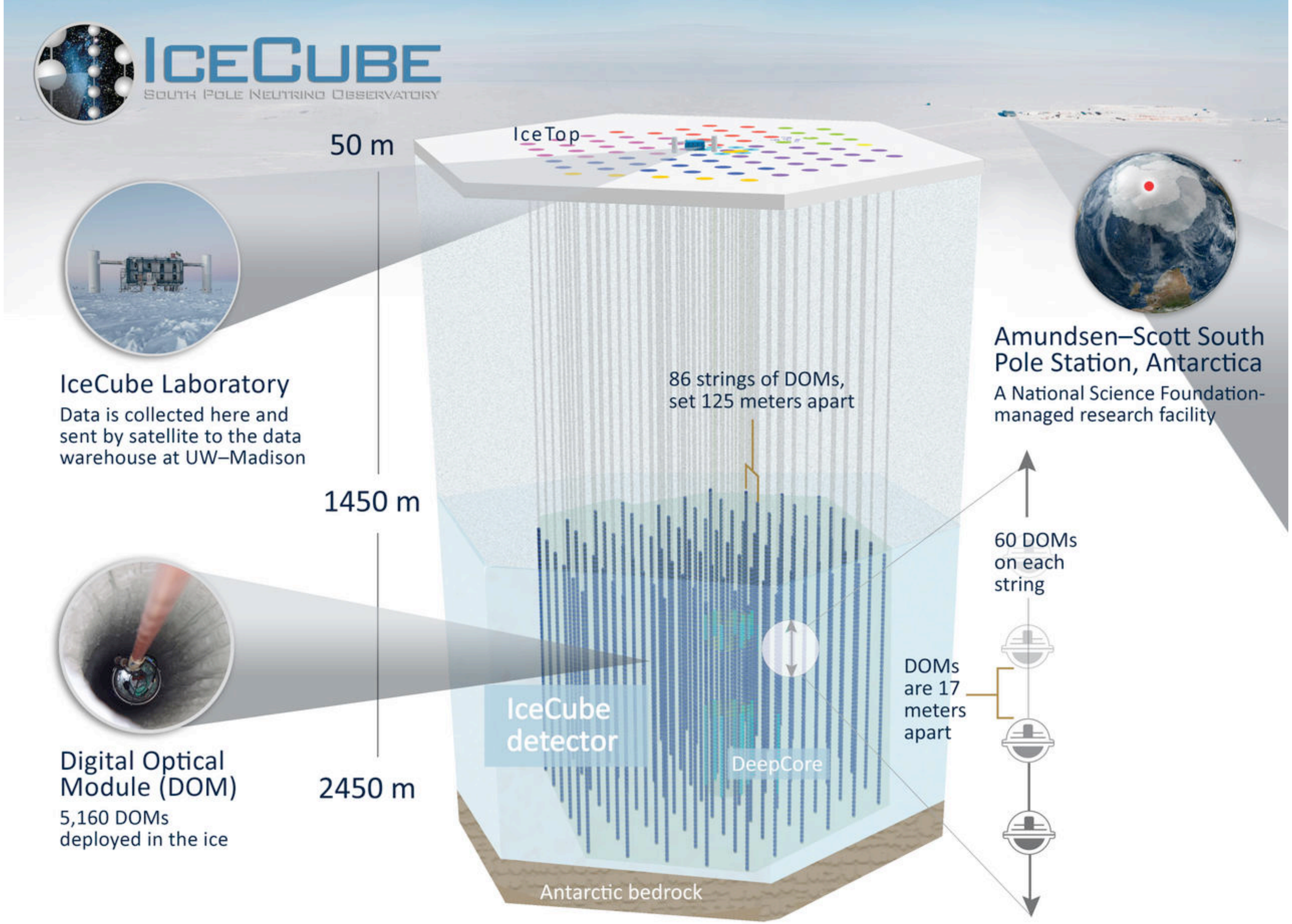
Parallelization of Air Shower Simulation with IceCube

Kevin Meagher
University of Wisconsin
May 8, 2023
CHEP2023 Norfolk, VA

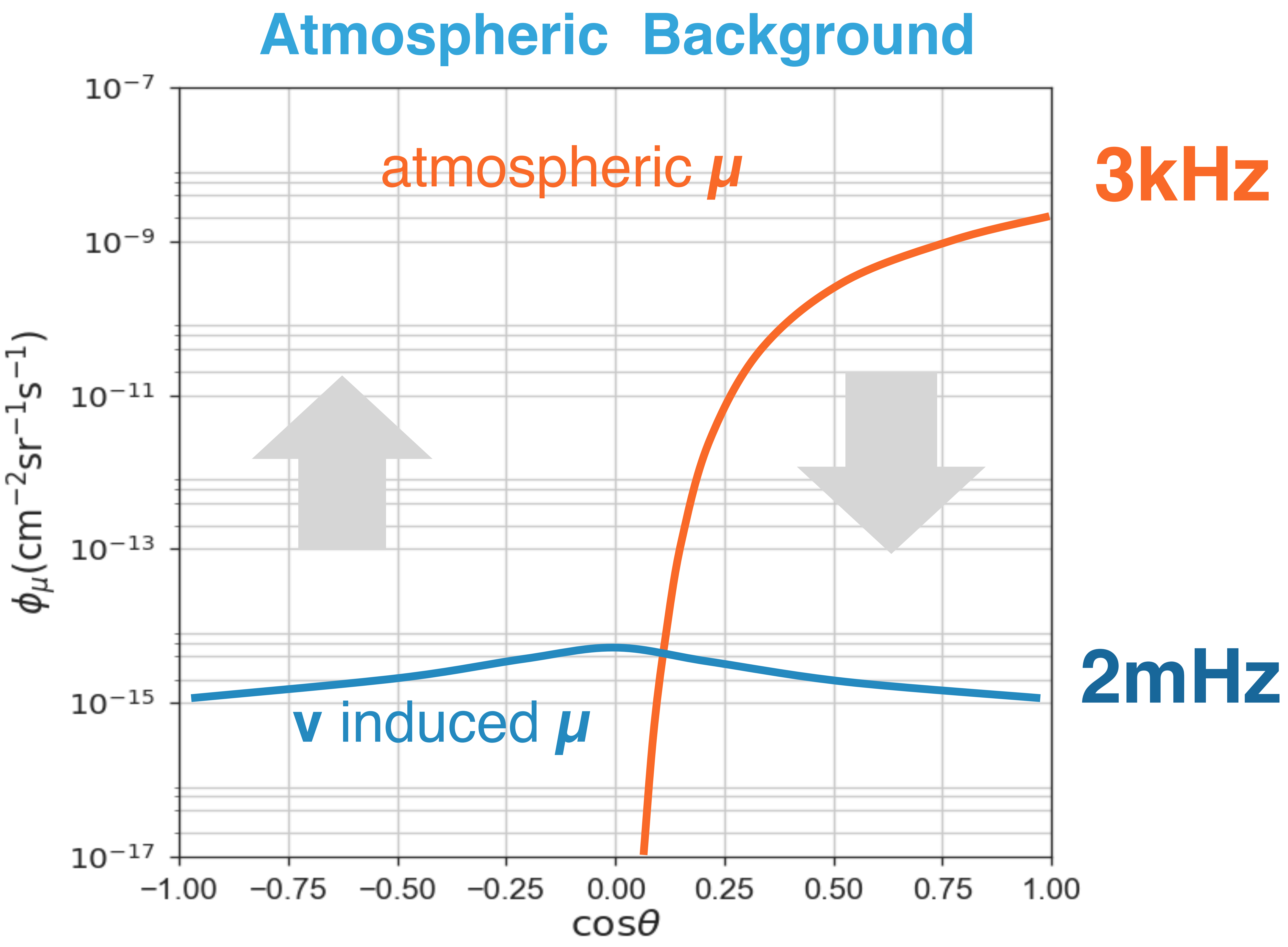
Overview

- Introduction to IceCube Neutrino Observatory
- IceCube's old simulation chain for cosmic-ray air showers
- Improvements to the simulation chain
- Results

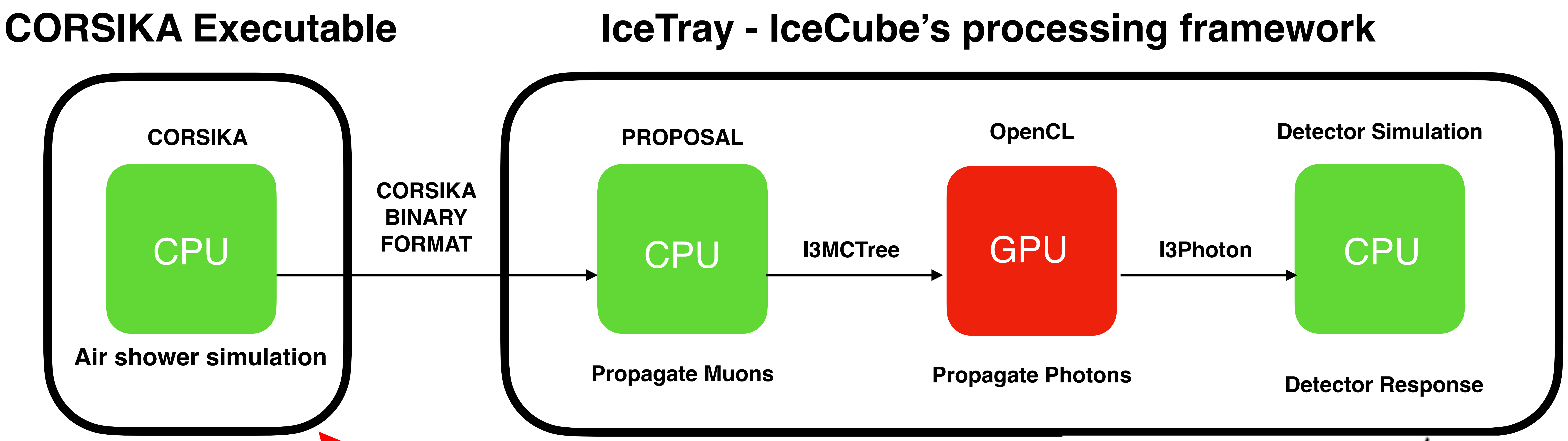

ICECUBE
SOUTH POLE NEUTRINO OBSERVATORY
50 m
IceTop
IceCube Laboratory
Data is collected here and
sent by satellite to the data
warehouse at UW–Madison
1450 m
Digital Optical
Module (DOM)
5,160 DOMs
deployed in the ice
2450 m
86 strings of DOMs,
set 125 meters apart
IceCube
detector
DeepCore
Antarctic bedrock
Amundsen–Scott South
Pole Station, Antarctica
A National Science Foundation-
managed research facility
60 DOMs
on each
string
DOMs
are 17
meters
apart

