Markus Elsing

Introduction to ATLAS (Offline) Software

Introduction

- will give an overview of ATLAS offline software
  - ➡ will not say much about online software (Trigger, DAQ and DCS)
- offline software domain
  - ➡ applications:
    - generation, simulation, digitization, reconstruction, physics analysis tools, ...
  - ➡ software infrastructure:
    - software repository, framework, releases building and validation ...
    - detector description, event data model, conditions database ...
  - ➡ not to forget: documentation (!)
- use following 30 min. to give a first overview
  - ➡ more detailed presentations and practical sessions later this week

ATHENA - the Software Framework

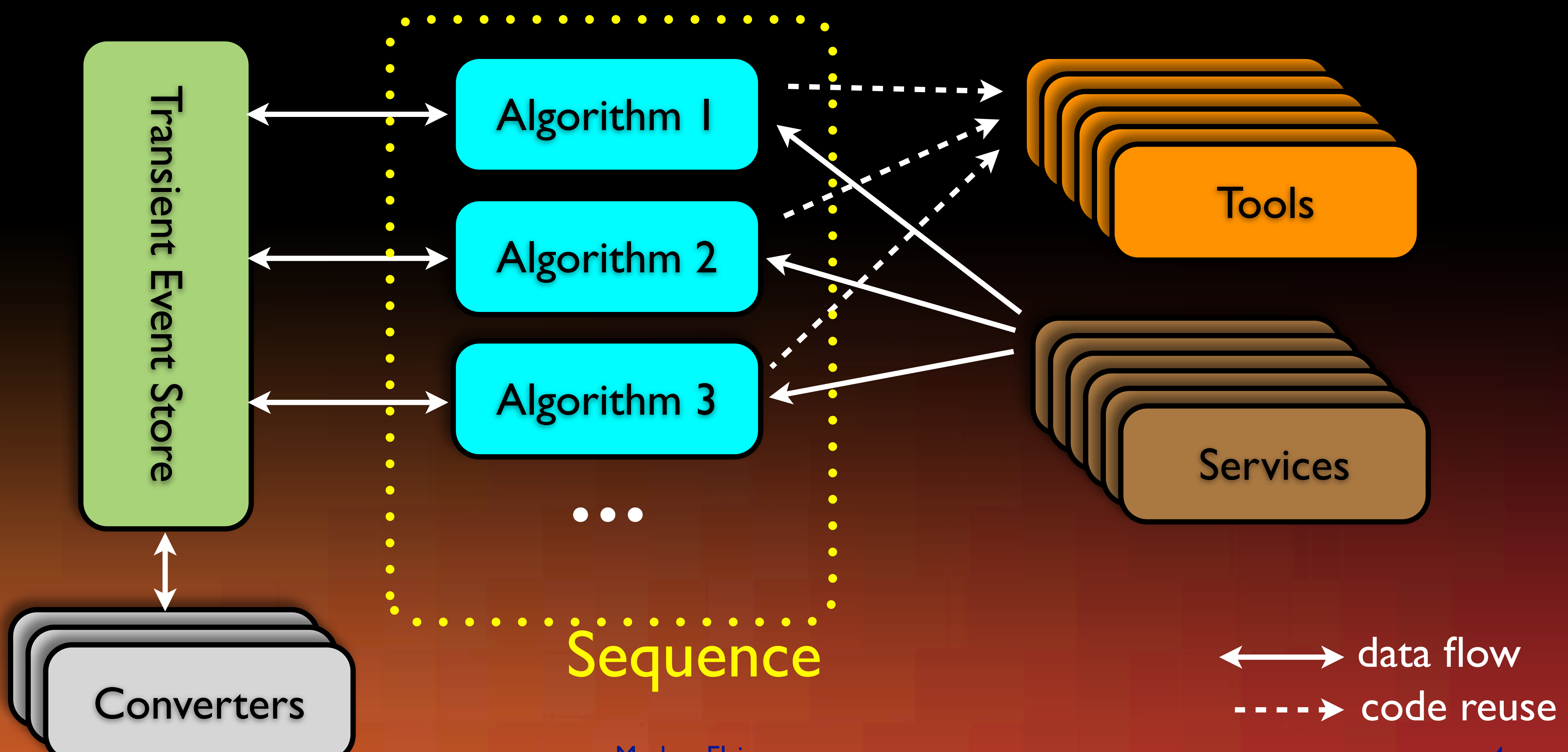
- what is a software framework ?
  - ➡ skeleton for all application into which developers plug in their software
  - ➡ predefines an high level "architecture" or software organization
  - ➡ provides functionality common to several applications
  - ➡ controls the configuration, loading and execution of the software

- **ATHENA** is based on GAUDI (originally from LHCb)
  - ➡ used for Simulation, Reconstruction and even for the High Level Trigger
  - ➡ used as well for data analysis (but less popular)
  - ➡ software written in **C++** (some FORTRAN90 in muon reconstruction)
  - ➡ scripting and configuration in **PYTHON**

- another Framework in use in ATLAS is **ROOT**
  - ➡ mostly used for (ntuple) analysis and data visualization
  - ➡ applications writting in C++ (CINT) or PYTHON (pyroot)
  - ➡ ROOT is as well a collection of software libraries
    - functionality used in ATHENA, like "ROOT I/O" for persistency

... in Practice ?

- ATHENA enforces a strict software architecture
  - ➡ structures the development process (!)
  - ➡ event data is separated from algorithmic code (“Object Based Model”)
- high level design:



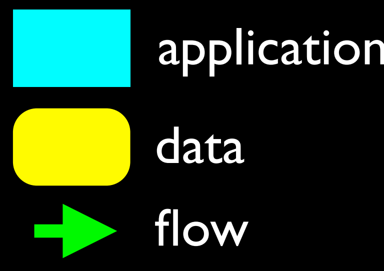
Main ATHENA Components

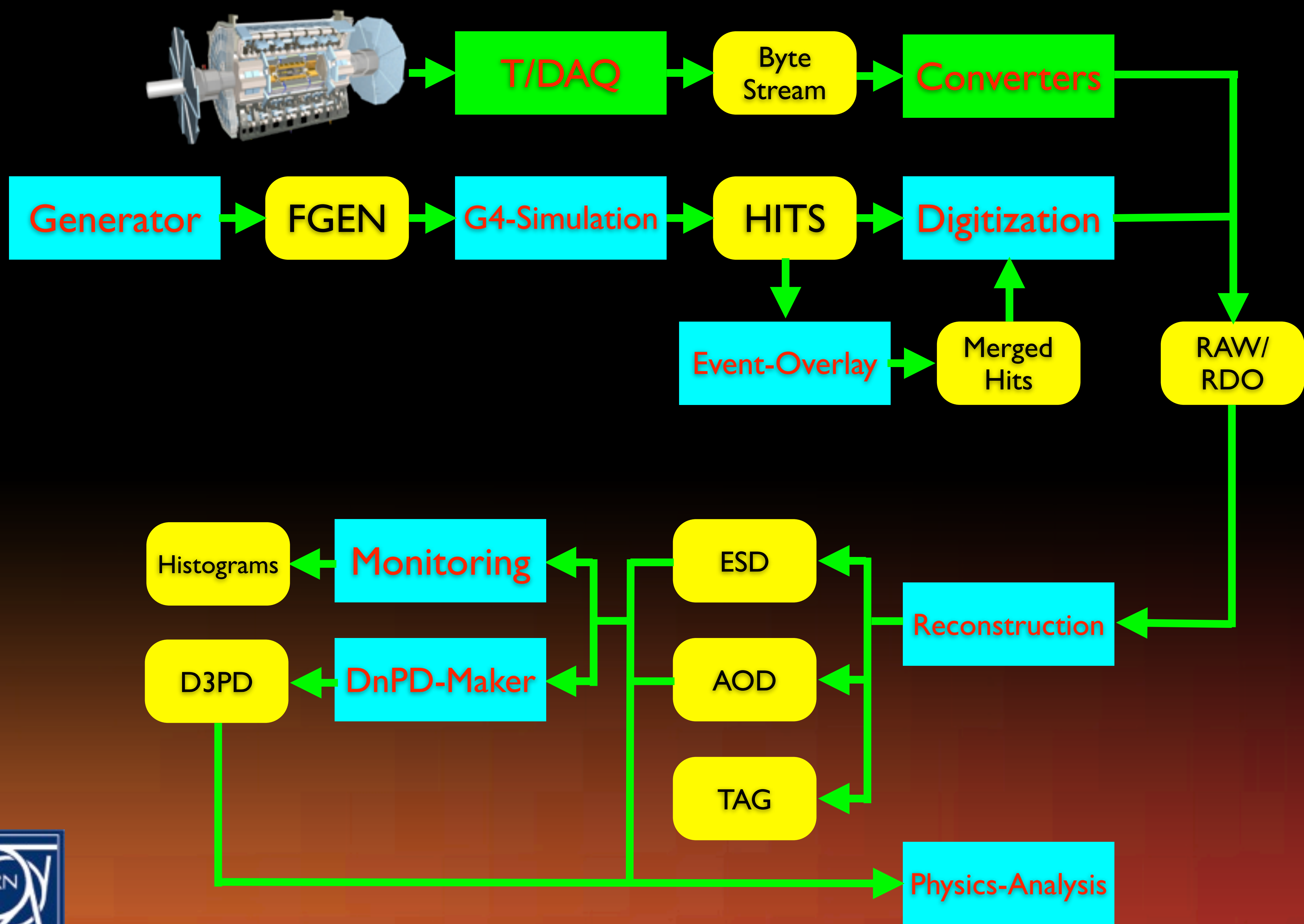
- Algorithm
  - ➡ a piece of code that **"does something"**
  - ➡ inherits methods from base class, especially
    - initialize() ~ runs once at beginning to prepare processing
    - execute() ~ runs for each event
    - finalize() ~ runs once after last event at end of processing
- Sequence
  - ➡ a ordered list of Algorithms invoked by framework for each event
- Tool
  - ➡ a piece of code shared between algorithms
  - ➡ can be used many times in an execute() of an algorithm
- Service
  - ➡ provide an interface to meta data, message printing, histogram drawing...
- Transient Event Store
  - ➡ **StoreGate** - holds all event related data objects
- Converter
  - ➡ code that implements file input/output of data objects

Assembling an ATHENA Application

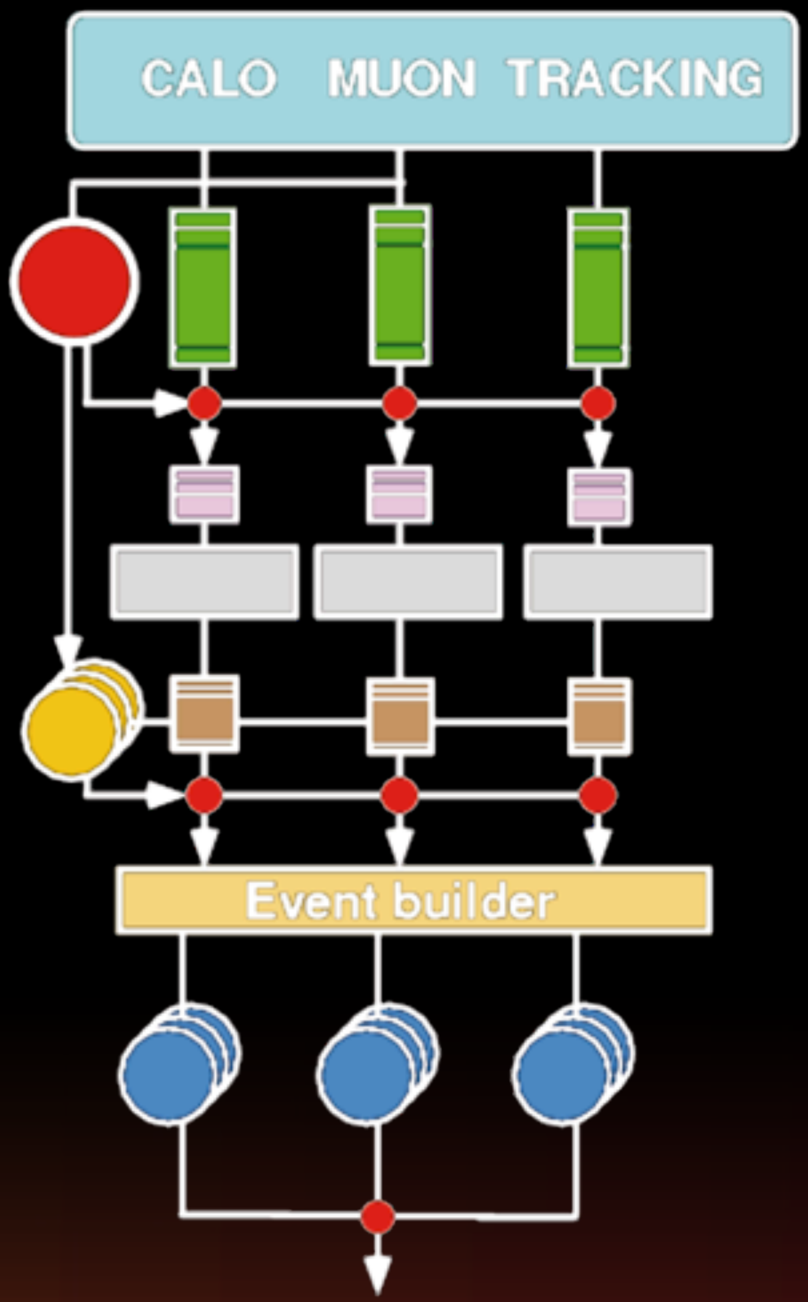
- applications are configured using **JobOptions**
  - ➡ written in PYTHON
- ATHENA framework auto-generates PYTHON classes
  - ➡ so called "Configurables"
  - ➡ available for all Algorithms, Tools and Services in a software release
  - ➡ expose all their configuration parameters and their defaults
    - e.g. , a cut value for a p_T of a jet
- hand written JobOption files to assemble application
  - ➡ import PYTHON classes for Algorithms, Tools and Services you need
  - ➡ overwrite configuration parameters if needed for your use-case
- JobOptions are structured hierarchically, e.g.
  - ➡ JobOptions per domain to configure e.g. Muon Spectrometer reconstruction
  - ➡ top level JobOption integrate everything into full reconstruction application
    - called **RecExCommon** ~ the top level JobOptions for reconstruction

