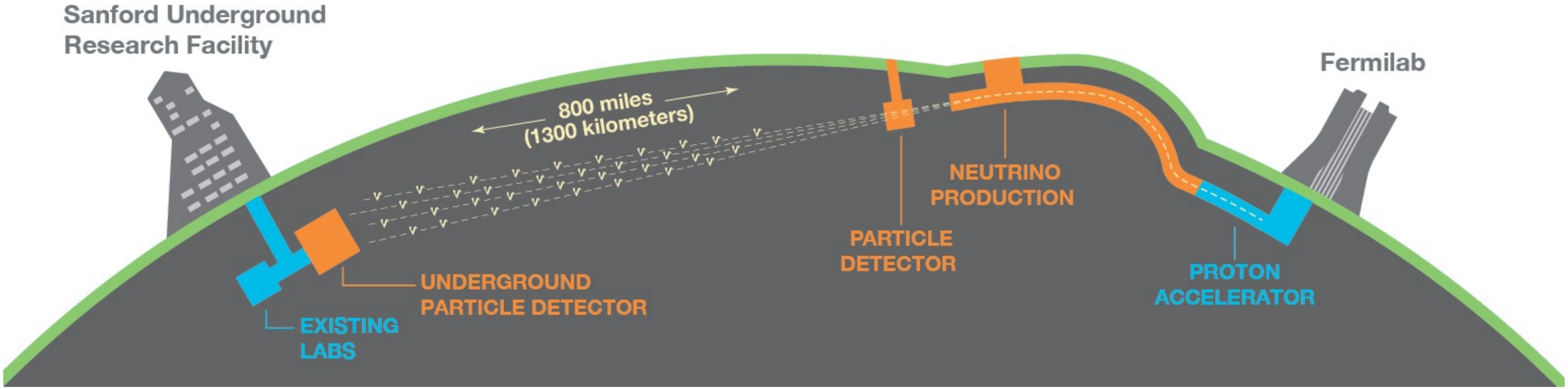
DUNE Near Detector / 2x2 Simulation Workflow

Matt Kramer (LBNL)

HEP-CCE PAW Technical Meeting
Feb. 14, 2024

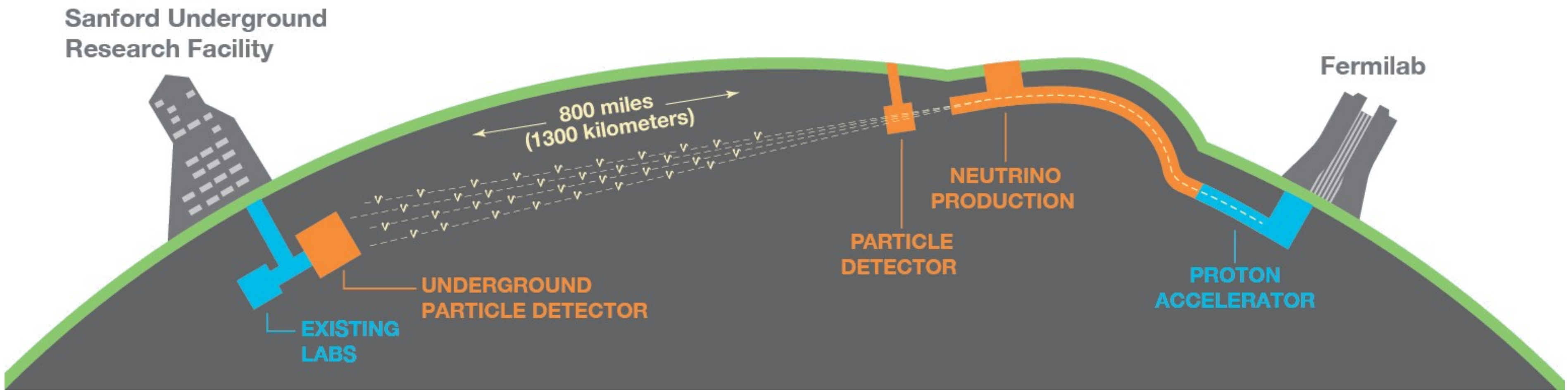
Deep Underground Neutrino Experiment

- Next generation long-baseline accelerator neutrino oscillation experiment in U.S.
- Physics program: Leptonic CP violation, neutrino mass hierarchy, and much more!
- Very long baseline → matter effect enhances mass ordering sensitivity
- Requires higher-energy beam → complicated interaction modes
- LArTPC technology for detailed detection abilities
- Very intense beam and very large far detectors → systematics must be understood

