interTwin

D3.5 DTE Blueprint Architecture, Functional Specifications, and Requirements Analysis, third version

Status: Under EC Review
Dissemination Level: public

Funded by the European Union

Disclaimer: Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them

Abstract

Key Words Architecture, functional specifications, requirements analysis, Digital Twin Engine

This document presents the last architectural blueprint for the interTwin Digital Twin Engine (DTE), detailing its functional specifications, requirements analysis, and components. The blueprint guides the technical Work Packages (WPs) within the interTwin project, ensuring alignment with associated subsystem requirements. Furthermore, it integrates insights from related initiatives and projects to identify architectural components that can be adopted within the interTwin framework. This blueprint acts as the definitive conceptual model for the DTE.

Document Description			
D3.5 DTE Blueprint Architecture, Functional Specifications, and Requirements Analysis, third version			
Work Package number 3			
Document type	Deliverable		
Document status	Under EC Review	**Version**	1.0
Dissemination Level	Public		
Copyright Status	CC BY This material by Parties of the interTwin Consortium is licensed under a Creative Commons Attribution 4.0 International License.		
Lead Partner	EGI.eu		
Document link	**https://documents.egi.eu/document/3934**		
DOI	**https://zenodo.org/records/14034231**		
Author(s)	● Raul Bardaji (EGI.eu) ● Andrea Manzi (EGI.eu) ● Ivan Rodero (EGI.eu) ● Thomas Geened (ECMWF) ● Adam Warde (ECMWF)		
Reviewers	● Andrea Cristofori(EGI.eu) ● Germán Moltó (UPV)		
Moderated by:	● Andrea Anzanello (EGI.eu)		
Approved by	Andrea Cristofori (EGI.eu) on behalf of TCB		

Revision History

Version	Date	Description	Contributors
V0.1	07/05/2024	ToC	A. Manzi (EGI.eu), R. Bardaji (EGI.eu), I. Rodero (EGI.eu)
V0.2	22/05/2024	Chapter 2 written	R. Bardaji (EGI.eu), A. Manzi (EGI.eu), I. Rodero (EGI.eu)
V0.3	28/05/2024	Chapter 5 written – 5.7 not finished	R. Bardaji (EGI.eu), A. Manzi (EGI.eu), I. Rodero (EGI.eu)
V0.4	29/05/2024	Chapter 4 written	R. Bardaji (EGI.eu), A. Manzi (EGI.eu), I. Rodero (EGI.eu)
V0.5	02/05/2024	Chapters 6 and 1 are written	R. Bardaji (EGI.eu), A. Manzi (EGI.eu), I. Rodero (EGI.eu)
V0.6	03/07/2024	Chapters 3.1, 3.2, 3.3 and 3.7 are written	R. Bardaji (EGI.eu), A. Manzi (EGI.eu), I. Rodero (EGI.eu)
V0.7	15/07/2024	Chapters 3.5.3, 3.6, are written. Chapter 3.7 is updated	R. Bardaji (EGI.eu), A. Manzi (EGI.eu), I. Rodero (EGI.eu)
V0.8	26/08/2024	Update figures and review all chapters	R. Bardaji (EGI.eu), A. Manzi (EGI.eu), I. Rodero (EGI.eu)
V0.9	30/08/2024	Corrected typos. Added a text in Section 1.2 explaining the document modifications. Completed Section 1.3 with definitions. Added Environmental DTs names in Section 2.2. Updated Figure 5. Enlarged text in Figures 6 and 7 for readability. Added Figure 8, 9, 10, 11, 12 and 13.	R. Bardaji (EGI.eu), A. Manzi (EGI.eu), I. Rodero (EGI.eu)
V0.10	06/09/2024	Added some references. Chapter 5.8. created.	R. Bardaji (EGI.eu), A. Manzi (EGI.eu), I. Rodero (EGI.eu)
V0.11	13/09/2024	Document formatted with the project template	R. Bardaji (EGI.eu), A. Manzi (EGI.eu), I. Rodero (EGI.eu)

v0.12	02/10/2024	Version ready for internal Review	R. Bardaji (EGI.eu), A. Manzi (EGI.eu), I. Rodero (EGI.eu)
v0.13	04/10/2024	Internal review	G. Moltó (UPV)
v0.14	25/10/2024	Internal and TCB reviews	A. Cristofori (EGI.eu)
v0.15	28/10/2024	Version ready for QA	R. Bardaji (EGI.eu), A. Manzi (EGI.eu), I. Rodero (EGI.eu)
V1.0	04/11/2024	**Final**	

When the terminology/acronyms are available via link below, please remove this table.

Terminology / Acronyms

Term/Acronym	Definition
AAI	Authentication and Authorization Infrastructure - Framework for managing authentication and authorisation processes.
API	Application Programming Interface - Set of rules and tools for building software applications.
CGNN/GNN	Convolutional Graph Neural Networks / Graph Neural Networks - Neural networks designed for processing graph-structured data.
CI/CD	Continuous Integration / Continuous Deployment - Practices for automating code integration and deployment processes.
CNN	Convolutional Neural Network - A deep neural network used primarily for image processing.
CPU	Central Processing Unit - The primary component of a computer that performs most of the processing.
CVAE	Convolutional Variational Auto-Encoder - A type of autoencoder used to generate new data samples.
CI/CD	Continuous Integration and Delivery
CWL	Common Workflow Language
DEDL	Destination Earth Data Lake
DNN	Deep Neural Network - A neural network with multiple layers between input and output.
DT	Digital Twin - A virtual representation of a physical system used for simulation and analysis.
DTE	Digital Twin Engine - The core system for managing and operating digital twins.
ESGF	Earth System Grid Federation - A collaboration for climate data infrastructure.
EOSC	European Open Science Cloud - A federated environment for research data across Europe, enabling seamless access and

	reuse of scientific information while supporting a wide array of research infrastructures..
FAIR	Findable Accessible Interoperable Reusable
FPGA	Field-Programmable Gate Array - An integrated circuit that the customer can configure after manufacturing.
GAN	Generative Adversarial Network - A class of machine learning frameworks designed to generate realistic data samples.
GDPR	General Data Protection Regulation
GPU	Graphics Processing Unit - A specialised processor designed to accelerate graphics rendering and parallel processing.
HPO	Hyper-Parameter Optimisation
HPC	High-Performance Computing
HTC	High-Throughput Computing
IBTrACS	International Best Track Archive for Climate Stewardship - A global dataset of tropical cyclone best-track data.
IaaS	Infrastructure as a Service
IdP	Identity Providers
IDPS	Intrusion Detection and Prevention Systems
IDS	Intrusion Detection Systems
iRODS	Integrated Rule-Oriented Data System
K8S	Kubernetes
KER	Key Exploitable Result - Important project outcomes with significant exploitation potential.
LHC	Large Hadron Collider - The world's largest and most powerful particle accelerator.
ML	Machine Learning - A field of artificial intelligence that uses statistical techniques to give computer systems the ability to learn from data.
MPI	Message Passing Interface - A standardised and portable message-passing system for parallel computing.
NetCDF	Network Common Data Form - A set of machine-independent data formats for array-oriented scientific data.
OS	Operating System - System software that manages computer hardware and software resources.
PaaS	Platform as a Service
QCD	Quantum Chromodynamics - A theory of the strong interaction between quarks and gluons.
REST	Representational State Transfer - An architectural style for designing networked applications.
SaaS	Software as a Service

S3	Simple Storage Service - An object storage service that offers industry-leading scalability, data availability, security, and performance.
SFINCS	Super-Fast Inundation of Compound Flooding - A model for fast dynamic simulation of compound flooding events.
SKA	Square Kilometre Array - A large multi-radio telescope project to build the world's largest radio telescope.
SQAaaS	Software Quality Assurance as a Service
SSO	Single Sign-On
STAC	SpatioTemporal Asset Catalog - A specification for describing geospatial information.
ToC	Table of Contents - An organised listing of the chapters and sections in a document.
TPC	Third-Party Copies
VM	Virtual Machine
WP	Work Package

Terminology / Acronyms: https://confluence.egi.eu/display/EGIG

Table of Contents

Table of Tables

Table of Figures

Executive summary

This document presents the third and final version of the architectural blueprint for the interTwin Digital Twin Engine (DTE). It builds upon the previous versions by incorporating the latest functional specifications, requirements analysis, and essential building blocks, reflecting the project's advancements and insights gained throughout its progression.

The final blueprint serves as a comprehensive guide for the technical Work Packages (WP5, WP6, and WP7) of the interTwin project. It aligns with the related software components' original and newly identified requirements. This document synthesises information and requirements gathered from various Work Packages, demonstrating an evolved understanding and development since the previous versions.

This iteration presents an advanced and refined conceptual model of the DTE, showcasing its continuous evolution through collaborative co-design. It emphasises incorporating feedback and findings from selected communities, interactions with other Digital Twin (DT) initiatives, and refining the widely accepted DT conceptual model to suit a broader range of applications.

In addition, this version includes perspectives from related initiatives and projects, identifying potential architectural components for integration within the interTwin framework. It highlights the importance of achieving interoperability and adaptability, ensuring the architecture remains flexible and responsive to new requirements and technological developments.

1. Introduction

This document presents the third version of the Digital Twin Engine (DTE) Architectural Blueprint. It aims to provide an updated and detailed overview of the DTE's architectural structure, incorporating the latest advancements and insights. Based on developer interviews and analyses of related project documents, this version offers a comprehensive understanding of the requirements across various DT domains. Additionally, it includes information from related initiatives and projects to enhance the interoperability and effectiveness of the proposed architectural framework.

1.1 Scope

The primary objective of this document, as the third and final version of the Digital Twin Engine (DTE) Architecture blueprint, is to present a comprehensive and updated overview of the DTE's architectural framework. This blueprint reflects the latest advancements and insights, incorporating additional requirements identified from developer interviews and key findings from other project documents. It ensures a thorough understanding of the needs across various scientific Digital Twin (DT) domains.

This iteration includes perspectives from other related initiatives and projects, helping to identify potential architectural components that could be integrated within the interTwin framework. By highlighting areas where achieving interoperability is beneficial, the blueprint ensures that the architecture remains flexible and adaptable.

1.2 Document Structure

This document is structured to offer a detailed and systematic overview of our study and findings. The structure is as follows:

1. **Introduction**: Sets the stage for the document.
2. **Requirements Analysis for the Architecture**: Examines the requirements for the proposed architecture.
3. **Blueprint Architecture**: Presents the proposed architectural framework.
4. **Alignment to Destination Earth**: Compares the proposed architecture with architecture from the Destination Earth project.
5. **Relation to External Initiatives**: Explores the relationship of our work with various external initiatives.
6. **Conclusions**: Summarise the key findings and implications of the work.

In this document version, several important updates have been made to the structure and content to reflect the project's progress and newly identified priorities. The requirements analysis section has been updated to include additional detailed information about the use cases and rewritten for better comprehension, reinforcing alignment with the specific needs of each scientific discipline involved. The DTE architecture presentation has also been restructured to provide a clearer and more modular view of its components, making it easier for technical teams to understand and develop. New sections have also been integrated to explain the Digital Twin Data Lake and address security and Authentication/Authorization aspects. The text on interoperability with other relevant initiatives, such as Destination Earth, has been updated for better comprehension, with additional information added where necessary. Concretely, the updates are the following:

- Section 2: **Requirements Analysis**
    - The text has been rewritten for improved clarity, ensuring alignment with the specific needs of each scientific discipline involved.
- Section 3: **Blueprint Architecture**
    - The architecture of the Digital Twin Engine (DTE) has been restructured to offer a more modular and coherent view. Notable updates include:
        - Figures provide additional visual context and are updated with larger text to improve readability.
        - **3.4.1 The Data Lake in interTwin**: A new subsection has been added to explain the role and architecture of the Digital Twin Data Lake, outlining its key characteristics
        - **3.5.1 Real-time data acquisition and processing**: Clarifications on the framework for data acquisition and processing in real-time scenarios.
        - **3.5.2 Workflow Composition**: This subsection has been expanded to detail the automation, scalability, and integration of multiple data sources within workflows.
    - **3.7: Security and Privacy**
        - A new section has been introduced to address security and privacy concerns, specifically focusing on the Authentication and Authorization Infrastructure (AAI), which ensures secure access and data protection within the DTE.
- Section 4: **Alignment to Destination Earth**
    - The text and figures regarding the interoperability between the DTE and Destination Earth, has been revised for better clarity, and new information has been added where necessary.
- Section 5: **Relation to External Initiatives**
    - The EOSC Section has been enriched to include the new EOSC EU Node and EOSC Federations concepts
    - A new subsection (**5.6**) has been created to incorporate additional analysis of DT-Geo and BioDT projects.
    - The Gaia-X section has been removed as no particular engagement activity with our project has been found.

1.3 Definitions and Glossary

A **Digital Twin (DT)** is a virtual representation of a physical object, process, or system. It is created and sustained with information derived from one or many data sources, such as sensors or models considering historical and real-time observations. A Digital Twin is a digital replica that mimics the behaviour, performance, and characteristics of its physical counterpart, enabling researchers, engineers and operators to monitor and study the physical system in a controlled environment and, more importantly, simulate its behaviour in many different scenarios.

An **Environmental DT** is a comprehensive digital representation of large, dynamic, and complex natural systems, such as river basins, oceans, or climate systems. These twins integrate data from various sources, including real-time observations and historical records, to simulate and predict environmental processes over different time scales. They are used for operational purposes, such as real-time data monitoring and multi-day forecasting, and long-term scenario simulations to support policymaking and environmental management. Unlike DTs of smaller objects, Environmental DTs must manage natural and anthropogenic processes' complexity and inherent uncertainties, often requiring interdisciplinary collaboration and adaptive approaches.

The **Digital Twin Engine (DTE)** is an open-source integrated platform underpinned by open standards, APIs, and protocols. It facilitates the development and implementation of specific Digital Twins. DTE supports the setup, training and exploitation of the digital twin.

A **Digital Twin Application** is a user-facing implementation of a DT. DT applications are the consumers of the capabilities offered by the DTE, thus introducing use case-specific requirements.

The **Digital Twin Engine (DTE) infrastructure modules** provide specific capabilities for implementing Digital Twins, such as federated data and computing resources needed for modelling and simulation tasks.

The **Digital Twin Engine (DTE) core modules** offer cross-domain capabilities, simplifying the creation and operation of data-intensive and compute-intensive DT applications.

The **Digital Twin Engine (DTE) thematic modules** are add-ons providing capabilities tailored to the needs of specific application groups. They implement core functionalities for a DT but domain specific. They can evolve into core modules following successful adoption by multiple resource communities across different domains.

DT Developers are people who interact with the DTE, developing Digital Twins and occasionally thematic modules. These modules introduce domain-specific tools and best practices, responding to researchers' needs by creating new DT applications, which are then accessed by scientists.

DT Users are people who can either select a pre-packaged DT application and link it to their use case (physical twin) or optionally update or adapt it when necessary.

DT Infrastructure Providers provides computational resources and storage, to build and run the DTs and eventual connectivity with the physical twin existing in the real world.

Co-design is the process of involving multiple stakeholders in the design and development of products, services, or systems with the goal of creating solutions that are more relevant, effective, and satisfying to the people who will use them.

A **model** is a mathematical representation of a real world process.

A **physical-based model** tries to mimic or predict a real/world process based on the laws of physics and the current understanding of those processes from theoretical and empirical studies.

A **machine learning (ML) or data-driven model** is a mathematical representation of a real-world process based on data. These models are constructed by algorithms that "learn" from input data, hence the term "machine learning." The primary goal of these models is to make predictions or decisions without being explicitly programmed to perform the task.

A **hybrid model** combines a physical based model with data driven elements for example for parameter calibration or the inclusion of observational data streams.

A **Machine Learning (ML) Framework** is a library or tool that simplifies the process of building, training, and validating machine learning models. It provides high-level APIs for data preprocessing, model construction and training, and model evaluation and optimization. These frameworks expedite the development of machine learning applications by providing predefined and optimised algorithms and utilities, reducing the complexity and improving the efficiency of model development.

Iterative Model Updating is a generic term used within the field of machine learning and data science. It refers to a process wherein a predictive model is periodically updated or refined based on new data or information that becomes available over time.

2. Requirements Analysis

Analysing architectural requirements is essential for developing an efficient, reliable, scalable, and European initiative-compatible DTE. This chapter explores the requirements for creating a platform that enables developers and users to build and utilise precise, efficient, and powerful DTs.

This chapter begins by outlining the methodology for requirements collection and analysis. It then describes the high-level requirements for each use case and provides a summary. After presenting the methods and requirements, the chapter introduces the foundational elements of the blueprint architecture.

The interTwin project categorises its architecture components into three main areas: Thematic Modules, Core Modules, and Infrastructure. The following paragraphs explain each category.

Thematic Modules in the DTE include domain-specific modelling techniques, data sources, and algorithms. These modules are designed to address the requirements of each scientific discipline. For instance, they contain functionalities for performing data processing or analysis tasks specific to a use case or DT.

Core Modules of the DTE support creating and operating data-intensive and computation-intensive DT applications. Their key features include:

- **Workflow Management**: Advanced tools for arranging and combining core module functions with domain-specific thematic modules tailored to different use cases.
- **Efficient Data Handling**: Real-time data collection and analysis, ensuring fast and efficient processing of data as it is gathered, using advanced serverless computing techniques.
- **Data Fusion**: This process involves harmonising, aligning, or merging data from diverse sources to create a unified dataset.
- **AI Integration**: Managing the entire lifecycle of AI processes, from developing and training models to validating, verifying, and tracking the uncertainty in models to ensure quality.

The **DTE infrastructure** provides a network of data and computing resources for modelling and simulation tasks, including:

- **Unified Data Management**: Coordinating current and future data infrastructures to establish an integrated and interoperable data services cloud with the goal of building a Data Lake.
- **Versatile Computing Resources**: Offering a range of on-demand computing capabilities to support diverse workflow requirements.
- **Resource and Workflow Orchestration**: Coordinating various resources and workflow elements to ensure efficient execution of complex tasks.
- **Access and Utilisation Management**: Implementing Federation services and AAI for secure access and data usage monitoring across multiple providers.

2.1. Requirements Analysis Process Description

We adopted a collaborative design approach in the initial requirements analysis phase and worked closely with DT developers from key areas such as high energy physics, radio astronomy, gravitational wave astrophysics, climate research, and environmental monitoring. These DT developers challenged us with complex modelling, simulation, and data management needs involving multiple digital infrastructures and advanced processes. This collaboration focused on adopting open standards and ensuring interoperability, which was crucial for developing a unified DT Blueprint Architecture. We aimed to establish a common approach for the implementation of DTs across various research domains.

