CALICE Software

Niels Meyer
DESY Hamburg

Roman Pöschl
LAL Orsay

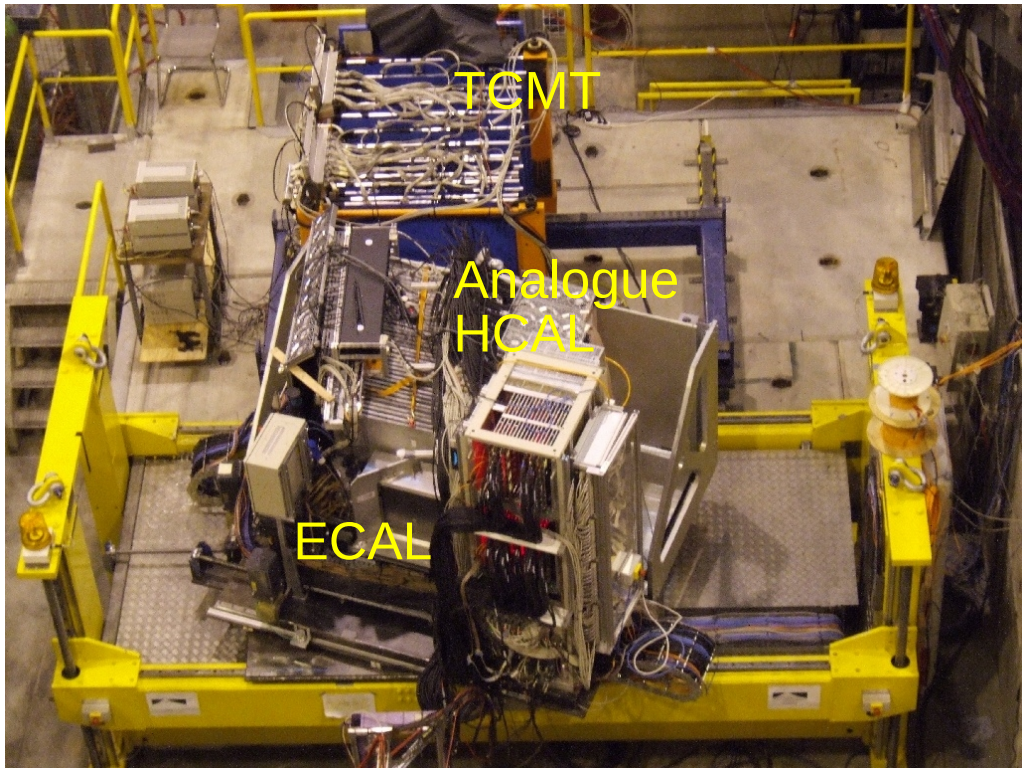
- Calice Testbeam Data Taking
- Data Management
- Event Building and Reconstruction Software
- Summary and Outlook

LCTW09 Orsay/France November 2009

CALICE Testbeam Data Taking

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

Testbeam Setup at CERN 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

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

CALICE “TIER 0” – Infrastructure in the Control Room

Picture courtesy of C. Rosemann DESY

Gigabit Uplink
- High Speed Connection to the outside world
- Serves all Calice Control Room Computers

