Plans for EUDET NA2

Task ANALYS

Common Analyis and Simulation Software

Frank Gaede
DESY

EUDET, NA2 Meeting, CERN
December 14, 2005

Objectives for task ANALYS

- **development of a common data analysis and simulation infrastructure**
  - 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 experiments needed for the interpretation of the measurements
  - creation of a repository for experimental and simulation data
  - embedding into existing GRID infrastructure to allow easy exchange of data and transparent exploitation of other available computing resources.

(from annex1)

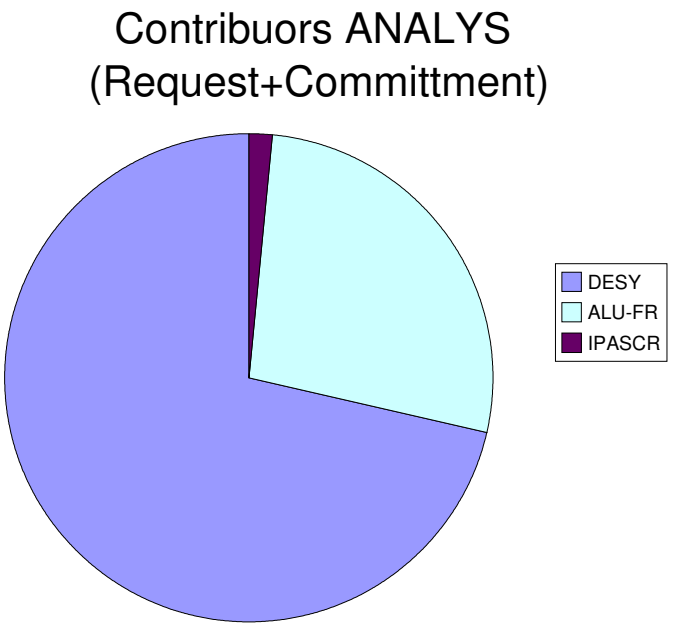
Deliverables and Requirements

- documentation and its regular update are of utmost importance
  - “spread the information”
- the participants in this Networking Activity (only?) 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**
  - development however must continue throughout the whole duration of the project to cope with
    - the increasing demands caused by the accumulation of data
    - the expected increasing complexity of the experiments.

(from annex1)

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**



Personnel planning ANALYS

- DESY
  - commitment 12ppm: F.Gaede 25% for full project length
  - request:12ppm (scientist) plan to hire programmer for ~18 month
    - start early 2006
    - possibly extend position with other funding sources
- ALU-FR (K.Desch)
  - request: 8ppm (scientist): plan to combine with funds (8ppm) from COMP to hire a postdoc that works part-time on COMP and ANALYS
    - Apr/May 2006
- IPASCR (J.Cvach)
  - commitment: 3ppm: PhD student that works part time on calorimeter simulation with geant4
    - ongoing

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
  - insufficient resource assigned via EUDET
- the testbeam software effort will be part of 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

