Online Event Selection at the LHC

Part II: Regional and Partial Event Reconstruction

Teddy Todorov

CERN School of Computing 2003
Krems an der Donau, Austria

Outline

- Event reconstruction
  - Local reconstruction
  - Global reconstruction
- Conceptual differences between on-line and offline reconstruction
  - Reconstruction stages from raw data to trigger decision
  - Seeded reconstruction
- Regional event reconstruction
- Partial event reconstruction
- ATLAS implementation: Regions of Interest
- CMS implementation: Reconstruction on demand

Reconstruction

- I will not attempt to define reconstruction, I will just describe (some of) it
- Can be divided in “local” and “global”
  - Local: at the level of one detector
  - Global: using several detectors
  - Depends on what you call a detector!

What is a detector?

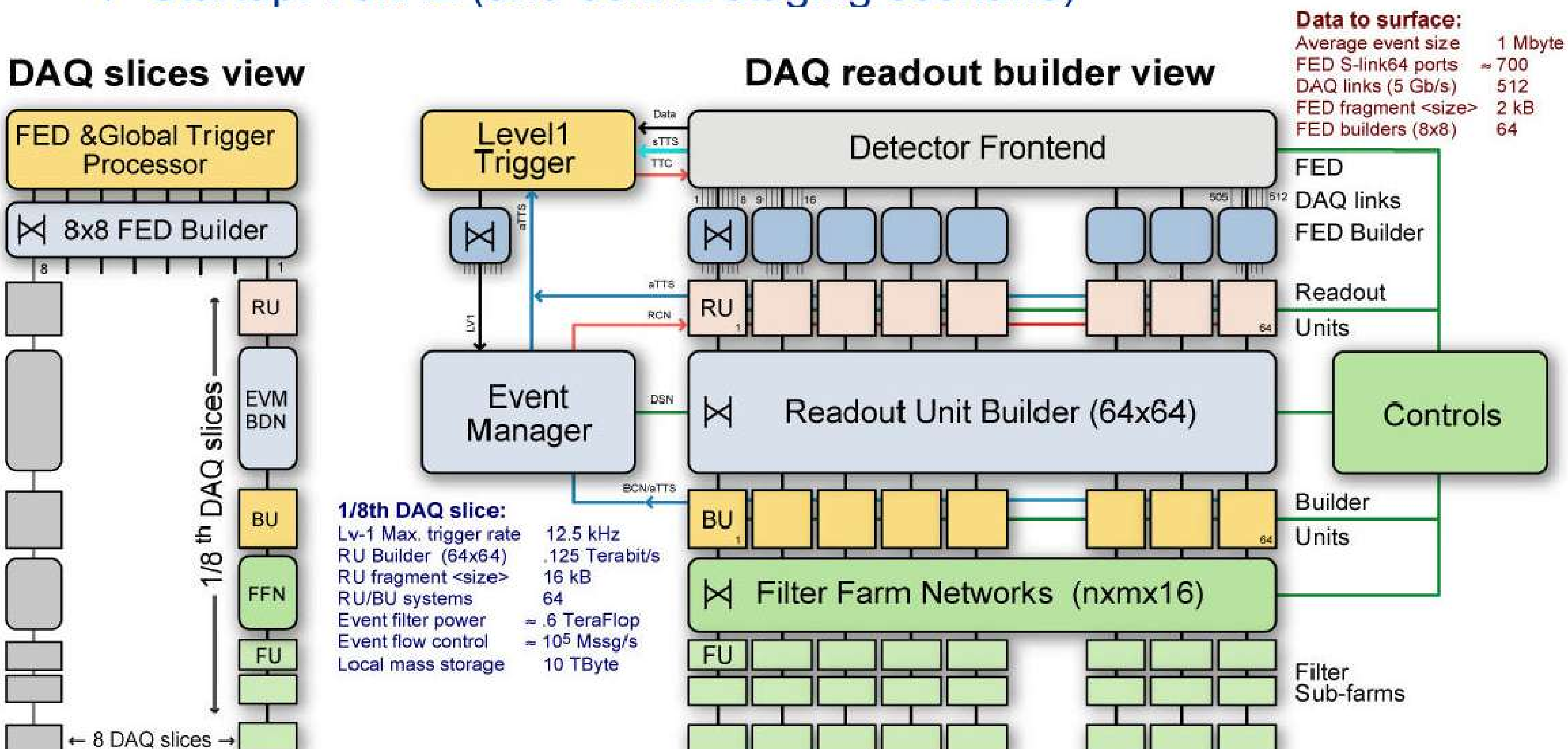
- Detectors are hierarchical. For example, CMS consists of
  - Muon system
  - Electromagnetic calorimeter
  - Hadronic calorimeter
  - Inner tracker system
- The inner tracker of CMS consists of hierarchical arrangment of silicon sensors
- A silicon sensor cannot be divided further – let's call it a detector unit
- From the event data point of view a detector unit is a collection of channels
- A channel typically contains one scalar amplitude per event, but can be more complex
  - e.g. CMS ECAL channels have 10 time-sampled amplitudes

Reconstruction hierarchy

- Reconstruction is also hierarchical
  - Channel level
    - e.g. Applying calibration
  - Detector unit level
    - e.g. Finding clusters of strips or pixels in a silicon detector and assigning positions and errors to them
  - Detector system level
    - e.g. Track reconstruction in the muon sytem
  - Global
    - e.g. Combined muon system – inner tracker track reconstruction
- Normally you don't care – it's obvious! But in On-line reconstruction you pay (in CPU time) for every bit of reconstruction you do

