Report from the Software Session

Steve Aplin
DESY

EUDET Annual Meeting ’09 Geneva
20th October 2009

ANLYS

The ANLYS task comprises the development of a common data analysis and simulation infrastructure. It sub-divides into:

- Development of a software framework using modern software technology to exchange test beam data and software for common analysis and comparison of measurements;
- Development of a software framework for the simulation of test beam experiment needed for the interpretation of the measurements;
- The creation of a repository for experimental and simulation data;
- Embedding into existing GRID infrastructure to allow easy exchange of data and a transparent exploitation of other available computing resources.

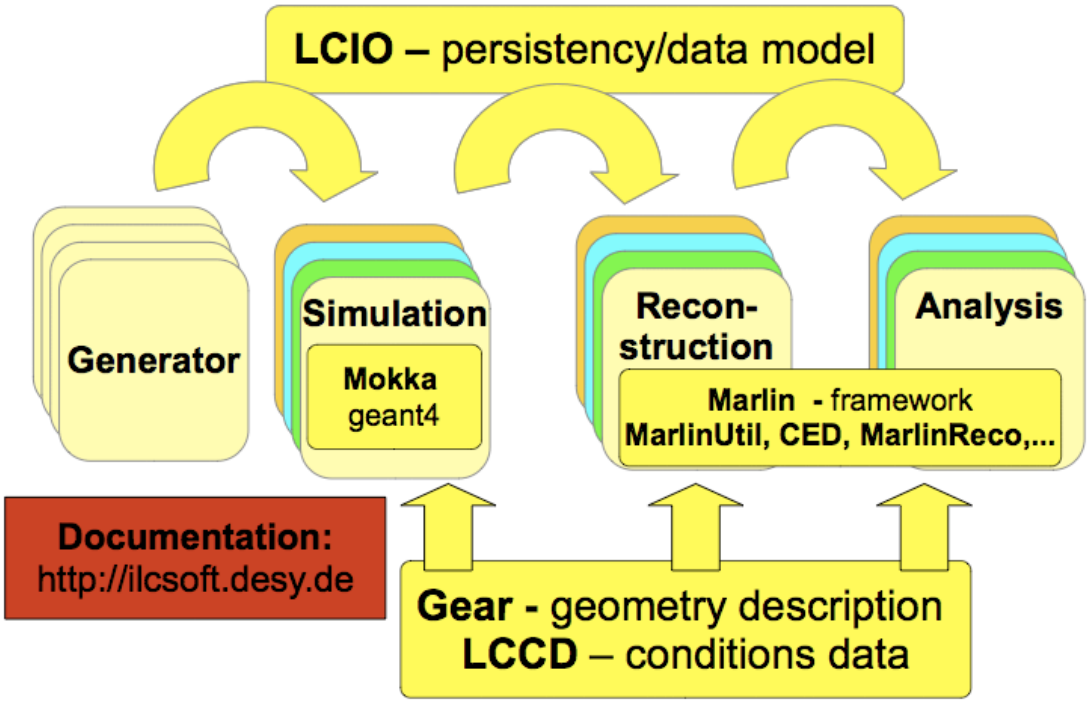
Session Overview

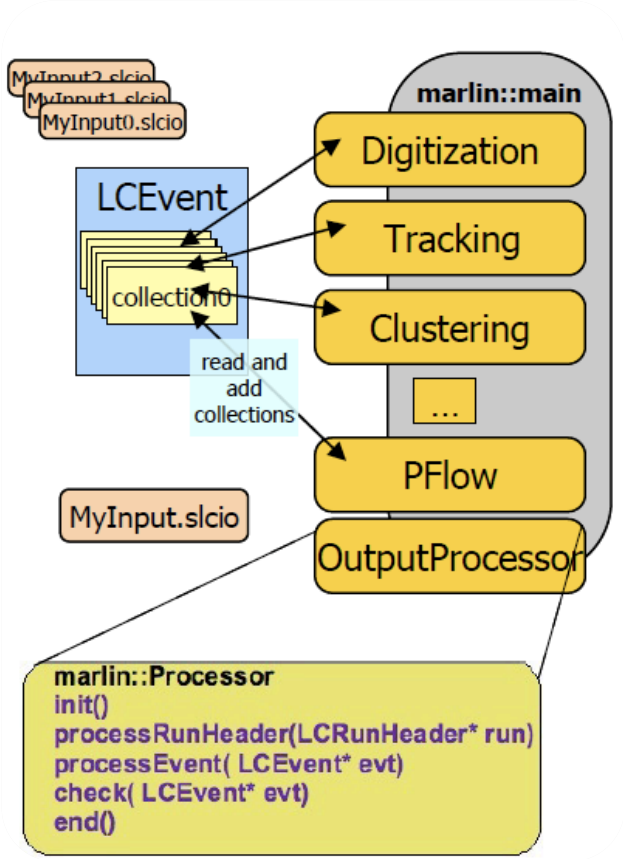
Contribution List | Time Table

Monday, 19 October 2009

14:00	[89] **Status and Plans for Core Software Tools** by Dr. Steve APLIN (DESY) (Salle SC1 102: 14:00 - 14:20)	slides
	[86] **Software for TPC R&D - A status report of the EUDET JRA2 activties** by Dr. Martin KILLENBERG (Univ. Bonn) (Salle SC1 102: 14:20 - 14:40)	slides
