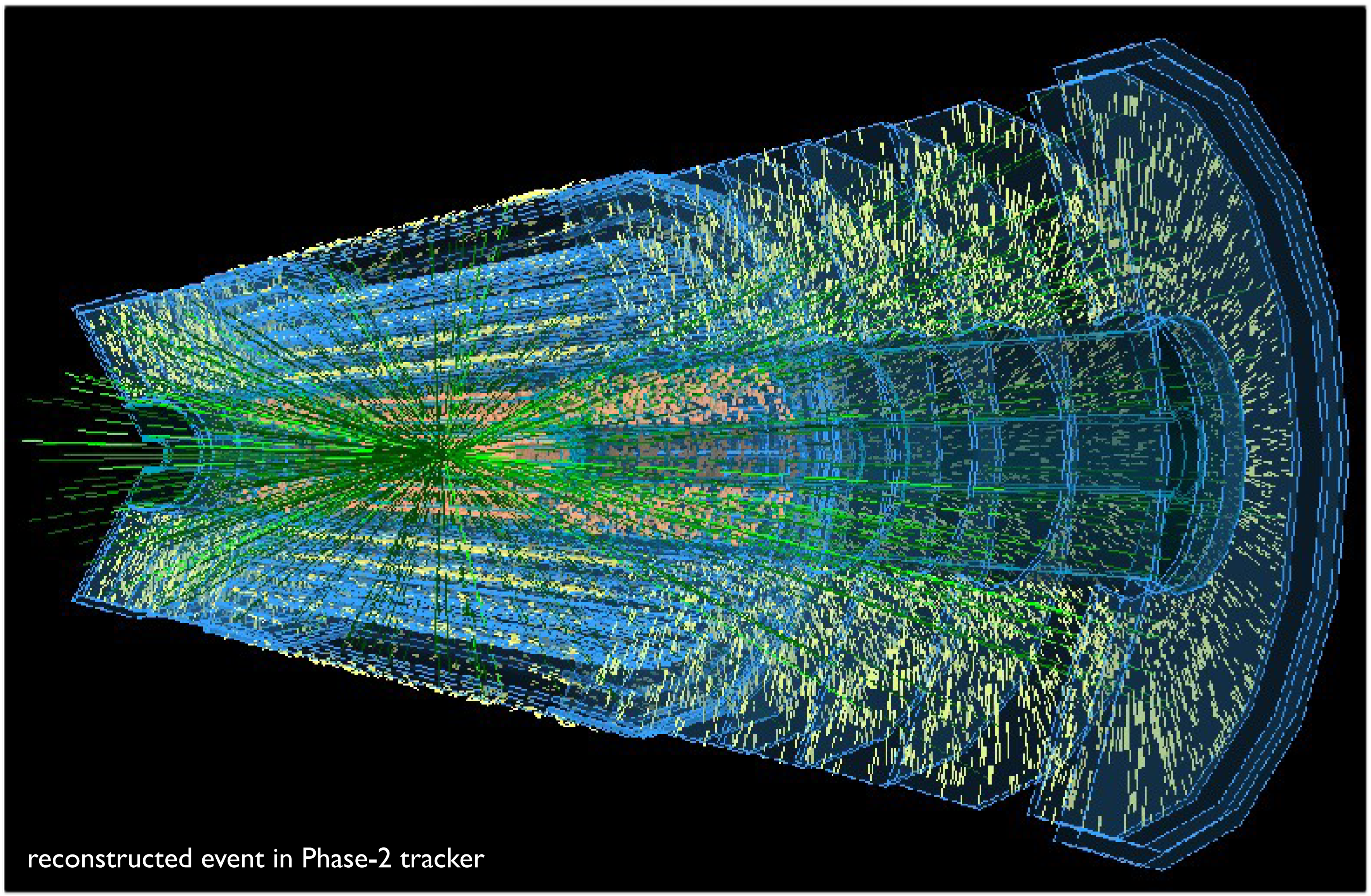
LS1 Activities of the ATLAS Software Project

Markus Elsing

report at the PH-SFT group meeting

December 9th, 2013

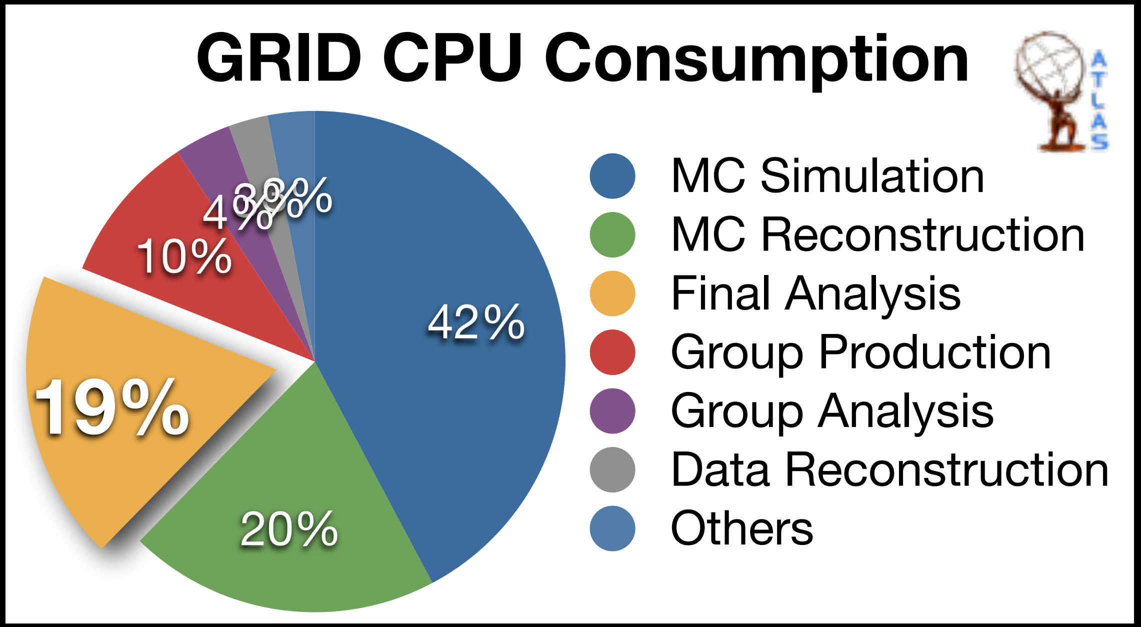


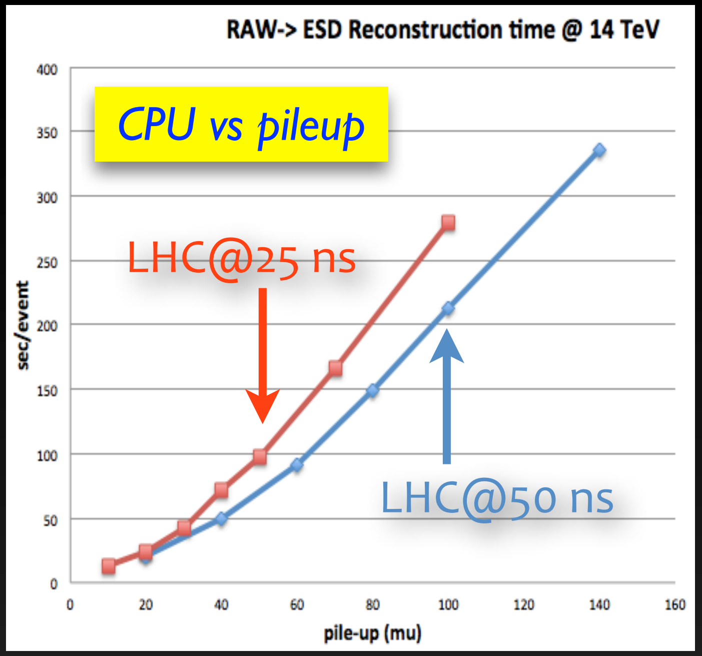
reconstructed event in Phase-2 tracker

CERN

Introduction and Outline

- the challenges
  - ➡ pileup drives resource needs
    - not only in Tier-0
  - ➡ GRID "luminosity" is limited
    - full simulation is costly
  - ➡ physics requires to increase rate
    - Run-2 data taking rate 1kHz (?)
  - ➡ technologies are evolving fast
    - software needs to follow
  - ➡ support detector upgrade studies
    - not covered in this talk

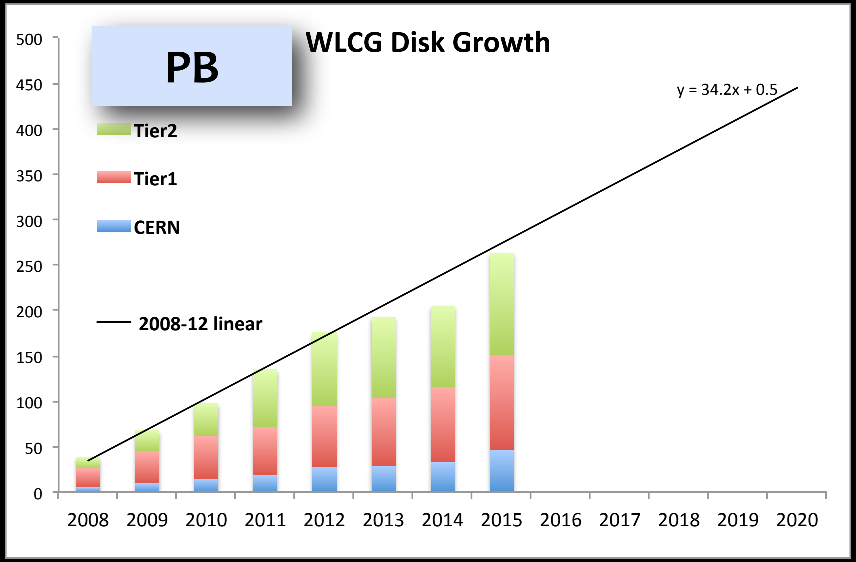


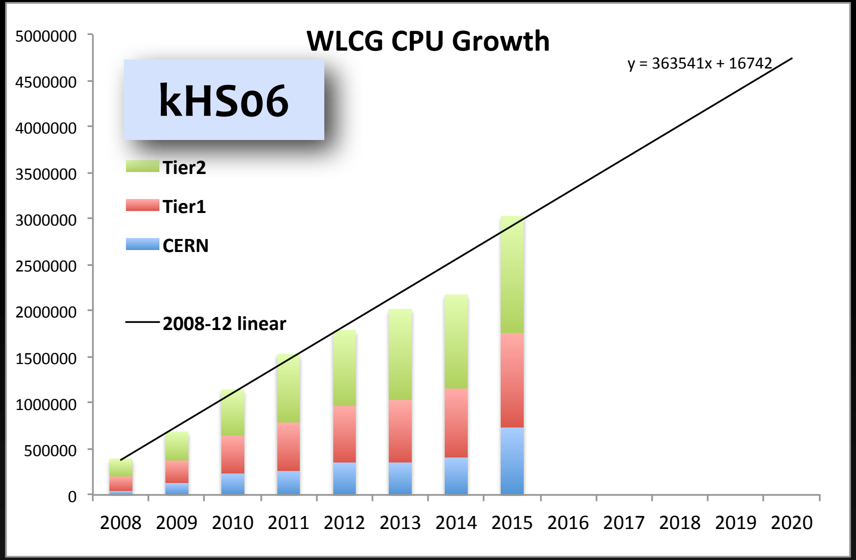


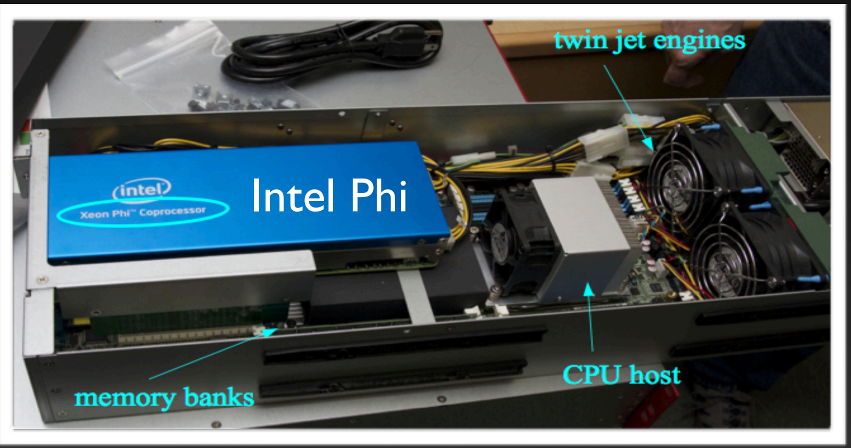
- outline of the talk
  1. work of Future Software Technologies Forum (FSTF)
  2. algorithmic improvements
  3. the Integrated Simulation Framework (ISF) for Run-2
  4. new Analysis Model for Run-2
  5. goals and plans for Data Challenge-14 (DC-14)
  6. completion of LS1 program for restart of data taking

Evolution of WLCG Resources

- upgrades of existing centers
  - ➡ additional resources expected mainly from advancements in technology (CPU or disk)
  - ➡ will not match additional needs in coming years
- todays infrastructure
  - ➡ x86 based, 2-3 GB per core, commodity CPU servers
  - ➡ applications running "event" parallel on separate cores
  - ➡ jobs are send to the data to avoid transfers
- technology is evolving fast
  - ➡ network bandwidth fastest growing resource
    - data transfer to remote jobs is less of a problem
    - strict Monarc Model no longer necessary
    - flexible data placement with data popularity driven replication, remote I/O and storage federations
  - ➡ modern processors: vectorization of the applications and optimization for data locality (avoid cache misses)
  - ➡ "many core" processors like Intel Phi (MIC) or GPGPUs
    - much less memory per core !



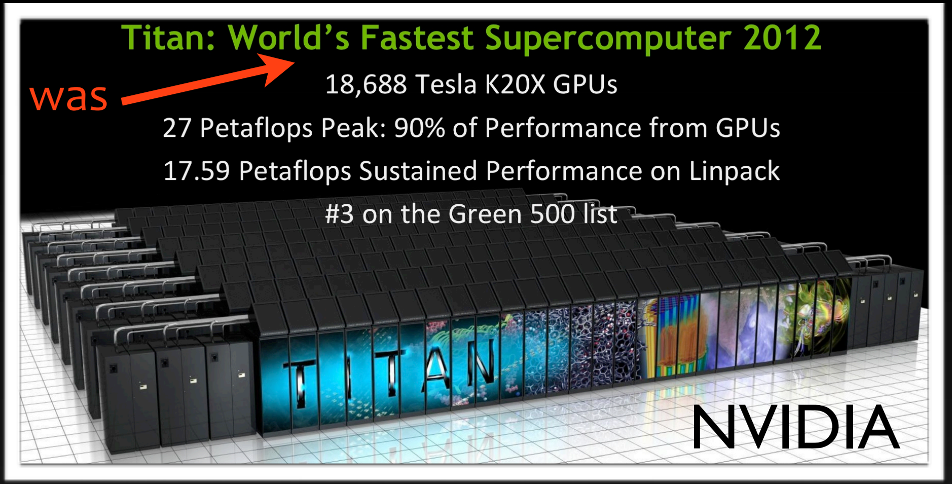




High Performance Computing in ATLAS

- infrastructure is getting heterogeneous
  - ➡ mostly opportunistic usage of additional resources
    - commercial Cloud providers (i.e. Google, Amazon)
    - free CPU in High Performance Computing centers
  - ➡ big HPC centers outperform WLCG in CPU
    - X86, BlueGene, NVIDIA GPUs, ARM, ...
  - ➡ GRID (ARC Middleware) or Cloud (OpenStack) interface

- suitable applications
  - ➡ CPU resource hungry with low data throughput
    - physics generators or detector simulation
  - ➡ X86 based systems
    - small overhead to migrate applications
  - ➡ GPU based systems
    - complete rewrite necessary (so far) or dedicated code



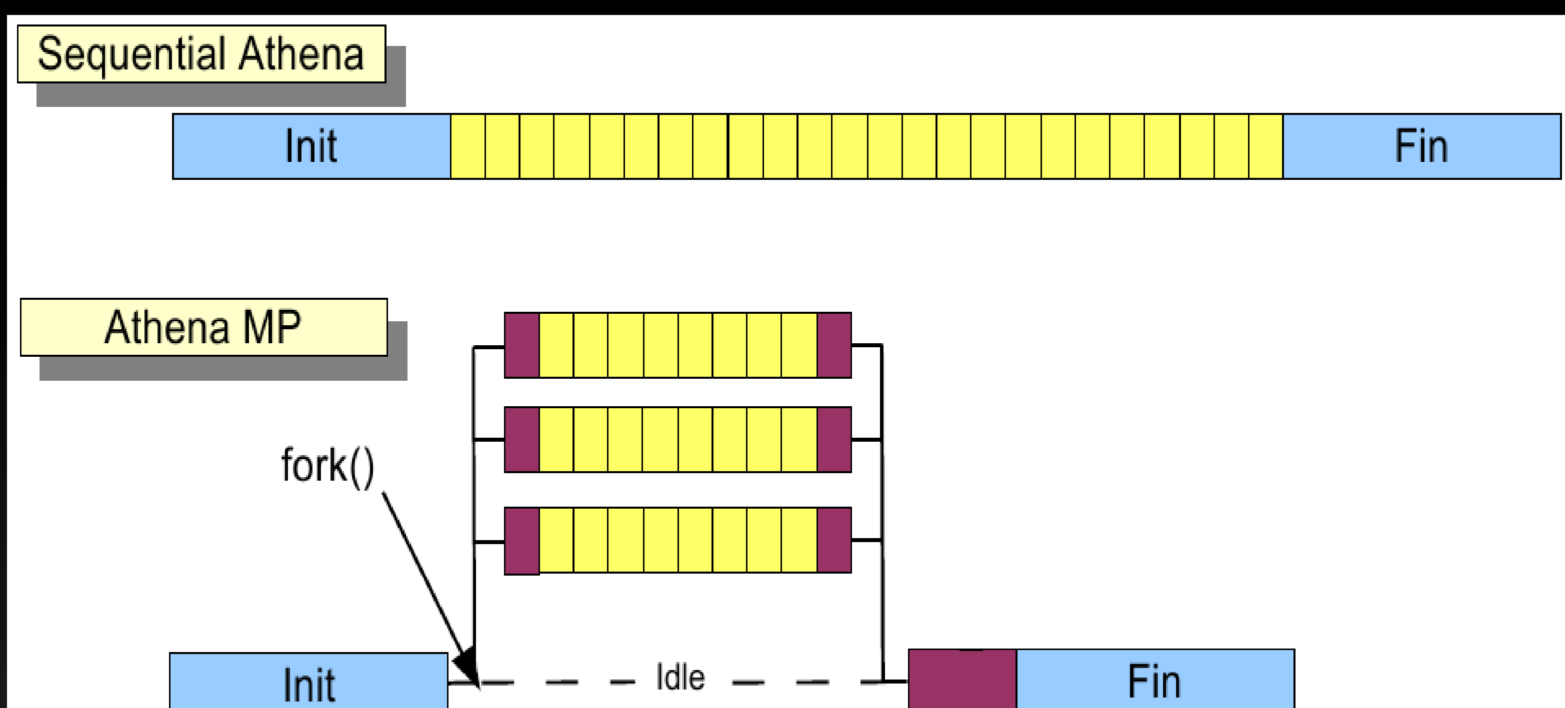
- ATLAS (ADC) working group to evaluate HPC opportunities
  - ➡ first successful test productions on commercial clouds and HPC clusters

