

slic & lcsim

Norman Graf (SLAC)

(for the ALCPG Simulation & Reconstruction WG)

HPS Software Meeting

April 7, 2011

LCD Simulation Mission Statement

- Provide full simulation capabilities for Lepton Collider physics program:
 - Physics simulations
 - Detector designs
 - Reconstruction and analysis
- Need flexibility for:
 - New detector geometries/technologies
 - Different reconstruction algorithms
 - Different machine environments
- Limited resources demand efficient solutions, focused effort.

Overview: Goals

- Facilitate contribution from physicists in different locations with various amounts of time available.
- Use standard data formats, when possible.
- Provide a general-purpose framework for physics software development.
- Develop a suite of reconstruction and analysis algorithms and sample codes.
- Simulate benchmark physics processes on different full detector designs.

Detector Performance Studies

- The ILC community recently finished a very intensive round of detector performance and optimization studies, culminating in the submission of LOI's and is engaged in preparing more detailed updates for the DBD in 2012.
- The CLIC community is currently engaged in an aggressive effort to provide a CDR later this year.
- The Muon Collider community will be using these tools for physics and detector studies.
- Provides an excellent opportunity for HPS to benefit from the very large investment in software and the lessons learned.
 - What worked
 - What didn't work

LCIO



ALCPG

SiD

ECFA-LC

LDC

ACFA-LC

GLD

slic

org.lcsim

Java

MOKKA

MarlinReco

C++

JUPITER

Satellites

root

LCIO

A world map with a blue background and light green landmasses. Three yellow rectangular boxes are overlaid on the map, each containing red text. The first box is over North America, the second over Europe and Africa, and the third over Asia and Australia.

ALCPG

SiD

ECFA-LC

LDC

ACFA-LC

GLD

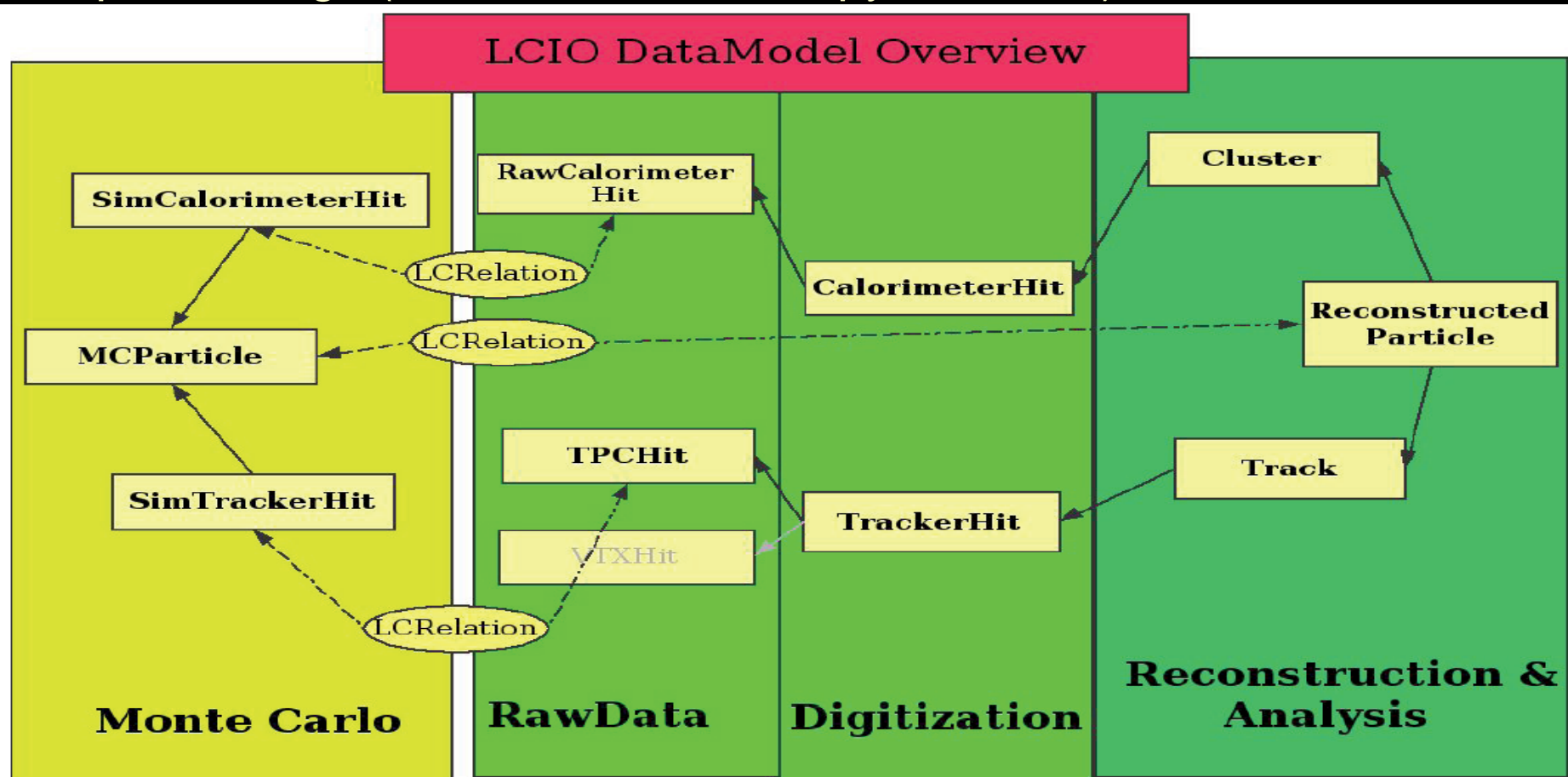
LCIO

Common Data Model

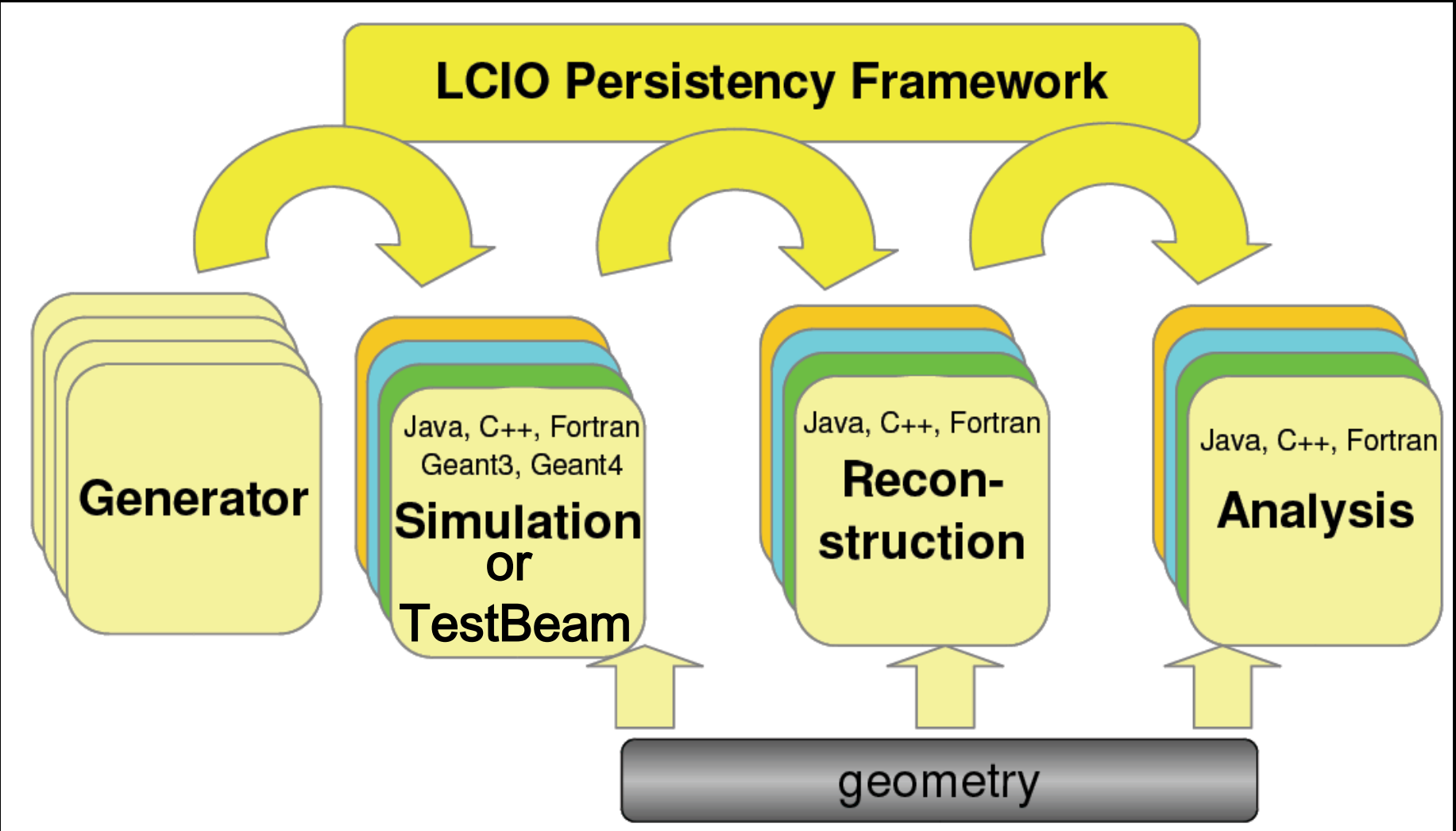
Common IO Format

LCIO Overview

- Object model and persistency format for HEP events
 - MC simulation
 - Test Beam data
 - Reconstructed Objects
- Multiple bindings (C++, Java, Fortran, python, root)



LCIO Overview



LCIO

- Direct access to events
 - Overlay of random background events
 - physics analysis using preselection on metadata
- ROOT dictionary
 - use LCIO classes in root macros
 - write simple ROOT trees
 - write complete LCIO events in one ROOT branch
 - comes at a cost, of course, slower than native access
- LCIO 2.0 will add requested user functionality.

LCIO Interoperability

- All three regional LCD simulators write LCIO
 - Cross-checks between data from different simulators
 - Read/write LCIO from
 - Fast MC / Full Simulation / Test Beams
 - Different detectors
 - Different reconstruction tools
 - Different frameworks, languages, operating systems
- Reconstruction also targets LCIO
 - Can run simulation or reconstruction in one framework, analysis in another, benefit from worldwide developer community.
 - Generate events in Jupiter, analyze in MarlinReco.
 - org.lcsim to find tracks in Java, LCFI flavor-tagging to find vertices using MarlinReco (C++) package, slicPandora to run Particle Flow Analysis.

LCIO Event Browser

- Fully supported within JAS.
- Open any LCIO file, browse collections and objects.
- Traverse MC particle hierarchies
- Print, sort, analyze.