The process of analysing requirements involved several key steps, detailed further in this document:

1. **Interviews for Requirement Gathering:** We conducted video conferences to understand the needs and expectations for the DTE from different use cases, involving discussions with technical team members and use case representatives.
2. **Summarising High-Level Requirements**: We summarised and synthesised the information from each use case to identify requirements and features for the DTE.
3. **Consolidating Requirements**: Guided by the project's technical coordination team and use case representatives, we examined and integrated similar needs across different use cases.
4. **Collecting Feedback**: We shared the consolidated requirements with stakeholders for input and validation.
5. **Analysing and Integrating Feedback**: We analysed stakeholder feedback and incorporated it into the initial requirements set, enhancing the requirements for the DTE.
6. **Gathering Additional Information**: We reviewed other project documents that detailed work done in each WP throughout the project and contacted the authors for any additional clarifications or information.
7. **Integrating New Information**: We added details obtained from other project documents and responses to any questions posed to their authors and DTE developers, enhancing the overall requirements analysis.

In the initial phase of analysing requirements, as detailed in section 2.2, we identified uncertainties that required further attention and refinement. This ongoing refinement was crucial to the DTE's collaborative design process. It ensured the project could continuously evolve and improve, adapting to new requirements and technological developments. For instance, as we developed the DTE, we encountered new data management methods or collaborative tools incorporated into the architecture. By maintaining a flexible and ongoing refinement process, the DTE blueprint remained agile and capable of responding to the dynamic needs of research collaborations and the constantly changing technology landscape.

As shown in **Figure 1**, the collaborative process allowed interTwin to develop the Blueprint Architecture with user communities. This joint effort directly led to achieving several Key Exploitable Results (KER), outlined as follows:

- KER1 Interdisciplinary Digital Twin Engine - offers both standard and specialised modules for modelling and simulation. This platform aids in creating and implementing Digital Twins for a variety of scientific challenges.
- KER2: Interoperability Framework - This includes guidelines, specifications and the blueprint architecture. The interTwin interoperability framework ensures consistent technical approaches and fosters collaboration across different scientific domains for modelling and simulation.
- KER3: AI Workflow and Lifecycle Management Toolkit - Utilising AI to efficiently process and extract specific information from large-scale research data, improving the effectiveness and precision of simulations and models.
- KER4: Quality Framework - Tools and standards for ensuring quality and trust in outputs, including development of benchmarks and indicators to clearly convey the quality of inputs and outputs from Digital Twins, covering data and model origins, accuracy, and gaps in knowledge.
- KER5: Federated Infrastructure Integration - A distributed computing platform that combines various technologies like High Throughput Computing (HTC), High-Performance Computing (HPC), Cloud, and Quantum Computing. This platform is integrated with EOSC and EU Data Spaces for wider data access and processing capabilities.
- KER6: interTwin Open Source Community - A community of developers, users, and maintainers focused on the design, development, and upkeep of the DTE software.

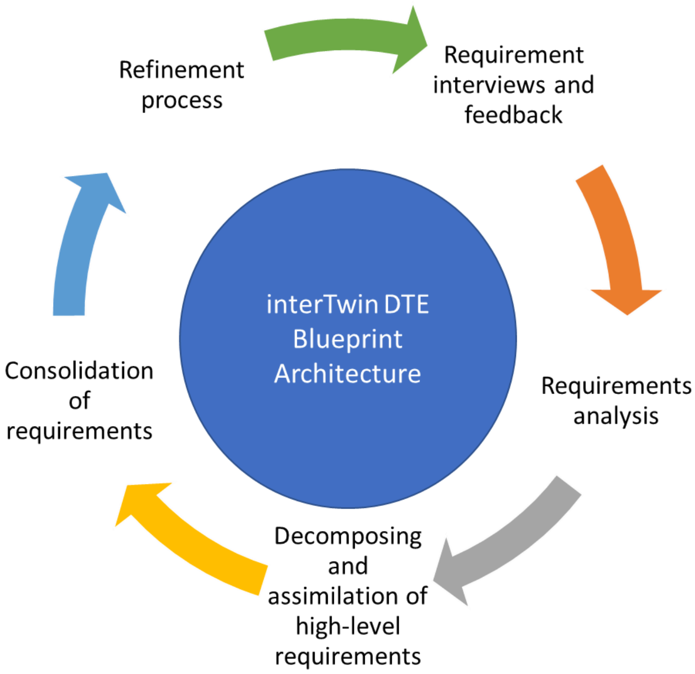

Refinement process
Requirement interviews and feedback
interTwin DTE Blueprint Architecture
Requirements analysis
Decomposing and assimilation of high-level requirements
Consolidation of requirements

Figure 1. Collaborative process to develop the interTwin DTE Blueprint Architecture

By working closely with a wide variety of user communities and valuing their feedback, interTwin consistently improved and updated its interoperability framework. This ongoing collaboration ensured that the framework stayed relevant and practical, meeting the specific needs of different scientific areas. Developers' active involvement played a key role in ensuring that the framework could successfully tackle the diverse challenges these scientific fields presented.

2.2. Use cases' Requirements

This section overviews the requirements for different DT use cases derived from interviews and project-defined scenarios. These use cases spanned a broad range of scientific disciplines, including:

- **High Energy Physics**
  - Lattice QCD Simulations
  - Detector Simulation
- **Radio Astronomy**
  - Noise Simulation
- **Gravitational Wave (GW) Astrophysics**
  - VIRGO Noise Detector DT
- **Environmental Monitoring**
  - Tropical storms change in response to climate change
  - Wildfire risk assessment in response to climate change
  - Flood early warning in coastal and inland regions
  - Alpine droughts early warning
  - Extreme rainfall, temperature and wind weather event changes in response to climate change

  - Flood climate impact in coastal and inland regions

The requirements of each use case were essential to understanding the functionalities needed for the DTE. These requirements will be presented in the upcoming subsections with a consistent approach: starting with a summary of the use case's objective, detailing the specific capabilities required, and concluding with key insights from interviews with developers. This structured presentation ensured a clear and focused depiction of each use case, enabling a thorough understanding of the necessary features and capabilities for the DTE.

Each use case's requirements were categorised into thematic capabilities, core capabilities, and DTE infrastructure. This categorisation helped identify each use case's requirements and how they fit with the DTE's overall capabilities and infrastructure. This approach helped find areas to improve and discover potential connections among the use cases, which was crucial for making the DTE more effective across various scientific disciplines. More specific requirements for each use case can be found in the project's specification Deliverables [**R2**, **R5**, **R6**, **R7**].

2.2.1 Lattice QCD Simulations - High Energy Physics

The Lattice QCD Simulations use case focuses on developing a digital model of quarks and gluons within a lattice structure. This model enables researchers to simulate and analyse particle behaviours under various extreme conditions, which is crucial for advancements in high-energy physics. Building on insights from "D4.2 First Architecture Design of the DTs Capabilities for High Energy Physics, Radio Astronomy, and Gravitational-Wave Astrophysics" [**R2**], this DT integrates advanced data management and machine learning techniques to explore complex areas of quantum chromodynamics.

Thematic Capabilities of the DTE for Lattice QCD Simulations:

- Modelling QCD within a lattice for accurate simulations of subatomic particle interactions.
- Support for analysing the behaviour of quarks and gluons under various conditions, incorporating machine learning techniques to navigate complex parameters in quantum field theories.
- The application of generative models, like Normalizing Flows, aiming to improve the generation of field configurations, reflecting ongoing research and development efforts.
- Effective data management for large-scale simulations in High-Performance Computing (HPC) settings, facilitating data accessibility and collaboration among researchers.

DTE Infrastructure for Lattice QCD Simulations:

- Advanced storage solutions for input and output data, designed to integrate seamlessly with local storage systems and HPC centres.
- A streamlined approach that eliminates the need for external databases during training, focusing on efficient data handling.
- Robust computing resources leveraging both local systems and HPC centres, with capabilities for CPU-based computations and parallel processing using multiple GPUs.

- An operating system and execution framework that are compatible and supportive of the specific requirements for Lattice QCD simulations, ensuring smooth operation and integration.

Requirement Categories:

- **Thematic Capabilities**:
  - Precise modelling of QCD within a lattice.
  - Support for detailed analysis of quarks and gluons in various conditions.
- **Core Capabilities:**
  - **Machine Learning**:
    - **Language**: Python for flexibility and widespread use.
    - **Framework**: PyTorch for robust model development.
    - **Models**: Utilisation of normalising flows for importance sampling.
    - **Monitoring**: Continuous monitoring of acceptance rates.
    - **Validation**: Accept/reject methods to ensure model accuracy.
    - **Hyperparameter Optimization**: Prioritising efficiency for smaller-scale problems.
  - **Workflow Management**:
    - Streamlined processes, including OS command line.
    - Jupyter Notebooks for monitoring.
    - Automated batch scripts.
  - **Data Formats**:
    - Compatibility with binary, text, and serialised (pickle) data formats.
- **DTE Infrastructure**:
  - **Storage**: Versatile input/output solutions compatible with local and HPC centre storage.
  - **Computing**:
    - Integration of local and HPC resources.
    - Support for multi-GPU setups.
  - **Operating System and Framework**: Linux-based environment to meet project specifications.

2.2.2 Detector Simulation - High Energy Physics

The main goal of the Detector Simulation LHC use case is to simulate the LHC detectors, focusing on the multi-dimensional challenges of detector simulation in high-energy physics. This involves addressing the increasing complexity of the detectors, the precision required for advanced experiments, the surge in data volume, and the implementation of sophisticated physics models. As experiments become more advanced, there is a projected significant increase in annual CPU consumption. The necessity for rapid simulation methods is driven by the extensive use of computing resources, with calorimeters often being the most time-intensive sub-detectors.

Thematic Capabilities of the DTE:

- Advanced simulation of LHC detectors, processing generated input data and implementing cutting-edge physics models.
- Efficient transformation of output files into universally compatible formats for seamless integration with various analysis frameworks.

- Generalisation of training data to optimise file sizes without compromising the quality and accuracy of simulations.

Core Capabilities of the DTE:

- **Machine Learning**:
  - Select appropriate ML languages, frameworks, and models like Python and TensorFlow to support the training process and model optimisation.
  - A workflow engine that executes training and model operations, allowing for streamlined and efficient processing.
  - Continuous monitoring and dynamic adjustment of the training process, with the flexibility to adapt parameters or halt training as necessary.
  - Implementation of custom metric definitions to optimise models effectively, aligning with the objectives of high-energy physics simulations.

DTE Infrastructure:

- Robust input/output storage solutions that accommodate local and object storage are designed for high availability and durability.
- A computing infrastructure that supports HPC with Message Passing Interface (MPI) for both CPU and GPU, aligned with the computational demands of high-energy physics.
- An operating system and execution framework tailored to the project's needs, including Linux and containerisation environments.
- Real-time data acquisition and processing capabilities, complemented by offline post-processing for handling larger datasets.
- Workflow tools for executing training steps, pre-processing files, and monitoring, with support for ROOT[1] and HDF5[2] data formats.

Requirement Categories:

- **Thematic Capabilities**:
  - Simulation of LHC detectors using Monte Carlo-generated input data processed through the GEANT4[3] toolkit.
  - Convert specific format output files to standard formats for compatibility with other analysis frameworks.
  - Training data generalisation to abstract input data structures, significantly reducing file sizes.
  - Optimization of generative models to produce data closely resembling Monte Carlo output.
- **Core Capabilities**:
  - **Machine Learning**:
    - **Language**: Python for advanced functionality and community support.
    - **Framework**: TensorFlow for robust, scalable model development.
    - **Models**: GAN, Transformer, Flow, or Energy-based models for effective training.
  - **Workflow Engine**: To execute training and model operations efficiently.

[1] https://root.cern/

[2] https://www.hdfgroup.org/solutions/hdf5/

[3] https://geant4.web.cern.ch/

    - **Monitoring**: Continuous oversight of the training process with adaptability.
    - **Custom Metrics:** Enabling users to define metrics for model optimisation.
- **DTE Infrastructure**:
    - **Storage**: A high-availability, durable system capable of handling large data volumes and adaptable to input and output requirements.
    - **Computing**: HPC with MPI infrastructure to cater to CPU and GPU needs.
    - **OS and Execution Framework**: Linux with containerisation, supported by Jupyter Notebooks.
    - **Data Acquisition and Processing**: Capabilities for real-time online and offline processing for larger datasets.
    - **Workflow Tools**: Comprehensive tools for training, pre-processing, and monitoring, supporting data formats like ROOT and HDF5.

2.2.3 Noise Simulation - Radio Astronomy

This use case outlines the development of an ML system for analysing and filtering data from radio telescopes such as Effelsberg (Germany), MeerKAT (South Africa), and upcoming projects like the Square Kilometre Array (SKA) in South Africa and Australia. The focus is on creating a DT of an astronomical source-telescope system to generate synthetic output signals that replicate the data recorded by real telescopes, encompassing scientifically valuable data and various noise signals.

Thematic Capabilities of the DTE:

- Accurate noise simulation in radio astronomy tools.
- Capability to distinguish different types of radio signals.
- Compatibility with both existing and future radio telescopes.
- Generation of synthetic data tailored to specific observation types and conditions for ML training.
- Support for processing synthetic data through pipelines and analytic tools, enabling debugging and configuration before accurate data processing.

Core Capabilities of the DTE:

- **Machine Learning**:
    - Use ML languages and frameworks like Python and TensorFlow for task execution and model creation.
    - Strategies for efficient ML model training and application.
    - Tools facilitating workflow management, enhancing distributed training, and optimising multi-thread computing.
    - Developing an ML model and a scalable C++ implementation for efficient operation.
    - Parallel configuration of DT and ML data classifier training on computing clusters for near real-time data processing.

DTE Infrastructure:

- Storage solutions for small (under 10 Mb) and medium-sized (under 1 Gb) data files.
- A database or online service for retrospective analysis of model histories.
- Computing resources supporting local CPU and GPU processing.

- Compatibility with operating systems and execution frameworks, including Linux, container environments, and Jupyter Notebooks for monitoring and analysis.

Requirement Categories:

- **Thematic Capabilities**:
    - Accurate noise simulation for radio astronomy instruments.
    - Classification of radio signal sources.
    - System adaptability for current and future radio telescopes.
    - DT-generated synthetic data usability for ML classifier training and pipeline processing.
- **Core Capabilities**:
    - **ML Languages**: Python, supplemented with C++ for efficient implementation.
    - **ML Frameworks**: TensorFlow and similar tools.
    - **ML Training**: Distributed training on HPC clusters, minimising data involvement.
    - **ML Deployment**: Future FPGA firmware implementation.
    - **Workflow Tools**: Distributed training, multi-thread computing, and Jupyter notebooks.
- **DTE Infrastructure**:
    - **Computing**: Local CPU and GPU support.
    - **OS and Execution Framework**: Linux, containerisation using Singularity and Jupyter Notebooks.

2.2.4 VIRGO Noise Detector DT - GW Astrophysics

The primary objective of this use case is to enhance the precision of the VIRGO gravitational wave detector. This DT employs GANs to simulate transient noise and utilises ML and AI for astrophysics research.

Thematic Capabilities of the DTE:

- Accurate simulation of transient noise in the Virgo interferometer using advanced ML.
- Optimization of auxiliary channels for efficient signal processing in astrophysical research.

Core Capabilities of the DTE:

- Using appropriate ML language (Python) and framework (TensorFlow) for astrophysical data processing.
- Real-time processing of parallel input streams and data analysis for large-scale astrophysical data handling.
- Adaptive learning through buffering and retraining in dynamic research environments.
- Essential quality verification and monitoring for accurate data processing.
- Workflow management tools for distributed training and computation in astrophysics.

DTE Infrastructure:

- Database or online service for model history retrieval, supporting collaborative ML model development.
- Computing resources for local CPU and GPU processing.

- Compatible OS and execution framework, including Linux, for astrophysical research environments.

Requirement Categories:

- **Thematic Capabilities**:
  - Accurate noise simulation in the Virgo interferometer.
  - Auxiliary channel optimisation for enhancing signal processing in astrophysical research.
- **Core Capabilities**:
  - Programming Language: Python.
  - ML Framework: TensorFlow, suitable for complex data processing in astrophysics.
  - Real-time processing, buffering, and retraining.
  - Quality verification and monitoring are needed.
- **DTE Infrastructure**:
  - Storage, database, and computing resources on demand.
  - OS and Execution Framework: Linux and containerisation.

2.2.5 Climate Change Future Projections of Extreme Events - Natural Hazards

The DT use case for Climate Change Future Projections of Extreme Events focuses on developing a comprehensive solution to analyse and predict extreme weather events, such as storms, fires, and droughts. This solution involves creating a DT that integrates climate data from sources like COPERNICUS[4], ESGF[5], and IBTrACS[6]. The project includes generating fire risk maps using DNNs and developing a drought early warning system, in addition to data processing suites ensuring ML compliance and delivering ML models for extreme event predictions using an Iterative Model Updating framework.

Thematic Capabilities of the DTE:

- Accurate analysis and prediction of extreme weather events, including storms, fires, and droughts.
- Integrating climate data from multiple sources to generate fire risk maps and early warning systems for floods and droughts.

Core Capabilities of the DTE:

- **Machine Learning**:
  - ML languages and frameworks (Python, TensorFlow, and Keras) suitable for complex tasks in extreme weather event analysis.
  - Support for various ML models, including CNNs, GANs, and CGNN/GNNs, for diverse applications like fire risk mapping and flood early warning.
  - Utilisation of multiple GPUs for training and parallel processing of large-scale climatic data.
- Data pre-processing, visualisation, and workflow management tools such as Jupyter Notebooks and CI/CD for efficient handling and analysis of climate data.

[4] https://www.copernicus.eu/en/access-datas

[5] https://aims2.llnl.gov/search

[6] https://www.ncei.noaa.gov/products/international-best-track-archive

DTE Infrastructure:

- Storage solutions for handling diverse and large-scale climatic datasets.
- Computing resources supporting HPC/HTC environments for complex ML model training and data processing.
- Compatible OS and execution frameworks to support various climate data analysis tools.
- Real-time data acquisition and processing capabilities for immediate response and prediction.
- Advanced workflow management tools for automating and managing the data processing pipeline.

Requirement Categories:

- **Thematic Capabilities**:
    - Simulation and prediction of extreme weather events.
    - Integration of climate data for comprehensive analysis.
- **Core Capabilities**:
    - **ML Language**: Python and its libraries for advanced functionalities.
    - **ML Framework**: TensorFlow, Keras for robust model development.
    - **ML Models**: Support for CNNs, GANs, CGNN/GNNs for specific use cases.
    - **Data Pre-processing and Visualization**: Essential for accurate interpretation of climate data.
- **DTE Infrastructure**:
    - **Parallel Processing**: Capability to handle large datasets efficiently.
    - **Workflow Management**: Tools for managing complex data processing steps.

