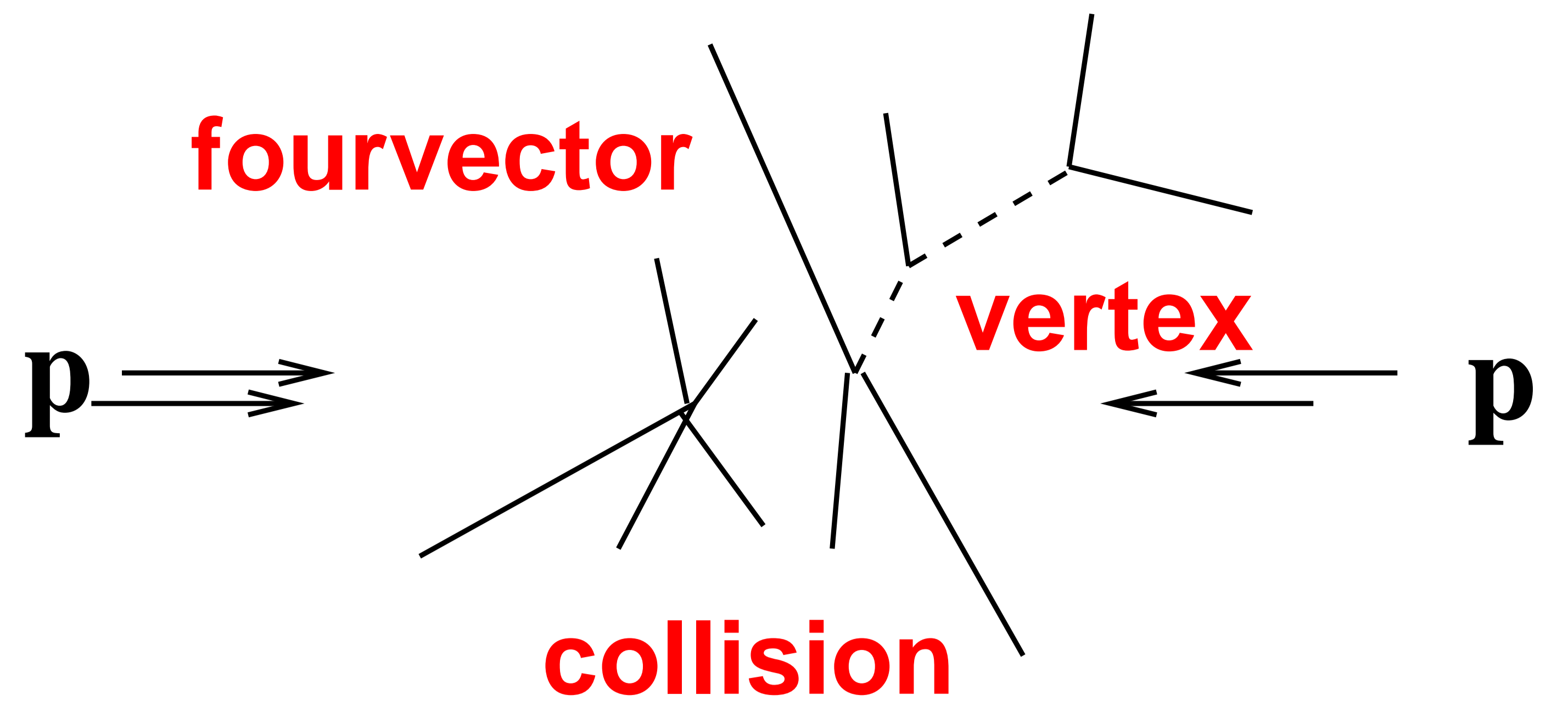




Physics Analysis eXpert

<http://cern.ch/pax>



New Applications of PAX in Physics Analyses at Hadron Colliders

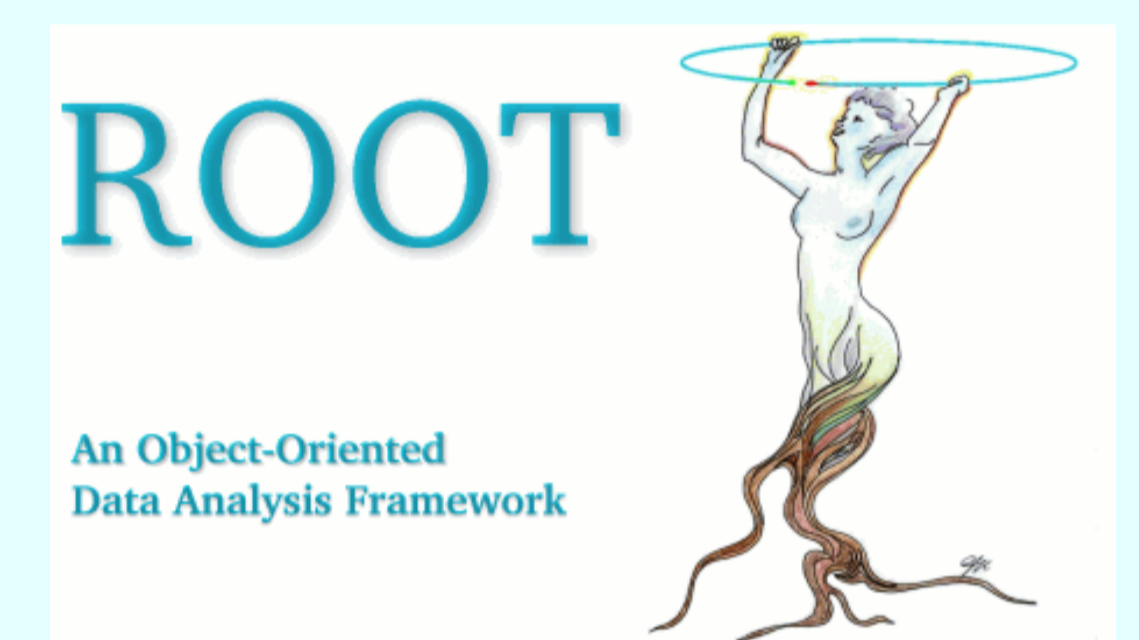
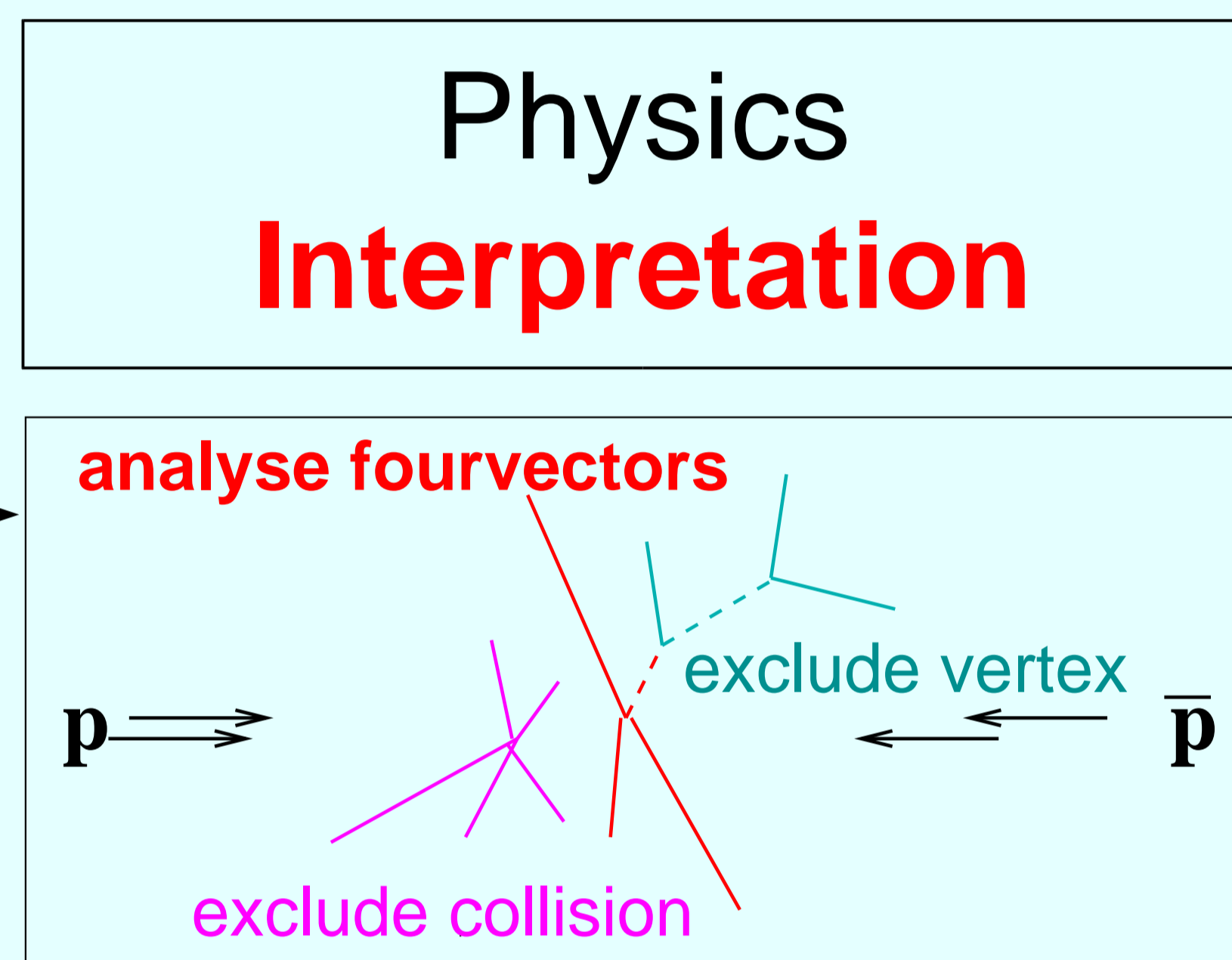
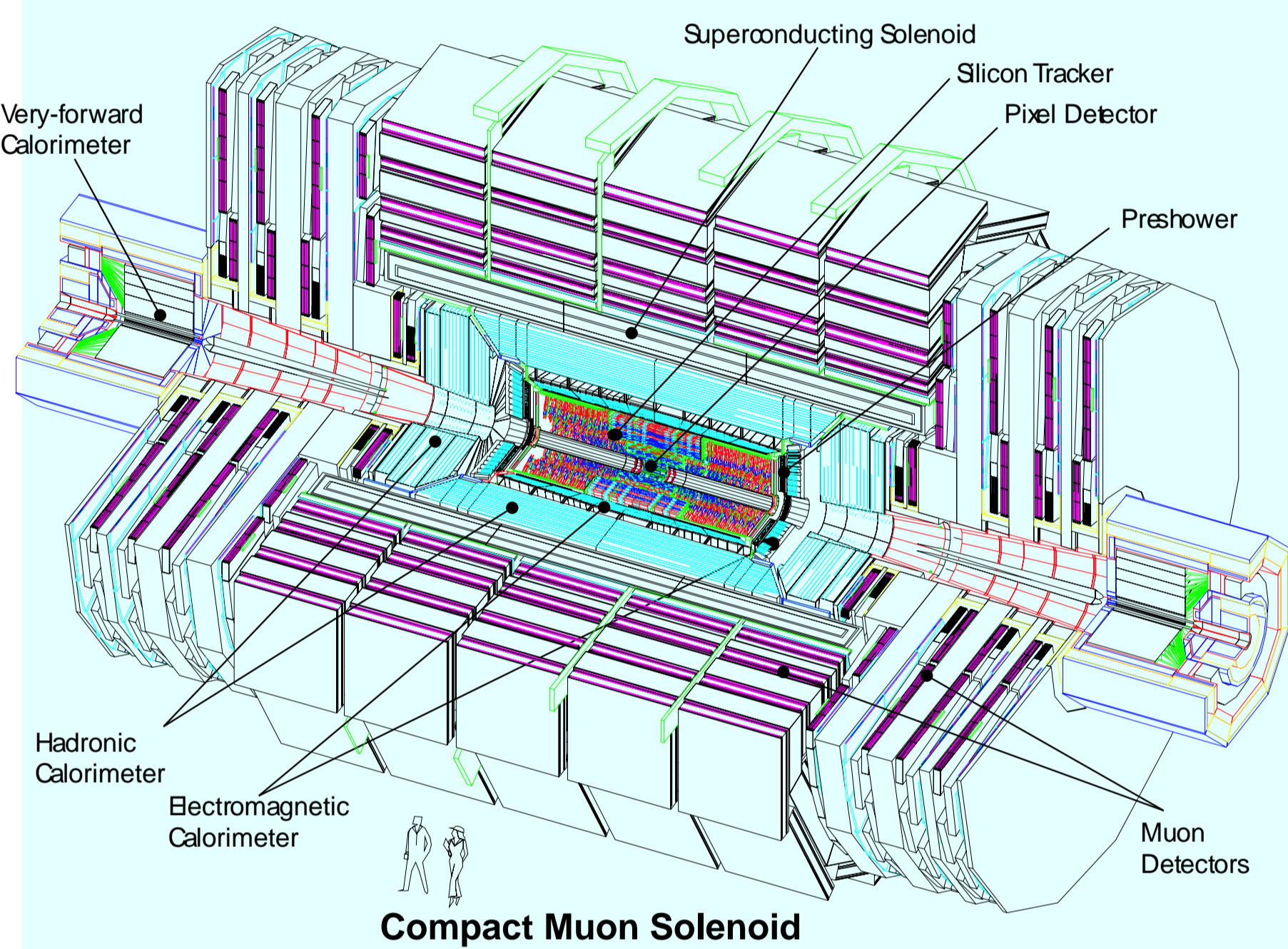
Authors:

Martin Erdmann (RWTH Aachen)
 Ulrich Felzmann (University of Karlsruhe)
 Dominic Hirschbühl (University of Karlsruhe)
 Christopher Jung (Forschungszentrum Karlsruhe)
 Steffen Kappler (University of Karlsruhe)

Matthias Kirsch (RWTH Aachen)
 Günter Quast (University of Karlsruhe)
 Klaus Rabbertz (University of Karlsruhe)
 Jens Rehn (CERN)
 Sven Schalla (University of Karlsruhe)

Patrick Schemitz (now at web.de)
 Alexander Schmidt (University of Karlsruhe)
 Thorsten Walter (University of Karlsruhe)
 Christian Weiser (University of Karlsruhe)

Mission:



etc.

The output of a detector simulation, reconstruction, monte carlo generator, or of real data....

....is interfaced to **PAX**. (Fourvector and Vertex objects)...

