

Simulation Production



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

Juan Carlos Díaz-Vélez

IceProd/SimProd Workshop

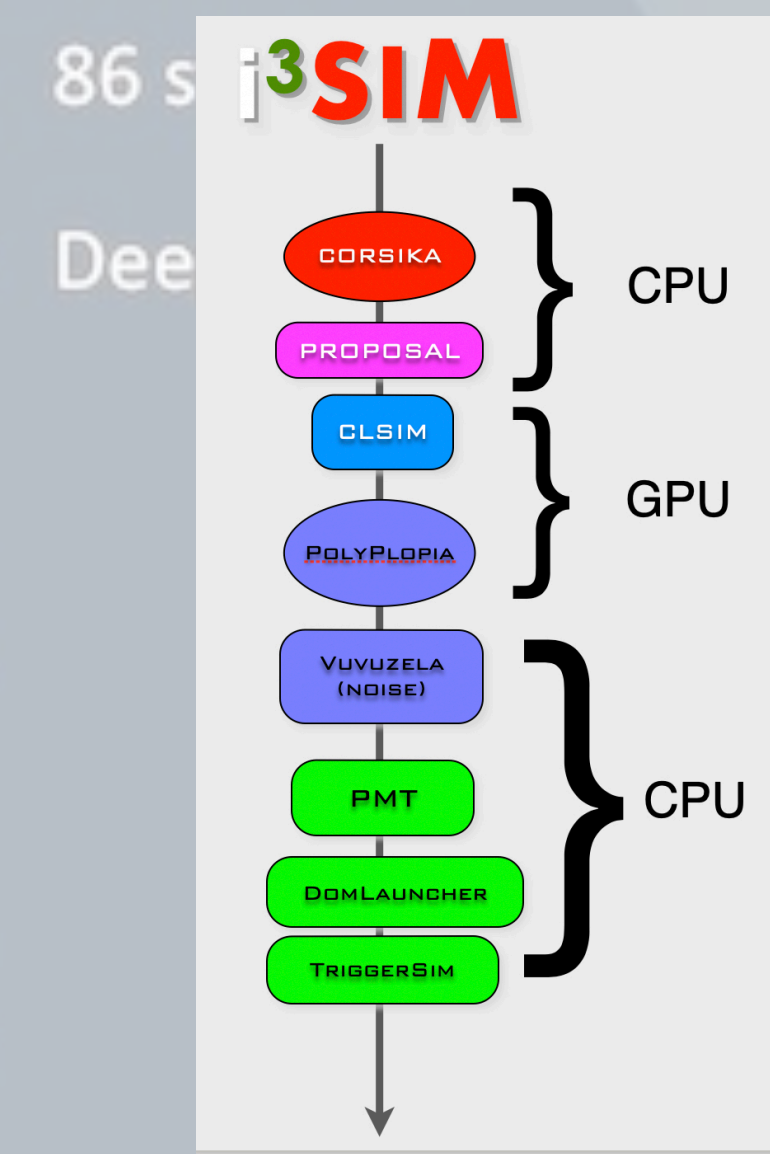
Madison, WI USA

June 14, 2024

1450 m

50 m

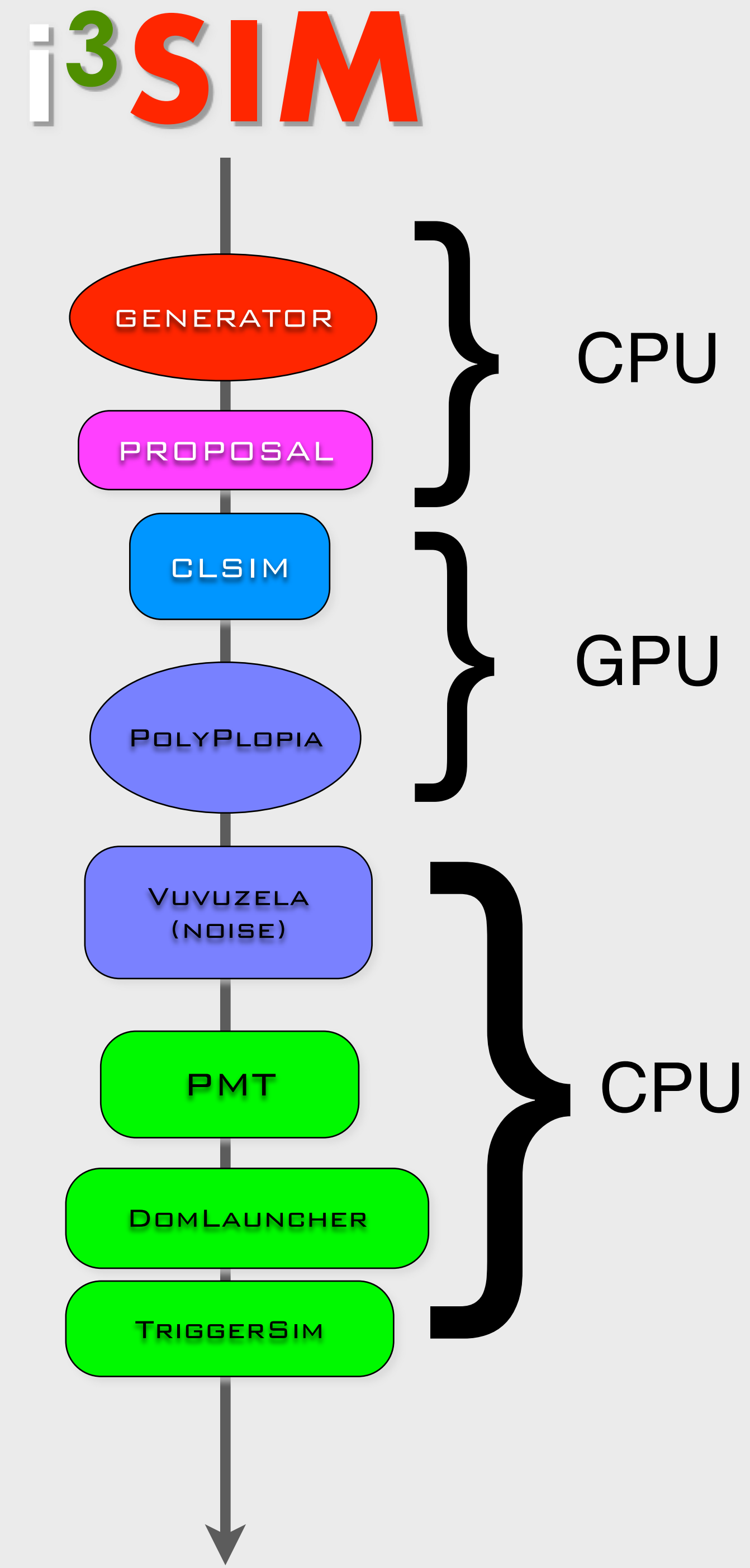
IceTop



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



simprod-scripts

<https://github.com/icecube/icetray/tree/main/simprod-scripts>

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

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)

