The ATLAS Event Data Model

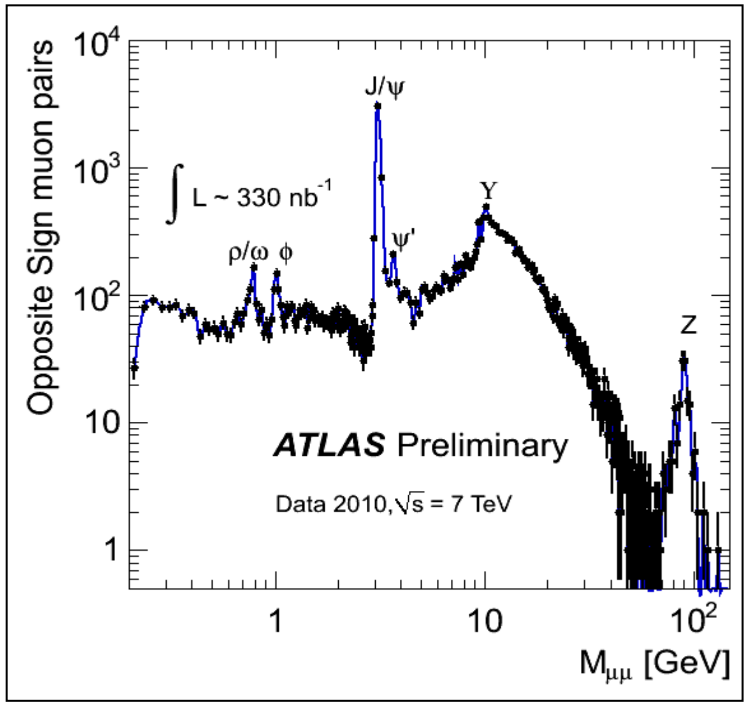
James Walder
Lancaster University

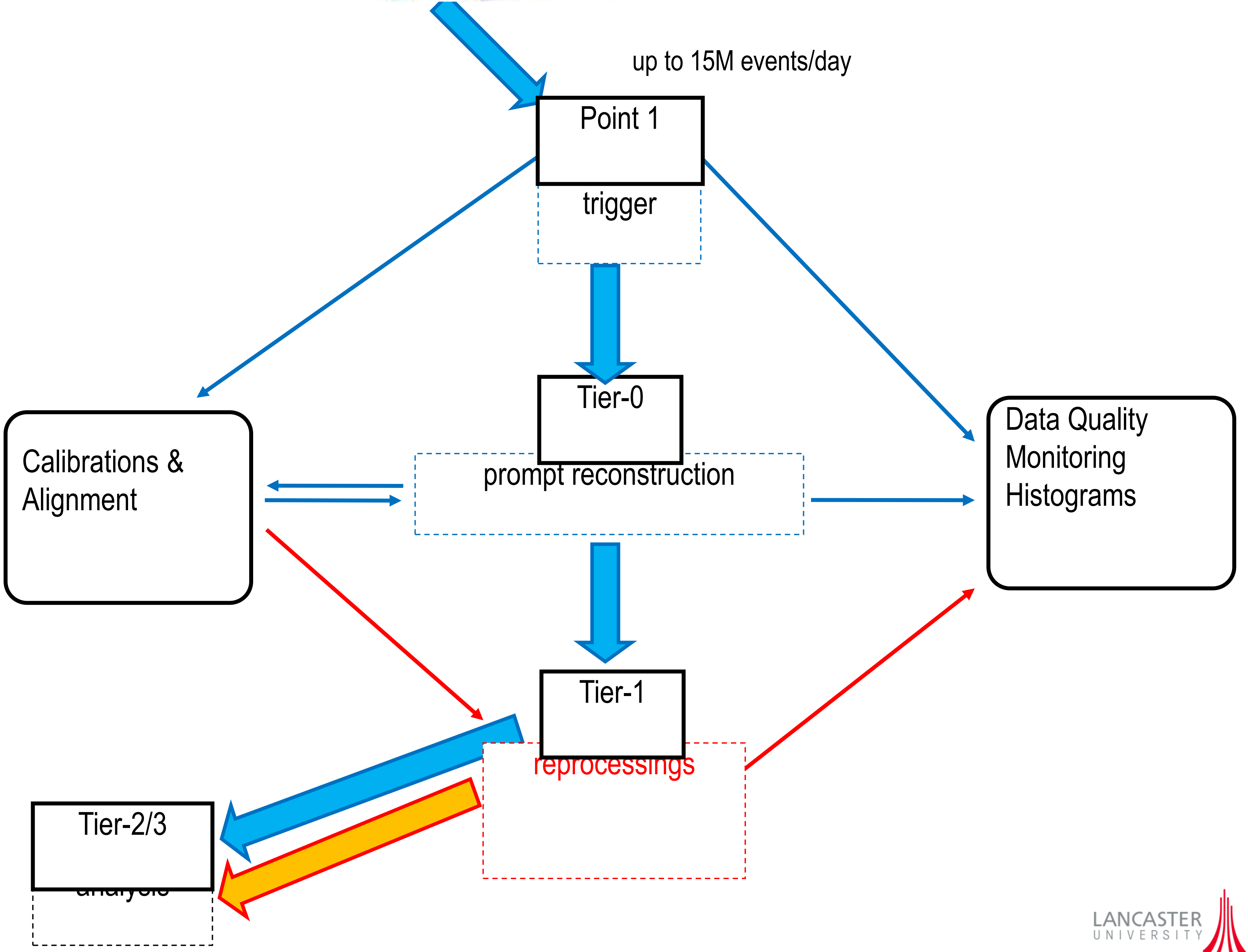
What is an Event Data Model (EDM)?

