LCIO-based Data Processing

Experiences from the test-beam pit

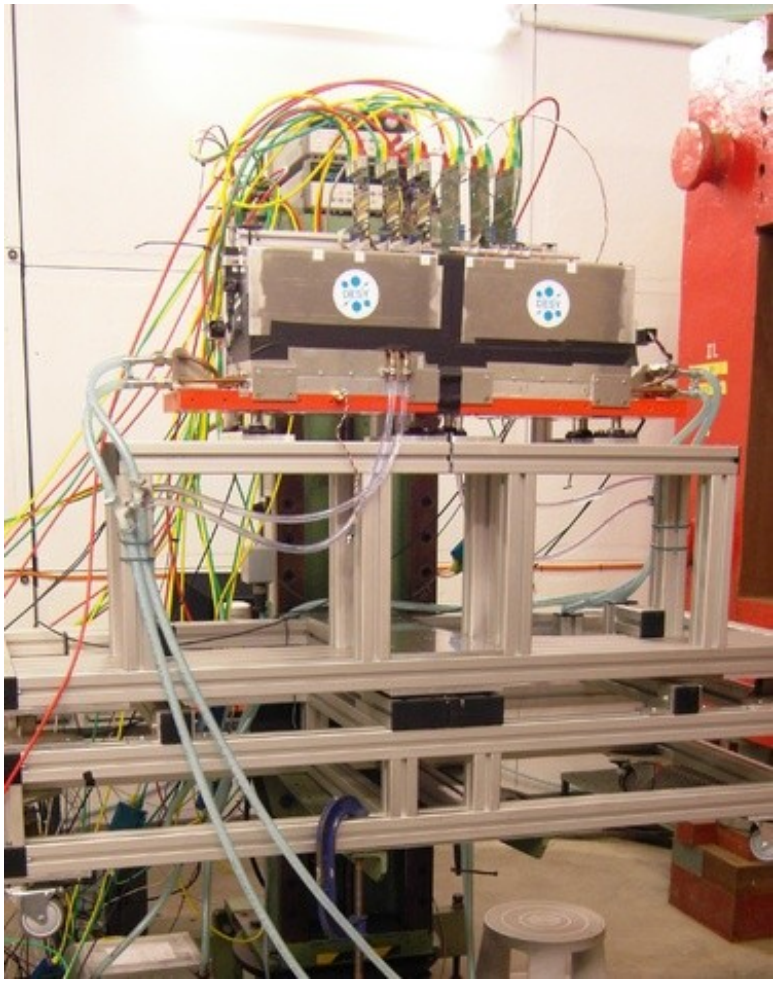
Niels Meyer, DESY

ILD Software Meeting, KEK

April 16, 2009

EU Pixel Telescope

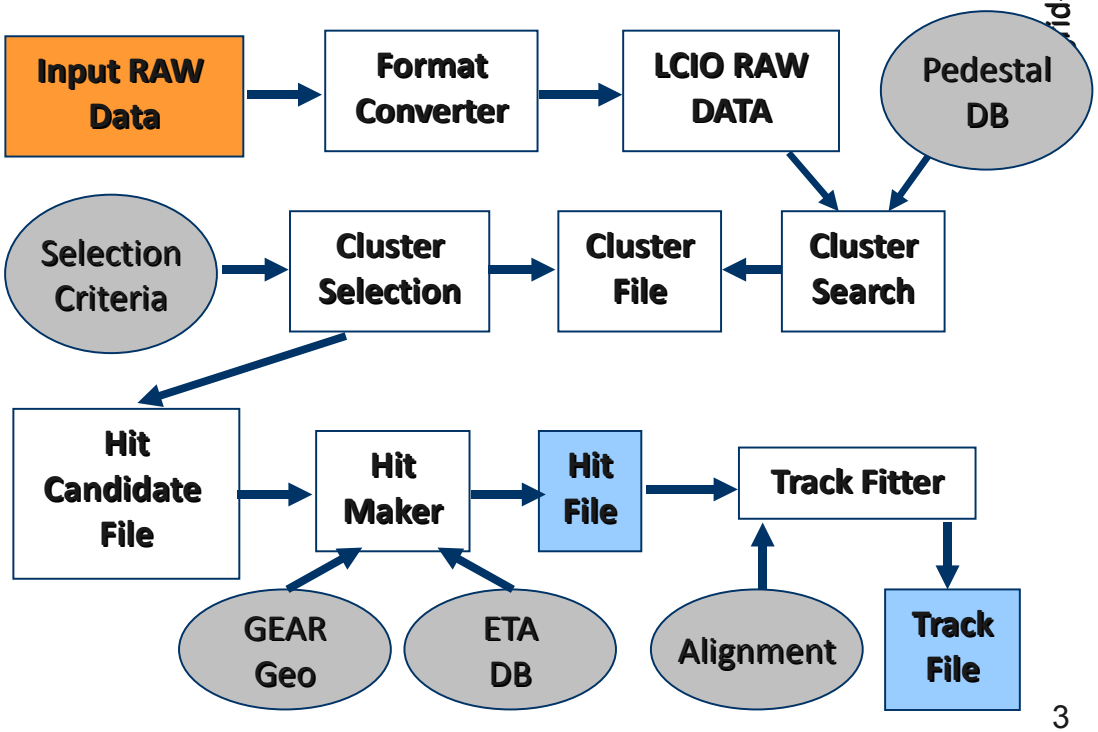
- Vertex detector prototype, position monitor for test-beam setups
- One technology, one team
- Provided to users, so aim is to have automated, easy-to-use reconstruction software
- European effort



