



Snakemake pipelines @ the LHCb experiment at CERN

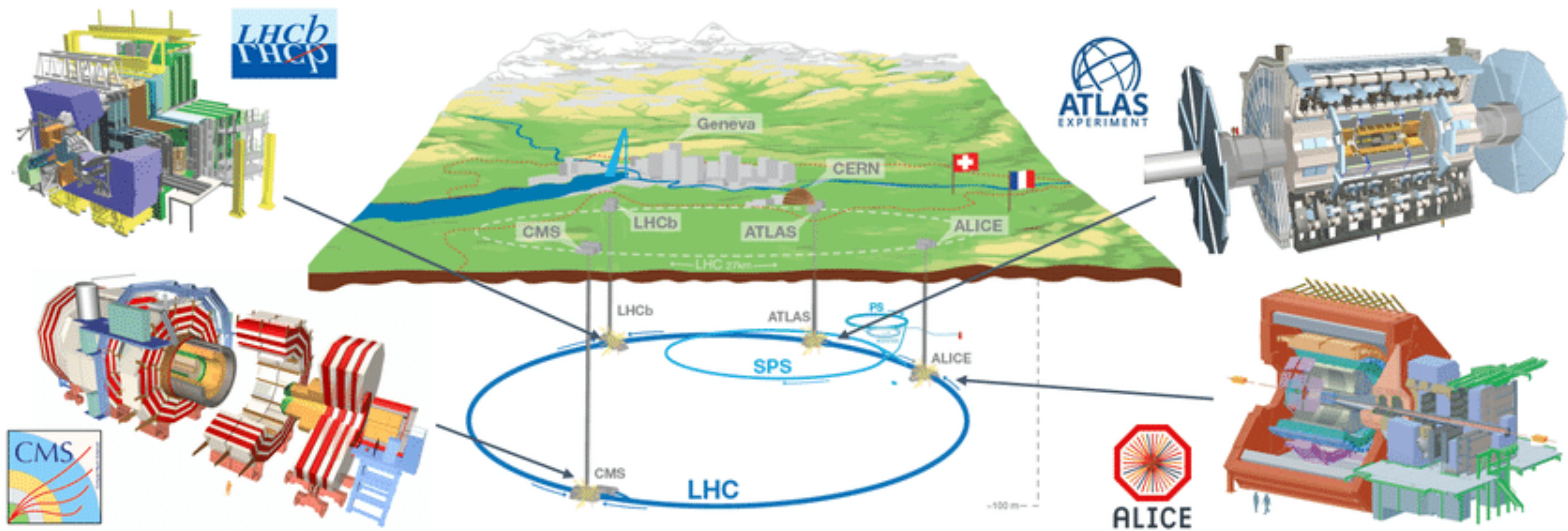
Workshop on Basic Computing Services in the Physics Department - subMIT

2024-02-02 @ MIT

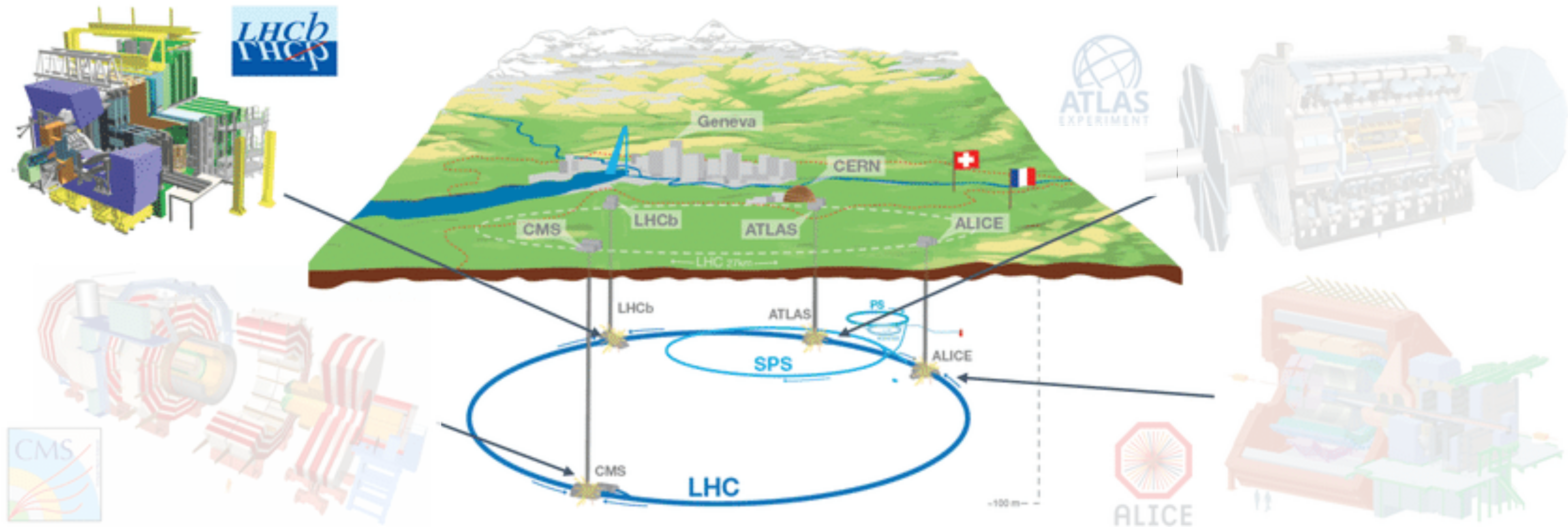
Blaise Delaney [[blaised at mit.edu](mailto:blaised@mit.edu)]

