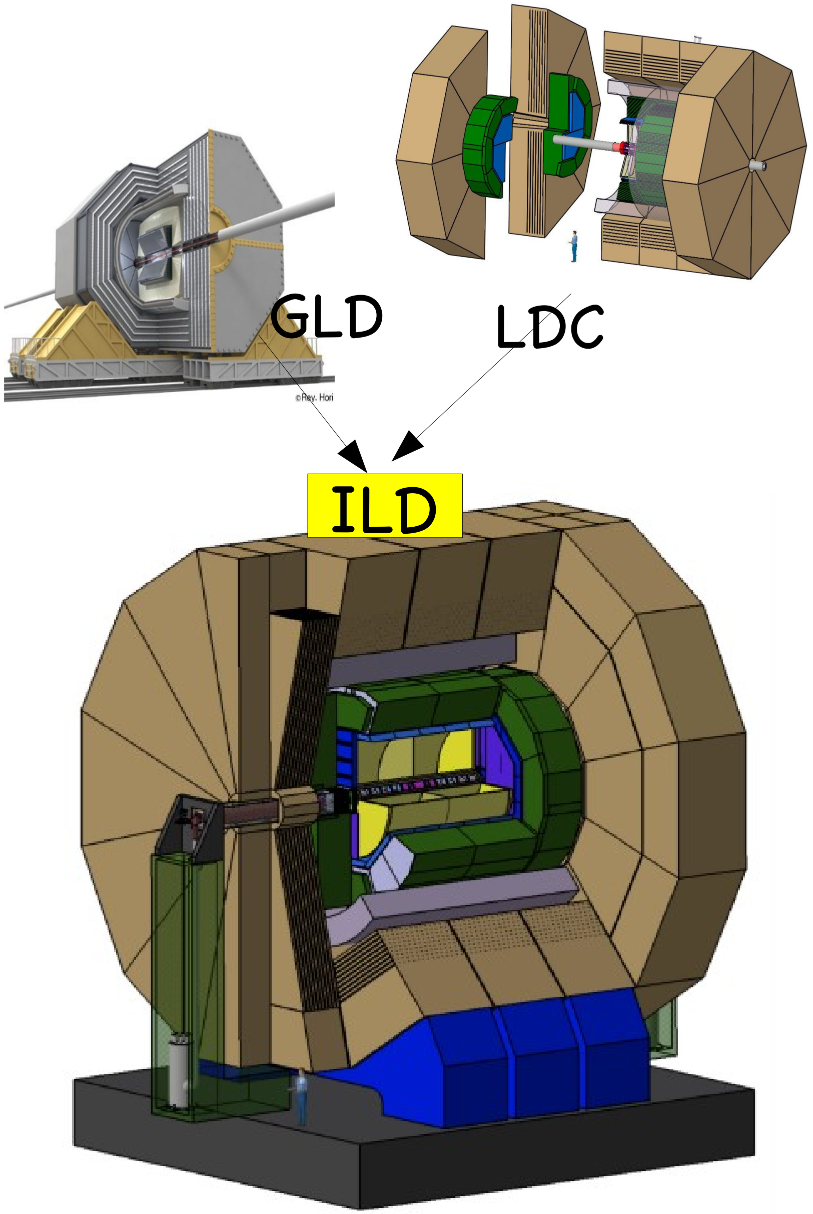
The ILD software framework – status and plans

Frank Gaede
DESY

TILC09, Tsukuba, 17-21 April2009

Outline

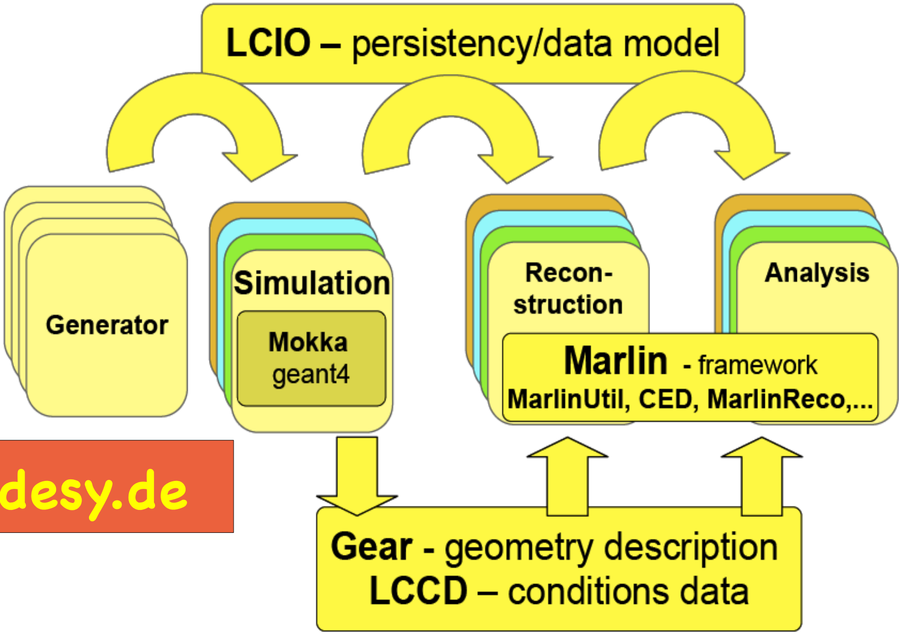
- Introduction
- The framework tools
- Simulation and Reconstruction
  - LOI Monte Carlo production
- Plans
- Summary

