

Calibration Infrastructure & Gaudi

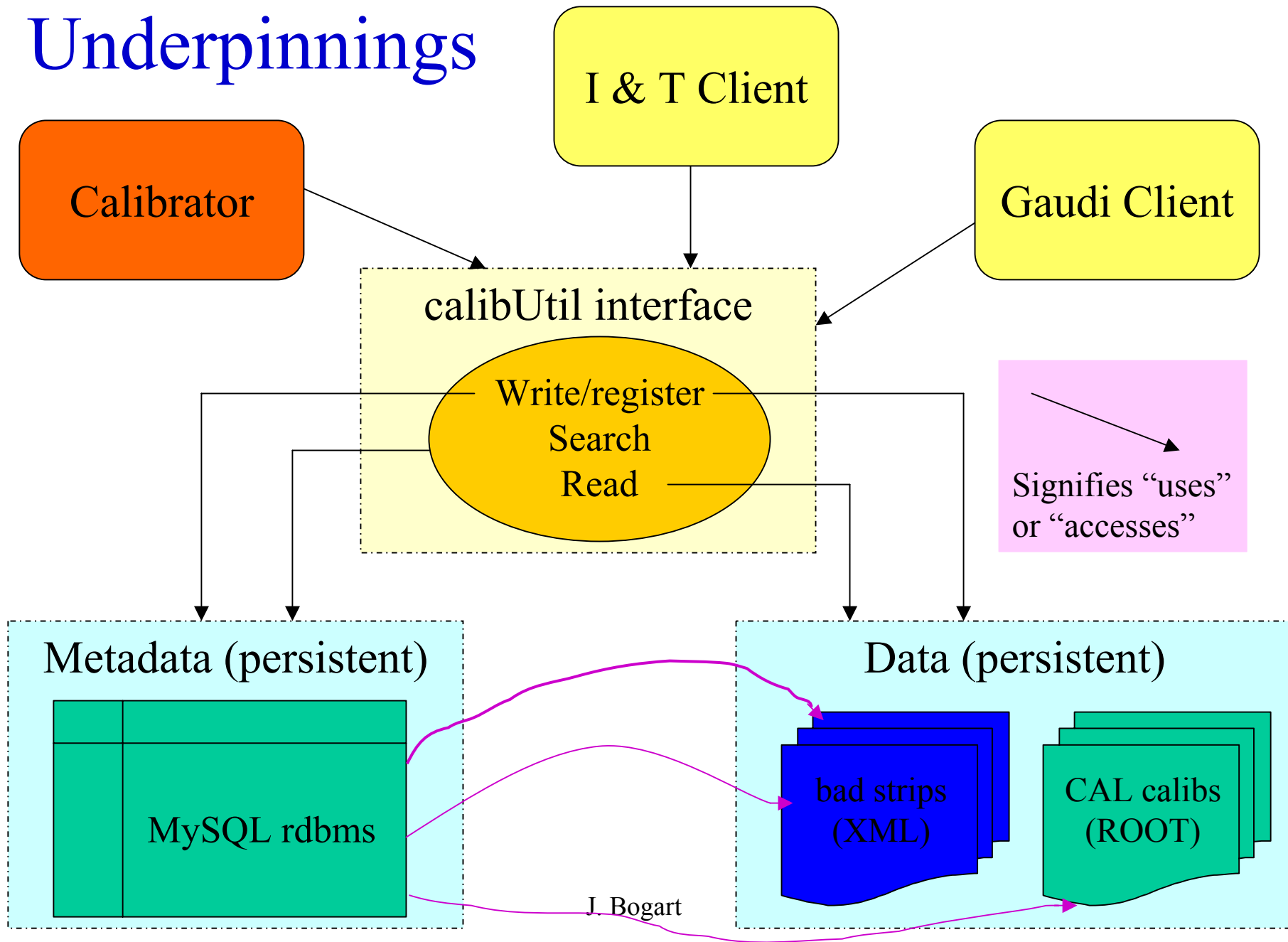
Design and Status

Review

- Calibration data comes in two pieces
 - Metadata, stored in MySQL database
 - Bulk data, probably XML or ROOT file
- Support 2 levels of access
 - Via Gaudi TDDS* for event analysis, recon.
 - Sans Gaudi for hardware evaluation, writing calibrations, browsing
- <http://www-glast.slac.stanford.edu/software/calib/> for more docs.