Laboratory for Nuclear Science & *IAFI*

The Large Hadron Collider at CERN

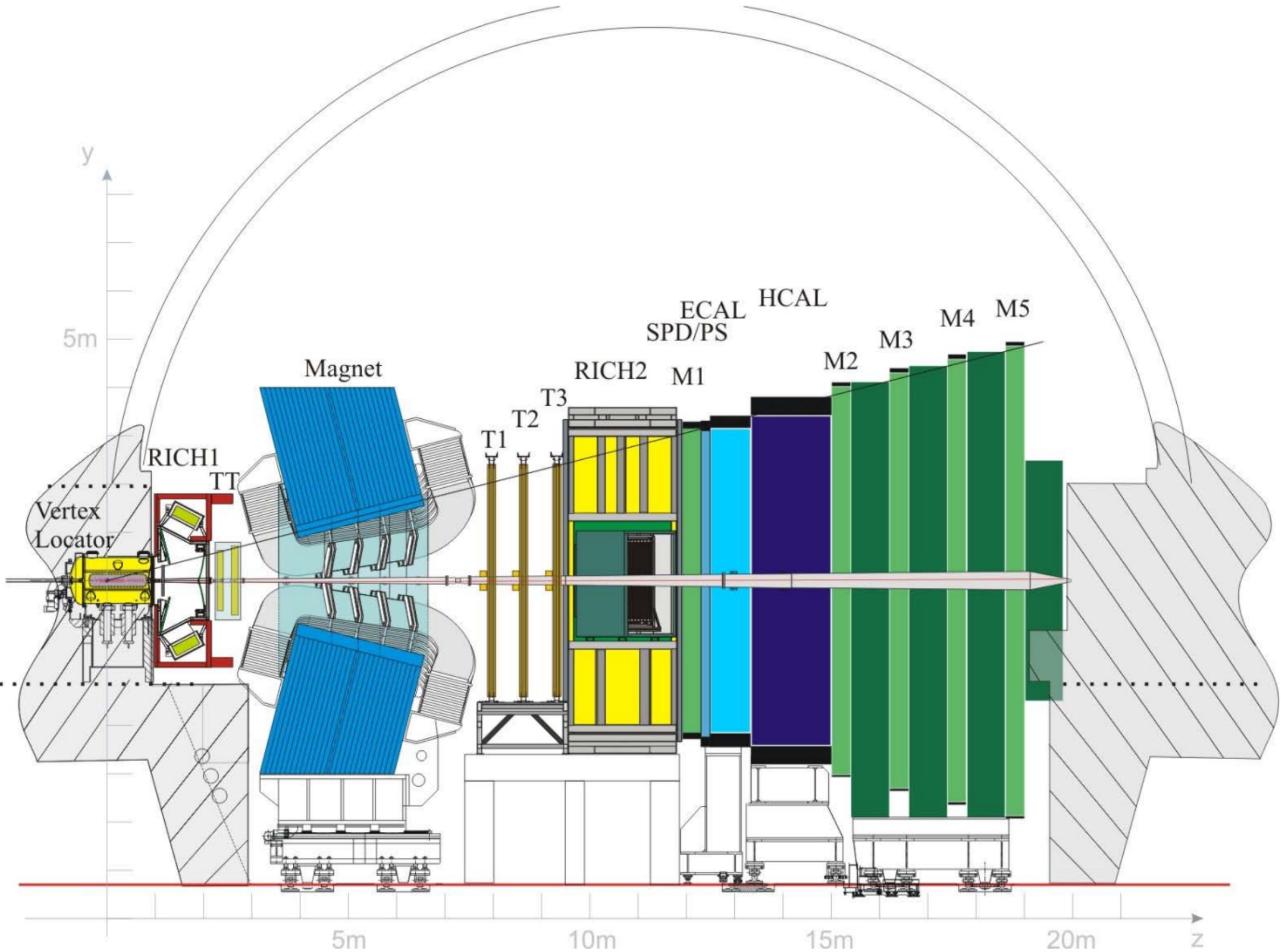


The Large Hadron Collider beauty experiment @ LHC

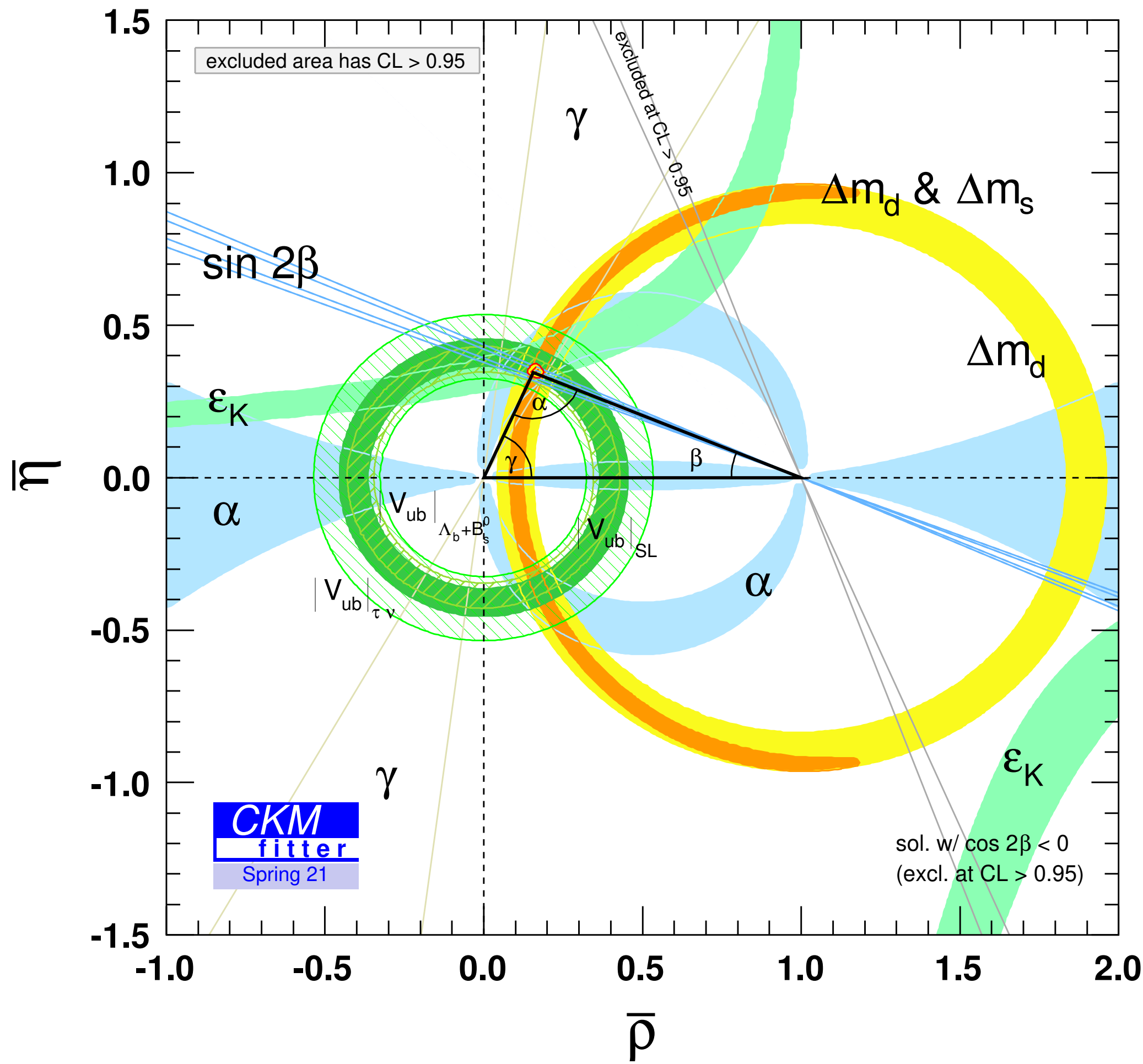


LHCb @ MIT

- Precision tests of the Standard Model (**CKM metrology**)



