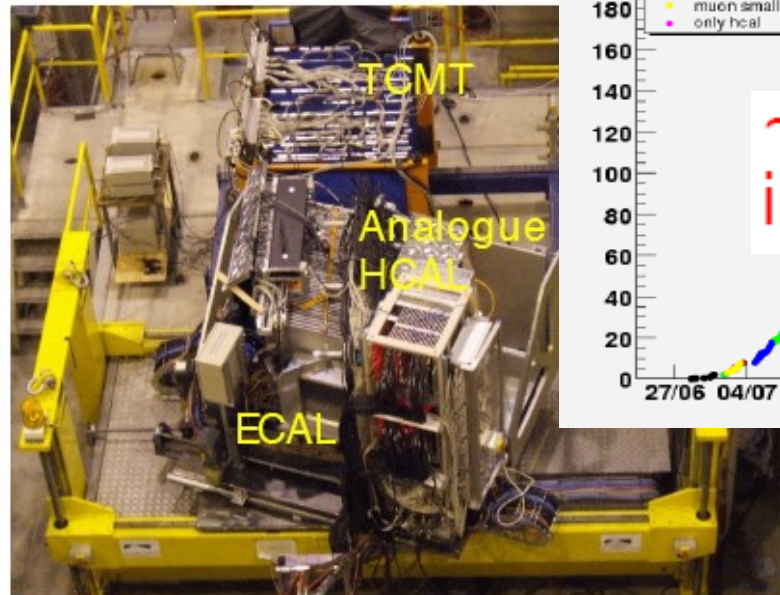
CALICE Testbeam Data Taking

CALICE collaboration is preparing/performing large scale testbeam
Data taking in Summer 2006/2007

Testbeam program poses
software/computing “
challenges”

- Data processing from Raw Data to final Clusters in a coherent way
- Handling of Conditions Data Detector Configuration Calibration, Alignment etc.
- Comparison with simulated data
'Physics' Output

Testbeam Setup at CER



O(15000) calorimeter cells
readout by Calice DAQ
No Zero Suppression

Roman Pöschl

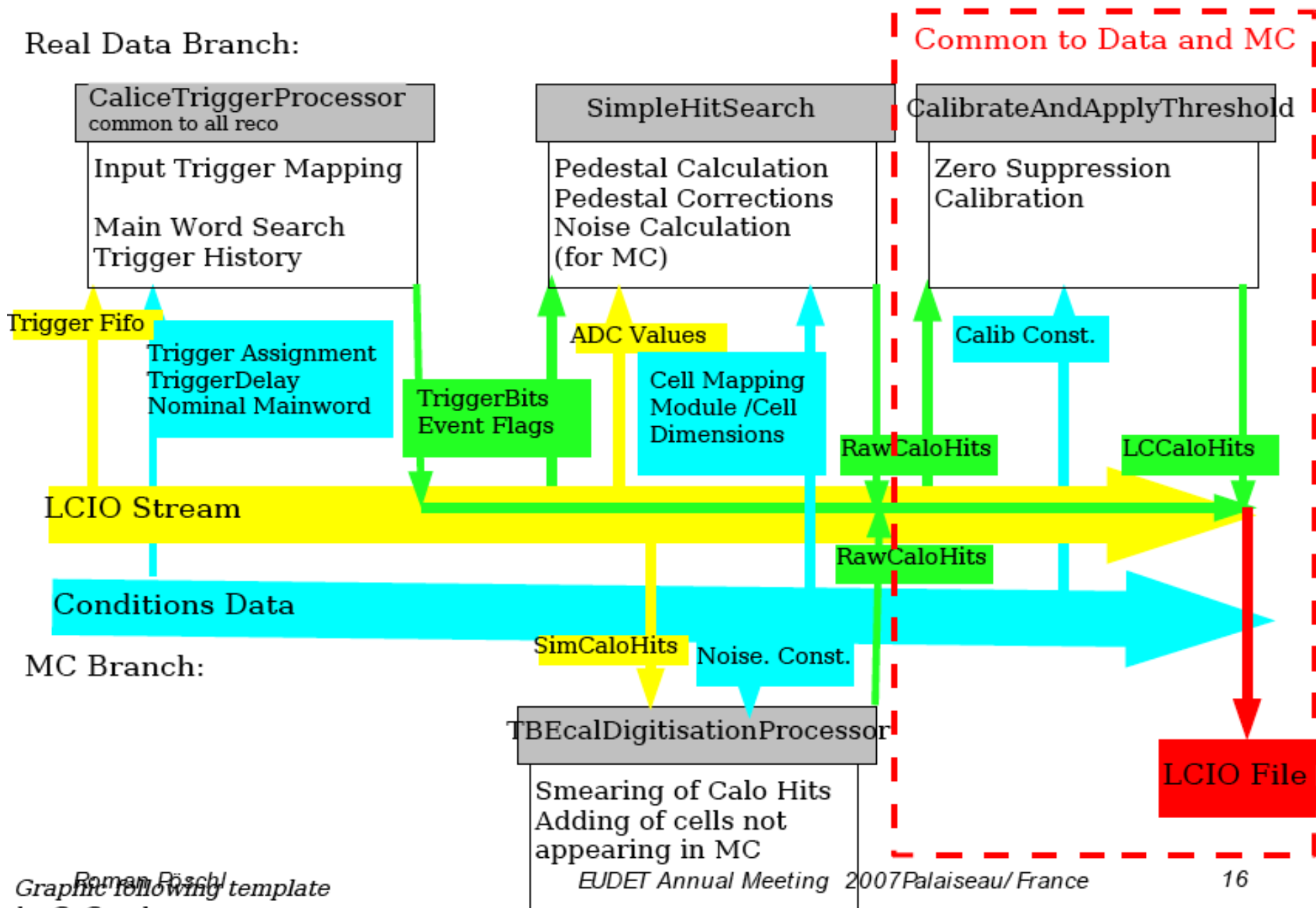
EUDET Annual Meeting 2007Palaiseau/France

