Data-handling and Infrastructure

Training course: Machine learning and Destination Earth

Julien Lefaucheur, on behalf of many

ECMWF

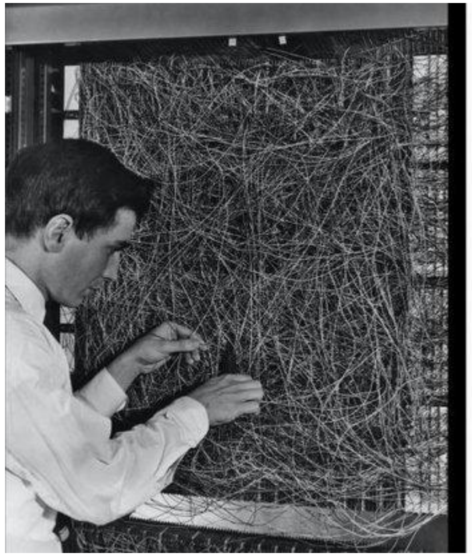
julien.lefaucheur@ecmwf.int

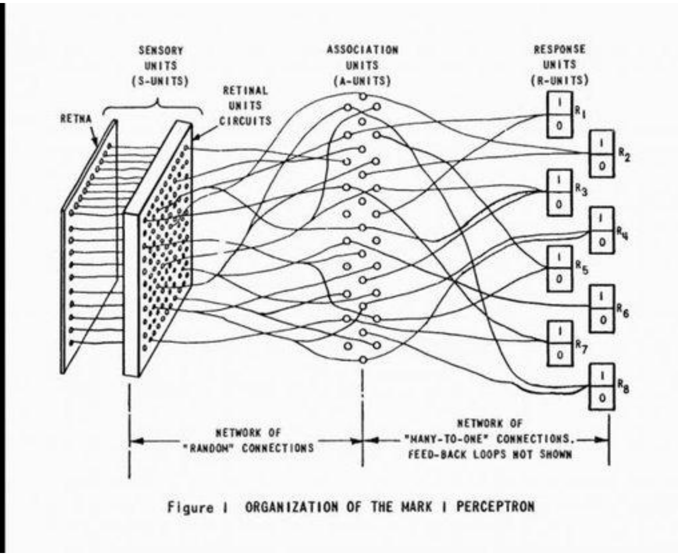
Outline

1. **Data Handling (C. O'Brien, F. Pinault, E. Pinnington, N. Zelenka)**
   - Introduction
   - Data Flow
   - Access Patterns & Datasets
2. **DestinE & Infrastructure (DestinE folks @ECMWF, N. Zelenka)**
   - EuroHPC
   - ML applications
     - *Datasets transfer across EuroHPC*
     - *Orchestration of training workflows prototype across EuroHPC*

Introduction – Background

- Machine learning is an old idea

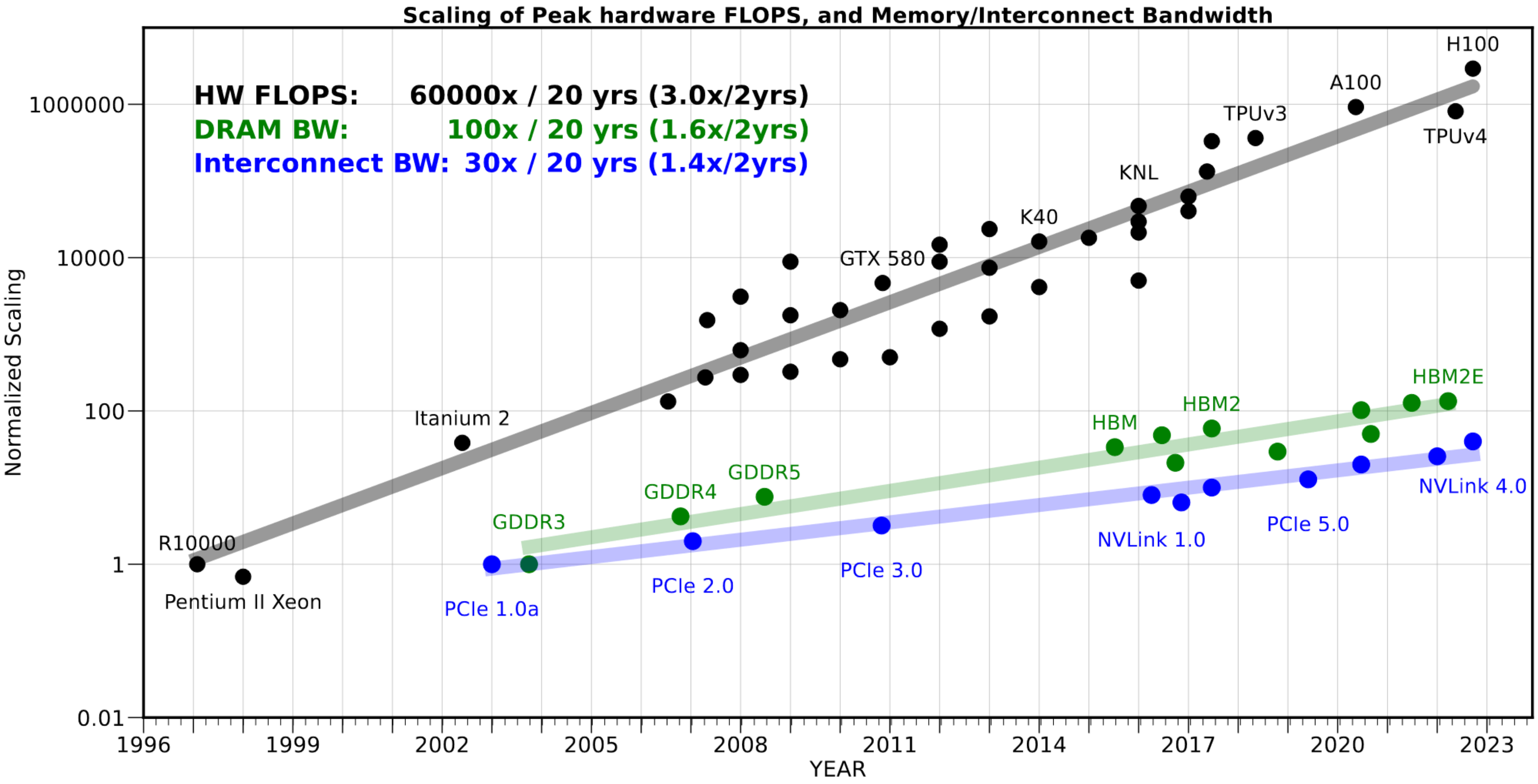




Frank Rosenblatt programming his Perceptron, 1970s

- Increased compute power (and available data) makes machine learning possible today
  - Moore's law
  - Domain specific compute architecture, GPUs
  - Infrastructure (data center, High Performance Computer)
- But there is a problem… as compute got faster, other components couldn't keep up

Introduction – The Memory Wall



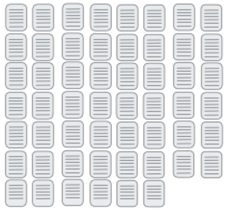
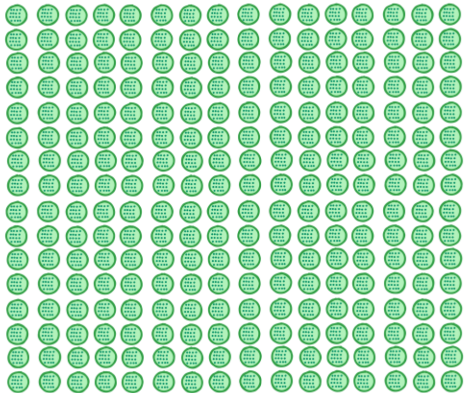
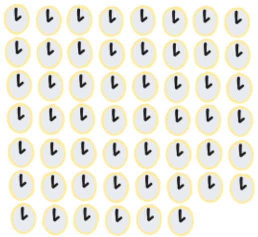
https://siliconmatter.substack.com/p/the-memory-wall-and-its-implications

Introduction – Training examples

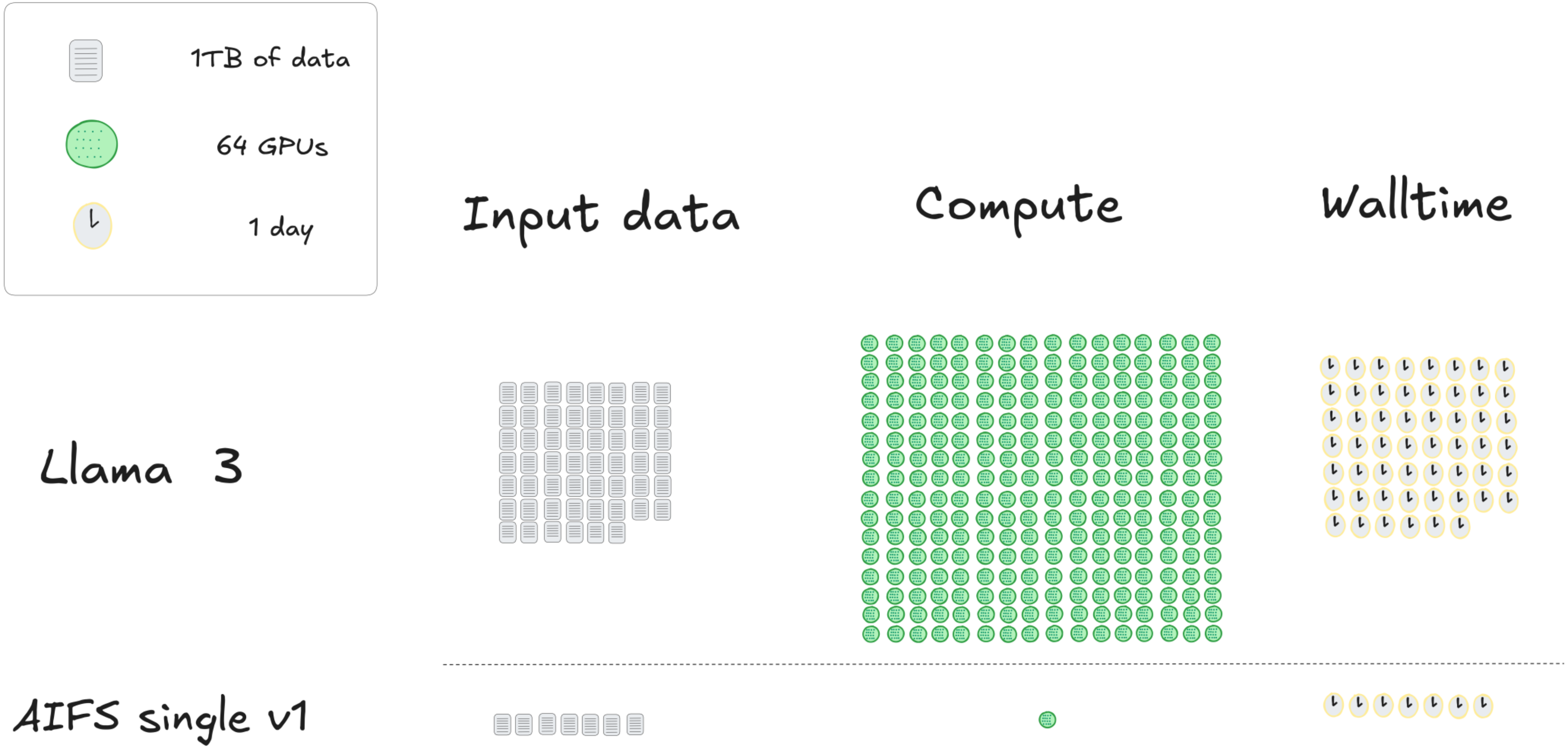
1TB of data

64 GPUs

1 day

	Input data	Compute	Walltime
Llama 3			

Introduction – Training examples



Outline

1. **Data Handling**
   - Introduction
   - Data Flow
   - Access Patterns & Datasets
2. **DestinE & Infrastructure**
   - EuroHPC
   - ML applications
     - *Datasets transfer across EuroHPC*
     - *Orchestration of training workflows prototype across EuroHPC*

How to handle data?

Questions

How do I transfer my data to … ?

Which data should I copy? Where?

Which format should I choose for my data?

Should I use more memory/machines/nodes/GPUs?

How to make “it” fast?

How do I parallelise part … of my workflow?

How to handle data?

Questions

How do I transfer my data to … ?

Which data should I copy? Where?

Which format should I choose for my data?

Should I use more memory/machines/nodes/GPUs?

How to make “it” fast?

How do I parallelise part … of my workflow?

Know the technology

HPC (High Performance Computing)? Cloud?

S3 buckets vs Lustre filesystems?

“Bring the code close to the data instead of data to the code.”

“Cloud-friendly” format?

Know your dataset

Total size on disk?

Total uncompressed size?

How many files?

Any missing data? Nans ?

Dimensions of the data? Full n-dimensional array? Several arrays?

For machine learning :

- What is the size of one training sample? Of one batch?
- Where is the data from? On which data will I run inference? How?

Know your read/write patterns

Random read (shuffling)

Transpose the data if needed

Data Flow: How to handle data?

Universal answer

➔“It depends”

The best solutions will usually depend

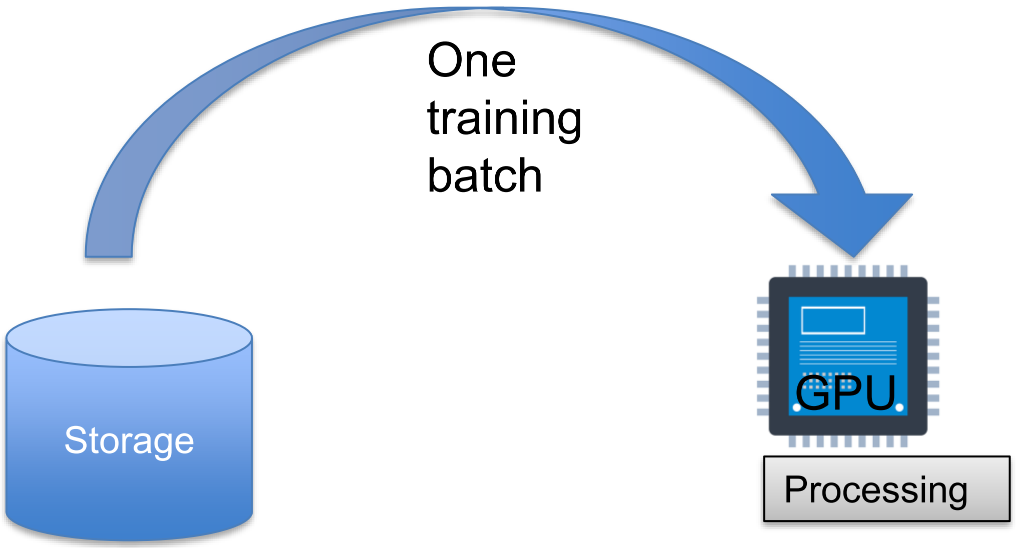
- on the size of the data
- on the project requirements
- on the available funding
- on previous experience
- on personal preferences
- and more…

“When you have a hammer, everything looks like a nail.”

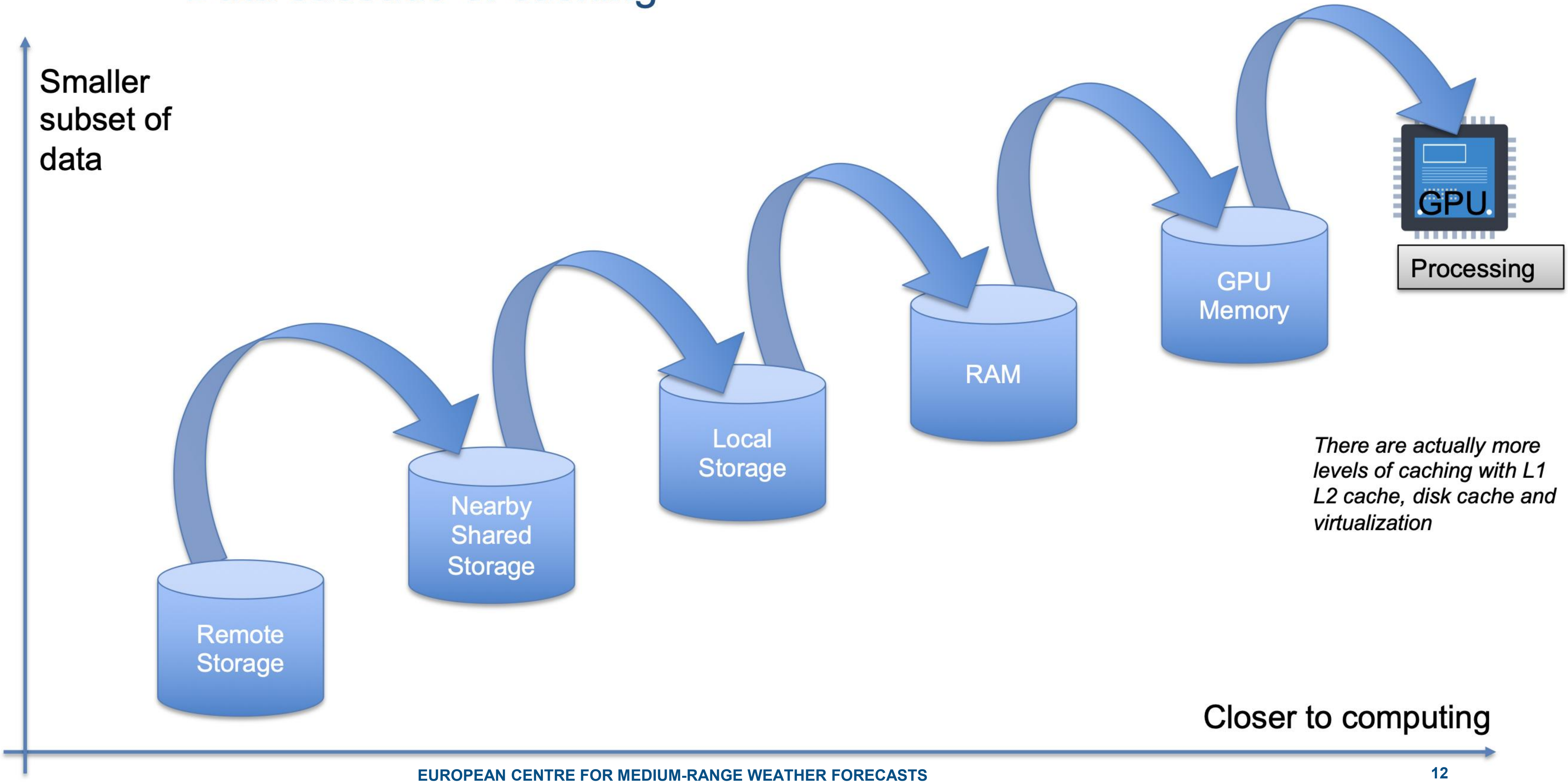
Data Flow: Feeding the GPU with data

- Data is on disk
- We need to move it to the GPU
- One main requirement:

