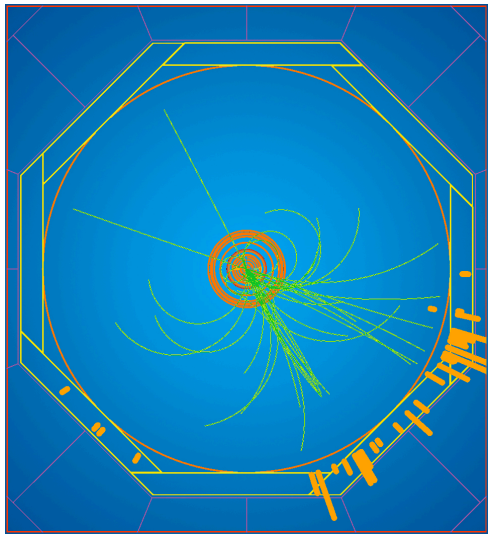
Tracking: New tracking tools in the context of ILD and how it will evolve to generic tracking

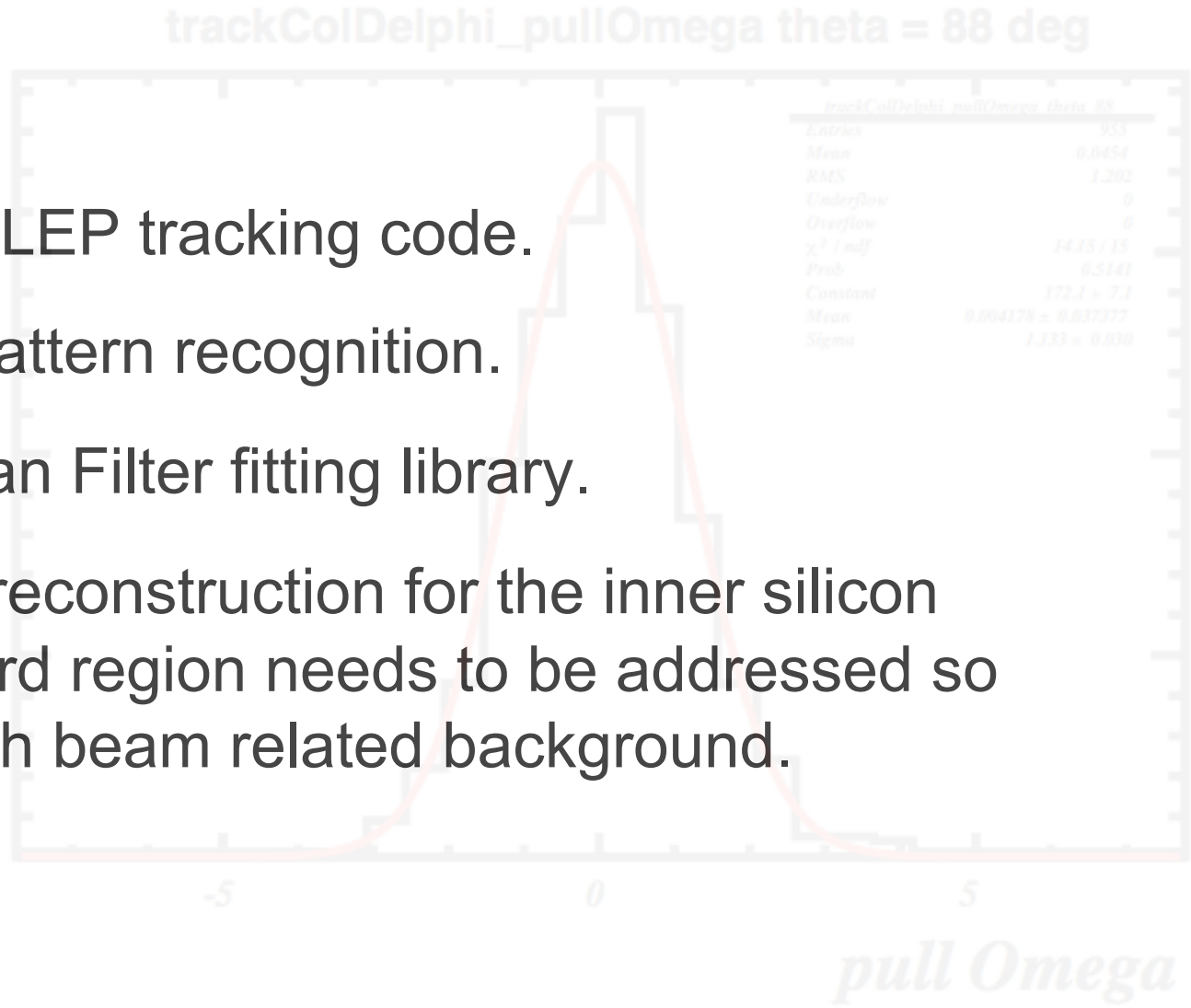
Steve Aplin



Linear Collider Software Meeting
3rd January 2011

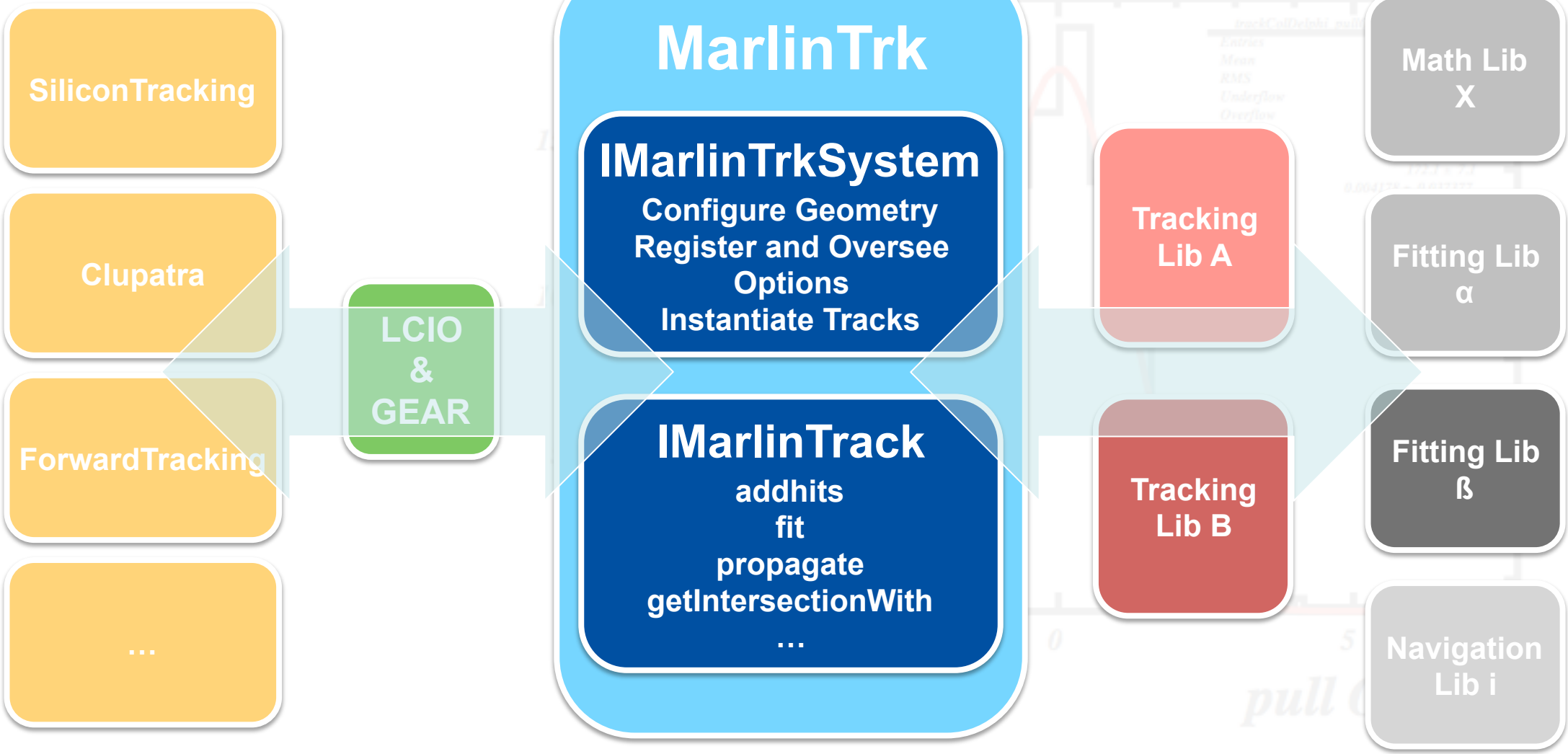
iLC SOft

Tracking Code rewrite for the DBD

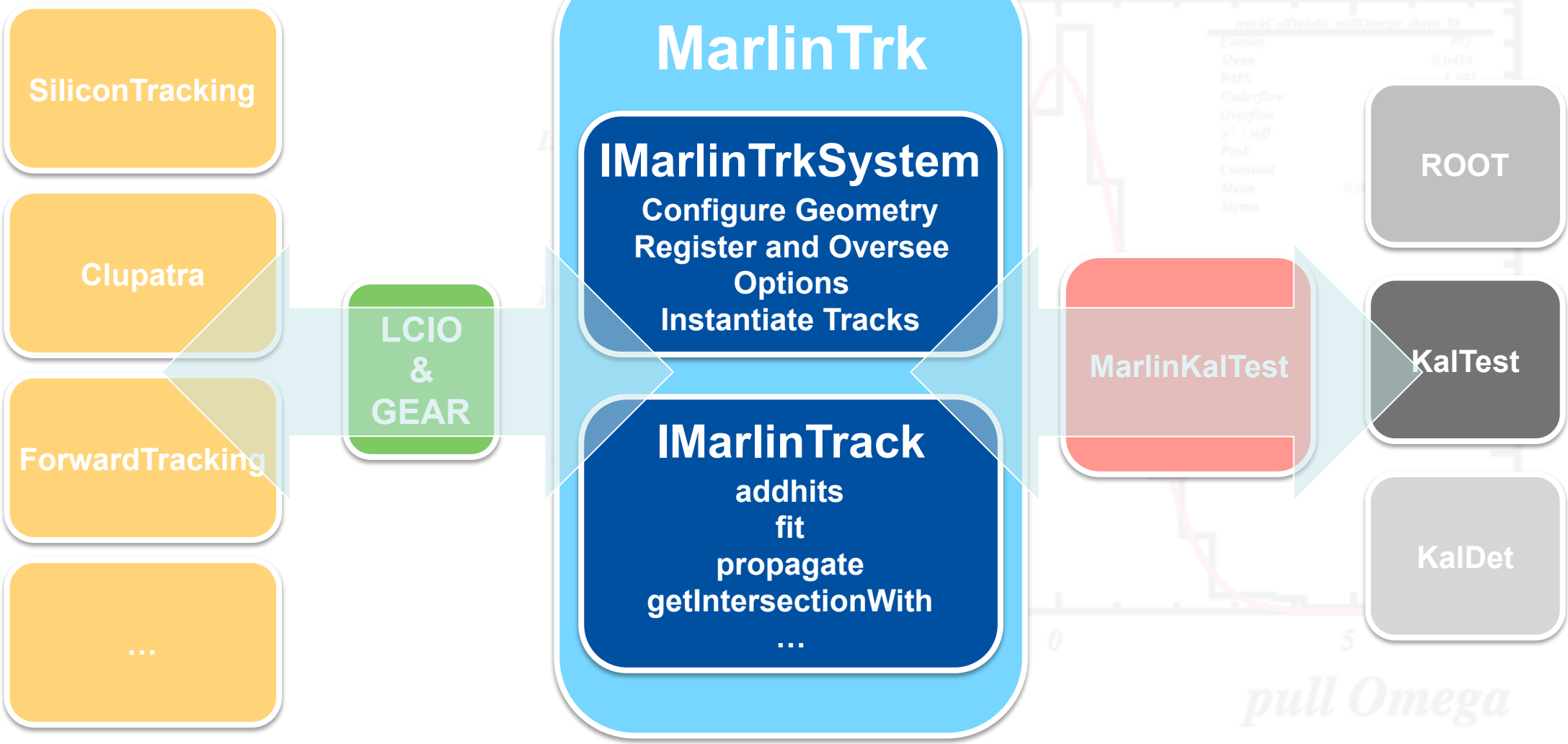


- Leave behind F77 LEP tracking code.
- Rewrite the TPC pattern recognition.
- Use KalTest Kalman Filter fitting library.
- Stand alone track reconstruction for the inner silicon trackers and forward region needs to be addressed so that it can cope with beam related background.

IMarlinTrack and IMarlinTrkSystem



MarlinTrk KalTest Implementation



IMarlinTrack Interface

- **IMarlinTrack** interface provides a convenient interface when using an iterative fitter and also during pattern recognition.
- Examples of methods provided:

```
//** initialise the fit using the supplied hits only, using the given order to determine the direction of the track
virtual int initialise( bool direction ) = 0 ;

//** initialise the fit with a track state
virtual int initialise( const IMPL::TrackStateImpl& ts) = 0 ;

//** update the current fit using the supplied hit, return code via int. Provides the Chi2 increment to the fit from adding the hit via reference.
virtual int addAndFit( EVENT::TrackerHit* hit, double& chi2increment, double maxChi2Increment=DBL_MAX ) = 0 ;

//** get track state, return code via int
virtual int getTrackState( IMPL::TrackStateImpl& ts ) = 0 ;

//** get track state at measurement associated with the given hit, return code via int
virtual int getTrackState( EVENT::TrackerHit* hit, IMPL::TrackStateImpl& ts ) = 0 ;
```