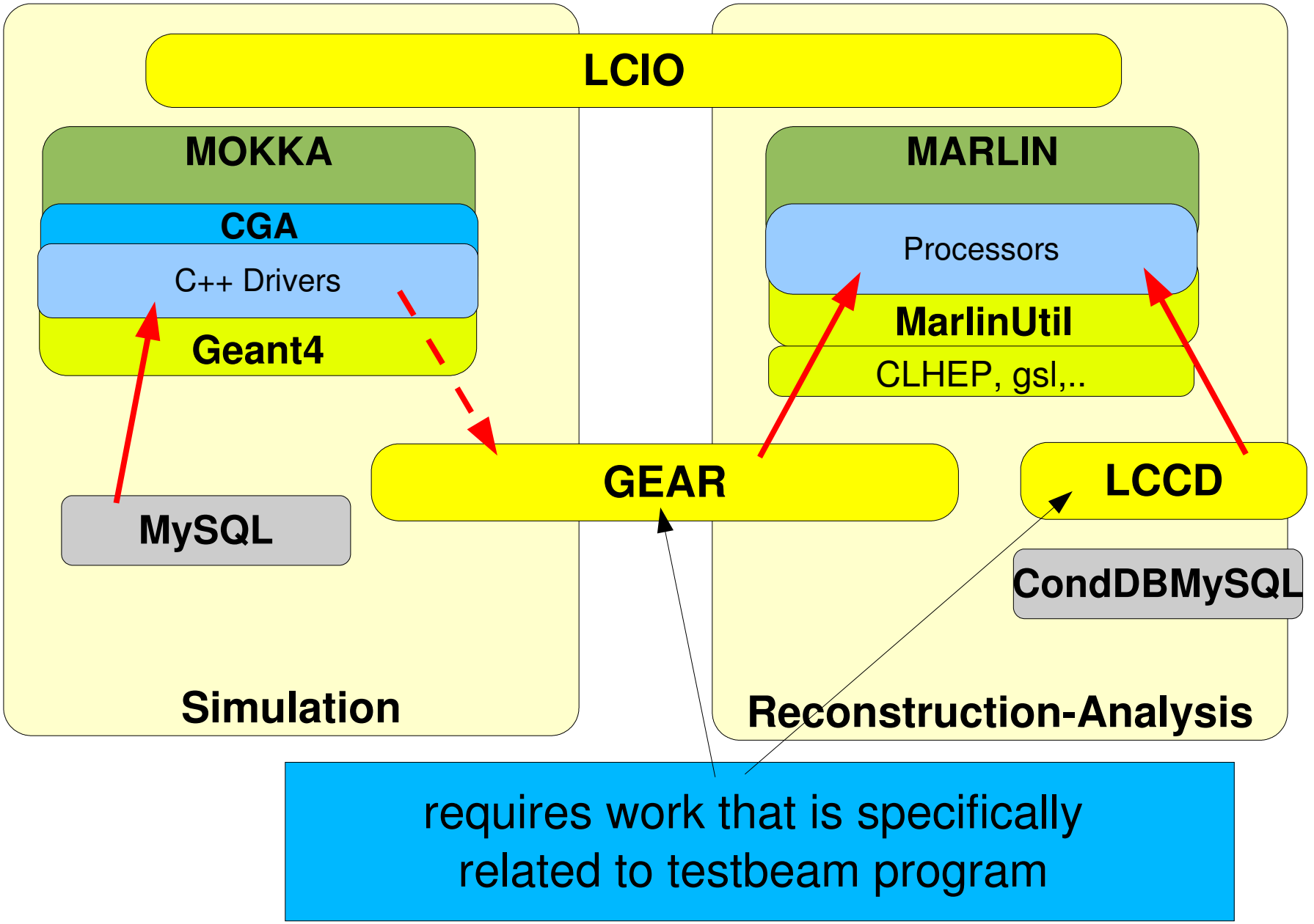
Software packages for ILC framework

- LCIO
  - data model & persistency
- Mokka
  - geant4 full simulation
- Marlin
  - C++ application framework
  - MarlinReco
    - Marlin based reconstruction
- LCCD
  - conditions data toolkit
- GEAR
  - geometry description

