# Software Summary

Frank Gaede
DESY

Linear Collider Testbeam Workshop
LAL Orsay, November 3-5, 2009

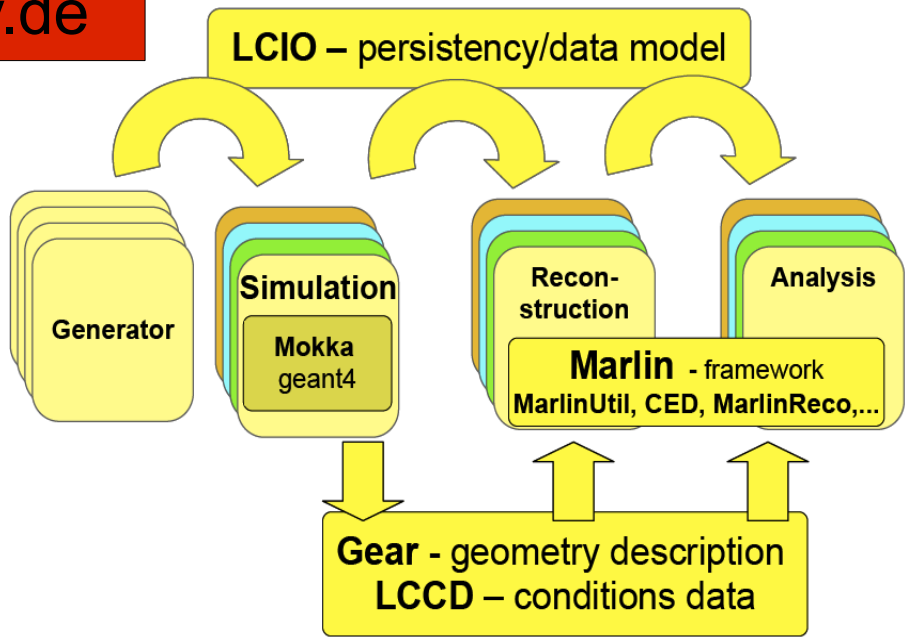
# Software session: 4 Talks

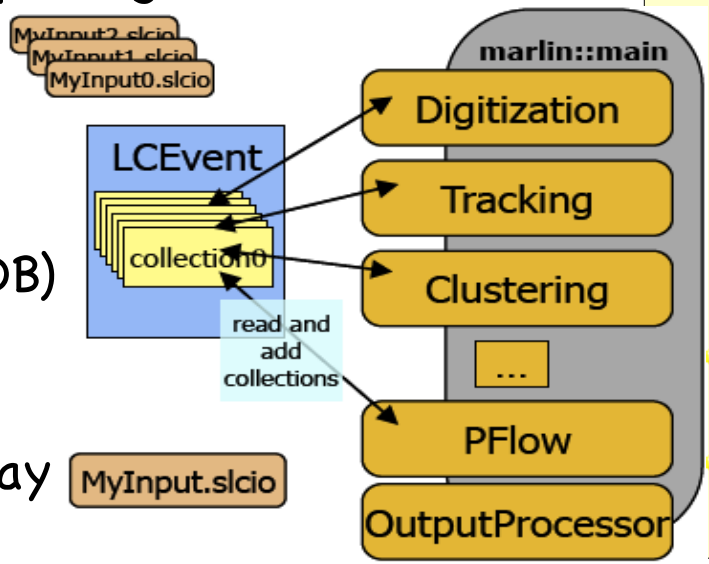
- Overview and plans for core software tools [FG]
- TPC tbeam software - experience and needs [M.Killenberg]
- Calice tbeam software - experience and needs [R.Poeschl]
- Communications tools [Sven Karstensen]

# ILD & tbeam Core Software Tools

http://ilcsoft.desy.de

- **Mokka** (LLR)
  - geant4 simulation application
- **LCIO** (DESY/SLAC)
  - international standard for persistency format / event data model
- **Marlin**
  - core application framework for reconstruction & data analysis
- **GEAR** geometry package f. reconstruction
- **LCCD**
  - conditions data toolkit (DB)
- **CED**
  - 3d event display



