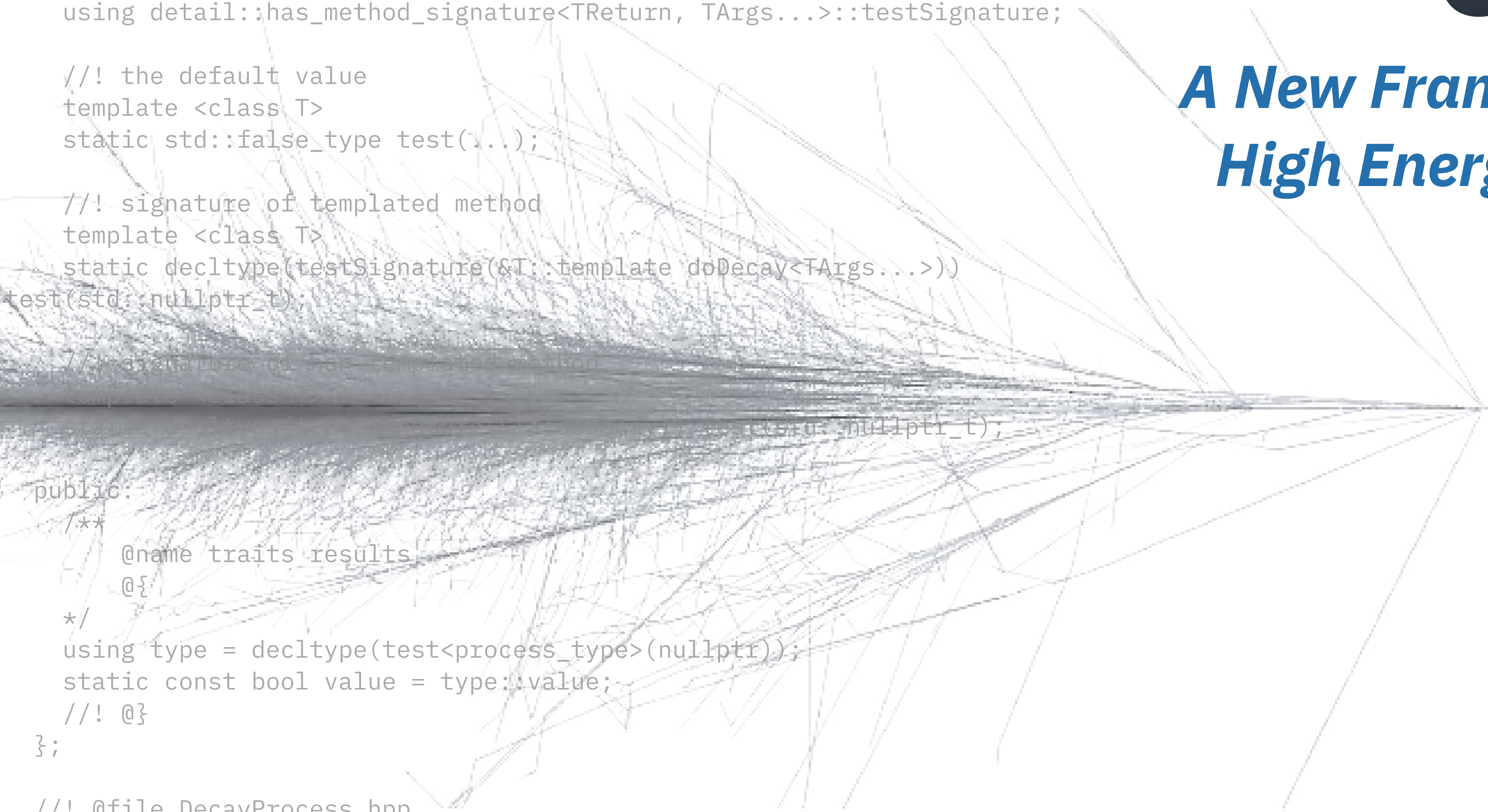
```
  template <class TProcess, typename TReturn, typename... TArgs>
  struct has_method_doDecay : public detail::has_method_signature<TReturn,
TArgs...> {

    //! type of process to be studied
    typedef std::decay_t<TProcess> process_type;

    ///! method signature
    using detail::has_method_signature<TReturn, TArgs...>::testSignature;

    //! the default value
    template <class T>
    static std::false_type test(...);

    //! signature of templated method
    template <class T>
    static decltype(testSignature(&T::template doDecay<TArgs...>))
test(std::nullptr_t);

                                                 (std::nullptr_t);

  public:
    /**
       @name traits results
       @{
    */
    using type = decltype(test<process_type>(nullptr));
    static const bool value = type::value;
    //! @}
  };

  //! @file DecayProcess.hpp
  //! value traits type
  template <class TProcess, typename TReturn, typename... TArgs>
  bool constexpr has_method_doDecay_v =
```



CORSIKA 8

A New Framework for Simulating High Energy Cascades in Matter

Remy Prechelt

for the CORSIKA 8 collaboration

Unless otherwise stated, all material from arXiv:2112.11761

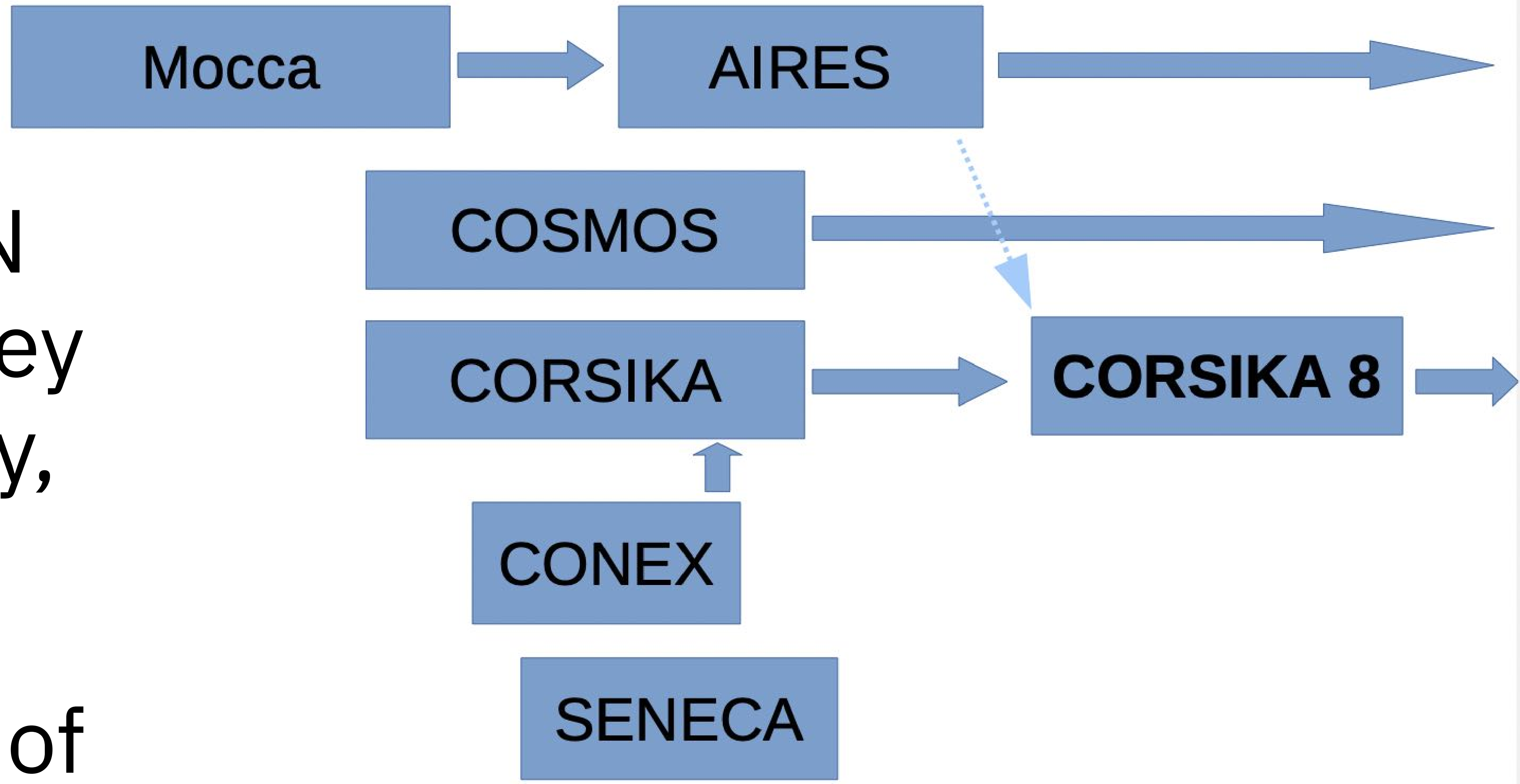
Why a new framework?

Why not just improve AIRES / CORSIKA 7?

- CORSIKA7, AIRES (among others) are **well-studied** and tested shower simulation tools with decades of heritage, and used by the largest particle astrophysics experiments.

- However, these tools are becoming challenging to develop and extend: FORTRAN is no longer *a lingua franca* of physics and they each make rigid assumptions about geometry, environments, and included physics.

- The C/C++ ecosystem is the target for a host of **advanced performance improvements**: GPU/accelerator drivers, compile-time optimization, vectorization, etc.



Geometries inaccessible to current frameworks

There are simulations that *cannot* be properly simulated by existing tools, including:

- **High zenith angle** UHECRs, where Earth *curvature* becomes important.
- ANITA, which observes the radio emission from UHECRs after it **reflects** *off the Antarctic ice*, as well as *stratospheric* EAS, that are atmosphere skimming and never intersect the ground.
- **Up-going showers** potentially from the decay of τ-leptons, created by Earth-skimming ν_τ (but AIRES & CORSIKA 7 have *modified* versions to produce such showers).
- Orbital observatories, with FoVs that require accurate treatment of the spherical Earth.

And more!

Extending the physics can be challenging

Existing tools are not modular

- *Most,* but not *all*, tools still assume a shower in the ***Earth's atmosphere*** (i.e. no in-ice showers) and do not support ***mixed media showers*** (i.e. showers developing in air and *penetrating into* ice)

- The **lack of modularity** in these large *monolithic* air shower tools can make it challenging to extend the included physics: either modifying existing models, adding new processes, or complete BSM extensions to air shower physics.

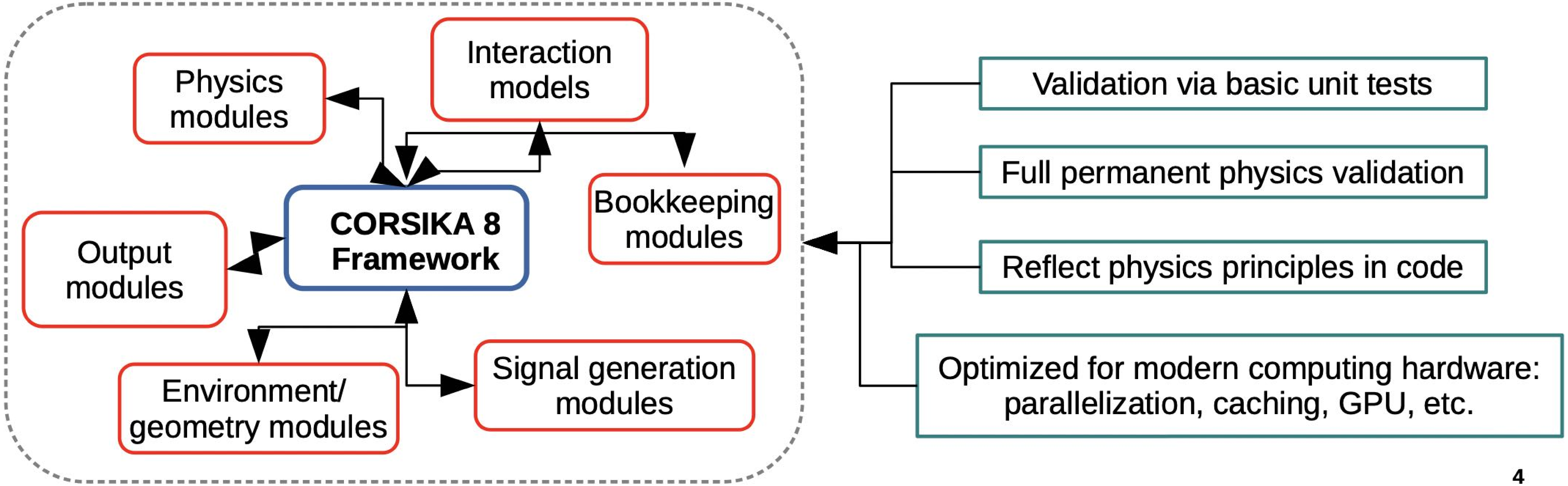
- **Developing new extensions**, such as new radio propagation (raytracing) codes, or experiment-specific optimizations, can be **challenging** and requires familiarity with the internals of these decades-old monolithic codes (although COAST for CORSIKA 7 makes this easier!)