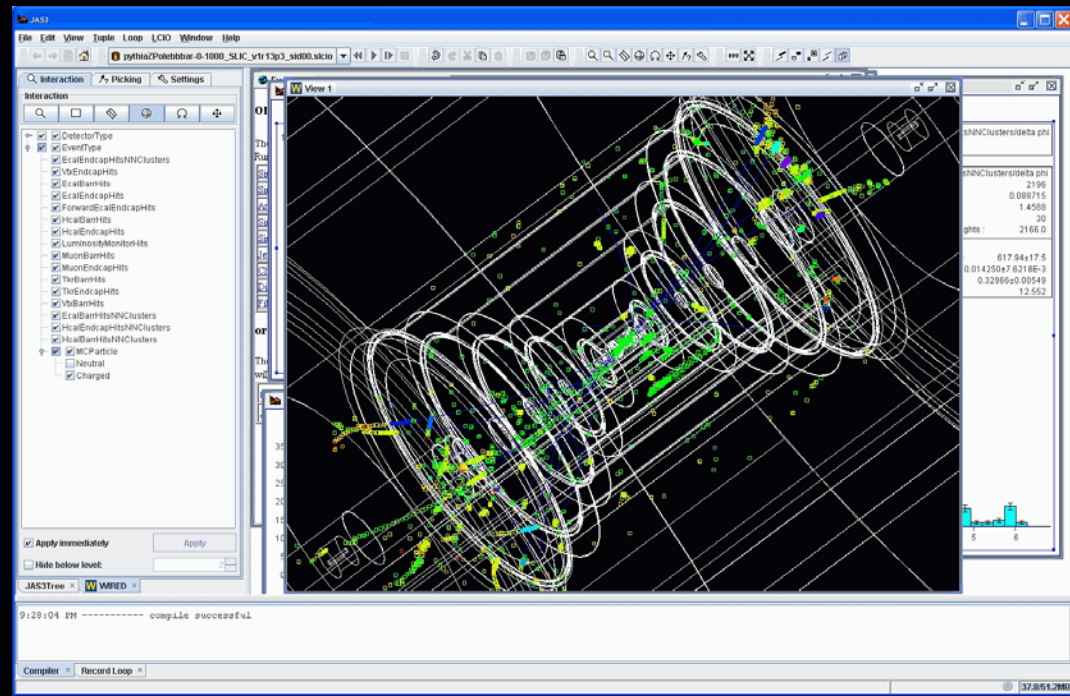
The screenshot shows the JAS interface with the following components:

- Left Panel (JAS Tree):** A hierarchical tree view showing the event structure. The selected event is Event 179, which is a Run0 event.
- Center Panel (LCIO Event Header):** A detailed view of the selected event (Event 179). It shows the event's name, time stamp, detector name, and a list of collections with their names, types, and sizes.
- Bottom Panel (Table):** A table showing the event data. The table has columns for event ID, system, layer, barrel, x, y, raw energy, collected energy, and time.

Event	System	Layer	Barrel	x	y	Raw Energy	Collected Energy	Time (ns)
0	0	2	77	304	1.0653E-4	0.004204	271.25	1065.0
1	0	2	77	304	1.1771E-4	0.002868	271.25	1065.0
2	2	2	77	305	1.0897E-4	0.005871	271.25	1069.2
3	2	2	77	306	1.2495E-4	0.01008	271.25	1072.6
4	4	2	77	306	4.6335E-5	0.030550	271.25	1092.0
5	4	2	77	307	1.2163E-4	0.005883	271.25	1076.2
6	10	2	83	303	3.8268E-4	0.28614	292.25	1062.2
7	26	2	105	200	2.0871E-5	0.014465	309.25	761.25
8	21	2	55	297	1.4120E-5	0.011147	194.25	1041.2
9	6	2	77	309	8.8875E-4	0.07019	271.25	1083.2
10	6	2	78	299	3.4449E-4	0.07332	267.25	1048.2
11	7	2	83	313	5.7790E-4	0.53483	227.25	1097.2

LCIO Event Display

- Fully integrated within JAS using Wired.
- Fully interactive event display
- Detector & Event objects selectable, pickable, queryable, can have cuts applied, etc.
 - Not just a static image.



Raw Data

- LCIO has been used for many years by various testbeam experiments, both tracking and calorimetry.
 - EDM supports raw data taking and analysis.
 - Simple, robust & fast
- Many tools exist for data monitoring, QA, analysis, etc.
- Will need to learn what infrastructure and support exists for EVIO before deciding how best to interface to that file format and event data model.

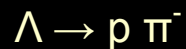
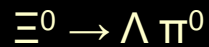
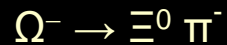
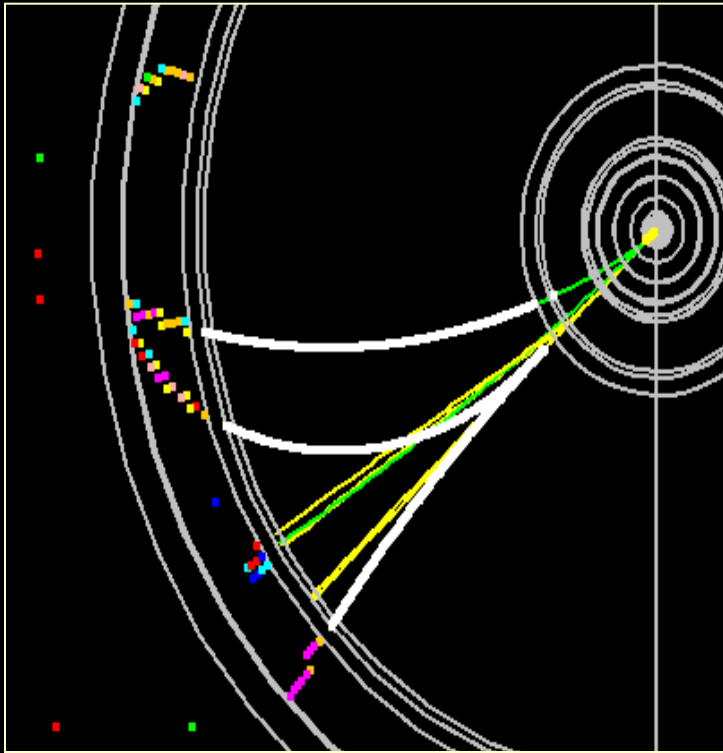
Fast Detector Response Simulation

- Covariantly smear tracks with matrices derived from geometry, materials and point resolution using Billoir's formulation.
 - Analytically from geometry description.
- Smear neutrals according to expected calorimeter resolution (EM for γ , HAD for neutral hadrons)
 - Derived from full Geant4 simulations
- Create reconstructed particles from tracks and clusters (γ , e , μ from MC, $\pi^{+/-}$, K_L^0 for others)
- Can also dial in arbitrary effective jet energy resolution.
 - Derived from full simulation, reconstruction & analysis.
- “Supersymmetry, the *ILC*, and the LHC *inverse problem*”

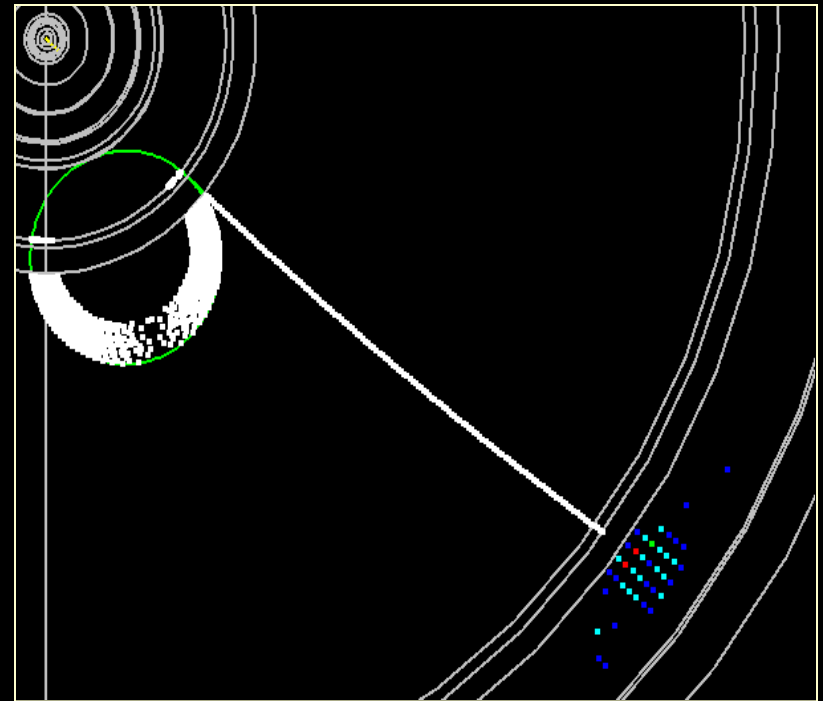
lelaps

- Fast detector response package.
- Handles decays in flight, multiple scattering and energy loss in trackers.
- Parameterizes particle showers in calorimeters using Grindhammer & Peters, and Bock algorithms.
- Produces detector data at the hit level.
 - Feeds directly into full reconstruction.
- Uses runtime geometry (compact.xml → godl).

Lelaps: Decays, dE/dx , MCS



$\pi^0 \rightarrow \gamma \gamma$ as
simulated by Lelaps for the
LDC model.



gamma conversion as
simulated by Lelaps for the
LDC model.

Note energy loss of electron.

Detector Design (GEANT 4)

- Need to be able to flexibly, but believably simulate the detector response for various designs.
- The daunting machine backgrounds expected at the HPS will require detailed full detector simulations.
- GEANT is the de facto standard for HEP physics simulations.
- Use runtime configurable detector geometries
- Write out “generic” hits to digitize later.

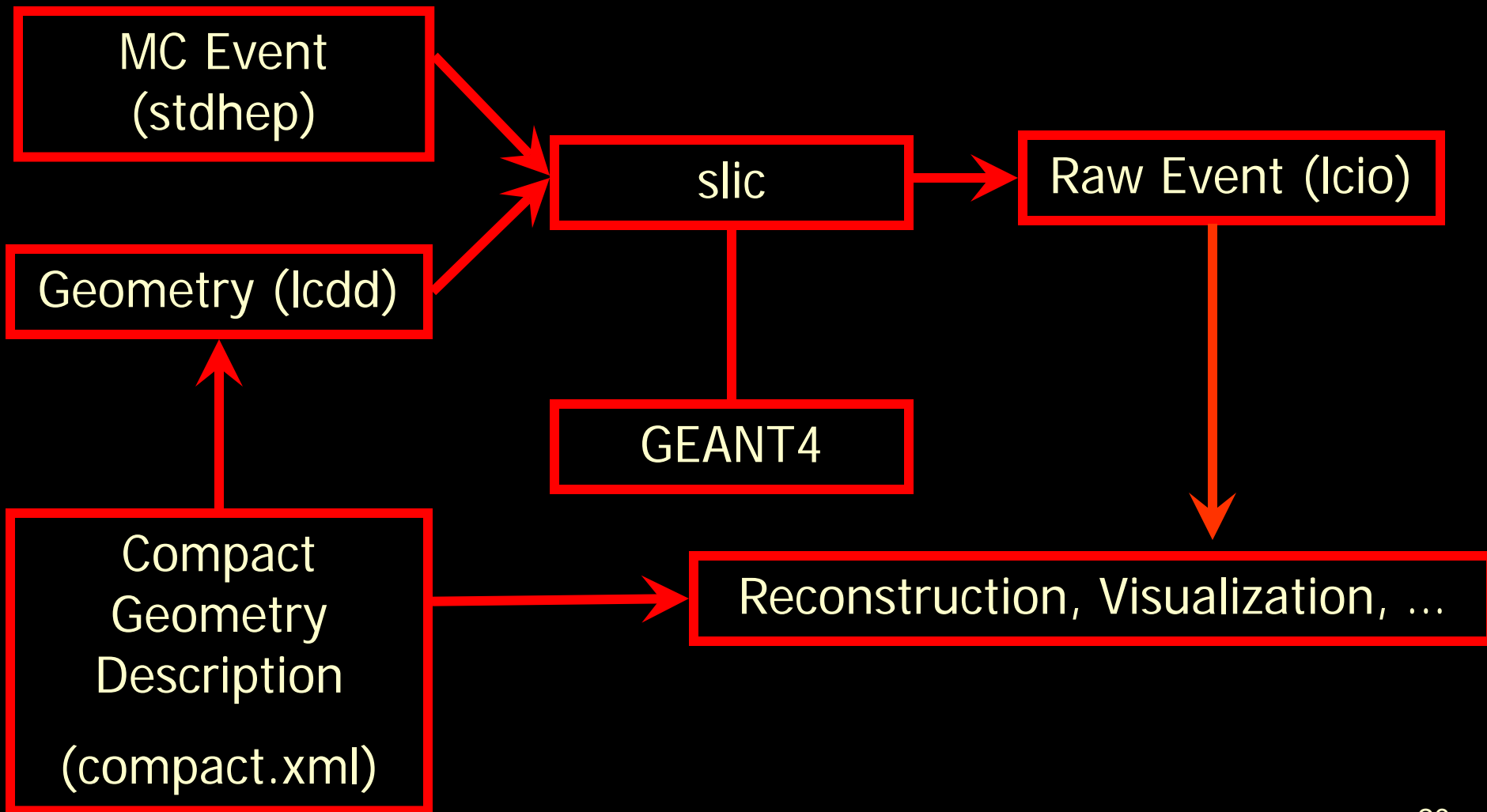
Full Detector Response Simulation

- Use Geant4 toolkit to describe interaction of particles with matter and fields.
- Thin layer of LC-specific C++ provides access to:
 - Event Generator input (binary stdhep format)
 - Detector Geometry description (XML)
 - Detector Hits (LCIO)
- Geometries fully described at run-time!
 - In principle, as fully detailed as desired.

