ILC Software in Future (Testbeam) Data

- From Raw to Physics Data

Roman Pöschl
LAL Orsay

Linear Collider Workshop LCWS07 DESY/Hamburg Germany May/June 2007

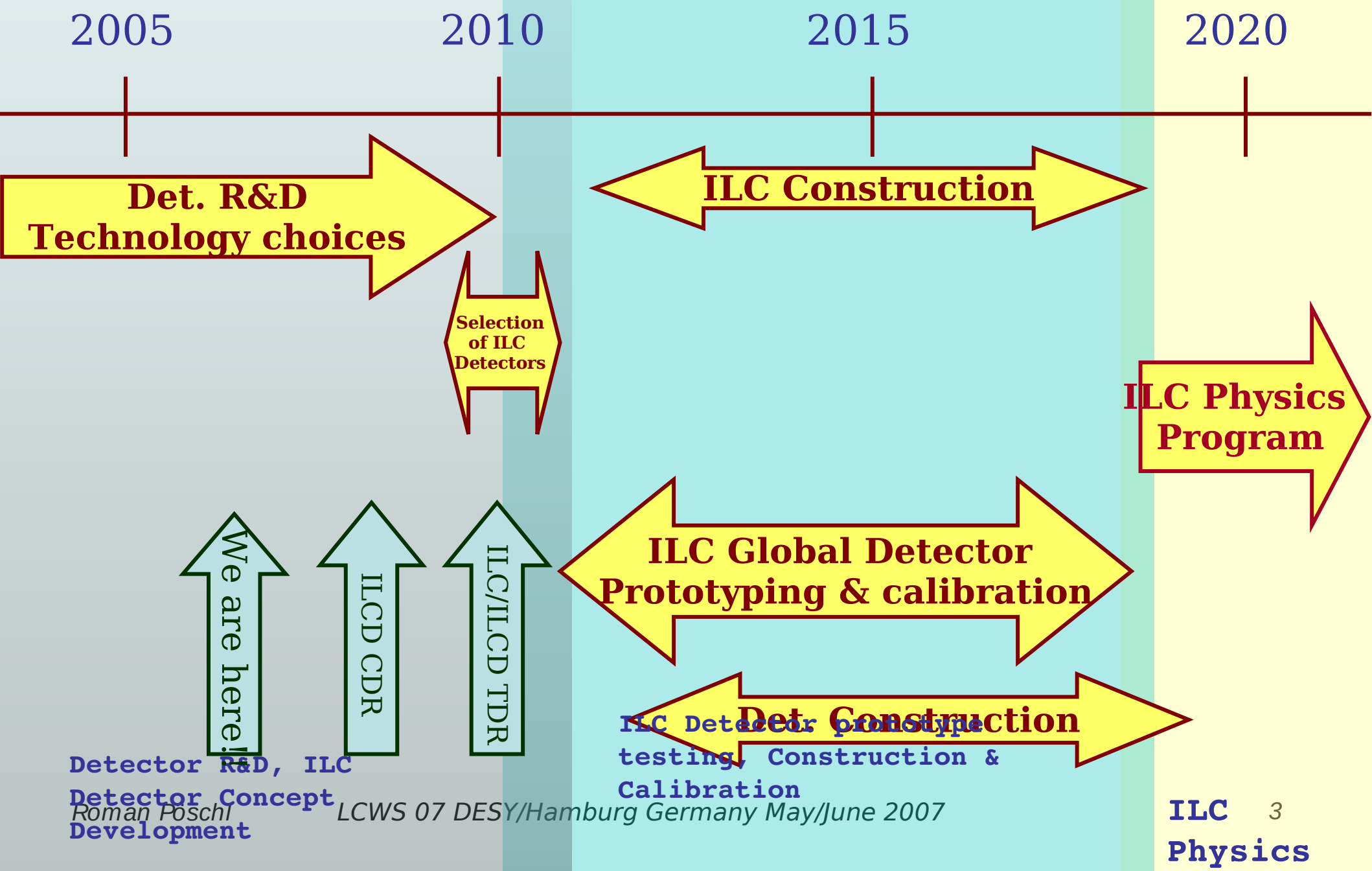
Prologue – Citations from the LDC CDR

"The DAQ System will also benefit from the use of common operation system like Linux and high level programming languages already at the event building and finding stage making the separation of online and offline code obsolete and therefore avoids the need to re-write code for on-line and off-line purposes. This results in a more efficient use of common ressources"

"The machine and beam conditions are vital for the high precision physics analysis ... a common data storage model should be used"

What is true w.r.t. the machine is even more true for the detector

LC Detector Time Line



ILC Detector Testbeam Workshop - IDTB 07

- Workshop: 17/1/07 – 19/1/07 at Fermilab

- Aim of the workshop
  Overview on status of the activities of the various R&D groups/collaborations for the ILC
  Overview on tentative sites for testbeam measurements with some focus on Fermilab facilities
  Identify the future needs of these activities and outline a Roadmap

- ~120 participants from all areas of detector R&D in discussion with maintainers and leaders of the potential test facilities

- Starting gun for the write-up of a roadmap document

Notable requests at the workshop

- Large bore, high field magnet (up to 5T)
  - VTX and tracking groups
- ILC beam time structure (1ms beam + 199ms blank)
  - VTX, TRK and CAL electronics
- Mimicking hadron jets
  - VTX, TRK and CAL

