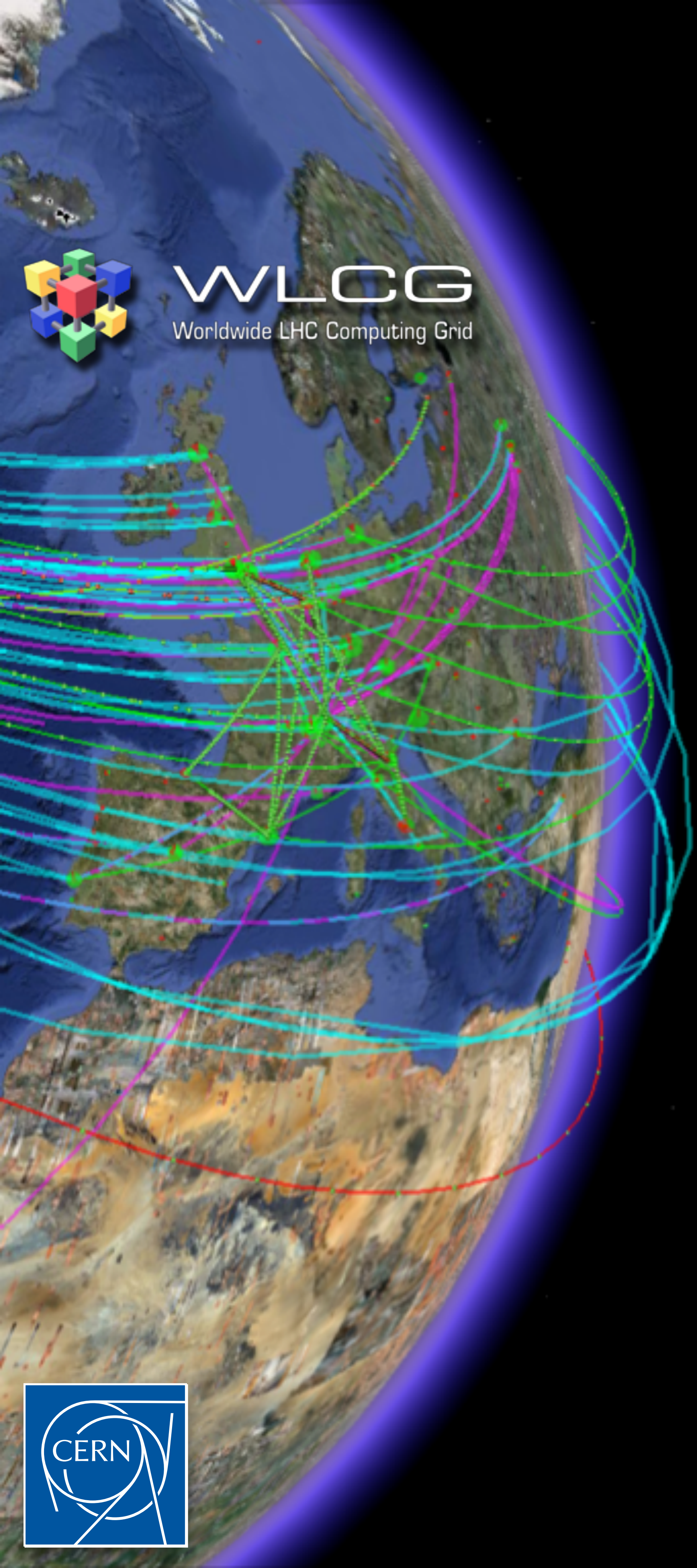


KEK-JAPAN

Markus Elsing

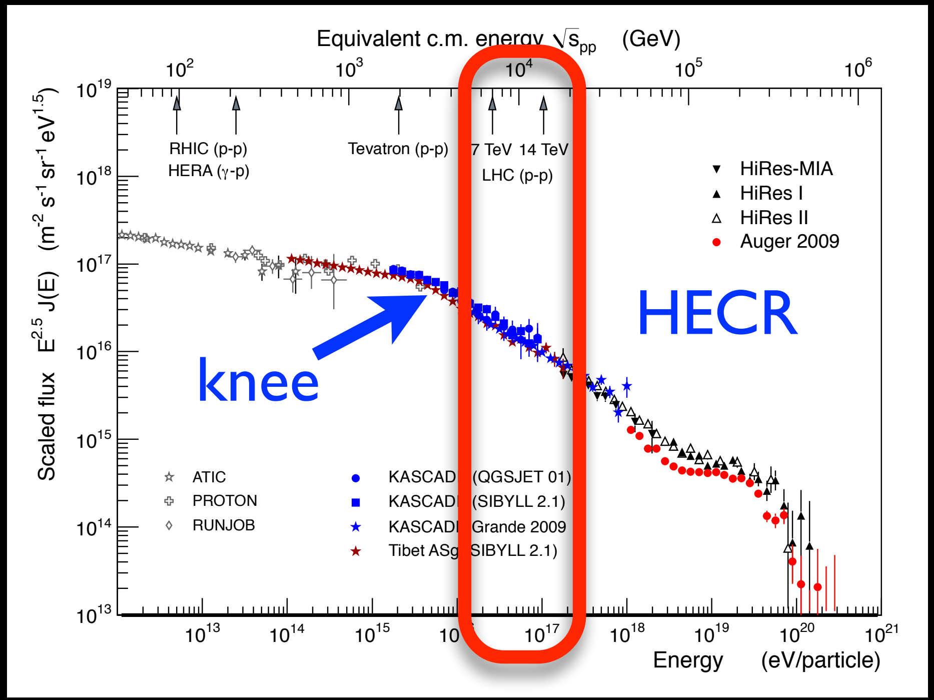
Offline Software and Tracking at the LHC

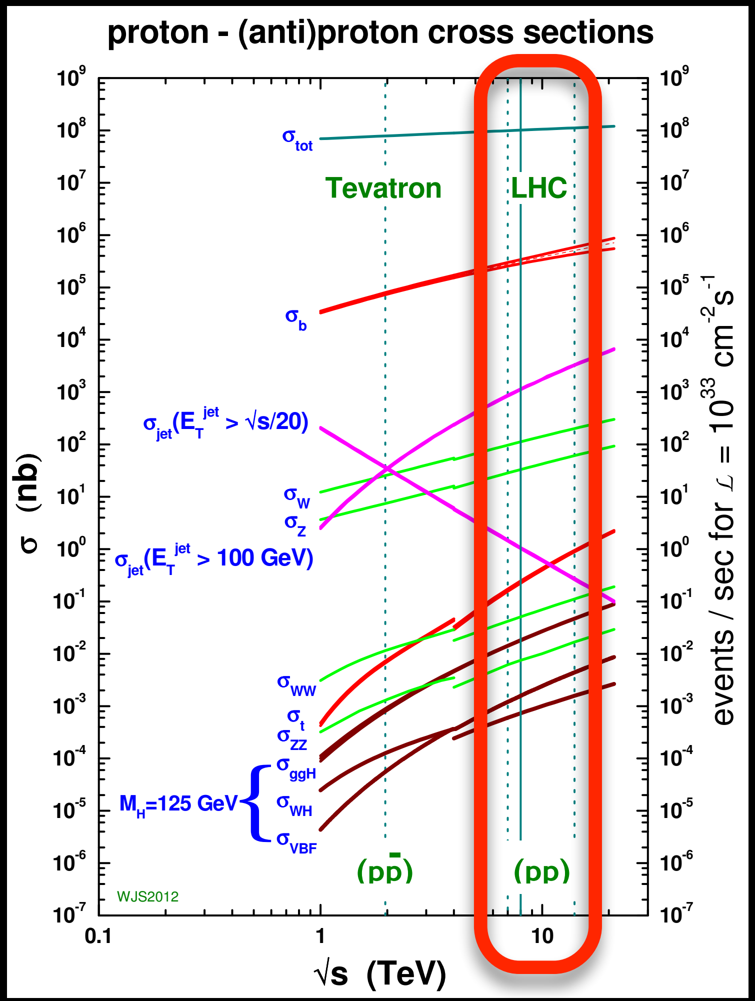
Developments in
offline software and tracking,
experience from Run-1,
recent shutdown upgrade activities,
as well as challenges ahead

- an ATLAS centric view -

Introduction: LHC

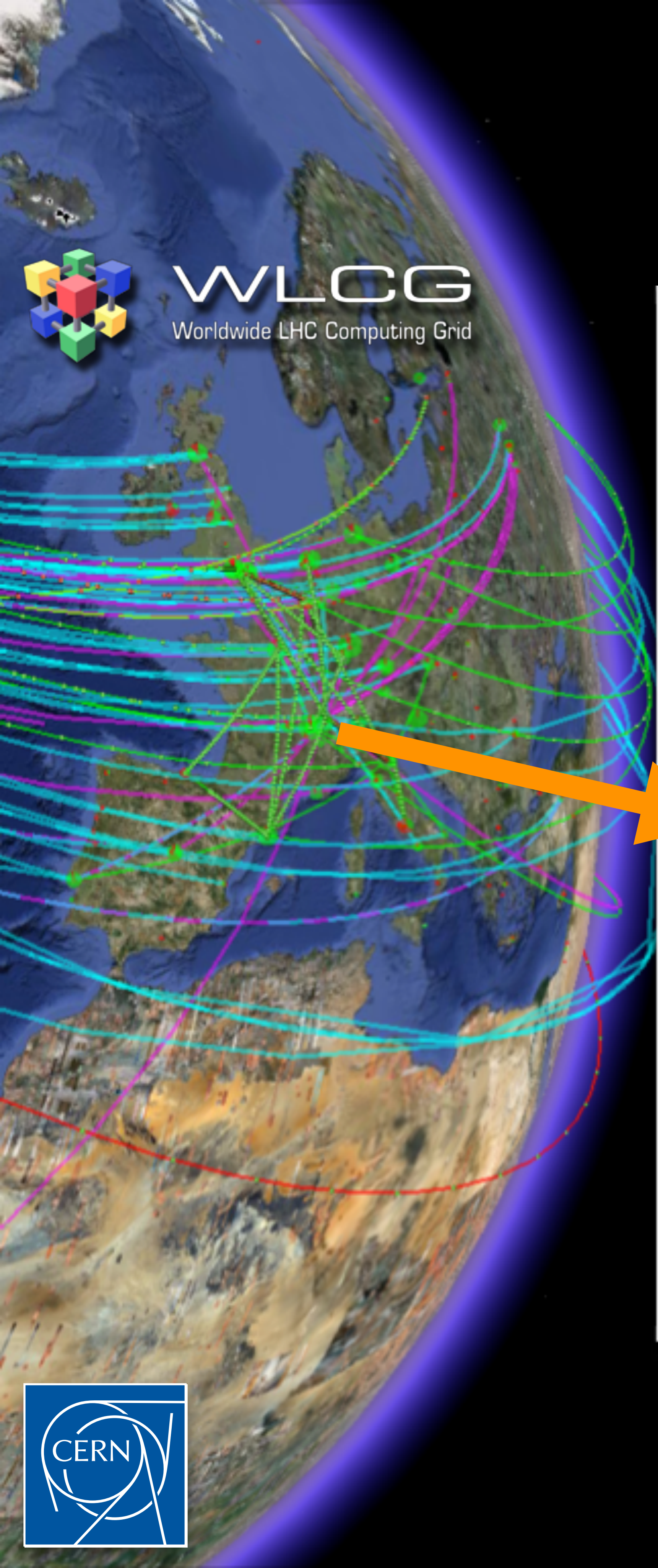
- LHC is a high energy and high luminosity proton-proton collider
  - ➡ design centre-of-mass energy is *14 TeV* and design luminosity is $\mathscr{L} = 10^{34}\ cm^{-2}s^{-1}$
  - ➡ first collider to reach energy regime of high energy cosmic rays (HECR)
  - ➡ expect ~23 *p-p* collisions at a bunch crossing frequency of 40 *MHz* (!)
- LHC is a unique machine
  - ➡ first collider to explore the physics at the *TeV* scale
  - ➡ excellent sensitivity to rare (new physics) processes
- expected production cross-sections
  - ➡ large inclusive *b*, *W/Z* and *top* production rates
    - LHC is a combined *b*-, *W/Z*- and *top*-factory
  - ➡ cross-section for jet and *W/Z* production orders of magnitude larger than e.g. expected for Higgs
  - ➡ total cross-section dominated by soft interactions

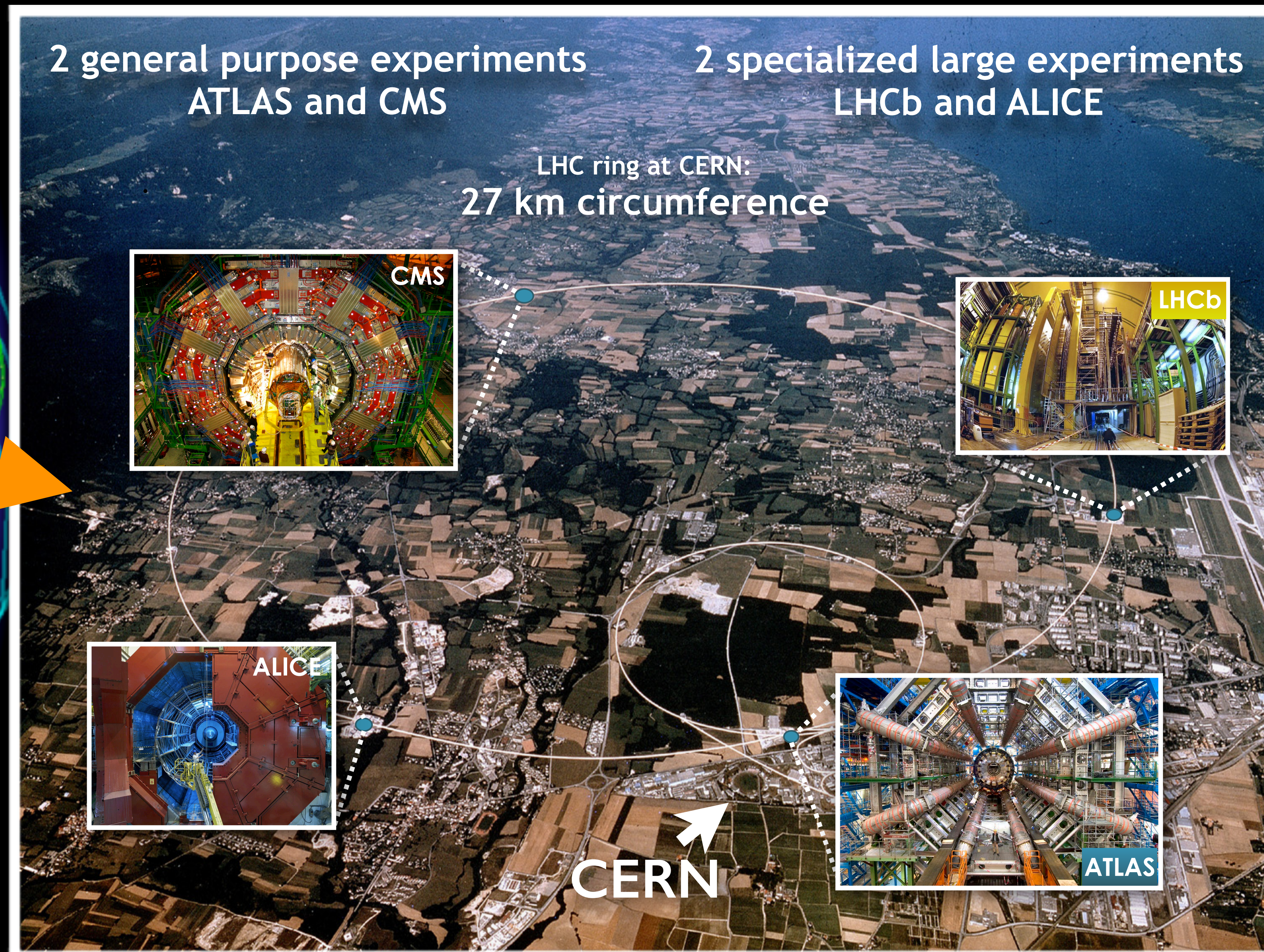




W.J. Stirling, private communication

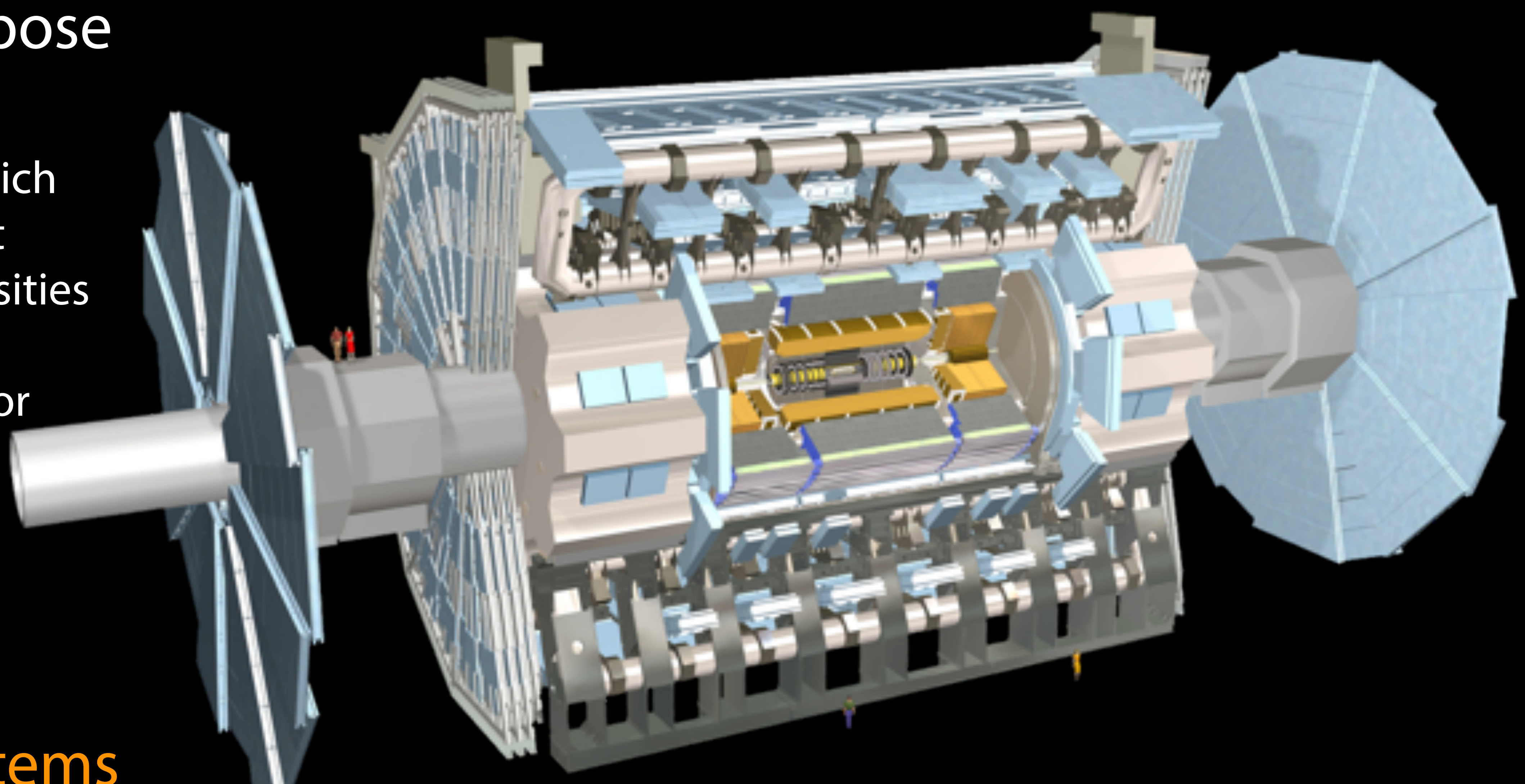
Introduction: LHC Experiments





ATLAS and Track Reconstruction

- general purpose detector
  - ➡ optimised for rich *p-p* program at design luminosities
  - ➡ as well good performance for heavy ions
- excellent calorimetry
- two major tracking systems
  - ➡ Inner Detector
  - ➡ Muon Spectrometer

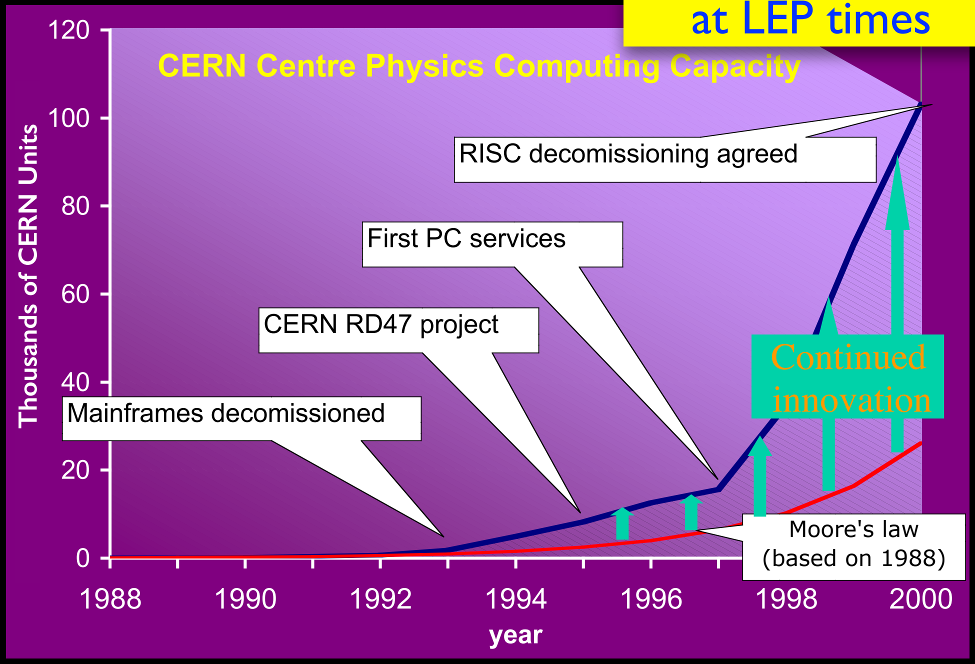


- tracking used all across object reconstruction
  - ➡ leptons ($e/\mu/\tau$) and photons
  - ➡ primary vertexing and flavour tagging
  - ➡ pileup removal for jet and missing ET reconstruction

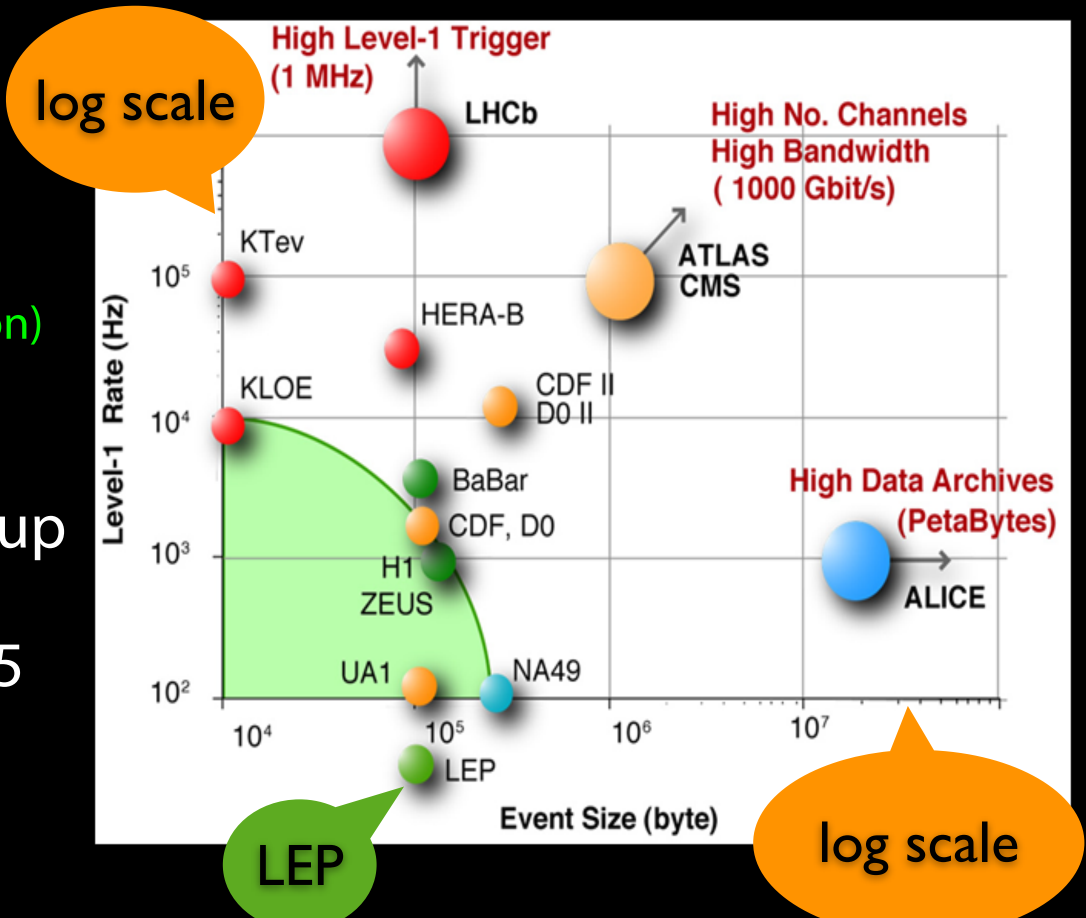
The early Times of LHC/ATLAS Software

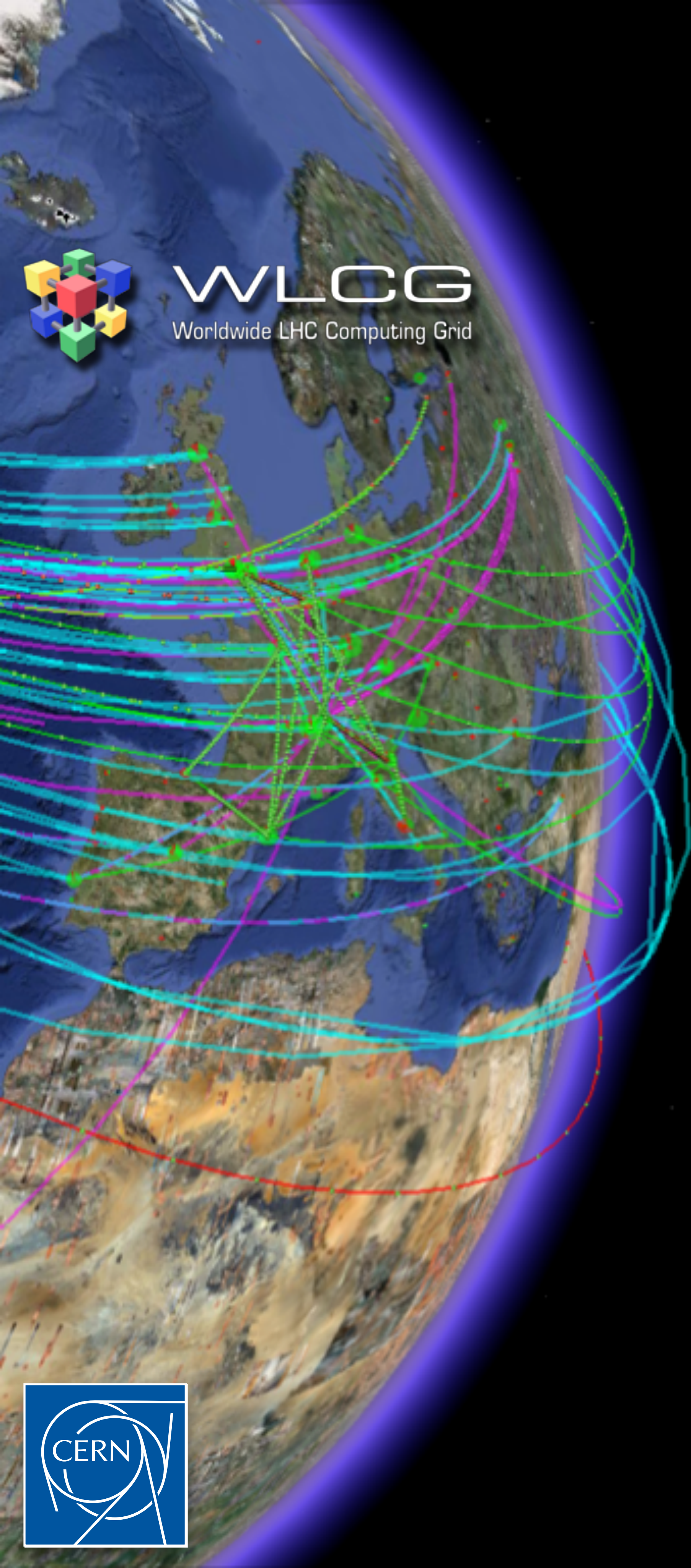
- project started during LEP era in '90s
  - ➡ LoI and TDRs done with infrastructure of the time
    - software in FORTRAN 77, CERNLIB incl. PAW, Geant3
    - general LINUX services at CERN started in 1997
- huge challenges ahead
  - ➡ LHC is a high energy and high luminosity machine
    - unprecedented trigger rates, event sizes, pileup
  - ➡ lots of questions to answer...
    - design the High Level Trigger systems ?
      (can it be done in software, even re-using offline code)
    - how to build up the software infrastructure ?
      (move to C++/OO, learn from BaBar and CDF/D0 Run-2 preparation)
    - a computing infrastructure matching the needs ?
      (building "the" LHC computing centre at CERN wasn't an option)
    - how to do high performance tracking at LHC pileup
      (and how to do this within the available computing resources)
  - ➡ not to forget, LHC startup was supposed to be 2005
    (well, it came different after all)

CERN computing at LEP times



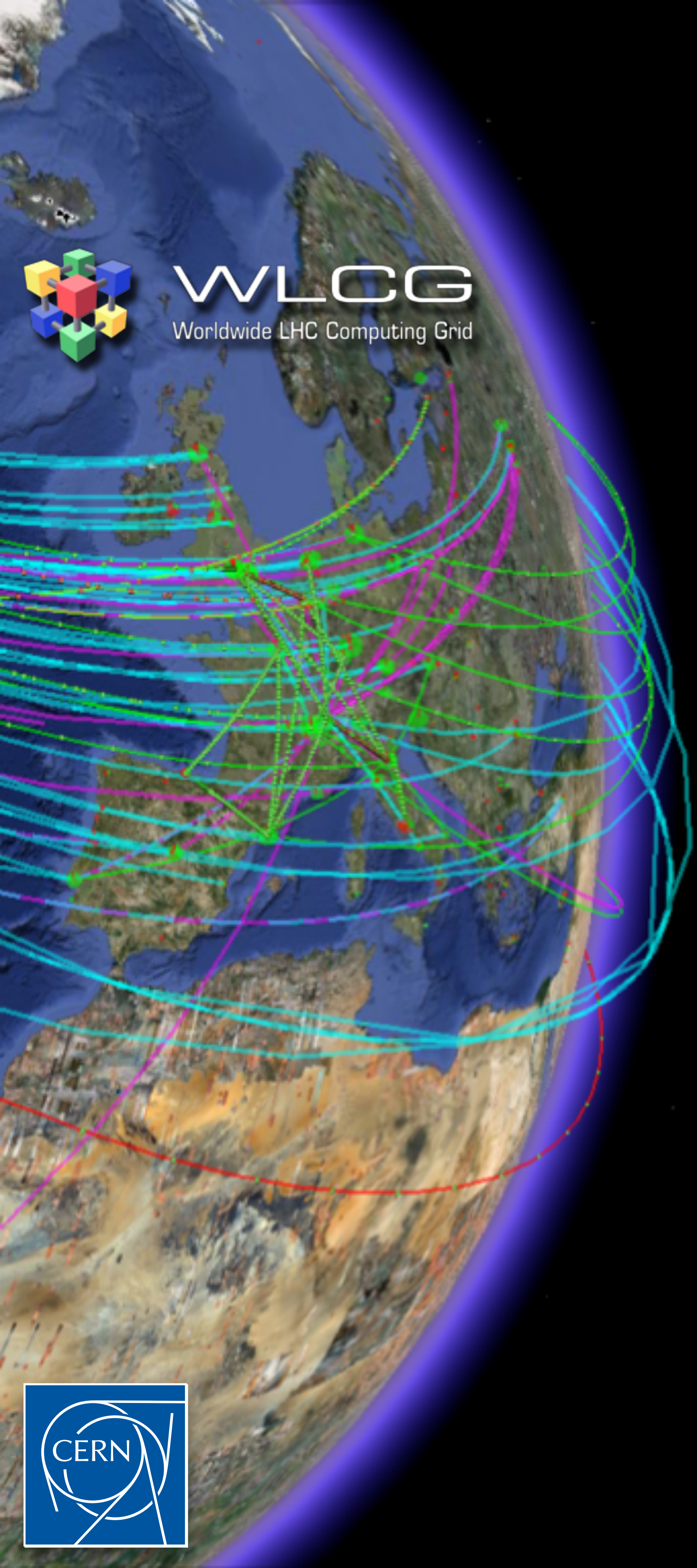
S.Bethke, LHC Computing Review, 2001





Outline of this Talk

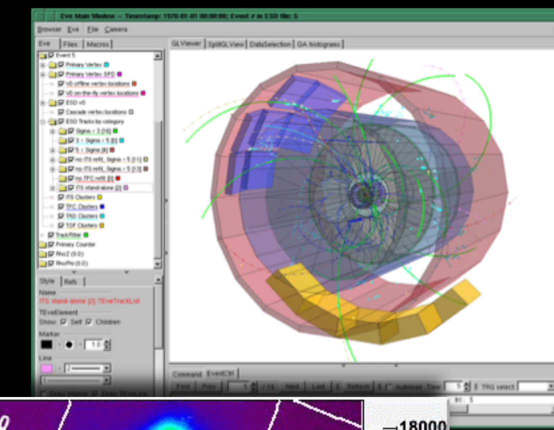
- building up the software of the experiments
- ATLAS tracking software and its concepts
- early physics and experience from Run-1
- the Higgs discovery
  - ➡ the role of the offline software
- preparing for Run-2
  - ➡ first upgrades of the offline software
- future offline software challenges
- summary and outlook

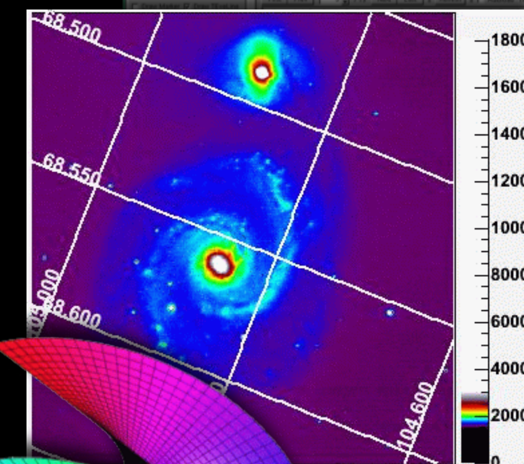


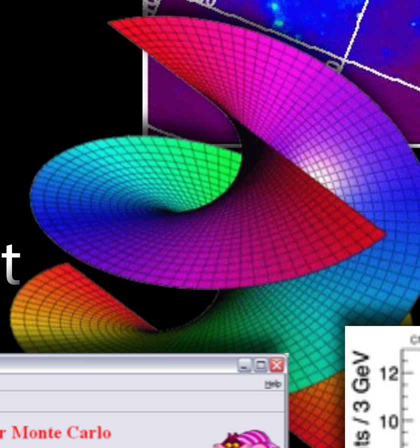
Building up the Software of the Experiments

ROOT (Rene and Rdm OO Technology*)

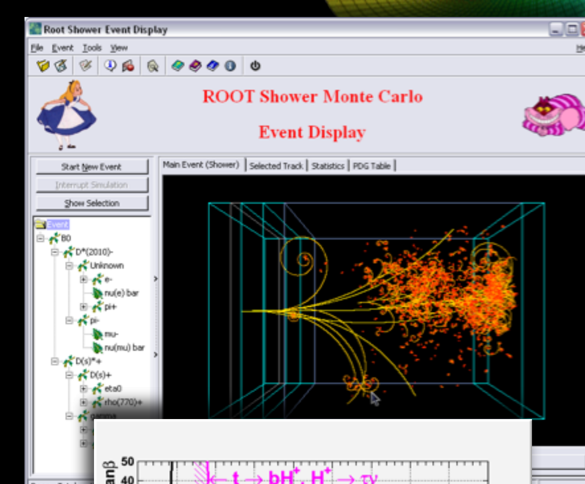
- project started 1995
  - ➡ by R.Brun and F.Rademacher (hence the name)
    - OO framework, having in mind the future LHC needs
    - as well, provided alternative to Objectivity/DB at the time
    - 1998 selected by Fermilab for Run-2 experiments
  - ➡ became "the standard" for HEP and LHC data analysis
    - used by Astrophysics, other sciences and fields
  - ➡ core team at CERN, effort at FNAL and large community input

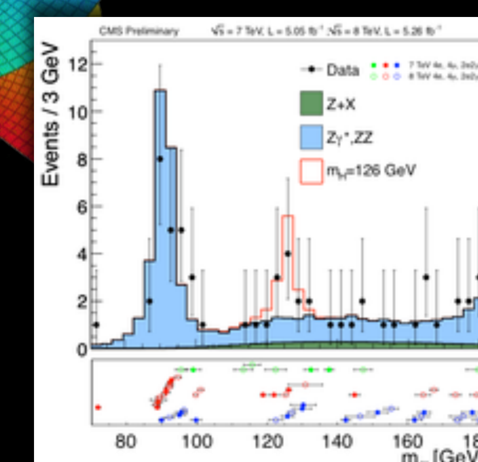


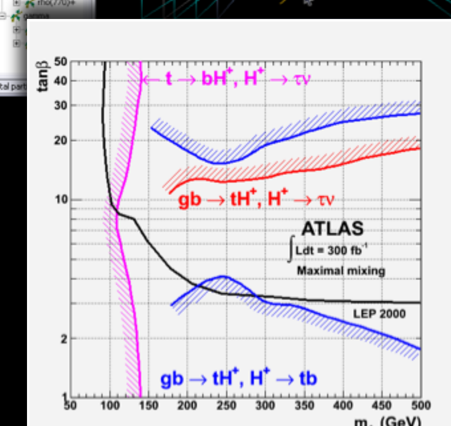




- framework for interactive analysis
  - ➡ visualisation, math libraries, I/O
    - LHC data is based on ROOT persistency
  - ➡ distribution includes suite of other tools
    - xrootd, TMVA, RooFit/RooStats, ...
  - ➡ total about 1.7 *million* lines of code
    - OpenHUB "estimated cost" is 27 M$
      https://www.openhub.net/p/ROOT/estimated_cost





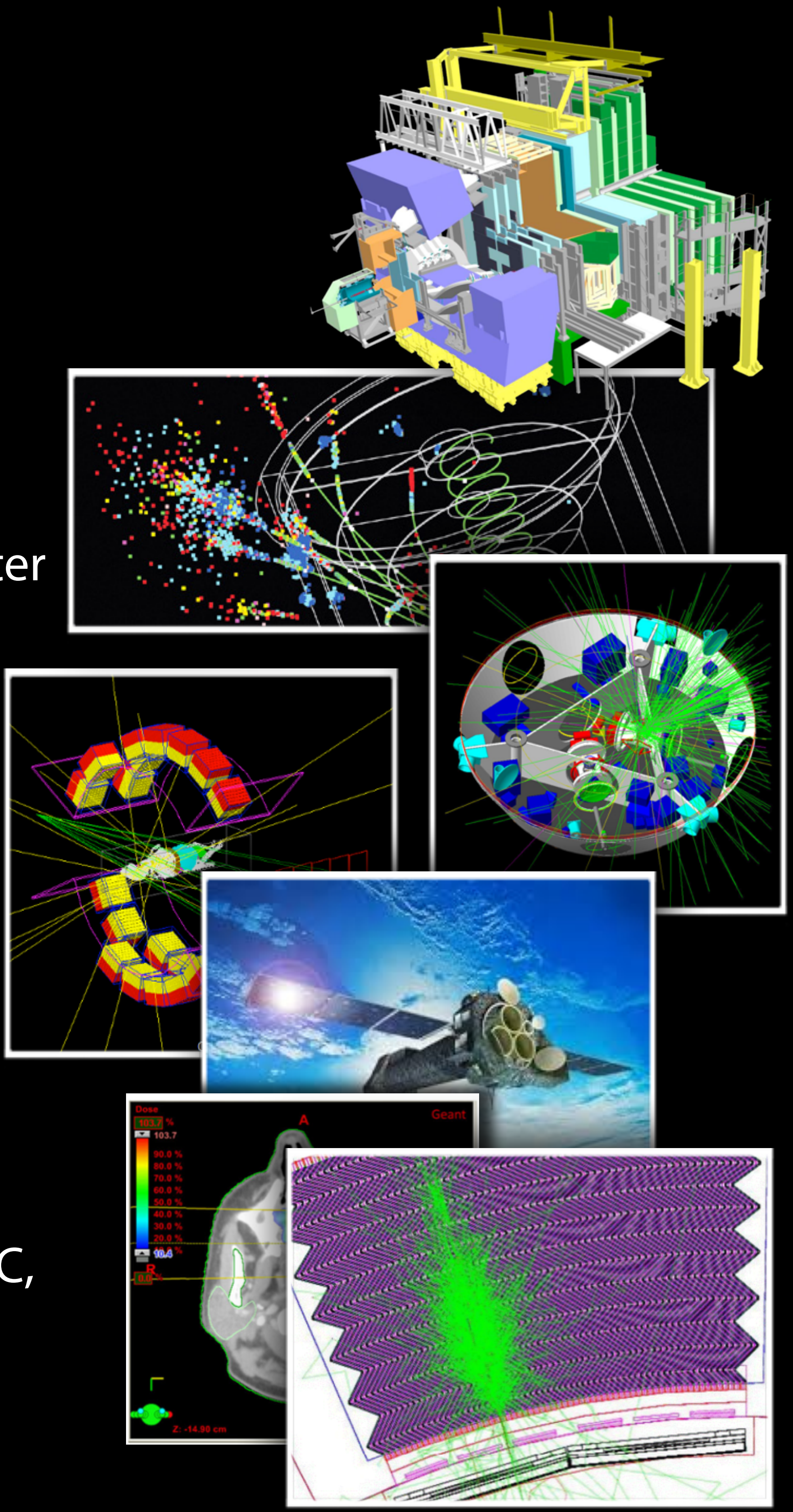


ROOT Data Analysis Framework

CERN

*http://ph-news.web.cern.ch/content/interview-rené-brun

Geant 4



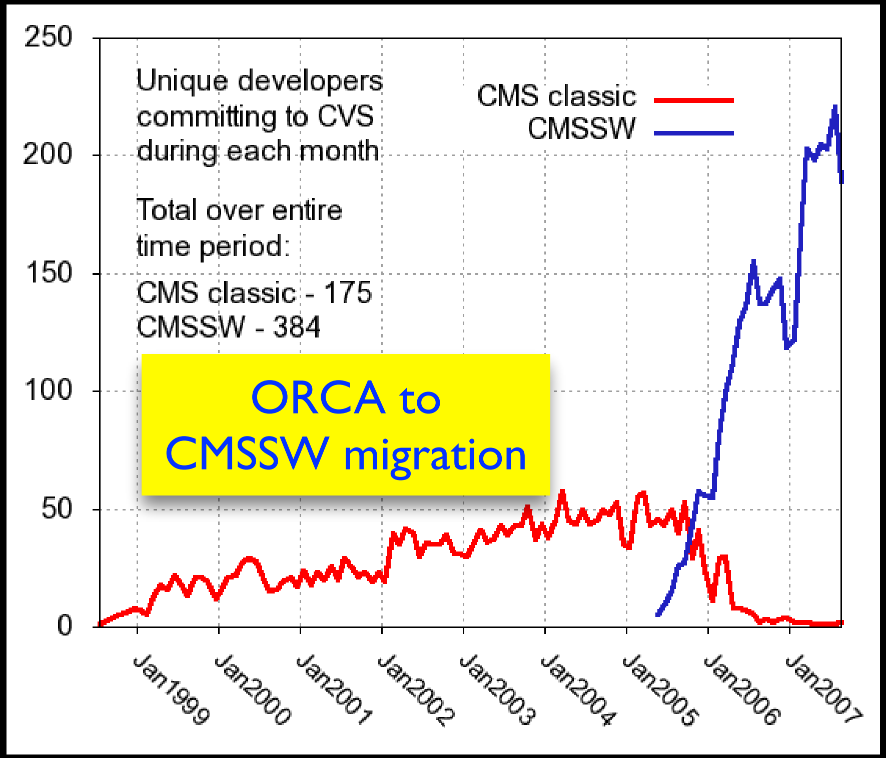
- Geant4 Collaboration started in 1999
  - ➡ successor of Geant series toolkits developed at CERN
    - early studies at KEK and CERN resulted in RD44
    - OO simulation of passage of particles through matter
  - ➡ today effort at many large laboratories:
    CERN, FNAL, SLAC, KEK, ESA/ESTEC, ...
  - ➡ detector simulation for CMS, LHCb, ATLAS, (ALICE), ...
  - ➡ used by nuclear, accelerator and medical physics,
    as well as space science
  - ➡ about 2.1 *million* lines of code
    - OpenHUB "estimated cost" is 33 M$
      https://www.openhub.net/p/geant4/estimated_cost

- equally important: event generators
  - ➡ Alpgen, Jimmy, Pythia6/8, Tauola(++), Sherpa, HepMC,
    Herwig(++), Photos, etc.
  - ➡ C++ and Fortran, about 1.4 *million* lines of code

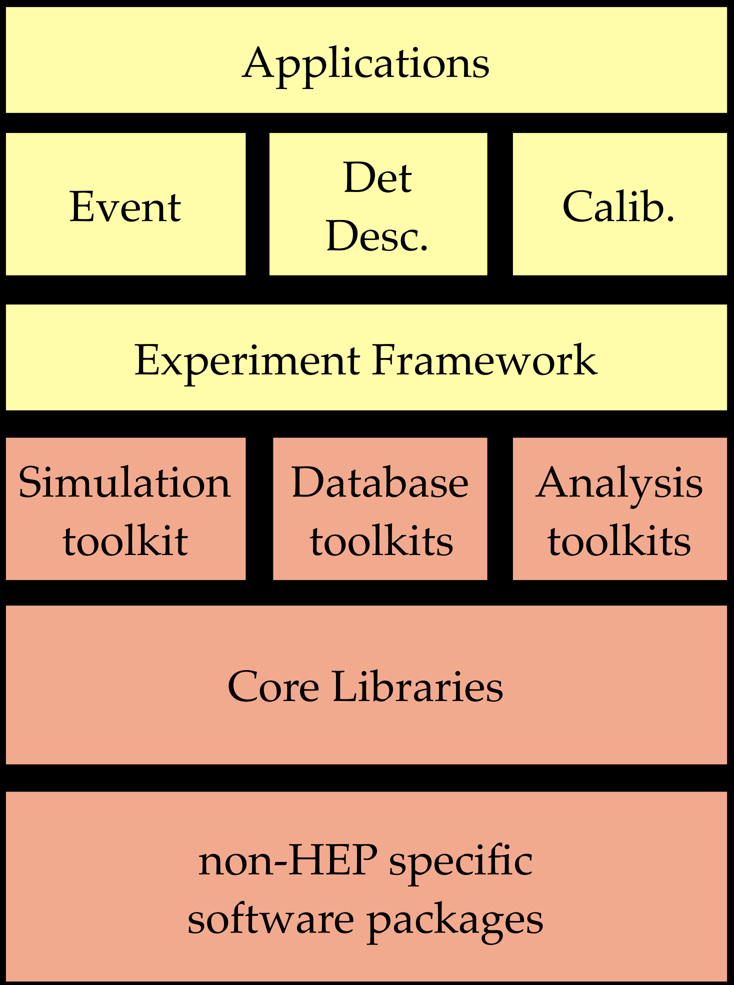
Software of Experiments

- all developed their own OO frameworks
  - ➡ ORCA (CMS), AliRoot based on ROOT (Alice), GAUDI (LHCb)
  - ➡ ATLAS added its layer to GAUDI and called it ATHENA
- CMS started 2005 CMSSW to replace ORCA
  - ➡ based on experience from FERMILAB experiments
    - huge effort, took >3 years
  - ➡ today a full CMSSW release has 7.5 *million* lines of code
    - OpenHUB "estimated cost" is 125 M$

      https://www.openhub.net/p/cms-sw-cmssw/estimated_cost
    - framework itself is only a fraction of this
- software stacks of the experiments
  - ➡ applications implemented in framework
    - detector simulation, trigger, reconstruction, ...
  - ➡ based on common software toolkits
    - development organised within LCG Application Area (Pool, Cool, Coral, Geant4, Root, ...)



