Modeling and Computing in Elementary Particle Physics

From particle tracks to physics understanding in ATLAS

Introduction – Modeling - Computing

CHATS - CERN, October 2011 - Hans von der Schmitt

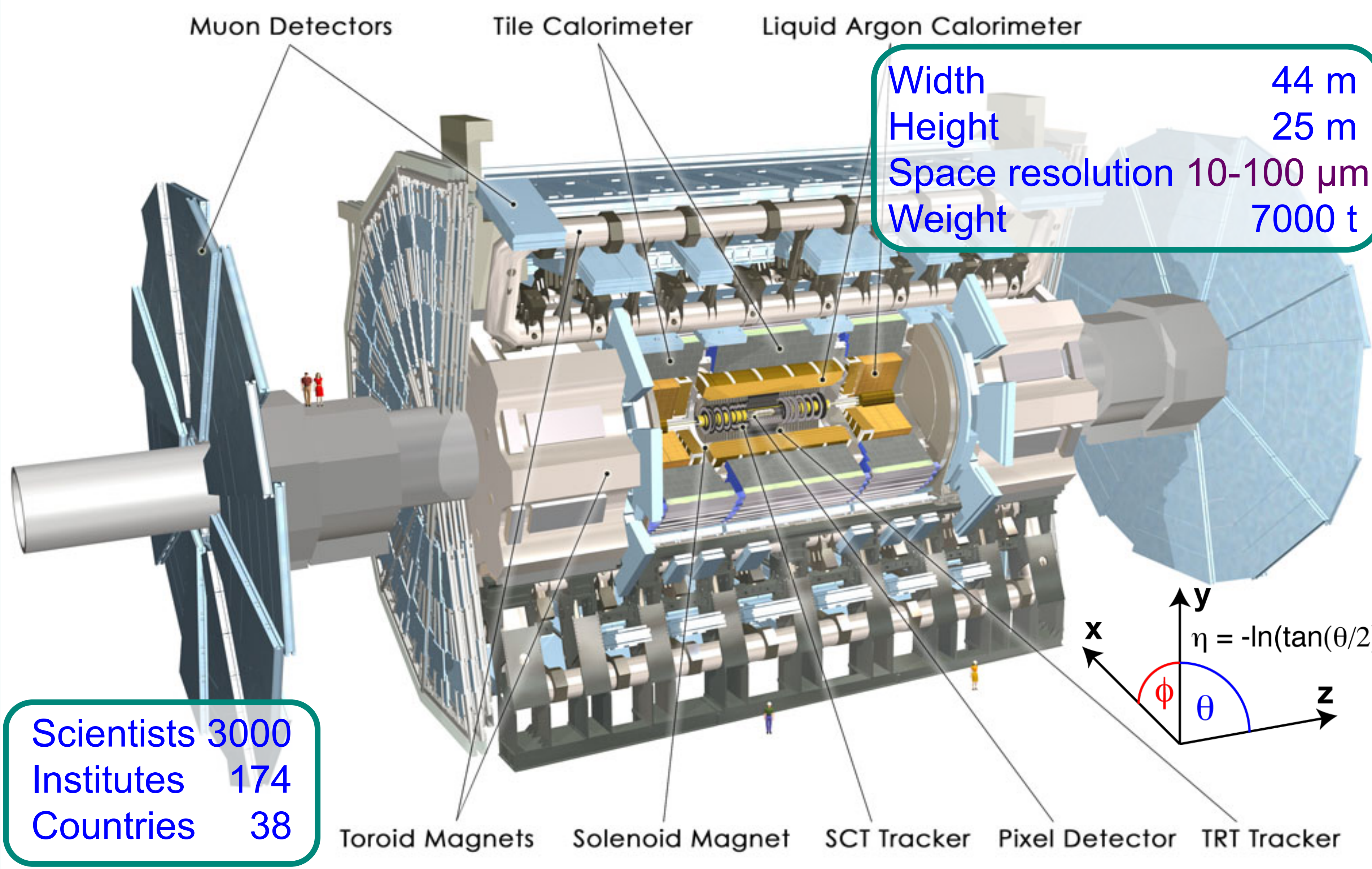
The Large Hadron Collider

Experiment ATLAS located close to CERN main site

Relations between Superconductivity and Particle Physics – and a little bit of history

- Superconductivity is *applied* at large scale in HEP
  - in accelerators and experiments
  - for magnets and for RF cavities
- Interplay in development of S.C. and HEP *theory*
  - now is 50 years after BCS who provided a dynamic, microscopic explanation for superconductivity
  - Ginzburg, Landau; Stückelberg, Anderson …
  - the symmetry of the laws of electromagnetism have to be broken somehow to accommodate superconductivity
  - this recognition of *spontaneous symmetry breaking* produced a revolution in elementary particle physics
  - Nambu; Goldstone, Salam, Weinberg … related symmetry breaking with new gauge particles – massless particles though, excluded by experiment, unless a *local* symmetry is broken
  - just how is electroweak symmetry broken in particle physics? By the Higgs* field? Or dynamically after all? (* and Englert, Brout, Guralnik, Hagen, Kibble)
  - still a major question - at LHC …

ATLAS overview



Width	44 m
Height	25 m
Space resolution	10-100 μm
Weight	7000 t

Scientists	3000
Institutes	174
Countries	38

$$\eta = -\ln(\tan(\theta/2))$$

ATLAS data volumes

Inner Detector	Channels	Fragment size/kB
Pixels	80×10^{6}	60
SCT	6.2×10^{6}	110
TRT	3.7×10^{5}	307

Muon Spectrometer	Channels	Fragment size/kB
MDT	3.7×10^{5}	154
CSC	6.7×10^{4}	256
RPC	3.5×10^{5}	12
TGC	4.4×10^{5}	6

Calorimeter	Channels	Fragment size/kB
LAr	1.8×10^{5}	576
Tile	10^{4}	48

Trigger		Fragment size/kB
LVL1		28

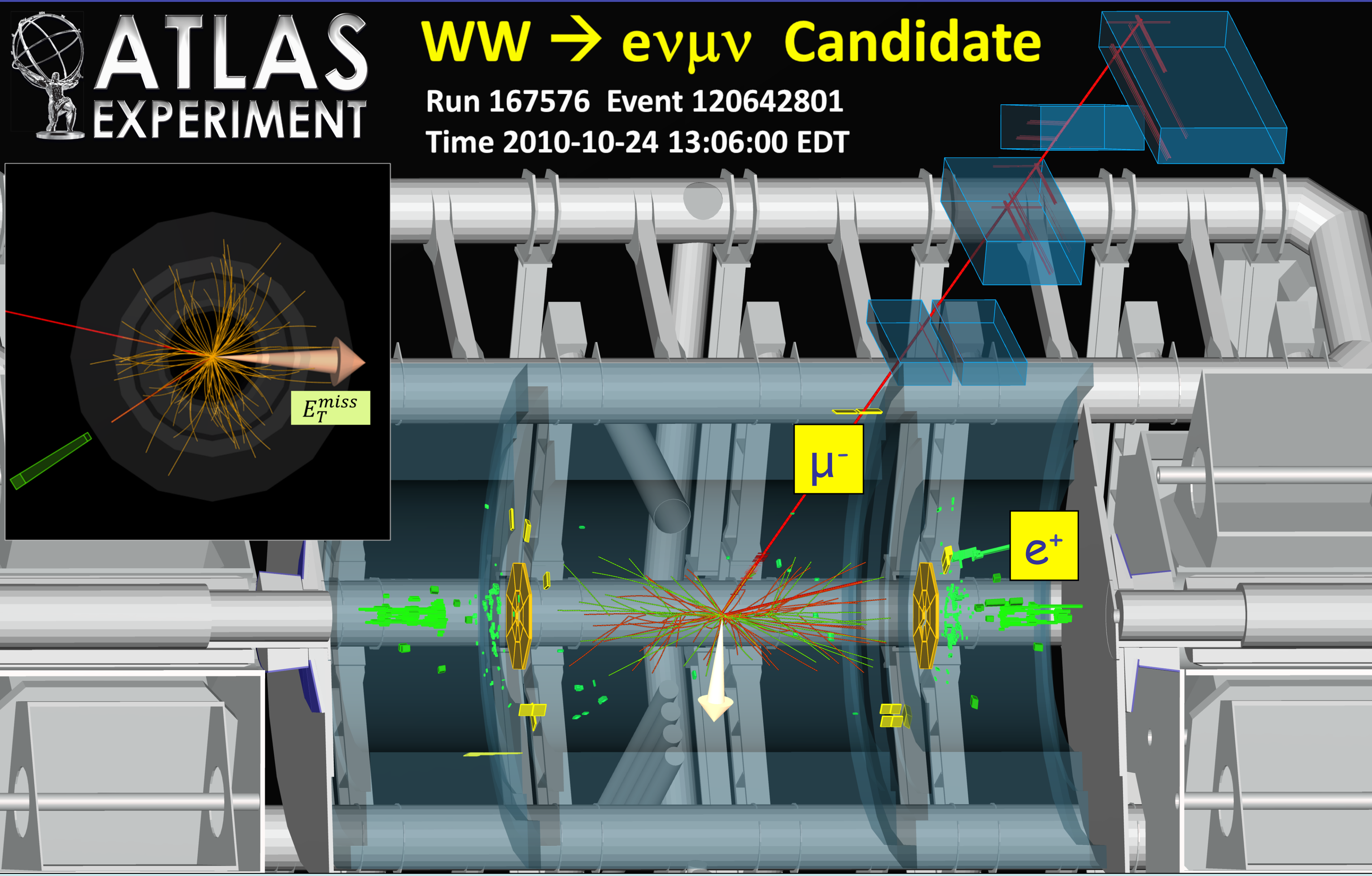
ATLAS event size: ~1.5 Mbytes
Recording rate: ~400 Hz
~100 Million electronic channels

Rate to permanent storage: 500 MB/sec

3 PB/year go to reconstruction + analysis on the worldwide LHC Grid computing facilities

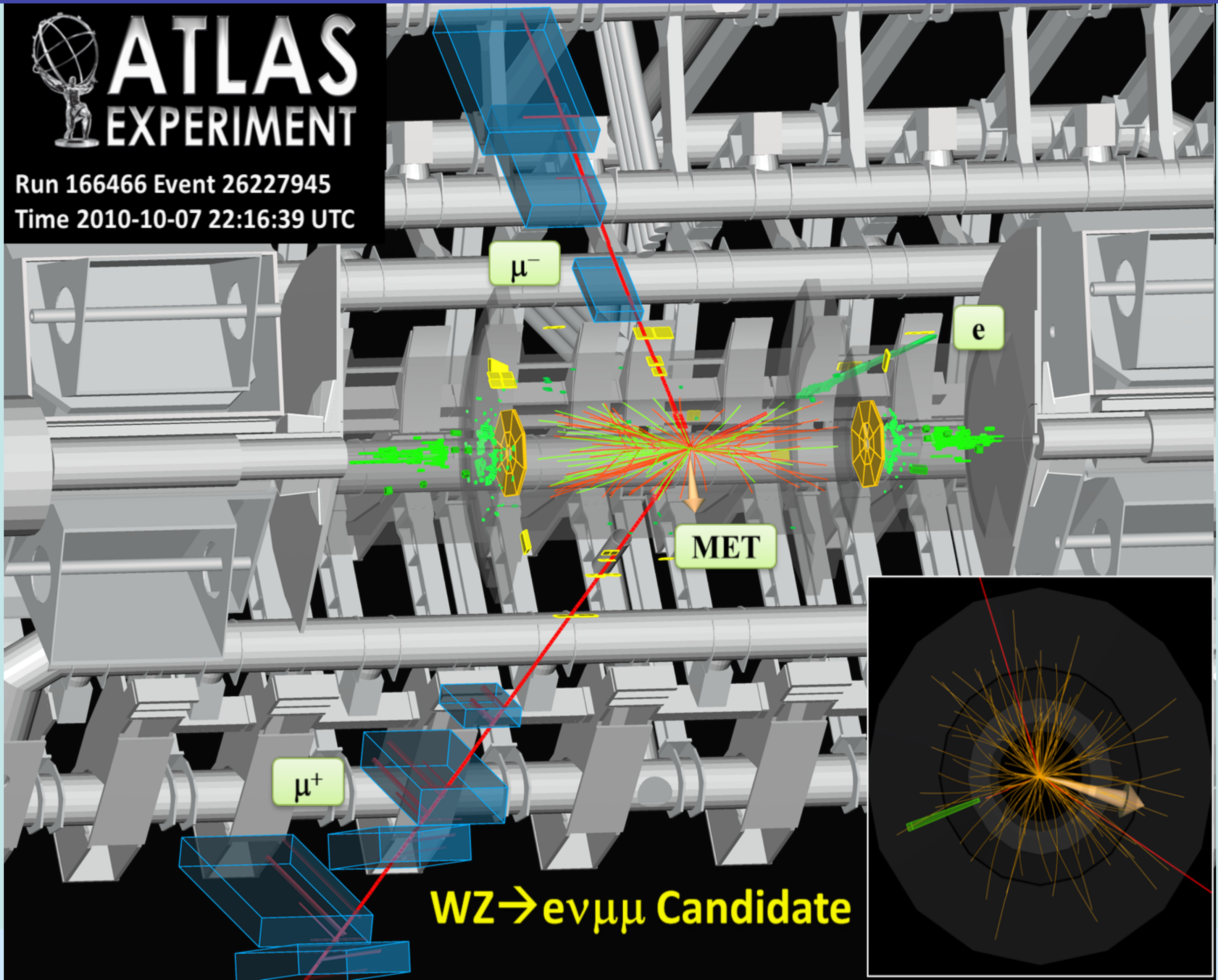
Some interesting events: $W^+W^- \rightarrow e^+\nu\ \mu^-\nu$ candidate

p_T (e)~ 20 GeV, p_T (μ)~ 68 GeV, E_T^{miss} ~ 70 GeV

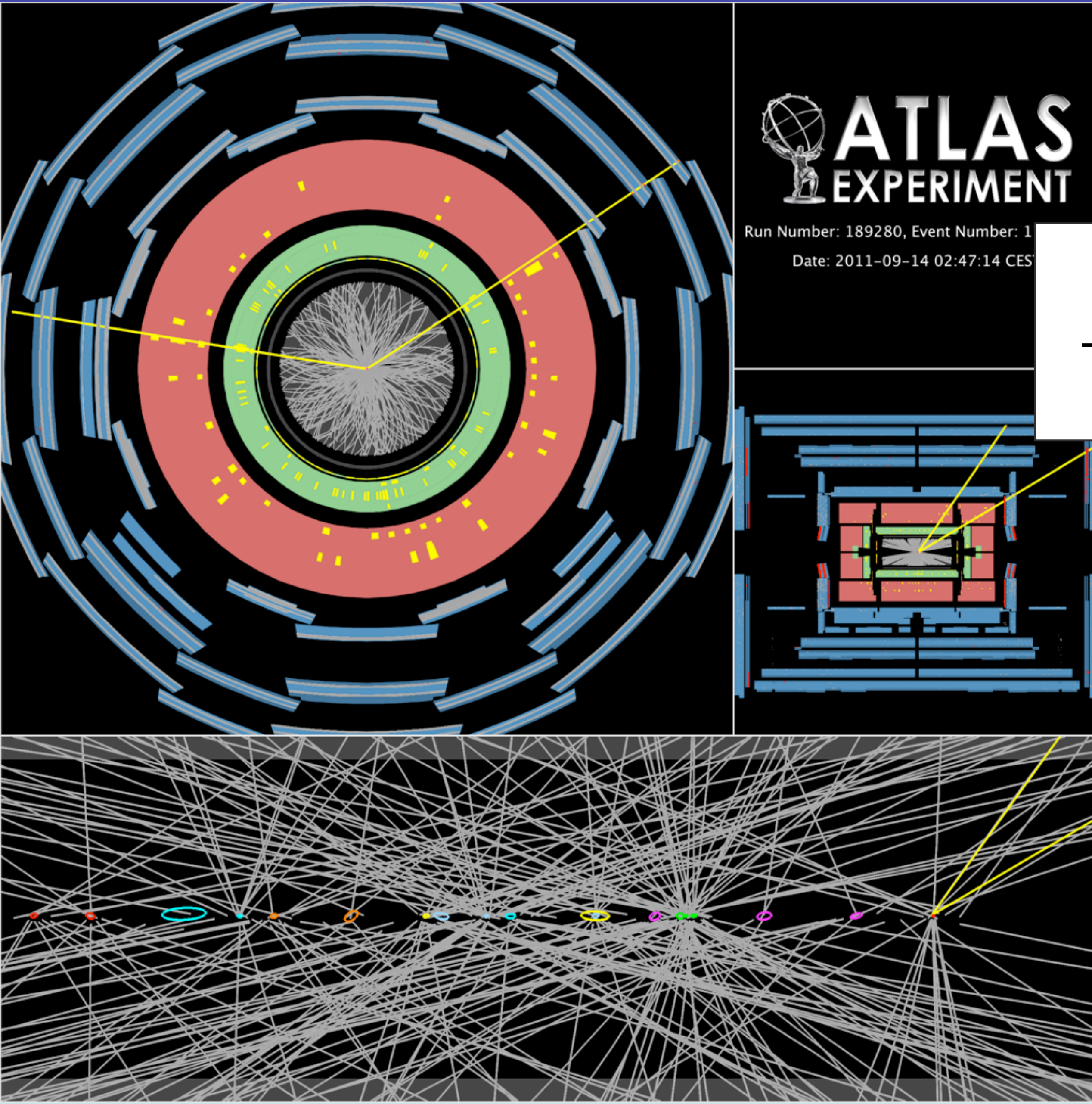


WZ → eν μμ candidate

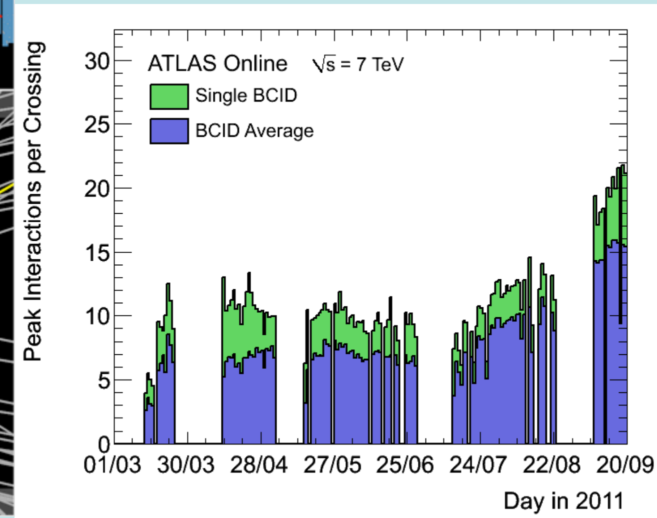
$p_T(\mu^+)$ = 65 GeV, $p_T(\mu^-)$ = 40 GeV, $p_T(e)$ = 64 GeV, E_T^{miss} = 21 GeV



