



HEPHY

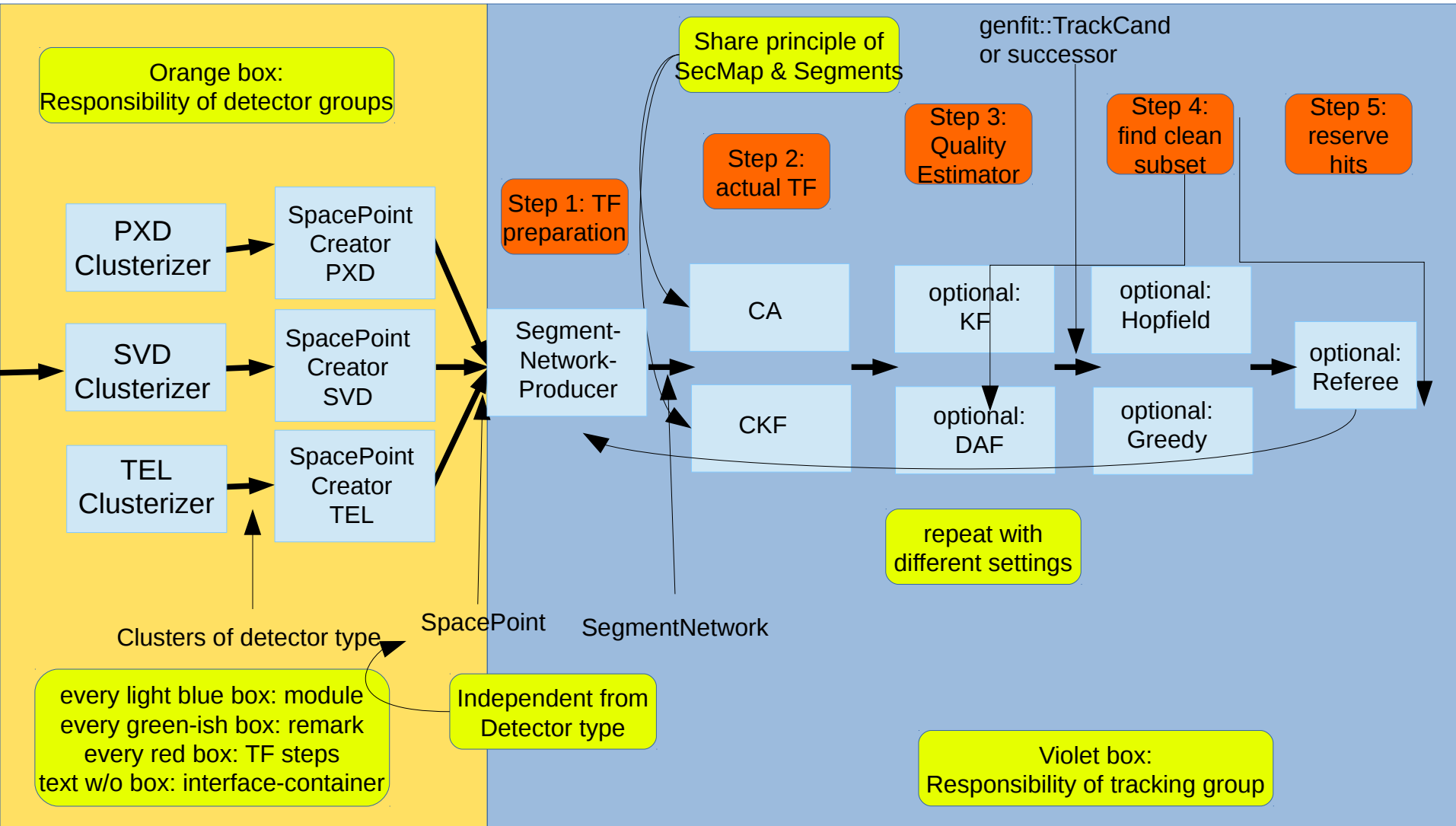
Institute of High Energy Physics

Current state of redesign

Jakob Lettenbichler

