Outline
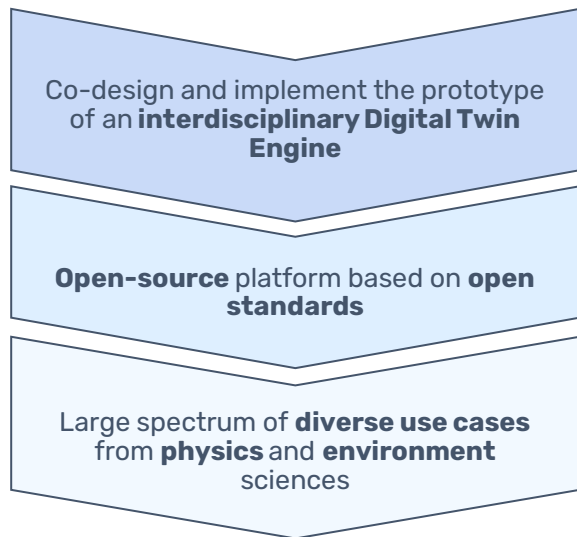
- Project intro

- Digital Twin Engine Architecture

- First Software release

- DTE Core components highlight and example use case

- Conclusions

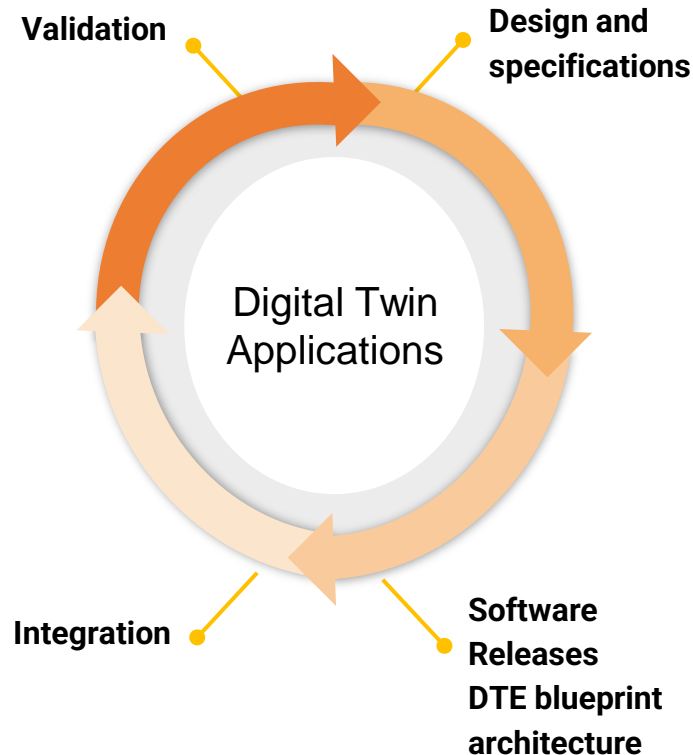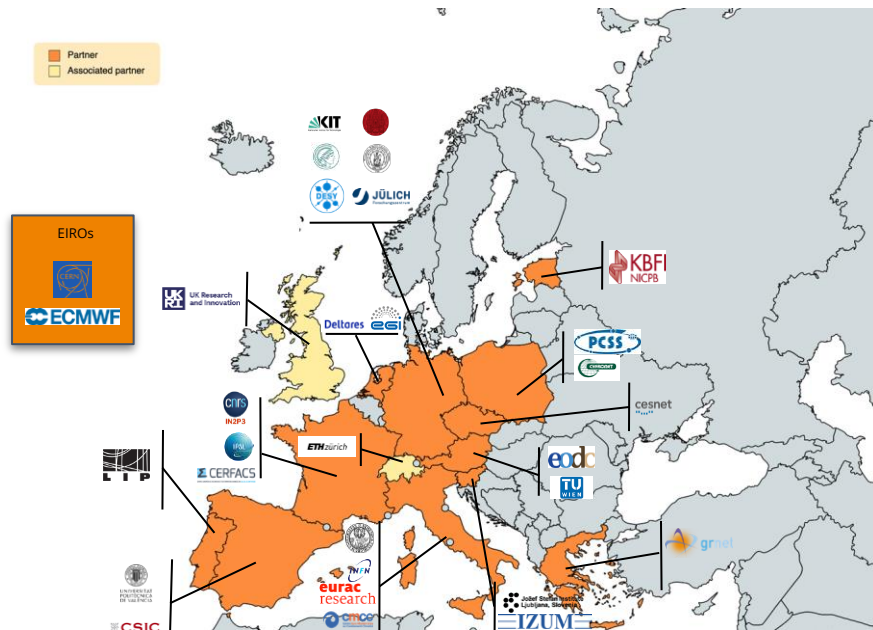A **Digital Twin (DT)** is a **virtual** representation of a **physical** object, process, or system. It is created and sustained with information derived from one or many sources of data such as sensors or models considering historical as well as real-time observations.

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 **environment** sciences

See Bjorn's talk for new definitions from DELTARES

Validation

Design and specifications

Digital Twin Applications

Integration

Software Releases DTE blueprint architecture

3

Consortium

Coordinator: EGI

30

Participants,
1 affiliated entity
2 associated partners

Cloud — EODC — EGI — KBFI — HTC — GRNET — UKRI — TU Wien — PSNC — Juelich — EuroHPC — HPC

