

Intro to IceCube Simulations

Juan Carlos Díaz-Vélez

IceCube Summer School

Madison, WI

June 6, 2025

IceCube

50 m

IceTo

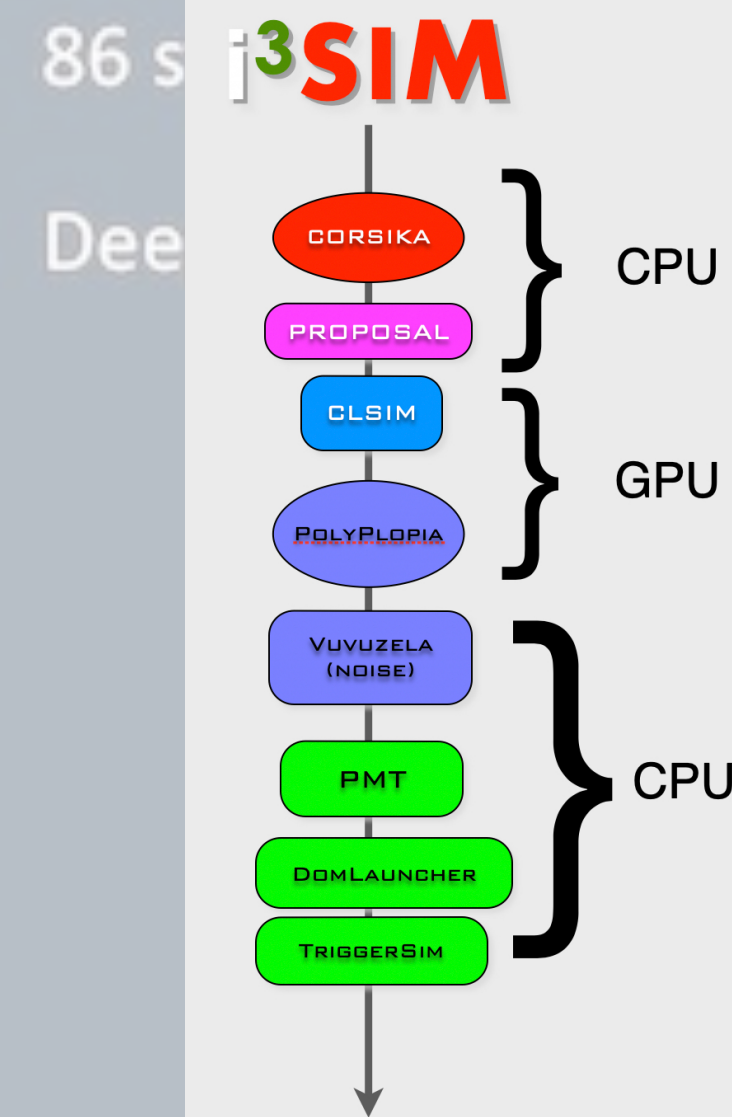
1450 m

IceCube Laboratory

Data from every sensor is
collected here and sent by
satellite to the IceCube data
warehouse at UW-Madison

Digital Optical
Module (DOM)
5,160 DOMs deployed
in the ice

Amundsen–Scott South
Pole Station, Antarctica
A National Science Foundation-
managed research facility



Monte Carlo Simulations

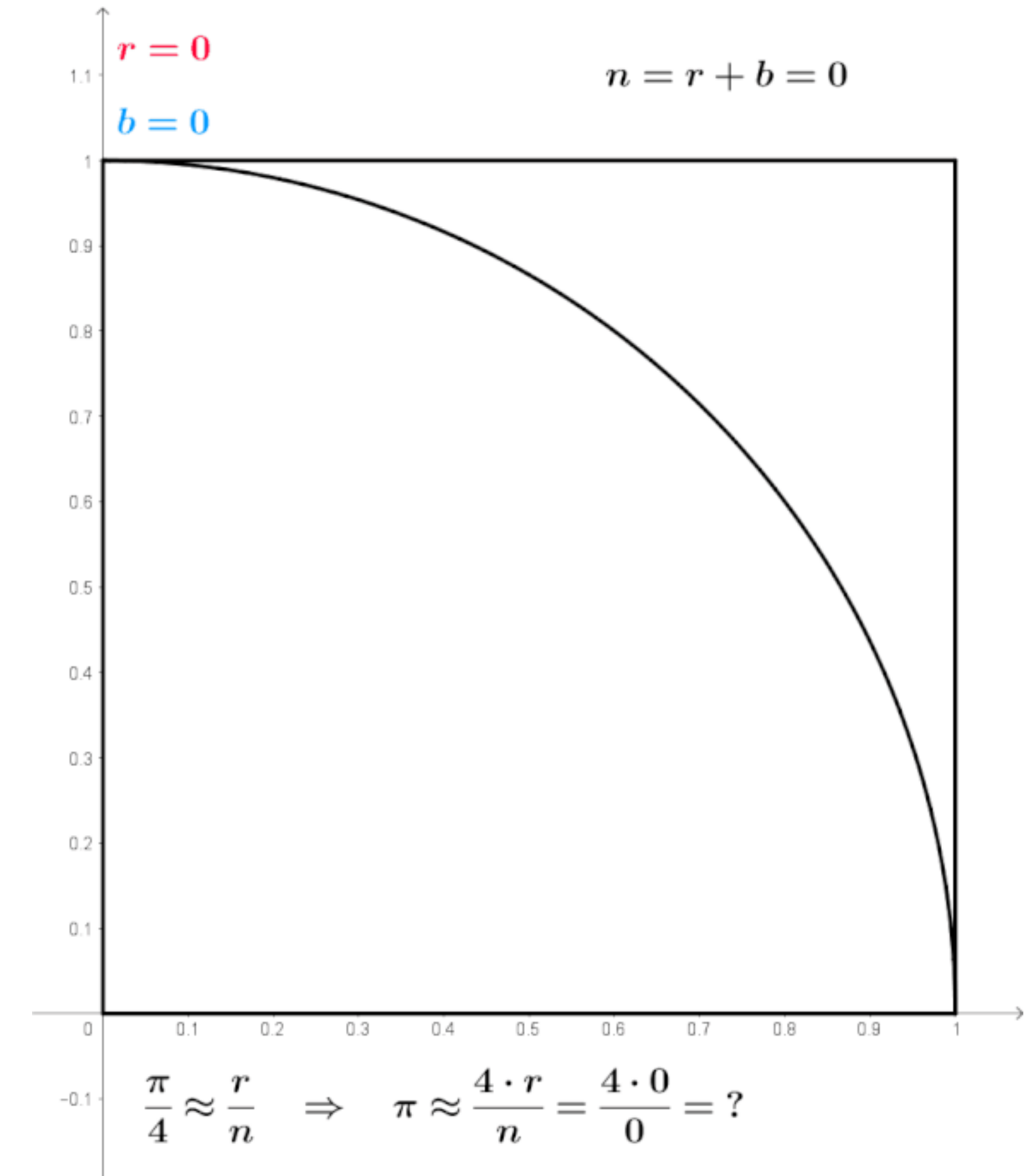
Monte Carlo methods vary, but tend to follow a particular pattern:

1. Define a domain of possible inputs
2. Generate inputs randomly from a probability distribution over the domain
3. Perform a deterministic computation of the outputs
4. Aggregate the results

Example: Monte Carlo method applied to approximating the value of π

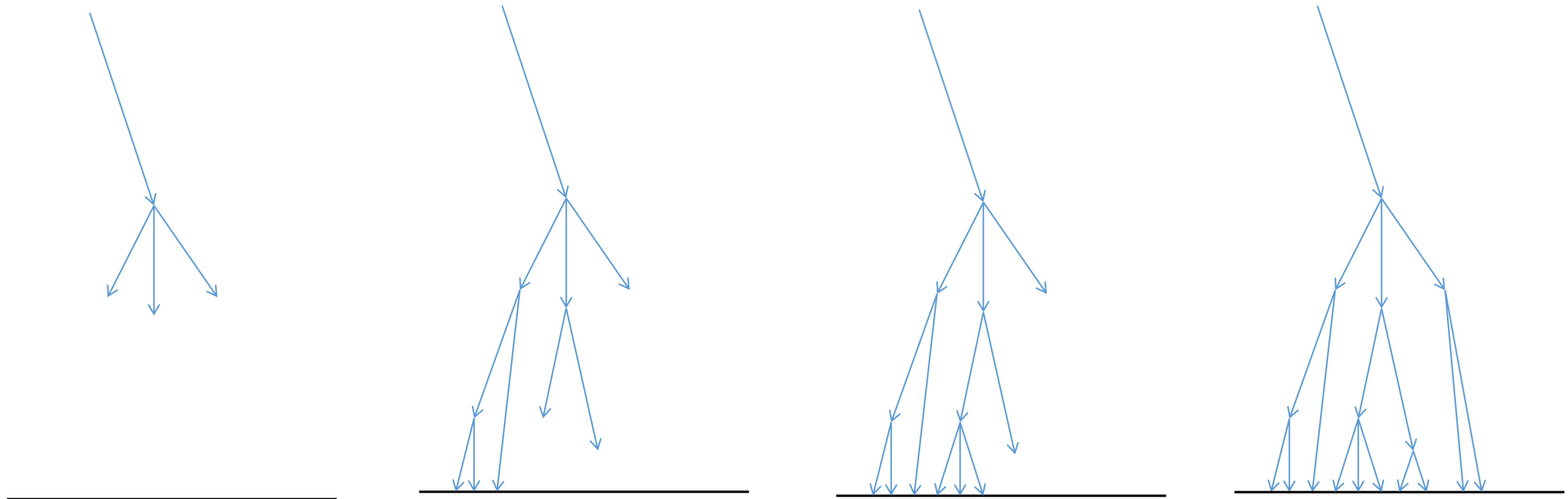
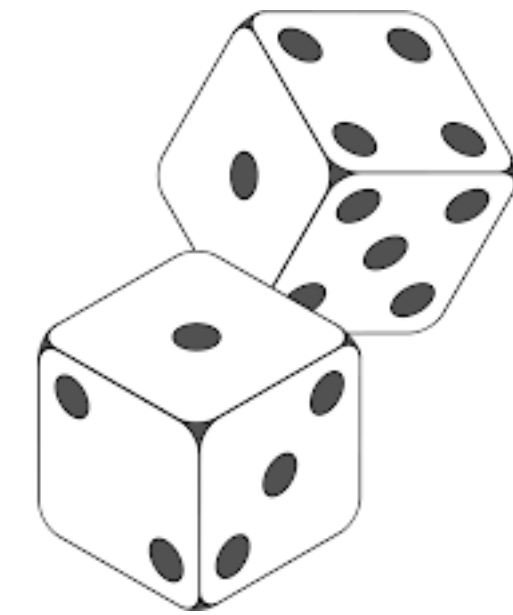
Consider a quadrant (circular sector) inscribed in a unit square. Given that the ratio of their areas is $\pi/4$, the value of π can be approximated using a Monte Carlo method:

1. Draw a square, then inscribe a quadrant within it
2. Uniformly scatter a given number of points over the square
3. Count the number of points inside the quadrant, i.e. having a distance from the origin of less than 1
4. The ratio of the inside-count and the total-sample-count is an estimate of the ratio of the two areas, $\pi/4$. Multiply the result by 4 to estimate π .

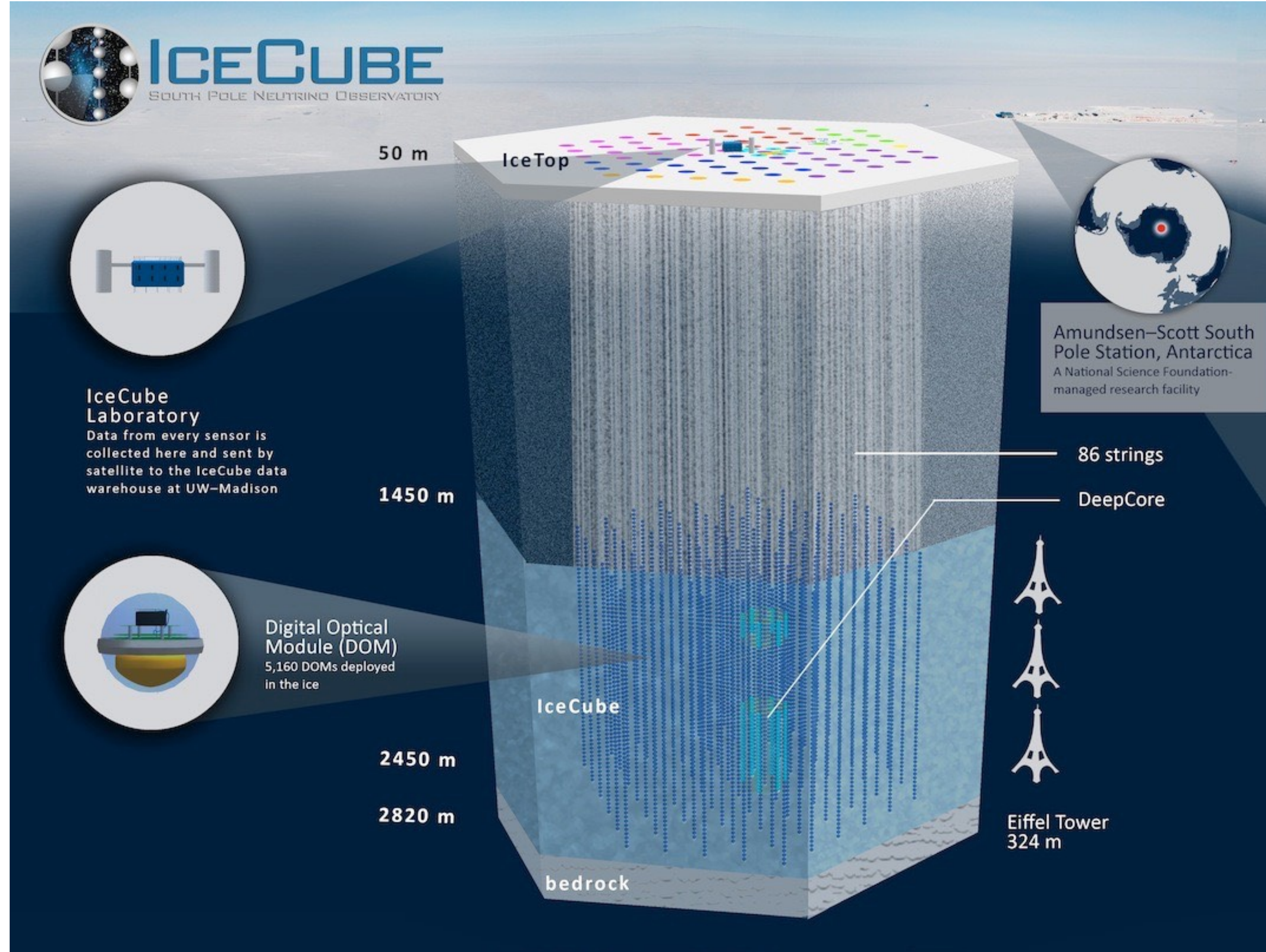


Monte Carlo Simulations: Cosmic-Ray Showers

- Propagate through step calculate probability of interaction from probability distribution
- Choose interaction from distribution
- Repeat process for each of the interaction (decay products)

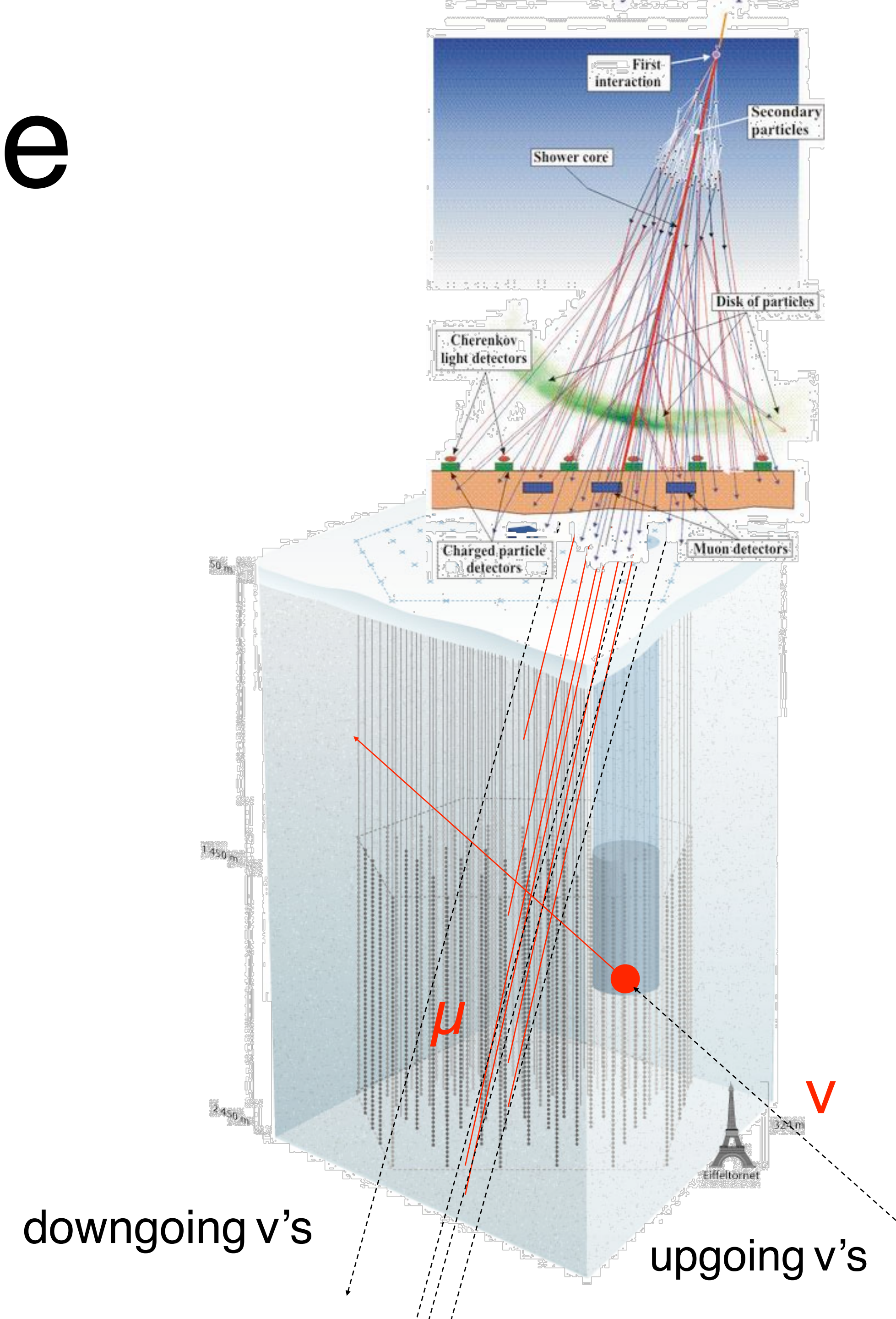


The IceCube Observatory



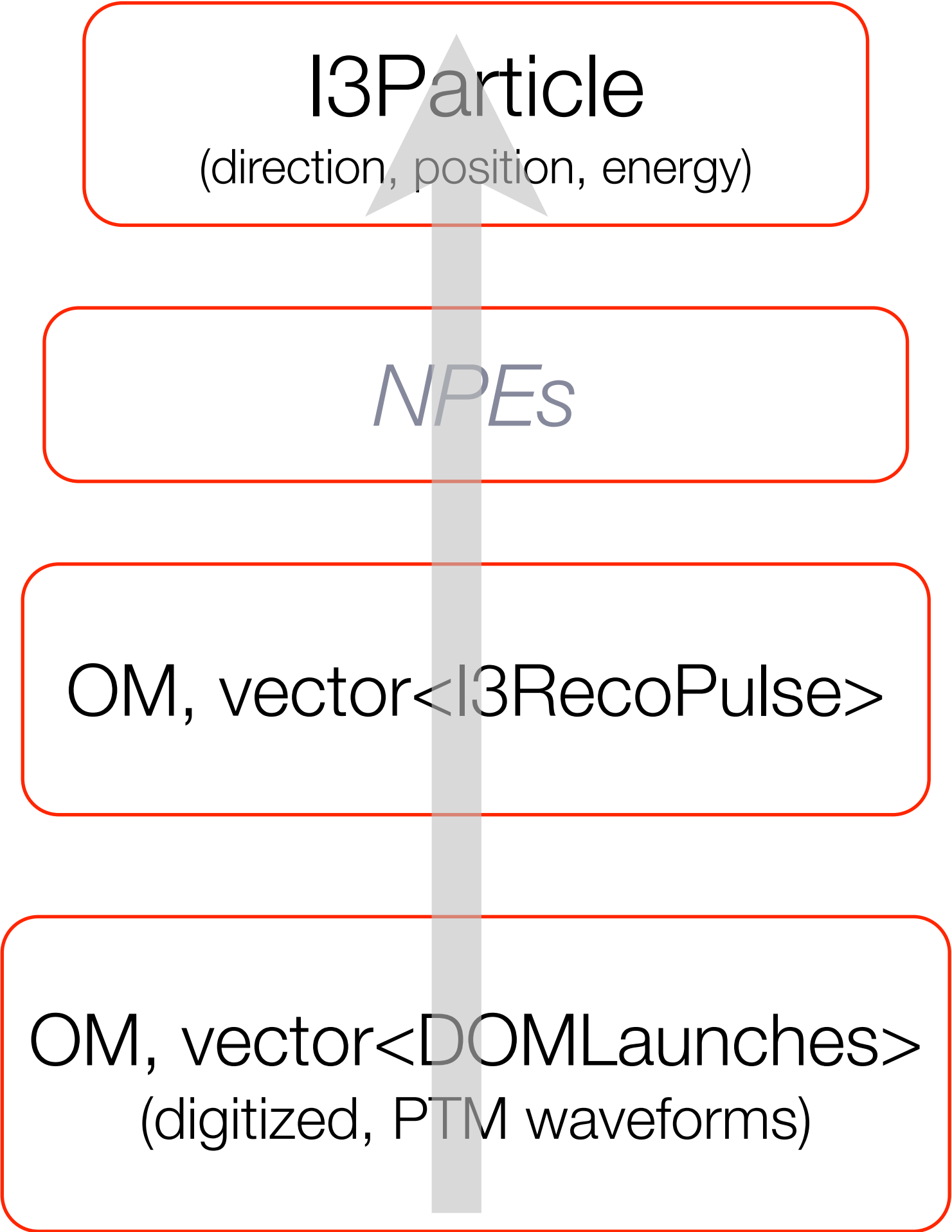
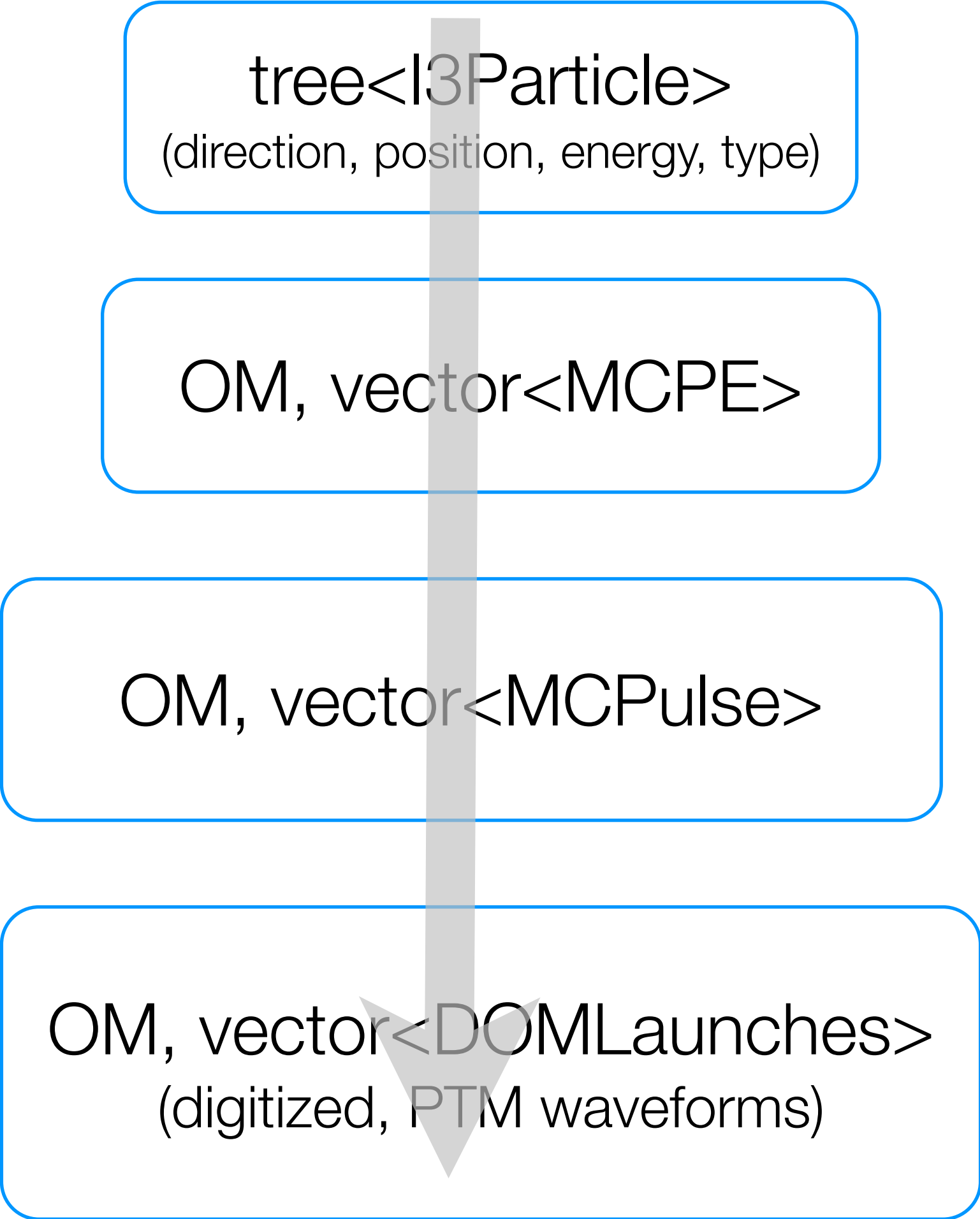
Events in icecube

- Air shower detection @ surface
- Penetrating muon detection in deep ice
- Events dominated by cosmic ray muons : $10^6 \mu$ for every ν that interacts in IceCube

