Integrated
Tracking-Clustering

Dmitry Onoprienko

Fermilab

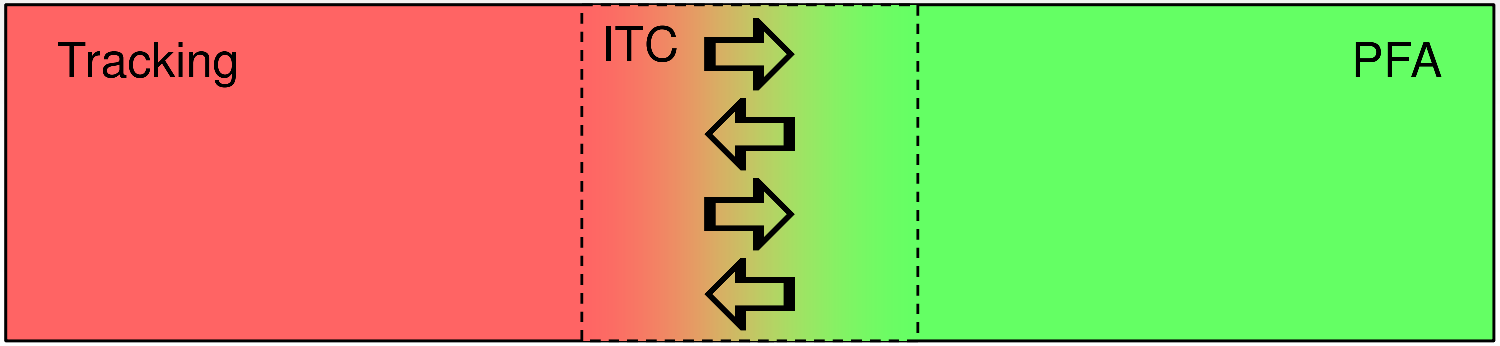
2009 Linear Collider Workshop of the Americas.
Albuquerque, September-October 2009

What is ITC

Evolved from Calorimeter-Assisted track finder for SiD

↳immediate task: non-prompt track reconstruction

Can do more...