1.09.22 - 31.08.25

Budget 11,7 M euro
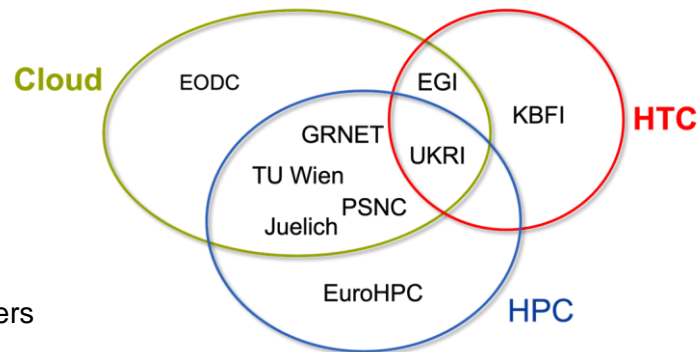
10 Providers	11 Technology providers	14 Community representants
Resources: **Cloud** **HTC** **HPC**	delivering the DTE infrastructure and horizontal capabilities	from 5 domains requirements and developing DT applications and thematic modules

interTwin use cases

Early Flood Warnings

Fire Hazard Prediction

Cyclone Detection

Drought Prediction

Radio Astronomy

High-energy Physics

Lattice QCD

Gravitational Waves

https://www.intertwin.eu/use-cases/

Tropical Cyclone Detection
CMCC, CNRS, Univ. of Trento

WildFire Hazard Map Generation
CMCC, CNRS, Univ. of Trento

Early warning for Extreme events
Deltares, EURAC, Technical Univ. of Wien

Extreme events impacts
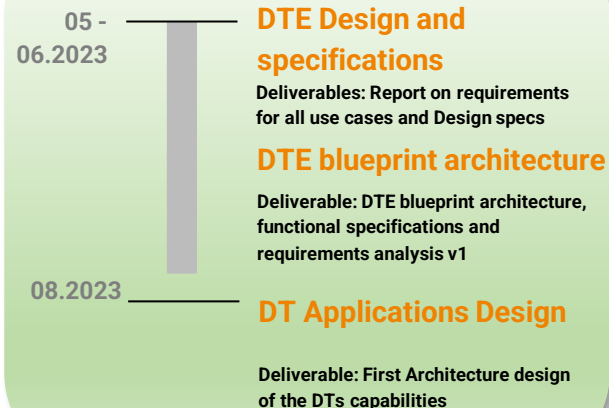CERFACS, EURAC, Deltares

6

Benefits of interdisciplinary approach

- **Collaboration**: Increase in cross-community development efforts and unification of frameworks used - "breaking down silos".

- **Portability**: Run DT workflows infrastructure agnostic across multiple Cloud/HPC centers in Europe.

- **Extensibility**: Easy addition of new use cases.

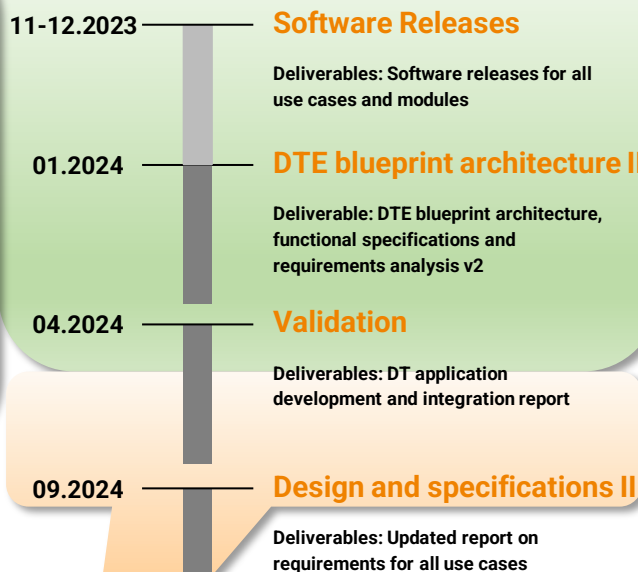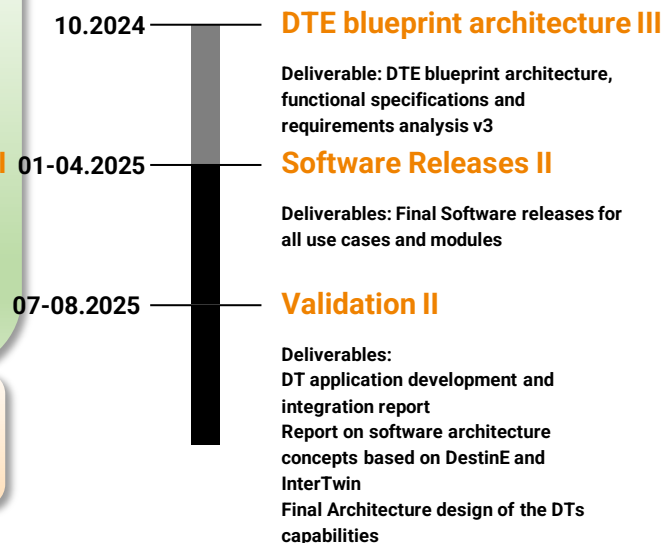- **Modularity**: Customizable according to specific use case's needs.

Timeline

Project Year 1 — PUBLISHED

05 - 06.2023

DTE Design and specifications
Deliverables: Report on requirements for all use cases and Design specs

DTE blueprint architecture
Deliverable: DTE blueprint architecture, functional specifications and requirements analysis v1

08.2023

DT Applications Design
Deliverable: First Architecture design of the DTs capabilities

Project Year 2 — RELEASED

11-12.2023

Software Releases
Deliverables: Software releases for all use cases and modules

01.2024

DTE blueprint architecture II
Deliverable: DTE blueprint architecture, functional specifications and requirements analysis v2

04.2024

Validation
Deliverables: DT application development and integration report

09.2024

Design and specifications II
Deliverables: Updated report on requirements for all use cases

Project Year 3

10.2024

DTE blueprint architecture III
Deliverable: DTE blueprint architecture, functional specifications and requirements analysis v3

01-04.2025

Software Releases II
Deliverables: Final Software releases for all use cases and modules

07-08.2025

Validation II
Deliverables:
DT application development and integration report
Report on software architecture concepts based on DestinE and InterTwin
Final Architecture design of the DTs capabilities

2023
2024
2025

Focusing on now!