- Common DAQ hardware and software
- Common online and offline software
  - Reconstruction and analysis software

CALICE Testbeam Data Taking

CALICE collaboration is performing large scale testbeams

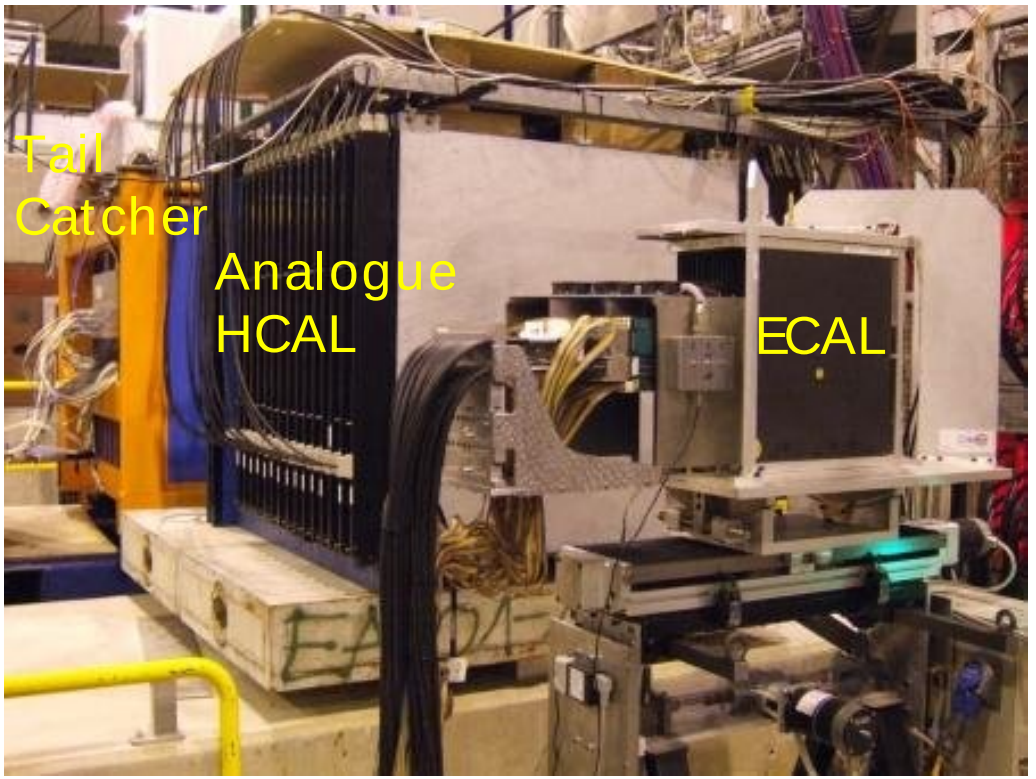
Testbeam program poses software “challenges”

- Data processing from Raw Data to final Clusters in user coherent way

- Handling of Conditions Data Detector Configuration Calibration, Alignment etc.

