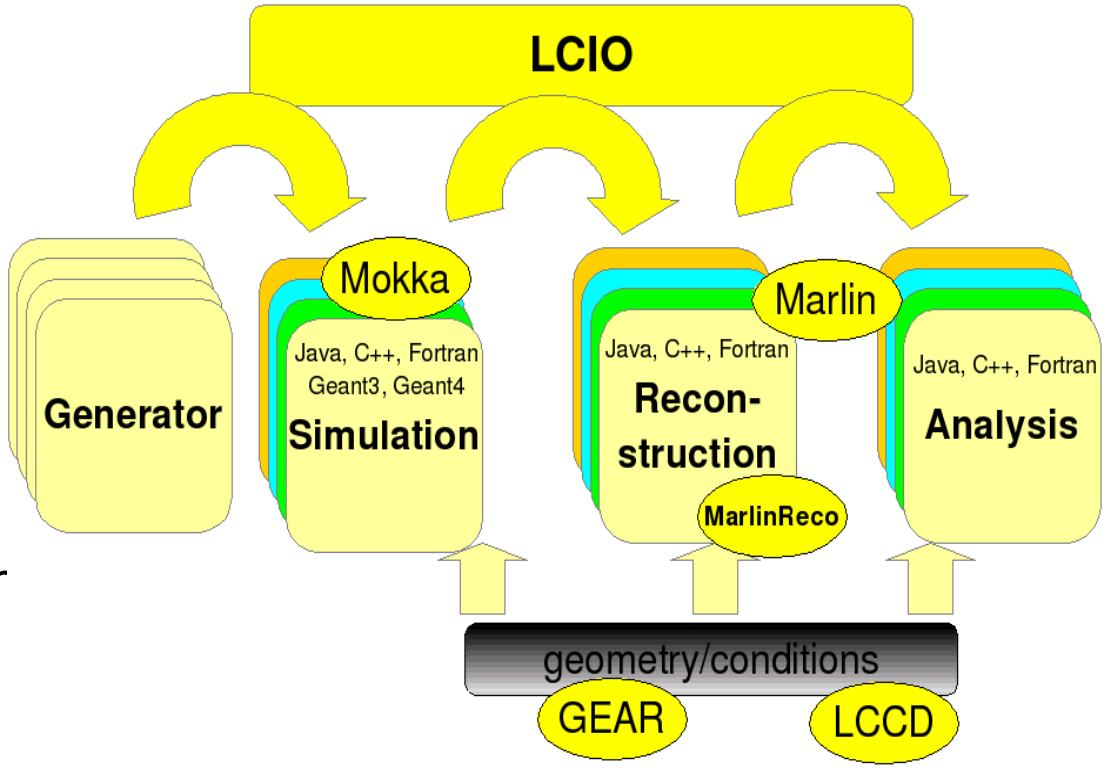
LDC Core Software

Status and new developments

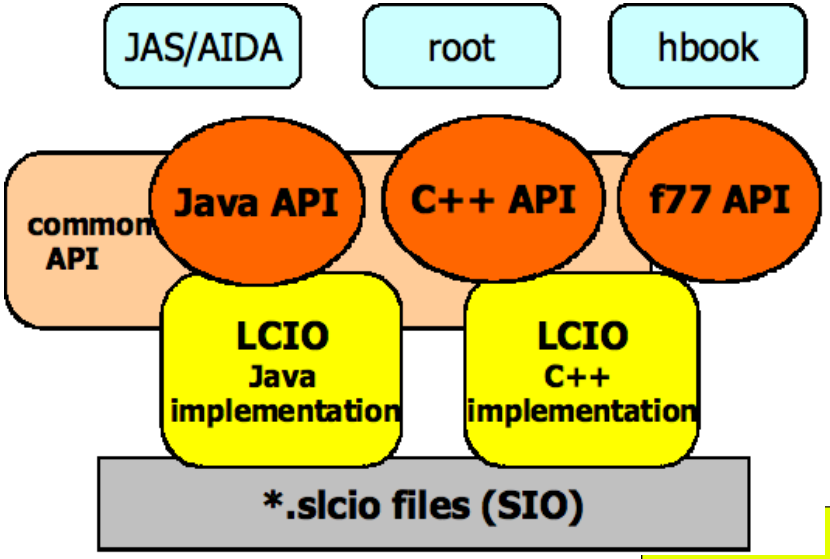
Frank Gaede
DESY
ILC Software Workshop,
Orsay, 2-4 May 2007

Outline

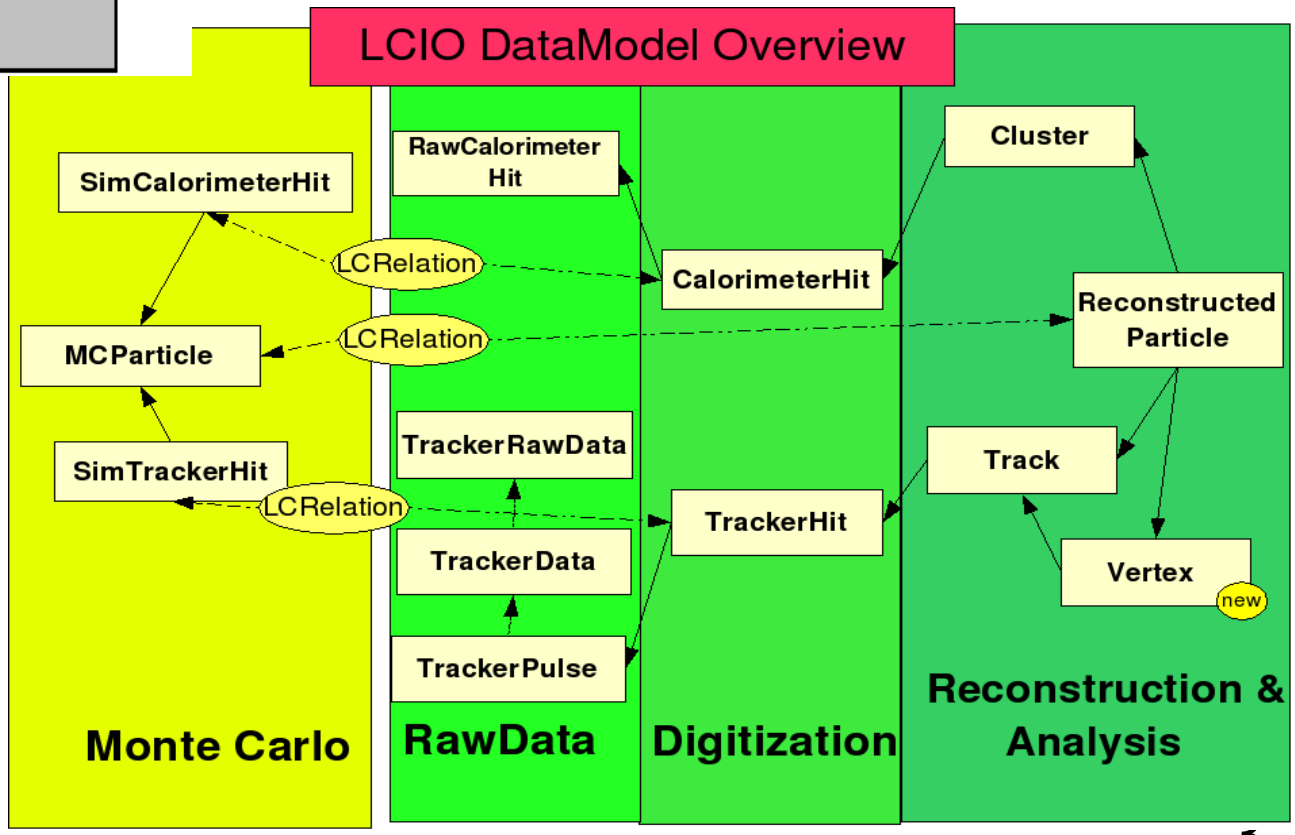
- Core software tools - status and latest developments
  - LCIO
    - data model & persistency
  - Marlin
    - C++ application framework
  - LCCD
    - conditions data toolkit
  - GEAR
    - geometry description
  - MarlinReco
    - Marlin based reconstructior
  - ilc sw-installation
- Summary