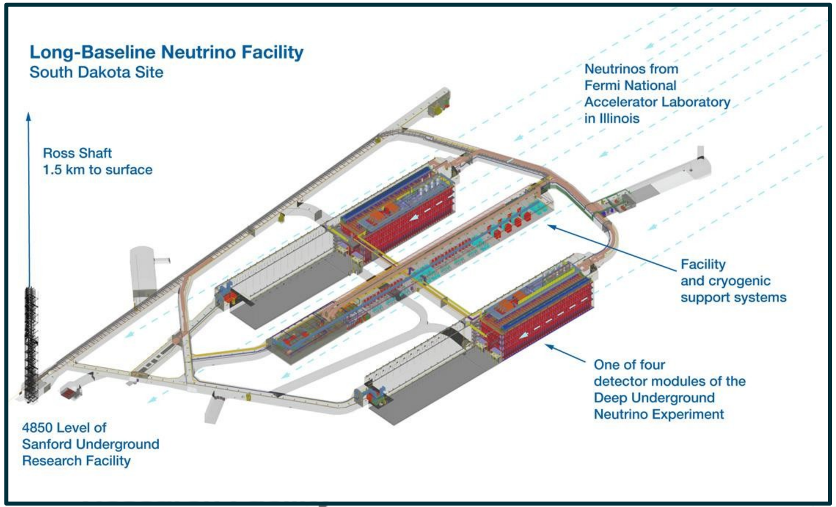


DUNE's beam

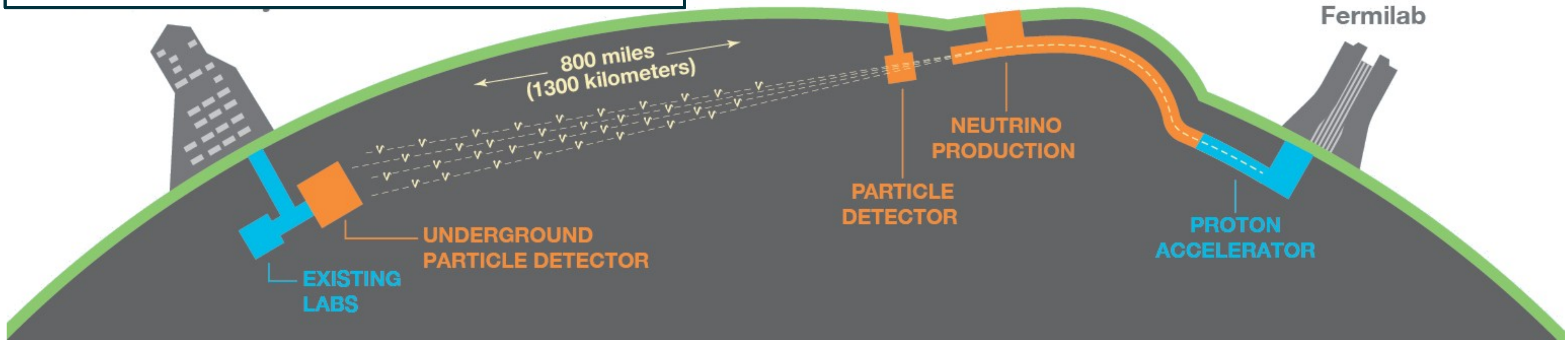
- 120 GeV protons slam into a target
- Outgoing π+ (π-) are focused by magnetic horns
- Pions in decay tunnel produce muon (anti)neutrinos
- After 800 miles through the earth, they reach the far detectors



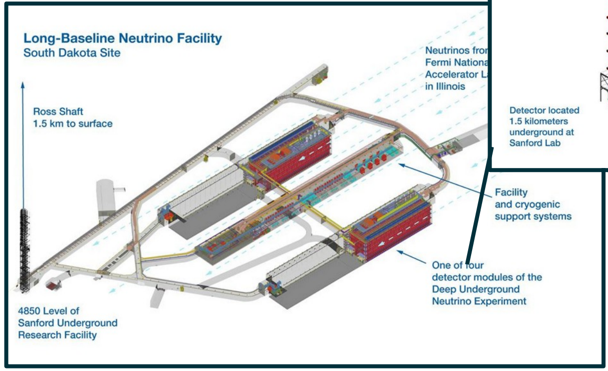
DUNE's far detectors

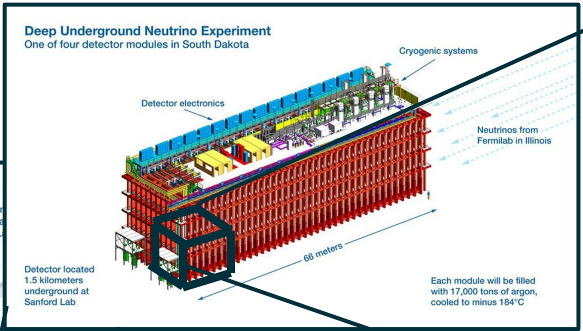


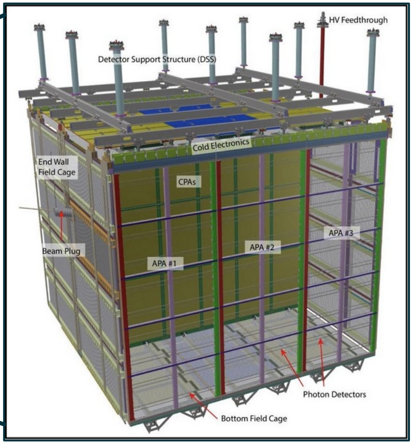
- Excavation complete!
- Phase I: 2 single-phase LArTPCs ~17kton each
  - Vertical drift
  - Horizontal drift
- Phase II: addition of 3rd module