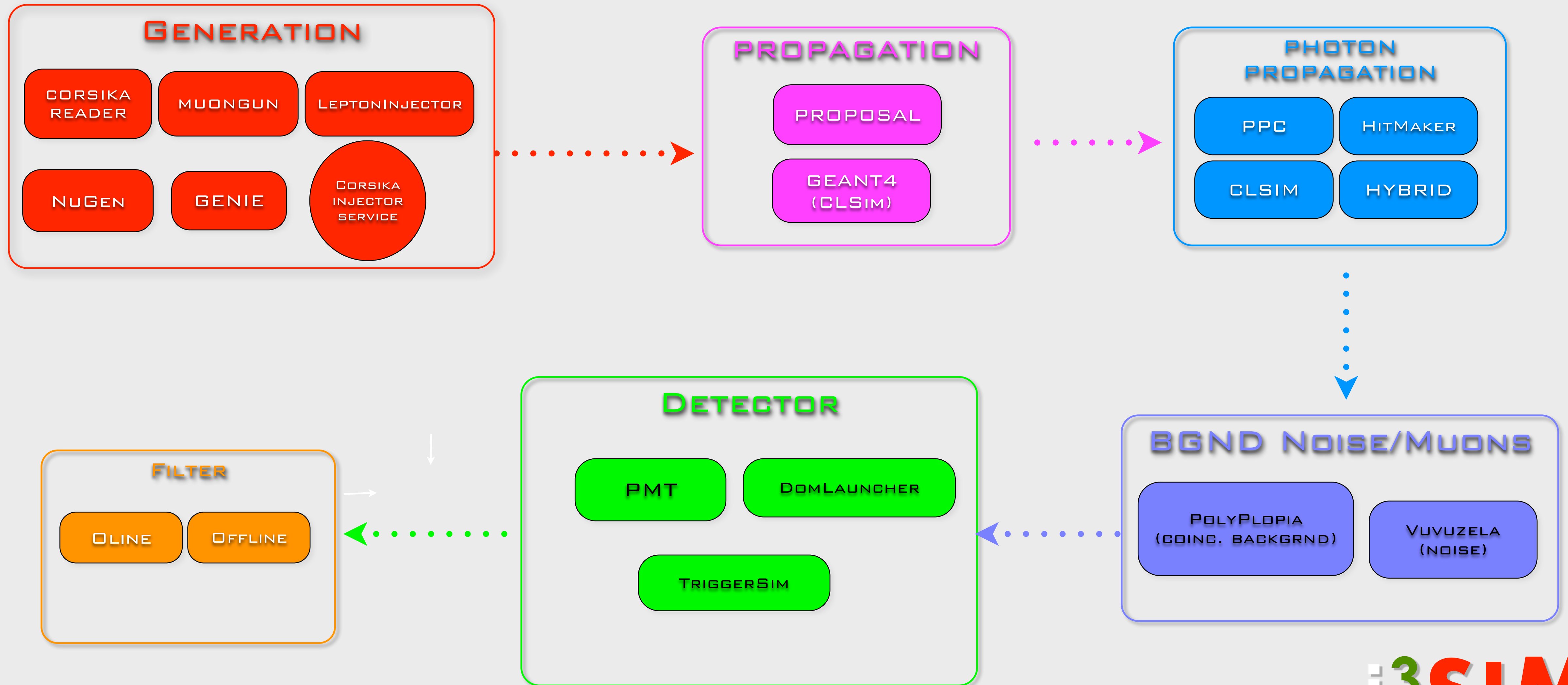


Simulation

Reconstruction



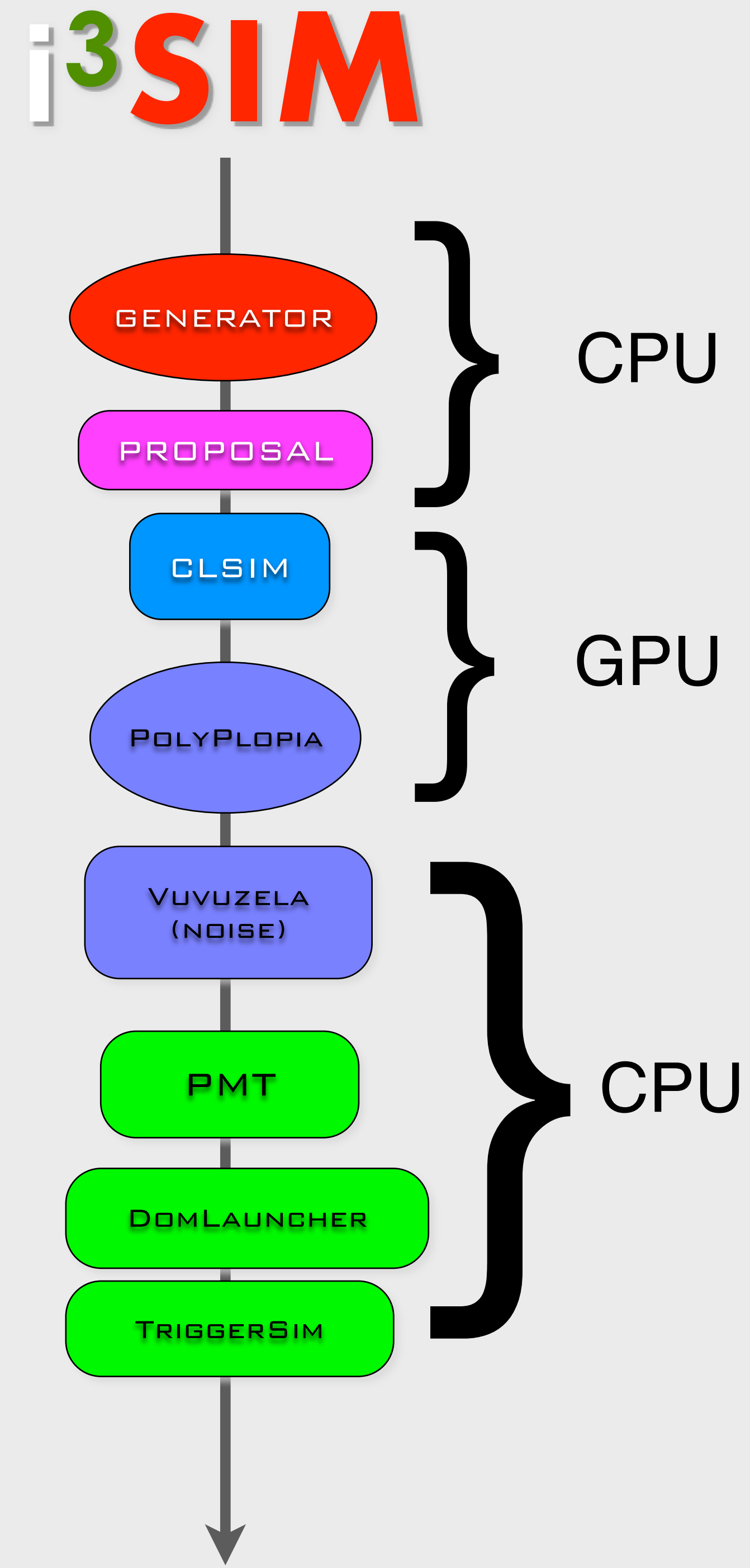
Simulation Chain



The Simulation Shish Kabob

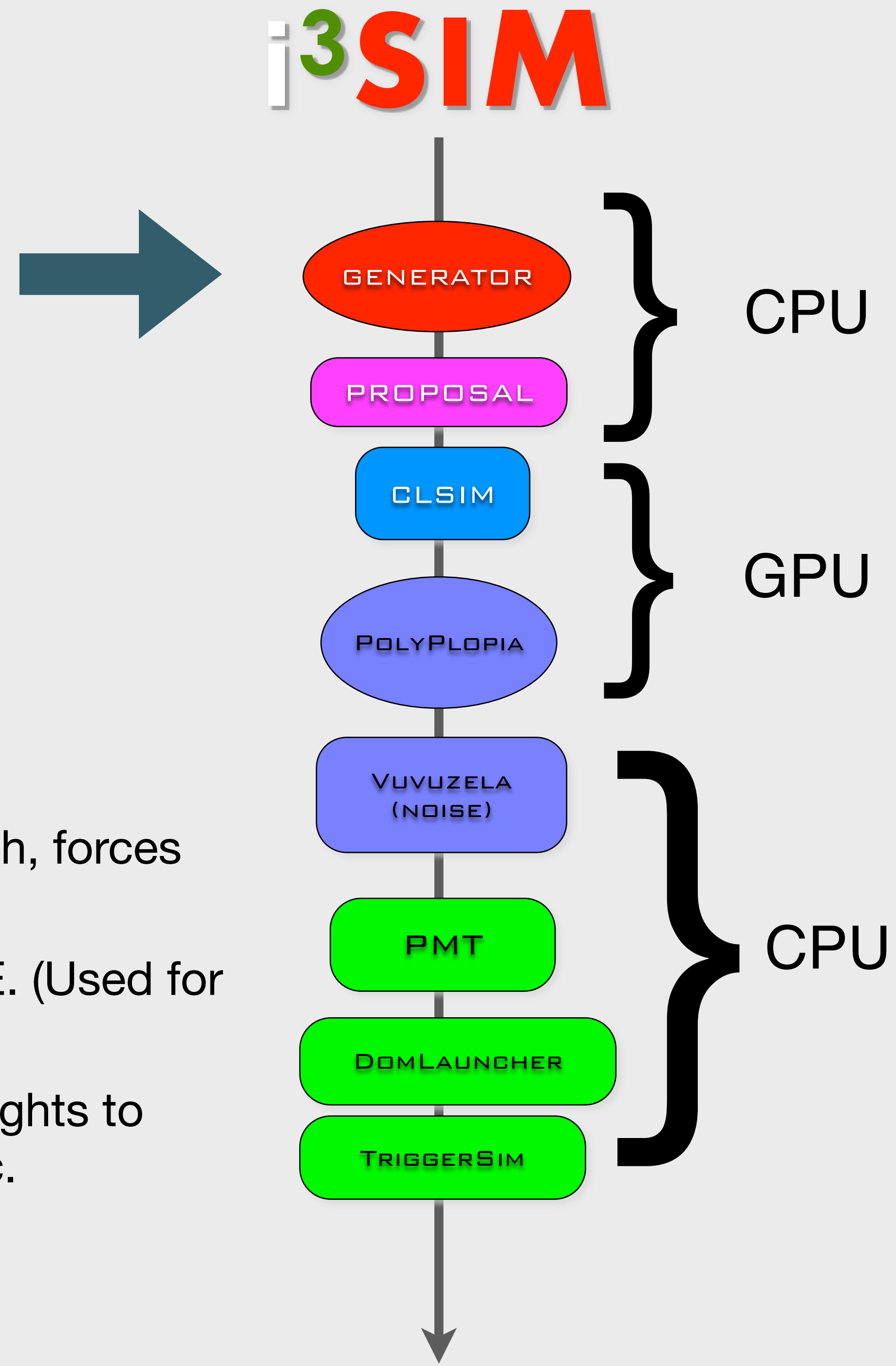
(Computing Resource Optimization)

- Optimizing the shish kabob:
 - Different parts of the simulation chain have different resource requirements.
 - CORSIKA is CPU-intensive and requires little RAM
 - Photon propagation run almost exclusively on GPUs
 - Detector simulation is CPU bound and requires more memory.
- Things to keep in mind:
 - Running whole chain on a GPU node will waste GPU resources and limit throughput.
 - Intermediate storage:
 - breaking up chain requires transferring/storing intermediate files.
 - Reduce complexity in workflow



Generators

- ▶ Cosmic-ray Air Showers:
 - ▶ **Triggered-CORSIKA** (parallel corsika client/server w clsim)
 - ▶ **corsika-reader**: IceTray reader for standard format (deprecated)
 - ▶ **CorsikaInjectorService** (IceTop)
- ▶ Muons:
 - ▶ **MuonGun**: parametrization of flux of atm. muons under the ice.
- ▶ Neutrinos:
 - ▶ **neutrino-generator**: injects neutrinos, propagates them through Earth, forces interaction in detector volume.
 - ▶ **genie-reader**: detailed simulation of neutrino interactions with GENIE. (Used for low-energy simulations)
 - ▶ **LeptonInjector / NuFSGen** (not yet available): weighted leptons+weights to account for flux models, interaction models, in-earth propagation, etc.

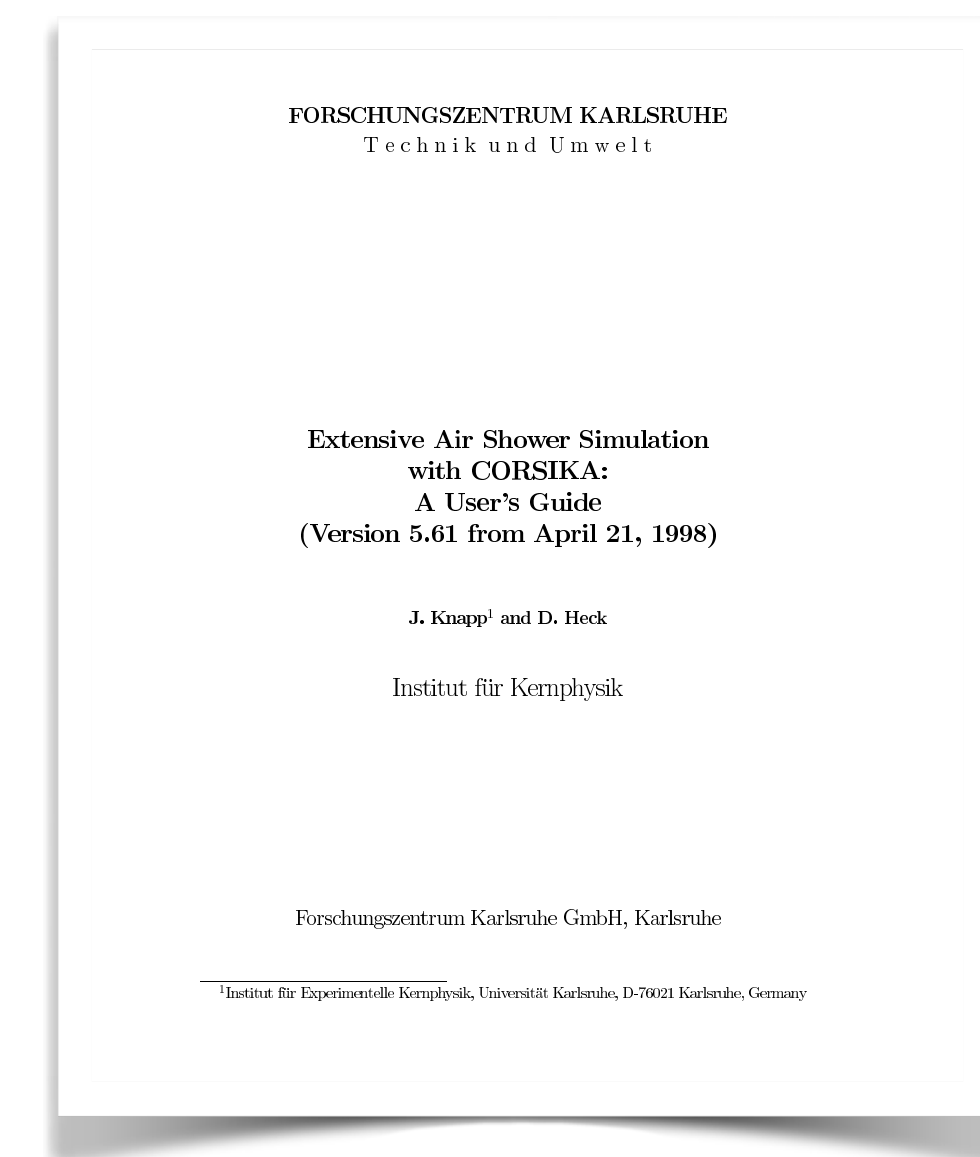
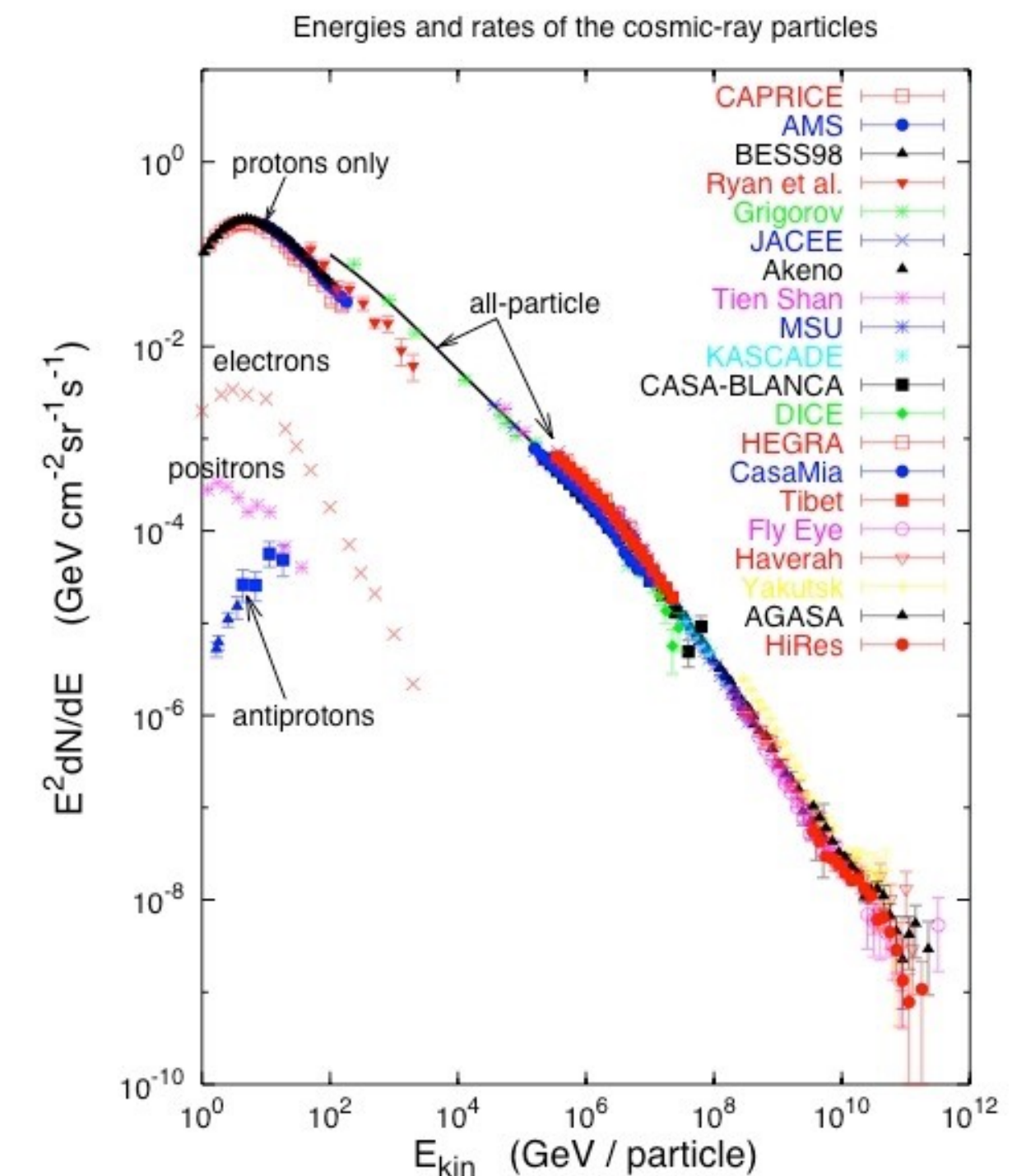


Generators : CORSIKA

(**CO**smic **R**ay **SI**mulations for **KA**scade)

- Particles are tracked through the atmosphere until they undergo reactions with the air nuclei or - in the case of instable secondaries - decay.
- The hadronic interactions at high energies may be described by several reaction models alternatively:
 - *VENUS*, *QGSJET*, and *DPMJET* (Gribov-Regge theory),
 - SIBYLL (minijet model).
 - *neXus*, *EPOS* (combination of *QGSJET* and *VENUS*).
 - *HDPM* (Dual Parton Model).
- Hadronic interactions at lower energies:
 - *GHEISHA*, *FLUKA*, or *UrQMD* models.
- For electromagnetic interactions
 - Tailored version of *EGS4*.
 - Analytical *NKG* formulas.

- ▶ weighted events : artificial spectrum $E^{-\gamma}$
- ▶ 5 representative mass groups: (H,He,CNO,Mg,Fe)



MuonGun (IceCube implementation of MUPAGE)

arXiv:0907.5563v1 [astro-ph.IM] 31 Jul 2009

- Parametrized muon distributions from many, many CORSIKA showers.
- Produce muons in detector volume by sampling from splinned PDF tables
 - zenith angle
 - depth
 - energy
 - bundle multiplicity
 - radius

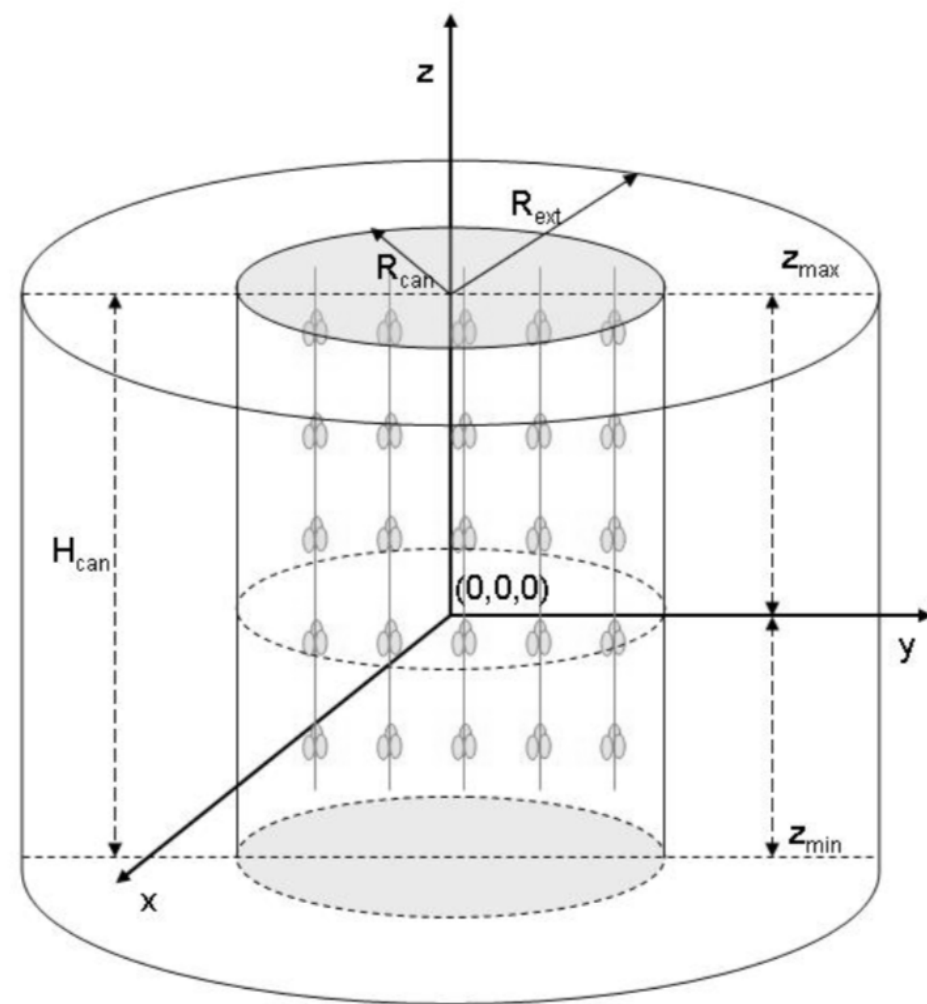
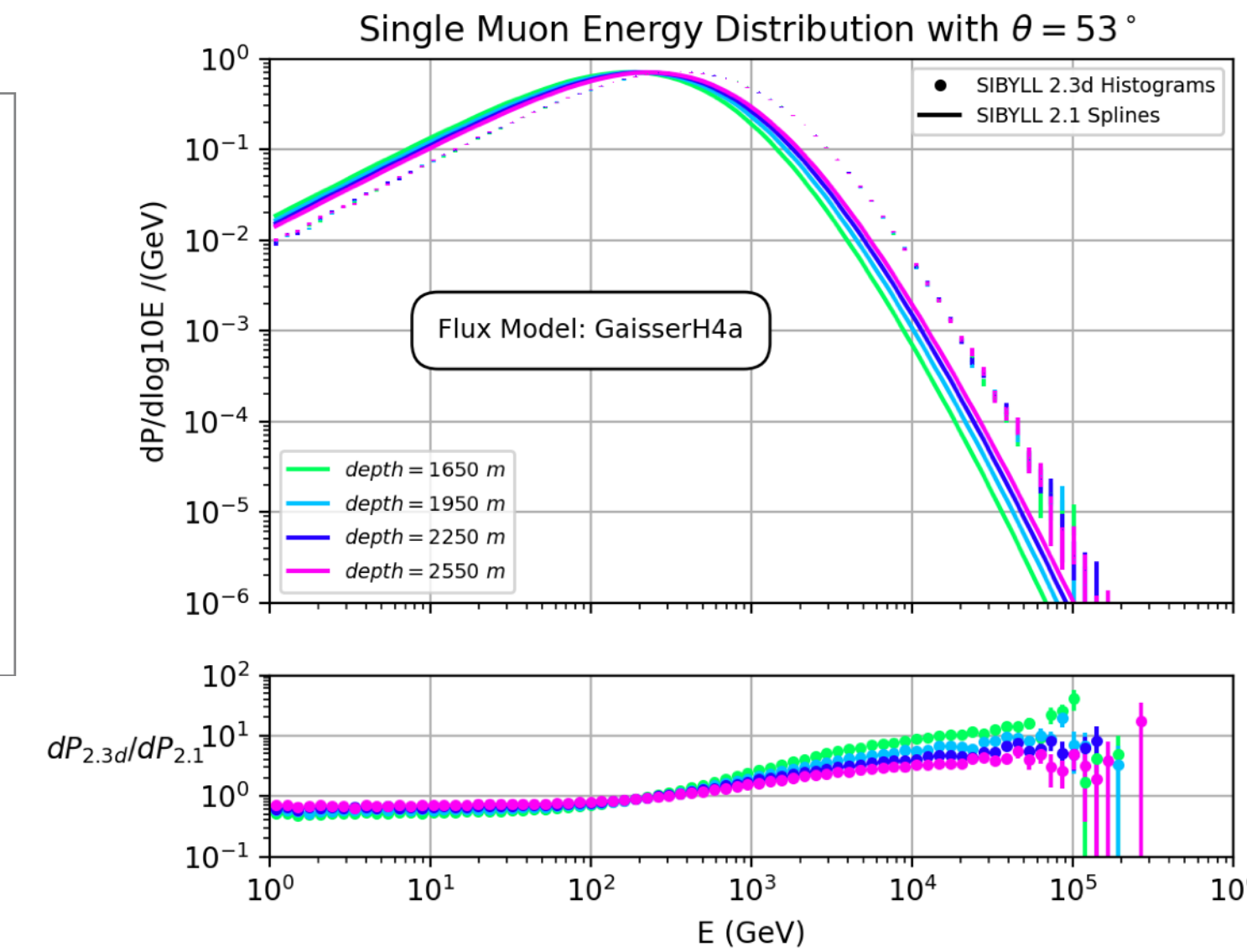
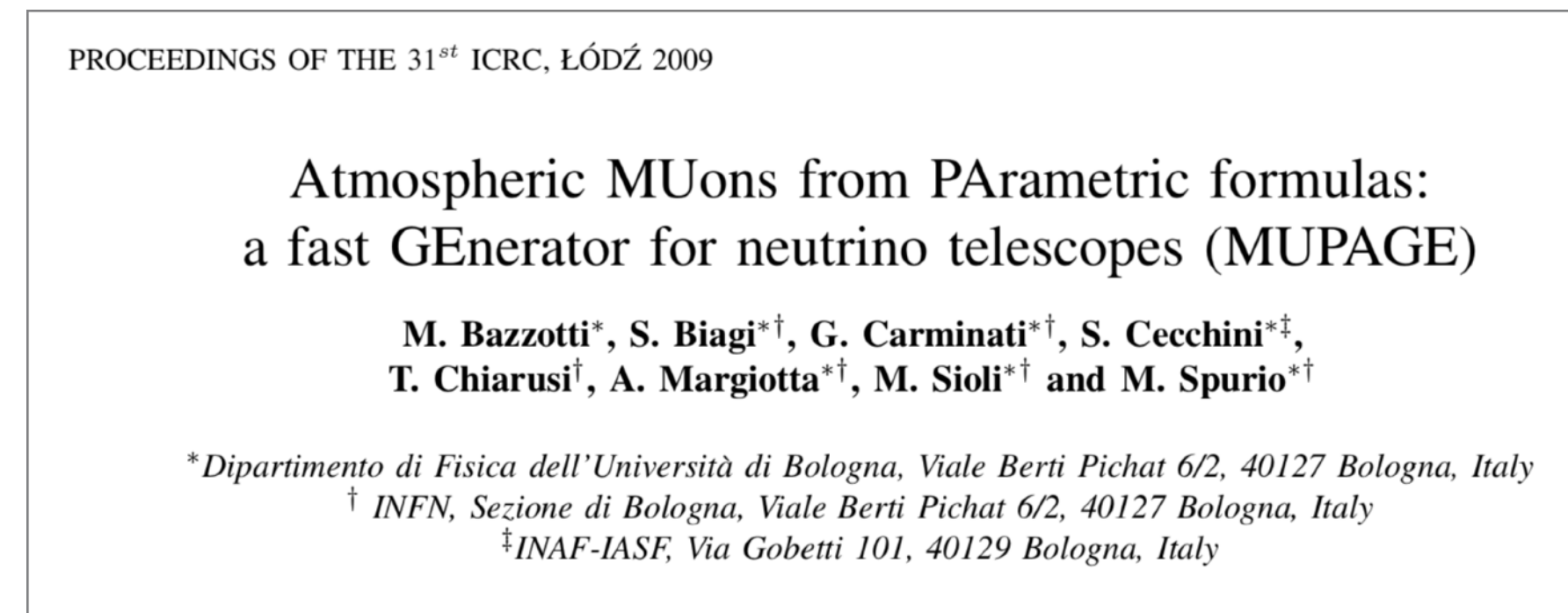
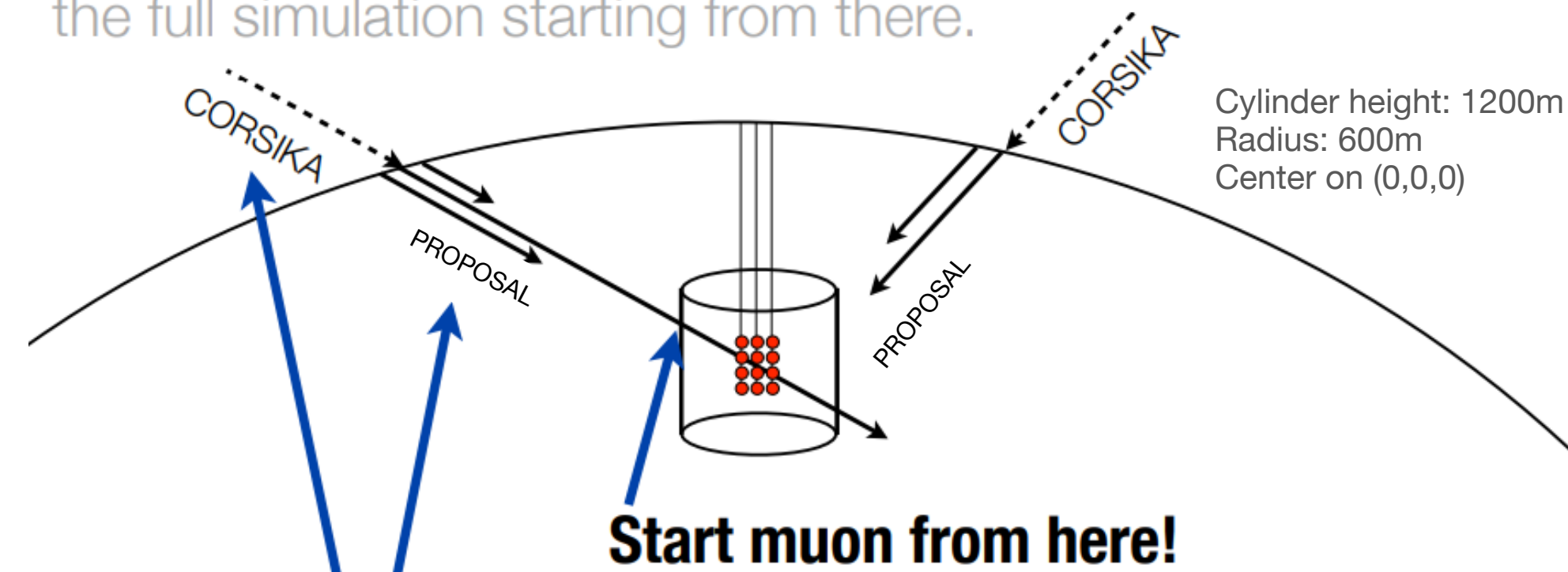


Fig. 1: Sketch of some input parameters. The cylinder surrounding the instrumental volume is the *can*, with radius R_{can} and height H_{can} . The events are generated on an extended can with R_{ext} . The origin of the coordinate system does not have to be located at the center of the detector. The lower disk is at a depth H_{max} with respect to the sea/ice surface.