ATLAS Data Processing Chain





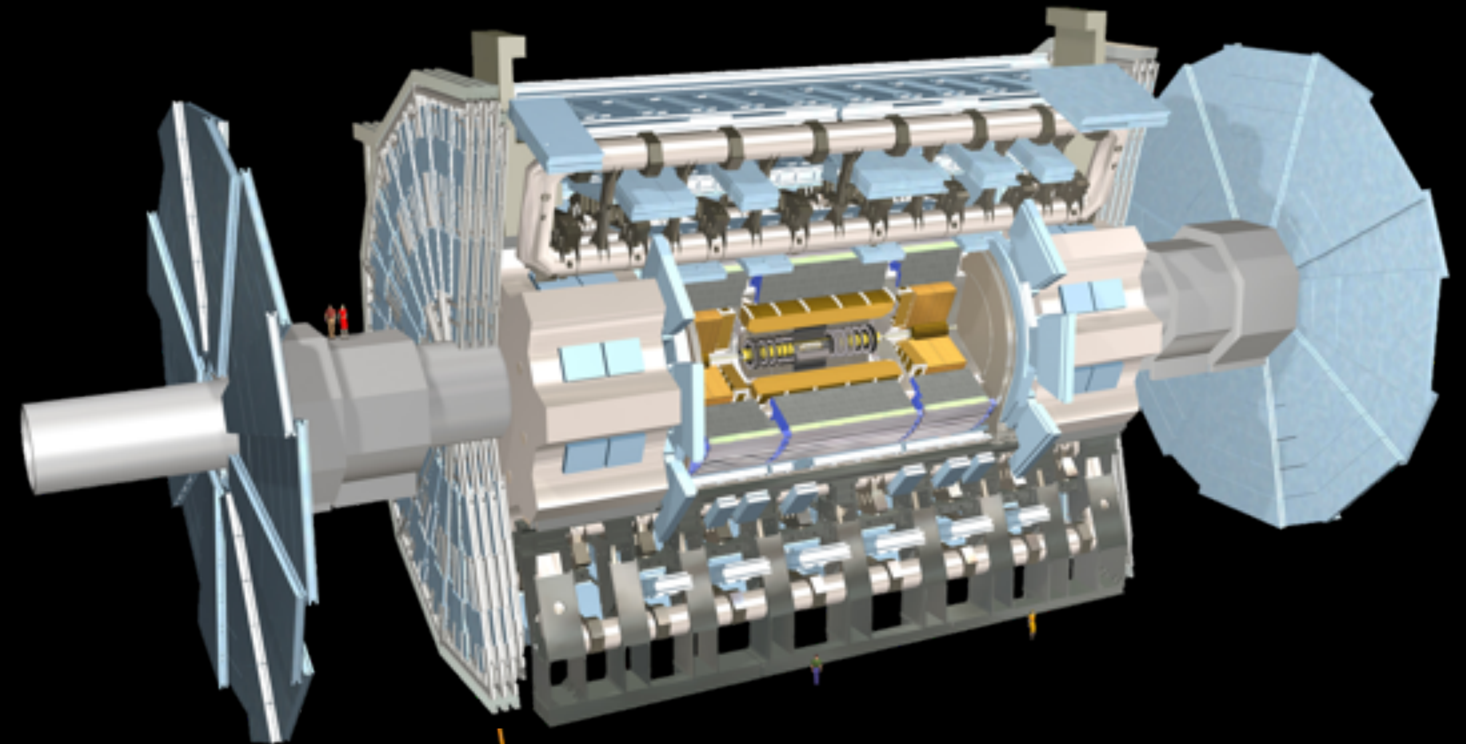
Real Data taken by ATLAS

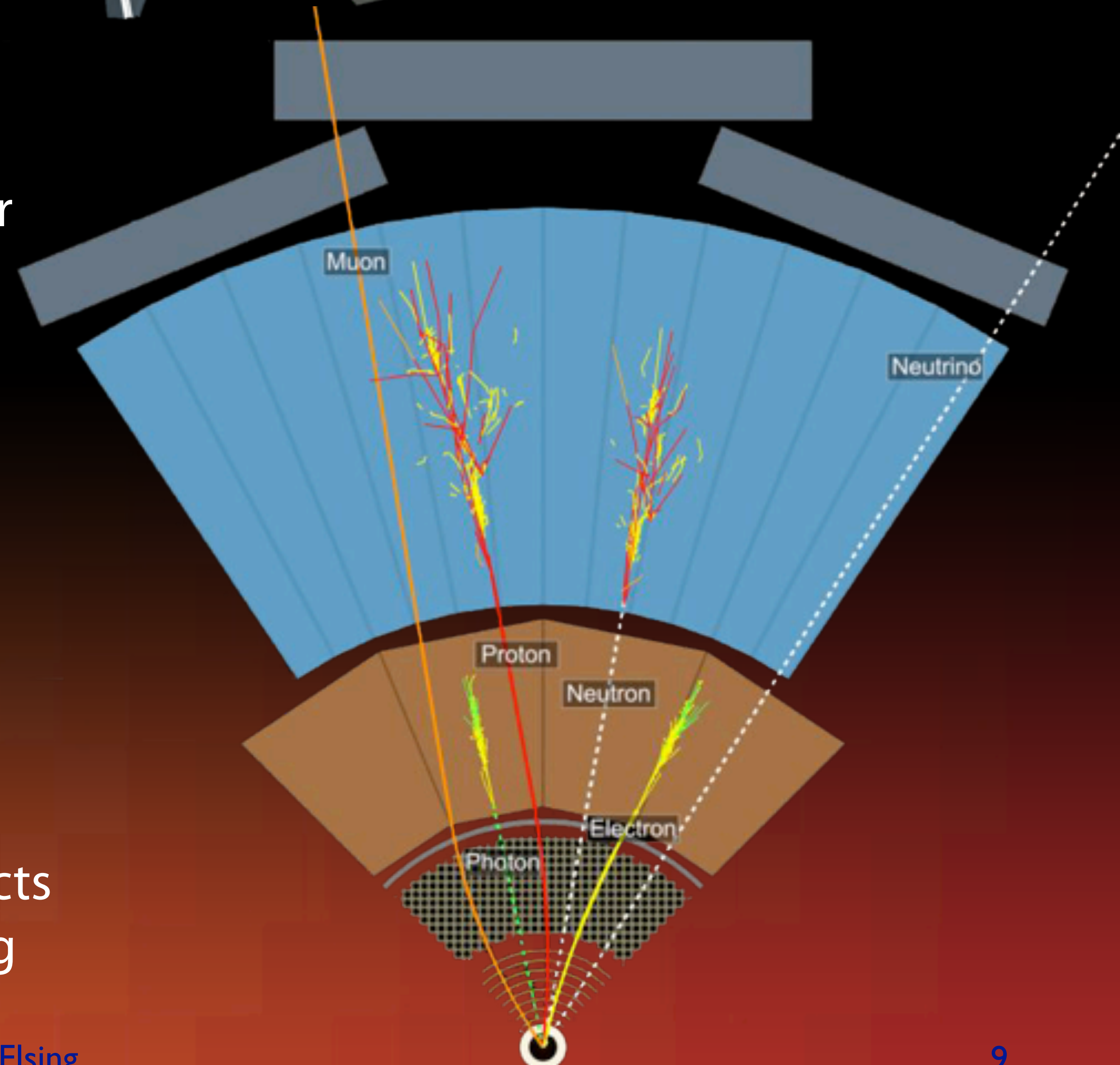


- experiment operation at Point-1 is domain of Online Software
  - ➡ sophisticated 3 level Trigger system to select events
    - Level-1 in hardware
    - Level-2 and Event Filter in software (so called HLT)
  - ➡ Data Acquisition system to collect event data
  - ➡ ATLAS writes raw events at a nominal rate of 200 Hz
- raw events are "Byte Stream" encoded
  - ➡ "Event Format" defined as collection of data blocks from Read-Out-Drivers (RODs)
  - ➡ each data block contains information digitized in front-end electronics of a particular piece of detector
  - ➡ detailed Level-1 and HLT results in special ROD blocks