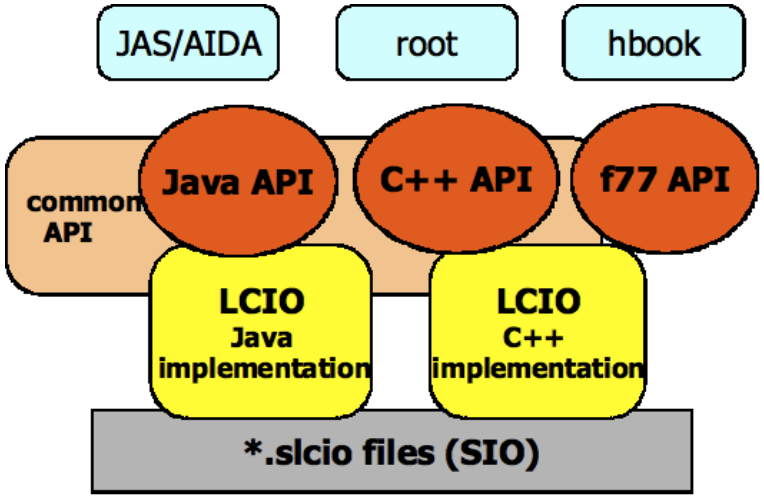

- used for ILD and testbeam:
  - **ILD detector concept** studies
  - **Calice** calo testbeam
  - **LC-TPC** testbeam
  - **EU-PixelTelescope**
- **benefitted from EUDET project**
- **synergies between testbeam and global detector optimization**

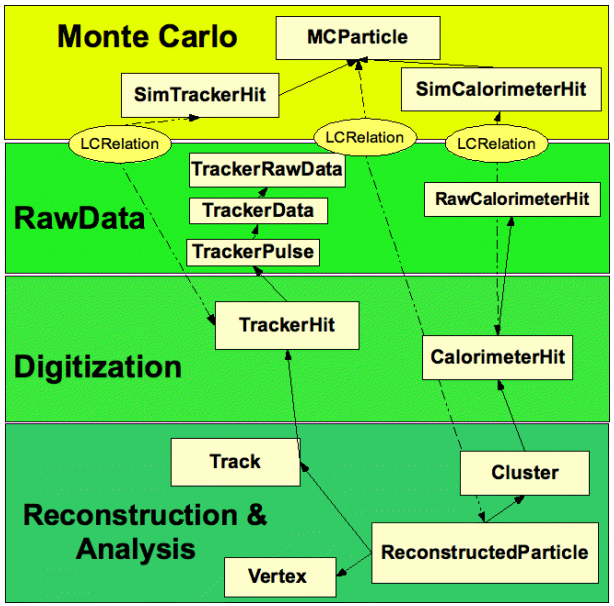
# Plans for (ILD) core software

- after LOI we need to further improve ILD software and get ready for TDR phase (2012)
- most if this will of course be beneficial for the testbeams as well !
- plans:
  - improve the simulation & reconstruction
  - develop a test system
  - develop new GRID production system
  - improve the geometry description
  - develop LCIOv2

# LCIOv2

- improve event data model
  - 1d, 2d hits & Track class
  - as requested by testbeams
- Improve I/O
  - splitting of files - read events partially
  - direct access
  - srteaming of user classes
- look into ROOT I/O for LCIO
  - LCEvent in ROOT macros
  - look into optional ROOT I/O for LCIO
- optionally improve existing SIO
  - tens of TByte of testbeam data exist in .SLCIO
  - need to fully support I/O for this data in the future !



# Grid computing for ILC

- massive computing resources will only be available on the GRID
- major Gris Sites do commit resources to ILC but no MoUs exist as in WLCG (yet)
- data storage needs commitments from sites - large testbeam data repositories have to be allocated
- successfully used by ILD and testbeams
  - calice: 55TByte/80CPU*y , ilc: 87TByte/500CPU*y
- ->need to negotiate further commitment of Grid sites for CPU & storage for ILC in the future

# fp7 proposal AIDA - WP2

- in WP2 'Common Software' will develop common tools for HEP community with focus on future accelerators, i.e. sLHC, ILC, CLIC (budget ~3FTEs*4 years )

**Objectives**

**Task1: Coordination of Work package**

- monitor the progress of the work in the work package
- coordinate and schedule the execution of the tasks and subtasks
- prepare progress reports – internal and on deliverables

