Triggering at the LHC: Past, Present & Future

Andrew Rose

Imperial College London

Acknowledgement

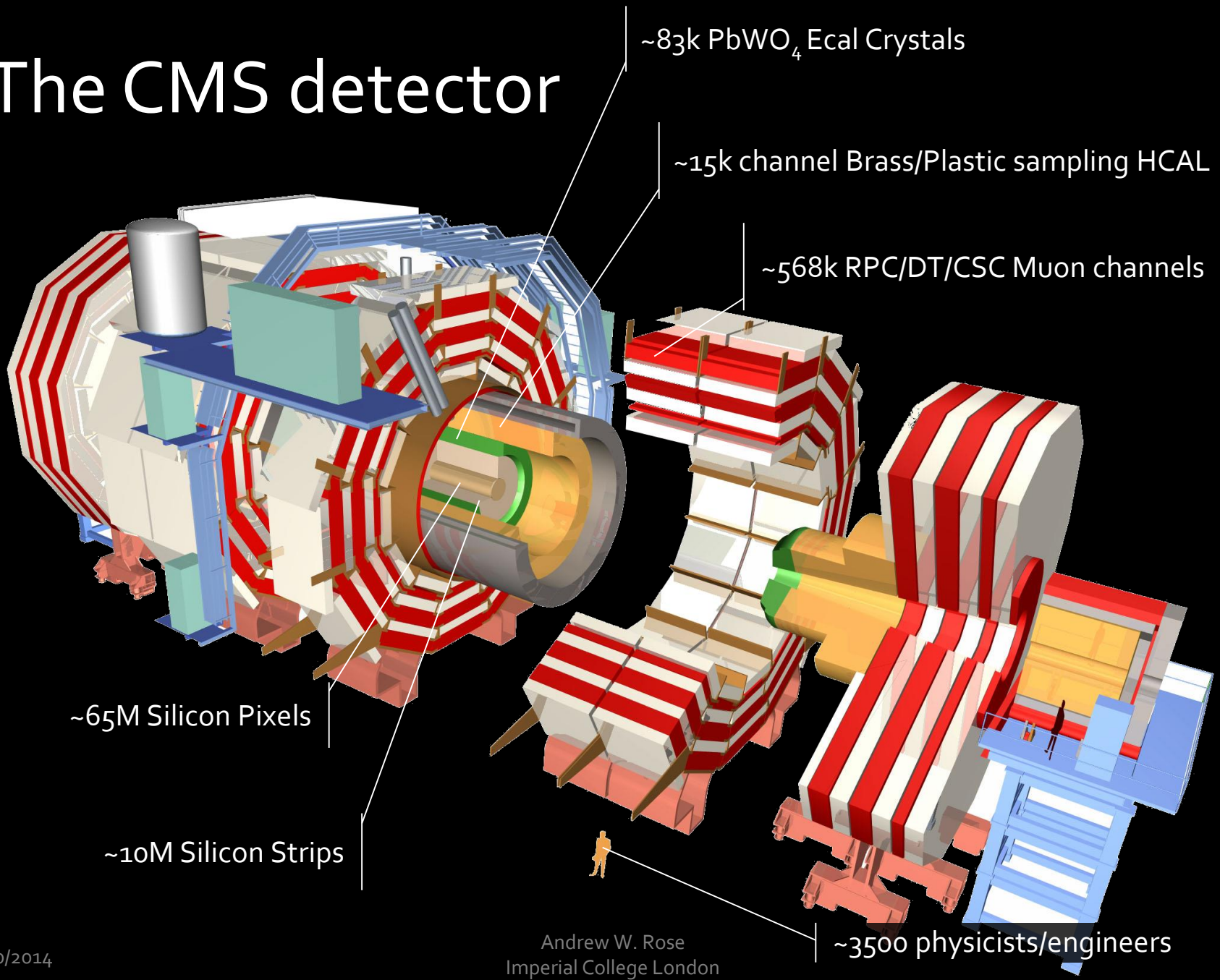
My thanks to Maxeler Technologies who are sponsoring my participation at HPSP14

Overview

- The CMS experiment
  - The challenge
- The CMS trigger – conceptually
- The CMS trigger – Run 1: The Past
  - What is it?
  - Lessons which (should) have been learned
- Time-multiplexing & Spatial Pipelining
- The CMS trigger – Run 2: The Present
  - Current state of the system
- The CMS trigger – Run 3: The Future
  - When "Big Data" really means Big Data

