Future Generation Computer Systems 179 (2026) 108312

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

interTwin: Advancing Scientific Digital Twins through AI, Federated Computing and Data

Andrea Manzi [a], Raul Bardaji [a], Ivan Rodero [a], Germán Moltó [b], Sandro Fiore [c], Isabel Campos [d,*], Donatello Elia [e], Francesco Sarandrea [f], A. Paul Millar [g], Daniele Spiga [h], Matteo Bunino [i], Gabriele Accarino [e,j,k], Lorenzo Asprea [f], Samuel Bernardo [l], Miguel Caballer [b], Charis Chatzikyriakou [m], Diego Ciangottini [h], Michele Claus [n], Andrea Cristofori [a], Davide Donno [e,o], Emanuele Donno [e,o], Iacopo Ferrario [n], Massimiliano Fronza [c], Alexander Jacob [n], Javad Komijani [p], Marina Krstic Marinkovic [p], Federica Legger [o], Ivan Palomo [d], Estíbaliz Parcero [b], Rakesh Sarma [q], Gaurav Sinha Ray [d], Sara Vallero [f], Juraj Zvolensky [c]

[a] *EGI Foundation, Science Park 140, Amsterdam, 1098 XG, Netherlands*
[b] *Instituto de Instrumentación para Imagen Molecular (I3M) Centro mixto CSIC - Universitat Politécnica de Valéncia, Camino de Vera s/n, Valencia, 46022, España*
[c] *Department of Information Engineering and Computer Science, University of Trento, Trento, 38122, Italy*
[d] *Instituto de Fisica de Cantabria - CSIC, Avda de los Castros s/n, Santander, 39005, España*
[e] *CMCC Foundation - Euro-Mediterranean Center on Climate Change, Via Marco Biagi, 5, Lecce, 73100, Italy*
[f] *INFN Torino, Via Pietro Giuria 1, Torino, 10125, Italy*
[g] *Deutsches Elektronen-Synchrotron DESY[ROR], Notkestralle 85, Hamburg, 22607, Germany*
[h] *INFN Perugia, Via Pascoli snc, Perugia, 06121, Italy*
[i] *CERN, Esplanade des Particule, 1, Geneva, 1211, Switzerland*
[j] *Columbia University, Department of Earth and Environmental Engineering, 500 W 120th St, New York, 10027, USA*
[k] *Learning the Earth with Artificial Intelligence and Physics (LEAP) NSF STC, 2276 12th Ave, 2nd Floor, New York, 10027, USA*
[l] *Laboratory of Instrumentation and Particles - LIP, Av Prof Gama Pinto 2, Lisbon, 1649-003, Portugal*
[m] *EODC Earth Observation Data Centre for Water Resources Monitoring GmbH, Lothringerstraße 4/1, Vienna, 1040, Austria*
[n] *Eurac Research, Viale Druso 1, Bolzano, 39100, Italy*
[o] *University of Salento, Department of Engineering for Innovation, Via per Monteroni, Lecce, 73100, Italy*
[p] *ETHZ, Raemistrasse 101, Zurich, 8092, Switzerland*
[q] *Forschungszentrum Jülich GmbH, Jülich Supercomputing Centre, Wilhelm-Johnen-Stralle, Jülich, 52428, Germany*

ARTICLE INFO

Keywords:
Digital twins
Computing
Data management
HPC
Machine learning

ABSTRACT

The EU project interTwin, co-designed and implemented the prototype of an interdisciplinary Digital Twin Engine (DTE), an open-source platform that provides generic and domain-specific software components for modelling and simulation to integrate application-specific Digital Twins (DTs). The DTE is built upon a co-designed conceptual model - the DTE blueprint architecture - guided by open standards and interoperability principles. The ambition is to develop a unified approach to the implementation of DTs that is applicable across diverse scientific disciplines to foster collaborations and facilitate developments. Co-design involved DT use cases from

* Corresponding author.
E-mail addresses: andrea.manzi@egi.eu (A. Manzi), raul.bardaji@egi.eu (R. Bardaji), ivan.rodero@egi.eu (I. Rodero), gmolto@dsic.upv.es (G. Moltó), sandro.fiore@unitn.it (S. Fiore), isabel@campos-it.es (I. Campos), donatello.elia@cmcc.it (D. Elia), francesco.sarandrea@to.infn.it (F. Sarandrea), paul.millar@desy.de (A.P. Millar), daniele.spiga@pg.infn.it (D. Spiga), matteo.bunino@cern.ch (M. Bunino), gabriele.accarino@cmcc.it (G. Accarino), lorenzo.asprea@gmail.com (L. Asprea), samuel@lip.pt (S. Bernardo), micafer1@upv.es (M. Caballer), Charis.Chatzikyriakou@eodc.eu (C. Chatzikyriakou), diego.ciangottini@pg.infn.it (D. Ciangottini), michele.claus@eurac.edu (M. Claus), andrea.cristofori@egi.eu (A. Cristofori), davide.donno@cmcc.it (D. Donno), emanuele.donno@cmcc.it (E. Donno), iacopo.ferrario@eurac.edu (I. Ferrario), massimiliano.fronza@unitn.it (M. Fronza), alexander.jacob@eurac.edu (A. Jacob), jkomijani@ethz.ch (J. Komijani), marinama@ethz.ch (M.K. Marinkovic), federica.legger@to.infn.it (F. Legger), palomo@ifca.unican.es (I. Palomo), esparig@i3m.upv.es (E. Parcero), r.sarma@fz-juelich.de (R. Sarma), sinha@ifca.unican.es (G. Sinha Ray), svallero@to.infn.it (S. Vallero), juraj.zvolensky@eurac.edu (J. Zvolensky).

https://doi.org/10.1016/j.future.2025.108312
Received 13 May 2025; Received in revised form 21 November 2025; Accepted 28 November 2025
Available online 15 December 2025

high-energy physics, radio astronomy, astroparticle physics, climate research, and environmental monitoring, which drove advancements in modelling and simulation by leveraging heterogeneous distributed digital infrastructures, enabling dynamic workflow composition, real-time data management and processing, quality and uncertainty tracing of models, and multi-source data fusion.

1. Introduction

Scientific Digital Twins (DTs) face unique challenges that set them apart from industrial ones. They must integrate diverse data streams from multiple locations, support complex simulations across hybrid computing resources, and combine outputs from different scientific fields with varying data formats, computing needs, and research methods. Unlike industrial DTs operating in controlled environments with established protocols, scientific DTs have only recently been prototyped by diverse research communities spanning high-energy physics to climate science. Each field maintains distinct computing systems, data management practices, and simulation processes [1].

Current digital twin platforms mostly focus on single-domain applications or depend on centralized computing systems. These systems cannot support the federated, multi-institutional aspect of modern scientific research. Existing solutions do not have the necessary frameworks for interoperability, which hinders collaboration across disciplines. They also fail to deliver the scalable, distributed computing power needed for complex scientific simulations [2]. This limitation has created a significant gap. Even though the scientific community is starting to see the transformative potential of digital twins, researchers still do not have a unified platform that can handle the complex, collaborative, and computation-heavy nature of scientific digital twins.

To address these challenges, we introduce the Digital Twin Engine (DTE), a collaborative computing framework aimed at scientific applications. Unlike other platforms, the DTE allows smooth integration of high-performance computers, cloud services, and data storage spread across different locations. It features a modular design that supports real-time data updates, flexible computing, and standard interfaces across various fields. Prototyped through the European interTwin projec[1] the DTE marks a shift from centralized to distributed digital twin platforms. This change allows scientific communities to use distributed computing resources while keeping a strong link between physical and digital systems, which is vital for effective digital twins.

The main contributions of the DTE are: (1) a federated architecture that allows seamless integration of distributed computing and storage resources across various institutions, (2) standardized interfaces and protocols that support interoperability among different scientific fields, (3) a co-design approach that includes requirements from high-energy physics, radio astronomy, gravitational-wave astrophysics, climate research, and environmental monitoring, and (4) strong methods for assessing model quality, traceability, and uncertainty measurement in federated settings. These contributions tackle significant limitations in current digital twin platforms and offer the scientific community a scalable, interoperable base for future digital twins.

We validate the DTE by using real-world cases across various scientific fields. This shows its ability to support complex, multi-institutional research workflows. It also meets the performance and reliability standards needed for scientific applications.

The paper is organised as follows. Section 2 reviews related scientific Digital Twin initiatives and establishes the research gap. Section 3 introduces the diverse use cases that drove DTE development requirements. Section 4 presents the DTE architecture and core design principles. Section 5 details the system implementation and federated testbed deployment. Section 6 describes two representative use cases demonstrating DTE capabilities. Section 7 discusses interoperability with Destination Earth (DestinE). Section 8 concludes with contributions and future directions.

[1] https://intertwin.eu

2. Related work

2.1. Scientific digital twin initiatives

Several recent initiatives have established the foundation for scientific digital twins. The European Commission's Destination Earth (DestinE) program develops a high-precision digital twin of the Earth system for climate change adaptation and disaster risk management [3]. The Biodiversity Digital Twin (BioDT) project creates a prototype for biodiversity conservation, integrating various data sources and ecological models[4].[2]

The DT-GEO project develops Digital Twin Components (DTCs) for geophysical extremes as virtual labs for analyzing natural hazards in near real-time [5,6]. The European Digital Twin of the Ocean (EDITO) provides ocean knowledge through innovative visualization tools[3,4] while the UK's TWINE programme demonstrates digital twinning across ocean monitoring, climate projections, and flood forecasting.[5]

In healthcare, the European Virtual Human Twins Initiative advances personalized medicine through digital representations of human health states.[6] NASA's Earth System Digital Twin improves Earth system modeling[7,8] while the NSF FDT-BioTech program advances biomedical digital twins.[9] International collaboration includes the NSF-Japan partnership on Disaster Digital Twins for urban resilience.[10] Marine applications include DTOceanPlus and the Digital Twin Ocean Initiative.[11,12,13]

2.2. Digital twin platform initiatives

2.2.1. Research and development platforms

The DIGITbrain project[14],[15] involved 73 European partners developing "Digital Product Brain" (DPB) concepts with cognitive functions and lifecycle memory storage [7]. Using a "Manufacturing as a Service" model, it has facilitated 21 application experiments across many manufacturing sectors.

