Comments on Olivier's physics event model

I think it would be nicer if all reconstructed particles, simple and composite, are of the same base type, eg. ReconstructedParticle.
Why must there be 2 types, ProtoPart and Particle for Brunel and DaVinci?
Surely Brunel will produce some composite particles, eg. V^0, π^0.

Similarly, I think all vertices, including primary, should be of the same base class, ReconstructeVertex.

In Olivier's design the identity of a ReconstructedParticle is given by its concrete class type.
I think the identity of a ReconstructedParticle should be given by a data member of the class and not by the concrete class type.
The identity of a ReconstructedParticle may not be defined, and when defined it should be easy to change, eg. after application of an alternative ParticleID algorithm.

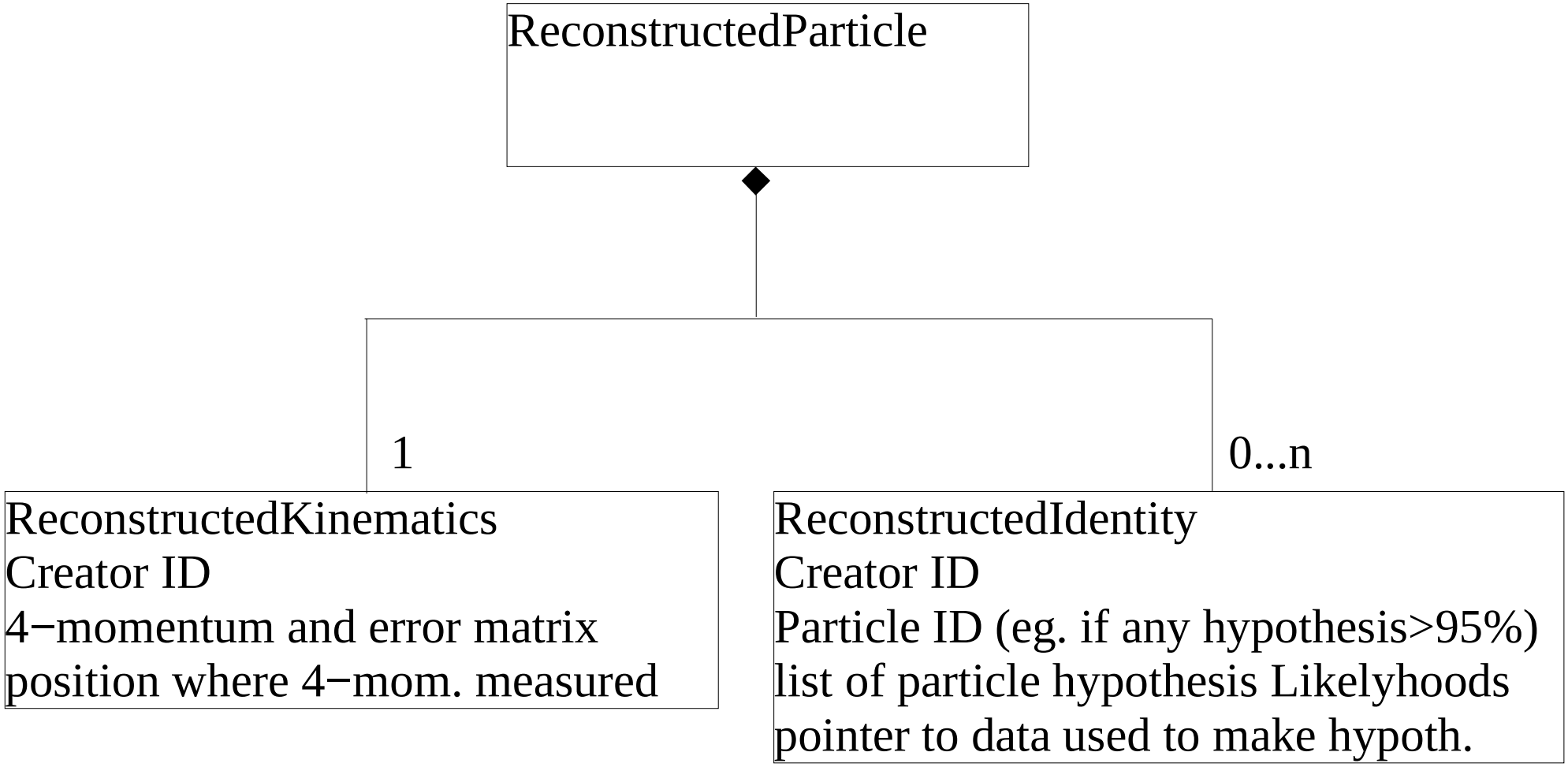
I think the kinematics and the identity of a ReconstructedParticle should be independent, and implemented as two contained objects, eg of abstract type ReconstructedKinematics and ReconstructedIdentity.