What is CORSIKA 8

A new framework for cascade simulations in matter

- **CORSIKA 8 (C8)** is a completely *new* simulation **framework**, designed from the ground up in *modern* C++, to serve the *next generation* of particle astrophysics (and physics) experiments.

- C8 is designed to be **modular**: there are no *required components*, physics, or structure. **Everything** is a “*pluggable" user-replaceable module* (particle stack, physics processes, environment, geometry, etc.).



CORSIKA8 is a Geant4-like framework for high energy particle cascades…

…and includes all the standard modules to immediately start simulating particle cascades

High-level goals of the CORSIKA 8 project

- A fast, flexible, **modern** framework for high- and ultrahigh-energy cascades using all the modern C++17 "bells and whistles".

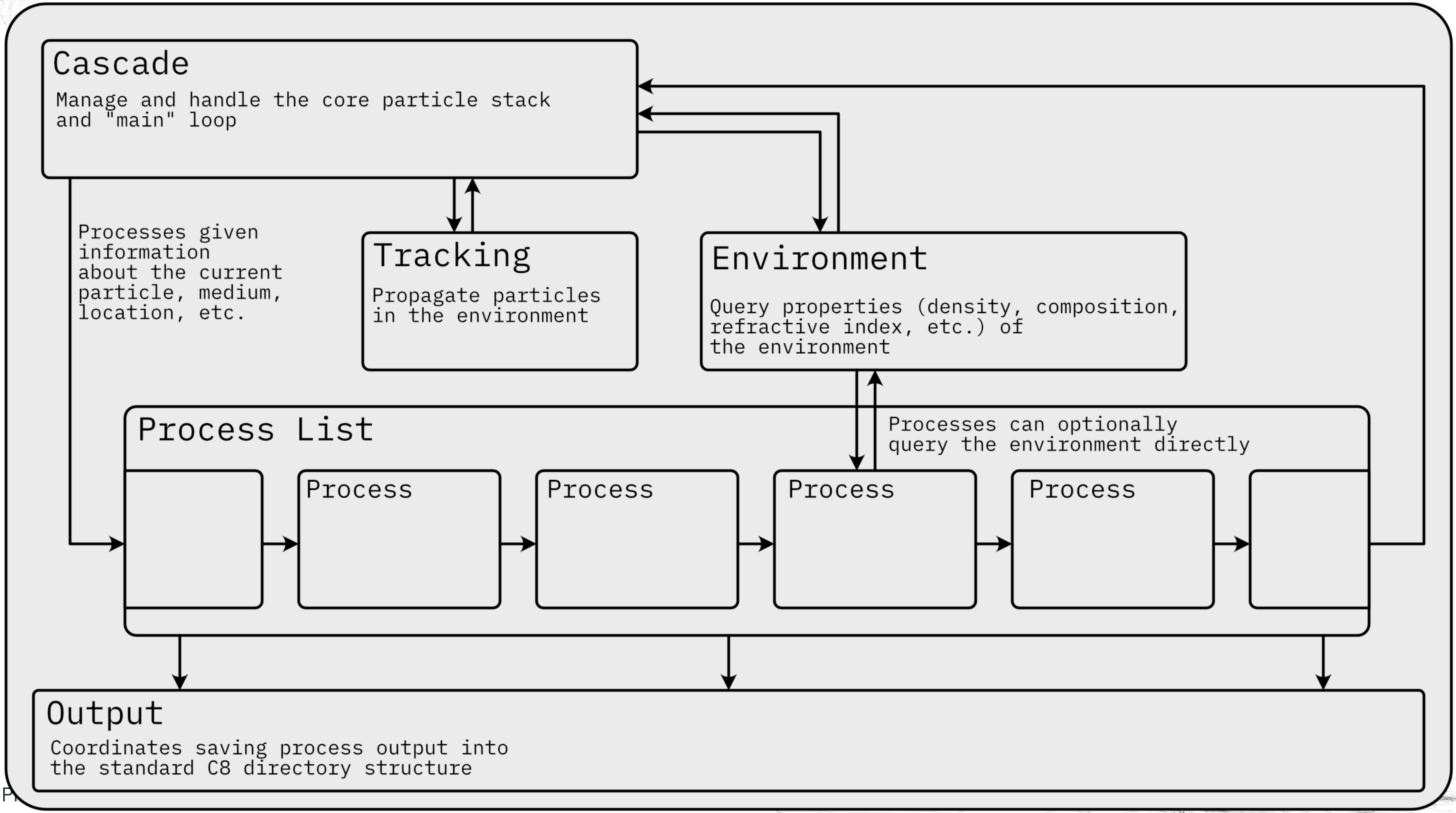
- Standard **pluggable**, *optional*, **modules** (SIBYLL, PYTHIA, UrQMD, radio, etc.), but easy to write *custom* modules. *"You can use our pre-made modules"*, or you can just **build your own** as long as they match our interfaces.

- Support arbitrary **geometries**, **environments**, and **mixed media** (a la Geant4) with showers propagating in user-defined 3D space.

- **Fast** -> Built-in support for multi-core CPU, parallelization, and hardware accelerators (GPUs, co-processors, etc.)

- Configured in C++ with an official Python library for reading & writing CORSIKA 8 data files.

CORSIKA 8 structure

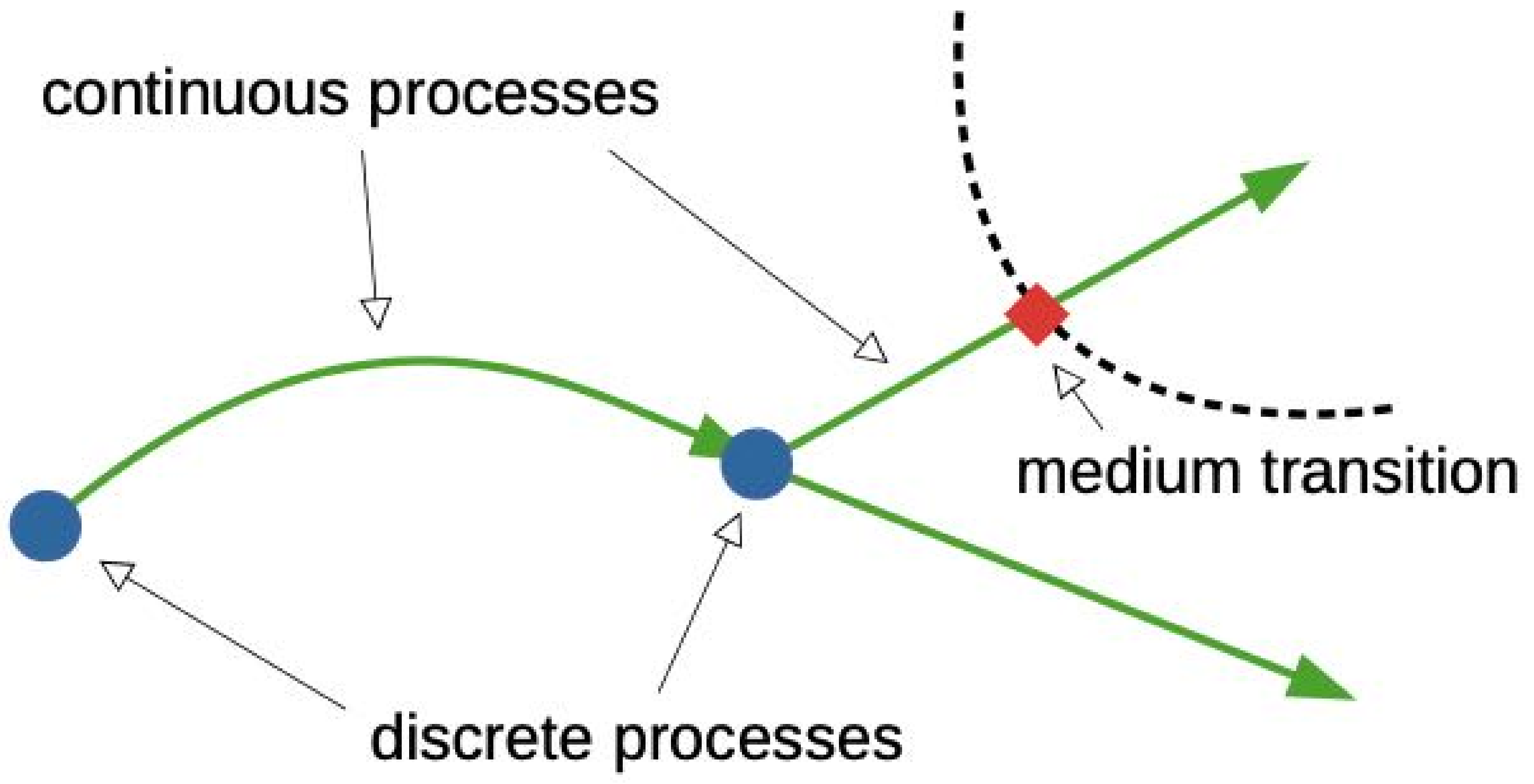


Processes

The main method of working with C8

- `Processes` are how functionality is added into CORSIKA 8: they get given information about the current particle, track, medium, etc. They can be **completely passive**, like the `ObservationPlane` process, or can **change the state**, i.e. particle *decay*!

Physics lists:



discrete processes:
- collisions
- decays, interactions

continuous processes:
- energy losses
- cherenkov, radio emission

boundary crossing processes:
- observation level
- transition radiation

All of CORSIKA 8 is implemented as processes

There is no "special" functionality

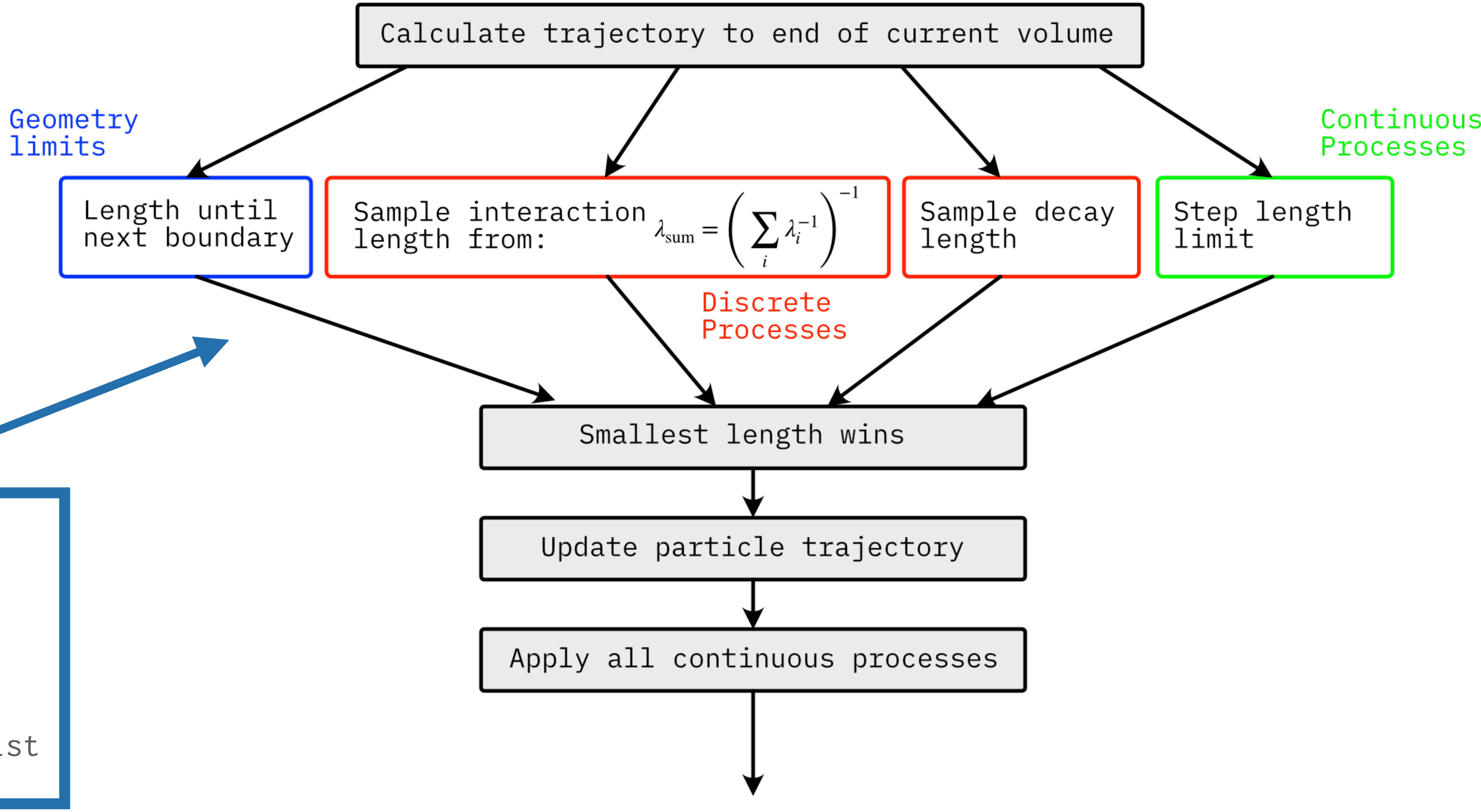
- There are **no** *special processes* or *internal* code. All of the **core physics modules** are regular *user* processes. The C8 developers use the **same interface** as a user module does!