2

Calice was first collaboration/group to adopt the common software framework serving as a real world “testbed” for developing and improving the framework

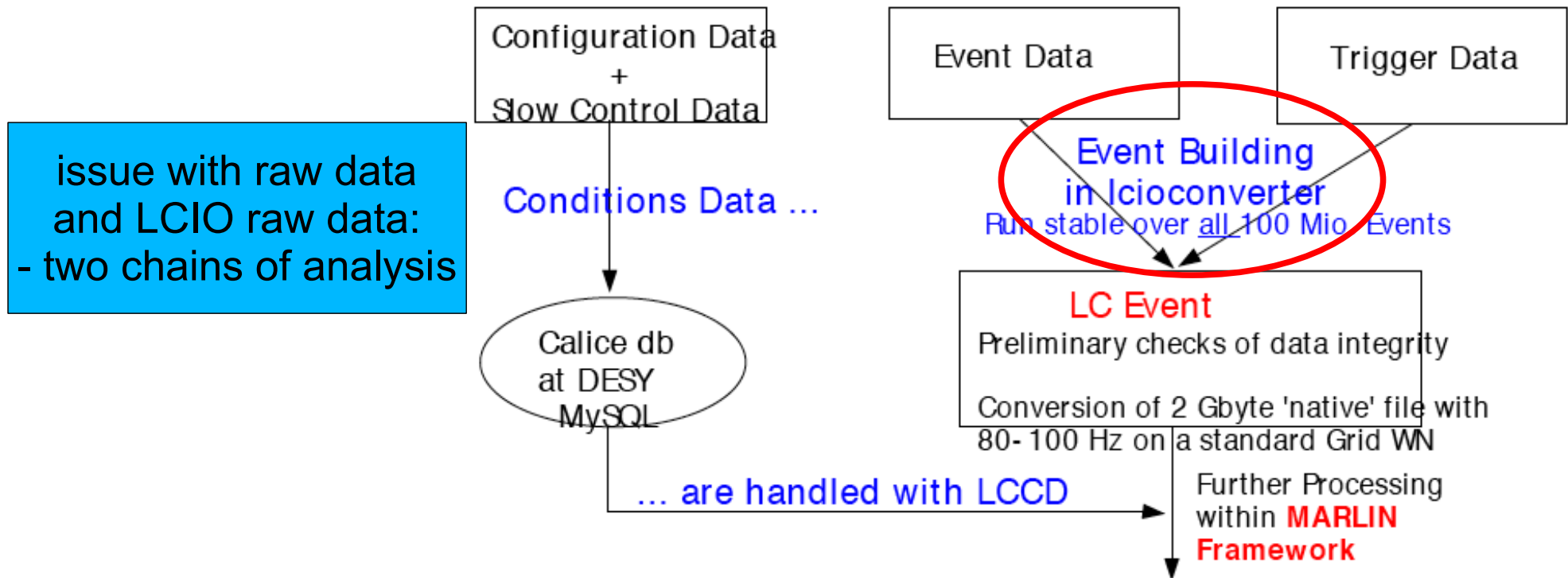
JRA3 – Calice software II

Real Data Branch:



JRA3 – Calice software III

DAQ Data Files/ Types



Summary and Outlook

- Calice uses ILC Software for processing of Testbeam Data

ILC Datataking in a (big) nutshell

Very important input for current and future developments of ILC Software
Allows for a revision of the ILC Software concepts on a 'living' beast