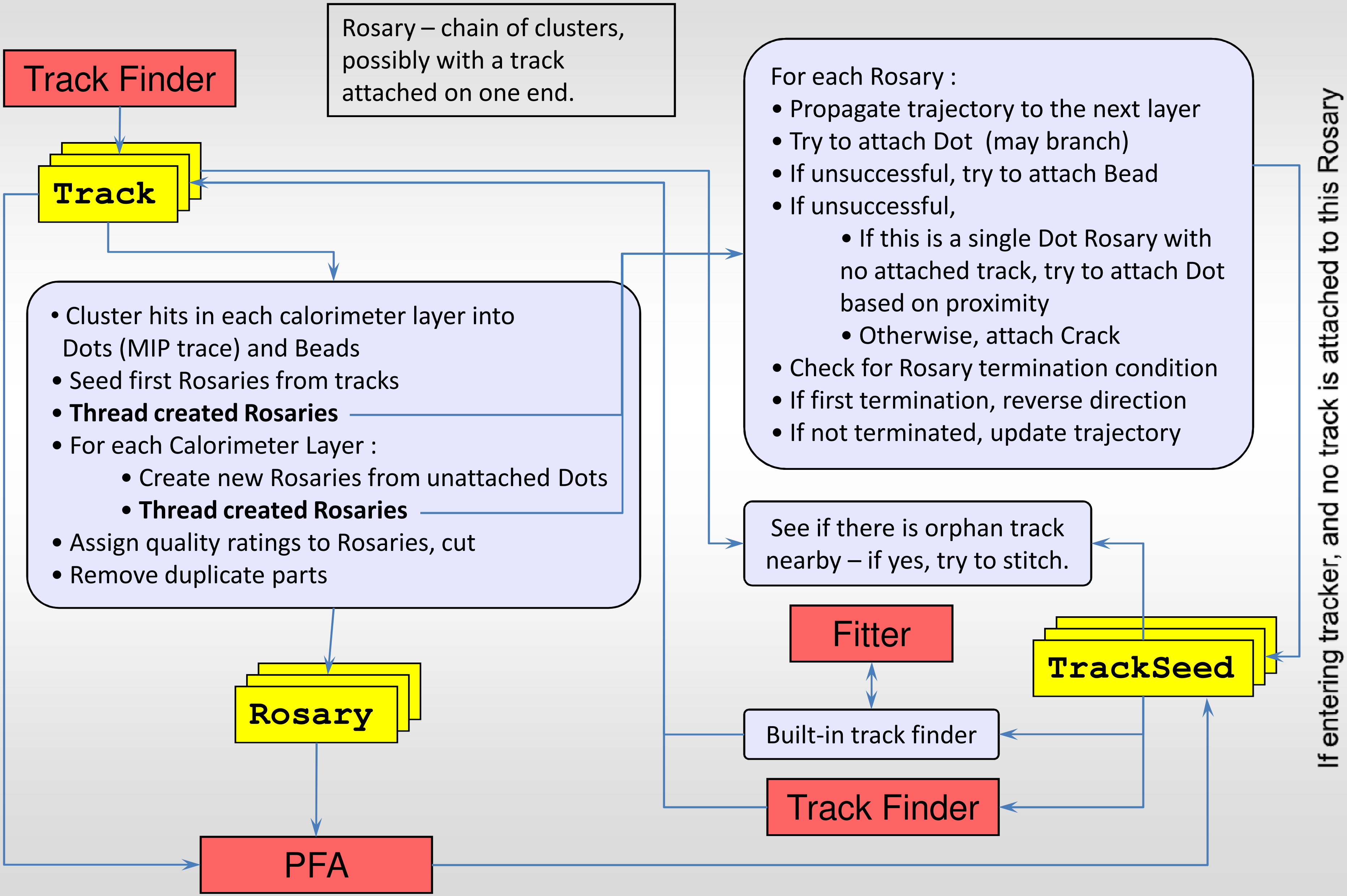


Two-way information exchange between tracker and calorimeter

Requirements:

- extremely flexible
- decoupled from any particular algorithm on either side
- extendable

Basic Algorithm

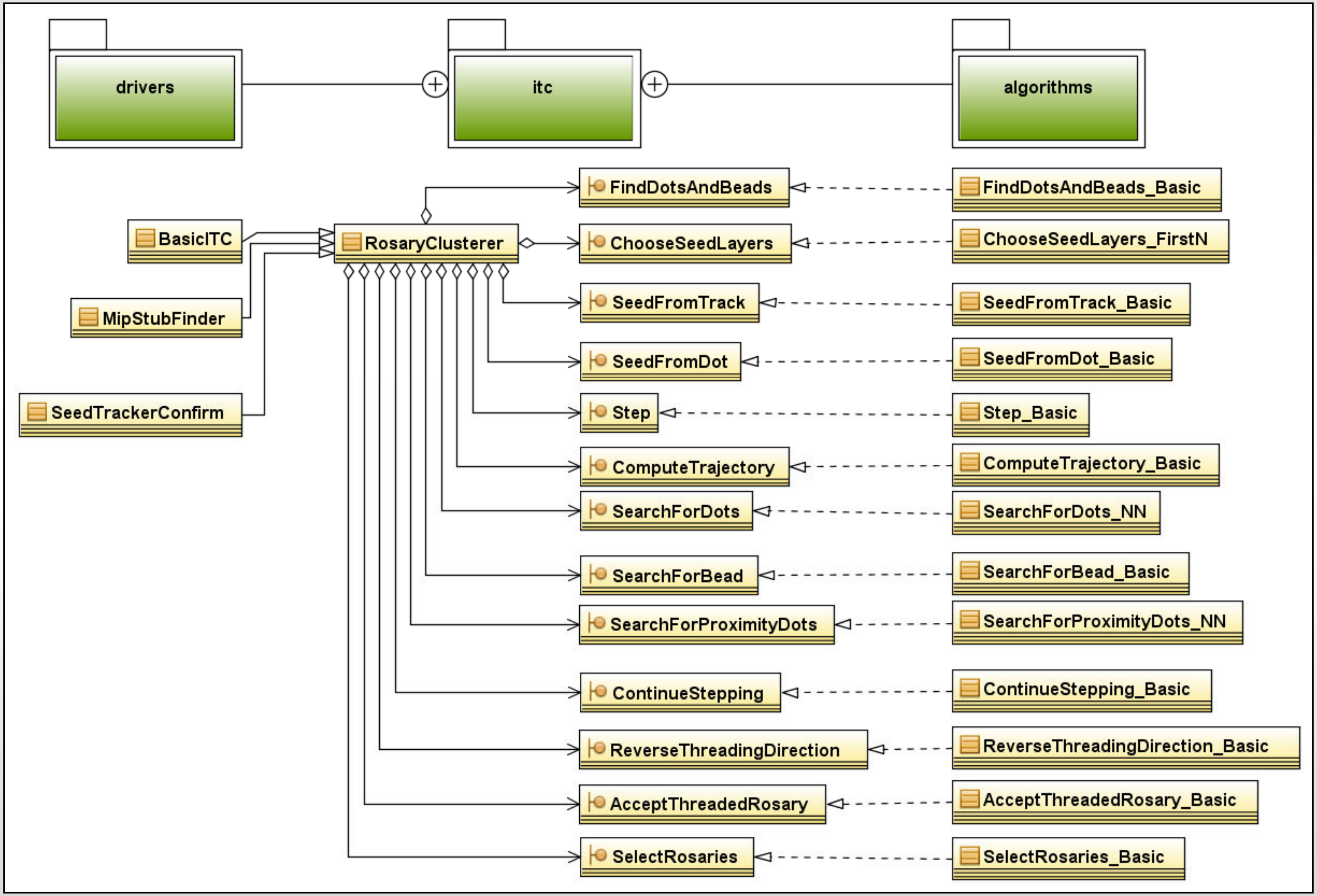


Main Driver + Libraries

Pre-configured drivers:

Skeleton Driver:

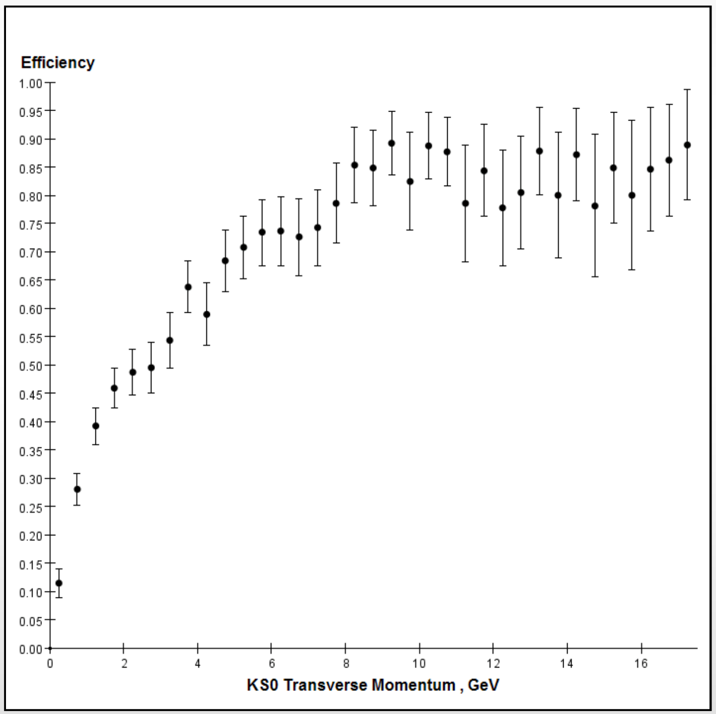
Plug-Ins (more available, new can be added):



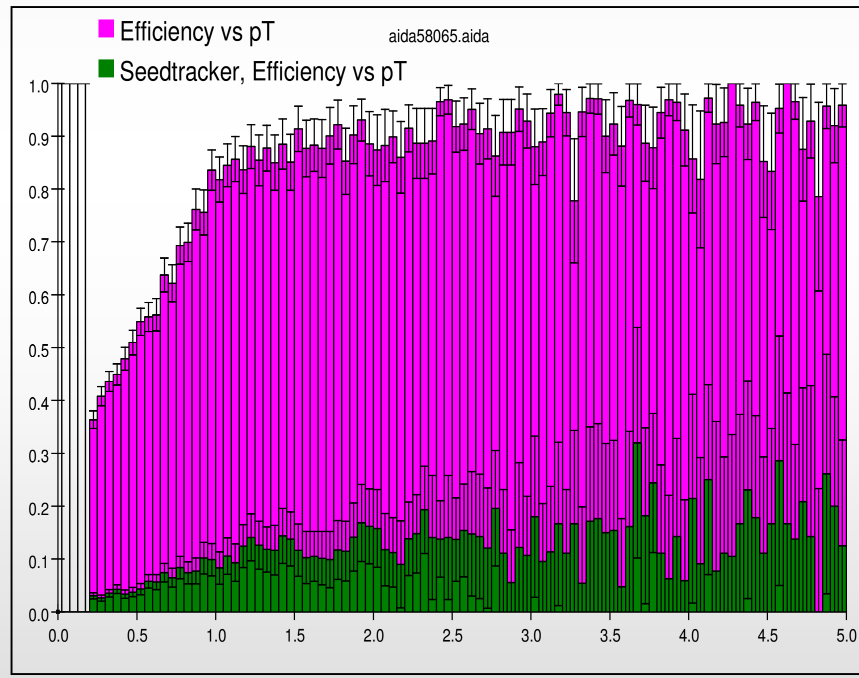
Example of Performance

SiD02 detector, ttbar @ 500GeV

Out-of-the-box MipStubFinder



Findable K_S^0



Non-prompt tracks

Examples of Use Scenarios

(focusing on tracking)

Follow-up on standalone track finder:

- Run SeedTracker with default settings
- Run ITC with built-in track finder and fitter

Fake rate reduction:

- Run SeedTracker with default settings
- Run ITC with no track finder
- Run SeedTracker with default set of strategies but relaxed cuts on found tracks
- Remove tracks not matched to ITC-produced seeds

Seed standalone track finder:

- Run SeedTracker with default settings
- Run ITC with SeedTracker as a track finder

Integrate with standalone track finder:

- Run SeedTracker with default settings
- Run ITC with no track finder
- Run SeedTracker with outside-in set of strategies and relaxed cuts, use ITC-produced seeds instead of confirmation layer

All of the above in a single reconstruction job !

