LArSoft DB interface integration

Erica Snider
Fermilab

May 7, 2015

Database interfaces*

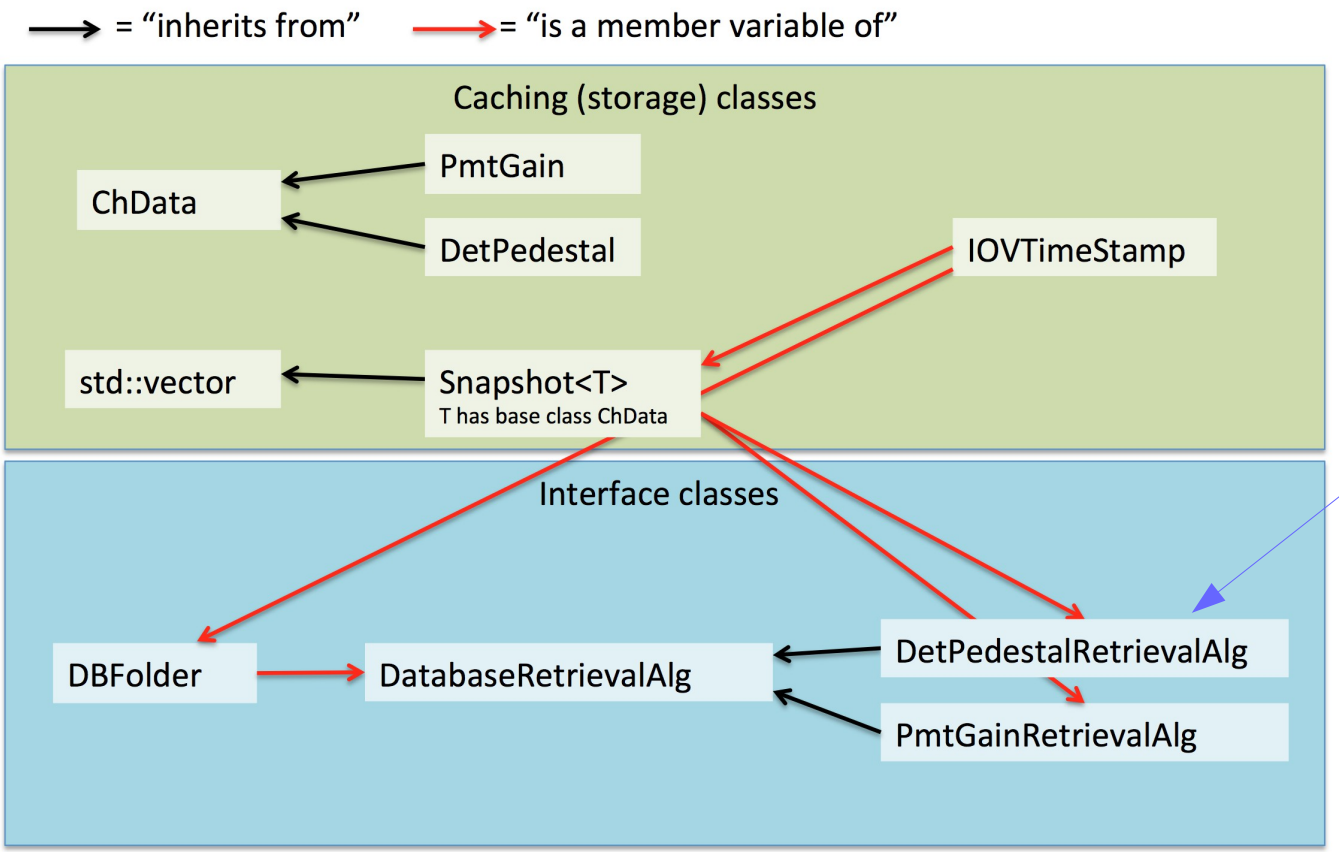
* Talking in terms of “database interfaces”, but solution should support multiple data sources for each type of data: text files, fcl parameters, multiple databases, etc, all configurable via fcl

- Two questions to discuss and agree on
  - Where to put the abstraction layer and what should it look like?
  - What data needs to have this treatment*, and what is the reconstruction-level interface to it?

    Having both answers should allow us complete the end-to-end code that will work for all experiments

- Where to put the abstraction layer?
  - Need one common interface for each set of data that can come from a DB.
    - Naturally have one where LArSoft interfaces to the actual data values

From Brandon's DB interface Coord. Meeting talk on 4/21



The reconstruction sees only DetPedestalRetrievalAlg, PmtGainRetrievalAlg, etc.

These must therefore be common interfaces.

