EDM Status

MDST EDM

- Has at the moment the highest priority, because it should be finished this year.

mdst Classes

StoreArrays

- Track
- TrackFitResult
- PIDLikelihood
- ECLShower
- ECLGamma
- ECLPi0
- EKLMK0L
- MuidLikelihood
- MCParticle

Relations

- Trackd → PIDLikelihoods
- ECLGammas → ECLShowers
- ECLPi0s → ECLGammas
- MCParticles → Tracks
- ECLShowers → MCParticles

Thomas @Computing Workshop

MDST EDM

- Has at the moment the highest priority, because it should be finished this year.

mdst Classes

StoreArrays

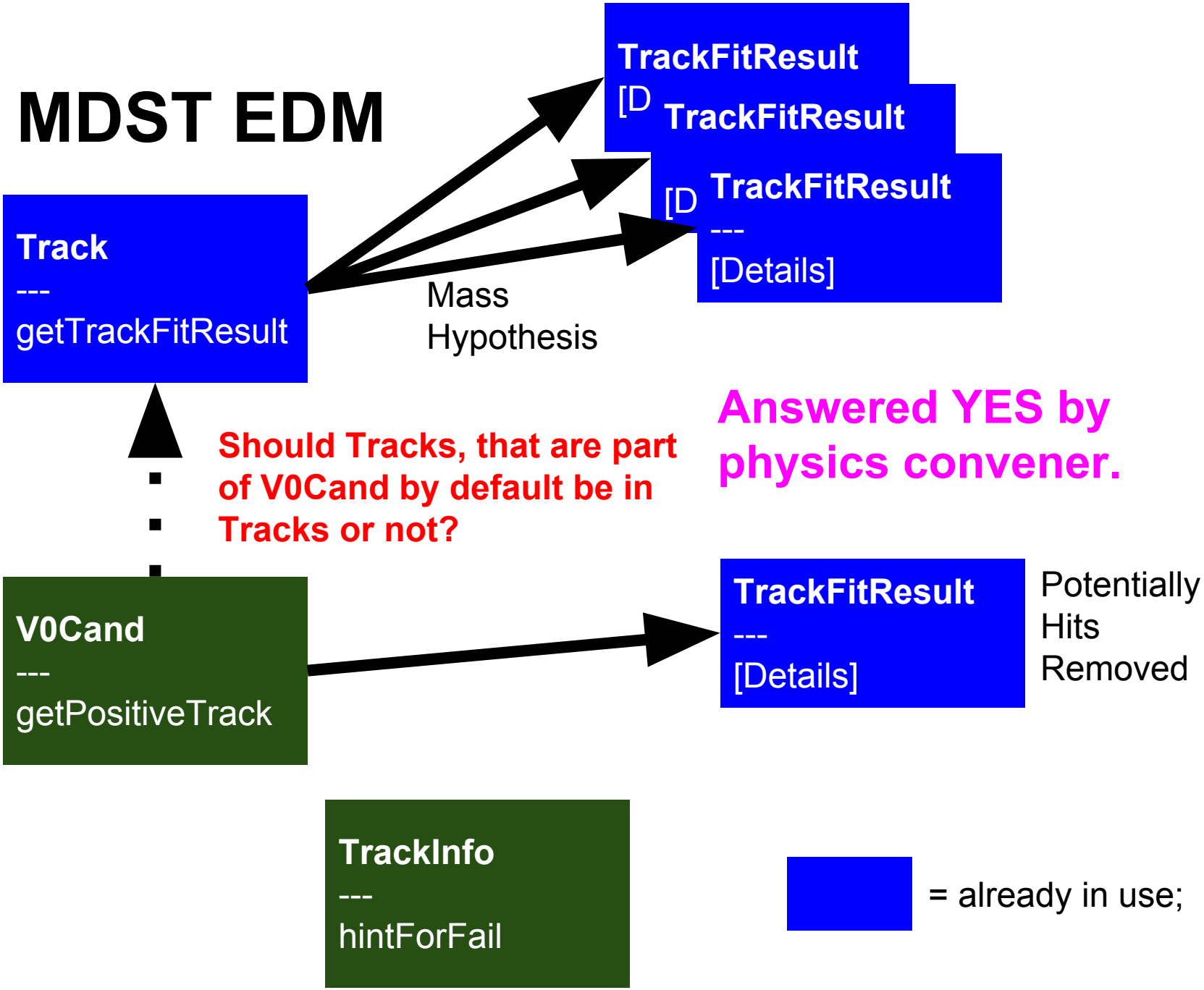
- Track
- TrackFitResult
- PIDLikelihood
- ECLShower
- ECLGamma
- ECLPi0
- EKLMK0L
- MuidLikelihood
- MCParticle