-Comparison with simulated data 'Physics' Output

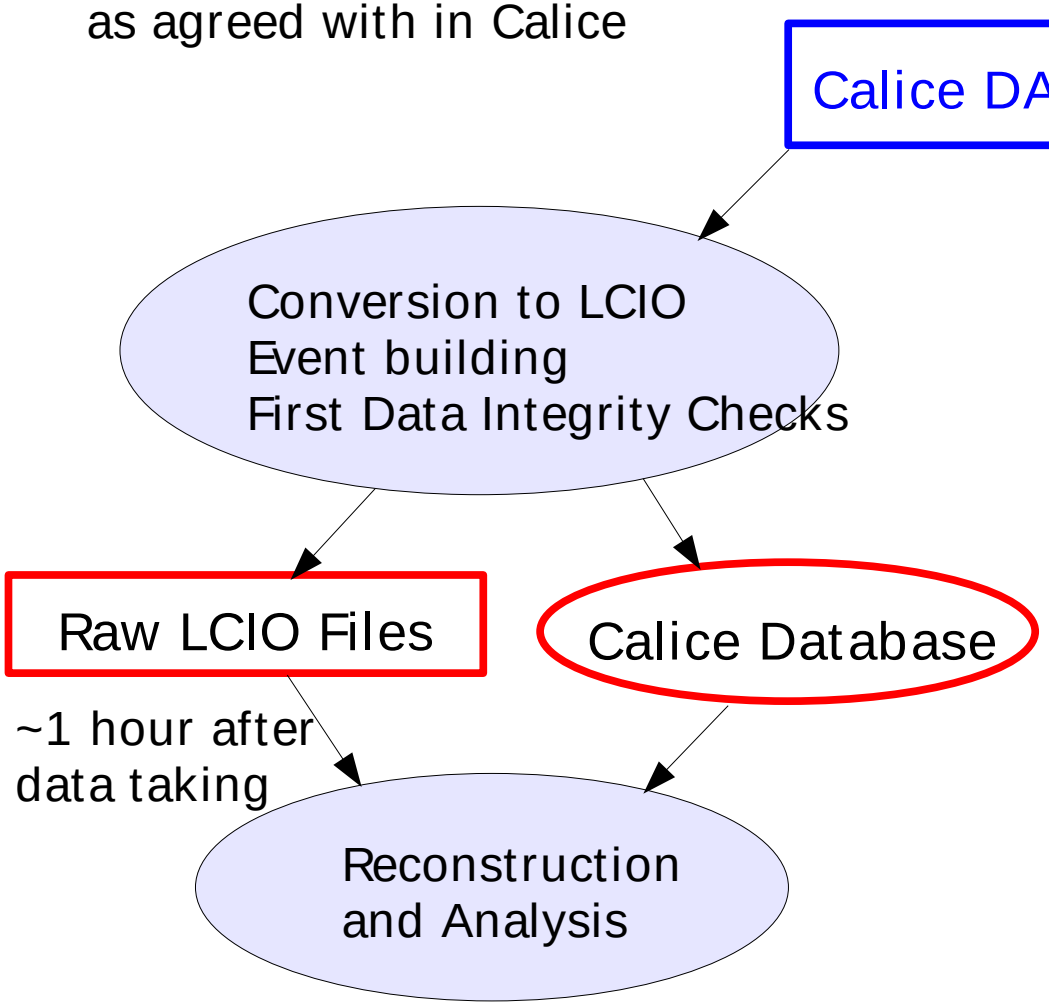
Testbeam Setup at CERN



O(15000) calorimeter cells readout by Calice DAQ
No Zero Suppression
r/o speed 5 Mbyte/s continously

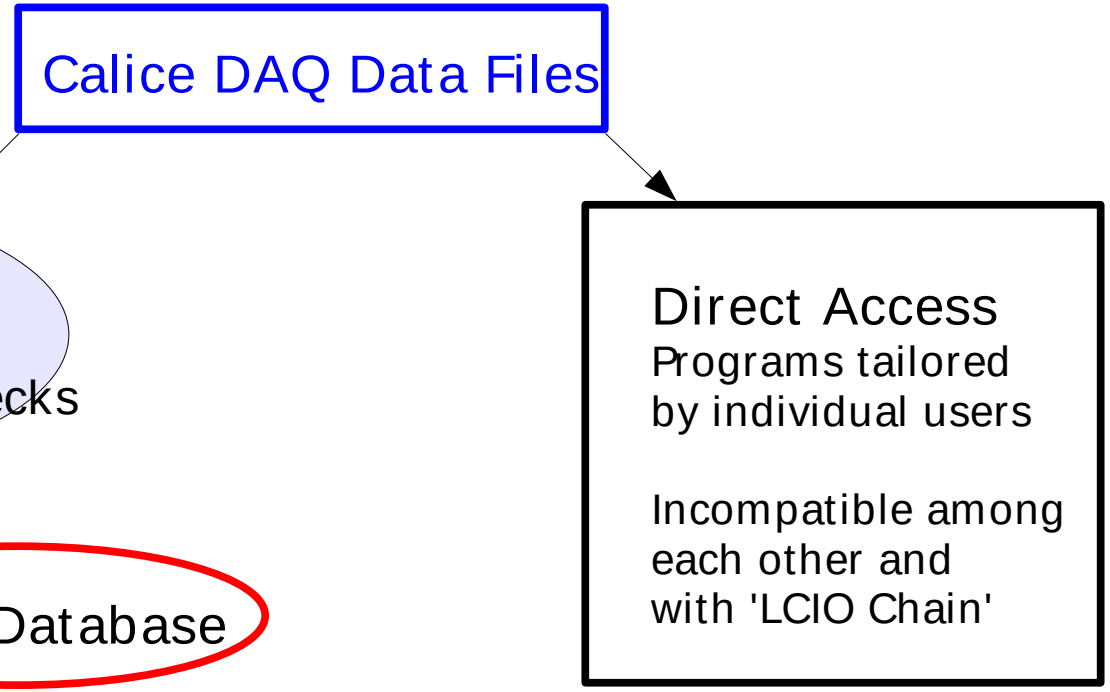
CALICE Dataprocessing

The desired chain
using LCIO and MARLIN
as agreed with in Calice



Complex chain with advantages
and disadvantages of a strucutered
approach

(Part of the) Reality



Individual Approach with
advantages and disadvantages
of a private hack

Pros and Cons using ILC Software Software for Calice Testbeam Data

Pros

Benefit from existing tools/features for/of ILC Software
e.g. LCEvent allows to gather information on event

Newcomers can work in one software framework for testbeam and physics studies

Define at an early stage the of the ILC R&D the needs for a coherent data processing
Coherent s/w concept at time of ILC Detector TDR
Not just guesswork!!!

Cons

Need to wait for converted files
No quick turnaround in particular during development of DAQ and tests
Needs tight communication between DAQ and s/w developpers

Some overhead generated by usage of ILC Software

Source of (potential) errors unclear

Is in the spirit of the LDC CDR!!!!