IceCube Needs a Lot of Background Simulation

- Cosmic-ray air showers occur at a rate of about 3kHz
- Neutrinos at a rate of about 2mHz
- An event selection that keeps 1 in 10^6 means we need a lot of background simulation
- Old simulation chain suffered from low CPU utilization, IO bottlenecks, and memory spikes
- Trend in Computing centers less Memory per CPU core

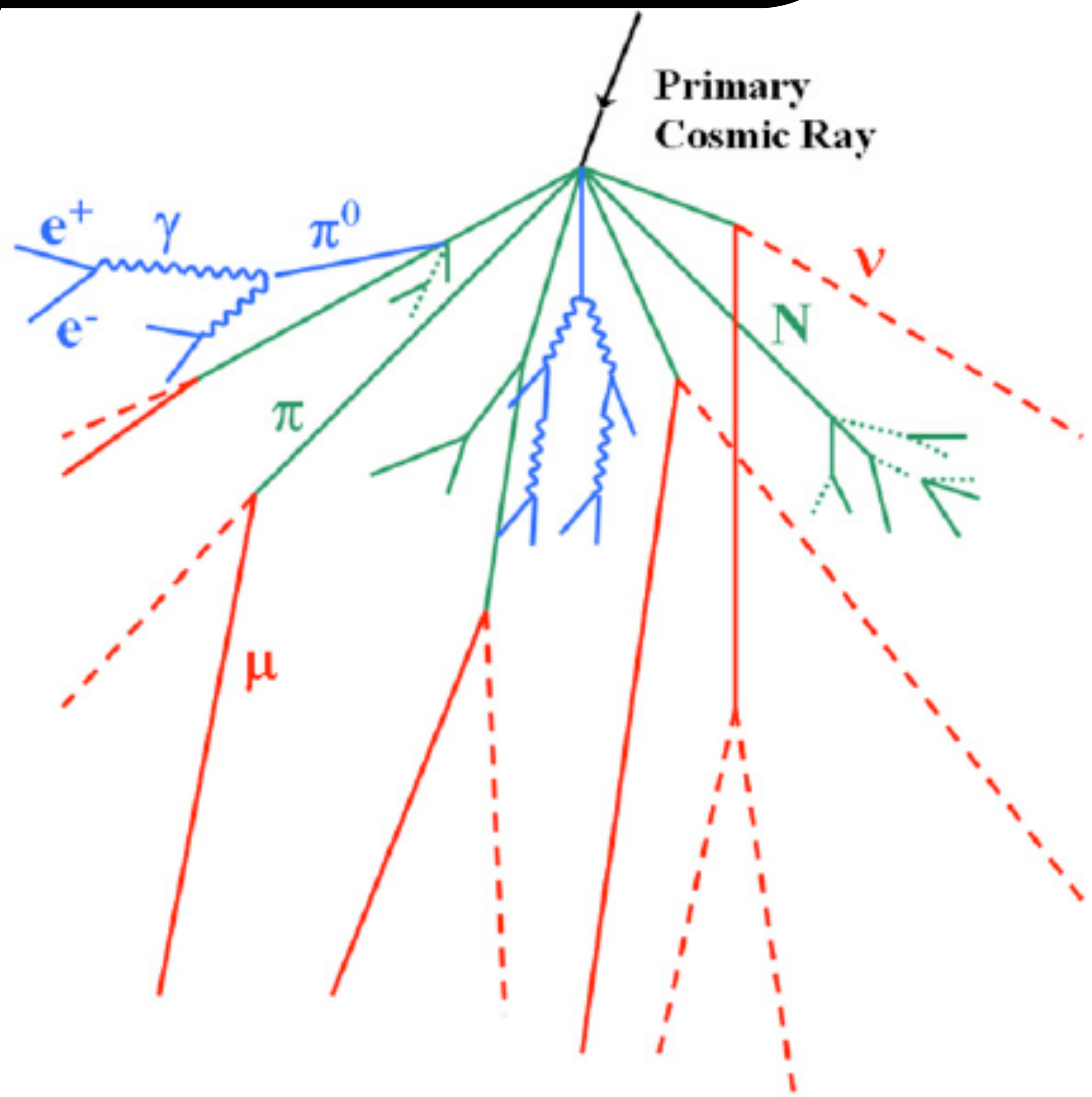


Old Simulation Chain

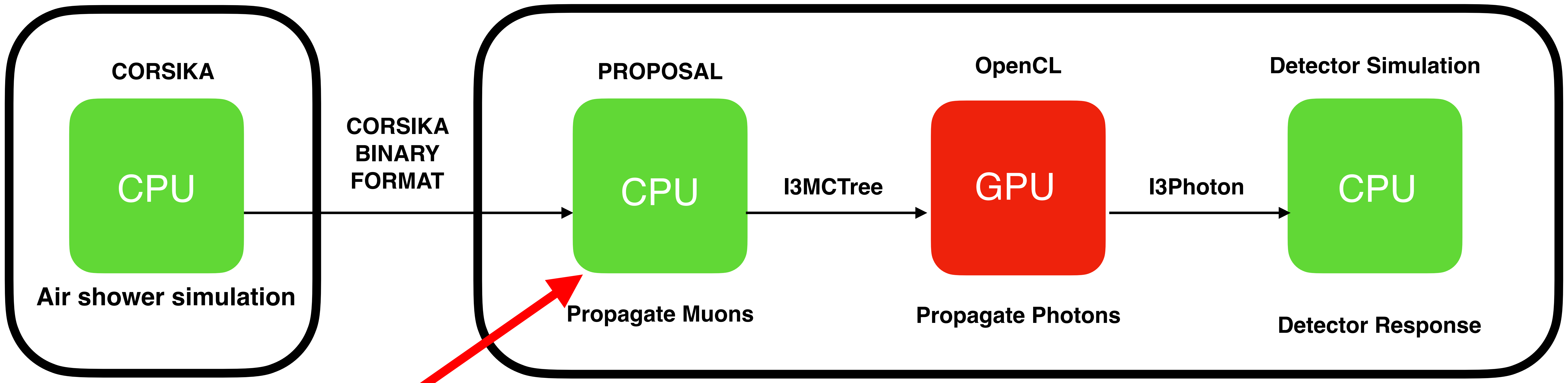


CORSIKA

- Simulates the development of Cosmic-ray air showers
- The energy distribution of daughter particles is highly stochastic
- Muons are the only daughter particle which is relevant for simulating IceCube’s deep ice detectors
- CORSIKA is a single 100k fortran file that has to be compiled as a separate binary

