Advancing KM3NeT Data Management

HARNESSING SNAKEMAKE AND GRID COMPUTING

Anna Sinopoulou on behalf of the KM3NeT Collaboration
INFN - Laboratori Nazionali del Sud

European Physical Society Conference on High Energy Physics 2025

KM3NeT

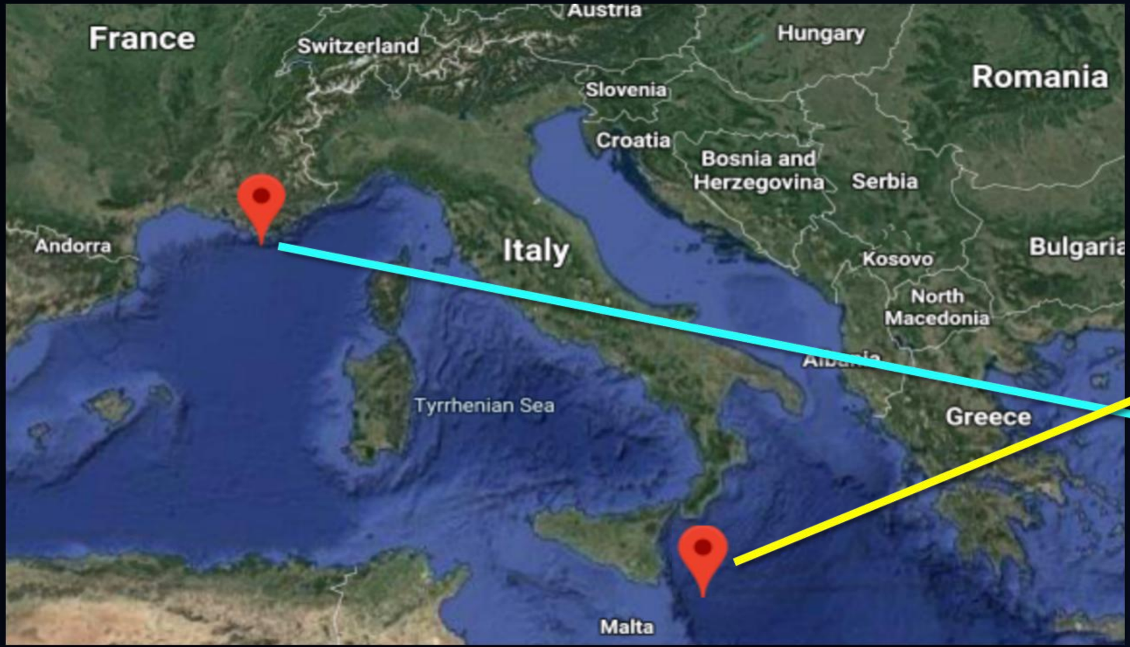
Supernovae | ν oscillations ν mass ordering | Dark Matter searches Exotics searches | Cosmic neutrinos Multimessenger Astronomy

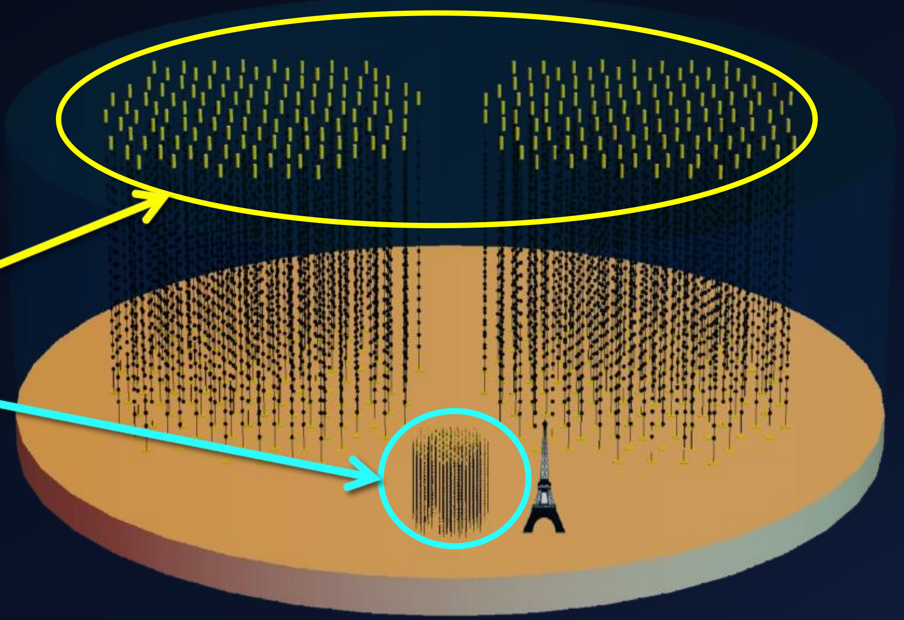
MeV → GeV → TeV → PeV

International collaboration building 2 underwater neutrino detectors in the Medditerannean Sea; the ORCA and ARCA detectors