The Open Digital Twin Platform (ODTP) project provides an open-source framework for creating digital twins through modular architectures.[16] However, while ODTP demonstrates effective tools for digital twin creation, its architecture is not designed for federated module deployment or resource reutilization across multiple digital twin instances or platforms.

[2] https://biodt.eu/
[3] https://www.edito.eu/
[4] https://research-and-innovation.ec.europa.eu/funding/funding-opportunities/funding-programmes-and-open-calls/horizon-europe/eu-missions-horizon-europe/restore-our-ocean-and-waters/european-digital-twin-ocean-european-dto_en
[5] https://www.ukri.org/news/digital-twin-projects-to-transform-environmental-science/
[6] https://digital-strategy.ec.europa.eu/en/policies/virtual-human-twins
[7] https://esto.nasa.gov/earth-system-digital-twin/
[8] https://science.nasa.gov/biological-physical/why-does-the-world-and-nasa-need-digital-twins/
[9] https://new.nsf.gov/funding/opportunities/foundations-digital-twins-catalyzers-biomedical/nsf24-561/solicitation
[10] https://www.bu.edu/igs/research/projects/digital-twins/
[11] https://www.dtoceanplus.eu/
[12] https://ec.europa.eu/digital-ocean
[13] https://iliadproject.eu/
[14] Grant Agreement 952071
[15] https://www.digitbrain.eu/
[16] https://github.com/odtp-org/odtp

NIST has initiated a study to identify opportunities in measurement science and standards for Digital Twin systems across manufacturing, construction, smart Cities, Healthcare, and Energy.[17]

2.3. Technological approaches and architectures

2.3.1. Cognitive digital twins

Cognitive Digital Twins (CDTs) enhance traditional digital twins with AI functions and self-learning capabilities [8]. CDTs integrate data, information, and knowledge throughout system lifecycles, providing advanced representations using machine learning for predictive maintenance and remaining useful life estimation across manufacturing, construction, and healthcare [9].[18]

2.3.2. Data integration architectures

Modern data architectures address integration challenges through decentralized approaches. Data mesh transfers data ownership to domain-focused teams[19] while data fabric provides unified technological frameworks for consistent data views.[20] These architectures support digital twin implementations by enabling real-time data access and cross-domain integration.[21],[22]

Polystore systems address multi-database querying challenges [10]. The BigDAWG system demonstrates multiple storage engine integration [11], while CloudMdsQL enables parallel processing across distributed stores [12]. The HKPoly architecture uses knowledge graphs for distributed heterogeneous data queries [13]. The ESCAPE project provides insights on federated data management across astrophysics and particle physics.[23]

2.4. Commercial cloud platforms

2.4.1. Major cloud provider offerings

Major cloud providers offer dedicated digital twin platforms. MS Azure Digital Twins provides PaaS solutions using Digital Twins Definition Language (DTDL) with ecosystem integration.[24],[25] AWS IoT TwinMaker combines existing IoT and enterprise data, automatically generating knowledge graphs with 3D visualization and Grafana support.[26],[27],[28] Google Cloud Platform focuses on data analytics and machine learning services for digital twin projects.[29]

2.4.2. Differentiation from traditional cloud data platforms

Digital twin engines differ from traditional cloud platforms by providing specialized capabilities for bidirectional physical-virtual connections, real-time synchronization, and domain-specific workflows [14]. They incorporate physics-based simulation integration, temporal data management, and specialized modeling languages absent in general-purpose platforms [15].

2.5. Evolution and conceptual foundations

DTs have evolved significantly from Grieves and Vickers' original concept [16], now integrating real-time data with simulations and machine learning for monitoring, prediction, and optimization. Applications extend beyond traditional scientific domains into healthcare for personalized medicine [17], smart cities for utility monitoring, and manufacturing for predictive maintenance.

2.6. Comparative analysis and positioning

Current digital twin implementations fall into three categories: domain-specific scientific initiatives, general-purpose cloud platforms, and emerging architectural models. Domain-specific initiatives like DestinE and BioDT effectively address specific needs which are tailored to specific communities and domains. Commercial platforms offer scalability but face vendor lock-in and centralized designs conflicting with scientific research needs. Emerging architectures like data mesh and polystore systems address distributed data challenges but lack comprehensive scientific digital twin solutions.

This analysis reveals a critical gap: no existing platform adequately addresses scientific digital twin requirements including federated architecture, domain-agnostic design, HPC integration, FAIR data principles, and specialized scientific workflows. The interTwin Digital Twin Engine addresses these limitations through its federated, multi-scientific approach, providing a unified platform supporting multiple disciplines while maintaining domain-specific capabilities. Unlike existing solutions, the DTE offers an open-source, vendor-neutral platform embracing distributed scientific computing infrastructure, representing a fundamental shift toward scalable, interoperable scientific research foundations.

3. Use cases and design challenges

The interTwin project encompasses ten diverse use case scenarios that drove the development requirements for the Digital Twin Engine (DTE). These applications span environmental and climate science, high-energy physics, radio astronomy, and gravitational-wave astrophysics, each presenting unique computational, data management, and integration challenges. This section presents the use cases and analyses the common challenges that arise from their requirements, which inform the architectural design of DTE discussed in Section 4. Detailed descriptions of each use case can be found on the project website.[30]

3.1. Use case overview

3.1.1. Environmental and climate science applications

The environmental domain focuses on real-time monitoring and prediction systems for climate-related phenomena. Six applications address critical environmental challenges:

Climate Extremes and Weather Events: Generic detection of climate extremes uses CVAE-based anomaly detection. Tropical cyclone detection combines machine learning with deterministic tracking for climate projection analysis while wildfire prediction integrates satellite imagery with machine learning for global-scale burned area estimation.

Flood Adaptation and Early Warning: Two flood-related applications provide early warning systems and climate impact assessment for coastal and inland regions, incorporating real-time alert mechanisms and interactive scenario modelling.

Drought Monitoring: The Alpine drought early warning system employs surrogate models trained on hydrological simulations across seven

[17] https://www.nist.gov/news-events/news/2024/01/nist-launches-exploratory-digital-twins-study
[18] https://www.nibs.org/events/cognitive-digital-twins-roadmap-evolving-operations-and-maintenance-age-ai
[19] https://www.eckerson.com/articles/data-fabric-and-data-mesh-complementary-frameworks-for-a-unified-data-architecture
[20] https://www.precisely.com/blog/data-integrity/modern-data-architecture-data-mesh-and-data-fabric-101
[21] https://www.pwc.com/gx/en/issues/technology/tech-translated-data-mesh-data-fabric.{html}
[22] https://blog.purestorage.com/purely-educational/data-mesh-vs-data-fabric-whats-the-difference/
[23] https://projectescape.eu/
[24] https://learn.microsoft.com/en-us/azure/digital-twins/overview
[25] https://azure.microsoft.com/en-us/blog/azure-digital-twins-now-generally-available-create-iot-solutions-that-model-the-real-world/
[26] https://aws.amazon.com/iot-twinmaker/
[27] https://www.infoq.com/news/2022/05/aws-iot-twinmaker-ga/
[28] https://medium.com/globant/modeling-digital-twins-8b758dc4b4d6
[29] https://cloud.google.com/docs/get-started/aws-azure-gcp-service-comparison
[30] https://www.intertwin.eu

river basins, integrating ECMWF seasonal forecasts for predictive analysis.

3.1.2. Physics domain applications

Four physics applications demonstrate the DTE's capability to handle large-scale simulations and real-time data processing:

High Energy Physics: Lattice QCD simulations developed normalizing flows for quantum field theory studies. Fast particle detector simulations employ generative AI to create synthetic datasets, reducing computational overhead while maintaining accuracy compared to Monte Carlo workflows.

Radio Astronomy and Gravitational Wave Astrophysics: Radio astronomy noise simulation develops digital twins of telescope systems for training machine learning classification tools. Gravitational wave astrophysics creates digital twins of interferometers using GAN to simulate transient noise for real-time filtering applications.

3.2. Cross-domain challenges and requirements

Analysis of these diverse applications reveals four fundamental challenges that must be addressed by the DTE architecture:

3.2.1. Heterogeneous data integration

Digital twins require seamless integration of diverse data sources with varying formats, standards, and temporal characteristics. Environmental applications combine real-time sensor streams (climate stations, satellite data) with historical datasets (wildfire records, flood events) and model outputs (hydrological simulations, climate projections). Physics applications integrate observational data (gravitational wave strain, telescope signals) with simulation results (Monte Carlo calculations, detector responses) and auxiliary monitoring channels.

This heterogeneity creates significant interoperability challenges, particularly when adhering to FAIR (Findable, Accessible, Interoperable, Reusable) data principles. The challenge intensifies when integrating with external services like ECMWF forecasts or when collaborating across scientific domains with different data standards and metadata schemas.

3.2.2. Real-time processing and low-latency requirements

Several applications demand immediate data processing and rapid response capabilities with varying latency constraints. Environmental early warning systems require near real-time processing for flood alerts and drought monitoring, while physics applications need ultra-low latency (microseconds to milliseconds) for gravitational wave and radio astronomy experiments.

These requirements necessitate sophisticated middleware capable of handling event-driven workflows, dynamic resource allocation, and seamless integration between high-performance computing (HPC) and cloud infrastructures. The challenge is compounded by the need to maintain computational accuracy while meeting strict temporal constraints.

3.2.3. Dynamic scalability and resource management

The computational demands vary dramatically across applications and operational phases. Resource requirements range from computationally intensive batch processing (lattice QCD simulations, climate model training) to distributed real-time streaming (radio astronomy, gravitational wave monitoring) and geographically distributed processing (Alpine drought monitoring across seven basins).

Effective scalability requires dynamic resource provisioning that can adapt to workload variations, seamlessly transition between HPC and cloud environments, and optimize resource utilization while maintaining cost efficiency. This challenge is particularly acute for applications that experience unpredictable computational spikes, such as extreme weather event detection.