Analysis and reconstruction software

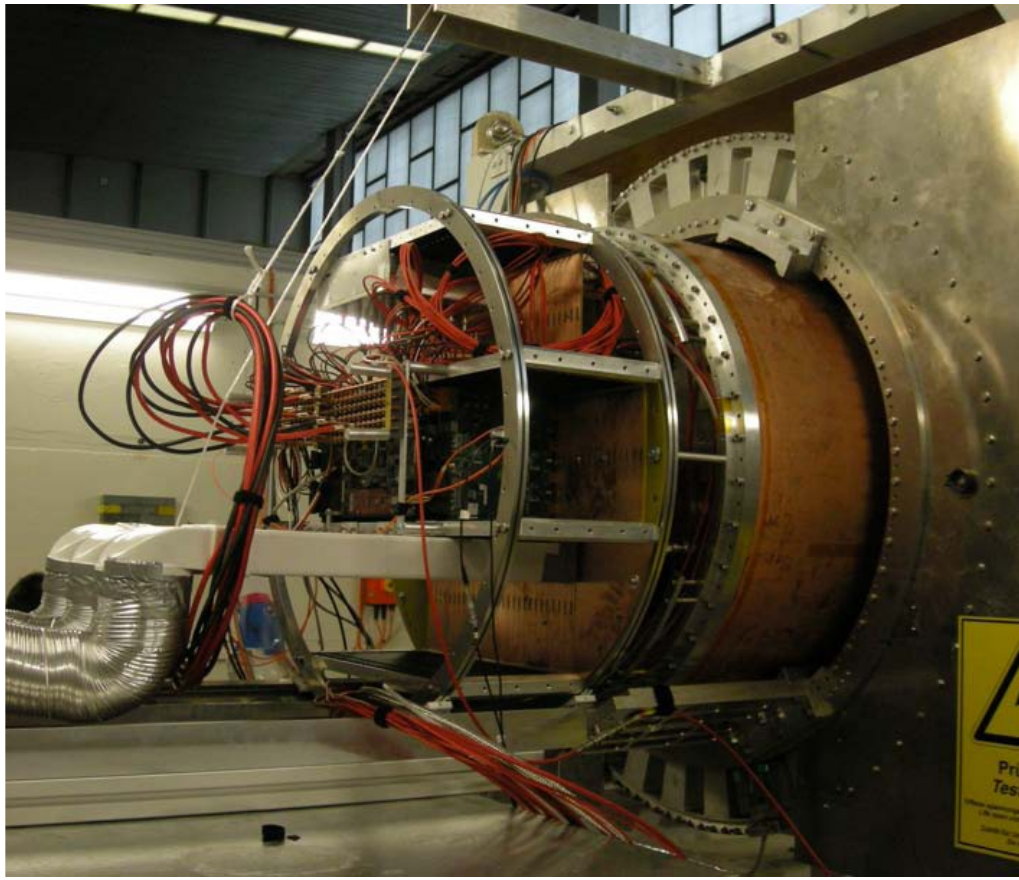
- Gain as much as possible from past experience and already available and tested software tools:
  - Single sensor analysis → **sucimaPix** (INFN)
  - Eta function correction → **MAF** (IPHC)
  - Track fitting → **Analytical track fitting** and straight line fitting
  - Alignment → **Millepede II**
  - Framework → ILC Core software = **Marlin + LCIO + GEAR + (R)AIDA + CED (+ LCCD).**