Oscillation **R**esearch with **C**osmics in the **A**byss

Astroparticle **R**esearch with **C**osmics in the **A**byss

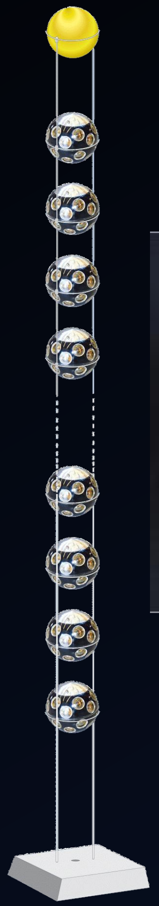




KM3NeT Technology and Data Rates

The basic detector elements:

- Optical sensors ► DOMs (Digital Optical Modules)
- Strings ► DU (Detection Unit)
- Seafloor network ► Electro-optical cables and Junction Boxes



Detection of Cherenkov light produced by relativistic particles

Photo-multiplier tubes (PMT) measure the arrival time of the light at the quantum level

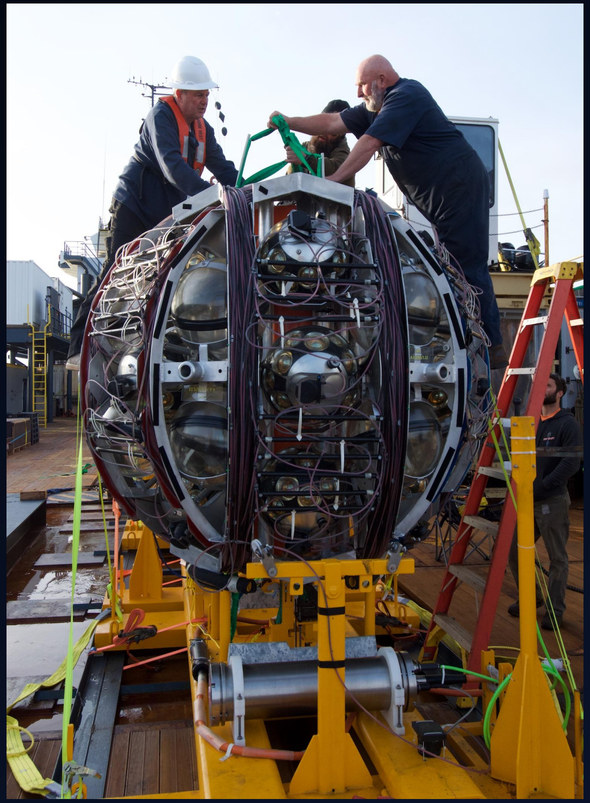