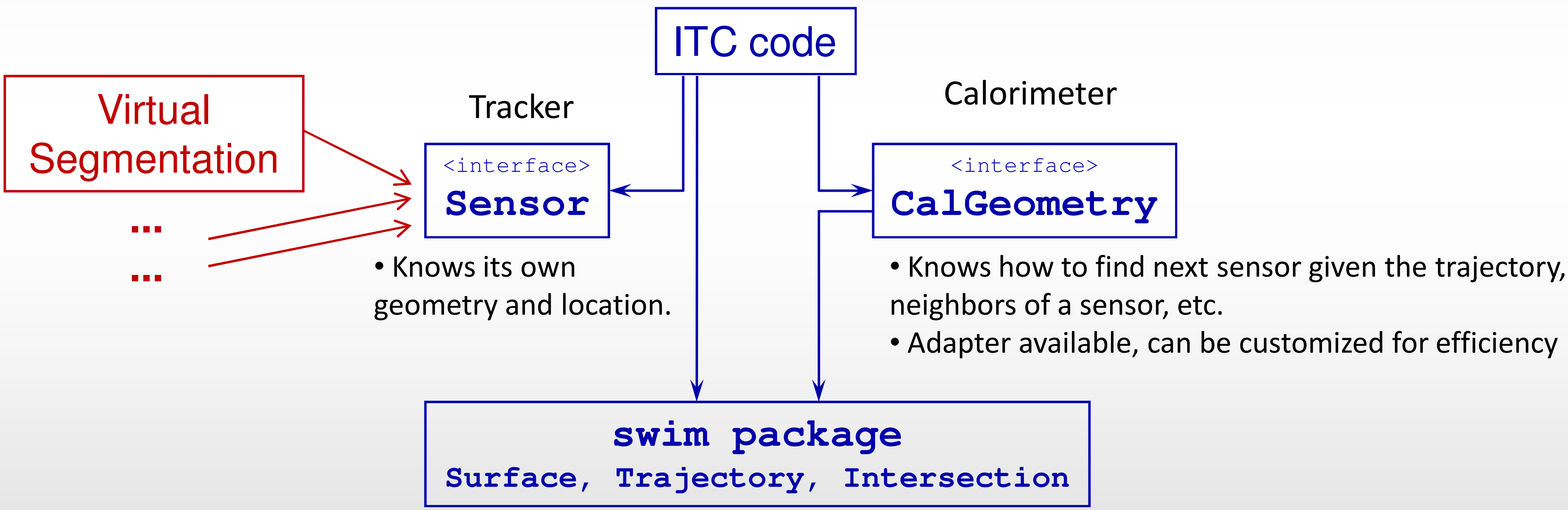
Interoperability with Other Packages

Implemented as a package in org.lcsim

Uses its own object model:

`org.lcsim.contrib.onoprien.data`
LCIO-compatible, WIRED-compatible
Converters provided to/from other hit/track classes used by several SiD algorithms

Geometry independent:



Small Print:

ITC expects to have standard org.lcsim geometry services available
Some classes in the algorithms library are wrappers for other people code

Status & Plans

Core package – functional and documented

- need to provide example drivers
- waiting for better standalone fitter to become available in org.lcsim

Integration with SeedTracker – prototype functional

- releasable version is in the works

Interoperability – requires use of converters

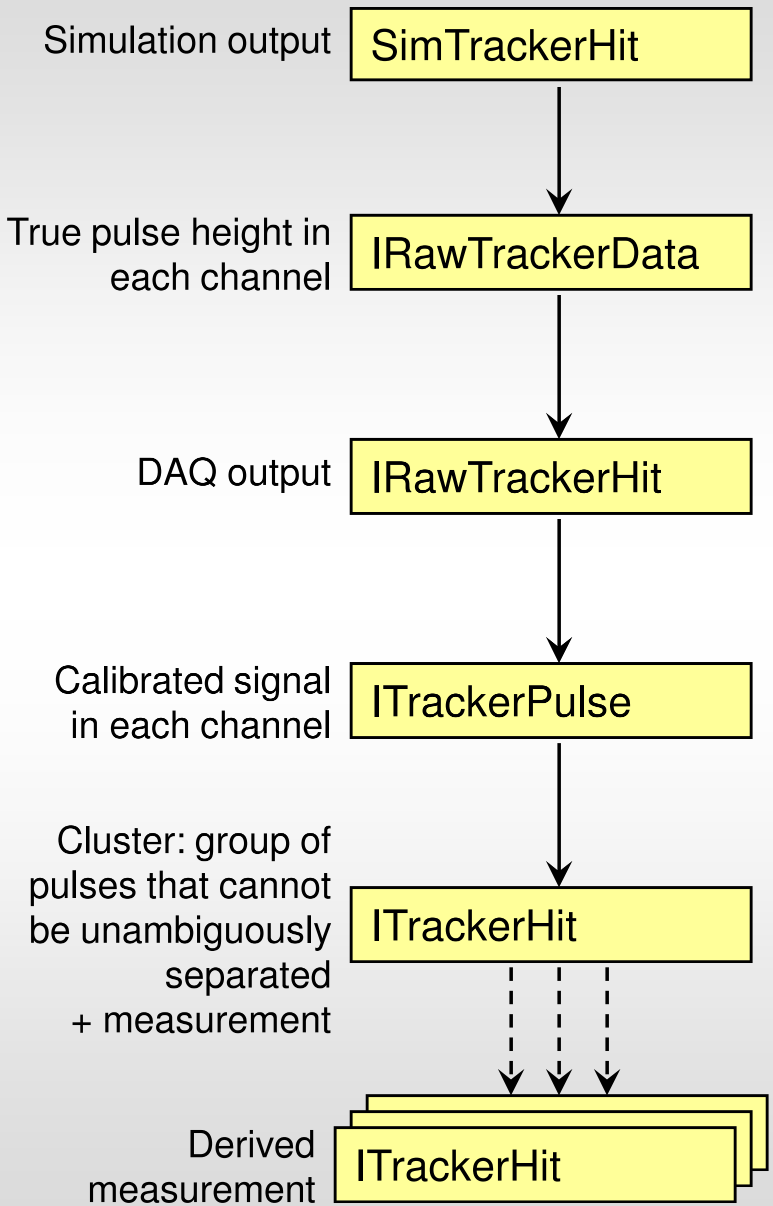
- need to standardize object model and infrastructure !
- waiting for LCIO 2.0 ?