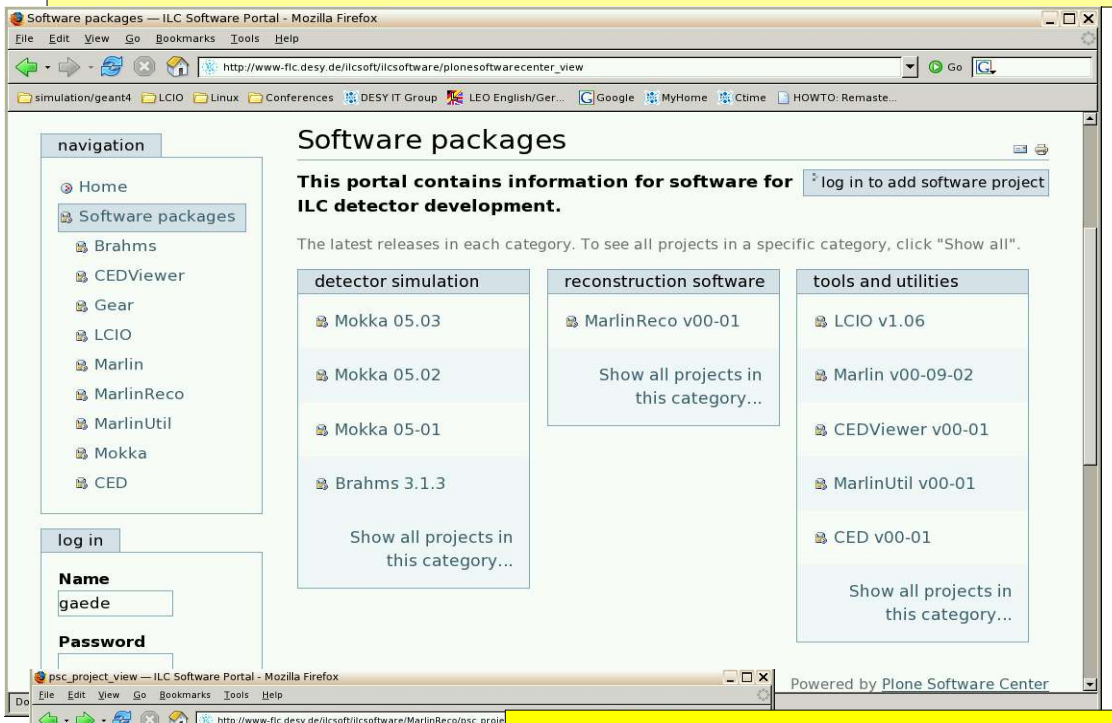
all packages developed at or with contributions from DESY

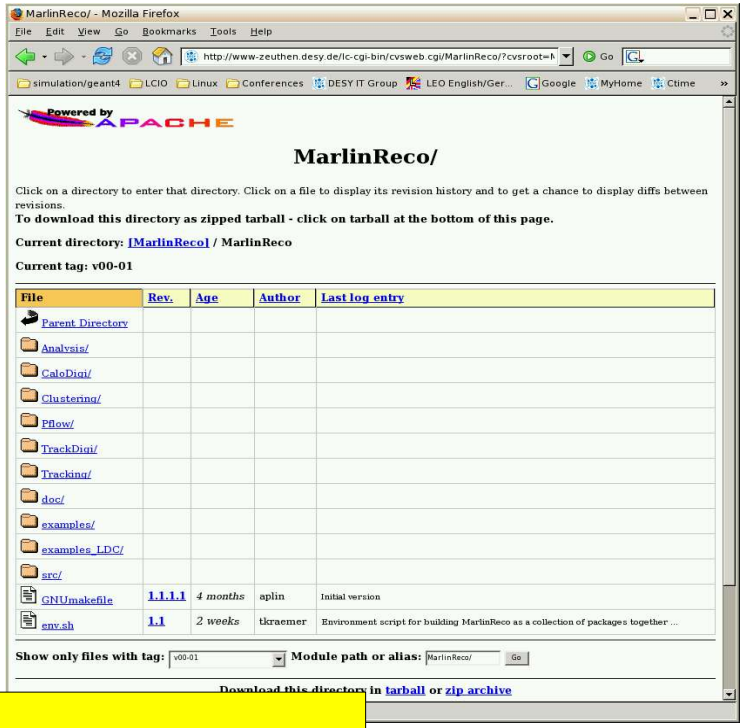
-> DESY will naturally continue to develop and improve these tools in the context of EUDET

