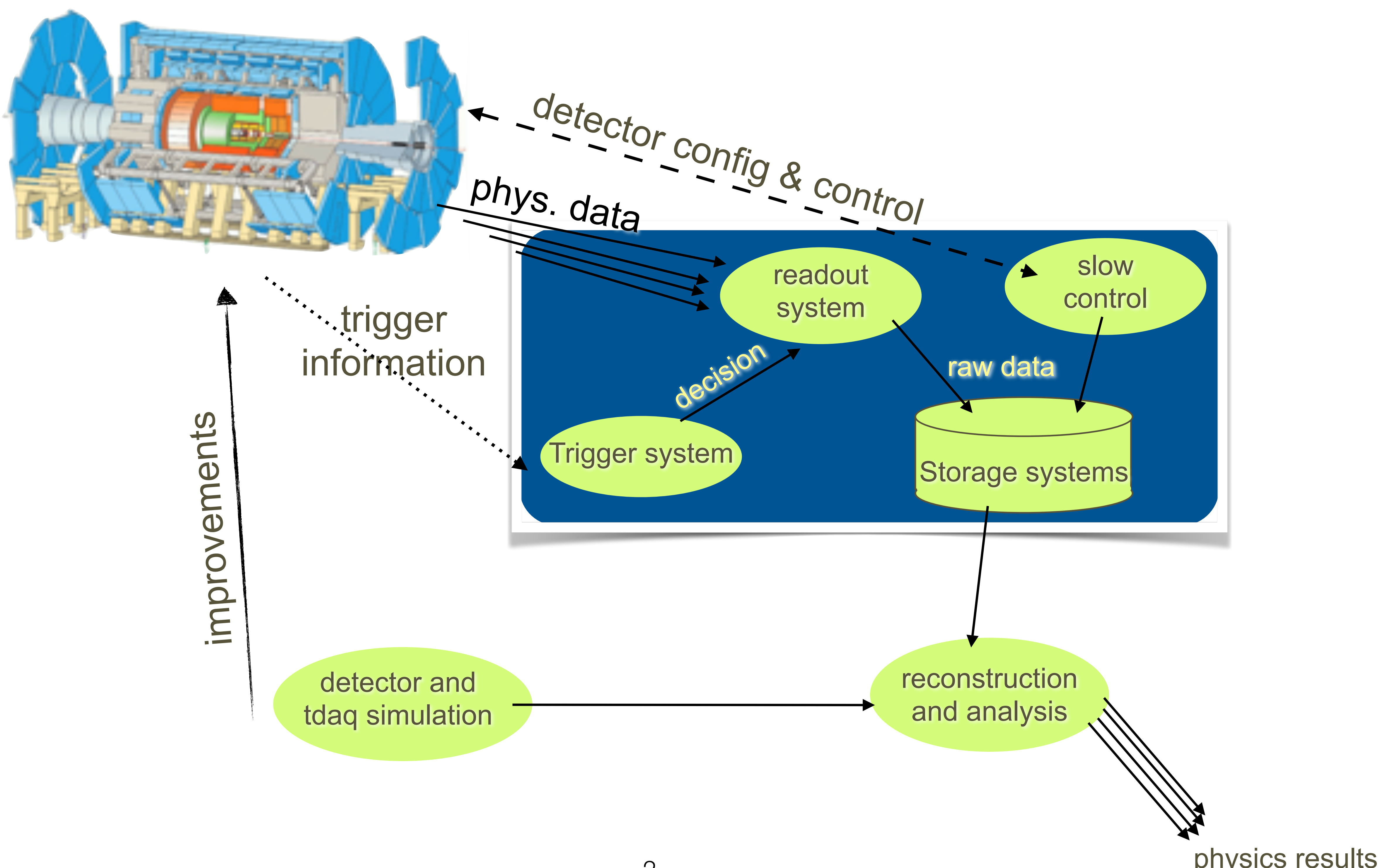
ISOTDAQ 2022

Introduction to Trigger

Gökhan ÜNEL / UCI

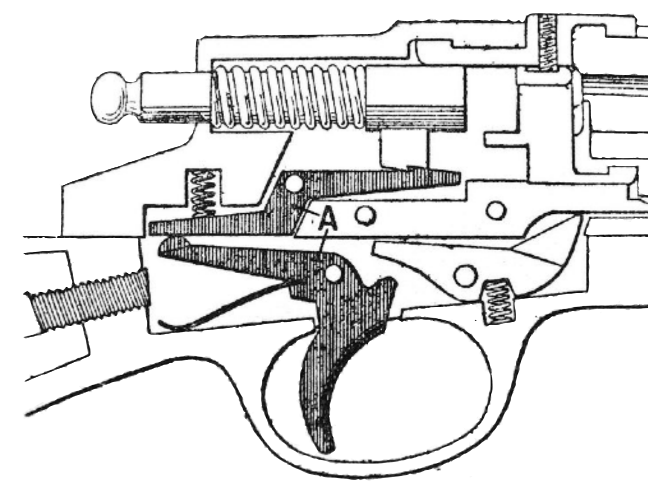
the big picture

trigger and data acquisition



Nomenclature

- Detector
  - ➡ Device to capture events
  - ➡ Event: something interesting like a p-p collision



- Triggering
  - ➡ Quick reaction to the event in question,
    - ➡ "low latency" & "high efficiency"
  - ➡ Quick execution of tasks following the occurence
    - ➡ "low deadtime"
  - ➡ Allow for errors
    - ➡ "redundant"
  - ➡ Allow for quick modification while running
    - ➡ "flexible"
  - ➡ Dont spend all the money on TRG
    - ➡ "affordable"

- Data acquisition
  - ➡ Record the data from the events

Triggering (TRG)

* Fast and reduced information!
  * did it pass?
  * did it happen?
  * is it bigger than x?
* How?
  * mostly using dedicated equipment (HW trigger)
  * sometimes with additional help with dedicated software (SW trigger)
* What?
  * if … then, …
    * If a particle passe, go and read the modules.
* Trigger types
  * Repetitive: write data every N seconds
  * Random: write data at random to check
  * Self: delay signals & decide to write or not
  * …

Modern Particle Physics Experiments

- **Data Size**
  - few MB/event
  - summing of few 100 M channels
- **Data Rate**
  - About 40 M evts /s at 13 TeV (LHC)
  - About 1 evts /100years /km at EeV (PAO)
- **Bandwith**
  - 40MHz x 1 MB = 40TB/s
    - too much data! can not write them all.
  - select and record only the most important events

exa	E	10^{18}
peta	P	10^{15}
tera	T	10^{12}
giga	G	10^{9}
mega	M	10^{6}
kilo	k	10^{3}

TRIGGER

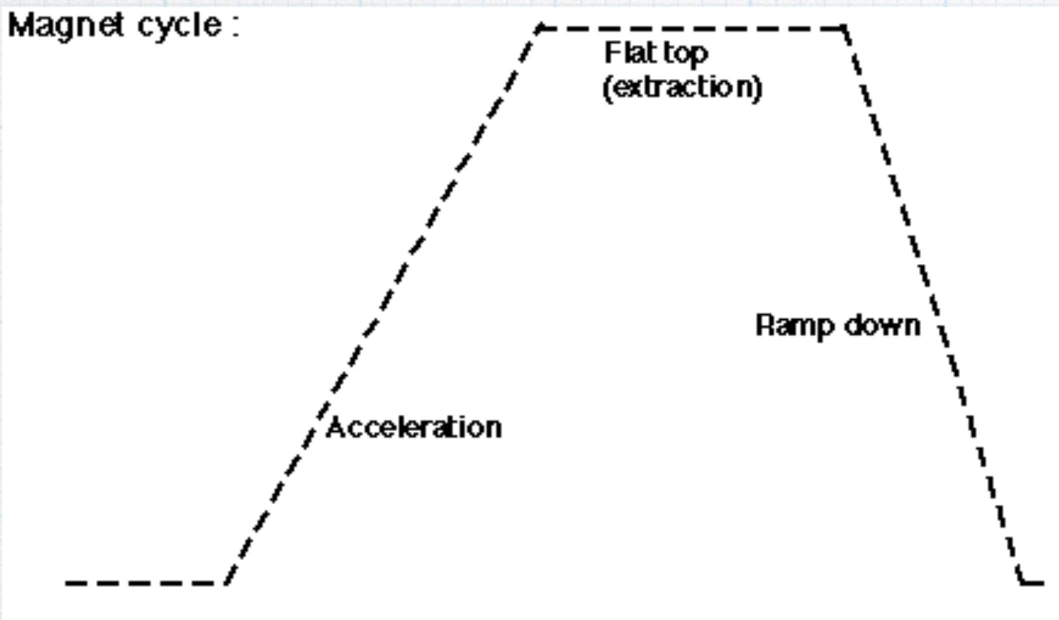
* select and record only the most interesting events
  * decide as fast as possible, in **real-time**
    * keep promising candidates, remove clutter