- For complex systems within large collaborations must provide a common and coherent approach to software interfaces and data objects
  - Allow for maintenance throughout lifetime of experiment
  - Flexibility to develop for upgrade components etc.
- For the purpose of ATLAS this means:
  - Must provide a well-defined set of tools and methods to take raw data from each detector component
  - Prepare the Raw data in such a way to have common input reconstruction algorithms, for example pattern recognition.
  - Create the basic analysis objects (tracks, muons, etc..) that will be used to in physics analysis.
  - Define tools for persistifying data and retrieving this data into the transient store
- Where possible, apply common interfaces and tools to simplify reconstruction.
  - Utilisation of the C++ concept of ***Inheritance***.and
  - Object-orientated class structure,
- Separation of event data classes from algorithmic classes
  - Provides modular flow.
  - Example: Track fitting algorithm reads input tracking objects and produces new tracking output – track itself does not contain any fitting code.

Data flow: From ATLAS to Results

- Non-trivial procedure to go from
- to:





up to 15M events/day

Point 1

trigger

Tier-0

prompt reconstruction

Calibrations & Alignment

Data Quality Monitoring Histograms

Tier-1

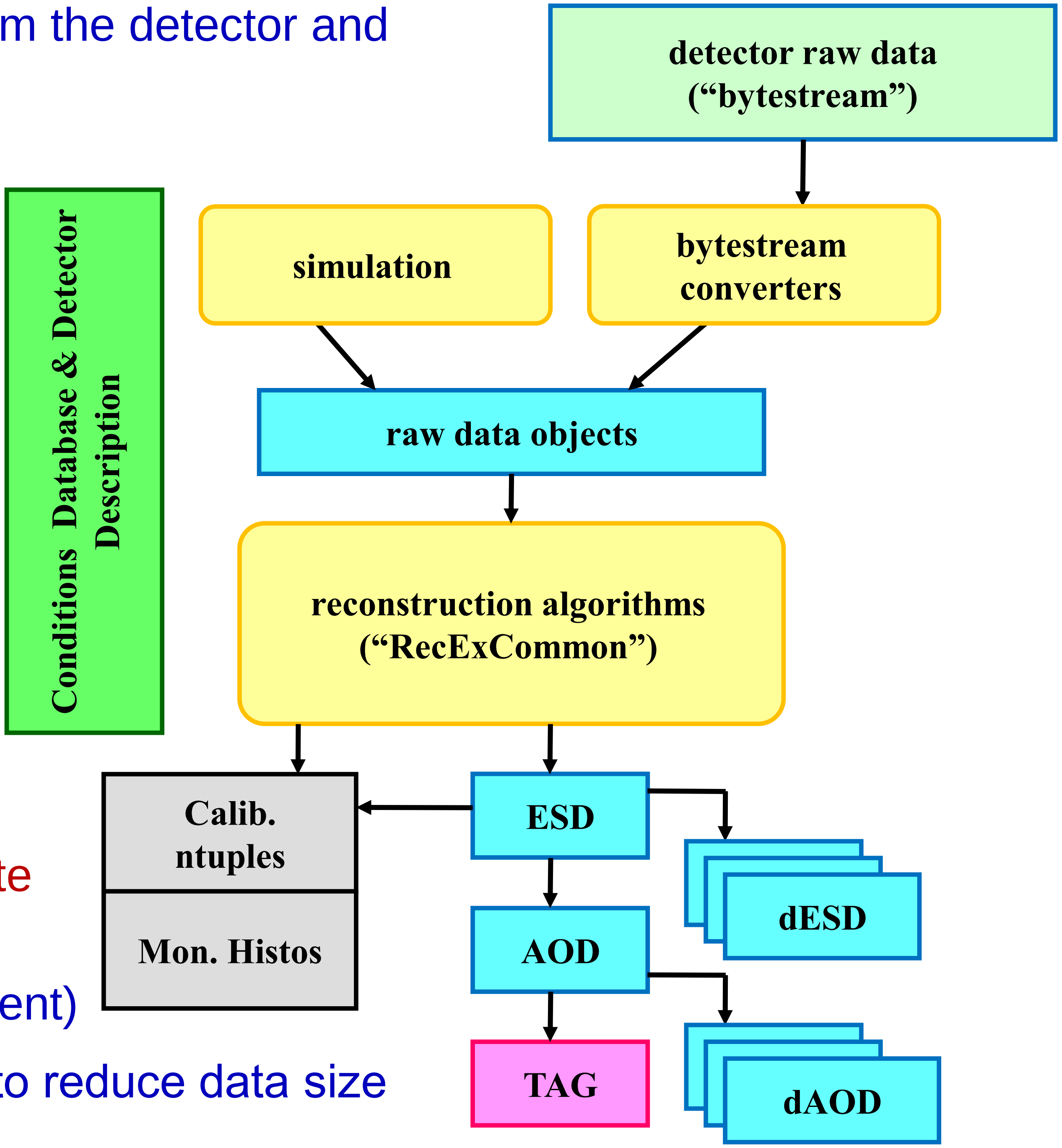
reprocessings

Tier-2/3

analysis

Reconstruction and Data Formats

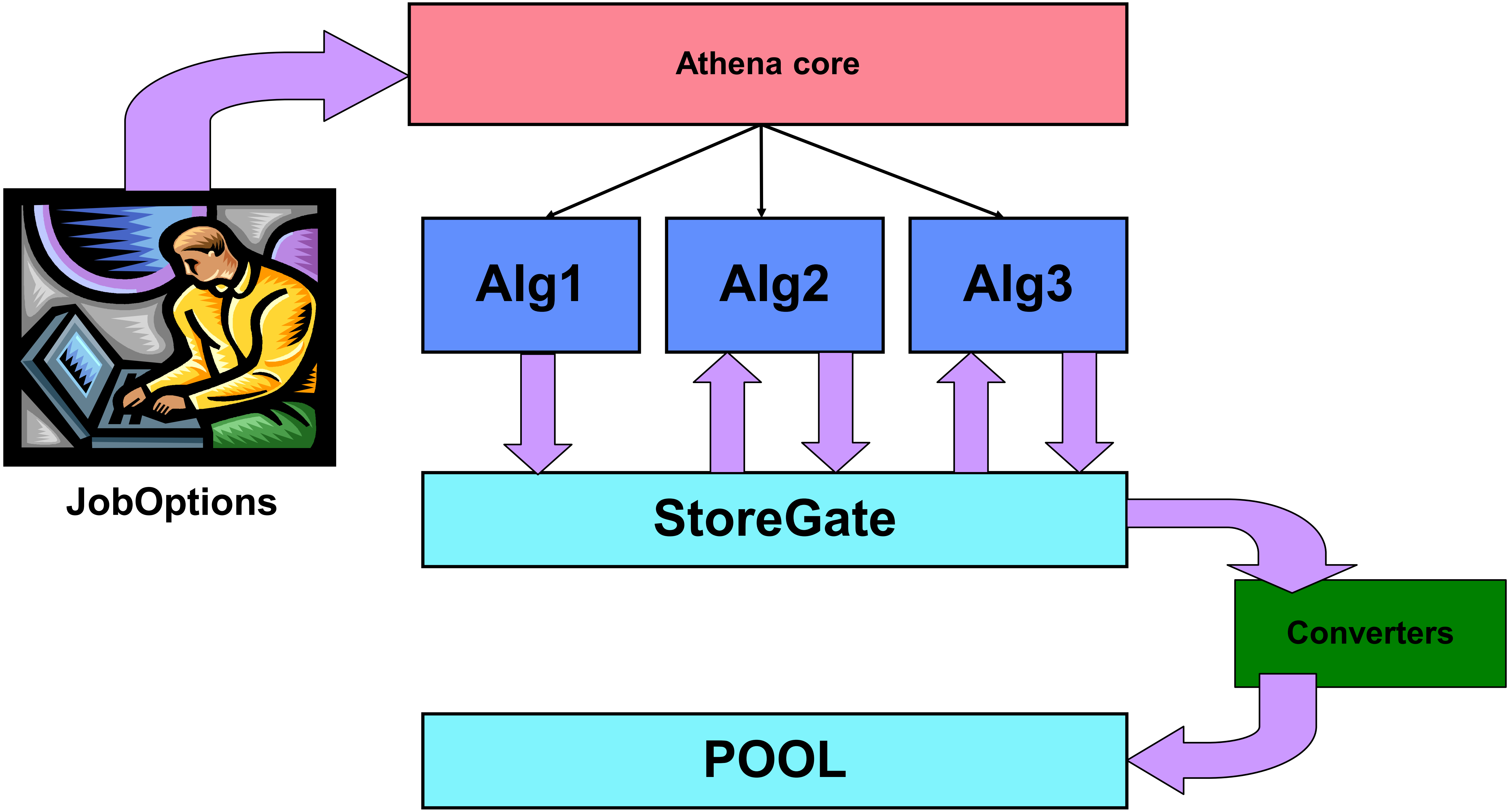
- Chain of processing to take raw input from the detector and create physics objects for analysis.
- RAW data input from detector (or simulation)
- Reconstruction on RAW using
  - Reco_trf.py / RecExCommon
- Producing data formats
  - ESD > AOD > TAG
- Event Summary Data (ESD)
  - Detailed output of detector reconstruction
- The AOD (Analysis Object Data)
  - Subset of ESD data, more appropriate for analysis.
- TAG is thumbnail of the event (~1kb / event)
- Filtering ('skimming') of events possible to reduce data size
  - dESD, dAOD.



