

# Architecture review status report

## Core service review

Gianluca Petrillo

Fermi National Accelerator Laboratory

LArSoft Coordinators' Meeting, November 24<sup>th</sup>, 2015



This is a short update on top of [the one from one week ago](#):

- renamed also two “non-core” services:
  - `IChannelStatusService`  $\Rightarrow$  `ChannelStatusService`
  - `IDetPedestalService`  $\Rightarrow$  `DetPedestalService`

Their providers have lost their `I` too, but not the `Provider`<sup>1</sup>;  
everything still in `lariov` namespace

- written legacy `LArPropertiesServiceArgoNeuT` and `DetectorPropertiesServiceArgoNeuT`, intended for use by `ArgoNeuT`

---

<sup>1</sup>Both `lariov::ChannelStatus` and `lariov::DetPedestal` classes are already defined and the standard naming pattern would conflict with them.

## Status (II)

This is a short update on top of [the one from one week ago](#):

- putting together a script to help with the code changes;  
**it will not be sufficient! manual checks and edits are still needed**
- periodically updating [the wiki page](#); now mostly stable

# Temperature is changing

We agreed to **move** `Temperature()` **method** from `LArProperties` to `DetectorProperties`.

This drags also:

- `Density()`
- `Eloss()`
- `ElossVar()`

Still ok to go?

The to-do list from last week is getting shorter and colder:

- **move** `Temperature()` to `DetectorProperties` (better be soon)
- **remove** `DatabaseUtil` from where it's not needed (can wait)
- **verify and update all** `art::ServiceHandle` uses and service dependences (can wait)
- **fix the design issue with** `DetectorClocksStandard` (can wait)  
*first try: failed!*
- **write unit tests** (can wait)

Some of these won't make it for the merging and will be postponed.

# Short-term plans

- updating user code:
  - LArIAT
  - ArgoNeuT
  - SBND, if the update to LArSoft 4.30.0 is not too traumatic
- delivery with the invitation to test it
- code (updated to LArSoft v04\_30\_00):
  - LArSoft feature/jpaley\_LArPropertiesBreakup
  - uboonecode feature/gp\_ServiceCoreReview

MicroBooNE code (uboonecode) is ready for testing!

# Backup

# Dismantling DatabaseUtil

DatabaseUtil service contains two classes of methods:

- 1 direct access to a database, only used by ArgoNeuT
- 2 MicroBooNE specific database access code

What's going to happen:

- 1 a legacy version of core properties will be created for use to ArgoNeuT
- 2 MicroBooNE specific code will be moved to `uboonecode`
- 3 each service in need will implement access to a specific DB
- 4 recommended access is not direct but via `libwda`
- 5 if a common pattern will emerge, common functions can be collected and coagulated into a service



## Design choices:

- core service providers assume they are always up to date
- users should also assume that
- the framework is responsible to update them as needed

## Funny facts:

- `DetectorClocksStandard` providers information based on a data product (`raw::Trigger` collection)
- that data product might have been created in this job, and `DetectorClocksStandard` would not know
- MicroBooNE code explicitly forced `TimeService` to check
- such a command is not available in `DetectorClocks` interface

Trying to force automatic service update on each event before each module (`sPreModule` action) did not work: *art* does not support reading from the event at that time (may be hackable...).