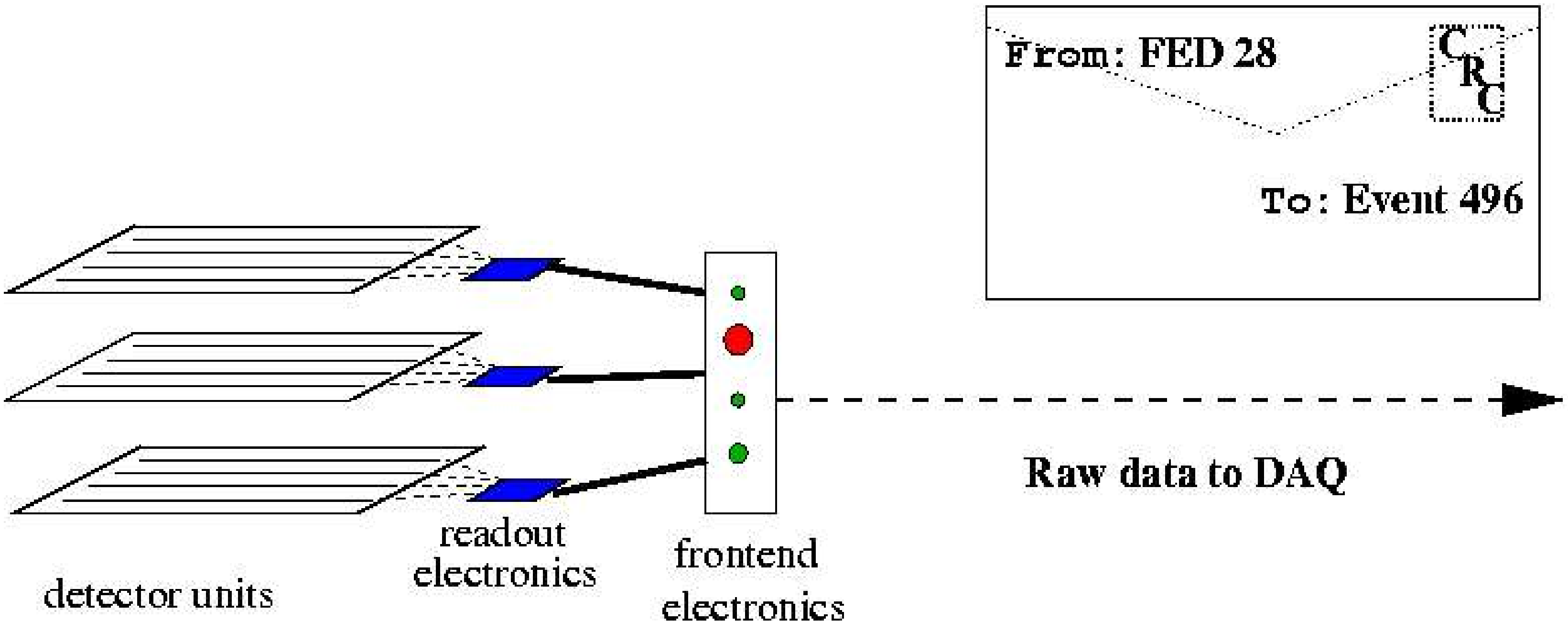
DAQ: multiples of 12.5 kHz

- Installation of the DAQ/HLT system will proceed in multiples of segments, each segment being 1/8 of the full system
  - Startup: 75kHz (and 50kHz staging scenario)



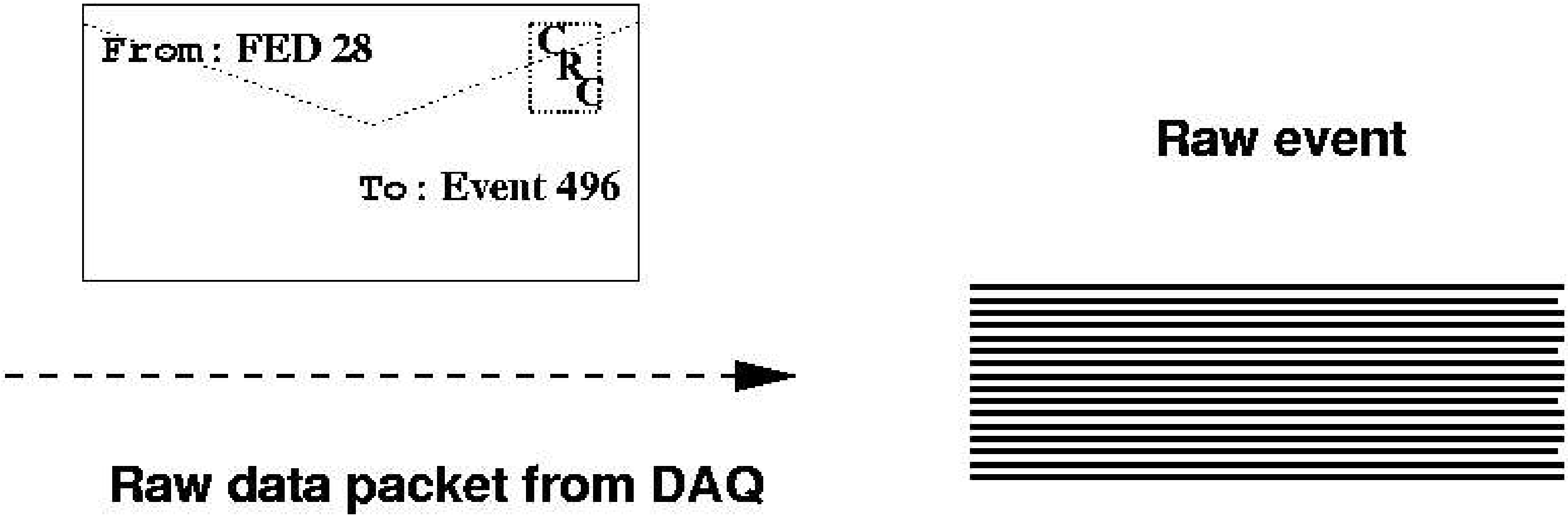
Raw data

- Raw data formation is not reconstruction
- For the purpose of on-line reconstruction DAQ is like the post: the front ends send packets…



Raw event

- … And the event builder receives them
- The raw event is just a collection of raw data packets
- All packets must be present



Scope of HLT reconstruction

- High-level trigger reconstruction starts from unpacking of raw data packets and "objectifying" the channel data (so-called *Digis*)
  - This can take a significant part of the high level trigger time!
  - Everything that happends before is "for free"
- HLT reconstruction ends when the trigger decision is reached
  - Everything that comes after, like on-line classification of the accepted events, should not be "charged" to HLT: it can be done off-line if necessary