PMT singles rate 10 kHz

↓

1 string 300 Mbps

↓

Full infrastructure 100 Gbps

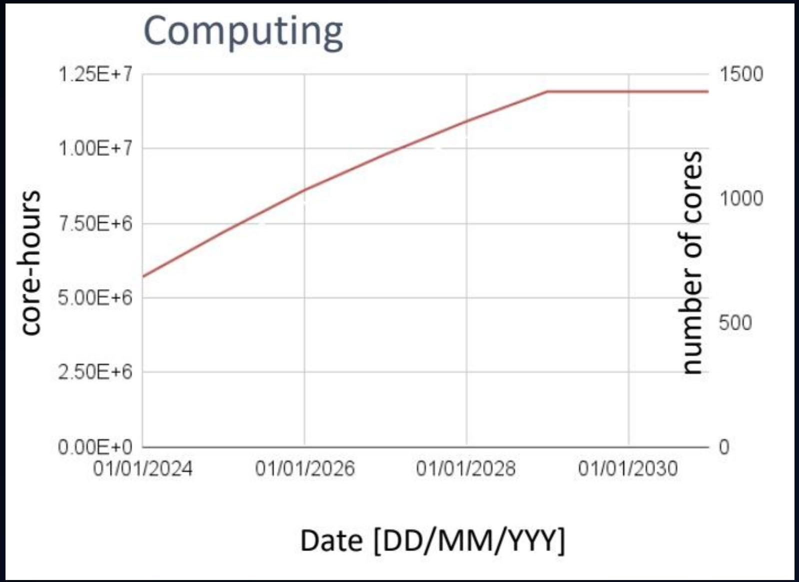


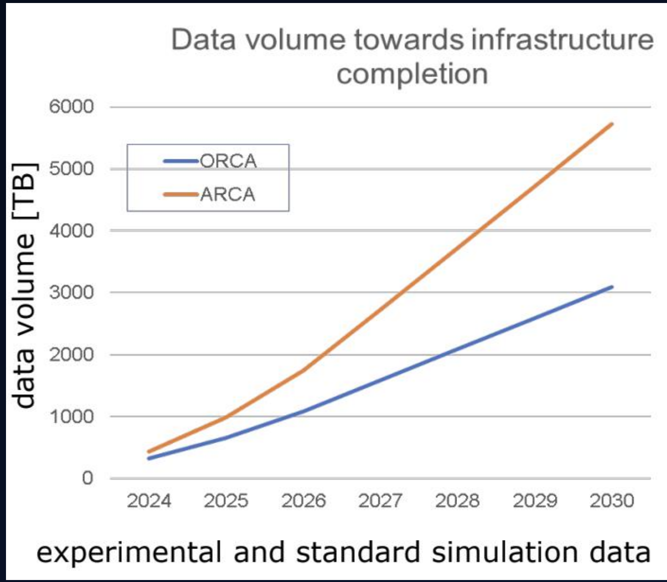
Computing Challenges & Motivation

1. Handling variations in detector alignment or deep-sea installation conditions require recalibration and tailored simulations, increasing complexity → **run-based data processing** (run-level consistency essential).

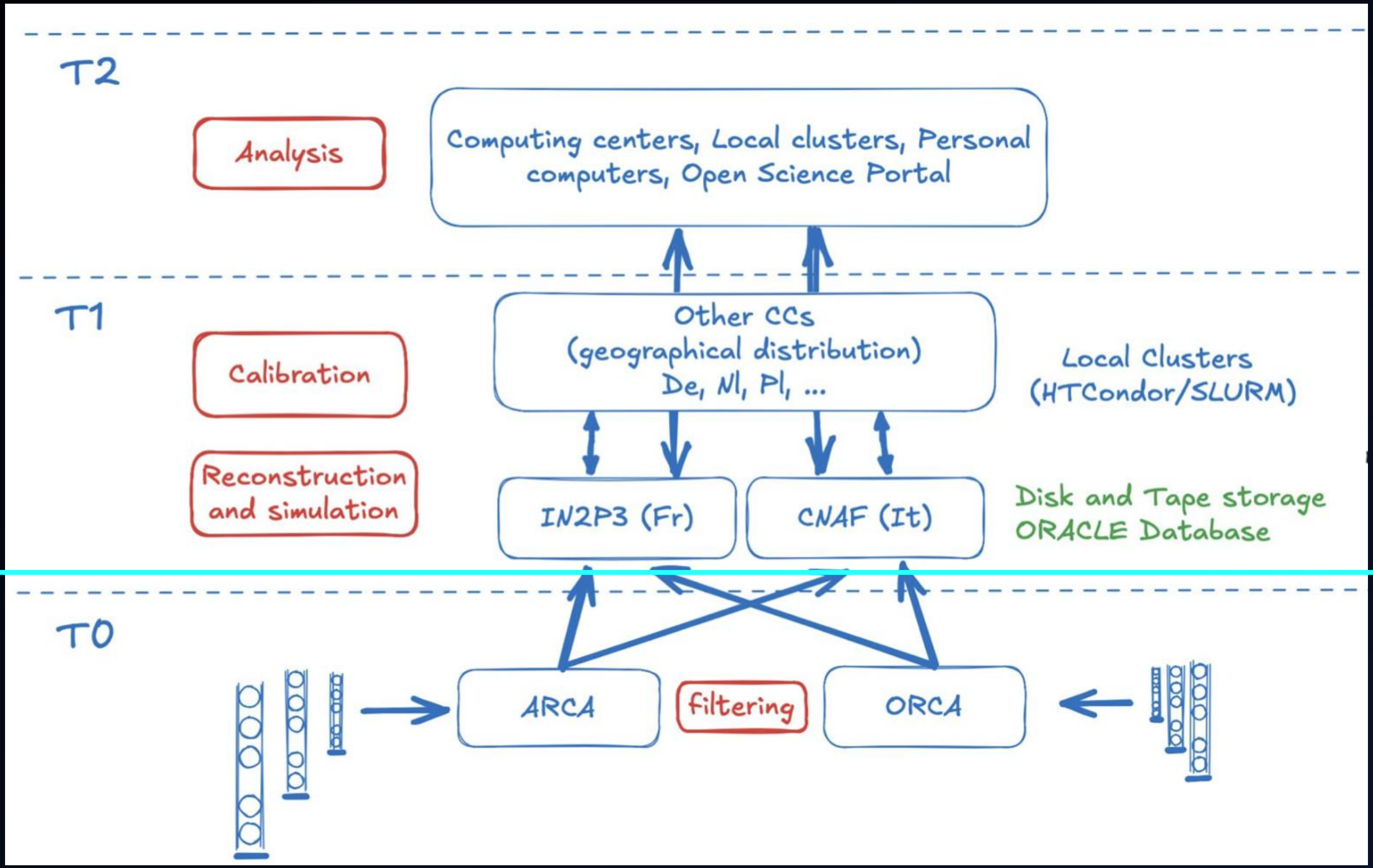
2. Surging detector scale → extremely large data volumes → need for **robust, scalable data processing and management**.

