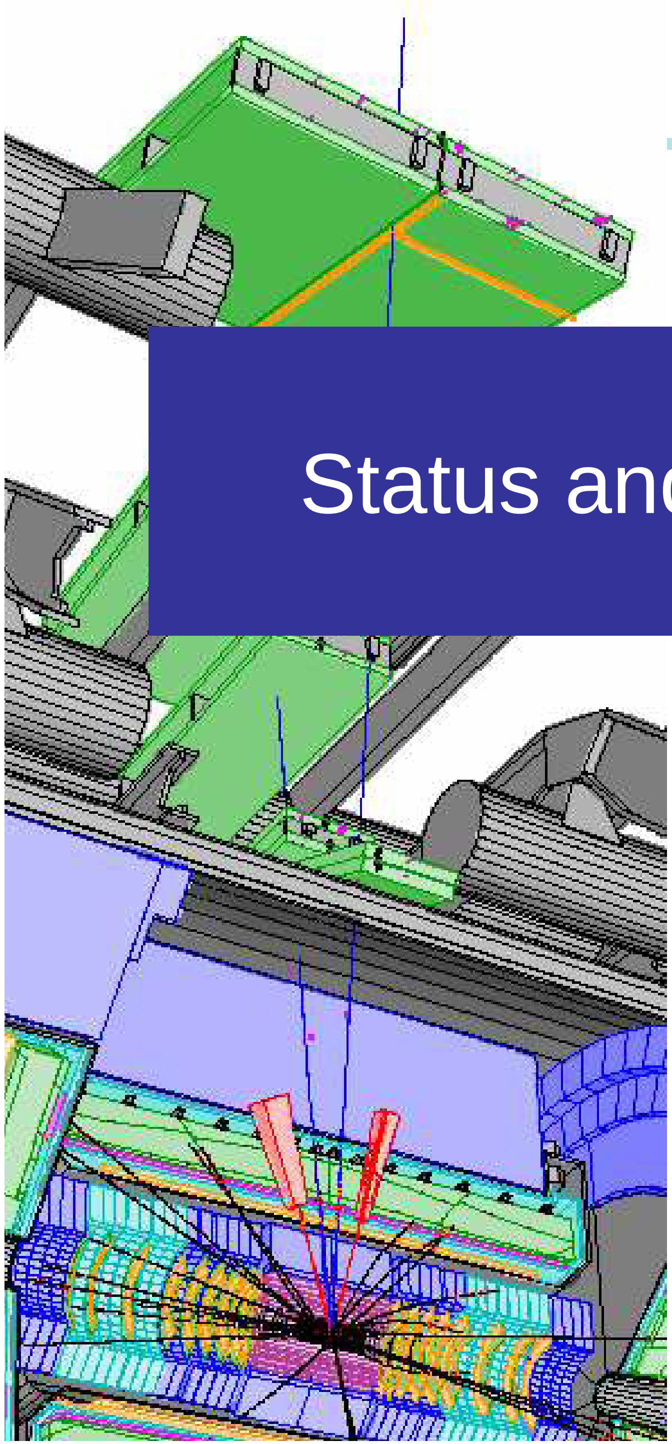


Status and outlook of HLT software

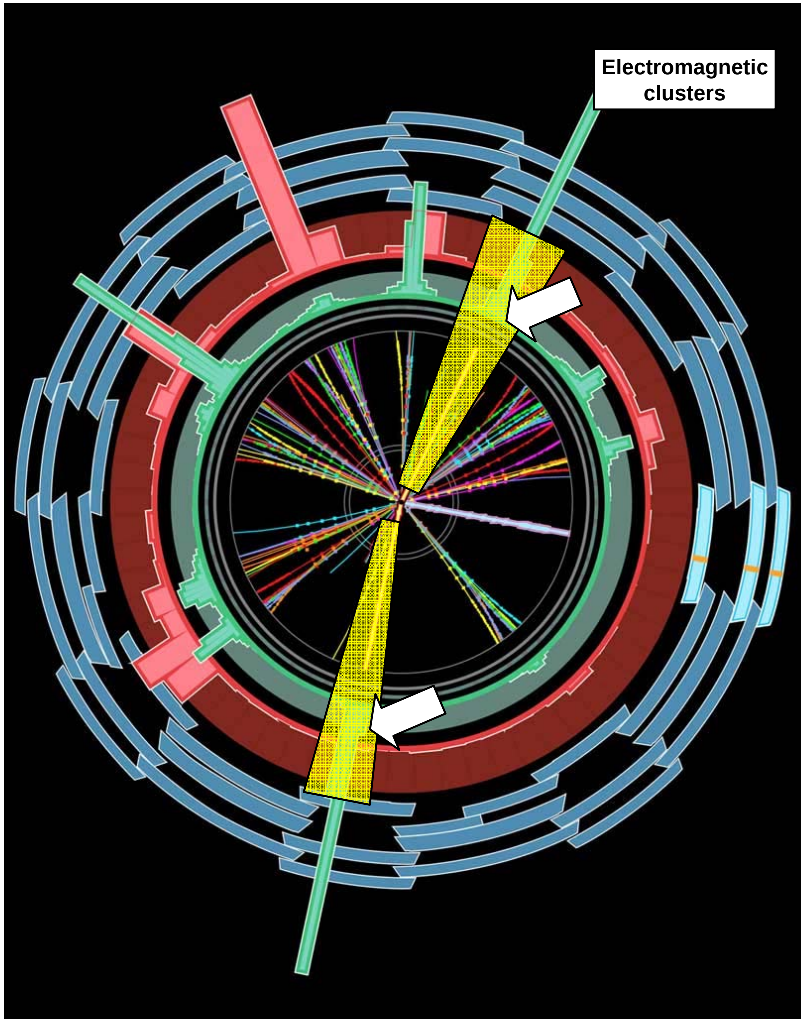
- High Level Trigger algorithms for e/γ
- How to analyse HLT data?
- Trigger information in ESD/AOD

Ricardo Gonçalo (RHUL) reporting
on work by several people

- I’ll try to give an overview of the HLT software
- Will not discuss the steering, RegionSelector or configuration much
- Will sometimes focus on level 2 classes, as these are probably not as well known in this audience
- Will try to give a good idea of the current status and of when more should become available

Introduction

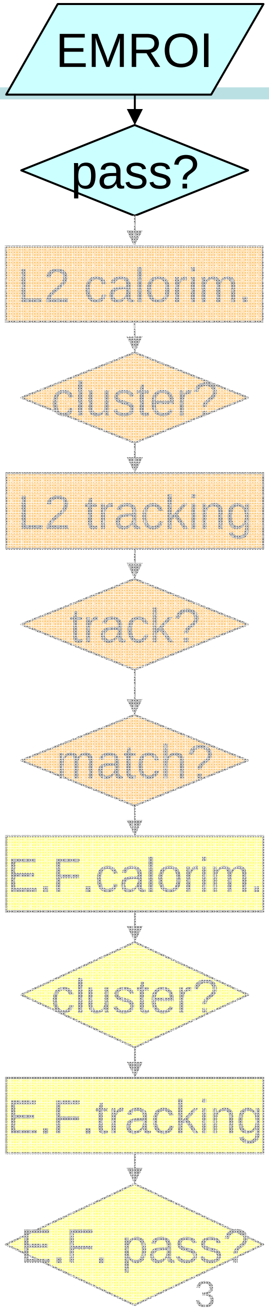
Chain can be abandoned at each step



Level1 **Region of Interest** is found with **coarse granularity** and position in EM calorimeter is passed to Level 2

Level 2 seeded by Level 1
Fast reconstruction algorithms
Full granularity
Reconstruction **within RoI**

Ev.Filter seeded by Level 2
Offline reconstruction algorithms
Refined alignment and calibration