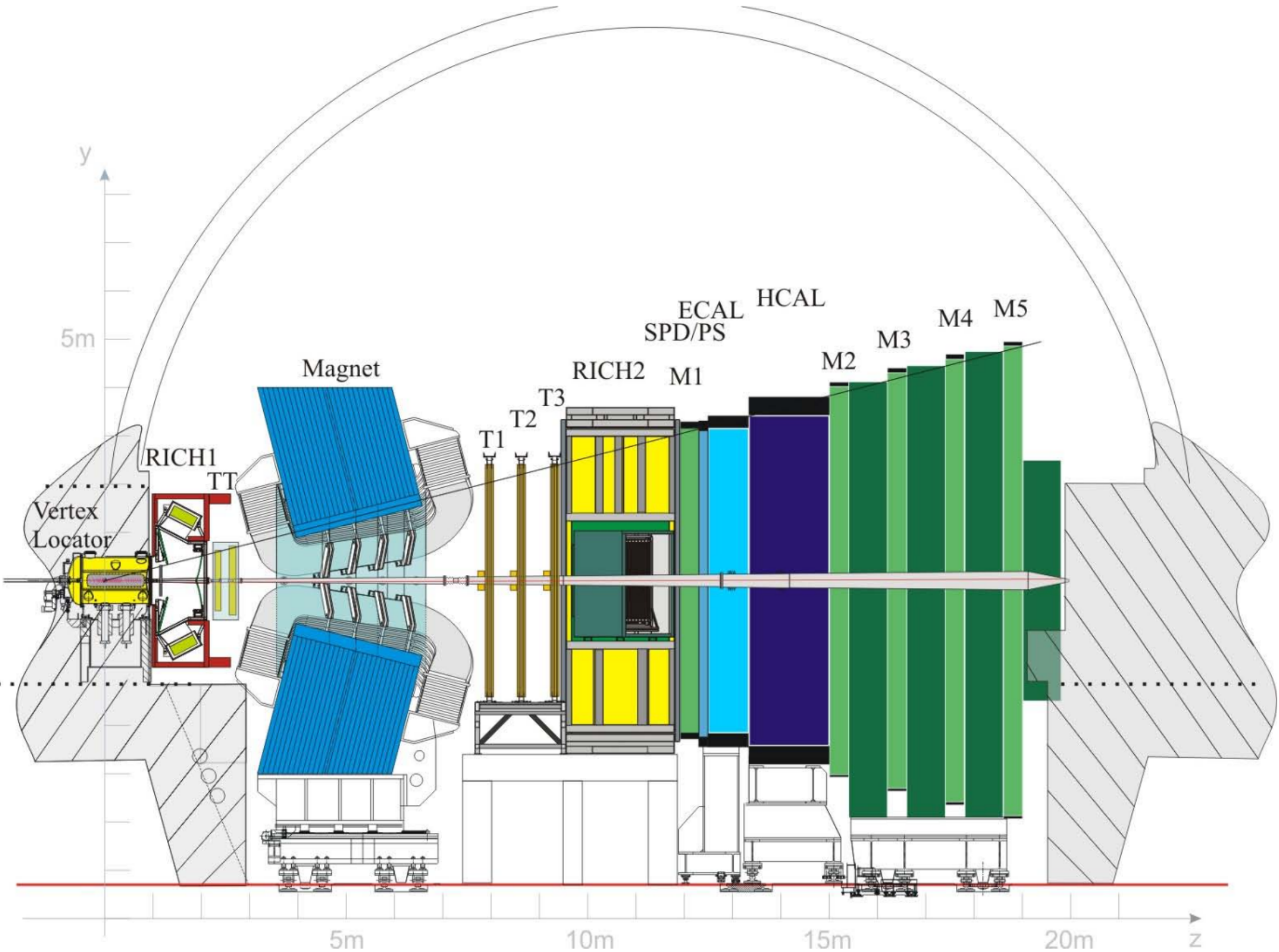
JINST 3 (2008) S08005



LHCb @ MIT

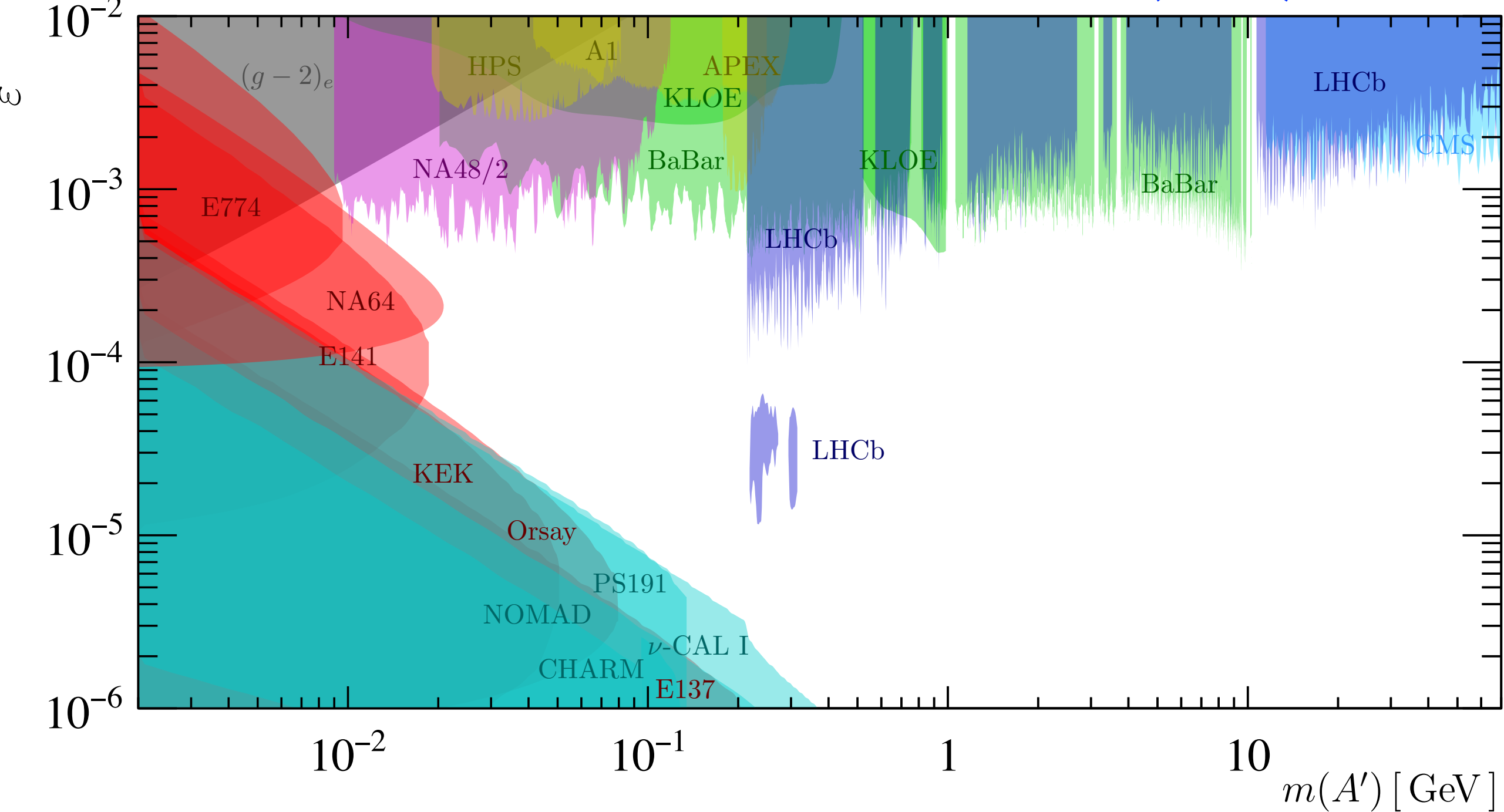
► Precision tests of the Standard Model (**CKM metrology**)