8

DTE Blueprint and co-design

- Second version of the Blueprint architecture and design specifications are available in Github and Zenodo.
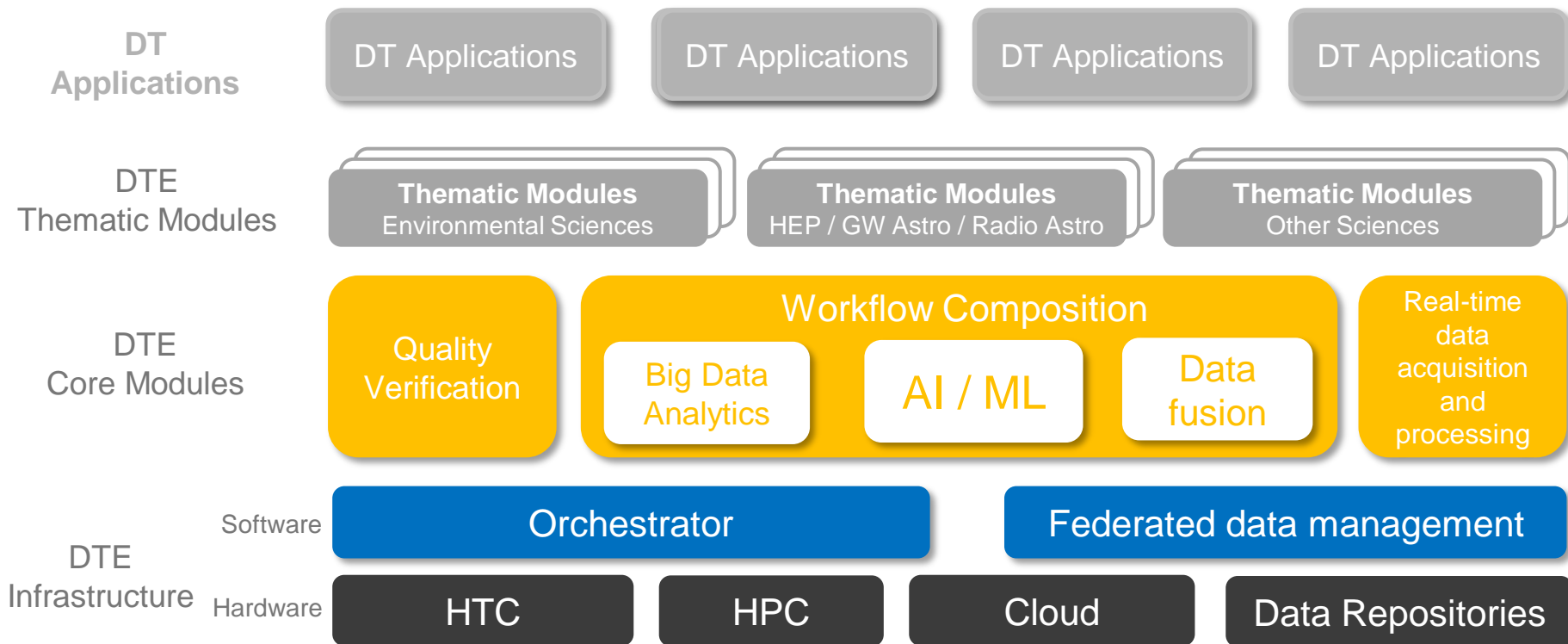 Final version is planned for Q4 2024
  - https://zenodo.org/records/10650440
  - https://github.com/interTwin-eu/architecture-diagrams/tree/main/Blueprint%20architecture

- It also considers other relevant initiatives and projects (*DestinationE, EOSC, ESCAPE, C-SCALE, Digital Twin Consortium, Gaia-X, and other EU Data Spaces*) to identify potential architectural components that can be incorporated within the interTwin context and where interoperability is desirable.

interTwin Components

DT Applications

DT Applications | DT Applications | DT Applications | DT Applications

DTE Thematic Modules

Thematic Modules Environmental Sciences

Thematic Modules HEP / GW Astro / Radio Astro

Thematic Modules Other Sciences

DTE Core Modules

Quality Verification

Workflow Composition

Big Data Analytics

AI / ML

Data fusion

Real-time data acquisition and processing

DTE Infrastructure

Software

Orchestrator

Federated data management

Hardware

HTC | HPC | Cloud | Data Repositories

interTwin DTE First Release

interTwin DTE first release description available on our Website

https://www.intertwin.eu/intertwin-digital-twin-engine/

- 38 components in total
- New components developed and extension to existing software
- https://github.com/interTwin-eu
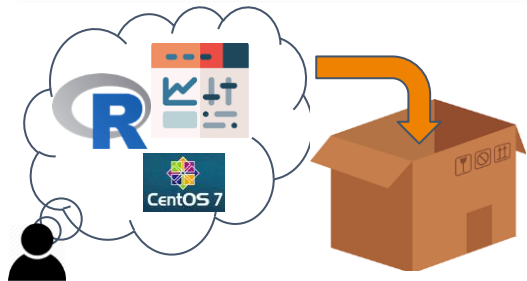
interTwin Premise:
The building block of Digital Twins are software containers

Container: unit of software, packaging a given code and all its dependencies
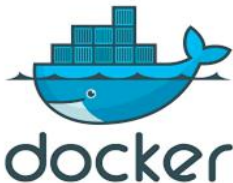
Docker: platform designed to create and manage containers, repos of containers, etc..

Container orchestration tool to execute containers across computing resources

What is container orchestration ?

- Provisioning and deployment of "containerized applications" across computing infrastructures, in an <u>automated way</u>

Why do we need container orchestration ?

- Each Digital Twin will be composed of several containers
- <u>Running the Digital Twin consists in the execution of all those containers following a given workflow.</u>
- This can become a <u>very tedious job if done manually and not accessible to general end-users.</u>
- The tools for automated container orchestration make that operational complexity manageable

12

From the Software stack to the Digital Twin - I

1 Code repository

2 Container repository

+ support recipes.