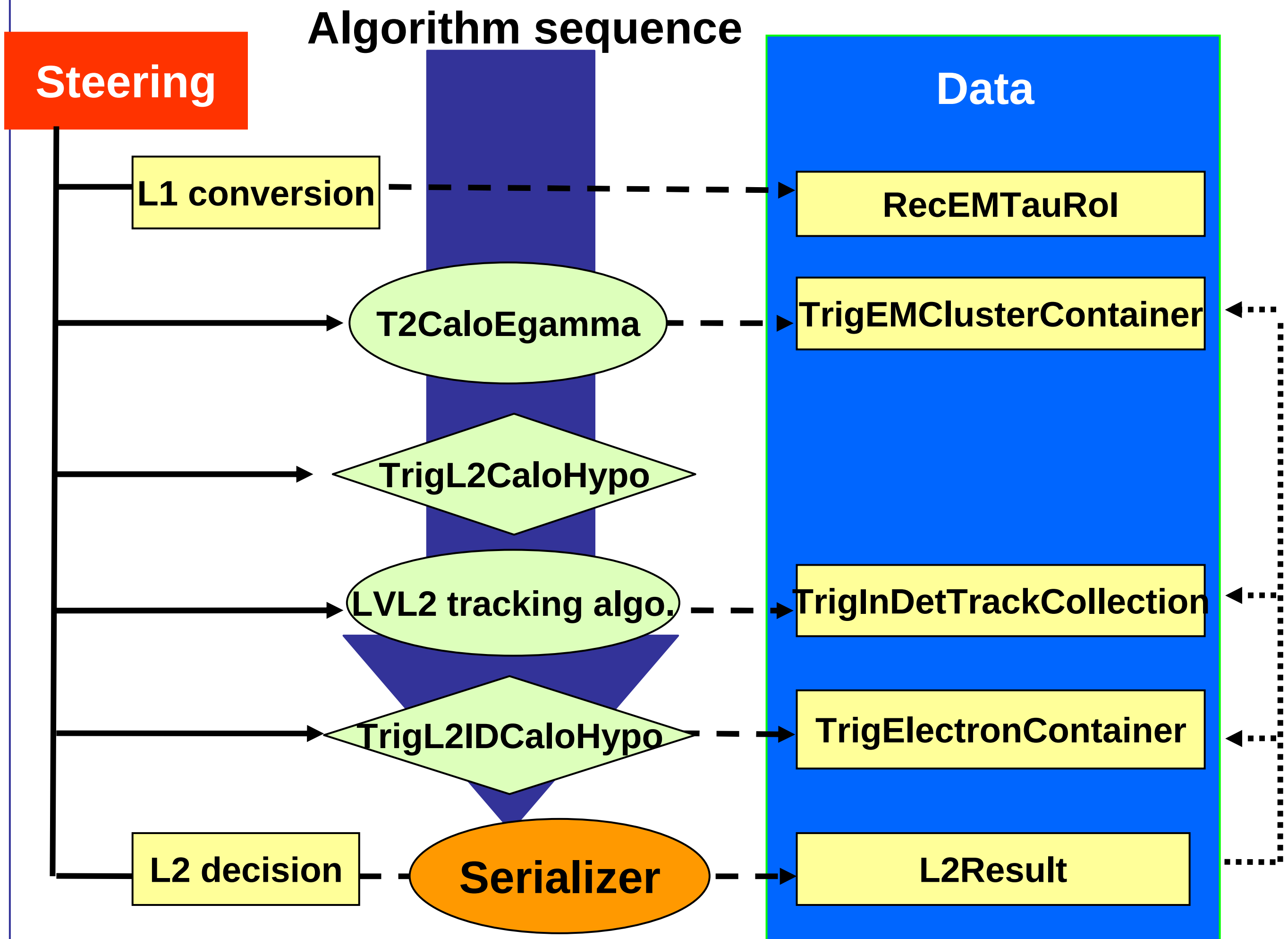
Concrete example: L2 electron chain

Level 1:

- Trigger Tower deposits are stored at digitization time
- Level 1, emulated by software, produces EM region of interest

HLT:

- Feature Extraction (FEX) algorithms interspersed with Hypothesis (Hypo) algorithms,to achieve early hypothesis rejection
- Navigation provided by steering using TriggerElements
- Steering also produces L2/EFResult



Overview of software

Trigger requirements

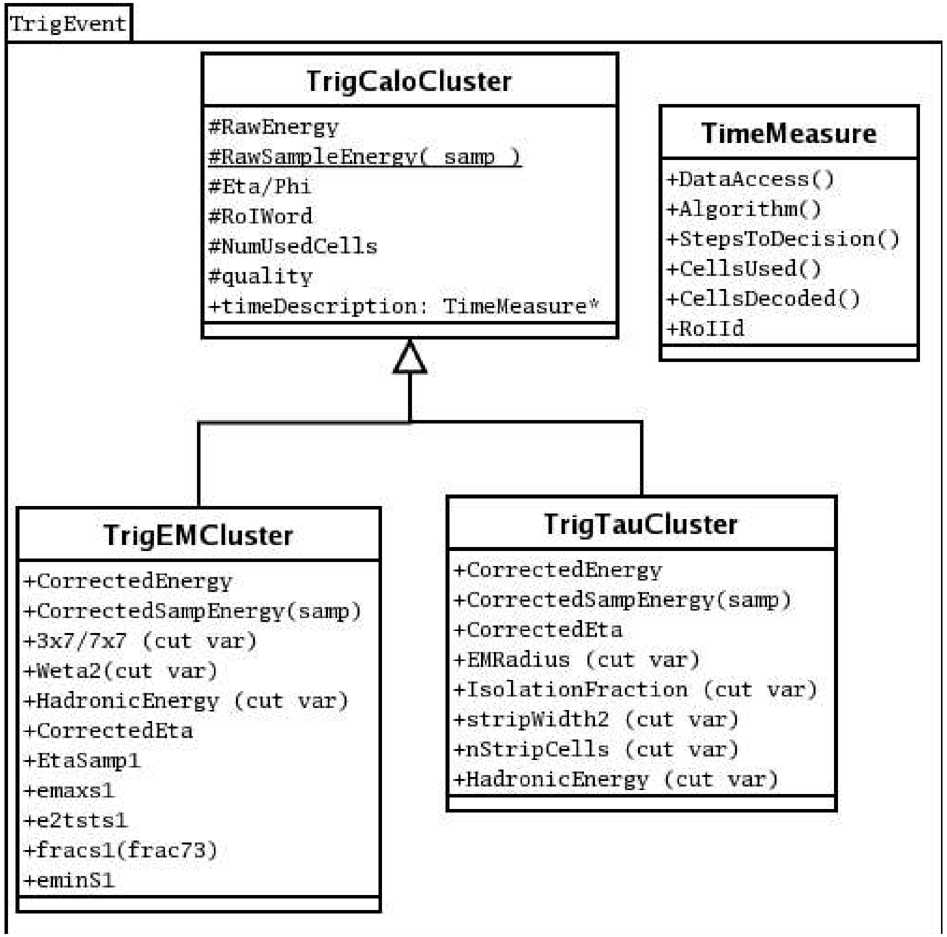
- These are some requirements that many algorithms and data classes should satisfy
- Algorithms must be:
  - **Fast**
  - Able to run **several times** per event (once or more per RoI)
  - Inherit from HLTAlgo
  - Called by the Steering
- Different instances of FEX algorithms can be run with different properties, depending on the type of RoI (EM, muon, jet, etc) to maximize efficiency
- Data classes must:
  - Be **small** (limited space for adding info to event)
  - Minimize access **time overheads**
  - Be **serializable** into L2/EFResult if they should ever be part of the monitoring or debug info from HLT
  - The L2Result and EFResult are the only way to bring online information to the offline domain, no AOD are produced in online running

Level 2 algorithms

- Access to full detector granularity
- Constrained by short execution time
- Alignment and calibration probably not optimal
- Feature Extraction (FEX) algorithms relevant for e/γ:
  - Calorimetry: T2CaloEgamma
    - Produces **TrigEMCluster** (since 11.0.5)
    - Currently being reviewed (review started last TDAQ week)
  - Tracking: SiTrack, IDScan, TRTxK, TRT extension tool
    - Produce collections of **TrigInDetTrack**
- Mostly mature software; well adapted to online running environment
- Hypothesis algorithms:
  - TrigL2CaloHypo, TrigL2IDCaloHypo since 11.0.x
    - Produce collections of **TrigElectron**
  - Photon hypothesis algorithm should be available soon, at least as first iteration
    - Will produce **TrigPhoton**

TrigEMCluster

- TrigEMCluster
- Base class in common with TrigTauCluster
- Cluster and shower shape definition variables
- Persistifiable in POOL
- Will be serializable, for (awaiting functionality in Serializer)



TrigEvent

TrigCaloCluster

#RawEnergy
#RawSampleEnergy(samp)
#Eta/Phi
#RoIWord
#NumUsedCells
#quality
+timeDescription: TimeMeasure*

TimeMeasure

+DataAccess()
+Algorithm()
+StepsToDecision()
+CellsUsed()
+CellsDecoded()
+RoIId

