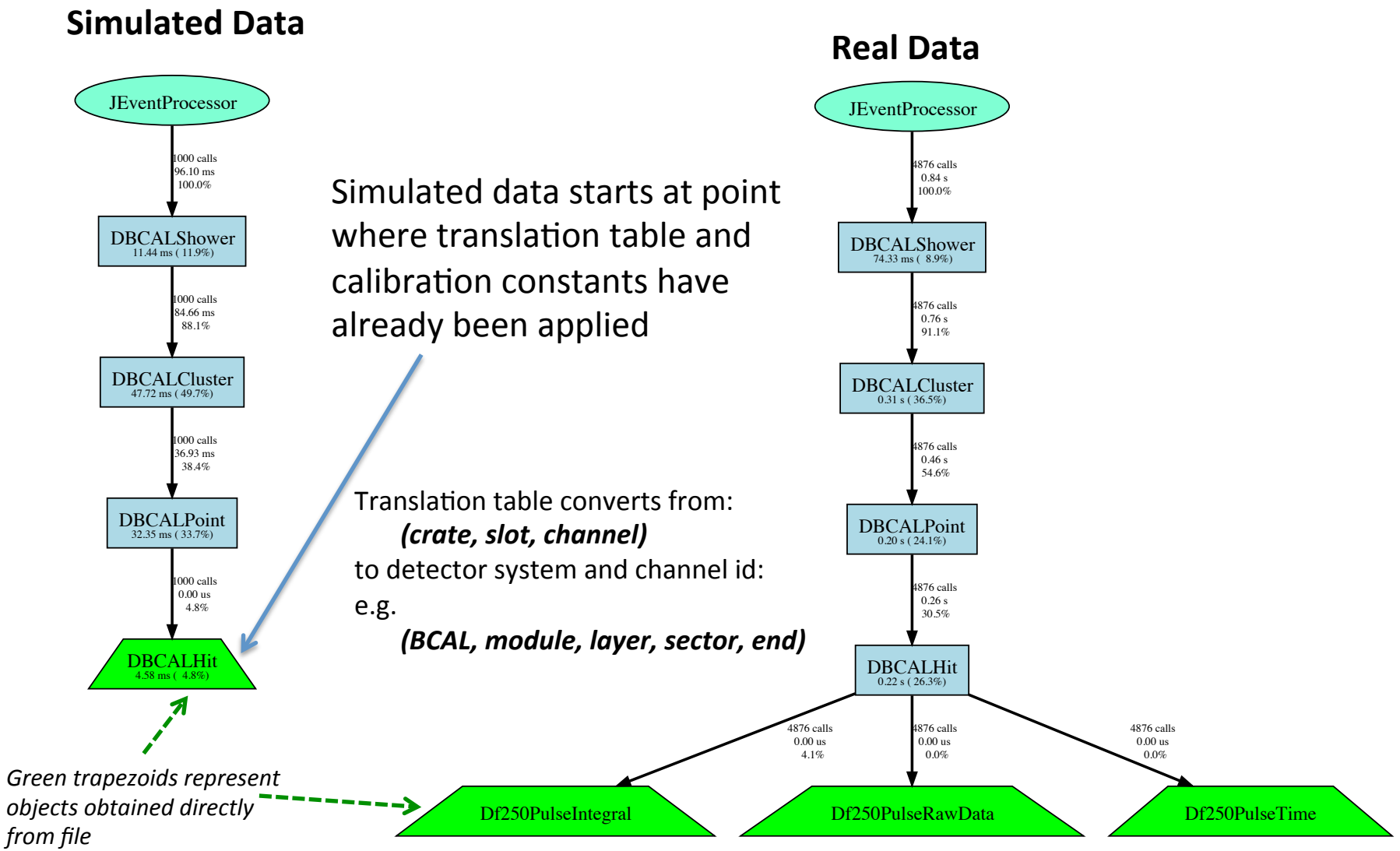
JANA and Raw Data

David Lawrence, JLab

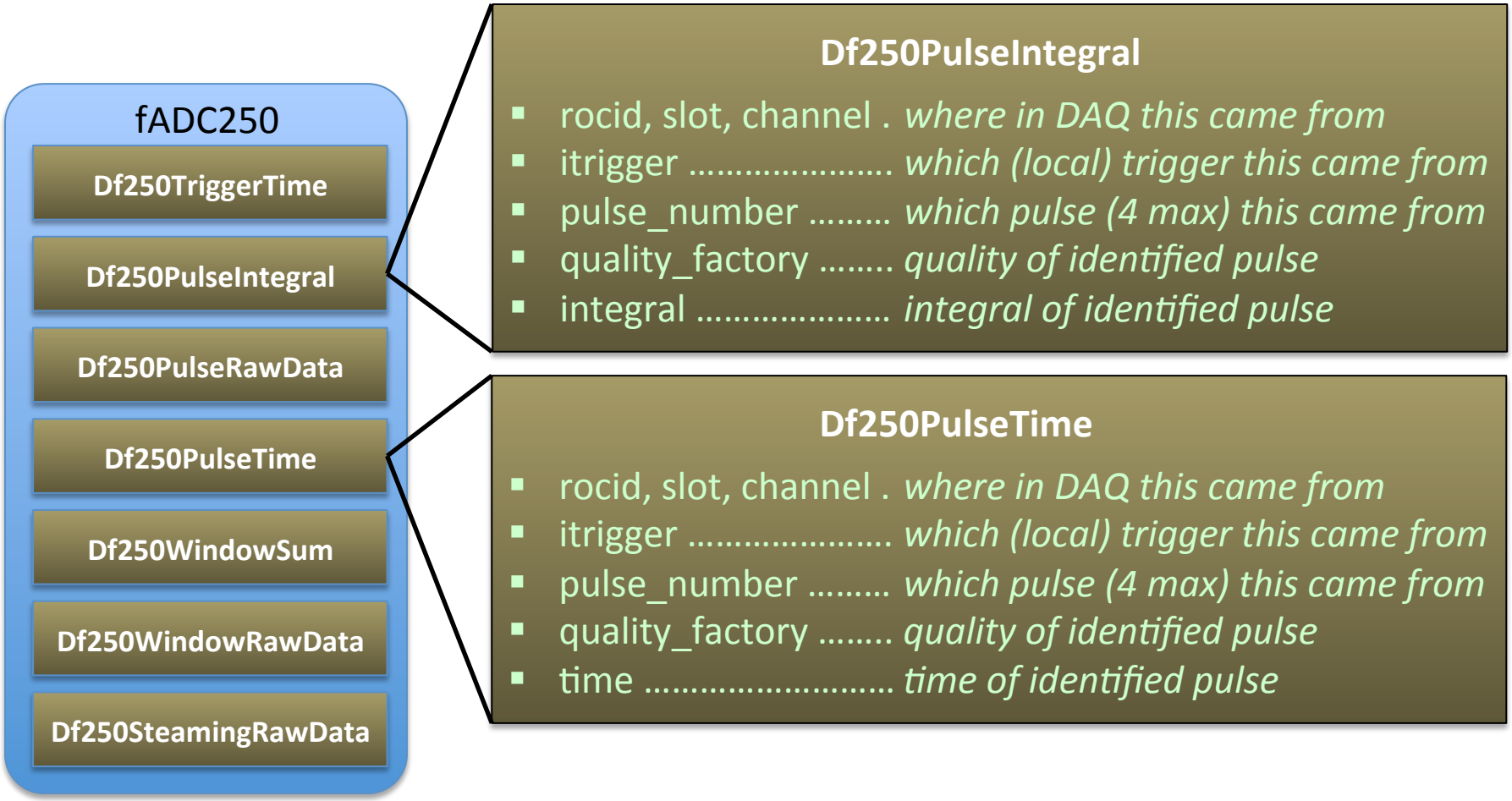
Oct. 5, 2012

