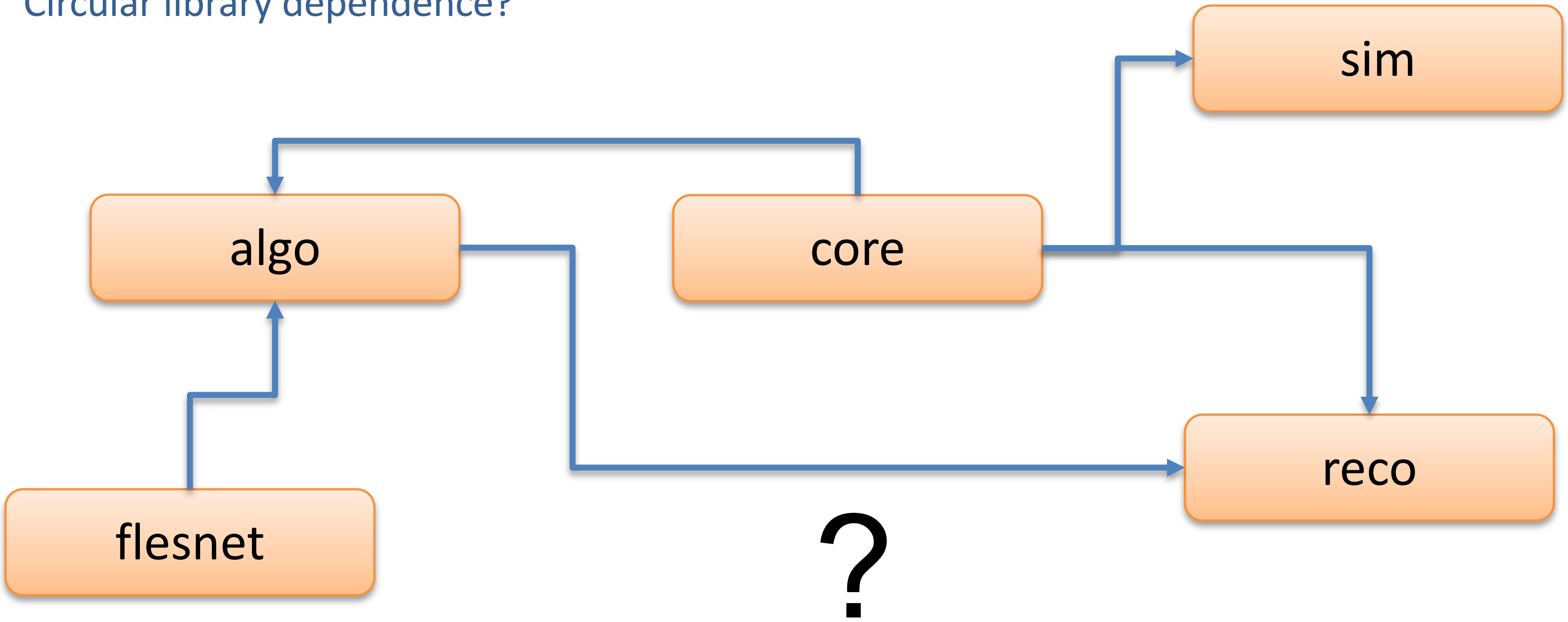
How to make use of utilities in algo?

- E.g., EnumDict or Yaml-class interface
- Requires linking of core to algo
- Circular library dependence?



sim

algo

core

reco

flesnet

?

Enum initialisation

- I have an enum class as member of some class. I have to member-initialise it.
- I do not want to choose a default value. What to do?
- My (clumsy) approach: Define an enum value Undefined and initialise with that.
- Is std::optional the better way?
  - If (myEnum == Undefined) throw....
  - If (! myEnum) throw...