Status of and of ILC Software w.r.t 'Real' Data

- Existing software mainly developped for simulation studies

  For more information on ILC Software packages see ilcsoft.desy.de

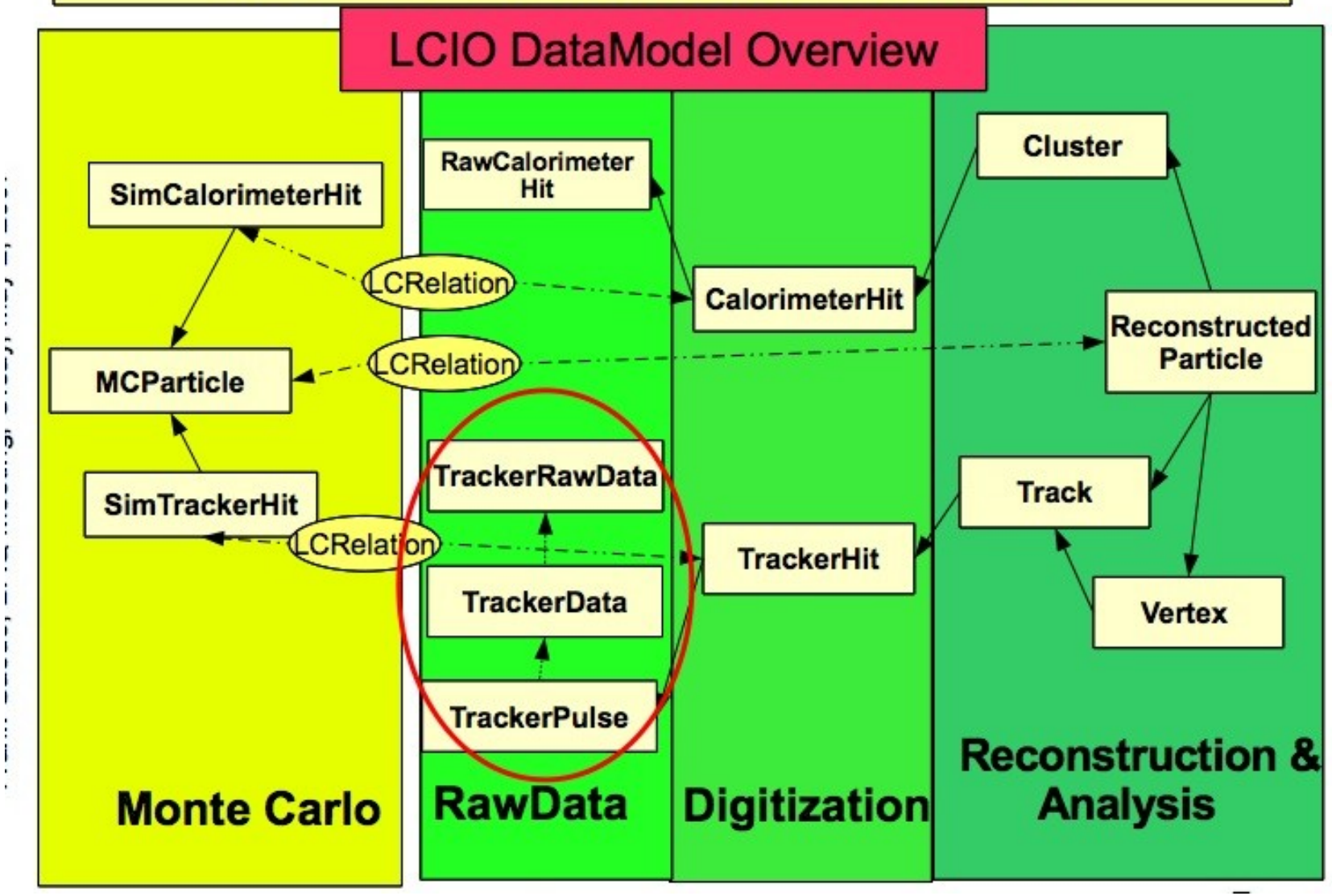
- LCCD

  conditions data framework was/is first approach to store data like Calibration Constants, Detector Configurations Currents Temperatures and to read them back into e.g. your MARLIN Analysis