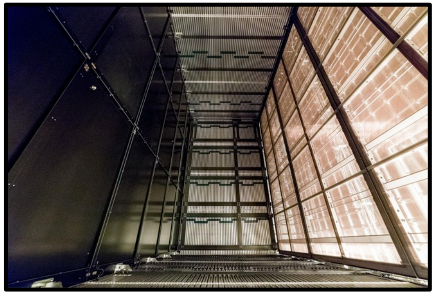


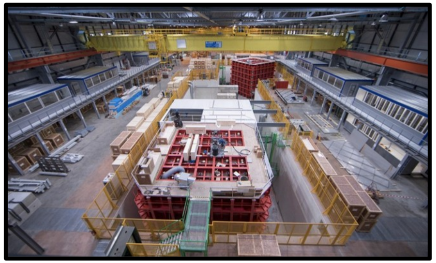
DUNE’s far detectors

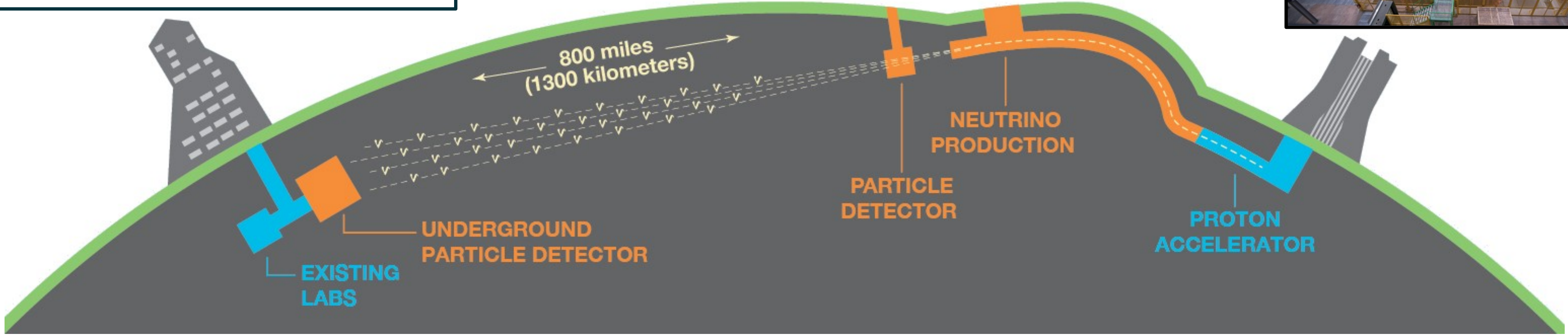






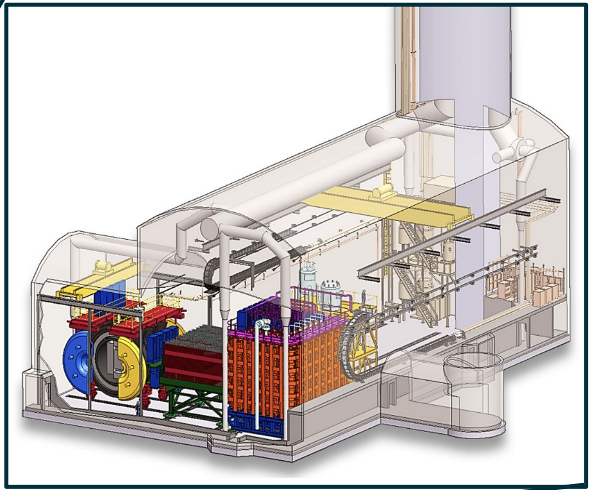


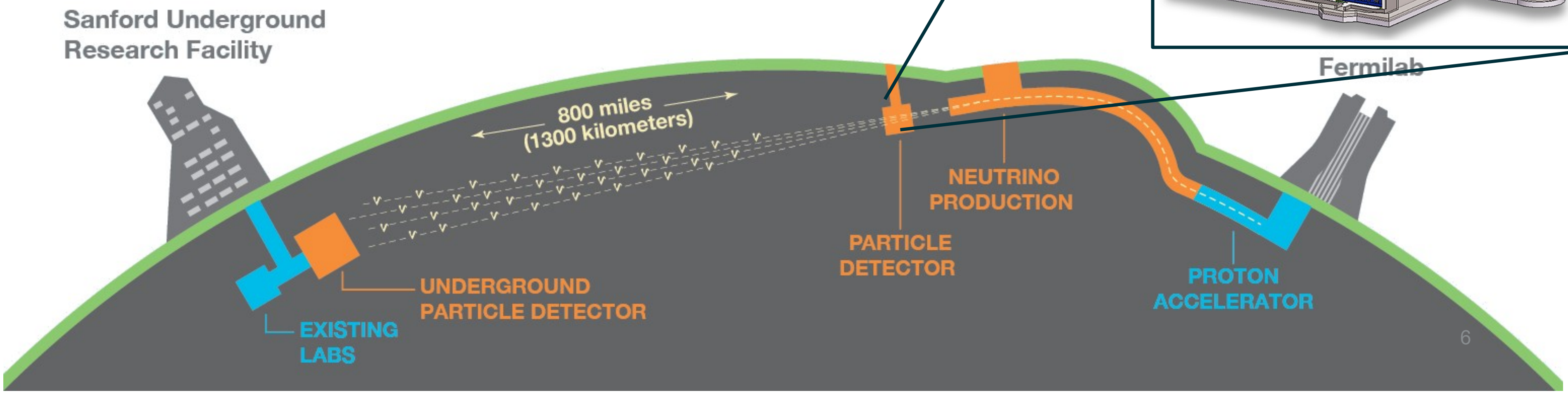




DUNE's near detector complex

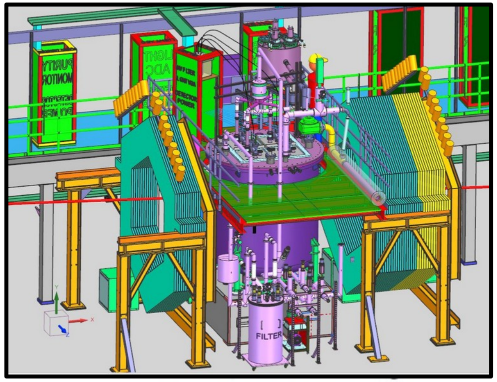
- ND-LAr: LArTPC, analogous to FD
- Muon spectrometer (TMS)
  - Phase II: Upgrade to e.g. GArTPC
- SAND: Monitor flux, constrain interaction systematics
- PRISM: Move ND-LAr + TMS up to 30 m off-axis

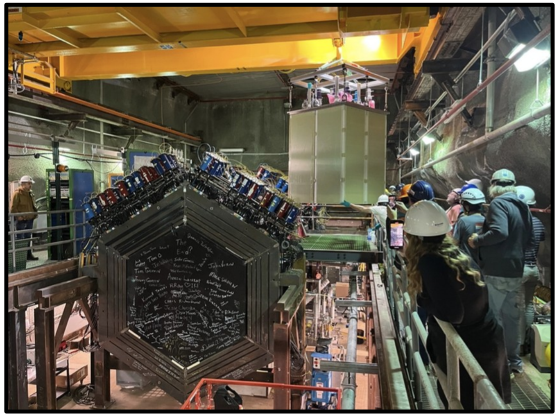


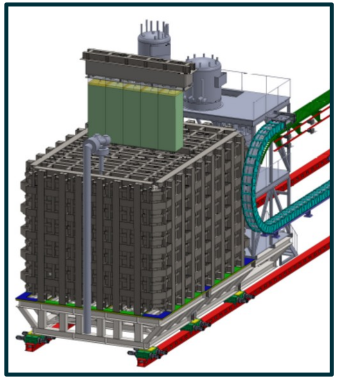


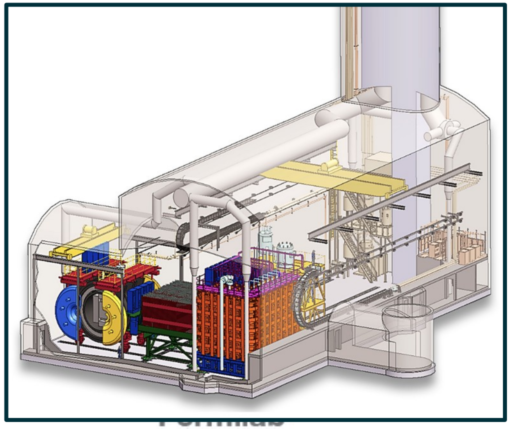
DUNE's LAr near detector

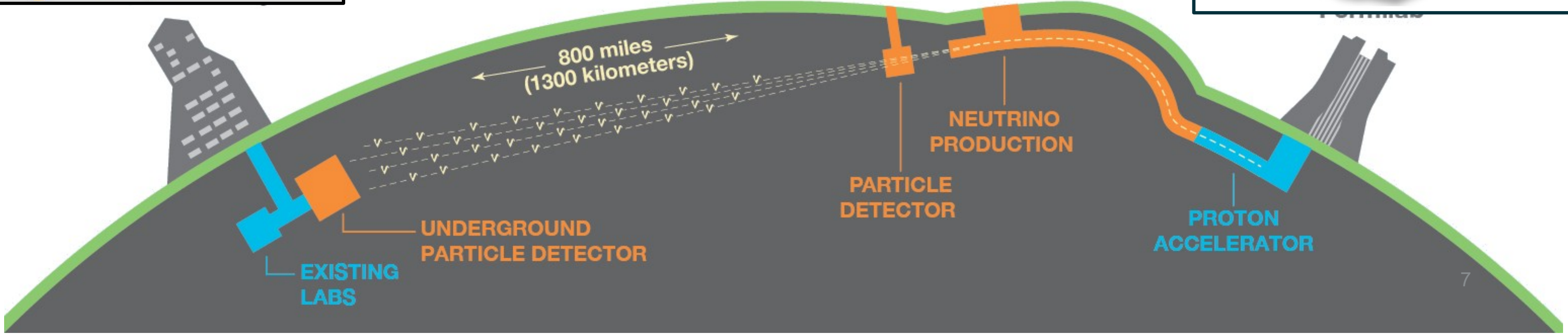
2x2 demonstrator
now commissioning!







