

# Introduction to LArSoft

A practical guide to getting started

Erica Snider  
*Fermilab*

---

*LArSoft core support team:*

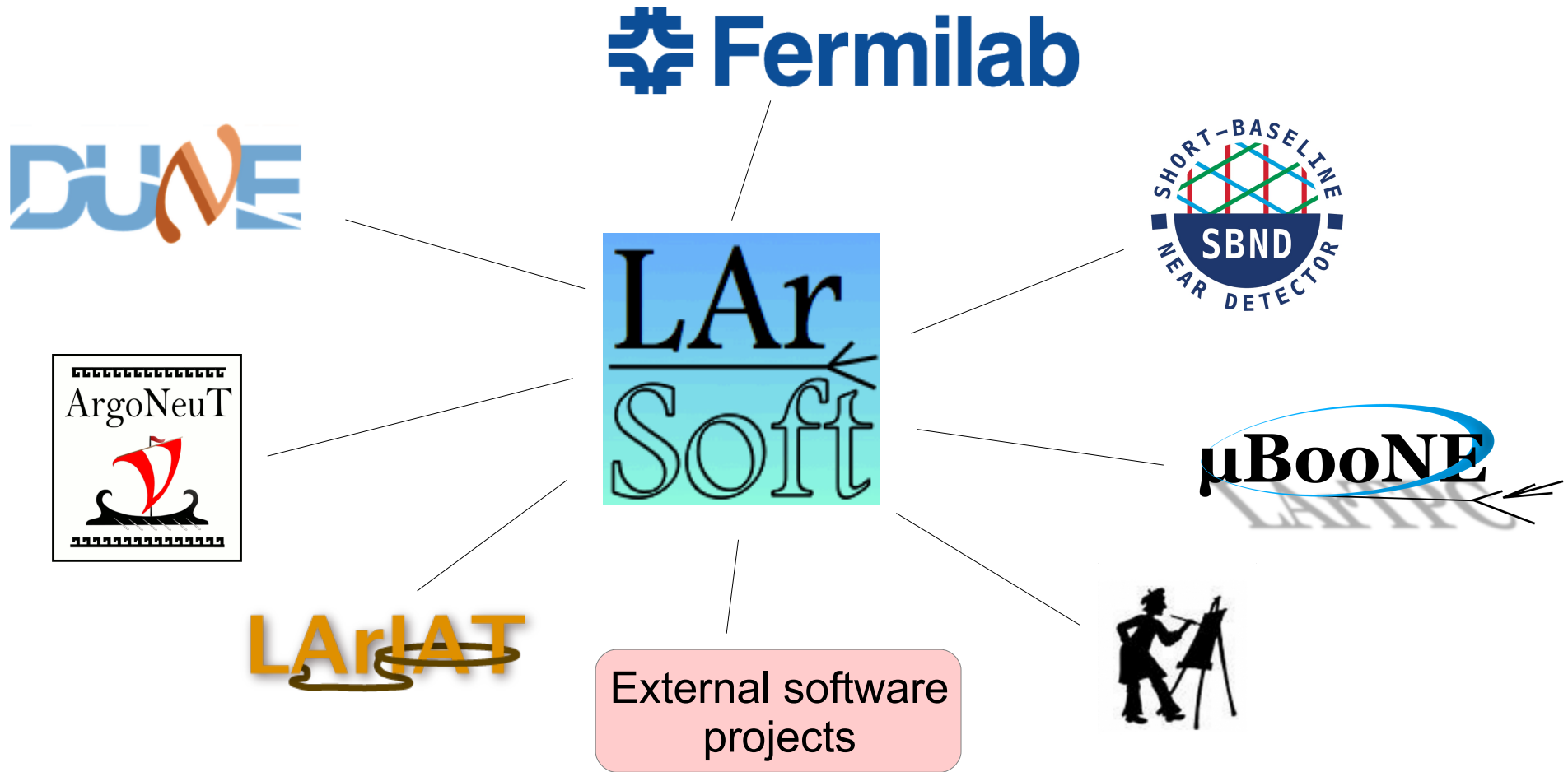
Vito di Benedetto, Giuseppe Cerati,  
Lynn Garren, Katherine Lato, Gianluca Petrillo,  
Saba Sehrish, Erica Snider

DUNE Collaboration Meeting  
Computing Tutorial  
Aug 14, 2017  
Fermilab

# Outline

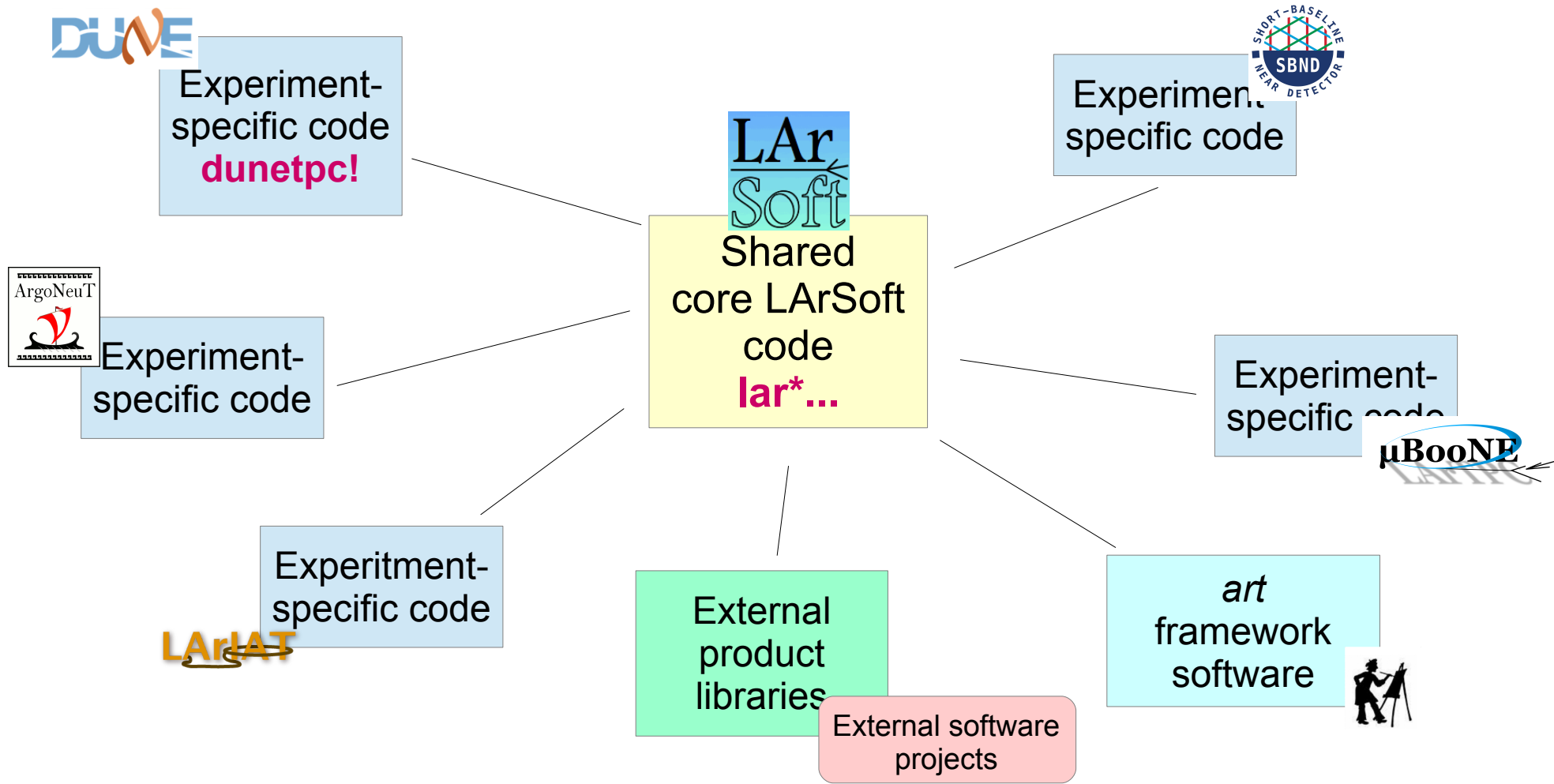
- Preliminaries
- Running a simple event generator job
- What just happened?
  - Code organization
  - LArSoft job configuration, and how to manage it
- The full event processing chain
- Modifying and navigating the code
- Resources and support

**What is LArSoft?**: (1) **A collaboration** of experiments, Fermilab, other stakeholders



To provide integrated, experiment-independent software tools for LAr TPC neutrino experiments to perform simulation, reconstruction analysis.

**What is LArSoft?:** (2) **A body of code** that interfaces with experiment-specific, *art* framework and external product software



Experiments contribute common "core" LArSoft code

290k lines of C++ in core LArSoft  
440k+ lines including expt code

# Running a simple job

# The homework example: run single muon generator job

- The homework instructions:

```
mkdir example_may_collab_2017
cd example_may_collab_2017
```

```
# setup the dunetpc environment
source /cvmfs/dune.opensciencegrid.org/products/dune/setup_dune.sh
```

```
# see what versions of dunetpc there are
ups list -aK+ dunetpc
```

```
# setup a recent one on the list:
```

```
setup dunetpc v06_34_00 -q e14:prof
```

```
# ups automatically sets up all the packages dunetpc depends on. Show a list a list of active packages
ups active
```

```
# the main executable program is called "lar". Find out where it is:
which lar
```

```
# root is also set up
which root
```

```
# run the generator for a single muon in the 1x2x6 far detector workspace geometry
# all lar commands, is successful, will end with a line - Art has completed and will exit with status 0.
lar -n 1 -c prod_muminus_0.1-5.0GeV_isotropic_dune10kt_1x2x6.fcl
```

# The homework example: run single muon generator job

- **The lines that actually matter:**

```
mkdir example_may_collab_2017
cd example_may_collab_2017
```

```
# setup the dunetpc environment
```

```
source /cvmfs/dune.opensciencegrid.org/products/dune/setup_dune.sh
```

```
# see what versions of dunetpc there are
ups list -aK+ dunetpc
```

```
# setup a recent one on the list:
```

```
setup dunetpc v06_34_00 -q e14:prof
```

```
# ups automatically sets up all the packages dunetpc depends on. Show a list a list of active packages
ups active
```

```
# the main executable program is called "lar". Find out where it is:
which lar
```

```
# root is also set up
which root
```

```
# run the generator for a single muon in the 1x2x6 far detector workspace geometry
```

```
# all lar commands, is successful, will end with a line - Art has completed and will exit with status 0.
```

```
lar -n 1 -c prod muminus 0.1-5.0GeV isotropic dune10kt 1x2x6.fcl
```

# The homework example: run single muon generator job

- **The lines that actually matter:**

```
# setup the DUNE software environment
source /cvmfs/dune.opensciencegrid.org/products/dune/setup_dune.sh

# set up some version of DUNE software (+ LArSoft implicitly)
setup dunetpc v06_34_00 -q e14:prof

# run something, here the generator for a single muon in the 1x2x6 FD workspace geometry
lar -n 1 -c prod_muminus_0.1-5.0GeV_isotropic_dune10kt_1x2x6.fcl
```

- Points to note:

- You **don't need to build any code** to run LArSoft for DUNE
- You **don't need your own fcl file** - if using "installed" fcl files
- You **don't need to be in a particular directory**
  - Just set-up your experiment environment, experiment code, then ready to go



That was simple.

...What just happened?

# What just happened (1)?

- Start by looking at the setup commands
  - Need to examine exactly what is being set up
  - This takes us to the structure of the code and ups products

```
# setup the dunetpc environment
source /cvmfs/dune.../dune/setup_dune.sh

# set up some version of DUNE software
setup dunetpc v06_34_00 -q e14:prof

# run something
lar -n 1 -c prod_muminus_..._1x2x6.fcl
```

# The setup commands

## Setting up the local environment

- The important things for LArSoft
  - sets up **ups** utility
    - A tool that configures the environment to use software products / packages
    - Will come back to this momentarily...
  - sets up **git** + **gitflow**
    - **git**: code version control system
    - **gitflow**: tool to support branching model
  - sets up **mr**b****
    - The software build tool used by LArSoft
  - defines **MRB\_PROJECT** env variable
    - needed by **mr**b**** - more later...

```

# setup the dune environment
source /cvmfs/dune.../dune/setup_dune.sh

# set up some version of DUNE software
setup dune v06_34_00 -q e14:prof

# run something
lar -n 1 -c prod_muminus_..._1x2x6.fcl
  
```

Note: The build system will change soon, so this setup may change. Details of the new system can be found in [slides from LArSoft Workshop held on June 20](#)

# The setup commands

## Set up DUNE software

- Deconstruct this a bit...

`setup dunetpc v06_34_00 -q e14:prof`

ups command to  
configure a product  
for use  
(defined as an alias)

the ups product instance  
and version version of the  
DUNE offline software

```
# setup the dunetpc environment
source /cvmfs/dune.../dune/setup_dune.sh

# set up some version of DUNE software
setup dunetpc v06_34_00 -q e14:prof

# run something
lar -n 1 -c prod_muminus_..._1x2x6.fcl
```

ups “qualifiers” that specify a  
particular build of that version

Take a (long) aside to look at the layout of the code and how ups fits in

# LArSoft **code repositories**

- Code lives in a set of git repositories hosted at Fermilab

larcore	Low level utilities
larcoreobj	Low level data products
larcorealg	Low level utilities
lardata	Data products
lardataobj	Data products
lartoolobj	Low level art tool interfaces (new!)
larsimtool	Low level simulation tool implementations (new!)
lardataalg	Low level algorithms
larevt	Low level algorithms that use data products
larsim	Simulation code
larreco	Primary reconstruction code
larana	Secondary reconstruction and analysis code
lareventdisplay	LArSoft-based event display
larpandora	LArSoft interface to Pandora
larexamples	Placeholder for examples

# LArSoft **code repositories**

- Code lives in a set of git repositories hosted at Fermilab

larcore                      Low level utilities

larcoreobi                      Low level data products

1) All publicly accessible at <http://cdcv.s.fnal.gov/projects/<repository name>>

2) For read/write access:

ssh://p-<repository name>@cdcv.s.fnal.gov/cvs/projects/<repository name>  
(requires valid kerberos ticket)

larsim                      Simulation code

larreco                      Primary reconstruction code

larana                      Secondary reconstruction and analysis code

lareventdisplay              LArSoft-based event display

larpandora                      LArSoft interface to Pandora

larexamples                      Placeholder for examples

# LArSoft **products**

- The build procedure creates and installs a **ups product** from the code in each repository

larcore	Low level utilities
larcoreobj	Low level data products
larcorealg	Low level utilities
lardata	Data products
lardataobj	Data products
lartoolobj	Low level art tool interfaces (new!)
larsimtool	Low level simulation tool implementations (new!)
lardataalg	Low level algorithms
larevt	Low level algorithms that use data products
larsim	Simulation code
larreco	Primary reconstruction code
larana	Secondary reconstruction and analysis code
lareventdisplay	LArSoft-based event display
larpandora	LArSoft interface to Pandora
larexamples	Placeholder for examples

Each product is self-contained, aside from dependencies

# LArSoft releases

- A **LArSoft release** is a **consistent set of LArSoft products** built from **tagged versions of code** in the repositories
  - Implicitly includes corresponding versions of all external dependencies used to build it
  - Each release of LArSoft has a **release notes** page on [scisoft.fnal.gov](http://scisoft.fnal.gov)
    - <http://scisoft.fnal.gov/scisoft/bundles/larsoft/<version>/larsoft-<version>.html>
- **larsoft**
  - An umbrella product that binds it all together under one version, one setup command
    - `setup larsoft v06_06_00 -q ...`
- **larsoft\_data**
  - A ups product for large configuration files

larsoft v04.16.00	
Product	Version
larcore	v04.13.00
lardata	v04.11.00
larevt	v04.08.06
larsim	v04.08.03
larreco	v04.12.00
larana	v04.08.00
lareventdisplay	v04.06.00
larpandora	v04.04.16
larexamples	v04.04.16
larsoft_data	v0.04.00

...



## dunetpc releases

- Similarly, a **dunetpc release** are bound to a particular version of LArSoft
  - By convention, the version numbering is kept in sync, aside from possible patching of production releases
- **dune\_pardata**
  - A ups product for large DUNE-specific configuration files

# Types of LArSoft releases

- Two types of LArSoft releases
  - **Integration**
    - Created weekly or on demand for special purposes
    - Contents approved at Coordination Meetings
      - Head of develop + additional branches approved at a CM or via email
    - May be removed without notice after about a month
      - In practice, we announce our intentions in advance
  - **Production**
    - Any release designated as “production” by an experiment
    - Created on demand (but usually on the weekly schedule)
    - Contents approved by the experiment declaring production
      - Typically also coordinated through the CM to keep other experiments informed
    - Production releases are retained on disk indefinitely
- List of all available tagged releases
  - [https://cdcvs.fnal.gov/redmine/projects/larsoft/wiki/LArSoft\\_release\\_list](https://cdcvs.fnal.gov/redmine/projects/larsoft/wiki/LArSoft_release_list)

## ups products

- **ups** is a tool that allows multiple concurrent versions of software libraries / products to co-exist on a single machine
  - A ups “product”
    - Collection of software, libraries, configuration files..., that define a single instance
  - The “setup” command
    - Selects a single instance to use by **defining a set of environment variables** that point to the relevant software / libraries
    - E.g., <product>\_DIR, <product>\_INC, <product>\_LIB, etc...
    - The “setup” command also **performs “setup” for any required dependencies**
    - May also define aliases for commands, etc.

The larsoft ups product is a required dependency of dunetpc, so “**setup dunetpc ...**” sets up both

# ups products

- **ups** is a tool that allows multiple concurrent versions of software libraries / products to co-exist on a single machine
  - A ups “product”
    - Collection of software, libraries, configuration files..., that define a single instance
  - The “setup” command
    - Selects a single instance to use by **defining a set of environment variables** that point to the relevant software / libraries
    - E.g., <product>\_DIR, <product>\_INC, <product>\_LIB, etc...
    - The “setup” command also **performs “setup” for any required dependencies**
    - May also define aliases for commands, etc.
  
- Other useful ups commands
  - `ups list -aK+ <product name>`
    - Readable listing of all instances of <product name>
  - `ups active`
    - List of all products currently set up, and their dependencies

# ups qualifiers

- The qualifiers
  - For LArSoft, there is **always a letter-number combination**, and **either “prof” or “debug”**
  - The letter-number combination
    - Specified the compiler version + certain compiler options
    - Currently at **“e14”** for SLx
      - Which means: **gcc v6.3.0, -std=c++14, -std=gnu (gfortran)**
  - The **“prof”** vs. **“debug”**
    - **“prof”** = full compiler optimization + with symbol tables in executable
    - **“debug”** = no optimization + with symbol tables in executable
  - **See the [CET qualifier summary wiki page](#) for more details**

...back to What just happened (2)?

- Now look at the “lar” command
  - Examine the general structure of how *art* is configured
  - Look at how to manage that configurations

```

# setup the dunetpc environment
source /cvmfs/dune.../dune/setup_dune.sh

# set up some version of DUNE software
setup dunetpc v06_34_00 -q e14:prof

# run something
lar -n 1 -c prod_muminus_..._1x2x6.fcl

```

# Running “lar”

Run *art* / LArSoft job

`lar ... -c some-file.fcl`

An alias to “art”

Q: Why not just “art -n 1 -c ...”?

A: Allows LArSoft-customized build and configuration

```
# setup the dunetpc environment
source /cvmfs/dune.../dune/setup_dune.sh

# set up some version of DUNE software
setup dunetpc v06_34_00 -q e14:prof

# run something
lar -n 1 -c prod_muminus_..._1x2x6.fcl
```

The run-time job configuration specified in FHiCL\*

First talk about *art*, then examine the fcl file used to configure our job

# The *art* event processing framework

See <https://art.fnal.gov> for links to documentation

- LArSoft is build on top of the *art* event processing framework
- The *art* framework
  - Reads events from user-specified input source
  - Invokes user-specified modules to perform reconstruction, simulation analysis, event-filtering tasks
  - May write results to one or more output files
- Modules
  - Configurable, dynamically loaded, user-written units with entry points called as specific times within the event loop
  - Three types
    - Producer: may modify the event
    - Filter: may alter flow of module processing within an event
    - Analyzer: may read information from an event, but not change it



# The *art* event processing framework

See <https://art.fnal.gov> for links to documentation

- Services
  - Configurable global utilities registered with framework, with entry points to event loop transitions and whose methods may be accessed within modules
- Tools\*
  - Configurable, local utilities callable inside modules
- The run-time configuration of art, modules, services and tools specified in FHiCL
  - See [art workbook](#) and [FHiCL quick-start guide](#) for more information on using FHiCL to configure *art* jobs
  - See <https://cdcv.s.fnal.gov/redmine/projects/fhicl-cpp/wiki/Wiki> for C++ bindings and using FHiCL parameters inside programs

More on this momentarily

# The lar run-time configuration

- For the homework example, specified in `prod_muminus_01-5.0GeV_isotropic_dune10kt_1x2x6.fcl`
  - Take a close look at that file

# Inside prod\_muminus\_01-5.0GeV\_isotropic\_dune10k5\_1x2x6.fcl

Almost everything is happening here.

Need to look there to figure out ...

...this line, which specifies the DUNE 10kt FD 1x2x6 geometry to use.

```
#include "prodsingle_common_dunefd.fcl"
```

```
process_name: SinglesGen
```

```
outputs.out1.fileName:
```

```
"prod_muminus_0.1-5.0GeV_isotropic_dune10kt_1x2x6_gen.root"
```

```
services.Geometry: @local::dune10kt_1x2x6_geo
```

```
source.firstRun: 20000014
```

```
physics.producers.generator.PDG: [ 13 ]      # mu-
physics.producers.generator.PosDist: 0        # Flat position dist.
physics.producers.generator.X0: [ 0 ]
physics.producers.generator.Y0: [ 0.0 ]
physics.producers.generator.Z0: [ 695 ]
physics.producers.generator.T0: [ 500.0 ]
physics.producers.generator.SigmaX: [ 360 ]   # x = (-3.6, 3.6)
physics.producers.generator.SigmaY: [ 600 ]   # y = (-6, 6)
physics.producers.generator.SigmaZ: [ 695 ]   # z = (0, 13.9)
physics.producers.generator.SigmaT: [ 500.0 ] # In time
physics.producers.generator.PDist: 0          # Flat momentum dist. (0.1-2.0 GeV)
physics.producers.generator.P0: [ 2.55 ]
physics.producers.generator.SigmaP: [ 2.45 ]
physics.producers.generator.AngleDist: 0      # Flat angle dist.
physics.producers.generator.Theta0XZ: [ 0. ]  # y-azimuth
physics.producers.generator.Theta0YZ: [ 0. ]  # y-latitude
physics.producers.generator.SigmaThetaXZ: [ 180. ] # Quasi-isotropic
physics.producers.generator.SigmaThetaYZ: [ 90. ]
```

Inside prod\_muminus\_01-5.0GeV

## dunetpc/fcl/dunefd/gen/single/ prodsingle\_common\_dunefd.fcl

```
#include "services_dune.fcl"
#include "singles_dune.fcl"

process_name: SinglesGen

services:
{
  # Load the service that manages root files for histograms.
  TFileService: { fileName: "single_hist_dune.root" }
  TimeTracker: {}
  MemoryTracker: { ignoreTotal: 1 } # default is one
  RandomNumberGenerator: {} #ART native random number generator
  FileCatalogMetadata: @local::art_file_catalog_mc
                        @table::dunefd_simulation_services
}

#Start each new event with an empty event.
source:
{
  module_type: EmptyEvent
  timestampPlugin: { plugin_type: "GeneratedEventTimestamp" }
  maxEvents: 10 # Number of events to create
  firstRun: 1 # Run number to use for this file
  firstEvent: 1 # number of first event in the file
}

# Define and configure some modules to do work on each event.
# First modules are defined; they are scheduled later.
# Modules are grouped by type.
physics:
{

producers:
{
  generator: @local::microboone_singlep
  rns: { module_type: "RandomNumberSaver" }
```

EmptyEvent data source module  
to create empty event record  
(since no input file)

Single-particle generator  
is set here

Inside prod\_muminus\_01-5.0GeV

But, still no dune10kt\_1x2x6\_geo.fcl,  
so drill down here...

EmptyEvent data source module  
to create empty event record  
(since no input file)

Single-particle generator  
is set here

## dunetpc/fcl/dunefd/gen/single/ prodsingle\_common\_dunefd.fcl

```
#include "services_dune.fcl"
#include "singles_dune.fcl"

process_name: SinglesGen

services:
{
  # Load the service that manages root files for histograms.
  TFileService: { fileName: "single_hist_dune.root" }
  TimeTracker: {}
  MemoryTracker: { ignoreTotal: 1 } # default is one
  RandomNumberGenerator: {} #ART native random number generator
  FileCatalogMetadata: @local::art_file_catalog_mc
                        @table::dunefd_simulation_services
}

#Start each new event with an empty event.
source:
{
  module_type: EmptyEvent
  timestampPlugin: { plugin_type: "GeneratedEventTimestamp" }
  maxEvents: 10      # Number of events to create
  firstRun: 1        # Run number to use for this file
  firstEvent: 1      # number of first event in the file
}

# Define and configure some modules to do work on each event.
# First modules are defined; they are scheduled later.
# Modules are grouped by type.
physics:
{

producers:
{
  generator: @local::microboone_singlep
  rns: { module_type: "RandomNumberSaver" }
```

root"

2.0 GeV

Inside prod\_muminus\_01-5.0GeV

...and here...

## dunetpc/dune/Utilities/services.fcl

```
#include "geometry_dune.fcl"
#include "detectorproperties_dune.fcl"
...

dunefd_services: {
  ExptGeoHelperInterface: @local::dune_geometry_helper
  Geometry: @local::dune10kt_geo
  ...
}

dunefd_simulation_services: {
  ...
```

```
}

#Start each new event with an empty event.
source:
{
  module_type: EmptyEvent
  timestampPlugin: { plugin_type: "GeneratedEventTimestamp" }
  maxEvents: 10 # Number of events to create
  firstRun: 1 # Run number to use for this file
  firstEvent: 1 # number of first event in the file
}
```

```
# Define and configure some modules to do work on each event.
# First modules are defined; they are scheduled later.
# Modules are grouped by type.
physics:
{
```

```
  producers:
  {
    generator: @local::microboone_singlep
    rns: { module_type: "RandomNumberSaver" }
```

Inside prod\_muminus\_01-5.0GeV



## dunetpc/dune/Utilities/services.fcl

```
#include "geometry_dune.fcl"  
#include "detectorproperties_dune.fcl"  
...
```

...and here are the definitions we need

## dunetpc/dune/Geometry/geometry\_dune.fcl

```
BEGIN_PROLOG  
...  
dune10kt_geo:  
{  
  Name: "dune10kt_v1"  
  
  GDML: "dune10kt_v1.gdml"  
  ROOT: "dune10kt_v1.gdml"  
  
  SortingParameters: {ChannelsPerOpDet: 12}  
  
  SurfaceY: 147828 # Underground option. 4850 feet to cm. from DocDb-3833  
  
  DisableWiresInG4: true  
}  
...  
dune10kt_1x2x6_geo: @local::dune10kt_geo  
dune10kt_1x2x6_geo.Name: "dune10kt_v1_1x2x6"  
dune10kt_1x2x6_geo.GDML: "dune10kt_v1_1x2x6.gdml"  
...
```

Note the overrides of "Name" and "GDML"

```
producers:  
{  
  generator: @local::microboone_singlep  
  rns: { module_type: "RandomNumberSaver" }
```

Inside prod\_muminus\_01-5.0GeV



## dunetpc/dune/Utilities/services.fcl

```
#include "geometry_dune.fcl"
#include "detectorproperties_dune.fcl"
...
```

## dunetpc/dune/Geometry/geometry\_dune.fcl

```
BEGIN_PROLOG
...
dune10kt_geo:
{
  Name: "dune10kt_v1"

  GDML: "dune10kt_v1.gdml"
  ROOT: "dune10kt_v1.gdml"

  SortingParameters: {ChannelsPerOpDet: 12}

  SurfaceY: 147828 # Underground option. 4850 feet to cm. from DocDb-3833

  DisableWiresInG4: true
}
...
dune10kt_1x2x6_geo: @local::dune10kt_geo
dune10kt_1x2x6_geo.Name: "dune10kt_v1_1x2x6"
dune10kt_1x2x6_geo.GDML: "dune10kt_v1_1x2x6.gdml"
...
```

The details of the geometry are defined in a **GDML** file

They are readable by both Geant4 (for the simulation) and Root (for the reconstruction)

(Note that this is an override of a value specified above.)

```
producers:
{
  generator: @local::microboone_singlep
  rns: { module_type: "RandomNumberSaver" }
```



Inside prod\_muminus\_01-5.0GeV

## dunetpc/dune/Utilities/services.fcl

```
#include "geometry_dune.fcl"
#include "detectorproperties_dune.fcl"
...
```

**Q:** How does art know where \*.gdml is?

**A:** FW\_SEARCH\_PATH env variable

Defined / modified during ups product setup procedures

All geometry options in:  
dunetpc/dune/Geometry/gdml

This directory is listed in  
FW\_SEARCH\_PATH

## dunetpc/dune/Geometry/geometry\_dune.fcl

```
BEGIN_PROLOG
...
dune10kt_geo:
{
  Name: "dune10kt_v1"

  GDML: "dune10kt_v1.gdml"
  ROOT: "dune10kt_v1.gdml"

  SortingParameters: {ChannelsPerOpDet: 12}

  SurfaceY: 147828 # Underground option. 4850 feet to cm. from DocDb-3833

  DisableWiresInG4: true
}
...
dune10kt_1x2x6_geo: @local::dune10kt_geo
dune10kt_1x2x6_geo.Name: "dune10kt_v1_1x2x6"
dune10kt_1x2x6_geo.GDML: "dune10kt_v1_1x2x6.gdml"
...
```

```
producers.
{
  generator: @local::microboone_singlep
  rns: { module_type: "RandomNumberSaver" }
```



Inside prod\_muminus\_01-5.0GeV

## dunetpc/fcl/dunefd/gen/single/ prodsingle\_common\_dunefd.fcl

```
#include "services_dune.fcl"
#include "singles_dune.fcl"

process_name: SinglesGen

services:
{
  # Load the service that manages root files for histograms.
  TFileService: { fileName: "single_hist_dune.root" }
  TimeTracker: {}
  MemoryTracker: { ignoreTotal: 1 } # default is one
  RandomNumberGenerator: {} #ART native random number generator
  FileCatalogMetadata: @local::art_file_catalog_mc
                        @table::dunefd_simulation_services
}

#Start each new event with an empty event.
source:
{
  module_type: EmptyEvent
  timestampPlugin: { plugin_type: "GeneratedEventTimestamp" }
  maxEvents: 10 # Number of events to create
  firstRun: 1 # Run number to use for this file
  firstEvent: 1 # number of first event in the file
}

# Define and configure some modules to do work on each event.
# First modules are defined; they are scheduled later.
# Modules are grouped by type.
physics:
{

  producers:
  {
generator: @local::microboone_singlelep
    rns: { module_type: "RandomNumberSaver" }
  }
}
```

Going back to prodsingle\_common...

The same exercise here...

**larsim/larsim/EventGenerator/singles.fcl**

```

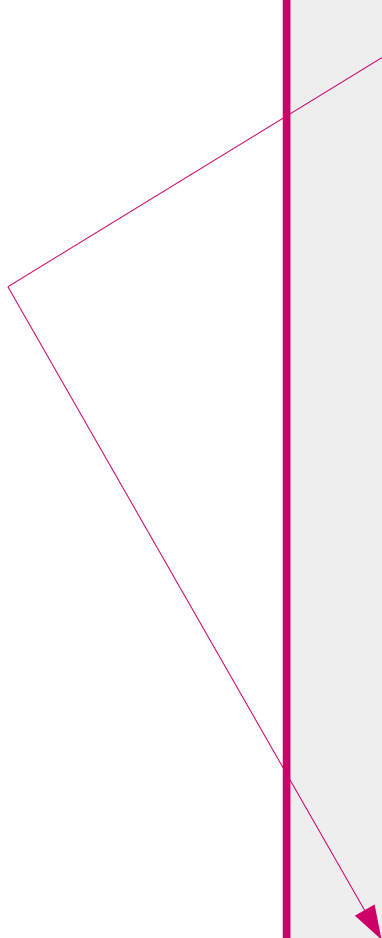
BEGIN_PROLOG

#no experiment specific configurations because SingleGen is detector agnostic

standard_singlep:
{
  module_type:      "SingleGen"
  ParticleSelectionMode: 0      # 0 = use full list, 1 = randomly select
  PadOutVectors:    false      # false: require all vectors to be same length
                                # true: pad out if a vector is size one
  PDG:              [ 13 ]     # list of pdg codes for particles to make
  PO:               [ 6. ]     # central value of momentum for each particle
  SigmaP:           [ 0. ]     # variation about the central value
  PDist:            1          # 0 - uniform, 1 - gaussian distribution
  XO:               [ 25. ]    # in cm in world coordinates, ie x = 0 is at the
                                # and increases away from the wire plane
  YO:               [ 0. ]     # in cm in world coordinates, ie y = 0 is at the
  ZO:               [ 20. ]    # in cm in world coordinates, ie z = 0 is at the
                                # the TPC and increases with the beam direction
  TO:               [ 0. ]     # starting time
  SigmaX:           [ 0. ]     # variation in the starting x position
  SigmaY:           [ 0. ]     # variation in the starting y position
  SigmaZ:           [ 0.0 ]    # variation in the starting z position
  SigmaT:           [ 0.0 ]    # variation in the starting time
  PosDist:          0          # 0 - uniform, 1 - gaussian
  TDist:            0          # 0 - uniform, 1 - gaussian
  ThetaOXZ:        [ 0. ]     #angle in XZ plane (degrees)
  ThetaOYZ:        [ -3.3 ]   #angle in YZ plane (degrees)
  SigmaThetaXZ:    [ 0. ]     #in degrees
  SigmaThetaYZ:    [ 0. ]     #in degrees
  AngleDist:       1          # 0 - uniform, 1 - gaussian
}
...
microboone_singlep: @local::standard_singlep
microboone_singlep.ThetaOYZ: [ 0.0 ] # beam is along the z axis.
microboone_singlep.XO: [ 125 ]      # in cm in world coordinates,
microboone_singlep.ZO: [ 50 ]       # in cm in world coordinates
    
```

...takes us here...

...where we find the definitions we need



root"

2.0 GeV

# The lar run-time configuration

- Q: How does *art* find the fcl file?
- A: FHICL\_FILE\_PATH environment variable
  - Path to FHiCL directories defined by dunetpc and the LArSoft ups products

## For dunetpc:

```

.
./job
/cvmfs/dune.opensciencegrid.org/products/dune/dunetpc/v06_34_00/job
/cvmfs/dune.opensciencegrid.org/products/dune/lbne_raw_data/v1_04_11/fcl
.
./job
/cvmfs/fermilab.opensciencegrid.org/products/larsoft/larwirecell/v06_04_02/fcl
/cvmfs/fermilab.opensciencegrid.org/products/larsoft/larpandora/v06_10_02/job
/cvmfs/fermilab.opensciencegrid.org/products/larsoft/larana/v06_05_02/job
/cvmfs/fermilab.opensciencegrid.org/products/larsoft/larexamples/v06_03_02/job
/cvmfs/fermilab.opensciencegrid.org/products/larsoft/lareventdisplay/v06_05_02/job
/cvmfs/fermilab.opensciencegrid.org/products/larsoft/larreco/v06_26_00/job
/cvmfs/fermilab.opensciencegrid.org/products/larsoft/larsim/v06_20_00/job
/cvmfs/fermilab.opensciencegrid.org/products/larsoft/larevt/v06_13_00/job
/cvmfs/fermilab.opensciencegrid.org/products/larsoft/lardata/v06_20_01/job
/cvmfs/fermilab.opensciencegrid.org/products/larsoft/lardataobj/v1_15_02/job
/cvmfs/fermilab.opensciencegrid.org/products/larsoft/nutools/v2_12_01/fcl
/cvmfs/fermilab.opensciencegrid.org/products/larsoft/nusimdata/v1_07_01/fcl
/cvmfs/fermilab.opensciencegrid.org/products/larsoft/larcore/v06_09_01/job
/cvmfs/fermilab.opensciencegrid.org/products/larsoft/larcoreobj/v1_13_01/job

```

# The lar run-time configuration

- Q: How does *art* find the fcl file?
- A: FHICL\_FILE\_PATH environment variable
  - Path to FHICL directories defined by dunetpc and the LArSoft ups products

## For dunetpc:

**./job** ← Note that these are always first in the path

```

./job
/cvmfs/dune.opensciencegrid.org/products/dune/dunetpc/v06_34_00/job
/cvmfs/dune.opensciencegrid.org/products/dune/lbne_raw_data/v1_04_11/fcl
.
./job
/cvmfs/fermilab.opensciencegrid.org/products/larsoft/larwirecell/v06_04_02/fcl
/cvmfs/fermilab.opensciencegrid.org/products/larsoft/larpandora/v06_10_02/job
/cvmfs/fermilab.opensciencegrid.org/products/larsoft/larana/v06_05_02/job
/cvmfs/fermilab.opensciencegrid.org/products/larsoft/larexamples/v06_03_02/job
/cvmfs/fermilab.opensciencegrid.org/products/larsoft/lareventdisplay/v06_05_02/job
/cvmfs/fermilab.opensciencegrid.org/products/larsoft/larreco/v06_26_00/job
/cvmfs/fermilab.opensciencegrid.org/products/larsoft/larsim/v06_20_00/job
/cvmfs/fermilab.opensciencegrid.org/products/larsoft/larevt/v06_13_00/job
/cvmfs/fermilab.opensciencegrid.org/products/larsoft/lardata/v06_20_01/job
/cvmfs/fermilab.opensciencegrid.org/products/larsoft/lardataobj/v1_15_02/job
/cvmfs/fermilab.opensciencegrid.org/products/larsoft/nutools/v2_12_01/fcl
/cvmfs/fermilab.opensciencegrid.org/products/larsoft/nusimdata/v1_07_01/fcl
/cvmfs/fermilab.opensciencegrid.org/products/larsoft/larcore/v06_09_01/job
/cvmfs/fermilab.opensciencegrid.org/products/larsoft/larcoreobj/v1_13_01/job

```

# Tools and techniques for fcl files

- Best practices for writing fcl files
  - Guidelines explained in presentation by Kyle Knoepfel
    - [Presentation from 2016 LArSoft Workshop](#)
    - Not things that the typical *user* needs to know, but...
      - ...helps to answer why things are this way
    - It is *required* information for people who *write* modules or production workflows
      - E.g., fcl validation features
    - Basically calls for **highly nested structures that layer overrides**

!!!

Bottom line: need good tools to help validate and debug

# Tools and techniques for fcl files

- How do I examine final parameter values for a given fcl file?
  - `fhicl-expand`
    - Performs all “#include” directives, creates a single output with the result
  - `fhicl-dump`
    - Parses the entire file hierarchy, **prints the final state** all FHiCL parameters
    - **Using the “--annotate” option, also lists the fcl file + line number** at which each parameter takes its final value
    - Requires FHiCL\_FILE\_PATH to be defined
- How do I tell the FHiCL parameter values for a processed file?
  - `config_dumper`
    - Prints the full configuration for the processes that created the file

All of these programs:

- print instructions given “--help” option
- are available when art is set up

# Tools and techniques for fcl files

- Finding fcl files (ok, no centralized tools for this...)
  - Follow FHICL\_FILE\_PATH
    - Most people write their own scrips / aliases, e.g.  
`echo $FHICL_FILE_PATH | sed -e "s:/\n/g" | xargs ...`
    - **This only finds the installed locations**, which does not tell you where to find them within any repository
  - Finding files / parameters inside repositories
    - Again, most people have their favorites, e.g.  
`find . -name *fcl | xargs grep ...`

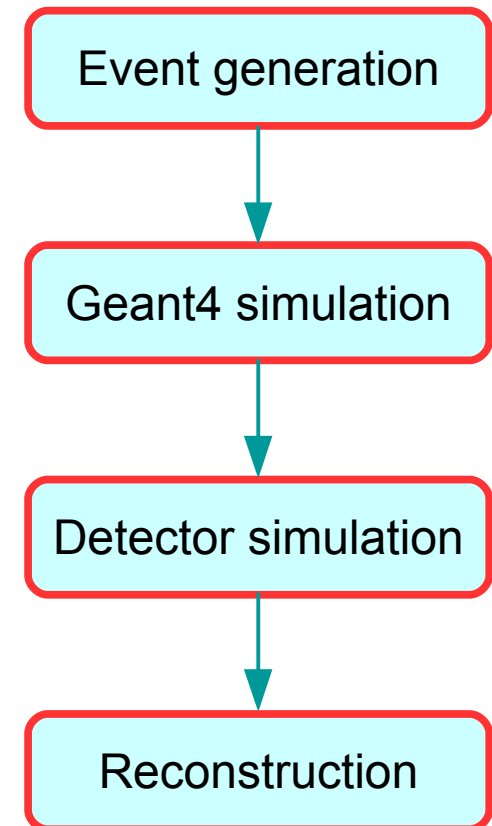


# The full event processing chain

## The rest of the processing chain

Major processing steps are in a set of pre-defined fcl files

- e.g., the “homework” example

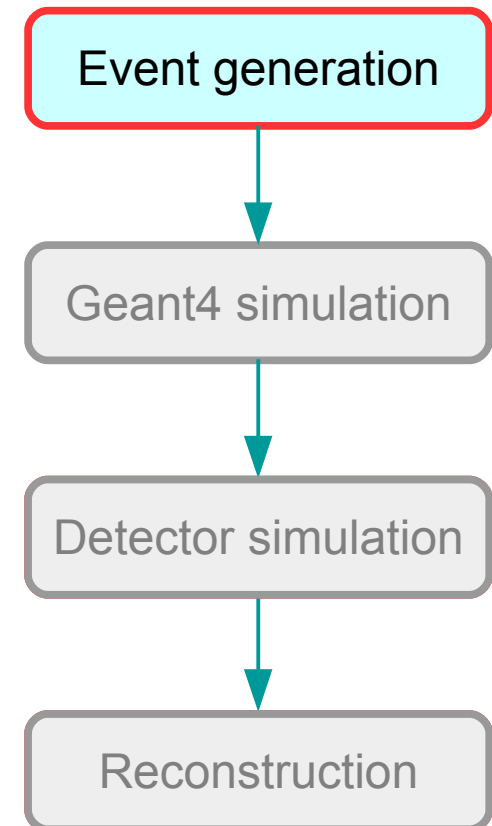


# The rest of the processing chain

Major processing steps are in a set of pre-defined fcl files

- e.g., the “homework” example

- First example was SingleGen Module
  - In `larsim/larsim/EventGenerator`
    - fcl was in `dunetpc/fcl/dunefd/gen/single/`



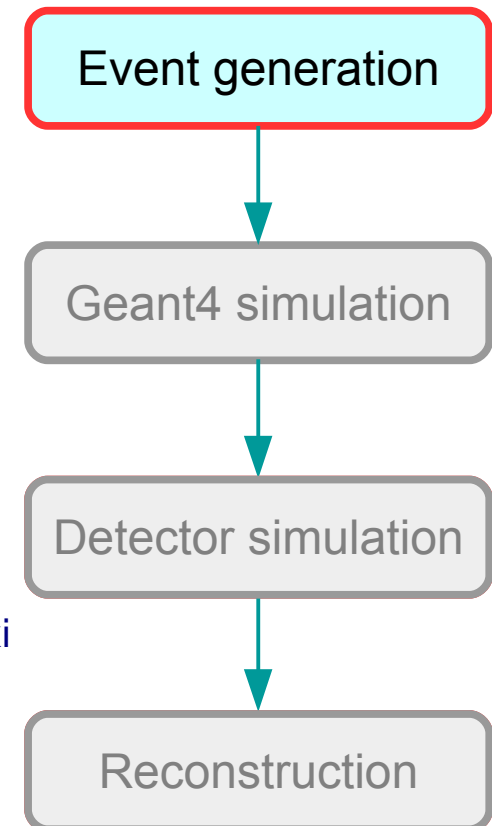
# The rest of the processing chain

Major processing steps are in a set of pre-defined fcl files

- e.g., the “homework” example

- Other event generation options

- GENIE: **GENIEGen** module
  - Neutrino event generator
  - Direct interface to GENIE neutrino event generator
  - [larsim/larsim/EventGenerator/GENIE/](#)
  - See [genie.fcl](#) in that directory
  - More documentation on the NuTools wiki page <https://cdcv.sfnal.gov/redmine/projects/nutools/wiki>



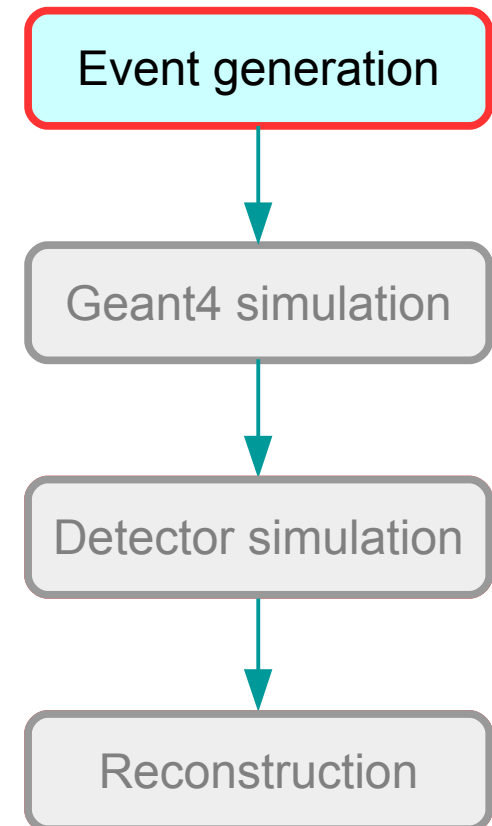
# The rest of the processing chain

Major processing steps are in a set of pre-defined fcl files

- e.g., the “homework” example

- Other event generation options

- GENIE: [GENIEGen](#) module
- NuWro: [NuWroGen](#) module
  - Neutrino event generator
  - Indirect interface: reads input root file produced by stand-alone NuWro neutrino event generator to create `simb::MCTruth` data products
  - [larsim/larsim/EventGenerator/NuWro/](#)
  - See [prodnuwro.fcl](#) in that directory



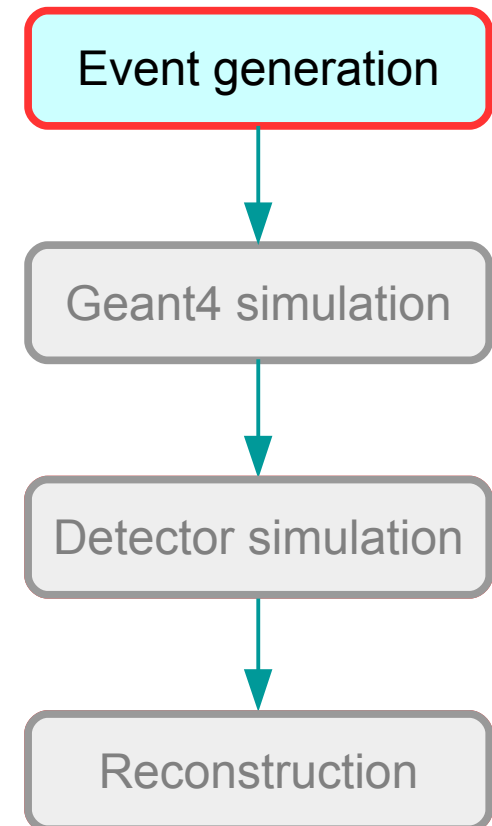
# The rest of the processing chain

Major processing steps are in a set of pre-defined fcl files

- e.g., the “homework” example

- Other event generation options

- GENIE: [GENIEGen](#) module
- NuWro: [NuWroGen](#) module
- CORSIKA: [CORSIKAGen](#) module
  - Cosmic ray generator
  - Indirect interface: reads externally produced SQLite database files with comic ray air showers produced by stand-alone CORSIKA program
  - [larsim/larsim/EventGenerator/CORSIKA/](#)
  - See [CORSIKAGen.fcl](#) in that directory



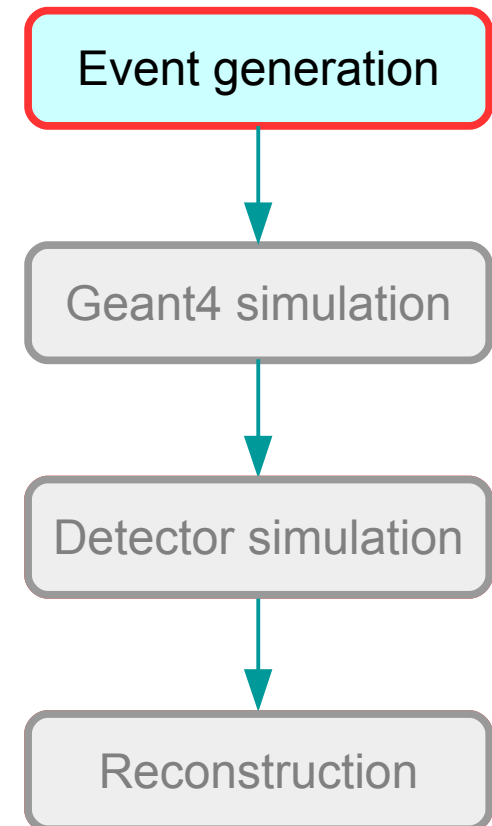
# The rest of the processing chain

Major processing steps are in a set of pre-defined fcl files

- e.g., the “homework” example

- Other event generation options

- GENIE: [GENIEGen](#) module
- NuWro: [NuWroGen](#) module
- CORSIKA: [CORSIKAGen](#) module
- CRY: [CosmicsGen](#) module
  - Cosmic ray generator
  - Embedded generator: the module contains the event generation code
  - [larsim/larsim/EventGenerator/CRY/](#)
  - See [cry.fcl](#) in that directory



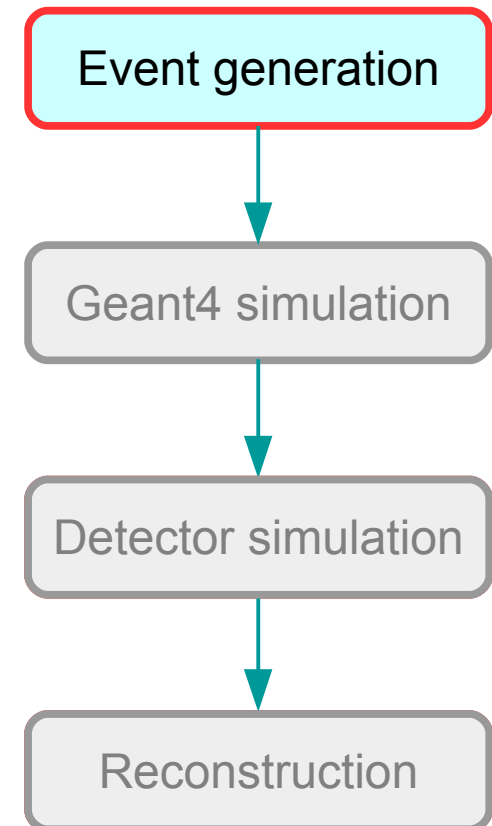
# The rest of the processing chain

Major processing steps are in a set of pre-defined fcl files

- e.g., the “homework” example

- Other event generation options

- GENIE: [GENIEGen](#) module
- NuWro: [NuWroGen](#) module
- CORSIKA: [CORSIKAGen](#) module
- CRY: [CosmicsGen](#) module
- NDK: [NDKGen](#) module
  - Nucleon decay generator
  - Indirect interface
  - [larsim/larsim/EventGenerator/](#)