- LCGenericObjects

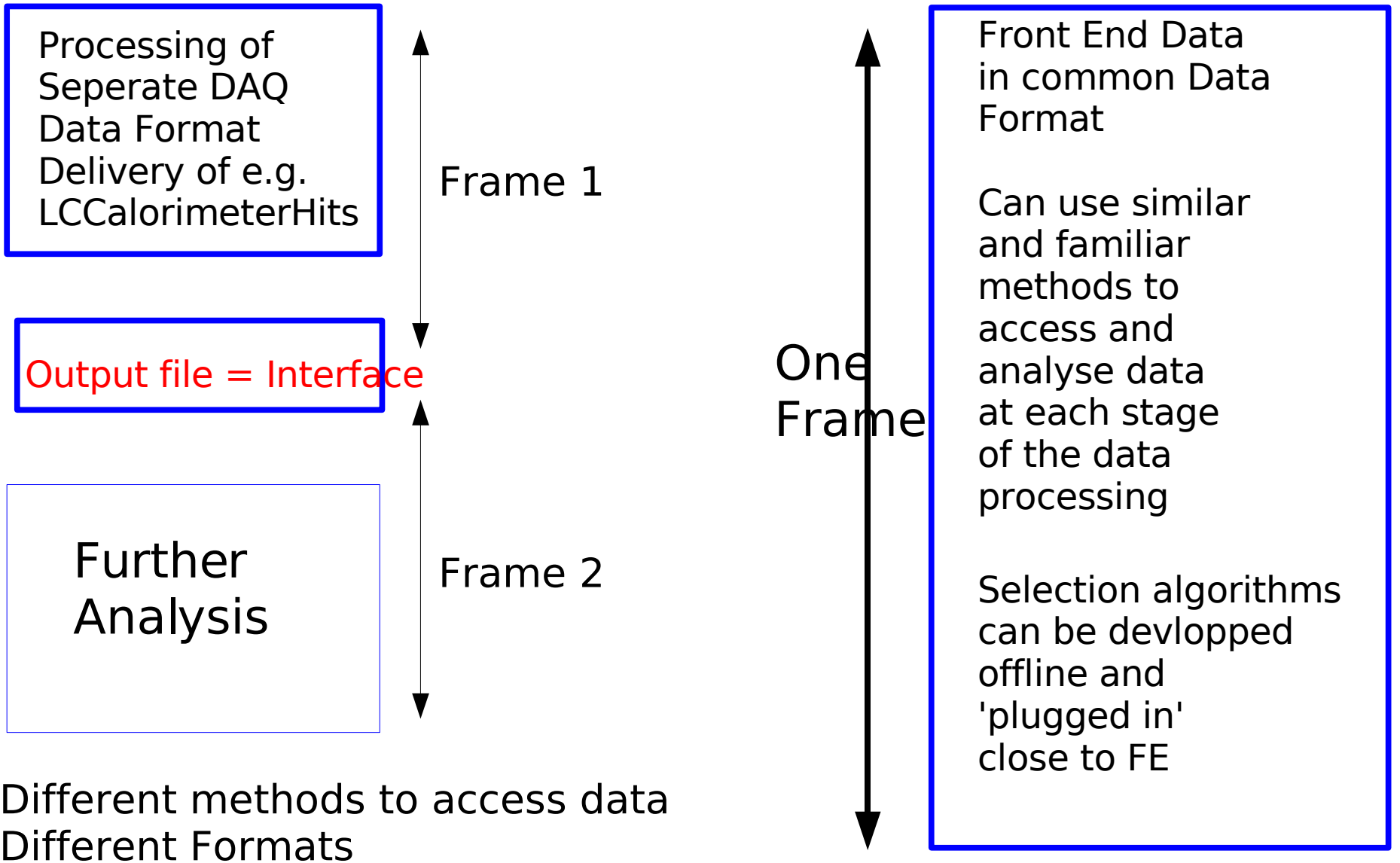
  Allow for an arbitrary defintion of user defined classes within LCIO Application should be the exception Extensive use in calice LCIO conversion Lack of type safety, no official LCIO DataType

LCIO Event Data Model



Interface between 'Online' and 'Offline' World

Decision to be taken on first data format visible to the world



Different methods to access data
Different Formats
Impossible to access information created/used by the other frame

Launching the next steps ...

Need to get overview on ideas for data processing for 'next' generation testbeam projects

Dedicated Meeting at Orsay prior to the ILC software workshop

- Review on Calice experience

- Two projects present their ideas
  'EUDET' Projects so maybe not fully represantive for ILC

- Ideas from software developers

DAQ requirements for LCIO

- in order to further design and develop new features in LCIO it is important that
  - the requirements from the DAQ systems are well known:
    - expected/needed I/O performance
    - data structure – class layout
      - is the data restructered defore written to file is the structure defined by the readout electronics ?
    - will there be an event builder that collects raw data (in non LCIO format?) from the various subdetectors and assembles the LCIO event ?
    - if so, will there be an event builder solution common to all/some DAQ systems ?

LCIO plans

- currently working on a new implementation of the underlying I/O library SIO in C++
- will address these known short comings and provide:
  - direct access
  - split events over several records and files
    - e.g. could write LCIO raw data to one file and processed data (calibrated, zero suppressed,...) to another file
      - users could go back to raw data when needed
  - possibility to read and write user defined classes/records
    - would give DAQ systems flexibility for changes
  - timescale: unclear - depends on manpower situation and needs/requirements of DAQ systems