LCIO



international standard persistency & datamodel for ILC software



current release:
v01-08-02

LCIO Status

- LCIO v01-08
  - new Vertex class
  - C++ runtime (user) extensions and relations
  - new java command line tool
- LCIO v01-08-01
  - patch release – bug fixes
    - storing large arrays (SIO/C++)
    - enforce valid collection names
- LCIO v01-08-02
  - patch release – bug fixes
    - several issues with templates and gcc 4.x fixed
    - introduced cmake support (experimental)

LCIO runtime extensions (C++)

- long pending user request:
  - attach user objects to LCObjects
  - fast and easy creation of links (relations) between various LCObject subtypes, eg. TrackerHits and Track
- features
  - extension of the object with arbitrary (even non-LCObject) classes
  - extension of single objects or vectors, lists of objects
  - optionally ownership is taken for extension objects (memory management)
  - bidirectional relations between LCObjects
    - one to one
    - one to many
    - many to many

LCIO runtime extensions

extensions and relations identified through a tagging class T

```
// a simple int extension
struct Index : LCIntExtension<Index> {} ;

// a many to many relationship between MCParticles
struct ParentDaughter : LCNToNRelation<ParentDaughter,MCParticle,MCParticle>
//..
 MCParticle*  mcp = dynamic_cast<MCParticle*>( mcpcol->getElementAt(i) ) ;
//..

 mcp->ext<Index>() = i;    // set an int


 const MCParticleVec& daughters = mcp->getDaughters() ;

 for(unsigned j=0 ; j< daughters.size()   ; j++ ){

   // ---- set biderctional relation
   add_relation<ParentDaughter>( mcp, daughters[j] ) ;
 }

//-----------------------------------------------------

 cout << " myindex = " << mcp->ext<Index>  << endl ;

 ParentDaughter::to::rel_type daulist =  mcp->rel<ParentDaughter::to>() ;

 for( ParentDaughter::to::const_iterator idau = daulist->begin();
   idau != daulist->end(); ++idau){

    cout << (*idau)->ext<Index>() << ", " ;
  }
  cout << endl ;
```

for extensions use ext<T>()
for relations use rel<T>

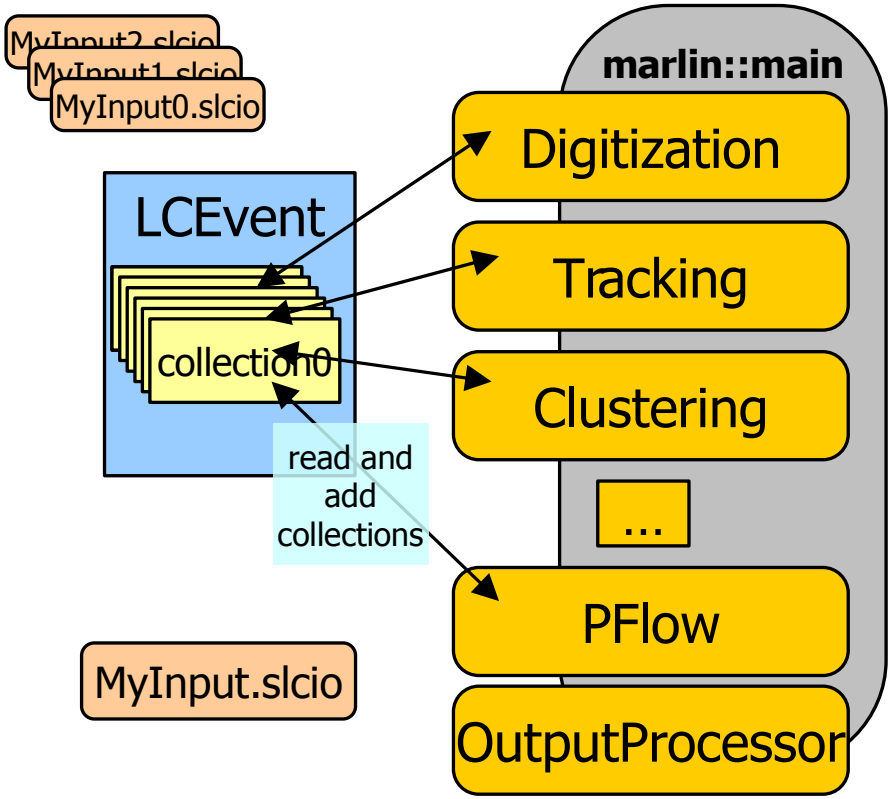
LCIO plans

- working on new C++ implementation of SIO
- aim:
  - provide direct access to events
  - allow splitting of event data across files
  - improve I/O performance
  - allow for persistency of user defined classes
    - needed by DAQ systems
  - work in progress...