3 Workflow registry

CWL standard

openEO , Ophidia, Airflow, Streamflow (WfM tools)

eg. templates workflow as code

```
from datetime import datetime

from airflow import DAG
from airflow.decorators import task
from airflow.operators.bash import BashOperator

# A DAG represents a workflow, a collection of tasks
with DAG(dag_id="demo", start_date=datetime(2022, 1, 1), schedule="0 0 * * *") as dag:

    # Tasks are represented as operators
    hello = BashOperator(task_id="hello", bash_command="echo hello")

    @task()
    def airflow():
        print("airflow")

    # Set dependencies between tasks
    hello >> airflow()
```

13

1 Code repository

2 Container repository + support recipes TOSCA Templates Repository ANSIBLE Artifacts Repository

3 Workflow registry CWL standard COMMON WORKFLOW LANGUAGE openEO , Ophidia, Airflow, Streamflow (WfM tools)

4 Reuse the building blocks among the **core modules**

Quality Verification

Workflow Composition
Big Data Analytics
AI / ML
Data fusion

Real-time data acquisition and processing

Thematic Modules
Environmental Sciences

Thematic Modules
HEP / GW Astro / Radio Astro

Thematic Modules
Other Sciences

1 Code repository

2 Container repository

3 Workflow registry

APACHE kafka   ml*flow*   CernVM File system

+ support recipes   TOSCA Templates Repository   ANSIBLE Artifacts Repository

CWL standard   COMMON WORKFLOW LANGUAGE   openEO , Ophidia, Airflow, Streamflow (WfM tools)

Thematic Modules Environmental Sciences

Thematic Modules HEP / GW Astro / Radio Astro

Thematic Modules Other Sciences

4 Reuse the building blocks among the **core modules**

Quality Verification

Workflow Composition

Big Data Analytics   AI / ML   Data fusion

Real-time data acquisition and processing

Run the execution of the DT **5**

Orchestrator   Federated data management

HTC   HPC   Cloud   Data Repositories

15

DTE Core components

Automated DT Validation in connection with workflow provenance

yProv

SQAaaS

Distributed data analysis embedded with specific workflow tools

openEO

Connecting Real-time data with serverless processing
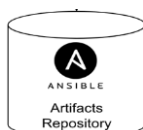
OSCAR

Workflow Composition

Quality Verification

Big Data Analytics

AI / ML

Data fusion

Real-time data acquisition and processing

TOSCA
Templates Repository

ANSIBLE
Artifacts Repository

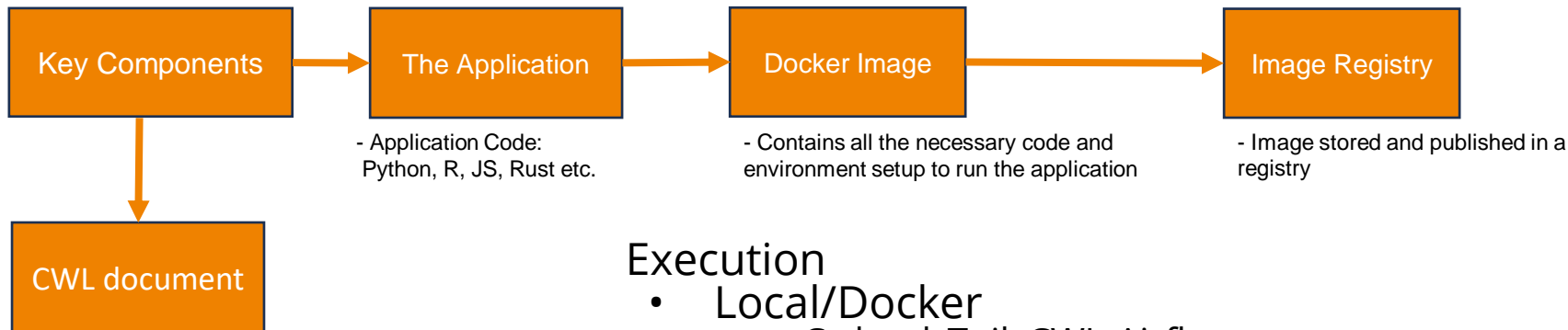Standardized deployment of Big Data Analytics tools

itwinai

Generic ML / AI training platform with support to generic workflow management and model validation

Linking interTwin Thematic Modules into workflows via OGC Application Packages

"Describes how to **package** EO computational workflows targeting their execution **automation, scalability, reusability and portability** while also being workflow-engine and **vendor neutral**." (Terradue, n.d)

```
Key Components  →  The Application  →  Docker Image  →  Image Registry
       ↓
CWL document
```

- Application Code: Python, R, JS, Rust etc.

- Contains all the necessary code and environment setup to run the application

- Image stored and published in a registry

- Interface to setup define:
  - Inputs and Outputs
 - Steps
  - Requirements (Containers, Resources…)