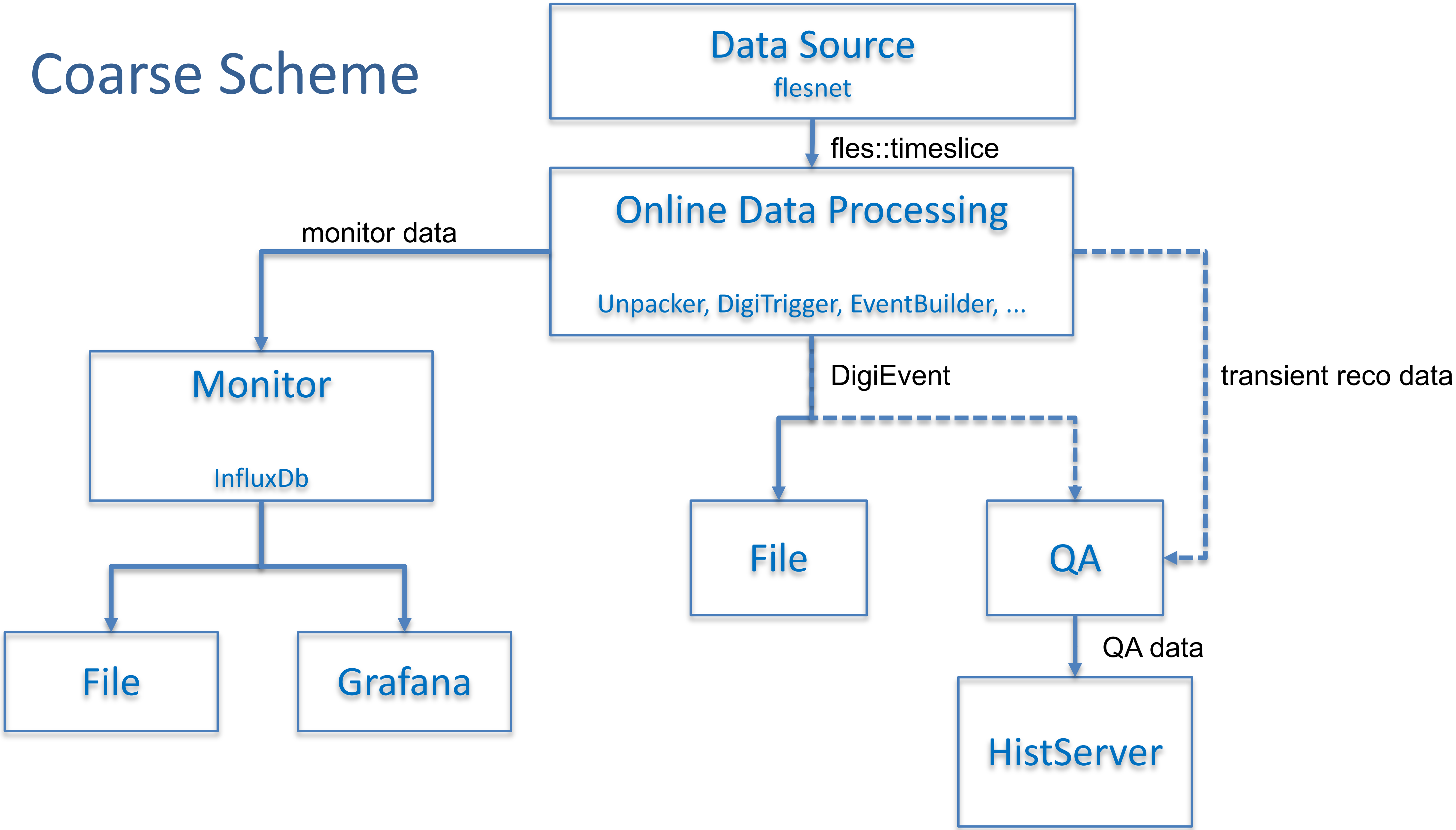
Enum names

- By convention, our enums start with ‘E’, e.g. ECbmModuleId
- My personal taste is not to include an indicator of the variable type into the variable name.
- Might by different for enums?
- Or should we drop the convention?