	[87] **Software Status Report of the EUDET JRA3 activties** by Dr. Roman POESCHL (LAL) (Salle SC1 102: 14:40 - 15:00)	slides
15:00	[88] **Proposal for an LCIO format for the DHCALs** by Dr. Vincent BOUDRY (LLR) (Salle SC1 102: 15:00 - 15:10)	slides
	[90] **Towards an Abstract System for Massive Data Processing** by Mr. Jan ENGELS (DESY) (Salle SC1 102: 15:10 - 15:30)	slides

ILCSoft – Overview

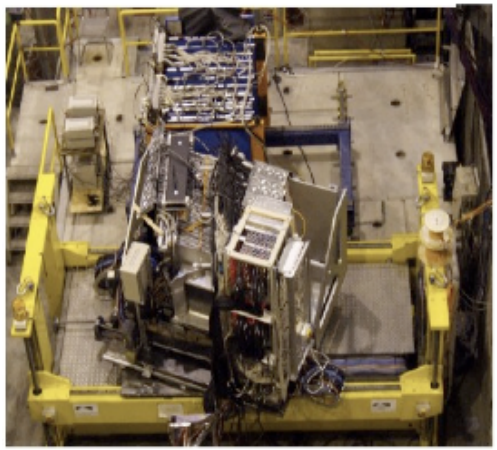


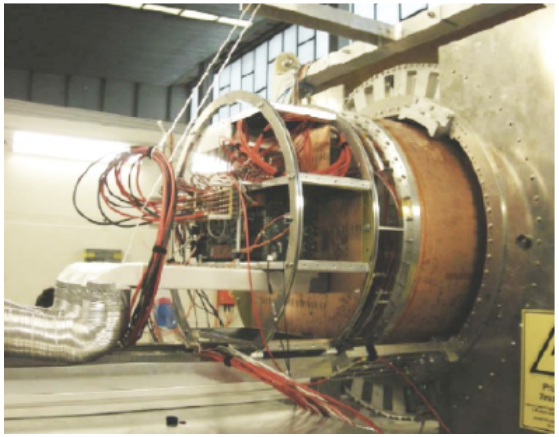


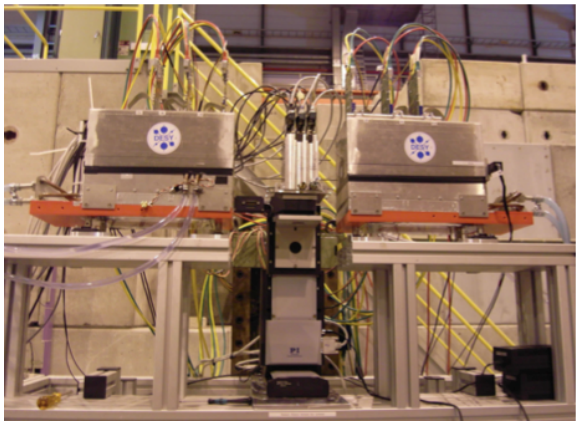
ILCSoft – Present Status

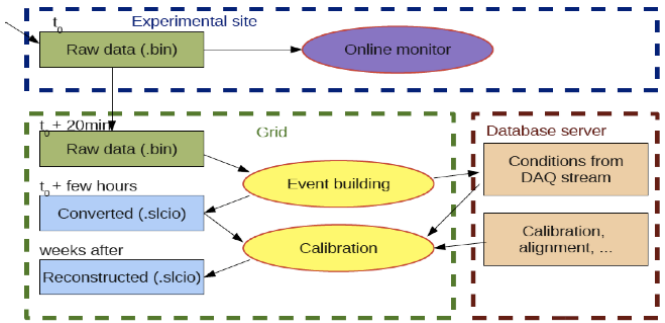
- First version of the common data analysis and simulation framework was completed after 21 months (2007). Fully adopted within all three JRA's.
- Continued support: minor feature development, bug fixing, etc.
- Last year spent working almost exclusively on LOI production and studies.
- The common data analysis and simulation framework was adopted for the ILD LOI studies with over 60 million events simulated and reconstructed (v01-06)