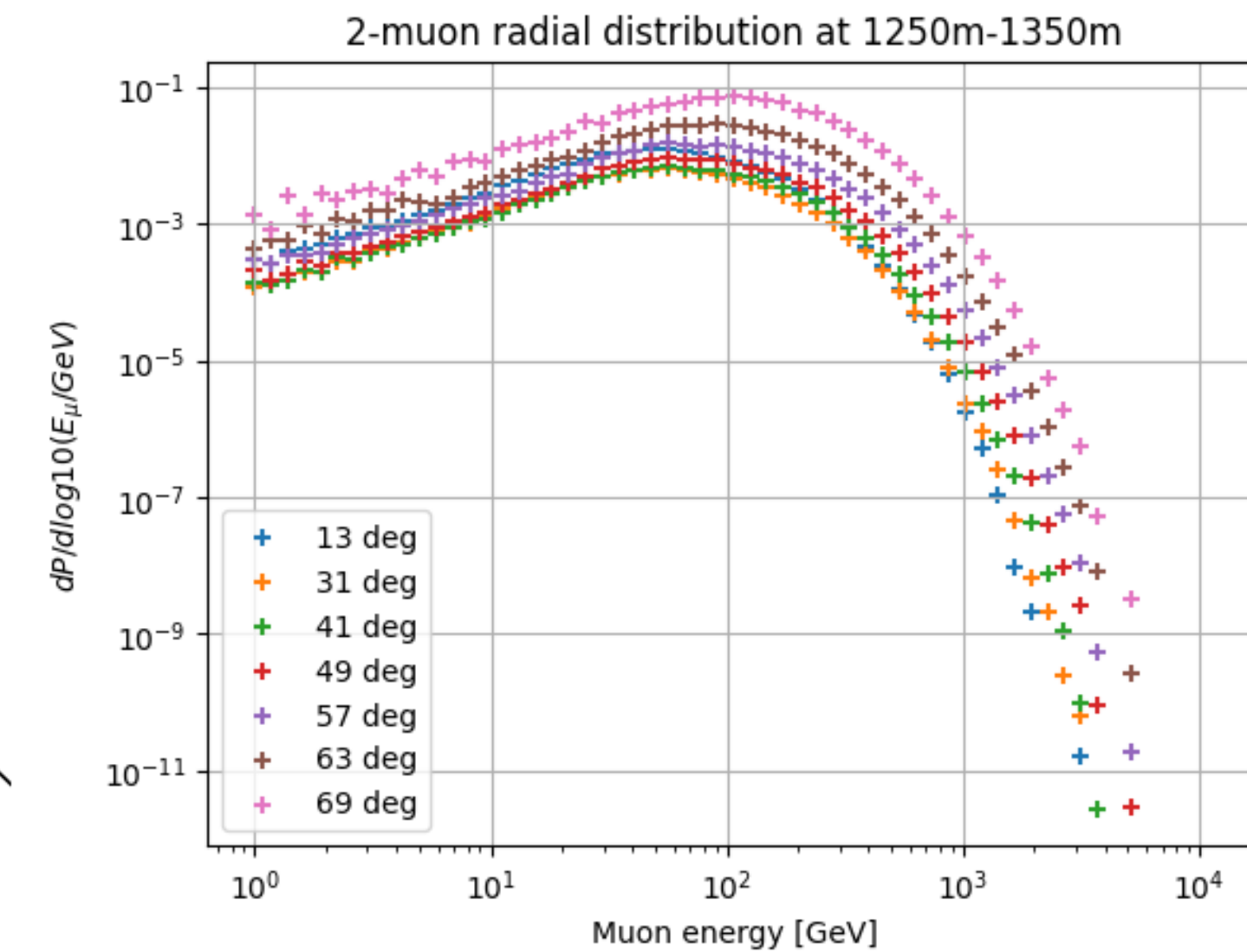
MuonGun

Start muons on a cylinder around the detector and only to the full simulation starting from there.



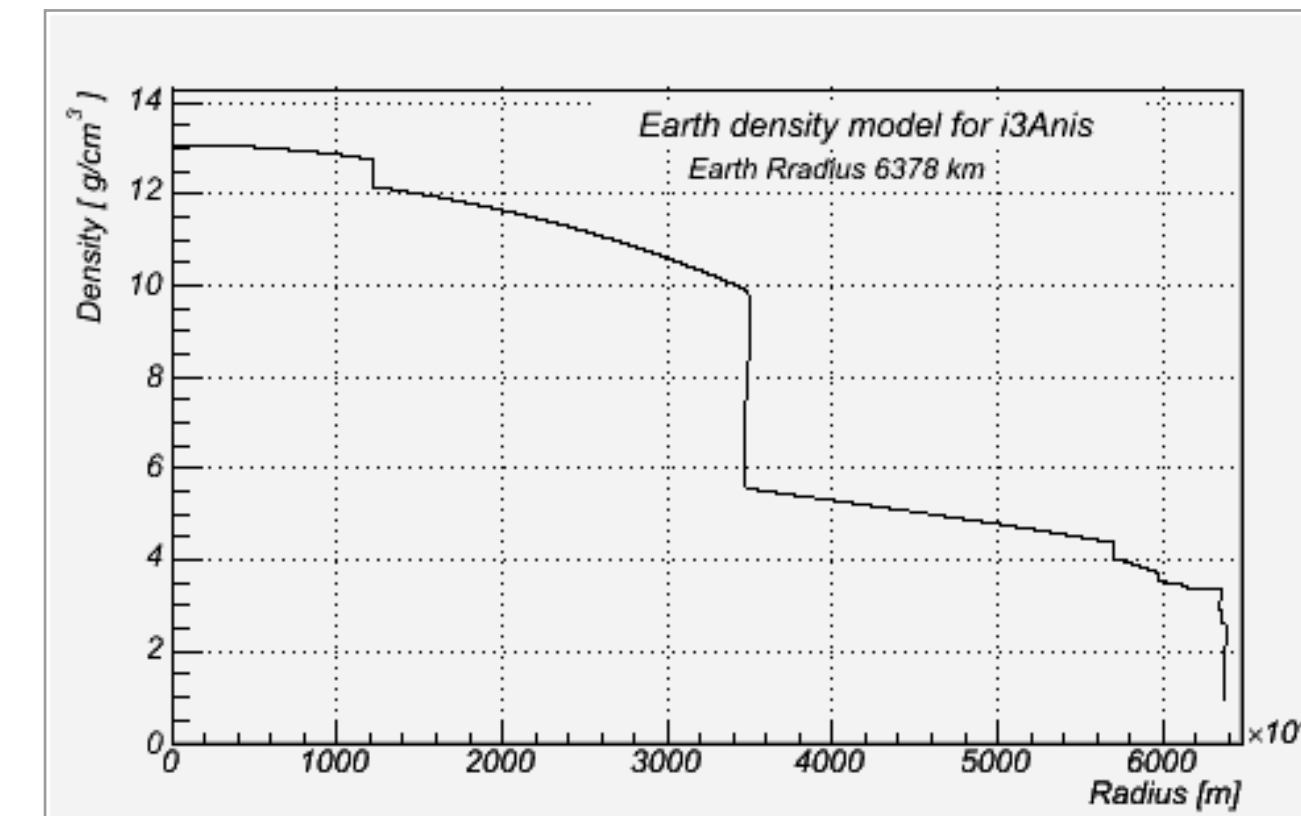
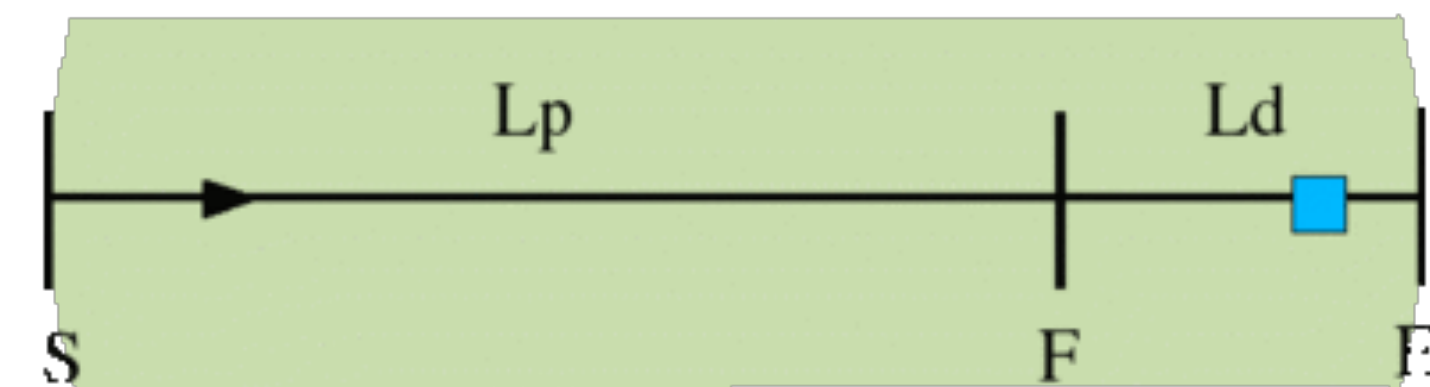
Handled by the
MuonGun weighting code

Siwei Wang, Diffuse parallel Tuesday 14:00

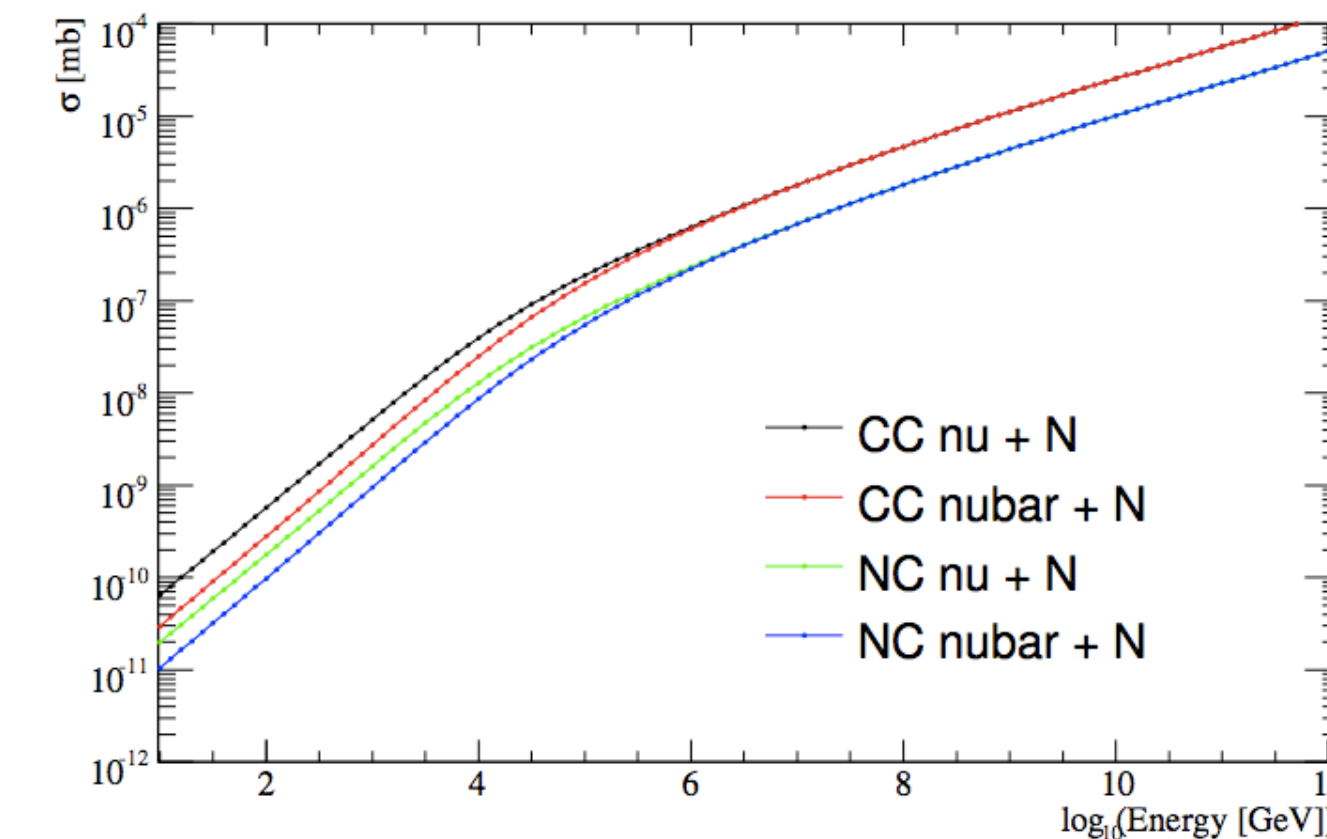


neutrino-generator

1. Calculate total path length inside the Earth using injected neutrino geometry.
 - a. Separate the total path length into propagation area (SF) and
 - b. detection volume (FE).
2. Define a step length $dx[m]$ using propagation area and step number.
3. For each step:
 - a. Calculate a column depth and Earth's density at the step point.
 - b. Calculate a total cross section at the step point.
 - c. Calculate a probability that the injected neutrino interacts within the step. Try Monte-Carlo, and decide whether an interaction happened within the step.
 - d. If interaction occurred: choose interaction randomly.
 - i. If CC-interaction is selected with injection particle NuMu or NuE, break (event is killed).
 - ii. else, generate secondaries and continue to next step.
 - e. If nothing happens, continue next step.
4. Finish propagation when injected neutrino + secondaries reach surface of detection volume (point F), then process a weighted interaction.



- produce a $E^{-\nu}$ ν_μ , ν_e , ν_τ with
 - PRELIM Earth's density model



- parton distribution functions
- prop & interaction of neutrinos into a weight

LeptonInjector/LeptonWeighter

- LeptonInjector and LeptonWeighter are designed for large-volume Cherenkov neutrino telescopes such as IceCube.
- The neutrino event generator allows for quick and flexible simulation of neutrino events within and around the detector volume
- Implements the leading Standard Model neutrino interaction processes relevant for neutrino observatories:
 - neutrino-nucleon deep-inelastic scattering
 - neutrino-electron annihilation.

This is publicly available code.

<https://arxiv.org/abs/2012.10449>

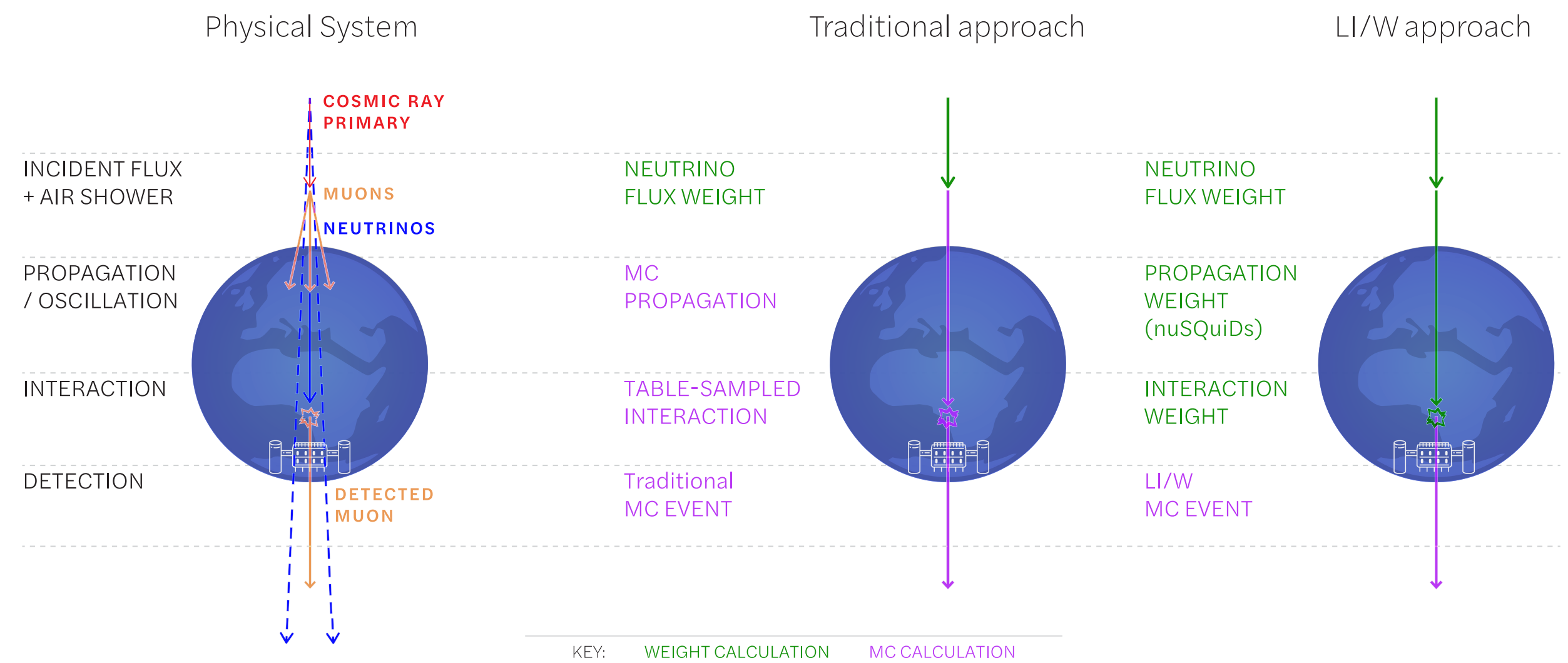


Figure 1.1: A diagram illustrating the different event generation and weighting steps for traditional methods compared with the LeptonInjector and LeptonWeighter philosophy.

Lepton propagation

- ▶ **PROPOSAL**: parametrized interactions with the medium.

[Comp. Phys. Com. 184, 9 \(2013\), p2070-2090](#)

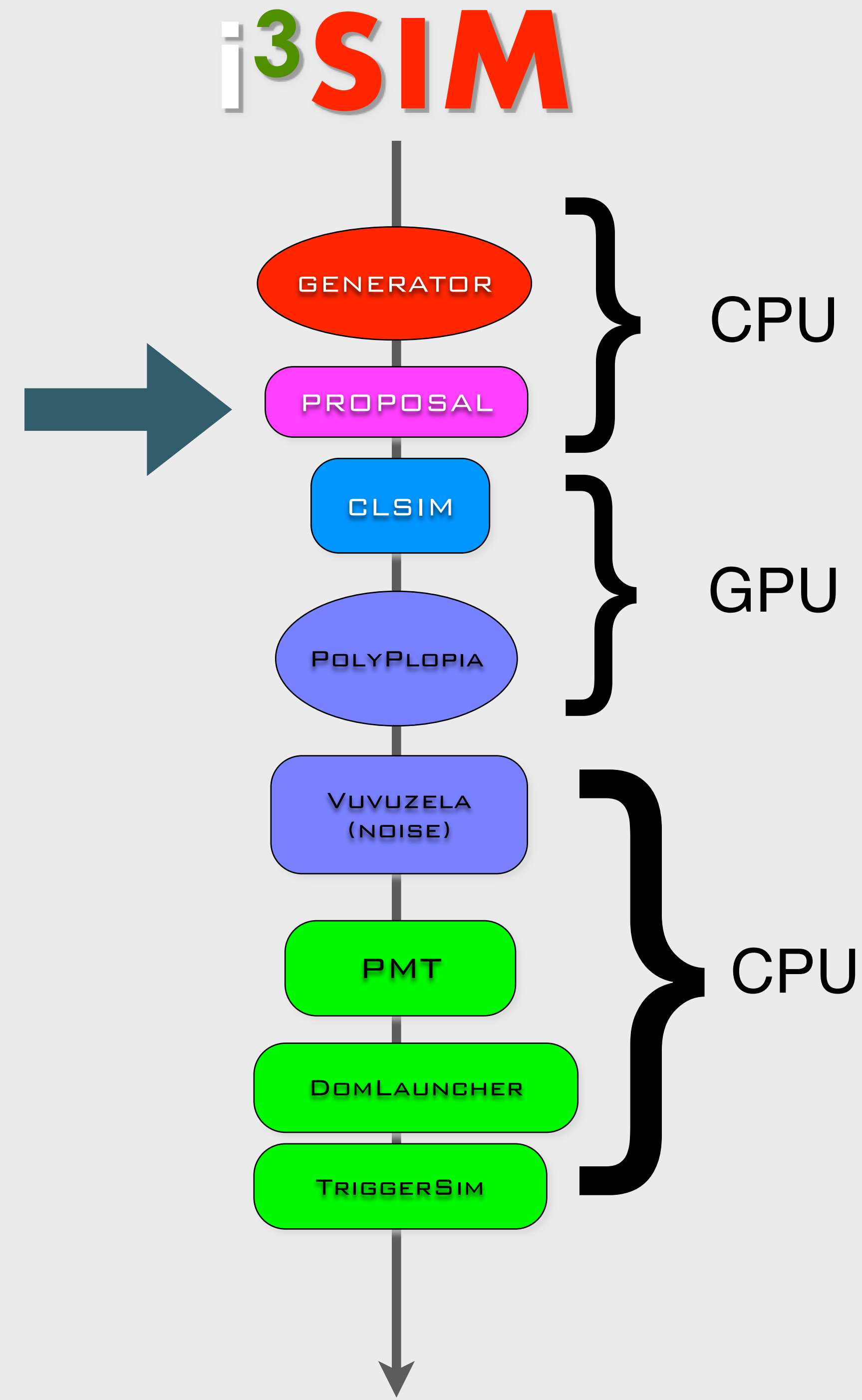
- ▶ Stochastic energy losses include:

- ▶ ionization
- ▶ electron-pair production
- ▶ bremsstrahlung
- ▶ photo-nuclear interaction
- ▶ decay

- ▶ **GEANT4**: Detailed particle propagation in media.

<https://geant4.web.cern.ch/>

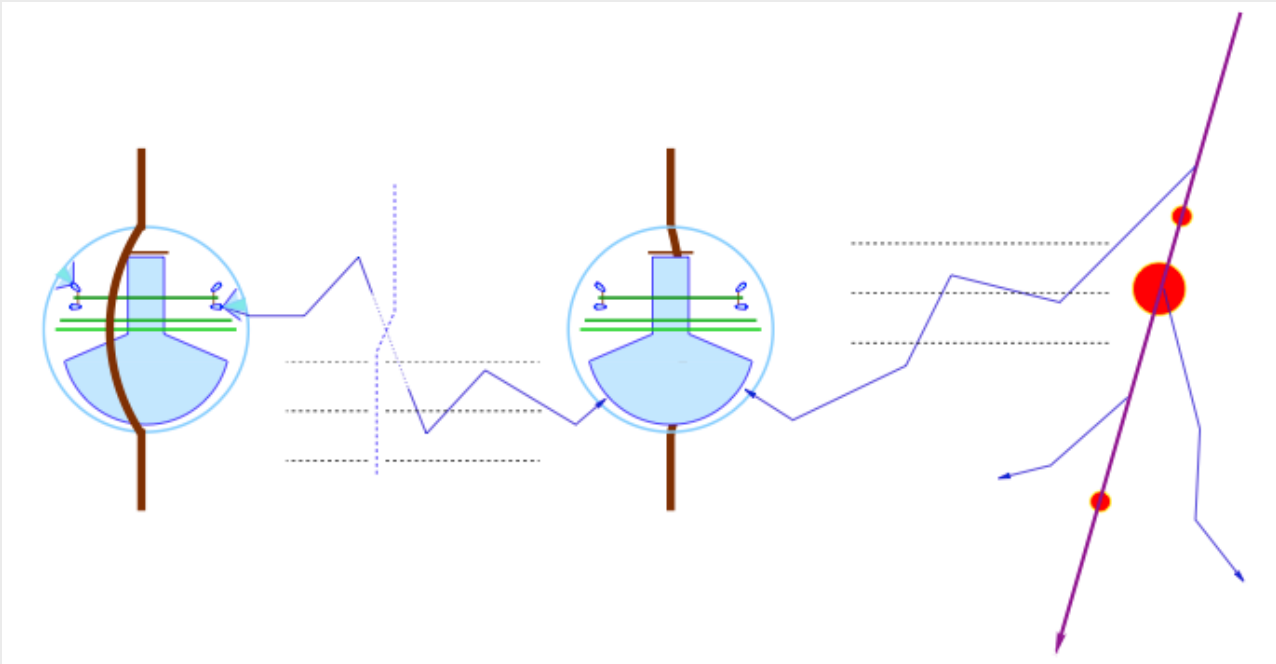
- ▶ 3rd-party G4 library used by CLSim to propagate leptons for low-energy simulations (CPU-intensive).



Photon Propagation (PPC, CLSim)

μ energy loss + cascades $\rightarrow$ photons $\rightarrow$ p.e.