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
- LCIO as transient data model
- software modules: Processors
- provides main program
- xml steering files
  - program flow
  - per processor and global variables
- **Plug&Play** of processors



Marlin recent developments v00-09-07

- MarlinGUI (v00-09-06)
- creation of flow charts (.dot files)
  - processors with input and output collections (B. Jeffrey)
- new logging mechanism
  - display a message depending on the verbosity level: DEBUG, MESSAGE, WARNING, ERROR
- optionally use working directory for building (see tutorial...)
  - set $MARLINWORKDIR to scratch directory
    - if not set $MARLIN will be used
- added support for cmake (experimental)
- support for runtime plugins (see tutorial...)
  - shared libraries with processors loaded at program start up
  - no relinking necessary

example: MarlinGUI

Marlin GUI

File

List of all Collections Found in LCIO Files

	Name	Type
1	MCParticle	MCParticle
2	ecal02_EcalBarrel	SimCalorimeterHit
3	hcalFeScintillator_HcalBa...	SimCalorimeterHit
4	sit00_SIT	SimTrackerHit
5	tpc04_TPC	SimTrackerHit
6	vxd00_VXD	SimTrackerHit
7	LumiCalS_LumiCal	SimCalorimeterHit
8	MCParticle	MCParticle
9	SEcal01_EcalBarrel	SimCalorimeterHit
10	SEcal01_EcalEndcap	SimCalorimeterHit
11	SHcal01_HcalBarrelEnd	SimCalorimeterHit
12	SHcal01_HcalBarrelReg	SimCalorimeterHit
13	SHcal01_HcalEndCaps	SimCalorimeterHit
14	STpc01_FCH	SimTrackerHit
15	STpc01_TPC	SimTrackerHit

LCIO Files

muons.slcio
zpole1.slcio

Add New LCIO File

Remove LCIO File

View Options

Hide Inactive Processors

Hide Active Processor Errors

Active Processors

	Name	Type
1	MyAIDAProcessor	AIDAProcessor
2	MyVTXDigiProcessor	VTXDigiProcessor
3	MyFTDDigiProcessor	FTDDigiProcessor
4	MyTPCDigiProcessor	TPCDigiProcessor
5	MyCheckPlotsBenjamin	CheckPlotsBenjamin

Active Processor Operations

Add New Processor

Edit Selected Processor

Delete Selected Processor

Deactivate Selected Processor

Move Selected Processor Up

Move Selected Processor Down

Error Description from selected Processor

Some Collections are not available

Collection [ftd01_FTD] of type[FTDTrackerHit] is unavailable!!
* Following available collections of the same type were found:
-> Name: [ftd02_FTD] Type: [FTDTrackerHit] in processor with Name: [MyTestProcessor] and Type: [TestProcessor]

Collection [ftd02_FTD] of type[FTDTrackerHit] is unavailable!!
* Following inactive processors have a matching available collection:
-> Name: [MyTestProcessor] Type: [TestProcessor]
-> TIP: Activate the processor [MyTestProcessor] and set it before [MyFTDDigiProcessor]