2.2.6 Early Warning for Extreme Events - Natural Hazards

The primary goal of this DT is to develop a comprehensive simulation and early warning system for high-impact hydrometeorological events, such as floods and droughts. Leveraging multidisciplinary observations and models, the project aims to improve the quality and accuracy of early warnings. This includes generating flood risk maps that trigger early warning alerts when a flood is predicted, mainly focusing on historical flood events in Humber, United Kingdom. The system combines models like SFINCS[7] and Wflow[8] with Sentinel-1 based flood maps generated by the openEO[9] implementation of the Global Flood Monitor[10].

Thematic Capabilities of the DTE:

- Simulations that accurately predict high-impact hydrometeorological events across all earth system components, including using models for super-fast dynamic modelling of compound flooding.
- Improved early warning quality and accuracy for extreme events like floods, utilising combined data from models and satellite-based flood maps.

Core Capabilities of the DTE:

[7] https://www.deltares.nl/en/software-and-data/products/sfincs

[8] https://www.deltares.nl/en/software-and-data/products/wflow-catchment-hydrology

[9] https://openeo.org/

[10] https://global-flood.emergency.copernicus.eu/technical-information/glofas-gfm/

- **Machine Learning**:
    - Appropriate ML language and framework to support the desired ML models and tasks.
    - Processing input data from diverse sources and formats, including river discharge data and satellite imagery.
- Time-scale output information for early warning systems, enabling timely alerts for high-risk areas.
- Workflow management tools to facilitate development, documentation, monitoring, and event-based processing.
- Real-time data acquisition and processing capabilities, triggering alerts like email notifications in high-risk situations.

DTE Infrastructure:

- Storage solutions suitable for objects and files, including database management for data indexing and storage.
- Computing resources supporting local, HPC, and cloud processing for CPU and GPU tasks.
- Compatible OS and execution framework to support the project's requirements, including real-time data processing for early warning systems.

Requirement Categories:

- **Thematic Capabilities**:
    - Accurate simulation of high-impact hydrometeorological events.
    - Provision of improved early warning quality and accuracy.
    - Downscaling from low-resolution climate model output to higher-resolution input for physical and data-driven hydrological models.
- **Core Capabilities**:
    - **ML Language**: Python (PyTorch) and TensorFlow.
    - **ML Models**: Real-time data acquisition and processing.
    - **ML Training**: Processing diverse input data, including Copernicus Climate Data Store[11] and IBTrACS[12] data in NetCDF[13] format.
    - **ML Output**: Timely and accurate information for early warning systems.
    - **Workflow Tools**: openEO process graph-based workflows, Jupyter notebooks, event-triggered monitoring, and event-based platform.
    - **Real-time Data Acquisition and Processing**: Managed through openEO API.
    - **Data Formats**: Support for various formats, including binary, text, and geospatial formats like GeoJSON[14], GeoPackage[15], Shapefiles, JPEG2000[16], GeoTIFF, NetCDF[17], HDF5 and Grib[18].
- **DTE Infrastructure**:

[11] https://cds.climate.copernicus.eu/

[12] https://www.ncei.noaa.gov/products/international-best-track-archive

[13] https://www.unidata.ucar.edu/software/netcdf/

[14] https://geojson.org/

[15] https://www.geopackage.org/

[16] https://jpeg.org/jpeg2000/

[17] https://www.ogc.org/publications/standard/geotiff/

[18] https://en.wikipedia.org/wiki/GRIB

  - **Storage**: Local file-based and object storage (S3) solutions.
  - **Databases**: Indexing into STAC-based catalogues[19] or ingestion into Datacube[20] engines.
  - **Computing**: Local, HPC, and cloud CPU and GPU processing resources.
  - **OS and Execution Framework**: Windows and Linux, with Docker or Singularity for containerisation. Visualisation using Python libraries like matplotlib[21] and cartopy[22].

2.2.7 Climate Change Impacts of Extreme Events - Natural Hazards

This use case is dedicated to evaluating the changes in characteristics of climate-related extreme events and their impacts, employing advanced ML methodologies. It aims to deepen our understanding of future potential impacts of climate extremes and to provide critical insights to support climate change mitigation and adaptation policies.

Thematic Capabilities of the DTE:

- Precise evaluation of the spatial extent, frequency, duration, and intensity of extreme climatic events, utilising global climate simulations.
- Systematic assessment of uncertainties in climate change impacts, utilising multiple greenhouse gas scenarios and simulation ensembles, reinforced by the employment of climate indices.

Core Capabilities of the DTE:

- **Machine Learning**:
  - Deploy diverse ML models tailored to specific domains such as storms, wildfires, floods, and droughts.
  - Comprehensive training of ML models, harnessing GPU power across all grids for the designated area, utilising extensive historical data and computed indexes from input data.
  - Periodic retraining of the ML model, potentially every 5-10 years, to incorporate new data and findings.
- Efficient workflow tools for pre-processing, ensuring data is processed once, stored, and reused.
- Adaptation to various data formats, primarily focusing on NetCDF files and REST APIs for data filtration and retrieval.

DTE Infrastructure:

- Storage solutions can handle large volumes of climate simulation data tailored for hundreds of gigabytes per simulation. Each file, encapsulating monthly data per variable, is estimated to be around 25 MB.
- Robust computing resources, particularly GPU-oriented, for intensive ML model training processes.
- Flexible and adaptable operating systems, predominantly Linux, suited to the climate science domain, coupled with visualisation services.

[19] https://stacspec.org/en

[20] https://en.wikipedia.org/wiki/Data_cube

[21] https://matplotlib.org/

[22] https://scitools.org.uk/cartopy/docs/latest/

- Real-time data acquisition and processing capabilities, sourcing grid data from the Copernicus Climate Data Store, historical and future projection data from ESGF climate data infrastructure[23], facilitated via REST APIs.

Requirement Categories:

- **Core Capabilities**:
  - **ML Language and Framework**: Tailored to suit diverse ML models and tasks.
  - **Training and Retraining**: Adapting to evolving climatic data and models.
  - **Workflow Management**: Streamlining pre-processing, data storage, and utilisation.
  - **Data Acquisition and Processing**: Versatile support for multiple data formats.
- **DTE Infrastructure**:
  - **Storage**: Varied solutions for an extensive array of climate simulations.
  - **Databases and Online Services**: For model history retrieval and offline analysis.
  - **Computing**: GPU-backed resources for rigorous ML model training.

2.3. Requirements Conclusions

This summary encapsulates these diverse requirements and draws conclusions to guide the development of the DTE.

- **Interdisciplinary Approach**: The DTE must accommodate a wide range of scientific disciplines, requiring an interdisciplinary approach to architecture and functionality.
- **Modularity and Scalability**: Given the diverse requirements, the DTE should be modular and scalable, capable of adapting to various use cases without extensive reengineering.
- **Advanced Data Management**: Effective data management is crucial for handling large datasets, ensuring data accessibility, and facilitating collaboration.
- **Integration with HPC**: Many use cases demand integration with HPC resources, indicating a need for robust computing capabilities within the DTE.
- **Machine Learning and AI**: The application of ML and AI is central to most use cases, highlighting the need for the DTE to support advanced ML frameworks and algorithms.
- **Real-time Processing**: Several use cases, especially environmental monitoring, require real-time data processing capabilities for early warning and rapid response.
- **Customization and Flexibility**: The DTE should offer customisation and flexibility to cater to the specific needs of each use case, particularly in terms of data formats, processing tools, and ML models.
- **Collaboration and Workflow Management**: The platform should facilitate smooth workflow management and collaborative efforts, especially in interdisciplinary projects.

[23] https://esgf.llnl.gov/

3. Blueprint Architecture

This chapter provides a comprehensive overview of the components of the DTE and explains how these components work together to form an effective and efficient system.

First, the chapter explains the methodology for developing the Blueprint, introducing C4 diagrams[24]. Next, we will discuss the types of users we anticipate and provide an overview of the DTE's architecture and user base. Finally, the document will delve into the details of each DTE's building blocks and components, offering insights into their respective roles and integration within the overall system framework.

Detailed information about the specific components of the architecture can be found in the architecture and implementation deliverables of the project [R4]

3.1. Methodology

In this document, we use the C4 methodology to describe the architecture of the Digital Twin Engine. This approach simplifies and encourages software diagrams.

The C4 model organises architectural representation into four layers, each serving a specific purpose and audience:

- **Context**: This top-level view outlines the system's interaction with external entities, including other systems, software, and users. It sets the software system's boundaries and communication channels.
- **Container**: This level delves deeper into the architecture and segments the system into specific sections or 'containers', such as web servers, databases, and applications, delineating their roles and interactions.
- **Component**: This more granular level details the functional areas within each container, revealing their responsibilities and how they contribute to the system's operation.
- **Code**: This is the most detailed layer, focusing on the actual implementation and code structure. It is typically visualised through UML class diagrams or similar tools.

This document will focus on the **Context** and technology-agnostic **Container** level, providing a comprehensive yet high-level overview of the system's architecture. Deeper levels, while crucial for technicians and developers, are covered in other project deliverables, ensuring a focused and coherent presentation in this specific context.

3.2. DTE Users

As introduced above, the interTwin DTE will serve mainly three categories of users:

- **DT developers** interact with interTwin DTE, seen as PaaS (Platform as a Service), developing DT applications and occasionally thematic modules tailored to the needs of specific user communities.

[24] https://c4model.com

- **DT Users** access the DTE as a SaaS (Software as a Service) via the DT applications developed by the DT developers. An end user can choose an "out of the box" DT application and connect it to its use case (physical twin) or configure the needed parameters for their experiments.
- The **DT Infrastructure Providers** deliver computational resources, storage, and eventual connection with the physical twin from the real world.

Given this categorisation, the document describes how DT developers and users interact with the system. The interactions of DT Infrastructure Providers are not depicted in the current diagrams, as they were not discussed during the initial iteration and interviews with the project partners.

3.3. DTE System Landscape

The context model explains the architectural overall components and their interrelations within the DTE, optimising efficient, reliable, and scalable operations. The DTE architecture is strategically partitioned into layers, each serving a distinct purpose yet interlinked through APIs to ensure seamless functionality and integration.

The DTE is designed to adhere to standard principles summarised as follows:

- **Standard-based integration and portability**: The DTE is designed to provide end-to-end integration and a ‘one-stop-shop' for domains and target groups outside the project use cases. This will require DT applications to gradually shift from developing in-house solutions to increasing the adoption and development of common open-source modules.
- **Extensibility and modularity**: Interfaces (e.g., APIs and GUIs) need to decouple the DTE from the DT applications implemented.
- **Scalability and sustainability**: The DTE must integrate with application-specific data, computing facilities, and current/future infrastructures from the national to pan-European levels, such as the European Open Science Cloud (EOSC). Investments in digital infrastructures sustain the DTE infrastructure.
- **PaaS and SaaS provisioning**: The DTE needs to provide a PaaS layer for developing custom applications and creating a user work environment integrating relevant data to be accessed by the modelling and simulation tasks and a Software as a Service (SaaS) layer for consuming the functionalities of the digital twins as dedicated services.

Figure 2 illustrates the system context of the overall DTE architecture, focusing on its components' practical aspects and integration. It ensures that each system component is well-connected and aligned with the rest of the DTE architecture.

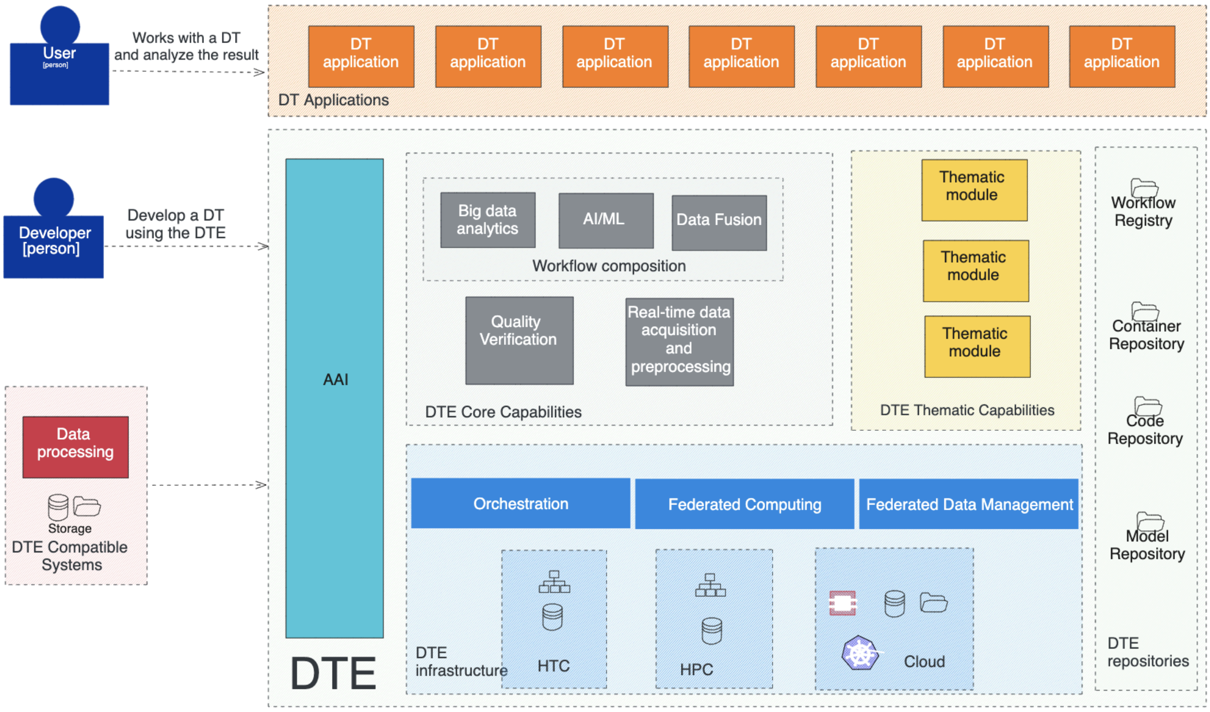


Figure 2. System landscape diagram of the DTE

The elements of the context model figure are outlined below.

DTE Infrastructure: At the foundation lies the DTE Infrastructure, composed of HTC and HPC clusters, cloud resources like OpenStack and Kubernetes (K8S), and storage solutions, both centralised and distributed. This layer forms the backbone, providing the necessary computing and storage capabilities to support the digital twins.

Core Capabilities: The core capabilities, built upon the infrastructure, encompass Workflow composition, Quality Verification, Big Data Analytics, AI/ML processes, Data Fusion, and real-time data analysis. These capabilities enable the DTE to perform real-time data acquisition and pre-processing, advanced workflow execution, and analytics essential for accurate digital representation and analysis.

Thematic Capabilities: Beyond the core, there are Thematic Modules, which integrate domain-specific tools and best practices to enhance the DTE's utility. These modules simplify complex workflows into manageable operational sequences, making them more accessible for DT developers to implement.

Applications: Above the capabilities layer are the DT Applications, which DT developers make in response to specific research needs. These applications leverage the DTE's thematic and core capabilities to provide robust services for DT users.

User Interaction: Users interact with the system through dedicated interfaces. DT Users can employ pre-built or adapt applications to their unique use cases. A user interface layered over the applications facilitates the interaction, simplifying access and manipulation of DT.

Repositories and Registries: Various repositories support the applications and capabilities that store the necessary building blocks for DT development. A Workflow Registry maintains the workflows, ensuring they are accessible and reusable.

Orchestration and Data Management: Orchestration is the management mechanism, coordinating the optimised deployment of components over the infrastructure resources Federated Data Management system ensures datasets are correctly stored, retrieved and transferred over the storage resources available.
Security and Compliance: The Authentication and Authorisation Infrastructure (AAI) is crucial to safeguarding the DTE. It ensures data protection, integrity of simulations and workflows, trust, and compliance and secures interconnected systems against various security threats.
Integration with Real-World Data: Intermediate infrastructures, such as sensors, bridge the gap between the DTE and physical twins. These are critical for filtering, pre-processing, or buffering the data before it is assimilated into the DTE.

3.4. DTE Infrastructure Capabilities

The DTE Infrastructure is structured around four foundational elements, each contributing to the system's functionality and efficiency. **Figure 3** provides a model representation, offering a high-level overview of user interactions within the DTE.

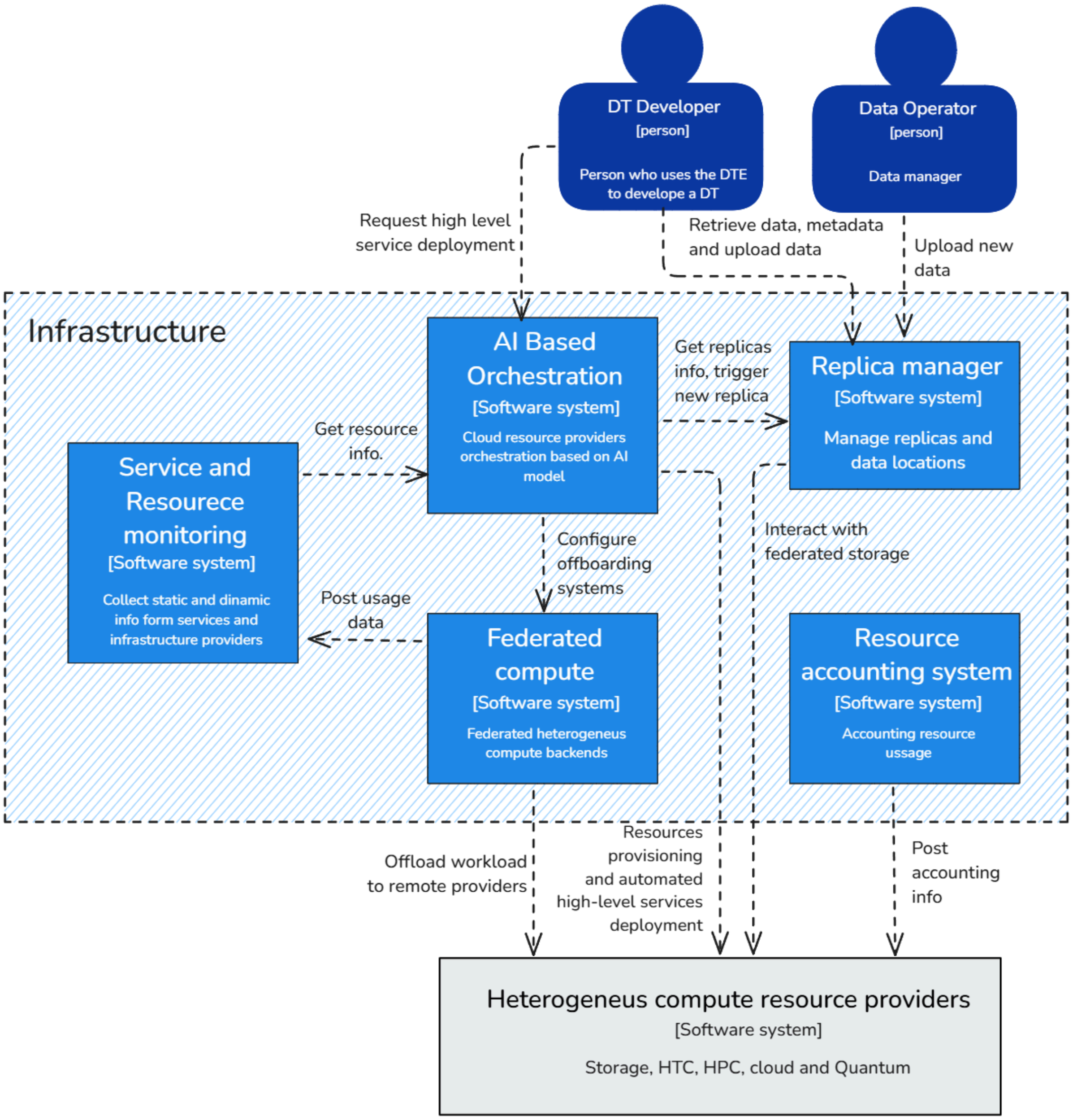

DT Developer
[person]
Person who uses the DTE to develope a DT
Data Operator
[person]
Data manager
Request high level service deployment
Retrieve data, metadata and upload data
Upload new data
Infrastructure
AI Based Orchestration
[Software system]
Cloud resource providers orchestration based on AI model
Get replicas info, trigger new replica
Replica manager
[Software system]
Manage replicas and data locations
Get resource info.
Service and Resource monitoring
[Software system]
Collect static and dinamic info form services and infrastructure providers
Configure offboarding systems
Interact with federated storage
Post usage data
Federated compute
[Software system]
Federated heterogeneus compute backends
Resource accounting system
[Software system]
Accounting resource ussage
Offload workload to remote providers
Resources provisioning and automated high-level services deployment
Post accounting info
Heterogeneus compute resource providers
[Software system]
Storage, HTC, HPC, cloud and Quantum

Figure 3. System landscape diagram of the DTE Infrastructure

AI-Based Orchestrator

At the core of the DTE Infrastructure lies the AI-Based Orchestrator. This advanced software system uses artificial intelligence to manage the orchestration of cloud resources. Its capability lies in evaluating the most suitable cloud resource provider for deploying specific services. By intelligently selecting resources, the orchestrator ensures optimal performance and cost-efficiency.