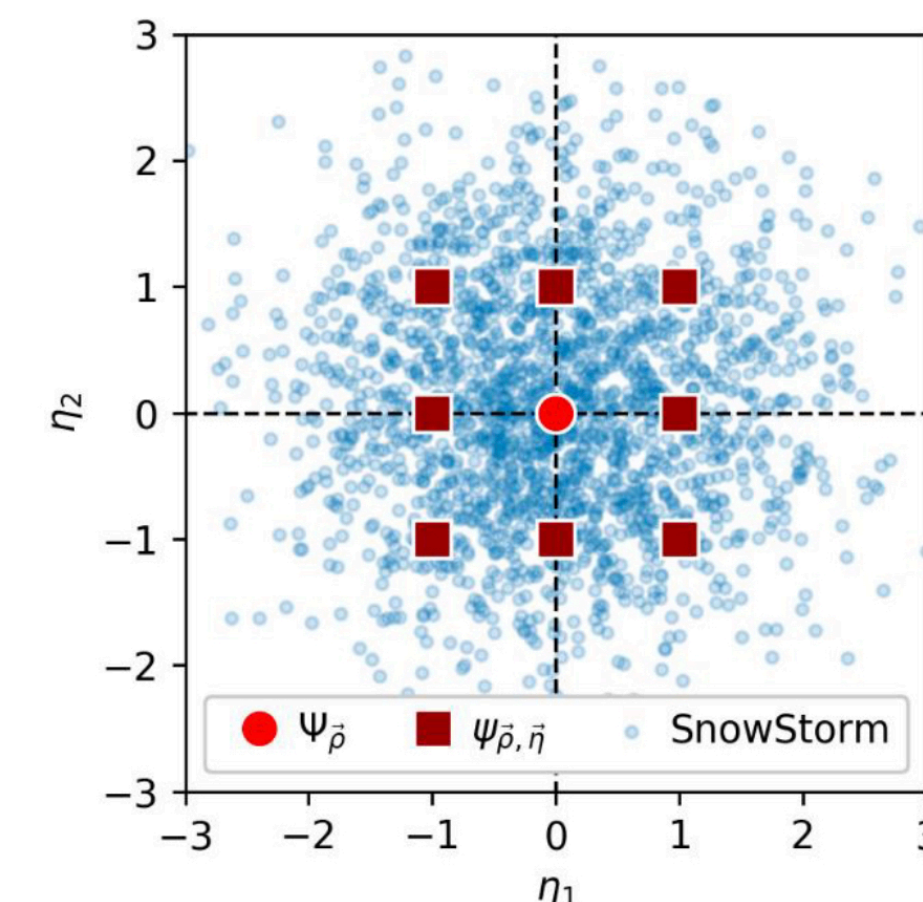
Scripts:

\$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-Polyploia.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]$ 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