- Byte Stream data is converted into RDOs in ATHENA for offline event reconstruction and analysis

Simulation

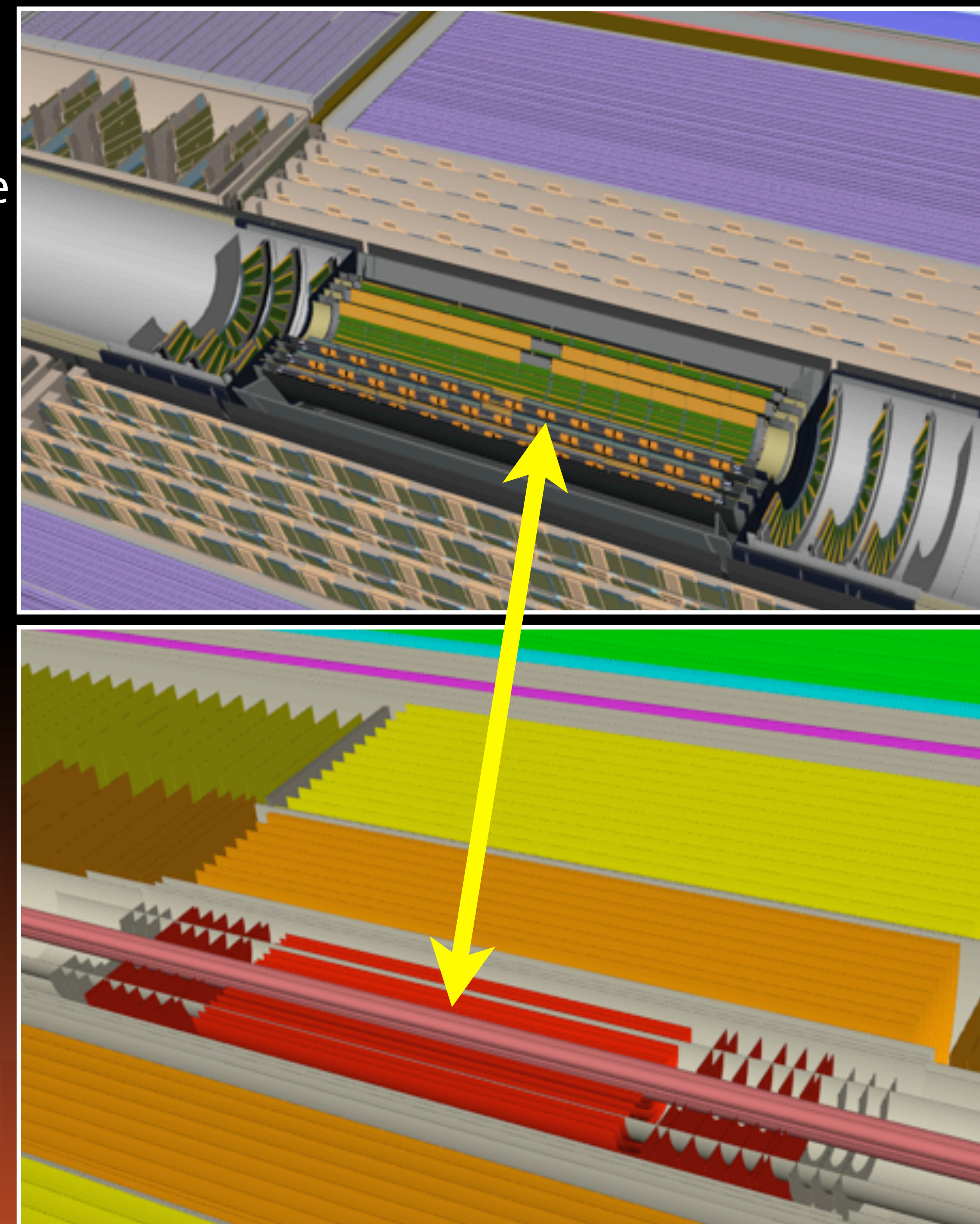
- Generators
  - ➡ Monte Carlo generators for pp interactions, output in **FGEN**
- Geant4 (G4) - Simulation
  - ➡ reads (filtered) FGEN events
  - ➡ tracks each particle through detector
    - detailed magnetic field
    - full list of physics processes
    - detailed description of material
  - ➡ G4 geometry built from **GeoModel**
  - ➡ output are energy deposits or **HITS**
- Digitization
  - ➡ detailed emulation of response to energy deposits in detectors
  - ➡ include electronics and readout effects
  - ➡ produces raw data or **RDO** as coming from ATLAS detector