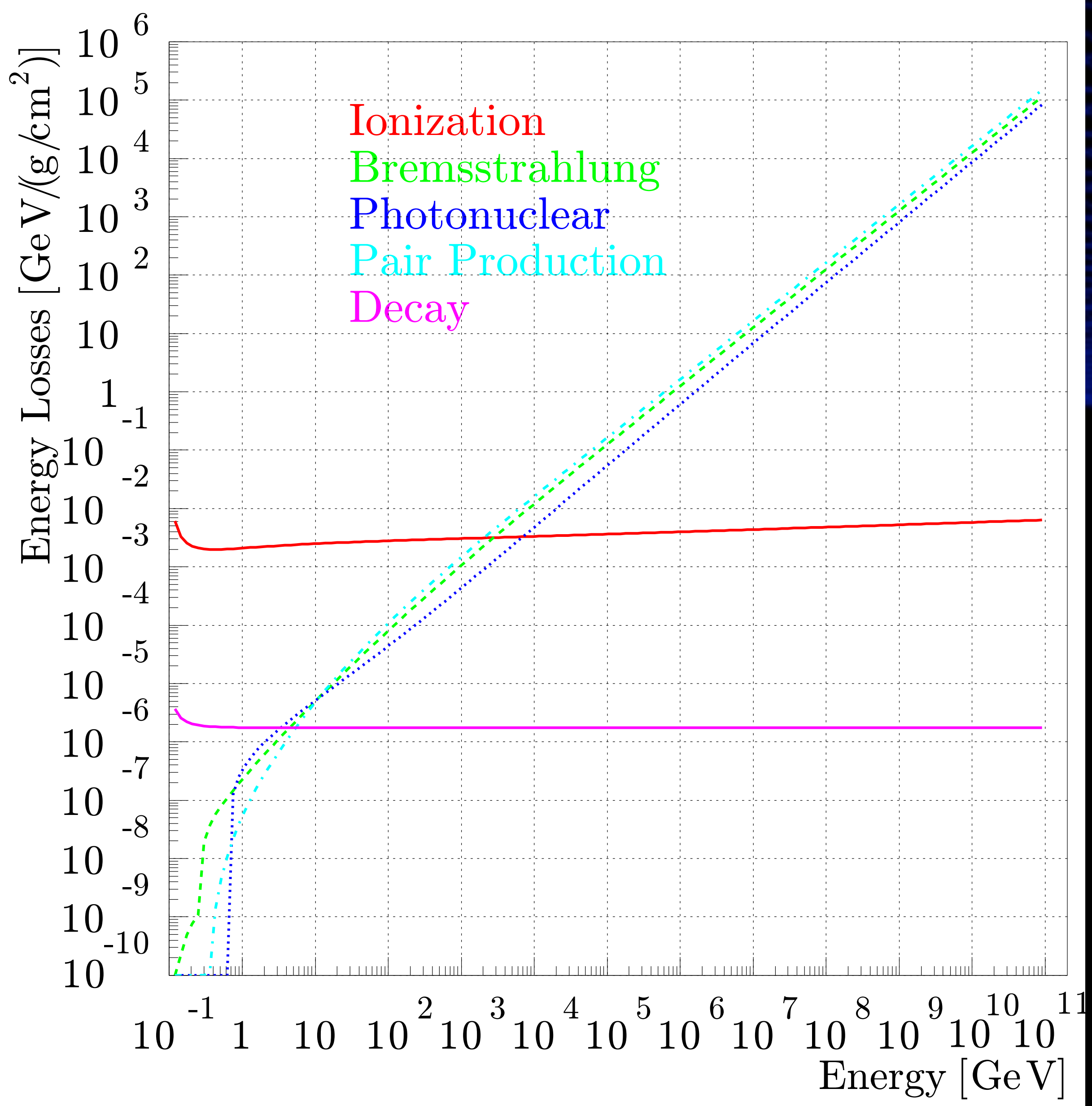


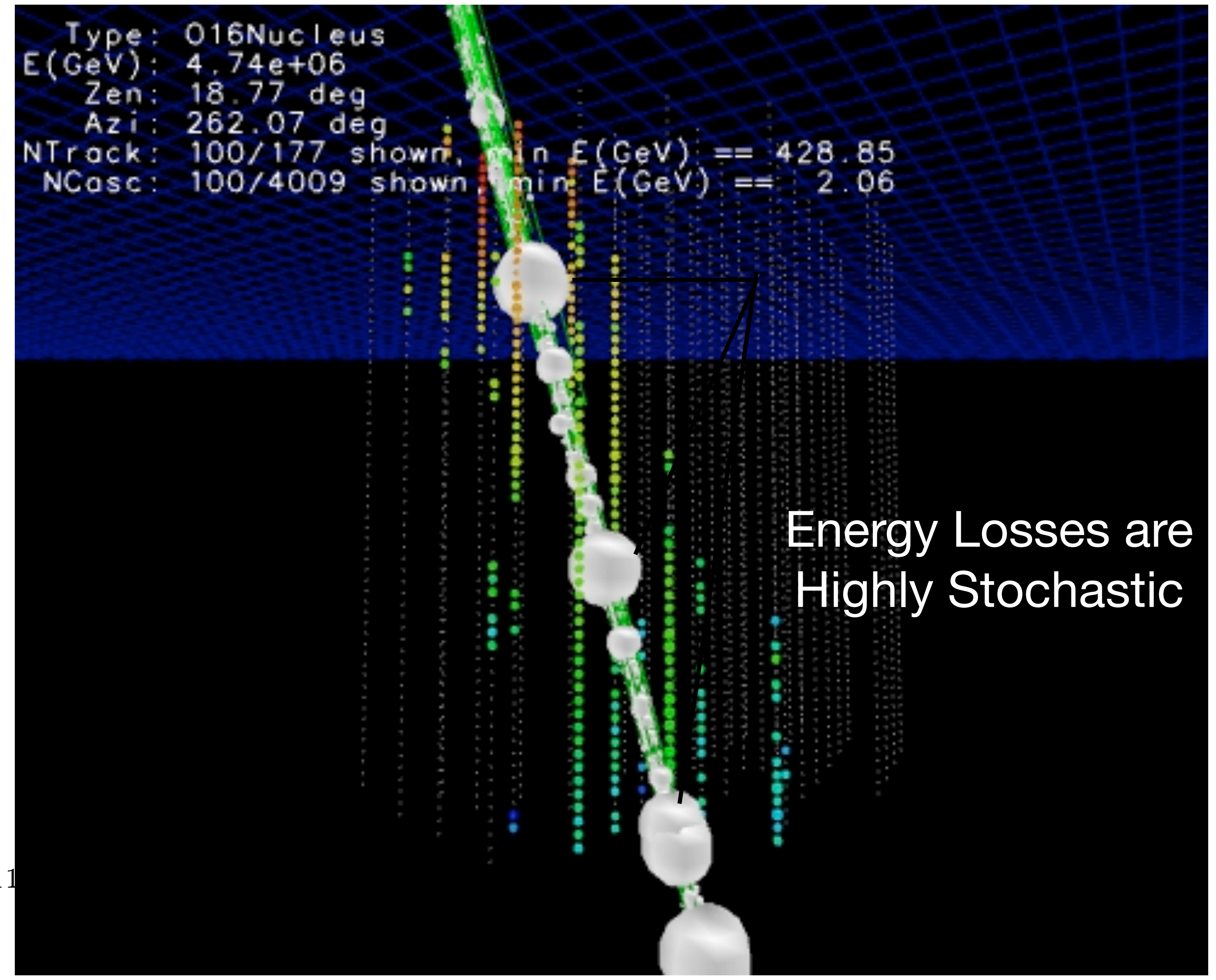
Old Simulation Chain



PROPOSAL

- "PROPOSAL" is the name of a software package not an actual proposal
- Propagates muons as they travel through ice
- Muons energy loss is also highly stochastic and non-parallelizable
- All muons will lose a significant fraction of their energy before making it to the deep ice detector
- Only muons with very high energy will make it to the detector