Geometry Definition

- Goal was to free the end user from having to write any C++ code or be expert in Geant4 to define the detector.
- All of the detector properties should be definable at runtime with an easy-to-use format.
- Selected xml, and extended the existing GDML format for pure geometry description.

LC Detector Full Simulation

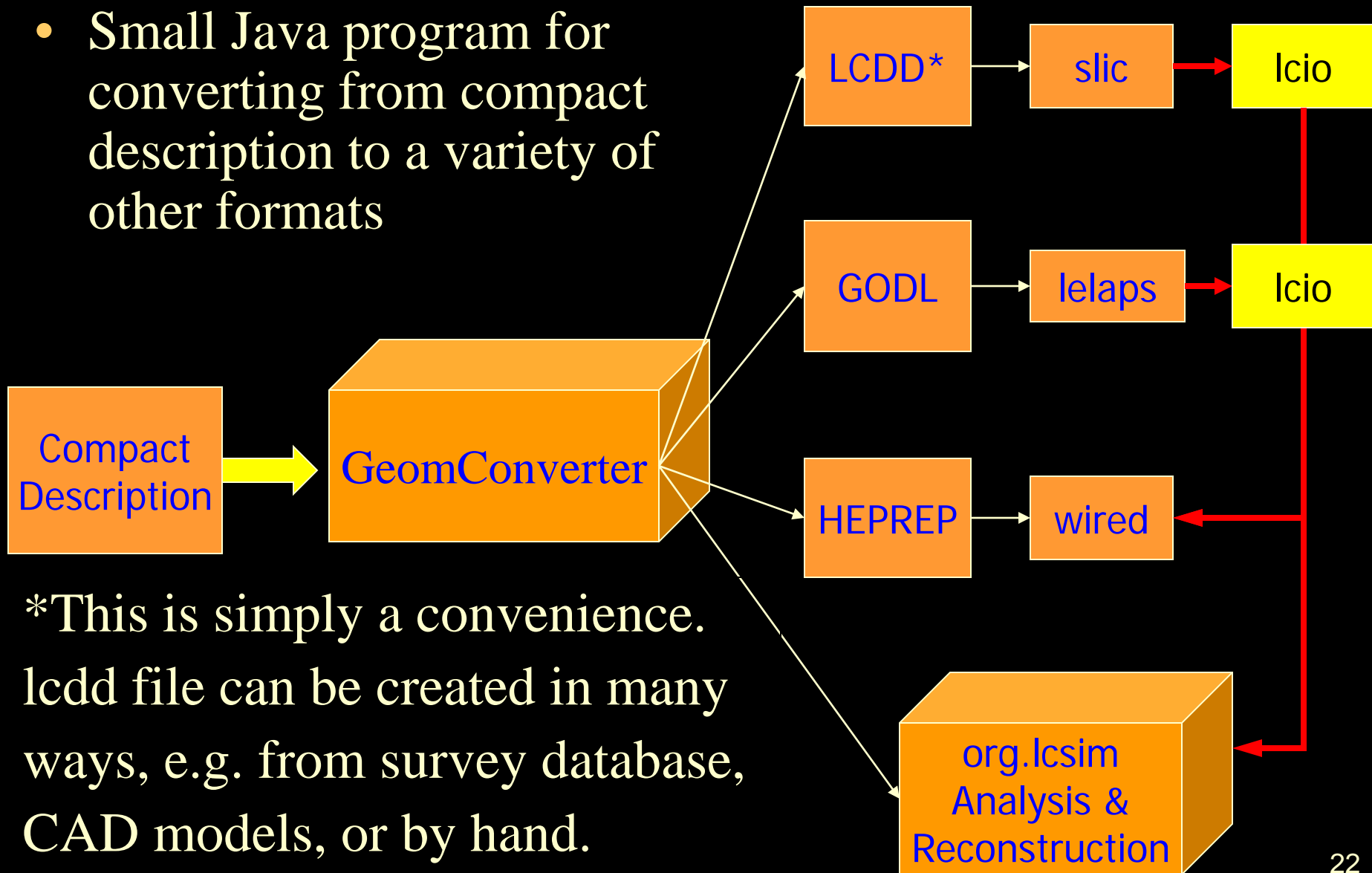


slic: The Executable

- Build static executables on Linux, Windows, Mac.
- Commandline or G4 macro control.
- Only dependence is local detector description file.
 - Trivial Grid usage (no database call-backs, etc.)
 - Grid ready, Condor and lsf scripts available.
- Event input via stdhep, particle gun, ...
- Detector input via GDML, lcdd
- Response output via LCIO using generic hits.
- Fast (~30s/event for TeV ILC)

GeomConverter

- Small Java program for converting from compact description to a variety of other formats

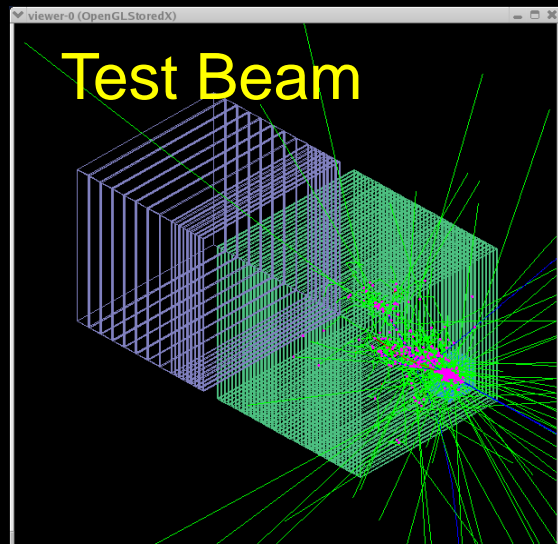
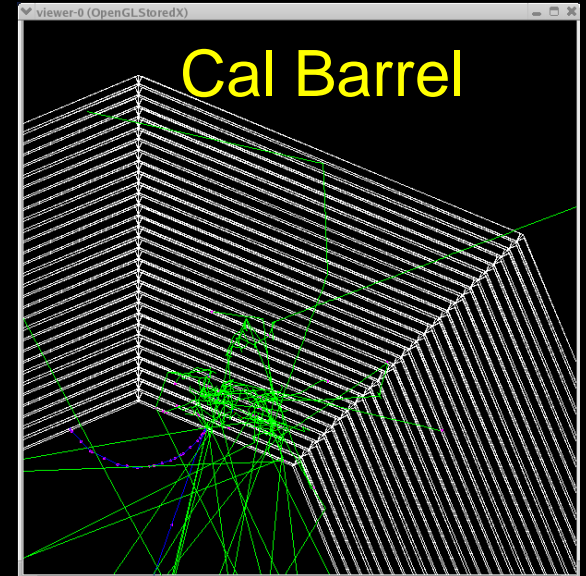
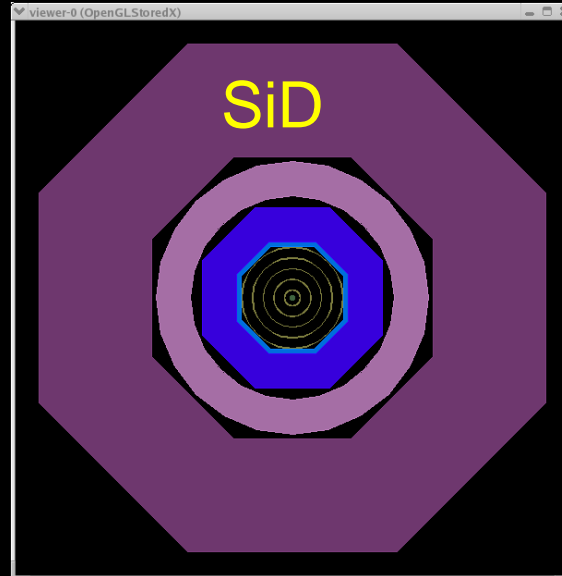
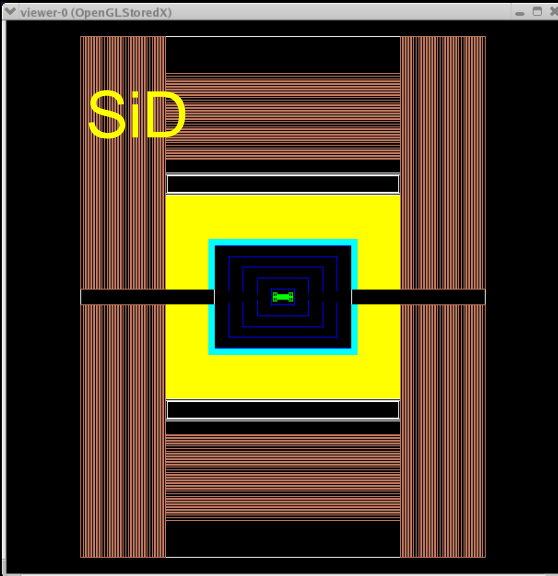


*This is simply a convenience. lcid file can be created in many ways, e.g. from survey database, CAD models, or by hand.

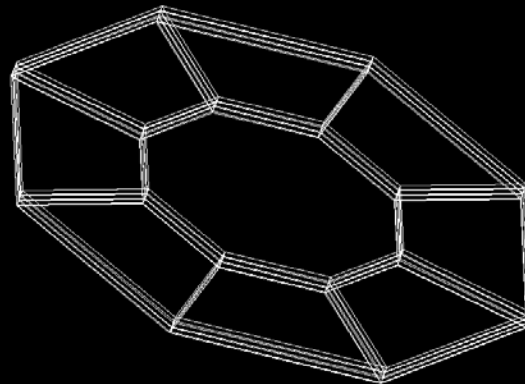
Detector Variants

- Runtime XML format allows variations in detector geometries to be easily set up and studied:
 - Absorber materials and readout technologies for sampling calorimeters
 - e.g. Steel, W, Cu, Pb + RPC vs. GEM vs. Scintillator readout
 - Optical processes for dual-readout or crystal calorimeters
 - Layering (radii, number, composition)
 - Readout segmentation (size, projective vs. nonprojective)
 - Tracking detector technologies & topologies
 - TPC, Silicon microstrip, pixels, ...
 - “Wedding Cake” Nested Tracker vs. Barrel + Cap
 - Far forward MDI variants, shielding, field strength, etc.