Photon propagation : ice properties + PMT response + DOM glass/gel



i3SIM

GENERATOR

PROPOSAL

CLSIM

POLYPLOPIA

VUVUZELA
(NOISE)

PMT

DOMLAUNCHER

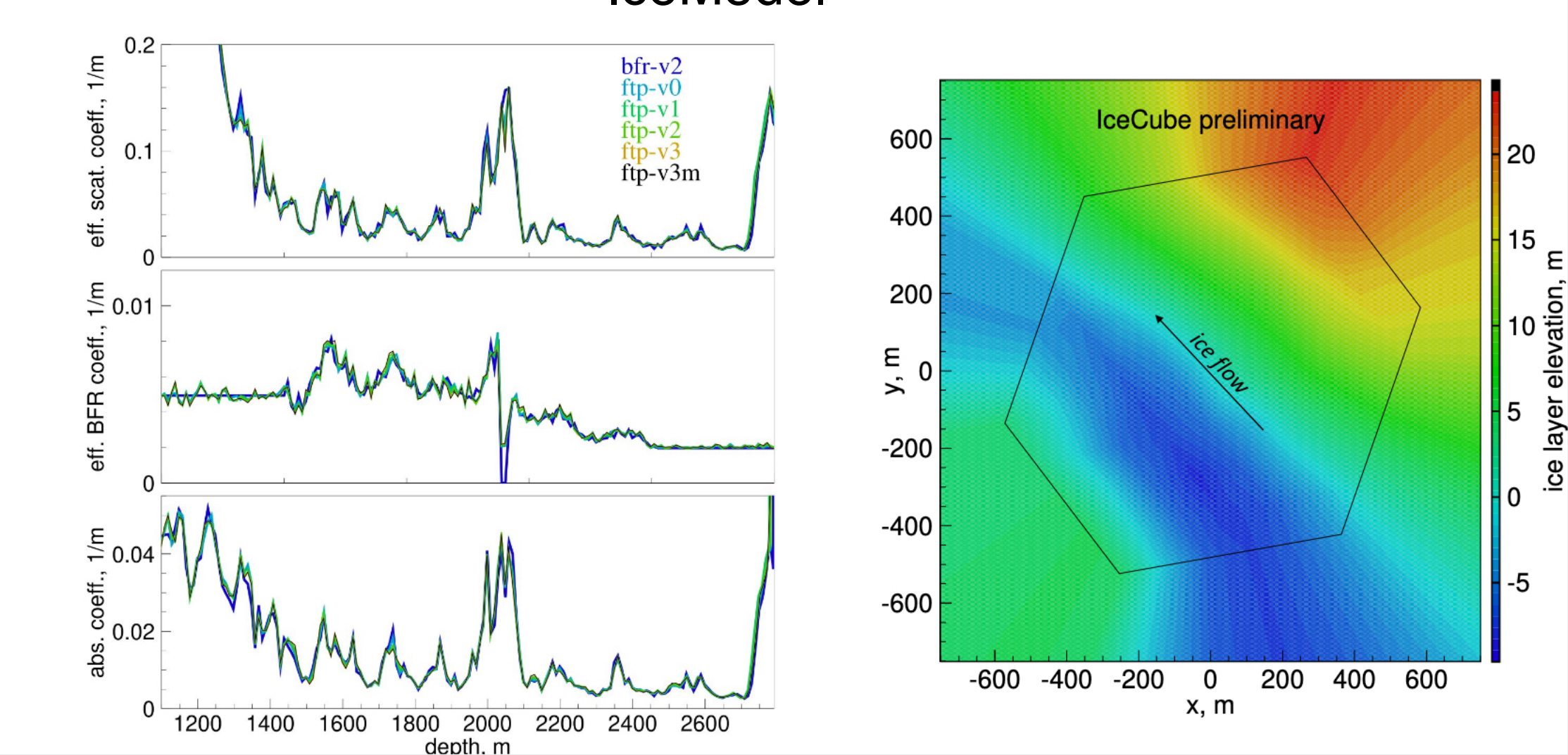
TRIGGERSIM

CPU

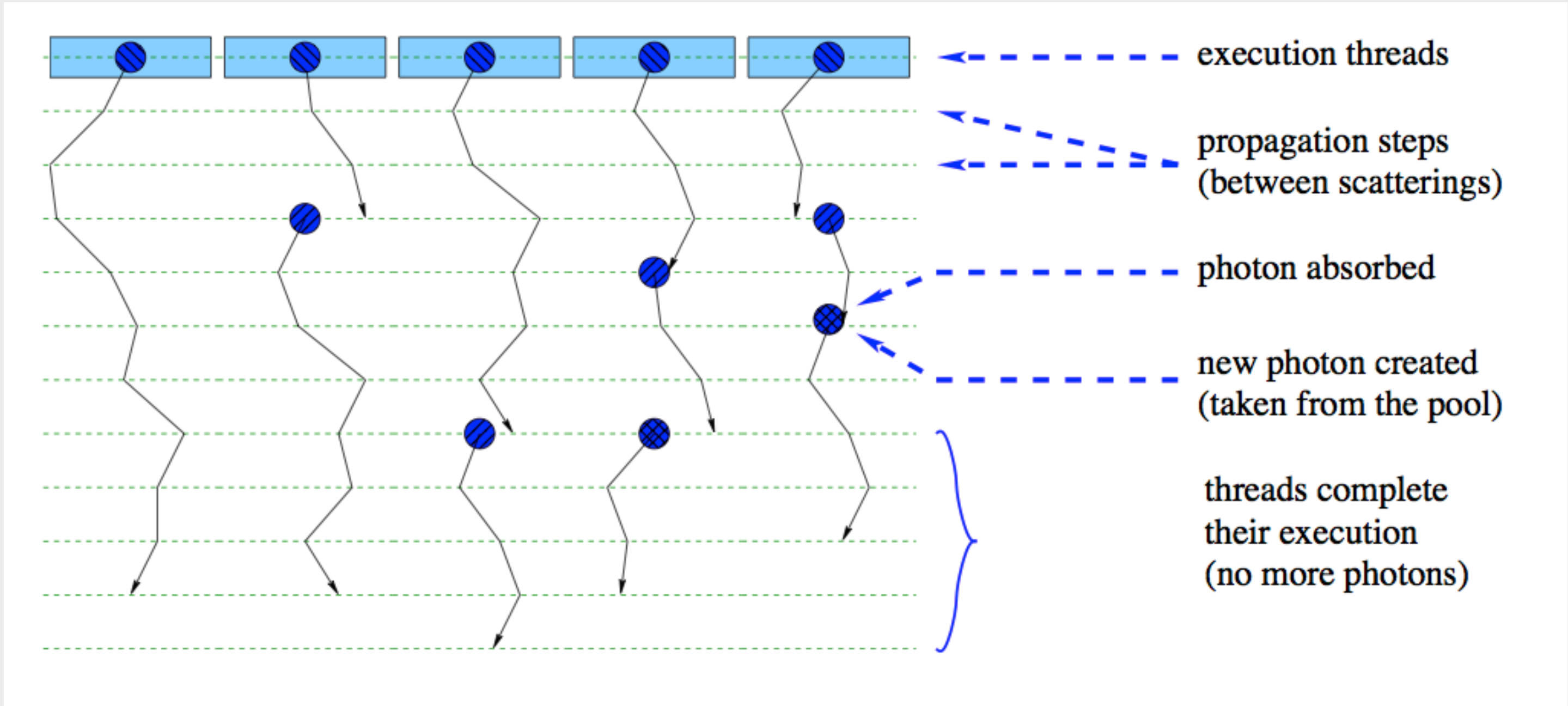
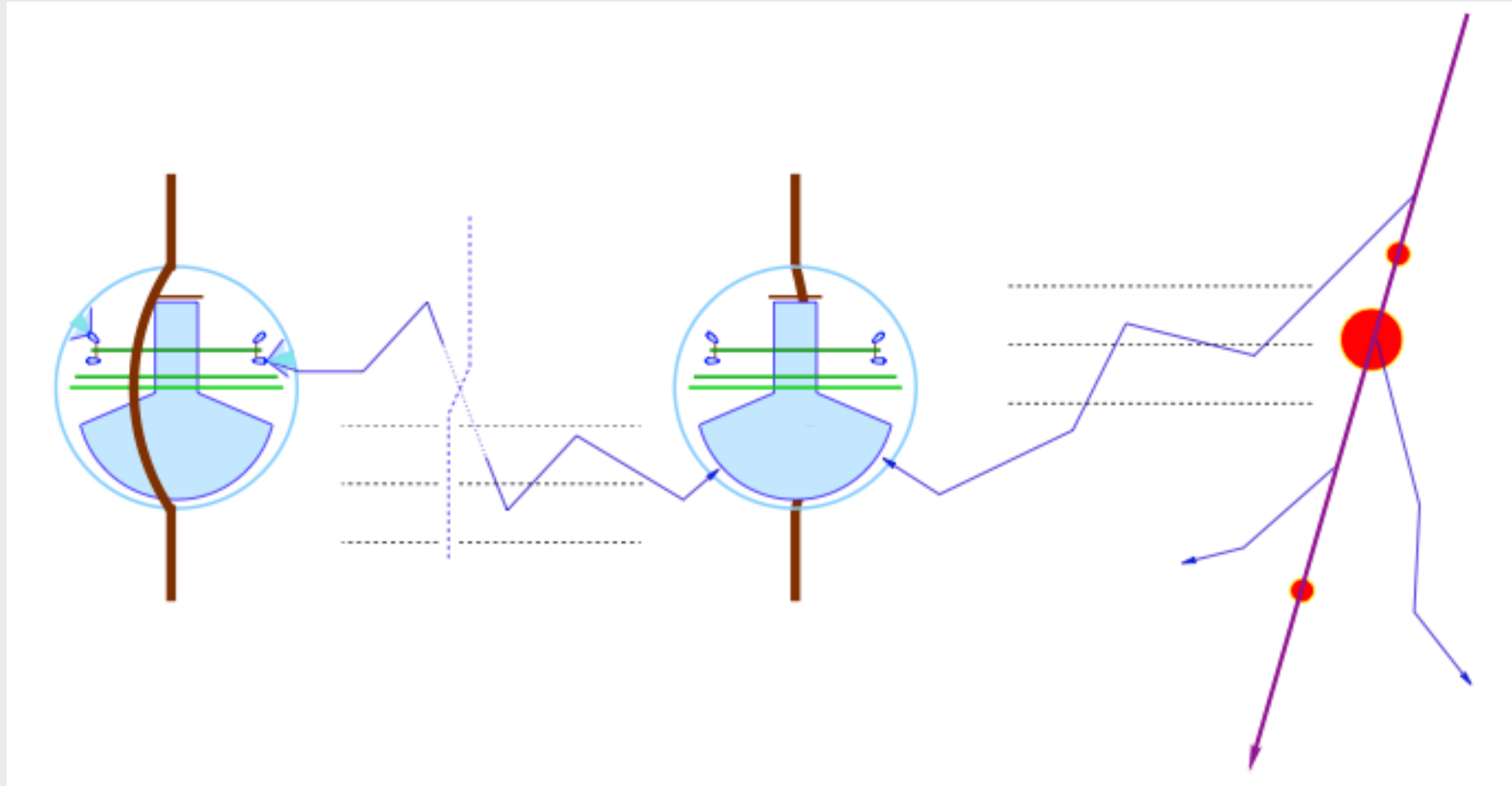
GPU

CPU

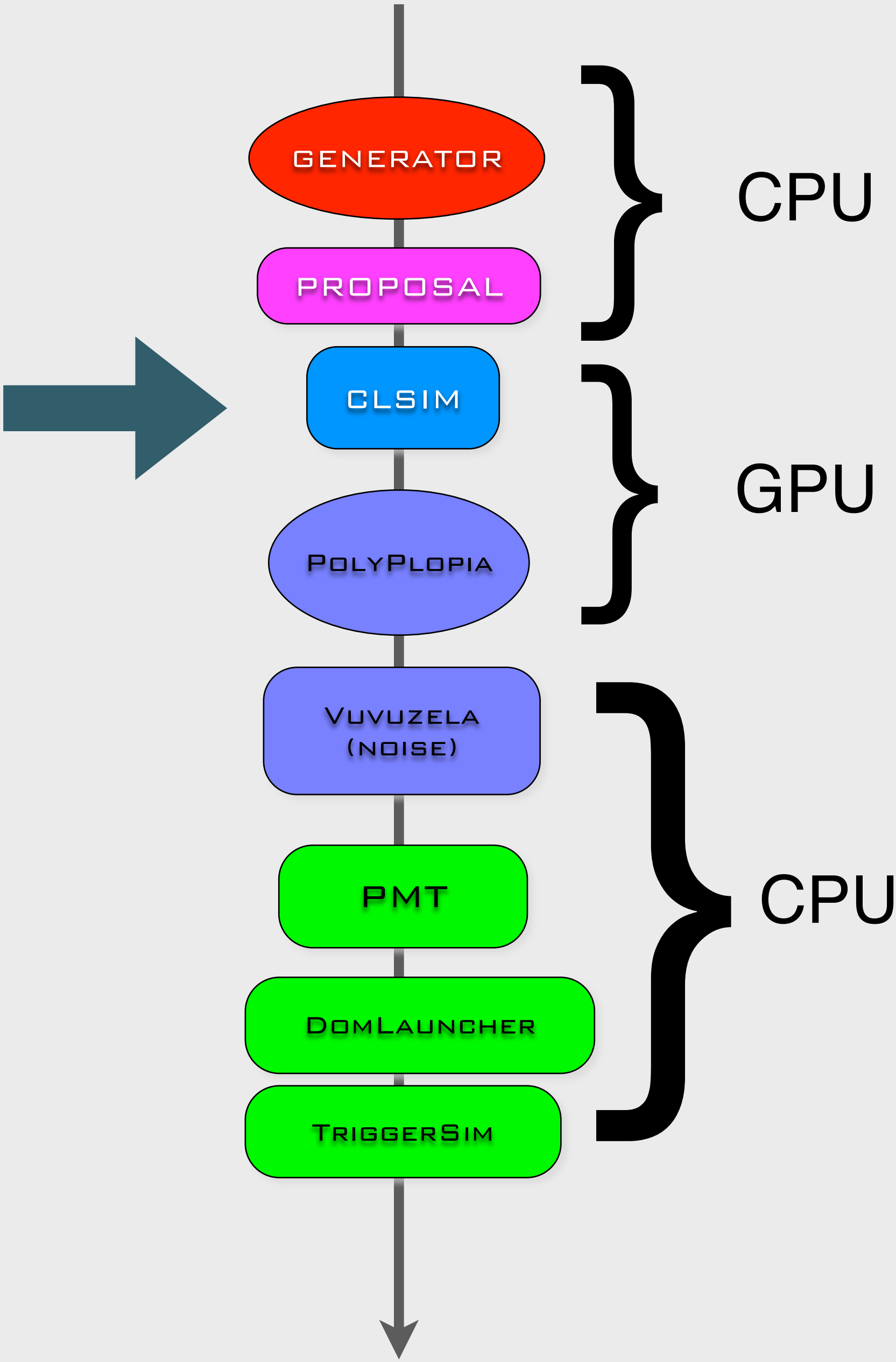
IceModel



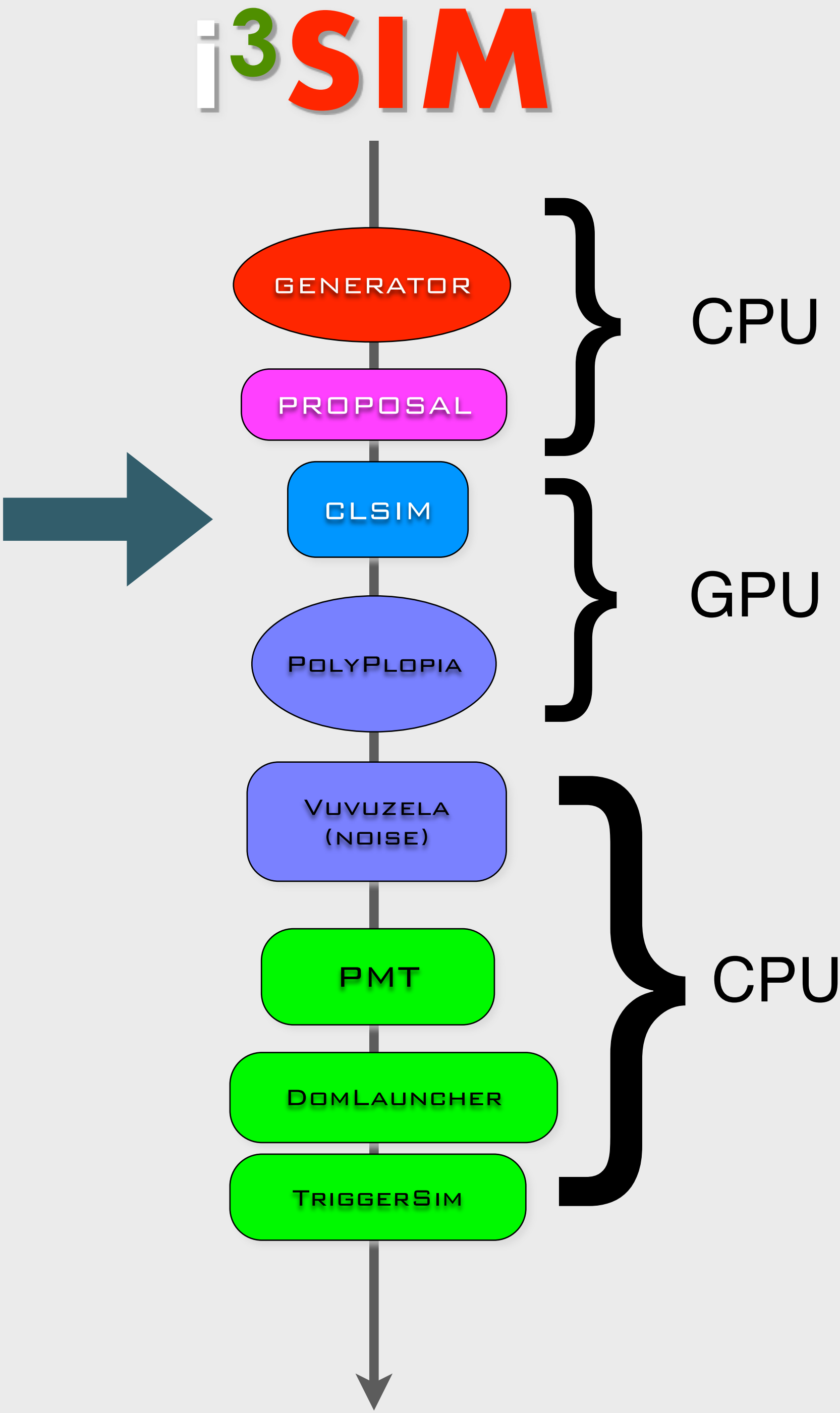
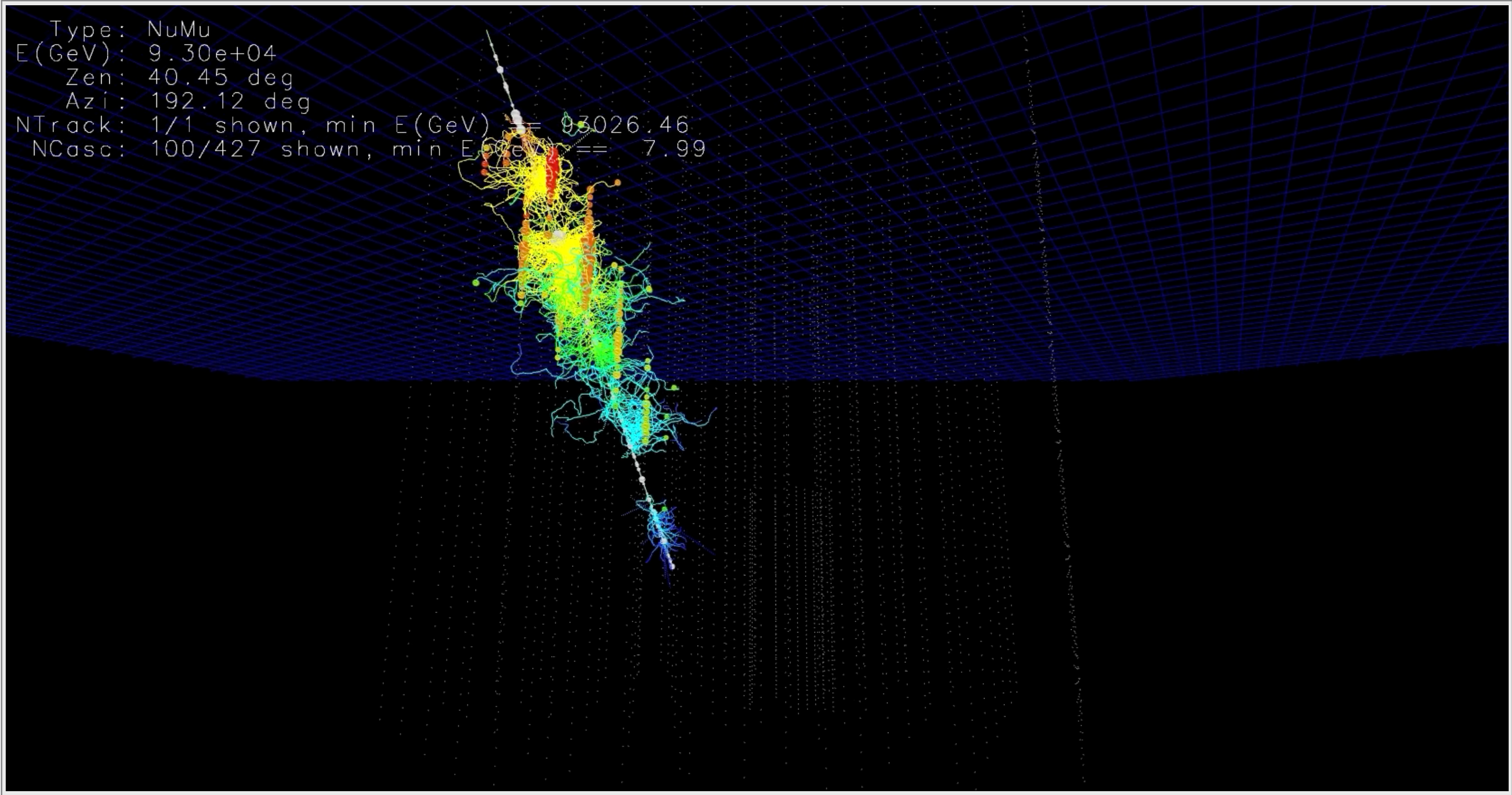
Photon Propagation (PPC, CLSim)



i3SIM



Photon Propagation (PPC, CLSim)



Polyplopia

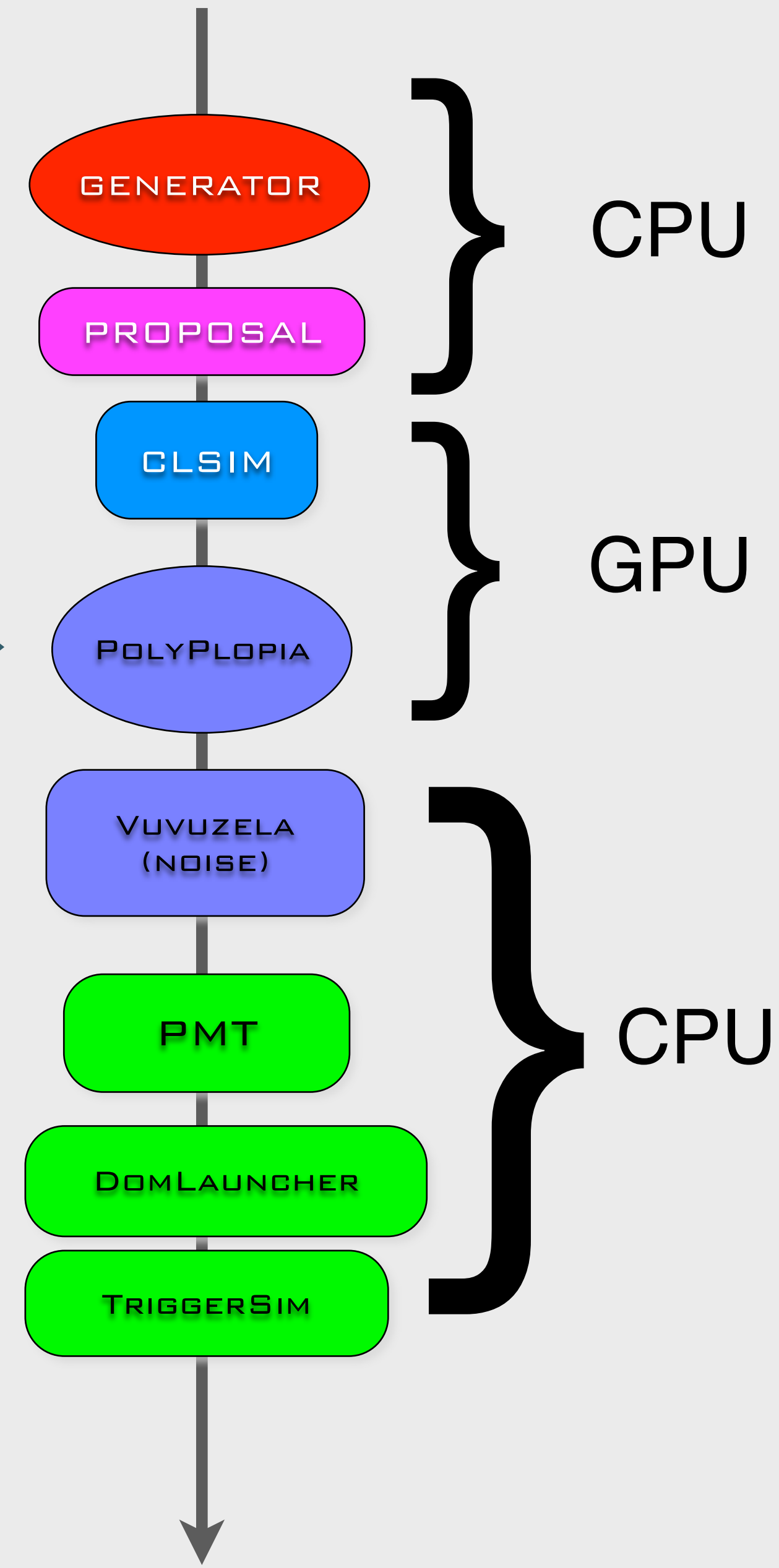
(from gr., πολύς - polús, "many," and ὥψ-ops , "vision")

Coincident atmospheric shower events in IceCube

- **PoissonMerger**

- Injects background event read from *CoincidentEventService* on top of primary events in the chain by sampling from a Poisson distribution over a time window Δt .
- Writes a separate I3MCtree with background particles.
- Writes a combined I3MCPE map for signal and background.
- Q-frames are later split into P-frames by trigger-splitter based on timing, topology