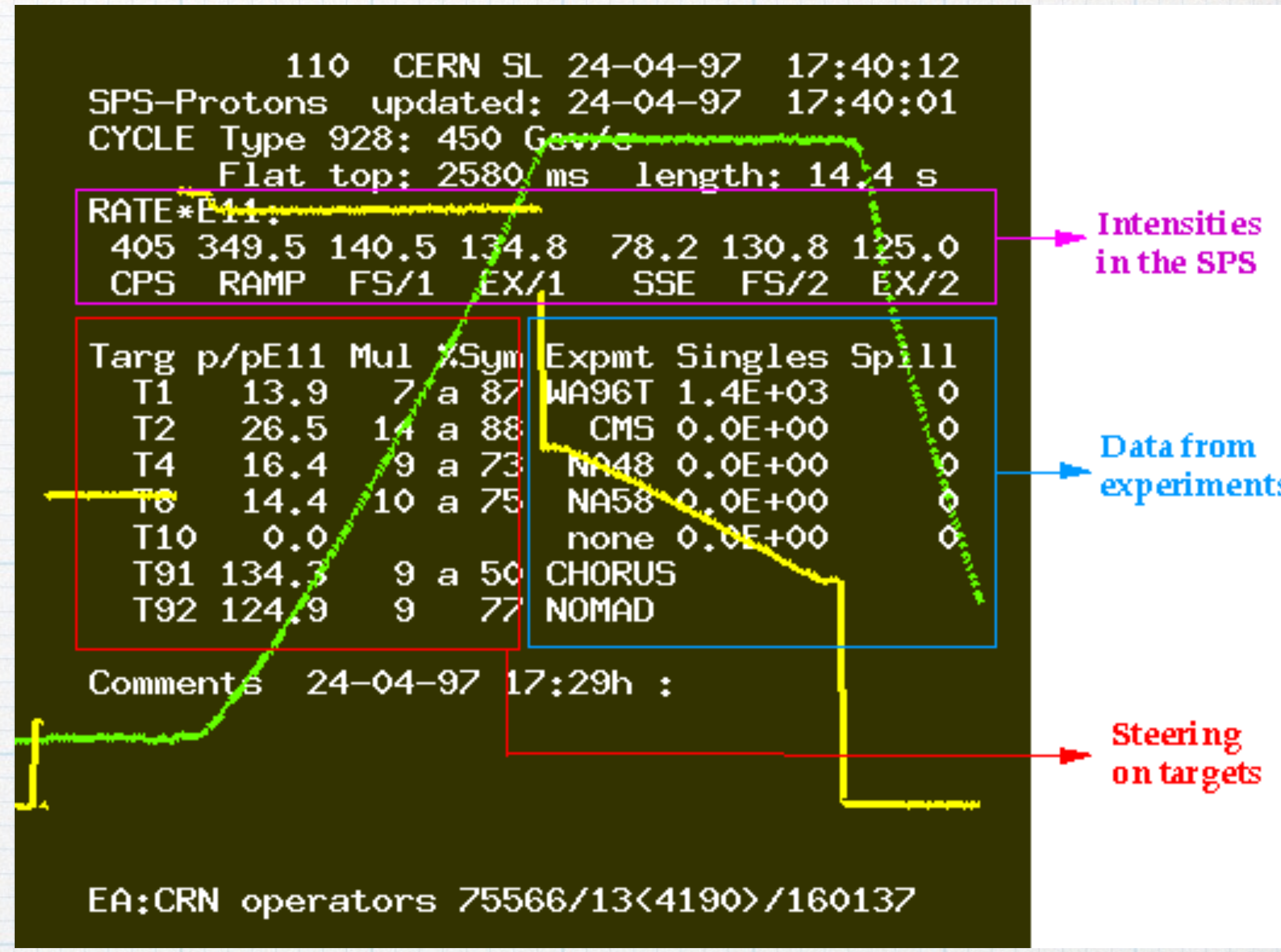
with a limited budget

* But pay attention to
  * do not introduce any bias
  * good sensitivity (do not miss)
  * good synchonization (recorded and triggered events should match)
  * good rejection (dont accept by mistake)
  * do some monitoring (keep an eye on the system)

some concepts

* **Fixed target experiment**
  * spill, flat top, extraction, break





* **Collider experiment**
  * Bunch Crossing (**BC**)
  * compromise between high bg rejection and efficient selection
    * reject bg events to match the DAQ capacity (LHC: GHz to ~ 100 Hz)
    * should understand the background completely to reject correctly
  * different event types: Pile-up, Minimum Bias, physics, QCD...