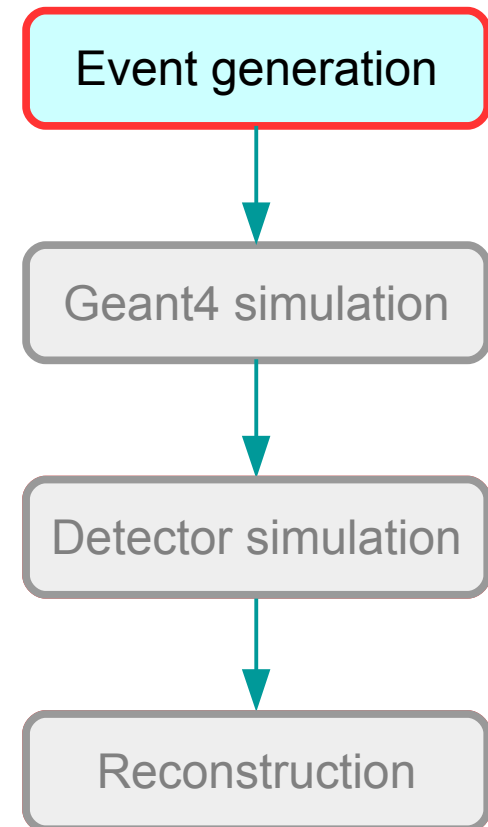
# The rest of the processing chain

Major processing steps are in a set of pre-defined fcl files

- e.g., the “homework” example

- Other event generation options

- GENIE: **GENIEGen** module
- NuWro: NuWroGen module
- CORSIKA: **CORSIKAGen** module
- CRY: **CosmicsGen** module
- NDK: **NDKGen** module
- **TextFileGen** module
  - When all else fails...reads a text file, produces simb::MCTruth
  - [larsim/larsim/EventGenerator/](#)
- **Others in larsim/larsim/EventGenerator**



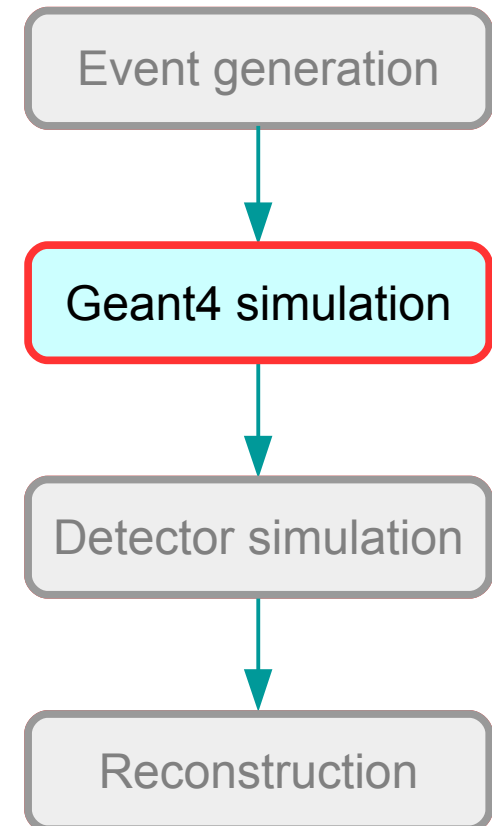
# The rest of the processing chain

Major processing steps are in a set of pre-defined fcl files

- e.g., the “homework” example

- Geant4 simulation

- Traces energy deposition, secondary interactions within LAr
- Also performs electron / photon transport
- **LArG4** module in `larsim/larsim/LArG4`
- Note:
  - Many generator / simulation interfaces are defined in `nutools` product.
- Homework fcl:
  - `standard_g4_dune10kt_1x2x6.fcl`
  - In `dunetpc/fcl/dunefd/g4/`



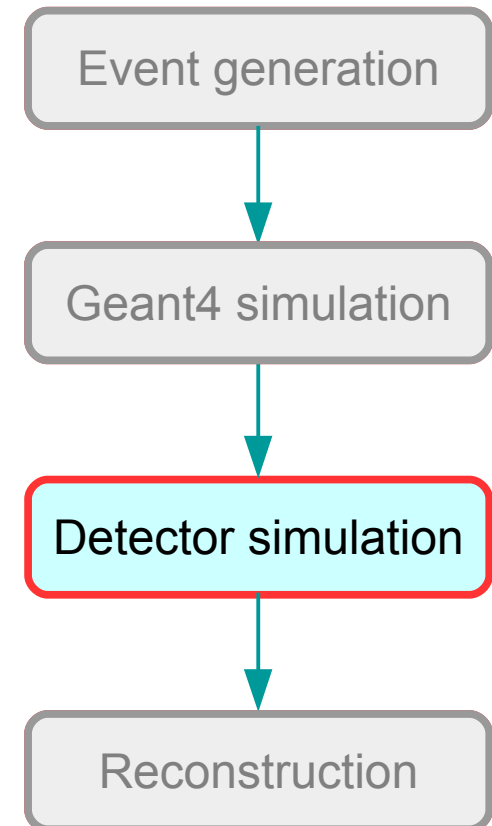
# The rest of the processing chain

Major processing steps are in a set of pre-defined fcl files

- e.g., the “homework” example

- **Detector simulation**

- Detector and readout effects
- Field response, electronics response, digitization
- Historically, most of this code is experiment-specific
  - `dunetpc`
  - More recently, the active development is part of wire-cell project with interfaces to LArSoft
- Homework fcl:
  - `standard_detsim_dune10kt_1x2x6.fcl`
  - In `dunetpc/fcl/dunefd/detsim/`



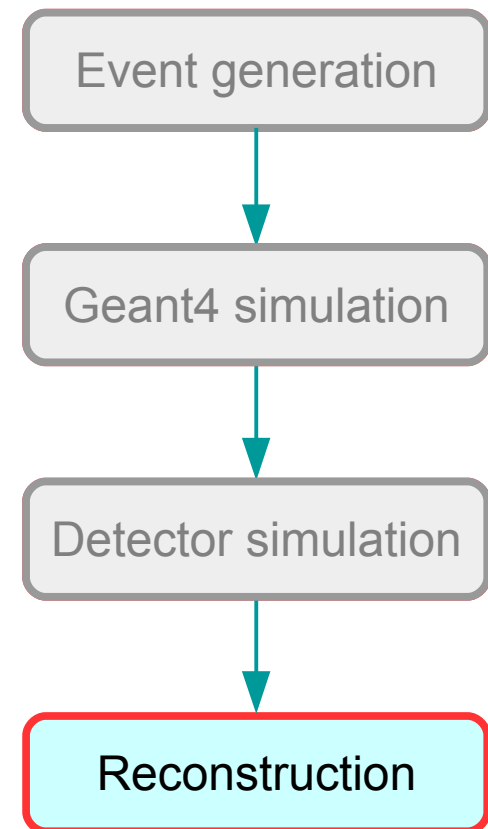
# The rest of the processing chain

Major processing steps are in a set of pre-defined fcl files

- e.g., the “homework” example

- **Reconstruction**

- Performs pattern recognition, extracts information about physical objects and processes in the event
- May include signal processing, hit-finding, clustering of hits, view matching, track and shower finding, particle ID
  - 2D and 3D algorithms
  - External RP interfaces for Pandora and Wire-cell
- Homework fcl:
  - `standard_reco_dune10kt_1x2x6.fcl`
  - In `dunetpc/fcl/dunfd/reco/`



# Modifying and navigating the code

# Modifying the configuration of an existing job

- Suppose you need modify a parameter in a pre-defined job
- Several options. Here are two.
  - Option 1
    - Copy the fcl file that *defines* the parameter to “pwd” for the `lar` command
    - Modify the parameter
    - Run `lar -c ...` as before
      - The modified version will get picked because “.” is always first in `FHICL_FILE_PATH`
  - Option 2
    - Copy the top-level fcl file to the “pwd” for the `lar` command
    - Add an override line to the top-level fcl file
    - E.g., in the homework generator job, all those lines at the bottom:

```

...
services.Geometry: @local::dune10kt_1x2x6_geo
source.firstRun: 20000014

physics.producers.generator.PDG: [ 13 ]      # mu-
physics.producers.generator.PosDist: 0      # Flat position dist.
...

```

More obvious  
what changed  
wrt default

# Modifying the code of an existing job

In cases where configuration changes will not be sufficient, you will need to modify, build, then run code:


- Create a new working area from a fresh login + DUNE set-up

```
mkdir <working_dir>
cd <working_dir>
mrB newDev -v <version> -q <qualifiers>
```

(Note, if dunetpc/larsoft is already set up, then only need “`mrB newDev`”)

- This creates the three following directories inside <working\_dir>

```
<working_dir>/localProducts_<MRB_PROJECT>_<version>_<qualifiers>
    /build_<os flavor>
    /srcs
```



Source directory

Build directory

Local products directory

See lecture by Saba Sehrish at 2015 LArSoft class for more complete description of how to check out, build, run code and more. I will not improve on that...

# Modifying the code of an existing job

In ca  
nee

## An aside:

- `mrbs` : multi-repository build system
  - Purpose is to simplify building of multiple products pulled from separate repositories
  - `setup mrbs` command executed in experiment setup
  - Most commonly used commands
    - `mrbs --help` prints list of all commands with brief descriptions
    - `mrbs <command> --help` displays help for that command
    - `mrbs gitCheckout` clone a repository into working area
    - `mrbs setenv` set up build environment
    - `mrbs build / install -jN` build/install local code with N cores
    - `mrbs slp` set up all products in localProducts...
    - `mrbs z` zap everything in build area

Se  
of ho



# Modifying the code of an existing job

- Set up local products and development environment

```
source localProducts_<MRB_PROJECT>_<version>_<qualifiers>/setup
```

- Creates a number of new environment variables, including
  - `MRB_SOURCE`           points to the `srcs` directory
  - `MRB_BUILDDIR`       points to the `build_...` directory
  - Modifies `PRODUCTS` to include `localProducts...` as the first entry

- Check out the repository to be modified

(and maybe others that depend on any header files to be modified)

```
cd $MRB_SOURCE
mrbs g dunetpc           # "g" is short for gitCheckout
```

- Clones `dunetpc` from current head of "develop" branch
- Adds the repository to top-level build configuration file (`CMakeLists.txt`)
- Optionally can check out a particular tag using `-t <tag>` option
  - To get the code for a particular release:
  - `mrbs g -t LARSOFT_SUITE_<version> <repository>`

## Modifying the code of an existing job

- Make changes to the code
  - More on this momentarily...
- Go to build directory and set up development environment
 

```
cd $MRB_BUILDDIR
mrbsetenv
```
- Build the local code
 

```
mrb b [-jN]      # "b" is short for "build"
```

  - Libraries are in the build directory at this point.
- Install local ups products from the code you just built
 

```
mrb i [-jN]      # "i" is short for "install". This will do "build" also
```

  - Files are re-organized and moved into localProducts... directory
    - All fcl files are put into a top-level "job" directory with no sub-structure
    - All header files are put into a top-level "include" directory with sub-directories
    - Other files are moved to various places, including source files, while some, such as build configuration files, are ignored and not put anywhere in the ups product

## Modifying the code of an existing job

- Now set-up the local versions of the products just installed

```
cd $MRB_TOP  
mrbslp
```

- Run the code you just built

```
lar -c <whatever fcl file you were using> ...
```

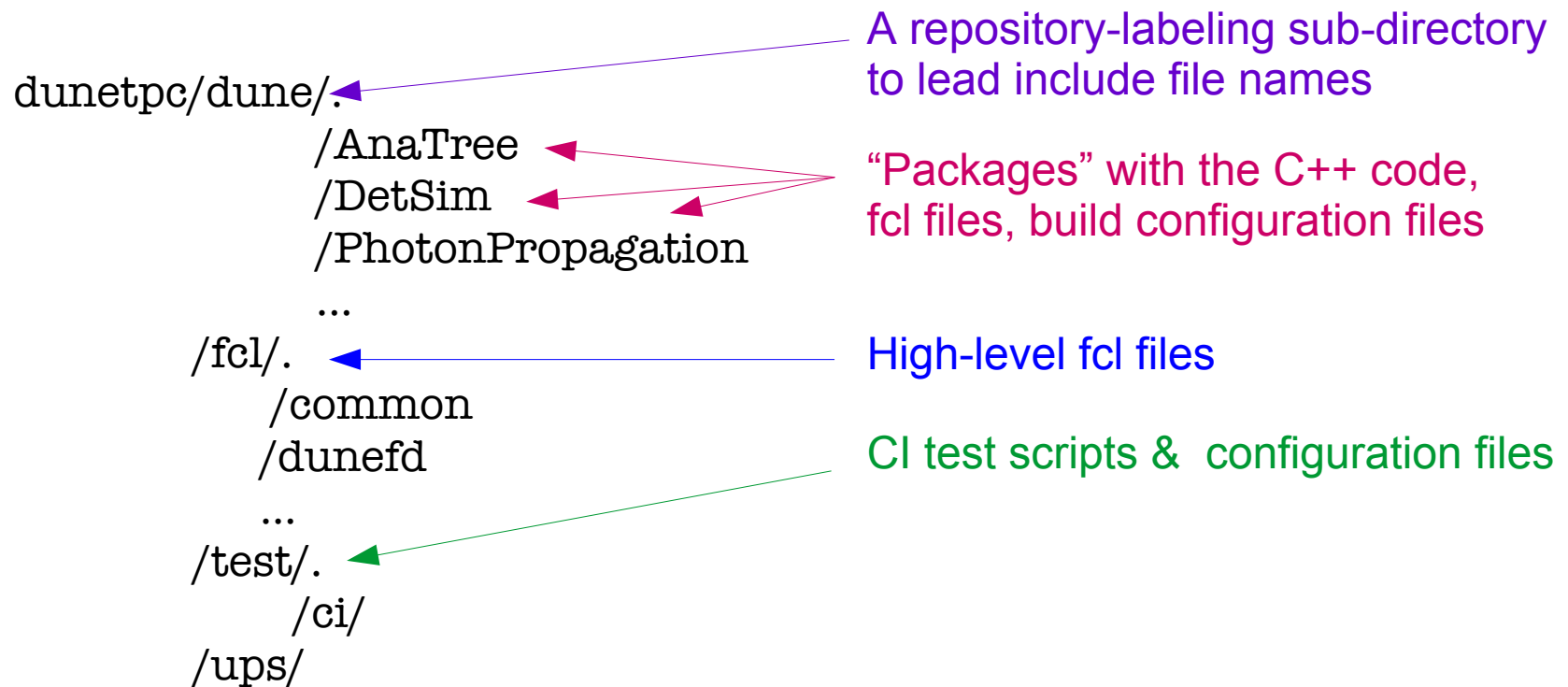
- Another useful command: get rid of the code you just built so you can start over from a clean build

```
cd $MRB_BUILDDIR  
mrb z
```

# Modifying the code of an existing job

Go back to “make changes to the code”. Where is the code??

- Look in `<working_dir>/srcs/<repository-name>`
  - Repositories all have the same basic structure



# Navigating the code

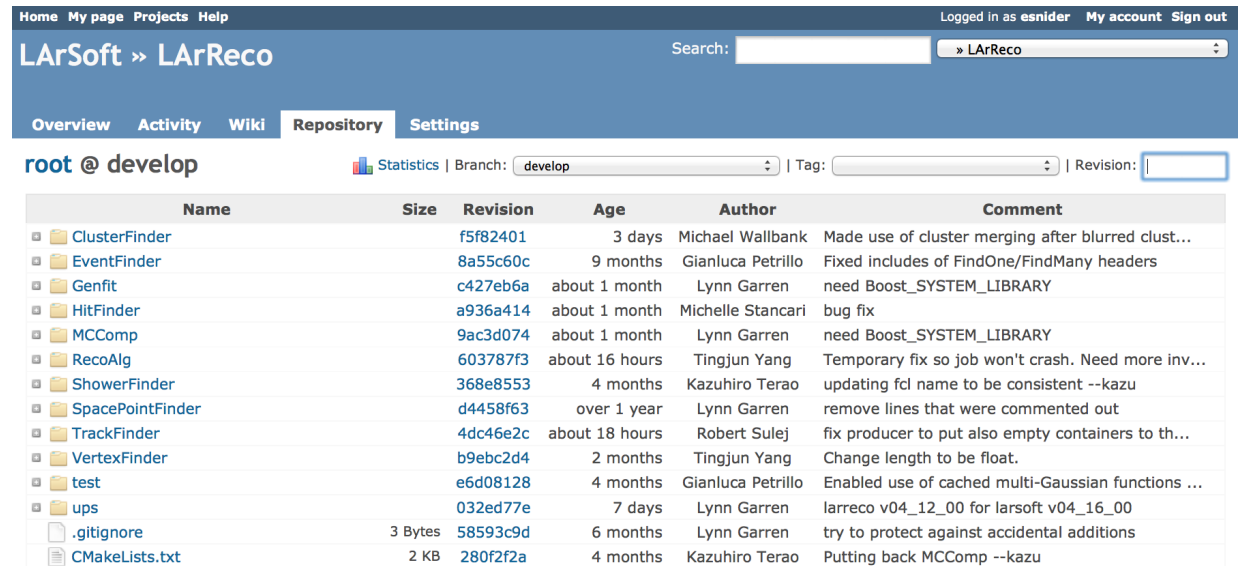
- Redmine code browser
  - <https://cdcvns.fnal.gov/redmine/projects/<repository>/repository>

Knows about branches

Only knows about a single repository at a time

No search

Fairly slow...



Name	Size	Revision	Age	Author	Comment
ClusterFinder		f5f82401	3 days	Michael Wallbank	Made use of cluster merging after blurred clust...
EventFinder		8a55c60c	9 months	Gianluca Petrillo	Fixed includes of FindOne/FindMany headers
Genfit		c427eb6a	about 1 month	Lynn Garren	need Boost_SYSTEM_LIBRARY
HitFinder		a936a414	about 1 month	Michelle Stancari	bug fix
MCComp		9ac3d074	about 1 month	Lynn Garren	need Boost_SYSTEM_LIBRARY
RecoAlg		603787f3	about 16 hours	Tingjun Yang	Temporary fix so job won't crash. Need more inv...
ShowerFinder		368e8553	4 months	Kazuhiro Terao	updating fcl name to be consistent --kazu
SpacePointFinder		d4458f63	over 1 year	Lynn Garren	remove lines that were commented out
TrackFinder		4dc46e2c	about 18 hours	Robert Sulej	fix producer to put also empty containers to th...
VertexFinder		b9ebc2d4	2 months	Tingjun Yang	Change length to be float.
test		e6d08128	4 months	Gianluca Petrillo	Enabled use of cached multi-Gaussian functions ...
ups		032ed77e	7 days	Lynn Garren	larreco v04_12_00 for larsoft v04_16_00
.gitignore	3 Bytes	58593c9d	6 months	Lynn Garren	try to protect against accidental additions
CMakeLists.txt	2 KB	280f2f2a	4 months	Kazuhiro Terao	Putting back MCComp --kazu

## Latest revisions

#	Date	Author	Comment
603787f3	07/22/2015 04:06 PM	Tingjun Yang	Temporary fix so job won't crash. Need more investigation on why this happened.
4dc46e2c	07/22/2015 02:18 PM	Robert Sulej	fix producer to put also empty containers to the event, add one more TTree with info to inspect tracks
77c4e4cd	07/22/2015 02:12 PM	Robert Sulej	add fn to returns mean angle between consecutive segments
7b1b1fcf	07/21/2015 05:03 PM	Robert Sulej	add optio to flip tracks downward
b2e460f0	07/20/2015 04:53 PM	Tingjun Yang	Add protection when TrackTrajectoryAlg does not reconstruct trajectory points.
b2e905c2	07/20/2015 04:52 PM	Tingjun Yang	Merge branch 'develop' of ssh://cdcvns.fnal.gov/cvs/projects/larreco into develop
33890c78	07/20/2015 04:52 PM	Tingjun Yang	Add protection for the case bin is negative. This only happens when divided by a very small number.
85a54d56	07/20/2015 04:27 PM	Bruce Baller	Merge branch 'feature/bb_ccwork' into develop
02a39c7c	07/20/2015 04:25 PM	Bruce Baller	initialize matcomb
8342da80	07/20/2015 04:18 PM	Robert Sulej	add projection of 3D vector to 2D plane in [cm] domain

# Navigating the code

- Ixr Cross Referencer
  - <https://cdcvs.fnal.gov/lxr/<repository>>

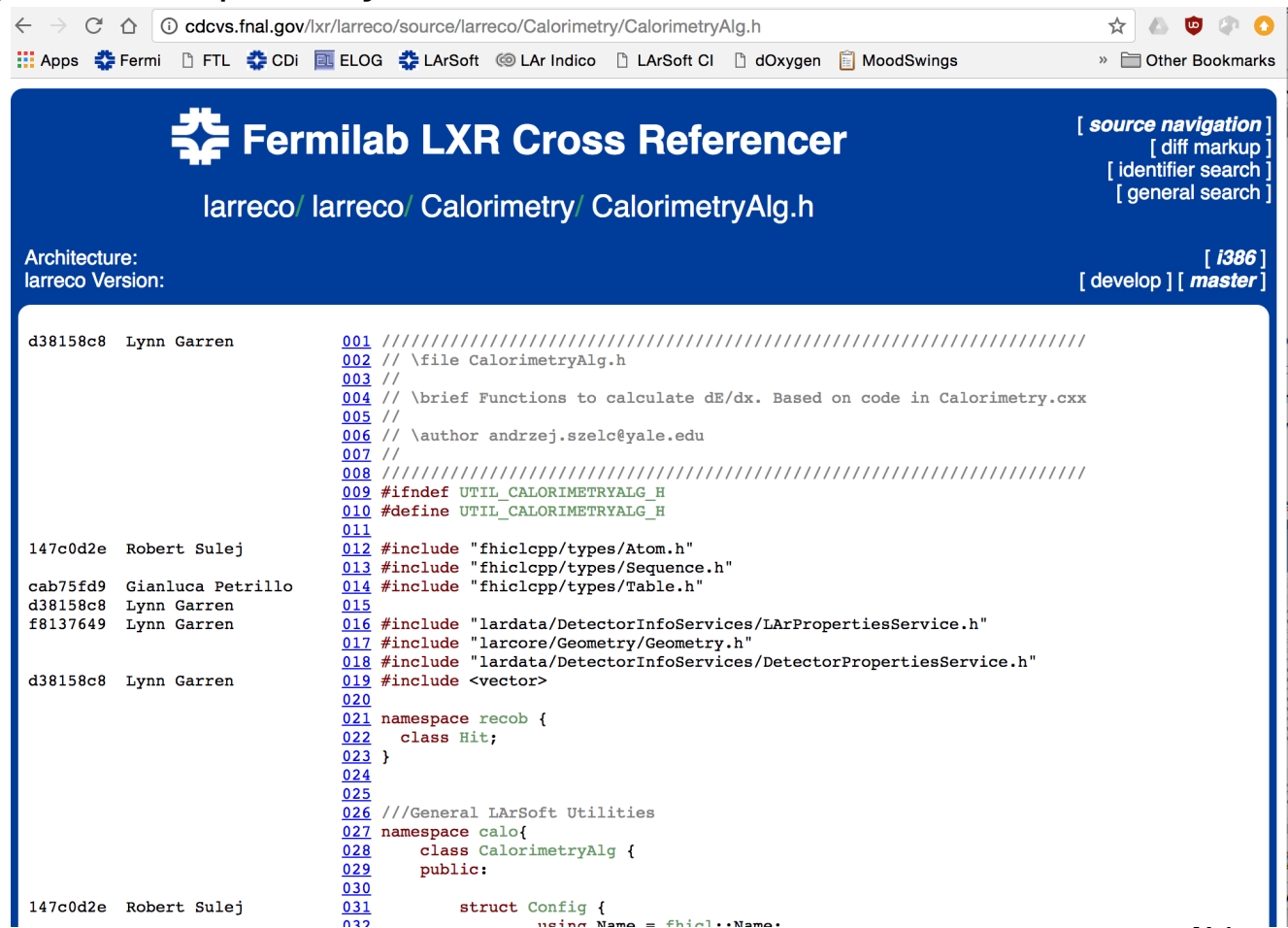
Symbol and string searches across repositories

Links to includes, symbols in the browser window...

...but only within a single repository

Browser window only knows about one repository

Not all searches work across repositories



The screenshot shows the Fermilab LXR Cross Referencer interface for the file `larreco/larreco/Calorimetry/CalorimetryAlg.h`. The interface includes navigation links for source navigation, diff markup, identifier search, and general search. It also shows version information for the repository, including `[i386]`, `[develop]`, and `[master]`.

The code displayed is as follows:

```

d38158c8 Lynn Garren      001 ////////////////////////////////////////////////////////////////////
                          002 // \file CalorimetryAlg.h
                          003 //
                          004 // \brief Functions to calculate dE/dx. Based on code in Calorimetry.cxx
                          005 //
                          006 // \author andrzej.szczelc@yale.edu
                          007 //
                          008 ////////////////////////////////////////////////////////////////////
                          009 #ifndef UTIL_CALORIMETRYALG_H
                          010 #define UTIL_CALORIMETRYALG_H
                          011
147c0d2e Robert Sulej    012 #include "fhiclcpp/types/Atom.h"
cab75fd9 Gianluca Petrillo 013 #include "fhiclcpp/types/Sequence.h"
d38158c8 Lynn Garren      014 #include "fhiclcpp/types/Table.h"
f8137649 Lynn Garren      015
                          016 #include "lardata/DetectorInfoServices/LArPropertiesService.h"
                          017 #include "larcore/Geometry/Geometry.h"
                          018 #include "lardata/DetectorInfoServices/DetectorPropertiesService.h"
d38158c8 Lynn Garren      019 #include <vector>
                          020
                          021 namespace recob {
                          022   class Hit;
                          023 }
                          024
                          025
                          026 ///General LArSoft Utilities
                          027 namespace calo{
                          028   class CalorimetryAlg {
                          029   public:
                          030
147c0d2e Robert Sulej    031       struct Config {
                          032           using Name = fhicl::Name;
  
```

# Navigating the code

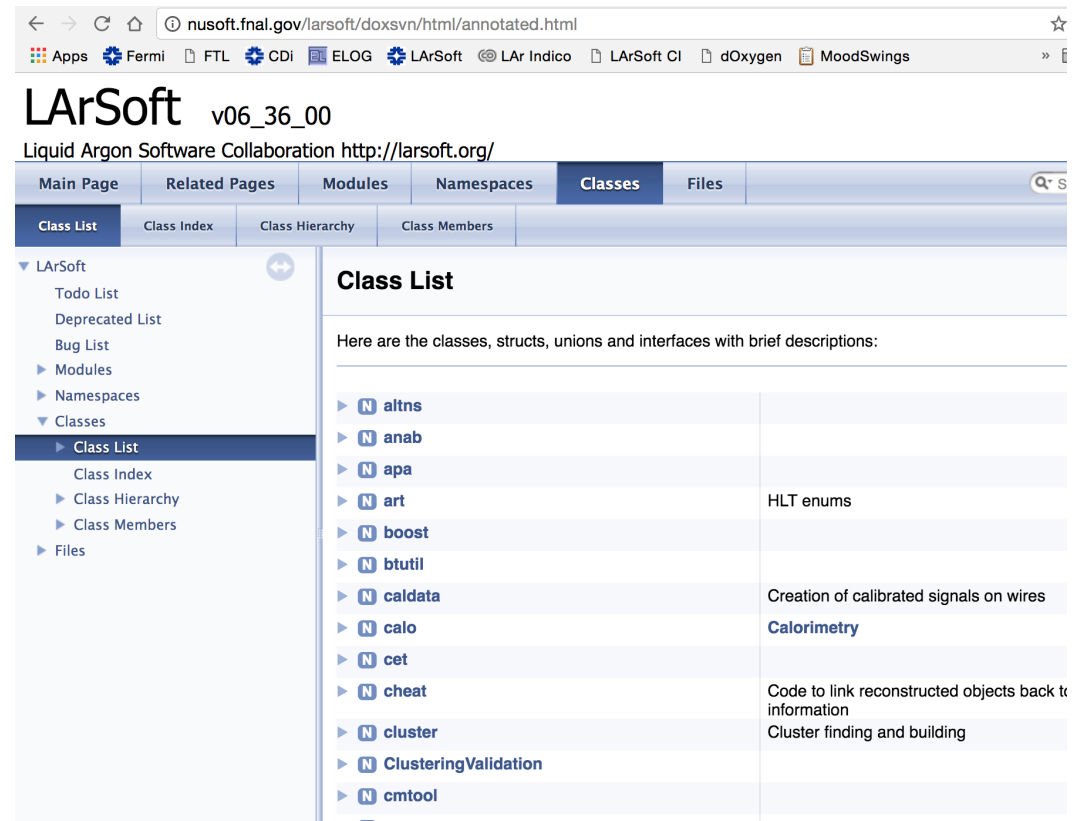
- Doxygen
  - <http://nusoft.fnal.gov/larsoft/doxsvn/html/index.html>

Self-documenting via markup tokens embedded in C++ comments

Understands C++ class structure, namespaces

Searches

Not very good at navigating directory hierarchies – does not really know about repositories



The screenshot shows a web browser displaying the LArSoft Doxygen documentation. The page title is "LArSoft v06\_36\_00" and the subtitle is "Liquid Argon Software Collaboration <http://larsoft.org/>". The navigation menu includes "Main Page", "Related Pages", "Modules", "Namespaces", "Classes", and "Files". The "Classes" menu is active, and the "Class List" sub-menu is selected. The main content area displays a "Class List" with a table of classes, structs, unions, and interfaces. The table has two columns: the class name and a brief description. The classes listed are: `aitns`, `anab`, `apa`, `art` (HLT enums), `boost`, `btutil`, `caldata` (Creation of calibrated signals on wires), `calo` (Calorimetry), `cet`, `cheat` (Code to link reconstructed objects back to information), `cluster` (Cluster finding and building), `ClusteringValidation`, and `cmtool`.

## Navigating the code

- Check out everything into local directory + grep
  - Super simple
  - Fast
  - Can pipe commands together to do interesting things
    -
  - Does not know anything about C++
  - Extremely inelegant

...I do this, and I know I'm not the only one...



# Contributing code to LArSoft










- LArSoft collaboration thrives because code is shared
- In order to ensure
  - stable development environment
  - code is interoperable
  - people are made aware of changes in behavior, major design choices
  - etc.

there are some coding guidelines, design principles and practices, procedures to follow

- See “Developing with LArSoft” wiki page
  - “LArSoft architecture and design principles” section
  - “Writing code” section
  - The git branching model that we use

To commit code, need to be a “developer” in redmine: Talk to Tom Junk, Andrew Norman, Tingjun Yang

# Source code documentation standard

- In every header file, module, and service file, include:
  - File  Use @file and @class Doxygen markup for these
  - Class  Use @file and @class Doxygen markup for these
  - Description  Purpose of the class, what it does, instructions for using it. If appropriate, general description of algorithm. Point is to make it concise.
  - Inputs  Most important for modules, since these are fixed, but not obvious without looking deep into the implementation. Need to list all fcl parameters, what they do, possible values
  - Outputs  Most important for modules, since these are fixed, but not obvious without looking deep into the implementation. Need to list all fcl parameters, what they do, possible values
  - Configuration options  These are rarely obvious without reading code, but are critical for users to know. Could be in the “description” section, but best to make them easy to see
  - Assumptions / pre-requisites  These are rarely obvious without reading code, but are critical for users to know. Could be in the “description” section, but best to make them easy to see
  - Original author / date  Date + who + one/two line summary of significant changes (can't get this with git)
  - Revision history  Date + who + one/two line summary of significant changes (can't get this with git)

# Source code documentation standard

- In every header file, module, and service file, include:

- File
- Class
- Description
- Inputs
- Outputs
- Configuration
- Assumptions
- Original author / date
- Revision history

The goal is to allow someone to use the code, so only need to write as much as needed for that

Detailed description of methods should be included with the method prototype in the class definition (with Doxygen markup)

Detailed comments on the implementation should be embedded in the code

# Navigating the contents of art/root files

- `lar -c eventdump.fcl -s <file>`
  - Uses the FileDumperOutput module to produce this:

```
Begin processing the 1st record. run: 20000014 subRun: 0 event: 1 at 17-May-2017 01:59:11 CDT
PRINCIPAL TYPE: Event
PROCESS NAME | MODULE_LABEL... | PRODUCT_INSTANCE_NAME | DATA_PRODUCT_TYPE... | SIZE
SinglesGen.. | generator..... | ..... | std::vector<simb::MCTruth>..... | ...1
SinglesGen.. | rns..... | ..... | std::vector<art::RNGsnapshot>..... | ...1
SinglesGen.. | TriggerResults | ..... | art::TriggerResults..... | ...-
G4..... | largeant..... | ..... | std::vector<sim::OpDetBacktrackerRecord>..... | ..99
G4..... | rns..... | ..... | std::vector<art::RNGsnapshot>..... | ...2
G4..... | TriggerResults | ..... | art::TriggerResults..... | ...-
G4..... | largeant..... | ..... | std::vector<simb::MCParticle>..... | ...8
G4..... | largeant..... | ..... | std::vector<sim::AuxDetSimChannel>..... | ...0
G4..... | largeant..... | ..... | art::Assns<simb::MCTruth,simb::MCParticle,void>..... | ...8
G4..... | largeant..... | ..... | std::vector<sim::SimChannel>..... | ..684
G4..... | largeant..... | ..... | std::vector<sim::SimPhotonsLite>..... | ..99
Detsim..... | TriggerResults | ..... | art::TriggerResults..... | ...-
Detsim..... | opdigi..... | ..... | std::vector<raw::OpDetWaveform>..... | ..582
Detsim..... | daq..... | ..... | std::vector<raw::RawDigit>..... | 4148
Detsim..... | rns..... | ..... | std::vector<art::RNGsnapshot>..... | ...1
Reco..... | TriggerResults | ..... | art::TriggerResults..... | ...-
Reco..... | trajcluster... | ..... | std::vector<recob::Vertex>..... | ...2
Reco..... | pmtrajfit..... | kink..... | std::vector<recob::Vertex>..... | ...0
Reco..... | pandora..... | ..... | std::vector<recob::PCAxis>..... | ...0
Reco..... | pmtrack..... | ..... | std::vector<recob::Vertex>..... | ...2
Reco..... | pandoracalo... | ..... | art::Assns<recob::Track,anab::Calorimetry,void>..... | ...3
Reco..... | pandora..... | ..... | art::Assns<recob::PFParticle,recob::SpacePoint,void>..... | ..581
...
...
```

# Navigating the contents of art/root files

- Examine the file within a root TBrowser

The screenshot shows the ROOT Object Browser interface. On the left, a file tree displays the structure of a ROOT file. A pink arrow points to the 'Events;1' folder, labeled 'The event TTree'. Blue arrows point to various sub-branches under 'Events;1', labeled 'Data product branches'. On the right, a histogram plot is displayed, showing a single sharp peak. The plot title is 'simb::MCTruths\_generator\_\_SinglesGen.obj.fMCNeutrino.fNu.fmass'. A statistics box in the top right corner of the plot shows: 'htemp', 'Entries: 1', 'Mean: -2.147e+09', and 'Std Dev: 0'. The x-axis is labeled with values from -214742096 to 4836468, with a multiplier of  $\times 10^6$ .

# Navigating the contents of art/root files

- Dumping individual data products

`lardata/lardata/ArtDataHelpers/Dumper`

`larsim/larsim/MCDumpers`

- Dedicated modules named “Dump<data product>” produce formatted dump of contents of that data product
- Run then with fcl files in those same directories: `dump_<data type>.fcl`
- E.g.: `lar -c dump_clusters.fcl -s <file>`

# Resources and support

## Using LArSoft off-site

- Two options for using LArSoft off-site

- 1) Use cvmfs

- CernVM file system: <https://cernvm.cern.ch/portal/filesystem>
- http-based virtual file system with local caching
- LArSoft code distributed within the US on [/cvmfs/fermilab.opensciencegrid.org/products/larsoft](https://cvmfs/fermilab.opensciencegrid.org/products/larsoft)
  - Also a CERN stratum 1 service?? (..not sure...need to check...)
- [cvmfs client installation instructions on LArSoft wiki](#)

By far the easiest method to get going.

- 2) Local installation or build

- Instructions in release notes. [See LArSoft releases wiki page](#)
- Requires manual upkeep to stay current



# Supported platforms

See [https://cdcv.s.fnal.gov/redmine/projects/larsoft/wiki/\\_Supported\\_platforms\\_](https://cdcv.s.fnal.gov/redmine/projects/larsoft/wiki/_Supported_platforms_)

- Scientific Linux
  - SLF6 (the reference system) + SLF7
    - Should work on any SL variant; Works on SLC6 (CERN), Redhat 6 (SLAC)
  - SLF7
    - Binary distributions only - no cvmfs installation
- Mac OSX
  - Yosemite
    - ups qualifier d14
  - “Known to work” on El Capitan and Sierra
    - Must disable SIP and install openssl. Qualifiers d15 and d16, respectively
- Ubuntu
  - “Known to work” with Ubuntu14 and 16
    - LArSoft team distributes installation tarballs for u16 (best effort now, but working to make it “supported”)
    - Ubuntu 14 available only by request

## Installation instructions:

See links in release notes available at  
[https://cdcv.s.fnal.gov/redmine/projects/larsoft/wiki/LArSoft\\_release\\_list](https://cdcv.s.fnal.gov/redmine/projects/larsoft/wiki/LArSoft_release_list)

# Resources

- **Main public LArSoft web page:** <https://larsoft.org>
  - Basic concepts, training materials, pointers to other resources
- LArSoft wiki: <https://cdcvs.fnal.gov/redmine/projects/larsoft/wiki>
  - Quick page with links to quick-start guides by experiment
- [www.larforum.org](http://www.larforum.org)
  - A forum to discuss LArTPC software
- LArSoft email list: [larsoft@fnal.gov](mailto:larsoft@fnal.gov)
  - General announcements. Some technical questions too.
  - Can self-subscribe. See <http://listserv.fnal.gov/> for instructions.
- **LArSoft Coordination Meeting**
  - Bi-weekly at 09:00 Central Time in WH3NE
  - Remote connections via Zoom. Slides, notes posted to [LArSoft Indico site](#).
- LArSoft issue tracker:
  - <https://cdcvs.fnal.gov/redmine/projects/larsoft/issues/new>
- 2015 LArSoft course material
  - <https://indico.fnal.gov/conferenceTimeTable.py?confId=9928#20150807>

# Core LArSoft support team

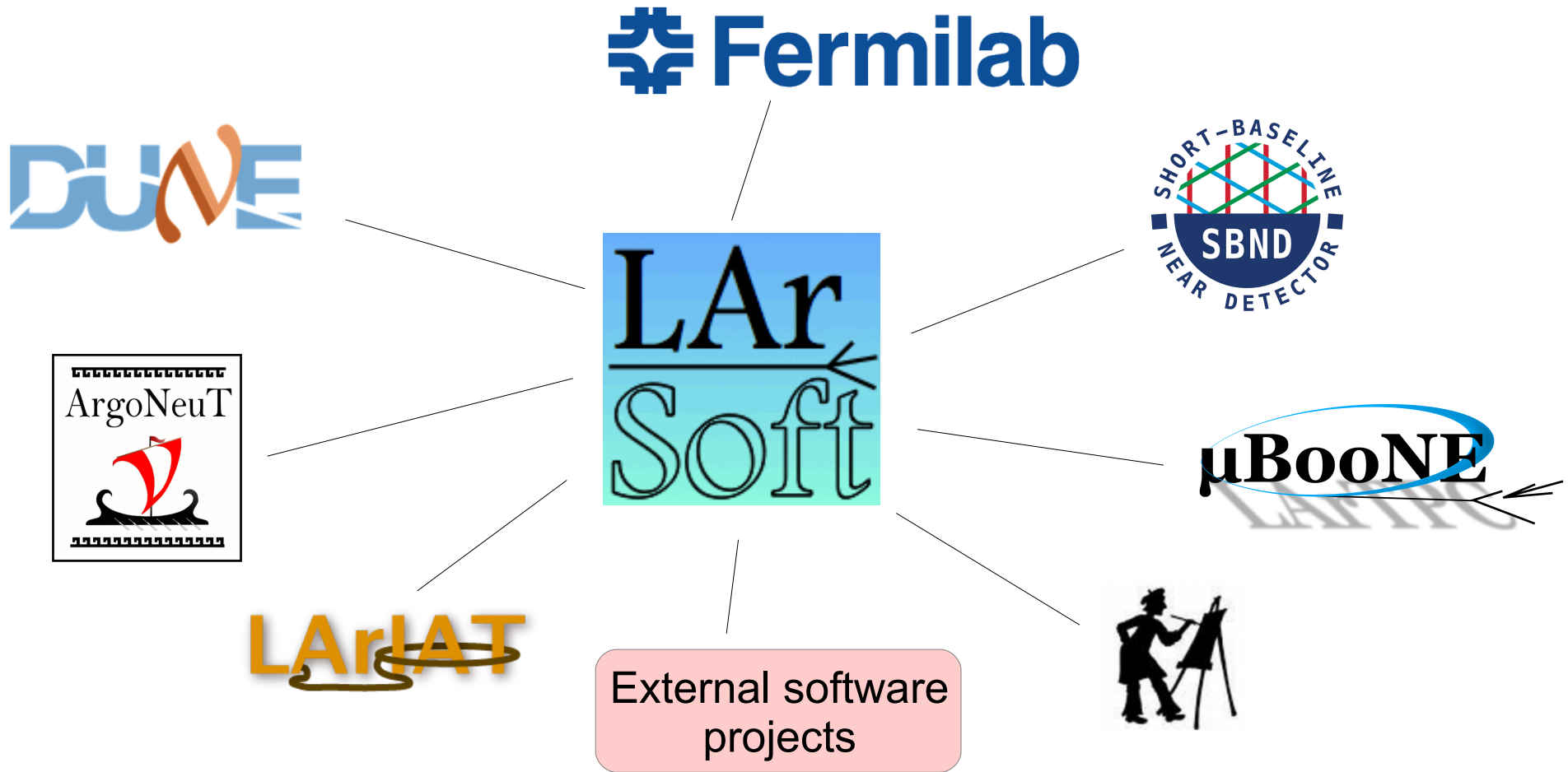
- Core team members

- Technical lead: Erica Snider  
[erica@fnal.gov](mailto:erica@fnal.gov)
- Project manager: Katherine Lato  
[klato@fnal.gov](mailto:klato@fnal.gov)
- Lead developer: Gianluca Petrillo  
[petrillo@fnal.gov](mailto:petrillo@fnal.gov)
- Developers: Giuseppe Cerati  
[cerati@fnal.gov](mailto:cerati@fnal.gov)  
Saba Sehrish  
[ssehrish@fnal.gov](mailto:ssehrish@fnal.gov)
- Code management and distribution: Lynn Garren  
[garren@fnal.gov](mailto:garren@fnal.gov)
- CI operations and testing support: Vito di Benedetto  
[vito@fnal.gov](mailto:vito@fnal.gov)
- Documentation: Katherine Lato

The end

# Backup

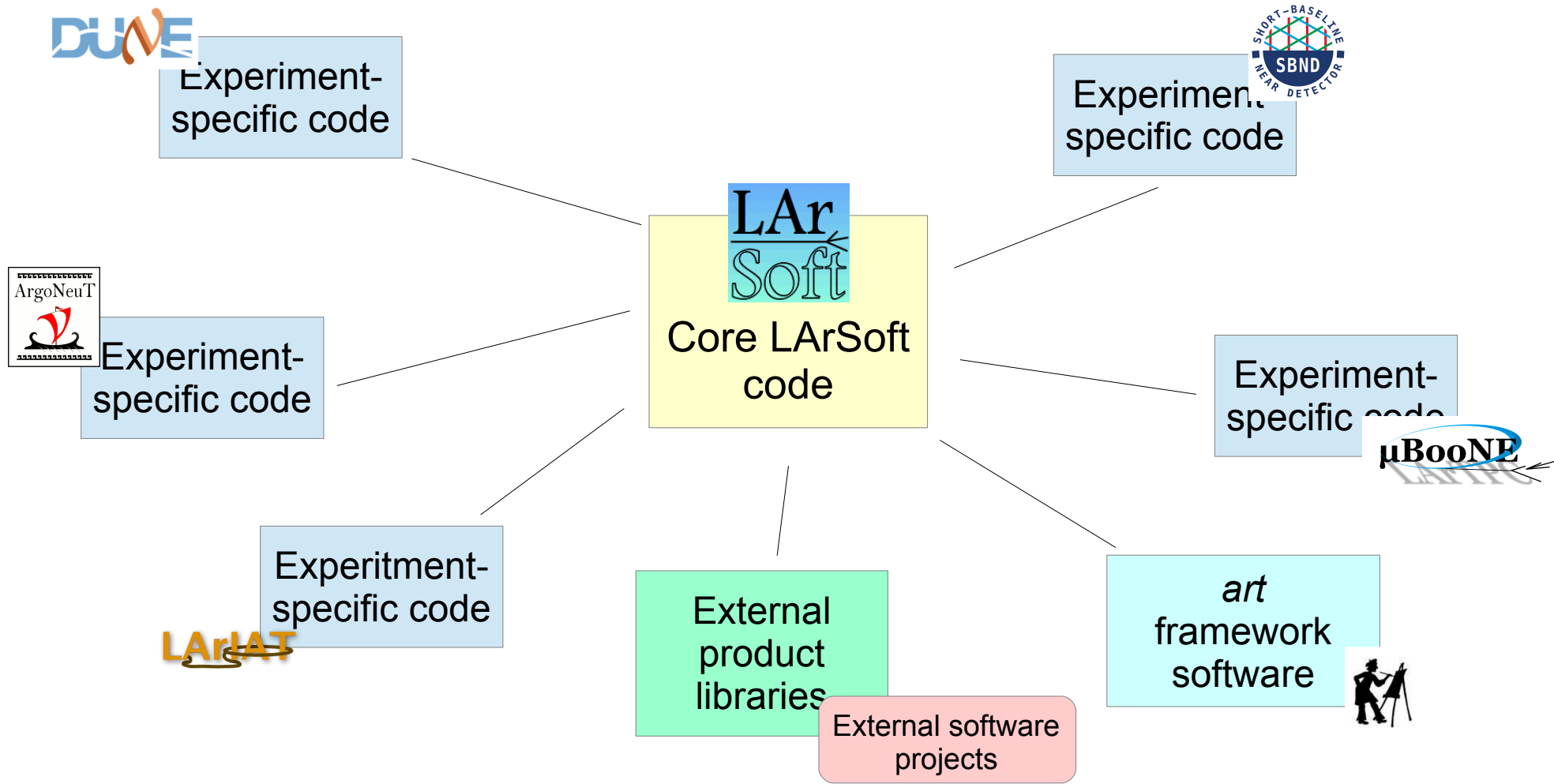
**What is LArSoft?**: (1) **A collaboration** of experiments, Fermilab, other stakeholders



To provide integrated, experiment-independent software tools for LAr TPC neutrino experiments to perform simulation, reconstruction analysis.