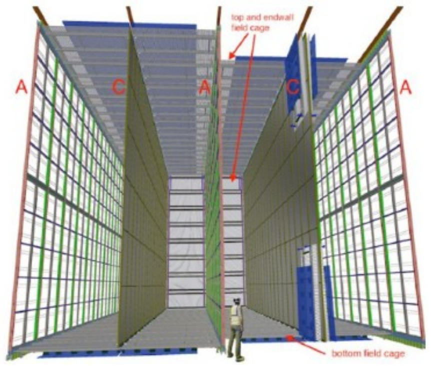


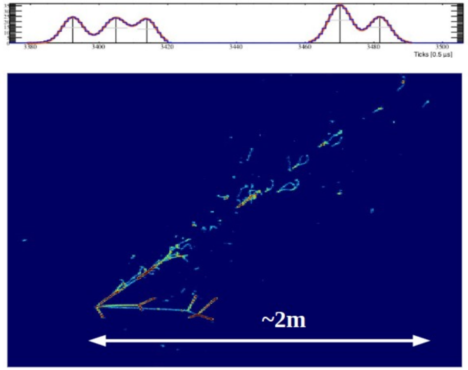


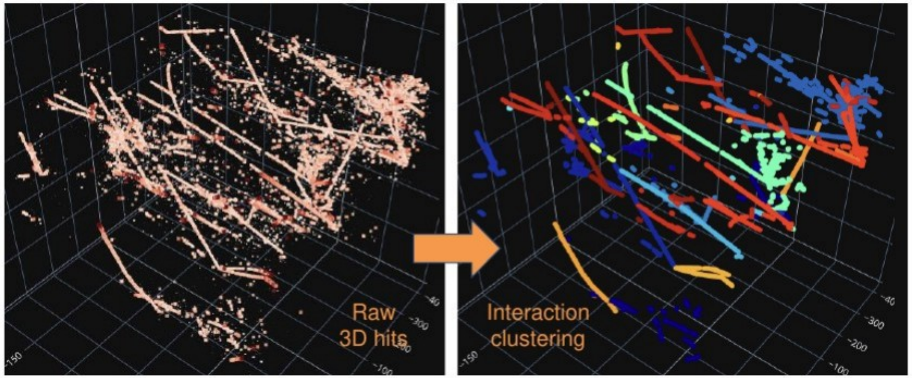
FD vs ND computing: Different challenges

- **FD** — 5 GB per beam trigger @ ~mHz: 3M wires, digitized waveforms, sparse activity
  - Main challenge: Memory management
  - “Traditional” signal processing and reconstruction
  - Uses a software framework developed for previous LArTPCs
  - Currently runs almost entirely on CPUs (but greater future role of GPUs is likely)

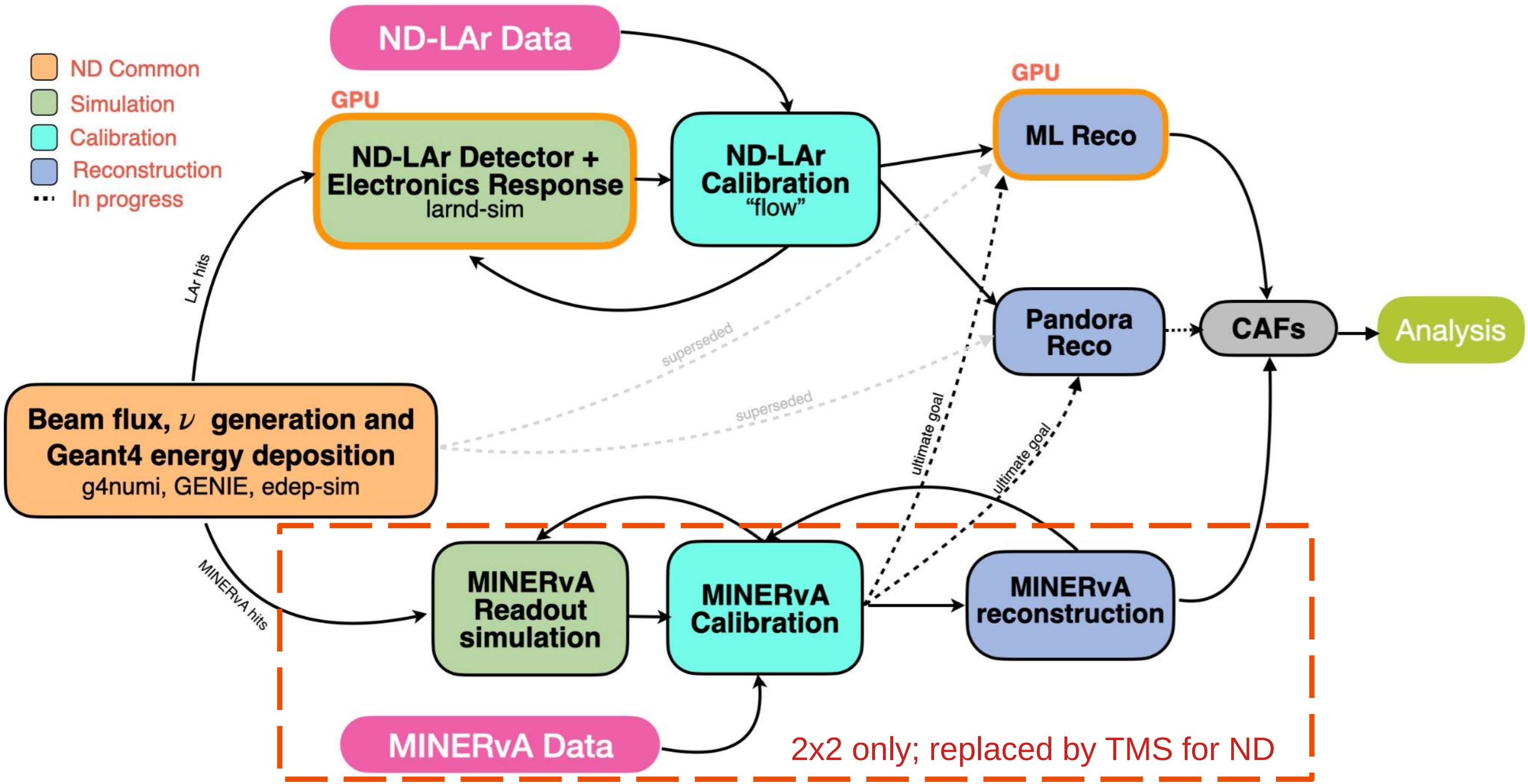
- **ND(-LAr)** — 5 MB for every beam spill @ ~Hz: 12M pixels, digitized free-streaming hits, dense activity
  - Unique feature: High channel count, lending itself to parallelized simulation (on GPUs)
  - ML algorithms (on GPUs) ideal for reconstruction
  - High beam intensity: bulk of DUNE’s computing requirements
  - **Loosely coupled, heterogenous software stack**