  Faster than GPU processing

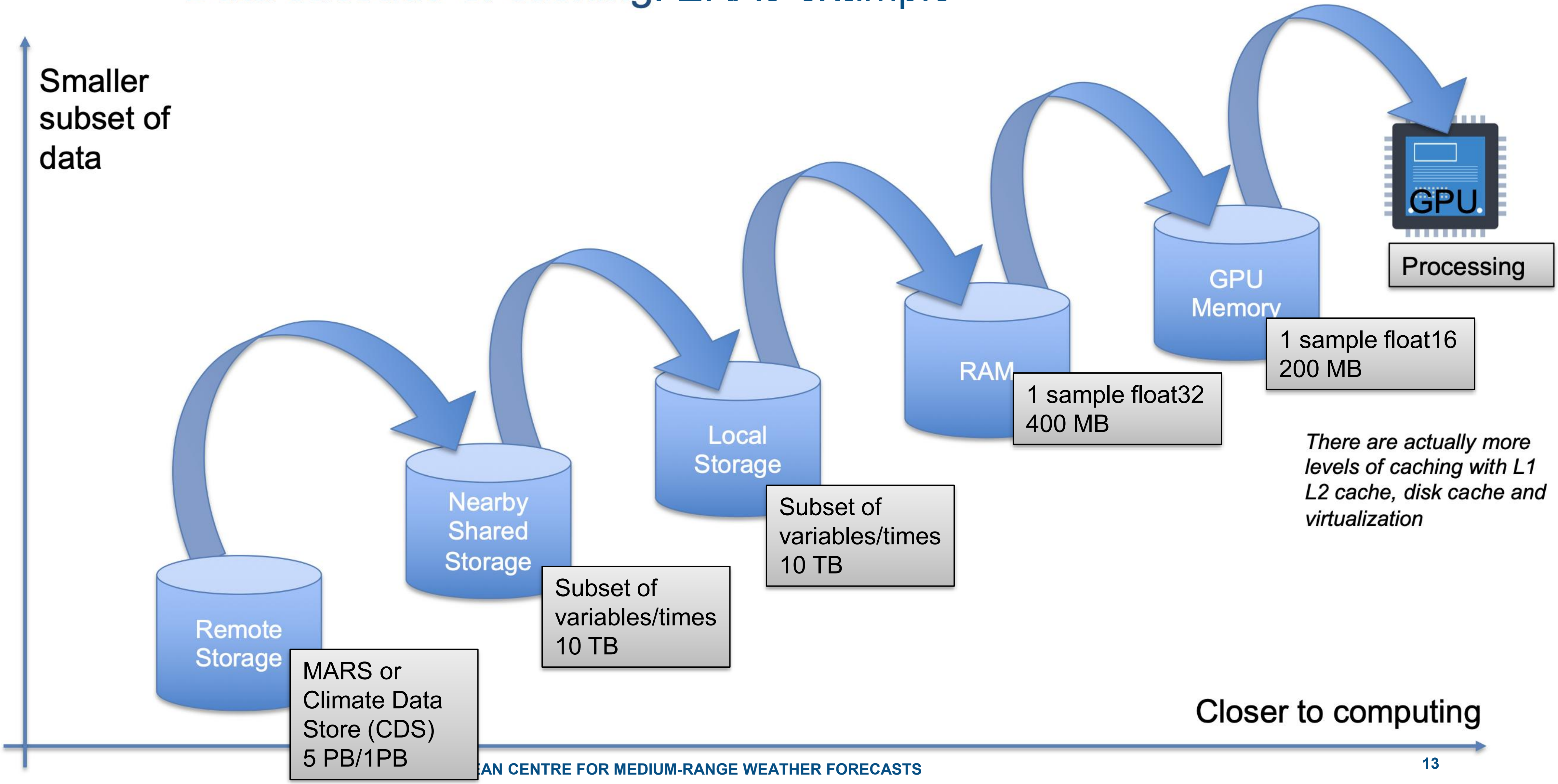


Data cascade of caching



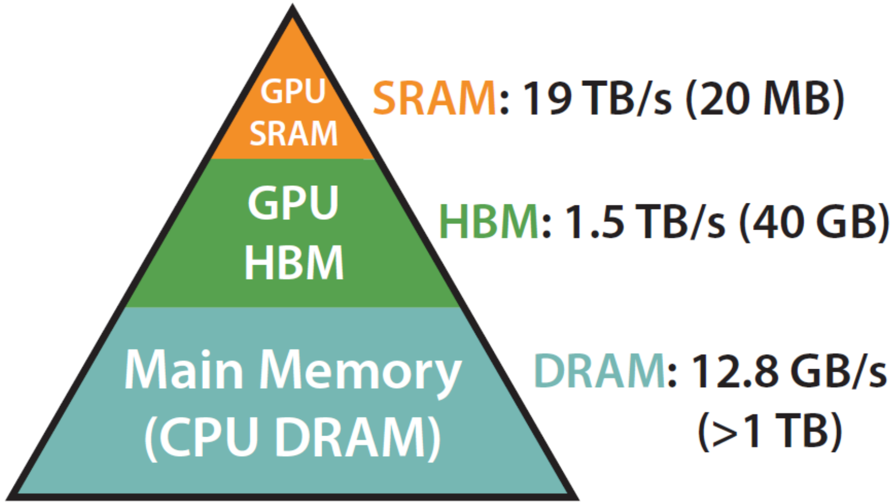
There are actually more levels of caching with L1 L2 cache, disk cache and virtualization

Data cascade of caching: ERA5 example



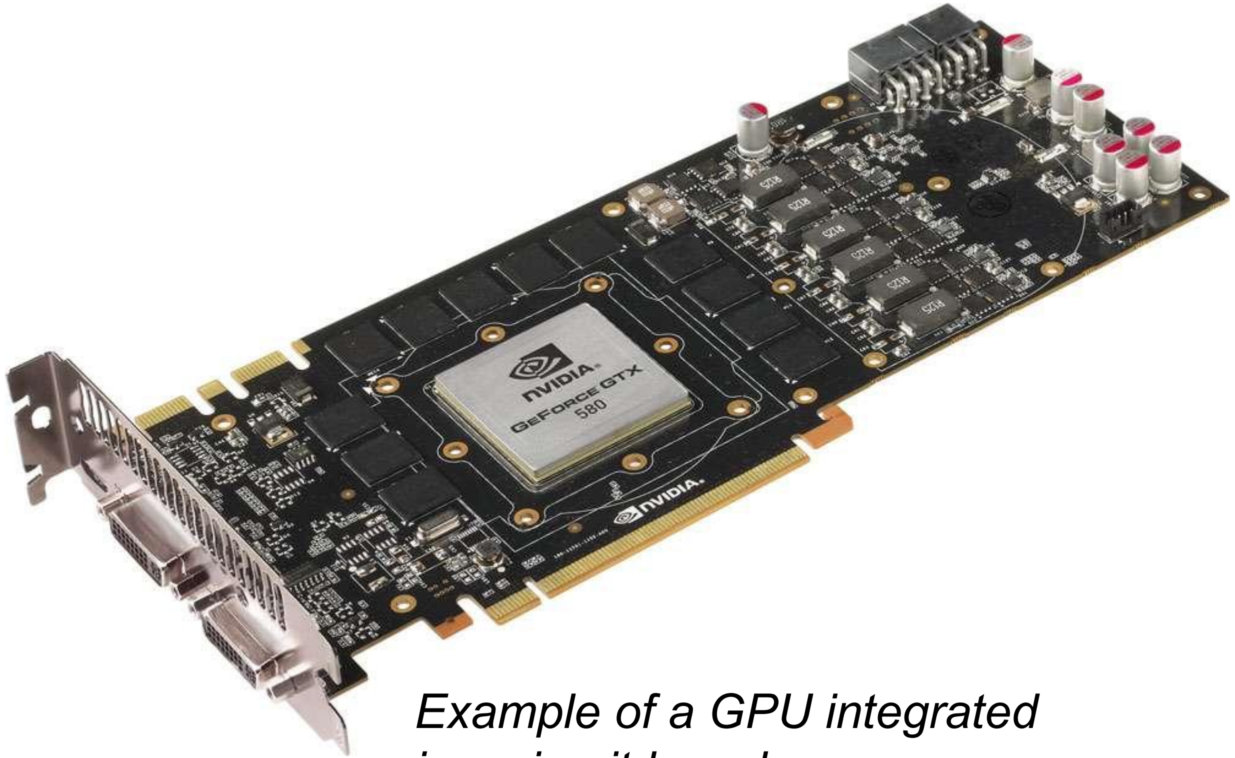
GPU Memory Hierarchy

- Three different level of cache
  - SRAM: Static Random Access Memory
  - HBM: High Bandwidth Memory
  - DRAM: Dynamic Random Access Memory



Memory Hierarchy with Bandwidth & Memory Size

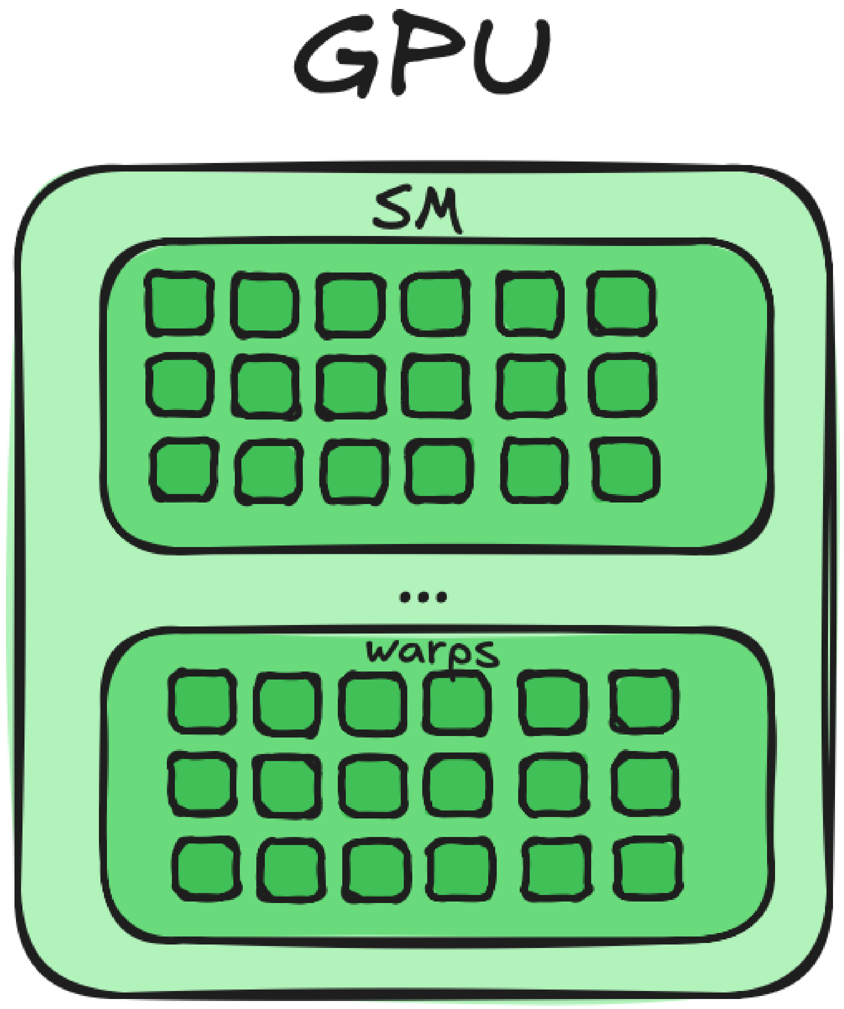
From the FlashAttention paper https://arxiv.org/pdf/2205.14135



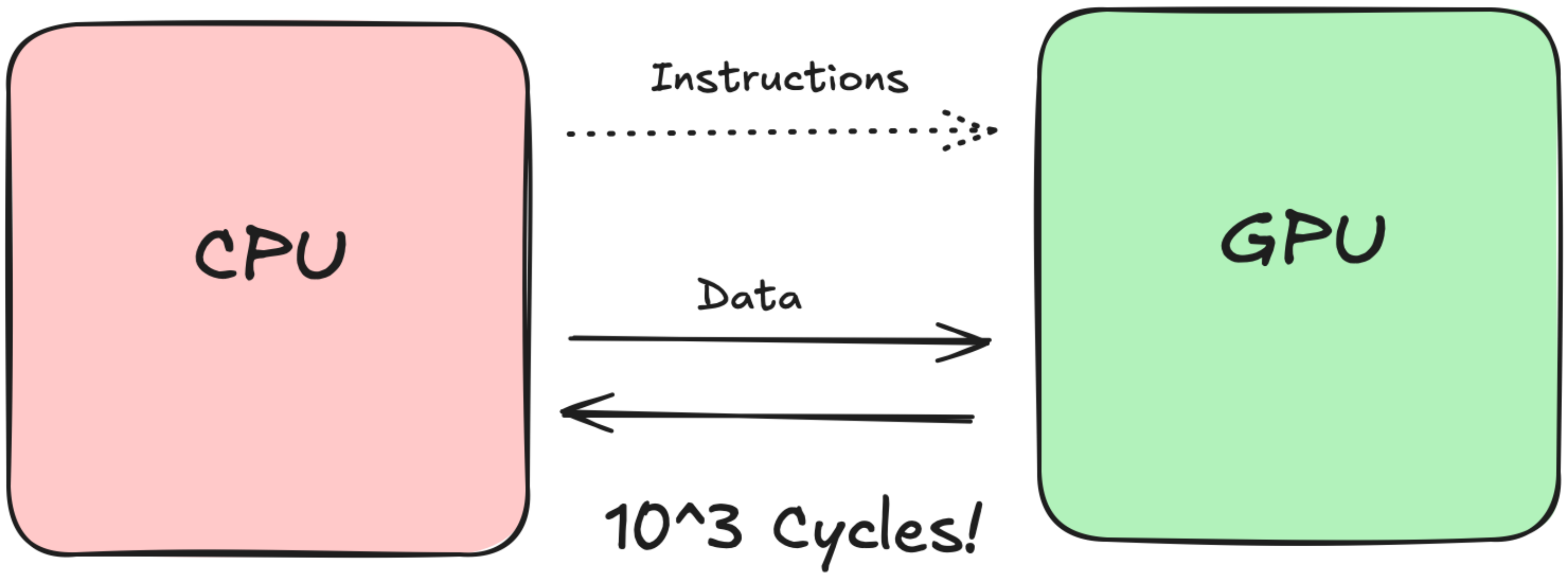
Example of a GPU integrated in a circuit board

What is a GPU?

- GPU global processor assigns work to SM
- Multiple Streaming multiprocessors (SM)
  - Multiples cores
  - Shared memory
  - Register files
  - Warp scheduler
    - Divide thread block into warps of threads (e.g. 32)
- GPU has thousands of simple cores that are great for simple operations (element-wise, convolution and matrix operations, etc.)



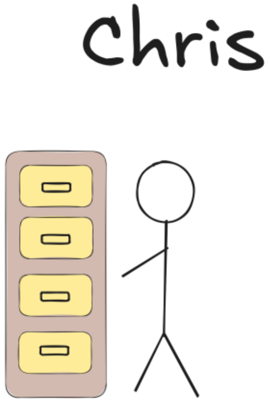
What is a CPU?

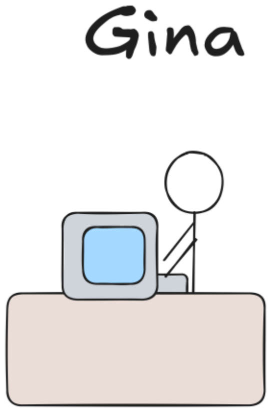


- GPU must be driven by a CPU
  - CPU launches all programs (called ‘kernels’) on GPU
  - CPU and GPU have separate memory spaces and must explicitly transfer data (slow)

CPU

- Primary use in machine learning is to prepare data for the GPU



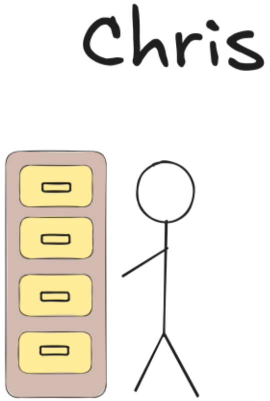


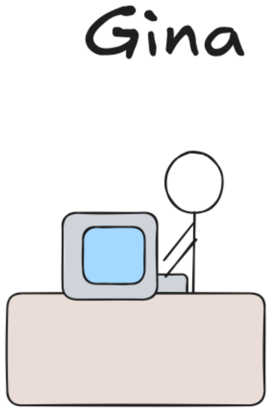
CPU

- Primary use in machine learning is to prepare data for the GPU

CPU

- Primary use in machine learning is to prepare data for the GPU



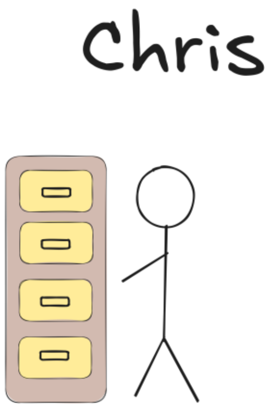


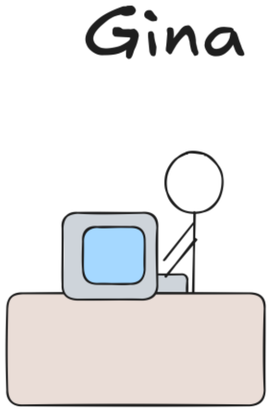
CPU

- Primary use in machine learning is to prepare data for the GPU

CPU

- Primary use in machine learning is to prepare data for the GPU





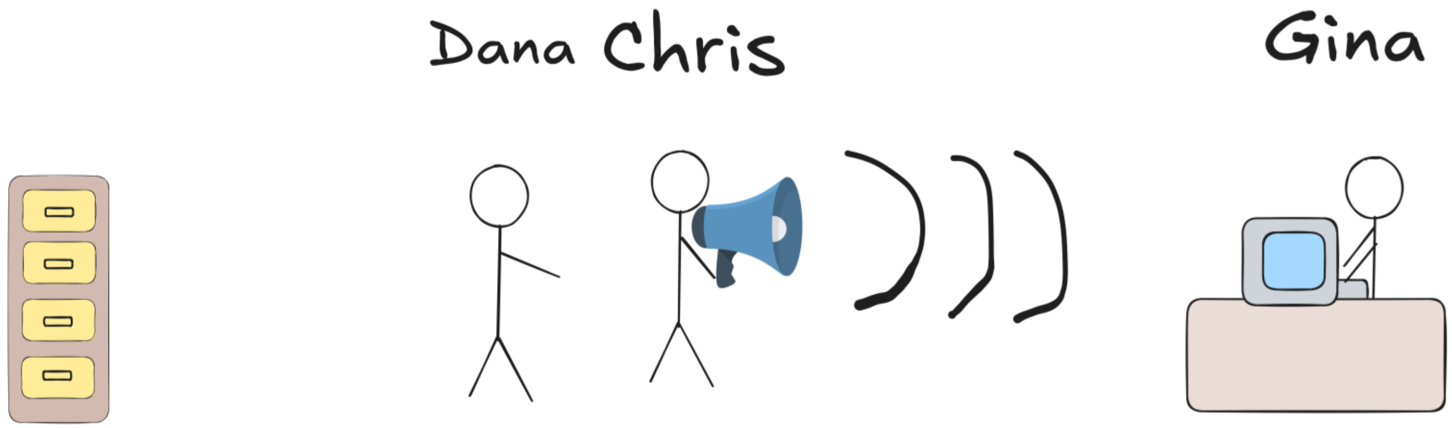

```
dataloader:
  num_workers:
    training: 0
```