**What is LArSoft?:** (2) **A body of code** that interfaces with experiment-specific, *art* framework and external product software



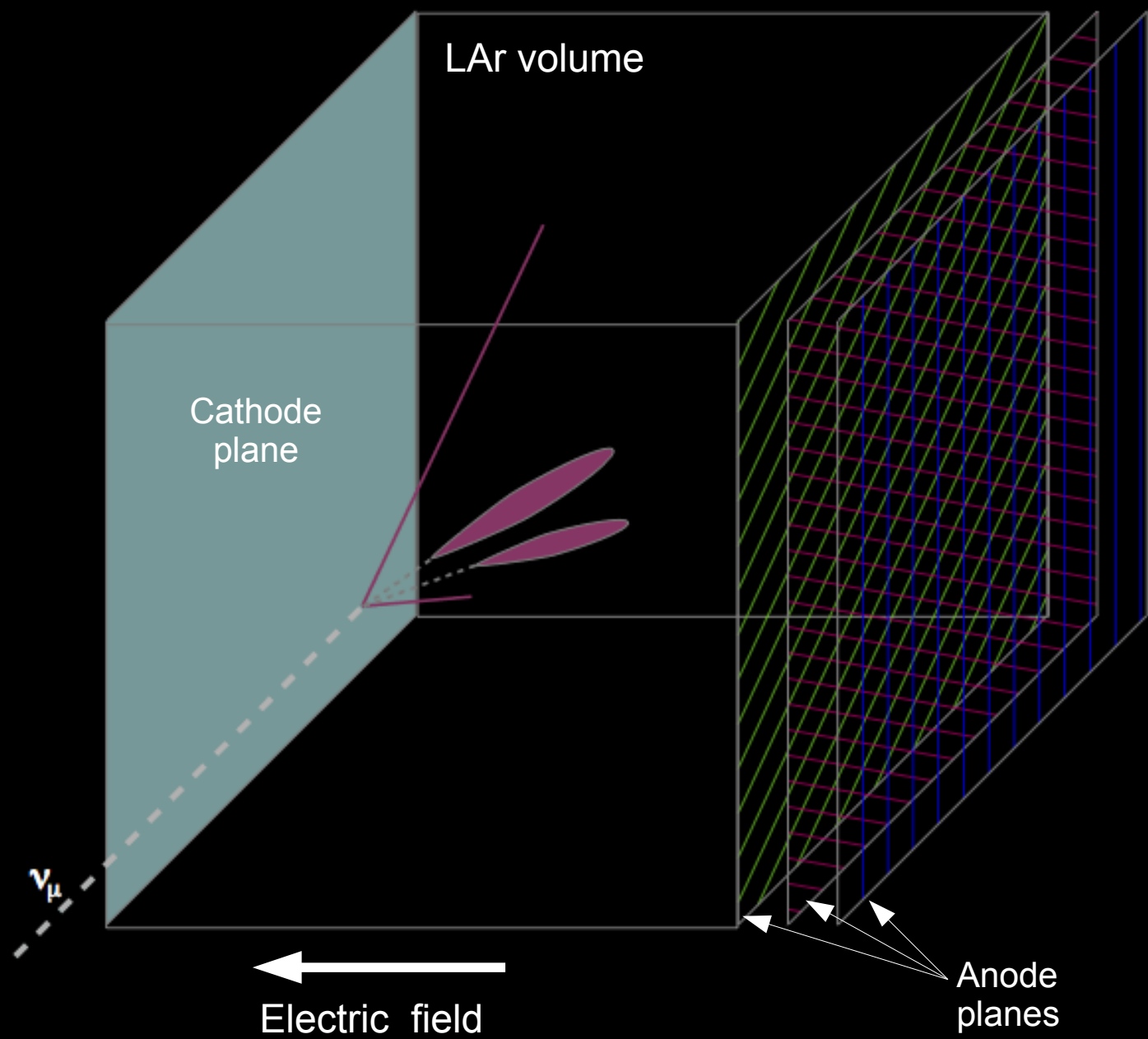
Experiments contribute common “core” LArSoft code

290k lines of C++ in core LArSoft  
440k+ lines including expt code

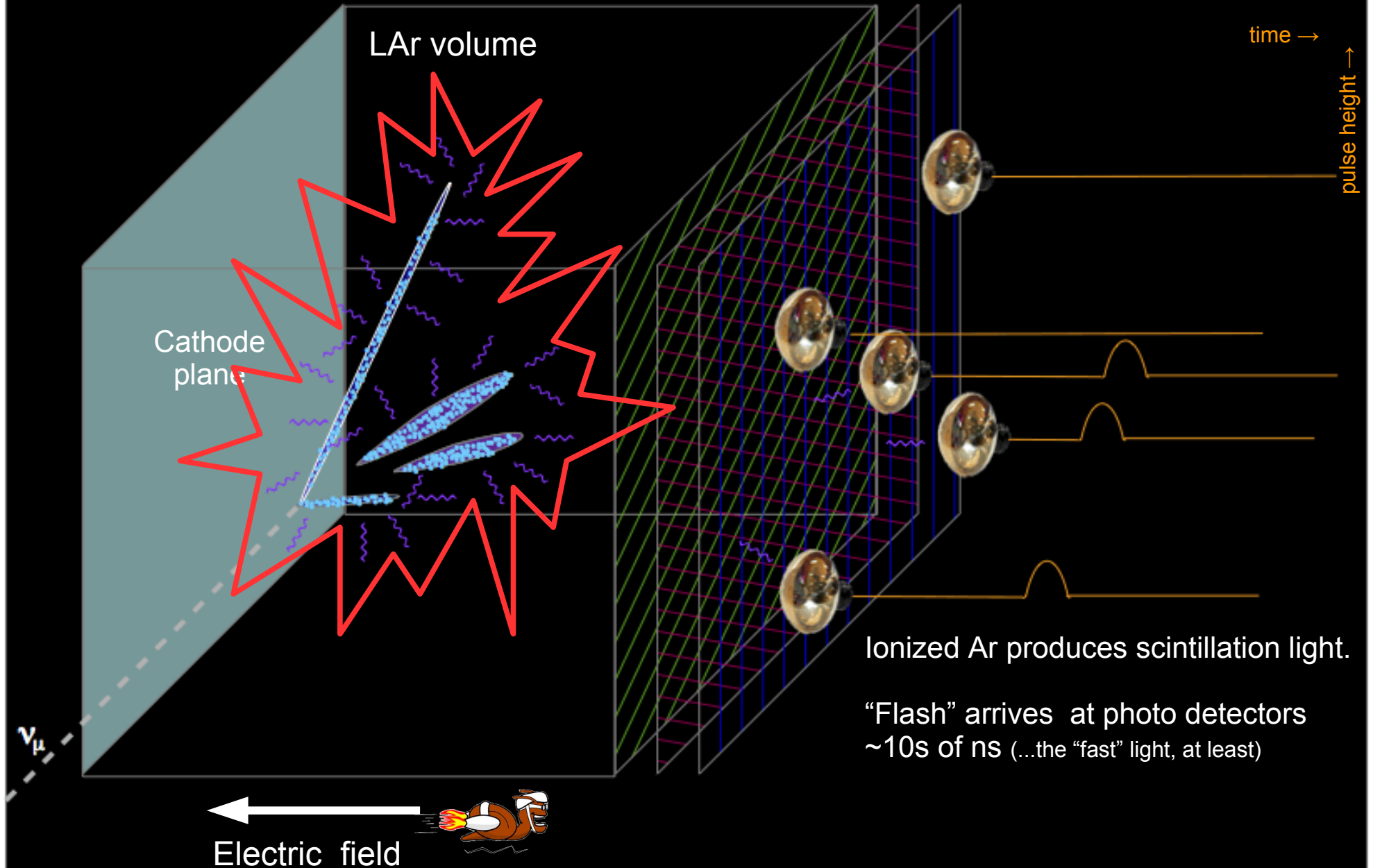


# Operation of a single-phase LAr TPC

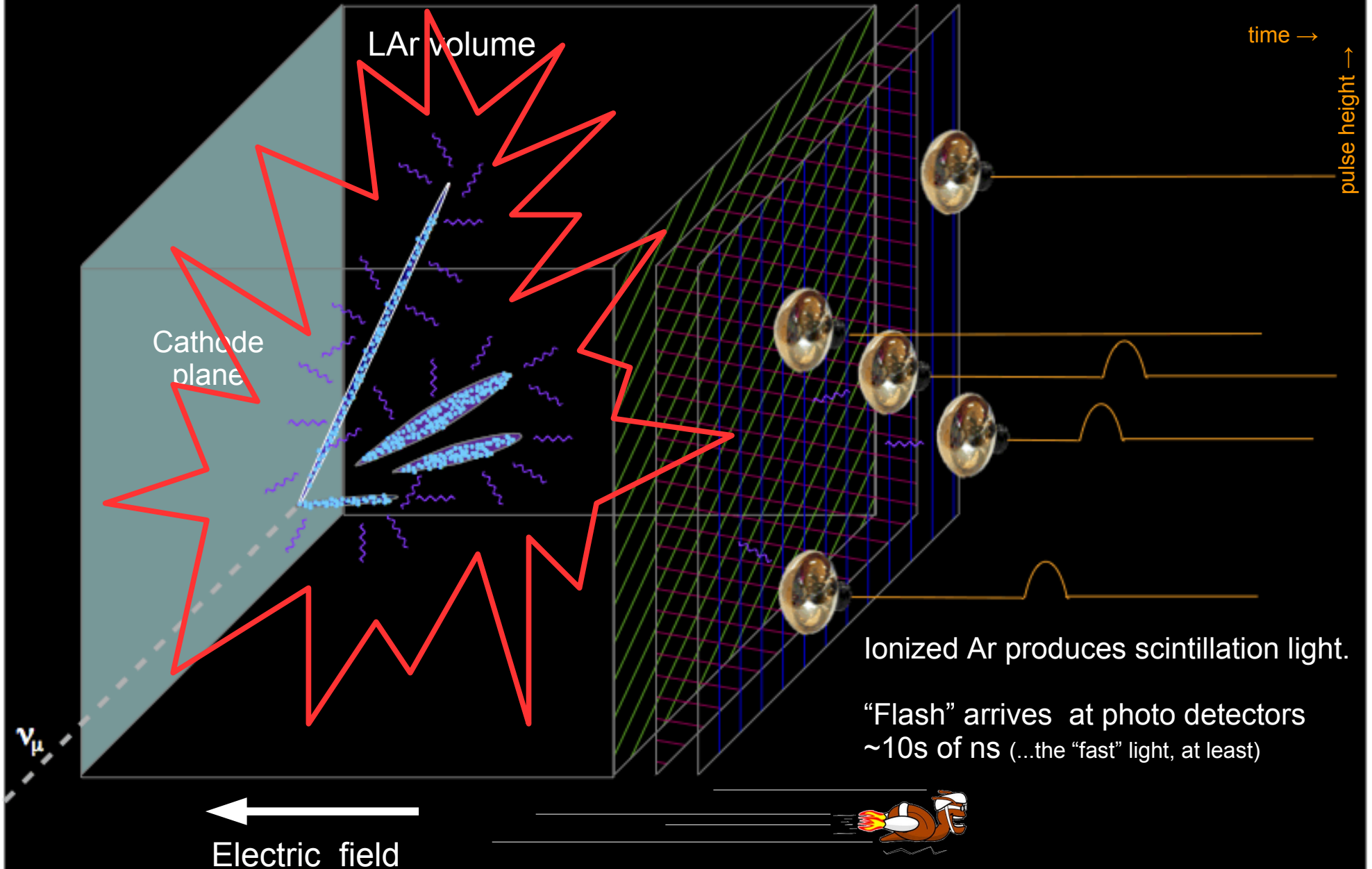
# Operation of single-phase LAr TPC



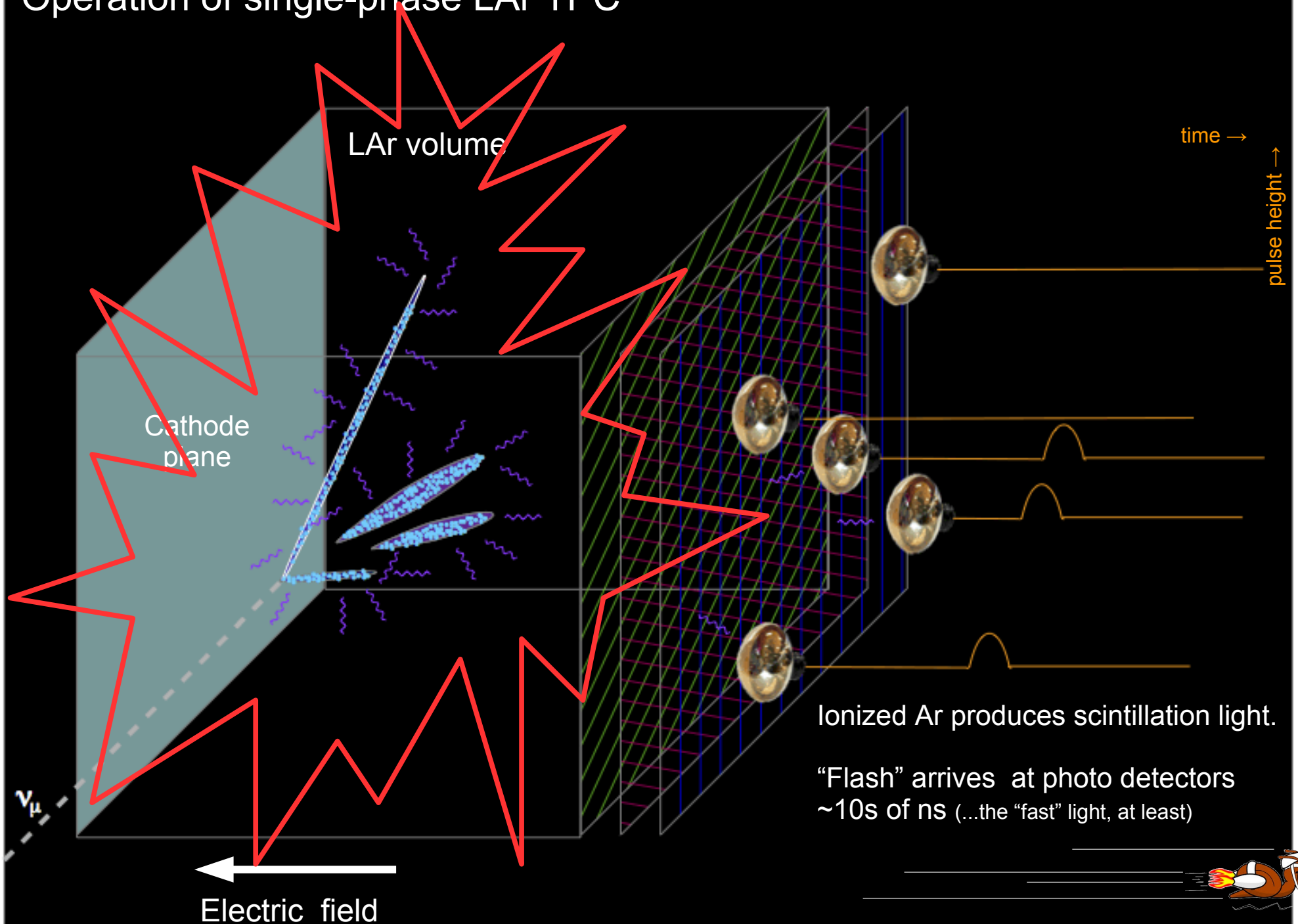
# Operation of single-phase LAr TPC



# Operation of single-phase LAr TPC



# Operation of single-phase LAr TPC



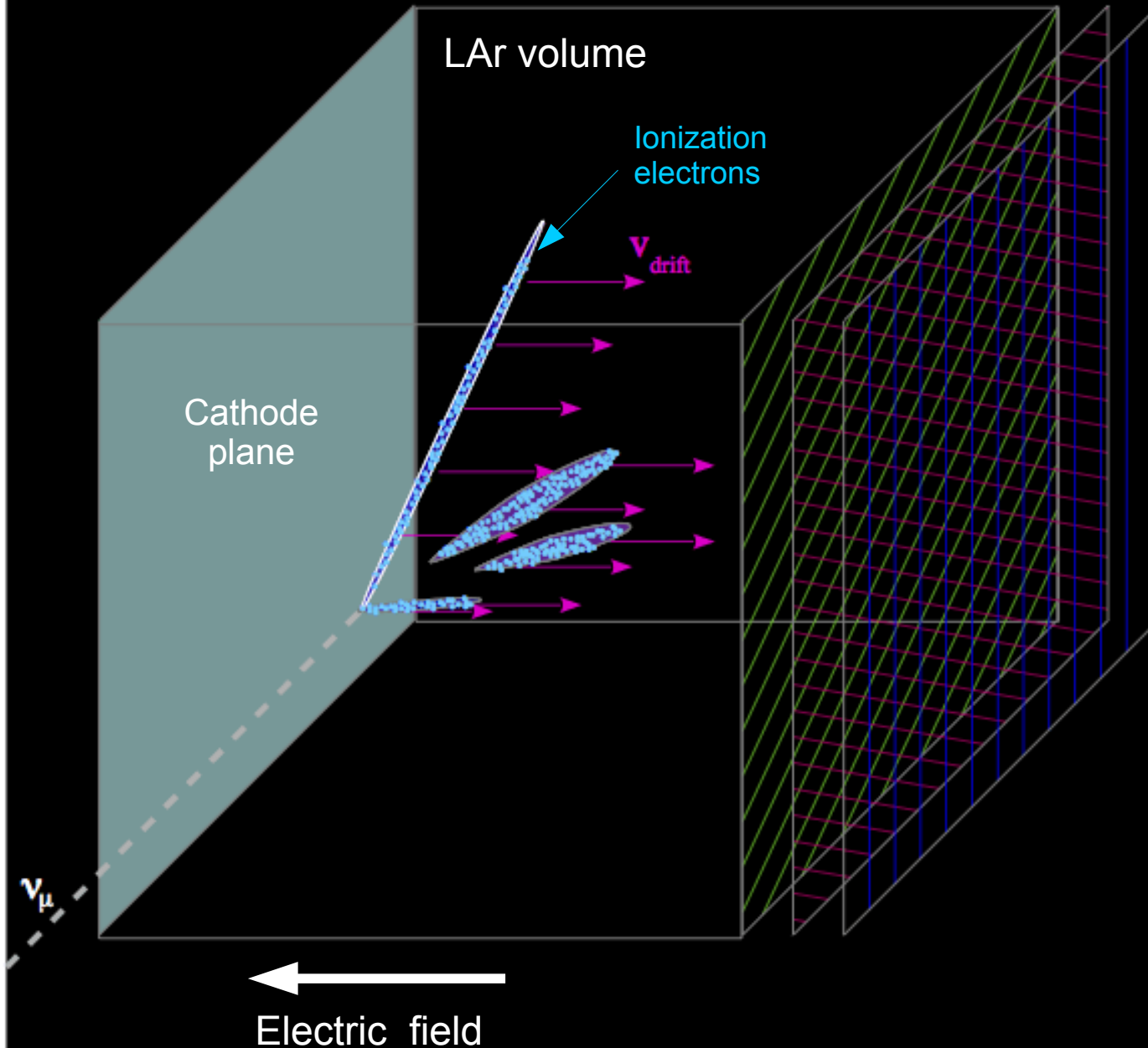
time →  
pulse height →

Ionized Ar produces scintillation light.

"Flash" arrives at photo detectors  
~10s of ns (...the "fast" light, at least)



# Operation of single-phase LAr TPC



Neutrino interacts with Ar nucleus

Charged secondaries ionize the Ar

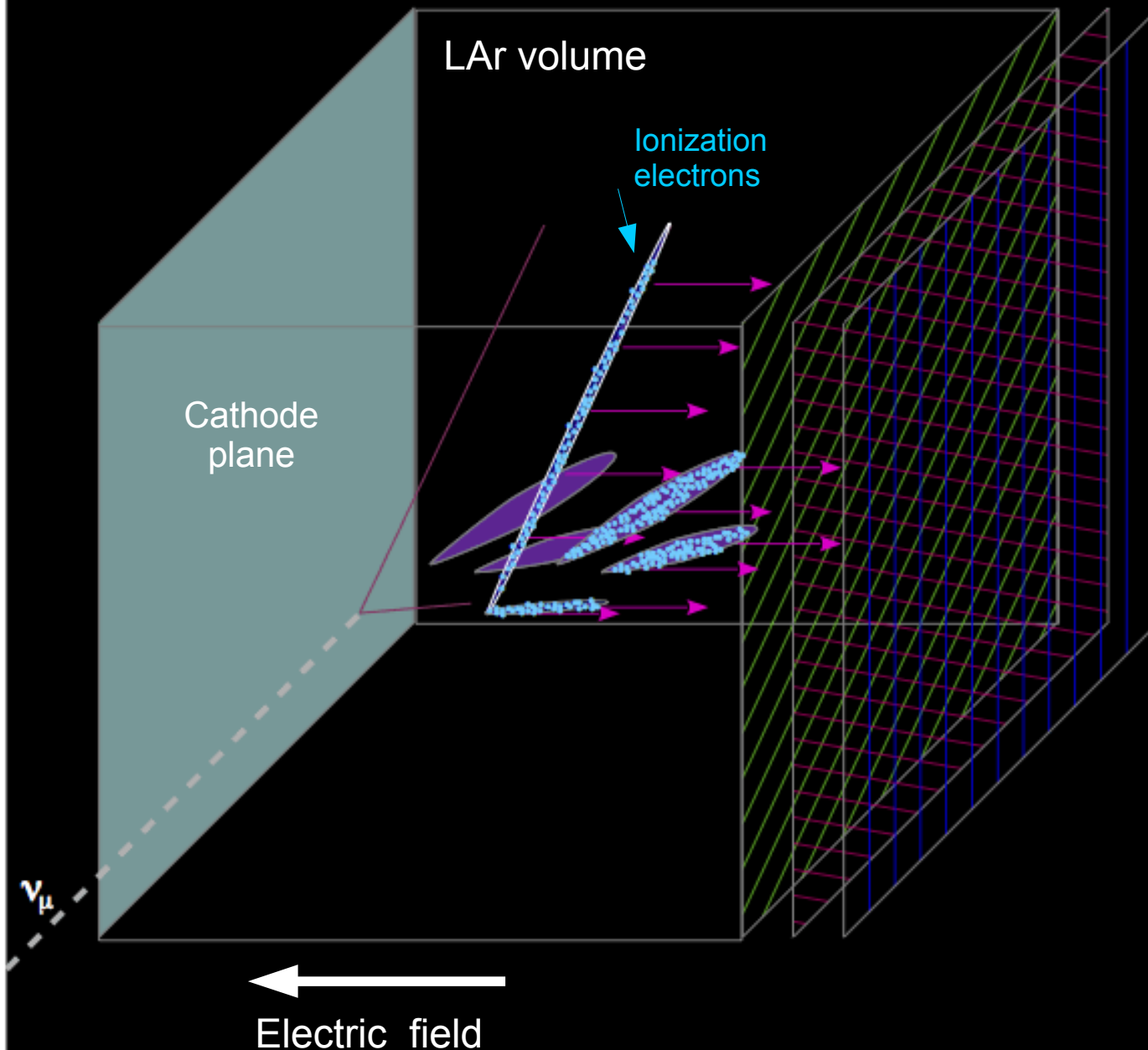
Electrons drift in the electric field toward anode wires

$$v_{drift} \approx 1 - \text{few mm}/\mu\text{s}$$



Max drift time ~ ms!!

# Operation of single-phase LAr TPC



Neutrino interacts with Ar nucleus

Charged secondaries ionize the Ar

Electrons drift in the electric field toward anode wires

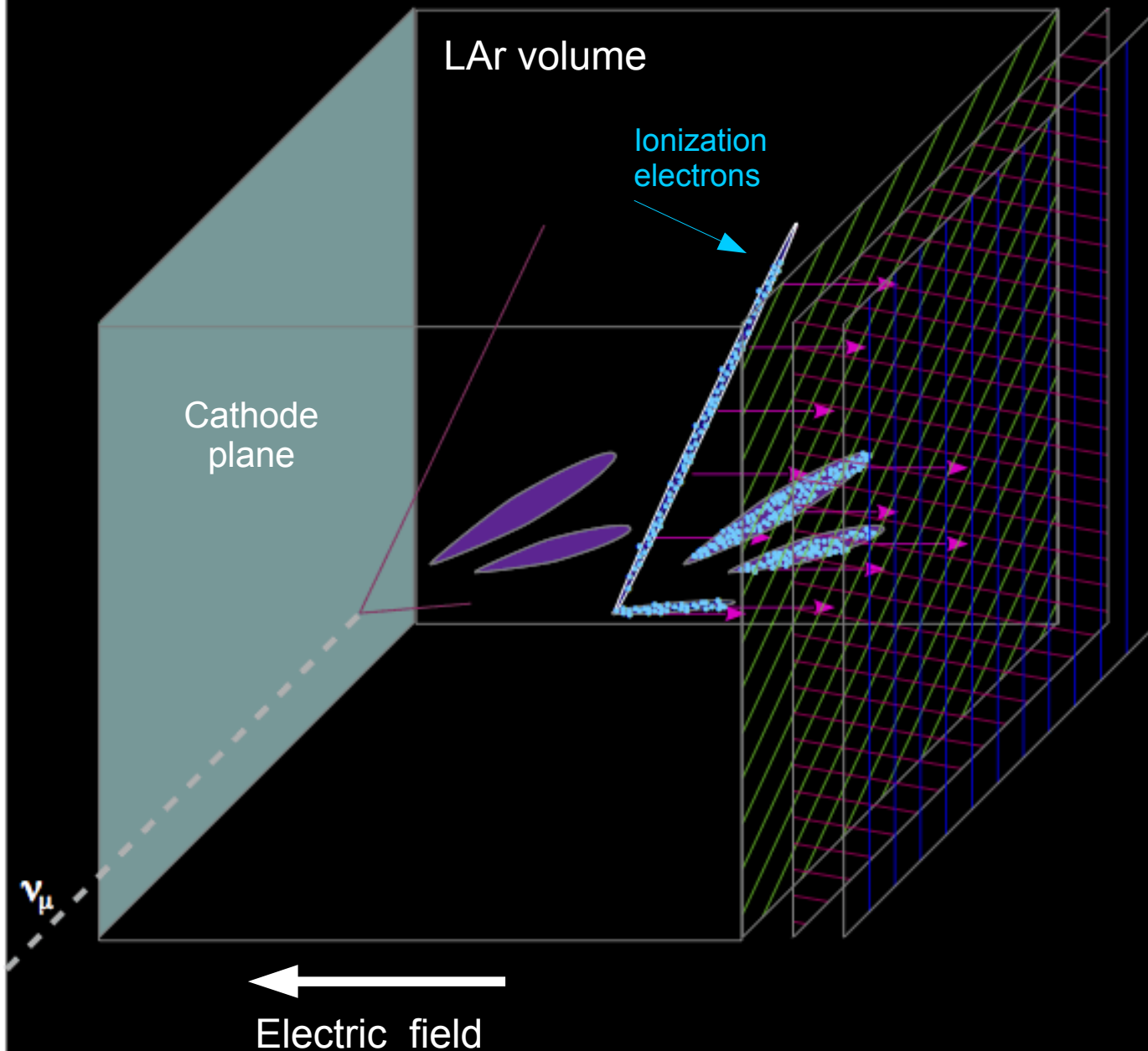
$$v_{\text{drift}} \approx 1 - \text{few mm}/\mu\text{s}$$



Max drift time ~ ms!!



# Operation of single-phase LAr TPC



Neutrino interacts with Ar nucleus

Charged secondaries ionize the Ar

Electrons drift in the electric field toward anode wires

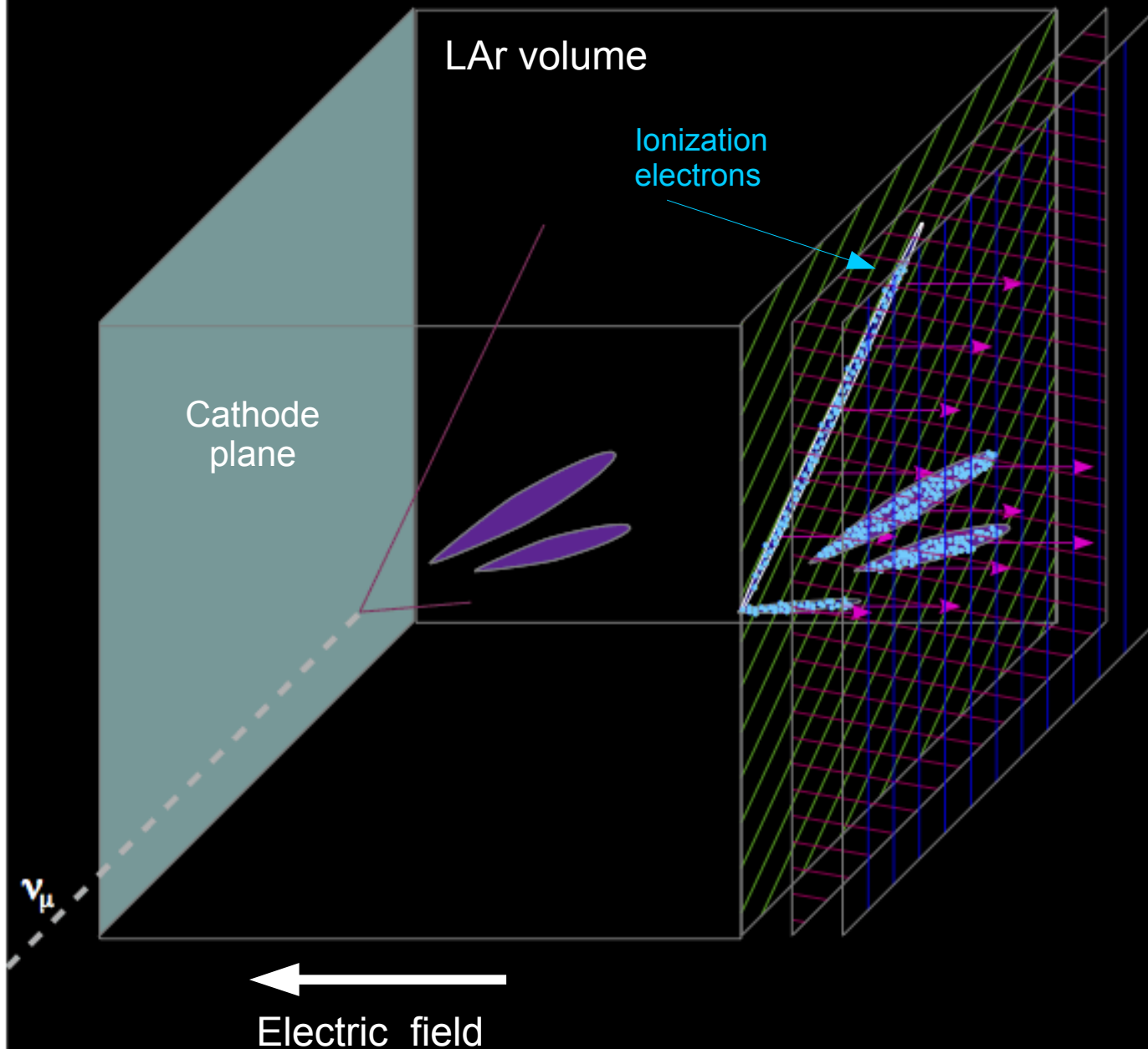
$$v_{\text{drift}} \approx 1 - \text{few mm}/\mu\text{s}$$



Max drift time ~ ms!!



# Operation of single-phase LAr TPC



Neutrino interacts with Ar nucleus

Charged secondaries ionize the Ar

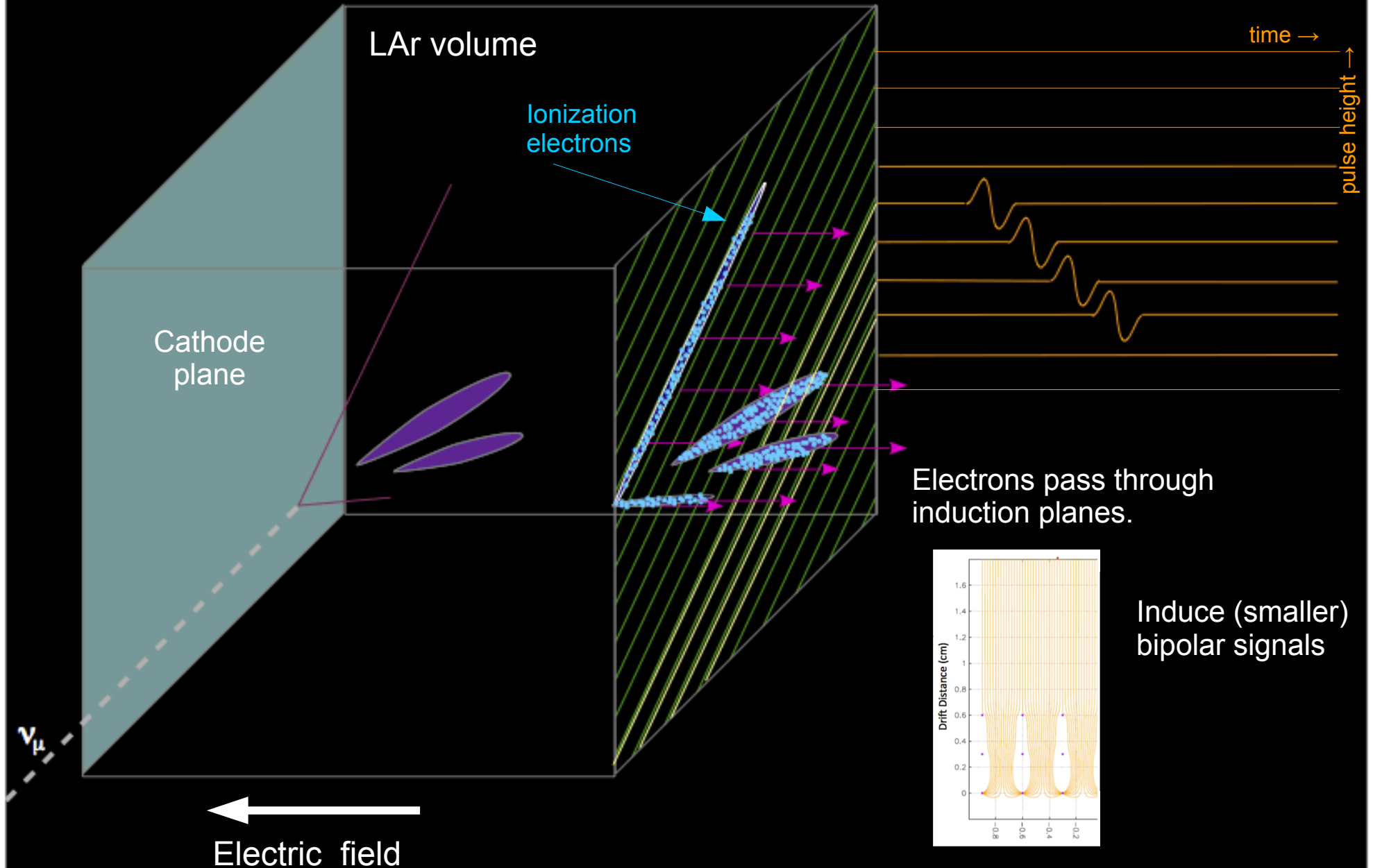
Electrons drift in the electric field toward anode wires

$$v_{\text{drift}} \approx 1 - \text{few mm}/\mu\text{s}$$

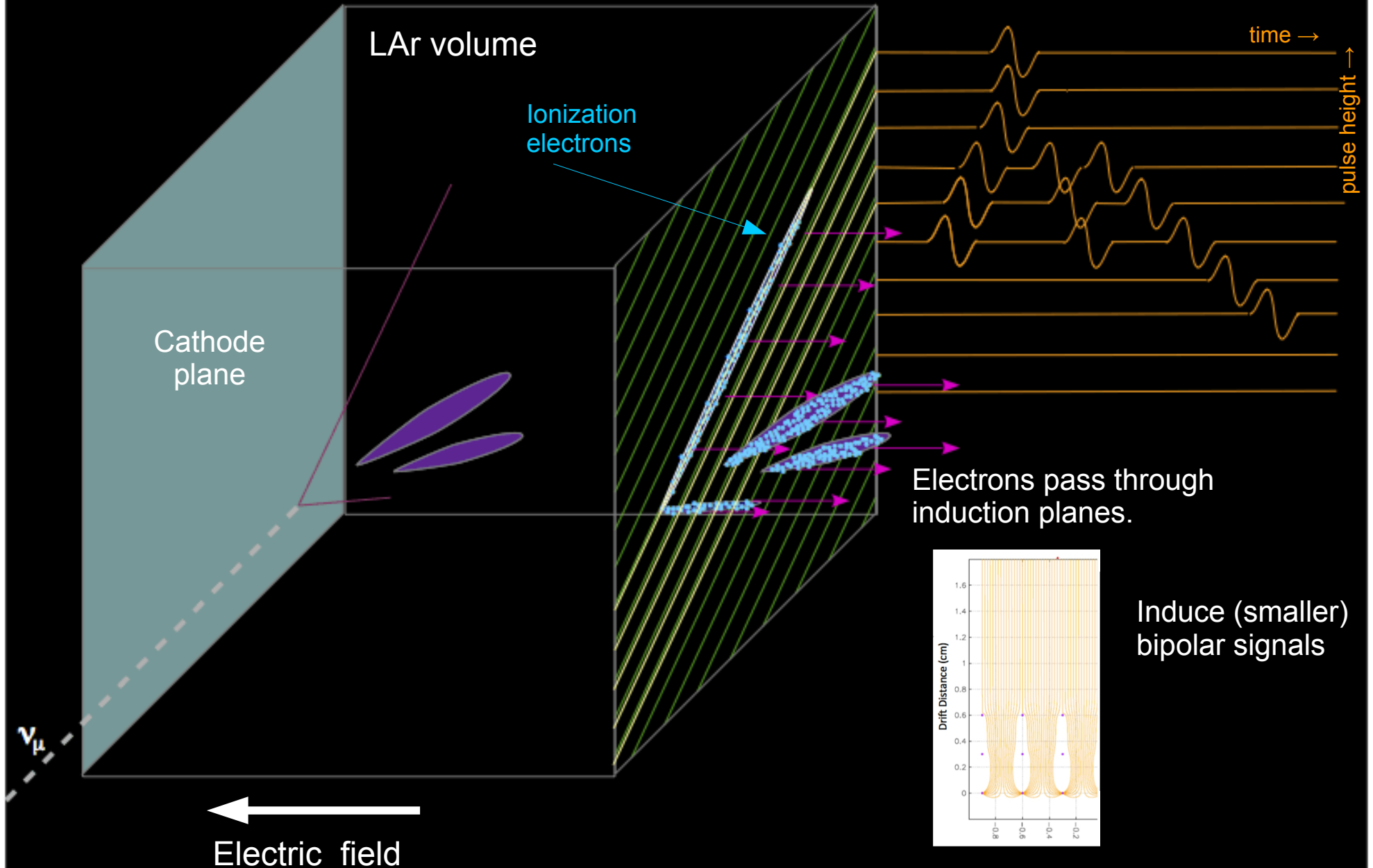


Max drift time ~ ms!!

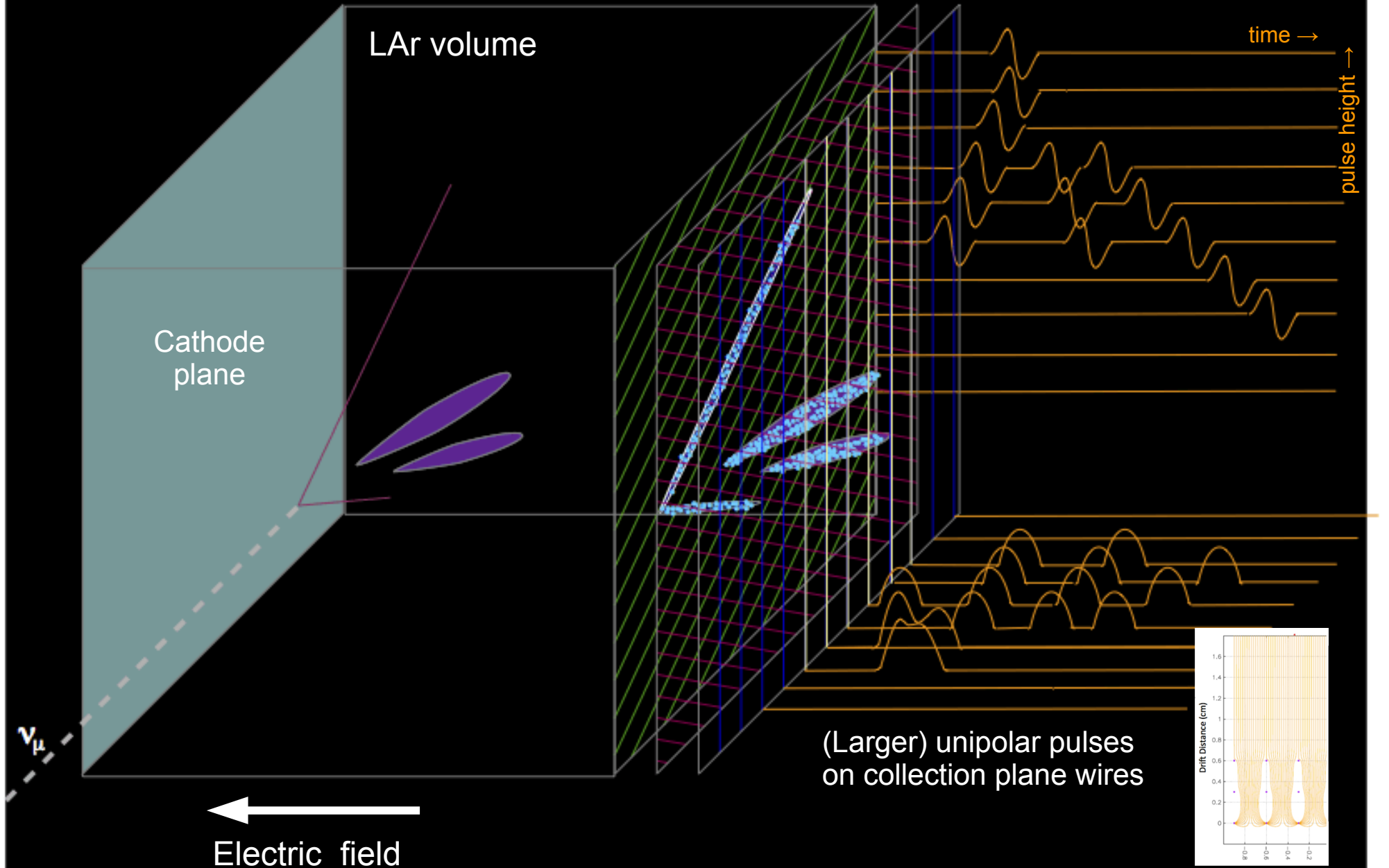
# Operation of single-phase LAr TPC



# Operation of single-phase LAr TPC



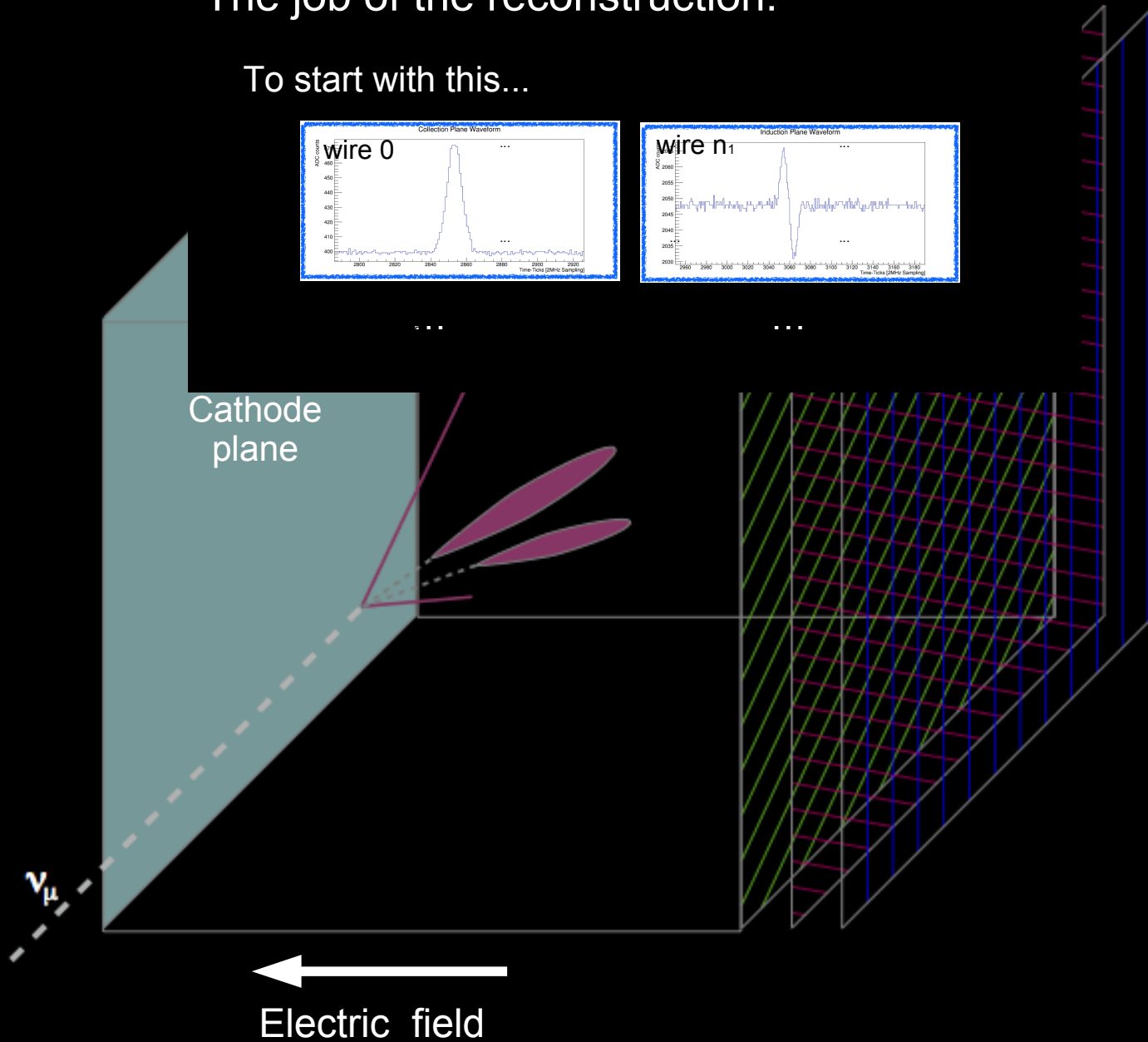
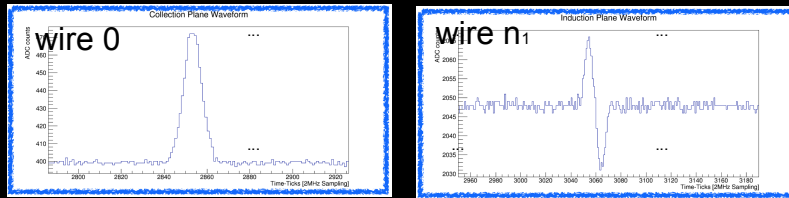
# Operation of single-phase LAr TPC



# Operation of single-phase LAr TPC

The job of the reconstruction:

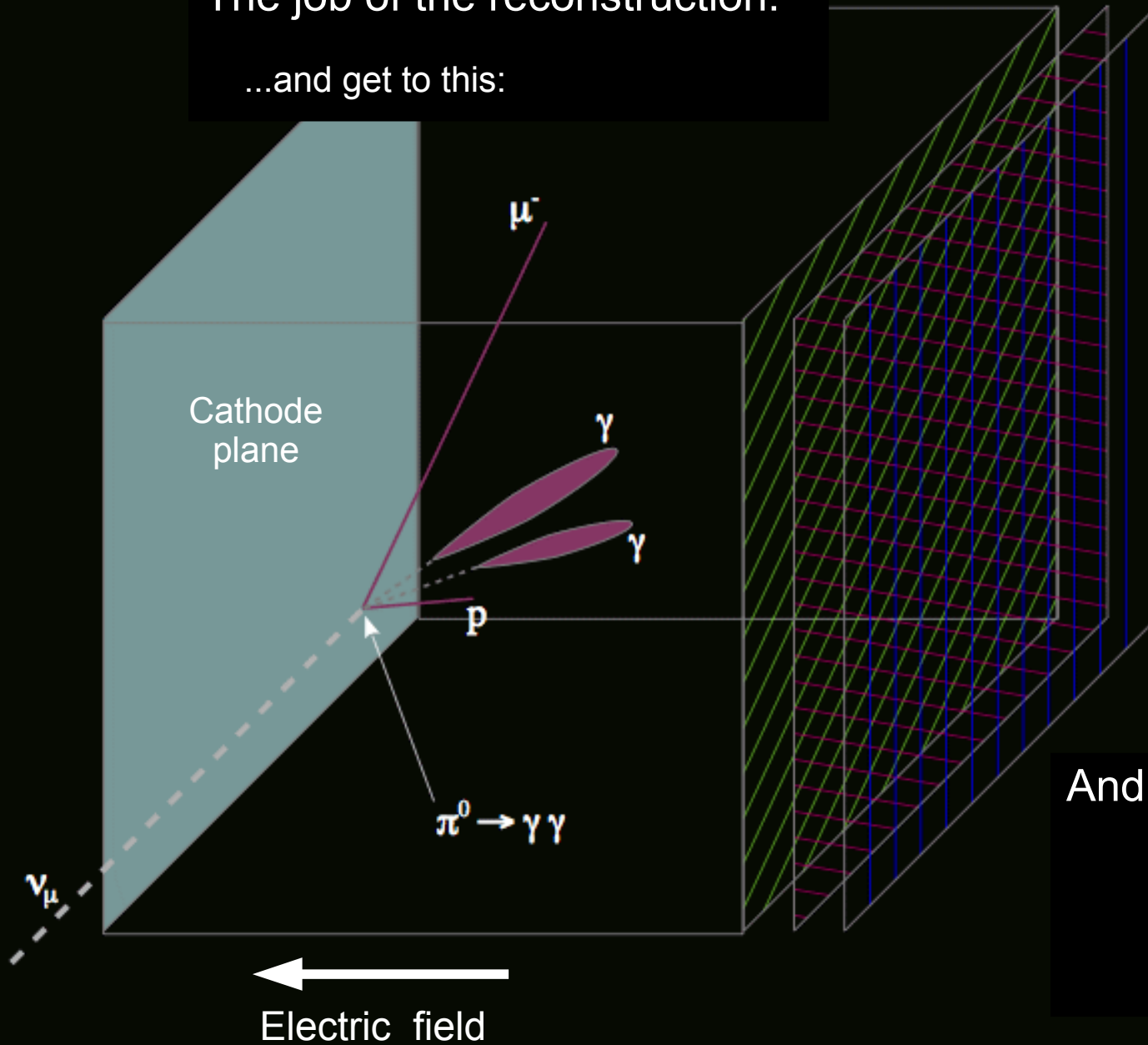
To start with this...



# Operation of single-phase LAr TPC

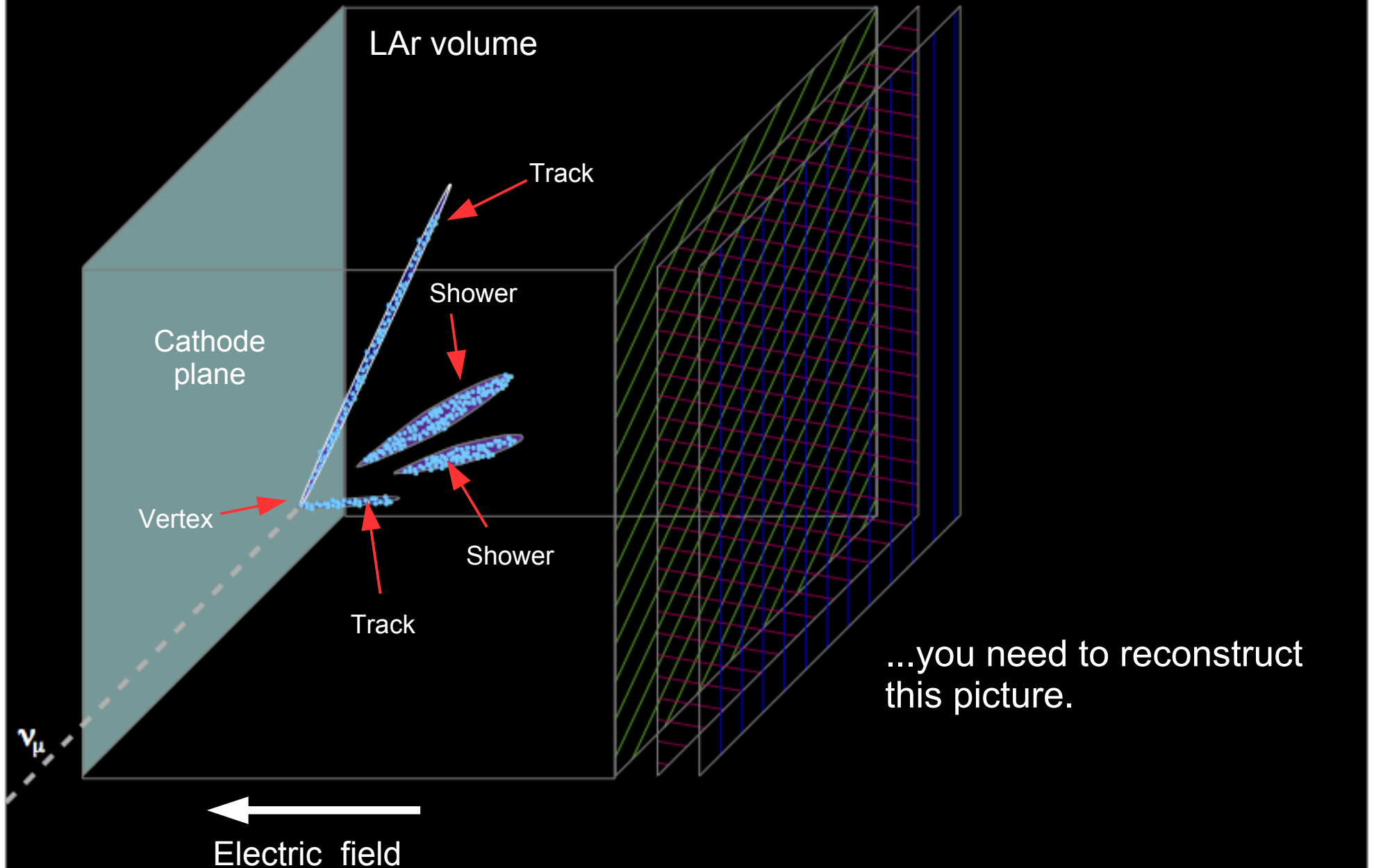
The job of the reconstruction:

...and get to this:



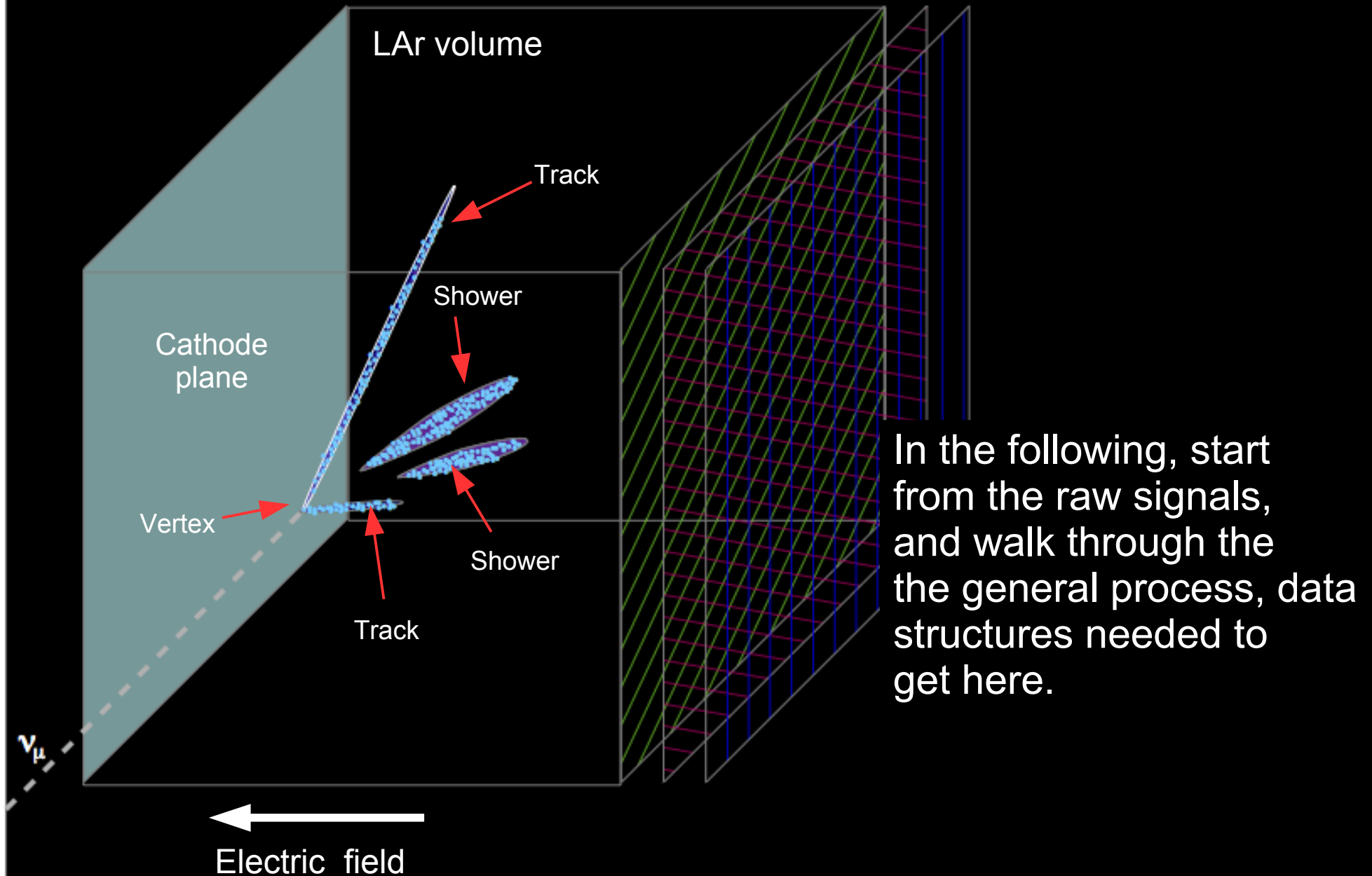
And to get here...