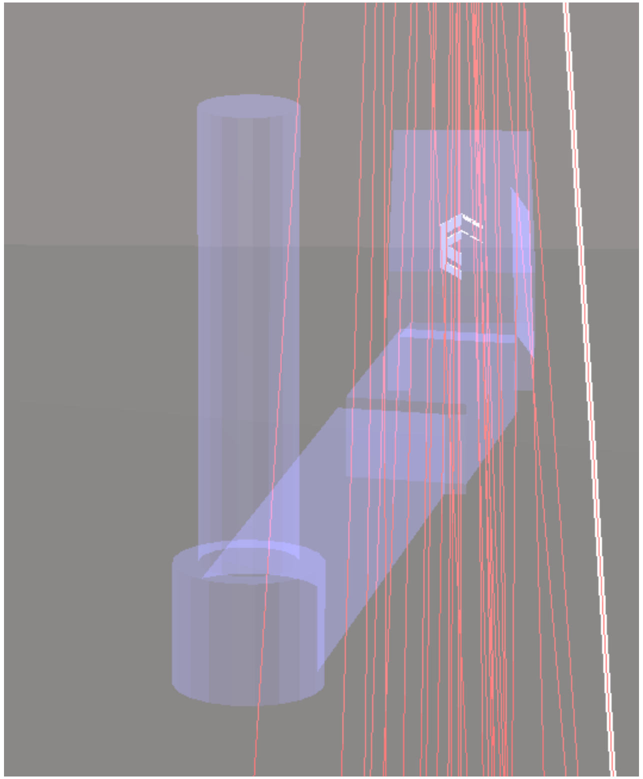


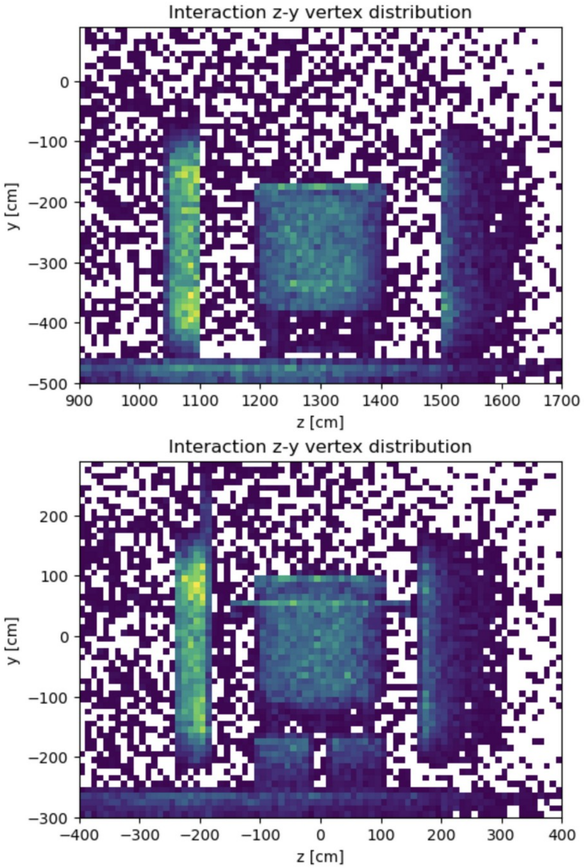
ND/2x2 workflow: Bird's eye view



GENIE + edep-sim

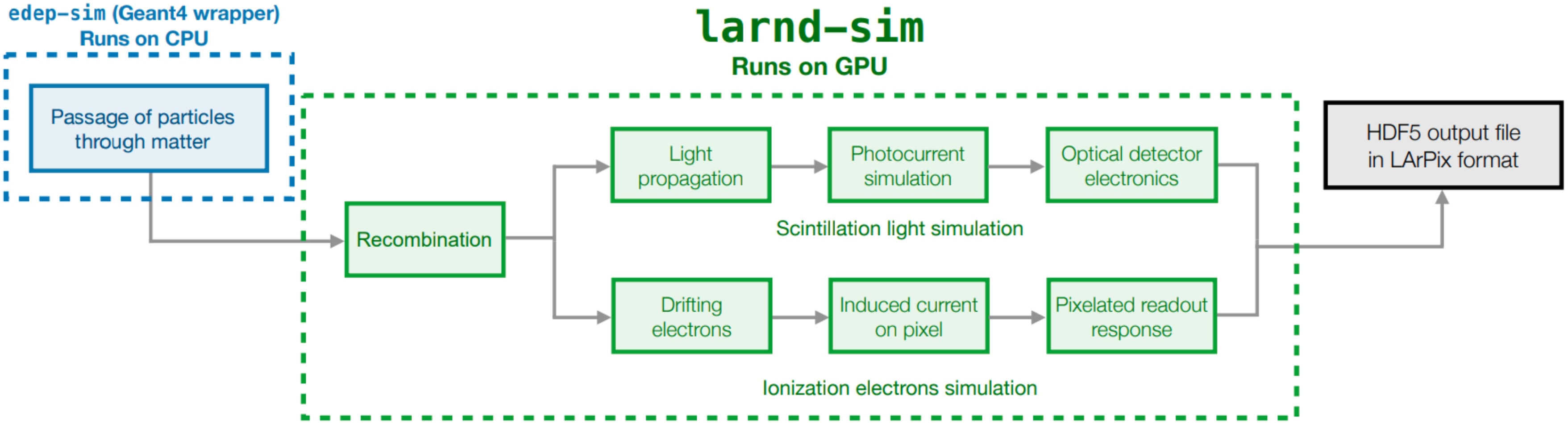
- Start with pregenerated flux files from g4beamline
- Pass to GENIE event generator + GDML geometry
- Propagate particles, record energy deposits using edep-sim (Geant4 wrapper)