CPU

- Primary use in machine learning is to prepare data for the GPU

CPU

- Primary use in machine learning is to prepare data for the GPU

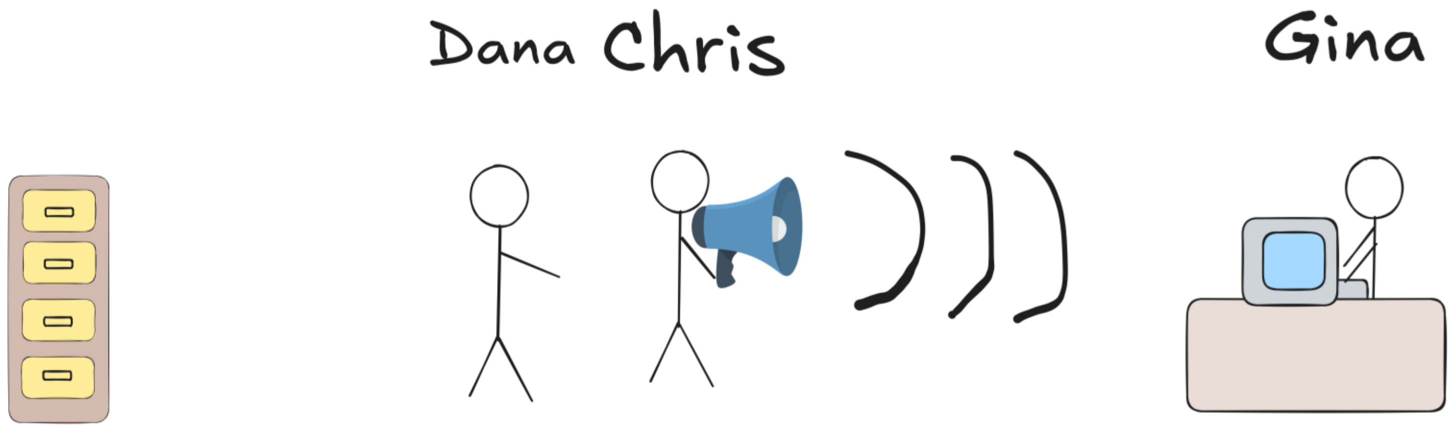


CPU

- Primary use in machine learning is to prepare data for the GPU

CPU

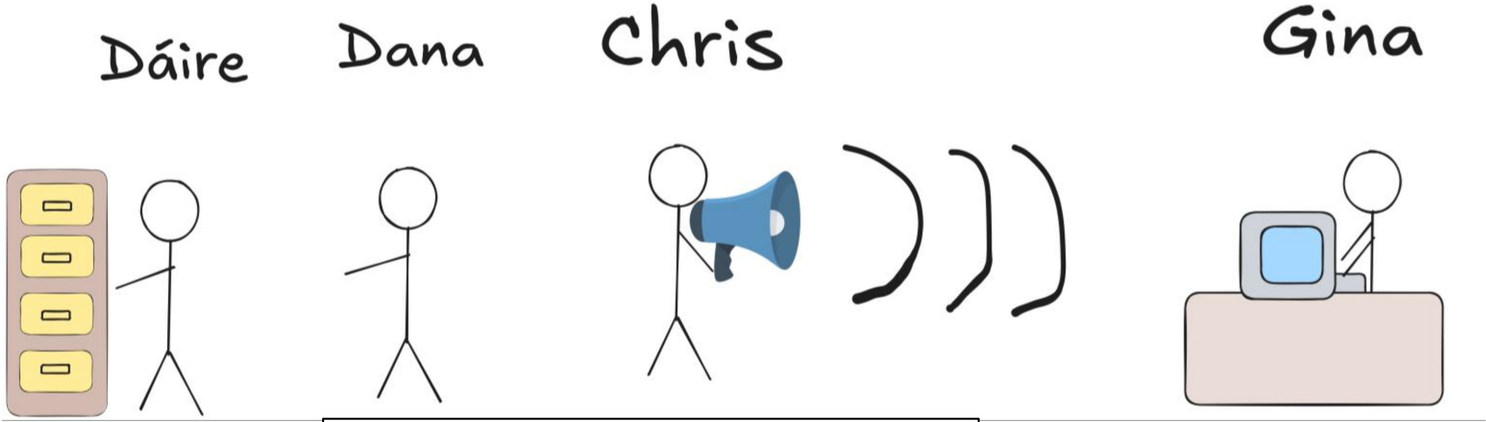
- Primary use in machine learning is to prepare data for the GPU




```
dataloader:
  num_workers:
    training: 1
```

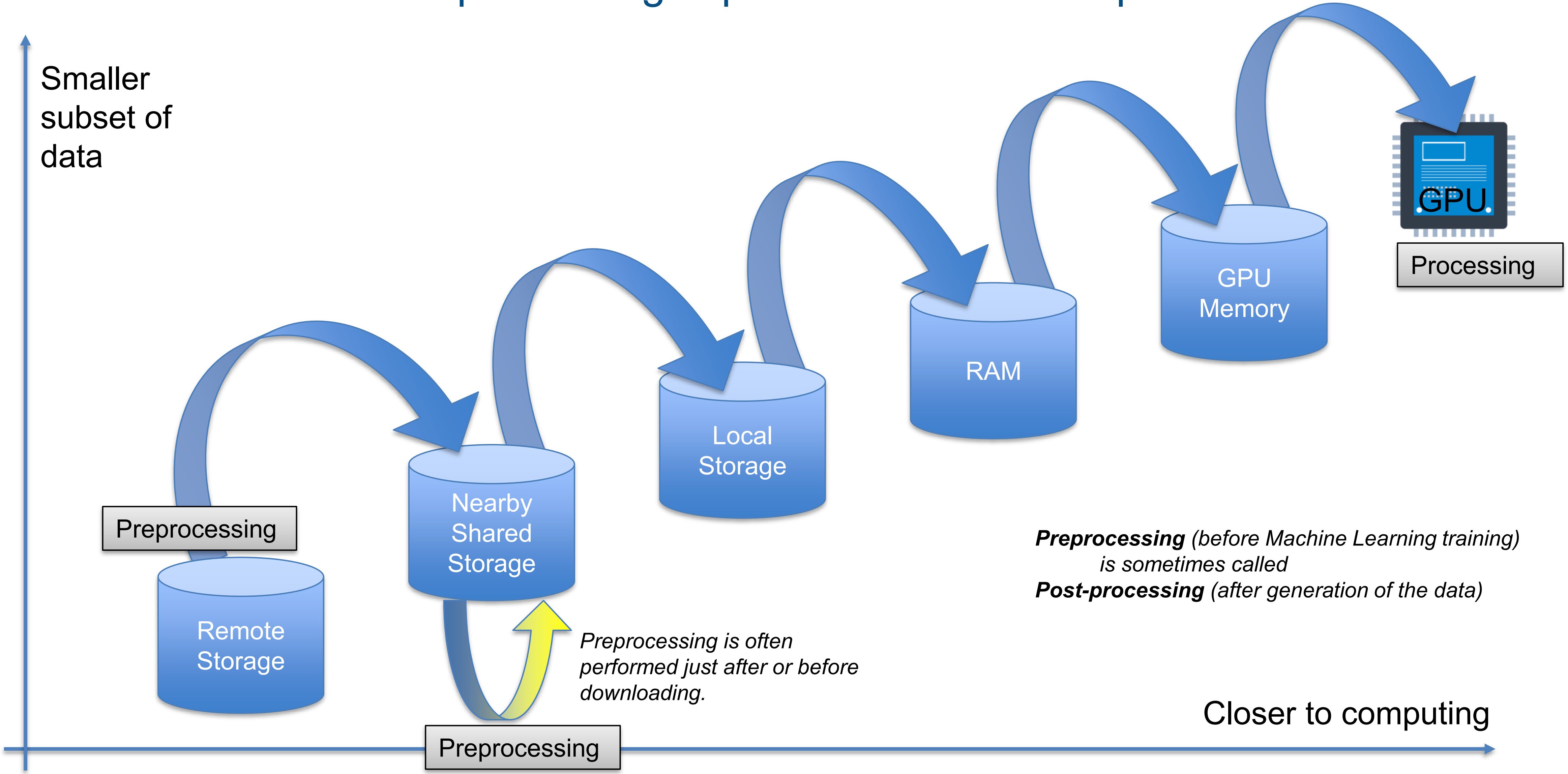
CPU

- Primary use in machine learning is to prepare data for the GPU




```
dataloader:
  num_workers:
    training: 2
```

Data flow: Preprocessing is possible on each step



Preprocessing is often performed just after or before downloading.

Preprocessing *(before Machine Learning training) is sometimes called* ***Post-processing*** *(after generation of the data)*

Summary: But what does this all mean for you?

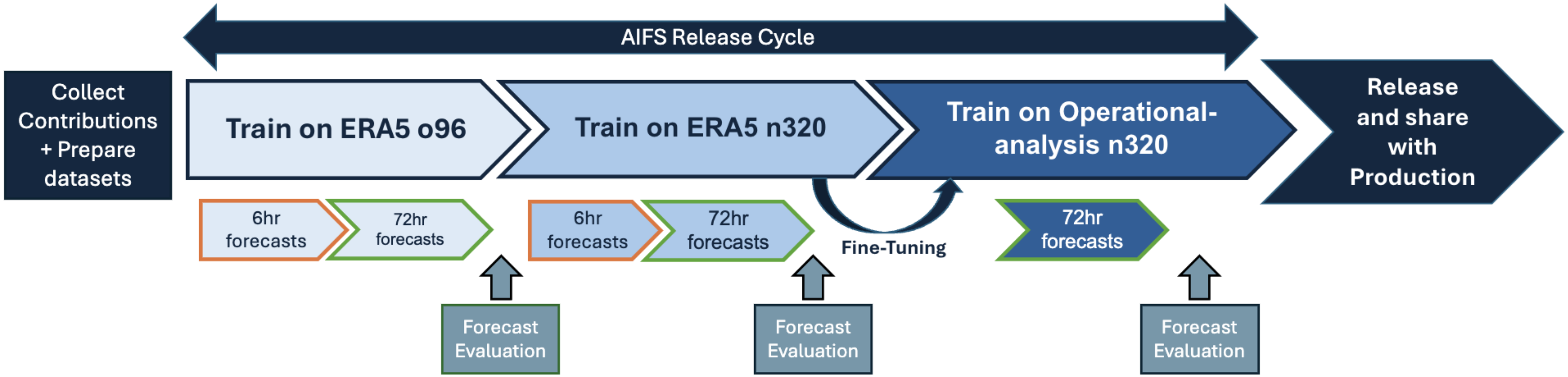
- Most of the complexity of managing memory is handled by PyTorch, Anemoi, etc.
  - Don’t over engineer if working with smaller datasets!
- Try to fill your GPU memory as much as possible
  - Increase batch size, #channels
- Try to fill your CPU memory as much as possible
  - Play with the num_workers parameter
- Think about where your data is coming from
  - Downloading from the internet vs loading from an SSD vs loading from a high performance file system
- Answers will depend on your problem!

Outline

1. **Data Handling**
   - Introduction
   - Data Flow
   - Access Patterns & Datasets
2. **DestinE & Infrastructure**
   - EuroHPC
   - ML applications
     - *Datasets transfer across EuroHPC*
     - *Orchestration of training workflows prototype across EuroHPC*

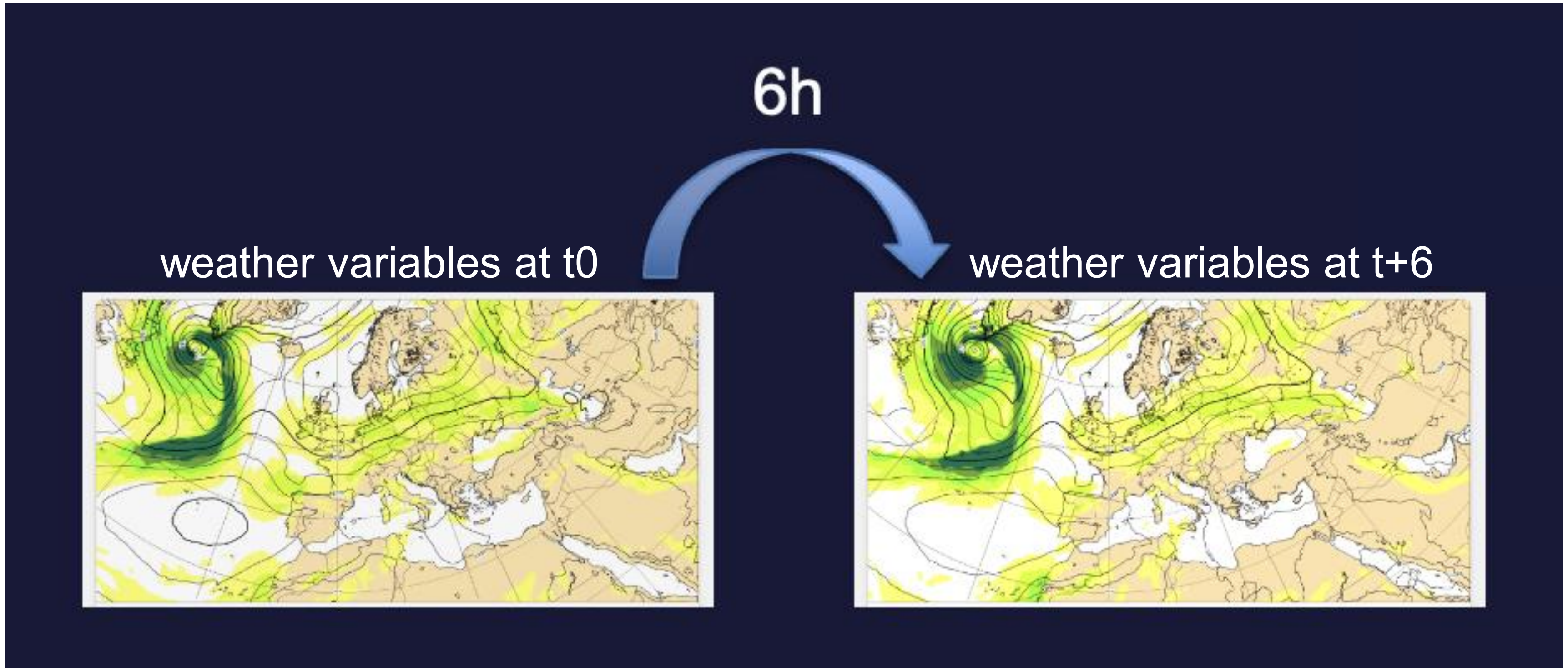
Data Formatting: Know your access patterns!

- The AIFS is trained on the reanalysis datasets **ERA5** produced by the Copernicus Climate Change Service (C3S) at ECMWF and fine-tuned on operational real-time analysis
- The forecast is produced by **autoregressively** stepping 6h into the future



Data Formatting: Know your access patterns!

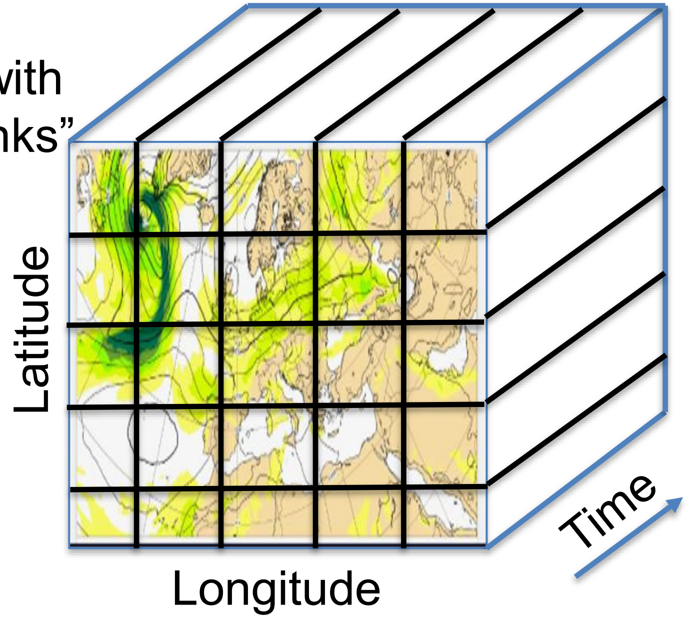
- The AIFS is trained on the reanalysis datasets **ERA5** produced by the Copernicus Climate Change Service (C3S) at ECMWF and fine-tuned on operational real-time analysis
- The forecast is produced by **autoregressively** stepping 6h into the future
- Dataset dimensions: [**time**, **latitude**, **longitude**, **variable**]



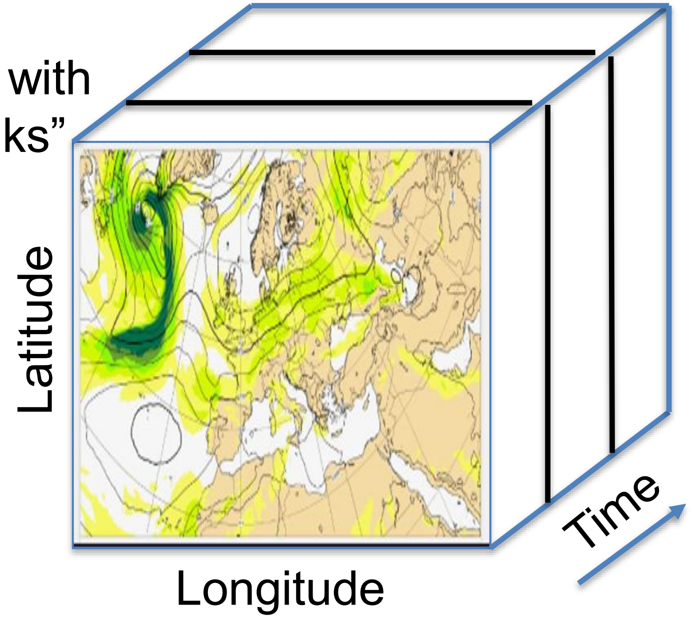
Data Formatting: Know your access patterns!