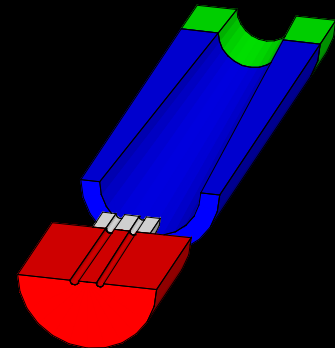
Example Geometries



Cal Endcap

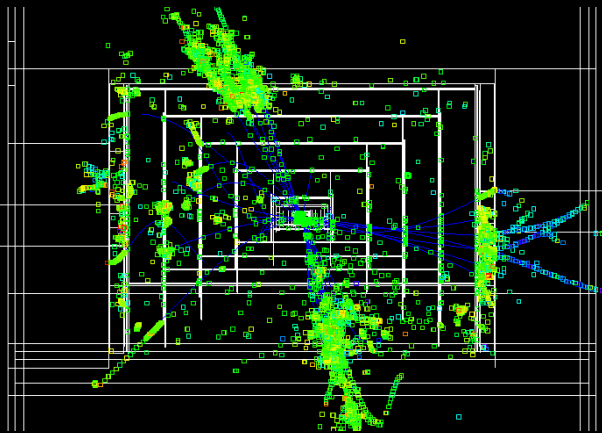
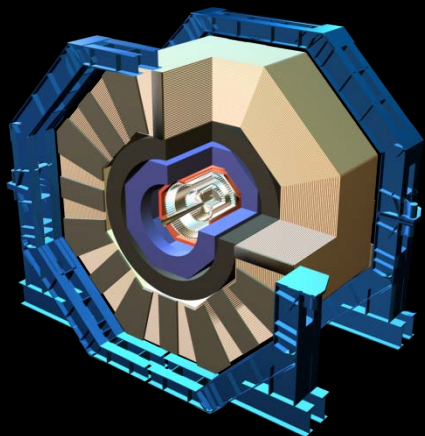


MDI-BDS

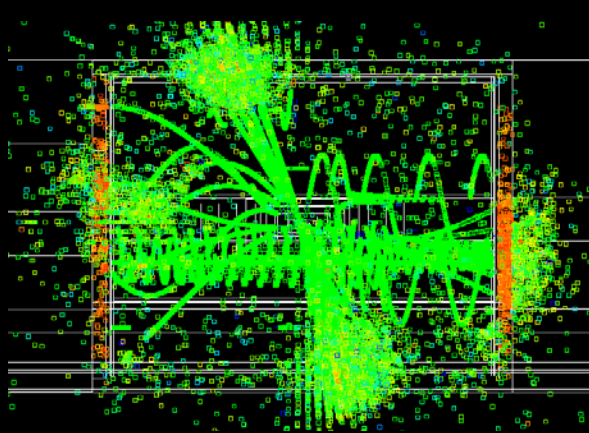
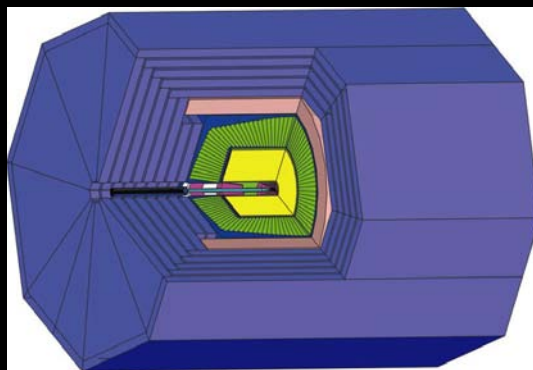


ILC Full Detector Concepts

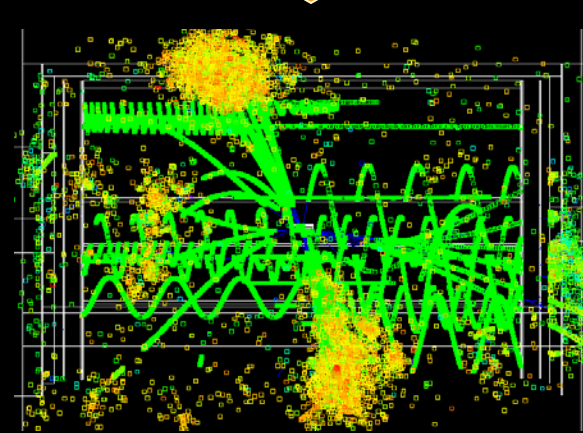
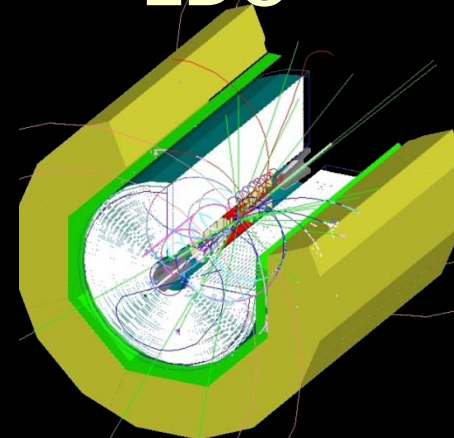
SiD