- `ObservationPlane's` (that save particle distributions at specific locations), `TrackWriter's` (saving all tracks that meet some condition), `EnergyLoss`, `PYTHIA`, `SIBYLL`, `UrQMD` (etc.), are all *"normal"* **user processes**.

- Users can choose to use **any** combination of these processes - none are required! As part of the setting up the simulation, users provide a "physics list" or "process list" that configures the physics of the simulation.

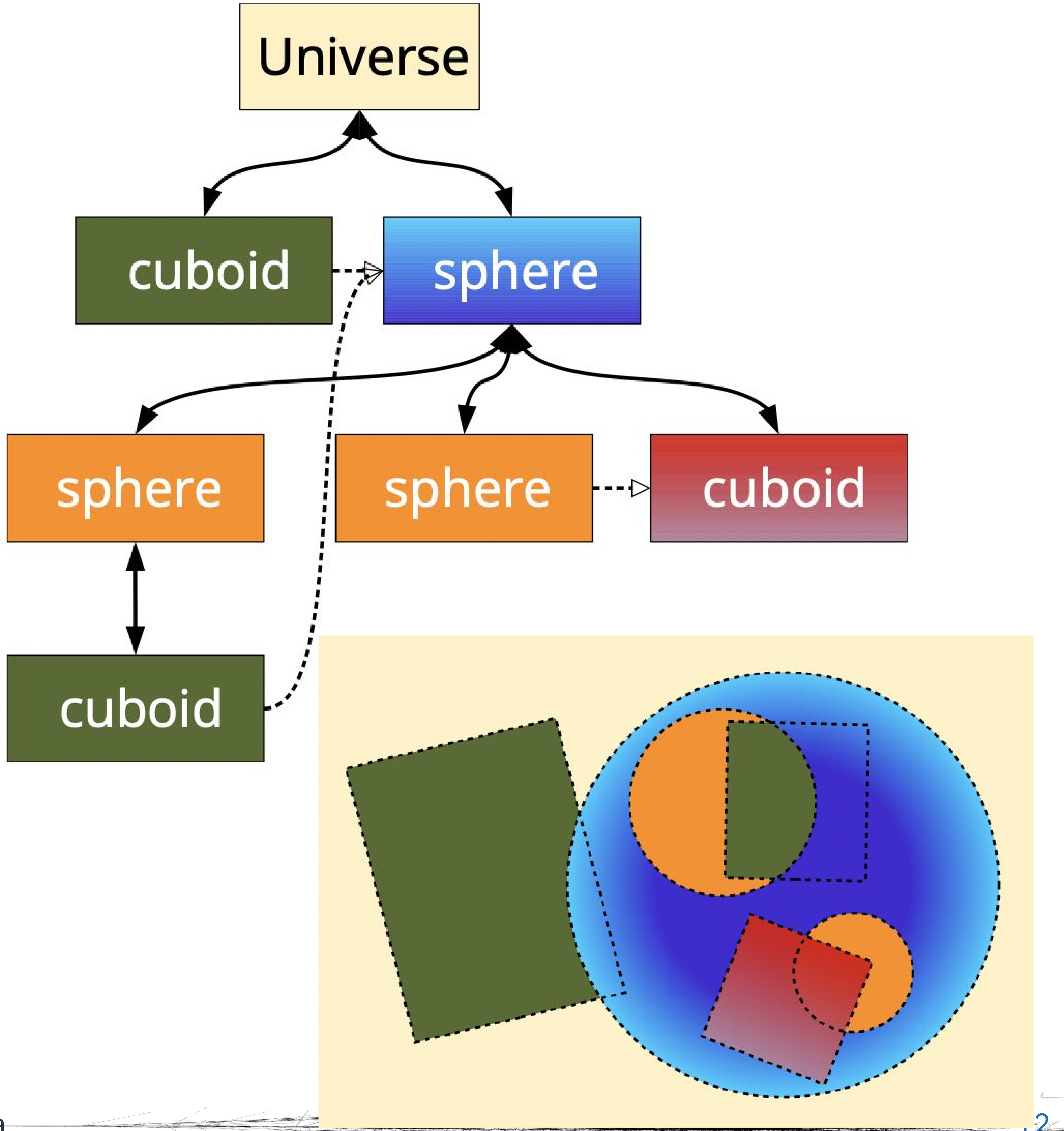
The core loop of CORSIKA 8

```
// main simulation run loop
run() {

  // loop over all the particles in the stack
  while (!stack.isEmpty()) {
    while (!stack.isEmpty()) {

      // get the next particle off the stack
      auto next = stack.getNextParticle();

      // step the particle to the boundary
      step(next);

      // push the particle through the physics list
      sequence.doStack(stack);
    }
  }
}
```



Composable environment and geometry

- CORSIKA 8 has a fully **composable geometry system**, much like Geant4.
- The simulation is composed of **geometric primitives** (spheres, cuboid, cones) - each with their own composition!
- Material properties can **depend on the location** of the particle *within* each primitive i.e. *radially exponential* atmosphere inside a spherical primitive.
- Composition and media is provided to processes at each step!



Particle tracking

Magnetic fields supported (electric fields soon)!

- CORSIKA 8 supports **multiple tracking algorithms** for propagating showers.
- The chosen particle "propagator" can be swapped out by the user during configuration.
- The standard propagator, a relativistic Verlet integrator, is used whenever a magnetic field is configured (decided at compile time).
- In the no-magnetic field case, a simpler tracking implementation is used that is faster and more numerically stable.
- A new integrator supporting simultaneous **electric** and **magnetic** fields is in progress and should be available within several weeks.

Modern programming techniques

C++17 required!

- CORSIKA 8 is being developed with modern **C++17 programming concepts.**
- Entire code base is **compile-time unit-correct** using compile-time unit annotations and conversions (`3_m`, `5_GeV`, `1_ns / 1_GeV`).
- We make extensive use of C++ template parameters to **swap out functionality**: *any* user-provided class can be used for *any* part of C8, including the "core" particle stack, as long as it meets the required interface!
- Continuous integration tests run for every commit, checking configuration and physics.

Check the **code** here: https://gitlab.iap.kit.edu/AirShowerPhysics/corsika/

How to use CORSIKA 8?

- CORSIKA 8 is **configured via C++**. Using C++, you
  1. Assemble the required environment, geometry, and media.
  2. Configure your required physics processes and cuts.
  3. Setup your output (observation planes, radio, track writers, etc.)
  4. Populate the initial stack (an unlimited number of particles supported)
  5. Run the simulation!

- We provide **scripts** to run **standard simulations** that can be configured from the command line, no C++ required (down-going showers, in-ice showers, etc.). More scripts will be added as we continue development.

- We also provide a **Python package** for reading and analyzing the **standard output structure.** Default file format is Apache Arrow/Parquet but users can easily replace this with other formats.

Hadronic physics models

- CORSIKA 8 already provides several hadronic interaction models **available as processes** and more are in progress (i.e. GENIE-HEDIS):
- High energy models: **Sibyll 2.3d, QGS-JET II, EPOS-LHC, PYTHIA 8, CONEX**
- Low energy models: **UrQMD, CONEX**, (Next: FLUKA, Hillas-splitting)
- Decays: **PYTHIA 8, Sibyll 2.3d, TAUOLA**
- Some of these are Monte Carlo, others are Cascade Equation-based, C8 can already do **hybrid showers!**
- Hadronic models can be **switched between per-particle** using energy, PID, location, interaction, etc., or based on any other user-provided "selection condition" (i.e. different models for early shower vs. late shower).

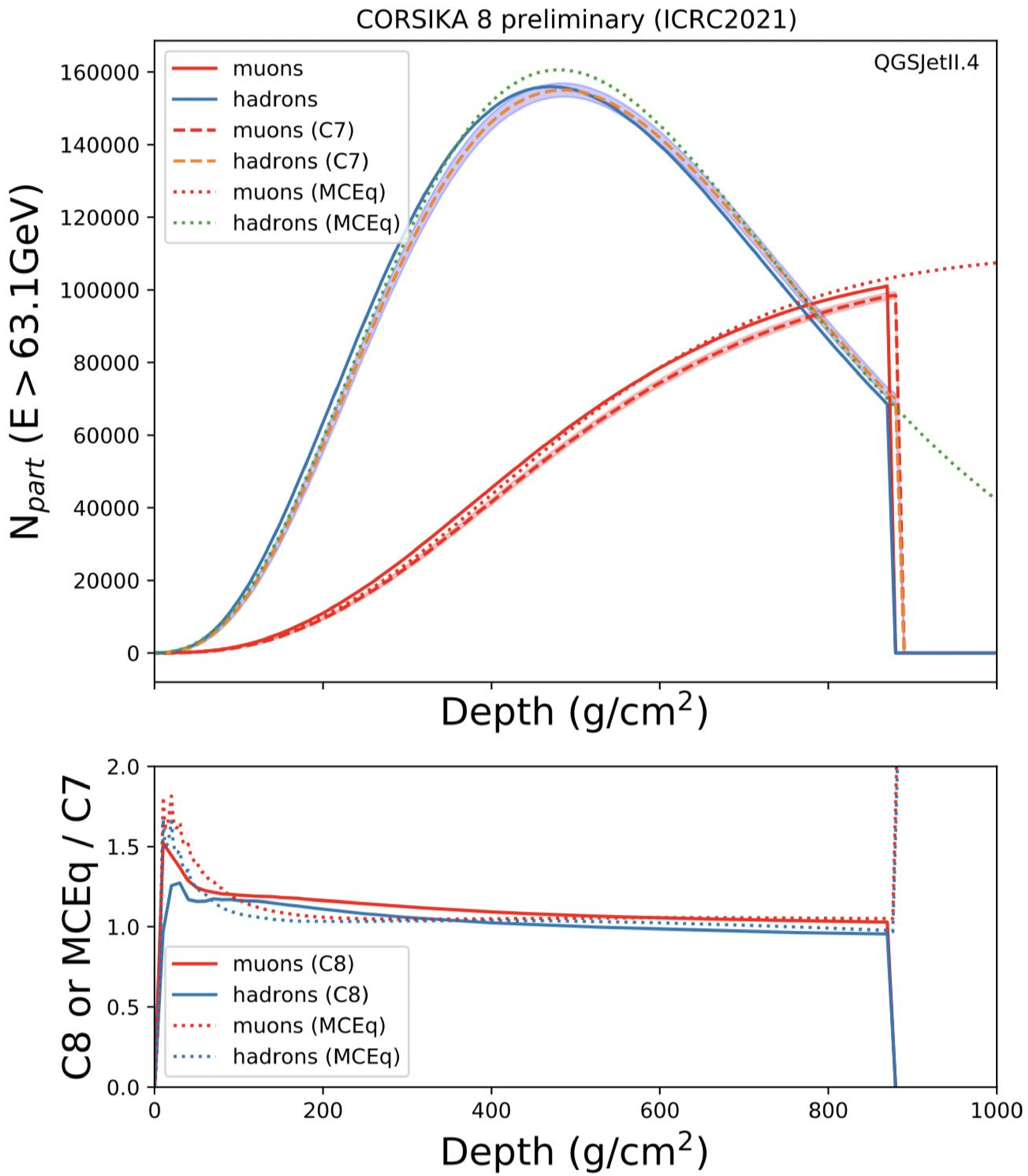
Hadronic physics status

- CORSIKA 8 has many tunable parameters, since we have many different hadronic physics models that can be used!