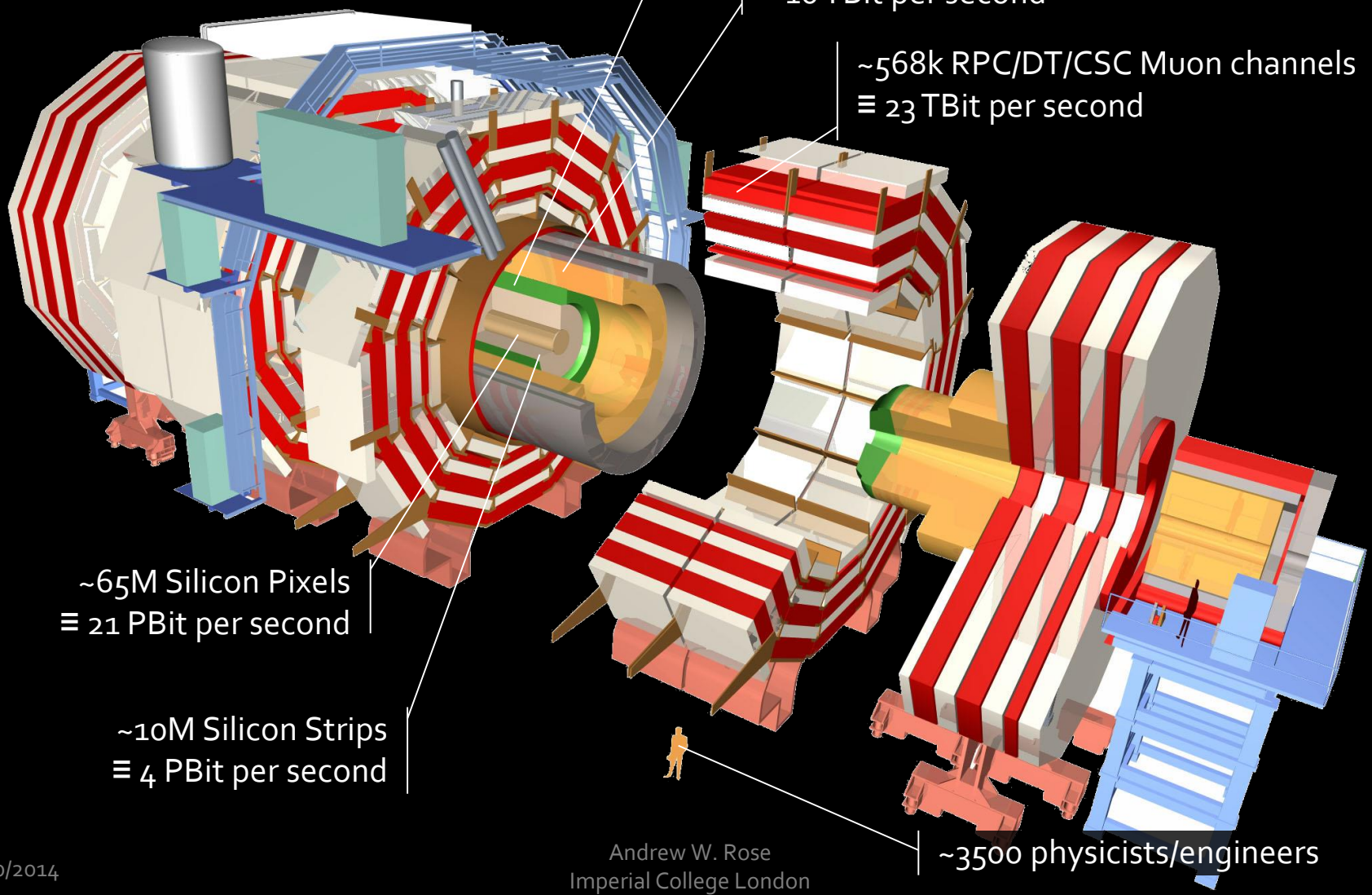
The CMS experiment

The CMS detector

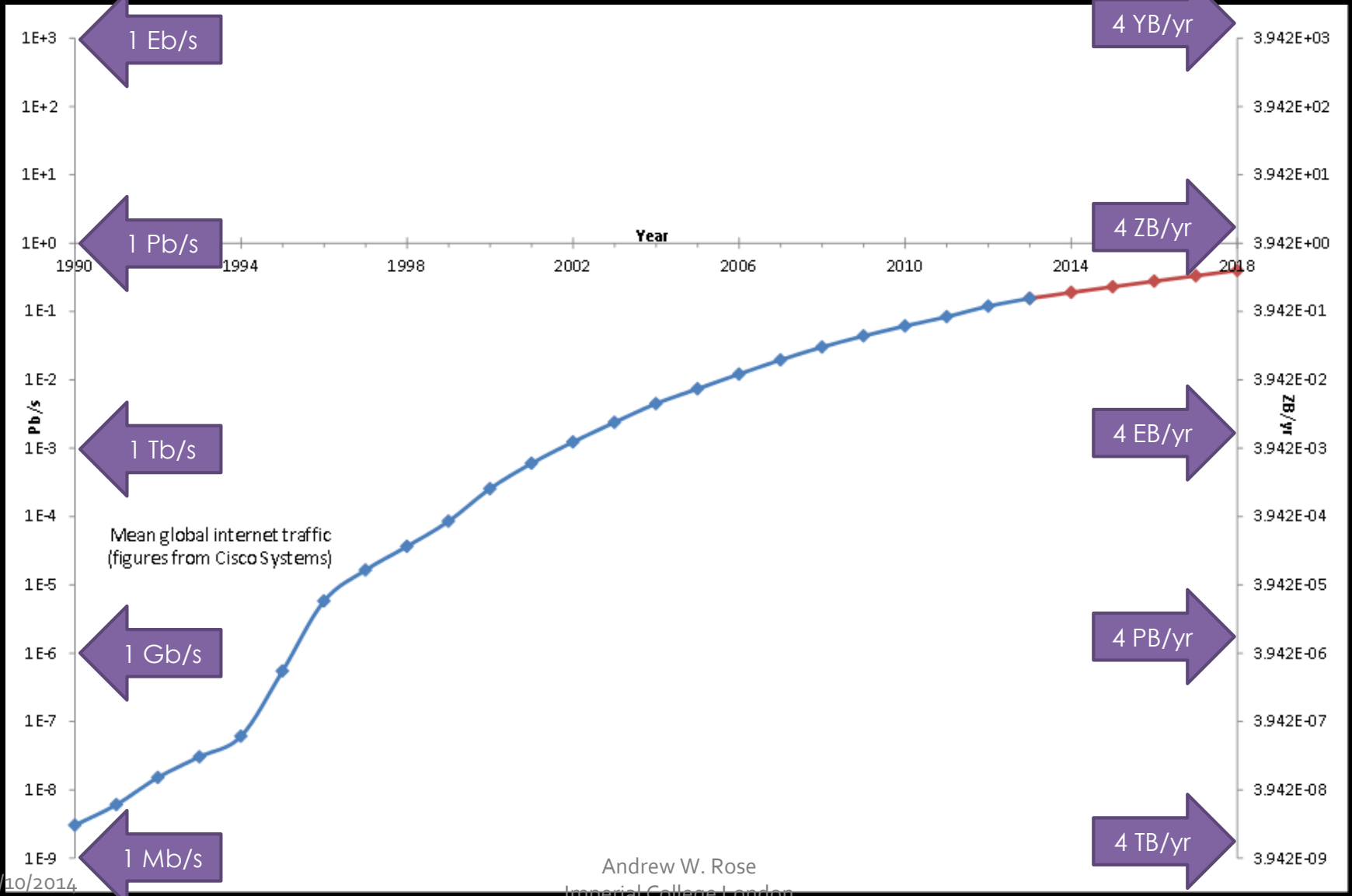


The CMS detector

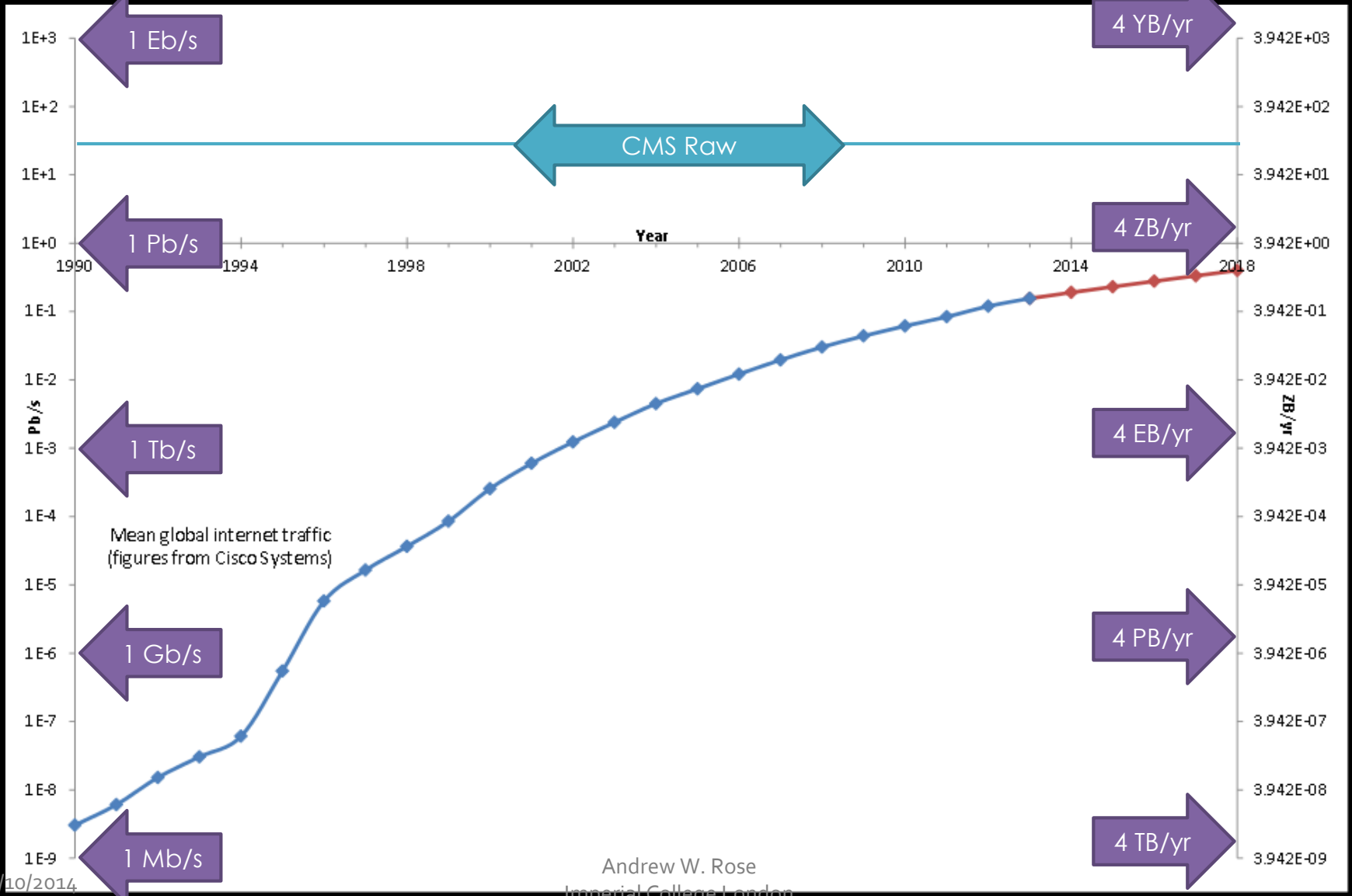
Data rates with no zero-suppression



The Internet: Visualizing big numbers

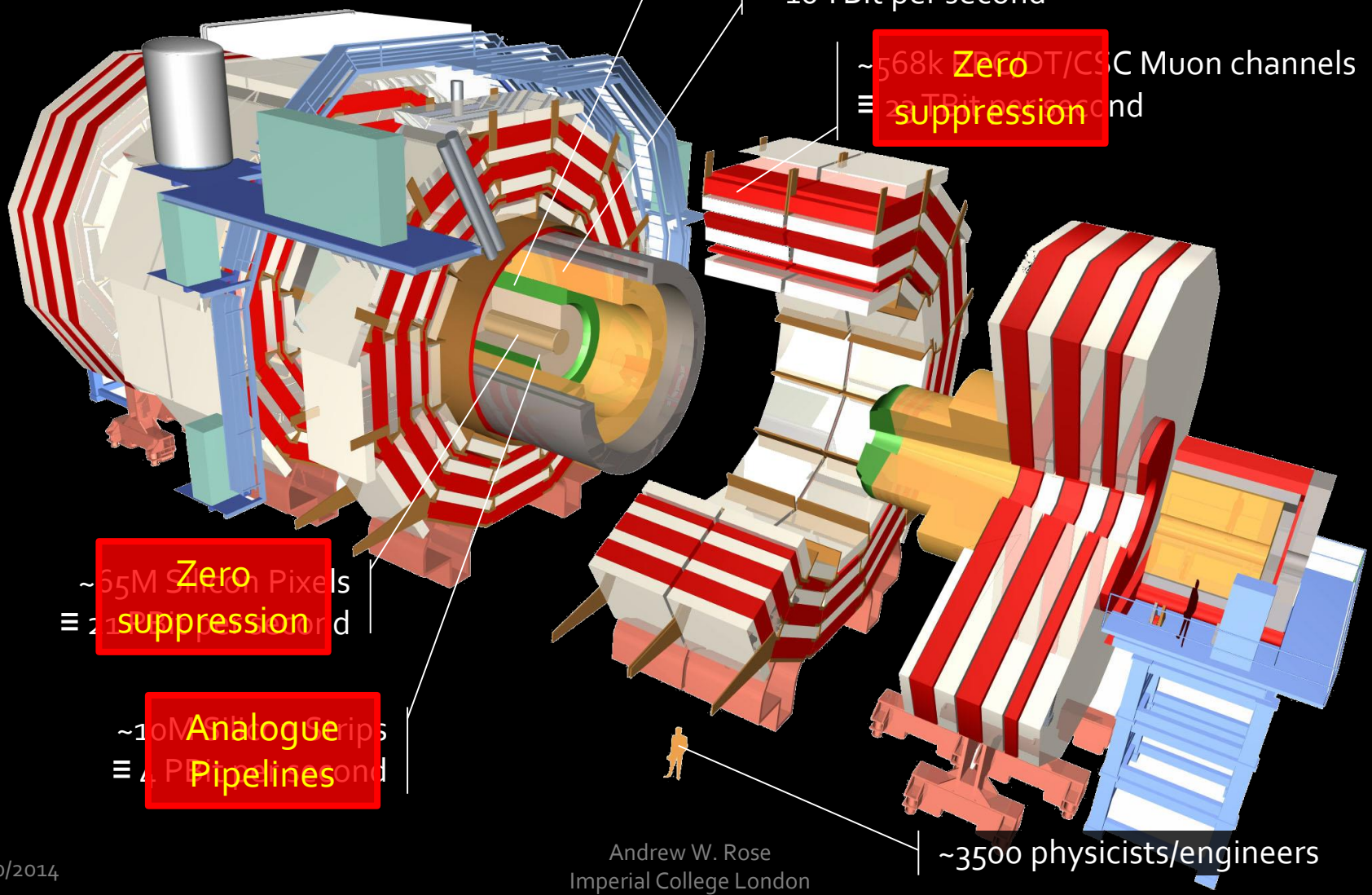


CMS: Visualizing the big numbers

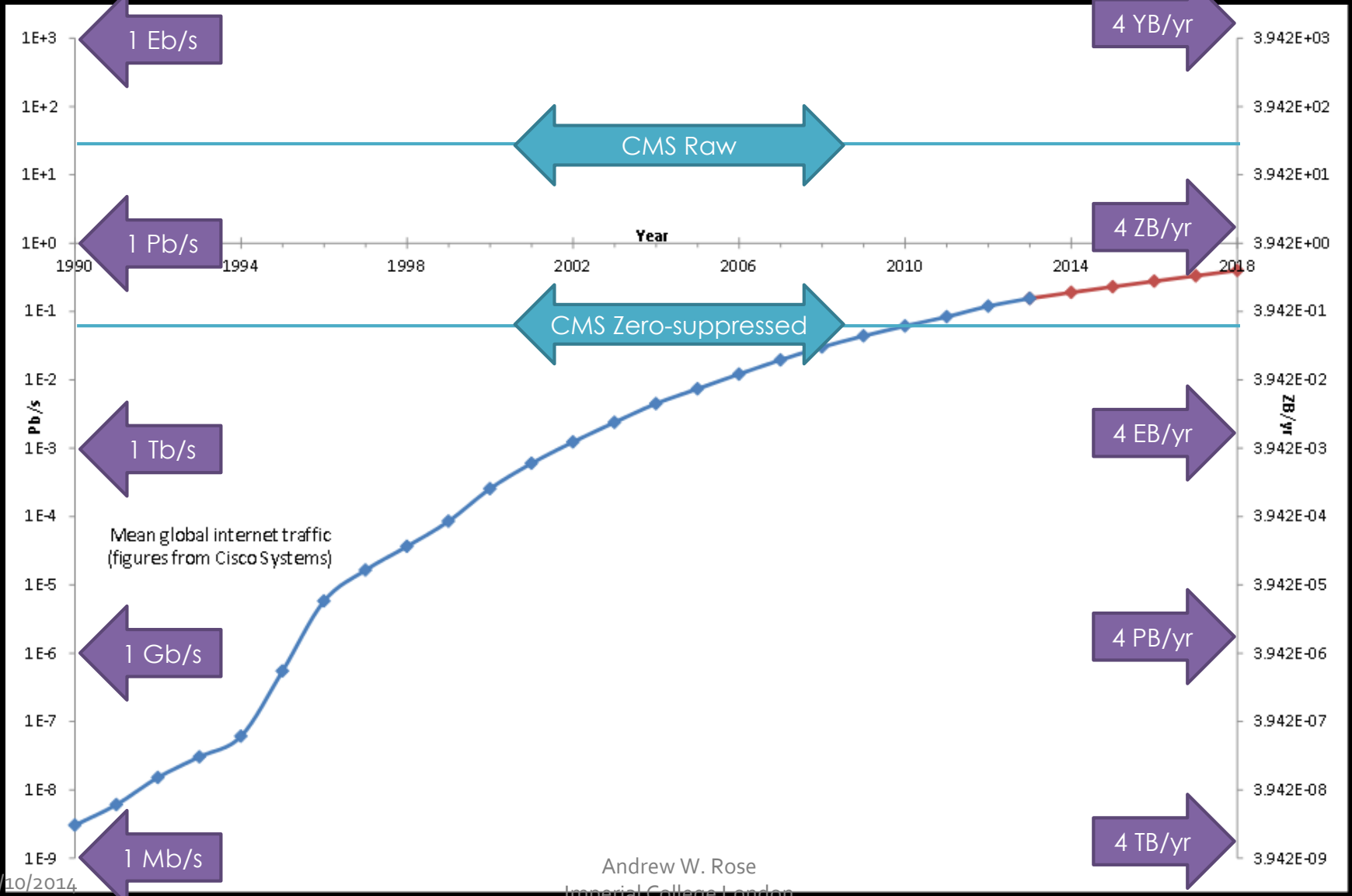


The CMS detector

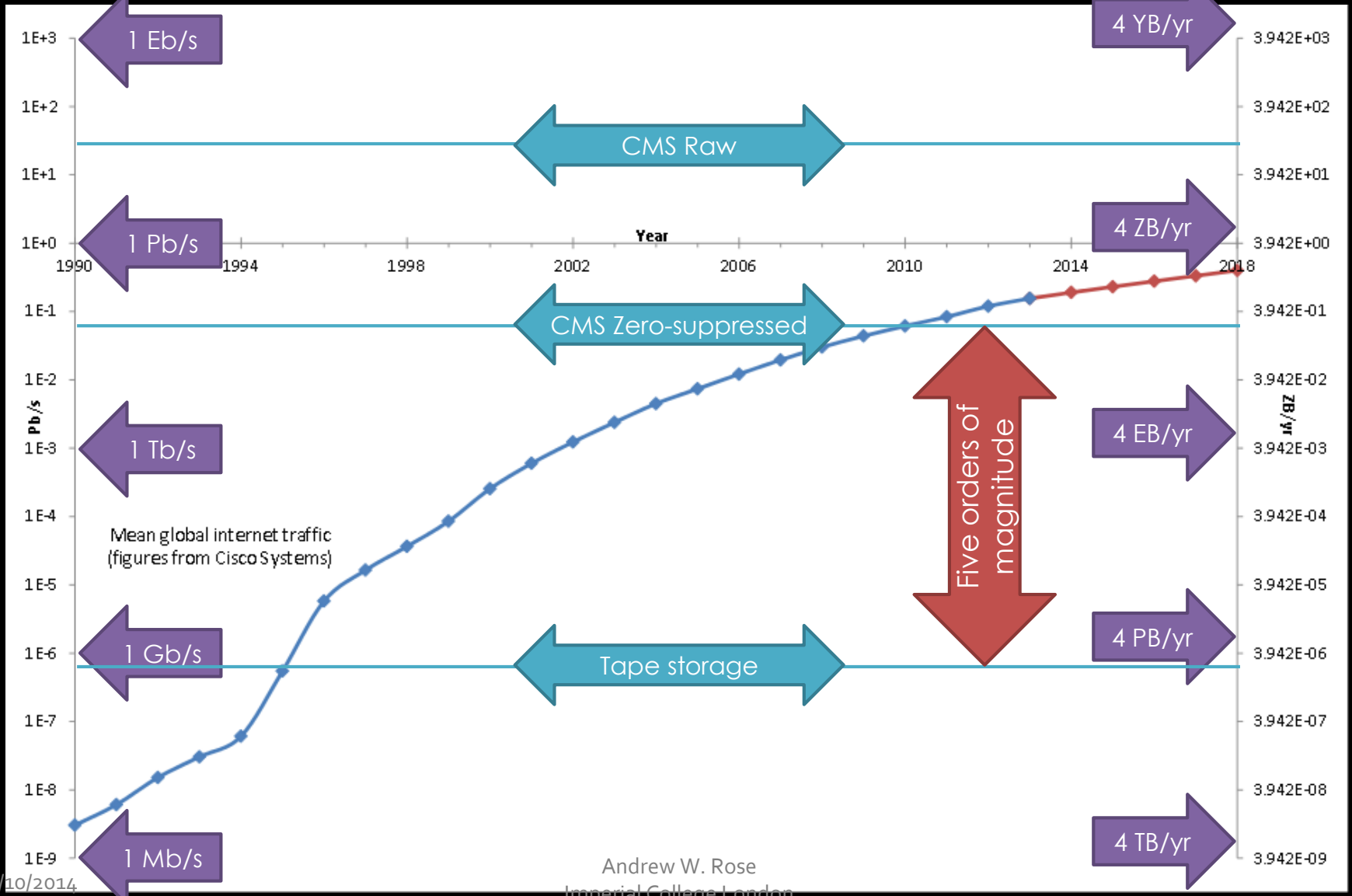
Data reduction



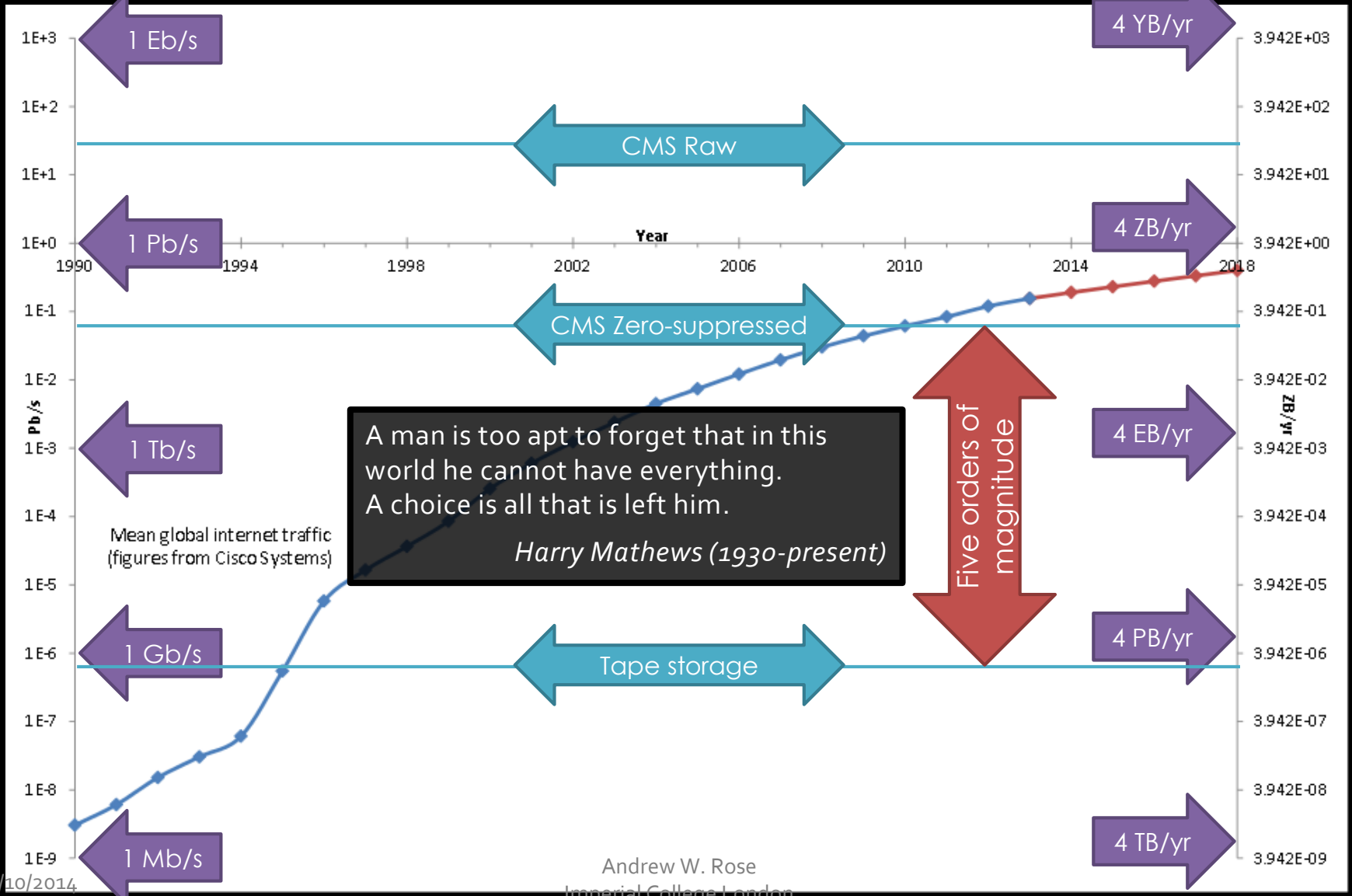
CMS: Visualizing the big numbers



CMS: Visualizing the big numbers

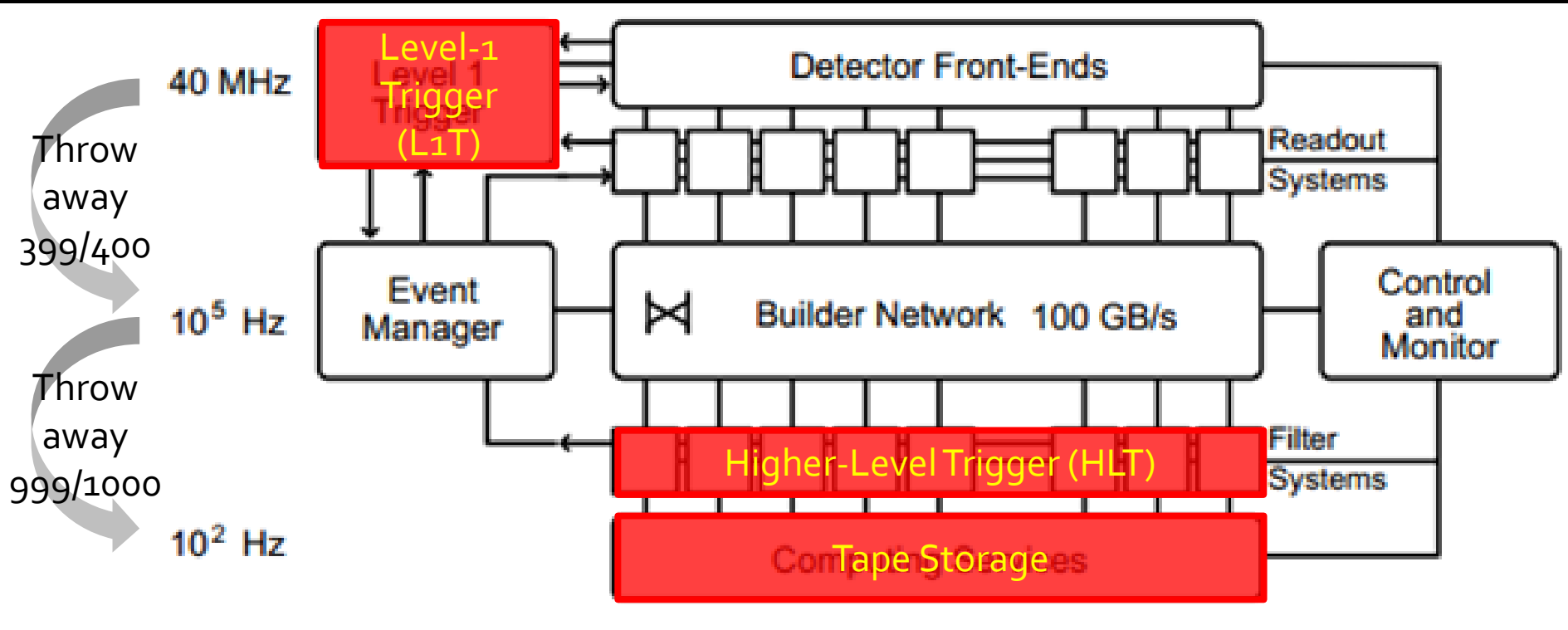


CMS: Visualizing the big numbers

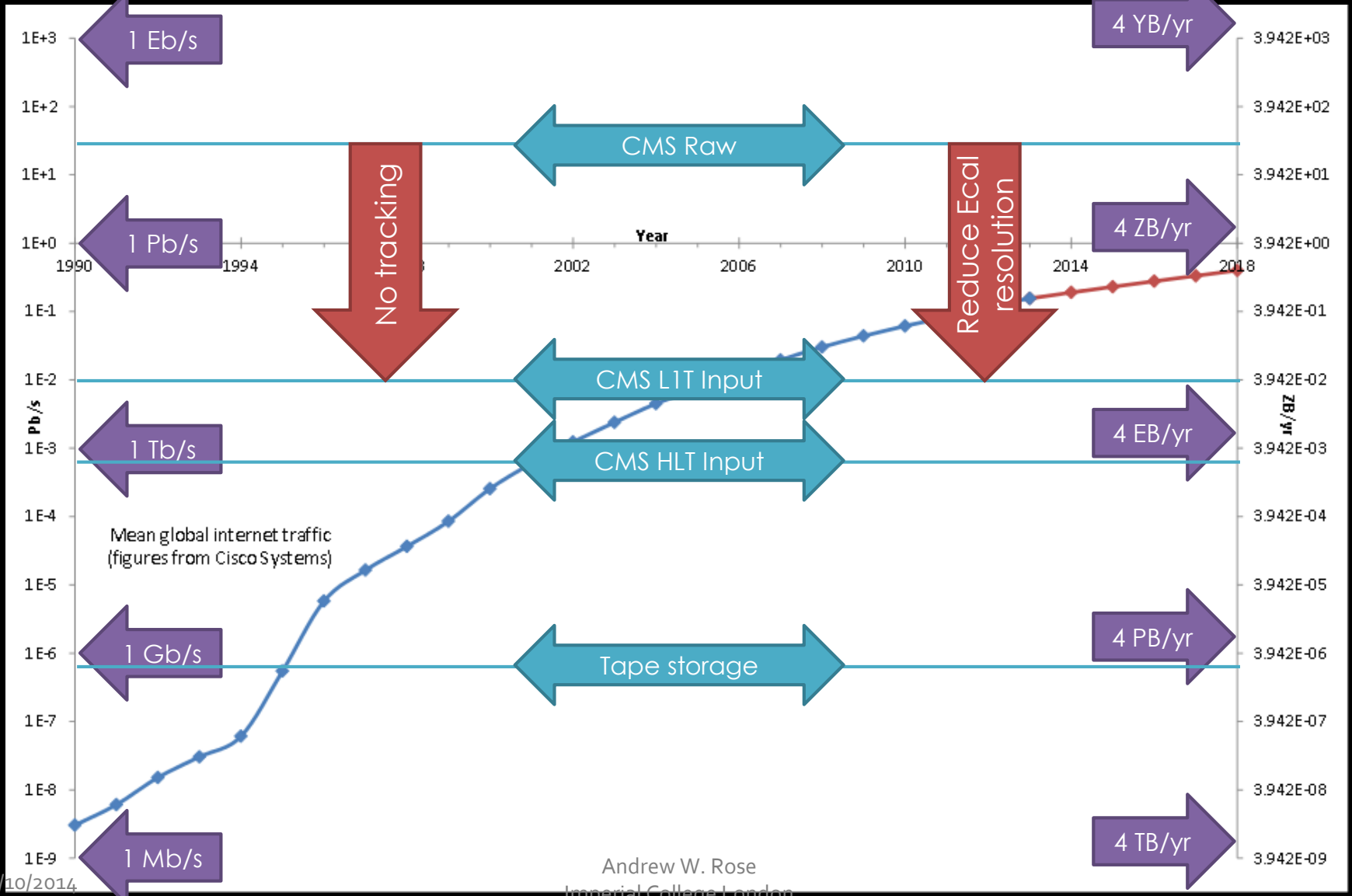


The CMS trigger

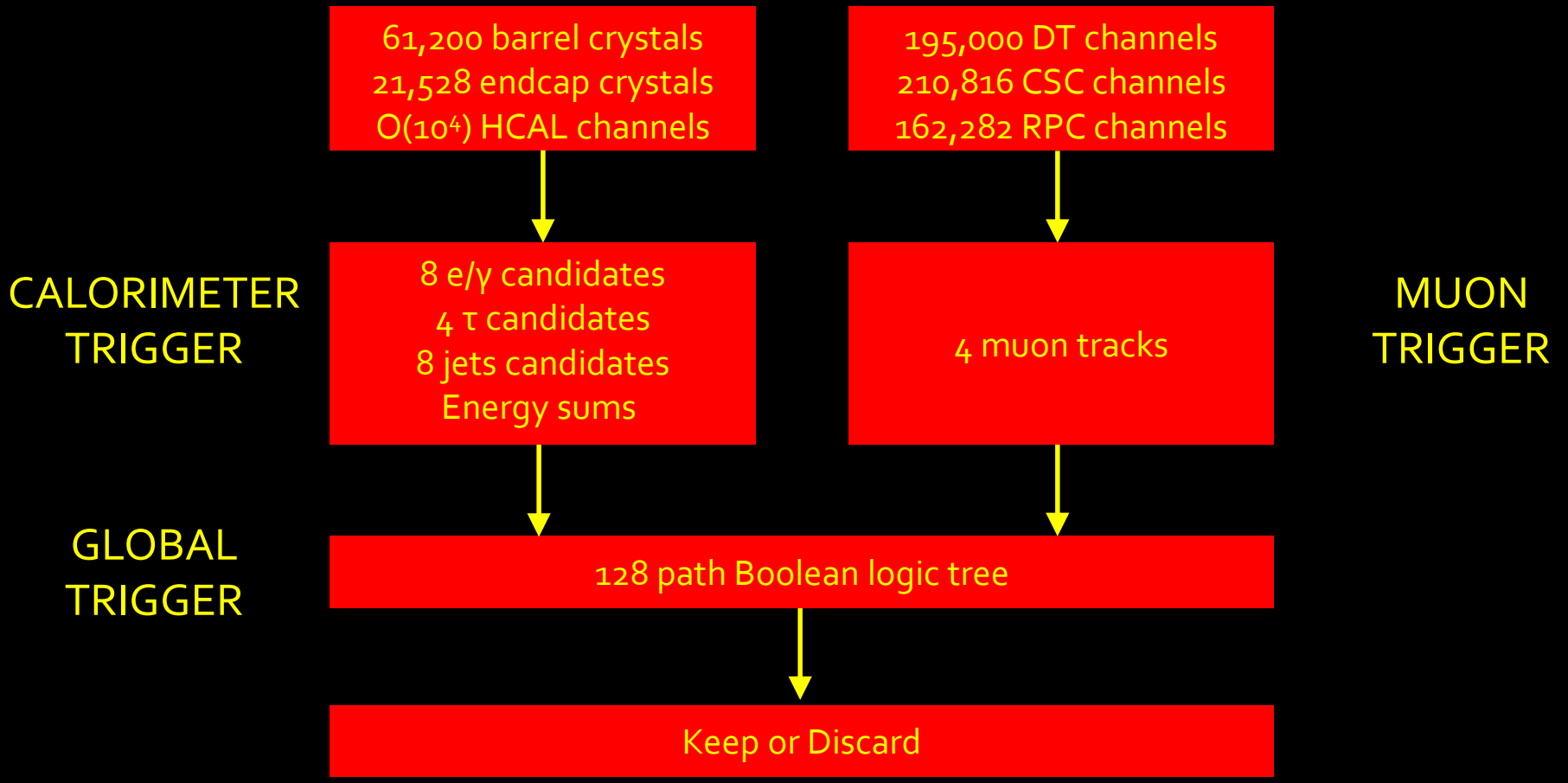
The CMS trigger – Run 1



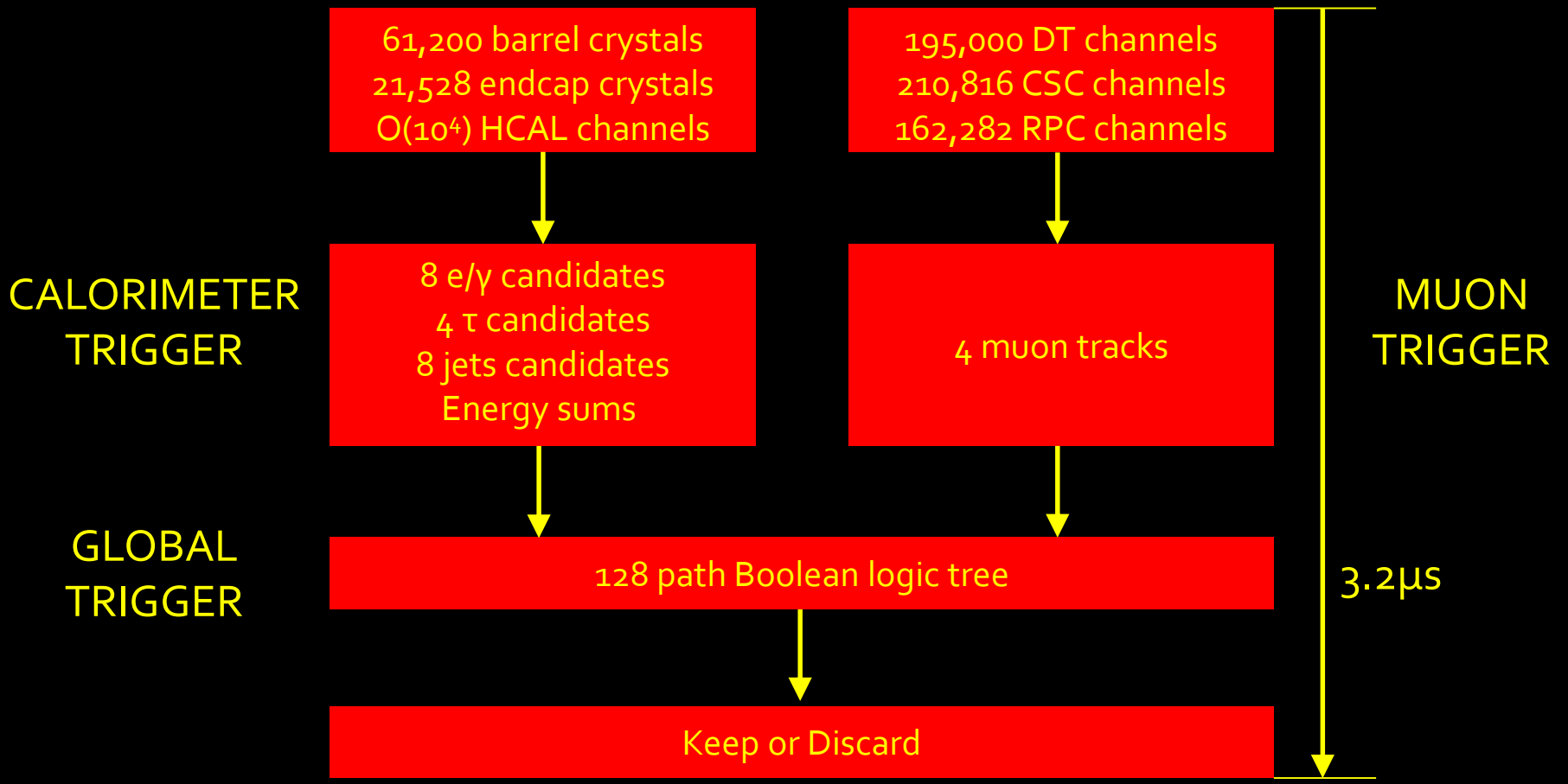
CMS: Visualizing the big numbers



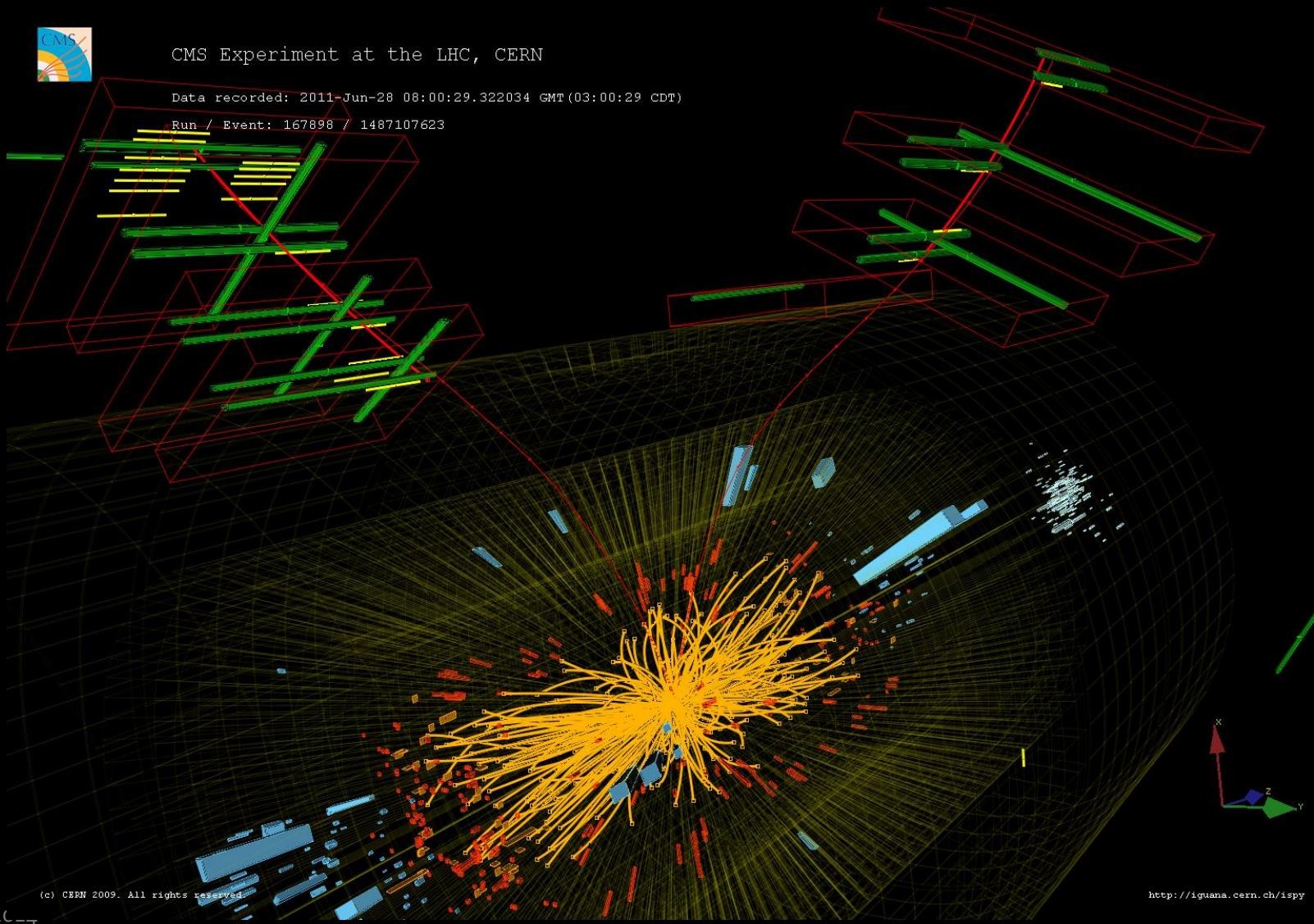
The CMS Level-1 trigger



The CMS Level-1 trigger

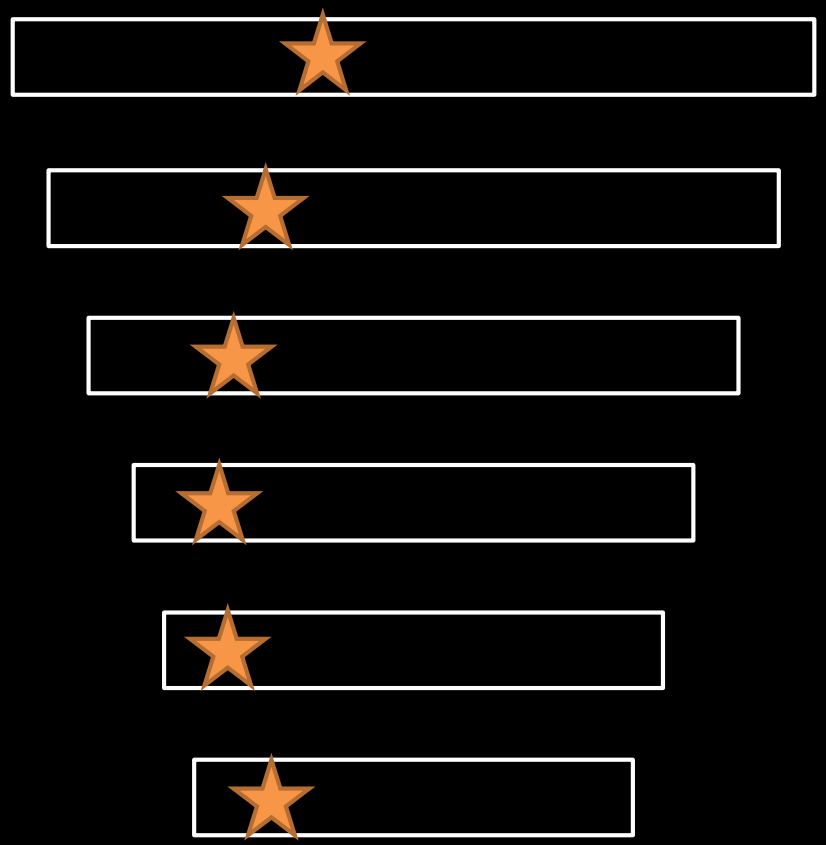


Example: Muon Event

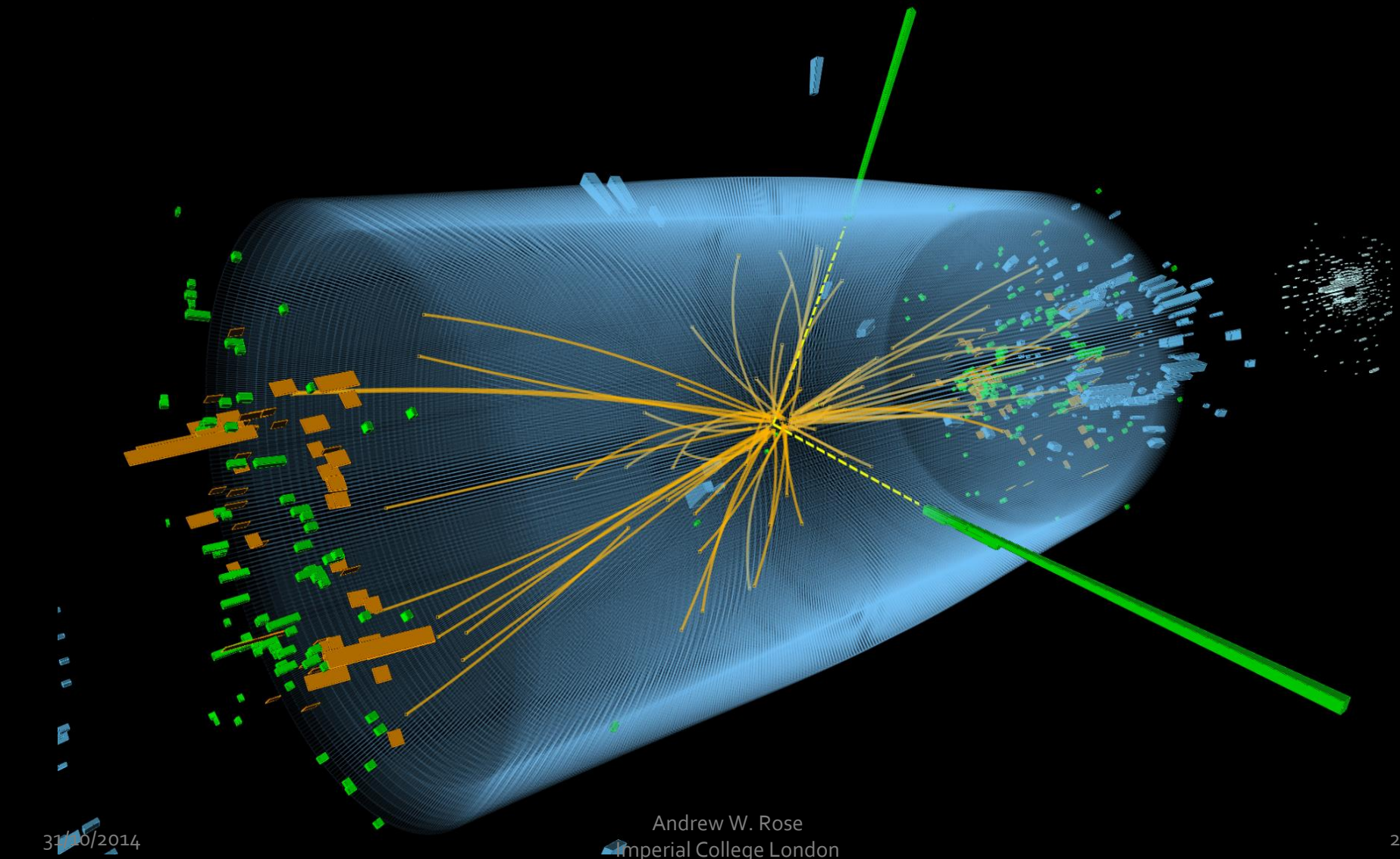


Example: Muon Algorithm

- Join the dots
  - In 3D
