ATLAS Analysis Model

Amir Farbin
University of Texas, Arlington

Overview

- Present aspects of analysis computing...
  - Context/background
    - Challenges & constraints
  - ATLAS Software Fundamentals
  - Analysis Stages

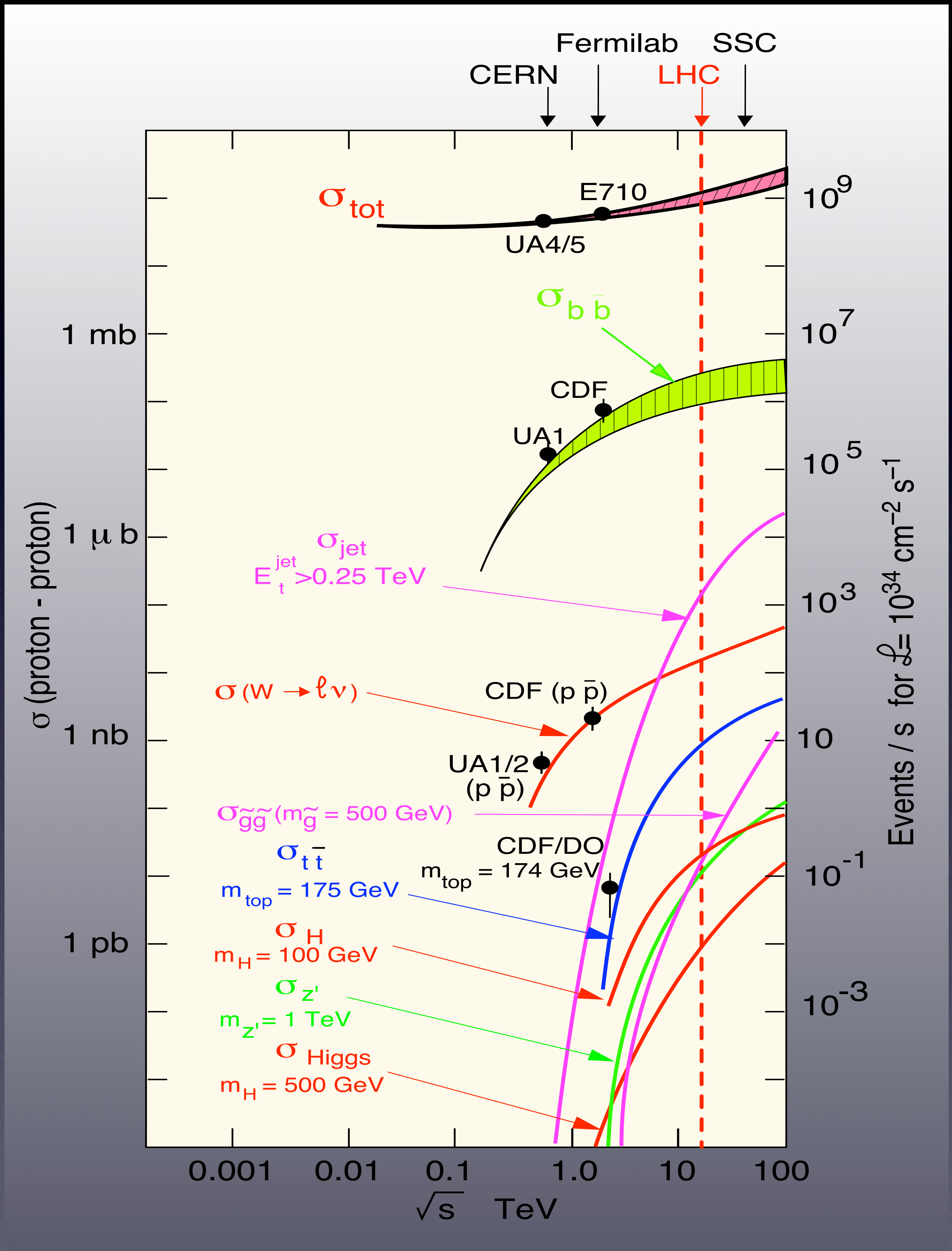
Context

Computing in HEP- Recent History

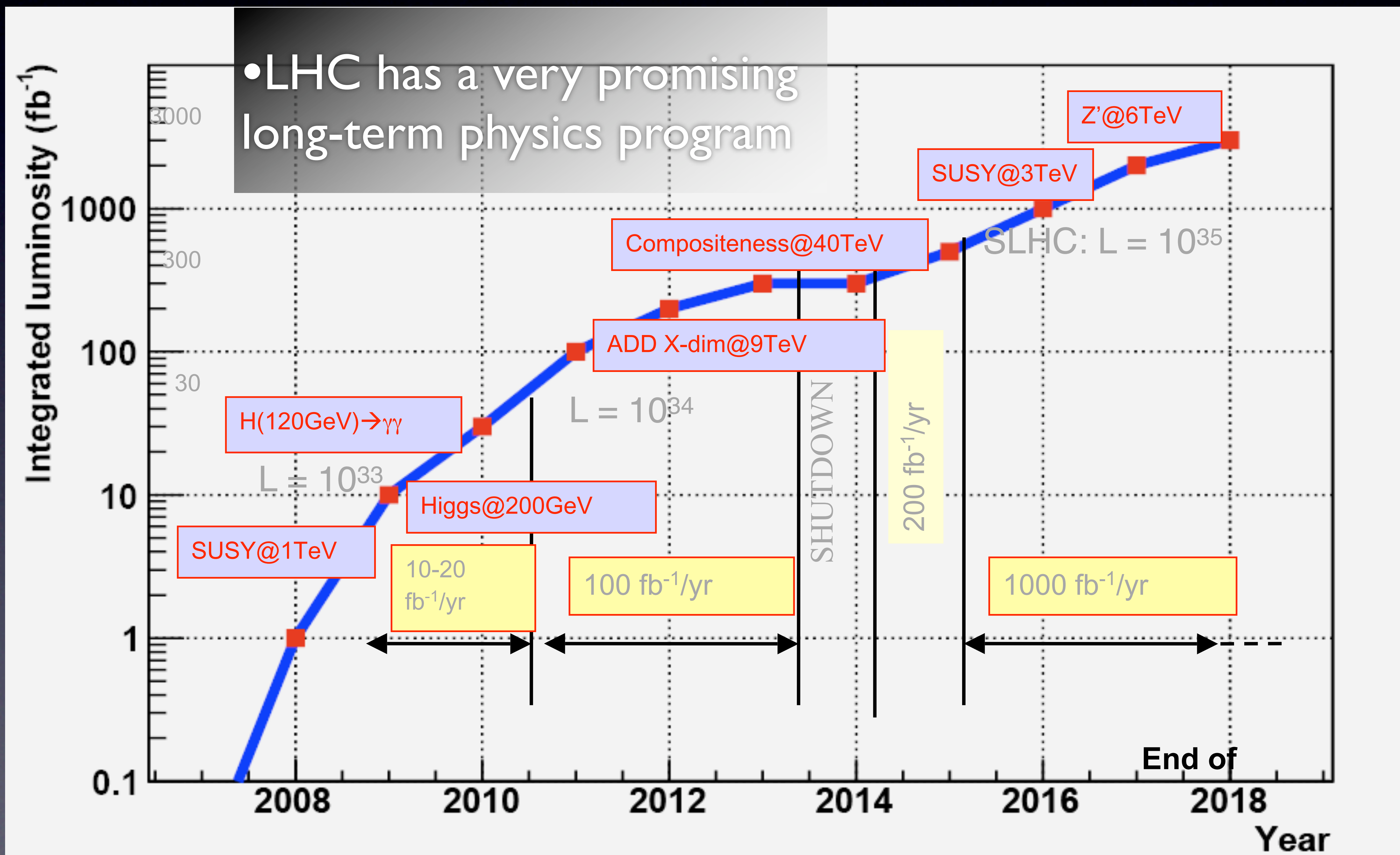
- First generation of experiments with C++ based software are now mature.
  - Tevatron Run II
    - In the beginning, the time from recording an event ➔ available for analysis was months. Now better stream-lined.
    - Hadron collider: Took years to understand detector, accumulate lots luminosity, and publish first results.
  - BaBar
    - Objectivity was a failure... so persistency (technology to write data to disk) had to be redone.
    - At some point, user made ntuples (Derived Physics Data) started dominating the disk... DPD production became primary bottleneck to results.
    - ➡ Computing Model II
      - Root-based persistency, with xrootd based data management.
      - Standardized DPD format based on Event Data Model.
      - Centralized analysis-specific DPD production.
- These experiments have now also transitioned from large computing centers to some GRID use.
- Advanced Analysis tools: Maximum likelihood fitters, multi-variate discriminants, ... better analyses.
- Recent postdocs coming to LHC often understand computing issues and have “modern” computing experience (not afraid of C++)...

LHC Data

- At L=10^{34} cm^{-2} s^{-1} (~100x less at startup):
  - W→lν, Z→ll ~10^2 Hz
  - top at 10 Hz
  - Higgs at 1- 10^{-1} Hz (m_H=100 - 600 GeV)
  - SUSY up to 10 Hz (depending on scale)
- At full lumi, 23 simultaneous interactions (pileup)
- 200 Hz trigger output from the beginning (1.6MB/event)
  - Sig/Bkg ratio increase with higher luminosity... requires understanding.
- Significant increase in SM x-sections over Tevatron ⇒ Lots of control samples to quickly:
  - Understand detector
  - Tune MC to 14 TeV
- Great potential for early discovery.
- Important to get things working from the beginning.

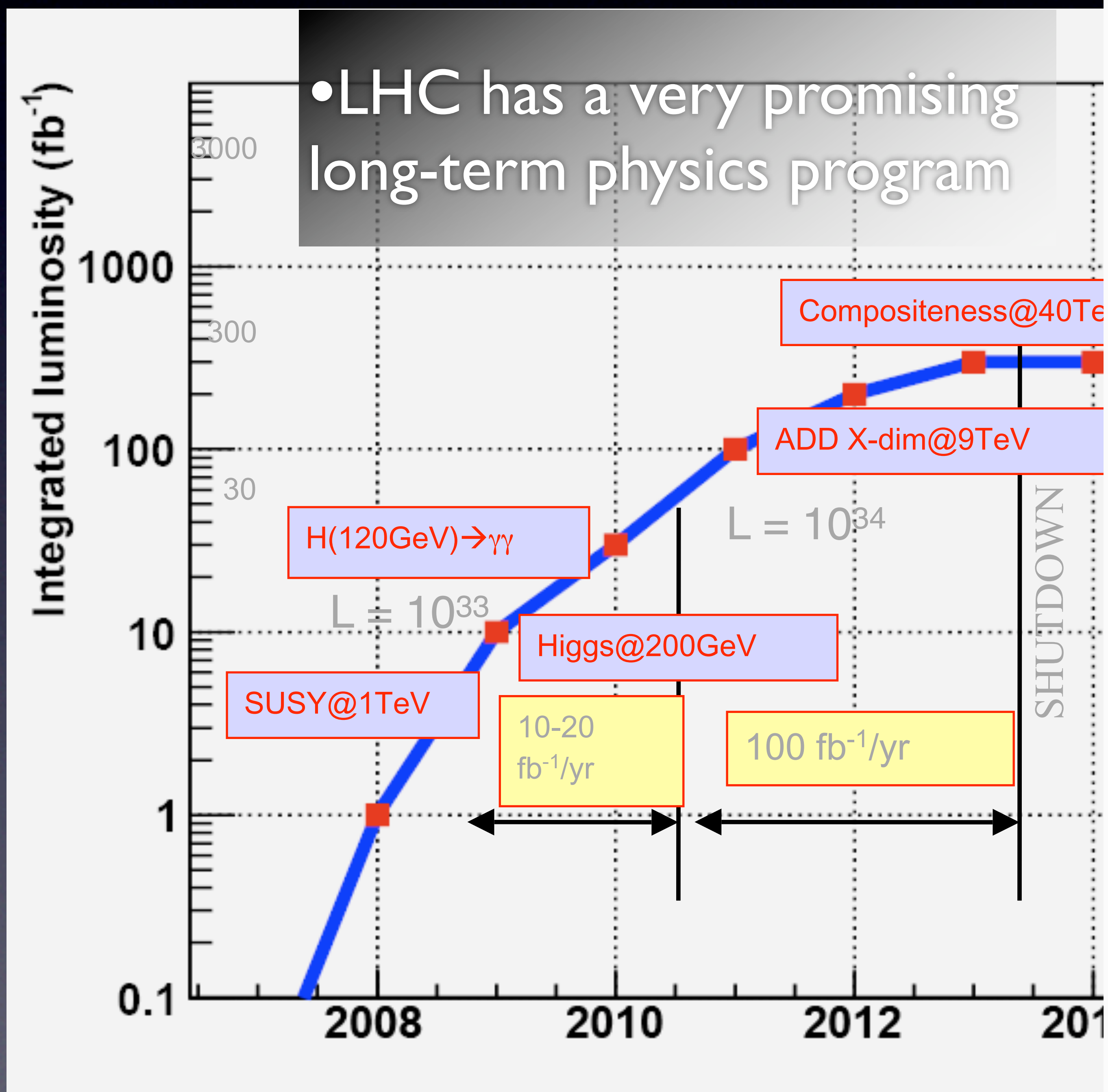


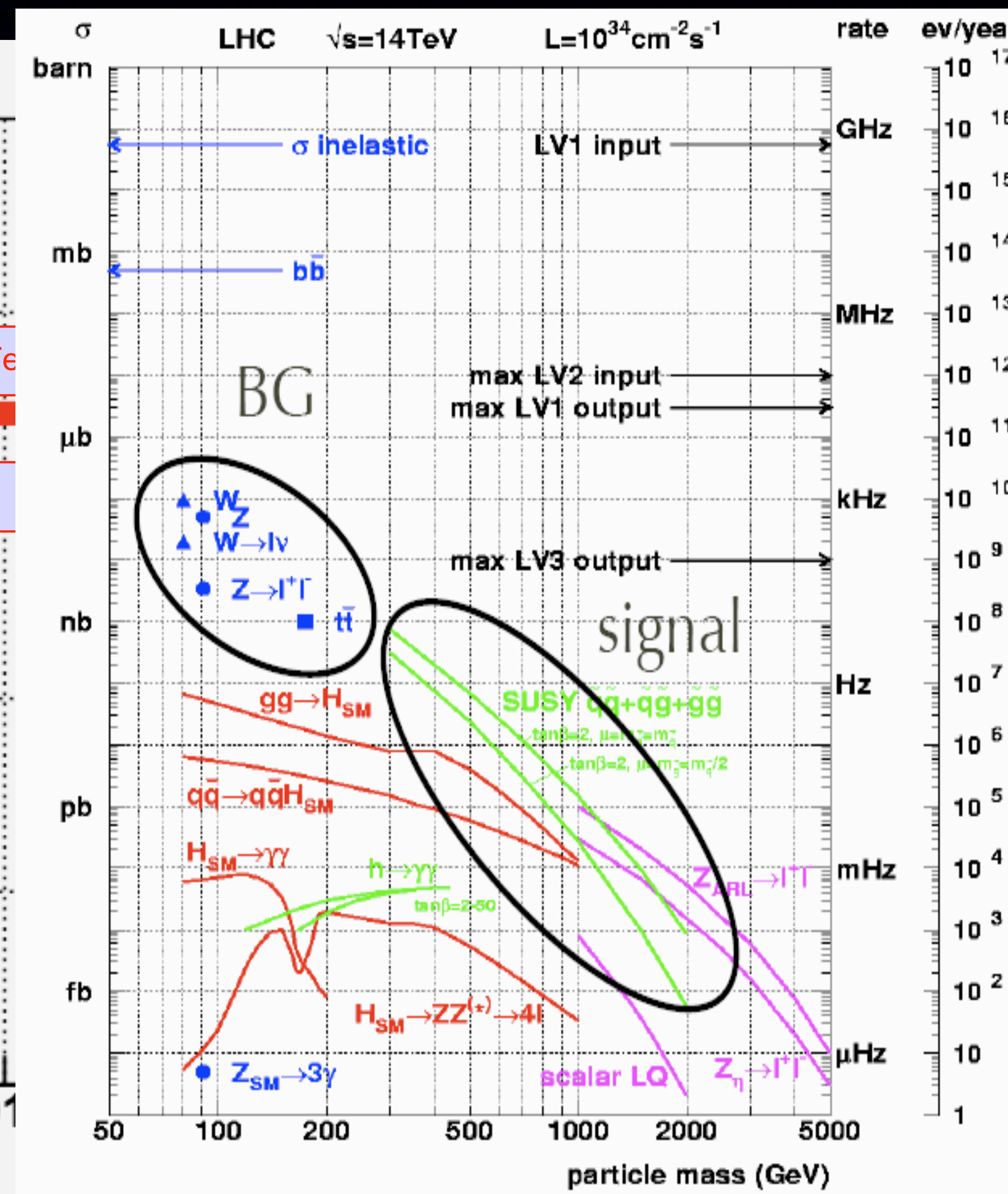
New Physics in 2009?