fixed target point of view

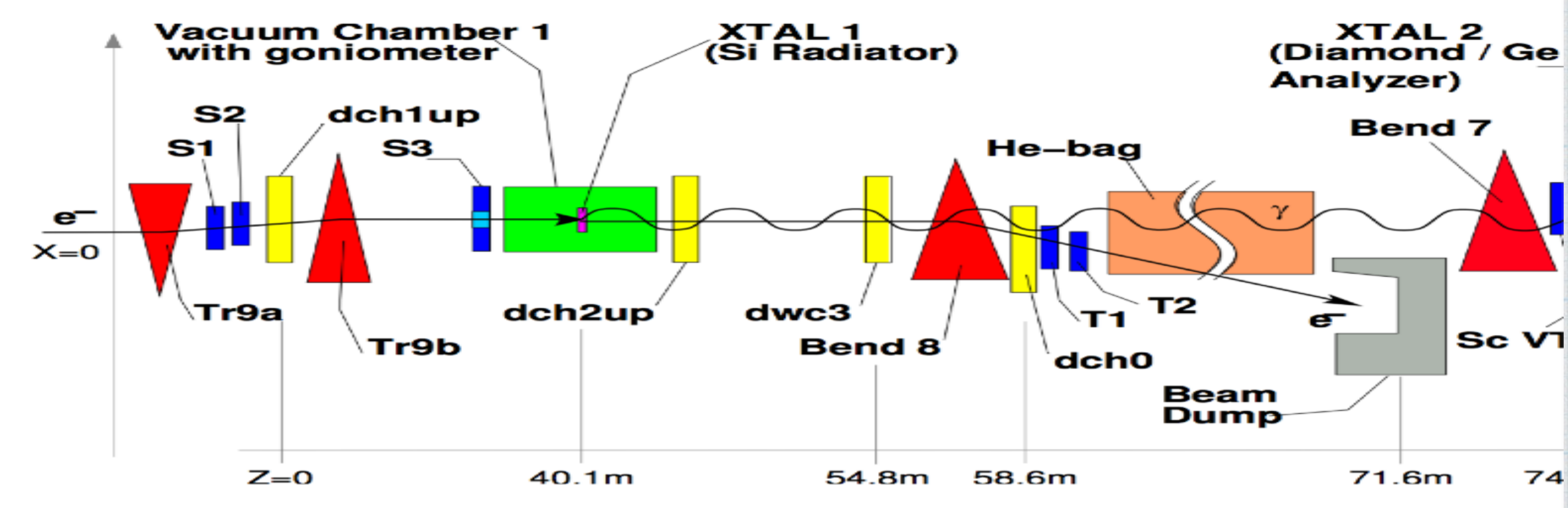
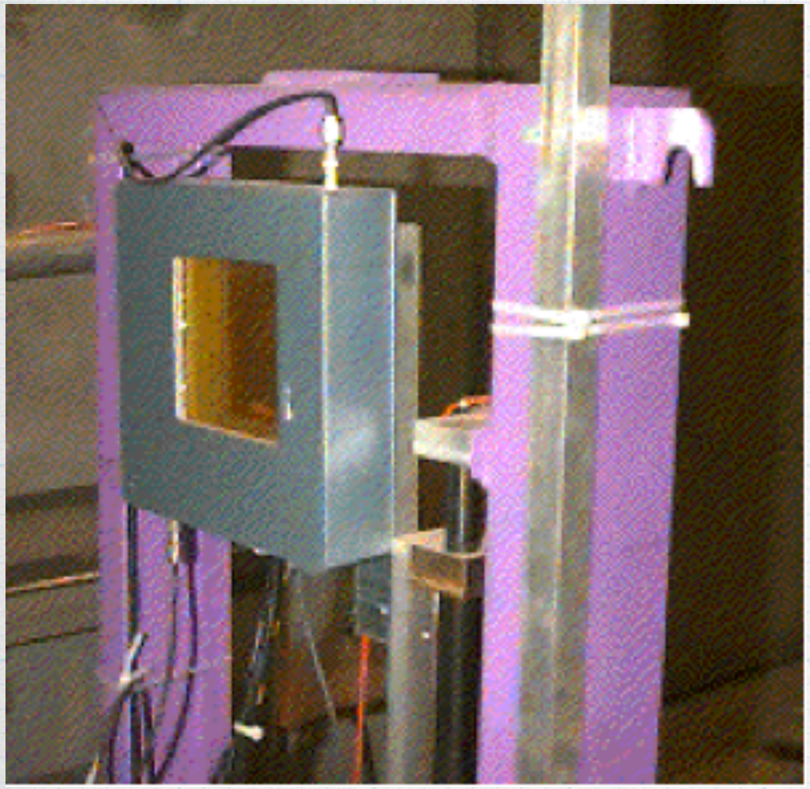
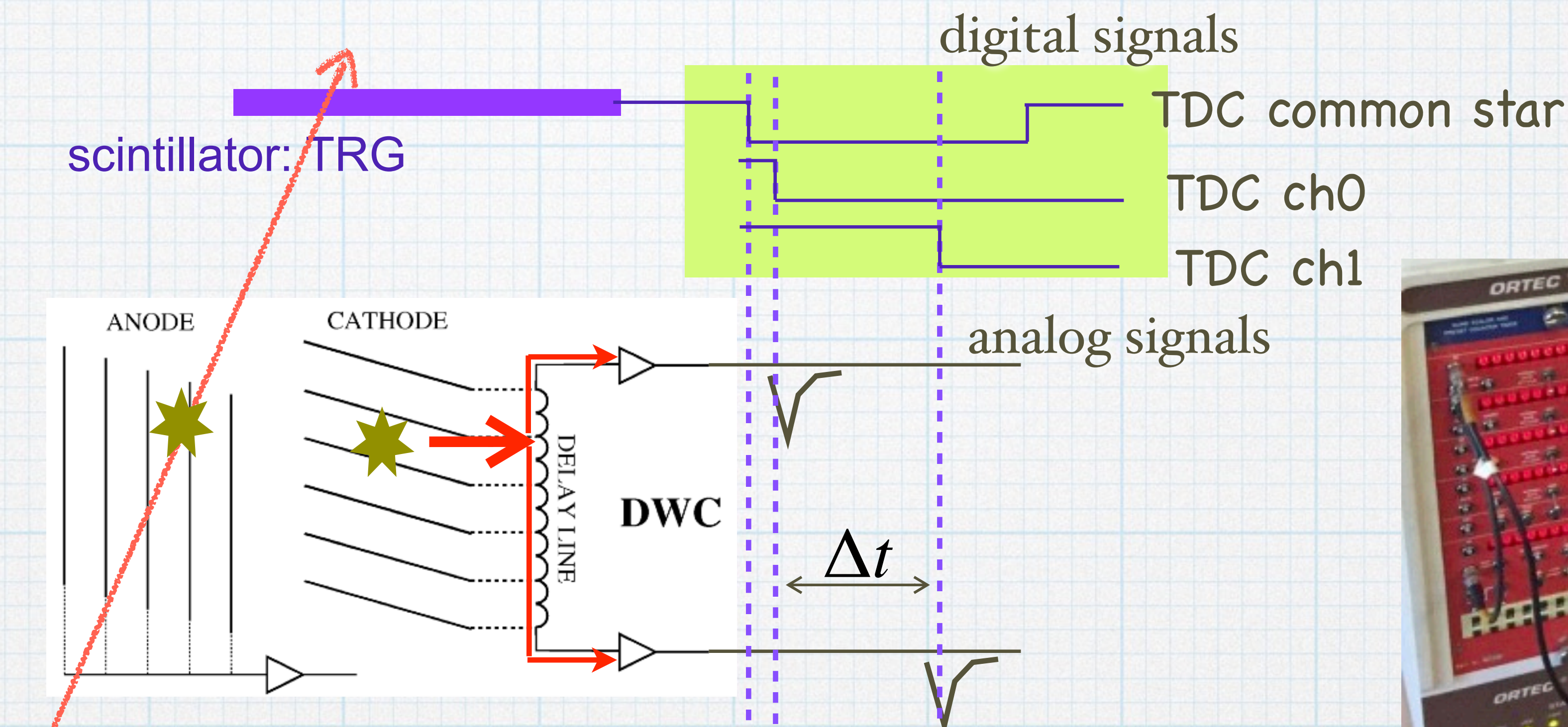


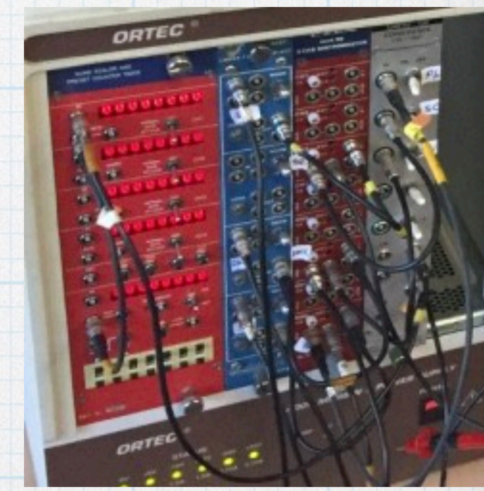
Fig. 1. Setup of the Na59 Experiment

* S: a device that gives T/F information
  * will setup logic statements to
* classify and count event types
  * N1= s1 & s2 & !s3 ---> an electron is coming and it is not away from the central axis
  * N2= N1.(T1 || T2) ---> the above electron was diverted in B8 magnet