  - In 2Tesla magnetic field (outside solenoid) – curved tracks
- Remove duplicate candidates
- Sort candidates

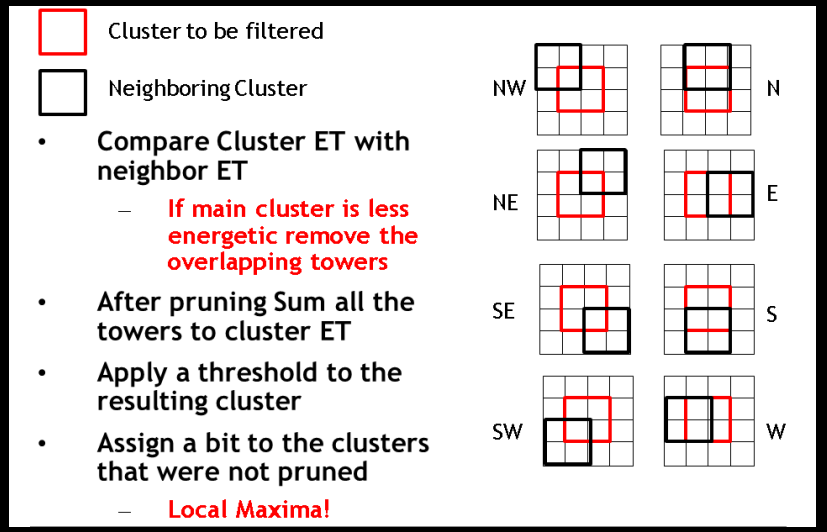


Example: Electron/Photon Event



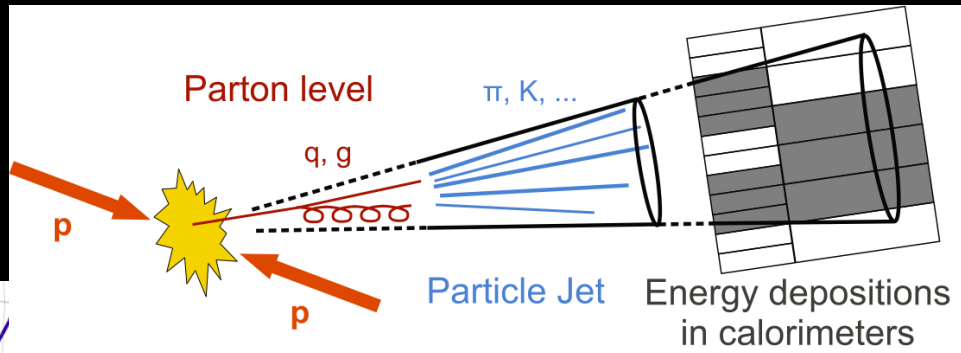
Example: "Simple" e/γ/τ Algorithm

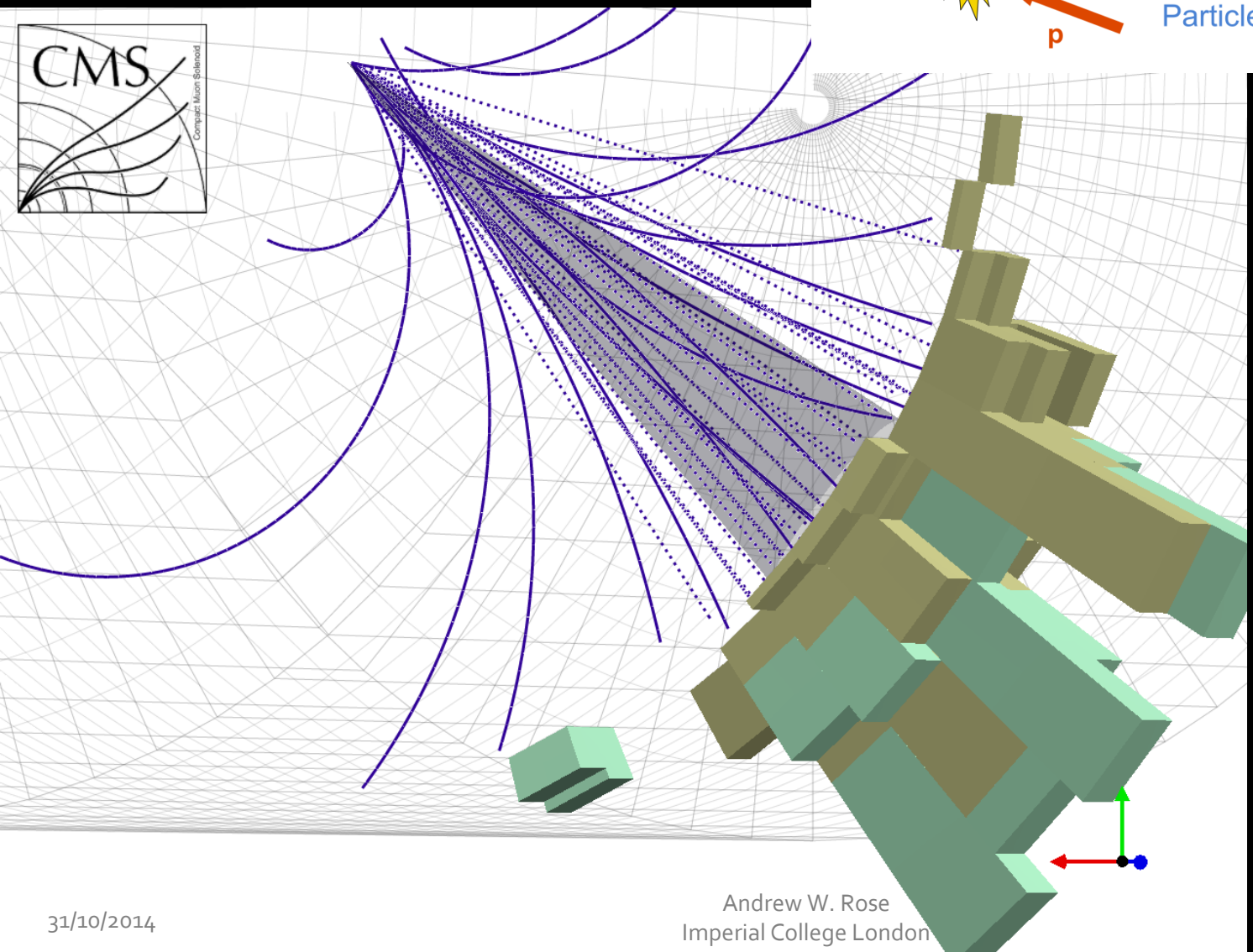
- Form 2x2 sums & filter to prevent multiple counting
- Calculate energy-weighted position within cluster
- Pileup estimation and pileup subtraction
- Classify cluster based on E & H components



- Count clusters over threshold for surrounding 8x8 region (isolation)
- Select e/γ/τ candidates from generic clusters based on the above criteria
- Sort the candidates per ring around the detector
- Sort the candidates along the detector

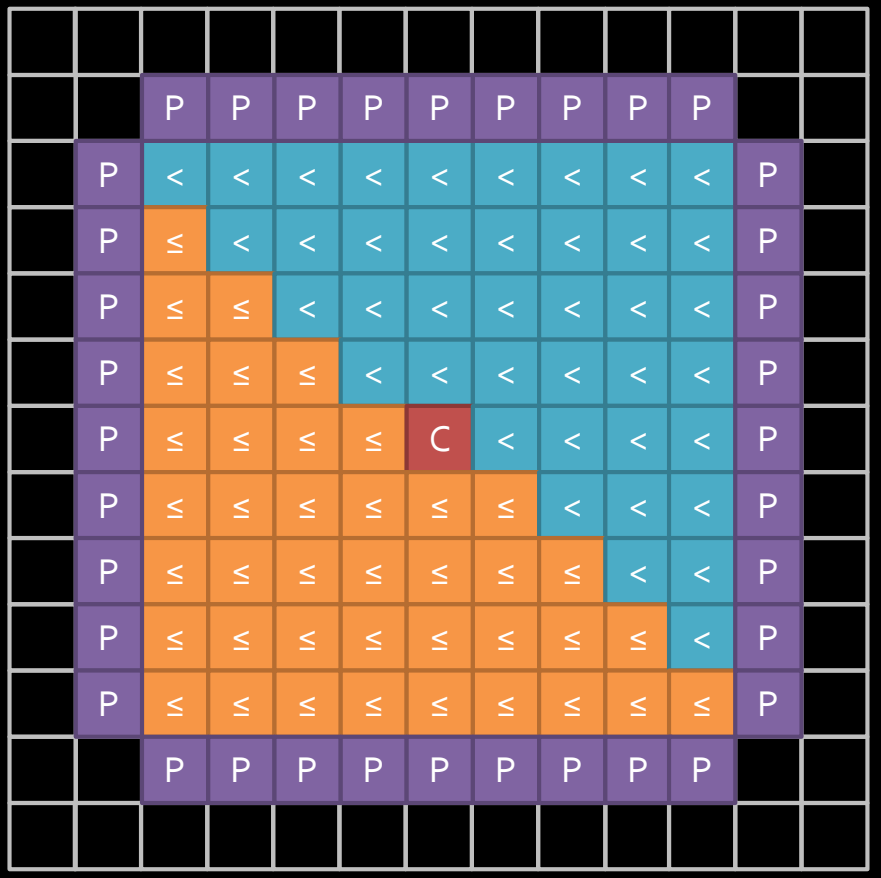
Example: Jet Event





Example: Jet Algorithm

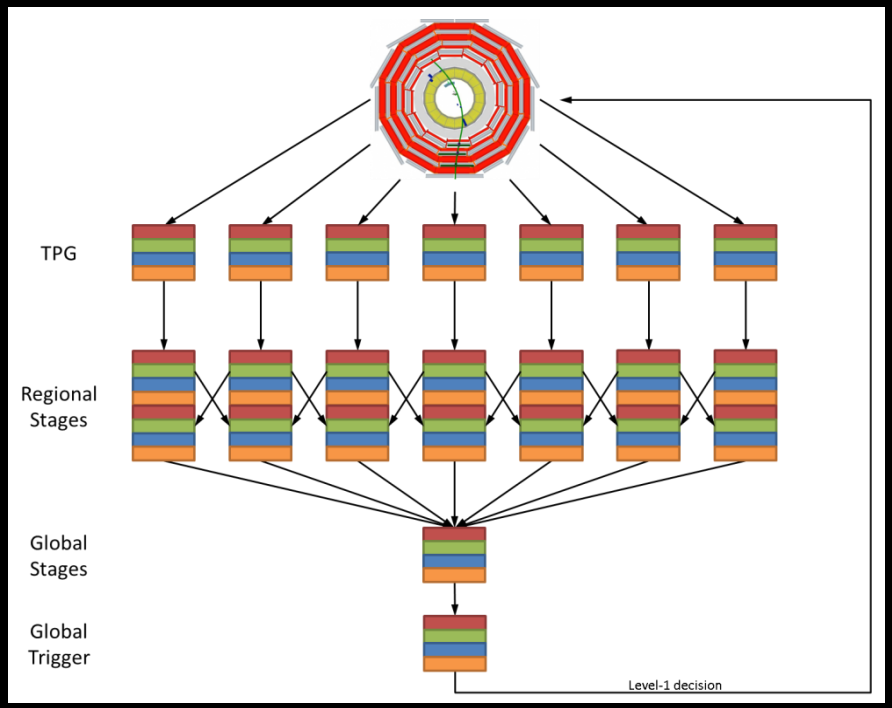
- 9×9 sum of trigger towers at every site
- Fully asymmetric jet veto calculation
- Local Pile-up estimation
- Full overlap filtering
- Pile-up subtraction
- Sort the candidates per ring around the detector
- Sort the candidates along the detector