Execution
- Local/Docker
  - Cwltool, Toil, CWL-Airflow
- In Kubernetes
  - Application Deployment and Execution Service (ADES)

openEO cwltool FastAPI Architecture

- Based on CWL-FastAPI
- Extended functionality via OGC-API Processes to store and execute Application Packages
- Execution based on cwltool
- Execution called from openEO process graphs
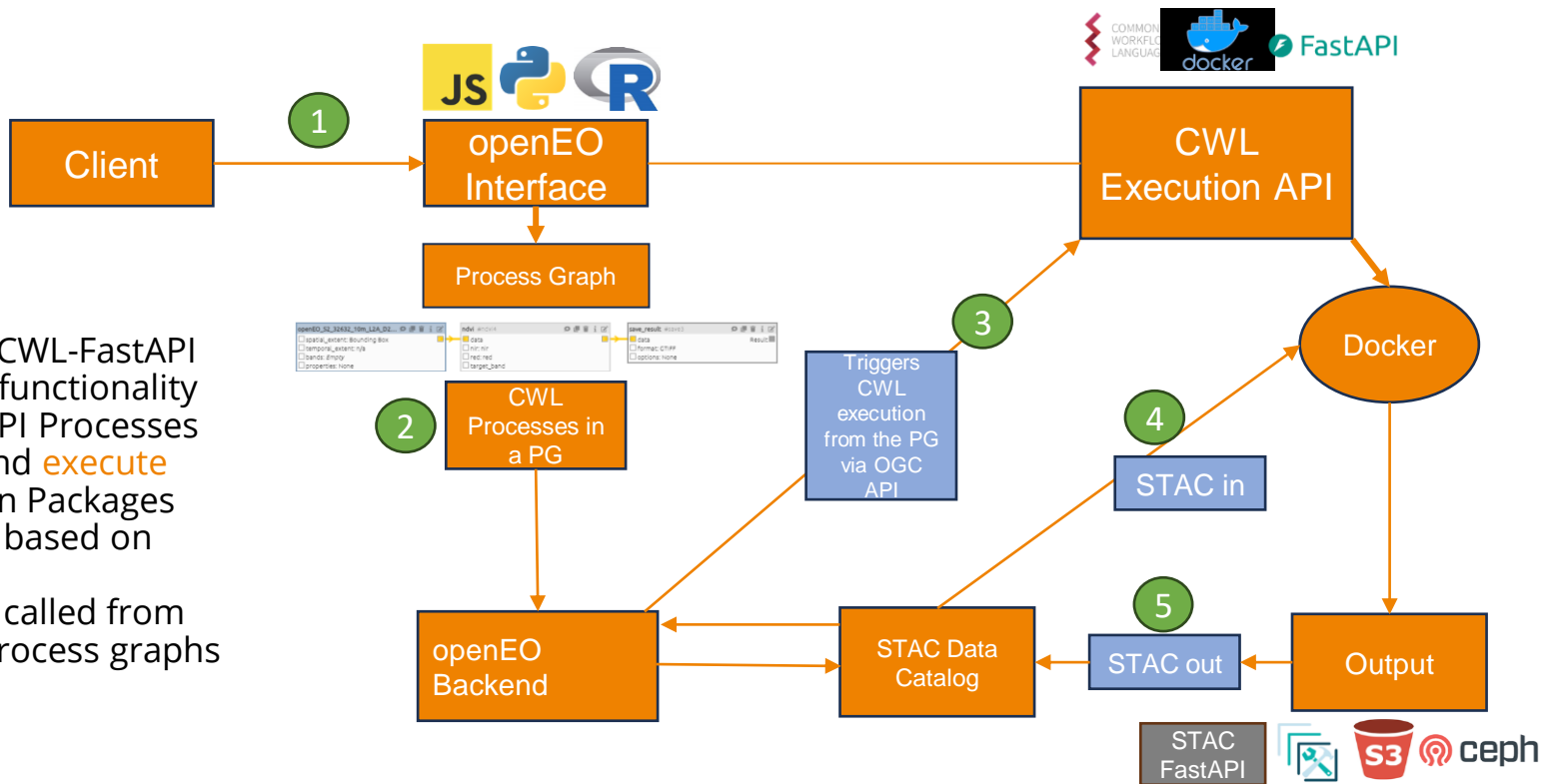
# Itwinai - ML tooling for DT applications

Support AI-based digital twin applications in science:

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

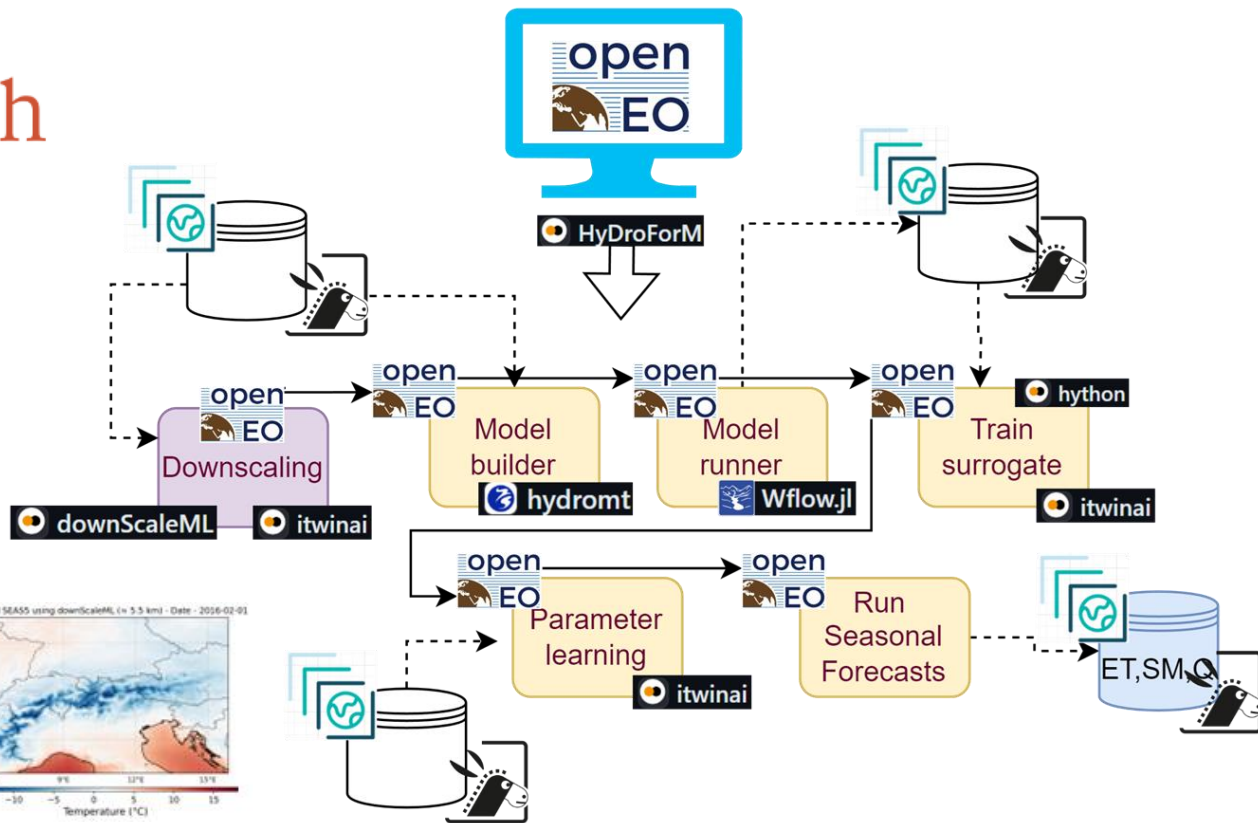# Itwinai - ML tooling for DT applications