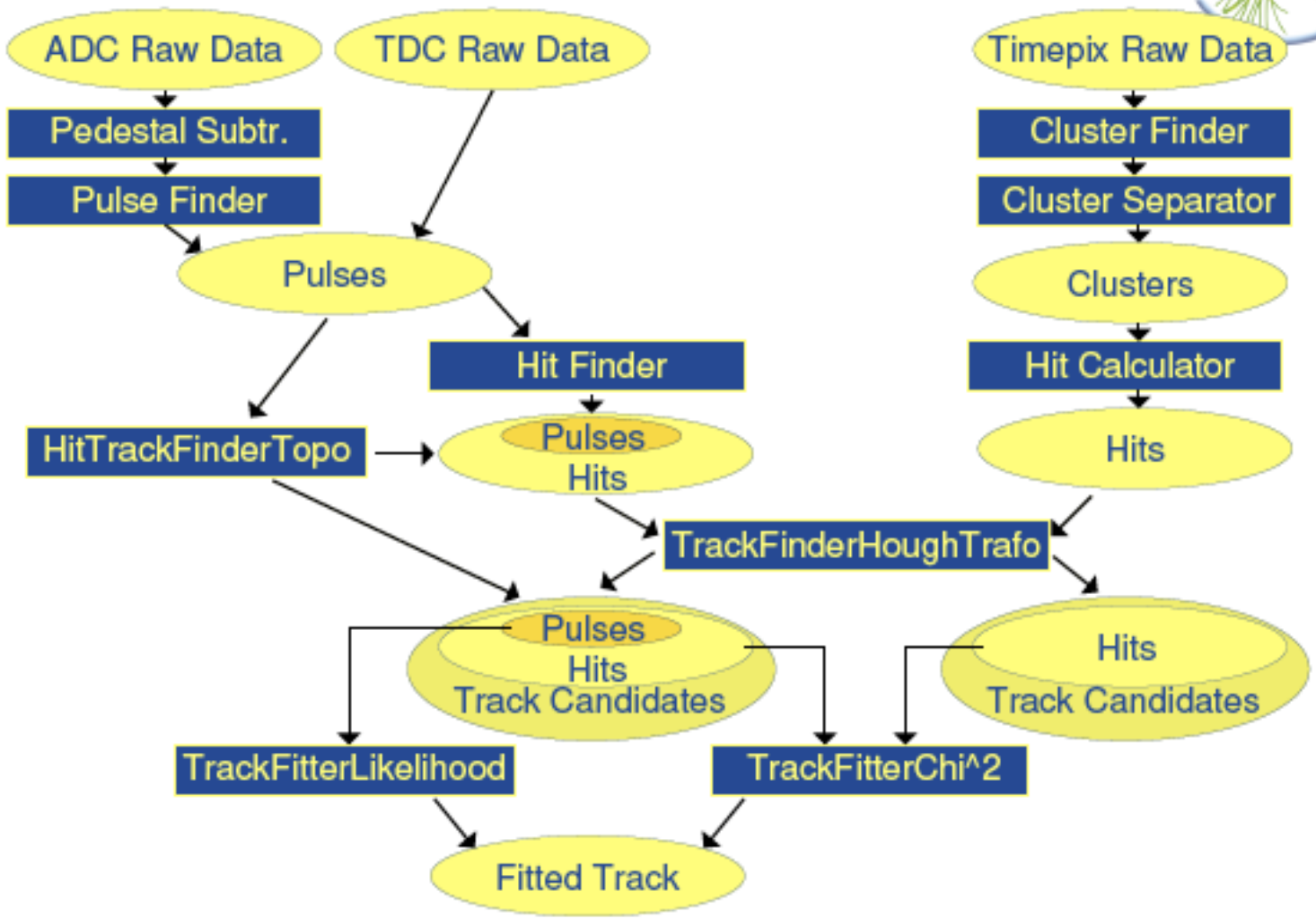
**Task2: Development of a geometry toolkit**

- allow the description of complex geometrical shapes, materials an sensitive detectors
- provide interfaces to full simulation programs (Geant4), fast simulations, visualization tools and reconstruction algorithms
- allow for the misalignment of detector components
- provide an interface to calibration constants and conditions data