- C8 can **accurately reproduce hadronic showers** as simulated by C7 or MCEq, when configured similarly.

- However, C8 is not perfect: there are still small discrepancies in particular observables between C7, C8, and MCEq.

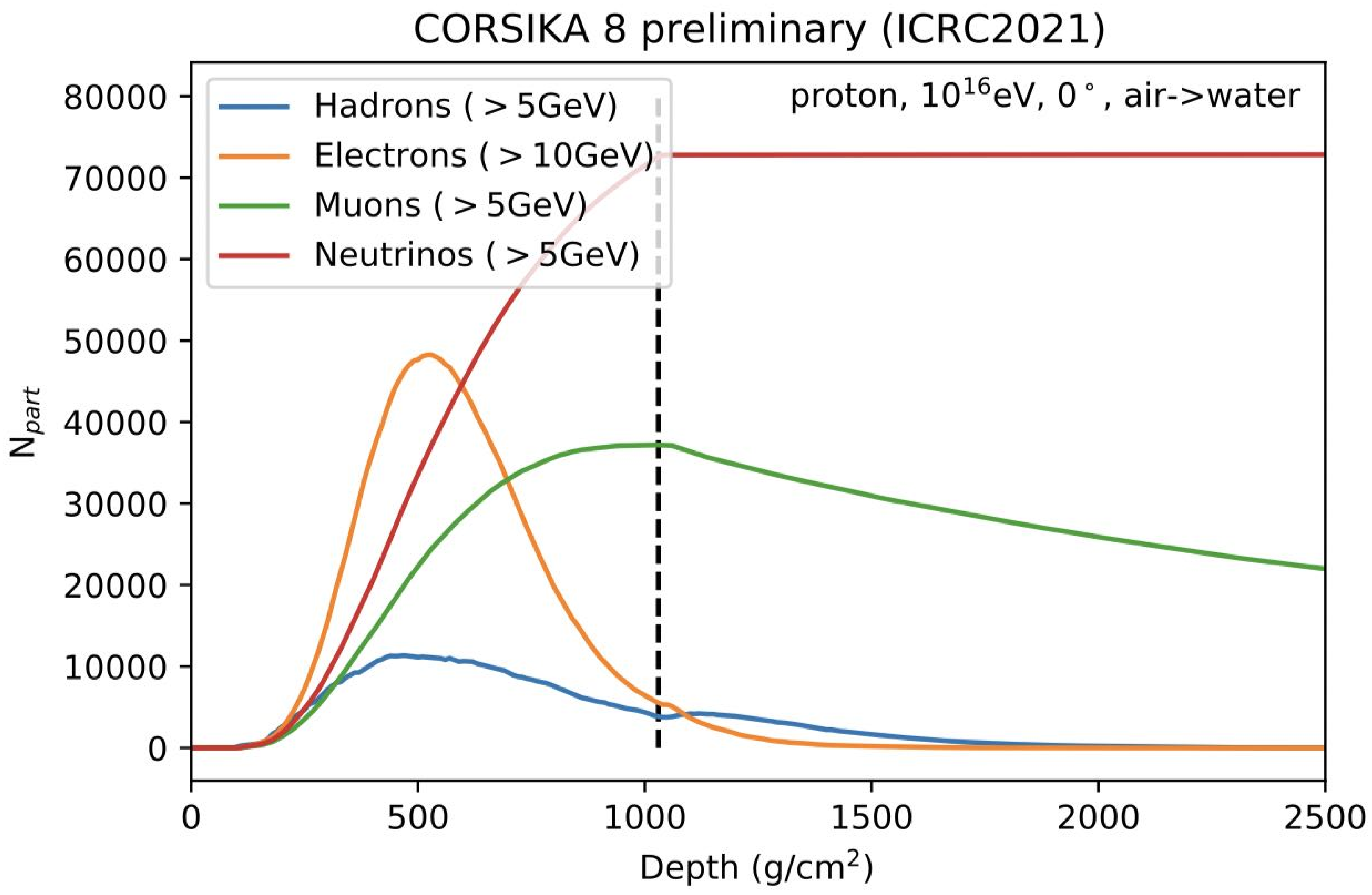
- These are being studied by the C8 hadronic working group.



We already support non-standard showers

Air showers penetrating into ice!

- We can immediately use the flexibility of C8 to do non-standard showers, in this case, an **extensive air shower penetrating into ice.**
- The dashed line shows the transition between air and ice, where the particle evolution changes.
- arXiv:2112.11761 also presents simulations of **extensive air showers on Mars!**

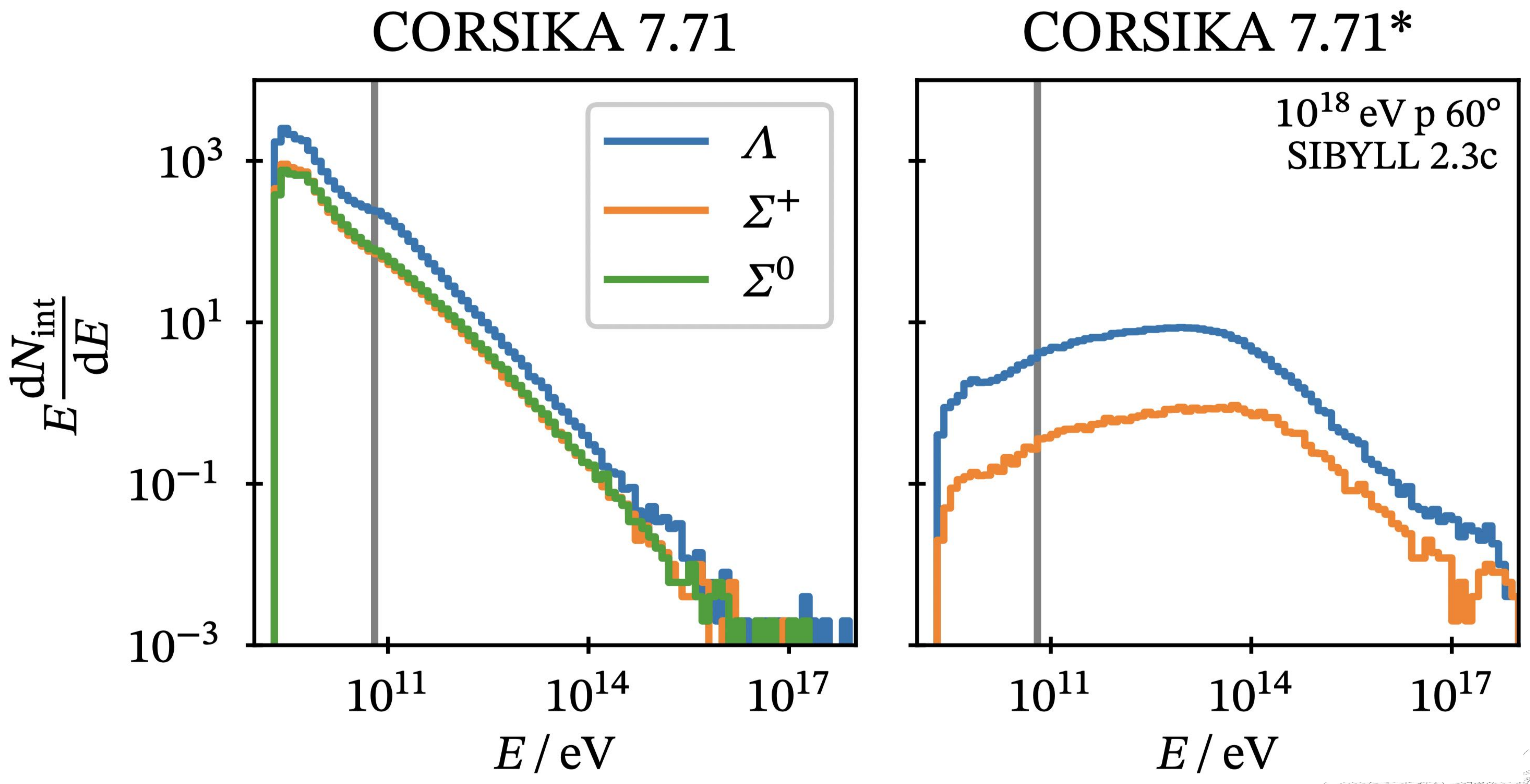


Even discovered bugs in CORSIKA 7!

The "Hyperon" bug

- The CORSIKA 8 collaboration has been extensively studying the hadronic physics for several years - has also identified bugs in CORSIKA 7, including this "hyperon" bug, relating to Σ^0 hyperons.

Before



After

Electromagnetic showers

PROPOSAL

- CORSIKA 8 uses PROPOSAL as the *default* **electromagnetic interaction model** (although, since it's a regular process, it can be replaced).

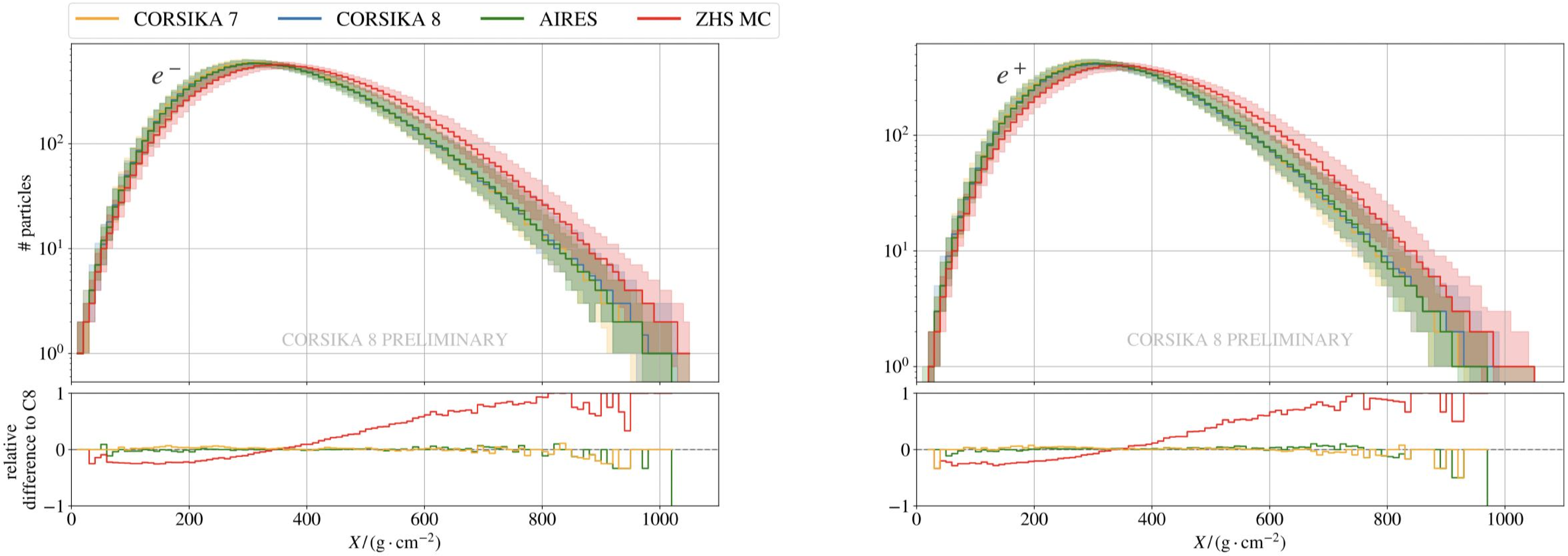
- Unlike CORSIKA 7, CORSIKA 8 does not currently support EGS4 (due to difficulties in integrating it with the broader framework - however the work is *underway*).

- PROPOSAL implements **most**, but not **all**, of the EM processes provided by EGS4. As of the time of writing, photo-electric effects, potentially important at low energies, and some photo-hadronic processes, are currently missing.

- These can all be included and **configured from CORSIKA 8**, with the exception of the LPM effect that, while present in PROPOSAL, is not *yet* available in C8.

