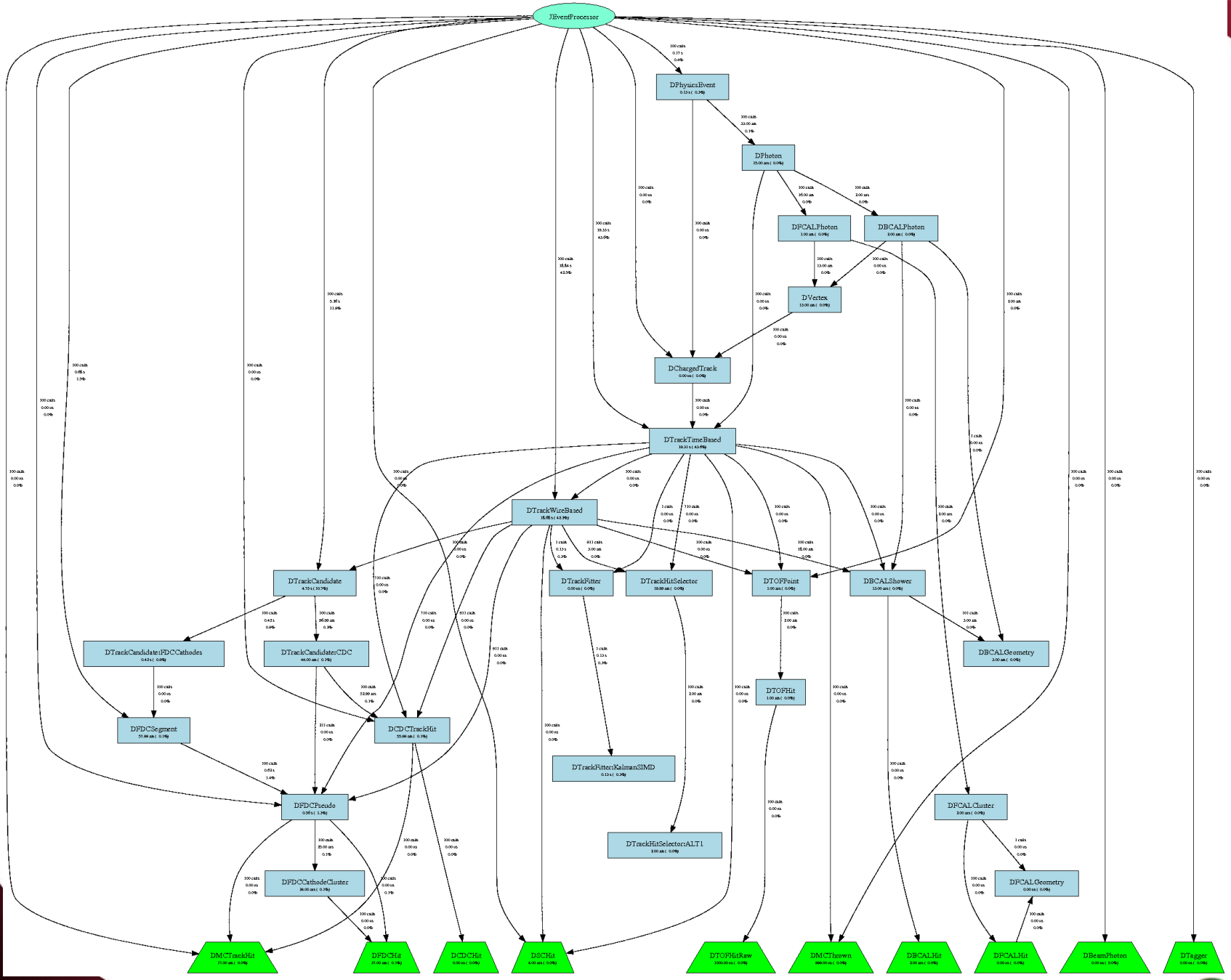
GlueX Reconstruction

Simon Taylor / JLab

- Overview of full reconstruction
- Charged particle reconstruction
- Photon reconstruction