► **Dark-matter** searches: ALP and dark-scalar production in rare B decays, dark photons



JINST 3 (2008) S08005

PRL 124 (2020) 041801



LHCb @ MIT

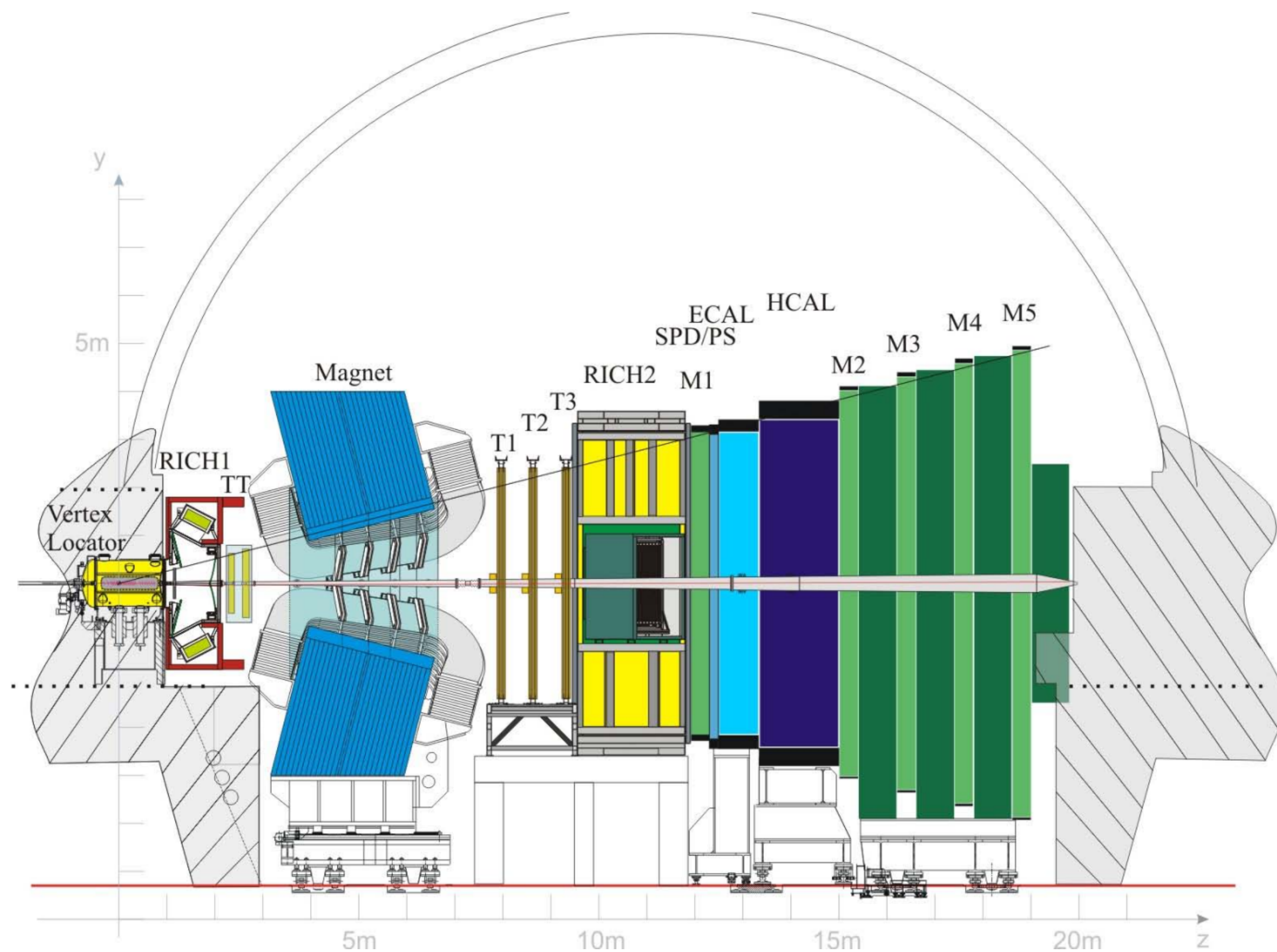
- ▶ Precision tests of the Standard Model (**CKM metrology**)

- ▶ **Dark-matter** searches: ALP and dark-scalar production in rare B decays, dark photons

- ▶ Applications of robust and interpretable **machine learning** for the (fully software!) Run 3