....**PAX** assists physicists in the physics interpretation of events, like establishing relations between objects (**decay trees**) and mastering combinatorial multiplicities....

....The result of this analysis is filled into....

....standard histograms and trees (ROOT or PAW).

Example Application:

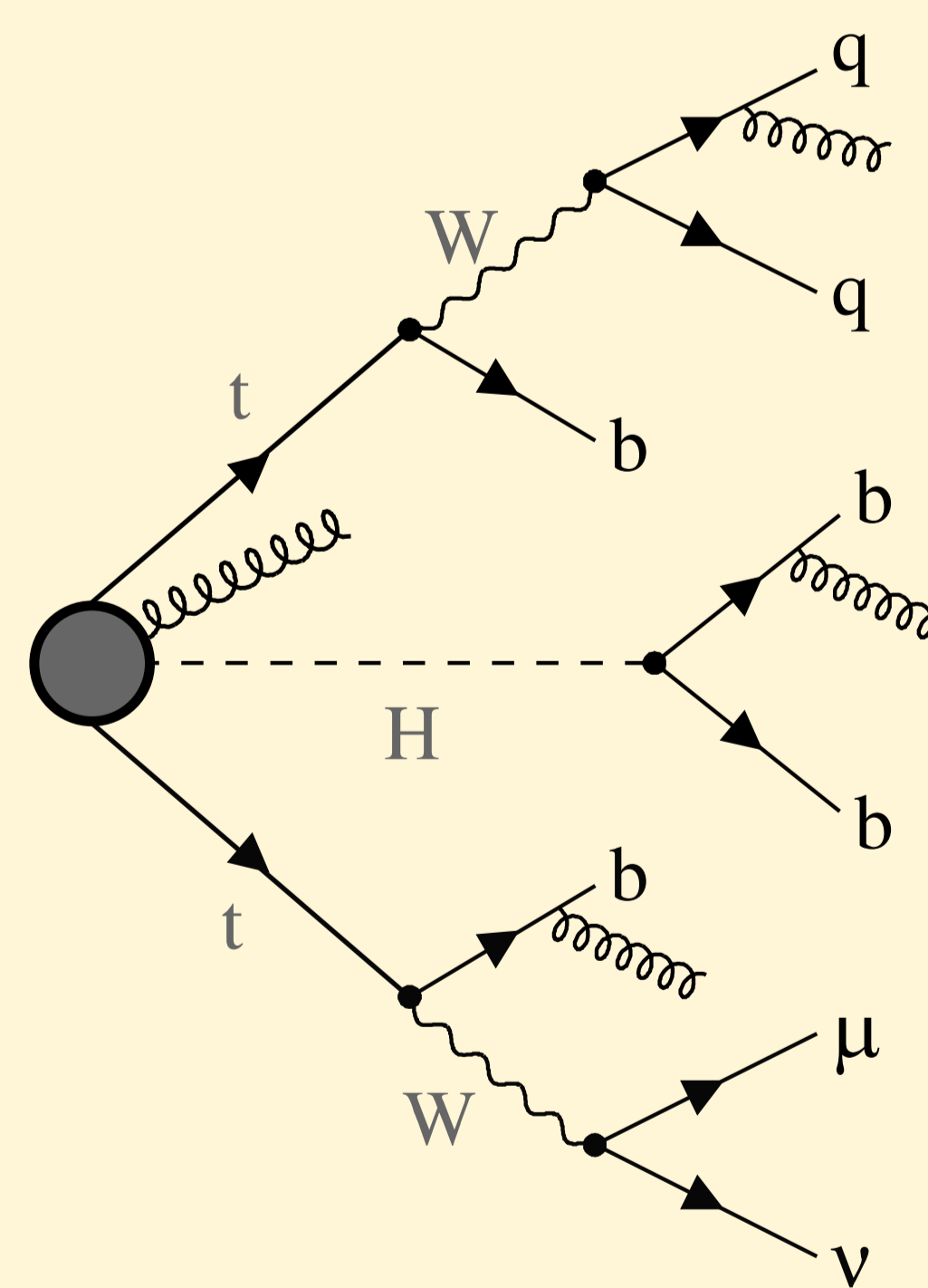
$$t\bar{t}H, H \rightarrow b\bar{b}$$

Challenge: Complete reconstruction of $t\bar{t}H$ partonic process

In this particular case: **24** possible interpretations in case of perfect b-tagging