High demands especially in particle tracking close to the interaction point



Example of Z → μμ decay
with 20 primary vertices
Total range along the beam axis
is ~ ± 15 cm

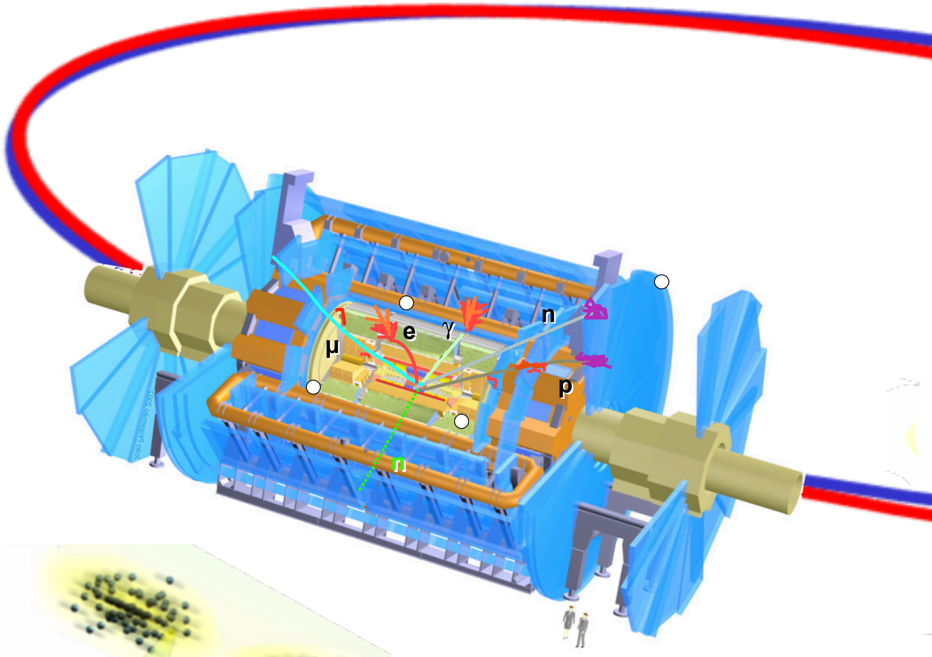


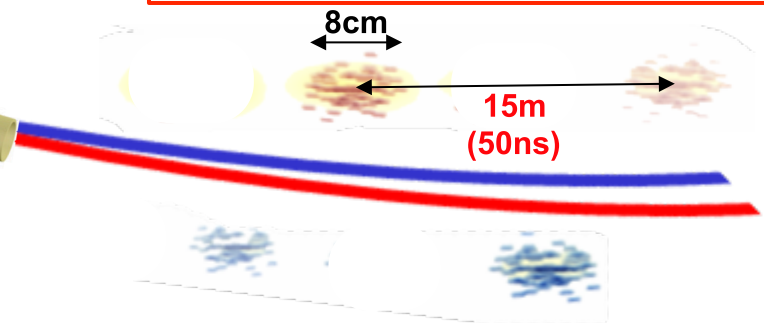
Modeling

Proton-Proton collisions at LHC – parameters 2011

LHC parameters 2011:

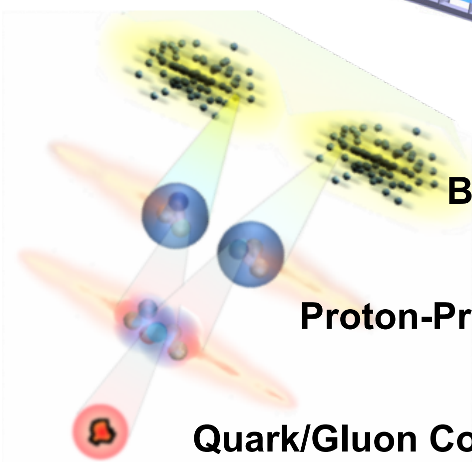
3.5 TeV	Proton energy
$3*10^{33} cm^{-2}s^{-1}$	Luminosity
1380	Bunches per beam
$1.3*10^{11}$	Protons per bunch
50 ns	Bunch spacing
34 μm	Beam spot size
3…30	Interactions/bunch crossing





0.3 A
per beam

120 MJoule
per beam

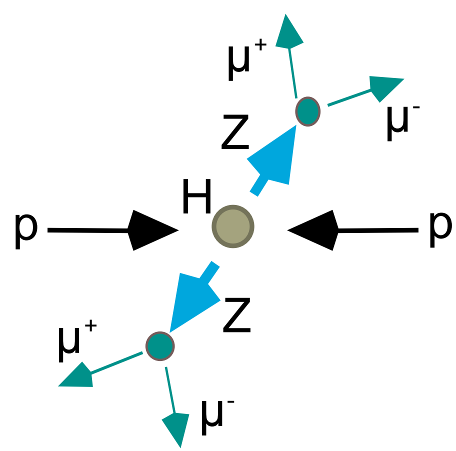


Bunch Crossings $2x10^{7}$ Hz

Proton-Proton Collisions $0.2*10^{9}$ Hz

Quark/Gluon Collisions

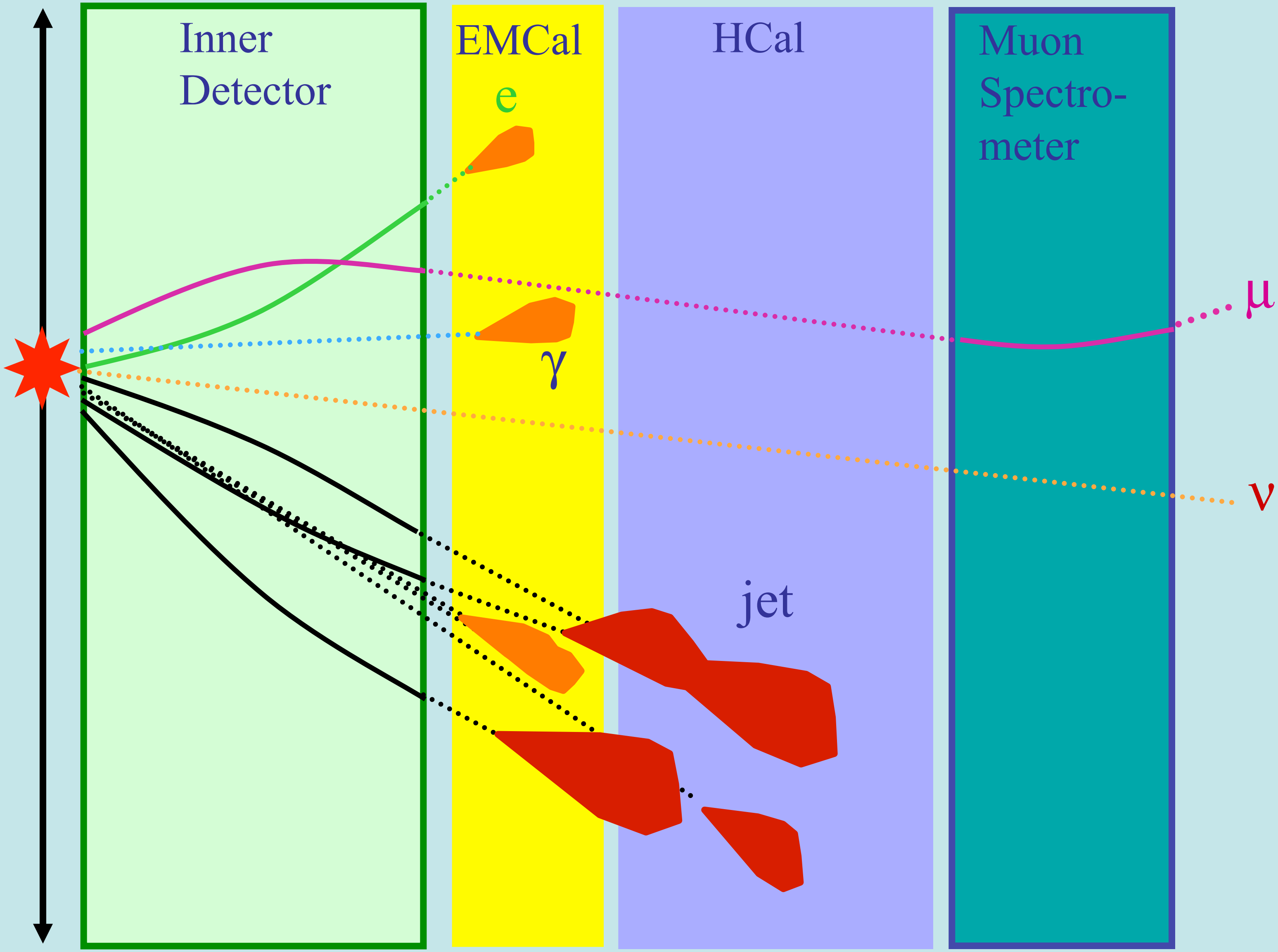
Production of heavy particles $10^{+3…-7}$ Hz
(W, Z, t, Higgs, SUSY,…)



What do we model?

- The primary interaction between the beam particles (quarks, gluons; protons) in the middle of the detector
  - this models the physics we are interested in, and which we need to *reconstruct* from the events observed with the detector
  - Higgs, SUSY, micro-gravity, and known particles are generated here
- The further “life” of the particles emerging from this primary interaction
  - mostly extremely short-lived: flight path less than the diameter of a proton; some travel a few mm or m; some are stable
- The path of the particles through the detector
  - interactions of particles with detector material: energy deposition … slight ionisation in gas or silicon so the track can be “seen”; or full absorption for calorimetric energy measurement
  - bending of charged particles in magnetic fields
- …of course, for that we need to model the detector
  - geometry, material, magnetic fields, …

Particle paths in the detector
(see also the videos – references slide)

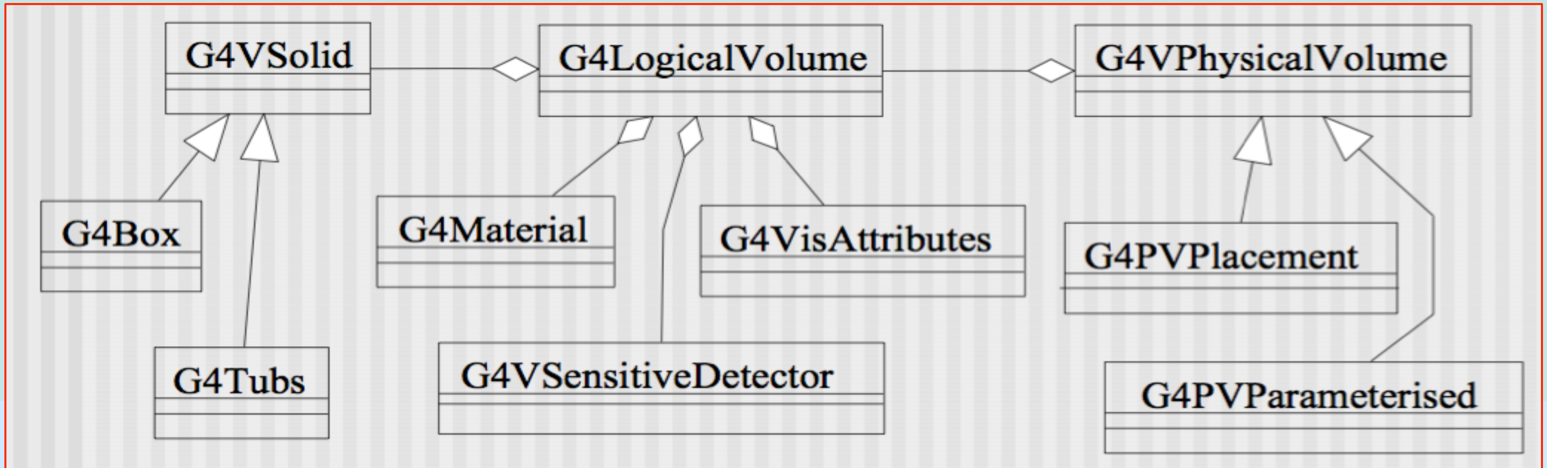


Modeling the detector

- Framework used: Geant4
  - used for most of the modeling involved – not only detector model
  - used throughout HEP experiments, but also in other fields
- Technically:
  - written in C++, providing useful classes from which one can derive for one's own detector description
- Detector model used for many purposes in ATLAS
  - simulation and reconstruction
  - visualisation
  - always starting from the same geometrical model (GeoModel) ensuring identical geometry for all purposes

Creating a detector volume
using three conceptual layers

- Start with its shape and size
  - this is a *Solid*
  - e.g. box 3*5*7 cm, sphere R = 8 cm
- Add properties
  - a *Logical Volume*
  - material, E field, B field
  - make it "sensitive", e.g. for a drift tube, a liquid-argon cell
- Place it in another volume
  - a *Physical Volume*
  - in one place at a time, repeatedly using a function

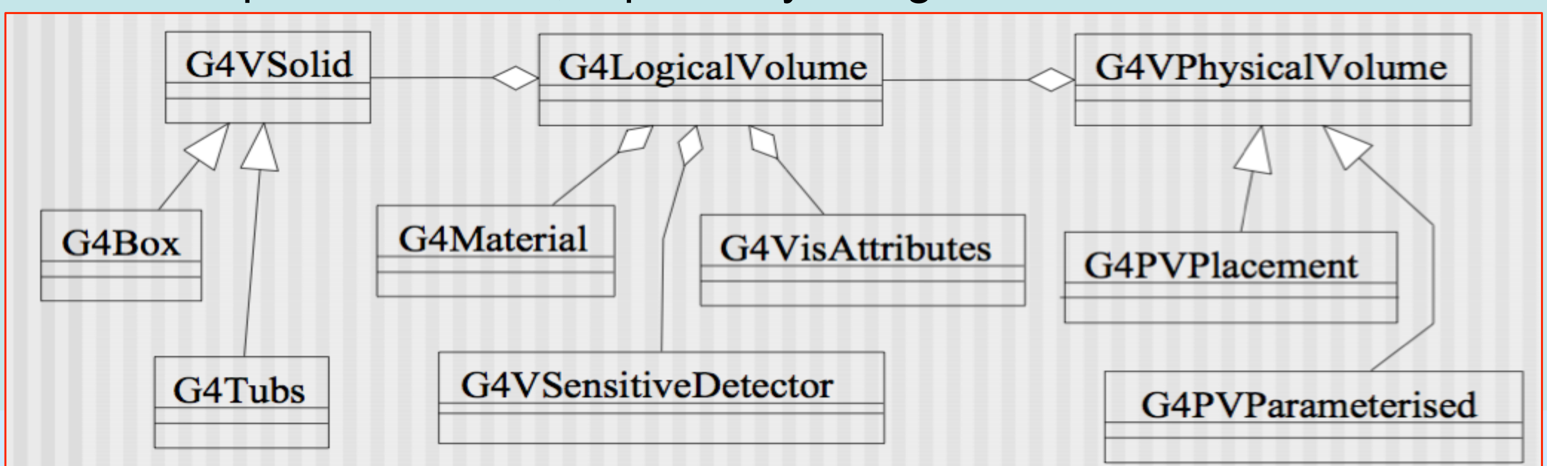


Creating a detector volume
using three conceptual layers

- Start with its shape and size
  - this is a *Solid*
  - e.g. box 3*5*7 cm, sphere R = 8 cm
- Add properties
  - a *Logical*
  - material,
  - make it "s

```
G4double density = 1.390*g/cm3;
G4double a = 39.95*g/mole;
G4Material* lAr =
 new G4Material("liquidArgon",z=18.,a,density);
```