Miscellaneous supporting packages – functional and documented

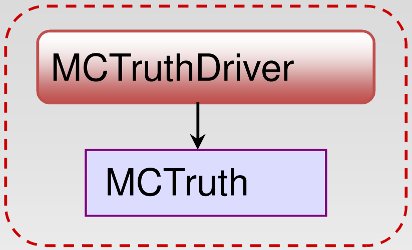
job services and management, MC truth access, performance testing and configurable cheaters, geometry services, auxiliary drivers, etc.

Backup slides

Object model



LCIO based persistency



ITrackerHit extends ITrackerObject:

List<ITrackerHit> getParentHits()
List<ITrackerHit> getClusters()
List<ITrackerPulse> getPulses()

IRefFrame getRefFrame()
Type getType()
Hep3Vector getDimensions();

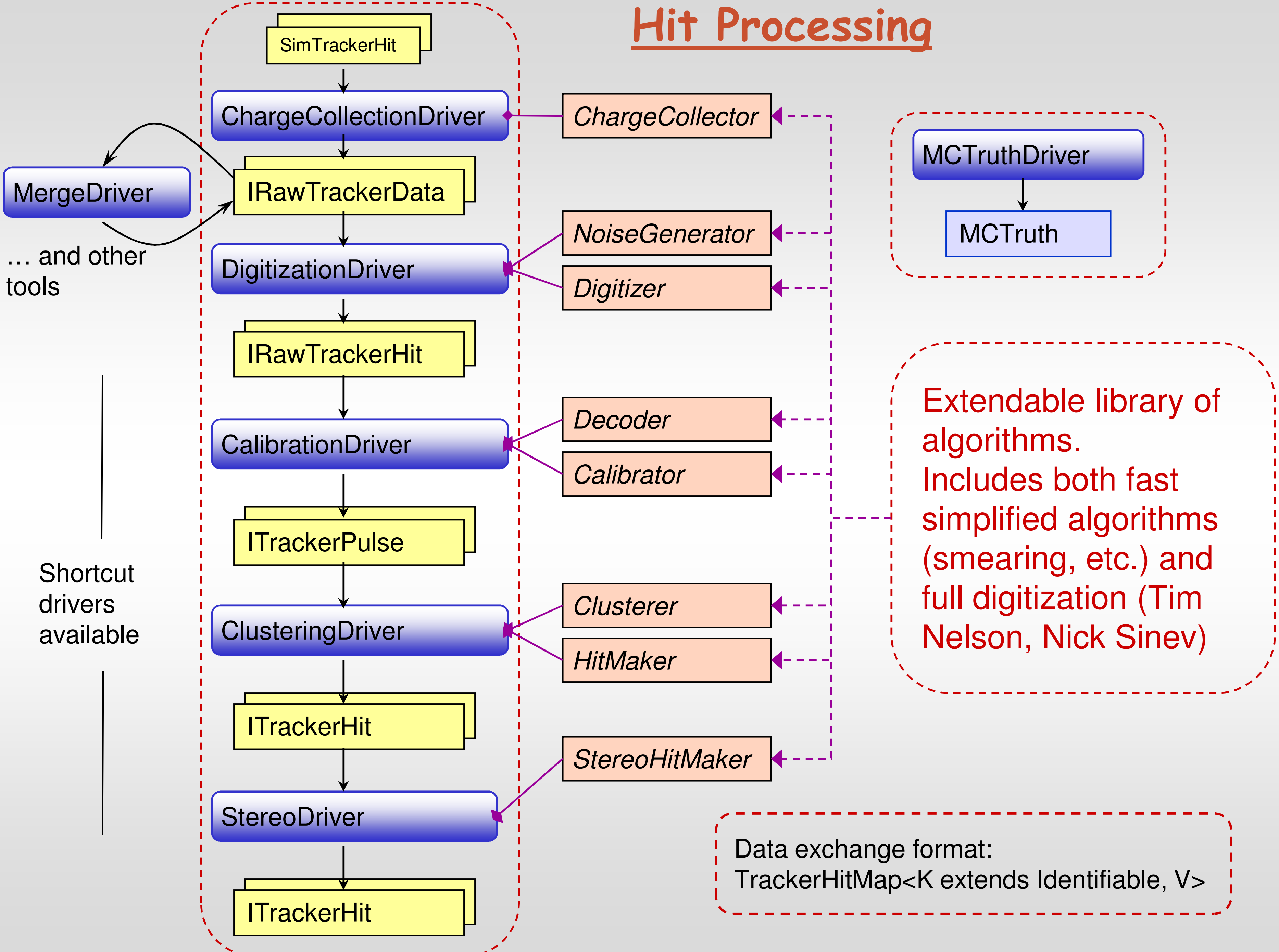
double getdEdx()
double getTime()

Hep3Vector getLocalPosition()
Hep3Vecror getLocalErrors()

Hep3Vector getPosition(IRefFrame referenceFrame)
SymmetricMatrix getCovMatrix(IRefFrame referenceFrame)

Sensor getSensor()

Hit Processing



org.lcsim.contrib.onoprien.data.mctruth

Object model interfaces do not have any Monte Carlo specific methods.