Old Simulation Chain

CORSIKA Executable

CORSIKA

CPU

Air shower simulation

CORSIKA BINARY FORMAT →

IceTray - IceCube's processing framework

PROPOSAL

CPU

Propagate Muons

I3MCTree →

OpenCL

GPU

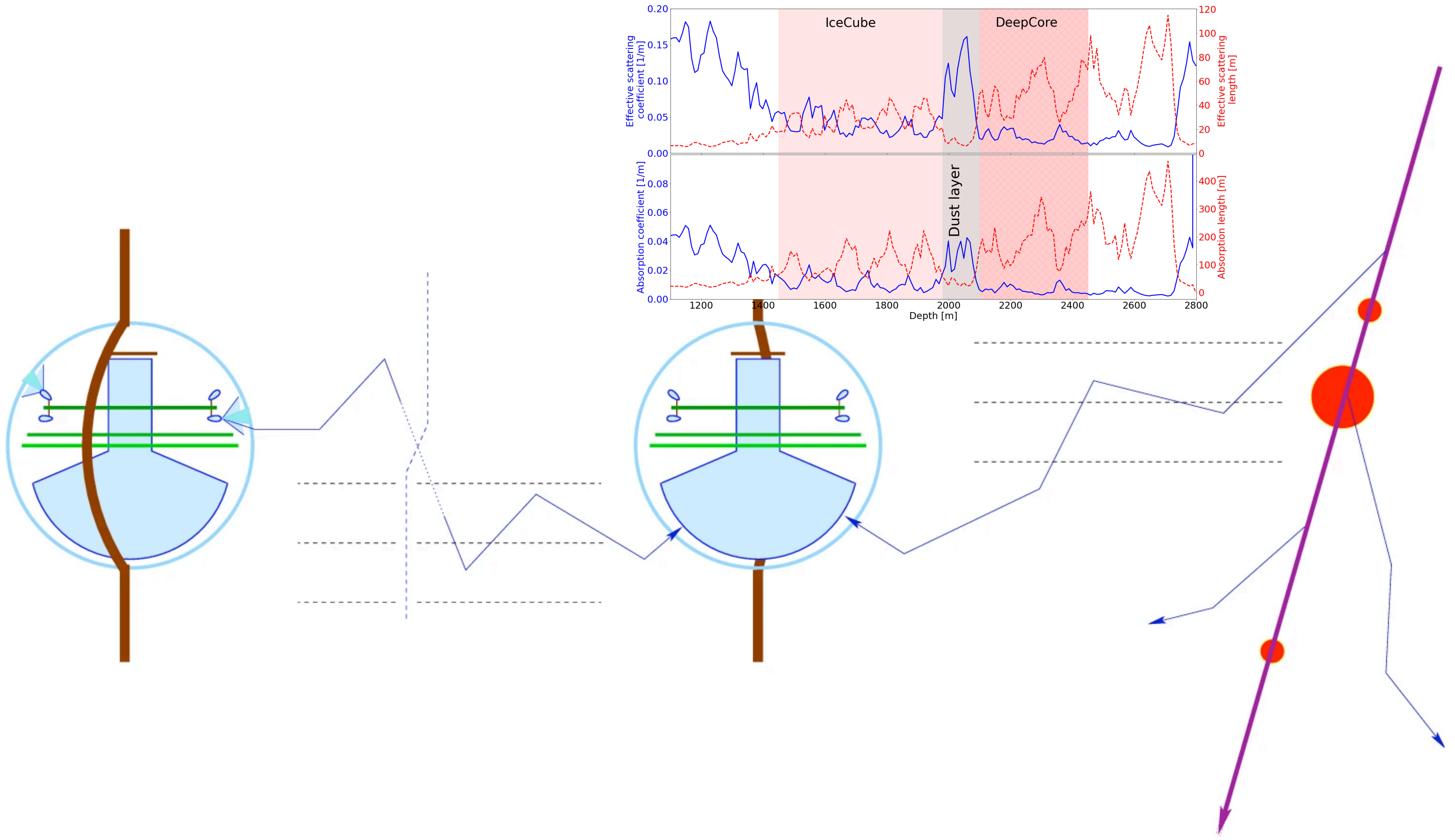
Propagate Photons

I3Photon →

Detector Simulation

CPU

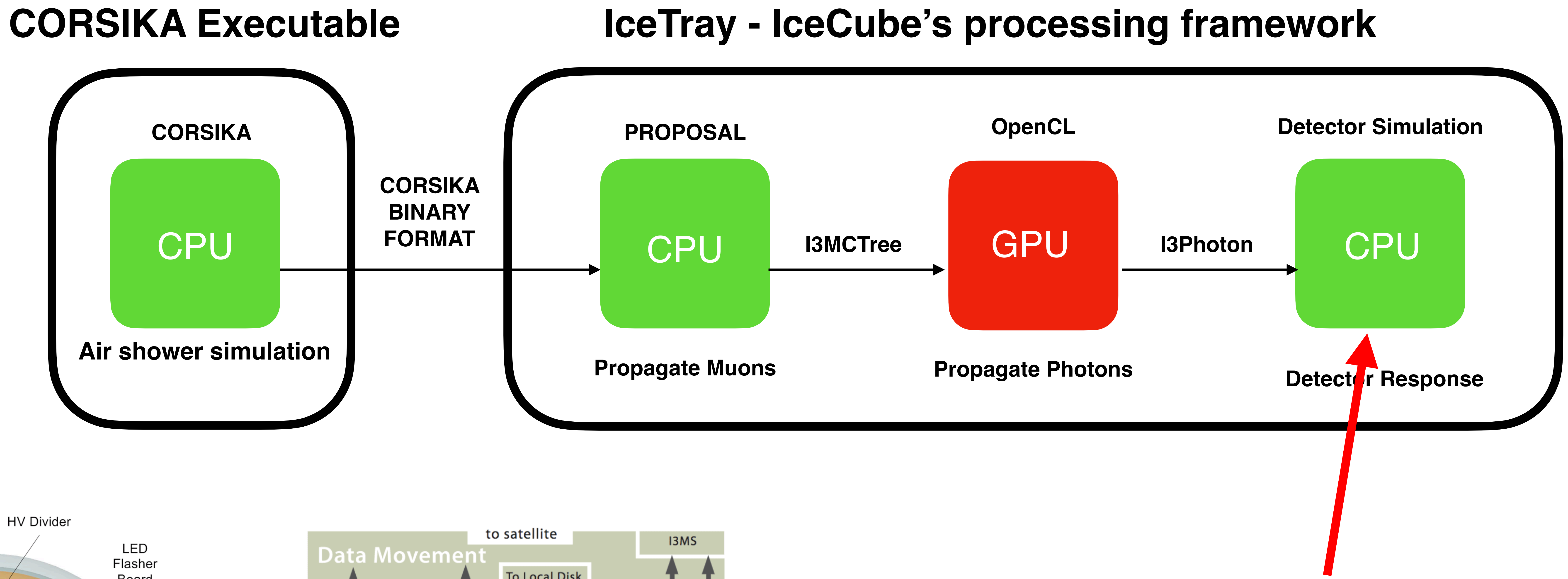
Detector Response

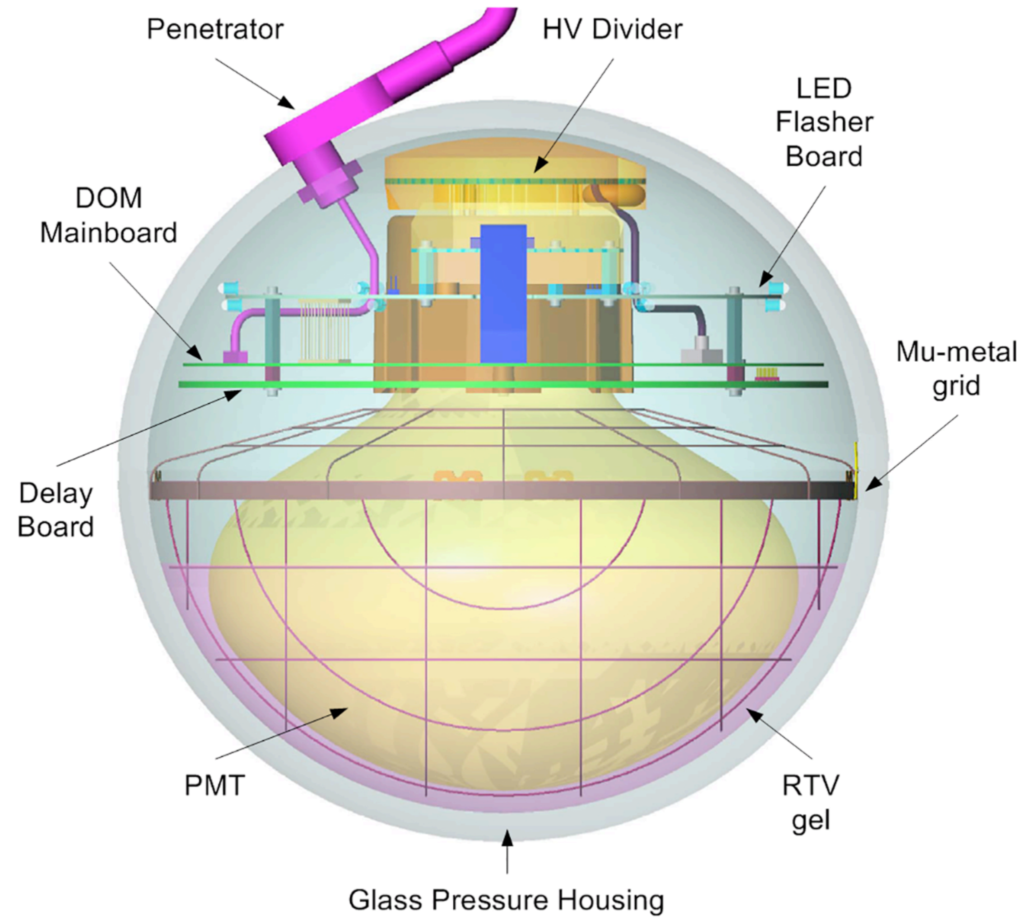


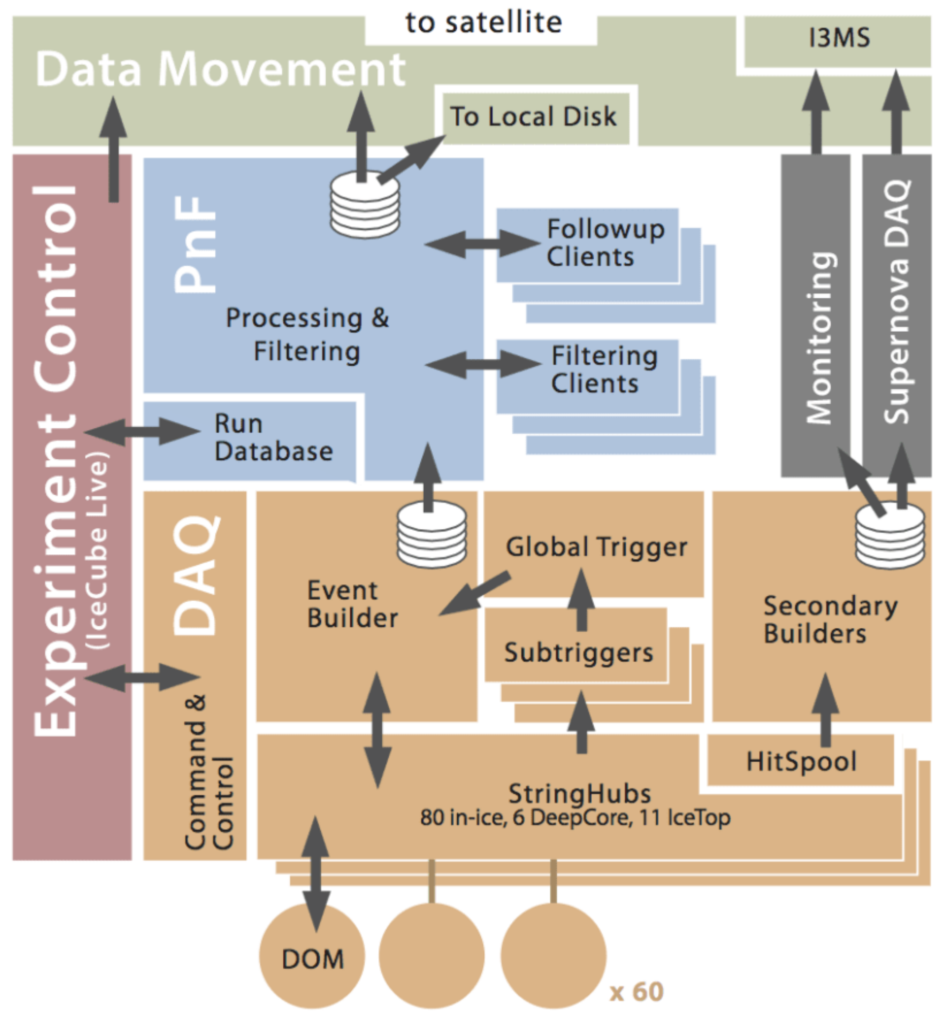
Photon Propagation

- Light emitted by muons is propagated through the Ice
- Layers of different concentrations of dust cause the scattering and absorption of the ice to be different at different heights
- Photon propagation is highly parallelizable and can only be effectively performed on GPUs with OpenCL/CUDA

Old Simulation Chain





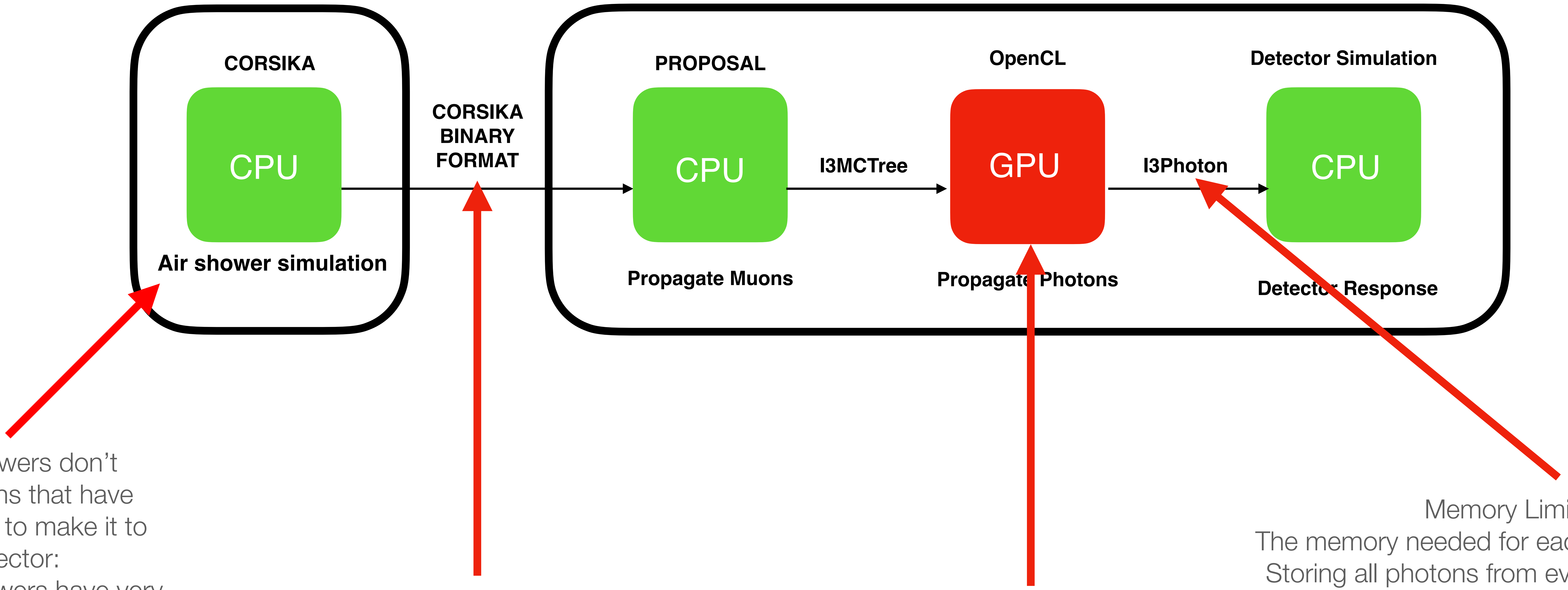


Detector Response/Reconstruction

- Simulates DOM Photomultiplier tube and front-end electronics
- Simulates noise
- Simulates the data acquisition system and trigger
- Performs all reconstructions the same as if it were experimental data

Problems with Old Simulation Chain

CORSIKA Executable **IceTray - IceCube's processing framework**



Most air showers don't produce muons that have enough energy to make it to the detector:
Low energy showers have very small chance of producing a high energy muon
Inclined showers have a much higher threshold to make it to the detector

IO Bottleneck
CORSIKA needs to write all the events to a file
Almost all air shower muons don't have sufficient energy to reach the IceCube active volume