Jefferson Lab

Top level: DPhysicsEvent

- The *DPhysicsEvent* class describes fully reconstructed events consisting of the following:
  - Vertex information: class=*DVertex*
  - Lists of charged particles: class=*DTrackTimeBased*
    - π^+, π^-, K^+, K^-, proton, other +, other -
  - List of reconstructed photons: class=*DPhoton*
    - Derived from clusters reconstructed in FCAL and BCAL

```
vector<const DPhysicsEvent*>physics_events;
eventLoop->Get(physics_events);
...
// assuming we've checked that the number of protons>0...
const DTrackTimeBased *proton=physics_events[0]->proton[0];

// assuming we've checked that the number of photons>0
const DPhoton *photon=physics_events[0]->photon[0];
```

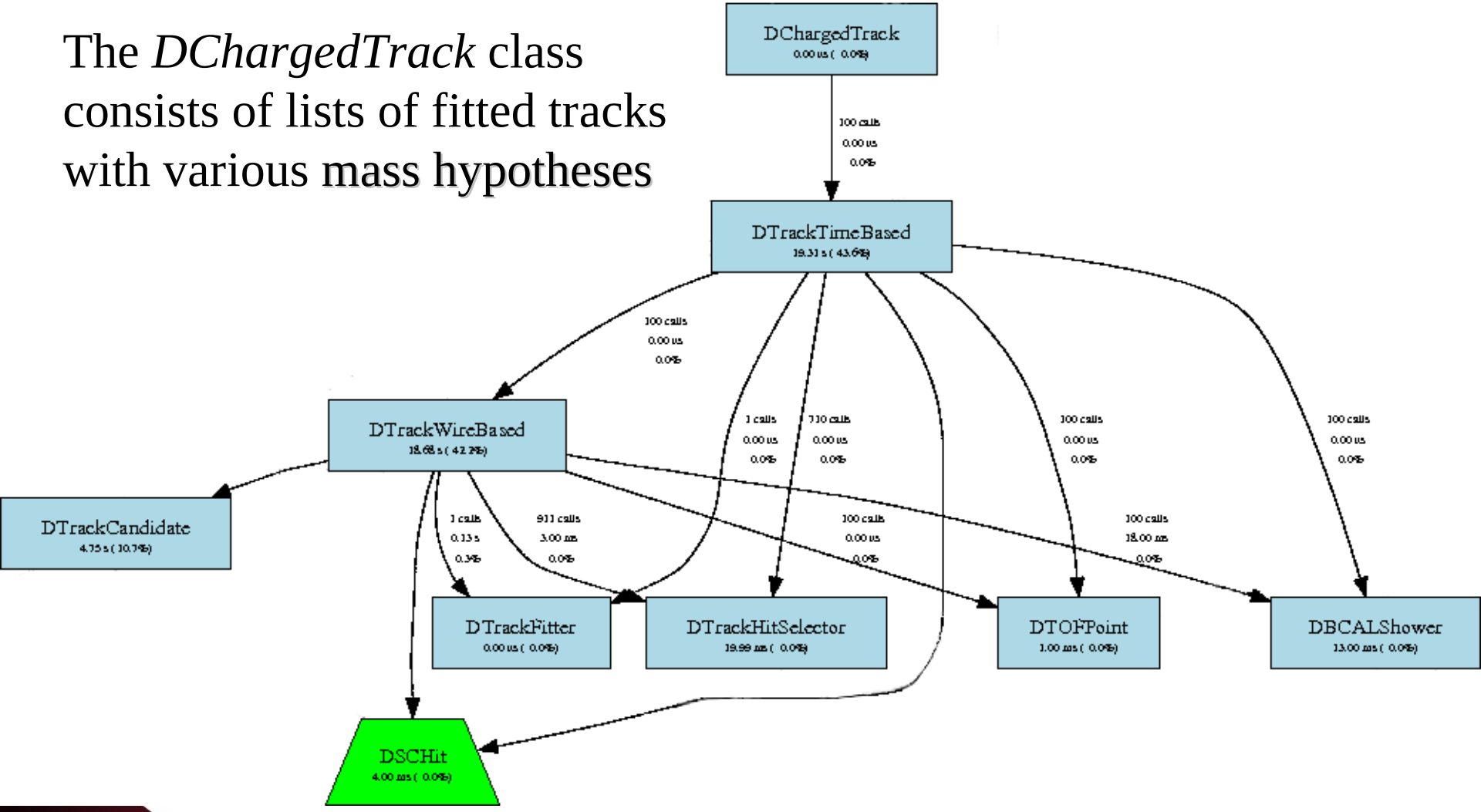
Kinematic data class

- Both *DPhoton* and *DTrackTimeBased* inherit from the *DKinematicData* class
- *DKinematicData* contains methods for 3-vector and 4-vector storage and manipulation for kinematic quantities
  - Error matrices also included

```
// Get the 4-momentum of the first pi-minus in the event ev
const DLorentzVector pim=ev->pim[0]->lorentzMomentum()

// Get the magnitude of the momentum of the first proton
double p=ev->proton[0]->momentum().Mag();

// Get the proton's 5x5 tracking error matrix
const DMatrixDSym cov=ev->proton[0]->TrackingErrorMatrix();

// Get the pion's 7x7 (4-momentum and position) error matrix
const DMatrixDSym cov2=ev->pim[0]->errorMatrix();
```

Charged particle reconstruction

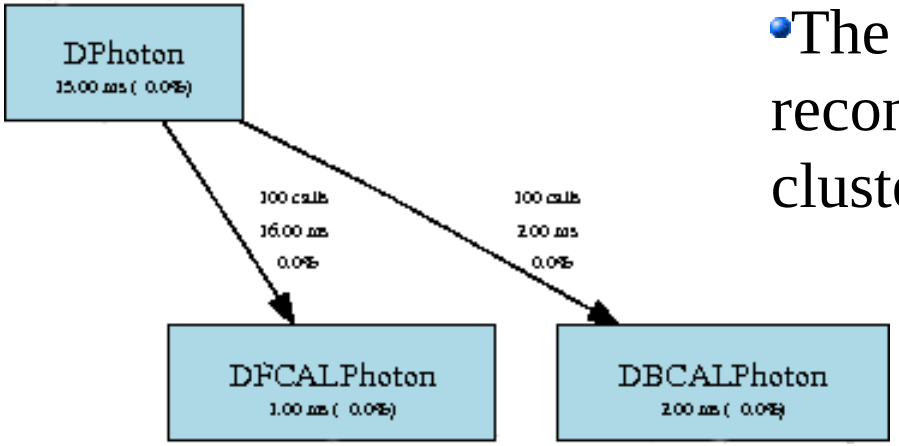
The *DChargedTrack* class consists of lists of fitted tracks with various mass hypotheses



Charged particle reconstruction

- Step 1: hits in CDC and FDC are associated into segments and fit with a helical model (assuming constant B) to form track candidates → *DTrackCandidate*