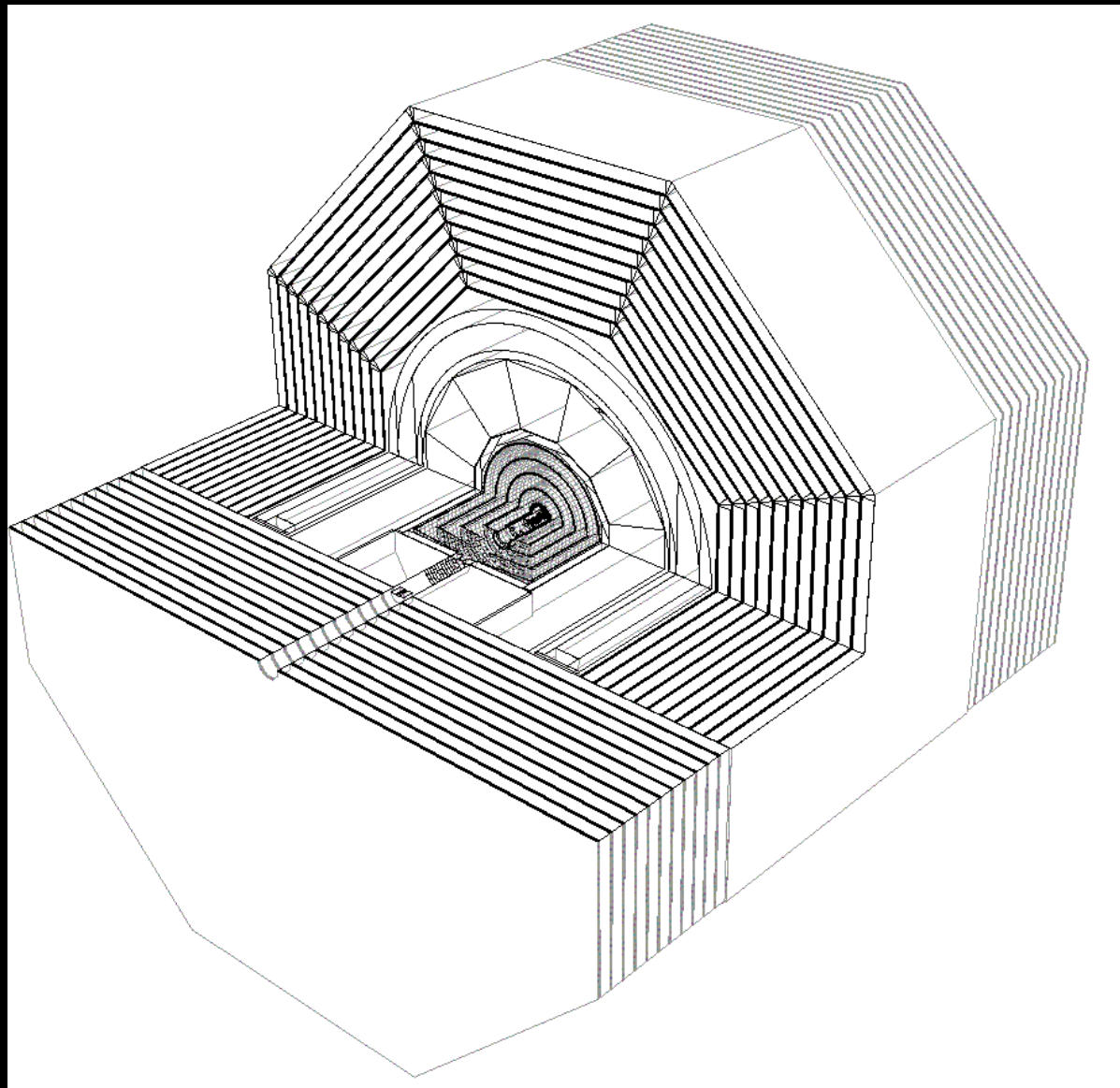
GLD



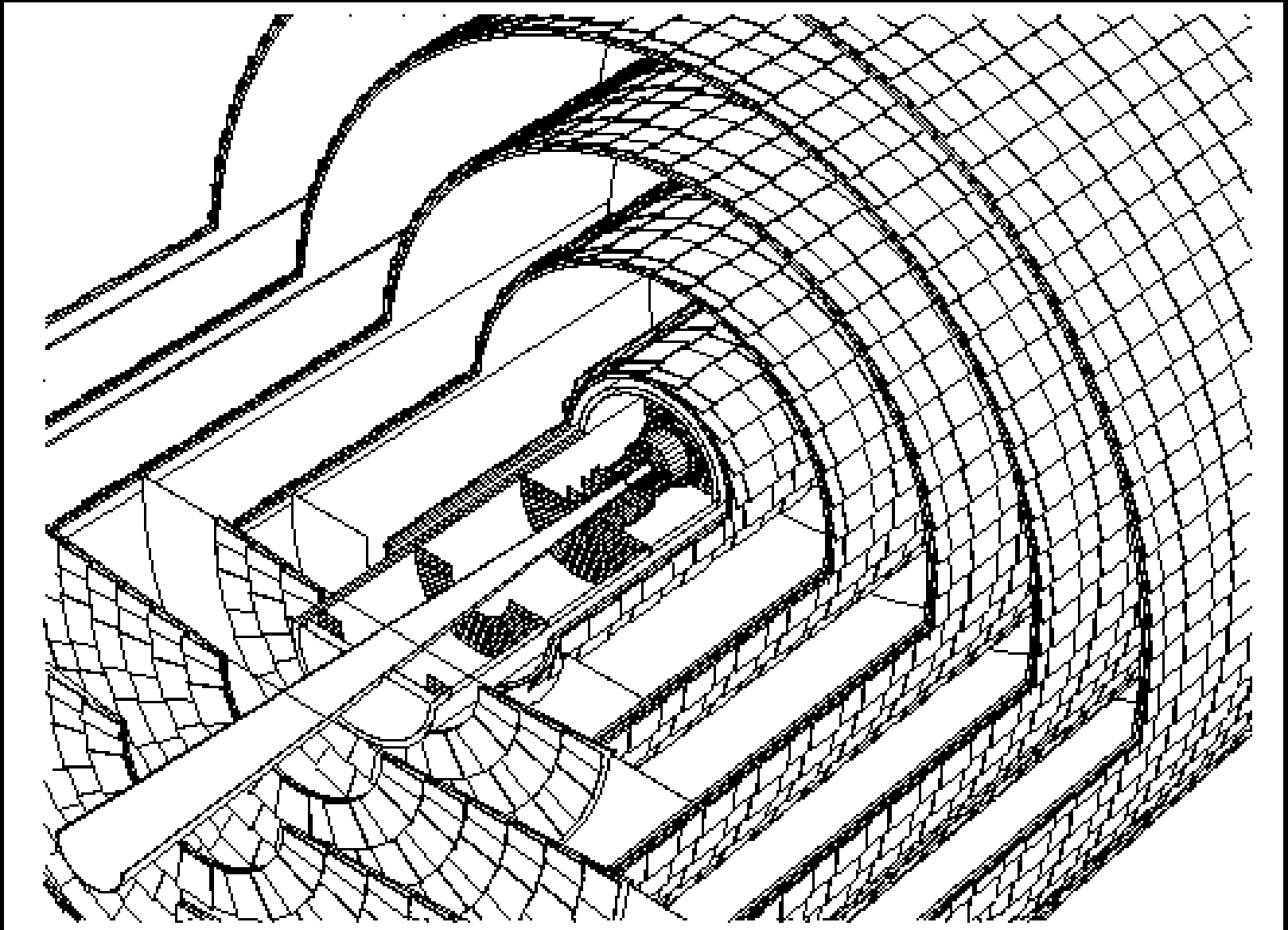
LDC



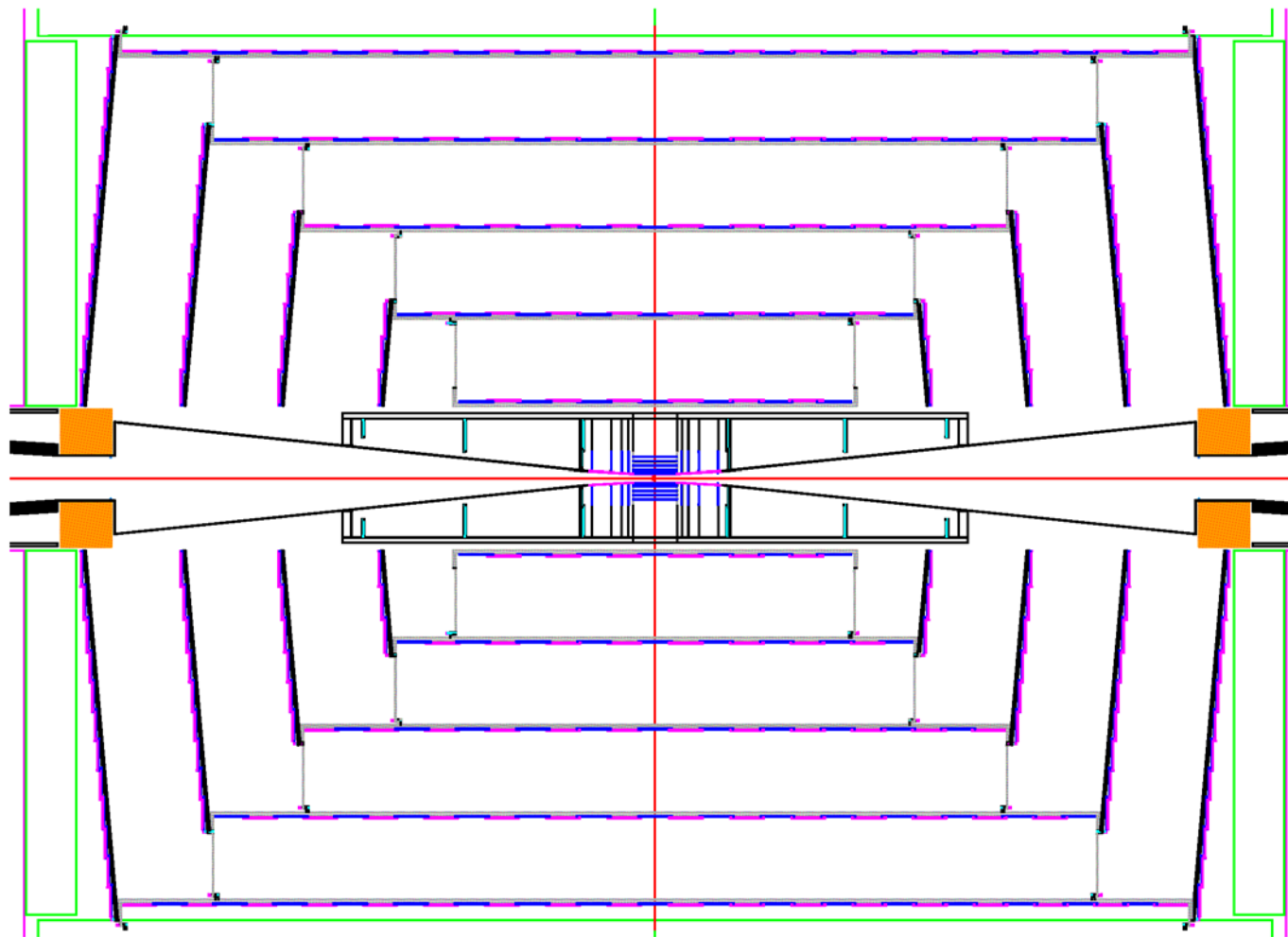
Silicon Detector Tracker



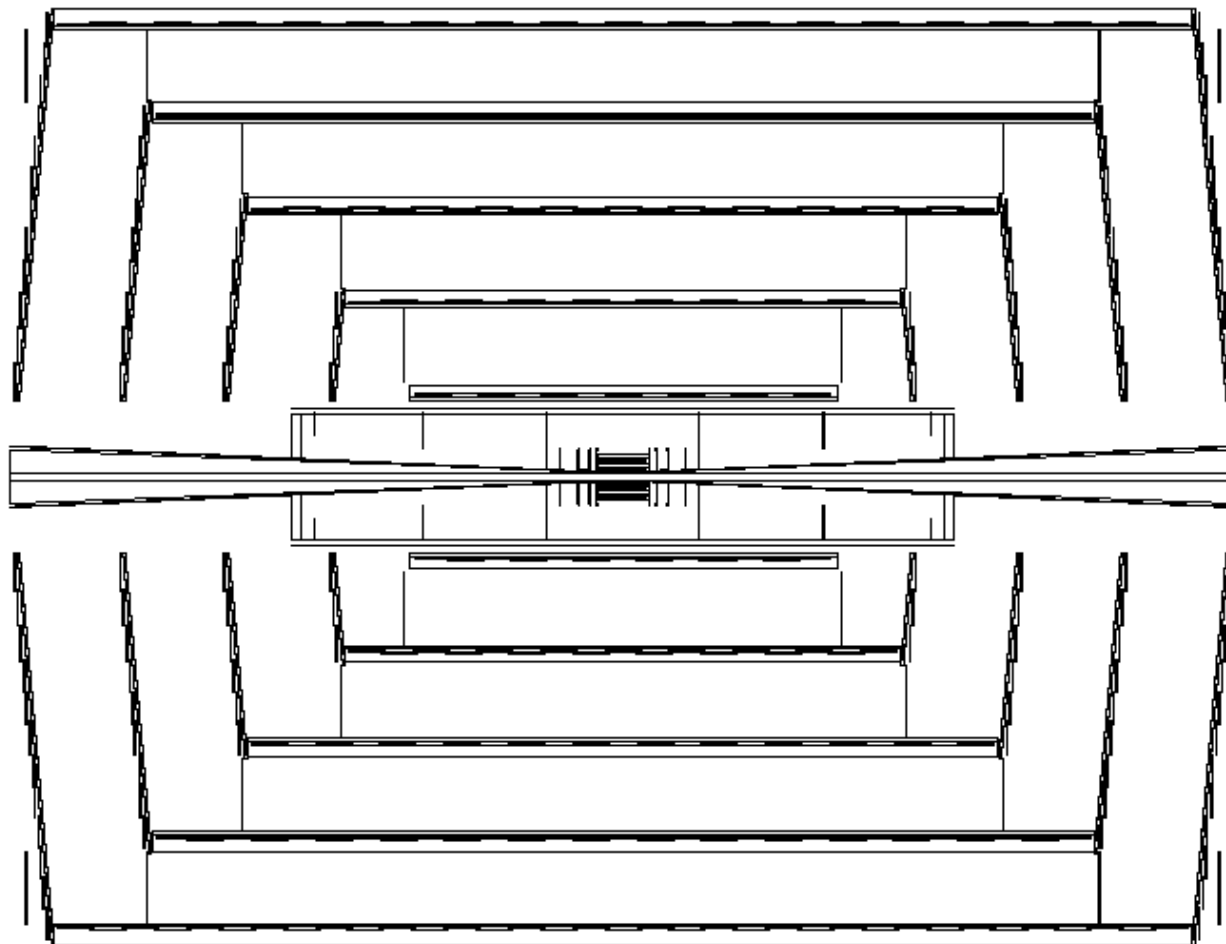
Silicon Detector Tracker



Silicon Detector Tracker



Silicon Detector Tracker



Simulating the HPS Experiment

- Compared to Silicon Detector, HPS tracker is straightforward to implement.
 - Complete control over definitions of tracker sensitive wafers and support structures.
 - Very detailed models for charge deposition, drift and diffusion.
 - Detailed model for the electronics response.
 - MC Hits → Channel ID & Pulse Height → Clusters → Hits ($x \pm \delta x$)
 - Matt and Rich have implemented the designs, simulated the response, reconstructed tracks and vertices and presented analysis results many times.
 - will not repeat here.

Simulating the HPS Experiment II

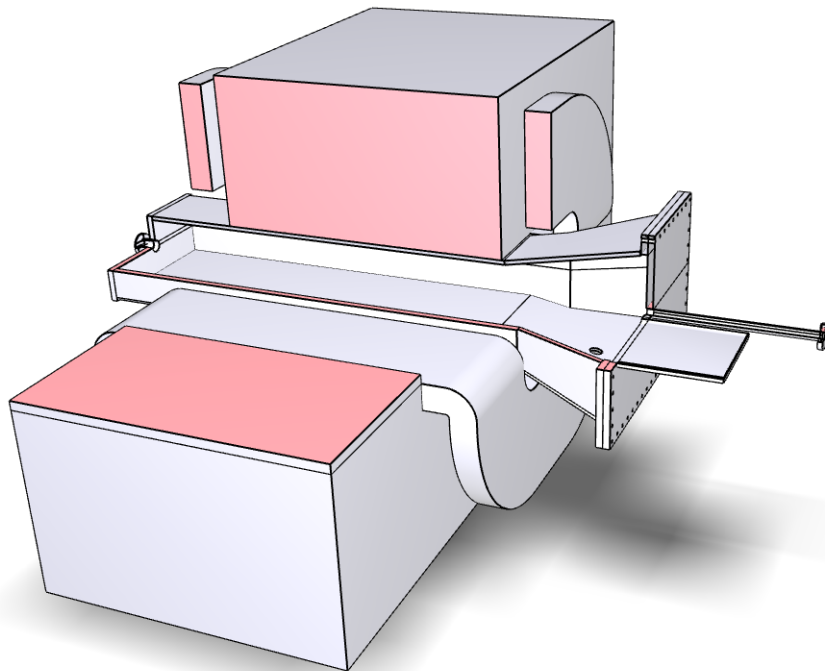
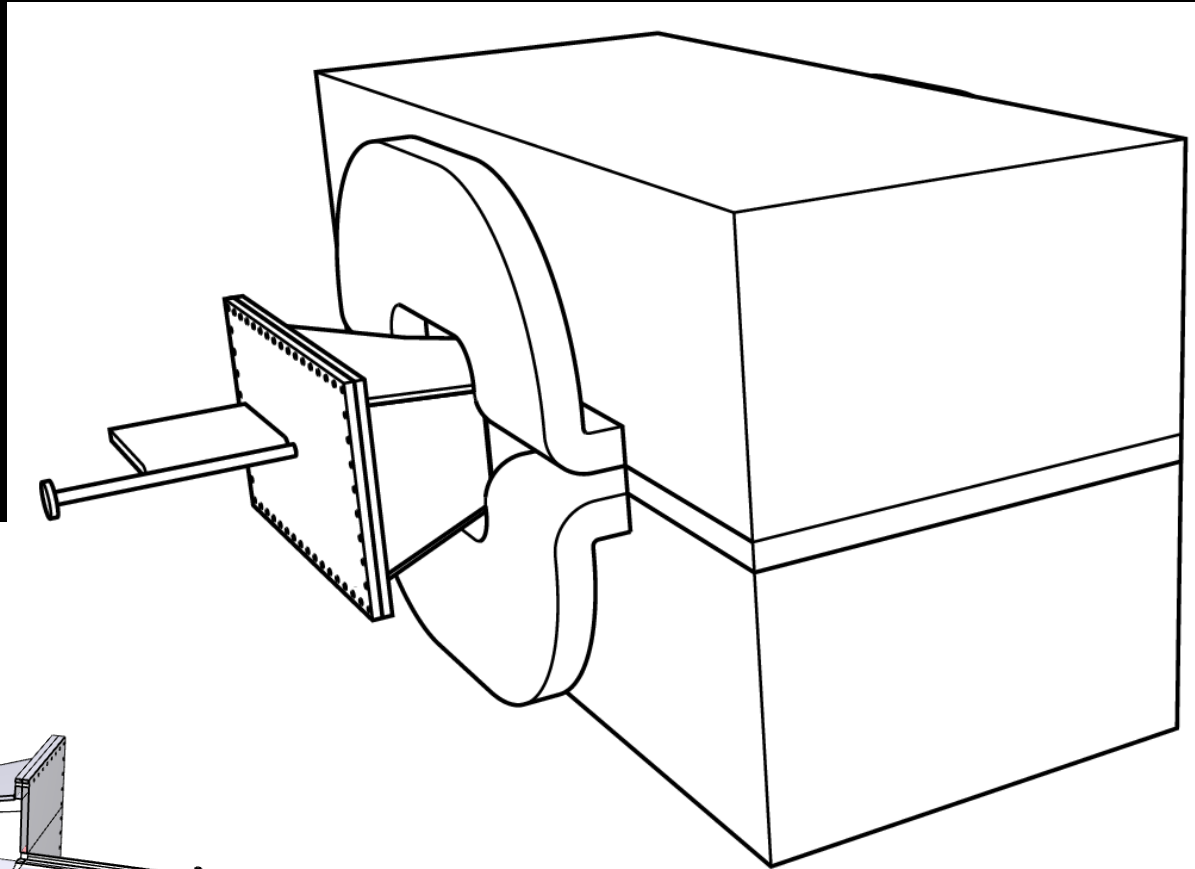
- Other major detector elements are the electromagnetic calorimeter and the muon system.
 - Will need same geometry in reconstruction.
 - Additions to compact.xml & GeomConverter needed to provide same geometry to sim and reco.
 - Straightforward, but still requires expert development.
 - Needs time from Jeremy McCormick
- Need detailed simulation of the beamline and support elements.
 - Complex geometries, but not needed in reconstruction.
 - Can go straight from CAD to slic via GDML.