P.Elmer et al.

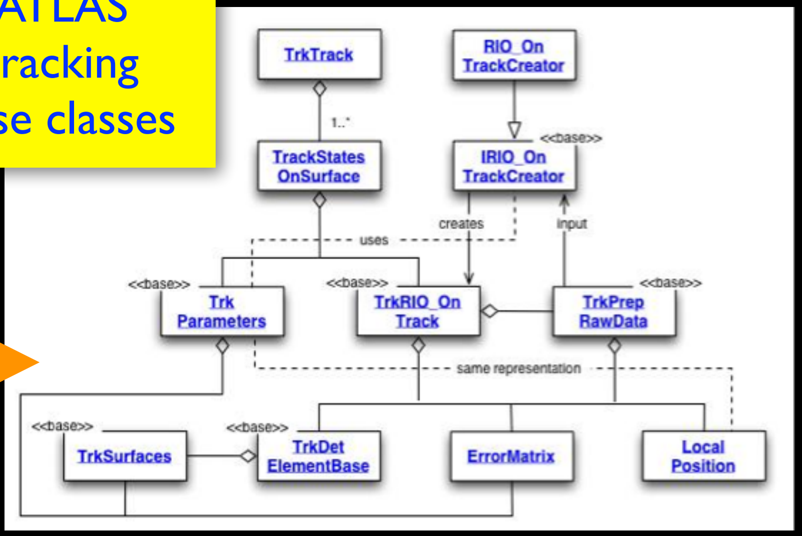


Building the Offline Reconstruction

- migration to C++ based reconstruction
  - ➡ existing FORTRAN algorithmic code often state of the art
    - new ideas from LEP experience, later BaBar and CDF/D0
  - ➡ lot of work (too much) went into OO design
    - "hip" at the time, today we have to back off again (see later)

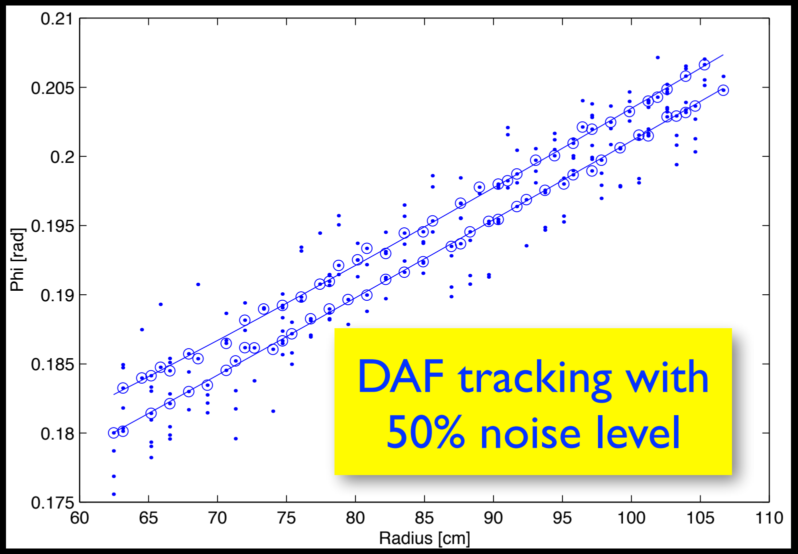
ATLAS tracking base classes

M.E. et al.

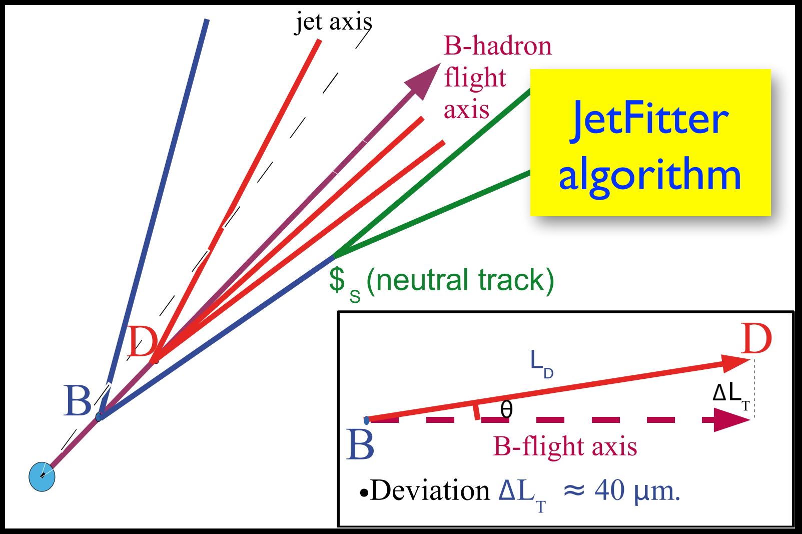


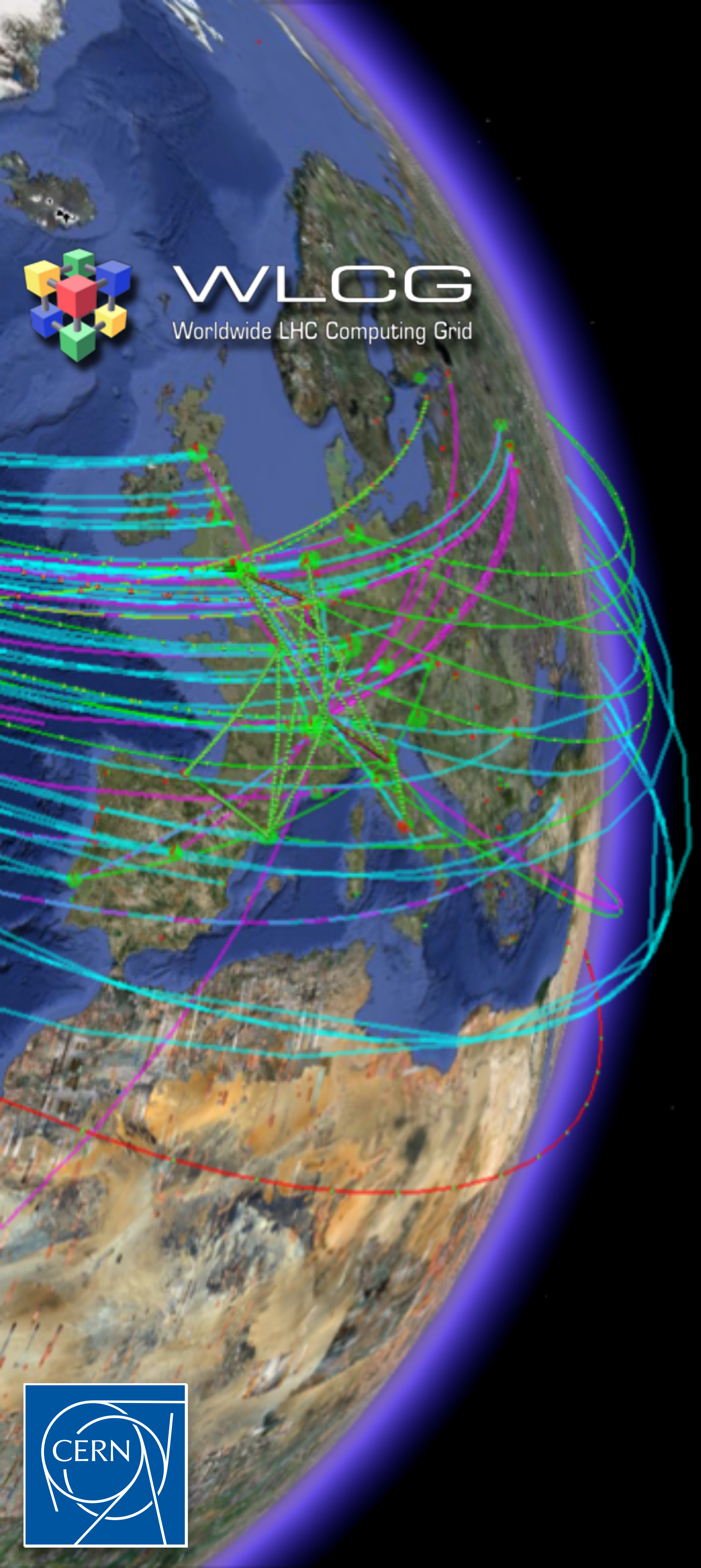
- new ideas to meet the LHC challenges
  - ➡ driver for innovation, lots of examples:
    - Deterministic Annealing Filters (Com.Phys.Com. 120 (1999) p.197) ~ tracking in ATLAS TRT at high pileup
    - STEP (J. Instr. 4 (2009) p.04001) ~ Runge-Kutta field integration for ATLAS+CMS muon tracking
    - JetFitter (J.Phys.Conf.Ser. 119 (2008) 032032) ~ novel secondary vertexing in jets for b-tagging
    - FastJet (hep-ph/0512210) ~ fast jet finding
    - Particle Flow (hep-ex/0810.3686) ~ reconstruction in CMS
  - ➡ later significant influx from CDF/D0, example:
    - Jet-Vertex-Fraction (hep-ex/0612040) ~ pileup suppression

A.Strandlie et al.



G.Pacquadio

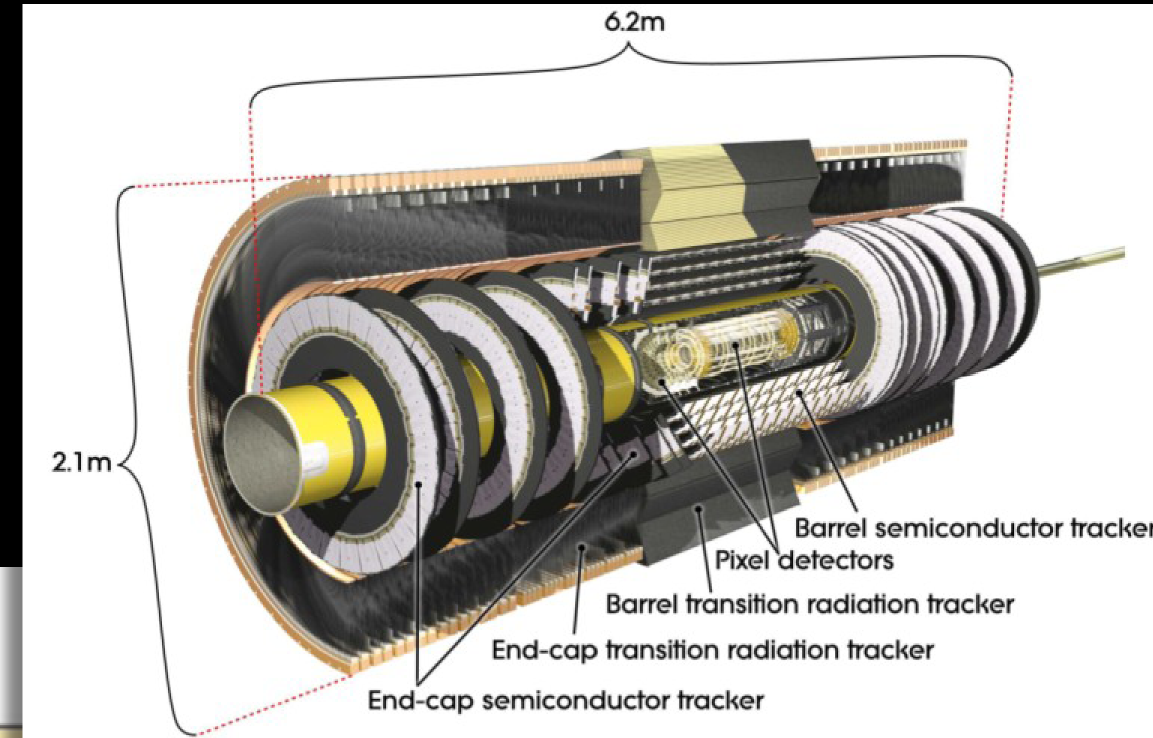


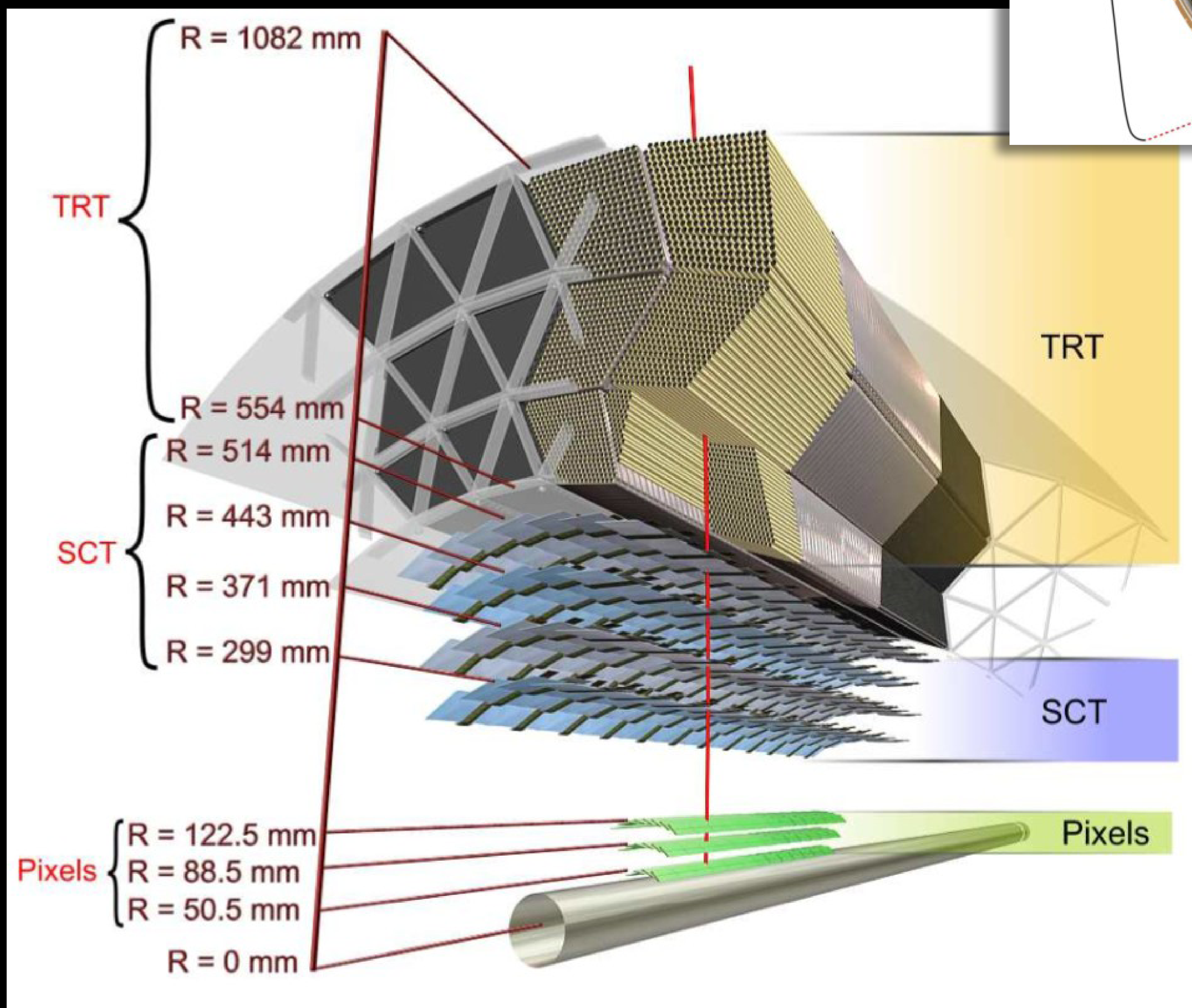


ATLAS Tracking Software and its Concepts

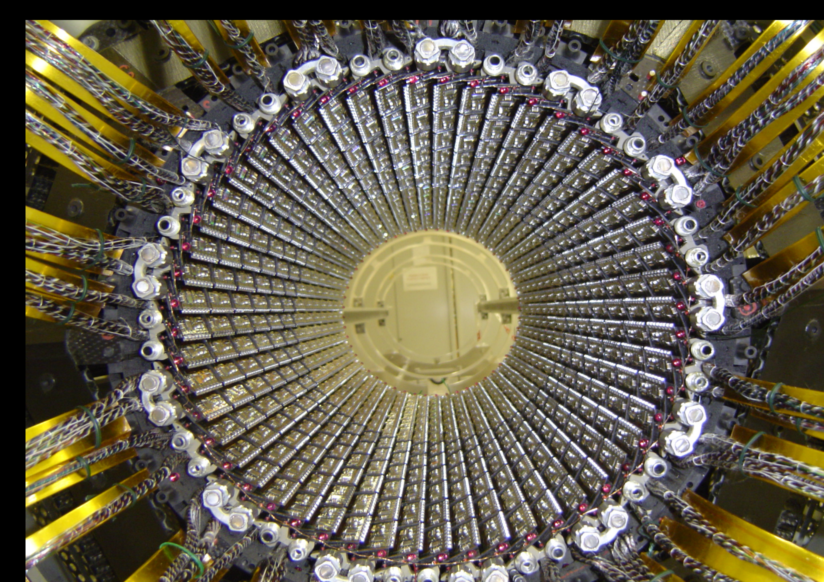
ATLAS Inner Detector

- optimised for 24 pileup events



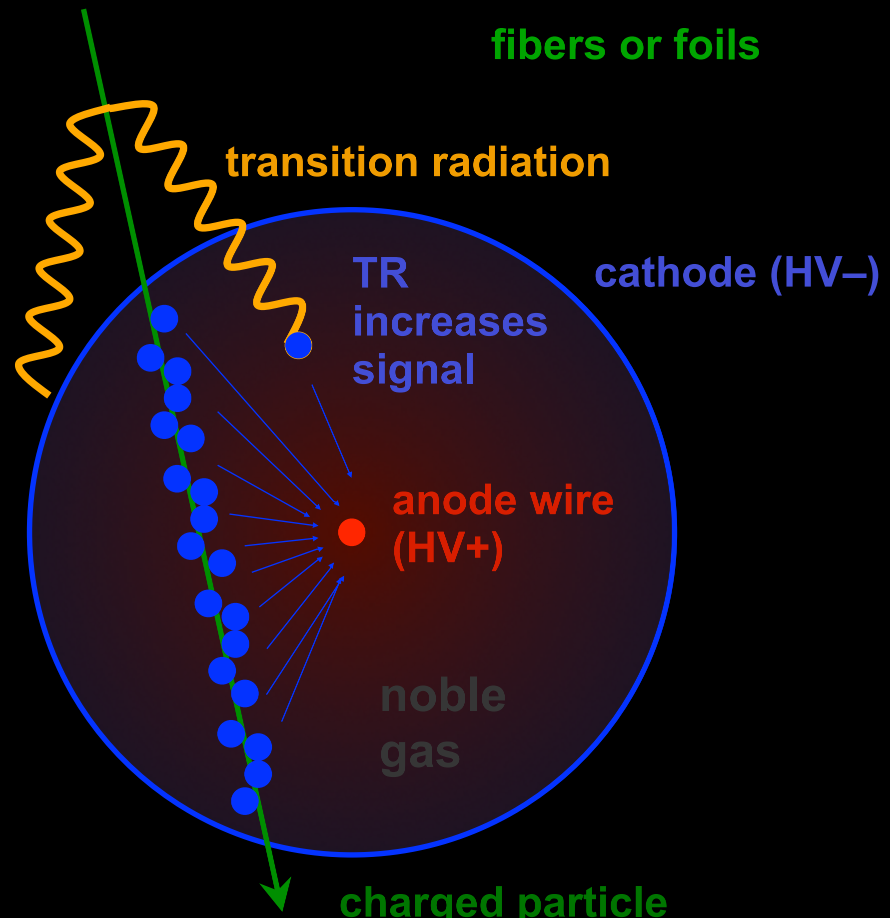


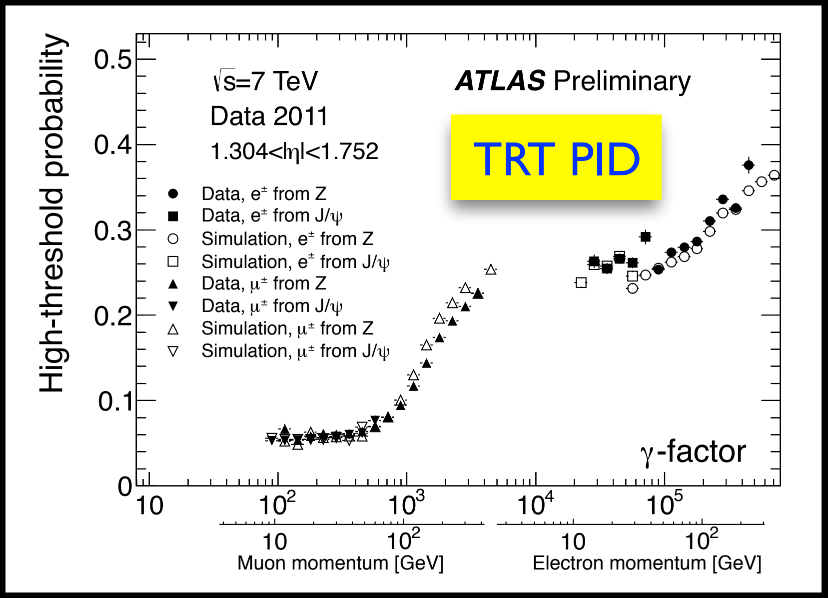
- barrel track passes:
  - ➡ 3 Pixel layers 250 *mm* thick
  - ➡ 4x2 Si strips on stereo modules12 *cm* x 80 *mm*, 285 *mm* thick
  - ➡ ~36 TRT 4 *mm* straws

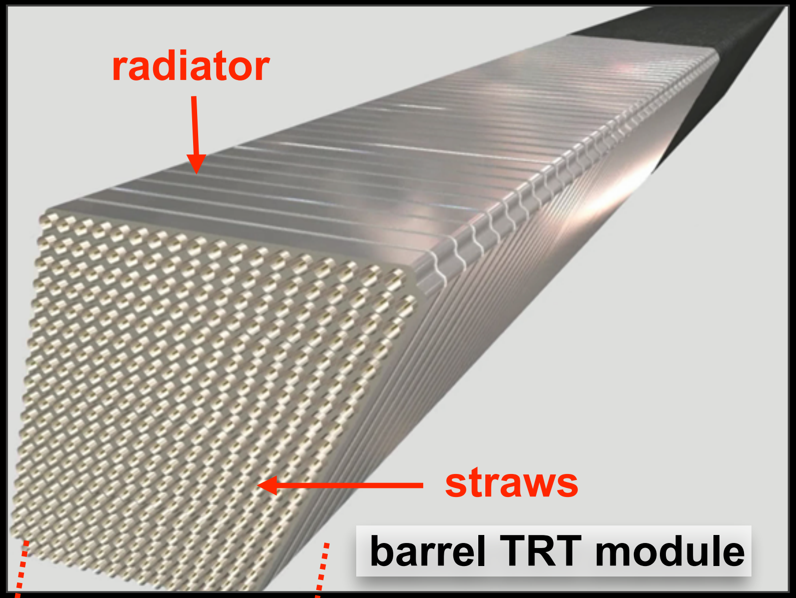


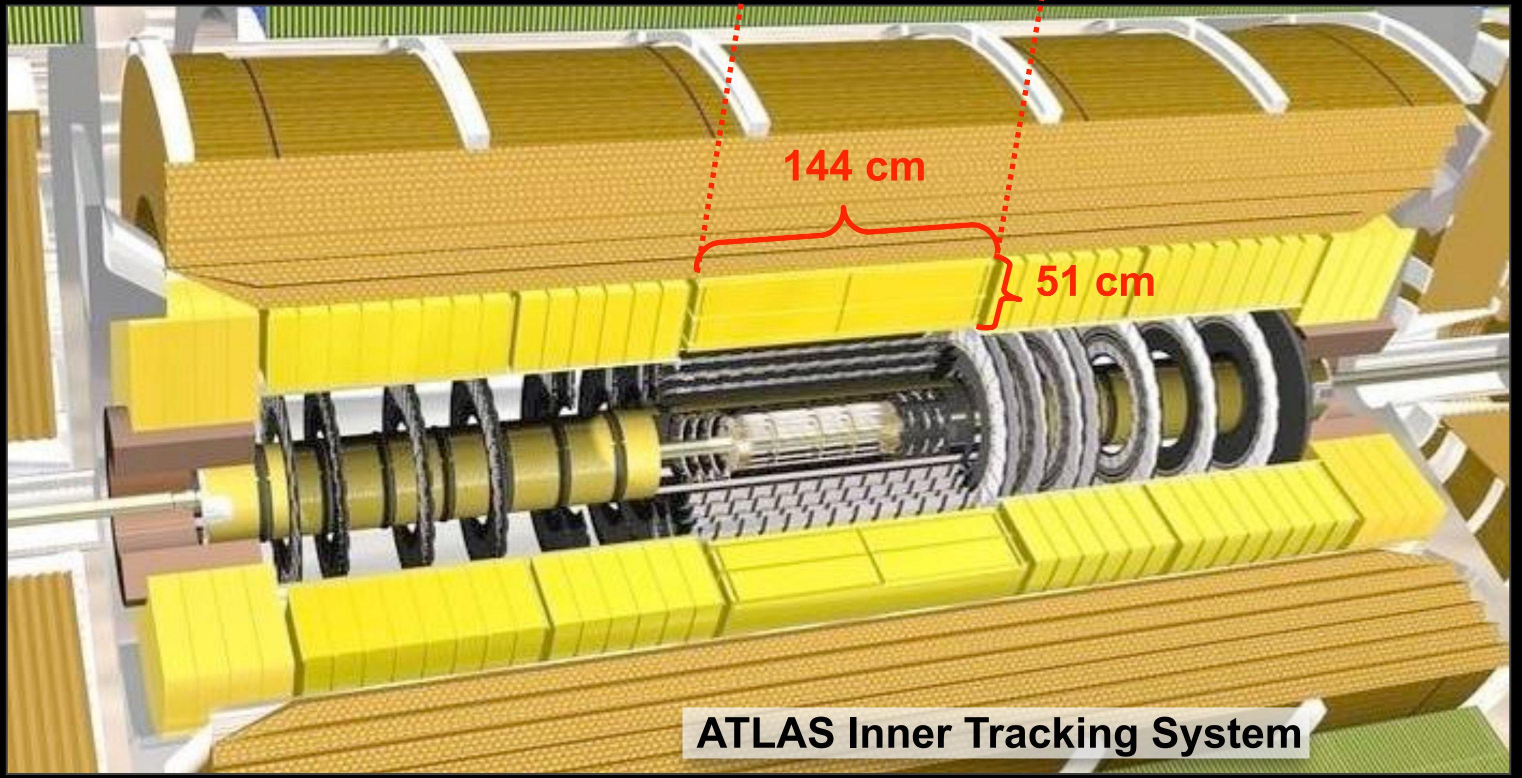
Electron Identification in the ATLAS TRT

➡ e/π separation via transition radiation: polymer (PP) fibers/foils interleaved with drift tubes



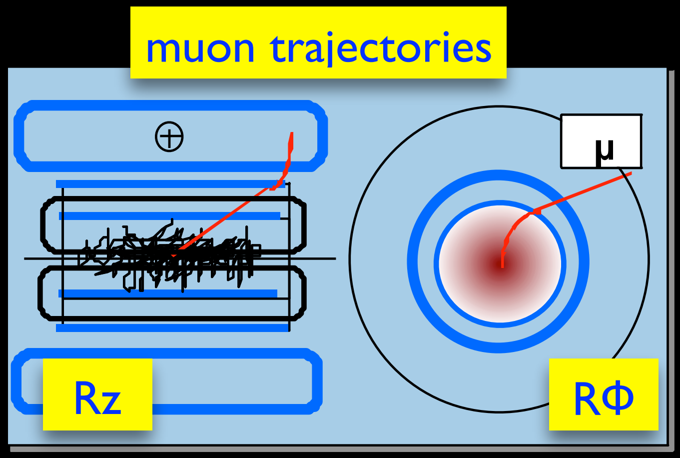


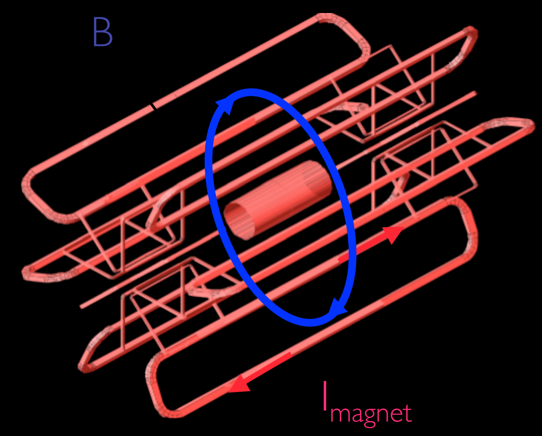


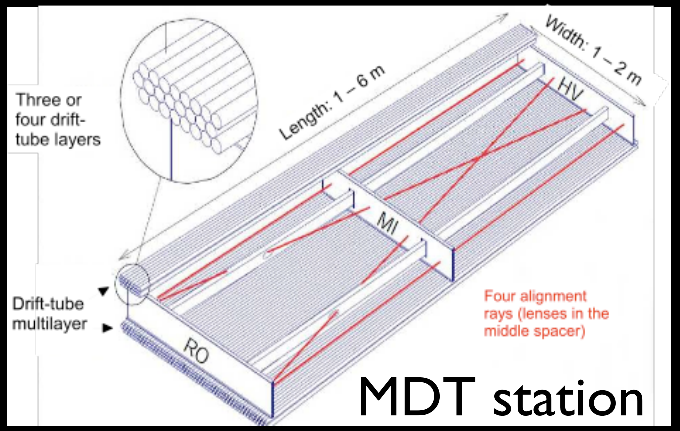


➡ electrons radiate → higher signal
- PID info by counting high-threshold hits component precisely

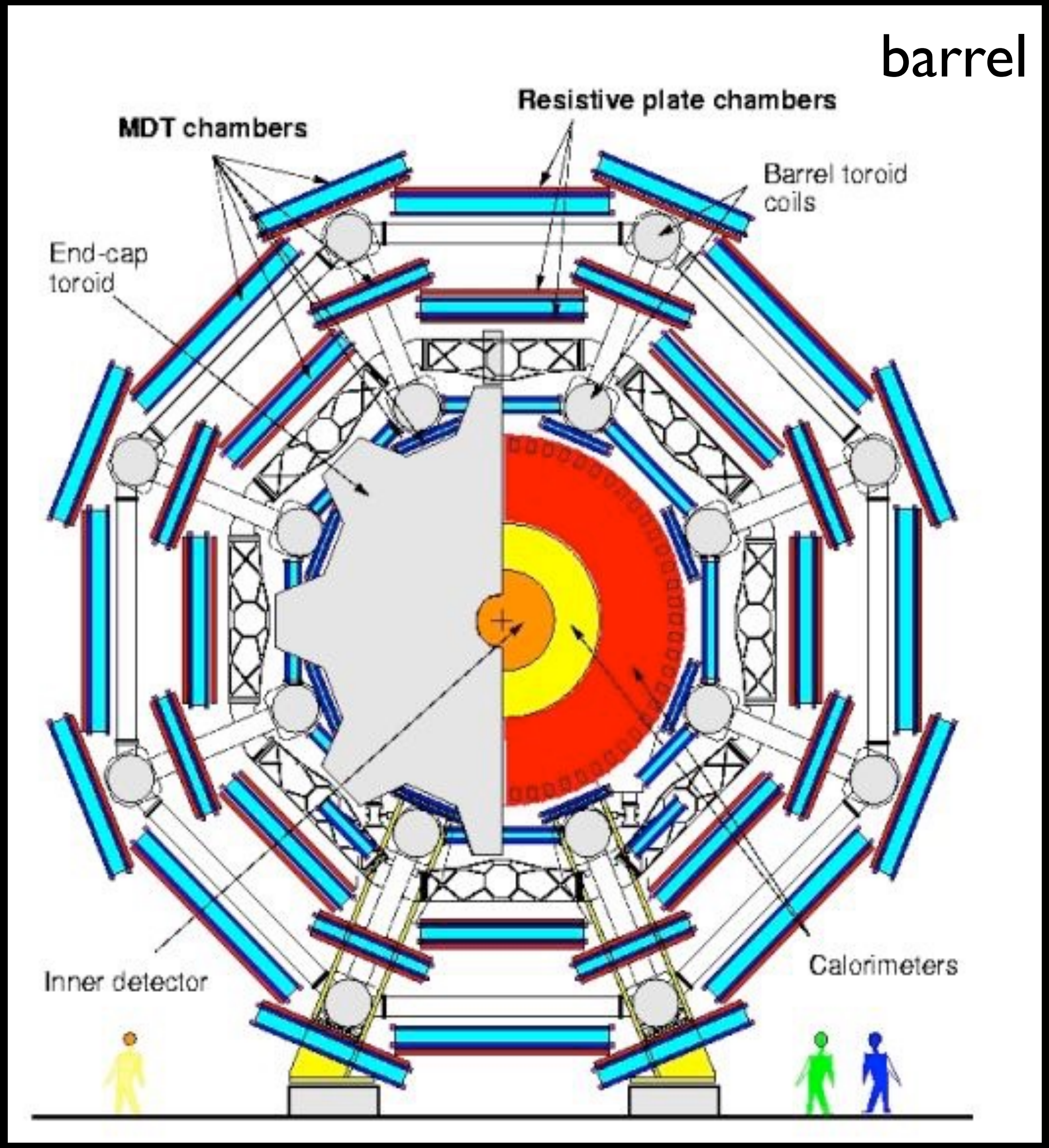
ATLAS Muon Spectrometer

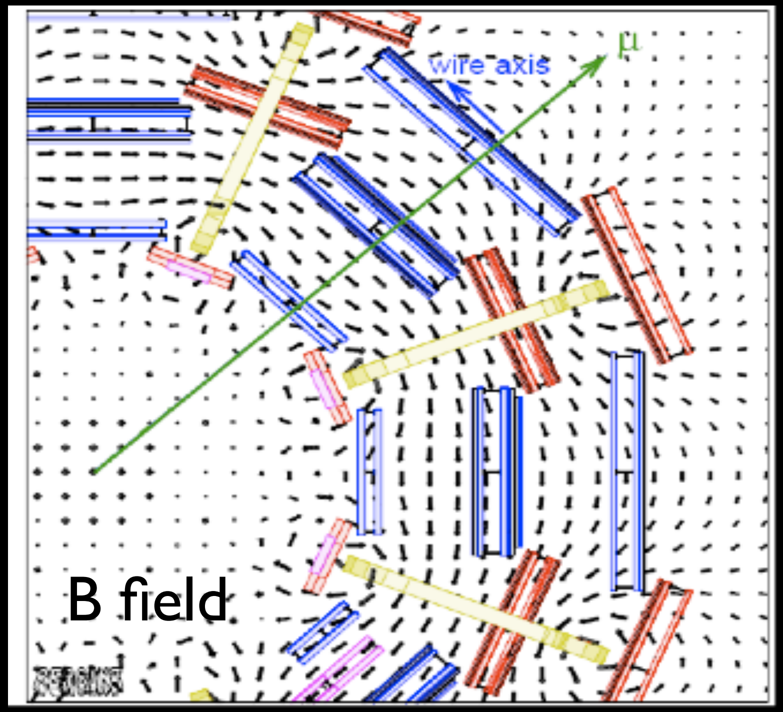


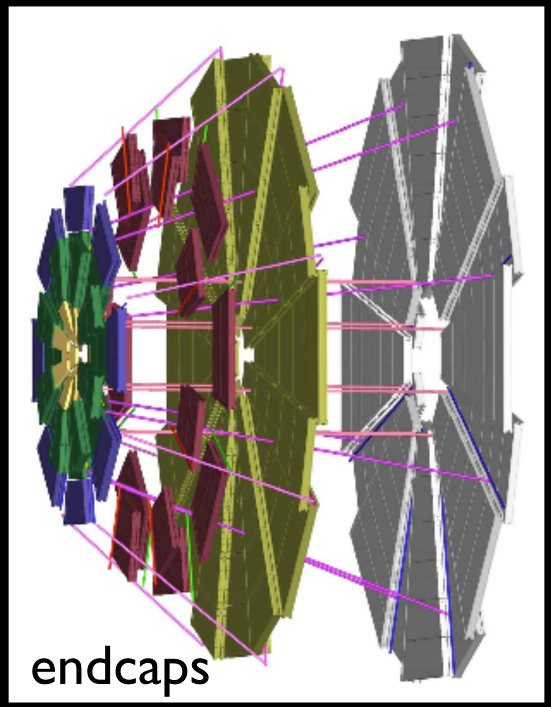




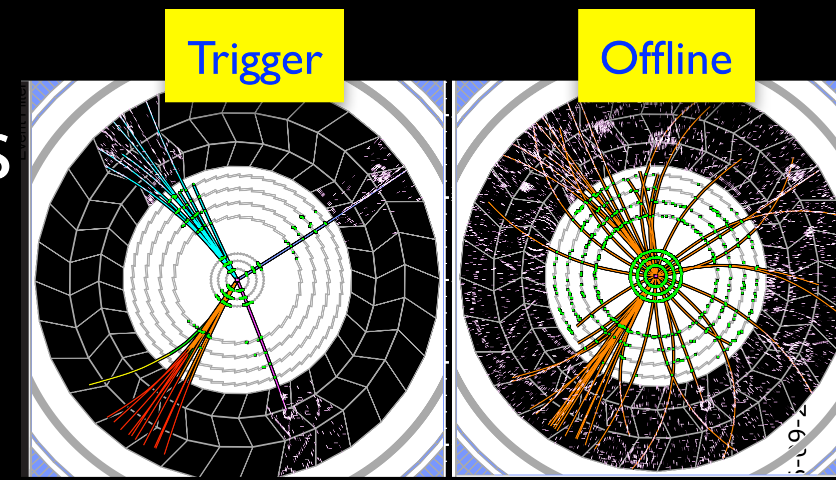
- a huge system
  - ➡ 4 different technologies (MDT,CSC,RPC,TGC)
  - ➡ large area (10.000 m^2)
  - ➡ many channels (1 M)
- toroid field configuration
  - ➡ large magnetic field variations in toroid
  - ➡ field 4 *Tesla* near coils
- optical alignment system





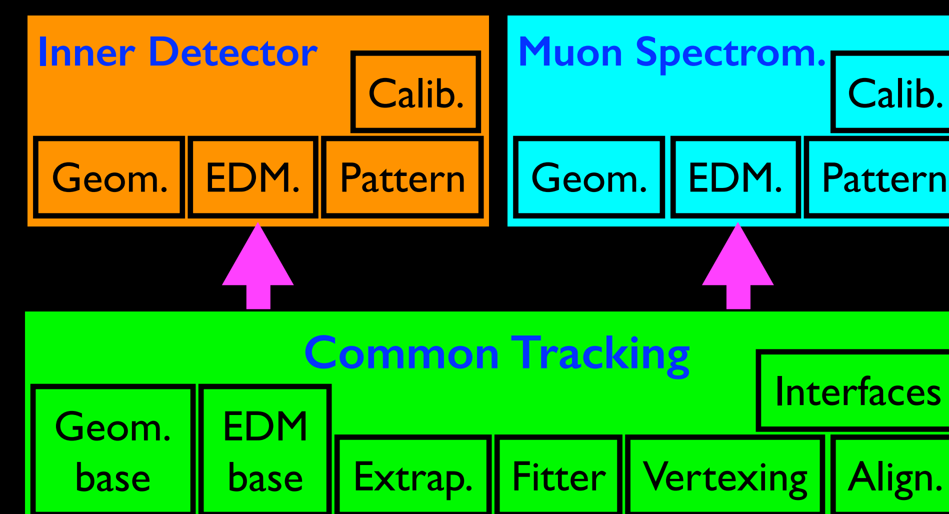


Tracking Software Concepts



- developing the tracking for LHC detectors
  - ➡ how to do high performance tracking at LHC pileup ?
    - and how to do this within the available resources ?
    - keeping in mind trigger and offline use-cases

- ATLAS has 2 tracking systems, 7 different detector technologies
  - ➡ reflected in high level software design
    - detector independent "Common Tracking" layer
    - detector specific layers building on it
  - ➡ base classes, interfaces, mathematical tools all in common tracking layer
    - e.g. event data model, extrapolation, fitters...