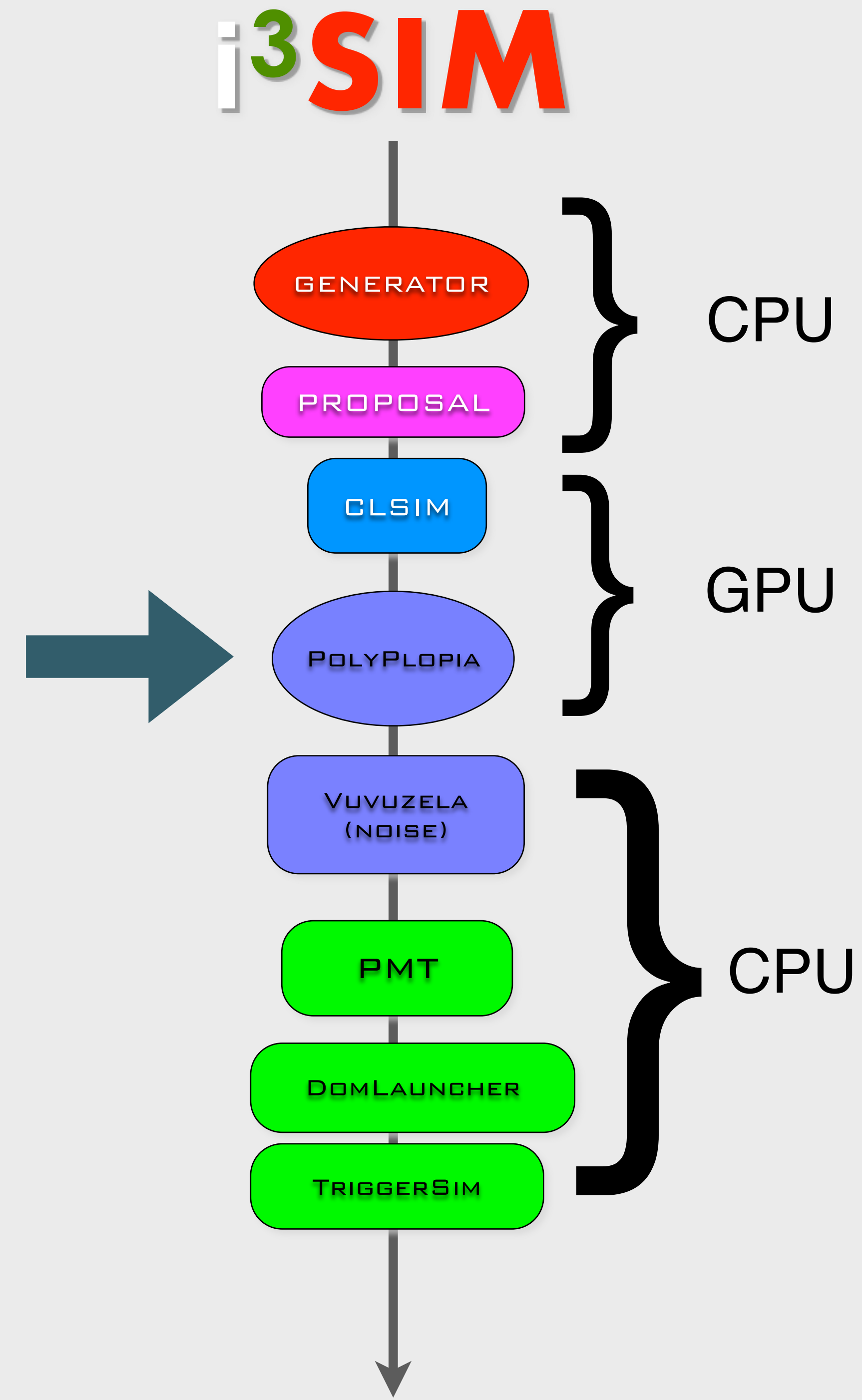
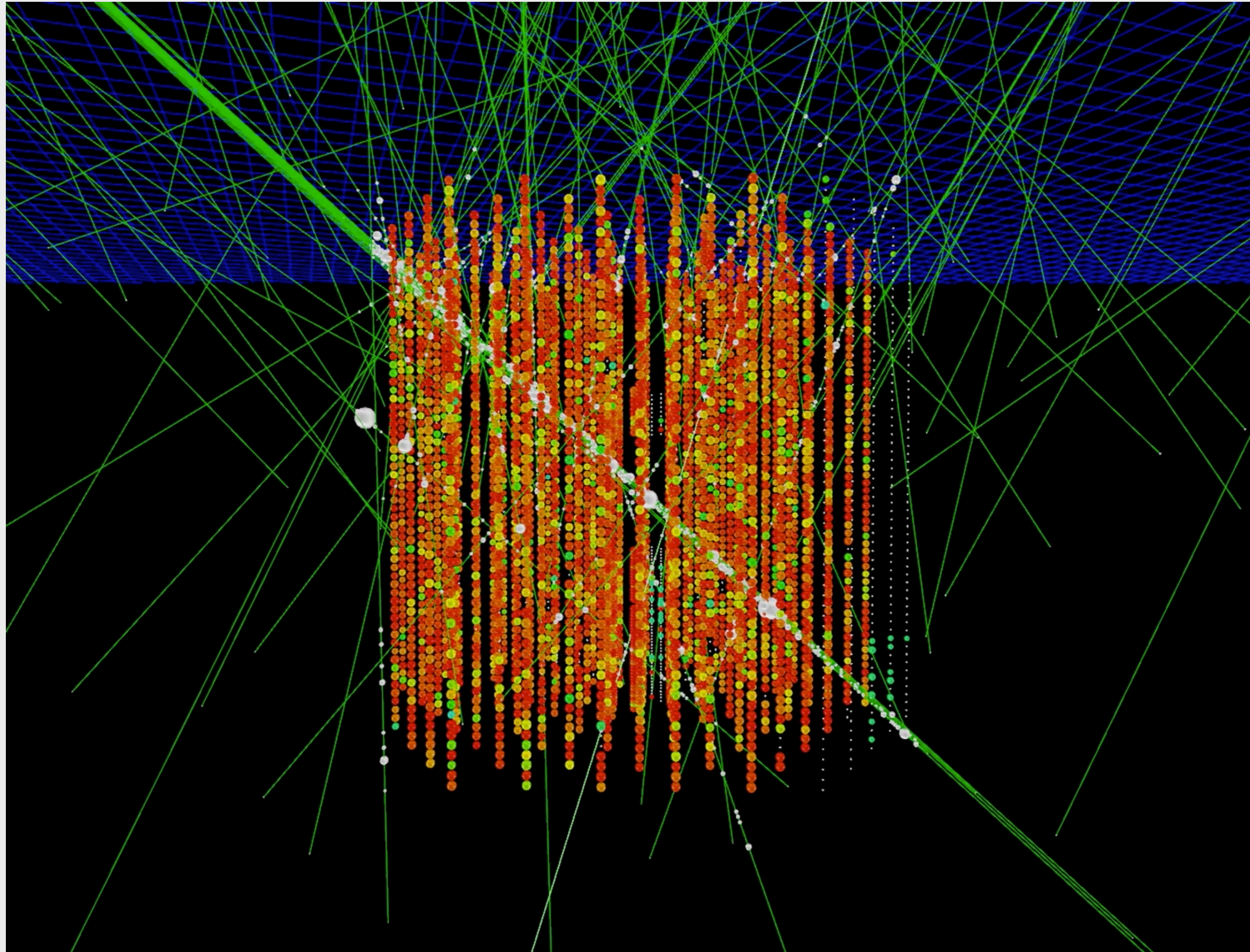
i3SIM



Polyploia

(from gr., πολύς - polús, "many," and ὄψ-ops , "vision")

Coincident atmospheric shower events in IceCube



Vuvuzela (Noise)

Noise Model

Thermal Noise ($\sim$ few Hz)
[Poisson process]

$\sim$ ms Timescales

+

Radioactive Decay in Glass
[Poisson process]

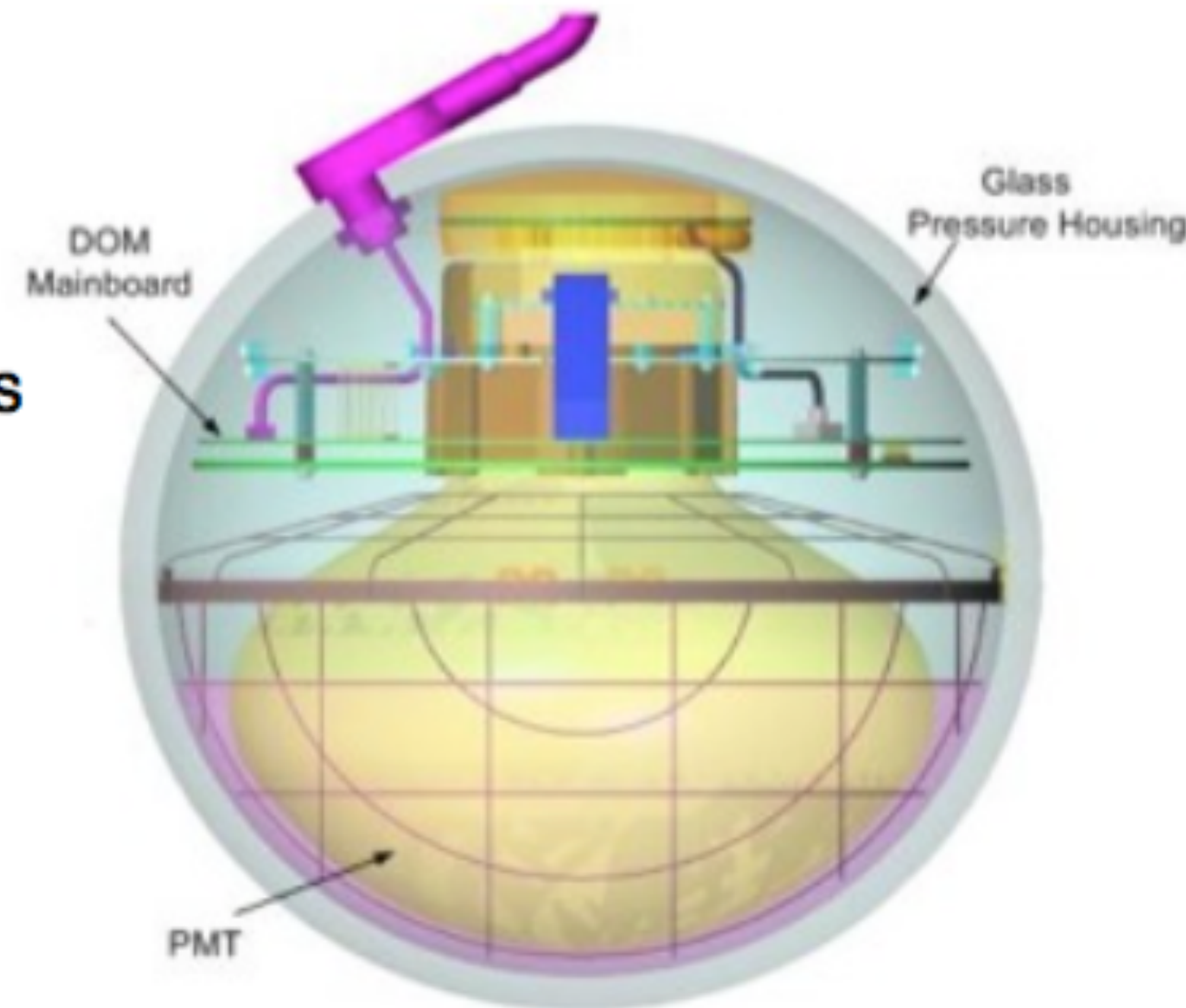
$\sim$ ms Timescales

Energy deposited in glass

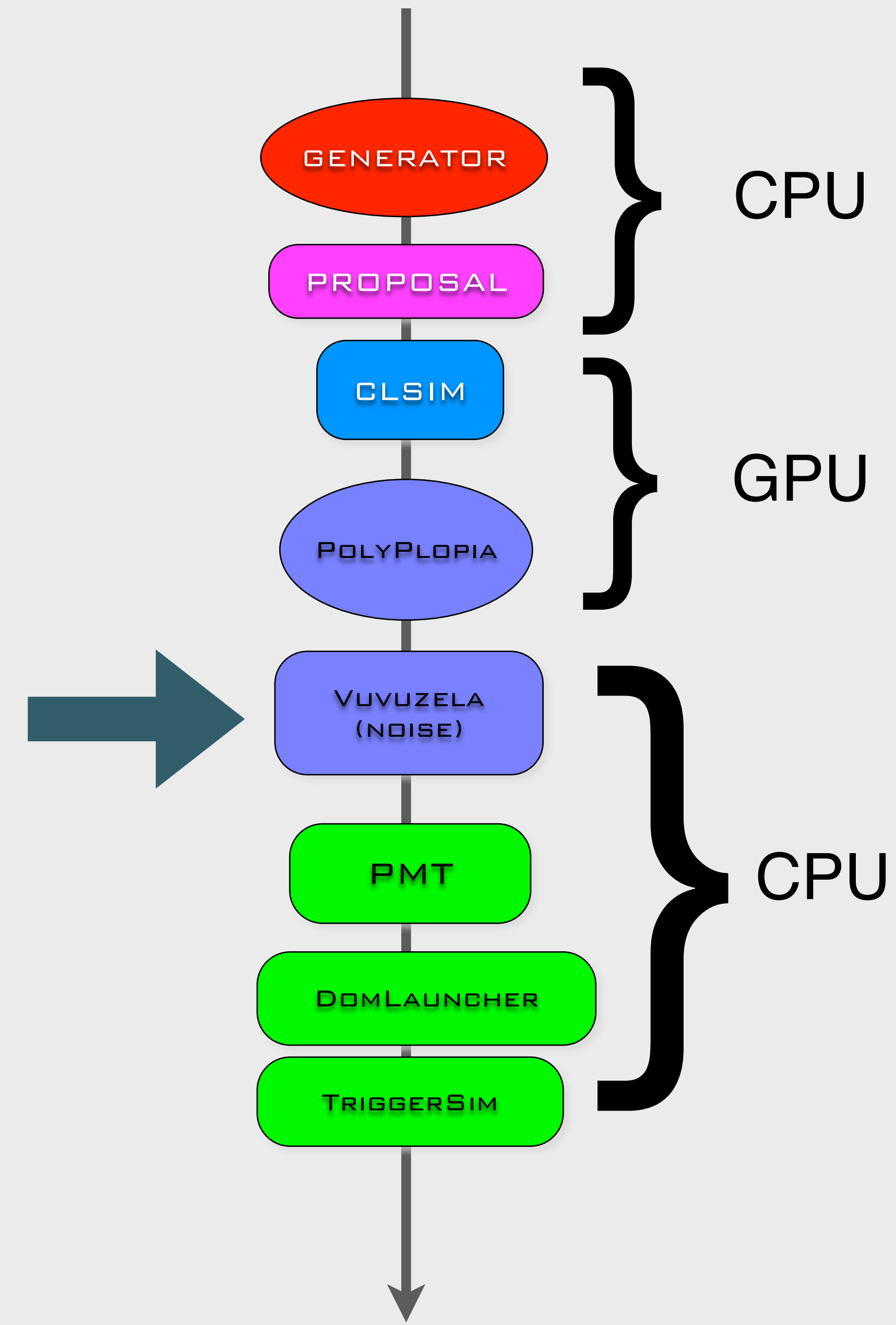
Glass scintillates/fluoresces
over long timescale
[Log-normal]

$\approx$ 500 μ s Timescales

$\rightarrow$ (MCPEs)



i3SIM



PMTResponseSimulator

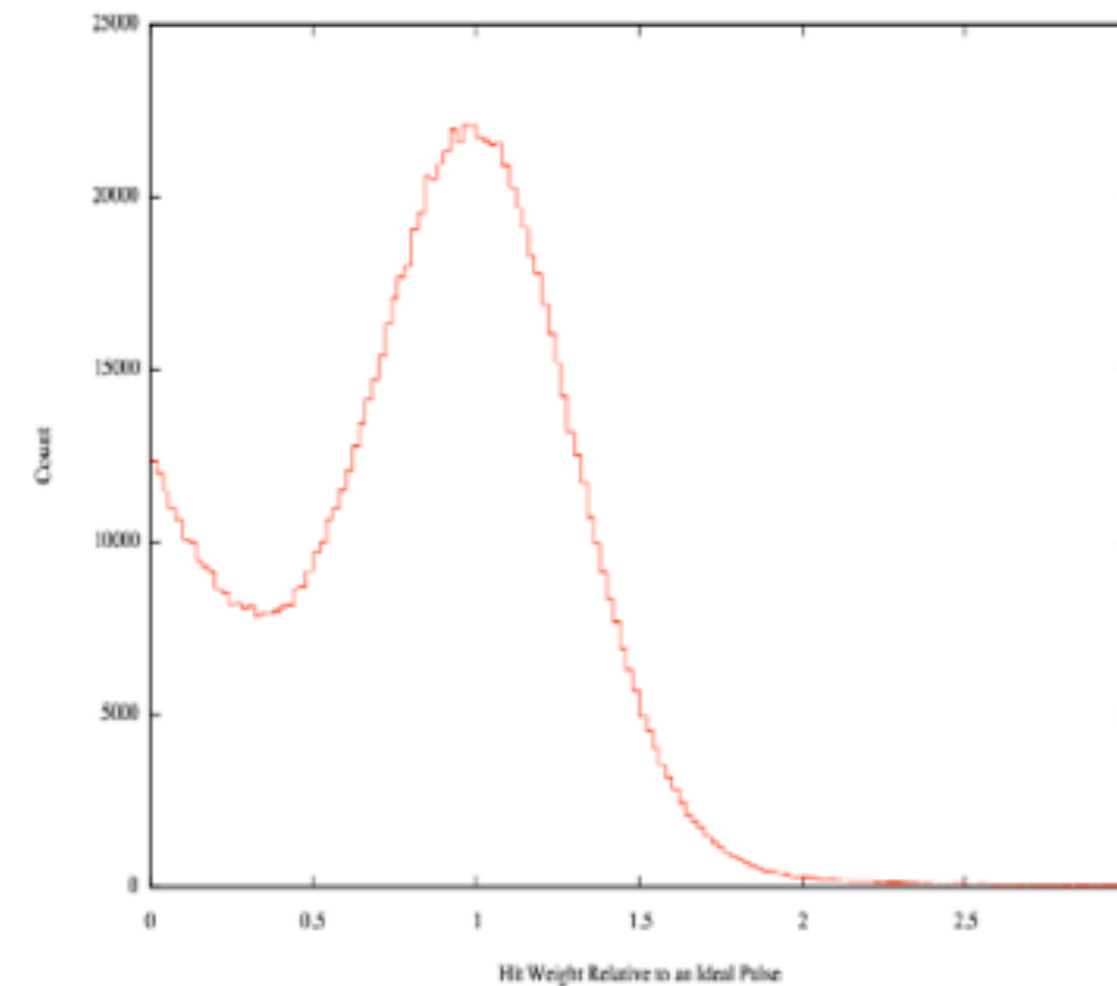


In- and output:

- Takes MCHits as input.
- Produces weighted MCHits.

Processing MCHits:

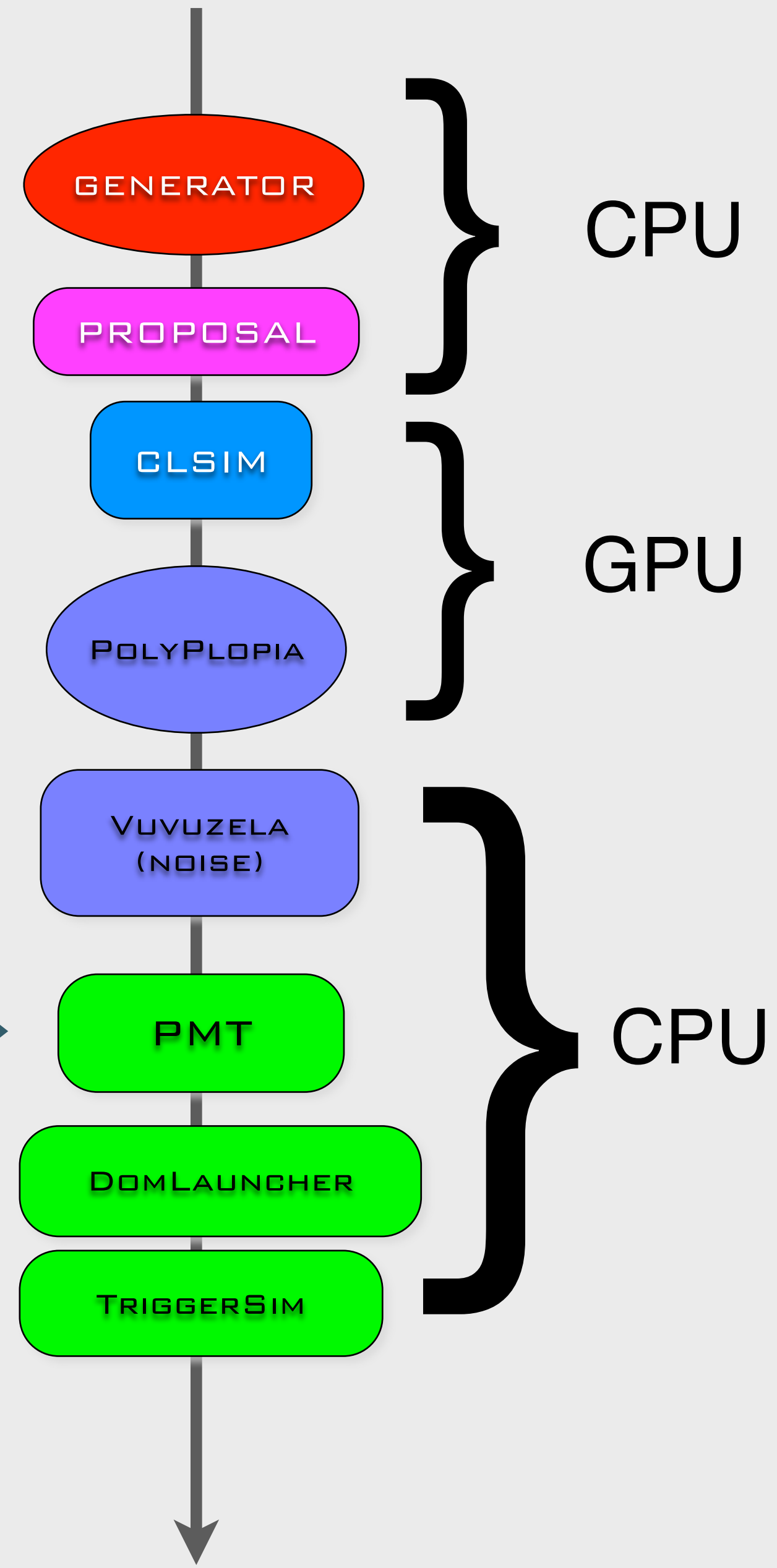
- Give each MCHit a weight corresponding to the pulse charge that photon would yield.
- Generate prepulses, late pulses and after pulses.
- Apply time jitter.
- Simulate the effect of saturation.



Distribution of hit weights

<http://software.icecube.wisc.edu/documentation/projects/DOMLauncher/index.html>

i3SIM

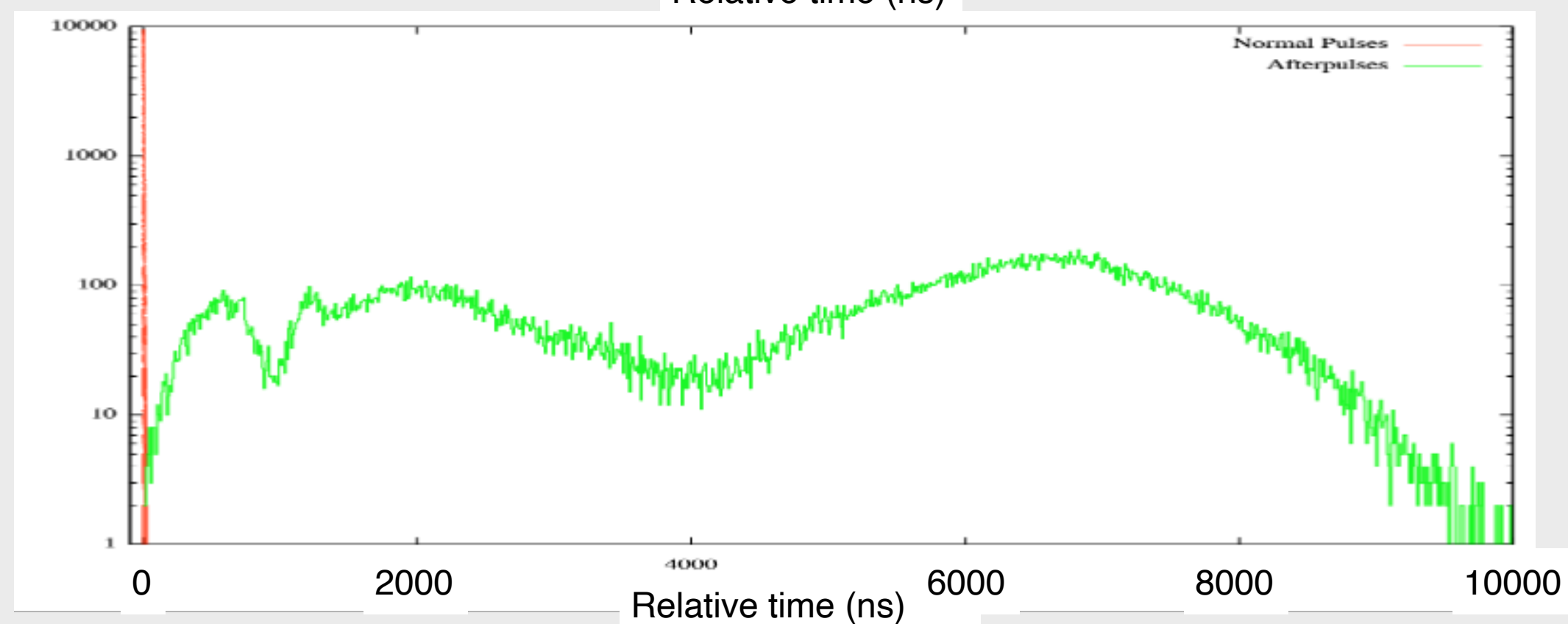
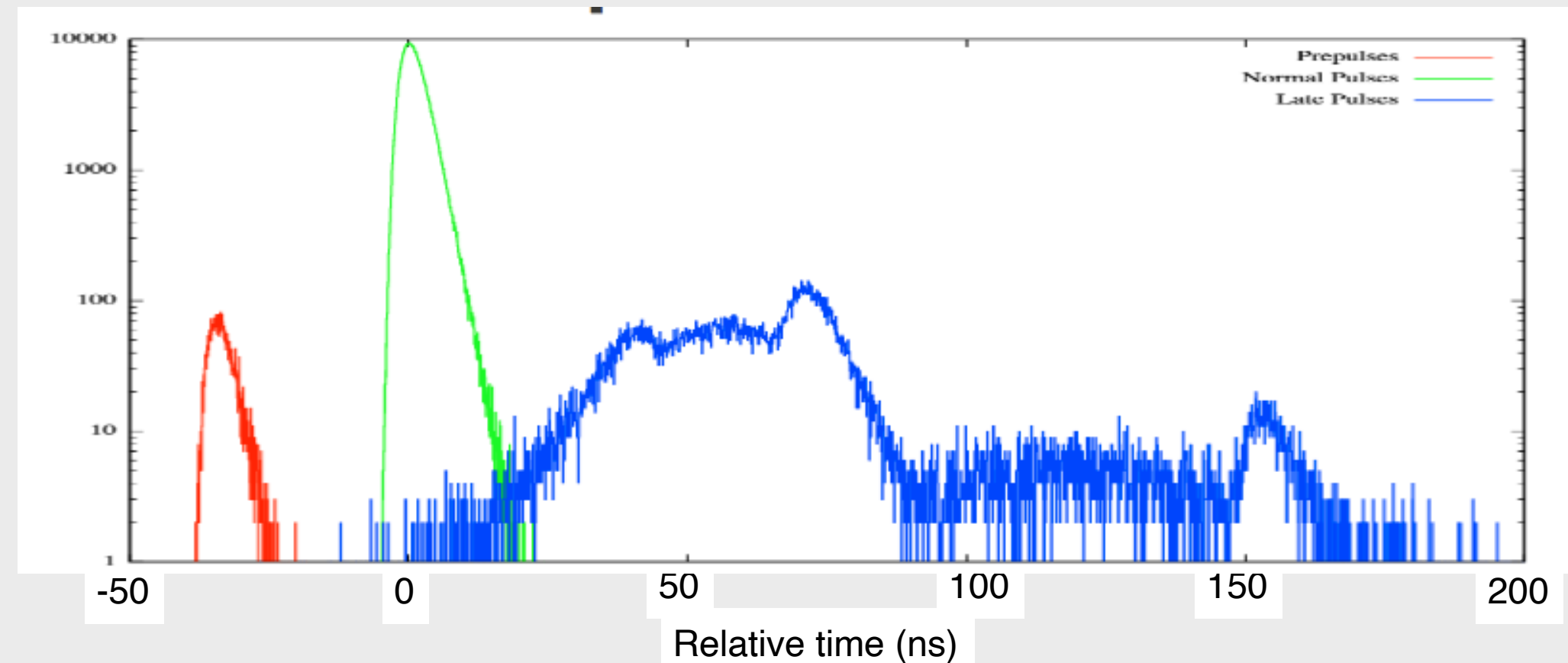


PMT Simulation

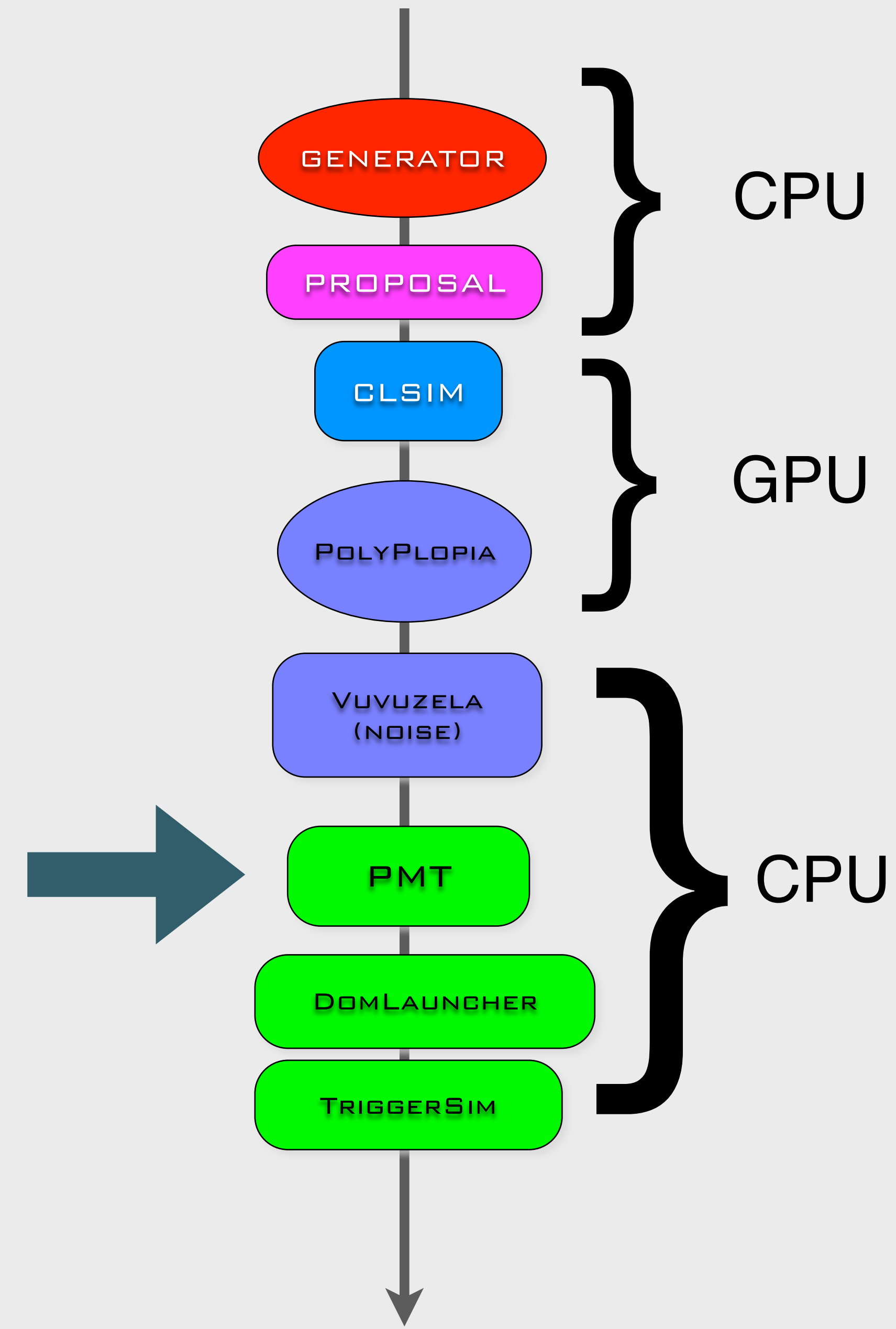
Pre-pulses: photoelectrons ejected from the first dynode,

Late pulses: electrons backscatter from dynode to cathode.