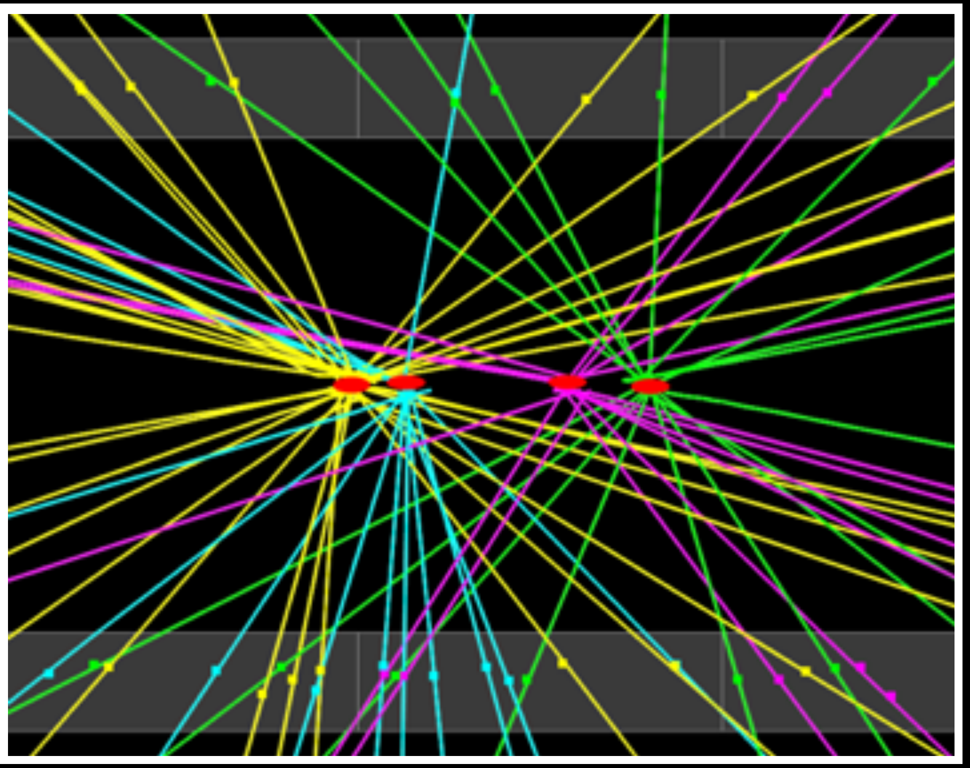


Fast Simulation, Simplified Geometries

- full G4 simulation is slow
  - 2000 KSi2Ksec ➡ precise simulation with full detail of physics processes and detector response
- G4/GeoModel geometry
  - ➡ tracking particles through is slow
  - ➡ simplify geometries
    - 4.8 M volumes in G4
    - 0.01 M volumes in FATRAS+ATLFAST II
- fast simulation
  - 760 KSi2Ksec ➡ G4 + fast shower
    - replace detailed simulation of showers
  - 8 KSi2Ksec ➡ FATRAS+ATLFAST II
    - parameterized calorimeter response
    - tracking with simplified detector description and physics lists
    - output are hits fed into digi+reco
  - 0.1 KSi2Ksec ➡ ATLFAST I:
    - fully parameterized
    - output are physics objects



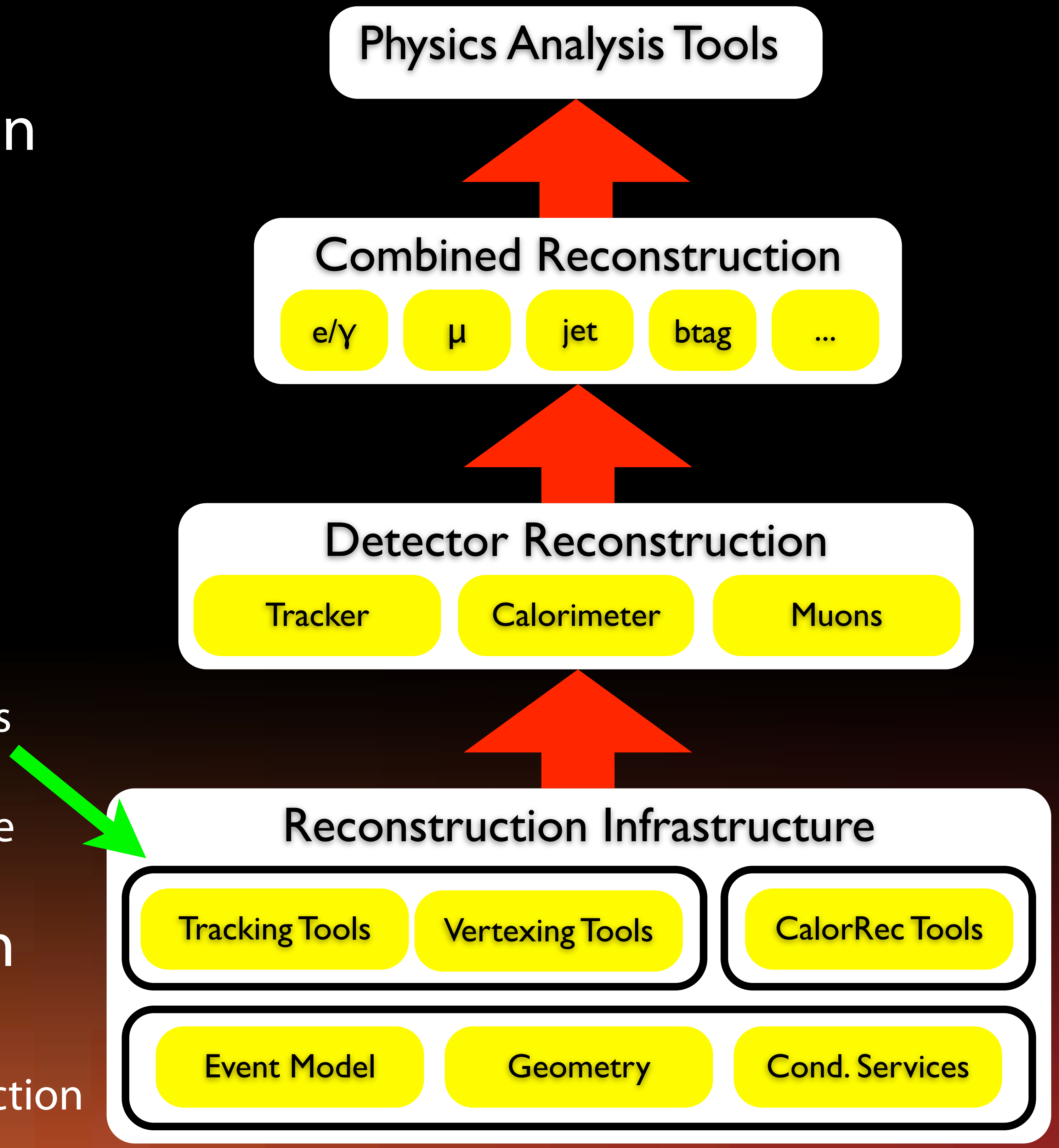
Event Overlay for Pileup



- at LHC luminosities
  - ➡ several interactions per bunch crossings
  - ➡ display of an event with pileup
- Event-Overlay for simulated data
  - ➡ add "minimum bias" to (high-p_T) physics events
  - ➡ add beam- and cavern-backgrounds
- HIT level Event-Overlay
  - ➡ simulated HITs for all inputs, add energy deposits and feed into digitization
  - ➡ advantage: uses same detailed modeling of detector response
  - ➡ mostly used today
- real data Event-Overlay
  - ➡ uses real data RDOs for minimum bias and backgrounds
  - ➡ overlay with signal HITs from simulation, model effects of detector response
  - ➡ advantage: real data backgrounds and no problems with statistics
  - ➡ under development to solve technical problems (e.g. alignment)