3. High-throughput simulation, calibration, reconstruction and analyses → **standardized workflows, cross-site reproducibility, and GRID integration.**





KM3NeT Computing Model - I

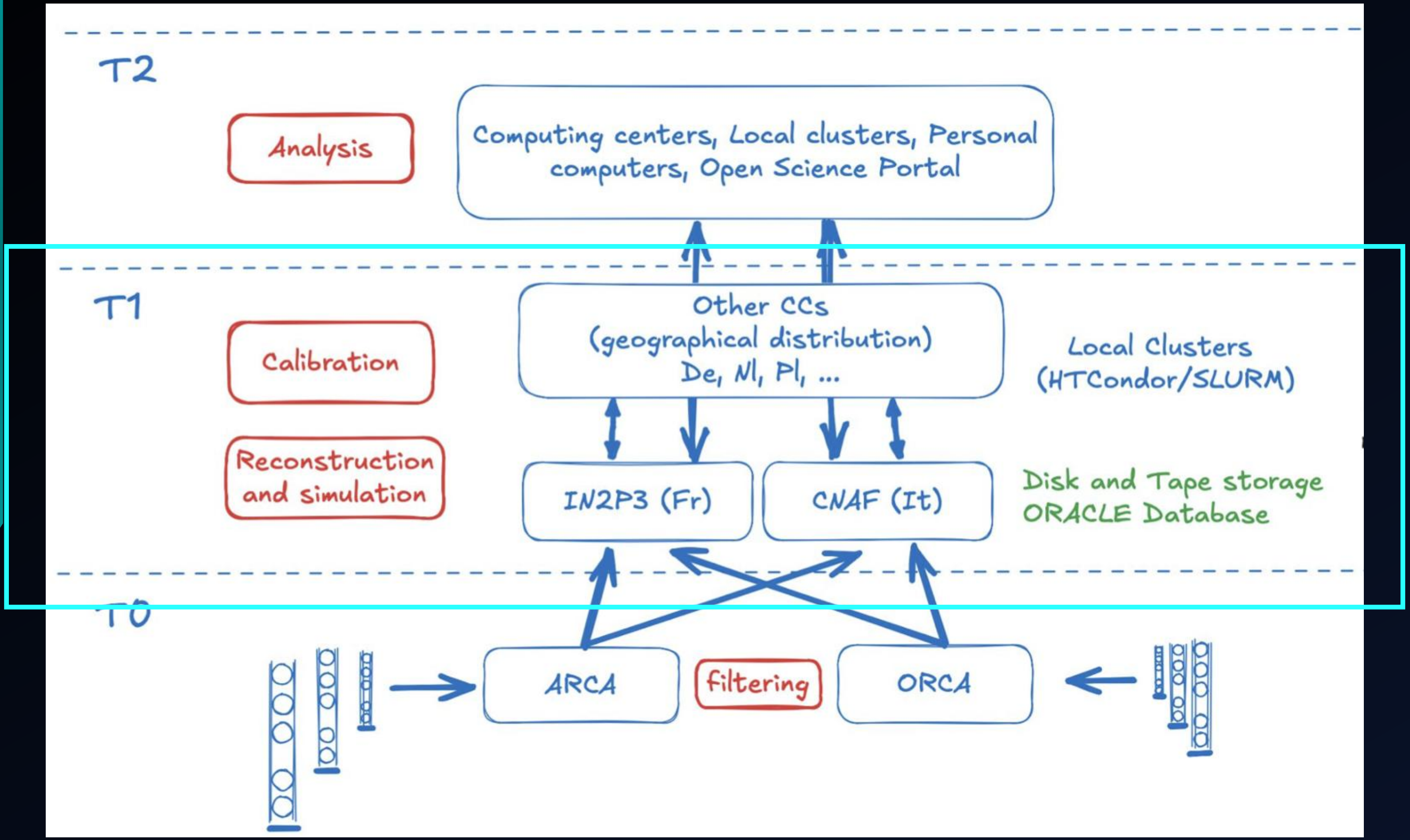


Tiered Architecture

T0: @Shore Station

- *"All-data-to-shore"* principle
- Real-time raw data filtering & event trigger
- Quasi-online calibration & reconstruction
- Reduces raw streams (~5 GB/s) to filtered permanent storage (~5 MB/s)