Differences with off- line

- Correctness requirement : bugs in HLT selection cannot be fixed “later”: the events are lost
- Time requirement: must complete on average in 10 (Atlas) – 50 (CMS) ms (but some events may take much longer)
- Main conceptual difference: it's not the reconstruction of accepted events that matters, it's the rejected events one!
  - Rejection factor is about 1000 in Atlas and CMS
  - Optimizing the reconstruction of the signal will not improve the CPU time significantly
  - Accepted events will be reconstructed much better off-line
  - The lectures should be called “On-line event rejection”

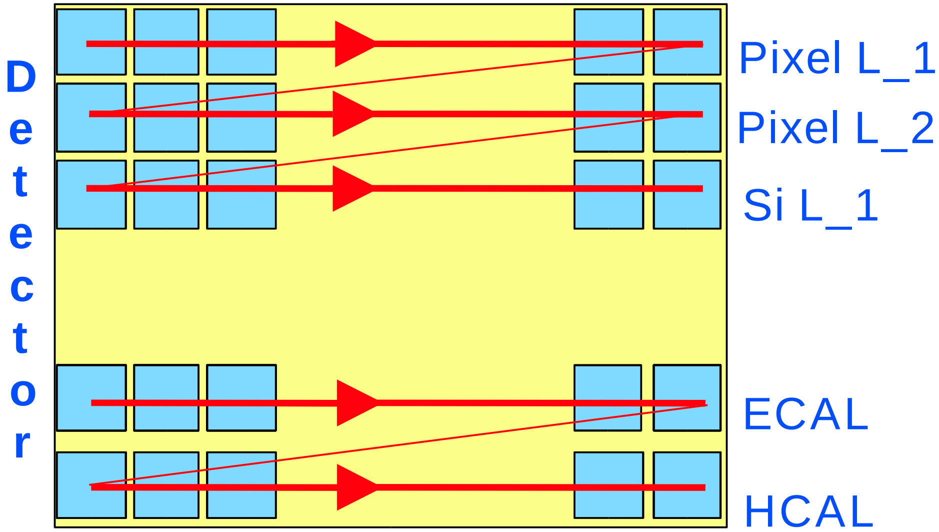
Architectural differences with off- line

- The hardware on which the HLT algorithms will run is (currently planned as) a farm of Linux PC's
- The software framework (at least for Atlas and CMS) is supposed to look the same to algorithms in HLT and off-line
  - The source of raw data changes transparantly
- Algorithms can be freely migrated from off-line to HLT
  - If they satisfy the HLT requirements
  - Some services may be unavailable in the HLT environment
    - calibrations

Seeded reconstruction

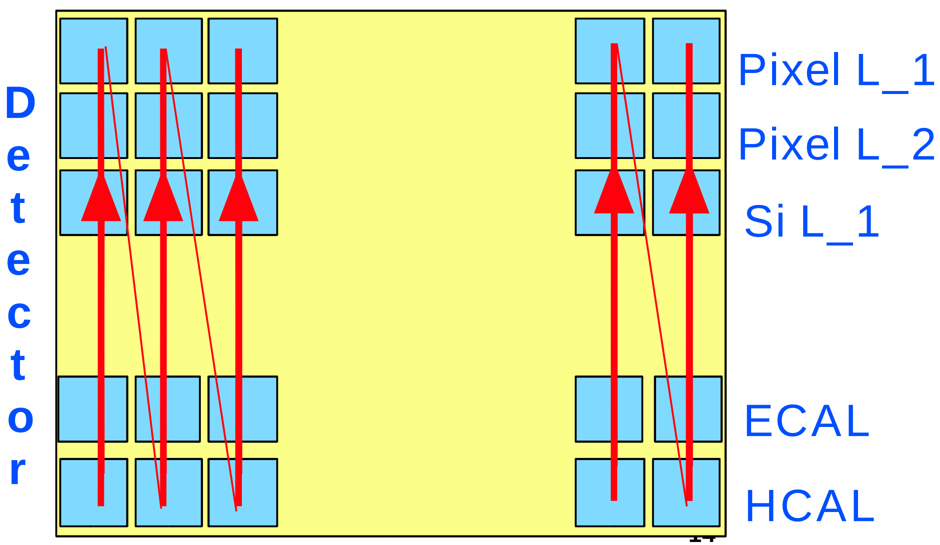
- Another essential difference: the HLT reconstruction is “seeded” by the Level 1 trigger results
  - HLT mostly verifies the L1 decision
  - e.g. If an event was triggered *only* by L1 single muon trigger, then HLT has *only* to reconstruct a muon in the region given by the L1, and with Pt above the threshold.
    - The rest of the event (like calorimeter data) will not be looked at at all: if the muon L2 does not pass, but the event contains a higgs to $\gamma\gamma$ and the di-photon L1 did not trigger on it, the event is as lost as if there was no L1 at all
  - The L1 information is not precise enough to form real seeds, it rather defines “regions of interest” where HLT reconstruction should work
- HLT reconstruction is regional
  - Except for global quantities like missing transverse energy

Regional Reconstruction



Global

- process (e.g. DIGI to RHITs) each detector fully
- then link detectors
- then make physics objects