# Operation of single-phase LAr TPC





# Operation of single-phase LAr TPC



In the following, start from the raw signals, and walk through the the general process, data structures needed to get here.



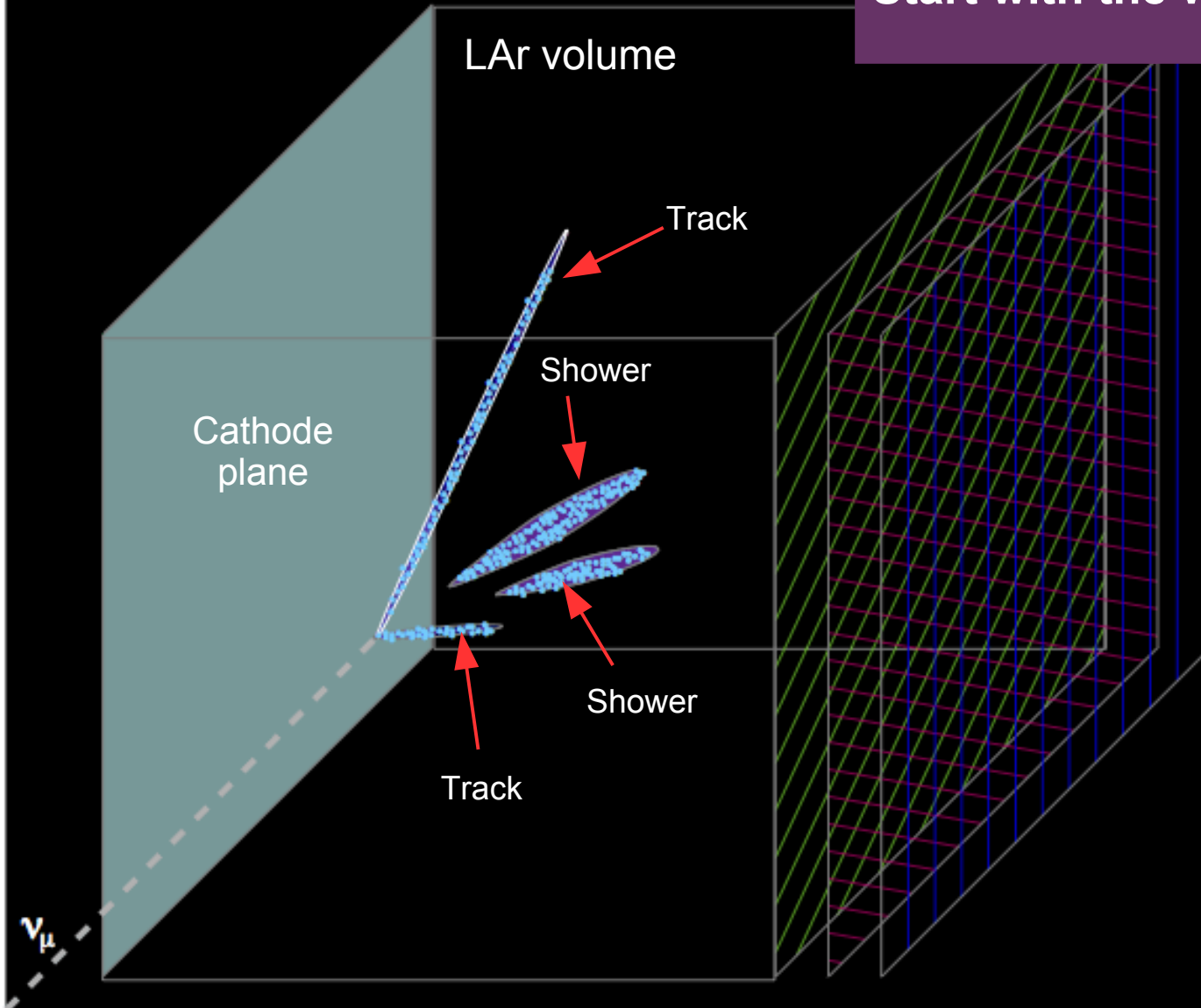
# Reconstruction workflow and data structure overview

## Comments on reconstruction workflow

- What follows is a simple, highly idealized representation of the typical reconstruction workflow
- In practice, the reconstruction
  - is highly iterative
  - has multiple parallel algorithm workflows
  - has multiple steps interleavedas increasingly complex structures are extracted from the data
- Will come back to talk about this later

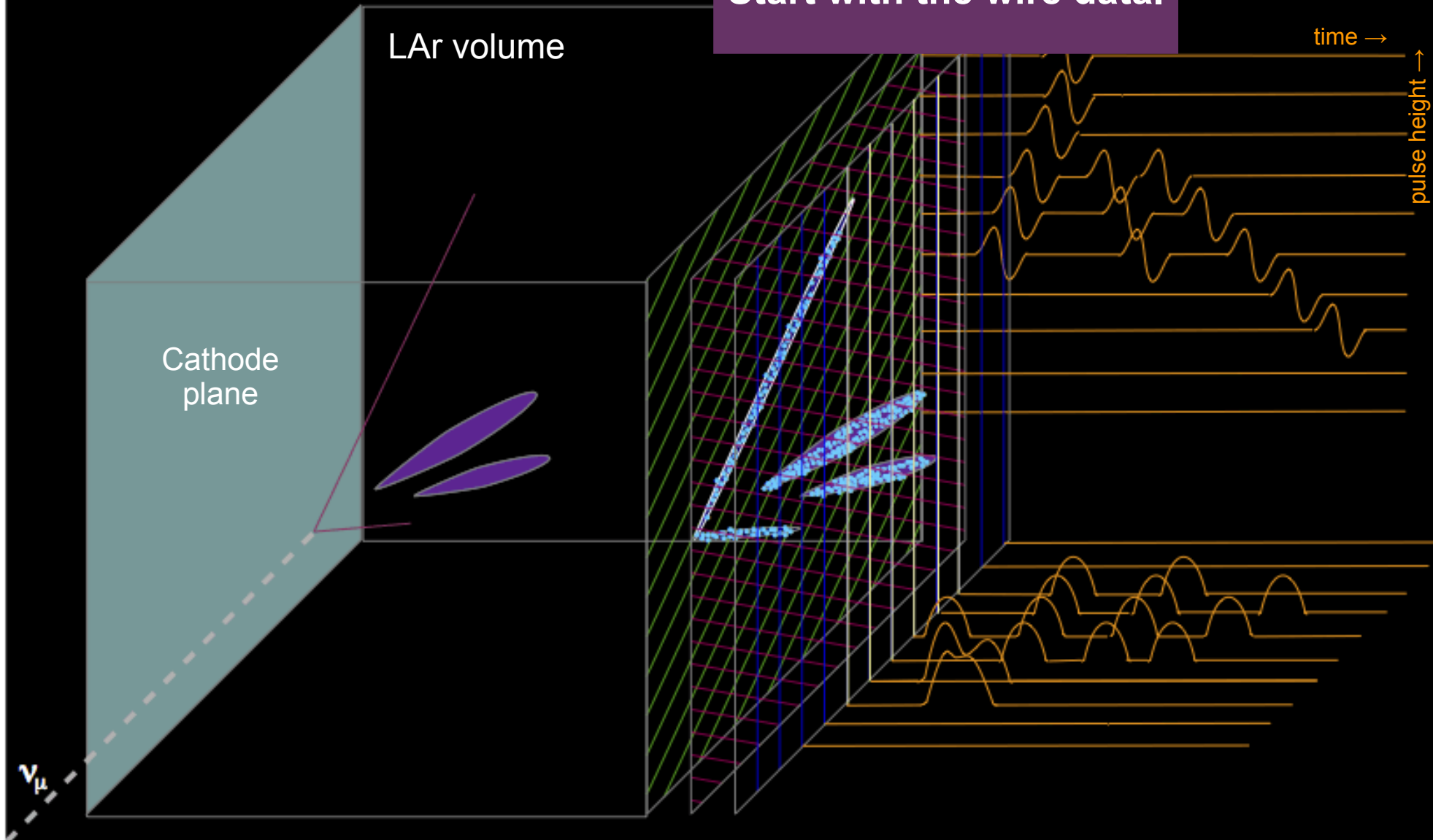
# Reconstruction workflow and data structures

Start with the wire data.



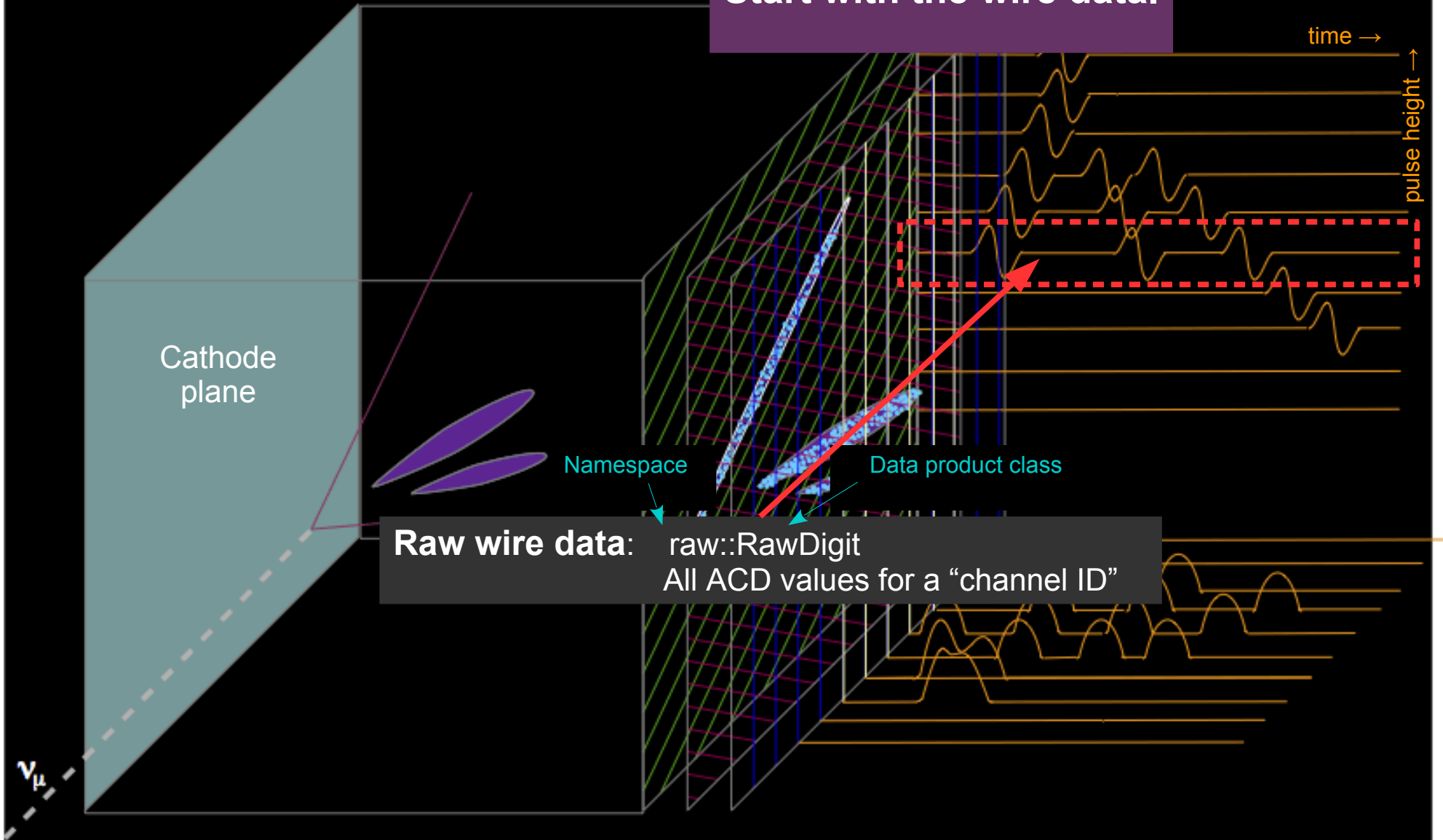
# Reconstruction workflow and data structures

Start with the wire data.



# Reconstruction workflow and data structures

Start with the wire data.



**Raw wire data:** `raw::RawDigit`  
All ACD values for a "channel ID"

# Reconstruction workflow and data structures

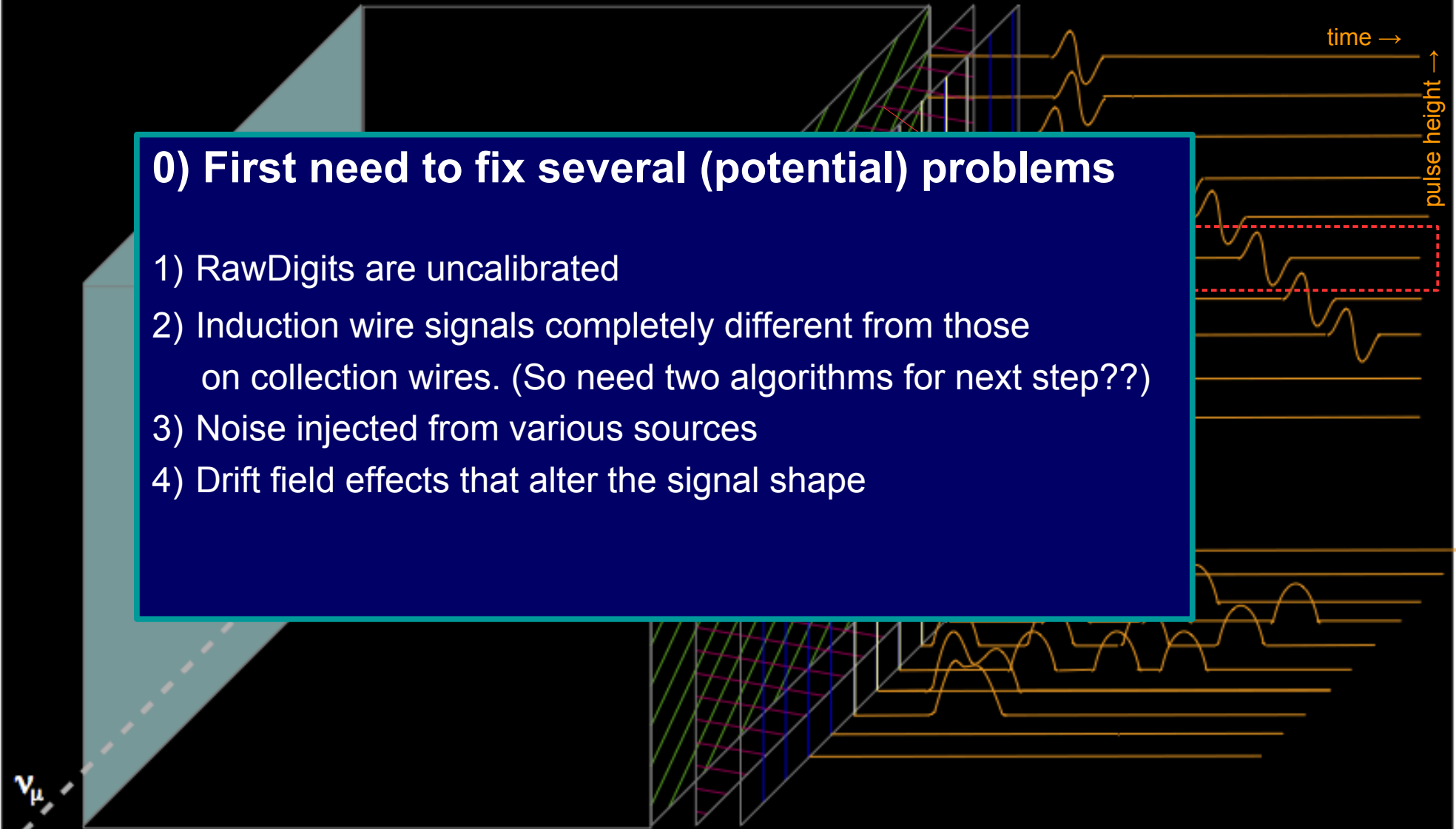
## 0) First need to fix several (potential) problems

- 1) RawDigits are uncalibrated
- 2) Induction wire signals completely different from those on collection wires. (So need two algorithms for next step??)
- 3) Noise injected from various sources
- 4) Drift field effects that alter the signal shape

$v_{\mu}$

time →

pulse height →



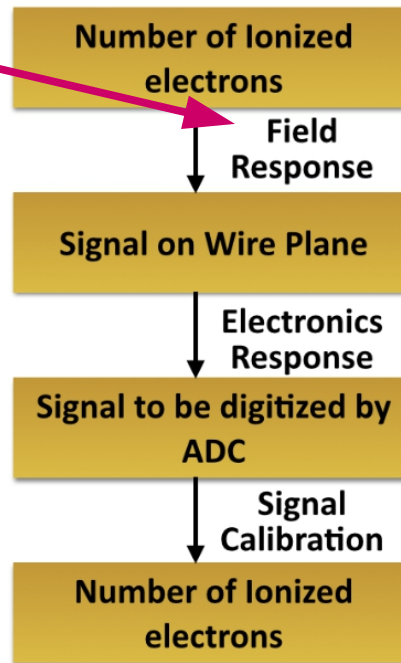
## (0) Signal processing and calibration

- Want to recover charge vs time on a wire
  - The level of detail may depend on the task
  - E.g., calorimetry may require more fidelity than pattern recognition
- Consider MicroBooNE example: simple workflow at present  
(essentially the same for DUNE single-phase detectors)
  - Calibration: baseline subtraction
  - Noise removal
  - Deconvolution
    - Removes effects of electronics response, field response

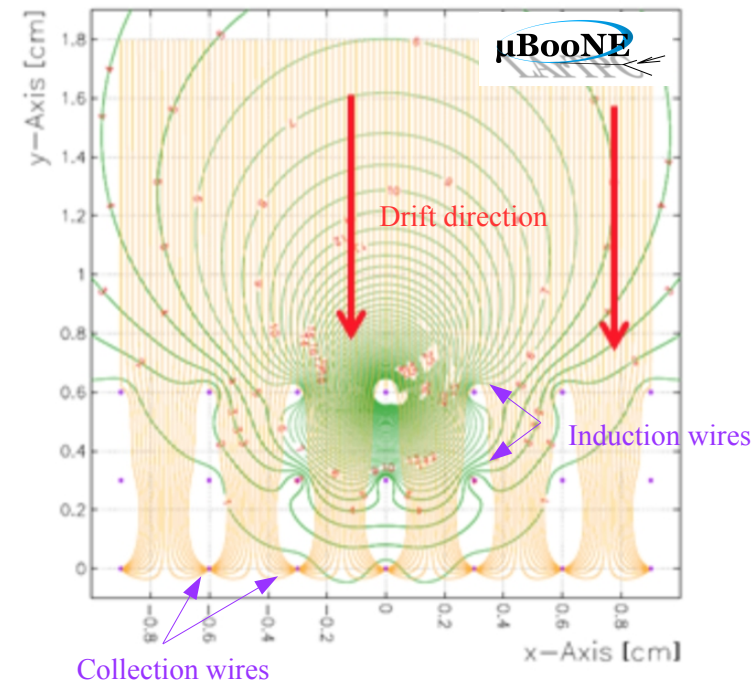
# TPC signal formation

Induced current  $i \propto q \vec{E}_w \cdot \vec{v}_q$   
 $\vec{E}_w$  = "weighting E field"  
 $\vec{v}_q$  = velocity of charge 'q'

In principle, see charge drifting to neighboring wires too

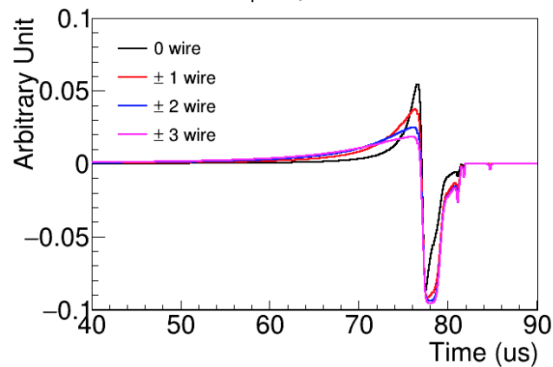


Weighting potential of a wire



MicroBooNE simulated signals

U plane, Garfield



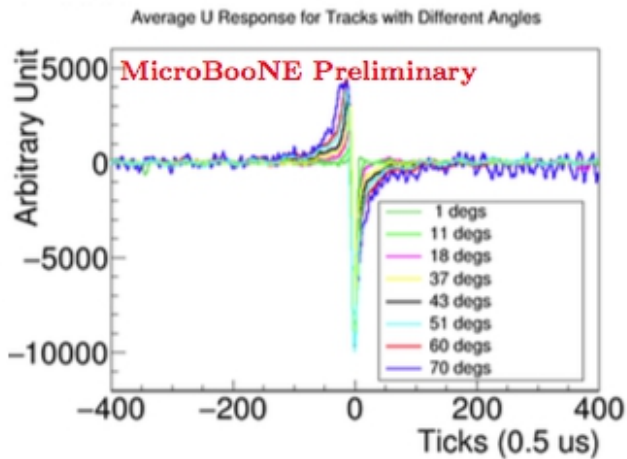
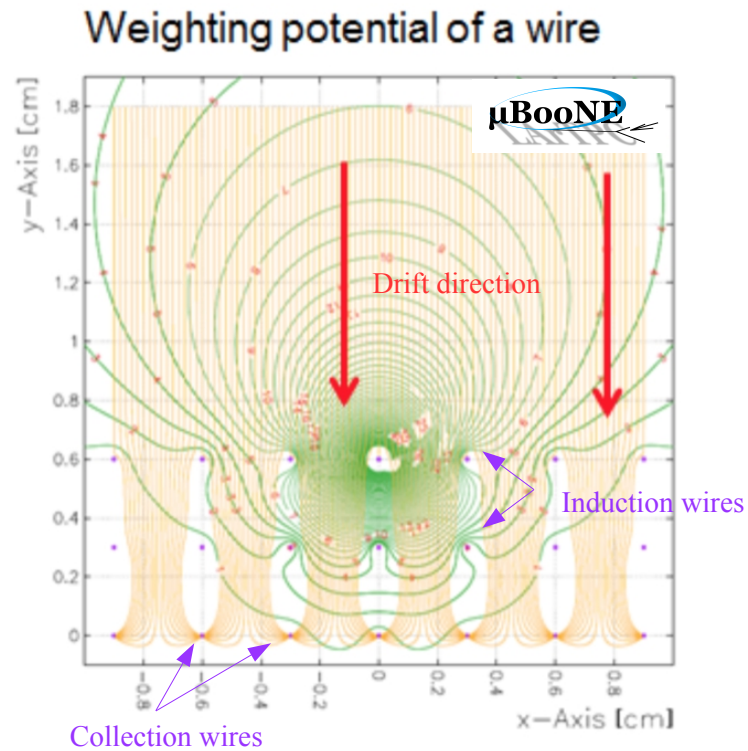
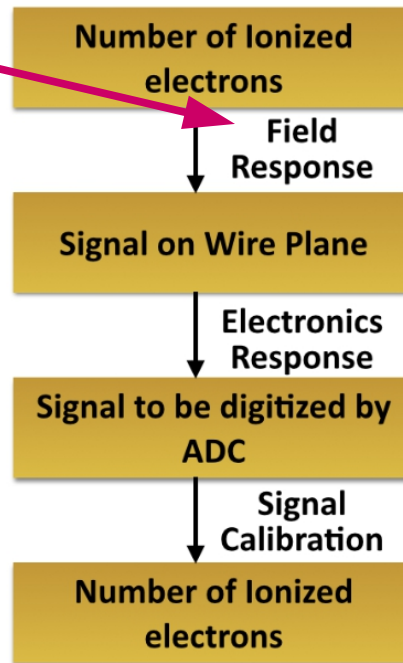
From MicroBooNE-Note-1017-pub



# TPC signal formation

Induced current  $i \propto q \vec{E}_w \cdot \vec{v}_q$   
 $\vec{E}_w$  = "weighting E field"  
 $\vec{v}_q$  = velocity of charge 'q'

The signal shape also depends on the track angle



From MicroBooNE-Note-1017-pub

Will come back to this point...

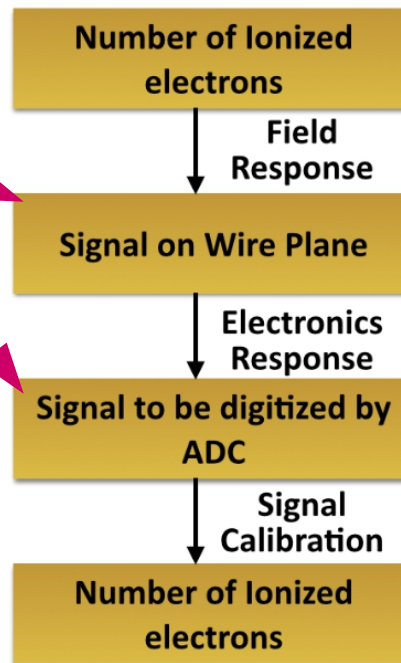
# TPC signal formation

Noise injection

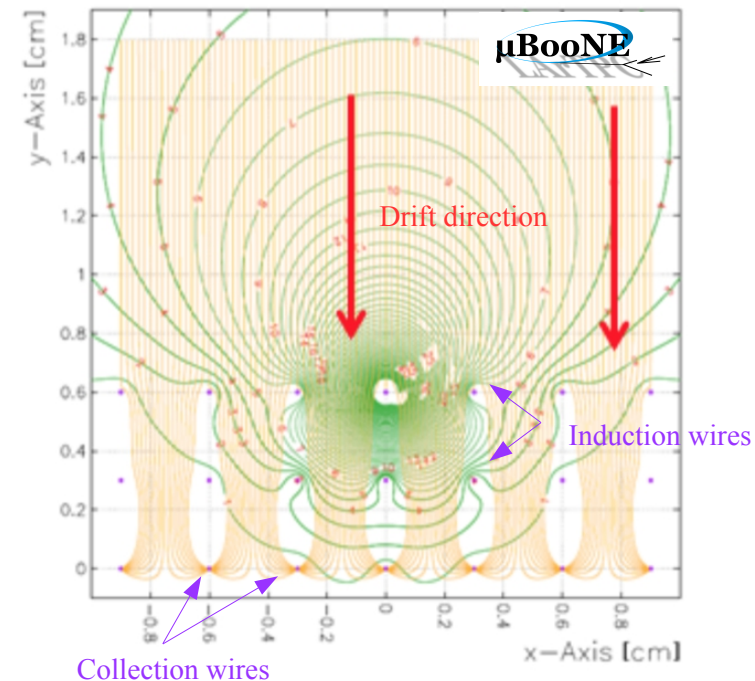
MicroBooNE example

Identified sources of noise

- First input transistor
- Warm shaping amplifier (small)
- Other readout circuits (coherent)
- Wire bias power supplies (coherent)
- Cathode HV (coherent)
- ASIC saturation due to wire motion



Weighting potential of a wire



From MicroBooNE-Note-1017-pub

Treat coherent noise with specialized algorithms

Optimal filter applicable (in principle) to remaining sources during **deconvolution**

# Deconvolution

- Given

$$y(t) = (h * x)(t) + n(t)$$

- where  $y(t)$  = **measured output signal** (raw digits)

$(h * x)(t)$  = convolution of impulse response  $h(t)$  and (unknown) input signal  $x(t)$

$n(t)$  = (unknown) noise

- Optimal signal estimate (minimum mean squared error) given only

- Finite **impulse response** of the front-end electronics
- Estimated **mean power spectrum** for the **signal** and the **noise**

(used in a Weiner filter)

assuming stationary signal and noise power spectra

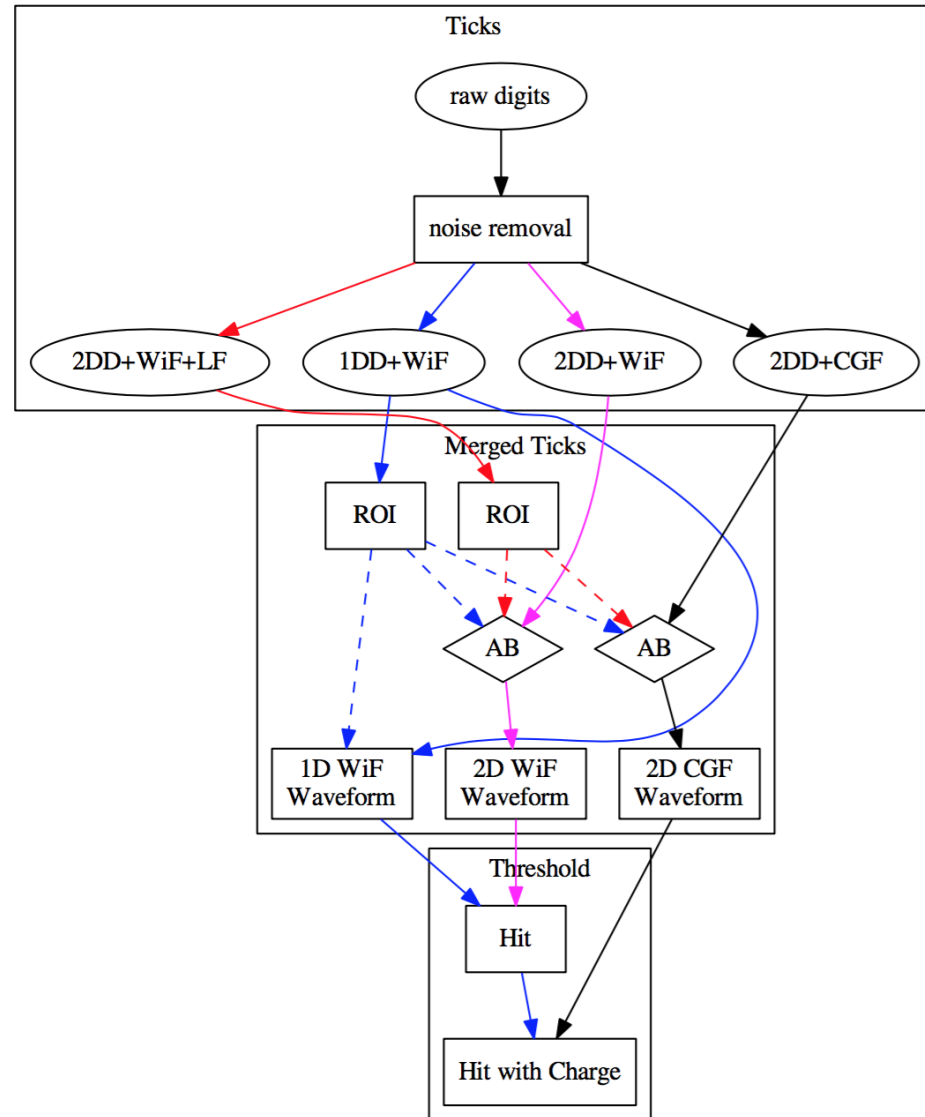
- But signal spectrum

- depends on track angle
- signals on nearby tracks

In practice, the deconvolution procedure is more complex

Again, the MicroBooNE example ca. 2016

Involves additional Gaussian and low-frequency filtering to control artifacts from noise, 2D deconvolution to remove effects from charge on neighboring wires, region-of-interest identification to reduce data size

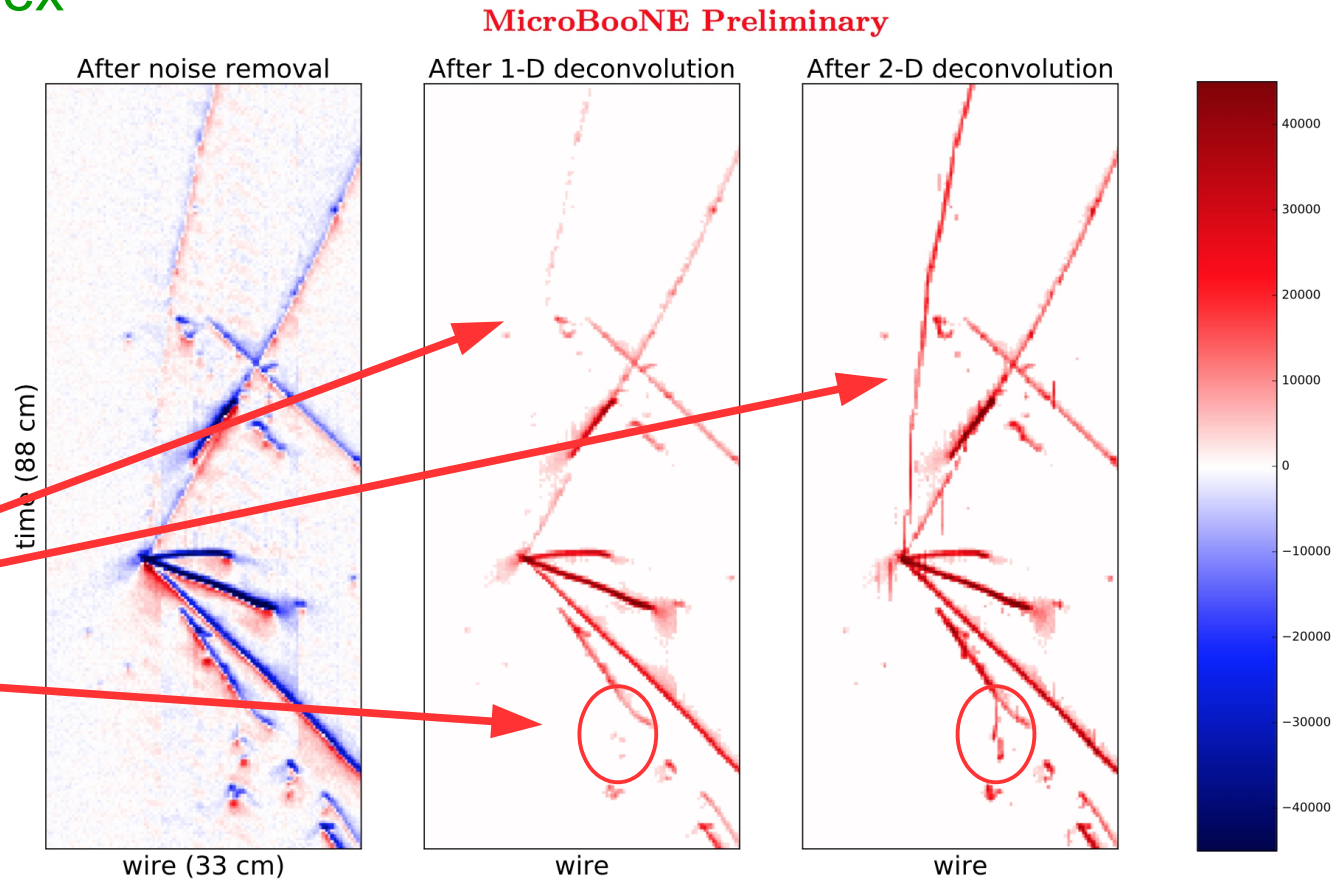


From MicroBooNE-Note-1017-pub

In practice, the deconvolution procedure is more complex

Again, the MicroBooNE example ca. 2016

It pays to get this right!!



From MicroBooNE-Note-1017-pub

# Wire calibration and deconvolution

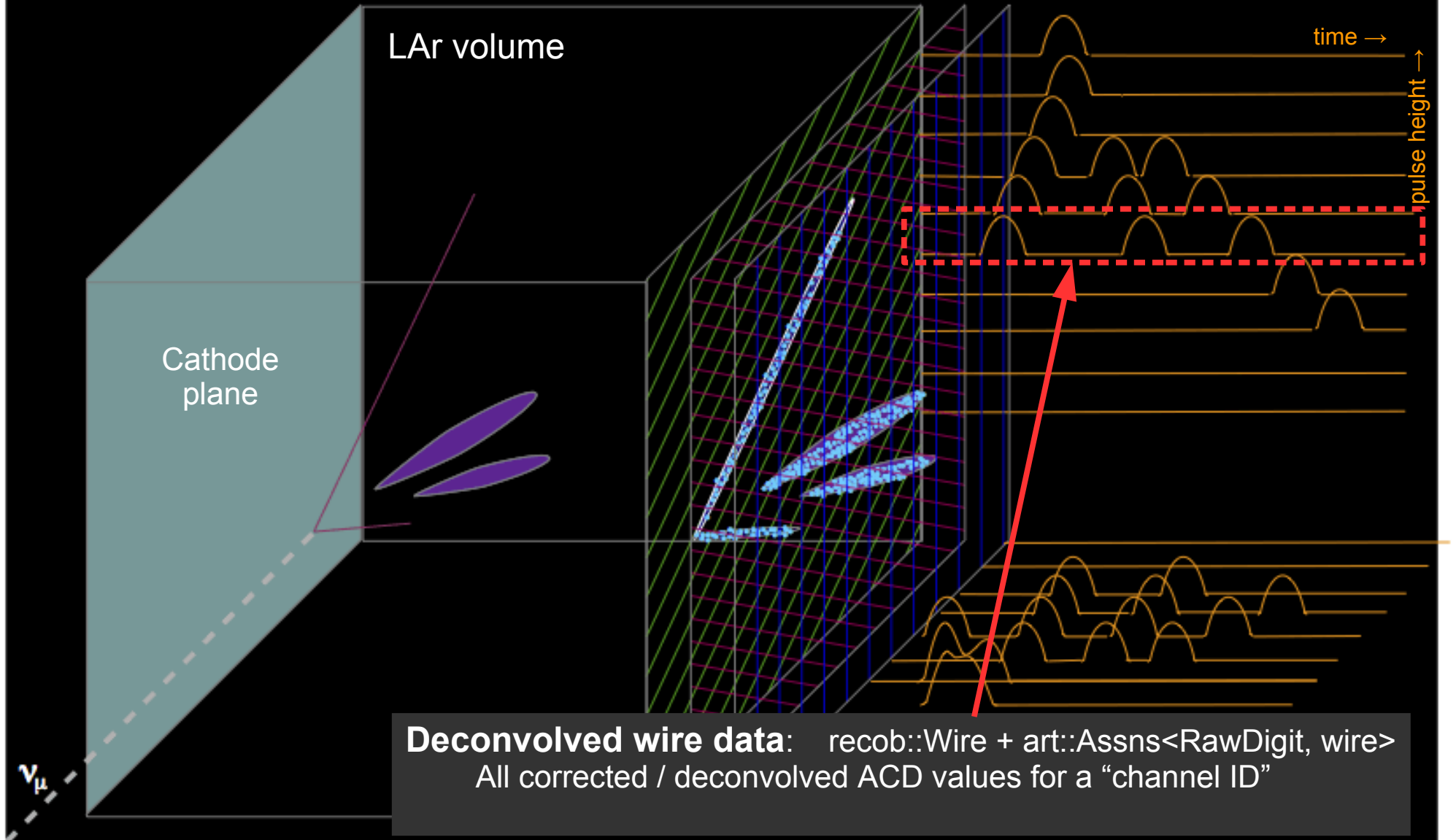
- Calibration
  - Only pedestal subtraction at this phase.  
(Channel gains come later...)
- Deconvolution

**Performed by a number of classes**

CalWire ...		
CalROI ...		
SignalShapingService ...		(1D)
SignalShaping		
LArFFT		
Wire-cell		(2D)

(Experiment-specific at the moment, but can be shared)

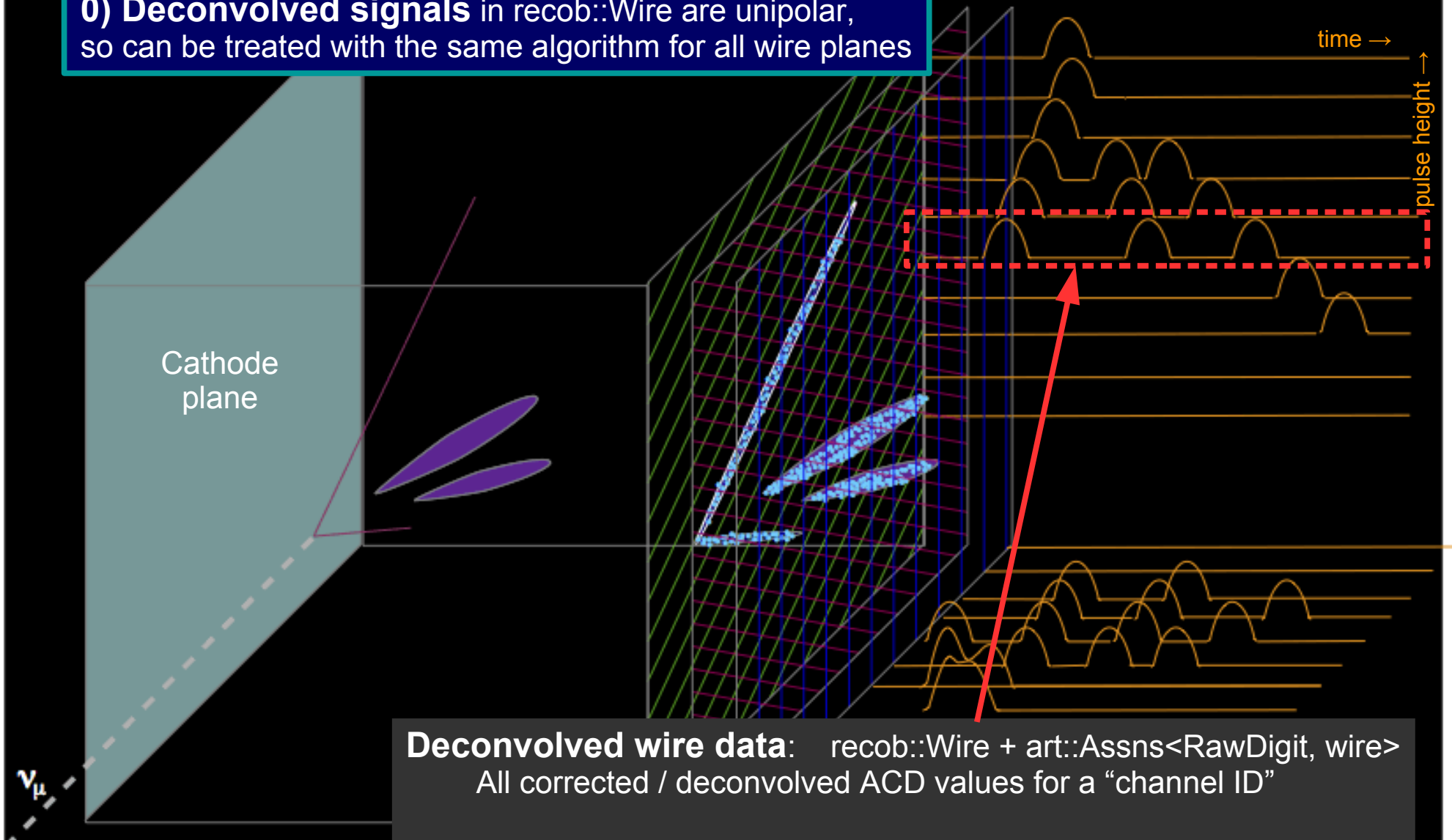
# Reconstruction workflow and data structures





# Reconstruction workflow and data structures

**0) Deconvolved signals** in `recob::Wire` are unipolar, so can be treated with the same algorithm for all wire planes





# Reconstruction workflow and data structures

**0) Deconvolved signals** in `recob::Wire` are unipolar, so can be treated with the same algorithm for all wire planes

Cathode plane

For “ROI” algorithms, keep only the values in “regions of interest” (i.e., zero-supressed), but still store result in `recob::Wire`

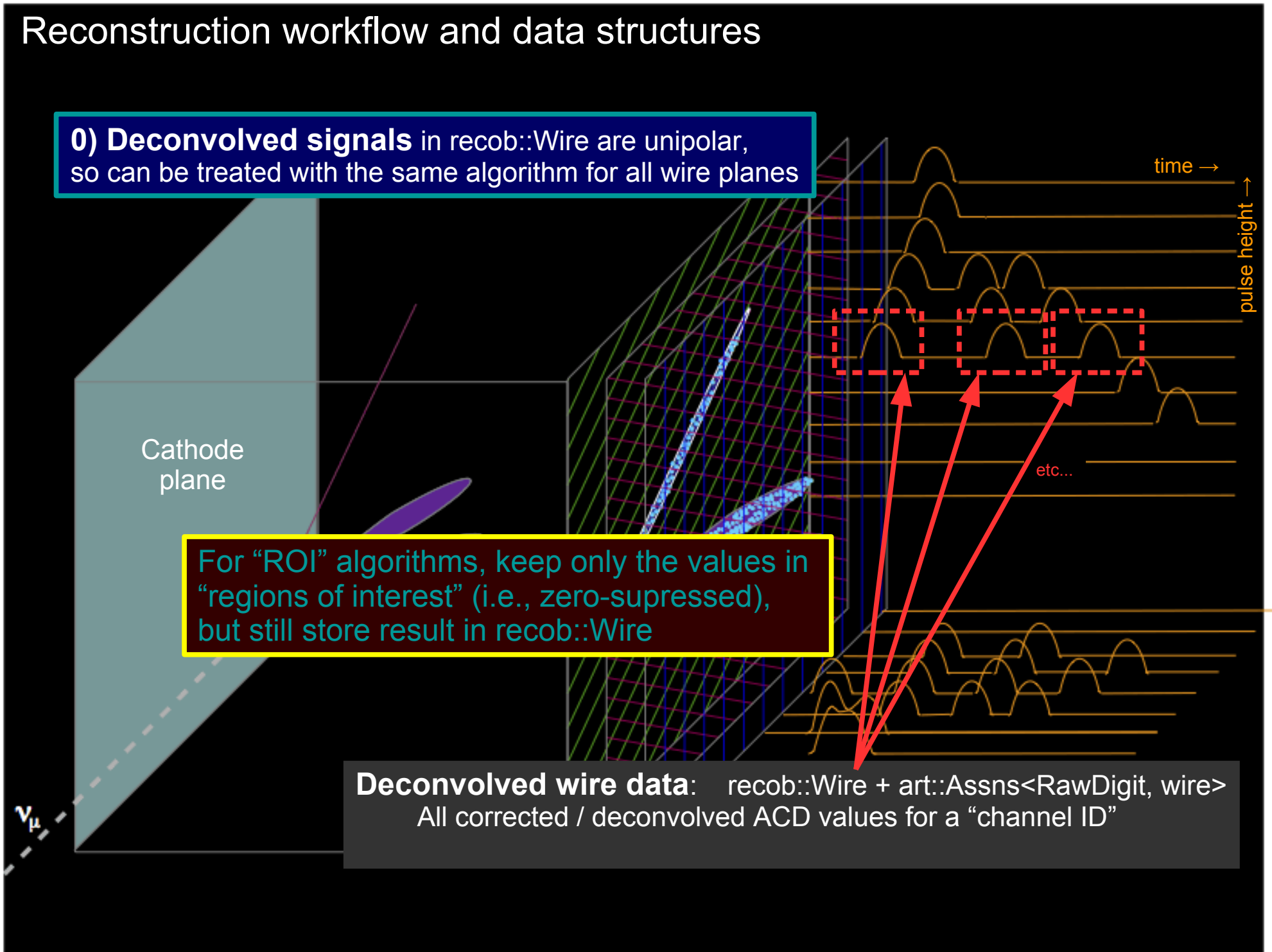
**Deconvolved wire data:** `recob::Wire + art::Assns<RawDigit, wire>`  
All corrected / deconvolved ACD values for a “channel ID”

time →

pulse height →

etc...

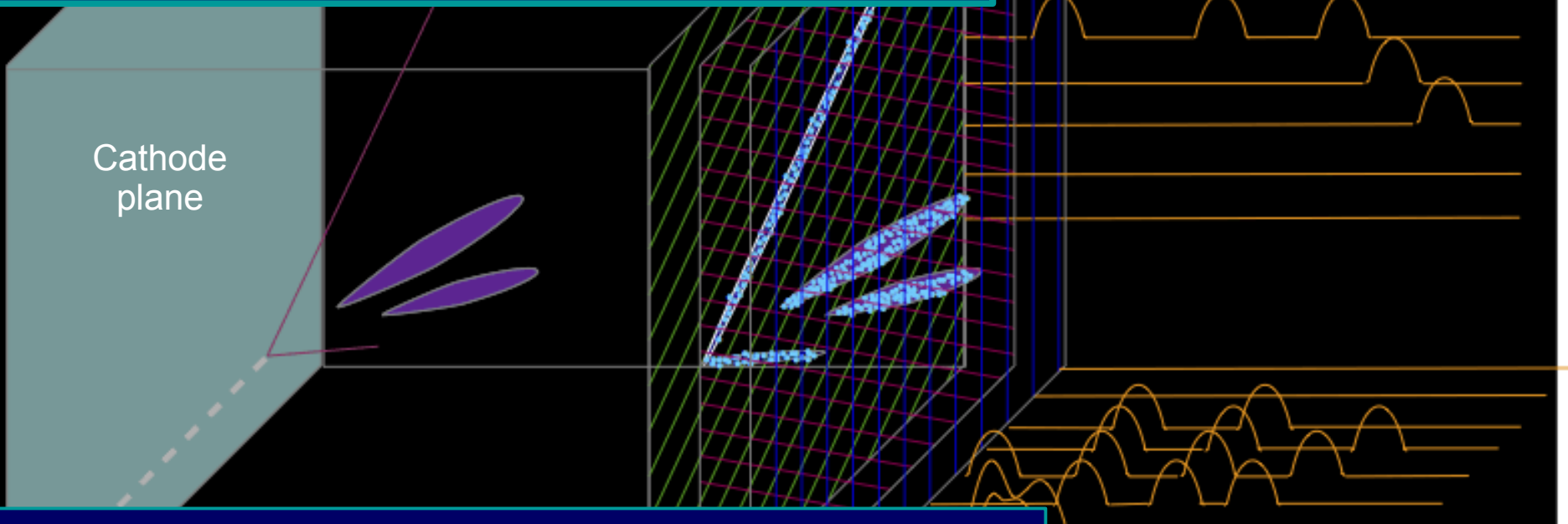
$v_{\mu}$



# Reconstruction workflow and data structures

**1) Now group together the ADC values** on each wire that correspond to the ionization associated with a single particle (...more or less...) as it traverses the measurement volume for that wire.

This is “hit-finding”.