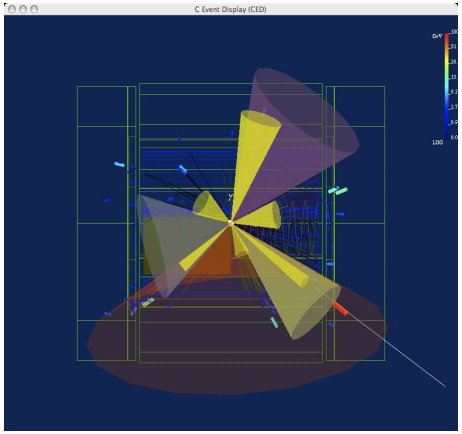


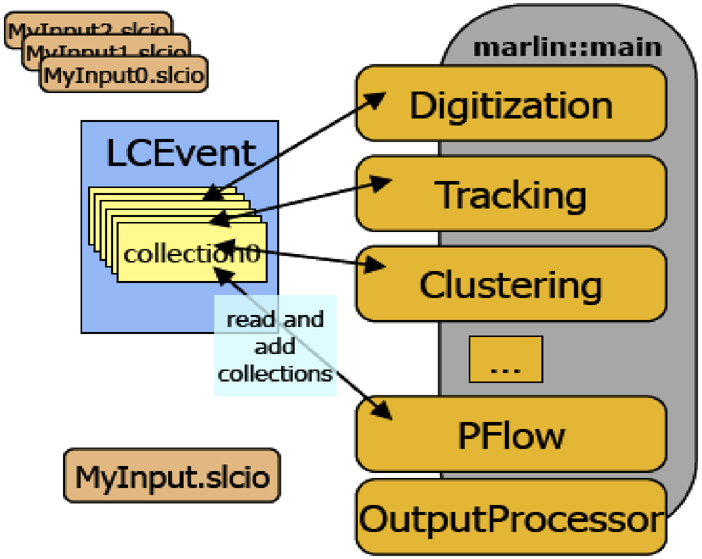
The ILD software framework - LDC flavor

- **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

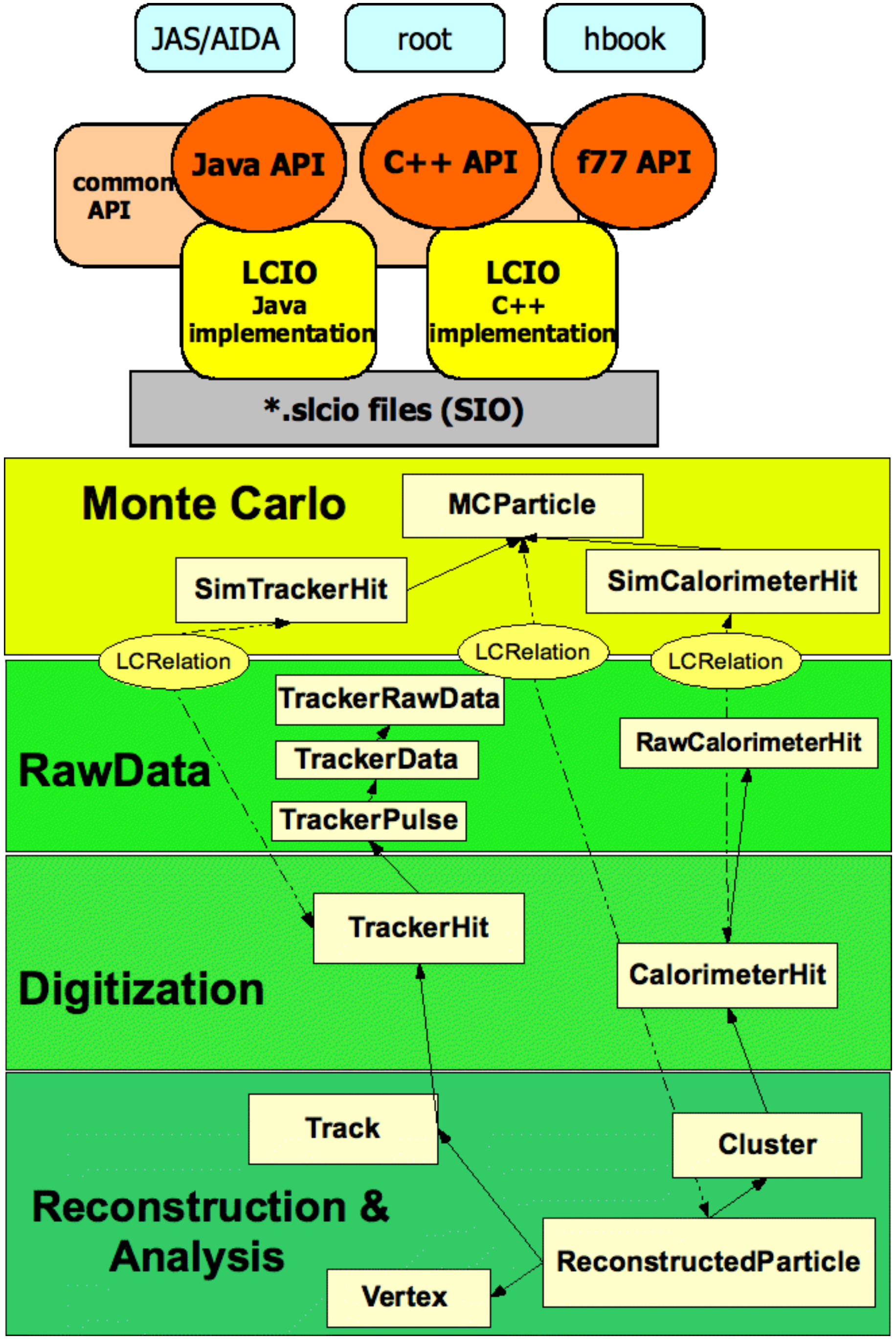
details at **http://ilcsoft.desy.de**







LCIO: persistency & event data model



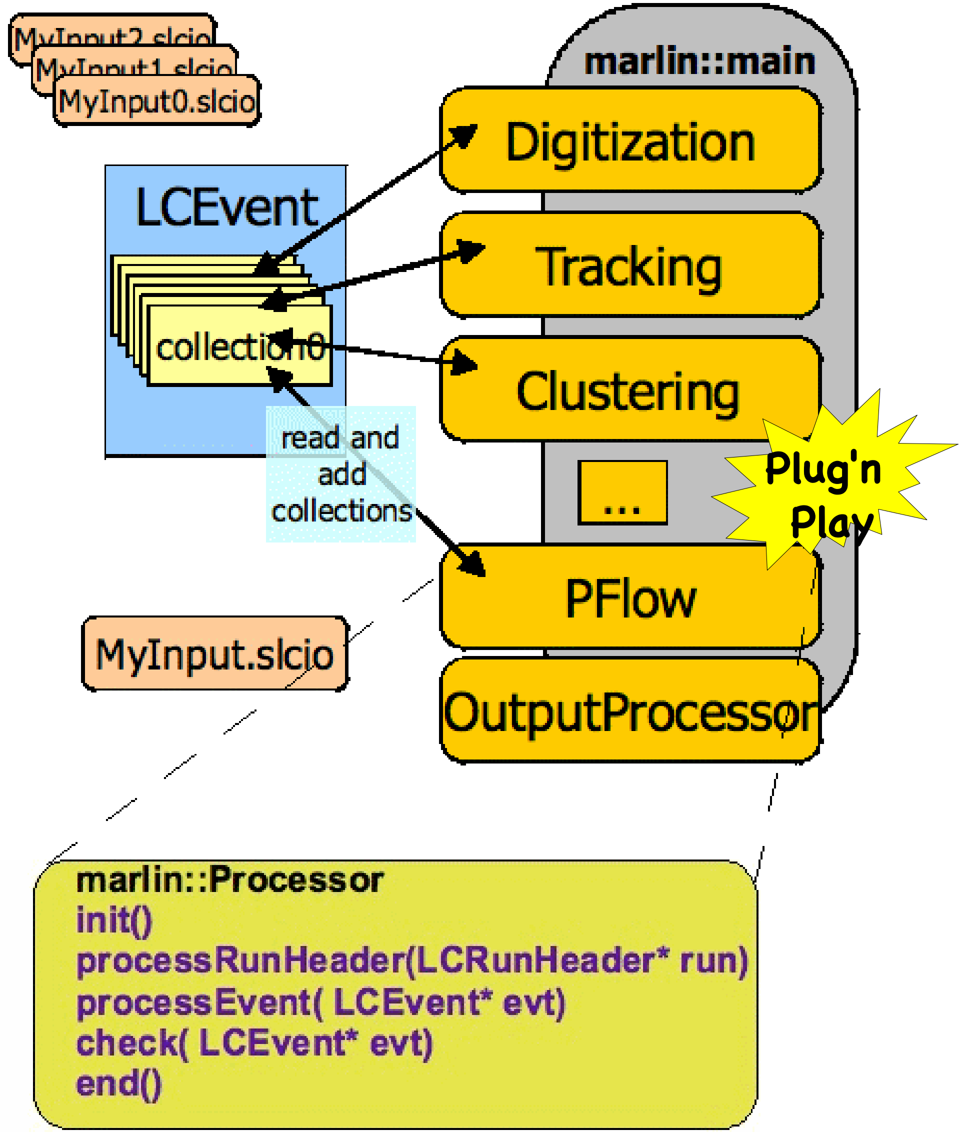
- joined DESY and SLAC project – first presented @ CHEP 2003
- provides persistency (I/O) and an event data model to ILC detector R&D community
- features:
  - Object I/O (w/ pointer chasing)
  - schema evolution
  - compressed records
  - hierarchical data model
  - decoupled from I/O by interfaces
    - C++, Java (and Fortran)
  - some generic user object I/O

LCIO is used by ILD, SID, Calice, EUPixelTelescope, LCTPC,...

MARLIN application framework

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

- modular C++ application framework for ILC detector R&D
- component based
- shared library **plugins**
- **LCIO** as transient data model
- xml steering files
  - configure application @ runtime
  - processor parameters
- self documenting
- consistency check of input/output collection types
- built in logging mechanism

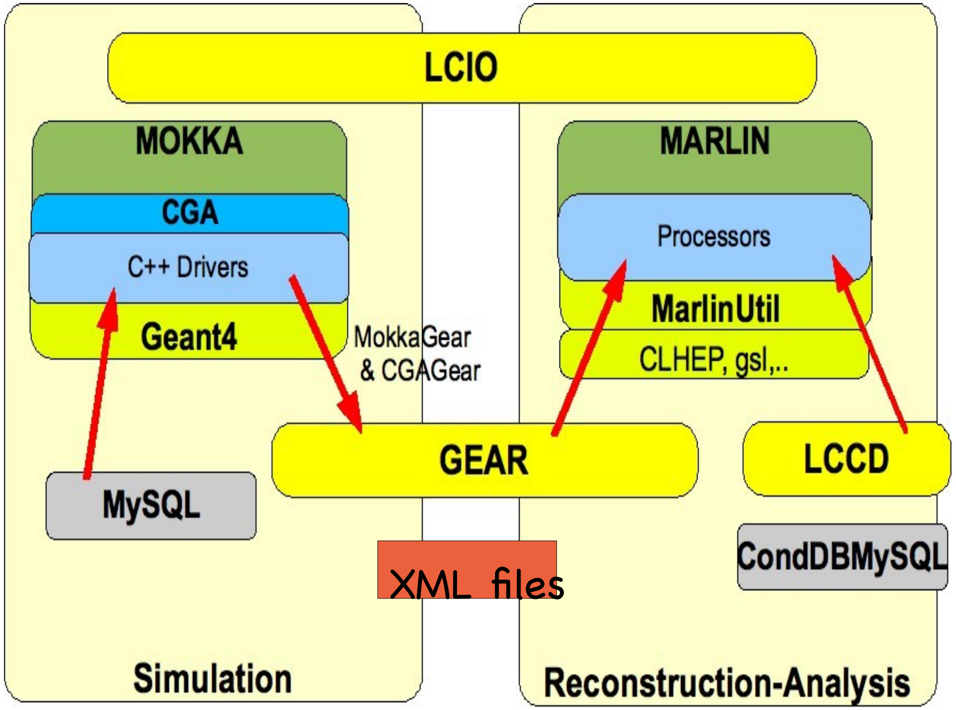


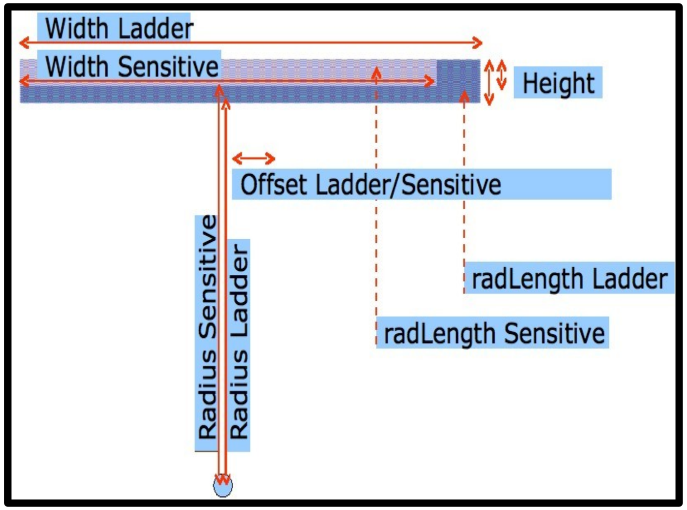
marlin::Processor
init()
processRunHeader(LCRunHeader* run)
processEvent(LCEvent* evt)
check(LCEvent* evt)
end()