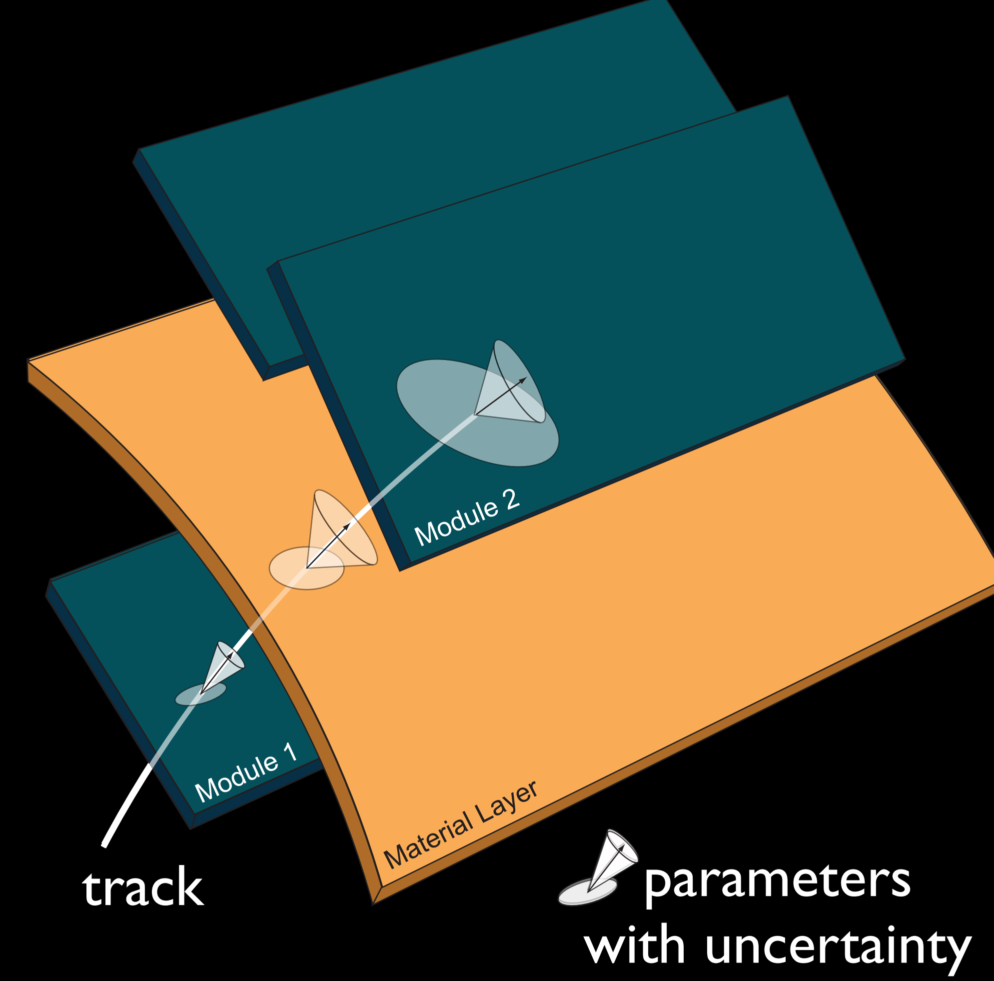


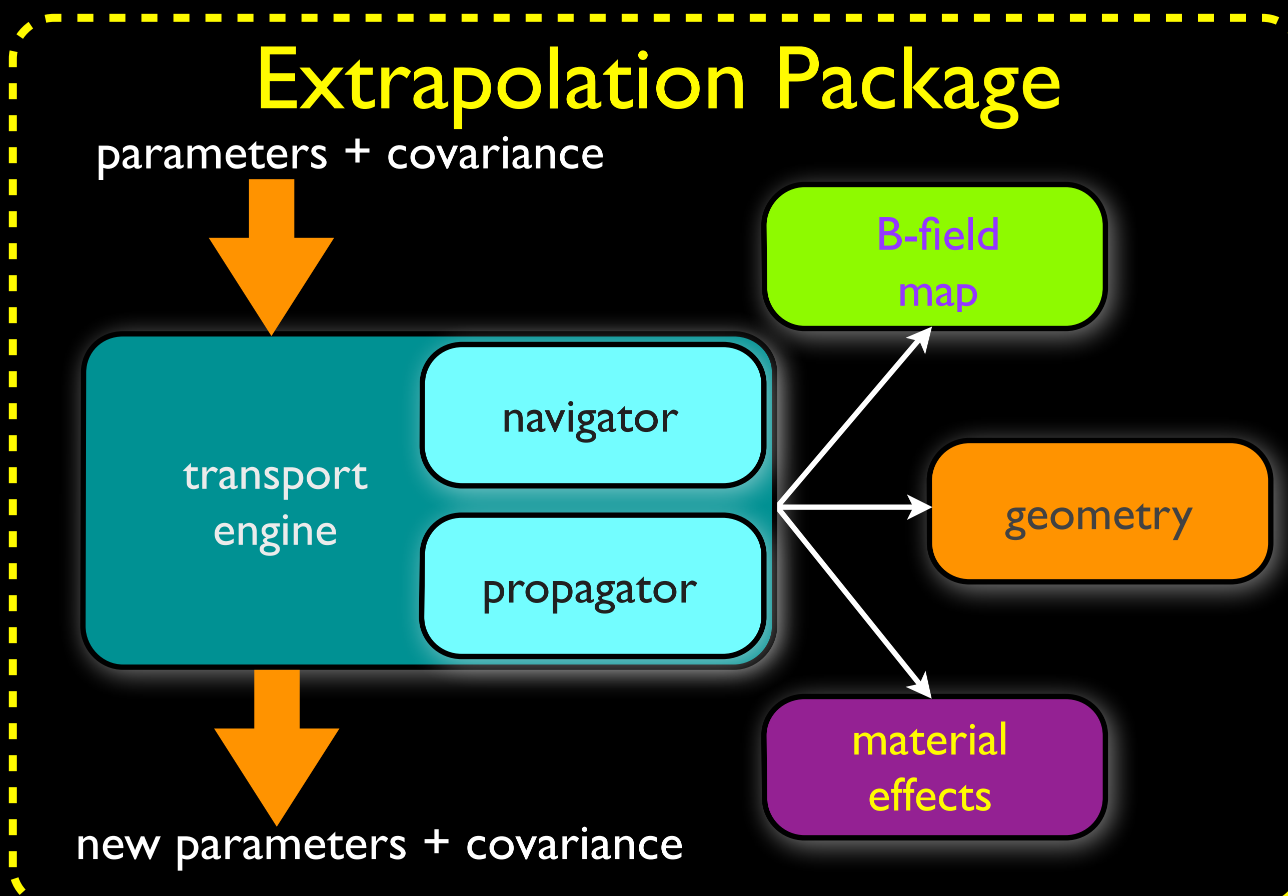
- informal collaboration by CMS and ATLAS
  - ➡ R&D on fitting techniques (e.g. Deterministic Annealing Filters)
  - ➡ R&D on novel tracking geometries with embedded navigation (see later)
  - ➡ R&D on modern Runge-Kutta field integration techniques
    (Runge-Kutta-Nystrom with continuous energy loss and multiple scattering (STEP), J. Instr. 4 (2009) p.04001)
  - ➡ later series of LHC alignment workshops across all 4 experiments

The Extrapolation Package

- parameter transport engine used in tracking software
  - ➡ central tool for pattern recognition, track fitting, etc.
  - ➡ parameter transport from **surface to surface**, including covariance
  - ➡ encapsulates the track model, geometry and material corrections

- main components
  - ➡ modern Runge-Kutta propagators
  - ➡ navigation system (see below)
  - ➡ B-field map with caching
  - ➡ geometry model (see below)
  - ➡ material effects corrections





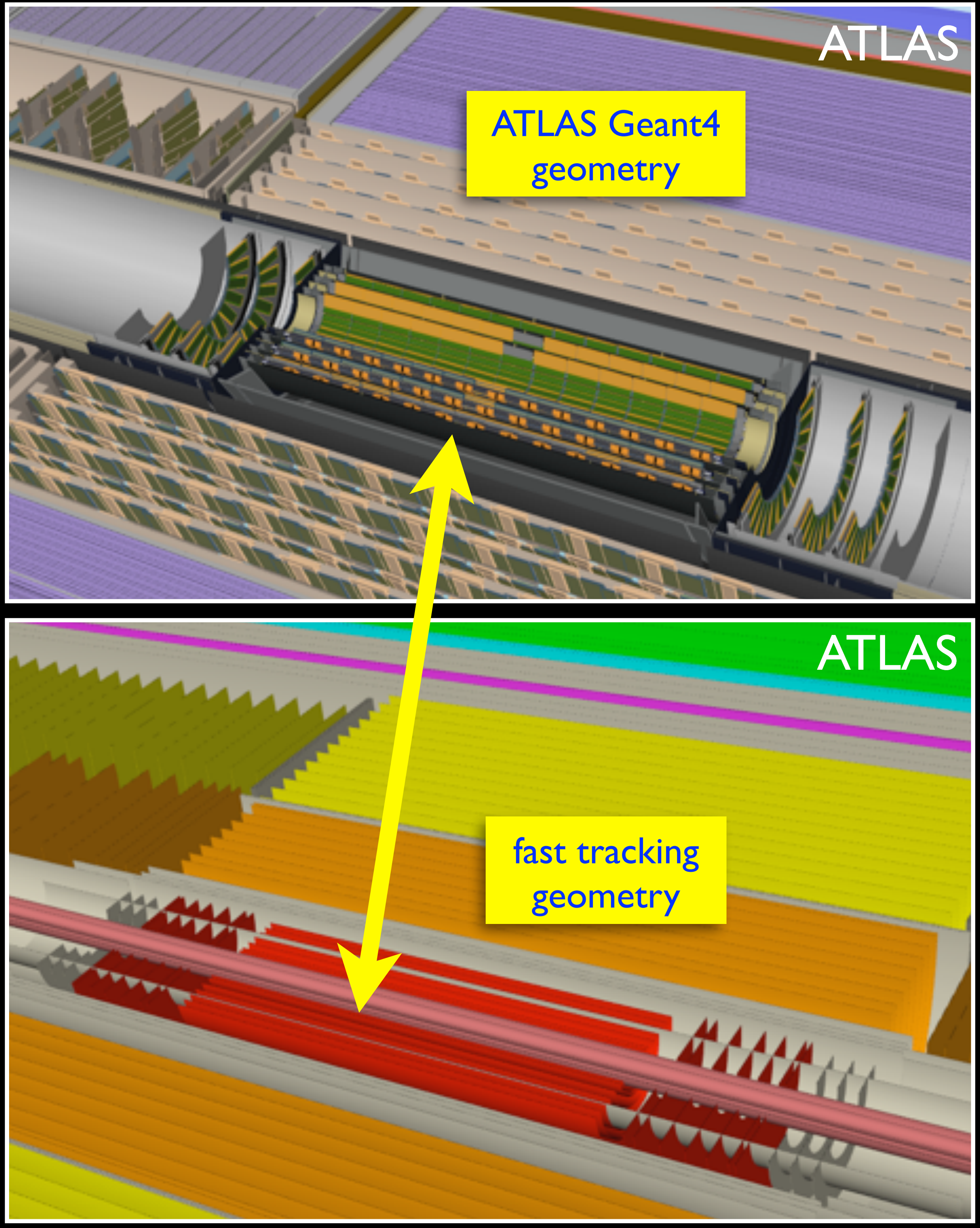
A.Salzburger

Full and Fast (Tracking) Geometries

- complex G4 geometries not optimal for reconstruction
  - ➡ simplified **tracking geometries**
  - ➡ material surfaces, field volumes
- reduced number of volumes
  - ➡ blending details of material onto simple surfaces/volumes
  - ➡ surfaces with 2D material density maps, templates per Si sensor...

	G4	tracking
ALICE	4.3 M	same *1
ATLAS	4.8 M	10.2K *2
CMS	2.7 M	3.8K *2
LHCb	18.5 M	30

*1 ALICE uses full geometry (TGeo)
*2 plus a surface per Si sensor

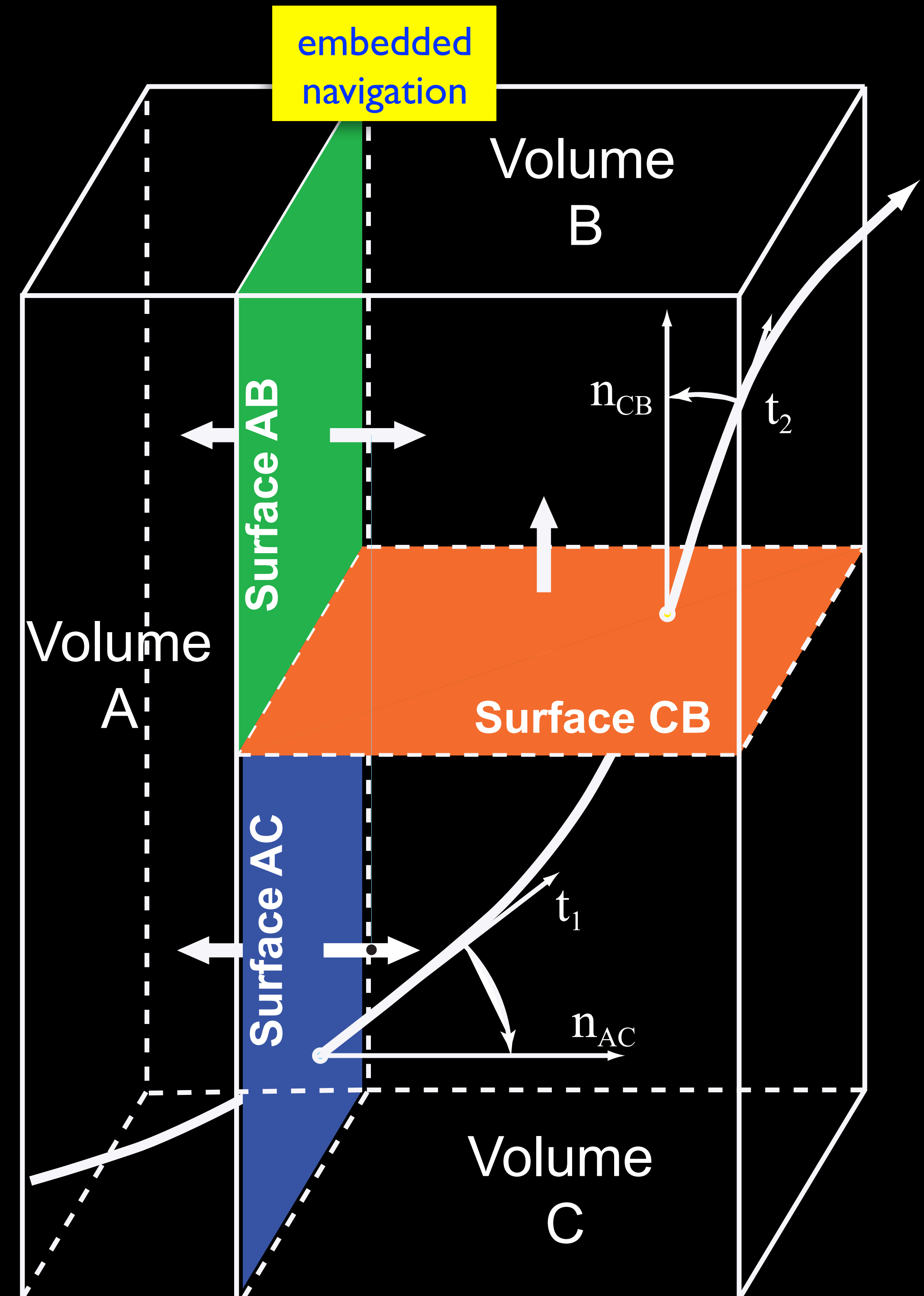


Embedded Geometry Navigation Scheme

- embedded navigation scheme in tracking geometries
  - ➡ G4 navigation uses voxelisation as generic navigation mechanism
  - ➡ **embedded navigation** for simplified models
    - used in pattern recognition, extrapolation, track fitting and fast simulation
- example: ATLAS
  - ➡ developed geometry of connected volumes
  - ➡ boundary surfaces connect neighbouring volumes to predict next step

ATLAS	G4	tracking	ratio
crossed volumes in tracker	474	95	5
time in SI2K *sec*	19.1	2.3	8.4

(neutral geantinos, no field lookups)

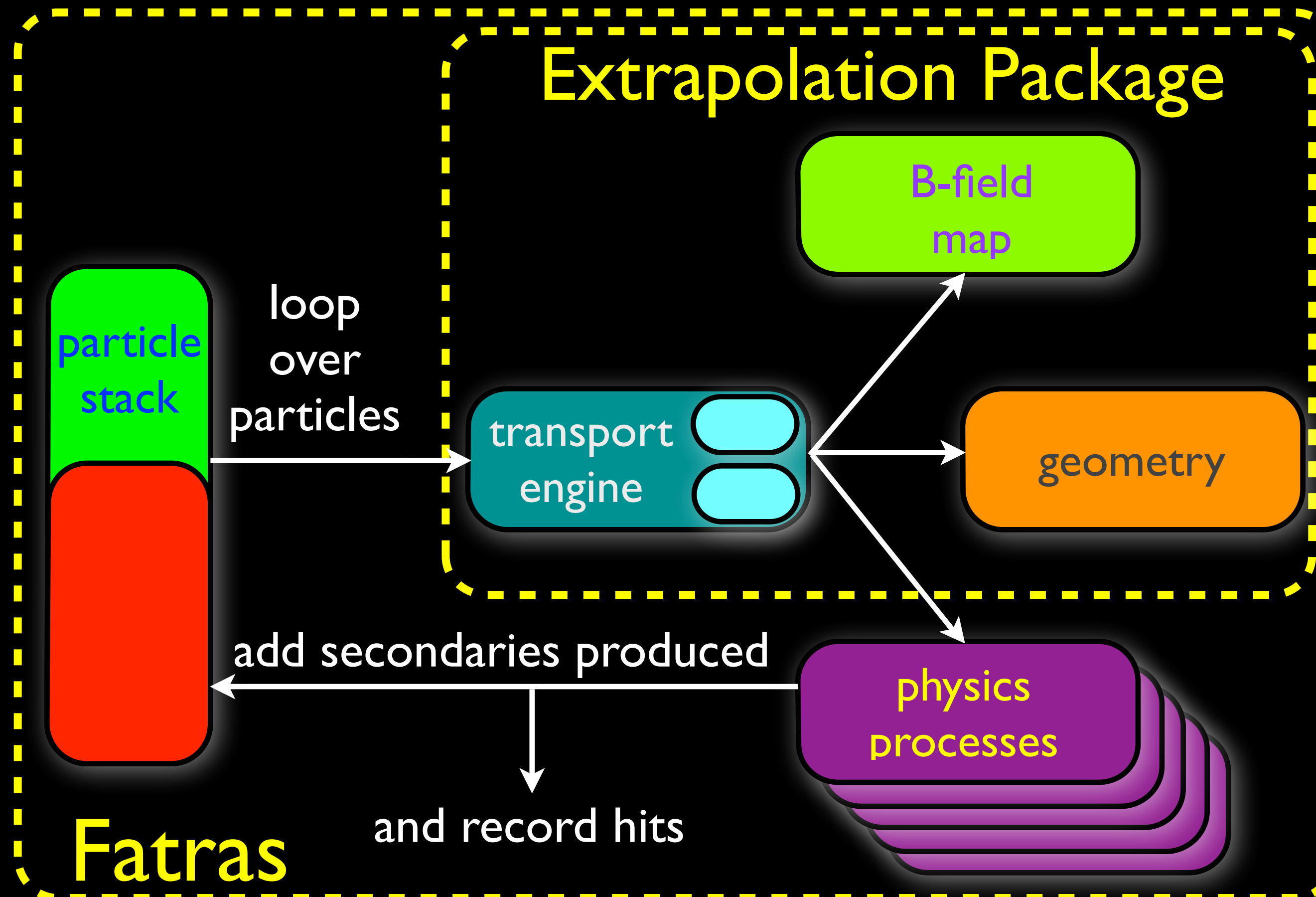


A.Salzburger

Fast Track Simulation (Fatras)

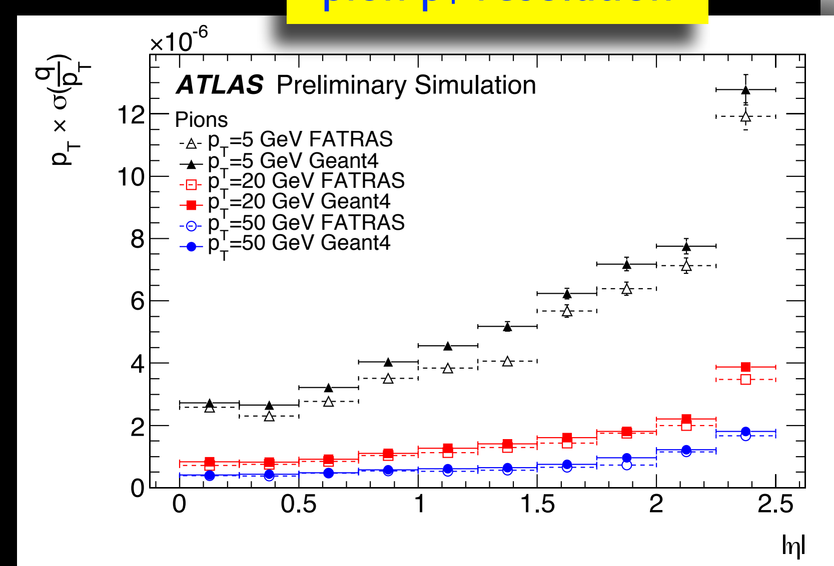
A.Salzburger

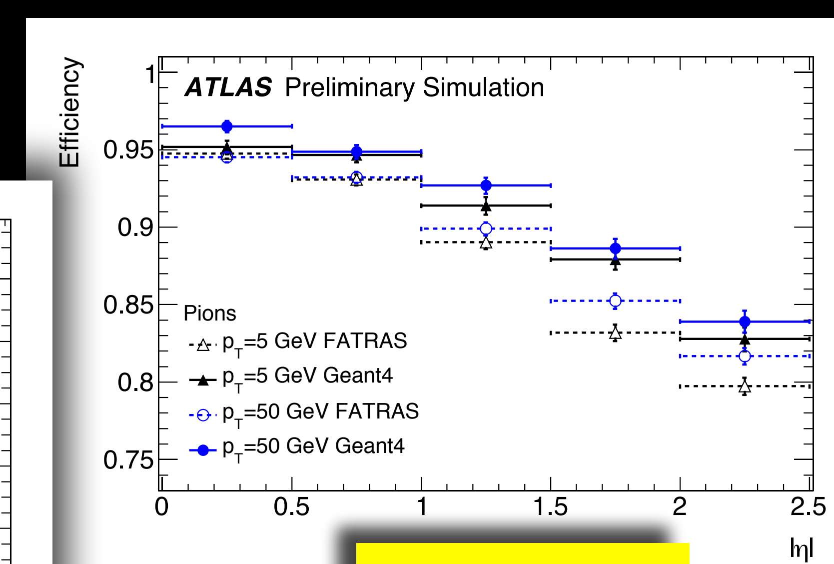
- convenient to construct fast track simulation
  - ➡ re-use **extrapolation package** to propagate each particle:
    - transport engine with navigation
    - geometry model
    - B-field map
  - ➡ add **stack** to keep track of all particles produced and stack manager
  - ➡ add set of **physics processes** describing interaction of particles with matter



ATLAS	G4	fast sim.
CPU tine	1990	7.4

pion p_T resolution





pion efficiency

Strategy of NewTracking in ATLAS

pre-precessing
- ➡ Pixel+SCT clustering
- ➡ TRT drift circle formation
- ➡ space points formation

combinatorial track finder
- ➡ iterative :
  1. Pixel seeds
  2. Pixel+SCT seeds
  3. SCT seeds
- ➡ restricted to roads
- ➡ bookkeeping to avoid duplicate candidates

ambiguity solution
- ➡ precise least square fit with full geometry
- ➡ selection of best silicon tracks using:
  1. hit content, holes
  2. number of shared hits
  3. fit quality...

extension into TRT
- ➡ progressive finder
- ➡ refit of track and selection

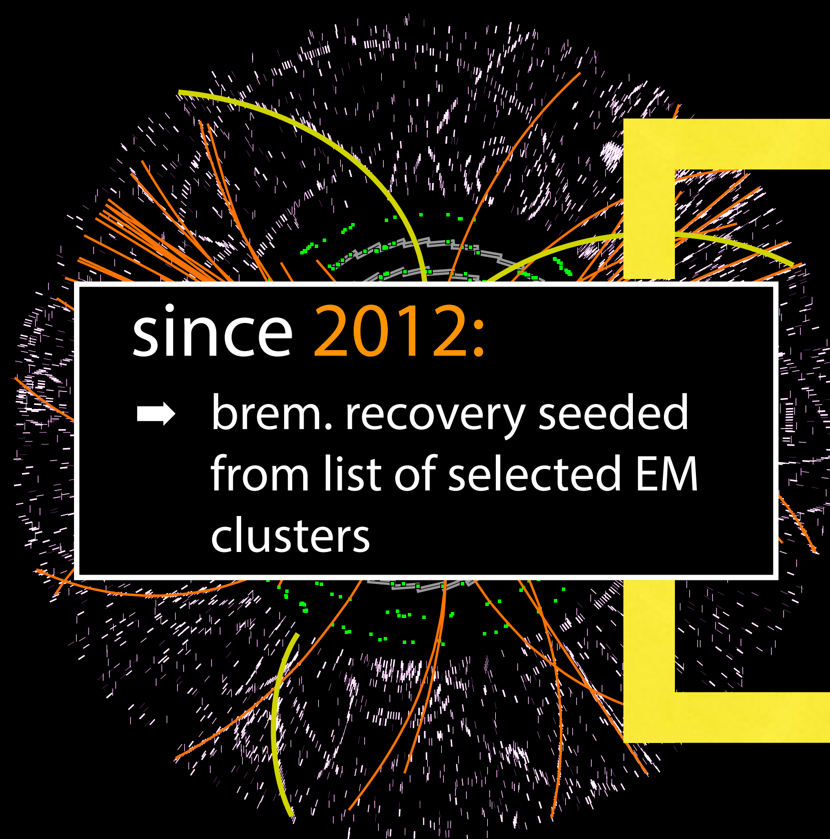
TRT segment finder
- ➡ on remaining drift circles
- ➡ uses Hough transform

TRT seeded finder
- ➡ from TRT into SCT+Pixels
- ➡ combinatorial finder

ambiguity solution
- ➡ precise fit and selection
- ➡ TRT seeded tracks

standalone TRT
- ➡ unused TRT segments

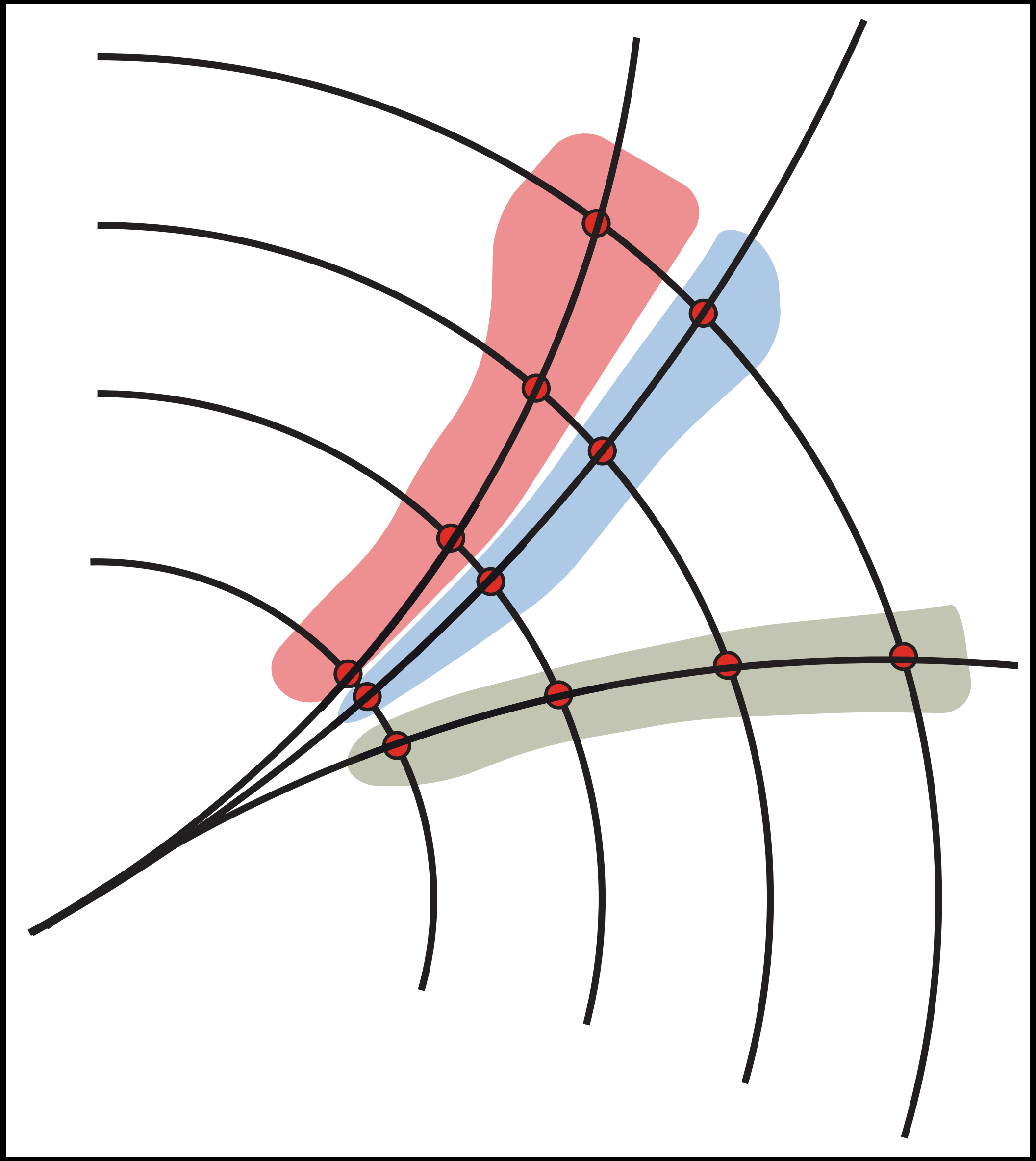
vertexing
- ➡ primary vertexing
- ➡ conversion and V0 search



since 2012:
- ➡ brem. recovery seeded from list of selected EM clusters

Iterative Seeding Strategy

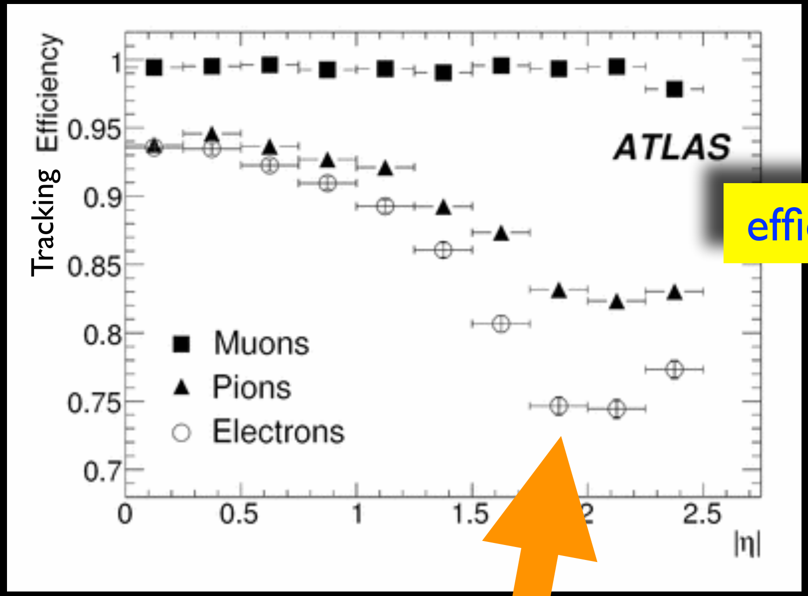
- the track finding algorithm
  - ➡ find **seed** from combination of 3 hits
    - search using hough transform
  - ➡ build **road** along the likely trajectory
  - ➡ run **combinatorial Kalman Filter** for a seed
    - full **exploration** of all possible candidates
    - update trajectory with hits at each layer
    - take material effects into account



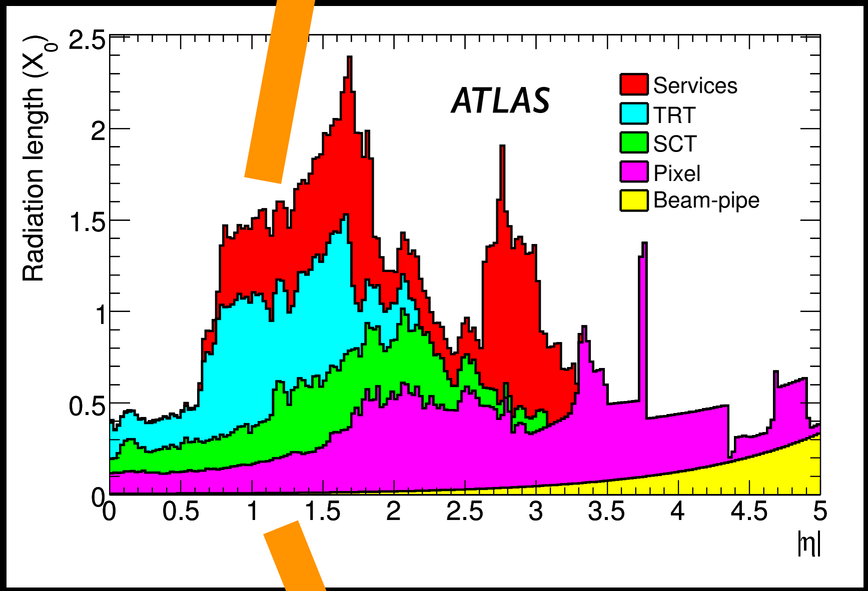
- iterative seeding approach (Run-1)
  - ➡ seeds are worked on in an ordered list
    - start with 3 Pixels, 2 Pixel+Strip, 3 Strips
  - ➡ bookkeeping layer:
    - hits from good candidates removed
    - build next seed ONLY from left over hits
  - ➡ sequential seed finding to avoid combinatorial explosion (see later w.r.t. parallel tracking)
    - unlike in the animation, tracks are found for one-after-the-other
    - hence, the ordering matters !!! (especially sorting in p_T bins)

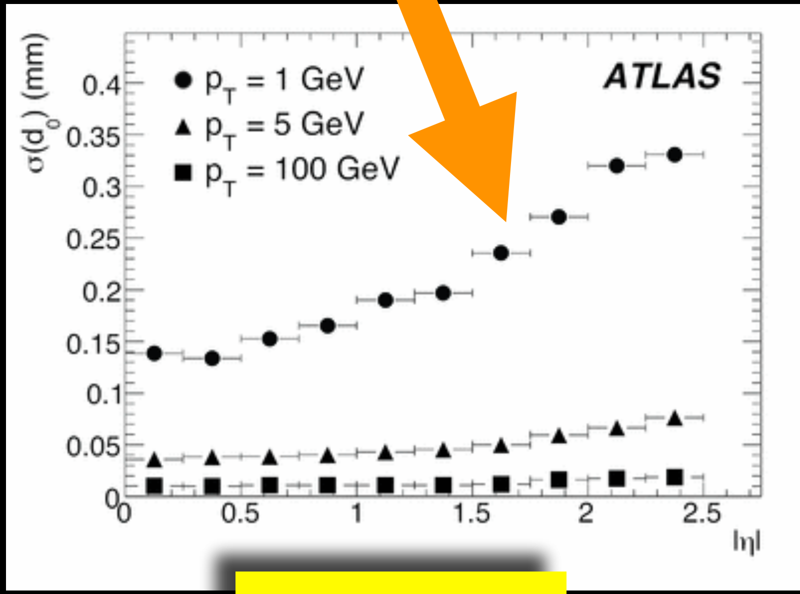
Expected Performance

- excellent preparation before startup
  - ➡ more than 10 years of simulation and test beam
  - ➡ cosmics data taking in 2008 and 2009
- detailed simulation studies
  - ➡ document expected performance in TDRs
  - ➡ few of the known critical items:
    - material effects limit efficiency and resolution at low p_t
    - good (local) alignment for *b*-tagging
    - momentum scale and alignment "weak modes"
  - ➡ focus for commissioning of tracking and vertexing



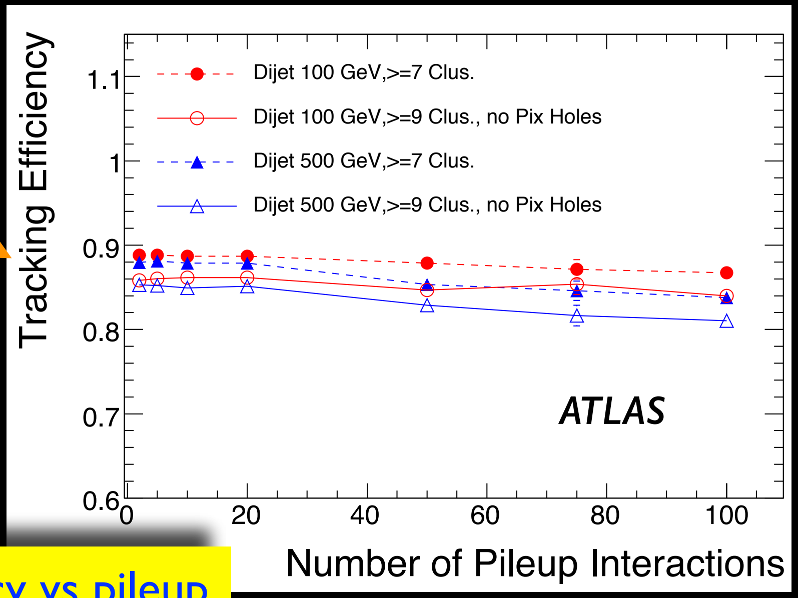
efficiency





d0 resolution

performance with event pileup



efficiency vs pileup

Weighing Detectors during Construction

- huge effort in experiments
  - ➡ put each individual detector part on balance and compare with model
  - ➡ CMS and ATLAS measured weight of their tracker and its components
  - ➡ correct the geometry implementation in simulation and reconstruction

example: ATLAS TRT measured before and after insertion of the SCT

CMS	estimated from measurements	simulation
active Pixels	2598 g	2455 g
full detector	6350 kg	6173 kg

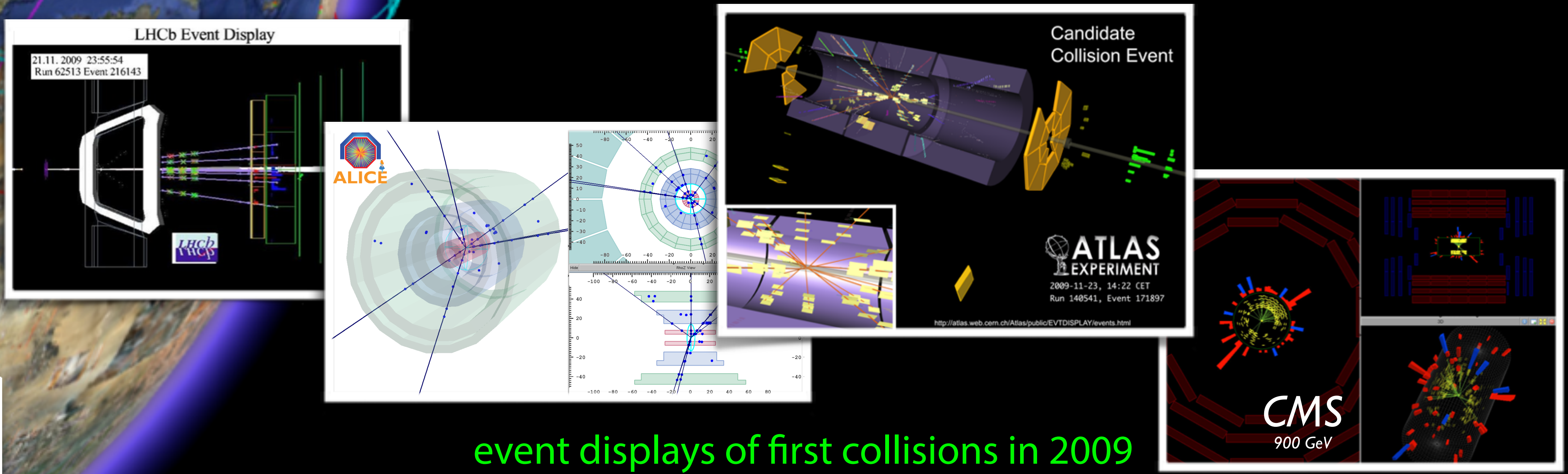
Preliminary

ATLAS	estimated from measurements	simulation
Pixel package	201 kg	197 kg
SCT detector	672 ±15 kg	672 kg
TRT detector	2961 ±14 kg	2962 kg

evolution of X_0 in tracker

Date	ATLAS $\eta \approx 0$	ATLAS $\eta \approx 1.7$	CMS $\eta \approx 0$	CMS $\eta \approx 1.7$
1994 (Technical Proposals)	0.20	0.70	0.15	0.60
1997 (Technical Design Reports)	0.25	1.50	0.25	0.85
2006 (End of construction)	0.35	1.35	0.35	1.50

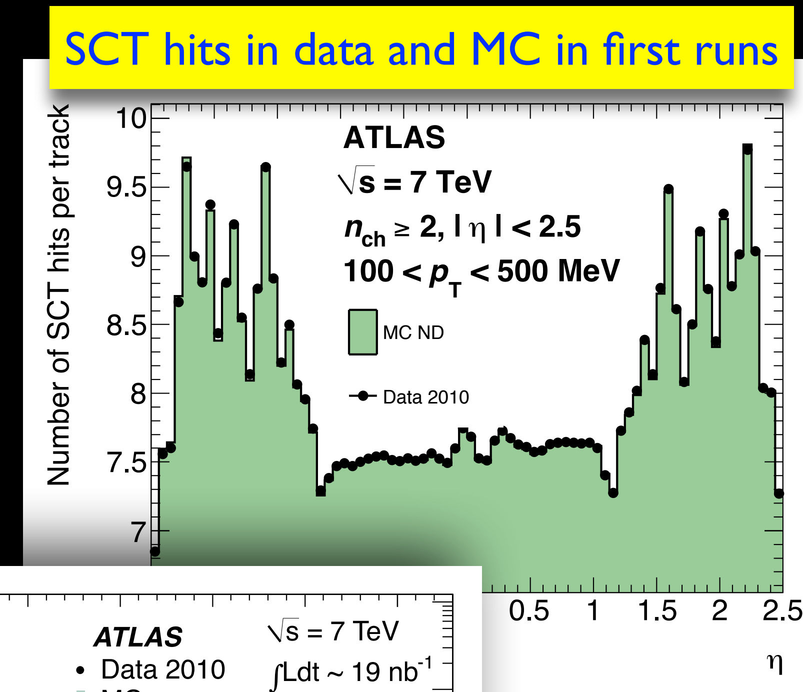
Early Physics and the Experience from Run-1

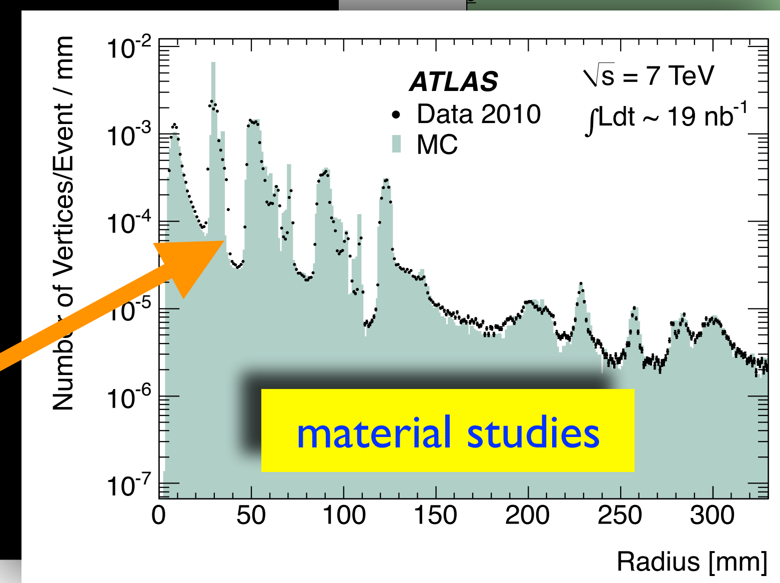


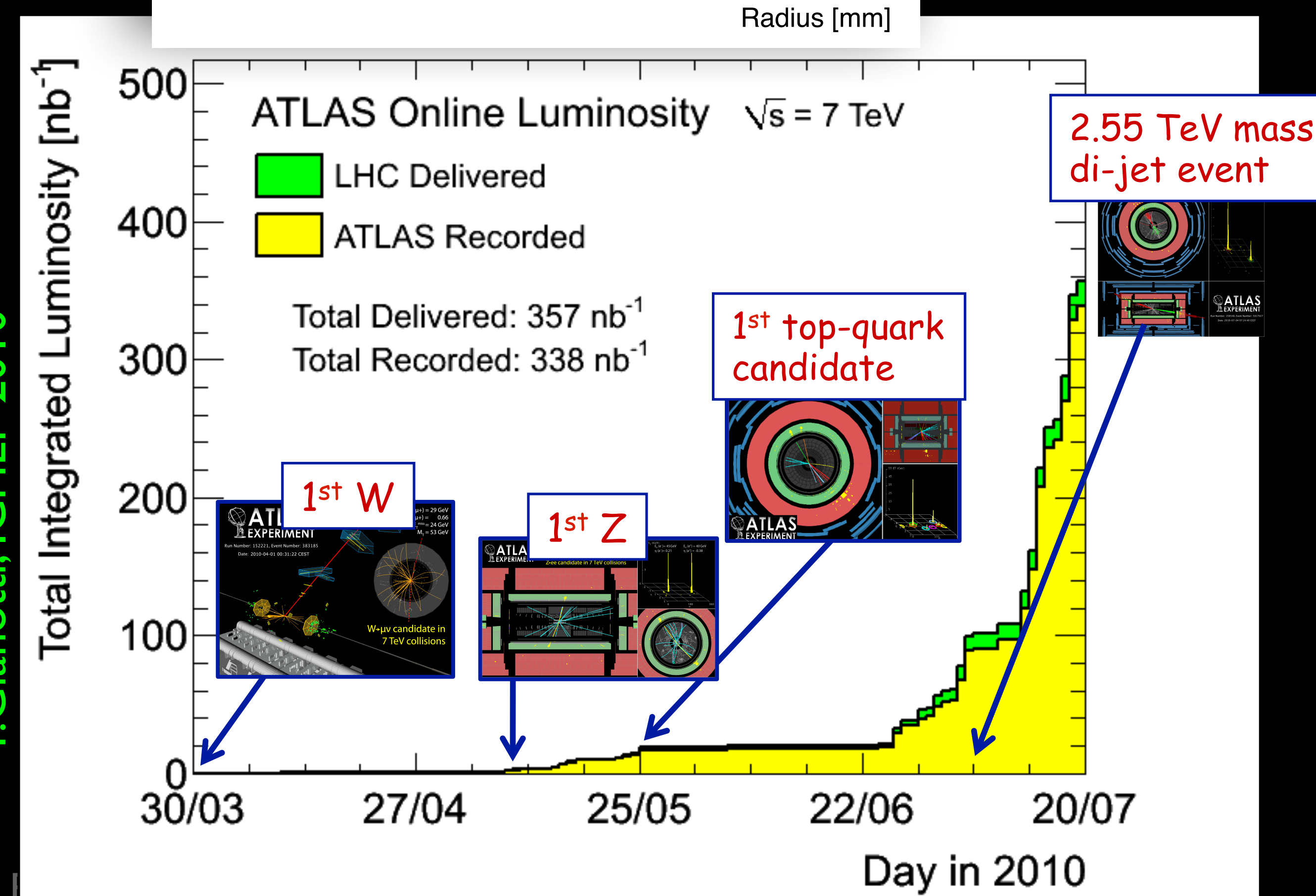
event displays of first collisions in 2009