KM3NeT Computing Model - II

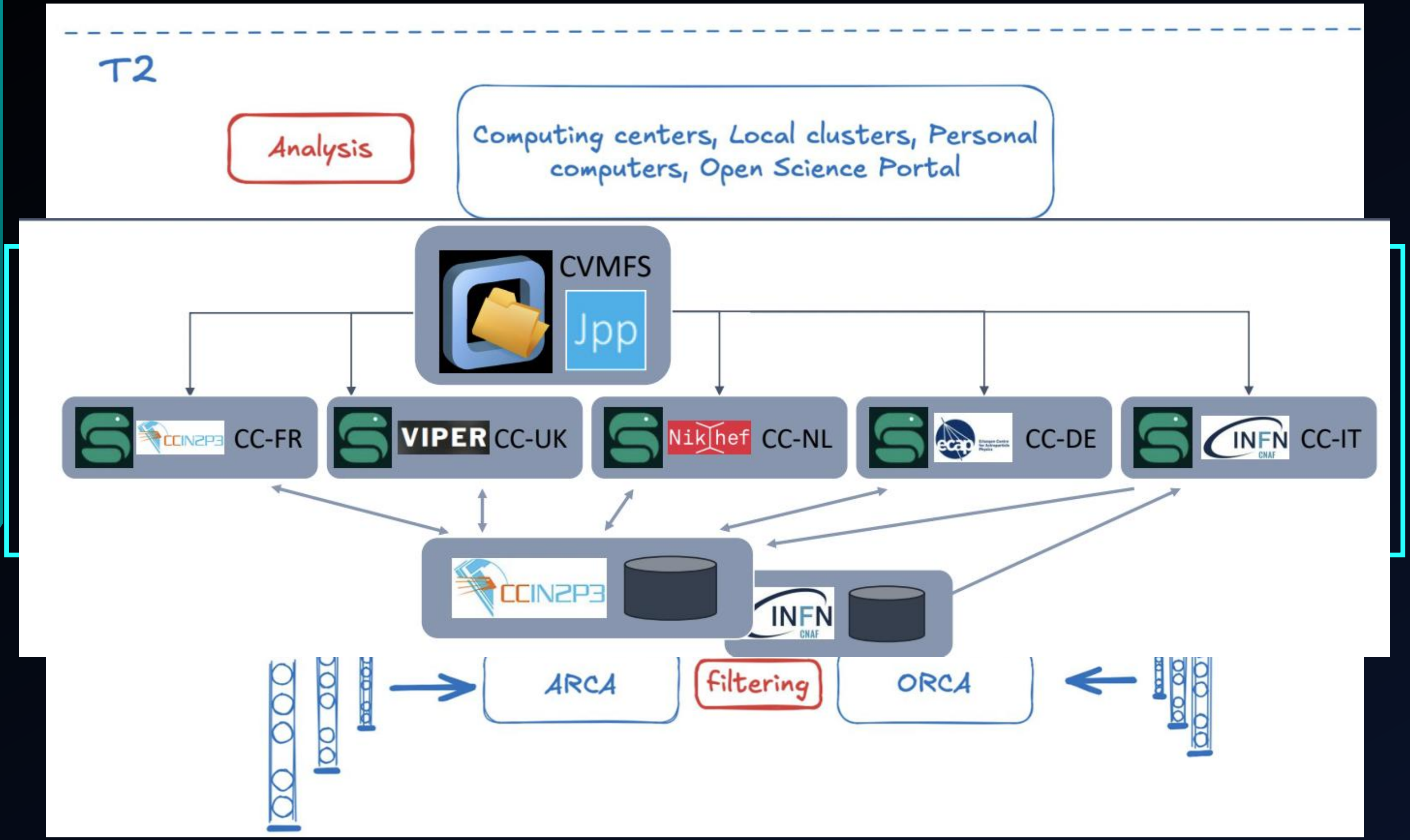


Tiered Architecture

T1: Offline Data Processing & Storage

- (Time and Position) Calibration of raw events
- Reconstruction of calibrated data events
- Simulation and reconstruction of events
  - atmospheric muons
  - atmospheric neutrinos
  - noise events

KM3NeT Computing Model - III



Tiered Architecture

T1: Offline Data Processing & Storage

- (Time and Position) Calibration of raw events
- Reconstruction of calibrated data events
- Simulation and reconstruction of events
  - atmospheric muons
  - atmospheric neutrinos
  - noise events

T2: Analysis

- Analysis, custom processing, MC studies, Open Science

KM3NeT Data Workflow Overview

Raw hit data acquisition & online filtering (DL0)