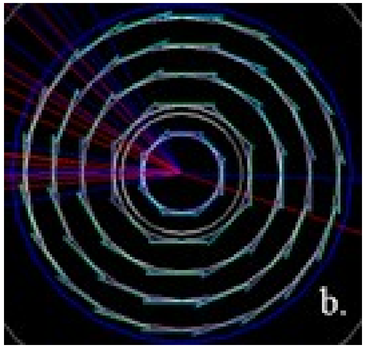
GEAR geometry description

- detailed geometry for simulation with Mokka/geant4:
  - MySQL data base with parameters
  - C++ drivers per subdetector
- at reconstruction:
  - high level abstract interface:
  - per subdetector type (Hcal,TPC,...) parameters/quantities for reconstruction
  - geometry + some navigation
  - implementation uses xml files
  - abstract interface for detailed geometry &materials:
    - point properties
    - path properties
    - implementation based on geant4

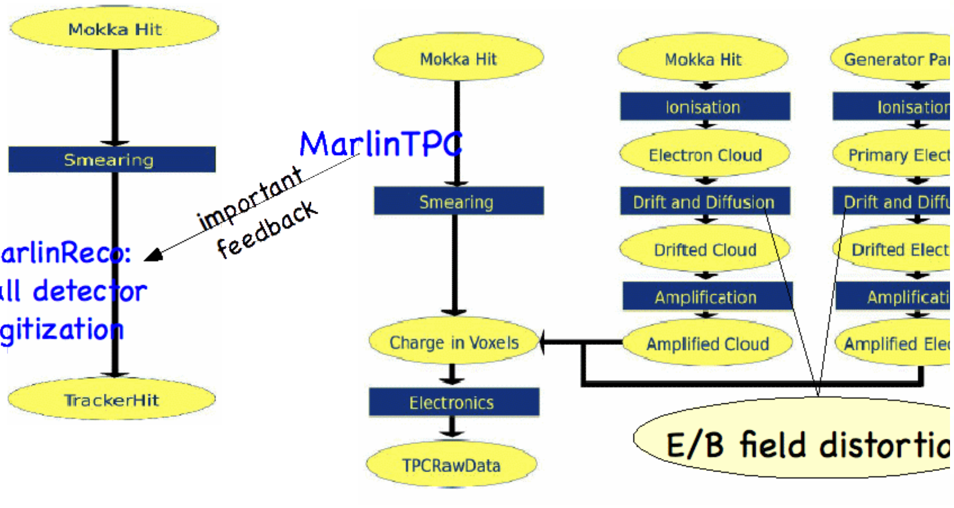
GEometry API for Reconstruction

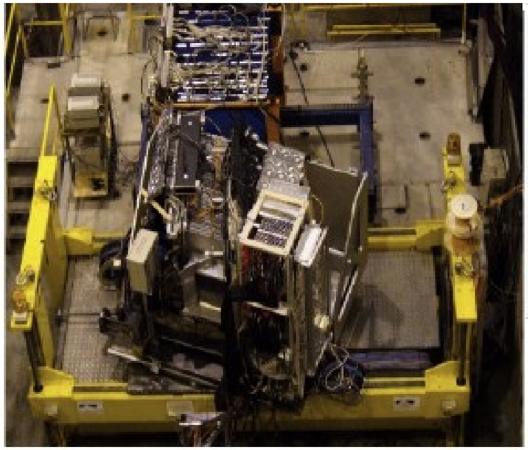


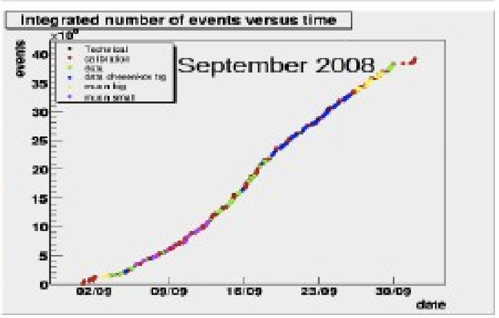




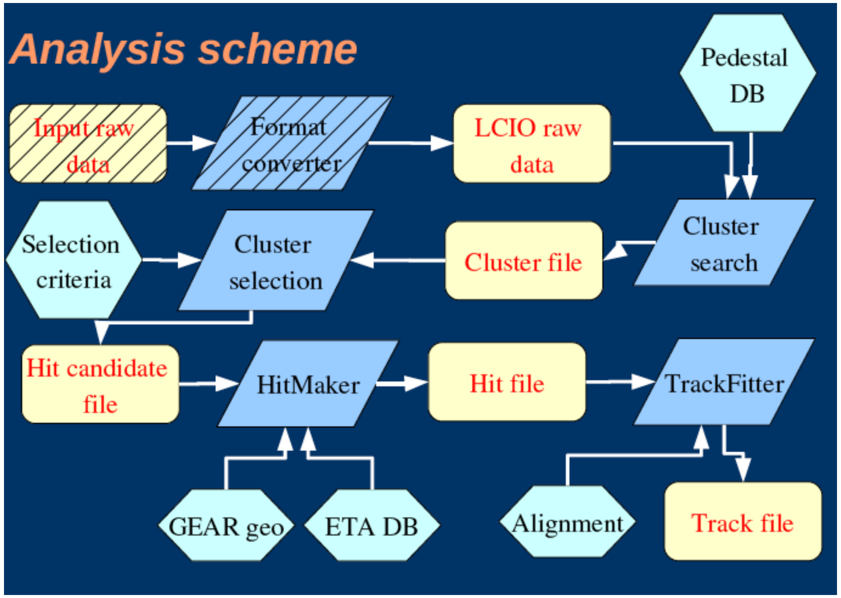
other framework users







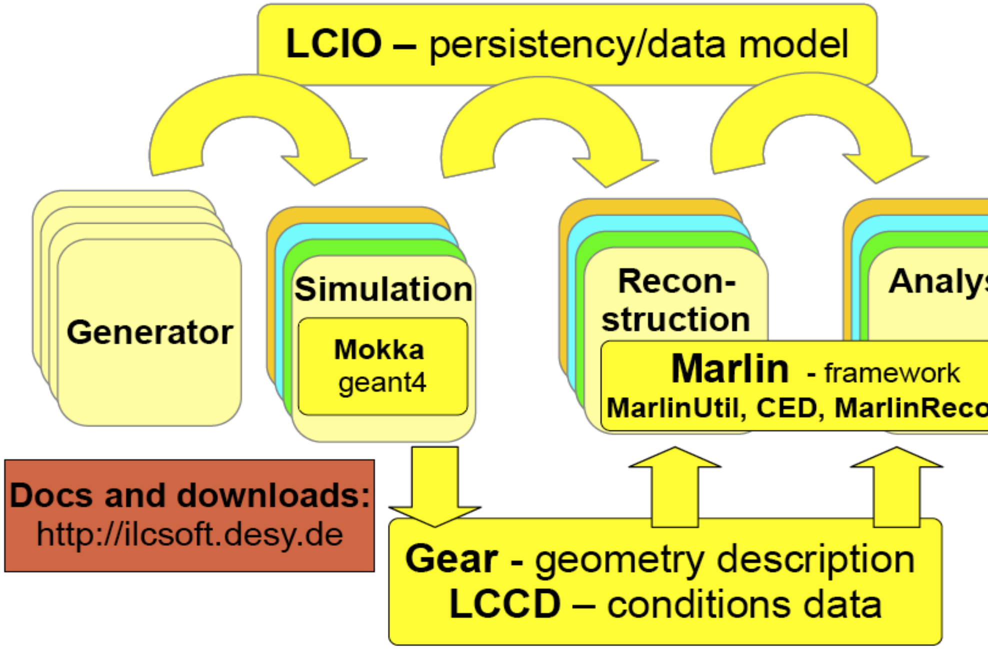
>300 Mio events
~40 TB (incl.MC/processed)



- framework not only used for ILD detector optimization - also for ILC testbeam experiments:
- **CALICE**
- **MarlinTPC**
- **EUTelescope**
- synergies from using the same framework for testbeam and large detector studies