ILCSoft – With Real Data

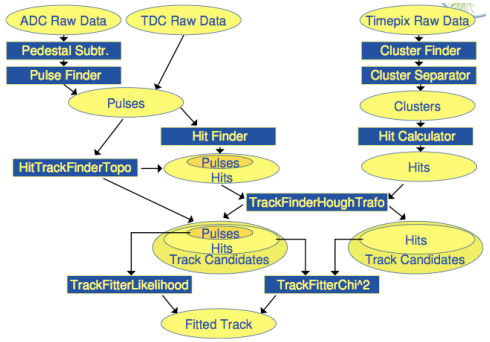




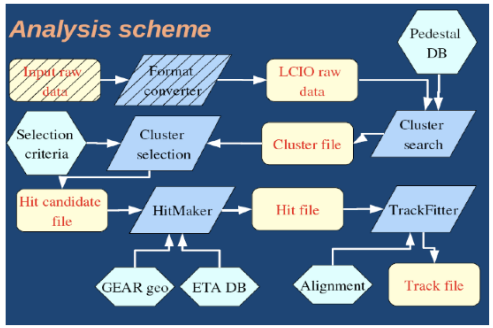




CALICE

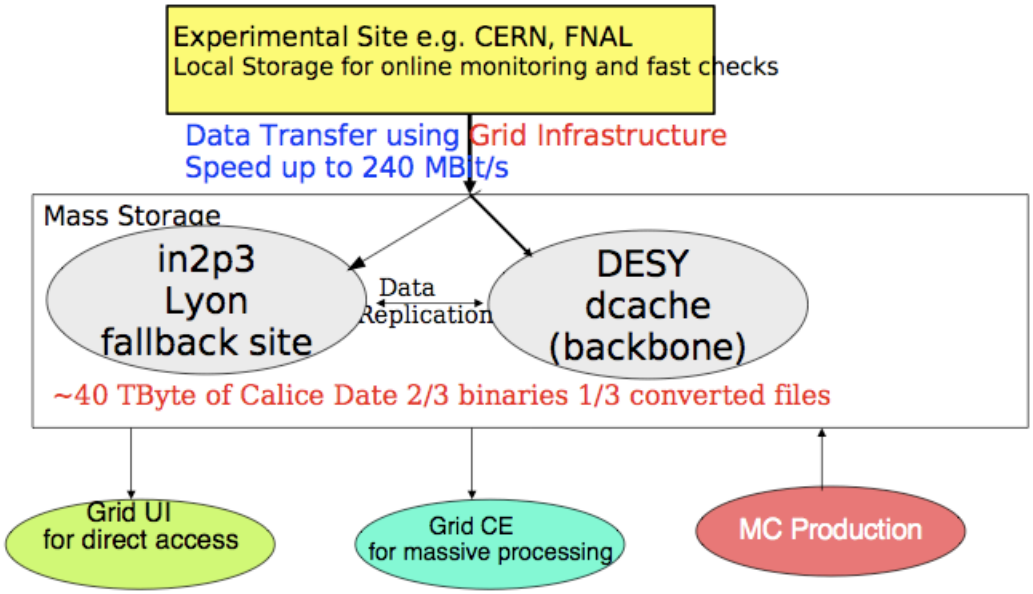


LC-TPC



EUTelescope

ILCSoft – With Real Data



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

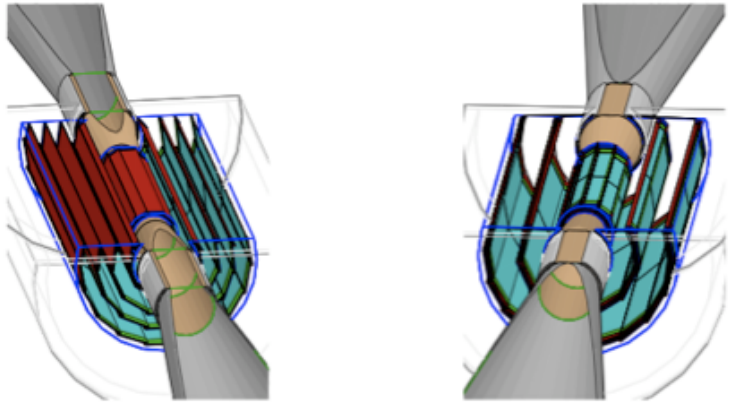
24h/24h 7h/7h during CERN, FNAL testbeams 2006-2009

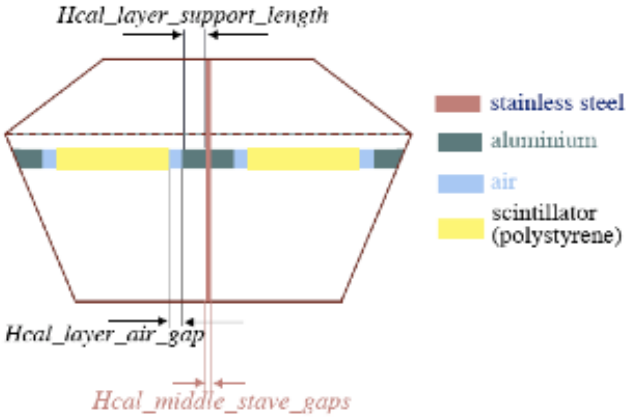
Experience with testbeam data clearly reveals the needs for a coherent concept to handle 'low level' data within ILC Software

