Software organization and plans

- from simulations to IRFs -

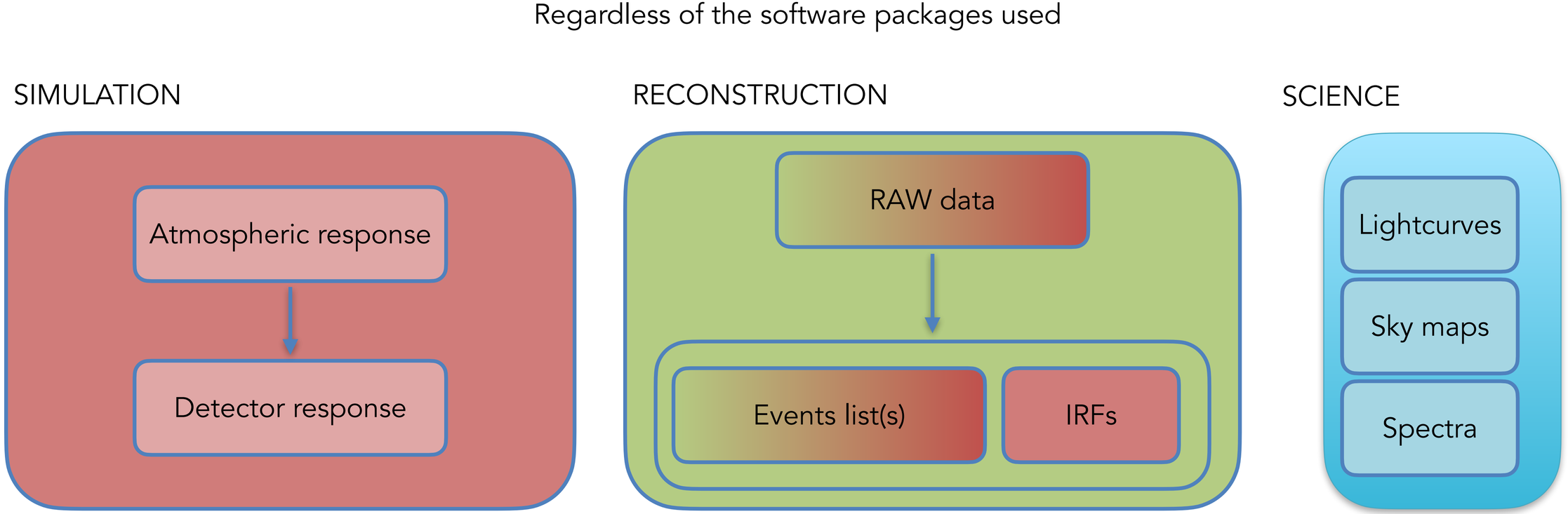
M. Peresano 1,★

★ Speaker
[1] Università degli Studi di Torino
[2] INFN-Torino

SWGO Italia meeting - Naples - 24/11/23

The general steps

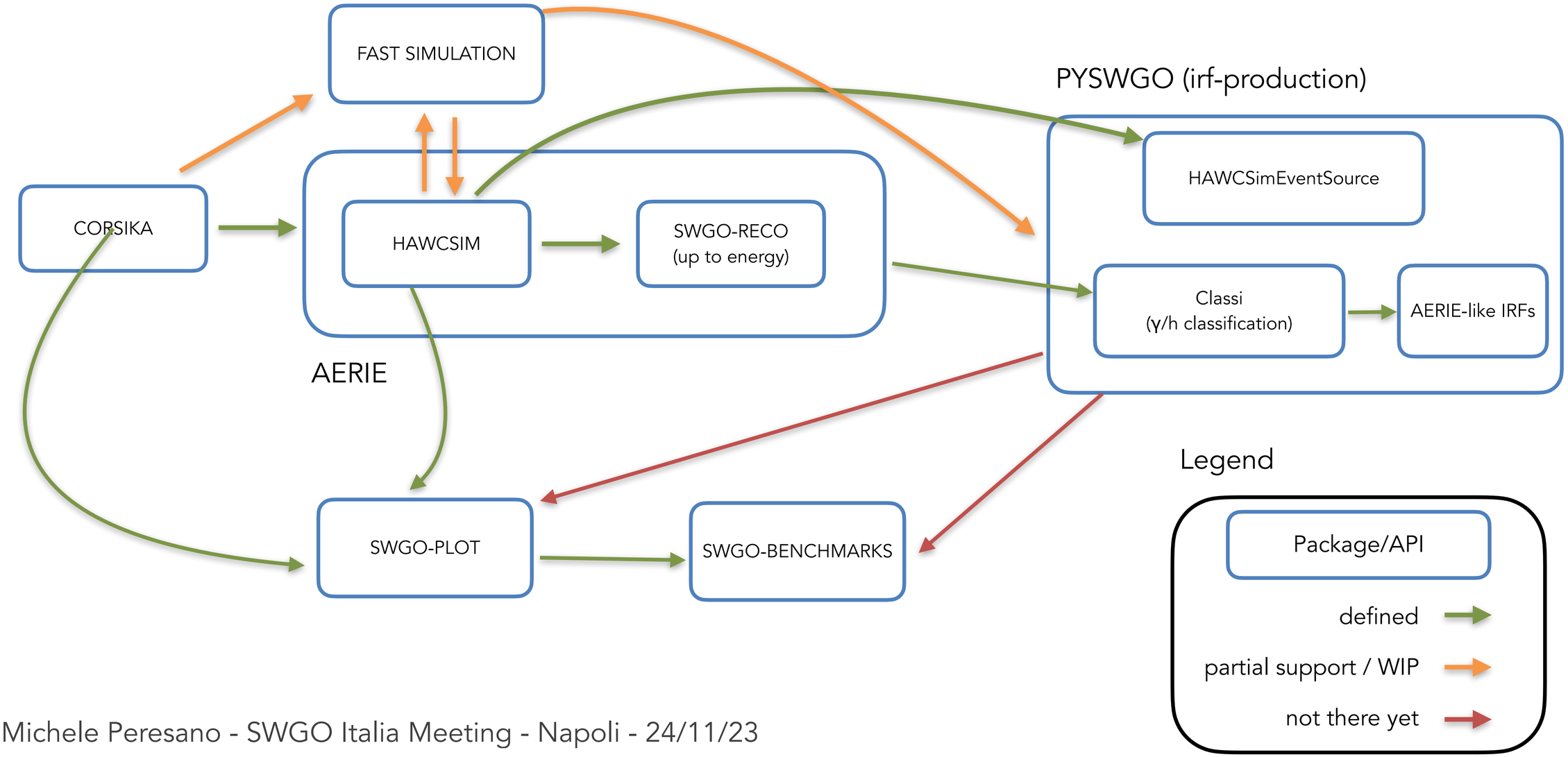
Regardless of the software packages used



Simulation

- Atmospheric response : Corsika v7.*
  - tip: use pycorsikaio to read these files with Python3
- Detector response: HAWCSim (part of AERIE - recycled HAWC data processing pipeline - executable name `hawcsim-exe`)
  - successor under design planning
- Fast simulation, see A. Negro presentation in this meeting

Current situation



AERIE reconstruction configurations