Relations

- Trackd → PIDLikelihoods
- ECLGammas → ECLShowers
- ECLPi0s → ECLGammas
- MCParticles → Tracks
- ECLShowers → MCParticles

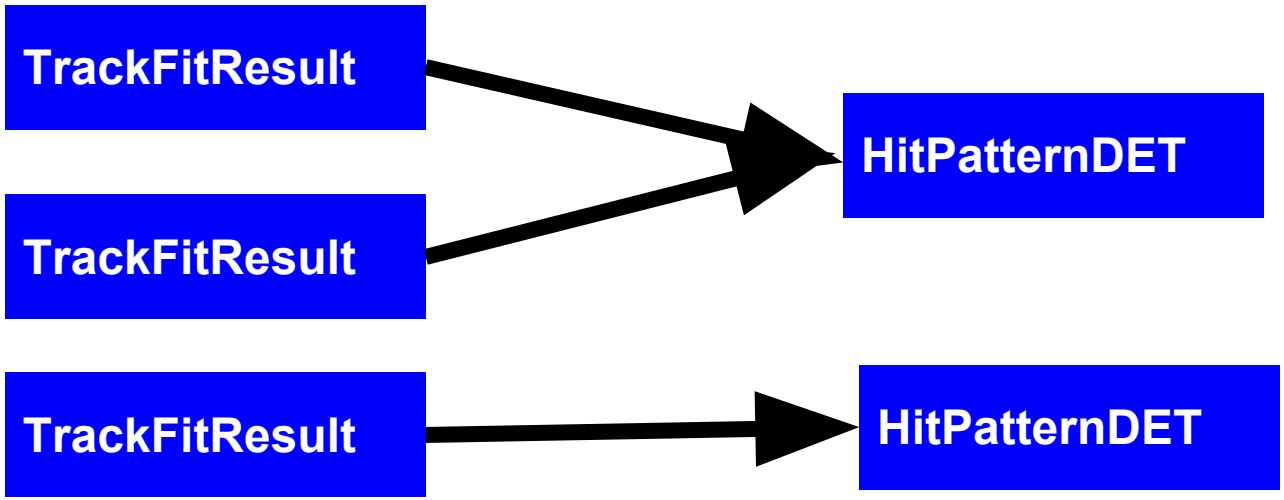
Thomas @Computing Workshop
+ V0 + HitPatternCDC +HitPatternVXD

MDST EDM


TrackFitResult
[D
TrackFitResult
[D
TrackFitResult

[Details]
Track
---
getTrackFitResult
Mass
Hypothesis
Should Tracks, that are part of V0Cand by default be in Tracks or not?
Answered YES by physics convener.
V0Cand
---
getPositiveTrack
TrackFitResult

[Details]
Potentially Hits Removed
TrackInfo
---
hintForFail
= already in use;



OR

Will use this option at least temporarily [until we get complaints about size of MDST]

TrackFitResult

HitPatternDET

Depending on frequency with which alternative Mass Hypothesis Fits produce different hit patterns.