- Place it in another volume
  - a *Physical Volume*
  - in one place at a time, repeatedly using a function



Example detector layouts

- Same logical volumes can be used in different concrete geometries – full ATLAS, partially completed ATLAS, test beam setups:

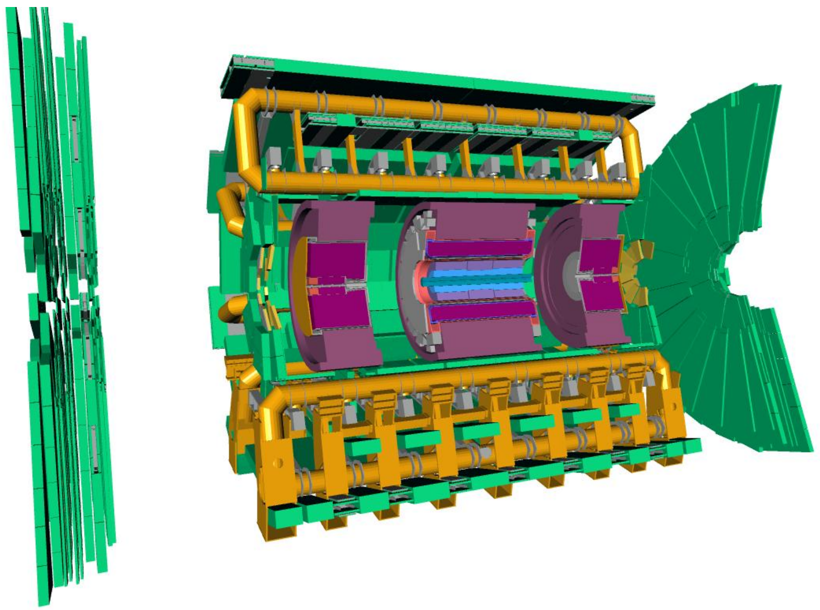


Figure: Cosmic commissioning

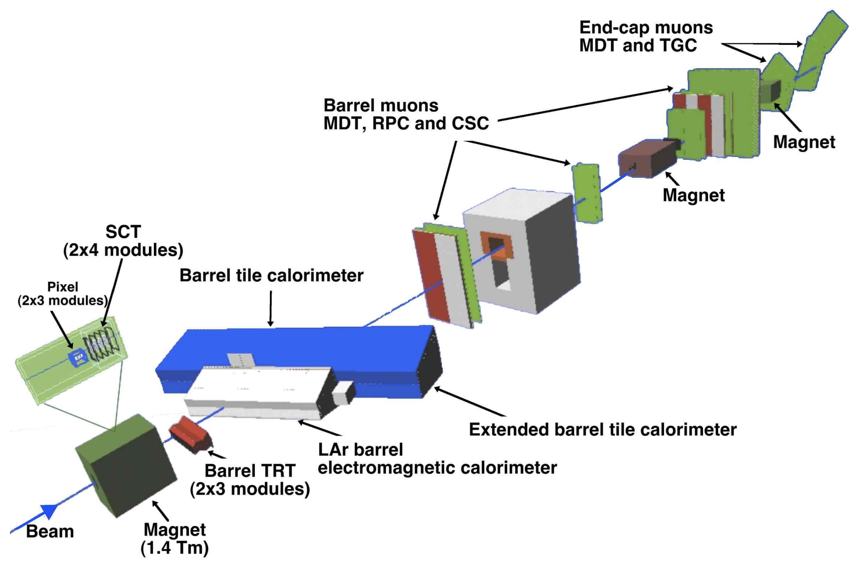
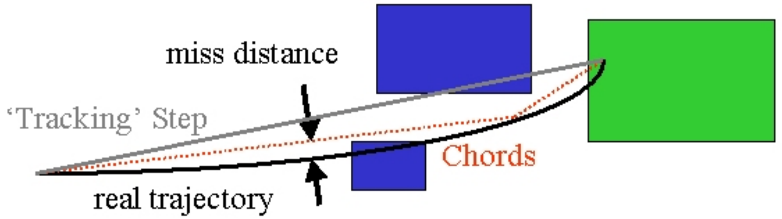


Figure: Combined test beam

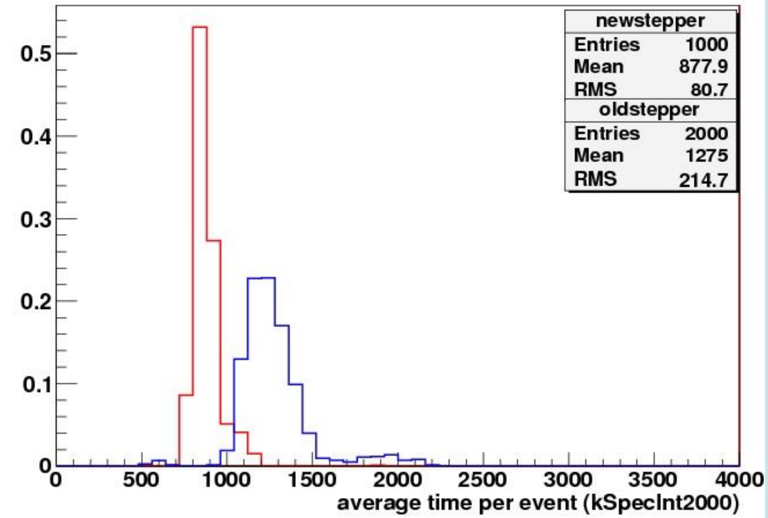
Magnetic field integration

- Numerical integration with Runge-Kutta et al.
- Very CPU time consuming, large fraction of simulation time spent here

Steppers (EM field integration steps)
Lots of simulation time spent on this...
Different steppers:



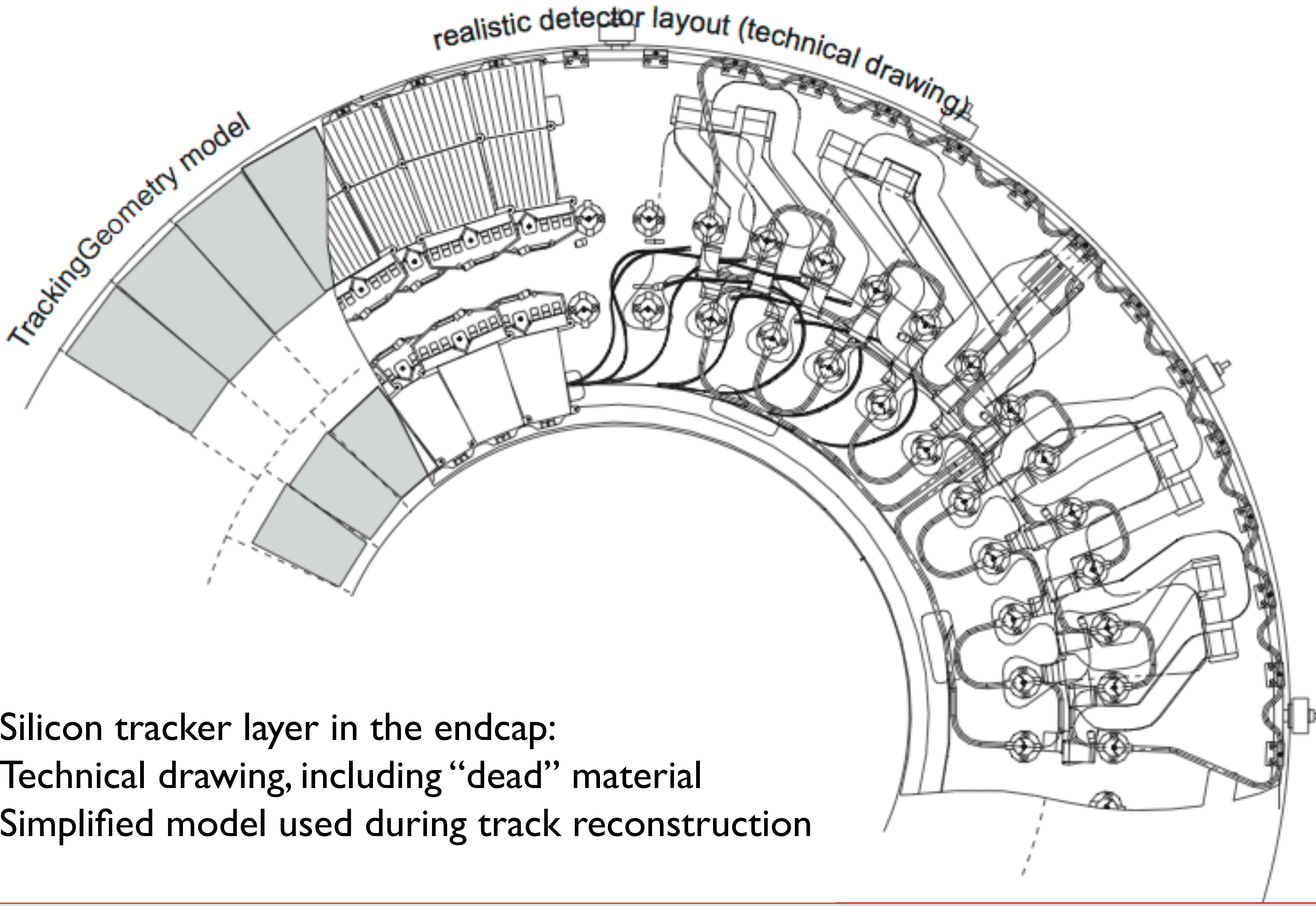
- G4ClassicalRK4: 12 calls per step)
- New stepper: RK-Nystrom integration and intermediate calc. steps cached (20-30% CPU improvement)
  $\Rightarrow$ New stepper G4Nystrom (G4.9.3)
- New G4CachedMagneticField (G4.9.3)
- G4ConstRK4: 1 call per step)



Abstraction of complex geometry for track reconstruction

- Complete detector description used in full *simulation* needs too much CPU when used in track *reconstruction*
  - so generate a simplified version starting from the full geometry
- Dedicated geometry suited for track finding and fitting
  - necessary for the definition of the measurement surfaces where the particle tracks enter/exit a detector element
  - importance of particle interaction with detector material along track (multiple scattering, described stochastically)
- Each detector element has its local reference frame
  - the choice of specific surface representations (e.g. cylinder, plane,…) in general needs intrinsic local co-ordinate systems
  - this is important to establish a coherent track parametrisation w.r.t. the measured co-ordinates given by the detector elements
  - not necessarily cartesian
  - transformations to global ATLAS co-ordinates necessary

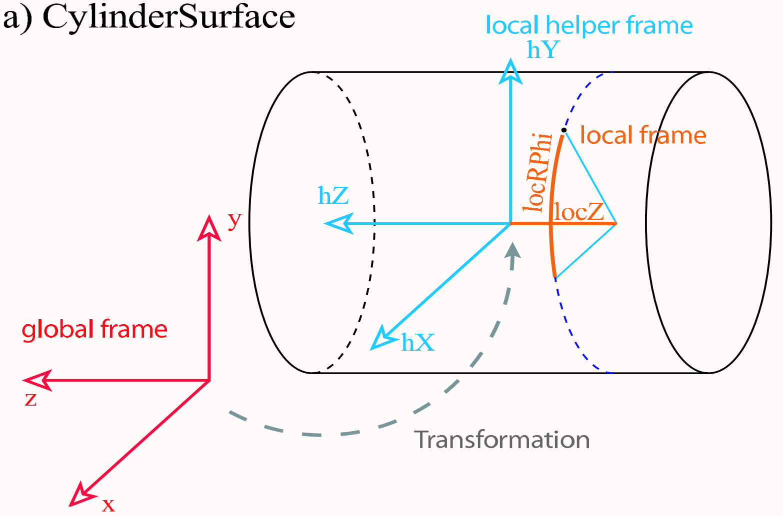
Abstraction of complex geometry for track reconstruction



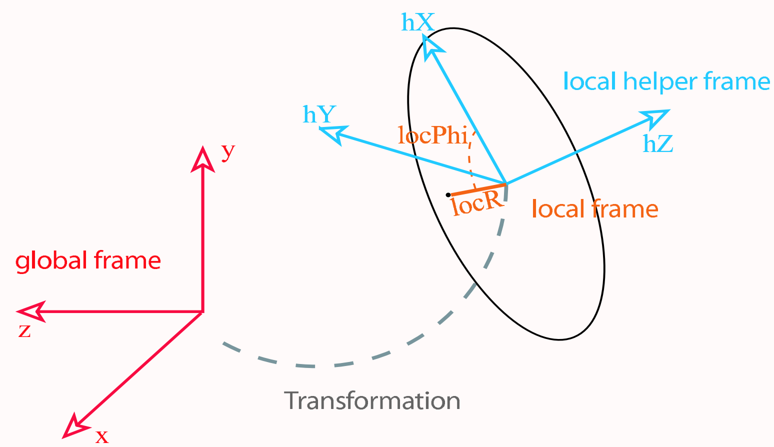
Silicon tracker layer in the endcap:
Technical drawing, including "dead" material
Simplified model used during track reconstruction