biweekly tracking meeting

November 28th, 2014

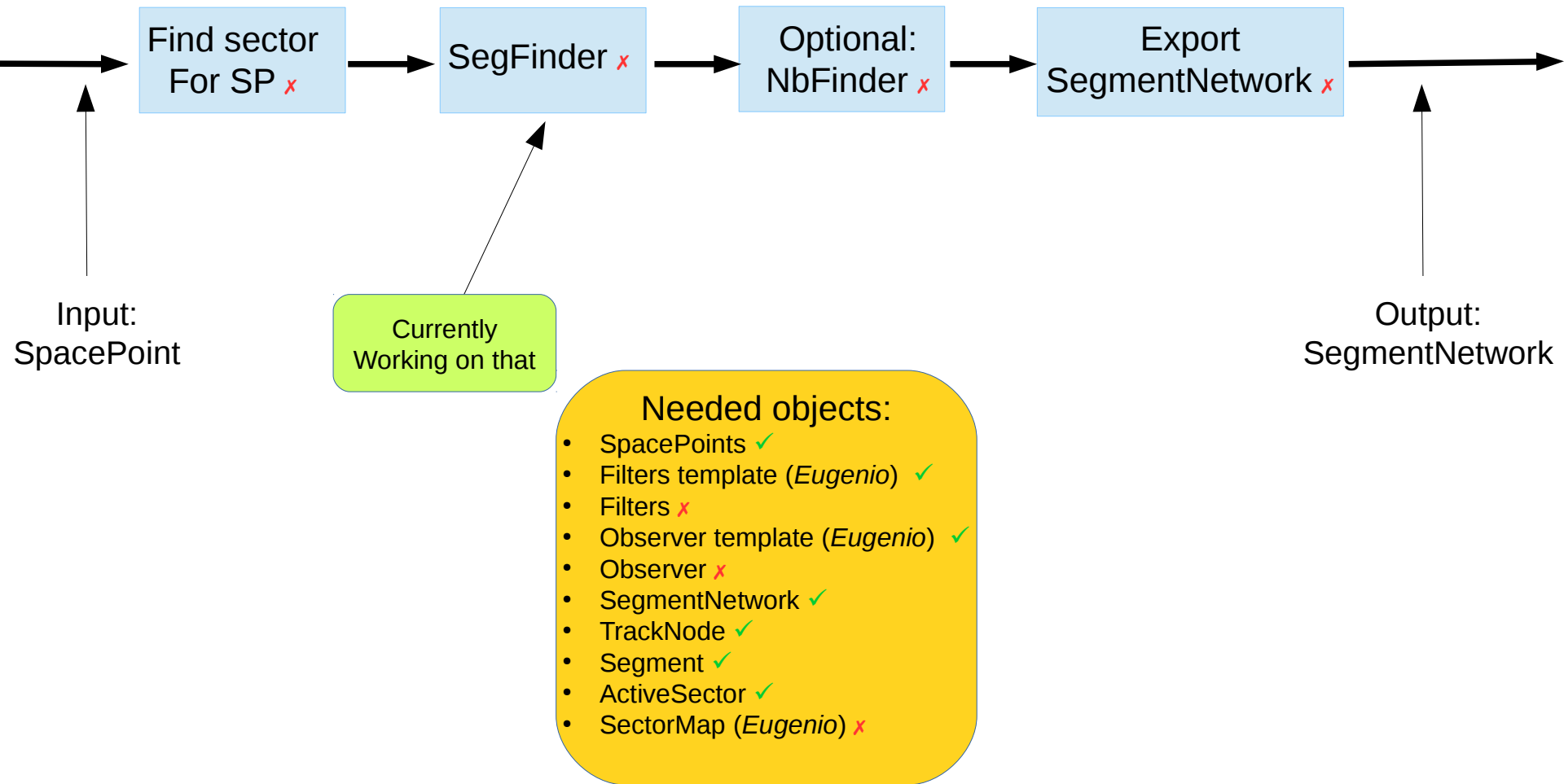


Current state of the redesign (estimated work done so far)