**Hits are input to determine:**

- 1) the position of the track / energy deposition
- 2) the total charge in the hit, and therefore the energy deposition within the measurement volume of the wire

# Reconstruction workflow and data structures

**1) Now group together the ADC values** on each wire that correspond to the ionization associated with a single particle (...more or less...) as it traverses the measurement volume for that wire.

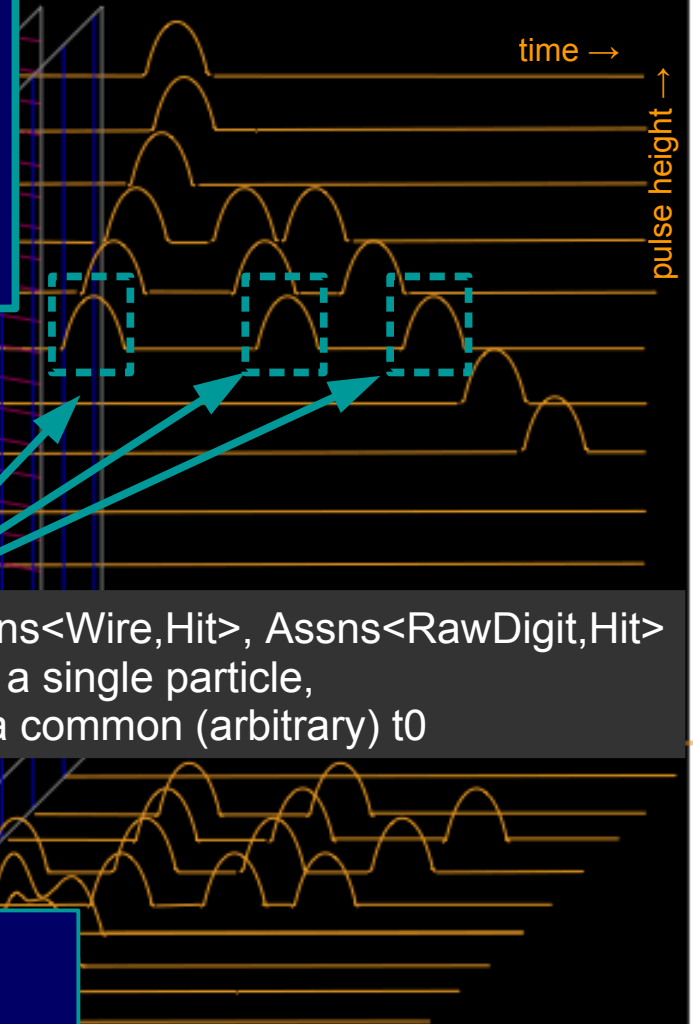
This is “hit-finding”.

Cathode plane

**The output of hit-finding:** `recob::Hit + Assns<Wire,Hit>, Assns<RawDigit,Hit>`  
All ADC values on a given wire attributed to a single particle, and the arrival time of ionization relative to a common (arbitrary)  $t_0$

**Hits are input to determine:**

- 1) the position of the track / energy deposition
- 2) the total charge in the hit, and therefore the energy deposition within the measurement volume of the wire



# Reconstruction workflow and data structures

**1) Now group together the ADC values** on each wire that correspond to the ionization associated with a single particle (...more or less...) as it traverses the measurement volume for that wire.

This is “hit-finding”.

Cathode plane

**The output of hit-finding:** `recob::Hit + Assns<Wire,Hit>, Assns<RawDigit,Hit>`  
All ADC values on a given wire attributed to a single particle, and the arrival time of ionization relative to a common (arbitrary)  $t_0$

**Hit-finding performed by:**

CCHitFinder  
GausHitFinder  
RawHitFinder

...

Mainly use this at present

time →

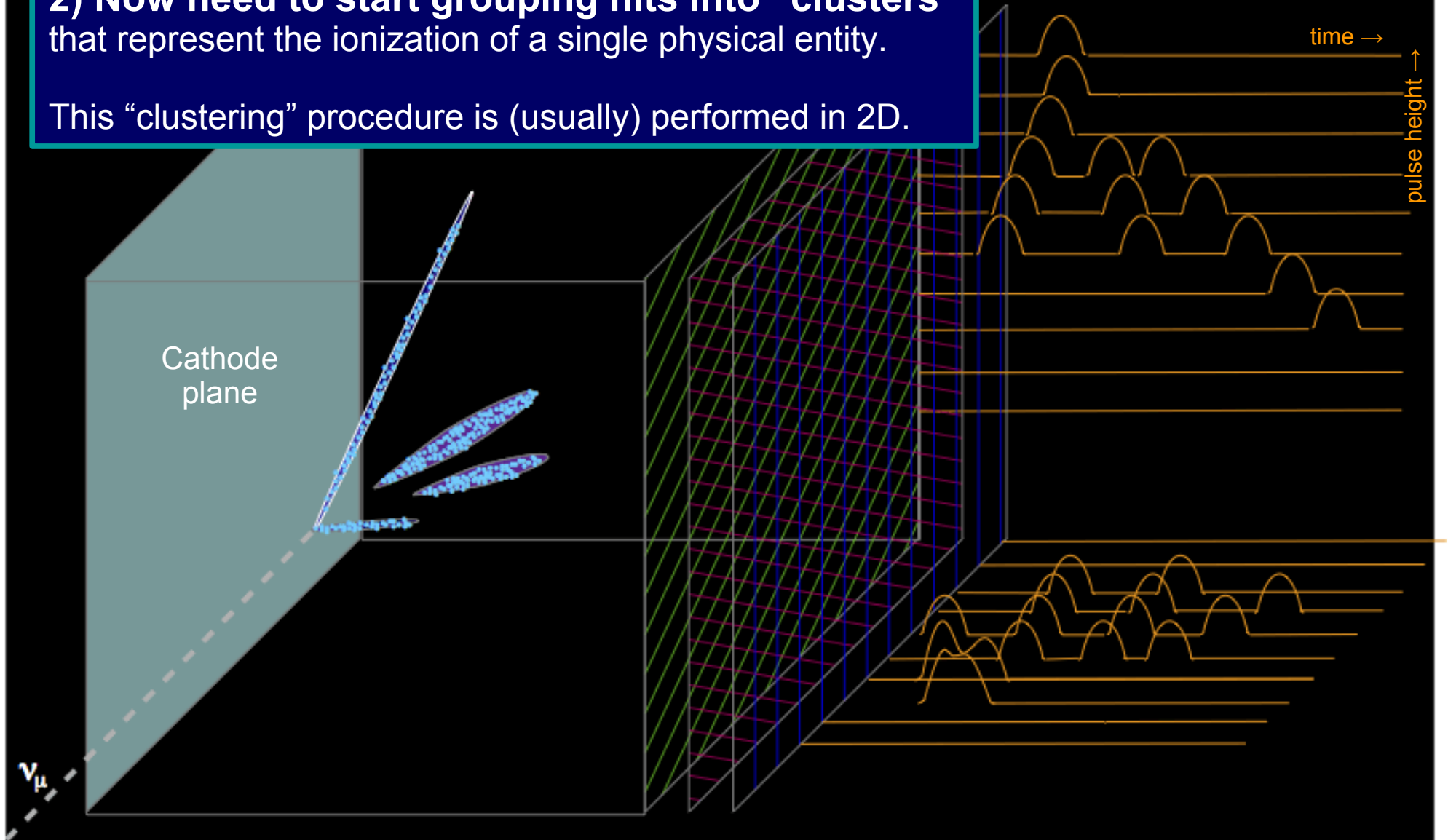
pulse height →

$v_{\mu}$

# Reconstruction workflow and data structures

**2) Now need to start grouping hits into “clusters”** that represent the ionization of a single physical entity.

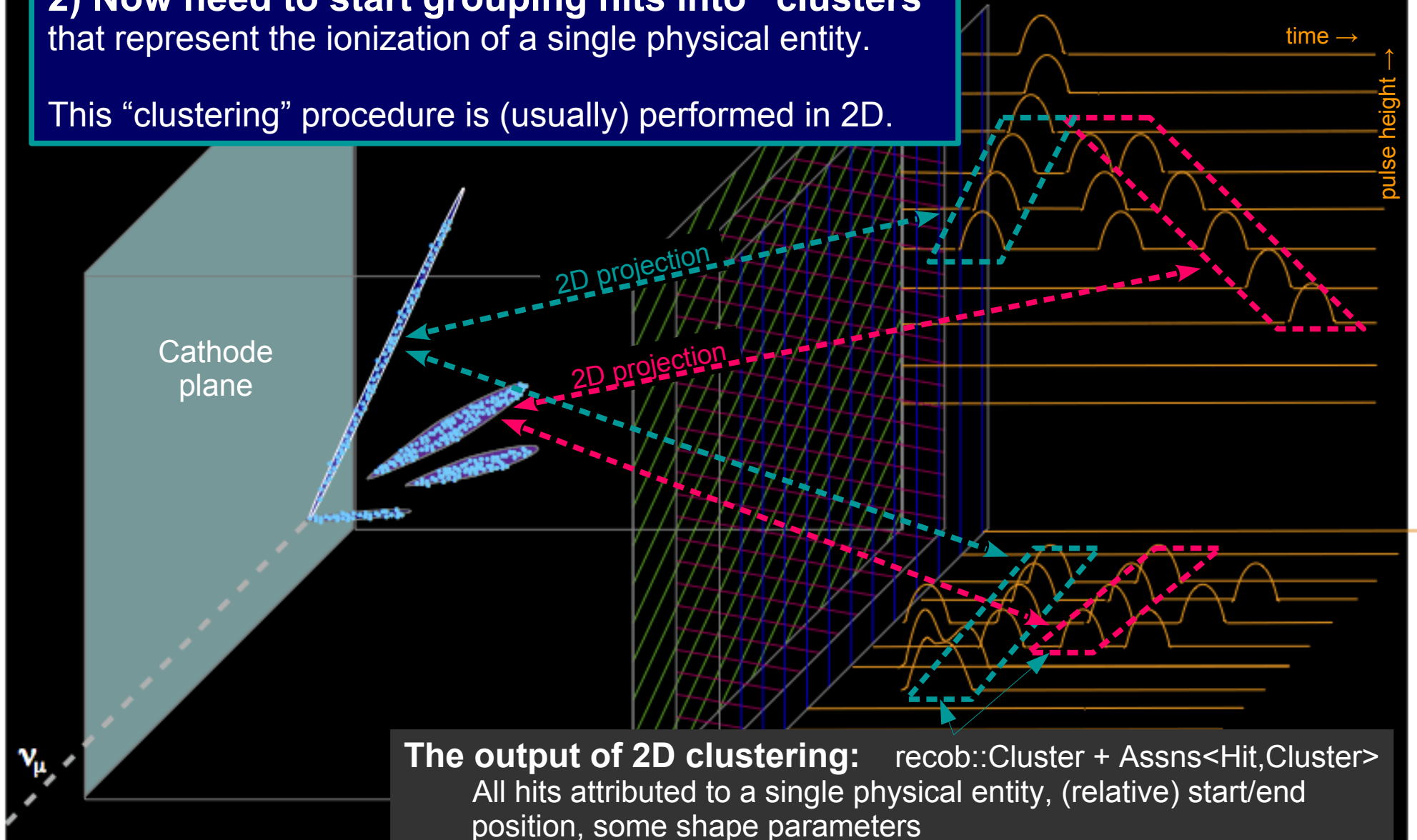
This “clustering” procedure is (usually) performed in 2D.



# Reconstruction workflow and data structures

**2) Now need to start grouping hits into “clusters”** that represent the ionization of a single physical entity.

This “clustering” procedure is (usually) performed in 2D.





# Reconstruction workflow and data structures

**2) Now need to start grouping hits into “clusters”** that represent the ionization of a single physical entity.

This “clustering” procedure is (usually) performed in 2D.

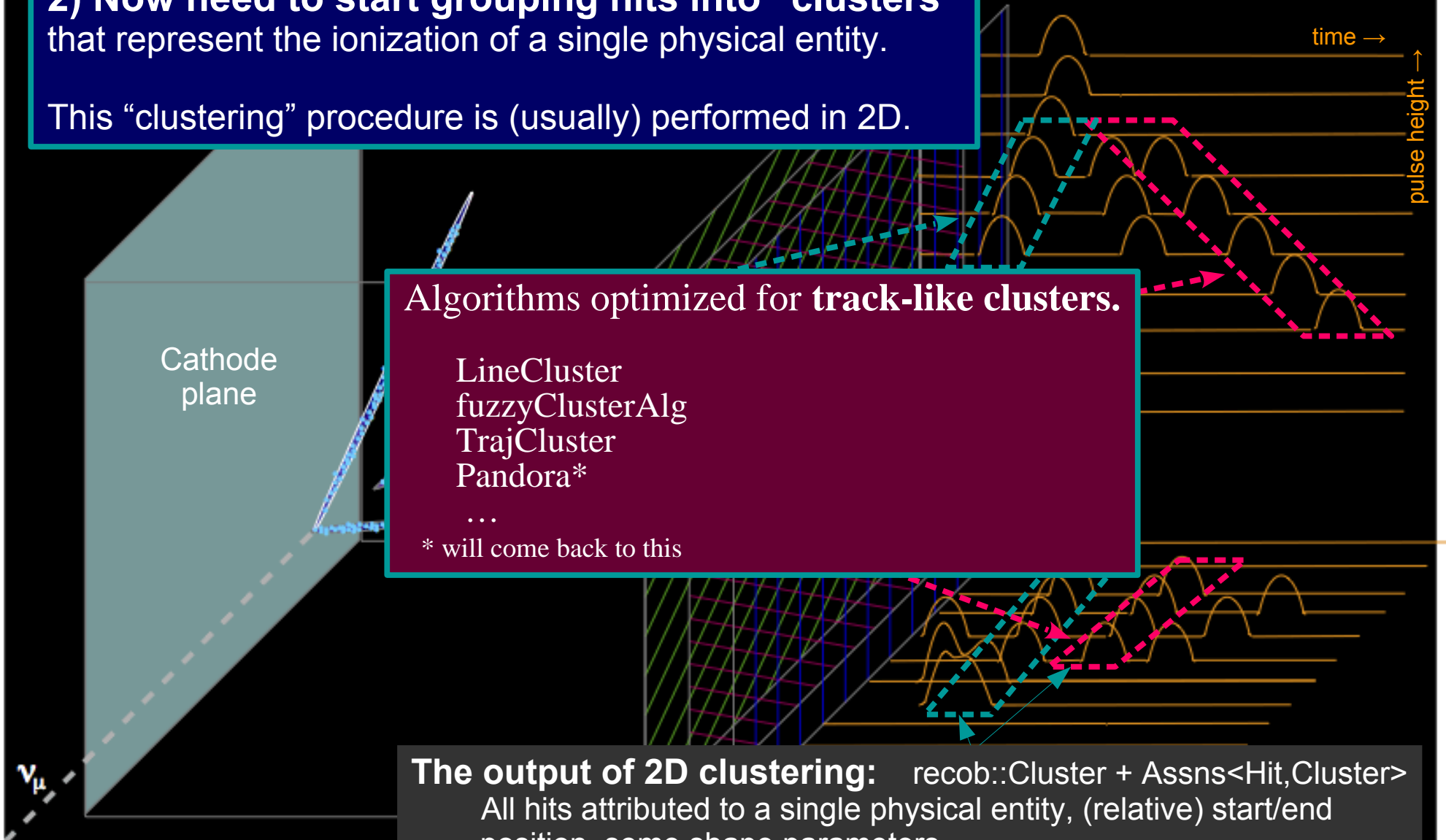
Algorithms optimized for **track-like clusters**.

LineCluster  
fuzzyClusterAlg  
TrajCluster  
Pandora\*

...

\* will come back to this

**The output of 2D clustering:** `recob::Cluster + Assns<Hit,Cluster>`  
All hits attributed to a single physical entity, (relative) start/end position, some shape parameters



# Reconstruction workflow and data structures

2) Now need to start grouping hits into “clusters” that represent the ionization of a single physical entity.

This “clustering” procedure is (usually) performed in 2D.

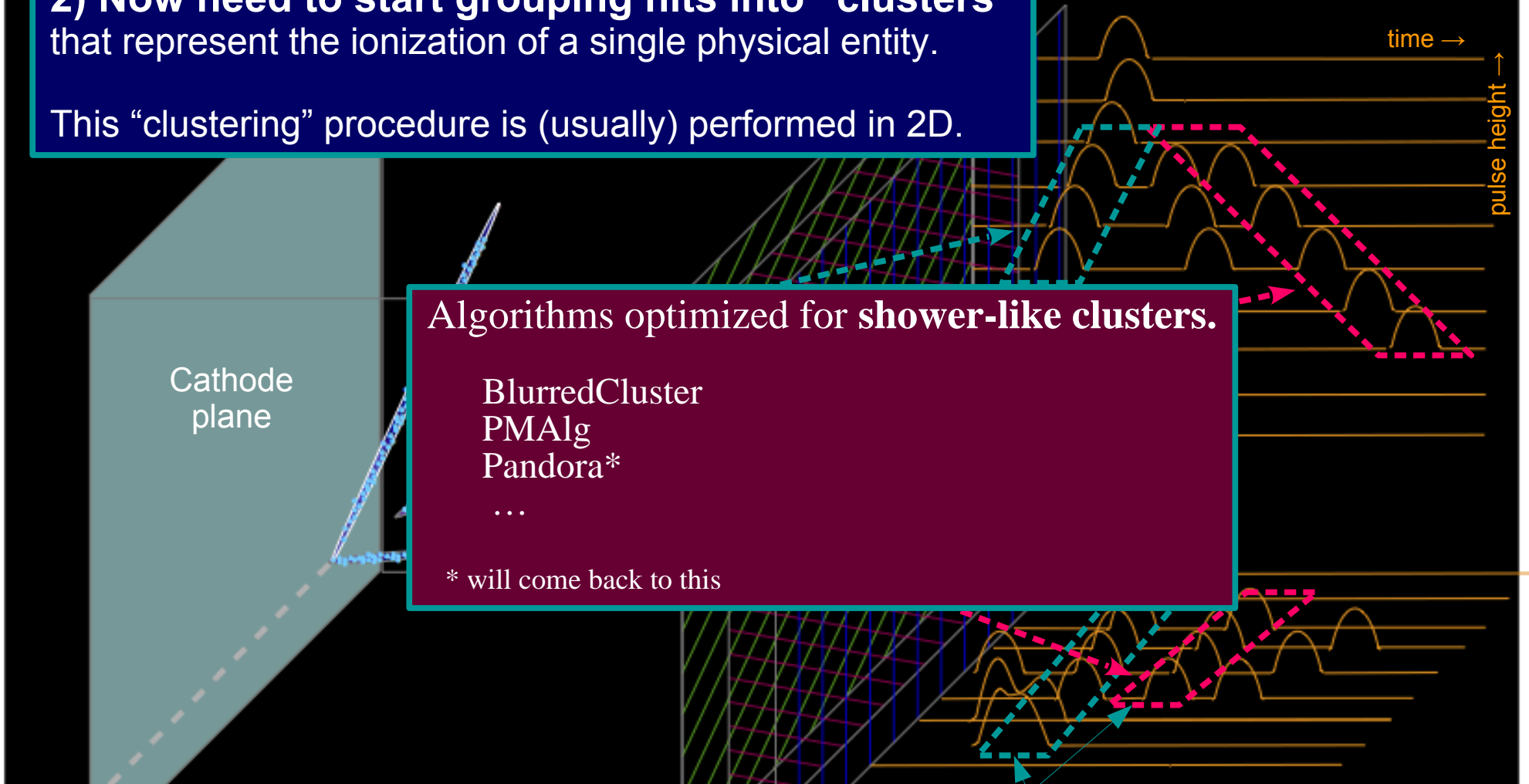
Algorithms optimized for **shower-like clusters**.

BlurredCluster  
PMAlg  
Pandora\*

...

\* will come back to this

**The output of 2D clustering:** `recob::Cluster + Assns<Hit,Cluster>`  
All hits attributed to a single physical entity, (relative) start/end position, some shape parameters





# Reconstruction workflow and data structures

**2) Now need to start grouping hits into “clusters”** that represent the ionization of a single physical entity.

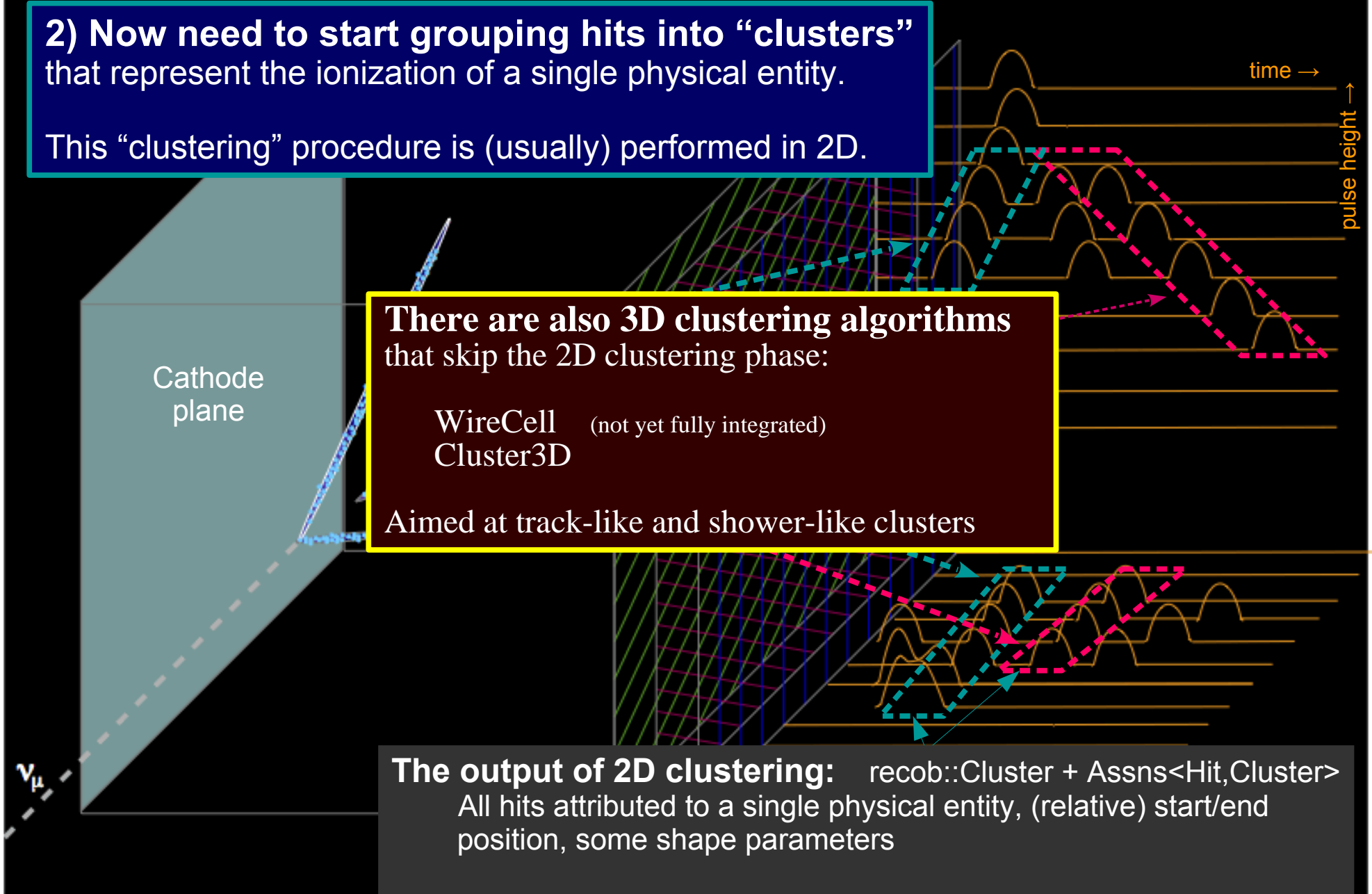
This “clustering” procedure is (usually) performed in 2D.

**There are also 3D clustering algorithms** that skip the 2D clustering phase:

WireCell (not yet fully integrated)  
Cluster3D

Aimed at track-like and shower-like clusters

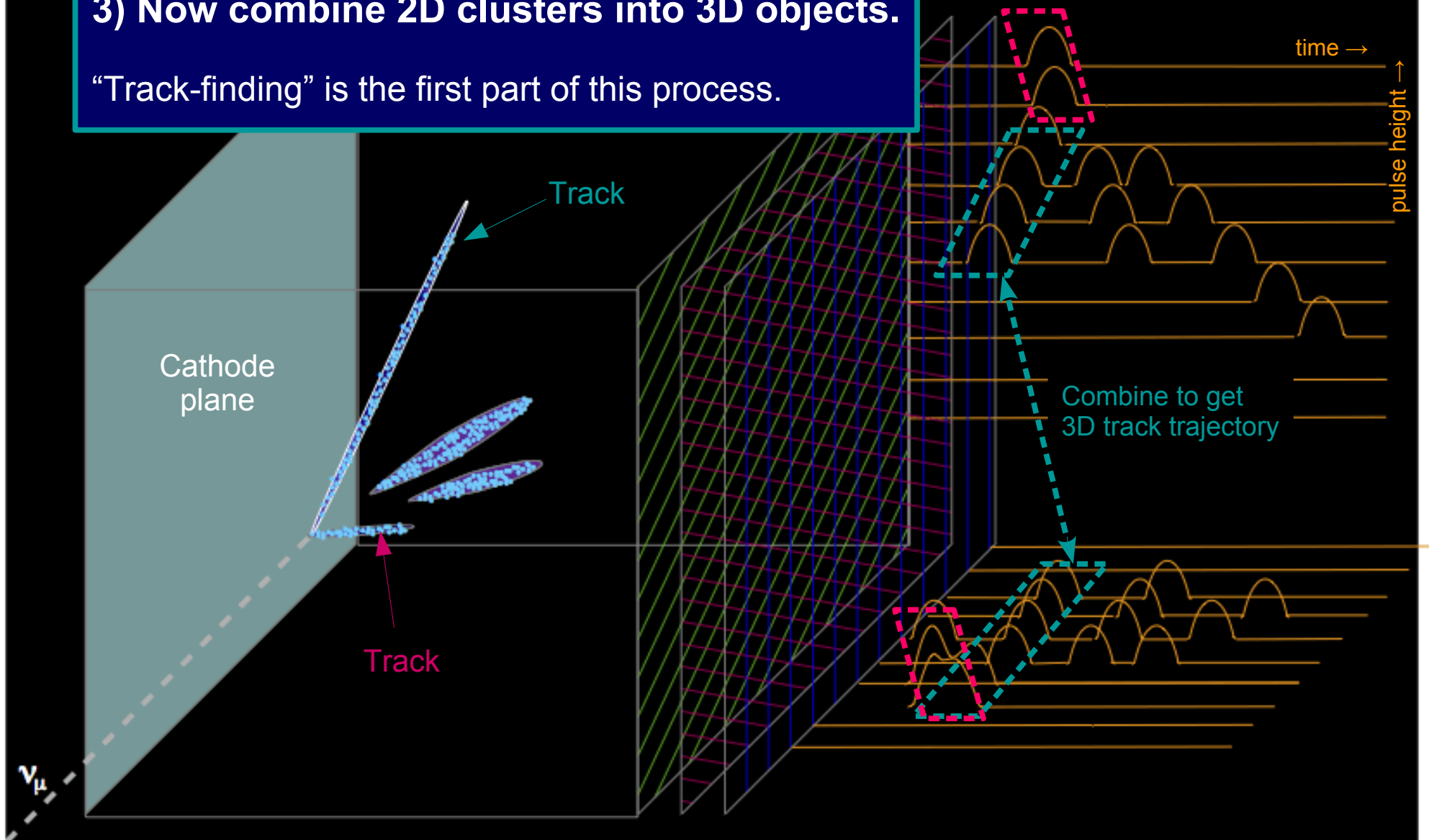
**The output of 2D clustering:** `recob::Cluster + Assns<Hit,Cluster>`  
All hits attributed to a single physical entity, (relative) start/end position, some shape parameters



# Reconstruction workflow and data structures

**3) Now combine 2D clusters into 3D objects.**

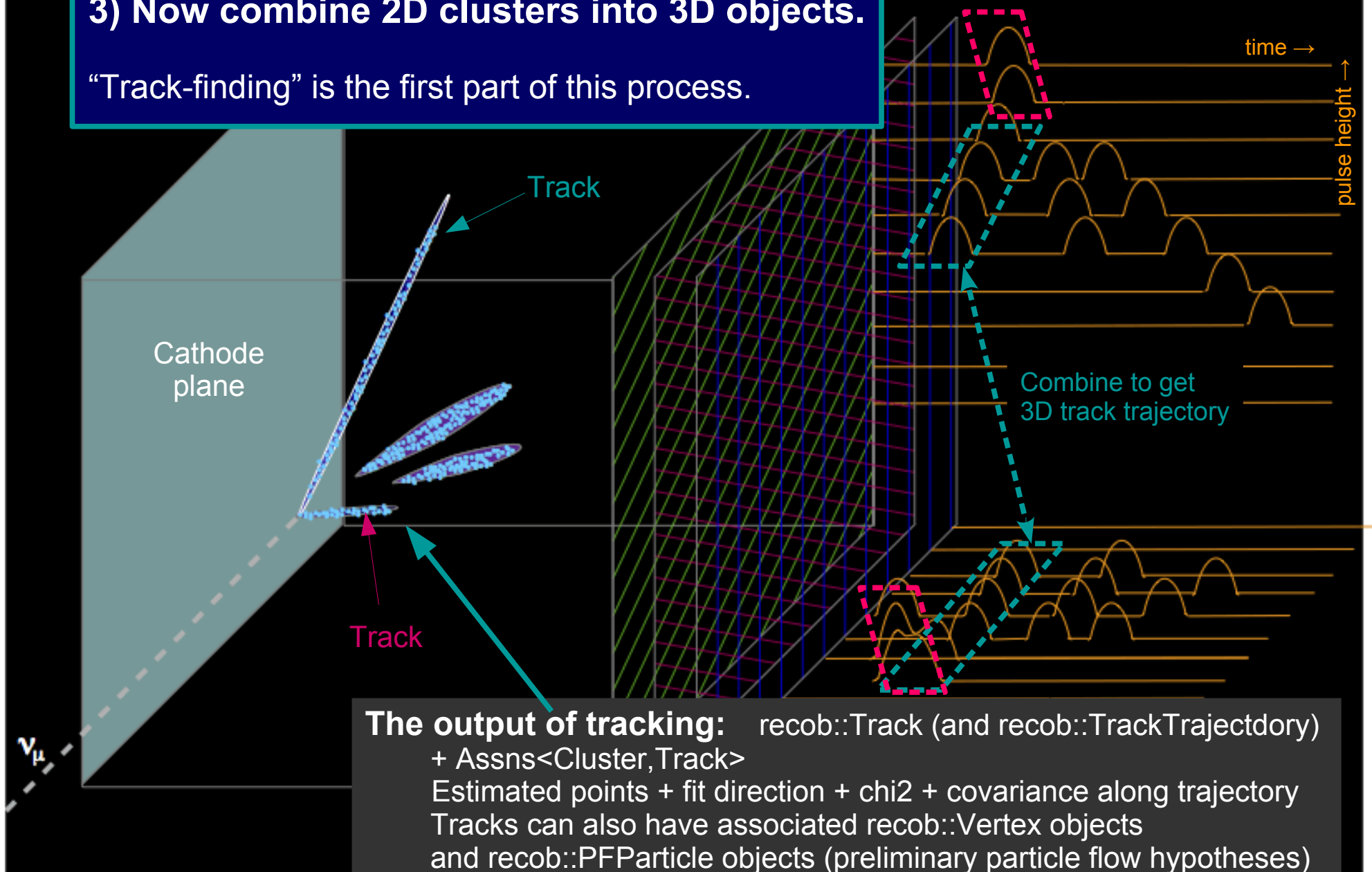
“Track-finding” is the first part of this process.



# Reconstruction workflow and data structures

## 3) Now combine 2D clusters into 3D objects.

“Track-finding” is the first part of this process.

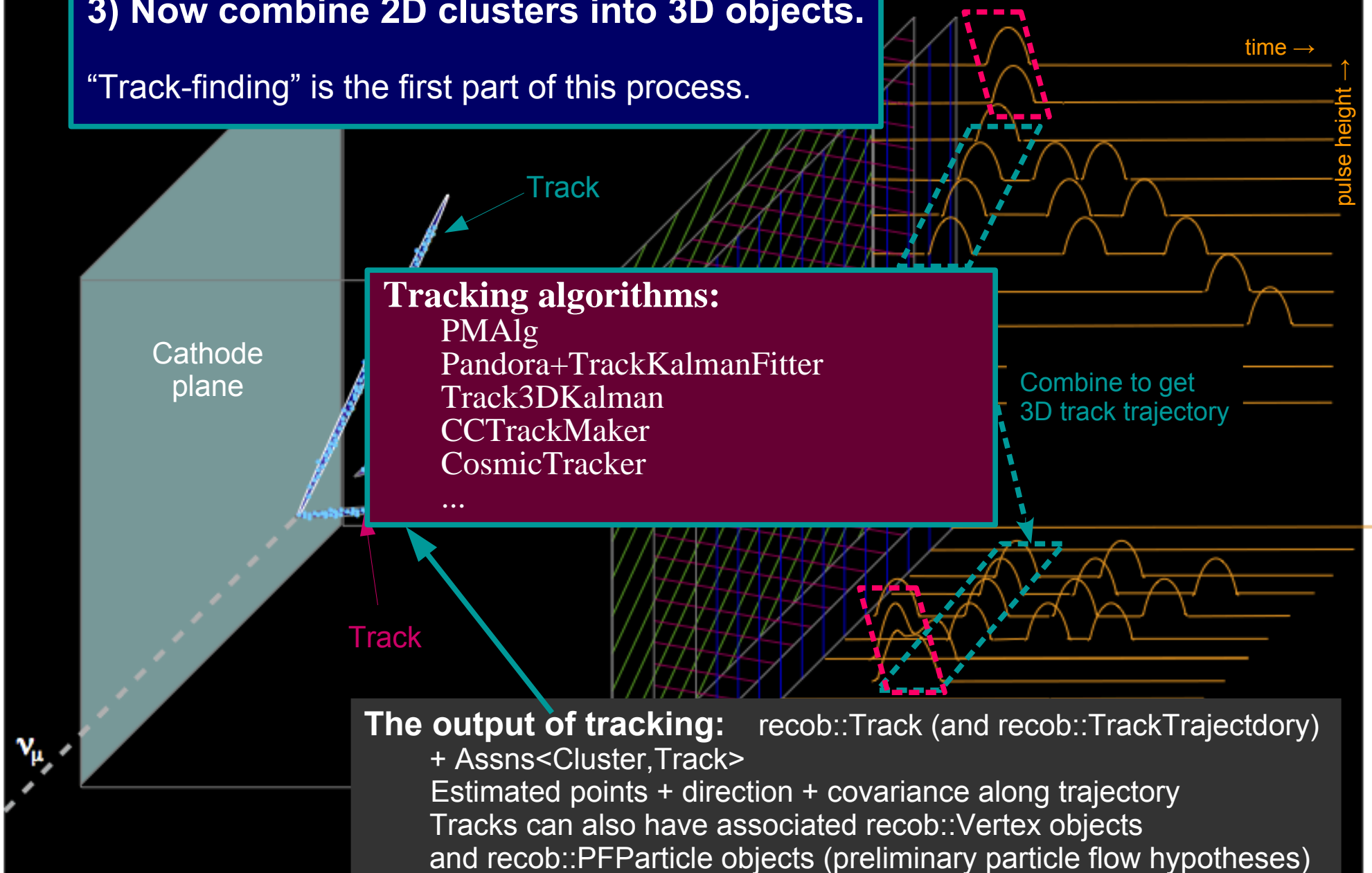


**The output of tracking:** `recob::Track` (and `recob::TrackTrajectory`)  
+ `Assns<Cluster,Track>`  
Estimated points + fit direction + chi2 + covariance along trajectory  
Tracks can also have associated `recob::Vertex` objects  
and `recob::PFParticle` objects (preliminary particle flow hypotheses)

# Reconstruction workflow and data structures

## 3) Now combine 2D clusters into 3D objects.

“Track-finding” is the first part of this process.

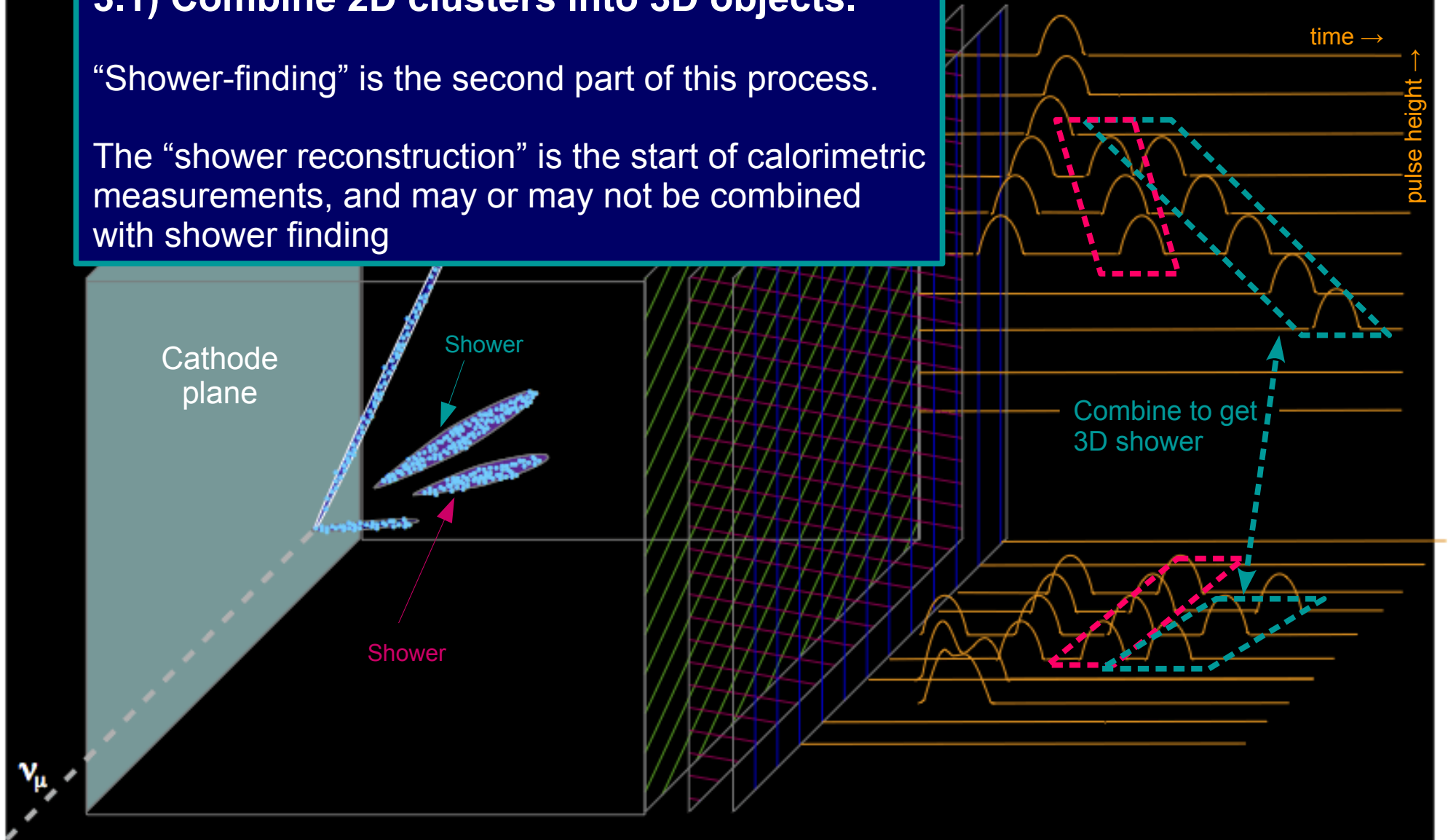


# Reconstruction workflow and data structures

## 3.1) Combine 2D clusters into 3D objects.

“Shower-finding” is the second part of this process.

The “shower reconstruction” is the start of calorimetric measurements, and may or may not be combined with shower finding

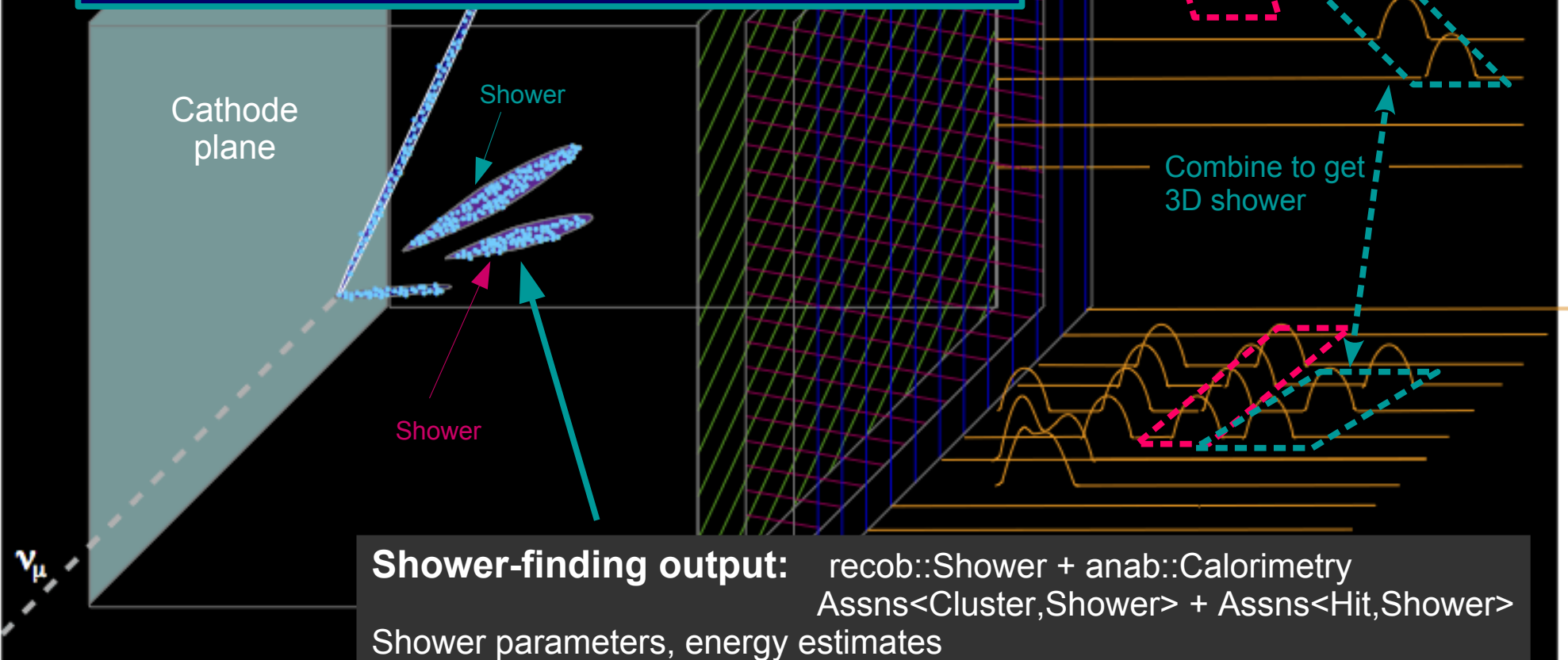


# Reconstruction workflow and data structures

## 3.1) Combine 2D clusters into 3D objects.

“Shower-finding” is the second part of this process.

The “shower reconstruction” is the start of calorimetric measurements, and may or may not be combined with shower finding





# Reconstruction workflow and data structures

## 3.1) Combine 2D clusters into 3D objects.

“Shower-finding” is the second part of this process.

The “shower reconstruction” is the start of calorimetric measurements, and may or may not be combined with shower finding

Cathode  
pla

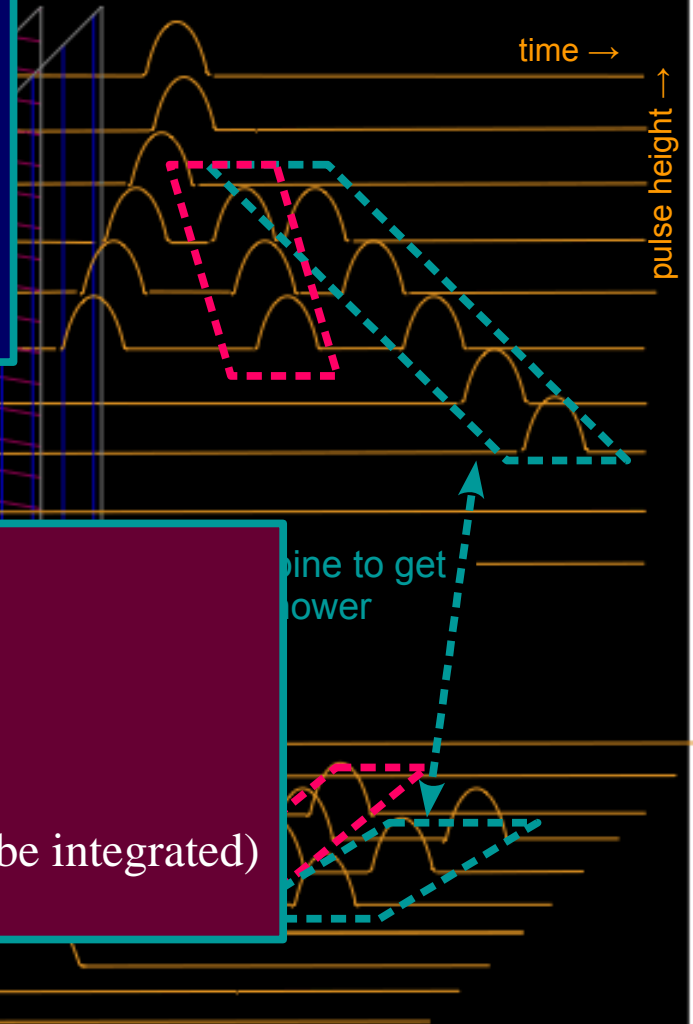
Shower

### Shower-finding algorithms:

EMShower  
ShowerReco3D  
PMA1g  
Pandora  
(some private uBooNE analysis code soon to be integrated)  
...

line to get  
lower

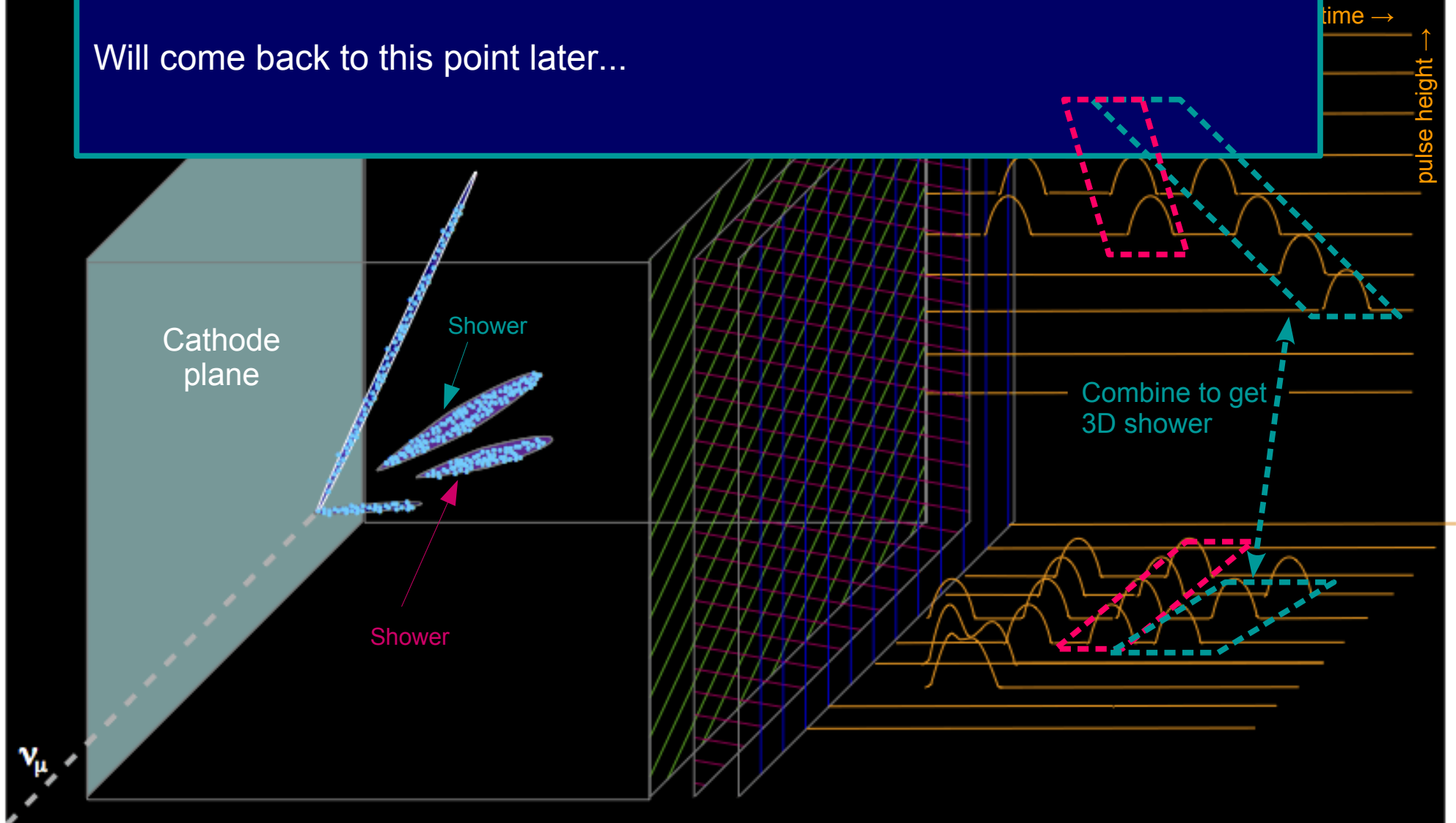
**Shower-finding output:** recob::Shower + anab::Calorimetry  
Assns<Cluster,Shower> + Assns<Hit,Shower>  
Shower parameters, energy estimates



# Reconstruction workflow and data structures

## 3.2) Can assemble 2D clusters into hierarchies of 3D objects

Will come back to this point later...