Small set of surfaces and boundary shapes used in tracking

a) CylinderSurface

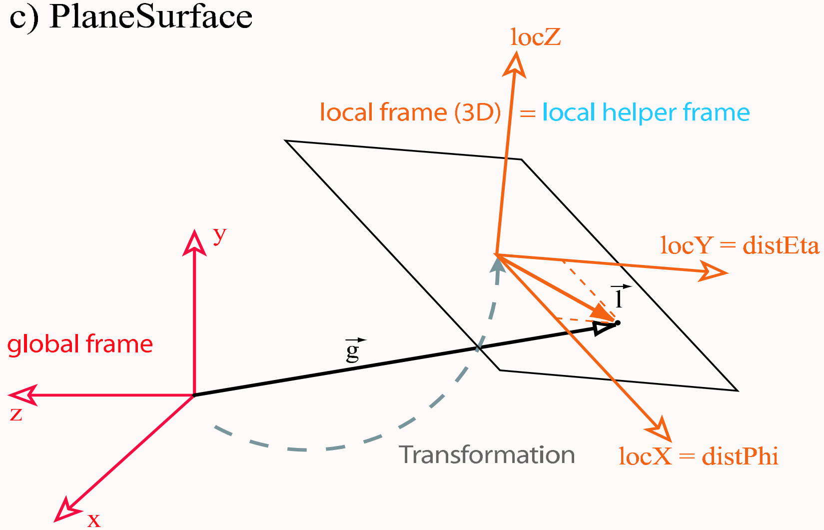


b) DiscSurface

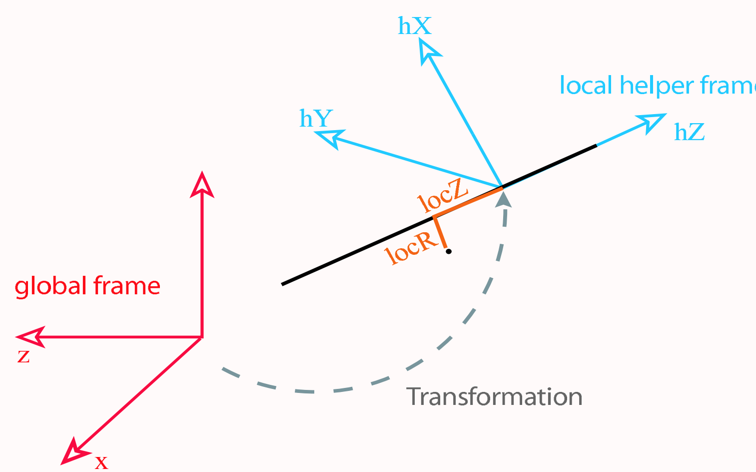


Surface types and their global to local transformations, as used in tracking

c) PlaneSurface

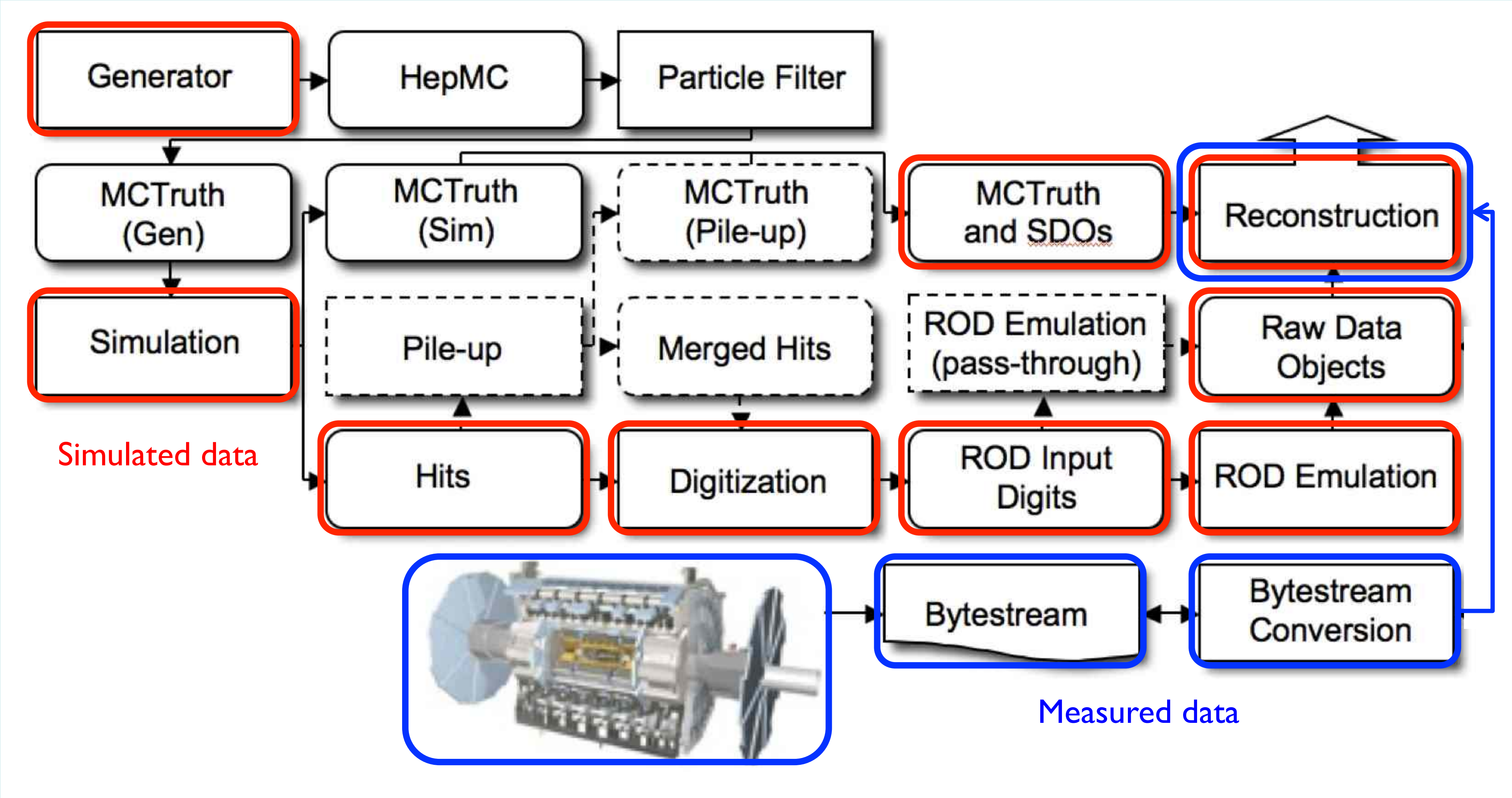


d) StraightLineSurface

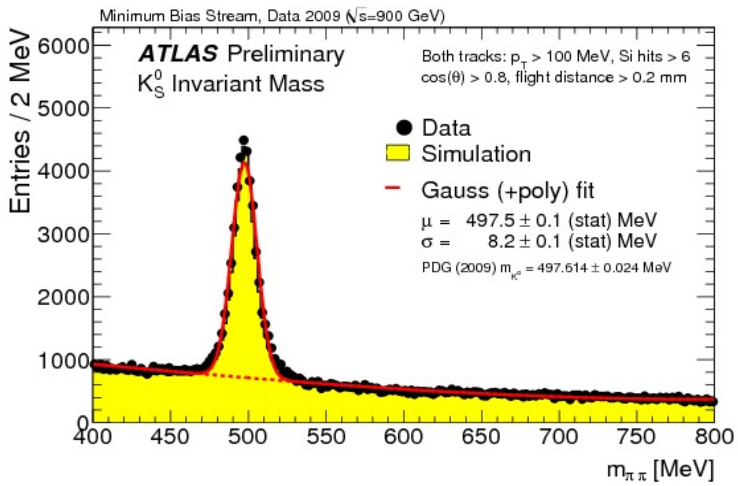


Overall flow of simulation

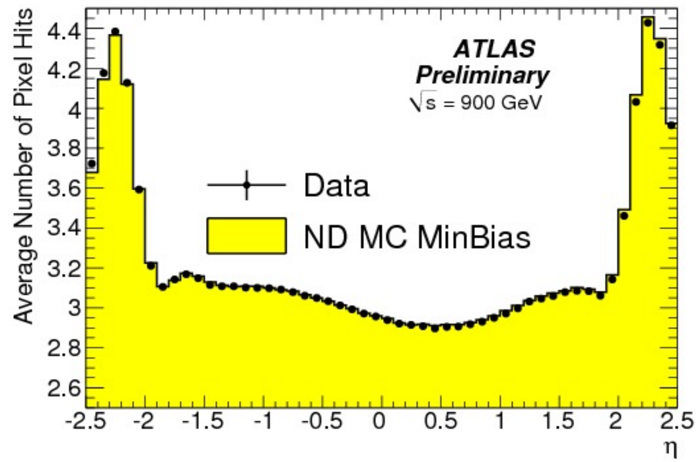
- Reconstruction done in the same way for simulated and measured data



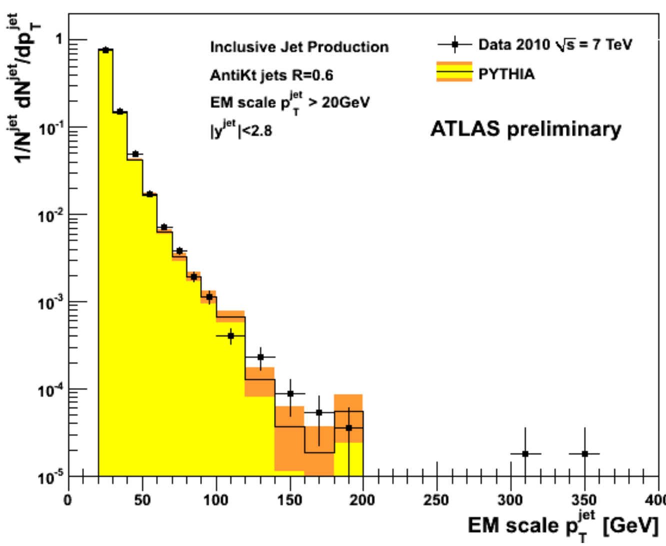
Importance of comparing simulated with measured data for validation of both



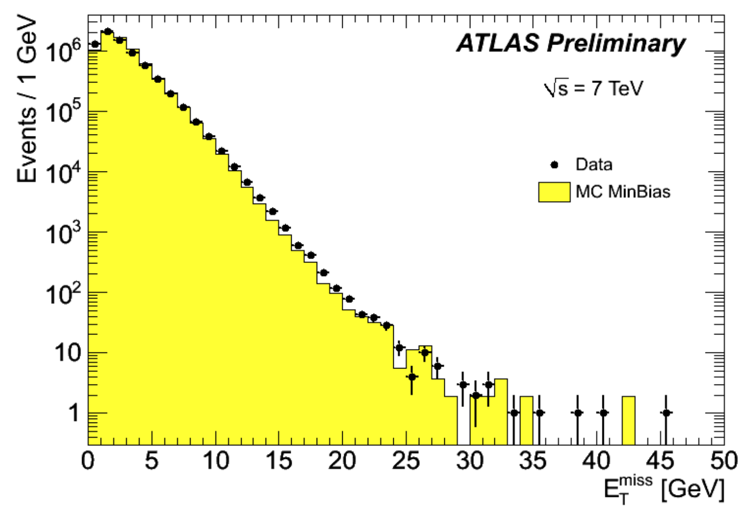
K_S^0 invariant mass



Average no. of pixel hits



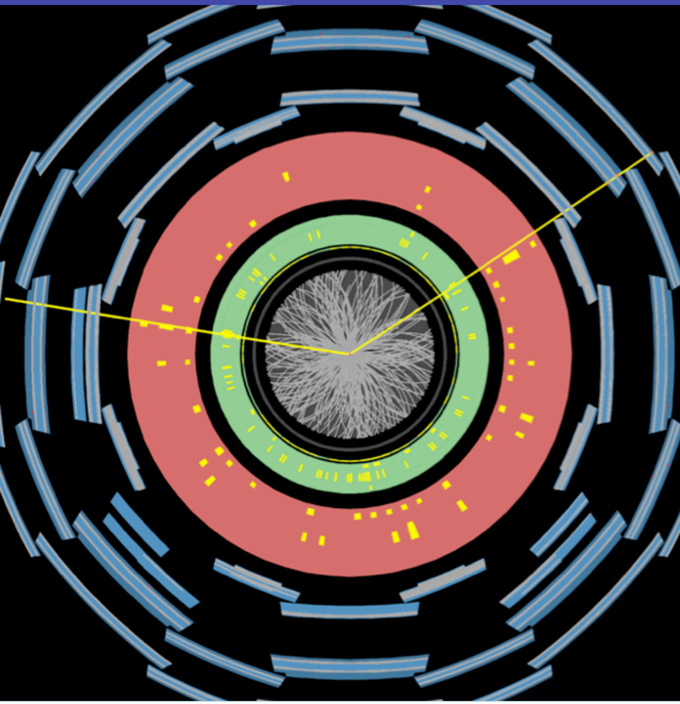
Jet scale

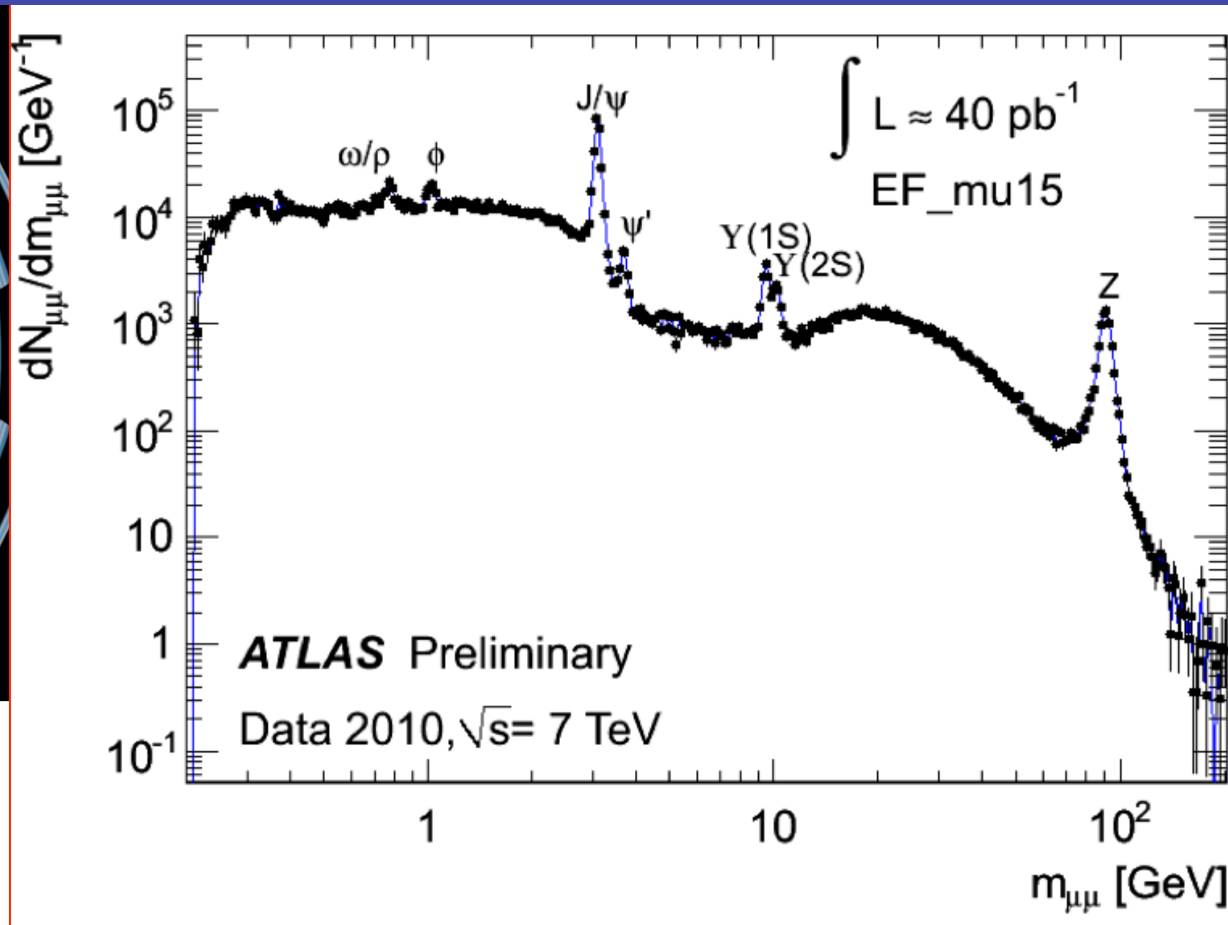


Missing E_T spectrum

Just one example of reconstruction/analysis

calculating properties of unobserved partices from observed tracks





This event contains two muon tracks with 4-momenta p_1, p_2 .

$p = (E, \boldsymbol{p})$, $\boldsymbol{p}$ measured in the toroid, $E^2 = m^2 + \boldsymbol{p}^2$ known because the particle type (muon) is known from the track behaviour in the detector, hence mass m is known.

We calculate the invariant mass of the combination of the two muons – i.e. of an assumed primary particle which decays into the two muons: $m^2_{\mu\mu} = (p_1 + p_2)^2 = (E_1+E_2)^2 - (\boldsymbol{p}_1+\boldsymbol{p}_2)^2$

We obtain the mass spectrum in the right plot, showing peaks at the masses of the known primary particles decaying into two muons – plus continuous background.