Hits from DOMs arrive at shore (Tier 0) and are filtered in real-time to identify potential neutrino events

Run-based calibration (DL1)

Per-run calibration for timing, sensor positions & environmental conditions

Run-based MC simulation

Monte Carlo events generated for each run in order to follow the corresponding data run conditions, executed alongside data reconstruction at Tier 1

Reconstruction (DL2)

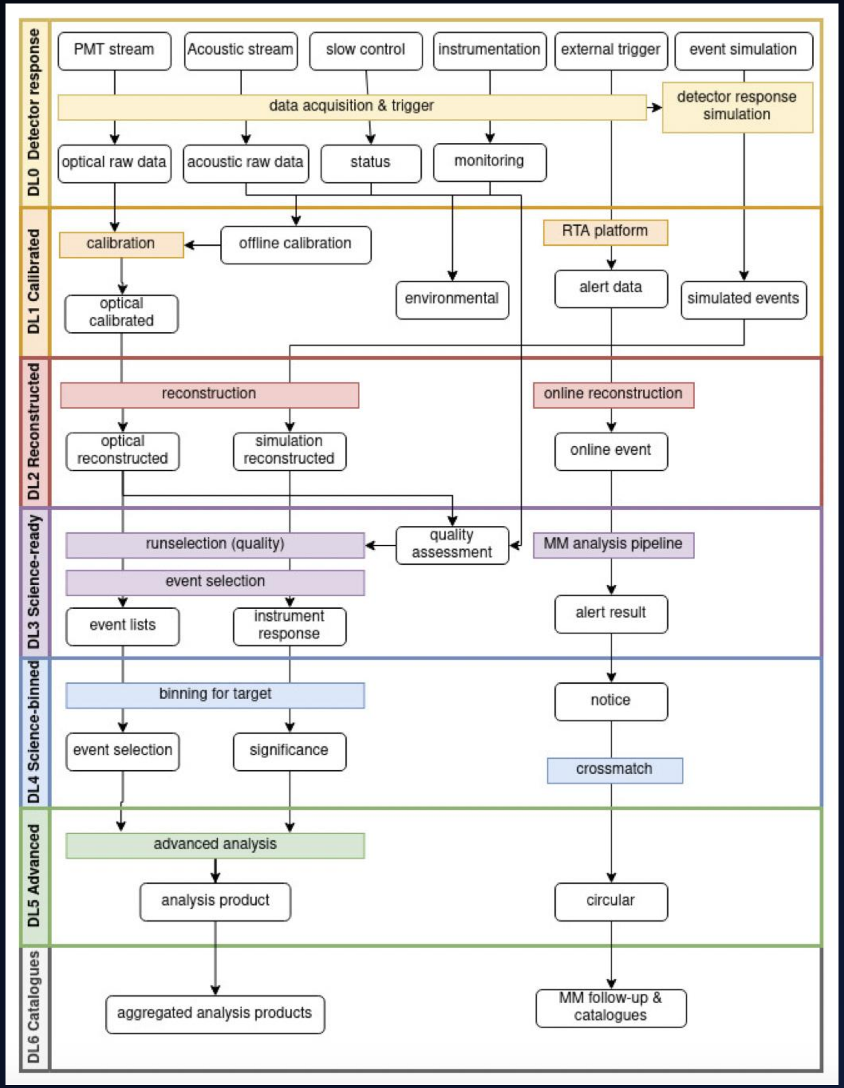
Calibration + MC inputs are used to reconstruct physics events (tracks, cascades) with C++/PyROOT tools

DST production (DL3)

Reconstructed events are filtered into reduced Data Summary Tiers for further analysis.

Aggregation & user-level data (DL4-DL6)

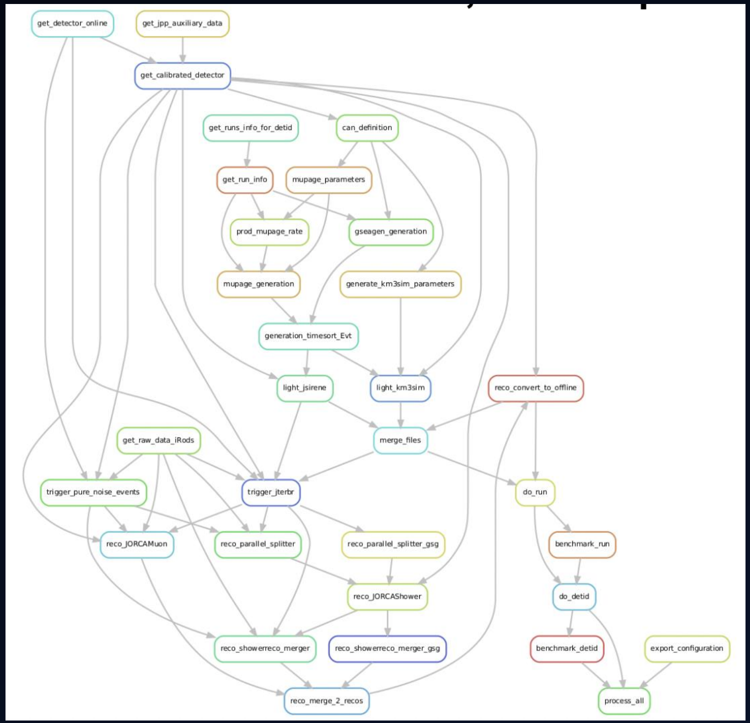
DSTs from multiple runs are aggregated to form science-ready datasets for user analysis