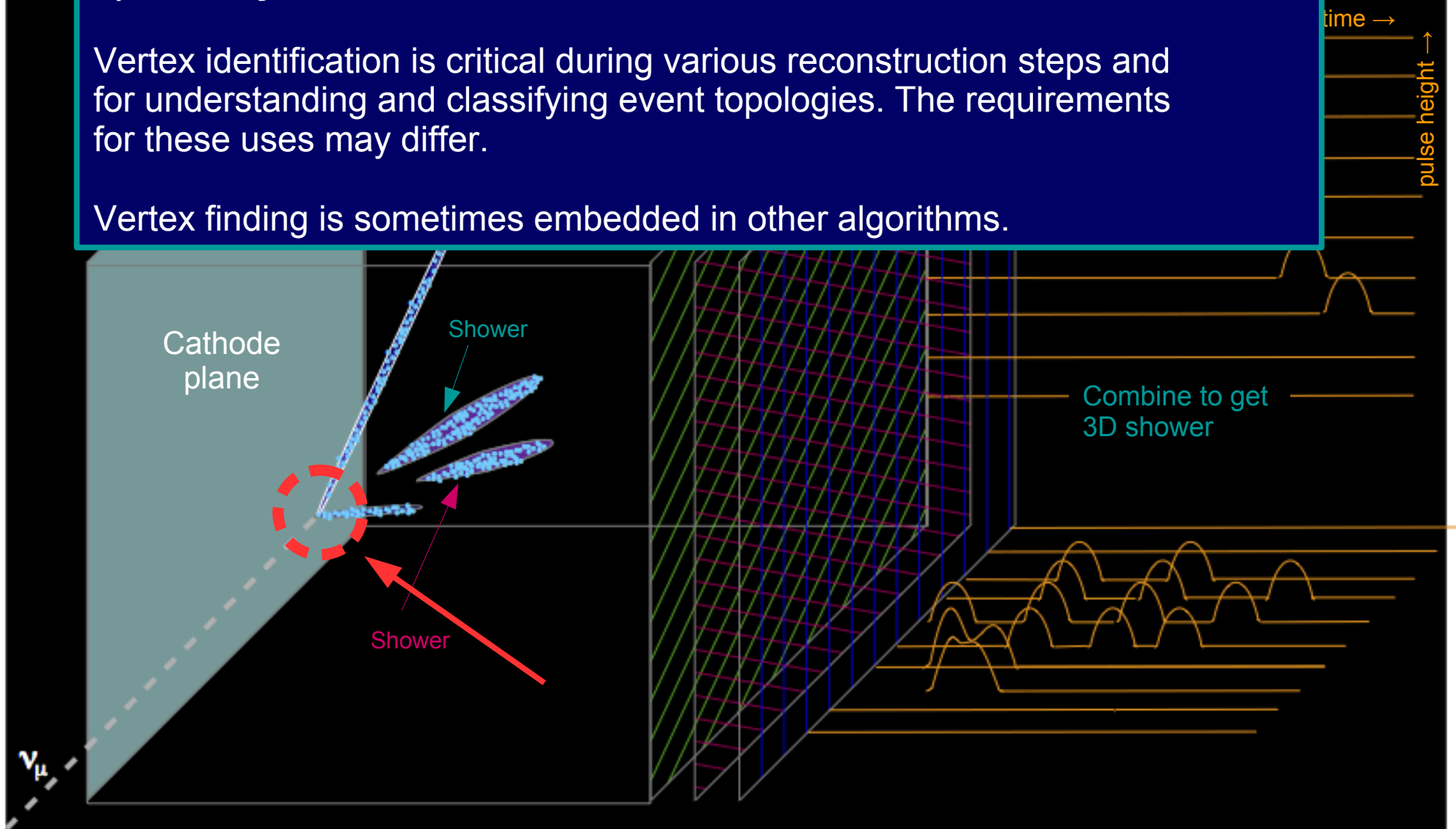


# Reconstruction workflow and data structures

## 4) Identify vertices in the event from tracks in the event

Vertex identification is critical during various reconstruction steps and for understanding and classifying event topologies. The requirements for these uses may differ.

Vertex finding is sometimes embedded in other algorithms.

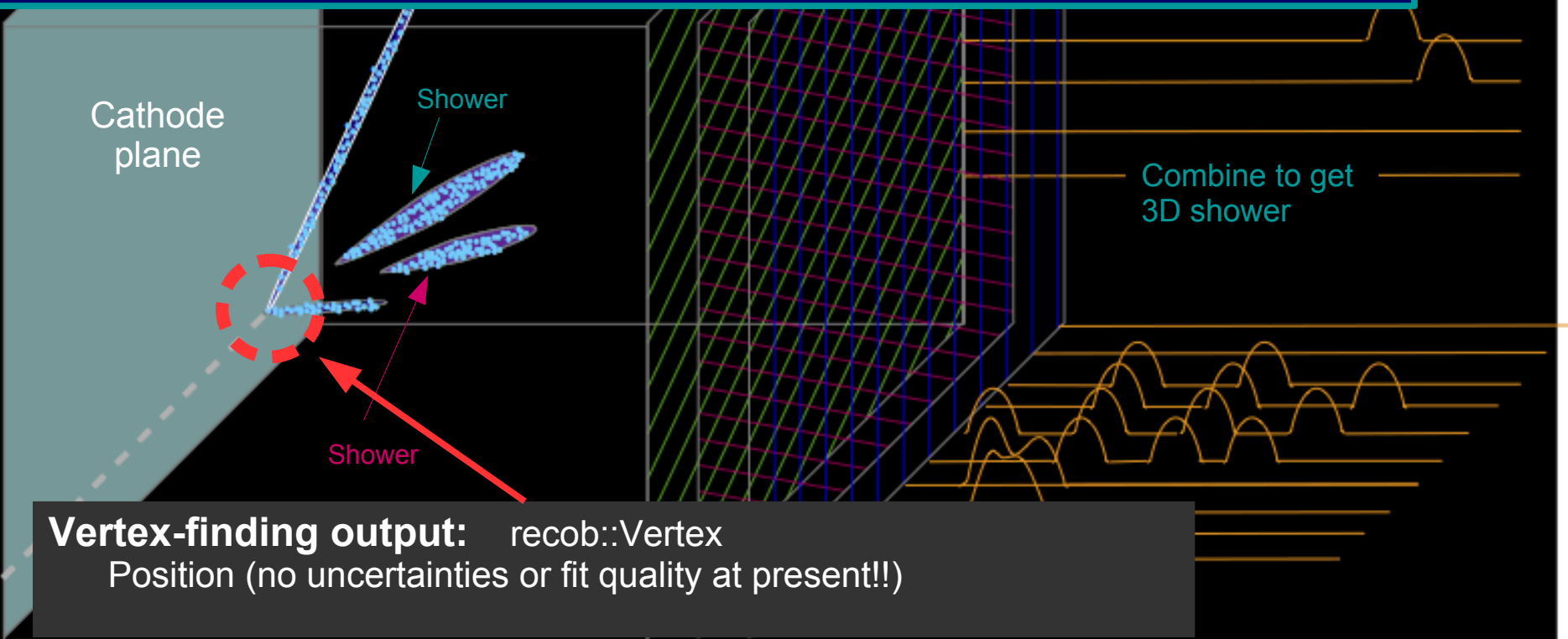


# Reconstruction workflow and data structures

## 4) Identify vertices in the event from tracks in the event

Vertex identification is critical during various reconstruction steps and for understanding and classifying event topologies. The requirements for these uses may differ.

Vertex finding is sometimes embedded in other algorithms.



# Reconstruction workflow and data structures

## 4) Identify vertices in the event from tracks in the event

Vertex identification is critical during various reconstruction steps and for understanding and classifying event topologies. The requirements for these uses may differ.

Vertex finding is sometimes embedded in other algorithms.

time →  
pulse height →

Cathode plane

Shower

Shower

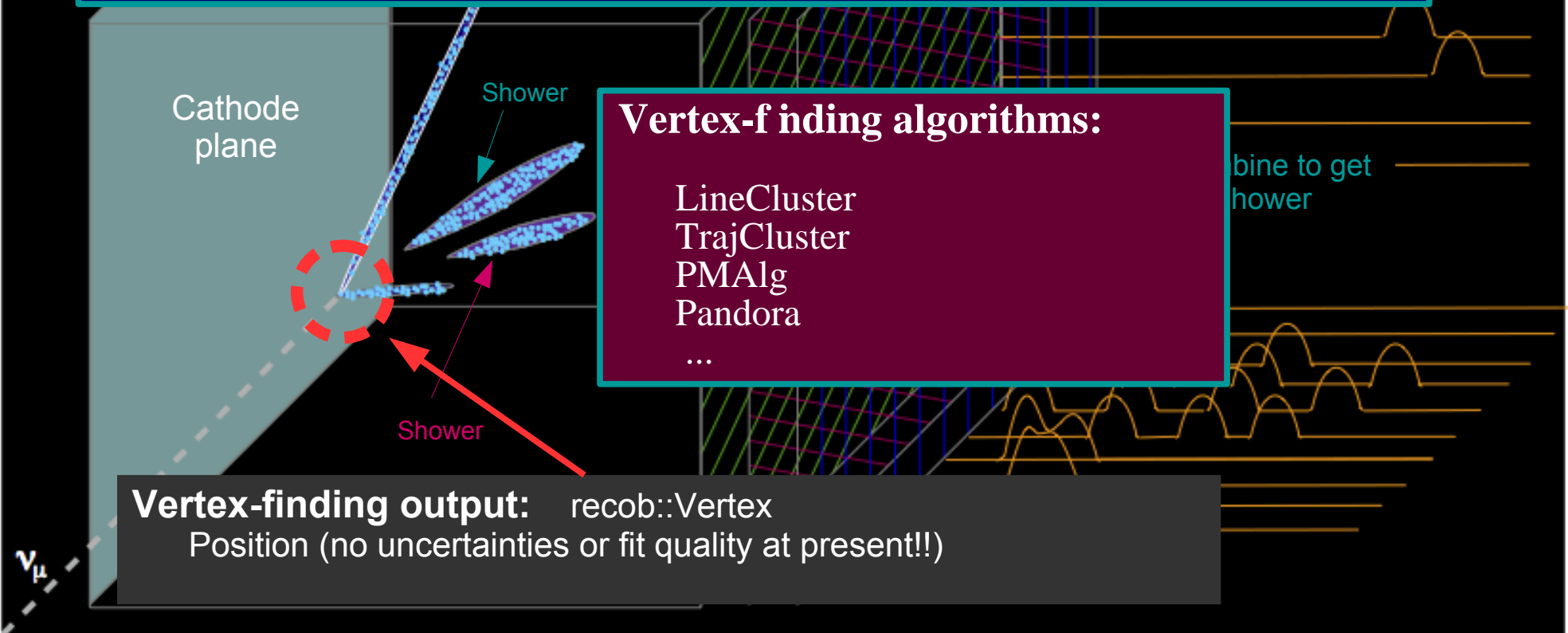
### Vertex-finding algorithms:

LineCluster  
TrajCluster  
PMAlg  
Pandora  
...

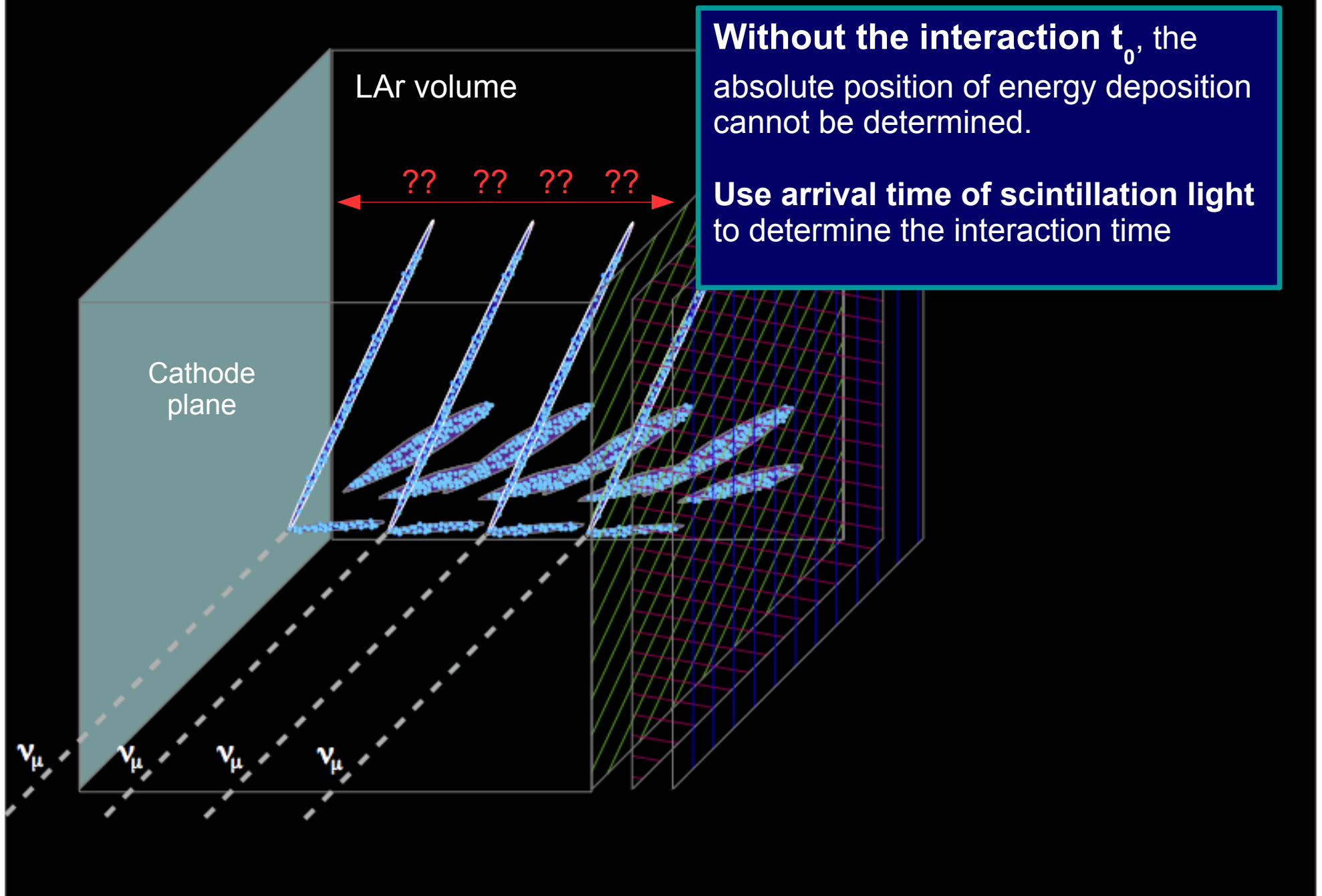
combine to get shower

**Vertex-finding output:** recob::Vertex  
Position (no uncertainties or fit quality at present!!)

$v_{\mu}$



# Reconstruction workflow and data structures



# Reconstruction workflow and data structures

LAr volume

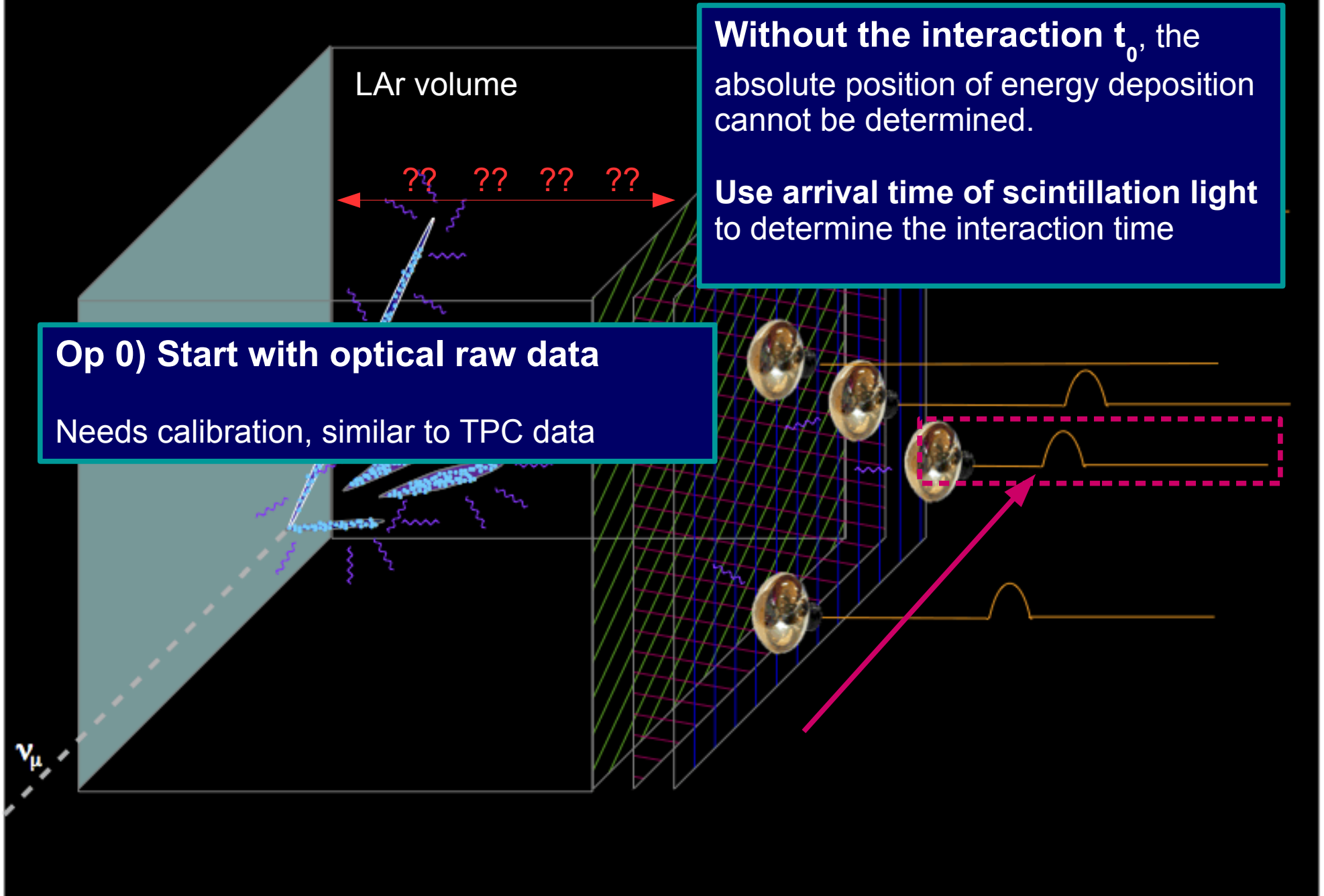
Without the interaction  $t_0$ , the absolute position of energy deposition cannot be determined.

Use arrival time of scintillation light to determine the interaction time

Op 0) Start with optical raw data

Needs calibration, similar to TPC data

$\nu_\mu$



# Reconstruction workflow and data structures

LAr volume

Without the interaction  $t_0$ , the absolute position of energy deposition cannot be determined.

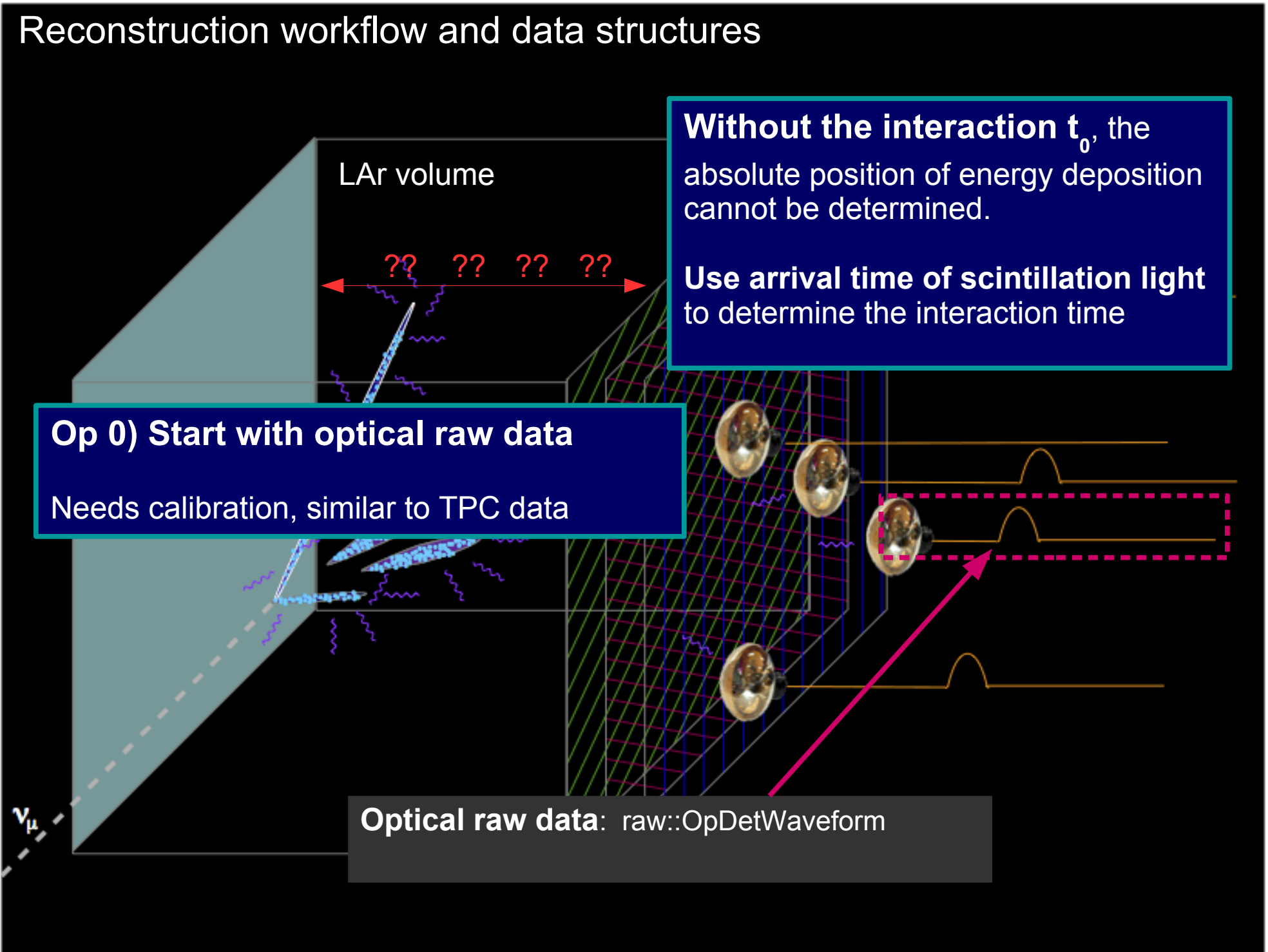
Use arrival time of scintillation light to determine the interaction time

Op 0) Start with optical raw data

Needs calibration, similar to TPC data

Optical raw data: `raw::OpDetWaveform`

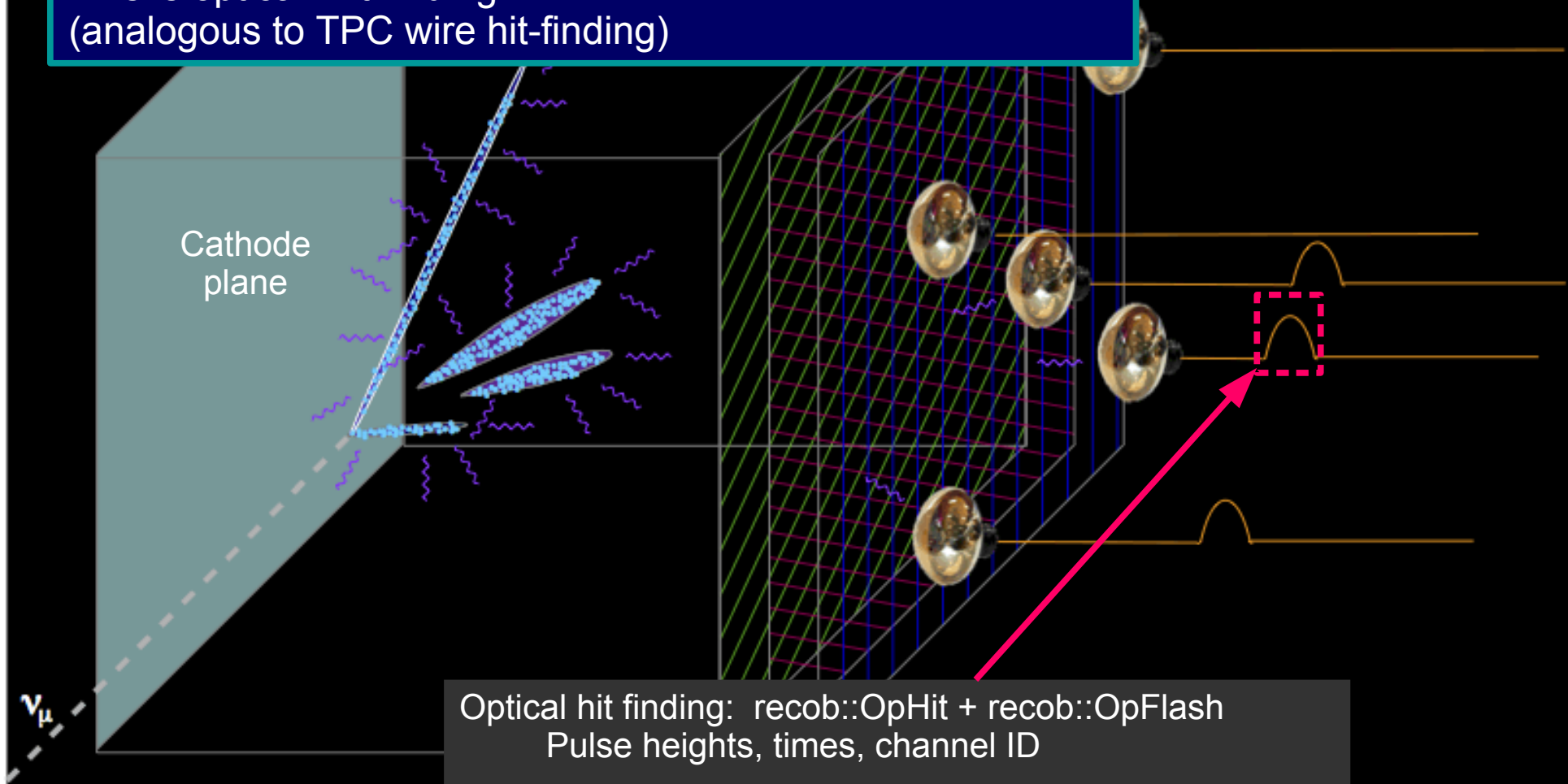
$\nu_\mu$



# Reconstruction workflow and data structures

**Op 1) First identify signals from single interactions on each optical channel**  
on each optical channel

This is optical “hit-finding”  
(analogous to TPC wire hit-finding)



Optical hit finding: `recob::OpHit` + `recob::OpFlash`  
Pulse heights, times, channel ID



# Reconstruction workflow and data structures

**Op 1) First identify signals from single interactions on each optical channel**  
on each optical channel

This is optical “hit-finding”  
(analogous to TPC wire hit-finding)

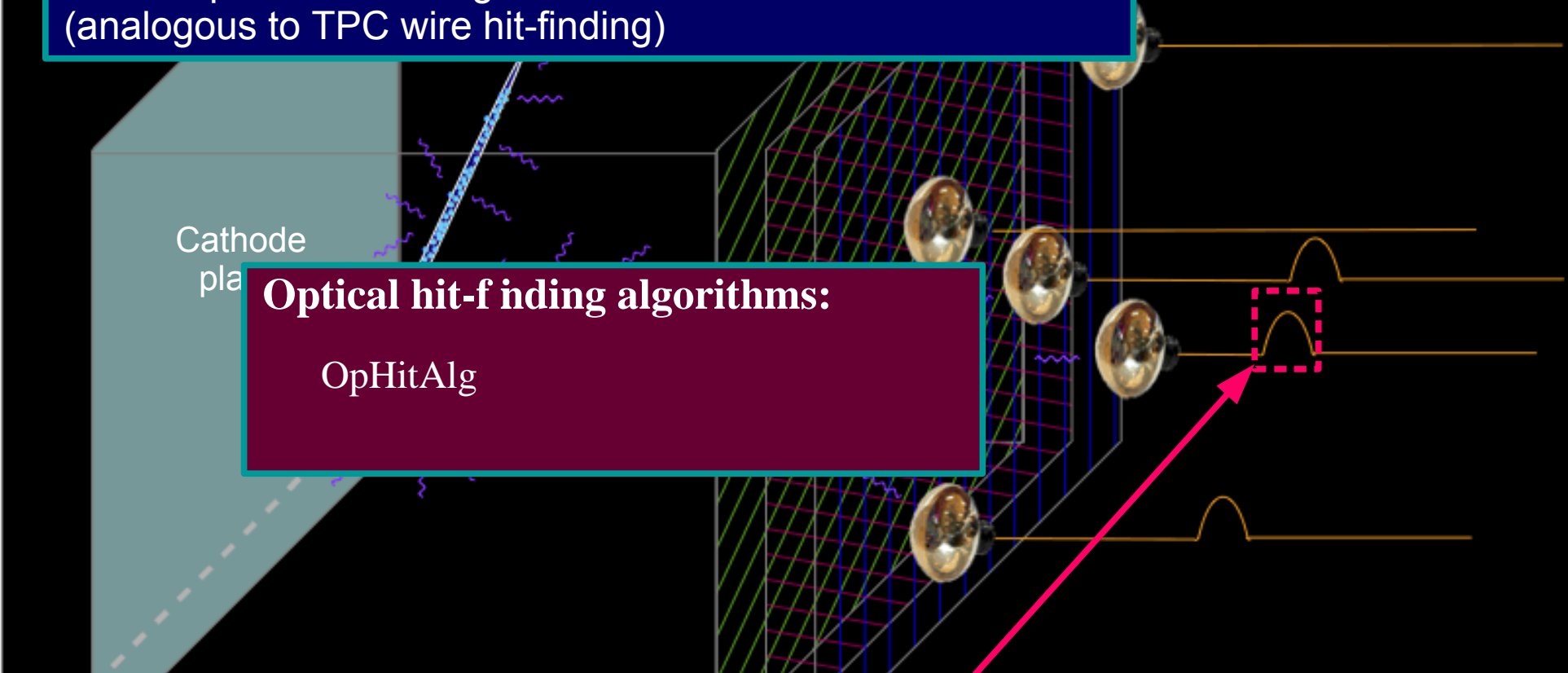
Cathode  
pla

**Optical hit-finding algorithms:**

OpHitAlg

Optical hit finding: `recob::OpHit + recob::OpFlash`  
Pulse heights, times, channel ID

$v_{\mu}$



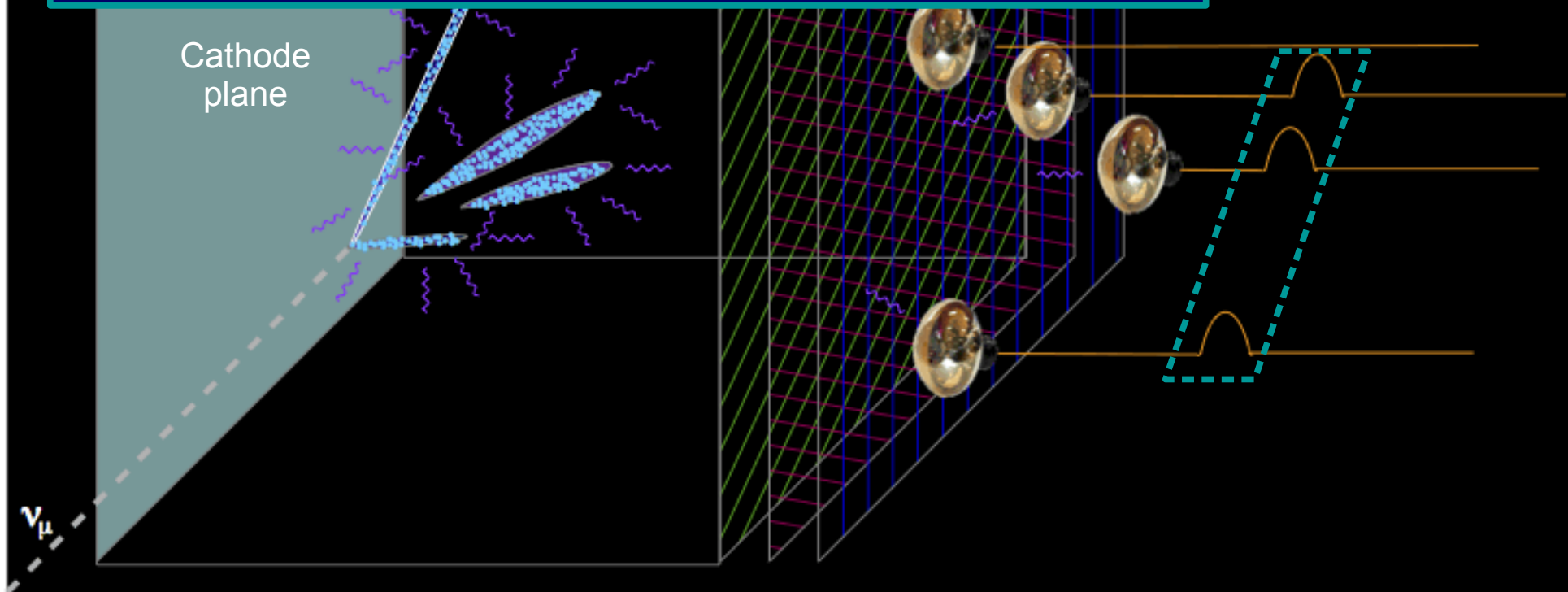


# Reconstruction workflow and data structures

**Op 2) Find all hits associated with a single interaction in the TPC.**

All tracks, showers from a single interaction produce a “flash”.  
“Flash-finding” identifies all such associated hits.

The combination of hit and flash finding is sometimes called  
“optical reconstruction”

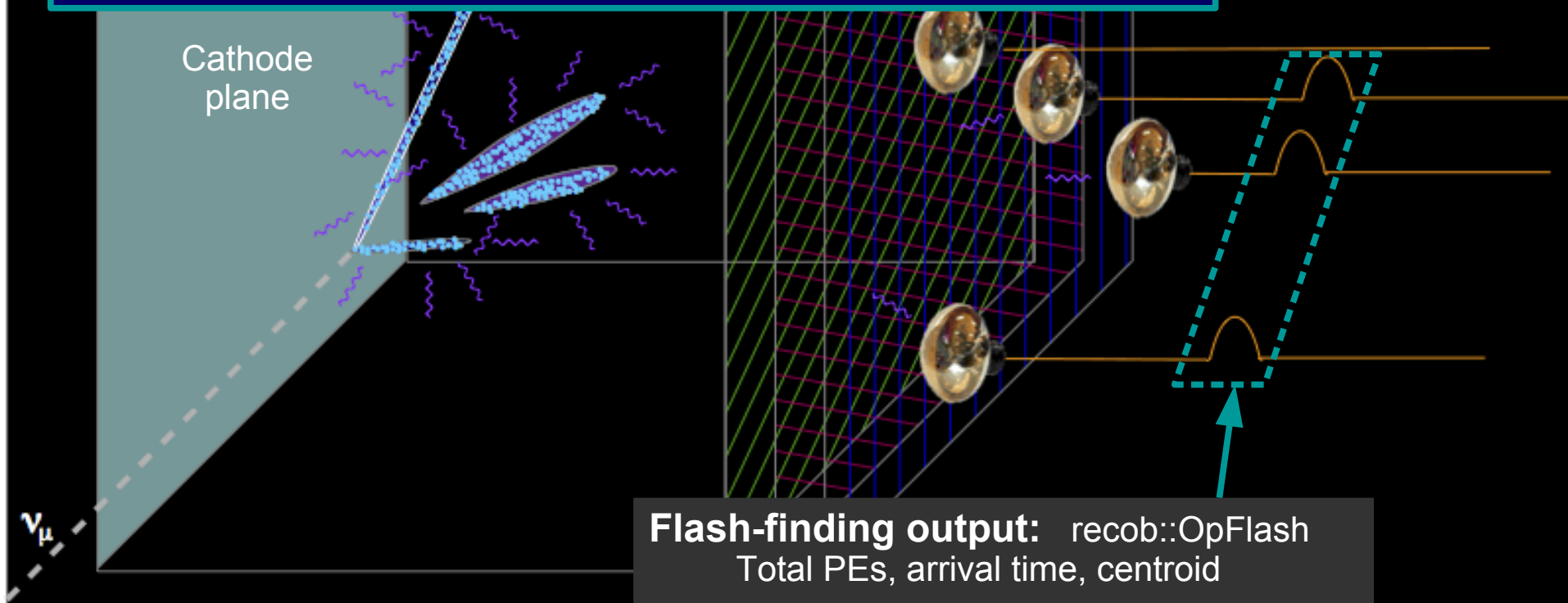


# Reconstruction workflow and data structures

**Op 2) Find all hits associated with a single interaction in the TPC.**  
in the TPC.

All tracks, showers from a single interaction produce a “flash”.  
“Flash-finding” identifies all such associated hits.

The combination of hit and flash finding is sometimes called  
“optical reconstruction”

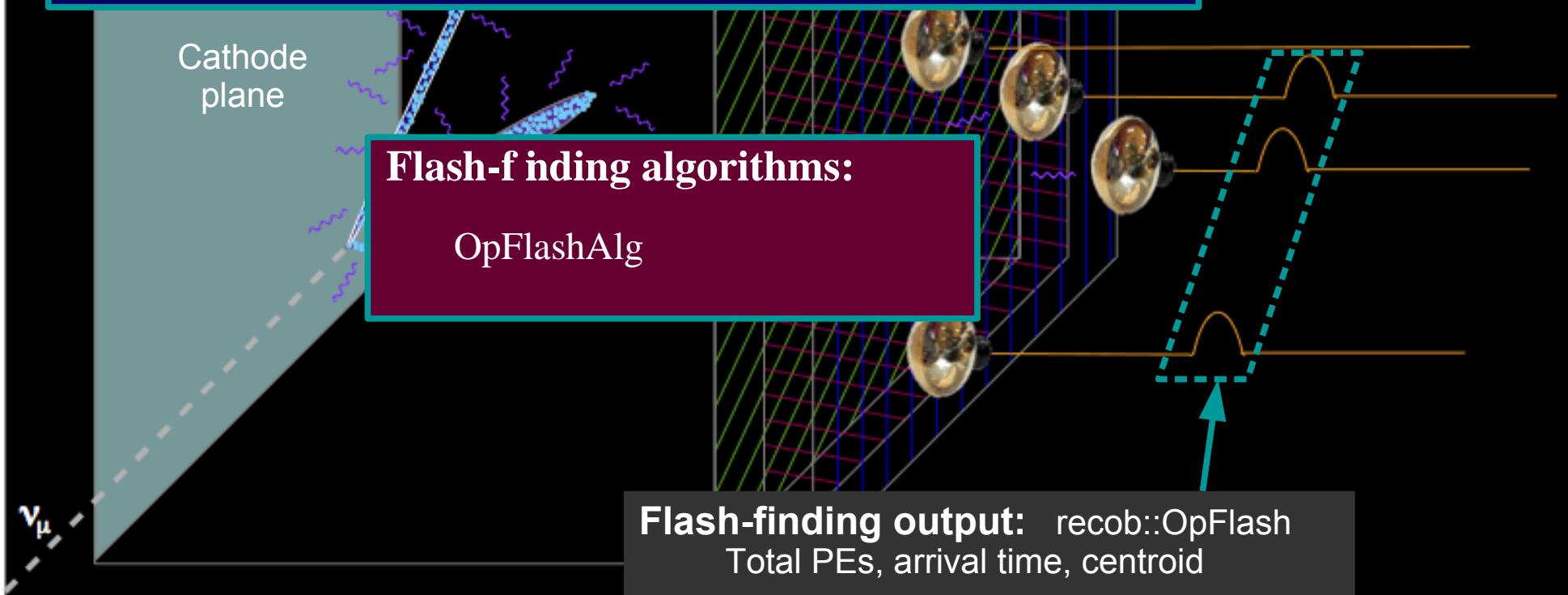


# Reconstruction workflow and data structures

**Op 2) Find all hits associated with a single interaction in the TPC.**  
in the TPC.

All tracks, showers from a single interaction produce a “flash”.  
“Flash-finding” identifies all such associated hits.

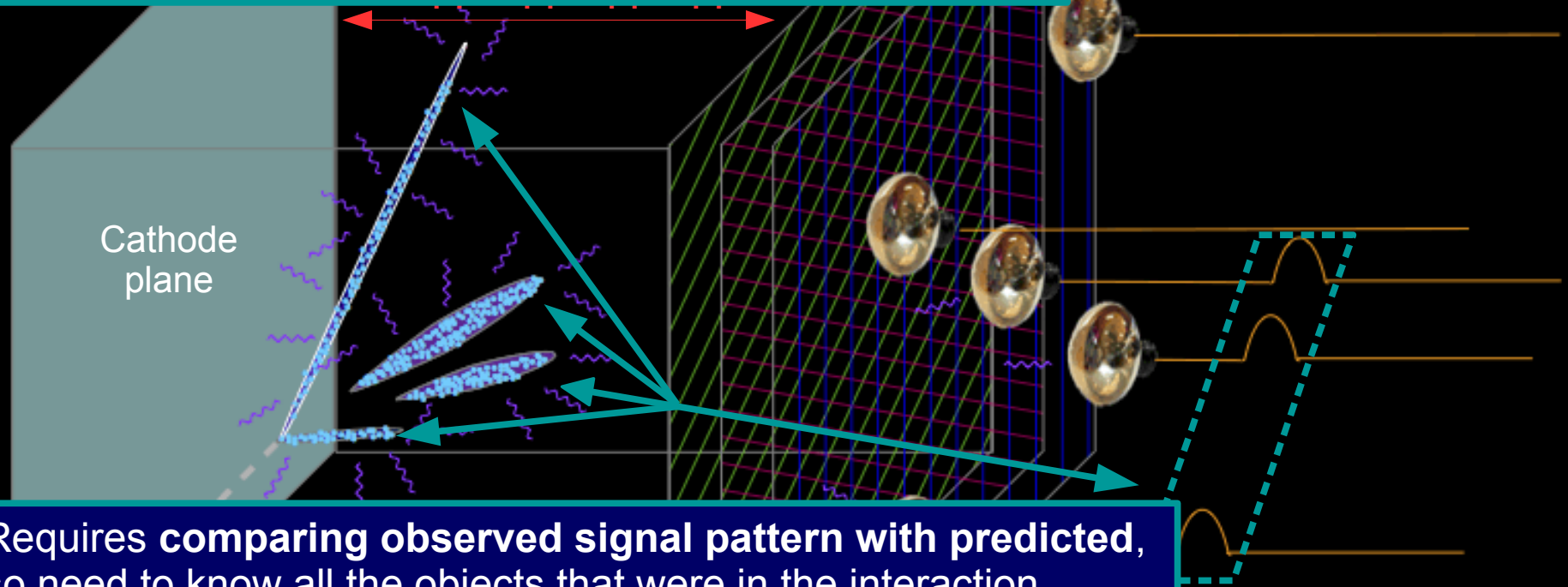
The combination of hit and flash finding is sometimes called  
“optical reconstruction”



# Reconstruction workflow and data structures

**Op 3) Now match a given flash to a set of objects**  
from a single interaction event in the TPC

This step is “flash-matching”



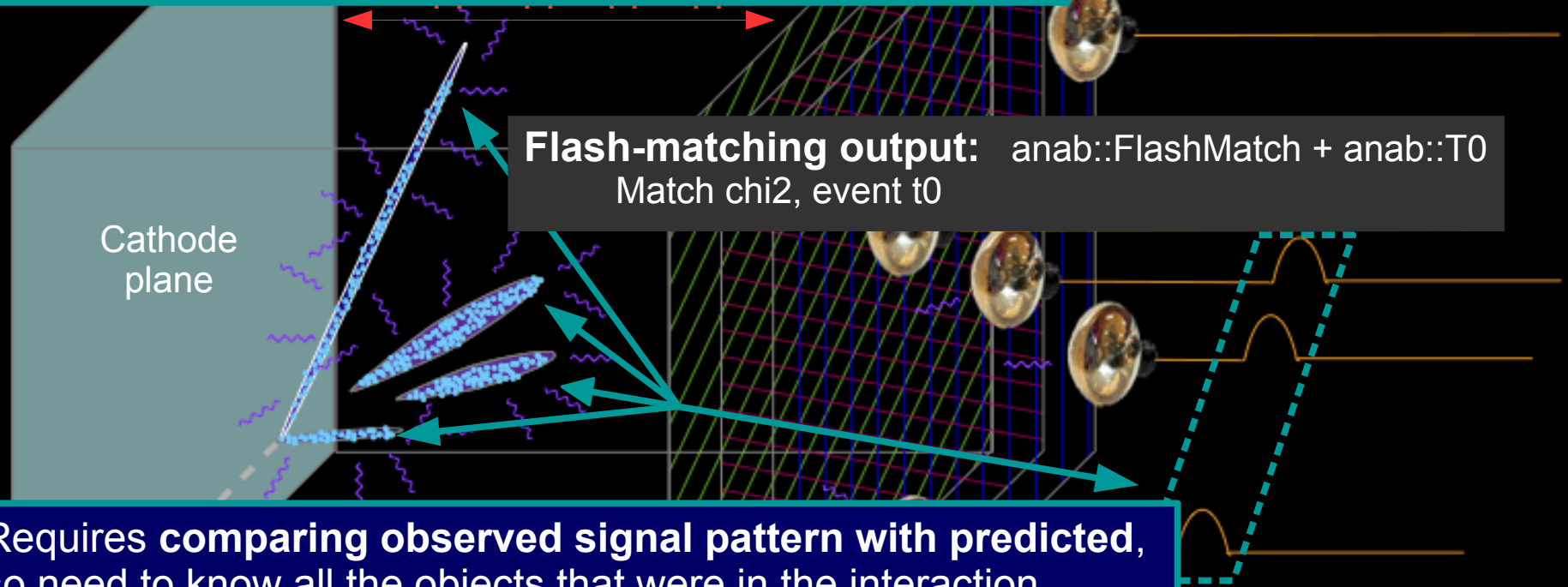
Requires **comparing observed signal pattern with predicted**,  
so need to know all the objects that were in the interaction  
**Estimate event  $t_0$**  from this process.

Because it requires an event hypothesis, **may be performed**  
**at analysis-level** rather than at primary reconstruction

# Reconstruction workflow and data structures

**Op 3) Now match a given flash to a set of objects**  
from a single interaction event in the TPC

This step is “flash-matching”



**Flash-matching output:** `anab::FlashMatch + anab::T0`  
Match  $\chi^2$ , event  $t_0$

Requires **comparing observed signal pattern with predicted**,  
so need to know all the objects that were in the interaction  
**Estimate event  $t_0$**  from this process.

Because it requires an event hypothesis, **may be performed**  
**at analysis-level** rather than at primary reconstruction

# Special algorithm notes

# Pandora

- Multi-algorithm pattern recognition framework

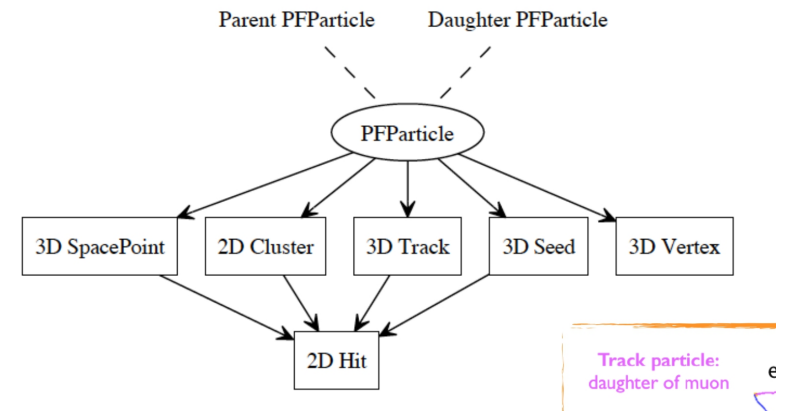
See [the intro at Jan 2017 DUNE collab meeting](#), and [the LArSoft Note at larsoft.org/pandora](http://larsoft.org/pandora)

- Takes `recob::Hit` collections as input, and performs

- 2D cluster finding (track-like and shower-like)
- 3D matching
- shower/track discrimination
- vertex finding and classification

- Produces `recob::PFParticles`

- Mother-daughter hierarchy for all particles from a single event vertex
  - Distinguish tracks, delta rays, showers, vertices



# Pandora

- Multi-algorithm pattern recognition framework

See [the intro at Jan 2017 DUNE collab meeting](#), and [the LArSoft Note at larsoft.org/pandora](http://larsoft.org/pandora)

- Takes reco

- 2D cl
- 3D m
- show
- verte

- Produces

- Moth
- from
- 
- v

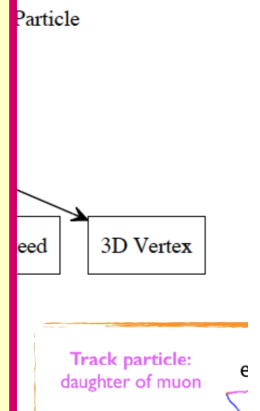
## Framework operates externally to LArSoft

- Dedicated, direct interface modules and algorithms in LArSoft

- larpandora, larpandoracontent repositories

- Typically run in stages with intervening LArSoft algorithms

- e.g., Pandora cosmic tagging
- LArSoft cosmic hit removal
- Pandora neutrino event reco





# Wire-cell

- 3D object reconstruction from 2D time slices

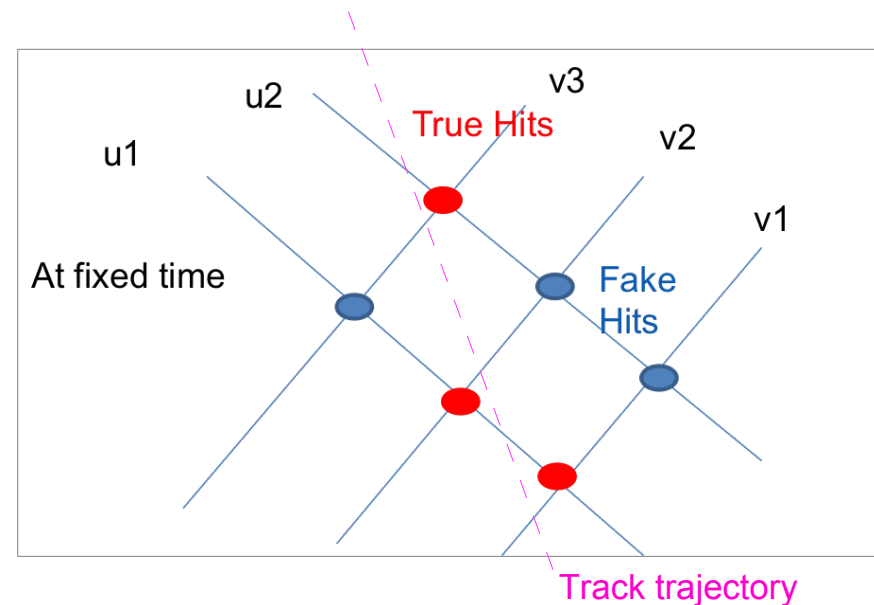
See <http://www.phy.bnl.gov/wire-cell/>

- Uses charge as constraint to match hits at each wire crossing

- Demanding signal processing requirements
  - Significant contributions to LArSoft in this area already
- Computationally challenging, but lots of progress on this over the past year
- High reward (3D from waveforms only!)

- Developed externally to LArSoft

- Direct, multipoint interfacing to LArSoft in progress



# Deep learning networks

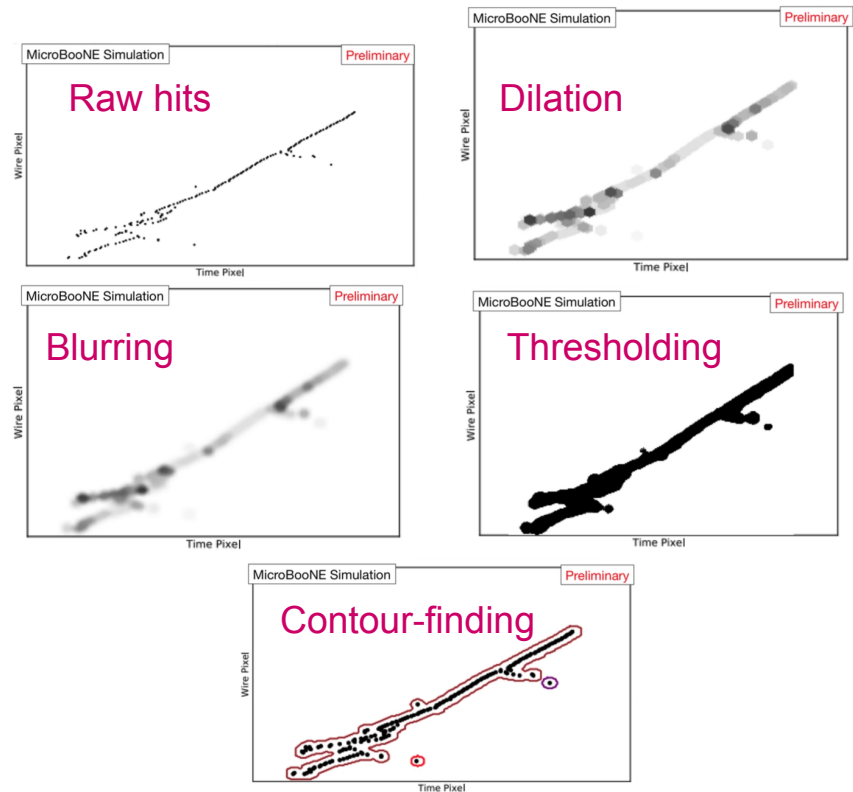
- A variety of efforts are in progress
  - At present, aimed mainly at event classification, region of interest identification, track/shower discrimination
- One already integrated into LArSoft
 

[See talk by P. Plonski, D. Stefan, R. Sulej on Wednesday](#)

  - Provide hit-level shower/track discrimination and vertex identification upstream of conventional LArSoft algorithms
- Much to learn about how to use these types of algorithms
  - Envision continuing integration work

# Image processing techniques

- Effort underway to perform 2D clustering with image processing software (MicroBooNE)
  - Topology-based clustering
  - OpenCV product applied to 2D TPC image data
  - interesting early results
- Future work
  - How to use this technique
  - How to integrate into LArSoft



From [MicroBooNE-Note-1012-Pub](#)

## Other techniques...

...that I've not mentioned, or am not aware of

Still lots of room for innovation, evolution, so be bold!!