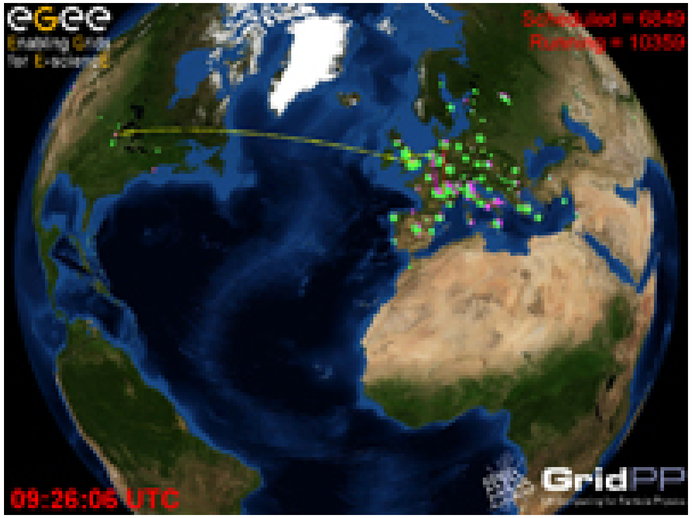
Computing

Computing effort for simulation

Significant time per event for G4 ⇒ Large scale Production

Signal	CPU time (kSI2k.s)
Jets	1139.093
$H \rightarrow ll$	1241.541
MinBias	478.994
SUSY	1923.755
$Z \rightarrow ee$	1204.342
$Z \rightarrow \mu\mu$	960.114
$Z \rightarrow \tau\tau$	1036.051

CPU time per event simulated: ~ 20 min



- Grid Tasks (e.g. 500k $t\bar{t}$)
- Split into jobs (typically 25/50 events) to fit within 48 hrs
- Output registered Distributed Data Management (DDM)
- Extensive physics validation of samples before use

Eight million events can be produced daily

Failure rate is less than 10^{-6}

i.e. need O(100000) CPU cores

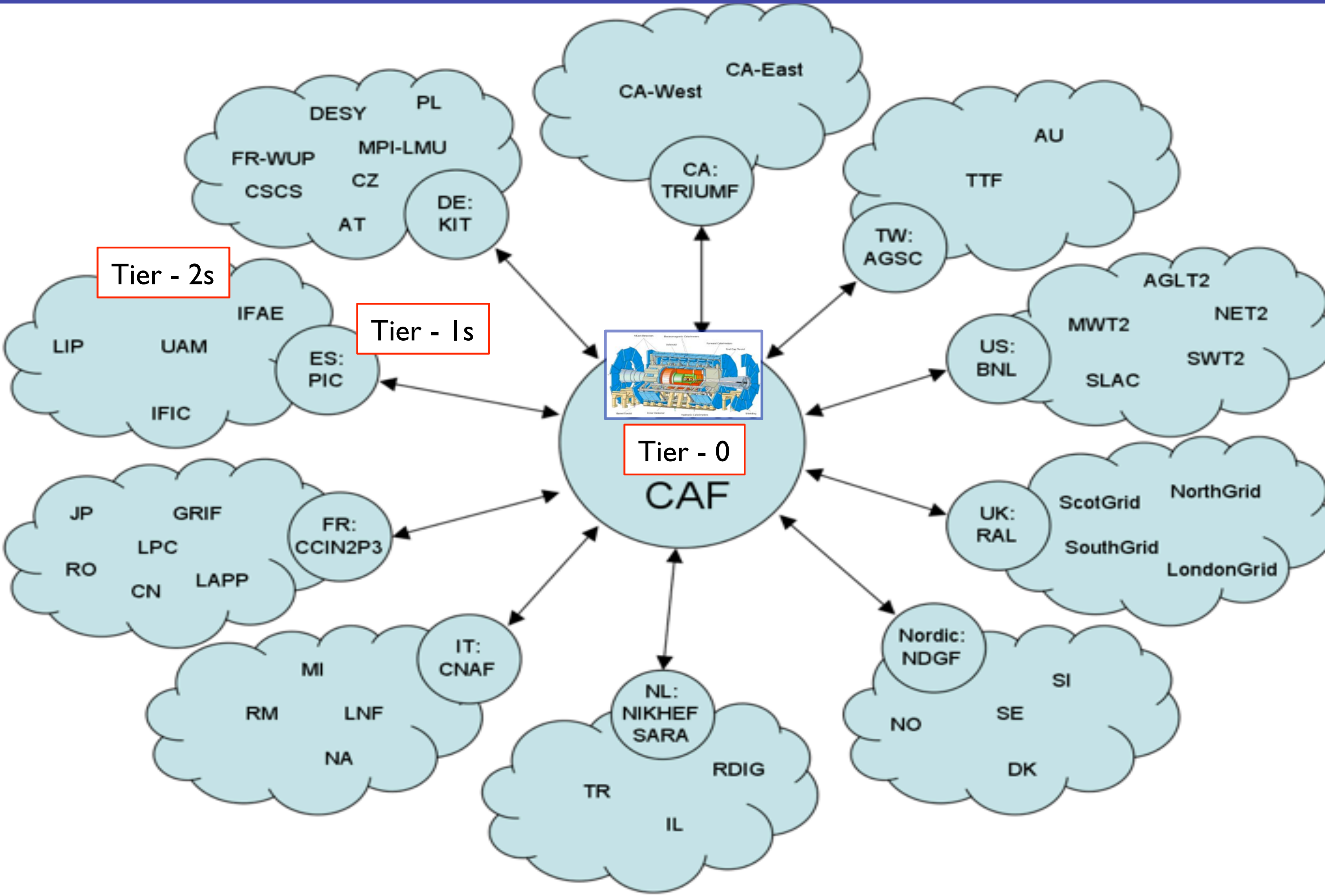
Tasks performed on ATLAS computing infrastructure

- Simulation, as we have seen
  - very CPU intensive full simulation. O(100000) CPU cores was the ballpark – if we were simulating all the time
- Reconstruction and analysis
  - reconstruction is done in a co-ordinated, central effort – a few people do it for the entire collaboration, mostly batch processing. Need 10-50s CPU per event - O(10000) CPU cores all the time
  - analysis is usually done by small groups or individual physicists, much interactive processing, mostly graphical output, iterated many times
- Why do we need so many *measured* events?
  - we have seen a few event displays. In principle a single event can be very instructive
  - but in general an analysis in particle physics is a statistical analysis
  - esp. in proton-proton (or heavy ion) collisions, many more background events are produced than “interesting” ones (like Higgs), so need much statistics to separate the signal from the background
  - we record ~2 billion events per year, occupying ~3 petabytes storage

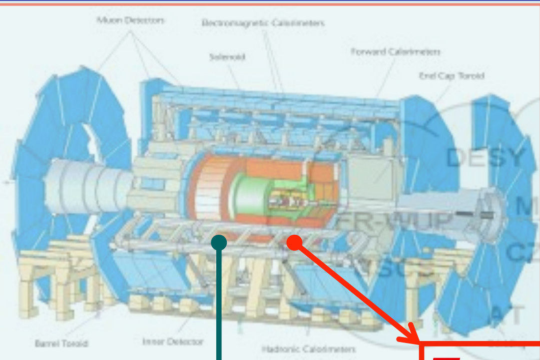
How can we use so many CPUs in parallel?

- Trivial in principle. Event data are independent so they can be processed easily in parallel
  - only almost true: several/many events share common metadata like running conditions
- CPU boxes are coming with more and more cores, esp. true for the graphics processing units (GPU) or CPU-GPU integrated architectures
  - from now 8 cores per box soon to >100 cores per box
  - the answer is to use more fine-grain parallelism in addition to event parallelism – we are actively working on this
  - the linear algebra in the inner loops of track reconstruction are especially suited for more parallelism
  - …also the neural-network algorithms which disentangle multiple hits in the pixel detector
  - useful tools arriving – thread and array building blocks, CILK; all C++

ATLAS and the worldwide Grid computing infrastructure



Information flow from detector to publication



Simulated data (files)

- Generated/reconstructed on Tiers, to Tiers

Event data (files, TAG also on DB)

- RAW from detector reconstructed to ESD, AOD, TAG, NTUP at Tier0, to Tiers; re-reconstructed at Tier1s, to Tiers

Metadata (DB, files) include:

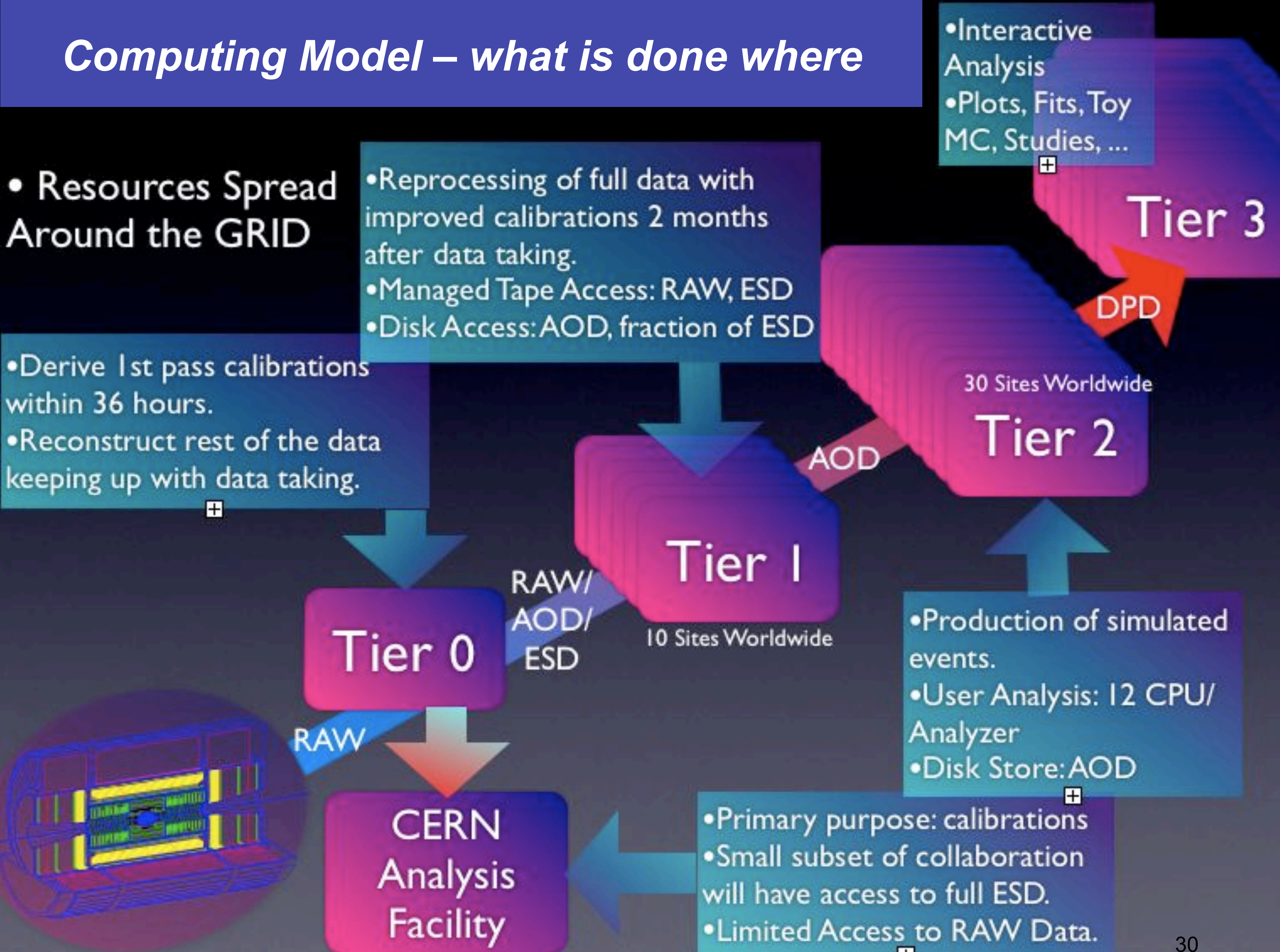
- trigger setup (Trigger DB), dataset data
- running conditions / data quality (COOL DB)
- alignment / calibration, luminosity, backgrounds, ...

- software (SVN)
- software tags used in processing
- less-structured input (logbooks, doc, knowledge)

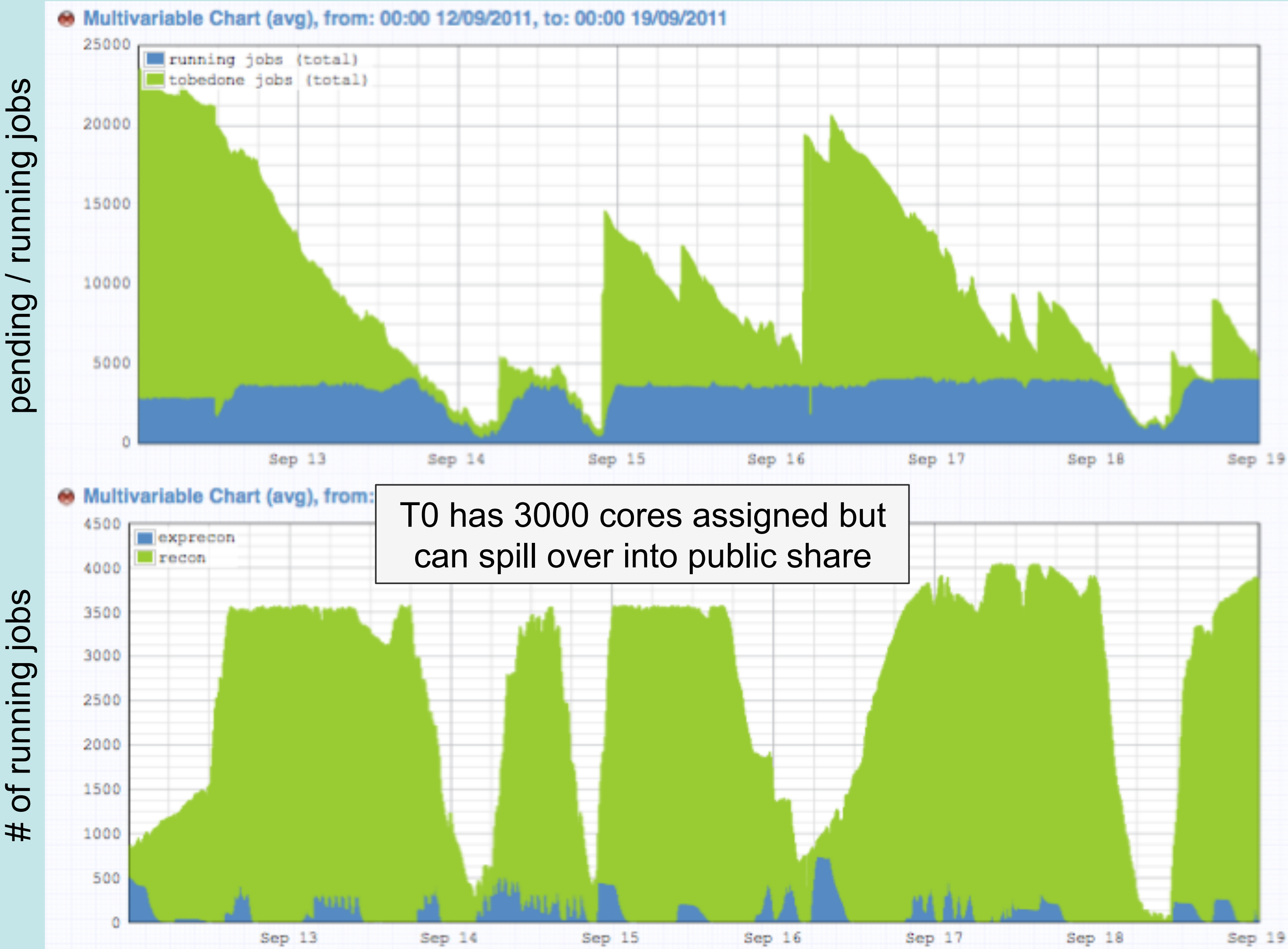
Physics analyses

Publications

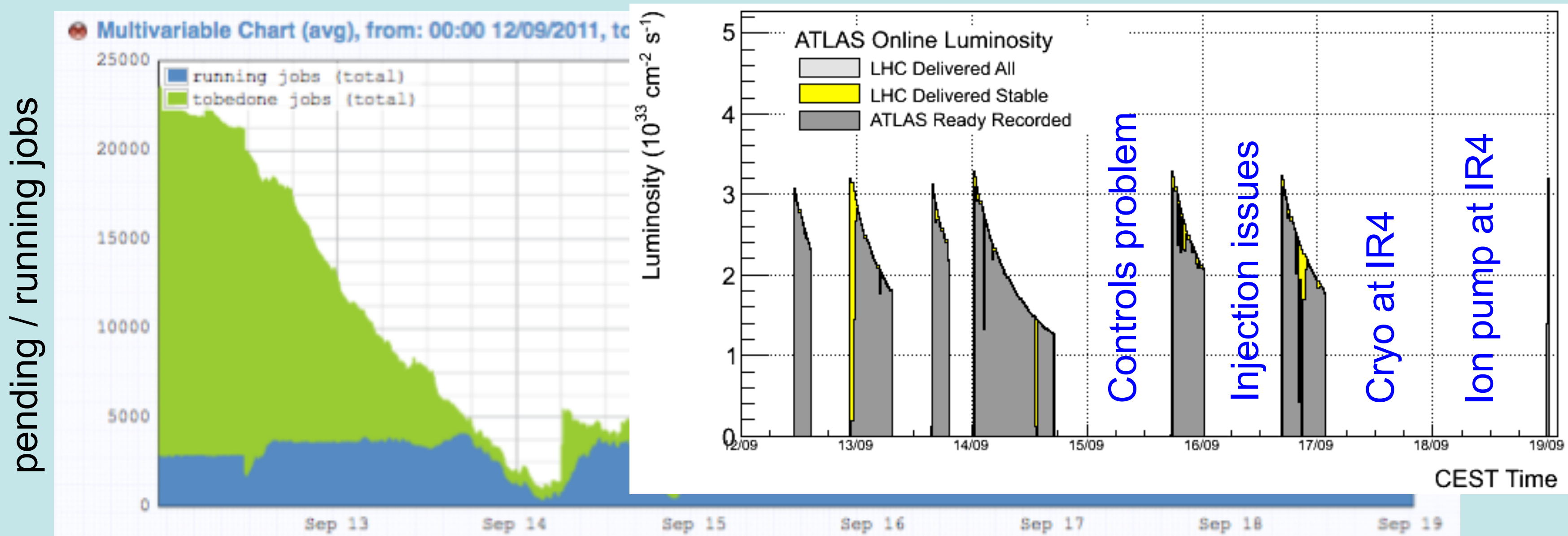
Computing Model – what is done where

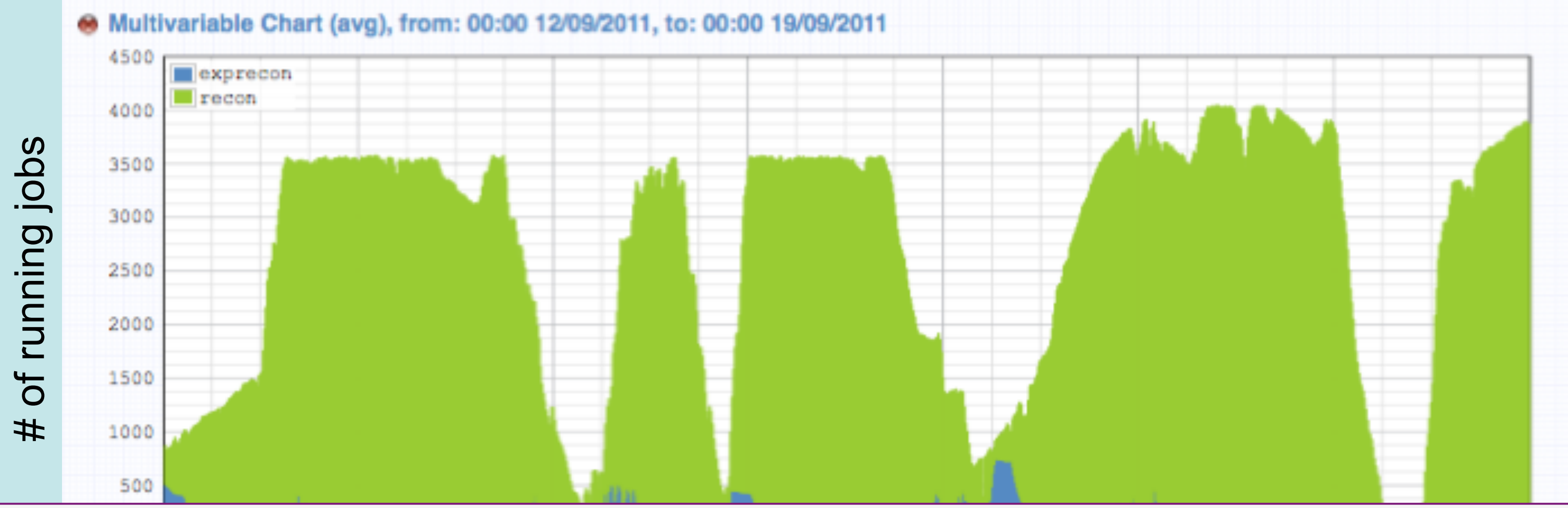


Tier-0 does immediate processing of subset of events coming from Point1, then full processing after calibration