- Each module is implemented in a Marlin processor
- execute all of them together, or stop after every single step
- Advantages when debugging the system
- Can offer the user different level of information

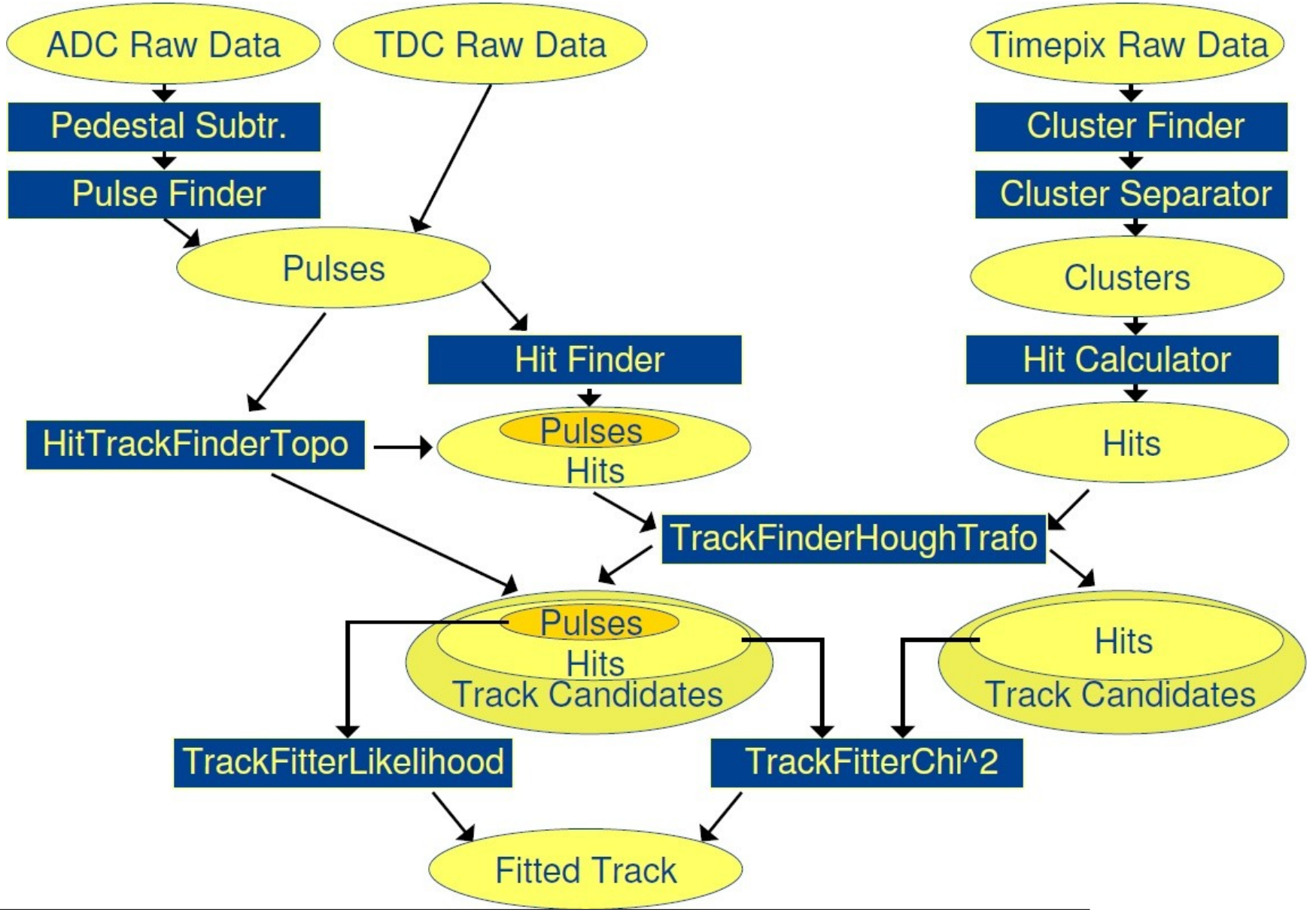


LC-TPC Large Prototype

- Main tracker prototype including magnetic field
- Back-plane for various readout technologies, collaborated effort
- Complex set of conditions information (electronic and surrounding)
- Inter-regional effort