9×9 jet at tower-level resolution

The CMS trigger – Run 1: The Past

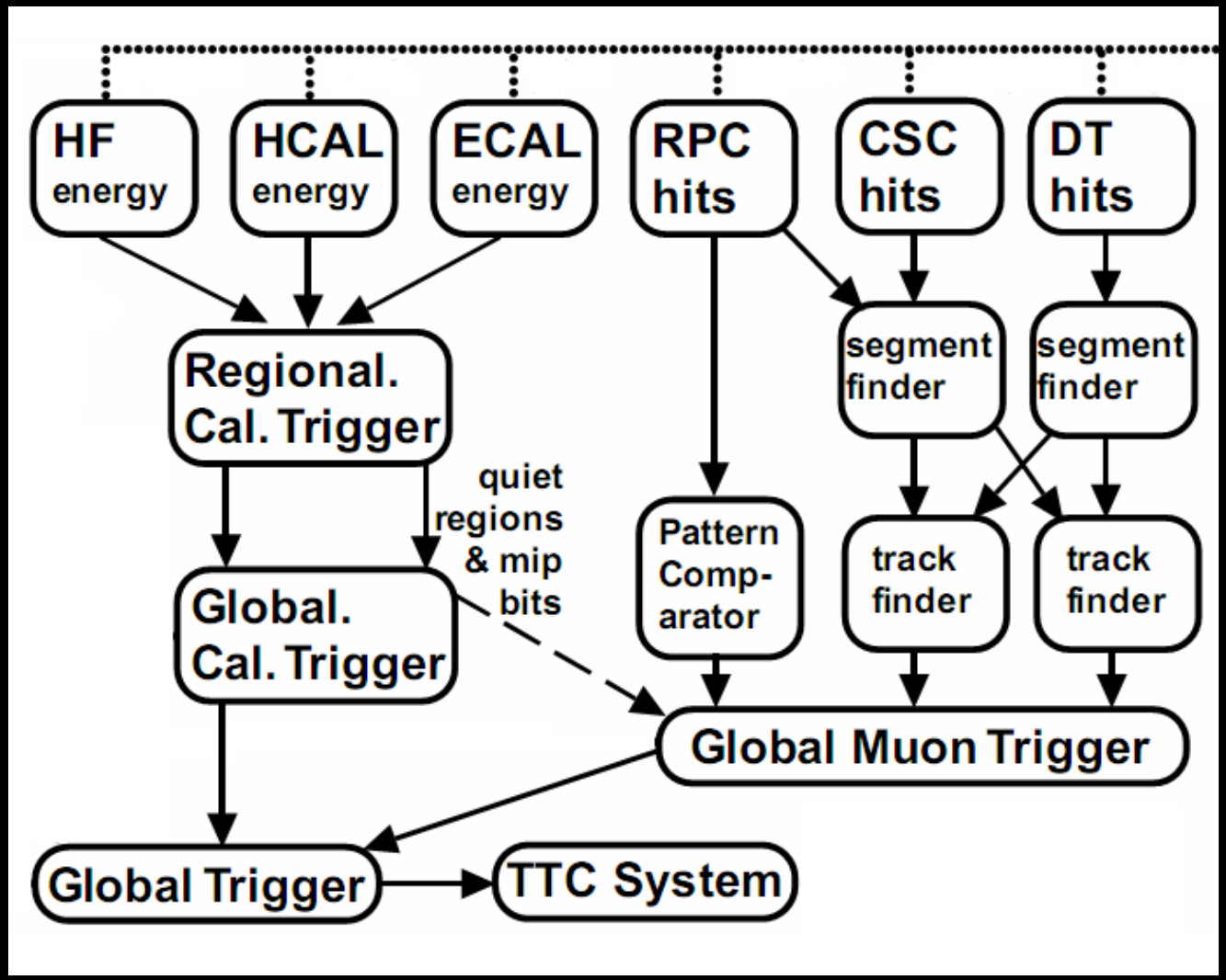
Run-1 Trigger Architecture: Conventional



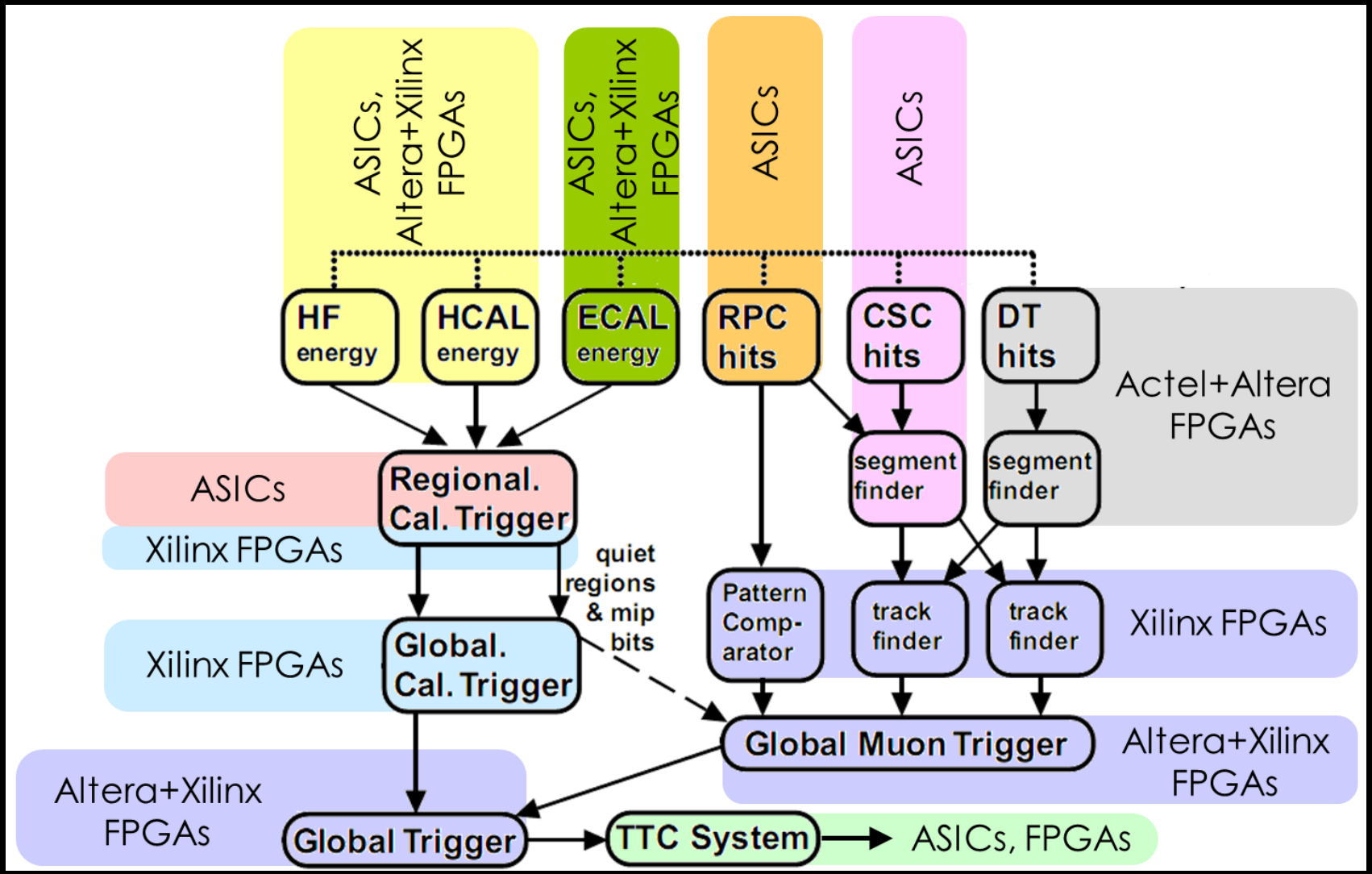
Conventional Trigger Architecture

- Data is processed in regions
- Boundaries between regions must be handled by sharing or duplicating inputs
- Volume of data reduced at each stage by coarsifying data or by selecting and discarding candidates
- When volume of data has been sufficiently reduced it can be passed to the global trigger