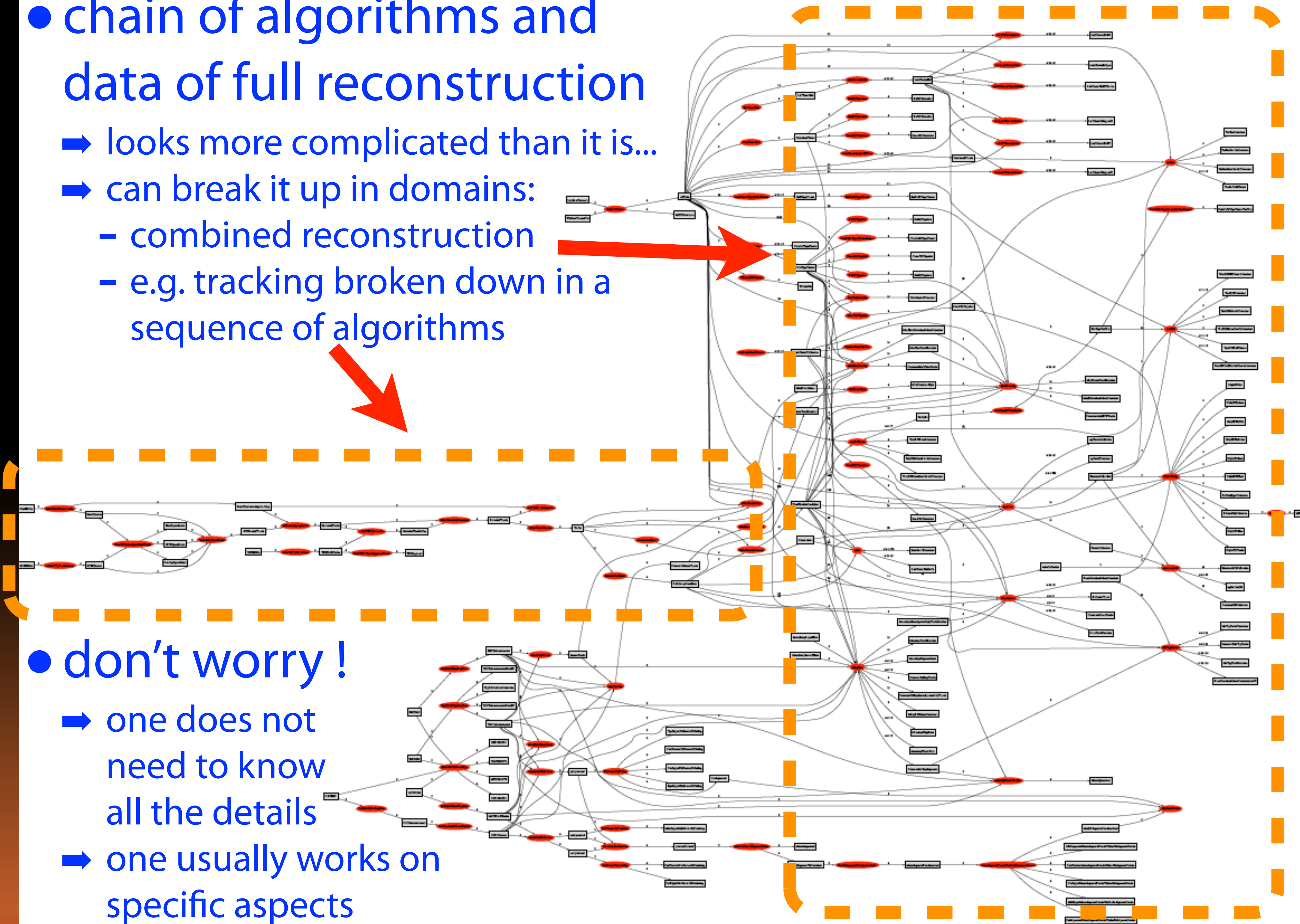
Reconstruction

- “natural” SW organization
  - ➡ first run detector specific reconstruction algorithms
    - – for tracks and calorimeter clusters, ...
  - ➡ then combined reconstruction
    - – identify physics objects
- based on infrastructure
  - ➡ common Tools and Services
  - ➡ e.g. tracking and vertexing tools
    - – fitting, propagation...
  - ➡ used in all software layers above
- common code base with High Level Trigger
  - ➡ regional vs full event reconstruction



And in Detail? ... RecExCommon

- chain of algorithms and data of full reconstruction
  - ➡ looks more complicated than it is...
  - ➡ can break it up in domains:
    - combined reconstruction
    - e.g. tracking broken down in a sequence of algorithms



- don't worry !
  - ➡ one does not need to know all the details
  - ➡ one usually works on specific aspects

Output Data Formats

- FGEN
  - ➡ Monte Carlo generator output in HepMC
- HITS
  - ➡ G4 output as energy deposits in detector
  - ➡ includes "truth information" with link between hits and particles in HepMC
- RAW/RDO (Raw Data Objects)
  - ➡ bit encoded event as produced by the readout of the experiment
    - includes detailed trigger result
  - ➡ simulated data: includes "truth information" for each RDO

dESD

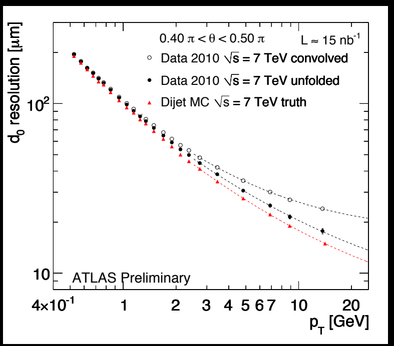
several derived skims

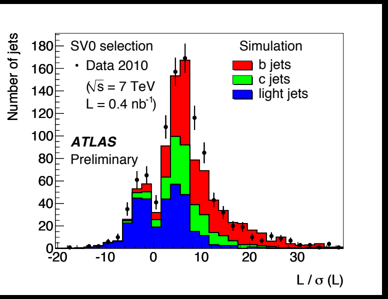
- ESD (Event Summary Data)
  - ➡ detailed reconstruction output
    - clusters and calibrated calormeter cells...
    - tracks, calo-clusters, vertices...
    - identified leptons (e/μ/τ), photons, jets, b-jets, V^0...
    - simulated data: "truth information"
- AOD (Analysis Object Data)
  - ➡ reduced format for physics, less detector information
- D3PD (Derived Physics Data)
  - ➡ several ntuple formats for analysis in ROOT
- TAG
  - ➡ summary information for fast event lookup