# Additional reconstruction considerations

# Additional reconstruction phases

- Cosmic ray removal

- Particularly important for surface detectors
  - SBN detectors at Fermilab
  - Test beam detectors
- Current algorithms primarily geometry-based
  - Look for out of time tracks, or in-time tracks that cross a boundary
  - CR hits can be removed for downstream clustering / tracking / shower-finding
- Representative algorithms:
  - CosmicTrackTagger
  - CosmicPFParticleTagger
- Output: `anab::CosmicTag`



# Additional reconstruction phases

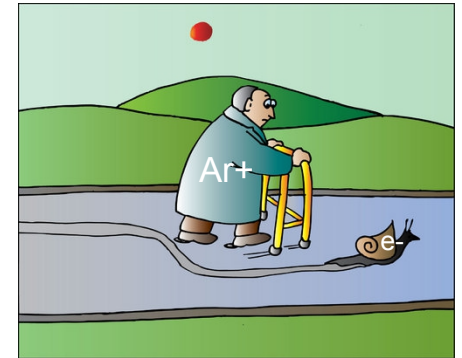


- Non-shower calorimetric measurements
  - Energy and  $dE/dx$  estimates for Tracks
  - Representative algorithms: CalorimetryAlg, TrackCalorimetryAlg
  - Output: `anab::Calorimetry`
- Momentum estimation and particle identification
  - Use range,  $dE/dx$  and multiple Coulomb scattering of tracks
  - Representative algorithms:
    - TrackKalmanFitter, TrajCluster yield MCS momentum
    - Chi2PIDAlg, PIDAAlg perform particle ID
  - Output: `anab::ParticleID`, `Assns<Track, ParticlePID>`, or `TTree`

# Other complications

- Space-charge distortions

- Ion drift mobilities about  $10^6\times$  smaller than for electrons
  - Cation drift velocities are  $\sim \text{nm} / \mu\text{s} !!$
- **For surface detectors**, cosmic rays introduces large (+) ion load
  - At **ProtoDUNE SP**, electron drift distortions reach **few 10's cm scale**
  - Need to map and correct for these field distortions
- A common service interface exists to access the offsets
  - Allows experiment-dependent implementations



Alexei Talimonov, used w/o permission

- Charge attenuation

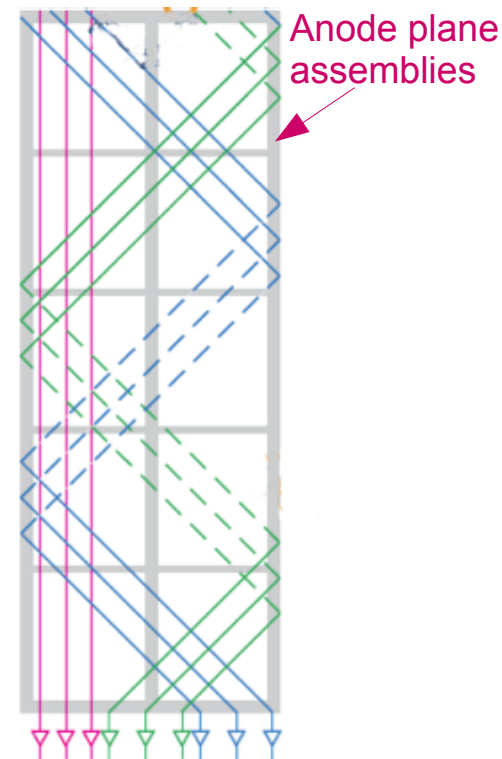
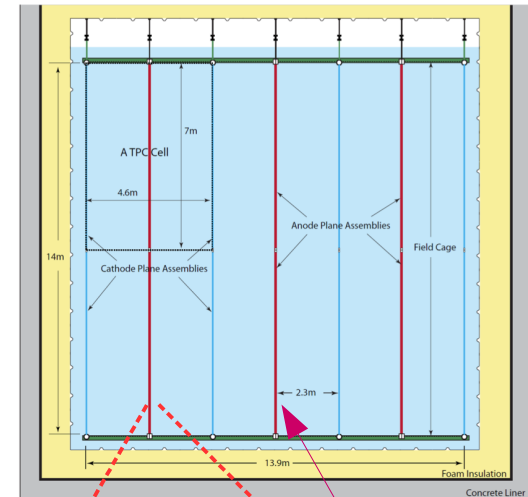
- Electron lifetime can be comparable to maximum drift time
- Charge yield at wires will be drift-length dependent
- Affects S:N ratio and charge / energy measurements



# Other complications

- Hit disambiguation

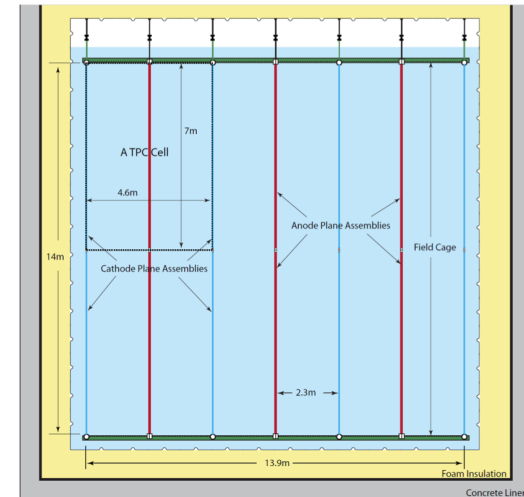
- DUNE TPCs have wrapped induction wires
  - See signals in two TPCs, and in many cases, multiple places within the same TPC
- Some LArSoft nomenclature:
  - “Wire”: a segment of a physical wire in an anode plane
    - A geometric concept
  - “Channel”: a readout channel connected to one or more “wires”
    - A DAQ concept



# Other complications

- Hit disambiguation

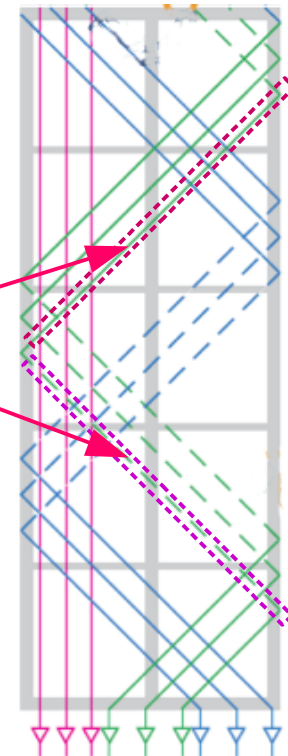
- DUNE TPCs have wrapped induction wires
  - See signals in two TPCs, and in many cases, multiple places within the same TPC



- Some LArSoft nomenclature:

- “Wire”: a segment of a physical wire in an anode plane
  - A geometric concept
- “Channel”: a readout channel connected to one or more “wires”
  - A DAQ concept

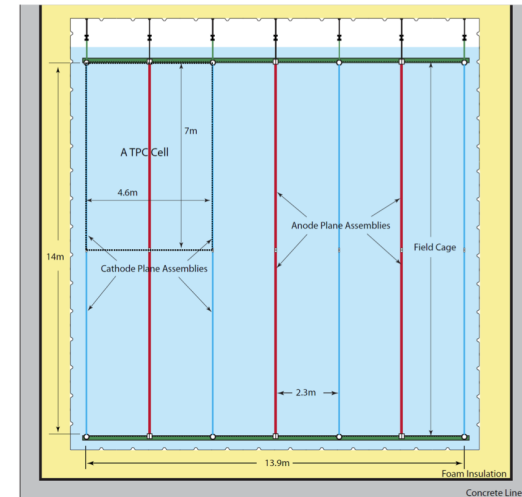
One “channel”  
(same physical wire)



# Other complications

- Hit disambiguation

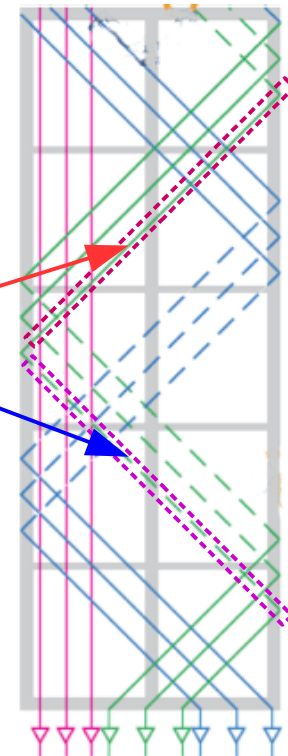
- DUNE TPCs have wrapped induction wires
  - See signals in two TPCs, and in many cases, multiple places within the same TPC



- Some LArSoft nomenclature:

- “Wire”: a segment of a physical wire in an anode plane
  - A geometric concept
- “Channel”: a readout channel connected to one or more “wires”
  - A DAQ concept

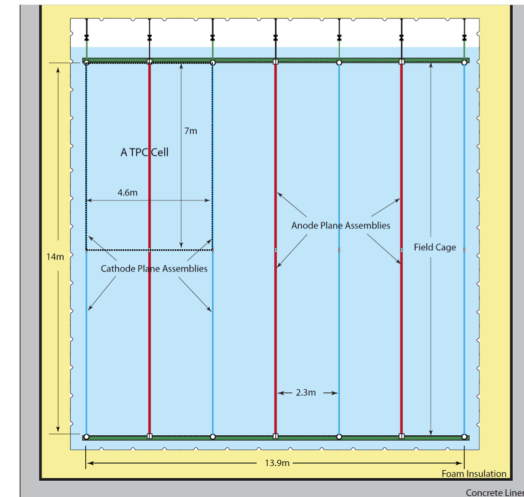
Two “wires”



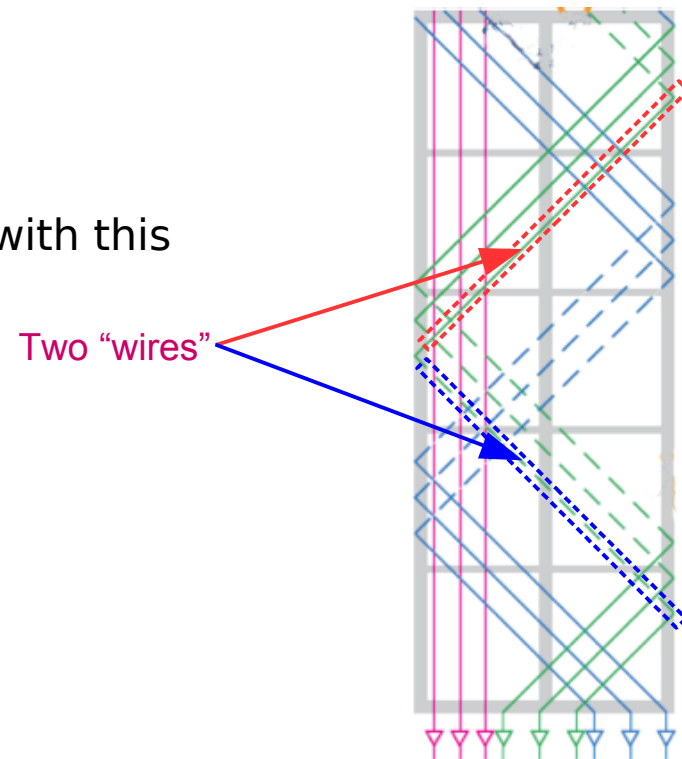
# Other complications

- Hit disambiguation

- DUNE TPCs have wrapped induction wires
  - See signals in two TPCs, and in many cases, multiple places within the same TPC



- Introduce a disambiguation step to deal with this
  - Resolves the TPC ambiguity of each induction hit
  - Currently performed after hit-finding
    - Existing algorithms use timing information and neighboring activity



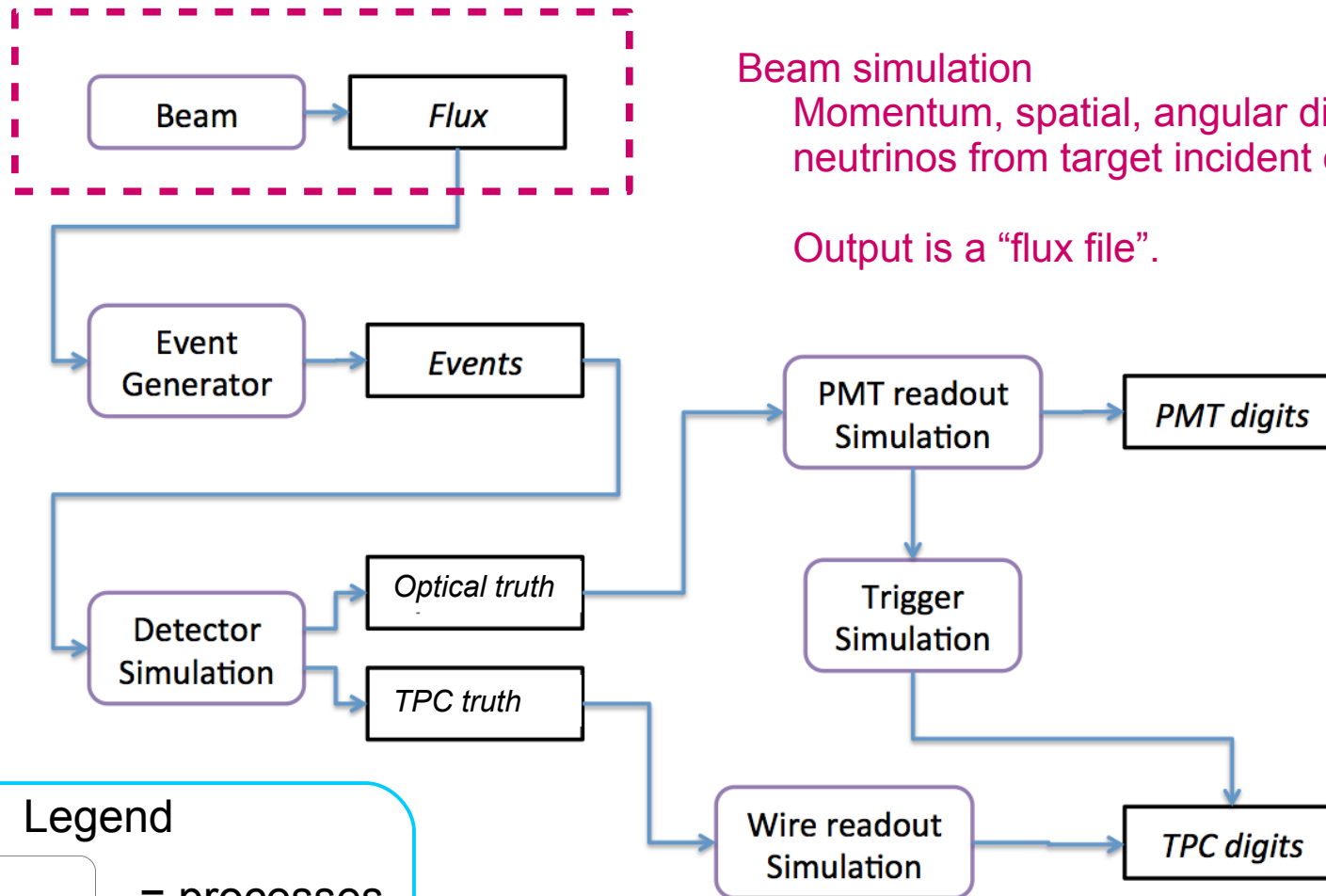
# Dual phase TPC

- ProtoDUNE DP
  - Significant progress in integrating DP TPC into LArSoft
    - See various [FD sim/reco talks at Jan 2017 DUNE Collaboration Meeting](#)
  - A number of needed changes to LArSoft recently completed
  - Work proceeding on
    - ProtoDUNE DP geometry (exists)
    - Specialized detector simulation and hit reconstruction
    - Calorimetric reconstruction
    - Core LArSoft reconstruction algorithm tests

# Detector simulation in LArSoft

# Simulation workflow

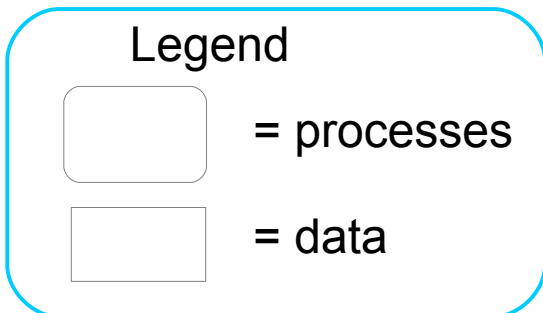
Three phases, typically run separately



## Beam simulation

Momentum, spatial, angular distribution of neutrinos from target incident on detector

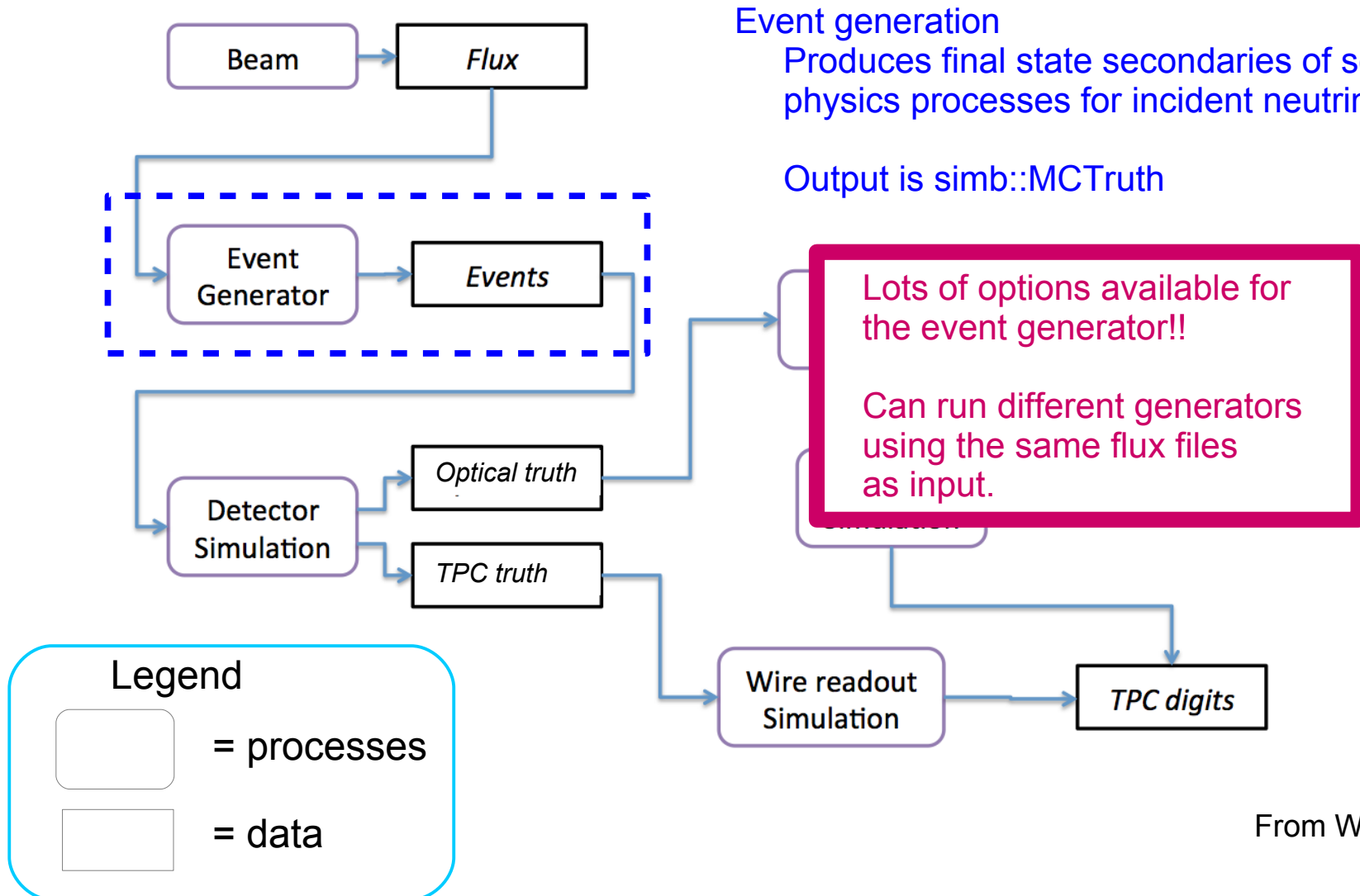
Output is a “flux file”.



From W Seligman

# Simulation workflow

Three phases, typically run separately

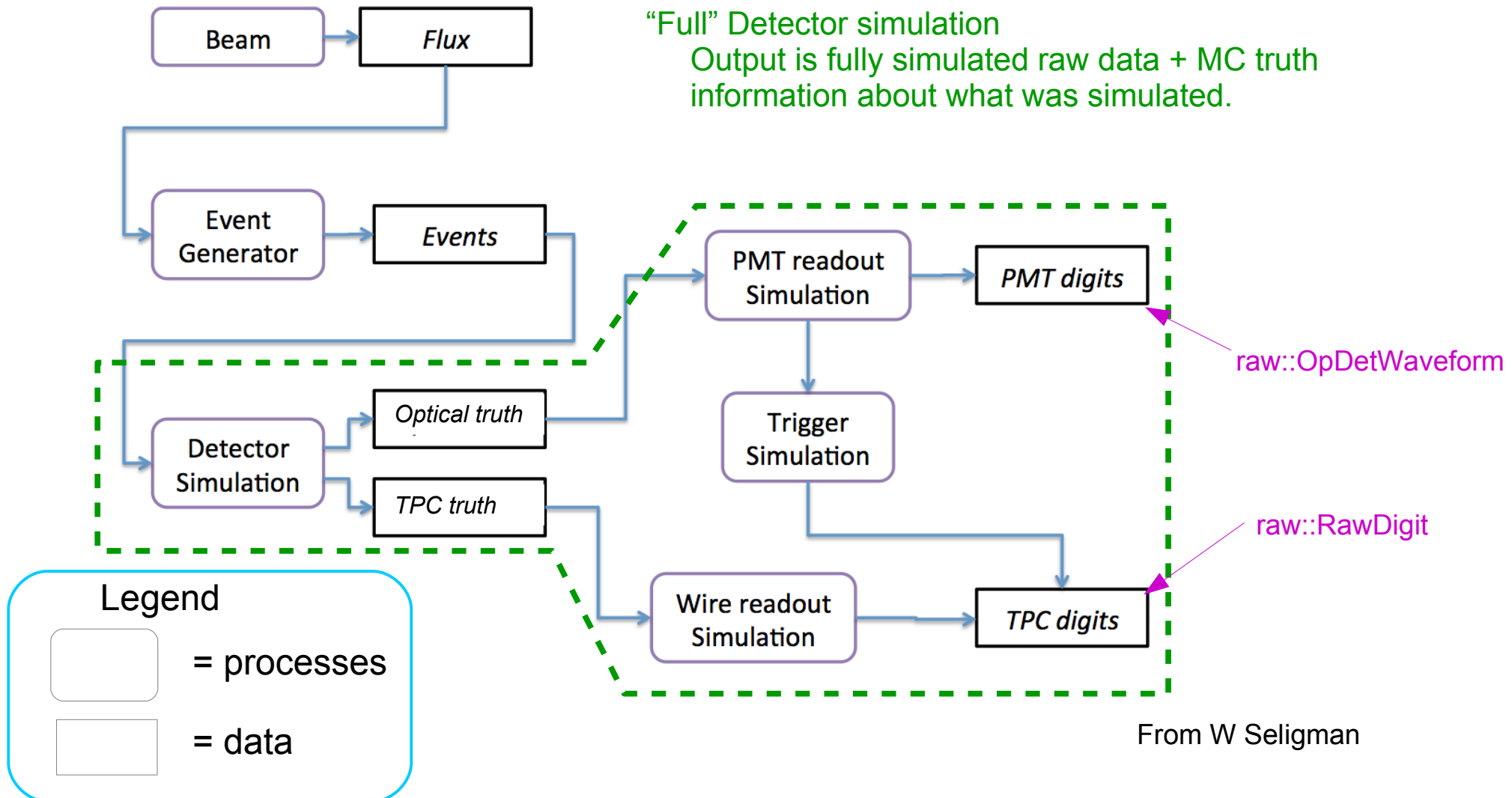


From W Seligman



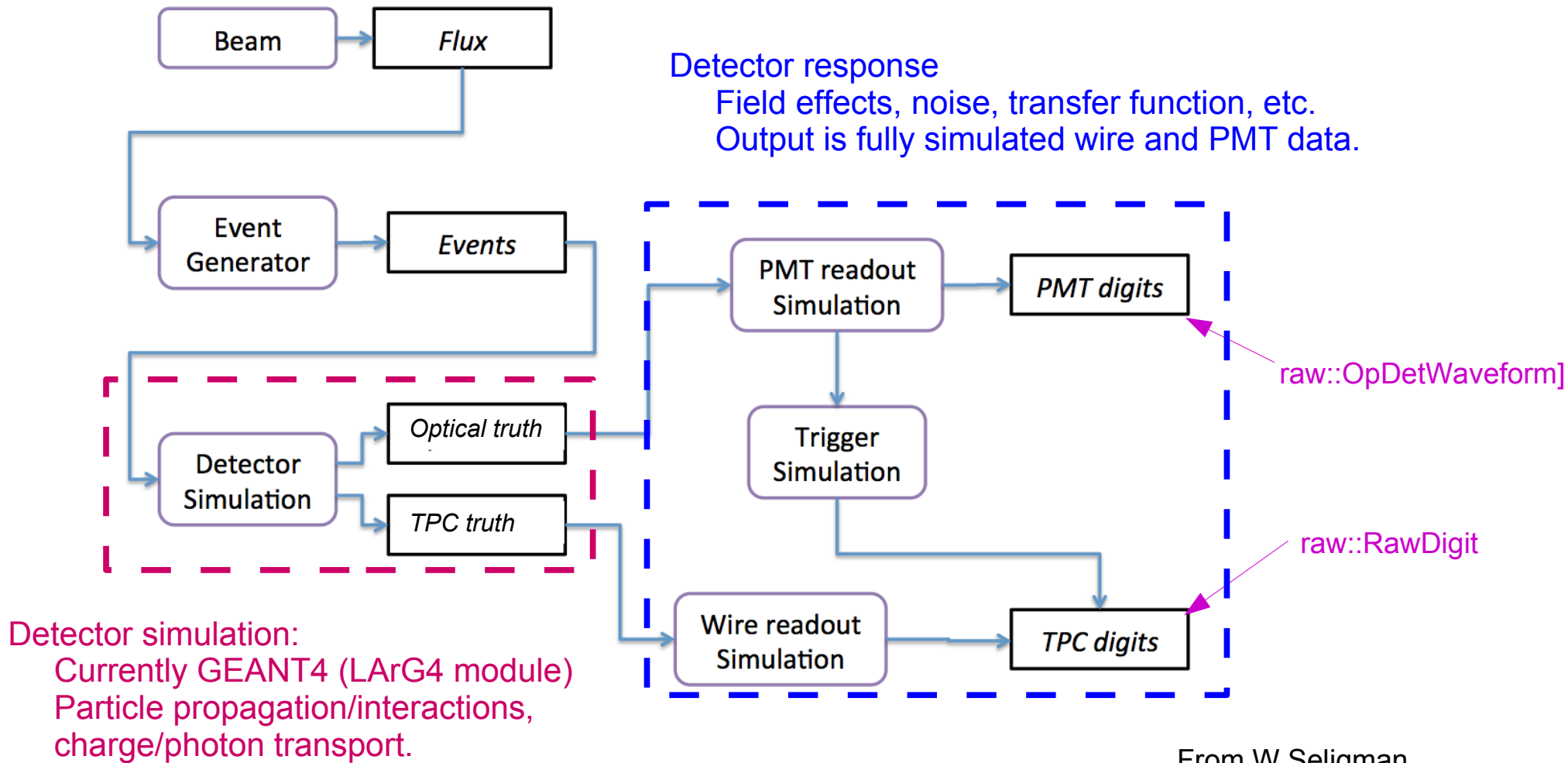
# Simulation workflow

Three phases, typically run separately



# Simulation workflow

The full detector simulation includes two separable sub-phases



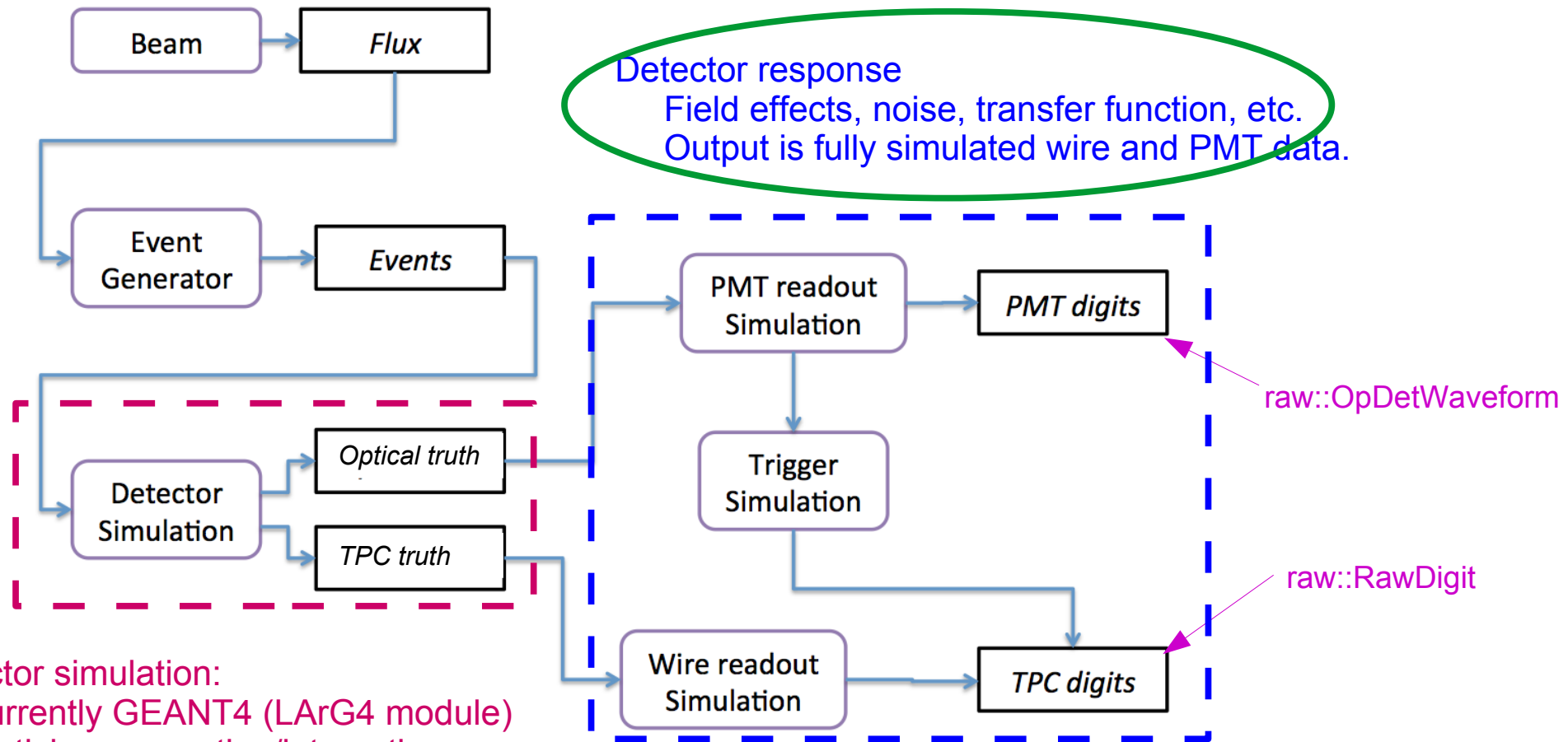
Detector simulation:  
Currently GEANT4 (LArG4 module)  
Particle propagation/interactions,  
charge/photon transport.

Output is MC truth information in `simb::MCParticle`, `sim::SimPhoton`,  
`sim::SimChannel`, `sim::AuxDetSimChannel`,

From W Seligman

# Simulation workflow

Most detect-specific customizations go into the detector response

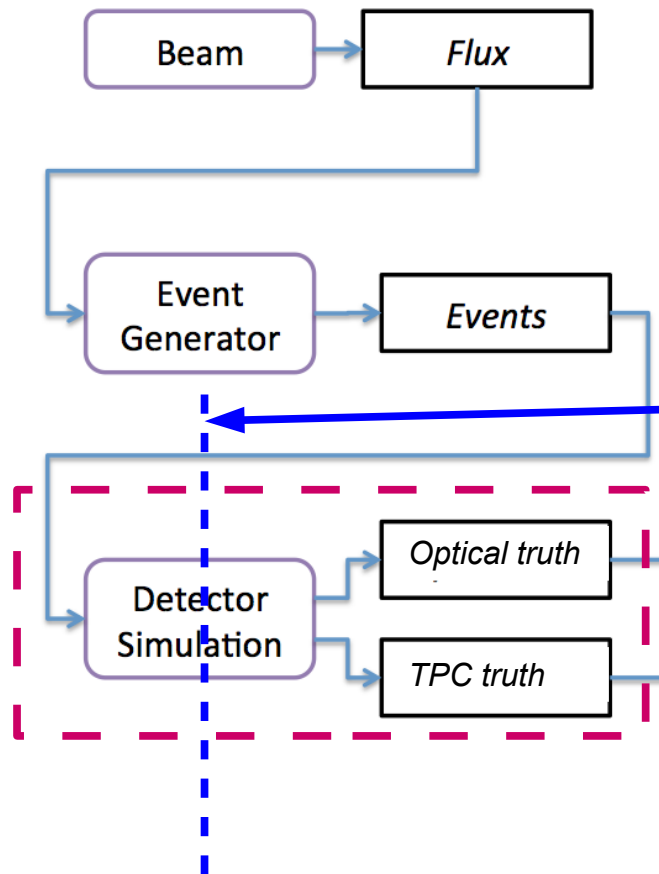


Detector simulation:  
Currently GEANT4 (LArG4 module)  
Particle propagation/interactions,  
charge/photon transport.

Output is MC truth information in `simb::MCParticle`, `sim::SimPhoton`,  
`sim::SimChannel`, `sim::AuxDetSimChannel`,

From W Seligman

# Simulation workflow



- LArG4 re-factoring (in progress)

Separate Geant4 energy deposition from electron and photon transport

Will expose a new interface layer between the two

Simplifies introduction of alternate particle simulation, transport models

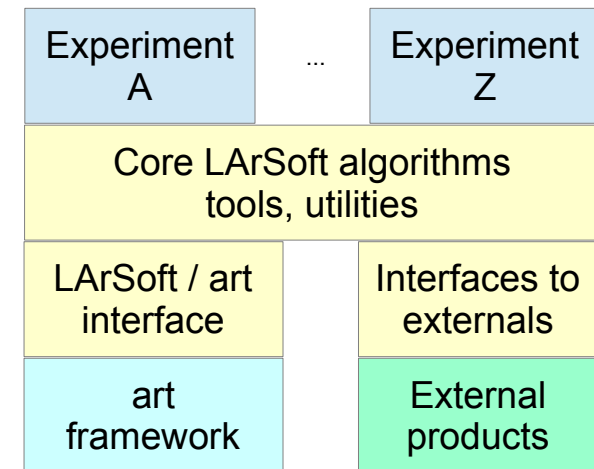
Expect to see this in coming weeks

# LArSoft design principles

# LArSoft design principles and practices



- Detector interoperability
  - The most important design objective for the LArSoft project
  - Define (and use!!) common interfaces for accessing detector-specific configuration information and functionality
  - Applies to geometry, channel mappings, LAr properties, E-field map...
  
- Separation of framework and algorithm code
  - Encapsulate algorithms, configuration, tools and utilities into a layer that is independent of the art framework
  - Many benefits follow from this



**General disclaimer:**

In examining the code, you may note that only a portion currently adheres to these principles.

- An on-going architecture review project is intended to address this

**Strongly encourage people to adopt these practices for all new code**

# LArSoft design principles and practices

- Use of standardized algorithm interfaces
  - Define standard interfaces for well-defined steps in the workflow to promote modularity, layering of algorithms
- Modularity
  - Build sophistication by applying algorithms in a layered, iterative structure
- Design / write testable units of code
  - Include unit and integration testing in the development process
  - Follow the practice of continuous integration
    - Perform automated, broad-scale testing at frequent intervals in order to catch unintended side-effects quickly

# LArSoft design principles and practices

- Continuous integration
  - Automated tests run every time code is pushed to central git repositories
    - Code authors need to provide the relevant tests!!
  - See <http://larsoft.org/continuous-integration>
  
- Document code in the source files as it is written
  - See many files with **no comments at all** describing what the code does
  - At very least, need the have at the top of all header files:
    - the purpose of the file / code
    - pre-requisites and assumptions
    - Anything else people **need to know** in order to use it
  
- Document algorithms and services on <http://larsoft.org/add>
  - High-level description, the principal author, etc.



# Using LArSoft

# Supported platforms

See [https://cdcv.s.fnal.gov/redmine/projects/larsoft/wiki/\\_Supported\\_platforms\\_](https://cdcv.s.fnal.gov/redmine/projects/larsoft/wiki/_Supported_platforms_)

- Scientific Linux
  - SLF6 (the reference system) + SLF7
    - Should work on any SL variant; Works on SLC6 (CERN), Redhat 6 (SLAC)
  - SLF7
    - Binary distributions only - no cvmfs installation
- Mac OSX
  - Yosemite
    - ups qualifiers d13:noifdh and d14:noifdh respectively
  - “Known to work” on El Capitan and Sierra
    - Must disable SIP and install openssl
- Ubuntu
  - “Known to work” with Ubuntu14 and 16
    - LArSoft team distributes installation tarballs for u16 (best effort now, but working to make it “supported”)
    - Ubuntu 14 available only by request

## Installation instructions:

See links in release notes available at  
[https://cdcv.s.fnal.gov/redmine/projects/larsoft/wiki/LArSoft\\_release\\_list](https://cdcv.s.fnal.gov/redmine/projects/larsoft/wiki/LArSoft_release_list)

# LArSoft code repositories

- Code lives in a set of git repositories hosted at Fermilab

larcore	Low level utilities
larcoreobj	Low level data products
larcorealg	Low level utilities
lardata	Data products
lardataobj	Data products
lartoolobj	Low level art tool interfaces (new!)
larsimtool	Low level simulation tool implementations (new!)
lardataalg	Low level algorithms
larevt	Low level algorithms that use data products
larsim	Simulation code
larreco	Primary reconstruction code
larana	Secondary reconstruction and analysis code
lareventdisplay	LArSoft-based event display
larpandora	LArSoft interface to Pandora
larexamples	Placeholder for examples

# LArSoft code repositories

- Code lives in a set of git repositories hosted at Fermilab

larcore                      Low level utilities

larcoreobj                  Low level data products

1) All publicly accessible at <http://cdcvcs.fnal.gov/projects/<repository name>>

2) For read/write access: `ssh://p-<repository name>@cdcvcs.fnal.gov/cvs/projects/<repository name>>`  
(requires valid kerberos ticket)

lardataalg                  Low level algorithms

larevt                      Low level algorithms that use data products

larsim                      Simulation code

larreco                    Primary reconstruction code

larana                     Secondary reconstruction and analysis code

lareventdisplay          LArSoft-based event display

larpandora                LArSoft interface to Pandora

larexamples              Placeholder for examples

# LArSoft products

- The build procedure creates and installs a **ups product** from the code in each repository

larcore	Low level utilities
larcoreobj	Low level data products
larcorealg	Low level utilities
lardata	Data products
lardataobj	Data products
lartoolobj	Low level art tool interfaces (new!)
larsimtool	Low level simulation tool implementations (new!)
lardataalg	Low level algorithms
larevt	Low level algorithms that use data products
larsim	Simulation code
larreco	Primary reconstruction code
larana	Secondary reconstruction and analysis code
lareventdisplay	LArSoft-based event display
larpandora	LArSoft interface to Pandora
larexamples	Placeholder for examples

Each product is self-contained, aside from dependencies

# LArSoft releases

- A **LArSoft release** is a consistent set of LArSoft products built from tagged versions of code in the repositories
  - Implicitly includes corresponding versions of all external dependencies used to build it
  - Each release of LArSoft has a release notes page on [scisoft.fnal.gov](http://scisoft.fnal.gov)
    - <http://scisoft.fnal.gov/scisoft/bundles/larsoft/<version>/larsoft-<version>.html>
- **larsoft**
  - An umbrella product that binds it all together under one version, one setup command
    - `setup larsoft v06_06_00 -q ...`
- **larsoft\_data**
  - A ups product for large configuration files

larsoft v04.16.00	
Product	Version
larcore	v04.13.00
lardata	v04.11.00
larevt	v04.08.06
larsim	v04.08.03
larreco	v04.12.00
larana	v04.08.00
lareventdisplay	v04.06.00
larpandora	v04.04.16
larexamples	v04.04.16
larsoft_data	v0.04.00

...

# LArSoft releases

- Two types of releases
  - Integration
    - Created weekly or on demand for special purposes
    - Contents approved at Coordination Meetings
      - Head of develop + additional branches approved at a CM or via email
    - May be removed without notice after about a month
      - In practice, we announce our intentions in advance
  - Production
    - Any release designated as “production” by an experiment
    - Created on demand (but usually on the weekly schedule)
    - Contents approved by the experiment declaring production
      - Typically also coordinated through the CM to keep other experiments informed
    - Production releases are retained on disk indefinitely
- List of all available tagged releases
  - [https://cdcvs.fnal.gov/redmine/projects/larsoft/wiki/LArSoft\\_release\\_list](https://cdcvs.fnal.gov/redmine/projects/larsoft/wiki/LArSoft_release_list)

# How to set up and run art/LArSoft

- First point to note

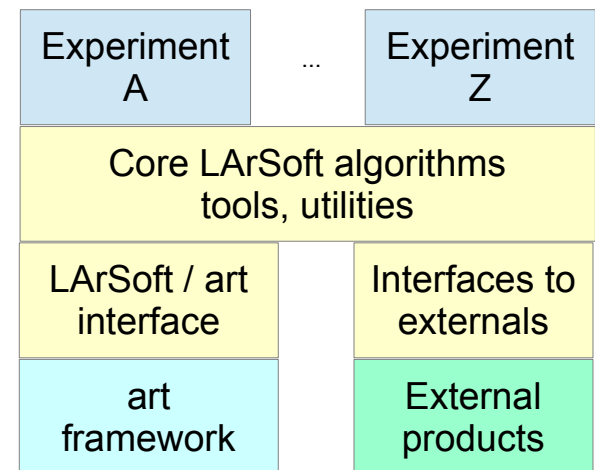
- LArSoft is designed to be run by experiments

- Need detector-specific parts to run it
- So start with the code of your experiment

DUNE	dunetpc
LArIAT	lariatsoft
MicroBooNE	u Boonecode
SBND	sbndcode

The setup procedures for each are different so refer to the relevant setup instructions

- For DUNE, [https://cdcvs.fnal.gov/redmine/projects/dunetpc/wiki/\\_Tutorial\\_](https://cdcvs.fnal.gov/redmine/projects/dunetpc/wiki/_Tutorial_)



- Second point to note

- Do not need a “working area” to run LArSoft. Just need to set up the appropriate products + a fcl file



# How to set up and run art/LArSoft

- The most simple scenario: run from a tagged LArSoft release

- First, set up the working environment:

```
# Set up ups
source <ups location>/setup
#
# Set up the working environment
# for your experiment
<the setup procedure for your experiment here>

# Set up a LArSoft release
setup larsoft v06_06_00 -q e10:<prof|debug>
```

- In most cases, experiment-specific setup scripts will include all of the above
- Note that setting up for development requires additional steps

- Now run art (the LArSoft version is called `lar`)

```
# List art command-line options
lar --help
#
# Now run LArSoft
#
lar -c <some fcl file>.fcl [-i <input file>]
```

That's it! ...if you have a fcl file and have no code to change

# Setting up and running LArSoft as a developer

Use this basic procedure to modify code, including your own art-based analysis code

**Start with the same setup procedure from the previous page**

Note that **mr**rb**** is the primary build tool used by LArSoft

**mr**rb** -help** to list commands  
**mr**rb** <command> --help** for more information

```
# Create working area
mkdir workdir
cd workdir
mrrb newDev -v v06_06_00 -q e10:prof

# Now perform local setup
source localProducts_larsoft_v06_06_00_e10_prof/setup

# Move to source area and check out code
cd srcs
mrrb g <repository name> # or mrrb gitCheckout ...

# Develop code
cd <repository name>/...

# Set up build environment
cd $MRB_BUILDDIR
mrrbsetenv

# Build and install ('install' will actually do both)
mrrb b -j<N> # or mrrb build; N = # cores to use
mrrb I -j <N> # or mrrb install

# Set up newly built code
mrrbslp # 'slp' = setup local products

# Run as before
lar -c ...
```

# Getting LArSoft to work with a new detector

- LArSoft is really a toolkit
  - It requires a certain amount of detector-specific information and plug-in functionality in order to work
- The minimum needed to run LArSoft for a new detector
  - Define the geometry for the new detector in a GDML file
  - Customize E-field, drift velocity, readout parameters, etc, as needed
  - Customize digitization for simulation, as needed
  - Write a fcl file
  - `lar -c sim_new_det.fcl` → simulated data; `lar -c <reco...>.fcl` → results!!

...A bit over-simplified, but this is basically what happens

# Resources

larsoft.org



The public facing home page for the LArSoft Collaboration

Contains introductory information + links to further information and resources

Tabs across the top link to different types of content

## LArSoft

The Liquid Argon Software (LArSoft) Collaboration develops and supports a shared base of physics software across Liquid Argon (LAr) Time Projection Chamber (TPC) experiments.

June 22-23 Usability Workshop report is available [here](#).

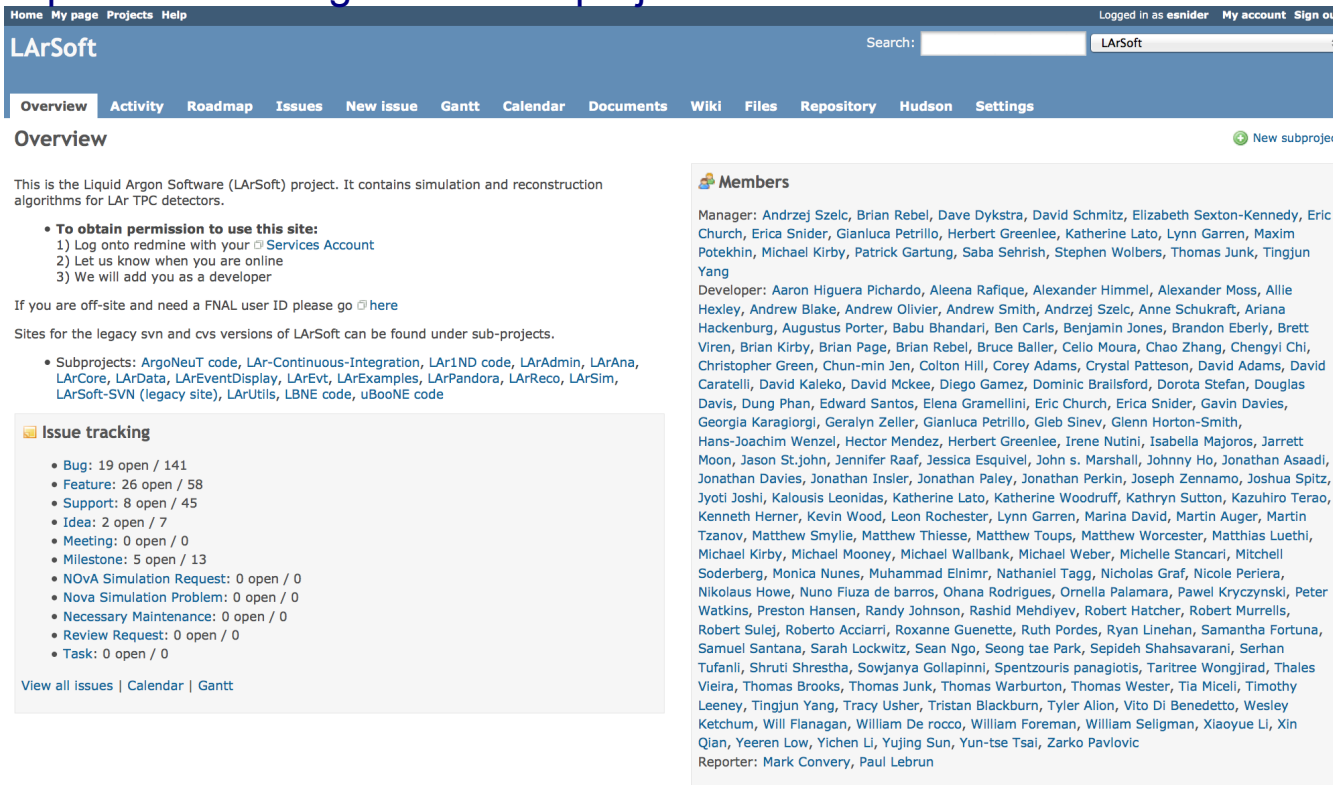
More information about LArSoft is at:

- [LArSoft Article](#) – introduction for general public
- [What is LArSoft](#) – explains collaboration versus software aspects of LArSoft
- [Concepts in LArSoft](#) – big picture understanding of LArSoft code
- [LArSoft wiki](#) – covers everything from introduction to details on code releases
- [LArSoft Issues \(on the above wiki\)](#) – tracks work areas, support questions, etc.
- [Coordination meetings](#) – decisions about the content of releases
- [Steering group meetings](#) – policy decisions for the collaboration
- [larforum.org](#) – forum hosted by the University of Manchester to discuss LArTPC software
- [CI results display page](#) – continuous integration status page (it takes time to load)
- [LArTPC\\_Software\\_Glossary](#) – as a PDF file
- [Introduction to LArSoft from training class August 2015](#)

# LArSoft Redmine site

Redmine sites are called “projects”

<https://cdcvns.fnal.gov/redmine/projects/larsoft>



The screenshot shows the Redmine interface for the LArSoft project. At the top, there is a navigation bar with links for Home, My page, Projects, and Help. A search bar is present with the text 'LArSoft' entered. Below the navigation bar, there are tabs for Overview, Activity, Roadmap, Issues, New issue, Gantt, Calendar, Documents, Wiki, Files, Repository, Hudson, and Settings. The 'Overview' tab is selected.

**Overview**

This is the Liquid Argon Software (LArSoft) project. It contains simulation and reconstruction algorithms for LAr TPC detectors.

- To obtain permission to use this site:**
  - 1) Log onto redmine with your [Services Account](#)
  - 2) Let us know when you are online
  - 3) We will add you as a developer

If you are off-site and need a FNAL user ID please go [here](#)

Sites for the legacy svn and cvs versions of LArSoft can be found under sub-projects.