- The AIFS is trained on the reanalysis datasets **ERA5** produced by the Copernicus Climate Change Service (C3S) at ECMWF and fine-tuned on operational real-time analysis
- The forecast is produced by **autoregressively** stepping 6h into the future
- Dataset dimensions: [**time**, **longitude**, **latitude**, **variable**]
- Approximate sizes for AIFS: [**67000**, **1440**, **720**, **100**]

Save data with space “chunks”

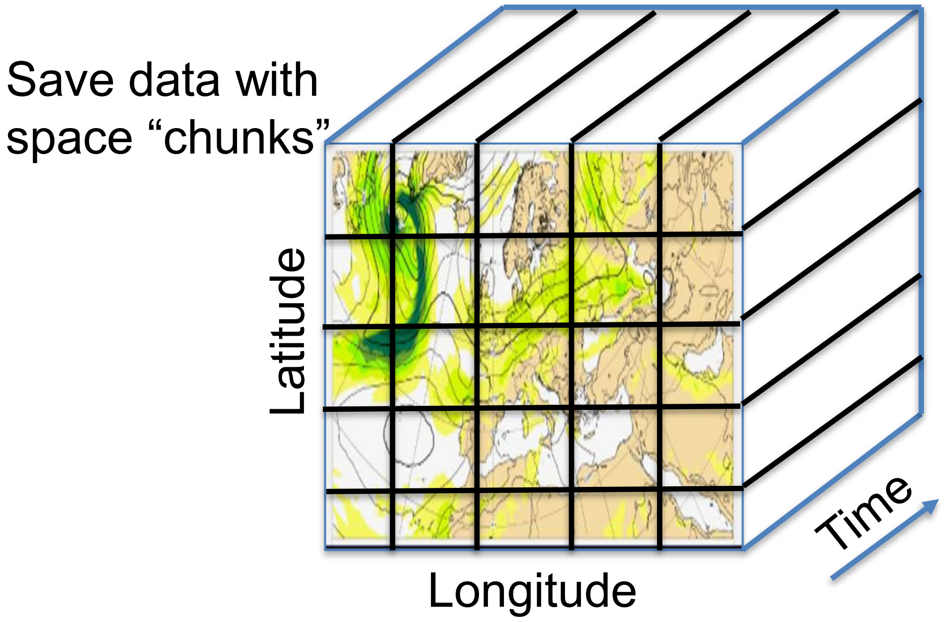


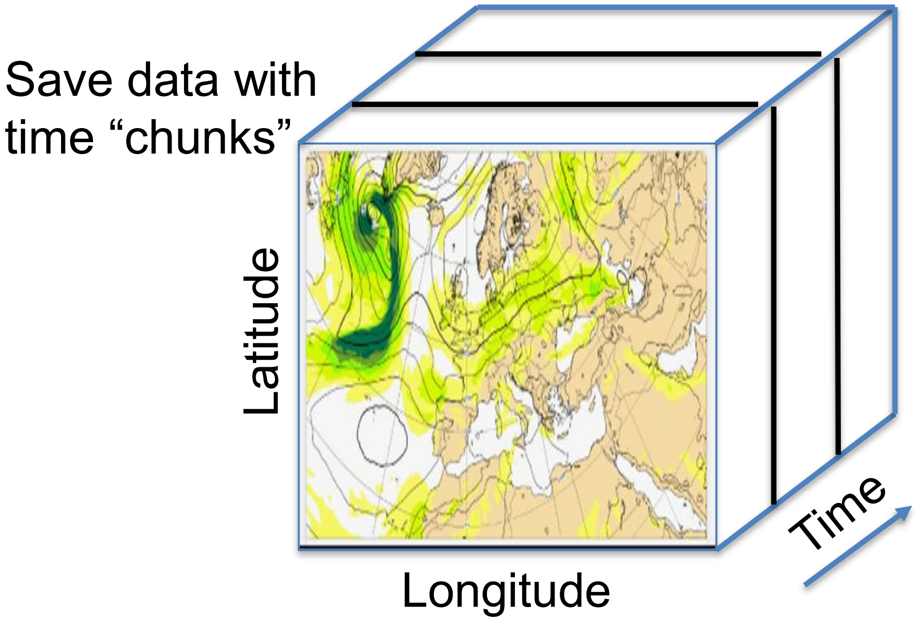
Save data with time “chunks”



Data Formatting: Know your access patterns!

- The AIFS is trained on the reanalysis datasets **ERA5** produced by the Copernicus Climate Change Service (C3S) at ECMWF and fine-tuned on operational real-time analysis
- The forecast is produced by **autoregressively** stepping 6h into the future
- Dataset dimensions: [**time**, **space**, **variable**]
- Approximate sizes for AIFS: [**67000**, **542080**, **100**]





Know your access patterns: transposition

Example data

9 fields (f)

14 dates (t)

Real data

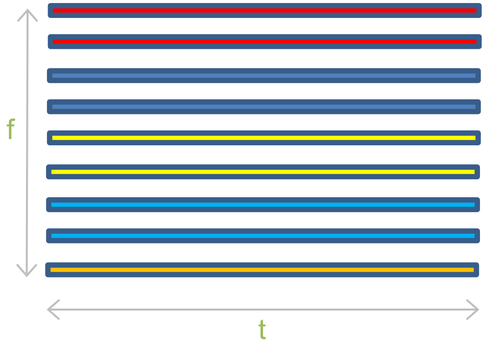
100 fields

1M dates

+ additional

dimensions

Data saved by field

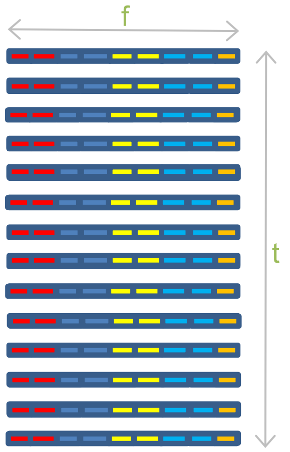


One file contains the timeseries of one given **field**

9 fields

≠

Data saved by date

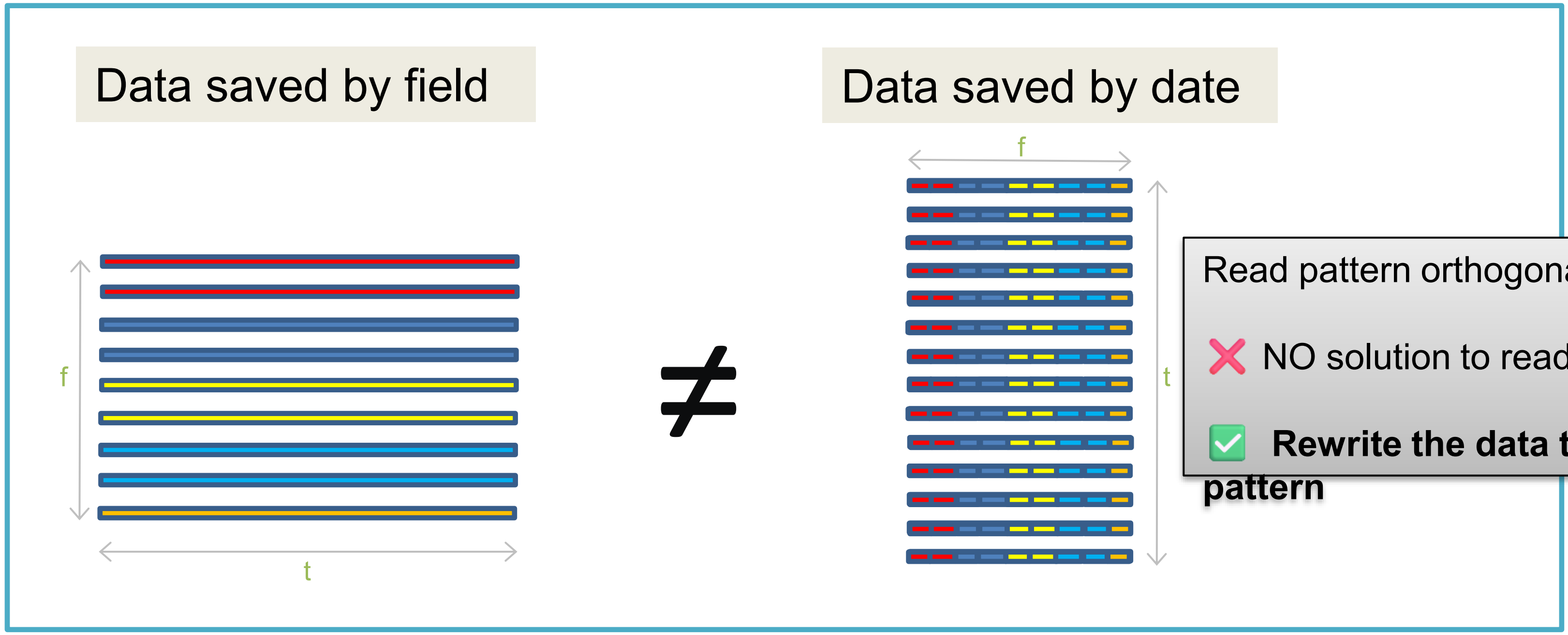


One file contains the set of values for one given **date**

9 dates

More generally:
Each "file" here could be an S3 object, or a part of a file, a record in a database, etc.

Know your access patterns: transposition

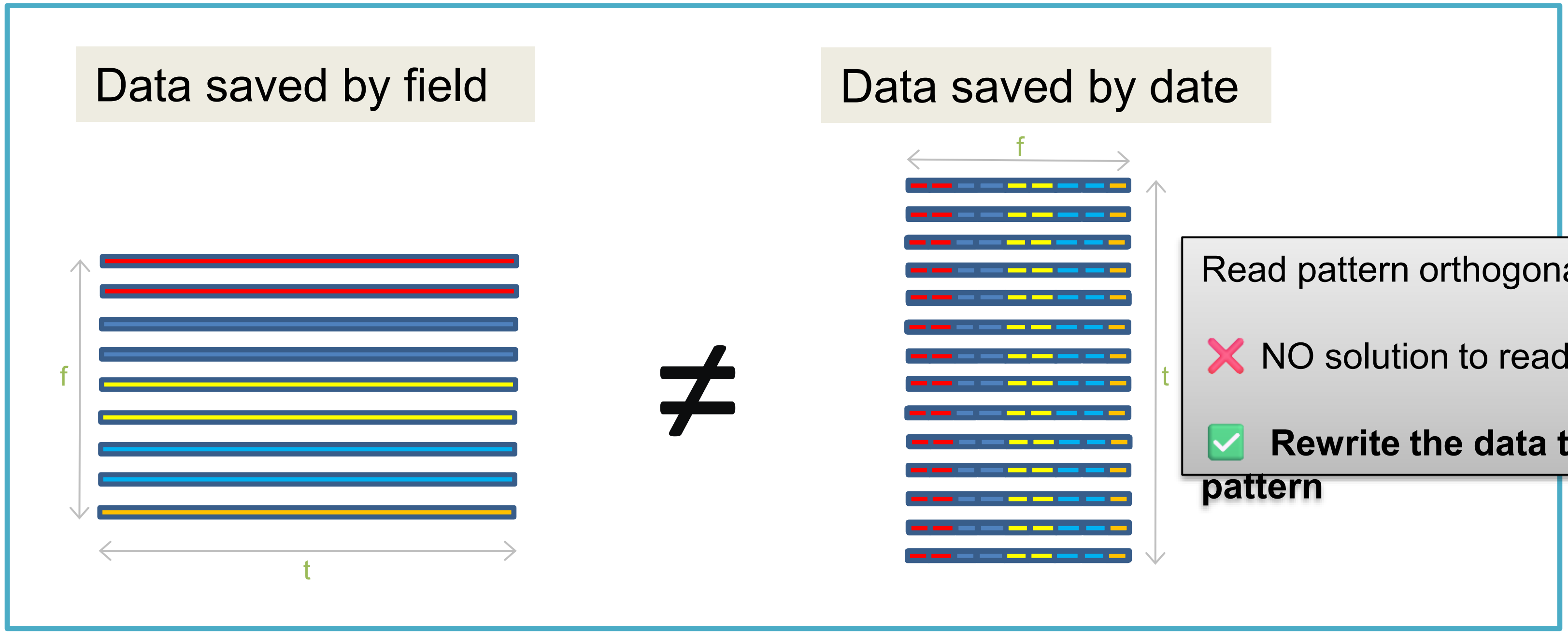


Read pattern orthogonal to the pattern on disk ?

❌ NO solution to read big data efficiently

✅ **Rewrite the data to adapt to the read pattern**

Know your access patterns: transposition



Read pattern orthogonal to the pattern on disk ?

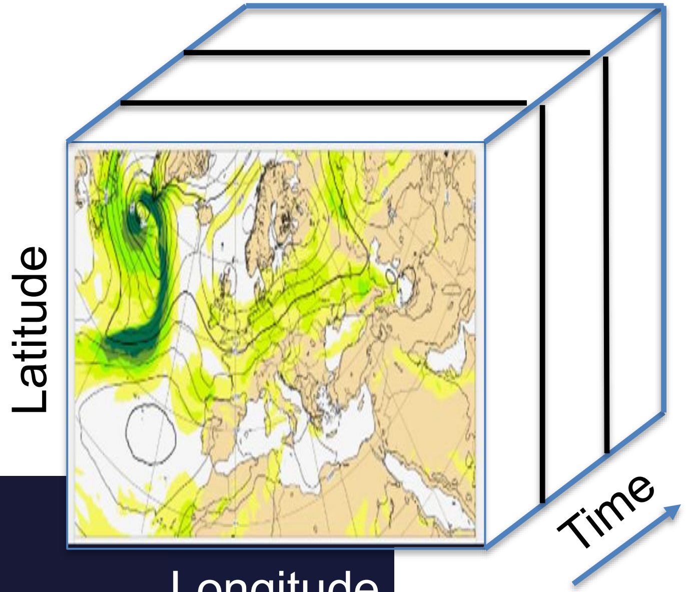
❌ NO solution to read big data efficiently

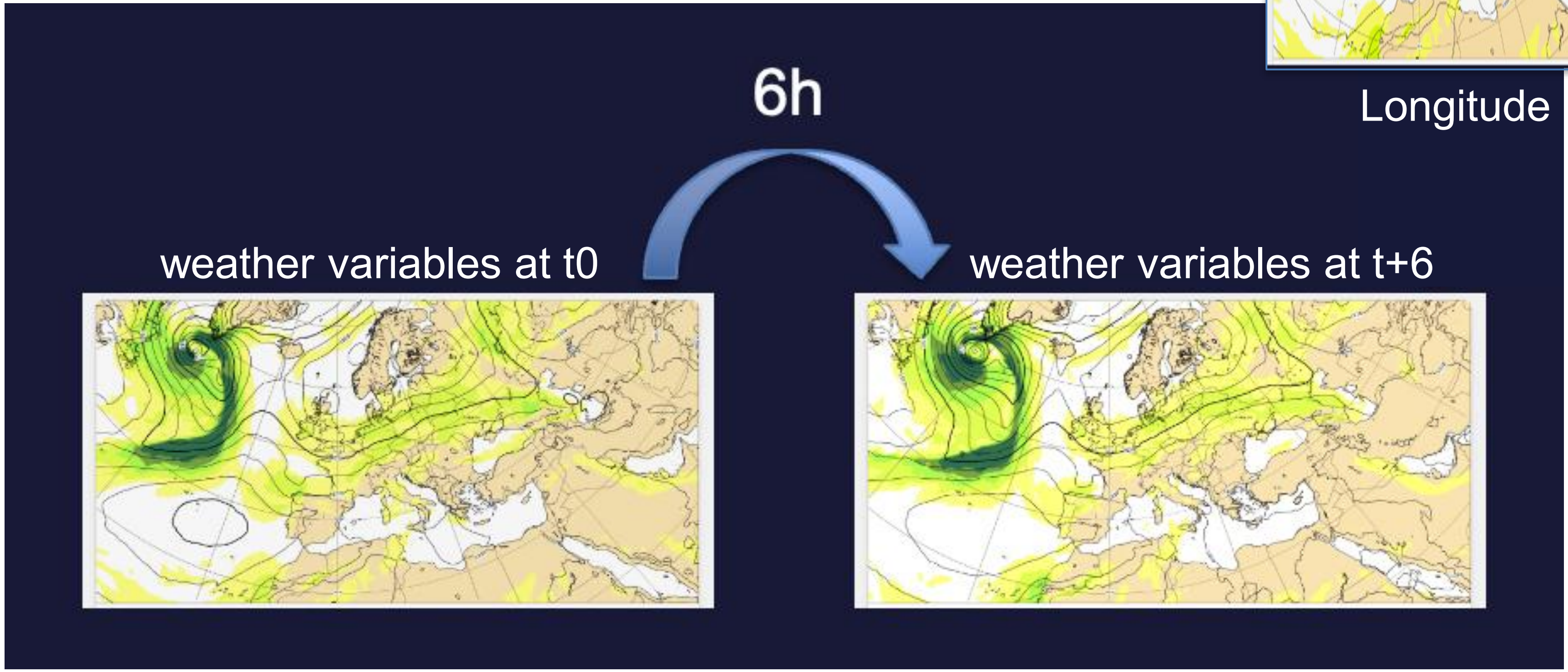
✅ **Rewrite the data to adapt to the read pattern**

- If you need to read data by timeseries. Store the data by timeseries.
- If you read training samples with 100 parameters for 1 date, do save files containing 100 parameter for 1 dates (but ”it depends”)
- General solution: create a dataset dedicated to your training task (perform the transposition **offline** if required).