- In order to run swgo-reco and the LUT generator one needs a set of configuration files
- these depend on the simulated properties of array and tank designs
- `git clone git@gitlab.com:swgo-collaboration/config-swgo.git`
- Follow instructions at https://gitlab.com/swgo-collaboration/config-swgo/-/blob/main/README.md?ref_type=heads
- see Reconstruction talk for details

pyswgo

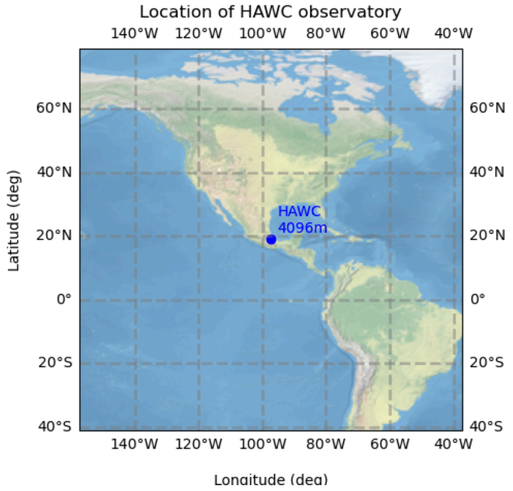
- **Python3 package** (currently ***irf-production***, but doing more than that…)
- **can read SWGO-AERIE reconstructed data**
- perform **event classification**
- generate **IRFs** compatible with gammapy
- current artisanal framework in use to support AERIE and (tight) M6 milestone