Simulated and real data chains



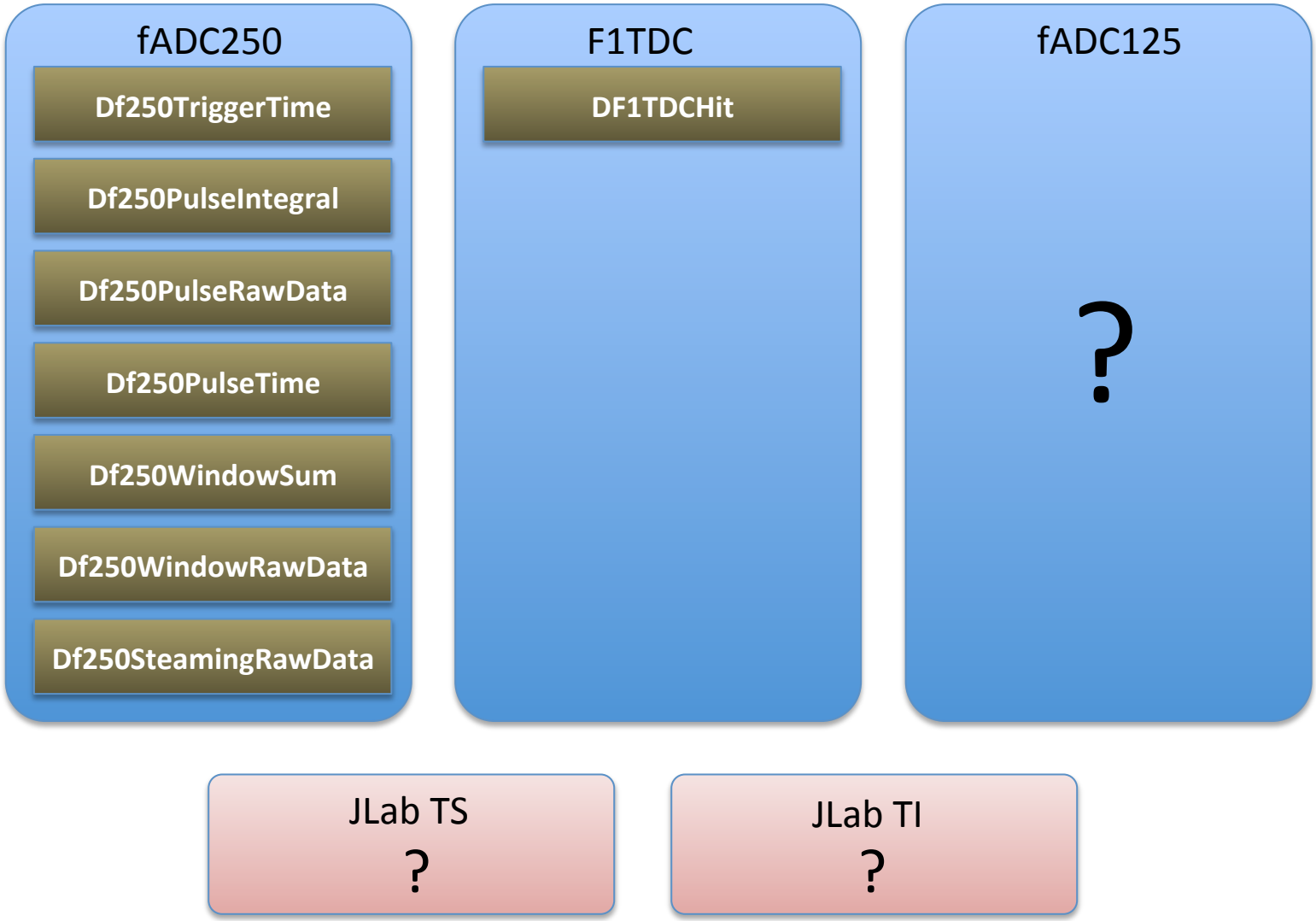
Digitization Module Data Objects

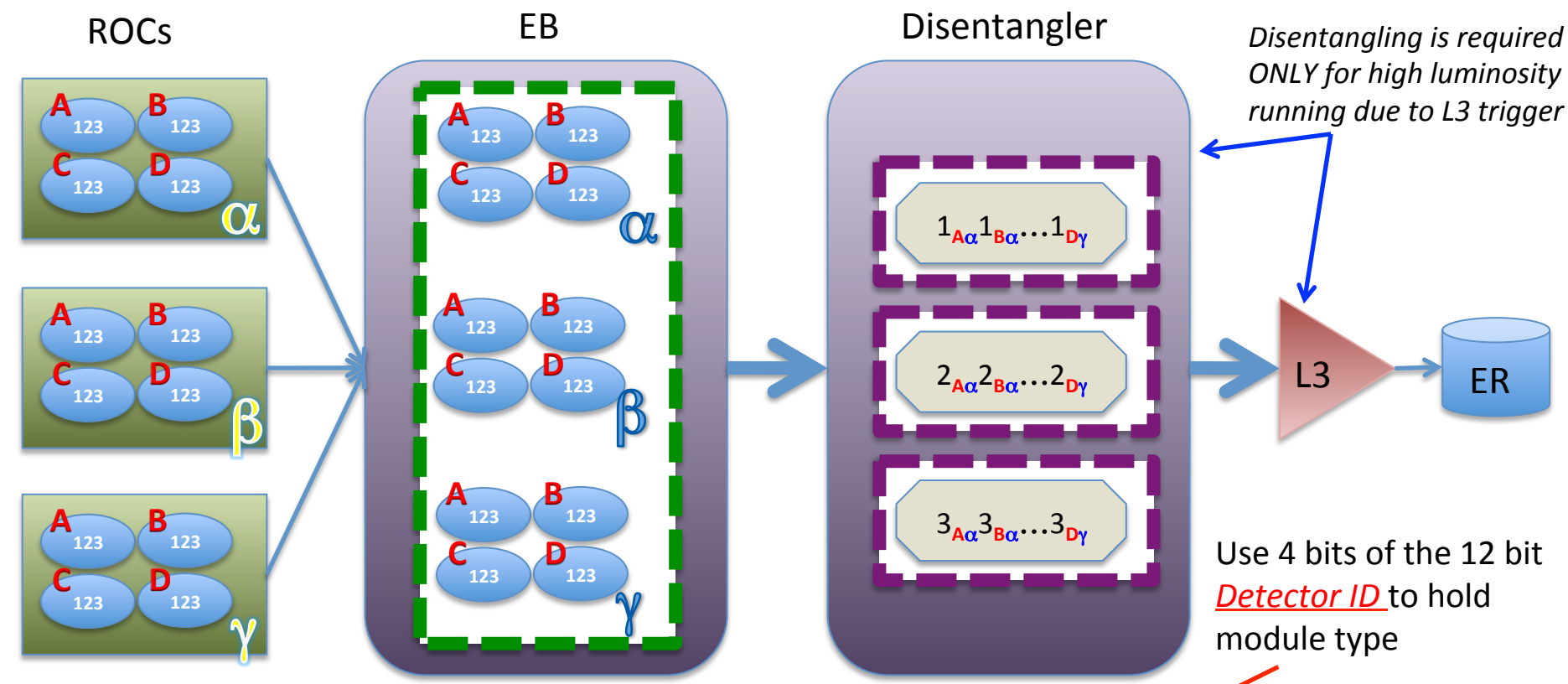
Data objects created at lowest level in JANA reflect the information contained in the data types produced by the digitization modules



Digitization Module Data Objects

List of module types and descriptions maintained in DModuleType.h. Written by human for now, but could be derived from JInventory in future.



