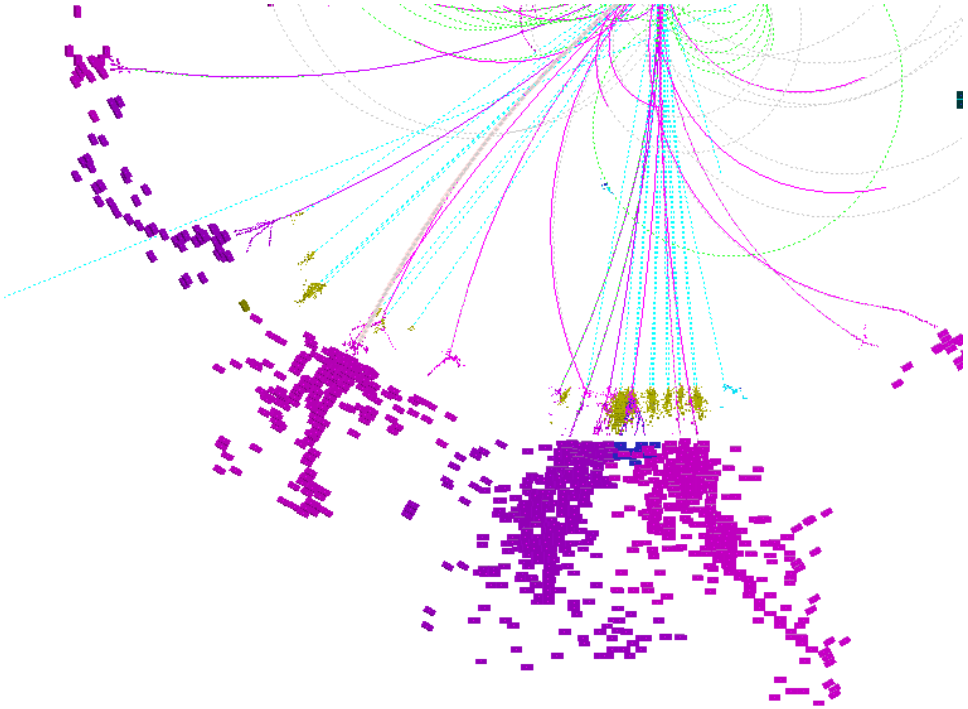
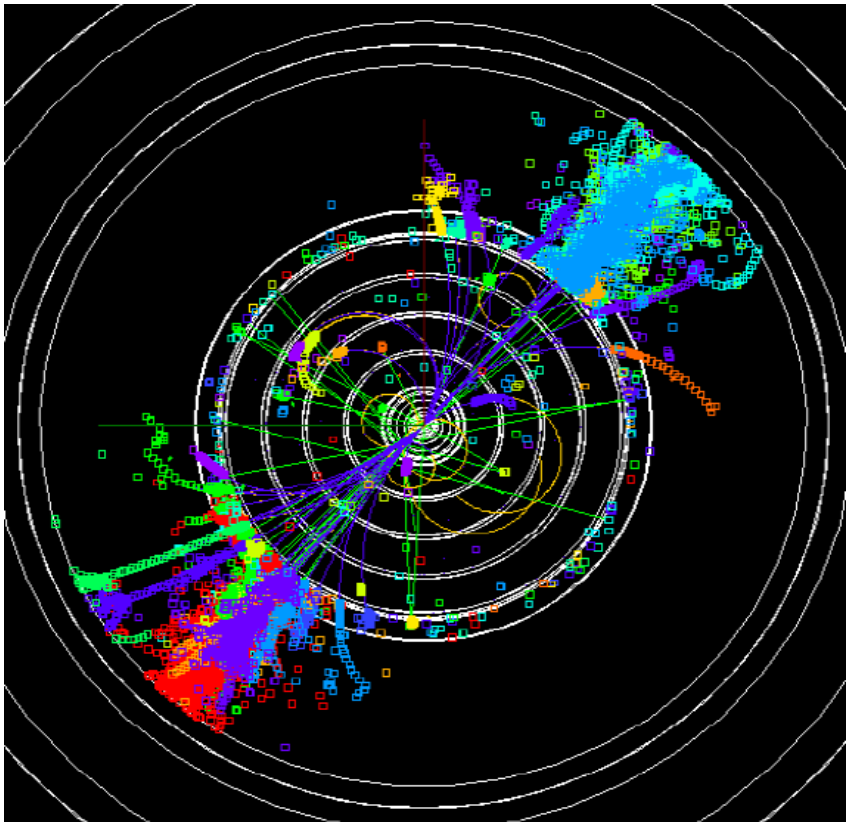
SlicPandora Software