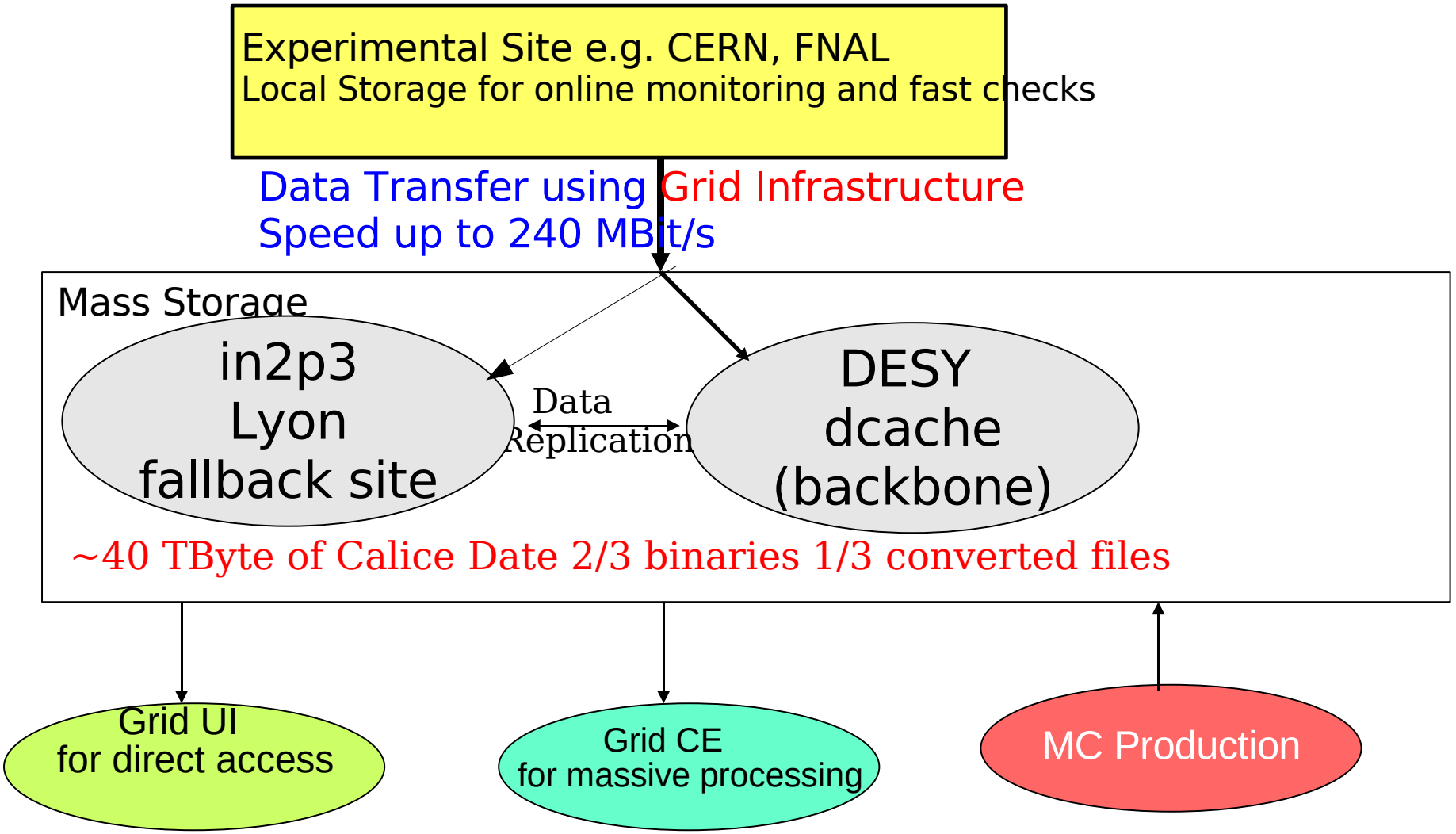
caliceserv.cern.ch
- Online Monitoring
- Grid Transfers