Modules:

- *XYZSpacePointCreator*: working, tests are fine (~95%)
- *SegmentNetworkProducer*: under construction (~25%)
- CA only old code exists still (10%)
- CKF only preliminary design work done yet (2%)
- *Quality estimator – KF*: only old code still exists (10%)
- *Quality estimator – CF*: only old code still exists (10%)
- *Quality estimator – DAF*: only preliminary design work done yet (1%)
- *Clean subset – Hopfield*: only old code still exists (10%)
- *Clean subset – Greedy*: only old code still exists (10%)
- *Reserve hits*: only old code still exists (10%)

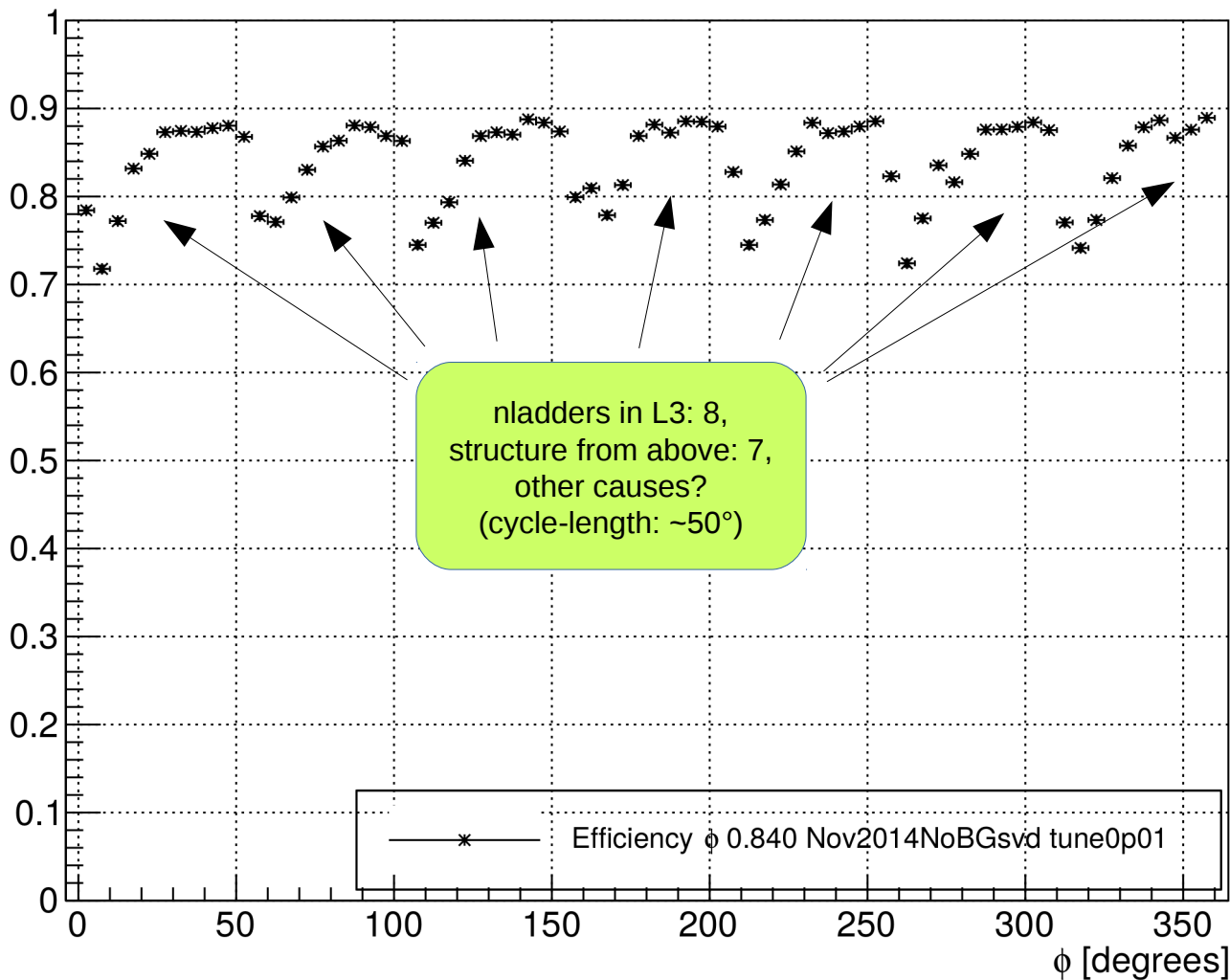
Sketch of the Segment network producer module:



Current issues/todo (open for discussion):

- Howto treat cuts:
`m_inf < x && x < m_sup;` (excludes cuts)
vs
`return !(m_sup < x || x < m_inf);` (includes cuts, advantage: use single track for secMapGen and reco works out of the box)
- Howto deal with `isnan(result) == true` and `isinf(result) == true`? Current approach: `if true result = 0;`
- Longterm replacement for *slopeRZ*:
slopeRZ: `result = atan(sqrt(pow(a.x-b.x),2) + pow(a.y-b.y),2)) / (a.z-b.z)`
vs
slopeZabsXY: `result = (a.z-b.z) / (abs(a.x-b.x) + abs(a.y-b.y))`, where `isinf(result) == true` only if `a.x==b.y && a.y==b.y`
- Howto deal with rounding errors, case *slopeRZ*, where rounding issue produces `EXPECT_TRUE(false)`; in a test
- Needed observers
 - count number of times used (already provided by Eugenio)
 - catch result and print to screen or root file
 - count number of times result was accepted/rejected (+ store to root file)
 - count number of times good/bad combination was accepted/rejected (uses MCInfo, didn't have that before)
 - count number of times for `isinf/isnan true`
- Implementing missing *segFinder*-filter: *normedDistance3D* (right after finishing *slopeRZ*)

And now something completely different: Efficiency of ϕ



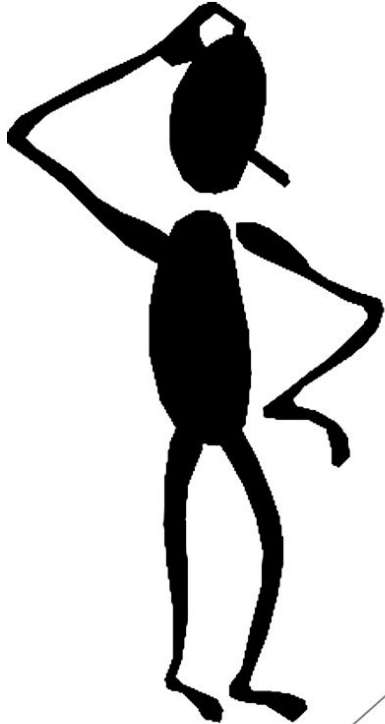
Beware:
Phi taken from
momentumVector
Of the mcParticle at
Innermost hit of mcTC

Settings:
Sample size: 200000 events
SVD-only
EvtGen, new SVD-geo
No background added