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 \  
    --skiptoptions 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
```


IceProd

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/>

Datasets Profile Logout

Dataset 21889 Details

[View Config](#) [Edit 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:
debug: False
jobs_immutable: False

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 screenshot displays the GitHub interface for the repository `simprod-templates`. The left sidebar shows the file tree with the `corsika` directory expanded, highlighting `Triggered-CORSIKA.json`. The main content area shows the file's commit history, with the latest commit by `jcdiazvelez` titled "Update Triggered-CORSIKA.json" from 9 hours ago. The file content is a JSON configuration for a server task, showing requirements for GPU, CPU, disk, time, and memory, along with a name, parameters, depends, batchsys, and trays section.

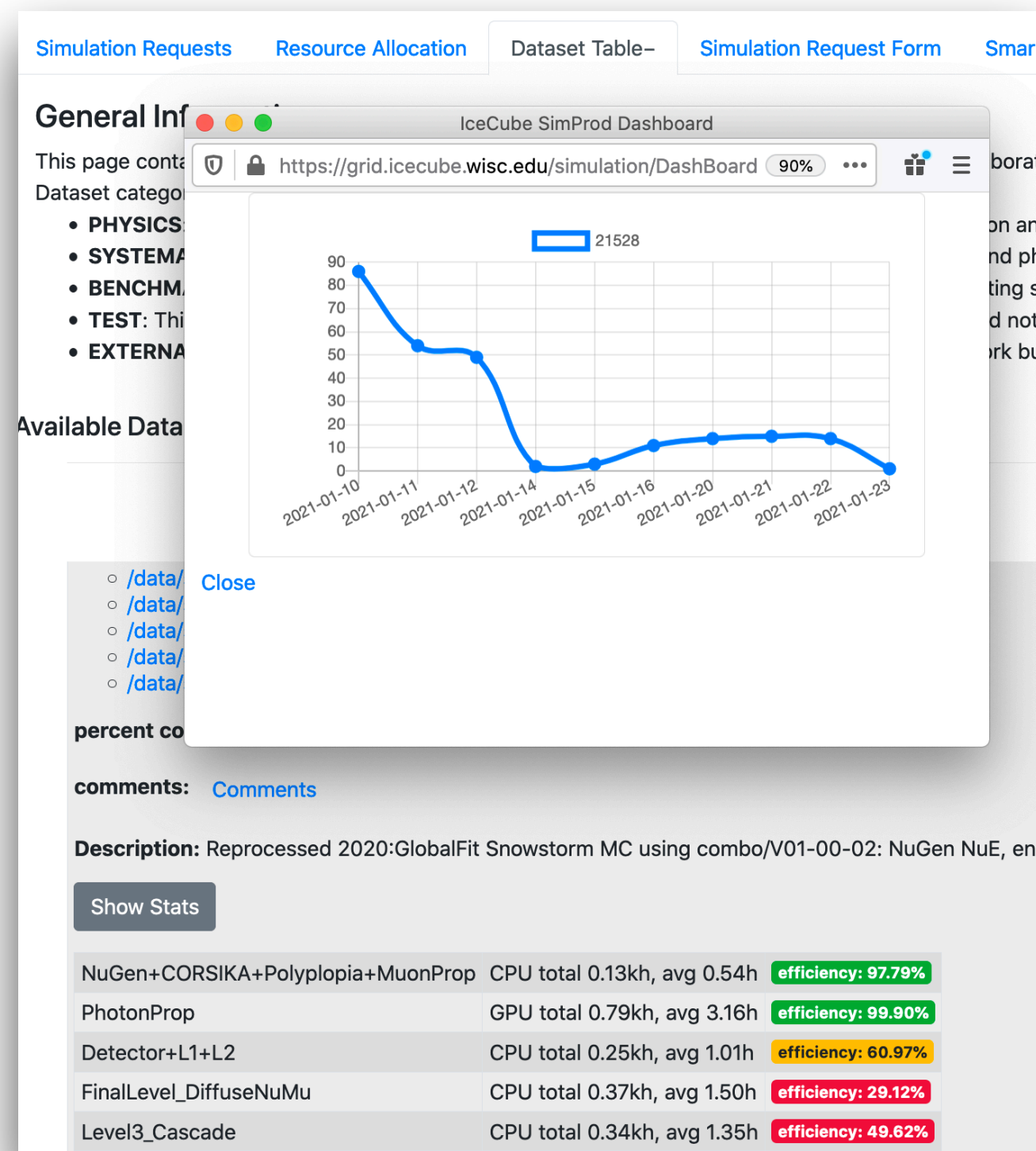
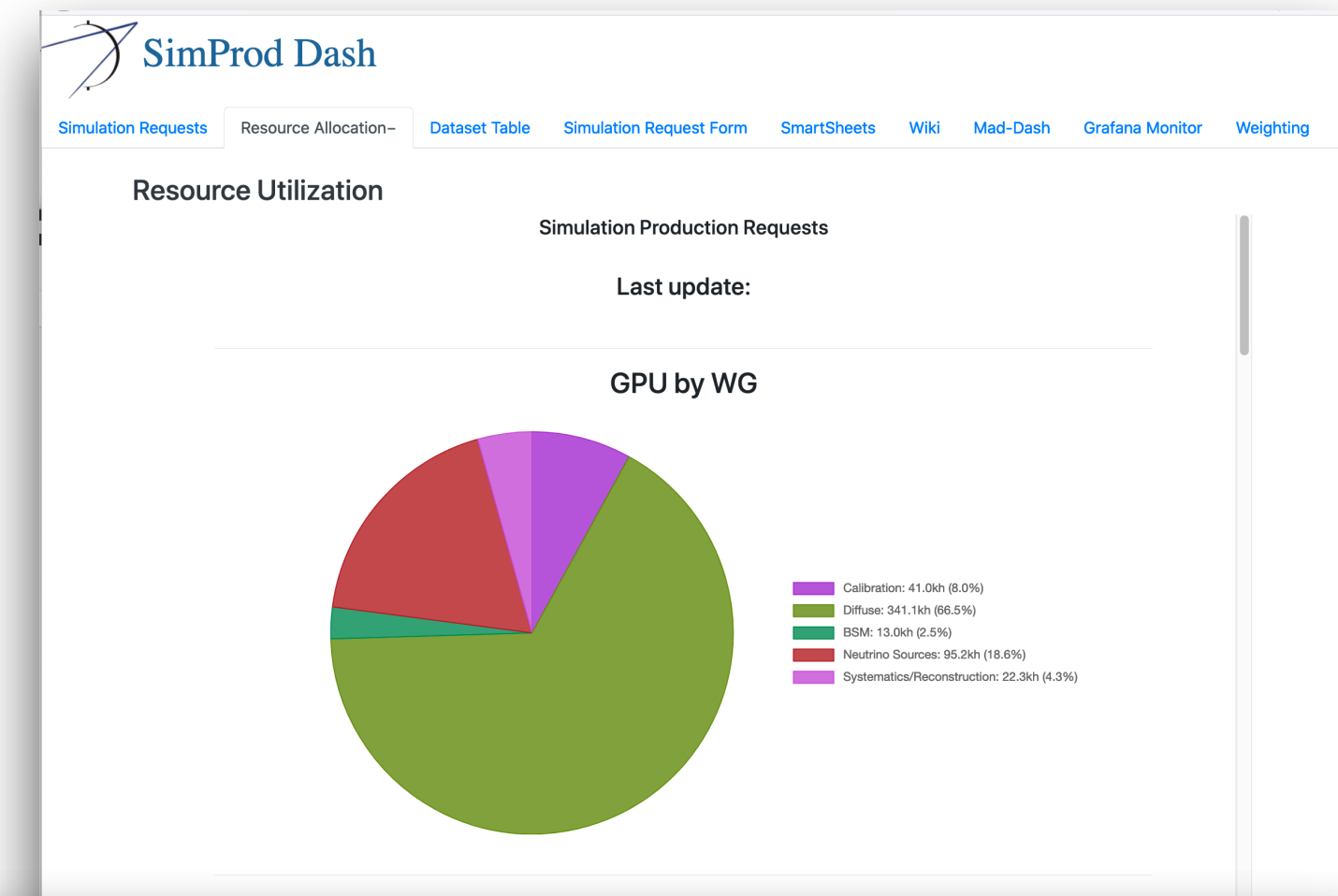
```
1  {
2    "difplus":null,
3    "tasks":[
4      {
5        "requirements":{
6          "gpu":1,
7          "cpu":8,
8          "disk":8,
9          "time":2.5,
10         "memory":11
11       },
12       "name":"server",
13       "parameters":{},
14       "depends":[],
15       "batchsys":{
16  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