3.2.4. Domain-specific workflow requirements

Each scientific domain presents unique computational patterns and quality assurance needs. Physics applications emphasize high-precision computations, reproducible research workflows, and comprehensive data provenance tracking. Environmental applications prioritize user-friendly interfaces, scenario analysis capabilities, and integration with decision-support systems for policy makers and emergency responders.

These domain-specific requirements must be supported within a unified framework while maintaining interoperability and enabling cross-domain collaboration. The challenge lies in providing sufficient flexibility to accommodate specialized needs without compromising the coherence and maintainability of the overall system.

3.3. Derived architectural requirements

The challenges identified above translate into five core capabilities that the DTE must provide:

1. **Federated Data Integration**: Standardized interfaces and protocols supporting diverse data formats, real-time streams, and external service integration while ensuring FAIR compliance and cross-domain interoperability.
2. **Adaptive Workflow Orchestration**: Advanced orchestration capabilities supporting both batch and streaming workflows, with automated resource provisioning and standards-based workflow composition using standards like Common Workflow Language (CWL).
3. **Multi-Modal Processing Infrastructure**: Unified middleware supporting ultra-low latency requirements for physics applications and near real-time processing for environmental monitoring, with seamless HPC-cloud integration.
4. **Elastic Resource Management**: Container-based deployment with dynamic scaling capabilities, supporting diverse computational patterns from intensive simulations to distributed streaming applications while optimizing resource utilization.
5. **Comprehensive Quality Assurance**: Integrated validation, provenance tracking, and uncertainty quantification tools ensuring reproducible research and transparent operations across all scientific domains.

These requirements form the foundation for the DTE architecture presented in Section 4, which provides a modular, federated framework capable of supporting the diverse needs of scientific digital twin applications while promoting interoperability and collaboration across domains.

4. Requirements and architectural principles

This chapter offers a clear and detailed overview of the DTE architecture, outlining its essential components and discussing important security considerations.

4.1. High-level architecture

The DTE is designed to enable scientists and developers to create, deploy, and operate complex DTs in a federated environment. At a high level, its architecture outlines how the platform coordinates computing power, orchestrates data flows, and supports specialized thematic modules that address diverse scientific needs. Fig. 1 (adapted from the project blueprint [18]) provides a conceptual overview of the engine.

Two primary user roles have been defined for the platform: scientists (end users) and developers. While both interact with DTs, each group has different needs and objectives.

- **Scientists (End Users):** Focus on running existing DT applications and analysing results to derive insights. Typically, they do not modify low-level infrastructure but rather: launch or schedule simulations with minimal configuration; monitor and interpret outputs, either in

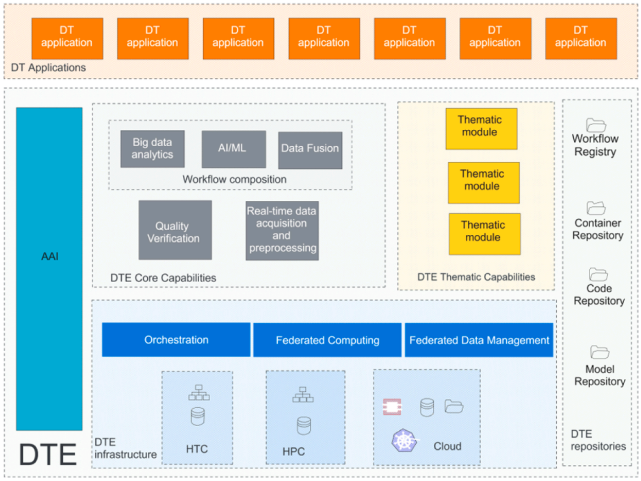

Fig. 1. High-level diagram of the DTE.

real time or offline; adjust basic model parameters as needed, without delving into complex setup details.

- **Developers:** Responsible for creating and maintaining DT applications, as well as the specialized modules that power them. Their tasks may include: integrating new data sources or external APIs; designing and optimising advanced workflows or simulation pipelines; extending analytic libraries or simulation models to meet domain-specific requirements; managing containers, code repositories, and overall platform configuration.

At the infrastructure level, a federated framework unifies computing and data resources – from HPC and High Throughput Computing (HTC) systems to cloud services – into a single operational model. This layer also integrates components for orchestration, federated computing, and federated data management, enabling the dynamic allocation of resources based on workload demands.

At the heart of the DTE, a series of core capabilities work together to streamline the creation and operation of DTs. These include *workflow composition*, which orchestrates complex data pipelines and computational tasks; *real-time data acquisition and preprocessing*, essential for integrating up-to-date sensor streams; and advanced *AI/ML* and *big data analytics* tools that derive insights from large, diverse datasets. To ensure results remain accurate and reliable, a *quality verification* mechanism continuously evaluates model performance and data integrity. Meanwhile, *data fusion* techniques unify information from varied sources into a consistent environment.

The DTE also includes domain-specific components – referred to as *thematic modules* – which target particular scientific disciplines. These modules may comprise complete toolsets, specialised models, or dedicated libraries that ensure a DT operates effectively within its specific field. For instance, an environmental monitoring module could integrate climate models and geospatial analytics, while a high-energy physics module might bundle simulation frameworks tailored to particle collisions. By offering these focused add-ons, the DTE accommodates a broad range of scientific challenges and fosters deeper customisation for specialised use cases.

Beyond its core capabilities and thematic modules, the DTE provides dedicated repositories for storing and managing the various artifacts essential for building and running DT applications. These repositories implement a mandatory open licensing framework prioritizing Creative Commons-compatible licenses for shared artifacts. These artifacts include container images, executable programs, domain-specific libraries, and configuration files that can be shared or reused across multiple projects. By centralizing these resources, the platform streamlines the process of assembling new DTs, ensures version control, and promotes consistent deployment practices throughout different scientific domains.

The DTE employs a robust Authentication and Authorization Infrastructure (AAI) based on industry-recognized protocols such as OpenID Connect (OIDC) and OAuth 2.0. This framework handles user identification, ensures appropriate access rights, and enforces security policies across different services within the platform.

Each DT application interacts with these components through well-defined APIs or specialised interfaces that expose the platform's capabilities. This design allows applications to seamlessly integrate advanced data processing, orchestration services, and security features without directly handling the underlying infrastructure details. As a result, developers can focus on domain-specific logic while relying on a consistent, standardised framework provided by the DTE.

4.2. DTE infrastructure

The DTE addresses the challenge of unifying computing resources and data management across diverse scientific communities, while remaining flexible enough to support specialized requirements. At a broad level, the platform is organized around three main pillars–**Federated Computation**, **Federated Data Management**, and **Intelligent Resource Orchestration**–forming a cohesive ecosystem that can accommodate a wide range of DT applications.

Federated Computation. A key goal is to provide seamless access to computing power, whether it comes from commercial Cloud services, HPC clusters, or HTC systems. To achieve this, the DTE relies on a federated compute framework that abstracts away provider-specific details, allowing developers and scientists to tap into the most suitable type of resources (i.e. local Cloud vs possibly remote HPC/HTC) without dealing with incompatibility or workflow issues. This approach fosters scalability and responsiveness, letting DT applications scale up or down in real time based on actual demand.

Federated Data Management. Because most DT use cases involve substantial amounts of data, the platform includes robust data services aligned with its federated compute architecture. Using a unified data model and advanced federation technologies, the DTE integrates historical data, streams and external datasets into a shared *Data Lake*. This design addresses differences in data formats and protocols, ensuring that applications can securely locate, retrieve, and manipulate the information they need, regardless of its physical location.

Intelligent Resource Orchestration. An intelligent orchestration layer leverages Machine Learning (ML) and predictive analytics to monitor Cloud resource usage and adjust resource allocation.

4.3. Core capabilities

The core capabilities enable the creation and operation of DTs supporting diverse scientific domains. Building on the federated infrastructure, they provide essential services such as workflow composition, real-time data handling, and AI/ML. They also include quality checks to keep simulations accurate and consistent.

Workflow Composition and Management. Researchers can design, schedule, and track workflows without having to configure each environment manually. The platform connects multiple data sources, databases, and sensor feeds–into orchestrated pipelines that can run on HPC, HTC, or cloud resources. This automation allows scientists to focus on extracting insights rather than handling infrastructure details.

Real-Time Processing. The real-time data acquisition and processing framework supports event-triggered execution of workflow engines, detecting when new data that requires processing is made available. It performs data staging and pre-processing (e.g. to perform data cleansing or data quality assessment) and delegates the complex data processing to external workflow management systems which are in charge of executing applications on resources that can be dynamically provisioned from a Cloud-based infrastructure.

Machine Learning and Predictive Analytics. The AI/ML subsystem focuses on developing data-driven models for DTs. This subsystem is primarily concerned with training and deploying ML models, which enhance DTs' capabilities with advanced data insights. The characteristics of ML/AI subsystem includes model training, Hyper Parameter Optimization (HPO) and inference.

Quality Verification. A dedicated process ensures that the outputs of the DTs remain reliable and consistent. It checks data integrity, assesses model performance, and ensures workflows stay aligned with real-world conditions.

4.4. Security

The platform employs an AAI that governs who can access data or run specific workflows. By using industry-standard protocols, the DTE enforces detailed permissions for launching simulations, retrieving datasets, or performing sensitive tasks.

5. DTE system design and implementation

The DTE forms the backbone of the interTwin project, supporting the integration and operation of complex DT applications across various scientific domains. This section presents the system design and implementation approach following a progression from requirements through technology selection to deployment results. The section begins by establishing the implementation requirements for federated computing, data management, and workflow orchestration. It then details the technology selection rationale and specific implementation choices made for each core component, including authentication infrastructure, compute federation, data lake management, AI orchestration, workflow composition, real-time processing, and machine learning support. Finally, Section 5.8 presents the practical deployment results through the DTE testbed, demonstrating the integration of multiple European computing centers. The new developments performed in the project are available in the interTwin organization[31] in GitHub.

5.1. Authentication and authorization infrastructure

Security is a fundamental concern in any distributed computing environment, and the DTE infrastructure includes a comprehensive security framework to protect data and resources. The security framework is built around a federated identity management system that provides authentication and authorization services for users and applications accessing the DTE. The project has adopted the EGI Check-in[32] service based on Keycloak and, in particular, OIDC and OAuth 2.0.