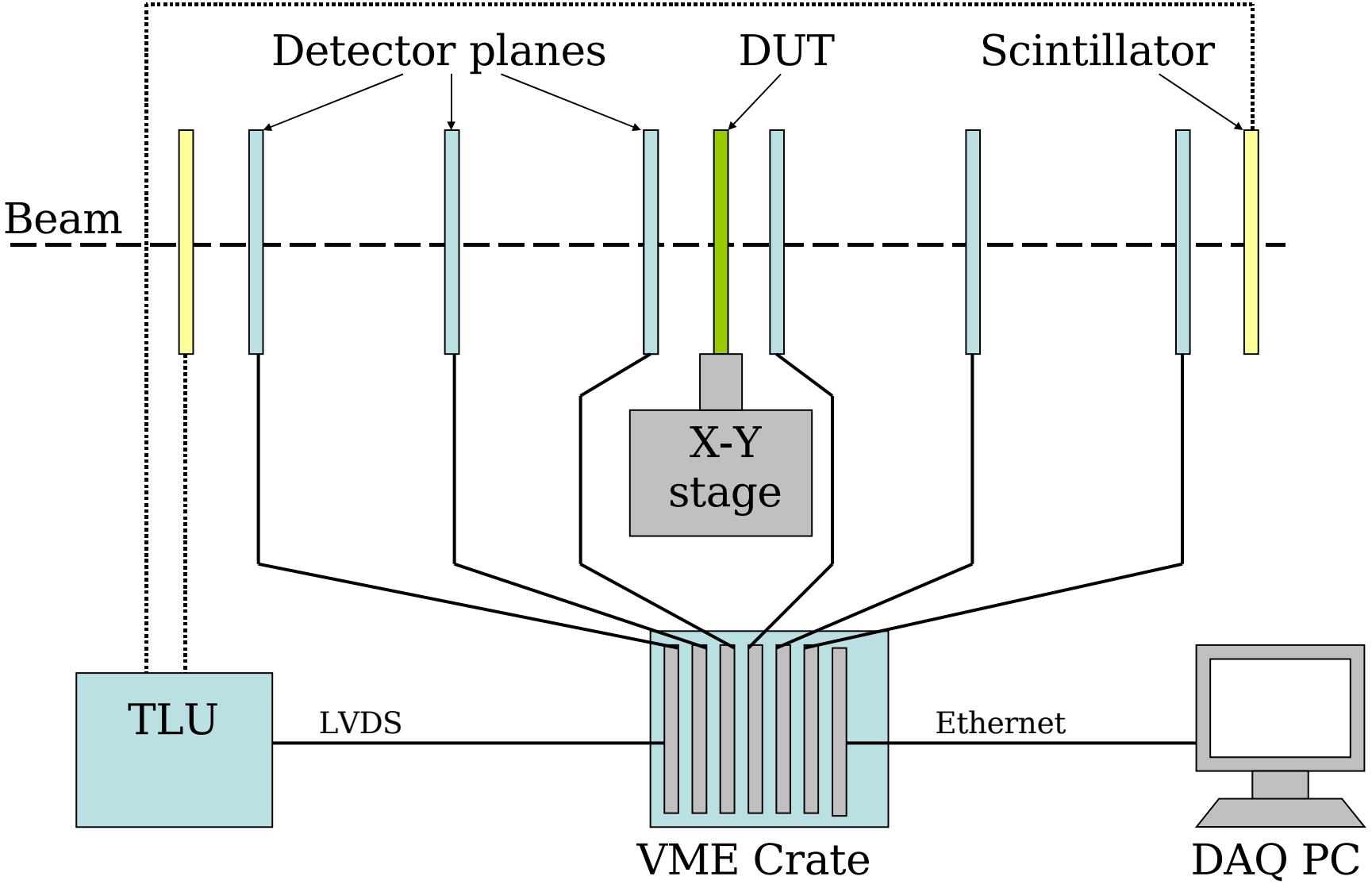
Project(s) which will use lcio as the

Data format for data processing

In the following material
by Emlyn Corrin, David Bailey and Frank Gaede
will be shown

The beam telescope

JRA1 Project within EUDET



Data Rates

- 6+1 planes (telescope + DUT)
- Event size:
  - Worst-case: 3 frames * 1 Mpixel * 16-bit: 6 MB per plane (Raw).
  - With sparsification: ~10kB?
- Trigger rates:
  - From ~1Hz (initial debugging, fully raw)
  - Up to ~1kHz?

Data Format

- Custom classes used internally
- DataCollector writes LCIO files
- Event model:
  - EUTelRunHeaderImpl for header (reimplementation of LCRunHeaderImpl)
  - TrackerRawDataImpl for raw data
  - TrackerDataImpl for noise, calibrated data, and clusters (EUTelFFClusterImpl)