SimProd Dashboard

Simprod DashBoard

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

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



SimProd Dash

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

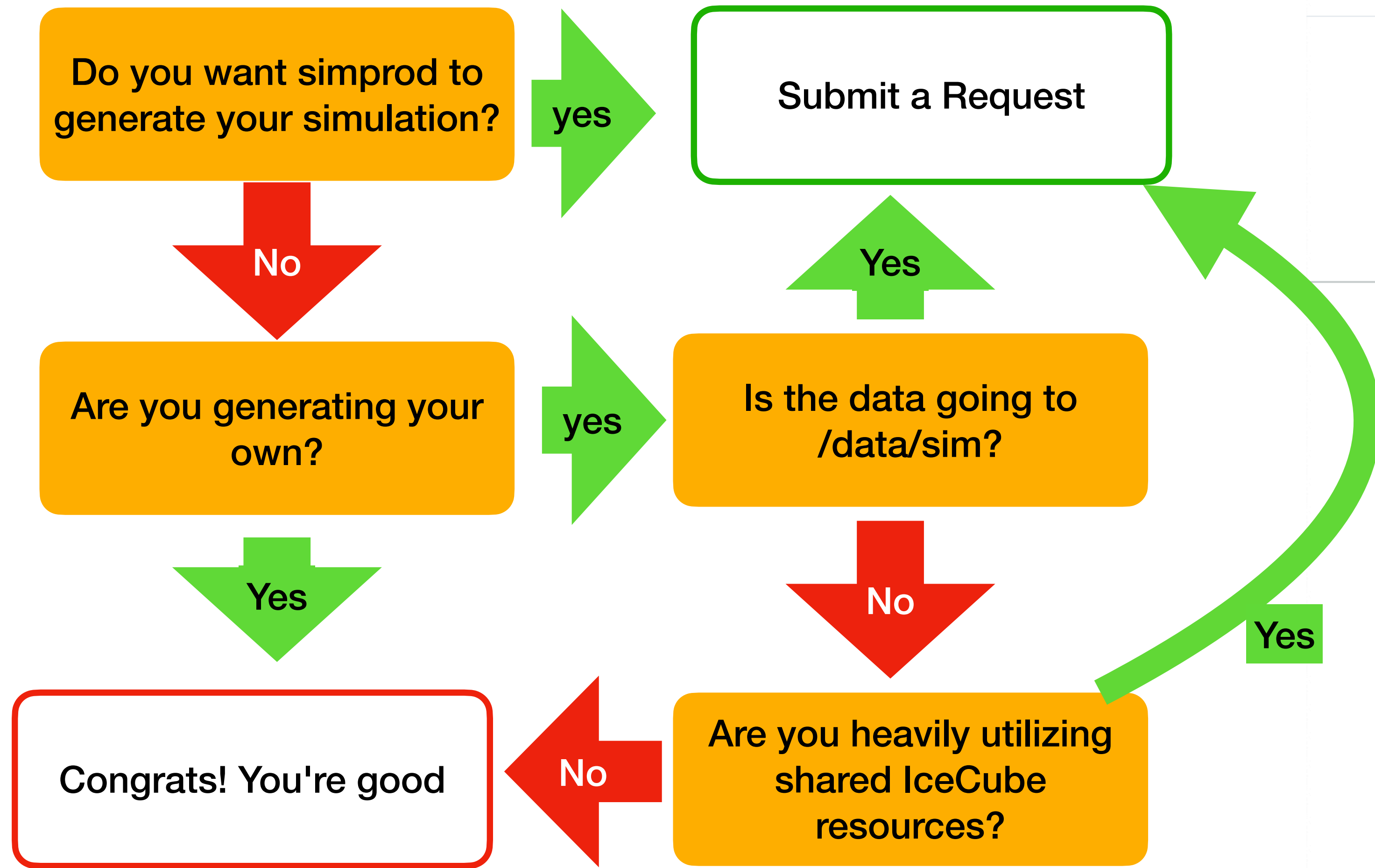
Need new simulations?

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

The screenshot shows the SimProd Dash web interface. At the top right is a "Logout" link. The main navigation bar includes "Simulation Requests", "Resource Allocation", and "Dataset Table". Below this is a secondary navigation bar with "Simulation Request Form" (highlighted), "Wiki", "Mad-Dash", and "Grafana Monitor". A "Weighting" link is also visible. The main content area is titled "Simulation Request Form" and contains a dropdown menu for "Instructions:". Below this is a form titled "Simulation Request Form" with the instruction "Specify the details of what you need." The form fields are: "Requester" (text input with "juancarlos@icecube.wisc.edu"), "Working Group" (dropdown menu with "GENERAL" selected), "Title/Description" (text input), "Analysis" (text input), "Date needed by" (text input), "Estimated req. CPU" (spin box with "0.0"), "Estimated req. GPU" (spin box with "0.0"), "Estimated req. storage" (spin box with "0.0"), "Documentation" (text input), and "Additional comments" (text area).

SimProd Requests

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



The screenshot shows the SimProd Dash interface. At the top right is a "Logout" link. Below the "SimProd Dash" logo are navigation links: "Simulation Requests", "Resource Allocation", and "Dataset Table". A secondary row of links includes "Simulation Request Form", "Wiki", "Mad-Dash", and "Grafana Monitor". A "Weighting" link is also present. The main content area is titled "Simulation Request Form" and features a dropdown menu for "Instructions:". Below this is a form titled "Simulation Request Form" with the instruction "Specify the details of what you need." The form fields are: "Requester" (text input with value "juancarlos@icecube.wisc.edu"), "Working Group" (dropdown menu with value "GENERAL"), "Title/Description" (text input), "Analysis" (text input), "Date needed by" (text input), "Estimated req. CPU" (spin box with value 0.0), "Estimated req. GPU" (spin box with value 0.0), "Estimated req. storage" (spin box with value 0.0), "Documentation" (text input), and "Additional comments" (text input).

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

The screenshot shows the SimProd Dash interface. At the top right is a "Logout" link. Below the "SimProd Dash" logo are navigation links: "Simulation Requests", "Resource Allocation", "Dataset Table", "Simulation Request Form", "Wiki", "Mad-Dash", "Grafana Monitor", and "Weighting". The "Simulation Request Form" is the active page, featuring a dropdown menu for "Instructions:". Below this is a form titled "Simulation Request Form" with the instruction "Specify the details of what you need." The form fields are: "Requester" (text input with "juancarlos@icecube.wisc.edu"), "Working Group" (dropdown menu with "GENERAL" selected), "Title/Description" (text input), "Analysis" (text input), "Date needed by" (text input), "Estimated req. CPU" (spin box with "0.0"), "Estimated req. GPU" (spin box with "0.0"), "Estimated req. storage" (spin box with "0.0"), "Documentation" (text input), and "Additional comments" (text area).