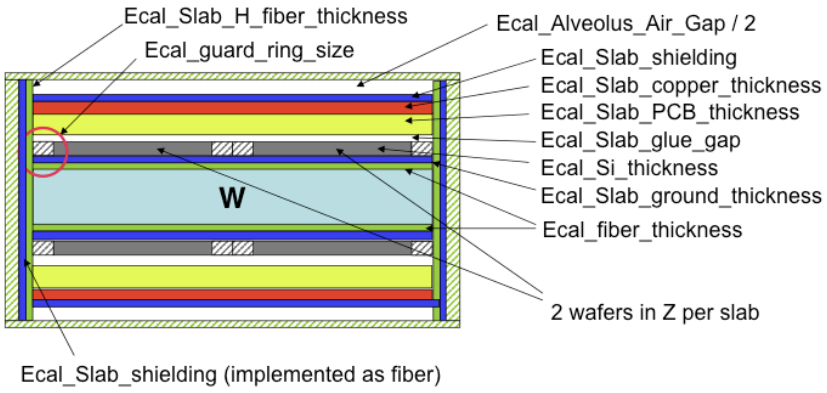
R. Poesch

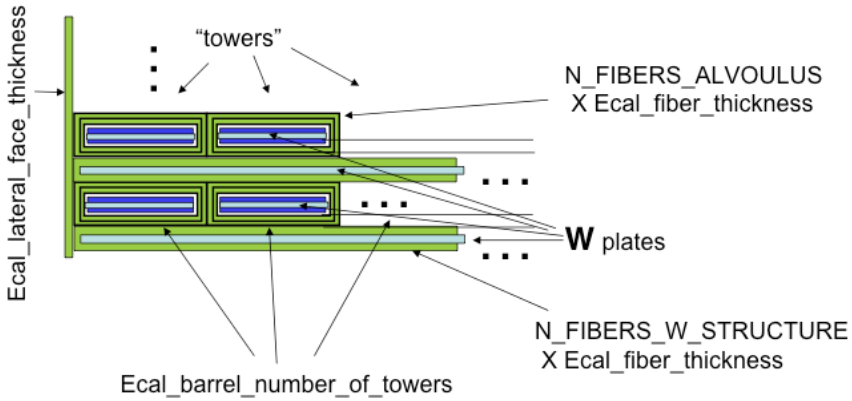
Simulation – Mokka

A lot of effort made to increase the engineering detail for the LOI studies. This will need to be increased for the TDR, meaning that it is vital to ensure a continuation of the good support from the R+D communities.