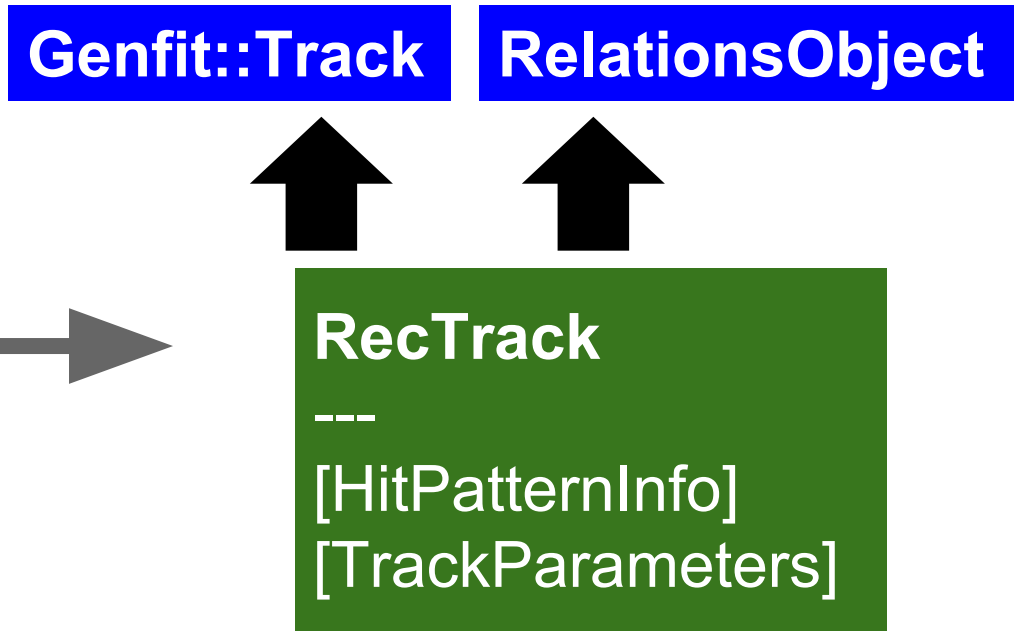
OR

Track

HitPatternDET

Reconstruction EDM

- Object inheriting from Genfit::Track will be at the center of the connection to the other reconstruction parts.



PID etc. should use this class as soon as we have it.

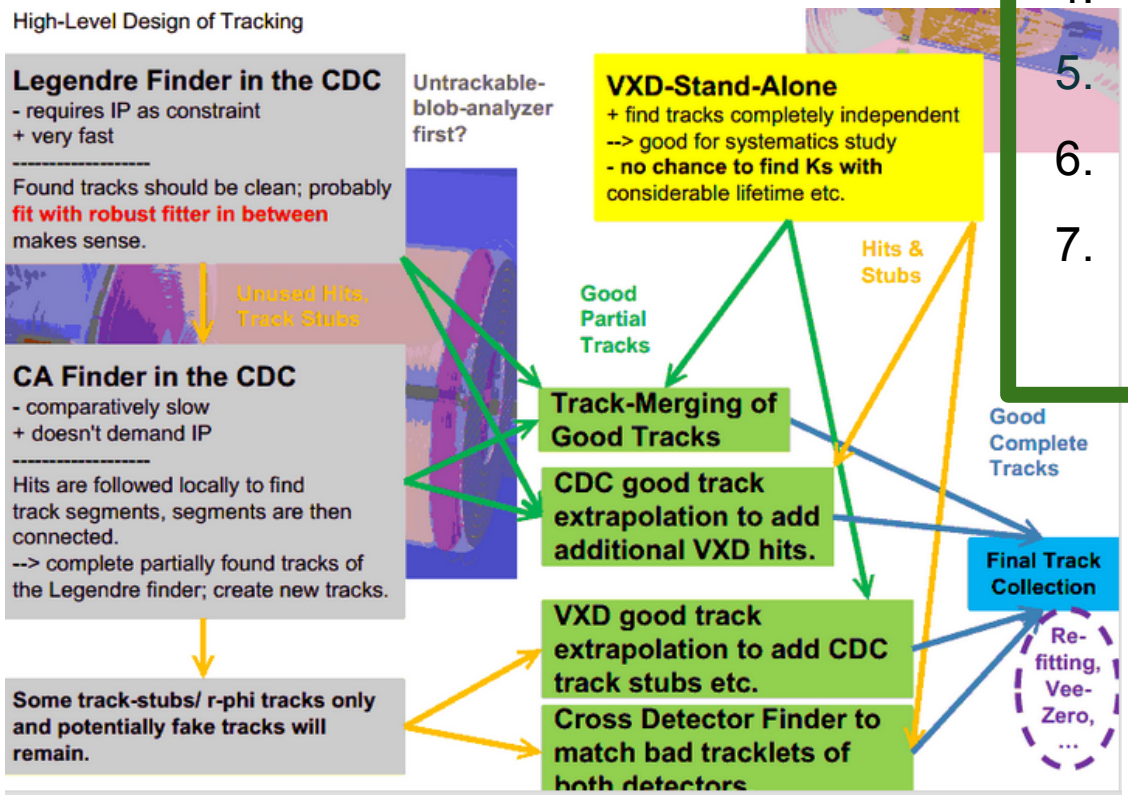
TrackingInfo

[List of Track Indices with CDC only tracks etc.]

