NA2 Status Computing and Analysis

Frank Gaede
DESY
EUDET – Annual Meeting,
MPI Munich
October 18-20, 2006

Outline

- NA2 – tasks: COMP + ANALYS
  - objectives
  - contributers and activities
- Core software tools
  - LCIO
  - Marlin
  - LCCD
  - GEAR
- Usage of software for testbeams:
  - JRA2
  - JRA3
- Grid
- Summary

tasks COMP + ANALYS

- **COMP: setup of a high performance dedicated computer cluster for the common data analysis and simulation using grid technology**
  - -> set up the clusters during the first three years

- **ANALYS: development of a common data analysis and simulation infrastructure**
  - development of a software framework for:
    - analysis and comparison of testbeam measurements
    - simulation of test beam experiments
    - -> have “version 1.0” of framework after 18 months
  - creation of a repository for experimental and simulation data
  - embedding into existing GRID infrastructure

(from annex1)

Contributors for task COMP

Participating Institutes

- Tel Aviv University, Israel
- DESY, Germany
- University of Bonn, Germany

Budget (EU contribution)

	ppm	consumables (kEUR)
DESY	10	0
TAU	0	10
U-Bonn	8	30

(P.Wienemann)

activities and spending - COMP

Tel Aviv University

- TAU grid site supports VO ilc now
- In process of purchasing 5 dual core machines with 3 TB disk storage for approx. 10 k€
- Will be included in grid cluster, ilc will have highest priority on these machines
- Plans to add usage of 28 PCs outside teaching hours by end of October, more will probably follow
- Pay grid admin for EUDET related work
- In total: 22 CPUs, 105 GB storage (status: Sept. 05, 2006)

(P.Wienemann)

activities and spending - COMP

DESY

- Hosts VO ilc and calice
- Bought 6 SunFire X4100 with 4 CPUs for 23 010 € from EUDET money in May
- Remaining 6 990 € go into two SunFire X5400 file servers (partly dedicated to ilc) with 24 TB disk storage each. Cost: 28 k€ per server.
- Resources in 2006: 55890 CPU days available, 21372 CPU days used, 1889 CPU days used by ilc (9 %), 9.6 TB used by ilc

(P.Wienemann)

activities and spending - COMP

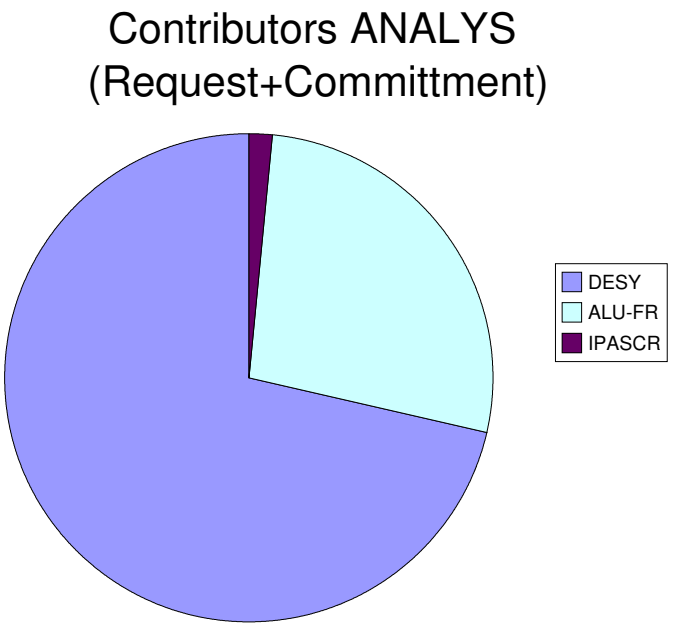
University of Freiburg/Bonn

- We're just in the process of moving.
- ILC VO supported at Freiburg (100 CPU, ~10TB) (shared with ATLAS and local VOs)
- 50% of Computer Hardware (15kEUR) will be bought still in 2006.
- postdoc starts working on NA2 (Comp+Analys) starting (presumably) November 06.

(P.Wienemann)

Contributors for task ANALYS

	DESY	ALU-FR	IPASCR	TOTAL
REQUEST				
Perm Staff ppm				
Temp Staff ppm	12.000	8.000		20.000
Perm Staff Cost kEUR				
Temp Staff Cost kEUR	62.500	46.875		109.375
Travels kEUR	1.300	0.867		2.167
Consumables kEUR				
Overheads kEUR	12.760	9.548		22.308
Total Manpower ppm	12.000	8.000		20.000
Total Cost kEUR	76.560	57.290		133.850
COMMITTMENT				
Perm Staff ppm	12.000		3.000	15.000
Temp Staff ppm				
Perm Staff Cost kEUR	62.500		9.000	71.500
Temp Staff Cost kEUR				
Travels kEUR				
Consumables kEUR				
Overheads kEUR	12.500		1.800	14.300
Total Manpower ppm	12.000		3.000	15.000
Total Cost kEUR	75.000		10.800	85.800
TOTAL BUDGET				
Perm Staff ppm	12.000		3.000	15.000
Temp Staff ppm	12.000	8.000		20.000
Perm Staff Cost kEUR	62.500		9.000	71.500
Temp Staff Cost kEUR	62.500	46.875		109.375
Travels kEUR	1.300	0.867		2.167
Consumables kEUR				
Overheads kEUR	25.260	9.548	1.800	36.608
Total Manpower ppm	**24.000**	**8.000**	**3.000**	**35.000**
Total Cost kEUR	**151.560**	**57.290**	**10.800**	**219.650**