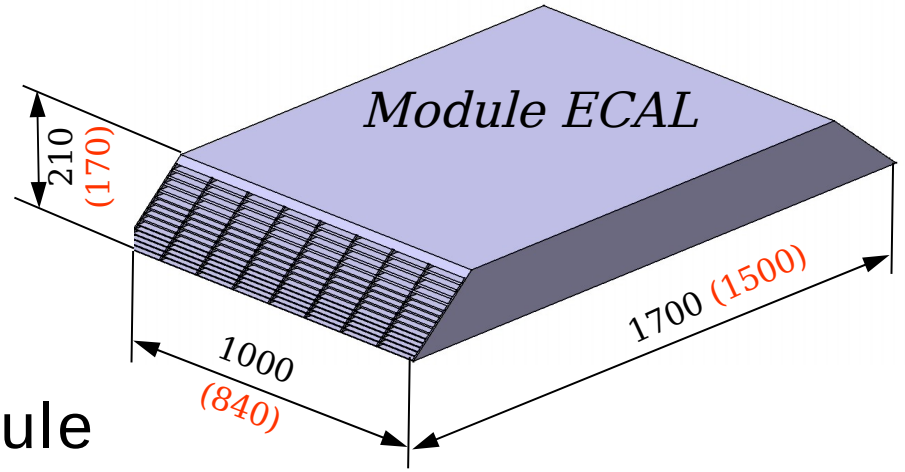
EUDET/Next Generation DAQ

Dave Bailey for the UK DAQ Group

This DAQ will be used for next generation Calice/EUDET Calo Prototypes

Coming close to the ILC – The EUDET Module(s) and a common Calo DAQ

EUDET module of Ecal is ~1st Modules of ILC Detector

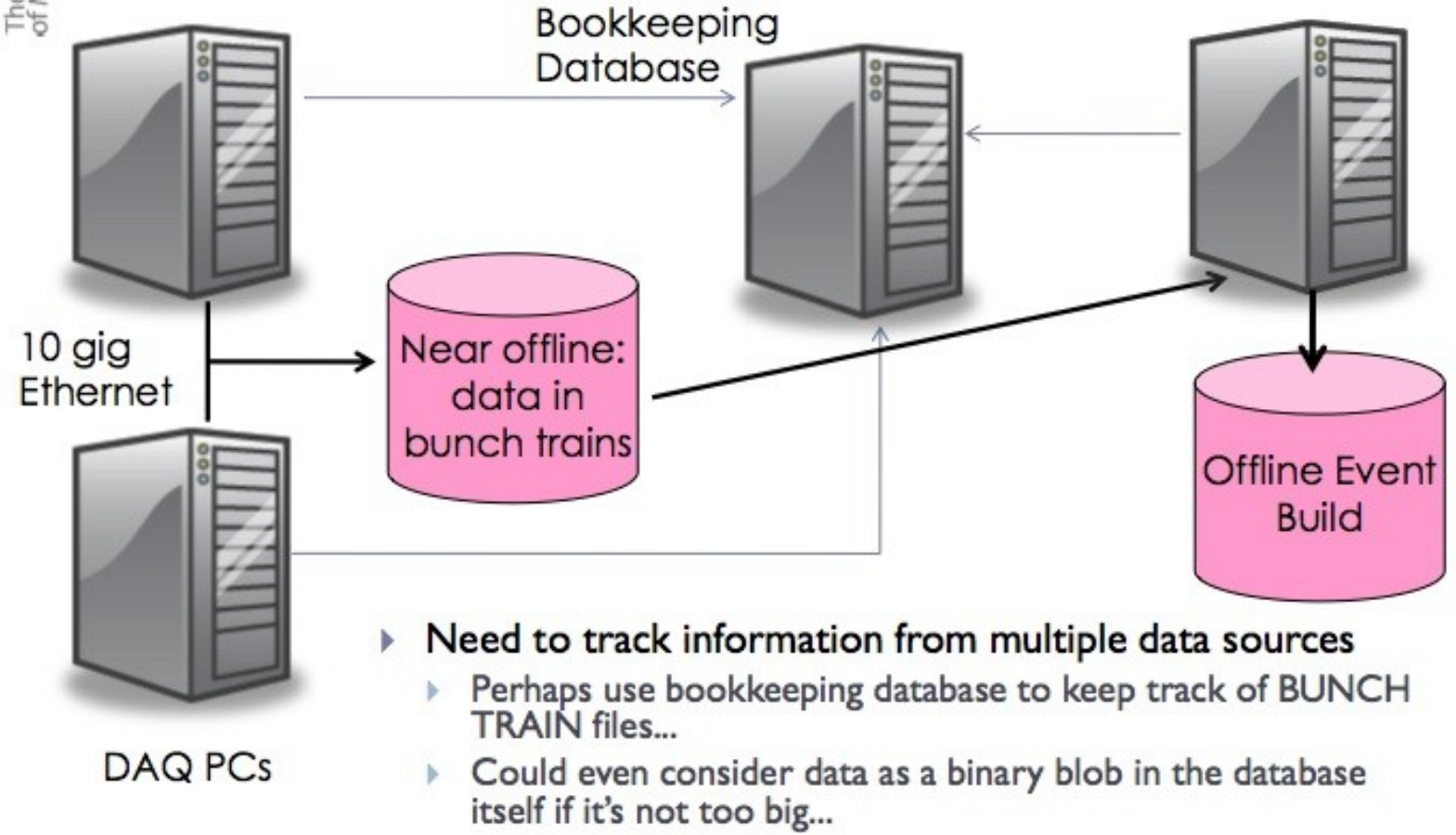


Common DAQ for EUDET module (and other EUDET Calos) may be ~ (Calo) DAQ Concept for ILC Detector