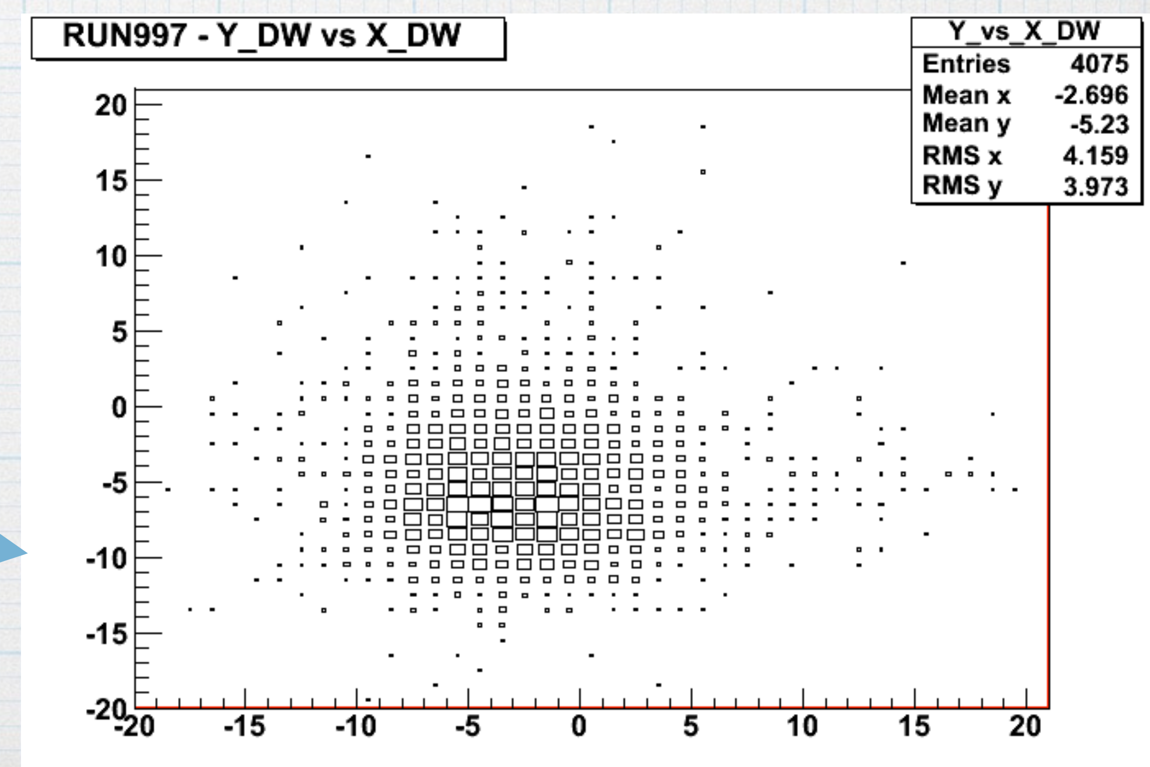
triggering by a scintillator







$$y = \alpha \cdot \Delta t + \beta = \alpha \cdot (t_{top} - t_{bottom}) + \beta$$



- DWC: Delay Wire Chamber
  - ➡ Simple detector to typically measure the beam profile on fixed target experiments.
  - ➡ gaseous & multiwire
  - ➡ TDC readout: 2CH /plane.

collider point of view

* pp / ee collisions yield events according to cross sections
    * Interesting events have very SMALL cross sections: few events
* Find the needle in a Large Haystack Collection
    * Typically 1 in 10^{10} or smaller odds.
    * depends on signal and backgroung cross sections
* Number of events,
  $N = \sigma$ (cross section) x L (luminosity) x ε (efficiencies)
    * L = how long we run the accelerator
    * ε has many components...

Efficiencies

* ε_{geom}
  * Is the event inside the detector volume?
  * any cracks/ un-instrumented areas?
* $\varepsilon_{detector}$
  * Is the relevant sensor working correctly?
  * Is the readout electronics working correctly?
* $\varepsilon_{trigger}$
  * was I supposed to quickly notice the event? did I?

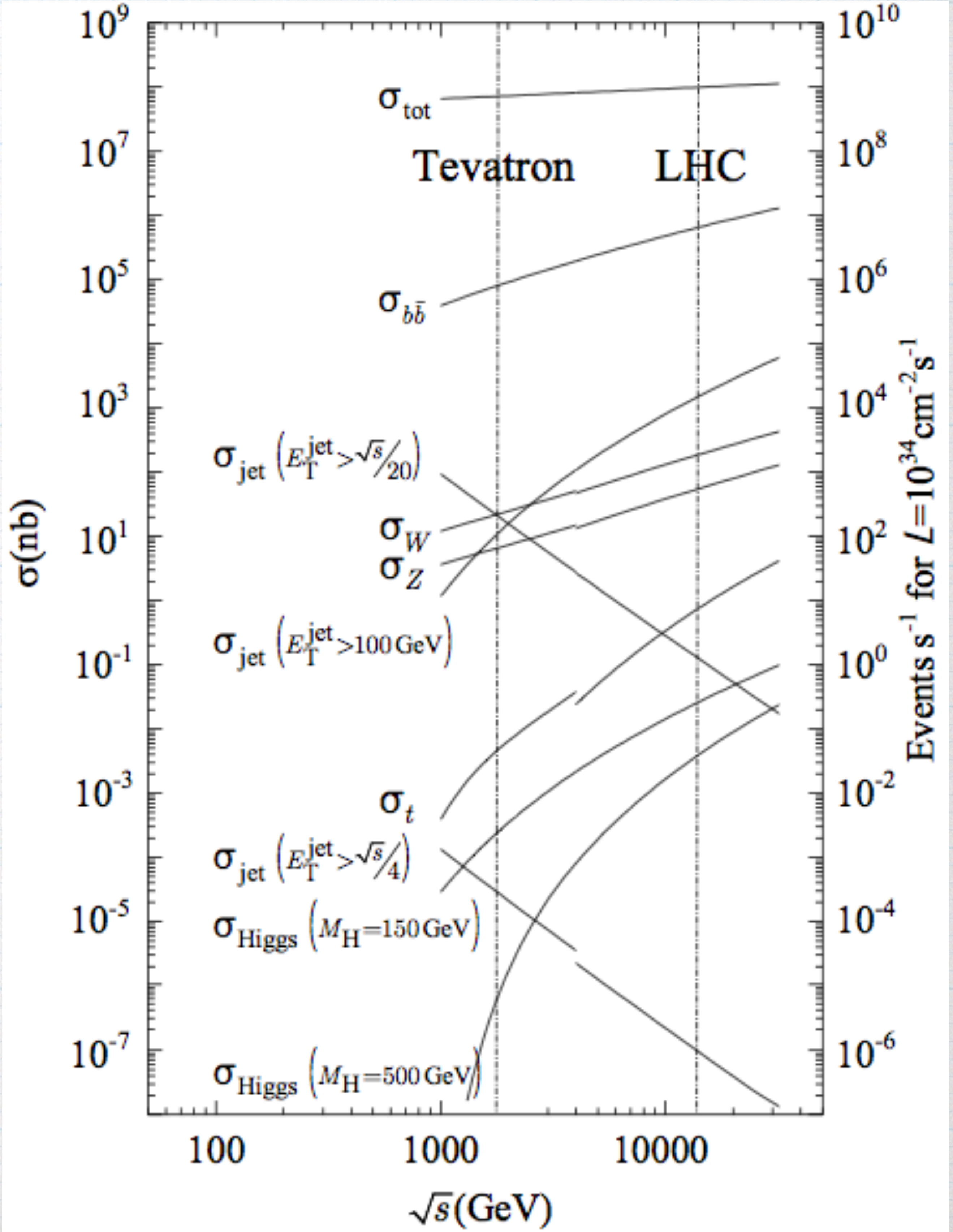
$$\epsilon = \frac{success}{try}$$

Trigger Rates - Dead Time

- Instantaneous Luminosity
  - L = #interactions/cm²/s
- N = σ L
  - both are estimations
- High efficiency needed

$$\epsilon_{TRG} = \frac{\#Accepted}{\#Produced}$$

- TRG decision takes time
  - it contributes to toal DeadTime: use buffers



Busy

* after TRG comes readout
  * reading out electronics, buffers takes time.
  * what if another TRG comes while data is read ?
    * this is bad, should be prevented.
* Flag that system as "busy" before readout
  * this will block futher triggering ("back pressure")
  * Do the reading and clear the flag: "busy logic"
  * for simple systems achived via standard HW modules
* In real life data (events) come at random & data rate fluctuates
  * use buffers to absorb deadtime
    and to minimize busy duration

TRG objects and efficiencies

* We usually trigger on high energy (or pT) objects
  * e, γ, μ, jets, MET, Σ_{ET}
  * or their combinations

* Trigger efficiency calculation options:
  * MC simulations: This many events should have been triggered (try), How many actually were? (success)
  * compare with another looser trigger, do offline analysis
  * Use data, e.g. tag and probe method
    * requires clean sample, known physics and fake information
    * I triggered on this many muons (success), offline analysis tells me they come from Z-bosons which decay into muon pairs, therefore I should have had this many triggers (try).