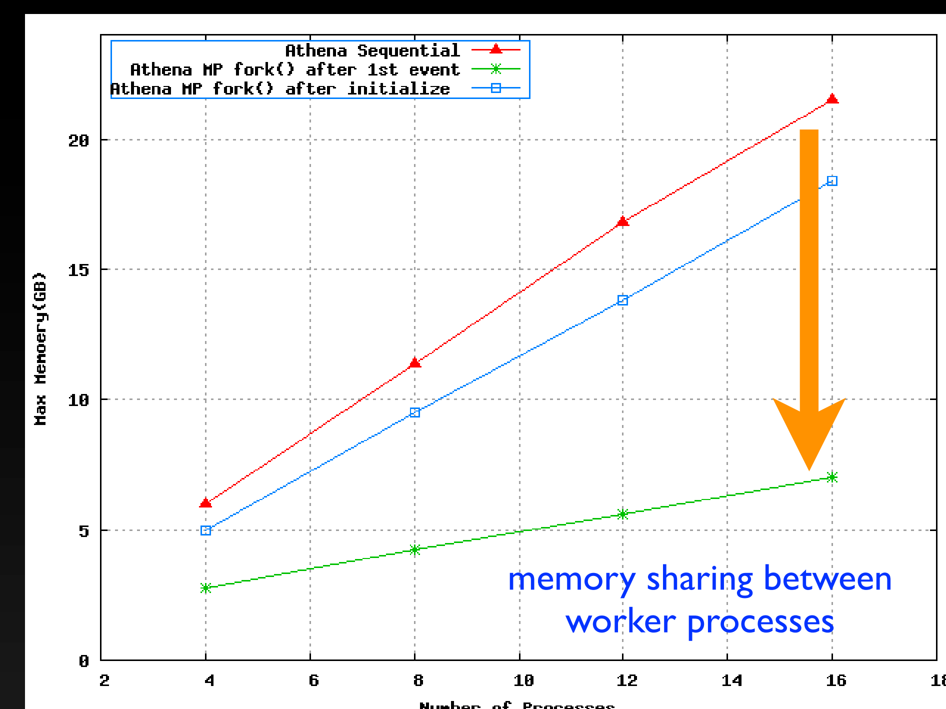
Future Software Technologies Forum

- coordinates all technology R&D efforts in ATLAS
  - ➡ drives ATLAS developments on vectorization and parallel programming
    - examples: AthenaMP, AthenaHive, Eigen, VDT/libimf, ...
    - studies of compilers, allocators, auto-vectorization, ...
    - explore new languages (ISPC, cilk+, openMP4 etc)
  - ➡ forum for R&D on GPGPUs and other co-processors
    - algorithm development, share experience, identify successful strategies
    - get experience on ARM and Intel Phi
  - ➡ pool of experienced programmers
    - educating development community
  - ➡ software optimization with profiling tools (together with PMB)
    - tools like: perfmon, gperftools, GoODA
    - code optimization and identification of hot spots in ATLAS applications
    - examples: b-field access, z-finder in HLT, optimizing neural-nets
- liaison with Concurrency Forum and OpenLab
  - ➡ integration of ATLAS efforts in LHC wide activities

AthenaMP (Multi-Process)

- not a new development, but not yet in production
  - ➡ event parallel processing, aim to share memory (see GaudiMP)
  - ➡ successful simulation, digitization and reconstruction tests recently
    - still issues with I/O, e.g. on EOS
  - ➡ goal is to put AthenaMP in full production by ~ this summer



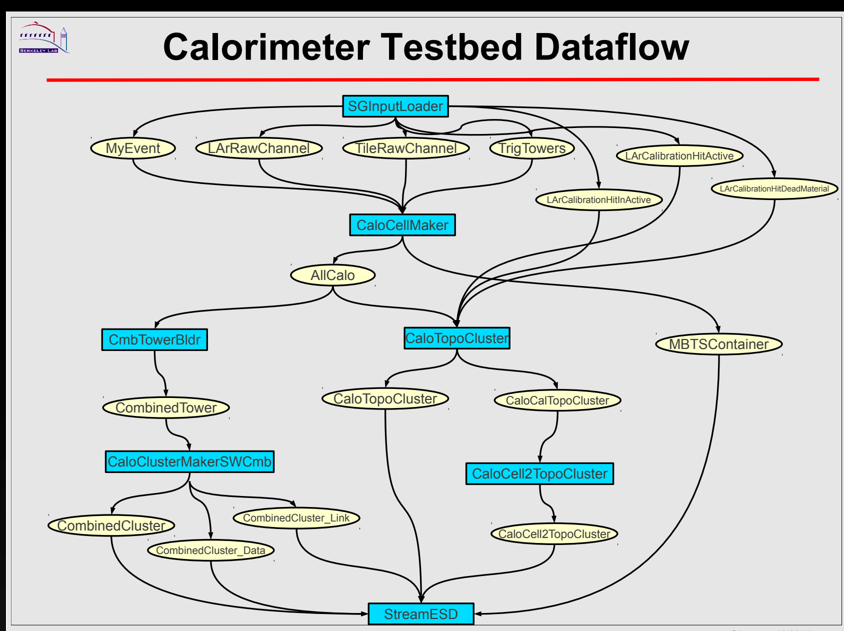


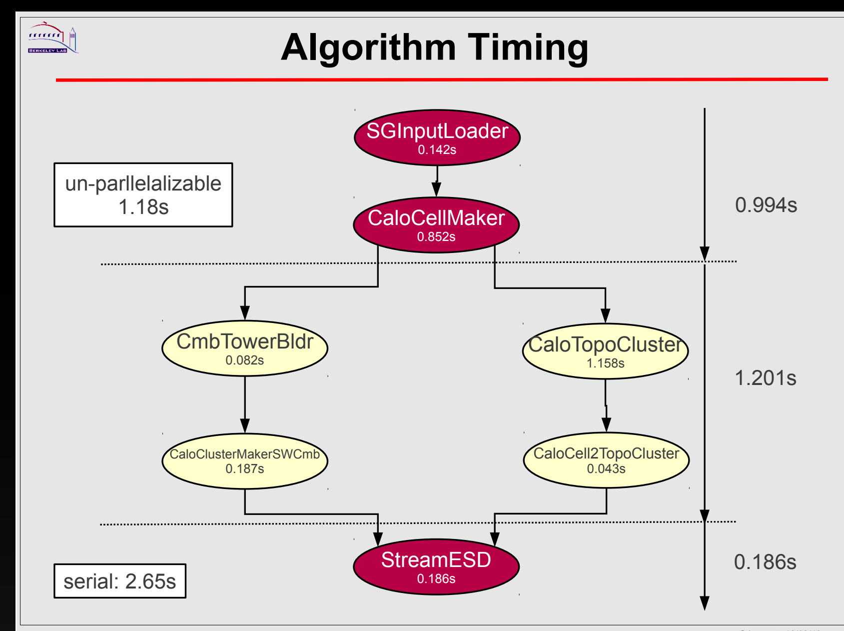
- next version of AthenaMP improves GRID integration
  - ➡ including new "event service" I/O model in ProdSys-2

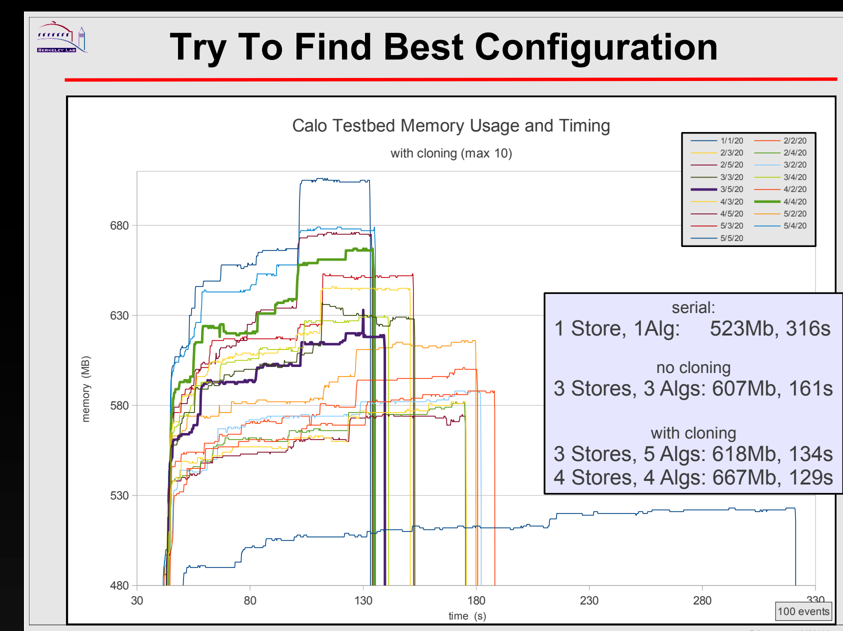
C.Leggett

AthenaHive Testbed

- based on GaudiHive project
  - ➡ model is multi-threading at the algorithm level (DAG)
  - ➡ demonstrator study using calorimeter reconstruction
    - factor 3.3 speedup w.r.t. sequential (on more cores), 28% more memory







- still a long way to go
  - ➡ all framework services need to support multi-threading
  - ➡ making ATLAS services, tools and algorithms thread safe, adapt configuration
  - ➡ in the demonstrator we see limits of DAG (Amdahl's law at play)
    - work on Hive necessary step towards final multi-threading goal
    - need parallelism at all levels (especially for tracking algorithms)

Current Tracking Software Chain

- tracking is resource driver in reconstruction
  - ➡ current software optimized for early rejection
    - 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 !
  - ➡ good scaling with pileup (factor 6-8 for 4 times pileup) - still catastrophic

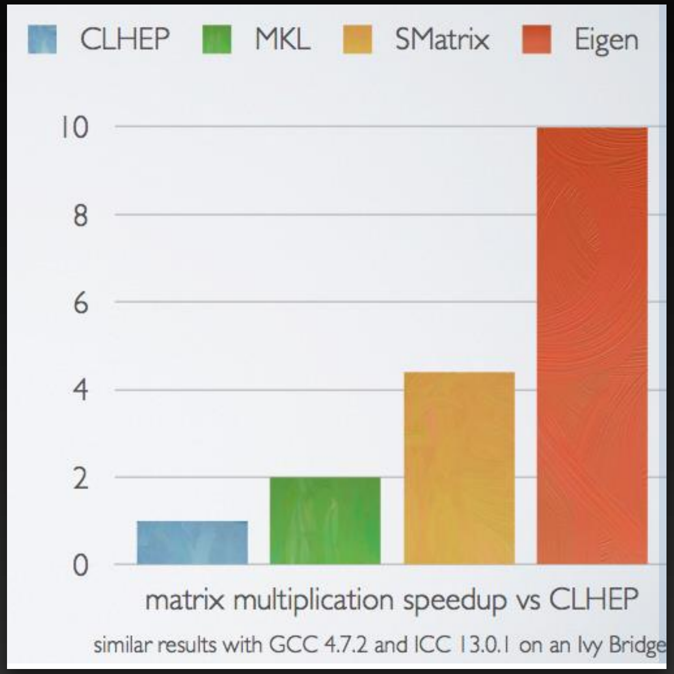
- implications for making it heavily parallel ?
  - ➡ Amdahl's law at work:

$$t_{||} = p/n + s$$

    - current strategy has small parallel part P, while it is heavy on sequential S
  - ➡ hence: if we want to gain by a large N threads, we need to reduce S
    - compromise on early rejection, which means more combinatorial overhead
    - as a result, we will spend more CPU if we go parallel
  - ➡ makes only sense if we use additional processing power that otherwise would not be usable ! (many core processors)

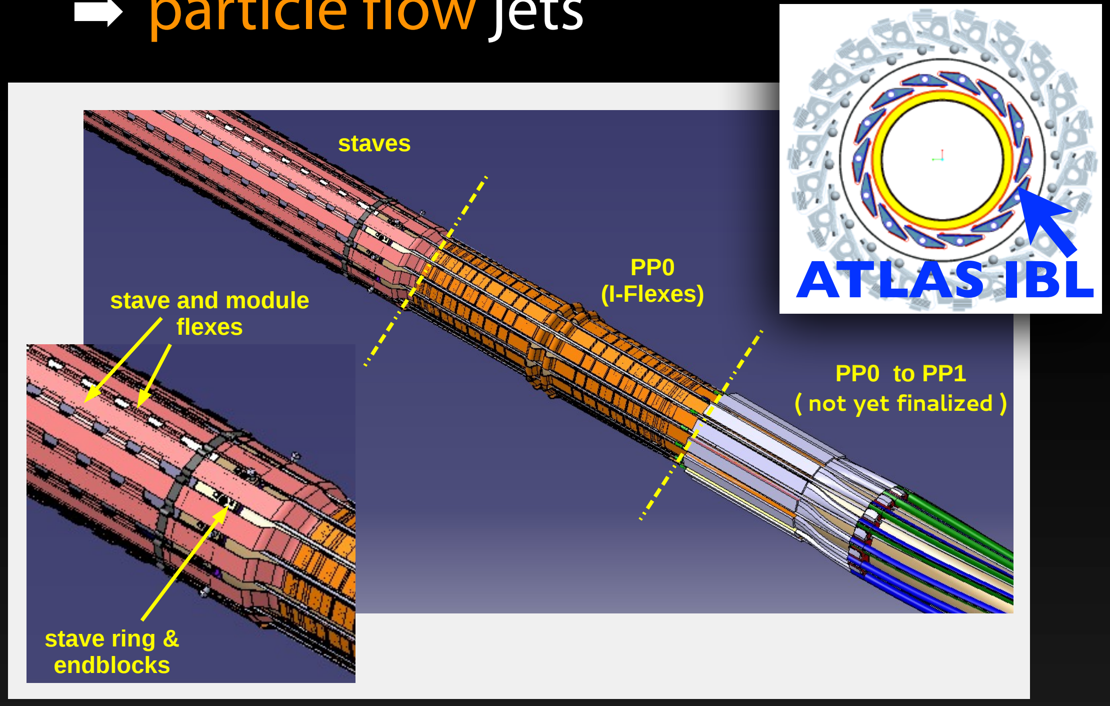
Tracking Developments during LS1

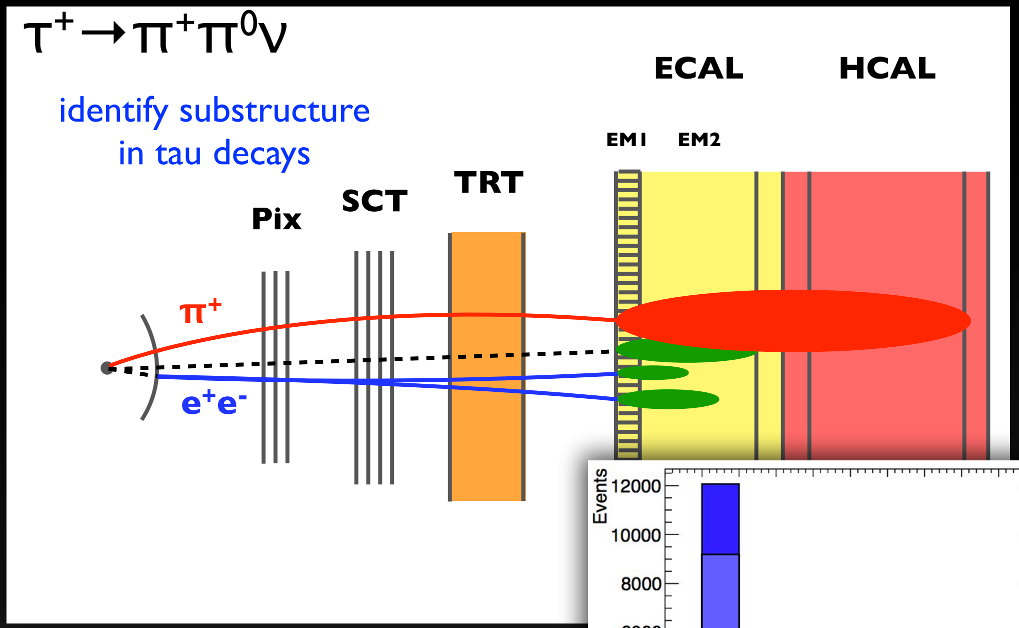
- work on technology to improve CURRENT algorithms
  - ➡ modified track seeding to explore 4th Pixel layer
  - ➡ Eigen migration - faster vector+matrix algebra
  - ➡ use vectorized trigonometric functions (VDT, INTEL libimf)
  - ➡ F90 to C++ for the b-field (speed improvement in Geant4 as well)
  - ➡ simplify EDM design to be less OO (was the "hip" thing 10 years ago)
  - ➡ xAOD: a new analysis EDM, maybe more... (may allow for data locality)
- work will continue beyond this, examples:
  - ➡ (auto-)vectorize Runge-Kutta, fitter, etc. and take full benefit from Eigen
  - ➡ use only curvilinear frame inside extrapolator
  - ➡ faster tools like reference Kalman filter
  - ➡ optimized seeding strategy for high pileup
- hence, mix of SIMD and algorithm tuning
- may give us a factor 2 (maybe more...)
  - ➡ further speedups probably requires "new" thinking