TrigEMCluster

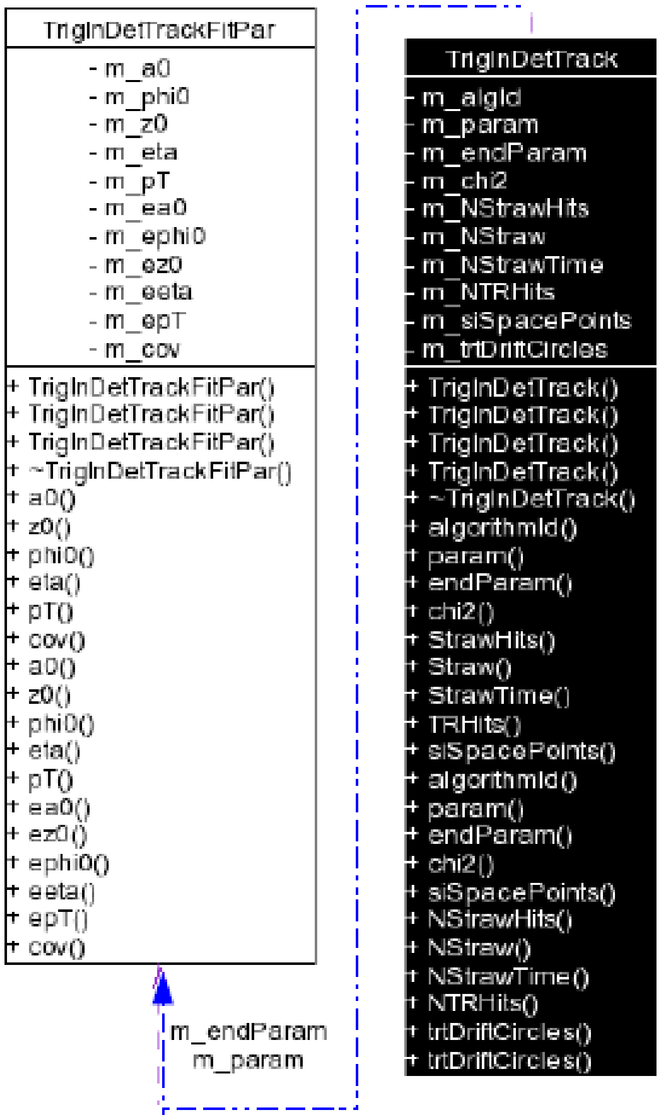
+CorrectedEnergy
+CorrectedSampEnergy(samp)
+3x7/7x7 (cut var)
+Weta2(cut var)
+HadronicEnergy (cut var)
+CorrectedEta
+EtaSamp1
+emaxs1
+e2tsts1
+fracs1(frac73)
+eminS1

TrigTauCluster

+CorrectedEnergy
+CorrectedSampEnergy(samp)
+CorrectedEta
+EMRadius (cut var)
+IsolationFraction (cut var)
+stripWidth2 (cut var)
+nStripCells (cut var)
+HadronicEnergy (cut var)

TrigInDetTrack

- Produced by all tracking algorithms at level 2
- One TrigInDetTrack points to one or two sets of fit parameters (at least perigee)
- Is persistifiable in POOL and serializable in L2Result
- Total size is about 20 doubles and a few pointers which are not stored



TrigInDetTrackFitPar

- m_a0
- m_phi0
- m_z0
- m_eta
- m_pT
- m_ea0
- m_ephi0
- m_ez0
- m_eeta
- m_epT
- m_cov

+ TrigInDetTrackFitPar()
+ TrigInDetTrackFitPar()
+ TrigInDetTrackFitPar()
+ ~TrigInDetTrackFitPar()
+ a0()
+ z0()
+ phi0()
+ eta()
+ pT()
+ cov()
+ a0()
+ z0()
+ phi0()
+ eta()
+ pT()
+ ea0()
+ ez0()
+ ephi0()
+ eeta()
+ epT()
+ cov()

m_endParam
m_param

TrigInDetTrack

- m_algId
- m_param
- m_endParam
- m_chi2
- m_NStrawHits
- m_NStraw
- m_NStrawTime
- m_NTRHits
- m_siSpacePoints
- m_trtDriftCircles

+ TrigInDetTrack()
+ TrigInDetTrack()
+ TrigInDetTrack()
+ TrigInDetTrack()
+ ~TrigInDetTrack()
+ algorithmId()
+ param()
+ endParam()
+ chi2()
+ StrawHits()
+ Straw()
+ StrawTime()
+ TRHits()
+ siSpacePoints()
+ algorithmId()
+ param()
+ endParam()
+ chi2()
+ siSpacePoints()
+ NStrawHits()
+ NStraw()
+ NStrawTime()
+ NTRHits()
+ trtDriftCircles()
+ trtDriftCircles()

TrigParticle

- To store the candidate object that was accepted by a signature
- Should be light and small, with no ElementLinks or heavy inheritance, to ease persistency
- Example:
  - `TrigElectron`
  - Summary data to use for **debugging** and **analysis**
  - `TrigElectron` data members:
    - `Roi_Id` // set by LVL2 steering
    - `eta, phi`
    - `Z vertex`
    - p_T, E_T
    - **`valid` // set by hypo algorithm**
    - **pointer** to track
    - **pointer** to cluster

    ("best estimate" values from HLTHypoAlg)

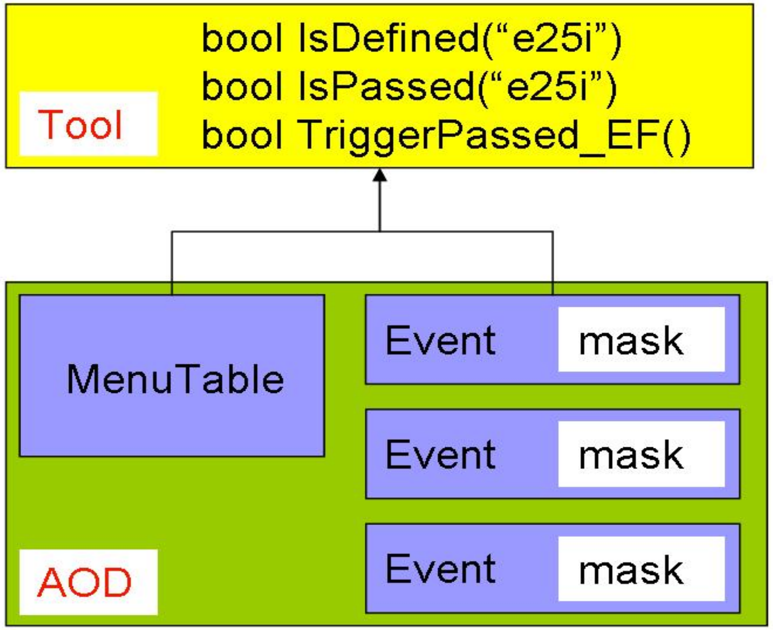
```
class TrigElectron {
 public:
    TrigElectron();
...
    TrigTrack*   track();
    TrigCluster* cluster();
    int RoI();

    bool isValid();
 private:
int m_roiID;
float m_eta, m_err_eta;
float m_phi, m_err_phi;
float m_Zvtx, m_err_Zvtx;
float m_Pt, m_err_Pt;
bool m_valid;
const TrigEMCluster* m_cluster;
const TrigInDetTrack* m_track;
};
```

Event Filter algorithms

- Running offline software in seeded mode (wrapped by trigger algorithms)
- EDM is common between trigger and offline, but is impossible to serialize, and so cannot be used for offline debugging of trigger
- A workaround has been found: “serializing” references to objects in POOL
- The separation between transient and persistent EDM could help!
- Feature Extraction (FEX) algorithms relevant for e/γ:
  - Calorimetry: **TrigCaloRec** wraps CaloRec
  - Tracking: EF ID **new tracking** used by trigger
  - e/gamma reconstruction: **TrigEgammaRec** wraps egammaRec (see Teresa’s talk)
  - Produce CaloCluster, Trk::Track, TrackParticle, egamma, etc
- Hypothesis algorithms: cannot be re-run on ESD (see below), more expected in 12.0.x

TriggerDecision

- Signatures passed/failed/prescaled encoded in a **bit pattern** stored once per event



- The bit pattern will change and must be **interpreted** through a MenuTable (in Conditions Database)

- A Tool would provide the user interface to L1/L2/EF and individual signature results by interpreting bit patterns in AOD. It would give:
  – Decision bit for each signature
  – Access to configuration through methods like isDefined()

TriggerDecision

- **Short term implementation**: while there are only a few signatures and no database
  - Store object in AOD consisting basically of a map:

    `map<string label, bool accept>` and little else
  - Derive trigger decisions from Hypothesis algorithms and L2/EFResult
  - Only a few signatures - wasted AOD space by repeating labels each event is negligible
- This gives the “user” a similar look and feel as the real thing when doing analysis
- To use:
  - For each event, retrieve the **TriggerDecision** object from StoreGate
  - Then use TriggerDecision methods to determine if the event passed signature XYZ, etc
- Algorithm (TriggerDecisionMaker) to fill TriggerDecision being tested; expected in 12.0.1