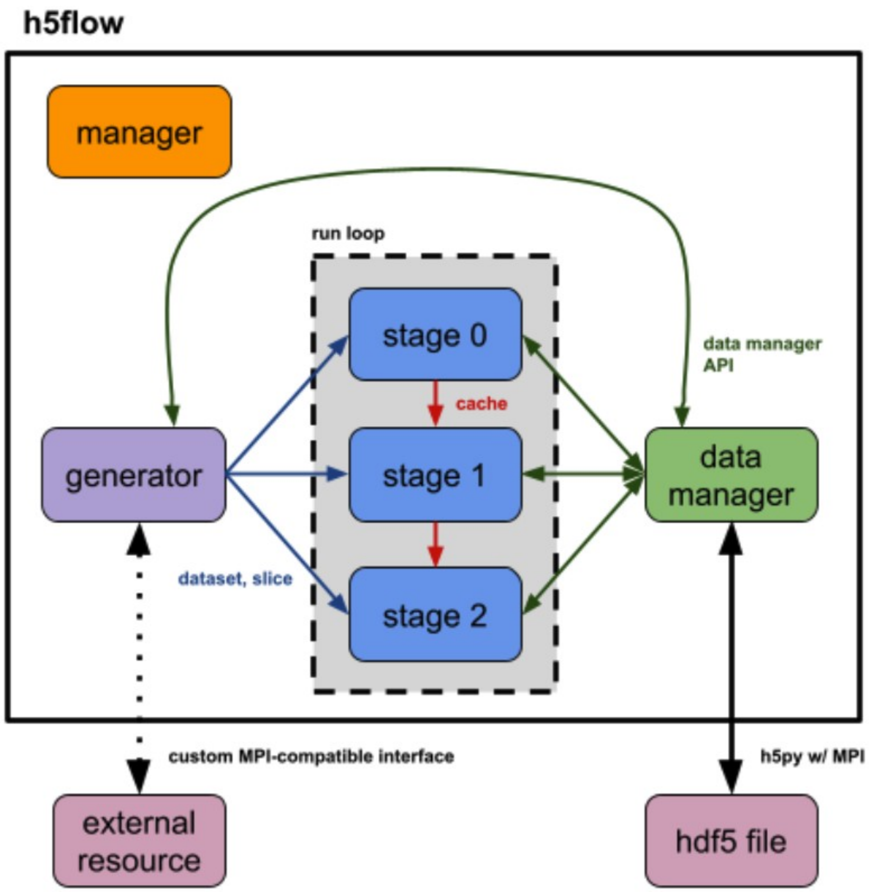


larnd-sim: GPU-accelerated simulation

- Completely written in Python (Numba)
- All heavy computations on GPU (~15 kernels)
- Largely developed by 2 people
  - Physicists, not GPU experts: A testament to Numba's accessibility
- Input: edep-sim energy deposits in HDF5
- Output: "Packet" data, as from DAQ; plus MC truth info
- Idiomatic Python, JIT-compiled to CUDA
  - Just apply @numba.cuda.jit decorator
  - cupy: numpy on the GPU



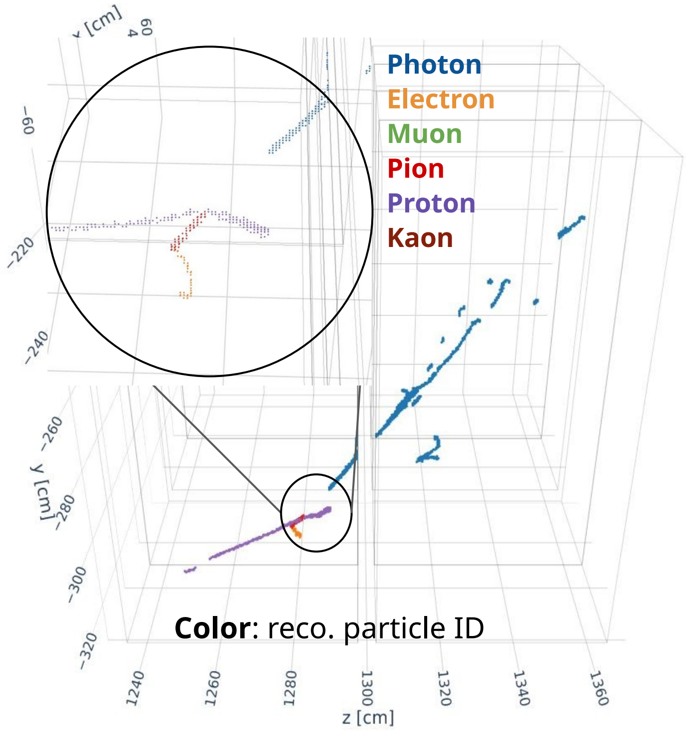
flow: Low-level calibration



- Pure Python; runs on CPUs
- Performance from numpy ops, avoiding loops
- Built on "h5flow" framework:
  - Can be mentally mapped onto <insert_framework_here>, but much simpler
  - "Automatic" parallelism: Dataset slices distributed via MPI (not currently used)
  - Flexible configuration via YAML files
  - Provenance tracking: Reference links are stored between parent and child datasets
    - Dereferencing possible in both directions, across multiple links
  - Start with arrays of raw data; successively add arrays of higher-level quantities

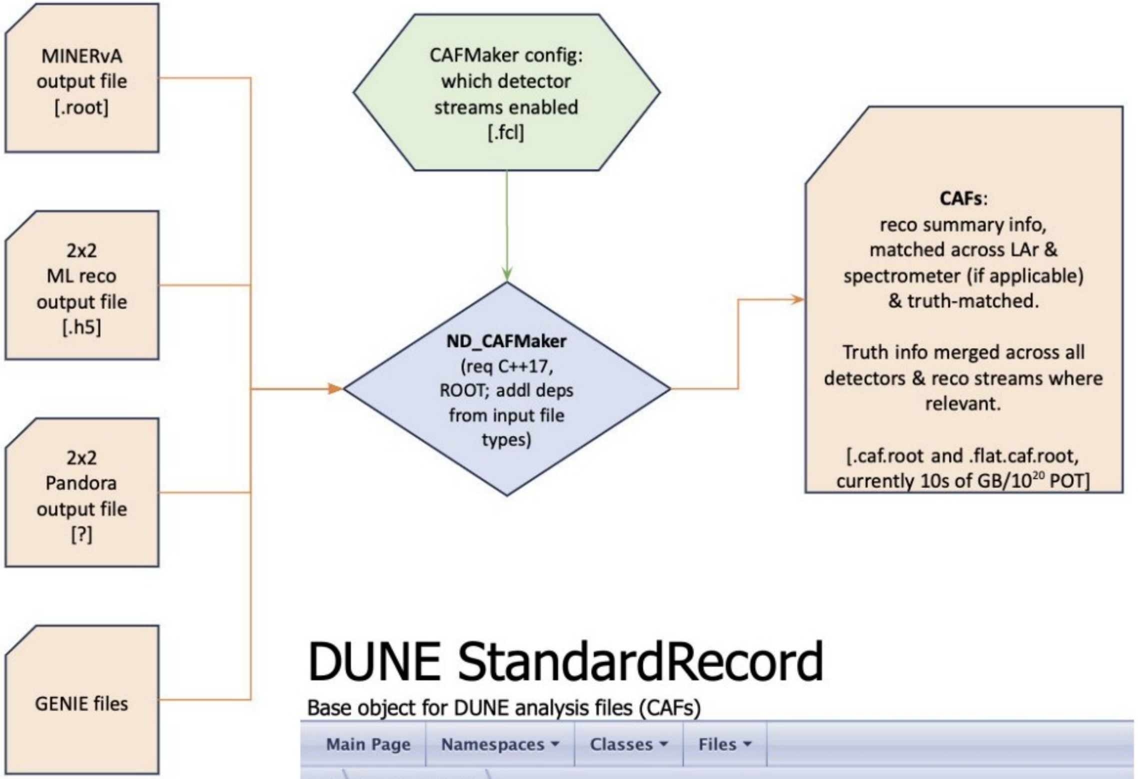
Two reconstruction packages

- mlreco: Machine learning reconstruction
  - Development led by DeepLearnPhysics group at SLAC
  - Neural networks for clustering, vertexing, particle ID, etc.
  - GPU-native (PyTorch)
- Pandora: “Traditional” reconstruction
  - Modular collection of processing and pattern-recognition algorithms
  - Also used for FD and by other LArTPC experiments
  - C++/CPU, but supports optional ML algorithms