Federated Compute Module

Complementing the orchestrator is the Federated Compute module, designed to foster a Compute Federation within a heterogeneous environment. It aims to utilise every available resource, including specialised hardware when necessary, to meet specific workflow demands. For example, it may opt for HPC when computational intensity is required, or HTC for tasks needing extensive parallel processing. The Federated Compute acts as a bridge, connecting various computational resources and enhancing their collective effectiveness.

The diagram in Figure 3 illustrates the bidirectional flow between the end-users and these systems. End-users interact with the DTE infrastructure services, either directly or through the workflow GUI, triggering the AI-Based Orchestrator. End-users represent both human and machine-to-machine interactions (e.g., upper layer software). This system then configures and deploys services across the Federated Compute resources, which span multiple heterogeneous backend systems. The federated approach ensures an agile and flexible infrastructure capable of scaling and adapting to varying computational demands.

Replica manager

The DTE's architecture prioritises data accessibility and processing through a distributed federation of storage systems, forming a Data Lake. The infrastructure ensures that required scientific data is readily accessible by the service and the offloaded processes. This is achieved by developing sophisticated software that implements abstraction layers, allowing straightforward interactions with a complex and distributed storage topology.

Resource Accounting System

The Resource Accounting System serves as the infrastructure's governance pillar. It is used in monitoring and ensuring equitable resource allocation and utilisation within this diverse environment. The system uses a centralised mechanism to aggregate and analyse usage data, promoting transparency and accountability across the entire DTE Infrastructure.

This component ensures that resources in a heterogeneous distributed environment are allocated and used fairly. The Accounting system collects and processes usage data from various providers, providing statistical summaries for visualisation and analysis. This system should support various accounting types, including grid batch jobs, cloud VMs, and storage space.

3.4.1. The Data Lake in interTwin

The combination of the interTwin infrastructure storage components are used to develop the interTwin Data Lake. A Data Lake is an architecture that allows for storing and managing large volumes of data in its native format, facilitating the integration of diverse datasets from various sources. The primary goal of a Data Lake is to provide a centralised repository that can store all structured and unstructured data at any scale.

The characteristics of a the interTwin Data Lake are the following:

- **Scalability**: Data Lakes are designed to handle large-scale data storage needs. They can scale up to accommodate multi-Petabyte levels, making them suitable for managing extensive datasets from numerous scientific and research activities.
- **Flexibility**: Unlike traditional databases, Data Lakes can store unstructured, semi-structured, and structured data. This flexibility allows them to accommodate a wide range of data types and formats, from raw data to processed analytics.

- **Data Management**: Effective data management within a Data Lake involves organising, orchestrating, and cataloguing data. Policies for data replication, retention, and deletion are essential to maintaining the integrity and accessibility of data over time.
- **Data Accessibility**: Data Lakes ensure that data is easily accessible to various users and applications. This is facilitated through APIs and interfaces that allow seamless data retrieval and integration into analytics and machine learning workflows.
- **Cost Efficiency**: By centralising data storage and management, Data Lakes can optimise storage costs. This is particularly important for large-scale scientific projects that generate massive amounts of data.
- **Integration with Compute Services**: Data Lakes are often integrated with various compute services, including HPC, cloud resources, and grid interfaces. This integration enables efficient data processing and analysis across different computational environments.
- **Security and Governance**: Robust mechanisms for authentication, authorisation, and identity management are crucial for securing a Data Lake. Ensuring compliance with data governance policies and providing controlled access to sensitive data are key aspects of managing a Data Lake.

The Data Lake employs sophisticated authentication and authorization mechanisms to secure data access. Using AAI systems such as EGI Check-In[25] mediates access to data, ensuring that only authorised users can reach it.

The development and management of the Data Lake involve continuous monitoring and optimisation to ensure that it meets the evolving needs of the scientific community. The integration of advanced technologies and adherence to best practices in data management are essential to maintaining the Data Lake's functionality and efficiency.

Some of the Data Transfer Techniques that can be used within the Data Lake are the following:

- **Third-Party Copies (TPC)**: Direct data transfers between storage systems, optimising data movement efficiency.
- **Multihop Transfers**: Utilising intermediate storage systems to facilitate data transfer when direct TPC is not possible.
- **Streaming**: Enabling data transfer across a broader range of protocols, ensuring compatibility with diverse network environments.

3.5. DTE Core Capabilities

The DTE Core Modules are the central components of the interTwin project, providing essential functionalities and tools required for the development and operation of DT. They address critical challenges and enable efficient, reliable, scalable DT solutions.

One major challenge in the development of interTwin is extending "serverless" computing to DT, which requires a framework for real-time data acquisition and

[25] https://www.egi.eu/service/check-in/

processing based on event-triggered workflow execution. This framework is crucial for most DTs, especially for automating model validation using real-world data.

Another significant challenge is integrating AI techniques, such as ML, which requires advanced distributed training and optimisation methods like Hyperparameter Optimization. A flexible framework must be created to incorporate ML models and data pipelines, interfacing with backend computing and data resources.

Computing and data resources should support various components and best practices for effective data analytics. This involves implementing general-purpose data analytic environments that can be deployed on demand on Cloud resources or providing an interface for seamless integration of HPC and Cloud resources with container workload management services. A key innovation challenge is the on-demand provisioning, horizontal scaling, and integration of workflow mechanisms.

A comprehensive model quality validation strategy is needed to improve DTs, implementing best practices and standard quality measures for model validation. Inspired by DevOps practices, this strategy should leverage automation, Continuous Integration and Delivery (CI/CD), and evaluate FAIR data quality, resulting in the implementation of Model Quality Validation as a Service.

The following sections detail the architecture of each component of the Core Module.

3.5.1. Real time data acquisition and processing

The real-time data acquisition and processing framework for the DTE that supports event-triggered execution of workflow engines has to satisfy the following requirements:

- Detects when new data that requires processing is made available.
- Perform data staging and pre-processing (e.g. to perform data cleansing or data quality assessment).
- Delegate the complex data processing to external workflow management systems in charge of executing resources that can be dynamically provisioned from a Cloud-based infrastructure.

Figure 4 shows the overall architecture for data acquisition and event-driven triggering of workflows, including the high-level components.

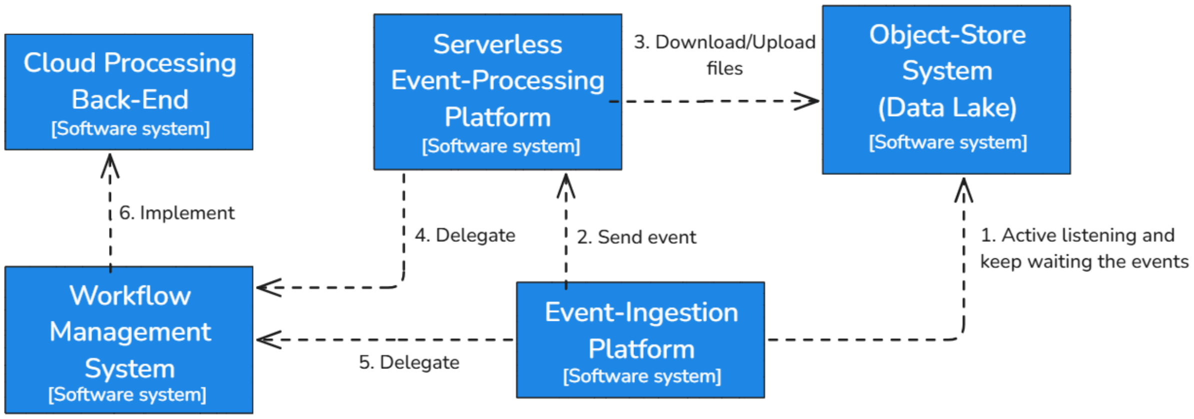

Cloud Processing Back-End [Software system]
Serverless Event-Processing Platform [Software system]
3. Download/Upload files
Object-Store System (Data Lake) [Software system]
6. Implement
4. Delegate
2. Send event
1. Active listening and keep waiting the events
Workflow Management System [Software system]
5. Delegate
Event-Ingestion Platform [Software system]

Figure 4. General architecture for data ingestion and event-driven triggering of workflows

The **event-ingestion system** receives notification events from the file/object-storage system and can execute simple transformation data flows using the system's built-in components.

The **file/object storage system** provides a solution for storing data to be analysed, whether temporary or long-term, as in data lakes. For fault-tolerance and high-availability reasons, the data is typically distributed among many heterogeneous servers while providing a unified vision of a virtual filesystem that can be accessed through various protocols.

The **serverless event-processing platform** receives data pre-processing requests from the event-ingestion system. It performs additional data transformations that the event ingestion system may not be able to handle. This could be due to a lack of support for certain operations or dependencies on external tools that are packaged as Docker images, which cannot run directly within the event-ingestion system.

The **workflow management system** receives tasks delegated by the serverless event-processing platform. This system coordinates and orchestrates multiple tasks or steps necessary to complete more complex data transformations. Its function is to schedule, manage, and monitor the various stages of the process, ensuring that all dependencies are met before a task is executed and that results are stored or sent to the appropriate system. Often, the workflow management system can handle parallel tasks, schedule jobs based on specific events, and delegate specific functions to other systems.

Depending on the type of task or workflow, the workflow management system delegates execution to one of the associated systems. In the case of complex transformations that require more specialised services, it may delegate the work to **the serverless event-processing platform** to handle the heavy lifting of data transformation, taking advantage of the platform's scalability and dependency on containers like Docker. When more intensive or prolonged processing is required, the platform may delegate execution to a cloud processing back-end.

The **cloud processing back-end** is responsible for handling computationally expensive tasks or those requiring extended execution times. This system leverages cloud infrastructure to offer elastic scalability, executing jobs in parallel within a distributed architecture as needed. It implements the instructions and workflows sent from the workflow management system. Operations that other systems cannot handle due to runtime, capacity, or infrastructure limitations are redirected here. Upon completing the processing, the results can be sent back to the workflow management system for further execution stages or stored directly in the object-store system.

Finally, the **Object-Store System (Data Lake)** acts as the final repository for processed or ongoing data. This system stores the transformed files and maintains a consistent, accessible state of the data. It provides interfaces for other systems to download or upload files, enabling efficient collaboration between the different components. With its ability to distribute data across multiple servers and its fault tolerance, it ensures data availability and persistence, both for immediate processing and long-term analysis.

3.5.2. Workflow Composition

The workflow composition allows developers to define and structure the necessary steps to complete complex tasks. This can include everything from data acquisition to processing and analysis, all organised in a coherent workflow. Its functionalities are the follows:

1. **Automation**: This facilitates the automation of repetitive processes and tasks, which is essential for model validation and real-time data integration. It reduces manual intervention and increases efficiency.
2. **Extensibility and Customization**: This feature enables the creation of custom plugins and extensions to add new functionalities or integrate with other systems. This is crucial for adapting the system to specific needs and quickly changing project requirements.
3. **Scalability and Flexibility**: Supports the execution of workflows in distributed and large-scale computing environments. This includes parallel task execution and dynamic task creation, adapting to different demands and workloads.
4. **Integration of Data and Tools**: This function facilitates the integration of multiple heterogeneous data sources and domain-specific tools into a general workflow. This is important for preparing data for advanced analytics and artificial intelligence components.
5. **User Interaction**: This provides interfaces for developers and end-users, allowing them to configure and execute workflows efficiently. Workflow composition tools offer both command-line tools and graphical interfaces for different user experience levels.
6. **Provenance and Traceability**: Integrates provenance components to track data lineage and workflows, ensuring traceability and transparency in processes.

We suggest adopting a Common Workflow Language (CWL[26]) as a specification to define workflow steps. It will eventually operate on different workflow management systems.

The composition and execution of workflows will interact with the **real-time data acquisition component**, triggering the execution of workflows upon data arrival. This component will also activate model validation and data accessibility and reusability services as part of the workflow execution. Additionally, a **provenance component** will be integrated to track the lineage metadata of the workflows.

The **data fusion** is also important in implementing workflows that integrate multiple heterogeneous data sources. The main challenge is to combine domain-specific datasets and tools into the general workflow, preparing them for use in general analytics and AI components. At the end of the process, results from multiple model runs need to be re-integrated for visualisation purposes.

Furthermore, the workflow composition integrates closely with **ML/AI** and **Big Data Analytics subsystems**. The AI subsystem focuses on training and deploying ML models, enhancing the capabilities of DTs with data-driven insights. Big Data Analytics involves constructing templates and managing resources for large-scale data processing, providing the necessary infrastructure and tools for executing complex analytics tasks efficiently.

[26] https://www.commonwl.org/

The following will delve deeper into the blocks that form the Workflow Composition layer:

Data Fusion

Data fusion is a component in the workflow composition that is responsible for integrating multiple heterogeneous data sources into a unified workflow. This process combines domain-specific datasets and tools, preparing them for general analytics and AI components.

The characteristics of Data Fusion are the following:

- **Integration of Heterogeneous Data Sources**: A significant challenge in Data Fusion is merging specialised datasets and tools. These datasets often come from different domains and must be harmonised to be used effectively in the overarching workflow. For example, environmental use cases require merging data from different sources, such as outputs from climate models, satellite imagery, and various types of vector data.
- **Preparation for Analytics and AI**: Data fusion involves preparing these integrated datasets for use in analytics and AI processes. This includes formatting the data appropriately and ensuring compatibility with general analytic components. Methods must be developed to consolidate datasets from diverse origins in collaboration with thematic module developers, managing both gridded and vector data types.
- **Reintegration for Visualization**: After the data analysis phase, results from multiple model runs need to be re-integrated for visualisation. This step makes data accessible for further analysis and decision-making. Procedures must be crafted to format data for integration into AI-driven workflows and to prepare analysed data for visualisation.

The tasks involved in Data Fusion are the following:

- **Establishing Guidelines**: Creating guidelines for developing thematic modules to ensure they can interoperate within the general workflow, particularly concerning data exchange.
- **Consolidating Diverse Datasets**: Developing methods to consolidate datasets from diverse origins in collaboration with thematic module developers.
- **Formatting for AI Workflows**: Ensuring that machine learning models and other AI processes can effectively utilise data.
- **Preparing for Visualization**: Working closely with thematic modules dedicated to visual representation to ensure final outputs are accessible and interpretable.

The design and functionality of the data fusion components are closely tied to the specific data sources being utilised. Ensuring interoperability and seamless data exchange between different components is crucial for the successful integration and utilisation of data in the workflow composition.

ML/AI

The AI subsystem within the DTE 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 are the following:

- **Training and Deployment of ML Models**: The AI subsystem is designed for training and deploying ML models onto various infrastructures, whether cloud services or local servers. Developers focus on ML training workflows, including model tuning and validation processes. Application users primarily deploy pre-trained ML models, with the flexibility to retrain these models with new data as needed.

- **Distributed Training and Optimization**: The subsystem supports distributed training mechanisms, metrics logging, a registry for models, and Hyper-Parameter Optimisation (HPO) processes. This infrastructure ensures that ML models are trained efficiently and effectively, meeting the specific requirements of different use cases.
- **User Interfaces for Different Expertise Levels**: Addressing diverse user expertise levels, the subsystem provides dual user interfaces:
    - DT Developer Profile: For experienced users or ML researchers who need comprehensive control over the ML workflow, including custom losses, metrics, neural network architectures, and ensemble methods. They interact directly with frameworks such as PyTorch, TensorFlow, and MLflow, and manage non-standard data formats.
    - Scientist Profile: For users with minimal experience in ML workflows, offering high-level definitions of ML tasks with largely automated engineering aspects. Specific requirements can be delegated to a DT developer.

When training a ML model, the first step is to acquire a dataset and divide it into sections for training and validation. In online learning, the dataset arrives as a continuous stream. The specifics of the ML model, including loss functions, evaluation metrics, network architecture, and optimiser types, are set up through a PaaS user interface. Developers can choose from various tools designed for distributed training and HPO. After the training, the model's performance is assessed using the validation dataset, and the results and relevant metrics are logged for review. The most effective models are then stored in a model registry.

The model is ready for deployment after domain-specific validation using the DTE's "Quality and Uncertainty Tracing" module. Users can choose a pre-trained model from the registry to integrate into the overall DT inference workflow. Multiple versions of the model may be available, allowing users to select the most suitable one for their live digital twin model. Once fully deployed, the digital twin will process real-world data streams, enabling experimenters to interact with it, conduct experiments, and generate predictions.

The ML/AI subsystem closely integrates with the data fusion process. It utilises prepared and formatted datasets for training and inference. Additionally, it interacts with big data analytics to leverage large-scale data processing capabilities for improved model training and evaluation. **Figure 5** illustrates the interplay between the ML training and deployment modules and other subsystem components, highlighting the comprehensive architecture that supports diverse AI workflows.