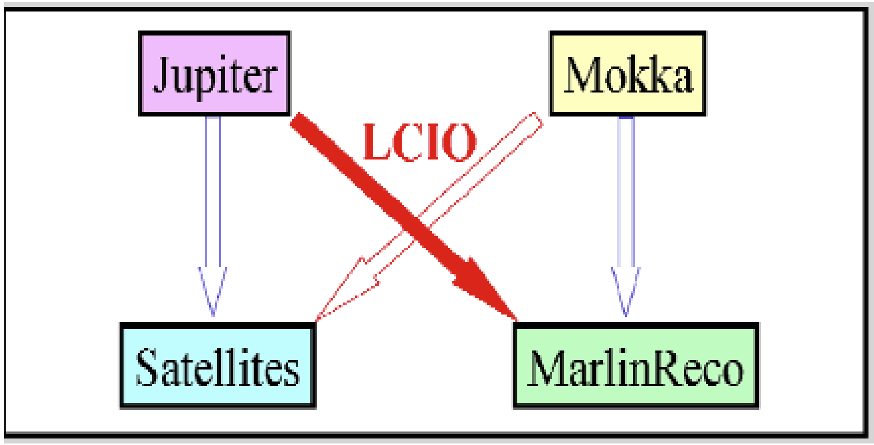
LDC & GLD sw frameworks

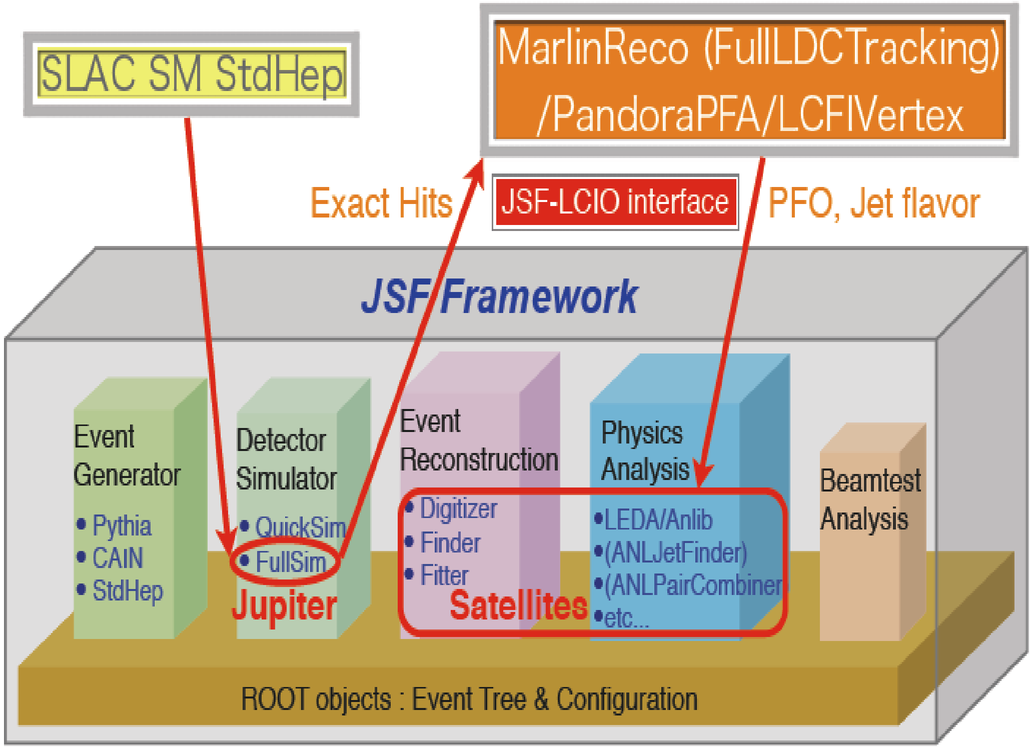
Frank Gaede, TILC09, Tsukuba, 16-21 April 2009



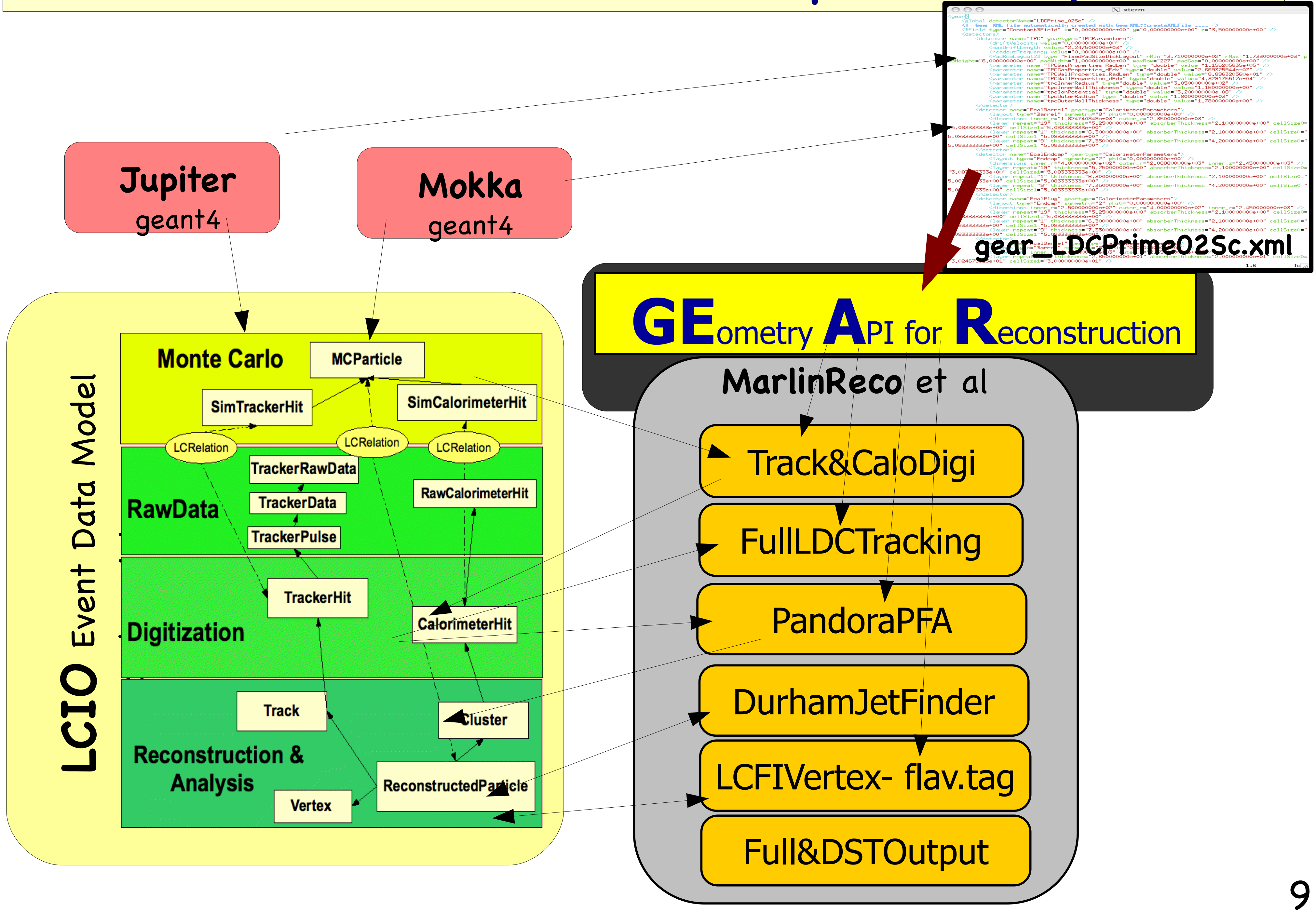
- ILD merged from GLD&LDC in 2007
- two independent frameworks existed the 2 regions
- use both for LOI detector optimization