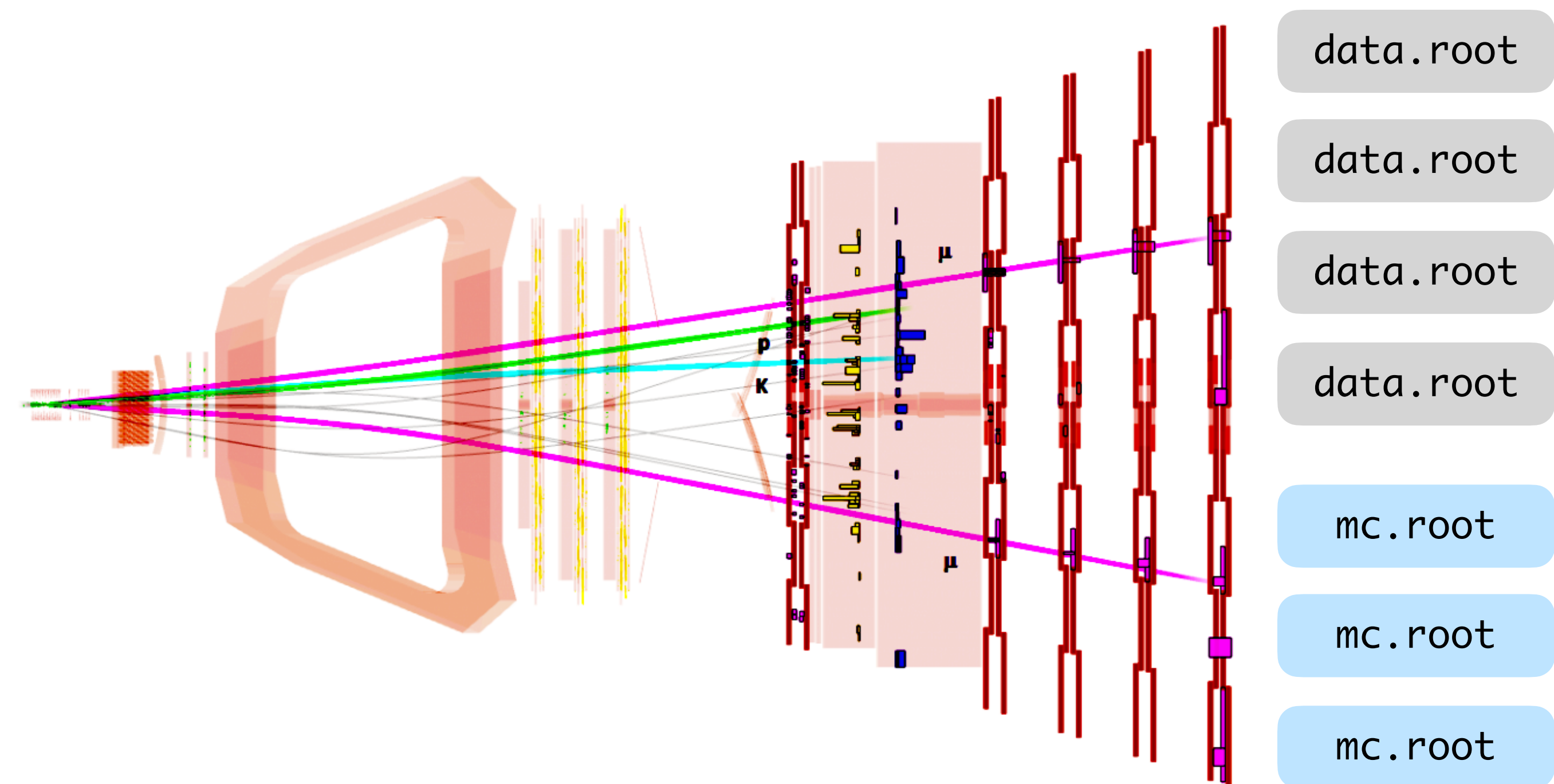
LHCb trigger system

[2306.09873](https://arxiv.org/abs/2306.09873), [2312.14265](https://arxiv.org/abs/2312.14265)

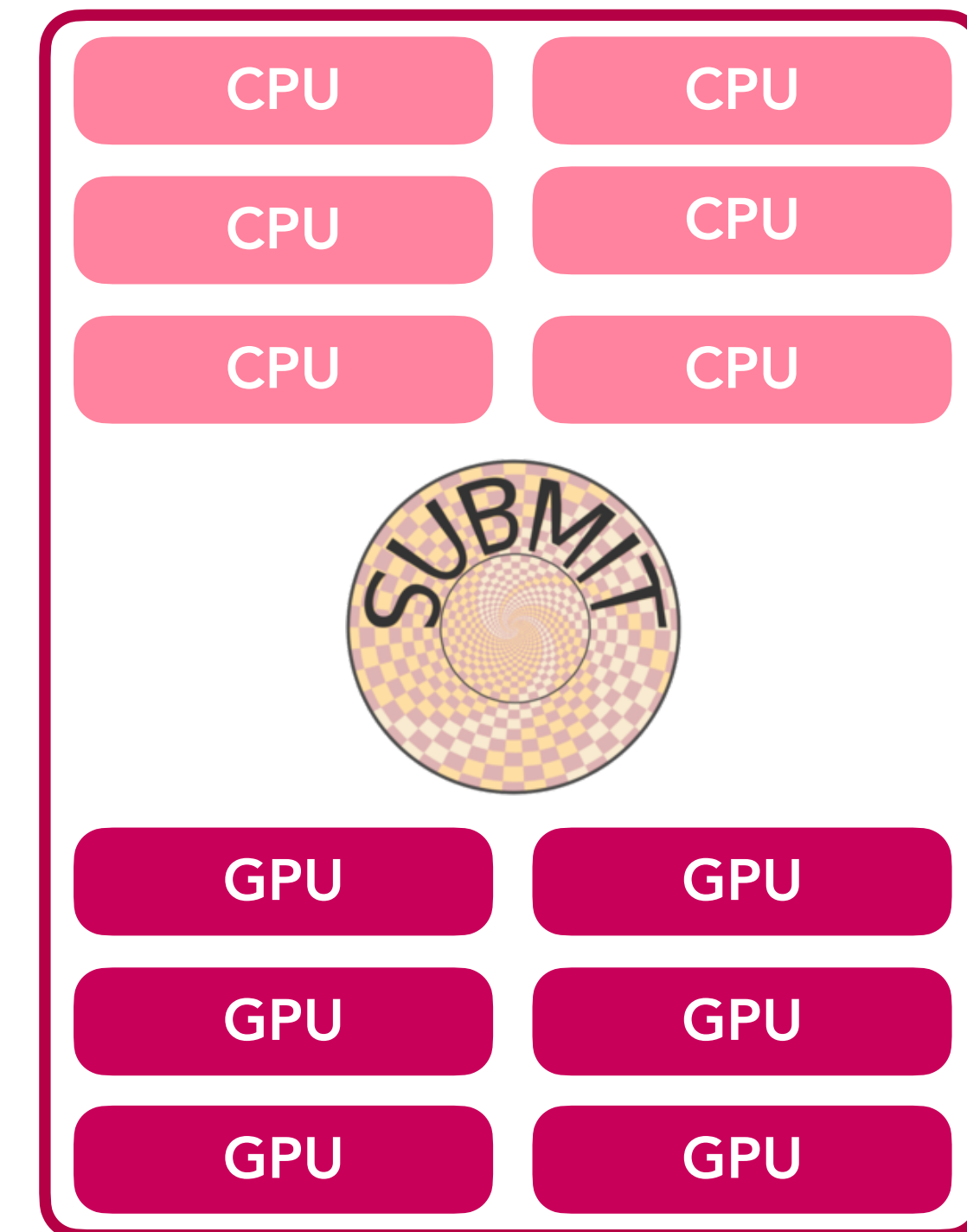
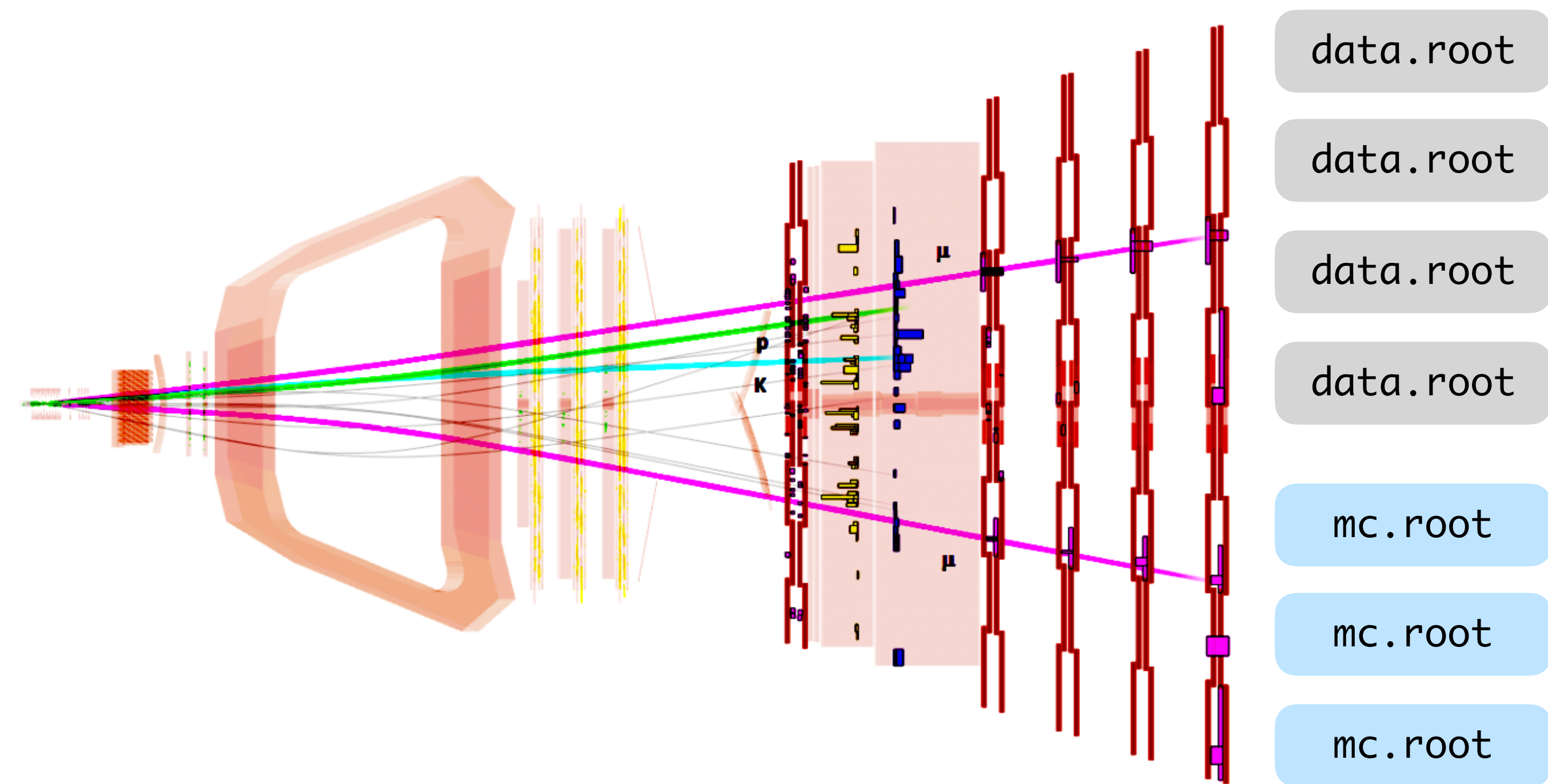