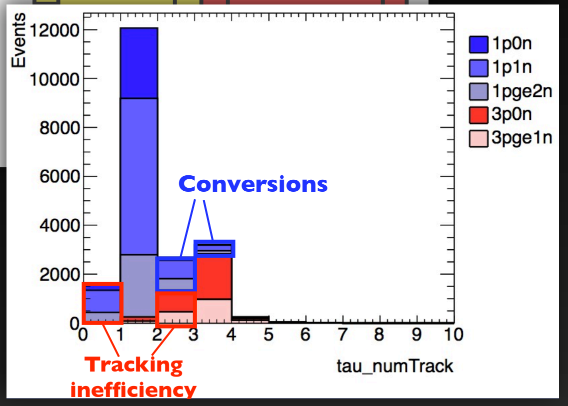


Improved Physics Performance

- algorithms essential part of LS1 development work, examples:
  - ➡ improved topo-clustering for calorimeter showers
  - ➡ new tau reconstruction exploring substructure
  - ➡ new jet and missing E_T software, improved pileup stability
  - ➡ particle flow jets



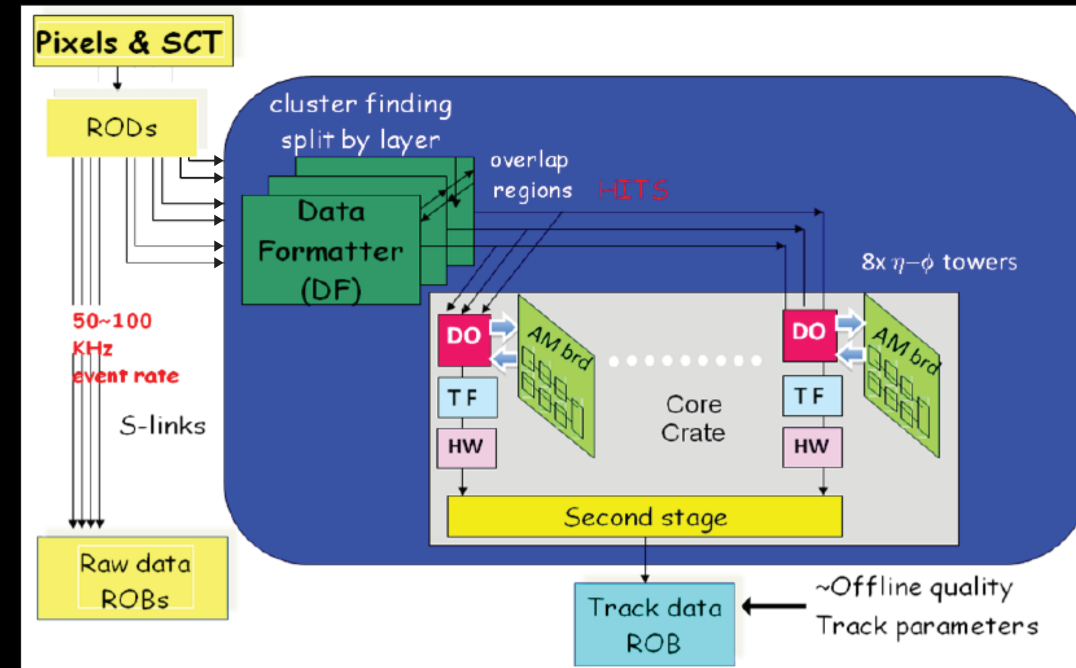


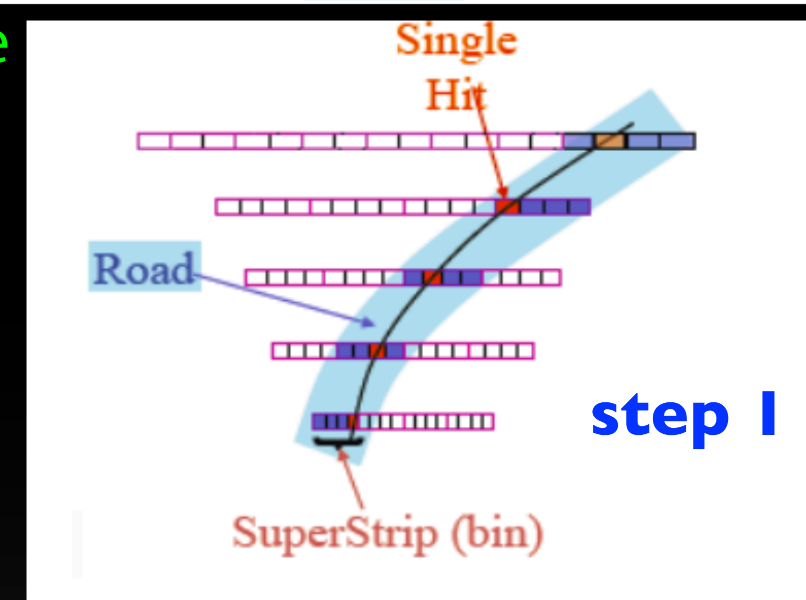


- software for Phase-0 upgrades
  - ➡ full inclusion of IBL in track reconstruction
  - ➡ emulation of FTK in Trigger simulation chain (next slide)

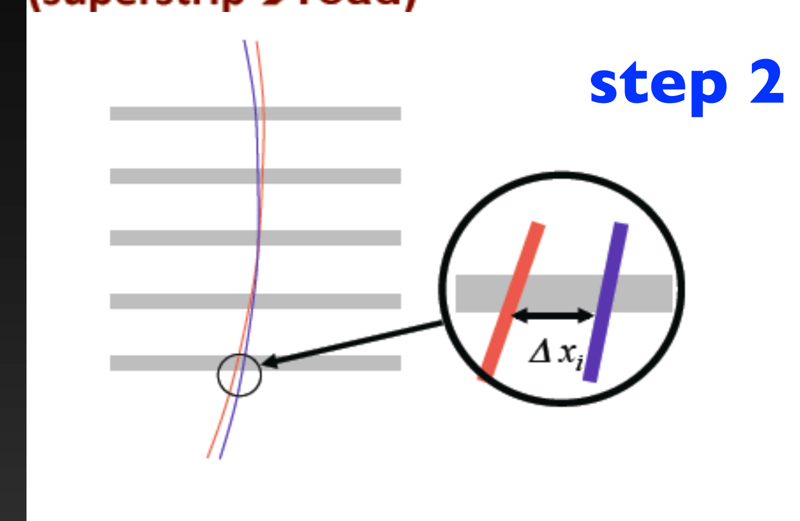
The Fast Tracker (FTK)

- current ATLAS trigger chain
  - ➡ Level-1: hardware based (~50 *kHz*)
  - ➡ Level-2: software based with RoI access to full granularity data (~5 *kHz*) ← tracking enters here
  - ➡ Event Filter: software trigger (~500 *Hz*)
- FTK: hardware tracking (co-processor)
  - ➡ descendent of the CDF Silicon Vertex Trigger (SVT)
  - ➡ inputs from Pixel and SCT
    - data in parallel to normal read-out
  - ➡ two step reconstruction
    - associative memories for parallel pattern finding
    - linearized track fit implemented in FPGAs
  - ➡ provides track information to Level-2 in ~ 25 μs
    - slice installed for 2015, full coverage in 2016
- software integration in simulation chain
  - ➡ FTK is part of digitization & trigger emulation
  - ➡ very resource hungry on CPUs (!)





Pattern recognition in coarse resolution (superstrip→road)

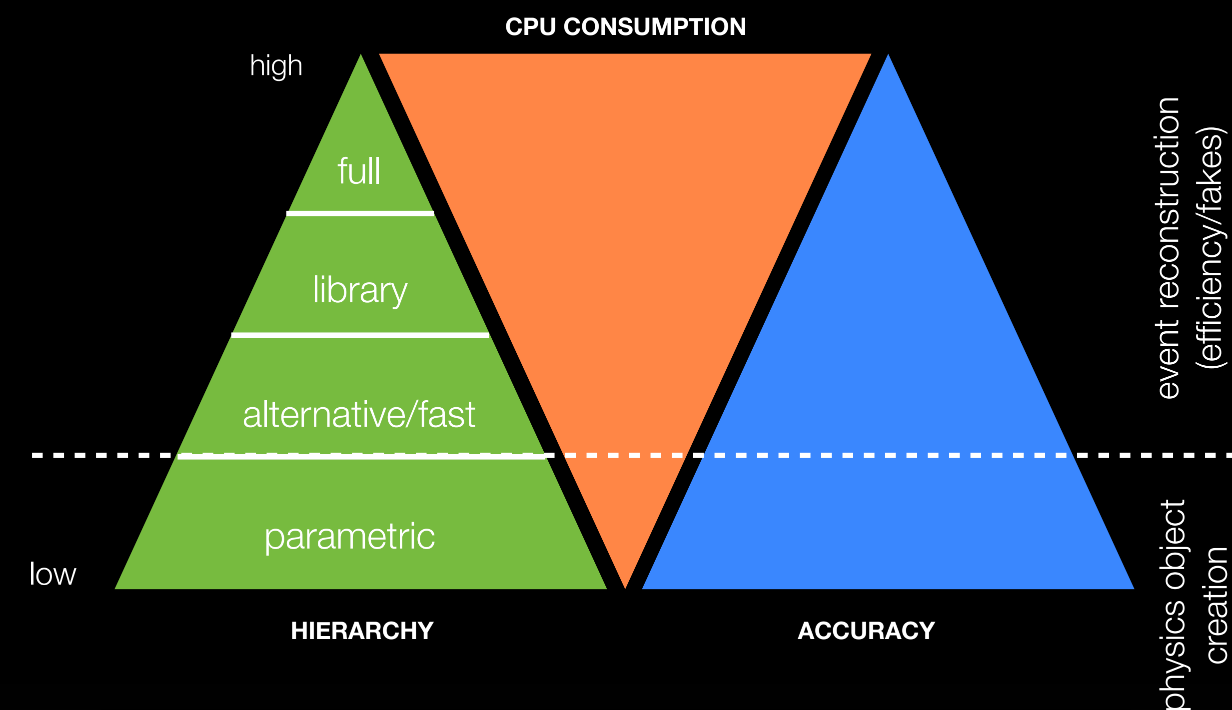


Track fit in full resolution (hits in a road)

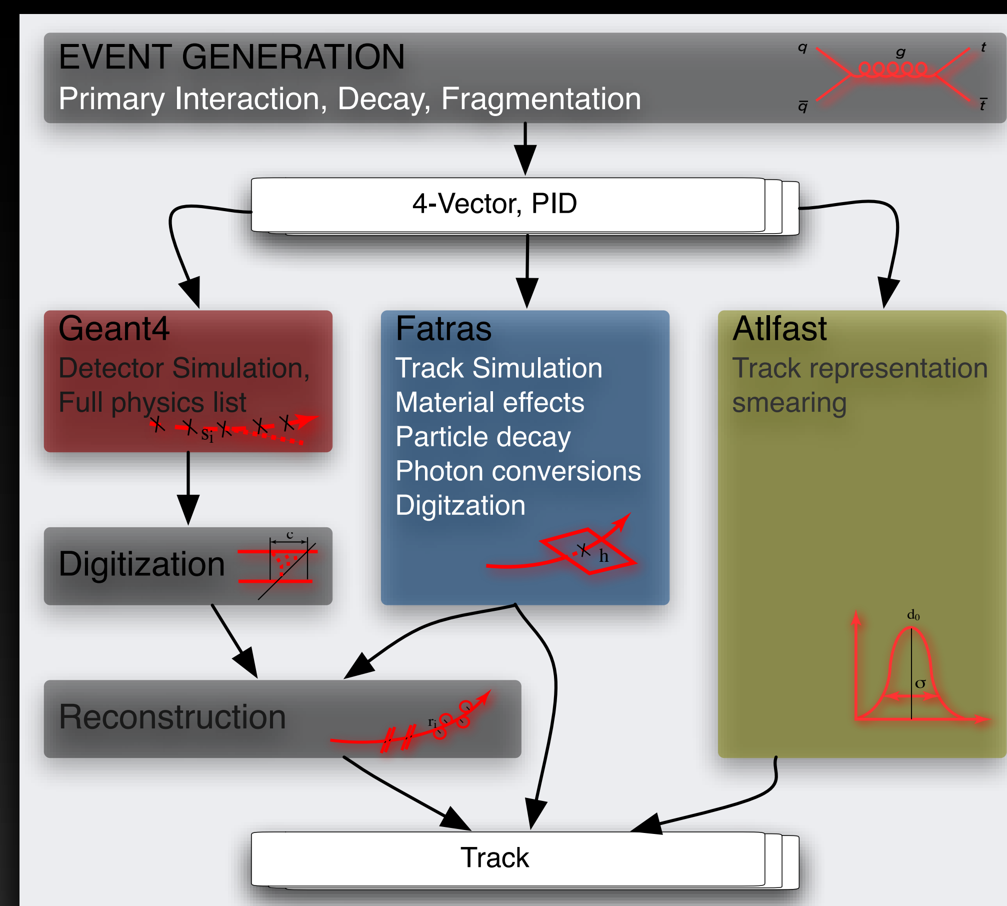
$F(x_1, x_2, x_3, \ldots) \sim a_0 + a_1\Delta x_1 + a_2\Delta x_2 + a_3\Delta x_3 + \ldots = 0$

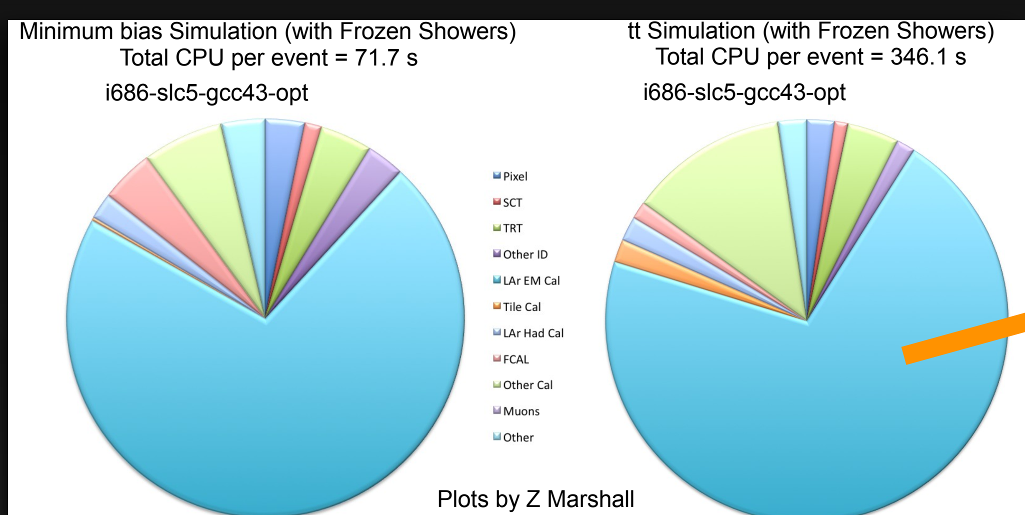
Towards Simulation for Run-2

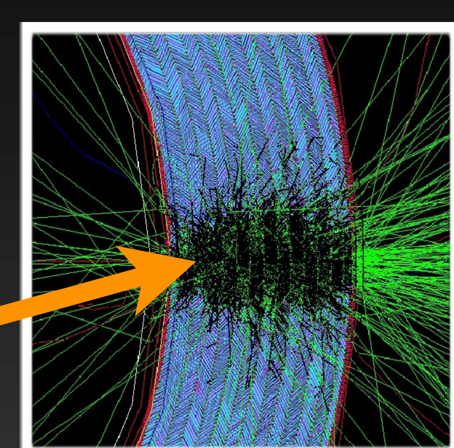
- full simulation is resource driver
  - ➡ various flavors of fast simulation available
    - frozen showers, AtlFast-2, parametric ...
    - fast track/muon simulation Fatras
  - ➡ question is what is the best compromise between CPU consumption and accuracy ?