-> LCIO & GEAR provide basis for interoperability



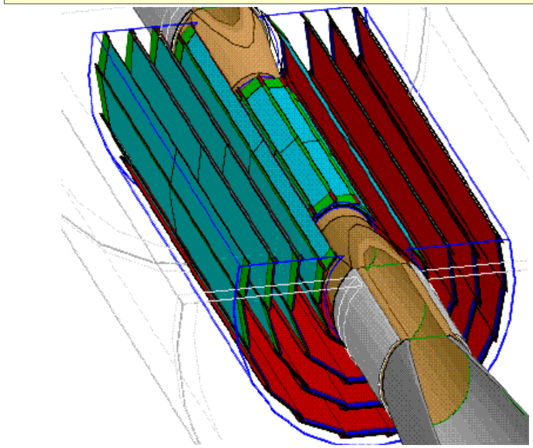


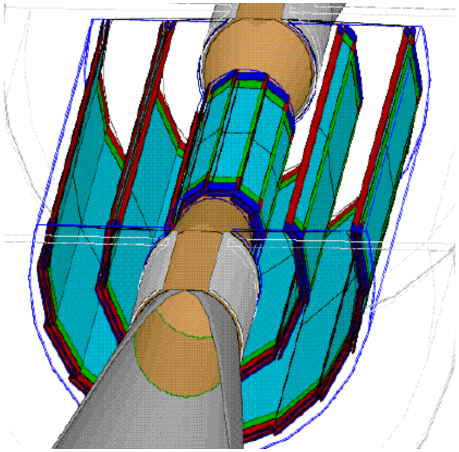
ILD software interoperability

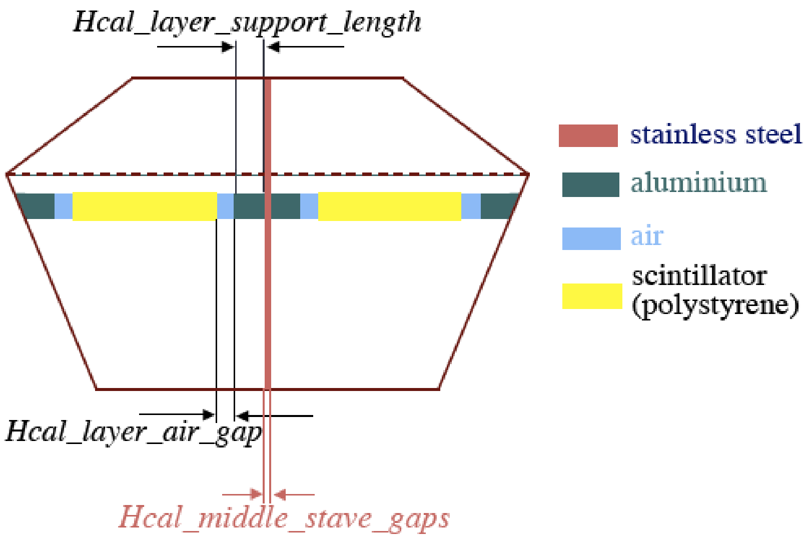


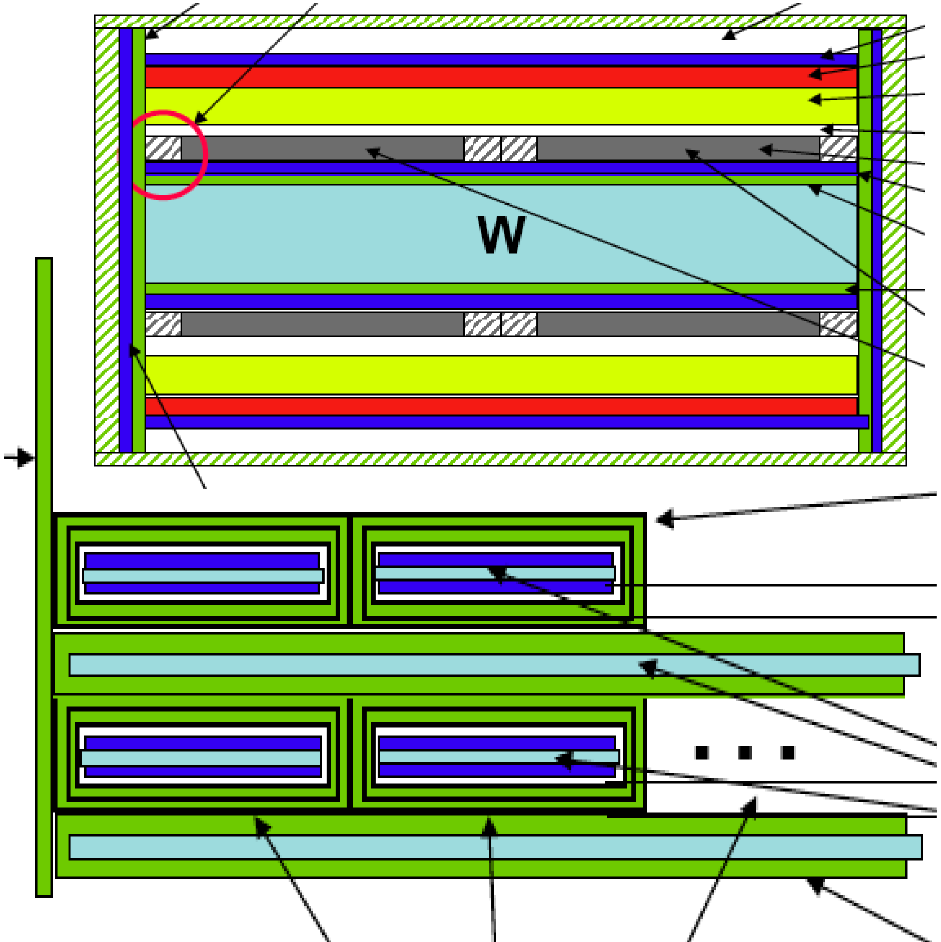
```
<gear>
    <global detectorName="LDCPrime_02Sc" />
    <!--Gear XML file automatically created with GearXML::createXMLFile ....-->
    <BField type="ConstantBField" x="0.000000000e+00" y="0.000000000e+00" z="3.500000000e+00" />
    <detectors>
        <detector name="TPC" geartype="TPCParameters">
            <driftVelocity value="0.000000000e+00" />
            <maxDriftLength value="2.247500000e+03" />
            <readoutFrequency value="0.000000000e+00" />
            <PadRowLayout2D type="FixedPadSizeDiskLayout" rMin="3.710000000e+02" rMax="1.733000000e+03" p
adHeight="6.000000000e+00" padWidth="1.000000000e+00" maxRow="227" padGap="0.000000000e+00" />
            <parameter name="TPCGasProperties_RadLen" type="double" value="1.155205835e+05" />
            <parameter name="TPCGasProperties_dEdx" type="double" value="2.669325944e-07" />
            <parameter name="TPCWallProperties_RadLen" type="double" value="8.896320560e+01" />
            <parameter name="TPCWallProperties_dEdx" type="double" value="4.329175517e-04" />
            <parameter name="tpcInnerRadius" type="double" value="3.050000000e+02" />
            <parameter name="tpcInnerWallThickness" type="double" value="1.160000000e+00" />
            <parameter name="tpcIonPotential" type="double" value="3.200000000e-08" />
            <parameter name="tpcOuterRadius" type="double" value="1.800000000e+03" />
            <parameter name="tpcOuterWallThickness" type="double" value="1.780000000e+00" />
        </detector>
        <detector name="EcalBarrel" geartype="CalorimeterParameters">
            <layout type="Barrel" symmetry="8" phi0="0.000000000e+00" />
            <dimensions inner_r="1.824740849e+03" outer_z="2.350000000e+03" />
            <layer repeat="19" thickness="5.250000000e+00" absorberThickness="2.100000000e+00" cellSize0=
"5.083333333e+00" cellSize1="5.083333333e+00" />
            <layer repeat="1" thickness="6.300000000e+00" absorberThickness="2.100000000e+00" cellSize0="
5.083333333e+00" cellSize1="5.083333333e+00" />
            <layer repeat="9" thickness="7.350000000e+00" absorberThickness="4.200000000e+00" cellSize0="
5.083333333e+00" cellSize1="5.083333333e+00" />
        </detector>
        <detector name="EcalEndcap" geartype="CalorimeterParameters">
            <layout type="Endcap" symmetry="2" phi0="0.000000000e+00" />
            <dimensions inner_r="4.000000000e+02" outer_r="2.088800000e+03" inner_z="2.450000000e+03" />
            <layer repeat="19" thickness="5.250000000e+00" absorberThickness="2.100000000e+00" cellSize0=
"5.083333333e+00" cellSize1="5.083333333e+00" />
            <layer repeat="1" thickness="6.300000000e+00" absorberThickness="2.100000000e+00" cellSize0="
5.083333333e+00" cellSize1="5.083333333e+00" />
            <layer repeat="9" thickness="7.350000000e+00" absorberThickness="4.200000000e+00" cellSize0="
5.083333333e+00" cellSize1="5.083333333e+00" />
        </detector>
        <detector name="EcalPlug" geartype="CalorimeterParameters">
            <layout type="Endcap" symmetry="2" phi0="0.000000000e+00" />
            <dimensions inner_r="2.500000000e+02" outer_r="4.000000000e+02" inner_z="2.450000000e+03" />
            <layer repeat="19" thickness="5.250000000e+00" absorberThickness="2.100000000e+00" cellSize0=
"5.083333333e+00" cellSize1="5.083333333e+00" />
            <layer repeat="1" thickness="6.300000000e+00" absorberThickness="2.100000000e+00" cellSize0="
5.083333333e+00" cellSize1="5.083333333e+00" />
            <layer repeat="9" thickness="7.350000000e+00" absorberThickness="4.200000000e+00" cellSize0="
5.083333333e+00" cellSize1="5.083333333e+00" />
        </detector>
        <detector name="HcalBarrel" geartype="CalorimeterParameters">
            <layout type="Barrel" symmetry="8" phi0="0.000000000e+00" />
            <dimensions inner_r="[illegible]e+03" outer_z="2.35[illegible]e+03" />
            <layer repeat="[illegible]" thickness="2.650000000e+01" absorberThickness="2.000000000e+01" cellSize0=
"3.02467[illegible]e+01" cellSize1="3.000000000e+01" />
                                                                                1,6           To
```