In addition, a new service named the Account LInking SErvice (ALISE), has been developed that provides a layer where identities, enrolment, group membership and other attributes and authorization policies on distributed resources can be managed in an homogeneous way. These activities may be achieved without requiring admin intervention: users of a facility will typically follow an enrol process once. Various services, typically local to a facility, may then query ALISE to map a user's federated identity to their corresponding facility-local identity. This in turn allows the services to process such requests while honouring site policies on authorisation, accounting and traceability.

5.2. Federated compute resources

The main purpose of the compute federation is to enable seamlessly integration of highly heterogeneous and disparate providers such as Clouds, world-class HPC and HTC centers. In order to achieve this objective the strategy defined was to implement a model based on transparent payload offloading. Given a workflow (pipeline) the goal is to cherry-pick a step and execute it over the most suitable type of computing resource available within the federation. As an example, assuming a multi-step workflow managed by cloud-native framework, this could benefit from executing specific steps, such as a GPU-accelerated statistical data analysis, on GPU equipped nodes available on an Exascale EuroHPC centre. The cherry-picking mechanism is based on Kubernetes (K8s) native match-making feature. From the start of the project, one of the goal that we set was transparency for the end user and for the target provider. To implement such a model, we decided to rely on a de facto Cloud standard API like K8s and to define a lightweight solution to exploit a heterogeneous provider exposing the very same experience of running a pod on the Cloud resources. As a result, we developed interLink.[33] interLink provides an abstraction for the execution of a K8s pod on any remote resource capable of managing a container execution lifecycle. From a technical perspective the interLink project is based on the Virtual Kubelet (VK) interface, a project in the CNCF Sandbox program that creates virtual K8s nodes capable of managing the payload execution in a custom fashion, therefore abstracting the actual container execution from the native API layer offered by K8s itself. The primary scenario for VK is to allow K8s to interact with serverless container platforms such as Azure Container Instances and AWS Fargate. Serverless platforms allow users to run containers without having to manage the underlying infrastructure. interLink project consists of two main components:

- A K8s Virtual Node: based on the VK technology. It translates requests for a K8s pod execution into a remote call to the interLink API server.
- The interLink API server: a modular and pluggable REST server where to create specific interfaces to dedicated resources, or simply to use the existing ones. The plugins currently available are: SLURM; HTCondor, UNICORE, Kueue and Docker

VK features a pluggable architecture integrated with K8s primitives, which makes it fully compatible with any workflow based on that platform. The latter is extremely valuable, as the overall architecture is expected to support a wide range of scientific and non scientific use cases. The key feature of the VK is to masquerade as a K8s Kubelet which enables K8s to be connected to other APIs. With the use of an interTwin API layer, deployed at the edge of any resource provider, we can transparently extend a K8s cluster running on a Cloud system to any remote resource, being either cloud or batch based. As such, the interLink layer represents the actual mean of the compute federation. The user application only access K8s and the related APIs while, at the same time, the target site is unaware of the K8s cluster and does not need to interact with it. In summary, the proposed solution follows a plug-and-play approach. The interLink API services comprises several independent processes that communicate via REST interfaces. The process exposed to the K8s cluster is an OAuth2 proxy that verifies incoming requests against an OIDC identity provider before forwarding the request to the final stage of the translation process. This component is the one that gives the name to interLink because it is responsible for the final request retouches that guarantee uniformity for all the plugins that are going to be contacted

[31] https://github.com/interTwin-eu
[32] https://www.egi.eu/service/check-in/
[33] https://interlink-project.dev

on an HTTP call. As stated above, the interLink API server is based on a plugin model and this represents a key feature in order to build a flexible model. First of all, each plugin is independent and separately talks to the interLink layer which translate the request and execute the actual job, or set of actions needed to manage the execution, on the provider. A plugin represents the only piece of the system where the backend specific configuration will be implemented. As a consequence, if a site has specific needs, custom modules can be implemented and configured in the site itself without affecting the overall architecture and implementation.

5.3. Federated data management and the data lake

When considering a distributed environment, where DTs workflows may take place at different geographical locations, easy and effective management of data becomes an important consideration. The data needed for training the models part of DTs in interTwin is varied in nature, including scientific datasets, real-time sensor data and simulation outputs. As a consequence, any common solution will need to be flexible enough to support these different data types.

The data models training takes considerable computing resources, often taking advantage of the benefits from using GPUs. In interTwin, these activities take place within HPC facilities. Therefore, one of the key goals is to allow easy data ingress into and egress from HPC facilities.

As inspiration, the project took the Data Lake concept from the ESCAPE[34] project as a starting point. This storage concept, originally coming from the High-Energy Particle Physics (HEP) community, includes Rucio[35] as a central component that allows for the management of large volumes of data, File Transfer Service (FTS)[36] as a service that manages, at scale, the transfer of individual files and various storage endpoints that offer one of the supported protocols. The data is made available to HPC worker nodes through the standard POSIX interface; i.e., a normal mounted filesystem. This allows software to load training data using standard I/O operations and without linking against specialist data access libraries. Such POSIX-based interactions are a common way for HPC jobs to accept data and HPC facilities provide such a distributed filesystem. Some HPC centres already offer a storage solution compatible with the interTwin data lake. For other HPC centres, a solution that allowed data transfers was needed: an edge service that follows the Data Lake model. An important requirement is that this edge service must integrate with the existing storage solutions without requiring that the storage is modified and while honouring the file system permissions. A new storage solution was also developed: teapot.[37] Unlike other solutions, teapot enables user-specific access to existing storage while honouring the file system's permissions. All file operations are undertaken with the authenticated user's identity at that facility. The interTwin data lake has been deployed by federating five different storage technologies (Ceph S3, dCache, Teapot, Onedata S3, StoRM WebDAV) from ten data centres in eight countries (DESY, CESGA, INFN, EODC, Cyfronet, DZA, Uni Vilnius, KBFI, Julich, PSNC).

5.4. AI orchestrator

The AI orchestrator is based on the INDIGO - PaaS Orchestrator component[38] the core component of the INDIGO PaaS layer. It collects high-level deployment requests and translates them into actions to coordinate resources interacting with the underlying cloud infrastructures. It allows the provisioning of virtualized compute and storage resources on different Cloud Management Frameworks like OpenStack, OpenNebula, AWS, MS Azure, Google Cloud, etc., by using the Infrastructure Manager [19]. The PaaS orchestrator features advanced federation and scheduling capabilities. It ensures transparent access to heterogeneous cloud environments and the selection of resource providers based on criteria like user's SLAs, services availability, special hardware availability and data location. It manages deployment requests, expressed through templates written in TOSCA, the standard language for describing application topologies in cloud, and coordinates the deployment on the most suitable cloud site. To achieve this it gathers SLAs, monitoring data and additional information from other platform services and it asks the cloud provider ranker for a list of the best cloud sites. The new ranker uses a proper set of metrics and AI algorithms, to provide the Orchestrator with a list of ranked providers that aims to minimize deployment errors and the time required to create a deployment. A dedicated component takes care of two main actions: to training ML models and storing them in an MLflow registry and to perform inference using the trained models retrieved from the registry. Currently a classification model is used for predicting the success or failure of a deployment while a regression model is used for estimating deployment creation or failure time. The inference outcome is used to make predictions for a give cloud provider

5.5. Workflow composition

A critical feature of the DTE is its advanced workflow orchestration and management capabilities. The DTE supports the composition, execution, and monitoring of complex scientific workflows that can span multiple computational environments.

By using the Common Workflow Language (CWL) as the standard for workflow definition, the DTE ensures interoperability with existing workflow management systems. The project has extended the Ophidia module to support the standard through its Python bindings [20]. The Ophidia framework is an open-source solution for the analysis of scientific multi-dimensional data, joining HPC paradigms and Big Data approaches [21,22]. It provides an environment targeting High Performance Data Analytics through parallel and in-memory data processing, data-driven task scheduling and server-side analysis. The framework supports the execution of complex analytics workflows in the form of Directed Acyclic Graphs (DAGs) of Ophidia operators [23].

In addition, for specific Earth Observation use cases, openEO[39] application programming interface (API) has been selected. openEO is an API that supports i) the management of workflows, ii) job handling, and iii) linking to data sources and processing capabilities on compatible cloud platform providers in a standardised way. openEO has been also extended to support execution of containerized software packages as execution of specific processes following the OGC API processes approach.

5.5.1. Provenance in workflow and AI

Workflows and provenance are two faces of the same medal. Tracking provenance in scientific workflows has twofold benefits: (i) it enables a better understanding and reproducibility of the results of a computational process, and (ii) it fosters trust, transparency and interpretability, by documenting in detail how a specific output has been generated. In this respect, the interTwin project has delivered a fully-fledged provenance solution supporting provenance management (yProv service [24]), exploration (yProvExplorer[40]) and tracking in (i) workflows [25], (ii) software quality assurance [26] and (iii) AI training processes [27]. The integrated ecosystem approach has enabled the implementation of an end-to-end traceability solution in the interTwin DTs, exploiting the yProv4WFs library to generate provenance documents at runtime during the DT workflow execution (as in the case of openEO [28]) and the yProv4ML library to address provenance tracking in the training phase of an AI model. Provenance documents have been persistently managed by the yProv service, which exposes a CRUD API for consumer applications (i.e., yProvExplorer). From an interoperability perspective, the

[34] https://projectescape.eu/
[35] https://www.intertwin.eu/article/infrastructure-component-rucio
[36] https://www.intertwin.eu/article/infrastructure-component-fts3
[37] https://www.intertwin.eu/article/infrastructure-component-teapot
[38] https://github.com/indigo-dc/orchestrator
[39] https://www.intertwin.eu/article/core-dte-module-openeo/
[40] https://explorer.yprov.disi.unitn.it/

yProv components leverage the W3C PROV family of standards, PROV-JSON serialization and RESTful interfaces. The yProv ecosystem supports the multi-level provenance implementation. It enables scientists to navigate within the provenance space across different dimensions (e.g., horizontal & vertical), which means both over computational tasks and across different levels of granularity. Although the project's main case studies have come from the climate change domain, the proposed solution is domain-agnostic having also been applied to interTwin DTs in earth observation and high-energy physics domains [29].