ALU-FR now U-Bonn

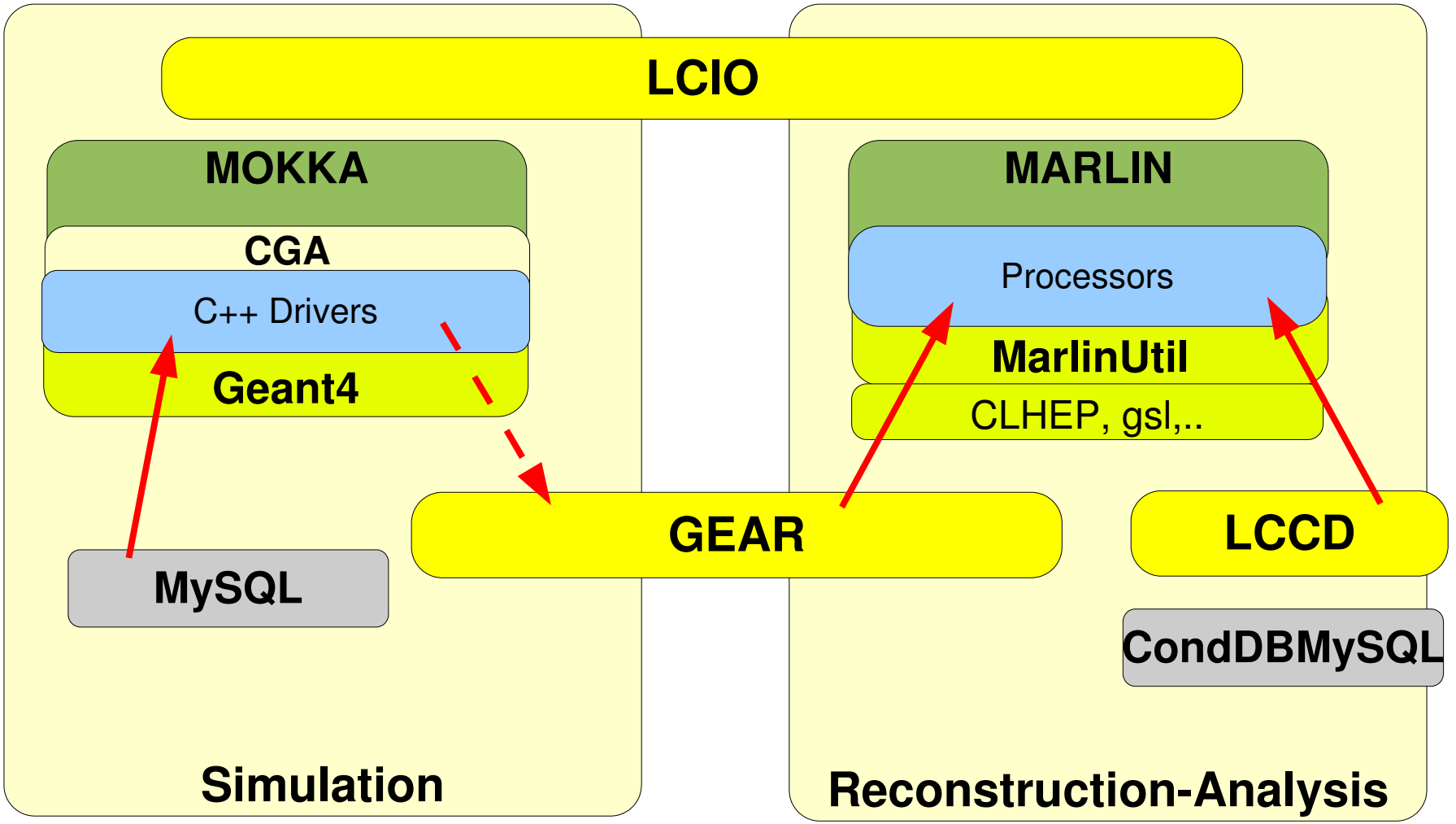
Usage of budget - ANALYS

- DESY
  - commitment 12ppm: F.Gaede 25% for full project length
  - 12ppm (scientist) converted to hire a programmer for 18 month
    - **started August 2006**
    - (possibly extend position with other funding sources)
- RFWU-Bonn (ALU-FR) (K.Desch, P. Wienemann)
  - request: 8ppm (scientist): plan to combine with funds (8ppm) from COMP to hire a postdoc that works part-time on COMP and ANALYS
    - will start in October (Nov.) 2006
- IPASCR (J.Cvach)
  - commitment: 3ppm: PhD student that works
  - part time on calorimeter simulation with geant4
    - not yet

General strategy for ANALYS

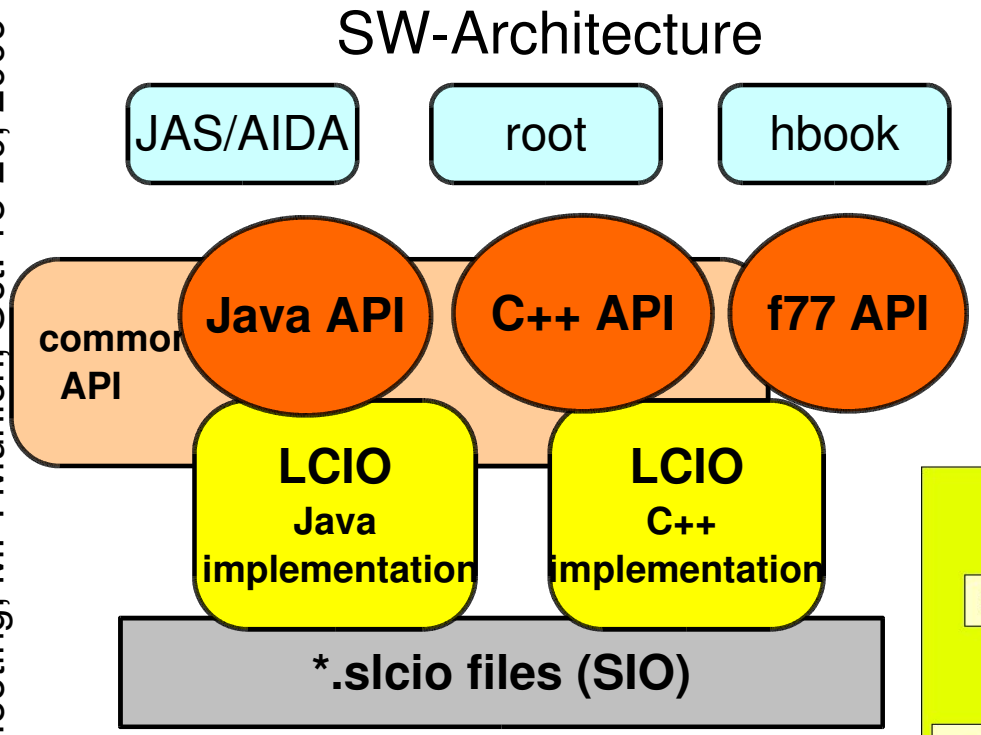
- there will be no EUDET/testbeam specific simulation and analysis software framework !
  - avoiding of double work
  - a lot of what's needed already exists
- the testbeam software effort is tightly integrated with the overall common ILC/LDC software effort !
  - implement tools and functionality specific to testbeams
  - benefit from synergies where possible, e.g. use geant4 application for full detector also for testbeam (Mokka/Calice)
- same for grid tasks: integrate with common ILC grid activities