JINST 3 (2008) S08005

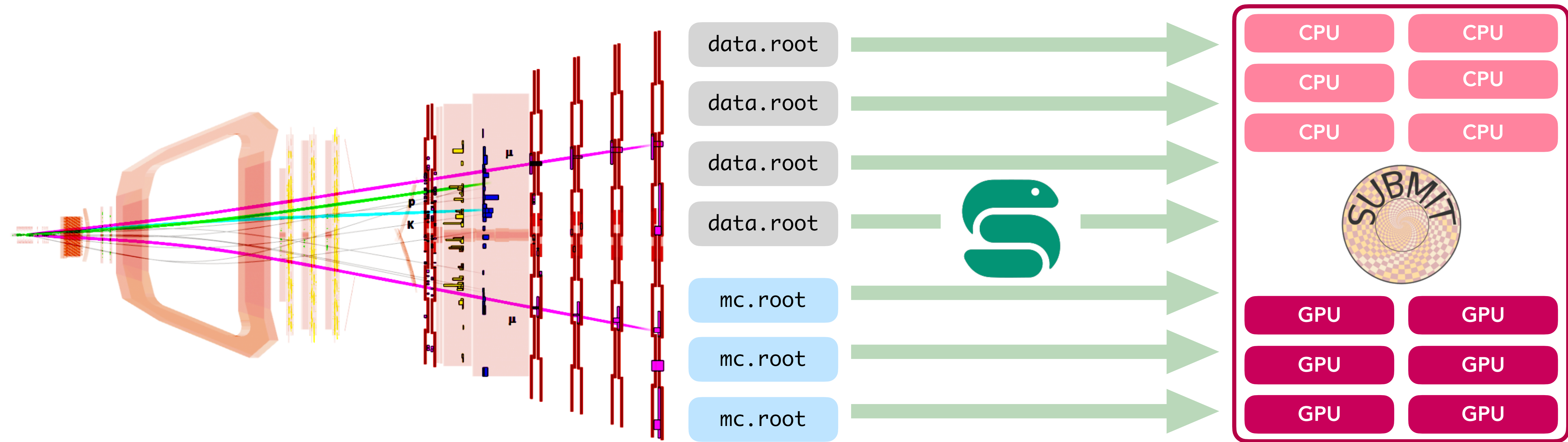
My workflow for measurements with LHCb + subMIT