ML-Based Reconstruction for 2x2, F. Drielsma (SLAC)

CAFs: Common analysis files



- Structured ROOT files for high-level analysis
- Common format for ND/FD, MC/data, multiple analysis packages
- Separate "CAF makers" for ND, FD
- ND/2x2 require matching across detectors

DUNE StandardRecord

Base object for DUNE analysis files (CAFs)

Main Page | Namespaces | Classes | Files

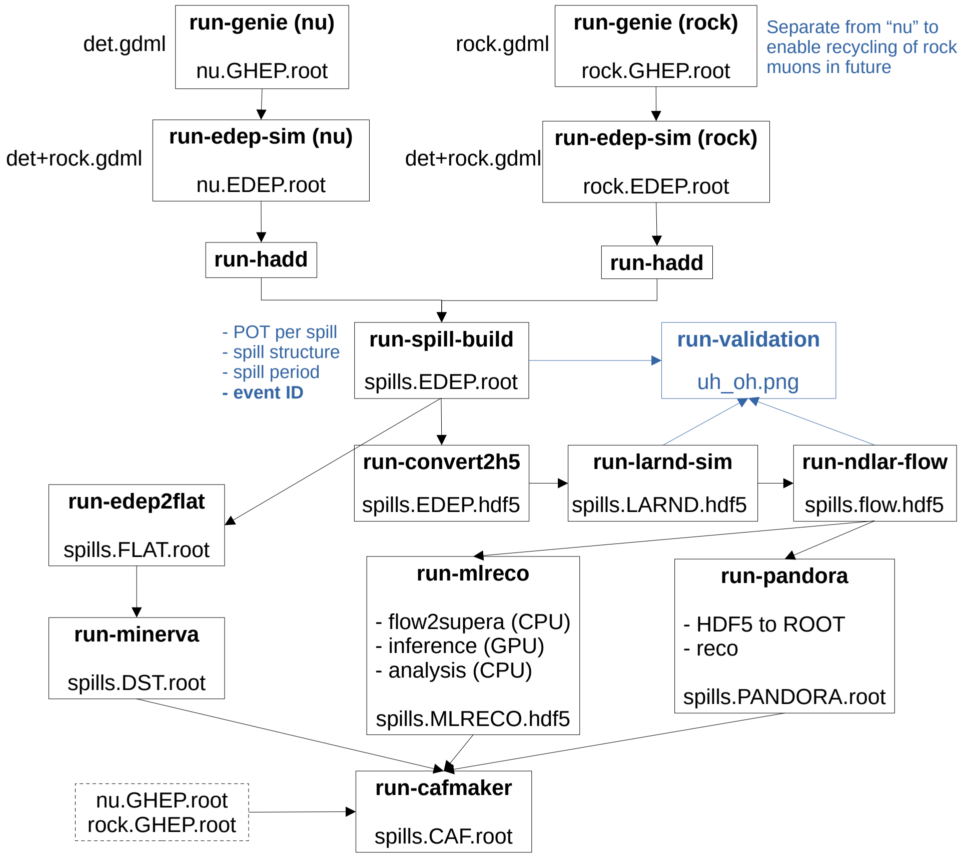
caf StandardRecord

caf::StandardRecord Class Reference

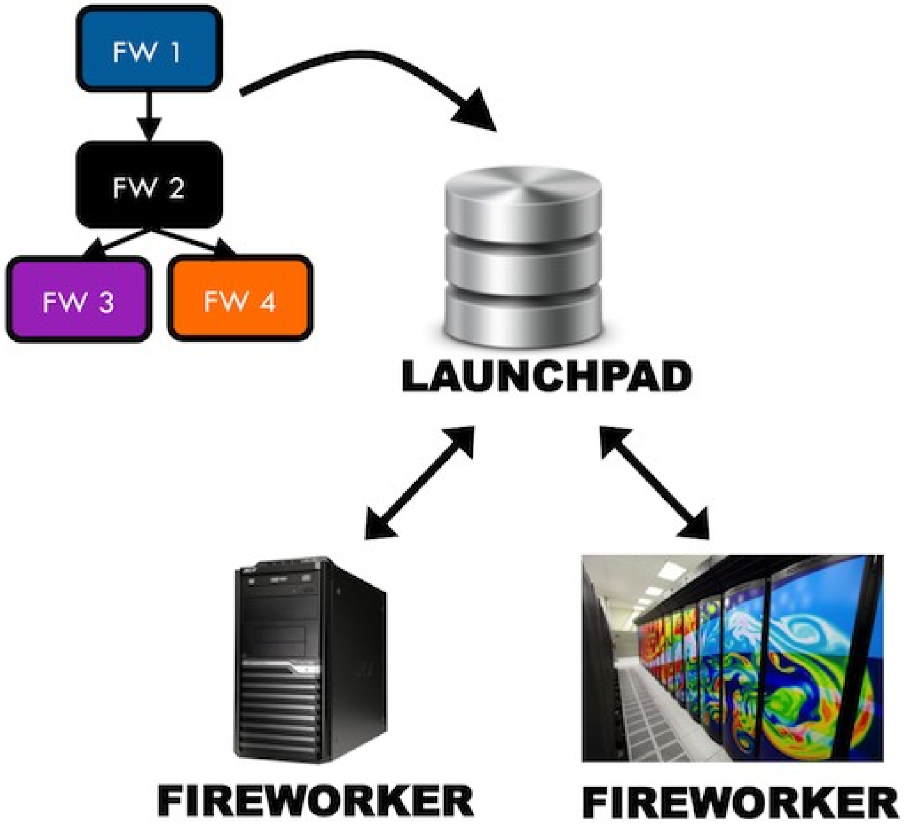
SRDetectorMetaBranch	**meta**	Metadata about the detectors.
SRBeamBranch	**beam**	Information about the beam configuration and beam pulse for this event.
SRTruthBranch	**mc**	Truth information.
SRCommonRecoBranch	**common**	Reconstructed info expected to be common to all (?) detectors.
SRFDBranch	**fd**	Reconstructed info unique to the FDs.
SRNDBranch	**nd**	Reconstructed info unique to the ND complex.

