



# interTwin

## An Interdisciplinary Digital Twin Engine for Science

CERN-IT: Matteo Bunino, Alexander Zoechbauer, Kalliopi Tsolaki, Ilaria Luise, Maria Girone, Sofia Vallecorsa, Alberto Di Meglio

JSC: Rakesh Sarma, Mario Ruettgers, Andreas Lintermann

interTwin consortium



Funded by the  
European Union

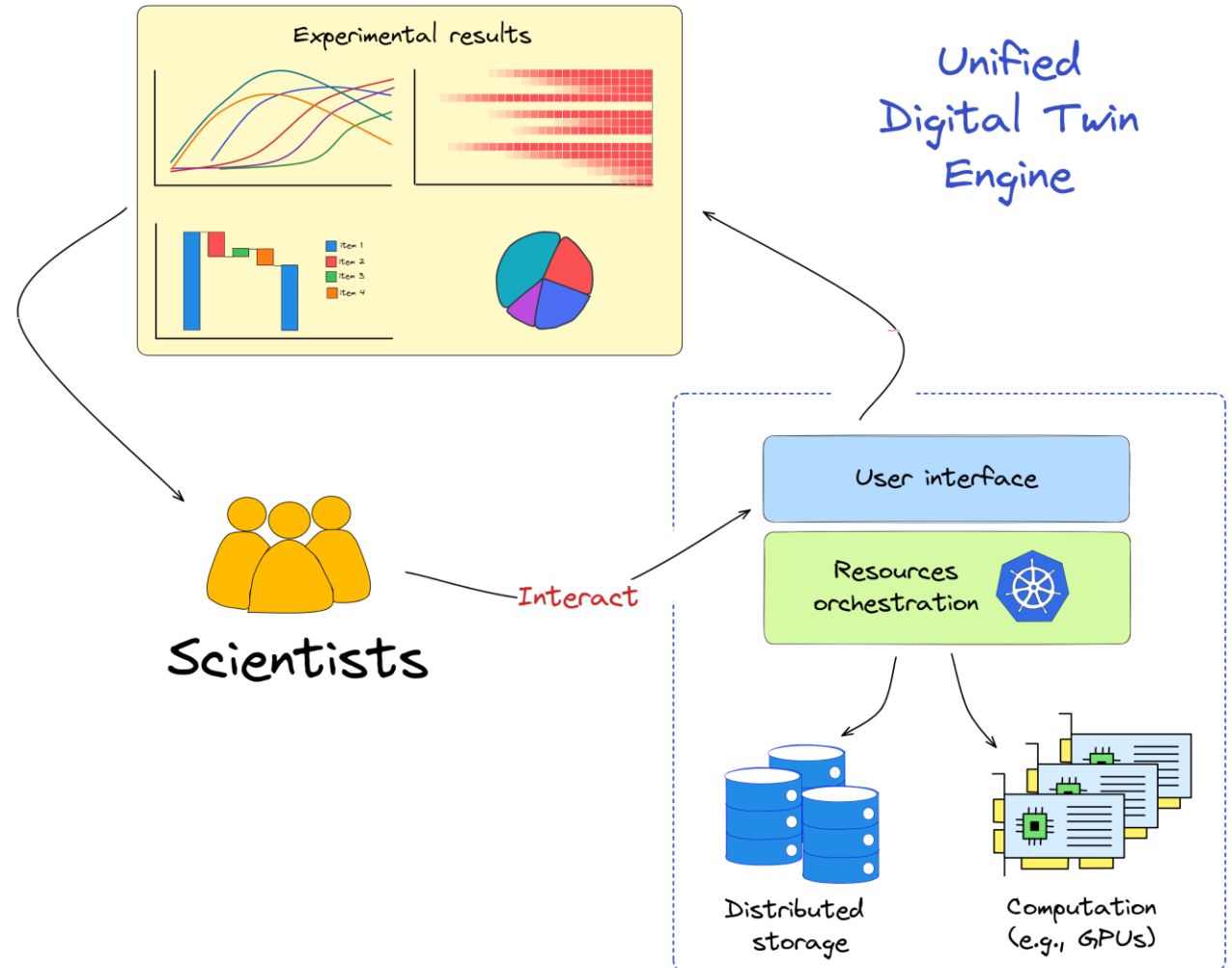
The interTwin project is funded by the European Union - Grant Agreement Number 101058386

# interTwin: a Digital Twin Engine for science

Co-design and implement the prototype of an **interdisciplinary Digital Twin Engine**

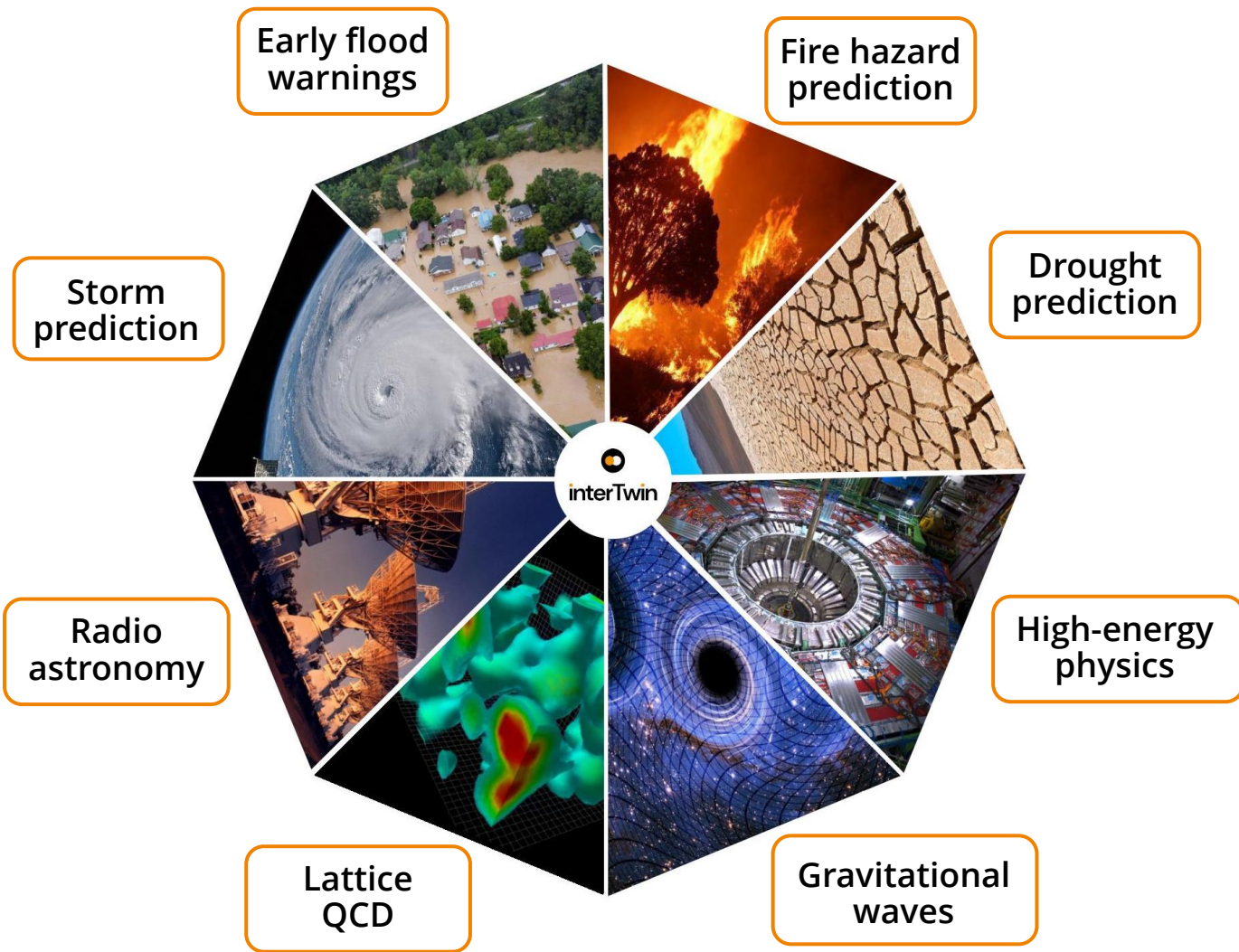
**Open-source** platform based on **open standards**

Large spectrum of **diverse use cases** from **physics** and **earth observation** sciences





# interTwin: a Digital Twin Engine for science

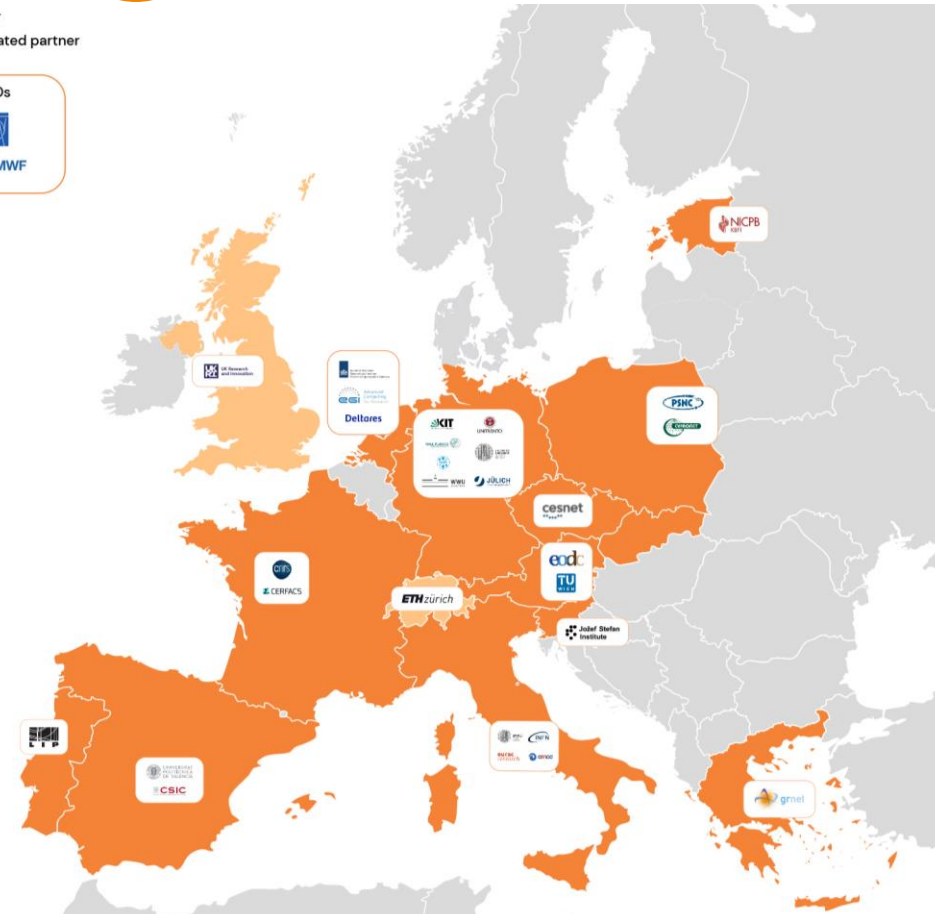


29

Participants, including sciences, technology and resource providers.