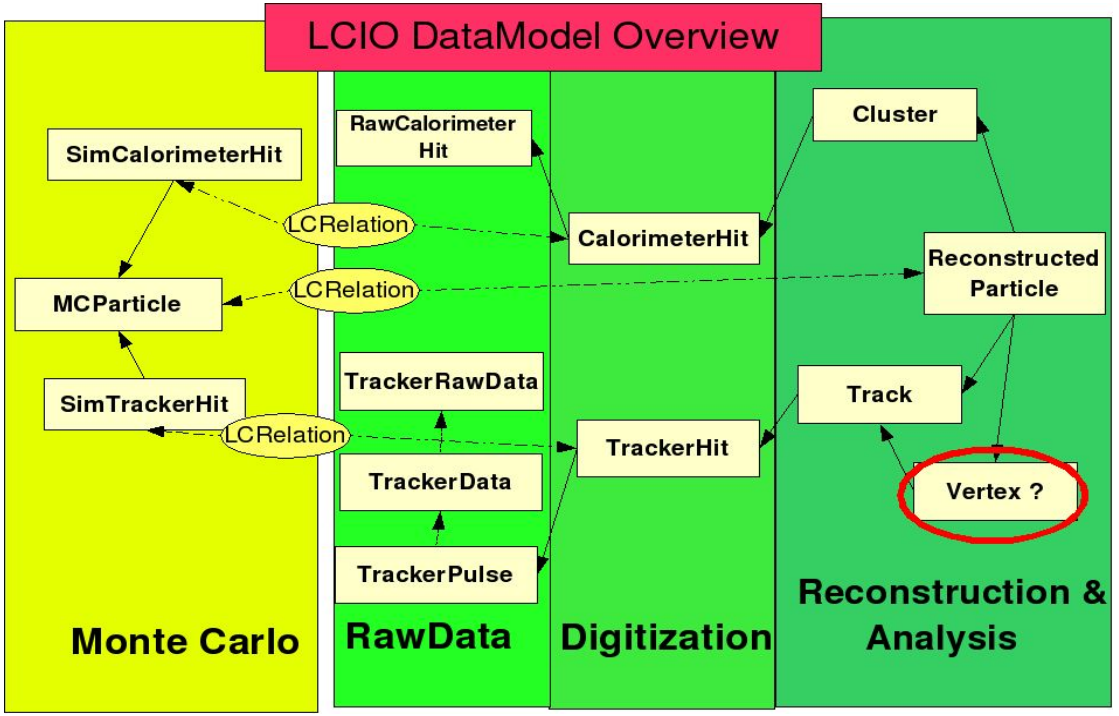
ILC-LDC software framework



all tools are also used in testbeam programs

LCIO overview





- standard persistency & datamodel for ILC
- used in all three regions

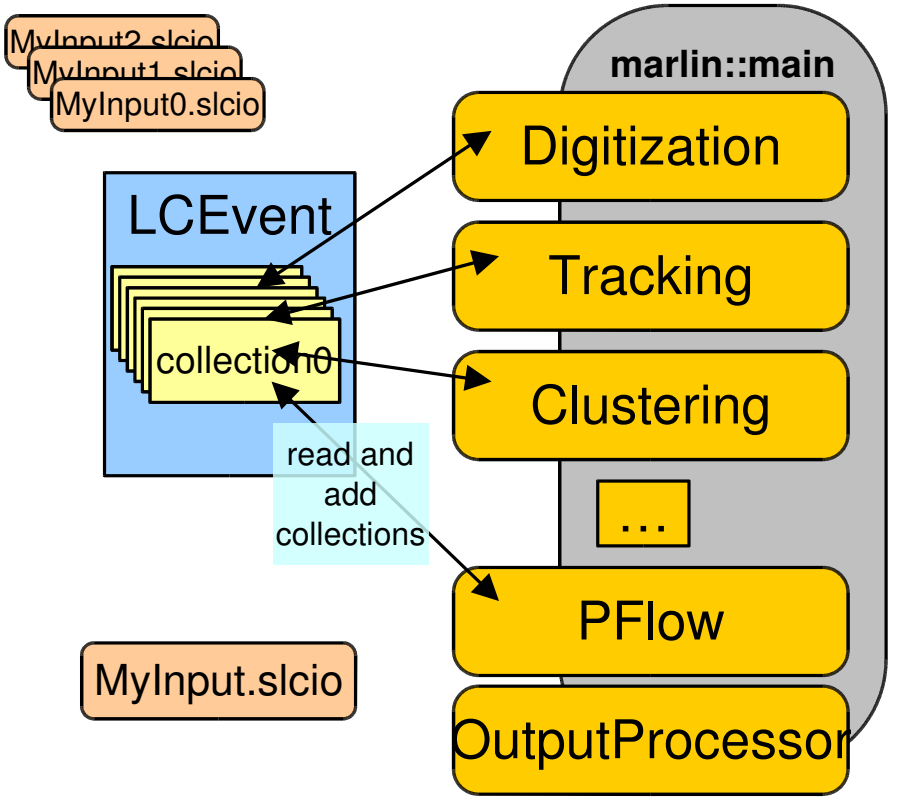
LCIO Event Data Model II

- the LCIO event data model is fairly complete and flexible
- however it is adapted and extended as needed by the community
  - maintaining downward compatibility
  - with international discussion and agreement
  - example: introduction of a new **Vertex** class in LCIO
    - originally proposed by LCFI group
      - see discussion @ http://forum.linearcollider.org/
    - test release v01-08-vtx
  - new raw data classes for prototypes
    - TPC uses TrackerRawData, TrackerData, TrackerPulse
      - also to be used for vertex prototypes
    - calorimeter (calice) could use RawCalorimeterHit or new additional classes if needed

Marlin

Modular**A**nalysis & **R**econstruction for the **L I N**ear Collider

- modular C++ **application framework** for the analysis and reconstruction of LCIO data
- uses LCIO as transient data model
- software modules called Processors
- provides main program !
- provides simple user steering:
  - program flow (active processors)
  - user defined variables
    - per processor and global
  - input/output files
  - **Plug&Play** of processors



Marlin – XML steering files

```
- <marlin>
  - <execute>
      <processor name="MyAIDAProcessor"/>
      <processor name="MyEventSelection"/>
    - <if condition="MyEventSelection">
        <group name="Tracking"/>
        <processor name="MyClustering"/>
        <processor name="MyPFlow"/>
        <processor name="MyLCIOOutputProcessor"/>
      </if>
    </execute>
  - <global>
      <parameter name="LCIOInputFiles"> simjob.slcio </parameter>
      <parameter name="MaxRecordNumber" value="5001"/>
      <parameter name="SupressCheck" value="false"/>
    </global>
  - <processor name="MyLCIOOutputProcessor" type="LCIOOutputProcessor">
      <parameter name="LCIOOutputFile" type="string">outputfile.slcio </parameter>
      <parameter name="LCIOWriteMode" type="string">WRITE_NEW</parameter>
    </processor>
  - <group name="Tracking">
      <parameter name="NTPCLayers" value="200"/>
      <processor name="MyTrackfinder" type="Trackfinder"/>
    - <processor name="MyTrackfitter" type="Trackfitter">
        <parameter name="Algorithm" value="DAF"/>
      </processor>
    </group>
    <!-- ... -->
  </marlin>
```

- Program flow defined in <execute>...</execute> section
- logical conditions from parameters evaluated at runtime

- global Parameters defined in <global/> section

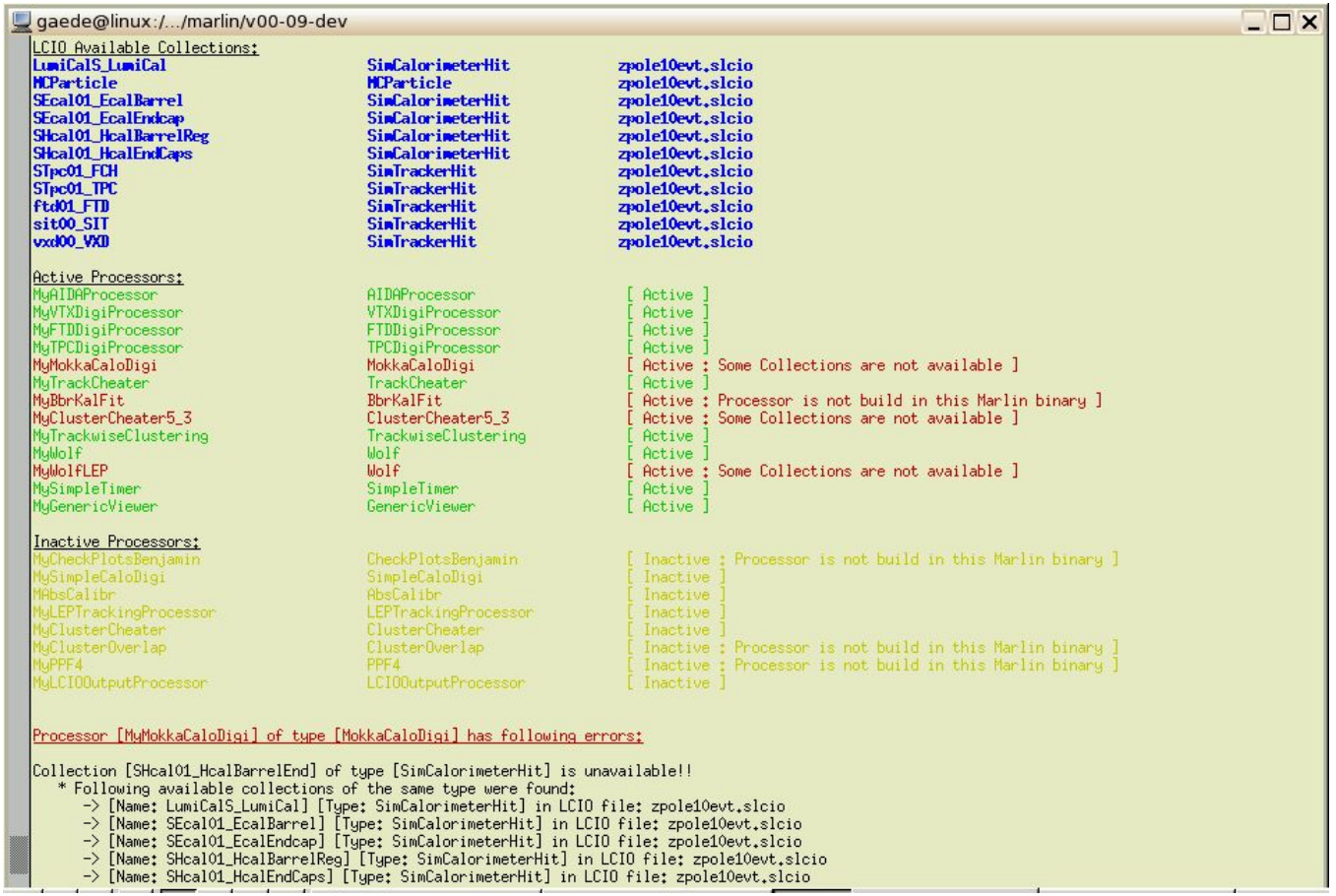
- local Parameters defined in mandatory <parameter/> section

- Processors can be enclosed by <group/> tag
- Parameters in <group/> joined by all processors

a Marlin application is fully configured through the steering files (no user main program) !!