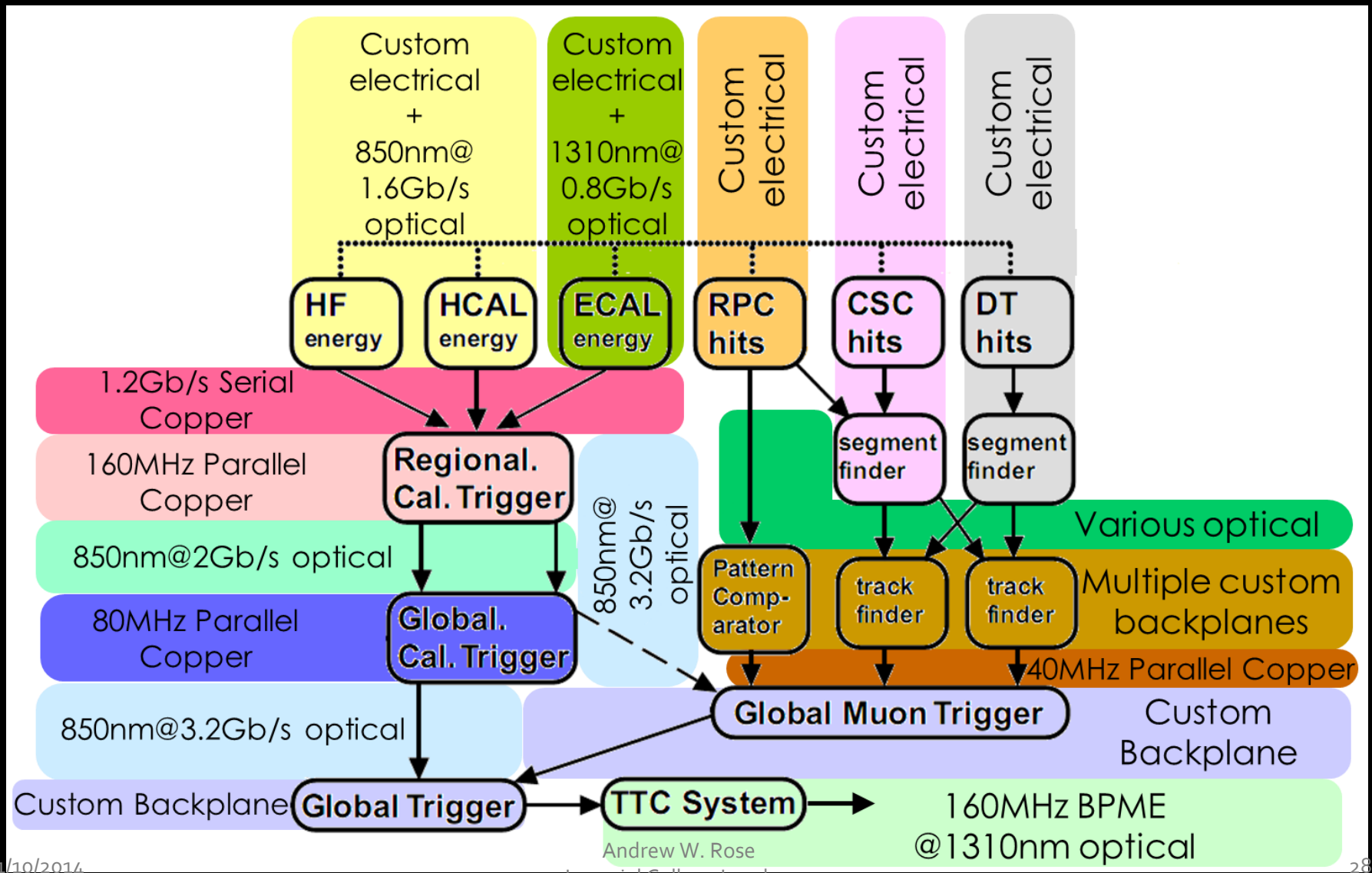
The CMS Level-1 trigger – Run 1



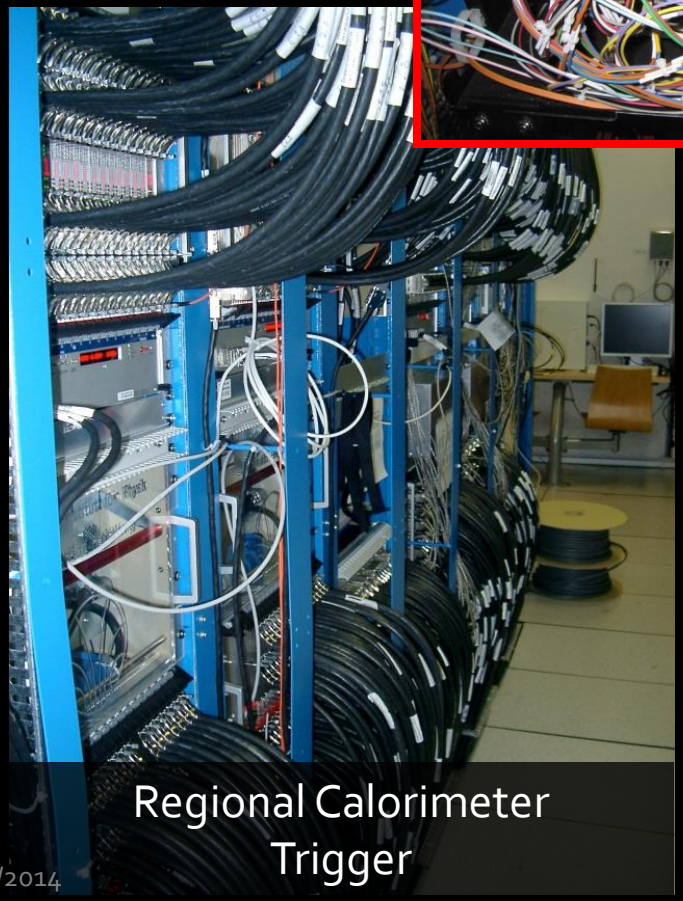
Run 1: Processing Platform



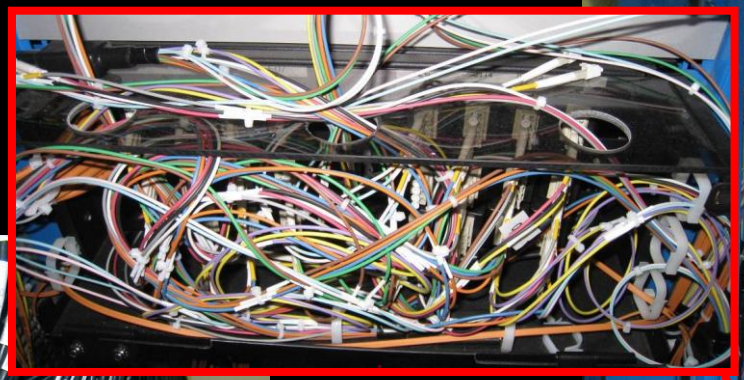
Run 1: Link standards

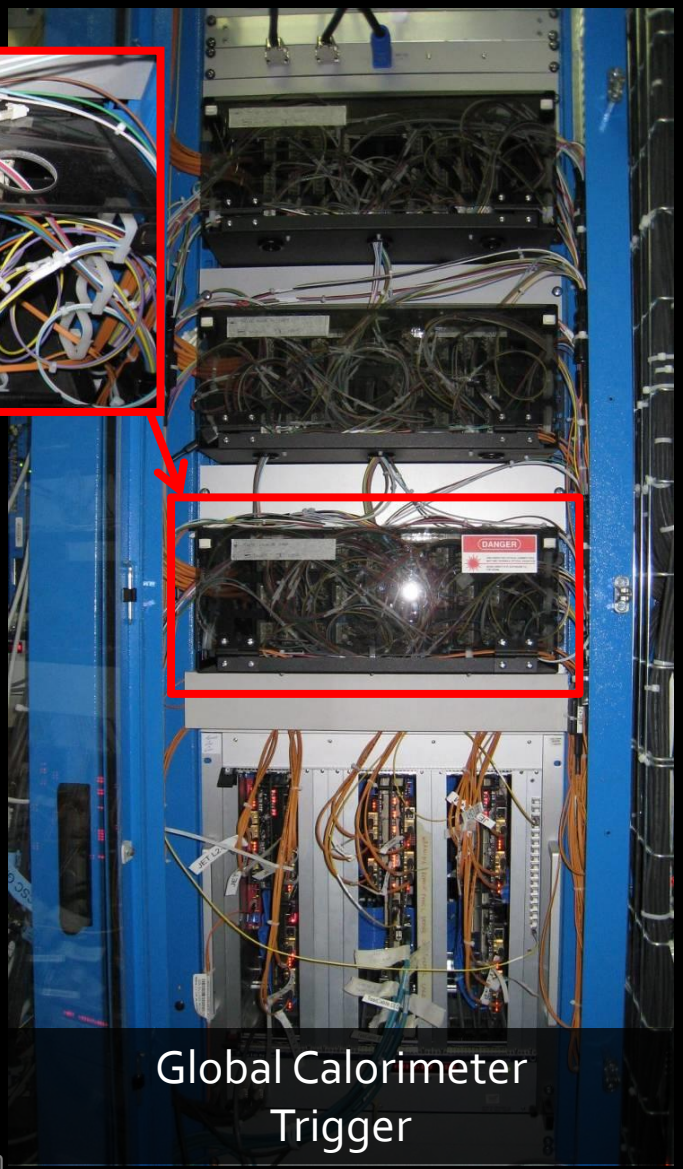


The CMS Level-1 trigger – Run 1



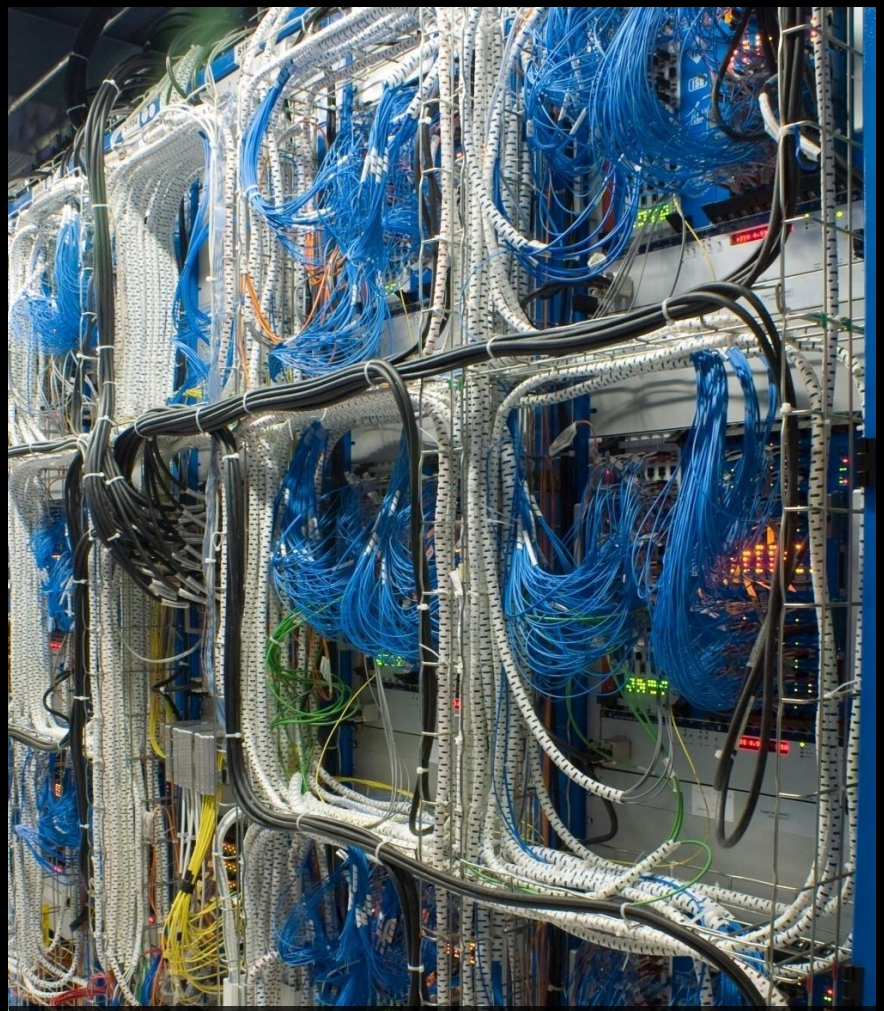
Regional Calorimeter Trigger



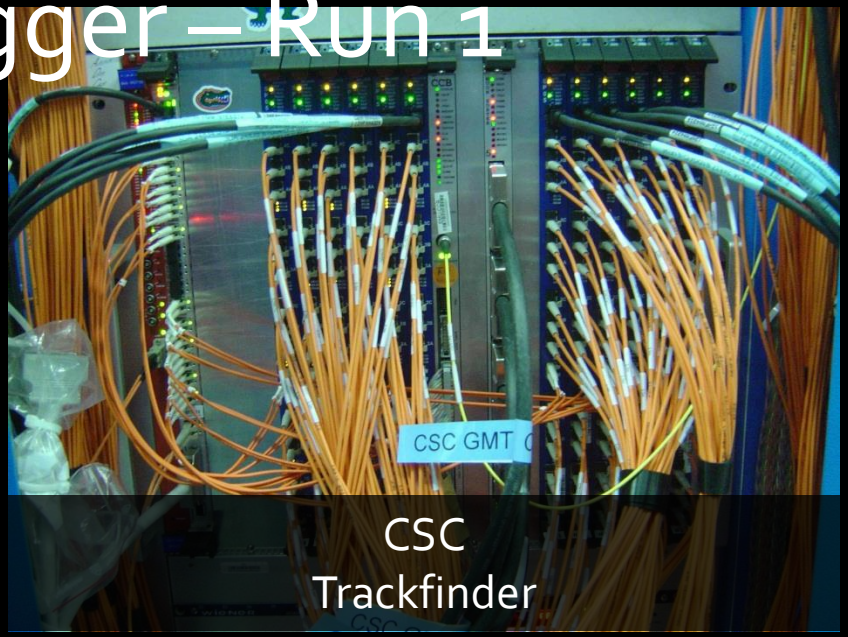


Global Calorimeter Trigger

The CMS Level-1 trigger – Run 1



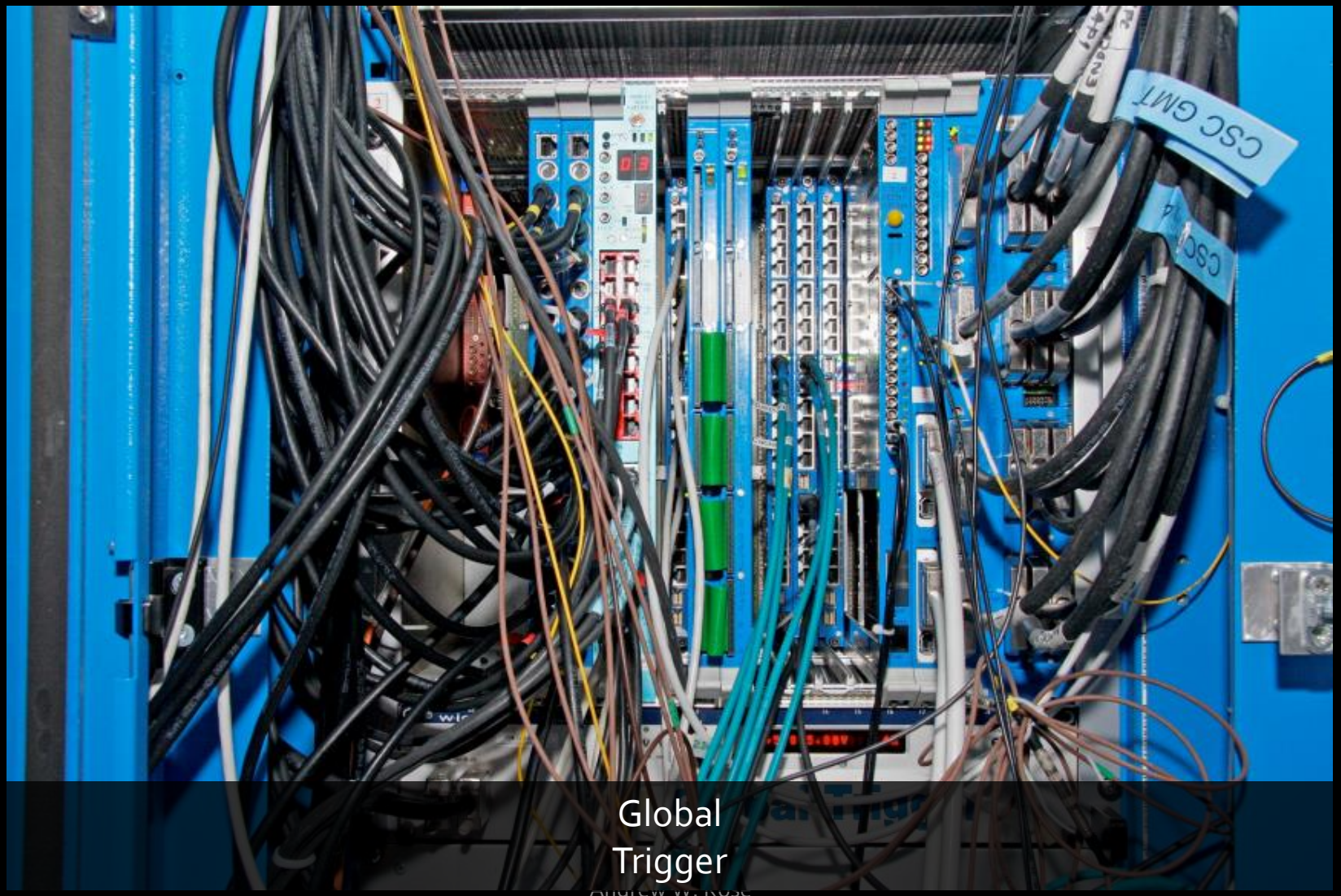
RPC
Pattern Comparator



CSC
Trackfinder

DT Trackfinder
(aka the green salad)

The CMS Level-1 trigger – Run 1



The CMS Level-1 trigger – Run 1

Pros

- It works! It contributed to the finding of the Higgs Boson
- It has been very reliable

The CMS Level-1 trigger – Run 1

Pros

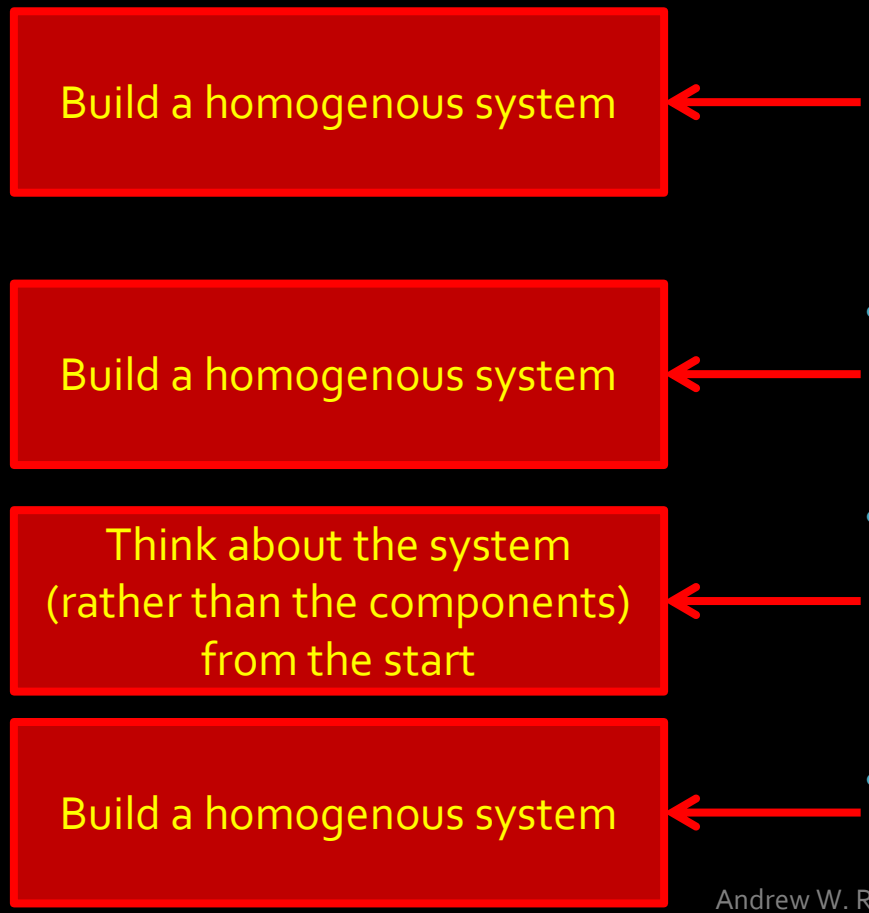
- It works! It contributed to the finding of the Higgs Boson
- It has been very reliable

Cons

- The inhomogeneity of the links and hardware meant that whenever a change was required during construction, new hardware was required
- The number of different board-types means that there is really only one expert per board
- Each piece of hardware requires different software. How these operate together was added as an after thought
- No software or firmware reuse: Maintainability nightmare

The CMS Level-1 trigger – Run 1

Lessons which (should) have been learned



Cons

- The inhomogeneity of the links and hardware meant that whenever a change was required during construction, new hardware was required
- The number of different board-types means that there is really only one expert per board
- Each piece of hardware requires different software. How these operate together was added as an after thought
- No software or firmware reuse: Maintainability nightmare

So why did CMS end up with such a diverse system?

- Boundary-handling
  - In a conventional trigger architecture, data is processed in regions and the boundaries between regions require very different handling for a calorimeter subsystem than, say, a muon subsystem

So why did CMS end up with such a diverse system?

- Boundary-handling
  - In a conventional trigger architecture, data is processed in regions and the boundaries between regions require very different handling for a calorimeter subsystem than, say, a muon subsystem
- Mindset
  - There is a mindset that the trigger is a "hardware problem"
  - This was true in the past but not really true now
  - There is a mindset that software and firmware "come for free"

So why did CMS end up with such a diverse system?

- Boundary-handling
  - In a conventional trigger architecture, data is processed in regions and the boundaries between regions require very different handling for a calorimeter subsystem than, say, a muon subsystem
- Mindset
  - There is a mindset that the trigger is a "hardware problem"
  - This was true in the past but not really true now
  - There is a mindset that software and firmware "come for free"
- Organizational
  - Building the trigger was approached like building the detector
  - The system was divided into chunks first and the interfaces between chunks decided internally by the groups on each side of the boundary.
  - Within each chunk, the group was free to come up with any solution that worked (see above)

So why did CMS end up with such a diverse system?

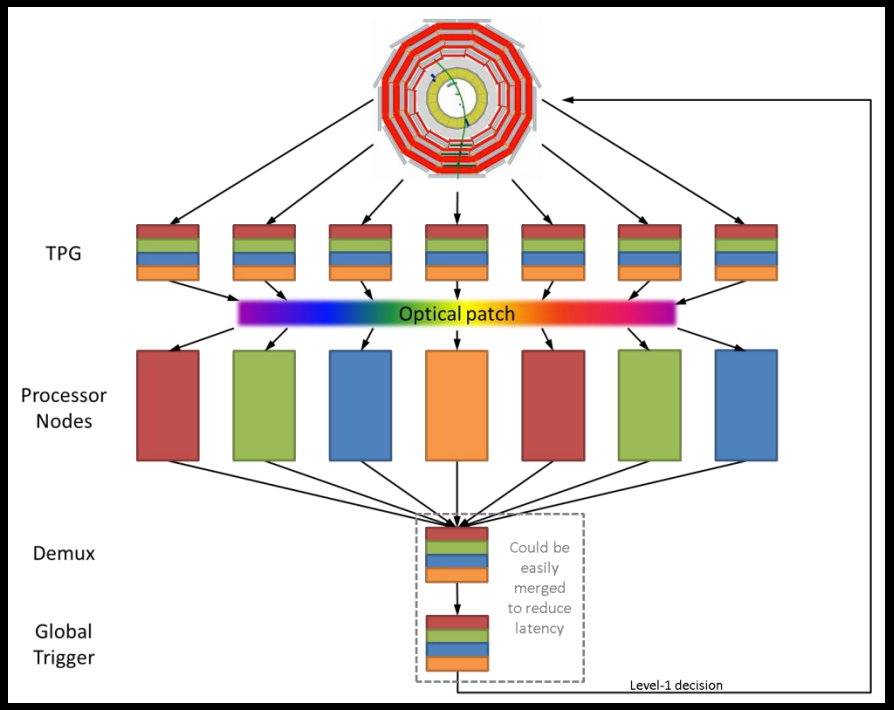
- Boundary-handling
  - In a conventional trigger architecture, data is processed in regions and the boundaries between regions require very different handling for a calorimeter subsystem than, say, a muon subsystem
- Mindset
  - There is a mindset that the trigger is a "hardware problem"
  - This was true in the past but not really the case now
  - There is a mindset that software and firmware "come for free"
- Organizational
  - Building the trigger was approached like building the detector
  - The system was divided into chunks first and the interfaces between chunks decided internally by the groups on each side of the boundary.
  - Within each chunk, the group was free to come up with any solution that worked (see above)

> Progress is impossible without change, and those who cannot change their minds cannot change anything
> *George Bernard Shaw (1856-1950)*

> That is simple, my friend. It is because politics is more difficult than physics
> *Albert Einstein, 1946*

Time-Multiplexing & Spatial Pipelining

Time-multiplexed trigger



Time-Multiplexed Trigger Architecture

- Data from an event is buffered and retransmitted to the first processing node over *N* bunch crossings
- Data from the next event is buffered and retransmitted to the second processing node, again, over *N* bunch crossings
- Process is repeated in round-robin fashion across ≥*N* processing nodes
- Because both algorithm latency and data volume are constant, dataflow is fully deterministic and no complex scheduling mechanism is required

Time-multiplexed trigger

Layer 1
"Preprocessors"

Layer 2
"Processing nodes"

The parallel approach to computing does require that some original thinking be done about numerical analysis and data management in order to secure efficient use.
In an environment which has represented the absence of the need to think as the highest virtue this is a decided disadvantage.

Daniel Slotnick, 1967

The advantages of Time-Multiplexing

Take the principle to its limits: Eliminate the boundaries completely

No specialization of hardware for a particular boundary handling situation

- "One-board-fits-all" design

No duplication of data needed

- Less hardware and fewer links for a given resolution

No data needs to be thrown away

- Should allow better physics

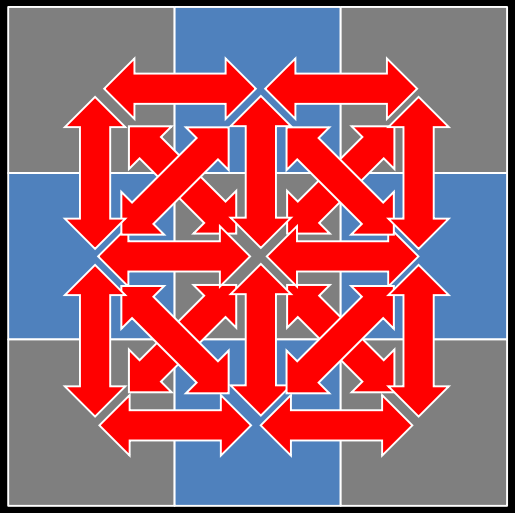
Equality of data

- Without duplication, all data is equal
- Can easily add new input sources
- Scalability

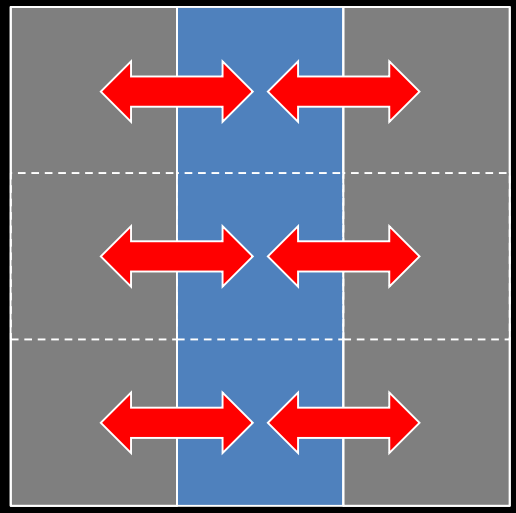
The advantages of Time-Multiplexing

But there is nothing to say that we have to go to that extreme…

We could use Time-Multiplexing to reduce a two-dimensional boundary handling problem to a one-dimensional boundary handling problem



Conventional Boundary Handling

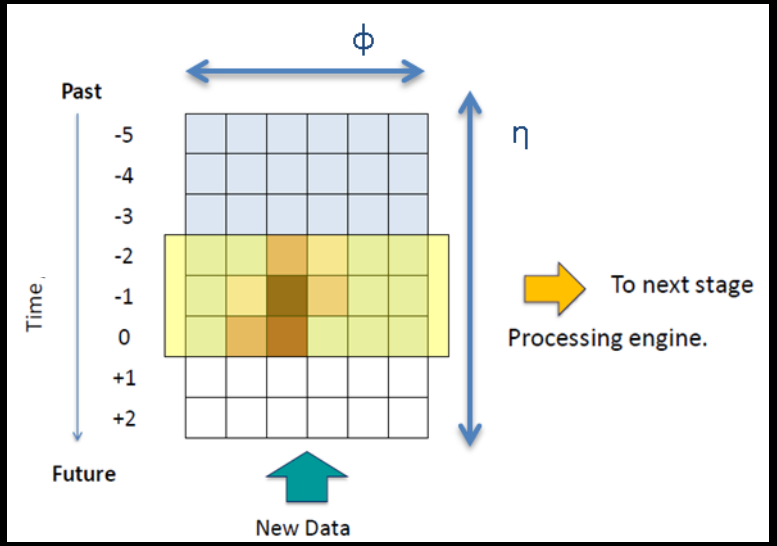


Partial TMT Boundary Handling

Further advantages of a TM trigger

Architecture is naturally matched to processing of data in FPGAs

- Parallel streams with pipelined steps at data link speed
- Put to great use in the calorimeter trigger
  - Reduces a 2D problem down to operations in a single dimension or localised 2D operations



Spatial Pipelining

Because data is buffered in a Time-Multiplexed Trigger and reordered for transmission, it can be sent in the order optimal for processing

If data is arranged suitably, processing begins as soon as the first data arrives…

YOU DO NOT WAIT FOR THE ENTIRE DATA TO ARRIVE BEFORE PROCESSING BEGINS!†

†Bold, red, super-sized and underlined because there are still people in the CMS trigger community who have not understood this and it is important!

Spatial Pipelining – Why bother?

By ordering the data geometrically, this allows all algorithms to be fully pipelined†, which in FPGAs means that:

- The processing is localised
- Fan-outs reduced
- Routing delays minimised
- Register duplication eliminated

Experience with Virtex-7 FPGAs has shown that this is VITAL in mapping the design into these very large FPGAs

This will only get more important as devices get larger in future

†That is, pipelined at the full data rate, not at the bunch-crossing rate

Spatial Pipelining – Why bother?

Full Spatial-Pipelining also allows MAXIMAL logic reuse:

- Case-study: Electron isolation via counting clusters in a ring
- Conventional implementation took >120% of resources available in XC7VX690T FPGA per proposed region
- Spatially-pipelined implementation took 1.5% of resources available in XC7VX690T FPGA per TM node

2 orders of magnitude saving if you do it properly

The Amdahl-Slotnick debate, 1967

"The parallel approach to computing does require that some original thinking be done about numerical analysis and data management in order to secure efficient use.

In an environment which has represented the absence of the need to think as the highest virtue, this is a decided disadvantage"

Daniel Slotnick, 1967

Spatial Pipelining – What's the problem?

You have swapped a spatial axis with the time axis

Positive and negative eta become forward and backward in time

In algorithms, data moving forward in time "interacts" with data moving both forwards and backwards in time

But algorithms are sequential in "real" time – essentially a third time axis

Different granularity data sees time pass at different rates (Time? Which Time?)

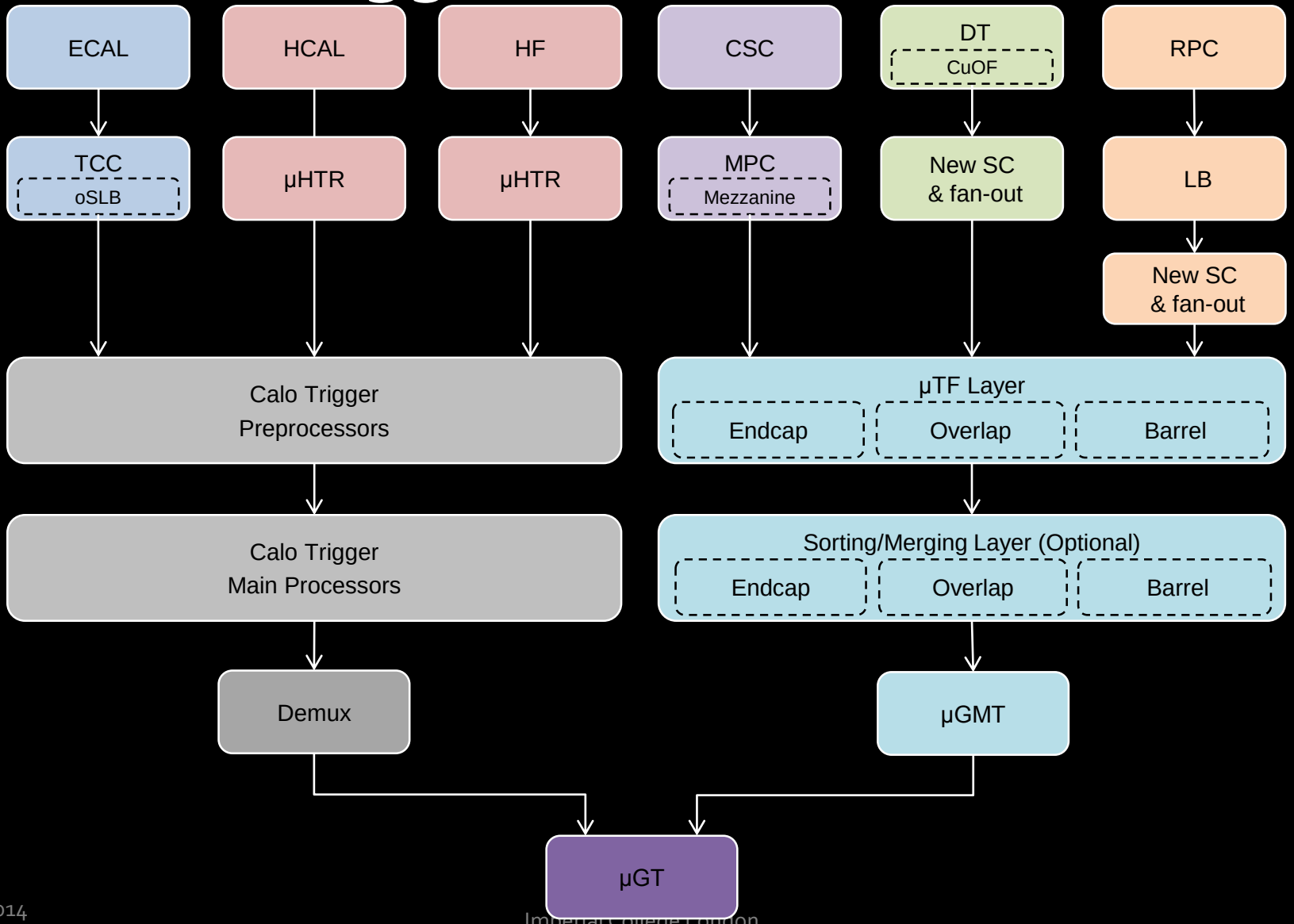
Physical data is in registers and transfers between registers also happen in real time but registers occupy physical space within the FPGA and relative spatial placement is important…

Is your head hurting yet?