5.5.2. Quality assurance

To ensure the reliability and accuracy of DT applications, the DTE incorporates Quality Assurance (QA) mechanisms into workflow compositions. These QA components validate the results of simulations and analyses by comparing them against established benchmarks and known data. The model validation architecture relies on the usage of the Software/Service Quality Assurance as a Service (SQAaaS) [26] platform.

The SQAaaS is a platform for quality assessment and awarding of multiple digital objects (source code, services, data). Data quality assurance is technically a challenging process which needs to involve community-specific aspects such as data integrity, accuracy completeness or consistency. The SQAaaS platform provides researchers with ready-to use CI/CD pipelines that cross-check the relevant quality criteria of any software project. In intertwin the SQAaaS platform has been expanded to embed data quality assessment. The platform can programmatically incorporate quality criteria in the CI/CD framework and produce automated data verification flows analogous to those we will have for the software in git repositories.

In order to do so, Data as Code principles have been applied to the development of CI/CD pipelines to evaluate metadata conformance and FAIR principles covering all the data quality dimensions. In particular DataOps technologies are applied to support data pipelines execution. Our approach relies on existing/proven standards for metadata (digital objects) and description workflows with data provenance support. For instance, we build on the pyOphidia capability to automatically tag workflows with metadata and plug it to the SQAaaS library in an analogous way as to how software is evaluated, and embedded in the DTE architecture.

The QA module high level capabilities includes:

- Automated testing of workflow components to detect errors and inconsistencies.
- Integration with validation datasets to ensure that workflow outputs meet the required standards.
- Continuous monitoring of workflows to track their performance and detect anomalies.

We implemented two possibilities in terms of technology to make possible the custom assessments needed by DT developers to perform model validation: triggering customs assessments from GitHub, relying on GitHub actions, and alternatively, triggering them from a workflow step document. This strategy allows embedding from the start in the development process the data quality checks. Therefore we follow general trends towards developing services producing FAIR data by design, which implies embedding the FAIR perspective at early enough stages of the development processes, so that the digital objects inherit properties related to FAIR such as reproducibility, provenance, etc.

The validation process is further automated by exploiting git actions technology. We make available two GitHub actions that enable the automated assessment of source code, including workflow and model code, by triggering the SQAaaS platform. A summary containing the quality criteria being analysed is provided, and, in the event that a certain level of these criteria has been fulfilled, the corresponding digital badge that recognizes those achievements. As a complement, the step actionadds the capability to define customized steps as part of the evaluation of a quality criterion within the SQAaaS source code assessment. This is required, for instance, for the testing criteria, where diverse testing frameworks might be used (e.g., Python pytest).

5.6. Real-time data processing

The ability to process real-time data is a key requirement for many DT applications, especially those in fields such as environmental monitoring. The DTE infrastructure includes a robust real-time data processing and streaming framework that allows DTs to continuously ingest, analyse, and respond to incoming data streams.

This real-time processing capability is made possible through event-driven architectures and streaming data pipelines, which enable the DTE to react to changes in data as they occur. For example, in a climate simulation, the DTE can process sensor data from weather stations in real-time, adjusting the simulation parameters as new data becomes available.

The serverless event-processing system is in charge of receiving data pre-processing requests from the event-ingestion system to perform additional data transformations that may not be performed within the event-ingestion system itself. This can be due to a lack of support for certain operations or the dependency on external tools that may be packaged as Docker images, which may not be able to run directly within the event-ingestion system.

To address the challenges and limitations of the event-ingestion systems, we adopted the OSCAR [30] serverless event-driven processing platform. OSCAR is deployed on top of elastic Kubernetes clusters, provisioned via the Infrastructure Manager (IM), and provides efficient execution of data-processing requests by executing user-defined scripts in dynamically provisioned containers that are triggered in response to events. In addition, it supports low-latency synchronous requests through Knative, the offloading of the workload to HPC through interLink, and the seamless execution of JupyterHub environments, ensuring the development and operation of DTs.

OSCAR supports several storage providers like MinIO, Amazon S3, OneData, dCache, and WebDAV storage providers (e.g. NextCloud). In addition, support for Rucio events allows OSCAR to process data stored in the data lake.

The event-ingestion system is responsible for receiving the notification events from the file/object-storage system and provides the ability to execute simple transformation data flows using the built-in components supported by the system. Apache NiFi is employed to create versatile data flows that enrich and route data from diverse sources, effectively decoupling file uploads from data processing. The event-ingestion system is integrated with multiple sources, including Amazon S3 and dCache, enhancing its capabilities. Additionally, Apache Kafka is used for high-throughput event streaming, buffering data for processing, and triggering OSCAR services in near real-time.

This multi-source approach not only diversifies the data intake but also increases the flexibility of the platform, handling a wide range of use cases and adapt to various data environments. DCNiOS[41] facilitates the deployment of dataflows to achieve integration between a source of events like a storage system such as Kafka or dCache, and OSCAR Services. This tool allows users to set up dataflows using simple YAML configuration files. These files detail data sources and destination endpoints together with intermediate steps processes. DCNiOS also comes with a CLI. This feature enables us to deploy and adjust the NiFi flows at runtime, for example, by changing the data processing rate, making our system adaptable to varying needs.

[41] https://github.com/intertwin-eu/dcnios

5.7. AI and machine learning support within the DTE

`itwinai`[42,43] is an open-source Python library that streamlines the deployment and scaling of ML workflows for scientific applications, with a particular focus on DTs. DTs rely heavily on AI and ML to enable advanced analysis, predictive modelling, and decision-making capabilities, often necessitating seamless integration with HPC resources. `itwinai` addresses this need by providing a user-friendly toolkit that abstracts much of the complexity associated with deploying and managing ML workflows on large-scale computing infrastructures. By employing a configuration-based approach, users can define, execute, and manage modular workflows that include tasks such as data preprocessing, distributed training, HPO, and inference, all while maintaining compatibility with widely-used ML frameworks like PyTorch and TensorFlow.

A distinguishing feature of `itwinai` is its support for distributed ML training, allowing researchers to scale their models across multiple GPUs or nodes without extensive code modifications. This functionality is powered by industry standard backends such as PyTorch Distributed Data Parallel (DDP), TensorFlow distributed strategies, and Horovod, ensuring efficient utilisation of HPC resources. Furthermore, `itwinai` includes advanced HPO capabilities, facilitated by Ray Tune, which allow users to explore large parameter spaces systematically and improve model performance with minimal manual intervention. These features are complemented by integration with popular ML logging tools, including MLflow, Weights&Biases, and TensorBoard, to provide experiment tracking and visualisation.

Beyond its technical capabilities, `itwinai` is designed to empower domain experts, such as scientists and engineers, to independently deploy and scale AI solutions. This reduces the need for specialised ML engineers, enabling researchers to focus on advancing their scientific objectives. The library also includes the possibility to connect to a ML model registry, allowing to store, version, and reuse trained models, thereby fostering reproducibility and efficiency in AI-driven research workflows. Furthermore, the extensible architecture of `itwinai` supports the integration of third-party plugins, allowing developers to tailor the toolkit to the unique requirements of specific scientific domains or applications.

In the context of DTs, `itwinai` aims at enabling scalable, AI-driven research while minimising the engineering overhead associated with HPC integration. By bridging the gap between domain expertise and scalable ML workflows, `itwinai` facilitates the development of robust and efficient AI solutions, addressing the challenges inherent in managing large-scale distributed computational resources for scientific DT applications.

5.7.1. Big data analytics

The Big Data Analytics deployment layer provides a set of topology templates and recipes for general-purpose data analytic environments to be deployed on demand on top of the cloud resources.

The cloud topology templates have been created using the TOSCA standard specification. They describe the virtualized resources and the software components required to deploy the final application. Furthermore, they provide the user with a set of input parameters, enabling them to customise the application configuration.

This layer enables the users to access the needed set of tools in a reproducible manner, in minutes, with the ability to deploy the amount of resources needed to process their data, and grow or shrink the resources if the initial set of resources was not correctly estimated, thanks to the capabilities provided by the IM.

[42] https://itwinai.readthedocs.io
[43] https://github.com/interTwin-eu/itwinai

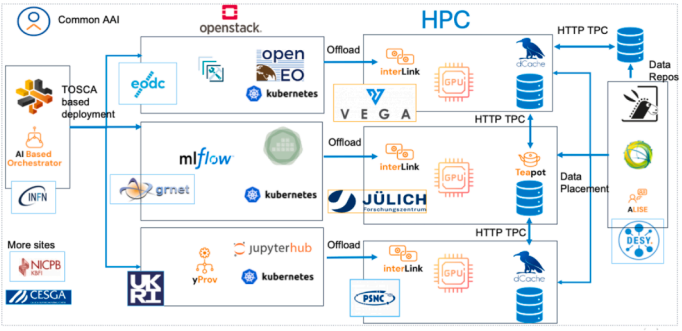

Fig. 2. DTE testbed showing the federated infrastructure with distributed HPC and Cloud sites.

5.8. DTE testbed

The practical implementation of the DTE is materialized through a federated infrastructure that integrates multiple specialized European computing centers, as illustrated in Fig. 2. The deployed architecture includes high-performance HPC sites (EuroHPC VEGA in Slovenia, JÜLICH in Germany, and PSNC in Poland) that provide supercomputing capabilities with advanced GPU architectures, integrated via interLink to High Level services. The AI-based orchestrator dynamically optimizes resource allocation over Clouds (GRNET, UKRI and EODC Openstack) where Kubernetes clusters are deployed to execute containerized workloads, while data management federates repositories with intelligent data placement optimized by computational proximity. The ecosystem integrates specialized tools such as MLflow for ML model management, JupyterHub for interactive development, all unified under a Common AAI authentication system. Other sites part of the testbed are INFN, CESGA, and DESY.

6. Use cases functionalities and integrations with DTE

This section highlights two specific DTs from environmental and physics domain and details their implementation and integration with the DTE.