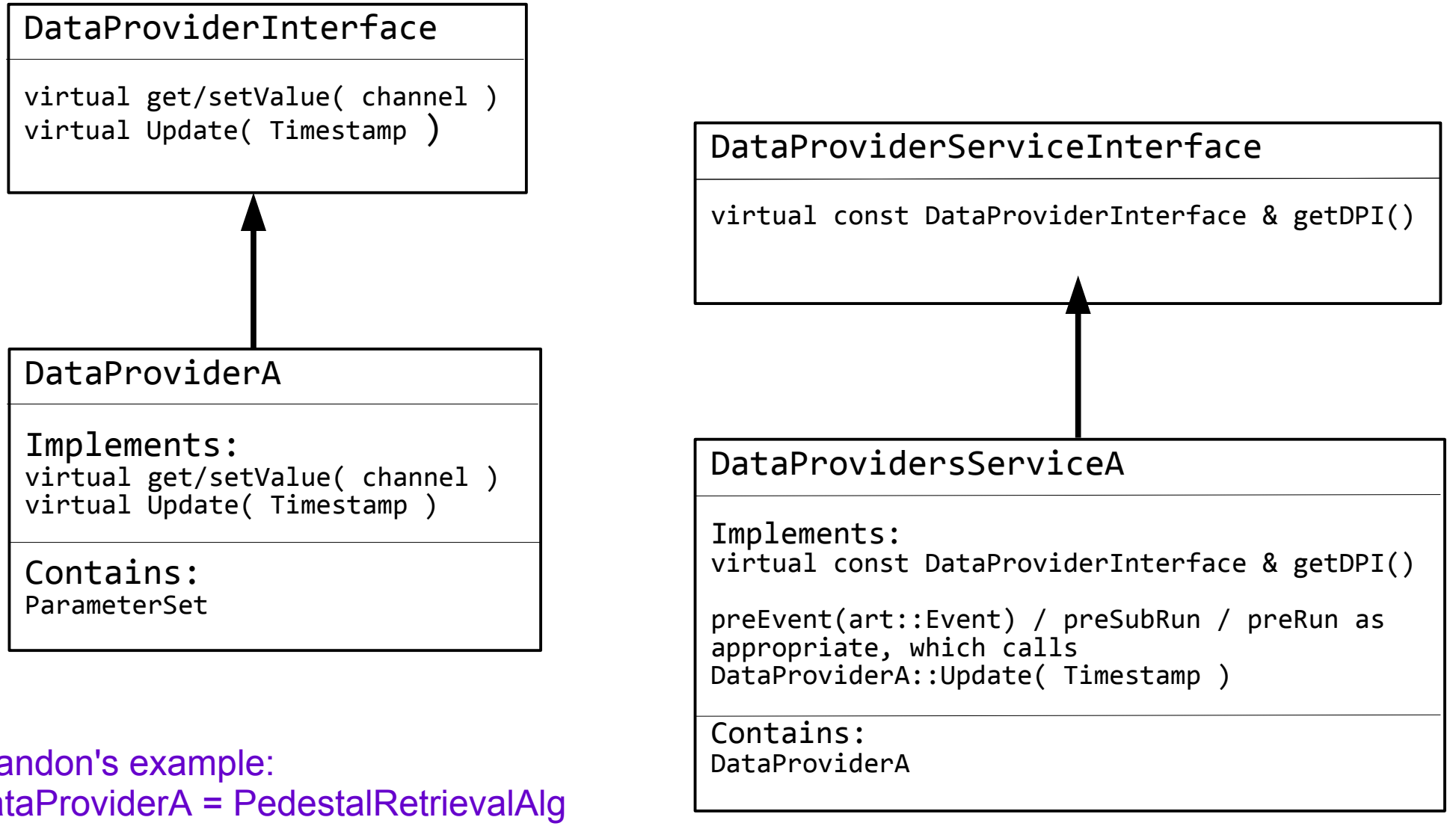
Will refer to them generically as "data provider"

Want only one class actually accessing the DB (or other sources) and managing data, so also need an art::Service class with a common interface between the DB and the data provider (DetPedestalRetrievalAlg in this case).

Service design

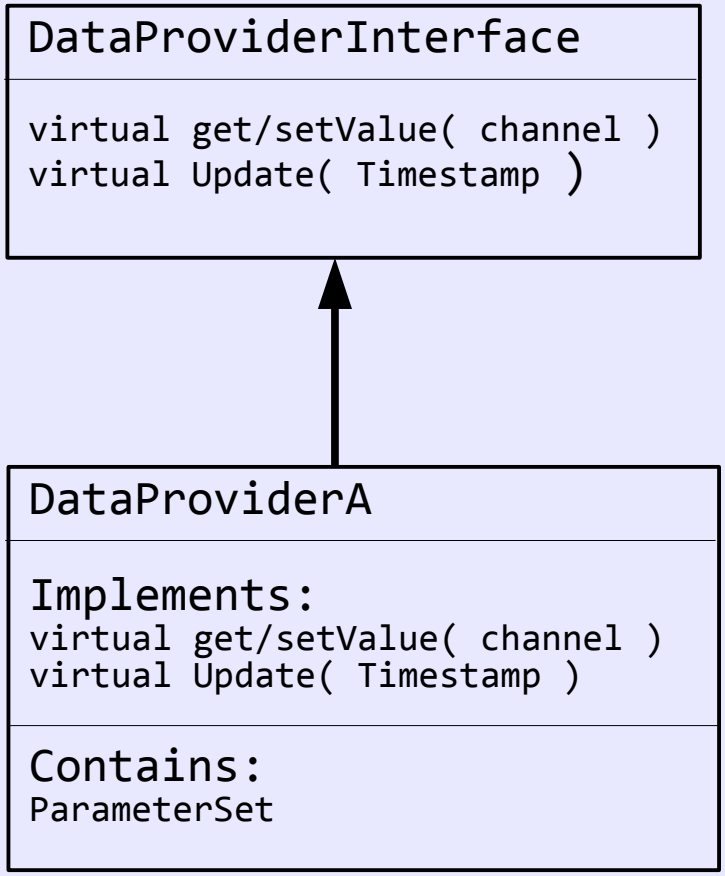
- Placing the service at the data provider level allows the weakest coupling to the code that retrieves and manages data from the DB
  - I.e, the service either contains or derives from the data provider
  - This scheme gives the experiments the greatest flexibility
    - Some trade-offs in performance, so need to watch that
  - Requires one new service for each data provider
- Design recommendations
  - Separate the framework interface from the data provider if possible
  - Service class should contain the class that implements the data provider
  - Perform global IOV updates on per-event, per sub-run, or per run IOV basis
    - Avoids checking the IOV on a per-channel basis for channel-oriented data