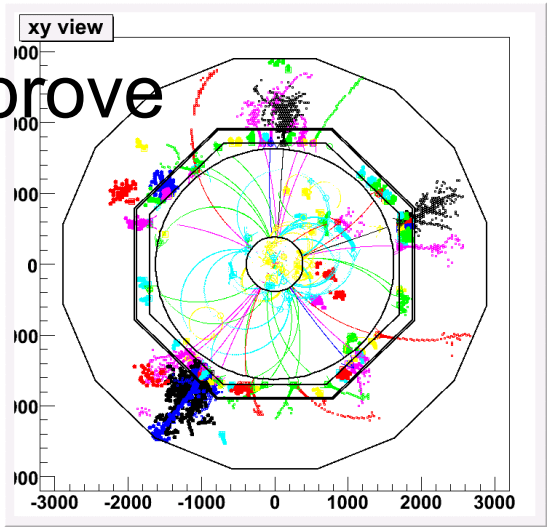
Jeremy McCormick, SLAC





Pandora PFA

- PFA originally created for ILD detector.
- Rewritten from scratch to be more modular and fully extensible; now it is much easier to apply to different detector models and software frameworks.
- No external dependencies.
- Available in a public SVN along with related projects and config files.
- Ongoing development to fix bugs, improve algorithm performance, etc.
- Regularly tagged software releases.
- Also used in ILD software framework.
  - MarlinPandora



Pandora PFA Benefits

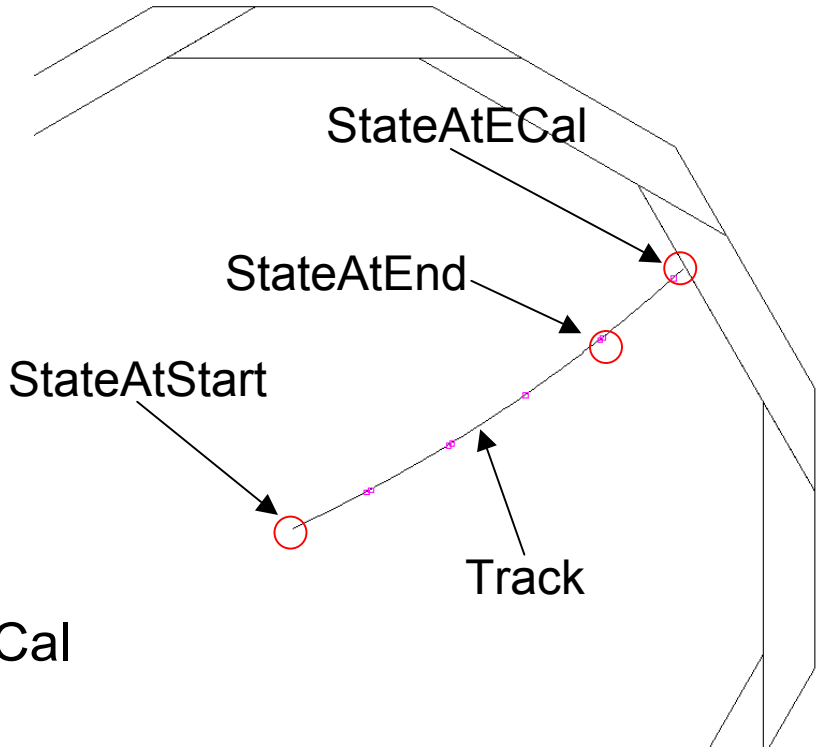
- Best performing PFA algorithm.
- Reasonable performance time, including on events @ 1 TeV and greater.
  - CLIC CDR benchmarks @ 3 TeV
- Allows direct comparison to other detector concepts (e.g. ILD) using the same reconstruction algorithm.
- Take advantage of dedicated developers/author and their expertise.
  - Get bug fixes and improvements “for free.”
- Established set of easy-to-use software tools and utilities.