The Serializer

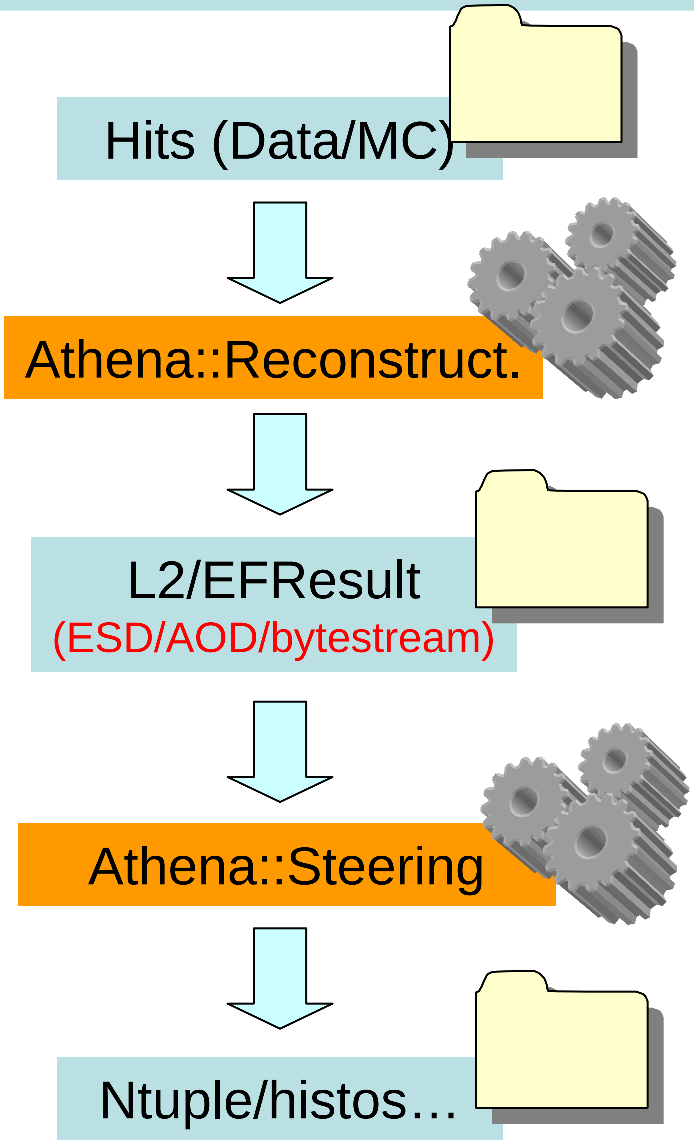
- Part of Steering
- Uses Reflex library to serialize classes into `L2Result` and `EFResult` in bytestream or POOL
- Works with:
  - Simple native types (**int**, **double**, **float**)
  - **Pointers** (and **NULL** pointers)
  - Follows (non-NULL) pointers
  - Classes need to have **SEAL** dictionaries (same requirement for POOL)
  - All unsupported class data members should be declared `transient` (in `selection.xml`)
  - Store references to POOL objects (e.g. to write `EFResult,` where objects are **not** serializable)
- To do:
  - STL containers (std::vector<>, DataVector<>)
  - Schema evolution

Trigger analysis & optimisation

How to analyse ESD/AOD/BS

- To produce ESD/AOD:
  1. Just set doTrigger = True
  2. The Steering serializes all relevant data objects into L2/EFResult (ok for L2Result at present)
  3. TriggerDecision object filled after Steering has run and put in ESD/AOD
- To analyse ESD/AOD:
  A. Either:
     1. Set doTrigger = False
     2. Retrieve TriggerDecision and find result

  B. Or
     1. Set doTrigger = True (& etc)
     2. The Steering de-serializes all data objects
     3. Run hypothesis algorithms only on reconstructed features
     4. Retrieve new TriggerDecision at the end
- Note: the cuts can only be tightened with respect to the original cuts



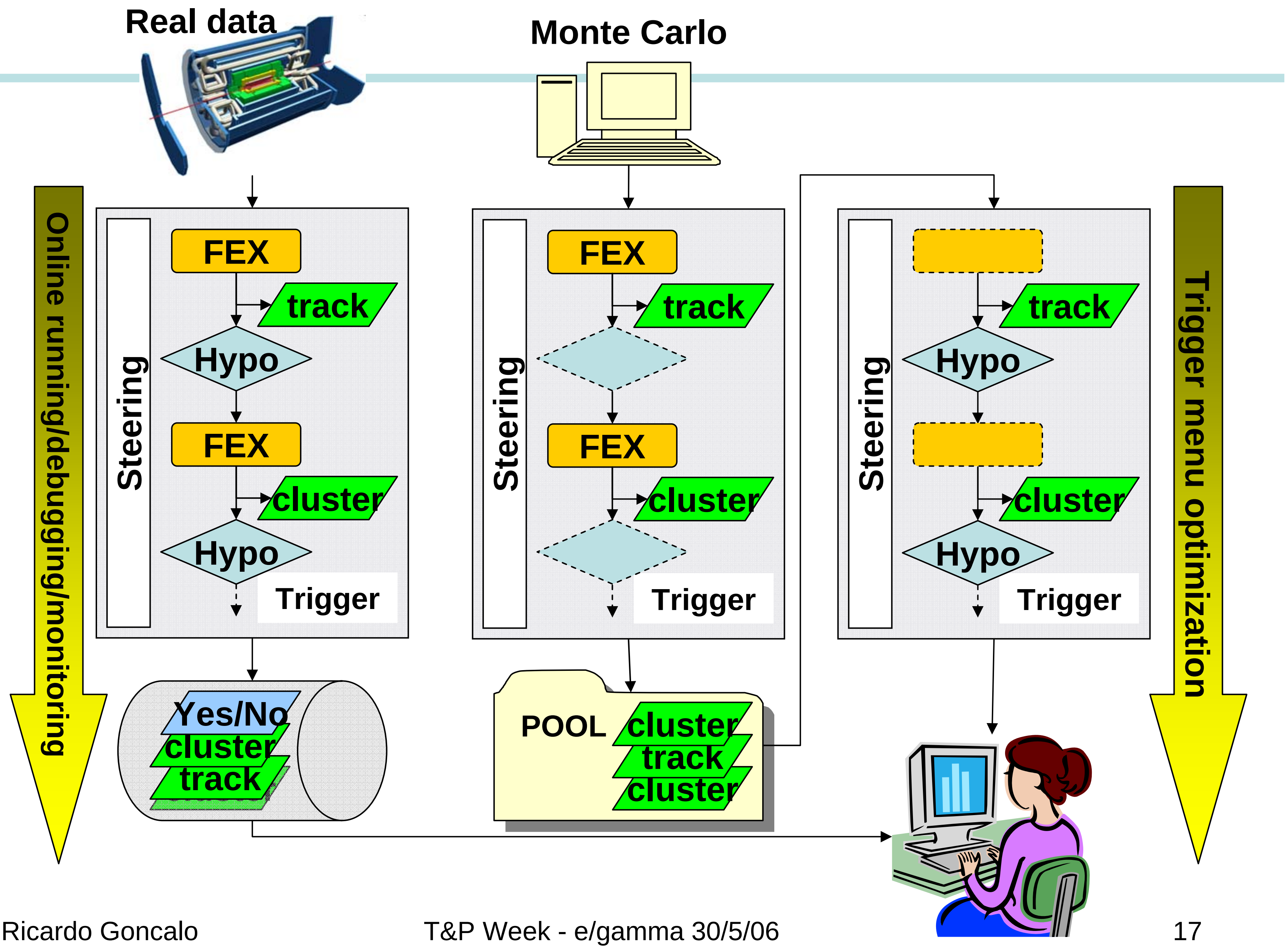
Hits (Data/MC)

Athena::Reconstruct.

L2/EFResult
(ESD/AOD/bytestream)

Athena::Steering

Ntuple/histos…


Real data
Monte Carlo
Online running/debugging/monitoring
Steering
FEX
track
Hypo
FEX
cluster
Hypo
Trigger
Steering
FEX
track
FEX
cluster
Trigger
Steering
track
Hypo
cluster
Hypo
Trigger
Trigger menu optimization
Yes/No
cluster
track
POOL
cluster
track
cluster

More information

- See Monika's tutorial in Japan for more information

`http://agenda.cern.ch/fullAgenda.php?ida=a062235`

- More on HLT persistency

`https://uimon.cern.ch/twiki/bin/view/Atlas/HLTPersistencyRecipe`

- List of available information in AOD/ESD

`https://uimon.cern.ch/twiki/bin/view/Atlas/TriggerEDM`

- See e/gamma analysis Wiki for example analysis job:

`https://uimon.cern.ch/twiki/bin/view/Atlas/EgammaTriggerAnalysis`

Software status and outlook

Software status for level 2 e/γ

- FEX algorithms mostly in good shape thanks to the work of many people
- TrigInDetTrack – Truth association not in good shape: prototype exists but needs to be finished and tested
  - Urgent! Expected for **12.0.x**
- Hypothesis algorithms exist in 11.0.5 and have been tried
  - Tried for electron chain only
  - Must be tuned
  - Different signatures can use instances of same hypo algorithms created with different cuts
  - TrigElectrons produced by last hypo in chain:
    - **AcceptAll** property means **all** TrigEMClusters accepted and one TrigElectron produced for **each** track-cluster combination
    - Hypothesis decision still kept (see **`isValid()`** class method)
  - Photon hypothesis expected soon (12.0.1? Prototype exists but not tested yet)
    - Will produce **TrigPhoton**, similar to TrigElectron

Software status for Event Filter e/γ

- TrigCaloRec: in place since 11.0.x
  - CaloCluster (offline class) produced: persistifiable in POOL

- EF ID algorithms available since release 11.0.0 (under InnerDetector/InDetTrigRecAlgs)
  - Under review
  - Most work is in performance studies and improvements
  - Was successfully run using electron settings (split from default)

- Hypothesis algorithms cannot be re-run on ESD/AOD due to technical problems
  - Expected early in 12.0.x

- TrigEgammaRec: available in 11.5.0; can be run in 11.0.5 with tag TrigEgammaRec-00-00-04
  - Difference found between offline and wrapped algorithm: under investigation