6.1. Gravitational waves detection and noise simulation

The Virgo DT aims at realistically simulate transient noise artifacts, called *glitches*, which appear in the main observation channel of the detector, termed *strain* channel. In order to achieve this goal, the GNN needs to learn the non-linear transfer function to map the glitches present in a subset of control channels, which are uncorrelated to the astrophysical signal, to the strain channel. The control channels are called auxiliary channels, and contain time series measurements of the interferometer control systems and environmental sensors, tracking conditions such as seismic activity, acoustic noise, and electromagnetic interference. Since the data used as input for the model does not contain any information on gravitational waves by design, any transient signal present in the simulated strain output can be identified as a glitch. The DT is organised as a pipeline operating in quasi real-time, with the goal of identifying a glitch in the incoming data and passing the information to downstream low-latency pipelines that search for transient astrophysical signals. The expression low-latency, in this context, indicates a lag in time of the order of tens of seconds. In the first phase, the information will be passed as a veto decision not to process the data if they contain a glitch. In a second phase, also in view of future detectors such as the Einstein Telescope, the DT will output data in which the glitch has been removed, to be further processed by the search pipelines. The removal of the glitches from the strain channel is referred to as de-noising and it consists in subtracting the generated strain data from the real one. The

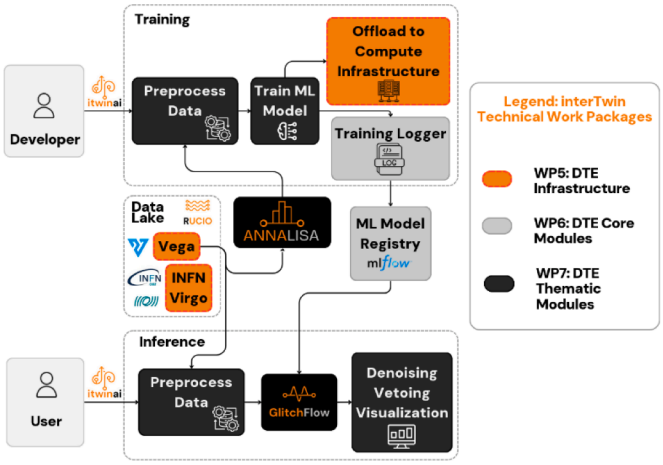

Fig. 3. A schematic representation of the training and inference subsystems which make up the DT. The diagram shows the involvement of the developer and the user in the different parts of the workflow, as well as the flow of data among the different modules. Both the vetoing and de-noising pipelines have similar structures, the main difference being the final module in the inference subsystem.

high accuracy of the generated data will ensure that only noise artifacts are removed.

The implementation of the DT consists of a train and inference subsystem, sketched in Fig. 3.

The three main modules which make up the training subsystem are `ANNALISA` (Advanced Non-linear transient-Noise Analyser of Laser Interferometer Sensor Arrays), `PreprocessAPI`, and `GlitchFlow`. The inference subsystem, on the other hand, comprises the same `PreprocessAPI` module, so that the inputs are processed in the same way as the training data, and a `GenerativeAPI` module which carries out the inference.

`ANNALISA` is a tool for identifying the relevant auxiliary channels which the GNN will use as input, i.e. those which do not contain any astrophysical information. It makes use of time-frequency domain analysis of the data, namely the Q-transform [31],to evaluate correlations among the main and auxiliary channels. This is achieved by counting the number of temporally coincident spikes in the energetic content of the signals above a critical threshold and dividing it by the total number of spikes in the main channel. The current version of `ANNALISA` employs a PyTorch-implemented [32] Q-transform which was developed in order to run the whole analysis on GPU. Our version of the Q-transform is equivalent to the one implemented in the standard package GWPy [33] up to some border effects which can be easily eliminated after the transformation; this development makes it possible to speed up the correlation analysis by two orders of magnitude. `PreprocessAPI` is used for data preprocessing and dataset creation, while `GlitchFlow` is the module which contains the GNN for generating the glitches. The current GNN model architecture is a U-Net [34] inspired encoder-decoder that also incorporates attention gates and residual blocks.

The training subsystem is maintained and operated by a DT developer, who performs the data pre-processing and the training of the model. The DT developer monitors the operations of the DT by using a monitoring system that collects and displays metrics on training convergence and inference accuracy. Once the training is over, the model is passed to the Training Logger and then stored in the Model Registry. The training of the model is repeated periodically at regular intervals, for example every month, or when there have been significant changes to the state of the detector. The most clear example of such changes, albeit not the only possible one, is the necessary recalibration at every new observation run, during which the auxiliary channels can undergo significant modifications and the background noise is expected to be different. After the initial set-up, the role of the developer can be performed by an automated procedure.

The DT user shown in Fig. 3 is the person using the pipeline for low-latency data analysis; they can process the data with the same `PreprocessAPI` and then pass it to the `GenerativeAPI`, which calls the most recent pre-trained model in the Model Registry and uses it to perform the inference, i.e. fast generation of glitches. The last module of the Inference subsystem is either the vetoing or de-noising one, depending on the operation being performed.

All modules within both subsystems are implemented as *itwinai* plugins. Itwinai is a DTE core module that offers several key features that are beneficial to the DT, including distributed training capabilities, a robust logging and model catalogue system, enhanced code reusability, and a user-friendly configuration interface for pipelines.

The current accuracy for the de-noising pipeline is over 90 % for a Signal-to-noise-Ratio (SNR) of 6, which represent a realistic lower bound for the glitches seen in the Virgo detector [35].

6.2. Tropical cyclones (TCs) detection and wildfires prediction on climate projections

In recent years, climate change has been leading to an exacerbation of extreme events, including tropical storms and wildfires, raising major concerns in terms of their increase of their intensity, frequency and duration as found by [36] and [37].

Advances in ML can provide cutting-edge modelling techniques to deal with extreme events detection and prediction tasks, offering cost-effective and fast-computing approaches. Solutions based on ML could support study and analysis of such events, providing scientists and policy makers with innovative data-driven tools. However, from an infrastructural point of view, supporting these applications requires multiple integrated software components including data gathering, pre-processing and augmentation pipelines, computing platforms for model training, results visualization tools, etc.

In particular, DT applications for the analysis of extreme events, focusing on: (i) detection and tracking of TCs and (ii) prediction of wildfire on a global scale in terms of extent of burned areas, are being developed relying on ML models as their core components. Different types of Deep Neural Networks (DNNs) models are being adopted as modelling tools for learning the mapping between environmental drivers and occurrences from past data and generalizing it to future projection data. The two DTs applications on TCs and wildfires are supported, respectively, by the `ML TC detection` and `ML4Fires` thematic modules.

The DT application on TCs relies on a “hybrid” ML approach that links a data-driven model, which detects and localizes TC centers, with a deterministic tracker [38]. ML models, such as VGG-like Convolutional Neural Networks (CNNs) [39] and Graph CNNs, are used to learn the non-linear relationships between input weather fields and TC occurrence in large climate datasets. The models are trained with ERA5 reanalysis data [40] joined with observed TC records from the International Best Track Archive for Climate Stewardship (IBTrACS) [41].

The second DT application is related to wildfires. It exploits the U-Net++ model [42], based on CNNs, for learning the relationships between different weather and vegetation variables for predicting wildfires occurrences in terms of burned areas on historical data. The SeasFire Cube dataset [43], a multivariate harmonised dataset designed for seasonal wildfires modelling, is used for training the ML model.

The trained models, from both the applications, can then be applied for detecting TCs and predicting wildfires under future climate scenarios, exploiting CMIP6 data (e.g., from ScenarioMIP [44] or HighResMIP [45] projects), available from the Earth System Grid Federation [46].

Although the two DTs have different scientific goals and exploit diverse data and ML model architectures, from a high-level perspective their workflows are similar. Fig. 4 provides an overview of the steps envisioned by the two DT applications.

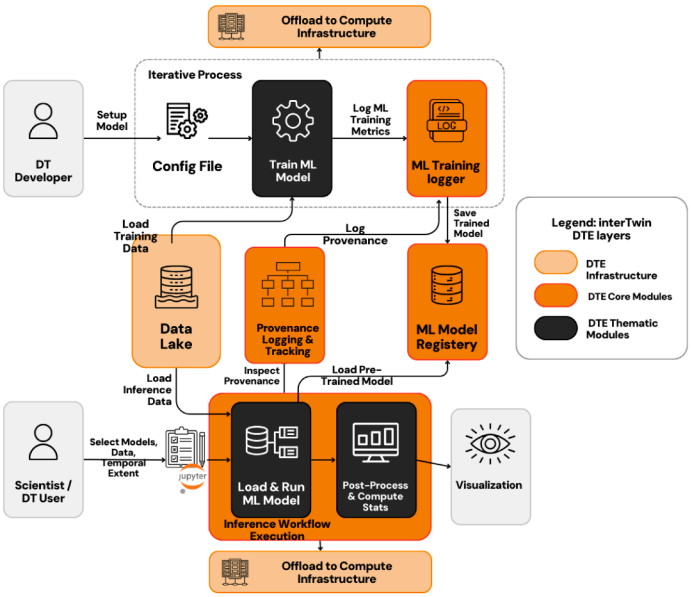

Fig. 4. High-level design of the workflows of the DT applications for tropical cyclones detection and wildfires prediction on climate scenarios. The DT developer and DT user workflows are depicted in the figure, as well as the links with the different component from the interTwin DTE.

Two distinct workflows are supported: the first for users with technical expertise that need to configure and train a ML models for extreme events detection and prediction (i.e., the DT developers), and the second one for end-users that need to apply the ML models for analysing changes in the events on climate data (i.e., DT users).