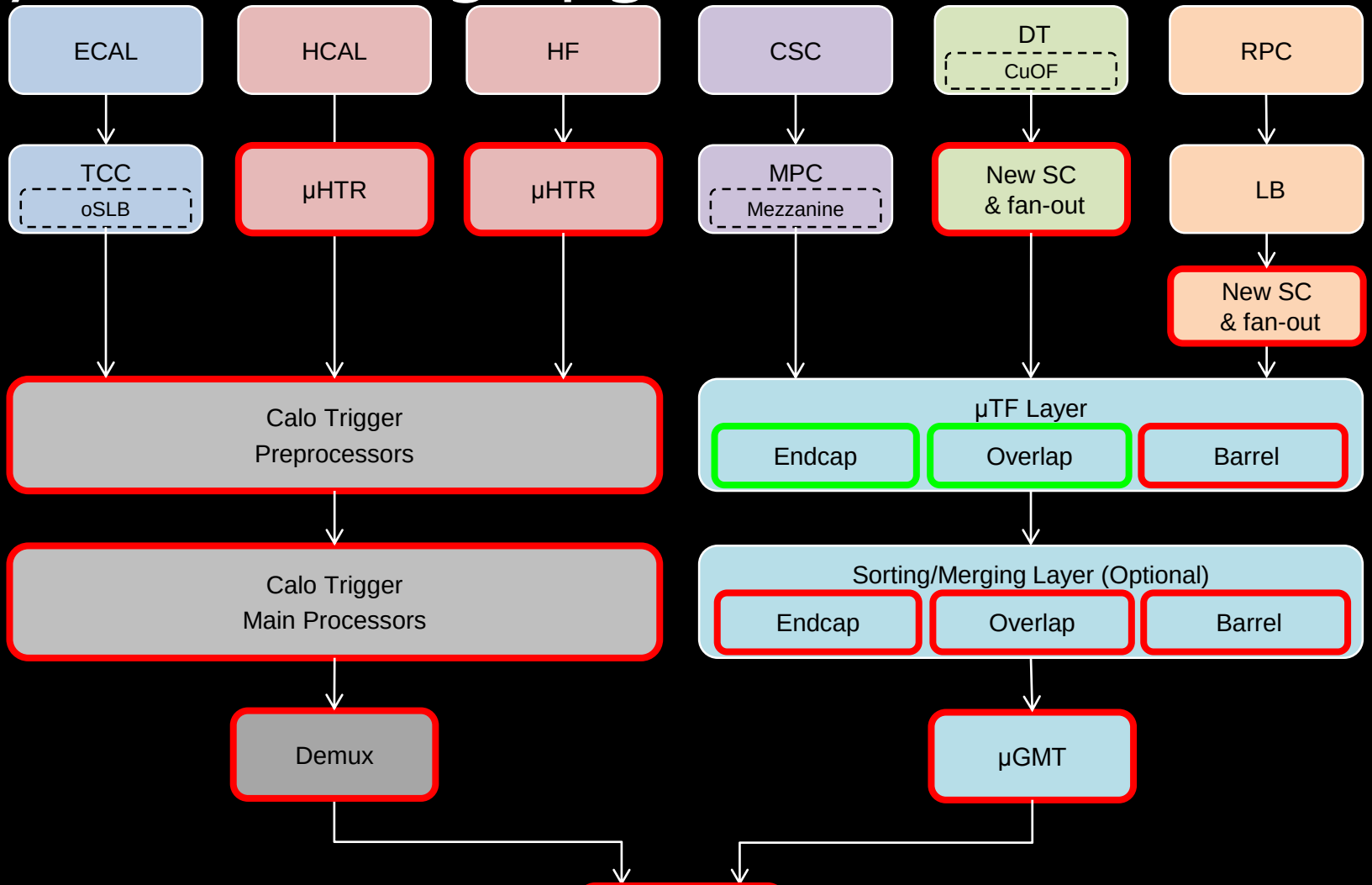
Forget General Relativity – spatially pipelined firmware is where the real fun is to be had

The CMS trigger – Run 2: The Present

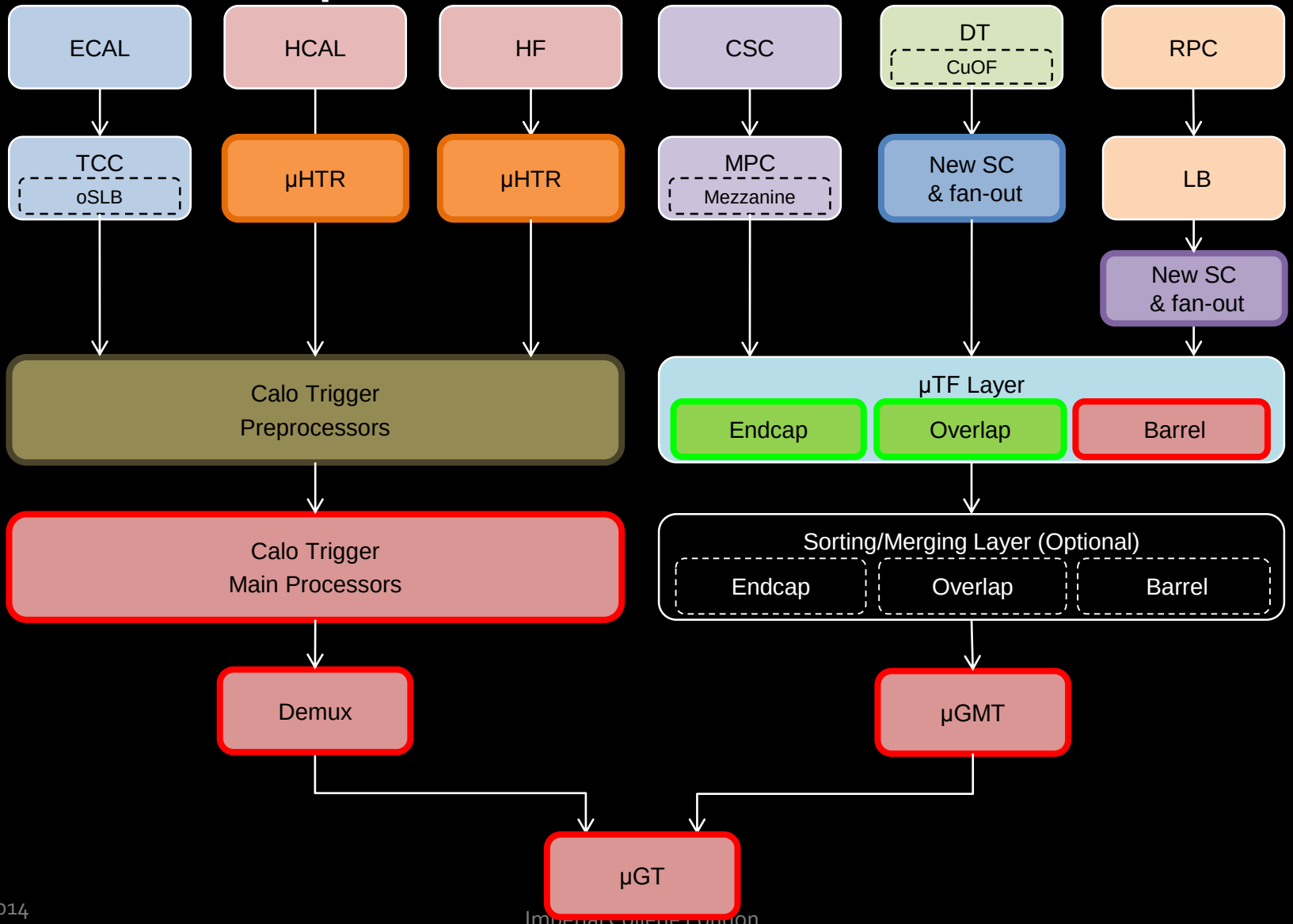
The CMS trigger – Run 2



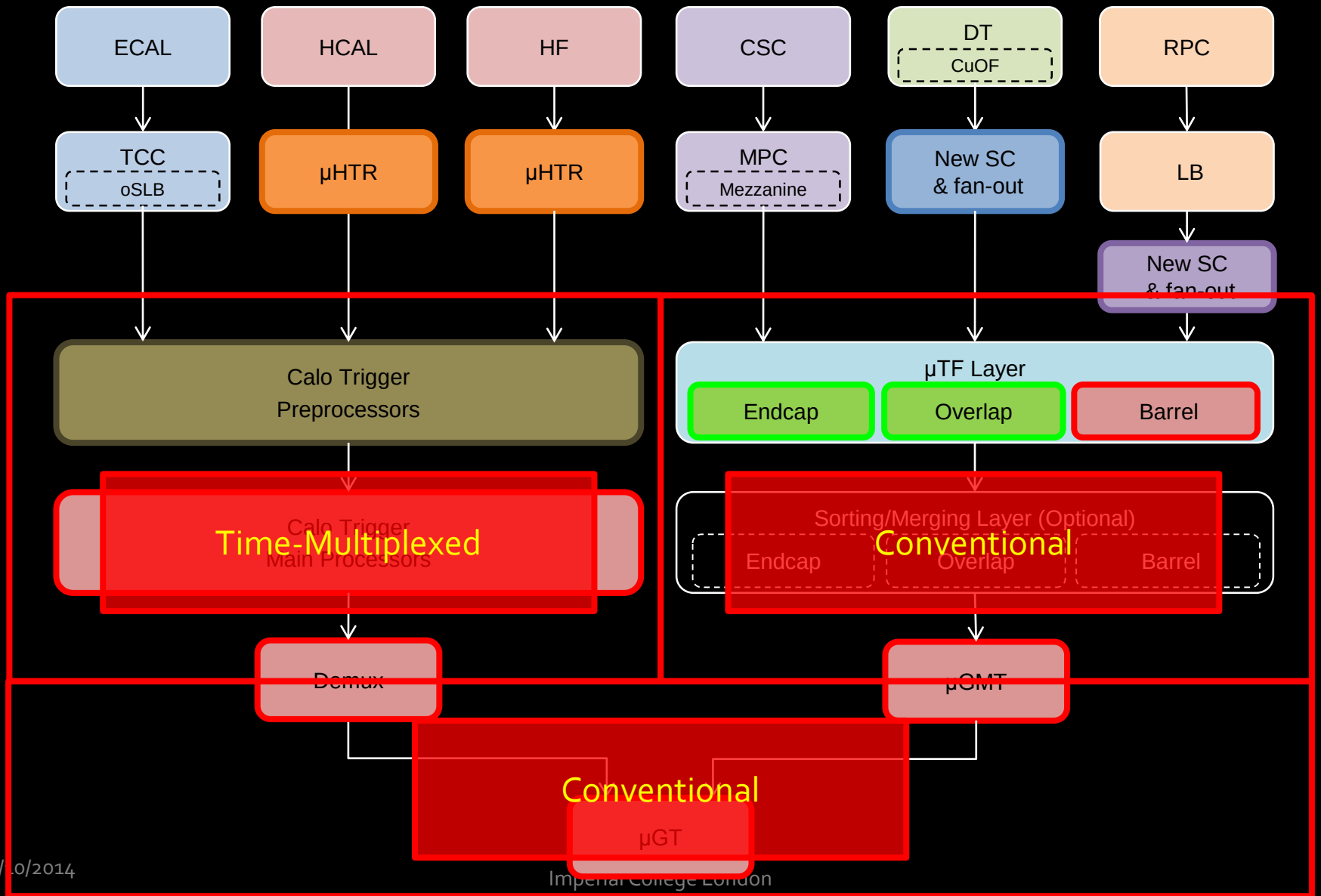
The CMS trigger – Run 2: Systems being upgraded



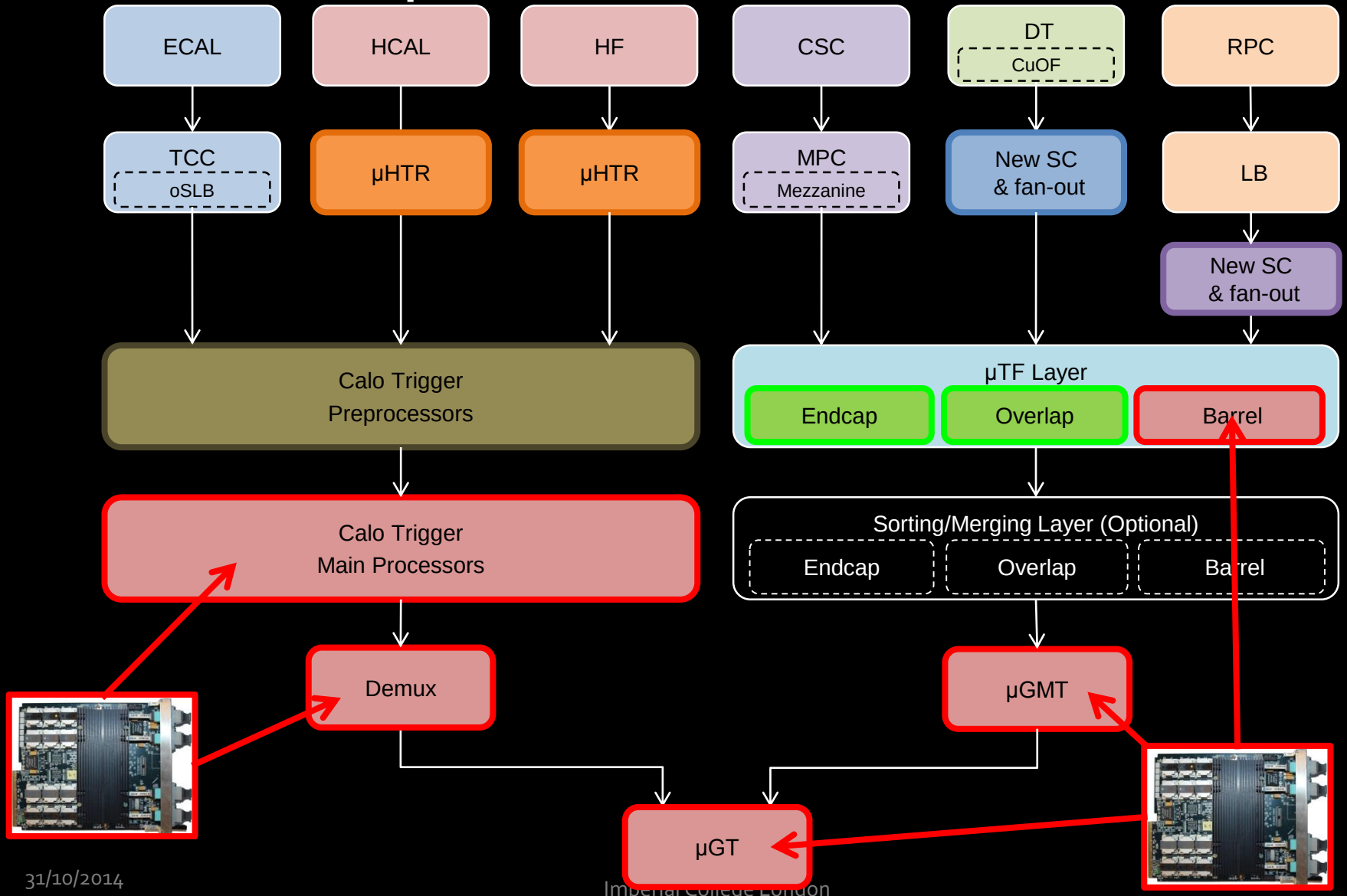
The CMS trigger – Run 2: Hardware platforms



The CMS trigger – Run 2: Architecture

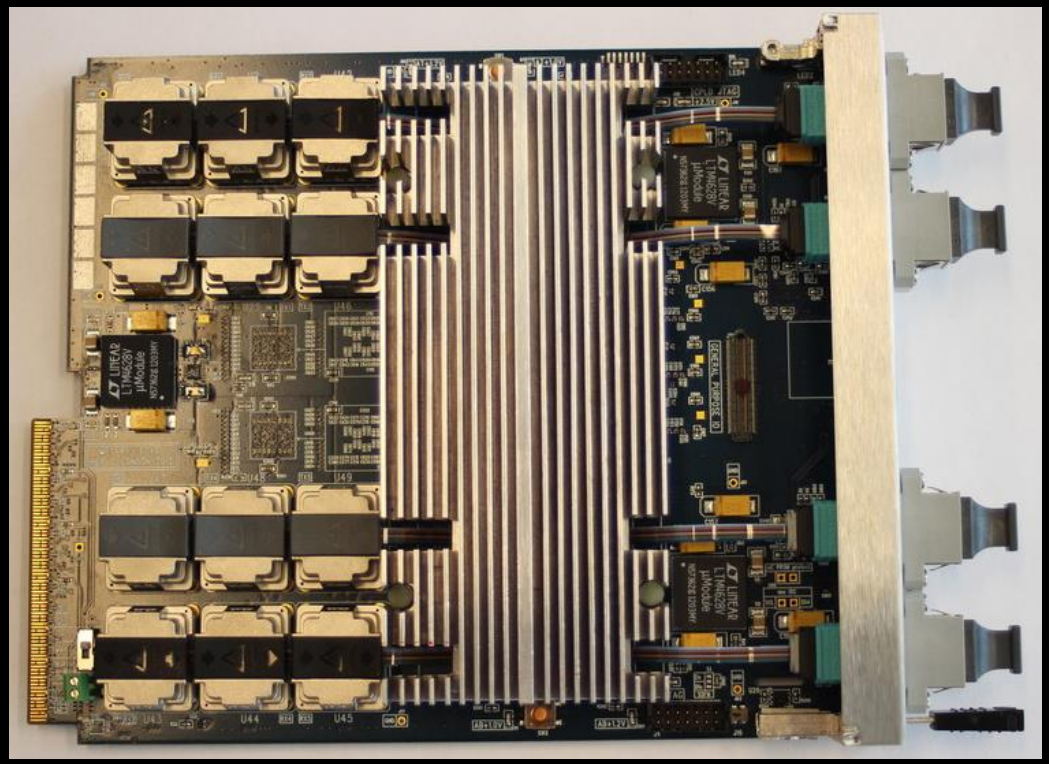


The CMS trigger – Run 2: Hardware platforms

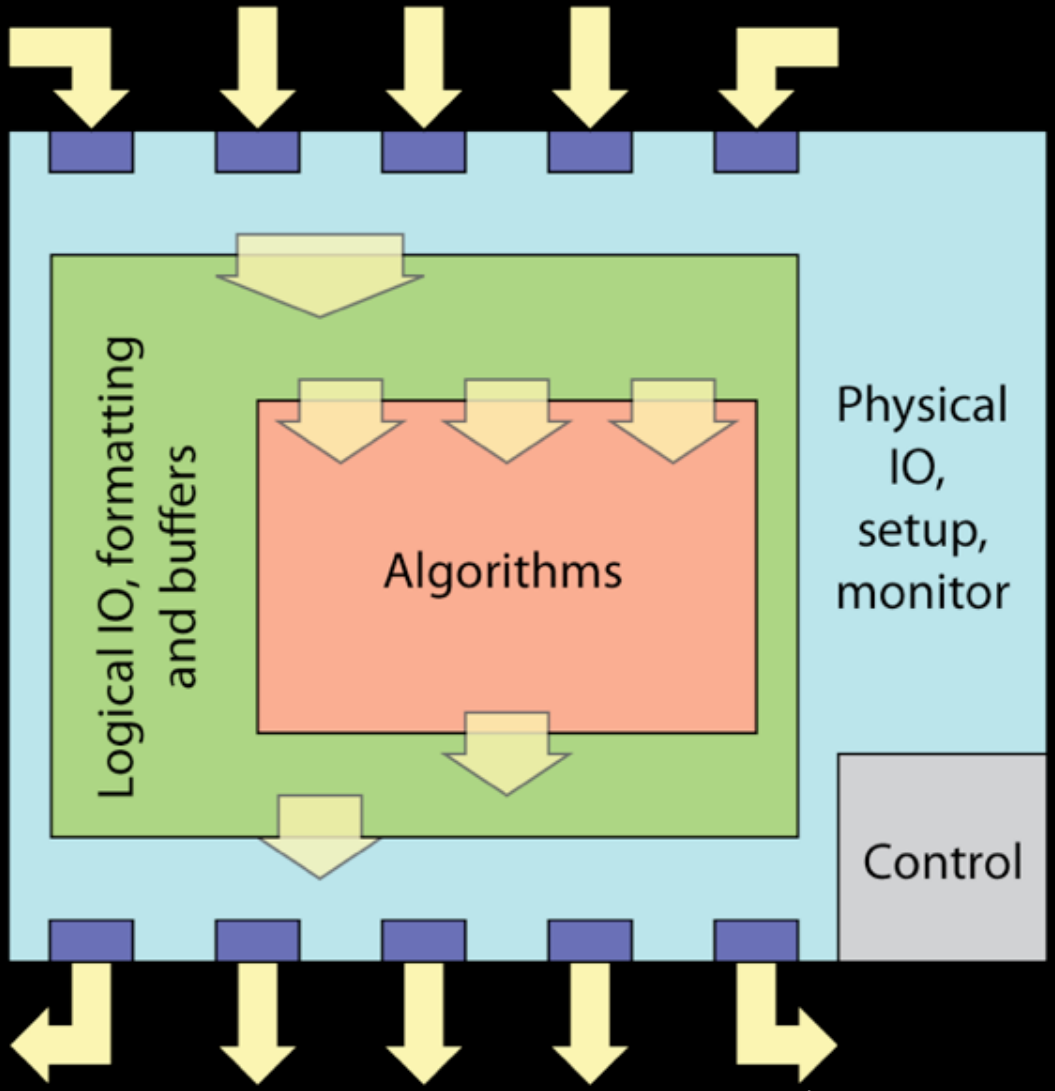


The Master-Processor, Virtex-7 (MP7)