Other examples at ECMWF

- ECMWF AIFS – Forecasting Weather in next 6-hour time slice
- Needs access to **previous 6-hour time slice** to predict next step
- Dataset dimensions: [**time**, space, variable] (**bold** = chunked dimension)

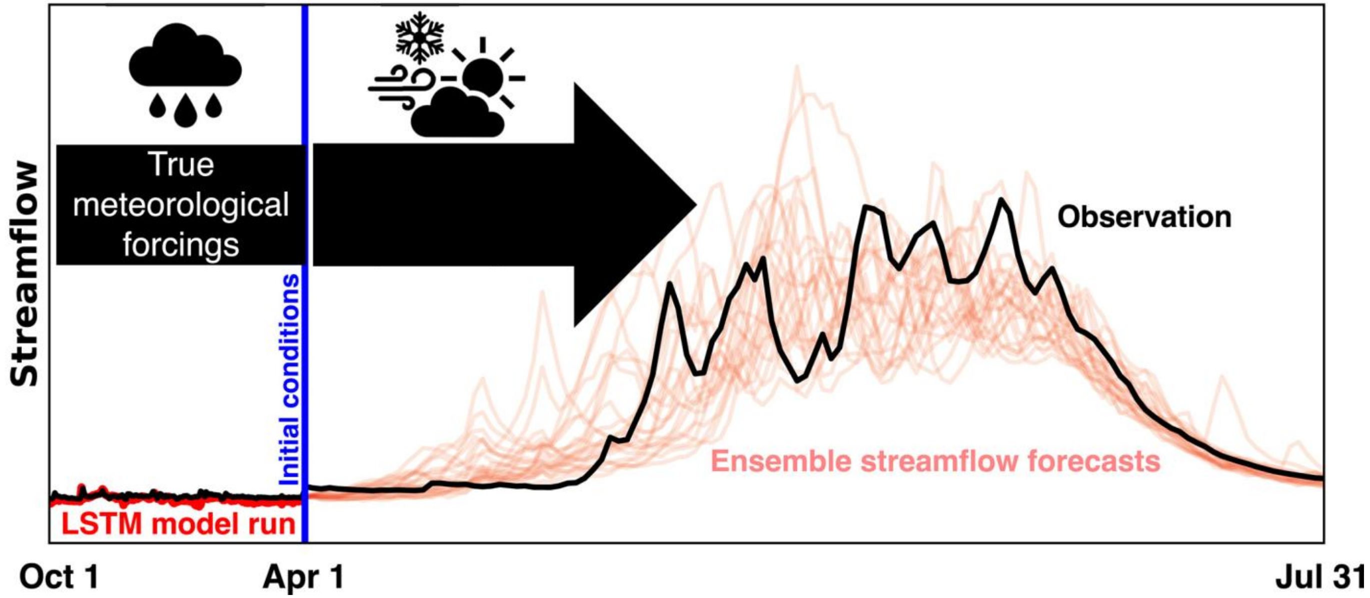


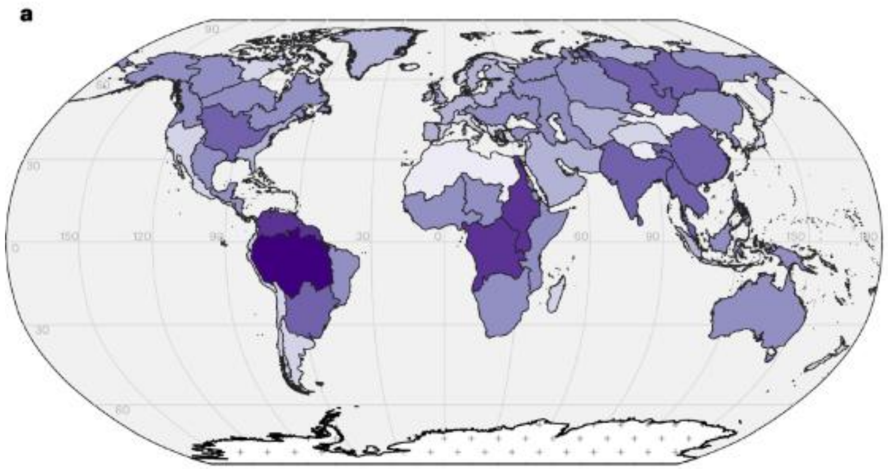


Other examples at ECMWF

Credit:
Maria Luisa Taccari
Kenza Tazi

- ECMWF AIFL – Streamflow forecasting using a Long Short-Term Memory Neural Network (LSTM)
- Needs access to **~180 days** of previously data to predict next step
- Dataset dimensions: [*time*, space, variable] (**bold** = chunked dimension)



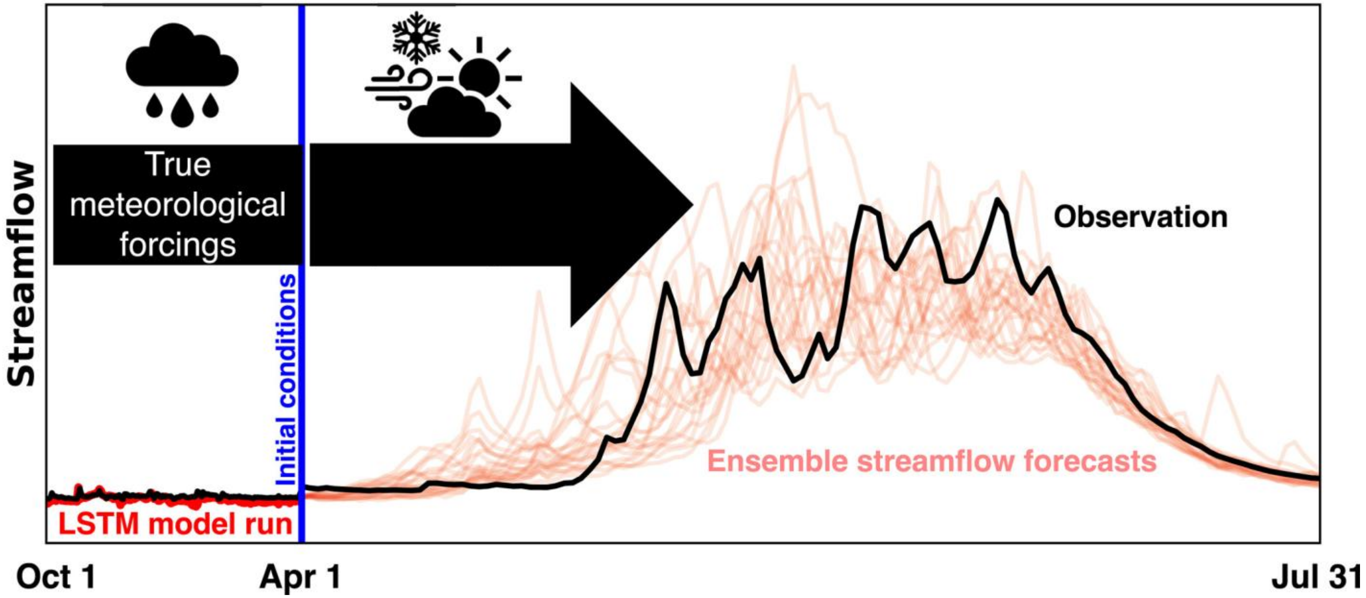


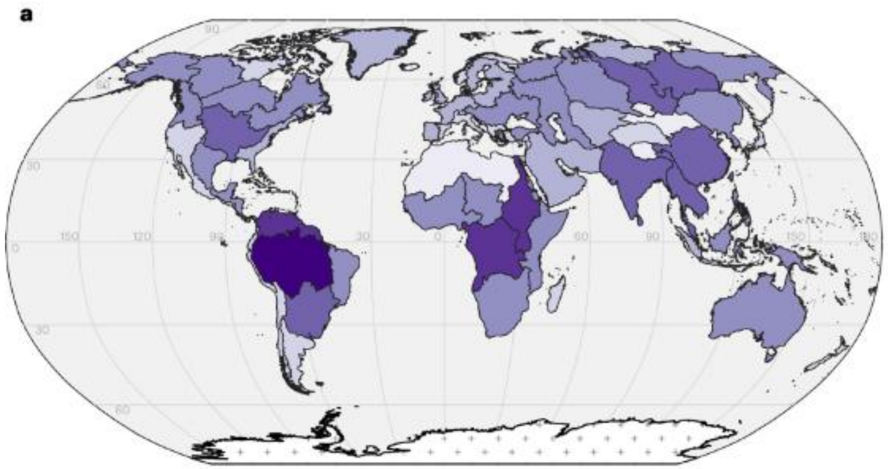
Global hydrological catchments

Other examples at ECMWF

Credit:
Maria Luisa Taccari
Kenza Tazi

- ECMWF AIFL – Streamflow forecasting using a Long Short-Term Memory Neural Network (LSTM)
- Needs access to ~**180 days** of previously data to predict next step
- Dataset dimensions: [*time*, **space**, variable] (**bold** = chunked dimension)



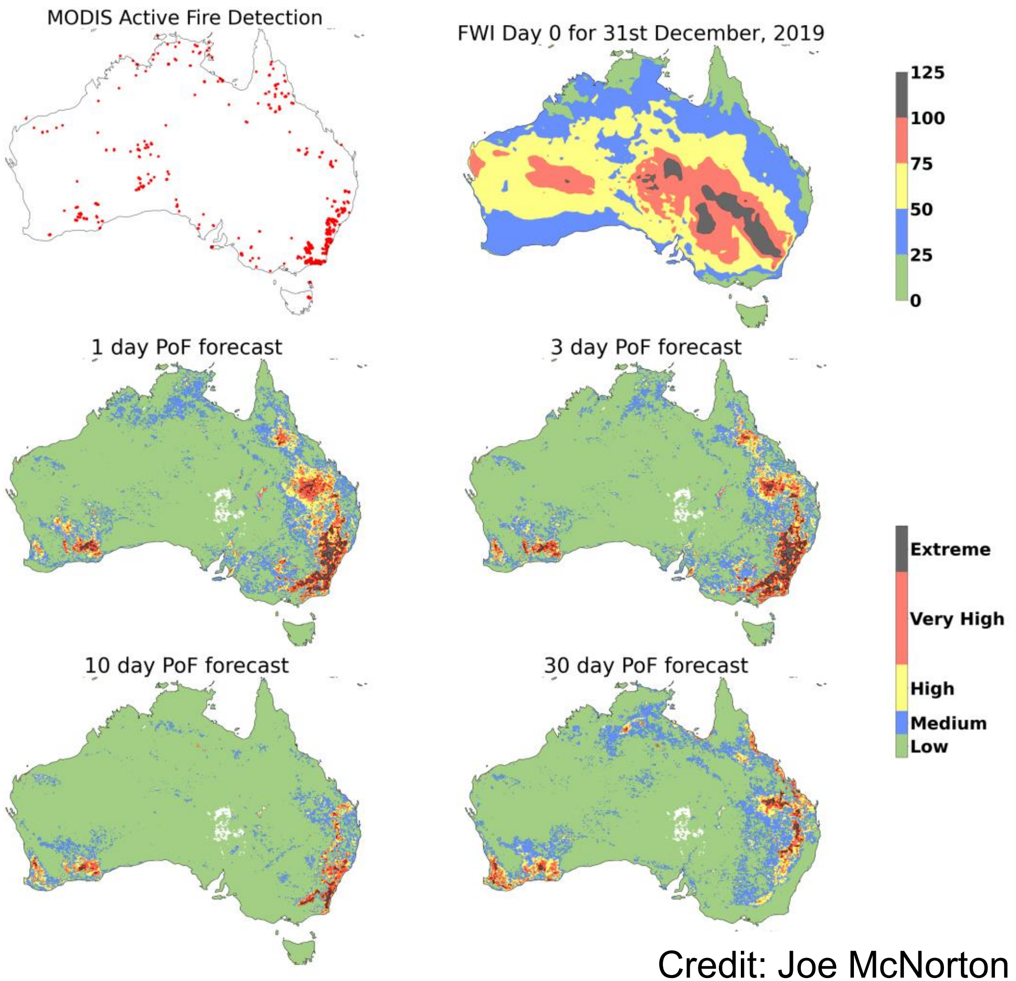


Global hydrological catchments

Other examples at ECMWF

- ECMWF Probability of Fire (PoF) – Predicting fire probabilities with XGBoost
  - (training on a subset of the data = 10%)
- Needs access to **instantaneous** data for set of variables for single grid points to predict probability of fire
- Dataset dimensions: [**time**, **space**, variable] (**bold** = chunked dimension)

Time	Lat	Lon	Weather	Fuel load	Static feats	MODIS active fires



Credit: Joe McNorton

Geophysical Research Letters

Research Letter | Open Access

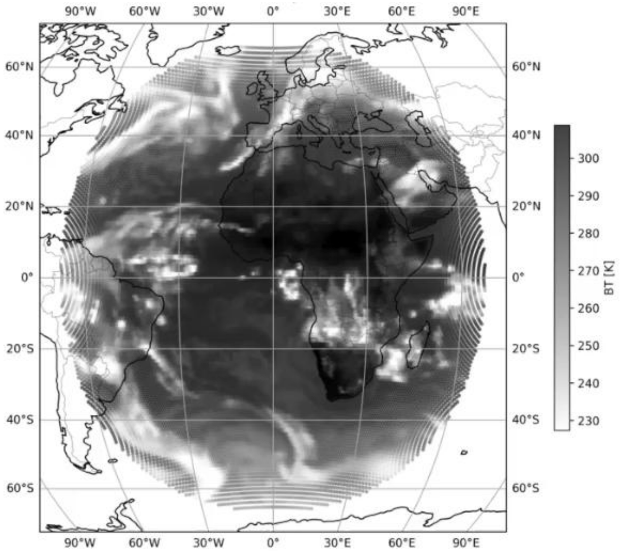
A Global Probability-Of-Fire (PoF) Forecast

J. R. McNorton, F. Di Giuseppe, E. Pinnington, M. Chantry, C. Barnard

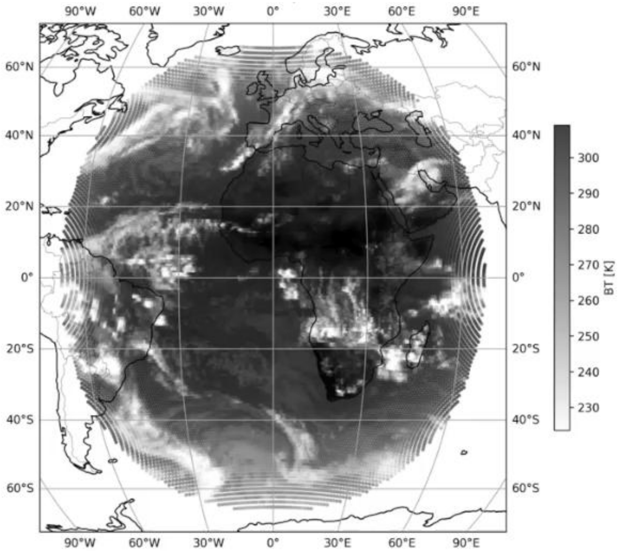
First published: 14 June 2024 | https://doi.org/10.1029/2023GL107929 |

Other examples at ECMWF

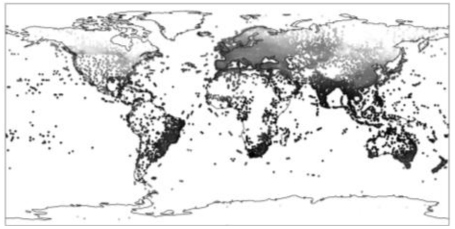
- ECMWF Direct Observation Prediction – Predicting observations into the future
- Gets observations in a time window (e.g. 12hrs) and predicts next time window of observations
- Dataset dimensions: [**time**, **space**, **observation-type**, variable] (**bold** = chunked dimension)

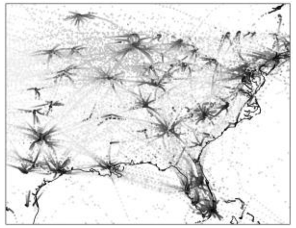


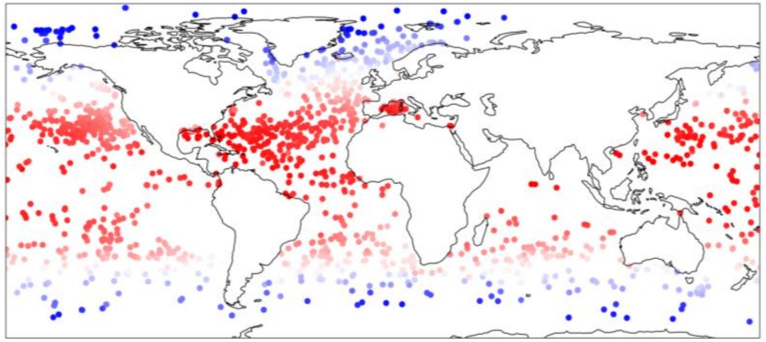
Model prediction

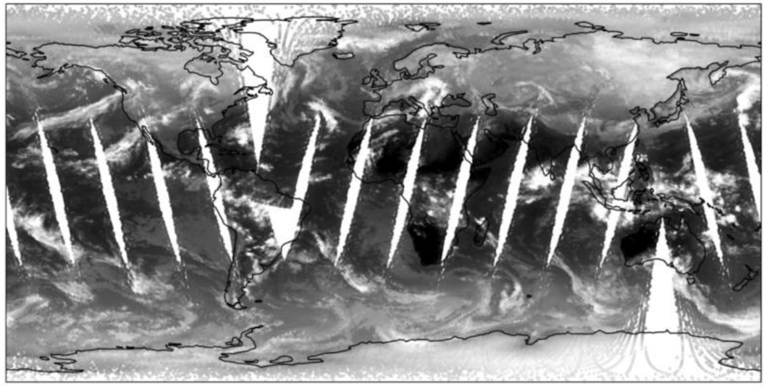


Target Observations









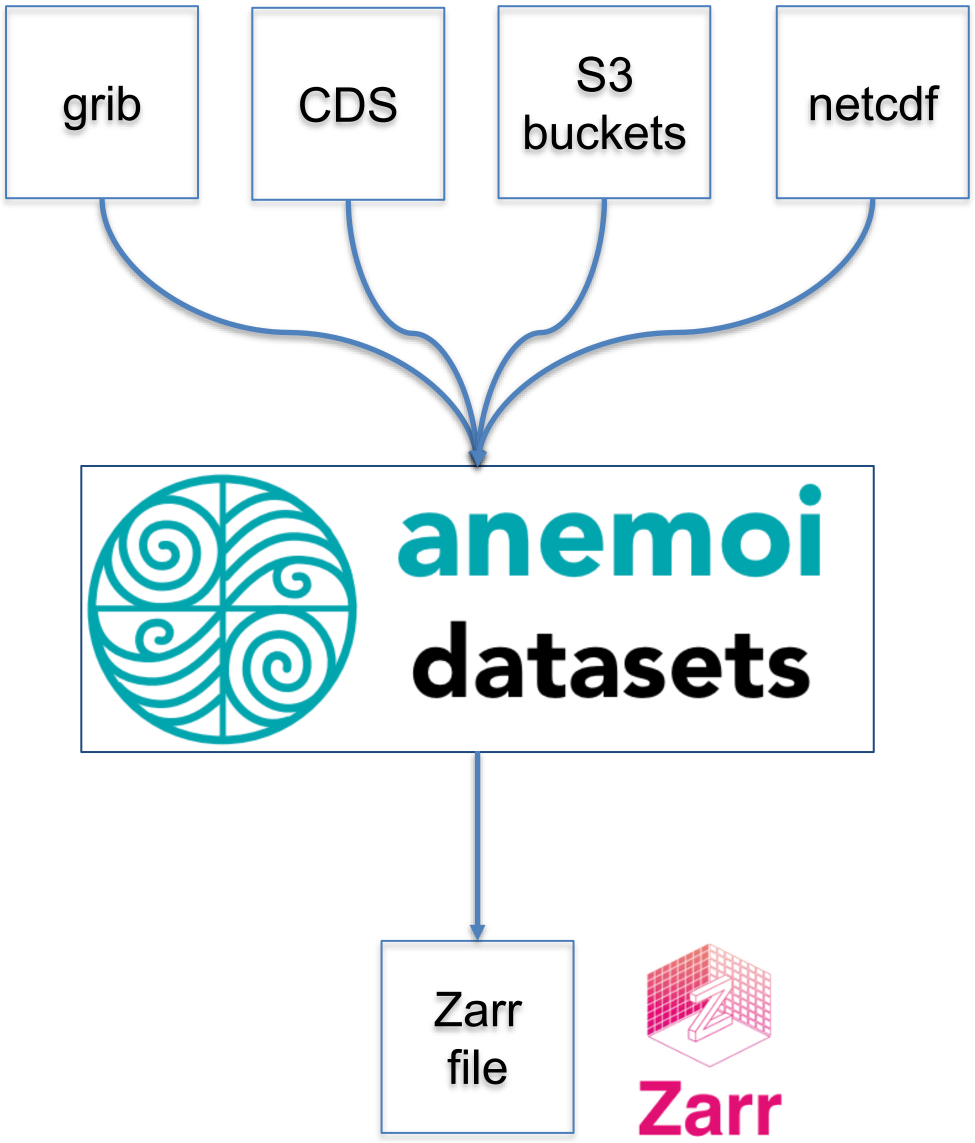
Credit: E. Piddington

Creating datasets optimized for training

- Create “machine-learning ready” datasets for training data-driven weather forecasts
- Make the loading of data samples as efficient as possible
- Provide rich metadata that can be used in training and inference
- https://github.com/ecmwf/anemoi-datasets
- Consider the datasets created by anemoi-datasets is pre-processed data that are as close as possible to the needs of training