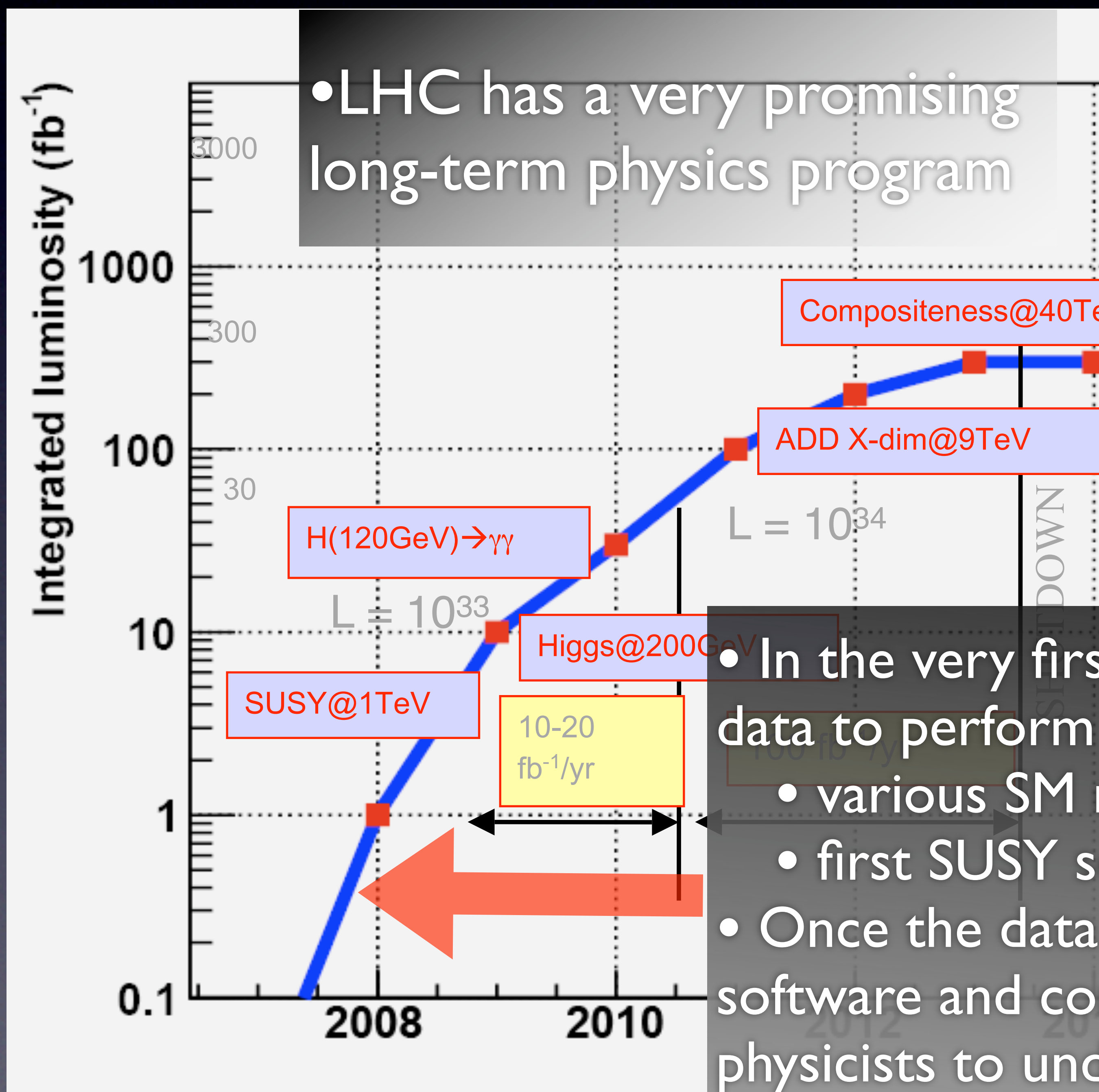
New Physics in 2009?

- LHC has a very promising long-term physics program

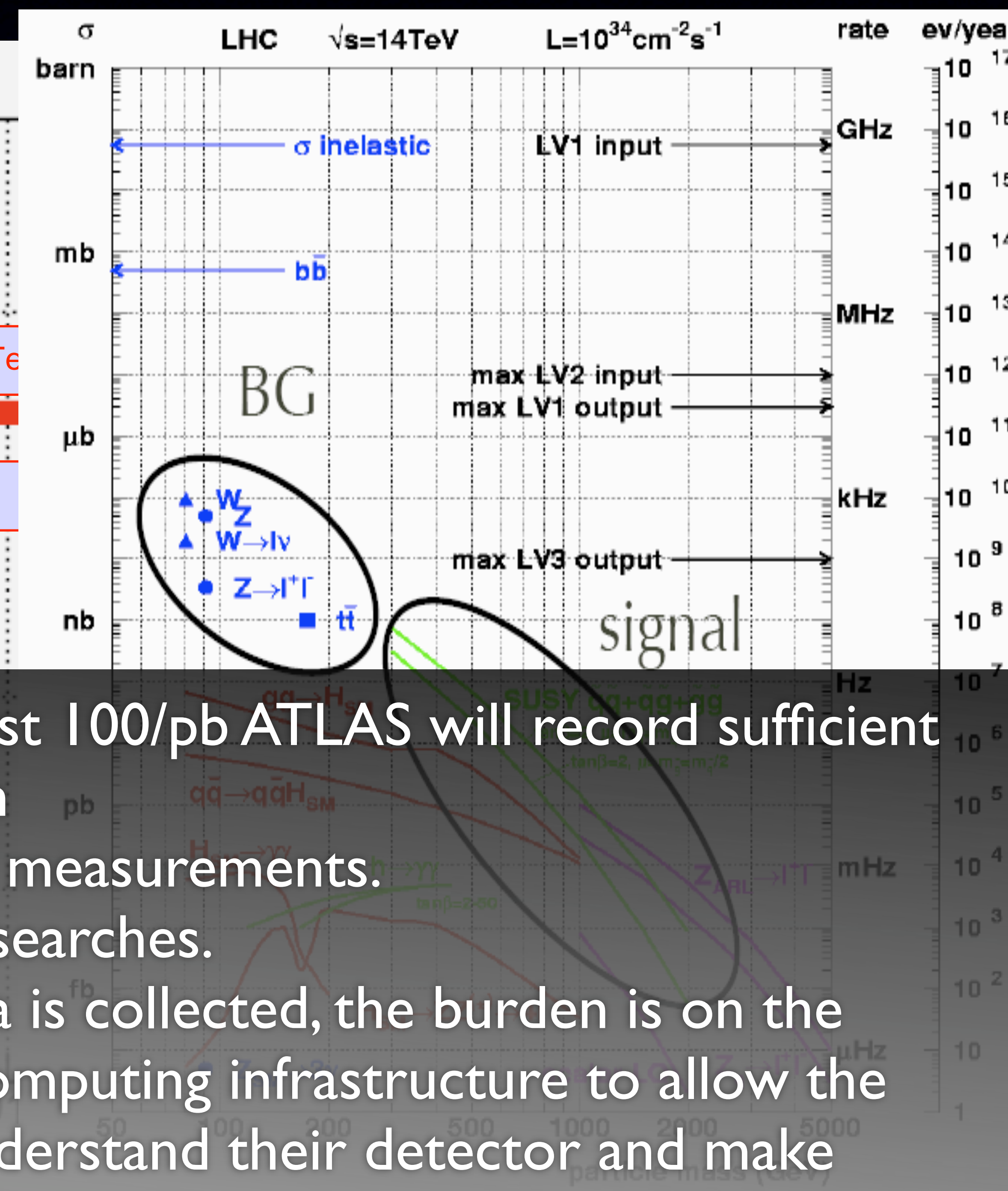




New Physics in 2009?



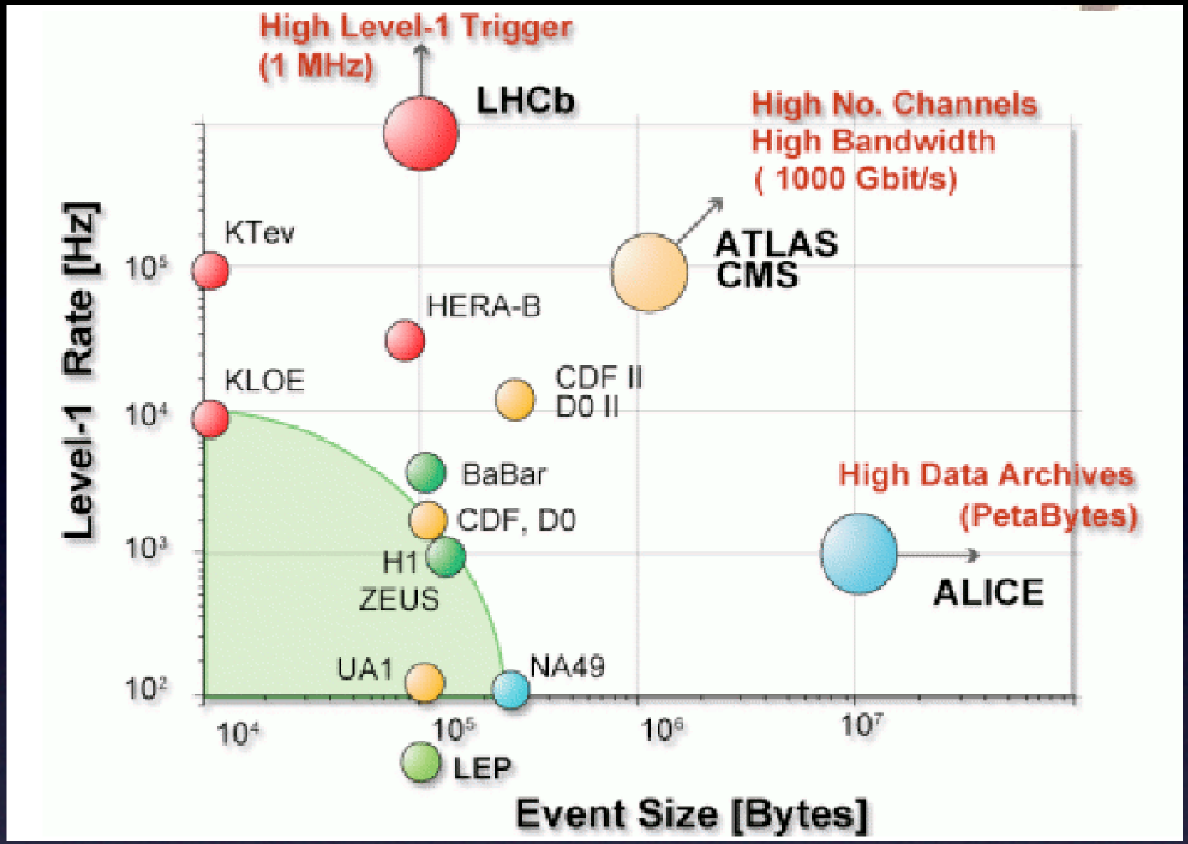
•LHC has a very promising long-term physics program



- In the very first 100/pb ATLAS will record sufficient data to perform
  - various SM measurements.
  - first SUSY searches.
- Once the data is collected, the burden is on the software and computing infrastructure to allow the physicists to understand their detector and make measurements.

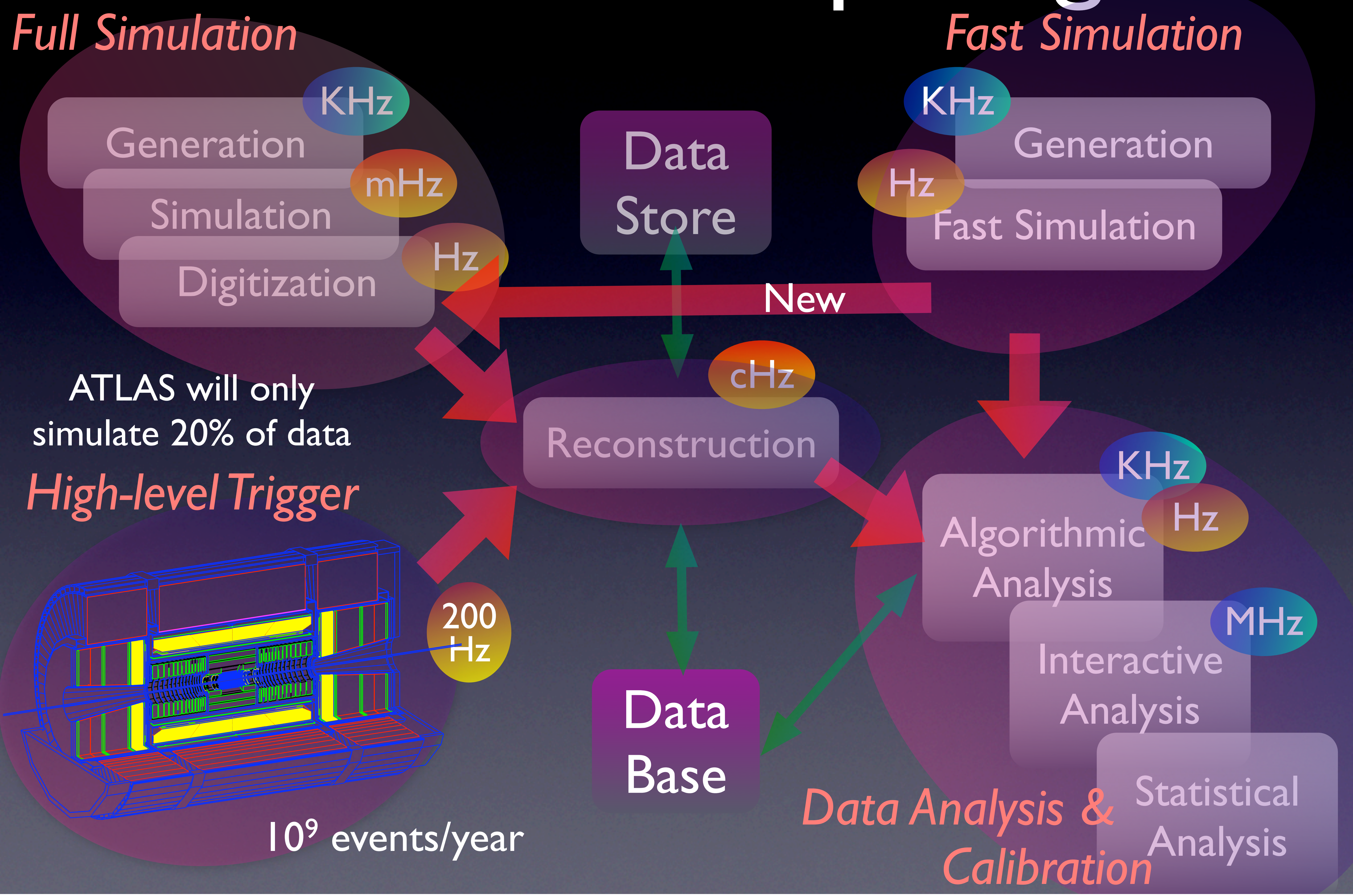
Computing at LHC

- Bigger challenges
  - More complex detectors (O(10) times more channels), environment (beam cross rate, pile up)
    - High rate, large events...
  - HEP is moving from O(500) person experiments to O(2000).
    - Variety of experiences:
      - from UA1/2 and LEP, to Tevatron and BaBar.
      - Lots of conflicting opinions making consensus increasingly difficult.
    - Management + SW challenge to establish and deploy standard procedures.
    - Greater division between software gurus and average users.