**Task3: Development of generic reconstruction tools**

- tracking toolkit based on best practice tracking and pattern recognition algorithms
- provide alignment tools
- allow for pile up of hadronic events
- calorimeter reconstruction toolkit for highly granular calorimeters based on Particle Flow algorithms

# MarlinTPC Reconstruction Overview




- nice example of modular Marlin application
- some boxe still have to be filled with life

## Conditions Data

- Linear Collider Conditions Data toolkit
- TPCCondData: LCIO / LCCD classes for the TPC

| Data needed | TPCCondData | Available for test beam run | | | | |
|---|---|---|---|---|---|---|
| | | Japanese ALTRO | GEM TDC | Micro-megas | Std. GEM | Timepix + GEM |
| Channel mapping | ✓ | ✓ | ✓ | ✓ | ✓ | |
| Channel quality | ✓ | × | × | × | × | × |
| Pedestals | ✓ | × | × | × | × | × |
| $v_{drift}$ + Diffusion | ✓ | × | × | × | × | × |
| Gas conditions | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| E-field settings | ✓ | × | × | × | × | × |
| B-field settings | × | | | | | |
| HV settings | × | | | | | |
| Field maps | ✓ | | | | | |
| Electronics | × | | | | | |
| Calibration | × | | | | | |

We have to get a woking data base and conditions data bookkeeping



request for improving LCCD:
store some conditions (Meta) data in DB tables for queries – vs BLOBS

## Event Display

For development of geometry descriptions and fist checks during data taking a graphical viewer is needed.

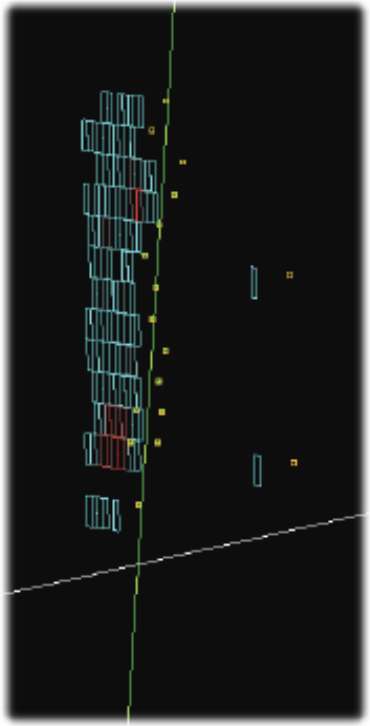
HepRepOutputProcessor produces HepRep XML file which can be displayed e. g. with Wired/JAS3

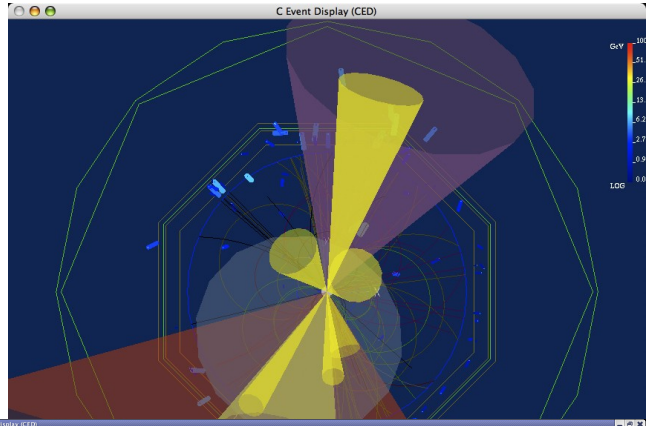
Event display shows

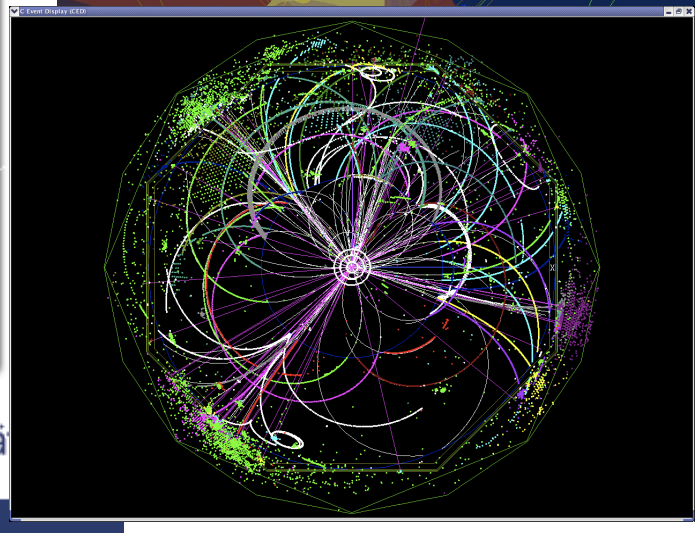
- TPC
- GEAR pad plane
- Charge on pads
- 3D hits
- Tracks