In the former, the DT developer exploits the thematic components (i.e., `ML TC detection` or `ML4Fires`) for building a new ML model using a pre-defined setup. During the training stage metrics and provenance are tracked using the DTE core modules (i.e., *itwinai* and *yProv*). Once the training is completed, the resulting model can be stored on a ML model registry. As the training process requires GPUs to be carried out efficiently, the whole workflow (e.g., training, testing and evaluation) can be offloaded on HPC machines using *interlink* to deploy software containers. Docker images, including the DTE core frameworks and libraries for climate data processing, are provided for supporting the execution of the DT applications in a portable way. Such images are then translated into Singularity images in order to be deployed on HPC infrastructures. In particular, a testbed has been implemented in the context of the interTwin project allowing users to specify their images from a JupyterHub interface which can be transparently deployed via interLink (following Section 5.2) on multiple HPC infrastructures, such as the Vega EuroHPC cluster.

In the second workflows, the DT user can select a pre-trained ML model from the registry for running analysis on extreme events. Jupyter Notebooks are provided as main tools for interactive analysis and visualization using the pipelines provided by the thematic software modules. Workflows based on *Ophidia* can be used as part of the pipelines (e.g., pre-processing). Also in this case, the containers for running the notebooks can be offloaded on a Cloud or HPC machine.

For both workflows, training and inference data is accessible from the interTwin Data Lake based on *Rucio*.

6.3. Non ML-based digital twins application

The DTE supports the development of non ML-based DT Applications, based on physics-based models such as the Post-Flood Analysis in coastal regions whose description is available on the interTwin website[44]

7. Interoperability with destination earth (DestinE)

A cornerstone of the interTwin project is its emphasis on interoperability, crucial for the seamless integration and collaboration between different DTs initiatives across the globe. Although interoperability as a whole is desirable, realistically interTwin focuses on the interoperability with the important EC initiative of DestinE for which there will be some examples in the following sections.

The interTwin architecture is designed with a strong focus on standards compliance, ensuring that its components and interfaces adhere to internationally recognized standards for data exchange, security, and communication. By aligning with standards such as TOSCA and CWL, interTwin ensures that its DTs can interact seamlessly with other systems, regardless of their underlying technology platforms. This also allows multiple digital twins deployed on DTE to access the same curated datasets through standardized APIs, rather than duplicating ingestion pipelines. For example, if two DTs require Sentinel-2 imagery, both can query the federated catalog and retrieve harmonized EO products

A commitment to open-source development underpins the interTwin project's approach to interoperability. By making key components of its architecture open source, interTwin encourages community contributions and the development of complementary tools and extensions. This openness fosters a vibrant ecosystem around the interTwin platform, enhancing its interoperability through community-driven innovation and adoption.

To validate its interoperability strategies, interTwin employs extensive testing methodologies centered around real-world use cases. By engaging with partners from various domains to conduct interoperability tests, the project identifies potential integration challenges and refines its approaches accordingly. This use case-driven testing ensures that the interTwin infrastructure remains adaptable and capable of integrating with a broad spectrum of Digital Twin initiatives.

Interoperability with external initiatives like DestinE is essential for maximising the potential of the DTE. DestinE, a flagship initiative of the European Commission, aims to develop a high-precision digital model of the Earth to monitor and predict environmental phenomena. The interoperability of the DTE with DestinE focuses mainly on data sharing to enable environmental DTs from interTwin to access and expose datasets from/to DestinE.

7.1. Overview of DestinE and interoperability points

DestinE[45] is designed to develop a Digital Twin of the Earth that can simulate the interactions between natural and human activities with high fidelity. It incorporates various domains, such as climate, weather, oceans, and biodiversity, to provide insights into global change and support decision-making at both local and global levels. By linking the DTE with DestinE, the project aims to improve the accuracy of environmental modelling and forecasting.

7.2. DestinE digital twin engine (DTE)

The DestinE DTE is the backbone for developing and operating Digital Twins. The interTwin DTE's interoperability with DestinE ensures that models and data from the DTE are compatible with DestinE's infrastructure, allowing DTs developed within the DTE to be deployed in DestinE's environment.

Key integration features include:

[44] https://www.intertwin.eu/intertwin-use-case-flood-early-warning-in-coastal-and-inland-regions

[45] https://destination-earth.eu/

- **Standardized Data Formats**: The DTE adheres to data standards used by DestinE, such as NetCDF, Zarr, and other geospatial data formats.
- **Federated Data Management**: The DTE supports federated data management, aligning with DestinE's data lake approach, which allows for seamless access to distributed datasets across multiple domains.

7.3. DestinE data lake

The DestinE Data Lake is a federated data infrastructure designed to store and manage vast amounts of Earth observation data, climate models, and other geospatial datasets. Interoperability between the DTE and the DestinE Data Lake ensures that data generated or processed by the DTE is available to the broader scientific community involved in DestinE.

Key points of interoperability include:

- **Data Ingestion and Sharing**: The DTE can ingest data from the DestinE Data Lake allowing models within the DTE to use the most current and comprehensive datasets available.
- **Data Contribution**: Simulations and models generated by the interTwin DTE, especially in environmental domains like flood risk assessment or climate impact simulations, can be accessed via the DestinE Data Lake, contributing to the overall repository of data for global environmental monitoring.
- **APIs for Data Access**: The DTE provides APIs that are compatible with DestinE's data access protocols (based on STAC APIs[46]), ensuring that researchers can retrieve data from both the DTE and DestinE without technical barriers.

7.4. DestinE core service platform (DESP)

The Core Service Platform (DESP) of DestinE provides the computational resources and tools necessary for running services user oriented simulations and data processing tasks close to where the Data are generated. DestinE offers the possibility to onboard services on the DESP cloud platform and to integrate with the suite of services available there. The onboarding of some of the interTwin services into the DESP is one of the aspects which will ensure sustainability of the interTwin services.

Another challenge is ensuring secure and authorised access to shared data and computational resources. The DTE integrates AAI systems that are compatible with DestinE's security protocols, ensuring that data sharing and model execution comply with strict security standards.

8. Conclusions

interTwin's contributions to the field of Digital Twins are multi-faceted and impactful. By developing a federated computing framework and an advanced data management system, interTwin has addressed the critical needs for scalability and efficient data integration in Digital Twin deployments. In our contribution, integration is a multi-faceted approach that involves AAI across the distributed infrastructures and services provided; seamless usage of distinct compute resource provisioning models (cloud orchestration, batch system, etc.); automated execution of multi-step workflows across geo-distributed infrastructures that involve Cloud and HPC and a federated data storage layer composed by multiple data storage providers with different technologies.

One of the hallmark achievements of interTwin is its advancement of interoperability among Digital Twin initiatives. Through the adoption of international standards, the implementation of flexible integration mechanisms, and active participation in collaborative frameworks, interTwin has played a pivotal role in fostering a more unified and collaborative Digital Twin ecosystem. This emphasis on interoperability not only facilitates seamless data exchange and system integration but also encourages innovation and shared advancements in the technology.

The impact of the interTwin project extends beyond the immediate advancements in Digital Twin technologies. By enabling more efficient, scalable, and interoperable Digital Twin solutions, interTwin contributes to the acceleration of research and development across various fields, from environmental monitoring and healthcare to smart cities and advanced manufacturing. Looking forward, the project's open-source approach and community engagement initiatives promise to sustain a vibrant ecosystem of developers, researchers, and industry practitioners who will continue to evolve and expand the capabilities of Digital Twin technologies.

Future directions for interTwin and other similar initiatives are towards interoperability, using a common architectural framework (beyond efforts by NIST and others) and developing a common language and glossary.

CRediT authorship contribution statement

Andrea Manzi: Writing – original draft, Supervision, Project administration, Investigation, Funding acquisition, Formal analysis; **Raul Bardaji:** Writing – review & editing, Writing – original draft, Investigation, Conceptualization; **Ivan Rodero:** Validation, Supervision, Methodology, Investigation, Conceptualization; **Germán Moltó:** Writing – original draft, Validation, Supervision, Software, Investigation, Funding acquisition; **Sandro Fiore:** Validation, Supervision, Methodology, Investigation, Funding acquisition, Conceptualization; **Isabel Campos:** Writing – review & editing, Writing - original draft, Funding acquisition, Conceptualization; **Donatello Elia:** Writing – review & editing, Writing – original draft, Supervision, Methodology, Investigation, Funding acquisition; **Francesco Sarandrea:** Software, Investigation; **A. Paul Millar:** Writing – review & editing, Validation, Supervision, Software, Investigation, Funding acquisition; **Daniele Spiga:** Writing – original draft, Supervision, Software, Methodology, Investigation, Funding acquisition, Conceptualization; **Matteo Bunino:** Writing – review & editing, Writing – original draft, Software, Investigation, Conceptualization; **Gabriele Accarino:** Investigation; **Lorenzo Asprea:** Investigation; **Samuel Bernardo:** Writing – review & editing, Software, Investigation; **Miguel Caballer:** Writing – review & editing, Writing – original draft, Software, Investigation, Conceptualization; **Charis Chatzikyriakou:** Writing – review & editing, Supervision, Funding acquisition, Conceptualization; **Diego Ciangottini:** Writing – review & editing, Validation, Software, Investigation; **Michele Claus:** Writing – review & editing, Supervision, Investigation; **Andrea Cristofori:** Writing – review & editing, Supervision; **Davide Donno:** Software, Investigation; **Emanuele Donno:** Software, Investigation; **Iacopo Ferrario:** Writing – review & editing, Software, Conceptualization; **Massimiliano Fronza:** Investigation; **Alexander Jacob:** Writing – review & editing, Validation, Supervision, Investigation, Funding acquisition, Conceptualization; **Javad Komijani:** Validation, Software, Investigation; **Marina Krstic Marinkovic:** Writing – review & editing, Methodology, Investigation, Funding acquisition; **Federica Legger:** Software, Investigation; **Ivan Palomo:** Software, Investigation; **Estíbaliz Parcero:** Writing – original draft, Software, Methodology, Investigation, Conceptualization; **Rakesh Sarma:** Validation, Software, Methodology, Investigation, Conceptualization; **Gaurav Sinha Ray:** Software, Investigation; **Sara Vallero:** Validation, Supervision, Software, Investigation, Conceptualization; **Juraj Zvolensky:** Software, Methodology, Investigation.

Data availability

Data will be made available on request.

[46] https://stacspec.org/

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Isabel Campos Plasencia reports financial support was provided by Horizon Europe. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work was supported by the project 'An interdisciplinary Digital Twin Engine for science' (interTwin) that has received funding from the European Union's Horizon Europe Programme under Grant 101058386.