Marlin new development

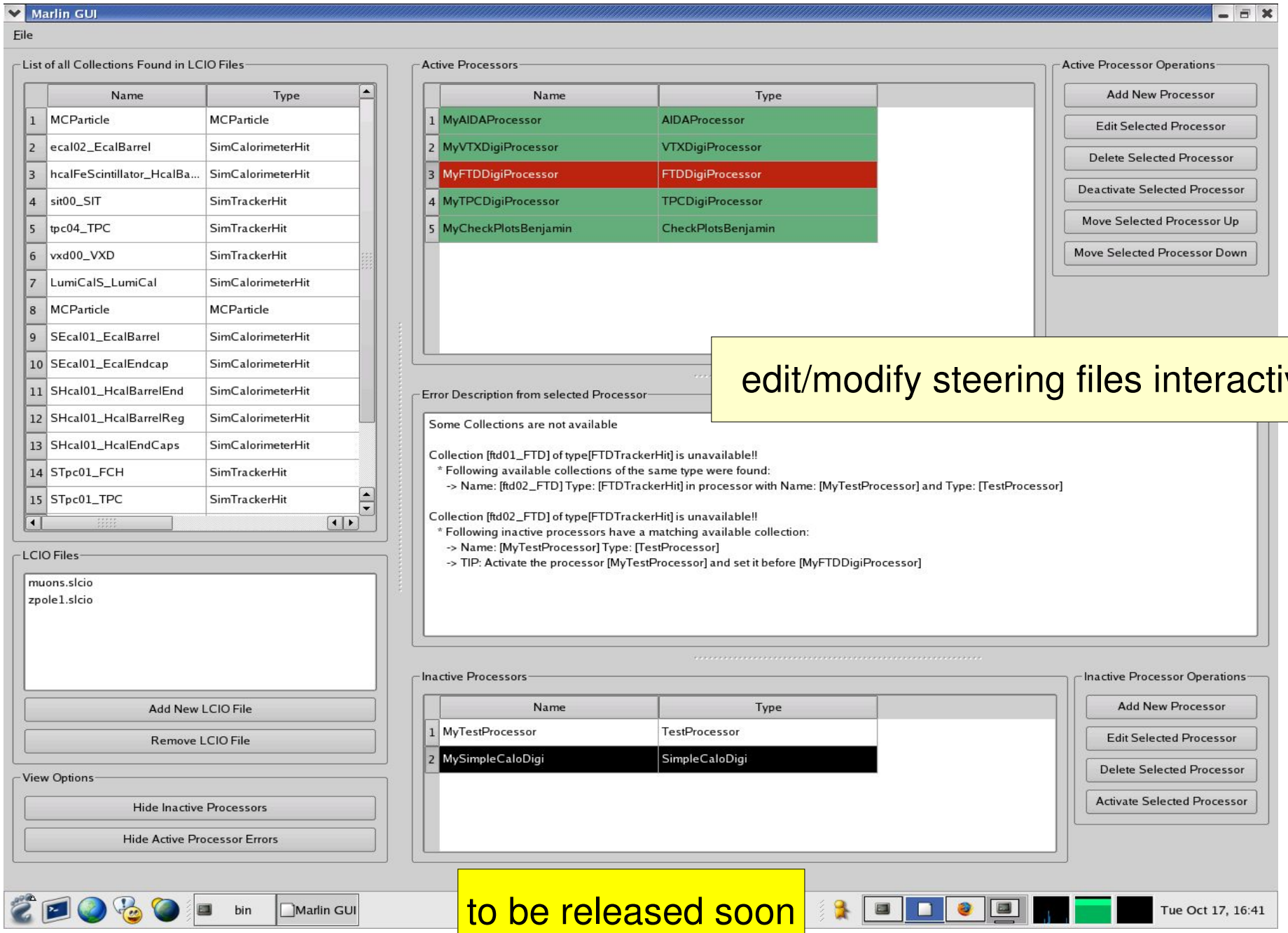
- user complaint:
  - marlin steering files are somewhat clumsy to edit
  - -> implement new feature to check consistency of steering files: Marlin -c steer.xml



J.Engels (EUDET)

to be released soon

new development: MarlinGUI



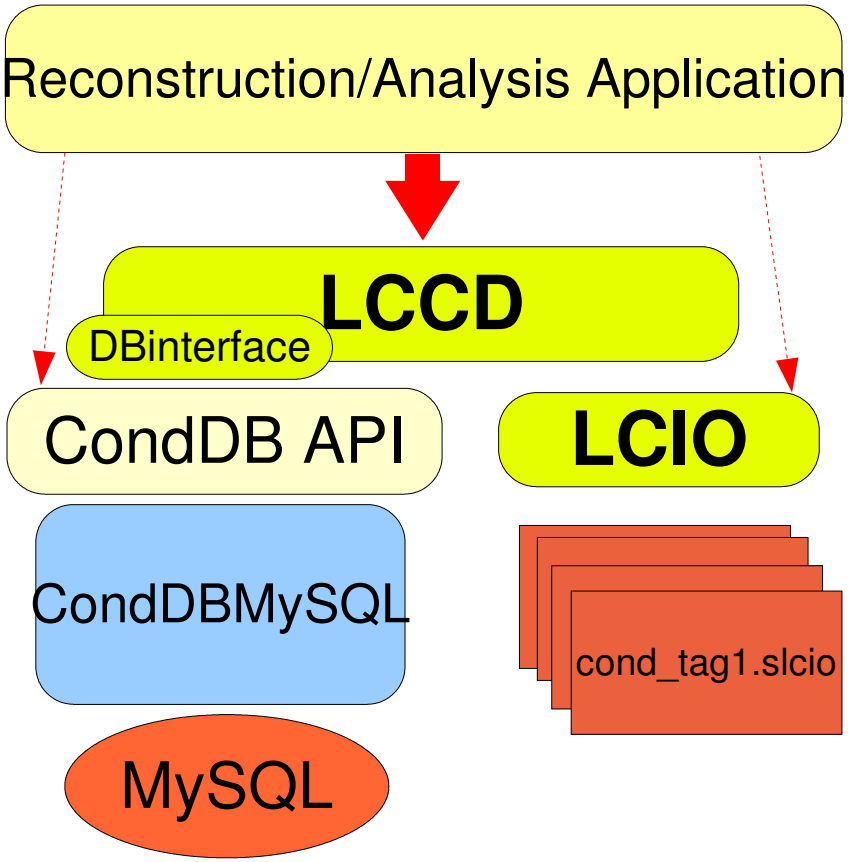
edit/modify steering files interactively

to be released soon

LCCD

Linear **C**ollider **C**onditions **D**ata Toolkit

- Reading conditions data
  - from conditions database
  - from simple LCIO file
  - from LCIO data stream
  - from dedicated LCIO-DB file
- Writing conditions data
- tag conditions data
- Browse the conditions database
  - through creation of LCIO files
    - vertically (all versions for timestamp)
    - horizontally (all versions for tag)



LCCD is used by Calice for the conditions data of the ongoing testbeam studies

Gear

GEometry **A**PI for **R**econstruction

```
- <gear>
  - <!--
      Example XML file for GEAR describing the LDC detector
    -->
  - <detectors>
    - <detector id="0" name="TPCTest" geartype="TPCParameters" type
        <maxDriftLength value="2500."/>
        <driftVelocity value=""/>
        <readoutFrequency value="10"/>
        <PadRowLayout2D type="FixedPadSizeDiskLayout" rMin="386.0"
        maxRow="200" padGap="0.0"/>
        <parameter name="tpcRPhiResMax" type="double"> 0.16 </para
        <parameter name="tpcZRes" type="double"> 1.0 </parameter>
        <parameter name="tpcPixRP" type="double"> 1.0 </parameter>
        <parameter name="tpcPixZ" type="double"> 1.4 </parameter>
        <parameter name="tpcIonPotential" type="double"> 0.00000003
      </detector>
    - <detector name="EcalBarrel" geartype="CalorimeterParameters">
        <layout type="Barrel" symmetry="8" phi0="0.0"/>
        <dimensions inner_r="1698.85" outer_z="2750.0"/>
        <layer repeat="30" thickness="3.9" absorberThickness="2.5"/>
        <layer repeat="10" thickness="6.7" absorberThickness="5.3"/>
      </detector>
    - <detector name="EcalEndcap" geartype="CalorimeterParameters">
        <layout type="Endcap" symmetry="2" phi0="0.0"/>
        <dimensions inner_r="320.0" outer_r="1882.85" inner_z="2820.
        <layer repeat="30" thickness="3.9" absorberThickness="2.5"/>
        <layer repeat="10" thickness="6.7" absorberThickness="5.3"/>
      </detector>
    </detectors>
  </gear>
```

compatible with US – compact format

- well defined geometry definition for reconstruction that
  - is flexible w.r.t different detector concepts
  - has high level information needed for reconstruction
  - provides access to material properties
- abstract interface (a la LCIO)
  - concrete implementation based on XML files
  - and Mokka-CGA