Disadvantages

- MarlinProcessor is offline software
- Events not browsable
- Graphical display has errors (depth)
- Very bad performance





universitä

Martin Killenberg (Universität Bonn) TPC Testbeam Software 9

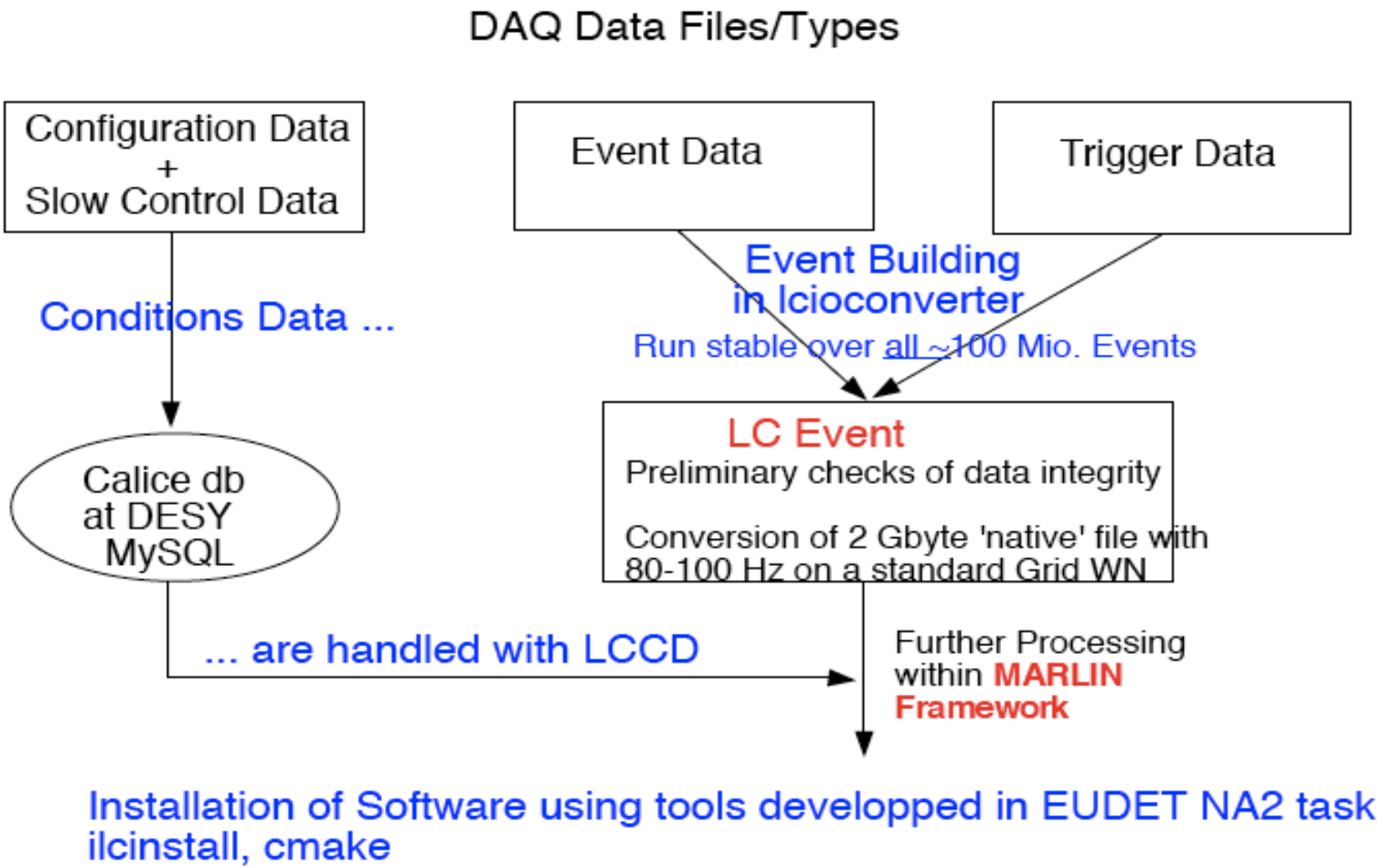
- other event displays exist, eg. CED in core framework, calice ...
- discussion about a common event display project
- issues: manpower and priority ?