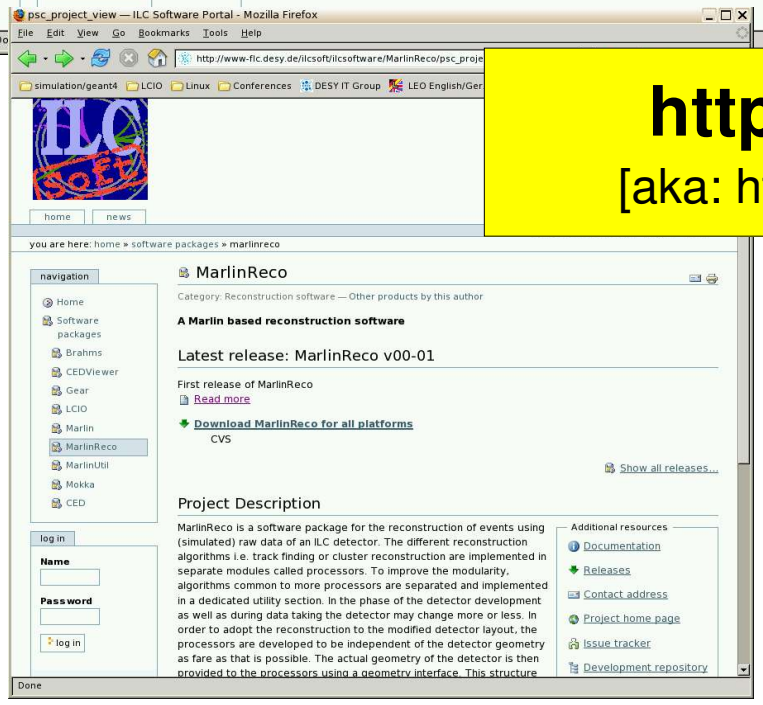
LDC software framework



ILC software portal

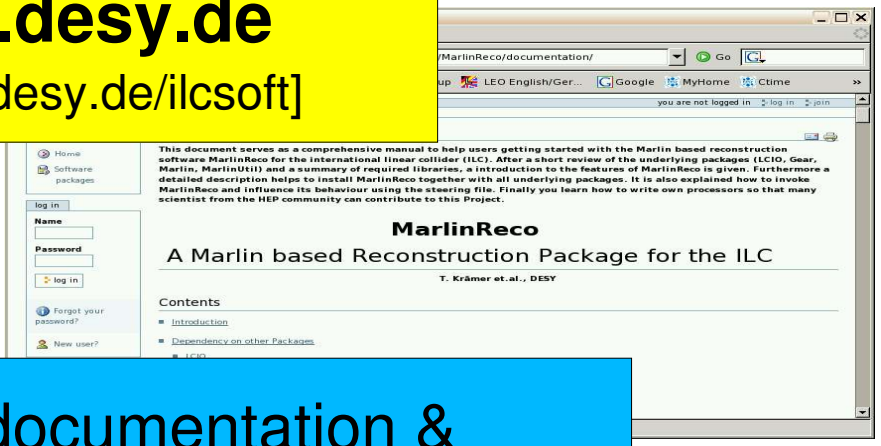






http://ilcsoft.desy.de

[aka: http://www-flc.desy.de/ilcsoft]



documentation &
spreading the information

Grid activities

- ongoing grid activities (not related to EUDET):
  - DESY
    - H1and ZEUS Monte Carlo production
    - ILC Monte Carlo production started
    - Tier2 for Atlas and CMS currently set up
  - ALU-Fr
    - Tier2 for Atlas currently set up
- use existing experience to create grid infrastructure:
  - job submission scripts (computing grid)
  - data catalogue (data grid)

Plan: mainly ALU-Fr (postdoc) activity

in close collaboration with DESY groups (FLC,IT)

Summary

- NA2 task ANALYS: “Provide a software framework for simulation and analysis (of testbeam data)”
- 35 ppm:
  - DESY: 24 , ALU-FR: 8, IPASCR: 3
- main activities planned:
  - DESY: overall simulation and analysis framework
  - ALU-FR: gridification of software and data
  - IPASCR: simulation software (geant4)

All EUDET software activities should be carried out in close collaboration with other ILC groups and in the context of the existing software framework/ grid installations

Backup slides ...