Inactive Processors

	Name	Type
1	MyTestProcessor	TestProcessor
2	MySimpleCaloDigi	SimpleCaloDigi

Inactive Processor Operations

Add New Processor

Edit Selected Processor

Delete Selected Processor

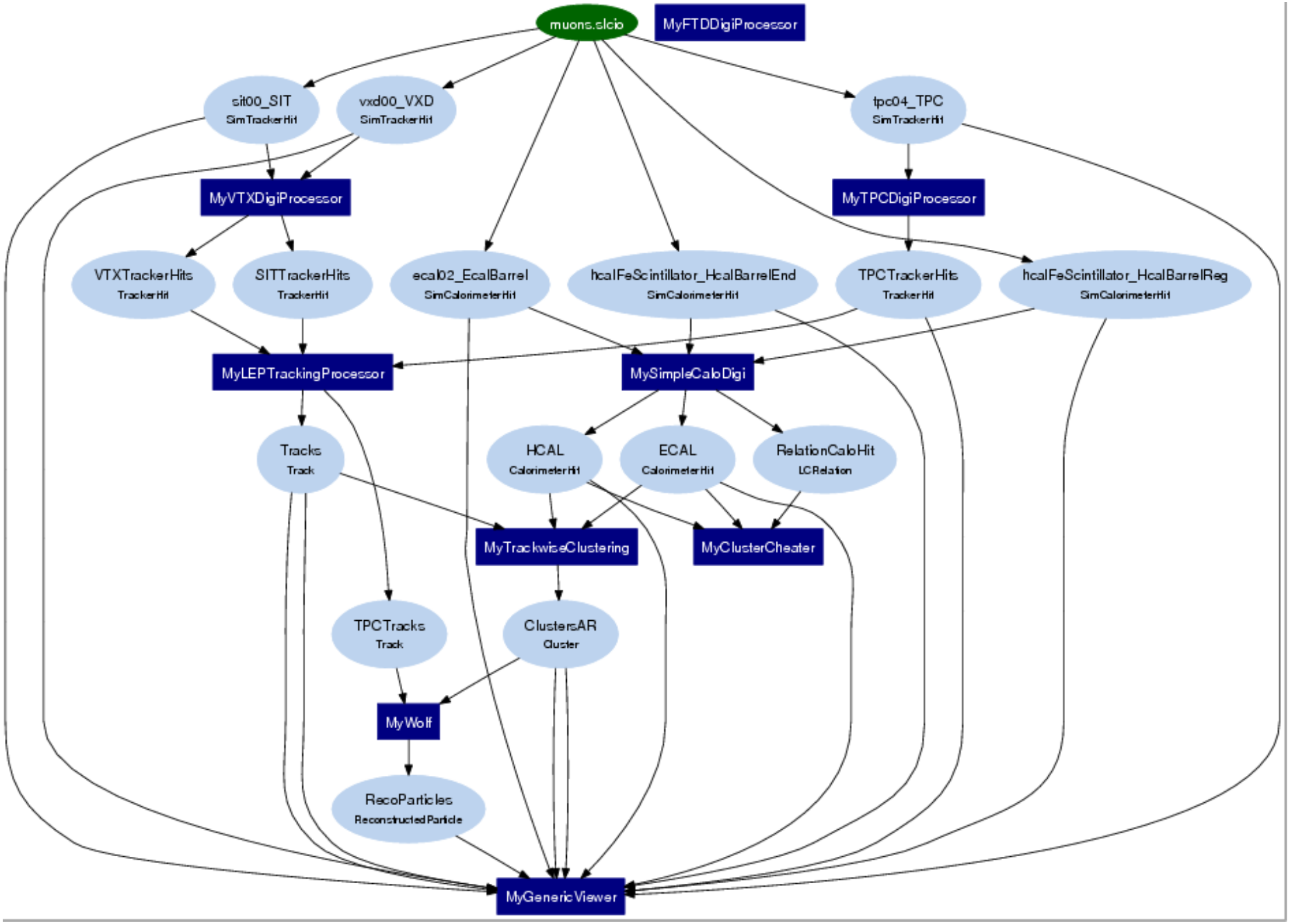
Activate Selected Processor

bin | Marlin GUI | Tue Oct 17, 16:41

see tutorial

author:
Jan Engels

example: Marlin Flow Chart



loggin mechanism

- Processor has a method message<VERB>() that can be called with one of DEBUG, MESSAGE, WARNING, ERROR
- output controled with global steering “Verbosity”
  - message<MESSAGE>(log()
  - << " processing event " << evt->getEventNumber()
  - << " in run " << evt->getRunNumber()
  -) ;
- DEBUG will not be compiled if not in MARLINDEBUG mode
  - -> no runtime overhead in production code

Marlin plans

- use plugin mechanism as default, ie.
  - all packages will be optionally loaded at runtime
  - no relinking necessary
  - could configure application as needed
  - issues:
    - existing static libraries, e.g. cernlib
- switch to cmake as default build tool
- work in progress...

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" typ
        <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

- geometry definition for reconstruction
- abstract interface per subdetector
  - VXD, TPC, Ecal, Hcal
- user defined parameters
- implementation uses XML files
- optionally create xml files from Mokka simulation
  - -> one source of geometry

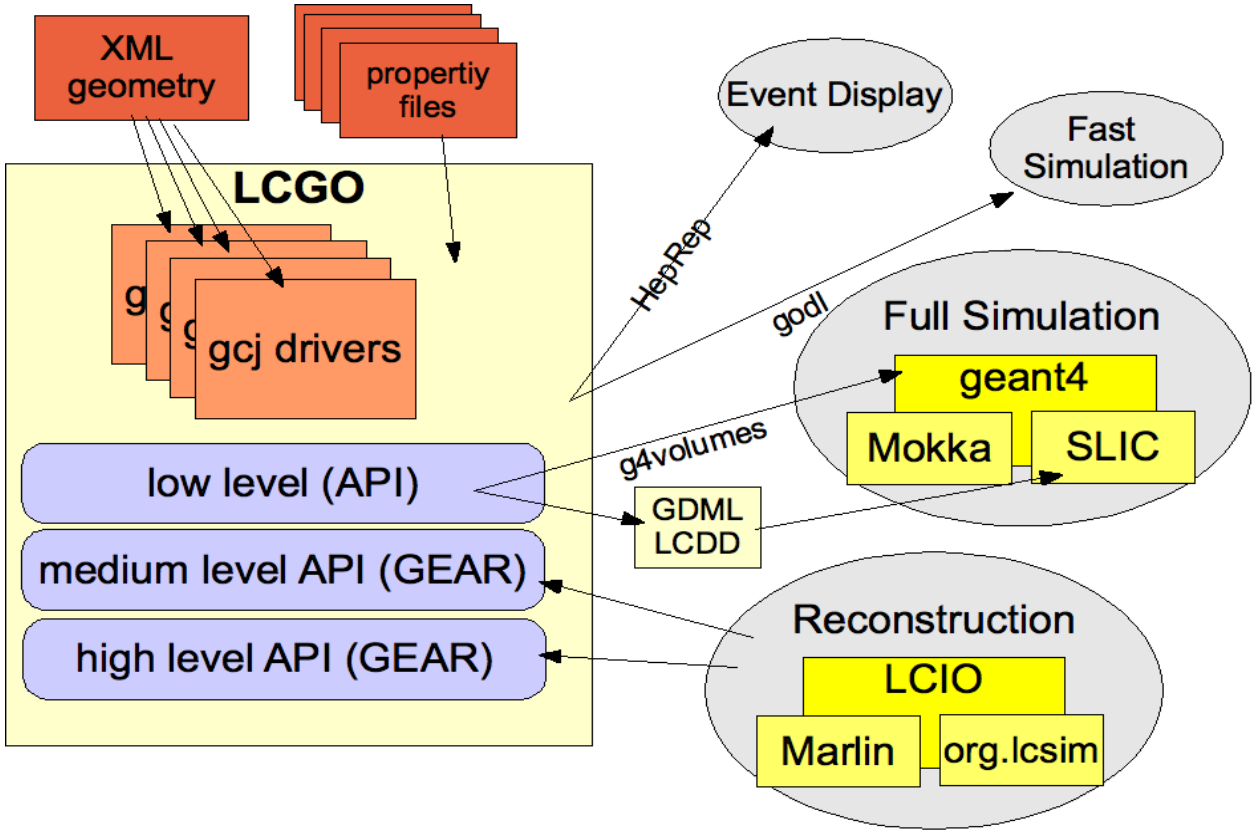
Gear status

- v00-05
- new: LCal (B.Pawlik) – v00-04-01
- added implementation of simple 3D vector with
  - scalar product, addition
  - coordinate transforms (cartesian, cylindrical and spherical).
  - automatic conversion to CLHEP and others (templates)
  - might also be used in a future LCIO release
  - Point2D and Point3D are now vectors !
- added global B field map (so far only constant)
  - **double bfield = gearMgr->getBField().at(Vector3D(0,0,0)).z() ;**
  - could implement full field map if needed

all tracking/reconstruction code should use this !

outlook: LCGO a new geometry system

- common geometry to be used in all ILC frameworks
  - SLAC-DESY project – (of course open for all collaborators)
- goals for LCGO:
  - be at least as functional as existing systems (LCCD/SLIC, GEAR/Mokka)
  - enable smooth transition path from existing systems
  - encourage/increase interoperability between systems
  - have no known principle short comings: “everything should be possible”