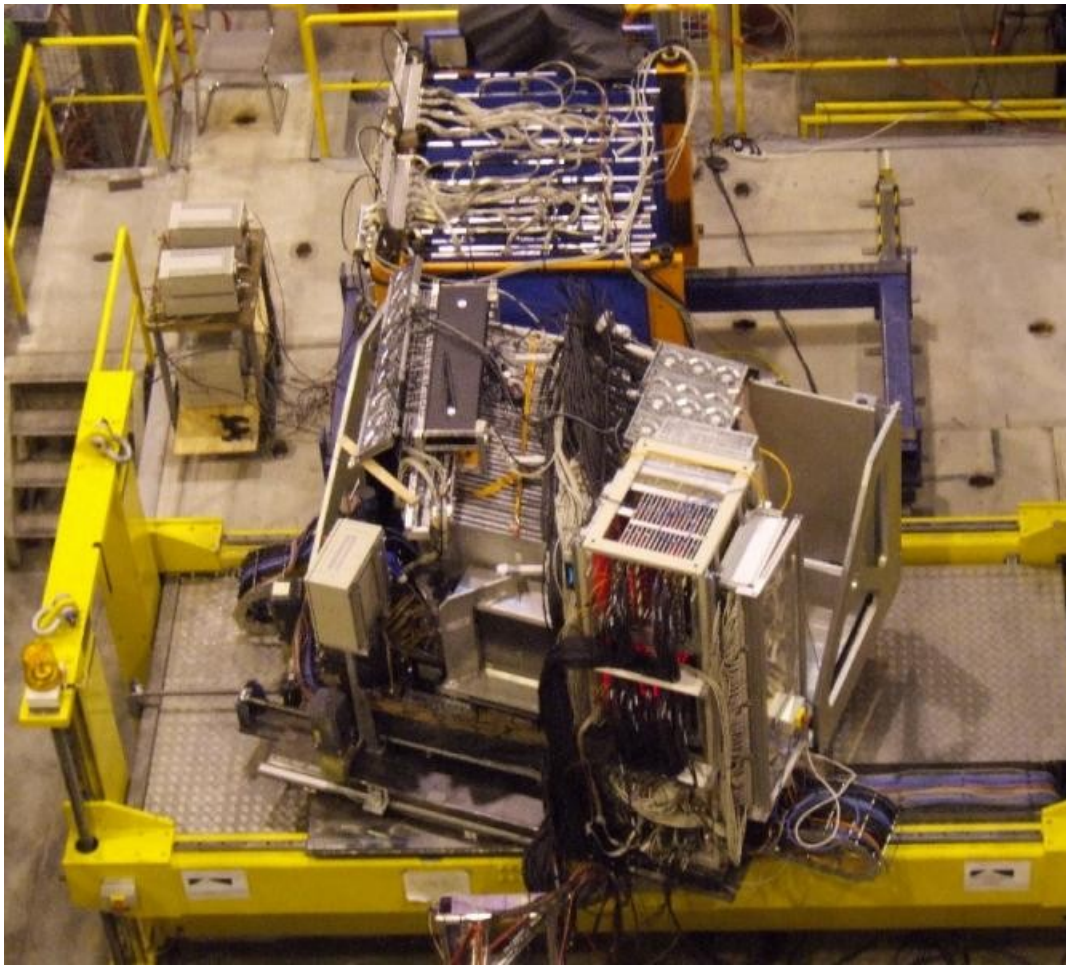


MarlinTPC

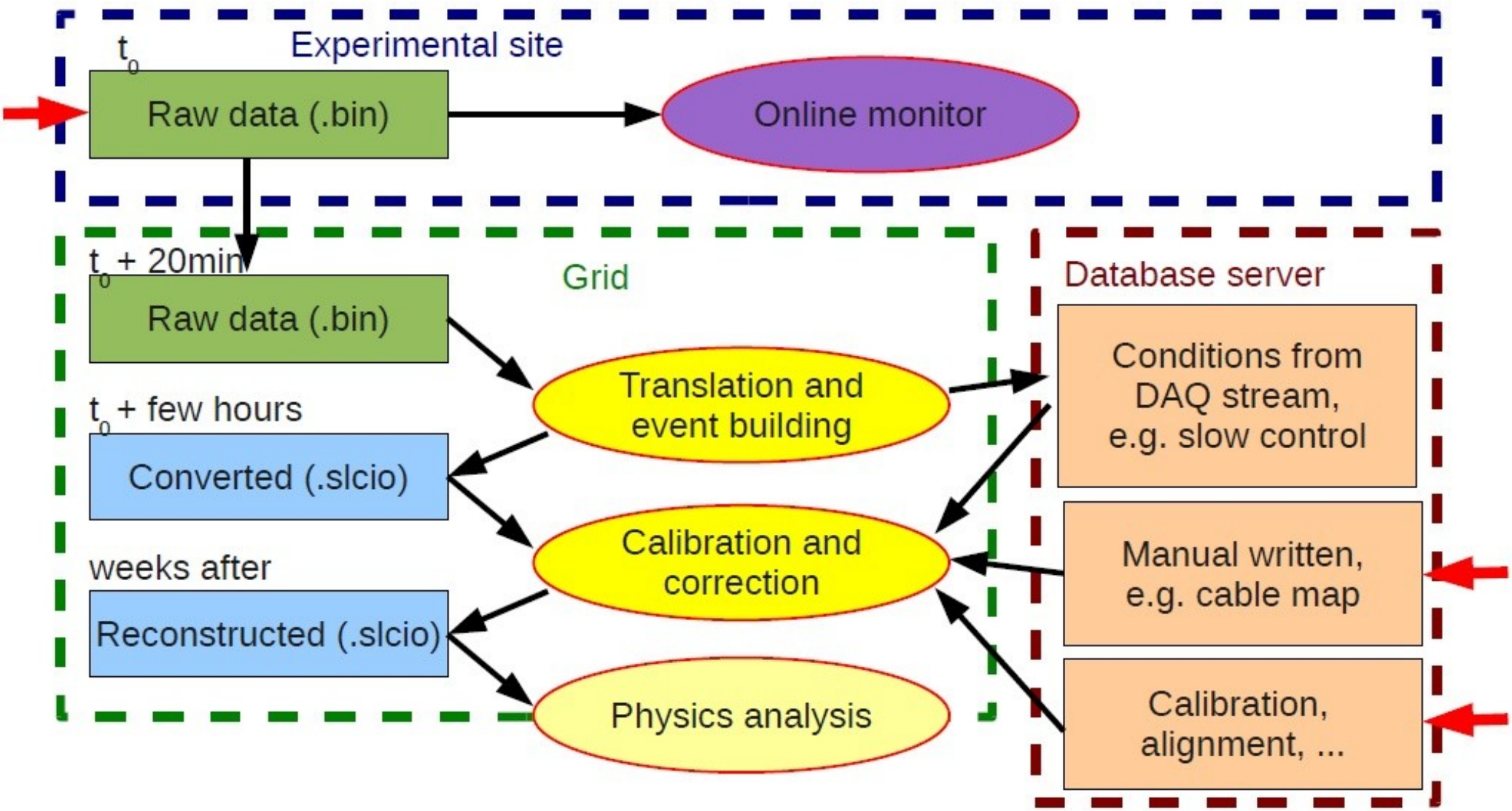

ADC Raw Data
TDC Raw Data
Timepix Raw Data
Pedestal Subtr.
Cluster Finder
Pulse Finder
Cluster Separator
Pulses
Clusters
Hit Finder
Hit Calculator
HitTrackFinderTopo
Pulses
Hits
Hits
TrackFinderHoughTrafo
Pulses
Hits
Track Candidates
Hits
Track Candidates
TrackFitterLikelihood
TrackFitterChi^2
Fitted Track