First Data to Physics Results

- a success story all along...
  - ➡ detector, DAQ and trigger worked !
  - ➡ excellent quality of first data
    - fast convergence of calibration and alignment procedures
    - much smoother than many expected
  - ➡ striking level of modelling by simulation
    - thanks to careful preparation work, e.g. excellent model of tracker material
    - helped a lot the fast production of physics results
- with luminosity increasing over the year 2010
  - ➡ quality of data approaching design levels with series of reprocessings
  - ➡ "re-discovered" the standard model particles one-by-one



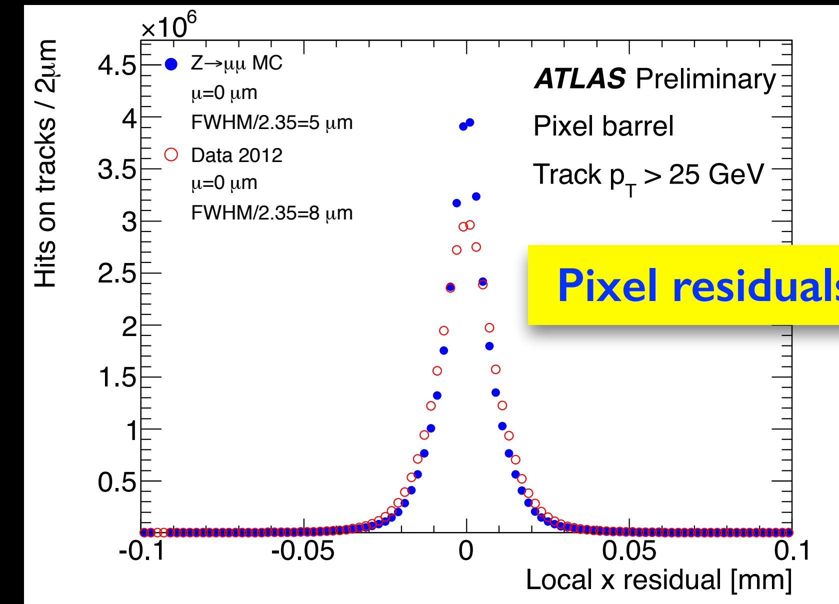


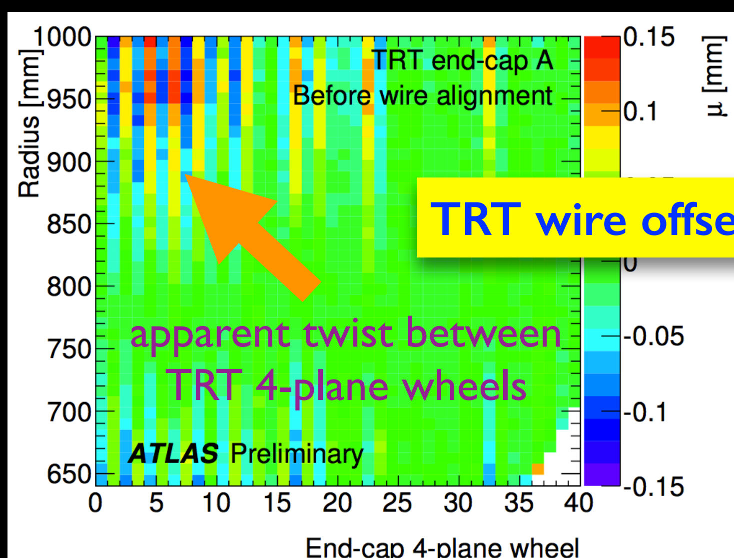


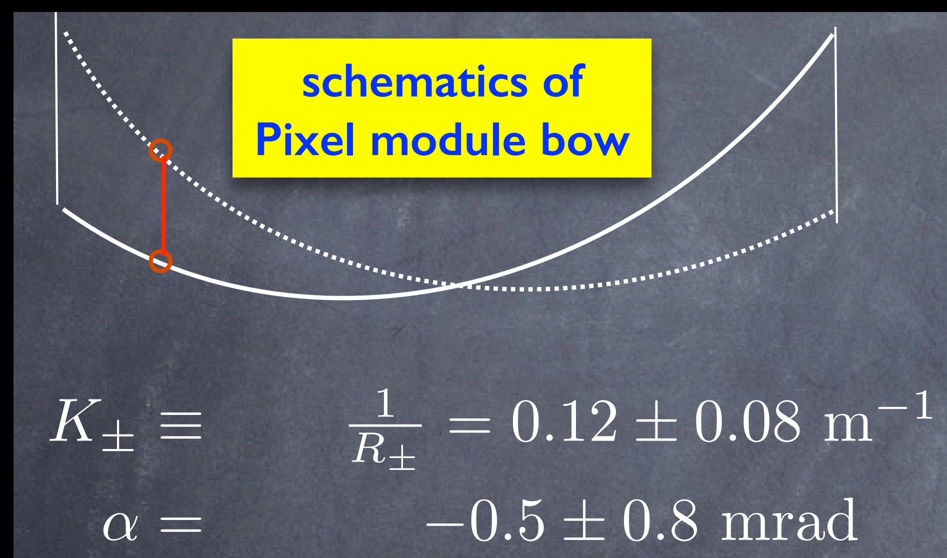
F.Gianotti, ICHEP 2010

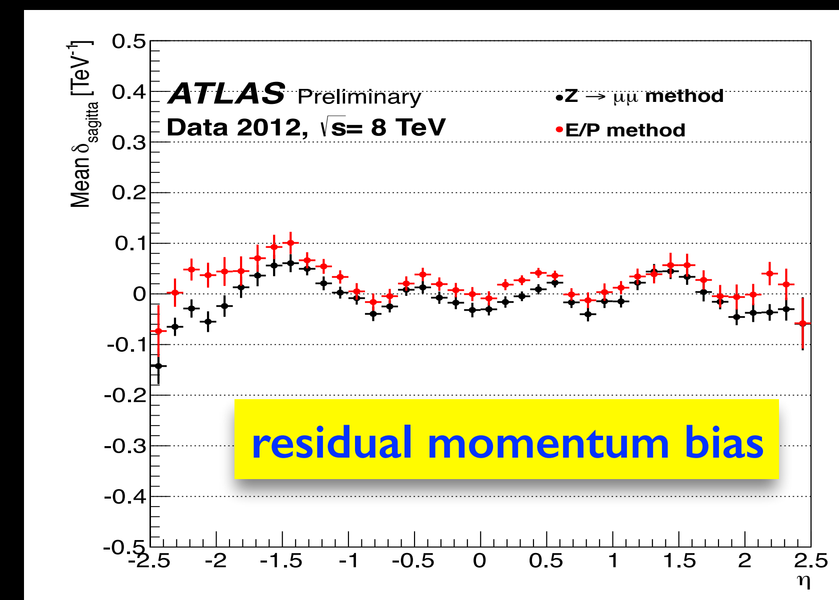
Run 1 Tracking Performance

- in the first year we achieve excellent control on alignment
  - ➡ local alignment, e.g. TRT wire plane offsets or Pixel bow

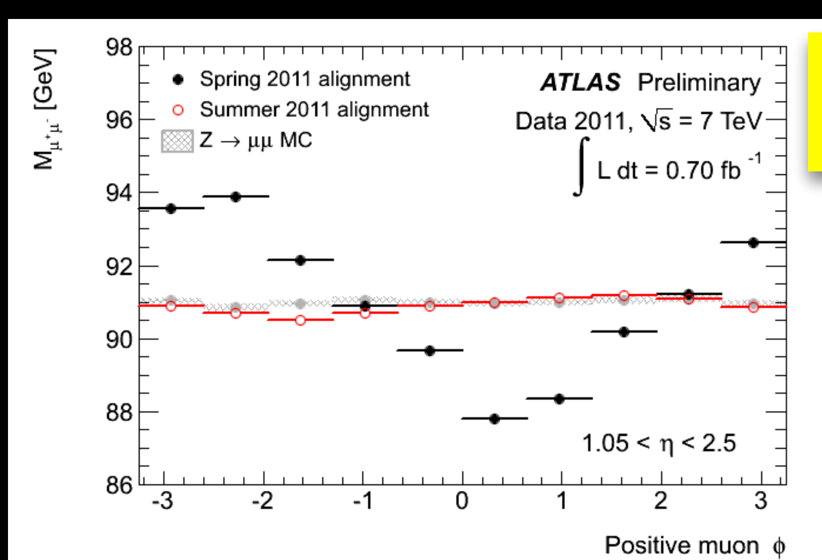




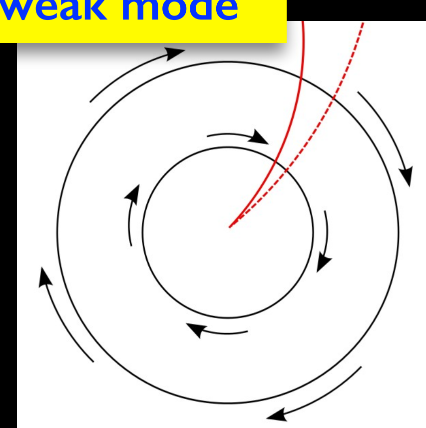


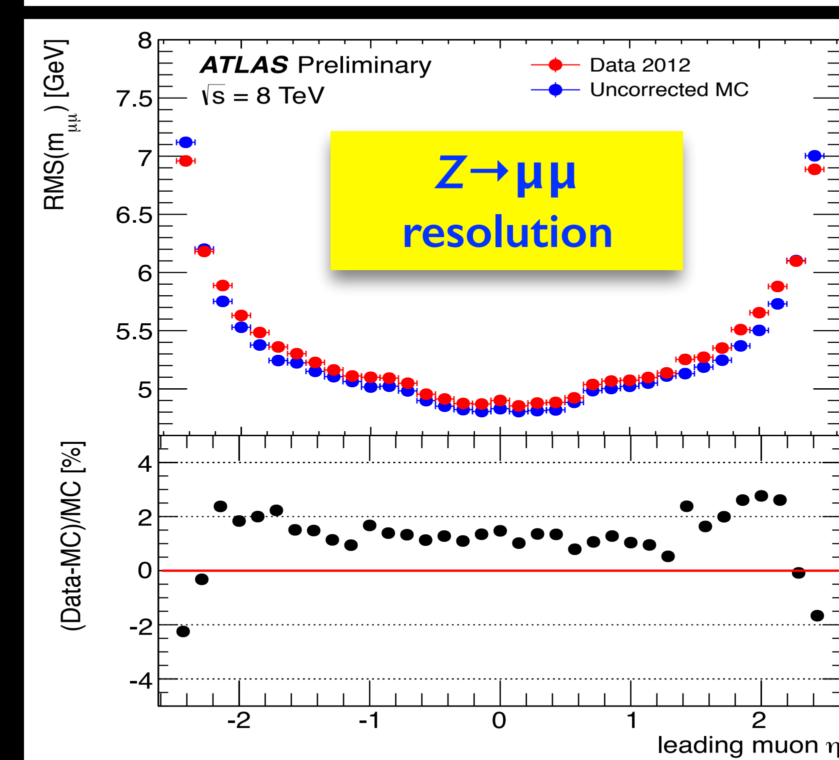


  - ➡ global weak mode and time variations corrections



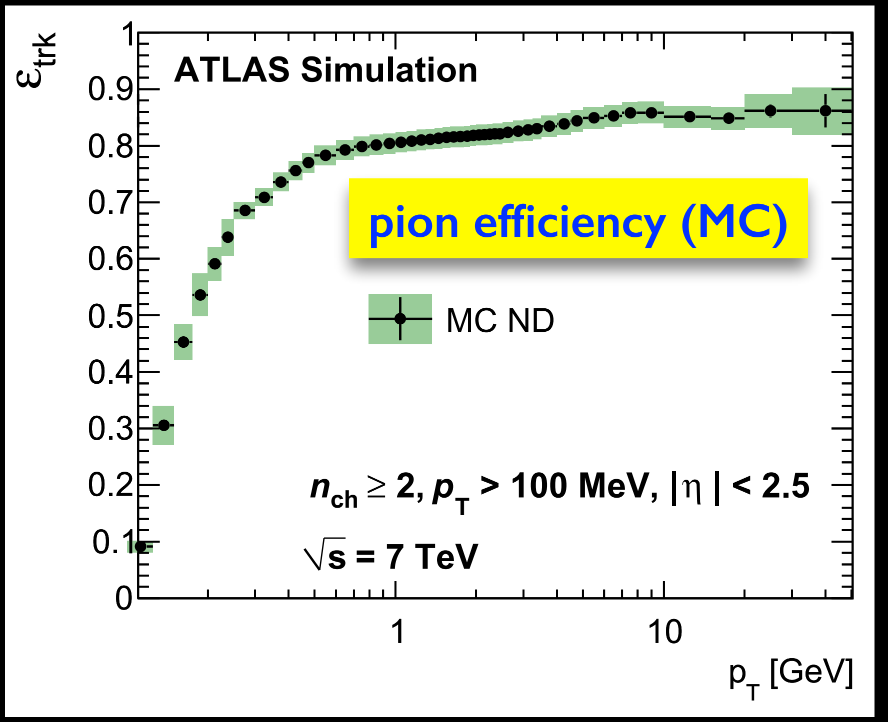
evidence for weak mode

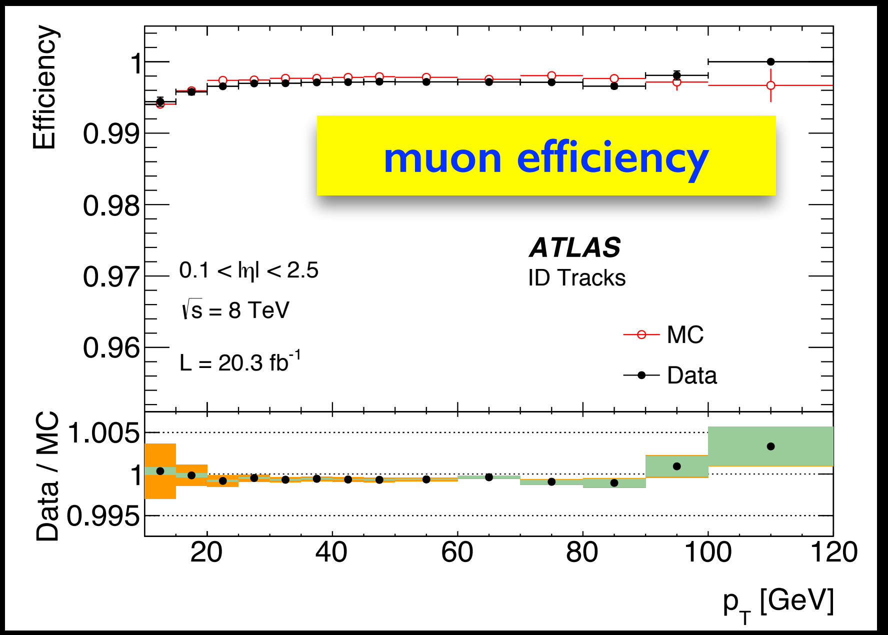


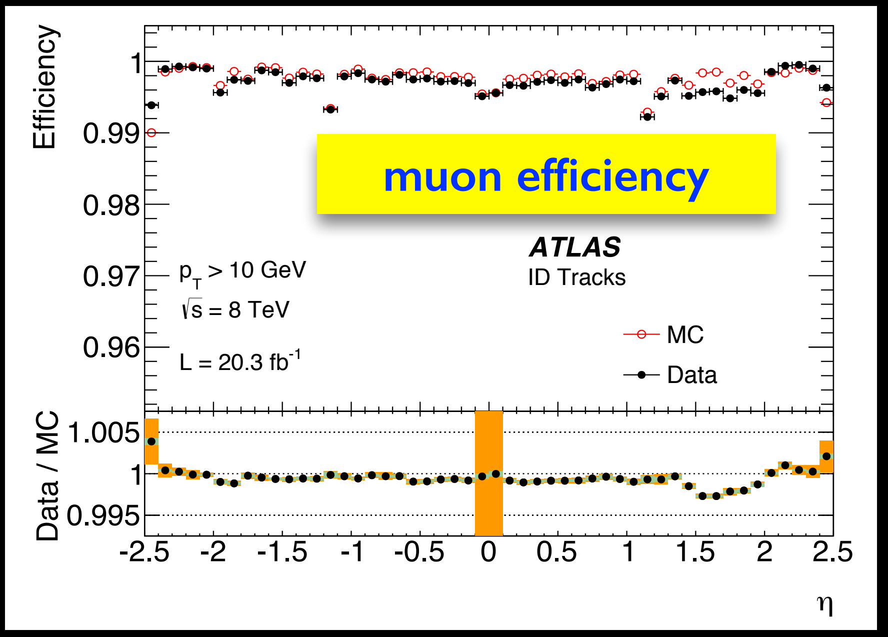


Run 1 Tracking Performance

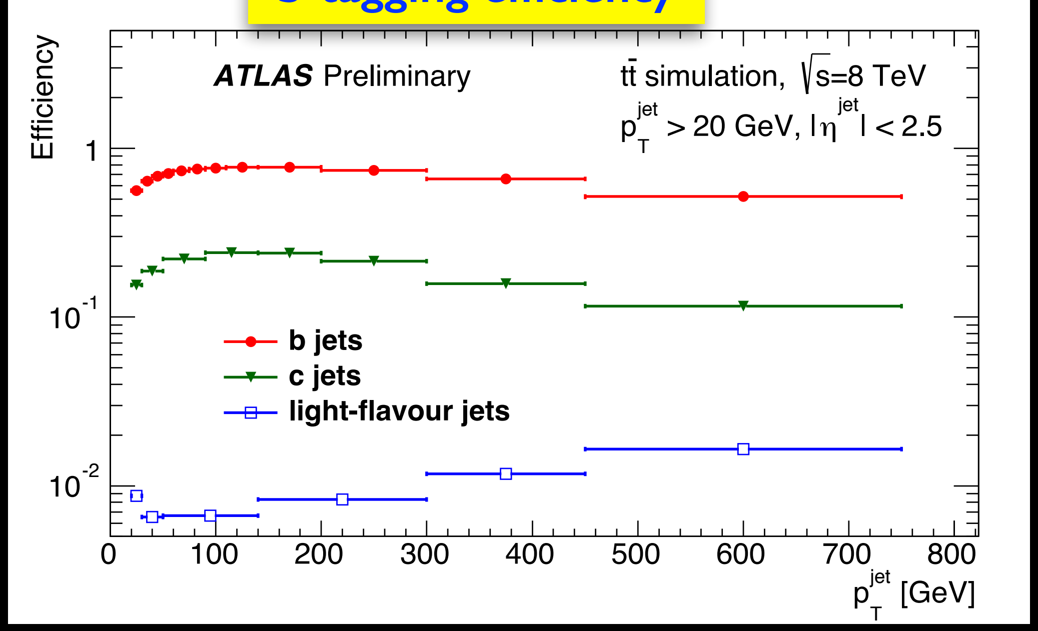
- tracking efficiency difficult to measure for hadrons
  - ➡ efficiency for entirely limited by material interactions
- muons are almost ideal MIPs
  - ➡ *Z*, *J/ψ* and *Υ* decays allow us to accurately measure the tracking efficiency
  - ➡ measured efficiency >99.5% for all Run-1 conditions
- excellent *b*-tagging performance
  - ➡ working point: 70% *b*-efficiency for light rejection >100



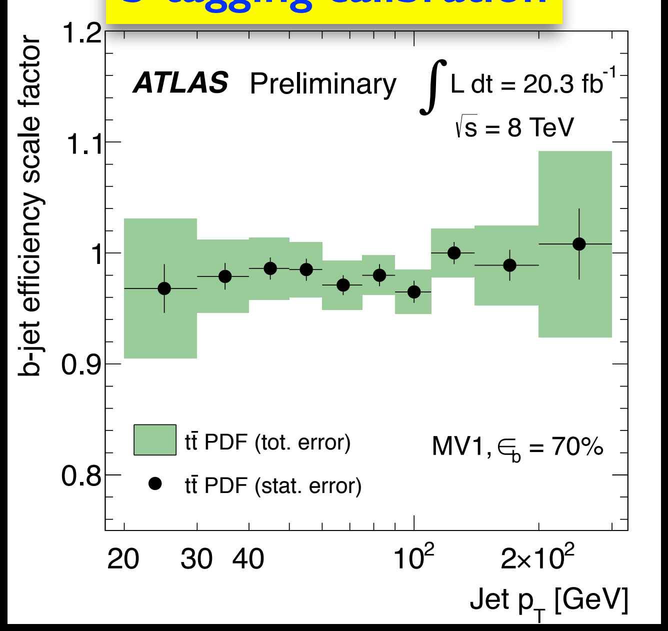


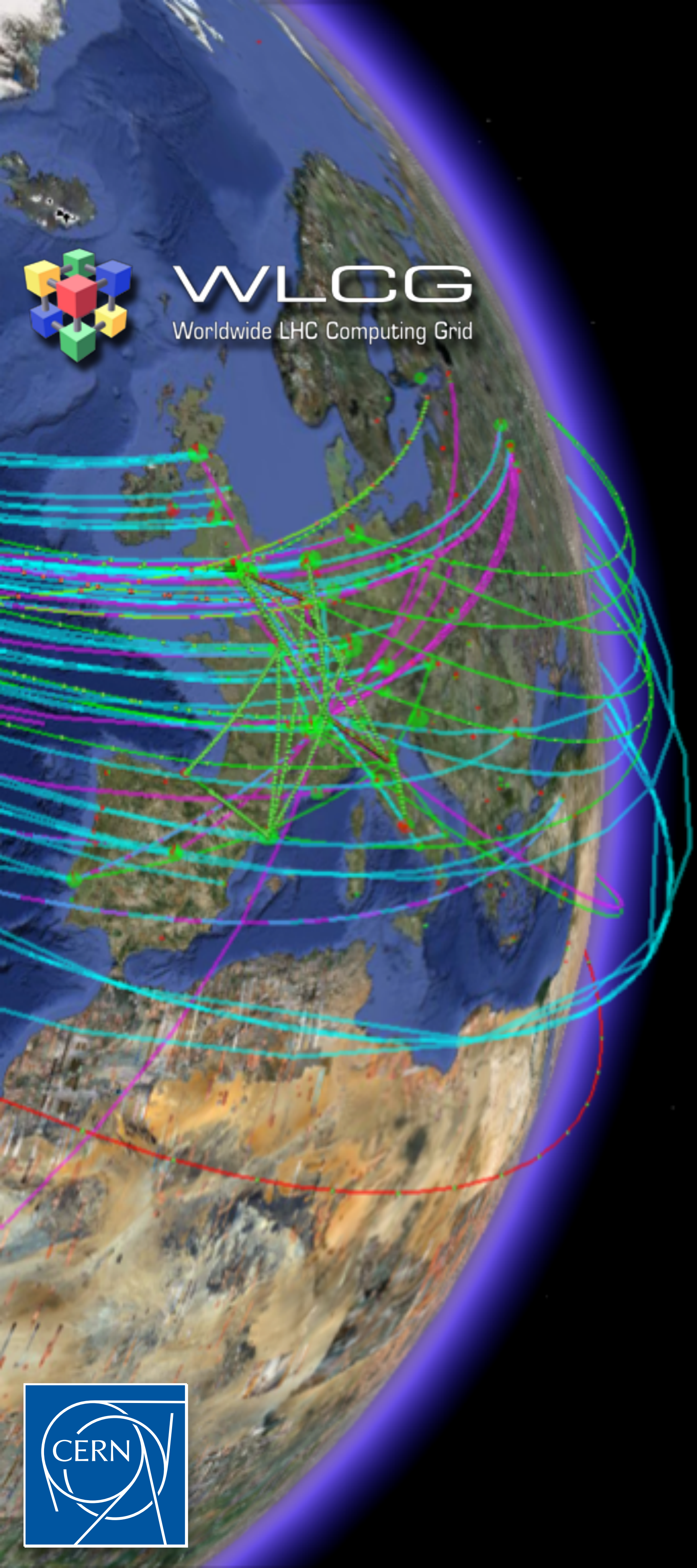


b-tagging efficiency



b-tagging calibration



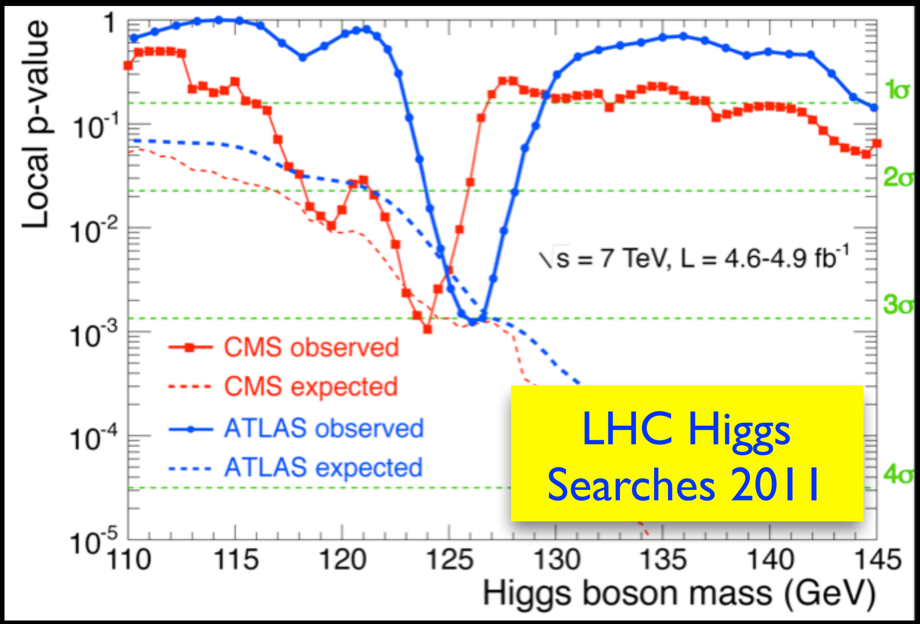


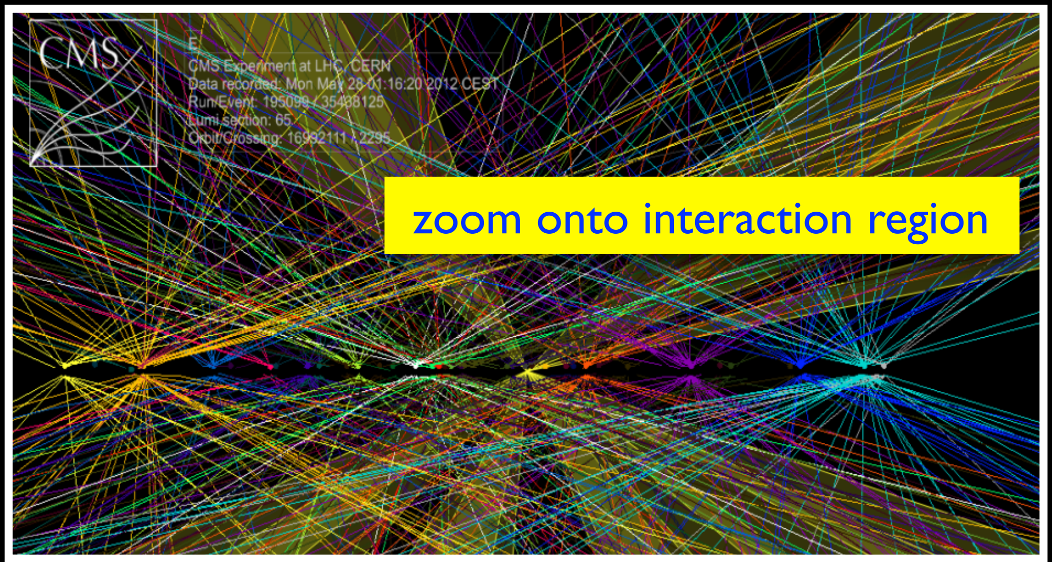
The Higgs Discovery: the Role of the Offline Software

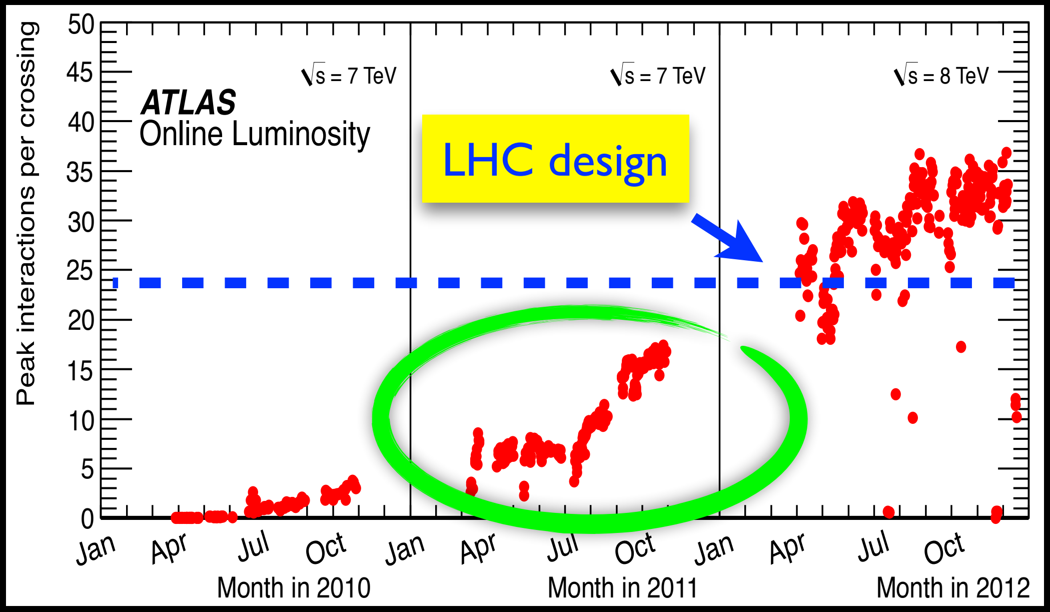
Situation in 2011

- Higks searches in 2011 data
  - ➡ both experiments saw "hints" for a light Higgs
    - about ~3σ each, ignoring "look elsewhere effect"
    - indications as well in TEVATRON data
  - ➡ low mass region at LHC
    - many decay modes accessible (γγ,ZZ,WW,ττ,bb)
    - $\gamma\gamma$ and ZZ yield excellent mass resolution (~1%)
  - ➡ detector performance crucial to all analyses (!)
- rapid increase in luminosity
  - ➡ pileup approaching design levels in 2011
    - mainly because of 50 *ns* operation
    - expectation was to exceed design level in 2012
  - ➡ concerns about pileup robustness and performance of object reconstruction
    - experiments did intensive software development in preparation for 2012 data taking

M.E., for illustration





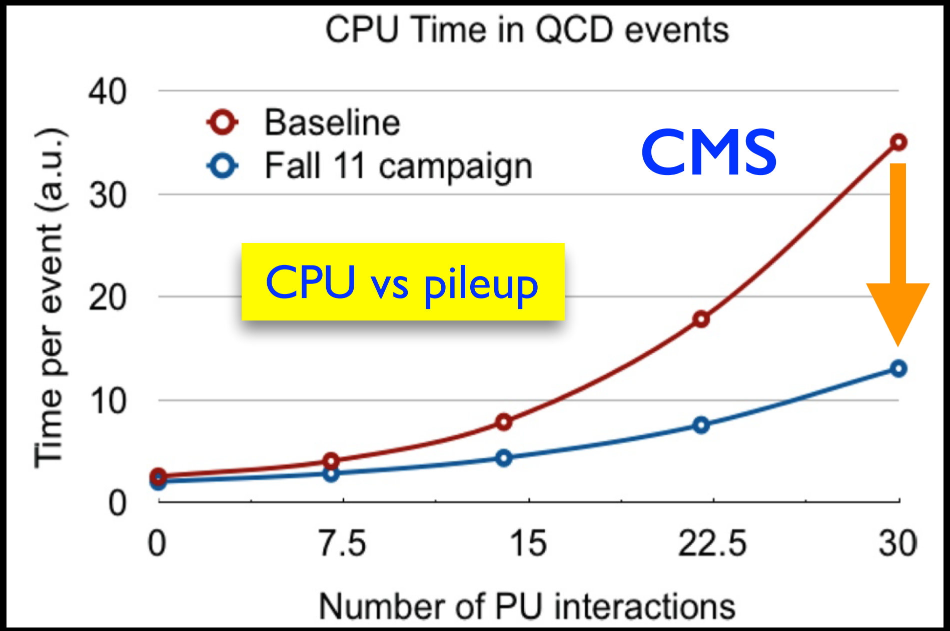


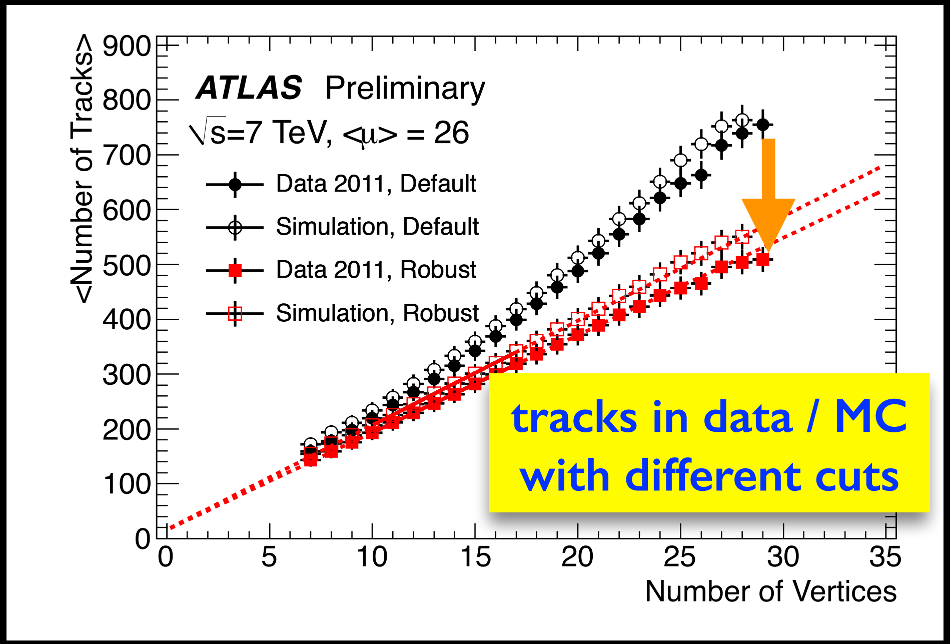
Updates to Tracking

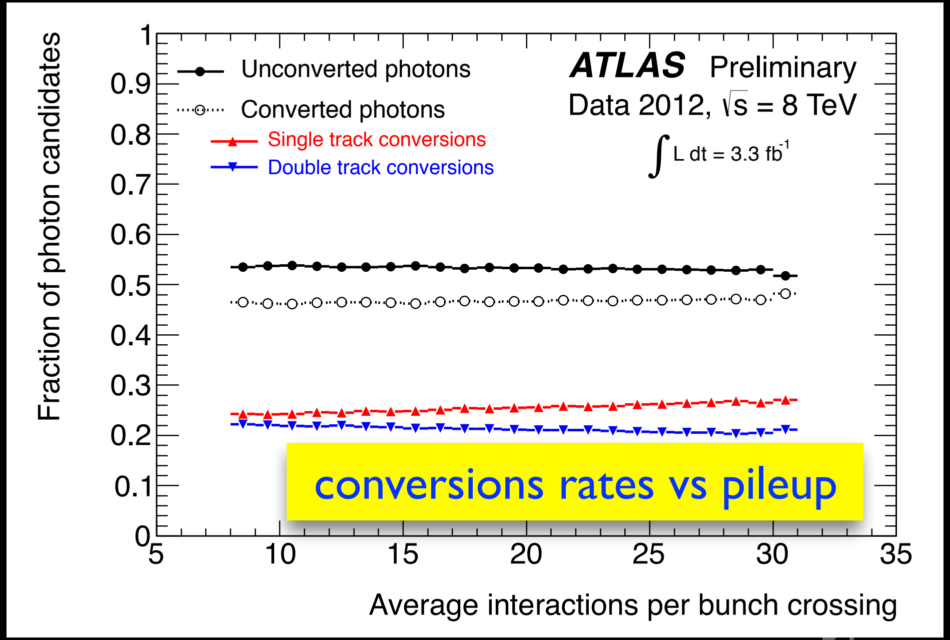
- CPU scales non-linear with pileup
  - ➡ combinatorial explosion
    - CMS ~50% in tracking

      (e/γ dominated by special tracking too)
    - ATLAS ~70% in tracking
  - ➡ e.g. CMS gained factor 2-3 in CPU
    - optimisation of pattern for 30 pileup
    - as well technical optimisation (memory)
    - similar optimisation done in ATLAS

- pileup robustness and performance
  - ➡ improve track selections to control fakes
  - ➡ more robust tracking cuts for object reconstruction
    - e.g., tracking for conversions in ATLAS optimised to improve pileup stability ($H \rightarrow \gamma\gamma$)

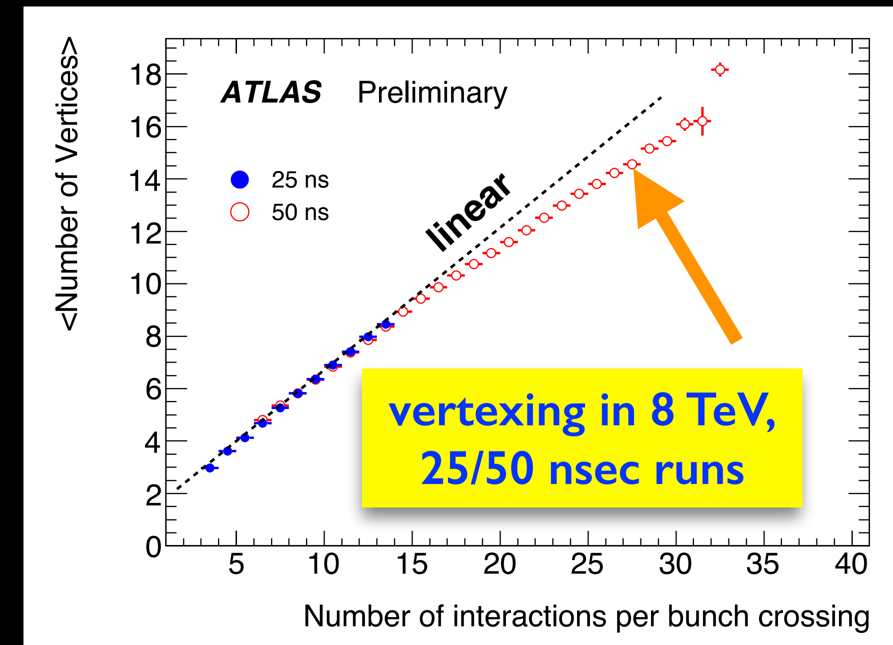


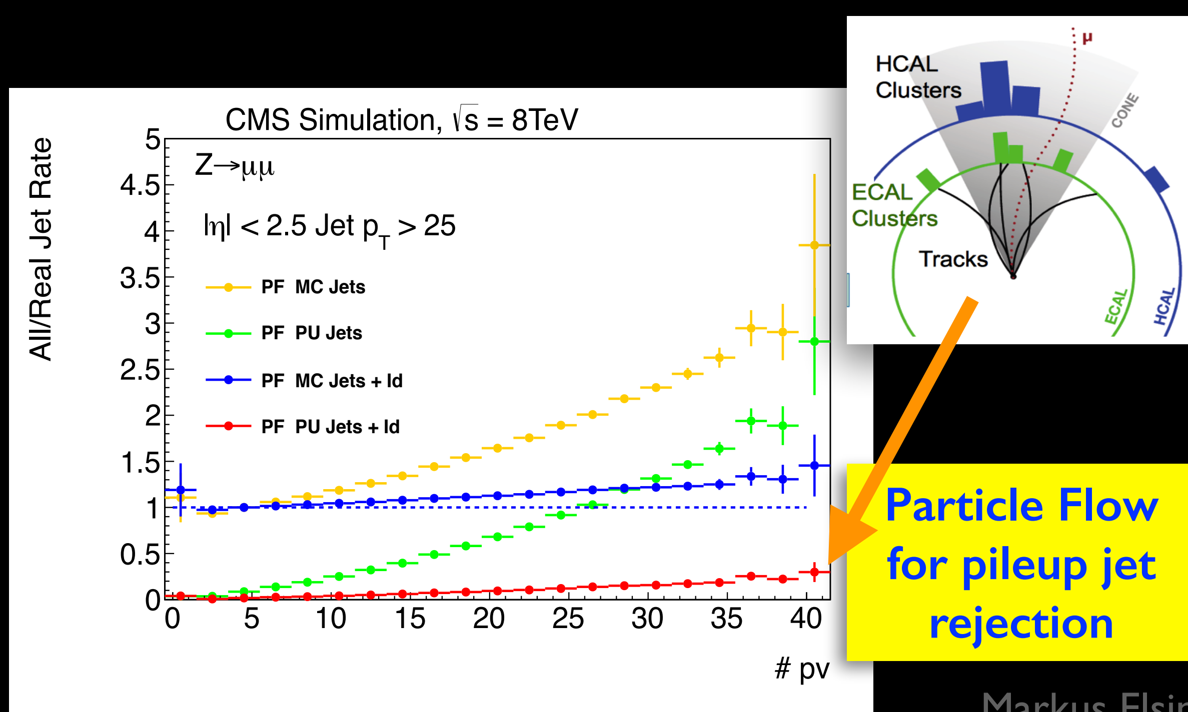


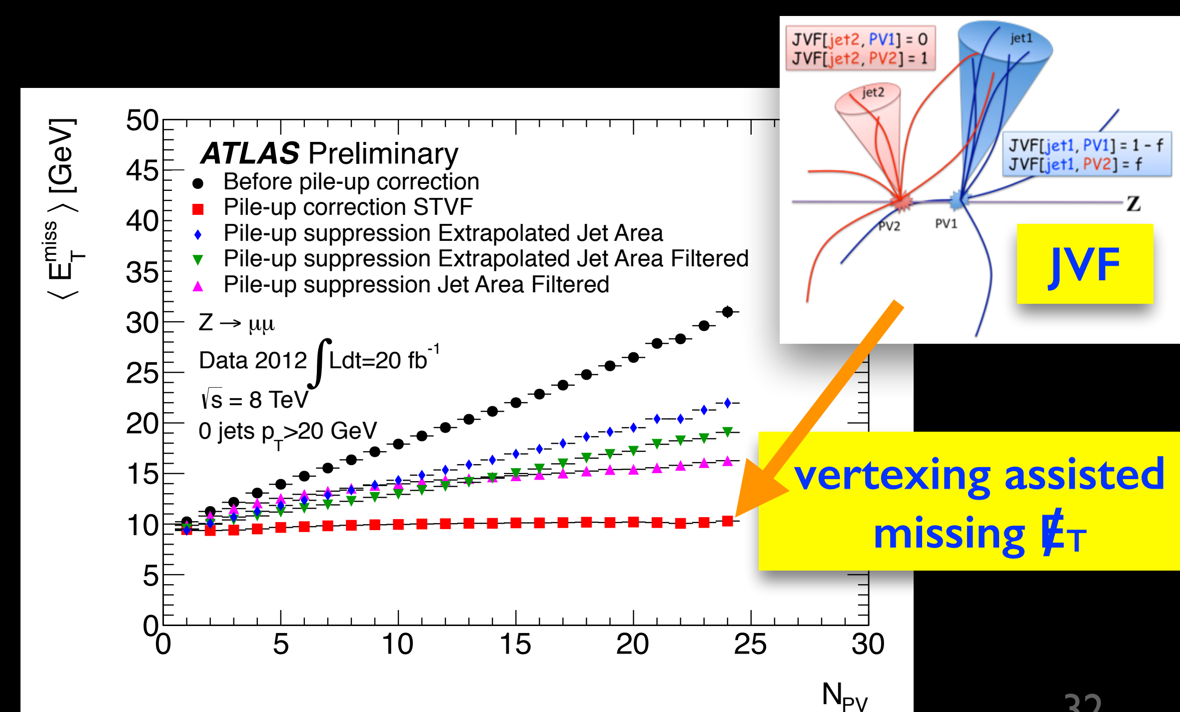


Updates to Vertexing and Jet/MET

- primary vertexing
  - ➡ more robust selections and algorithm updates
  - ➡ still visible effects of vertex merging at high μ
  - ➡ Σp_T based vertex tagging less and less optimal (see MC)
- tracking as a tool for pileup control
  - ➡ combining calorimeter and tracking information
    - CMS jets, $\not{E}_T$ and τ based on Particle Flow
    - ATLAS used vertexing for pileup jet tagging (JVF and variants of it)
  - ➡ such techniques will be even more important in the future