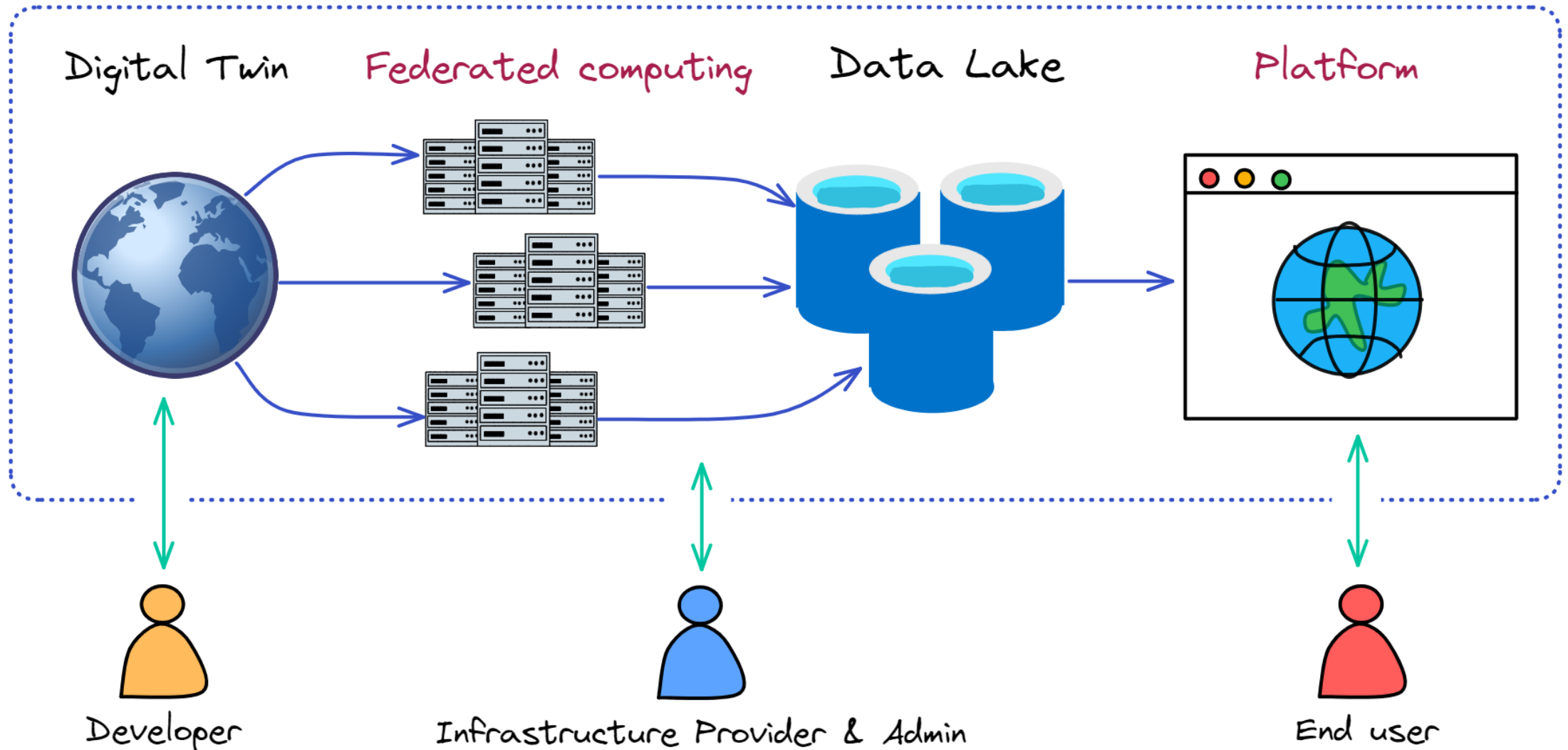
Partner  
Associated partner

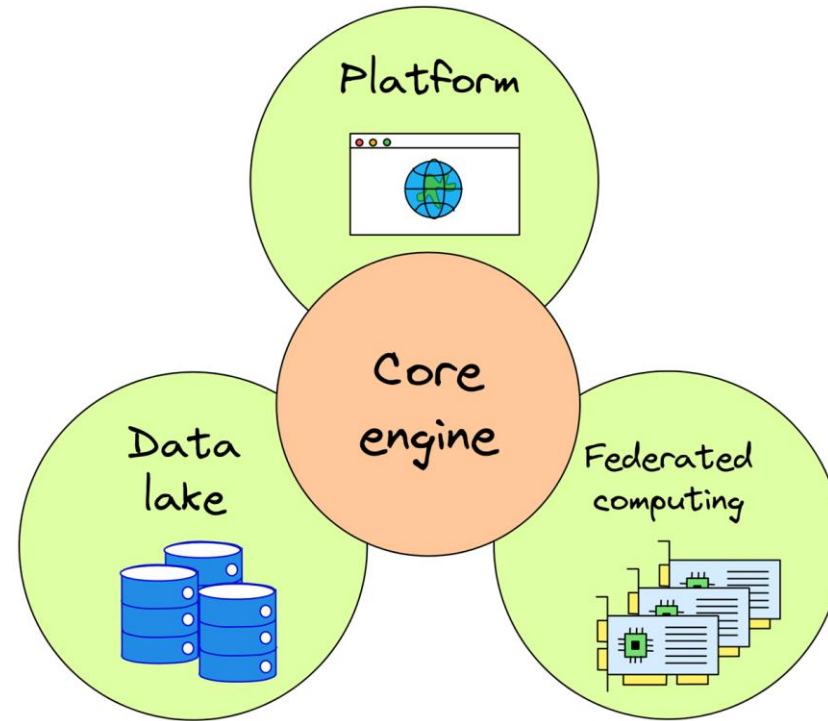
EIROs  
ECMWF

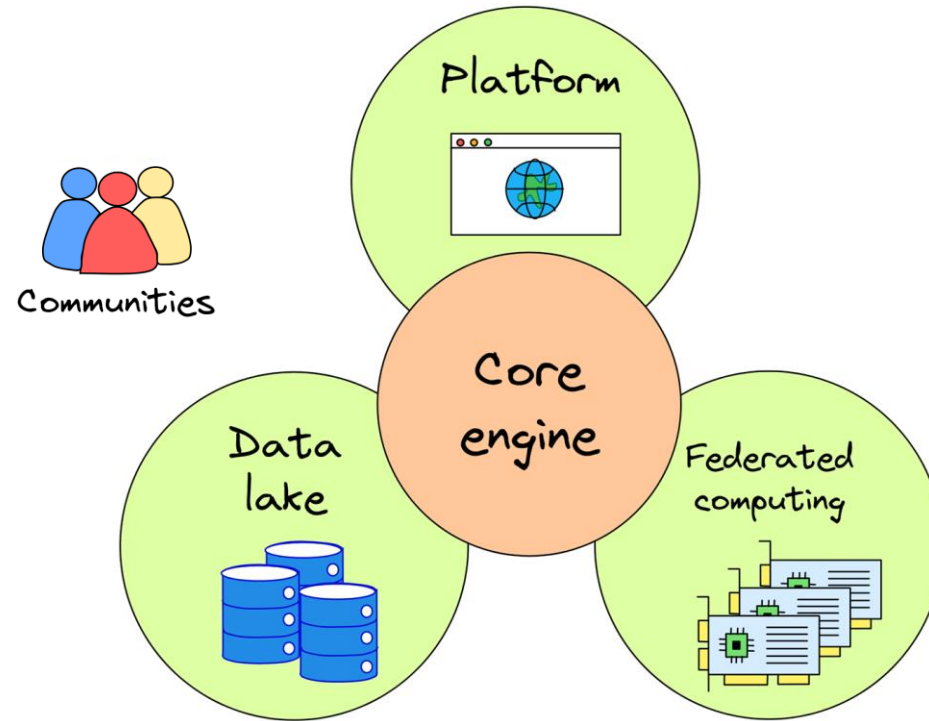


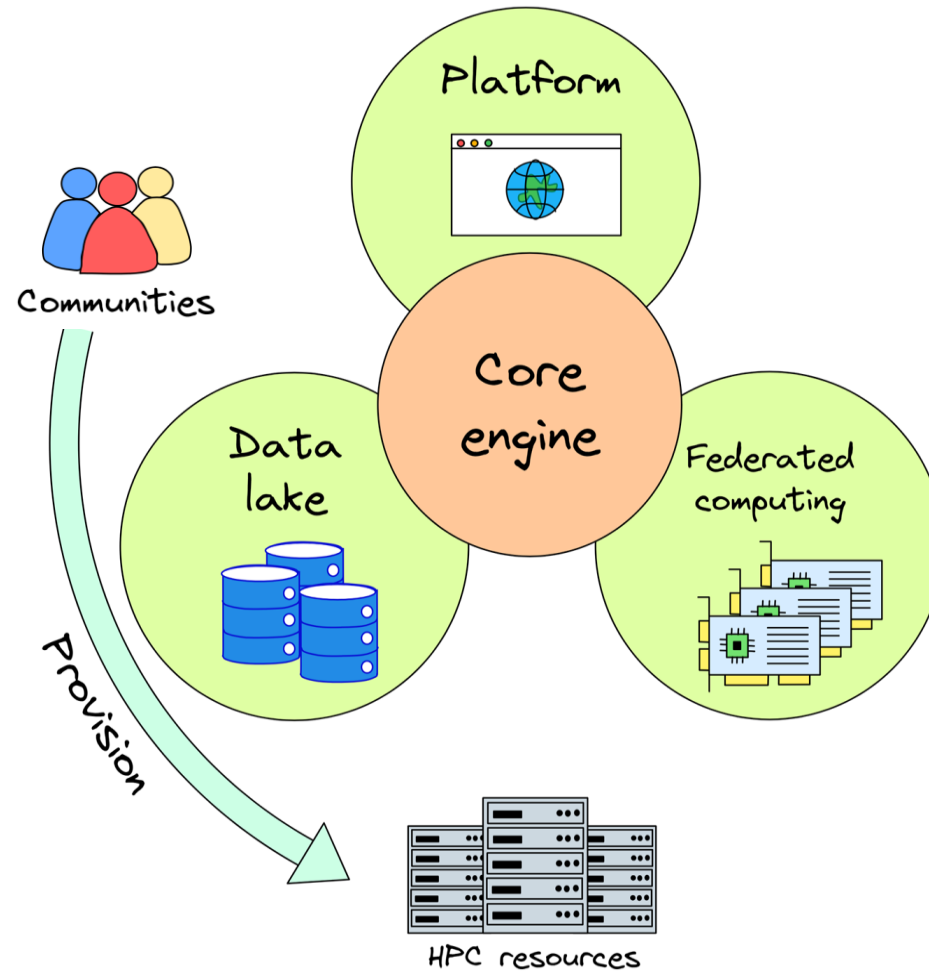


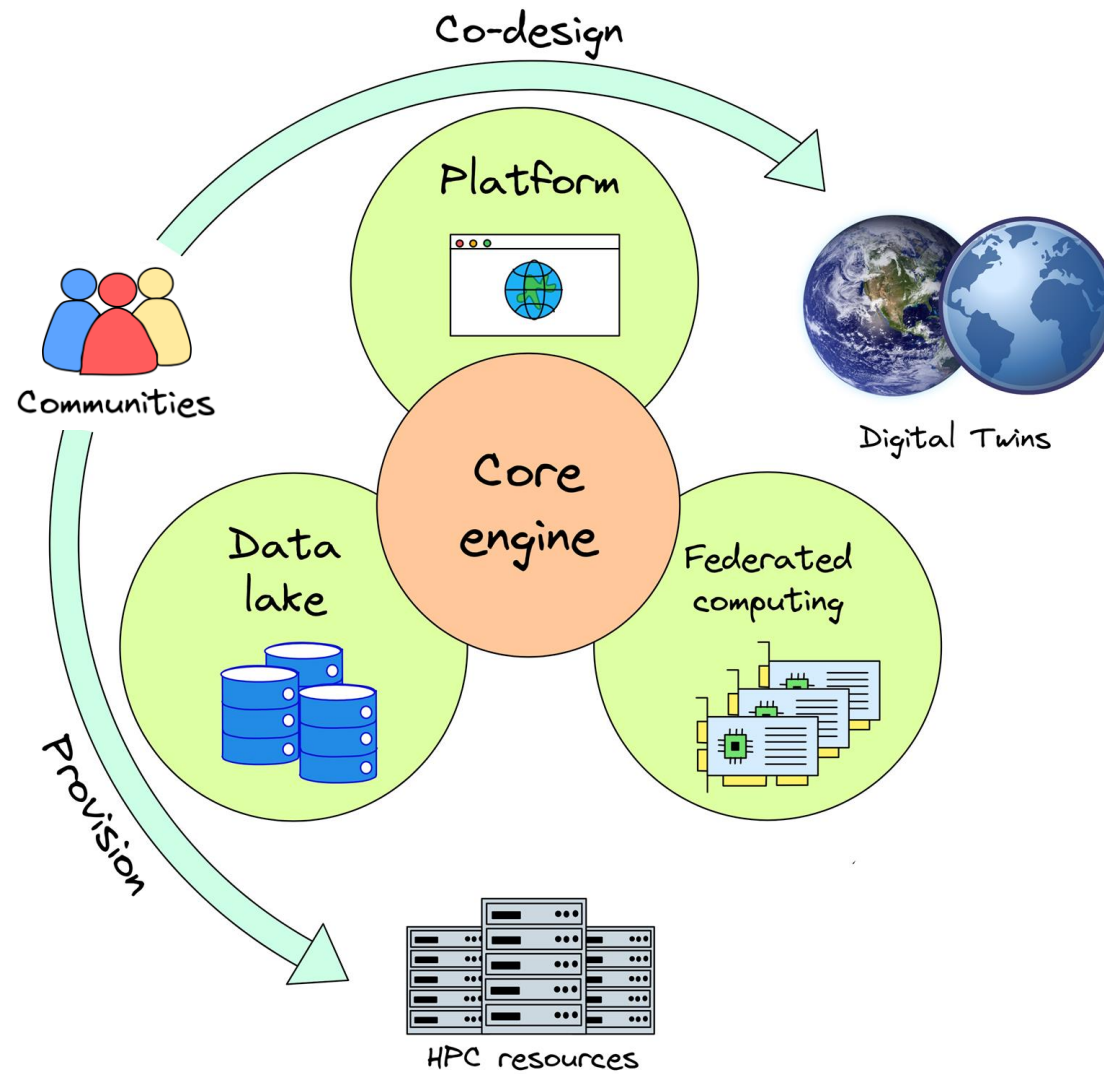
# interTwin: a Digital Twin Engine for science



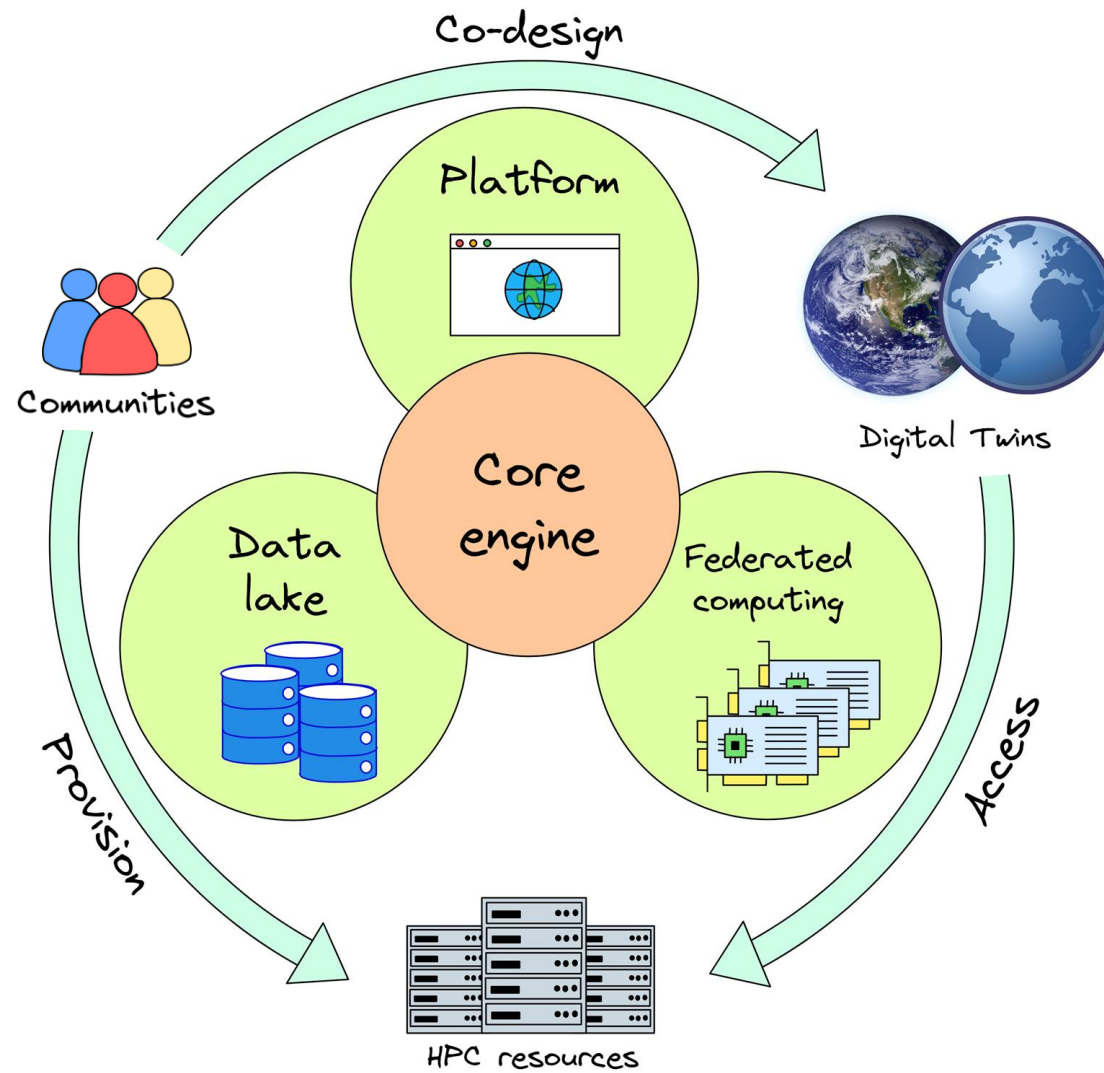














# Climate research and monitoring



## **Climate Change Future Projections of Extreme Events (storms & fire)**

CMCC, CNRS (IPSL), UNITN

## **Climate Change Impacts of Extreme Events (storms, fire, floods, drought)**

CERFACS, Deltares, EURAC

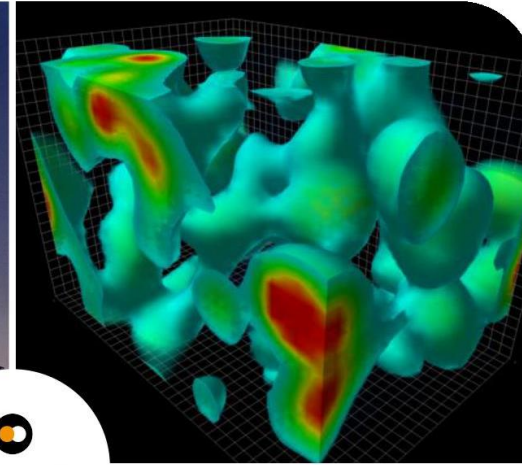
## **Early Warning for Extreme Events (floods & droughts)**