*Transient Detector Data Store

Underpinnings



Gaudi Interface Reqs

- Provide read access to calibration data in a form suitable for event analysis apps.
- Updates to acquire constants appropriate for current event happen automatically, without client intervention.
 - “appropriate” means
 - valid for timestamp of this event,
 - for correct instrument
 - of acceptable quality

Non-requirements

- Don't support conversion from TDDS to PDS
 - Calibration writers may choose to run in Gaudi environment, but will use calibUtil services directly to write & register data
- Don't require TDDS form to correspond closely to persistent form
- Don't require “leaf” TDDS classes to have simple structure.
 - Not convenient for bad strips
 - Consequence is that clients might need local cache, hence mechanism to be informed when TDDS class has been updated.

All of the above would be reasonably natural requirements, or at least expectations, for event TDS.

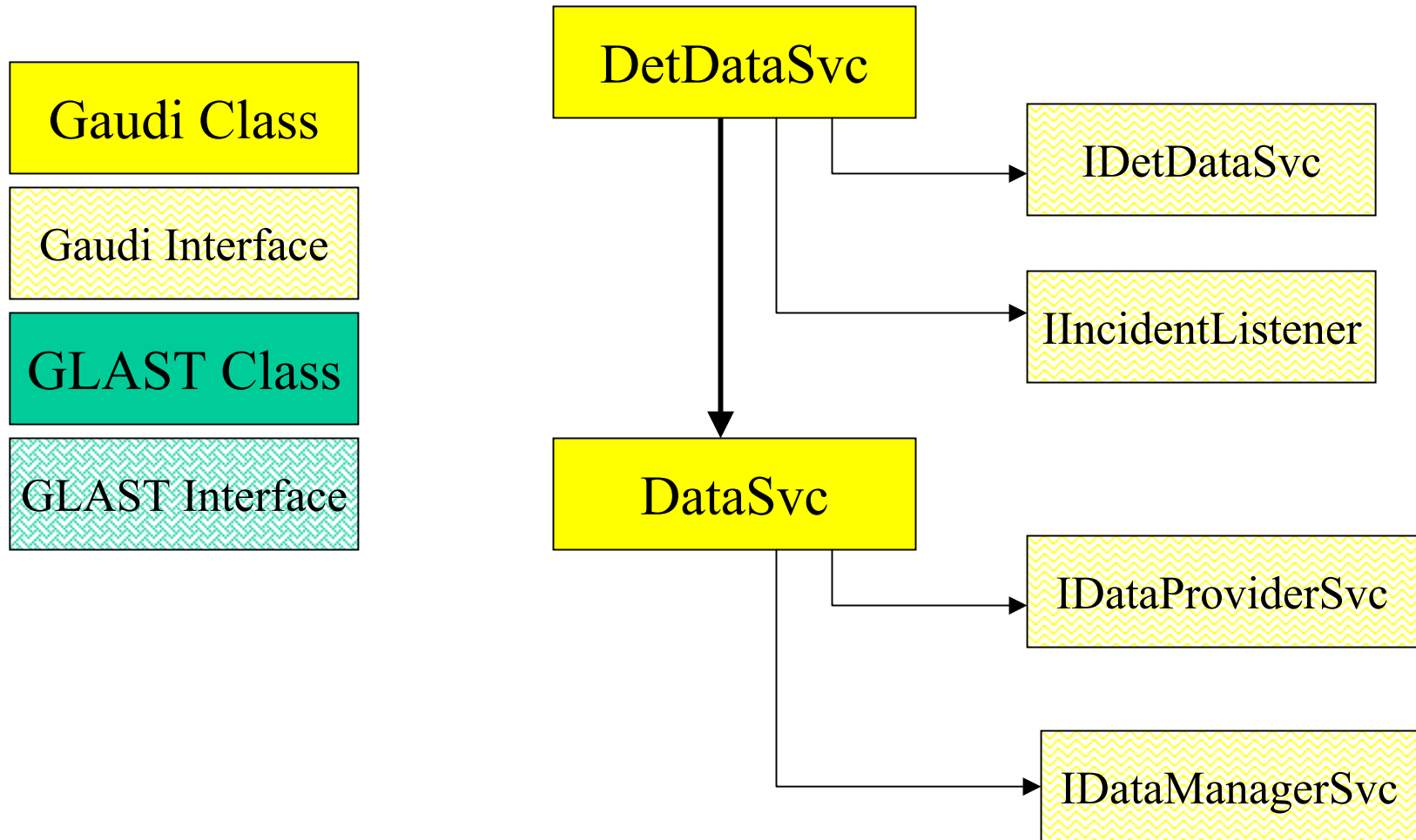
Inputs

- **Event time** must be accessible from event data.
- Identification of **instrument** should also come from event data.
- Other inputs fall under category of configuration, could be job options
 - Requirements on calibration quality
 - Host of calibration metadata database
 - Table name for metadata
 - Probably more I haven't thought of