- uTCA form factor
- 72Tx+72Rx 13Gbps optical links
- 0.9 + 0.9Tb/s signal processor
- Xilinx Virtex-7 FPGA
- GbE, AMC13/TTC/TTS, PCIe, SAS, SATA, SRIO
- On-board firmware repository
- Pin-compatible FPGAs allow cost-performance balance
- 2×144Mbit 550MHz QDR RAM (optional)



“Base Firmware” Concept



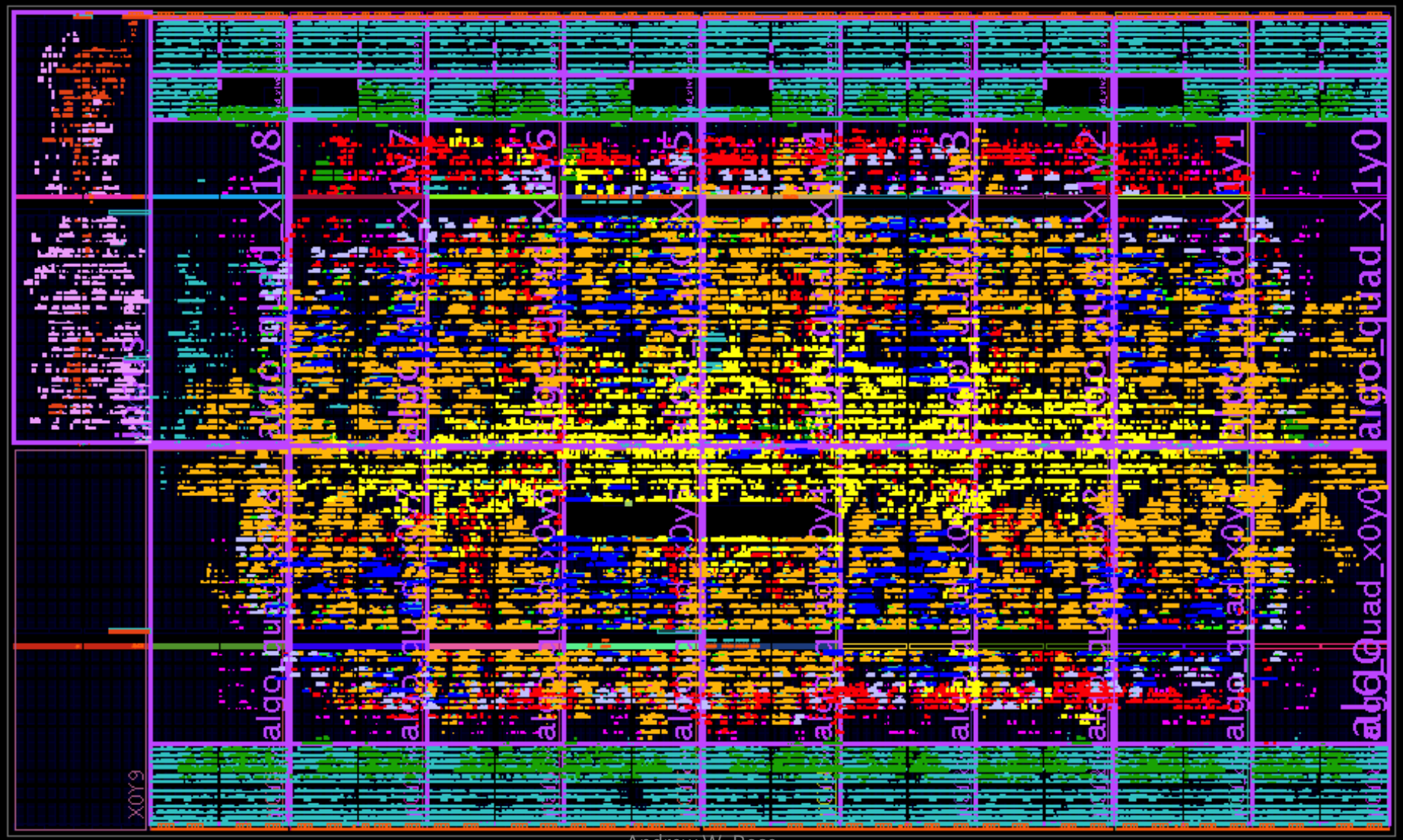
Trigger emulator
Open Development

System setup and test
Common across trigger

Low-level control
Hardware-specific development

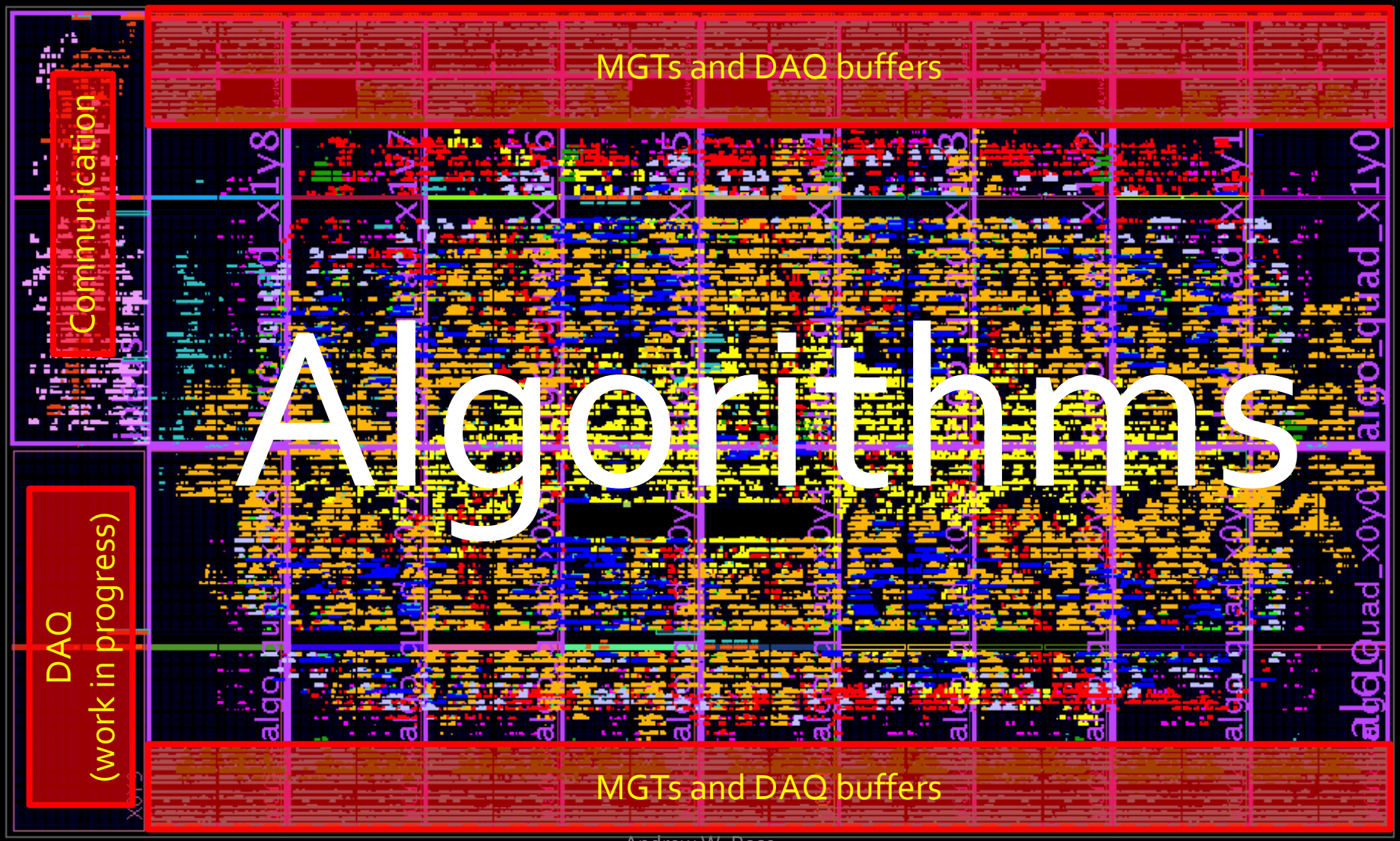
uTCA infrastructure
CMS common standard

Floorplan of FPGA



Floorplan of FPGA

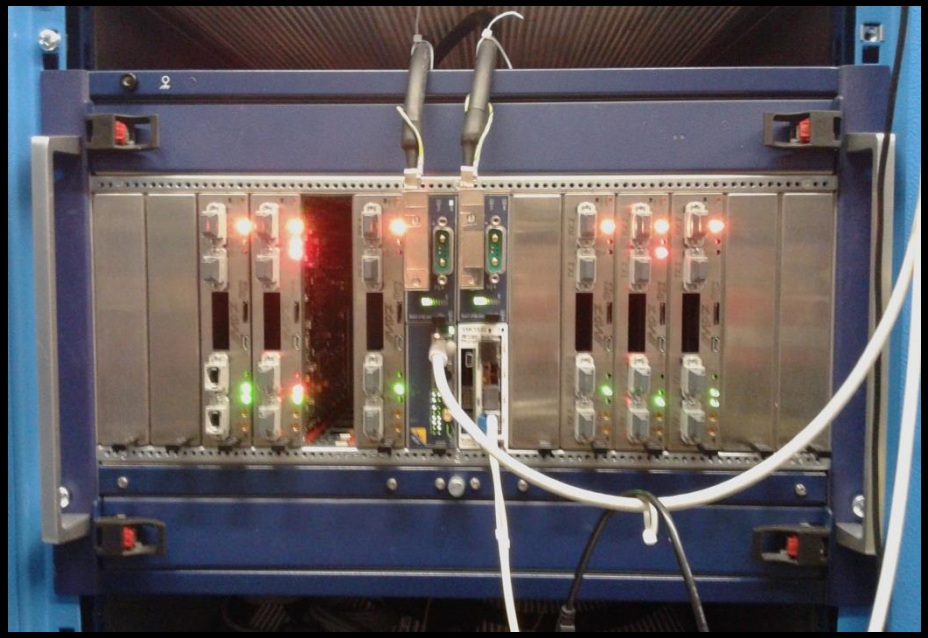
Total LUT usage: 46%
240 MHz clock

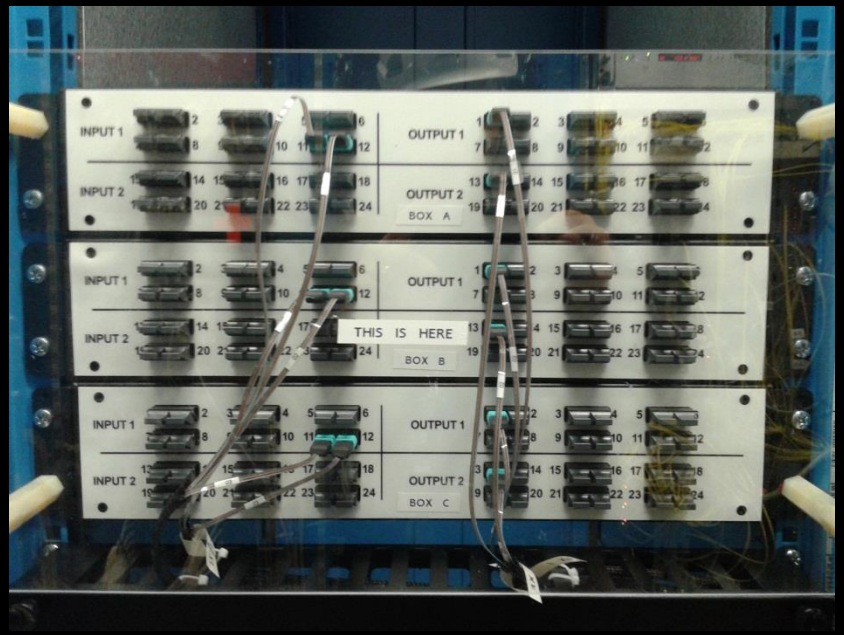


Lessons learned

- Floor-planning
  - Huge impact on algorithm design
  - Structure the algorithm to map optimally onto the FPGA
  - Reduces risk that after many hours of routing 6 million nets just 1 or 2 fail to meet timing - exceedingly annoying
  - Significant timing improvement
  - Only viable if signals remain relatively local
- Full pipelining of algorithms is essential - even relatively innocuous looking fan-outs in chips this large have the potential to kill off the entire design
- If you want the most out of your FPGA you really need to think creatively about how you structure your data

The CMS trigger – Run 2: Reality

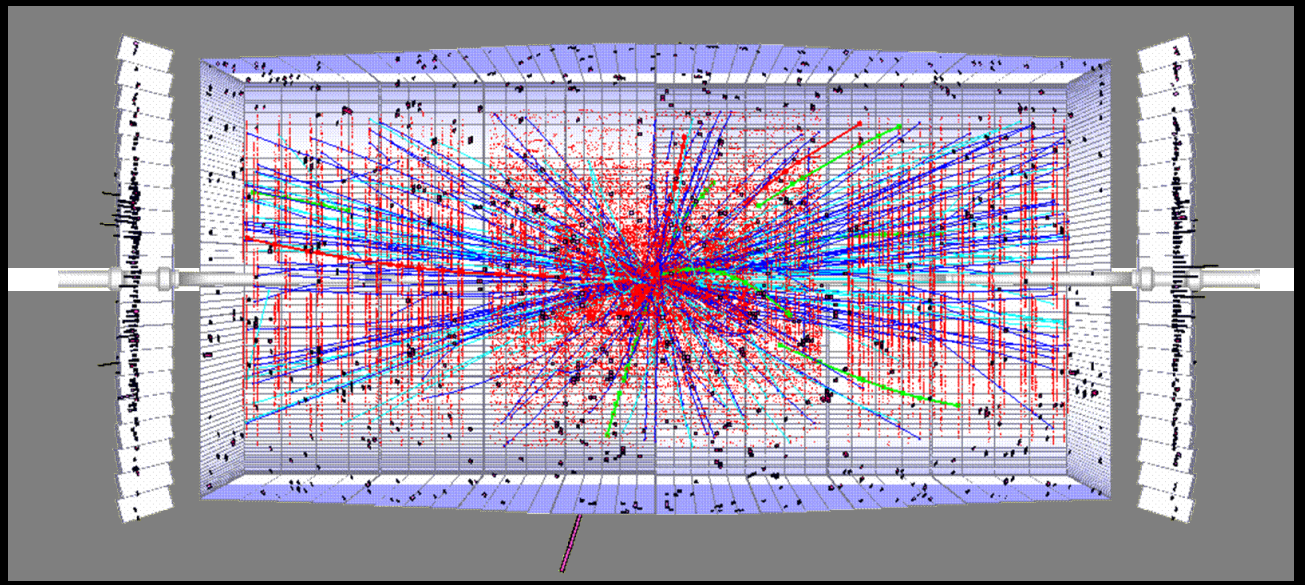




The CMS trigger – Run 3: The Future

The HL-LHC

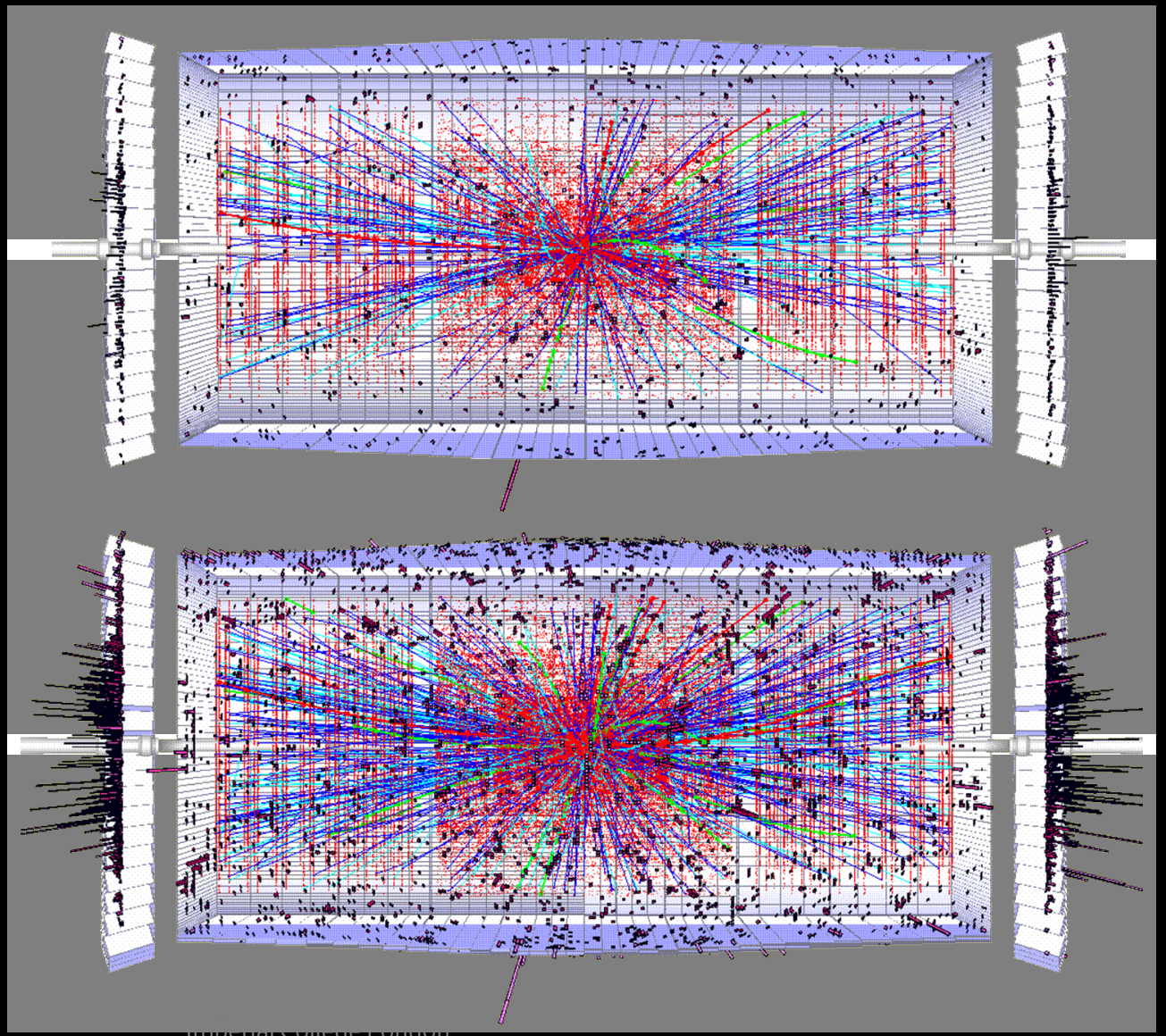
LHC Run-2
$L = O(10^{34})\ cm^{-2}s^{-1}$
O(10) interactions/bx



