LArSoft on art: architecture + typical event data flow
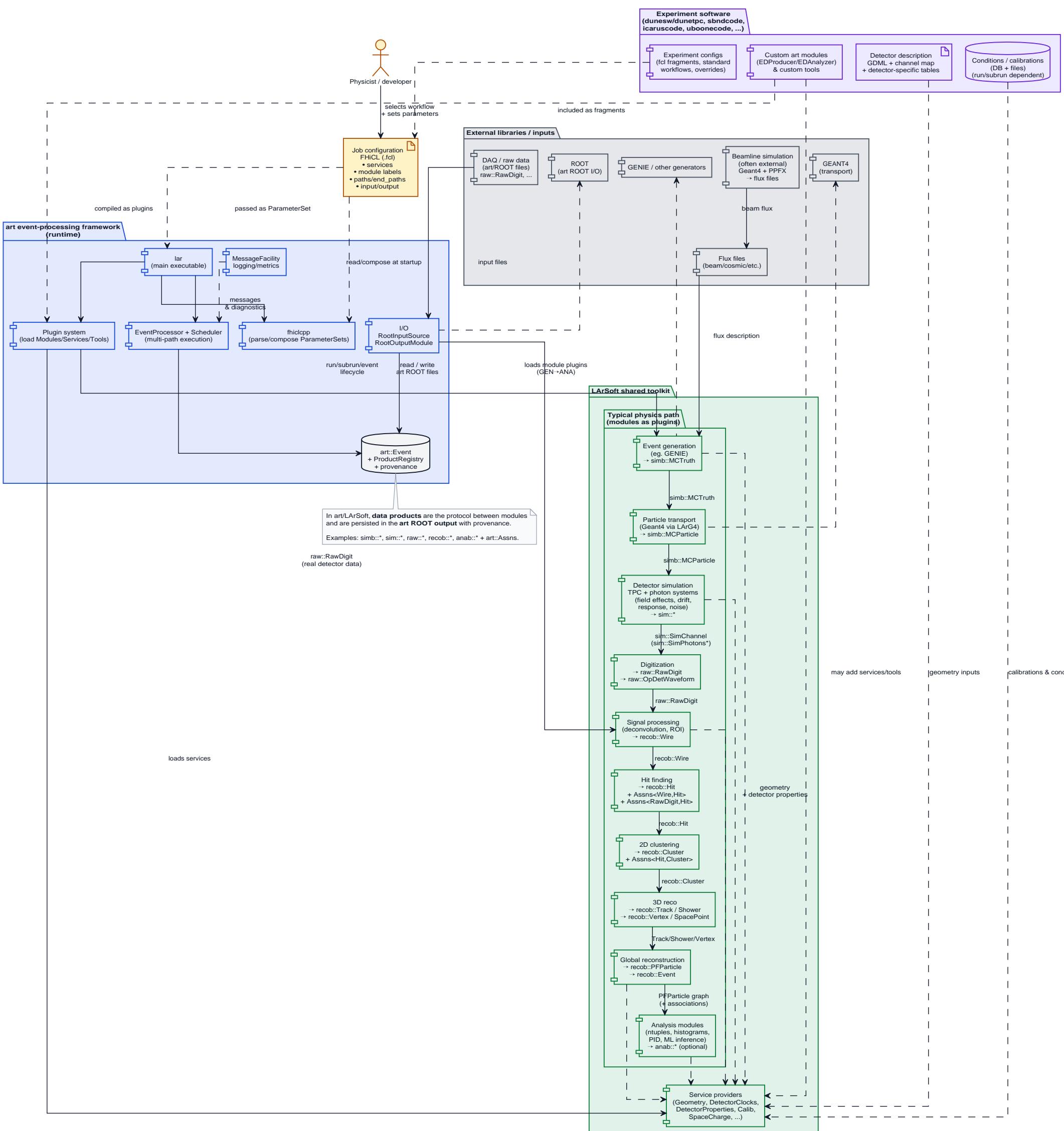Simulation & real data are both handled as art jobs configured by FHiCL

Experiment software
(dunesw/dunetpc, sbndcode,
icaruscode, uboonecode, ...)

Experiment configs
(fcl fragments, standard
workflows, overrides)

Custom art modules
(EDProducer/EDAnalyzer)
& custom tools

Detector description
GDML + channel map
+ detector-specific tables

Conditions / calibrations
(DB + files)
(run/subrun dependent)

Physicist / developer

selects workflow
+ sets parameters

included as fragments

Job configuration
FHiCL (.fcl)
• services
• module labels
• paths/end_paths
• input/output

External libraries / inputs

DAQ / raw data
(art/ROOT files)
raw::RawDigit, ...

ROOT
(art ROOT I/O)

GENIE / other generators

Beamline simulation
(often external)
Geant4 + PPFX
→ flux files

GEANT4
(transport)

compiled as plugins

passed as ParameterSet

read/compose at startup

beam flux

input files

Flux files
(beam/cosmic/etc.)

art event-processing framework
(runtime)

lar
(main executable)

MessageFacility
logging/metrics

flux description

messages
& diagnostics

Plugin system
(load Modules/Services/Tools)

EventProcessor + Scheduler
(multi-path execution)

fhiclcpp
(parse/compose ParameterSets)

I/O
RootInputSource
RootOutputModule

run/subrun/event
lifecycle

read / write
art ROOT files

loads module plugins
(GEN→ANA)

LArSoft shared toolkit

Typical physics path
(modules as plugins)

Event generation
(eg. GENIE)
→ simb::MCTruth

art::Event
+ ProductRegistry
+ provenance

simb::MCTruth

In art/LArSoft, **data products** are the protocol between modules
and are persisted in the **art ROOT output** with provenance.

Examples: simb::*, sim::*, raw::*, recob::*, anab::* + art::Assns.

raw::RawDigit
(real detector data)

Particle transport
(Geant4 via LArG4)
→ simb::MCParticle

simb::MCParticle

Detector simulation
TPC + photon systems
(field effects, drift,
response, noise)
→ sim::*

may add services/tools

geometry inputs

calibrations & conditions

sim::SimChannel
(sim::SimPhotons*)

Digitization
→ raw::RawDigit
→ raw::OpDetWaveform

raw::RawDigit

Signal processing
(deconvolution, ROI)
→ recob::Wire

recob::Wire

Hit finding
→ recob::Hit
+ Assns<Wire,Hit>
+ Assns<RawDigit,Hit>

recob::Hit

2D clustering
→ recob::Cluster
+ Assns<Hit,Cluster>

geometry
+ detector properties

recob::Cluster

3D reco
→ recob::Track / Shower
→ recob::Vertex / SpacePoint

Track/Shower/Vertex

Global reconstruction
→ recob::PFParticle
→ recob::Event

loads services

PFParticle graph
(+ associations)

Analysis modules
(ntuples, histograms,
PID, ML inference)
→ anab::* (optional)

Service providers
(Geometry, DetectorClocks,
DetectorProperties, Calib,
SpaceCharge, ...)

Legend

Mark	Meaning
■	art runtime (modules/services/tools, I/O, provenance)
■	LArSoft shared toolkit (simulation, reconstruction, utilities)
■	Experiment layer (geometry, calibrations, custom config/modules)
■	External deps & inputs (ROOT, GEANT4, GENIE, flux, DAQ)
■	User configuration (FHiCL job options)
■	Data-product store / persisted protocol

</size>