Deltares, EURAC, TU Wien



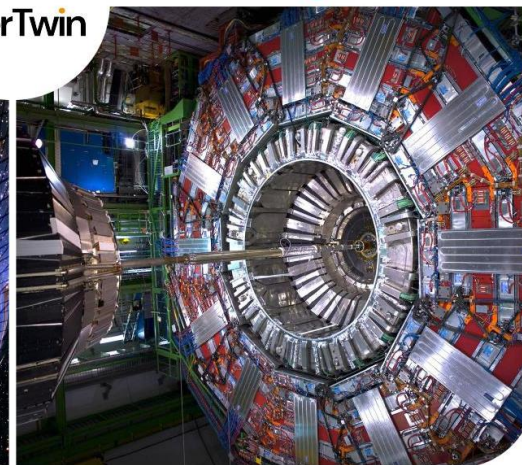
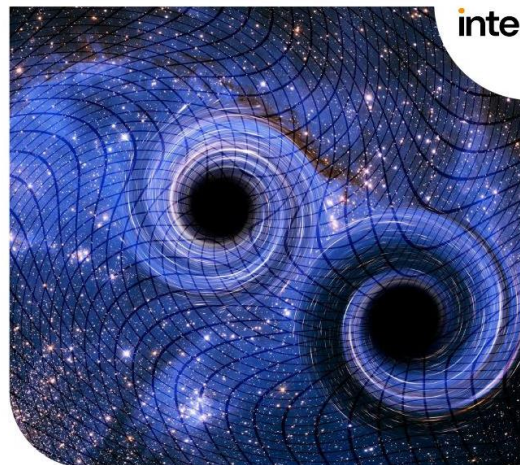
# Physics research

**Noise Simulation for  
Radio Astronomy**  
Univ. of Heidelberg,  
Max Planck Society



**Lattice QCD Simulations  
for High Energy Physics**  
ETHZ, CSIC

**VIRGO Noise Detector for  
Astrophysics**  
INFN



**Detector Simulation for High  
Energy Physics**  
CERN, CNRS (IN2P3-IRES)





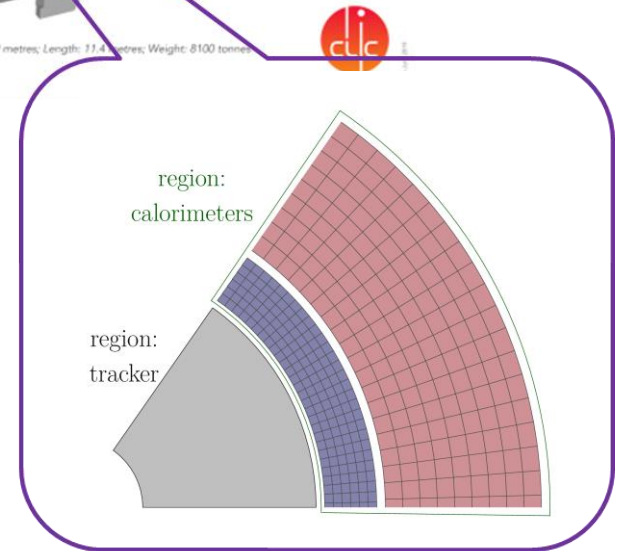
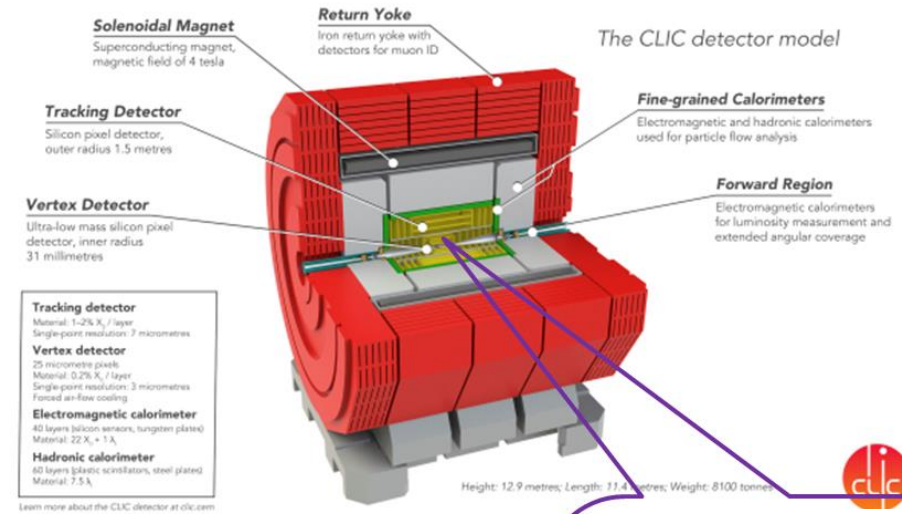
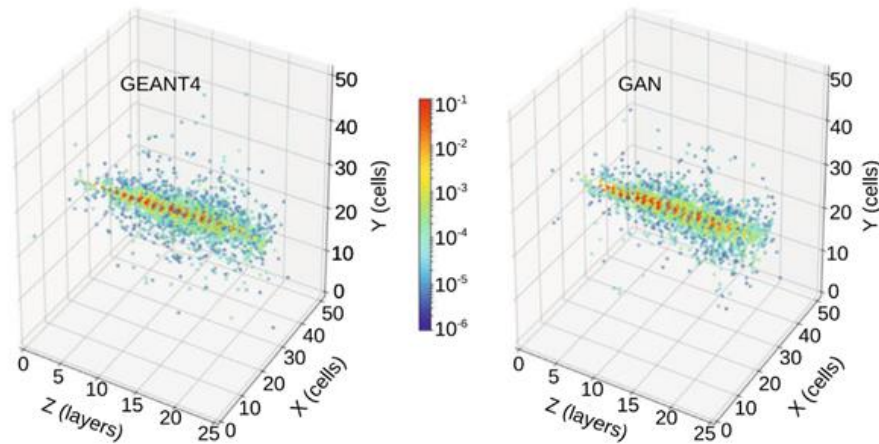
# HEP Digital Twins of particle detectors

## Detector Prototyping & Optimization

Build data-driven tool that **simulates detector response** and integrates operation conditions from experimental setups (test-beams).

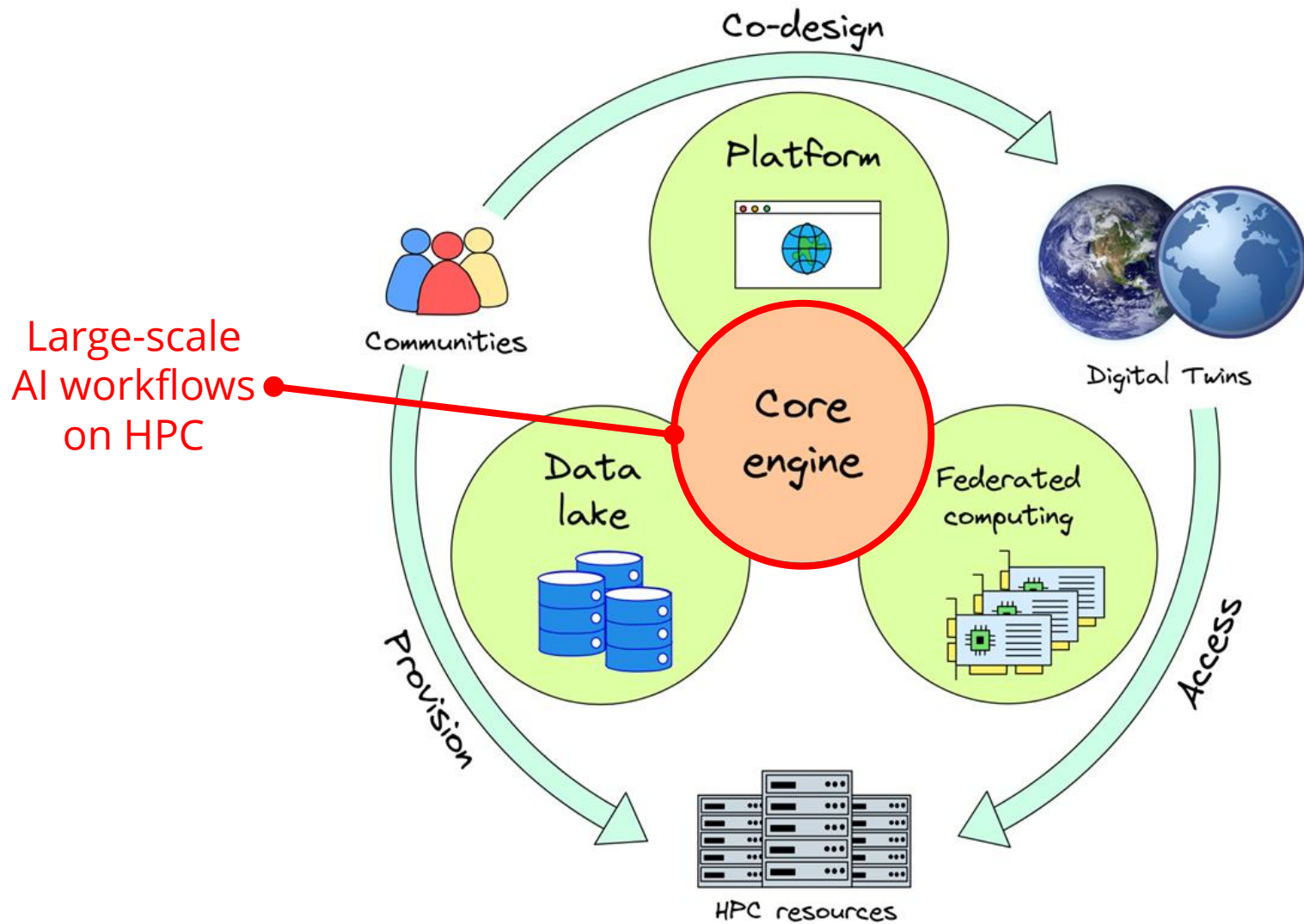
## Online ML for Detectors

Adapt **real-time** detector and/or data acquisition configuration with respect to run conditions



**Fast Simulation of a High Granularity Calorimeter by Generative Adversarial Networks.**

Gul Rukh Khattak et al. <https://arxiv.org/abs/2109.07388> DOI: <https://doi.org/10.48550/arXiv.2109.07388>