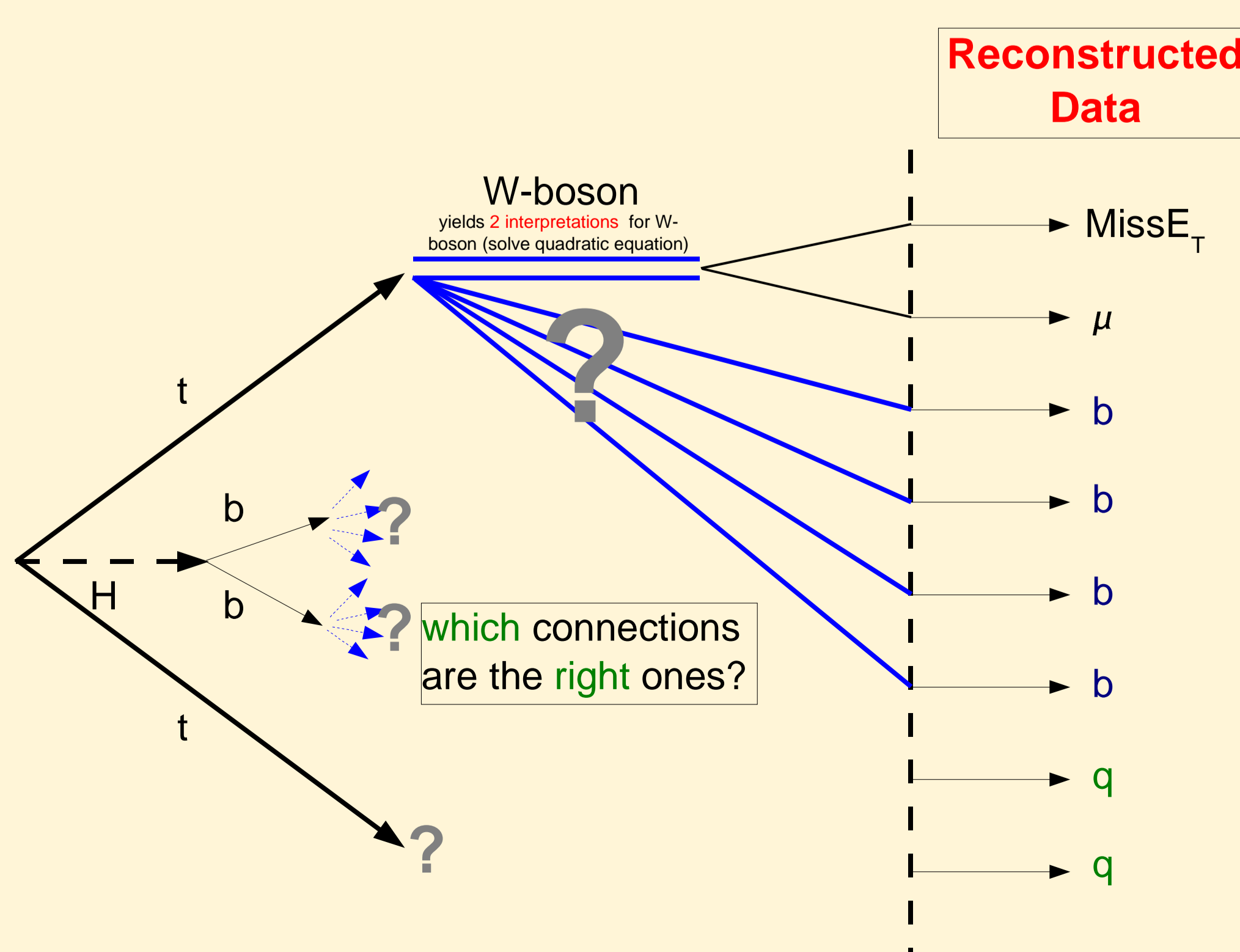
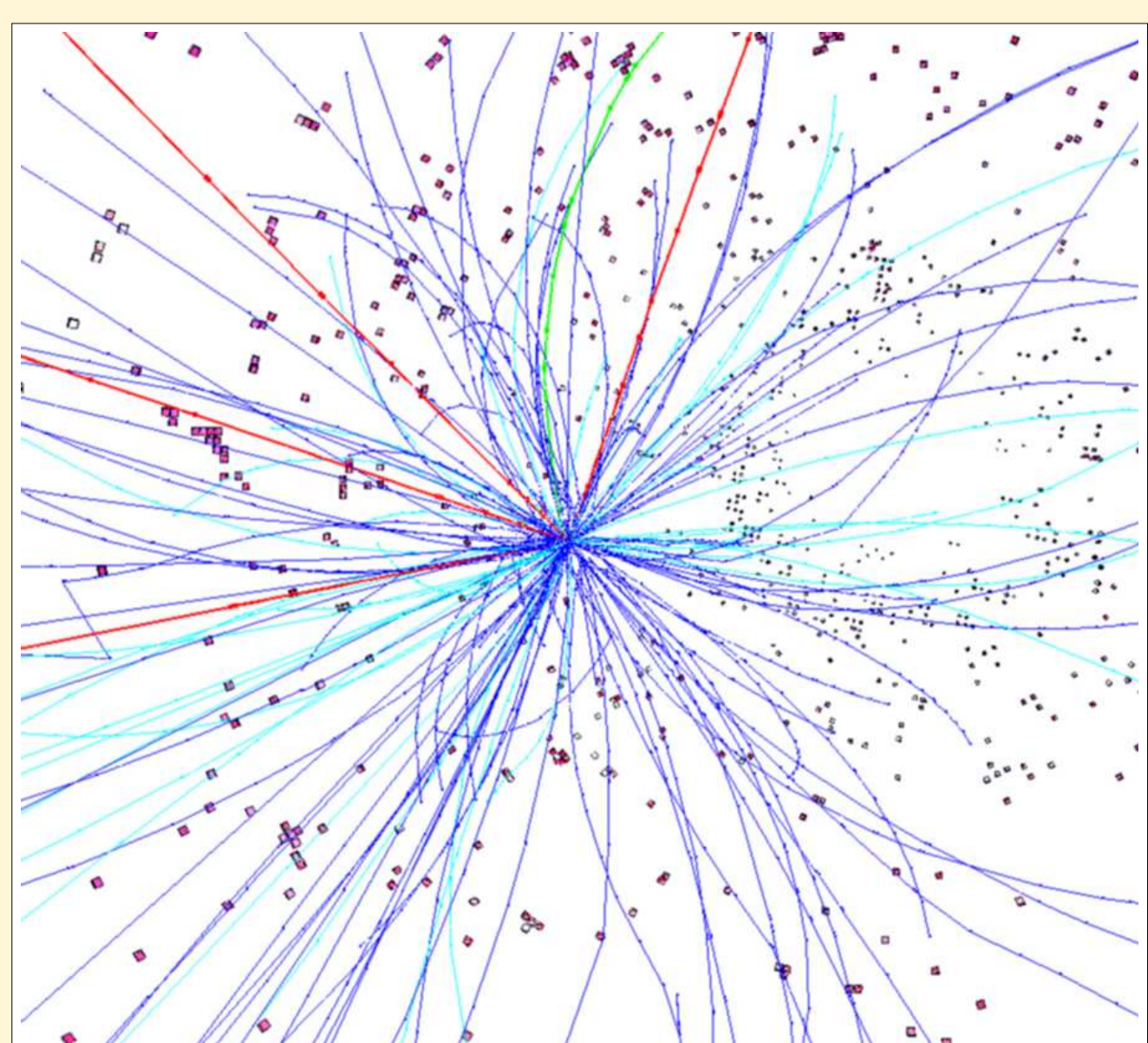
In case of gluon radiation or realistic b-tagging: have many more interpretation possibilities

Usage of **only one** analysis source code for studies on generator, detector simulation and data analysis level