- so far fast simulation used for
  - ➡ very forward showers in otherwise full sim.
  - ➡ for large productions of specific samples
    - e.g. SUSY parameter scans
    - Phase-2 upgrade studies



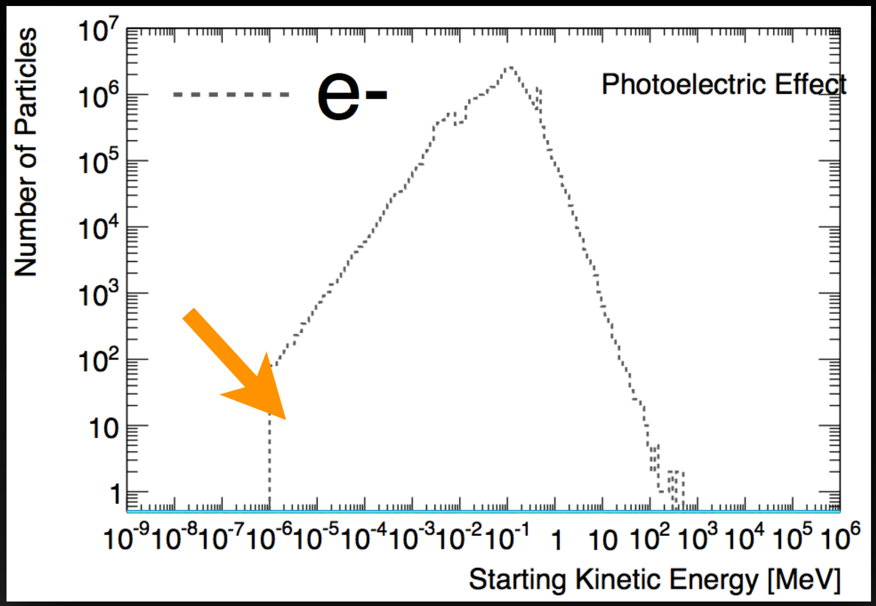


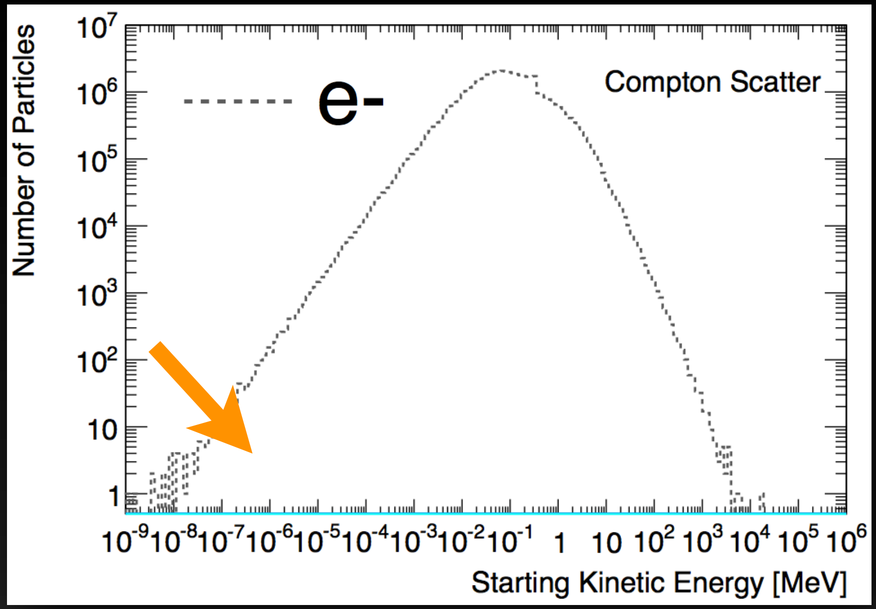


Zach Marschall

Fixing Features in Geant4

- recent profiling revealed a number of physics features
  - ➡ no major code hot spots other than known ones (EMEC)
  - ➡ a few surprises (pointer sets; physics processes that instantiate a stepper-in-field)
- features found that we in ATLAS should fix
  - ➡ removing all neutrinos and not letting them propagate
- issues that the G4 team has provided options for
  - ➡ removing low energy secondaries from certain processes (below) is optional (now in validation)
  - ➡ revising range cuts at the same time



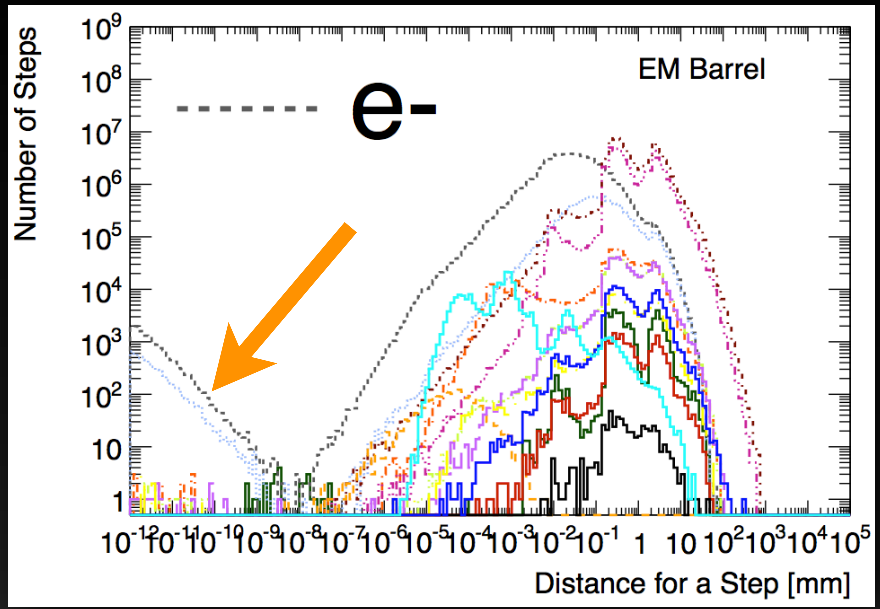


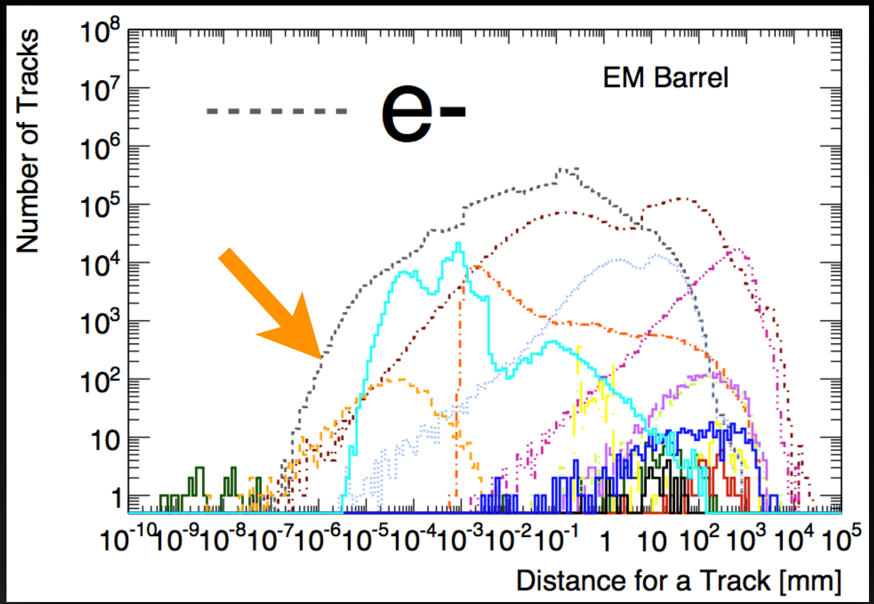
- support by Geant4 team is very important for ATLAS
  - ➡ e.g. debugging recent issue in G4PolyCone

Zach Marschall

Electron Propagation in Geant4

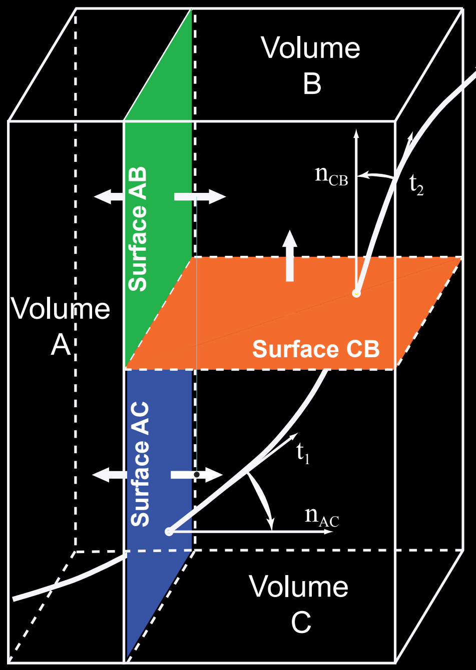
- in the EM and hadronic barrel calorimeters
  - ➡ there are a significant number of electrons propagating <100 fm in a step
  - ➡ re-running now to try to drop the x-range of the histogram (batch is slow)
- not many electrons with a total track length <100 pm
  - ➡ these are steps in a track, not single steps before the electron dies
- highlights one major issue:
  - ➡ there are very few people who fully understand the navigation and interplay with physics processes, and this is the major source of headaches and concern in terms of performance





A.Salzburger

Fatras Tracker Simulation

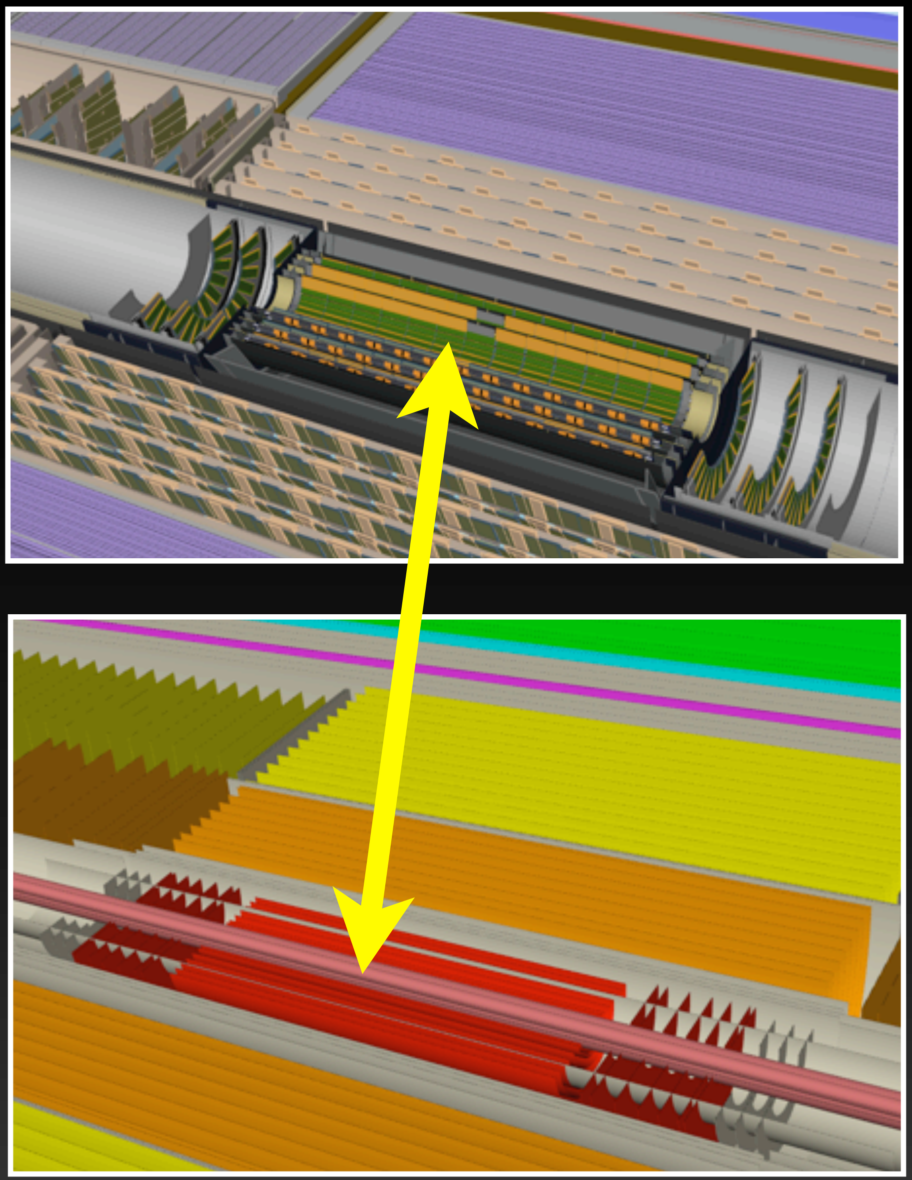


- ATLAS has 2 geometry systems (not special)
  - ➡ full model used in Geant4 with 4.8M placed volumes
  - ➡ reconstruction model for fast tracking
    - reduced complexity
    - material projected onto surfaces
- fast extrapolation engine
  - ➡ embedded navigation replaces voxialization

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)

  - ➡ plus: fast adaptive Runge-Kutta-Nystrom codes



- Fatras simulation engine
  - ➡ re-uses track reconstruction infrastructure
  - ➡ combined with particle stack and fast physics processes
  - ➡ optionally: fast digitization codes

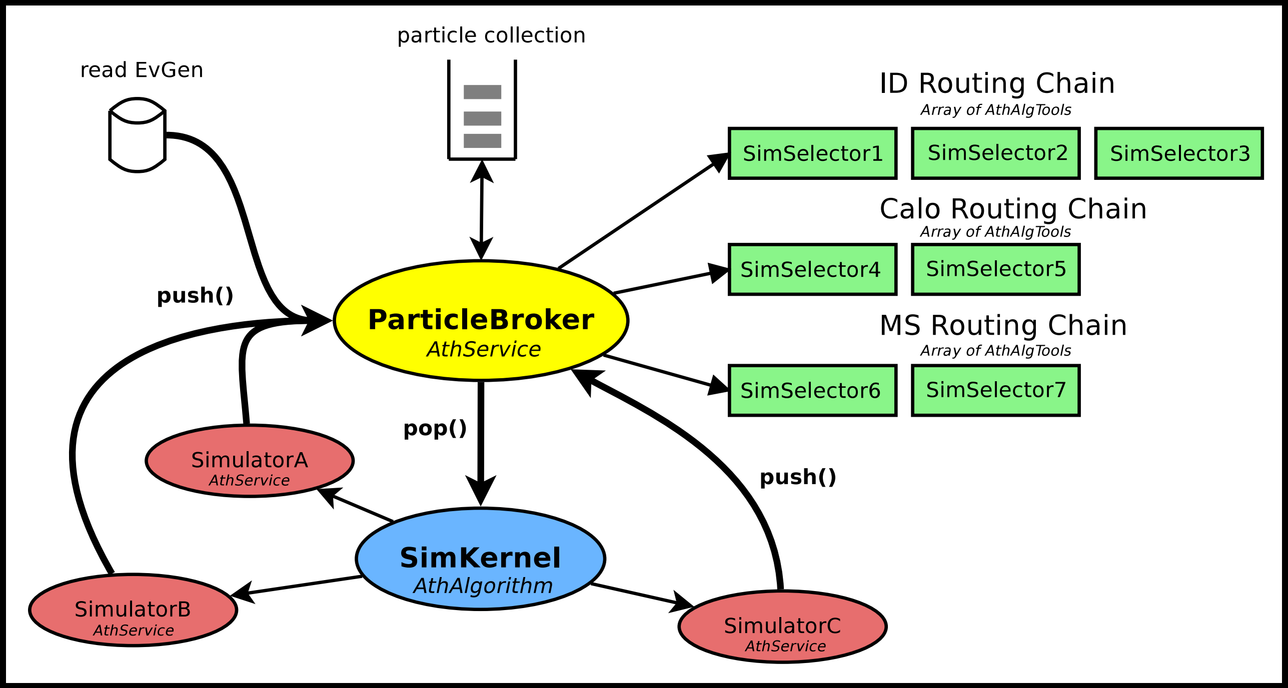
A.Salzburger, E.Ritsch et al.

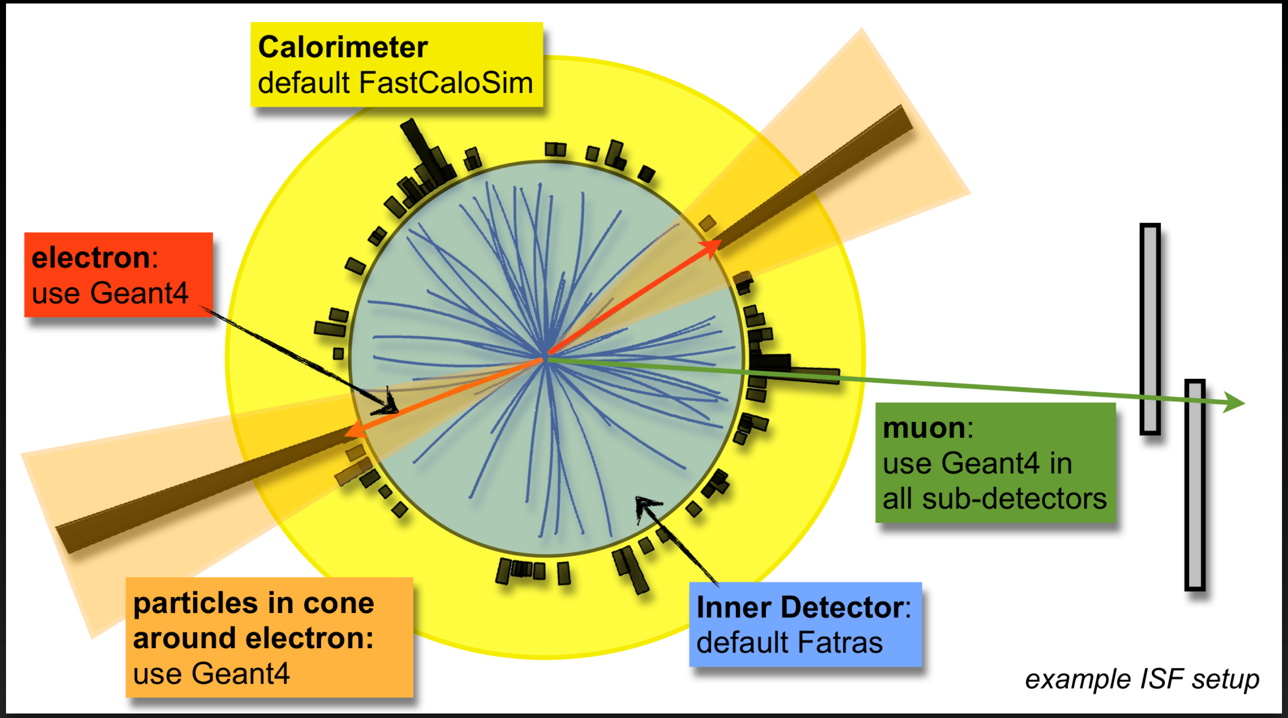
Integrated Simulation Framework (ISF)

- one framework for all
  - ➡ external particle broker and sim. kernel
  - ➡ simulation codes act as services
- vision behind ISF is broader !
  - ➡ based on RoI guidance used in Trigger
    - combine particle broker with selectors
  - ➡ mix different simulation types in 1 event
    - full simulation for regions of interest
    - fast simulation for underlying event pileup

Tracker	Calo.	Muons	speedup
full	fast	full	~20
fast	fast	fast/full	>100
RoI guided fast/full			~100

- ➡ exploring full potential requires:
  - fast digitization and reconstruction
  - ISF principle for both, not to loose precision in regions of interest