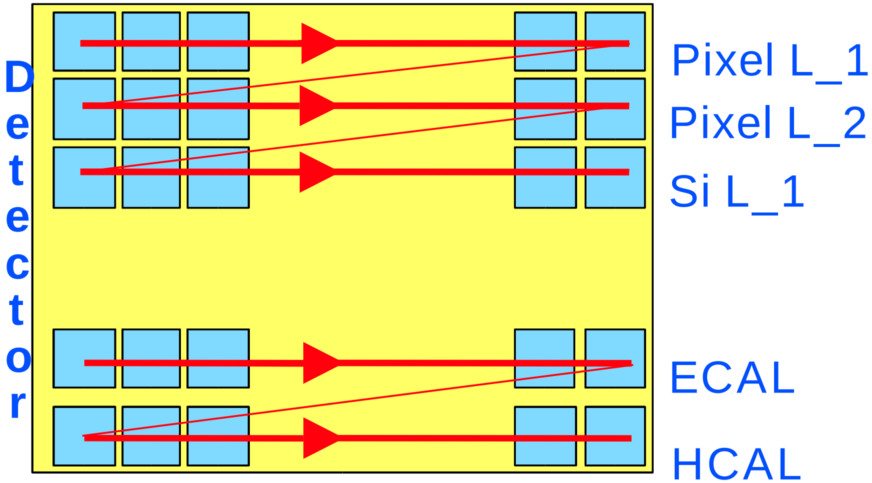


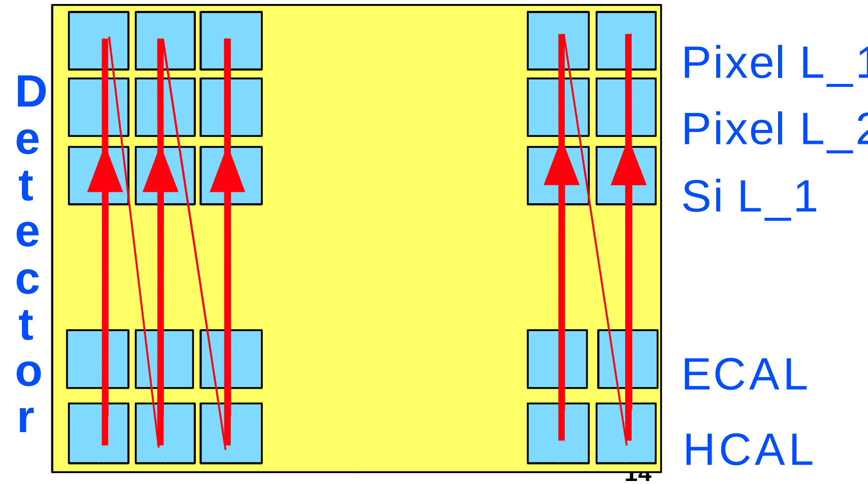
Regional

- process (e.g. DIGI to RHITs) each detector on a "need" basis
- link detectors as one goes along
- physics objects: same

Regional Reconstruction

Regional rather than Global reconstruction





- Slices must be of appropriate size
- Need to know where to start reconstruction (seed)
  - Seeds from Level-1:
    - e/γ triggers
    - μ triggers
    - Jet triggers

- Seeds ≈ absent:
  - Other side of lepton
  - Global objects (ΣE_T, E_T^{miss})

Partial reconstruction

- The question the HLT reconstruction has to answer is very well defined
  - In the example of an event triggered only by a L1 muon trigger, the question is "is there a muon in the L1 region with Pt above threshold?". This can be decomposed in two:
    - Is there a track in the L1 region with Pt above threshold?
    - If yes, is that track a muon?
- The reconstruction can take whatever shortcuts it chooses to answer the given quiestion
  - As long as the answer is correct

Example of partial reconstruction

- A slightly more complicated case: Applying tracker isolation to the found HLT muon
  - The question: is there a track with Pt above the isolation threshold in a cone of a given size around the reconstructed muon? If yes, **reject** the event
- Global reconstruction:
  - Find all tracks
  - Count the ones in the isolation cone
- Regional reconstruction:
  - find all tracks in the isolation cone with Pt > …
  - Check if more than zero were found
- +Partial reconstruction:
  - Start track finding in the isolation cone with Pt > …
  - Stop as soon as a track is found

- The partial reconstruction case can reduce the CPU time a lot
  - If there are on average 3 tracks in the isolation cone for the background sample, the track reconstruction time gain is a factor of 3
    - Provided the track reconstruction is written in a way that allows this
  - This example concentrates on the events to reject
    - No time gain on the accepted events, but that's OK
- Partial reconstruction can be used at any level
  - Pattern recognition example in lecture 4

Regions from Level 1

- Detectors: The silicon trackers are too slow to read out for L1. So L1 regions come from
  - Muon systems:
    - rough muon direction resolution of about 0.1 – 0.2 radians,
    - rough Pt estimate
  - Electromagnetic calorimeters
    - Rough direction of electron or photon candidate (trigger towers)
    - Rough energy estimate
  - Hadron calorimeters
    - Rough direction of jet
    - Rough energy estimate
    - Special τ jet region (much narrower)

Use of L1 regions

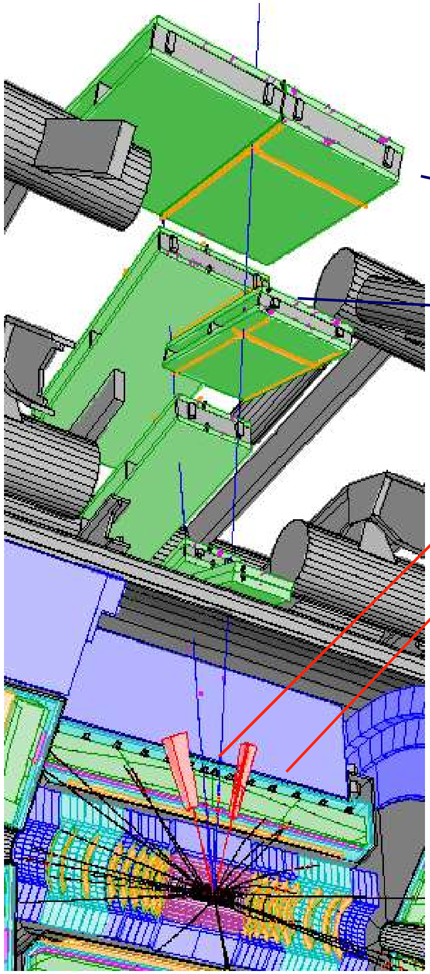
- The L1 regions are used in local reconstruction of the detector system that produced them
  - But can be used directly by another detector system
    - To provide a redundent trigger to check L2 efficiency
    - In case rejection can be achieved faster in this way
  - The result of local reconstruction in L1 regions is more or less Level 2
    - Formally separate from Level 3 in Atlas
    - Naming convention in CMS
  - Level 2 reconstruction also defines regions which are used to seed Level 3 reconstruction
    - Close to off-line position and energy resolution for calorimetric regions
    - Much improved resolution of Muon momenta
  - L2 regions smaller and/or more precise than L1