Tier-0 processing follows the time structure of LHC fills with stable beams (note the importance of monitoring)

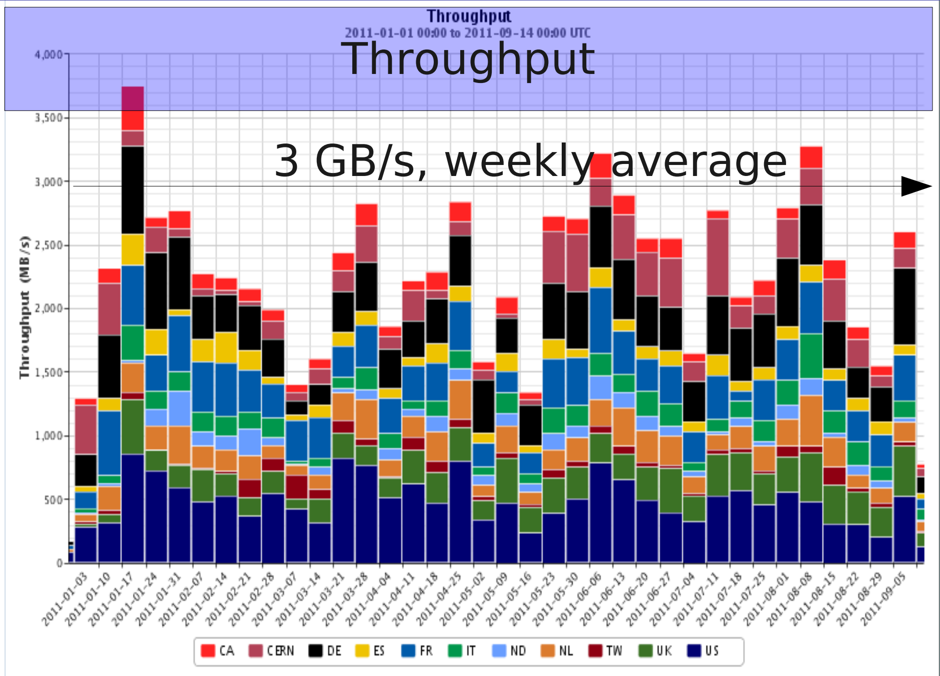


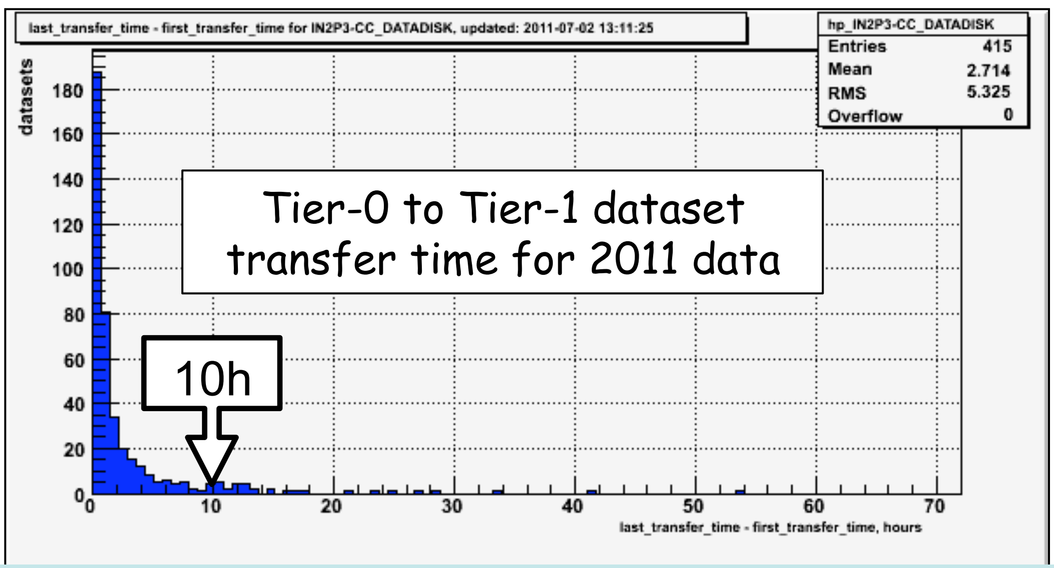


Data volume handled by Tier-0 in 2011 so far:
~2.5 PB RAW recorded, ~5 PB data distributed to Grid

Distributed Computing on the Grid: data transfers

- Data distribution
  - pre-placement
  - dynamic placement
  - user requests
- Peak throughput 10 GB/s
- Success rate 93% in 2011

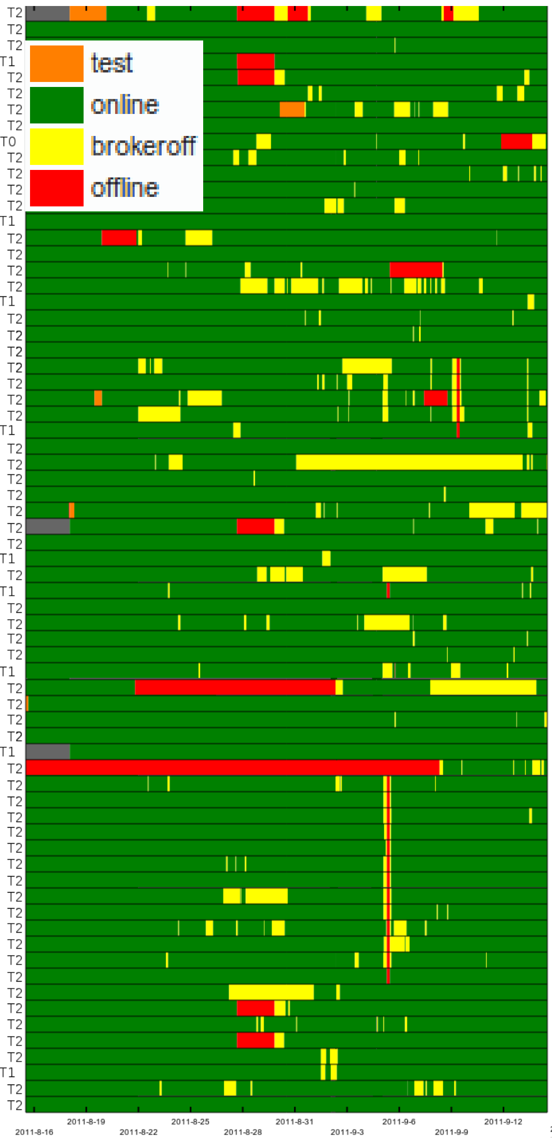




- Data are available for analysis in "almost-real" time. Example:
  - data11_7TeV AOD distribution (to one specific Tier-1 but they are all similar):
  - on average 2.7 hours to complete the dataset

Distributed Computing: data processing
(note importance of monitoring…)

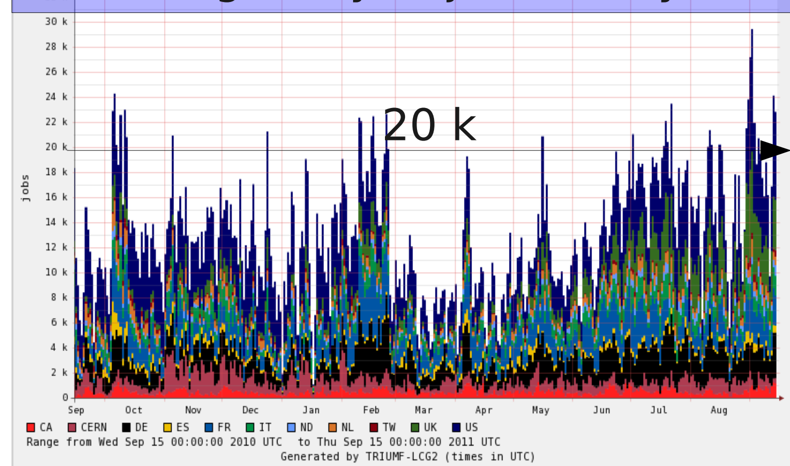
Site Status
Analysis activities



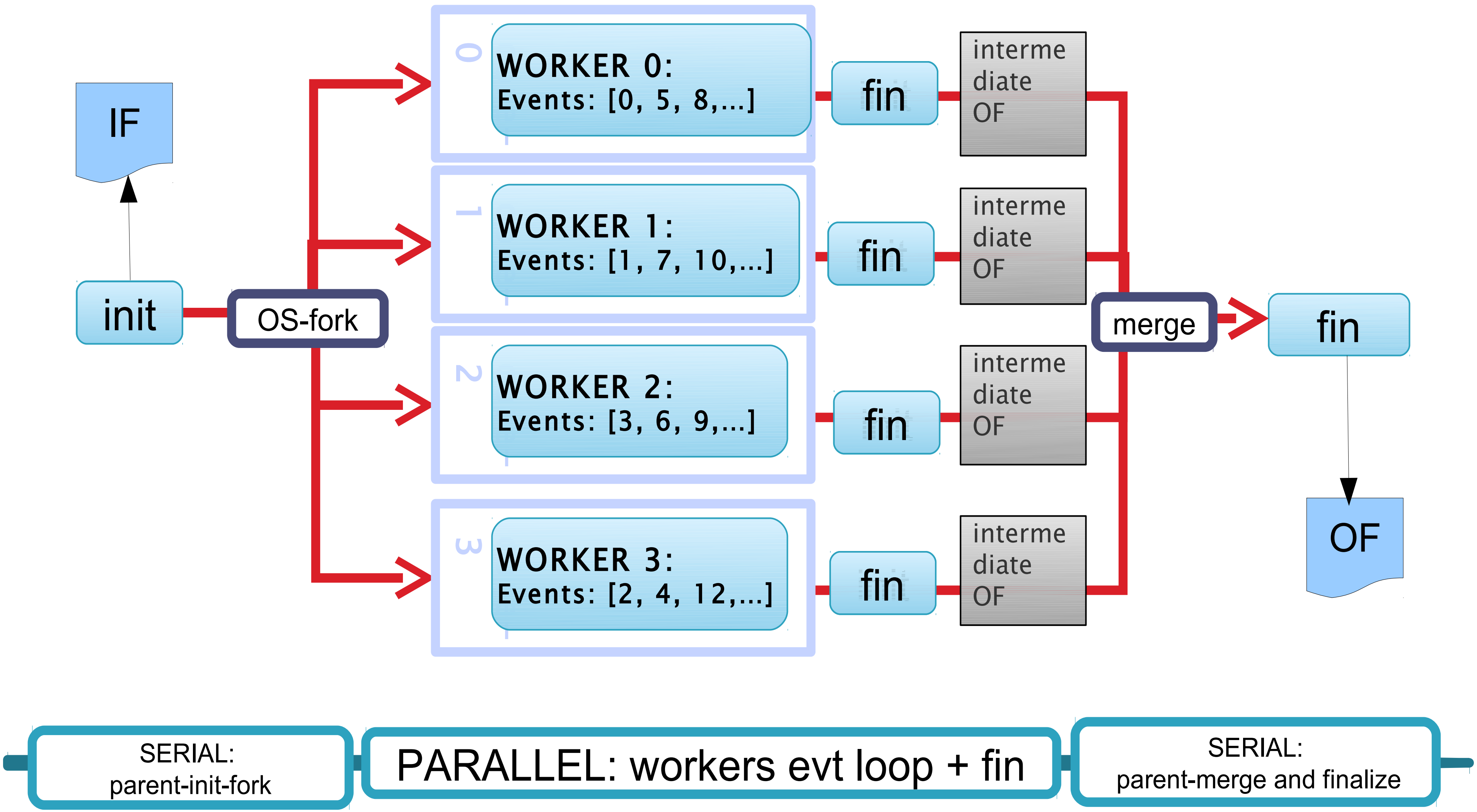
Site Status
Production activities

- Ca 80k jobs running simultaneously
- 12 % of CPU time spent on analysis
- Automatic job resubmission

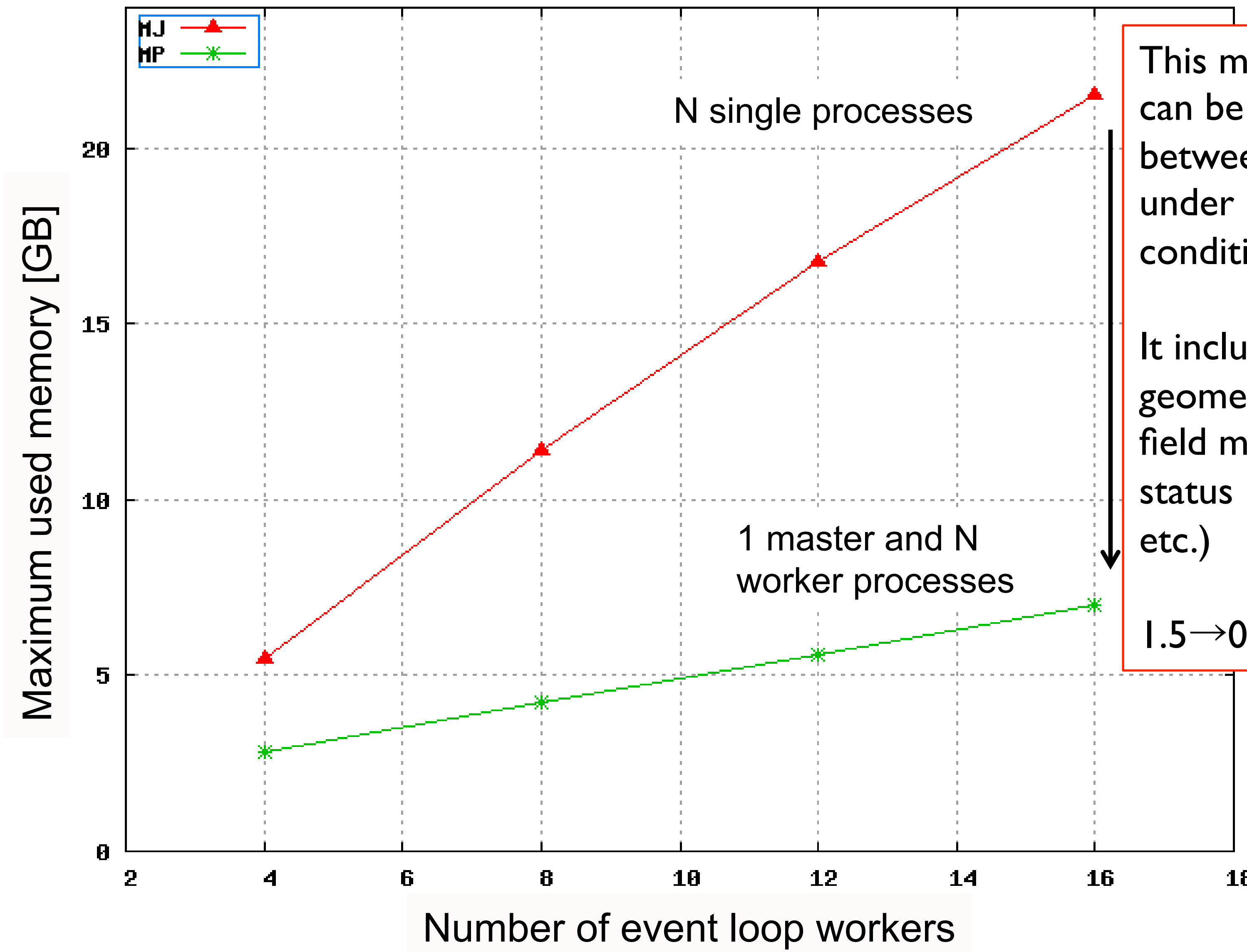
Running analysis jobs, last year



Event-level parallelism



Memory used (8-core machine with hyperthreading, 24GB)



This much memory can be shared between events taken under similar running conditions.