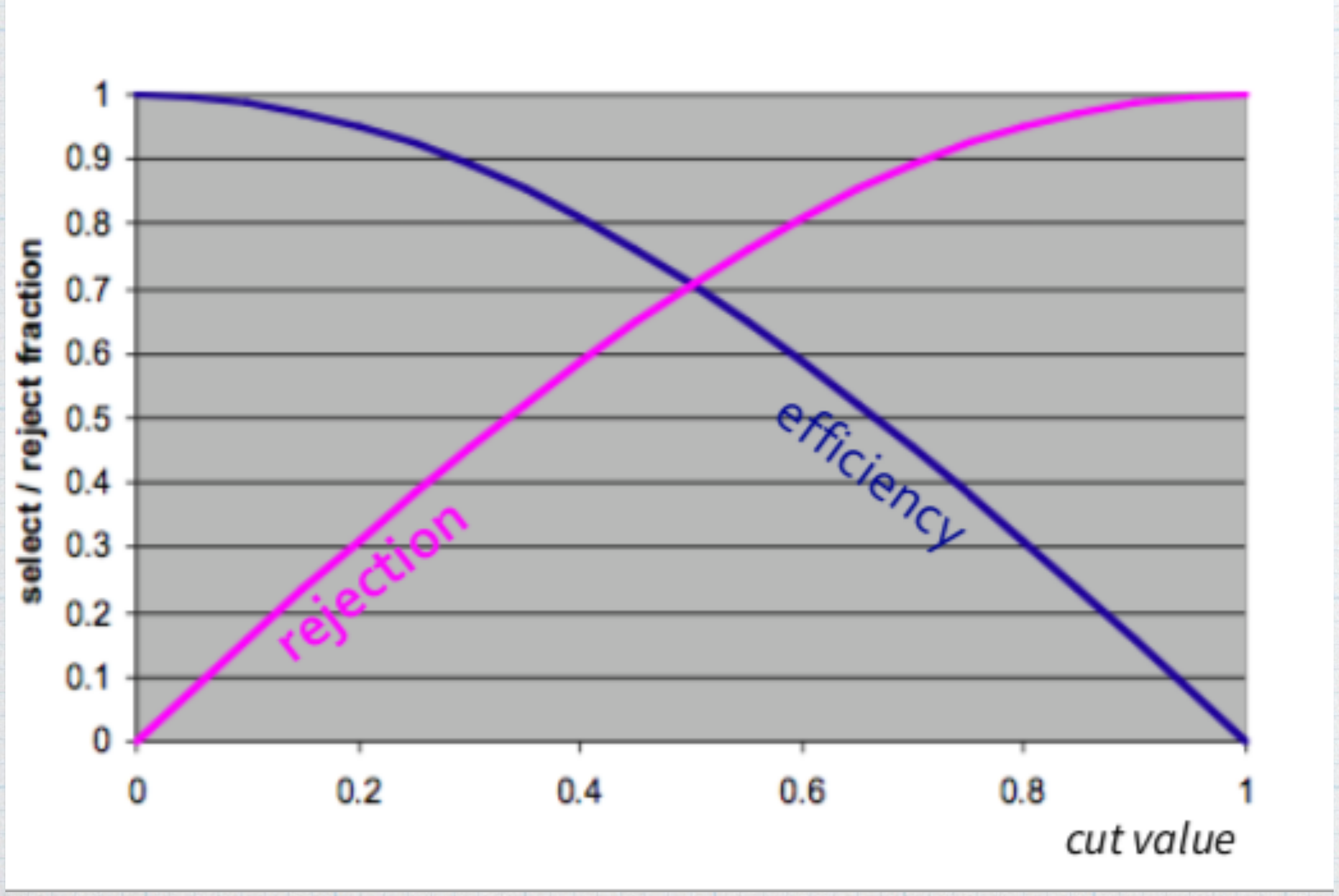
TRG tasks

* Identify interesting events
  * what did happen? what did not happen?
* Reject uninteresting events (as early as possible)
  * Especially for colliders
* Ensure sensitivity to a parameter
  * tunable system to compensate if Nature misbehaves
* Ensure good synchronization
  * multiple sub detectors will be sending data

acceptance vs rejection

* Accept good events
  * but there are some good ones that looked bad in that particular "photo"
* Reject bad events
  * but there are some bad ones that looked good in that particular "photo"

Backgrounds can be reduced, but not suppressed.



Particular needs of the experiment determine the optimal point.

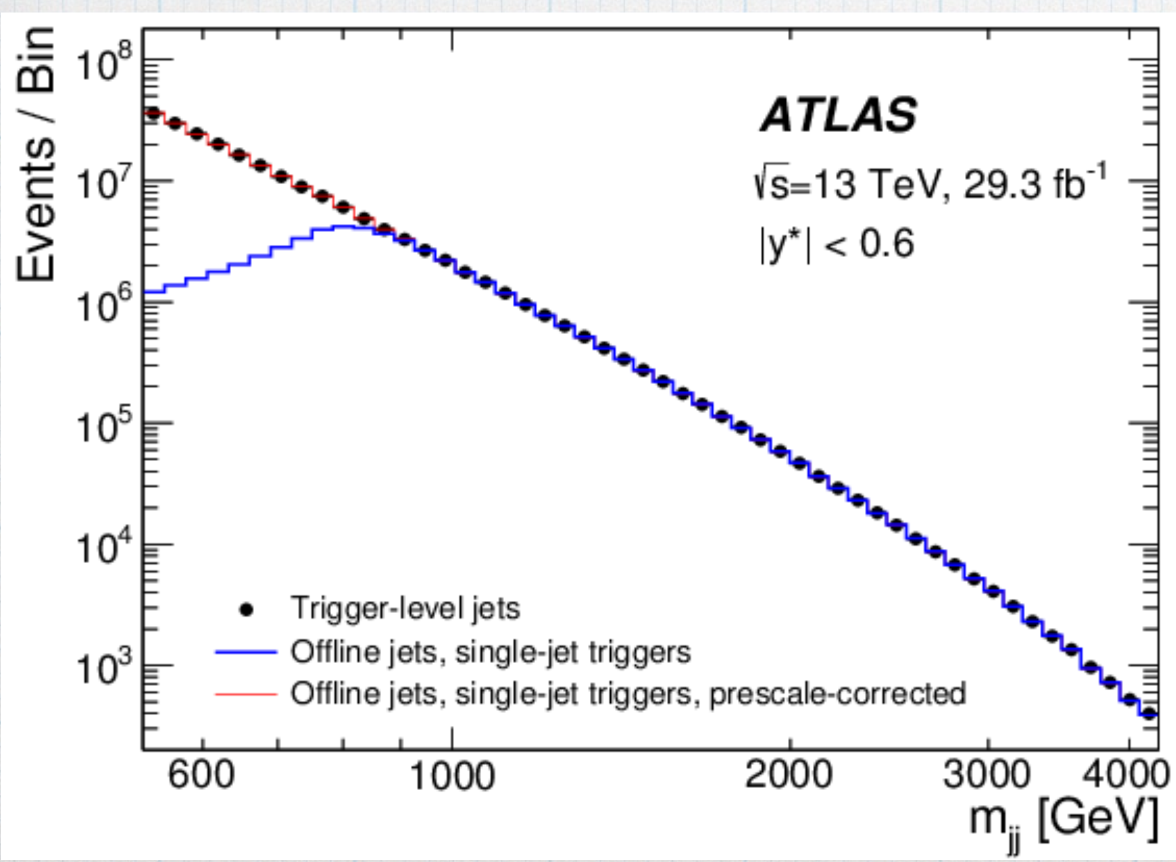
Rely on trigger algorithms
— logic operations
— conditional statement