CALICE Calorimeter Setup

- Calorimeter prototypes for hadronic shower measurements
- So far four different detectors, more to come, collaborated structure
- Large data volume, large number of channels, complex conditions data, non-trivial geometry
- Inter-regional and inter-concept effort



CALICE Data Flow



Software Coordinates

- All three R&D collaborations use the LDC scheme:
  - Implementations in C++
  - LCIO as data model and for data storage
  - Marlin for data processing
  - LCCD as interface to conditions data
  - MOKKA as interface to Geant4
- EU Telescope and LC-TPC also use GEAR for geometry description, CALICE has own model
- CALICE has decided to continue using this scheme also for non-European detectors (SciECal from Japan, DHCal from North America)

Private Comments - LCIO

- Most operations do not process all informations of an event. For these cases, the current I/O implementation is sub-optimal. Does not make a difference for mass-reconstruction, but for user analysis.
- Shortcommings in integration of user-defined classes (via LCGenericObjects)
  - Original class lost after storage/re-read, bears danger of confusing different objects with identical size
  - Often hardware-related configuration- or read-out blocks, so not of general interest for full portability

Private Comments - Marlin

- Marlin provides 'loop over events' to encapsulate modulare processing steps. Also processings on individual objects could be modularized. Second loop level (over one collection) of general use?

Private Comments - LCCD

- In principle, the LCIO/Marlin *concept* is threat-save - if it was not for the change listener pattern for conditions handling. Is there a better way, e.g. by 'collection has changed'-flags?
- The only available database implementation for LCCD is based on (non-maintained) CondDBMySQL. How to provide 'code reliability' here? Who takes care of fixes, e.g. removing I/O overhead to DB server?
- Scalability: Conditions for full scale detector will be huge (current HCal calib: ~35 floats per channel). Need clever memory ⇔ CPU ⇔ I/O balancing in the future, preferably steerable.

Private Comments - Geometry

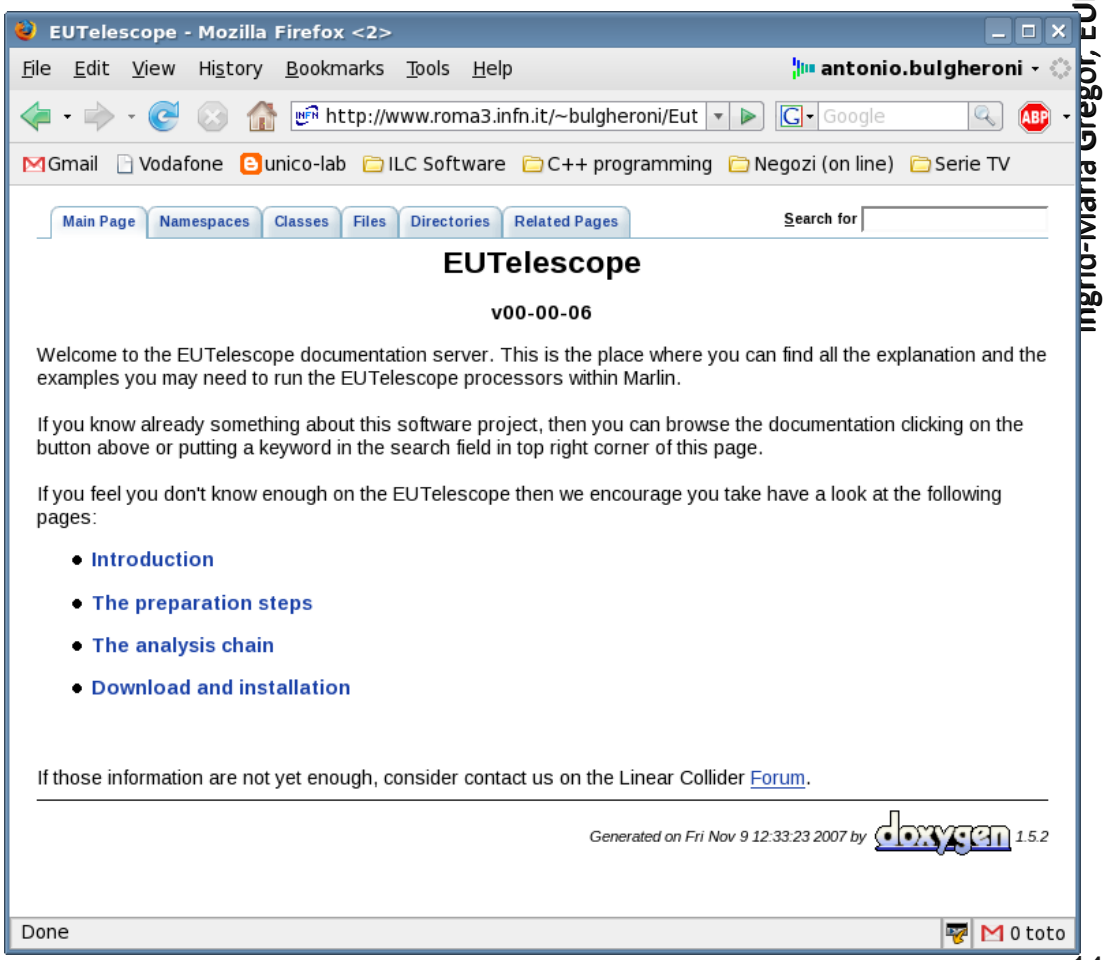
- GEAR currently simulation-driven: geometry defined in MOKKA, fed to reconstruction by XML file - this is not a useful ansatz for test beam setups (which are very flexible) and most probably also not for a full-scale detector
- Geometry data is also conditions data, some change with time (alignment), others are fixed relations (blue-prints) - natural would be LCCD-based handling
- Would want to have something that is (from the code point of view) identical in simulation and reconstruction