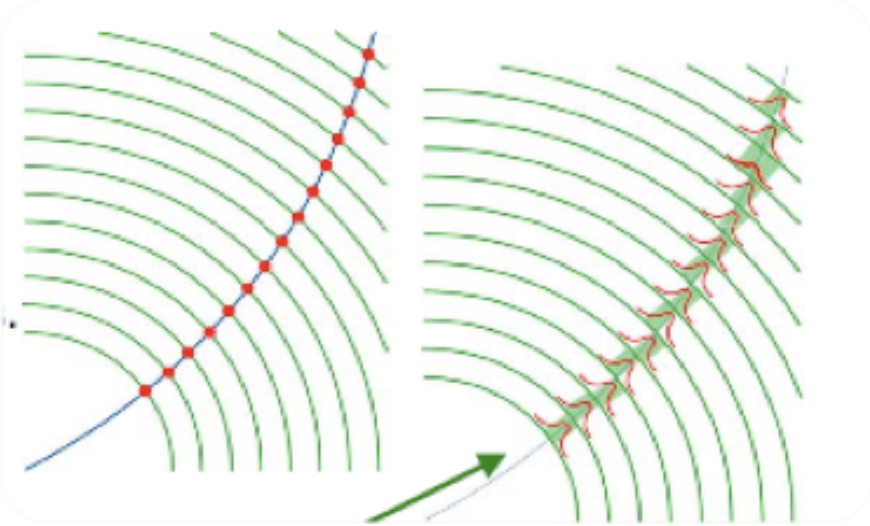


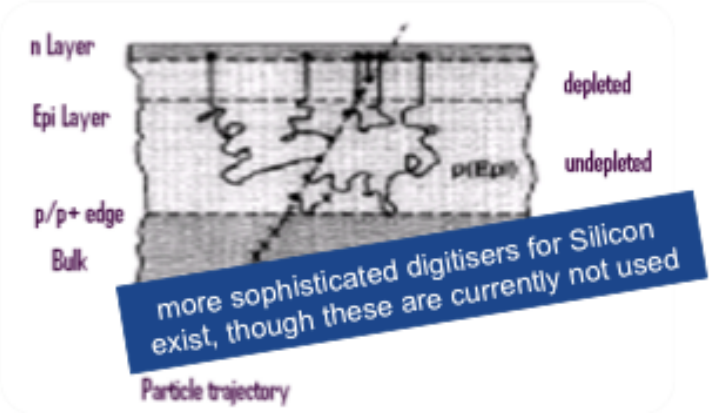






Digitisation within Mass Simulation

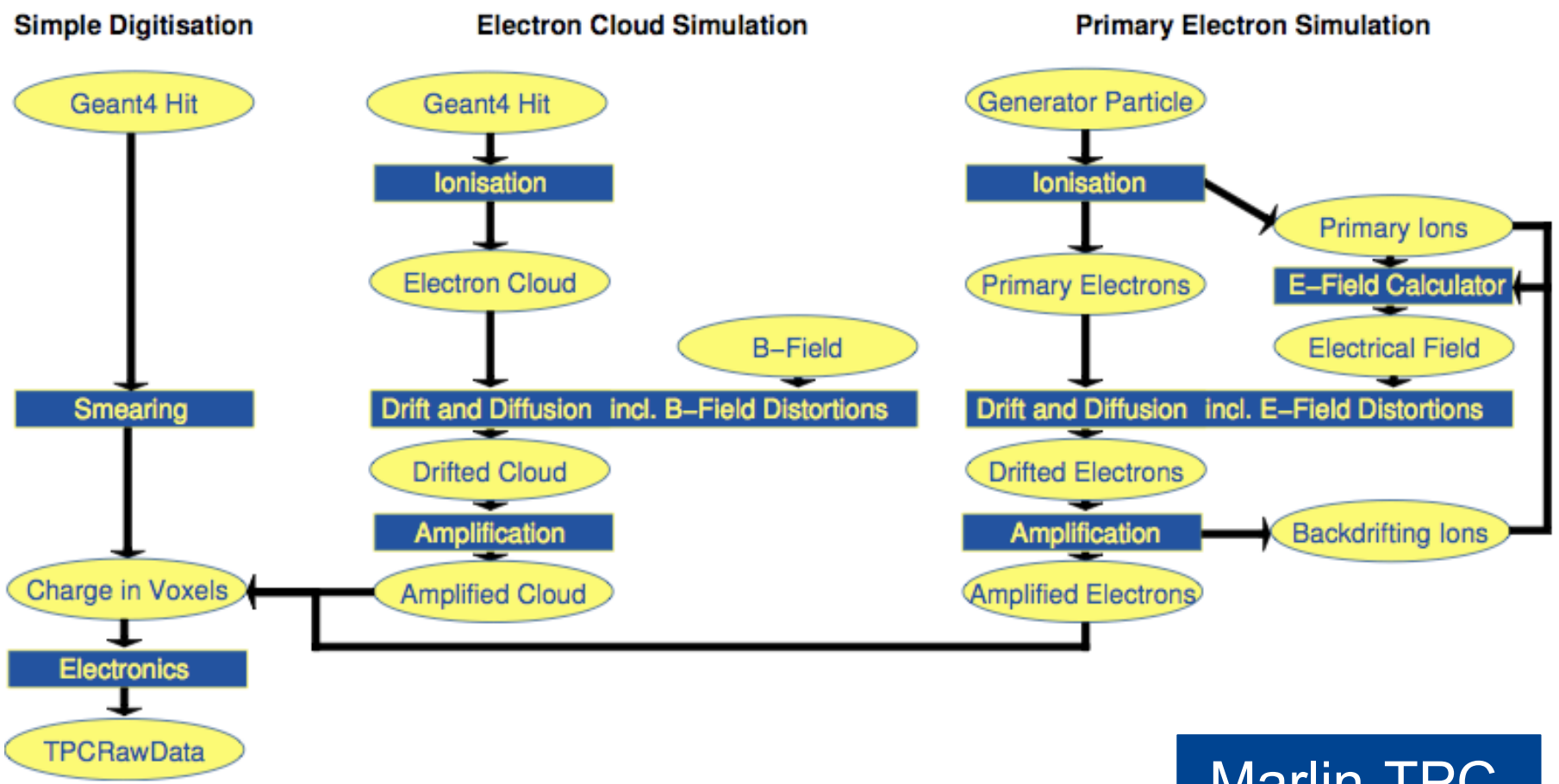




	$\sigma_{r-\phi}/\mu$m	σ_z/μm		$\sigma_{r-\phi}/\mu$m	σ_z/μm
VTX	2.8	2.8	FTD	5.8	5.8
SIT/SET	7.0	50.0	ETD	7.0	7.0
TPC	$\sigma^2_{r\phi} = 50^2 + 900^2 \sin^2\phi + \left((25^2/22) \times (4/B)^2 \sin\theta\right) z\,\mu\text{m}^2$ $\sigma^2_z = 40^2 + 8^2 \times z\,\mu\text{m}^2$				

… are we going to need more sophisticated digitisation for the Calorimeters?