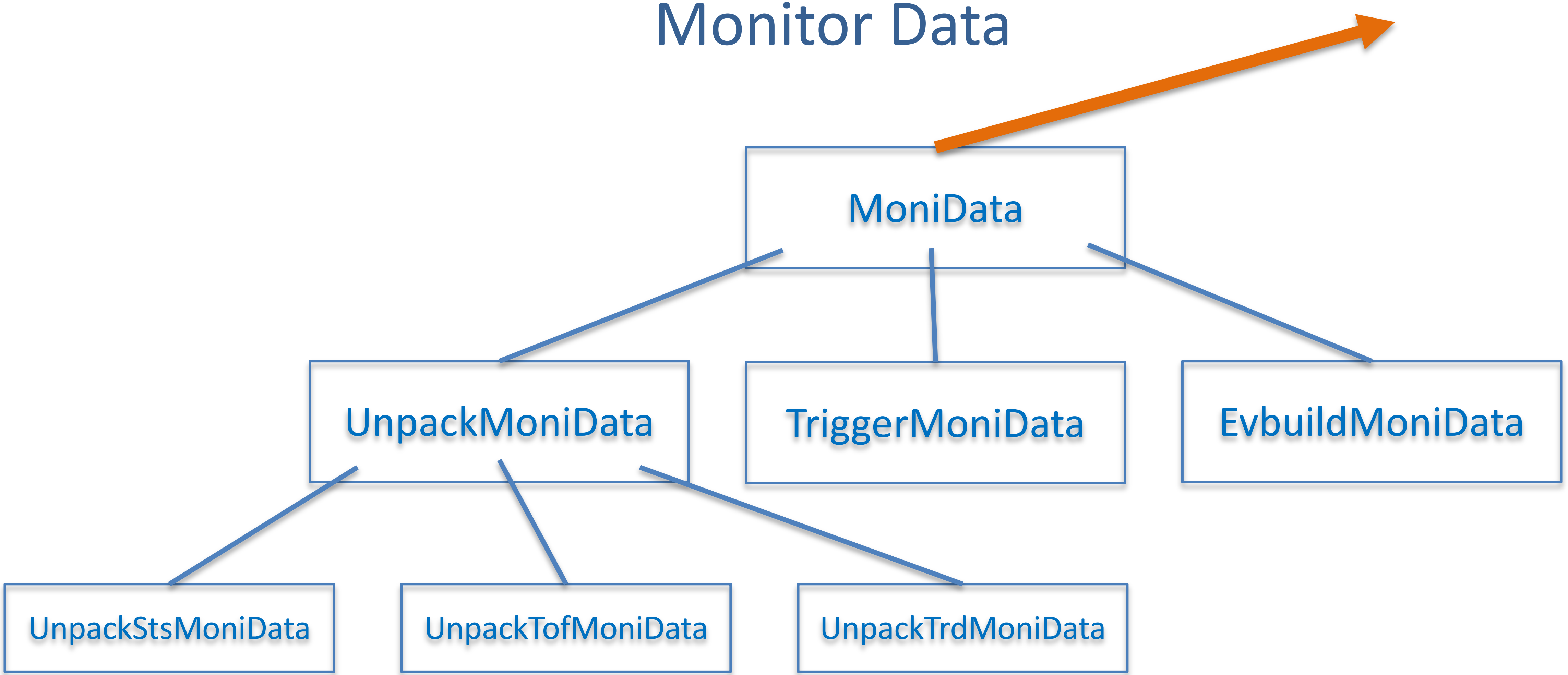
Logging

- In algo, we use fles_logger instead of FairLogger.
- Should we do so in the “offline” code, too?
- If so: must we change the entire code base at once, or can both loggers live side-by-side?

Coarse Scheme



Monitor Data



MoniData

UnpackMoniData

TriggerMoniData

EvbuildMoniData

UnpackStsMoniData

UnpackTofMoniData

UnpackTrdMoniData