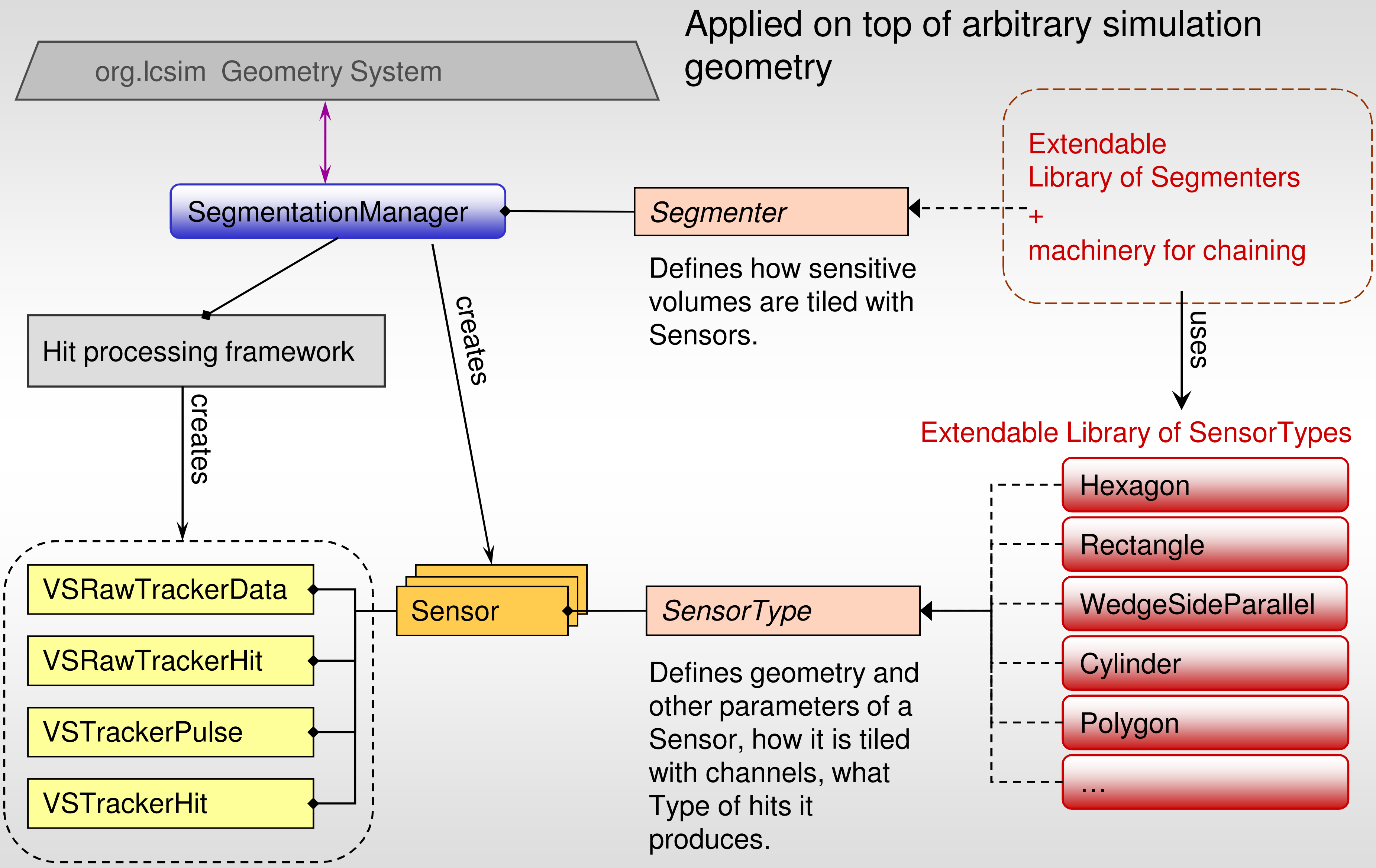
MC information bookkeeping is done automatically by hit processing framework, and accessible through MCTruth object.

RecType:

- TRACKER_CLUSTER
- TRACKER_HIT
- TRACK_SEED
- TRACK
- CAL_HIT
- CLUSTER
- PARTICLE

Method Summary	
WeightedList<MCParticle>	**findMCParticles**(RecType type, Object reconstructedObject) Compiles and returns a weighted list of MCParticles that contributed to the specified reconstructed object.
<T> WeightedList<T>	**get**(RecType type, MCParticle mcParticle) Returns a weighted list of reconstructed objects of the specified type to which the specified MCParticle has contributed.
WeightedList<MCParticle>	**getMCParticles**(RecType type, Object reconstructedObject) Returns a weighted list of MCParticles that contributed to the specified reconstructed object.
<T> WeightedTable<MCParticle,T>	**getMCParticleTable**(RecType type, Collection<? extends T> collection) Returns a relational table between MCParticles and reconstructed objects of the requested type from the specified collection.
<T> WeightedTable<MCParticle,T>	**getMCParticleTable**(RecType type, String... collectionNames) Returns a relational table between MCParticles and reconstructed objects of the requested type from the specified collections in the event record.
List<SimTrackerHit>	**getSimTrackerHits**(IRawTrackerHit hit) Returns a list of SimTrackerHits that contributed to the specified IRawTrackerHit.
List<SimTrackerHit>	**getSimTrackerHits**(ITrackerHit hit) Returns a list of SimTrackerHits that contributed to the specified ITrackerHit.
List<SimTrackerHit>	**getSimTrackerHits**(ITrackerPulse pulse) Returns a list of SimTrackerHits that contributed to the specified ITrackerPulse.
List<SimTrackerHit>	**getSimTrackerHits**(MCParticle mcParticle) Returns a list of SimTrackerHits produced by the specified MCParticle, sorted by time.
<T> void	**setMCParticleTable**(RecType type, WeightedTable<MCParticle,T> map) Sets the default relational table between MCParticles and reconstructed objects of the specified type.