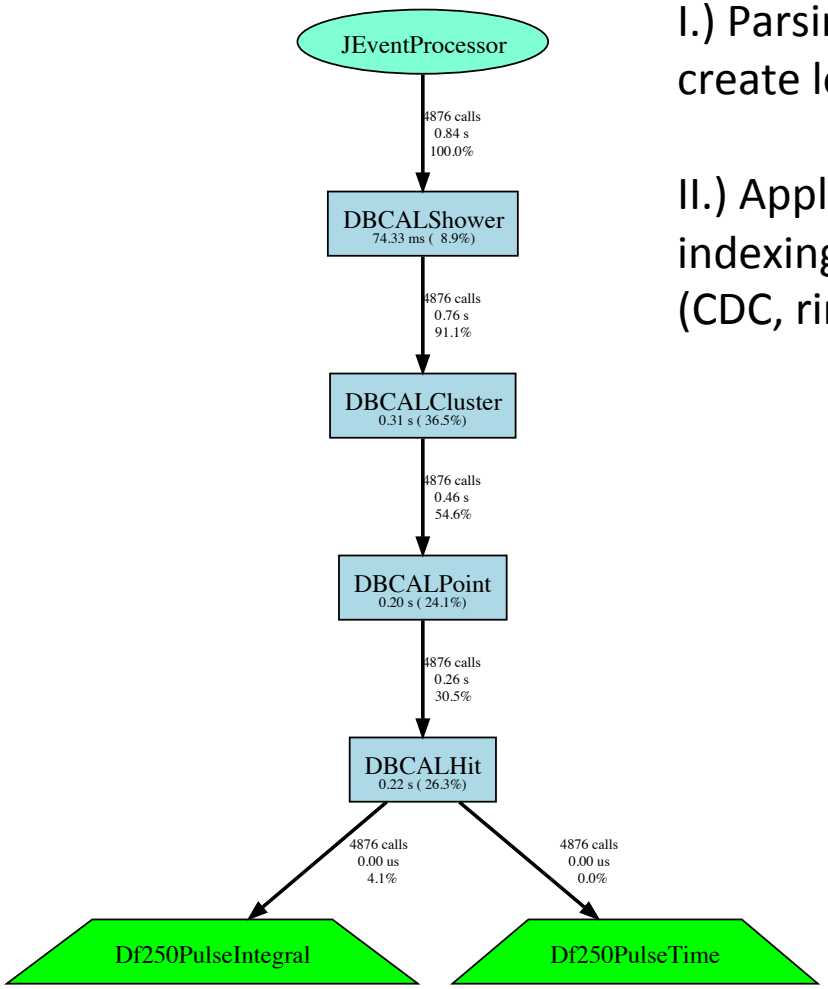


αβγ = crate
ABCD = module
123 = event

size		
status	Detector ID	Num. Events
Num. of first event in block		
Nevents from Module 1		
Nevents from Module 2		
…		

size		
status	Detector ID	Num. Events
Num. of first event in block		
Module 1, event 1		
Module 2, event 1		
…		
Module 1, event 2		
…		

Two phase data interpretation



<u>Interpretation of input data will be done in 2 phases:</u>

I.) Parsing of the data blocks based on module type to create low-level data objects (e.g. Df250PulseIntegral)

II.) Application of translation table to convert from DAQ indexing (rocid, slot, channel) to detector indexing (CDC, ring, wire)

- Only the module type is required for phase I and that will be available inside the file *(see previous slide)*
- Lots of low-level diagnostics can be done using the low-level data objects without requiring the translation table

The three* databases of GlueX

- JInventory
  - Database of electronics modules, crates, etc.
  - Plan to use for translation table as well
  - Translation table will be copied (when changes detected) into conditions DB
- Conditions DB
  - Translation table will be read from here by reconstruction programs
  - Contains values related to running conditions that change as run period progresses
  - Will contain a value (or multiple values) for every run. Overhead from calibration DB style index layer would be costly (e.g. CLAS experience)
- Calibration DB
  - Contains calibration constants
  - Values may span multiple runs or be included in multiple sets motivating an indexing layer

**This does not include the EPICS Archiver DB, DocDB, etc. …*

Responsibilities

- Parties responsible for detector systems will also be responsible for maintenance of the relevant translation table in the JInventory database
- Tools will be made available with documentation to make this easy for anyone *(if you are interested in helping to write these, let me know!)*
- Module maps and any other configuration info critical to operation of the DAQ will be maintained by the DAQ experts *(i.e. the DAQ system will not rely on the Translation Table)*

Summary

- Work has begun on event source code to read raw DAQ events into DANA
  - Several low-level objects have already been defined
  - Prototype already working with 2012 beam test and bench test data
  - https://halldsvn.jlab.org/repos/trunk/sim-recon/src/programs/Utilities/plugins/DAQ
- Data parsing will be done in 2 phases allowing for a level of online monitoring that does not require a translation table
  - Requires embedding module types in data file
- Translation table will be maintained in JInventory DB, but copied into Conditions DB
  - Copied as needed to maintain record of evolution
  - Reconstruction will access Translation Table through Conditions DB
  - Subsystem experts will maintain Translation Table through a set of tools written specifically for this.

Generic pedestals?