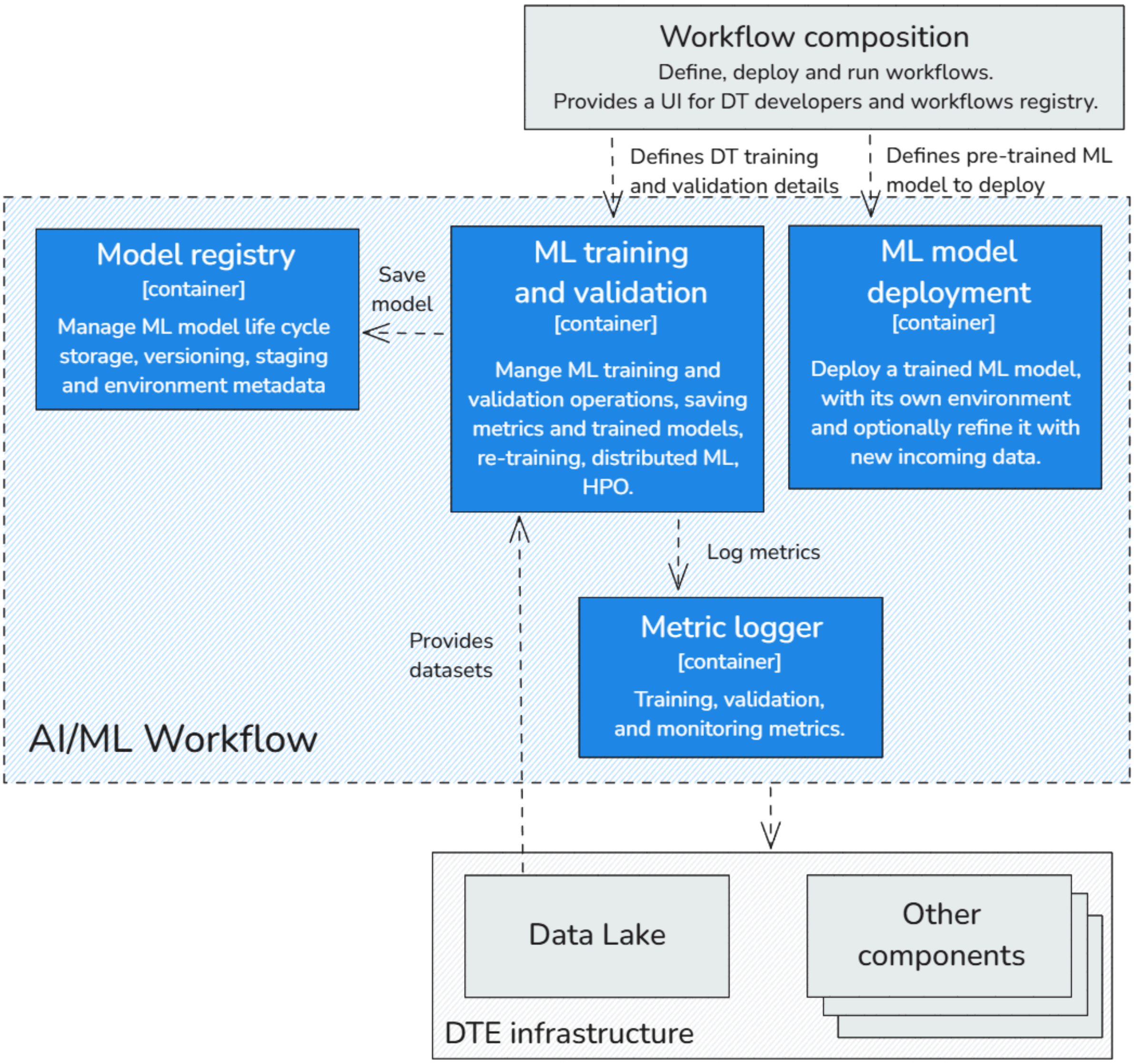

Workflow composition
Define, deploy and run workflows.
Provides a UI for DT developers and workflows registry.
Defines DT training and validation details
Defines pre-trained ML model to deploy
Model registry
[container]
Manage ML model life cycle storage, versioning, staging and environment metadata
Save model
ML training and validation
[container]
Mange ML training and validation operations, saving metrics and trained models, re-training, distributed ML, HPO.
ML model deployment
[container]
Deploy a trained ML model, with its own environment and optionally refine it with new incoming data.
Log metrics
Provides datasets
Metric logger
[container]
Training, validation, and monitoring metrics.
AI/ML Workflow
Data Lake
Other components
DTE infrastructure

Figure 5. AI/ML Workflow diagram

Big Data Analytics

Big Data Analytics within the DTE focuses on constructing templates and managing resources for large-scale data processing, providing the necessary infrastructure and tools for executing complex analytics tasks efficiently. This component handles vast amounts of data and derives meaningful insights to support DT operations.

The characteristics of Big Data Analytics are the following:

- **Template Construction and Resource Management**: The system constructs templates that map out various virtual resources and software elements required for application deployment. Users manage some configurable options that allow for the personalisation of the application's setup. These templates are maintained in a repository accessible through an Orchestrator Dashboard.
- **Orchestrator Dashboard**: The dashboard displays the templates and allows users to adjust specified settings before initiating deployment. The Orchestrator arranges the cloud resources, sets up the chosen analytics tools, and provides users with the necessary details to connect to the application.

- **Automation and Modularity**: Configuration recipes for analytics tools are written in an automation-specific language, encapsulated as modular components, and stored in a public repository. These components are designed for reuse across different configurations, enhancing the system's flexibility and scalability.
- **Resource Allocation and Optimization**: The Orchestrator manages resources from various providers and orchestrates the template deployment, selecting the most suitable provider based on data location, service level agreements, and monitoring information. It also provides APIs for managing these deployments.
- **User-Friendly Interface**: The dashboard associated with the Orchestrator is a web-based application that gives users a straightforward way to engage with the PaaS services, particularly for creating deployments. It is designed for ease of use, allowing users to manage and monitor their deployments without complication.

Big Data Analytics involves constructing templates detailing the software and infrastructure needed to execute analytics tools. Users can customise these templates through an Orchestrator Dashboard, which provides a user-friendly interface for managing deployment settings. Once the settings are configured, the Orchestrator handles the arrangement of cloud resources, sets up the chosen analytics tools, and provides users with access details.

Configuration recipes written in an automation-specific language are stored in a public repository as modular components. Similar to libraries in traditional programming, these components are designed for reuse across different setups, promoting flexibility and efficiency. A central service enables users to find and implement these roles in their automation scripts.

The Orchestrator is responsible for resource allocation, ensuring the most suitable provider is selected based on various factors such as data location and service agreements. It also monitors deployments to optimise performance and resource utilisation.

Big Data Analytics integrates closely with the ML/AI subsystem, providing the infrastructure for large-scale data processing necessary for training and deploying machine learning models. It also interacts with data fusion processes, ensuring that large, heterogeneous datasets are effectively processed and analysed. Figure 6 provides a visual representation of the architecture and interrelations of the components involved in setting up and using data analytics tools within this system.

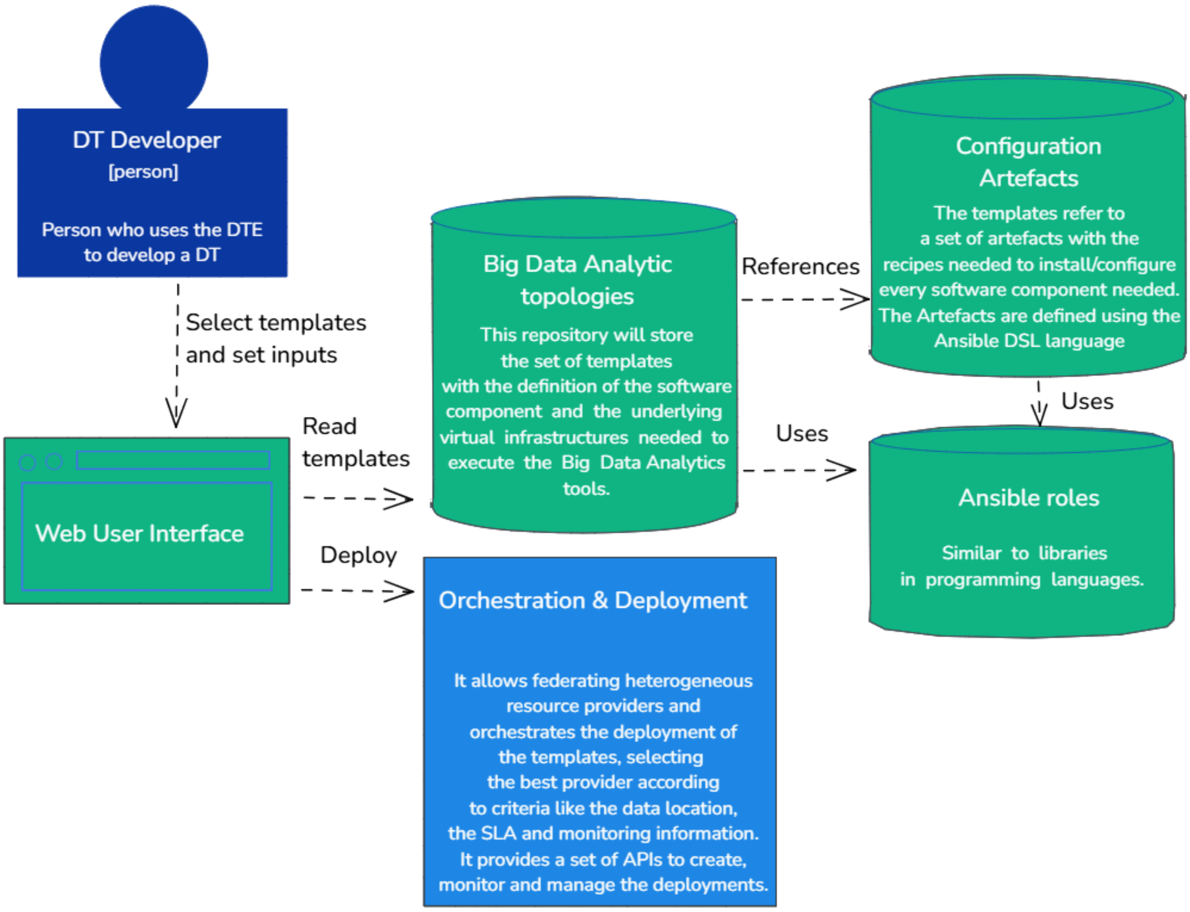

DT Developer
[person]
Person who uses the DTE to develop a DT
Select templates and set inputs
Web User Interface
Read templates
Deploy
Big Data Analytic topologies
This repository will store the set of templates with the definition of the software component and the underlying virtual infrastructures needed to execute the Big Data Analytics tools.
References
Configuration Artefacts
The templates refer to a set of artefacts with the recipes needed to install/configure every software component needed. The Artefacts are defined using the Ansible DSL language
Uses
Uses
Ansible roles
Similar to libraries in programming languages.
Orchestration & Deployment
It allows federating heterogeneous resource providers and orchestrates the deployment of the templates, selecting the best provider according to criteria like the data location, the SLA and monitoring information. It provides a set of APIs to create, monitor and manage the deployments.

Figure 6. Big Data Analytics Overview Diagram

3.5.3. Software Quality Assurance

Quality Verification is a critical aspect of the development life cycle for digital assets, such as source code, (web) services, and data. It ensures that potential failures are identified promptly, allowing immediate remediation. The concept of Software Quality Assurance as a Service (SQAaaS) plays a pivotal role in this process. It enables the composition of robust CI/CD pipelines, which serve as quality gates, validating the workflow of a DTE.

The CI/CD pipelines are crafted to be triggered by specific events or on-demand, providing flexibility for thorough quality checks. For instance, they can be initiated as the final acceptance test for a pre-trained ML model before deployment into production. Alternatively, they may be activated in response to events such as data ingestion or updates to the model's source code in the repository.

As described in **Figure 7**, the architecture of the SQAaaS module contains several components. The “core” component is the SQAaaS API, offering functionalities for composing CI/CD pipelines and assessing the quality of digital assets through a fixed set of criteria and tools. This dual approach includes Quality Assessment and Awarding, which gives a broad analysis and recognition of digital objects, and Pipeline as a Service, which aids in customising CI/CD pipelines to meet specific quality standards.

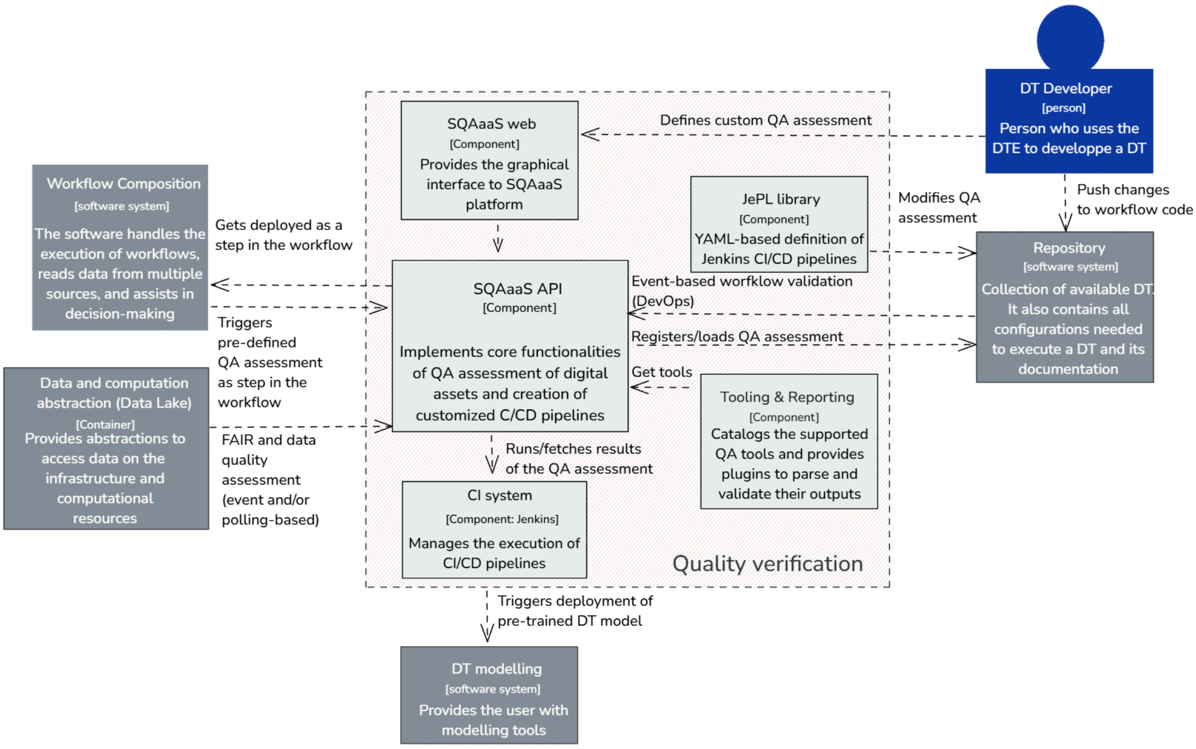

DT Developer
[person]
Person who uses the DTE to developpe a DT
Defines custom QA assessment
SQAaaS web
[Component]
Provides the graphical interface to SQAaaS platform
Workflow Composition
[software system]
The software handles the execution of workflows, reads data from multiple sources, and assists in decision-making
Gets deployed as a step in the workflow
JePL library
[Component]
YAML-based definition of Jenkins CI/CD pipelines
Modifies QA assessment
Push changes to workflow code
Repository
[software system]
Collection of available DT. It also contains all configurations needed to execute a DT and its documentation
SQAaaS API
[Component]
Implements core functionalities of QA assessment of digital assets and creation of customized C/CD pipelines
Event-based worfklow validation (DevOps)
Registers/loads QA assessment
Get tools
Triggers pre-defined QA assessment as step in the workflow
Data and computation abstraction (Data Lake)
[Container]
Provides abstractions to access data on the infrastructure and computational resources
FAIR and data quality assessment (event and/or polling-based)
Runs/fetches results of the QA assessment
Tooling & Reporting
[Component]
Catalogs the supported QA tools and provides plugins to parse and validate their outputs
CI system
[Component: Jenkins]
Manages the execution of CI/CD pipelines
Quality verification
Triggers deployment of pre-trained DT model
DT modelling
[software system]
Provides the user with modelling tools

Figure 7. Quality verification schema diagram

The SQAaaS tooling and reporting component supports the SQAaaS API server and provides integrated open-source tools for validating individual quality criteria. These tools are selected for their popularity and community support and complemented with plugins to validate their outputs. As the DTE evolves, new libraries and tools are incorporated to enhance data quality and model validation, helping developers identify and resolve issues early on.

The SQA library links the CI/CD pipelines with the CI solution, using YAML descriptions to define the pipeline work. This library utilises containers to create the necessary environment for executing checks associated with each quality criterion.

The elements shown in Figure 7 are the following:

- **Workflow Composition**: This is the core where the CI/CD pipelines are defined, deployed, and executed, often in response to specific triggers such as code changes or manual initiation.
- **Data and Computation Abstraction (Data Lake)**: This segment provides a layer that abstracts the underlying data and computational resources, ensuring that the CI/CD pipelines can operate without direct dependencies on specific technologies.
- **AI/ML Subsystem**: This part provides AI and ML operations tools, such as training and hyperparameter optimisation, which integrate with the CI/CD pipelines for on-demand checks or event-triggered actions.
- **SQAaaS Web Interface and API**: These components offer graphical and programmatic interfaces to facilitate the composition of the CI/CD pipelines and the assessment of digital assets' quality.

- **CI System**: While the specific technology is not mentioned, it's shown to manage the execution of the CI/CD pipelines, highlighting the architecture's technology-agnostic nature.
- **Tooling & Reporting**: These block catalogues support QA tools and provide plugins to parse and validate their outputs, which is crucial for quality verification.
- **Repository**: Indicated as storage for DTE applications, it suggests integration with version control systems.

Quality Verification within the DTE is streamlined through the SQAaaS platform, from creating customised CI/CD pipelines to deploying quality checks and integrating with advanced tooling for comprehensive quality assurance.

3.6. DTE Thematic Capabilities

The DTE Thematic modules are a powerful addition to the interTwin platform. These modules provide specialised capabilities tailored to the needs of specific application groups. They are designed to be of general applicability to multiple "adjacent" communities and developed to be promoted as Core modules after successful adoption by multiple resource communities from different domains.

One of the key strengths of the DTE Thematic modules is their versatility. They can be applied to the wide range of scientific disciplines targeted in this project. They are designed to be easy to use and flexible, making them an ideal tool for researchers in many fields. Examples of how the DTE Thematic modules can be used in these fields are shown below.