Not stored, because it can be rebuild from the RecTrack information.

Possible Final Tracking Configuration

1. LegendreFinder for high-momentum tracks;
2. Extrapolation to VXD;
3. VXD Stand-Alone stuff;
4. Extrapolation into the CDC;
5. LocalFinder;
6. Extrapolation to the VXD;
7. Cross-Detector Searches + “Calo Clean”;



Maximising the Use of the RecTrack

- Would be nice to use RecTrack (Genfit::Track part) as well for the extrapolation to the outer detectors.
  - This was previously discussed and Leo agreed to check e.g. timing of Geant4e vs. RKTrackRep extrapolation.
  - However, tracking group has too few people at the moment to attack this.
    → another competent group would be welcome.

Naming Issues

- We have VXDTF, CDCLocalTrackFinder,...
  - perhaps it would be nicer to have a common scheme like
    - VXDCellularAutomatonFinder
    - CDCLegendreFinder

      e.g. something indicating the method + Finder

      [I think “Track” is kind of the default in our package, and therefore doesn’t need to be named explicitly].

Several Modules in one folder?

https://belle2.cc.kek.jp/browse/viewvc.cgi/svn/trunk/software/tracking/

- e.g. cdcPatternReco, pxdDataReduction
  - I think this is OK, modules can be found via doxygen;
  - would like to have consistent naming as well for folders, e.g. not mctrackfinder, but mcTrackFinder… or no capitalisation at all, like we do for dataobjects.

No need for subname spaces in libraries

https://belle2.cc.kek.jp/browse/viewvc.cgi/svn/trunk/software/tracking/cdcLocalTracking/creators/include/GFTrackCandCreator.h?revision=7708&view=markup

- If only one module or collection of modules uses a library, better put the code directly to the module.
  - Makes the library smaller -> Less memory consumption, if the modules is not loaded.

```
#include "../include/GFTrackCandCreator.h"
#include <boost/foreach.hpp>

#include <framework/logging/Logger.h>
#include <framework/gearbox/Const.h>

using namespace std;
using namespace Belle2;
using namespace CDCLocalTracking;



GFTrackCandCreator::GFTrackCandCreator() {;}
```

Dataobjects in libraries

- Generally ROOTified objects should go into the dataobjects folder;

- dataobjects should be independent of main package library.

  The second point is the more important one. If you can’t make the ROOTified object independent of the main library, you can store it there.

Try Subgrouping

- Group your parameters to objects with a name indicating something about how they belong together.
  https://belle2.cc.kek.jp/browse/viewvc.cgi/svn/trunk/software/tracking/modules/VXDTF/include/VXDTFModule.h?revision=7747&view=markup

```
boostNsec totalTime;  /**< time consu
TimeInfo sectionConsumption; /**< one
int evtNumber; /**< number of current
int numPXDCluster; /**< number of pxd
int numSVDCluster; /**< number of svd
//
int numSVDHits; /**< number of possib
int segFinderActivated; /**< number o
int segFinderDiscarded; /**< number o
int nbFinderActivated; /**< number of
int nbFinderDiscarded;  /**< number o
int tccApprovedTCs;  /**< number of t
int numTCsAfterTCC; /**< number of tc
int numTCsAfterTCCfilter; /**< number
int numTCsKilledByCleanOverlap; /**<
int numTCsfinal; /**< number of tcs a
```