HPS Dipole and Vacuum Vessel

- CAD Model from Marco Oriunno
- Conversion to GDML by N. Graf
 - Resulting geometry is tessellated solid
 - performance not expected to be as good as using Geant4 primitives.
 - but most particles should never interact with these elements
 - included into compact.xml as gdml snippet

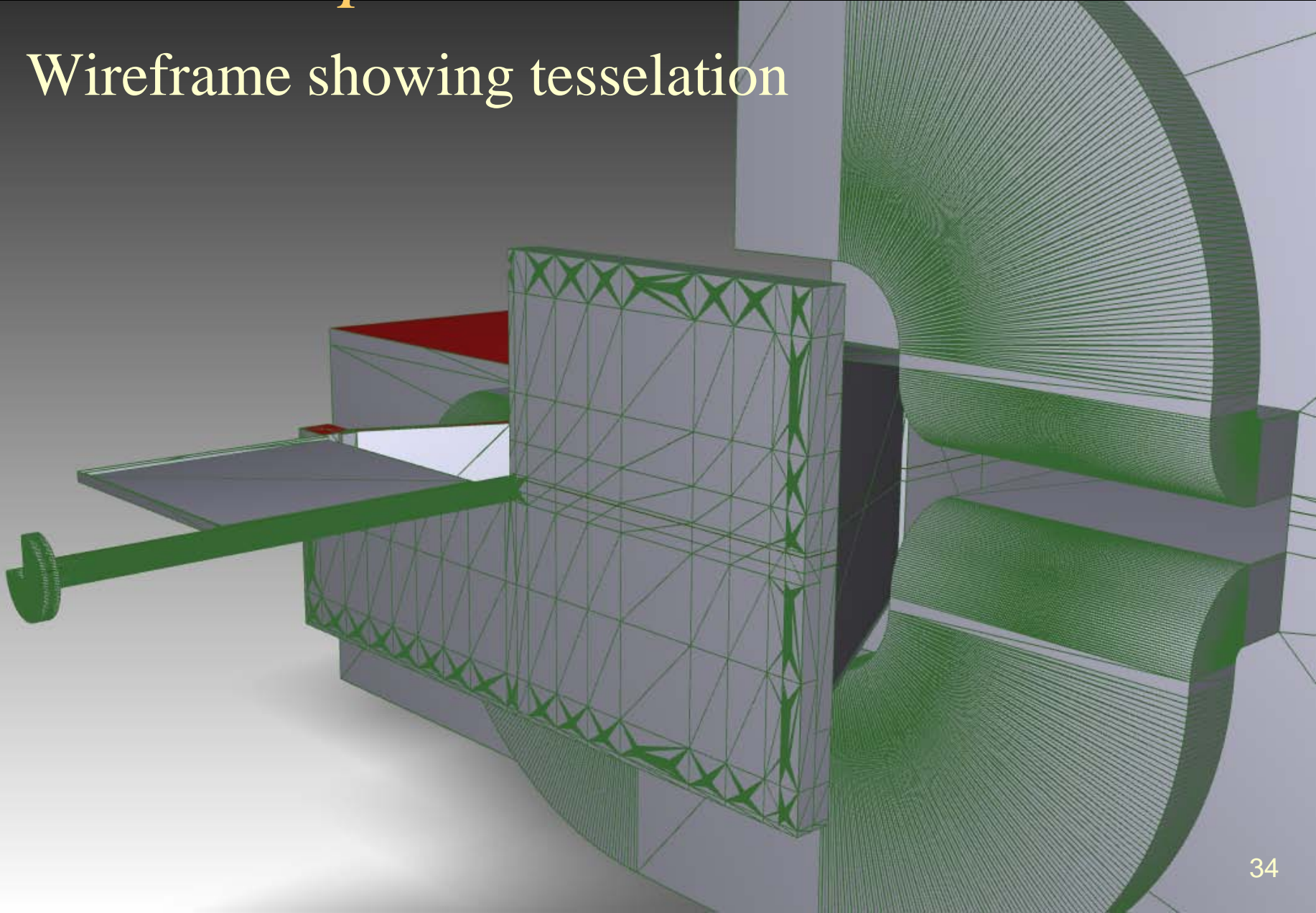
HPS Dipole and Vacuum Vessel

- Images from geometry output from slic.



HPS Dipole and Vacuum Vessel

- Wireframe showing tessellation

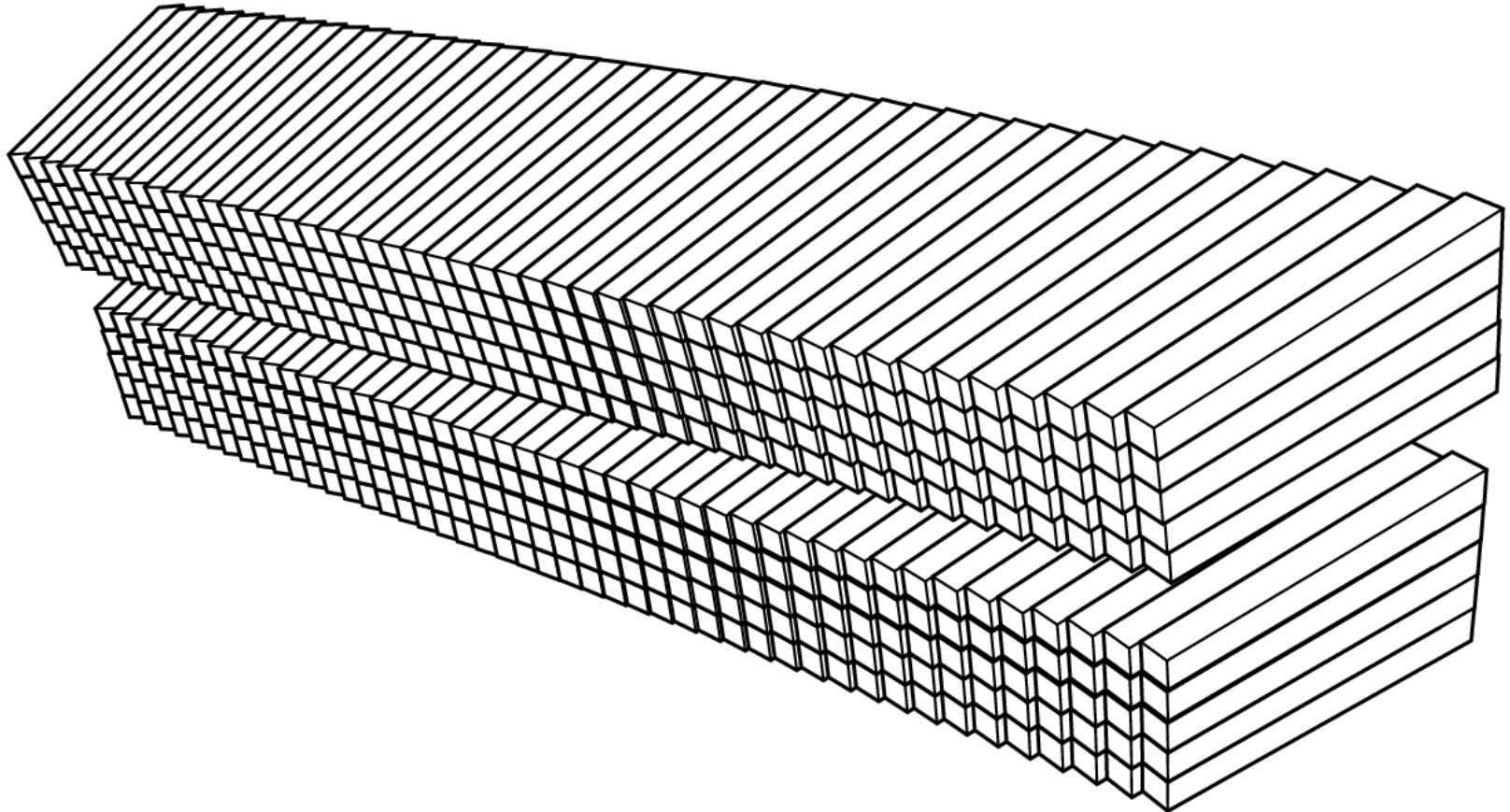


HPS ECal

- Will want to have the geometry definable from the compact.xml in order to have it accessible from the reconstruction.
- However, can also implement simply using the CAD model to get started and as proof-of-principle.