In high-energy physics, the DTE Thematic modules can analyse data from particle accelerators and related experimental devices. They can identify patterns and correlations in the data, which can help researchers understand the properties of subatomic particles and the fundamental nature of the universe.

In radio astronomy, the DTE Thematic modules can analyse data from radio telescopes. They can identify patterns and correlations in the data, which can help researchers understand objects' properties and the universe's structure.

In gravitational waves astroparticle physics, the interTwin DTE Thematic modules can be used to analyse data from experiments that study the properties of particles in the universe. They can be used to identify patterns and correlations in the data, which can help researchers understand the properties of the universe and its origins.

The interTwin DTE Thematic modules can analyse data from weather and climate models in climate research. They can identify patterns and correlations in the data, which can help researchers understand the Earth's climate and predict future climate change.

In environmental monitoring, the interTwin DTE Thematic modules can analyse data from environmental sensors. They can identify patterns and correlations in the data, which can help researchers understand the state of the environment and predict potential environmental hazards.

Lattice QCD simulations and data management use the following modules:

- Module for ML analysis of QCD configurations to optimise the input parameters in large-scale HPC simulations.

- Module to automate and generalise the parallelisation approach of the existing machine learning algorithms to generate QCD configurations.
- Module to include the usage of GANs in generating Lattice QCD configurations.
- Module to include Quantum enhancement of the ML-configuration generation algorithms.

Noise simulation for radio astronomy uses a module of ML methods for analysing time series to specify noise signals and classification according to their "complexity" (the complexity is estimated iteratively by determining how well they can be identified with ML methods) and explore the scaling behaviour and quality of distributed training on DT data sets.

Climate analytics and data processing uses a generic data gathering and filtering system to support environmental data collection from multiple data repositories, a generic module for data augmentation and spatio-temporal resolution adjustment, statistical downscaling and bias correction module using ML-based methodologies, a generic event detection algorithm module, and a specific attribution event detection module, as well as specific thematic ML and Data Mining based modules according to the climate change impacts User Stories: Statistical downscaling and Bias correction, Extreme Events Attribution, Compound Extreme Events and Automated Selection of Climate Simulations.

Earth Observation Modelling and Processing uses vector-based processes in openEO, vector neighbourhood analysis tools in openEO, GAP filling processes in openEO, improved versions of spatial and temporal resampling and re-gridding processes, near real-time automated triggering of EO processes, integrates SAR-based global flood monitoring toolchain in openEO workflows.

Hydrological model data processing automates the workflow to develop local flood hazard and impact models, connecting the forecasting engine to processing pipelines, assessing the availability of real-time satellite information, connecting the local flood hazard and impact models to processing pipelines, connecting the forecasting engine to local data, providing an intuitive interface for users to create and run the flood early warning workflow, integrating the flood early warning digital twin component, and connecting to DestinE project including input and output data ingestion.

For fast simulation with GAN, the project will be developing a GAN-based model and optimisation techniques for High Energy Physics detectors, developing the GAN-based data generation methods, developing the GAN-based simulation methods, and evaluating the performance of the GAN-based simulation methods.

3.7. Security and Privacy

Security is a paramount concern for any platform handling critical data and operations. The DTE infrastructure must guarantee data protection, process integrity, and user privacy. Implementing an AAI is crucial to maintaining high-security levels, regardless of where the infrastructure is deployed and for managing authentication and authorisation processes within the DTE, providing a centralised system that ensures secure access to all platform components, allowing users to authenticate once and access all necessary tools and services.

Centralised Authentication and Authorization

All DTE components should integrate with a single authentication system. This enables users to access multiple services with one login session, using a Single Sign-On (SSO) system to move between different tools without re-entering their credentials. Centralised authorisation is also essential, ensuring only users with the appropriate permissions can access specific resources. Authorisation services verify user permissions based on defined roles and policies before granting access to services or data.

The Authentication and Authorization Solution

The AAI is a system designed to manage user identities and control access to various services. It ensures secure, seamless login experiences across platforms through the use of SSO and federated identity management, allowing users to authenticate with their institutional credentials or other trusted identities. A possible solution for AAI is EGI Check-in.

EGI Check-in integrates with multiple Identity Providers (IdPs) and federations, allowing users to authenticate using their institutional credentials or other trusted identities. The process works as follows:

1. User Authentication: When users attempt to access a DTE service, they are redirected to the EGI Check-in service.
2. IdP Selection: The user selects their home institution or preferred identity provider from a list.
3. Credential Verification: EGI Check-in redirects the user to the chosen identity provider for authentication. The identity provider verifies the user's credentials.
4. Token Issuance: Upon successful authentication, the identity provider issues an authentication token, passed back to EGI Check-in.
5. Authorization: EGI Check-in checks the user's roles and permissions against predefined policies to determine access rights.
6. Access Granted: The user is granted access to the requested DTE service, and a session token is issued to facilitate SSO for other services.

This process ensures that users can authenticate once and access multiple services without re-entering their credentials. Additionally, EGI Check-in supports various authentication protocols, including SAML2[27], OAuth2[28], and OpenID Connect[29], making it compatible with multiple applications and services.

Session Management and Security Tokens

Efficient session management is necessary to control inactivity timeouts and session expirations, protecting against unauthorised access due to open sessions. Configuring session expiration policies and managing security tokens ensure that inactive sessions are automatically terminated. Security token management includes issuing, renewing, and validating tokens to maintain active sessions and authorise user actions. Proper

[27] https://en.wikipedia.org/wiki/SAML_2.0

[28] https://oauth.net/2/

[29] https://openid.net/

token management practices minimise the risk of session hijacking and unauthorised access.

Authentication Mechanisms

Authentication is the process of verifying the identity of a user or system. In a distributed environment, best practices for authentication highlight three principal solutions:

1. SSO: SSO is a solution wherein the user, upon login, is redirected and signs into an IdP, which authenticates them and creates an active session. The application the user is trying to access delegates authentication to the IdP. This allows the user to access multiple applications with one set of credentials. An example of this solution is the Google services environment, where users can switch between services such as Gmail, Drive, and YouTube without logging into each separately.
2. Federated Identity: In a federated identity scenario, users can log into a system using another system's credentials. This involves connecting two or more Identity Providers to create a trust chain. A real-life example is logging into a website with a Google or Facebook account. Identity Federations can implement SSO, but it does not necessarily imply SSO across different identity providers.
3. Delegated Identity: This system outsources user authentication to a third-party system. It shifts the user management issues and related problems (except identification and authorisation) to the IDP. An example is using Facebook Connect to authenticate users within an application. Delegated Identity is used in SSO but does not necessarily imply SSO.

These solutions are not mutually exclusive and can be combined. For example, SSO can be implemented with federated identity, and both can share components like the IdP.

Federation Requirements

For efficient interoperability, a standard identity federation protocol must meet the following requirements:

1. Availability in Off-the-Shelf Solutions: The protocol should be implemented in various ready-to-use solutions, including open-source ones, to avoid the need for developing custom protocols.
2. Widespread Adoption: It should be widely used in research and other real-world cases to facilitate integration with a broad range of systems.
3. Support and Maintenance: The protocol must be actively supported and maintained, avoiding obsolete or deprecated solutions.
4. Compatibility with Modern Technologies: It should work seamlessly with microservices, RESTful APIs, and cloud environments, supporting migrating services to distributed cloud infrastructures.
5. Usability by Humans and Applications: The protocol must enable human users and applications to authenticate and automate workflows without human intervention.
6. Authorization Handling: It should manage authorisations to secure access to APIs and other online resources.

Recommended Protocols
Among the well-known protocols, OAuth2 and OpenID Connect are highly recommended due to their compliance with the above requirements and widespread use. These protocols provide robust mechanisms for authentication and authorisation, facilitating secure and seamless integration across different systems.

Governance and Policy Framework
A clear policy framework is essential for the successful implementation of AAI. This includes security, privacy, and trust policies, ensuring compliance with regulations such as General Data Protection Regulation (GDPR[30]). Additionally, asset-related policies must be defined, covering data, data products, services, and software. These policies guide the IT subsystems for curation, provenance, and access control.

To reduce complexity, a simplified policy framework is advisable. For example, having two levels of users—administrators and normal users—can streamline the implementation of AAI components. The policies should be as simple as possible but as complex as necessary to meet organisational requirements.

Security in Cloud and On-Premise Environments
In cloud environments, organisations often leverage the security features provided by cloud services, including firewalls, Intrusion Detection Systems (IDS), and encryption. While cloud providers must comply with relevant security and privacy regulations, such as Europe's GDPR, it is essential that interTwin components integrate with these security features to ensure compliance and data protection.

In on-premise environments, organisations must internally implement and manage all security systems. This includes configuring firewalls, Intrusion Detection and Prevention Systems (IDPS), and backup and recovery mechanisms. On-premise security management allows greater control over the physical and logical infrastructure, enabling customised security measures to meet specific organisational needs. However, maintaining a robust security posture also requires significant resources and expertise.

4. Alignment to Destination Earth

This section introduces the European Commission initiative, Destination Earth. It reports the activities of architecture alignment and mapping to interTwin that have been carried out until the second version of this deliverable [R8].

The Final report on the activities and pilots between interTwin and DestinE will be included in the deliverable D3.7 led by ECMWF and due at M37.

4.1. Destination Earth (DestinE)

Destination Earth, often called DestinE, is an initiative committed to establishing a high-fidelity digital replica of our planet. This ambitious undertaking strives to deepen

[30] https://gdpr-info.eu/

our comprehension of climate change impacts and environmental catastrophes, thereby equipping policymakers with robust tools to devise more effective responses.

DestinE aligns with the European Commission's dual priorities, digital and climate transitions, to achieve carbon neutrality by 2050. As part of the Digital Europe Programme and the European Green Deal, the initiative is set to construct a high-precision simulation of Earth, a digital twin, incorporating real-time data from various sources, including climate monitors, meteorological sensors, atmospheric probes, and behavioural indicators.

This digital model of Earth is designed to serve many user groups, ranging from the public sector to scientific communities and private enterprises. By allowing the monitoring and simulation of natural and human activities, the system paves the way for developing various test scenarios to predict climate change's consequences more accurately.

DestinE's relevance to the pressing climate change challenge is unquestionable. Using the sophisticated modelling capabilities provided by DestinE, the European Commission, within the framework of the European Green Deal and individual nations, can conduct comprehensive assessments of the environmental impact and efficacy of legislative proposals.

4.1.1. DestinE Components

The overall goal of our project is to align its architecture with the one envisioned by DestinE. The construction of DestinE architecture is delegated to three organisations, each entrusted with a different part of the architecture. **Figure 8** shows a diagram that illustrates the three organisations and their main characteristics.

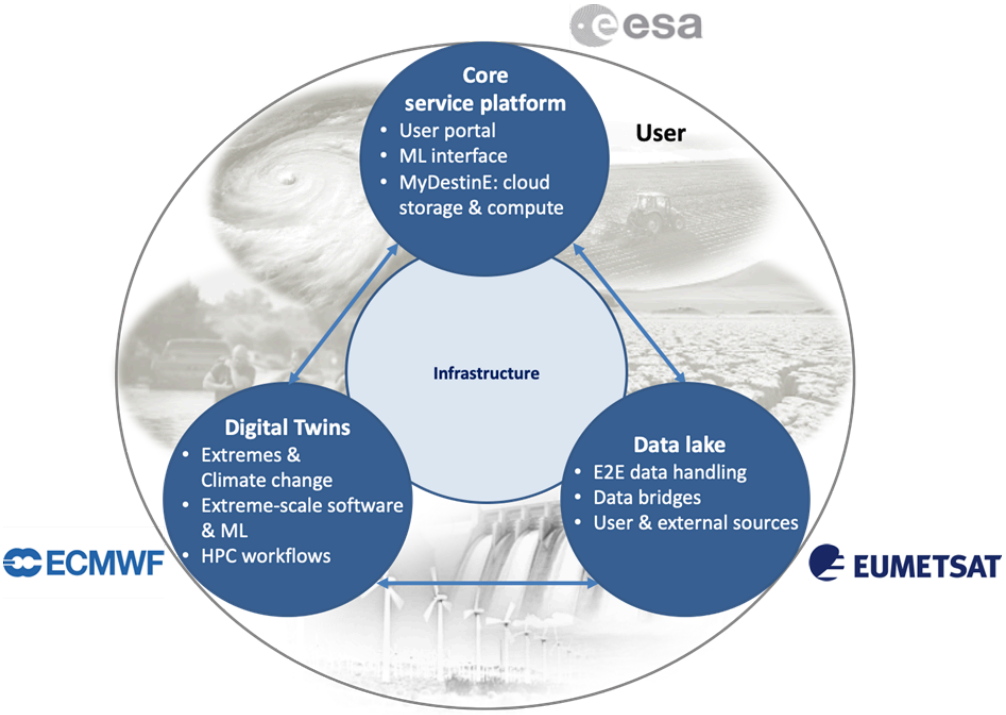


Figure 8. DestinE Organisational Responsibilities

- **DestinE Digital Twin Engine (DestinE DTE**): The European Centre for Medium-Range Weather Forecasts (ECMWF) manages this component.
- **DestinE Data Lake (DEDL)**: Managed by the European Organisation for the Exploitation of Meteorological Satellites (EUMETSAT), the DEDL is the repository for the vast amounts of data required for the DestinE system to function. It handles the collection, storage, and distribution of meteorological and other relevant data.
- **DestinE Core Service Platform (DESP)**: Managed by the European Space Agency (ESA), the DESP delivers the necessary resources and services to operate the DestinE DTE and DEDL efficiently. It forms the backbone of the DestinE architecture by integrating various services, including data processing, visualisation, and analysis tools.

By delineating responsibilities across these organisations, the DestinE initiative ensures that experts in their respective fields develop and manage each system component, thereby fostering an efficient, robust, and sophisticated Digital Twin of Earth.

4.1.2. DestinE DTE

The DestinE DTE is an important component of the DestinE platform. It comprises the software infrastructure for running extreme-scale simulations, handling data fusion, managing data, and executing machine learning algorithms. These capabilities are instrumental in efficiently deploying and linking various digital twins to the overarching DestinE platform.

The architecture of the DTE, as illustrated in **Figure 9**, integrates multiple components that ensure seamless data flow and control across the system. The **HPC (EuroHPC)** system is the central hub, managing high-priority digital twins through components like the Integrated Forecasting System (IFS), Atlas, and Plugin System. Data is processed and stored in the Field Database (FDB), with interfaces for high-performance data access, ensuring rapid and efficient data handling. The **DE Service Platform (Control)** manages the flow of data using systems such as **ecflow** and various control mechanisms, while the **DE Data Lake (DEDL)** acts as a repository for large-scale data storage, incorporating components like the DE Data Warehouse and services that handle notifications and secure data transfer via HTTPS APIs.

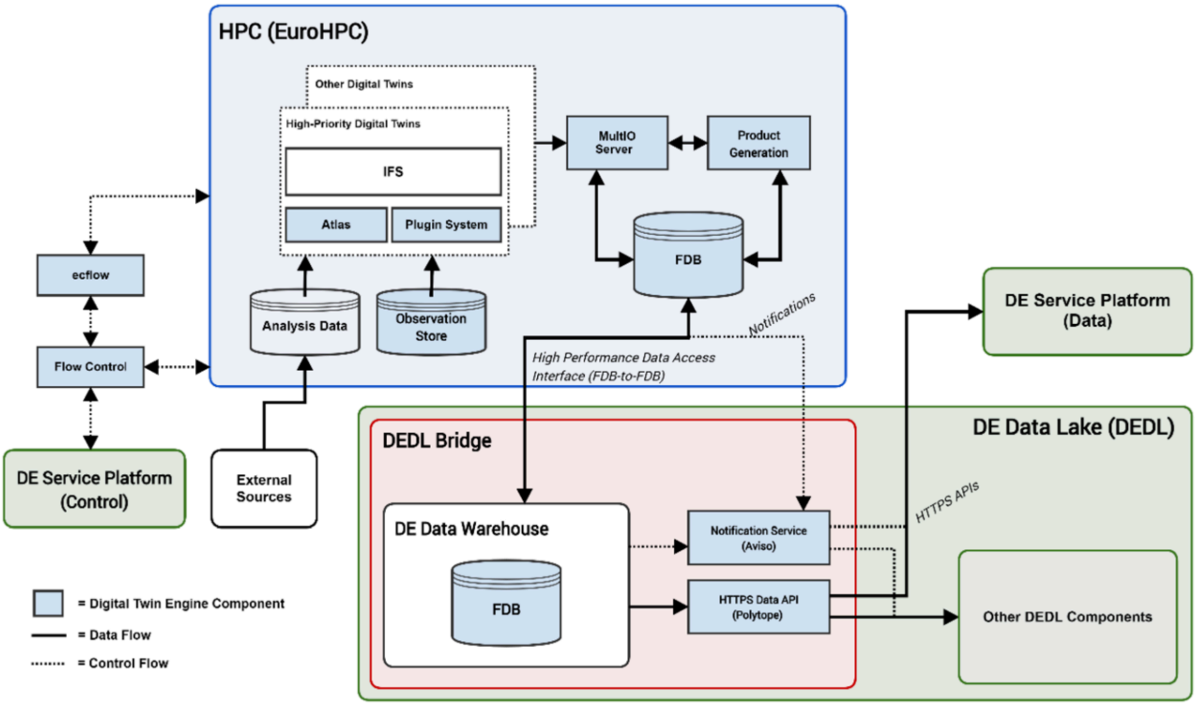


Figure 9. DestinE DTE Architecture

The DestinE DTE components, developed as opt-in modules, continually evolve to stay aligned with the dynamic standards governing data access and transformation, facilitating seamless interoperability with provided adapter hooks.

Compliance with standards is a key aspect of the DestinE DTE's design. Meteorological data aligns with World Meteorological Organisation (WMO) standards and, wherever feasible, follows Open Geospatial Consortium (OGC) standards to ensure the location information and services remain FAIR. Some of the digital twin data also adhere to directives about the availability of public datasets.

ECMWF[31] will contribute to developing and maintaining efficient data access methods. This includes providing hooks for connectors relevant to the community and ensuring interoperability with other tools (e.g., Climate Data Operators, Climate Data Store Toolbox), community software platforms (e.g., Pangeo), and infrastructure systems (Wekeo, European Weather Cloud, etc.).

In line with ECMWF's 2022 software strategy, all DTE data handling and processing components are openly developed. This encourages direct interaction with the community and ensures interoperability of standards, data formats, and APIs. Furthermore, DTE component development will leverage community software stacks and contribute back to them.

4.1.3. DestinE Data Lake

EUMETSAT[32] is responsible for designing, establishing, testing, operating, and procuring the multi-cloud DestinE Data Lake and Data Warehouse. As one of the core services, the data lake will accommodate a vast diversity of data spaces. These will include data from

[31] https://www.ecmwf.int/

[32] https://www.eumetsat.int/

EUMETSAT's Earth observation satellite systems and the European Copernicus Sentinel missions, ESA missions, and ECMWF. **Figure 10** provides a visual representation of the DestinE Data Lake's structure and its key role within the overall DestinE initiative.