Ideal testing ground to qualify concepts for whole ILC Dataprocessing

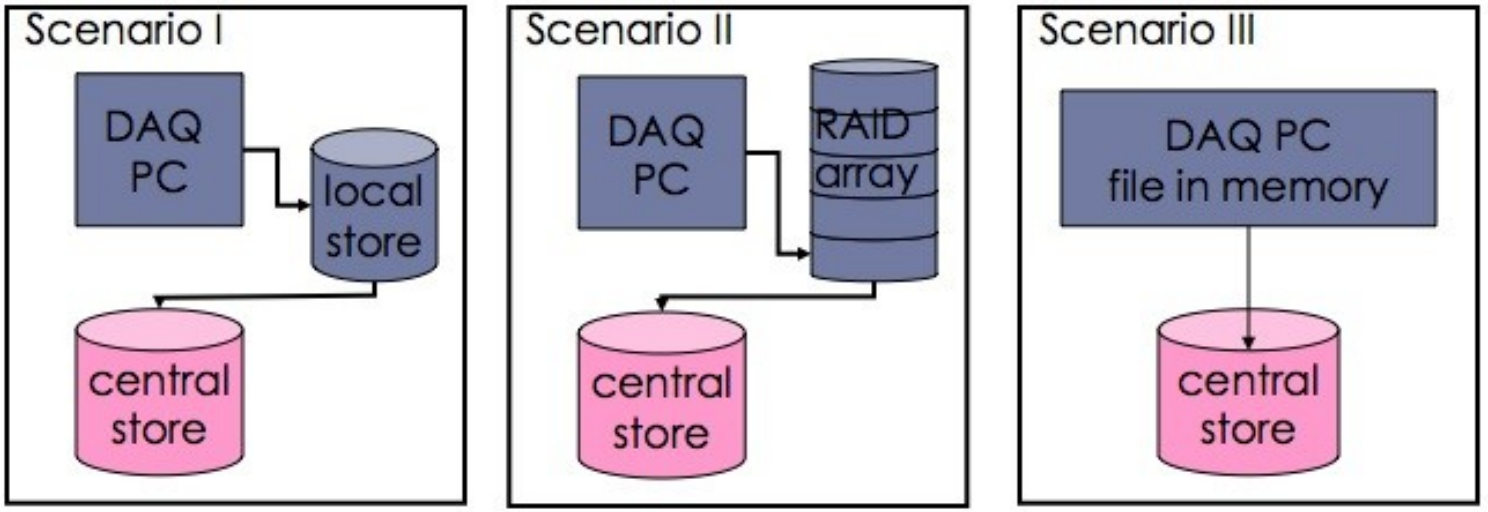
Develop and optimize ILC Software e.g. LCIO for 'Real Data'

Near-Offline and Event Assembly



- **Need to track information from multiple data sources**
  - Perhaps use bookkeeping database to keep track of BUNCH TRAIN files...
  - Could even consider data as a binary blob in the database itself if it's not too big...

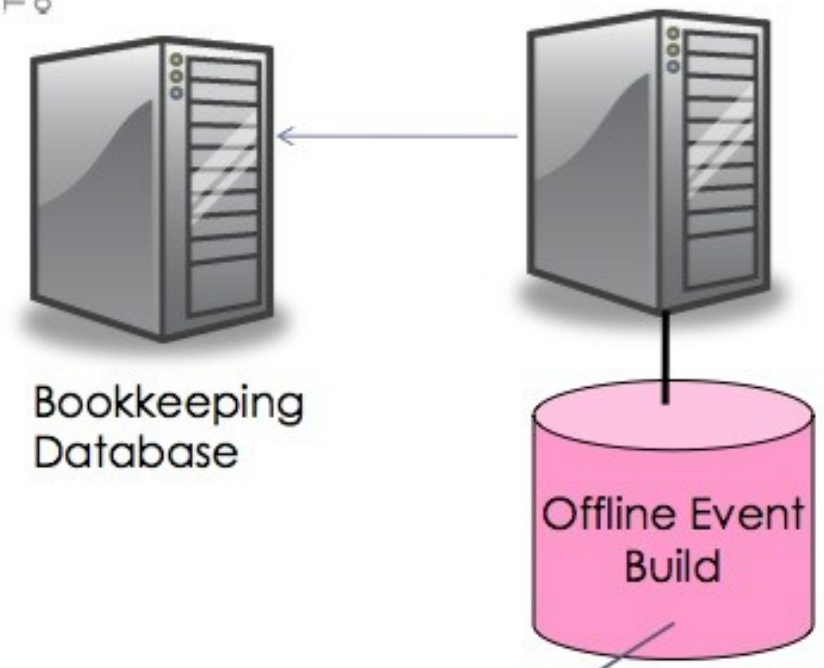
Focus on Off Detector System



- Various scenarios
  - Mostly depending on how much/how long we buffer the data on disk in the DAQ system
  - Important Point
    - Expect the data on the "Local Store" to be in **BUNCH TRAINS**, not events.

What is true for the Calo DAQ will be true for all DAQs(?)

Event Assembly



- Building EVENTS from the bunch train
  - Almost certainly non-trivial at the beginning
  - This is the first place in the data chain where the concept of an event starts to make sense
  - This is the first time that "real consumers" of the data get their hands on it

If you want LCIO then this is where it appears...

Decision to be taken: Event Building integrative part of DAQ chain or separate entity

Interface Definition !!!!

Remarks on Infrastructure – Apart from actual testbeam site

- Testbeam data taking has a lot in common with real data taking

Need to store and handle Conditions Data
ILC institute to provide database service (as DESY for Calice)

- Testbeam efforts are organized in Collaboration
+ combined testbeams which join R&D groups

Data needed to be distributed to collaboration members

This is the place where the Grid comes in – Virtualisation of Resources
Testbeam collaboration may identify one principle site and organize the interplay with other sites
(again Calice to some extent shows the way)

Summary and Outlook

- Integration of high level analysis algorithms into DAQ is part of LDC Detector Concepts

  All relevant detector data must be available in one format

- The need/wish to establish ILC Software also in testbeam projects has been identified and R&D projects try to incorporate this into their concepts – Positive tendency

- Calice has made first steps into that direction

- EUDET DAQ and EUDET Modules allow for developmen of and real life tests of coherent scheme for ILC data processing

  DAQ Development and Software Development NOT seperate entities

- Needs tight communication between 'online; and 'offline' world to obtain a well defined interface

- The setup of a working dataprocessing chain does need considerable manpower!!!