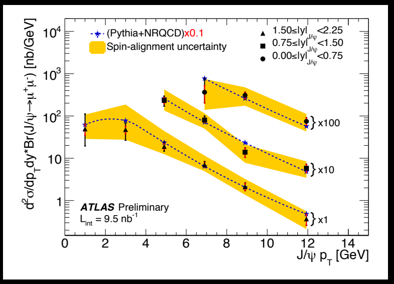
event size and level of detail

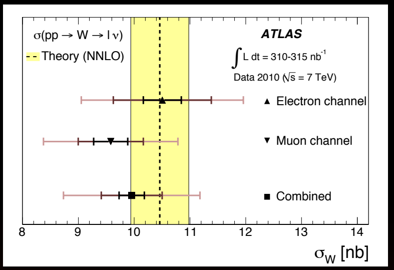
Physics Analysis Tools

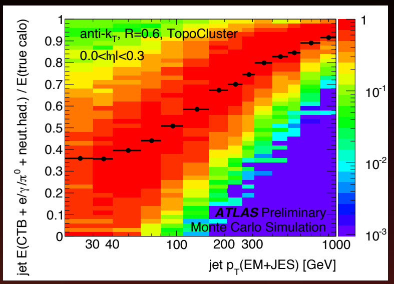
- several options to develop your analysis software
  - ➡ analysis groups usually use common software (e.g. for top reconstruction)
- physics analysis in ATHENA
  - ➡ read ESD or AOD, may use TAG data to pre-select events
  - ➡ full access to ATHENA Tools and Services, especially Meta Data
  - ➡ write ntuples or produce histograms
- ATHENA-Root-Access (ARA)
  - ➡ read ESD, AOD or TAG using ATHENA converters directly in ROOT
- ROOT or PYROOT
  - ➡ analyze D3PDs or your ntuples directly in ROOT
  - ➡ most popular, especially for final stages of analysis