Demo video: https://www.youtube.com/watch?v=NoVCfSxwtX0
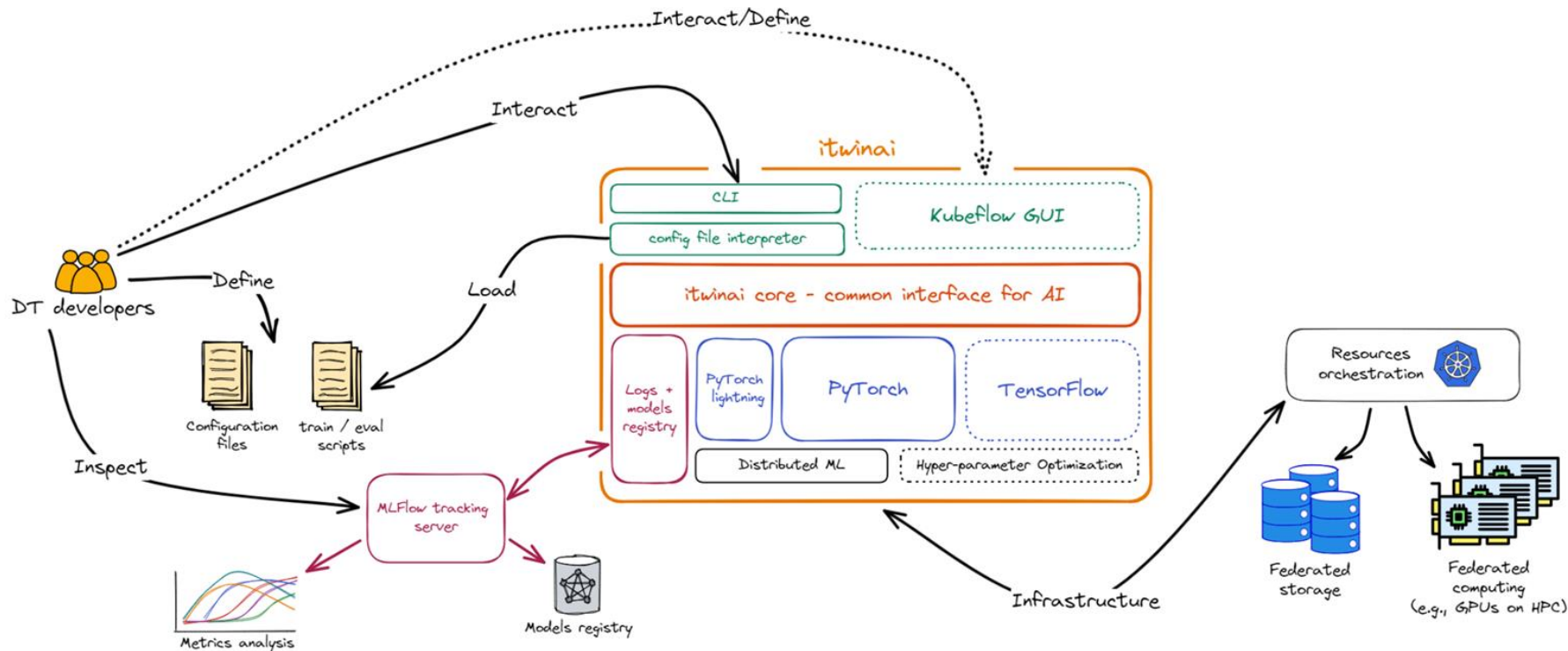
Conclusions

Successfull co-design process with User Communities leading to Blueprint Architecture and DTE Components definition

First Release available, new components opensourced in our Github community .https://github.com/interTwin-eu

Pilots running on our infrastructure providers and first integration with use cases completed end of May 2024

We aim at completing the developments in Jan 2025 and look for external early adopters

Final goal is to extend and operate the DTE after the end of the project as one of the services of the EGI Federation

22

Thank you!   Questions?

Backup

interTwin Specific Objectives

Objective 1. Co-design, develop and provide a Digital Twin Engine that simplifies & accelerates the development of complex application-specific DTs that benefits researchers, business and civil society

Objective 2. Co-design a Digital Twin Engine blueprint architecture that provides a conceptual framework for the development of DTs supporting interoperability, performance, portability & accuracy.

Objective 3. Extend the technical capabilities of the European Open Science Cloud with modelling & simulation tools integrated with its compute platform

Objective 4. Ensure trust and reproducibility in science through quality, reliability and verifiability of the outputs of Digital Twins

Objective 5. Demonstrate data fusion with complex modelling & prediction technologies

Objective 6. Simplify DT application development with tools to manage AI workflows and the model lifecycle while reinforcing open science practices

Interoperability & Link with DestinE

interTwin is conducting joint pilot activities with **DestinE** to **design a compatible architecture** that addresses the requirements of the largest set of user communities.