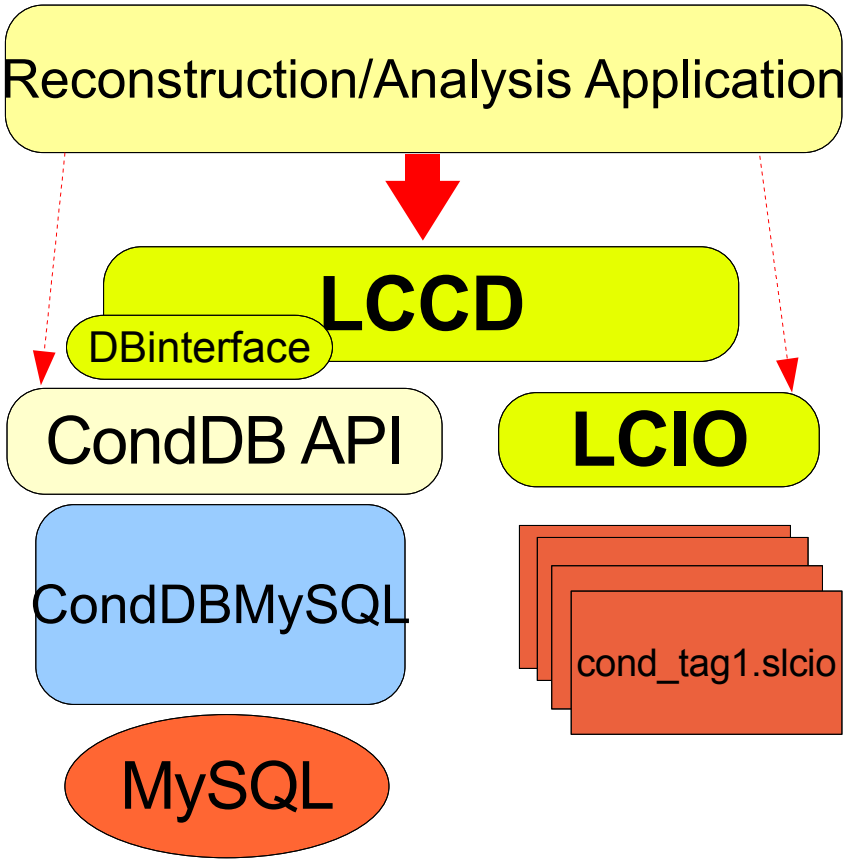


work in progress
- manpower issues

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)



- v00-03-05
- added cmake support
- made compatible w. SL4

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

MarlinReco

MarlinReco is a Marlin based **toolkit** providing reconstruction algorithms for the detector concept studies - packages:

- **TrackDigi**
  - TPCDigi
  - VTXDigi
- **CaloDigi**
  - LDCCaloDigi
- **Tracking**
  - LEPTracking (f77)
  - VTXTracking
  - TrackCheater
  - new FullLDCTracking
- **Clustering**
  - TrackwiseClustering
  - ClusterCheater
  - PhotonFinderKit new
- **Pflow**
  - Wolf
  - TrackBasedPFlow new
- **Analysis**
  - EventShapes
  - SatoruJetFinder

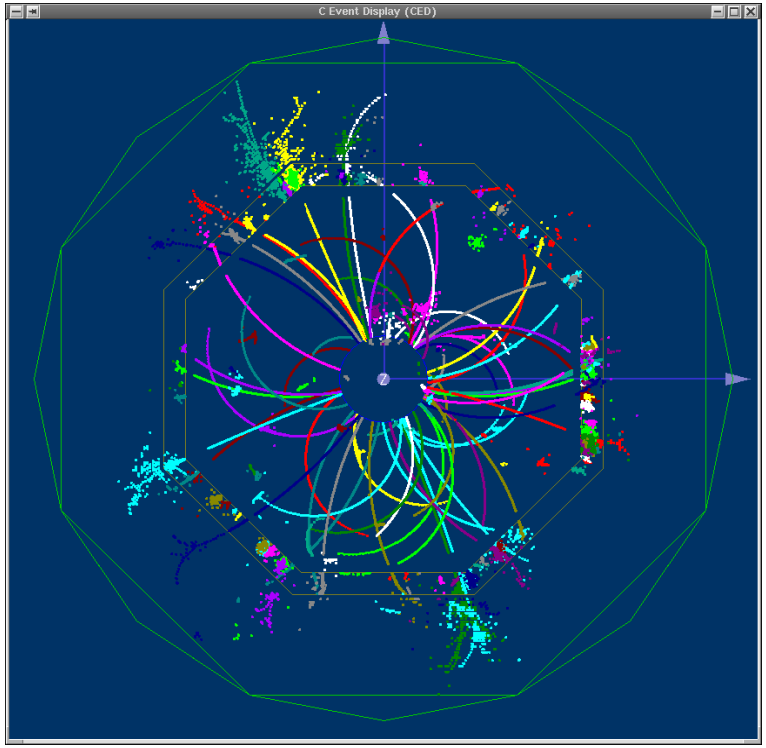
v00-03

- MarlinReco supports distributed development of reconstruction code
- packages can be seamlessly integrated with other packages, e.g.
  - **PandoraPFA** (M.Thomson)
  - **LCFIVertex** (S.Hillert/LCFI)
  - **SiliconDigi** (S.Shulga)