Disk Array

DAQ Computer

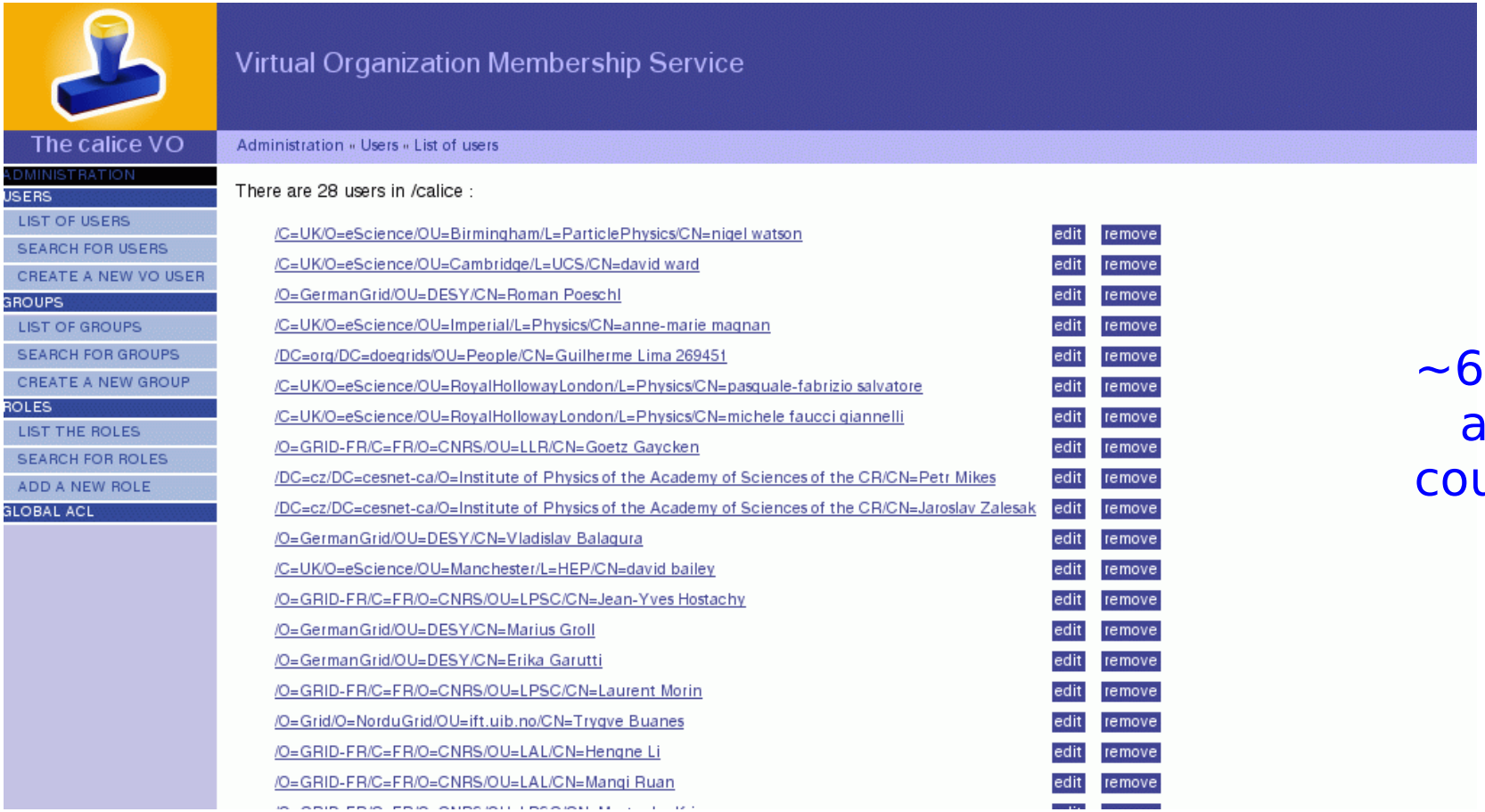
Well organized setup of computing
Thanks to B. Lutz

Data Handling and Processing



The Virtual Organisation - vo calice

Hosted by DESY:
Page for registration is https://grid-voms.desy.de:8443/voms/calice



~60 Members
and
counting ...