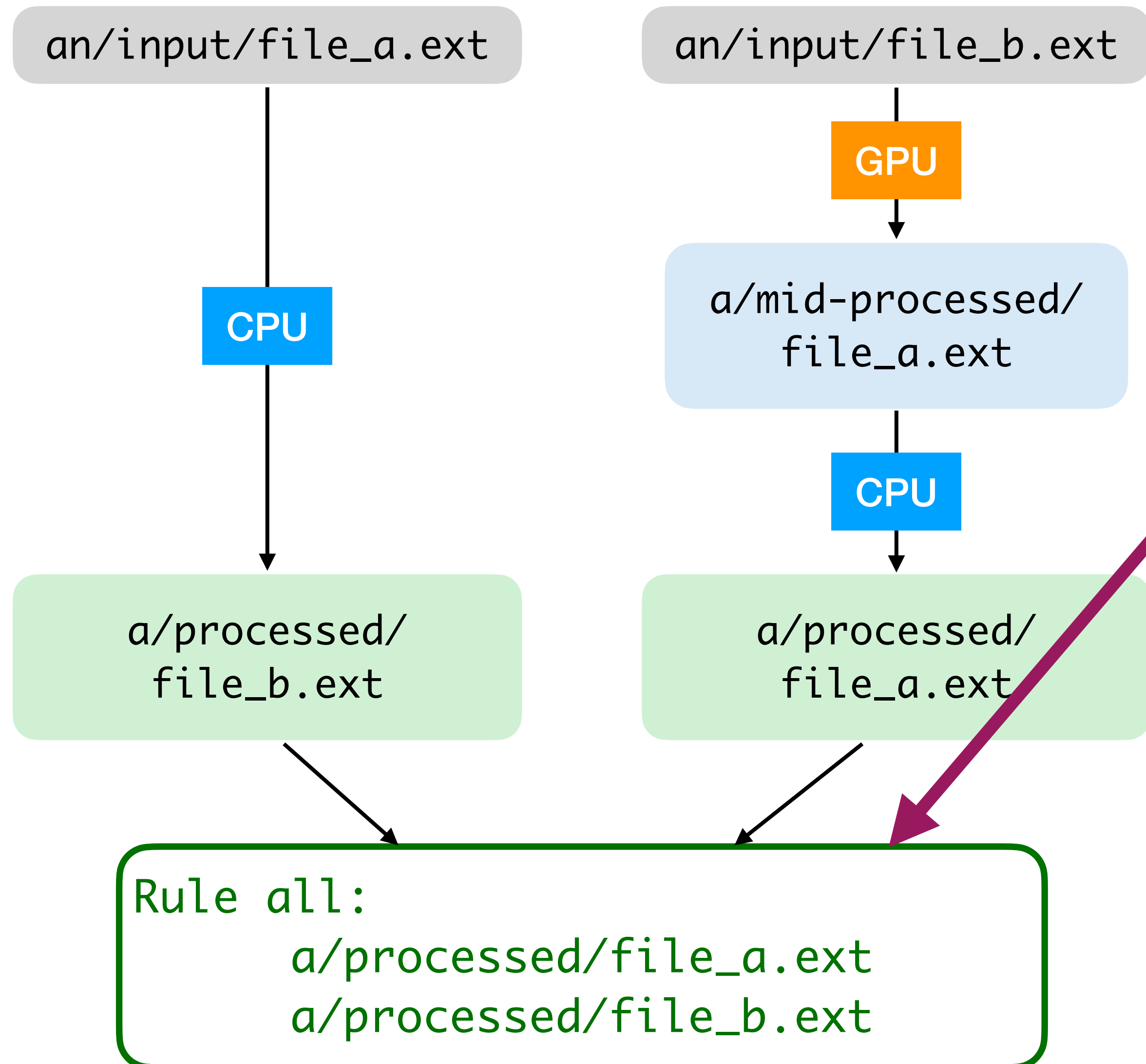
My workflow for measurements with LHCb + subMIT



My workflow for measurements with LHCb + subMIT



The Snakemake paradigm



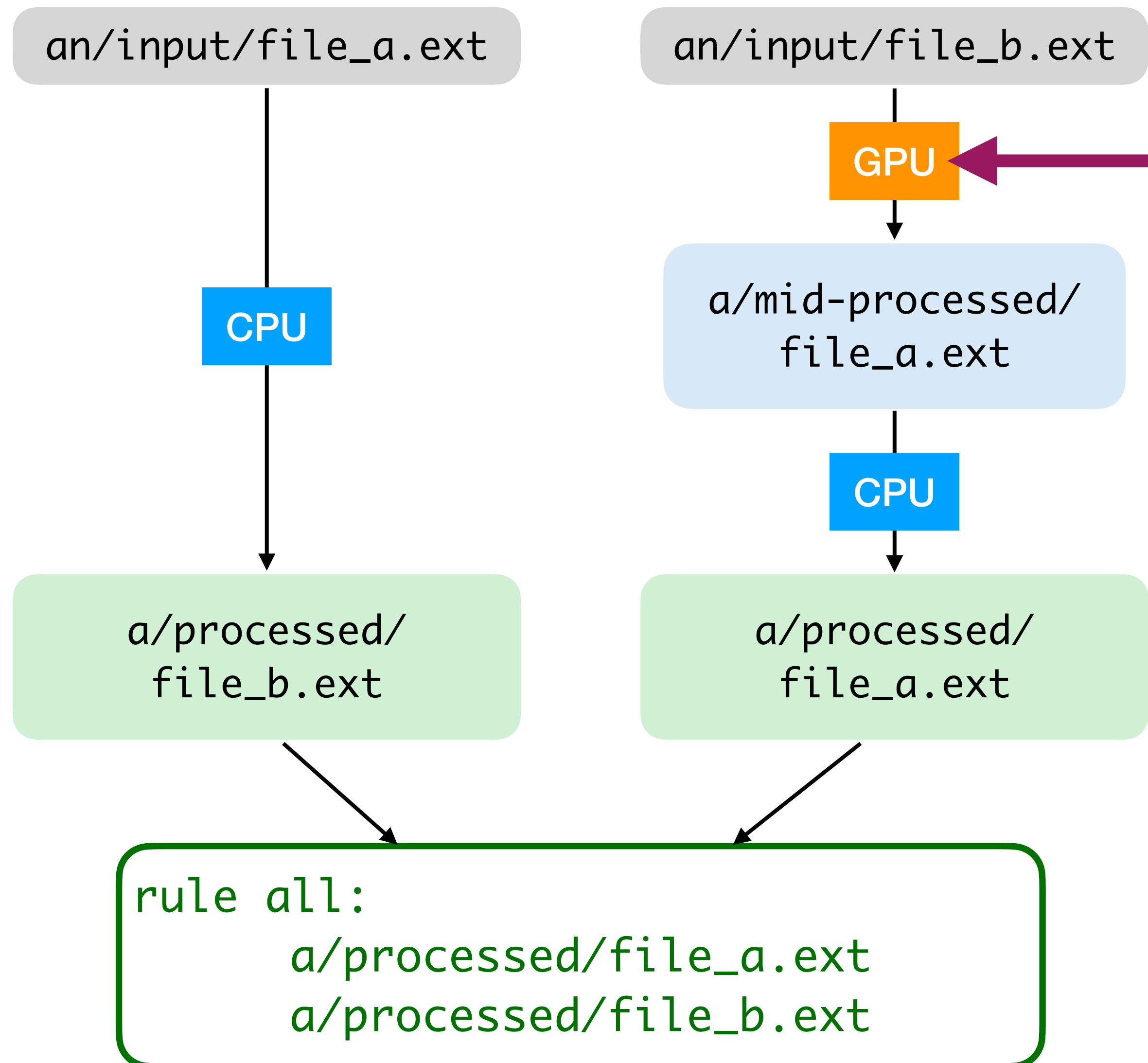
Target paths set the dependency chain, in the form of a direct acyclic graph (DAG).

"Here are the results I want; figure out how to get there."