ILD detector geometry in Mokka







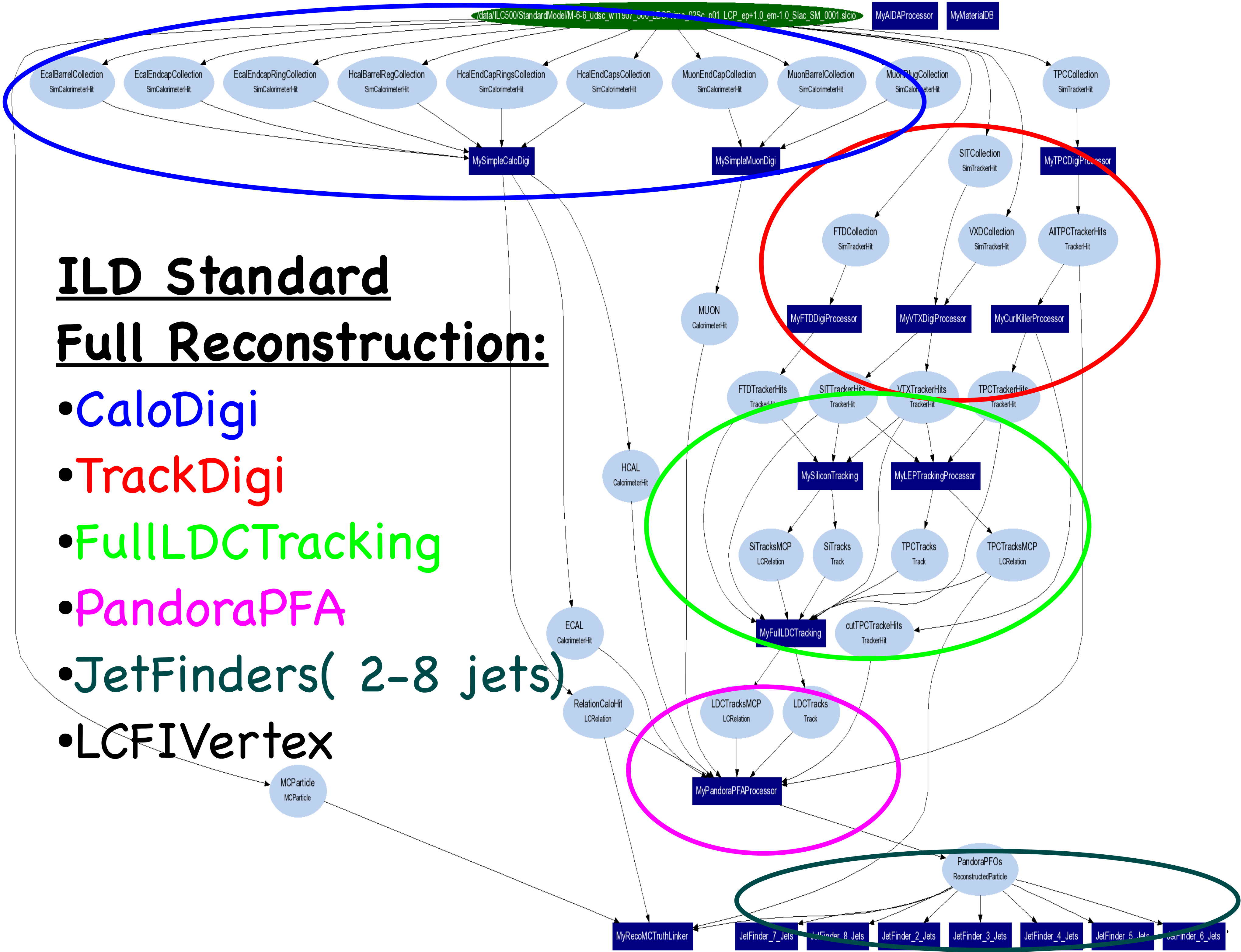


- Mokka allows various levels of detail when defining the detector geometry
- engineering level of detail done for ILD_00 :
  - get material budget right
  - get (in)efficiencies right
  - important for estimating PFA performance

Frank Gaede, TILC09, Tsukuba, 16-21 April 2009

ILD Standard Full Reconstruction:

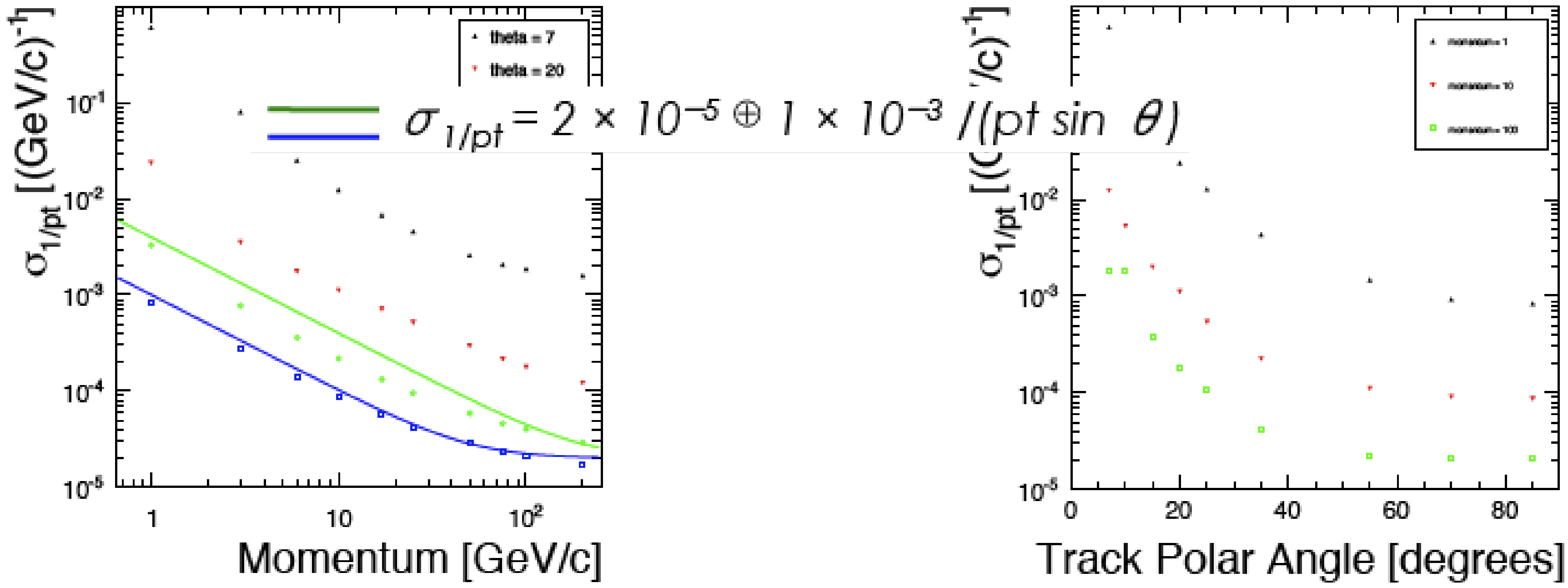
- CaloDigi
- TrackDigi
- FullLDCTracking
- PandoraPFA
- JetFinders(2-8 jets)
- LCFIVertex



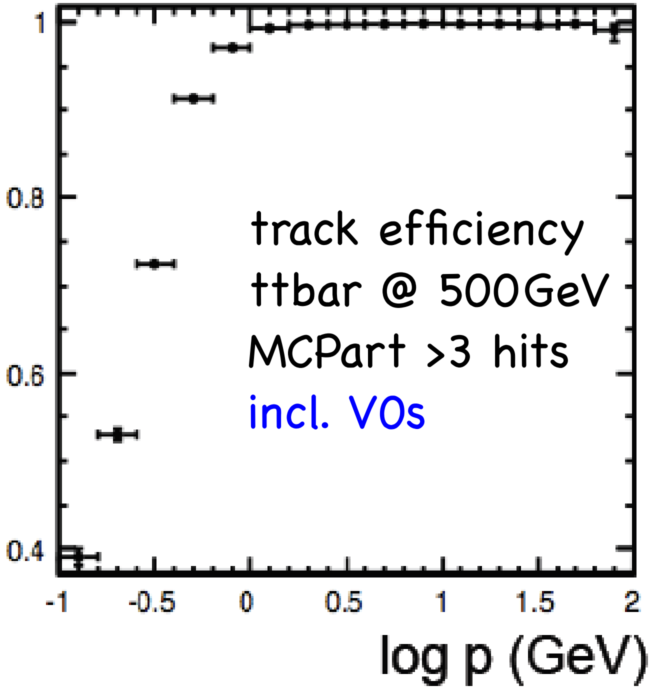
Digitization strategy

- VXD, SIT, FTD, SET, ETD Silicon hits
  - smearing of 3d space points (SimTrackerHits) according to envisaged detector resolutions
    - as established by R&D groups
  - also more detailed digitizers exist for Silicon detectors for dedicated studies
- TPC hits
  - smearing of 3d space points (SimTrackerHits) taking into account drift distance, polar and azimuthal angle of track
    - parameterization from TPC R&D groups
- Ecal,Hcal,Lcal,beamcal,LHcal, Muon Calo hits
  - calibration (single particle resolution)