After-pulses: ionization of residual gases by electrons accelerated in the space between dynode.

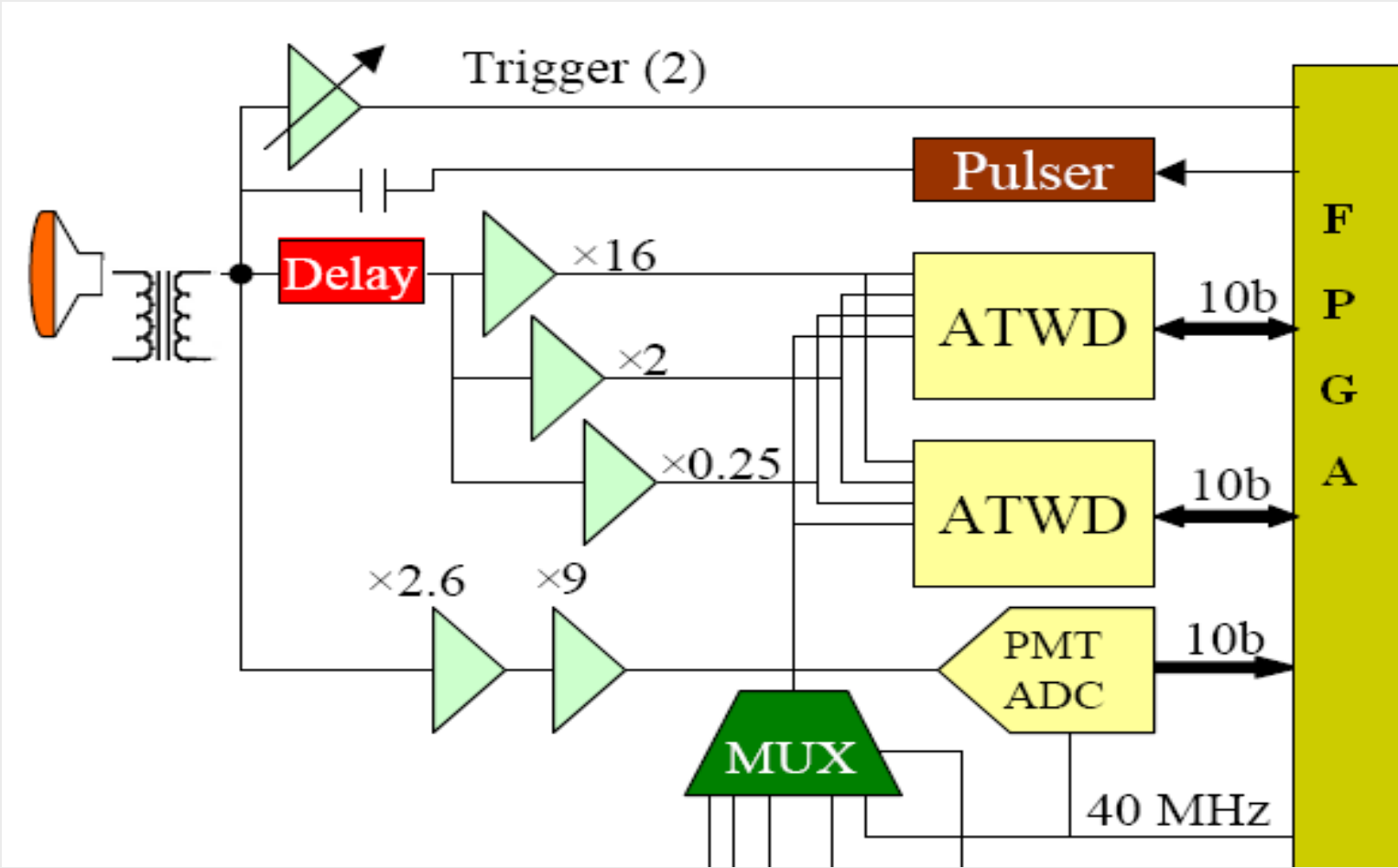
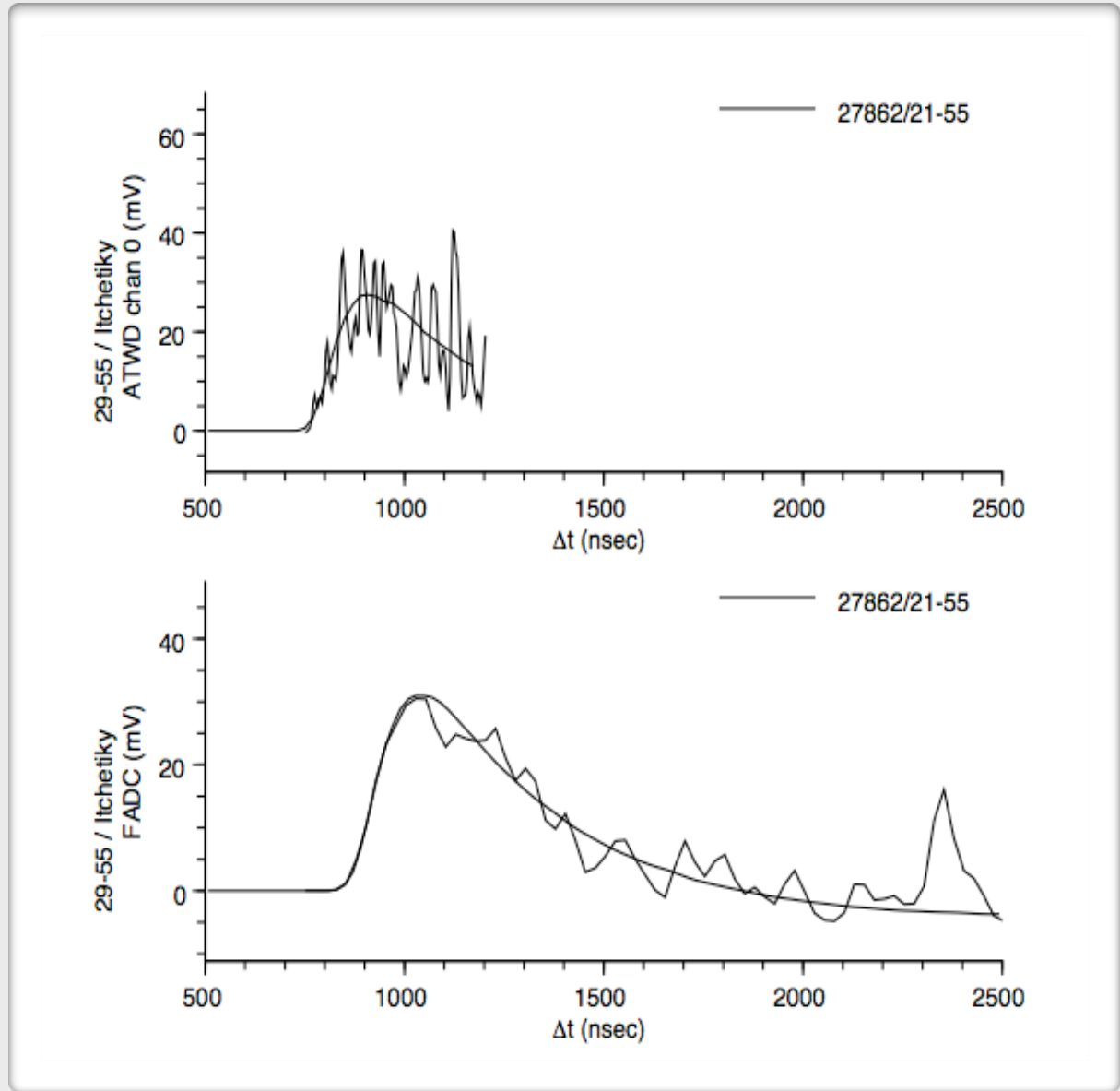
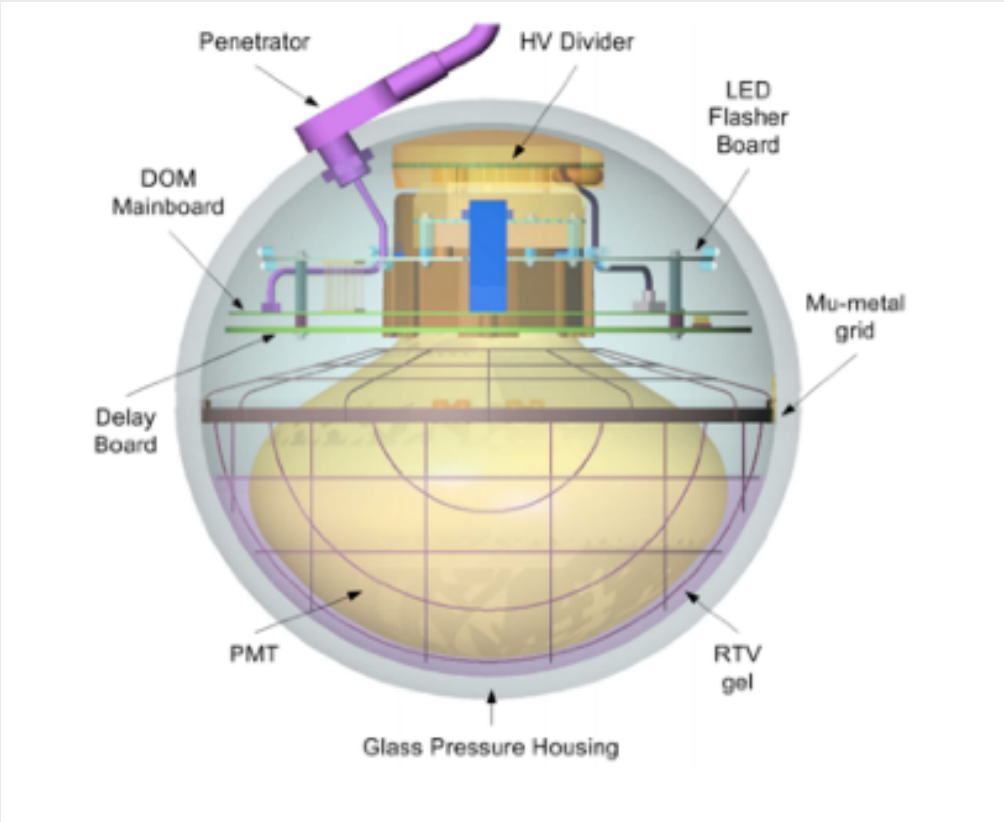


i³SIM

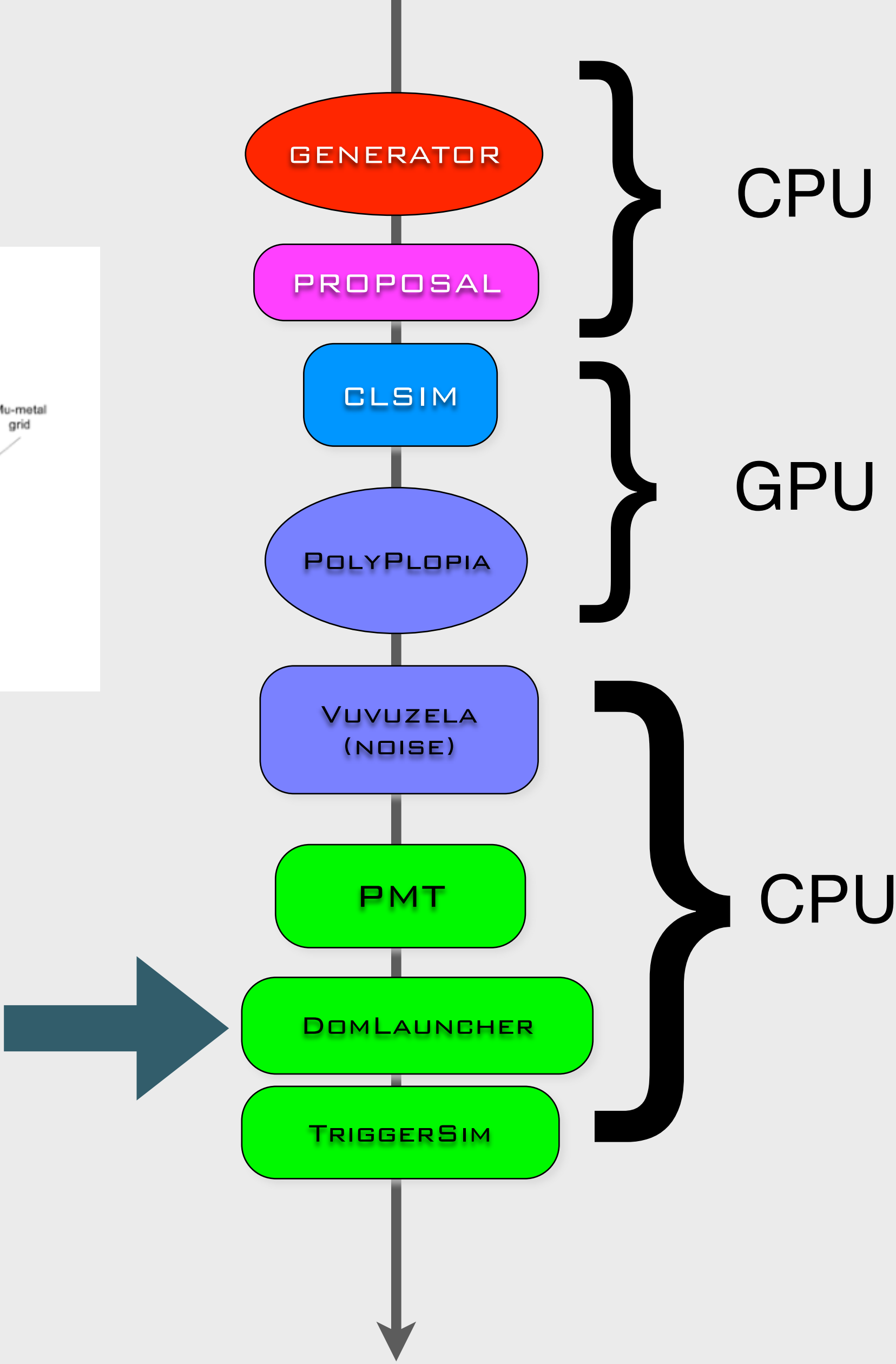


DOMLauncher: DOM electronics simulation

- Discriminator
- LC-logic
- Digitization
- Simulated effects
 - Electronic noise in the digitizers
 - Beacon launches (CPU triggered launches)
 - The FPGA Clock phase
 - RAPcal time uncertainty

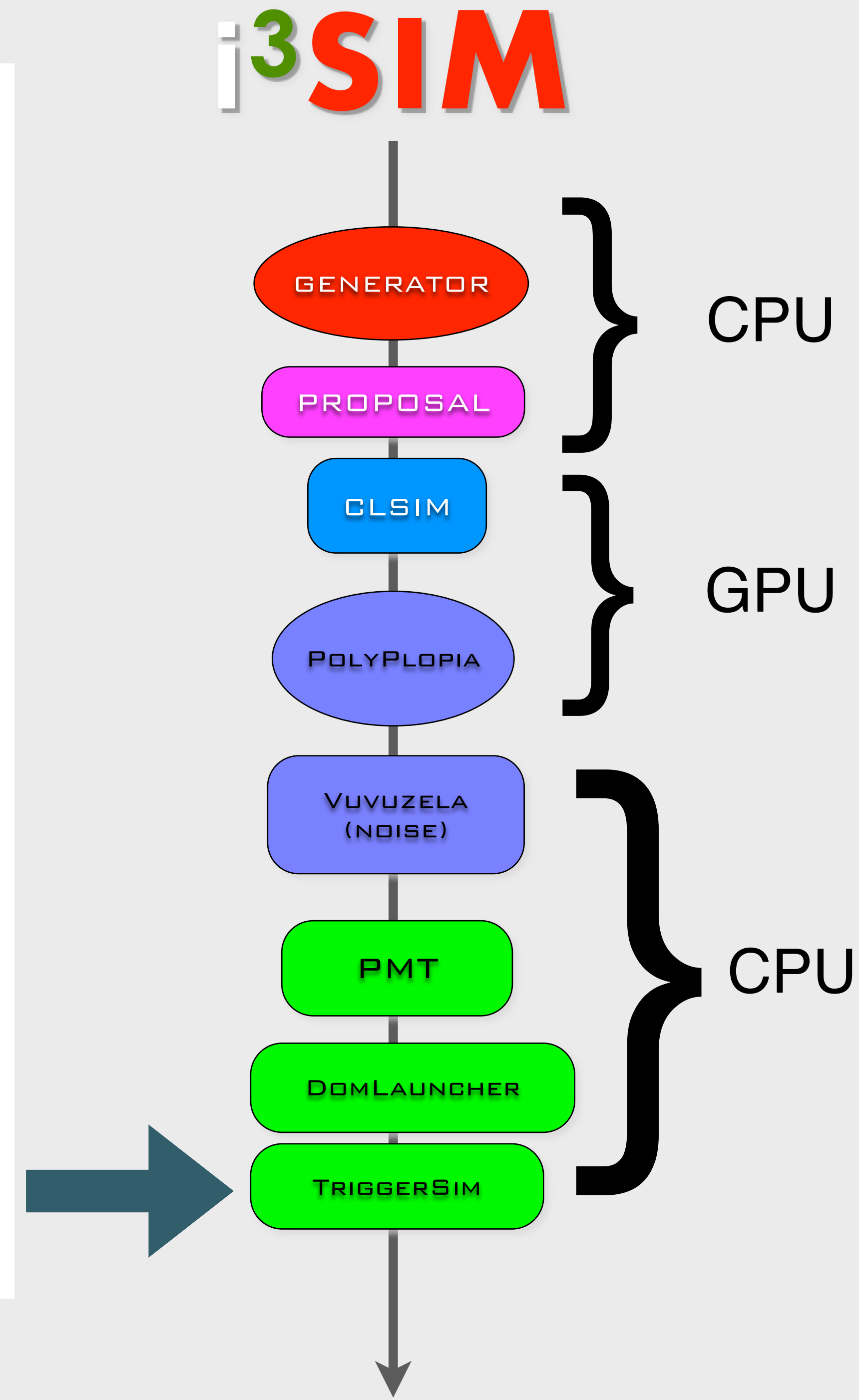


i3SIM



Trigger Types

- **Simple Multiplicity Trigger (SMT)**
 - N HLC hits or more in a time window
 - Example: InIce SMT8 with $N_{\text{hits}} \geq 8$ in $5 \mu\text{s}$
 - readout window around this captures early and late hits ($-4 \mu\text{s}$, $+6 \mu\text{s}$)
- **String** trigger (a.k.a. Cluster trigger in DAQ-land)
 - N HLC hits out of M DOMs on a string in a time window
 - Example: 5 hits from a run of 7 adjacent DOMs in a time window of 1500 ns
- **Volume** trigger (a.k.a Cylinder trigger in DAQ-land)
 - simple majority of HLC hits (SMT4) with volume element including one layer of strings around a center string
 - cylinder height is 5 DOM-layers (2 up and down from the selected DOM).
- **Slow Particle** trigger (SLOP)
 - slow-moving hits along a track
 - lengths of the order of $500\mu\text{s}$ and extending up to milliseconds
- **Fixed Rate** trigger, **Minimum Bias** trigger, **Calibration** trigger



Collection of scripts, tray segments ~~and IceProd modules~~ used in simulation production. Central place with standard segments for running simulation in both official production and private simulations.

- **Tray Segments:** IceTray meta-modules that contain several I3Modules with default parameters.
- **Scripts:** collection of python scripts used in simulation production

Tray Segments

\$I3_SRC/simprod-scripts/python/segments

GenerateNeutrinos

HybridPhotonicsCLSim

GenerateAirShowers

Polyplopia

GenerateCosmicRayMuons

PropagateMuons

GenerateIceTopShowers

DetectorSim

simprod-scripts

Scripts: `$I3_SRC/simprod-scripts/resources/scripts`
(run the individual pieces as broken down by production tasks)

```
$ python nugen.py -h
```

```
Usage: nugen.py [options]
```

```
Options:
```

```
-h, --help          show this help message and exit
```

```
--no-execute       boolean condition to execute
```

```
--outputfile=OUTPUTFILE
```

```
Output filename
```

```
--summaryfile=SUMMARYFILE
```

```
XMLSummary filename
```

```
--mjd=MJD           MJD for the GCD file
```

```
--seed=RNGSEED      RNG seed
```

```
--UseGSLRNG
```

```
...
```

simprod-scripts (SnowSuite)

Scripts: `$I3_SRC/simprod-scripts/resources/scripts`
(run the individual pieces as broken down by production tasks)

simprod-scripts/resources/scripts/SnowSuite/

- 1-process-Gen.py
- 2-Polyplopia.py
- 2-Propagate.py
- 3-Snowstorm.py
- 4-process-Weight.py
- demo
 - SimpleInjector.py
- iceprod2
 - iceprod_config_example.json
 - load_env
 - source_env
- jobs
 - condor_manager.py
- Snowstorm_Cfg.yml
- Snowstorm_FullSystematics.yml
- Snowstorm_Nominallce_Cfg.yml
- utils.py

Journal of Cosmology and Astroparticle Physics

Efficient propagation of systematic uncertainties from calibration to analysis with the SnowStorm method in IceCube

M.G. Aartsen¹⁶, M. Ackermann⁵⁴, J. Adams¹⁶, J.A. Aguilar¹², M. Ahlers²⁰, C. Alispach²⁶, B. Al Atoum⁴, K. Andeen³⁷, T. Anderson⁵¹, I. Ansseau¹² [+Show full author list](#)

Published 21 October 2019 • © 2019 IOP Publishing Ltd and Sissa Medialab

[Journal of Cosmology and Astroparticle Physics, Volume 2019, October 2019](#)

simprod-scripts

Example: Running scripts:

```
icecube@M16:~$ ssh submitter
[submitter]$ condor_submit -interactive 'request_gpus=1'
Submitting job(s).
1 job(s) submitted to cluster 120263704.
Waiting for job to start..
Welcome to slot1@gtx-00.icecube.wisc.edu!

[gtx-00]$ cd $CONDOR_SCRATCH_DIR
[gtx-00]$ /cvmfs/icecube.opensciencegrid.org/py3-v4.2.0/icetray-env icetray/v1.3.3
*****
*                                                                 *
*              W E L C O M E  t o  I C E T R A Y                *
*                                                                 *
*      Version icetray.stable      git:f5d21802                 *
*                                                                 *
*      You are welcome to visit our Web site                     *
*      http://icecube.umd.edu                                       *
*                                                                 *
*****
[gtx-00]$ python $I3_BUILD/simprod-scripts/resources/scripts/nugen.py \
          --outputfile nutau.i3 --nevents 100 \
          --seed=123 --procnum 0  --nproc=1 \
          --FromEnergy 1e5 --ToEnergy 1e6 --NuFlavor NuTau --UseGSLRNG

[gtx-00]$ dataio-shovel nutau.i3
```

simprod-scripts

Example: Running scripts:

```
[gtx-00]$ dataio-shovel nutau.i3
```

I3 Data Shovel

```
- Press '?' for help
```

Name	Type	Bytes
I3MCTree_preMuonProp	TreeBase::Tree<I3Particle, I3ParticleID, i3hash<I3...	422
I3MCWeightDict	I3Map<__cxx11::string, double>	1400
NuGPrimary	I3Particle	150

Key: 3/3
Frame: 3/101 (2%)
Stop: DAQ
Run/Event: (n/a)
SubEvent: (n/a)

StartTime: (n/a)

Duration: (n/a)

20

40

I000

simprod-scripts

Example: Running scripts:

```
[gtx-00]$ python $I3_BUILD/simprod-scripts/resources/scripts/clsim.py \  
    --gcdfile /cvmfs/icecube.opensciencegrid.org/data/GCD/ \  
        GeoCalibDetectorStatus_2020.Run134142.Pass2_V0.i3.gz gcdfile.i3.gz \  
    --inputfilelist nutau.i3 --outputfile mcpe.i3 \  
    --seed 123 --procnum 0 --nproc 1 --no-RunMPHitFilter \  
    --UseGPUs --UseGSLRNG
```

```
[gtx-00]$ dataio-shovel mcpe.i3
```

```
[gtx-00]$ python $I3_BUILD/simprod-scripts/resources/scripts/detector.py \  
    --gcdfile /cvmfs/icecube.opensciencegrid.org/data/GCD/ \  
        GeoCalibDetectorStatus_2020.Run134142.Pass2_V0.i3.gz gcdfile.i3.gz \  
    --inputfile mcpe.i3 --outputfile det.i3 \  
    --seed 123 --procnum 0 --nproc 1 --RunID 123 --UseGSLRNG
```

simprod-scripts

Example: Running scripts:

```
[gtx-00]$ dataio-shovel mcpses.i3
```

I3 Data Shovel			Press '?' for help
Name	Type	Bytes	
I3MCPESeriesMap	I3Map<OMKey, vector<I3MCPE> >	41	
I3MCPESeriesMapParticleIDMap	I3Map<OMKey, map<I3ParticleID, vector<unsigned int...	41	
I3MCTree	TreeBase::Tree<I3Particle, I3ParticleID, i3hash<I3...	6878	
I3MCTree_preMuonProp	TreeBase::Tree<I3Particle, I3ParticleID, i3hash<I3...	1538	
I3MCTree_preMuonProp_RNGState	I3GSLRandomServiceState	85	
I3MCWeightDict	I3Map<__cxx11::string, double>	1400	
MMCTrackList	I3Vector<I3MMCTrack>	40	
NuGPrimary	I3Particle	150	