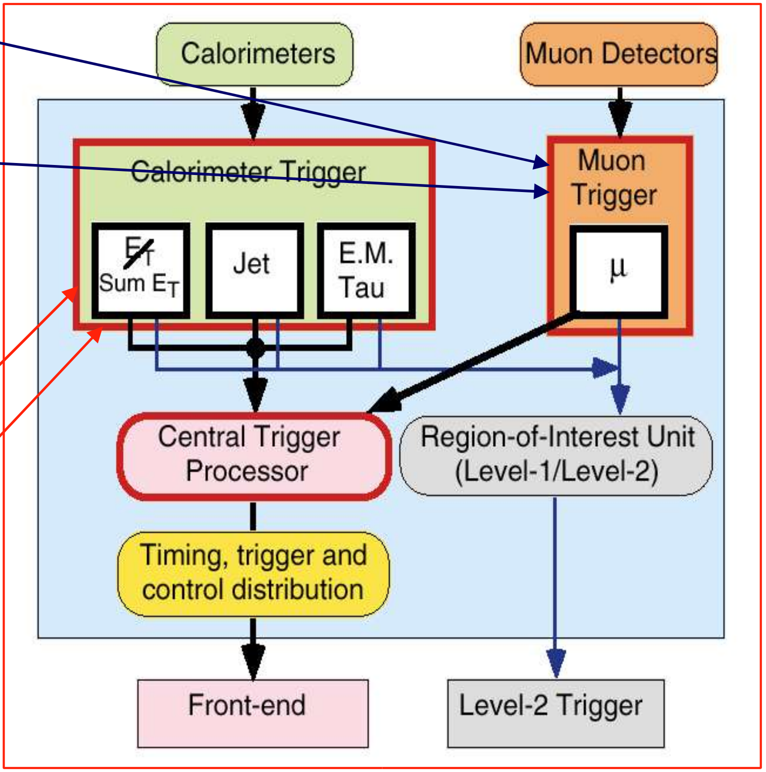
The Region of Interest Strategy for the ATLAS Second Level Trigger

Regions of Interest

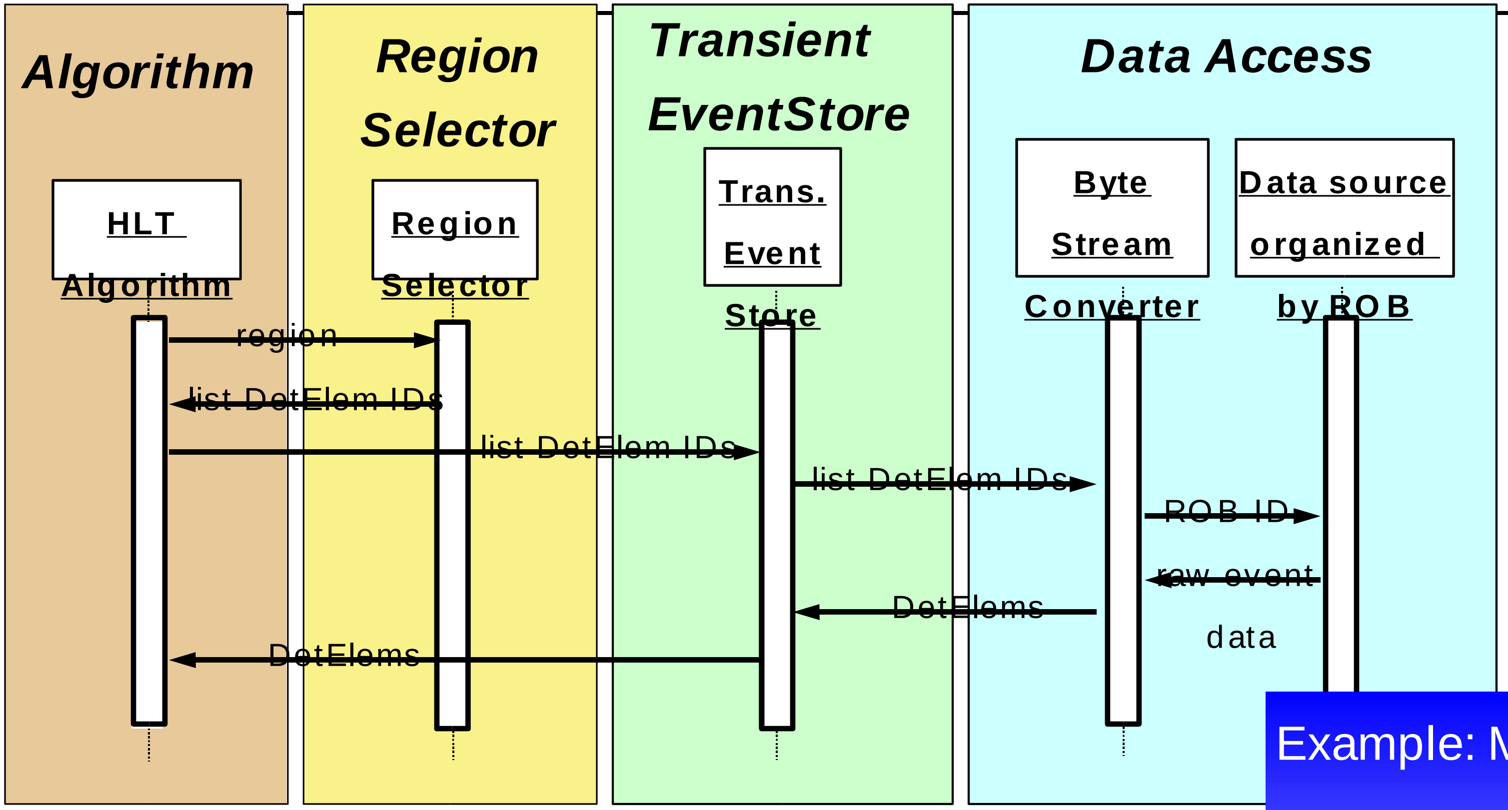
RoI Mechanism:

- Typically a few ROI / event
- Only few % of event data required!
- But more complex system