Status of global trigger software

- Steering: being redesigned to add missing functionality (e.g. topological triggers) – expected late in 12.0.x?
  - A record of the Steering navigation should be available offline, to connect L2/EF electron candidates with L1 RoIs: planned
- Serializer: need to handle STL containers; coupled with persistent/transient EDM
- TriggerDecisionMaker – expect in 12.0.1
- Analysis tools:
  - Would be very good to be able to use offline Analysis Tools machinery in trigger analysis
  - Discussion started on how to proceed
  - One option would be to make TrigElectron inherit from INavigable4Momentum, but this may impact on serialization
  - Expect this after more urgent functionality in place

Level	Class/Algorithm	11.0.5	12.0.x
L2	TrigEMCluster	✓	✓
L2	TrigInDetTrack	✓	✓
L2	TrigInDetTrackTruth	✗	12.0.3
L2	TrigElectron	✓	Inherit INav4Mom: 12.0.3
L2	TrigPhoton	✗	12.0.1
L2	Electron Hypothesis	✓	✓
L2	Photon Hypothesis	✗	12.0.1
L2	FEX algorithms	✓	✓
EF	EDM classes	✓	✓
EF	New Tracking FEX	✓	✓
EF	Track-truth association	✓	New version in 12.0.x
EF	TrigCaloRec	✓	✓
EF	TrigEgammaRec	With TrigEgammaRec-00-00-04	12.0.3
EF	Hypothesis algorithms	✗	12.0.x
All	TriggerDecision	✓	✓
All	TriggerDecisionMaker	✗	12.0.1
All	Serializer	✓	New functionality needed

- Comparison CBNT vs AOD for 25GeV single electrons (8500 events)

	$\|\eta\|$<2.5 incl crack		$\|\eta\|$<2.5 excl crack	
IDScan	Eff % AOD	Eff % CBNT	Eff % AOD	Eff % CBNT
L1	92.9	92.9	96.2	96.2
L2 Calo	91.1	91.1	95.1	95.1
L2 Match	86.0	86.5	89.8	90.6
EF	75.2	76.2	79.5	80.1

- Shows some differences which need to be understood/fixed

Conclusions

- With release 12 (most likely 12.0.3) there will be a significant amount of trigger information in the AODs
- This together with the possibility of re-running the trigger will generate enough information to tune signatures
- Together with other slices, we should manage to have first attempt at a menu later this year. In time for third Trigger & Physics week

Backup slides

Size of AOD/ESD L1/L2 objects (DS 4022: top)

L1EMTauObjectContainer	0.09 kB
L1JetObjectContainer	0.05 kB
L1EtmissObject	~0 kB
LVL1_ROI	0.12 kB
CTP_Decision	0.004 kB
LVL1::JetElement	1.91 kB
LVL1::TriggerTower	6.18 kB
L2Result	1.44 kB
TrigElectron	4.32 kB
TrigEMCluster	0.16 kB
TrigInDetTrackCollection	5.33 kB
TrigTauClusterContainer	1.05 kB
TrigT2JetClusterContainer	0.09 kB
MuonFeature	~0 kB

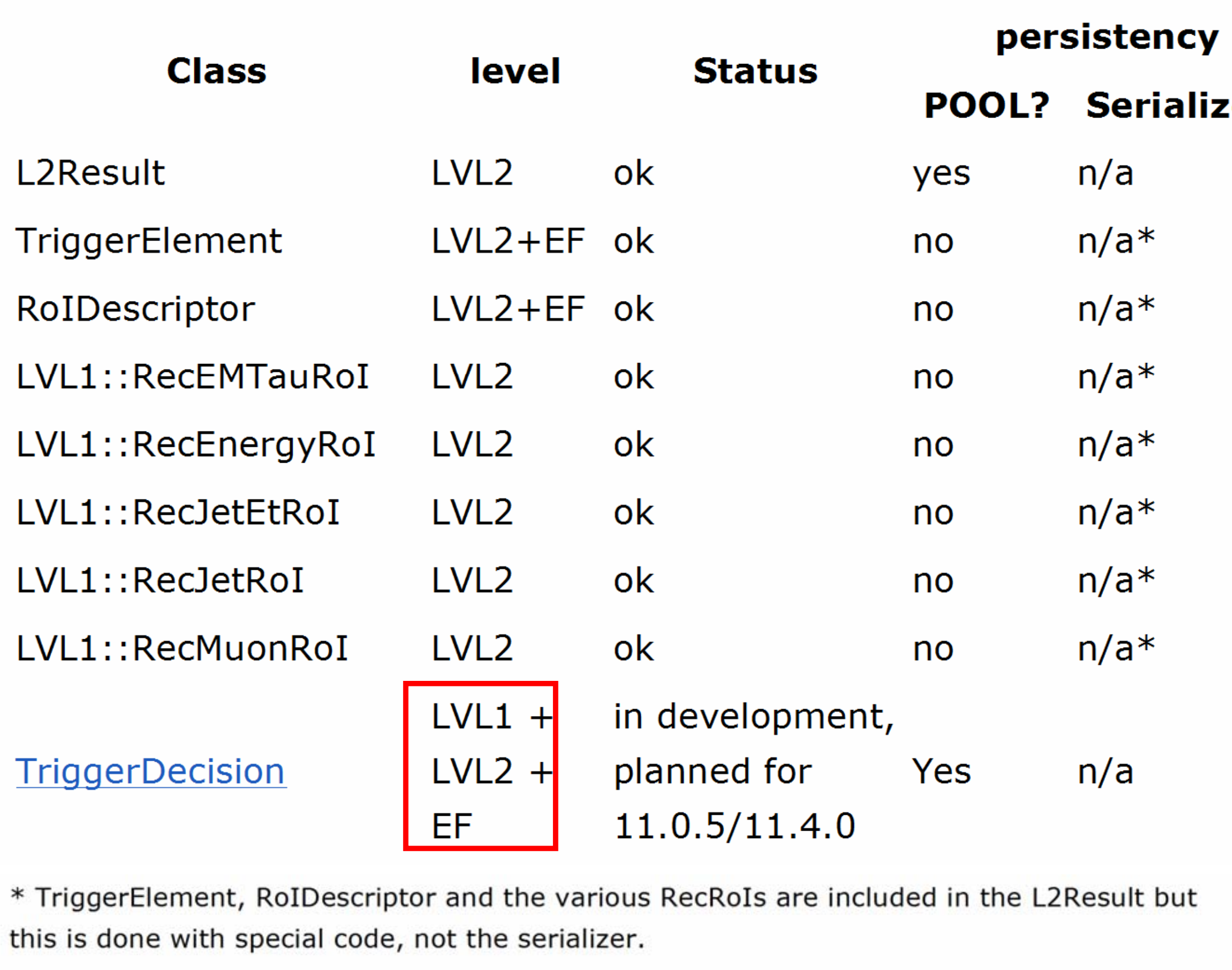
<10% of AOD contents

few% of ESD contents

What do we have at present?