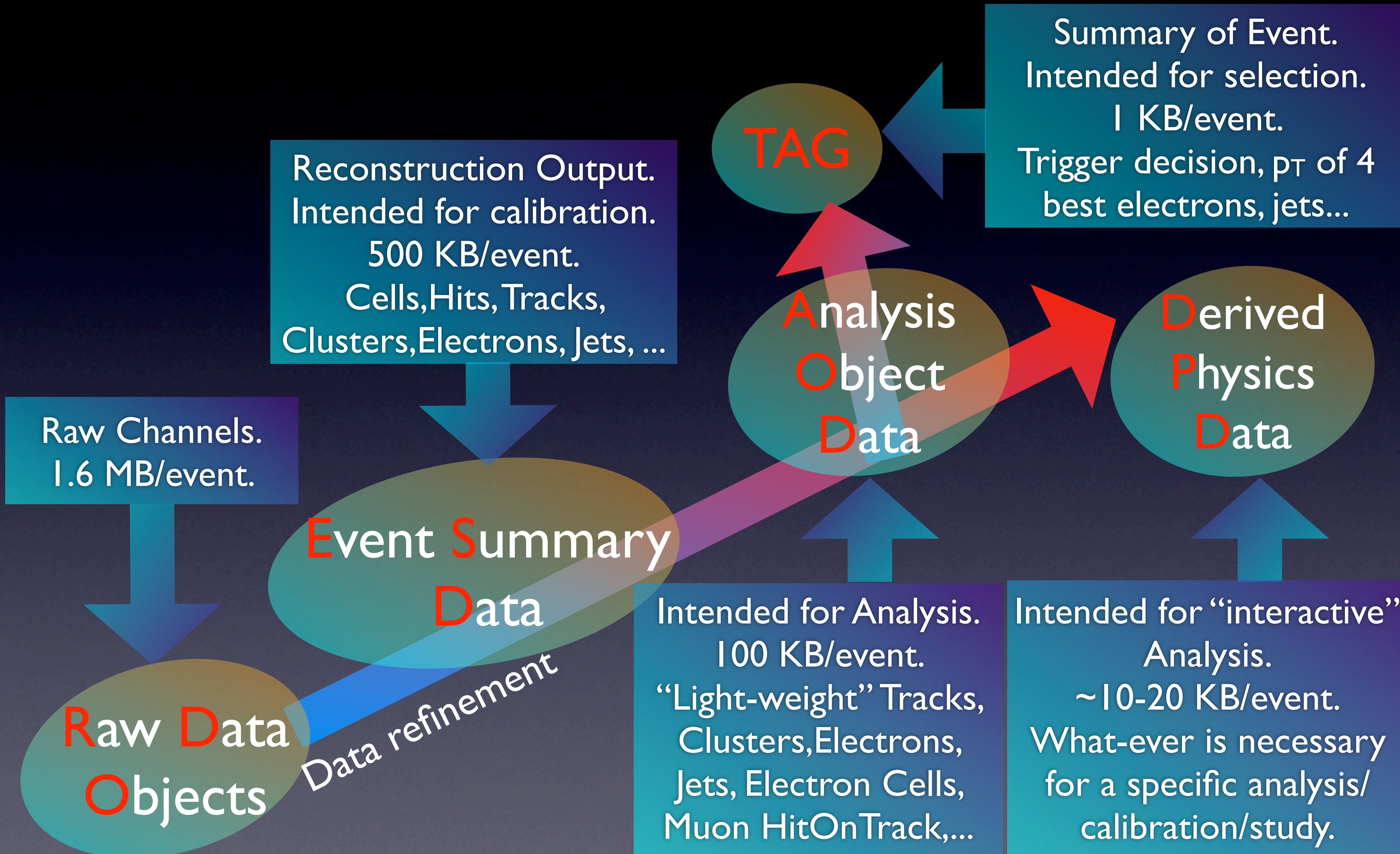


- Computing models based on globally distributed, locally funded, multi-tier computing/data storage
  - using GRID middle-ware on 3 different GRID implementations
  - + experiment specific software infrastructure.
- No more arguments about Fortran vs C++... now it is C++ vs python!

ATLAS Computing



The Event Data Model



The Computing Model

- Resources Spread Around the GRID

•Interactive Analysis
•Plots, Fits, Toy MC, Studies, ...

Tier 3

DPD

•Reprocessing of full data with improved calibrations 2 months after data taking.
•Managed Tape Access: RAW, ESD
•Disk Access: AOD, fraction of ESD

30 Sites Worldwide

Tier 2

AOD

•Derive 1st pass calibrations within 24 hours.
•Reconstruct rest of the data keeping up with data taking.

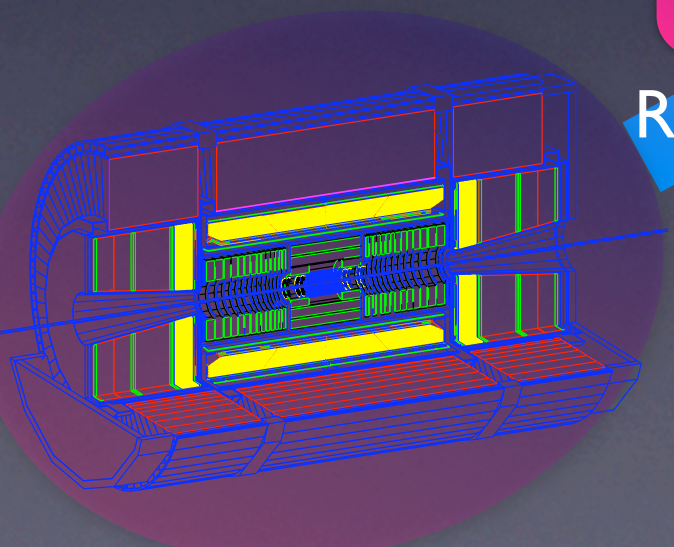
Tier 1

10 Sites Worldwide

RAW/
AOD/
ESD

Tier 0

RAW



CERN Analysis Facility

•Production of simulated events.
•User Analysis: 12 CPU/Analyzer
•Disk Store: AOD

•Primary purpose: calibrations
•Small subset of collaboration will have access to full ESD.
•Limited Access to RAW Data.

Physics Analysis

- Physics Analysis is where many users first encounter the software framework.
- For many, this is very "personal"...
    - prefer to do things on their own
    - but realistic analysis quickly become complicated... inefficient for everyone to do everything themselves
    - and difficult to establish common procedure, compare results, ...
- What can we learn from previous experiments?

Lessons from Other Experiments I

Observations from:
BaBar, CDF, D0, H1
ATLAS Analysis Model Workshop (Oct 2006)

- *Observation:* Speed is the most important factor in the Analysis Model adopted by users... no matter what the management says or sw-developers provide.
- When it is impractical to repeatedly iterate analyses on AOD, users dump large ntuples (DPD) which mostly copy AOD contents... and perform analysis outside the software framework.
- Solution:
  - Optimize AOD access speed to can close to the ROOT limit (10MB/s). (Transient/Persistent Separation)
  - Allow direct access to data written by the framework in ROOT...
    - AOD can be read directly in ROOT
    - DPD can be written by framework, read by ROOT

Lessons from Other Experiments II

Observations from:
BaBar, CDF, D0, H1
ATLAS Analysis Model Workshop (Oct 2006)

- *Observation:* Tasks naively thought to be addressed by "ESD"-based analysis or reprocessing (eg: calibration, alignment, track-fit, re-clustering) are routinely performed in the highest level of analysis.
  - ➡ As experiments evolve:
    - "ESD" bloated and too difficult to access ⇒ dropped
    - "AOD" is gradually augmented with some "ESD" quantities (eg: hits in roads/cells) to provide greater functionality at analysis time.
- Solution:
  - Make sure reconstruction and calibration can be applied to AOD objects.
  - Make it easy to adjust the content of the AOD.
  - Add sufficient information to the AOD permit foreseen analysis tasks. Lots of recent iterations on AOD content in the context of analysis model.

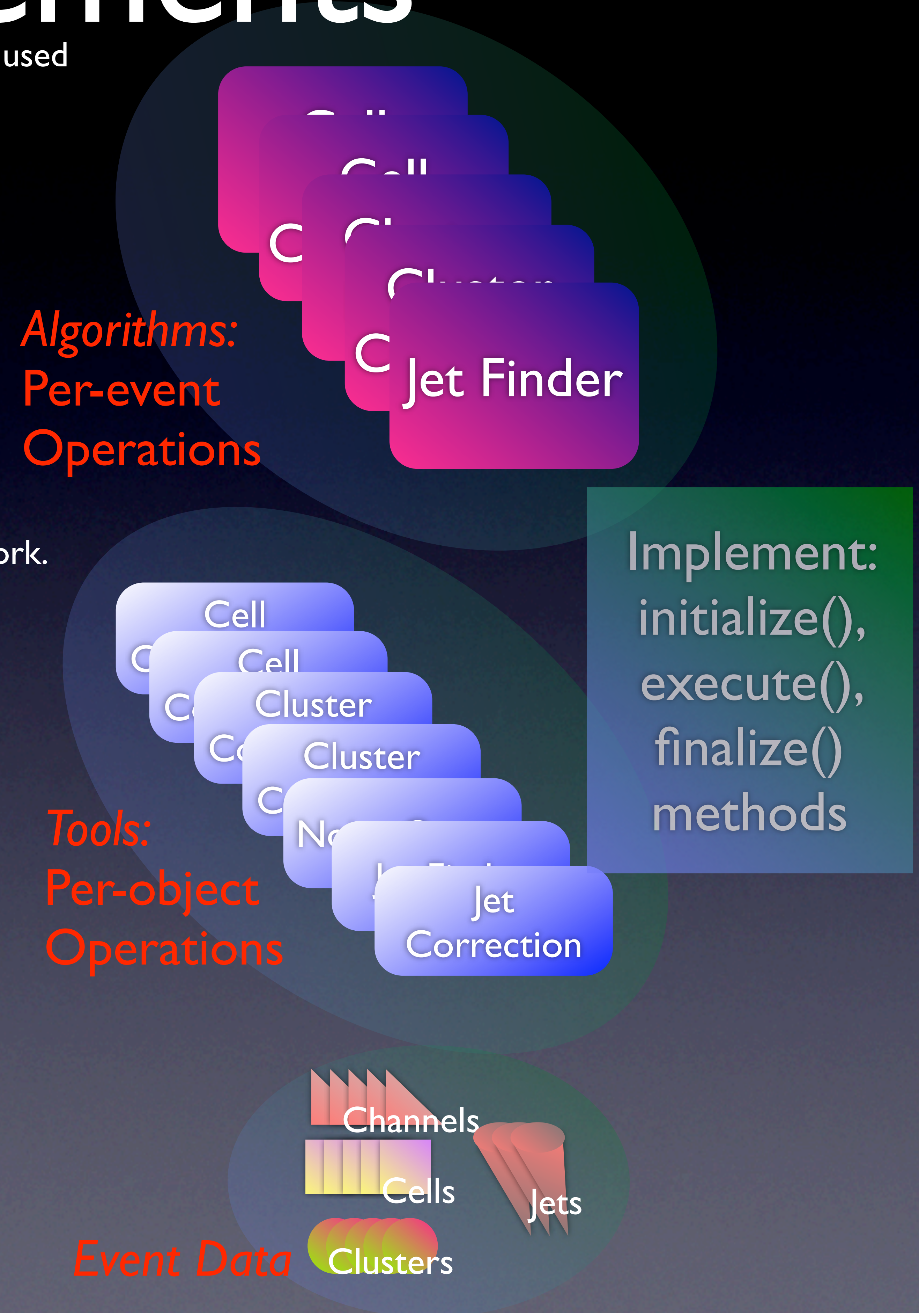
Lessons from Other Experiments III

- *Observation:* As experiments mature, physics groups (eg Top, SUSY,...) or analysis groups (eg graviton to diphoton search) converge on common analysis software which produces common DPDs.
  - Often this naturally occurs after years of data-taking and lots of trial and error.
  - Decentralized DPD production is eventually replaced by organized/centralized production.
  - The accepted solution at the end often is a reflection of the path taken... would have done it differently if starting again today.
- Solution: As soon as possible:
  - Develop a common DPD format.
  - Provide common DPD making tools.
  - Make it all very flexible...
  - Centralize DPD production.

ATLAS Software Fundamentals

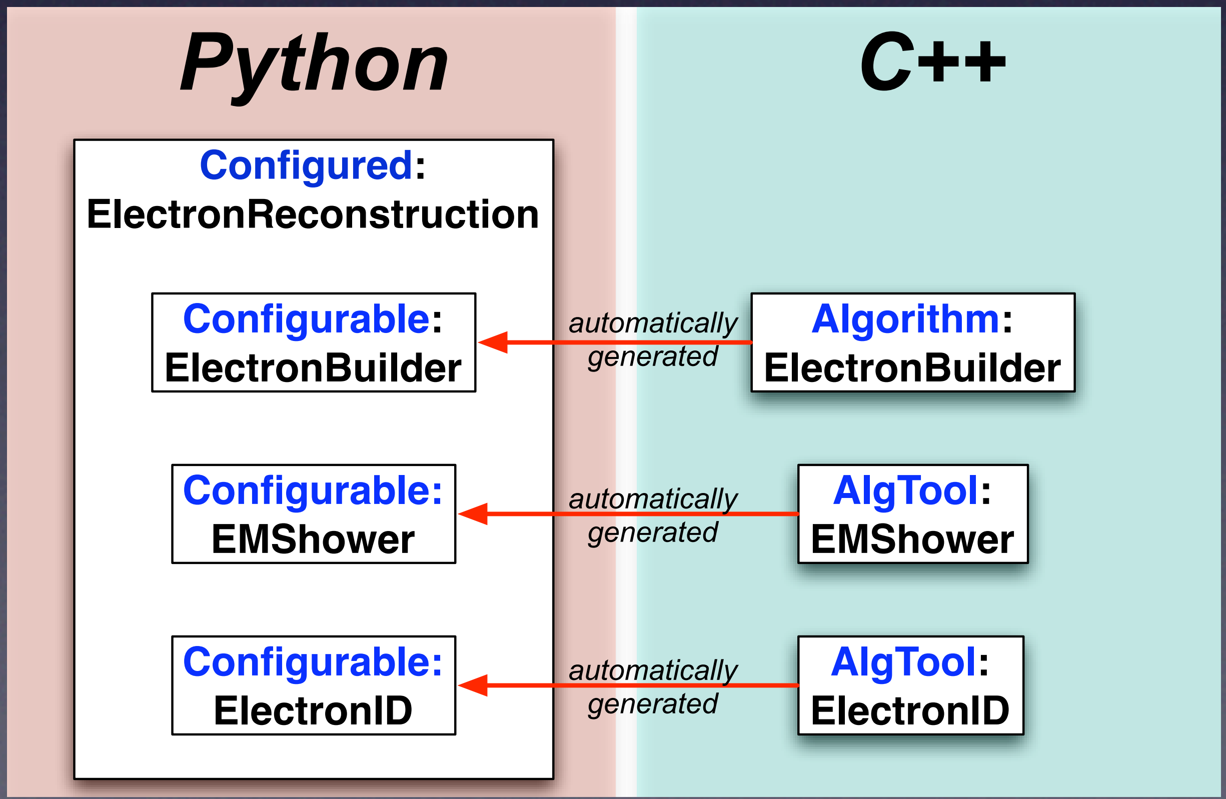
Framework Elements

- Athena is an extended version of LHCb’s Gaudi framework used for high-level trigger, simulation/reconstruction, and analysis.
- Principles... separation of:
  - Data and algorithms
  - Transient (in memory) and Persistent (on disk) data (in contrast to CMS)
- Elements:
  - *Algorithms*- one execute per event, managed by framework.
  - *Tools*- multiple executes per event.
  - Event Data
  - Services
    - StoreGate- Transient Data Store- Mechanism for communication between Algorithms
    - Tool Service- Tool Factory
    - Interval of validity
    - Histogram Service
    - POOL- Persistency