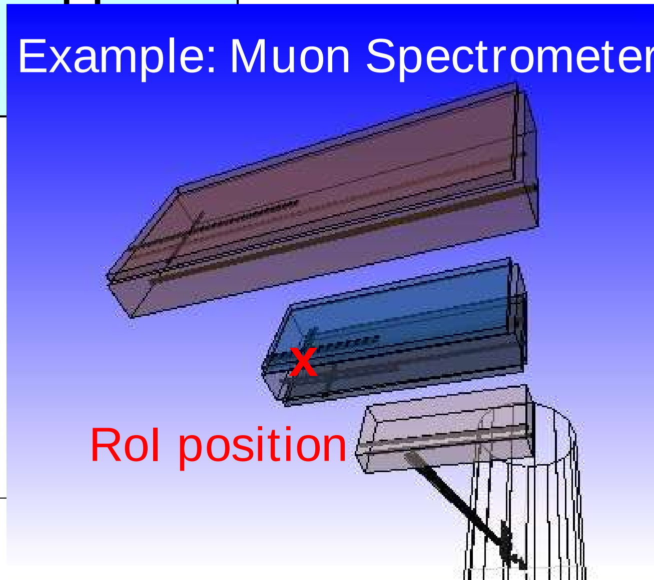


RoI in ATLAS



ByteStreamConverter maps DetectorElements onto minimum readout granularity

Geometry ↔ ROBs



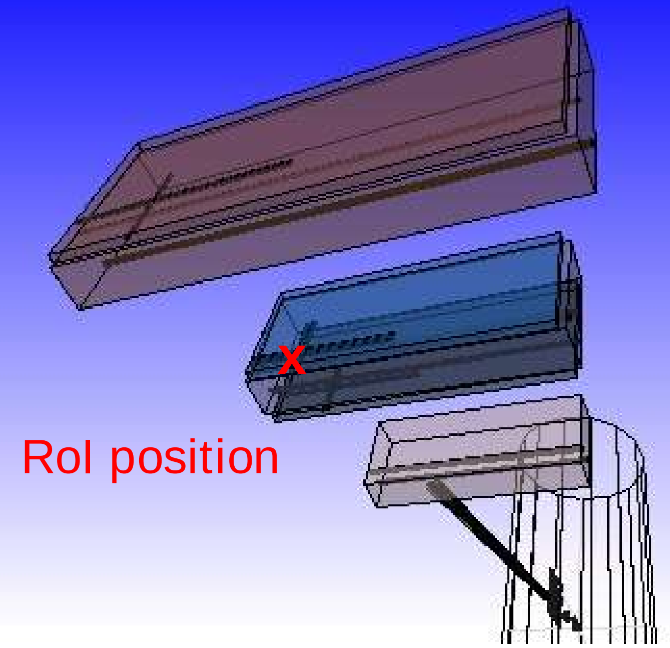
Collections and Identifiers

- Need to optimize collection granularity
  - Trade off between
    - judicious navigation for algorithms
    - Minimize data access requests
- Use of offline identifiers
  - In process of validating design of using “offline” code into “online” environment
    - Facilitate development of algorithms
    - Study boundary between Level2 and EF
    - Allow performance studies for physics analyses
  - Needed for Inner Detector Regions
    - No Level 1 hardware identifier available

The Region Selector Tool

- Algorithms at Level-2 have access to Level-1 RoI regions
  - Tool: Region in detector ⇔ Data collection identifiers
- Requirements
  - Fast!
  - Different geometrical region descriptions
    - Cone
    - Region following a track
    - Etc.
  - Code to be run in multi-threaded environment

Example: Muon Spectrometer



- HLT steering calls an HLT algorithm by passing to it a TriggerElement with a navigable link to a RoI DataObject that contains the RoI position
- The HLT algorithm typically needs to access the raw event data in the DetectorElements in a certain region around the RoI position
- The HLT algorithm requests from the RegionSelector the identifiers of all DetectorElements within that region
- The HLT algorithm passes the list of identifiers to the transient event store
- At this stage 2 cases can occur:
  1. The requested DetectorElements are already available and are just returned
  2. The information first needs to be retrieved from the DataFlow system

     The transient event store invokes a ByteStreamConverter which requests the bytestream data from the correct ROB

     The ByteStreamConverter extracts from the succession of 32-bit words in the ROB bytestream the relevant event data and fills the collections of RDOs

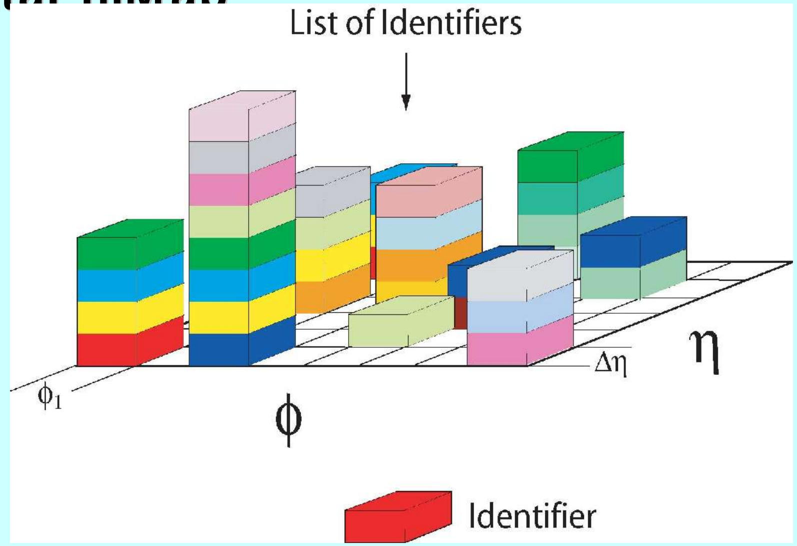
     The thus obtained collections of RDOs are stored in the transient event store

     The extraction of event data can be supplemented by invoking additional data preparation algorithms (e.g. cluster finders for the Cal)