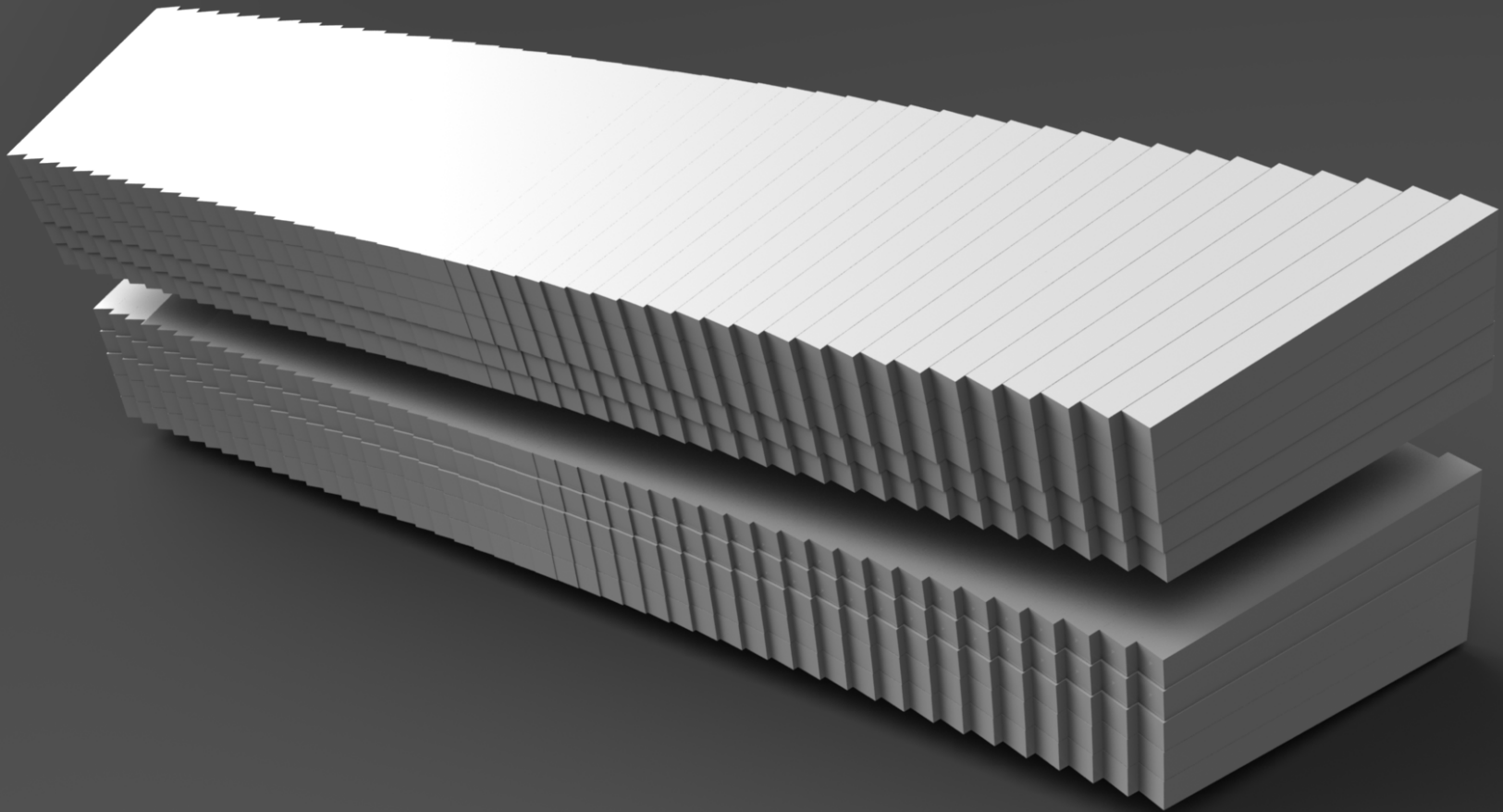
HPS ECal

- CAD Model from Marco Oriunno.
- Translated to GDML by N. Graf.
 - not yet a sensitive detector (but could be made so)



HPS ECal

- CAD Model from Marco Oriunno.
- Translated to GDML by N. Graf.

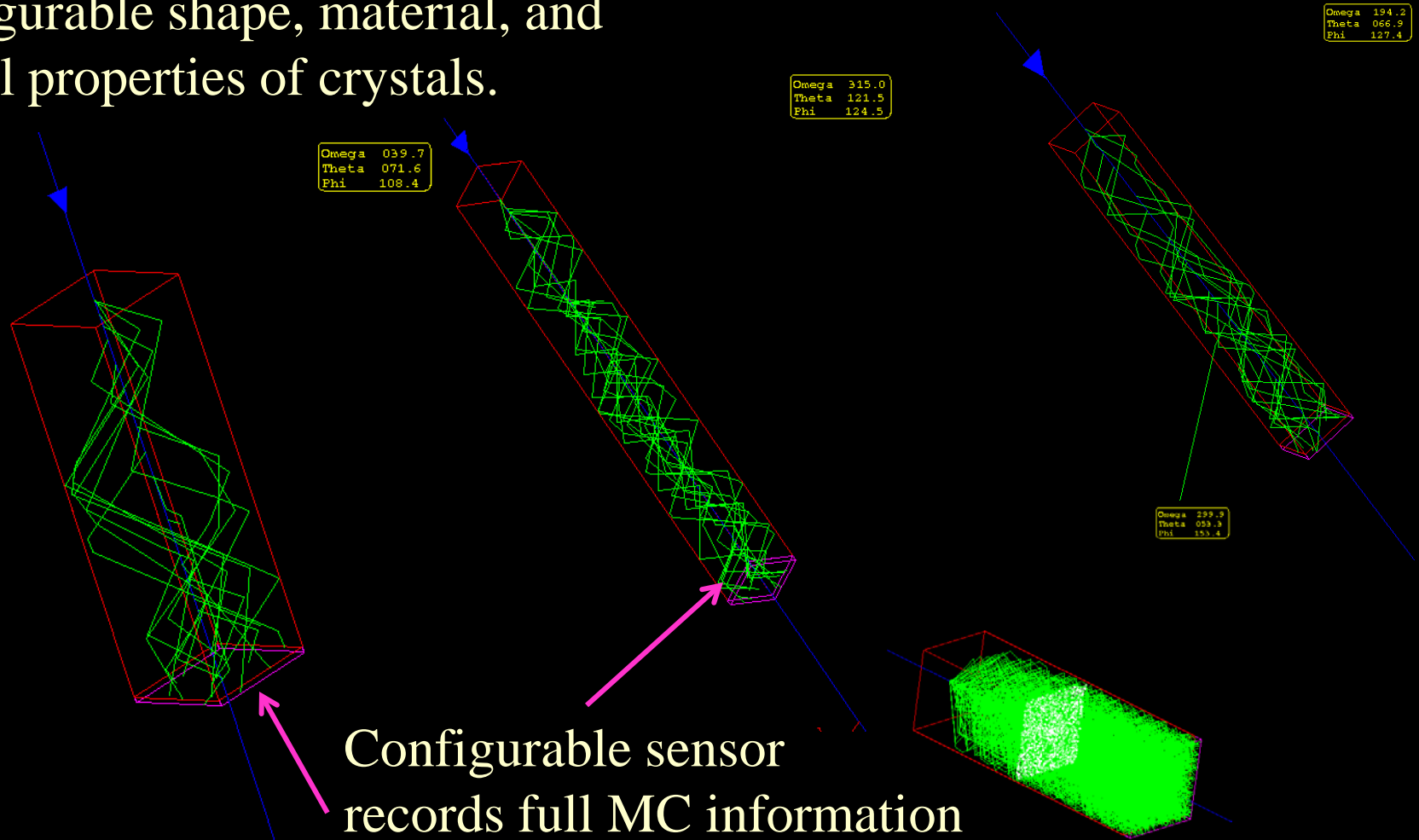


HPS ECal Response

- Default Sensitive Detector response for calorimeters is to simply record energy deposition, time and MC particle information.
- Hans Wenzel also implemented scintillation and Cherenkov light deposition within slic/lcsim for studies of total absorption, dual-readout crystal calorimetry.
 - simple accumulation of energy deposit in crystal
- Currently implementing full optical photon ray tracing within crystal and propagation to sensitive detector.

Optical Ray Tracing in Crystals

Configurable shape, material, and optical properties of crystals.



Configurable sensor records full MC information about each incident photon

HPS ECal & Muon System Todo

- Implement geometry description of crystal array, plus wrapping, support, etc. into compact.xml and GeomConverter.
 - Need some input on level of detail of simulation.
 - Is simple Σ dE good enough?
 - Is simple Σ (scint + Cherenkov) good enough?
 - Do we need full optical definition and ray tracing?
 - How detailed do we need to model the readout sensor?
- Ignorant of latest muon system geometry or readout.
 - Is current calorimeter support sufficient?

Beamline, Magnets and Supports Todo

- Is CAD to Geant solution usable?
 - Memory and time requirements on slic.
- If not, what level of simplification can be achieved automatically? What level requires manual intervention?
- Is a magnetic field map available for the experimental region?
 - What level of accuracy is required?

Reconstruction/Analysis Overview

- Java based reconstruction and analysis package
 - Runs standalone or inside Java Analysis Studio (JAS)
 - Fast MC → Smearred tracks and calorimetry clusters
 - Full Event Reconstruction
 - Beam background overlays at detector hit level, including time offsets.
 - detector readout digitization (CCD pixels, Si μ -strips, TPC pad hits)
 - *ab initio* track finding and fitting for ~arbitrary geometries
 - multiple calorimeter clustering algorithms
 - Individual Particle reconstruction (cluster-track association)
 - Analysis Tools (including WIRED event display)
 - Physics Tools (Jet Finding, Vertex Finding, Flavor Tagging)
- Write once run, run anywhere
 - Exact same libraries run on all platforms (Windows, Mac, Linux(es), Grid) using the Java Virtual Machine.

Tracking

- Analytic covariance matrices available for fast MC smearing for each detector.
- Track “cheater” available for studies of full detector simulation events. Assigns hits on basis of MC parentage.
- Ab initio track finding packages.
- Is fitting code incorporating multiple scattering and energy loss via weight matrix or Kalman Filter desirable?

Tracking Detector Readout

- Hits in Trackers record full MC information.
- Module tiling and signal digitization is deferred to analysis stage.
 - Used to rapidly study many possible solutions.
- Fully-featured package to convert MC hits in silicon to pixel hits. Fully configurable at runtime.
MC Hits → Pixel ID & ADC → Clusters → Hits ($x \pm \delta x$)
- Can correctly study occupancies, overlaps, ghost hits, etc.

Calorimeter Reconstruction

- MC \rightarrow ADC, time + channel ID \rightarrow energy in cell
 - Depends on how complicated the response simulation needs to be.
 - Can get started with existing simulation change and improve as needed.
- CalorimeterHits \rightarrow Cluster (energy + position)
 - Existing clustering algorithms most likely usable as-is.
 - New algorithms can be implemented as needed
- Track-shower association likely trivial compared to ILC or CLIC PFA.

Java Analysis Studio (JAS)

- Integrated Development Environment (editor, compiler)
- Cross-platform physics analysis environment with iterative, event-based analysis model
 - quick development, debugging, ad hoc analysis
 - additional functionality with plugins
- Dynamically load / unload Java analysis drivers
 - Supports distributed computing.
- Plotting and fitting and analysis (cuts, scripting) engine
 - 1D, 2D histograms, clouds, profiles, dynamic scaling, cuts
 - high-quality output to vector or raster formats
- Integrated event browser and event display

JAS editor/compiler

The screenshot displays the JAS3 editor/compiler interface. The main window is titled "JAS3" and contains a menu bar (File, Edit, View, Tuple, Loop, LCIO, Window, Help) and a toolbar. The left sidebar shows a file tree with "DataSets" and "Programs" folders, and a sub-folder "CombinedConeClusterAnalysisDriver". The central editor window displays the source code for "CombinedConeClusterAnalysisDriver.java". The code includes a class definition and a process method that iterates over hit collections and prints their sizes. The right sidebar shows a visualization window with a black background, overlaid with a white grid and several green and yellow points, representing a detector layout or hit distribution. The status bar at the bottom indicates "9:23:33 PM ----- compile successful" and "Analyzed 1 records in 190ms".

```
70 CombinedConeClusterAnalysisDriver.java
71
72 *
73 * Created on March 3, 2006, 4:21 PM
74 *
75 * To change this template, choose Tools | Template Manager
76 * and open the template in the editor.
77 *
78 */
79
80 /**
81 *
82 * @author ngraf
83 */
84
85 public class CombinedConeClusterAnalysisDriver extends Driver
86 {
87     private FixedConeClusterer _clusterer;
88     /** Creates a new instance of CombinedConeClusterAnalysisDriver */
89     public CombinedConeClusterAnalysisDriver()
90     {
91         double radius = 1.2;
92         double seedEnergy = 0.0;
93         double minEnergy = 0.0;
94         _clusterer = new FixedConeClusterer(radius, seedEnergy, minEnergy);
95     }
96
97     protected void process(EventHeader event)
98     {
99         // the list of hit cells to cluster
100         List<CalorimeterHit> cellsToCluster = new ArrayList<CalorimeterHit>();
101         // get all the calorimeter hits in this event...
102         List<List<CalorimeterHit>> collections = event.get(CalorimeterHit.class);
103
104         for (List<CalorimeterHit> collection : collections)
105         {
106             LCMetaData meta = event.getMetaData(collection);
107             System.out.println(meta.getName()+" has "+collection.size()+" hits");
108             for (CalorimeterHit hit : collection)
109             {
110                 // should apply cut here...
111                 // punt for now and add ALL hits
112                 cellsToCluster.add(hit);
113             }
114         }
115
116         System.out.println("Event has "+cellsToCluster.size()+" hit cells");
117     }
118 }
```