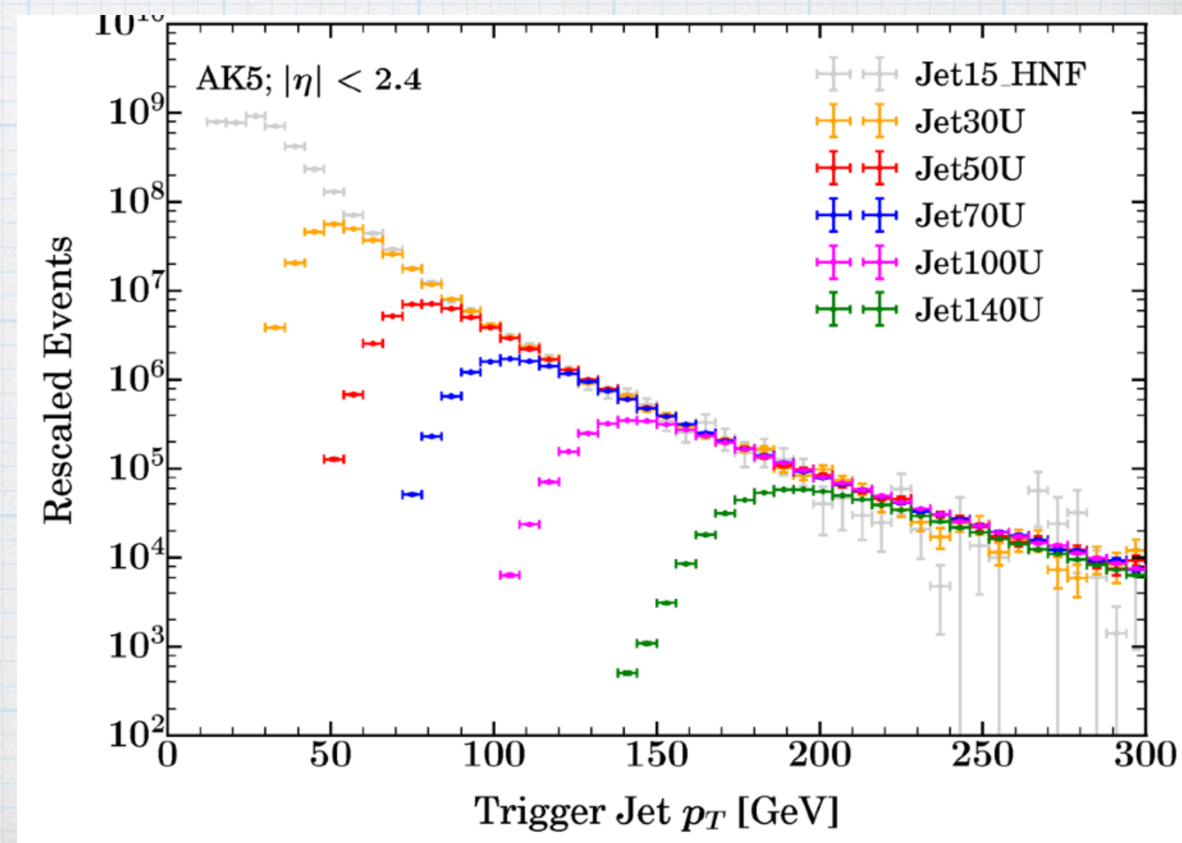
TRG types

- Zero-bias: just the BC event
- Minimum-bias: BC && minimum detector activity
- Physics: desired events, Instrumental and physics backgrounds
- Calibration: Detector and trigger efficiency measurement from data, calibration coefficients, energy scales
- Back-up: to spot problems, provide control samples (often pre-scaled)
- Pre-scaled: mostly known event types, backgrounds

prescaling

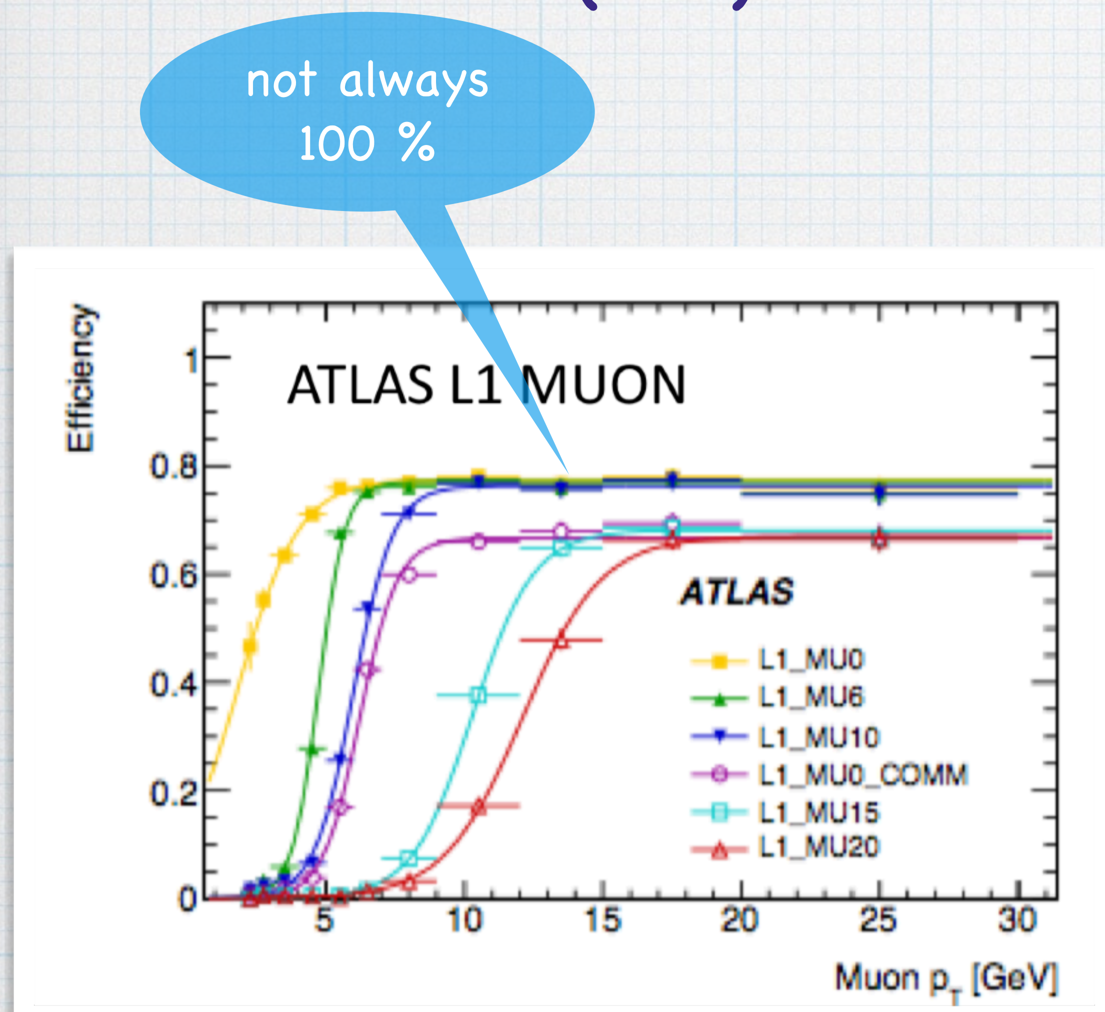
Only a fraction N of the events satisfying the relevant criteria is recorded, where N is a parameter called pre-scale factor. This is useful for collecting samples of high-rate triggers without swamping the DAQ system