VO Manager: Niels Meyer/DESY, Deputy: A. Gellrich/DESY

Institutes which provide Grid support for Calice

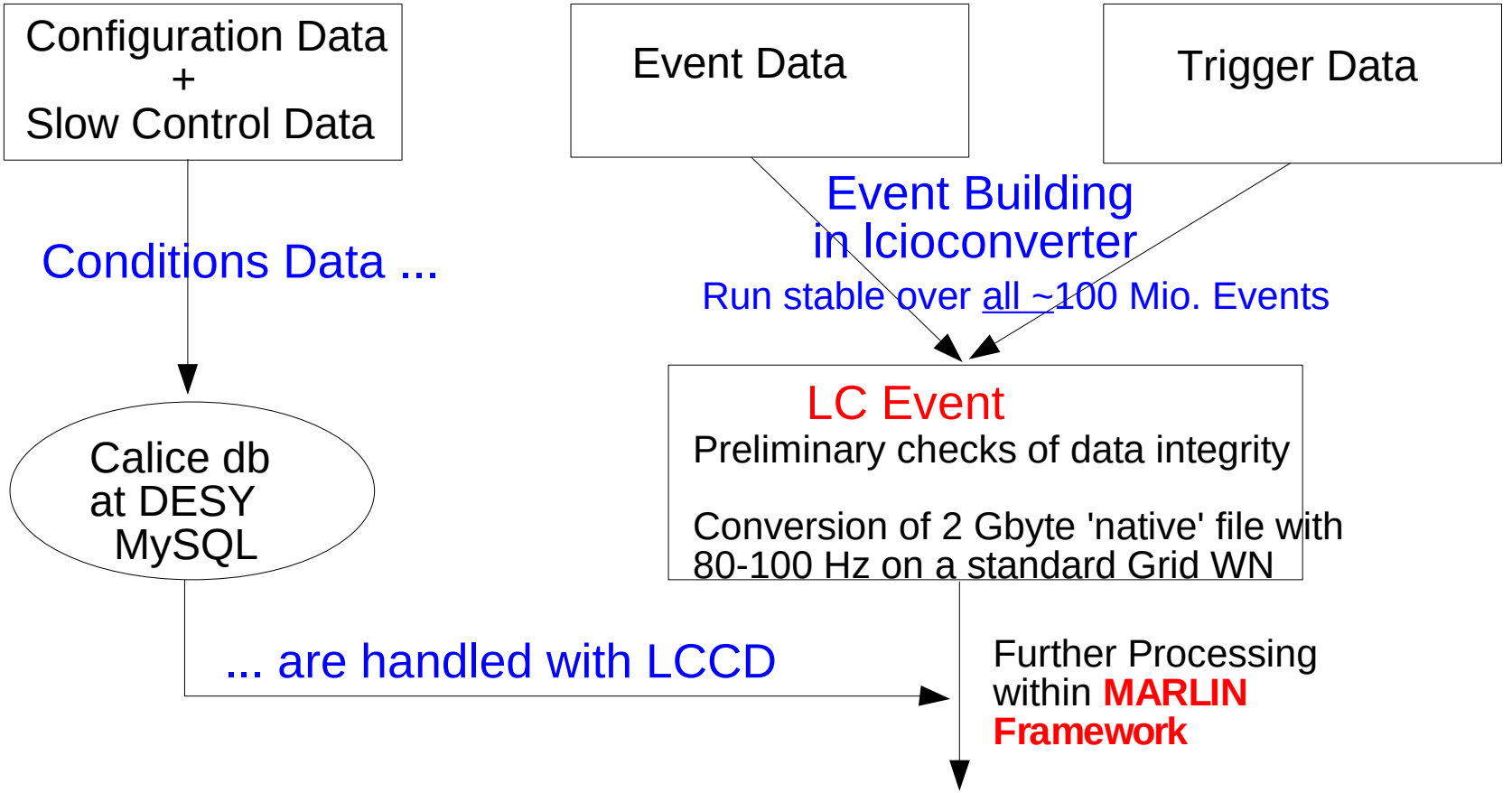
Supported by: DESY Hamburg	Hosting, Computing and Storage
LAL	Computing and Storage
LLR	Computing and Storage
DESY Zeuthen	Computing and Storage
Imperial College	Computing and Storage
Birmingham	Computing and Storage
cc in2p3 Lyon	Computing and Storage
Cambridge	Computing and Storage
Institute of Physics Prague	Computing and Storage
University College	Computing and Storage
KEK	Computing and Storage
Manchester	Computing and Storage
CIEMAT Madrid	Computing and Storage
Fermilab	Computing and Storage
NIKHEF	Computing and Storage
University of Bonn	Computing and Storage
Univ. Liverpool	Computing and Storage
Univ. Oxford	Computing and Storage