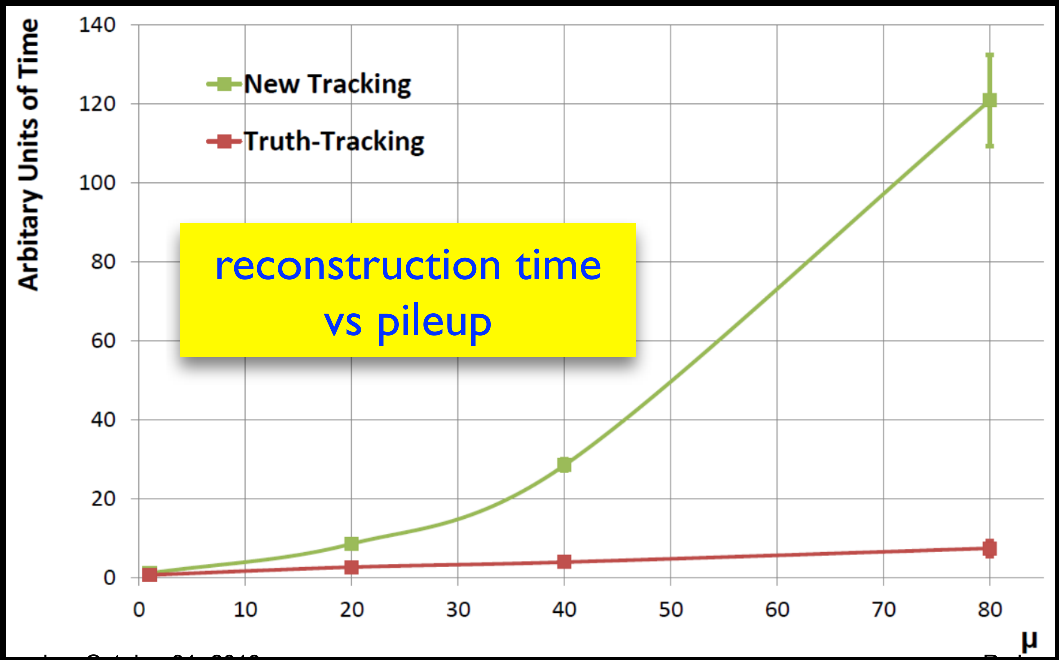


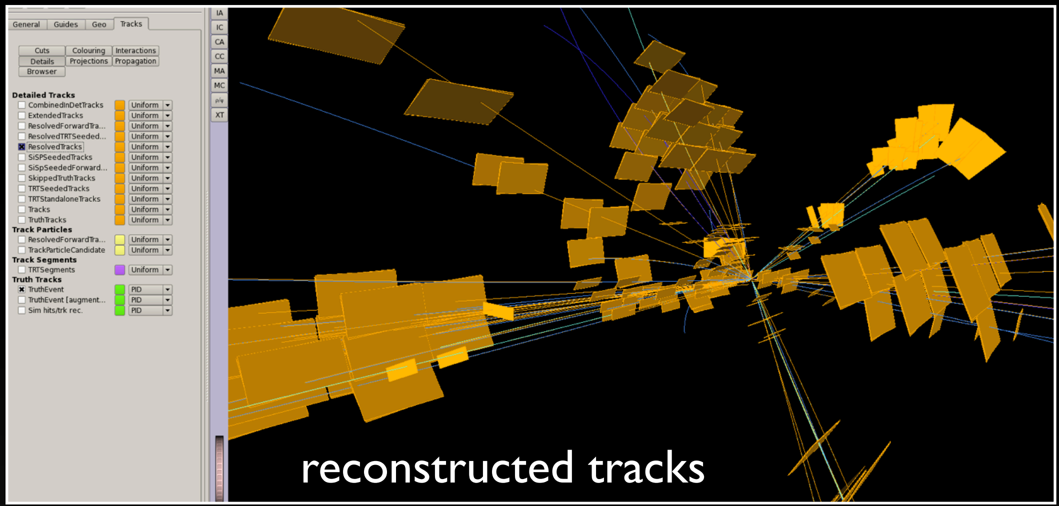


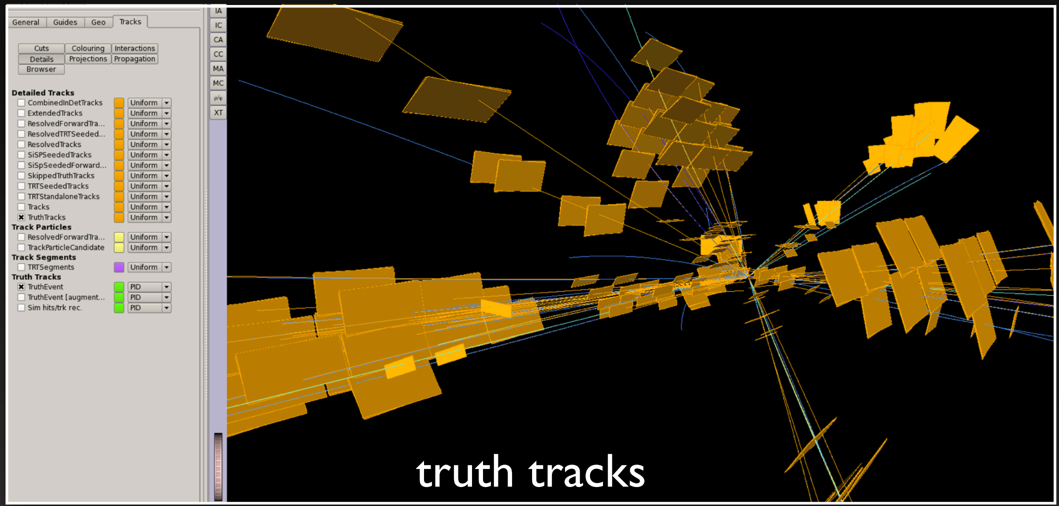
R.Jansky et al.

Truth Tracking from MC

- for very fast ISF simulation options
  - ➡ MC truth based hit filter to find tracks
  - ➡ replace pattern recognition in tracker
    - otherwise limiting CPU driver
- good results achieved
  - ➡ real pattern is very efficient and very pure
    - modeling of hit association mostly ok
  - ➡ models main source of inefficiencies well
    - this is hadronic interactions in material
  - ➡ uses full fit, so resolution come out right
  - ➡ and it is fast (trivial) !
- still, corrections are needed
  - ➡ especially double track resolution
    - affects jet cores, taus, maybe 140 pileup (?)
  - ➡ corrections are topology dependent







Rob Harrington

Geant4-MT Developments

- integration test of early Geant4-MT into ISF
  - ➡ encountered some technical issues:
    - semaphore class awkward to use
    - Athena issues: AthAlgTool not thread-safe
    - G4Atlas issues: FadsSteppingAction is a singleton
    - ISF integration: hit container is managed by ISF, not by Geant4-MT
- plan is to move to Geant4.10 next
  - ➡ new G4-MT version requires some interface changes
  - ➡ make user actions thread save
  - ➡ resolve ATHENA integration issues
  - ➡ move from semaphore to TBB
- work is still in early stages
  - ➡ need to understand best strategy of how to explore parallelization
  - ➡ realistically, timeline is more towards after LS1 (Run-3 ?)

J.Catmore

Three Reasons for new Analysis Model

J.Catmore

Three Reasons for new Analysis Model

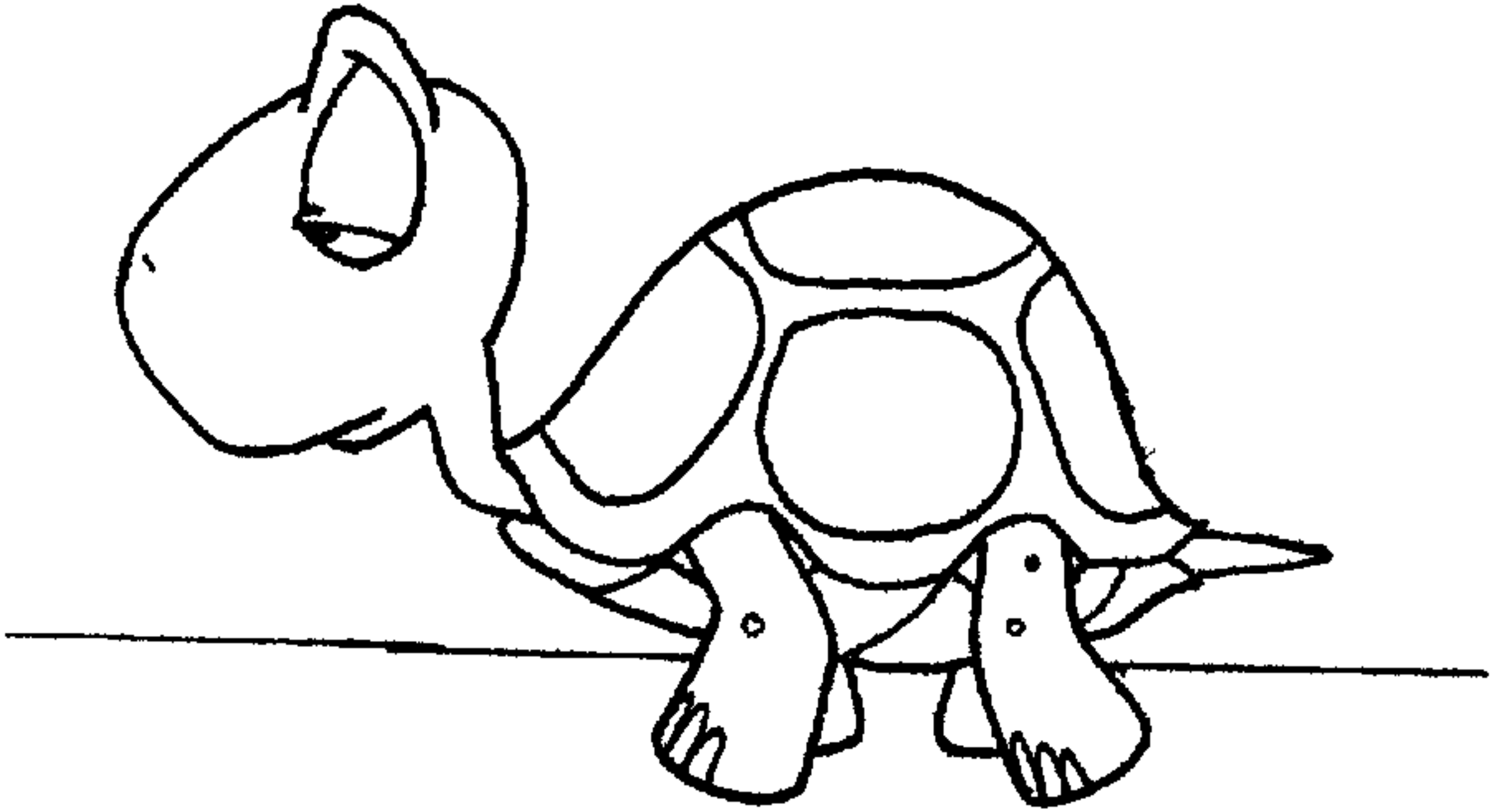
I: RESOURCES

- **Flat cash** for computing during the Run 2 period from many funding agencies
- Some existing equipment will need to be replaced
- We will not have the big increases in storage that we had in 2010-2012

J.Catmore

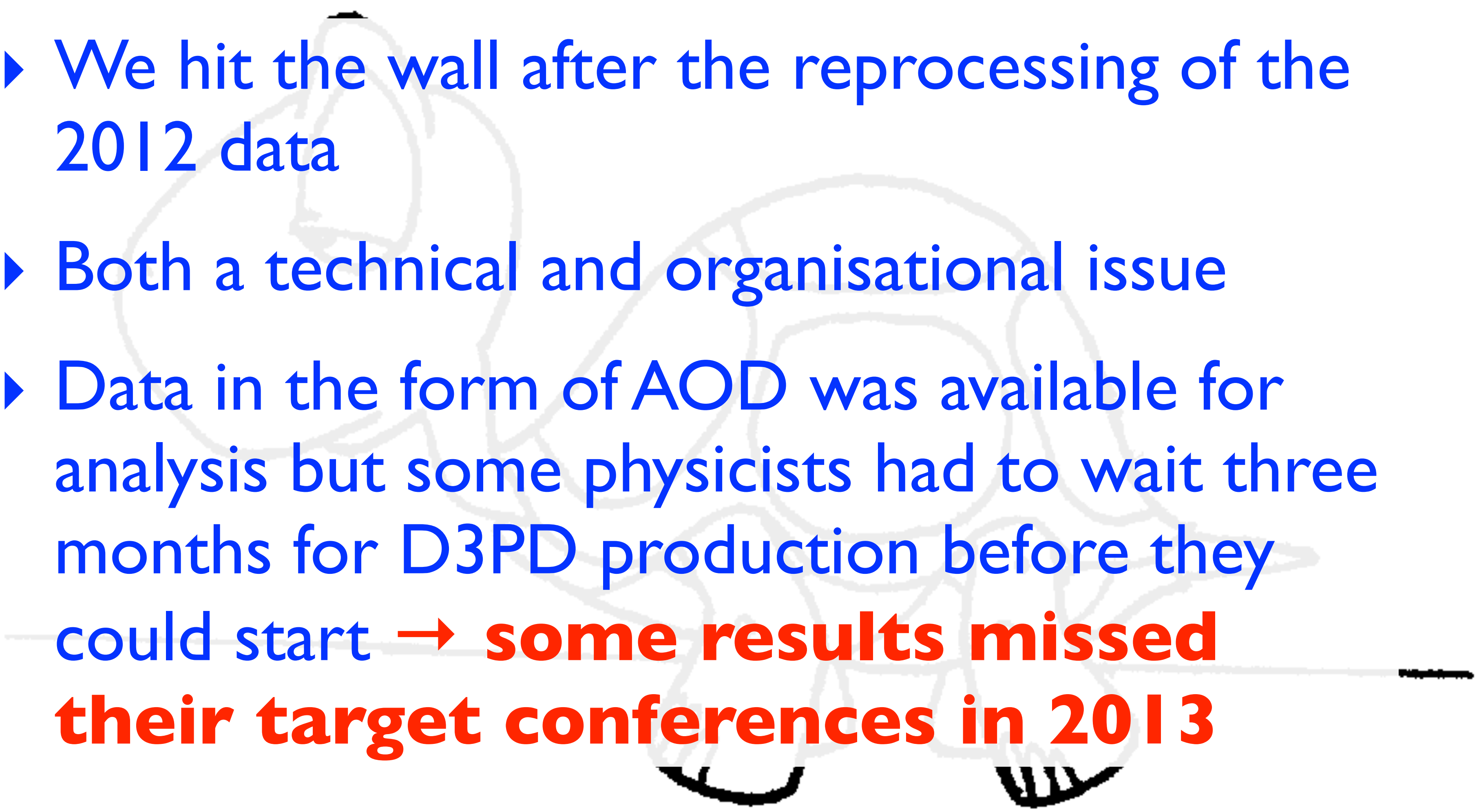
Three Reasons for new Analysis Model

2: SPEED



Three Reasons for new Analysis Model

2: SPEED

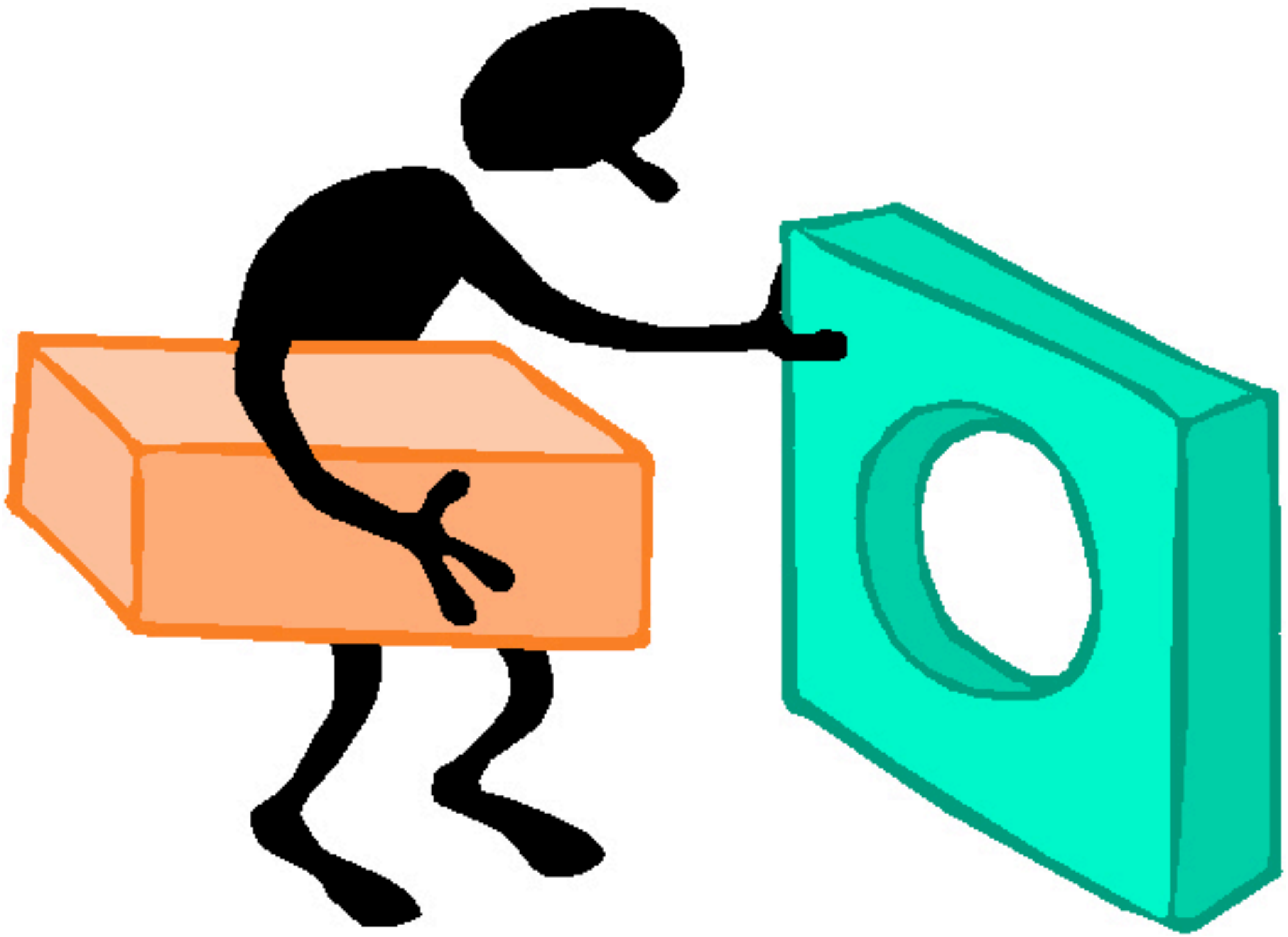


- We hit the wall after the reprocessing of the 2012 data
- Both a technical and organisational issue
- Data in the form of AOD was available for analysis but some physicists had to wait three months for D3PD production before they could start → **some results missed their target conferences in 2013**