MarlinReco - FullLDCTracking



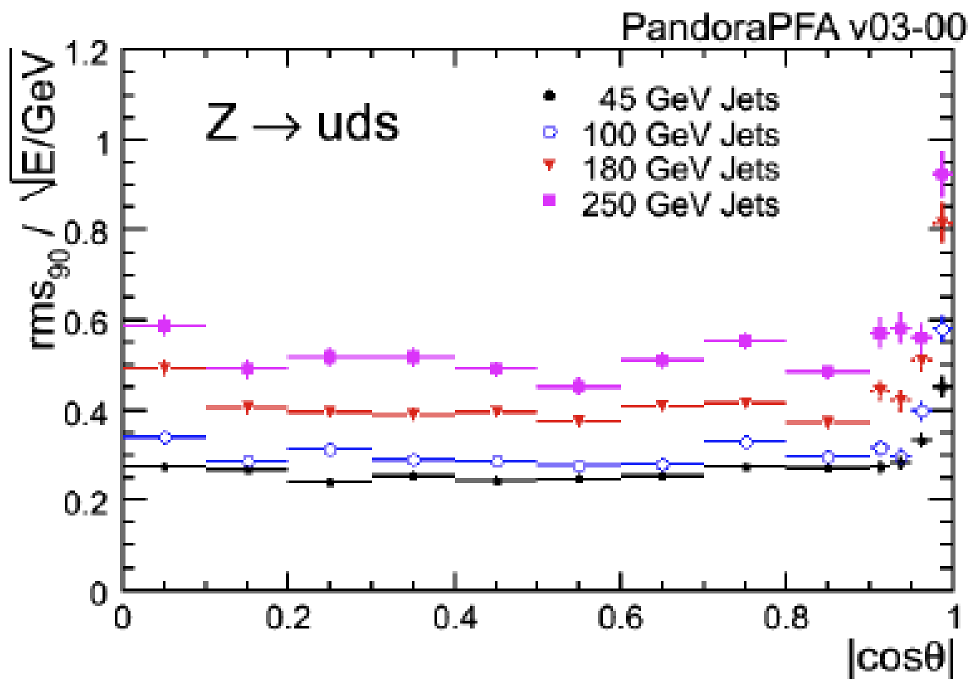
$$\sigma_{1/pt} = 2 \times 10^{-5} \oplus 1 \times 10^{-3} / (pt \sin \theta)$$



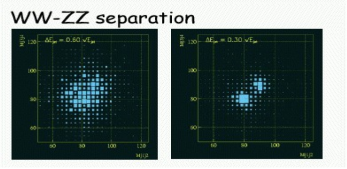
track efficiency
ttbar @ 500GeV
MCPart >3 hits
incl. V0s

- VTX, SIT, FTD: standalone tracking – track finding and Kalman-Fittter
- TPC: standalone Kalman-Filter based tracking (wraped LEP code)
- LDCTracking
  - combine tracks elements
  - find loopers
  - refit w/ Kalman-Filter

particle flow: PandoraPFA



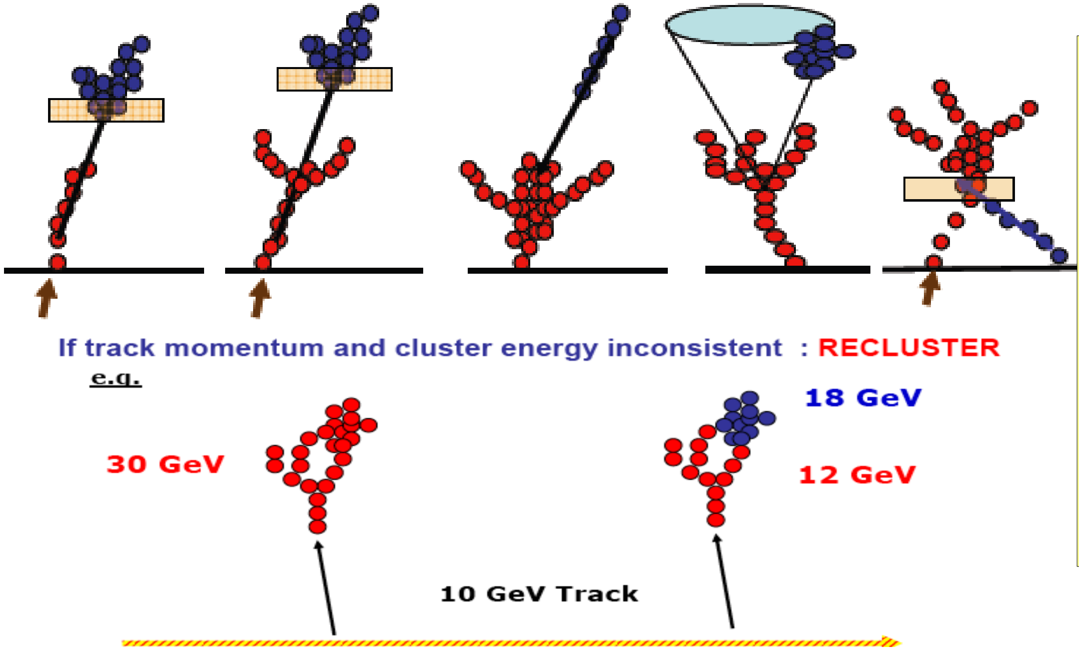
★For a Gauge boson mass resolution of order $\Gamma_{W/Z}$



$Z \rightarrow u\bar{u}, d\bar{d}, s\bar{s}$ decays at rest

|cosθ|<0.7

E_j	$\sigma(E_{jj})$	$\sigma(E_{jj})/\sqrt{E_{jj}}$	$\sigma(E_j)/E_j$
45 GeV	2.4 GeV	25 %	3.7 %
100 GeV	4.1 GeV	29 %	2.9 %
180 GeV	7.5 GeV	40 %	3.0 %
250 GeV	11.1 GeV	50 %	3.2 %

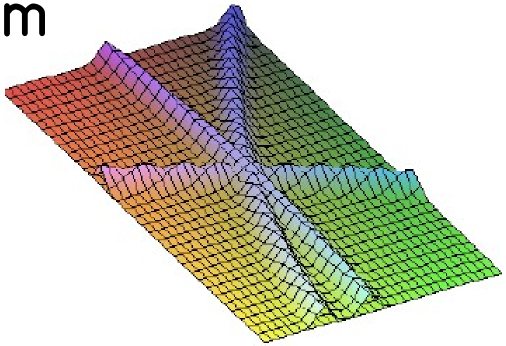


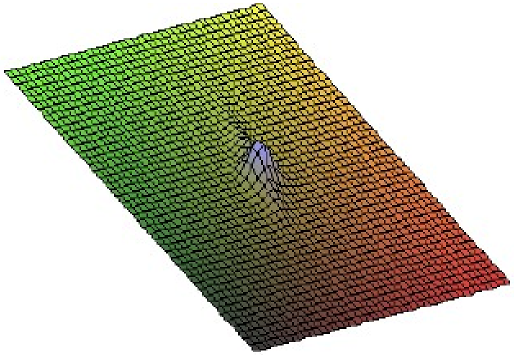
If track momentum and cluster energy inconsistent : RECLUSTER