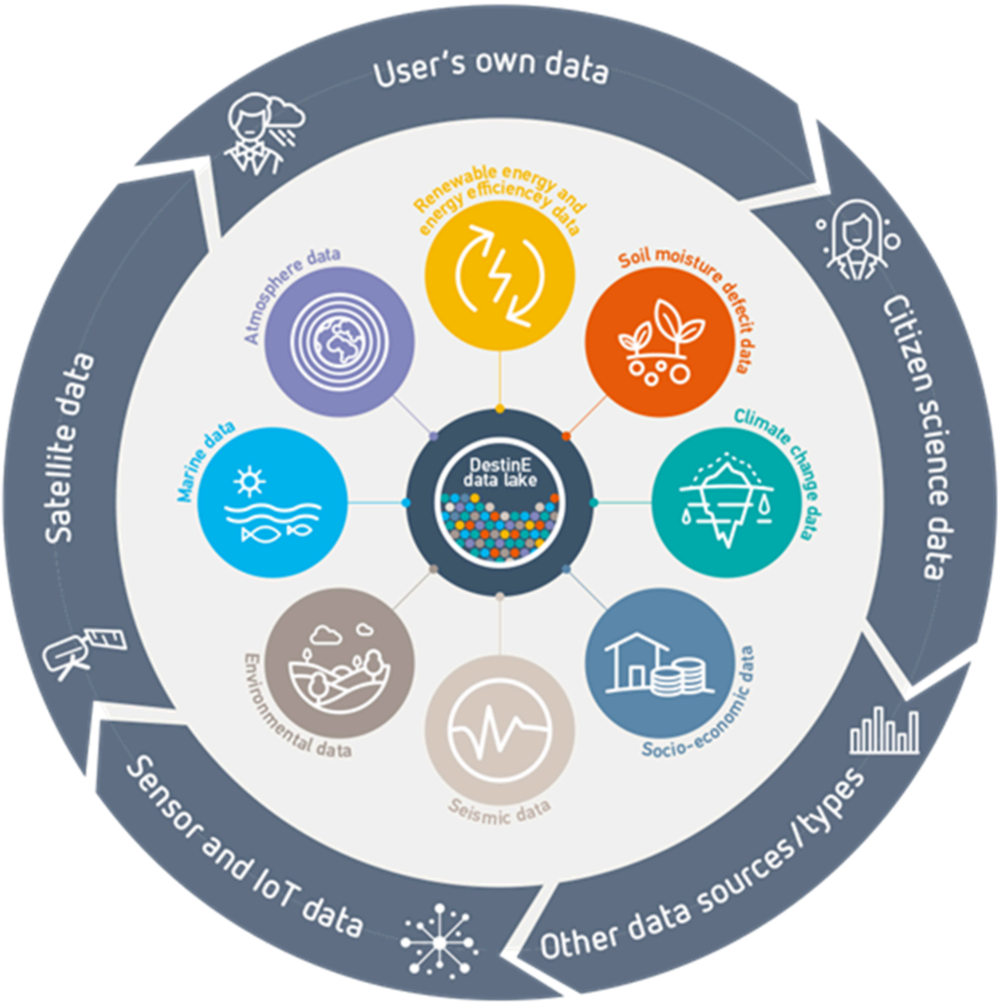


Figure 10. DestinE Data Lake Highlight

This Data Lake will act as the input source of data for the digital twin engines, AI, and machine learning algorithms to generate DestinE data and information.

The second core service provided by the data lake involves storing all DestinE data and information, making it accessible for decision-making and other users. This data will initially be available via the core service platform and directly accessible in subsequent phases.

4.1.4. DestinE Core Service Platform

ESA is responsible for developing the platform, which serves as a single access point for users of the DestinE ecosystem. The DESP[33] integrates and operates an open ecosystem of services (also called DESP Framework) to support DestinE data exploitation and

[33] https://platform.destine.eu/

information sharing for the benefit of DestinE users and third-party entities. **Fig 11** shows the entry point of DESP opened to users in October 2024.
The DESP Framework includes essential services such as:

- User identification, authentication, and authorisation service
- Infrastructure as a service with storage, network, and CPU/GPU capabilities
- Data access and retrieval service, particularly from the DestinE Data Lake operated by EUMETSAT
- Data traceability and harmonisation services
- Basic software suite service for local data exploitation
- Data and software catalogue services
- 2D/3D data visualisation services

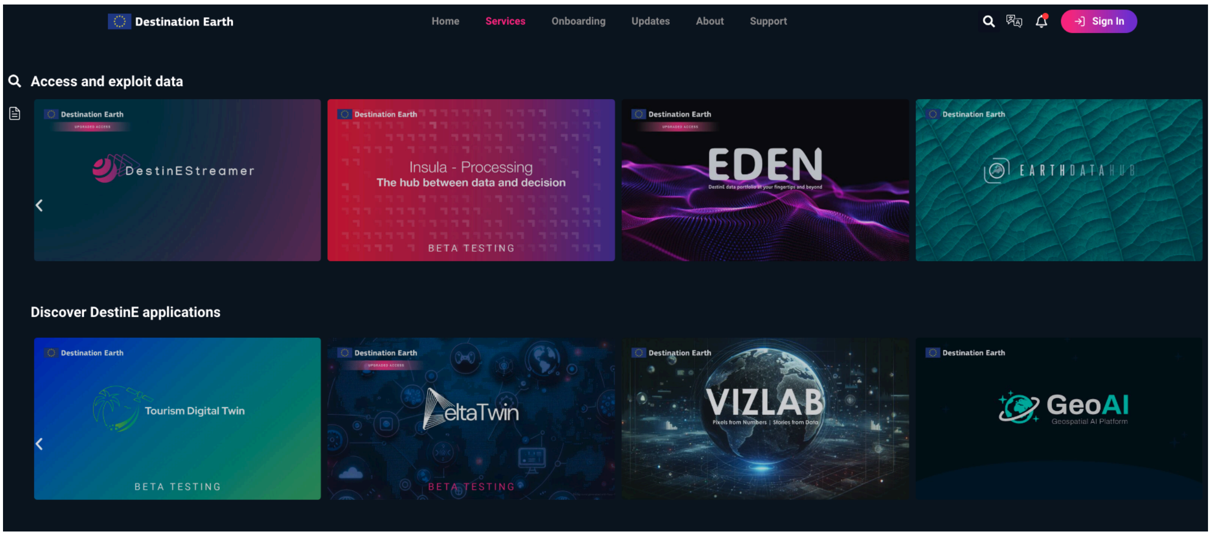


Figure 11. DESP Access point

4.2 Linking Activities with DestinE

ECMWF is leading a task in interTwin to align the architecture choices of interTwin with what DestinE is building. The task aims to study and provide suitable APIs to make the interTwin data accessible to the entire DestinE user base. This includes facilitating the harmonised data access, fusion, and analyses of a broader range of DTs with DestinE external data. As a result, the task should provide a TRL6 proof-of-concept for:

- Ingesting DestinE DT data to constrain the interTwin Earth-system thematic modules.
- Ingesting the interTwin output in the DestinE data stream feeding into the DestinE Data Lake.
- Pushing the DestinE DT uncertainty quantification into the interTwin DT.

4.2.1. interTwin Components mapping to DestinE Architecture

This section explores how interTwin's components align with and possibly complement the DestinE architecture. Synergies and potential integration points are identified through systematic comparison and mapping, enhancing both systems' functionality and scope.

DestinE C4 Architecture

The diagram on **Figure 12**, describes the high-level system context for the DestinE architecture. It has four main subsystems:

- **Service/Applications/Discovery Layer**: The front for the end-users to discover DTs or advanced applications.
- **Workflow/Software Catalogue Subsystem**: The services at this layer collect available modules, containers, and workflows that can be composed together. The upper layer uses this subsystem to access and run these components.
- **Core Capabilities Subsystem**: Includes components for data acquisition and handling, notification, and processing.
- **Orchestration/Execution Subsystem**: This subsystem executes workflows on the platforms and prepares the infrastructure needed for executions.

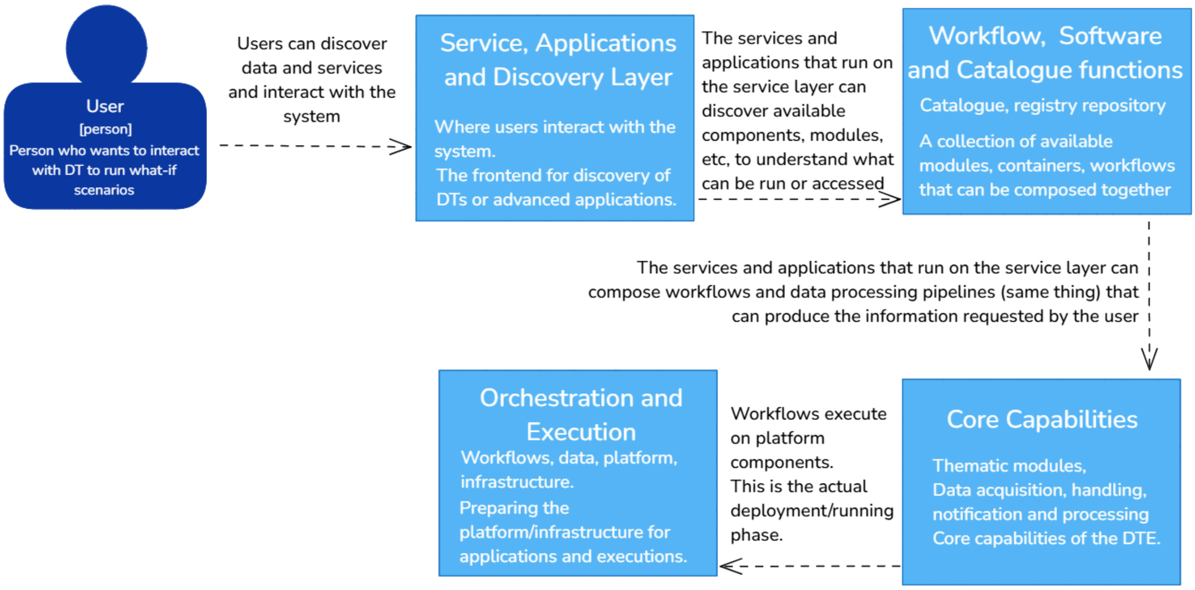

User
[person]
Person who wants to interact with DT to run what-if scenarios
Users can discover data and services and interact with the system
Service, Applications and Discovery Layer
Where users interact with the system.
The frontend for discovery of DTs or advanced applications.
The services and applications that run on the service layer can discover available components, modules, etc, to understand what can be run or accessed
Workflow, Software and Catalogue functions
Catalogue, registry repository
A collection of available modules, containers, workflows that can be composed together
The services and applications that run on the service layer can compose workflows and data processing pipelines (same thing) that can produce the information requested by the user
Orchestration and Execution
Workflows, data, platform, infrastructure.
Preparing the platform/infrastructure for applications and executions.
Workflows execute on platform components.
This is the actual deployment/running phase.
Core Capabilities
Thematic modules,
Data acquisition, handling, notification and processing
Core capabilities of the DTE.

Figure 12. System Diagram for Destination Earth

Destination Earth's primary users are end-users who access living DTs to run what-if scenarios or perform data analysis. InterTwin focuses the DTE on DT developers who access the platform to develop DTs and thematic modules for the end-users to access. Moreover, the types of DTs focus on the Earth System domain, but most DTs incorporated in the DestinE infrastructure will be multidisciplinary, covering land, marine, atmosphere, and biosphere.

The main entry point for users in DestinE is the Service Platform (developed by ESA), which includes functionalities of the Service/Applications and Discovery layer. This layer accesses and reuses the Workflows, Software, and Models Catalogues layer, which includes capabilities for:

- **Artefact repository (Workflows, Software, Models)**
- **CI/CD and Software Quality Assurance**

The Core Capabilities Subsystem defines three main containers:

- **DT Data Handling**: Components for access to the Data Lake (Harmonised data access), access to the DTs' output, and notification software.
- **DT Observability**: Components for Workflow Execution and alerting/logging of the DT Models.
- **DT Modelling**: Capabilities of Physics-based and Machine learning Modelling (also via software plugins).

The Orchestration/Execution Subsystem foresees two main containers:

- **DT Orchestration:** Components for distributed management of resources and job management.
- **Federated Data Management**: Components for object storage and fresh data pools.

The four DestinE subsystems (and the two initial DTs) have been implemented by the end of June 2024, marking the end of DestinE phase 1 implementation. The second implementation phase has been approved and started immediately after and it will last 2 more two years.

Figure 13 shows the Container Diagram from DestinE, and below are detailed the components that were developed and those that are under development as of 07/09/2024.

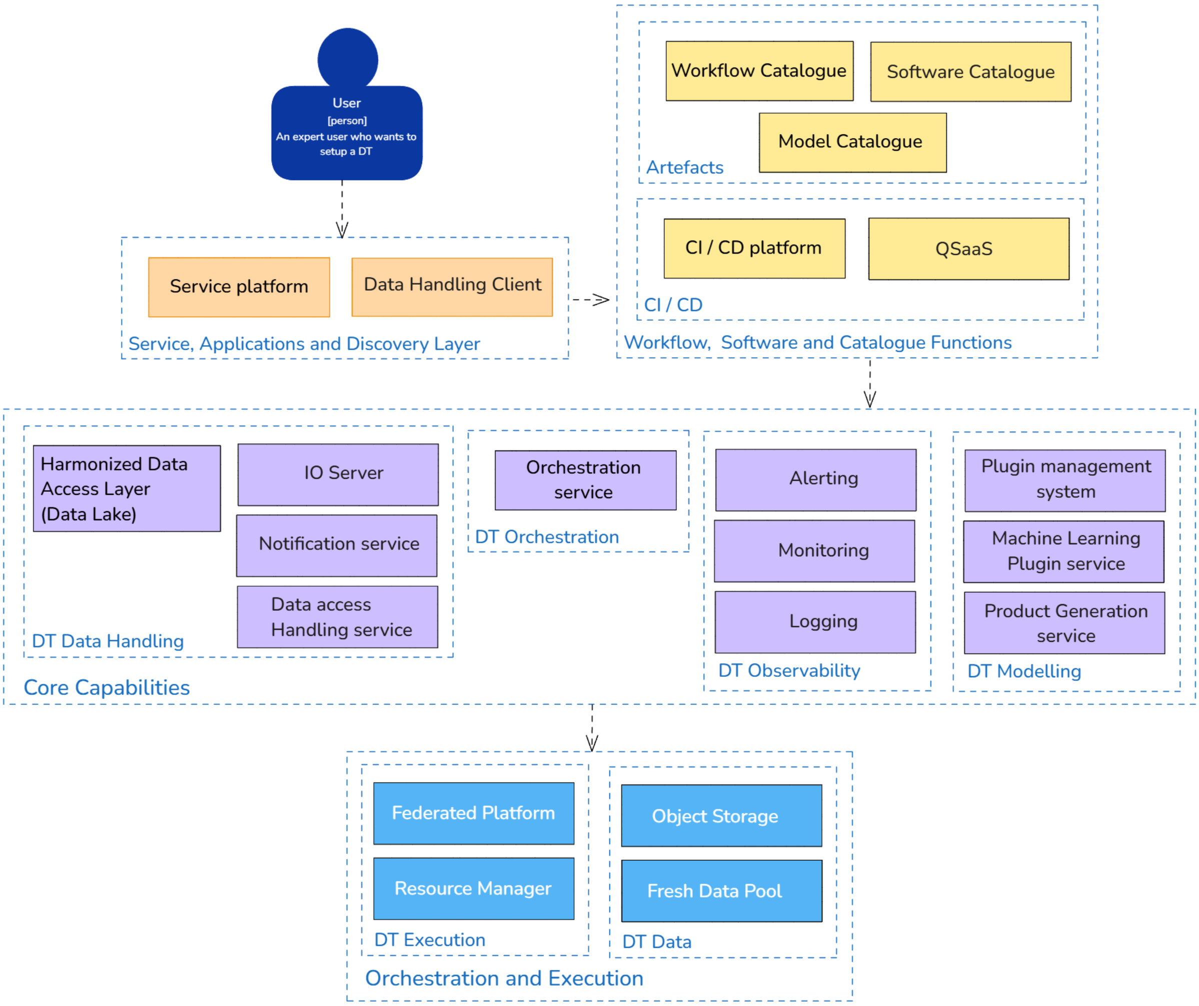


Figure 13. DestinE Container Diagram

Components already developed:

- **Polytope**: A software component to extract digital twin data using a semantic model.
- **MultIO**: A library for data transformations in memory before the data is written.
- **Aviso**: A service that notifies about specific events.
- **ECflow**: A workflow manager for weather models.
- **Plume**: A plugin system to extract data directly from the model.
- **Infero**: A plugin framework to replace a part of the simulation with an inference model.
- **PGEN**: The ECMWF product generation framework for pushing products to clients.
- **FDB**: A database to store meteorological data.

Components under implementation:

- **QSaaS**: Software quality as a service.
- **Fresh data pool**: A cache for federated data.
- **Harmonized data access layer**: A generic interface to access federated data in the data lake.

Initial Gap Analysis

To clarify the alignment and differences between interTwin and DestinE architectures, this section provides a mapping of components across both systems. In **Figure 14**, we used the main structure from **Figure 13**, which represents the DestinE container diagram, as a base. We then incorporated the corresponding components from interTwin to illustrate where the two architectures align and complement each other. This approach helps to visually map interTwin's unique elements within the existing DestinE framework, providing a clearer view of shared functionalities and distinguishing features.