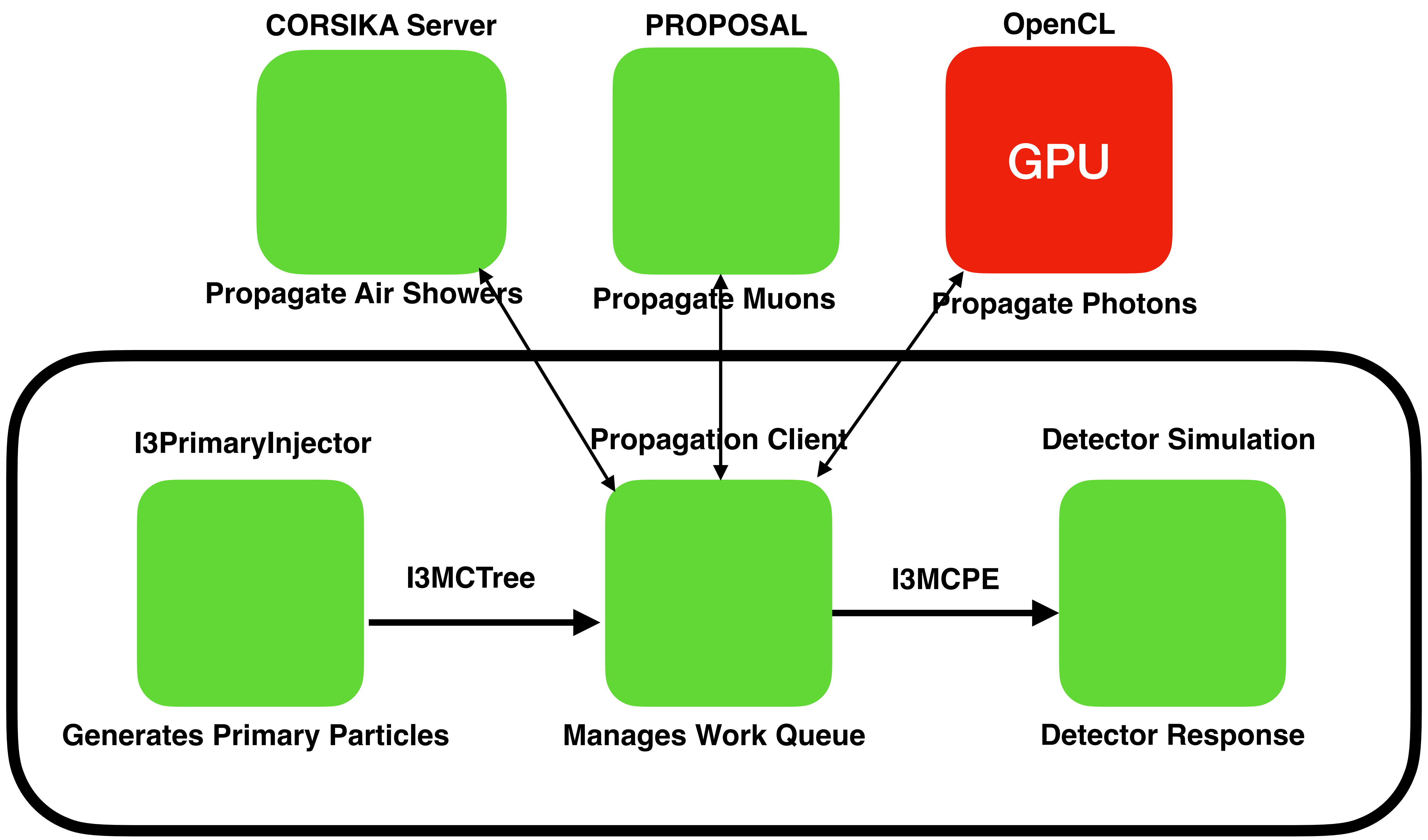
Inefficient use of resources
GPU is underutilized because the previous stages do not produce results fast enough to keep the GPU busy

Memory Limits Exceeded
The memory needed for each event is highly stochastic
Storing all photons from event results in brief memory allocations which exceed this limit and kill the job

Memory per CPU core is expected to decrease in computing environments in the future

ICECUBE NEUTRINO OBSERVATORY

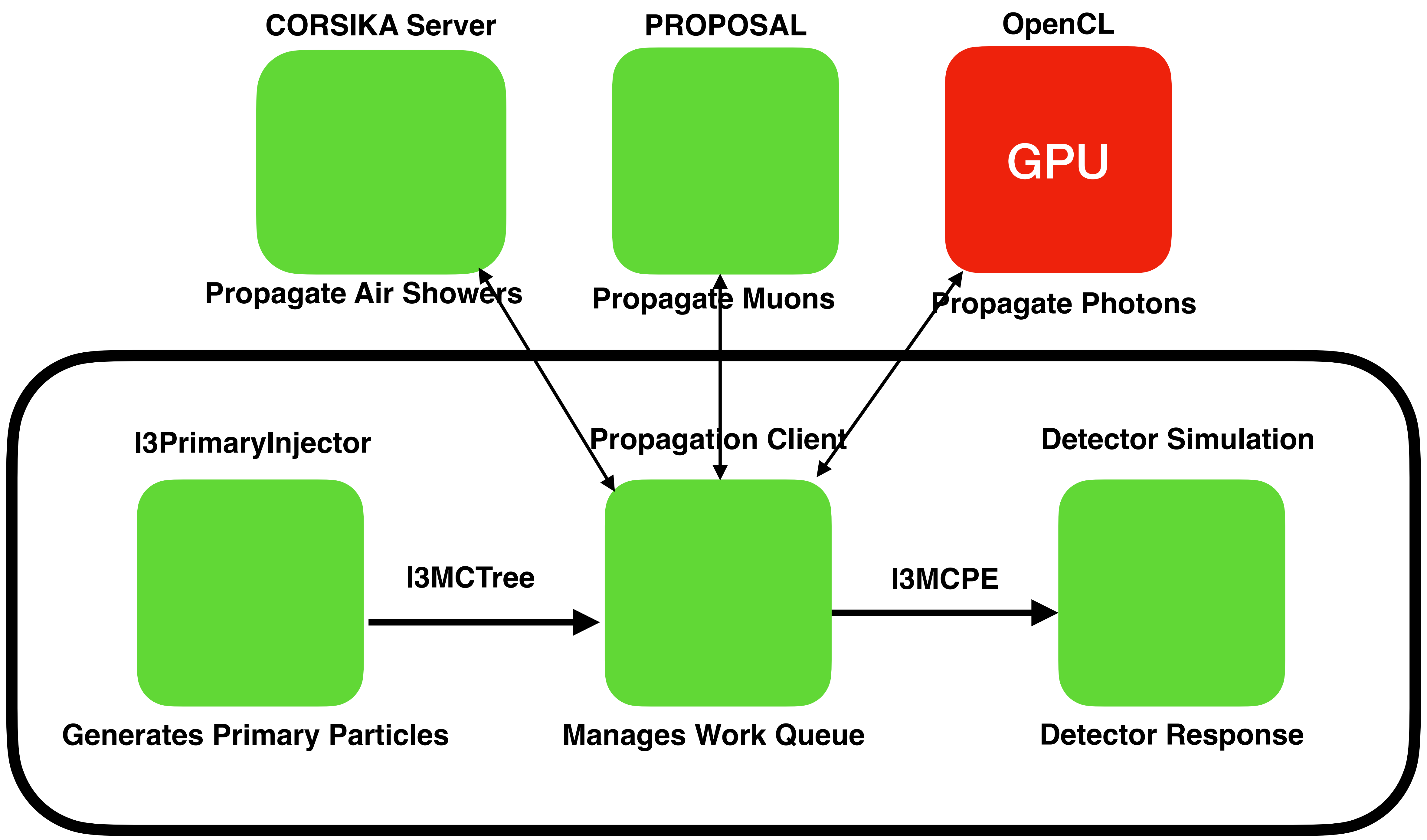
New Simulation Chain



8 Instances of IceTray are used with each GPU server

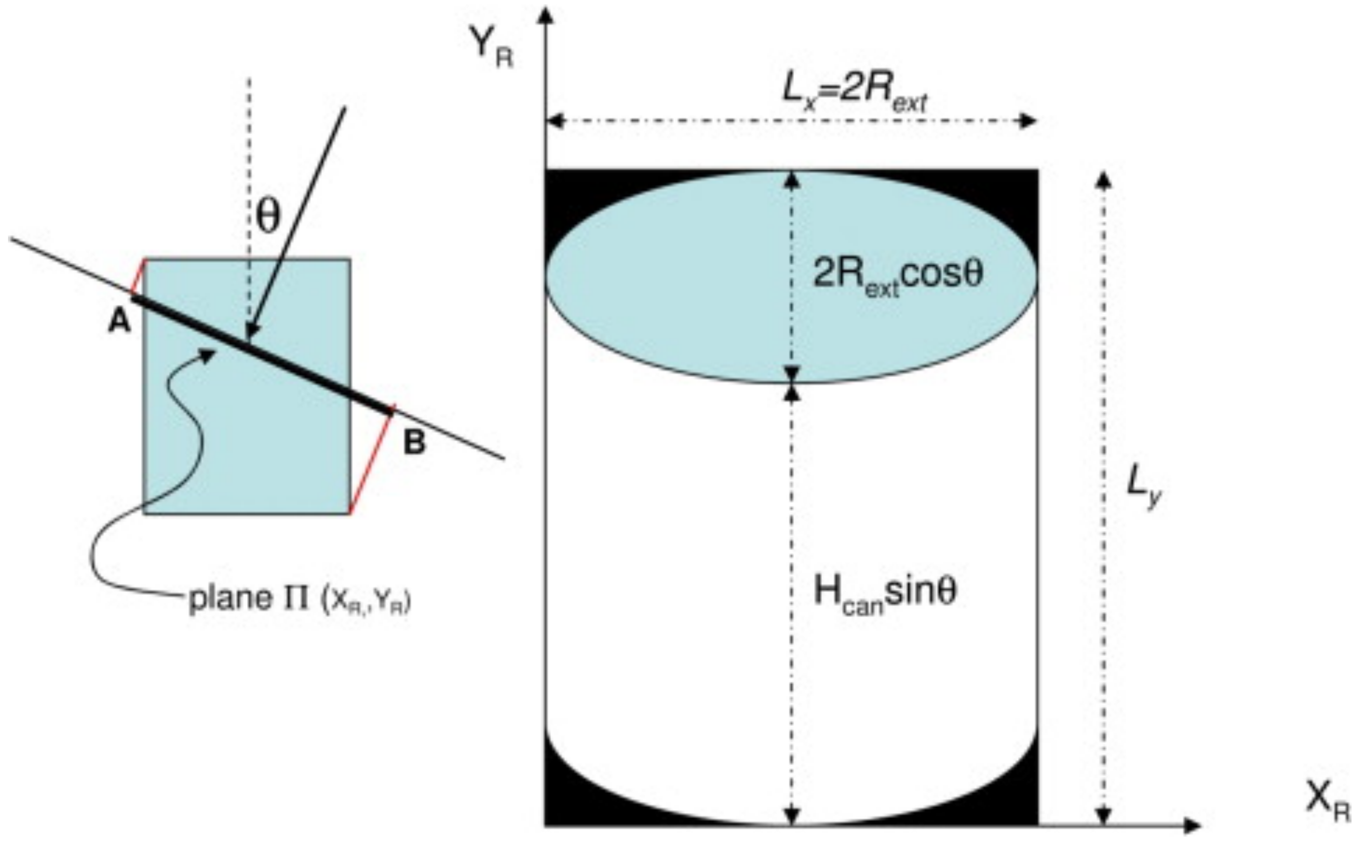
- Move to a Client/Server model for Air Shower, Muon, and Photon propagators
  - Each client has a work queue can span multiple events
- 1 instance of CORSIKA Server for air showers propagation
- 1 instance OpenCL Server collects work units and sends them to the GPU
- 8 instances of PROPOSAL for muon propagation
- 8 instances of the simulation chain are used in parallel to connect to the propagation servers
  - Keeps the GPU fully utilized