Gear status

- version v00-03
  - main detectors: TPC, Hcal, Ecal and VXT (new) interfaces defined and implemented
  - + free form user parameters for other detectors
  - description of TPC prototypes (rectangular pad plane)
  - description of calo prototype
- GearCGA (Mokka/geant4) - material properties
  - detailed material properties for every point (and distance)
- related work: MokkaGear
  - extract geometry information in Mokka drivers when detector is built in memory for simulation
  - use Gear to create XML files for reconstruction
  - -> have only one source of geometry information

JRA2 (TPC) software

JRA2 Software Status

Peter Wienemann
U Bonn / U Freiburg

EUDET Annual Meeting
October 18-20, 2006
Munich, Germany

MarlinTPC

- Project started to establish common TPC software based on LCIO data model and the Marlin analysis and reconstruction framework (see `http://ilcsoft.desy.de`)
- C++ programming language
- Modular design with well defined interfaces between modules (beyond what is already fixed by LCIO)
- Standards agreed upon by 6 TPC groups in a TPC software meeting at DESY in June 2006

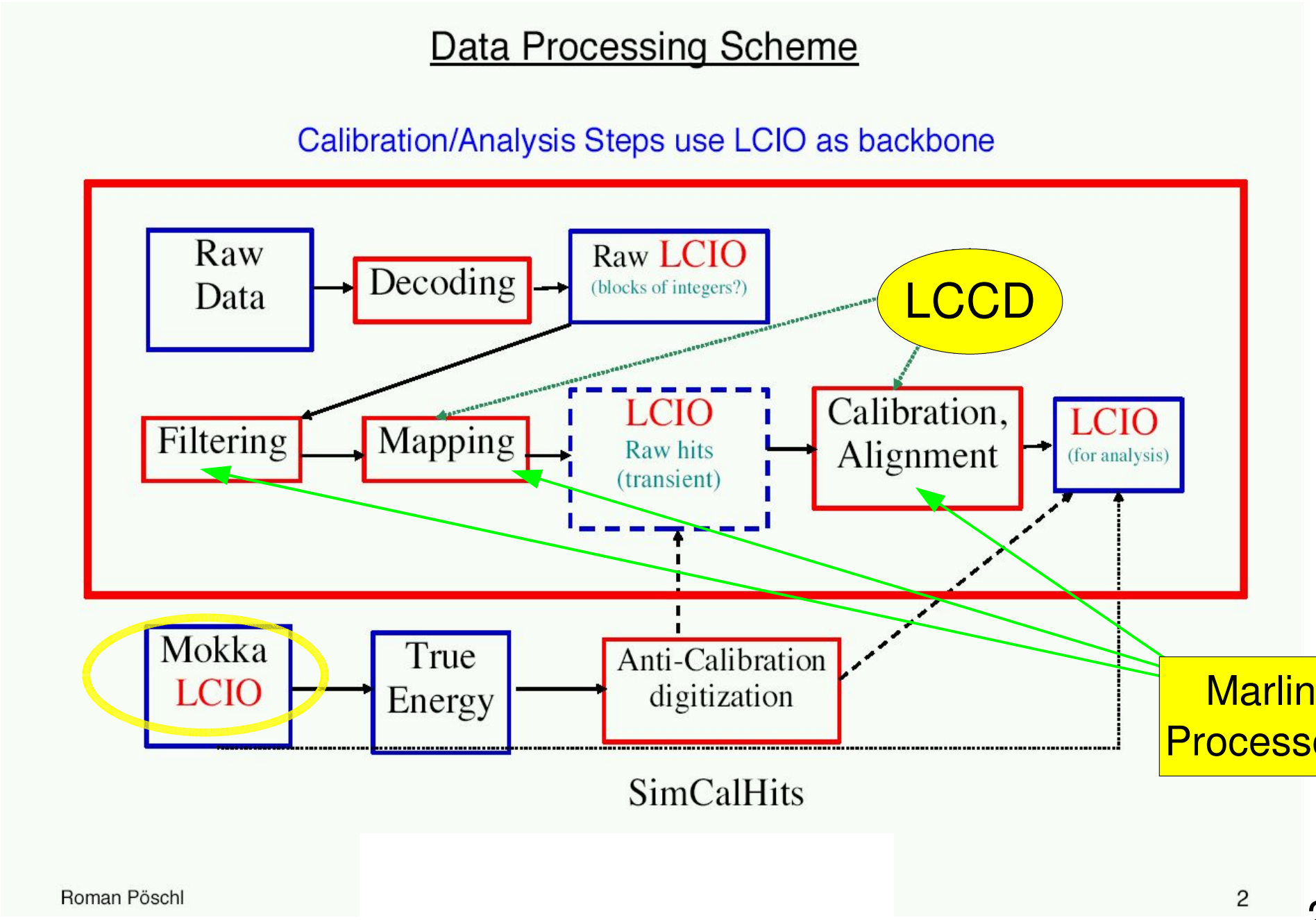
GEAR and LCCD

- GEAR: geometry API
- LCCD: conditions data toolkit
- "Static" information (pad geometry, readout frequency, etc.) stored in GEAR files (XML)
- Data subject to changes during data taking (drift velocity, voltages, B field, calibration data) stored using LCCD