Creating datasets optimized for training

- To make training efficient we often have to rewrite or “rechunk” our data to match the read patterns
- Lots of efficient data formats to do this: HDF5, NetCDF4, Grib, Zarr, Parquet (tabular data), Cloud Optimized Geotiffs, …
- For examples at ECMWF we are using **Anemoi-Datasets** to rewrite data into optimized Zarr datasets for Machine Learning
- https://github.com/ecmwf/anemoi-datasets
- More info @ Thursday Harrison



To summarise

- Training ML models can be limited by how fast we can load the data
- Need to consider data access patterns and pre-process data accordingly! (can use e.g. Zarr, Anemoi, etc. for this)
- Tweak batch size and num_workers to try and optimize training times once data formatted correctly
- Think about where data is coming from
  - Bring code to the data/compute

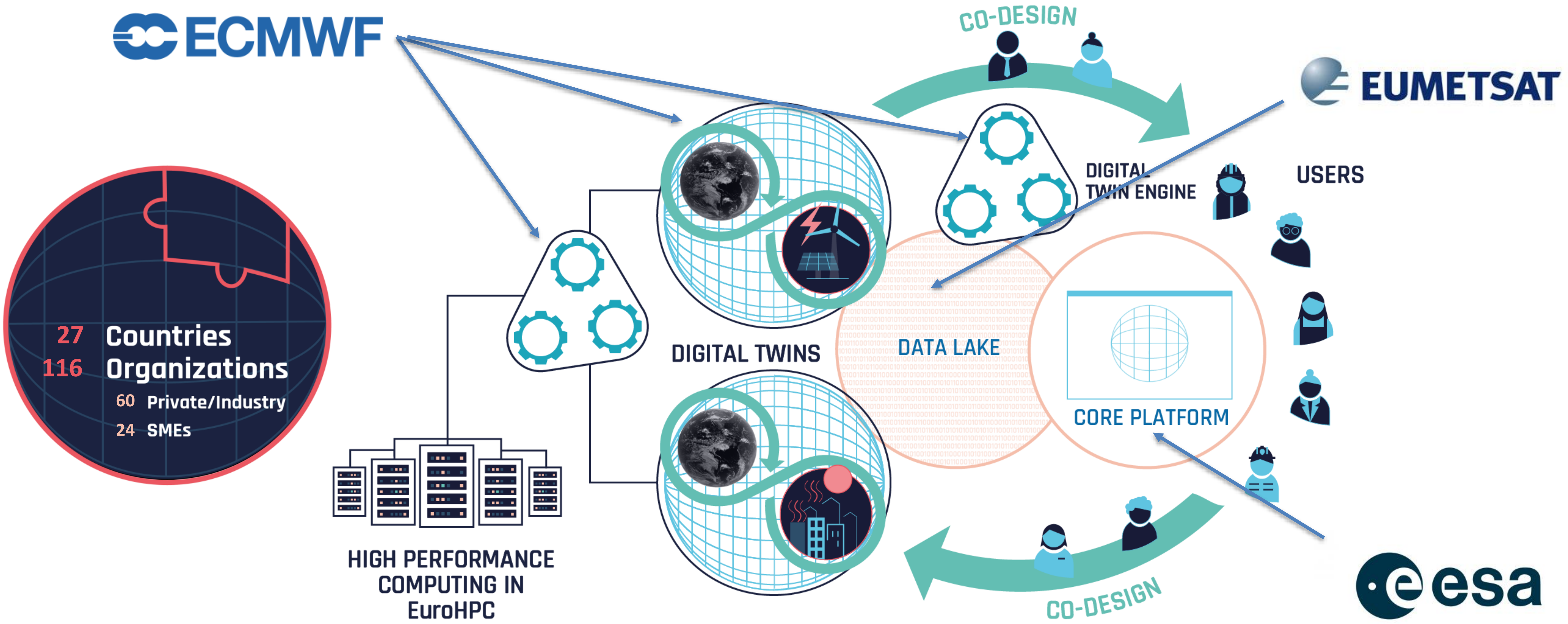
Outline

1. **Data Handling**
   - Introduction
   - Data Flow
   - Access Patterns & Datasets
2. **DestinE & Infrastructure**
   - EuroHPC
   - ML applications
     - *Datasets transfer across EuroHPC*
     - *Orchestration of training workflows prototype across EuroHPC*

Outline

1. **Data Handling**
   - Introduction
   - Data Flow
   - Access Patterns & Datasets
2. **DestinE & Infrastructure**
   - EuroHPC
   - ML applications
     - *Datasets transfer across EuroHPC*
     - *Orchestration of training workflows prototype across EuroHPC*

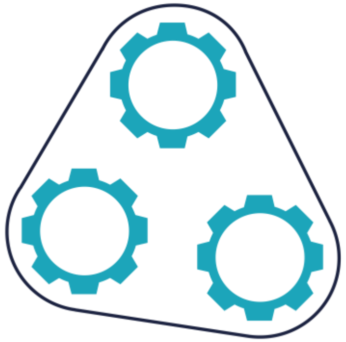
Destination Earth



https://destination-earth.eu

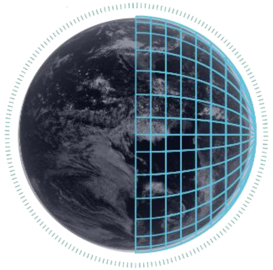
ECMWF

Destination Earth – Digital Twins Engine (DTE)

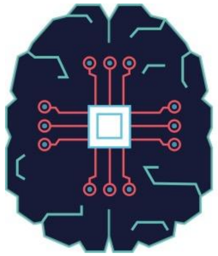


- Software stack
    - Supported by dedicated infrastructures (cloud-based)

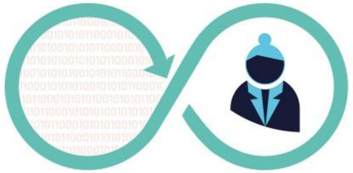
Ensuring complex simulations are run efficiently on EuroHPC



Powering the digital twins and managing big data



Using ML/AI to increase the efficiency of the digital twins and estimate uncertainty



Tailoring information to user's needs and interactivity

ECMWF

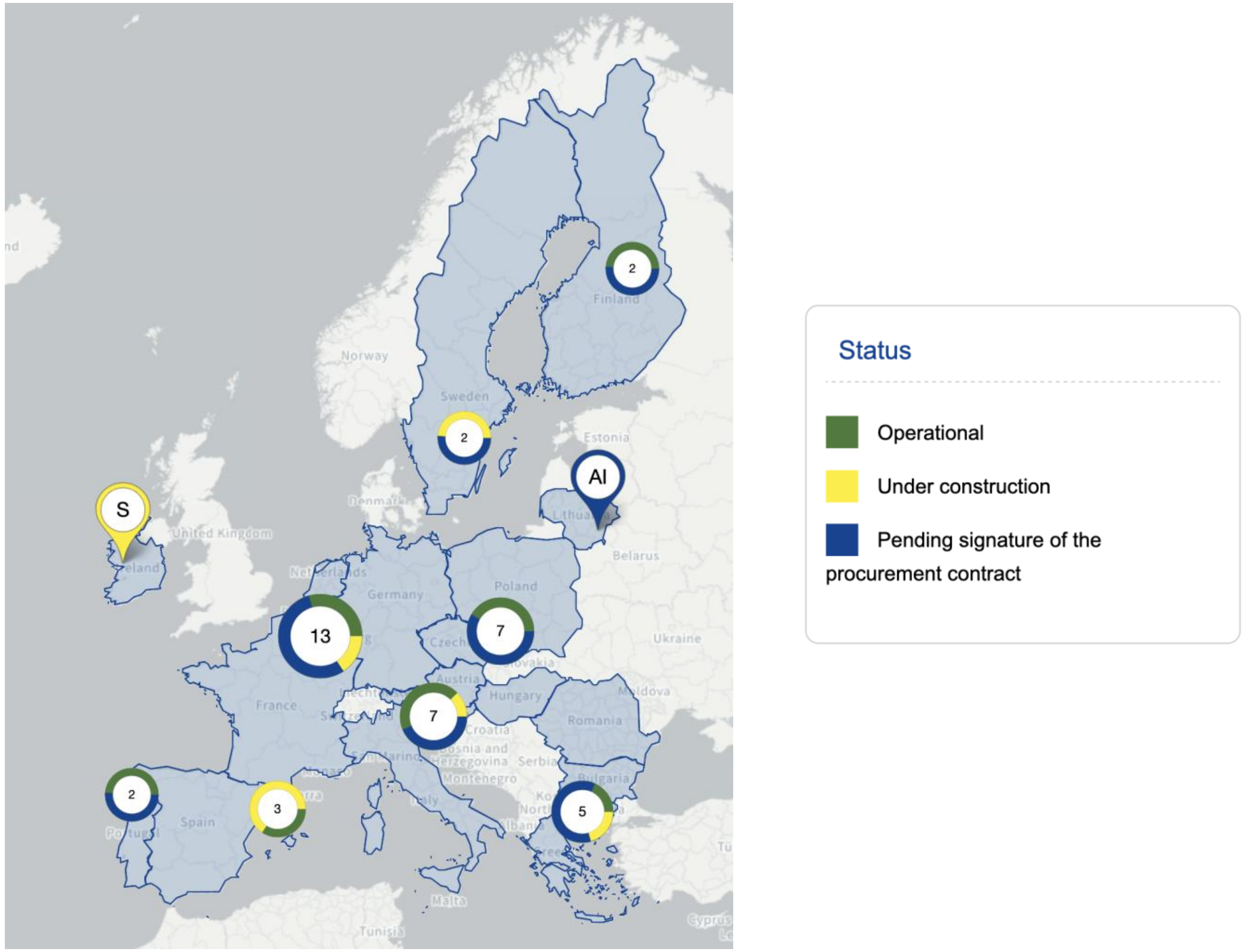
Destination Earth – Digital Twins (DT)

- Providing detailed high-resolution information from global to local scale
- Providing integrated Earth-system and impact sector information
- Providing interactive access to models, data and workflows, based on the DTE

Weather-induced Extremes DT	Climate Change Adaptation DT
• Init conditions from oper IFS • Approx 4.5 km resolution • 4 days deterministic forecast • Automatic daily cycling • Runs on LUMI EuroHPC	• IFS- and ICON-based • 5-10km resolutions • Simulations cover 1990-2050 • One shot simulations • Runs on LUMI/Marenostrum

Destination Earth – EuroHPC (The European HPC Joint Undertaking)

- Legal and funding entity started in 2018
- Support and the Digital transformation of EU
- Support for public and private entities
- Eleven HPC online and more to come
- AI factories on their way



https://www.eurohpc-ju.europa.eu/index_en

ECMWF

EuroHPC Joint Undertaking

Destination Earth – EuroHPC

LUMI, Finland (No 9 TOP500)

LEONARDO, Italy (No 10 TOP500)

MareNostrum, Spain (No 8 TOP500)

Destination Earth – EuroHPC

LUMI, Finland (No 9 TOP500)

LEONARDO, Italy (No 10 TOP500)

MareNostrum, Spain (No 8 TOP500)

	LUMI	LEONARDO	MARENOSTRUM
Performance (Petaflops)	386	249	215
GPU Partition	AMD Instinct	"Da Vinci" GPUs	NVIDIA Hopper
Storage (HDD/SSD PB)	110 / 7	100 / 5	248
Nodes' network specificities		Compute (no Internet)	Compute/login (no Internet)
Authentication	SSH keys	2FA	SSH keys

ECMWF

EuroHPC
Joint Undertaking

Destination Earth – EuroHPC

JUPITER, Germany (No 4 TOP500)

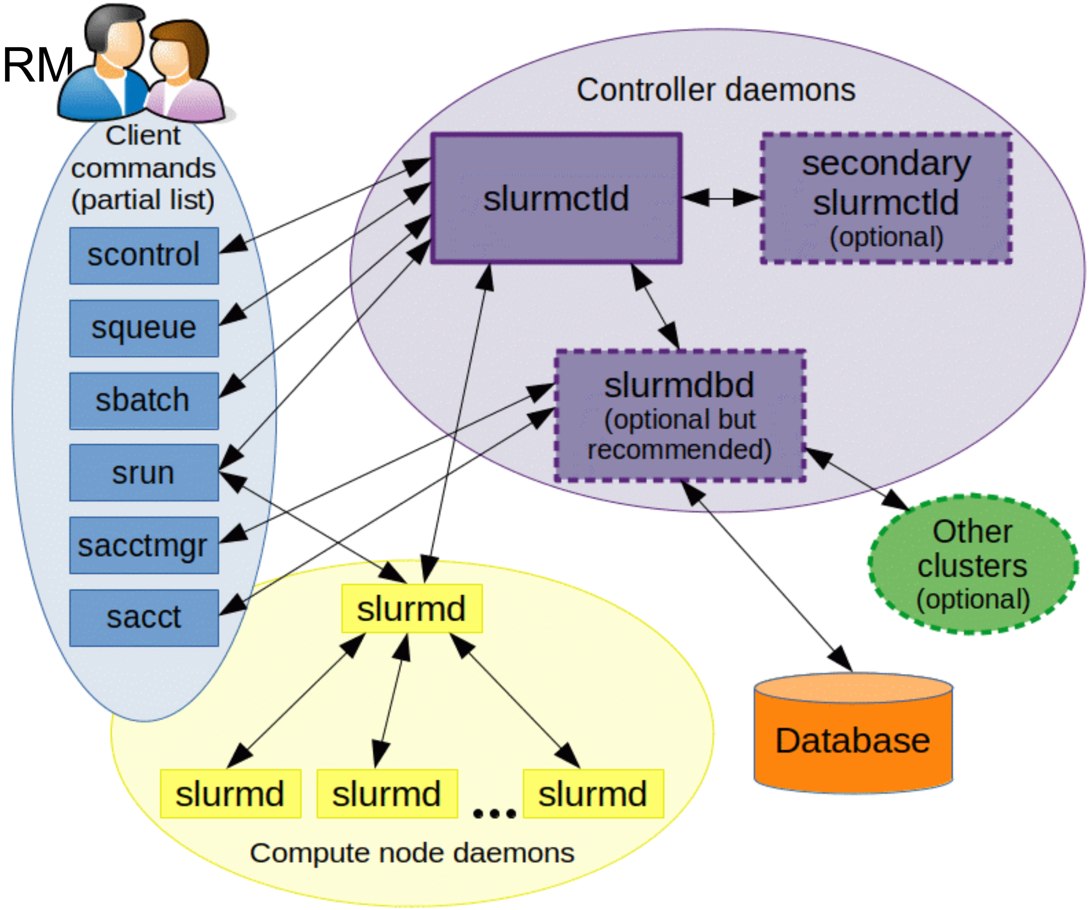
	JUPITER
Performance (Petaflops)	1000 ==> Exascale
GPU Partition	NVIDIA Grace Hoppers
Storage (HDD/SSD PB)	… / 20
Nodes’ network specificities	Compute (No Internet)
Authentication	2FA

ECMWF

EuroHPC Joint Undertaking

Destination Earth – How to use EuroHPC resources

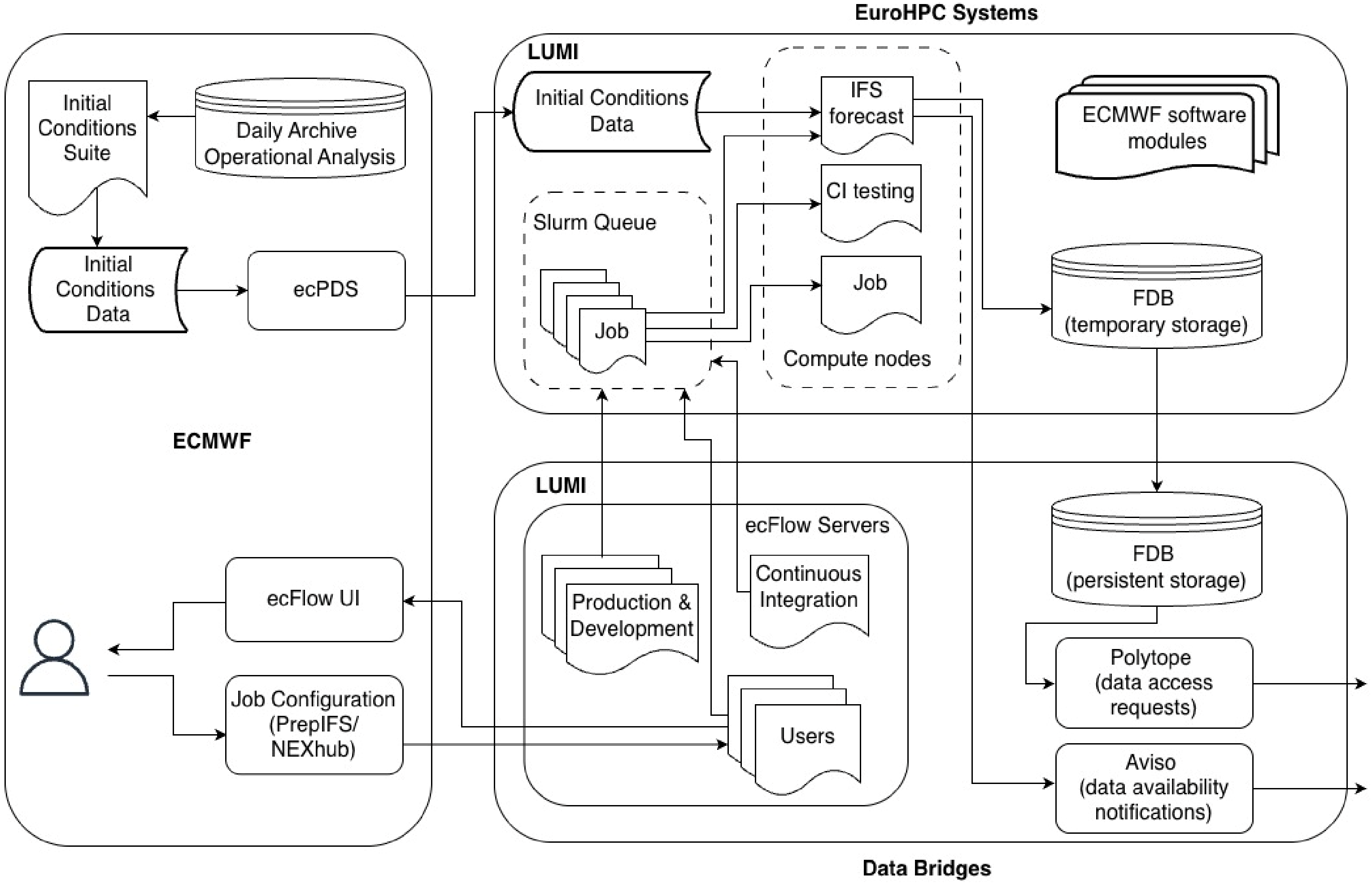
- Log into a EuroHPC
- Most EuroHPC resources are managed by SLURM
  - Job scheduler
  - Job monitoring
  - Job queueing system
- Single user use sbatch to submit jobs
- Possibility to build complex workflow from outside the HPC by complying to security policies



Destination Earth – External workflow example using SLURM

The **Global Weather-Induced Extremes DT** is operated by ECMWF on the **LUMI Supercomputer** in Finland.