SlicPandora Workflow

1. Generate LCIO events using slic and detector/stdhep of your choice.
2. Run SLIC output through LCSim tracking to generate Track and Track State collections.
3. Take LCSim LCIO output and run in slicPandora to generate PFOs.
4. Use LCIO analysis tool of your choice to work with PFOs (JAS3, ROOT, etc.).

LCSim Tracking

- SLIC output is run through LCSim's SiSim digitization and SeedTracker tracking algorithm to generate Tracks and TrackStates for Pandora.
- Track States (momentum, position, time)
  - StateAtStart
    - initial momentum
  - StateAtEnd
    - momentum at last hit
    - Pandora doesn't seem to use.
  - StateAtECal
    - momentum of Track swum to ECal
    - Very important to get right.



SlicPandora

- Frontend to the PandoraPFANew project.
  - LCIO binding
  - Geometry format
- Uses geometry XML file generated by GeomConverter from compact detector.
- Reads input LCIO file with simulated events, tracks, and track states.
- Outputs LCIO file with Reconstructed Particles.
- Uses standard XML config file for Pandora algorithm settings.

Software Dependencies

- slicPandora
  - top level project that provides the binary
- PandoraPFANew
  - standalone Pandora algorithm
- LCIO
  - common IO framework
- PandoraMonitoring (optional)
  - ROOT-based event visualization
- ROOT (optional)
  - need if monitoring is enabled
  - running with version 5.26

- NOTE: At this moment, it is best to use the CVS head of all packages unless otherwise indicated.

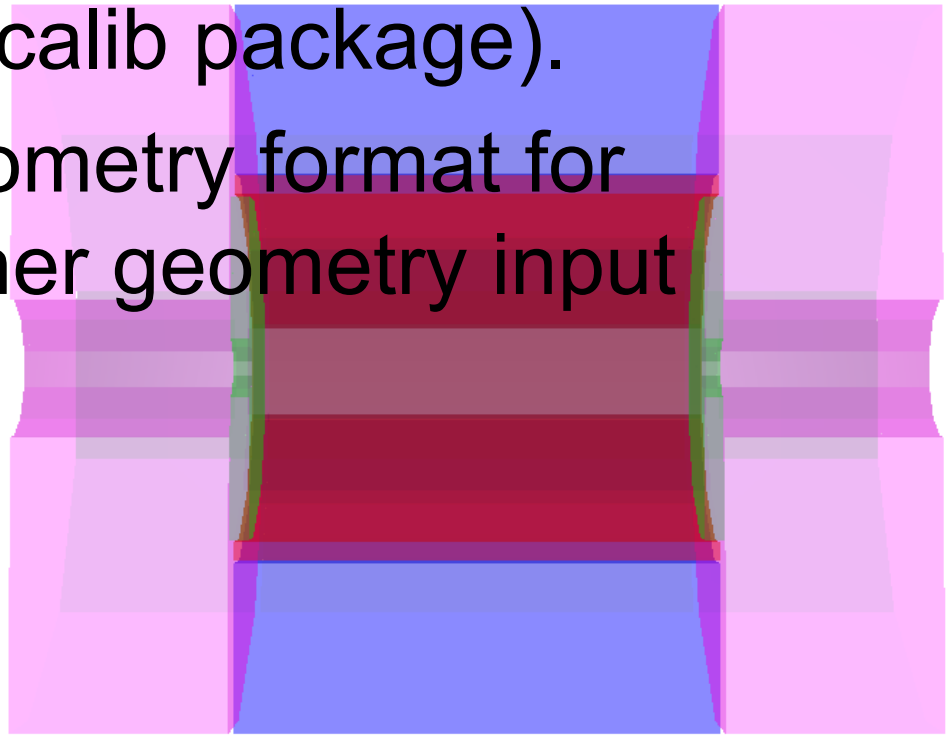
Command Line Interface

./bin/PandoraFrontend geometry.xml pandoraSettings.xml inputEvents.slcio reconOutput.slcio nevents nskip