Electromagnetic physics status

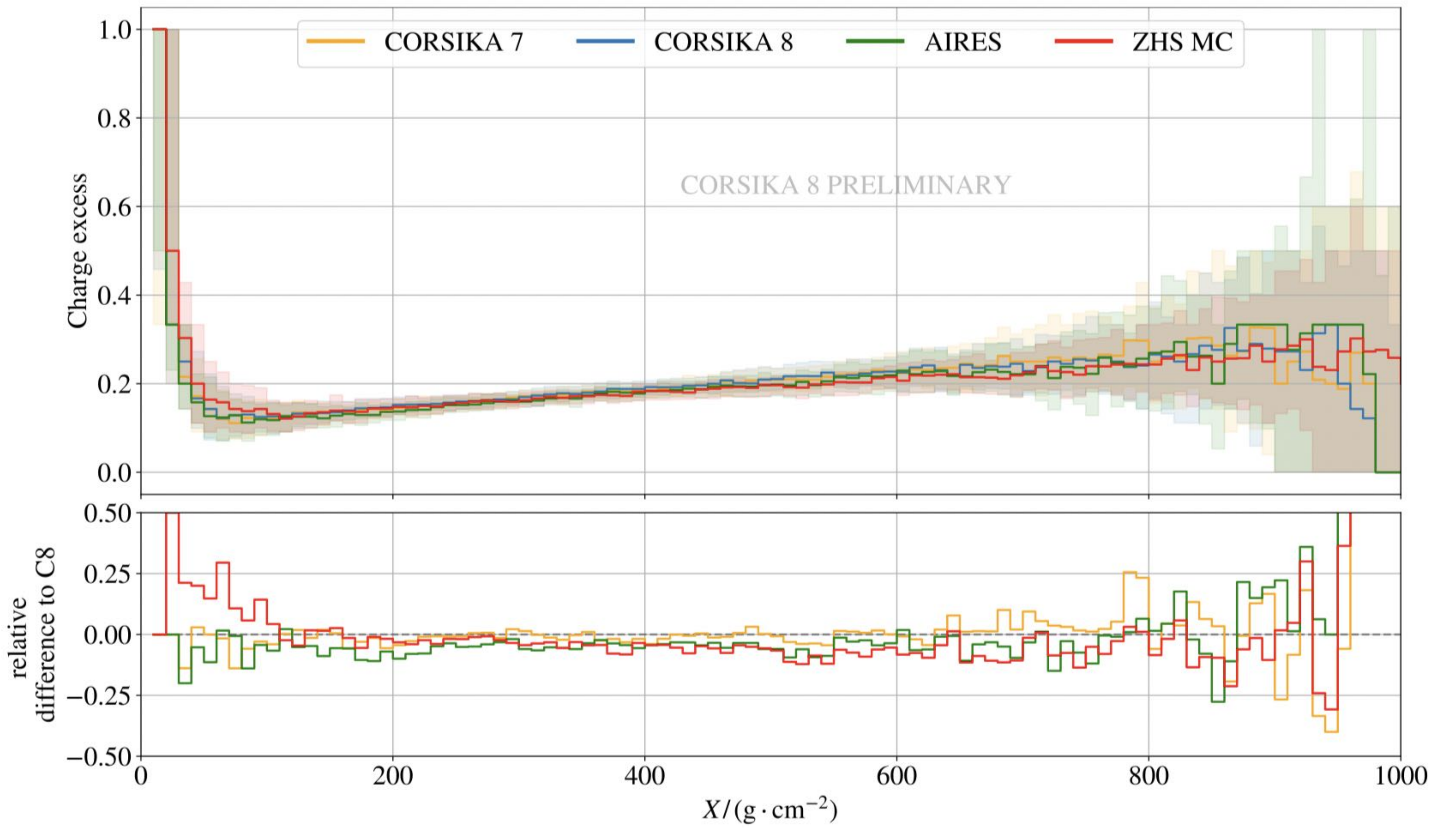
- The overall behavior of EM showers **agree remarkably well** between C7 and C8 (but they both disagree with ZHS). Published results in arXiv:2112.11761 are already *out of date* (new PROPOSAL version and C8 bug fixes)



Electromagnetic showers are not perfect

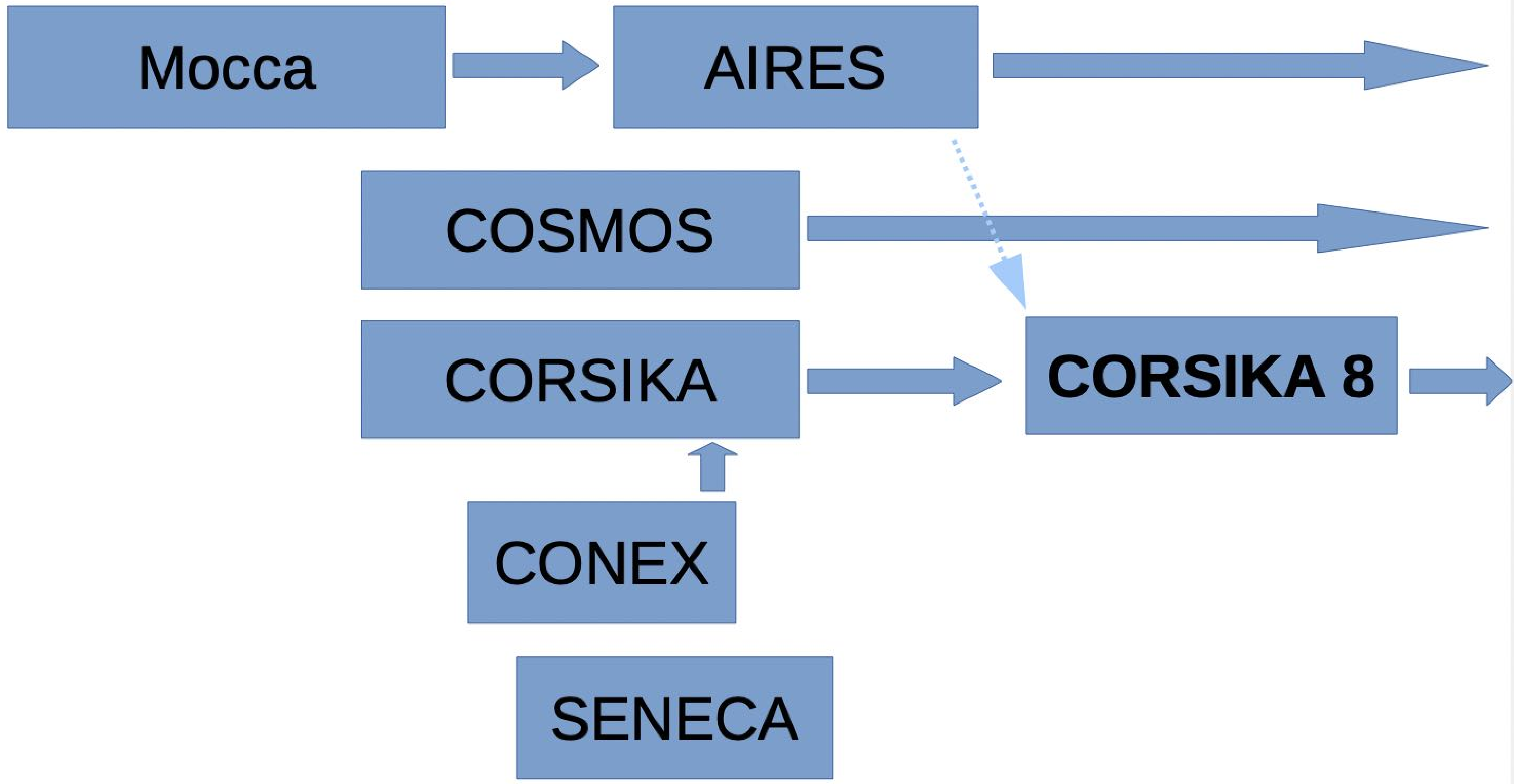
- While the overall behavior of EM showers is remarkably consistent, there are **still issues with EM showers in C8.**

- **Radio** is a remarkably **sensitive probe** to the *spatial*, *angular*, and *temporal distributions* of electromagnetic particles. Detailed simulations by the radio working group suggest that there are still *latent issues* with the timing and angular distributions from PROPOSAL - this study is on-going.



Particle genealogy

- **MOCCA** had the ability to track full particle histories, but AIRES and CORSIKA 7 do not (only two generations stored in C7).
- Full particle histories enable a plethora of studies of the processes that contribute to observations (i.e. the muon excess) or as training data for supervised learning or analyses.