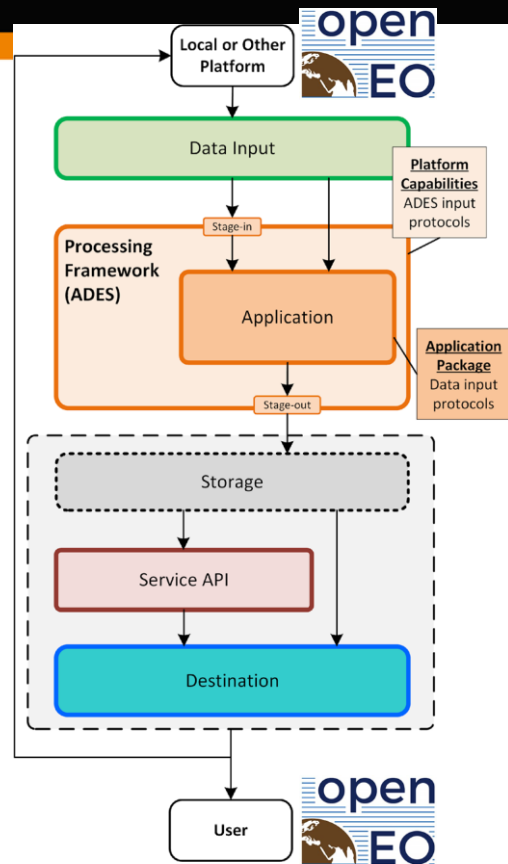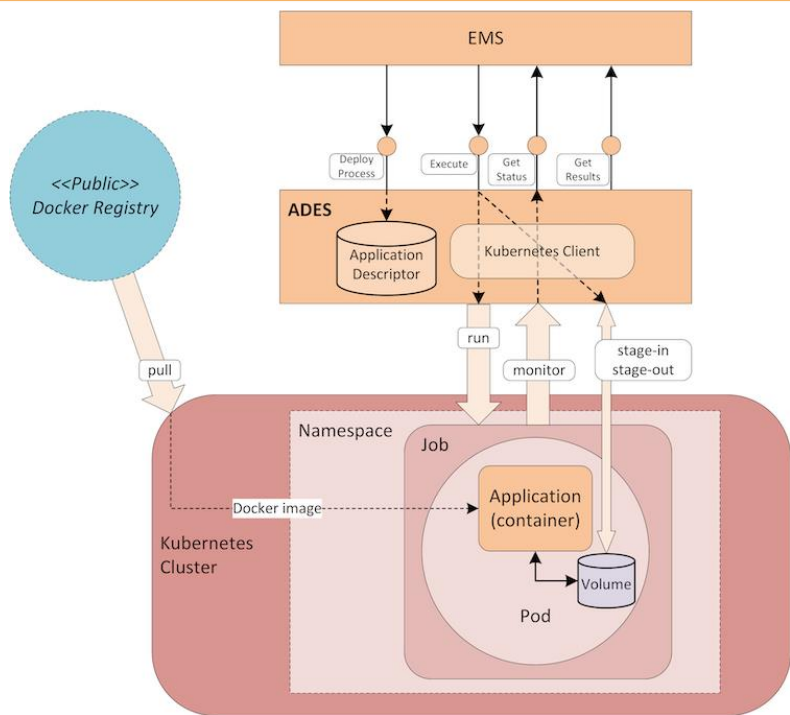**Interoperability** is the aim of this activity.
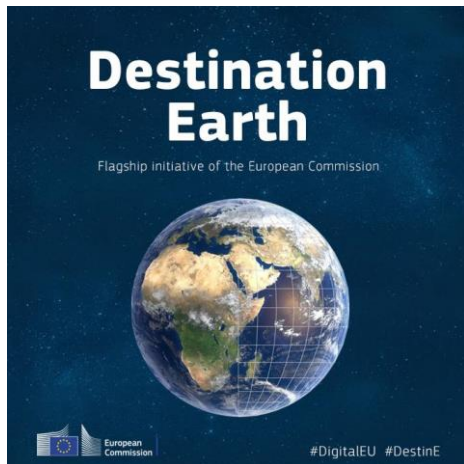
Demonstrators of data handling across interTwin and DestinE DTs for the Extremes and Climate in production-type configurations are under implmementaion in collaboration with **ECMWF**

Part of the collaboration with DestinE includes the **development of common software architecture concepts** that are also **applicable to other major DTs initiatives**.
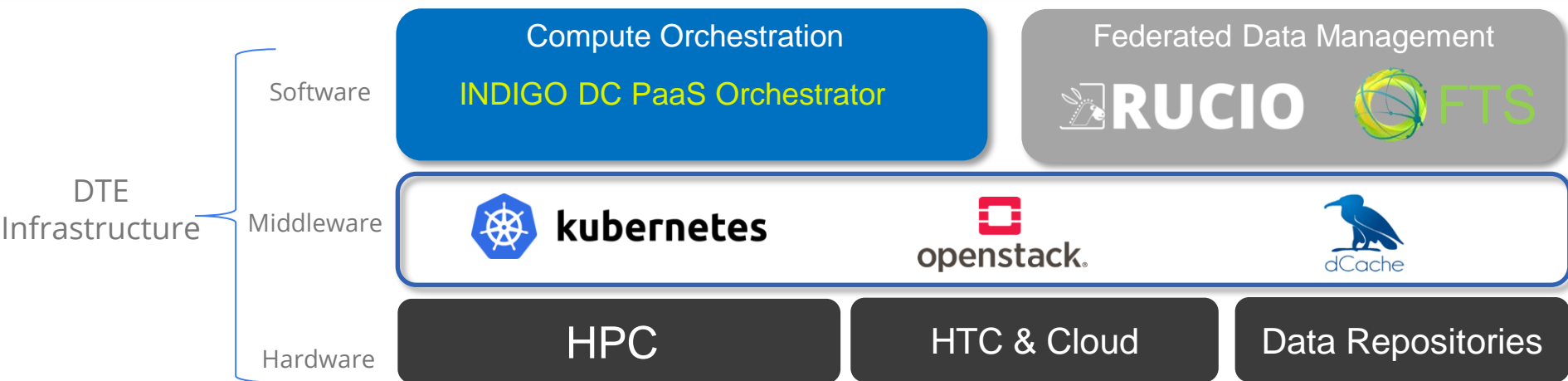
DTE Infrastructure components