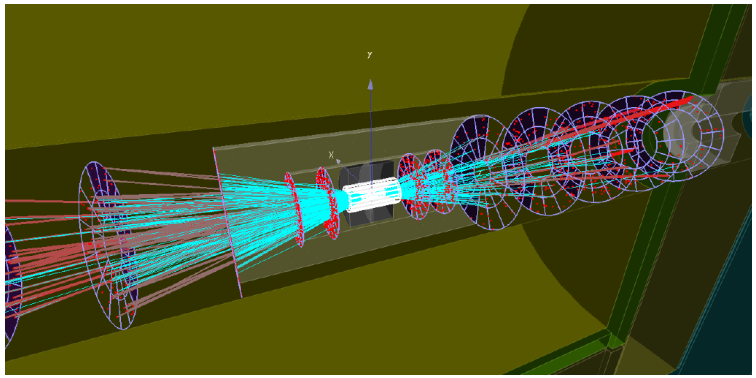
continued …

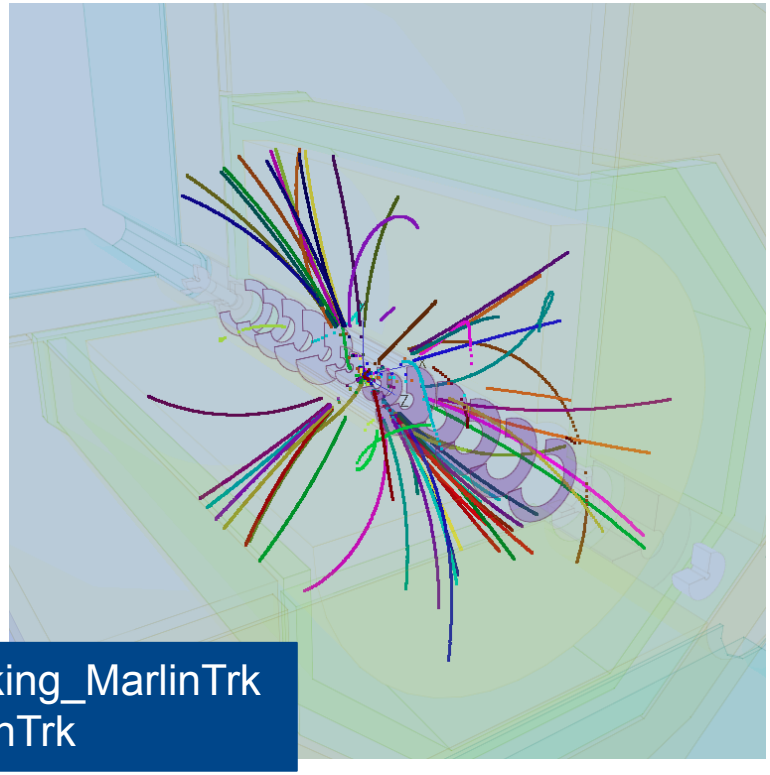
IMarlinTrack and IMarlinTrkSystem

```
//** propagate track state at measurement associated with the given hit, the fit to the point of closest approach to the given point.
virtual int propagate( const gear::Vector3D& point, EVENT::TrackerHit* hit, IMPL::TrackStateImpl& ts) = 0 ;

//** propagate track state at measurement associated with the given hit, to numbered sensitive layer, returning TrackState via provided reference
virtual int propagateToLayer( bool direction, int layerNumber, EVENT::TrackerHit* hit, IMPL::TrackStateImpl& ts) = 0 ;

//** extrapolate track state at measurement associated with the given hit, to the point of closest approach to the given point.
virtual int extrapolate( const gear::Vector3D& point, EVENT::TrackerHit* hit, IMPL::TrackStateImpl& ts) = 0 ;

//** extrapolate track state at measurement associated with the given hit, to numbered sensitive layer, returning TrackState via provided reference
virtual int extrapolateToLayer( bool direction, int layerNumber, EVENT::TrackerHit* hit, IMPL::TrackStateImpl& ts) = 0 ;

//** extrapolate track state at measurement associated with the given hit, to numbered sensitive layer, returning intersection point in global coordinates
virtual int intersectionWithLayer( bool direction, int layerNumber, EVENT::TrackerHit* hit, gear::Vector3D& point) = 0 ;
```

Complete Track Reconstruction

- TPC Pattern Recognition, Clupatra being refined for running on dense jets at the TeV scale.
- Both SiliconTracking and FullLDCTracking have been fully adapted to use MarlinTrk.
- Tracks produced by Forward Tracking still to be incorporated into the final track collection.
- Validation On-going.





ttbar event @ 500 GeV reconstructed using Clupatra and SiliconTracking_MarlinTrk then combined into full tracks using FullLDCTracking_MarlinTrk

Plans for Tracking Package in Aida

- We need to design it in a way that it is generic enough such that it could be used for other such studies into detector R&D.
- This places the emphasis on providing a toolkit which includes the necessary building blocks needed to put together a realistic track reconstruction.
- Joint activity between DESY and OEAW.

Summary

- Of course we need to have a fully working tracking package available at all times.
- We continue to prototype ideas in the iLCSoft framework, which can then be implemented within the Tracking Toolkit for AIDA as it develops.
- Interaction between tracking, geometry and clients will be key …