- Subprojects: ArgoNeUT code, LAr-Continuous-Integration, LAr1ND code, LArAdmin, LArAna, LArCore, LArData, LArEventDisplay, LArEvt, LArExamples, LArPandora, LArReco, LArSim, LArSoft-SVN (legacy site), LArUtils, LBNE code, uBooNE code

**Issue tracking**

- Bug: 19 open / 141
- Feature: 26 open / 58
- Support: 8 open / 45
- Idea: 2 open / 7
- Meeting: 0 open / 0
- Milestone: 5 open / 13
- NOvA Simulation Request: 0 open / 0
- Nova Simulation Problem: 0 open / 0
- Necessary Maintenance: 0 open / 0
- Review Request: 0 open / 0
- Task: 0 open / 0

[View all issues](#) | [Calendar](#) | [Gantt](#)

**Members**

Manager: Andrzej Szelc, Brian Rebel, Dave Dykstra, David Schmitz, Elizabeth Sexton-Kennedy, Eric Church, Erica Snider, Gianluca Petrillo, Herbert Greenlee, Katherine Lato, Lynn Garren, Maxim Potekhin, Michael Kirby, Patrick Gartung, Saba Sehrish, Stephen Wolbers, Thomas Junk, Tingjun Yang

Developer: Aaron Higuera Pichardo, Aleena Rafique, Alexander Himmel, Alexander Moss, Allie Hexley, Andrew Blake, Andrew Olivier, Andrew Smith, Andrzej Szelc, Anne Schukraft, Ariana Hackenburg, Augustus Porter, Babu Bhandari, Ben Carls, Benjamin Jones, Brandon Eberly, Brett Viren, Brian Kirby, Brian Page, Brian Rebel, Bruce Baller, Celio Moura, Chao Zhang, Chengyi Chi, Christopher Green, Chun-min Jen, Colton Hill, Corey Adams, Crystal Patteson, David Adams, David Caratelli, David Kaleko, David Mckee, Diego Gamez, Dominic Brallsford, Dorota Stefan, Douglas Davis, Dung Phan, Edward Santos, Elena Gramellini, Eric Church, Erica Snider, Gavin Davies, Georgia Karagiorgi, Geralyn Zeller, Gianluca Petrillo, Gleb Sinev, Glenn Horton-Smith, Hans-Joachim Wenzel, Hector Mendez, Herbert Greenlee, Irene Nutini, Isabella Majoros, Jarrett Moon, Jason St.John, Jennifer Raaf, Jessica Esquivel, John s. Marshall, Johnny Ho, Jonathan Asaadi, Jonathan Davies, Jonathan Insler, Jonathan Paley, Jonathan Perkin, Joseph Zennamo, Joshua Spitz, Jyoti Joshi, Kalousis Leonidas, Katherine Lato, Katherine Woodruff, Kathryn Sutton, Kazuhiro Terao, Kenneth Herner, Kevin Wood, Leon Rochester, Lynn Garren, Marina David, Martin Auger, Martin Tzanov, Matthew Smylie, Matthew Thiesse, Matthew Toups, Matthew Worcester, Matthias Luethi, Michael Kirby, Michael Mooney, Michael Wallbank, Michael Weber, Michelle Stancari, Mitchell Soderberg, Monica Nunes, Muhammad Elnimr, Nathaniel Tagg, Nicholas Graf, Nicole Periera, Nikolaus Howe, Nuno Fiuzza de Barros, Ohana Rodrigues, Ornella Palamara, Pawel Kryczynski, Peter Watkins, Preston Hansen, Randy Johnson, Rashid Mehdiyev, Robert Hatcher, Robert Murrells, Robert Sulej, Roberto Acciarri, Roxanne Guenette, Ruth Pordes, Ryan Linehan, Samantha Fortuna, Samuel Santana, Sarah Lockwitz, Sean Ngo, Seong tae Park, Sepideh Shahsavarani, Serhan Tufanli, Shruti Shrestha, Sowjanya Gollapinni, Spentzouris panagiotis, Taritree Wongjirad, Thales Vieira, Thomas Brooks, Thomas Junk, Thomas Warburton, Thomas Wester, Tia Miceli, Timothy Leeney, Tingjun Yang, Tracy Usher, Tristan Blackburn, Tyler Alion, Vito Di Benedetto, Wesley Ketchum, Will Flanagan, William De rocco, William Foreman, William Seligman, Xiaoyue Li, Xin Qian, Yeeren Low, Yichen Li, Yujing Sun, Yun-tse Tsai, Zarko Pavlovic

Reporter: Mark Convery, Paul Lebrun

This is the home page for the LArSoft Redmine project

Tabs across the top link to different types of content

# LArSoft Redmine site

Redmine sites are called “projects”

<https://cdcvns.fnal.gov/redmine/projects/larsoft>

The screenshot shows the LArSoft Redmine project home page. The navigation bar at the top includes tabs for Overview, Activity, Roadmap, Issues, New issue, Gantt, Calendar, Documents, Wiki, Files, Repository, Hudson, and Settings. The 'Issues' and 'New issue' tabs are circled in red. Below the navigation bar, there is an 'Overview' section with project information, a 'Members' list, and an 'Issue tracking' section with a list of open issues.

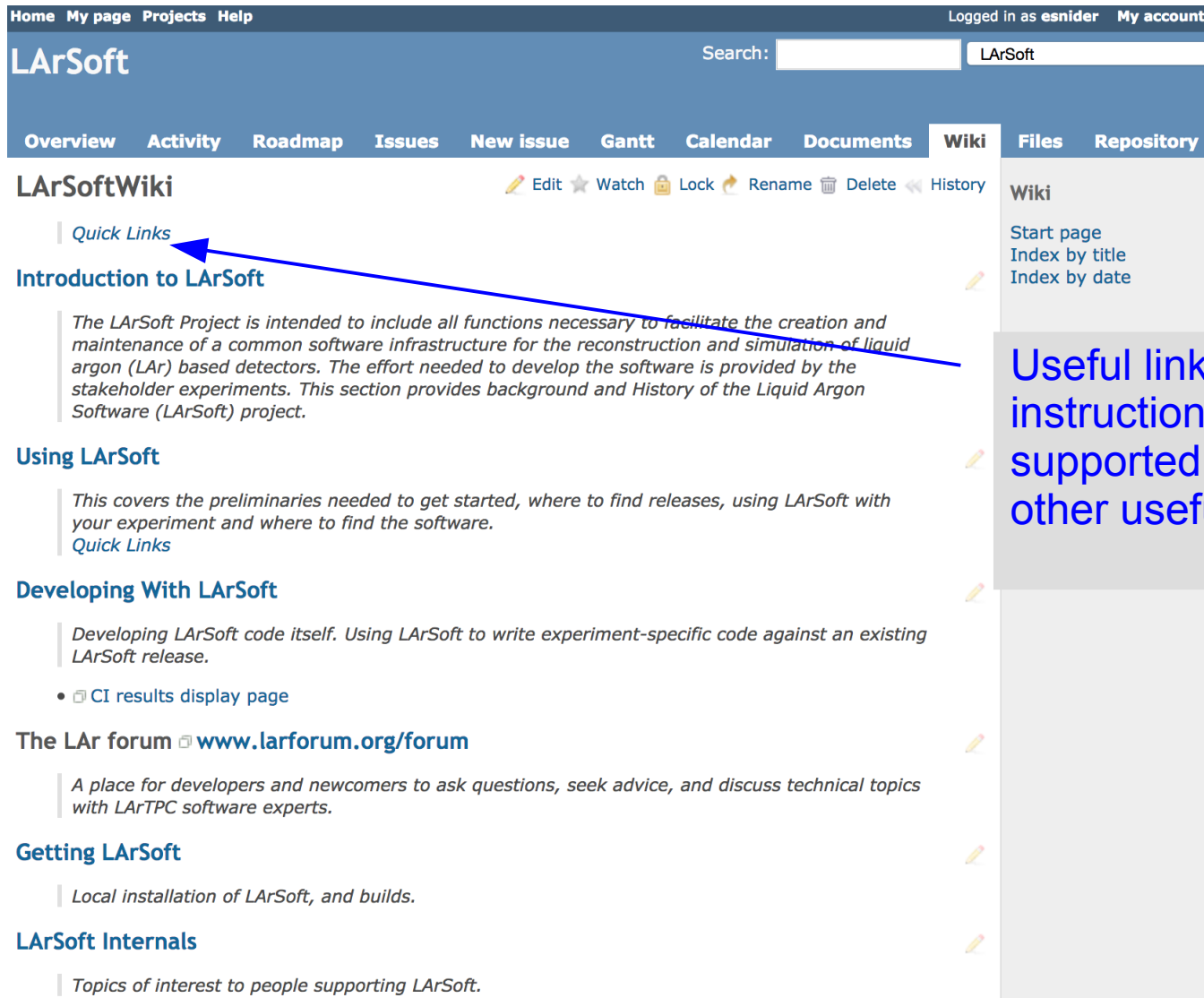
This is the home page for the LArSoft Redmine project

Tabs across the top link to different types of content

The most useful tabs:  
 Wiki  
 Issues  
 New Issues  
 Repository

# LArSoft wiki

<https://cdcvns.fnal.gov/redmine/projects/larsoft/wiki>



Home My page Projects Help Logged in as esnider My account

LArSoft Search: LArSoft

Overview Activity Roadmap Issues New issue Gantt Calendar Documents **Wiki** Files Repository

**LArSoftWiki** Edit Watch Lock Rename Delete History

Quick Links

**Introduction to LArSoft**

The LArSoft Project is intended to include all functions necessary to facilitate the creation and maintenance of a common software infrastructure for the reconstruction and simulation of liquid argon (LAr) based detectors. The effort needed to develop the software is provided by the stakeholder experiments. This section provides background and History of the Liquid Argon Software (LArSoft) project.

**Using LArSoft**

This covers the preliminaries needed to get started, where to find releases, using LArSoft with your experiment and where to find the software.

Quick Links

**Developing With LArSoft**

Developing LArSoft code itself. Using LArSoft to write experiment-specific code against an existing LArSoft release.

- CI results display page

**The LAr forum** [www.larforum.org/forum](http://www.larforum.org/forum)

A place for developers and newcomers to ask questions, seek advice, and discuss technical topics with LArTPC software experts.

**Getting LArSoft**

Local installation of LArSoft, and builds.

**LArSoft Internals**

Topics of interest to people supporting LArSoft.

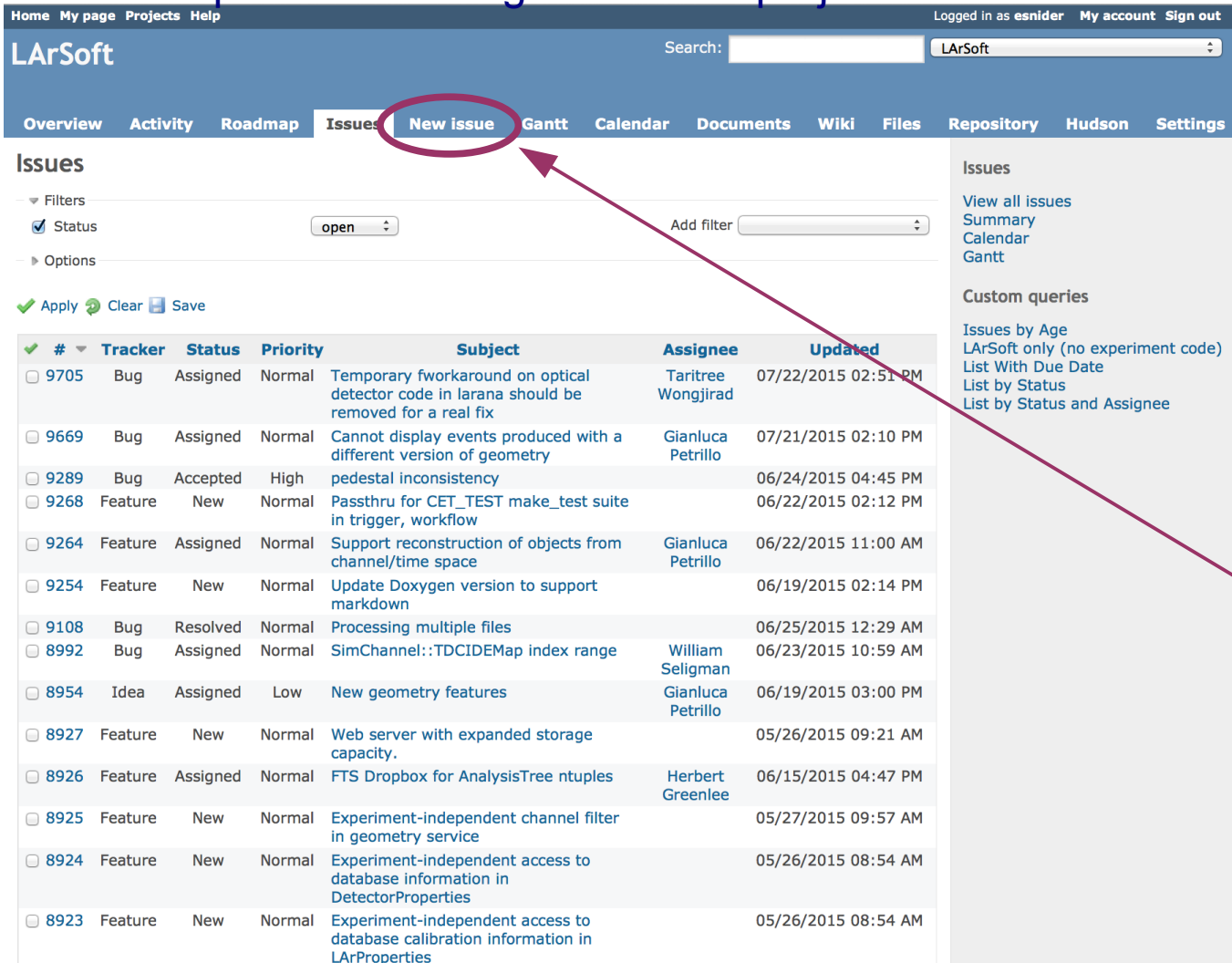
General information and documentation

Useful links to getting started instructions, list of releases, supported platforms and other useful information.



# LArSoft issue tracker

<https://cdcvs.fnal.gov/redmine/projects/larsoft/issues>



Home My page Projects Help Logged in as esnider My account Sign out

LArSoft Search: LArSoft

Overview Activity Roadmap **Issues** **New Issue** Gantt Calendar Documents Wiki Files Repository Hudson Settings

Issues

Filters

Status open Add filter open

Options

Apply Clear Save

#	Tracker	Status	Priority	Subject	Assignee	Updated
<input type="checkbox"/> 9705	Bug	Assigned	Normal	Temporary fworkaround on optical detector code in larana should be removed for a real fix	Taritree Wongjirad	07/22/2015 02:51 PM
<input type="checkbox"/> 9669	Bug	Assigned	Normal	Cannot display events produced with a different version of geometry	Gianluca Petrillo	07/21/2015 02:10 PM
<input type="checkbox"/> 9289	Bug	Accepted	High	pedestal inconsistency		06/24/2015 04:45 PM
<input type="checkbox"/> 9268	Feature	New	Normal	Pas thru for CET_TEST make_test suite in trigger, workflow		06/22/2015 02:12 PM
<input type="checkbox"/> 9264	Feature	Assigned	Normal	Support reconstruction of objects from channel/time space	Gianluca Petrillo	06/22/2015 11:00 AM
<input type="checkbox"/> 9254	Feature	New	Normal	Update Doxygen version to support markdown		06/19/2015 02:14 PM
<input type="checkbox"/> 9108	Bug	Resolved	Normal	Processing multiple files		06/25/2015 12:29 AM
<input type="checkbox"/> 8992	Bug	Assigned	Normal	SimChannel::TDCIDEMap index range	William Seligman	06/23/2015 10:59 AM
<input type="checkbox"/> 8954	Idea	Assigned	Low	New geometry features	Gianluca Petrillo	06/19/2015 03:00 PM
<input type="checkbox"/> 8927	Feature	New	Normal	Web server with expanded storage capacity.		05/26/2015 09:21 AM
<input type="checkbox"/> 8926	Feature	Assigned	Normal	FTS Dropbox for AnalysisTree ntuples	Herbert Greenlee	06/15/2015 04:47 PM
<input type="checkbox"/> 8925	Feature	New	Normal	Experiment-independent channel filter in geometry service		05/27/2015 09:57 AM
<input type="checkbox"/> 8924	Feature	New	Normal	Experiment-independent access to database information in DetectorProperties		05/26/2015 08:54 AM
<input type="checkbox"/> 8923	Feature	New	Normal	Experiment-independent access to database calibration information in LArProperties		05/26/2015 08:54 AM

Issues

View all issues  
Summary  
Calendar  
Gantt

Custom queries

Issues by Age  
LArSoft only (no experiment code)  
List With Due Date  
List by Status  
List by Status and Assignee

We track bugs, problems with LArSoft-related infrastructure, requests for support and new features, questions...

Open a new ticket using the “New Issue” tab if you have any of the above

Create a new issue using this tab.

Must be logged into Redmine using your Fermilab **services account** and password

# LArSoft Redmine code browser

<https://cdcv.s.fnal.gov/redmine/projects/larreco/repository>

Name	Size	Revision	Age	Author	Comment
ClusterFinder		f5f82401	3 days	Michael Wallbank	Made use of cluster merging after blurred clust...
EventFinder		8a55c60c	9 months	Gianluca Petrillo	Fixed includes of FindOne/FindMany headers
Genfit		c427eb6a	about 1 month	Lynn Garren	need Boost_SYSTEM_LIBRARY
HitFinder		a936a414	about 1 month	Michelle Stancari	bug fix
MCComp		9ac3d074	about 1 month	Lynn Garren	need Boost_SYSTEM_LIBRARY
RecoAlg		603787f3	about 16 hours	Tingjun Yang	Temporary fix so job won't crash. Need more inv...
ShowerFinder		368e8553	4 months	Kazuhiro Terao	updating fcl name to be consistent --kazu
SpacePointFinder		d4458f63	over 1 year	Lynn Garren	remove lines that were commented out
TrackFinder		4dc46e2c	about 18 hours	Robert Sulej	fix producer to put also empty containers to th...
VertexFinder		b9ebc2d4	2 months	Tingjun Yang	Change length to be float.
test		e6d08128	4 months	Gianluca Petrillo	Enabled use of cached multi-Gaussian functions ...
ups		032ed77e	7 days	Lynn Garren	larreco v04_12_00 for larsoft v04_16_00
.gitignore	3 Bytes	58593c9d	6 months	Lynn Garren	try to protect against accidental additions
CMakeLists.txt	2 KB	280f2fa	4 months	Kazuhiro Terao	Putting back MCComp --kazu

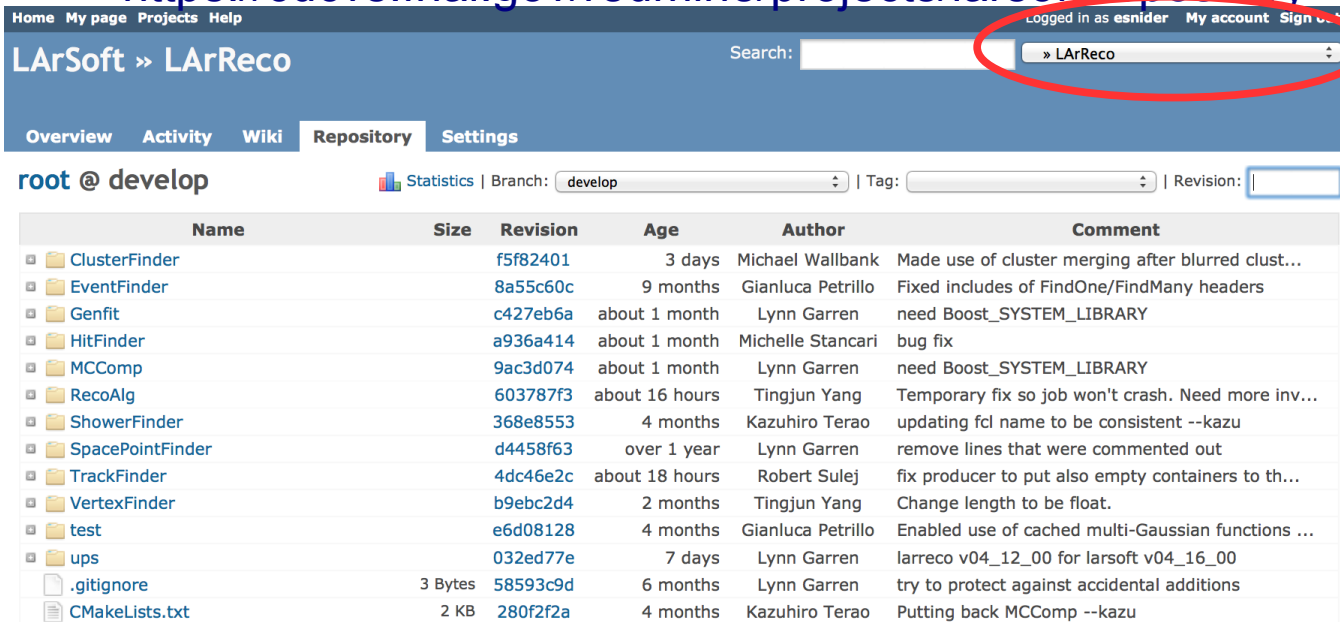
Each LArSoft repository lives in a separate Redmine project which is specified here.

## Latest revisions

#	Date	Author	Comment
603787f3	07/22/2015 04:06 PM	Tingjun Yang	Temporary fix so job won't crash. Need more investigation on why this happened.
4dc46e2c	07/22/2015 02:18 PM	Robert Sulej	fix producer to put also empty containers to the event, add one more TTree with info to inspect tracks
77c4e4cd	07/22/2015 02:12 PM	Robert Sulej	add fn to returns mean angle between consecutive segments
7b1b1fcf	07/21/2015 05:03 PM	Robert Sulej	add optio to flip tracks downward
b2e460f0	07/20/2015 04:53 PM	Tingjun Yang	Add protection when TrackTrajectoryAlg does not reconstruct trajectory points.
b2e905c2	07/20/2015 04:52 PM	Tingjun Yang	Merge branch 'develop' of ssh://cdcv.s.fnal.gov/cvs/projects/larreco into develop
33890c78	07/20/2015 04:52 PM	Tingjun Yang	Add protection for the case bin is negative. This only happens when divided by a very small number.
85a54d56	07/20/2015 04:27 PM	Bruce Baller	Merge branch 'feature/bb_ccwork' into develop
02a39c7c	07/20/2015 04:25 PM	Bruce Baller	initialize matcomb
8342da80	07/20/2015 04:18 PM	Robert Sulej	add projection of 3D vector to 2D plane in [cm] domain

# Navigating between LArSoft sub-projects

<https://cdcvs.fnal.gov/redmine/projects/larsoft/repository>



Home My page Projects Help Logged in as esnider My account Sign out

LArSoft » LArReco Search:

Overview Activity Wiki **Repository** Settings

root @ develop Statistics | Branch:  | Tag:  | Revision:

Name	Size	Revision	Age	Author	Comment
ClusterFinder		f5f82401	3 days	Michael Wallbank	Made use of cluster merging after blurred clust...
EventFinder		8a55c60c	9 months	Gianluca Petrillo	Fixed includes of FindOne/FindMany headers
Genfit		c427eb6a	about 1 month	Lynn Garren	need Boost_SYSTEM_LIBRARY
HitFinder		a936a414	about 1 month	Michelle Stancari	bug fix
MCComp		9ac3d074	about 1 month	Lynn Garren	need Boost_SYSTEM_LIBRARY
RecoAlg		603787f3	about 16 hours	Tingjun Yang	Temporary fix so job won't crash. Need more inv...
ShowerFinder		368e8553	4 months	Kazuhiro Terao	updating fcl name to be consistent --kazu
SpacePointFinder		d4458f63	over 1 year	Lynn Garren	remove lines that were commented out
TrackFinder		4dc46e2c	about 18 hours	Robert Sulej	fix producer to put also empty containers to th...
VertexFinder		b9ebc2d4	2 months	Tingjun Yang	Change length to be float.
test		e6d08128	4 months	Gianluca Petrillo	Enabled use of cached multi-Gaussian functions ...
ups		032ed77e	7 days	Lynn Garren	larreco v04_12_00 for larsoft v04_16_00
.gitignore	3 Bytes	58593c9d	6 months	Lynn Garren	try to protect against accidental additions
CMakeLists.txt	2 KB	280f2f2a	4 months	Kazuhiro Terao	Putting back MCComp --kazu

Can use the project navigation pull-down to get to the desired project.

## Latest revisions

#	Date	Author	Comment
603787f3	07/22/2015 04:06 PM	Tingjun Yang	Temporary fix so job won't crash. Need more investigation on why this happened.
4dc46e2c	07/22/2015 02:18 PM	Robert Sulej	fix producer to put also empty containers to the event, add one more TTree with info to inspect tracks
77c4e4cd	07/22/2015 02:12 PM	Robert Sulej	add fn to returns mean angle between consecutive segments
7b1b1fcf	07/21/2015 05:03 PM	Robert Sulej	add optio to flip tracks downward
b2e460f0	07/20/2015 04:53 PM	Tingjun Yang	Add protection when TrackTrajectoryAlg does not reconstruct trajectory points.
b2e905c2	07/20/2015 04:52 PM	Tingjun Yang	Merge branch 'develop' of ssh://cdcvs.fnal.gov/cvs/projects/larreco into develop
33890c78	07/20/2015 04:52 PM	Tingjun Yang	Add protection for the case bin is negative. This only happens when divided by a very small number.
85a54d56	07/20/2015 04:27 PM	Bruce Baller	Merge branch 'feature/bb_ccwork' into develop
02a39c7c	07/20/2015 04:25 PM	Bruce Baller	initialize matcomb
8342da80	07/20/2015 04:18 PM	Robert Sulej	add projection of 3D vector to 2D plane in [cm] domain

# Navigating between LArSoft sub-projects

Home My page Projects Help

LArSoft » LArReco

Search:

Logged in as esnider My account Sign out

Overview Activity Wiki **Repository** Settings

root @ develop Statistics | Branch:  | Tag:

Name	Size	Revision	Age	Author	Comment
ClusterFinder		f5f82401	3 days	Michael Wallbank	Made use of clu
EventFinder		8a55c60c	9 months	Gianluca Petrillo	Fixed includes
Genfit		c427eb6a	about 1 month	Lynn Garren	need Boost_SY
HitFinder		a936a414	about 1 month	Michelle Stancari	bug fix
MCComp		9ac3d074	about 1 month	Lynn Garren	need Boost_SY
RecoAlg		603787f3	about 16 hours	Tingjun Yang	Temporary fix s
ShowerFinder		368e8553	4 months	Kazuhiro Terao	updating fcl na
SpacePointFinder		d4458f63	over 1 year	Lynn Garren	remove lines that were commented out
TrackFinder		4dc46e2c	about 18 hours	Robert Sulej	fix producer to put also empty containers to th...
VertexFinder		b9ebc2d4	2 months	Tingjun Yang	Change length to be float.
test		e6d08128	4 months	Gianluca Petrillo	Enabled use of cached multi-Gaussian functions ...
ups		032ed77e	7 days	Lynn Garren	larreco v04_12_00 for larsoft v04_16_00
.gitignore	3 Bytes	58593c9d	6 months	Lynn Garren	try to protect against accidental additions
CMakeLists.txt	2 KB	280f2f2a	4 months	Kazuhiro Terao	Putting back MCComp --kazu

Navigation pull-down menu (circled in red):

- LArReco
- Build Service
- art-workbook
- artExtensions
- mrb
- FIFE
- art/LArSoft Course
- IFront
- LArSoft
- » ArgoNeuT code
- » LAr-Continuous-Integration
- » LAr1ND code
- » LArAdmin
- » LArAna
- » LArCore
- » LArData
- » LArEventDisplay
- » LArEvt
- » LArExamples
- » LArPandora
- » LArReco

Can use the project navigation pull-down to get to the desired project.

## Latest revisions

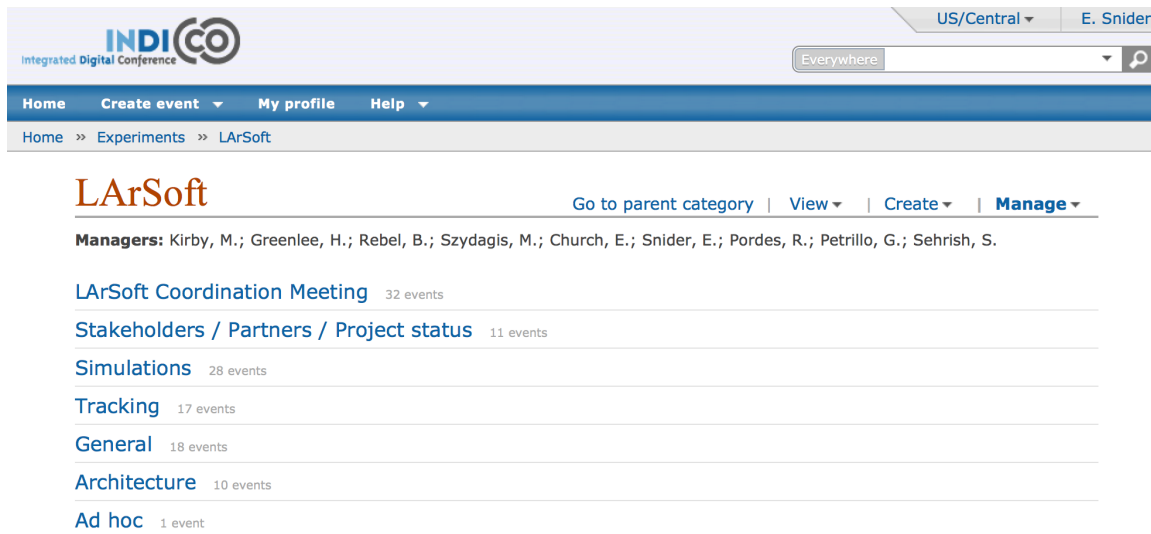
#	Date	Author	Comment
603787f3	07/22/2015 04:06 PM	Tingjun Yang	Temporary fix so job won't crash. Need more investigation on why this happened.
4dc46e2c	07/22/2015 02:18 PM	Robert Sulej	fix producer to put also empty containers to the event, add one more TTree with info to inspect tracks
77c4e4cd	07/22/2015 02:12 PM	Robert Sulej	add fn to returns mean angle between consecutive segments
7b1b1fcf	07/21/2015 05:03 PM	Robert Sulej	add optio to flip tracks downward

# LArSoft Indico site

- Slides and documents from meetings are posted to Indico

<https://indico.fnal.gov/categoryDisplay.py?categId=233>

- Or from the Indico home page: <https://indico.fnal.gov/index.py>, follow “Experiments”, then “LArSoft” links to arrive at the LArSoft page



The screenshot shows the Indico website interface. At the top, there is a navigation bar with the Indico logo and the text 'Integrated Digital Conference'. To the right of the logo, there are dropdown menus for 'US/Central' and 'E. Snider'. Below the navigation bar, there is a search bar with the text 'Everywhere' and a magnifying glass icon. The main content area displays the 'LArSoft' category page. The page title is 'LArSoft' in a large, bold font. Below the title, there are links for 'Go to parent category', 'View', 'Create', and 'Manage'. A list of managers is provided: Kirby, M.; Greenlee, H.; Rebel, B.; Szydagis, M.; Church, E.; Snider, E.; Pordes, R.; Petrillo, G.; Sehrish, S. Below the managers list, there is a list of sub-categories with their respective event counts: 'LArSoft Coordination Meeting' (32 events), 'Stakeholders / Partners / Project status' (11 events), 'Simulations' (28 events), 'Tracking' (17 events), 'General' (18 events), 'Architecture' (10 events), and 'Ad hoc' (1 event).

Can upload slides on the page for the particular meeting...

...but, must be logged in using your **Indico account** and password

# Resources

- LArSoft web page
  - <https://larsoft.org>
- LArSoft dOxygen documentation system:
  - <https://nusoft.fnal.gov/larsoft/doxsvn/html/index.html>
- LArSoft email list: [larsoft@fnal.gov](mailto:larsoft@fnal.gov)
  - General announcements. Some technical questions too.
  - Can self-subscribe. See <http://listserv.fnal.gov/> for instructions.
- **LArSoft Coordination Meeting**
  - Bi-weekly at 09:00 Central Time in WH3NE
  - Remote connections via Zoom. Slides, notes posted to [LArSoft Indico site](#).
- LArSoft wiki: <https://cdcvs.fnal.gov/redmine/projects/larsoft/wiki>
  - Quick page with links to quick-start guides by experiment
- LArSoft issue tracker
  - <https://cdcvs.fnal.gov/redmine/projects/larsoft/issues/new>
- 2015 LArSoft course material
  - <https://indico.fnal.gov/conferenceTimeTable.py?confId=9928#20150807>

# Core LArSoft support team

- Core team members

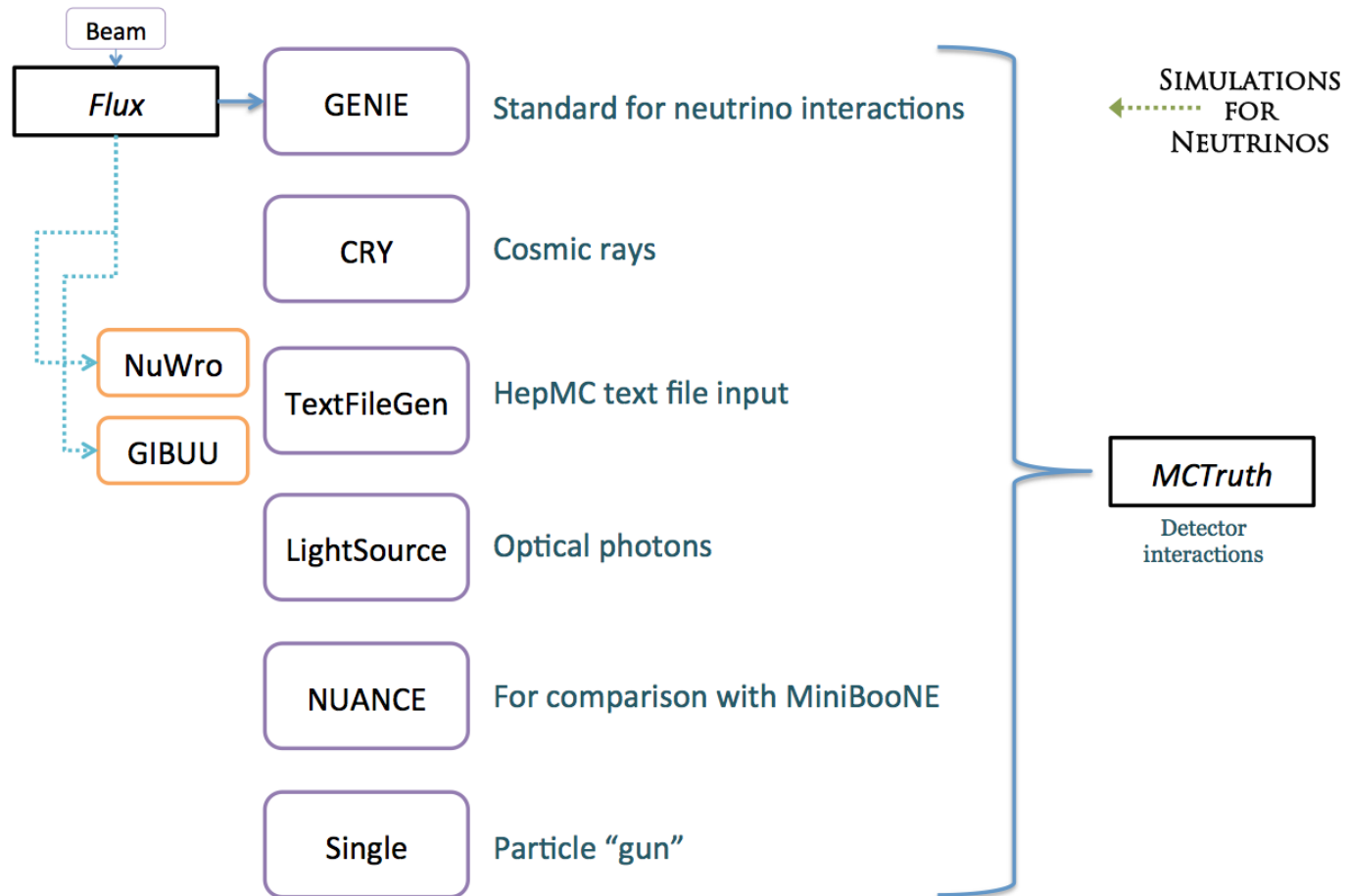
- Technical lead: Erica Snider  
[erica@fnal.gov](mailto:erica@fnal.gov)
- Project manager: Katherine Lato  
[klato@fnal.gov](mailto:klato@fnal.gov)
- Lead developer: Gianluca Petrillo  
[petrillo@fnal.gov](mailto:petrillo@fnal.gov)
- Developers: Giuseppe Cerati  
[cerati@fnal.gov](mailto:cerati@fnal.gov)  
Saba Sehrish  
[ssehrish@fnal.gov](mailto:ssehrish@fnal.gov)
- Code management and distribution: Lynn Garren  
[garren@fnal.gov](mailto:garren@fnal.gov)
- CI operations and testing support: Vito di Benedetto  
[vito@fnal.gov](mailto:vito@fnal.gov)
- Documentation: Katherine Lato

The end



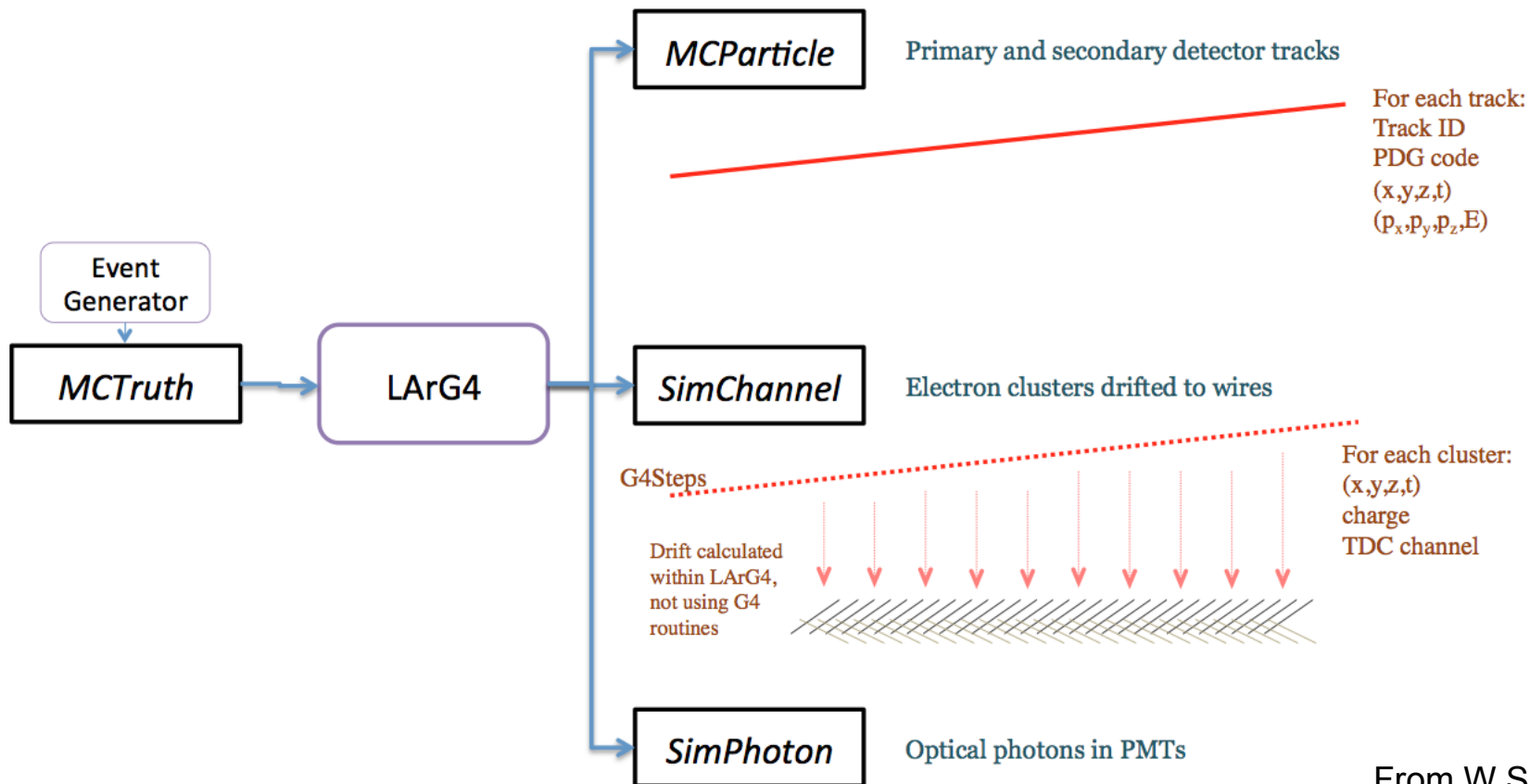
# Backup

# Event generators



From W Seligman

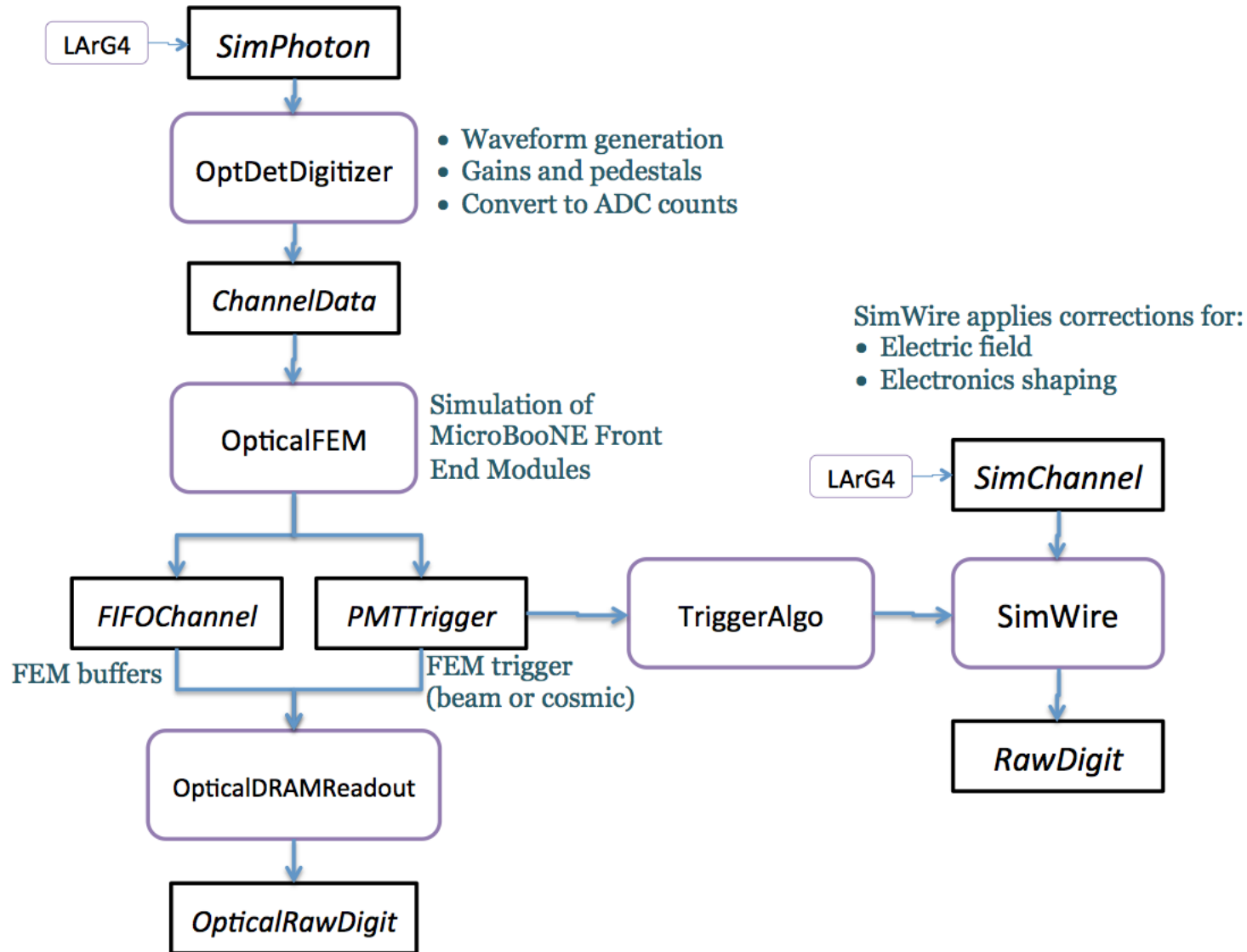
# Detector simulation



From W Seligman

# Simulation task workflow

## Detector response and digitization



# LArSoft design principles and objectives

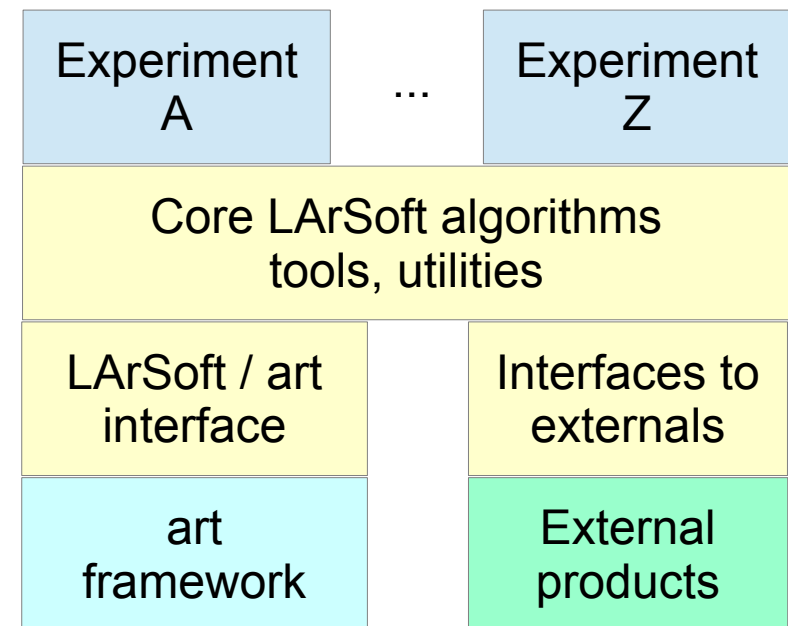


- Detector interoperability

- The most important design objective for the LArSoft project
- Requires care to define (and use!!) common interfaces for accessing detector-specific configuration information
- Good example: access to detector geometry information
  - A single interface that accommodates different (albeit very similar) geometries
    - Most differences a matter of configuration only
    - Also have detector-specific implementations of the interface where needed
  - Carefully avoid implied geometrical assumptions in algorithms
    - Position of the first plane or wire, the wire spacing, etc.
  - Introduced structures to facilitate generic loops over geometrical elements
    - Define detector / DAQ element IDs at all levels of detector geometry hierarchy
    - Can thereby avoid explicit reference to indices for loops, etc.
- Also applies to
  - Access to calibration data, LAr properties, detector properties, E-field map, handling of common metadata for data files...
- Have compiled a long list of do's and don'ts to ensure interoperability

# LArSoft design principles and objectives

- Separation of framework and algorithm code
    - Encapsulate algorithms, configuration, tools and utilities into a layer that is independent of the framework
    - Why??
      - Allow testing of small units of algorithm code outside the framework
      - Provide greater flexibility in using algorithms
      - To provide a means of integrating LArSoft code (data products and algorithms) with external frameworks
        - e.g., LArLite used by MicroBooNE for algorithm development, testing
- See [art guidance for writing modules](#) for further discussion



# LArSoft design principles and objectives

- Separation of framework and algorithm code

- Encapsulate algorithms, configuration, tools and utilities into a layer that is independent of the framework

- Why??

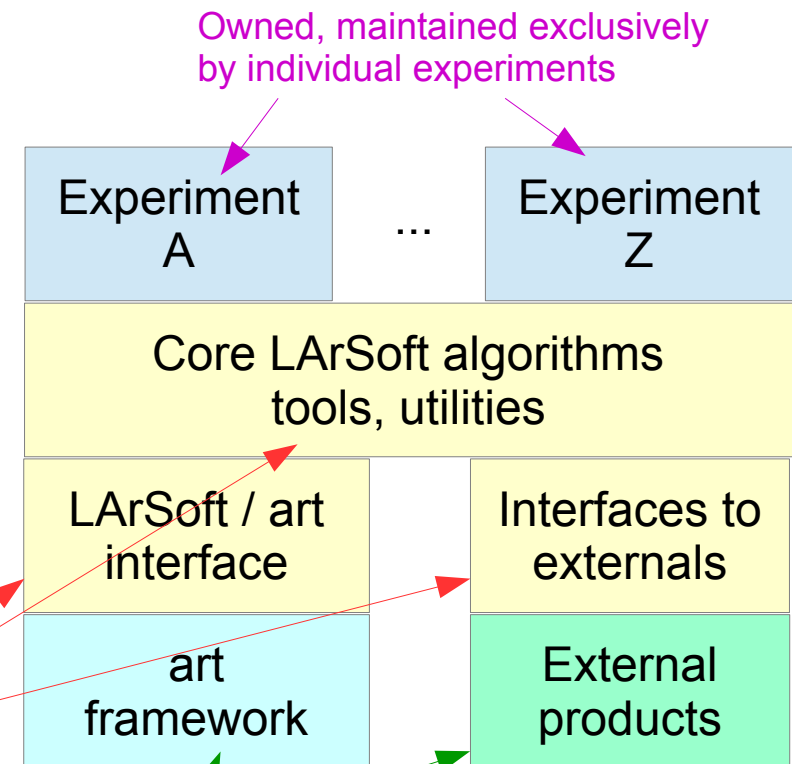
- Allow testing of small units of algorithm code outside the framework
- Provide greater flexibility in using algorithms
- To provide a means of integrating LArSoft code (data products and algorithms) with external frameworks

- e.g., LArLite used by MicroBooNE for algorithm development, testing

See [art guidance for writing modules](#) for further discussion

All this code lives in LArSoft repositories

Provided by third parties



# LArSoft design principles and objectives

- Separation of framework and algorithm code (cont'd)
  - How??
    - Adhere to particular design patterns for the “LArSoft / art interface” code
      - I.e, art modules and services (to be discussed later)
    - Access framework functionality only within the interface code
      - Includes finding input data, writing output data, retrieving any required services, making filter decision calls, etc.
    - Pass all required data, utility classes into algorithms, and all output data back out

General disclaimer: In examining the code, you may note that only a portion of the existing code adheres to this and the other design principles.

- The on-going architecture review project is intended to address this

Strongly encouraging people to adopt this practice for new code.



# LArSoft design principles and objectives

- Standardized algorithm interfaces
  - Define standard interfaces for well-defined steps in the workflow so that:
    - Multiple algorithms that address specific problems can share interfaces
  - Promote greater modularity, layering of algorithms
  - Generally enhances flexibility of the code
- Modularity
  - Build sophistication by applying small, targeted algorithms in a layered, iterative structure
  - Leads to code that is more easily tested, more maintainable, more flexible

# LArSoft design principles and objectives

- Continuous integration

- A development scheme in which changes to the main branch of development are integrated and tested frequently
  - At every push to the develop branch
  - Every night
  - At every release
- Primary LArSoft goals
  - Ensure that code performs as intended
  - Facilitate early detection of problems created in one experiment due to changes introduced by another experiment
  - Ensure that all major features in the develop branch work at all times
- Are now operating a continuous integration system for LArSoft
  - Currently runs at every push to develop branch
  - Can be triggered manually to run on a non-develop branch of a user's choosing



Requires that code authors write tests!

See <https://cdcvs.fnal.gov/redmine/projects/lar-ci/wiki> for details