J.Catmore

Three Reasons for new Analysis Model

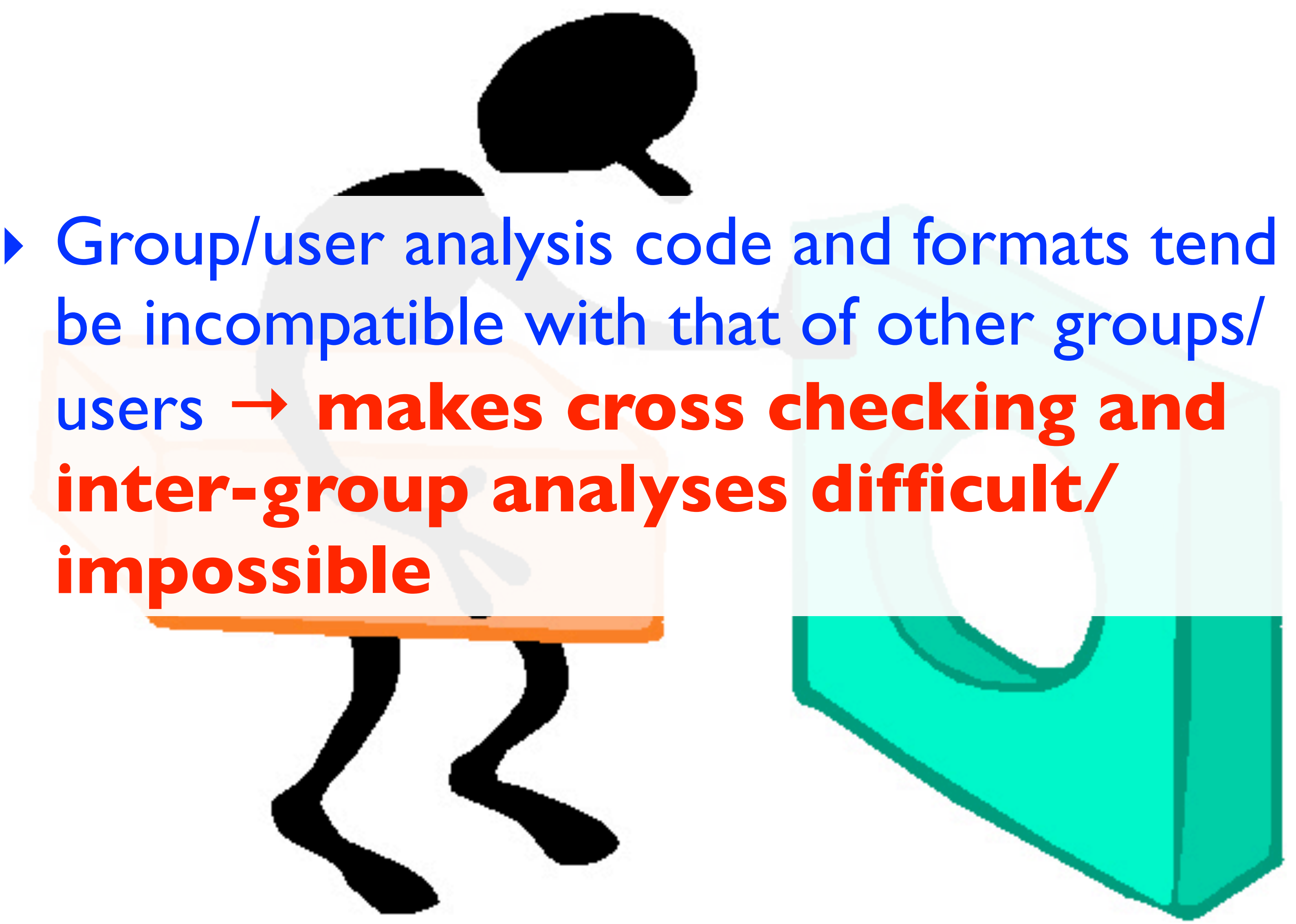
3: COMPATIBILITY



J.Catmore

Three Reasons for new Analysis Model

3: COMPATIBILITY

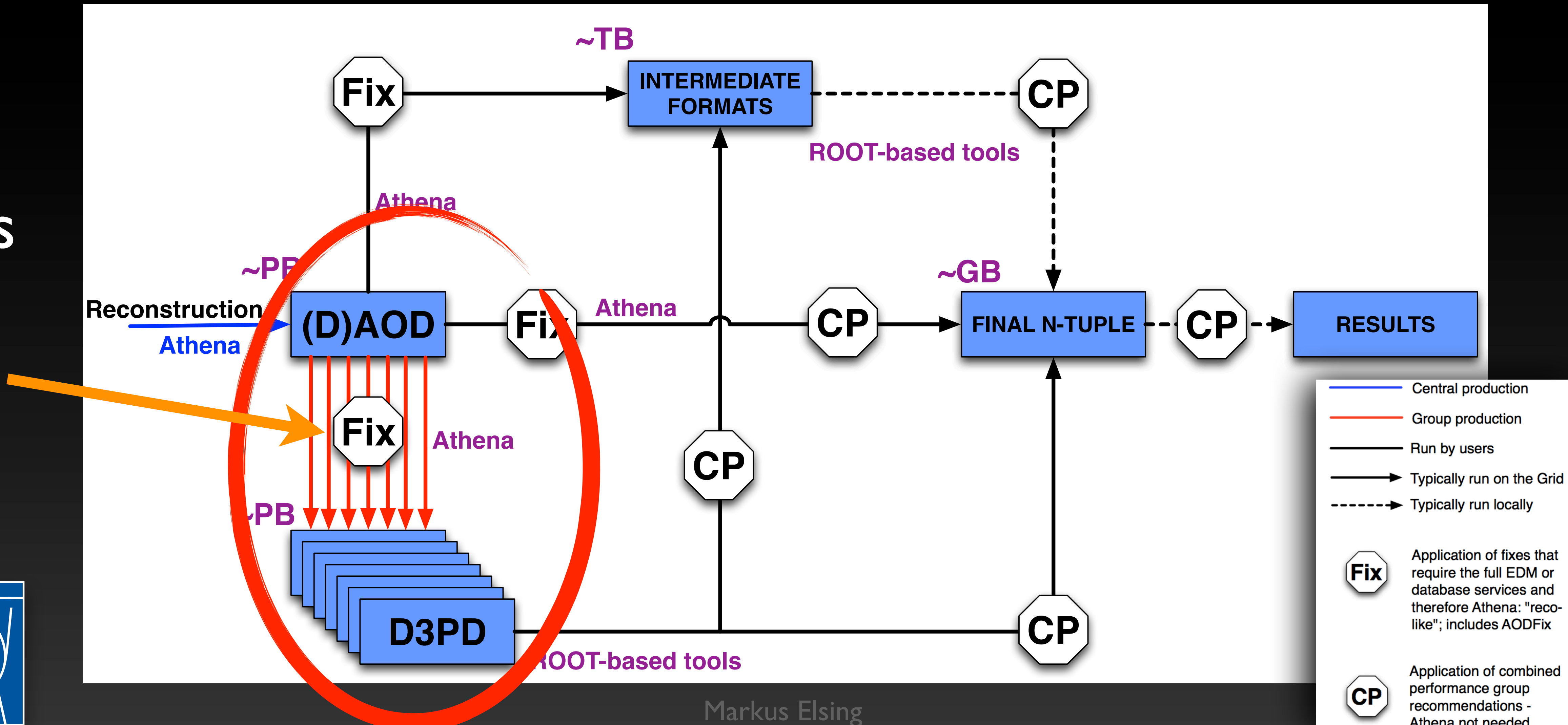


- Group/user analysis code and formats tend to be incompatible with that of other groups/users → **makes cross checking and inter-group analyses difficult/impossible**

Revising the Analysis Model

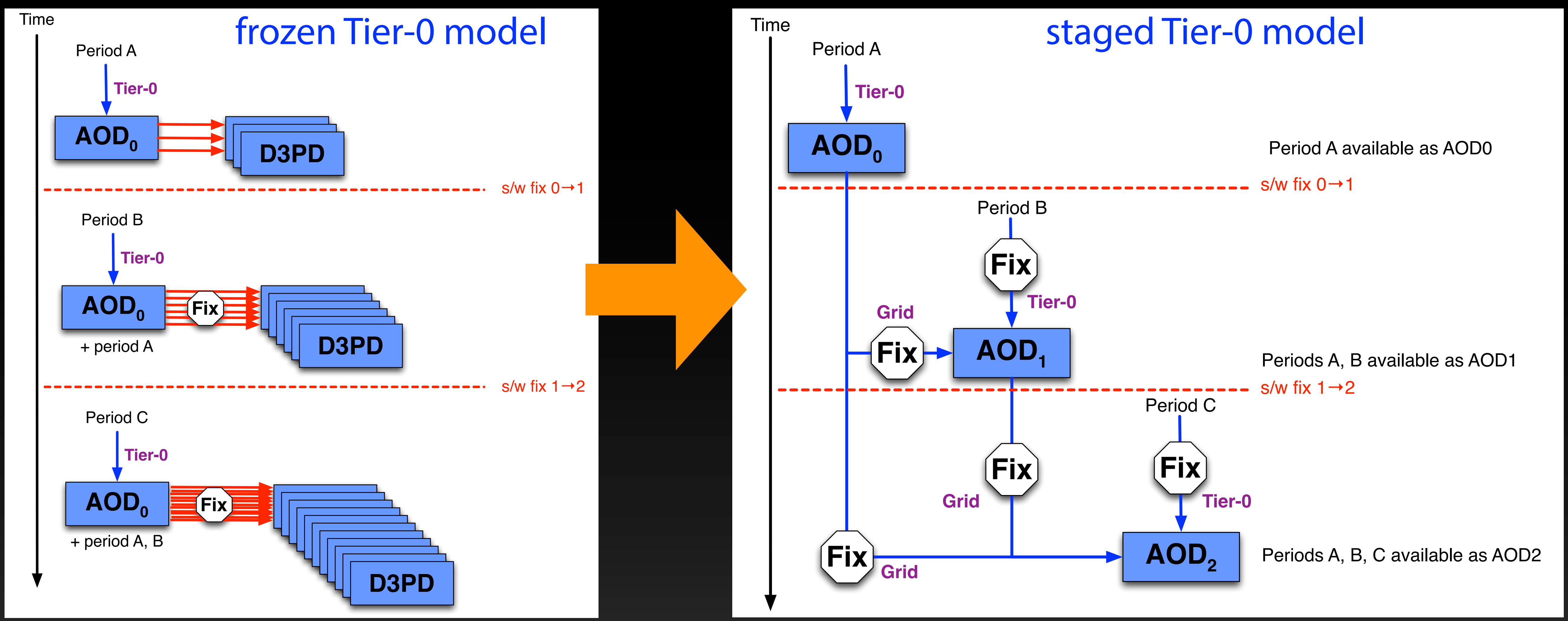
- Run-1 analysis model
  - ➡ 20% of analysis teams used AOD in ATHENA
  - ➡ mainly based on D3PD, flat ntuples customized per analysis team, and ROOT
  - ➡ resulting model grew complex, repetitive, with lots of overhead...
- D3PD production
  - ➡ factor 2-3 in disk space and CPU time compared to Raw reco. + AOD (!!!)

fixes on the fly



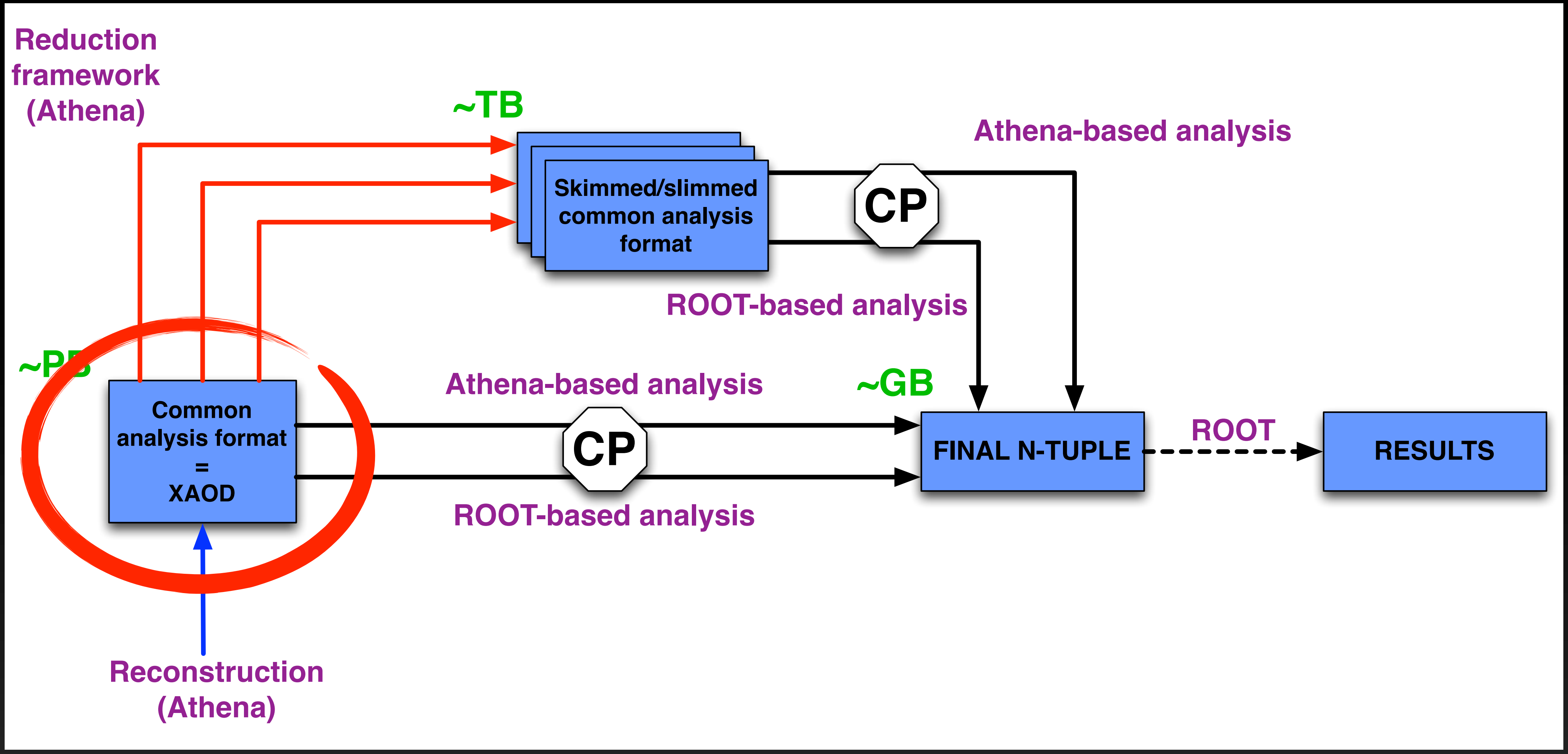
The New Analysis Model

- replace “frozen Tier-0 policy” with “stage Tier-0” policy
  - ➡ apply fixes and updates centrally in Tier-0 and update xAOD on GRID
  - ➡ more flexibility, reduces production overhead, validation is crucial (!)



The New Analysis Model

- key is xAOD as merger of AOD and D3PD
  - ➡ xAOD is ROOT and ATHENA writeable and readable
  - ➡ ROOT becomes official ATLAS software framework (for the first time)

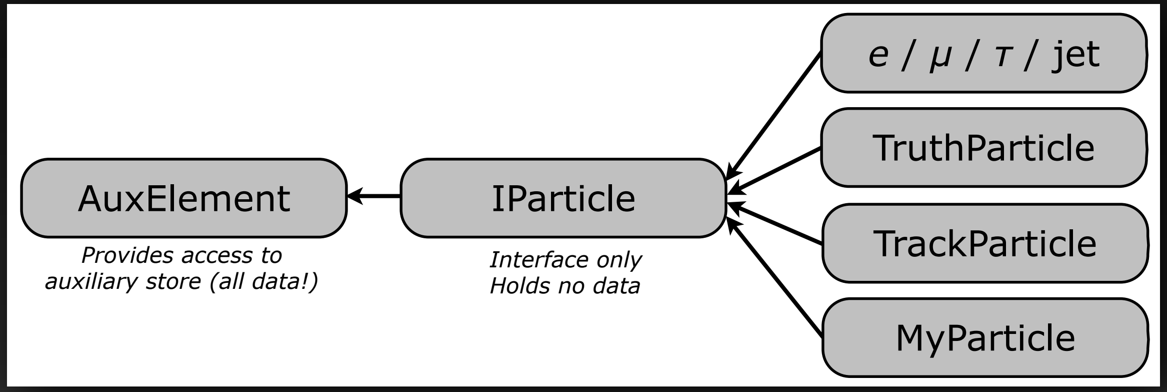


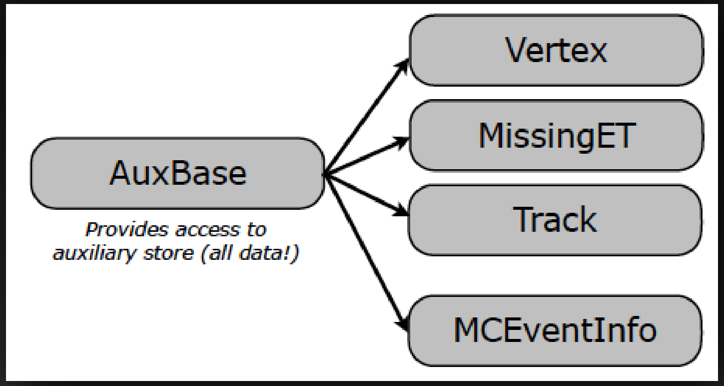
- xAOD is subject of ASG Task Force 1

Attila Krasznahorkay

xAOD File Format

- merges the good properties of ATLAS's AOD and D3PD formats, used in Run-1
  - ➡ provides an OO user interface
  - ➡ provides the same amount of flexibility for file content manipulation as the Run-1 D3PD files (flat ntuples)
  - ➡ provides partial & lazy information loading from the input file, down to the individual variable level
    - i.e. can read just a subset of the information about all the electrons easily
- transparent use in ROOT and ATHENA
  - ➡ using a small amount of EDM libraries (<100 MB)
- but: requires the use of many (O(10k)) branches
  - ➡ like for current D3PD files, see ROOT I/O workshop discussion