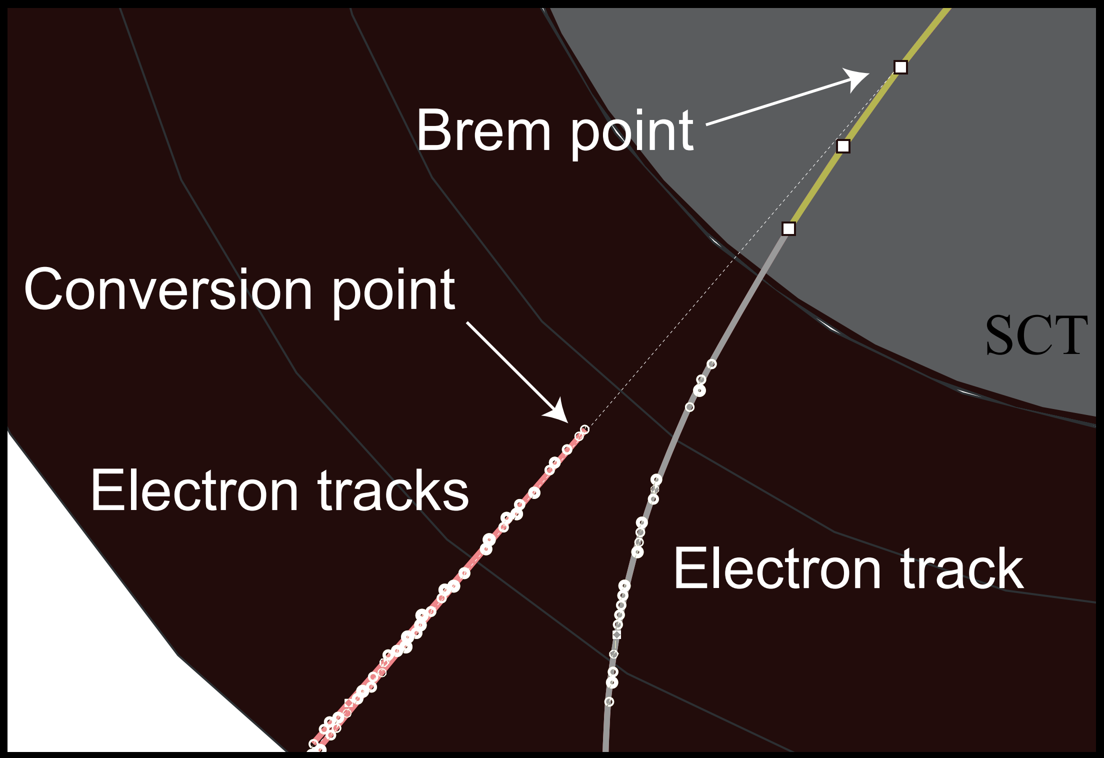




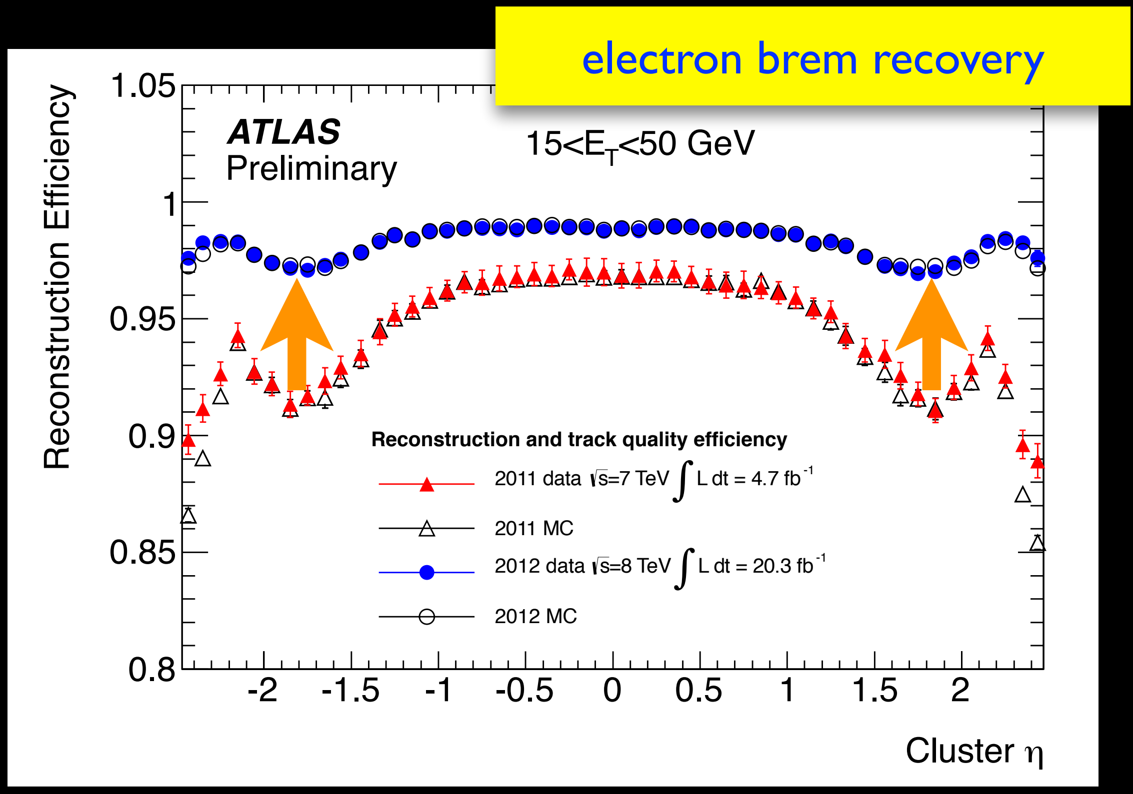
Tracking with Electron Brem. Recovery

- strategy for brem. recovery
  - ➡ restrict recovery to regions pointing to electromagnetic clusters (RoI)
  - ➡ pattern: allow for large energy loss in combinatorial Kalman filter
    - adjust noise term for electrons
  - ➡ global-χ^2 fitter allows for brem. point
  - ➡ adapt ambiguity processing (etc.) to ensure e.g. b-tagging is not affected
  - ➡ use full fledged Gaussian-Sum Filter in electron identification code

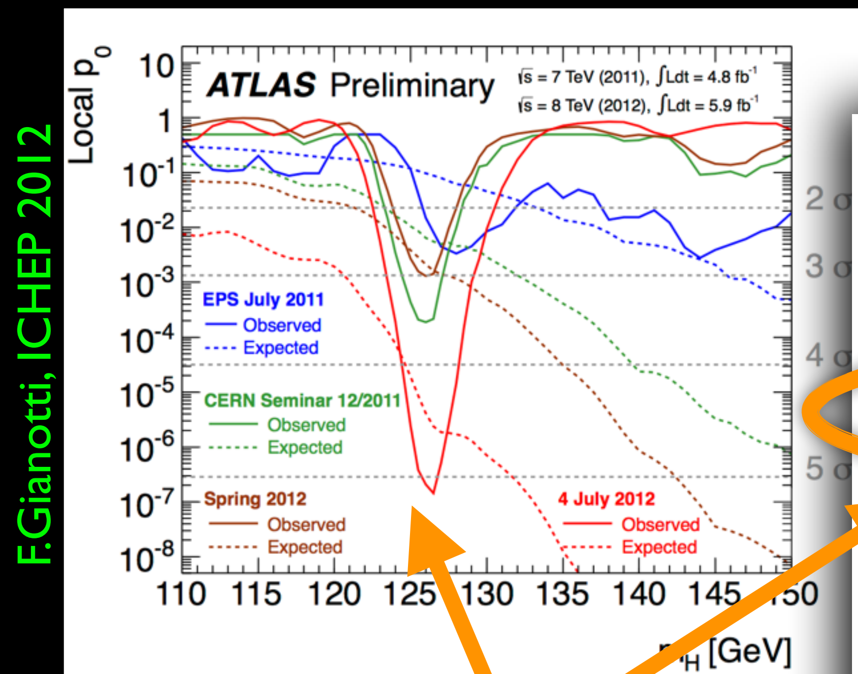
- deployed before 2012
  - ➡ improvements especially at low p_T (< 15 *GeV*)
    - limiting factor for *H→ZZ*→4e*
  - ➡ significant efficiency gain for Higgs discovery
    - similar techniques used in CMS



ATLAS-CONF-2013-1287



CERN Seminar July 4th, 2012: the Higgs



We present updated results on SM Higgs searches based on the data recorded in 2011 at $\sqrt{s}$=7 TeV (~4.9 fb^{-1}) and 2012 at $\sqrt{s}$=8 TeV (~5.9 fb^{-1})

Results are preliminary:
- ❑ 2012 data recorded until 2 weeks ago
- ❑ new, improved analyses deployed for the first time

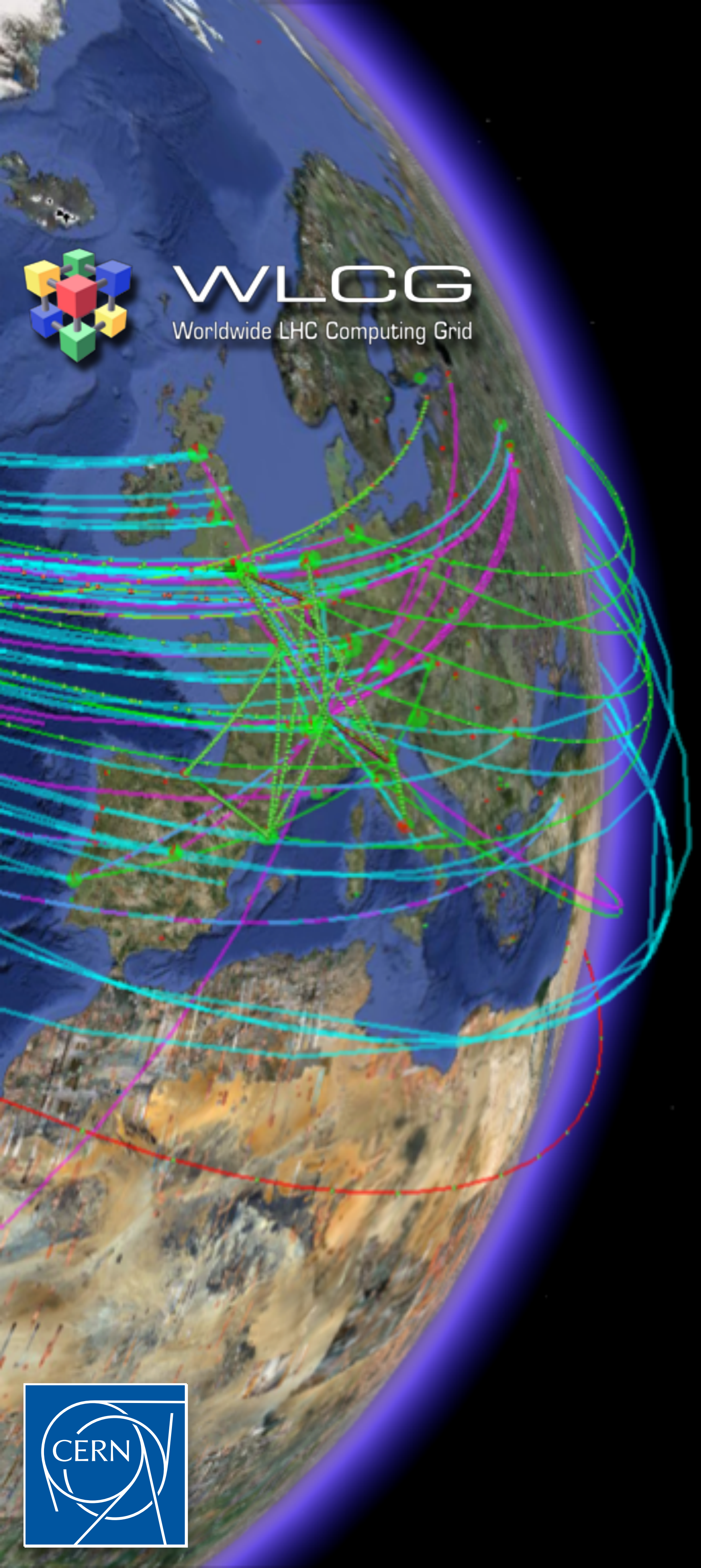
H → γγ and H→ 4l: high-sensitivity at low-m_H; high mass-resolution; pile-up robust
- ❑ analyses improved to increase sensitivity → new results from 2011 data
- ❑ all the data recorded so far in 2012 have been analyzed
- → results are presented here for the first time

Other low-mass channels: H→ $WW^{(*)}$→ lνlν, H→ ττ, W/ZH→ W/Z bb:
- ❑ E_T^{miss} in final state → less robust to pile-up
- ❑ worse mass resolution, no signal "peak" in some cases
- ❑ complex mixture of backgrou
- → understanding of the detect advanced, but results not yet
- → 2011 results used here for th

ATLAS: Status of SM Higgs searches, 4/7/20 3

- fantastic success (!!!)
  - ➡ software and computing had its share in it ...
  - ➡ full chain worked excellent:
    - from detector + trigger to
    - prompt calibration,
    - Tier-0 reconstruction,
    - GRID distribution and
    - fast distributed analysis !

We all know what happened next ...

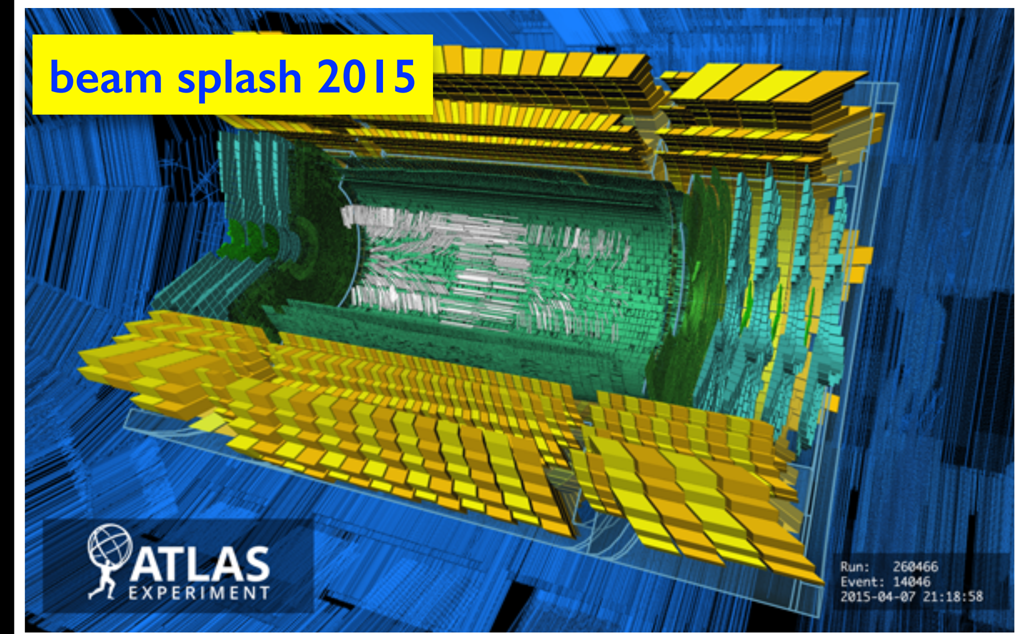


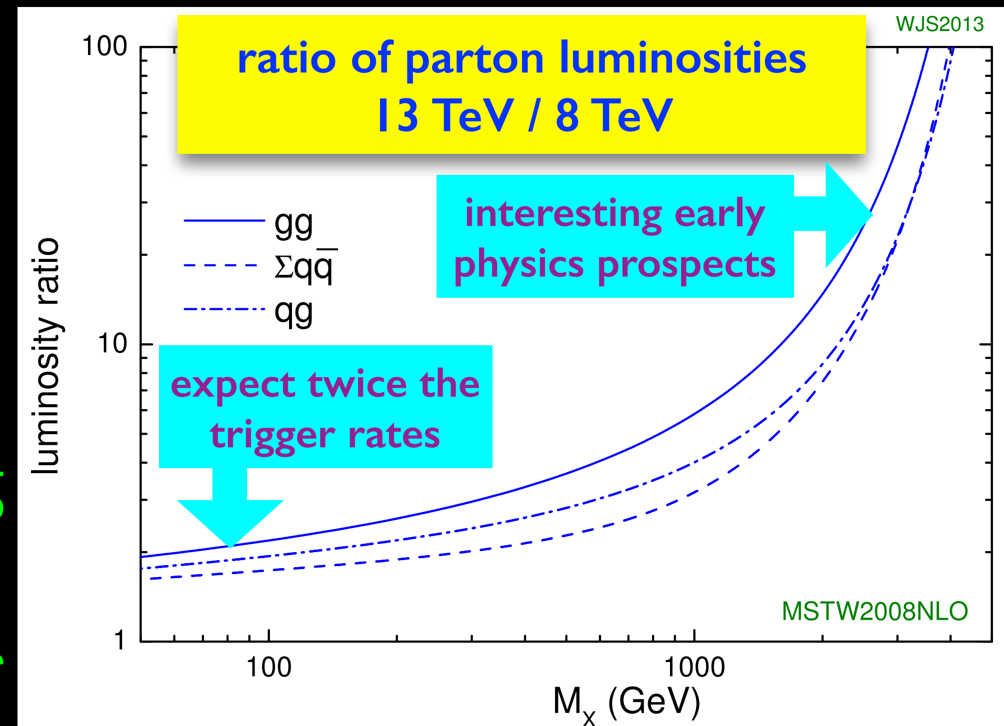
Preparing for Run-2: First Upgrades of the Offline Software

Run-2 has already started !

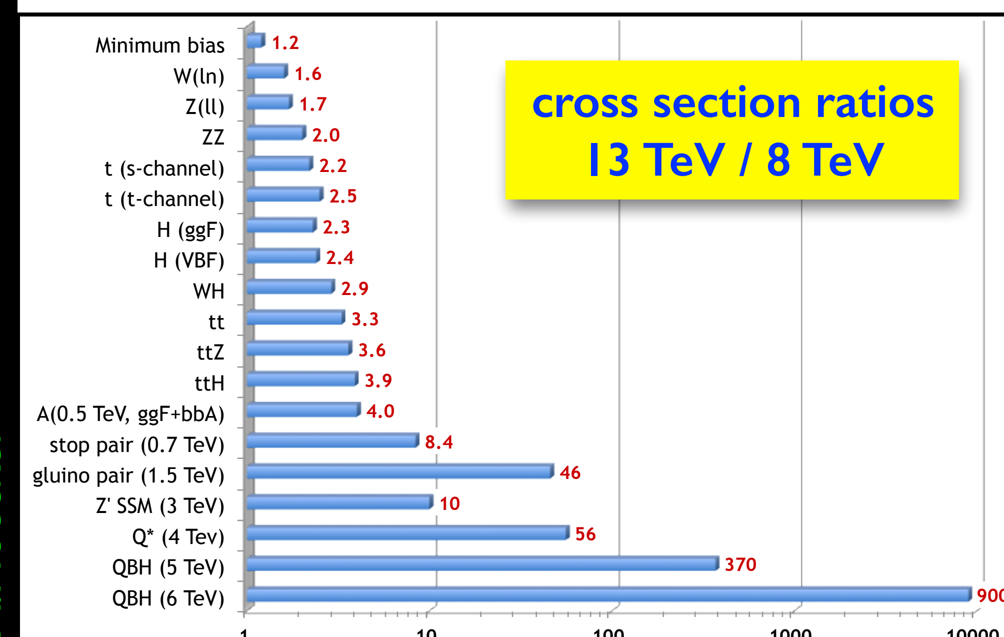
- LHC beam is back !
  - ➡ machine ready for 13 *TeV* operations
- Run-2 until 2018
  - ➡ expect L_{int} ~ 120 fb^{-1} with L_{peak} ~ $1.7 \cdot 10^{34}\ cm^{-2}s^{-1}$
    - need to be prepared for event pileup of 40
  - ➡ about factor > ~2 in interesting cross sections
    - expect twice trigger rates for same thresholds
- substantial discovery potential for high-mass objects running at 13 *TeV*
  - ➡ already with 1 fb^{-1} and m(system) > ~2 *TeV*
  - ➡ across all searches for ~ 10 fb^{-1}
- continue to explore the rich LHC physics program
  - ➡ Higgs, top, Standard Model, *b*-physics, ...

Markus Elsing





W.J. Stirling, private communication

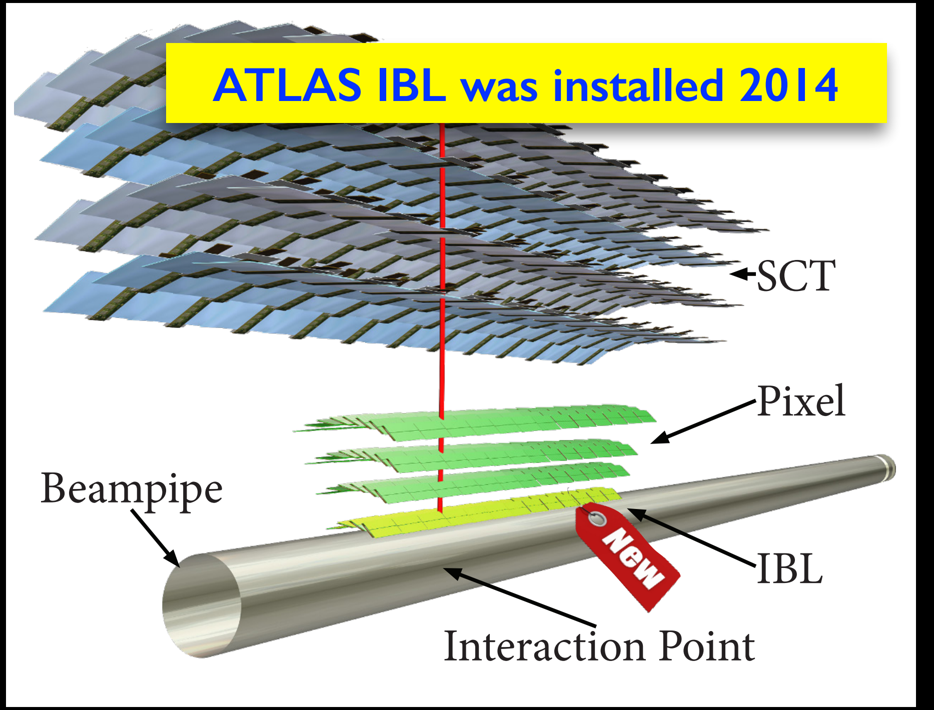


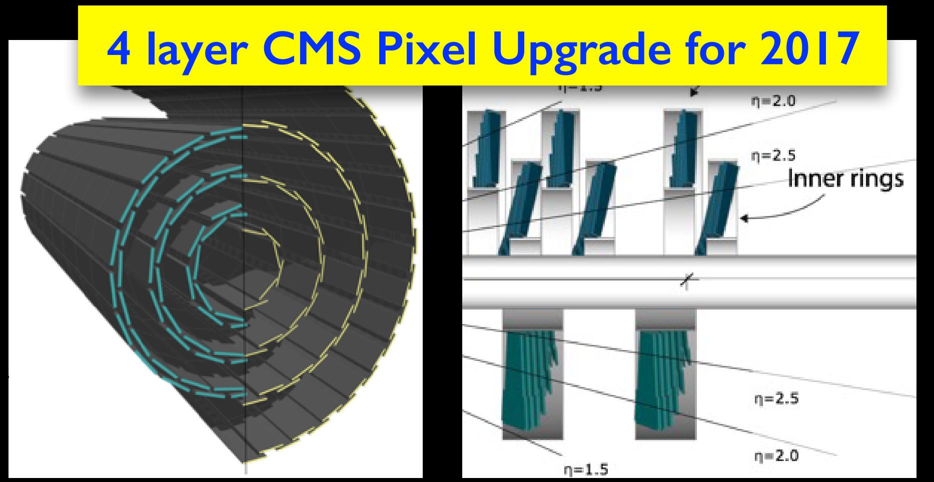
A.Hoecker

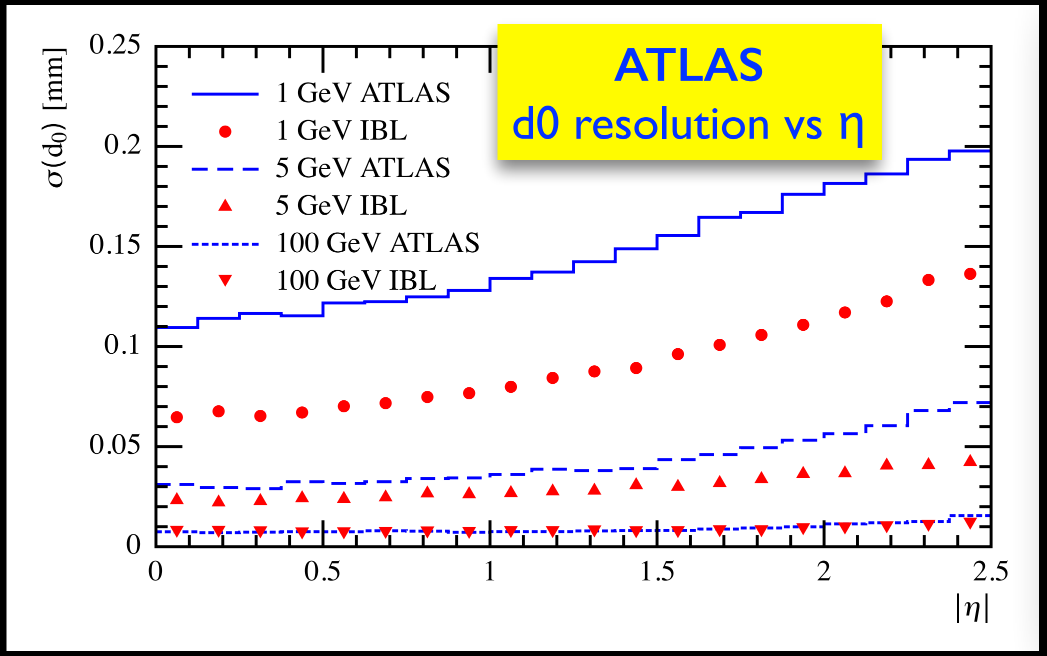
Pixel Upgrades for Run-2

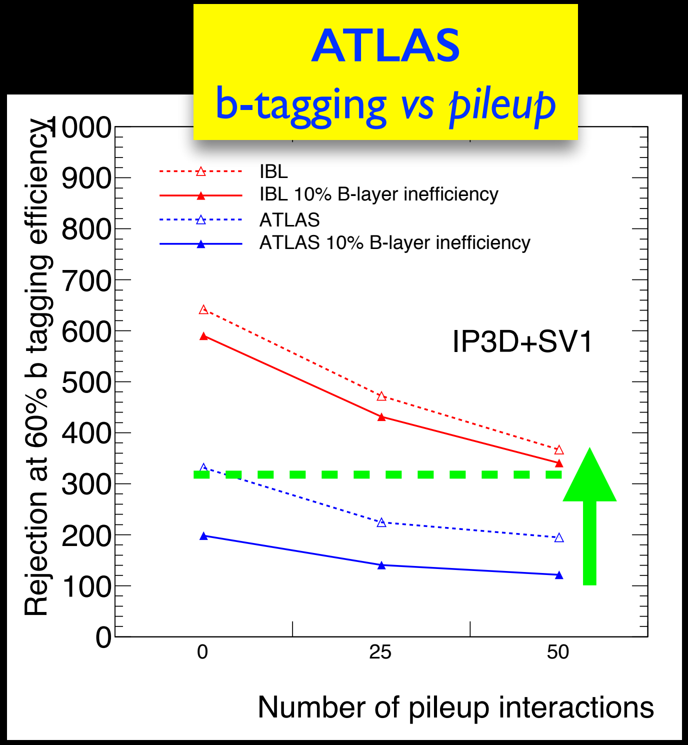
- aim is to mitigate effects of Run-2/3 pileup
  - ➡ ATLAS: IBL ready 2015, CMS: new 4 layer Pixels for 2017
  - ➡ both experiments add low mass Pixel layer close to beam
    - improves impact parameter resolution
  - ➡ additional hit to reduce fakes and/or improve efficiency
    - and use 4th layer in seeding to reduce CPU

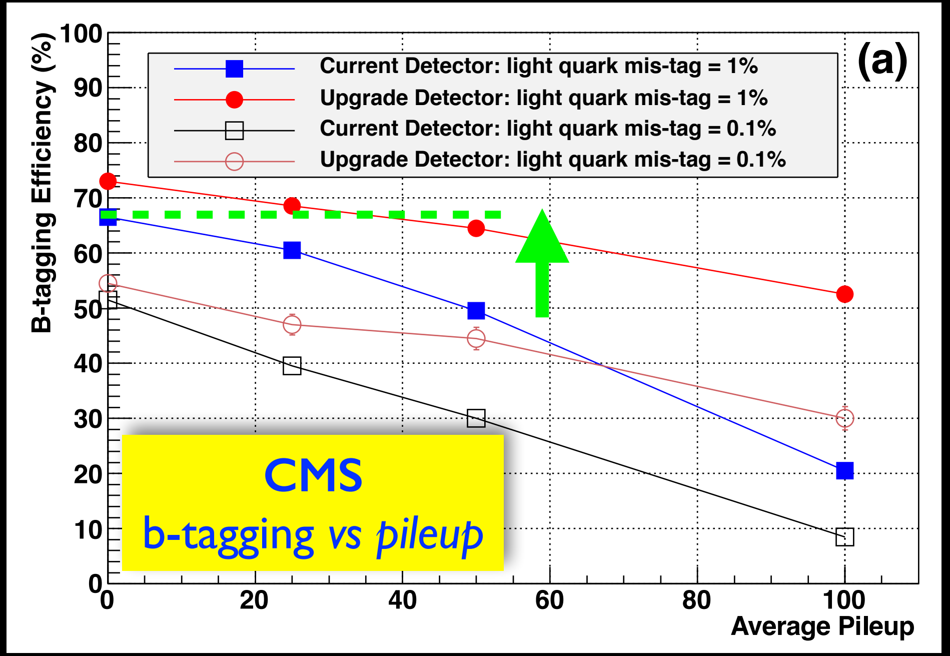
- significant improvements on b-tagging
  - ➡ at 50 pileup both experiments recover b-tagging performance like without pileup, or even improve upon it





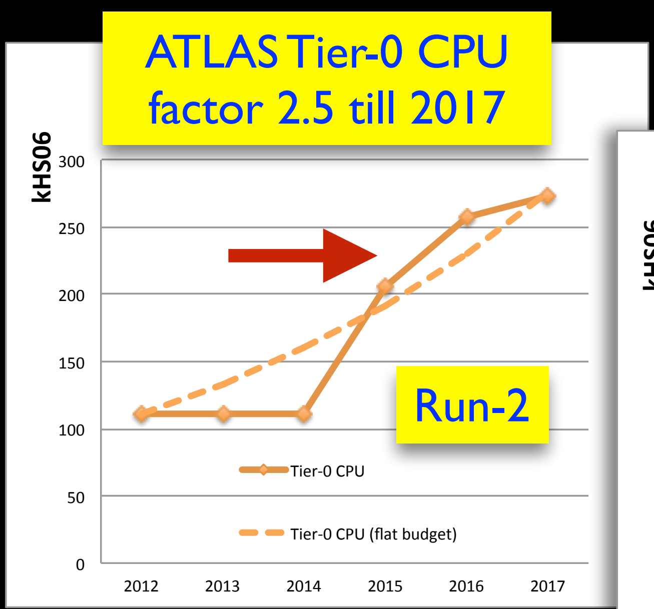




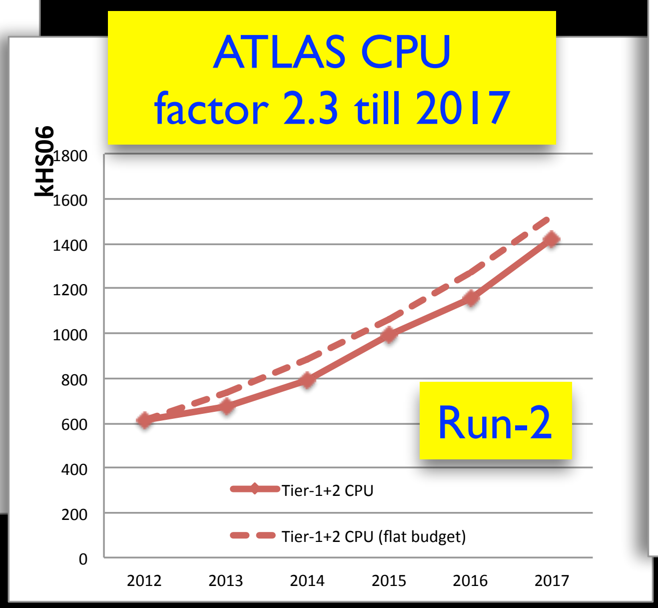


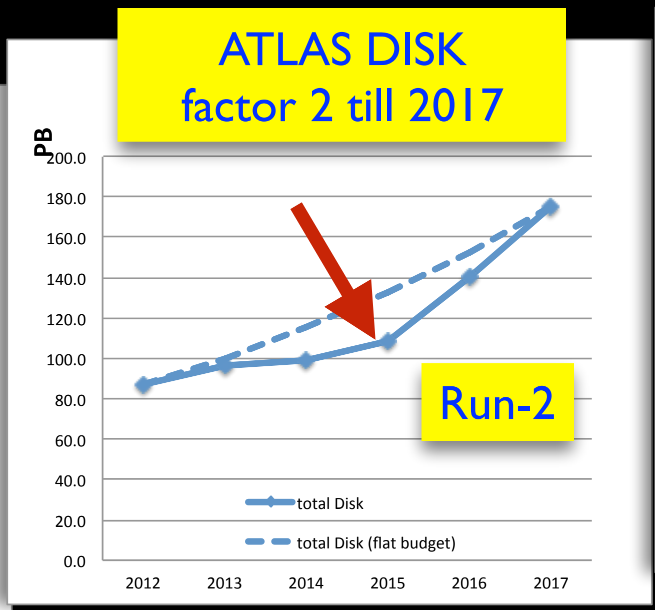
Computing Constraints for Run-2

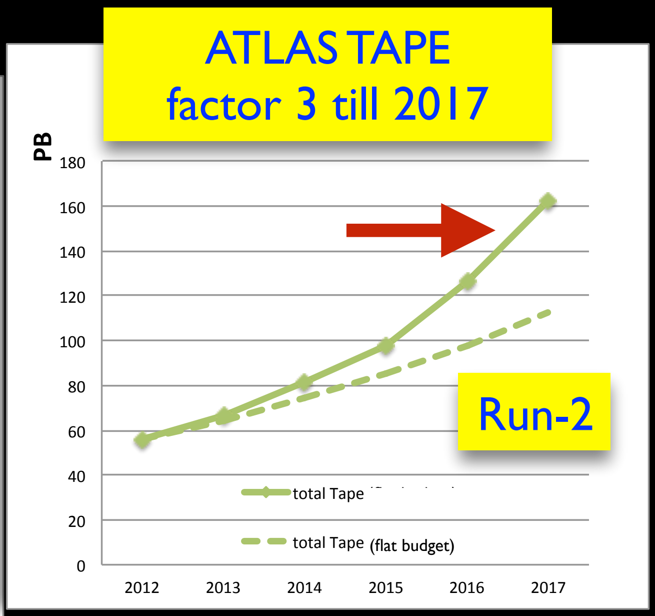
- unlike Run-1, computing resources will be limited !
  - ➡ assumption is a constant computing budget
  - ➡ interplay of technology advancement, market price and needed replacements



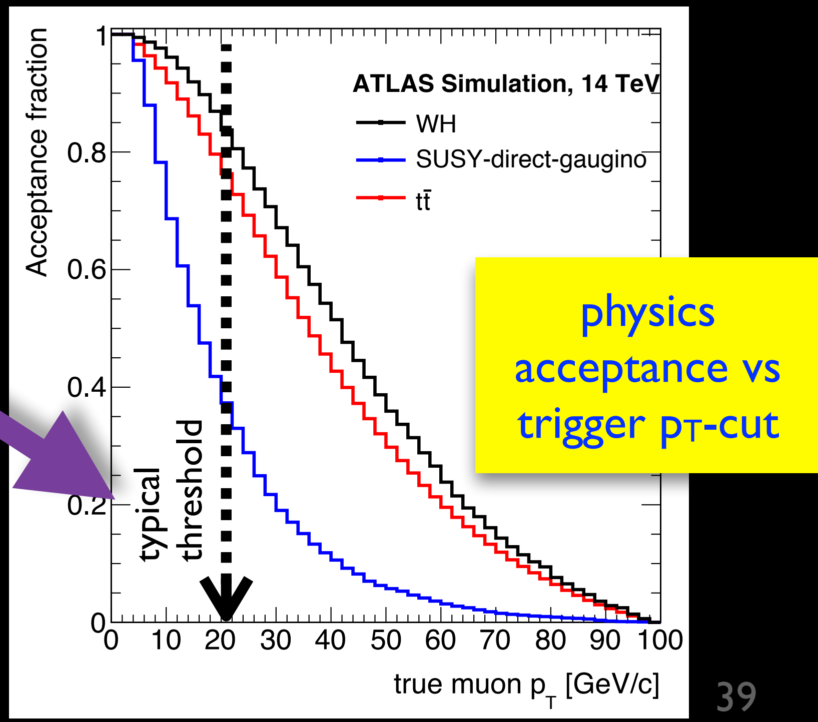
B.Kersevan et al.





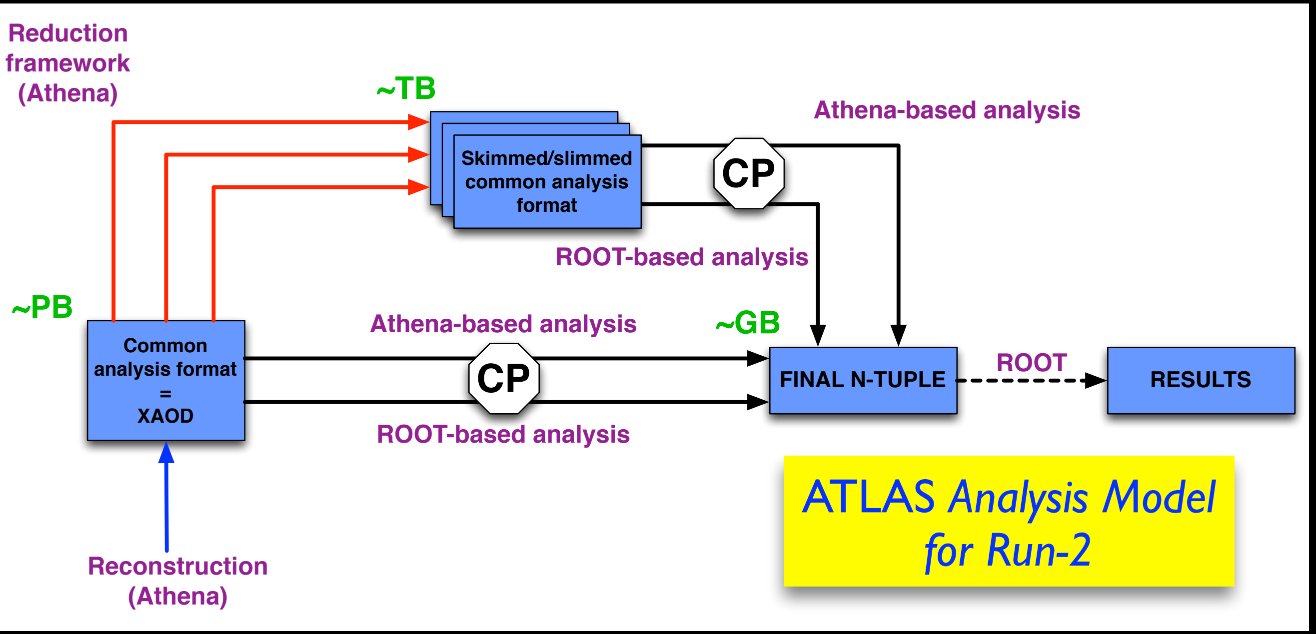


- motivation for LS1 software upgrades
  - ➡ ensure that Tier-0 can process 1 *kHz* trigger rate
  - ➡ optimise disk usage (e.g. ATLAS new Analysis Model)
- biggest problem will be disk !



ATLAS New Analysis Model for Run-2

- several issues with Run-1 model
  - ➡ analysis ntuples duplicate AOD (disk !)
  - ➡ production of ntuples costly (time !)
  - ➡ analysers develop in ROOT (compatibility !)