- **geometry.xml** is a Pandora format from GeomConverter.
- **pandoraSettings.xml** is the Pandora config (default of slicPandora/examples/PandoraSettings.xml is fine)
- **nevents** is optional and defaults to all events.
- **nskip** is optional and defaults to zero.
- Grid ready using Dirac on the LCG grid.
- Setup script provided by slicPandora for runtime libraries.

SlicPandora Geometry Format

- GeomConverter output binding that operates similar to LCDD, etc.
- Generated from compact description plus calorimeter sampling fractions conditions file (using lcsim-cal-calib package).
- Fully descriptive geometry format for slicPandora. No other geometry input data required.



Geometry XML Format

```
<pandoraSetup>
    <detector>
    <calorimeters>
        <calorimeter>
            <id>
            <layers>
    <coil>
    <tracking>
```

Pandora Detector Parameters

- calorimeter
  - type
  - innerR, innerZ, innerPhi, innerSymmetryOrder, outerR, outerZ, outerPhi, outerSymmetryOrder
  - hits collection name
  - cellSizeU, cellSizeV
  - mipEnergy, mipSigma, mipCut, timeCut
  - digital flag
  - id
  - layers
- coil
  - bfield, innerR, z, outerR
- tracking
  - innerR, outerR, z

Calorimeter Layer Parameters

- Parameters provided for *each layer*.
  - radiation lengths, interaction lengths
  - distance from IP
  - cell thickness (includes absorber + dead material)
  - sampling fraction
  - EM sampling fraction
  - HAD sampling fraction