- Step 2: each track candidate provides the initial guess for wires-only fits (no drift times are used at this stage) → *DTrackWireBased*
  - Wire-based fit performed multiple times over a list of mass hypotheses for each candidate
  - A few tracking algorithms are available:
    - Kalman Filter (default), Riemann Fitter (helical model), Global Least Squares fitter
- Step 3: each wire-based track is re-fitted using the drift-times from the hits associated with the track → *DtrackTimeBased*
  - "Figure of merit" (FOM) for each fit assesses quality of mass hypothesis
    - Step 4: Time-based tracks are sorted according to FOM → *DChargedTrack*

Photon reconstruction



- The *DPhoton* class contains "photons" reconstructed from FCAL and BCAL clusters
  - Code attempts to match track projections to showers

IMPORTANT NOTE: ***all reconstructed showers are listed, even those that are associated with tracks***

- Photon candidates that have been matched geometrically with tracks are flagged with the tag kCharge

Photon reconstruction

- FCAL reconstruction merges individual hits in FCAL crystals into clusters $\rightarrow$ *DFCALCluster*
  - Based on RadPhi algorithm
  - Cluster energy and position are corrected for shower depth and non-linear effects and nearby clusters are merged together to form photon candidates $\rightarrow$ *DFCALPhoton*
- BCAL reconstruction associates upstream and downstream hits in the BCAL fibers into single-cell hits, merges these hits into clusters, and merges nearby clusters into showers $\rightarrow$ *DBCALShower*
  - Based on KLOE algorithm
  - Showers are corrected for dark noise and non-linear scaling factors and further shower merging takes place to form photon candidates $\rightarrow$ *DBCALPhoton*