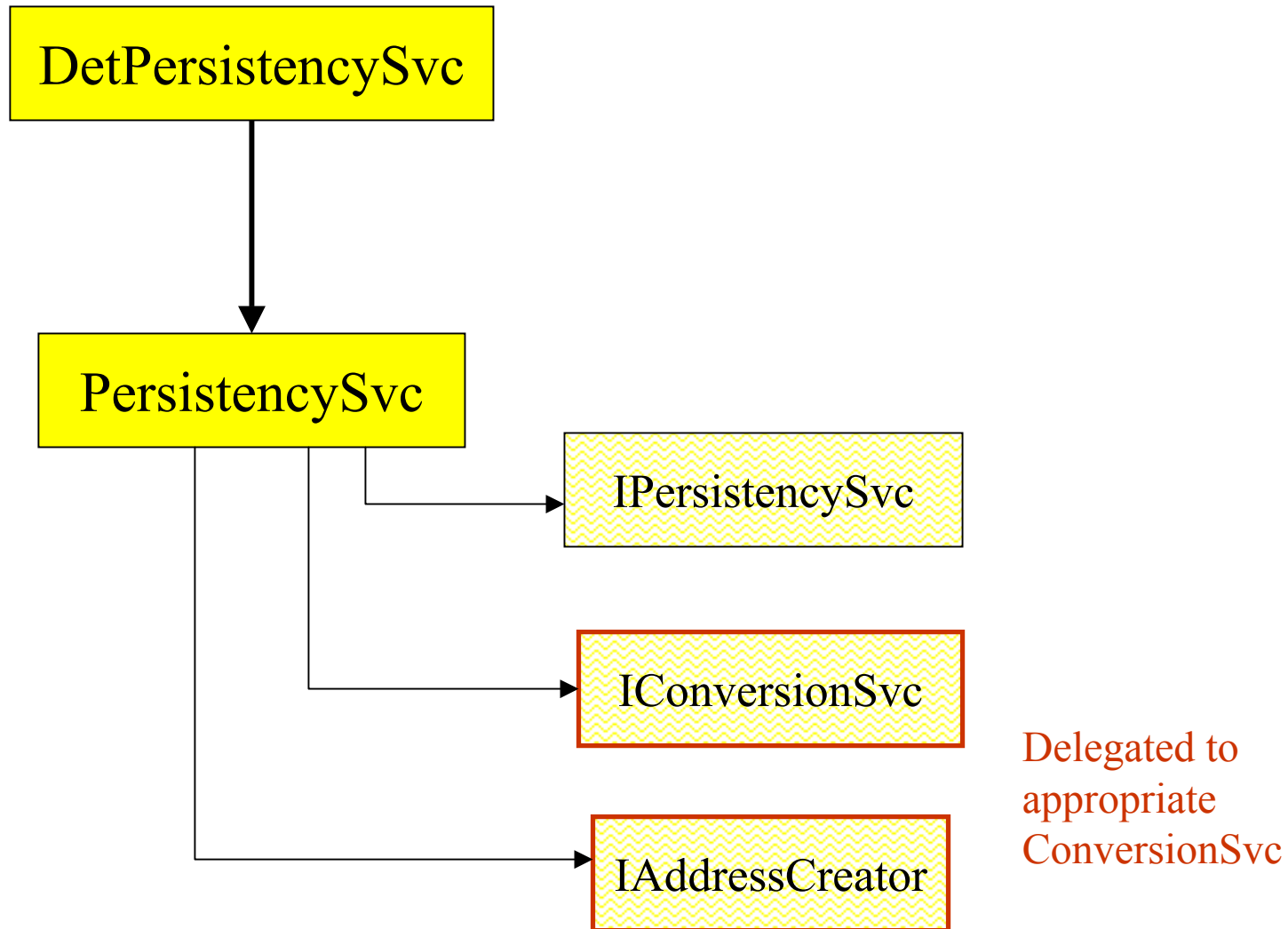
Classes

- Detector data service [DetDataSvc]
 - Base class DataSvc handles object registry, internal org. of TDDS. IDetDataSvc interface to manage event time.
- Detector persistency service [DetPersistencySvc]
 - Finds “right” conversion service – we’ll have only one
- Calibration conversion service(s) [ConversionSvc base handles converter registry]
 - One per “physical” format, but we consider all calibration data to be of one “physical” type. Finds right converter for particular DataObject. May also provide other services used by one or more converters. Here will maintain metadata database configuration, connection (instance of calibUtil::Metadata)