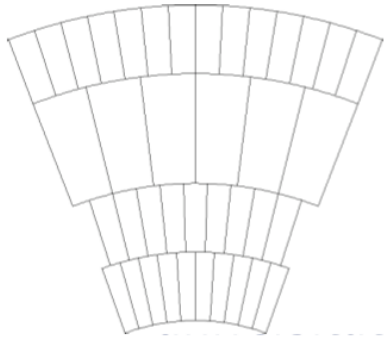
Digitisation within Testbeams



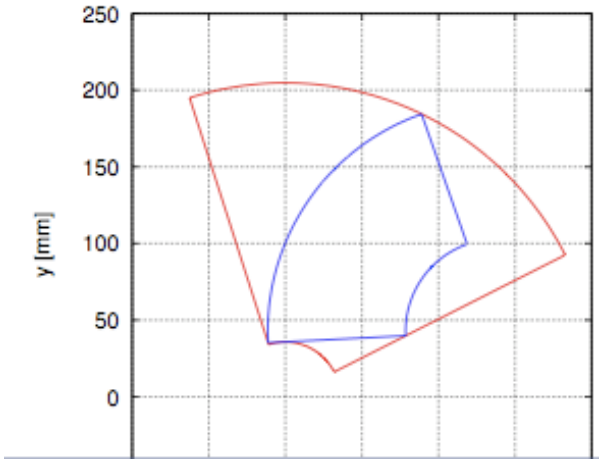
Marlin-TPC

GEAR Extension

- Existing TPC Padplane designs have variable pad sizes
  - New: FixedPadAngleDiskLayout
  - New: VersatileDiskRowLayout



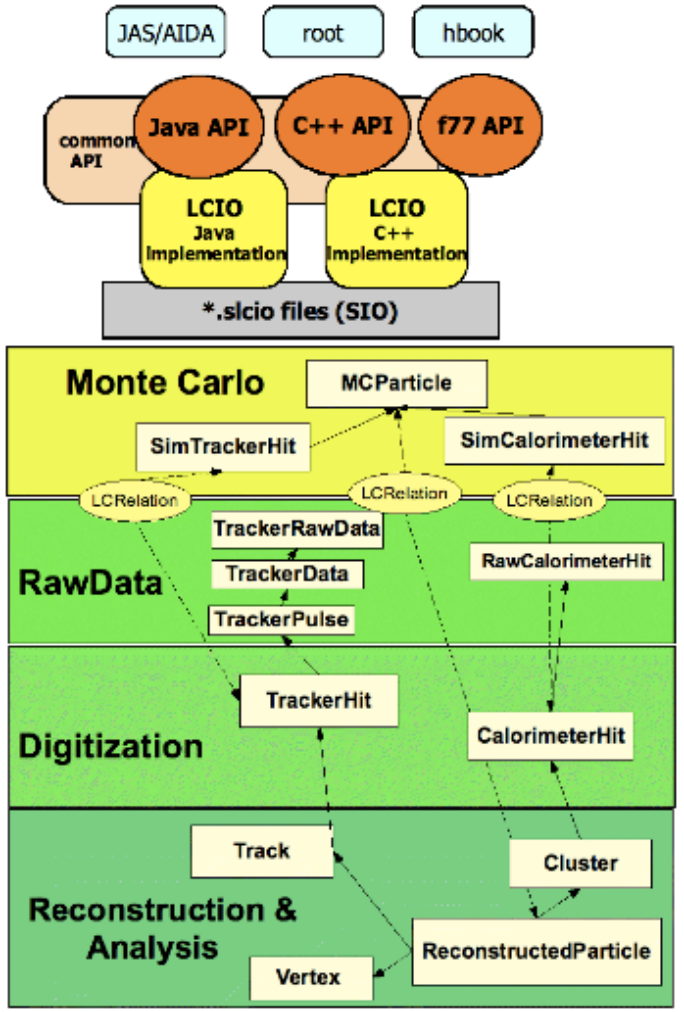
- A realistic TPC endplate (EUDETLP) consists of multiple modules
  - Created New TPCModule derived from PadRowLayout2D
  - full backward compatibility and transparency in user code



M. Killenberg

LCIO

- Joint DESY – SLAC Project
- Provides persistency (I/O) and an event data model (EDM) to ILC detector R&D community
  - object I/O (w/ pointer chasing)
  - schema evolution
  - compressed records
  - hierarchical data model
  - decoupled from I/O by interfaces C++, Java (and Fortran)
  - some generic user object I/O
- Used by ILD, SID, CALICE, EUPixelTelescope, LCTPC



LCIOv2?