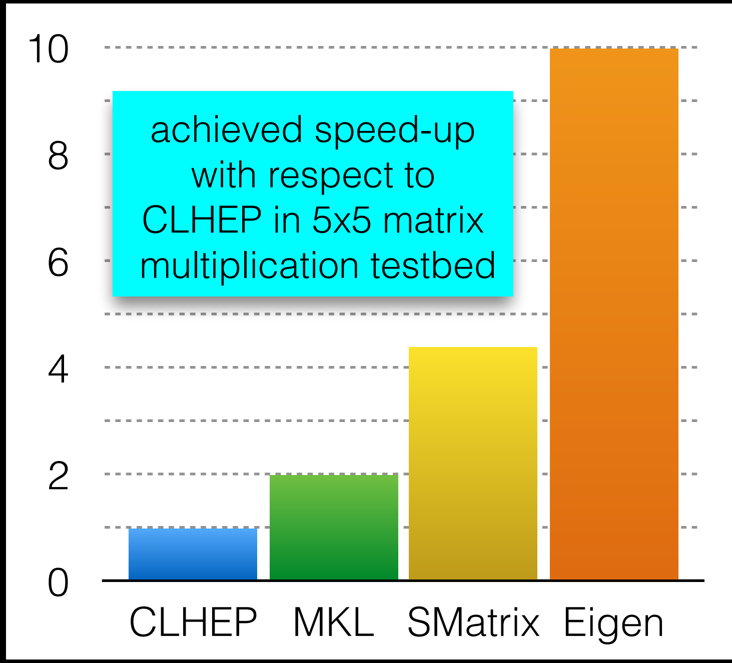


- "small" revolution for ATLAS
  - ➡ new format (xAOD) readable in ROOT
    - branch-wise reading at ROOT speed
    - object decoration with user data
  - ➡ centrally produce skims for analysers
    - train production model
    - smart slimming of xAOD objects
  - ➡ analysis tools transparently usable in ROOT and ATHENA
    - ROOT based and ATHENA based analysis software releases

- changes for other experiments are less extreme
  - ➡ similar pressure to reduce resource needs

Tracking Developments towards Run-2

- ATLAS and CMS focus on technology and strategy to improve CURRENT algorithms
  - ➡ improve software technology, including:
    - simplify EDM design to be less OO ("hip" 10 years ago)
    - ATLAS migrated to Eigen - faster vector+matrix algebra (CMS was already using SMatrix)
    - vectorised trigonometric functions (CMS: VDT or ATLAS: intel math lib)
    - work on CPU hot spots (e.g. ATLAS replaced F90 by C++ for B-field service)
  - ➡ tune reconstruction strategy (very similar in ATLAS and CMS):
    - optimise iterative track finding strategy for 40 pileup
    - ATLAS modified track seeding to explore 4th Pixel layer
    - CMS added cluster-shape filter against out-of-time pileup
- hence, mix of SIMD and algorithm tuning
  - ➡ CMS made their tracking as well thread-safe



Tuning the Tracking Strategy

- optimal seeding strategy depends on level of pileup (ATLAS)
  - ➡ fraction of seeds to give a good track candidate:

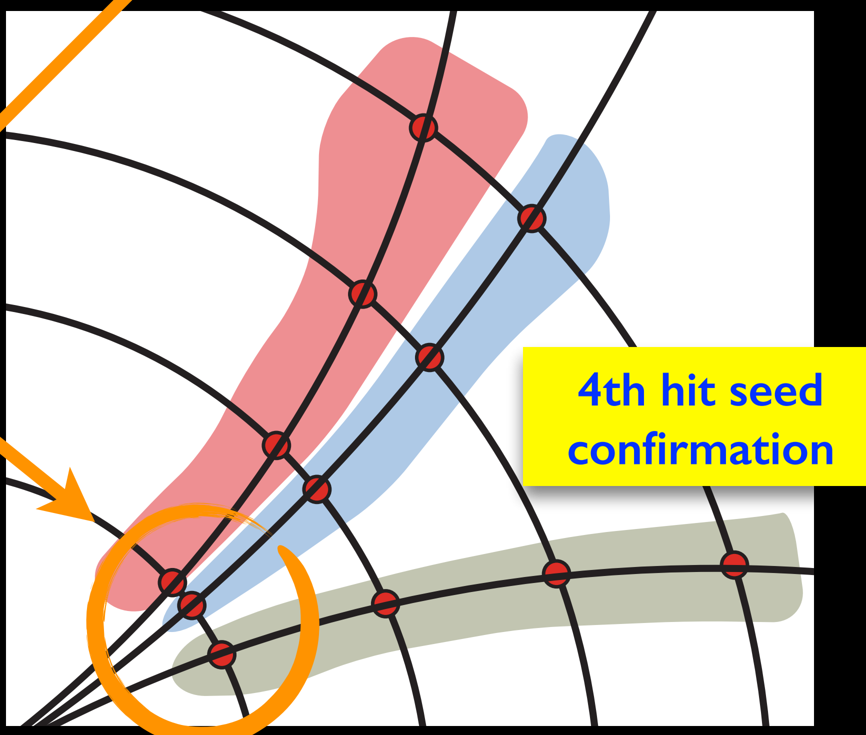
seed-triplets:
P = Pixel
S = Strips

pileup	"PPP"	"PPS"	"PSS"	"SSS"
0	57%	26%	29%	66%
40	17%	6%	5%	35%

  - hence start with SSS at 40 pileup !
  - ➡ further increase good seed fraction using 4th hit

pileup	"PPP+1"	"PPS+1"	"PSS+1"	"SSS+1"
0	79%	53%	52%	86%
40	39%	8%	16%	70%

  - takes benefit from new Insertable B-Layer (IBL)



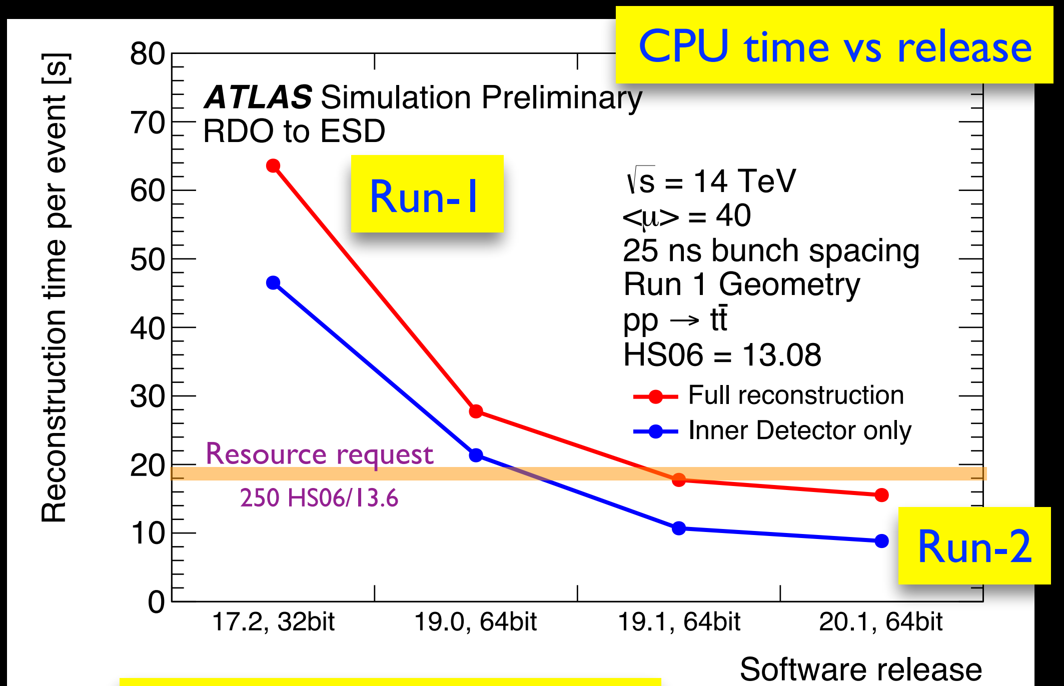
- final ATLAS Run-2 seeding strategy
  - ➡ significant speedup at 40 pileup (and 25 *ns*)

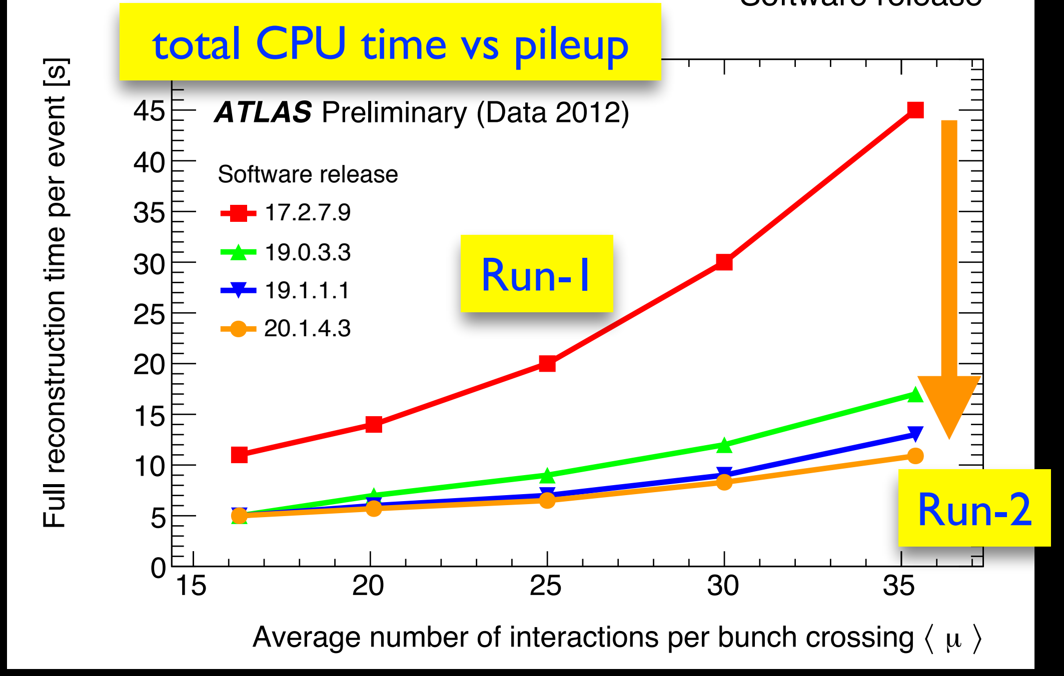
seeding	efficiency	CPU*
"Run-1"	94.0%	9.5 sec
"Run-2"	94.2%	4.7 sec

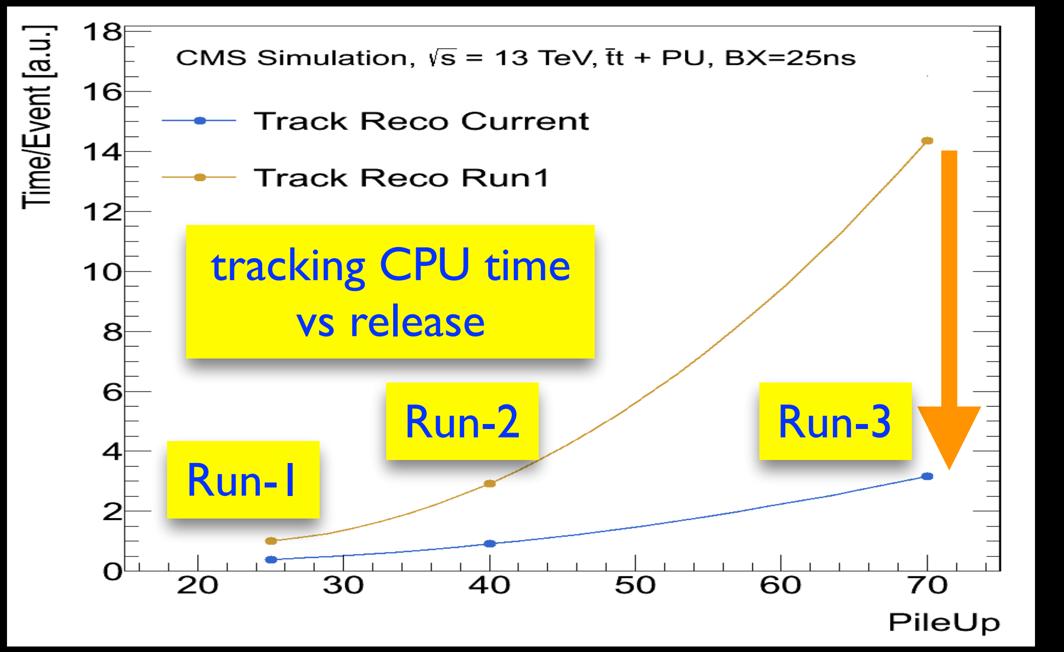
*on local machine

CPU for Reconstruction

- sum of tracking and general software improvements
  - ➡ improved software technology, including:
    - tracking related improvements
    - new 64 bit compilers, new tcmalloc
  - ➡ tune reconstruction strategy (very similar in ATLAS and CMS)
    - optimise track finding strategy for 40 pileup
    - faster versions of things like FastJet, ...
    - addressing other CPU hot spots in reconstruction
- huge gains achieved !
  - ➡ ATLAS reports overall factor > 4 in CPU time
    - touched >1000 packages for factor 5 in tracking
  - ➡ CMS reports overall factor > 2 in CPU time
    - on top of their 2011/12 improvements
    - as well dominated by tracking improvements
  - ➡ both experiments within 1 *kHz* Tier-0 budget
    - required to keep single lepton triggers

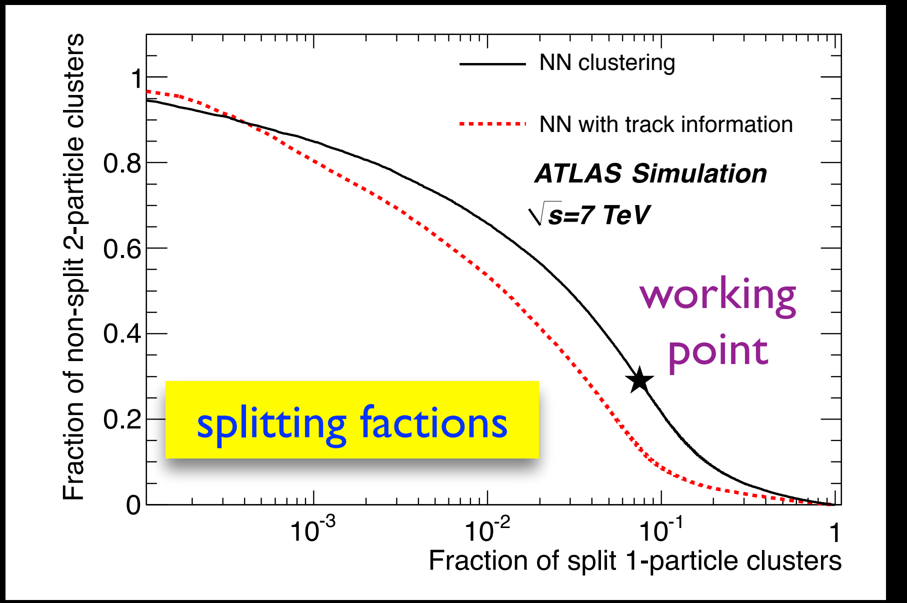


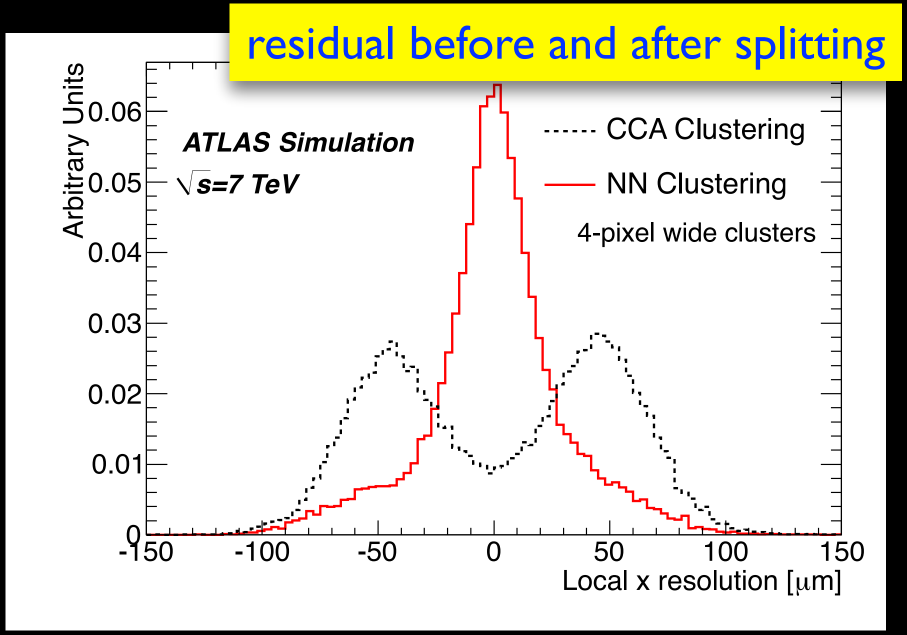


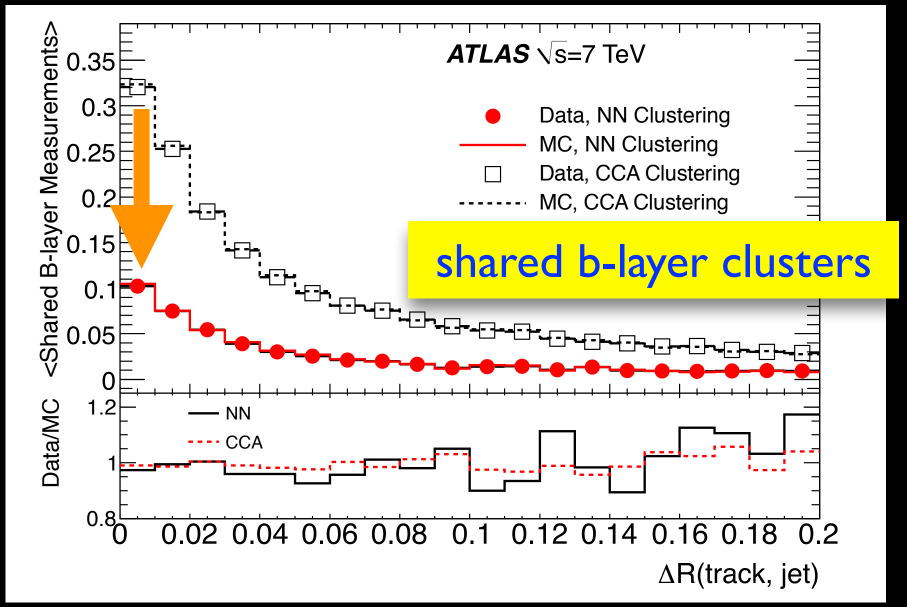


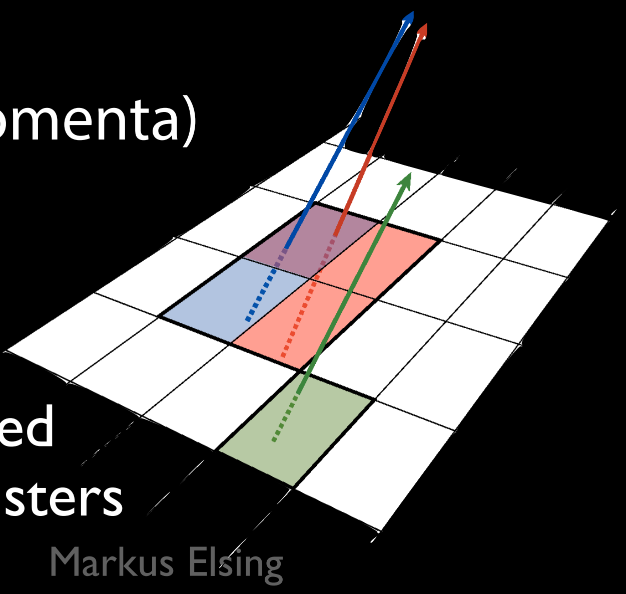
Tracking in dense Jets

- problem of cluster merging
  - ➡ merging when track separation reaches single Pixel size
  - ➡ during track reconstruction shared clusters are penalised to reduce fakes and duplicate tracks
- artificial neural network (NN)
  - ➡ identify merged clusters and splitting them
  - ➡ during Run-I these were duplicated
    - though with different cluster positions
  - ➡ performance in these environments was known to be suboptimal
- crucial in many areas:
  - ➡ b-tagging (especially at high momenta)
  - ➡ jet calibration and particle flow
  - ➡ 3-prong τ identification







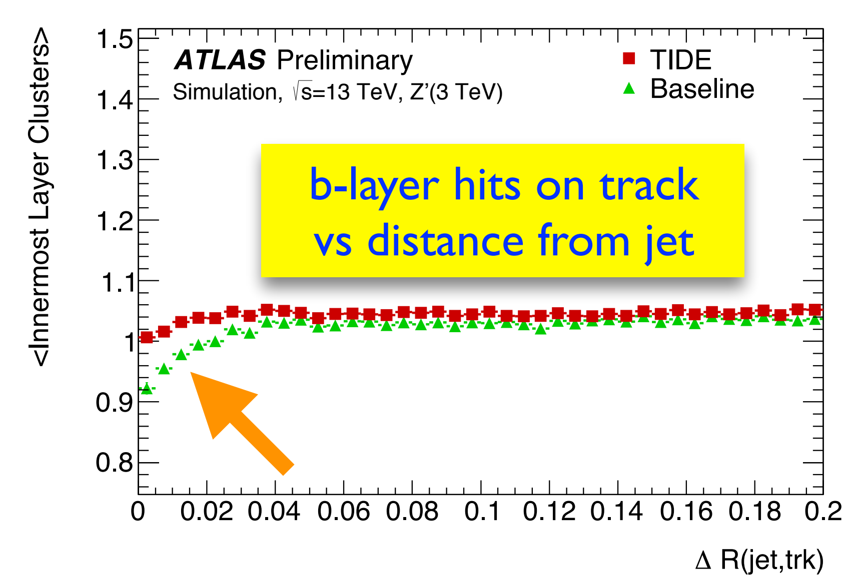


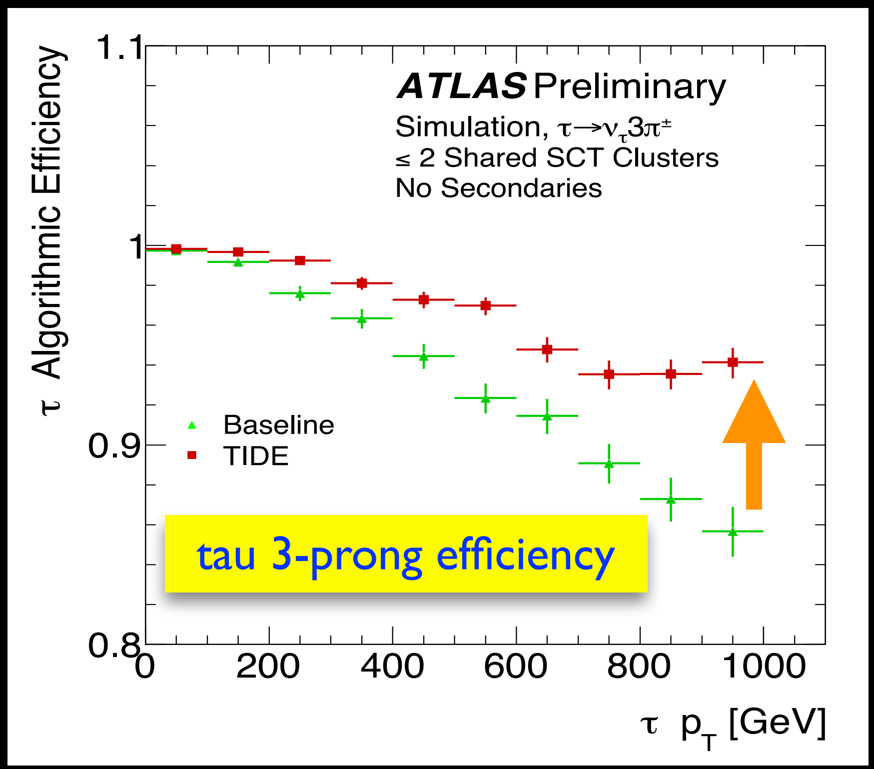
merged
Pixel clusters

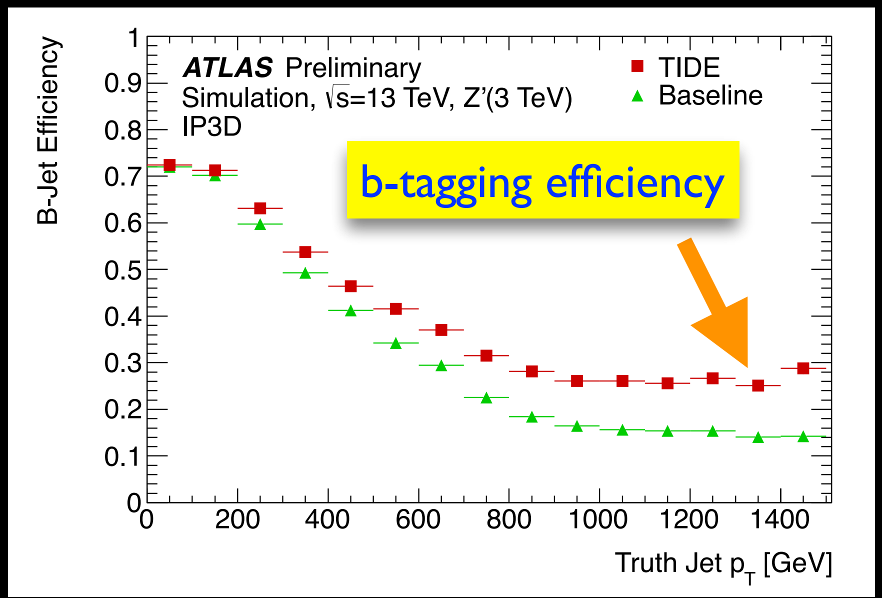
Run-2 Tracking in dense Jets

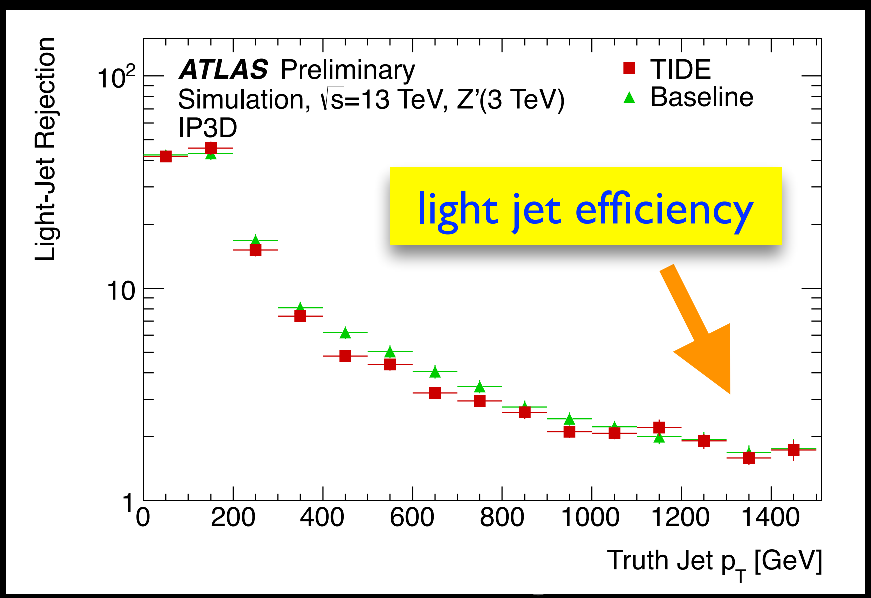
- new strategy delays NN cluster splitting
  - ➡ pattern runs with merged clusters to find all candidates
  - ➡ split clusters in ambiguity solution using tracks
    - more information used to improve splitting performance
  - ➡ improve logic to allow sharing (un-"splitable") clusters
- significant improvement at high-p_T
  - ➡ tau 3-prong inefficiency halved
  - ➡ b-tagging efficiency doubled

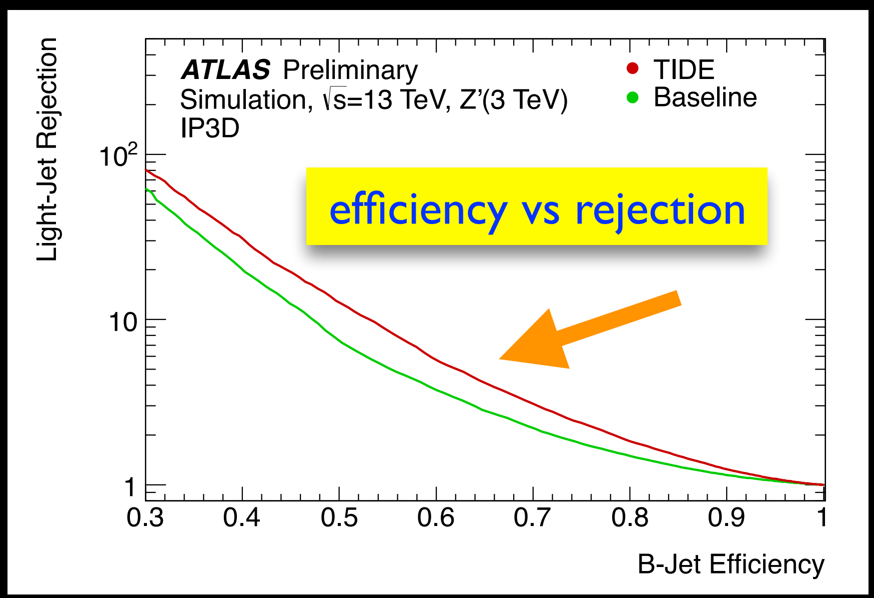
(CMS uses new splitting in clustering for Run-2)

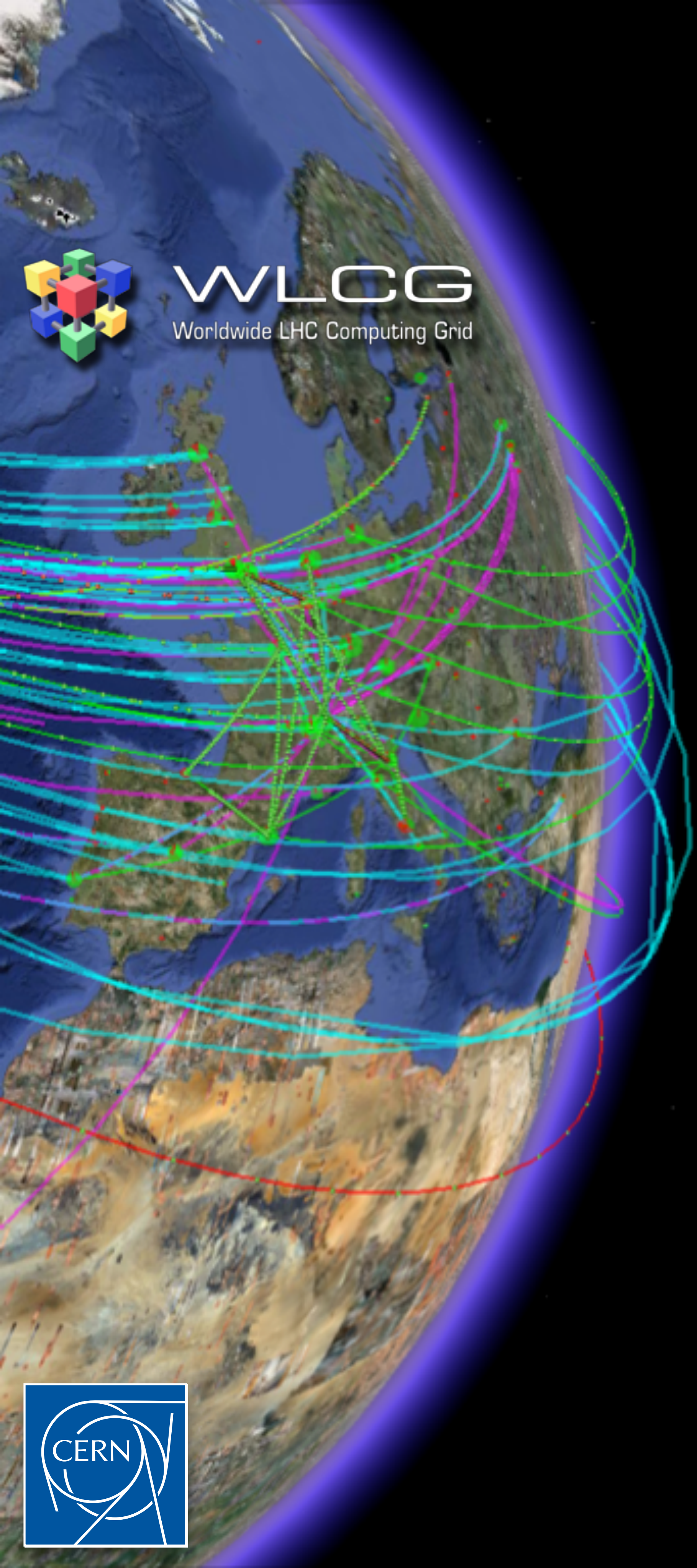


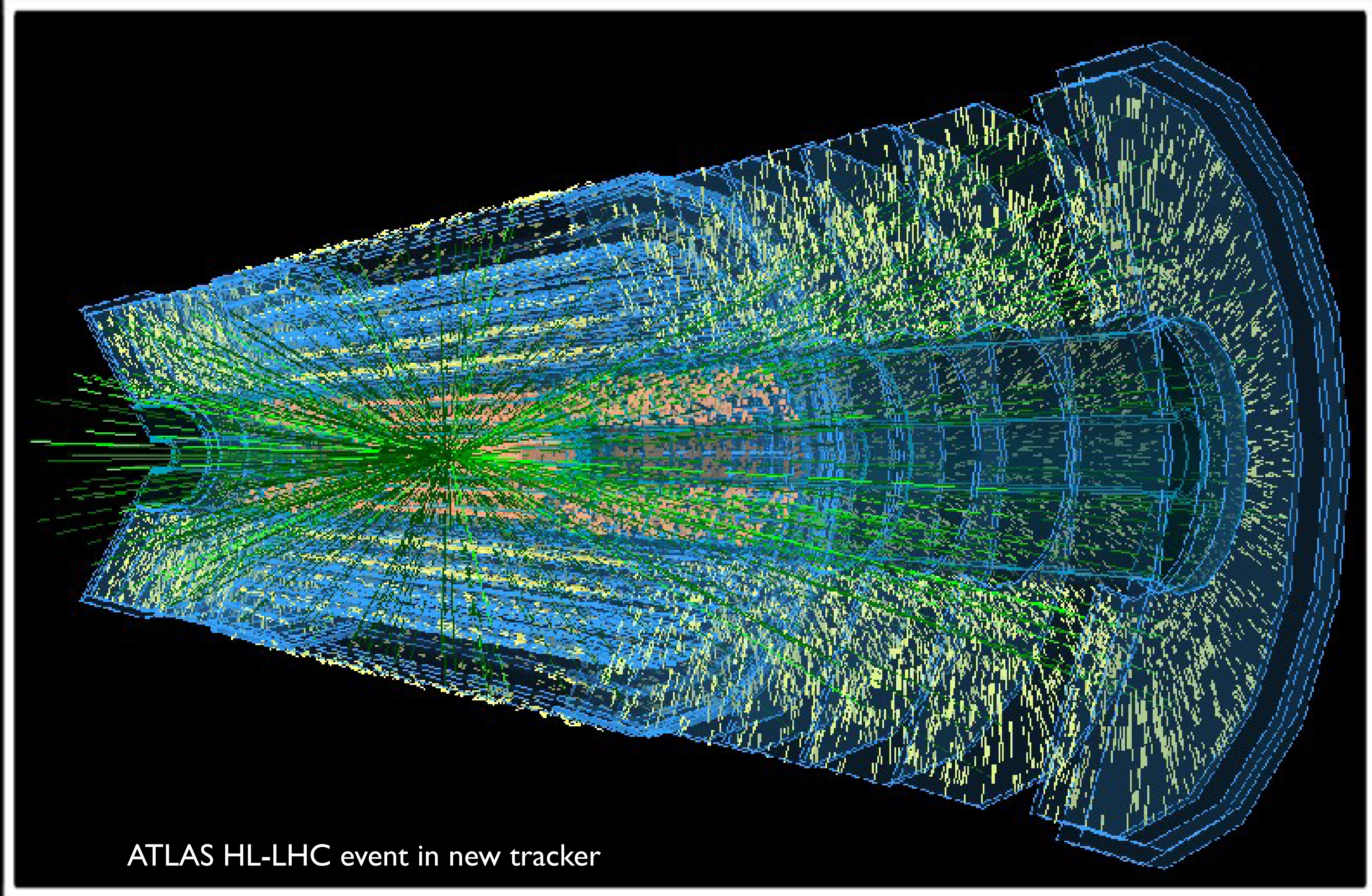










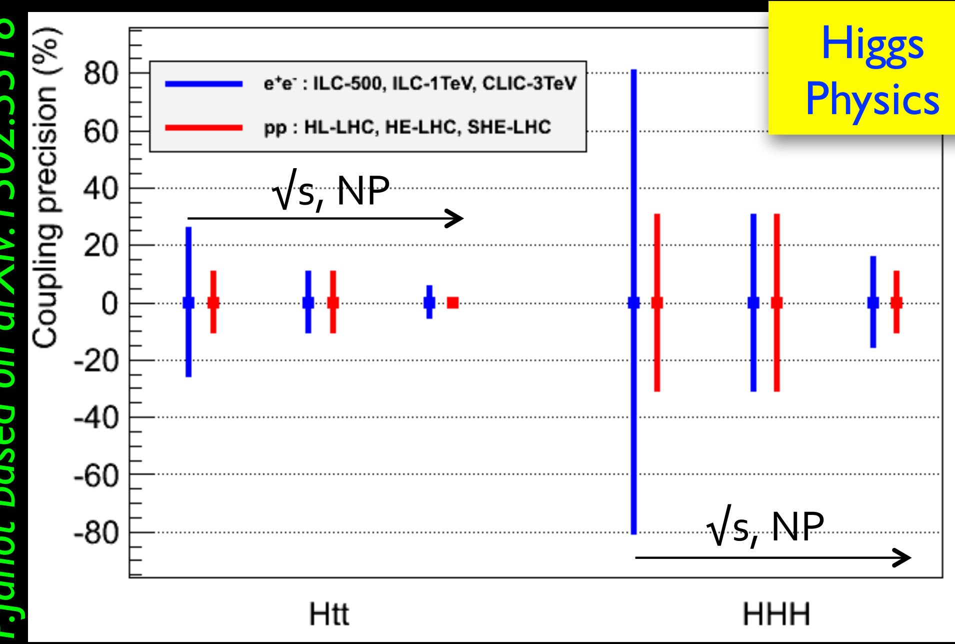

ATLAS HL-LHC event in new tracker

Software for
Detector Upgrades

LHC Upgrade Physics Goals

- **Higgs couplings and properties**
  - ➡ few % on couplings possible with 3000(350) pb^{-1}
  - ➡ new channels opening up (e.g. $H \rightarrow \mu\mu$)
  - ➡ measure *ttH* and 30% on Higgs self coupling
- study **vector boson scattering**
  - ➡ Higgs restores unitarity in VV scattering around 1 *TeV*
- extend reach for **new physics** searches
  - ➡ e.g. for 3rd generation squarks and gauginos
- **LHCb** physics reach with 50 fb^{-1}
  - ➡ unique for new physics searches in B_s system
    - precision measurement of $B_{(s)} \rightarrow \mu\mu$
    - few % in CP violating ϕ_s from $B_s \rightarrow \phi\phi$
    - CP violation in $B_s \rightarrow J/\Psi\phi$
  - ➡ unprecedented **charm** yields
    - search for CP violation in charm decays

Planot based on arXiv:1302.3318



ATLAS-PHYS-PUB-2013-02

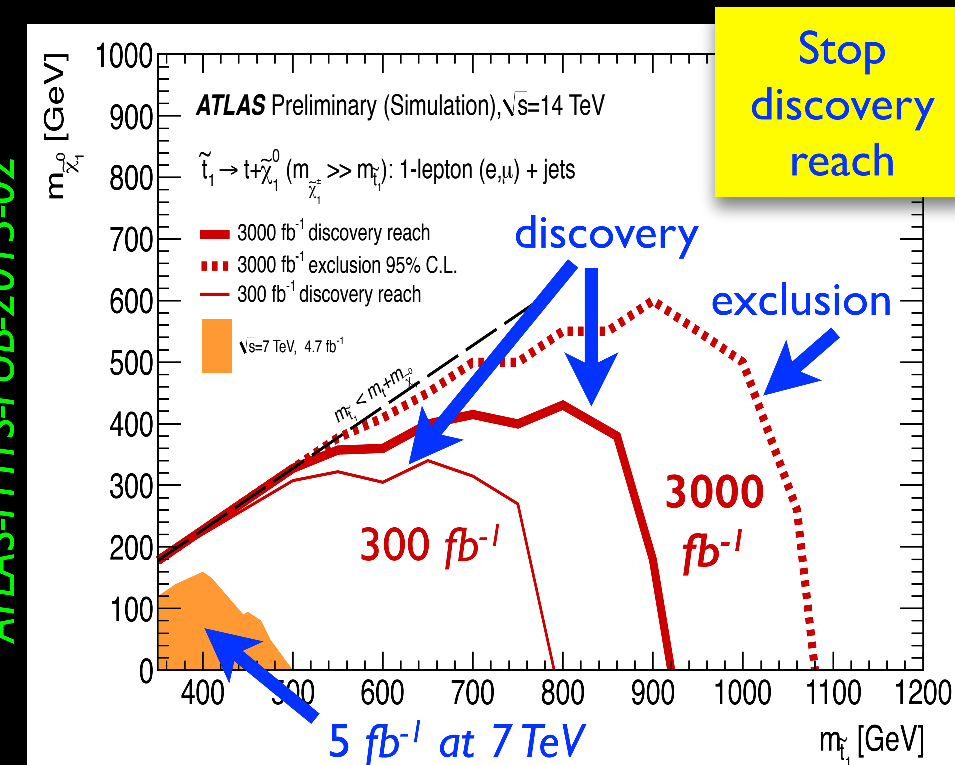
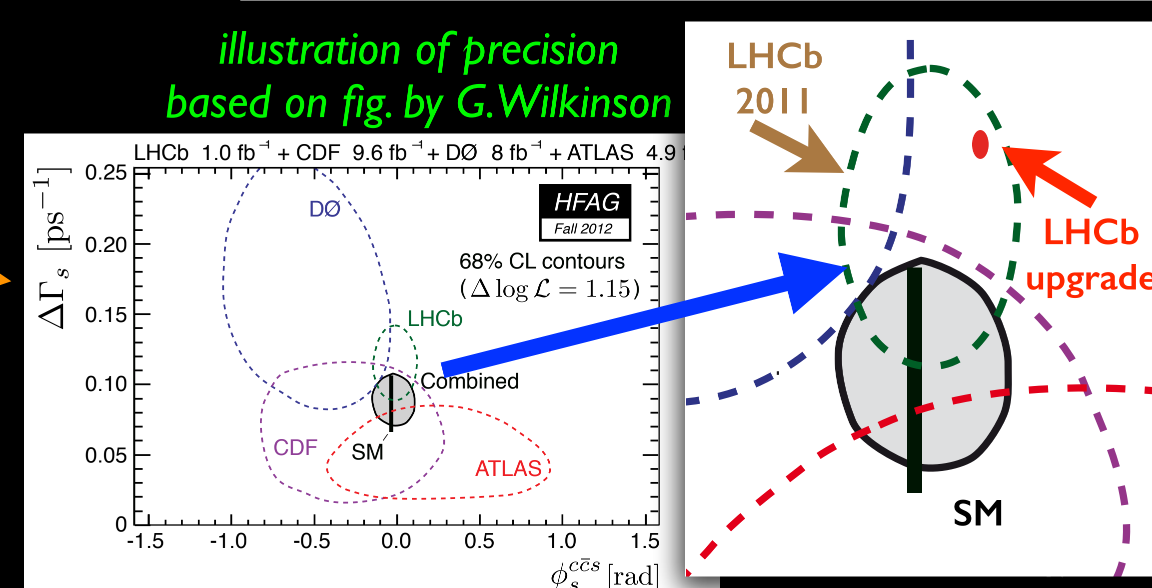
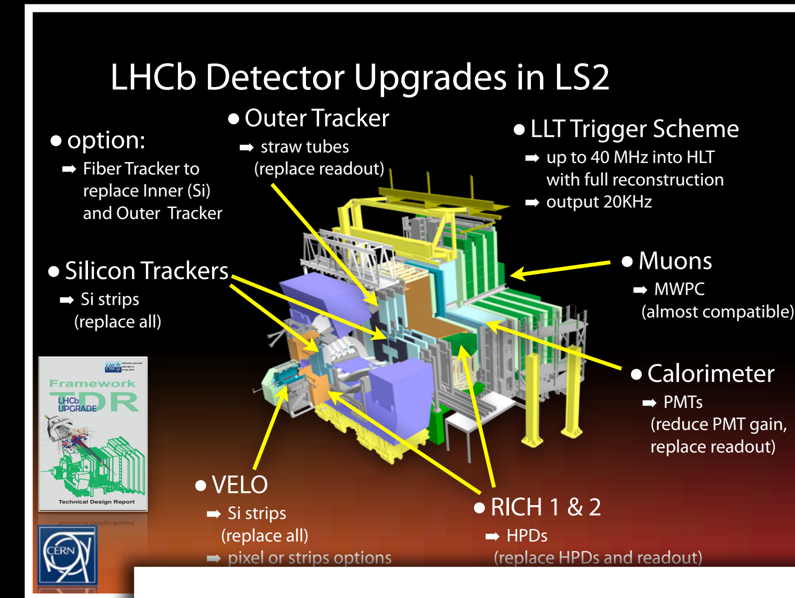


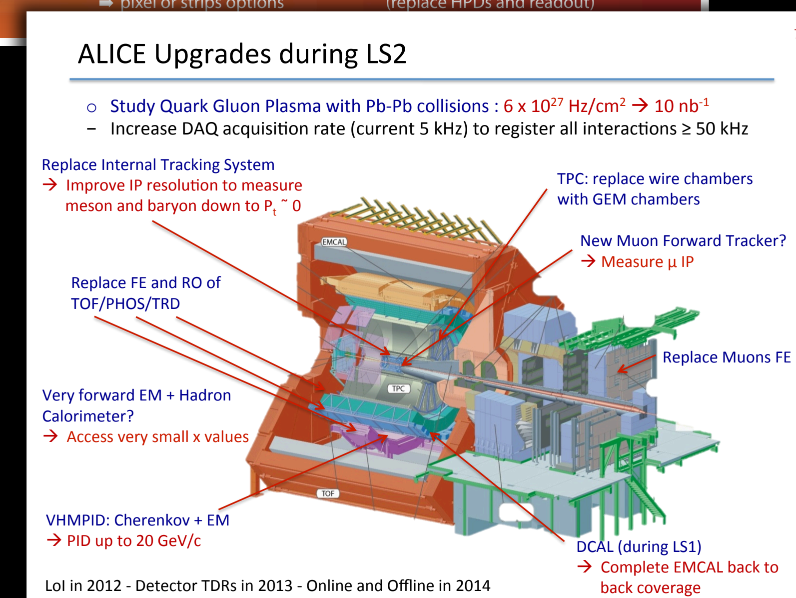
illustration of precision based on fig. by G.Wilkinson