For example:



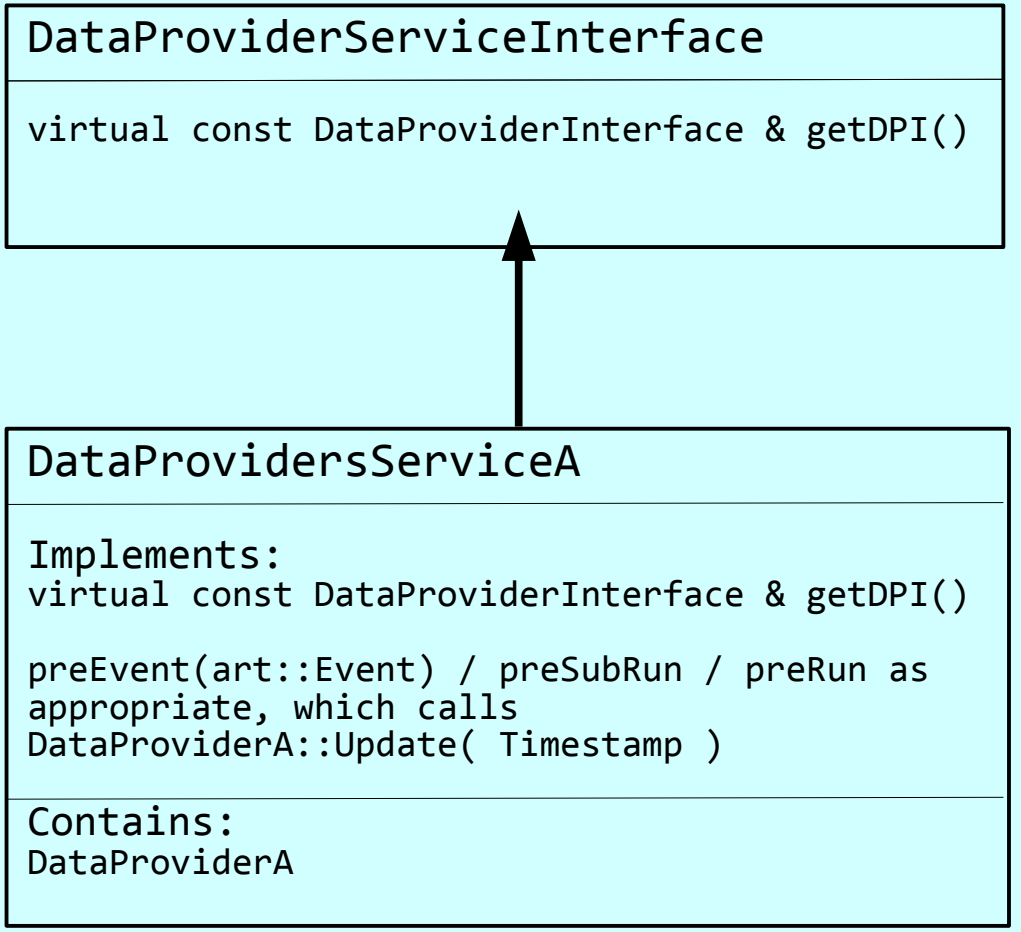
In Brandon's example:
DataProviderA = PedestalRetrievalAlg
DataProviderInterface = new abstract interface for PedestalRetrievalAlg

For example:



These classes are independent of art

These classes depend on and talk to art



What types of data do we need now

- The list:
  - Wire channel pedestals and noise (done?)
  - Wire channel gains
  - PMT channel pedestals and noise
  - PMT channel gains
  - LAr properties (temperature, electron lifetime, etc)
  - Space charge map (?)
  - Alignment data (?)
  - What else?
  - ...
- What are the interfaces to each in the reconstruction?