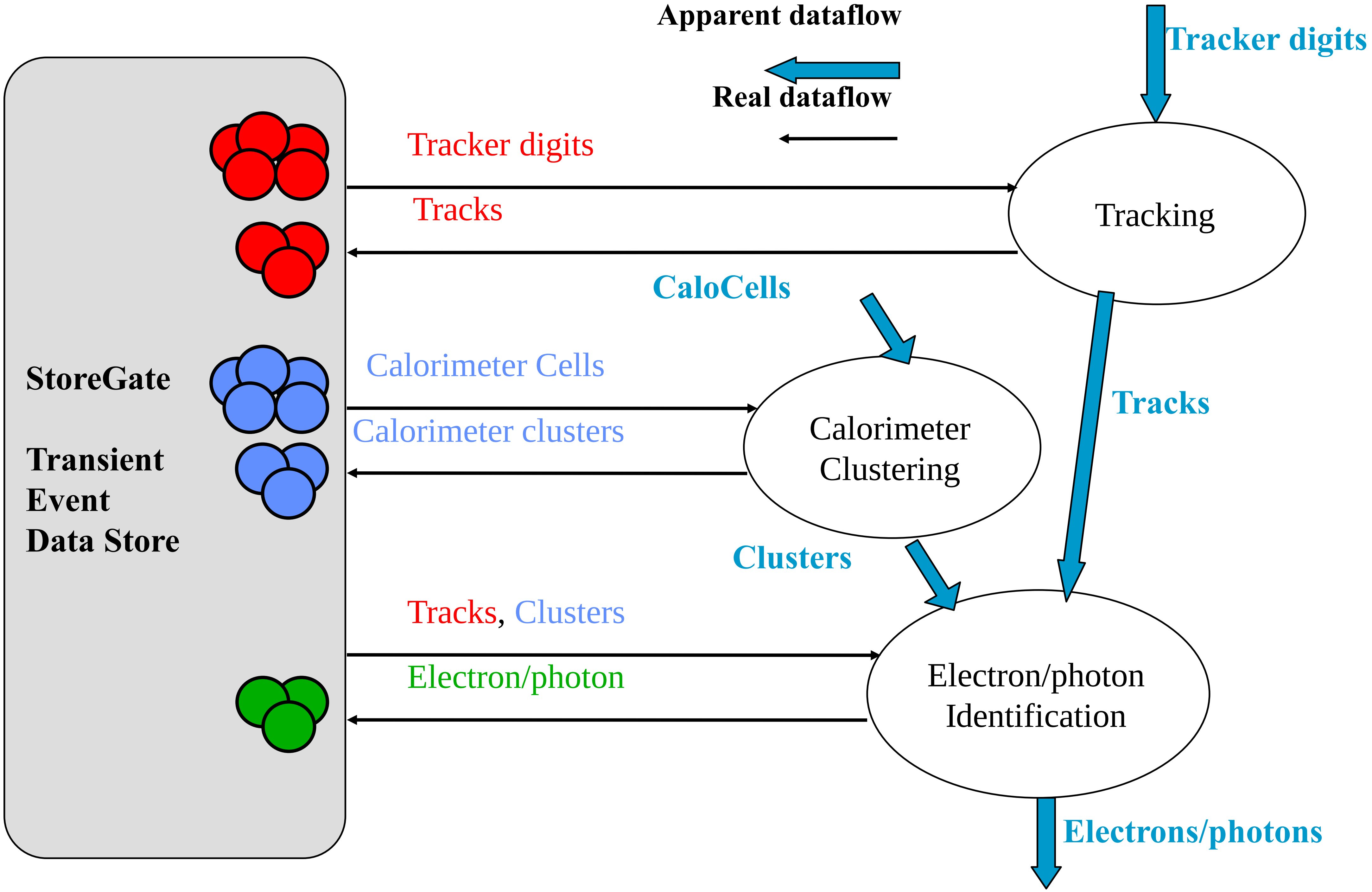
- Final analysis typically achieved using root ntuples (e.g. D3PD)