Snakemake: the Backbone of KM3NeT Data Pipelines

Why Snakemake?

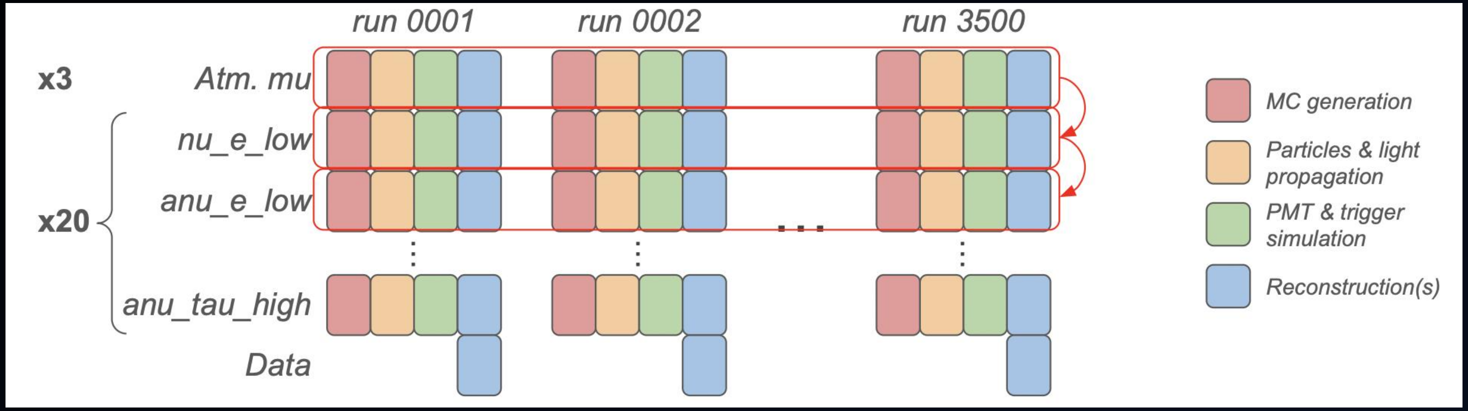
- **Portable**: Snakemake workflows can run seamlessly on local laptops, HPC clusters, and Grid systems.
- **Scalable**: Snakemake builds a DAG of jobs, automatically identifying and running tasks that can run in parallel.
- **Reproducible**: Rules specify inputs, outputs, and exact commands, eliminating hidden scripts and manual steps.