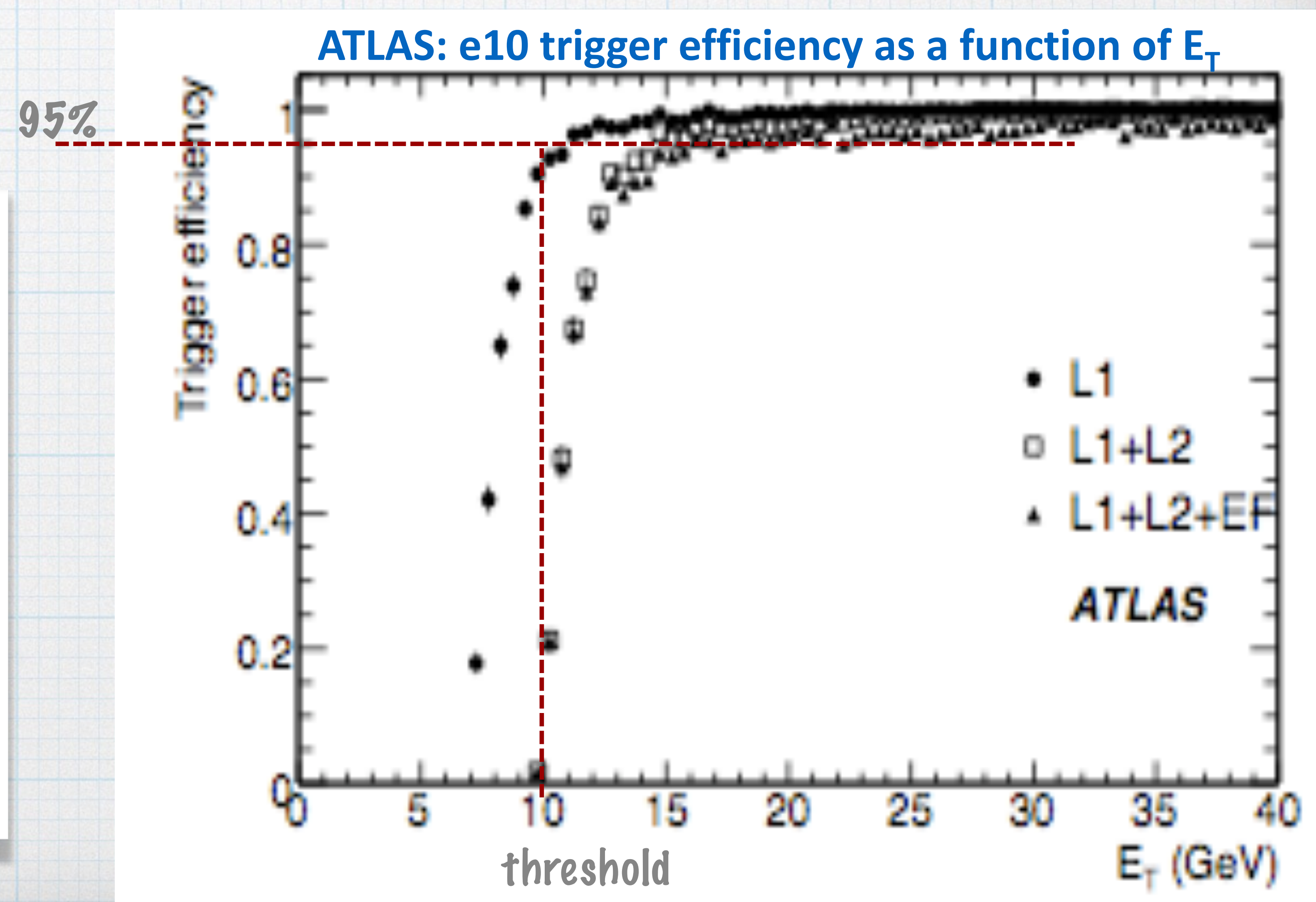




Turn On Curves

- A TRG has to be fast and mostly in accurate due to large measurement error
  - A TRG becomes efficient when the desired events are accepted.
- L1 threshold is chosen so that efficiency is 95% of its max value (MC)





“high” level trigger

* First-level selection must be fast: hardware based
  * Fast custom electronics making simple and fast selection
  * Coarse granularity data from detectors, mostly aimed at lepton identification
    * Hadron and e-m calorimeters for electrons/γ/jets
    * Muon chambers
    * Usually does not need to access data from the inner-tracking detectors (only if the rate can allow it)
* HLT must be selective: software based (PC farms)
  * Can be separated into levels
    * Level-2 accessing only a part of the event
    * Level-3 accessing the full event
  * Full-precision and full-granularity calorimeter information
  * High-precision readout from the muon detectors
  * Fast tracking in the inner detectors (for example to distinguish e/γ)

	No.Lev Trigger	First Level Rate (Hz)	Event Size (Byte)	Readout Bandw.(GB/s)	Filter Out MB/s (Event/s)
ATLAS	3	10^5 L2 10^3	10^6	10	~100 (10^2)
CMS	2	10^5	10^6	100	~100 (10^2)

TRG Levels & signatures

Exp.	No of Levels
ATLAS	3
CMS	2
LHCb	3
ALICE	4

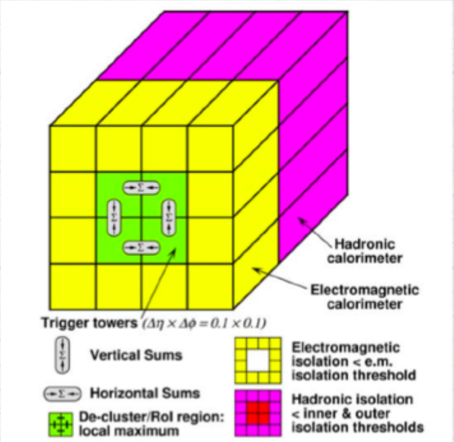
* LHC triggers are implemented in multiple levels
  * high efficiency with large background rejection
* ATLAS has 3 levels
  * L1 = fast (few μs) with limited information, hardware based
  * L2 = moderately fast (~10s to ms), hardware/software
  * L3 = Commercial processor(s)
* Signature=one or more parameters to use for discrimination, able to select "events" of potential interest for your studies
* The signature can be the amplitude of a signal passing a given threshold or a more complex quantity given by software calculation
  * First use intuitive criteria: fast and reliable
* Eventually combine more signals together following a certain trigger logic, giving redundancy.

HLT design ideas

* Early rejection
  * Alternate steps of feature extraction with hypothesis testing: events can be rejected at any step with a complex algorithm scheduling
* Event-level parallelism
  * Process more events in parallel, with multiple processors
  * Multi-processing or multi-threading
  * Queuing of the shared memory buffer within processors
* Algorithms are developed and optimized offline, often software is common between online

Thread	Process
Share memory space	Separate Memory Space
Lightweight	Needs more memory
Memory should be carefully managed	no special attention neded
OS usually needs little help to correctly distribute across multiple CPUs	OS handles the MultiCPU distribution