Naming

The MC produces MCParticle and MCVertex objects.

Therefore, I think it would be nice if the Reconstruction (Brunel and DaVinci) produced ReconstructedParticle and ReconstructedVertex objects.

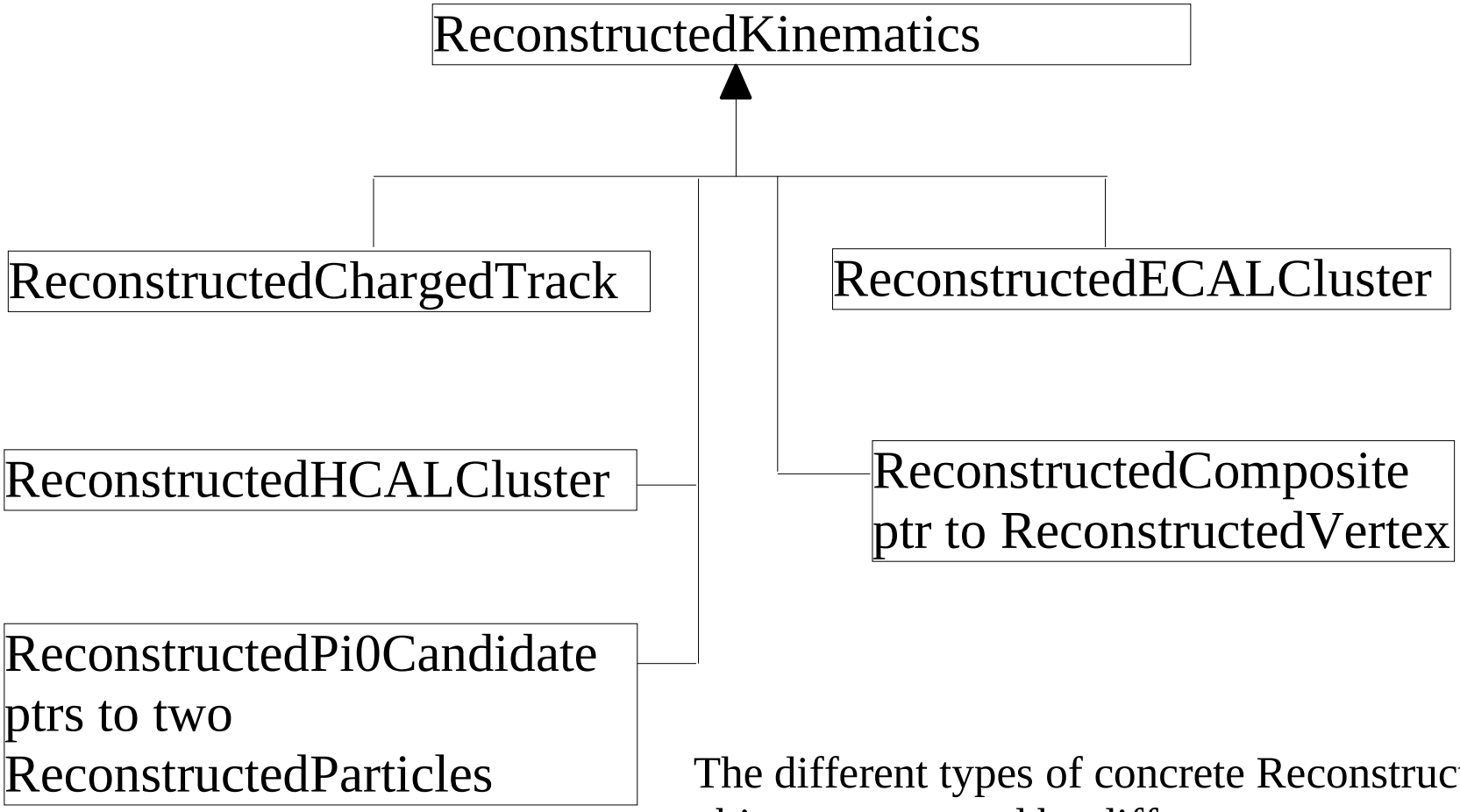
Top Level Class Diagram



Main design ideas : to seperate the Kinematics from the Identity
to encapsulate the Kinematics and the Identity into 2 classes
the following slides describe one possible implementation

Each ReconstructedParticle can have 0, 1 or more reconstructed identity objects from different ParticleID algorithms.