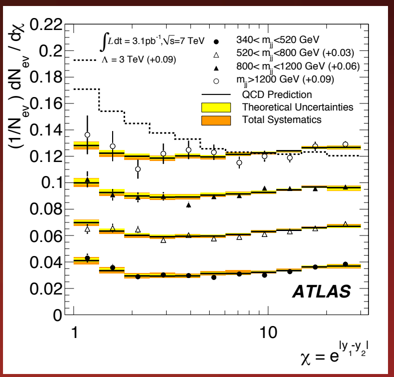






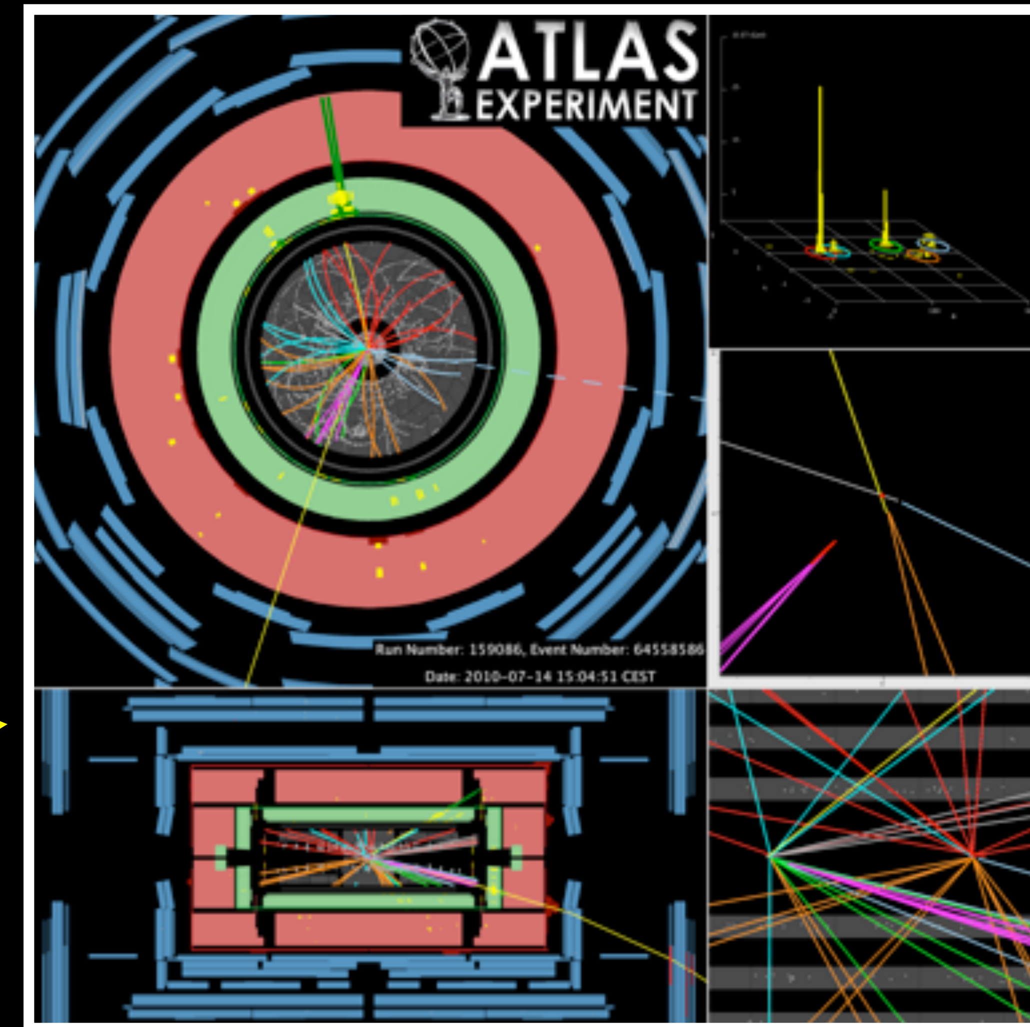


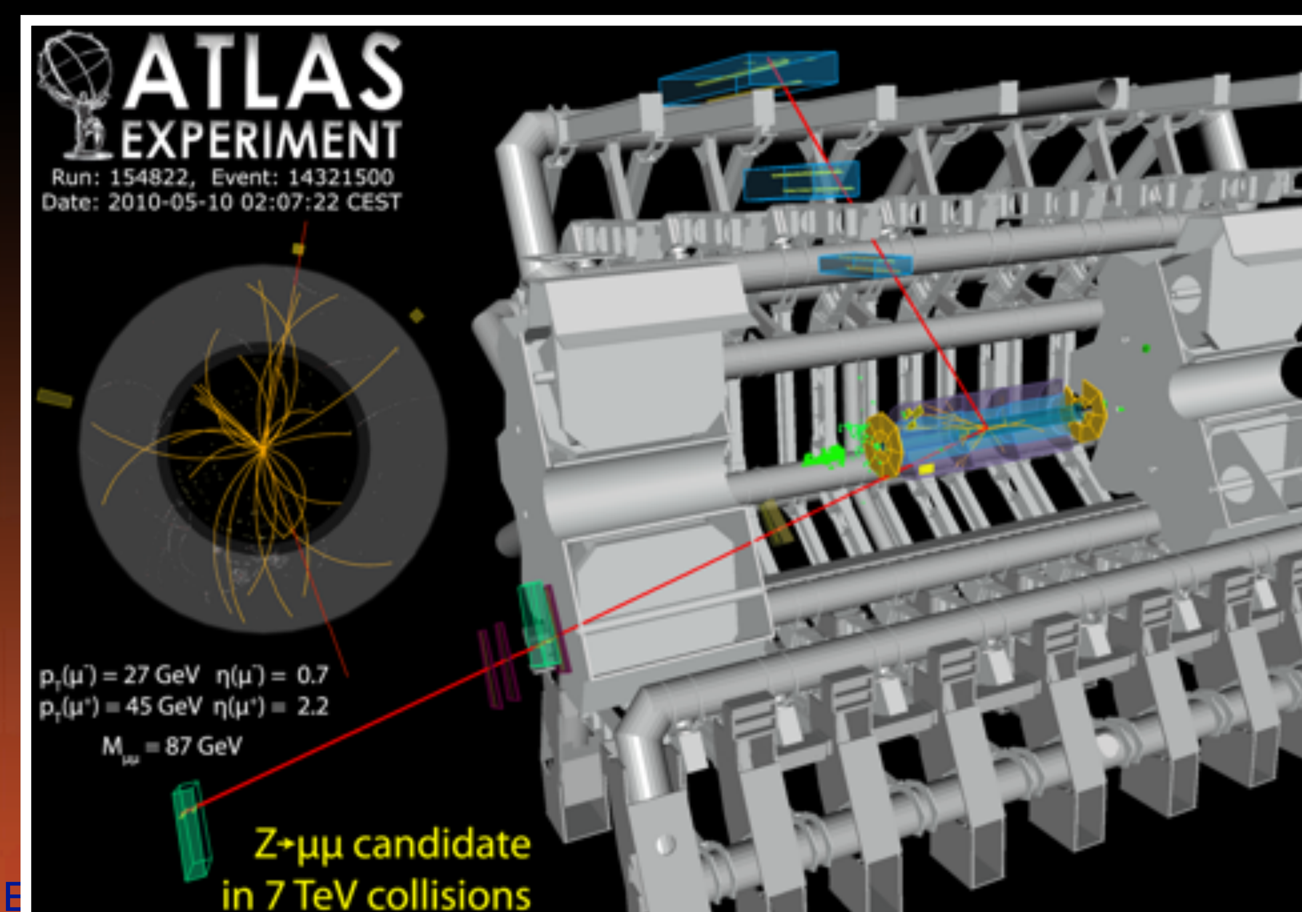




Event Displays

- several displays available
  - ➡ good tool to inspect your candidate events
  - ➡ helps to debug reconstruction and detector
- ATLANTIS
  - ➡ works in clever 2D projections
  - ➡ JAVA software runs on any laptop
  - ➡ need to convert events into JiveXML
- VP1
  - ➡ 3D event display in C++
  - ➡ it is an ATHENA application
  - ➡ makes use of full power of ATHENA Tools and Services
  - ➡ still under development
- not to forget PERSINT and AMELIA...





Software Releases

- basic software unit is a package
  - ➡ each package is entirely defined by its path and name
    - e.g. /Reconstruction/RecExample/RecExCommon
  - ➡ and its versioning tag
    - e.g. RecExCommon-00-09-00

- a software release is defined by its list of package tags

- frequent release strategy
  - ➡ new major release is built roughly every six months
  - ➡ developer release rebuilt (very) approximately every month
  - ➡ bugfix releases every (few) weeks
  - ➡ caches as often as necessary

- numbering scheme:

16.0.1.6-AtlasProduction

major release

developer release

bugfix release

release cache

release project

Software Development Infrastructure

- the repository: Subversion (SVN)
  - ➡ software packages are organized in directories by domain
    /Tracking/TrkFitter/TrkGlobalChi2Fitter
    /InnerDetector/InDetExample/InDetRecExample
  - ➡ provides support for versioning (tags), commit rights, etc.
- Configuration Management Tool (CMT)
  - ➡ used to check-out, resolve dependencies and built packages
- Tag Collector
  - ➡ web interface to manage integration of package versions into releases
- Nightly Build System (ATN)
  - ➡ compiles every night the set of release candidates under development
  - ➡ including automated software regression tests
- Run Time Tester (RTT)
  - ➡ automated daily performance validation of software release candidates

How Analysis looks like in practice ?

- typical workflow:
  - ➡ setup your environment using **CMT**
  - ➡ identify set of runs satisfying specific criteria with the **Run Query Tool**
  - ➡ look for datasets to analyze using a metadata browser **AMI**
  - ➡ download a few files using GRID Data Management client tools in **DQ2**
  - ➡ inspect the files using Athena-ROOT-Access (**ARA)** or look at the events visually with **ATLANTIS**
  - ➡ develop your analysis code in **ATHENA** or **ARA** using the local data to check that it is doing what you expect
  - ➡ send your jobs to the Grid using **PATHENA** or **GANGA** to process large datasets
  - ➡ download the results using **DQ2**
  - ➡ work on final small ntuple and make histograms using **ROOT**

- by the end of the week you will have seen all elements of this workflow...

Further Information...

- Main computing page:
  - ➡ https://twiki.cern.ch/twiki/bin/view/Atlas/AtlasComputing
- Code browsing (password needed):
  - ➡ https://svnweb.cern.ch/trac/atlasoff/browser
- Documentation for beginners:
  - ➡ WorkBook: https://twiki.cern.ch/twiki/bin/view/Atlas/WorkBook
  - ➡ Physics analysis WorkBook: https://twiki.cern.ch/twiki/bin/view/Atlas/PhysicsAnalysisWorkBook
  - ➡ Tutorials: https://twiki.cern.ch/twiki/bin/view/Atlas/ComputingTutorials
- Help forums:
  - ➡ https://espace.cern.ch/atlas-forums/Lists/Atlas%20forums/By%20category.aspx
  - ➡ This one in particular: https://groups.cern.ch/group/hn-atlas-offlineSWHelp/default.aspx

... and more to come !

- Event Data Model and Detector description
  ➡ talks by James, Vakho and Achil
- Physics Analysis Examples
  ➡ talk by Miguel and James
- VP1 Event Display
  ➡ talk by Giorgi
- Datasets and AMI
  ➡ talk by Solveig
- after introduction to GRID Distributed Computing
  ➡ practical session how to access data, run jobs and do analysis