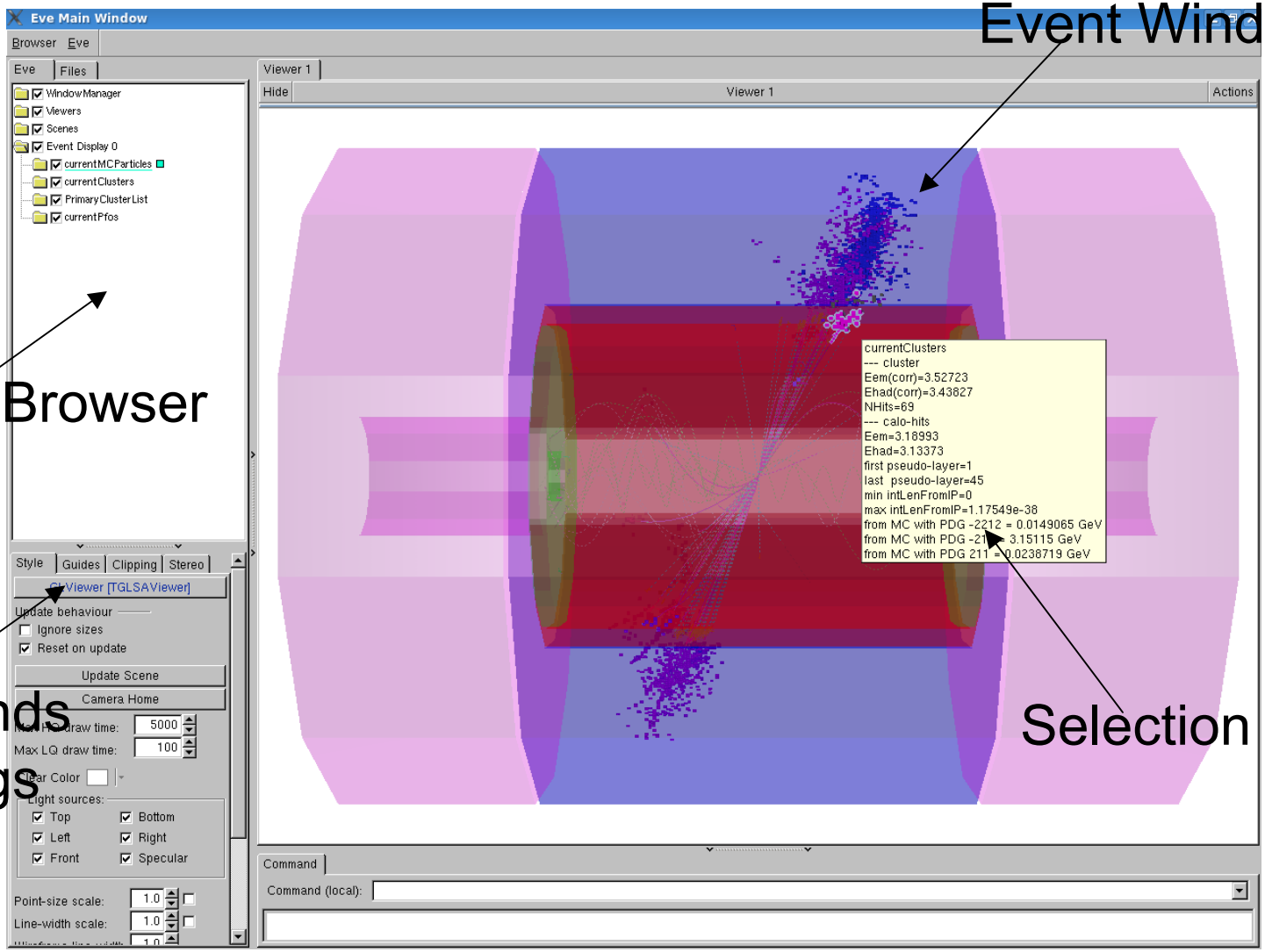
Calorimeter Calibration

- The lcsim-cal-calib package is used to generate sampling fractions from single particle data.
  - Range of single particle types and energies simulated in SLIC.
  - Events are analyzed in LCSim to derive sampling fractions.
  - Output sampling fractions put by hand into CalorimeterCalibration.properties file in detector directory.
- Same events are run in Pandora and fit again using a different Driver.
  - Isolated hits are removed by Pandora, so data must be refitted to match its final cluster energies.
  - Final sampling fractions put into CalorimeterCalibration.properties file.

Pandora Monitoring

- Event visualization using ROOT's Eve library.
- Visualization of detector, hits/cells, tracks, MCParticles, PFOs, etc.
- Various color coding schemes.
- Runs within Pandora, event by event.
- NOTE: JAS3/Wired4 can still be used as an event display with the output LCIO file from Pandora.