- **Daily** forecasts at **4.4 km**
- Data made available on **remote FDB** on **Data Bridges**
- ECMWF **software modules installation** on **remote HPC**
- ecFlow servers hosted on the Data Bridges for **production** and **user experiments**
- **Secure connections** to remote HPC via ssh



ECMWF

Connecting ECMWF Workflows to the Future: Leveraging Distributed Pre-Exascale Resources

ECMWF

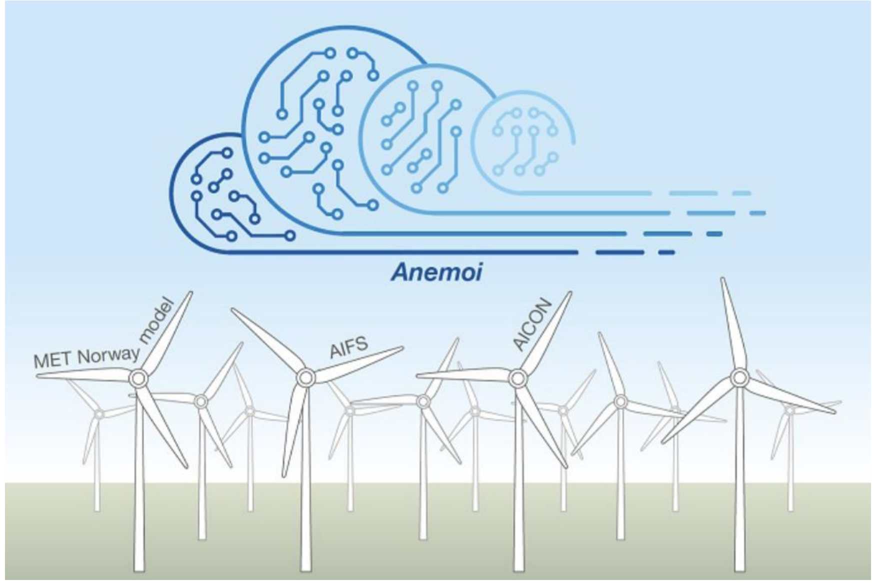
Johannes Bulin*, Cristina Duma†, Matthew Griffith§, Ulrike Hager*, Bentorey Hernandez-Cruz*, Gareth Jones*, Emma Kuwertz§, Julien Lefaucheur§, Giuseppe Misurelli†

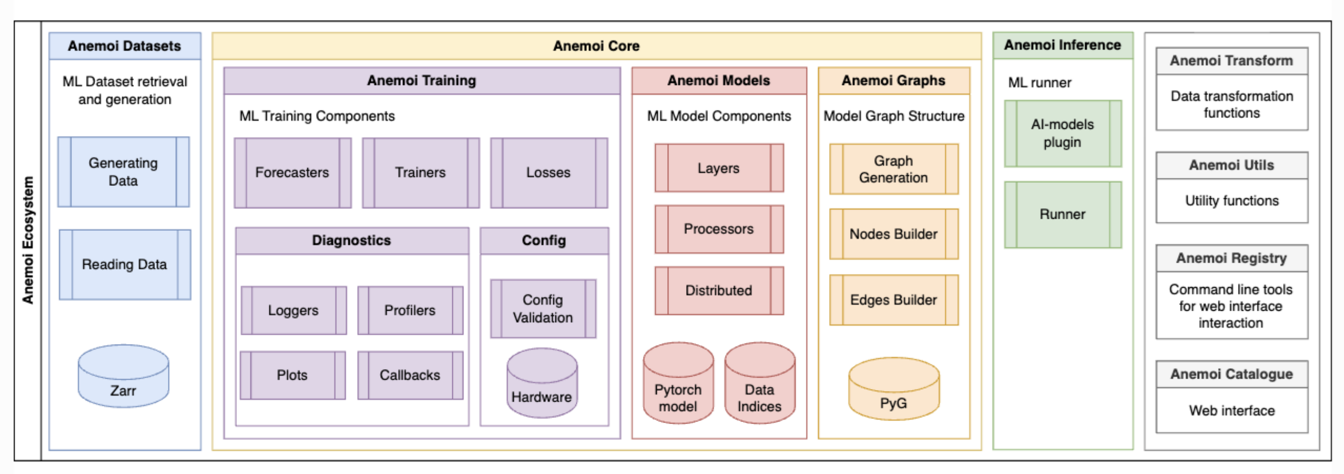
Outline

1. **Data Handling**
   - Introduction
   - Data Flow
   - Access Patterns & Datasets
2. **DestinE & Infrastructure**
   - EuroHPC
   - ML applications
     - *Datasets transfer across EuroHPC*
     - *Orchestration of training workflows prototype across EuroHPC*

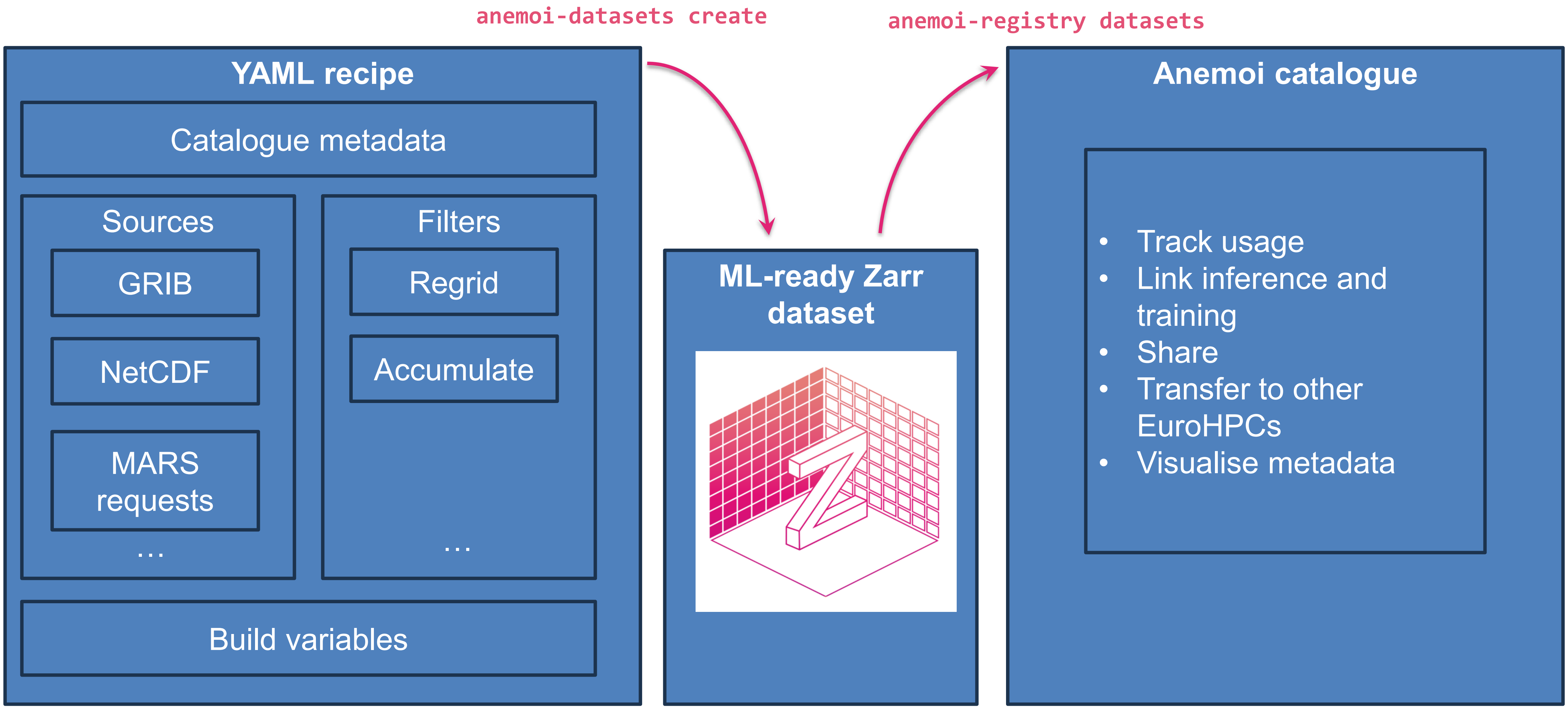
Anemoi Ecosystem Overview

- Anemoi is an open-source framework that provides a complete toolkit to develop data-driven weather models – from data preparation through to inference
- It's a highly modular framework organised into different Python packages
- The framework builds upon on established Python tool including PyTorch, PyTorch Lighting, Hydra, Pydantic, and earthkit.



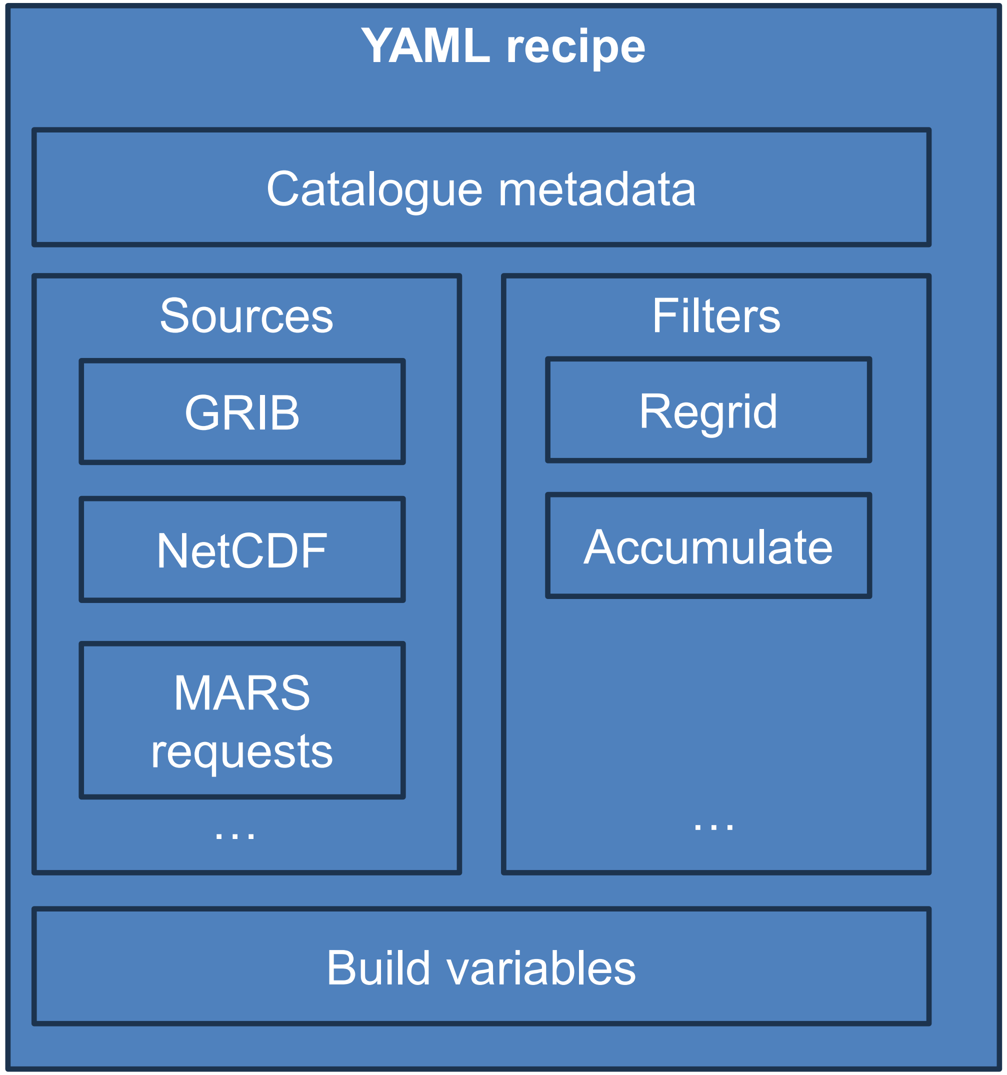


Anemoi dataset creation overview



Anemoi dataset creation

Step 1. Datasets are defined as YAML “recipes”




```yaml
name: aifs-ea-an-oper-0001-mars-o96-2026-2026-6h-v1-destine
description: Example dataset for DestinE presentation.
attribution: ECMWF
licence: CC-BY-4.0

dates:
  end: '2023-12-31T18:00:00'
  frequency: 6h
  start: '1979-01-01T00:00:00'

input:
  join:
  - mars:
      class: ea
      expver: '0001'
      grid: o96
      levtype: sfc
      param:[10u, 10v, 2d, 2t]
  - accumulate:
      class: ea
      expver: '0001'
      grid: o96
      param: [cp, tp]

build:
   group_by: monthly
```

Anemoi dataset creation

Step 2. Datasets are built as ML-ready zarr datasets with:

anemoi-datasets create <recipe.yaml>

Machine-learning-ready Zarr dataset

- Numerical data contained in lots of small files: one for each chunk
- Separate arrays for longitudes, latitudes, dates and statistics
- Separate files containing zarr metadata (.zarray, .zgroup, & .zattrs)

```
aifs-ea-an-oper-0001-mars-o96-2026-2026-6h-v1-destine.zarr/
├── data
│   ├── 0.0.0.0
│   ├── 1.0.0.0
…
│   ├── 98.0.0.0
│   ├── 99.0.0.0
│   ├── .zarray
│   └── .zattrs
├── dates
│   ├── 0
│   ├── .zarray
│   └── .zattrs
├── latitudes
│   ├── 0
…
│   ├── 7
│   ├── .zarray
│   └── .zattrs
…
├── stdev
│   ├── 0
│   ├── .zarray
│   └── .zattrs
├── .zattrs
└── .zgroup
```

Anemoi Catalogue

Step 3. Datasets are registered to the anemoi catalogue with :
`anemoi-registry datasets <dataset.zarr>`

anemoi.ecmwf.int

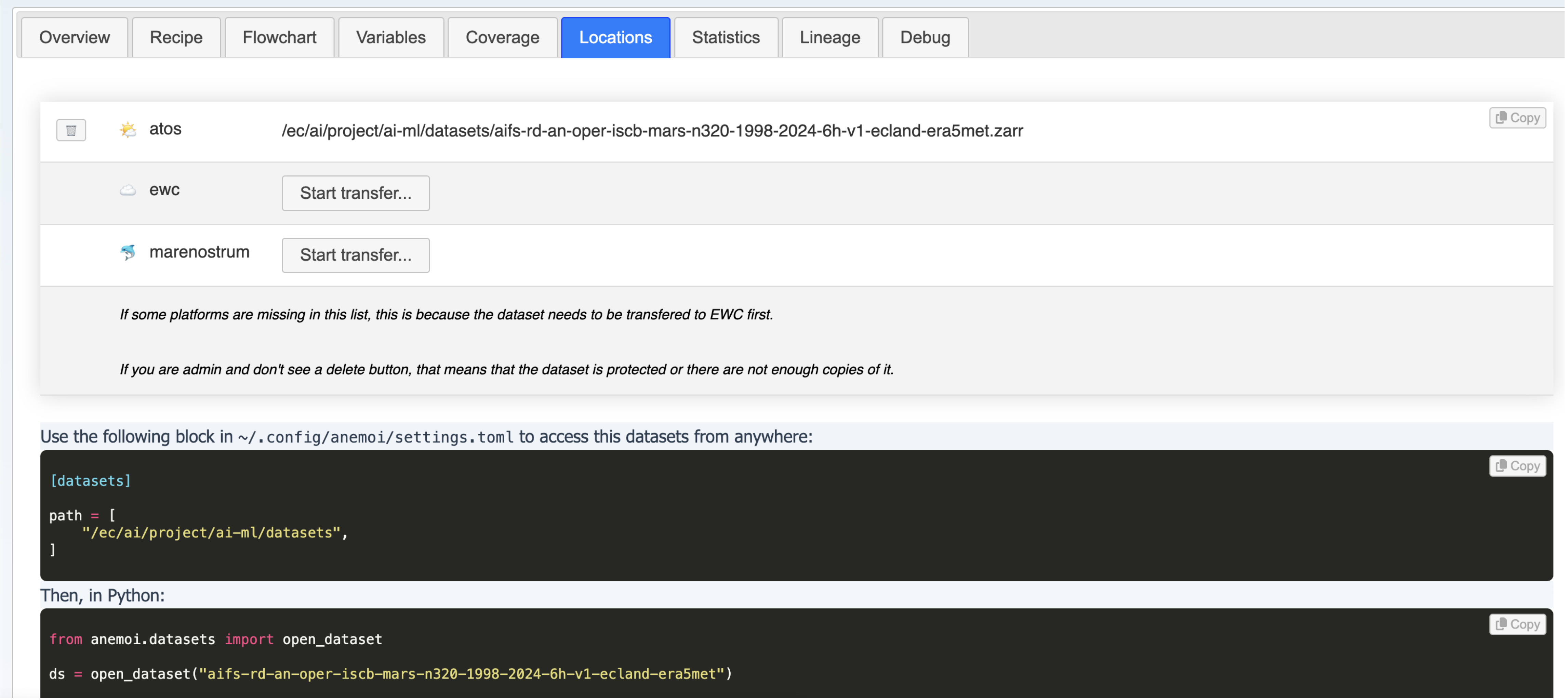
Datasets Trainings Checkpoints Models Experiments Tasks Chatbot PrepML Natalie Zelenka