From: https://uimon.cern.ch/twiki/bin/view/Atlas/TriggerEDM

HLT



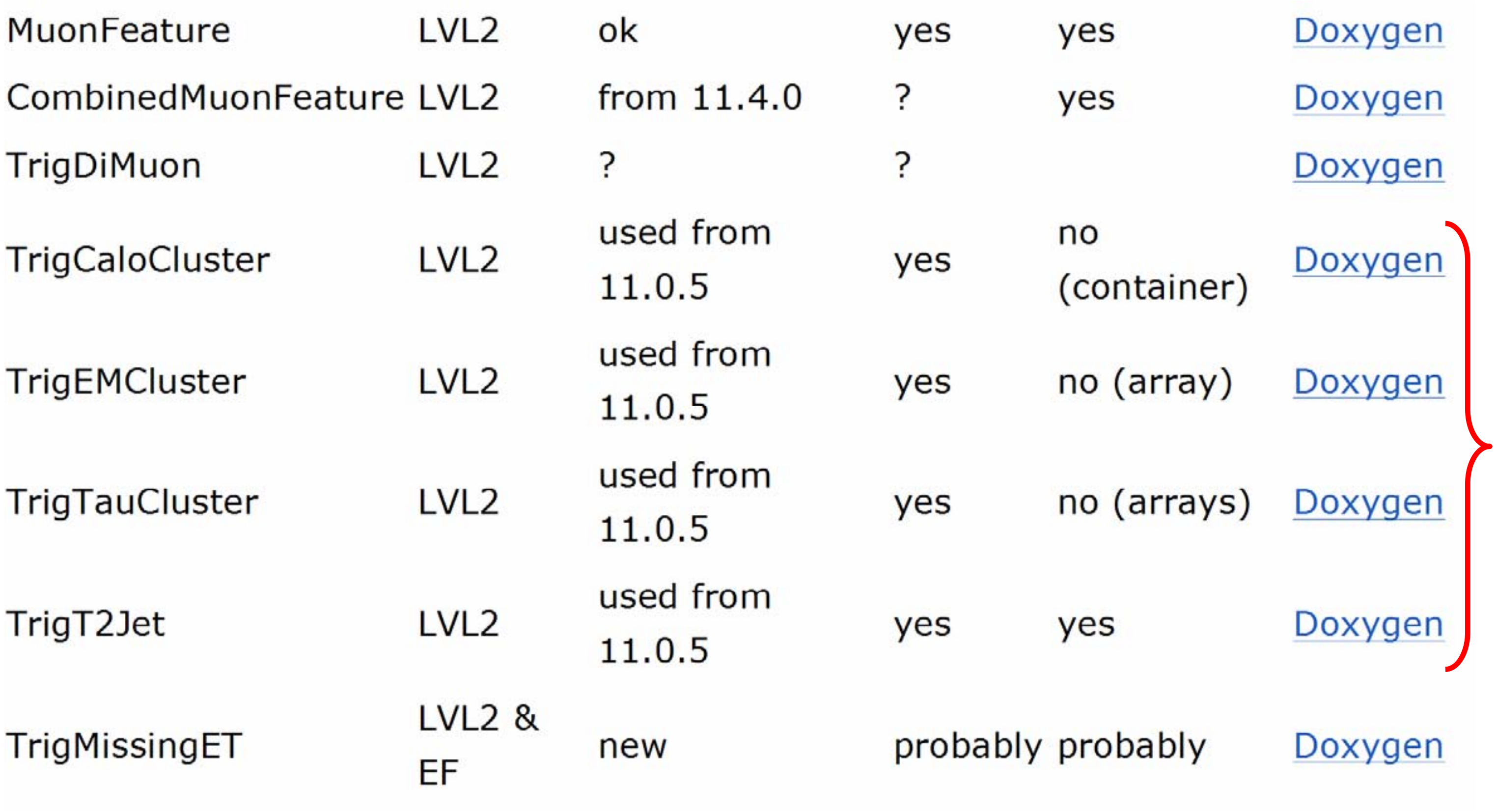
Class	level	Status	persistency		Documentation
			POOL?	Serializer?	in 11.0.5
L2Result	LVL2	ok	yes	n/a	Doxygen
TriggerElement	LVL2+EF	ok	no	n/a*	Doxygen
RoIDescriptor	LVL2+EF	ok	no	n/a*	Doxygen
LVL1::RecEMTauRoI	LVL2	ok	no	n/a*	Doxygen
LVL1::RecEnergyRoI	LVL2	ok	no	n/a*	Doxyegn
LVL1::RecJetEtRoI	LVL2	ok	no	n/a*	Doxygen
LVL1::RecJetRoI	LVL2	ok	no	n/a*	Doxygen
LVL1::RecMuonRoI	LVL2	ok	no	n/a*	Doxygen
TriggerDecision	LVL1 + LVL2 + EF	in development, planned for 11.0.5/11.4.0	Yes	n/a	Doxygen

* TriggerElement, RoIDescriptor and the various RecRoIs are included in the L2Result but this is done with special code, not the serializer.

Steering objects

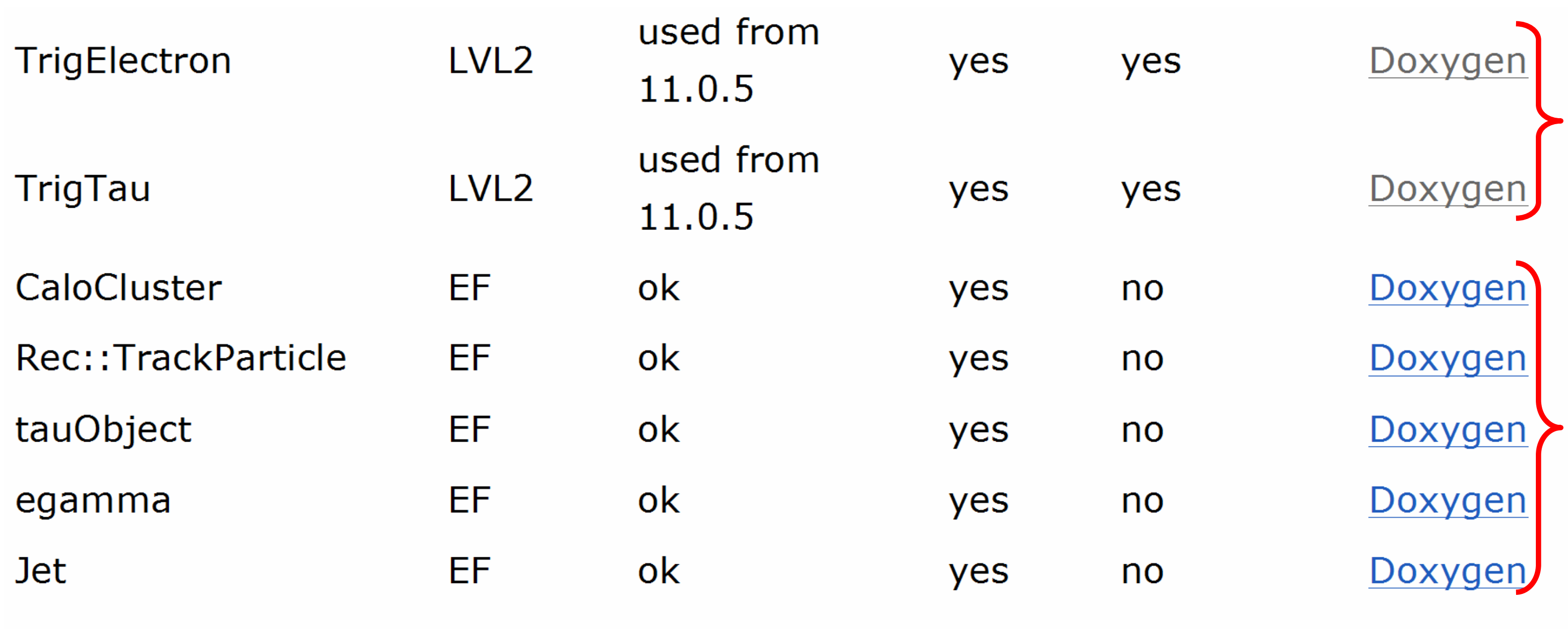
Accessible from both BS and POOL

Algorithm to fill it in 12.0.1



MuonFeature	LVL2	ok	yes	yes	Doxygen
CombinedMuonFeature	LVL2	from 11.4.0	?	yes	Doxygen
TrigDiMuon	LVL2	?	?		Doxygen
TrigCaloCluster	LVL2	used from 11.0.5	yes	no (container)	Doxygen
TrigEMCluster	LVL2	used from 11.0.5	yes	no (array)	Doxygen
TrigTauCluster	LVL2	used from 11.0.5	yes	no (arrays)	Doxygen
TrigT2Jet	LVL2	used from 11.0.5	yes	yes	Doxygen
TrigMissingET	LVL2 & EF	new	probably	probably	Doxygen

New Calo EDM from 11.0.5



TrigElectron	LVL2	used from 11.0.5	yes	yes	Doxygen
TrigTau	LVL2	used from 11.0.5	yes	yes	Doxygen
CaloCluster	EF	ok	yes	no	Doxygen
Rec::TrackParticle	EF	ok	yes	no	Doxygen
tauObject	EF	ok	yes	no	Doxygen
egamma	EF	ok	yes	no	Doxygen
Jet	EF	ok	yes	no	Doxygen

Only electron and tau for now…

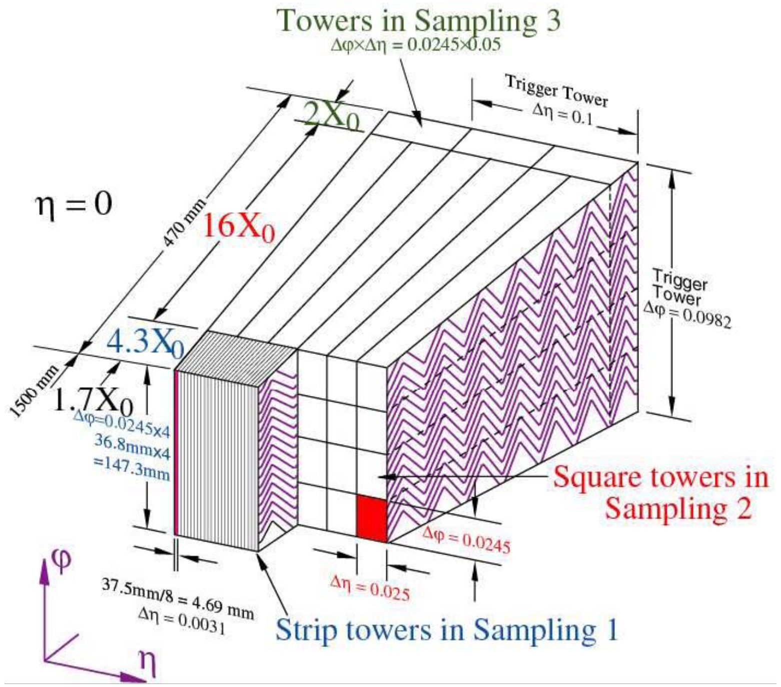
From Offline EDM: not serial.