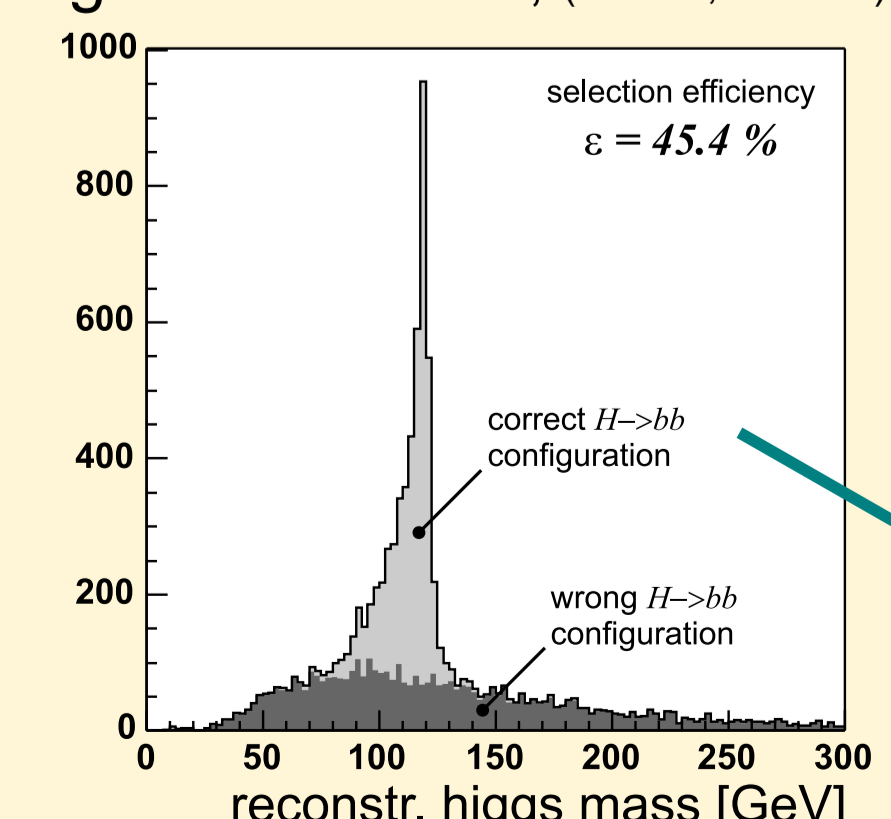


=> need to manage

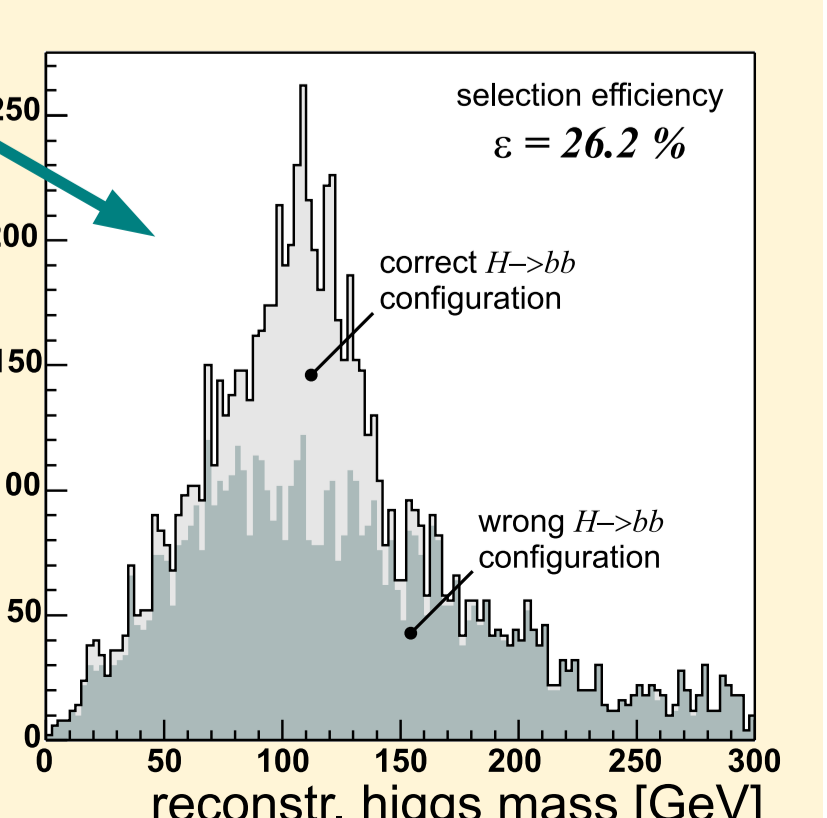
- relations between objects (e.g. decay trees, event generator information)
- combinatorial multiplicities
- general, experiment independent physics algorithms



generator level, (no ISR, no FSR)



full detector simulation and reconstruction



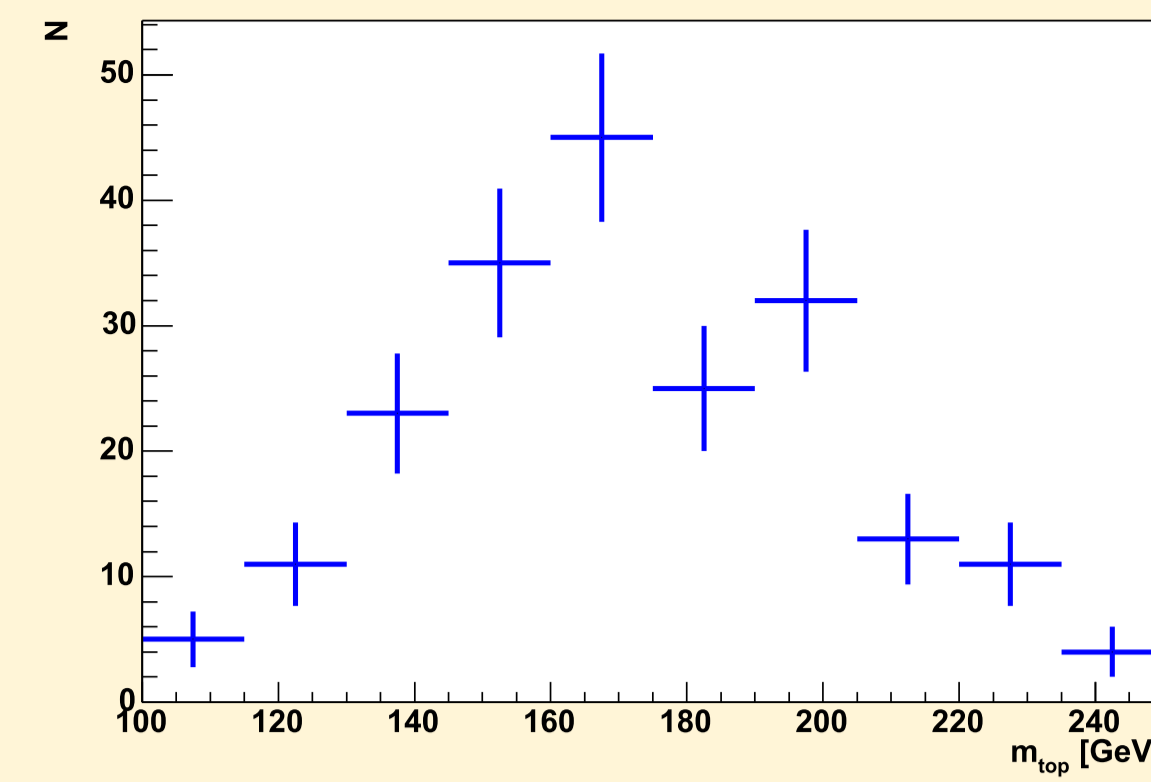
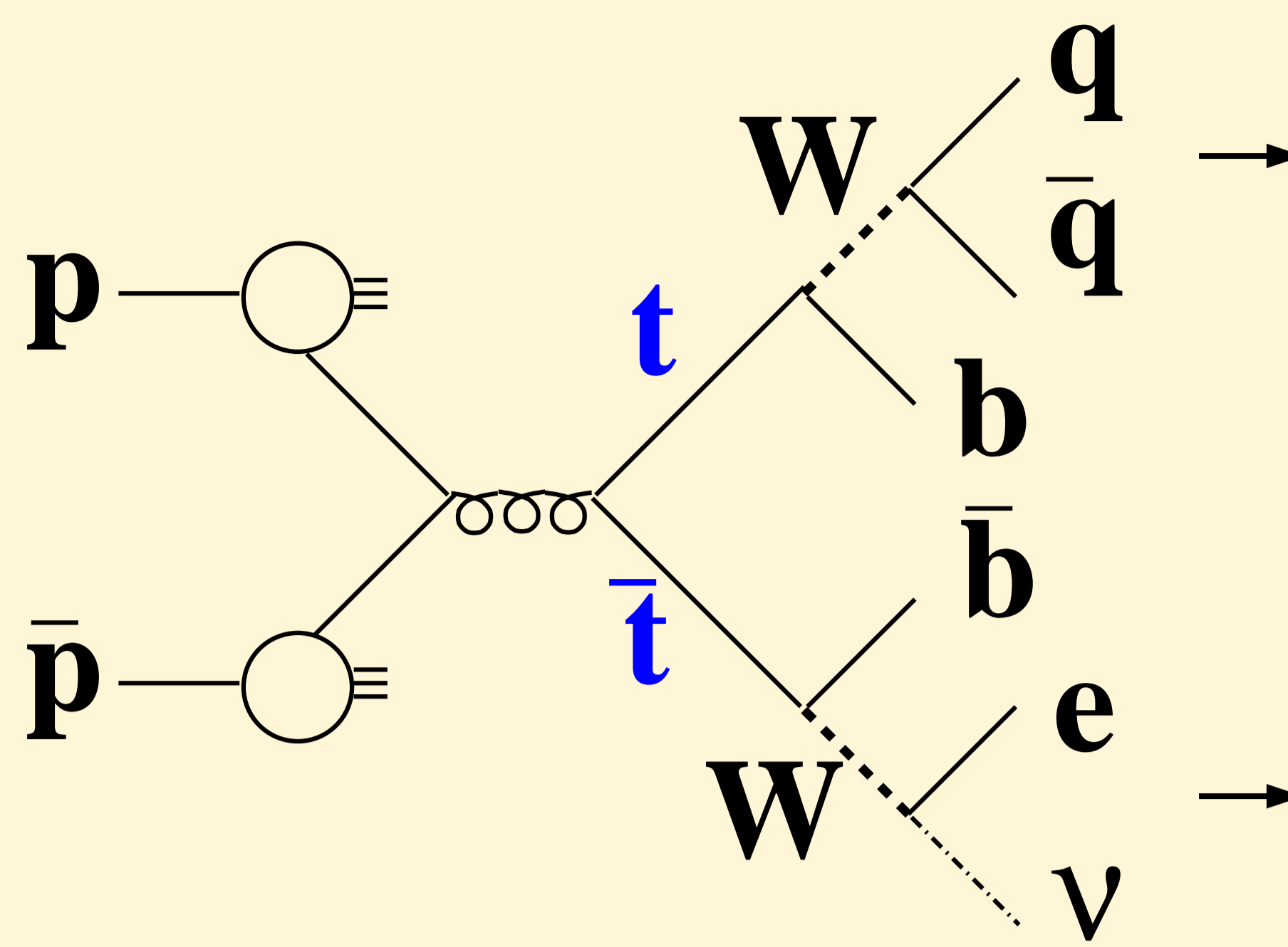
Abstract layer facilitates the study of detector effects on complex analyses