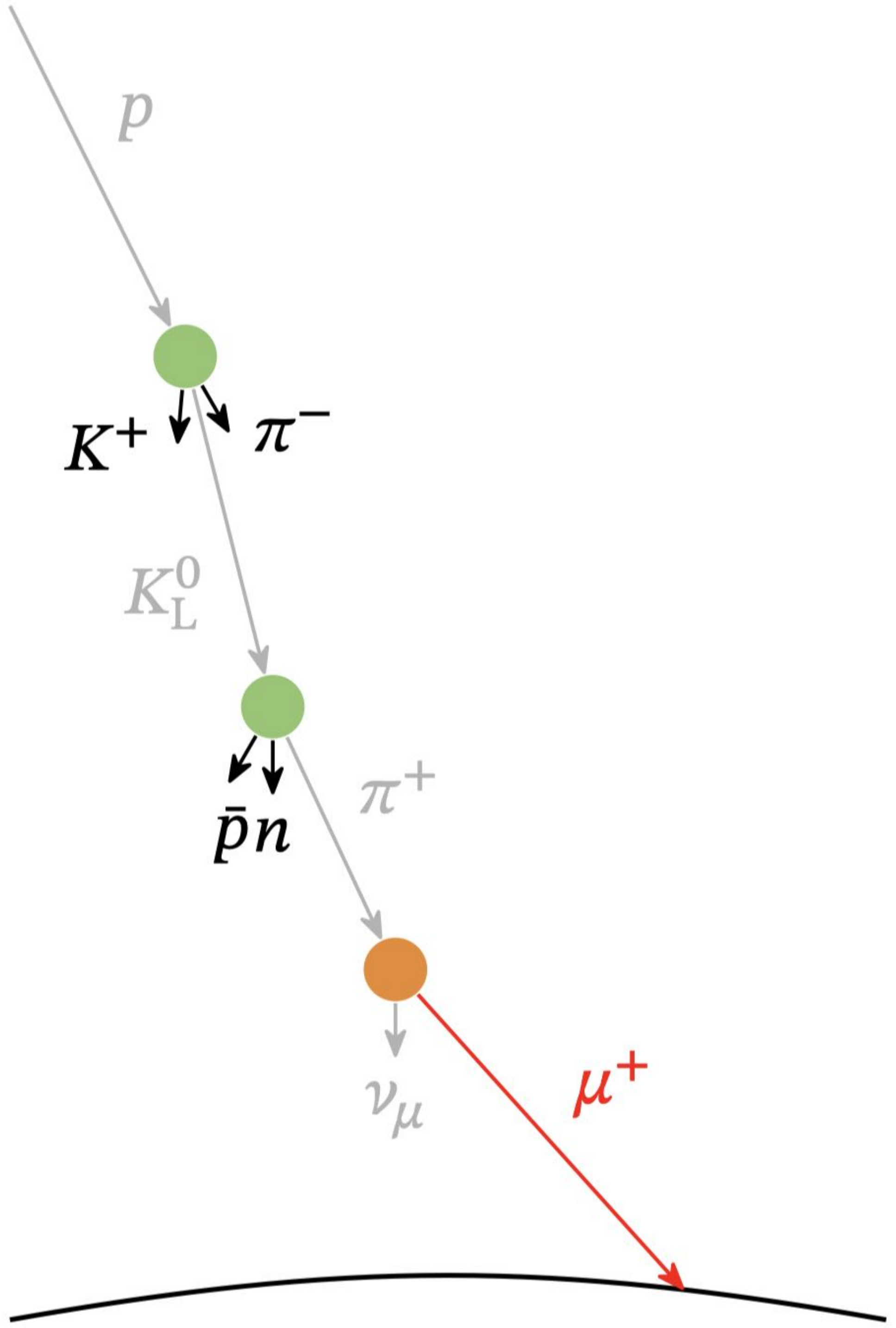


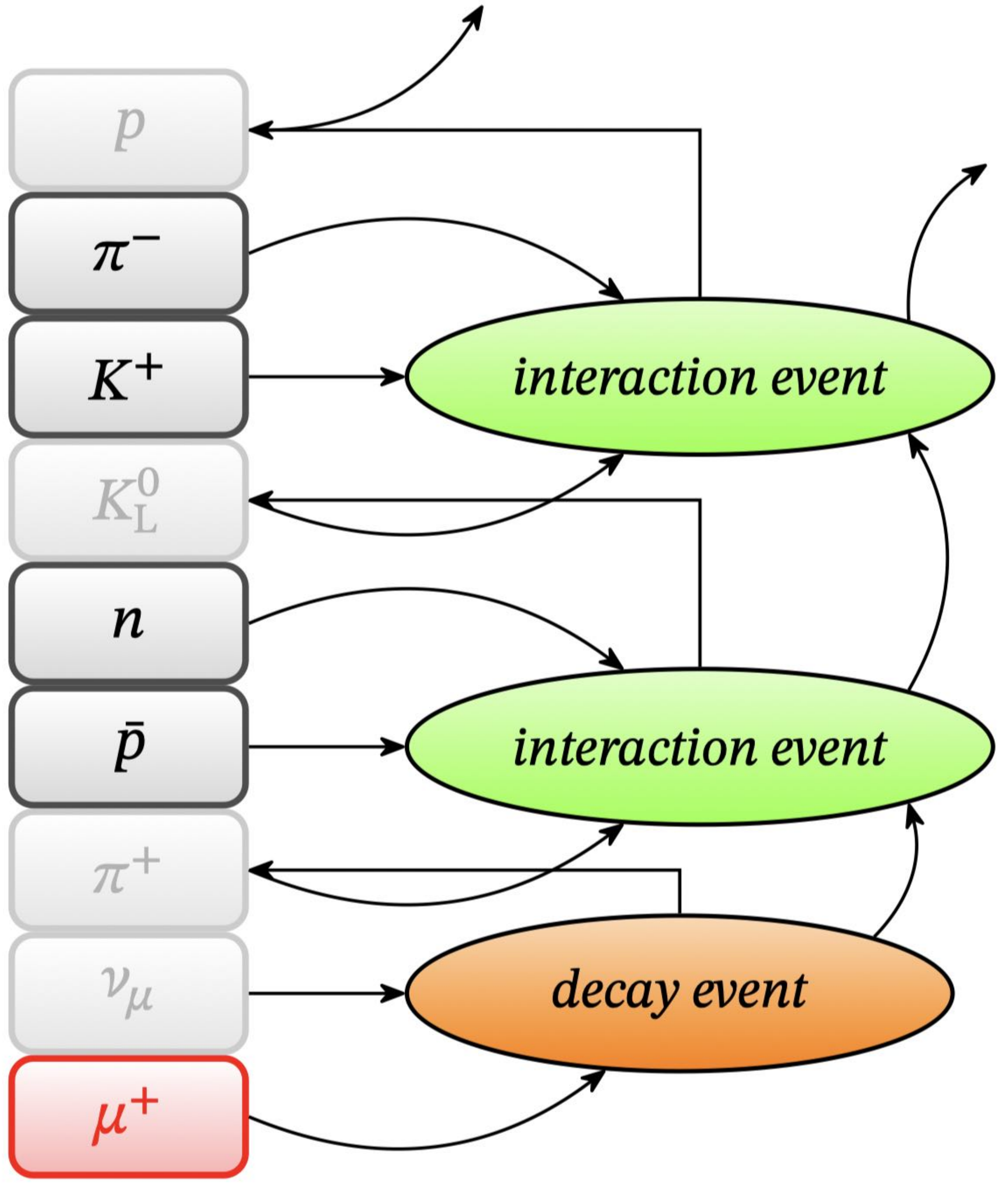
Particle genealogy in CORSIKA 8

- CORSIKA 8 allows for optionally storing the **entire generation history of particles** (this does not have to be the *full* shower - just particles that are *"selected"*, such as muons hitting an observation plane or the ground).

- This provides a **full parent tree**, including intermediate particles, locations, and interactions.

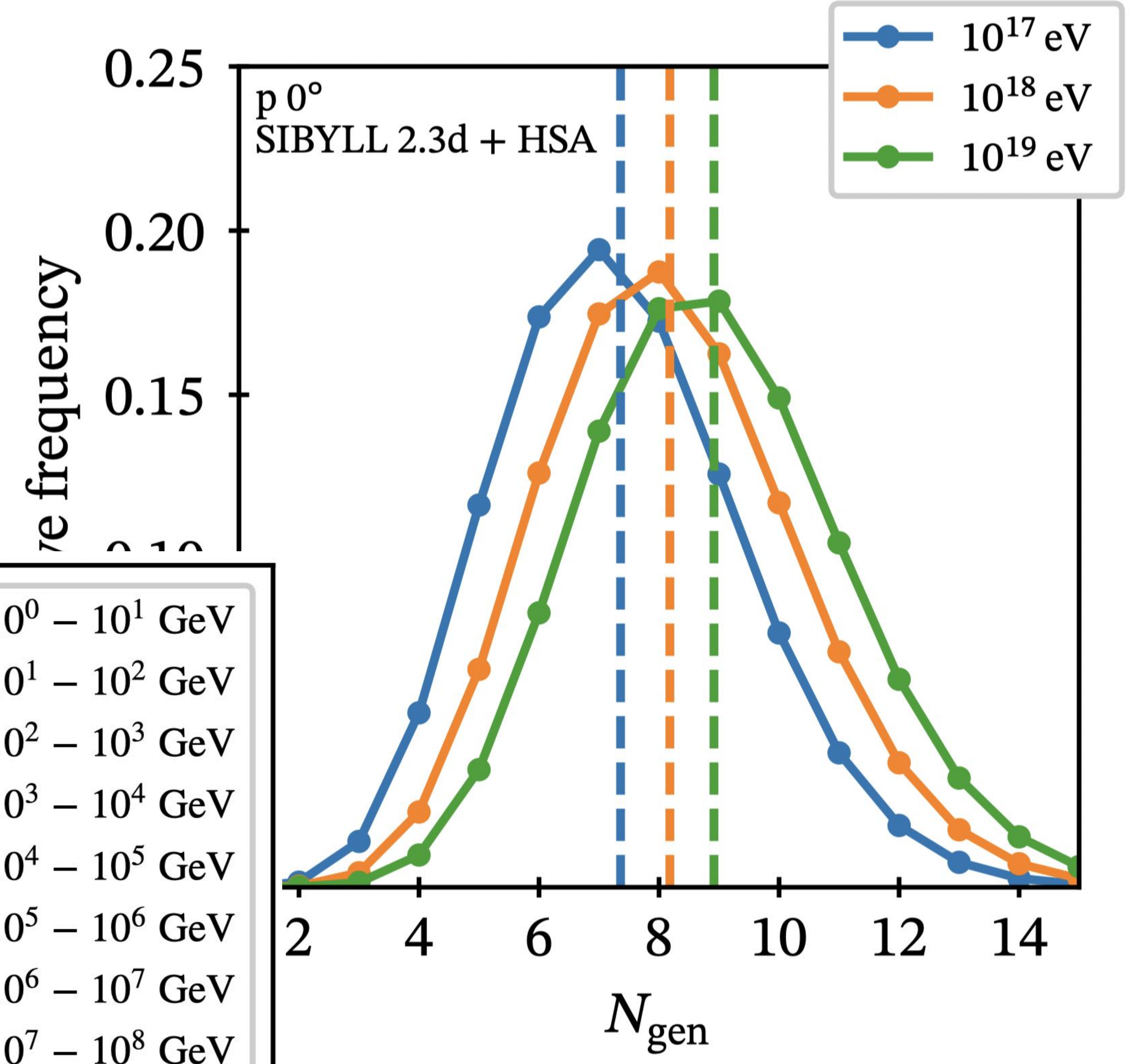
- C8 developers involved in studying the *muon excess* in EAS using this full genealogy ability. *200 MB for the muon histories for a 1 EeV proton.*

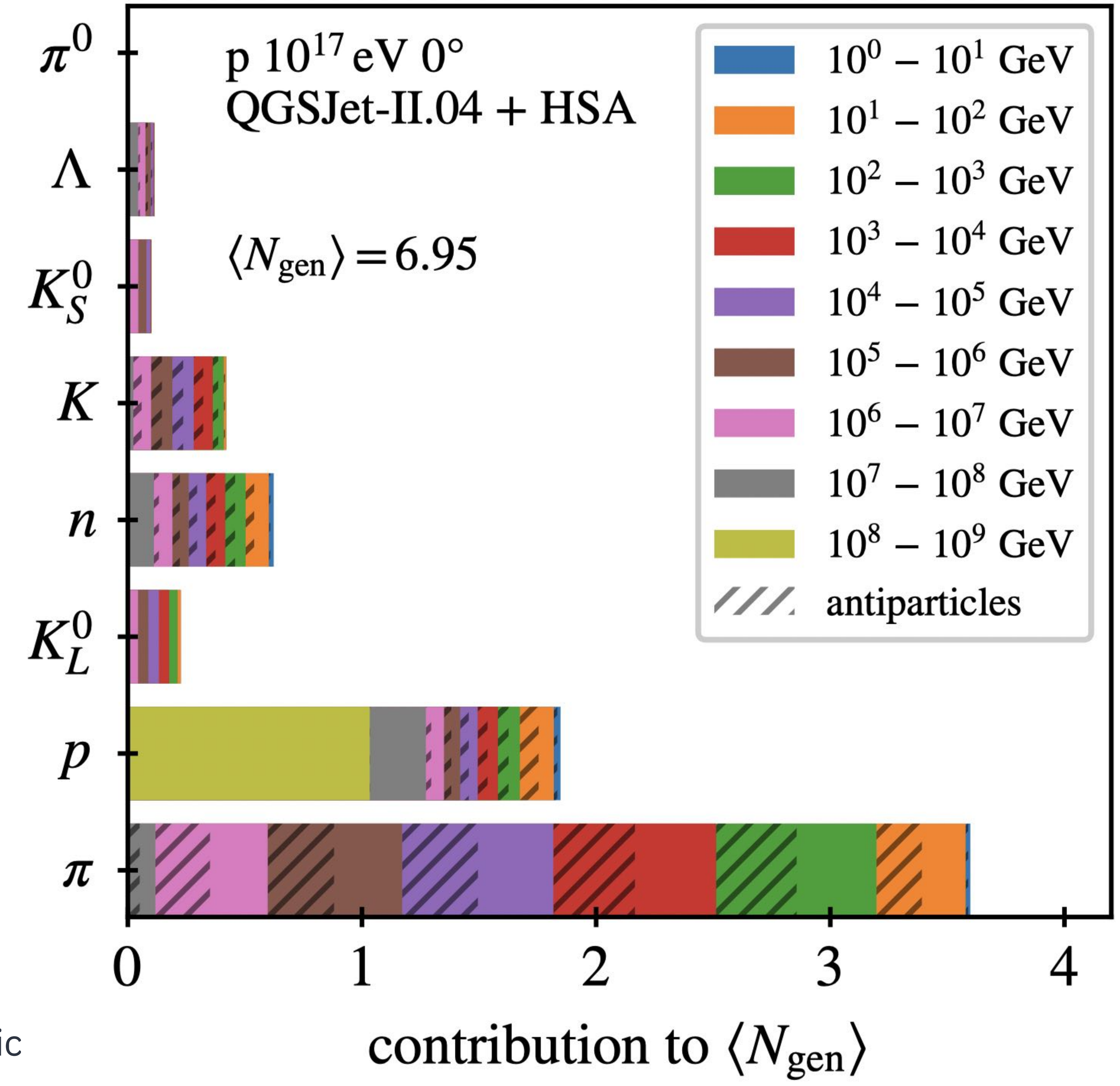




Muon genealogy

- Muon histories allowing for studying the **relative contribution of different particles** to the observed **muon flux** at the ground (bottom figure).
- A plethora of interesting physics studies are underway with this *new* genealogy feature.