LCIO overview

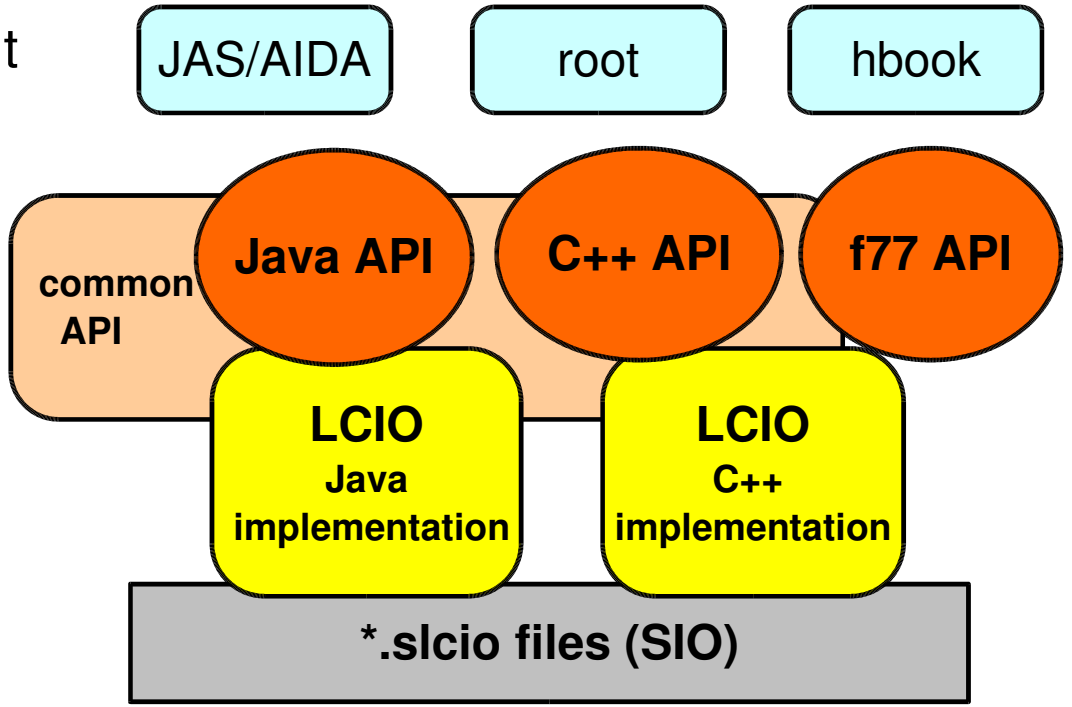
- DESY and SLAC joined project:
  - provide common basis for ILC software
- Features:
  - Java, C++ and f77 (!) API
  - extensible data model for current and future simulation and testbeam studies
  - user code separated from concrete data format
  - no dependency on other frameworks