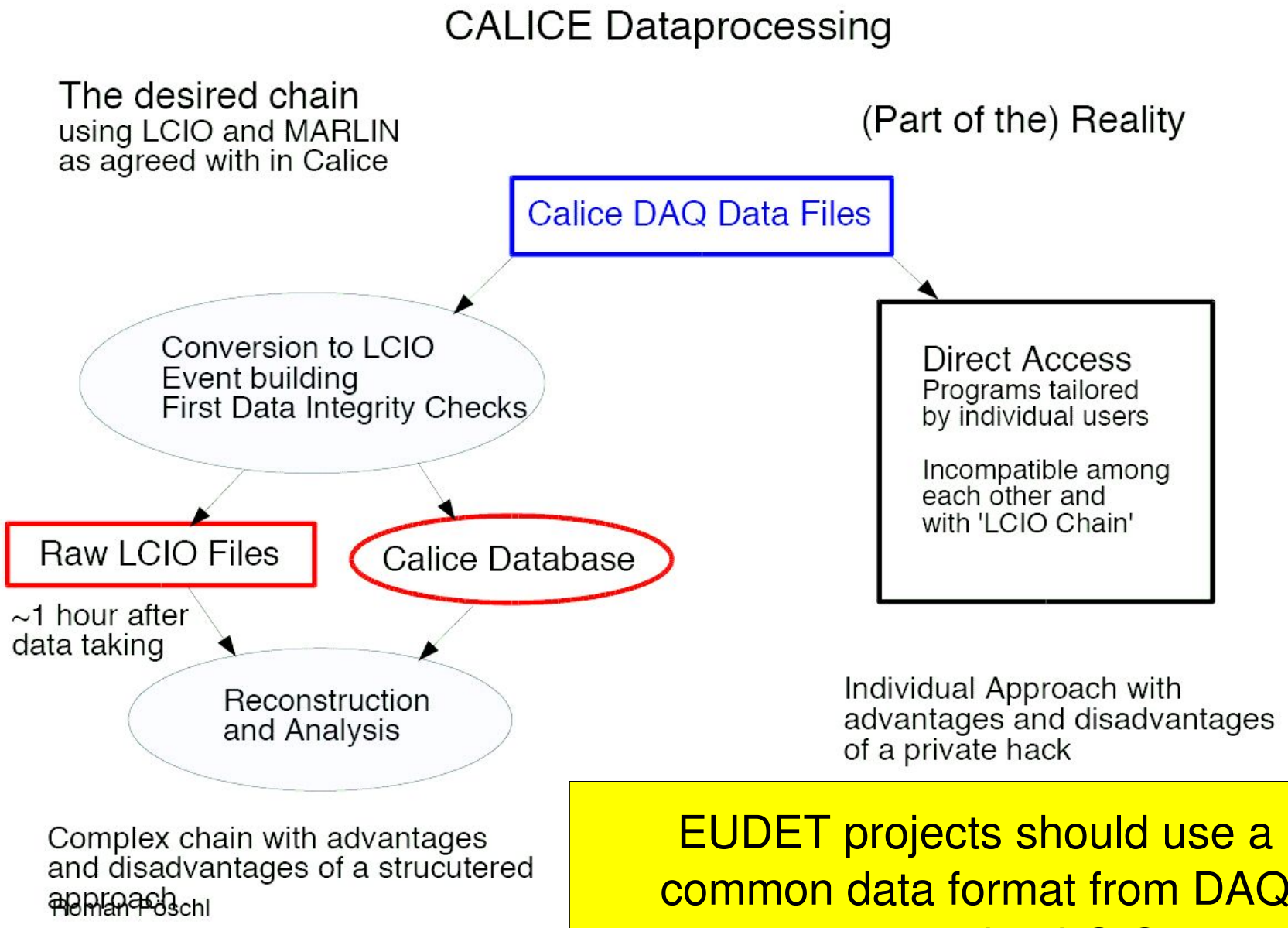
Processor structure

Data structure	Processor name	input/output collection name
`TrackerRawData`		TPCRawData
	TrackerRawData2DataConverter	
`TrackerData`		TPCConvertedRawData
	PedestalSubtractor	
	ChannelByChannelCorrector	
	LinearityCorrector	
	TimeShiftCorrector	
`TrackerData`		TPCData
	PulseFinder	
	ChannelMapper	
	GainCorrector	
`TrackerPulse`		TPCPulses
	HitFinder	
	HitPRFCorrector	
`TrackerHit`		TPCHits
	TrackFinder[Method]	
`Track`		TPCSeedTracks
	TrackFitter[Method]	
`Track`		TPCTracks

Calice testbeam software



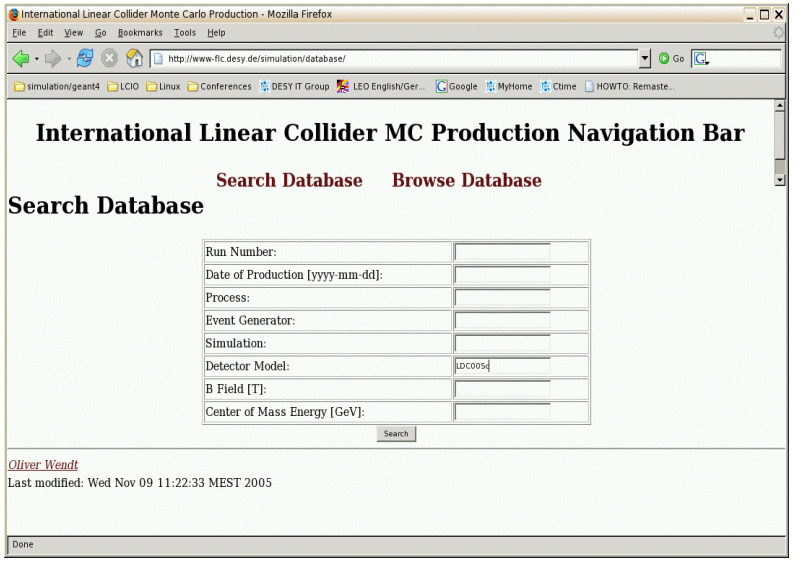
Problems with having two data formats

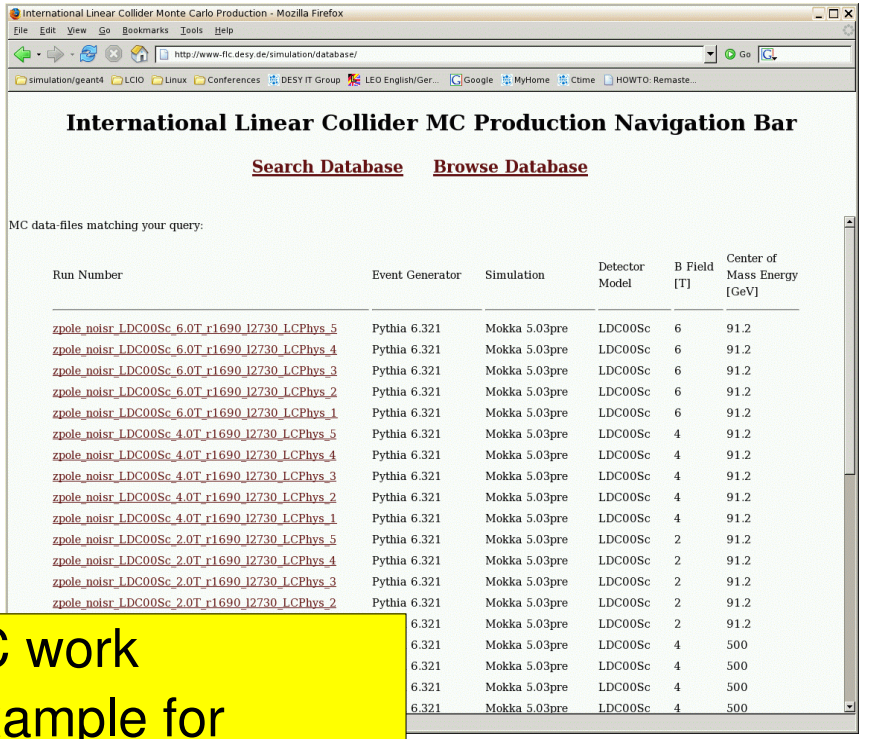


EUDET projects should use a common data format from DAQ to analysis - LCIO

IIC Grid activities

- two VOs: ilc + calice are available for ILC computing
- ilc used the grid for MC production for detector optimization – variyng
  - B,R_TPC,L_TPC,...
- produced these files on the **grid** for VO ilcs – 450 kevts:
  - Z0 and uds, ccbb , ttbar, WW, ZH @ 500 GeV
  - 4 detector variants, 3 T and 4 T field
- database with available data files
- use **grid tools** to distribute/download the data !





- the grid can be and is used for ILC work
- existing scripts can serve as an example for your computing projects

grid from an ILC user's perspective

Grid Computing
From a User's Point of View

A Thousand CPUs at Your Command

Adrian Vogel
DESY FLC

Certificates and VOs

Grid certificates

- authentication (proves who you are)
- one certificate per human being
- X.509-style with public and private key (RSA)
- signed by your regional Grid Certification Authority
- used to create "proxy certificates" as necessary (unprotected working copies with limited lifetime)

Virtual Organisations (VOs)

- authorisation (grants access to resources)
- world-wide "user groups" in the Grid
- everybody should be member of (at least) one VO

Getting Started (e.g. at DESY)

Log in to an SL3 machine, get a UI

- `ssh slref.desy.de` (for example)
- `source /afs/desy.de/group/it/grid/\ UI/GLITE/etc/profile.d/grid_env.sh`

Create a proxy (default lifetime 12 hours)

- `glite-voms-proxy-init --voms ilc`
- `glite-voms-proxy-info --all`
- `glite-voms-proxy-destroy` (after you're finishe

You should now have access to all Grid resources which are available to the VO `ilc`

Job Submission

Check the JDL file (and list suitable CEs)

- `glite-job-list-match test.jdl`

Submit the job, store the job ID

- `glite-job-submit -o test.jid test.jdl`

Check the job status (e.g. "scheduled", "running", "done")