It includes the geometry, magnetic field maps, detector status (high voltages, etc.)

1.5→0.5 GB/process

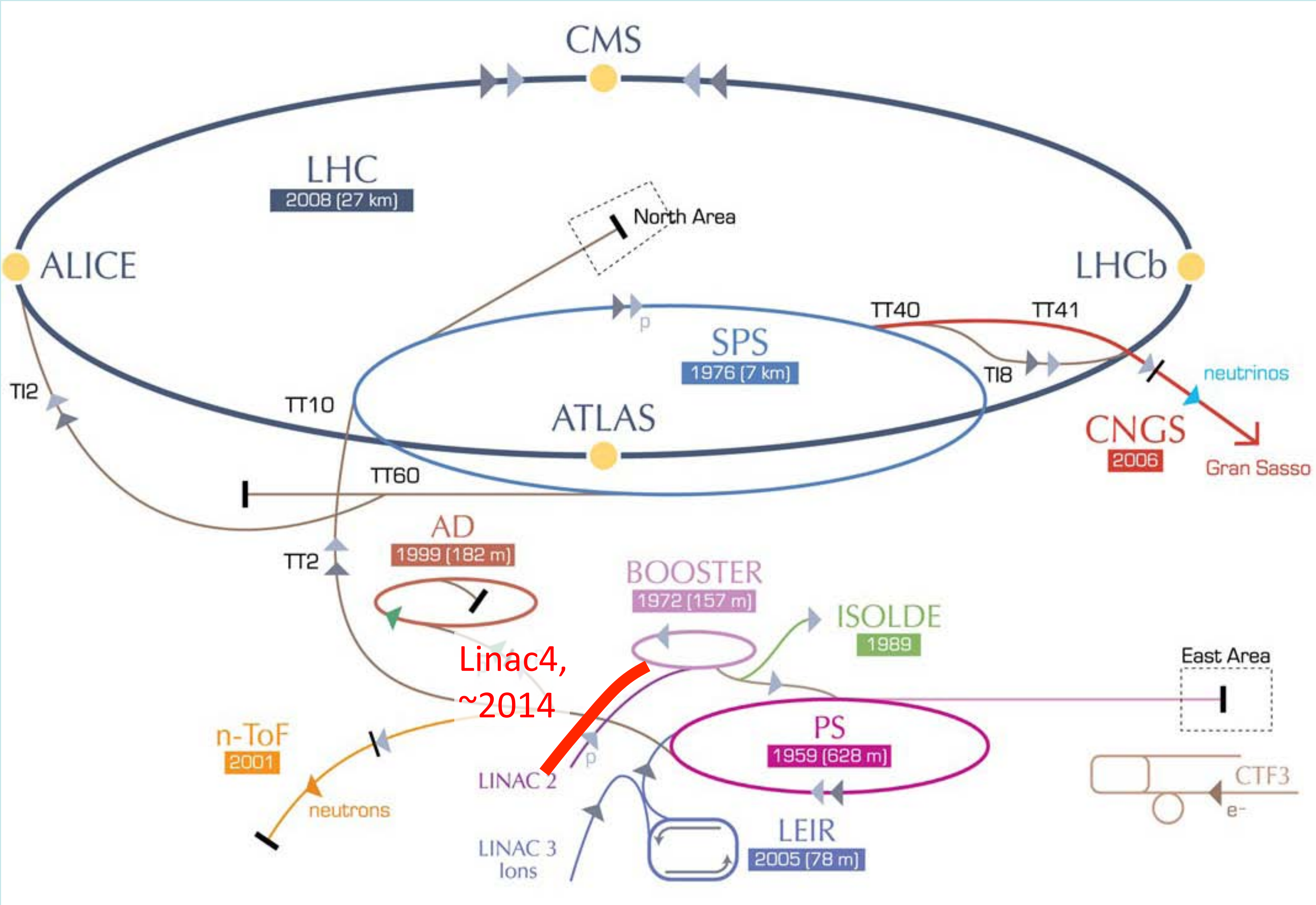
Some references

- BCS 50 Years, Leon Cooper and Dmitri Feldman (ed.), World Scientific, 2011
- ATLAS videos, http://atlas.ch/detector.html, http://atlas.ch/detector-overview/ (sorry for the music – correct otherwise – meant more for the general public)
- Geant4 Course, http://www.ge.infn.it/geant4/events/nss2003/geant4course.html
- ATLAS Simulation paper, arXiv:1005.4568
- ATLAS Tracking Geometry Description, ATL-SOFT-PUB-2007-004

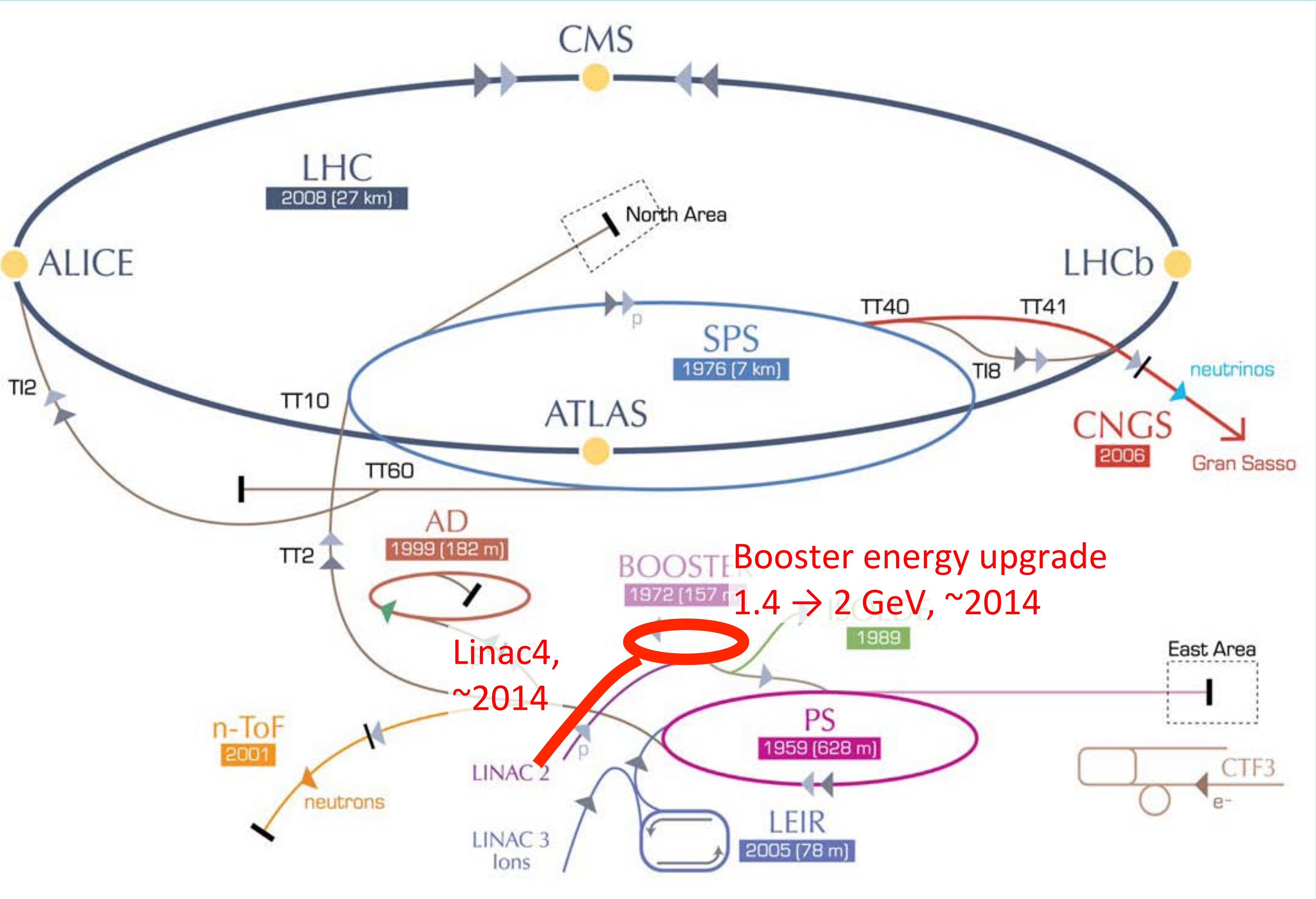
Thank you for listening!