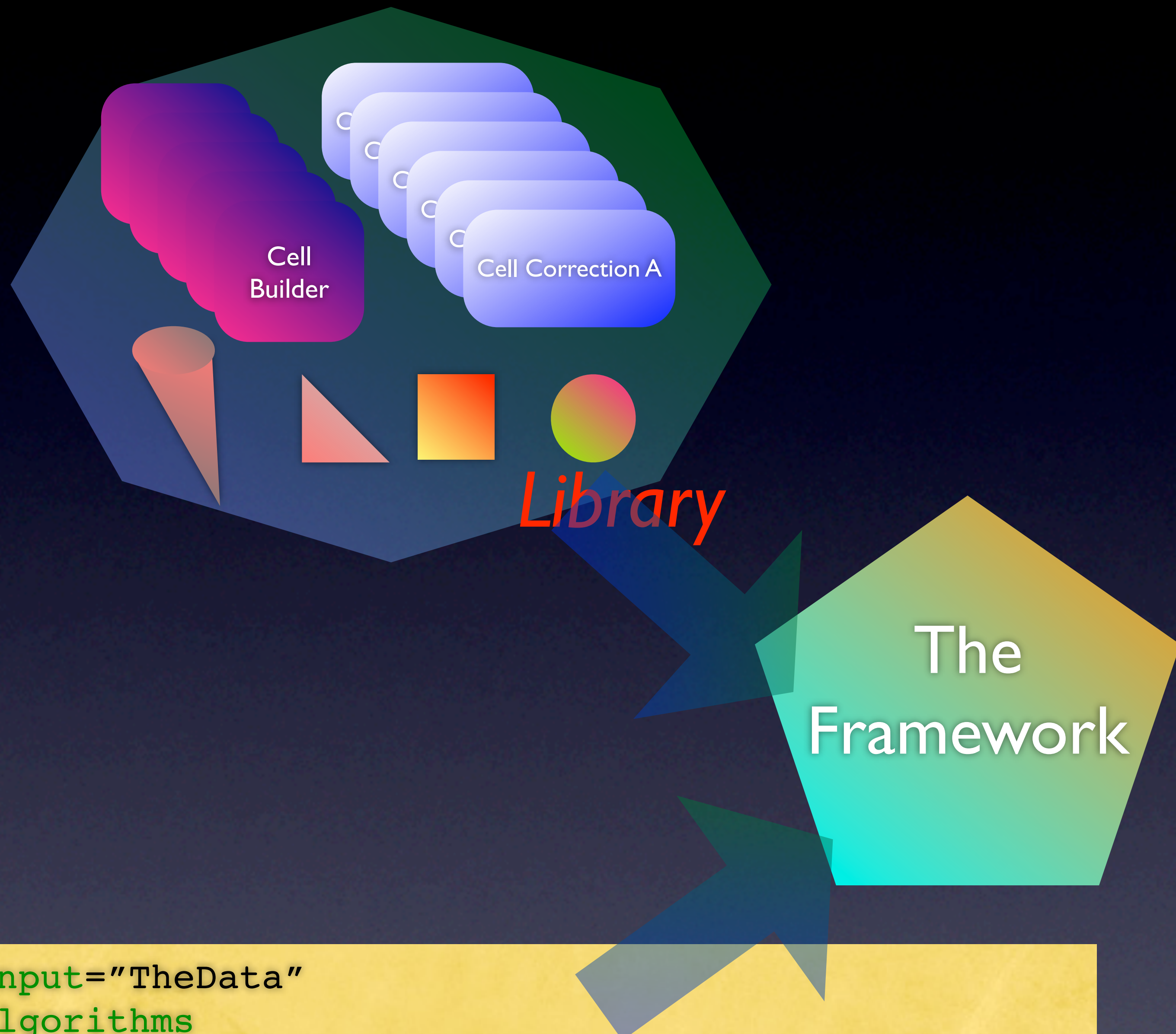


Configuration

- Framework elements (eg Algorithms, Tools, Services) declare properties which can be set at runtime
- Application defined in python:
    - Load libraries
    - Instantiate tools/algs, configure properties
    - Define input/output
- Configurables:
    - Auto-generated python reflection of C++ components
    - Build configuration purely in python, persistify the configuration, build application later.
    - Build higher level abstractions in python



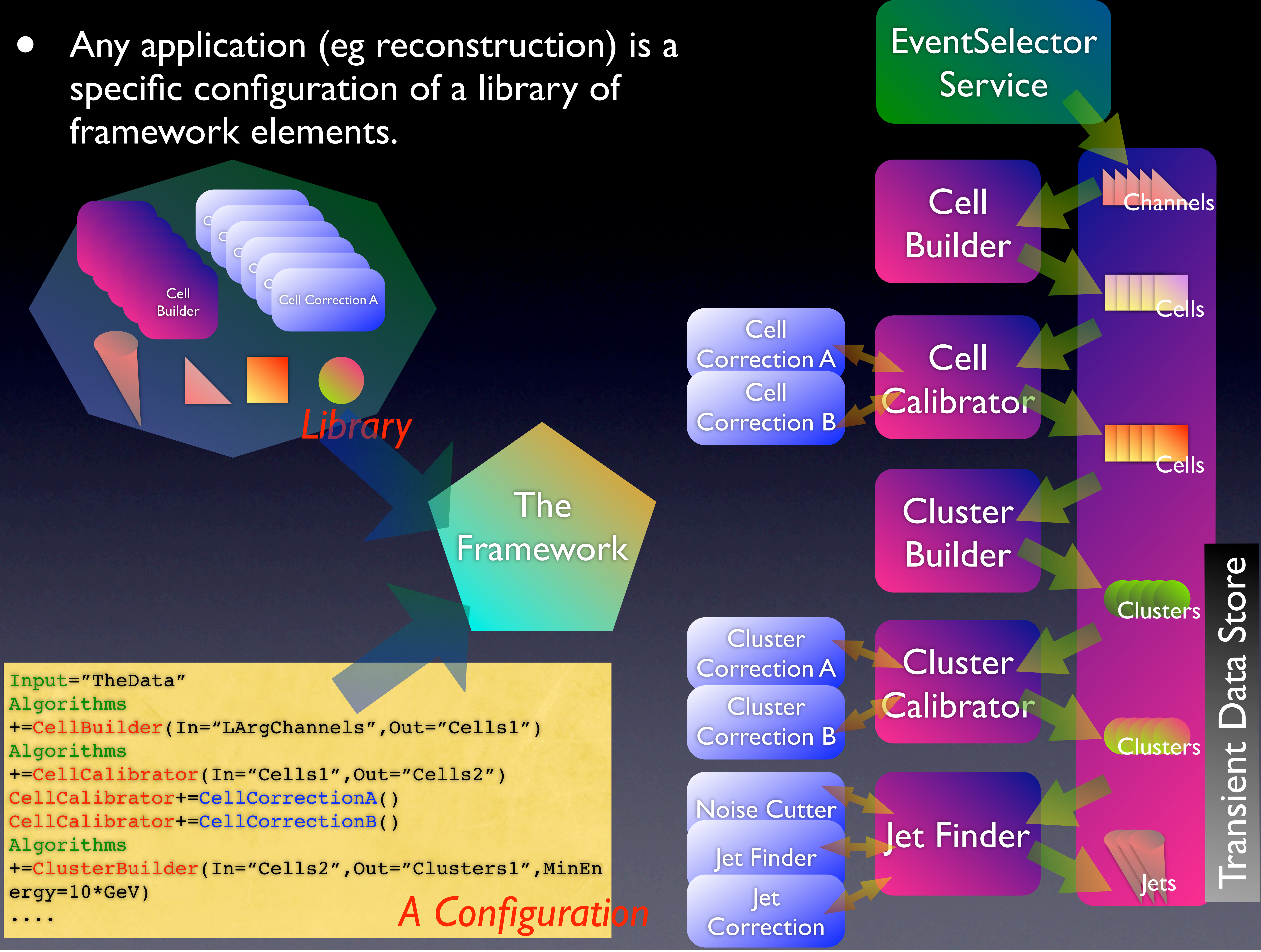
- Any application (eg reconstruction) is a specific configuration of a library of framework elements.




```
Input="TheData"
Algorithms
+=CellBuilder(In="LArgChannels",Out="Cells1")
Algorithms
+=CellCalibrator(In="Cells1",Out="Cells2")
CellCalibrator+=CellCorrectionA()
CellCalibrator+=CellCorrectionB()
Algorithms
+=ClusterBuilder(In="Cells2",Out="Clusters1",MinEn
ergy=10*GeV)
....
```

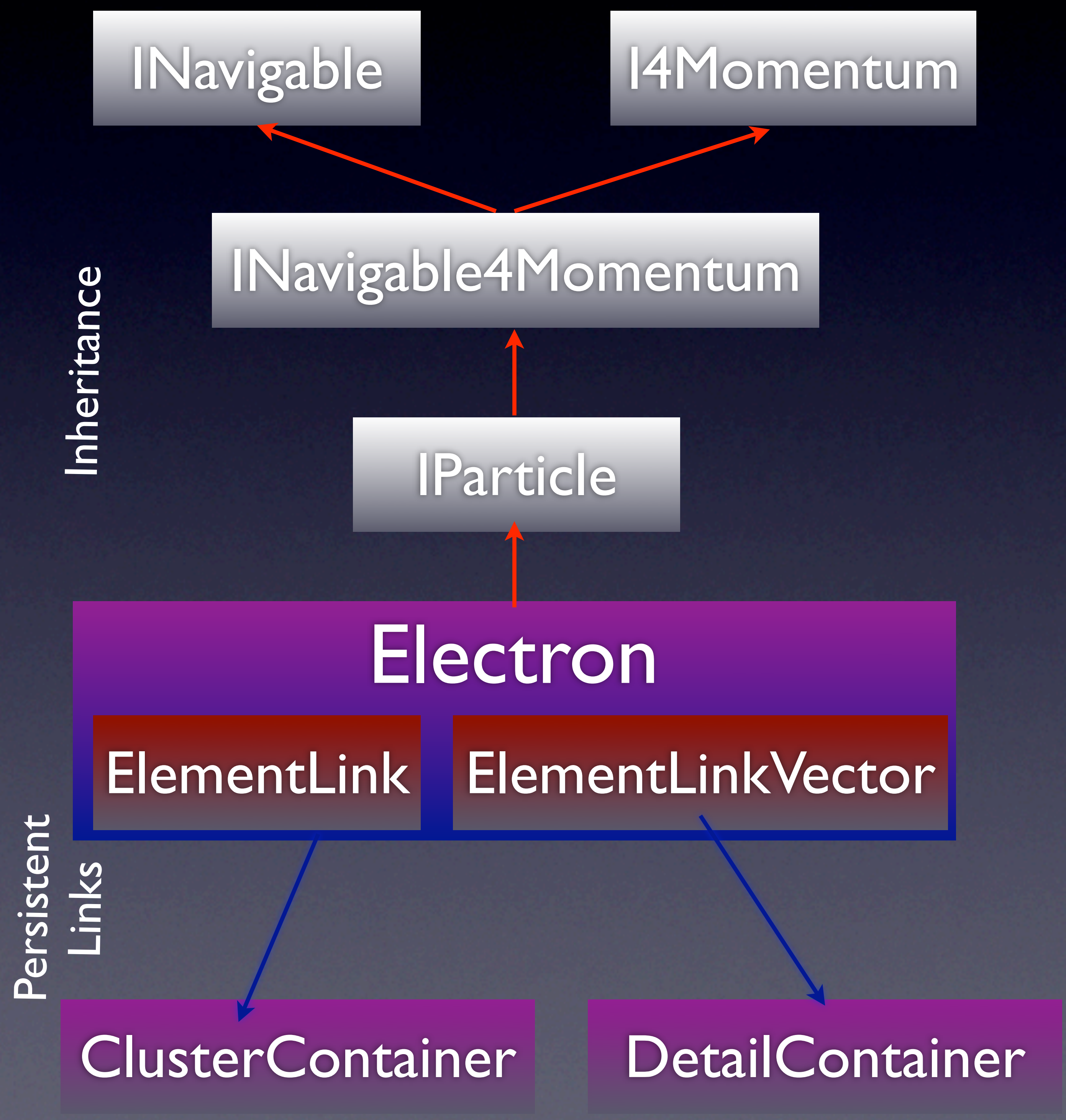
A Configuration

- Any application (eg reconstruction) is a specific configuration of a library of framework elements.



Event Data Model

- Particle like objects share common interface for
  - 4-momentum representation
  - navigation to other objects
- Links between objects implemented via ElementLinks
  - Persistifiable pointers
  - Retrieves data from StoreGate
    - On demand access
    - Works across files



Analysis Stages

What is Analysis?

- *Re-reconstruction/re-calibration*- often necessary.
- *Algorithmic Analysis*: Data Manipulations ESD→AOD→DPD→DPD
  - *Skimming*- Keep interesting events
  - *Thinning*- Keep interesting objects in events
  - *Slimming*- Keep interesting info in objects
  - *Reduction*- Build higher-level data which encapsulates results of algorithms
  - Basic principle: Smaller data → more portable & faster read
- *Interactive Analysis*: Making plots/performing studies on highly reduced data.
- *Statistical Analysis*: Perform fits, produce toy Monte Carlos, calculate significance.

Tier 1/2

Tier 3

Stages in Analysis

- Use TAG to quickly select subset of events which are interesting for analysis. (Skim)
- Starting from the AOD

Physics Group

- Stage 0: Re-reconstruction, re-calibration, selection (AOD)
  - Redo some clustering/track fitting, calculate shower shapes, apply corrections, etc...
  - Typical: 250 ms/event, In: 75% AOD, out 50% AOD

Analysis Group

- Stage 1: Selection/Overlap removal/complicated analysis (AOD/DPD)
  - Select electrons/photons→find jets on remaining clusters→b-tag→calculate MET
  - Perform observable calculation, combinatorics + kinematic fitting, ...
  - Typical: 20 ms/event, In: 25% AOD, Out: 10% AOD

Personal

- Stage 2: Interactive analysis (AOD/DPD)
  - Final selections, plots, studies.
  - Prototype earlier steps!
  - Typical: 0 ms/event, In: 1% AOD, Out: 0
- Stage 3: Statistical Analysis

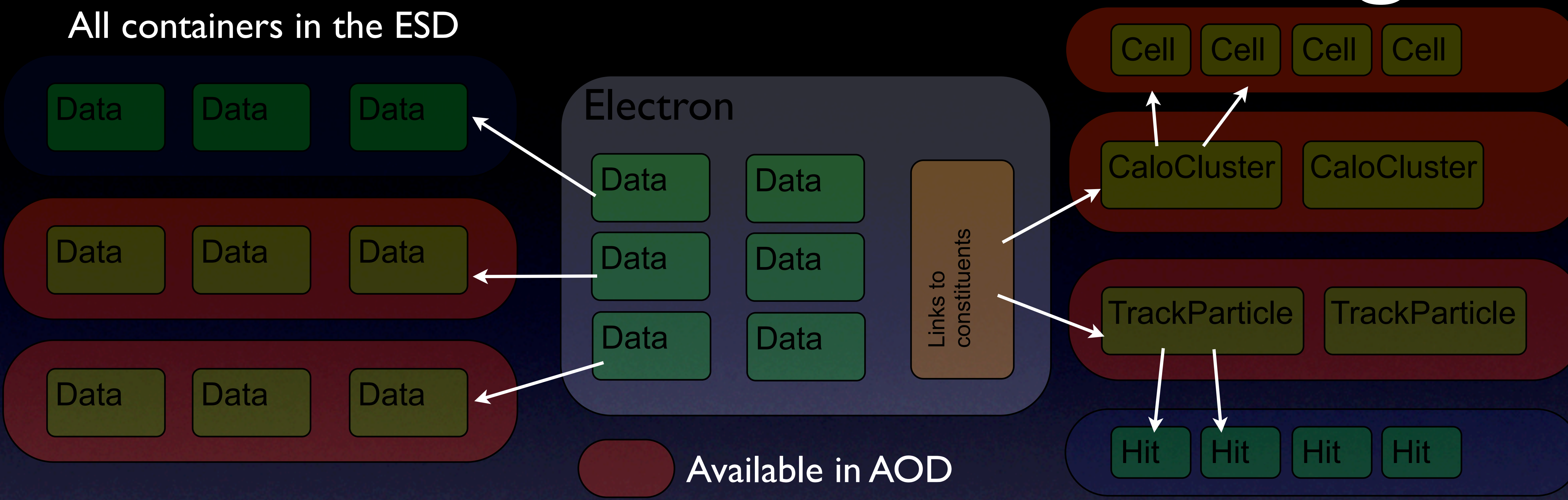
Stages vs Resources

- ATLAS will record 200 Hz of data, regardless of luminosity ➔ 10^9 event/year.
- CM Assumption 700 Analyzers: 12 tier 2 CPU/person for analysis at any give time.
- Not unusual for some analysis to start with 50% of the data.
- Assuming perfect software/hardware (10 MB/s read in = ROOT limit).