- `glite-job-status -i test.jid`

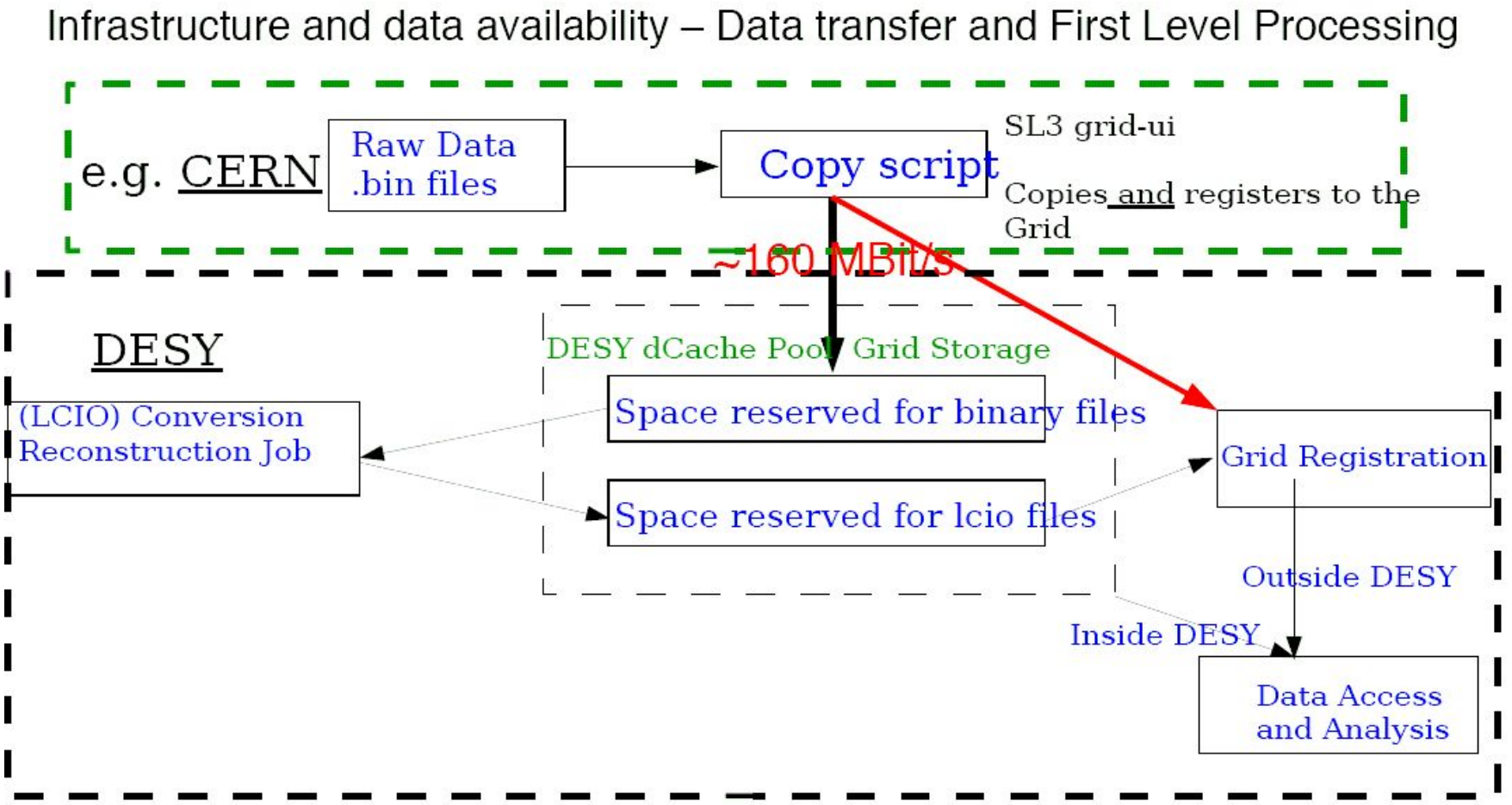
Retrieve the job output (→ status "cleared") – only once!

- `glite-job-output -i test.jid --dir .`

Cancel a job (→ status "cancelled")

- `glite-job-cancel -i test.jid`

CALICE – grid data storage



- Raw Data ~20 Min. after data taking available to whole collaboration
- Data taken at CERN and stored and (firstly) processed at DESY
- Datatransfer is very stable
- Model can be transferred to data taking at Fermilab

Grid tools well suited for decentralised collaboration like Calice

>7 TByte available in grid catalogue

Summary

- the EUDET tasks NA2 COMP+ANALYS
  - set up grid computing clusters
  - software framework for simulation and analysis of testbeam data
- are in a good shape and ready to provide the required deliverables on time
- both grid and software tools already actively used by testbeams (TPC/CALICE)
- spending profile somewhat delayed ...

All EUDET software activities
should be carried out in the context of
the existing software framework/ grid installations
e.g. DAQ software should use LCIO

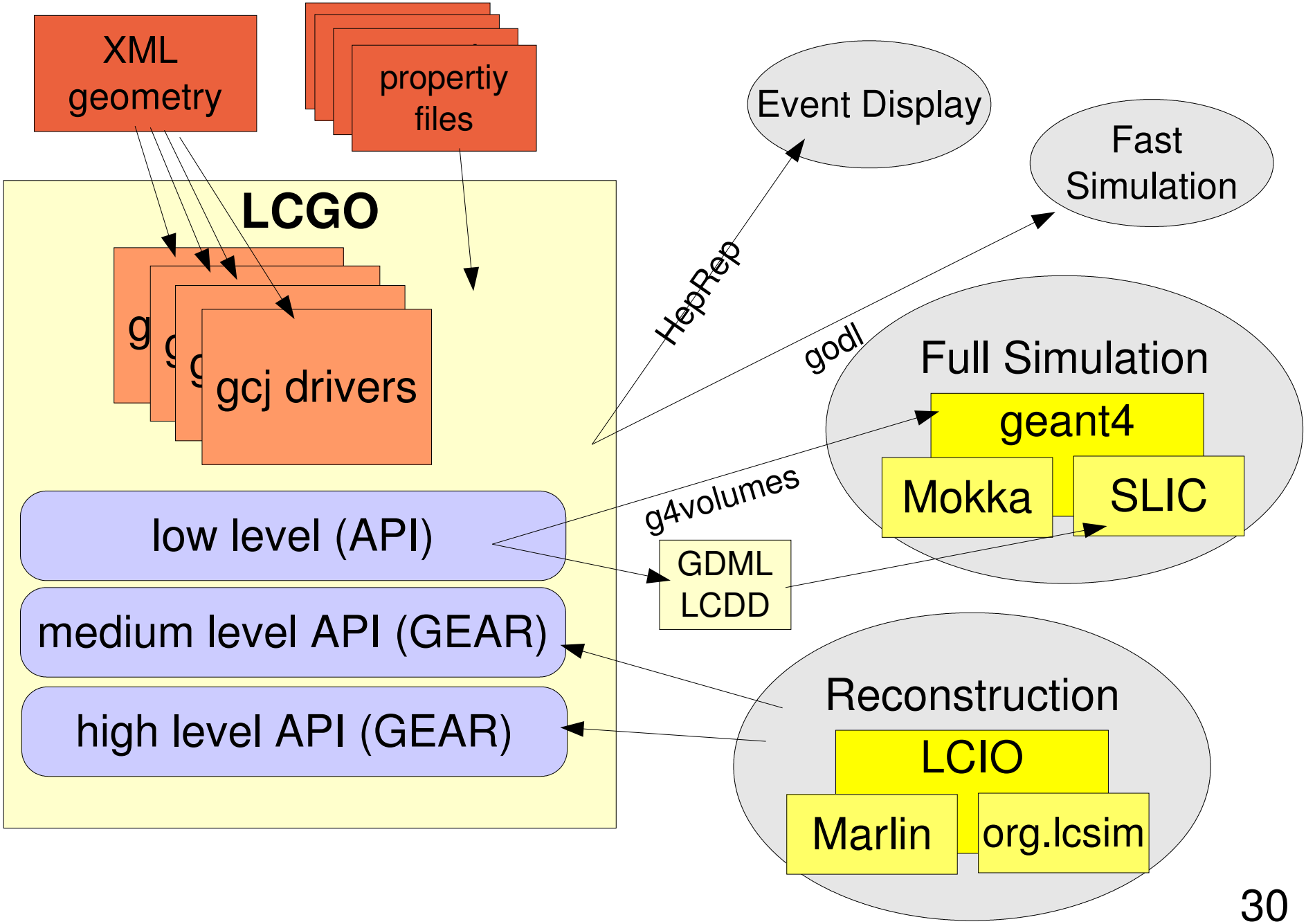
your input is needed to improve the software !