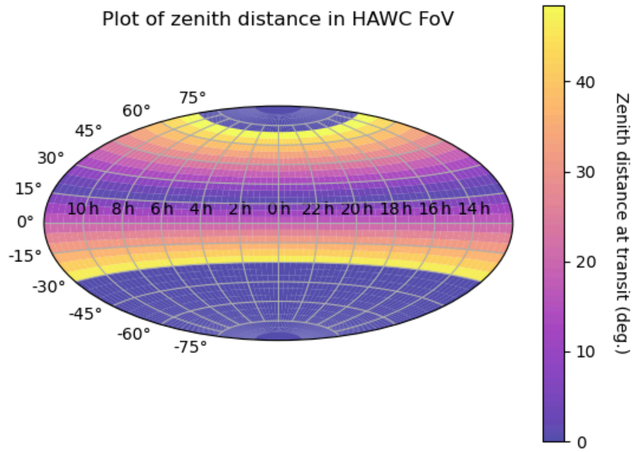
pyswgo

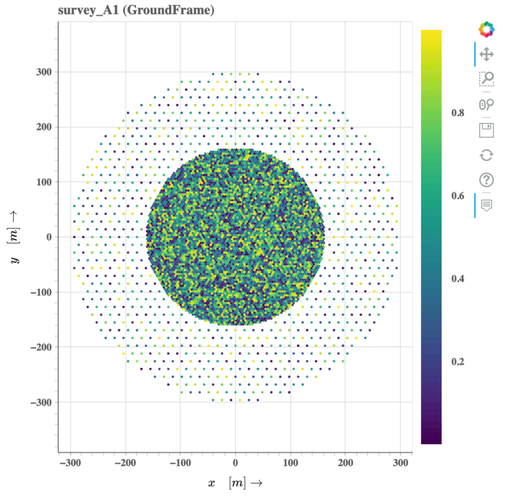
- In parallel, **new framework** being developed:
  - based on ctapipe
  - can fully load HAWCSim data files
  - soon reconstruction part should start (see this issue)
  - **Mission**
    - replace old framework (`AERIE` reconstruction + old `pyswgo` scripts)
    - public and open-source package
    - open to other drifting instruments

swgo-plot

- read from **multiple sources & data levels**
  - *Corsika / XCDF / XML / FITS available*
- **compare results** in a stable way
- **centralised and standardised**
- *quantities with* ***units***
- *Compute* ***metrics***
- ***agnostic plot functions***







See here what's planned or still missing (spoiler: a lot!)

Contributions are welcome!

Package modules

Corsika / Detector / Array-level / ML / IRFs / Sky / .....?

swgo-benchmarks

- prototype **benchmarking framework** based on Jupyter
- **standardised & parametrised notebooks**
- structured as a **full analysis workflow**
- **version controlled** and **convertible** to text formats with Jupytext
- fully customisable **analysis reports** with Jupyter Book
- **compatible with swgo-plot and pyswgo** by design

```
notebooks/
├── reconstruction
│   ├── cuts_optimization
│   ├── energy_models
│   ├── estimation
│   │   ├── classification
│   │   ├── energy
│   │   ├── multi_step_algorithms
│   │   │   └── template_fitting
│   │   └── signal_extraction_calibration
│   ├── irfs
│   ├── model_preprocessing
│   ├── model_validation
│   │   ├── classification
│   │   └── energy
│   └── shower_geometry
└── simulation
    ├── atmosphere_response
    └── detector_response
```

How to install/update AERIE

- Create an SSH key to talk with GitLab if you never did it before (see docs)
- `git clone git@gitlab.com:swgo-collaboration/aerie-install.git`
- To install the latest (kind of stable) version from scratch
  - `bash main.sh -d $WHERETOINSTALL -n $NCORES -c $CONDA_INSTALL_PATH -e -a ,,,, -b -i`
  - add a function in your shell init profile script which `sources` the file created at `$WHERETOINSTALL / initialize_swgo_aerie.sh`
  - to update this installation follow these instructions
- if you are also developing AERIE, clone the source code directory and use the previous installation dependencies while installing AERIE in a different directory
  - to update this installation just call the build (`-b`) and install (`-i`) options again