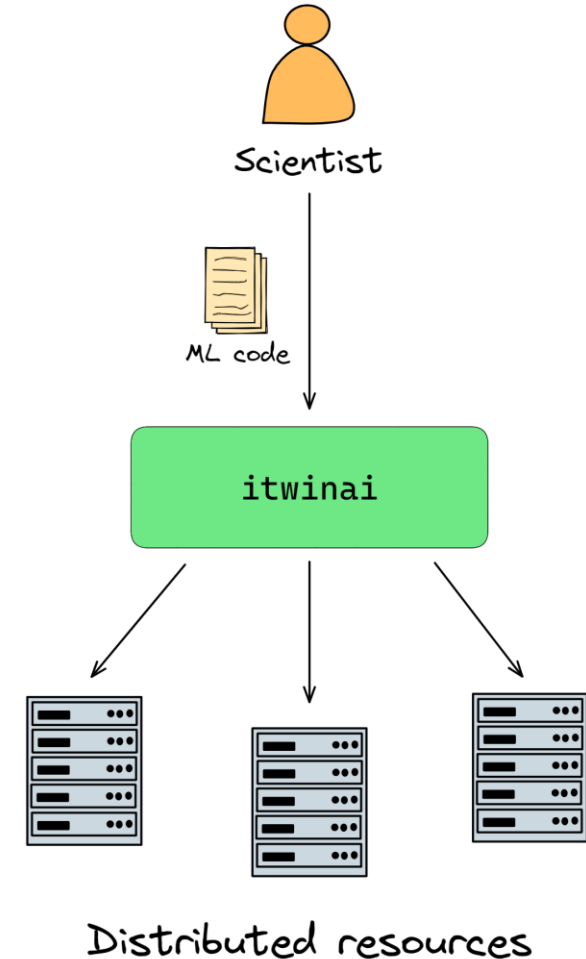
# DT core engine – AI features



# Our contribution to the Core Engine

Support AI-based digital twin applications in science:

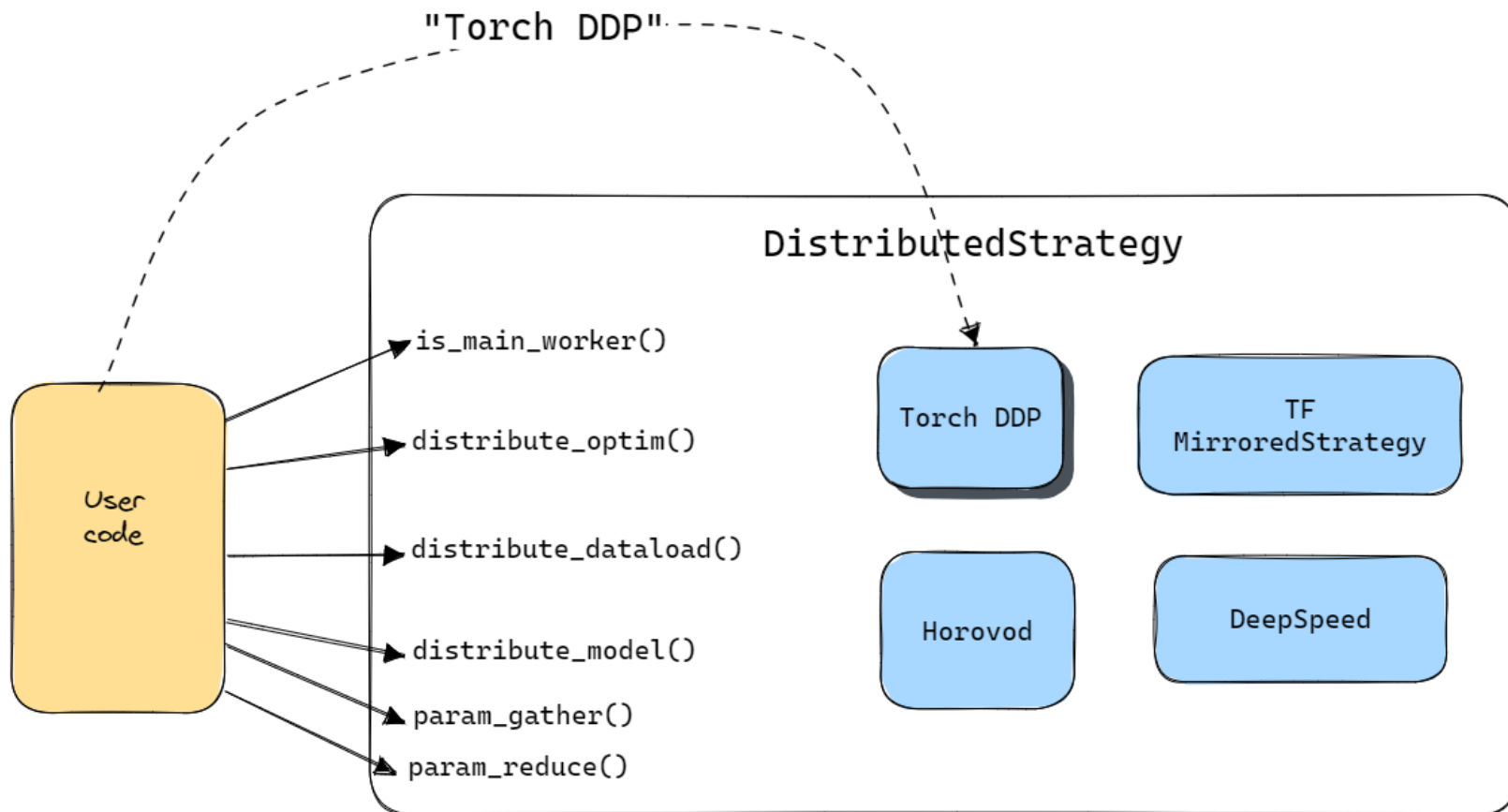
- **Reproducibility, Reusability, and Modularity**
- **Framework-independent** (e.g., PyTorch, TensorFlow, MLFlow, WandB)
- **UX/UI**: user-friendly GUI (e.g., JupyterLab)
- Off-the-shelf AI tooling, reducing engineering overheads:
  - **Hyper-parameters optimization**
  - Scalability (e.g., **distributed ML**)
  - State of the Art **models repository**
- **Seamless access to infrastructure** (cloud and HPC)





# Our contribution to the Core Engine

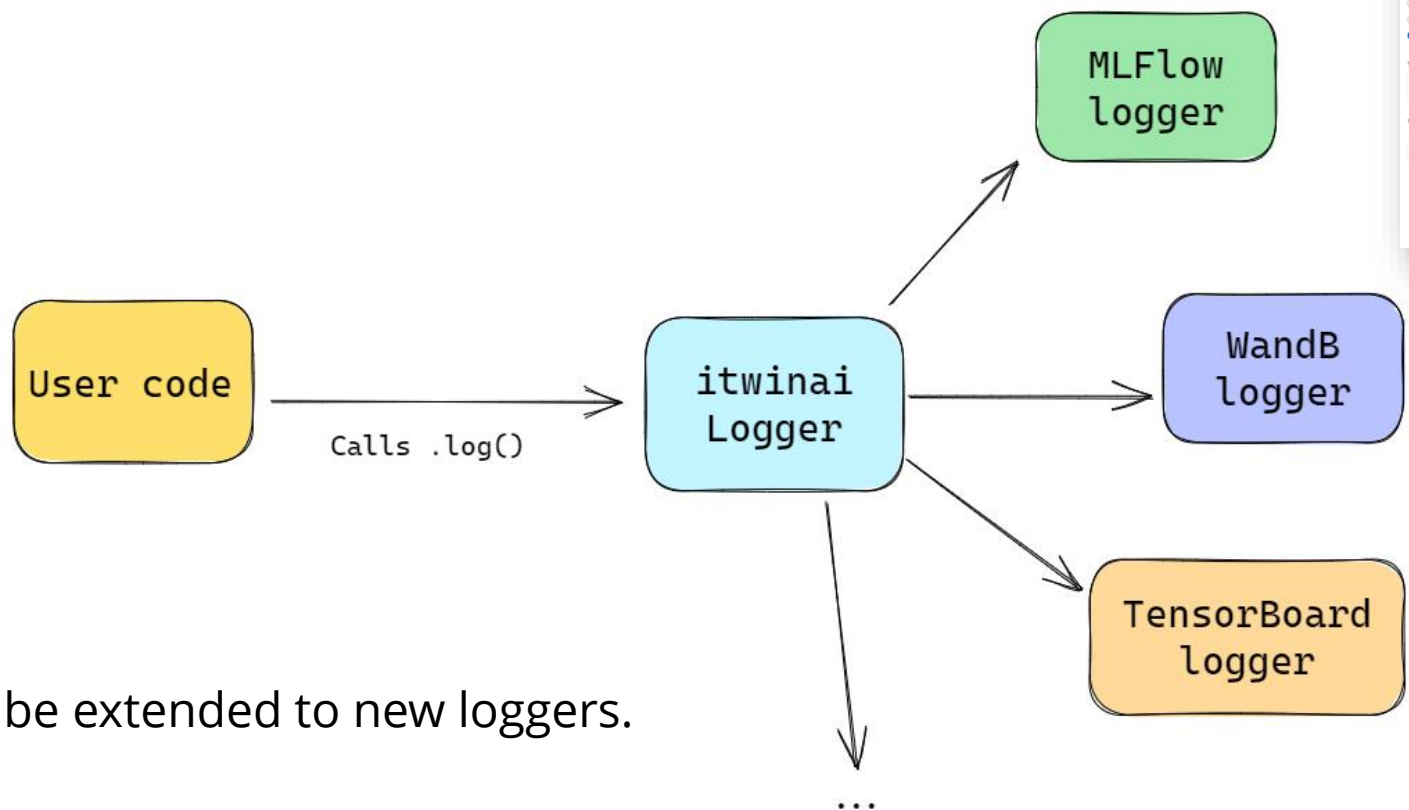
## Unified distributed training (informal representation)