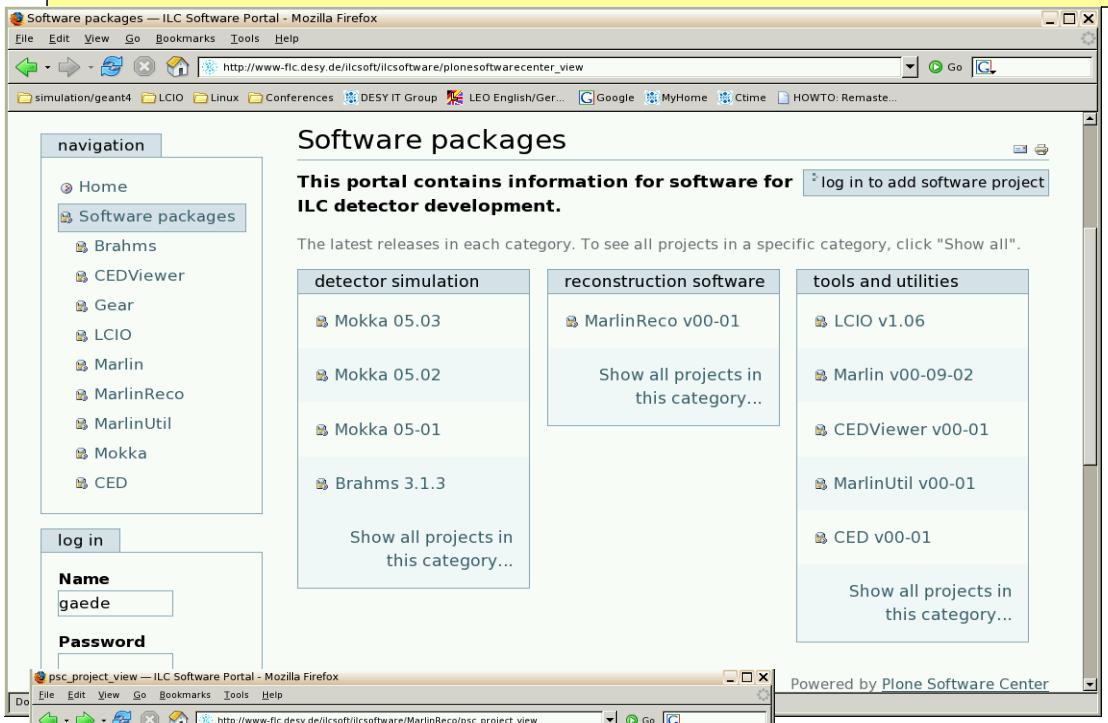
now part of MarlinReco cvs repository !

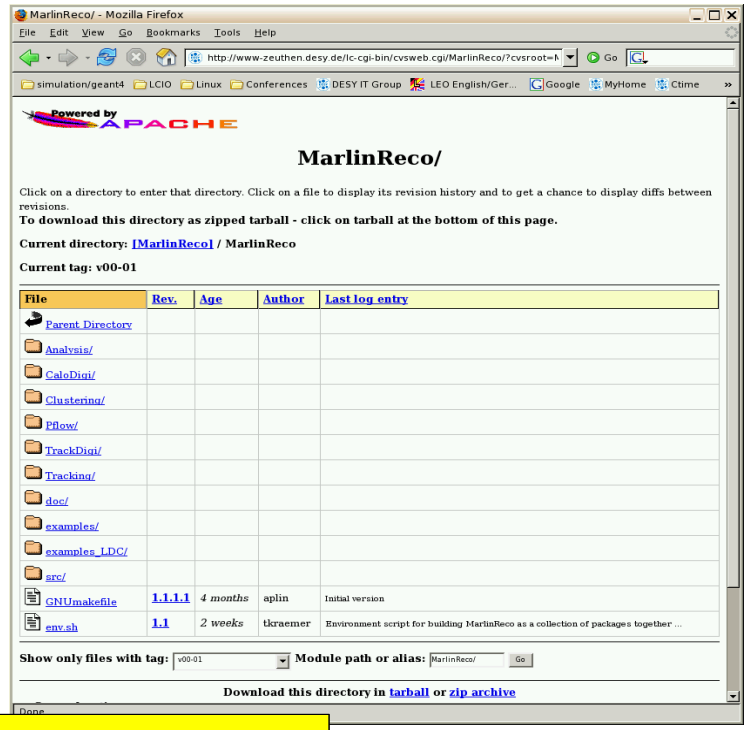
MarlinReco support packages

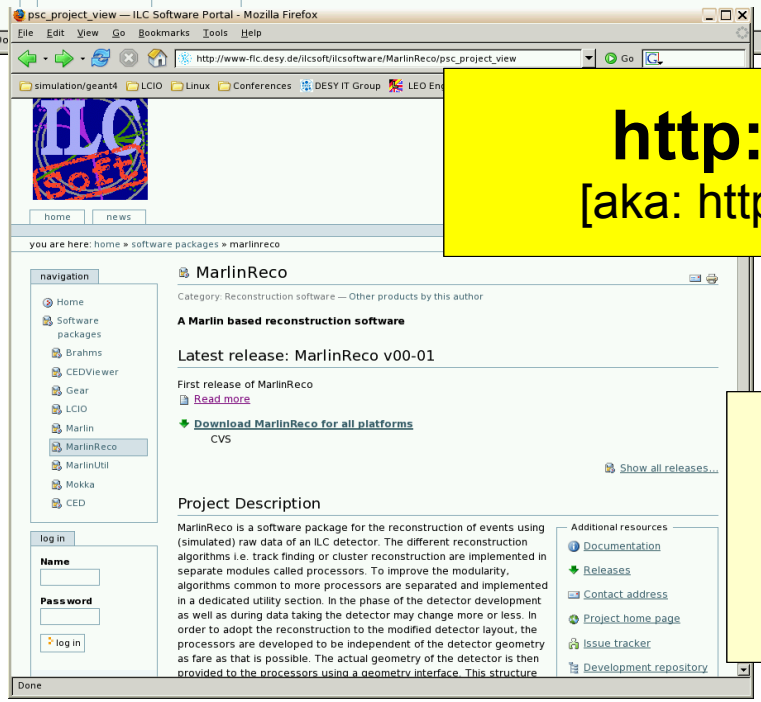
- MarlinUtil (O. Wendt, T.Kraemer,...) v00-03
  - Utility and Helper classes
    - helix fitter, cluster shapes,...
    - trajectory class / extrapolation
- RAIDA (T.Kraemer) v01-02
  - AIDA root implementation
- CED (A. Zhelezov) v00-01
  - event display based on GLUT/ OpenGL
  - client server architecture
- CEDViewer v00-02
  - event display client processors
    - CEDViewer, GenericViewer