Example Application: $t\bar{t}$ Production at Tevatron

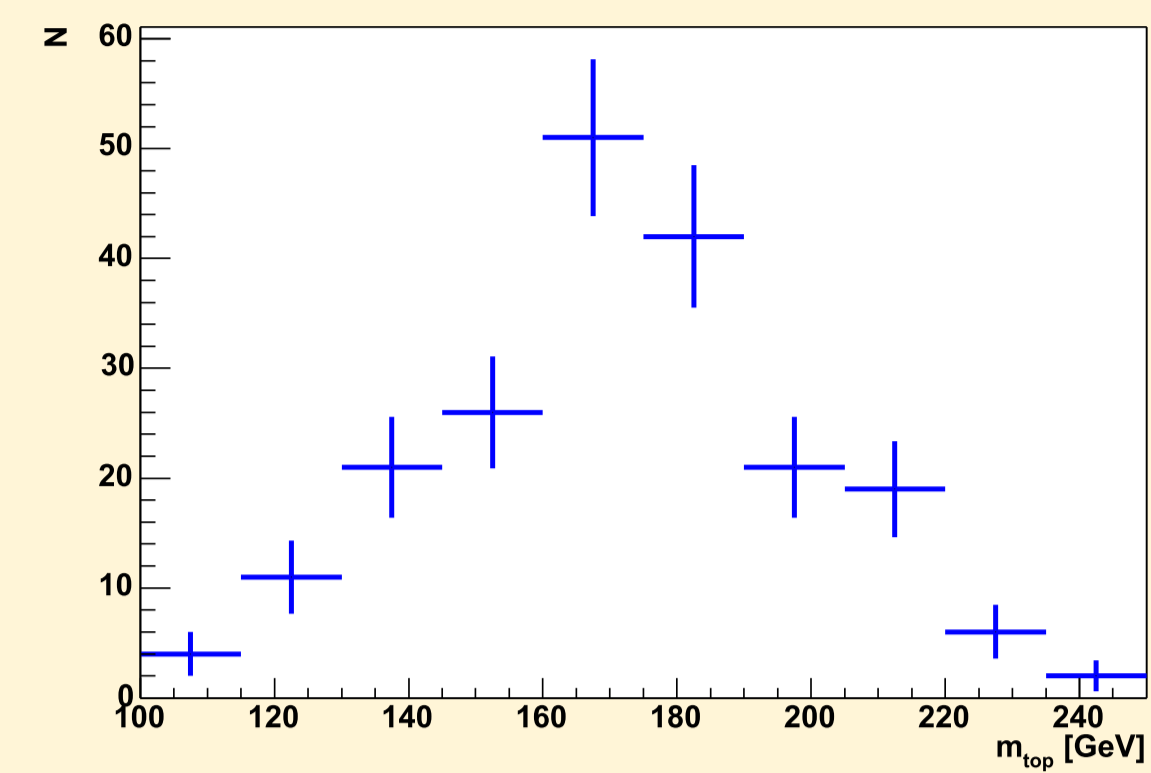
- Complete reconstruction of partonic decay chain
- Combinatorial challenge (see Example 1)

Simulation of the full data analysis chain includes:

- event selection with isolated lepton and four jets at least (one of them b-tagged)
- decision for “best version” of the possible decay trees
- top masses for the hadronic and leptonic W-decays