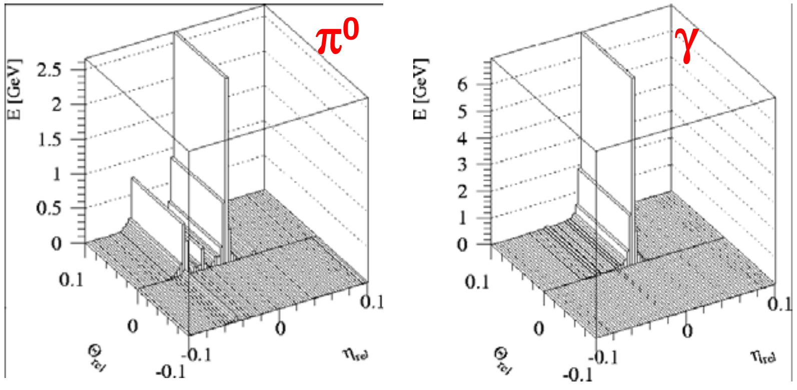
What’s missing?

- **Truth-association classes**
- **New functionality in Serializer (STL container serialization)**
- **Having all objects in L2/EFResult (EF “persistent” EDM? Review started.)**
- **Some steering functionality: “accept all” mode**
- **What else?**

Level 2: T2CaloEgamma

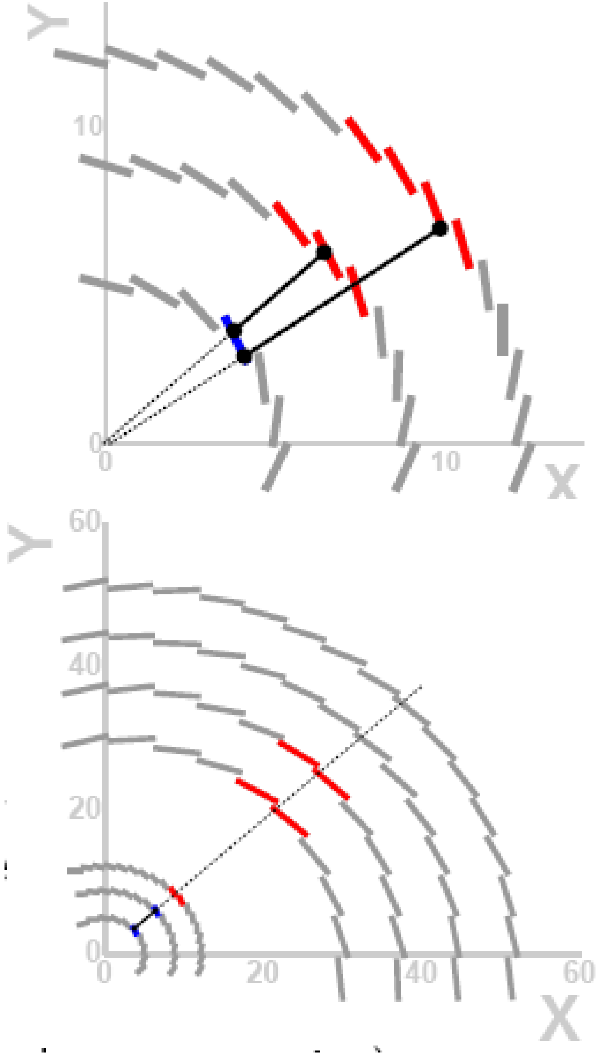
- Full granularity but short time and perhaps incomplete calibration & alignment
- Extends T2CaloCommon
- Data access factorized from algorithmic part
- T2CaloEgamma calls AlgTools:
  1. EgammaSamp2Fex: LAr sample 2; cluster position and size
  2. EgammaSamp1Fex: LAr sample 1; look for second maxima in strips
  3. EgammaEmEnFex: total cluster energy; include calibration
  4. EgammaHadEnFex: longitudinal isolation (leakage)
- Each calculates shower-shape variables with discrimination power
- Produced TrigEMCluster (since 11.0.5)
- Well developed code; important progress in time performance
- Review started last TDAQ week





Level 2 tracking: SiTrack

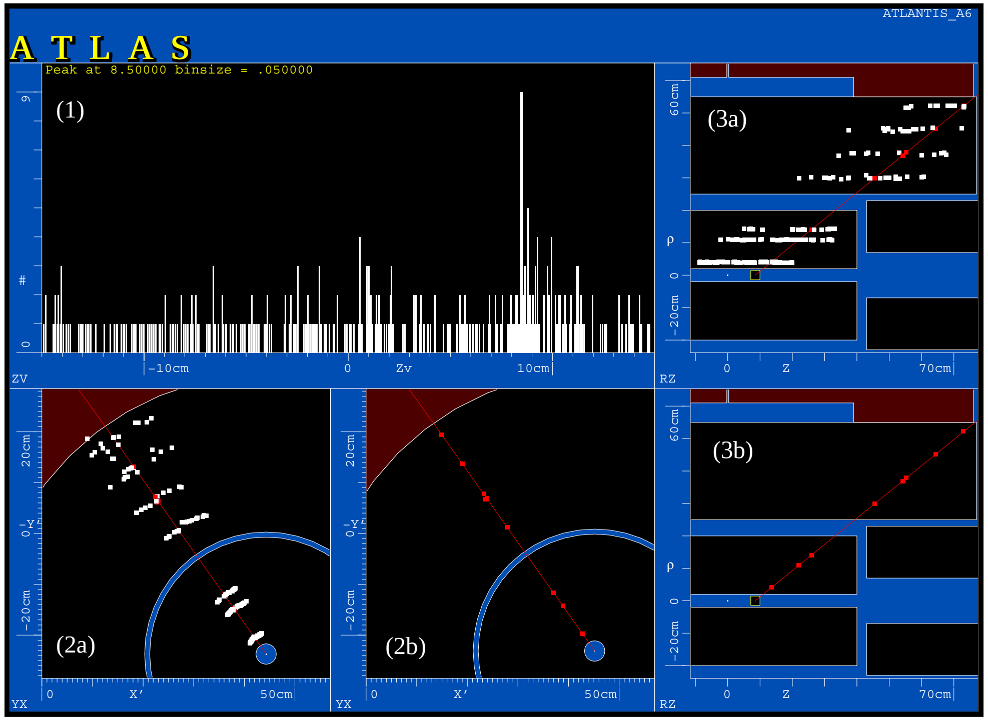
- SiTrack:
  1. Sorting: retrieve SpacePoints in ROI and sort them to speed up algorithm
  2. Seeding: look for pairs (seeds) of SpacePoints with one in b-layer
  3. Extension: extend outwards to find third point
  4. Merging triplets according to common hits
  5. Track fit: linear fit (analytic least squares) in r-Z; circle in r-ϕ
- Heavy use of lookup tables to achieve fast algorithm



Level 2 tracking: IDScan

IDScan:

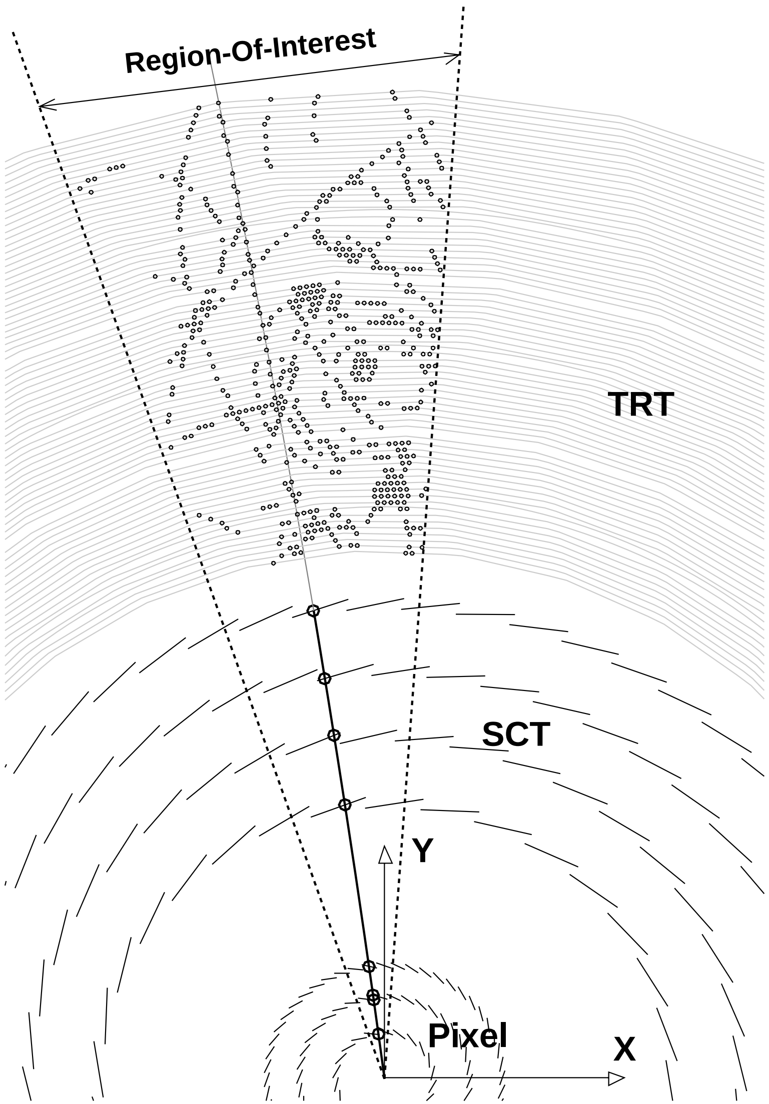
1. ZFinder: form pairs of hits in thin ϕ slices; extrapolate inwards to find Z_{vtx} at beamline from histogram
2. HitFilter: using Z_{vtx}, make 2D histogram of hits in η-ϕ plane; remove bins with hits from few layers
3. GroupCleaner: do 2D histogram using SpacePoint triplets in $1/p_T$-ϕ plane; tracks from bins with hits in 4/7 layers
4. Fitter: use Kalman technique on selected SpacePoints starting from already estimated Z_{vtx}, $1/p_T$, η, ϕ