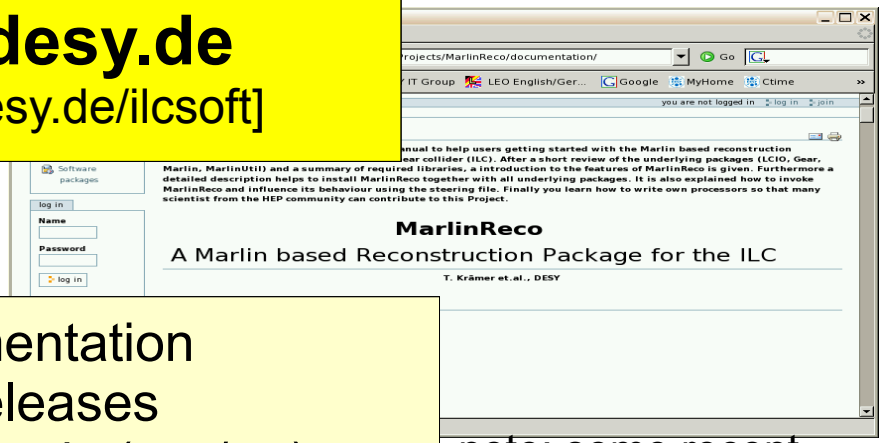


ILC software portal









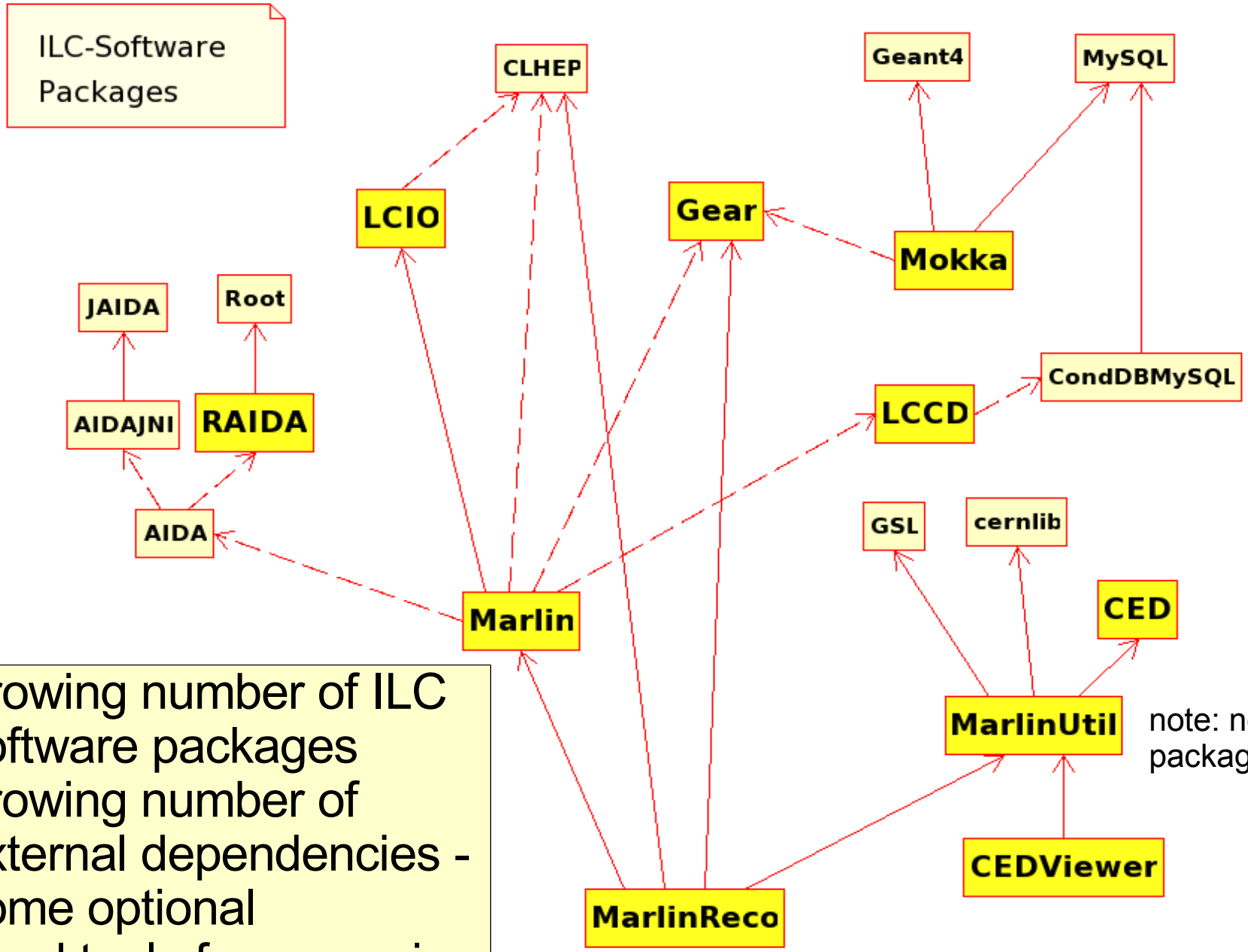
http://ilcsoft.desy.de

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

- documentation
- new releases
- downloads (cvs/tar)
- bug reports

note: some recent releases will be available after workshop

software package dependencies



note: not all packages shown !

- growing number of ILC software packages
- growing number of external dependencies - some optional
- need tools for managing

managing ILC (LDC) software

- ilcinstall script (next talk: J.Engels)

- versioned reference installations:
  - for standard plattforms **SL3 and SL4**
  - **/afs/desy.de/group/it/ilcsoft/v01-00**
  - defines set of tags that inter operate
  - allows to run binaries w/o installation
  - could link the core packages and only install some packages locally for development

ilcsoft release (SL3/SL4)

- /afs/desy.de/group/it/ilcsoft/v01-00/QT/4.2.2
- /afs/desy.de/group/it/ilcsoft/v01-00/**lccd/v00-03-05**
- /afs/desy.de/group/it/ilcsoft/v01-00/CondDBMySQL/CondDBMySQL_ILC-0-5-10
- /afs/desy.de/group/it/ilcsoft/v01-00/MySQL/5.0.26
- /afs/desy.de/group/it/ilcsoft/v01-00/**lcio/v01-08-02**
- /afs/desy.de/group/it/ilcsoft/v01-00/**gear/v00-05**
- /afs/desy.de/group/it/ilcsoft/v01-00/**CED/v00-01**
- /afs/desy.de/group/it/ilcsoft/v01-00/**RAIDA/v01-02**
- /afs/desy.de/group/it/ilcsoft/v01-00/root/5.08.00
- /afs/desy.de/group/it/ilcsoft/v01-00/java/1.5.0
- /afs/desy.de/group/it/ilcsoft/v01-00/cernlib/2006
- /afs/desy.de/group/it/ilcsoft/v01-00/CLHEP/2.0.2.2
- /afs/desy.de/group/it/ilcsoft/v01-00/gsl/1.8
- /afs/desy.de/group/it/ilcsoft/v01-00/**Marlin/v00-09-07**
- /afs/desy.de/group/it/ilcsoft/v01-00/**CEDViewer/v00-02**
- /afs/desy.de/group/it/ilcsoft/v01-00/**MarlinReco/v00-03**
- /afs/desy.de/group/it/ilcsoft/v01-00/**MarlinUtil/v00-03**
- /afs/desy.de/group/it/ilcsoft/v01-00/**LCFIVertex/v00-01**
- /afs/desy.de/group/it/ilcsoft/v01-00/**SiliconDigi/v00-01**
- /afs/desy.de/group/it/ilcsoft/v01-00/**PandoraPFA/v01-01**