top mass from three jets after full detector simulation

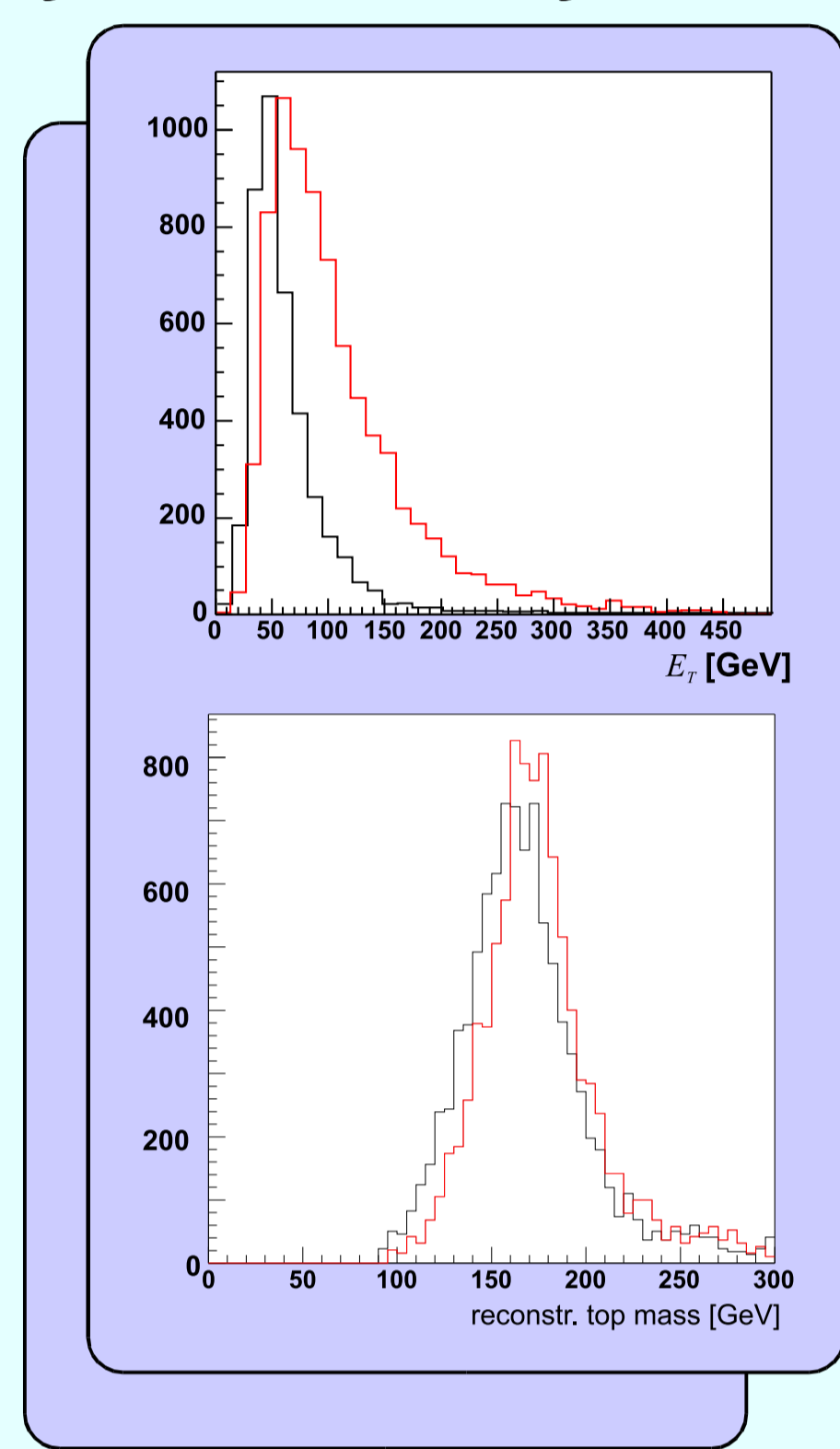


top mass from two leptons and one b-jet after full detector simulation

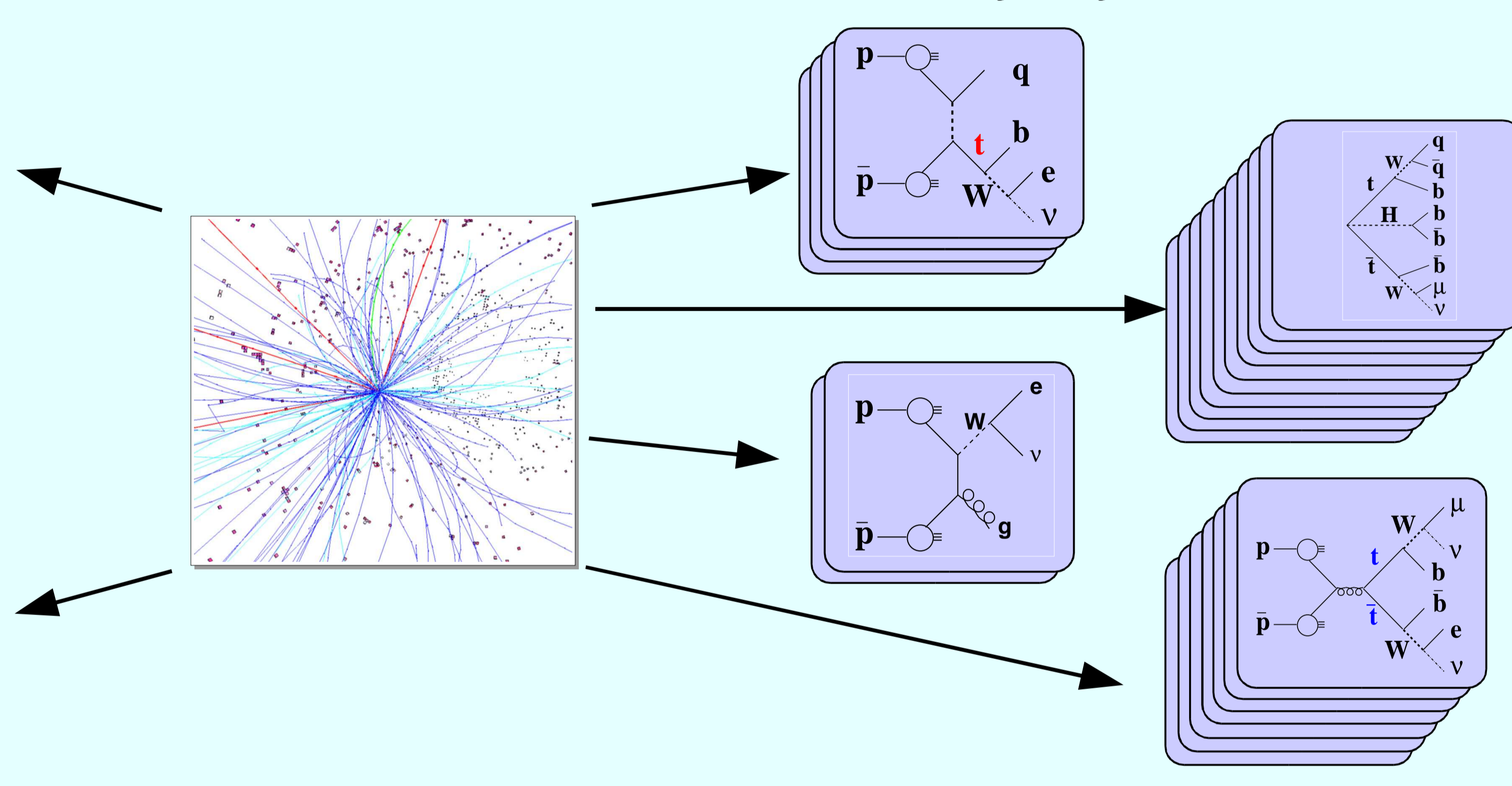
Many Tasks – One Job:

- Many interpretations of the same event can be handled in parallel:
- Determination of systematic errors caused, e.g., by uncertainty in calorimeter energy scale
- Event classification using an “Analysis Factory”