Extra slides

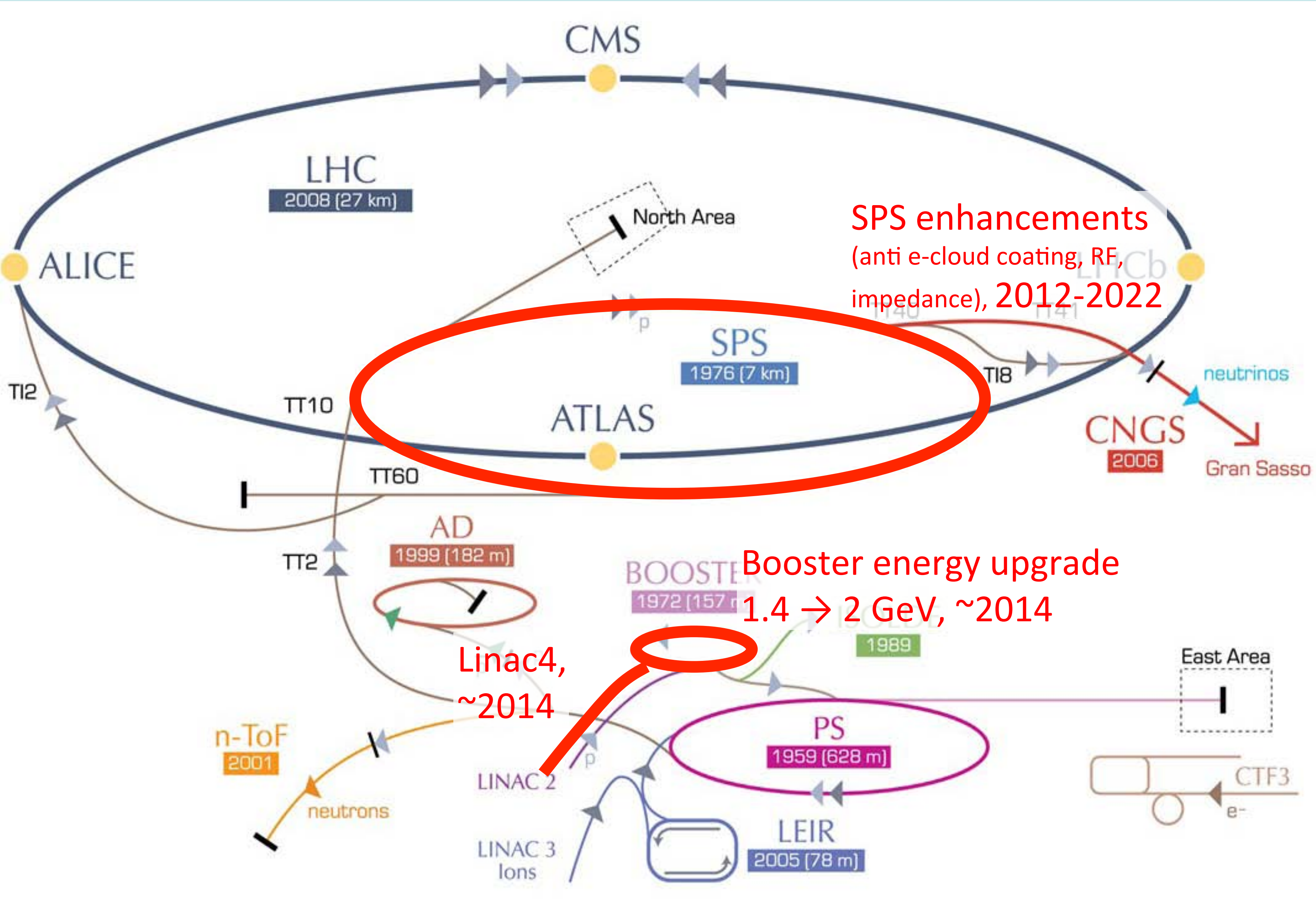
HL-LHC – accelerator modifications



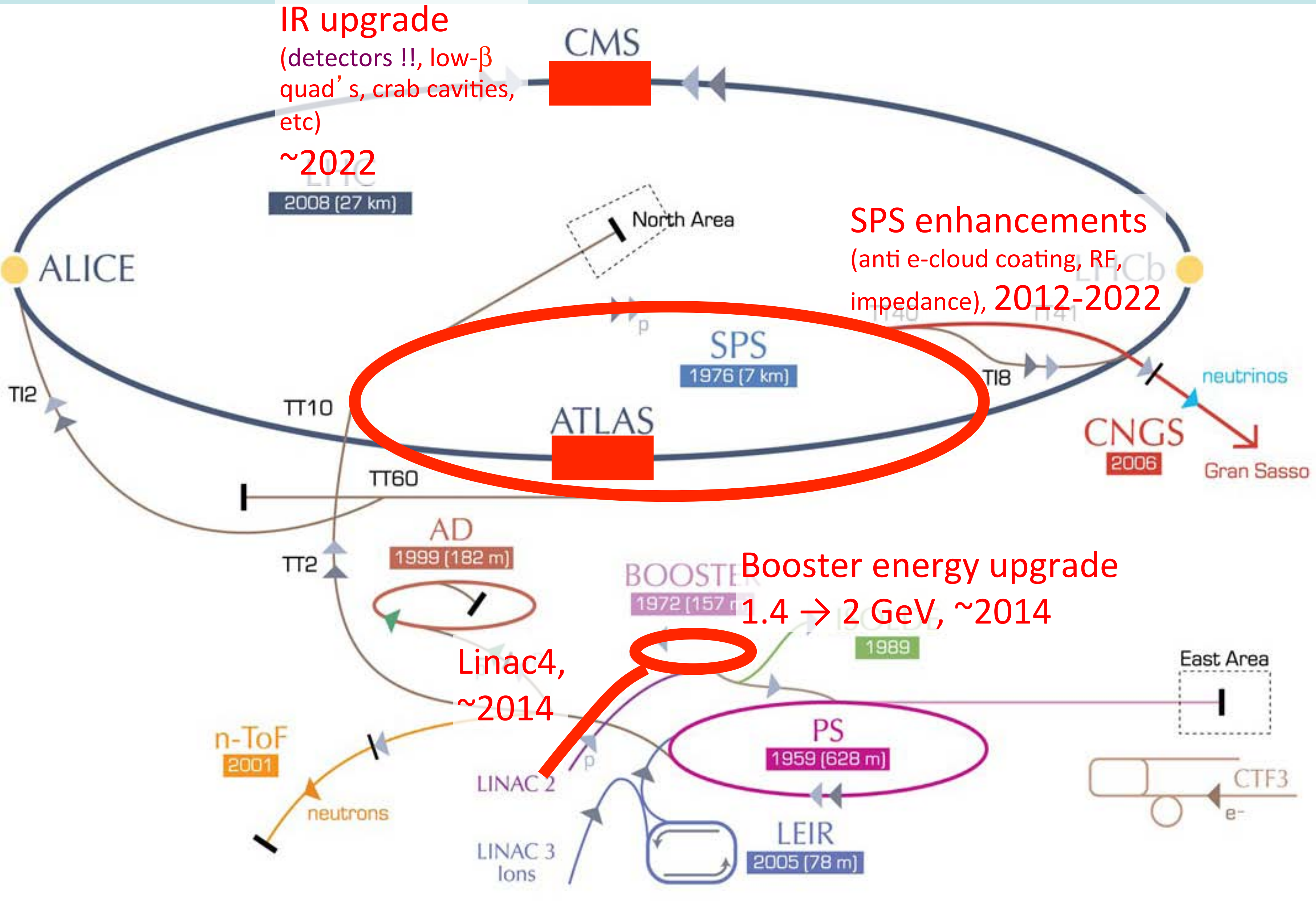
HL-LHC – accelerator modifications



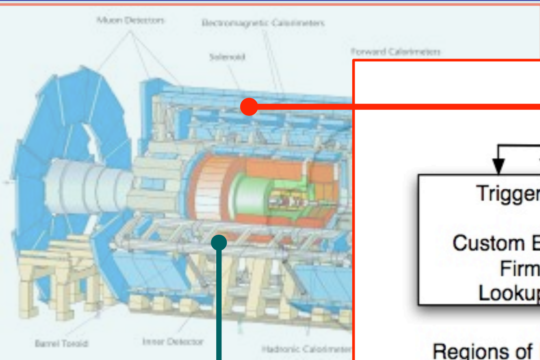
HL-LHC – accelerator modifications

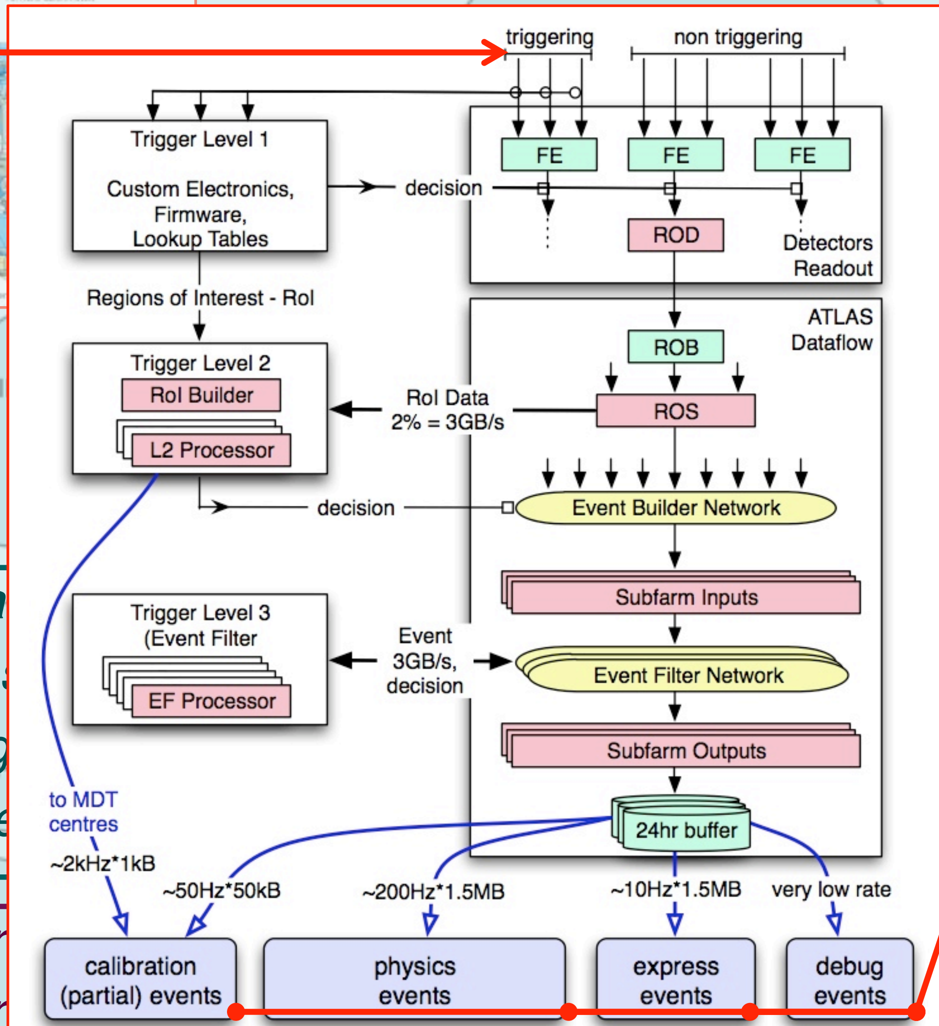


HL-LHC – accelerator modifications



Information flow – starting at the detector (Point1)





s)
tructed on Tiers, to Tiers

ESD, AOD, TAG,
cted at Tier1s, to Tiers

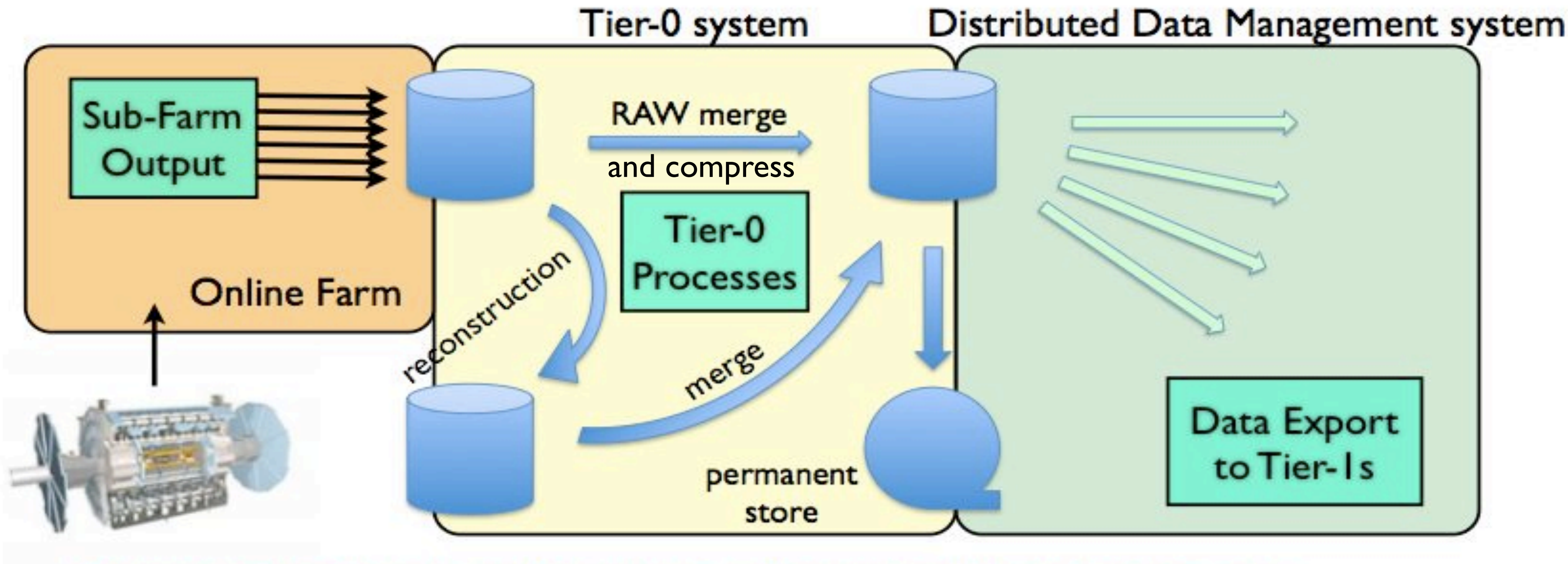
Metadata
- *trigger*
- *running*
- *alignme*

- *softwar*
- *softwar*
- *less-structured input (logbooks, doc, knowledge)*

Physics analyses

Publications

Data flow through the Tier-0 at CERN



Accepting data from the online system and ensuring it is archived to tape

- Merging small files to adequate size for tape archiving

Processing RAW data (event reconstruction) and archiving the products to tape

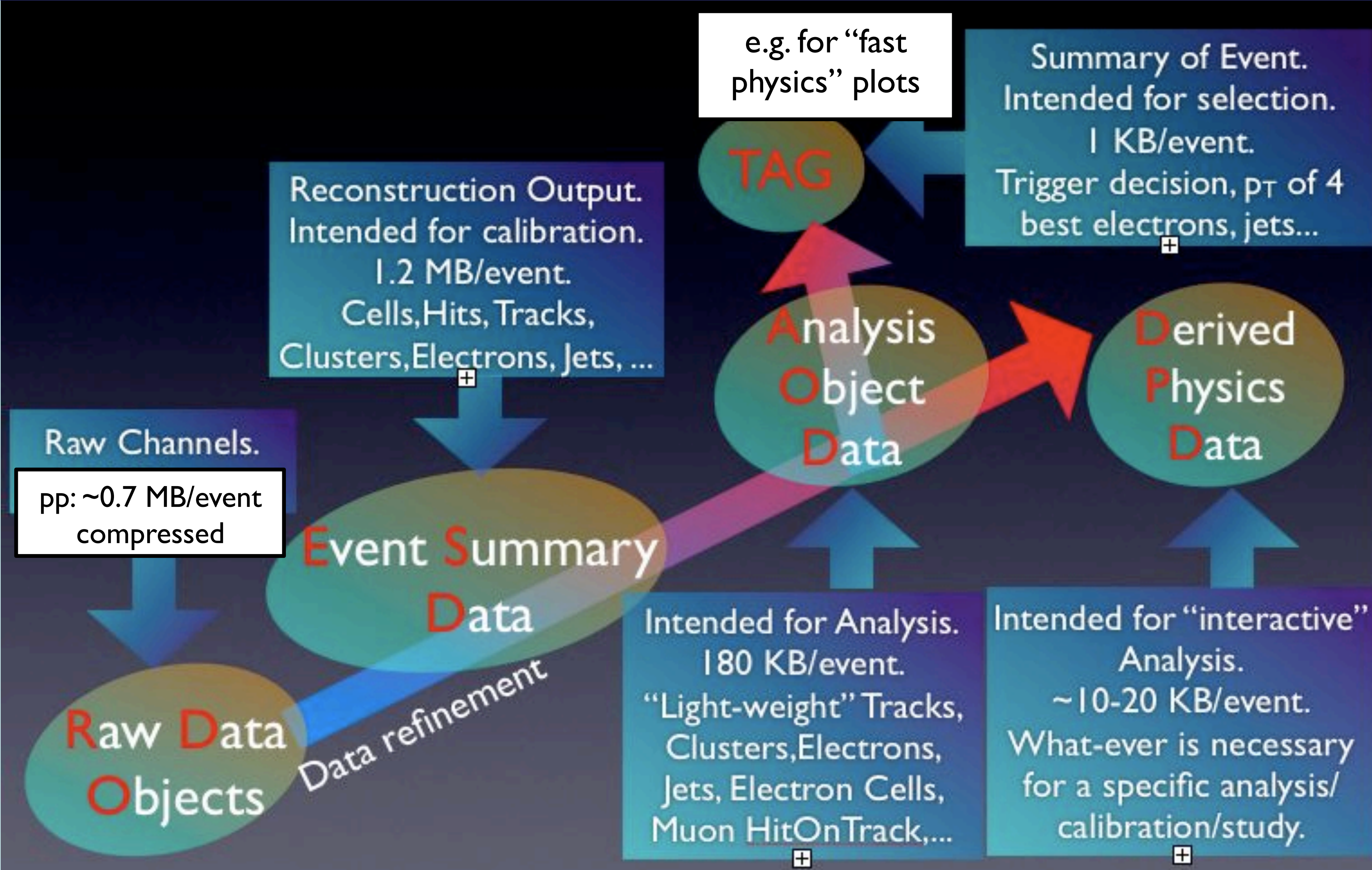
- Express stream for prompt calibration and alignment
- First-pass processing of all streams after 36h with calibration and alignment

Registering data to the ATLAS Distributed Data Management system

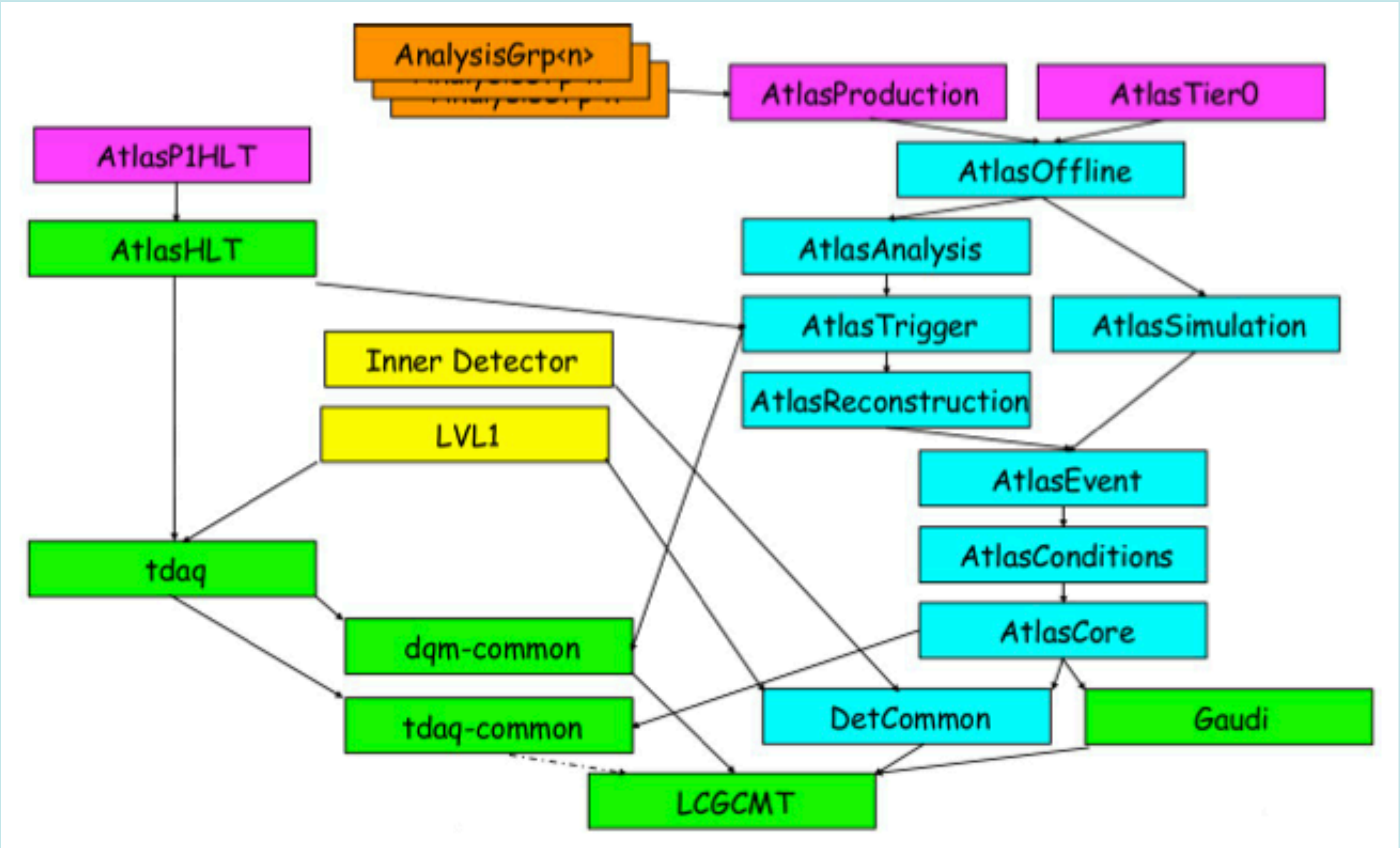
- Export data to Tier-1 and calibration Tier-2s, as well as CAF

Maximum overall I/O: 6GB/s -- including internal accesses within Tier-0

Event Data Model – the various RAW and derived data types



ATLAS software "projects"



ATLAS software: examples of #lines of code

Language	Files	Comment	Code
C++	930	24,000	120,000
FORTRAN	270	15,000	42,000
C/C++ Header	1,100	13,000	34,000
Python	430	16,000	27,000
HTML	62	130	15,000
Bourne Shell	390	1,000	7,300
C Shell	380	210	3,800
XML	52	1,200	3,400
Sum	3,600	70,000	250,000

Code in the Simulation project

Project	C/C++ Code	C/C++ Headers	PYTHON Code	Total Code
Core	390,000	43,000	240,000	860,000
Event	200,000	110,000	16,000	350,000
Conditions	280,000	90,000	21,000	620,000
Detector	38,000	6,100	8,400	140,000
Sum	910,000	250,000	280,000	2,000,000

Code in projects used by Simulation

Overall ATLAS Athena software:
4 M lines C++, 1.4 M Python, 100 k Fortran, 100 k Java, …

ATLAS software in numbers

- ATLAS offline software is called “Athena”
  – Algorithms are used also in High-Level Trigger, under a different framework
- 2000 packages
  – sorted in several “projects” for unidirectional dependency
- 4 Million lines C++, 1.4M Python, 100k Fortran, 100k Java, ...
- 1000 developers have committed software to the offline repository in the last 3 years
- 300 developers have requested 4000 package changes in first half 2011 (25 per day)
  – It never stops: data taking, reprocessing, conferences
- 3000 users have a Grid certificate in atlas VO (able to submit job, retrieve data)

From bunchcrossings to physics analyses