Summary

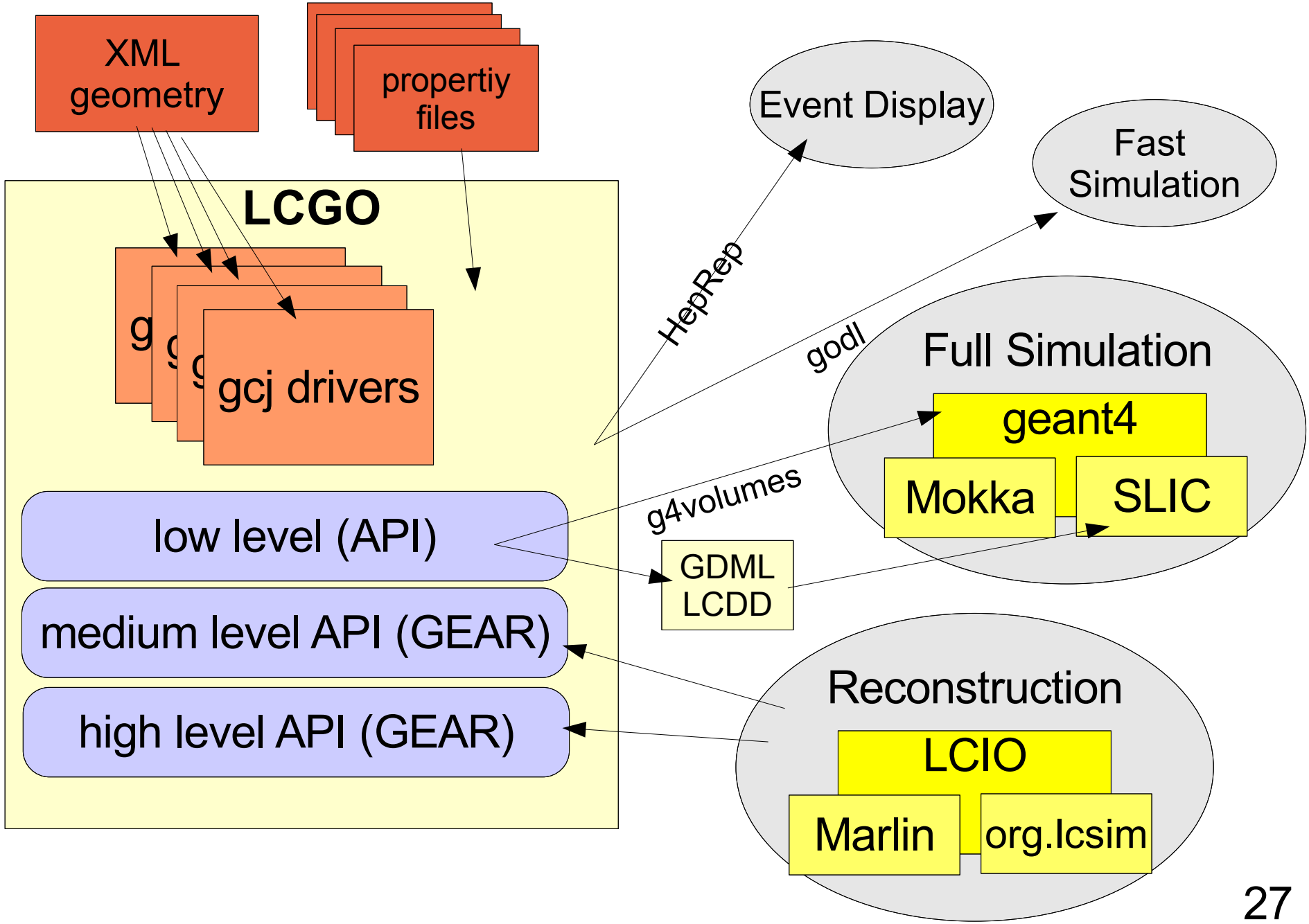
- recent developments in the ILC/LDC core software framework with (LCIO, Marlin, LCCD, GEAR):
  - runtime user extensions and relations in LCIO (C++)
  - plugin mechanism and logging method in Marlin
  - MarlinGUI
  - B field in Gear
  - versioned reference software installations in afs
  - ...
- plans – ongoing work
  - improve LCIO I/O
  - improve build process (cmake)
  - LCGO geometry framework
  - your requests/needs !?

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
  - work just started – aiming for summer 2007
- requirements/goals for LCGO:
  - be at least as functional as existing systems (LCCD/SLIC, GEAR/Mokka)
  - enable smooth transition path from existing systems
  - encourage/increase interoperability between systems
  - have no known principle short comings: "everything should be possible"
- key idea: combine best parts of the existing frameworks
  - xml/property files define parameters
  - java driver classes use parameters to build detector geometry
    - use **gcj** to compile java to true machine code for C++ frameworks
    - **gcjh** creates C++ header files
    - -> have identical code basis used in Java and C++ frameworks

LCGO implementation prelim.



some LCGO planned features

- extended GEAR interface (medium and high level)
  - tracking (and clustering PFA)
    - average material volumes
    - intersection with 'next' volume
    - dE/dx
    - field maps
    - access to volumes
    - extensions of detectors (a la gear)
      - e.g. #layers, thickness, width,..
- material database
- field maps
- properties (sampling fractions)
- readout properties
  - cellId <-> position
  - cellid range (noise simulation)
  - cell sizes
  - neighbors
- detailed geometry:
  - have geant4 like geometry classes with a one to one correspondence:
    - logical/physical volumes
    - placements
    - materials

ILC interoperable software chain