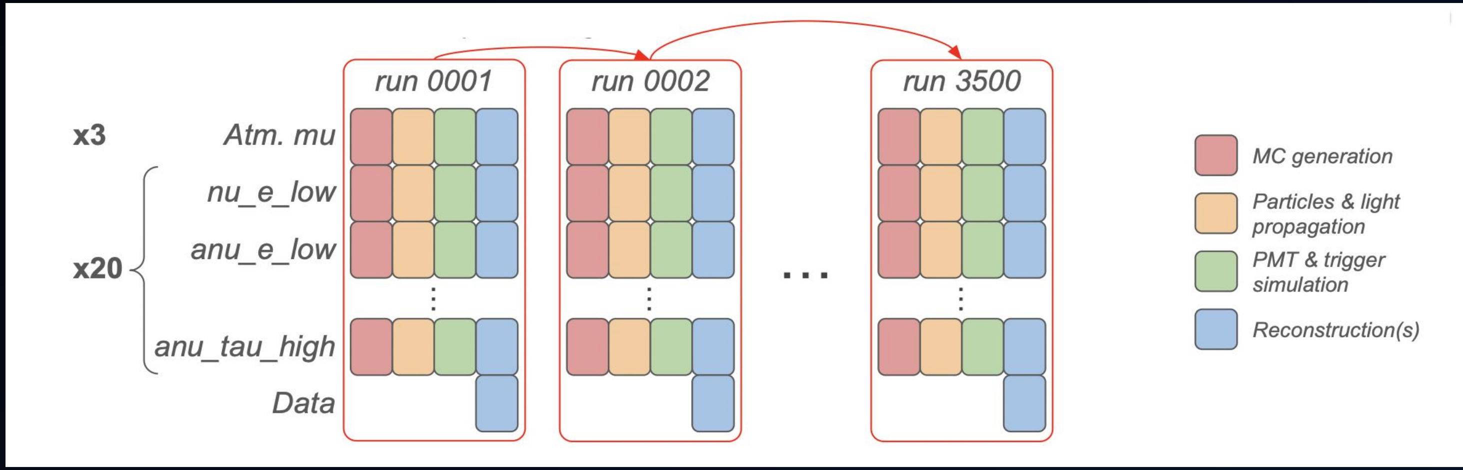



```
rule get_calibration:
    output:
        "{run}/calibration_{run}.txt"
    shell:
        """
        echo "{wildcards.run}" > {output}
        """

rule get_raw_data:
    output:
        "{run}/raw_{run}.data"
    shell:
        """
        echo "{wildcards.run}" > {output}
        """

rule analysis:
    input:
        calib = "{run}/calibration_{run}.txt",
        data = "{run}/raw_{run}.data"
    output:
        "{run}/analysis_{run}.hist"
    shell: "touch {output}"

rule do_plots:
    input:
        "{run}/analysis_{run}.hist"
    output:
        "{run}/analysis_{run}.plot"
    shell: "touch {output}"
```

- **Fine-Grained Resource Control:** Assigned CPU threads, memory, time limits per rule; Snakemake respects these when submitting jobs via SLURM, PBS, or Grid-directed systems.
- **Modularity and Code Reuse:** Rules, wrappers, and configurations can be shared and reused across projects—minimizing duplicate effort via containers.

Snakemake: an ORCA example





Old processing; per run contribution: 24 files
Snakemake processing; per run contribution: 3 files

Old run processing

Snakemake run-based processing

- Resources mutualization, allowing for early file merging (e.g. after light propagation for neutrinos).
- Optimize disk-space used during processing.
- No consecutive job completion.
- Containers usage.
- Benchmarking.

Entering GRID computing

From clusters	to GRID
One site	Mulitple sites
Shared local storage with home account	No shared storage
Username/password-based authentication	Certificate/Token- based authentication
Relatively homogeneous hardware	Heterogenous software
Direct job submission	Job submission through middleware

Meeting Demands of Scale & Performance

- KM3NeT needs **increased resources** for large-scale Monte Carlo simulations, reconstruction, and calibration.
- Workloads range from serial to multi-core and future GPU jobs, requiring scalable resource orchestration across sites:
    - **Serial Jobs;** Ideal for simple calibration or small dataset tasks (1 core).
    - **Multi-Core Jobs;** Reconstruction and simulation tasks often benefit from 8-core or higher processing – Snakemake handles this with threads.

- **GPU-Optimized Workloads** (Future); Through Snakemake's grouping support, multiple GPU tasks can be launched together on a single GPU
- **High-Throughput Workflows**; Large-scale MC campaigns and processing runs handled via Grid distribution and DIRAC orchestration.

GRID computing integrated into KM3NeT

Interacting with different schedulers across sites requires different protocols, user knowledge of environments and machine power

- **Integration with EGI Grid via DIRAC**

DIRAC handles job submission (to the EGI Grid infrastructure for production workflows), monitoring, and resource access across computing sites.

Rucio for Distributed Data Management

Snakemake workflows now leverage a prototype Rucio instance for input/output handling and dataset replication across Tier-1/2

Containerized Execution via CVMFS (CernVM File System) & Apptainer containers

Standard software environments are distributed via CVMFS and deployed with Apptainer containers, ensuring workflow portability and consistency

Conclusions & Future Outlook

KM3NeT's Data Framework Is Robust & Scalable
Tiered architecture combined with Snakemake and Grid integration ensures scalable, reliable data processing.

Snakemake Enables Portability & Reproducibility
fostering reproducible and environment-agnostic pipelines.

Grid & DIRAC Amplify Compute Throughput reducing bottlenecks and increasing capacity.

Rucio & Containers Streamline Data & Software Management strengthen consistency and accessibility.


Article
Observation of an ultra-high-energy cosmic neutrino with KM3NeT

https://doi.org/10.1038/s41586-024-08543-1
Received: 19 August 2024
Accepted: 18 December 2024
The KM3NeT Collaboration

Nature 638 (2025) 8050, 376

This foundation positions the experiment for MORE timely scientific discoveries as the detectors continue to expand!

Extras

Snakemake run processing in GRID