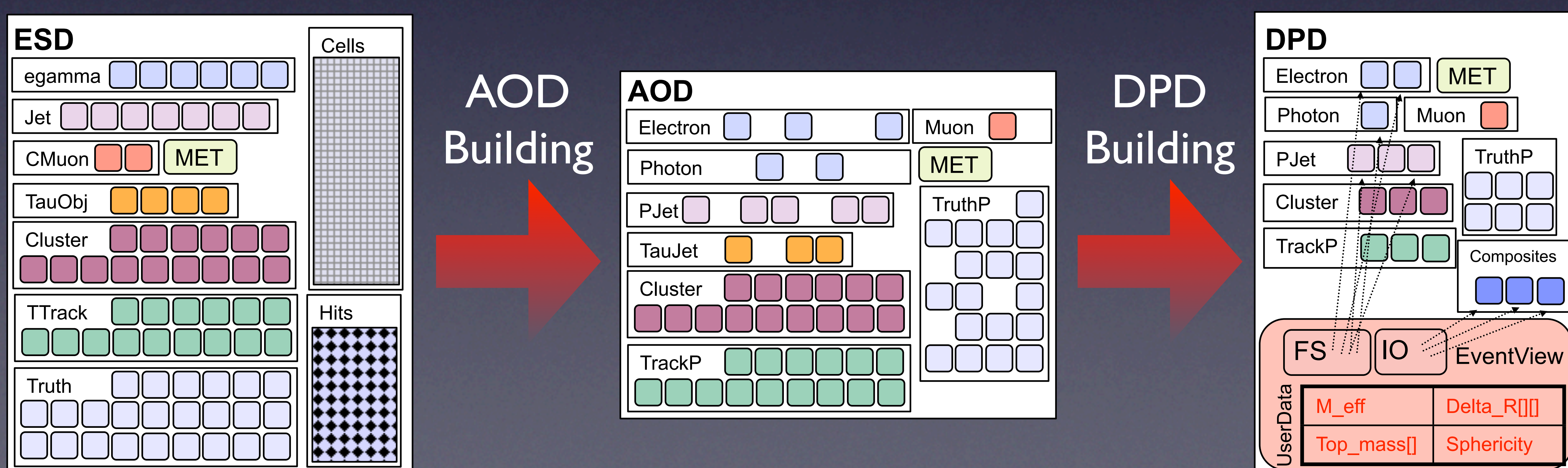
Step		Laptop 1 Core	Tier 3 25 Cores	Tier 2 10 Persons 100 Cores	Tier 2 100 Persons 1000 Cores	
Step 0	1 Hour	< 0.0001%	0.0035%	0.0140%	0.1398%	Working group on Tier 2
	Overnight	0.0017%	0.0419%	0.1678%	1.6777%	
	1 Week	0.0235%	0.5872%	2.3487%	23.4874%	
	1 Month	0.1007%	2.5165%	10.0660%	All	
Step 1	1 Hour	0.0016%	0.0400%	0.1600%	1.6000%	Analysis group on Tier 2
	Overnight	0.0192%	0.4800%	1.9200%	19.2000%	
	1 Week	0.2688%	6.7200%	26.8800%	All	
	1 Month	1.1520%	28.8000%	All	All	
Step 2	1 Hour	0.3600%	9.0000%	36.0000%	All	Single Analyzer on Tier 3
	Overnight	4.3200%	All	All	All	
	1 Week	60.4800%	All	All	All	
	1 Month	All	All	All	All	

Stage 0: Re-re construction, Recalibration

Event Data Model Design



Benefits: 1. Move data between ESD/AOD/DPD w/o schema change.   2. Read on Demand



Event Data Model Content

EDM Level	Contents	Primary Intent	Size/ Event (KB)	Max Ideal Input rate (Hz)	Access-ibility
Raw Data Objects	Raw Channels	Reconstruction (calibration)	1600	N/A	Central Reco/ Reprocessing: Tier 0/1
Event Summary Data	Cells, Hits, Clusters, Tracks, MET, Electron, Jet, Muon, Tau, Truth	Derive calibrations, Re-reconstruction, Re-calibration	500		CERN CAF (access limited), Tier 1 (on tape)
Analysis Object Data	Lepton Cells, Hits, Clusters, Tracks, MET, Electron, Jet, Muon, Tau, Slimmed Truth	Limited Re-reconstruction (eg Jets, b-tag), limited re-calibration, Analysis	100	1000	Full: Tier 1,2 (disk) Subset: Tier 3
Derived Physics Data	Any of the above + composites (eg top) + derived quantities (sphericity)	Interactive Analysis: Making plots, performing studies	Typically ~10	106	Tier 3: eg your laptop
TAG	Summary. Ex: p_T, η of 4 best e,γ,μ,τ,jet	Selection Events for analysis	1	108	Everywhere

Analysis Model Basics

Redundant Solutions

	Jets	Electrons	Missing Et
ESD All Calo Cells (not av... an...	Default Calibrate clusters to ...	Default Calibrate cells to EM ...	Default Build Missing Et from calibrated clusters + ... y in
A... All C... (Ca... +unc... samp... cells ...			...m ...ons.
clusters (available for analysis)	Build Jets From uncalibrated clusters, calibrate based on energy samplings	... cluster, recalibrate cells, re-calc shower shapes, re-calibrate electron	...m re-calibrated hard objects (eg jet, electron) + remaining contributions.

- Hypothetical Scenario:
  - 2 months from target conference, ATLAS discovers low level calorimeter calibration problem which hinders various measurements.
  - Not enough time to correct, reprocess, and redistribute data.

Analysis Model Basics

Redundant Solutions

	Jets	Electrons	Missing Et
ESD All Calo Cells (not available for analysis)	Default Calibrate clusters to hadronic scale based on cells	Default Calibrate cells to EM scale	Default Build Missing Et from calibrated clusters + out of cluster energy in cells. Save in components.
AOD All Clusters (Calibrated +uncalibrated samplings), All cells in lepton clusters (available for analysis)	Build jets from calibrated clusters, apply "out-of-cone"/Jet Alg Corrections	Choose electron cluster size, calibrate electrons based on samplings in clusters	Build Missing Et from individual contributions.
	Plan B Build Jets From uncalibrated clusters, calibrate based on energy samplings	Plan B Choose electron cluster, recalibrate cells, re-calc shower shapes, re-calibrate electron	Plan B Build Missing Et from re-calibrated hard objects (eg jet, electron) + remaining contributions.

Stage 1: DPD Building

Stage 1: DPD Building

Why?

- The AOD data-set is too big to store locally for interactive access (eg Tier 3/laptop) or to run on with one Core.
- AOD is general purpose, containing more information than necessary for any given analysis...

How?

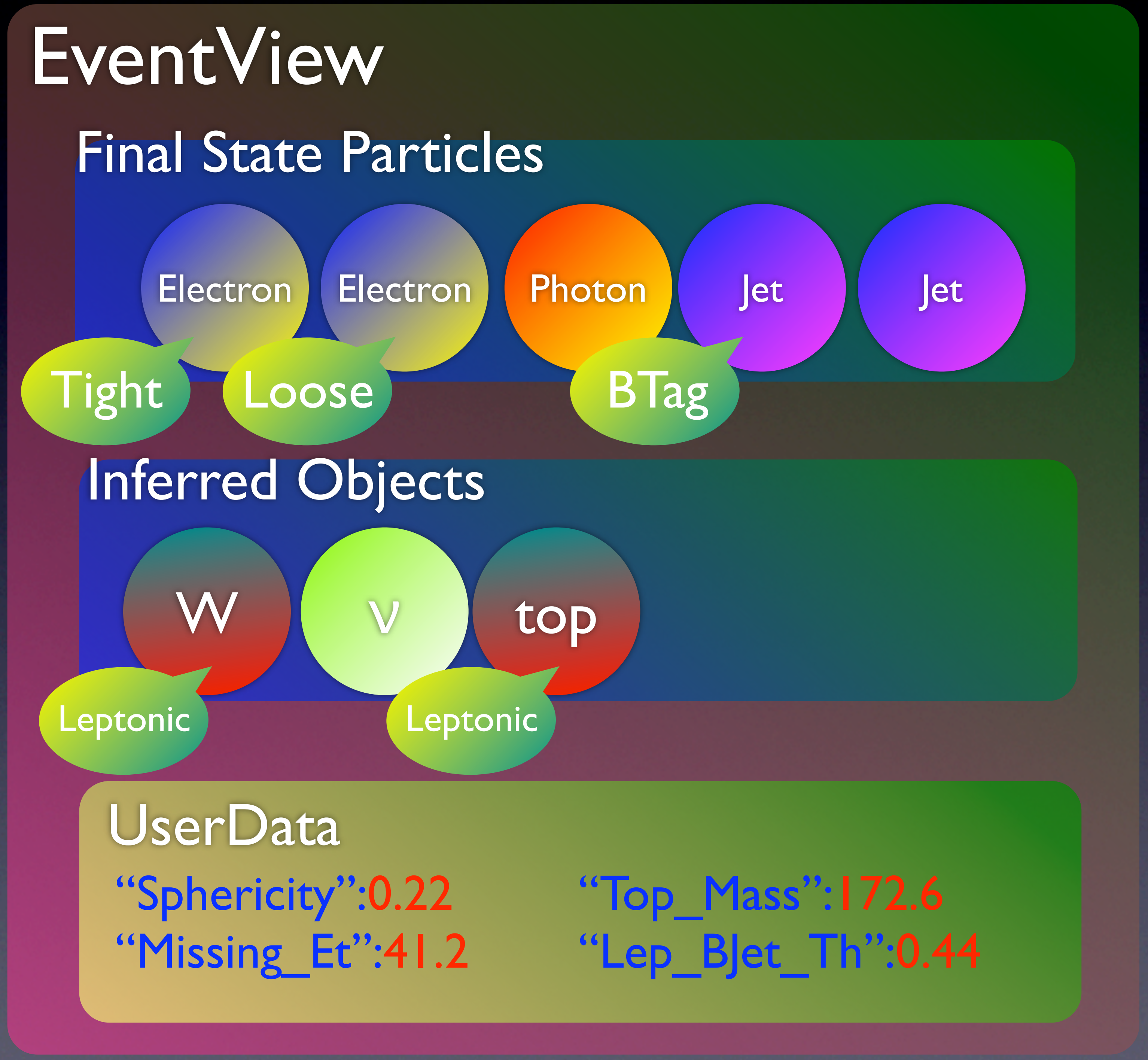
- So analysts should skim, thin, slim, & reduce the data to a manageable size for interactive analysis.
- 2 Aspects:
  - Basic framework support for such operations
    - *Skimming*: Easy... write out subset of what you read in. Gaudi Filters.
    - *Slimming*: write out subset of input containers. POOL output list.
    - *Thinning*: write out subset of object inside containers. Thinning service.
    - *Reduction*: User annotations. Add EventView/UserData. This hasn't been fully worked out.
  - Provide tools which encapsulate the physics decisions behind these operations... eg particle selection, overlap removal, combinatorics, observable calculation, ...

Collaborative Analysis

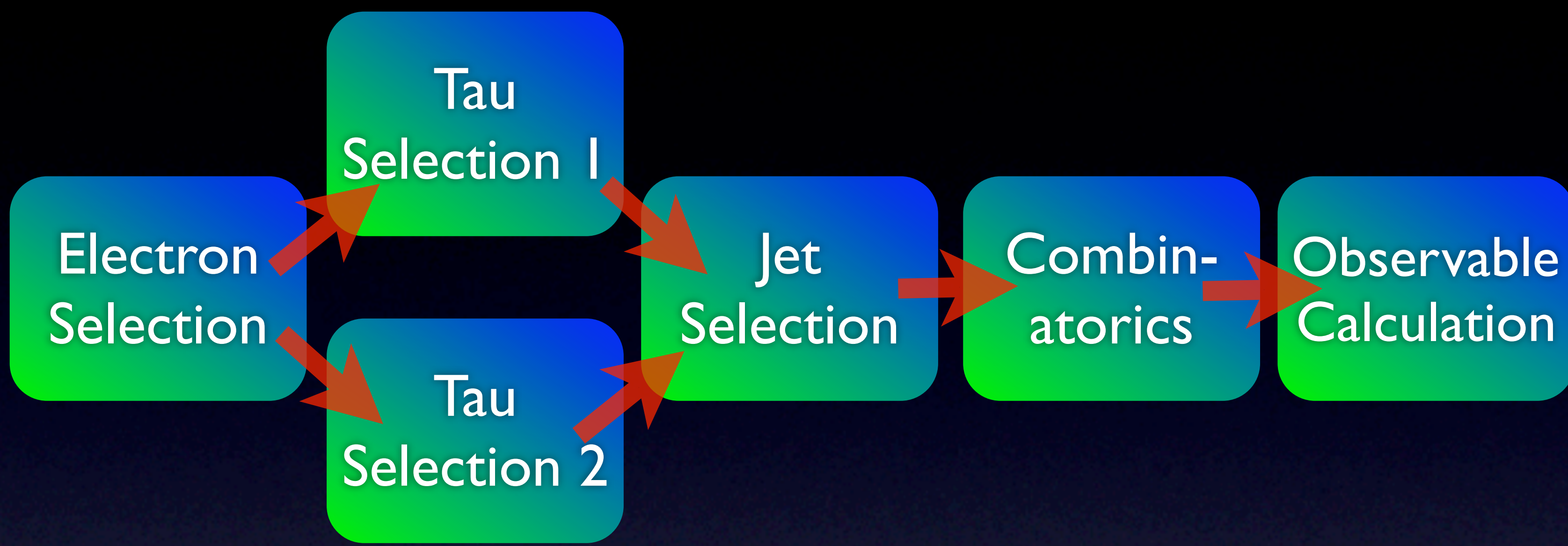
- ATLAS/CMS will have a dozen or more people contributing to a single measurement. And 100's within a physics working group.
- Experience from previous experiments and current analysis activities (CSC) show that:
  - Establishing common analysis procedures (eg Electron definition), validation, and consistency are critical for convergence to results.
  - Common Group DPDs ➔ efficient use of CPU, Disk, and man-power.
- Problem: how do you get 2000 physicists to
  - perform analysis in consistent ways
  - easily share & compare their work
- Same problem as reconstruction.
  - The reconstruction software is simultaneously developed by 100's of people over many years.
  - A common set of framework elements form the basic language of event processing.
  - Application is created at runtime.
- Solution: Apply the same framework design to analysis → EventView Framework...

The EventView

- EventView is a generic analysis data object.
- Holds the "state" of an analysis.
  - Objects in the AOD + Labels.
  - Objects created in the coarse of analysis + Labels.
  - UserData: Anything other data generated during analysis.
- Can be written/read from file and shared.
- Convention: each EventView holds *one* interpretation of an event... very natural book keeping tool.