Backup Slides

EUTelescope

- Providing the users a set of relevant high level objects (like tracks or space points) to characterize the DUT along with histograms of important figures of merit.
- Collaborating in the development of a common software framework in view of the future International Linear Collider experiment.

- **Developed within the Software Networking activity, it is based on the official ILC framework: Marlin + LCIO**
- **EUTelescope is a set of Processors taking to care to handle the data stream from the DAQ to the reconstructed tracks**
- **Sticking to the ILC de-facto standard offers the possibility to easily use the GRID.**

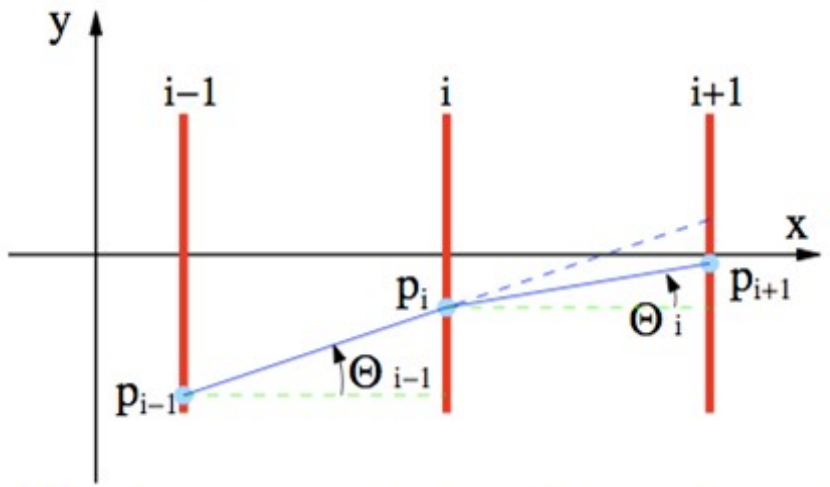


Analysis Method

Track Fitting

- Want to determine track position in each plane (including DUT), i.e. N parameters (pix i=1…N) from N-1 measured positions in telescope planes
- Use constraints on multiple scattering

Contribution of plane i to χ^2 of the fit



position measurement multiple scattering

$$\Delta\chi_i^2 = \left(\frac{y_i - p_i}{\sigma_i}\right)^2 + \left(\frac{\Theta_i - \Theta_{i-1}}{\Delta\Theta_i}\right)^2$$

where: $\Theta_i = \dfrac{p_{i+1} - p_i}{x_{i+1} - x_i}$

Both terms present for planes $i \neq 1, i_{DUT}, N$,
first term missing for DUT, second for first and last plane

χ^2 minimum can be found by solving the matrix equation.

- **As a by product we get also an expected error on the position reconstructed at DUT.**