Virtual segmentation



org.lcsim.contrib.onoprien.util.job

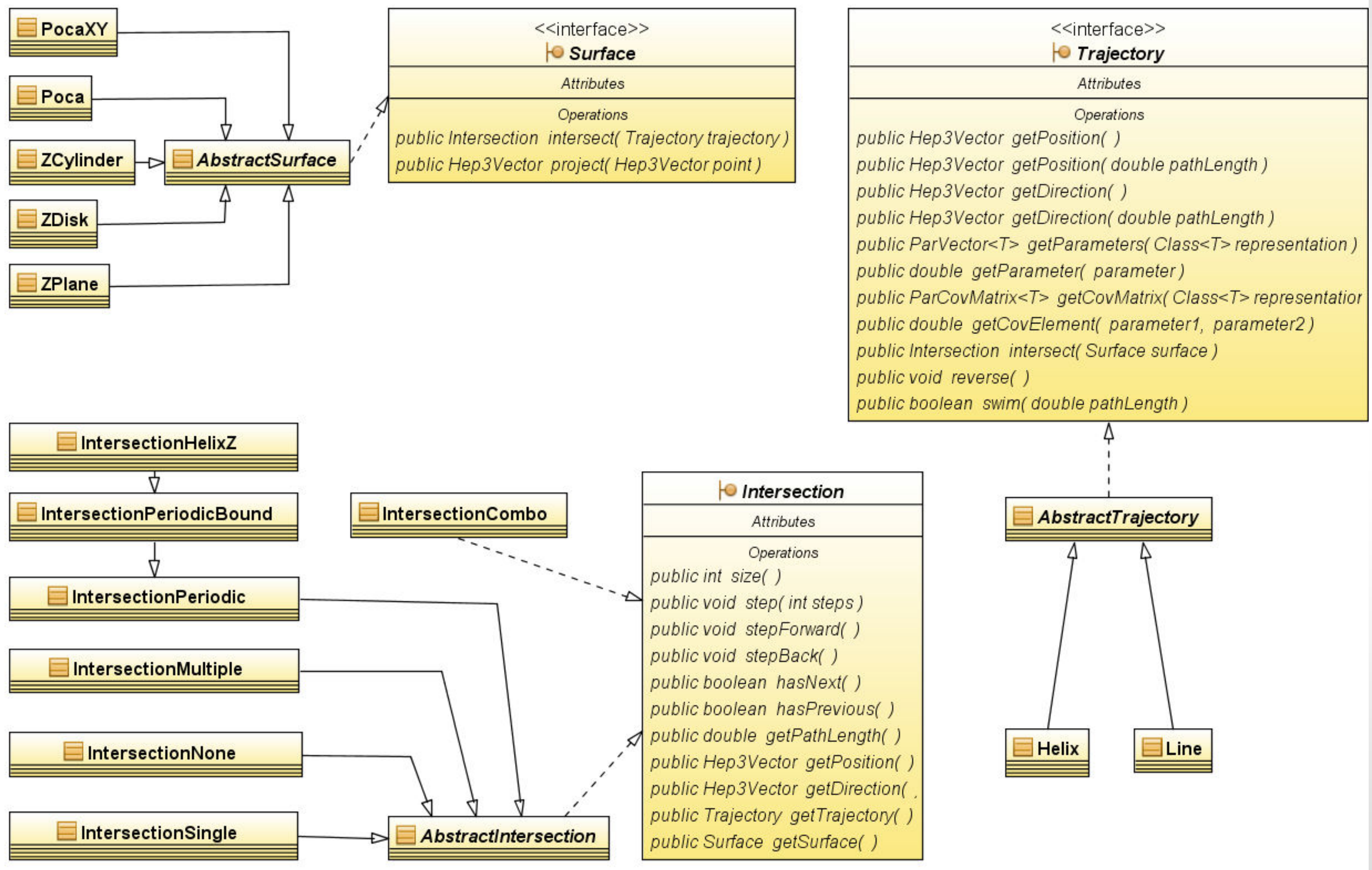
Driver :