ATLAS Scheme

- Demonstrates separation of the Data objects from the Algorithm classes,

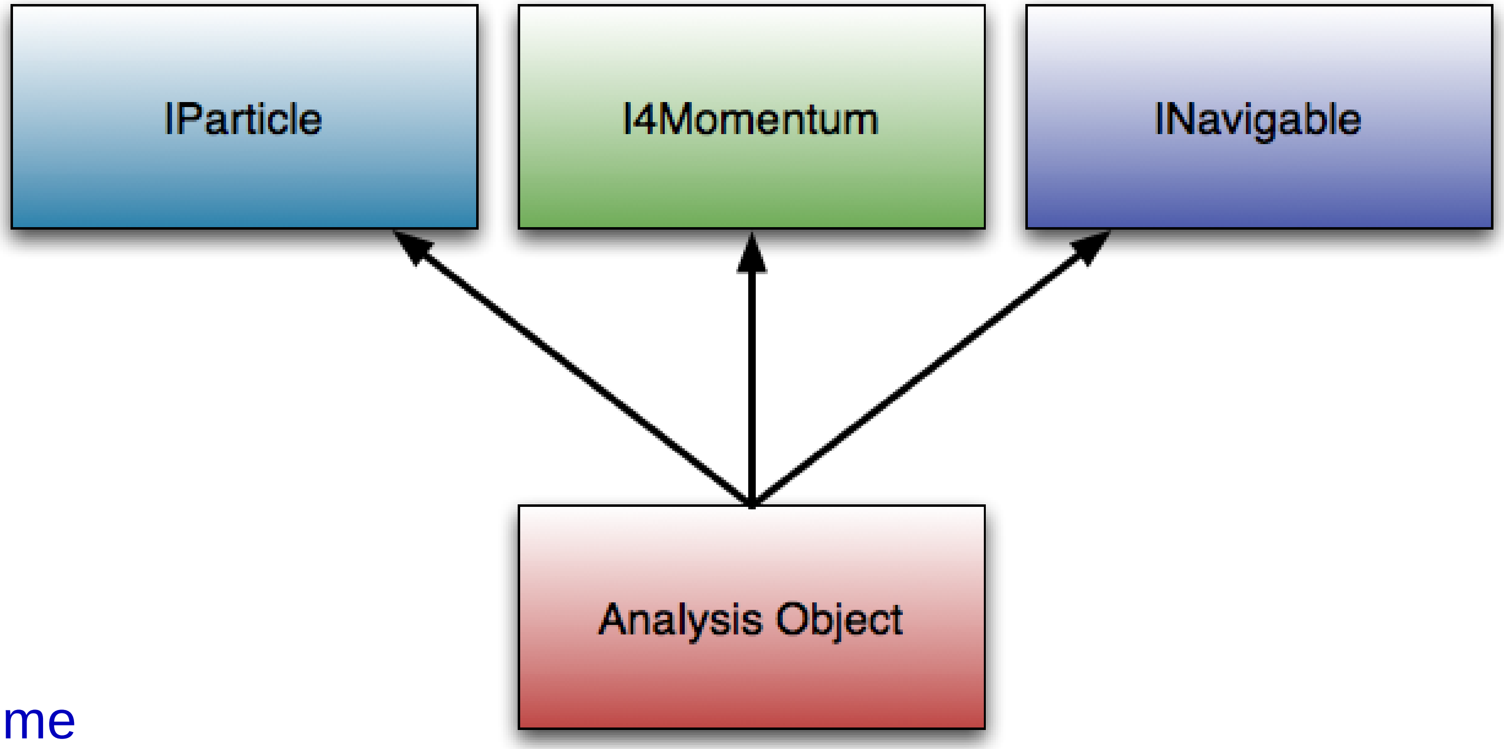


Example of implementation



ATLAS EDM Objects for User Analysis

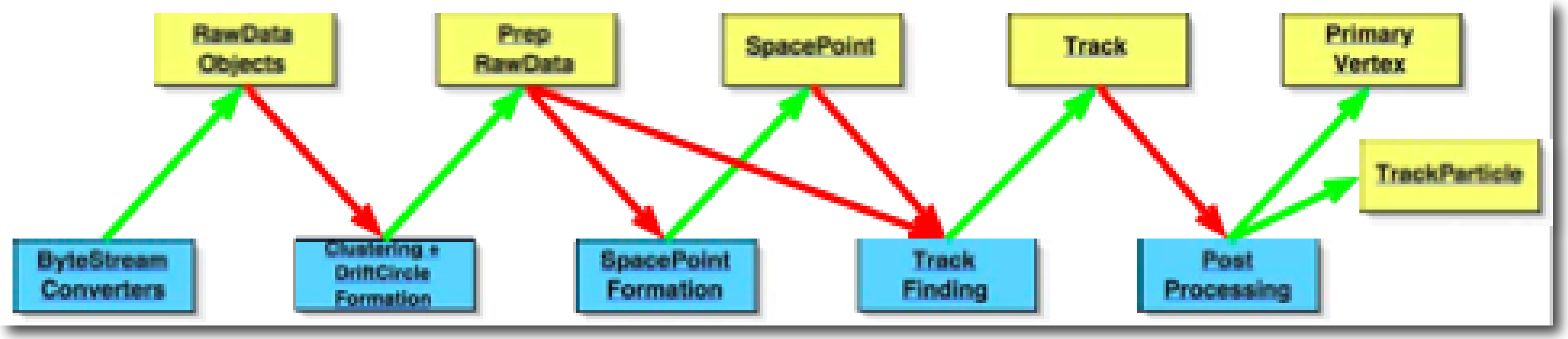
- For User analysis the main objects you will be interacting with are:
  - Trigger
  - EventInfo
  - TrackParticle (also Trk::Track)
  - Muon
  - Jet
  - Egamma
  - Tau
  - MissingET
  - Monte Carlo Truth information



