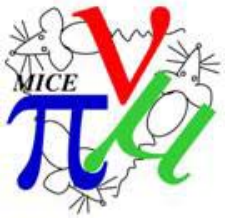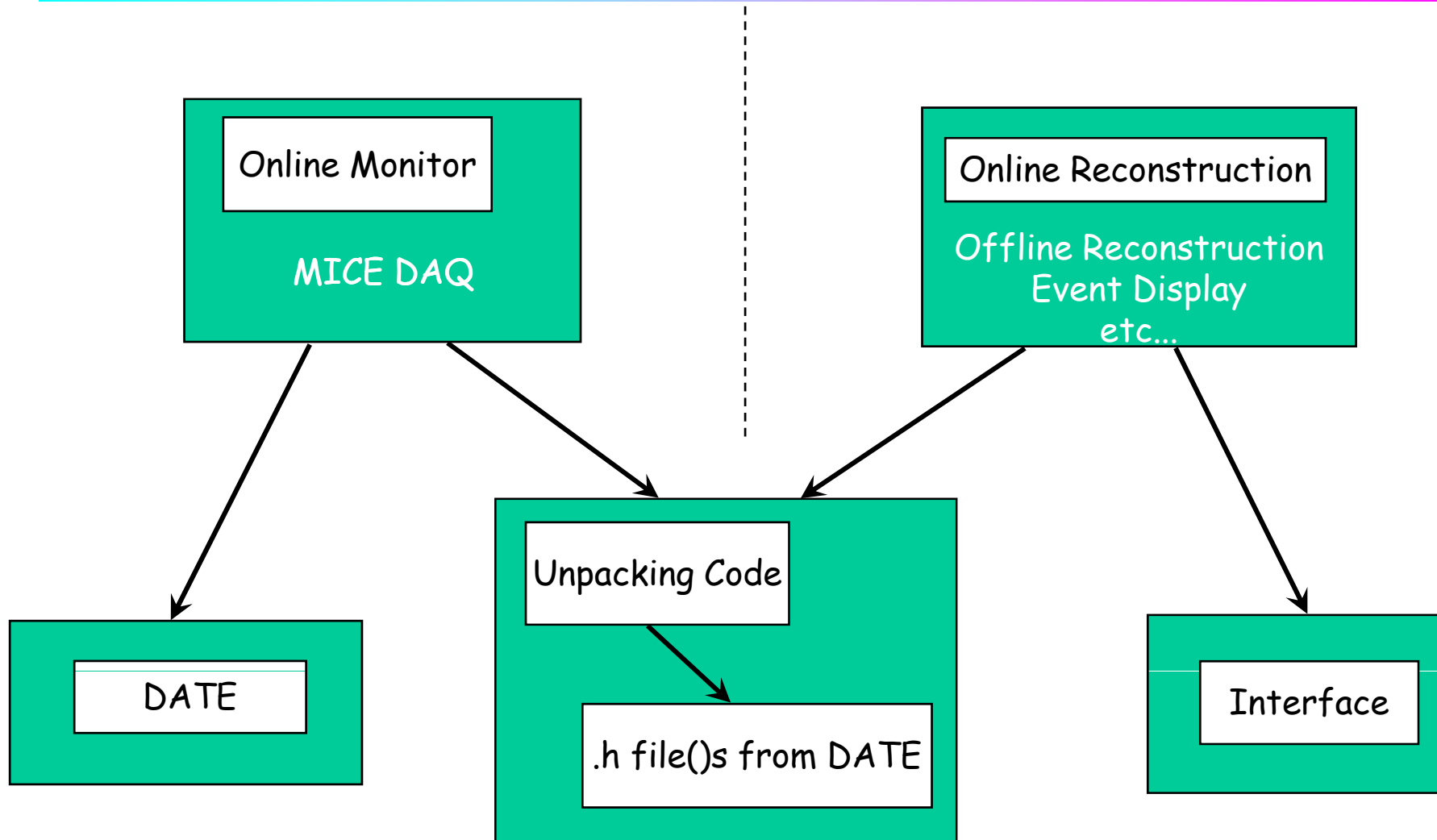# Unpacking Code in G4MICE

- Ben, Michael and I are starting to work on this now (will continue once I'm back in Chicago next week).

- Basic philosophy (as discussed yesterday) is to write a section of code that depends on the bare minimum amount of DATE (preferably one or two .h files and no libraries, purely typedefs and constants) and which has no dependence on any G4MICE classes.

- This code reads the DATE format file and returns the ADC, TDC, etc information (through an interface yet to be defined).

- The Monitoring code would then call these routines and produce histograms, etc while in G4MICE a class in the RealData domain would call the same routines and use the returned values to instantiate the G4MICE specific C++ classes used to represent the information.
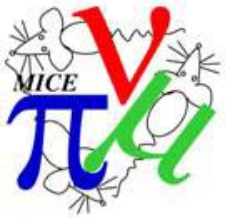
G4MICE and DATE

Online Monitor

MICE DAQ

Online Reconstruction

Offline Reconstruction
Event Display
etc...

Unpacking Code

DATE

.h file()s from DATE

Interface

In MICEDAQ CVS Area

?

In G4MICE CVS Area

2

Hit Representations in G4MICE

- At the moment we have 3 classes (and one abstract class) that have been used to date for tracker data taking.

- This is almost sufficient for MICE, but we will also need a class (or classes) to represent the information from the flash ADCs.

- The "RealData" part of G4MICE has code dedicated to reading in raw data files (in various different formats) and making instances of the relevant C++ classes containing the information extracted from these files.

- New code will do the same for the DATE format.

- Once these classes have been instantiated, the rest of the G4MICE code does not know or care about the data format that produced the hits, the type of DAQ system, etc. (So the same code that we use now to analyse the Tracker test at KEK, the cosmic ray data at Fermilab, Tracker QA at Imperial, etc can be used for the cosmic test of the tracker in R12 and finally real MICE data).
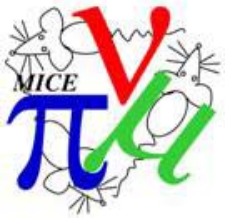
Hit Classes

- **VlpcHit** – this is one channel from the AFEIIt (or AFEII) board corresponding to the digitised output from one VLPC channel. The hit contains information about the AFE board, MCM, channel, ADC value, TDC value and discriminator status.

- **VmeBaseHit** – an abstract class that is common to the two classes below. It contains information about the VME crate, VME module and channel numbers.

- **VmeAdcHit** – inherits from the VmeBaseHit and adds the ADC information.

- **VmeTdcHit** – inherits from the VmeBaseHit and adds the TDC information.

- We will need to implement a VmeFlashAdcHit class as well.