- Extended version of Driver

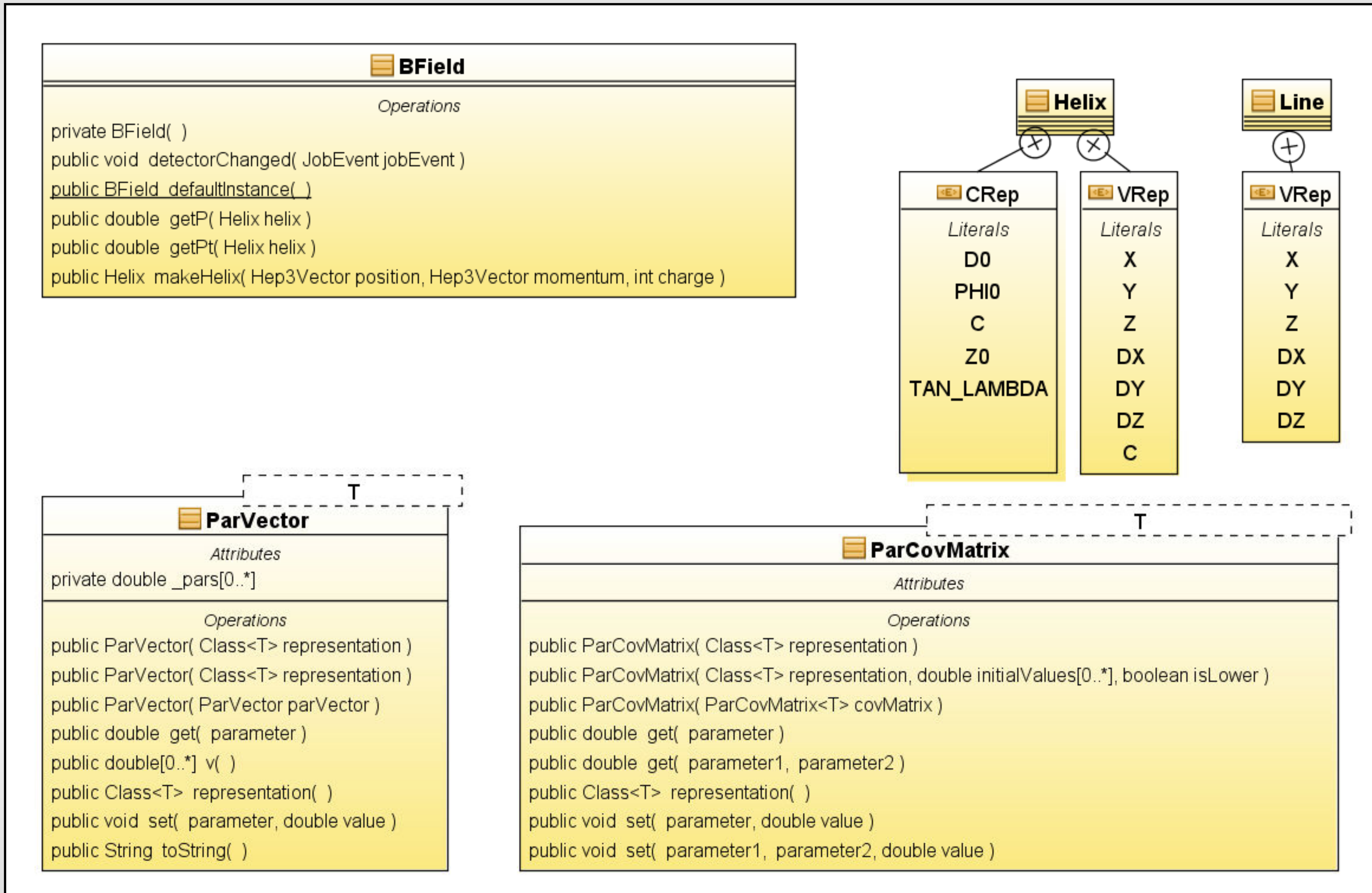
JobManager :

- Accepts listener registration and dispatch events that trigger geometry dependent initialization in client classes. Listeners can be registered along with dependencies among them - JobManager guaranties that listeners specified as "prerequisites" for some other listener will receive the event first.
- Allows registration and retrieval of singleton objects of any type.
- Provides access to the default AIDA object that can be used for histogramming, plotting, etc.
- Accepts HepRepCollectionConverter registration for visualizing Lists of objects in the event record using Wired event display.
- Can print message and/or save AIDA tree every specified number of events.

org.lcsim.contrib.onoprien.util.swim



org.lcsim.contrib.onoprien.util.swim



org.lcsim.contrib.onoprien.performance

Cheaters and performance analysis tools.

See:
Description

Interface Summary	
IDefinition	Interface to be implemented by classes that define "findable", "reconstructed" MCParticle, and "fake" reconstructed object - used by cheaters and performance analysis classes.

Class Summary	
AnalysisDriver	Base driver for testing event reconstruction algorithms performance.
CheatClusteringDriver	Cheater that attaches hits and tracks to reconstructed particles.
CheatRecoParticleCreator	Driver that creates a tree of reconstructed particles and vertices based on Monte Carlo truth information, packs it into `RecParticleList`, and puts it into the event record.
CheatRecoParticleDriver	Driver that uses MC truth info to produce a list of reconstructed particles.
CheatTrackFinderDriver	Driver that uses MC truth info to produce a list of tracks.
CheatVertexFinderDriver	Driver that uses MC truth info to produce a list of reconstructable vertices in the event, and to associate them with previously reconstructed tracks.
Definition	Configurable implementation of `IDefinition`.