# calice data processing scheme



## Conversion to LCIO

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




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

calice where the first to adopt global sw tools

# calice software packages



## Calice Software

### Three main packages

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

calice_lcioconverter ↔ calice_userlib ↔ calice_reco

**calice_lcioconverter**

Current version
v04-02-06

converts
calice DAQ format
into LCIO (LCGenericObjects

needs DAQ software
expert work

MARLIN processors

**calice_userlib**

Current version
v04-10

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

utililty functions e.g.
For TriggerHandling

**calice_reco**

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

# calice sw issues – wish list



## LCIO

- No data model for conditions- and meta-data, have to be implemented as LCGenericObjects
- This is fine, b re easily generalized ( configuration
- Current probl ct after storage/re-re
  - 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

• streaming of user classes
•improve I/O

## 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 ch
- Th LCCD is 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.

•replace CondDBMysql ?
•improve performance

## 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 r MC in order to do cell-to-cell comp proof detector understanding at le
- Thus: alignment is conditions data handled using LCCD interfaces
- Closely linked to geometry description (e.g. cell neighbours, cell positions, ...)

•generic geometry package
•including conditions data (alignment)

## Geometry

- GEAR is NOT a generic geometry interface, it is the geometry parameterization used for ILD

  en: geometry defined in uction by XML file - this is not a eam setups (which are very bly also not for a full-scale gnment

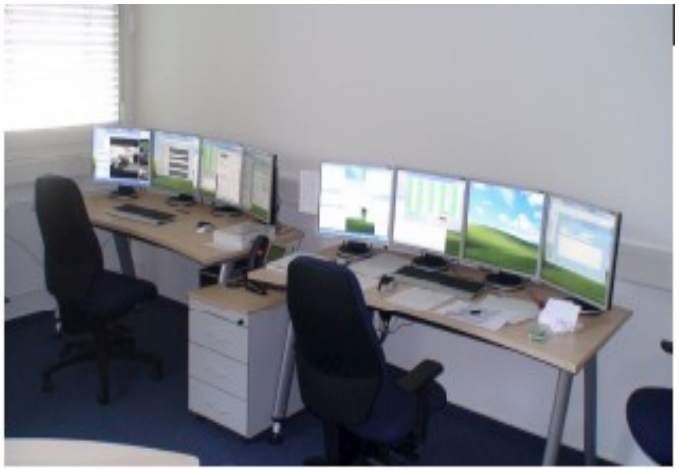
  me LCIO-embedded data model that is good for calibration, simulation, and analysis alike. Something more object oriented (hit has pointer to cell rather than complex 'find cell by index from somewhere') would be much easier to use in the end, but this is re-discussing basic LCIO paradigms...