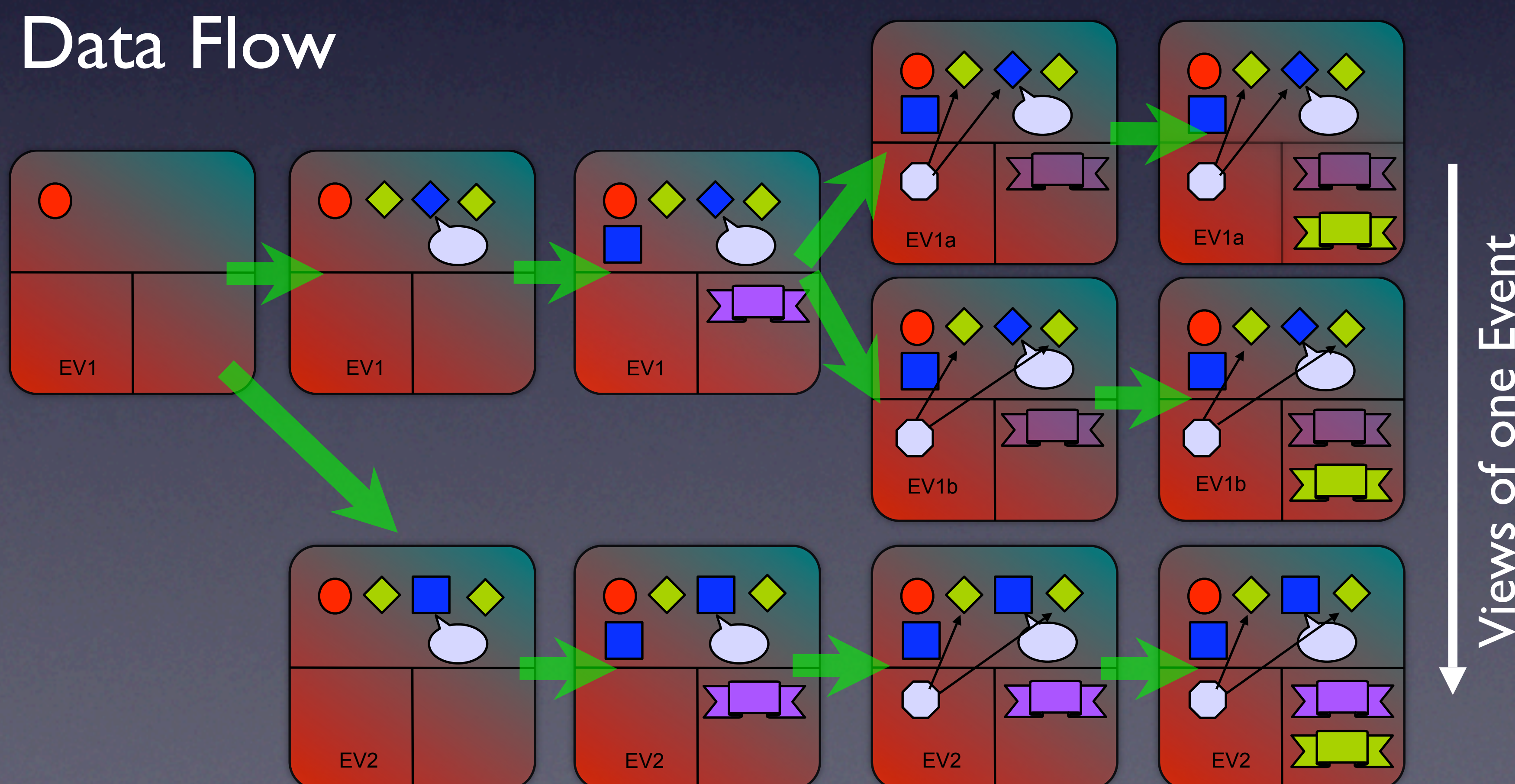


EventView Framework



Analysis Flow

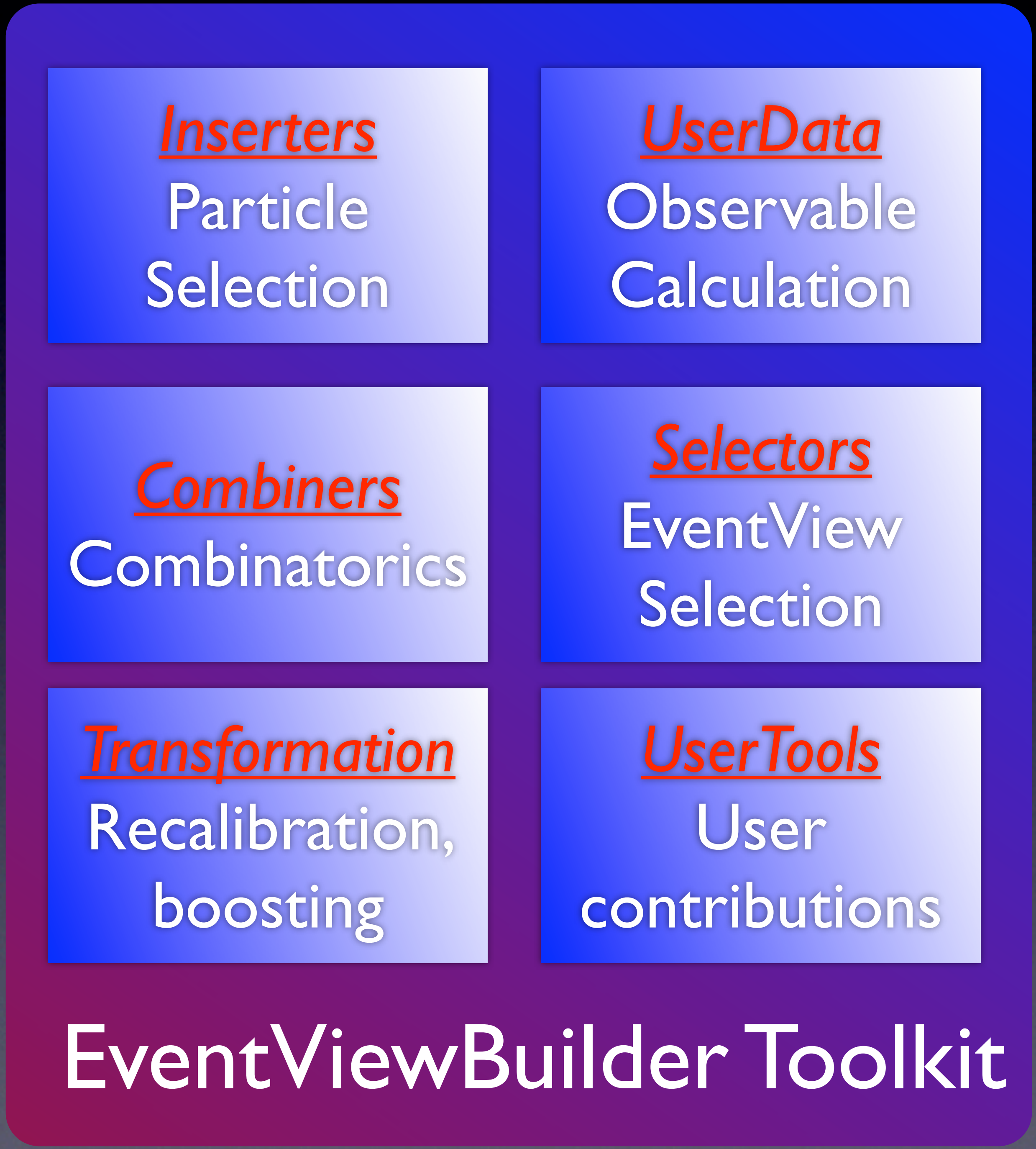
Data Flow



- Analysis is a series of EventView Tools executed in a particular order.
  - Modular Analysis
- Framework generates multiple Views of an event representing
  - Different analysis paths
  - Different combinatorics choices
  - Different input (eg: generator, full reconstruction, fast simulation)
- Everything consistent within one EventView ⇒ Framework handles bookkeeping.

EventView Toolkit

- 100's of generalized tools which can be configured to perform specific tasks.
- Tools instantiated/configured at runtime in python... users can perform complicated analyses w/o any C++.
- Provide the language for basic analysis concepts: "inserter", "looper", "associator", "calculator", "combiner", "transformer".
- Tools explicitly designed to be extended by users (when necessary).
  - Complicated Athena stuff in base classes.
  - Users only need to implement "the physics".
  - Users now routinely contribute new tools.



"View" Packages

- EventView Framework provides standardized mechanisms for building custom DPDs.
- EventView and software packages have a much faster development cycle than releases or patches! So the EV team provides/ distribute pacman caches.

- Analysis packages are mostly configurations of standard tools... minimal new C++.
- HighPtView: Generic Analysis package running in production ⇒ Standard:
  - Particle selections
  - Truth/Trigger Match
  - Output
  - ➡ Serves as benchmark/starting point for analyses
- 8 of the 9 ATLAS physics groups customizing use HighPtView or have custom packages SUSYView, TopView, ...
- And Performance packages: ElectornPhotonView, JetView, MuonView

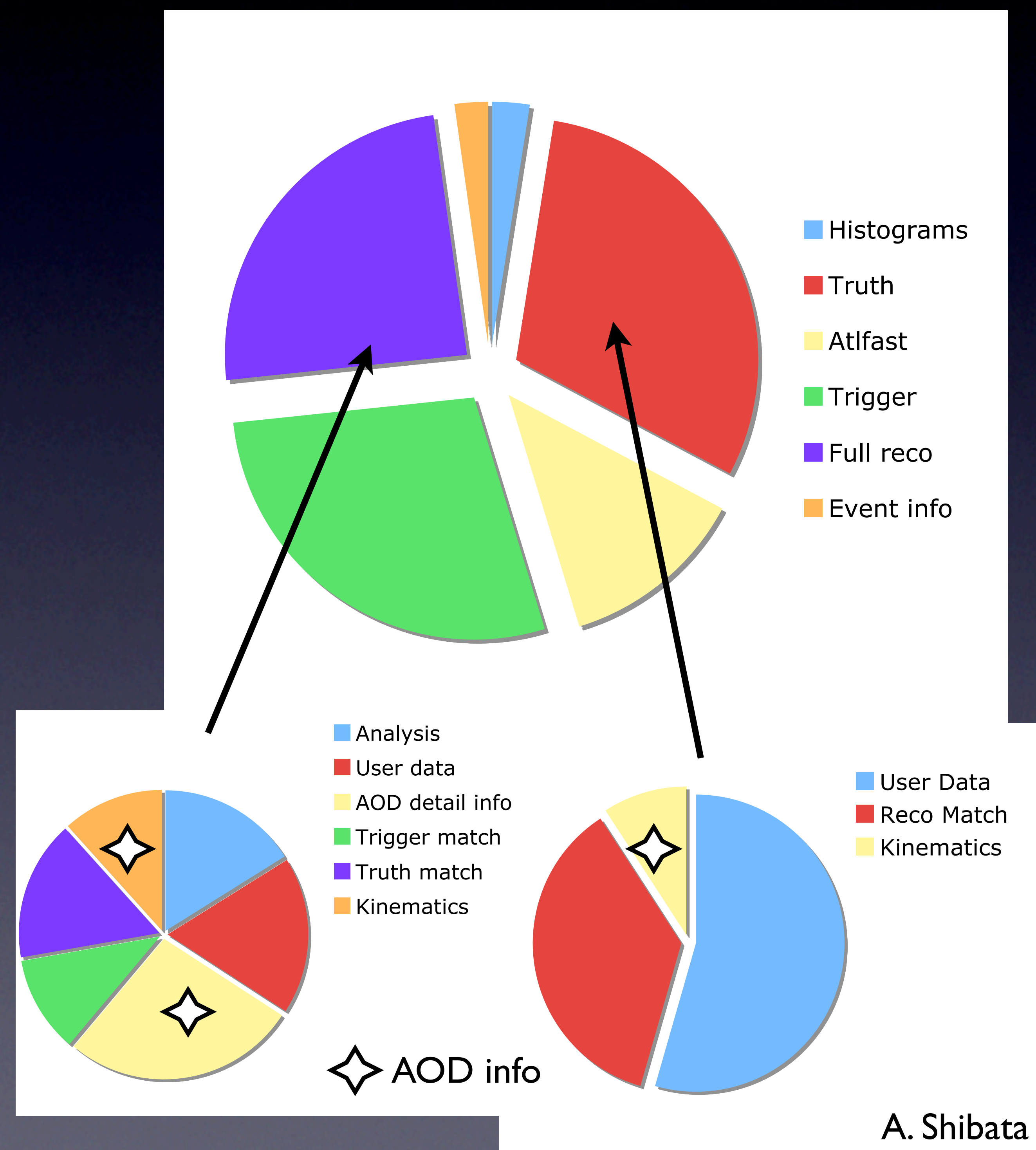
DPD Contents

- We are finding that there are two types of DPD, with one potentially derived from the other:
  - "Performance" DPDs: subset of information/events necessary for calibrations and performance studies. For early data or group wide DPD. Necessary to speed up iterations and/or use local resources.
  - "Analysis" DPDs: Tailored to specific analysis and user preferences.
- Two categories of information:
  - Information originally in the AOD (possibly re-reco'ed, re-calibrated, or corrected):
    - Ex: Tight/Medium Electrons, their tracks and clusters, and every track within cone 0.1 around them and the closest topo-cluster.
    - All true Electrons which come from a t->Wb->e nu jet chain.
  - Information not in AOD, often referred to as UserData: (Example)
    - "Labels": The fact that the electron is Tight or Medium, it was used in W reco... Flags that the true electron was reco'ed as Jet or Tau... that the true electron came from a W...
    - The association between the Electron and the tracks/clusters around it.
    - The association between the true, reco, trigger Electron.
    - Composites Objects (or just their kinematics)
    - Event Shape Variables etc...

"UserData"

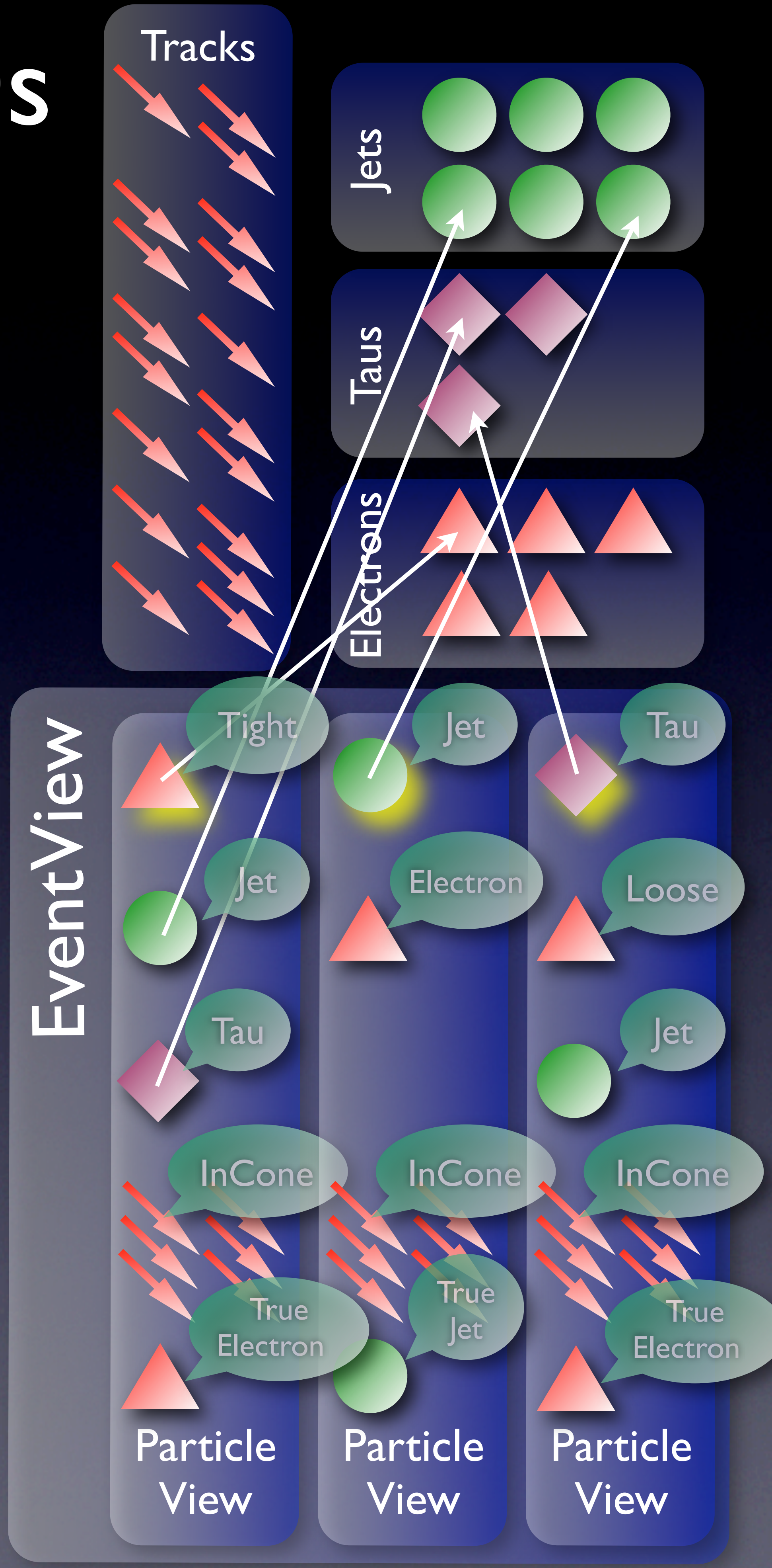
- Many of these quantities are calculable on the DPD in ROOT, but
  - often one double (per object?) is all you need in the rest of the analysis, so users can reduce DPD size by not saving the inputs to the calculation.
  - users can save a lot of ROOT processing time by caching the result in the DPD.
  - often very convenient to have these quantities pre-calculated.
    - eg: With well made DPDs you can make efficiency, resolution, scale plots for any reco or trigger object with single-line ROOT commands.

Top Group DPDs



EventView Generated DPDs

- In the EventView framework analysis is separated from the format of the DPD.
  - So EV can create flat (simple TTree) or POOL-based DPDs without any changes to the analysis.
  - EV presents a simple interface to users for defining in Athena what sub-set of AOD to be retained in DPD.
  - UserData: The EventView is stored in the DPD.
    - Annotates the AOD.
    - Provides a common format for the UserData part of the DPD.
- DPD can be read back into Athena and analysis continued with EventView or analyzed directly in ROOT.
- Note: Flat Ntuples are always faster than complicated formats... users are still likely to generate very simple flat ntuples at some point.