Key: 1/8 **StartTime:** (n/a)
Frame: 3/102 (2%) **Duration:** (n/a)
Stop: DAQ
Run/Event: (n/a) 20 40
SubEvent: (n/a)

simprod-scripts

Exercise: Running scripts:

```
[gtx-00]$ dataio-shovel det.i3
```

I3 Data Shovel

Press '?' for help

Name	Type	Bytes
BeaconLaunches	I3Map<OMKey, vector<I3DOMLaunch> >	46
I3EventHeader	I3EventHeader	99
I3MCPESeriesMap	I3Map<OMKey, vector<I3MCPE> >	113286
I3MCPESeriesMapParticleIDMap	I3Map<OMKey, map<I3ParticleID, vector<unsigned int...>	36649
I3MCPESeriesMapWithoutNoise	I3Map<OMKey, vector<I3MCPE> >	109543
I3MCPulseSeriesMap	I3Map<OMKey, vector<I3MCPulse> >	82000
I3MCPulseSeriesMapParticleI...	I3Map<OMKey, map<I3ParticleID, vector<unsigned int...>	40743
I3MCPulseSeriesMapPrimaryIDMap	I3Map<OMKey, map<I3ParticleID, vector<unsigned int...>	27299
I3MCTree	TreeBase::Tree<I3Particle, I3ParticleID, i3hash<I3...	10730
I3MCTree_preMuonProp	TreeBase::Tree<I3Particle, I3ParticleID, i3hash<I3...	422
I3MCTree_preMuonProp_RNGState	I3GSLRandomServiceState	85
I3MCWeightDict	I3Map<__cxx11::string, double>	1400
I3TriggerHierarchy	I3Tree<I3Trigger>	792
I3Triggers	I3Tree<I3Trigger>	414
IceTopRawData	I3Map<OMKey, vector<I3DOMLaunch> >	46
InIceRawData	I3Map<OMKey, vector<I3DOMLaunch> >	44640
MMCTrackList	I3Vector<I3MMCTrack>	2864
NuGPrimary	I3Particle	150
TimeShift	I3PODHolder<double>	36

simprod-scripts

Exercise: Running scripts:

```
[gtx-00]$ python $I3_BUILD/simprod-scripts/resources/scripts/corsika.py \  
            --nshowers 10000 --outputfile corsika_bg.i3 --seed 1234 \  
            --CORSIKAsed=123 --ranpri 2 \  
            --corsikaVersion v6960-5comp \  
            --corsikaName dcorsika --UseGSLRNG \  
            --skipoptions compress  
  
[gtx-00]$ dataio-shovel corsika_bg.i3  
  
[gtx-00]$ python $I3_BUILD/simprod-scripts/resources/scripts/polyplopia.py \  
            --gcdfile gcdfile.i3.gz \  
            --inputfile mcpes.i3 --outputfile merged_pes.i3 \  
            --seed 1234 \  
            --backgroundfile corsika_bg.i3 --mctype NuTau \  
            --UseGSLRNG  
  
[gtx-00]$ python $I3_BUILD/simprod-scripts/resources/scripts/detector.py \  
            --gcdfile gcdfile.i3.gz \  
            --inputfile merged_pes.i3 --outputfile det_wcoinc.i3 \  
            --seed 123 --RunID 123 --UseGSLRNG  
  
[gtx-00]$ dataio-shovel det_wcoinc.i3
```

simprod-scripts

Exercise: Running scripts:

```
[gtx-00]$ dataio-shovel corsika_bg.i3
```

I3 Data Shovel		Press '?' for help
Name	Type	Bytes
CorsikaInteractionHeight	I3PODHolder<double>	36
CorsikaWeightMap	I3Map<__cxx11::string, double>	484
I3CorsikaInfo	I3CorsikaInfo	109
I3MCTree	TreeBase::Tree<I3Particle, I3ParticleID, i3hash<I3...	546
I3MCTree_preSampling	TreeBase::Tree<I3Particle, I3ParticleID, i3hash<I3...	546

simprod-scripts

Exercise: Running scripts:

```
[gtx-00]$ dataio-shovel merged_pes.i3
```

I3 Data Shovel

Press '?' for help

Name	Type	Bytes
BackgroundI3MCPESeriesMap	I3Map<OMKey, vector<I3MCPE> >	41
BackgroundI3MCPESeriesMapPa...	I3Map<OMKey, map<I3ParticleID, vector<unsigned in...	41
BackgroundI3MCTree	TreeBase::Tree<I3Particle, I3ParticleID, i3hash<I...	32
BackgroundI3MCTreePEcounts	I3Map<unsigned int, unsigned int>	47
BackgroundI3MCTree_preMuonProp	TreeBase::Tree<I3Particle, I3ParticleID, i3hash<I...	32
BackgroundI3MCTree_preMuonP...	I3GSLRandomServiceState	85
BackgroundMMCTrackList	I3Vector<I3MMCTrack>	40
I3MCPESeriesMap	I3Map<OMKey, vector<I3MCPE> >	41
I3MCPESeriesMapParticleIDMap	I3Map<OMKey, map<I3ParticleID, vector<unsigned in...	41
I3MCTree	TreeBase::Tree<I3Particle, I3ParticleID, i3hash<I...	2902
I3MCTree_preMuonProp	TreeBase::Tree<I3Particle, I3ParticleID, i3hash<I...	422
I3MCTree_preMuonProp_RNGState	I3GSLRandomServiceState	85
I3MCWeightDict	I3Map<__cxx11::string, double>	1424
MMCTrackList	I3Vector<I3MMCTrack>	40
NuGPrimary	I3Particle	150
PhotonSeriesMap	I3Map<ModuleKey, I3Vector<I3CompressedPhoton> >	53
PolyplopiaInfo	I3Map<__cxx11::string, int>	135
PolyplopiaPrimary	I3Particle	150
SignalI3MCPEs	I3Map<OMKey, vector<I3MCPE> >	41
SignalI3MCTree	TreeBase::Tree<I3Particle, I3ParticleID, i3hash<I...	2902

Weighting

<https://docs.icecube.aq/simweights/main/>

TUTORIALS

Triggered CORSIKA Tutorial

Neutrino Generator Tutorial

IceTop CORSIKA Tutorial

HOW-TO GUIDES

How to Combine Datasets

How to Use NuFlux Models

How to Calculate Effective Areas

How to Calculate Effective Livetime

Adding Additional Parameters to Fluxes

How to Weight Without Using TableIO

EXPLANATIONS

Weighting in Monte Carlo Simulation

Weights in IceCube Simulation

Spatial Distributions

How SimWeight Reads Files

Reading Neutrino-Generator Files

Units in SimWeights

Neutrino Generator Tutorial

The easiest way to use simweights is to book your data to hdf5files using tableio.

```
#!/usr/bin/env python3

# SPDX-FileCopyrightText: © 2022 the SimWeights contributors
#
# SPDX-License-Identifier: BSD-2-Clause

from pathlib import Path

from I3Tray import I3Tray
from icecube import hdfwriter, simclasses

FILE_DIR = Path("/data/sim/IceCube/2016/filtered/level2/neutrino-generator/21217/0000000-0000999")
files = sorted(str(f) for f in FILE_DIR.glob("Level2_IC86.2016_NuMu.021217.00000*.i3.zst"))

tray = I3Tray()
tray.Add("I3Reader", FileNameList=files)
tray.Add(
    hdfwriter.I3HDFWriter,
    SubEventStreams=["InIceSplit"],
    keys=["PolyplopiaPrimary", "I3MCWeightDict"],
    output="Level2_IC86.2016_NuMu.021217.hdf5",
)

tray.Execute()
```

Distributed Computing

- IceCube Specific scheduler
- Used by Simulation Production to create official datasets
- Describe jobs to run using JSON
- Handles File transfer to data warehouse
- Web Interface /API
- Data provenance
 - Configuration
 - which software, what versions,
 - when/where it ran, ...
- Dataset submission
 - Monitor job status, resource usage
 - Retry failed jobs - resubmit with different requirements

<https://github.com/icecube/simprod-templates/>

IceProd

DatasetsProfileLogout

Dataset 21889 Details

View ConfigEdit Config

Settings

description:

ME IC86.2016 Triggered CORSIKA-in-ice 5-component model Sibyll2.3c (CORSIKA 77401) with weighted spectrum of E^-2.6, using Spice3.2 CIsim. Angular range of 0deg < theta < 89.99deg and energy range of 3e4GeV < Eprim < 1e6GeV. DOM oversize = 5

jobs_submitted:

100000

tasks_submitted:

300000

tasks_per_job:

3

group:

simprod

dataset_id:

5deb19300c1411eca9f2141877284d92

dataset:

21889

status:

processing

start_date:

2021-09-02T17:36:55.621073

username:

kmeagher

priority:

0.5

Update

debug:

False

jobs_immutable:

False

ResetHard ResetSuspendTruncateDelete Logs

Jobs

processing

1

complete

99999

Tasks

complete

299997

Task Status by Task Name

Name	Type	Waiting	Queued	Running	Complete	Error
server	GPU	0	0	0	99999	0
filtering	CPU	0	0	0	99999	0
L1L2	CPU	0	0	0	99999	0

Completion Statistics

Name	Avg/stddev (hours)	Max/min (hours)	Eff
server	0.31 / 0.14	3.00 / 0.17	84%
filtering	3.92 / 1.55	45.14 / 1.49	95%
L1L2	3.05 / 1.04	35.38 / 1.65	95%

Submit New Dataset

Submit Dataset like Current

IceProd 2.5.7

IceProd

<https://github.com/icecube/simprod-templates/>

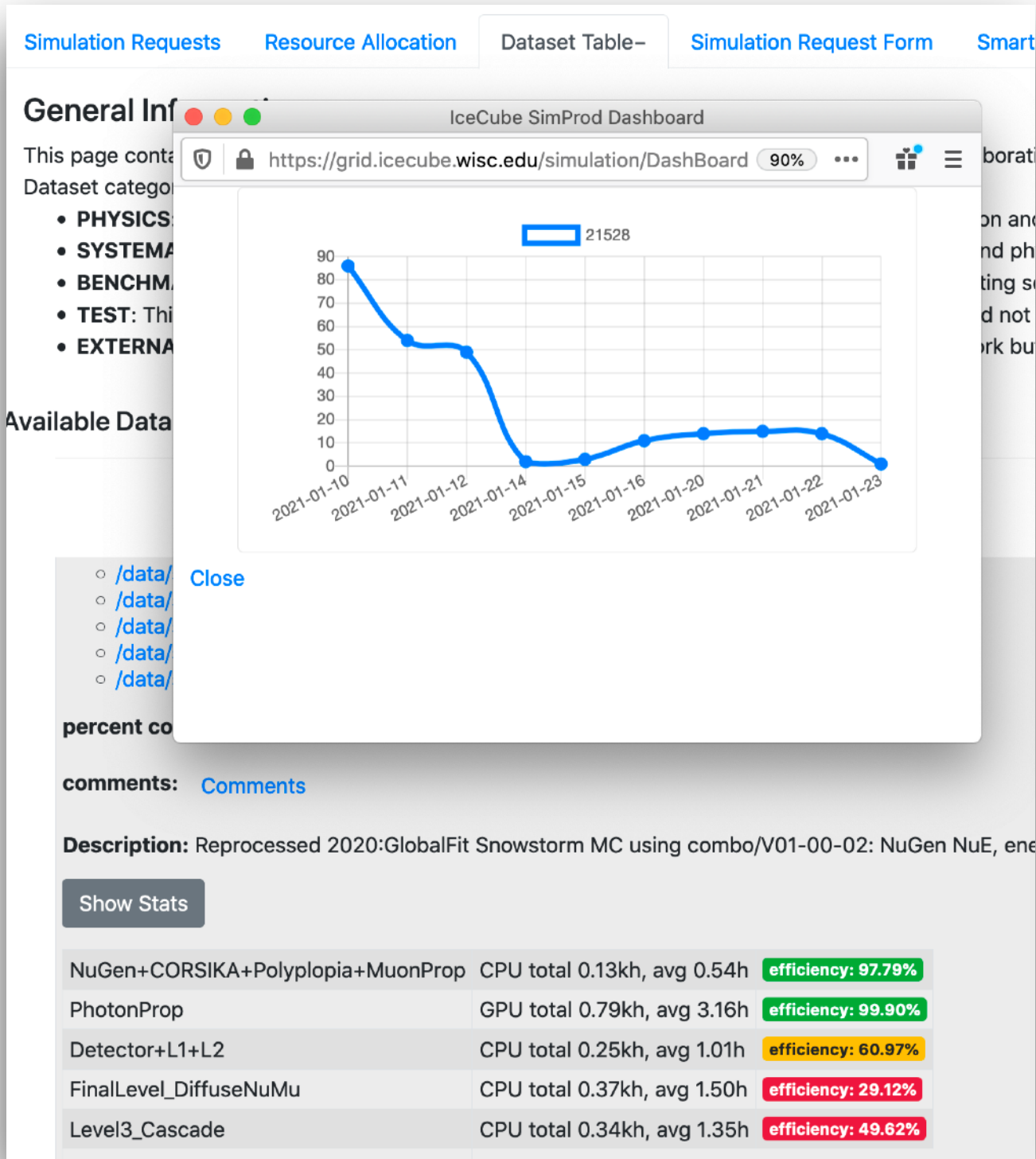
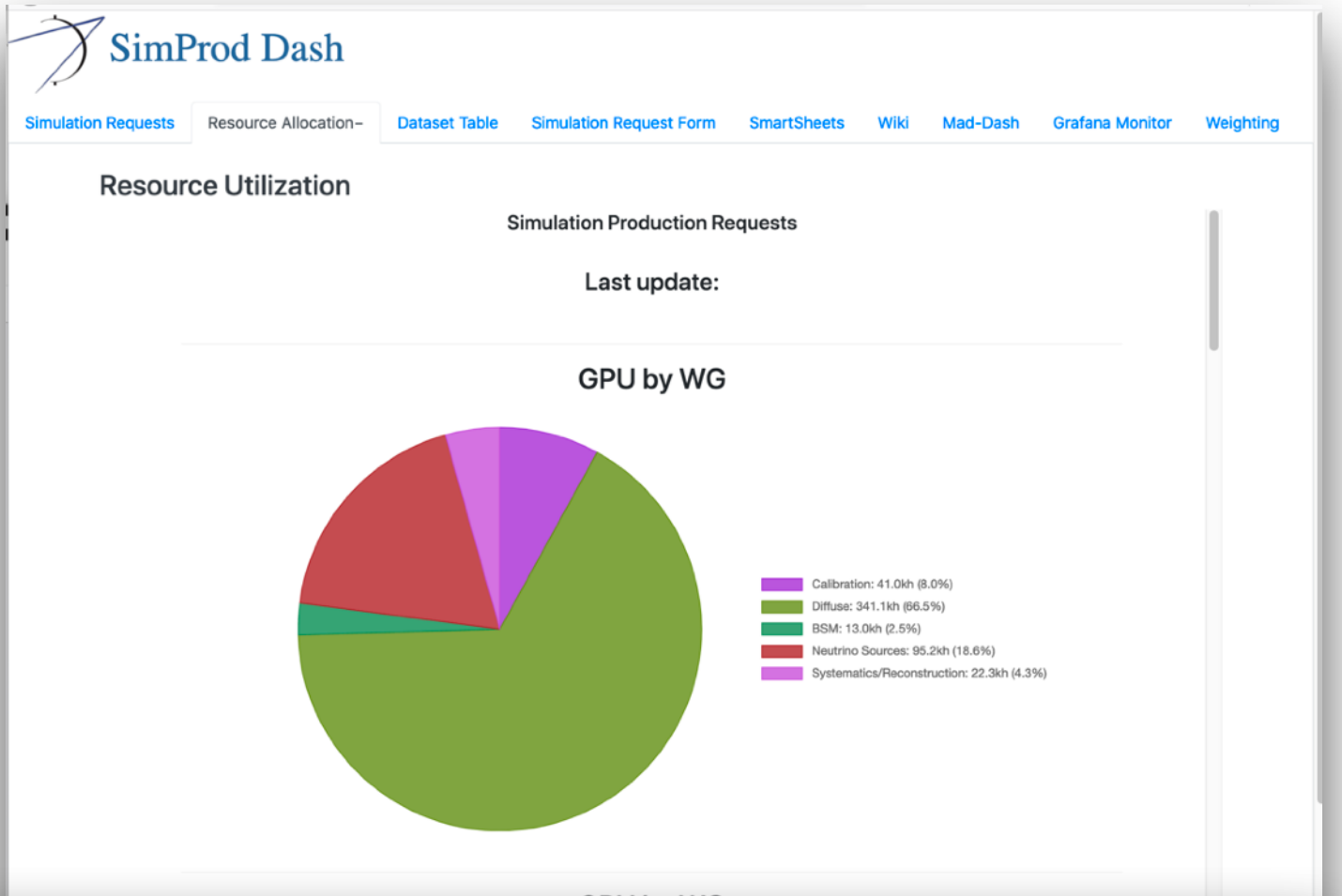
The image shows a code editor interface. On the left is a file explorer with a 'Files' tab. It shows a directory structure with folders like 'corsika', 'filtering', 'genie', 'icetop', 'lepton-injector', and 'nugen'. The 'corsika' folder is expanded, showing 'Triggered-CORSIKA.json' selected. On the right is the code editor for 'Triggered-CORSIKA.json'. The code is a JSON configuration for a server task. It includes fields for 'difplus', 'tasks', 'requirements', 'name', 'parameters', 'depends', 'batchsys', 'condor', 'trays', and 'modules'. The 'tasks' array contains a single task named 'server' with various requirements and parameters. The 'trays' array contains a single tray named 'corsika' with its own parameters and modules.

```
1  {
2    "difplus":null,
3    "tasks":[
4      {
5        "requirements":{
6          "gpu":1,
7          "cpu":8,
8          "disk":8,
9          "time":2.5,
10         "memory":11
11       },
12       "name":"server",
13       "parameters":{},
14       "depends":[],
15       "batchsys":{
16         "condor":{
17           "requirements":"!regexp(\"^Louisiana\", GLIDEIN_Site) && !regexp(\"^Portland\", GLIDEIN_Site) && !regex
18         }
19       },
20       "trays":[
21         {
22           "name":"corsika",
23           "parameters":{},
24           "modules":[
25             {
```

Simprod DashBoard

<https://simprod.icecube.wisc.edu>

- Catalog of official MC datasets
- Simulation Requests
- Dataset and resource Monitoring



General Information

This page contains information about centrally managed simulation production for the IceCube Collaboration. This page is dynamically updated from SmartSheets and IceProd.

Dataset categories

- **PHYSICS:** This category corresponds to datasets that simulate standard detector configuration and nominal parameters such as Ice scattering and absorption coefficients, DOM efficiency, neutrino x-sections, hole-ice, etc.
- **SYSTEMATICS:** This corresponds to datasets that include systematic variations in detector and physics parameters.
- **BENCHMARK:** This category corresponds to datasets that are meant for checking and validating software. These should *not* be used for data analyses.
- **TEST:** This is a dataset that is meant to test software and/or production framework and should not be used for analyses.
- **EXTERNAL:** This is a dataset that was generated outside of the standard production framework but is catalogued in our database. Such datasets are not maintained by the production team and are provided *as is*.

Available Datasets

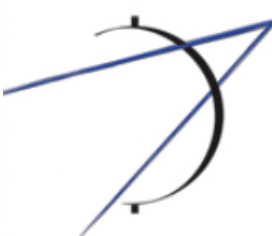
Geometry	Year	Generator	Flavor	Category	Spectrum	Energy Range	Dataset	Progress
IC86	2020	CORSIKA-in-ice	5-component model	PHYSICS	E^-2.0	600 GeV-1e8 GeV	21521	96.17%
IC86	2020	neutrino-generator	NuMu	SYSTEMATICS	E^-1.5	1e2 GeV - 1e4 GeV	21525	99.96%
IC86	2020	neutrino-generator	NuMu	SYSTEMATICS	E^-1.5	1e4 GeV - 1e6 GeV	21526	95.40%
IC86	2020	neutrino-generator	NuMu	SYSTEMATICS	E^-1.0	1e6 GeV - 1e8 GeV	21527	99.97%
IC86	2020	neutrino-generator	NuE	SYSTEMATICS	E^-1.5	1e2 to 1e4 GeV	21528	99.60%
IC86	2020	neutrino-generator	NuE	SYSTEMATICS	E^-1.5	1e4 to 1e6 GeV	21529	98.43%

SimProd Requests

<https://simprod.icecube.wisc.edu/submit>

Need new simulations?

- 1. Discuss with your WG
- 2. Submit a request
 - a. If possible provide estimates of resource utilization
 - b. If self-generated, test and benchmark
- 3. In case of competing requests, priority will be evaluated and determined by WG tech leads

 **SimProd Dash**

[Logout](#)

[Simulation Requests](#) [Resource Allocation](#) [Dataset Table](#)

Simulation Request Form

[Wiki](#) [Mad-Dash](#) [Grafana Monitor](#)

[Weighting](#)

Simulation Request Form

Instructions: ▾

Simulation Request Form

Specify the details of what you need.

Requester

juancarlos@icecube.wisc.edu

Working Group

GENERAL ▾

Title/Description

Analysis

Date needed by

Estimated req. CPU

0.0 ▴ ▾

Estimated req. GPU

0.0 ▴ ▾

Estimated req. storage

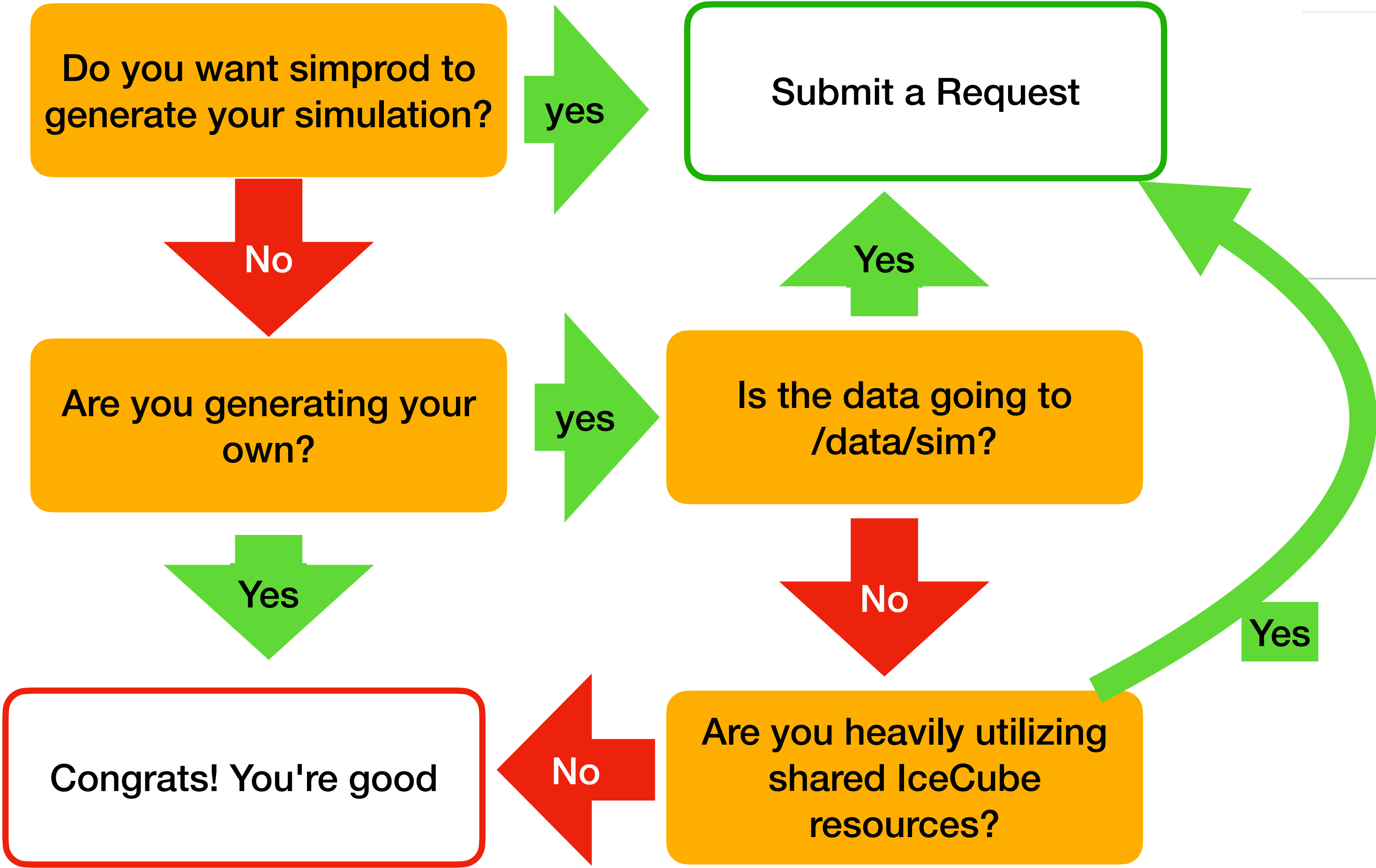
0.0 ▴ ▾

Documentation

Additional comments

SimProd Requests

<https://simprod.icecube.wisc.edu/submit>



Simulation Request Form

Instructions: 

Simulation Request Form

Specify the details of what you need.

Requester	juancarlos@icecube.wisc.edu
Working Group	GENERAL 
Title/Description	
Analysis	
Date needed by	
Estimated req. CPU	0.0  
Estimated req. GPU	0.0  
Estimated req. storage	0.0  
Documentation	
Additional comments	

SimProd Requests

Why should I make a simprod request if I'm generating my own simulations?

- For fairness: We are all sharing computing resources and this helps to allocate and keep track of usage.
- To share your simulation with others
 - Datasets associated with simreqs are added to global catalogue and accessible to all
 - The Simulation Requests can work as a meta-dataset or collection of datasets associated with a particular study.
- To take advantage of monitoring tools, etc.

SimProd Requests

<https://grid2.icecube.wisc.edu/submit>

Requirements for SimProd Datasets

If you are generating official simulation production,

1. Make sure you have submitted an official [simprod request](#).
2. Run some small samples (≤ 100 jobs) to test and benchmark your dataset config
3. Provide dataset numbers associated with request to simprod coordination once submitted

SimProd Requests

<https://grid2.icecube.wisc.edu/submit>

Add key information in ***steering parameters*** of your IceProd JSON config (even if they are not explicitly used in the modules)

- 1. **"energy_spectrum"**: (e.g. "E^-2.0")
- 2. **"energy_range"**: (e.g. "1e4GeV-1e8GeV")
- 3. **"IceModel"**: (e.g. "spice_ftp-v2") or **"IceModelLocation"**: (full path)
- 4. **"generator"**: ("neutrino-generator", "LeptonInjector", "CORSIKA-in-Ice", "CORSIKA-ice-top", "GENIE", "MuonGun")
- 5. **"flavor"**: (e.g. "5-component", "NuMu", "Muons")
- 6. **"year"**: (e.g. 2020)
- 7. **"geometry"**: (e.g. "IC86")



SimProd Dash

[Simulation Requests](#)

[Resource Allocation](#)

[Dataset Table](#)

[Simulation Request Form](#)

[Wiki](#)

[Mad-Dash](#)

[Grafana Monitor](#)

[Weighting](#)

Simulation Request Form

Instructions: ▾

Simulation Request Form

Specify the details of what you need.

Requester

juancarlos@icecube.wisc.edu

Working Group

GENERAL ▾

Title/Description

Analysis

Date needed by

Estimated req. CPU

0.0 ▴ ▾

Estimated req. GPU

0.0 ▴ ▾

Estimated req. storage

0.0 ▴ ▾

Documentation

Additional comments

45

SimProd Requests

<https://simprod.icecube.wisc.edu/submit>

Add key information in ***steering parameters*** of your IceProd JSON config (even if they are not explicitly used in the modules)

- 1. **"energy_spectrum"**: (e.g. "E^-2.0")
- 2. **"energy_range"**: (e.g. "1e4GeV-1e8GeV")
- 3. **"IceModel"**: (e.g. "spice_ftp-v2") or **"IceModelLocation"**: (full path)
- 4. **"generator"**: ("neutrino-generator", "LeptonInjector", "CORSIKA-in-Ice", "CORSIKA-ice-top", "GENIE", "MuonGun")
- 5. **"flavor"**: (e.g. "5-component", "NuMu", "Muons")
- 6. **"year"**: (e.g. 2020)
- 7. **"geometry"**: (e.g. "IC86")



SimProd Dash

[Simulation Requests](#)

[Resource Allocation](#)

[Dataset Table](#)

[Simulation Request Form](#)

[Wiki](#)

[Mad-Dash](#)

[Grafana Monitor](#)

[Weighting](#)

Simulation Request Form

Instructions: ▾

Simulation Request Form

Specify the details of what you need.