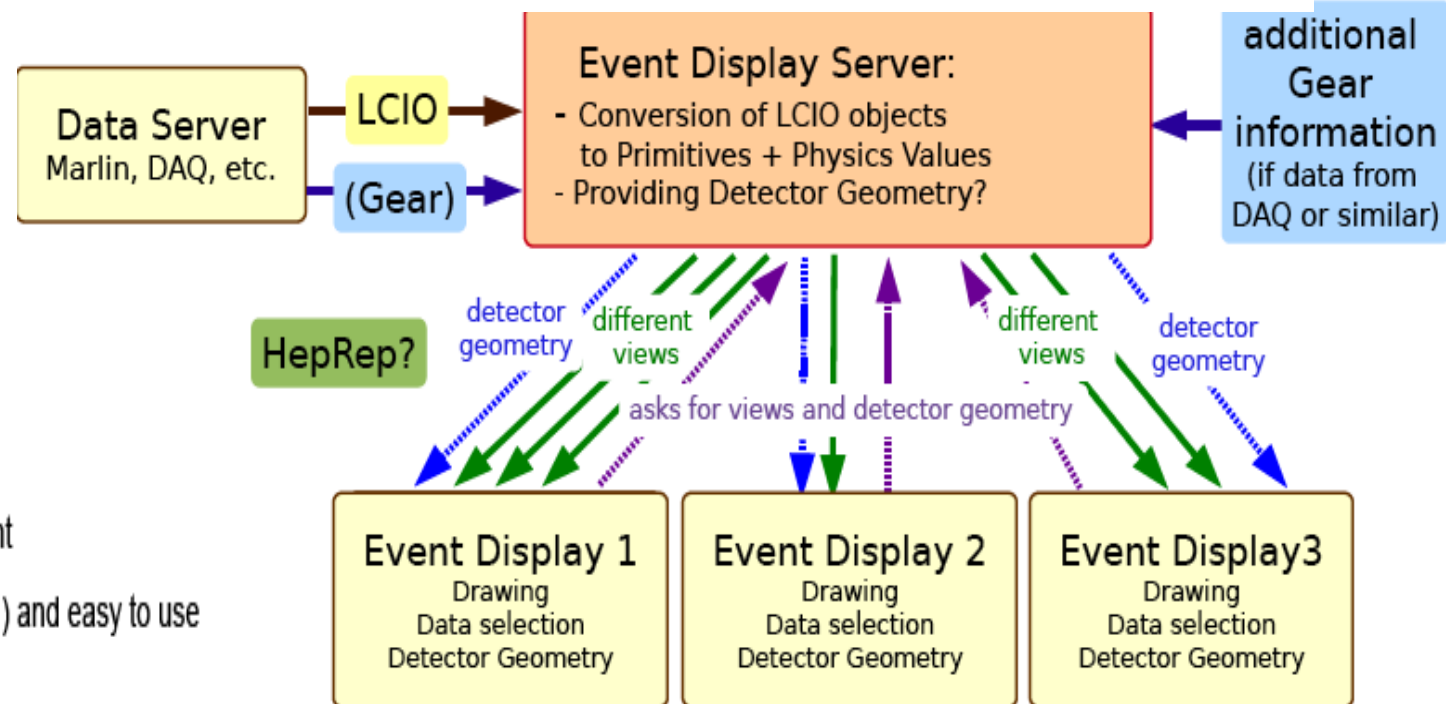
- Calice uses systematically Grid tools

- Experience with testbeam data clearly reveals the needs for a coherent concept to handle 'low level' data within ILC Software

common event display ?

- Discussion started because many 80% solutions are there but no 100%
Many groups seem to be interested
- CED
- MarlinED (based on CED) by Jason and TPC display from Jim Hunt
- CALICE display

Main Question: Where should the logic be?



WIRED:

- Version 4 included as plugin in JAS3
- Displays HepRep input
- Gets slow with detailed geometry / event
- Good functionality (picking, rotation etc.) and easy to use

Manpower could be a problem → separate tasks and distribute effort (different groups)?

common event display - thoughts

- requirements – wish list:
 - online: fast, simple displays, (histograms?)
 - offline: (fast), zooming, picking, interactive “analysis”
 - client sever
 - long term support guaranteed (it-division?)
 - need sth. now (test beams ongoing and coming up)
- ideas – plans
 - use existing tools e.g. HepRep/Wired
 - use LCIO as data foramt + sth else (root, Coin3d,....)
 - ... lots of other ideas
- need to continue discussion in forum
- **biggest question: manpower**

Summary - Outlook

- “version 1.0” of core framework exists
- build and installation tools developed
- all JRAs have software frameworks based on core framework
- **grid** is used for data processing and storage
- processing real data reveals requirements for further core software improvements
 - I/O, event data model, event display,...
 - progress limited by manpower
- **should make sure that (core) software development is integral part of possible EUDET-II**