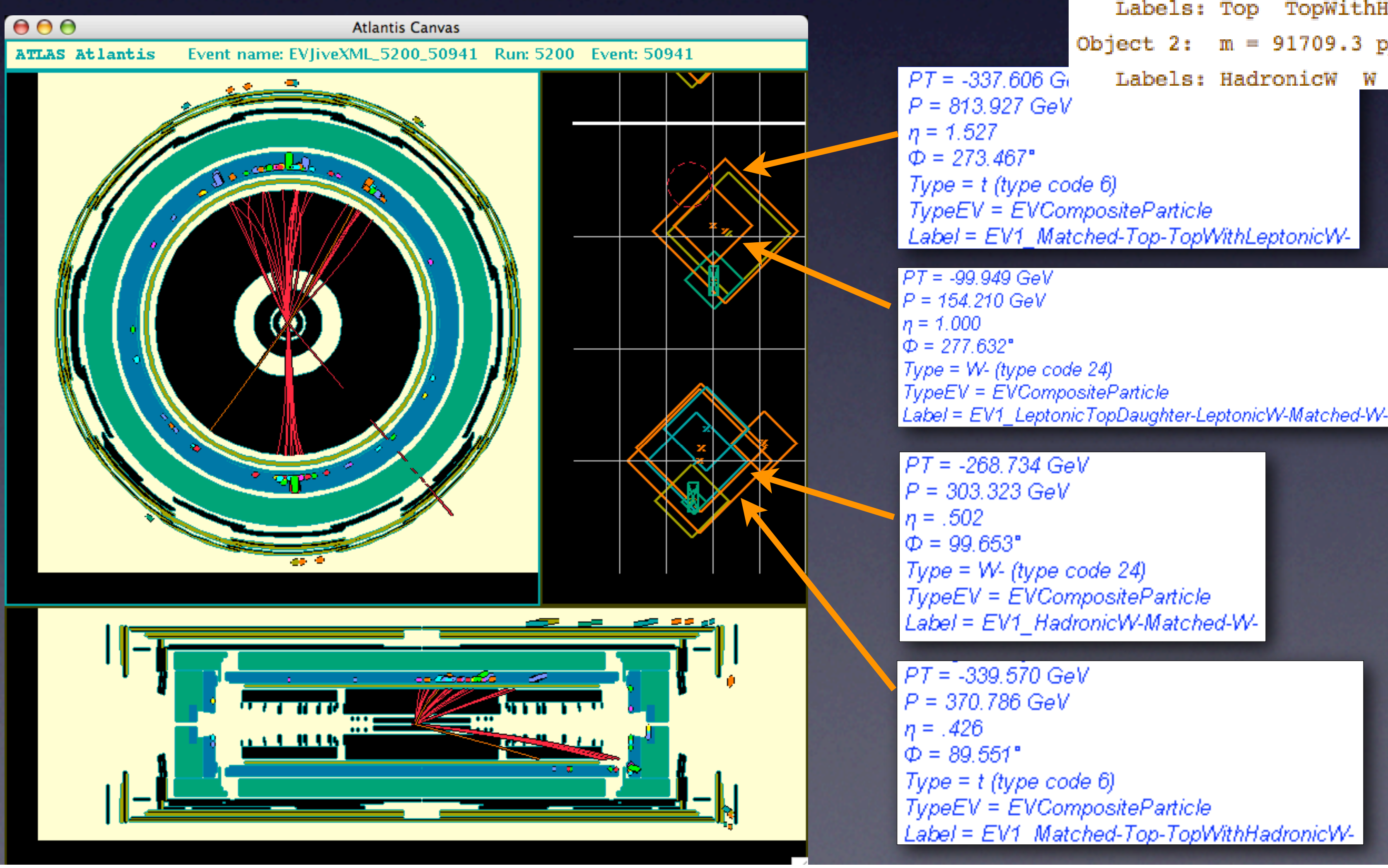


EV in the DPD

- EV stores all of the results of any EV analysis in a format that is common to all analyses... regardless of what was done in the analysis.

ScreenDump from Athena

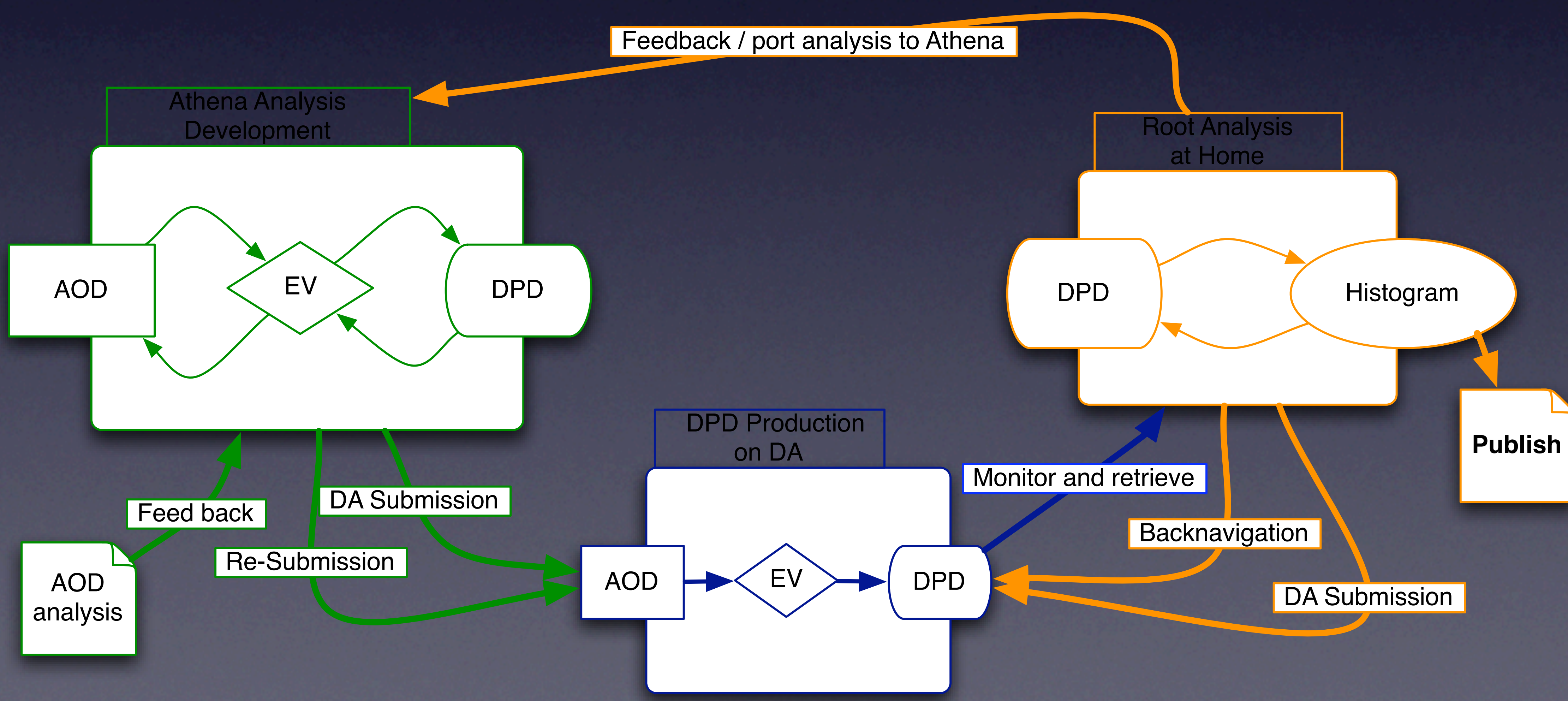
```
---------------- Final State Objects ------
Object 0:  p_T = 55461.3 phi = -2.18794 eta = 1.04014 type = Analysis::Muon
   Labels: Lepton  Muid  Muon  Tight
Object 1:  p_T = 10807.9 phi = 0.978227 eta = 0.17026 type = Analysis::Muon
   Labels: Lepton  Loose  Muid  Muon  Tight
Object 2:  p_T = 238032 phi = -1.54079 eta = 1.69019 type = ParticleJet
   Labels: CentralJet  Cone4  HardJet  ParticleJet
Object 3:  p_T = 160690 phi = 1.54603 eta = 0.308174 type = ParticleJet
   Labels: CentralJet  Cone4  HadronicTopDaughter  HadronicWDaughter  HardJet  ParticleJet
Object 4:  p_T = 115243 phi = 2.01035 eta = 0.720166 type = ParticleJet
   Labels: CentralJet  Cone4  HadronicTopDaughter  HadronicWDaughter  HardJet  ParticleJet
Object 5:  p_T = 88584.7 phi = 1.00186 eta = 0.0929353 type = ParticleJet
   Labels: BTagged  CentralJet  Cone4  HadronicTopDaughter  HardJet  ParticleJet
---------------- Inferred Objects -------------------
Object 0:  m = 814805 p_T = 67360 phi = 1.81452 eta = 3.22118 type = CompositeParticle
   Labels: AllObjVectSum
Object 1:  m = 168641 p_T = 339570 phi = 1.56295 eta = 0.425561 type = CompositeParticle
   Labels: Top  TopWithHadronicW
Object 2:  m = 91709.3 p_T = 268734 phi = 1.73927 eta = 0.502074 type = CompositeParticle
   Labels: HadronicW  W
```



- You can open someone else's POOL-based DPD, print the EVs and look at them in a the Event Display.
- You can read in the EVs in Athena and continue where the previous step left off.

Analysis Work Flow

- The current vision:
  - DPD is produced in multiple steps. eg: Physics Group DPDs ➔ Analysis Level DPDs.
  - A group defines the content of their DPD
  - A subset of experts implements the DPD making job (likely using EventView)
  - A subset of the group or central production generates the group DPD
  - The whole group analyzes the DPDs on local resources... provide feedback for next iteration



Central DPD Production

- Unmanaged DPD Production can be very taxing on computing infrastructure
  - IO intensive.
  - Peaks prior to conferences

Frank Winklmeier (Former BaBar Skim Coordinator)

On time
Can work while on train
Everybody has the same seat
Available resources managed centrally

Unpredictable delays
You are not productive while driving
Who starts first (maybe) arrives first
Non optimal use of resources

- But difficult to combine 100's of user coded DPD-making tasks in a reliable manner without infrastructure.
- Another argument for using the EventView framework... multiple EventView of Events built into framework.

Stage 2: Interactive Analysis

Stage 2: Interactive Analysis

- Format of the DPD
  - Use athena convertors to read EDM objects into ROOT... so the DPD format is the same as AOD/ESD.
  - Allow saving additional non-standard info... eg EventView/UserData.
- Dataset management
  - N datasets (eg data, signal MC, bkg1 MC, bkg2 MC, ...)
  - M_i files in each... different cross-section, preselection (trigger?) efficiency, ...
- Interactive Plotting (eg TTree::Draw).
  - Limited. Usefulness depends on DPD format. Ex: With EventView DPD you can make efficiency, resolution, scale plots because results of matching is stored in DPD.
  - But inefficient for making lots of plots from the same dataset because each plot requires its own loop over data.
- Batch Analysis (eg TTree::MakeClass ➔ Loop())
  - As sophisticated as your input DPD allows. Compile for speed.
  - Simultaneously generate multiple histograms, ntuples, etc...
- Finalizing plots, making tables, etc

Interactive Athena

- For several years now, users can start Athena and get a python prompt.
- Advantages:
  - TTree::Draw/Scan like plotting directly from StoreGate... uses PyROOT for plotting.
    - Don't need to write another data format, just to make a simple plot from RDO, ESD, or AOD. Great for validation.
  - Write/combine python (or C++) algorithms.
  - Access to full Athena functionality (eg geometry, conditions, analysis tools...)
- Disadvantages:
  - Algorithms written in python are slower than C++ (recent studies indicate CINT:PYTHON:Compiled C++ = 8:2:1 in speed)
  - Only works on platforms supported by Athena (ie linux).
  - Requires software installation... at least 1.5 GB... or larger if you want more Athena functionality.
- Not widely used because
  - Until very recently (release 13), AOD access was prohibitively slow.
  - Users reluctant to learn something new... they prefer ROOT.

AthenaROOTAccess

- Uses Athena Transient/Persistent convertors to read POOL data into ROOT
  - Builds a Transient TTree with the transient versions of AOD objects (so exact same speed/ initialization as Athena/interactive athena).
- Advantages:
  - Don't need to run an athena job to see a quantity stored in the EDM... simplifies validation of data...
  - DPD has the AOD structure... the same EDM objects appear in ROOT and athena analysis...simpler to migrate code between the two.
  - Users like being able to read data in familiar ROOT environment without knowing anything about ATLAS's software framework.... a great entry point.
  - Can use ROOT's parallel processing facilities (PROOF).
- Limitations:
  - Only works on athena supported platforms (ie linux)
  - Requires installation ~ 1GB of athena libraries
  - No access to athena services
    - No conditions/geometry: cannot read data which needs these services in the convertors. Ex: trigger decision, calorimeter cells
    - No ToolSvc, PropertySvc, StoreGate: cannot use Athena algorithms, Tools, etc.

Interactive Analysis Frameworks

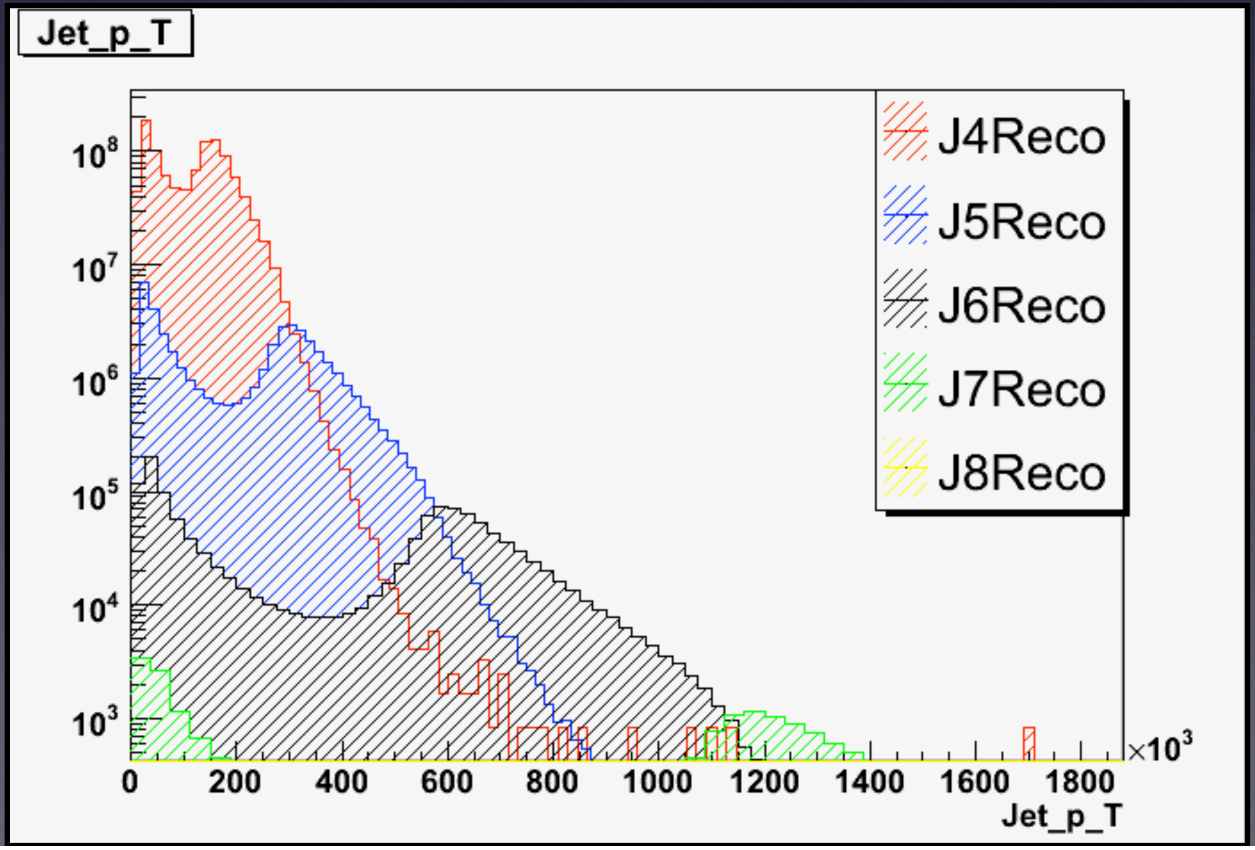
- ROOT/PyRoot frameworks emerge as analyses become more sophisticated than what is manageable in a macro.... Atlas is now considering requirements of a common framework for AthenaROOTAccess.