Study Detector Systematics



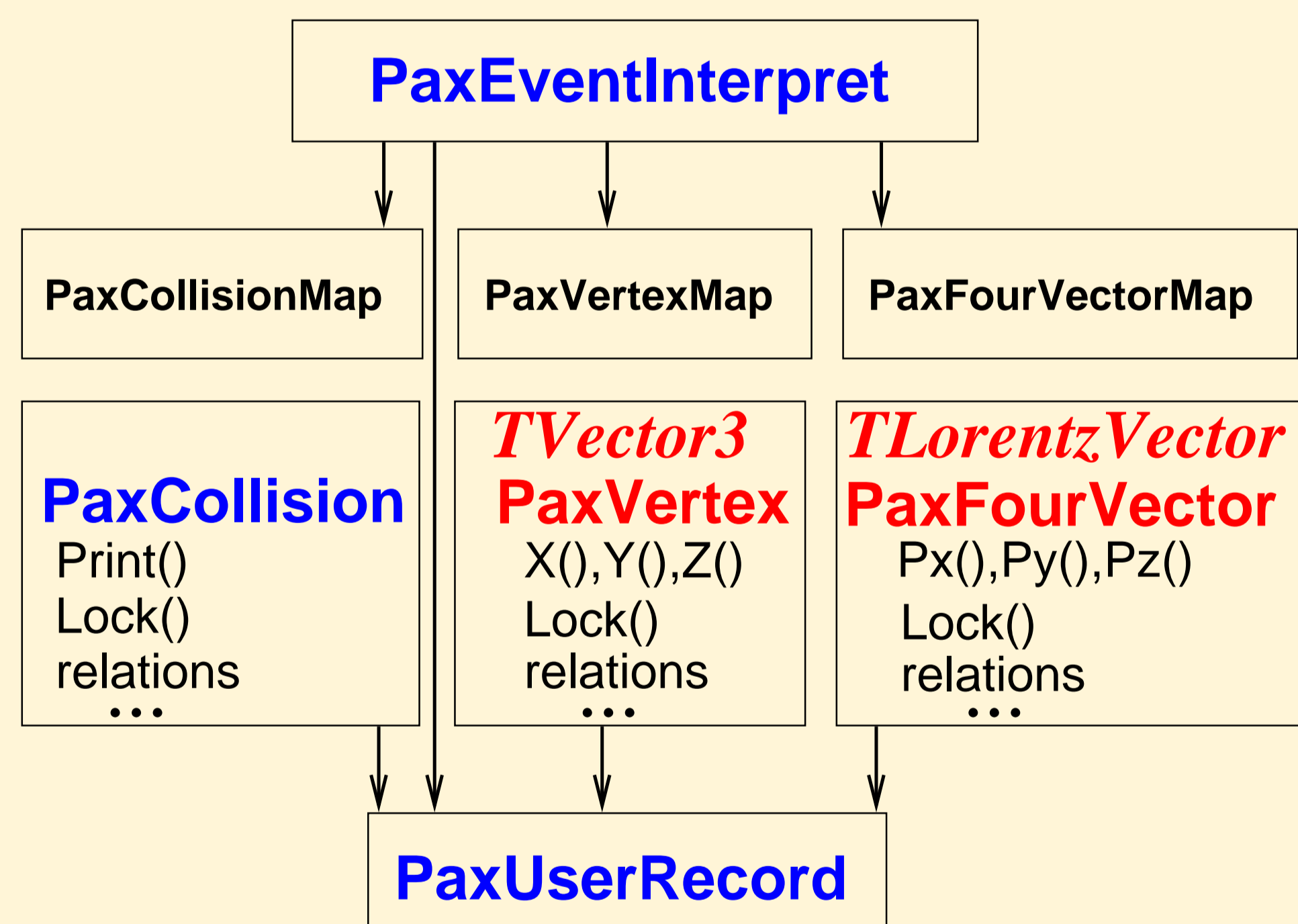
Identify Physics Process



Design Details:

The *PaxEventInterpret* Class

- Is a general container for HEP event interpretations
- Contains the physics objects like fourvectors and vertices
- Has functionality to deal with complex events
- Is persistent including all its objects and relations
- “deep copy” duplicates all objects and sets up the relations correctly



The *PaxFourVector* Class

ROOT: *TLorentzVector*
 • 4 components Px(), Py(), Pz(), E()
 • algorithms (rotation, Lorentz-boost, etc.)

OR

CLHEP: *HepLorentzVector*
 • 4 components px(), py(), pz(), e()
 • algorithms (rotation, Lorentz-boost, etc.)

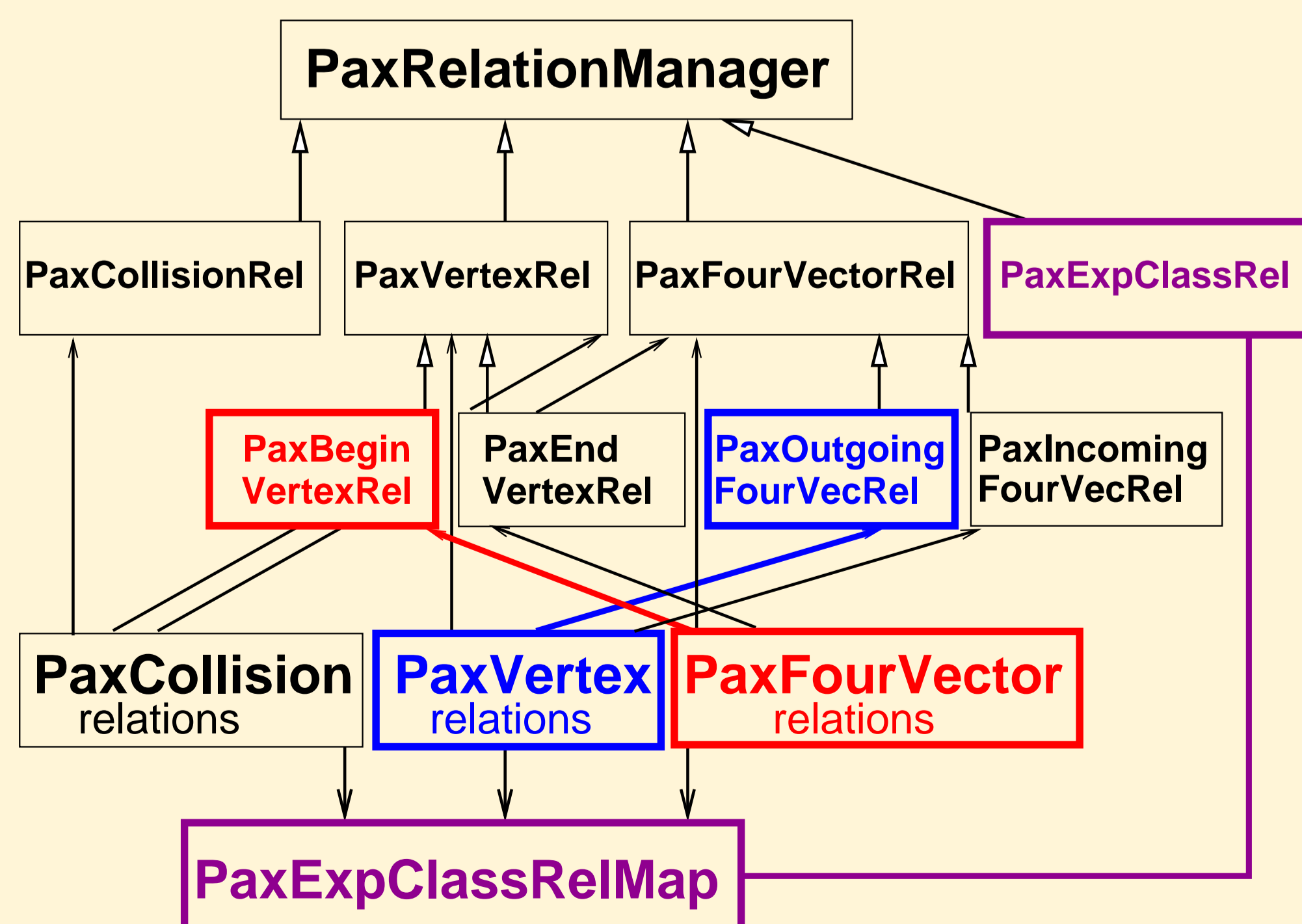
INHERITANCE

PaxFourVector: (is designed to access every possible information in the reconstruction output)

- `getCharge()`,
- `getParticleId()`,
- `lock()`: the decay tree can be locked and excluded from analysis
- `begin_vertex_relations`, `end_vertex_relations`: to establish decay trees and histories
- `user_record`: store additional information as string-double pairs
- `experiment_relations`: associate any pointer to original detector object with the *PaxFourVector*

The *PaxRelationManager* Class

- Takes care of “many-to-many” relations of fourvectors, vertices, ...
- Establishes decay trees and analysis histories
- Locking mechanism to exclude parts of trees from analysis
- Experiment relations keep pointer to original detector instance
- Based on “Mediator” design pattern



Interface to data sources