- Most of the sites have been involved in recent data and MC processing
  Connectivity to Asian Sites still and Issue
- Full set of Rawdata avalaible at DESY HH and CC in2p3 Lyon!!!
  Grid exploitation of Calice paved the way for successful mass production for ILC detector LOIs

Conversion to LCIO

DAQ data types are converted/wrapped into LCIO on the basis of LCGenericObjects

DAQ Data Files/Types



Installation of Software using tools developped in EUDET NA2 task
ilcinstall, cmake

Intermezzo – Conditions Data Handling

- LCCD – Linear Collider Conditions Data Framework:
  - Software package providing an Interface to conditions data
    - database
    - LCIO files

    Author Frank Gaede, DESY

LCCD works and is heavily used within calice !!!

Still too much an expert tool (No real development since 2005)

The importance of conditions data (not only) for 'real' data renders the development of a fully functional cd data toolkit to be a fundamental !!! piece of the ILC Software

- Efficient storage and access to conditions data

  Browsing, convenient interfaces
- How to 'distribute' conditions data (e.g w.r.t to grid) ?

  BTW.: LHC does have some headache with that!

Calice Software

Three main packages

Contributions by groups from
DESY, Imperial, LAL, LLR, NIU, RHUL

Current version
v04-02-06

converts
calice DAQ format
into LCIO (LCGenericObjects

needs DAQ software
expert work

MARLIN processors

Current version
v04-10

Interface classes to
LCGenericObjects
(these classes should
be defined by LCIO)

utililty functions e.g.
For TriggerHandling

Current version
v04-06-05
(v04-07 in prep.)

RawData into
CalorimeterHits
(standard LCIO)
TrackerHits

First stages of
higher level analysis
MARLIN processors

~250 classes or functions

Data of four different Calorimeter Prototypes are available in LCIO format

LCIO

- No data model for conditions- and meta-data, have to be implemented as LCGenericObjects
- This is fine, because most of these structures are easily generalized (e.g. slow control data, detector configuration, beam-line status)
- Current problem: object is bare LCGenericObject after storage/re-read
  - danger of confusing different objects with identical size
  - dynamic_cast for read-back collections does not work
    => need to work with constructors to be transparent
    => unnecessary increase of memory footprint

LCCD

- In principle, the LCIO/Marlin *concept* is thread-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: ~50 floats per channel). Need clever memory CPU I/O balancing in the future, preferably steerable.

Alignment

- For PFlow calorimeters, cells are smaller than Moliere radius => mis-alignment in data affects the energy distribution over cells
- At least for TB: need to have this mis-alignment also in MC in order to do cell-to-cell comparisons (crucial to proof detector understanding at least for Tile HCal)
- Thus: alignment is conditions data and should be handled using LCCD interfaces
- Closely linked to geometry description (e.g. cell neighbours, cell positions, ...)

Geometry

- GEAR is NOT a generic geometry interface, it is the geometry parameterization used for ILD
- 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 with floating alignment
- Would want to have some LCIO-embedded data model that is good for calibration, simulation, and analysis alike. Something more object oriented (hit has pointer to cell rather than complex 'find cell by index from somewhere') would be much easier to use in the end, but this is re-discussing basic LCIO paradigms...

Next Generation Protoypes – EUDET Modules

- From the beginning coherent interface between DAQ and offline processing (D. Decotigny et al.)

  LCIO will remain backbone!!!

  Consistent Handling of Low Level Data
  Coordinated handling of potentially frequent changes in startup phases

- Will continue to apply and help to develop ILC Software tools

  - Need for geometry package
    Consistent between Data and Simulation

Summary and Outlook

- Calice uses ILC Software for processing of Testbeam Data

  ILC Datataking in a (big) nutshell

  Allows users to switch easier between testbeam data analysis and physics/simulation studies

- Calice uses systematically Grid tools

  24h/24h 7h/7h during CERN, FNAL testbeams 2006-2009

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

- Effort will continue with EUDET Modules on an even broader basis

  Using of ILC s/w tools already in testbeams is (in the mean time) well established and accepted concept

CALICE did/does not only hardware-prototyping but also 'computing prototyping'

Computing benefits from collaborative effort and application of ILC software toosl