Show 20 entries Search: single

Name	Size	Files	Resol.	Freq.	Rows	Var.	Ens.	Values	Created	Where	Usage	Mentioned
aifs-ea-an-oper-0001-mars-n320-1979-2024-6h-v1-for-single-v2	10 TiB	67,339	N320	6h	67,208	149	1	542,080	2025-08-08		363	2025-11-17
aifs-od-an-oper-0001-mars-n320-2016-2025-6h-v1-for-single-v2	2 TiB	14,130	N320	6h	14,000	149	1	542,080	2025-08-19		191	2025-11-04
aifs-ea-an-oper-0001-mars-o96-1979-2024-6h-v1-aifs-single-v1	670 GiB	67,255	O96	6h	67,208	115	1	40,320	2025-04-02		88	2025-11-17
aifs-rd-an-oper-ioku-mars-n320-2024-2024-6h-v2	79 GiB	643	N320	6h	520	149	1	542,080	2025-09-24		21	2025-11-04
aifs-ea-an-oper-0001-mars-o96-1979-2024-6h-v1-for-single-v2	866 GiB	67,327	O96	6h	67,208	149	1	40,320	2025-08-29		12	2025-11-17
aifs-rd-an-oper-isgx-mars-n320-2024-2025-6h-v2	62 GiB	525	N320	6h	402	149	1	542,080	2025-10-22		6	2025-11-04
aifs-rd-an-oper-ioku-mars-n320-2024-2024-6h-v1	79 GiB	643	N320	6h	520	149	1	542,080	2025-09-13		4	2025-09-16

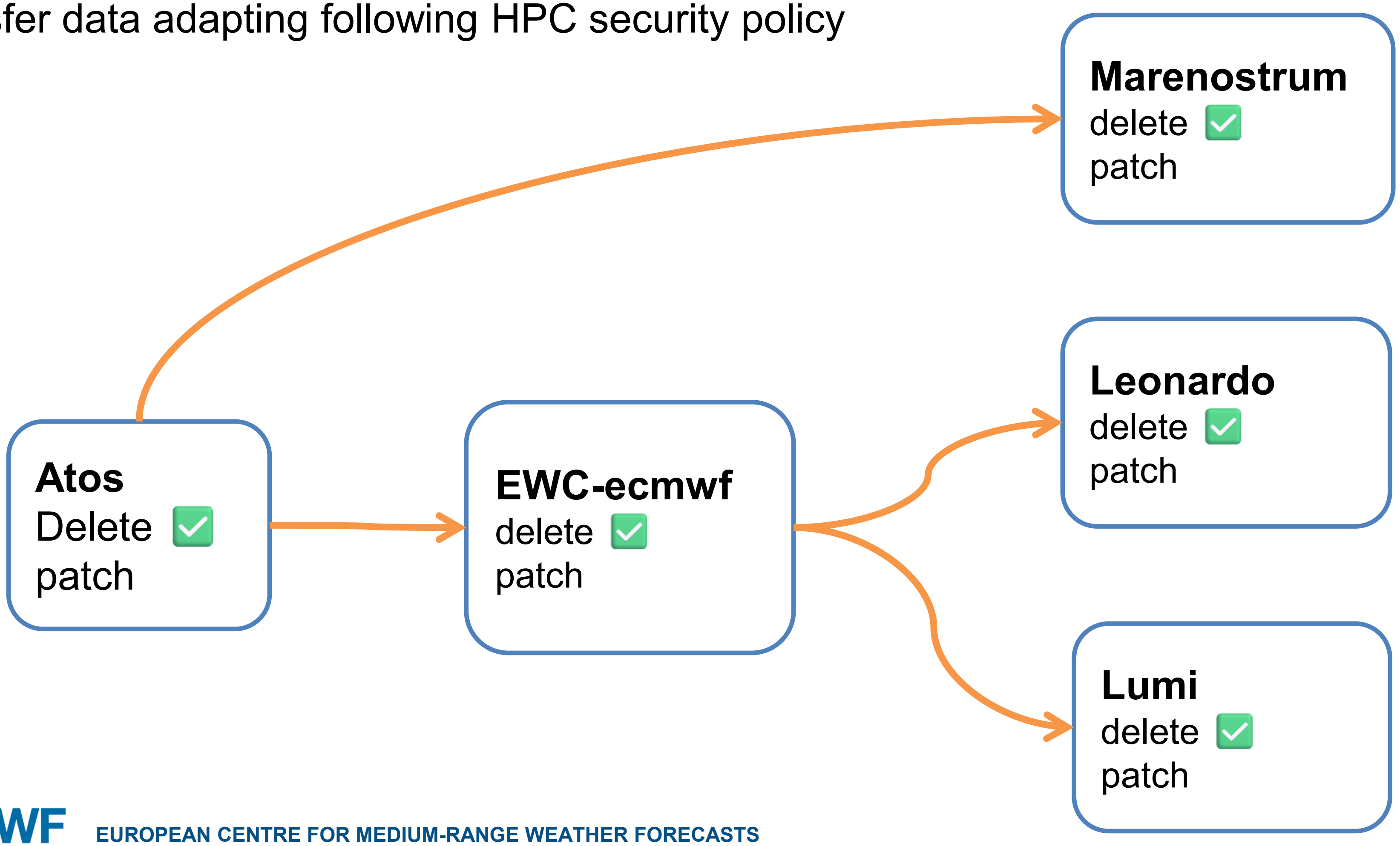
Anemoi Catalogue

Step 4. On the “locations” tab, datasets can be transferred to new locations - or used from their current locations.



Dataset transfer workflow

- Datasets are built on ECMWF HPC
- Datasets are uploaded on the European Weather Cloud (EWC)
- To transfer data adapting following HPC security policy

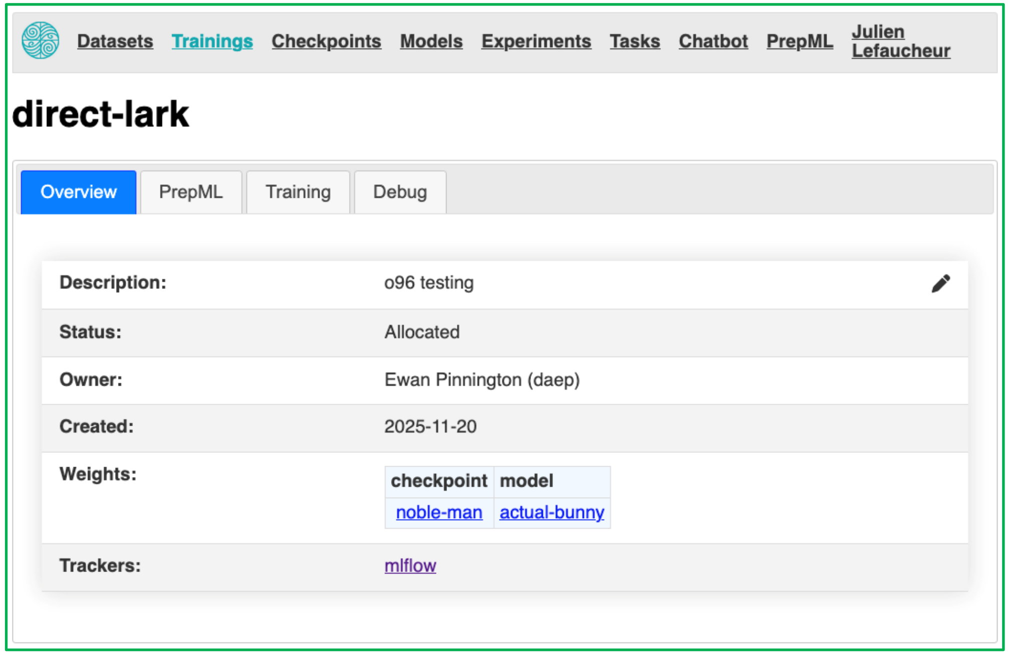


Outline

- Data Flow
- Access Patterns & Datasets
- EuroHPC
- DestinE & ML applications
  - Datasets transfer across EuroHPC
  - Orchestration of training workflows prototype across EuroHPC

Training workflow orchestration prototype across EuroHPC

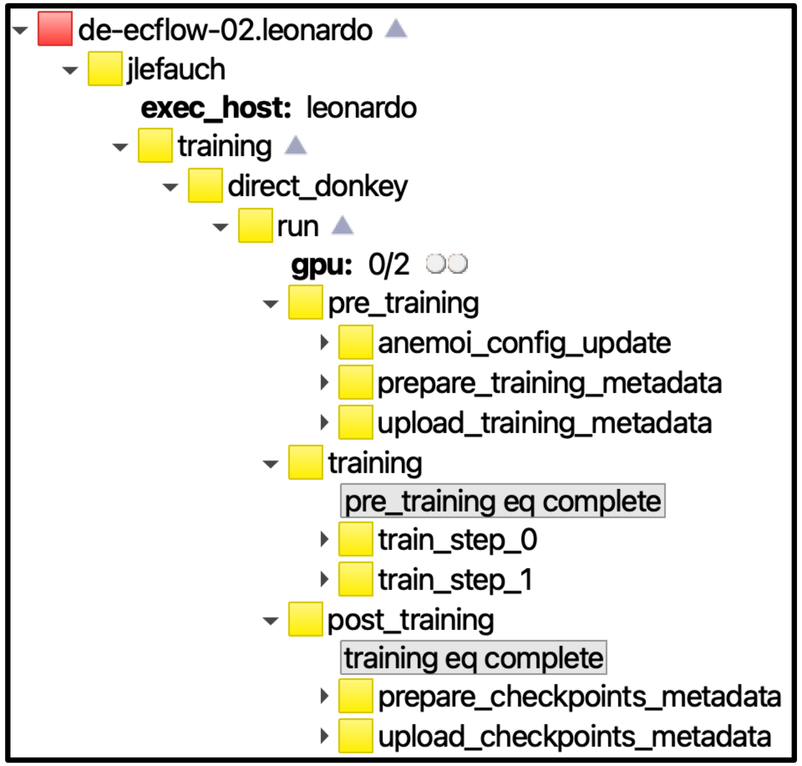
- Why orchestrate a training workflow?
  - **Seamlessly scale** training workflows **across distributed HPC systems**
  - Tap into pre-exascale systems through **familiar intuitive tools**
  - **Monitor and orchestrate** every component no matter **where it runs**
  - **Track** full **lineage** of ML artefacts
- Use cases covered
  - Research experiments
  - Operational model production
- How to use it
  - Python CLI driven by a configuration file
  - Wraps Anemoi tools (training, registry)
  - CLI to build and deploy an **ecflow workflow**



Training full lineage in the anemoi catalogue

Training workflow orchestration prototype across EuroHPC

- Why orchestrate a training workflow?
  - **Seamlessly scale** training workflows **across distributed HPC systems**
  - Tap into pre-exascale systems through **familiar intuitive tools**
  - **Monitor and orchestrate** every component no matter **where it runs**
  - **Track** full **lineage** of ML artefacts
- Use cases covered
  - Research experiments
  - Operational model production
- How to use it
  - Python CLI driven by a configuration file
  - Wraps Anemoi tools (training, registry)
  - CLI to build and deploy an **ecflow workflow**



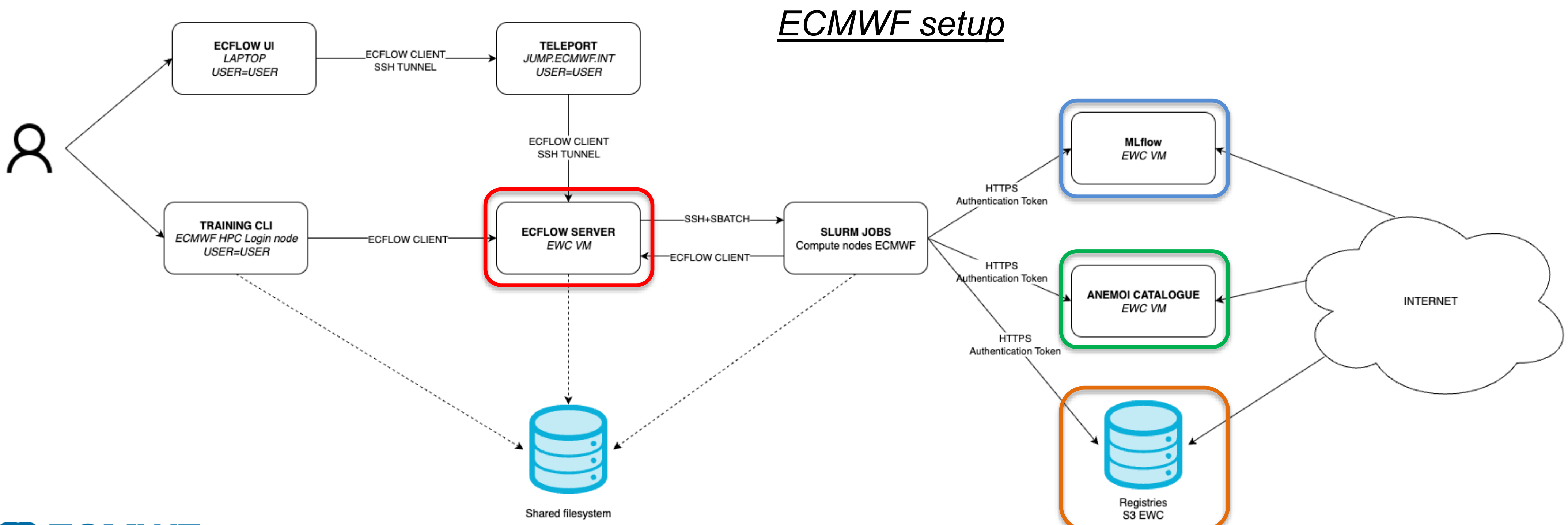
Example of ecflow training workflow running on LEONARDO EuroHPC

Training workflow orchestration prototype across EuroHPC

- Components & Workflow
  - Define workflows and schedule jobs, **ecflow**
  - Log training metrics, **Mlflow**
  - Lineage tracking (datasets, trainings, weights, evaluation), **Anemoi Catalogue**
  - Store ML artefacts, **S3 registries**

Training workflow orchestration prototype across EuroHPC

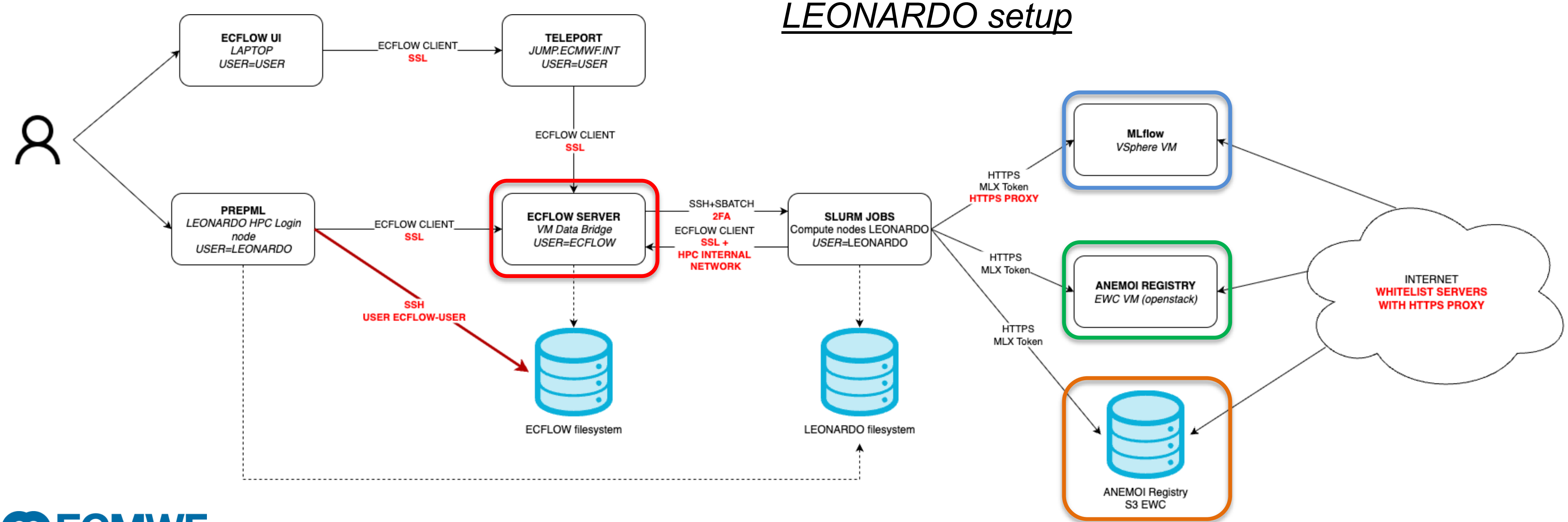
- Components & Workflow
  - Define workflows and schedule jobs, **ecflow**
  - Log training metrics, **Mlflow**
  - Lineage tracking (datasets, trainings, weights, evaluation), **Anemoi Catalogue**
  - Store ML artefacts, **S3 registries**



Training workflow orchestration prototype across EuroHPC

- Components & Workflow
  - Define workflows and schedule jobs, **ecflow**
  - Log training metrics, **Mlflow**
  - Lineage tracking (datasets, trainings, weights, evaluation), **Anemoi Catalogue**
  - Store ML artefacts, **S3 registries**

LEONARDO setup



Summary

- Dedicated HPC infrastructure for machine learning
- Need to adapt to
  - Different security policies
  - Hardware specificities
- Solution to transfer datasets across different EuroHPC for ML scientists
- Prototype to make use of resources across different EuroHPC

Outlook

https://destine.ecmwf.int/news/phase-three-of-destination-earth-confirmed/