- Many of these objects share some common interfaces:
  - 4-momentum
  - Charge, pdgID, vertex type information
  - Possible links to constituent objects
- Will next present some information for each of the main EDM physics objects.
  - In each example, the list is **not** exhaustive.
- ESD / AOD summary for release 15:
  - https://twiki.cern.ch/twiki/bin/viewauth/Atlas/AODClassSummary15
- With thanks to the many contributors on the following slides.

Tracking reconstruction chain

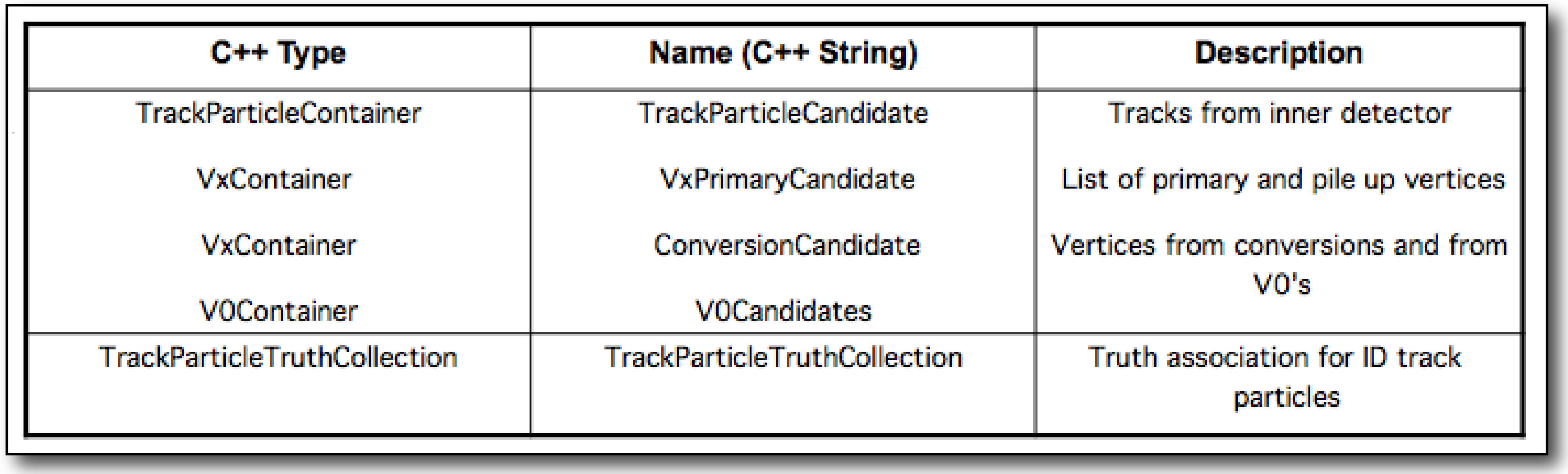
- Tracking software must support input from: Pixel, SCT, TRT, Muon detectors and create tracks using common track fitting algorithms.
  - Common Vertex fitting interface is also defined.
- Simplified idea of steps required to create Vertex and Track objects



- Track object is a detailed object,
  - TrackParticle is created to reduce data size, whilst keeping summary information such as: numbers of hits,
    - Although removes the details of the hit measurements themselves.
  - As AOD does not store individual hits, means must use ESD to re-run track finding, for example.