JAS3Tree x WIRED x

9:23:33 PM ----- compile successful

Compiler x Record Loop x

Analyzed 1 records in 190ms

24.6/34.4MB

JAS event browser

JAS3

File Edit View Tuple Loop LCIO Window Help

pythia2Polebbar-0-1000_SLIC_v1r13p3_sid00.slcio

DataSets
Programs
ClusterFinding
aida22594aida

e-(E=45.500 status=Documentation)
 e-(E=45.500 status=Documentation)
 e-(E=45.500 status=Documentation)
 Zo(E=91.000 status=Documentation)
 b(E=45.500 status=Documentation)
 b(E=38.330 status=Intermediate)
 unknown(E=91.000 status=Intermediate)
 B*(E=31.648 status=Intermediate)
 pi0(E=2.1998 status=Intermediate)
 pi+(E=4.0914 status=Final State)
 rho0(E=1.2687 status=Intermediate)
 K(892)*o(E=2.8169 status=Intermediate)
 K-(E=2.6171 status=Final State)
 pi0(E=1.5627 status=Intermediate)
 pi0(E=60782 status=Intermediate)
 Delta+(E=3.5210 status=Intermediate)
 Ko_bar(E=2.0863 status=Intermediate)
 Xi0_bar(E=7.6546 status=Intermediate)
 B(s)*o(E=30.926 status=Intermediate)
 gluon(E=7.0819 status=Intermediate)
 gluon(E=91563 status=Intermediate)
 gluon(E=49304 status=Intermediate)

LCSim Event
Run:0 Event: 179

LCIO Event Header

Run	0
Event	179
Time Stamp	Thu Feb 16 10:34:33 PST 2006
Detector Name	sid00

Collections

Name	Type	Size
EcalEndcapHitsNNClusters	org.lcsim.event.Cluster	22
EcalEndcapHitsNNClusters	org.lcsim.event.Cluster	22
EcalBarrHitsNNClusters	org.lcsim.event.Cluster	24
HcalBarrHitsNNClusters	org.lcsim.event.Cluster	7
HcalEndcapHitsNNClusters	org.lcsim.event.Cluster	3
VbEndcapHits	org.lcsim.event.SimTrackerHit	72
EcalBarrHits	org.lcsim.event.SimCalorime...	1036
EcalEndcapHits	org.lcsim.event.SimCalorime...	1084
ForwardEcalEndcapHits	org.lcsim.event.SimCalorime...	33
HcalBarrHits	org.lcsim.event.SimCalorime...	197
HcalEndcapHits	org.lcsim.event.SimCalorime...	129
LuminosityMonitorHits	org.lcsim.event.SimCalorime...	0
MuonBarrHits	org.lcsim.event.SimCalorime...	1

LCSim Event
Run:0 Event: 179

Collection: EcalEndcapHits size:1084 flags:e0000000

id: system	id: layer	id: barrel	id: x	id: y	raw energy (...)	corrected e...	x (mm)	y (mm)	z (mm)	time (ns)
6	0	2	77	304	1.0683E-4	.0084284	271.25	1065.8	-1683.3	6.7236
6	1	2	77	304	1.1771E-4	.0092868	271.25	1065.8	-1687.1	6.7386
6	2	2	77	305	1.0897E-4	.0085971	271.25	1069.2	-1690.9	6.7536
6	3	2	77	306	1.2685E-4	.010008	271.25	1072.8	-1694.6	6.7685
6	4	2	77	306	4.6335E-5	.0036556	271.25	1072.8	-1698.3	6.7834
6	4	2	77	307	1.2153E-4	.0095883	271.25	1076.2	-1698.3	6.7840
6	10	2	83	303	3.6268E-4	.028614	292.25	1062.2	-1720.8	7.2580
6	26	2	105	200	2.0871E-5	.0016466	369.25	701.75	-1798.3	101.67
6	21	2	55	297	1.4128E-5	.0011147	194.25	1041.2	-1767.1	12.514
6	9	2	77	309	8.8875E-4	.070119	271.25	1083.2	-1717.1	6.8521
6	6	2	73	299	3.4643E-4	.027332	257.25	1048.2	-1705.8	7.4273
6	7	2	93	313	6.7790E-4	.053483	327.25	1097.2	-1709.6	8.2976

JAS3Tree x WIRED x

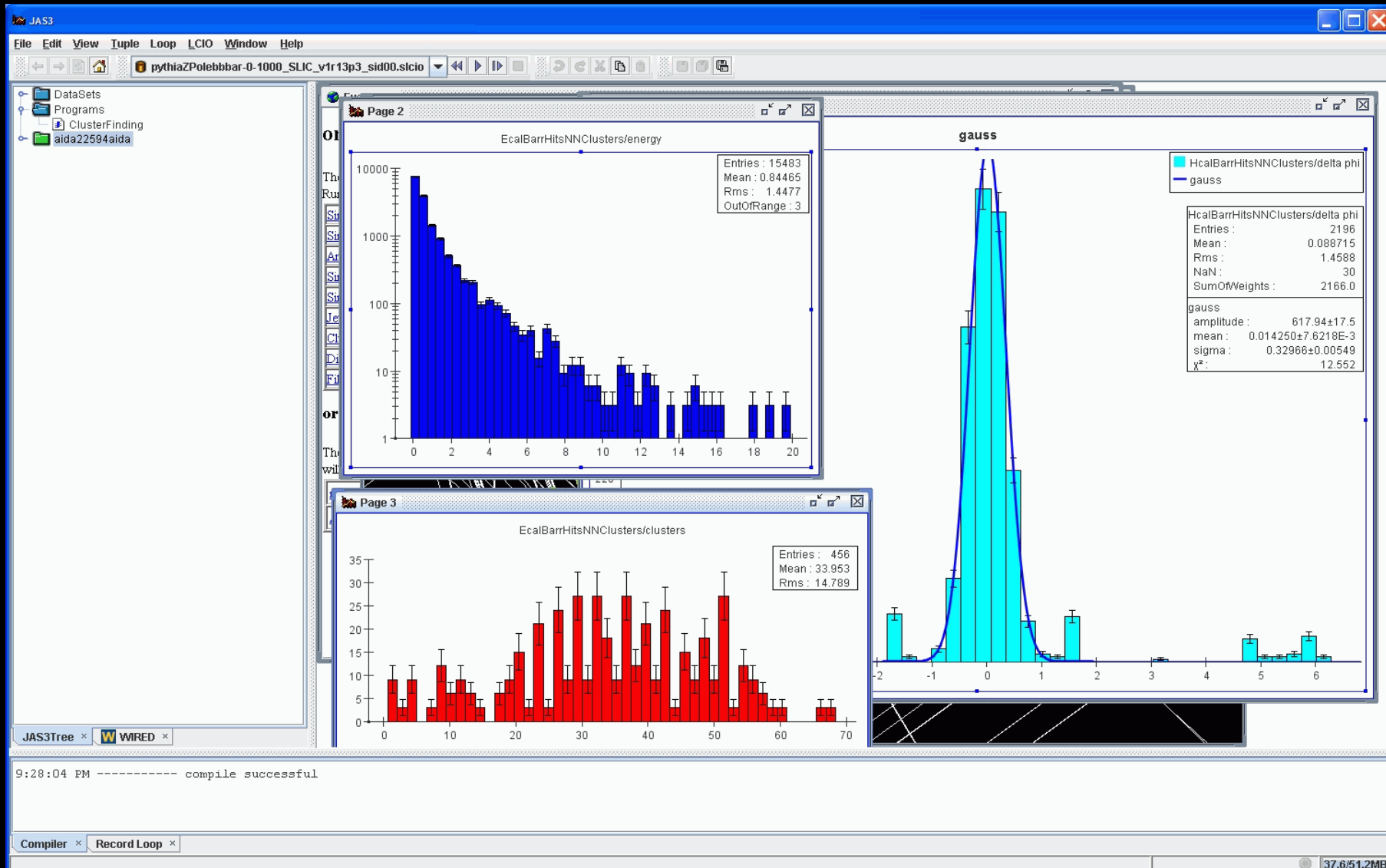
9:28:04 PM ----- compile successful

Compiler x Record Loop x

Analyzed 151 records in 114485ms

38.4/51.2MB

JAS histogramming/fitting



Wired LCD Event Display

The screenshot shows the JAS3 software interface. The main window displays a 3D visualization of particle tracks and detector hits. The tracks are shown as white lines, and the hits are represented by colored squares (green, yellow, blue, purple). The detector is shown as a grid of lines.

The left sidebar contains a tree view of detector components:

- Interaction
 - DetectorType
 - EventType
 - EcalEndcapHitsNNClusters
 - VtxEndcapHits
 - EcalBarrHits
 - EcalEndcapHits
 - ForwardEcalEndcapHits
 - HcalBarrHits
 - HcalEndcapHits
 - LuminosityMonitorHits
 - MuonBarrHits
 - MuonEndcapHits
 - TkrBarrHits
 - TkrEndcapHits
 - VtxBarrHits
 - EcalBarrHitsNNClusters
 - HcalEndcapHitsNNClusters
 - HcalBarrHitsNNClusters
 - MCPParticle
 - Neutral
 - Charged

The right sidebar contains a data table:

sNNClusters/delta phi	2196
sNNClusters/delta phi	0.088715
	1.4588
ghts :	30
	2166.0
	617.94±17.5
	0.014250±7.6218E-3
	0.32966±0.00549
	12.552

The bottom status bar shows the time 9:28:04 PM and the message "compile successful". The bottom right corner shows the memory usage 37.8/51.2MB.

JAS/Wired online

- Many components used by Babar and Fermi for online monitoring and data QA.
- Could be straightforwardly adopted for HPS needs.
- Tightly integrated system of sim/reco/online has many benefits.

Validated

- This suite of software tools provides:
 - Physics event generation & bindings to most legacy generators through the stdhep format.
 - Full detector response simulation using precompiled binaries & runtime geometry definition (no coding!).
 - Full detector digitization (x-talk, noise, diffusion, etc.)
 - Hit-level overlay of arbitrary background events.
 - Access to other LCIO-compliant software frameworks.
 - Full ab-initio event reconstruction and analysis suites.
 - Tested on hundreds of millions of events.
- “From zero to analysis in 15 minutes.”

Using root

- Can analyze output AIDA files using RAIDA.
 - non-official root binding to AIDA
- Can analyze output LCIO files two ways:
 - Using root LCIO Dictionary
 - Using rlcio files, LCIO event data model written as root files
 - output files are larger
 - read times are longer
- Roll your own
 - write out native root files yourself.
 - neither recommended nor supported!

User base

- ILC physics and detector community
 - primarily US and UK members of SiD
- CLIC physics and detector community
 - CERN-based SiD' studies
- MuC physics and detector community
 - FNAL-based
- JLAB heavy-photon search proposals
 - HPS?
 - DarkLight?
- FNAL dual-readout crystal calorimetry R&D group

Simulation Summary

- ALCPG sim/reco supports an ambitious international detector simulation effort. Goal is flexibility and interoperability.
- Provides a complete and flexible detector simulation package capable of simulating arbitrarily complex detectors with runtime detector description.
- Reconstruction & analysis framework was used to characterize the Silicon Detector and was essential to that concept's successful validation in the LOI process.
- LCIO provides interoperability with tools developed in other regions (e.g. jet flavor tagging (LCFI), particle flow (Pandora)), other languages (FORTRAN, java, C++, python) and other analysis frameworks (e.g. Marlin, root).

Additional Information

- Wiki - <http://confluence.slac.stanford.edu/display/ilc/Home>
- lcsim.org - <http://www.lcsim.org>
- ILC Forum - <http://forum.linearcollider.org>
- LCIO - <http://lcio.desy.de>
- SLIC - <http://www.lcsim.org/software/slic>
- LCDD - <http://www.lcsim.org/software/lcdd>
- JAS3 - <http://jas.freehep.org/jas3>
- AIDA - <http://aida.freehep.org>
- WIRED - <http://wired.freehep.org>