References

[1] D. Kierans, et al., Realising distributed digital twins within federated digital infrastructures, in: DiDiT 2024 - 1st International Workshop on Distributed Digital Twins, Groningen, Netherlands, 2024. http://ceur-ws.org.
[2] H. Wu, et al., A comprehensive review of digital twin from the perspective of total process: data, models, networks and applications, Sensors 23 (19) (2023) 8306. https://doi.org/10.3390/s23198306
[3] S.N. others, Digital ecosystems for developing digital twins of the earth: the destination earth case, Remote Sens. 13 (11) (2021) 2119. https://doi.org/10.3390/rs13112119
[4] D. Lecarpentier, et al., Developing prototype digital twins for biodiversity conservation and management: achievements, challenges and perspectives, Res. Ideas Outcomes 10 (2024) e133474. https://doi.org/10.3897/rio.10.e133474
[5] S. Cacciaguerra, et al., Digital twin components for geophysical extreme phenomena: the example of volcanic hazards within the DT-GEO project, in: Conferenza GARR 2023 - Saperi Interconnessi - Selected Papers, Firenze, Italy, 2023, pp. 8–15. https://doi.org/10.26314/GARR-Conf23-proceedings-01
[6] R. Carbonell, et al., Digital twinning of geophysical extreme phenomena (DT-GEO), in: Proc. EGU General Assembly 2023, Vienna, Austria, 2023. https://doi.org/10.5194/egusphere-egu23-5674
[7] Digital twins bringing agility and innovation to manufacturing SMEs, by empowering a network of DIHs with an integrated digital platform that enables manufacturing as a service (MaaS), 2020, Grant Agreement 952071, https://cordis.europa.eu/project/id/952071.
[8] P. Unal, et al., Cognitive digital twins: digital twins that learn by themselves, foresee the future, and act accordingly, 2025, https://www.digitaltwinconsortium.org/2022/09/cognitive-digital-twins-digital-twins-that-learn-by-themselves-foresee-the-future-and-act-accordingly/.
[9] M.N.K. Boulos, et al., An adapted model of cognitive digital twins for building lifecycle management, Appl. Sci. 11 (9) (2021) 4276. https://doi.org/10.3390/app11094276
[10] V. Gadepally, et al., The BigDAWG polystore system, ACM SIGMOD Rec. 45 (4) (2016) 11–16. https://doi.org/10.1145/2814710.2814713
[11] V. Gadepally, et al., The BigDAWG polystore system and architecture (2016). arXiv:1609.07548
[12] R.G. Patidar, et al., Polystore data management systems for managing scientific data-sets in big data archives, in: Big Data Analytics. BDA 2018. Lecture Notes in Computer Science, 11297, Springer, 2018. https://doi.org/10.1007/978-3-030-04780-1_15
[13] L.G. Azevedo, et al., HKPoly: a polystore architecture to support data linkage and queries on distributed and heterogeneous data, in: XX Brazilian Symposium on Information Systems (SBSI '24), ACM, 2024. https://doi.org/10.1145/3658271.3658322
[14] S.L. Chaparro-CÃ¡rdenas, et al., A technological review of digital twins and artificial intelligence for personalized and predictive healthcare 13 (14) (2025) 1763. https://doi.org/10.3390/healthcare13141763
[15] Y. Jiang, et al., Digital twin-enabled real-time synchronization for planning, scheduling, and execution in precast on-site assembly 141 (2022). https://doi.org/10.1016/j.autcon.2022.104397
[16] M. Grieves, Digital twin: manufacturing excellence through virtual factory replication, 2015, (White Paper). Available on ResearchGate, https://www.researchgate.net/publication/275211047.
[17] T. Sun, et al., Digital twin in healthcare: recent updates and challenges 10 (2022). https://doi.org/10.1177/20552076221149651
[18] R. Bardaji, et al., interTwin D3.5 DTE Blueprint Architecture, Functional Specifications, and Requirements Analysis, third version, Technical Report, Zenodo, 2024. Version under EC review, https://doi.org/10.5281/zenodo.14034231
[19] M. Caballer, et al., Infrastructure manager: a TOSCA-based orchestrator for the computing continuum, J. Grid Comput. 21 (3) (2023) 51. https://doi.org/10.1007/s10723-023-09686-7
[20] D. Elia, et al., PyOphidia: a python library for high performance data analytics at scale, SoftwareX 24 (2023) 101538. https://doi.org/10.1016/j.softx.2023.101538
[21] D. Elia, et al., Towards HPC and big data analytics convergence: design and experimental evaluation of a HPDA framework for escience at scale, IEEE Access 9 (2021) 73307–73326. https://doi.org/10.1109/ACCESS.2021.3079139
[22] S. Fiore, et al., Ophidia: a full software stack for scientific data analytics, in: 2014 International Conference on High Performance Computing & Simulation (HPCS), 2014. https://doi.org/10.1109/HPCSim.2014.6903706
[23] C. Palazzo, et al., A workflow-enabled big data analytics software stack for escience, in: 2015 International Conference on High Performance Computing & Simulation (HPCS), 2015, pp. 545–552. https://doi.org/10.1109/HPCSim.2015.7237088
[24] S. Fiore, et al., A graph data model-based micro-provenance approach for multi-level provenance exploration in end-to-end climate workflows, in: 2023 IEEE International Conference on Big Data (BigData), 2023, pp. 3332–3339. https://doi.org/10.1109/BigData59044.2023.10386983
[25] L. Sacco, et al., Enabling provenance tracking in workflow management systems, in: 2024 IEEE International Conference on Big Data (BigData), 2024, pp. 4402–4409. https://doi.org/10.1109/BigData62323.2024.10825405
[26] S. Bernardo, et al., Software quality assurance as a service: encompassing the quality assessment of software and services, Future Gener. Comput. Syst. 156 (2024) 254–268. https://doi.org/10.1016/j.future.2024.03.024
[27] yProv4ML: effortless provenance tracking for machine learning systems, SoftwareX 31 (2025) 102298. https://doi.org/10.1016/j.softx.2025.102298
[28] H. Omidi, et al., Towards provenance-aware earth observation workflows: the openEO case study, in: 21st IEEE International eScience Conference, 2025, to appear.
[29] Padovani, et al., A software ecosystem for multi-level provenance management in large-scale scientific workflows for AI applications, in: SC24-W: Workshops of the International Conference for High Performance Computing, Networking, Storage and Analysis, 2024, pp. 2024–2031. https://doi.org/10.1109/SCW63240.2024.00253
[30] S. Risco, et al., Rescheduling serverless workloads across the cloud-to-edge continuum, Future Gener. Comput. Syst. 153 (2024) 457–466. https://doi.org/10.1016/j.future.2023.12.015
[31] S. Chatterji, et al., Multiresolution techniques for the detection of gravitational-wave bursts, Classical Quantum Gravity 21 (20) (2004) S1809.
[32] A. Paszke, et al., PyTorch: an imperative style, high-performance deep learning library. arXiv 2019, 10 (1912). arXiv preprint arXiv:1912.01703
[33] D. Macleod, et al., GWPy: a python package for gravitational-wave astrophysics, SoftwareX 13 (2021) 100657.
[34] O. Ronneberger, et al., U-Net: convolutional networks for biomedical image segmentation, in: Medical Image Computing and Computer-assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18, Springer, 2015, pp. 234–241.
[35] F. Robinet, et al., Omicron: a tool to characterize transient noise in gravitational-wave detectors, SoftwareX 12 (2020) 100620.
[36] R. Mendelsohn, et al., The impact of climate change on global tropical cyclone damage 2 (3) (2012) 205–209. https://doi.org/10.1038/nclimate1357
[37] Y. Sun, et al., Impact of ocean warming on tropical cyclone size and its destructiveness 7 (1) (2017) 8154. https://doi.org/10.1038/s41598-017-08533-6
[38] G. Accarino, et al., An ensemble machine learning approach for tropical cyclone localization and tracking from ERA5 reanalysis data, Earth Space Sci. 10 (11) (2023) e2023EA003106. https://doi.org/10.1029/2023EA003106
[39] K. Simonyan, et al., Very deep convolutional networks for large-scale image recognition, (2015). arXiv:1409.1556
[40] H. Hersbach, et al., The ERA5 global reanalysis, Q. J. R. Meteorolog. Soc. 146 (730) (2020) 1999–2049. https://doi.org/10.1002/qj.3803
[41] K.R. Knapp, et al., The international best track archive for climate stewardship (IBTrACS): unifying tropical cyclone data, Bull. Am. Meteorol. Soc. 91 (3) (2010). https://doi.org/10.1175/2009BAMS2755.1
[42] Z. Zhou, et al., UNet++: a nested U-Net architecture for medical image segmentation, in: D. Stoyanov, a. et (Eds.), Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support, Springer International Publishing, Cham, 2018, pp. 3–11.
[43] I. Karasante, et al., SeasFire cube - a multivariate dataset for global wildfire modeling, Sci. Data 12 (1) (2025). https://doi.org/10.1038/s41597-025-04546-3
[44] B. B. C. O'Neill, et al., The scenario model intercomparison project (scenarioMIP) for CMIP6, Geosci. Model Dev. 9 (9) (2016). https://doi.org/10.5194/gmd-9-3461-2016
[45] R. J, Haarsma, et al., High resolution model intercomparison project (HighResMIP v1.0) for CMIP6, Geosci. Model Dev. (2016). https://doi.org/10.5194/gmd-9-4185-2016
[46] L. Cinquini, D. Crichton, C. Mattmann, J. Harney, G. Shipman, F. Wang, R. Ananthakrishnan, N. Miller, S. Denvil, M. Morgan, Z. Pobre, G.M. Bell, C. Doutriaux, R. Drach, D. Williams, P. Kershaw, S. Pascoe, E. Gonzalez, S. Fiore, R. Schweitzer, The earth system grid federation: an open infrastructure for access to distributed geospatial data, Future Gener. Comput. Syst. 36 (2014) 400–417. https://doi.org/https://doi.org/10.1016/j.future.2013.07.002