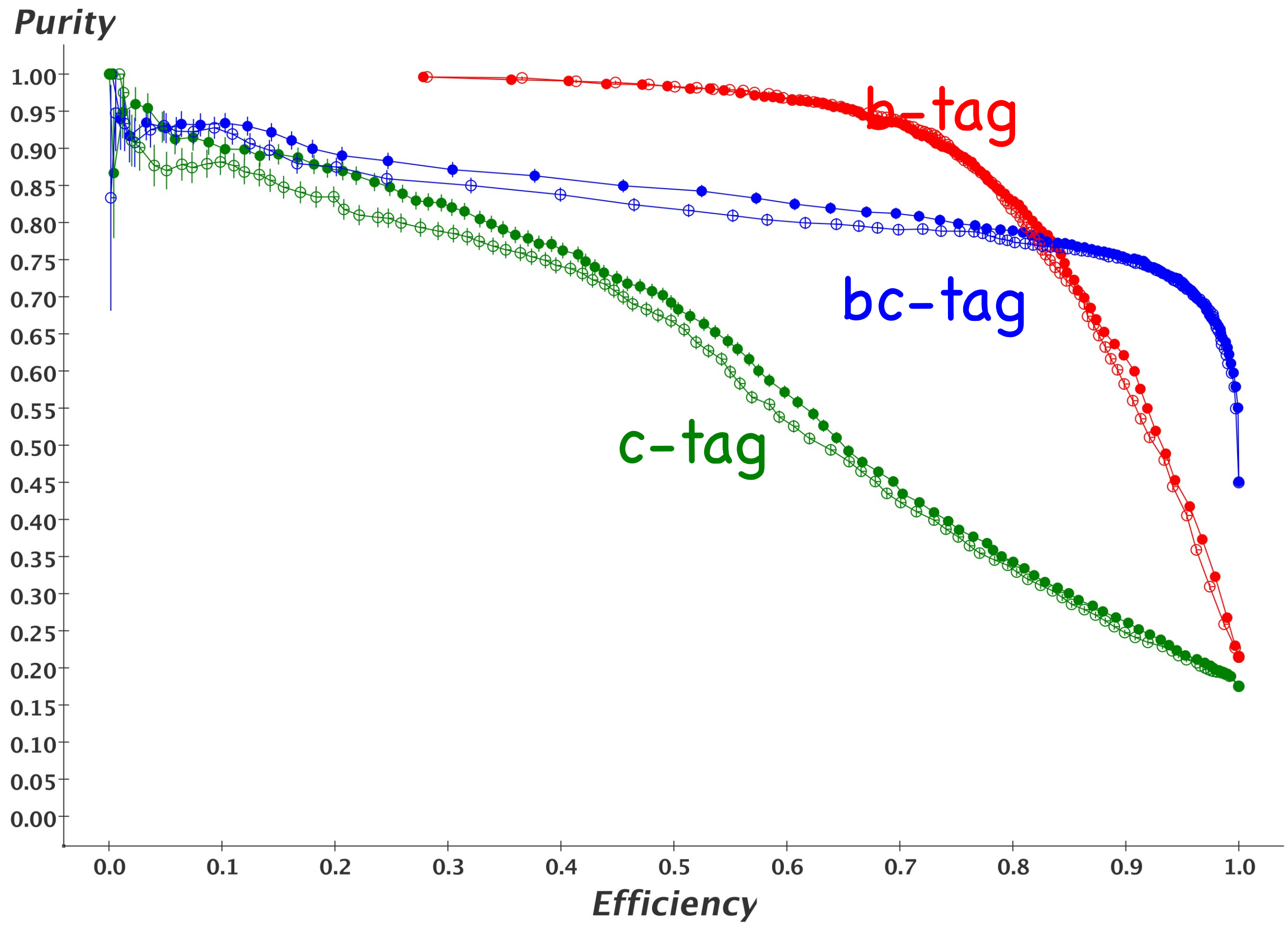
- PandoraPFA
  - best Particle flow for ILC to date
  - used in several studies for detector optimization
  - demonstration of PFA concept for the ILC

LCFIVertex

- Implementation of ZVTOP vertex finding algorithm
- Heavy-Flavor Tag based on neural networks
- Vertex-Charge for b and c jets







Frank Gaede, TILC09, Tsukuba, 16-21 April 2009

LOI Monte Carlo mass production

- massive production of Monte Carlo events needed for ILD optimization (based on GLD/LDC) and physics performance studies for current ILD model
- use LCG GRID resources (DESY, in2p3, UK,...)
  - have developed Grid job submission scripts, monitoring, web based data catalogues,...
- produced >50 M events w/ Mokka (geant4) - fully reconstructed w/ MarlinReco, PandoraPFA etc
  - LOI benchmark reactions (~500 1/fb)
  - corresponding SM sample (WHIZARD, generated at SLAC (DESY))

results: M.Thomson's plenary talk

- ILD software (incl.geant4) ran very stable - GRID sometimes flaky
- biggest bottle neck: job submission (needs work...)

Plans for ILD software

- ILD has software mature software tools that have been used successfully for the LOI
- next steps:
- create a common ILDsoft framework - based on LCIO, Marlin with 'goodies' from JSF framework
- for this need to investigate interface to ROOT
  - for user analyses based on macros/trees, I/O,...
- also some long planned and requested developments have been put on hold, due to LOI and limited manpower:
  - improve LCIO
  - improve the geometry system

collaboration w/ other groups is highly welcome

Improving LCIO -> LCIOv2

- LCIO is persistency and data model - separated through abstract interface
  - -> both can be developed independently !

- improving the persistency
  - I/O performance
  - direct access
  - splitting&merging files
  - allow for simple streaming of user defined classes (testbeam hardware)
  - evaluate using ROOT-I/O

- improving the data model:
  - allow multiple fits for Tracks
  - provide specialized TrackerHits, e.g. for strips
  - improved relations !?
  - learn from JSF data model...
  - user feedback needed !

it is probably time to abandon F77 !

Improving the geometry description

- Mokka-MySQL being the leading system not optimal
- should have standalone geometry system - for
  - simulation, reconstruction, analysis, event displays
  - provide interfaces with the appropriate level of detail at the various stages
  - based on common standards, e.g. GDML
- allow for smooth transition from existing tools (e.g. extend existing GEAR interfaces)
- unified/combined with conditions data base !?
- request from CALICE to extend GEAR...
- ideally this would be a common project for all concepts/groups working on ILC detector R&D !

Testing and Validation

- testing of software tools has been left to the judgement of the developer
- ideally we should have an automated test system (run at nightly build/on cvs commits)
  - possibly unit tests?
  - integration tests
- also validation of 'physics performance':
  - checkplots, resolutions , overlaps?,....
- -> would have saved us some work in the past

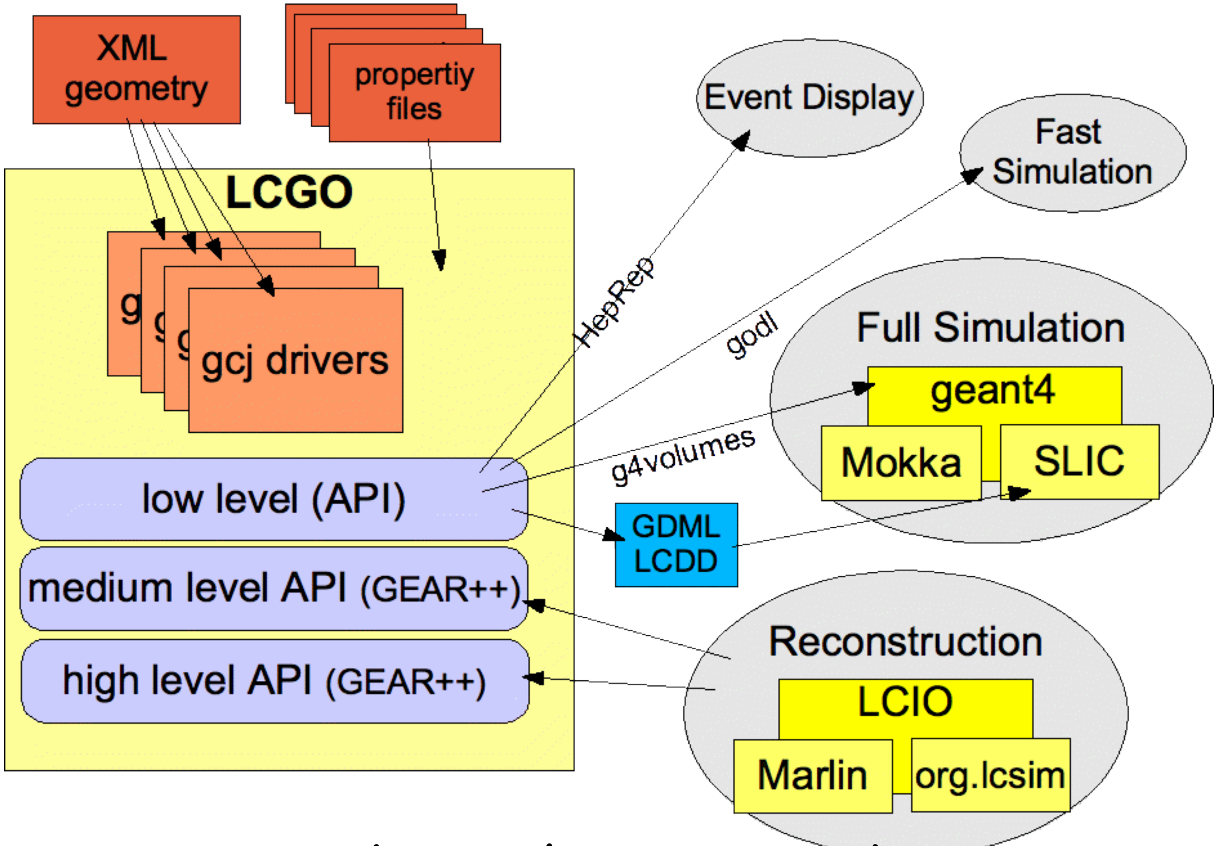
Summary & Outlook

- ILD software tools successfully used for LOI
- will move towards one software framework - based on LCIO, Marlin with goodies from JSF
- planned improvements and developments
  - automated test and validation system
  - investigate usage of ROOT
  - LCIOv2 - data model and persistency
  - geometry
- ILD is open for collaboration on software tools and we would welcome any new group to join in on using (and improving) LCIO and possibly work on a common geometry system

additional material

LCGO geometry tool - proposal

- driver based approach a la Mokka
- MySQL DB replaced by xml files



- LCGO - a planned geometry toolkit (DESY/SLAC 2006)
  - based on geometry drivers - written in JAVA !
  - use gcj-compiler to compile to binary & interface with C++
- issues with performance - 4 times slower than C++ (2007)
- -> could look into implementing similar concept in C++
- also investigate existing packages TGeo, VGeometry,...