- further improve LCIO -> LCIOv2?
- event data model
  - 1d, 2d hits
  - Track class – multiple fits per track
- Improve I/O
  - splitting of files
  - direct access
  - partial reading of events
- Investigate the use of ROOT with LCIO
  - LCEvent in ROOT macros
  - look into optional ROOT I/O for LCIO

Important to continue successful horizontal collaboration with SID on LCIO

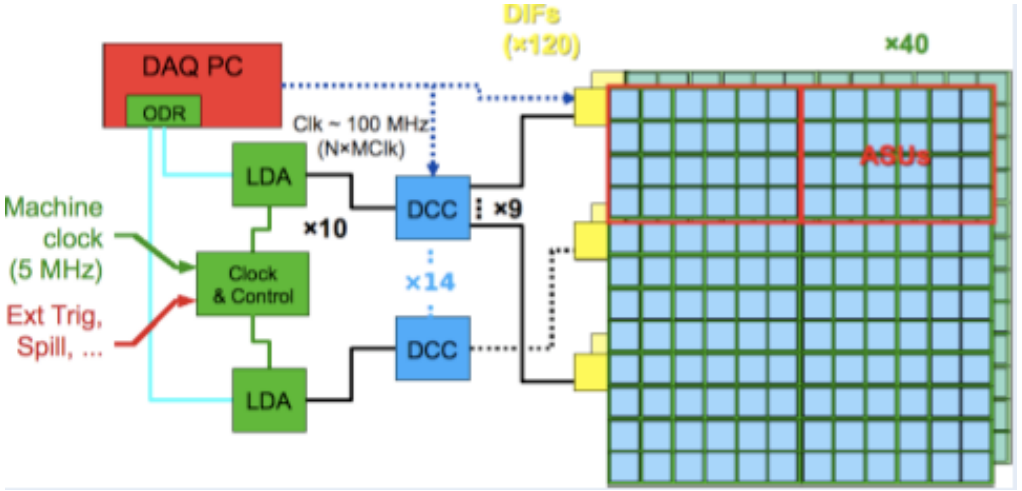
LCIO I/O

- started to investigate optional ROOT I/O for LCIO
- created dictionary with rootcint for LCIO classes
  - thanks to ROOT team for their help and for adding some features to ROOT 5.24.00 needed for LCIO
- write and read LCEvents transparently to/from ROOT files
  - no change in user code
- use LCEvents in ROOT macros
  - rapid development of analysis code based with LCIO in ROOT
- issues:
  - no branches due to pointers between object
  - no partial reading and splitting of events over files
  - need proper interface to ROOT I/O for java implementation

LCIO – Extention

V. Boudry

Store EUDET DAQ2 raw data originating from a DHCAL using ROC chips providing frames



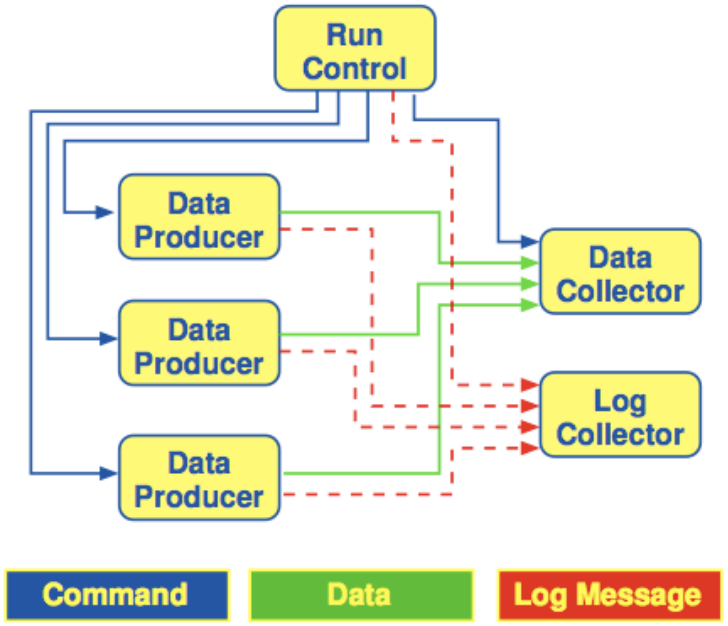

```
* Event:                                                                    V. Boudry, R. Cornat,
*  - DetectorName <- runHdr::DetectorName                                   D. Decotigny
*  - RunNumber    <- runHdr::RunNumber
*  - EventNumber  <- ROEventId::toUnsigned()
*  - TimeStamp: default
*  - Weight: default
*  - LCCollection "rawdata_DIF:str(DIF_id)":
*          + param "id_ODR" (unsigned): from raw event
*          + param "id_LDA" (u32b)
*          + param "id_LDA_diflink" (unsigned)
*          + param "id_DIF" (unsigned)
*          + param "TrainNumber" (unsigned)      // since DAQ start
*          + param "BC_DIF" (unsigned)           // since RUN start
*          + param "DeltaCounter" (unsigned)     // 40 MHz counter
*          + param optional extra DIF info (type, temp, adc) ???
*          + elt[0]: LCGenericObject
*              + int[0]: id_roc_chain
*              + int[1]: id_roc
*              + int[2]: index_in_dif_dump (starts at 0)
*              + int[3]: chipType
*              + int[4]: acqMode
*              + For HR2 chips:
*                 + int[5]: nframes (< 128)
*                 + int[6 + i*5]: bcid for frame i (i in [0, nframes])
*                 + int[7 and 8 + i*5]: t0 vector (msb 7=t0_63, lsb 8=t0_0)
*                 + int[9 and 10 + i*5]: t1 vector (msb 9=t0_63, lsb 10=t0_0)
*                 + int[11 + i *5...]: additional data ??
*        + elt[1]: other LCGenericObject (ROC event)
*          ...
*  - other LCCollection "rawdata_DIF:str(DIF_id)":
*    ....
```

```
EVENT::RawCalorimeterHit                                          ILD and TB compatible
{
    int  _cellID0;          // Chan (64 ==> 6b)
                            // + Asic (max 420 ==> 9-10b)
                            // + Dif_Id (48-144 ==> 7-8b)
                            // + Module_Id (40 Barrel + 24 Endcap ==> 6b)
                            // == 28-30b (remain 2-4b)  [6+6+7 == 19b in TB]
     int _cellID1;          // Time2Previous (in BC) ==> 24b (remain 8b) (CHBIT_ID1 must be set)
    int  _amplitude;        // 3 Thr ==> 2b (remains 30)
    int  _timeStamp;        // Rec Time on 32b wrt (Spill start | Ext. Trigger)
}
```