simple & lightweight

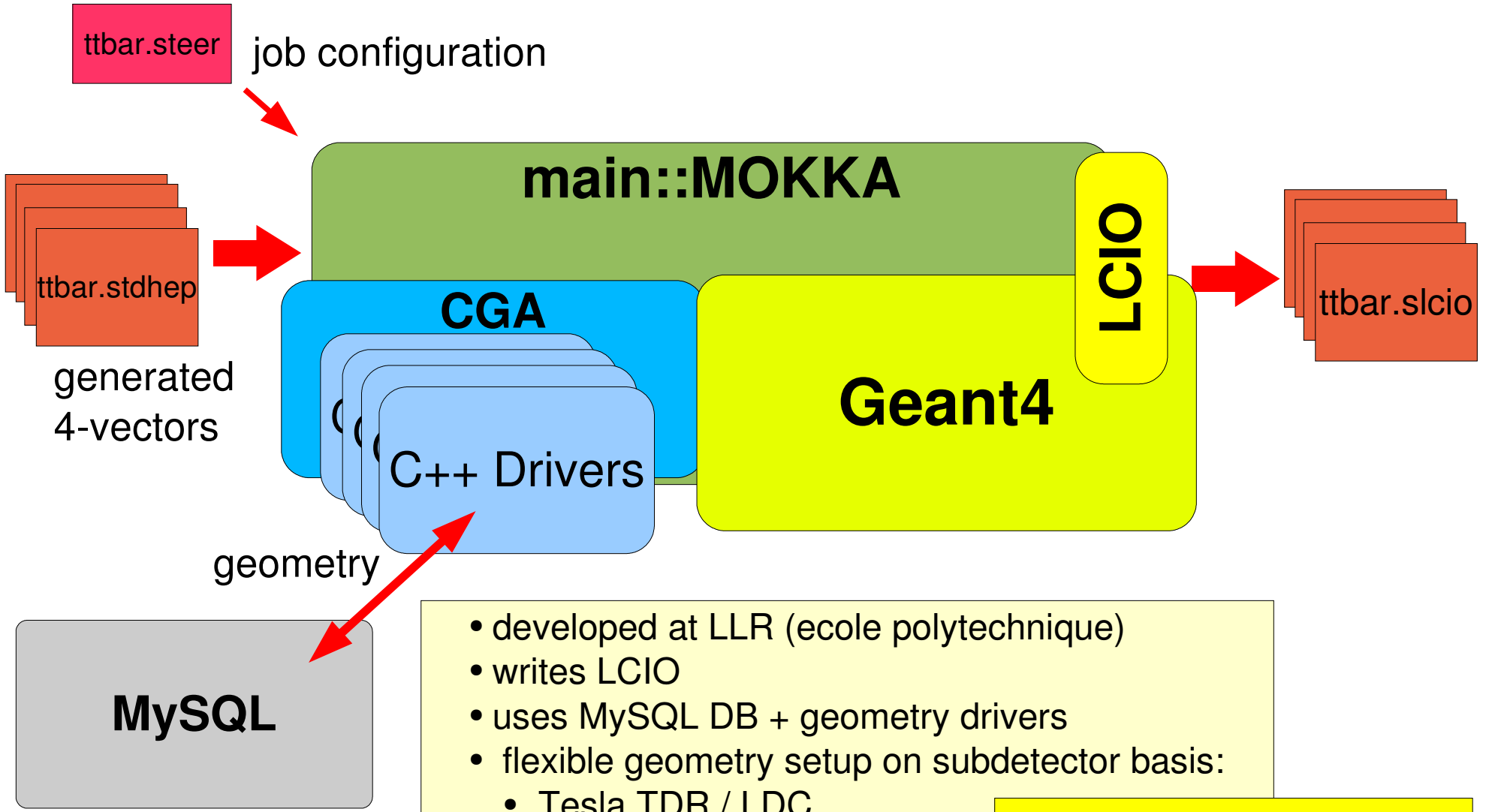
new release: **v01-06**

now de facto standard persistency & datamodel for ILC software

SW-Architecture



Mokka overview



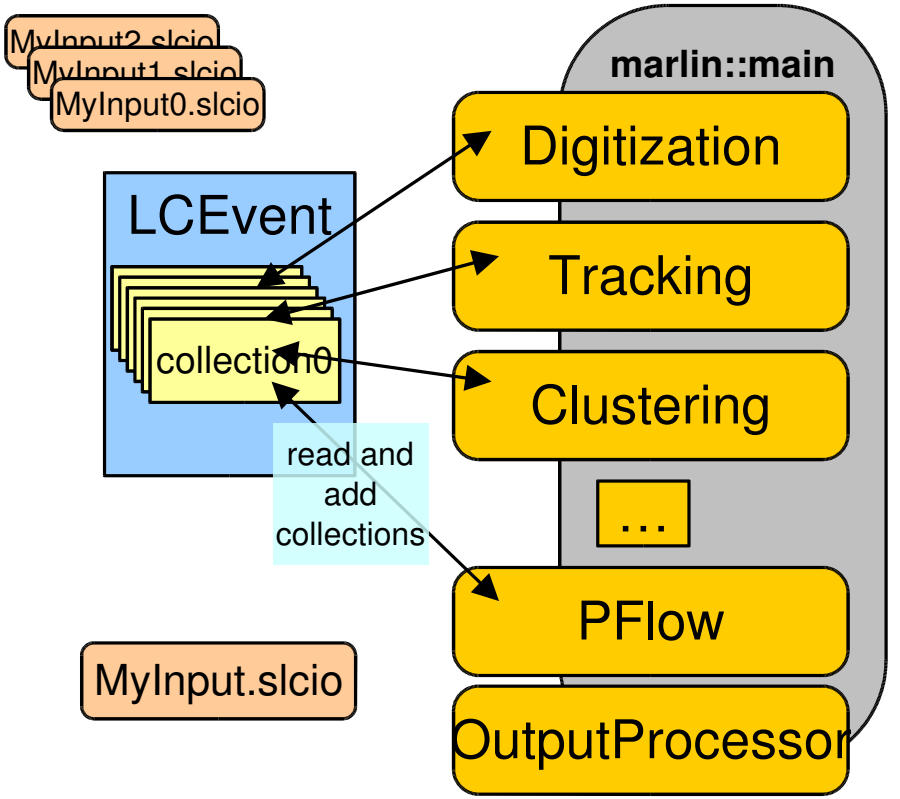
- developed at LLR (ecole polytechnique)
- writes LCIO
- uses MySQL DB + geometry drivers
- flexible geometry setup on subdetector basis:
  - Tesla TDR / LDC
  - SiD
  - Calice testbeam prototypes
- used in European/ LDC study

- new feature:
scalable detector models
-> detector optimization !!

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



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>
```

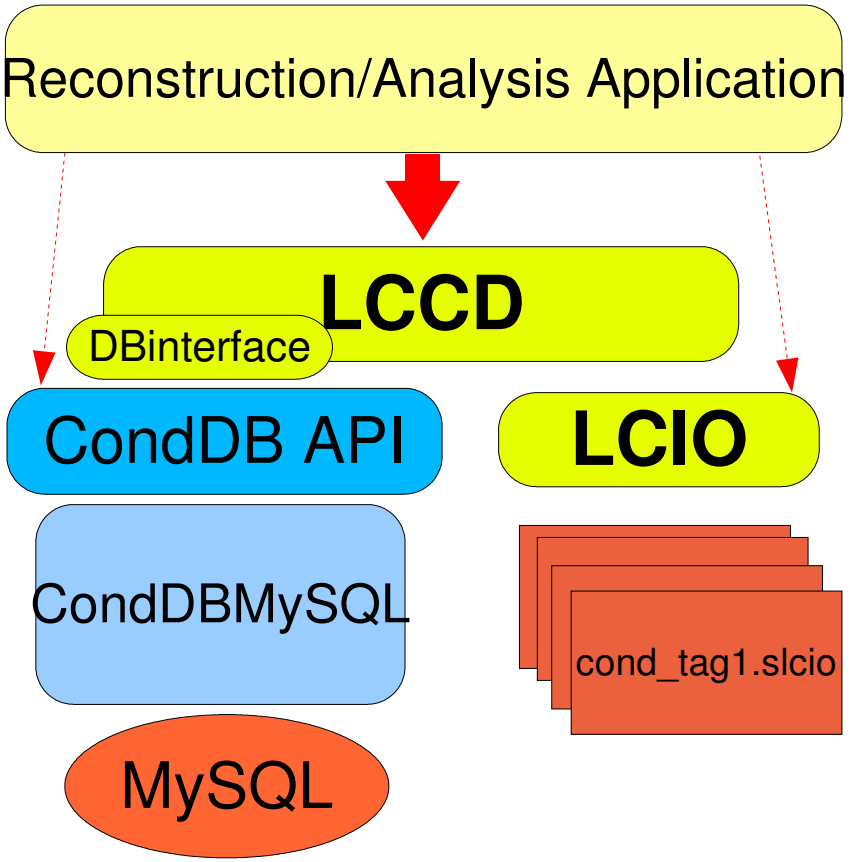
compatible with US – compact format

- 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 - planned
- abstract interface (a la LCIO)
- concrete implementation based on XML files
  - and Mokka-CGA – planned

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