PandoraMonitoring Example



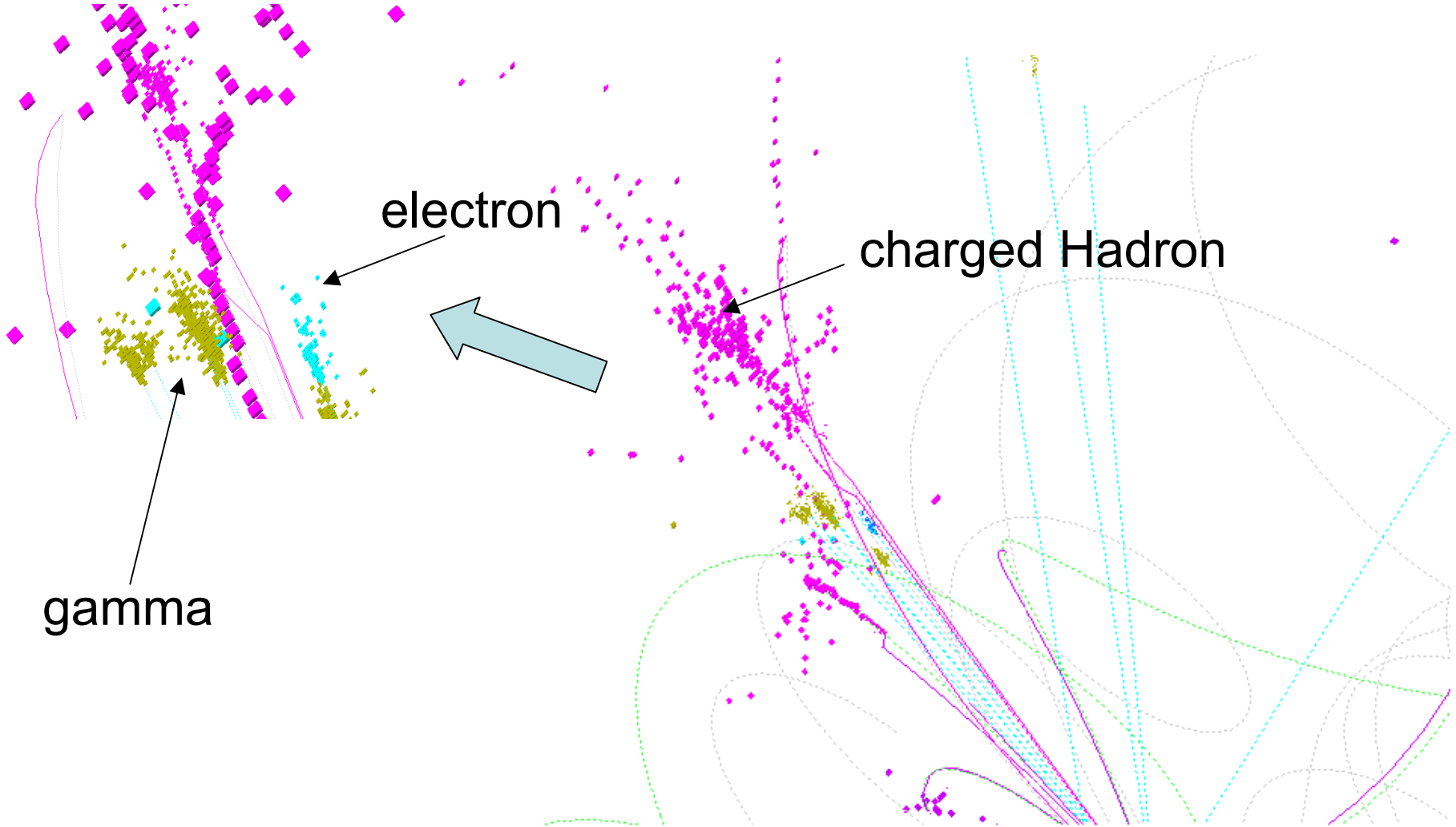
Event Window

Scene Browser

Commands & Settings

Selection Info

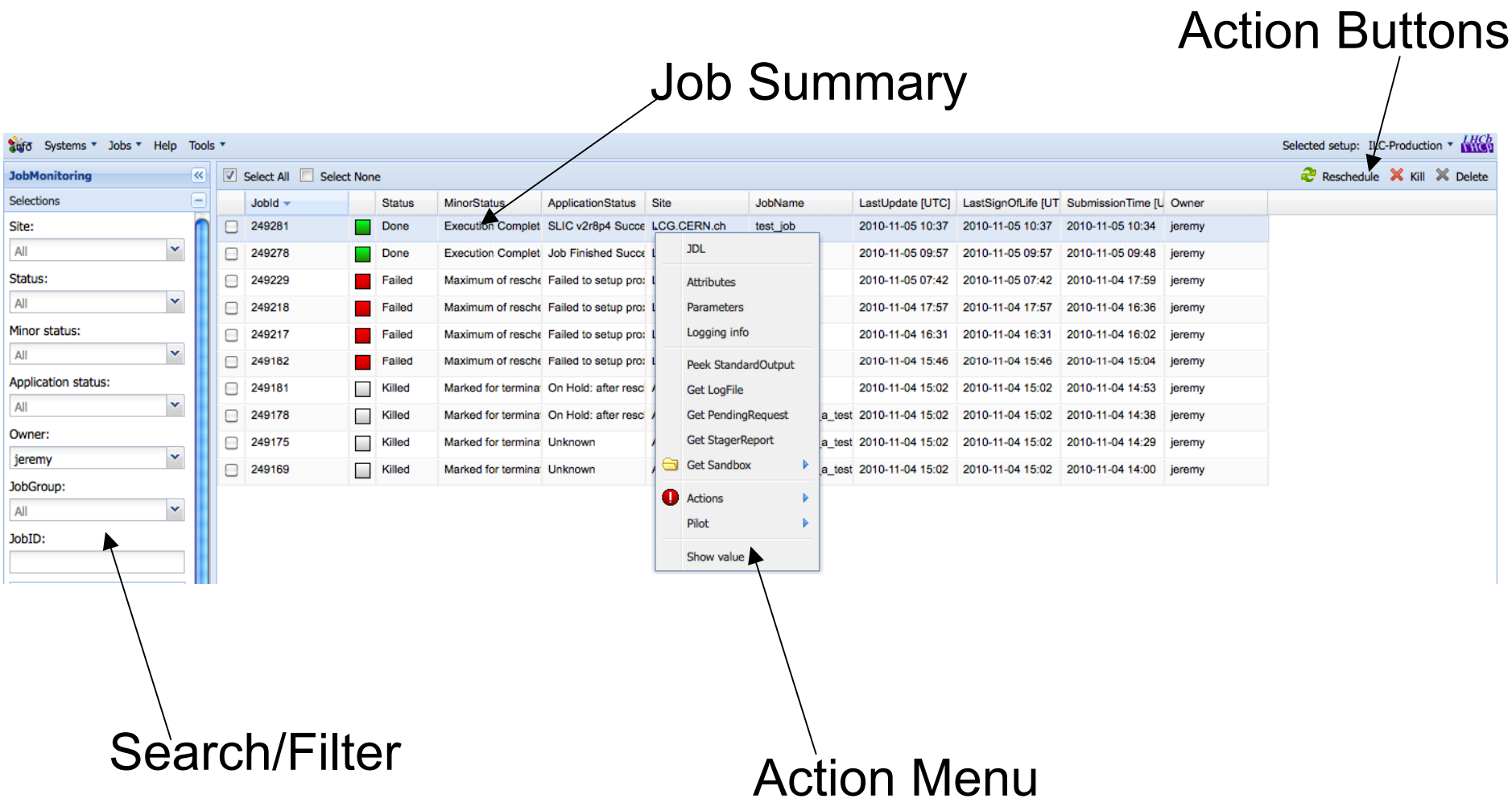
250 GeV W Jet



Grid Tools

- CLIC SiD group has provided a comprehensive set of tools for using slicPandora and related apps on the LCG grid.
  - slic, lcsim, slicPandora
- Built on Dirac gridware (from LHCb).
- Possible to run a job that chains applications
  - Run slic, lcsim, & slicPandora in the same job script.
- Need to get a grid certificate and then register with the European ILC VO to use it.
- Jobs are submitted using python scripts.
- Remote job submission possible and relatively straightforward (tested at SLAC).
- Output data files can be stored in the grid catalog and used by subsequent jobs.
- WWW monitoring tool to check status of jobs.

Online Job Monitoring



ILCDirac API

Example SLIC Job

```
dirac = DiracILC ( … )
job = ILCJob()
job.setSLIC( slicVer,
             macroFile,
             inputStdhepFile,
             outputFile,
             detectorModel,
             nevents)
[…more setup here…]
job.setName( “my_grid_job” )
dirac.submit( job )
```

Documentation

- SlicPandora Instructions
- LCD Wiki
- ILCDirac API