Secondary emission

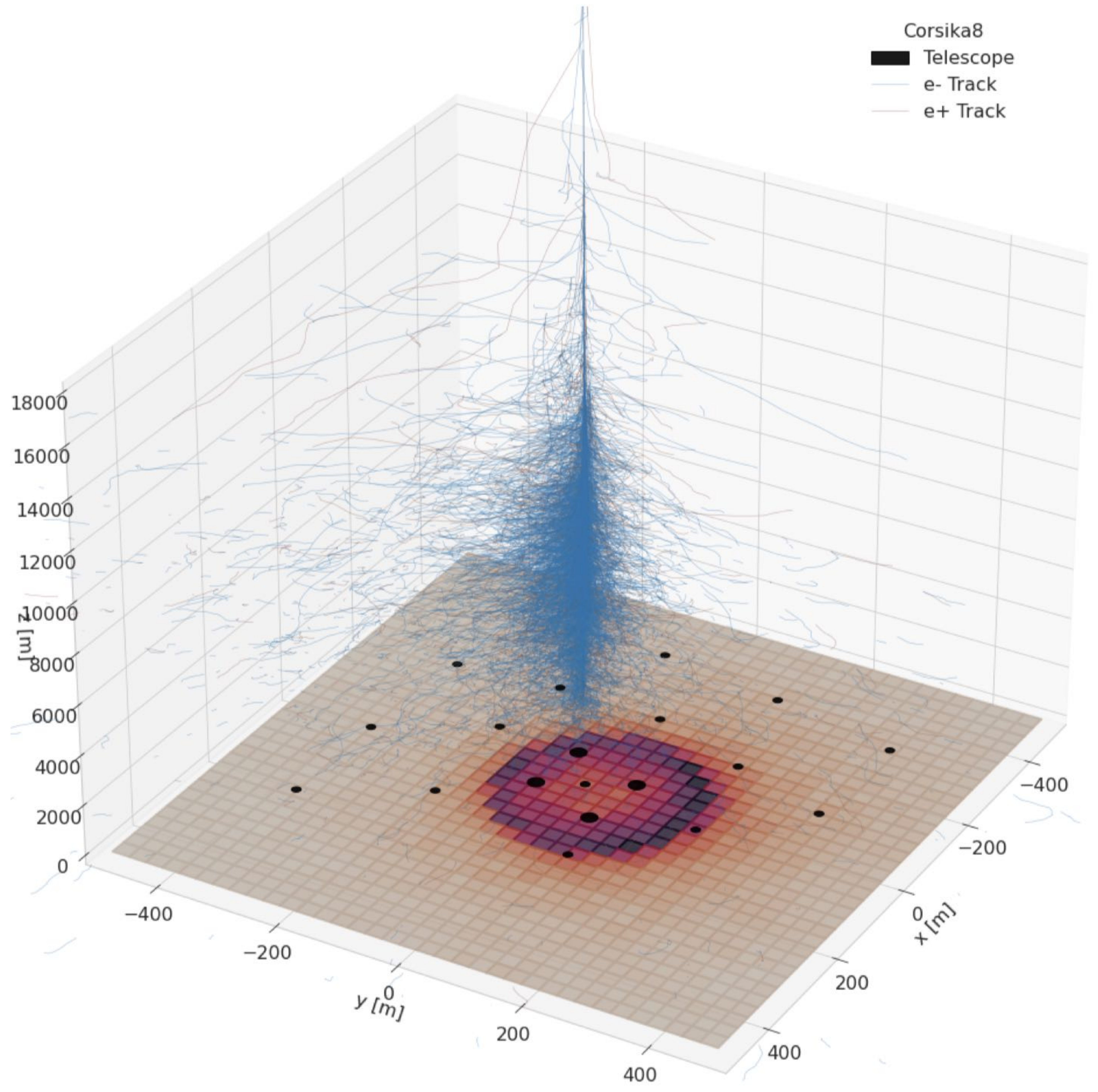
Radio, Cherenkov, Fluorescence, Acoustic, etc.

- Secondary emission models are implemented as *regular CORSIKA 8 processes.*
- The **radio emission** model is *highly developed*, tested, and as of recently, available in all standard C8 installations.
- There are **two** independently developed **optical emission** modules for Cherenkov and fluorescence (one of which is already GPU accelerated for photon propagation). Neither are *officially* merged but available in their own branches in the C8 repository.
- Others are in early stages (i.e. acoustic emission) - more help always welcome!
- Multiple “similar” emission models *are allowed* in C8 - no “official” modules.

Optical emission

CPU & GPU accelerated versions available

- There are **two independent optical emission** modules in development and are available for testing from the C8 repo.
- Given the same input track distributions, both **almost exactly reproduce** the photon yields simulated by CORSIKA 7.
- One of these modules is **vectorized** for CPU operation, while the other is **GPU-accelerated** (the GPU performs all of the photon propagation).
- While neither has been *officially merged* into C8, *yet*, we expect an official release this year.



Radio emission

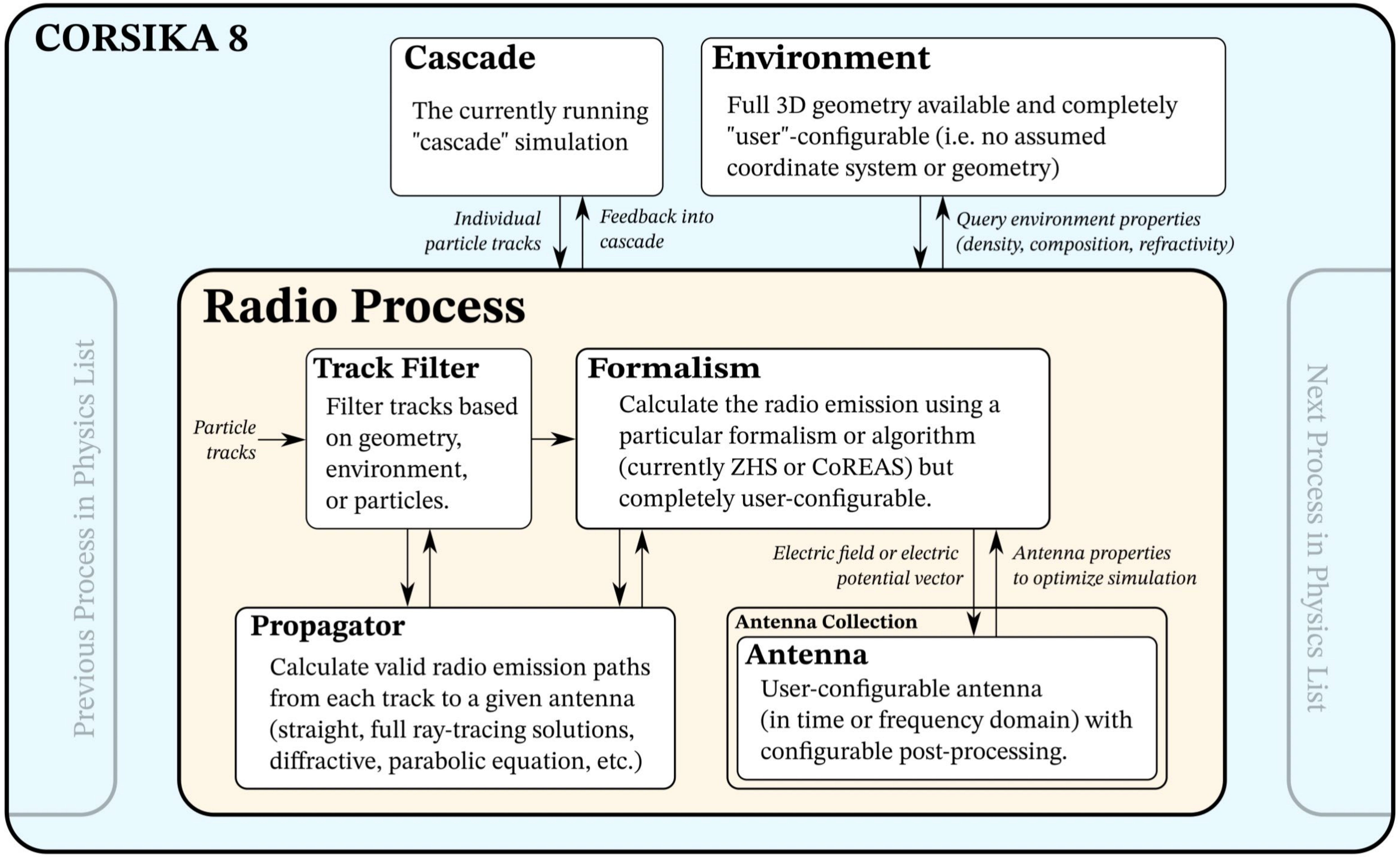
ZHS and 'Endpoint' supported!

- The built-in radio module for C8 is *highly developed* and has been *extensively tested*. It supports **both** of the standard radio formalisms, **"Endpoint"** and **"ZHS"**, and they can be run **simultaneously** on the **same** shower.
- The C8 implementations agree **exactly** with analytical calculations (for single particles) and existing third-party codes (for full showers) when *given the same reference particle tracks* and *setup.*
- However, pulses for C8-simulated showers do not exactly match C7 or AIRES. Our confidence in this implementation is being used as a **tool to study** the electromagnetic shower development in C8 - it's still not perfect!
- Pulses from C8 showers agree very well at the Cherenkov angle but disagree far away from the shower axis.

Radio emission module is customizable

- Current radio emission module is **parameterized** and allows for the user to **“swap out”** emission models, propagation algorithms (i.e. raytracing, FDTD, reflected), track and particle filters, and antenna models.

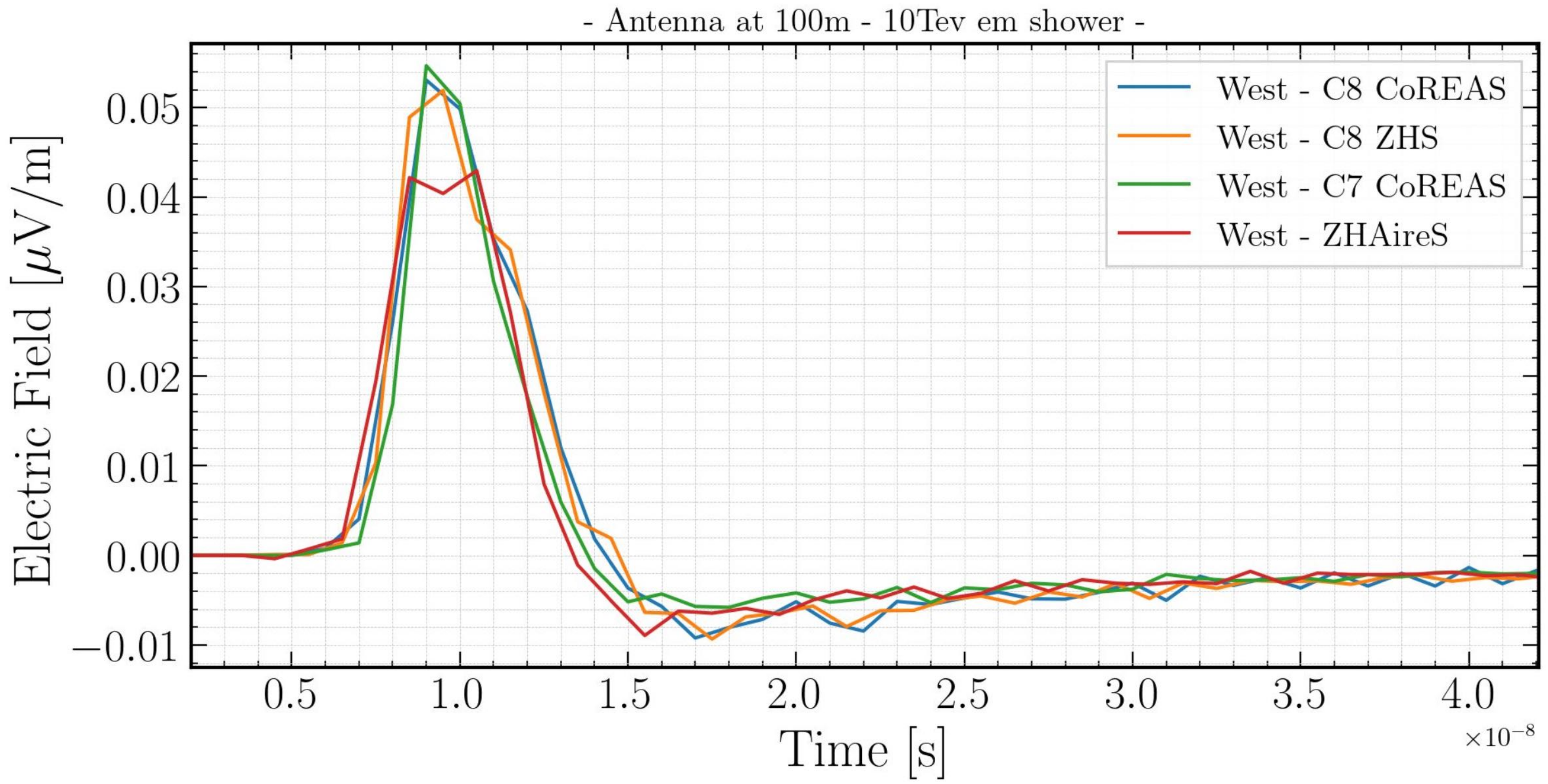
- Antennas can be perfect antennas, in the time or frequency domains, or immediately convolve in detector or antenna responses (or anything else that is needed for an experiment).