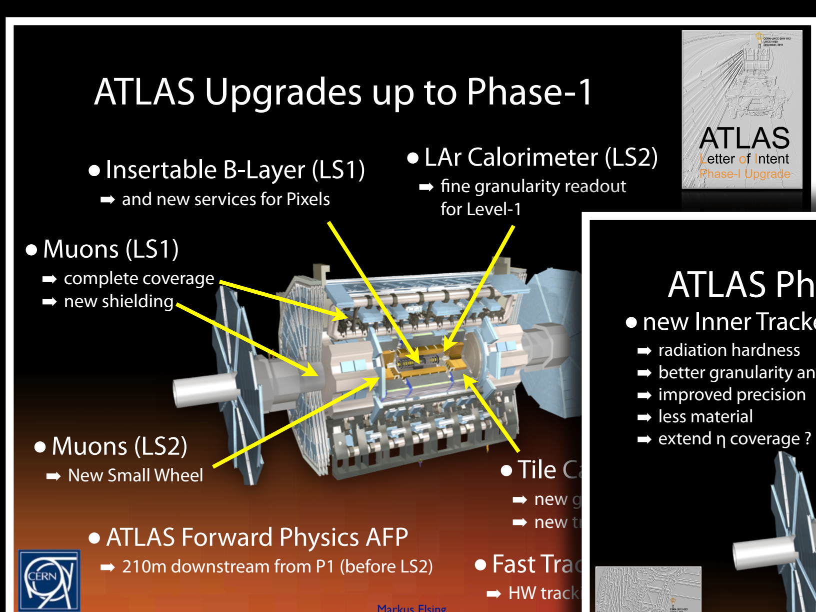


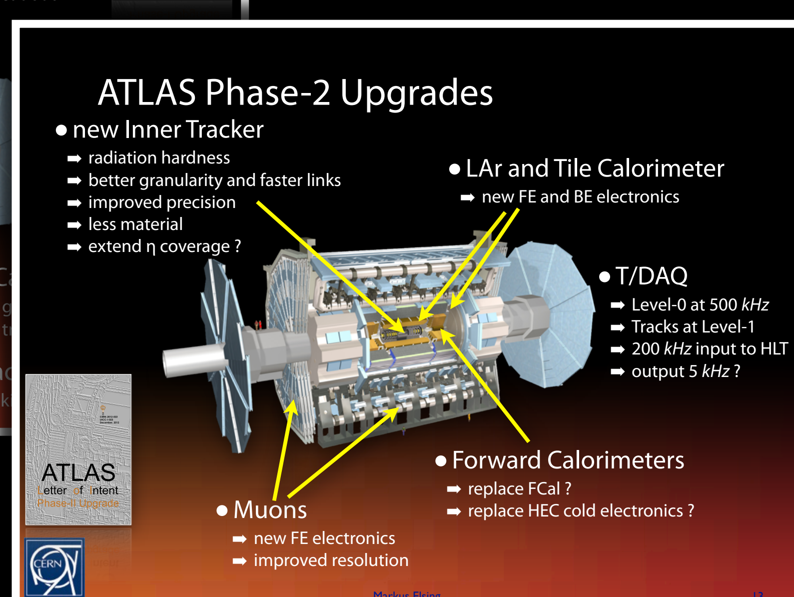
LHC Upgrade Program

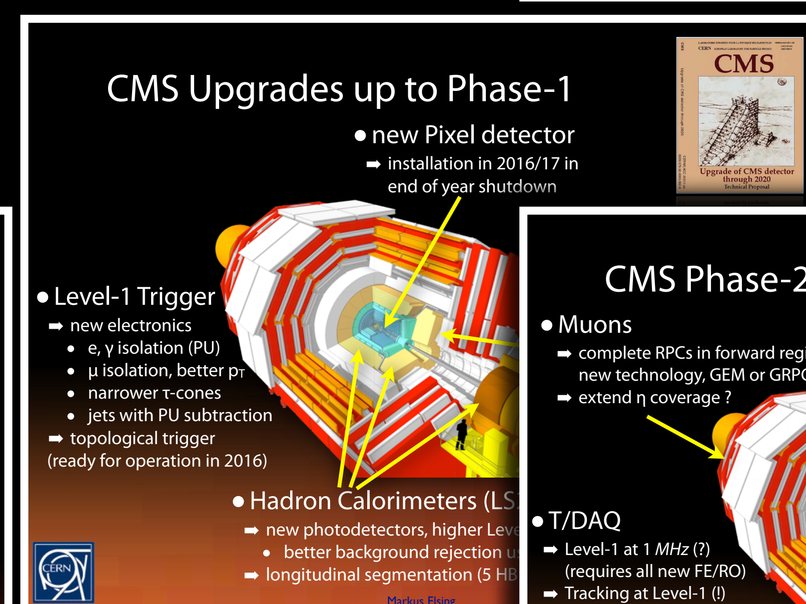
- Phase-1 upgrades (2018→)
  - ➡ LHCb and ALICE trigger-less readout
  - ➡ CMS and ATLAS ready for 350 fb^{-1}
- Phase-2 upgrades (2023→)
  - ➡ HL-LHC upgrades for CMS and ATLAS for 3000 fb^{-1}
- software plays key role in this program
  - ➡ physics prospects, detector design, TDRs...
  - ➡ preparing offline and trigger for detector upgrades itself

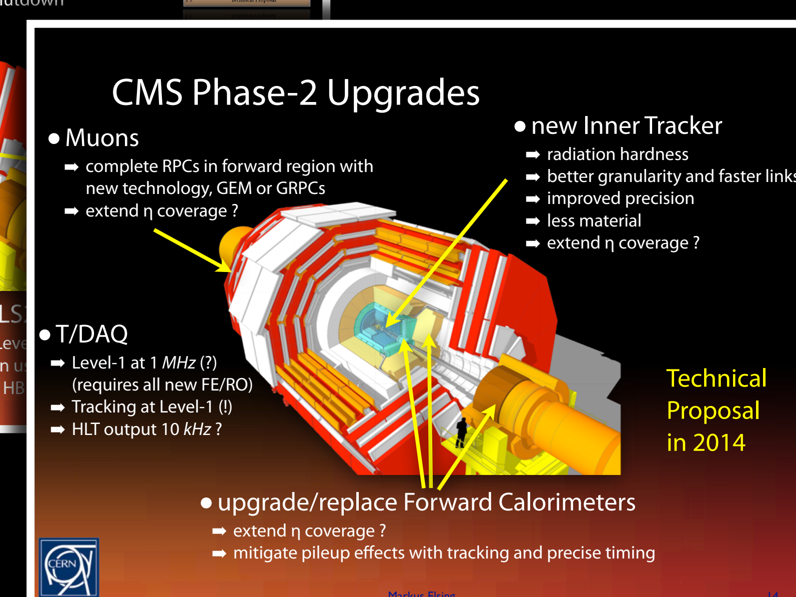










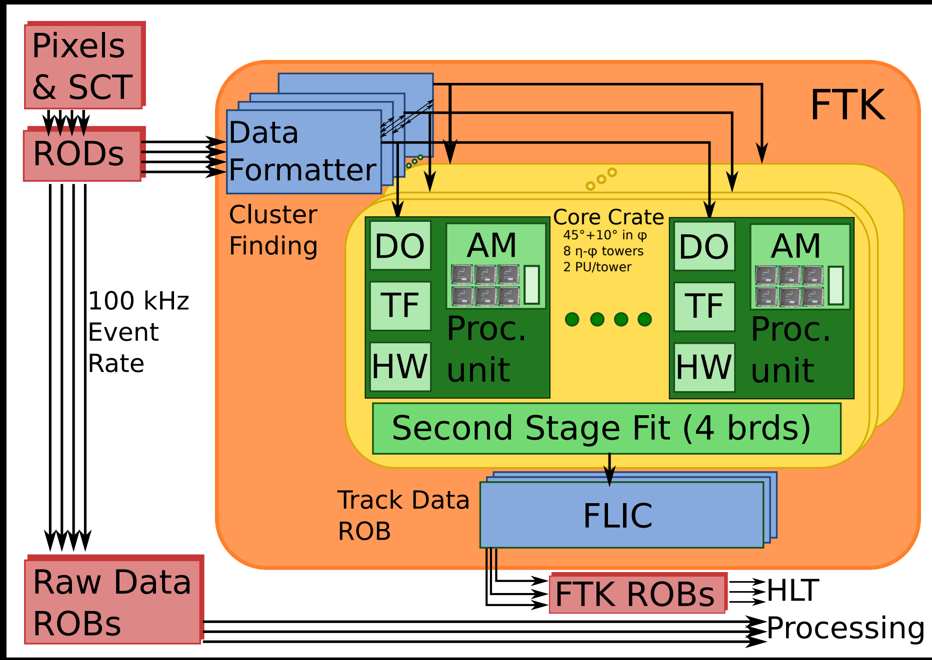


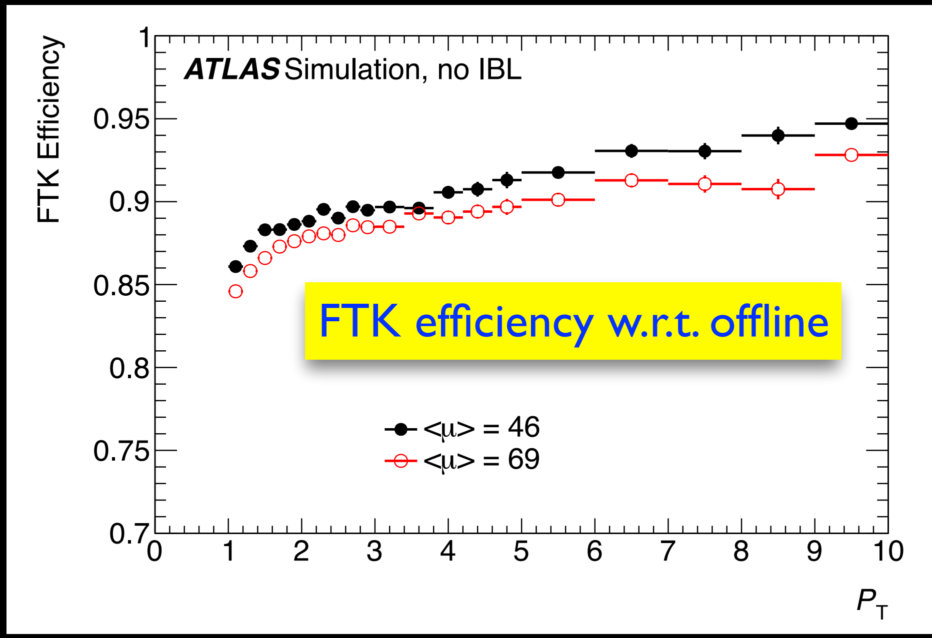
Hardware based Tracking ?

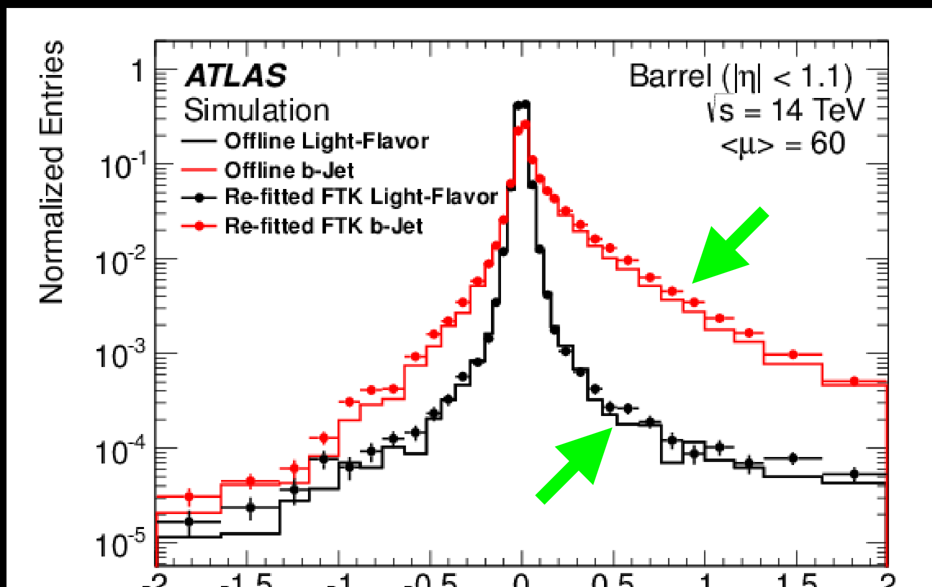
- current ATLAS trigger chain
  - ➡ Level-1: hardware based (~50 *kHz*)
  - ➡ Level-2: software based with regional access to full granularity data (~5 *kHz*)
  - ➡ Event Filter: software trigger (~500 *Hz*)

tracking enters here

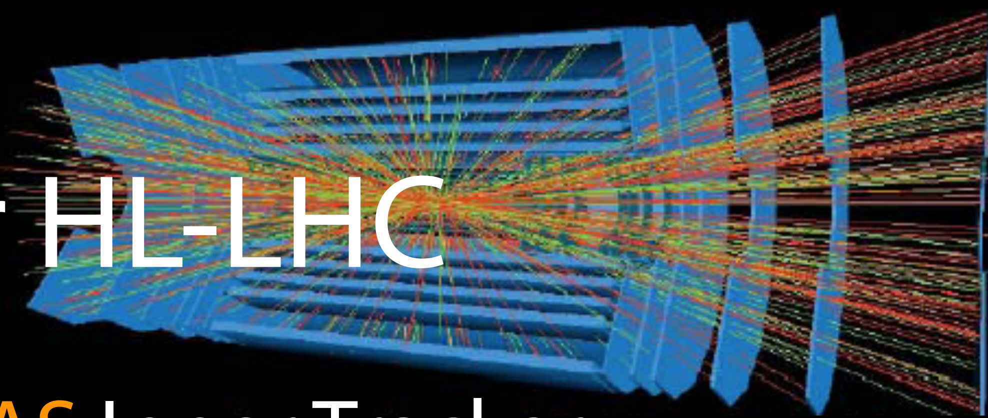
- ATLAS installs FTK during Run-2
  - ➡ hardware track reconstruction for Level-2 Trigger
    - associative memory (AM) chips to find patterns
    - FPGA based track parameter estimation
    - "Hit Worrier" (HW) to remove fakes
  - ➡ slice installed for 2015, full coverage in 2016
    - will replace software based Level-2 tracking in ATLAS
  - ➡ full event track reconstruction at latency of ~ 100 *μs*
    - fast track confirmation of Level-1 triggers
    - particle flow like tau tagging
    - fast b-jet tagging
    - pileup corrections for jets and missing E_T
  - ➡ excellent performance for Level-2 purposes
    - track efficiency is 90-95% w.r.t. offline
    - track refit using full fitter recovers offline resolution







impact parameter FTK+refit vs offline

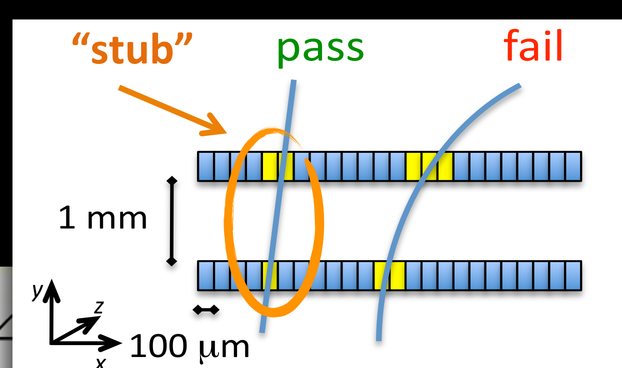


Inner Tracker Upgrades for HL-LHC

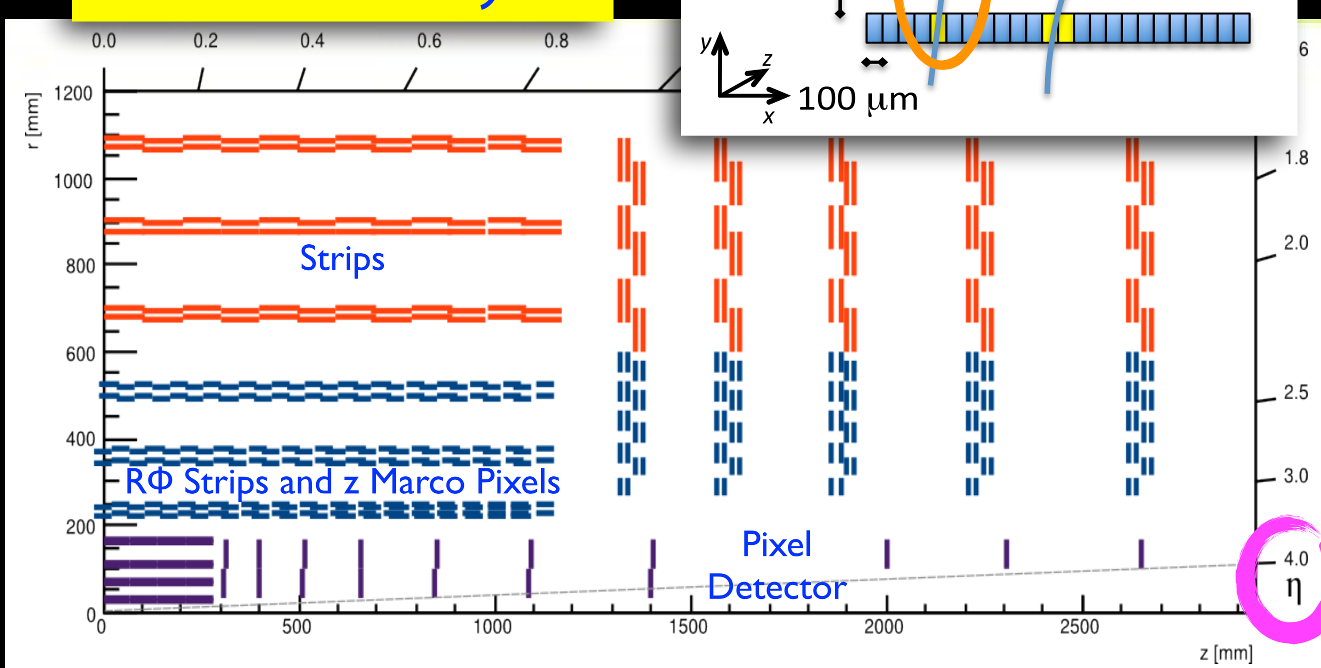
- **CMS Inner Tracker**
  - ➡ Strip tracker replacement
    - several layouts under consideration
    - short strips in $R\phi$, macro-pixels in z
  - ➡ Level-1 track trigger with high p_T stubs
    - correlate 2 sensors, threshold ~ *2 GeV*
    - pattern in FPGA or AM chips, FPGA fit
  - ➡ Pixels: extend η coverage to 4 (!)

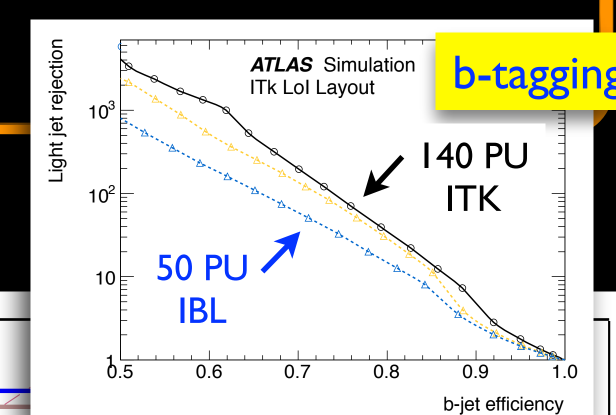
optimised for fast (HW) tracking

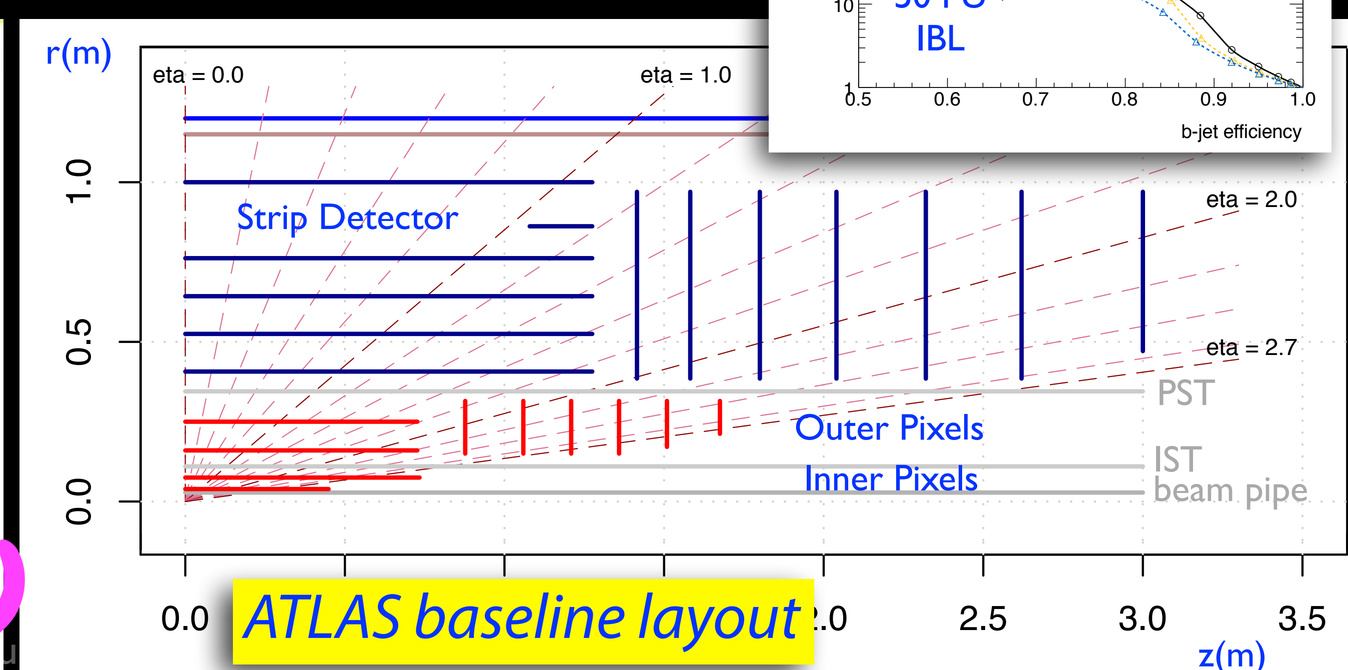
- **ATLAS Inner Tracker**
  - ➡ baseline: all silicon tracker, 14 hits
    - robust tracking @140 PU for η<2.5
  - ➡ Strip tracker with short strips + stereo
  - ➡ Pixels cover η < 2.7 (muons)
    - inner Pixels replaceable, reduced pitch
    - alternative layouts ("Alpine", conical)
  - ➡ Level-1 track trigger seeded by Level-0
    - FTK inspired, reduced latency



CMS baseline layout



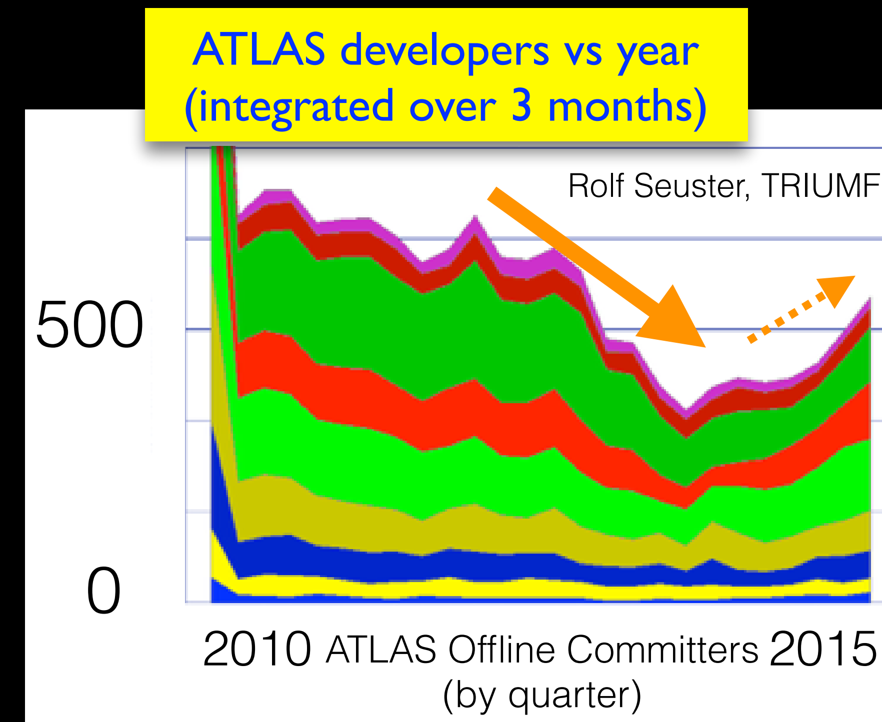


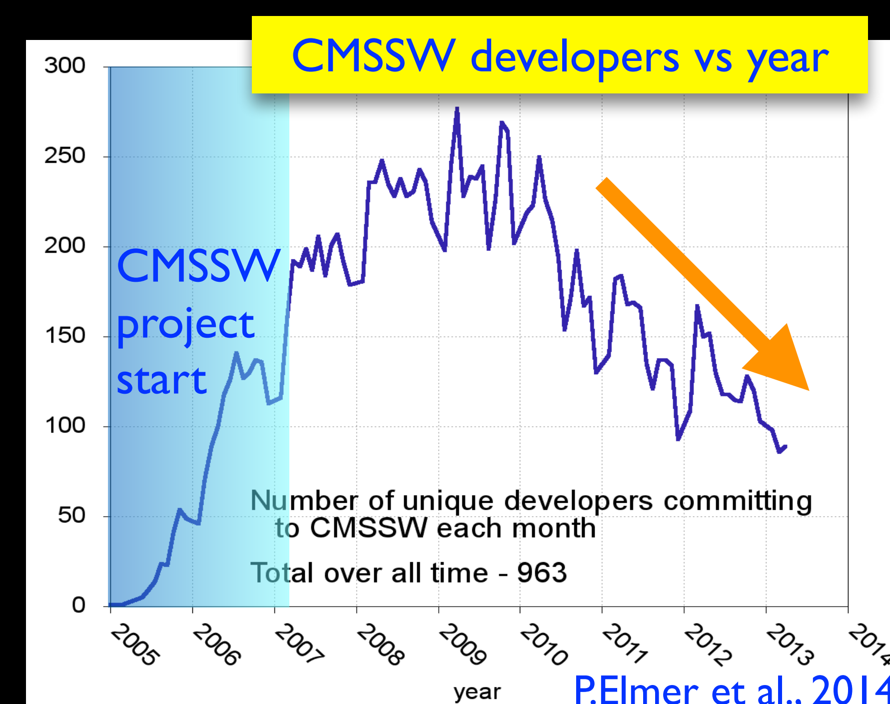


ATLAS baseline layout

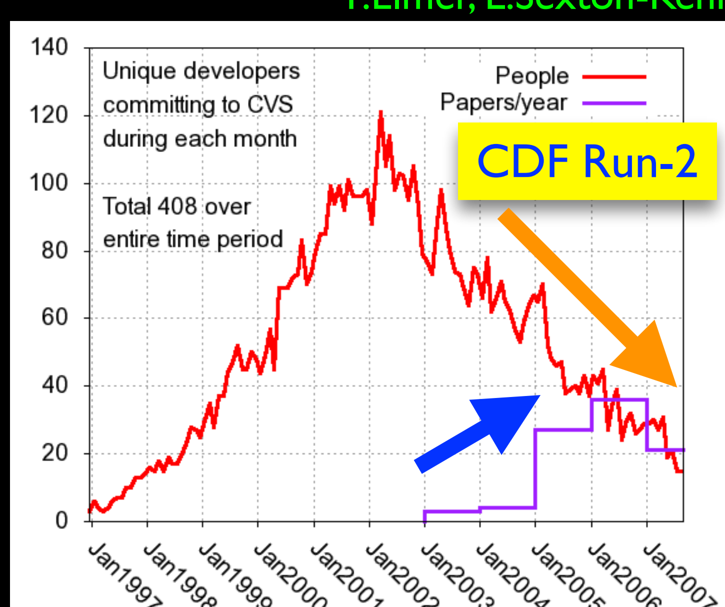
Software and Manpower

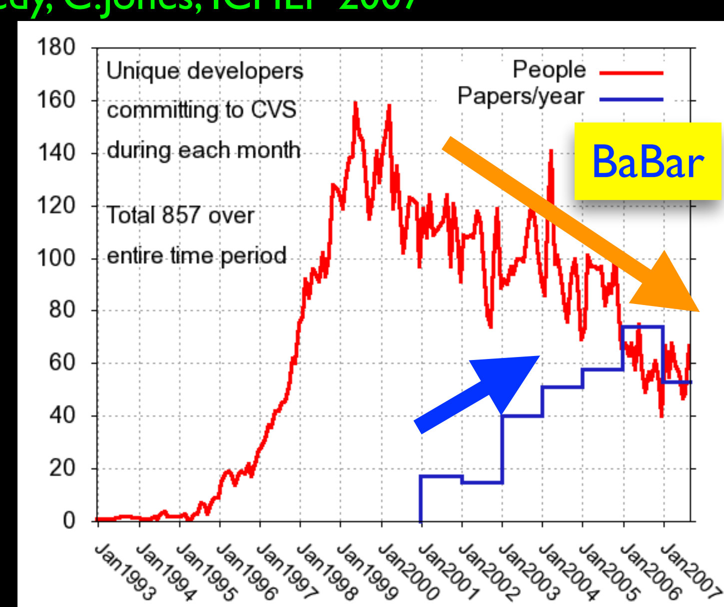
- software follows a natural life cycle
  - ➡ building up the software for an experiment
  - ➡ start of operations and data taking
  - ➡ data analysis and detector upgrades
- loss of software manpower in ATLAS/CMS
  - ➡ (mostly) students and postdocs moved on to do physics
    - same trend like in previous experiments
  - ➡ like CDF/D0 Run-2, LHC upgrade program is ambitious
    - need to find sufficient manpower to develop the software for the upgrade

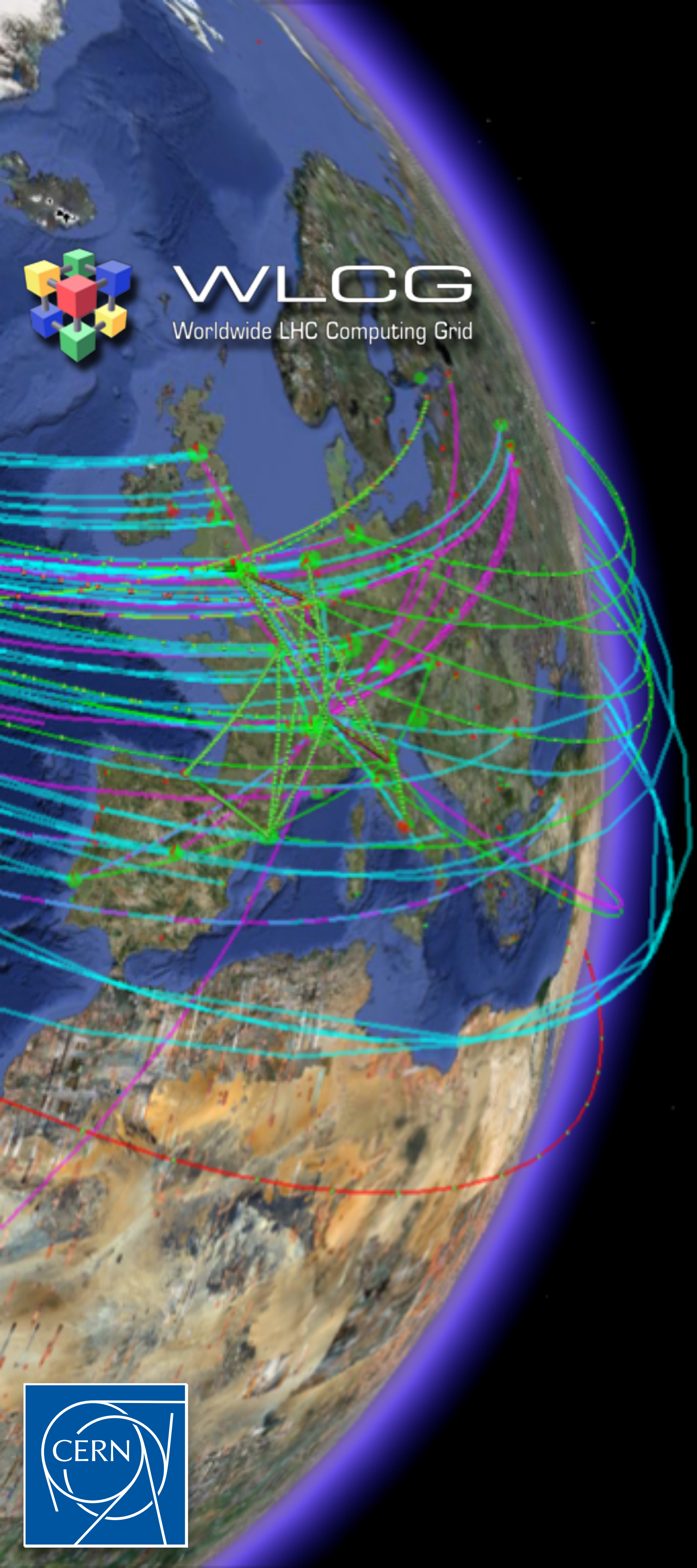


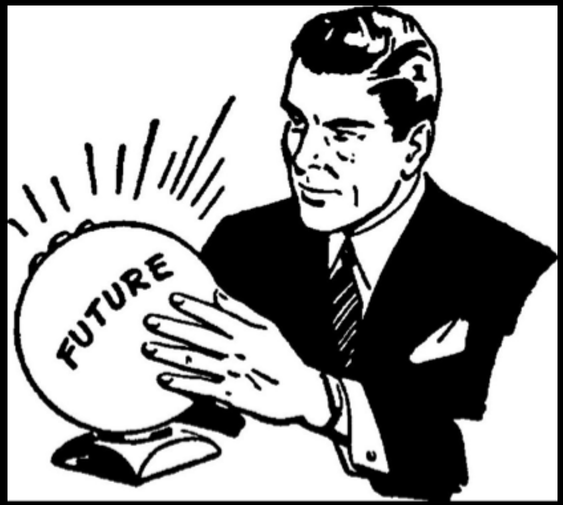


P.Elmer, L.Sexton-Kennedy, C.Jones, ICHEP 2007

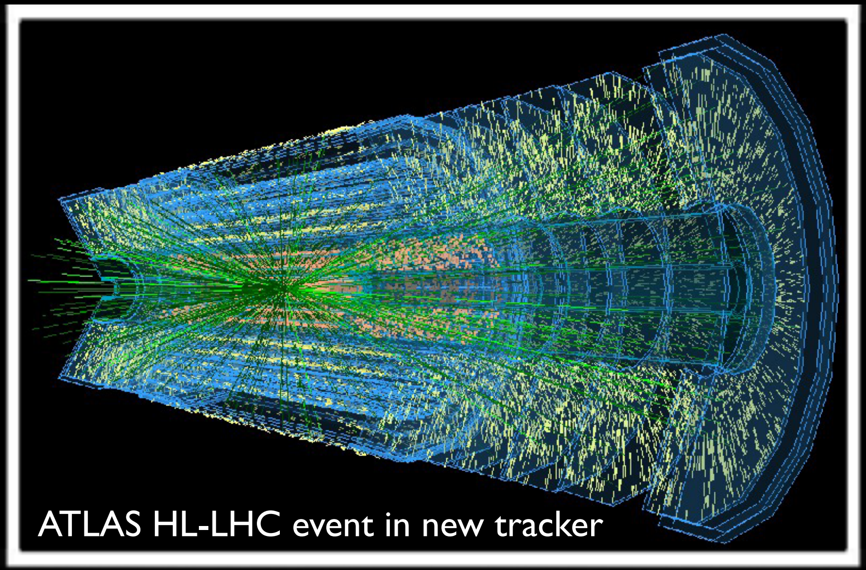








Future Offline Software Challenges

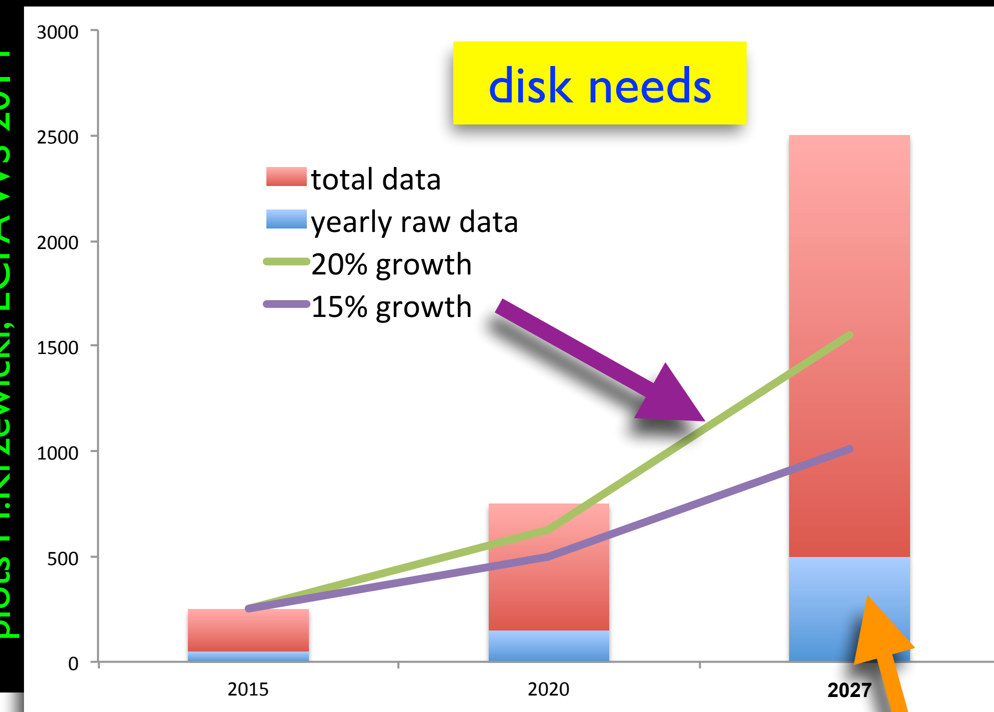

ATLAS HL-LHC event in new tracker

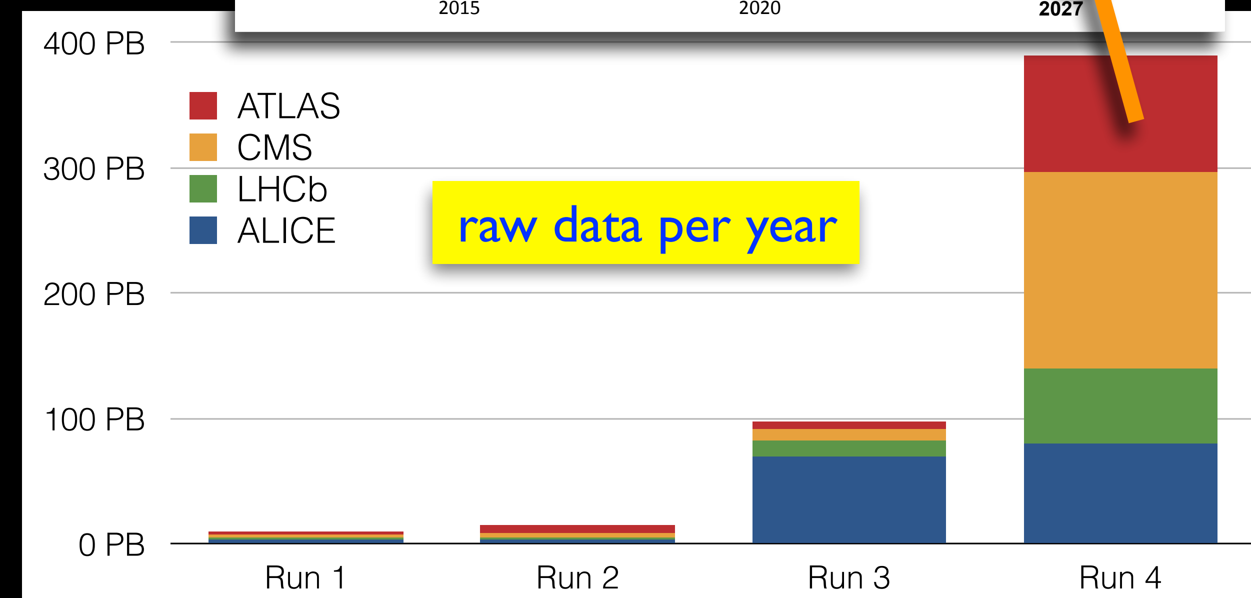
the million dollar question:
how to process HL-LHC events

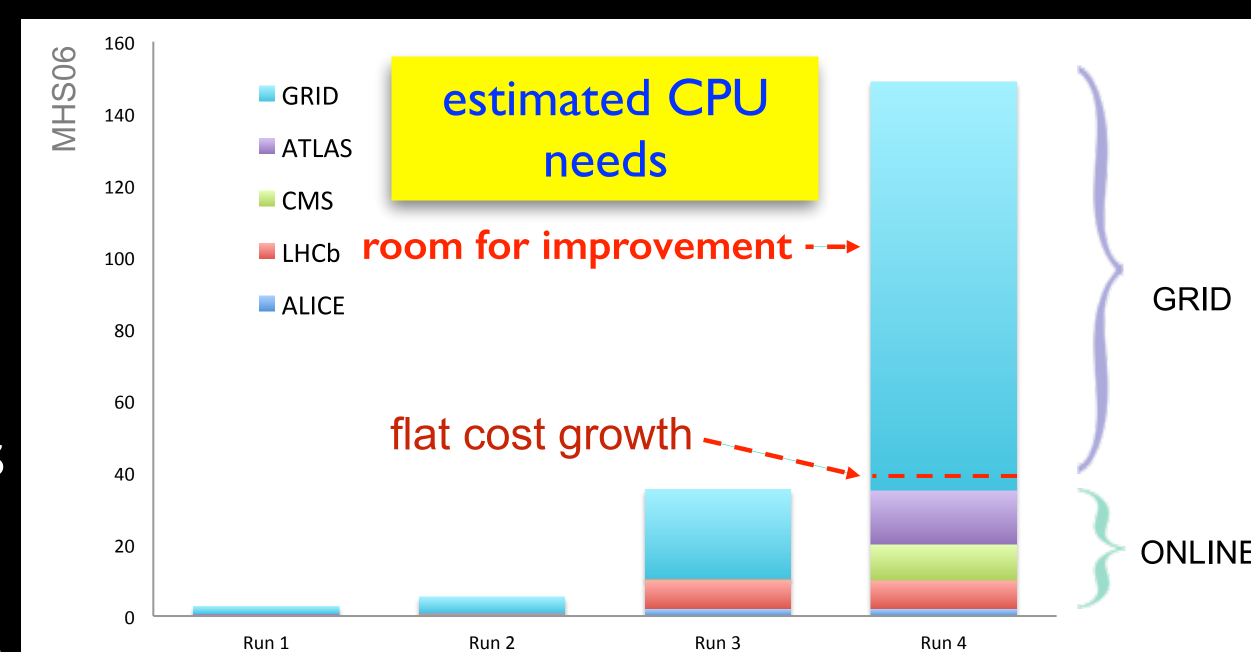
Future Computing Needs

- increase in raw data samples
  - ➡ driven by ALICE trigger-less readout
    - mostly for their online disk buffer
  - ➡ ATLAS and CMS increase of trigger rate and event size (pileup)
- total disk needs scales with raw
  - ➡ current models are above constant budget, hence need:
    - smaller data formats
    - new analysis models
    - use more tape (cheaper, continues to scale)
    - less replicas (use growing network bandwidth)
- CPU needs less certain
  - ➡ best estimates are factors above budget
    - based on current applications and models

plots M.Krzewicki, ECFA WS 2014

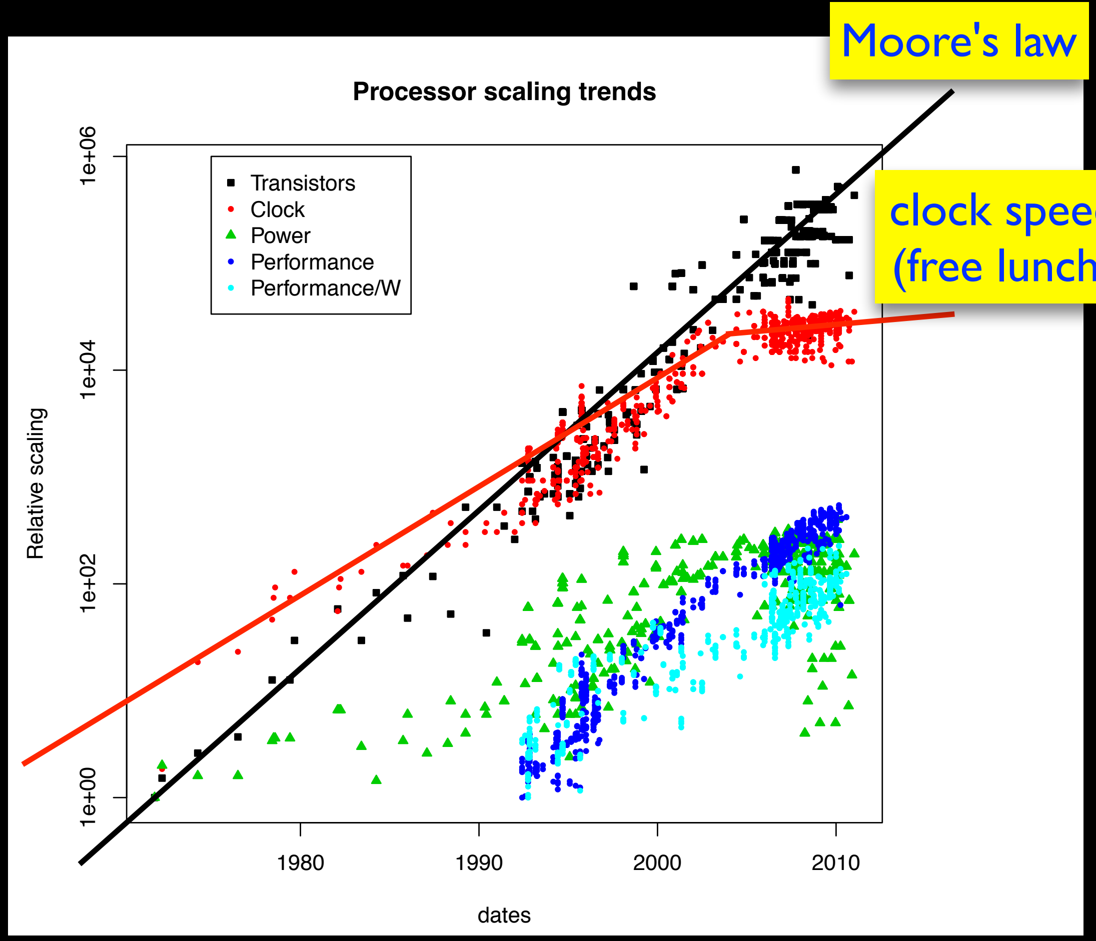




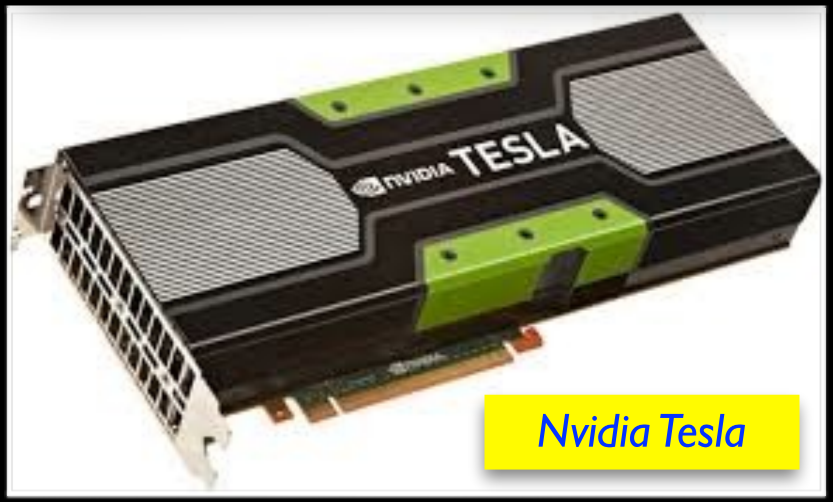


Processor Technology

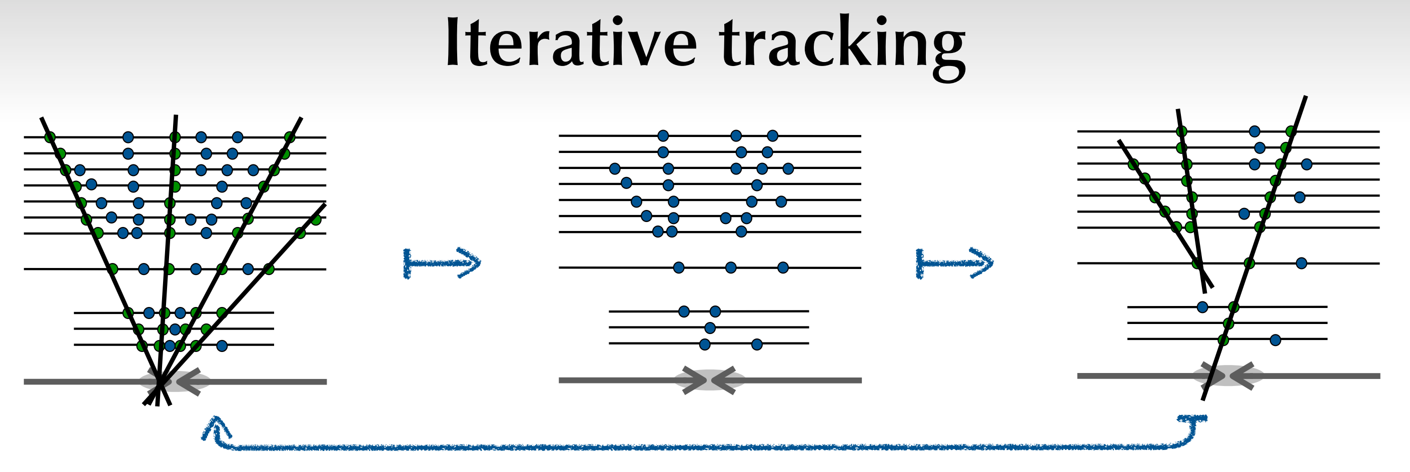
- Moore's law is still alive
  - ➡ number of transistors still doubles every 2 years
    - no free lunch, clock speed no longer increasing
  - ➡ lots of transistors looking for something to do:
    - vector registers
    - out of order execution
    - hyper threading
    - multiple cores
  - ➡ increase theoretical performance of processors
    - hard to achieve this performance with HEP applications



- many-core processors, including GPGPUs
  - ➡ e.g. Intel Xeon Phi, Nvidia Tesla
  - ➡ lots of cores with less memory
    - same for ARM or ATOM based systems
  - ➡ challenge will be to adapt HEP software
    - need to parallelise applications (multi-threading)
      (GAUDI-HIVE and CMSSW multi-threading a step in this direction)
    - change memory model for objects, more vectorisation, ...