2x2_sim: The shell scripts behind the curtain

- Each step in the chain has its own script
- All configuration from environment variables
  - Easy to wrap in a variety of workflow managers
- Scripts can relaunch themselves in appropriate container
- No concept of inter-step dependencies
  - Reponsibility of workflow manager



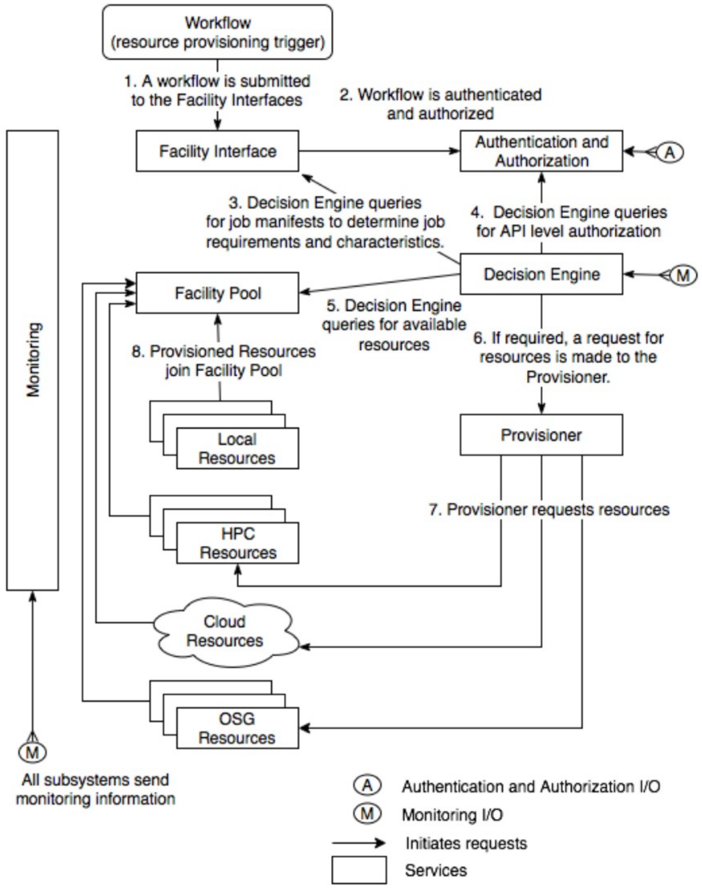
FireWorks: Site-local workflow management



- Initial simulation campaigns use FireWorks workflow manager
- MongoDB contains task definitions, statuses, dependencies, etc.
  - Hosted on NERSC's Spin cluster (Kubernetes)
- Direct submission of Slurm pilot jobs on Perlmutter
- Fine-grained matching of work to pilots using categories, priorities, etc.
- All FireWorks-related code, workflow definitions, etc live in a separate repo from 2x2_sim

DUNE production tools

- Future simulation campaigns to use common DUNE production and workflow tools
  - Battle-tested in FD campaigns
  - Flexibility to run at various sites beyond NERSC (OSG, etc.)
  - Advanced monitoring tools
- POMS (Production Operations Management System) serves analogous role as FireWorks
- Work submitted as HTCondor jobs at FNAL, routed to NERSC via HEPCloud, run by GlideinWMS pilots submitted to Slurm on demand
- Next-gen workflow manager justIN to allow matching of jobs to sites based on data availability
- Data management with Rucio



Containers

Build

```
> vim ./Dockerfile
> podman-hpc build -t me/myimage:latest .
```

Ship

```
> docker push me/myimage:latest
```

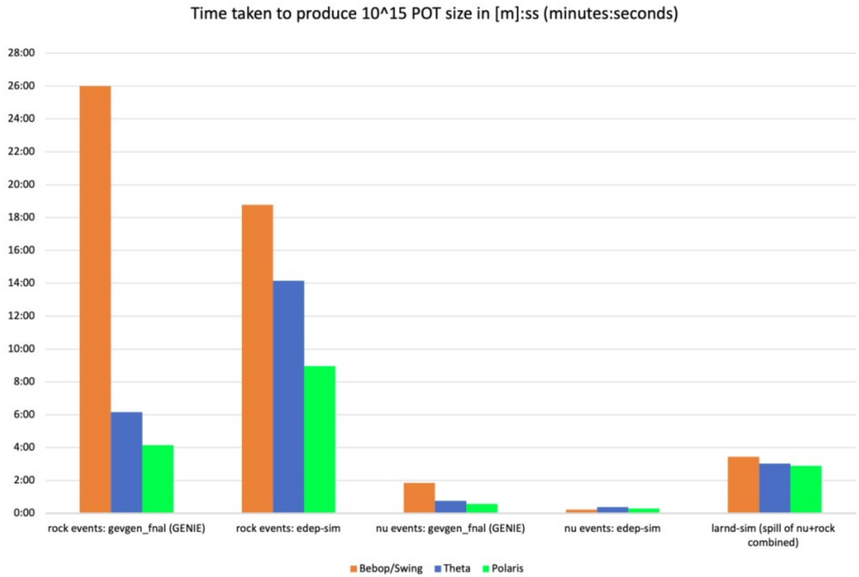
Run

```
> podman-hpc mig me/myimage:latest
> salloc -C cpu
> srun --ntasks-per-node=$SLURM_TASKS_PER_NODE
podman-hpc run-shared me/myimage:latest
```

- Containers used to decouple the software stacks of different steps
- Originally used a custom GENIE + edep-sim image built with Apptainer, converted to Docker, pulled into Shifter
- Now can use podman-hpc to build images directly on Perlmutter
- Moving toward generic FNAL SL7 container; load software from CVMFS
  - Same approach as FD
- mlreco currently still uses custom image
  - Switch HEPCloud from using Shifter to Apptainer or Podman; then the SL7 container (needed for Condor deps) can nest the mlreco container

Beyond NERSC: Running at Argonne

- Portability success story: 2x2_sim run by ANL colleagues
- Currently wrapped in Balsam workflow manager
  - Longer-term switch to common tools (POMS, etc.) likely
- larnd-sim run on Polaris (Nvidia GPUs)
- Porting larnd-sim to Aurora (Intel GPUs): An exciting challenge



Conclusion

- The DUNE near detector (and 2x2 demonstrator) currently use a heterogenous sim/reco chain requiring both CPUs and GPUs
  - Eventual goal is to get FD and ND code to live in a common framework (currently being defined)
- We've found happiness in disentangling our scripts (e.g. run_genie.sh) from the workflow managers that run them
  - Wrapped with 3 solutions so far:
    - FireWorks at NERSC
    - POMS+HEPCloud at NERSC (submitted at FNAL)
    - Balsam at ANL
- We hope our experience can provide a useful case study!