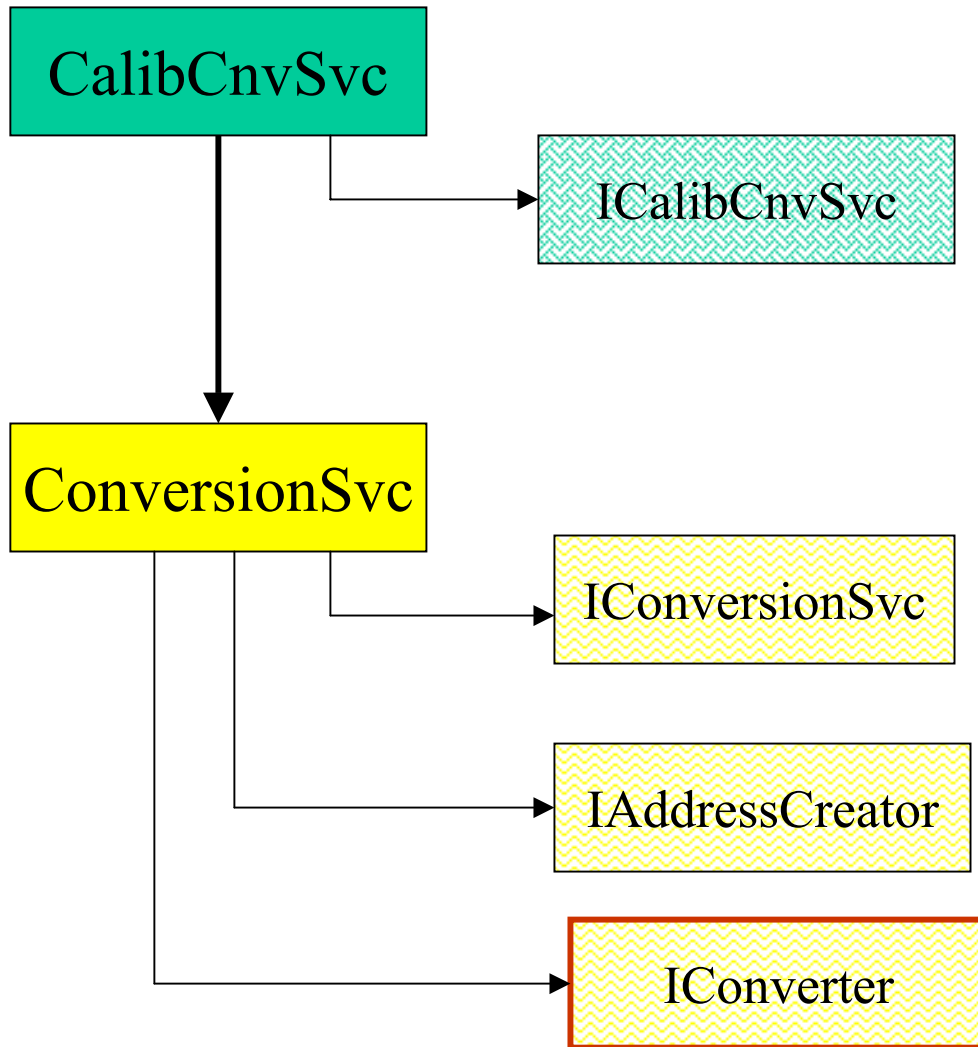
Data Service



Persistency Service



Conversion Service

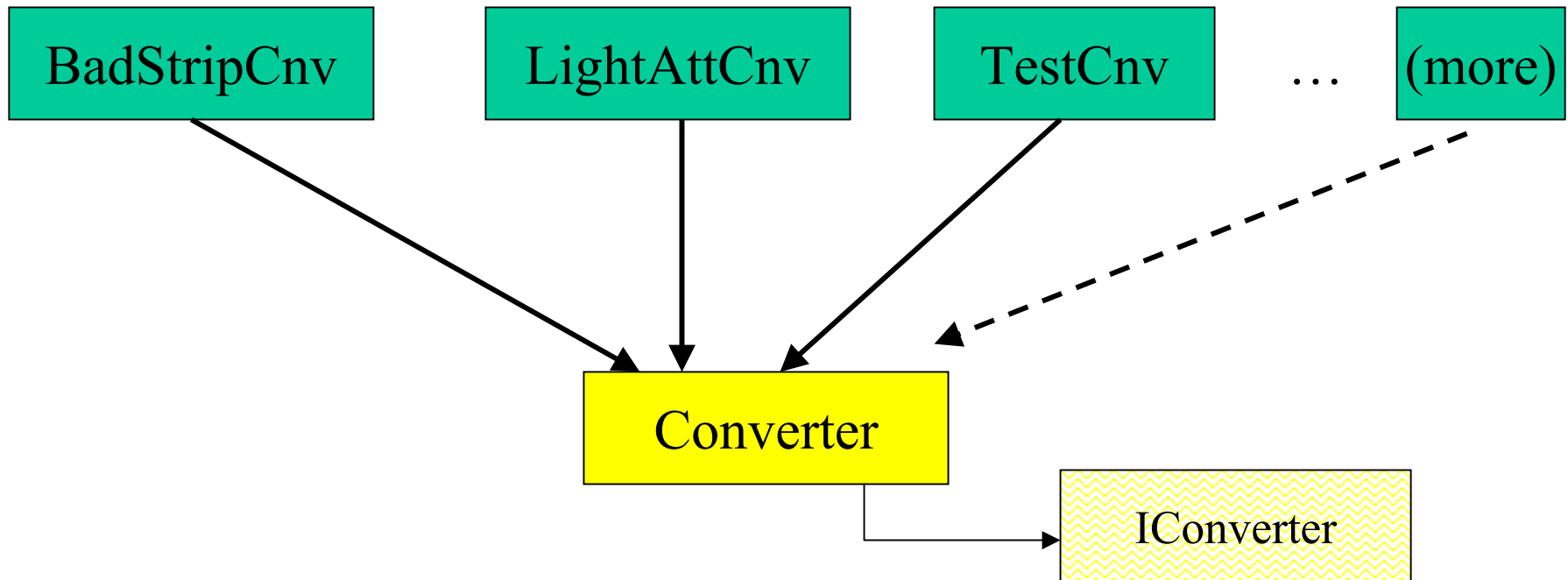


Delegated to appropriate converter

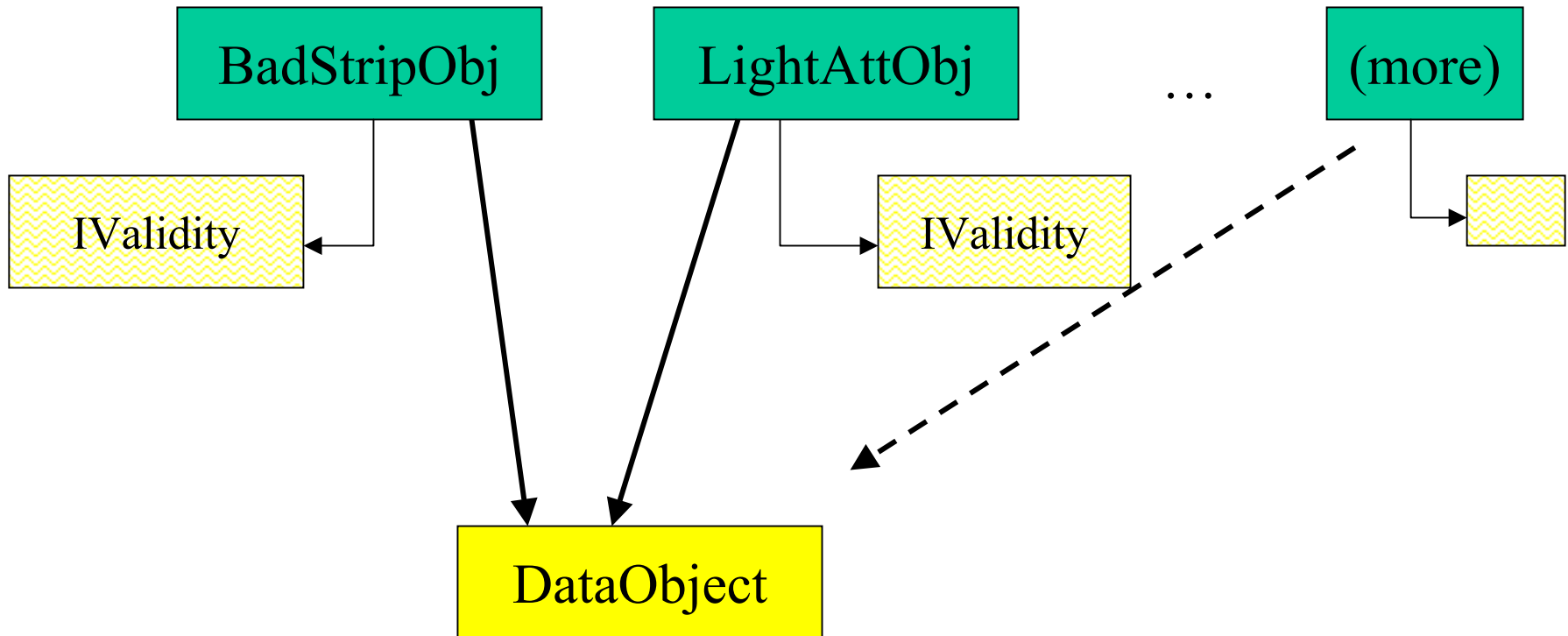
Classes (cont'd)

- Calibration converters (one per TDDS class)
- TDDS classes
 - DataObjects which also satisfy IValidity interface
- Opaque address
 - Can get by with Gaudi-supplied implementation
GenericAddress of IOpaqueAddress interface

Converters



TDDS Classes



Status, to do (non-Gaudi)

- Collaborate with CAL to design ROOT format for calibrations (ideally will be essentially identical to TDDS representation).
- Similarly for ACD?
- Provide ROOT services as needed in calibUtil
- Provide visitor for metadata database (Xin Chen request).
- Add new or modify existing Metadata class constructor so that configuration can be passed in.

Status, to do (Gaudi-related)

- Configuration of DetDataSvc, DetPersistencySvc (through job options?)
- Finish writing CalibCnvSvc
 - First pass is mostly done
- Design individual TDS classes
- Write individual converters.
 - Most should be straightforward; BadStrips is not.
- Callback mechanism?
 - Some clients of some TDDS classes might need to know if TDDS data has been updated so they can update caches.

To do (cont'd)

Test!

Many pieces have to be written and integrated before anything at all can be tested.

References

- Web accessible documents on related topics can found at <http://www-glast.slac.stanford.edu/software/calib/>
 - See especially link labeled “Calibration infrastructure and Gaudi”, which covers most of the same material as this presentation, but in more depth.
- Code.
 - Package **calibUtil** contains those classes which are independent of Gaudi.
 - **Coming soon:** two new packages for Gaudi-related code. **CalibCnv** will contain calibration conversion service and individual converters. **CalibData** will contain TDDS classes.