Massively parallel Tracking ?

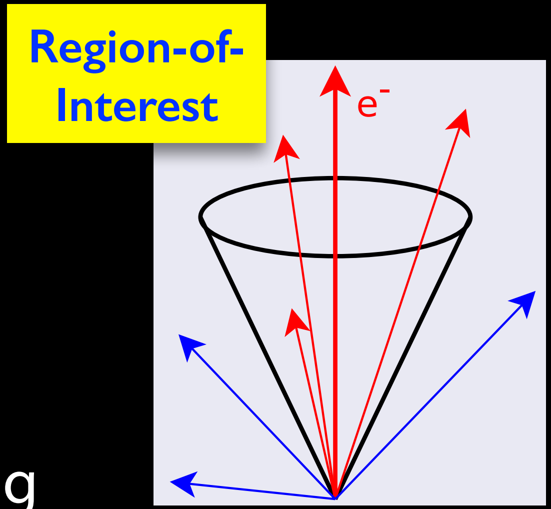


- ATLAS/CMS tracking strategy is for early rejection
  - ➡ iterative tracking: avoid combinatorial overhead as much as possible !
    - early rejection requires strategic candidate processing and hit removal
  - ➡ not a heavily parallel approach, it is a SEQUENTIAL approach !
- implications for making it massively parallel ?
  - ➡ Armdahl's law at work:

$$Time_{||} = Para / N + Seq$$

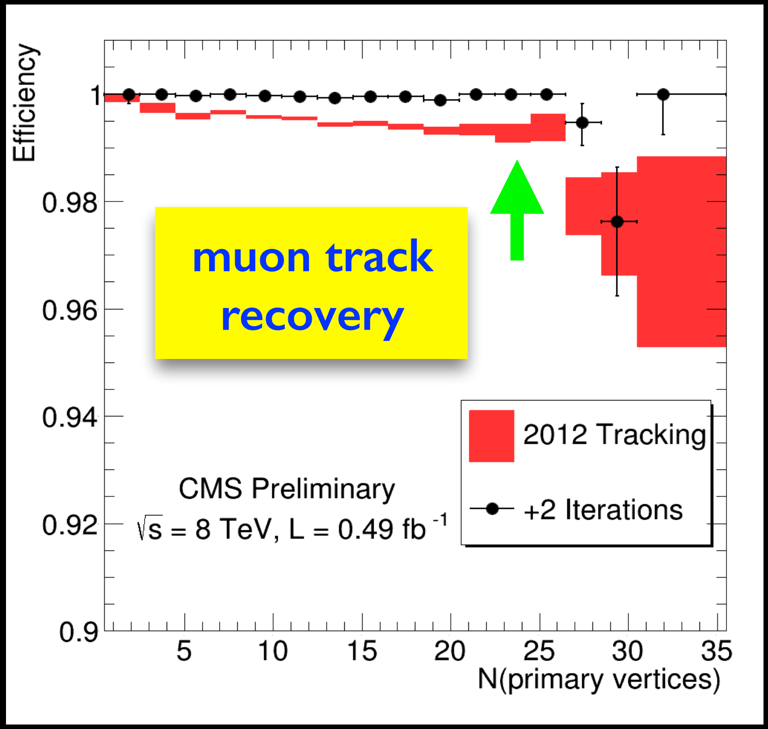
  - ➡ iterative tracking: small parallel part Para, heavy on sequential Seq
    - hence, if we want to gain by a large N threads, we need to reduce Seq
- CMS study: run combinatorial filter in parallel for seeds
  - ➡ find compromise on early rejection, but still limit combinatorial overhead
    - as a result, one spends somewhat more CPU, main gain is in memory
  - ➡ promising if one uses additional processing power that otherwise would not be usable (many core processors) or if latency is the main issue (trigger)

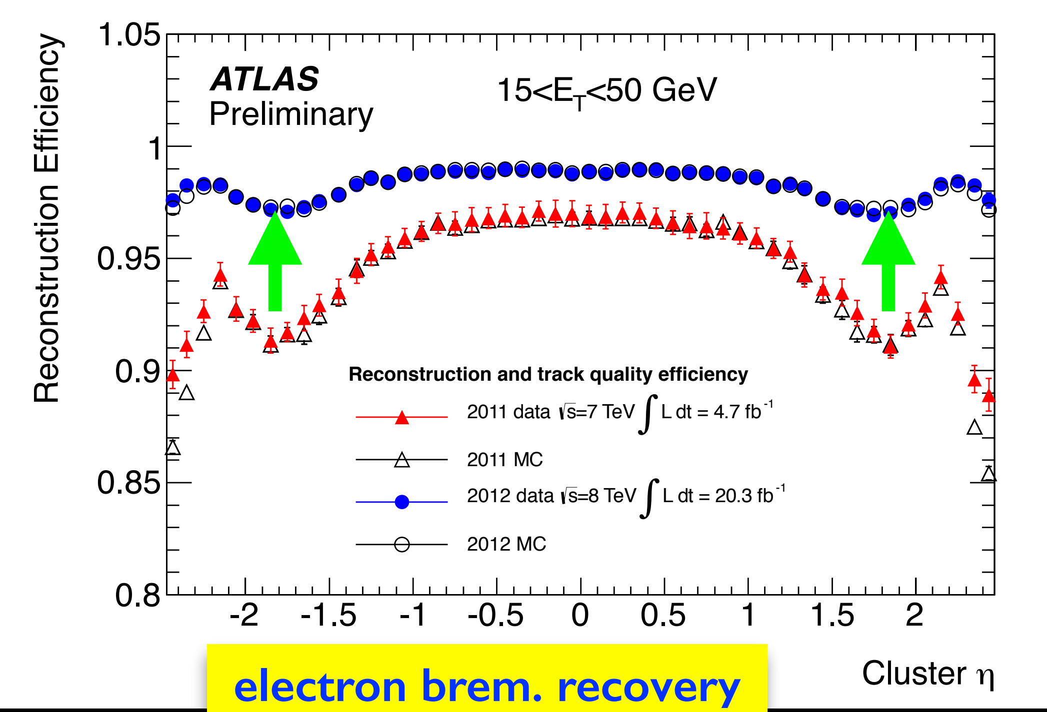
Tracking Algorithms for High Pileup



- alternative tracking techniques for parallelisation ?
  - ➡ CMS investigated using Hough Transforms, limited by multiple scattering

- tracking according to physics needs ?
  - ➡ idea: run different tracking inside/outside Region-of-Interest
    - best possible tracking for signal event or region
    - faster, approximate tracking on pileup and underlying event
      (extreme: truth guided tracking on MC to avoid pattern overhead)
  - ➡ experiments already started doing this in Run-1 !
    - CMS runs tracking passes to recover efficiency for muons
    - ATLAS runs brem. recovery for tracks pointing to EM clusters
  - ➡ and for Run-2
    - ATLAS regional tracking for photon conversions
    - both experiments have dedicated tracking in jets



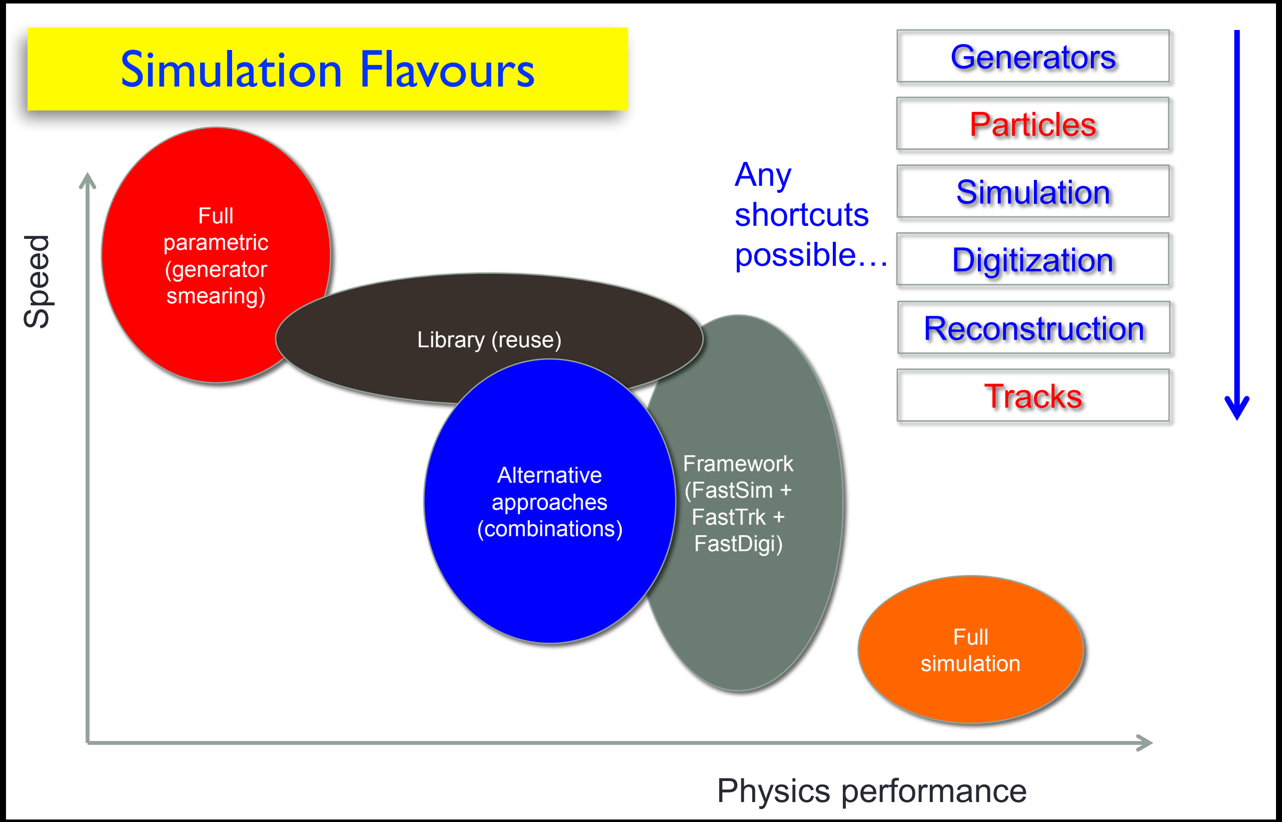


- need more R&D on future algorithms

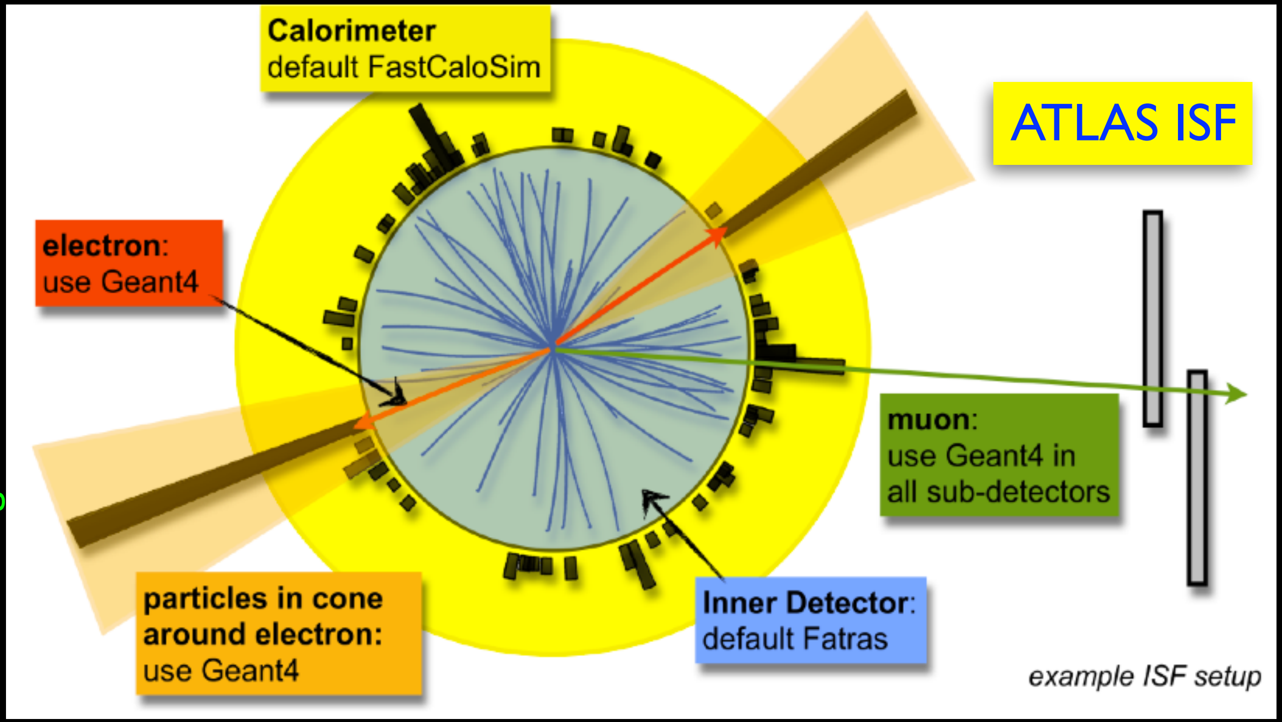
Detector Simulation

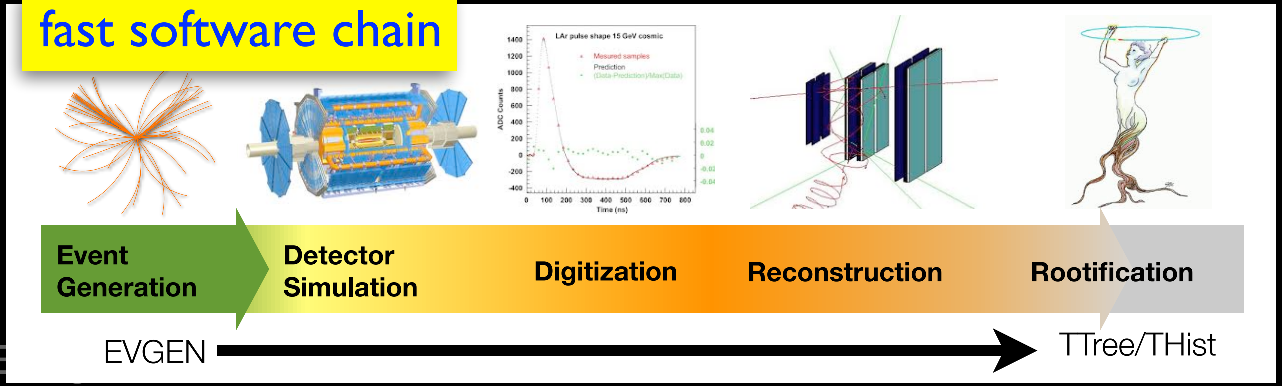
- simulation limited by CPU
  - ➡ avoid MC limiting physics precision
  - ➡ need to increase GRID "MC luminosity"
- major software technology developments in simulation
  - ➡ Geant 4.10 introduces multi-threading support
  - ➡ Geant V redesign to explore vectorisation
- ATLAS Integrated Simulation Framework (ISF)
  - ➡ mixes fast and full sim. in one event
    - spend time on important event aspects
  - ➡ towards complete fast software chain
    - avoid digit. and reco. bottleneck
    - directly produce analysis formats (disk)

A.Gheata, ACAT 2014



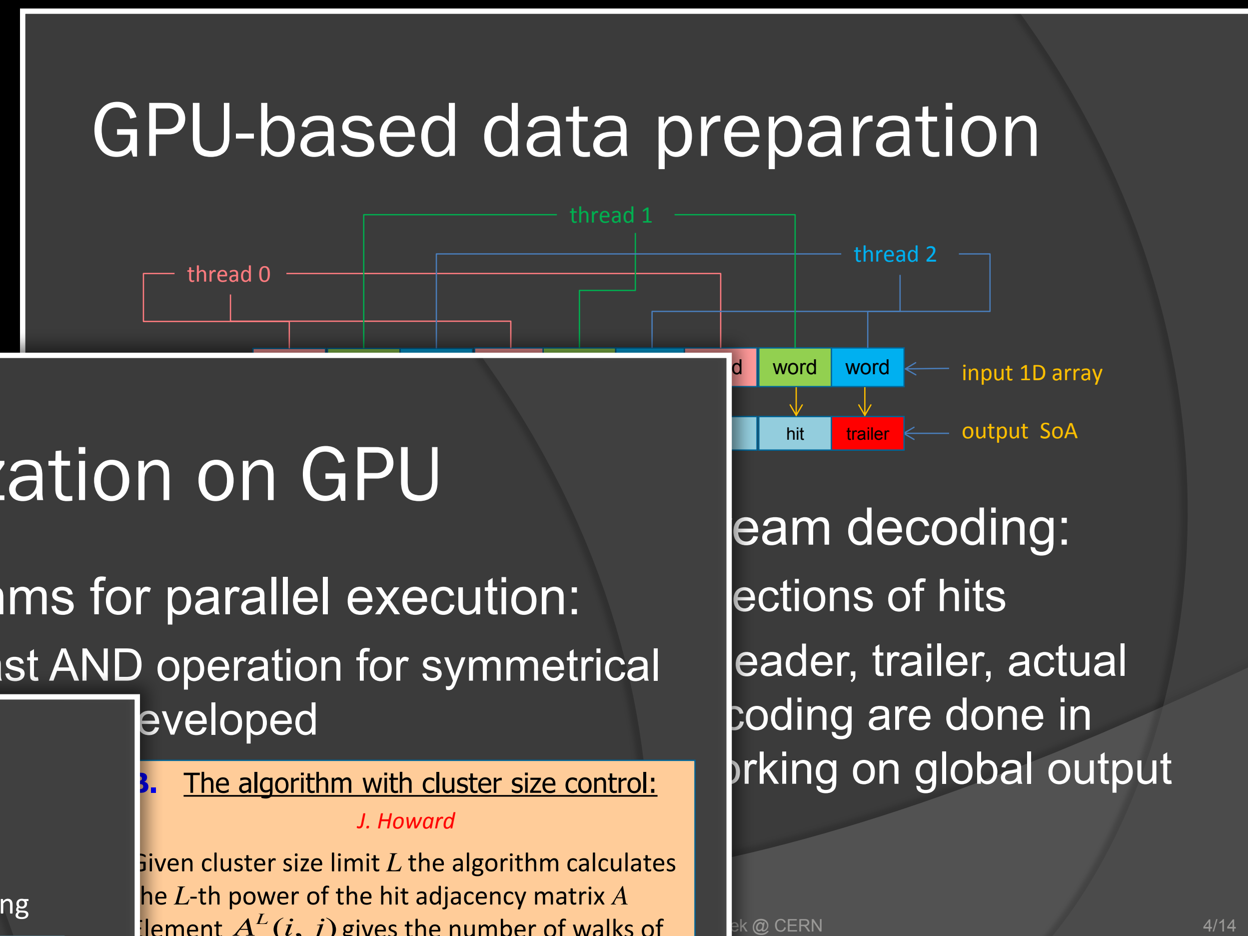
A.Salzburger

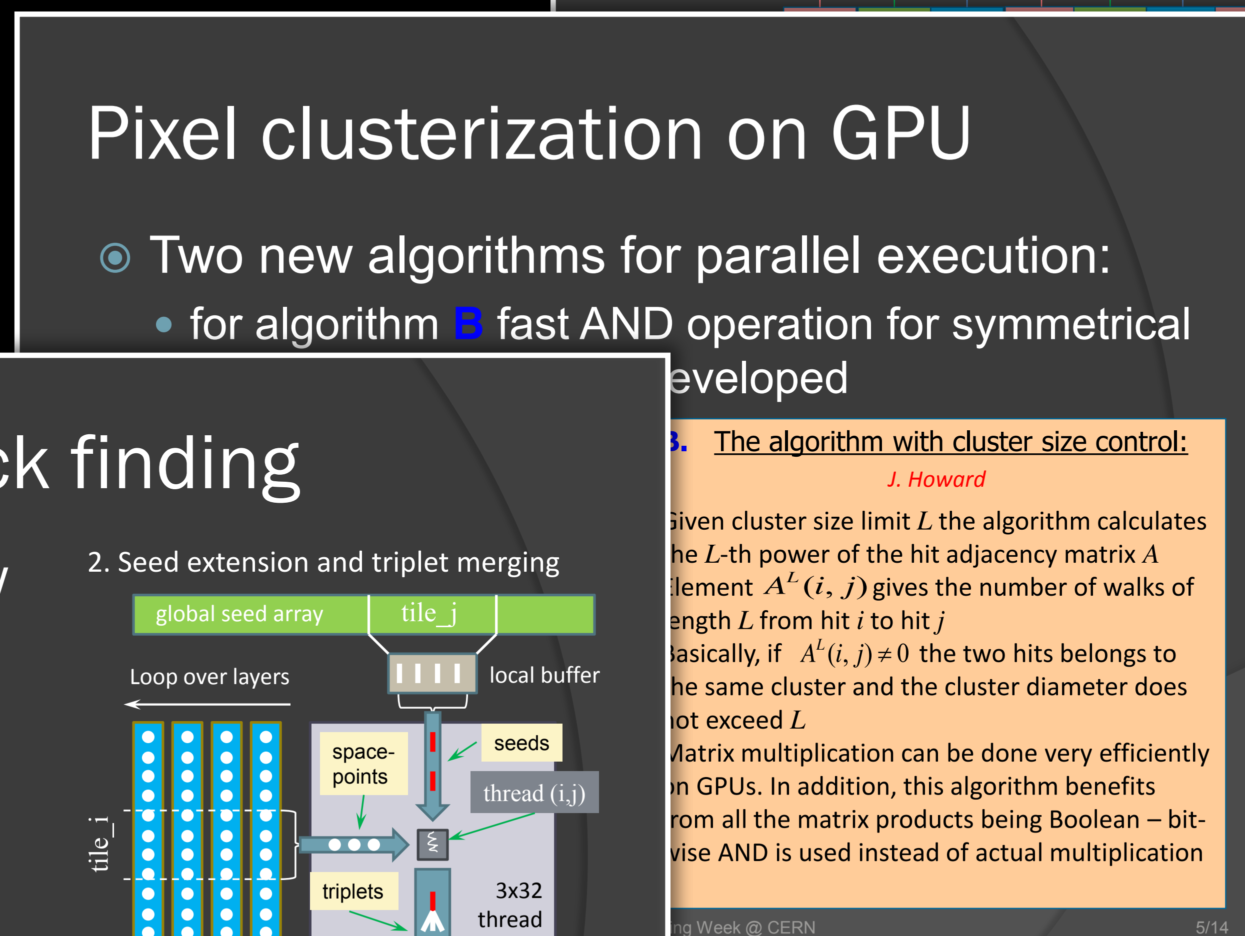


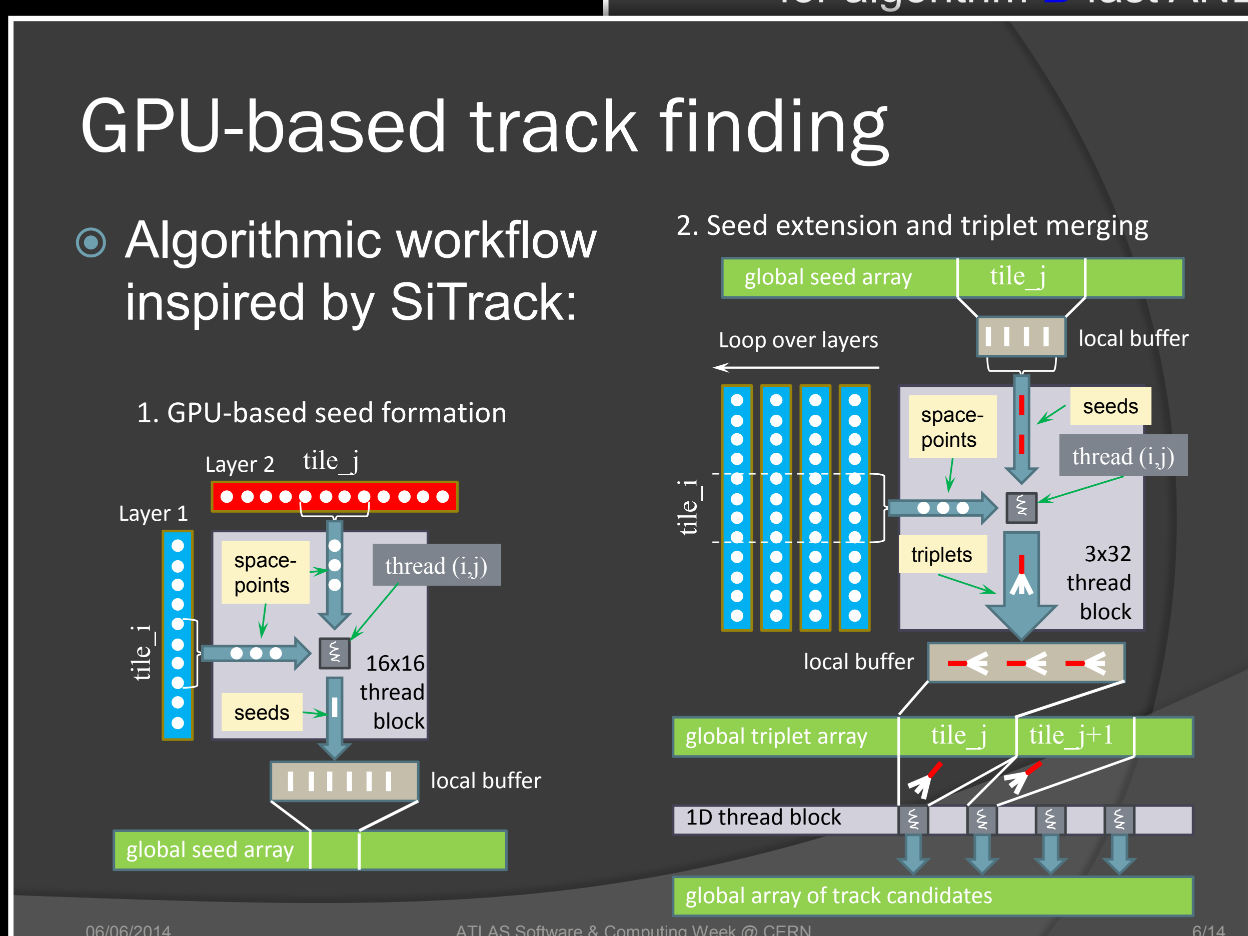


ATLAS Level-2 GPU Tracking Prototype

- as an example for a complete tracking chain on GPUs
  - ➡ from raw to tracks
  - ➡ currently many such R&D activities in CMS and ATLAS





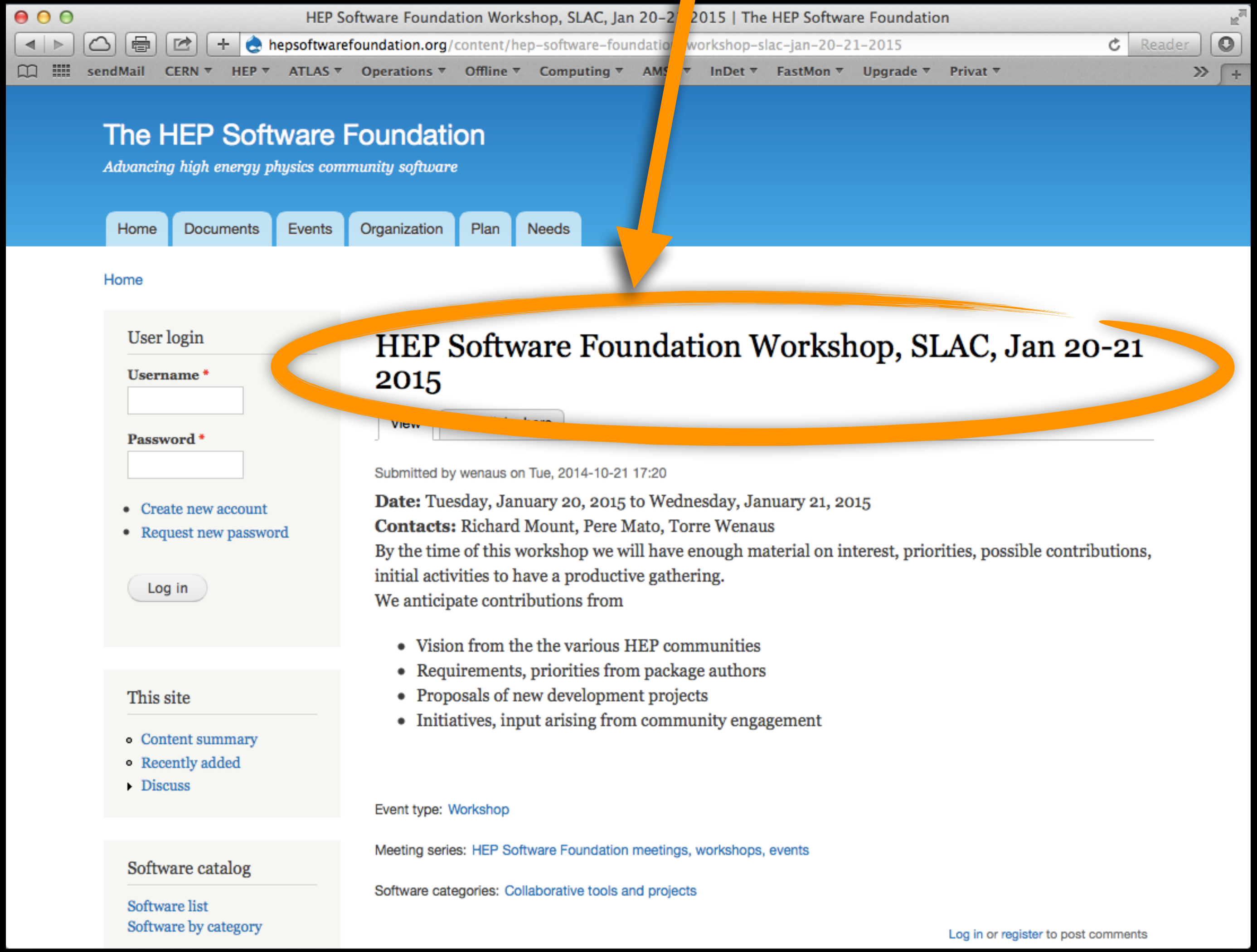


- ➡ significant speedup compared to running same chain on CPU
- ➡ CUDA vs openCL, development and maintenance cost ?

HEP Software Foundation

recent workshop, see as well CHEP

http://hepsoftwarefoundation.org



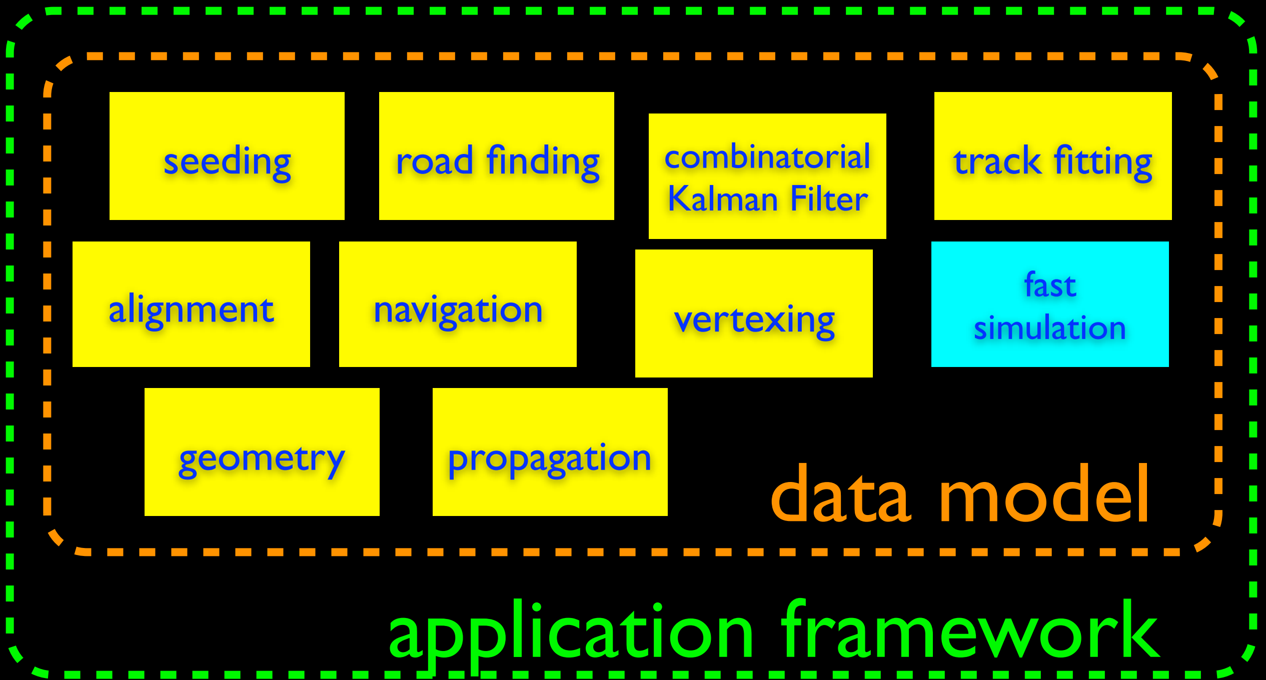
- initiative to raise profile of HEP software projects
  - ➡ building upon existing and previous initiatives
    - hepfroge.org
    - Concurrency Forum
    - (less known) US HEP Forum for Computational Excellence
    - previous LCG Application Area
  - ➡ as well, existing HEP SW projects
    - Geant4, Root, ...
  - ➡ hopefully as well GRID software
- foundation as a bottom-up approach
  - ➡ invite participation in projects across experiments and collaboration beyond HEP
  - ➡ hope to achieve synergies and bundle expertise on crucial technology developments
  - ➡ may host tracking (reconstruction) algorithm forum to foster collaboration

Common Algorithmic Software ?

from Graeme's talk

- examples for common algorithmic software
  - ➡ FastJet - de-facto standard for jet finding, distribution as part of LCG externals
  - ➡ TMVA, RooFit/RooStat, HistFitter, BAT - statistics and multivariate analysis
  - ➡ AIDA tracking - primarily targeting ILC / FCC
  - ➡ genfit - an implementation of standard track fitting techniques (Belle-II)
  - ➡ CMS vertexing suite - package of standard vertexing codes (CMS, Belle-II,...)
  - ➡ VDT, SMatrix, Eigen - vector algebra and math libs

- a real integrated common tracking implementation ?
  - ➡ AIDA is the one aiming at this ...
  - ➡ integration means picking a data model
    - determines Jacobians in math formulars
  - ➡ integration means framework interfaces
  - ➡ best physics performance ?
    - pattern strategy depends on experiment
  - ➡ manpower on AIDA vs (e.g.) CMS/ATLAS ?
  - ➡ discussion in ATLAS:
    - make tracking/vertexing suite public ? (for FCC)

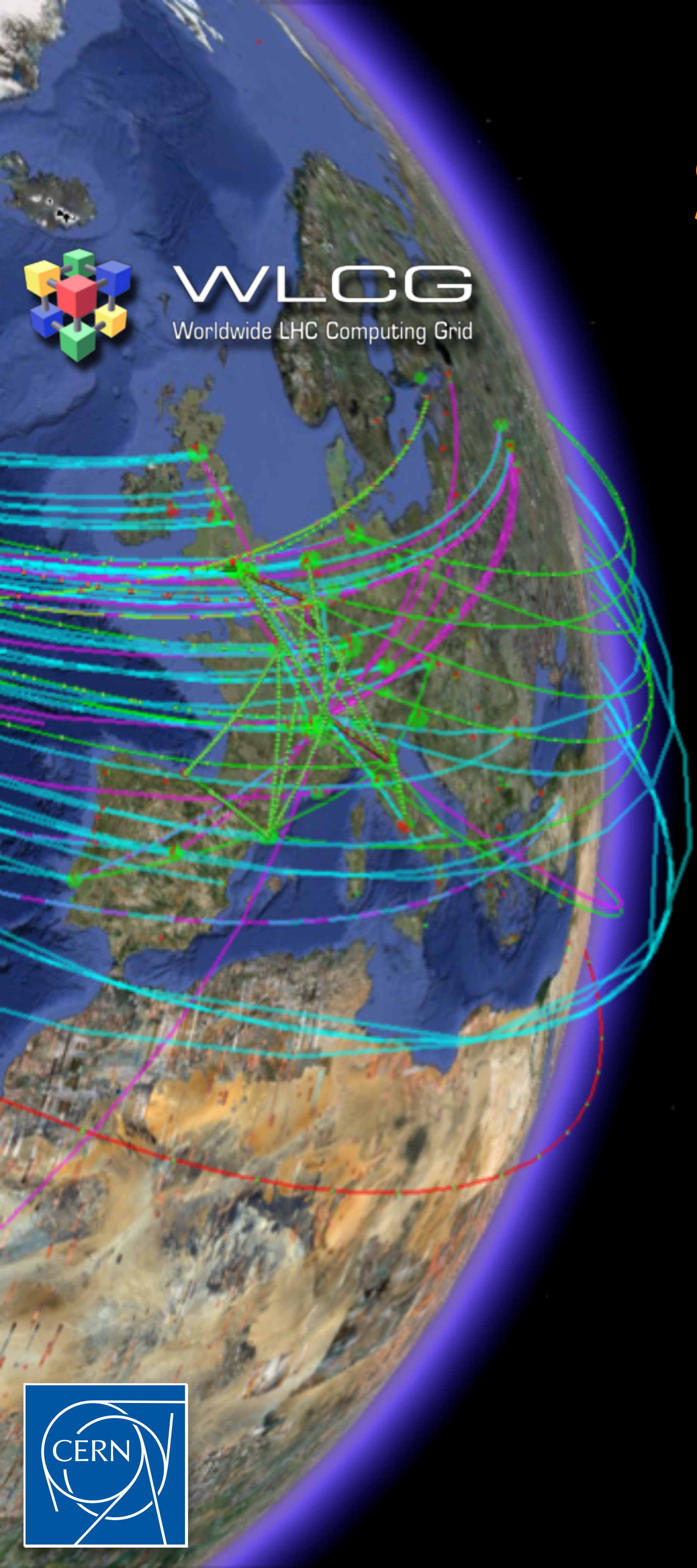


Building a "Forum" and a Community ?

- some obvious observations:
  - ➡ we need to make workshops like Connecting the Dots more regular
    - yearly like BOOST workshops ? every 18 months like CHEP and ACAT ?
  - ➡ we need to think about dedicated schools to teach algorithms to students
    - we need to invest in future experts (and give them career perspectives)
  - ➡ do we need some more regular forum alongside the Concurrency Forum ?
    - need will grow once we have more common developments to discuss
    - how often shall we do such a meeting initially ?

- focus on exchange of ideas, techniques, best practices ... ?
  - ➡ at Connecting the Dots meeting, not much enthusiasm across all experiments (but maybe FCC) to migrate to something like a common algorithm stack
  - ➡ common software projects may grow naturally out of needs we may identify

- created as well a generic HSF mailing list:
  http://hepsoftwarefoundation.org/content/reconstruction-algorithms-forum
  - ➡ to be used to bring together initiatives like Connecting the Dots for tracking and the communities working on boosted object reconstruction and alike



Summary

- building the LHC software and tracking
  - ➡ took almost a decade to master the challenge
  - ➡ resulted in sophisticated software stags for the experiments
    - including highly optimised track reconstruction
- excellent performance during Run-1
  - ➡ full benefit from careful preparation
  - ➡ good quality data and description in simulation
    - highly instrumental to fully explore physics reach, including the role of software in the Higgs discovery
- shutdown preparations for Run-2
  - ➡ even higher pileup and limited computing resources
  - ➡ first round of software upgrades to mitigate effects
- many more challenges ahead
  - ➡ Phase-1 and Phase-2 detector upgrades
  - ➡ IT technologies are changing dramatically