Implementation: The RegionSelector

- At initialization
  - Fill 2 maps for each layer
    - Map < ϕ, set <identifiers>>
    - Map < identifier, vector<η_{min}, η_{max}>>
- At Execution time: we want identifiers within < ϕ_{min}, ϕ_{max}> and <η_{min}, η_{max}>
  - Loop over the layers
  - Loop over ϕ, get identifiers
  - Loop over the identifiers and check η interval
  - Return list of identifiers
  - No duplicate identifier ensured

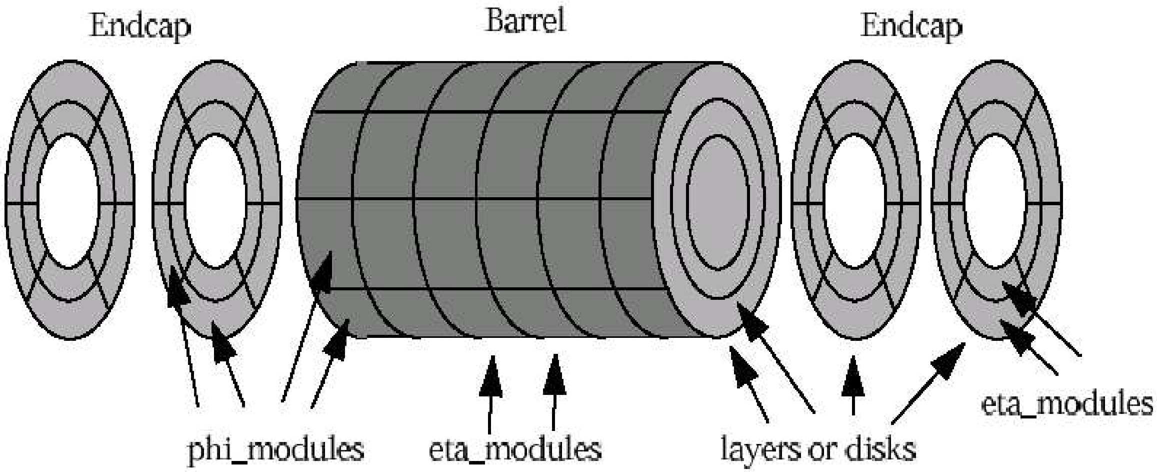
EtaPhiMap

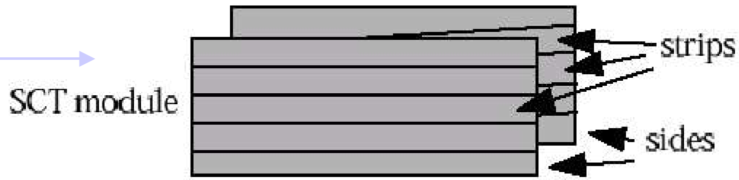


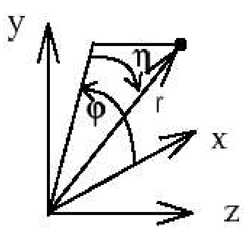
Implementation: data access granularity

	Collection	Number	Number of ROBs
Pixel	module	1744	81
SCT	side of module	8176	256
TRT	straw layer	19008	256
LAr	Trigger Tower	7168	768
Tile	module	256	32
Muon MDT	chamber	1168	192
Muon CSC	chamber	32	32
Muon RPC	chamber	574	32
Muon TGC	chamber	1584	32

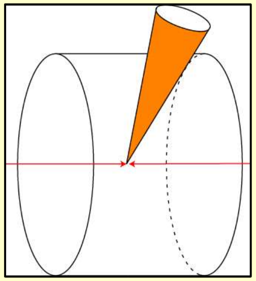
Preliminary







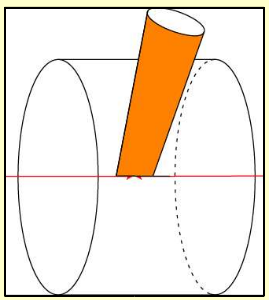
Implementation: Geometrical Regions



CONE

- **Simple: origin +** η, $\Delta\eta$, ϕ, $\Delta\phi$

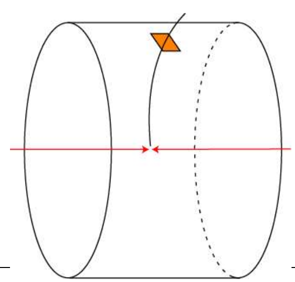
Implemented for outer most detectors TRT, LAr, Tile and Muon



WEDGE OF CHEESE

- **Complicated: origin, *z*, *r***
- **Realistic to account for LHC beam spread in *z***

Implemented for inner most detectors Pixel and SCT



HELICAL ROAD

- **Complicated: origin, r (min), r (max), tan , (min), (max), (min), (max)**
- **Trajectory-oriented (for uniform B field)**

Future development

Timing Measurements

Using 1 GHz Pentium 3

Numbers in ms

Preliminary

Improvements forthcoming

4 GHz scaled (div 3)

ΔΦ	Δη	0.1	0.2	0.5	0.1
	Pixel	0.2	0.2	0.2	0.067
	SCT	0.6	0.6	0.6	0.200
	TRT	0.7	0.7	0.7	0.233
0.1	LAr	0.04	0.04	0.04	0.013
	Tile	0.04	0.04	0.04	0.013
	MDT	1	1	1	0.333
	RPC	0.034	0.034	0.034	0.011
	Pixel	0.3	0.3	0.3	0.100
	SCT	0.9	0.9	1	0.300
	TRT	1.1	1.1	1.1	0.367
0.2	LAr	0.05	0.05	0.05	0.017
	Tile	0.05	0.05	0.05	0.017
	MDT	1.2	1.2	1.2	0.400
	RPC	0.054	0.054	0.054	0.018
	Pixel	0.6	0.6	0.6	0.200
	SCT	1.9	1.9	2	0.633
	TRT	2.4	2.4	2.4	0.800
0.5	LAr	0.07	0.07	0.07	0.023
	Tile	0.09	0.09	0.09	0.030
	MDT	1.5	1.5	1.5	0.500
	RPC	0.194	0.194	0.194	0.065