DTE Infrastructure

Software

Compute Orchestration

INDIGO DC PaaS Orchestrator

Federated Data Management

RUCIO · FTS

Middleware

kubernetes · openstack · dCache

Hardware

HPC | HTC & Cloud | Data Repositories

Orchestration

PaaS Orchestrator + **Infrastructure Manager** elaborating deployment requested expressed in **TOSCA** to be extended to deal AI based orchestration.
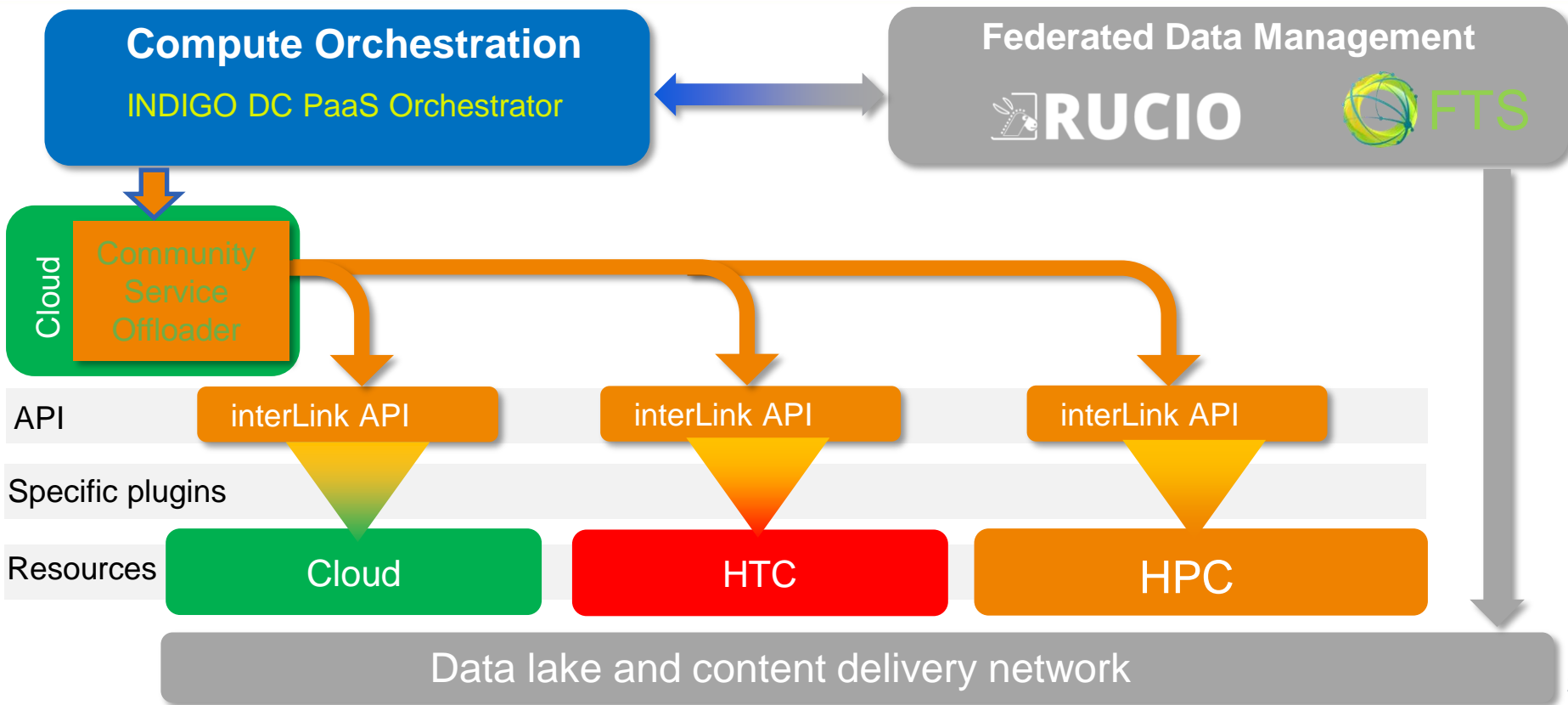
Federated Data Management

Based on **ESCAPE** Data Lake architecture and services, **Rucio**, **FTS** and HTTP accessed caches/storages. Data lake concept extended to HPC facilities

Continuum Integration

- **Single-sign-on** in simulation and modelling tasks to access data and different compute facilities,
- **Offloading to HPC**.
- Data repositories and computation with containers on HTC, Cloud and HPC

DTE Infrastructure in more detail

Compute Orchestration

INDIGO DC PaaS Orchestrator

Federated Data Management

RUCIO   FTS

Cloud

Community Service Offloader

API — interLink API   interLink API   interLink API

Specific plugins

Resources — Cloud   HTC   HPC

Data lake and content delivery network

Euro-HPCs integrated via interLink

HPC Vega is the first EuroHPC JU supercomputer hosted at the Institute of Information Science in Maribor, in Slovenia.

First HPC provider integrated, enabling super early prototyping

Demo video https://www.youtube.com/watch?v=-djIQGPvYdI

The Jülich Supercomputing Centre operates one of the most powerful supercomputers in Europe, JUWELS, and JUNIQ the first European infrastructure for quantum computing.

First volunteer for an external interLink plugin based on UNICORE