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")

The screenshot shows the SimProd Dash interface. At the top right is a "Logout" link. Below the "SimProd Dash" logo are navigation links: "Simulation Requests", "Resource Allocation", "Dataset Table", "Simulation Request Form", "Wiki", "Mad-Dash", "Grafana Monitor", and "Weighting". The "Simulation Request Form" is highlighted. Below the navigation is a dropdown menu labeled "Instructions:". The main content area is titled "Simulation Request Form" and contains the instruction "Specify the details of what you need." followed by a form with the following fields:

Requester	<input type="text" value="juancarlos@icecube.wisc.edu"/>
Working Group	<input type="text" value="GENERAL"/>
Title/Description	<input type="text"/>
Analysis	<input type="text"/>
Date needed by	<input type="text"/>
Estimated req. CPU	<input type="text" value="0.0"/>
Estimated req. GPU	<input type="text" value="0.0"/>
Estimated req. storage	<input type="text" value="0.0"/>
Documentation	<input type="text"/>
Additional comments	<input type="text"/>

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")

The screenshot shows the SimProd Dash interface. At the top right is a "Logout" link. Below the logo are navigation links: "Simulation Requests", "Resource Allocation", "Dataset Table", "Simulation Request Form", "Wiki", "Mad-Dash", and "Grafana Monitor". A "Weighting" link is also visible. The main heading is "Simulation Request Form". Below it is a dropdown menu labeled "Instructions:". The form itself is titled "Simulation Request Form" and includes the instruction "Specify the details of what you need." The form fields are: "Requester" (text input with "juancarlos@icecube.wisc.edu"), "Working Group" (dropdown menu with "GENERAL" selected), "Title/Description" (text input), "Analysis" (text input), "Date needed by" (text input), "Estimated req. CPU" (spin box with "0.0"), "Estimated req. GPU" (spin box with "0.0"), "Estimated req. storage" (spin box with "0.0"), "Documentation" (text input), and "Additional comments" (text input).

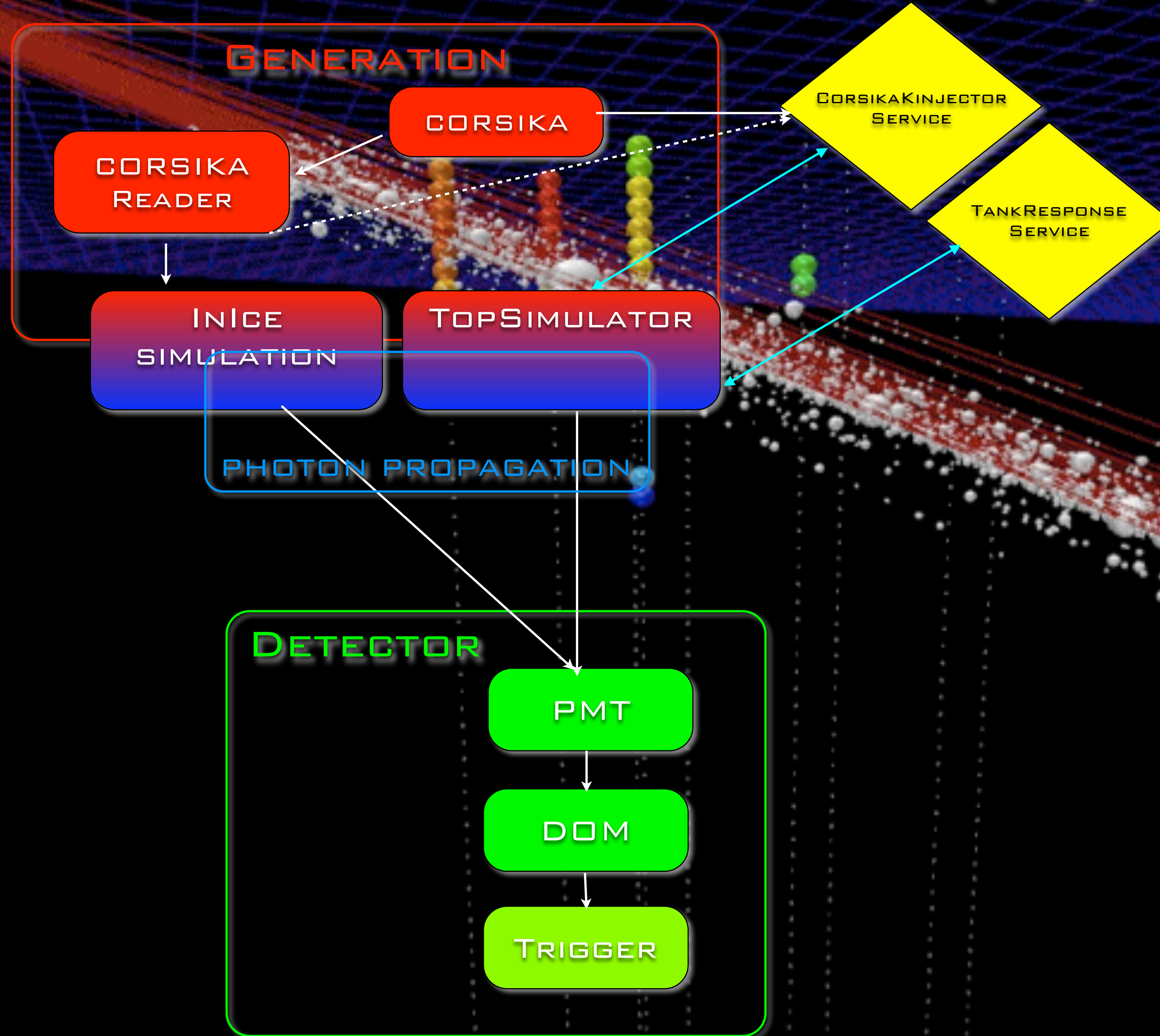
<http://simprod.icecube.wisc.edu>

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](#)