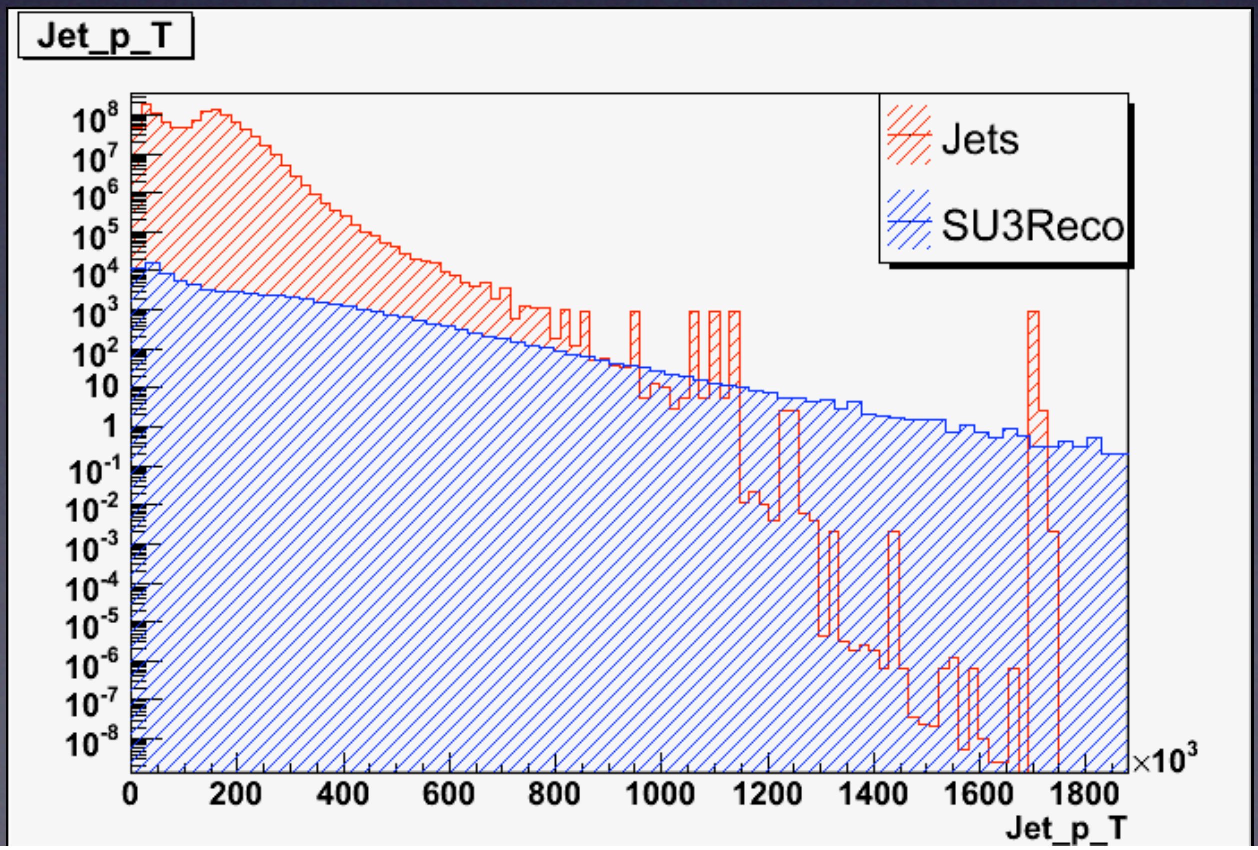
SPyRoot

```
import SampleHandler
Data=SampleHandler.SampleGroup()
BaseDirectory="/data/MyData/"
SampleNames= [ ["J4",3.08E+005], ["J5",12470], ["J6",360.4], ["J7",5.707], ["J8",
0.24], ["SU3",19.3] ]
for S in SampleNames:
        Data.AddDirectory(S[0]+"Reco",BaseDirectory+S[0],"EV0","Reco",S[1])
```

- Data.Compare(["J4Reco","J5Reco","J6Reco","J7Reco","J8Reco"],"Jet_p_T")

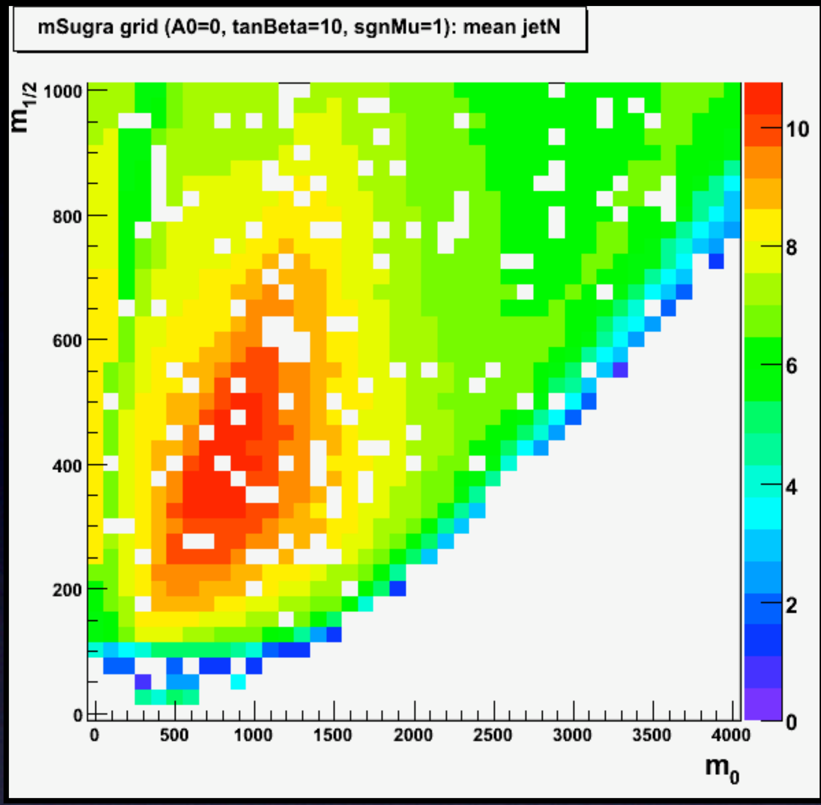
- Data.AddCombinedSample('Jets',Data,['J4Reco','J5Reco','J6Reco','J7Reco','J8Reco'])
- Data.Compare(['Jets','SU3Reco'],'Jet_p_T')





Batch Analysis

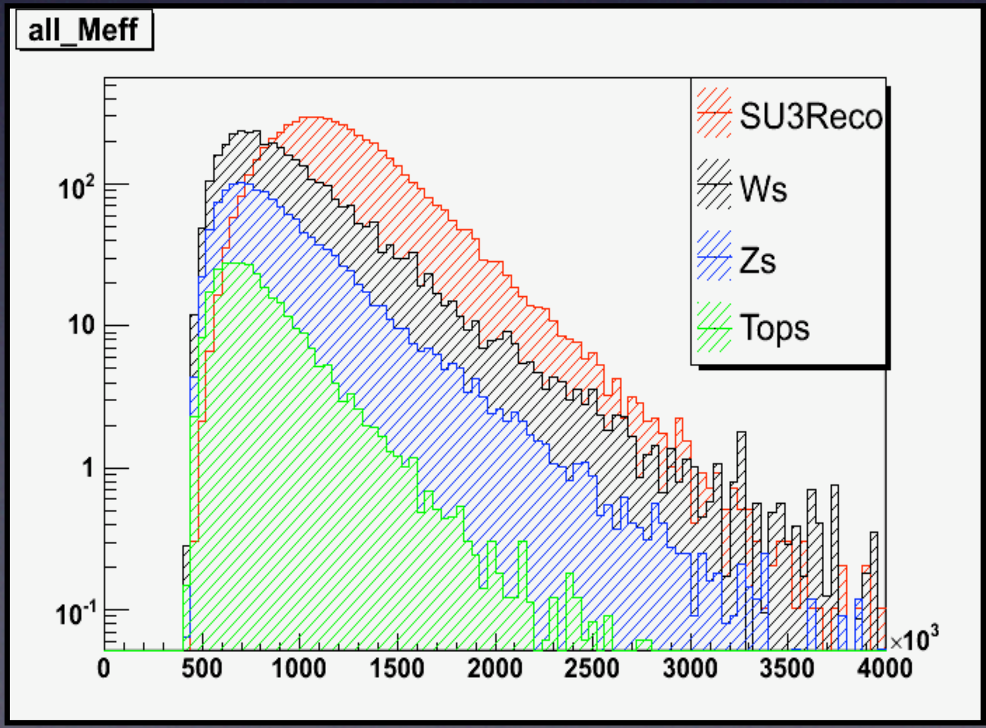
```
TheAnalysis=TTreeAlgorithmLooper("TestAnalysis")

TheAnalysis.AddAlgorithm(VarHistAlgorithm("JetN_hist","JetN_hist","JetN_hist","T.jetN",
20,0,20,["jetN"]))

TheAnalysis.AddAlgorithm(SimpleVarCutAlgorithm("4JetsCut","T.jetN>3",["jetN"]))

TheAnalysis.AddAlgorithm(VarHistAlgorithm("MET_hist","MET_hist","MET_hist","T.MissingEt
",100,0,1000000,["MissingEt"]))

TheAnalysis.AddAlgorithm(SimpleVarCutAlgorithm("METCut","T.MissingEt>100000.",
["MissingEt"]))

TheAnalysis.AddAlgorithm(TransverseMassAlg("M_T"))

TheAnalysis.AddAlgorithm(WriterAlgorithm(["M_T","Jet_N", ...])
```




```
import RunHandler

RH=RunHandler.RunHandler(["SU4Reco","J1Reco"], TheAnalysis, "myRH")
RH.Loop()

import pickleResults
pickleResults.save(RH.Results, "myAnalysis_")
```

```
res.GetCutTable(Samples=["SU3Reco"], Lumi=1000.0)
```

```
Sample: SU3Reco
Cut: 4JetsCut    -> Eff: 0.69 err: 0.001
Cut: METCut      -> Eff: 0.88 err: 0.0009
Cut: JetCutAlgo -> Eff: 0.47 err: 0.001
Final Cut Effc: 0.29 err: 0.001
exp. evts. (after cuts): 5654.1 err: 5.9
```



Stage 3: Statistical Analysis

- Several modern tools (mostly from BaBar) are being adopted by LHC
- Multivariate discriminant framework: TMVA.
  - Easily build and compare various discriminants... eg Fisher, Neural Network, boosted decision tree, ...
- General Statistics Framework (for LHC).
  - RooStats... based on RooFit... under development now.
  - Build models of data ⇒ fits, "toy" Monte Carlos, calculate significance... share models/data.
  - Provide standard (and correct) calculation of significance and handling of (systematic) errors.
  - Compare different techniques/calculations.
- Such activity is very CPU intensive, with little output
  - One fit can take O(12 hours) on one core... parallelization possible within machine.
  - Typically need O(1000) toy experiments (Toy MC/fit) to validate fits and calculate significance. This can be done simultaneous on multiple cores.
  - This means you need 500 cores to do this step in 1 day... resources for this kind of activity not allocated.

Tier 3s?

- Tier 2s are only accessible via GRID middleware... no interactive login.
- Users need a place to login, develop code, test, submit large scale batch jobs, and analyze the results of these jobs (w/ non-framework software).
- CERN, BNL, etc provide interactive access... but these can quickly be over-subscribed.
- The role of CERN CAF is not clear. This is likely going to be limited to calibration activity on ESD. Regardless, there is no way CERN CAF can support a significant fraction of all analysis activity.
- Currently, institutions with Tier 2s give their local users login access... this is unfair to everyone else because they are providing privileged access to global resources.
- National analysis facilities will provide interactive access...
  - it is not clear that everyone's needs will be met
  - analysis activity is very taxing... such centers may have difficulty supporting 100's of users.
- ATLAS is just starting to explore the role and size of Tier 3 (local computing resources at Universities).
- My estimate for Tier 3: 25 CPUs per analyzer, 40 TB of disk per analysis, fast disk/network.

Summary

- ATLAS Analysis Model focuses on ensuring framework, event data model, analysis tools, and persistency technologies allow analyzers to:
  - Re-reconstruct and re-calibrate objects on AOD while still remaining within the space budget.
    - Unify reconstruction and analysis objects.
    - Carefully tune AOD contents.
  - Build custom Derived Physics Data.
    - Identify basic operations: skimming, thinning, slimming, reducing
    - Provide framework support for these operations.
    - Provide a high-level framework for collaborative development of analysis packages based on common tools.
  - Efficiently analyze DPDs on local resources.
    - Make framework objects directly readable in ROOT.
    - Trying to understand the role of Tier3s and the requires.

Other Tier 1/2/3 Activity

- You are likely to want to do some fast simulation production. (I included this in tier 3 disk estimate)
  - Some organized group production... but much analysis specific production.
  - ~2500 events/hour/core. Much smaller output... + Step 1/2 analyses.
  - We will produce > 5x more FastSim events than FullSim.
- Maximum-likelihood fitting/toy experiments. (CPU intensive, minimal input/output)
  - One fit can take O(12 hours) on one core... parallelization possible within machine.
  - Typically need O(1000) toy experiments (Toy MC/fit) to validate fits and calculate significance. This can be done simultaneous on multiple cores.
  - So this is 20 days on your Tier 3... you really want this to be 1 day!
  - These are not athena jobs... so they are not supported in PANDA (of course they can be). So Tier 2's don't really support this now... I don't think it will be difficult to support because there is no disk requirement.
  - Most likely the batch systems at Tier 1 or CERN will need to satisfy this need.

Tier 3 CPU

- Note that your Tier 3 is the most likely place for your daily interaction with ATLAS data.
  - Every day you will work on your Tier 3... (develop, analyze, etc...)
  - But you will likely use Tier 2 CPU periodically... (run over lots of data) Tier 2s provide 12 cores/person for analysis at any given time... aggregate cores by working cooperatively (and working asynchronously).
- But Tier 3's are personal.
- Seems "logical" that a Tier 3 provides more CPU per person than at Tier 2... otherwise users might as well use tier 2.
- In the table I assumed 25 cores per simultaneously active person... less/more means you wait longer/less. This is 3 8-core, $4000K machines.
- This means over night, you can just barely make plots (step 2) on 1 year's worth of data. (With PROOF, for example).

Tier 3 Infrastructure

- The activity on Tier 3's will mostly likely be IO limited ➔ Good storage infrastructure and network.
- Tier 3 will likely include machines on your desk (including your laptop)...
  - So your SEs should be accessible on the physics department network (xrootd).
  - Your 8 core desktop will want 80 MB/s (gigabit network all the way to your desk)
- 25 Cores will be simultaneously reading data: disk infrastructure should be able to handle > 250 MB/s... again xrootd would be useful.
- Users will need to run parallel jobs:
  - Clearly need a batch system.
  - PROOF for parallel interactive analysis.
- Simulation must be done using production system... need GRID infrastructure.

Tier 3 Disk Space

- At tier 3: you will likely prototype Step 0 and 1 analyses, and run full Step 3.
- Assuming that you will at max wait for 1 week and 25 CPUs.
  - Step 0: Total (AOD): 100TB. Input: 1.5 TB, Output: 0.75 TB
  - Step 1: Total (DPD= 50% AOD): 50 TB. Input: 8.75 TB, Output: 1.75 TB.
  - Step 2: Total (DPD= 10% AOD): 10 TB. Input: 10 TB (1 processing = over-night)
    - Unlikely that users will need every event. Let's assume they need ~50% of events. (But can be significantly less)
- Likely need 2 versions of Step 2 DPDs!
- Addition data: Full Sim: 20% (but is 20% bigger). FastSim 100% of 10% AOD (1 version).
- So to take full advantage of your Tier 3 (ie steps 0-2): ~ 36 TB
- Just Step 2: ~26 TB
- Note: total doesn't really linearly scale with analysis... AOD/50% AOD may be used for > 1 analysis.
- Looks like ~40 TB/year is the reasonable scale. (Later years: more signal, but better selections).