Rule all:

a/processed/file_a.ext
a/processed/file_b.ext

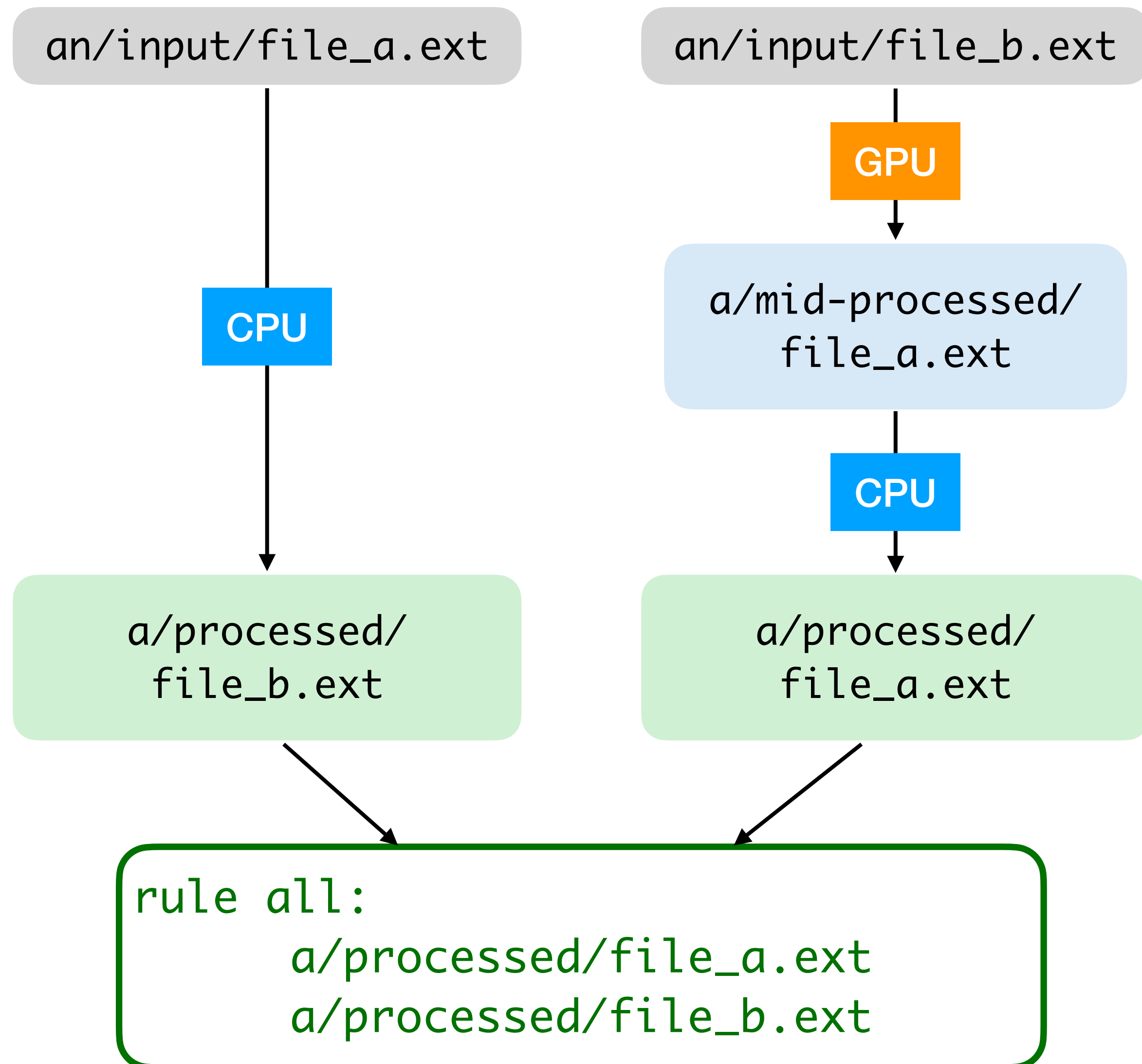
The Snakemake paradigm



Each analysis stage is a **node** in the DAG:

```
rule process_file_a_with_gpu:
  input:
    "an/input/file_a.ext"
  output:
    "a/mid-processed/file_a.ext"
  resources:
    nvidia_gpu=1
  shell:
    """
    your_gpu_processing_command
    {input} {output}
    """
```

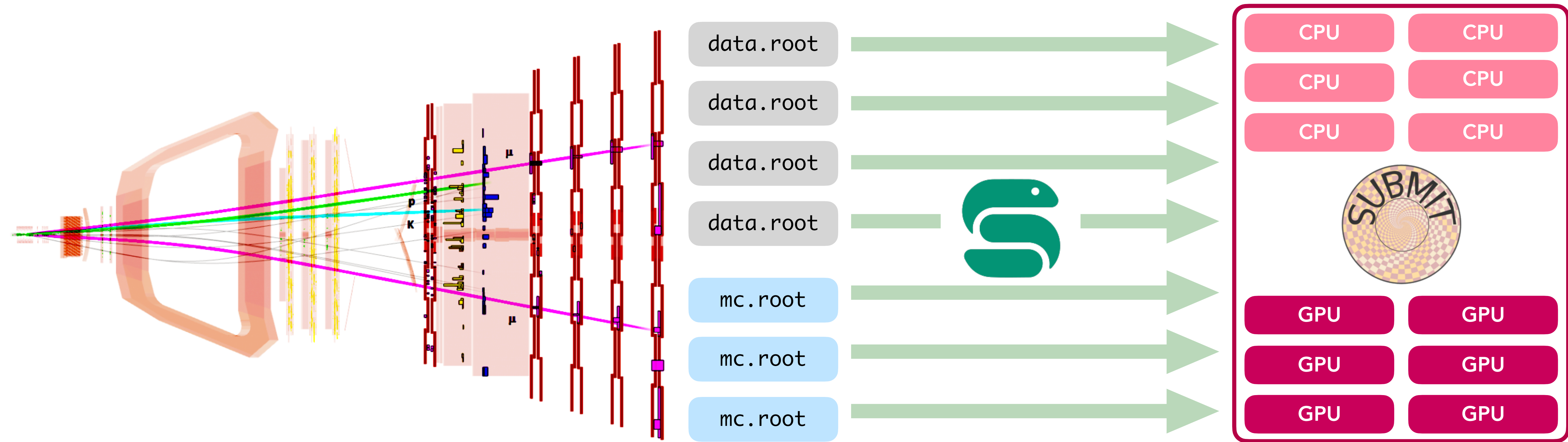
The Snakemake paradigm



The full workflow orchestrated in one-file, **Snakefile**:

- fully **Python**-based
- each analysis stage is defined by a rule
- the job dependency is entirely specified by the I/O path, and inferred by **string pattern matching**
- **no constraint** on *what* each rule **executes** (python, bash,...)

Tutorial



Git clone from <https://github.com/reallyblaised/snakemake-tutorial>

Skip the setup:

```
conda activate /work/submit/submit-software/conda/envs/snakem
```