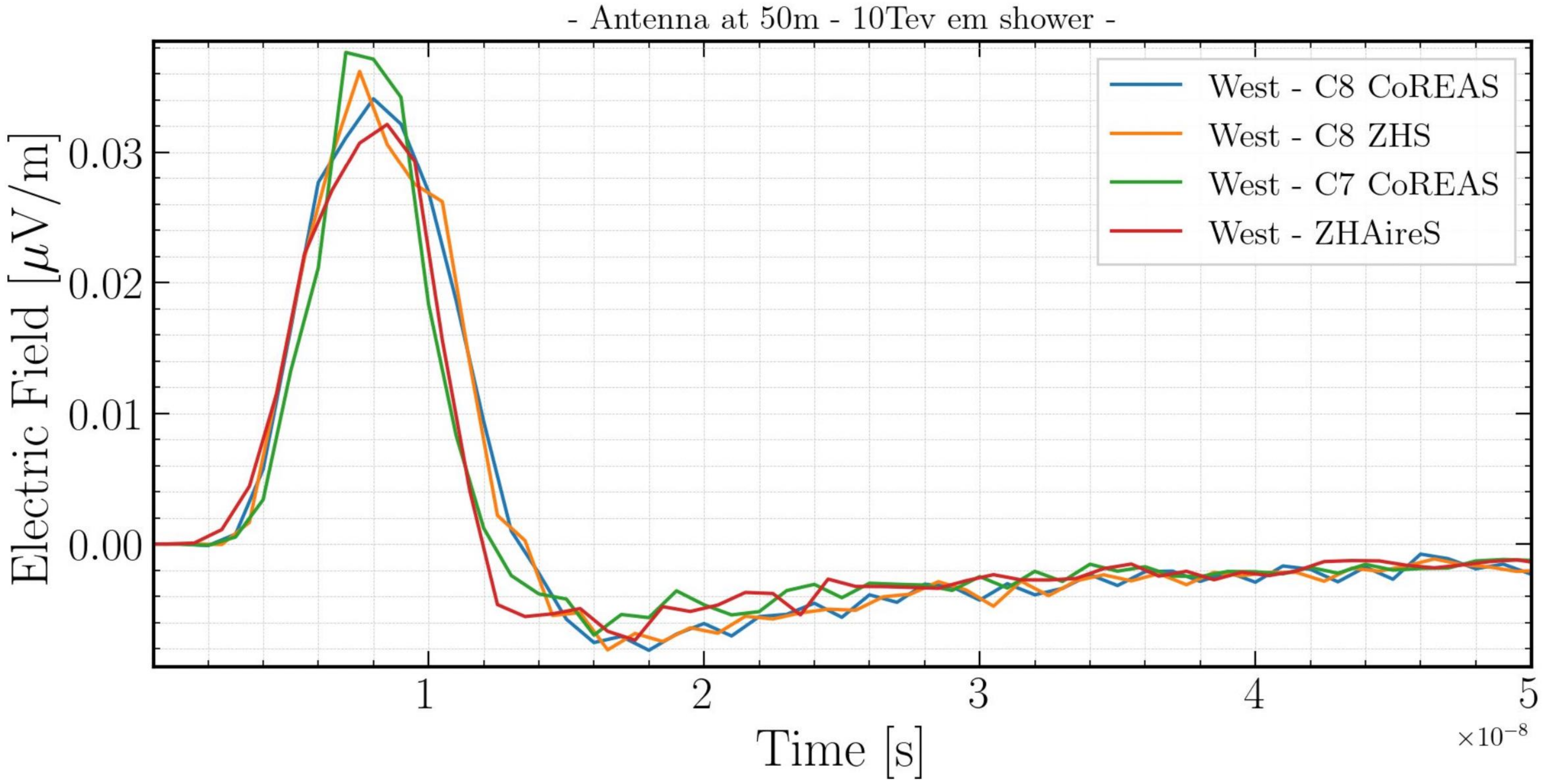


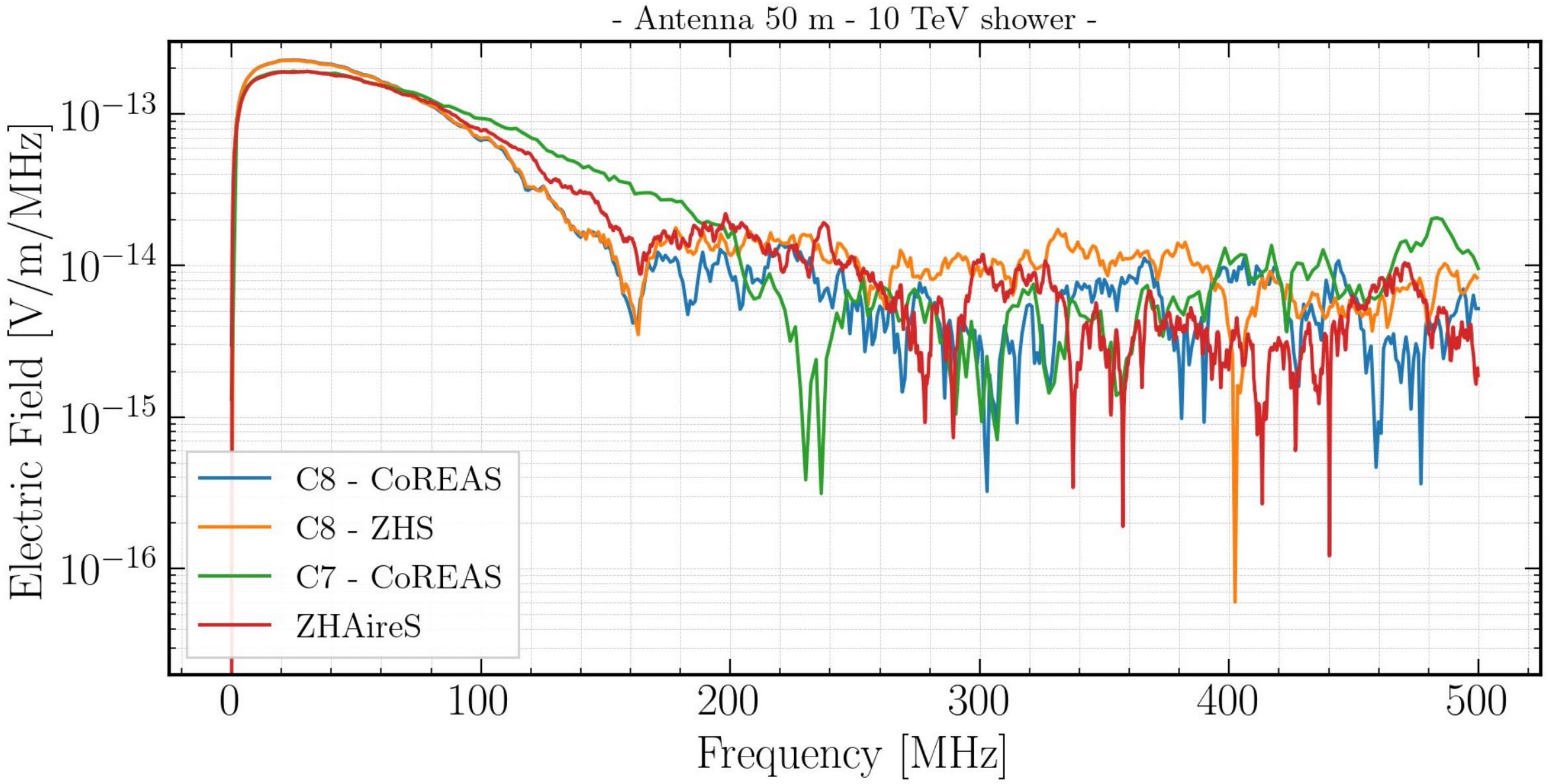
Electromagnetic showers are not perfect

Radio is a powerful diagnostic!

- Pulses agree within a few % near the Cherenkov cone but can accumulate 50%-100% errors in amplitude or pulse width for antennas far *outside* the shower axis.





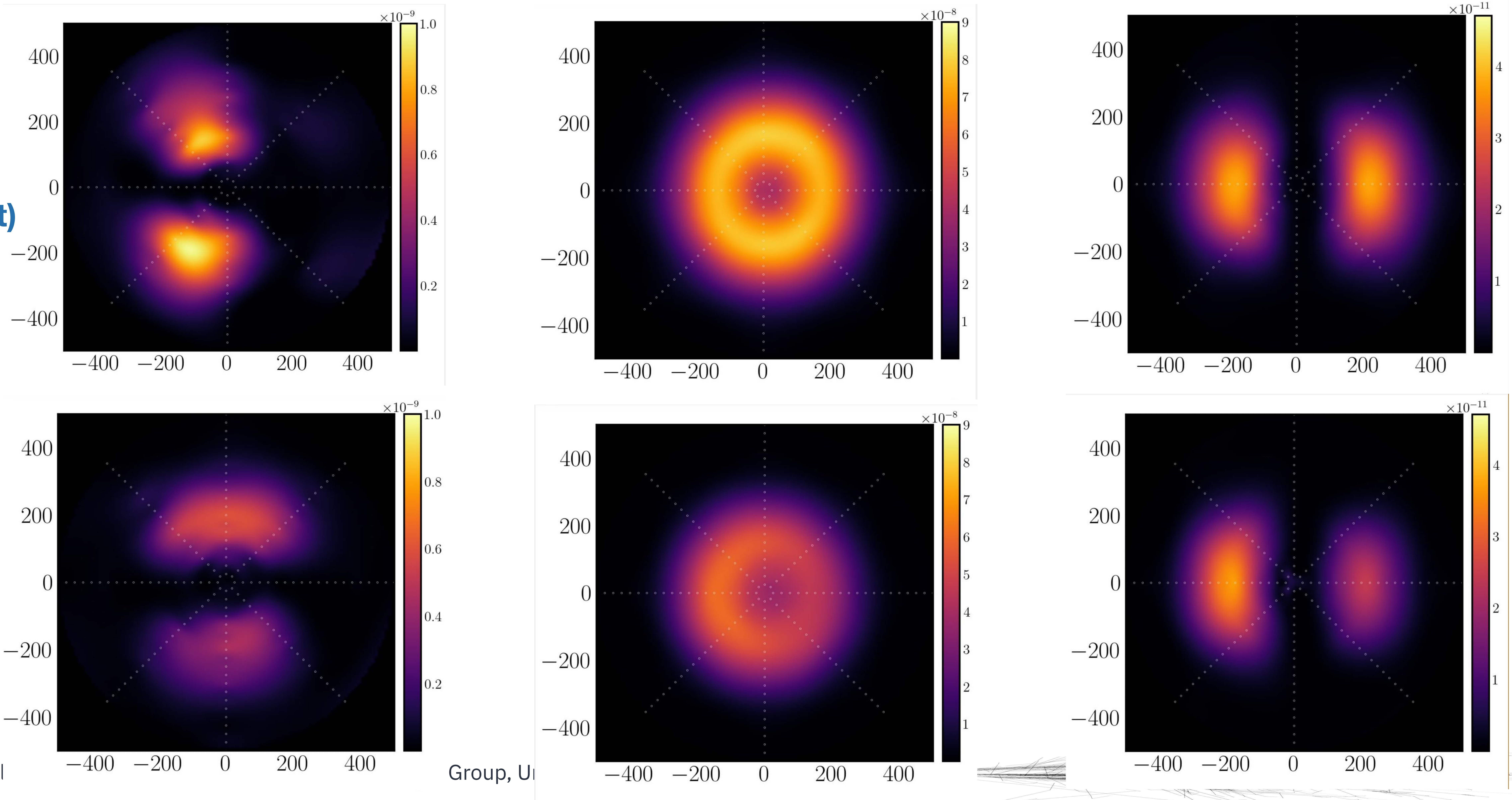


Radio Fluence Maps

Detailed studies between C7, C8 (Endpoint), C8 (ZHS), and ZHAireS...

C8 (Endpoint)

C7



Immediate future plans for CORSIKA 8

Just a summary!

- While CORSIKA 8 is currently *single-core-only* at this point, with GPUs working or in progress for secondary emission, there is an ambitious plan for **parallelization**, **performance tuning**, and **GPU support**. There is a dedicated monthly "performance meeting" that is starting to work on **multi-core C8**.
- Weekly radio meetings, bi-weekly electromagnetic meetings, and monthly general meetings - **all open to the public.**
- CORSIKA 8 is *useable*, **today**, but still requires some more physics investigation to resolve the last few *outstanding EM and hadronic issues*.
- CORSIKA 8's API is *fairly stable* at this point however, we plan a **full v1.0.0 official release** later **this year** (at which point we will commit to no breaking changes).
- We have a Slack for discussion - email me if you would like an invite!