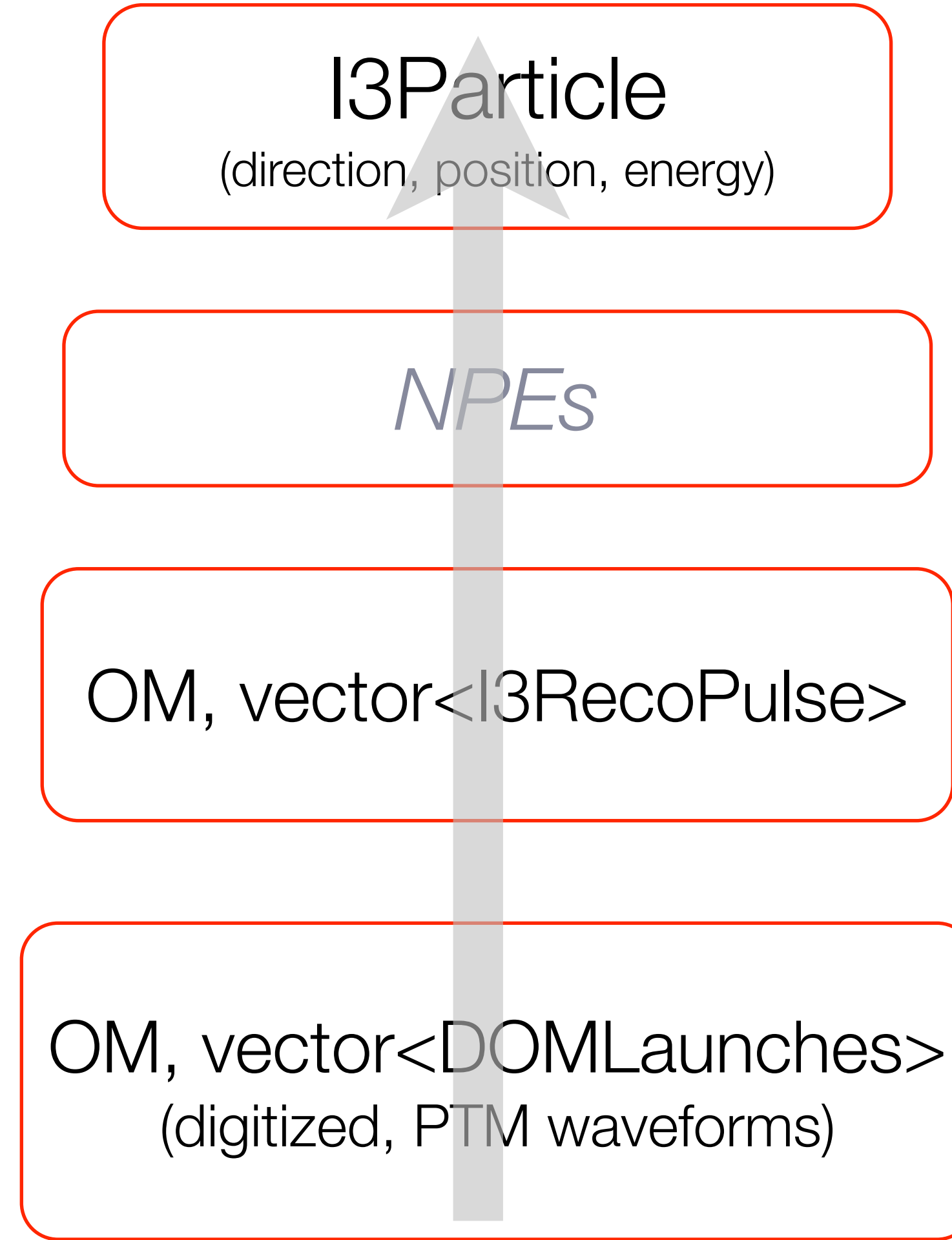
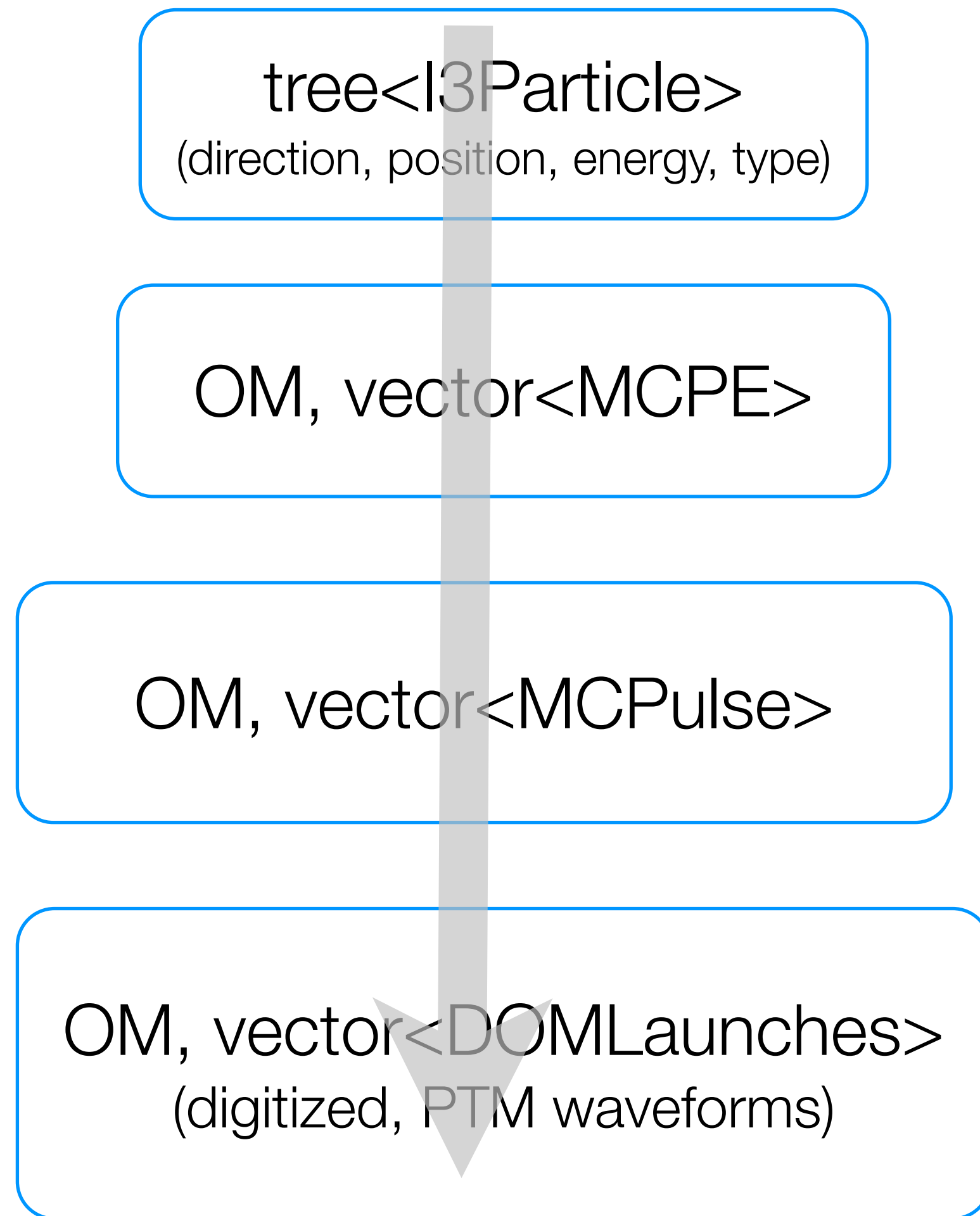
Backup

simulaton chain (IT)