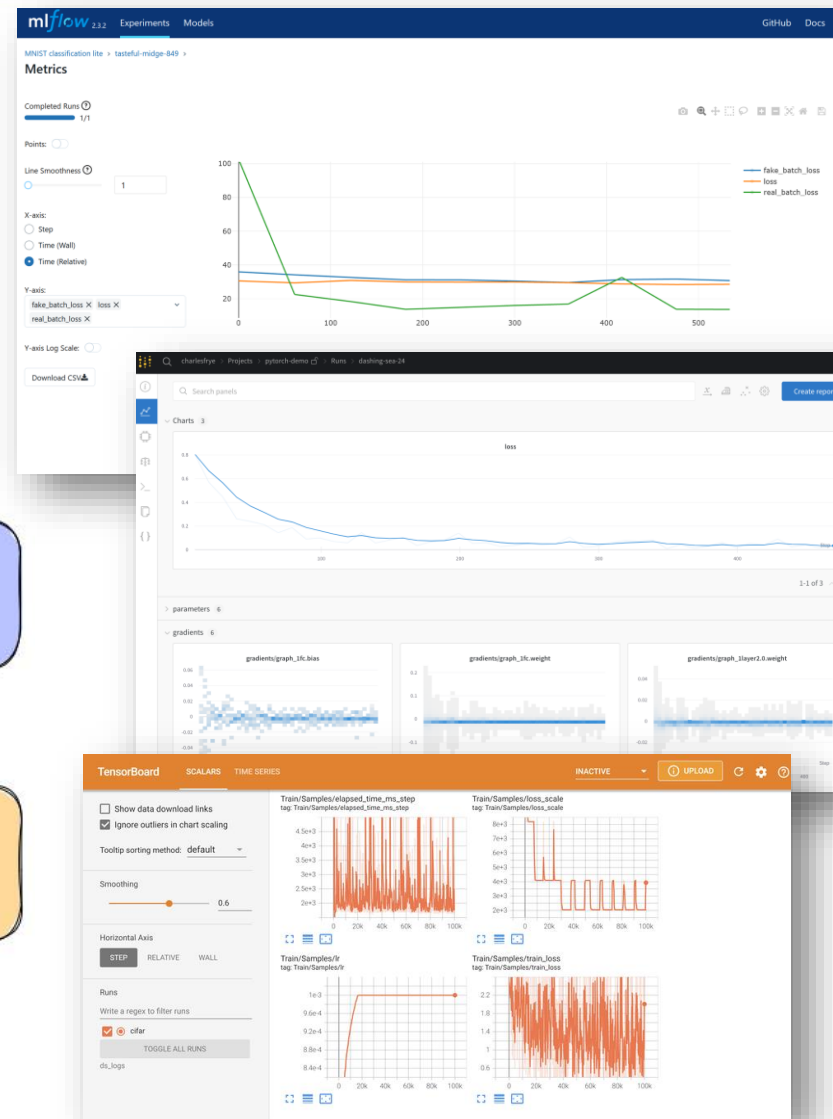


# Our contribution to the Core Engine

Unified logger (informal representation) to track ML metadata.

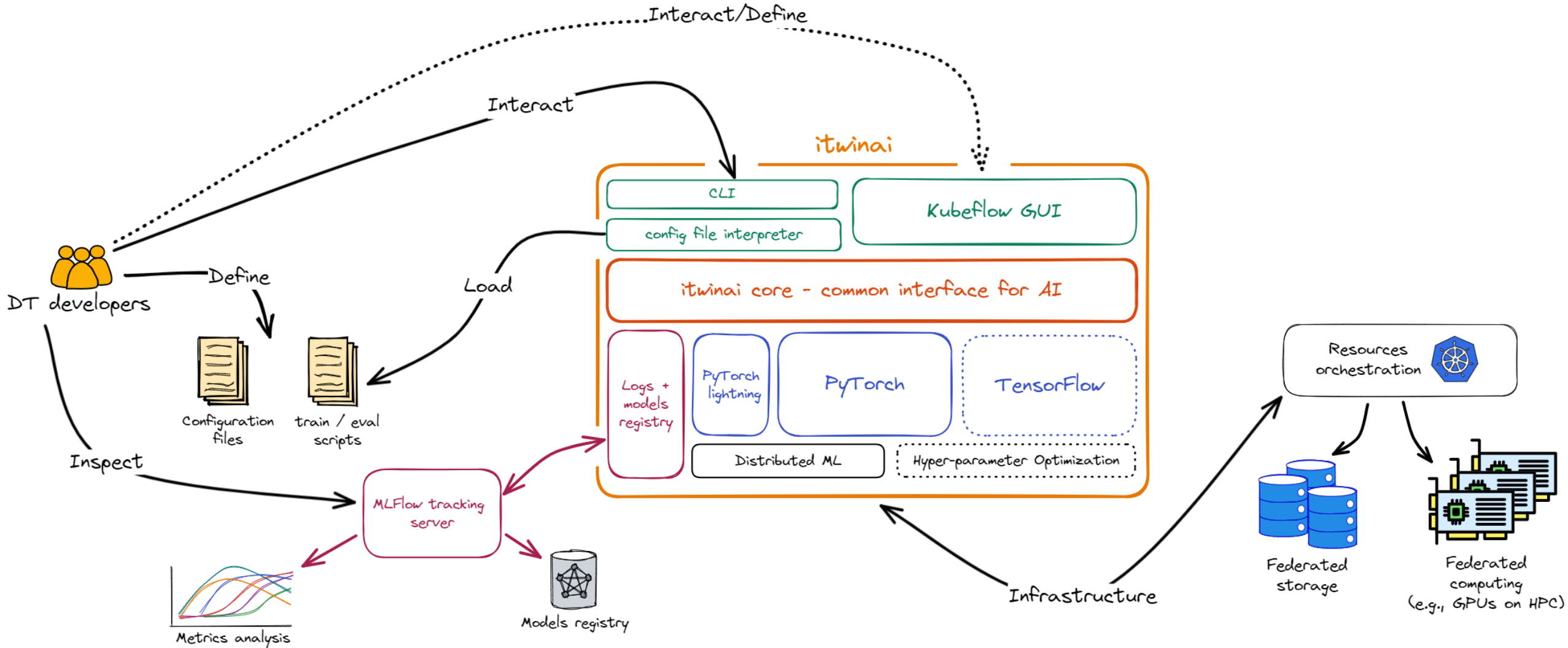


Can be extended to new loggers.





# itwinai - ML tooling for DT applications





# Latest developments

Development status of `itwinai` library:

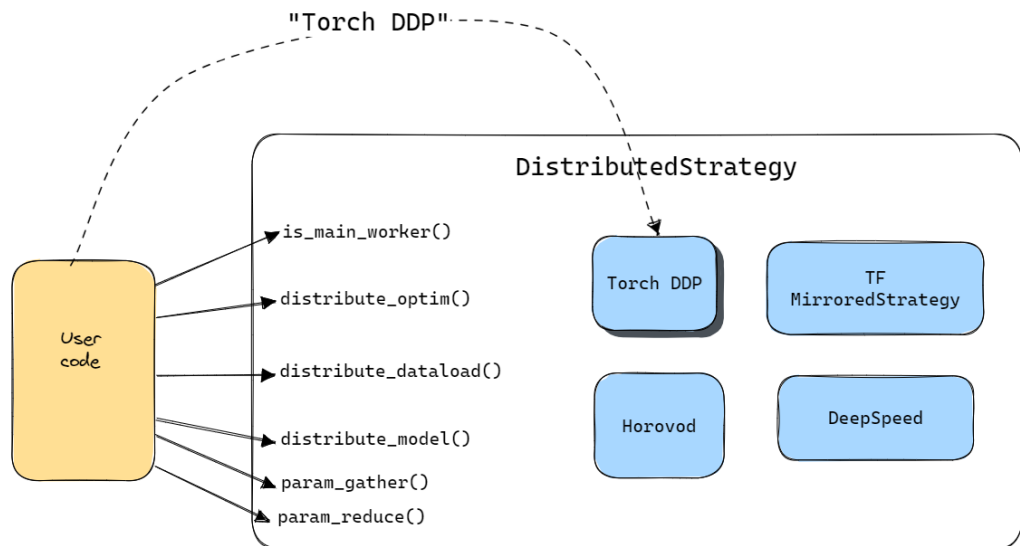
- Support for **PyTorch**, **TensorFlow** under development.
- **AI workflows**: exploration of **KubeFlow** Pipelines.
- **Distributed ML**: integration of existing strategies (e.g., ToCh DDP, Horovod, DeepSpeed).
- “Read The Docs”-style [documentation page](#).
- **Link with the infrastructure**: Docker/Singularity container, offloaded through WP5’s [interLink](#) on cloud/HPC systems.
- **ML logs and models**: MLFlow tracking and WandB.
- EC review demo [video](#).
- D6.2 First release of the DTE core modules: <https://doi.org/10.5281/zenodo.10224213>.



# Preliminary results – Distributed training

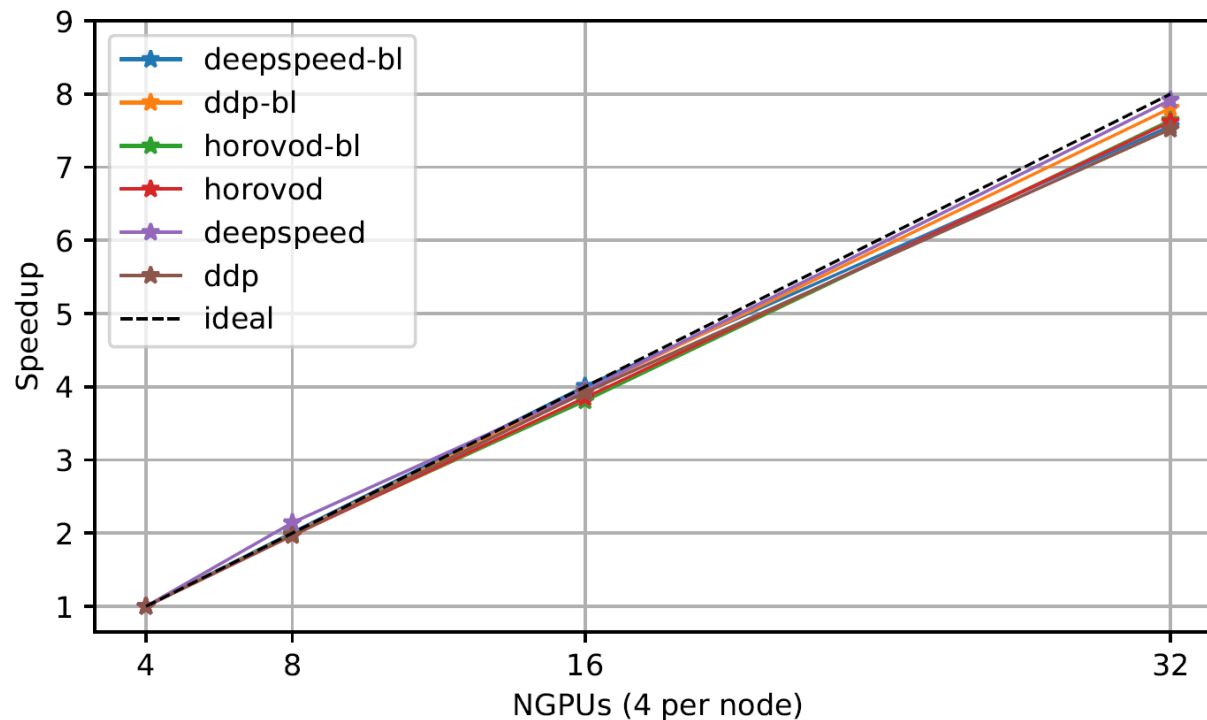
Scalability comparison between itwinai distributed ML API and native frameworks (Torch DistributedDataParallel, Microsoft DeepSpeed, and Horovod) serving as baselines (marked with "-bl" suffix).