New Simulation Chain



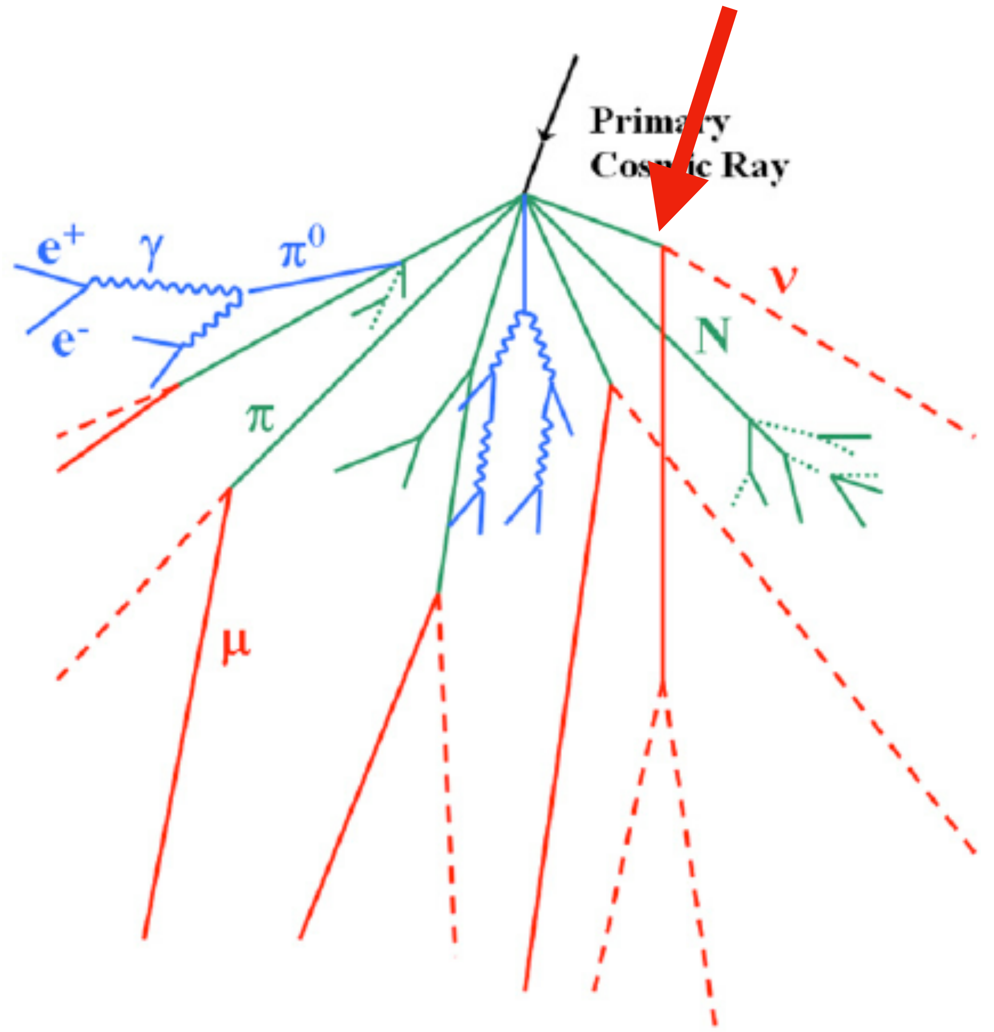
I3PrimaryInjector

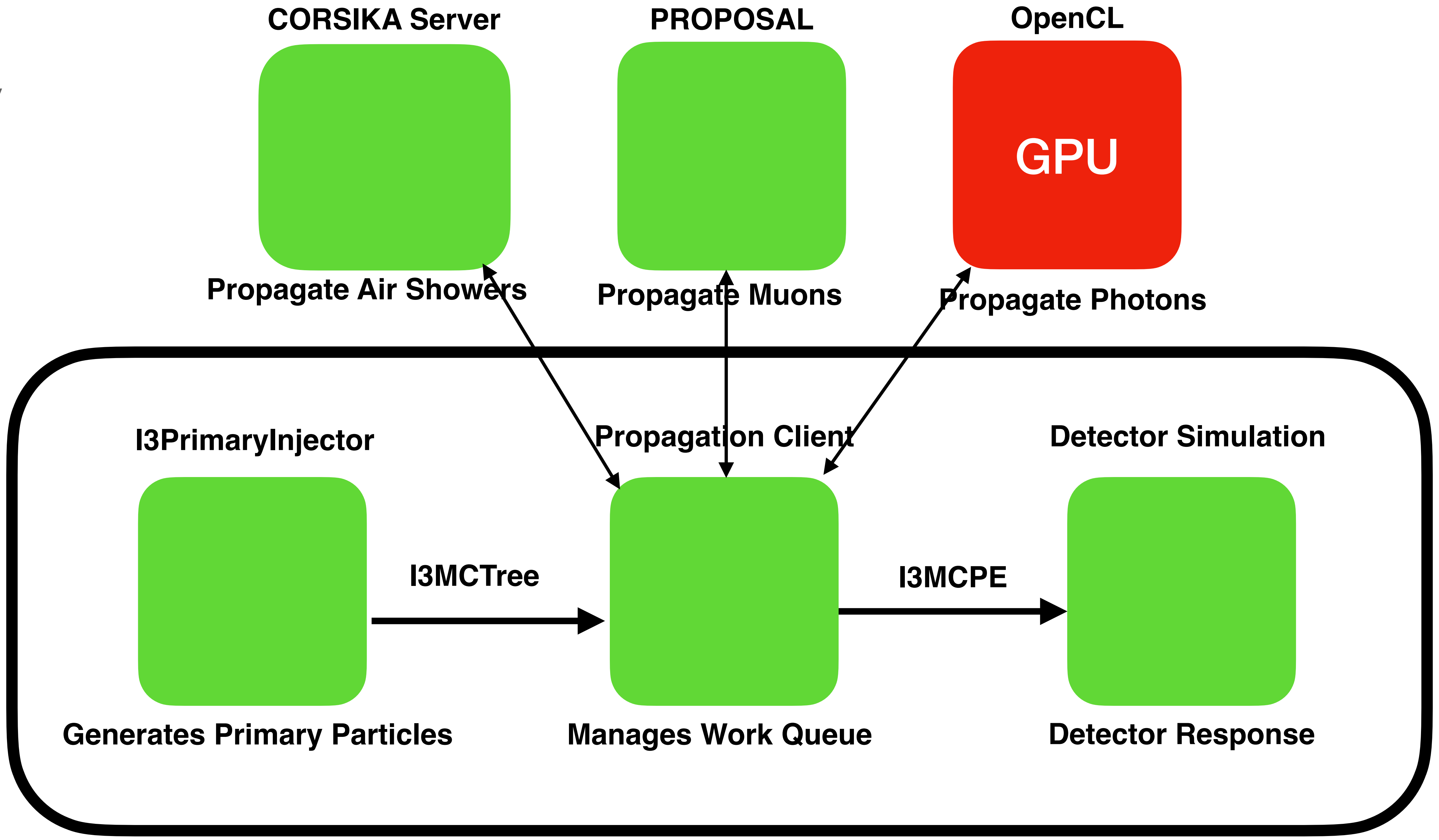
- Primaries are now sampled from new IceTray module rather than hard coded functions in CORSIKA
- Primaries are generated on directly on the detector surface with arbitrary energy and spatial distributions
- Allows for future schemes for oversampling regions that produce more background