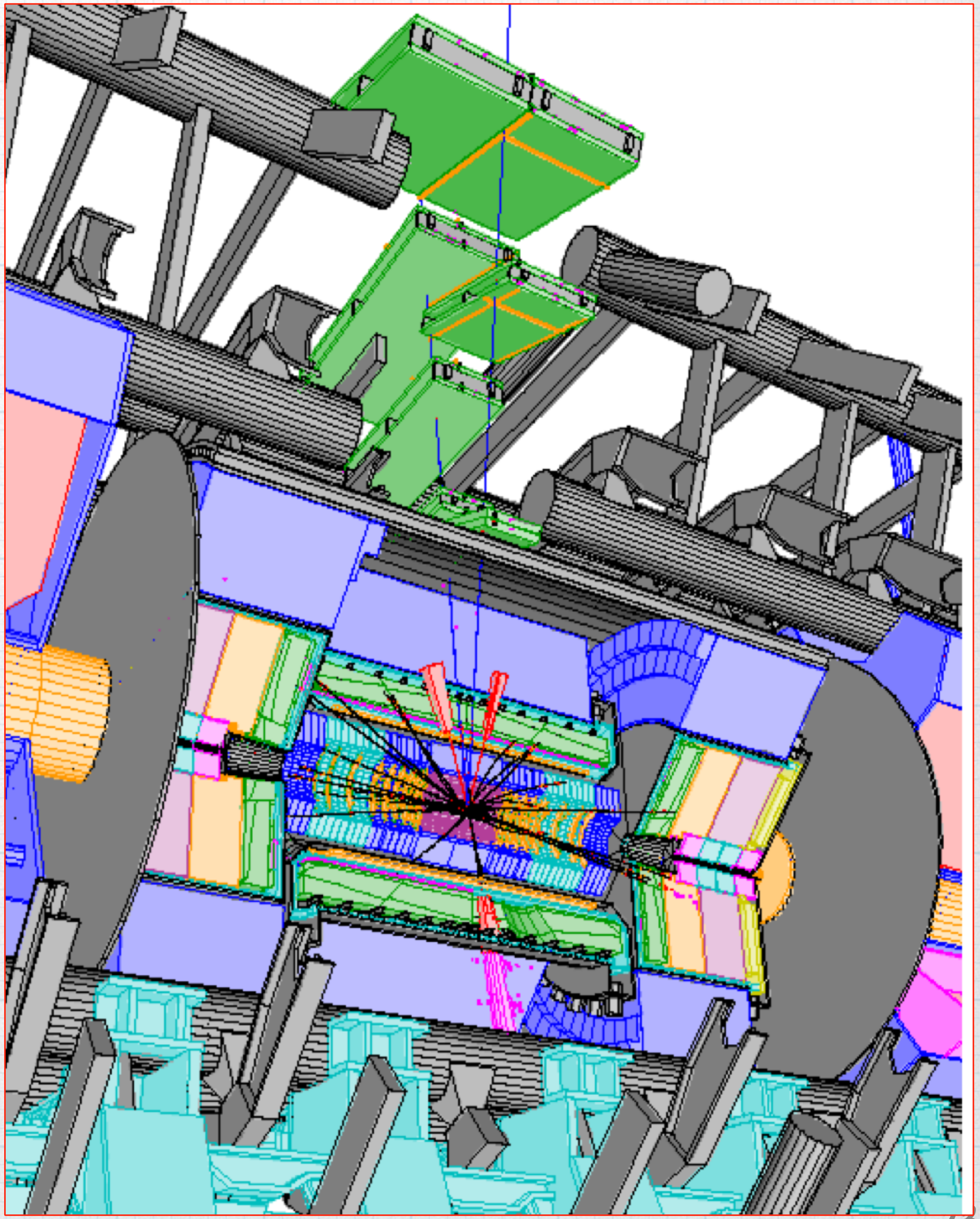
a multi level example



* e, γ, τ, jets, MET, ΣET
  * Various combinations of cluster sums and isolation criteria
* Level-1
  * Dedicated processors apply the algorithms, using programmable ET thresholds
  * Peak finder for BC identification
  * Sliding-window technique to find clusters
* High-Level trigger
  * More topological variables and tracking information for electrons from Inner Detectors
    * Tower clustering at L2
    * Jet algorithms at L3 (Event Filter)
  * Isolation criteria can be imposed to control the rate (reducing jet background at low energies thresholds)

seeded HLT: ATLAS

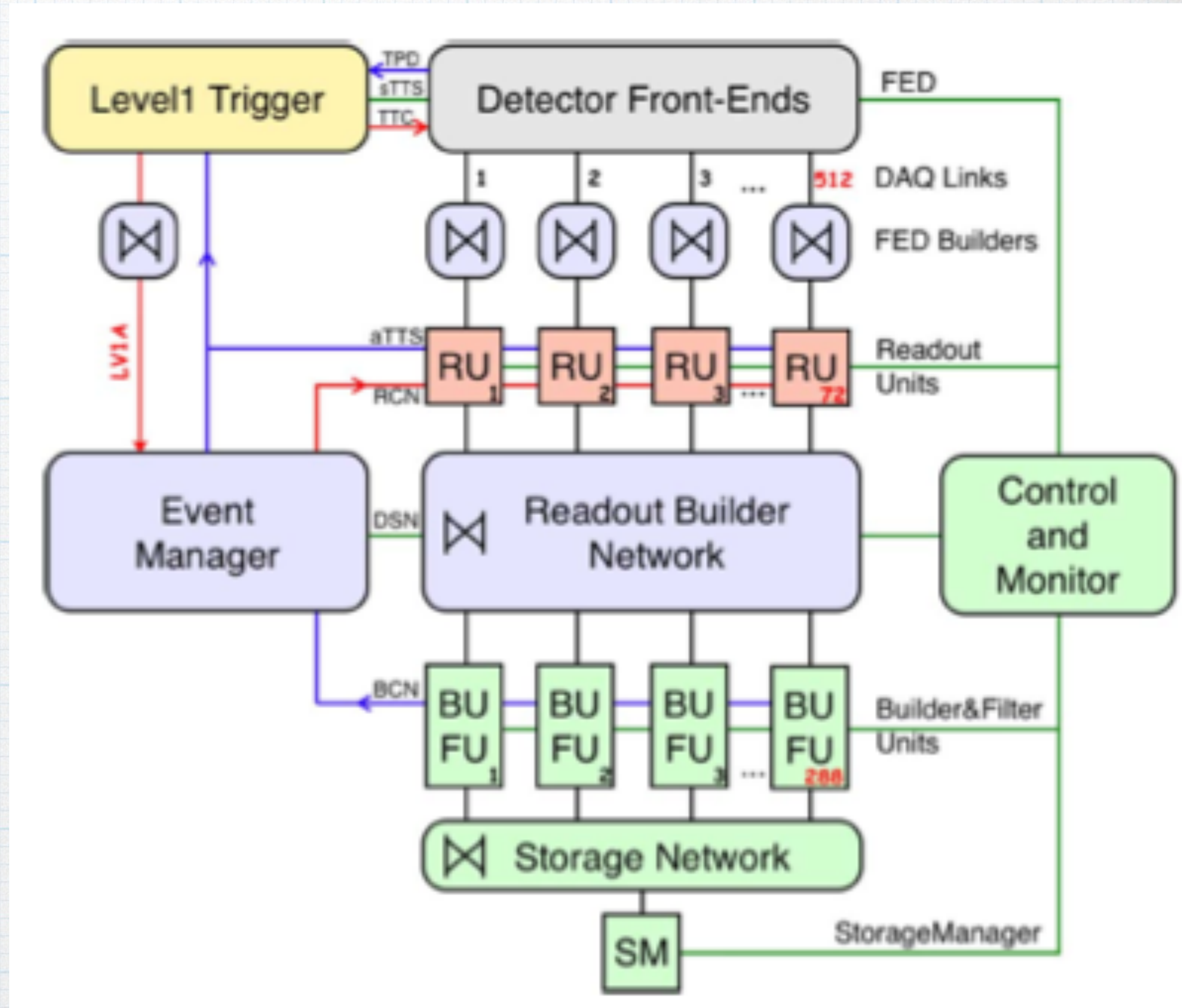
- Level-2 uses the information seeded by level-1 trigger
  - Only the data coming from the region indicated by the level-1 is processed, called **Region-of-Interest (RoI)**
  - The resulting total amount of RoI data is minimal: a few % of the Level-1 throughput
  - Level-2 can use the full granularity information of only a part of the detector
- No need of large bandwidth
- Complicate mechanism to serve the data selectively to the L2 processing



Typically, there are less than 2 RoIs per event accepted by LVL1

networked HLT: CMS

- Data from the readout system (RU) are transferred to the filters (FU) through a builder network
- Each filter unit processes only a fraction of the events
- Event-building is factorized into a number of slices, each one processing only 1/nth of the events
  - Large total bandwidth still required
  - No big central network switch
  - Scalable



FU= Filter Unit, several CPU cores= several filtering processes executed in parallel

a TRG menu

* Our recording power is limited like our stomach. Can't eat all, we must choose.
* Needed especially for collider detectors
  * A trigger menu is the list of our selection criteria
  * Each item on the menu is a trigger path = instructions for each trigger level (L1+L2+L3..)
  * An event is stored if one or more trigger chain criteria are met
* Prescales and menu to be adjusted during run
* Inclusive: Allow good candidates + anything else
  * collect the signal samples (mostly un-prescaled)
* Redundant: Issues in a single one detector or in a trigger input do not affect physics (reduced efficiency but still the measurement is possible)

Object	L1 (Hz)	L2 (Hz)	EF (Hz)
Single-electrons	5580	176	27.3
Multi-electrons	6490	41.1	6.9
Multi-photons	common	2.9	< 0.1
Single-photons	common	33.4	9.1
Multi-Jets	221	7.9	7.9
Single-Jets	24.4	24.4	24.4
Multi-Fjets	2.7	2.7	2.7
Single-Fjets	3.7	3.7	3.7
Multi-bjets	common	12.9	2.6
Single-bjets	common	11.6	11.6
Multi-taus	465	14.5	12.4
Single-taus	148	32.9	22.3
Multi-muons	68.6	5.8	2.3
Single-muons	1730	204	21.8
Missing E_T	37.9	31.	3.8
Total E_T	6.3	6.3	1
Total Jet E_T	1.6	1.6	1.6
BPhysics	common	25	13
Muti-Object	5890	134	48
Minimum Bias	1000	10	10
Total	**12000**	**620**	**197**

In summary: let your TRG be:

- ➡ "low latency"
- ➡ "high efficiency"
- ➡ "low deadtime"
- ➡ "redundant"
- ➡ "flexible"
- ➡ "affordable"

happy TRGing

more in
DAQ hardware
detector readout
lectures