The HL-LHC

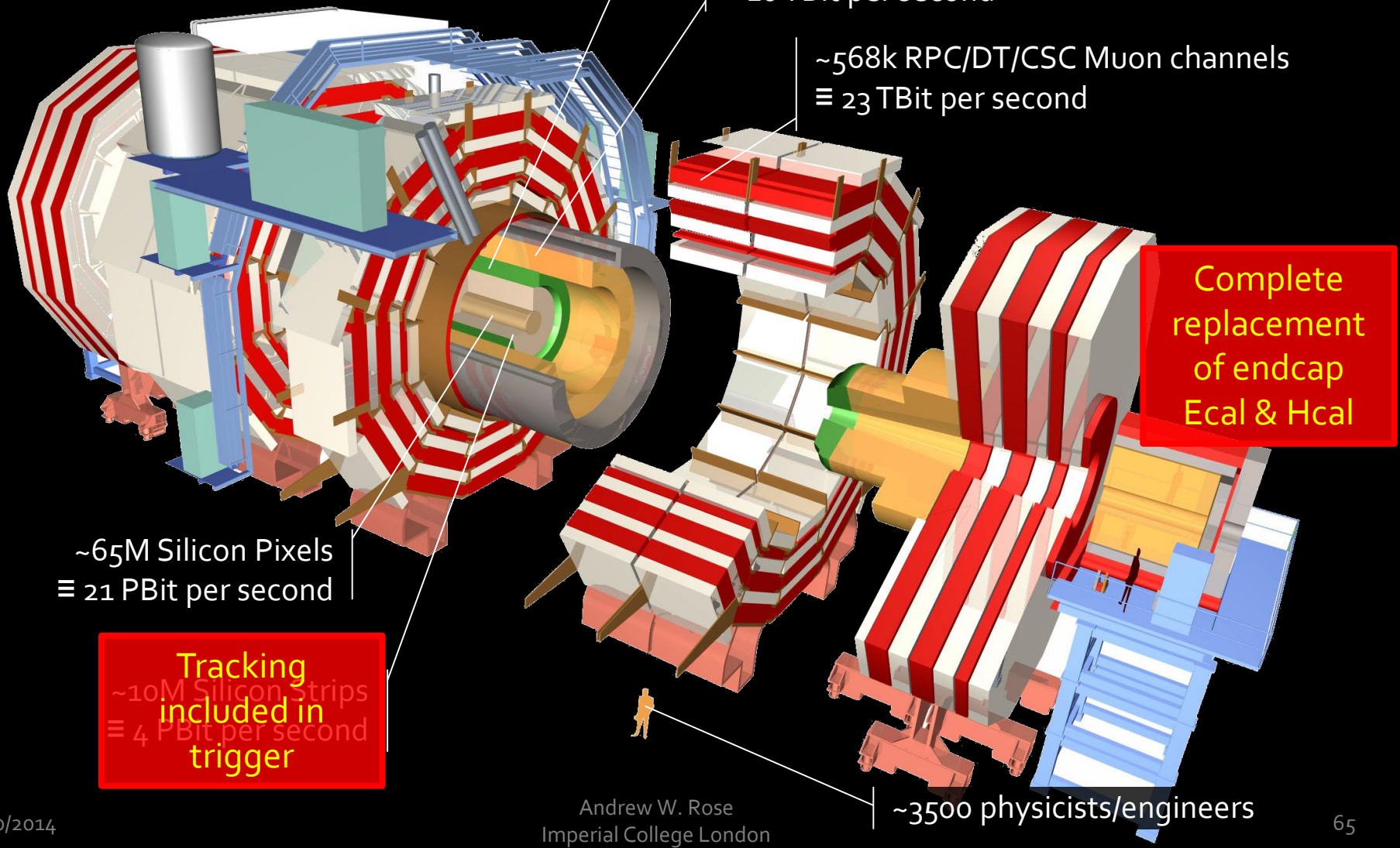
LHC Run-2
$L = O(10^{34})\ cm^{-2}s^{-1}$
O(10) interactions/bx

LHC Run-3
$L = O(10^{35})\ cm^{-2}s^{-1}$
O(100) interactions/bx

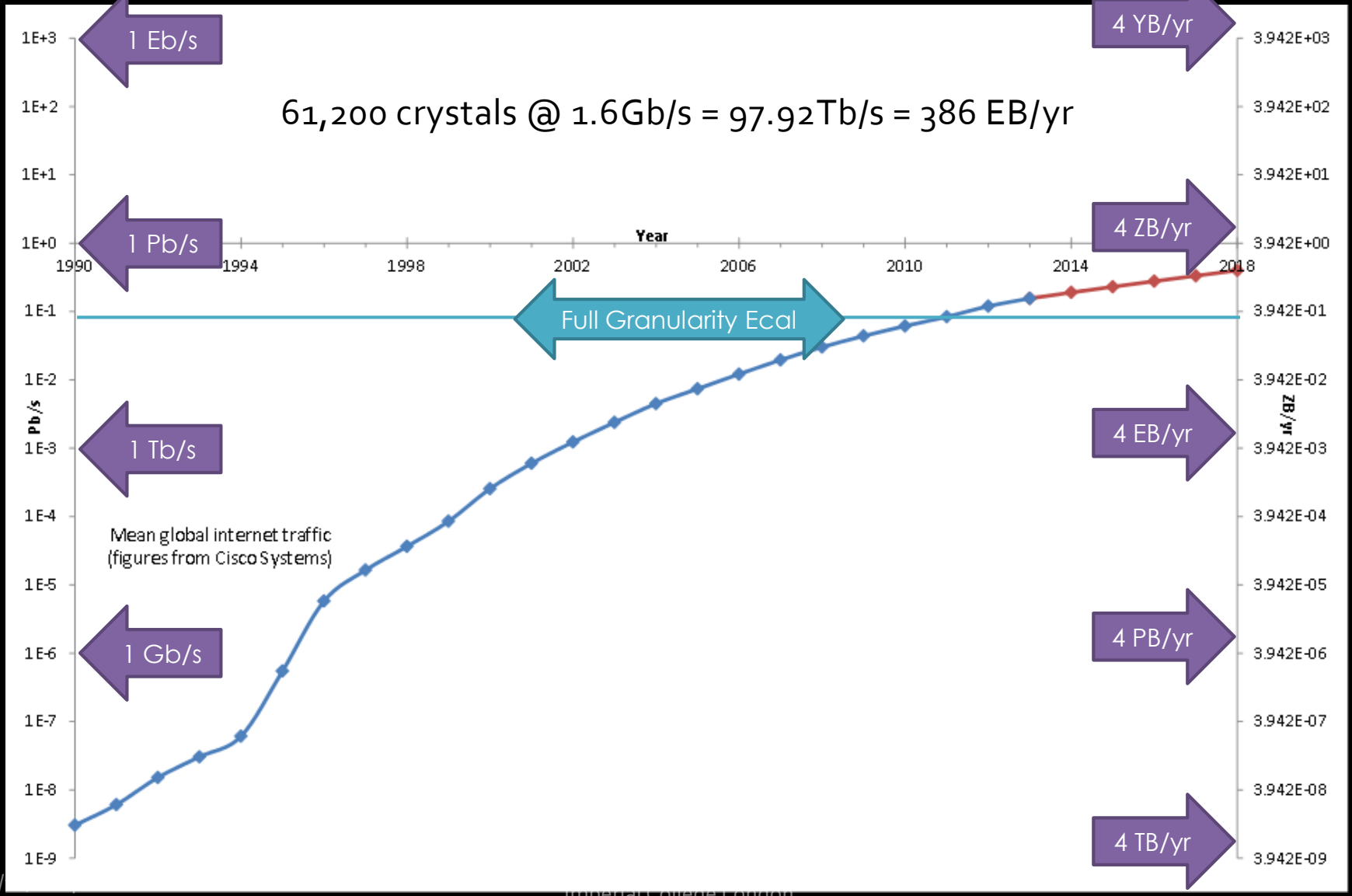


The CMS detector

Upgrades relevant to triggering



Big Data: Full granularity Ecal



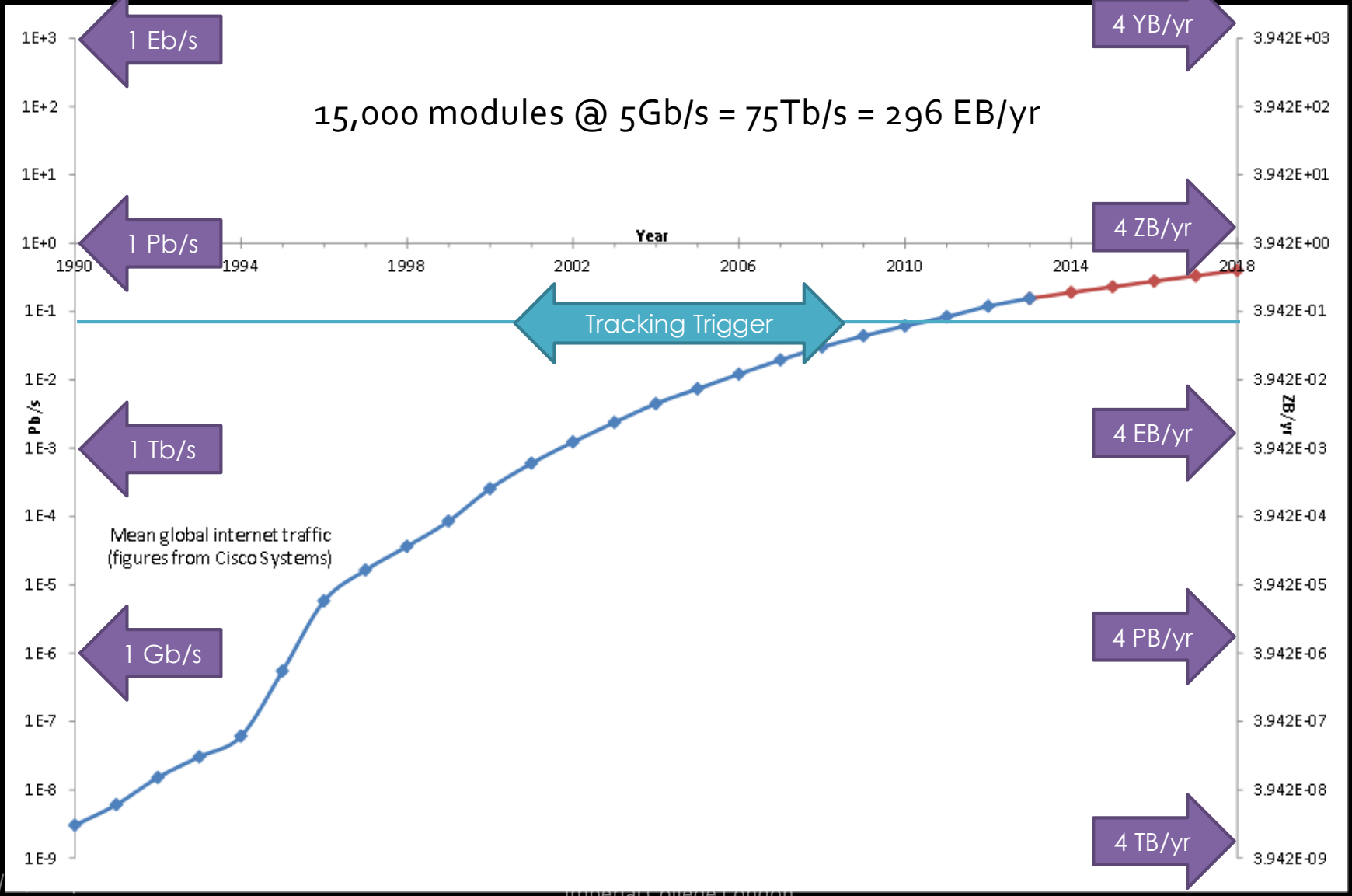
Full granularity Ecal: Algorithms

Data is still laid out on a 2D rectilinear grid

Essentially the same as what we do now, only with 5 times the resolution in each direction

Currently, the major question which needs answering is what do the physicists want to do with the extra resolution!

Big Data: Tracking trigger



Tracking trigger: Algorithms

Track-finding in a high-occupancy environment is HARD

A lot of fake coincidences, noise, overlaps…

Inside solenoid – 4T magnetic field means highly curved tracks

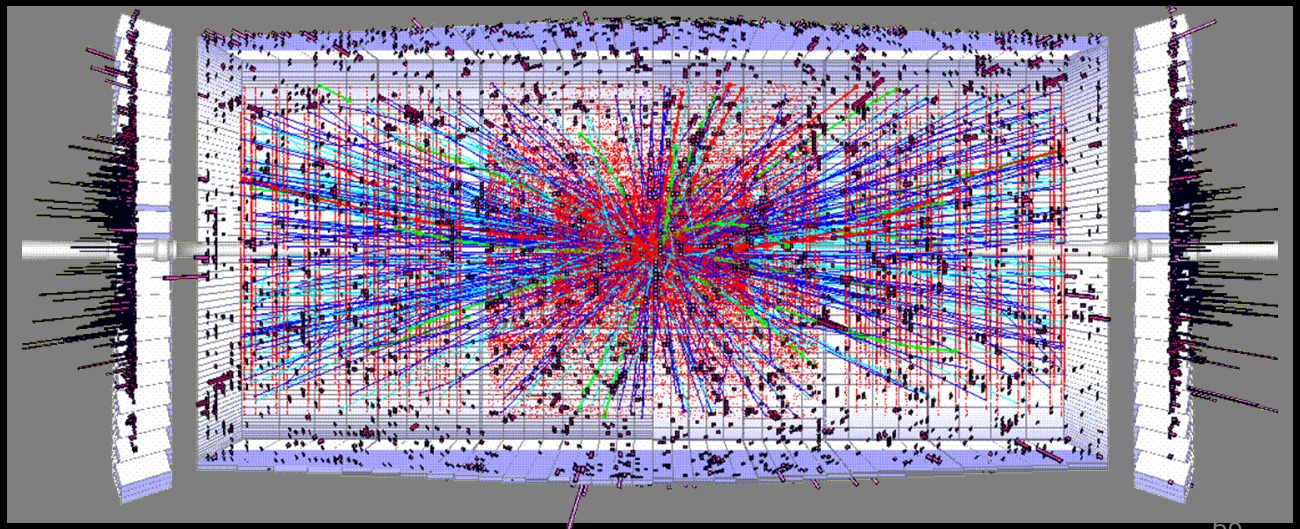
Hough transform

Road search

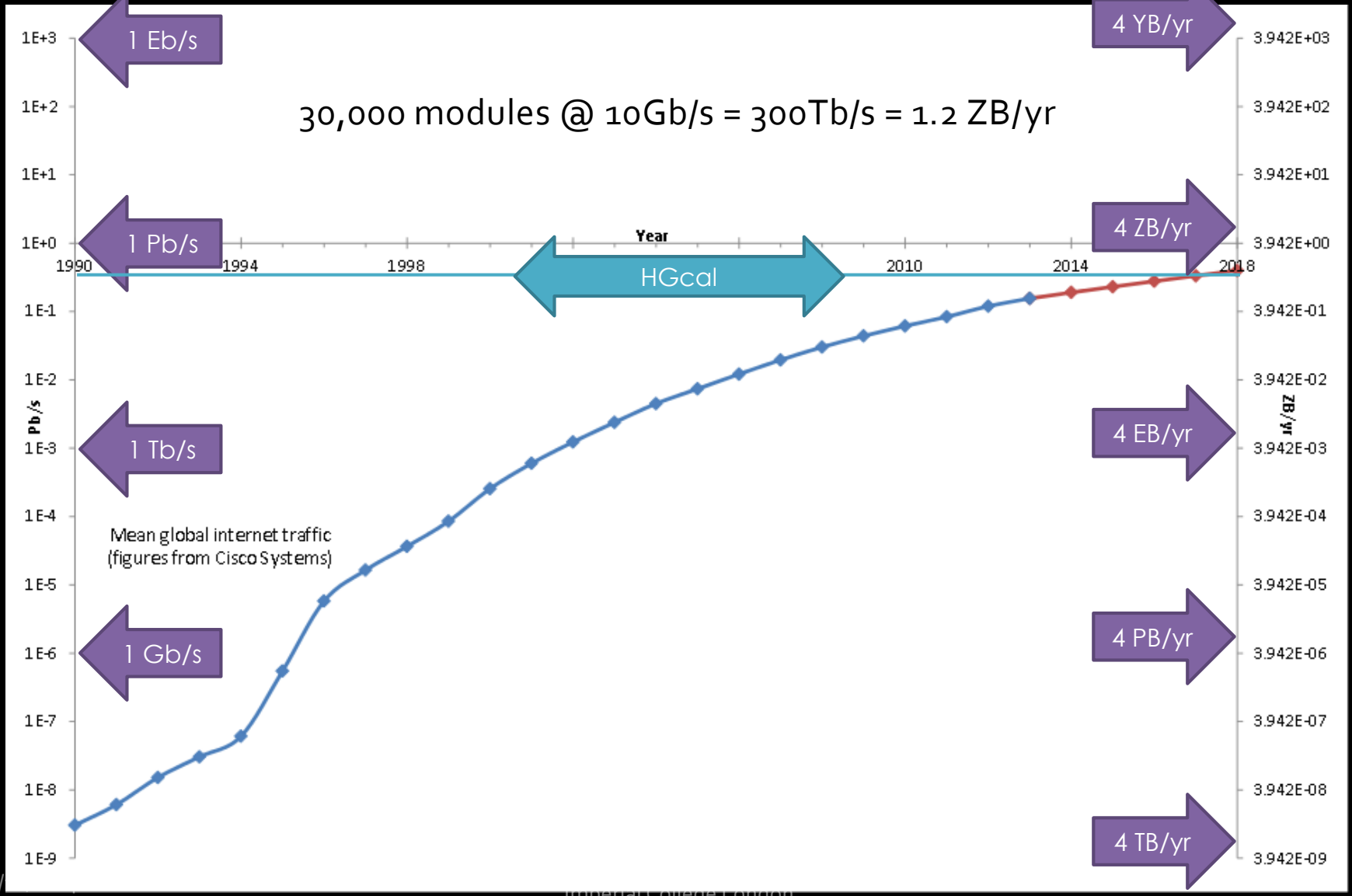
Projective Iteration

Other?!

?



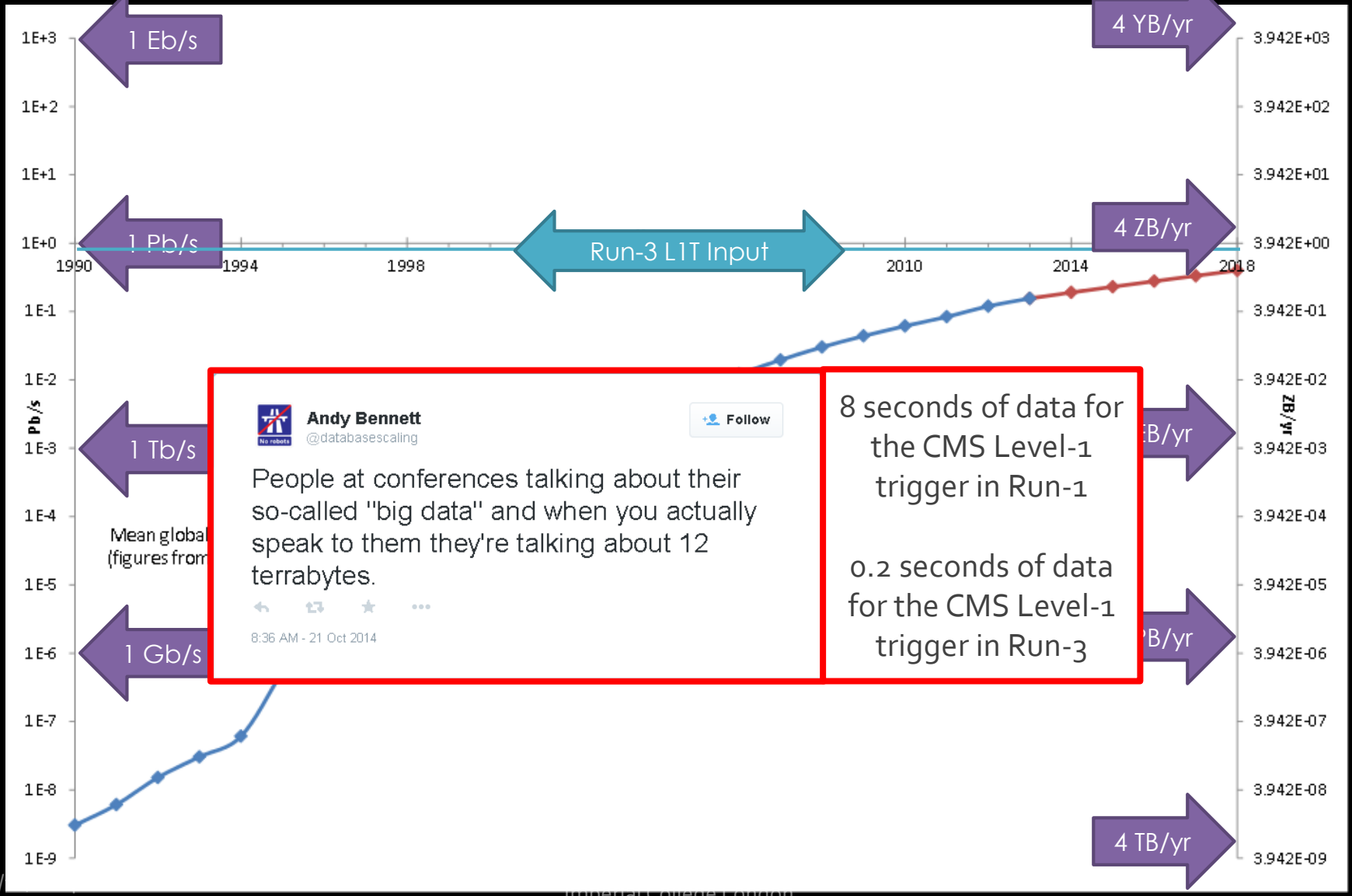
Big Data: High-granularity endcap calorimeters (one option)



High-granularity endcap calorimeters : Algorithms

Million dollar question!

Big Data:
Tech buzz-word of the moment



How to approach the problem

How to approach the problem

- All new systems will use the GBT link
- All sending data at the same order of magnitude
- It would be sensible for all systems to use a common back-end to receive the data

How to approach the problem

- All new systems will use the GBT link
- All sending data at the same order of magnitude
- It would be sensible for all systems to use a common back-end to receive the data

- We have a system architecture which is scalable to rise to the challenge
- We have a system architecture which simplifies boundary handling and is agnostic to the data content
- It would be sensible for all systems to use a common trigger platform and common trigger architecture to process the data

The problems needing solving

- To overcome the hardware-centric mindset
- Requires a model of collaborative firmware development which is as simple (and rigorous) as collaborative software development

The problems needing solving

- To overcome the hardware-centric mindset
- Requires a model of collaborative firmware development which is as simple (and rigorous) as collaborative software development
- To overcome the old organizational model

The problems needing solving

- To overcome the hardware-centric mindset
- Requires a model of collaborative firmware development which is as simple (and rigorous) as collaborative software development
- To overcome the old organizational model
- That task could make processing 0.5Pb/s look easy

The problems needing solving

- To overcome the hardware-centric mindset
- Requires a model of collaborative firmware development which is as simple (and rigorous) as collaborative software development
- To overcome the old organizational model
- That task could make processing 0.5Pb/s look easy

Thank you