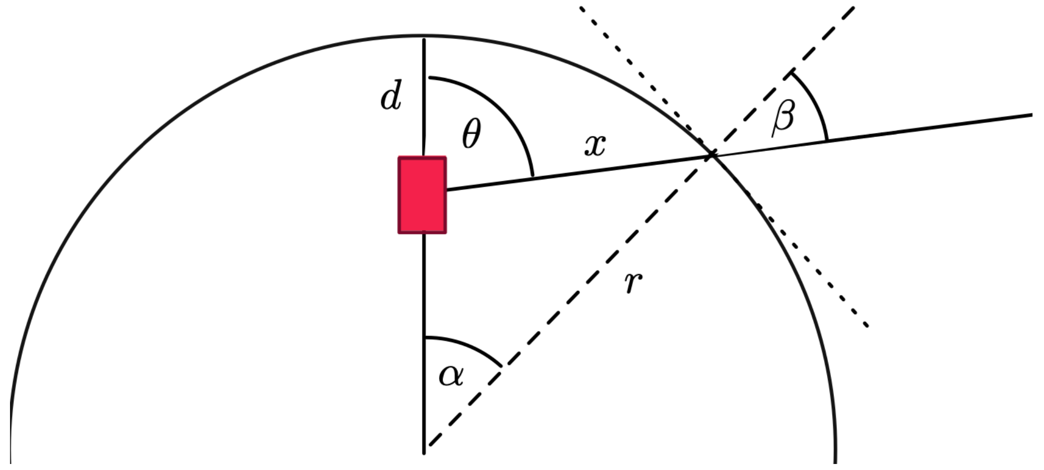


New Simulation Chain

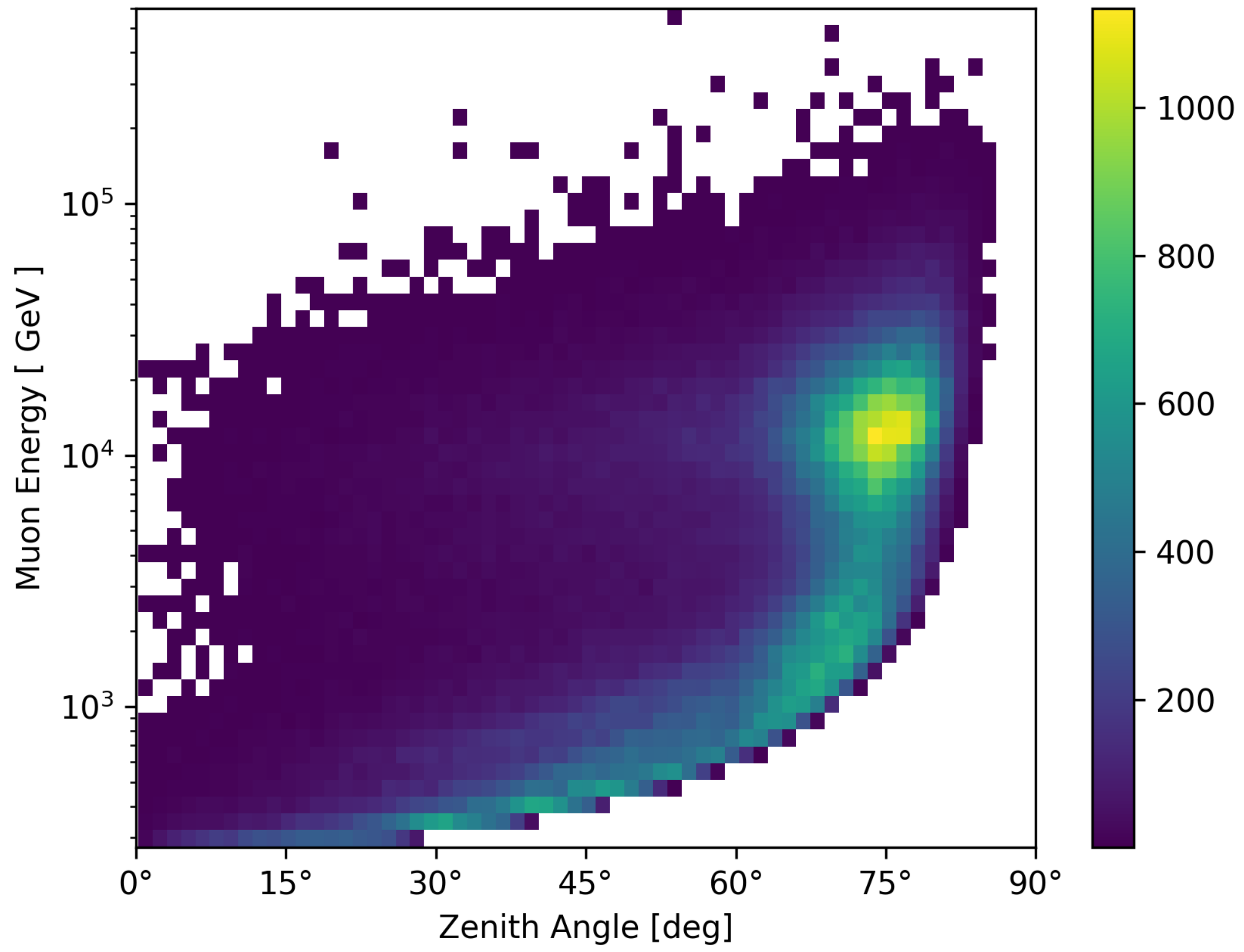
Leading Edge Muon is most likely to propagate all the way to the detector







More inclined showers encounter more matter between the surface and the detector and require higher energy muons to reach the detector

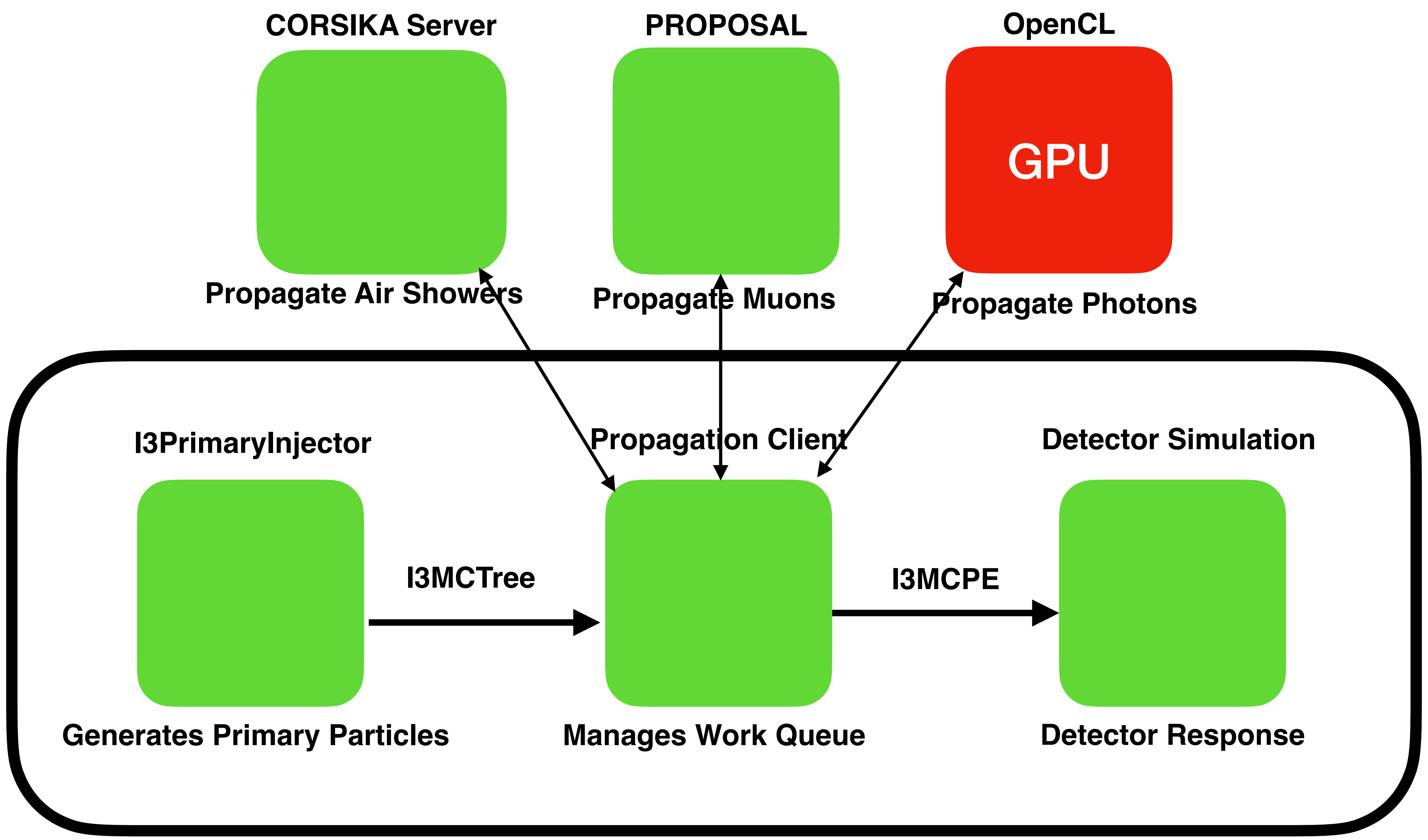


New CORSIKA Server will set the minimum muon energy higher for more inclined showers

CORSIKA Server

- CORSIKA was modified to run as a server which can be configured with individual primaries (thanks to D. Baack)
- Different CORSIKA cards for different showers
    - The energy needed to reach the detector is highly dependent on the inclination of the shower
    - The energy at which CORSIKA will stop propagating particles is now higher for inclined showers
- Showers with higher energy leading edge muons are undersampled:
    - Showers with low energy muons are killed before the rest of the shower is calculated
- Since individual particles are sent over IPC no files are written avoiding IO bottleneck