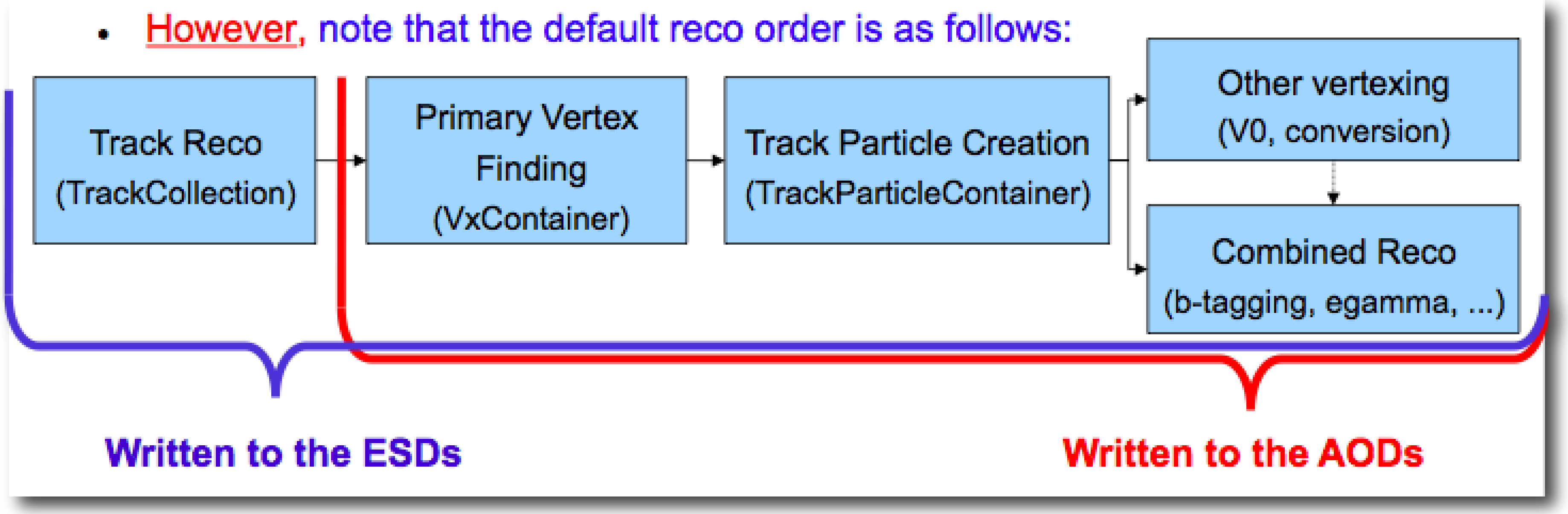
Inner Detector EDM

- For the Inner detector, the common analysis objects are given by:



C++ Type	Name (C++ String)	Description
TrackParticleContainer	TrackParticleCandidate	Tracks from inner detector
VxContainer	VxPrimaryCandidate	List of primary and pile up vertices
VxContainer	ConversionCandidate	Vertices from conversions and from V0's
V0Container	V0Candidates	
TrackParticleTruthCollection	TrackParticleTruthCollection	Truth association for ID track particles

- VxCandidate represents a reconstructed vertex, and contains:
  - Position and Covariance matrix, plus a link to the Track(Particles) used in the fit,
  - Can re-run Primary Vertex finding on AOD using TrackParticle



- However, note that the default reco order is as follows:

Track Reco (TrackCollection)
Primary Vertex Finding (VxContainer)
Track Particle Creation (TrackParticleContainer)
Other vertexing (V0, conversion)
Combined Reco (b-tagging, egamma, ...)
Written to the ESDs
Written to the AODs

Summary

- The ATLAS EDM is designed to provide:
  - Common interface to access data objects and perform standard reconstruction tasks, such as track finding
  - Stable interface over the lifetime of the experiment

- Data types designed for detailed studies, (d)ESD
  - Or general analysis, (d)AOD;
  - Both in a pool.root POOL format
- Common n-tuple making format (D3PD) framework also is quickly being adopted across groups.

- The Analysis-object Classes provides:
  - Common interfaces to access 'main' physics quantities (e.g. 4-mom.)
  - Quantities specific to each object type are also contained (e.g Author)
- More examples on the EDM objects in Physics analysis examples talk.

Backup