Scalability measured on ResNet152 on a subset of Imagenet.



Intuitive representation of itwinai distributed ML API.

Comparison of native frameworks with itwinai distributed





# UX/UI: KubeFlow

**Kubeflow**

kubeflow-user-example-c...

Dashboard Activity

### Quick shortcuts

- ⚡ Upload a pipeline (Pipelines)
- ⚡ View all pipeline runs (Pipelines)
- ⚡ Create a new Notebook server (Notebook Servers)
- ⚡ View Katib Experiments (Katib)

### Recent Notebooks

- 📄 3DGAN-CERN.ipynb (Accessed 12/13/2023, 1:05:58 PM)
- 📄 kf-pipeline.yaml (Accessed 12/13/2023, 1:04:05 PM)

### Recent Pipelines

- 🔧 [Tutorial] DSL - Control structures (Created 12/13/2023, 12:33:56 PM)
- 🔧 [Tutorial] Data passing in python components (Created 12/13/2023, 12:33:55 PM)

### Recent Pipeline Runs

- ✅ kf-pipeline.yaml 2023-12-13 12-04-08 (Created 12/13/2023, 1:04:08 PM)

### Documentation

- Getting Started with Kubeflow (Get your machine-learning workflow up and running on Kubeflow)
- MiniKF (A fast and easy way to deploy Kubeflow locally)
- Microk8s for Kubeflow (Quickly get Kubeflow running locally on native hypervisors)
- Kubeflow on GCP (Running Kubeflow on Kubernetes Engine and Google Cloud Platform)
- Kubeflow on AWS (Running Kubeflow on Elastic Container Service and Amazon Web Services)
- Requirements for Kubeflow (Get more detailed information about using Kubeflow and its components)

Privacy • Usage Reporting  
build version dev\_local



# UX/UI: KubeFlow and JupyterLab

The screenshot displays the JupyterLab user interface. On the left is a file browser with a search bar and a table of files. The main area is divided into a launcher and a notebook editor.

Name	Last Modified
3DGAN-CE...	a day ago
Y: kf-pipeline...	a day ago

The launcher contains several tool icons: Notebook, Console, Terminal, and Text File, each with a Python 3 (ipykernel) icon.

The notebook editor shows the following content:

## 3DGAN use case for particles detector simulation

### Configuration

```
[1]: from kfp import dsl
tdgan_image = "ghcr.io/intertwin-eu/itwinai-3dgan-inference:0.0.3-light-2"
```

### Components

1. Load and preprocess dataset
2. Train 3DGAN generative model

```
[2]: from kfp.dsl import Input, Output, Artifact

@dsl.component(base_image=tdgan_image)
def dataloader(particles_dataset: Output[Artifact]):
    from itwinai.components import load_pipeline_step
    print(f"Save path: {particles_dataset.path}")
    dataloading_step = load_pipeline_step(
        pipe='pipeline.yaml',
        step_id='dataloading_step',
        override_keys={
            'init_args.data_path': particles_dataset.path
        },
        verbose=True
    )
    dataloading_step()
```



# UX/UI: MLFlow models registry

mlflow 2.8.1

Experiments

Models

Registered Models > 3dgan-lite >

## Version 1

Registered At: 2023-12-13 13:10:10

Source Run: [unleashed-fox-951](#)

> Description [Edit](#)

> Tags

▼ Schema

Stage: None ▼

Transition to → [Staging](#)

Transition to → [Production](#)

Transition to → [Archived](#)

Name	Type
------	------

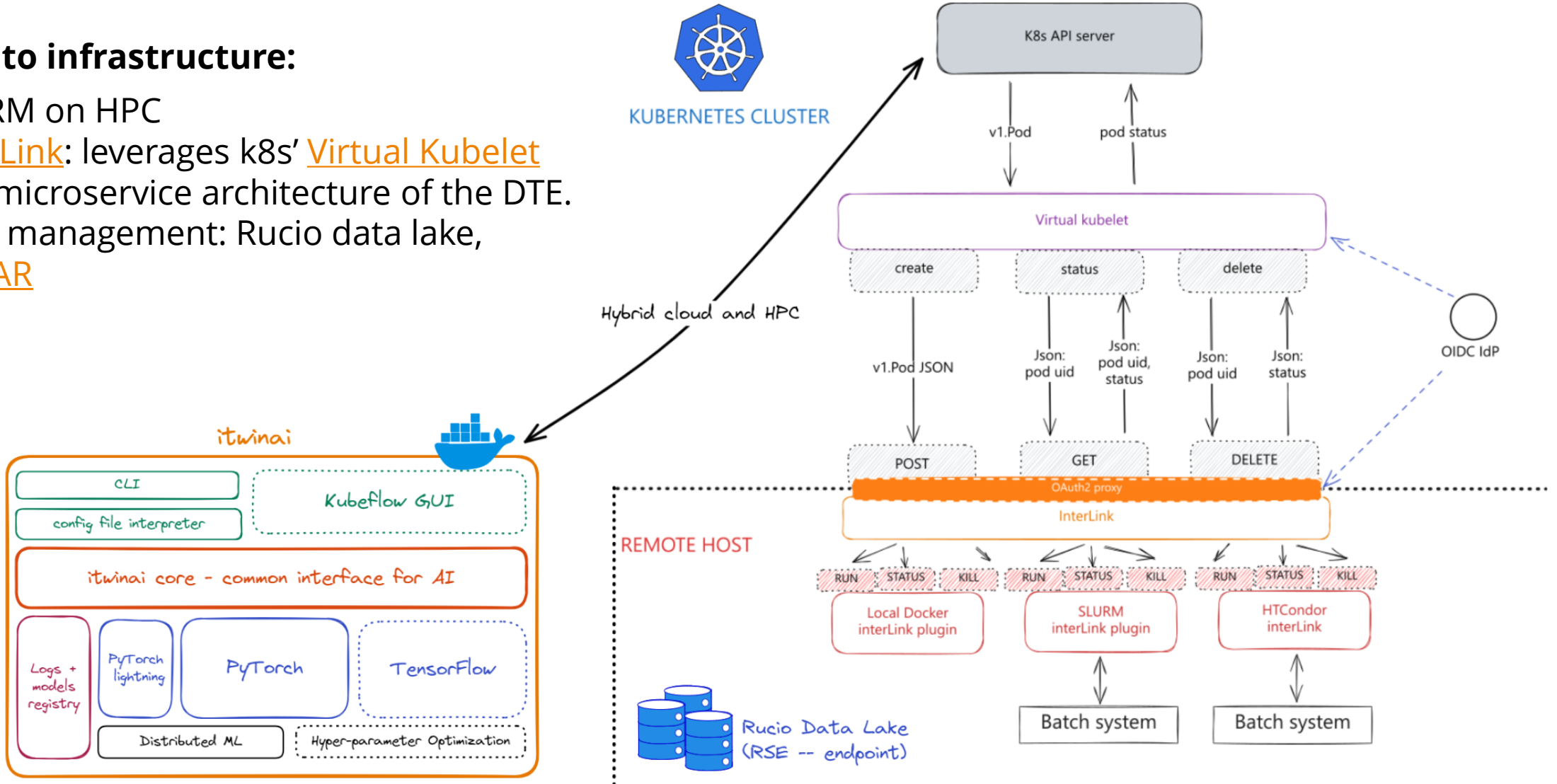
No schema. See [MLflow docs](#) for how to include input and output schema with your model.