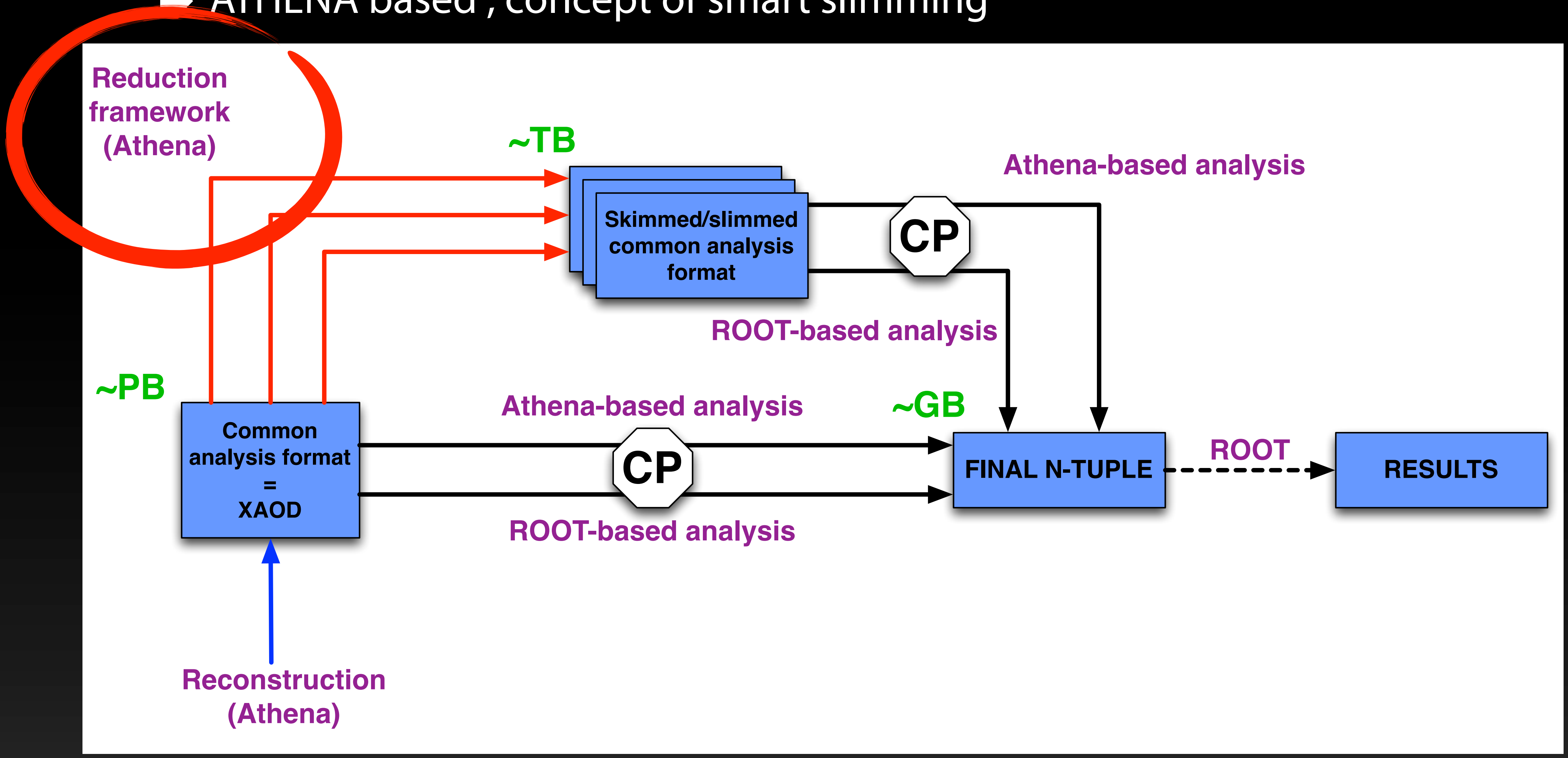
Attila Krasznahorkay

ROOT Features Used for xAOD

- custom read rules for the persistent pointer types
  - ➡ implementation required updates to ROOT I/O code
  - ➡ read rules themselves are very simple, just a way of resetting the cache of the smart pointers after an I/O operation.
- custom collection proxy for the ATLAS specific DataVector<T> type
  - ➡ allows us to read/write DataVector<T> objects as a simple list of T, while still allowing us to use the special abilities of DataVector transiently
- having the ROOT dictionary not take default template arguments into account in the class's name
  - ➡ needed to hide differences between classes that ROOT should not be aware of (when the I/O happens inside/outside of our offline software infrastructure)
  - ➡ still to be implemented in ROOT 6
    - plan exists for the development, it was just not a high priority for now
- support from ROOT team has been and will be vital !!!

The New Analysis Model

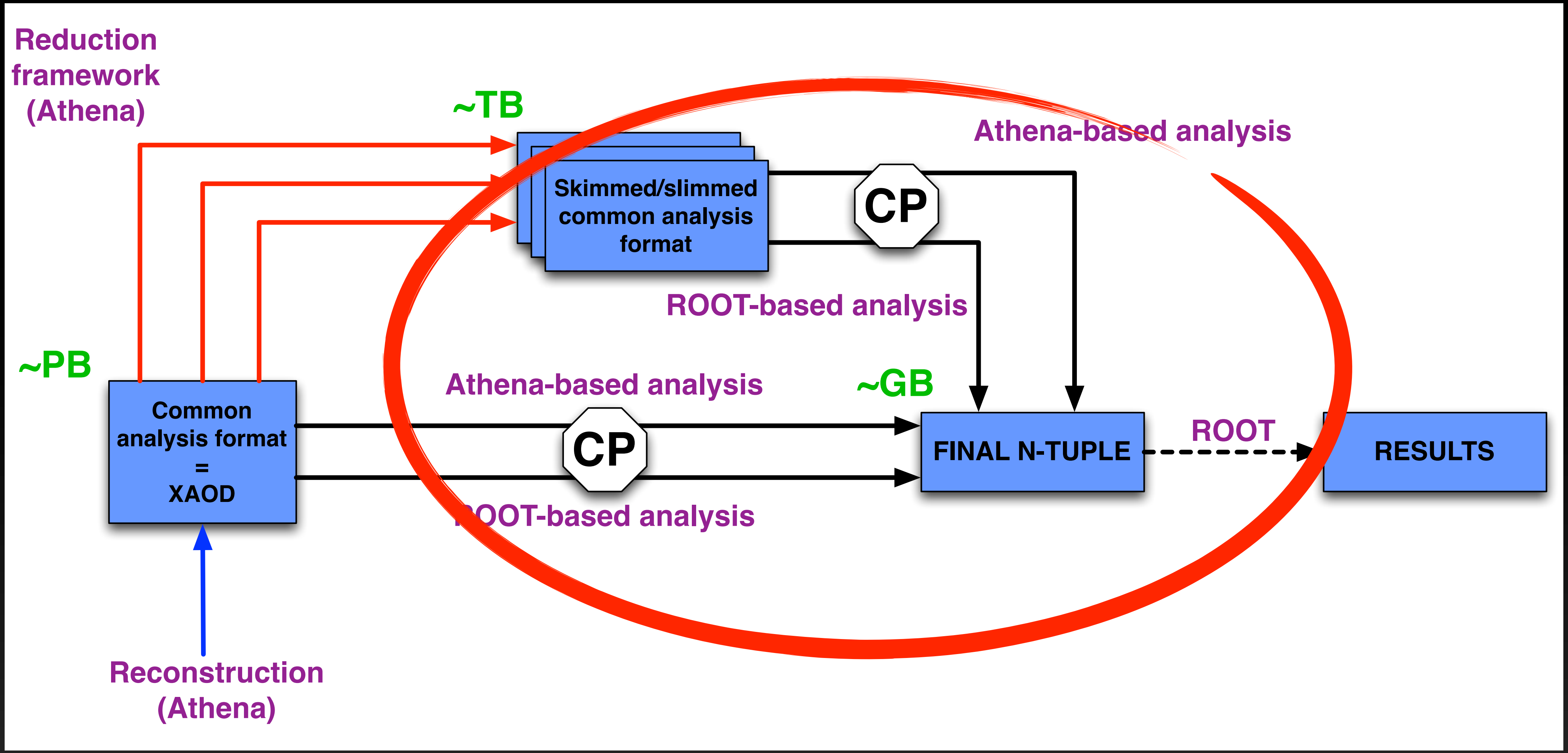
- reduction framework does heavy lifting
  - ➡ analysis trains per physics team or combined performance activity
  - ➡ ATHENA based , concept of smart slimming



- reduction framework is subject of ASG Task Force 2

The New Analysis Model

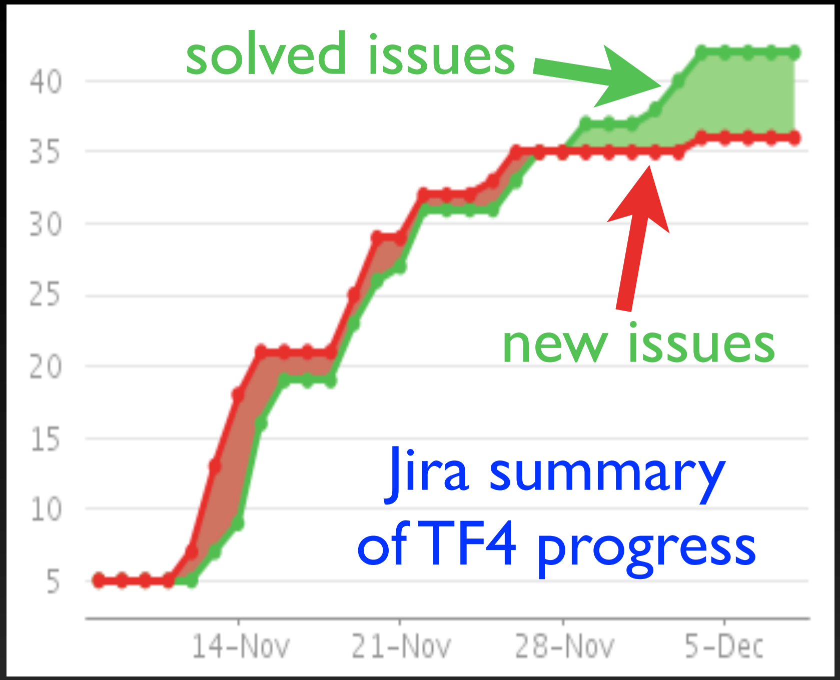
- analysis framework with dual use CP tools
  - ➡ establish new ROOT (and MANA/ATHENA) analysis releases (RootCore/HWAF)
  - ➡ tool interface (configuration, messaging, store) transparent to frameworks



- reduction framework is subject of ASG Task Force 3

Migration of Offline Reconstruction

- major migration work needed for reconstruction software
  - ➡ new output format xAOD for new Analysis Model
  - ➡ redesign of (simplified) tracking EDM
    - including CLHEP to Eigen migration
    - affects all combined reconstruction, etc.
- established Task Force 4 within Reconstruction Group
  - ➡ organizes migration following new tracking EDM
  - ➡ implements xAOD classes for all domains and adapts reconstruction accordingly
- critical path for LS1 software work
  - ➡ deadline for release 19.0.2 next March
    - start of DC-14 production (see later)



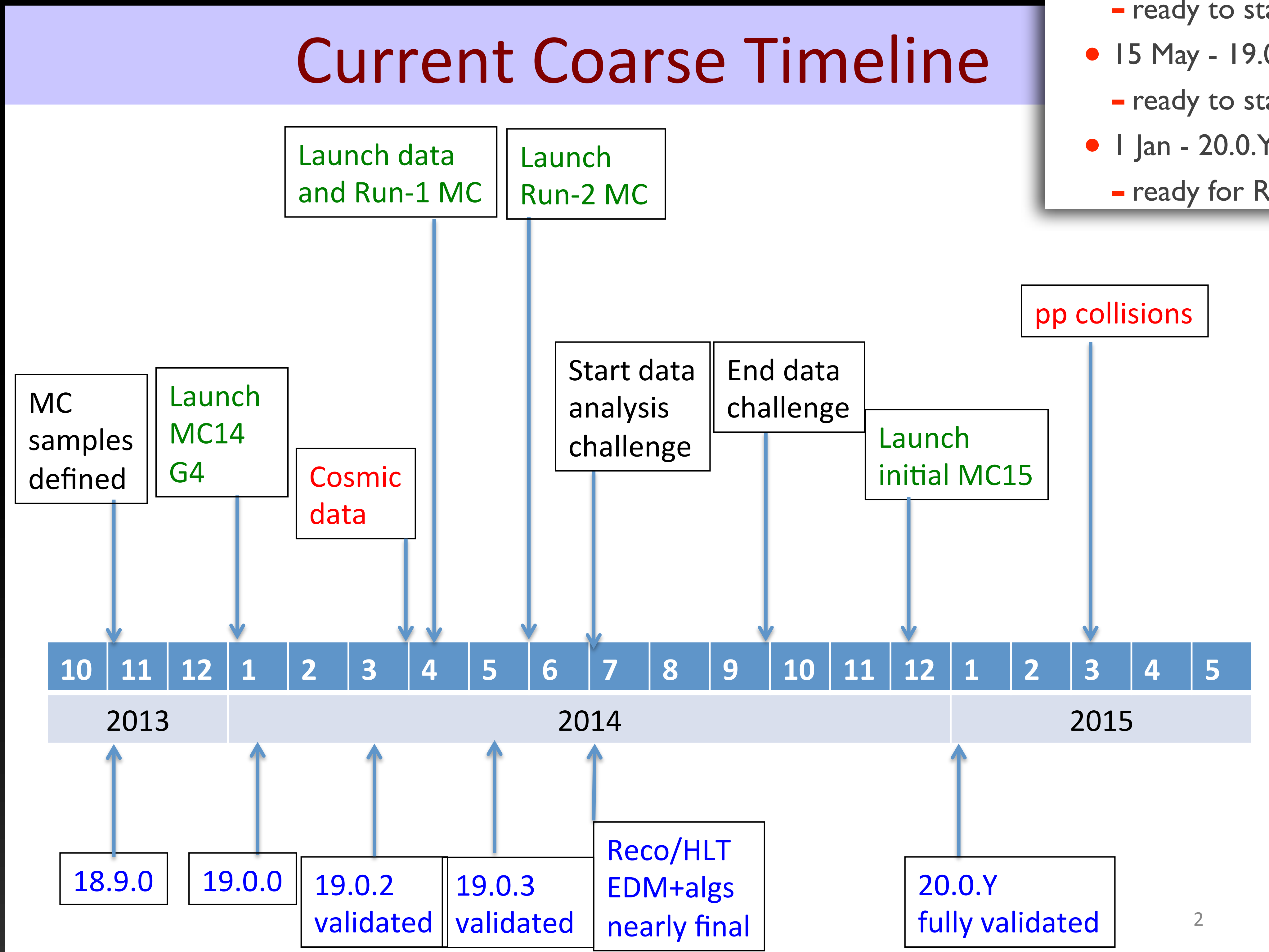
Data Challenge-14

- main goal: prepare ATLAS for Run-2 physics analyses
  - ➡ test the new Analysis Model
    - may need to react and adjust model depending on experience and feedback from physics groups
  - ➡ commission the ISF in context of physics analysis
    - full simulation and various aspects of fast and full simulation
  - ➡ test any updated reconstruction algorithms for Run-2
  - ➡ provide large scale test of upgraded distributed computing environment
    - ProdSys-2 (production system) and Rucio (data management system)
- DC-14 is main focus of Software Project until summer
  - ➡ priority over other activities, necessary to achieve main goals

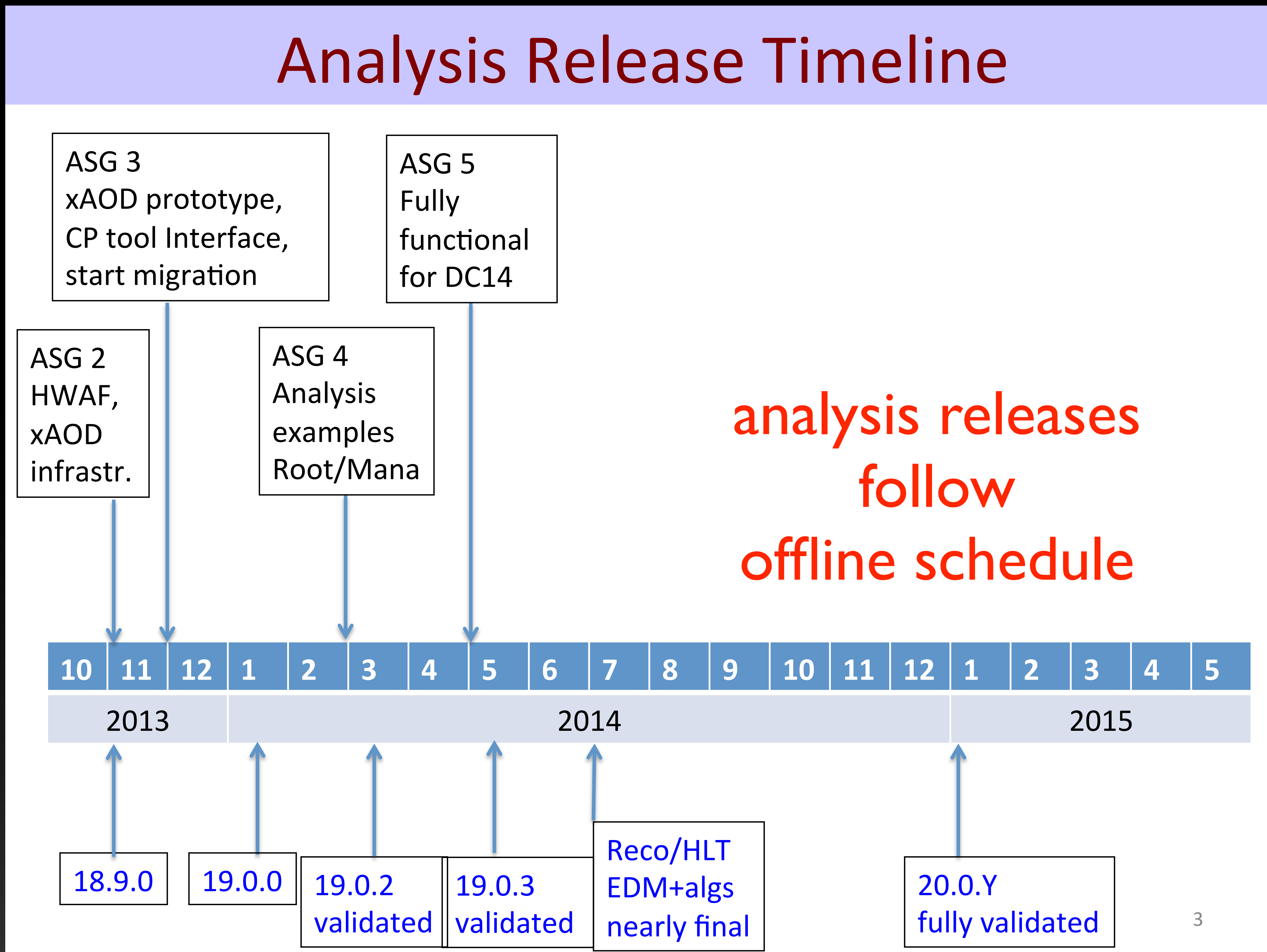
Data Challenge-14 Schedule

Key Deadlines

- 15 March - 19.0.2 validated
  - ready to start Run-1 reco
- 15 May - 19.0.3 validated
  - ready to start Run-2 reco
- 1 Jan - 20.0.Y validated
  - ready for Run-2 data