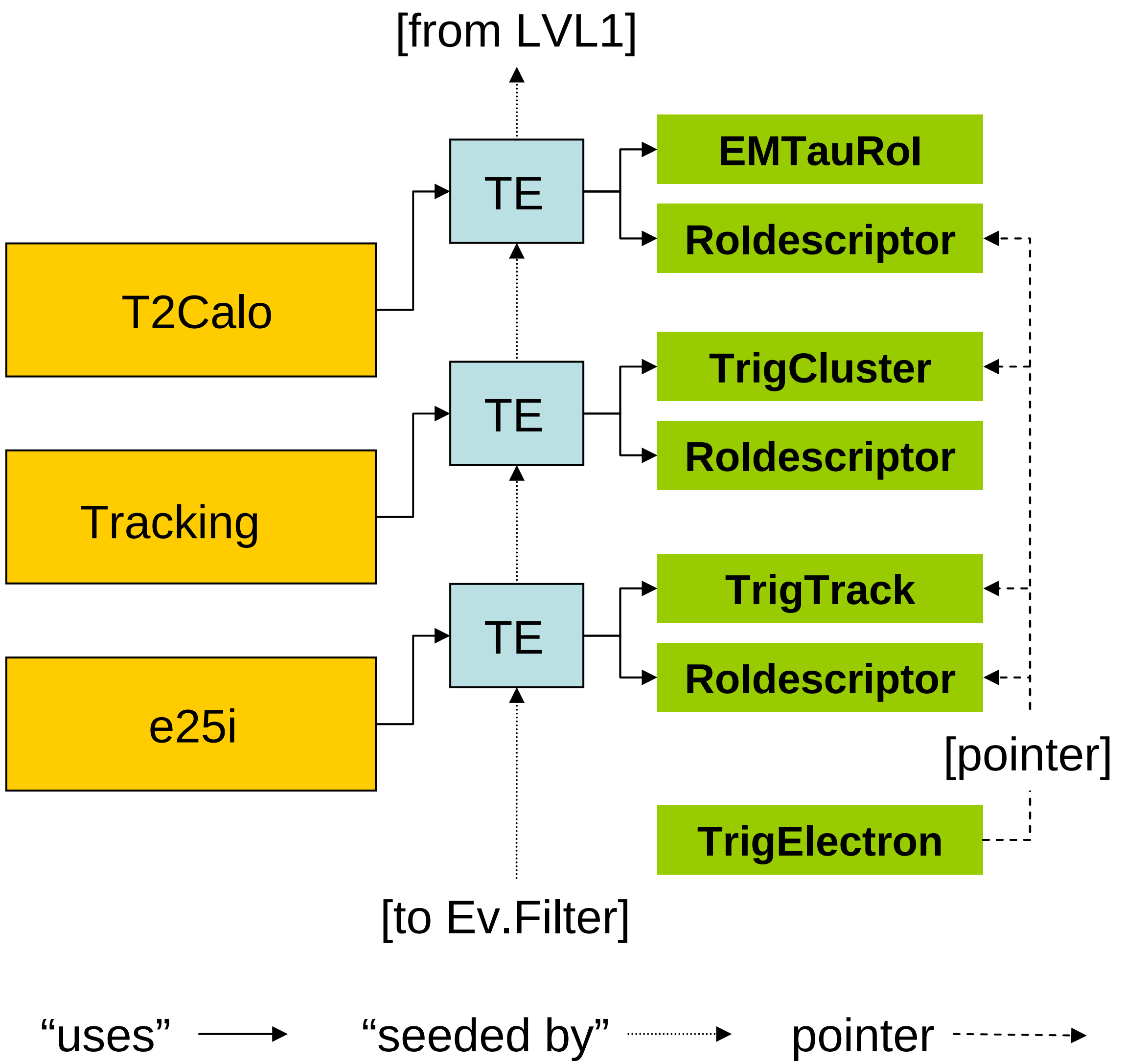
Level 2 tracking: TRT

- TRTxK
  - Wrapper for offline xKalman to be used in seeded mode
  - May be used to do standalone TRT tracking or using all ID detectors

- TRT extension tool
  - Extends tracks outwards to TRT by associating hits to the track according to the Probabilistic Data Association Filter (PDAF, com-daq-2005-022)
  - Tracking algorithms may be configured to use TRT extension



Navigation



[from LVL1]

T2Calo

Tracking

e25i

TE

TE

TE

EMTauRoI

RoIdescriptor

TrigCluster

RoIdescriptor

TrigTrack

RoIdescriptor

[pointer]

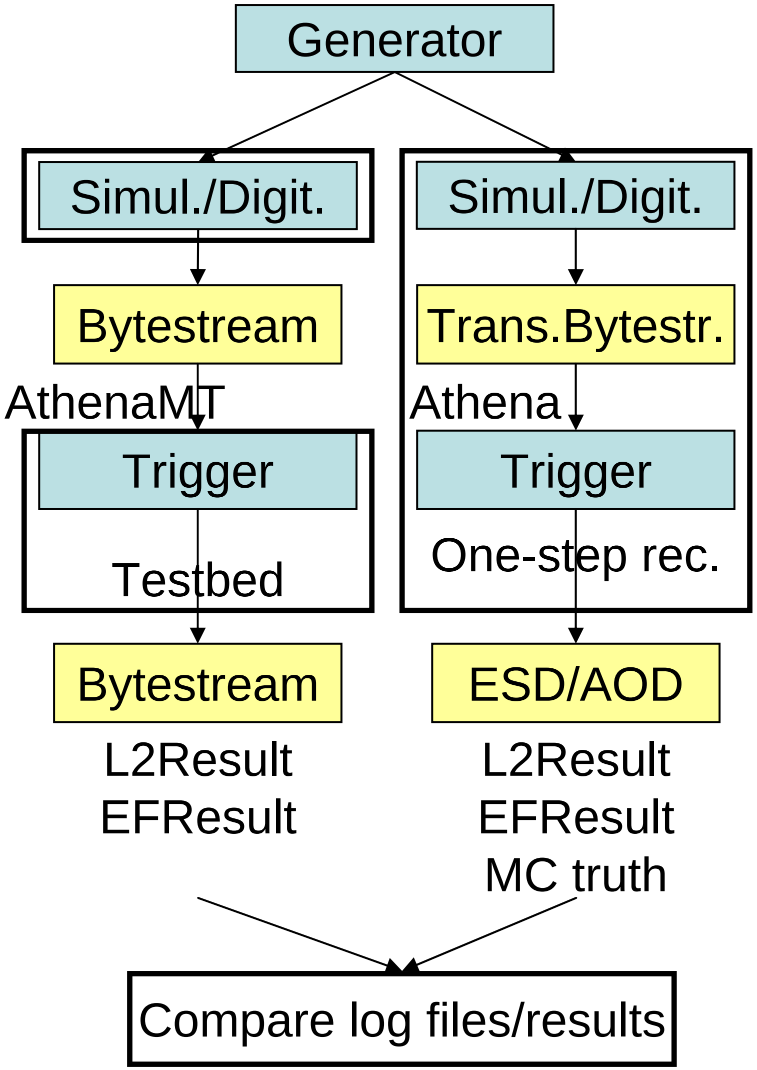
TrigElectron

[to Ev.Filter]

“uses” “seeded by” pointer

Trigger operation debugging

- Statistical: monitoring histograms
  - No need to store any data offline
  - Probably not useful if we're looking for small effects or unusual occurrences
- Event by event:
  - Only way to make HLT event data available is through L2/EFResult ⇒ Serializer
  - Regression-type test, running on bytestream data and comparing to ESD: in both cases, retrieve data from L2/EFResult and run hypothesis algorithms on it
  - MC truth information also available in ESD/AOD: compare reconstruction and truth


Generator
Simul./Digit.
Simul./Digit.
Bytestream
Trans.Bytestr.
AthenaMT
Athena
Trigger
Trigger
Testbed
One-step rec.
Bytestream
ESD/AOD
L2Result
EFResult
L2Result
EFResult
MC truth
Compare log files/results