Should be possible as well for module paramters directly; (besides this shouldn't have happened in the beginRun function)

```
addParam("sigmaSystU", m_sigmaSystU, " sy
addParam("sigmaSystV", m_sigmaSystV, " sy
addParam("numSigmaTotU", m_numSigmaTotU,
addParam("numSigmaTotV", m_numSigmaTotV,
addParam("maxWidthU", m_maxWidthU, " uppe
addParam("maxWidthV", m_maxWidthV, " uppe
```

```
void PXDDataReductionModule::beginRun()
{

  m_ROIinfo.sigmaSystU = m_sigmaSystU;
  m_ROIinfo.sigmaSystV = m_sigmaSystV;
  m_ROIinfo.numSigmaTotU = m_numSigmaTotU;
  m_ROIinfo.numSigmaTotV = m_numSigmaTotV;
  m_ROIinfo.maxWidthU = m_maxWidthU;
  m_ROIinfo.maxWidthV = m_maxWidthV;
```

No need to specify each collection with an input steering parameter, if you likely never change it.

```
addParam("trackCandCollName", m_gfTrackCandsColName, " name of th

e of the input collection of track candidates", std::string(""));
```

Using the wording collection is outdated. It once was used for StoreArrays and StoreObjects alike, but didn't really fly.
So it is probably better to use gfTrackCandsName in this case. (Being as well consistent between the c++ and the steering name)

```
void fillInterceptList(StoreArray<PXDIntercept>* listToBeFilled,
                       const StoreArray<genfit::TrackCand>& trackCandList,
                       RelationArray* gfTrackCandToPXDIntercepts);
```

Try to use (const) references;
use just “s” appendix for StoreArrays -> gfTrackCands, PXDIntercepts [ToBeFilled]

```
/**
 * Set the nuber of iterations of the Kalman Filter to numIterKalmanFilter
 */
void setNumIterKalmanFilter(int numIterKalmanFilter) {
  m_kalmanFilter.setMinIterations(numIterKalmanFilter);
};
```

If you need just one such array ever, better stick to default and avoid giving the StoreArray as argument.

```
private:

genfit::KalmanFitter m_kalmanFilter; /**< kalman filter object to fit the track */

ROIGeometry m_theROIGeometry; /**< the geometry of the Region Of Interest */

ROIinfo m_theROIinfo; /**< the ROI info structure */
```

Avoid meaningless name addendums like “the”

Be aware

- StoreArrays can be used like STL containers e.g. with for loops. This should work with StoreArrays as well.

```
std::vector<int> vec ;

for( int i: vec) // iteratively set i to every element in vec
{                // works on any class that has begin() and end()
    std::cout << i << std::endl ; // readonly access
}
```

- We have now C++11 and most useful features were shown at the Computing Workshop and the B2GM. Please have look there, especially as well const expressions.
  http://kds.kek.jp/conferenceTimeTable.py?confId=13846#20131111.detailed

Modules need clean up?

VXDTF/	7747 (4 days ago)
VXDTFHelperTools/	7747 (4 days ago)
cdcLocalTracking/	7768 (3 days ago)
cdcMCmatching/	7708 (6 days ago)
cdcPatternReco/	7708 (6 days ago)
ext/	7708 (6 days ago)
extrapolateToVXD/	7708 (6 days ago)
genfitVisModule/	7721 (5 days ago)
genfitter/	7732 (5 days ago)
mcTrackCandCombiner/	7708 (6 days ago)
mcTrackMatcher/	7793 (2 days ago)
mctrackfinder/	7728 (5 days ago)
muid/	7708 (6 days ago)

pxdDataReduction/	7708 (6 days ago) by
siCDCTrackMerger/	7708 (6 days ago) by
simplebackground/	7708 (6 days ago) by
standardTrackingPerfo...	7708 (6 days ago) by
trackFitChecker/	7708 (6 days ago) by
trackingEvaluation/	7768 (3 days ago) by
trasan/	7708 (6 days ago) by
trueHitTester/	6523 (3 months ago)
vertexer/	7708 (6 days ago) by
vxdSimpleClusterizer/	5438 (7 months ago)

Should WireID NOT inherit from TObject?

- It is similar as VXDId, which doesn't inherit from TObject.
- It is not serialized anyhow.
- But perhaps people would like to use pyroot together with it.
  - → MockRoot stuff from Oliver to be used ?!

- I don't think the CDC group actually uses this object.