Timing Measurements

- In Process of timing algorithms as well
  - Offline way
  - Fully timing instrumented TestBeds
- Running a Level-2 LAr Calorimeter algorithm making use of Region Selector
  - Preliminary numbers: O(1 ms)
  - RegionSelector: 0.04-0.07 ms
    - Small fraction of the algorithm time!
    - << ~ 10 ms

Atlas Summary

- The ATLAS High Level Trigger relies on the Region of Interest Mechanism to access only relevant part of the detector
- The RegionSelector is the software tool used to extract the needed data collection identifiers based on some arbitrary geometrical regions
- The timing measurements show a very minimal overhead on the algorithms coming from this tool
- The RegionSelector tool will also be used for offline reconstruction of the data

CMS use of regions

- In CMS, an event is passed to the HLT “filter unit” only after all the event raw data has been received from the DAQ
  – The price to pay: an expensive switch
  – The benefit: no need to implement RoI access infrastructure
- The granularity of raw data is a ReadOutUnit
  – Corresponds to 1 or 2 front end drives
  – Partitioning defined more by ease of cabling than by reconstruction regionality
  – 512 ROU's in full DAQ, so granularity still small

Action on demand

- Regional access to raw data in CMS is based on Action on Demand (also known as “lazy evaluation”) with caching of results
- Guiding principle of CMS software
- Don't compute anything which is not explicitly requested
- When something is explicitly requested, compute it and cache the result
- If the same thing is requested again, serve it from the cache

Sequence of actions

- Event loop at LEP is hard coded in FORTRAN
  - Read all raw data and fill zebra banks
  - Call local reconstruction for all detectors
  - Call global reconstruction for all detectors

individual stages can be skipped (title steering), but the sequence is fixed

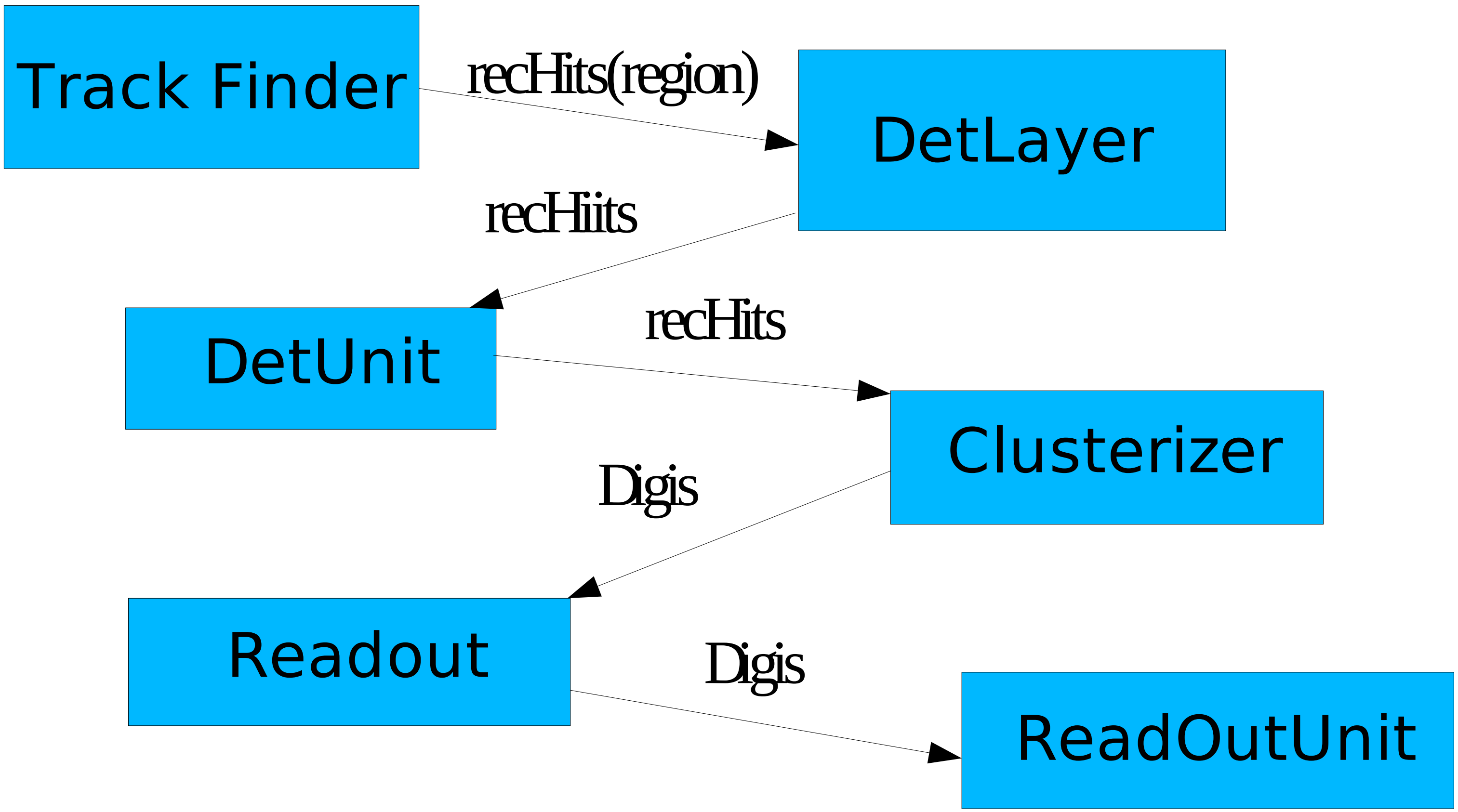
- CMS event loop is
  - Dispatch event to all registered lazy observers
  - Dispatch event to all active observers

A lazy observer provides a service (RecHits, Tracks...)

An active observer acts actively (e.g. HLT selection or user analysis)

Reconstruction on demand chain

- When a track finder is asked for tracks…



Benefits of lazyness

- The sequence of observer registration does not matter
- There is no need to steer the sequence of operations, since every result is guaranteed to be available when requested
- No unnecessary computations or data access
- In the case of HLT, automatically provides
  - Regional raw data access and unpacking
  - sub-regional local DetUnit level reconstruction
    - Not all DetUnits in a region will have track candidates pointing to them
  - Regional subsystem level and “global” reconstruction