Requester

juancarlos@icecube.wisc.edu

Working Group

GENERAL ▾

Title/Description

Analysis

Date needed by

Estimated req. CPU

0.0 ▴ ▾

Estimated req. GPU

0.0 ▴ ▾

Estimated req. storage

0.0 ▴ ▾

Documentation

Additional comments

46

Interactive SimProd Portal Demo

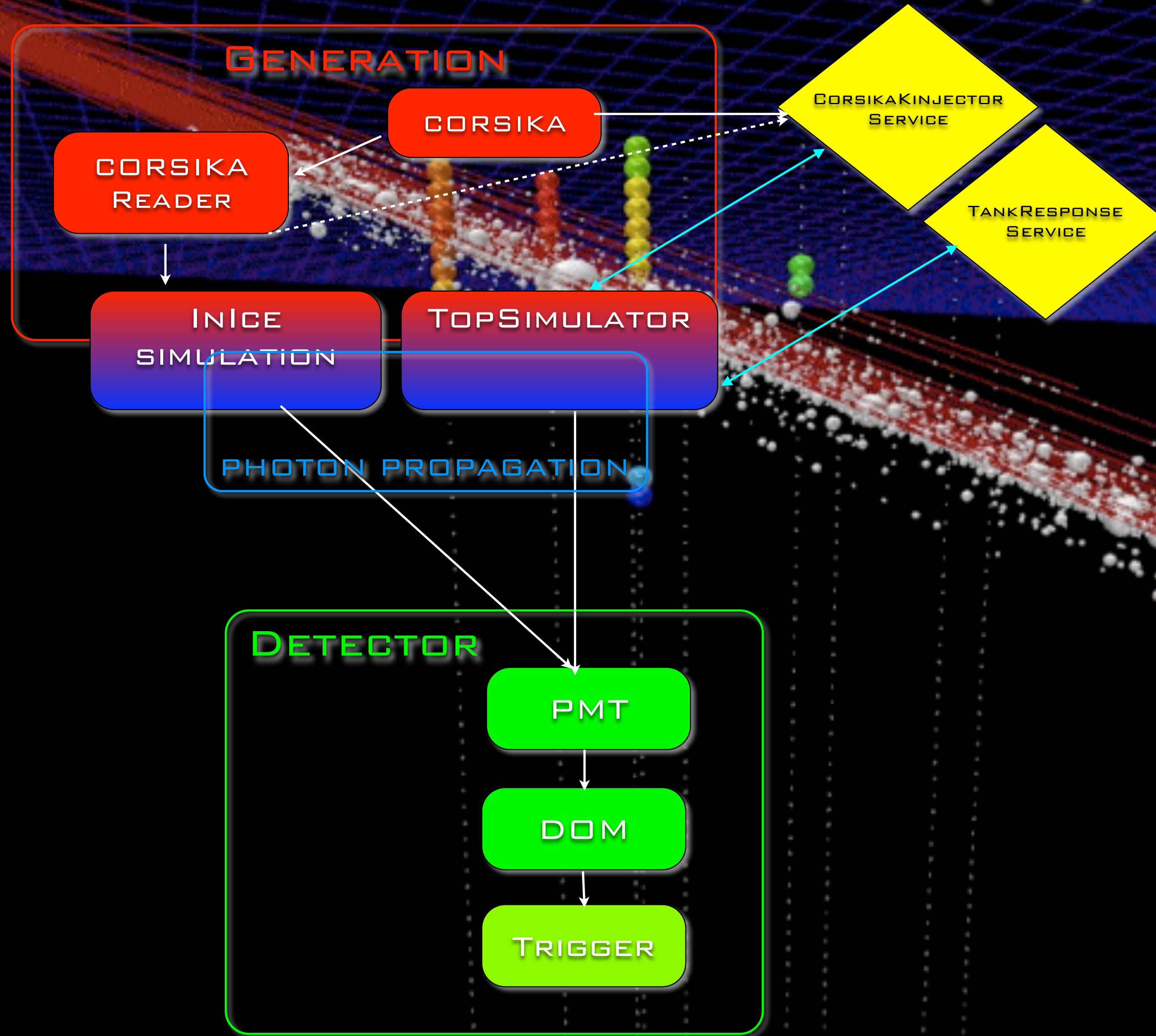
<http://simprod.icecube.wisc.edu>

More on Simulation

- GitHub repository: <https://docs.icecube.aq/icetray/main/>
- SimProd Portal: <http://grid2.icecube.wisc.edu>
- IceProd Docs: <https://wipacrepo.github.io/iceprod/>
- JSON Templates: <https://github.com/icecube/simprod-templates/>
- Wiki: http://wiki.icecube.wisc.edu/index.php/Simulation_Production
- Weighting: <https://docs.icecube.aq/simweights/main/>
- SLACK: [#simulation](#)

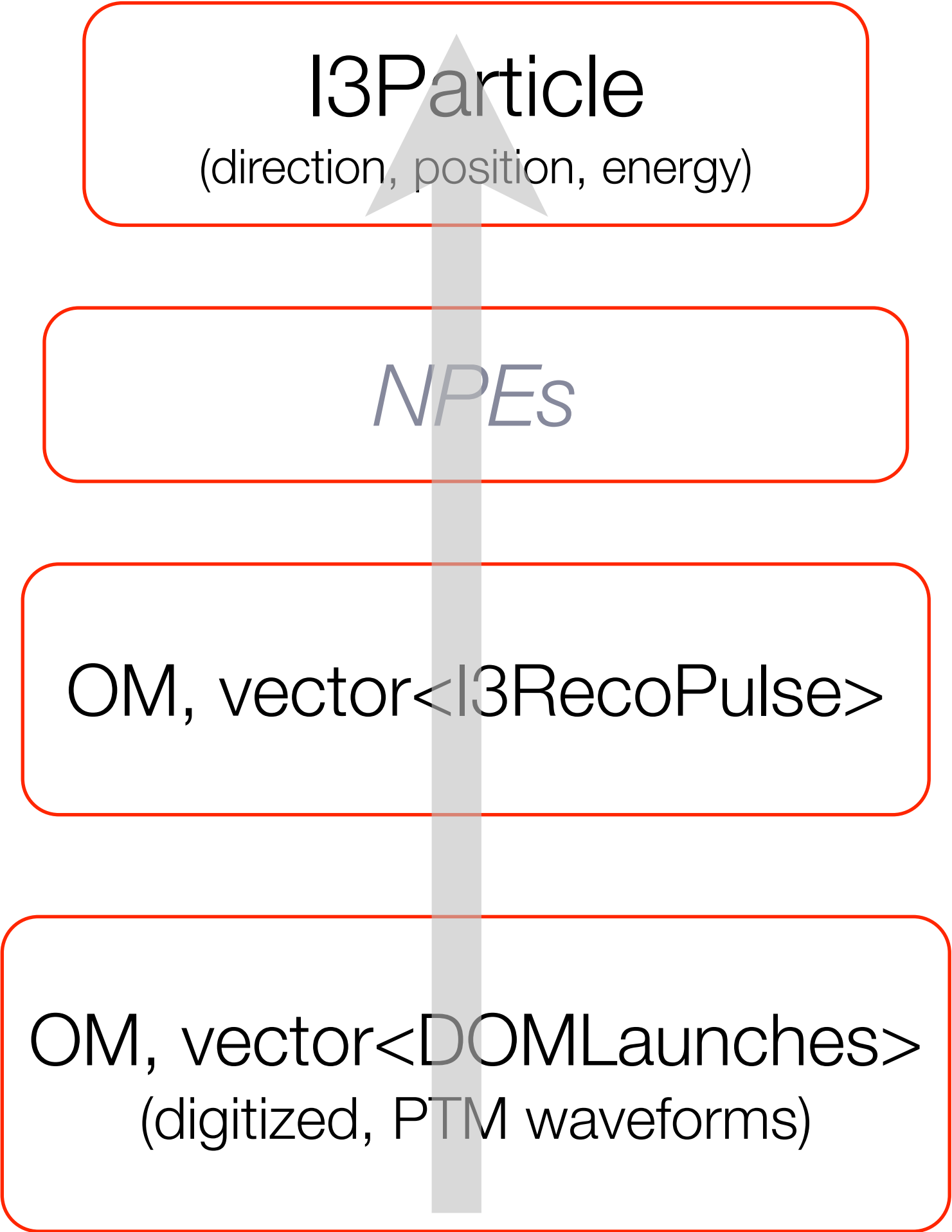
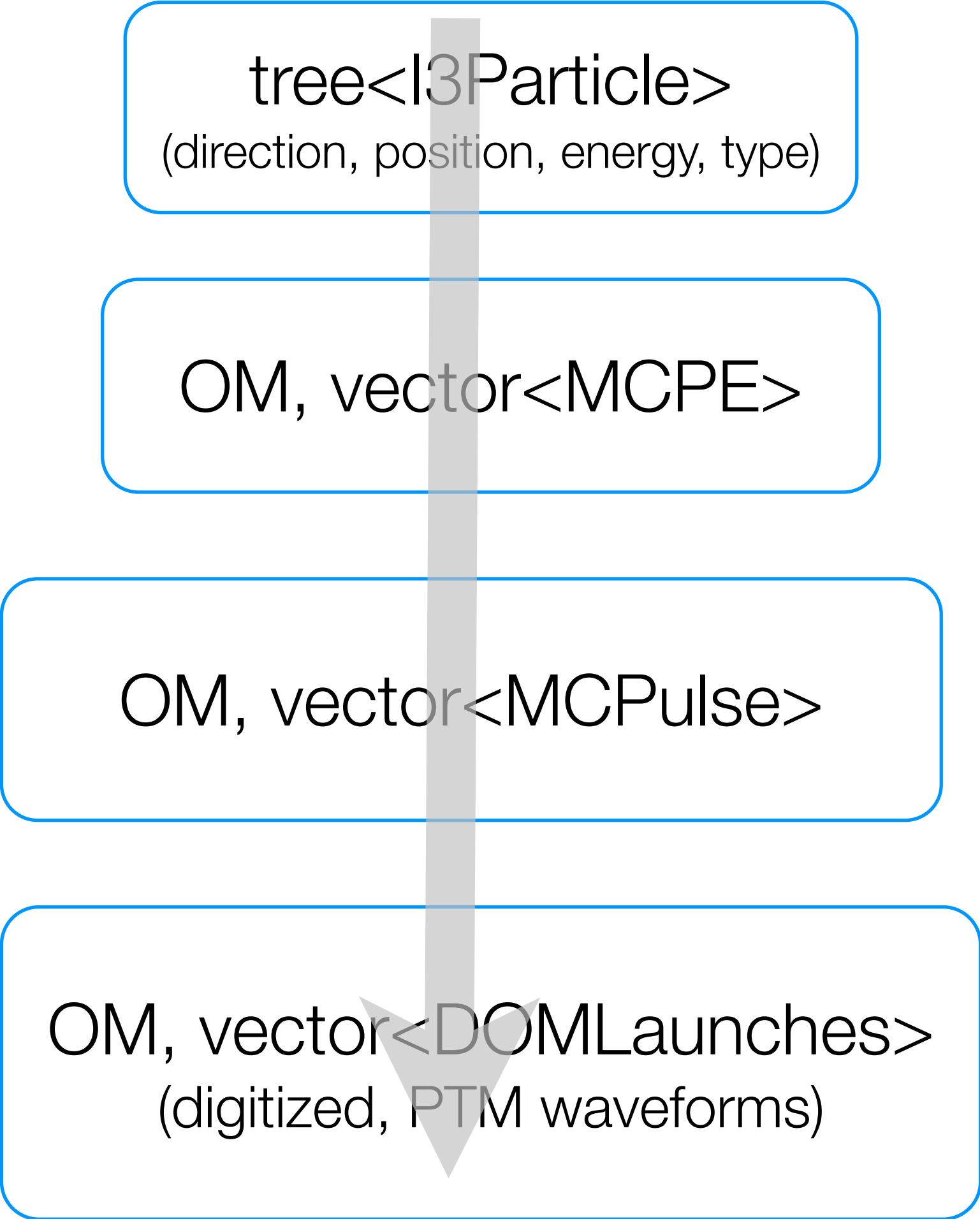
Backup

simulaton chain (IT)



Simulation

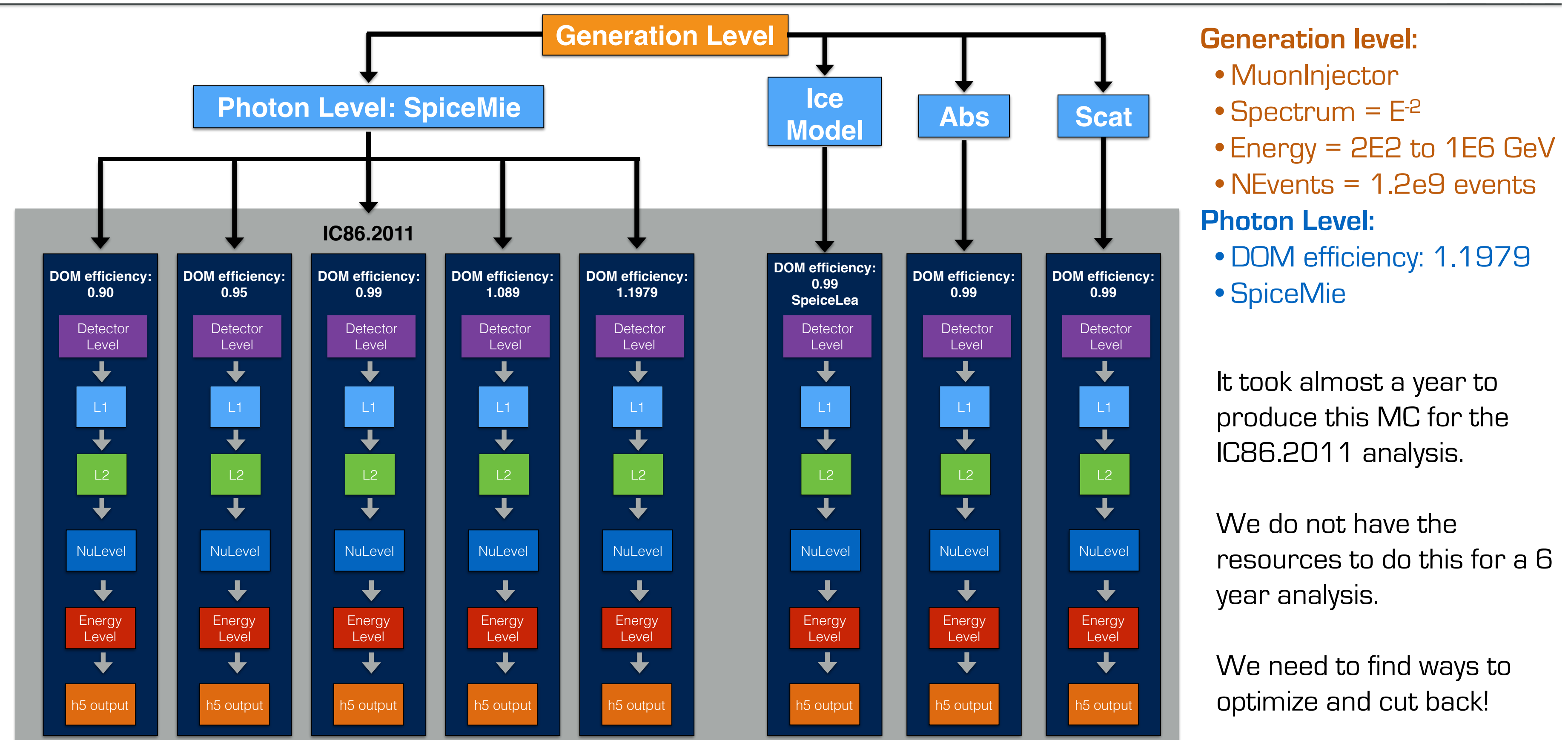
Reconstruction



Simulating Systematic Uncertainties

Example: High-Energy Sterile Neutrino MC Generation

Spencer N. Axani



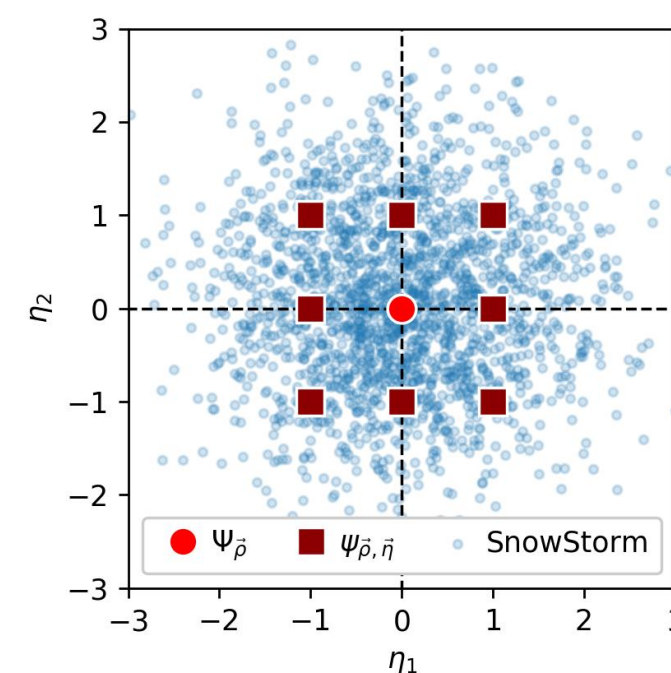
SnowStorm

https://events.icecube.wisc.edu/event/118/contributions/6499/attachments/5362/6082/DiffuseParallel_Brussels_SnowStormMCGlobalfit.pdf

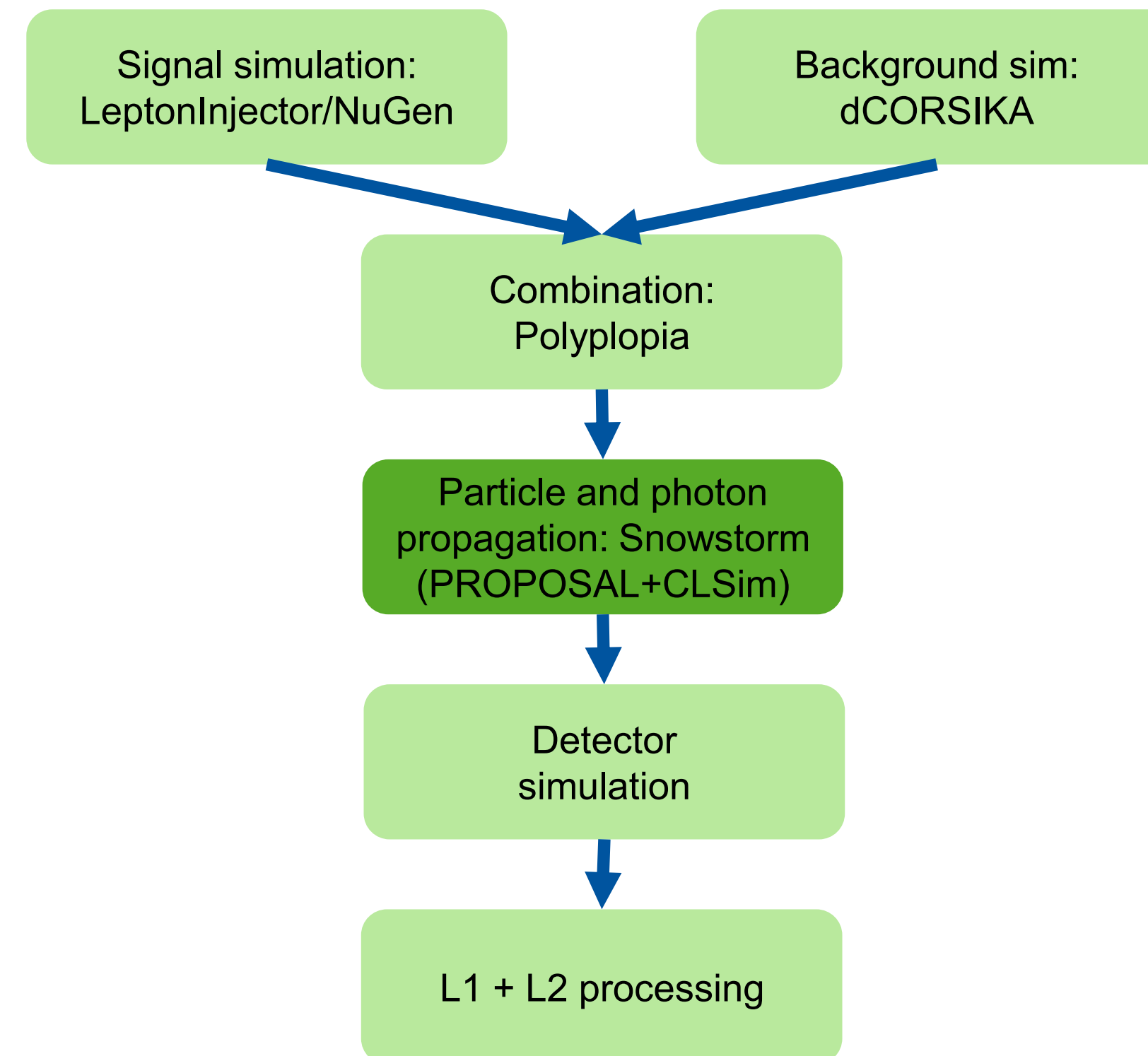
Erik Ganster

SnowStorm Simulation Chain – SnowStorm

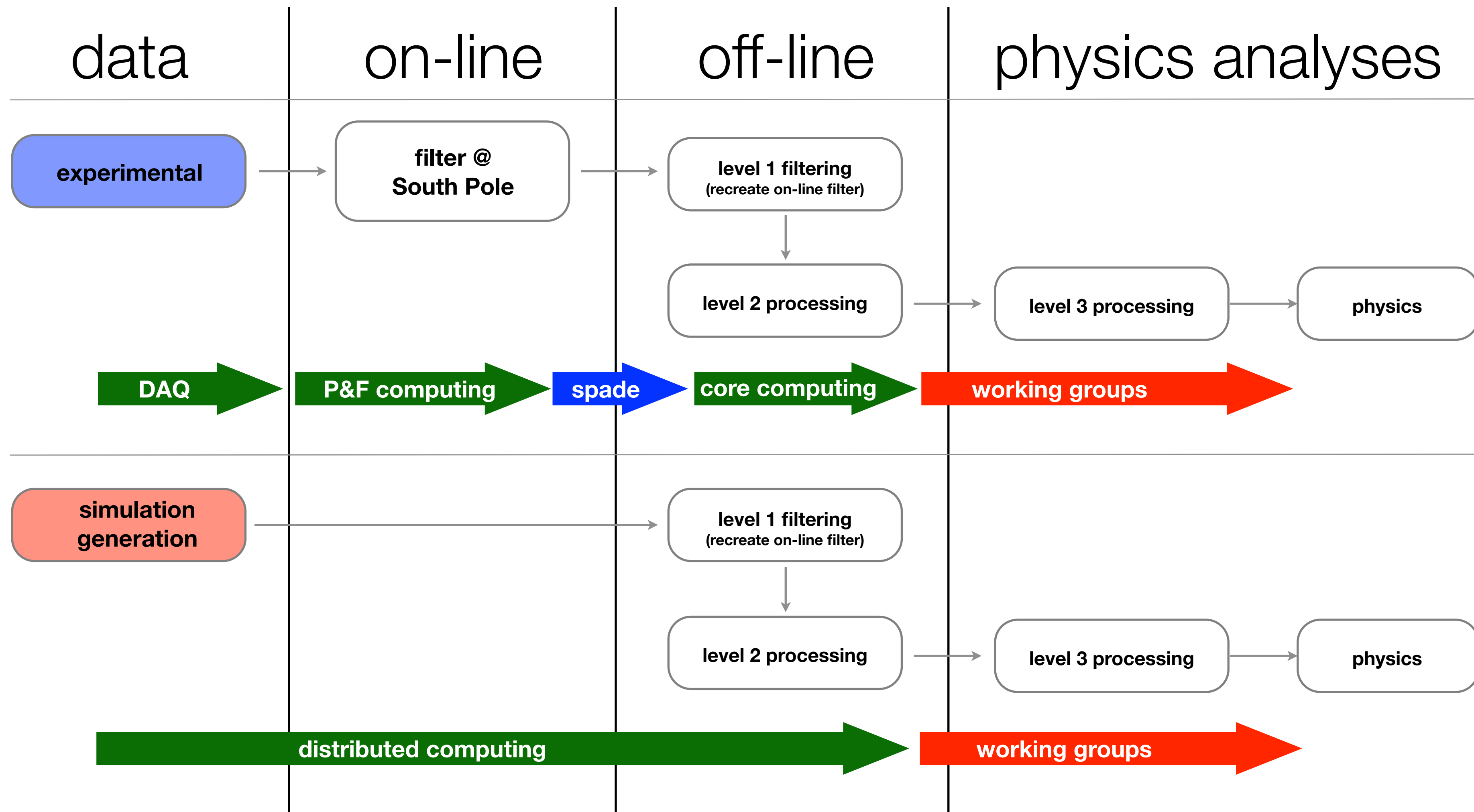
- Based on “standard” simulation chain
- Merge of signal+background I3MCTrees before any particle or photon propagation
 - Ensures that all particles get treated/propagated with the exact same parameters/settings further on
- Main SnowStorm simulation step:
 - Particle (muon) propagation with PROPOSAL
 - Photon propagation using CLSim
- Perturbing the ice model properties for chunks of frames using the *SnowStorm perturber*



SnowStorm short: Continuous variation of nuisance parameters (detector systematics) (blue) instead of discrete sets for specific values (red)



flow of experimental and simulation data



Simulating the online filter and L2 processing

```
[gtx-00]$ python filterscripts/resources/scripts/SimulationFiltering.py -h
```

```
usage: SimulationFiltering.py [-h] [-i INFILE] [-g GCDFILE] [-o OUTFILE]
                             [-n NUM] [--qify]
                             [--MinBiasPrescale MINBIASPRESCALE]
                             [--photonicdir PHOTONICSDIR] [--enable-gfu]
                             [--log-level LOG_LEVEL] [--log-filename LOGFN]
                             [--needs_wavedeform_spe_corr]
```

optional arguments:

-h, --help show this help message and exit

-i INFILE, --input INFILE

Input i3 file(s) (use comma separated list for multiple files)

-g GCDFILE, --gcd GCDFILE

GCD file for input i3 file

-o OUTFILE, --output OUTFILE

Output i3 file

-n NUM, --num NUM Number of frames to process

--qify Apply QConverter, use if file is P frame only

```
--MinBiasPrescale MINBIASPRESCALE
```

Set the Min Bias prescale to something other than default

```
--photonicdir PHOTONICSDIR
```

Directory with photonics tables

<code>--enable-gfu</code>	Do not run GFU filter
---------------------------	-----------------------

```
--log-level LOG_LEVEL
```

Sets the logging level (ERROR, WARN, INFO, DEBUG, TRACE)

--log-filename LOGFN If set logging is redirected to the specified file.

```
--needs_wavedeform_spe_corr
```

apply_spe_correction in wavedeform.

Simulating the online filter and L2 processing

```
[gtx-00]$ python filterscripts/resources/scripts/offlineL2/process.py -h
```

```
usage: process.py [-h] [-s] [-i INFILE] [-g GCDFILE] [-o OUTFILE] [-n NUM]
                  [--dstfile DSTFILE] [--gapsfile GAPSFILE]
                  [--icetopoutput ICETOPOUTPUT] [--eheoutput EHEOUTPUT]
                  [--slopoutput SLOPOUTPUT] [--rootoutput ROOTOUTPUT]
                  [--photonicsdir PHOTONICSDIR] [--log-level LOG_LEVEL]
                  [--log-filename LOGFN]
```

optional arguments:

```
-h, --help          show this help message and exit
-s, --simulation    Mark as simulation (MC)
-i INFILE, --input INFILE
                    Input i3 file(s) (use comma separated list for
                    multiple files)
-g GCDFILE, --gcd GCDFILE
                    GCD file for input i3 file
-o OUTFILE, --output OUTFILE
                    Output i3 file
-n NUM, --num NUM   Number of frames to process
--dstfile DSTFILE   DST root file (should be .root)
--gapsfile GAPSFILE gaps text file (should be .txt)
--icetopoutput ICETOPOUTPUT
                    Output IceTop file
--eheoutput EHEOUTPUT
                    Output EHE i3 file
--slopoutput SLOPOUTPUT
                    Output SLOP file
--rootoutput ROOTOUTPUT
                    Output root file
--photonicsdir PHOTONICSDIR
                    Directory with photonics tables
--log-level LOG_LEVEL
                    Sets the logging level (ERROR, WARN, INFO, DEBUG,
                    TRACE)
--log-filename LOGFN If set logging is redirected to the specified file.
```