Very first draft for discussion, being implemented for SDHCAL offline storage. Then needs to be finalised and validated.
Generally require and improvement of online writing with LCIO. Flush + crash recovery.

LCIO Plugin for EUDAQ

- For the Data Collector a new LCIO Plugin mechanism has been developed.
- The producer sends raw data, not LCIO.
- The data collector has to know how to convert to LCIO ⇒ Plugin
- DAQ developer should provide plugin together with the producer



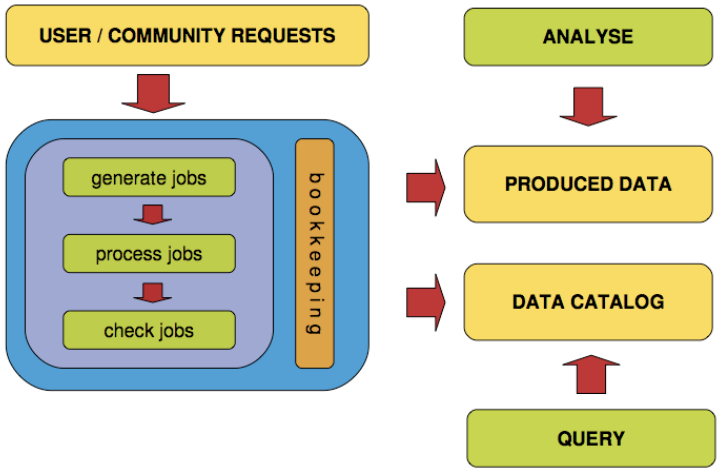
DAQ	Producer	Converter Plugin
ALTRO for LP	×	✓ [1]
ALTRO USB (Bonn)	✓	✓
AFTER	×	×
TDCs	×	×
Timepix	✓	✓

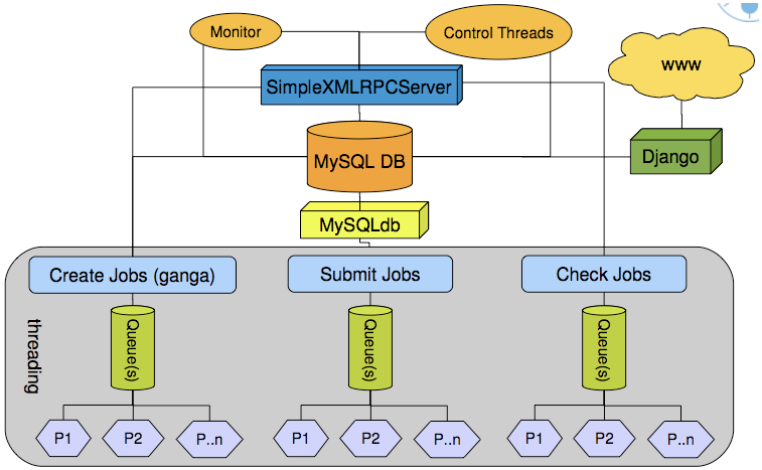
M. Killenberg

Mass Production System

J. Engels

- Prototype for a production system to effectively utilise large scale computing resources for mass production
- Abstraction layer between user and data model.
- Make use of existing tools such as GANGA and couple them with a carefully designed datamodel in a Database
- Implementation using modern standard tools, python, MYSQL, django etc.





Summary

- The common software tools developed have been used to produce an unprecedented amount of data at such an early stage
- The common adoption of the tools across the different R+D communities has maximised the limited resources available for software development and has produced valuable continuity
- The feedback into the development of the common software tools from the R+D communities is extremely welcome