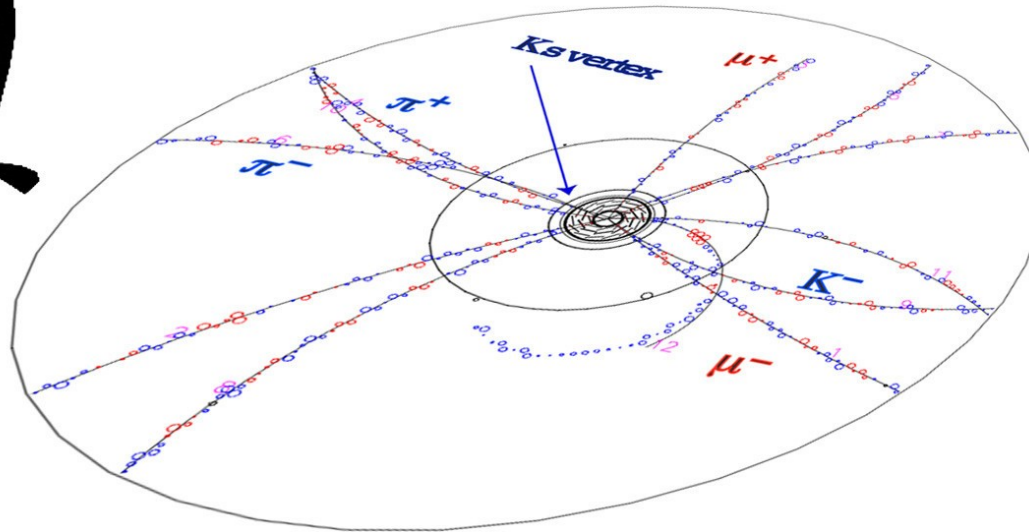
Outlook/ToDo until F2F Prague (Jan 2015)

(all points essential to fulfill milestone-requirements)

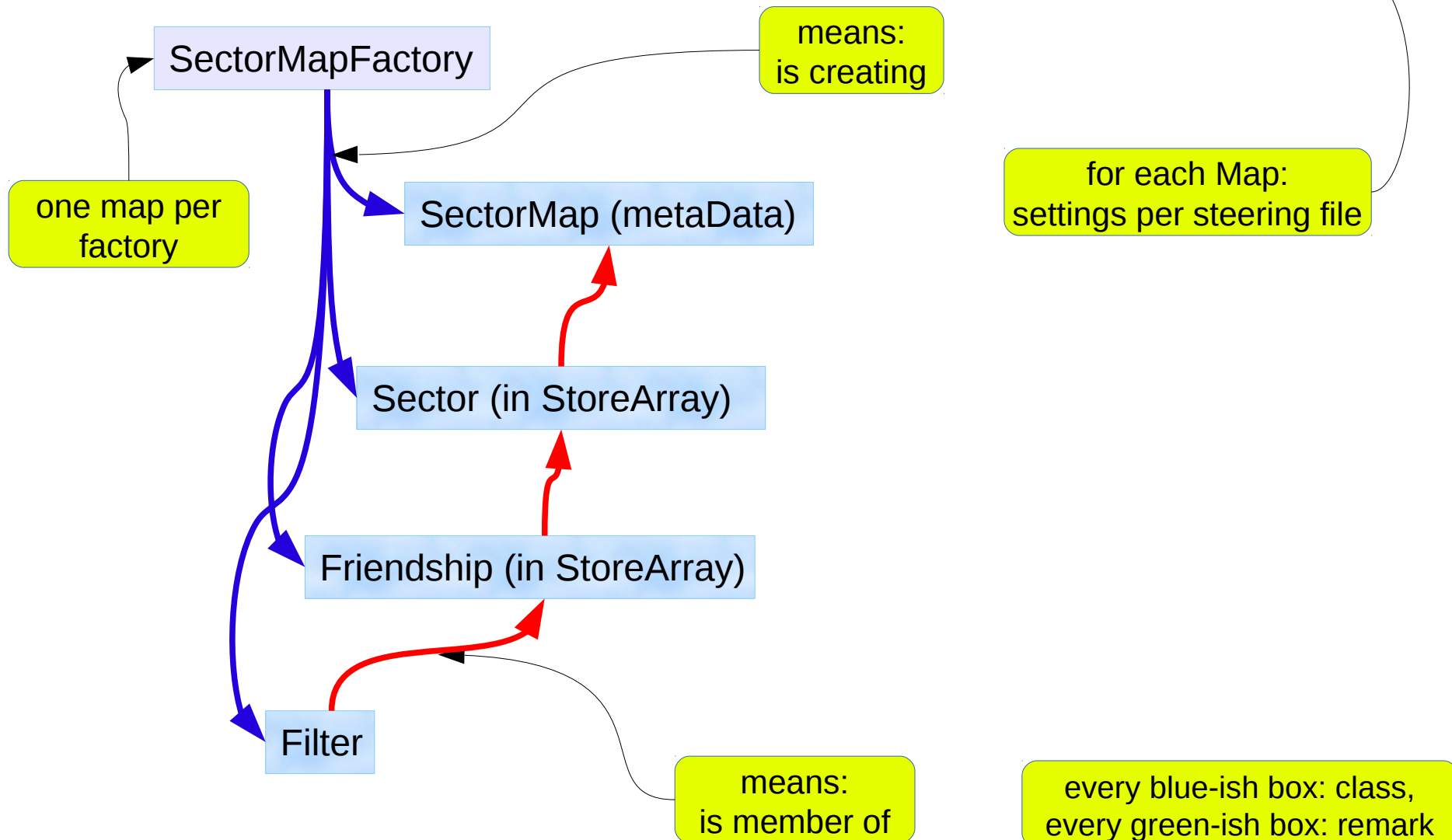
- Finish *segFinder* filters
- Provide observers needed
- Replace old segfinder and prove that old and shiny new design provide the same results
- After guaranteed working of *genfit::TrackCand* ↔ *SpacePointTrackCand* conversion: replace old way to get hits in the VXDTF
- After that, test filling of *segmentNetwork* using the new *segFinder* and *SpacePoints*
- Then, port *nbFinder*-filters and add to *SegmentNetworkProducer*
- Convert old CA-code to new CA-module (w/o TC-creation yet)