Analysis and Offline Release Schedule



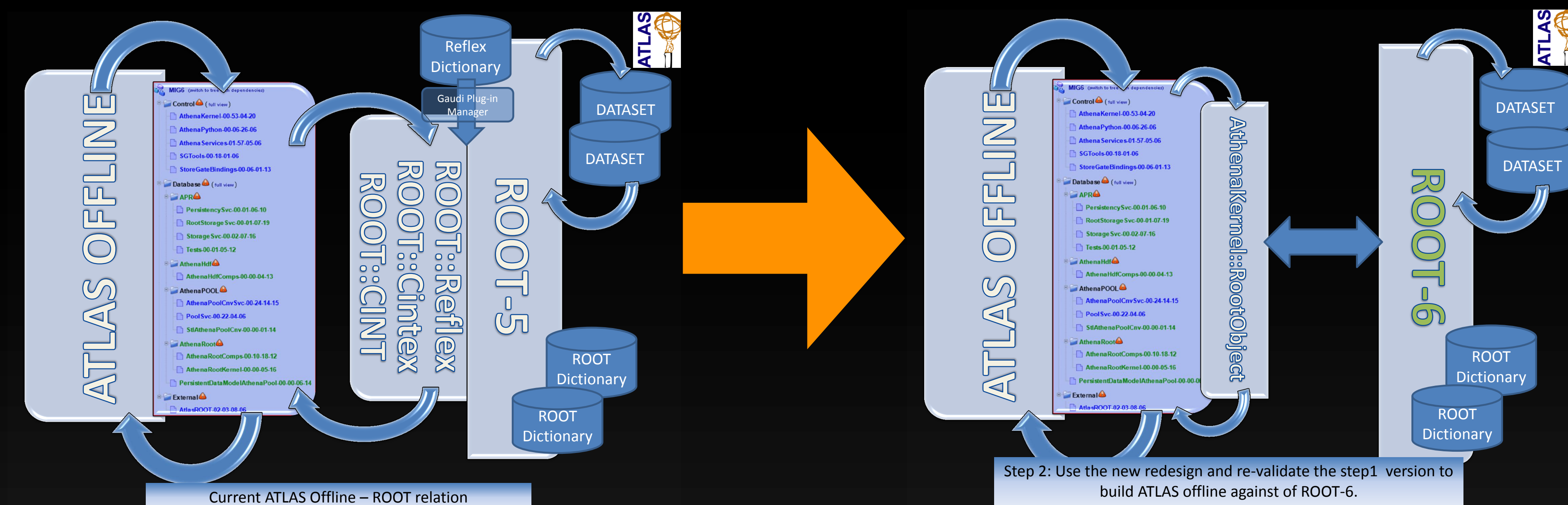
Release 20: Preparation for Data Taking

- release 19.1.0
  - ➡ merging of ISF simulation branch into current development release
  - ➡ T/DAQ project branches from offline dev. release
    - base release for Run-2 at Point-1
    - used for cosmic data taking with IBL
    - may import algorithmic improvements later from dev. release
- incorporate feedback from DC-14 and finalize updates of algorithmic code for 13 TeV running
  - ➡ including (auto-)vectorization and timing optimization
- reestablishing schema support for AOD to xAOD
  - ➡ using Athena T/P layer, non-trivial schema evolution
- migration from CMT to HWAF
  - ➡ ASG release and offline releases use same build system
- migration to Root6 (next slide)

V.Fine, S.Binet

Status of Root6 migration

- Root6 comes without Reflex, Cintex, Cint
  - ➡ ATLAS software currently relies heavily on them
    - and we need full support of new xAOD features
  - ➡ migration benefits from Root6 task force and direct help of Root team (!)
- strategy for changing software stack:



  - ➡ AtlasCore compiles without Reflex, in 17.2.X release branch

AtlasCore	x86_64-slc6-gcc46-dbg	rel_1	2013-10-22 17:46	0 (80)	10/22 17:56	N/A	N/A	N/A	N/A	N/A	10/22 18:15 F	86 (86)	10/22 18:22	tags
	x86_64-slc6-gcc46-opt	rel_1	2013-10-22 17:27	0 (80)	10/22 17:40	N/A	N/A	N/A	N/A	N/A	10/22 17:48 F	86 (86)	10/22 17:58	tags

- goal is to benefit for Run-2 from:
  - ➡ smaller, simpler to maintain and much faster “Conversions” and “I/O” code
  - ➡ new Root6 features and improvements

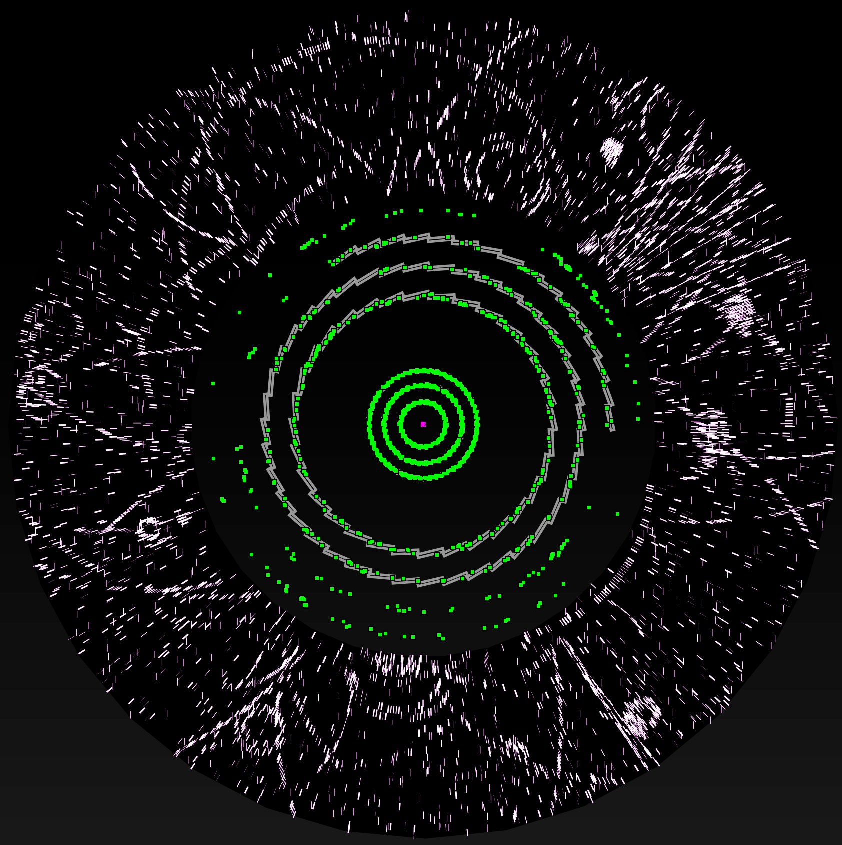
Summary

- ATLAS is running an ambitious software upgrade program in LS1 to prepare for Run-2
  - ➡ new Analysis Model with an all new event format (xAOD)
  - ➡ Integrated Simulation Framework with fast and full simulation in an event
  - ➡ integration of Phase-0 detector upgrades in software chain and algorithmic improvements
  - ➡ code optimization and vectorization, Eigen migration and simplification of tracking EDM
  - ➡ ADC: new GRID production system and data management system
- and we are preparing for the future
  - ➡ R&D on multi-threaded applications, new compilers and hardware technologies

Backups...

Current NewTracking Software Chain

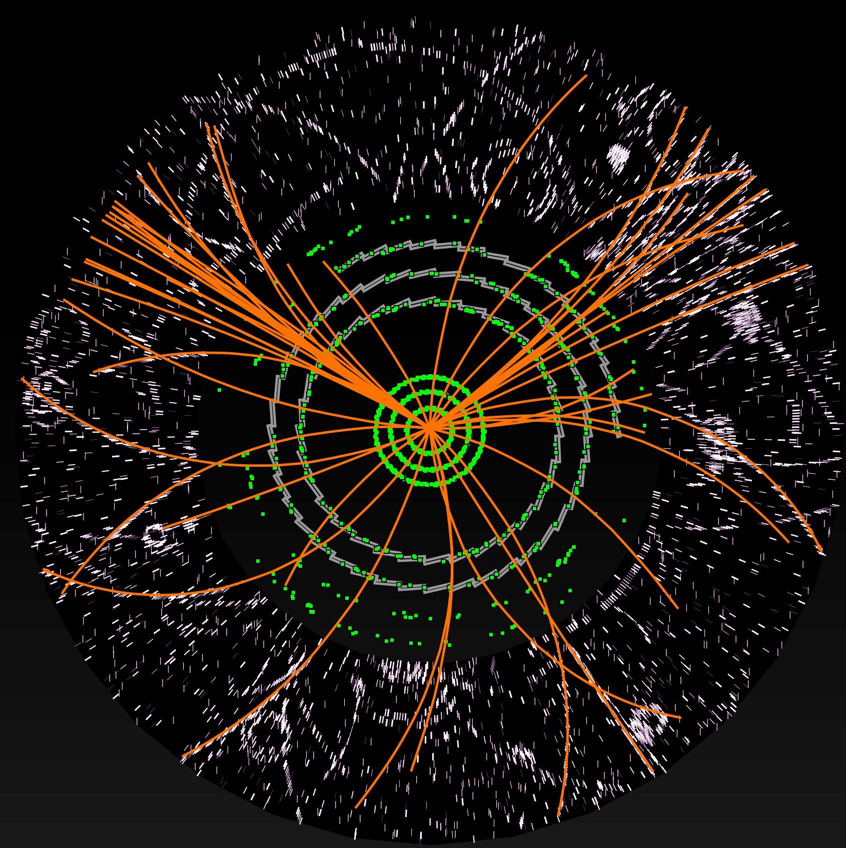
pre-precessing
- ➡ Pixel+SCT clustering
- ➡ TRT drift circle formation
- ➡ space points formation



Current NewTracking Software Chain

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

Current NewTracking Software Chain

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...

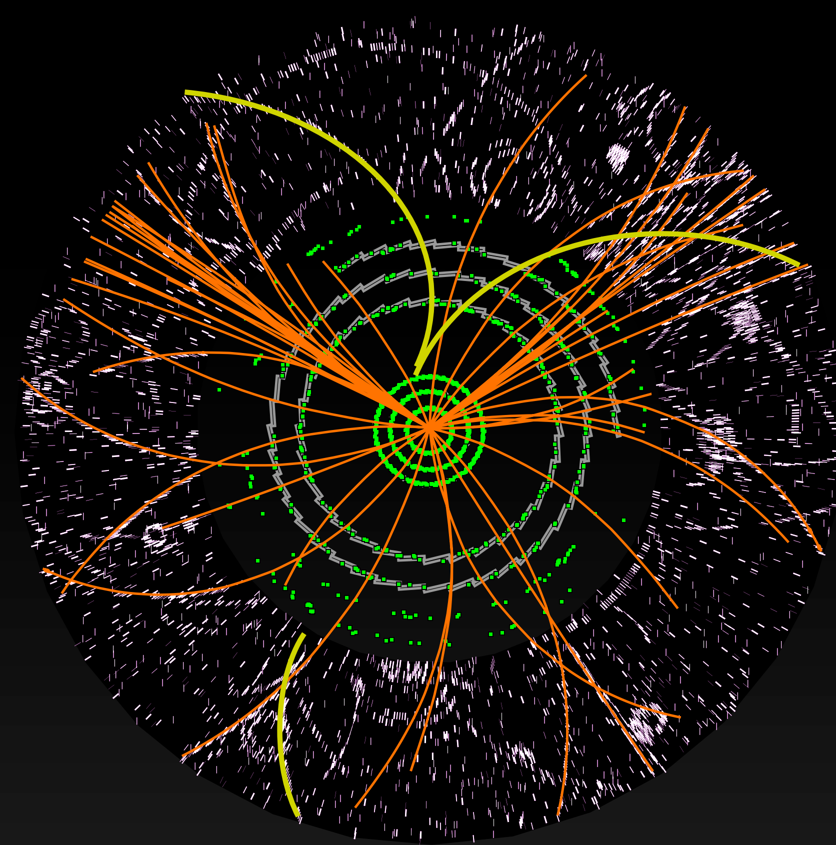
extension into TRT
- ➡ progressive finder
- ➡ refit of track and selection

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



Current NewTracking Software Chain

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

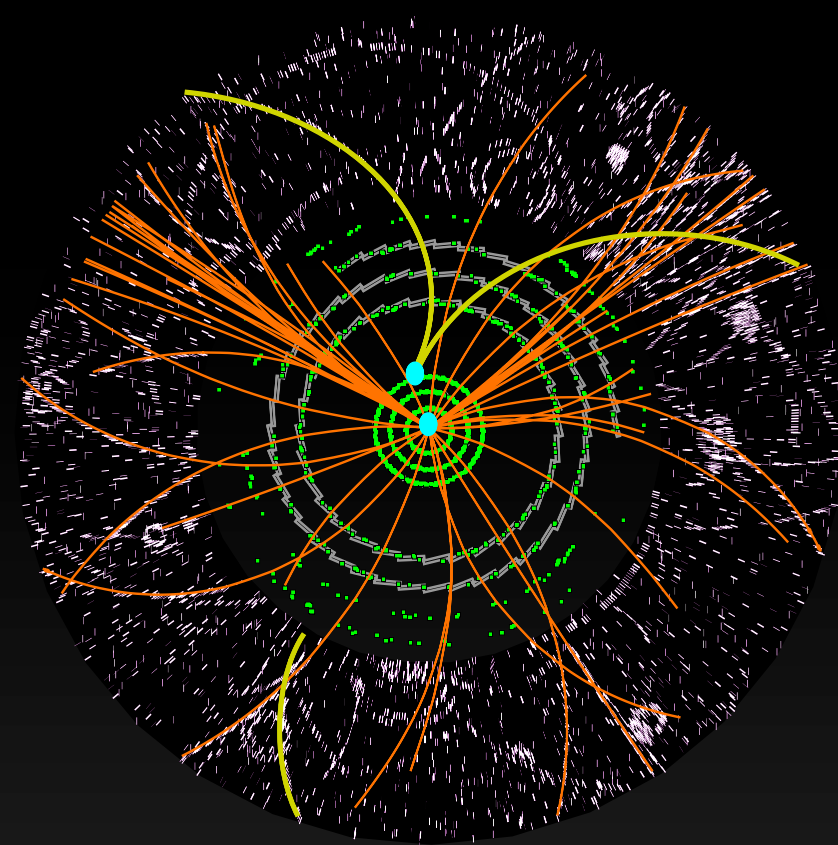
TRT segment finder
- ➡ on remaining drift circles
- ➡ uses Hough transform

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



Current NewTracking Software Chain

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

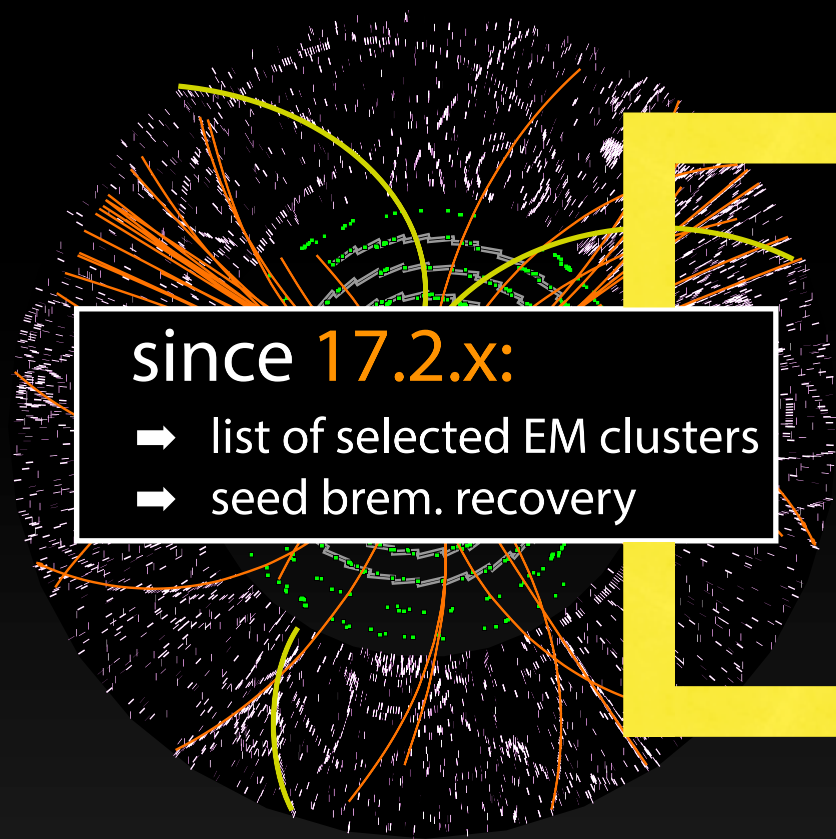
TRT segment finder
- ➡ on remaining drift circles
- ➡ uses Hough transform

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 17.2.x:
- ➡ list of selected EM clusters
- ➡ seed brem. recovery