ReconstructedKinematics Class Inheritance structure

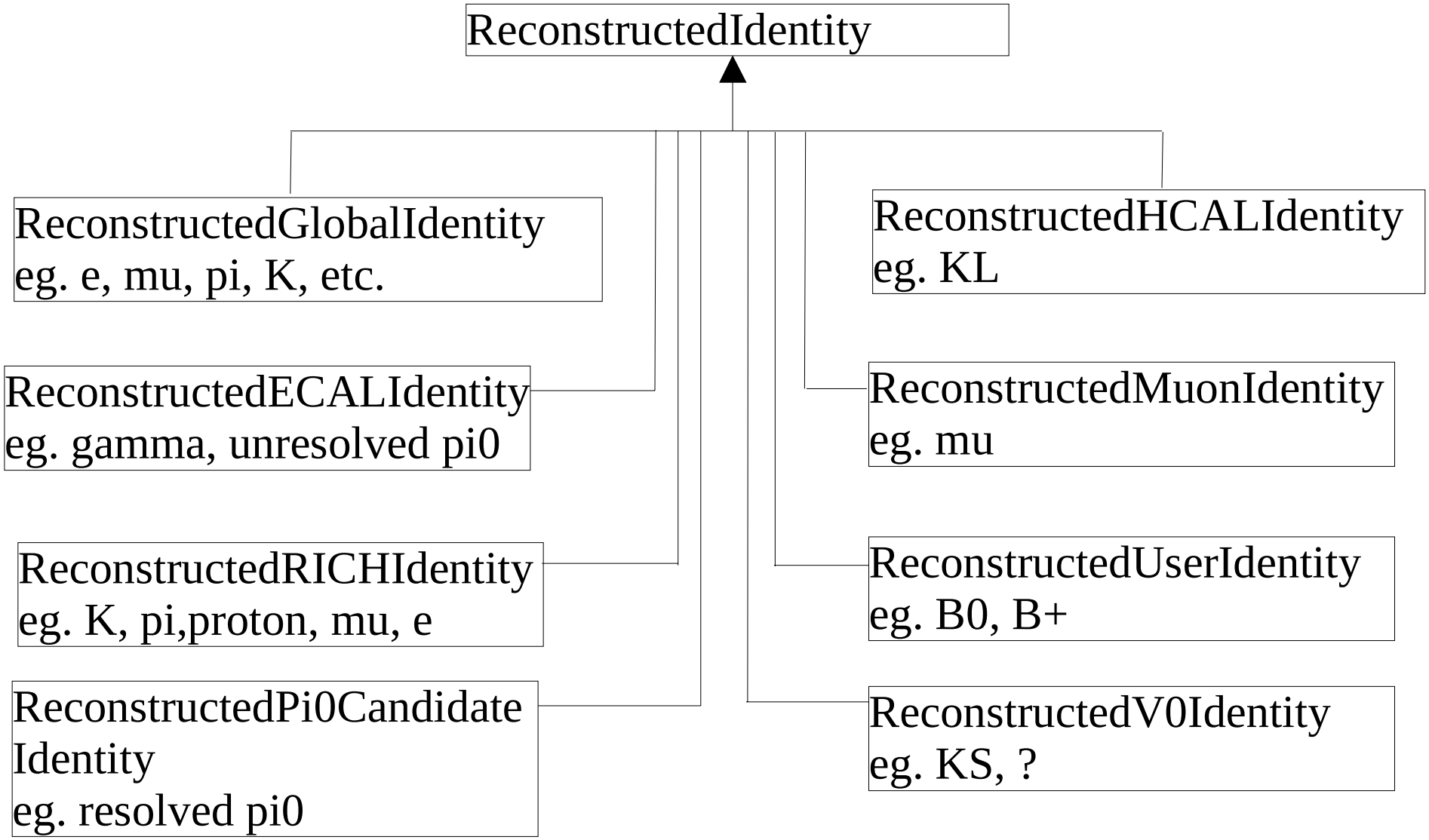


The different types of concrete ReconstructedKinematic objects are created by different reconstruction algorithms either in Brunel or DaVinci.

A ReconstructedComposite object could be created by a V0 finder in Brunel, or user analysis code in DaVinci (eg. B0 to pipi).

The kinematical variables (E, p) could be implemented as methods which take the hypothesized id (within the ReconstructedIdentity) as input.

ReconstructedIdentity Class Inheritance structure



The different types of concrete ReconstructedIdentity objects are created by different particle ID algorithms either in Brunel or DaVinci. A ReconstructedParticle can have 0...n different ReconstructedIdentities.
The first should be of type ReconstructedGlobalIdentity if it exists.

eg. Object Diagram for B to pi+ KL (only concrete instantiated objects)