Simulation

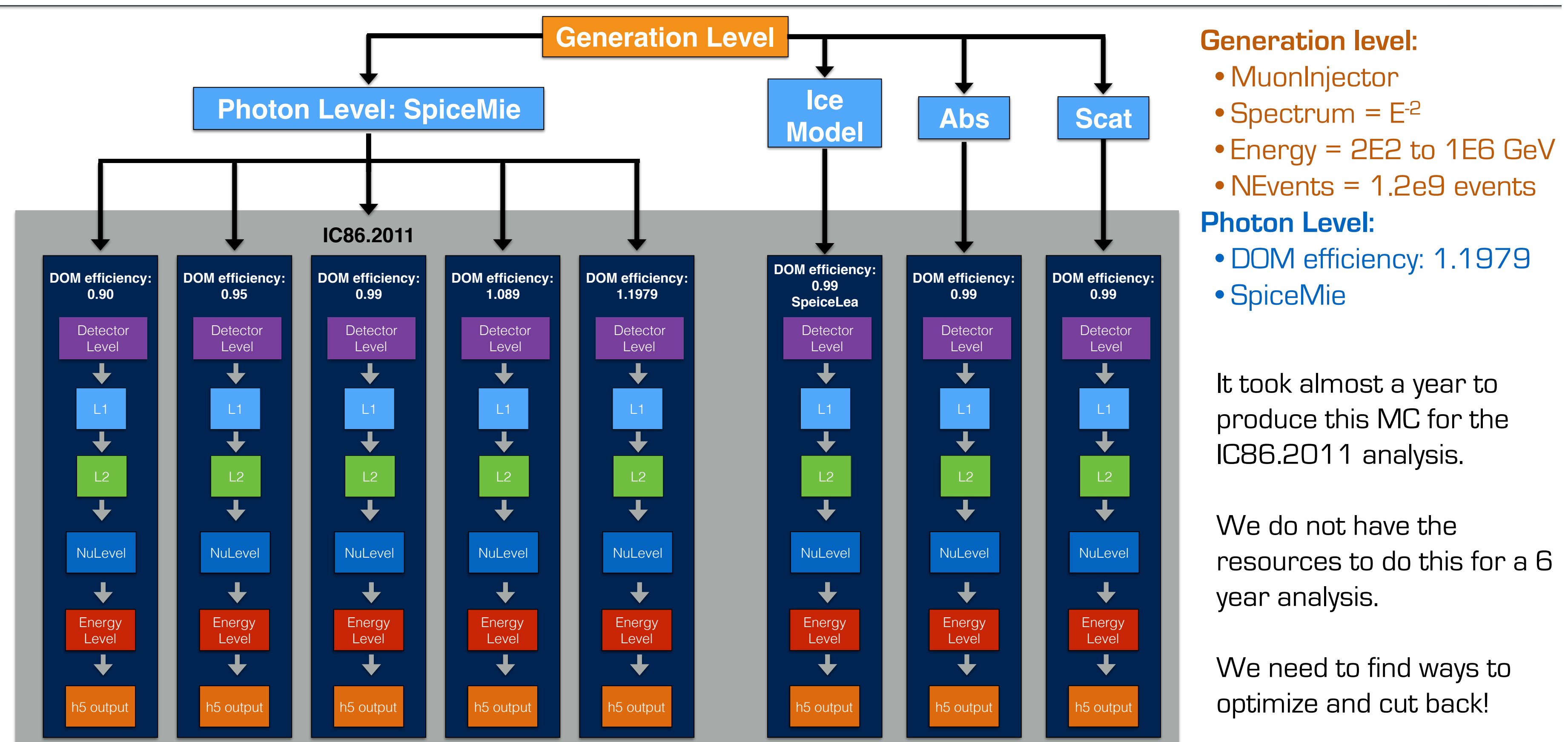
Reconstruction



Simulating Systematic Uncertainties

Example: High-Energy Sterile Neutrino MC Generation

Spencer N. Axani



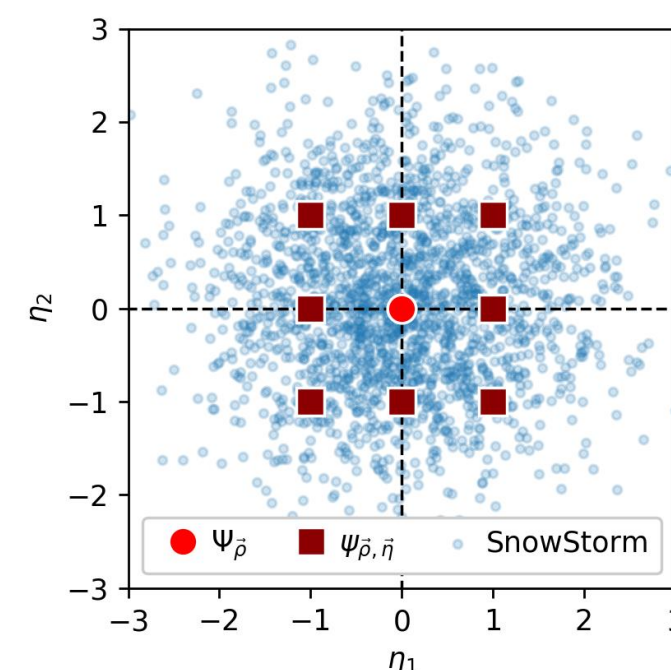
SnowStorm

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

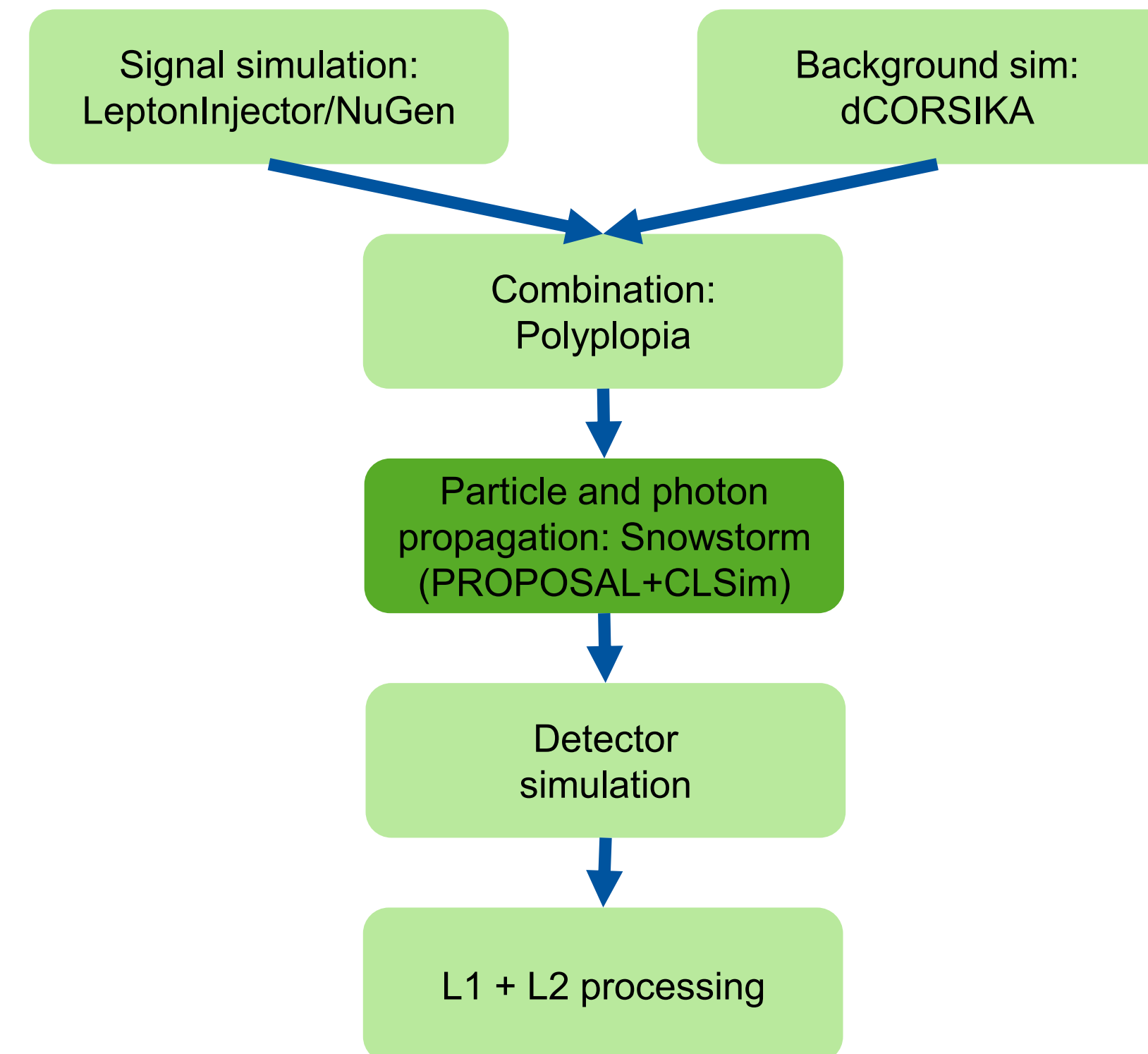
Erik Ganster

SnowStorm Simulation Chain – SnowStorm

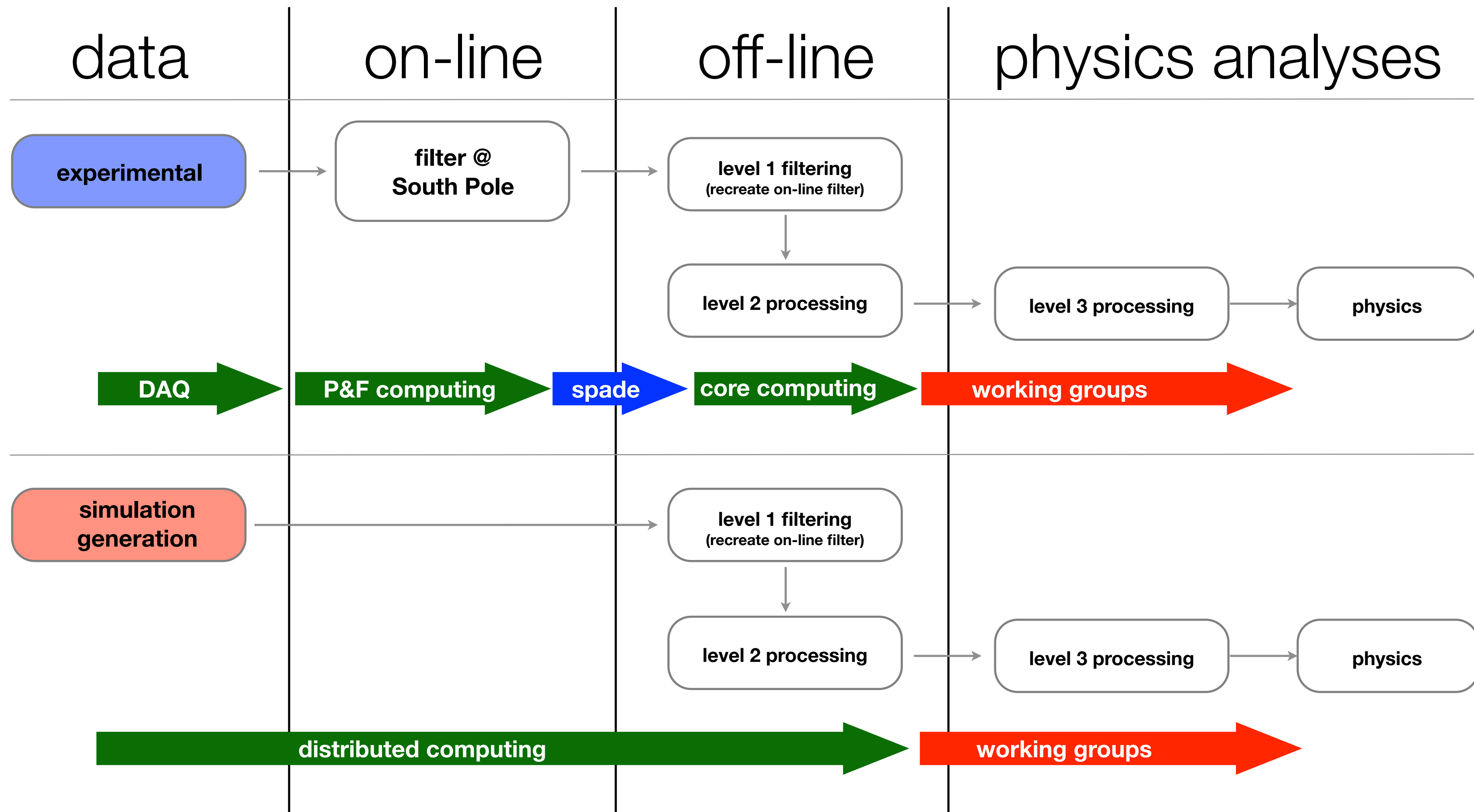
- Based on “standard” simulation chain
- Merge of signal+background I3MCTrees before any particle or photon propagation
→ Ensures that all particles get treated/propagated with the exact same parameters/settings further on
- Main SnowStorm simulation step:
 - Particle (muon) propagation with PROPOSAL
 - 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]
                             [--photonicsdir 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
--photonicsdir PHOTONICSDIR
                    Directory with photonics tables
--enable-gfu       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_corection 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.
```