Backup Slides

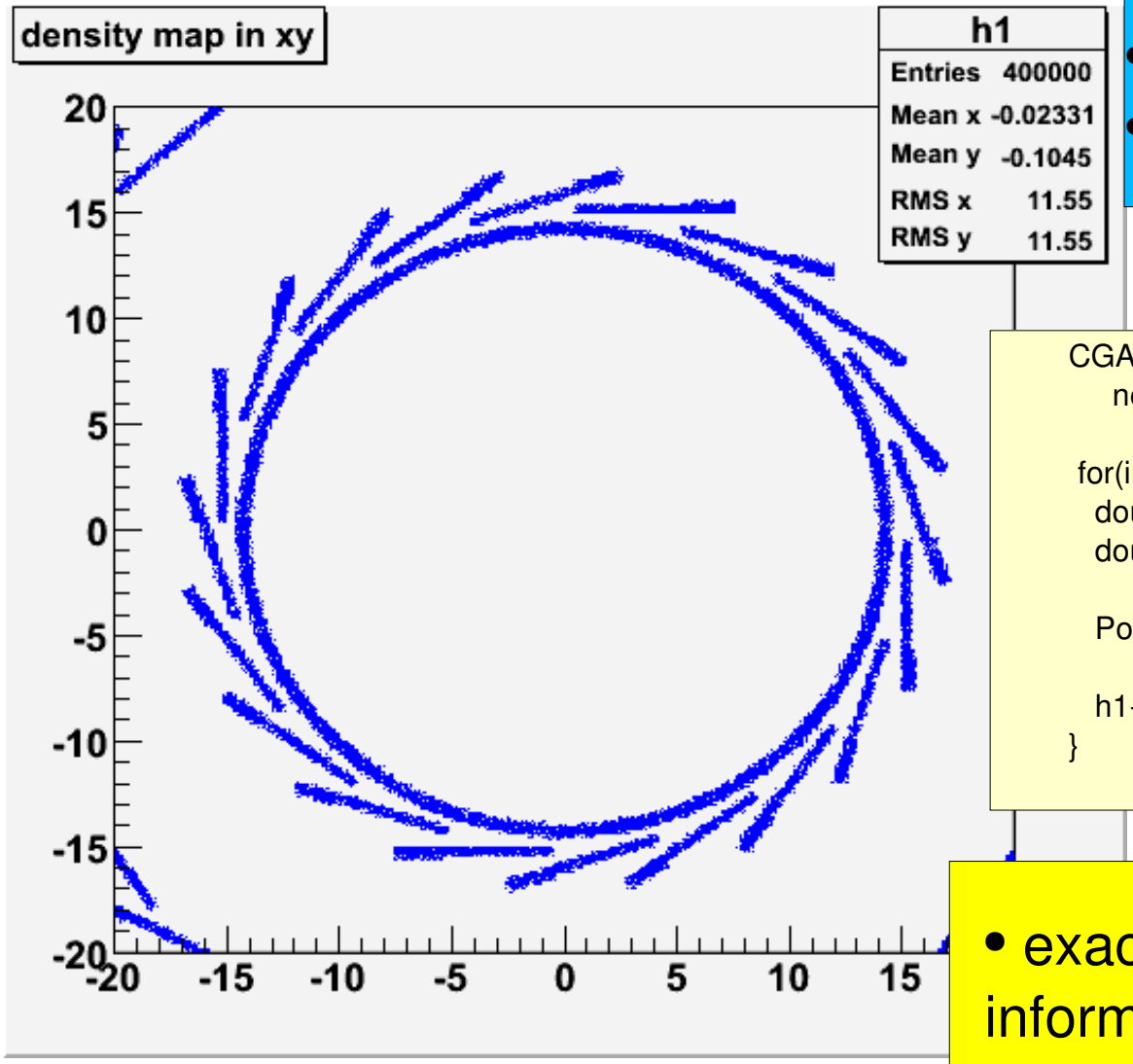
A Common Geometry Toolkit

- **LCGO**: A common geometry toolkit to be used in all (?) ILC frameworks
  - SLAC-DESY project - initially
  - -> of course open for all collaborators, e.g. FNAL
  - work just started – aiming for spring/summer 2007
- requirements/goals for LCGO:
  - be at least as functional as existing systems (org.lcsim, GEAR, Mokka, SLIC,...)
  - enable smooth transition path from existing systems
  - encourage/increase interoperability between systems
  - have no known principle short comings: "everything should be possible"

LCGO implementation prelim.



CGAGear

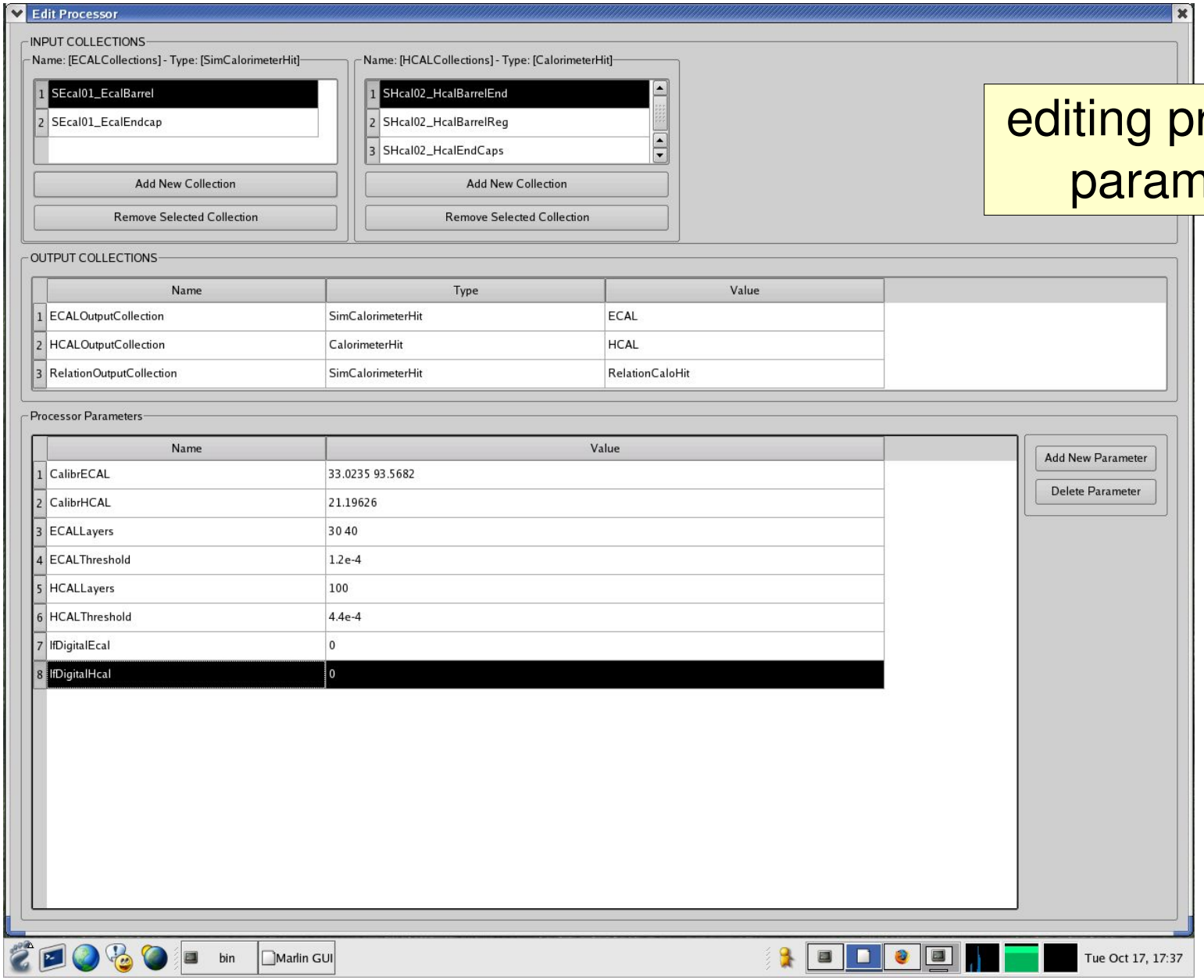


- implemented by G.Musat, LLR
- to be released soon

```
CGAGearPointProperties * pointProp =
    new CGAGearPointProperties(steer.str(),...);

 for(int i=0 ; i<nPoint ; ++i){
   double xr =  xmin +  ( xmax - xmin)  * random() ;
   double yr =  ymin +  ( ymax - ymin)  * random() ;

   Point3D p( xr, yr, z0 ) ;

   h1->fill(  xr, yr, pointProp->getDensity( p )  ) ;
}
```

- exact geant4 material & field information at runtime !
- performance ?
- practical issues (linking g4) ?

example: MarlinGUI II



editing processor parameters

Deliverables and Requirements

- **requirements**:
  - documentation and its regular update are of utmost importance
    - “spread the information”
  - other EUDET participants should contribute by:
    - properly defining the *requirements* of the framework
    - *providing* and interfacing *simulation and reconstruction* software for the various detector technologies
    - testing the framework.
  - **deliverables**:
    - we expect to have a **first version** of the common data analysis and simulation framework ready **after 18 month**
    - **-> should be a reasonable goal**
    - development however must continue throughout the whole duration of the project to cope with

(from annex1)