New Simulation Chain

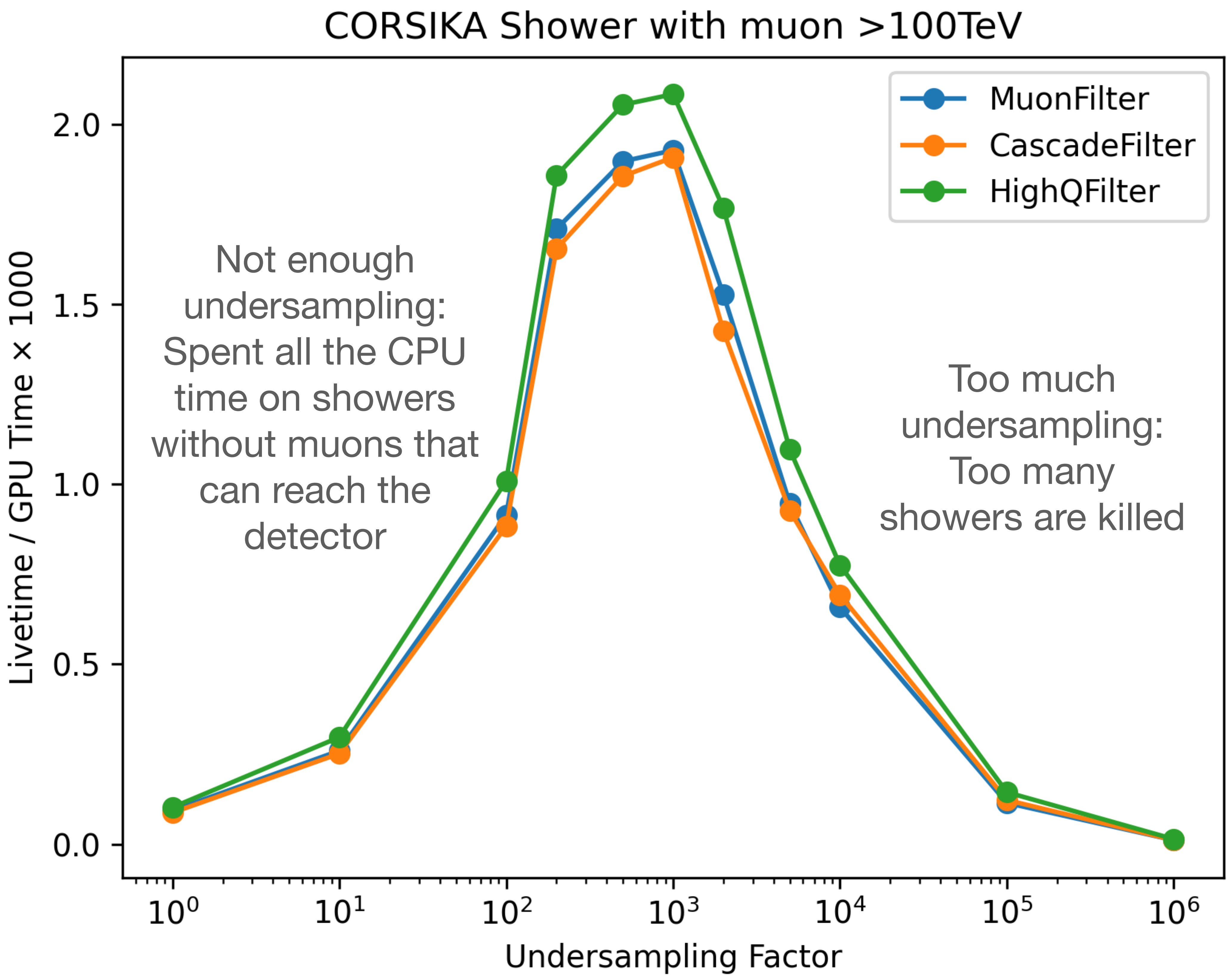


Memory Resources

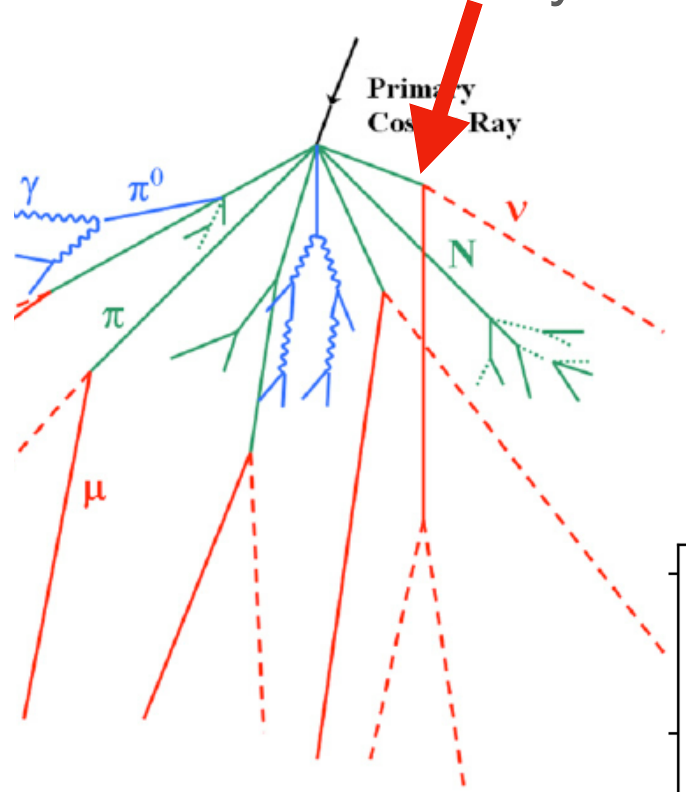
- Memory is saved with more efficient dataclass for storing photons: I3MCPE
- The Propagation Client's work queue doesn't have to store intermediate data for the entire event reducing memory usage
- The memory is amortized across the 8 instances of icetray
  - 8 instance sharing 8GB are much less likely to be taken down by a memory spike than 8 jobs with 1GB each

Optimizing Leading Muon Undersampling

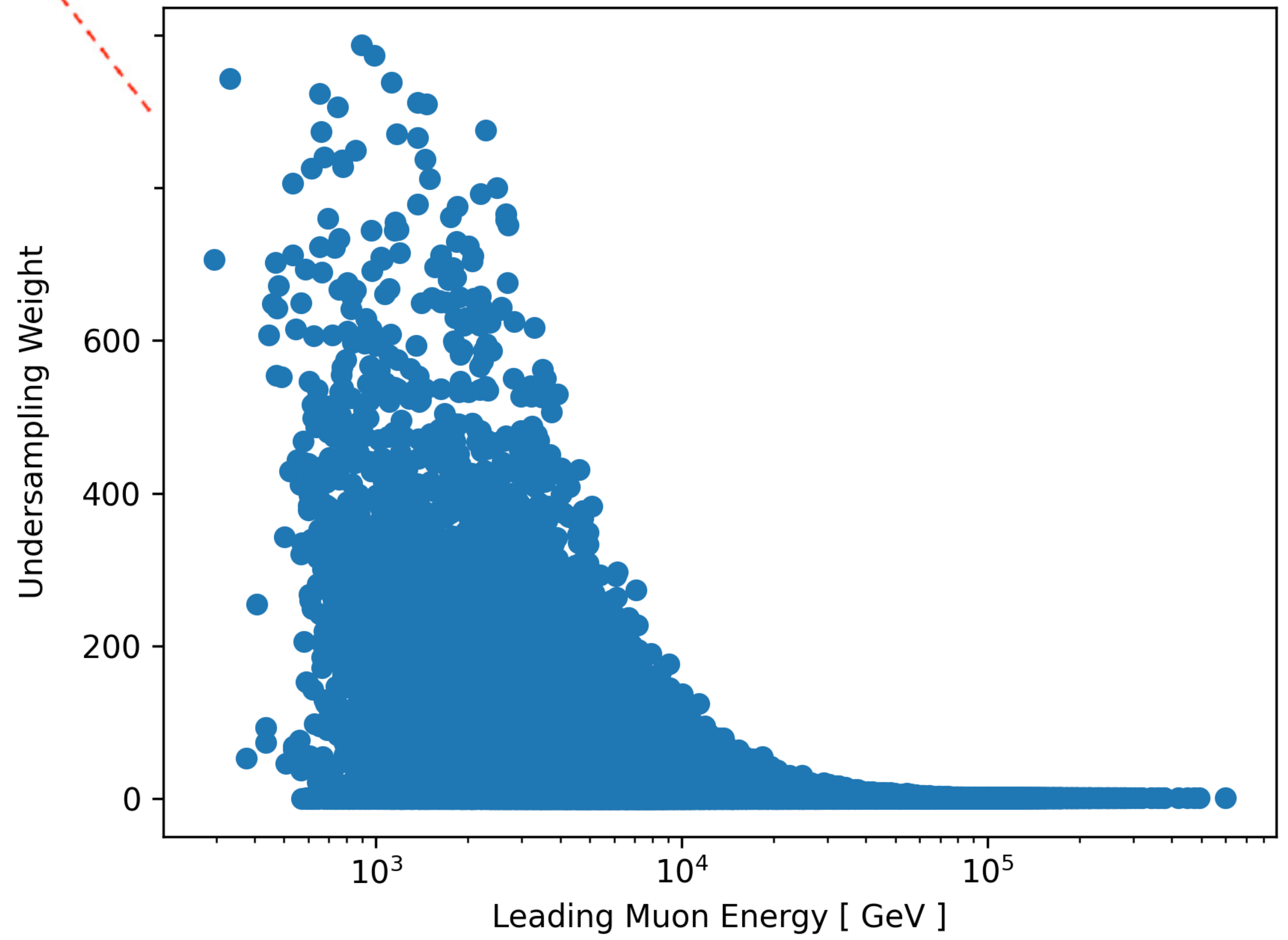
Air showers are undersampled based on the energy of the first muon to appear in the shower



Leading Edge Muon is most likely to propagate all the way to the detector



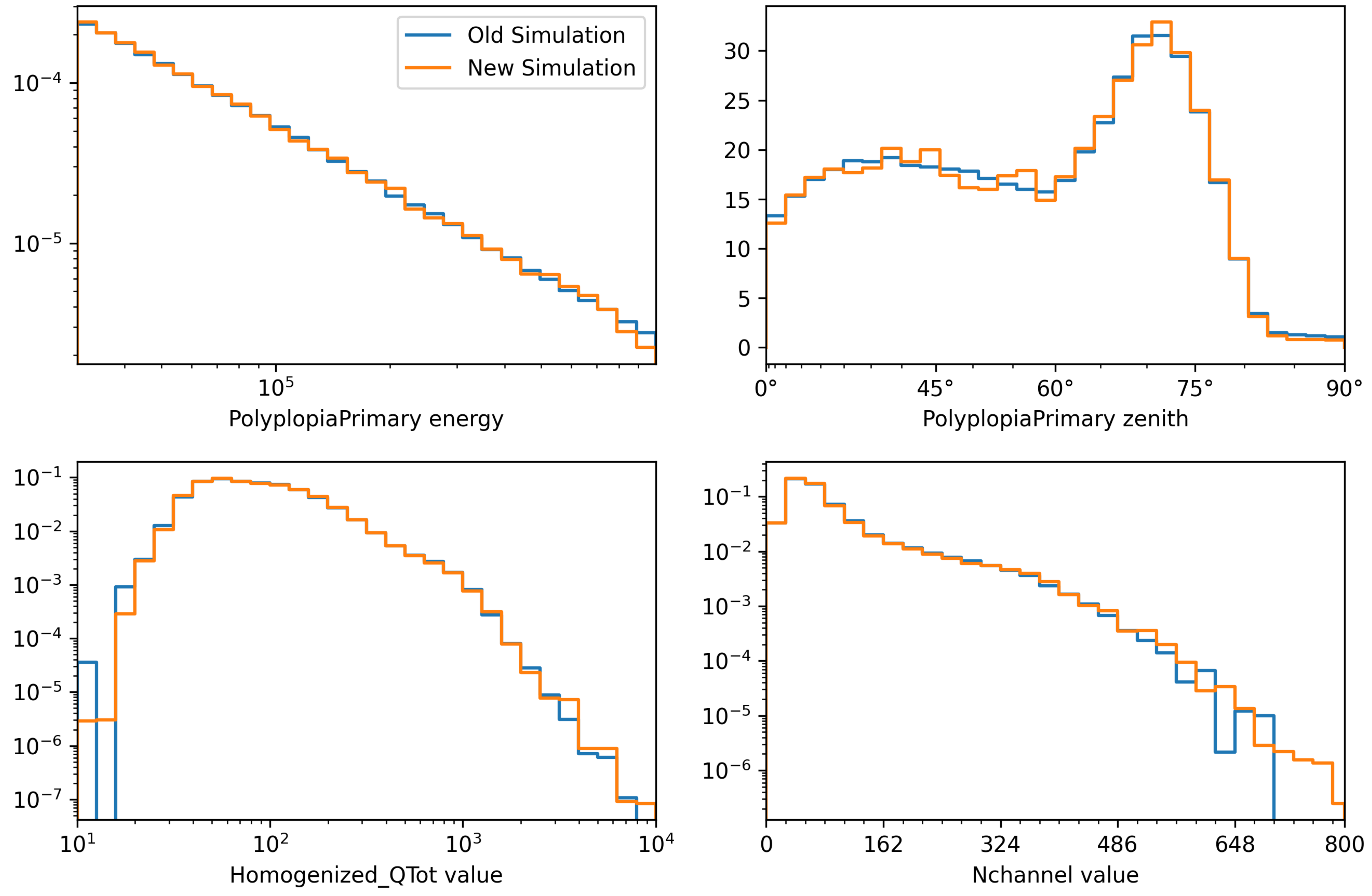
Showers with leading edge muons below 1TeV are undersampled by a factor of 10^3
Showers with Muons above 100TeV are not undersampled



Undersampling also take primary energy and zenith into consideration

This undersampling based on leading muon energy results in a 20x increase in the simulated lifetime per GPU time

Comparison Between New and Old Simulation



Good agreement is seen between old and new

Performance

	Old Simulation Chain	New Simulation Chain
Time Per Shower	101 μs	3.70 μs
Memory/Core	7GB	1GB
Walltime of Completed Jobs/ Total WallTime	60%	84%
GPU utilization	45%	80%

Questions