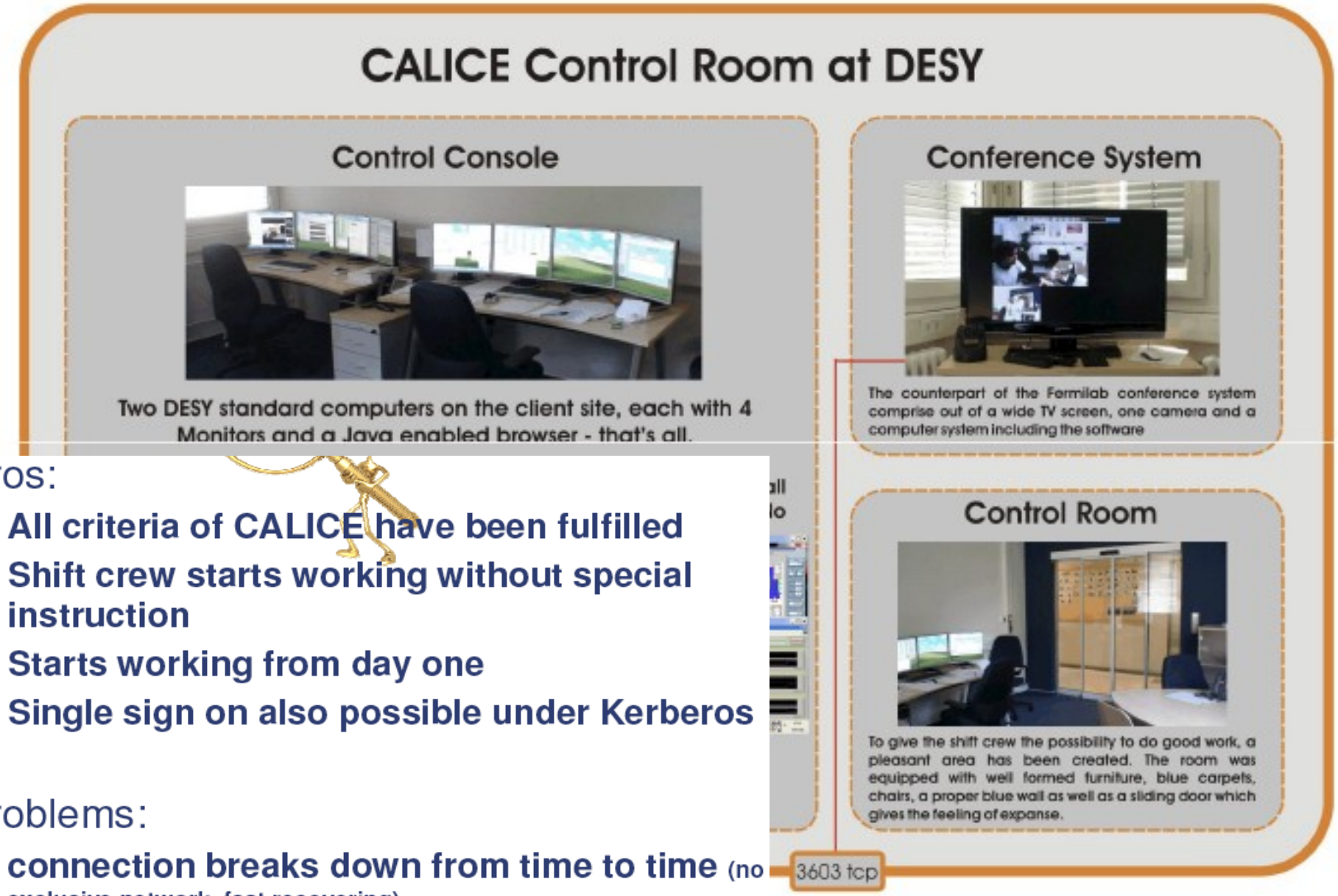
# realization of a remote control room for Calice



## Main Criteria for CALICE

- Web Based (no special software needed)
- Easy and fast to implement (<4 weeks)
- Easy to maintain
- Not too expensive (<10000 €)
- Nice to use (just start everywhere from everyone)






- Pros:
  - All criteria of CALICE have been fulfilled
  - Shift crew starts working without special instruction
  - Starts working from day one
  - Single sign on also possible under Kerberos
- Problems:
  - connection breaks down from time to time (no exclusive network, fast recovering)
  - video and audio quality must be better and faster

# now, near and far future

now:

- better communication quality
  (DFN, ES-net,…, H.323 Standard (Polycom, Mirial,) …)
- better picture quality
  (HD should become normal)

near future:

- better sound quality
  (much better echo cancellation, Dolby, …)
- more flexibility
  (faster wireless connections, movable stations, …)

and for the very far future

- better human feeling
  (smell; 3D; beam me up, Scotty ;-) )

# take home messages

- core software tools and testbeam software frameworks are in good shape – with some room for improvement,. e.g
  - LCIO I/O (performance)
  - LCCD conditions data bases
  - geometry and reconstruction
- what can be done depends on the manpower (AIDA)
- Grid computing is important for ILC - need to make Grid sites aware of that
- communication tools (virtual control rooms) are now available and work and can save considerable cost