Thank you!



SectorMapCreatorModule (does that during beginRun)

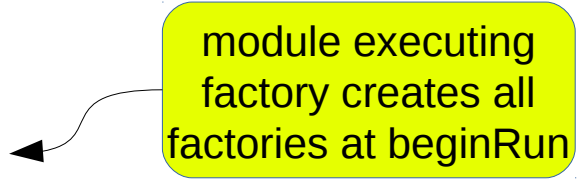


Detailed sketch for the new sectorMap-approach

SectorMapFactory

- one factory per setup
- creates own StoreArray for its Sectors
- another StoreArray for SectorFriendship
- links them by relations and pointers(to storeArray-entries)
- creates storeObjPtr for metaData („SectorMap“)

module executing
factory creates all
factories at beginRun



SectorMap :: storeObjPtr

- carries metadata like name of StoreArrays
- sorted container of <Sector*> for direct access
- defines origin/secMapCenter and sorting type

Sector :: RelationArray

- only static info like SectorFriendship
- form directed graph with other sectors (direction by secID or distance2Origin)
- container of <FriendRelations*> for direct access
- carries ActiveSector* (reset every event)
- carries segmentMaker called by ActivatedSector

Detailed sketch for the new sectorMap-approach II

SectorFriendship :: RelationArray

- one Friendship allowed for each compatible combination with current sector
- combination can contain any number of sectors in chain (useful lengths: 1-3)
- Carries only the filters allowed for that combi & secMap

Filter :: RelationArray

- can be a filter for any number of hits (currently there are 2-X-hitfilters)
- Filter applied only for current Friendship → cutoffs only for local case
- creates/updates compatibilityTable for each possible hit-combi
- following filters only execute their stuff on combis which are still alive

ActivatedSector (:: RelationArray?)

- 1:1 relation to a sector of current sectorMap
- created once per event (lightweight, maybe not inheriting anything)
- container of <Hits*> for direct access
- hits are passed to segmentMaker of Sector
- stores segments