# Federated computing

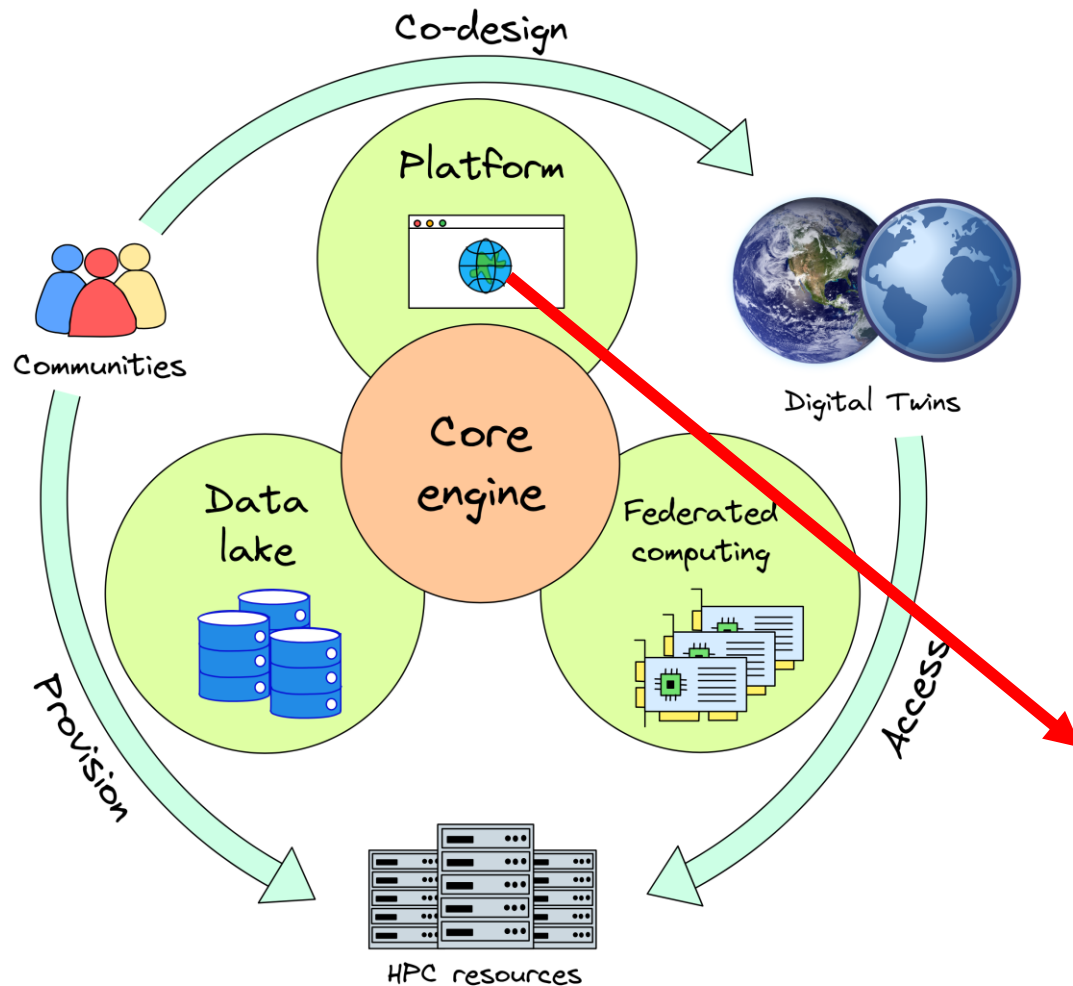
## Access to infrastructure:

- SLURM on HPC
- **InterLink**: leverages k8s' **Virtual Kubelet** and microservice architecture of the DTE.
- Data management: Rucio data lake, **OSCAR**



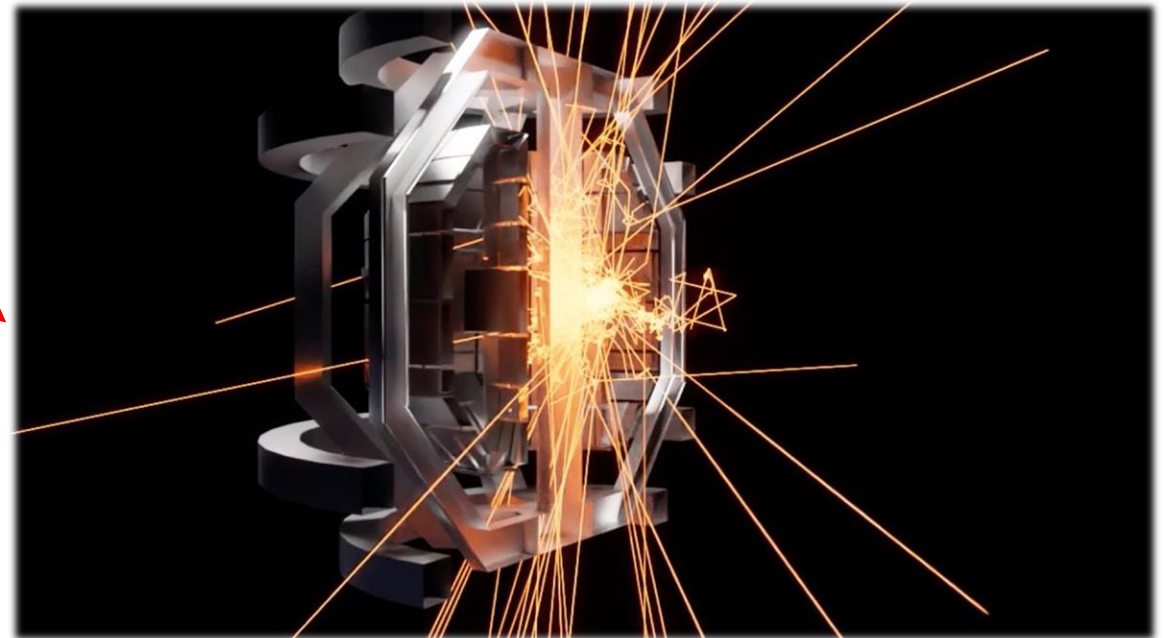


# Link with industry (CERN openlab)



## Nvidia Omniverse:

- Visualization of scientific data (HEP)
- Support the design of new detectors and accelerators by integration and simulation tests
- Training of RL-based agents (e.g., robots) in simulated environments ([IsaacGym](#))





# Thank you!      Questions?

[matteo.bunino@cern.ch](mailto:matteo.bunino@cern.ch)



[www.intertwin.eu](http://www.intertwin.eu)



[info@intertwin.eu](mailto:info@intertwin.eu)



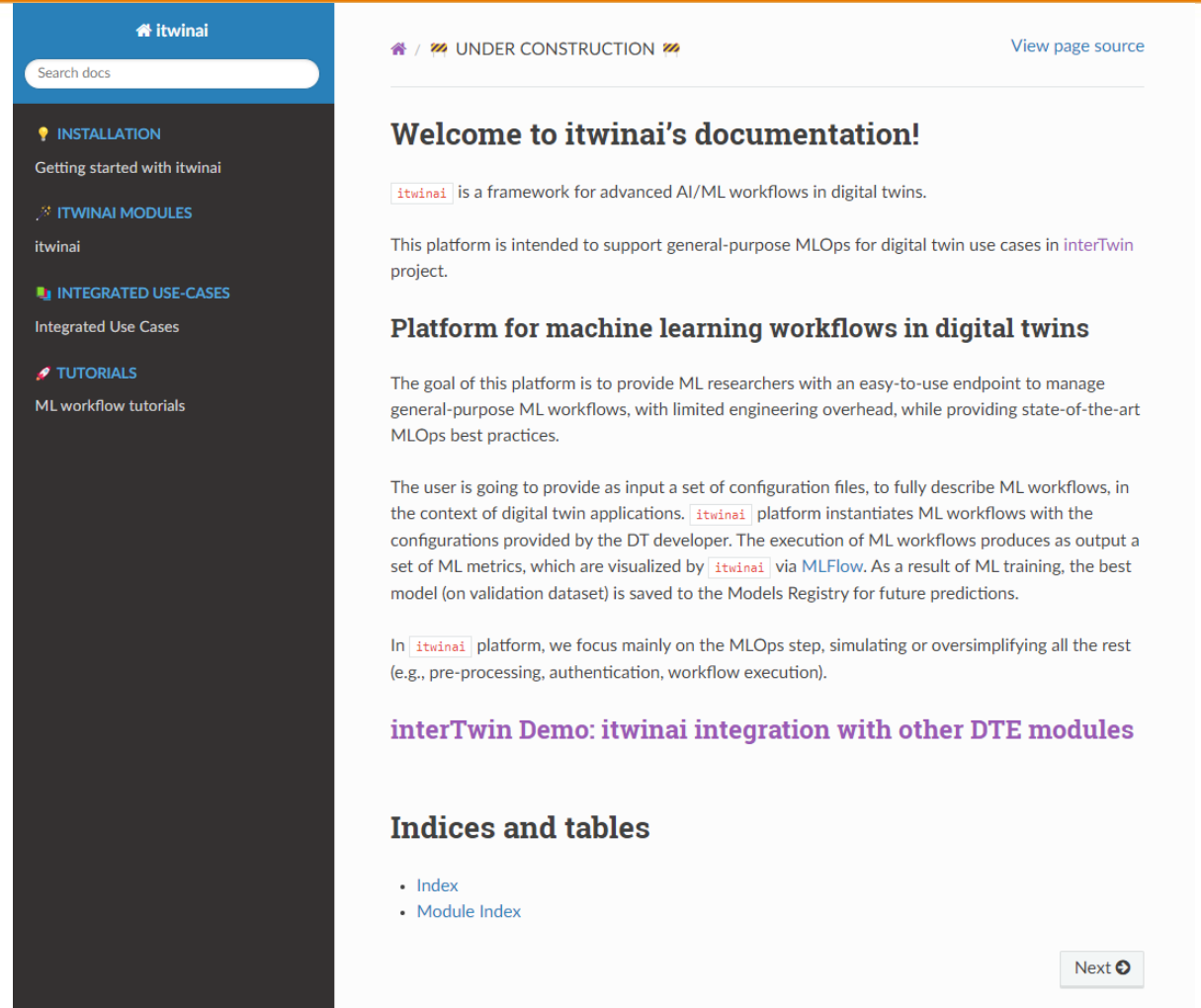
[intertwin\\_eu](https://twitter.com/intertwin_eu)



[intertwin](https://www.linkedin.com/company/intertwin)

GitHub repository: <https://github.com/interTwin-eu/itwinai>

- For the moment, please refer to the “dev” branch: <https://github.com/interTwin-eu/itwinai/tree/dev>
- Some **tutorials** available, **more to come**. Check “tutorials” folder: <https://github.com/interTwin-eu/itwinai/tree/dev/tutorials> and <https://github.com/interTwin-eu/itwinai/tree/distributed-strategy-launcher/tutorials/distributed-ml>
- Additional examples are under “use-cases” folder: <https://github.com/interTwin-eu/itwinai/tree/dev/use-cases>
- Docs: <https://intertwin-eu.github.io/itwinai/>



The screenshot shows the documentation page for itwinai. The left sidebar contains a navigation menu with sections: INSTALLATION (Getting started with itwinai), ITWINAI MODULES (itwinai), INTEGRATED USE-CASES (Integrated Use Cases), and TUTORIALS (ML workflow tutorials). The main content area has a header with 'itwinai' and 'UNDER CONSTRUCTION' status, and a search bar. The main text includes a welcome message, a description of itwinai as a framework for advanced AI/ML workflows, and a section titled 'Platform for machine learning workflows in digital twins'. It explains the platform's goal to provide an easy-to-use endpoint for managing ML workflows and mentions that configurations are provided by the DT developer. A section titled 'interTwin Demo: itwinai integration with other DTE modules' is also visible. At the bottom right, there is a 'Next' button with a right arrow icon.