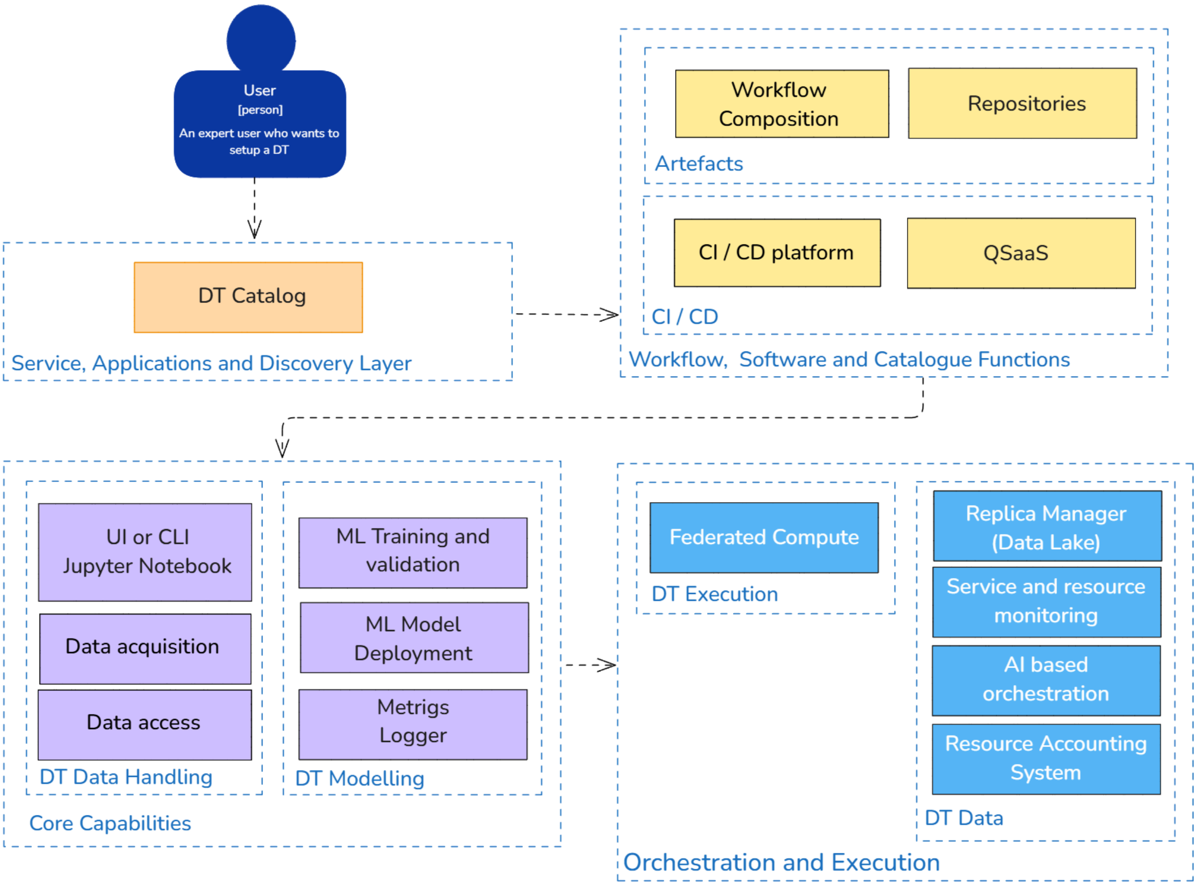

User
[person]
An expert user who wants to setup a DT
DT Catalog
Service, Applications and Discovery Layer
Workflow Composition
Repositories
Artefacts
CI / CD platform
QSaaS
CI / CD
Workflow, Software and Catalogue Functions
UI or CLI Jupyter Notebook
Data acquisition
Data access
DT Data Handling
ML Training and validation
ML Model Deployment
Metrigs Logger
DT Modelling
Core Capabilities
Federated Compute
DT Execution
Replica Manager (Data Lake)
Service and resource monitoring
AI based orchestration
Resource Accounting System
DT Data
Orchestration and Execution

Figure 14. interTwin capabilities mapped to DestinE container diagram

The main layers in both architectures, such as the Service, Applications, and Discovery Layer, Workflow, Software, and Catalogue Functions, Core Capabilities, and Orchestration and Execution, establish a foundation for alignment. However, within each layer, specific components serve unique purposes aligned with each project's goals:
Service, Applications, and Discovery Layer: interTwin's DT Catalog provides a streamlined entry point for users to access digital twin interfaces, while DestinE's Service Platform offers enhanced functionalities, including computational resources tailored for end-users exploring DT scenarios.

- **Workflow, Software, and Catalogue Functions**: Both architectures include infrastructures for repositories and workflow management. DestinE, however, goes further by incorporating continuous integration and software quality assurance, whereas interTwin emphasizes repository management with a focus on continuous integration.
- **Core Capabilities**: Data handling and modeling are core to both architectures. interTwin introduces a thematic Data Harmonization Layer that supports specialized applications, whereas DestinE's core capabilities are more generalized, allowing multidisciplinary DT applications across Earth systems.

- **Orchestration and Execution**: interTwin's approach includes hybrid resource management, utilizing both cloud and high-performance computing (HPC), and offloading mechanisms to optimize processing. DestinE incorporates batch systems to handle workflows, offering a more traditional orchestration structure. InterTwin also includes an orchestration and deployment component that was not planned for DestinE.

As already reported , we expect a full report on interTwin and DestinE in the deliverable "D3.7 Report on software architecture concepts based on DestinE and InterTwin" by the end of July 2025.

5. Relation to External Initiatives

This chapter reviews related projects and initiatives that align with the interTwin project's goals. This process is integral for recognising synergies and potentially reusing concepts or technologies.

The focus extends to various projects involved in creating DTs and those utilising IT infrastructures to develop frameworks for distributed hybrid computing and data analysis products.

The exploration aims to leverage existing knowledge, extract lessons from the challenges encountered by these initiatives, and identify opportunities for collaboration or integration. Understanding the broader landscape of similar projects and initiatives is crucial for strategically positioning interTwin within this more extensive ecosystem, thereby avoiding redundancy and ensuring the development of a DT that embodies effectiveness, efficiency, and innovation.

5.1. EOSC

The European Open Science Cloud (EOSC) is an environment for hosting and processing research data, supporting the advancement of science within the European Union. It is envisioned as a comprehensive, integrated platform facilitating the publication, discovery, and utilisation of data, tools, and services across multiple disciplines for research, innovation, and education.

Essential advancements enabled by the EOSC are:

- **Seamless Access**: Simplifying entry points for researchers to a vast array of data resources.
- **FAIR Management:** Ensuring data is Findable, Accessible, Interoperable, and Reusable, streamlining the data lifecycle management across diverse scientific domains.
- **Reliable Reuse**: Enabling confidence in the secondary use of research data, encompassing methodologies, software, and scholarly publications.

The EOSC's grand design is to establish a Web of FAIR Data and services. This foundational network serves as the base for a multitude of value-added services, including data visualisation and analytics, sustainable preservation, and the monitoring of open science practices adoption.

After 8 years of development as a system federated under one entry point (EOSC Portal), and developed by various grant projects (the last one is EOSC Future ended in June 2024), in 2024 EOSC took a change of its architectural approach and is now developed as a federation of Nodes. Each node will be a collection of resources (HW, SW and data), services and other open science assets (e.g. publications) from a country, region, and thematic area. Each node can act as an entry point to EOSC, can have one or more access portals, and can serve users directly, or in collaboration with other nodes. The following sections will describe the EOSC concepts until June 2024 and the new EOSC Node including the EOSC Procurement actions.

5.1.1. EOSC Until June 2024

Figure 15 illustrates the EOSC's High-Level Architecture as evolved till the EOSC Future project, which shows its integral components: the 'EOSC Core', which serves as the central coordination hub; the 'EOSC Exchange', which offers a diverse range of research-enabling services; and the 'EOSC Interoperability Framework', which ensures cohesive interaction across the EOSC's services and infrastructures. This is still relevant in the current EOSC landscape as the Core services and Interoperability framework have been inherited.

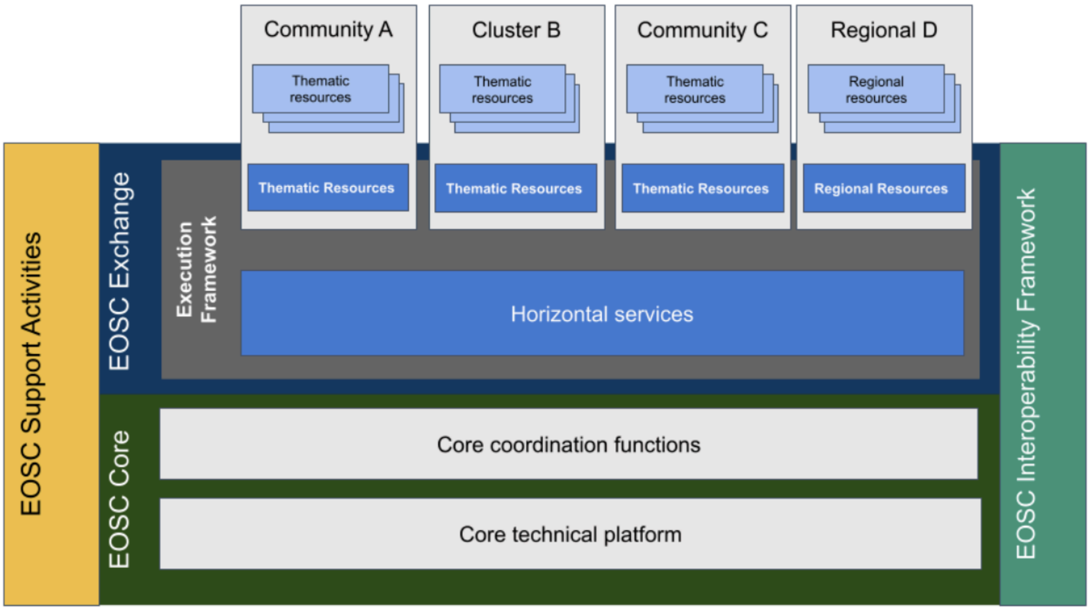


Figure 15. EOSC High-Level Architecture

EOSC Core

The EOSC Core services include the main technical platform and coordination functions for operations. The essential services are:

- **AAI:** This infrastructure offers a secure access and identity management framework, ensuring controlled and authenticated access across EOSC services.
- **Helpdesk**: It serves as a support hub for EOSC users, helping and addressing queries related to EOSC resources and services.
- **Accounting and Monitoring**: This function involves tracking and managing resources within EOSC, ensuring efficient resource utilisation and operational transparency.
- **EOSC Portal[34] and Marketplace**: The portal is the central gateway to EOSC's wealth of information and resources. It interlinks various research portals, resources, and services.

In addition to these services, the EOSC Core's capabilities were further strengthened by projects funded under the INFRAEOSC-07 call, including EGI-ACE[35], DICE[36], OpenAIRE[37] Nexus, C-SCALE[38], and RELIANCE[39]. These initiatives enhanced the EOSC's service offerings and operational efficiency.

EOSC Exchange

The EOSC Exchange is a collection of services designed to store, manage, and use research data in line with FAIR principles (Findable, Accessible, Interoperable, and Reusable). These services include data storage and preservation, data transport, and computational data analysis. Additionally, the EOSC Exchange incorporates horizontal services that are applicable and valuable across multiple disciplines or research areas rather than limited to a specific field. Various research groups use these services to foster interdisciplinary collaborations.

EOSC Interoperability Framework

The EOSC Future Project has been developing the EOSC Interoperability Framework (EOSC-IF)[40] to enhance interoperability across various EOSC components. The EOSC-IF is designed to ensure seamless interaction in two key areas:

- **Internal Interoperability within EOSC-Core** is essential for operational functionality, enabling effective communication and coordination among the EOSC's core components.
- **External Interoperability with EOSC Providers**: Integrating resources into the EOSC Exchange, which serves as the EOSC resource catalogue.

The connection of resources with EOSC Core's added-value services, including Order Management, Monitoring, Accounting, and Helpdesk.

The EOSC-IF includes interoperability guidelines to support Resource Providers. These guidelines will help Resource Providers become part of research infrastructures and work more efficiently with the EOSC-Core. The EOSC-IF builds on these efforts, creating

[34] https://eosc-portal.eu/
[35] https://doi.org/10.3030/101017567
[36] https://doi.org/10.3030/101017207
[37] https://doi.org/10.3030/101017452
[38] https://doi.org/10.3030/101017529
[39] https://doi.org/10.3030/101017501
[40] https://eosc-portal.eu/eosc-interoperability-framework

a broad framework that includes the EOSC Core and the necessary interfaces for community interoperability frameworks.

interTwin, from one side, will try to reuse some of the EOSC interoperability guidelines to be integrated with the EOSC Core, and from the other, will try to build new guidelines to be incorporated into the EOSC-IF. An example of EOSC interoperability guidelines to be followed is the EOSC AAI guidelines, built on the AARC guidelines[41]. AContributions from interTwin could be guidelines for AI Workflow management.

EOSC Compute platform and the EGI-ACE Project

The EOSC Compute Platform, an initiative of EGI-ACE, offered a comprehensive, distributed computing environment free at the point of use. This platform was architected on a robust hybrid infrastructure, integrating cloud resources, High-throughput Computing (HTC) sites, and High-Performance Computing (HPC) centres as pilots. It was designed to facilitate deploying and managing complex workflows, applications, containers, virtual research environments, and data spaces, leveraging this hybrid model.

As shown in **Figure 16**, the EGI-ACE architecture was systematically organised into functional blocks, each serving specialised roles:

- **Federated Resource Providers**: Positioned at the foundation, these providers offered a versatile infrastructure that supports research applications and data hosting. They encompass:
  - **IaaS Cloud Providers**: providing access to computing via virtual machines alongside object and block storage capabilities.
  - **HTC and HPC Centres**: substantial shared computational resources to execute large-scale jobs.
- **Core Services**: These services formed the platform's backbone, facilitating the integration and interoperability with the EOSC Core. They included:
  - **Configuration Database**: A repository holding detailed information about the federation's infrastructure.
  - **Accounting**: Tracking resource utilisation over time.
  - **Monitoring**: Ensuring resource status and availability are visible and monitored.
  - **AAI**: Implemented through EGI Check-in, offering a unified access control mechanism across all services.
  - **Helpdesk**: Offering direct human support for end-users.
- **Compute Federation**: Orchestrating user workloads across federated resources, this block enhanced data processing by optimising data locality and supports diverse computing environments through services like:
  - **Hybrid Cloud Orchestration**
  - **Workload Management**
  - **Software Distribution**
- **Data Federation**: This service layer focused on data management, providing tools for data exposure, staging, and transfer, all integral to the project operations.

[41] https://aarc-project.eu/guidelines/

- **Platforms**: Delivering value-added services that exploited the compute and storage resources, facilitating the construction of thematic end-user services. These platforms offer:
    - **Interactive Notebooks**
    - **PaaS Orchestration**
    - **AI/ML Integration**
    - **Scalable Big Data Tools**
- **Thematic Services**: These services merge capabilities from all other blocks to cater to specific research domains, emphasising data exploitation through simulation, ML, and analytics.

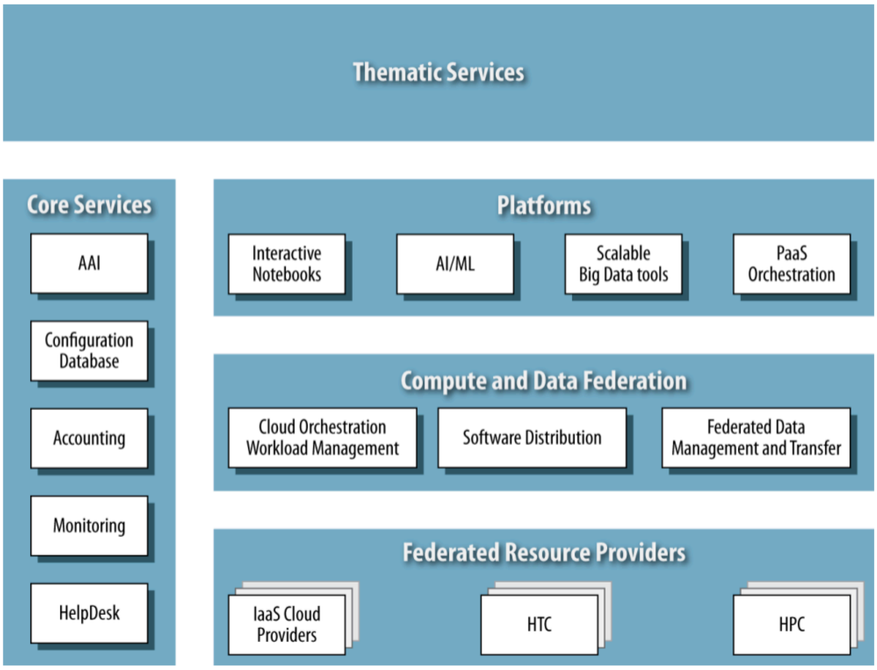


Figure 16. EGI-ACE Contribution to the EOSC Compute Platform

The EGI-ACE platform also served as a nexus for collaboration. Shared technologies and partners, such as the PaaS Orchestrator and Infrastructure Manager, evidence this synergy, fostering technological transfer and resource sharing within the wider EGI Federation.

The links between interTwin and EGI-ACE are various. They start with the project coordination by the EGI Foundation and some joint partners (e.g., UPV[42], DESY[43], INFN[44], etc.). Joint partners also mean standard technologies between the two projects. Based on the successful experience in EGI-ACE, some technologies, such as the PaaS Orchestrator and the Infrastructure Manager, are reused in the context of interTwin. Finally, the DTE infrastructure will also use some cloud providers federated in the EGI Federated Cloud.

[42] https://www.upv.es/
[43] https://www.desy.de/
[44] https://home.infn.it/it/

EOSC Federation and the EOSC EU-Node

The new EOSC Federation concept developed in 2024, will consist of multiple EOSC Nodes that are interconnected and can collaborate to share and manage scientific data, knowledge, and resources within and across those nodes.

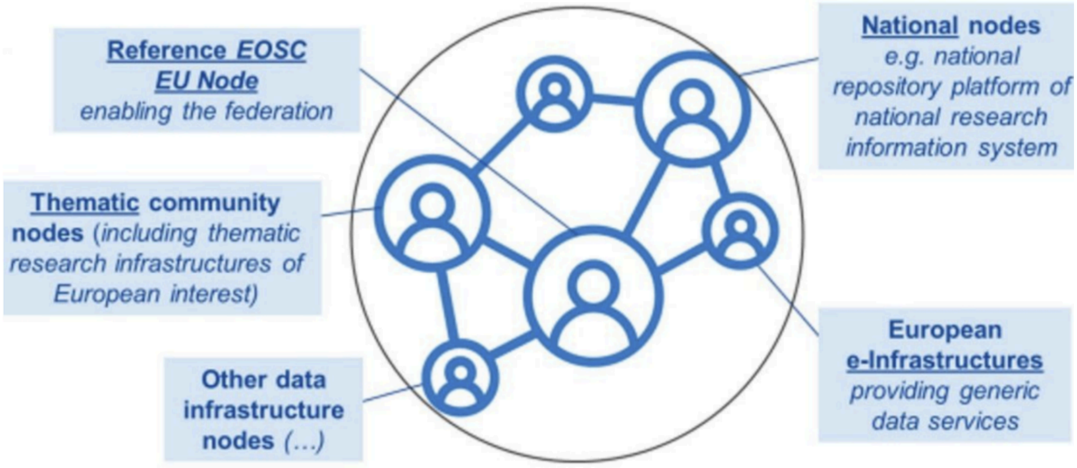

Reference EOSC EU Node
enabling the federation
Thematic community nodes (including thematic research infrastructures of European interest)
Other data infrastructure nodes (...)
National nodes
e.g. national repository platform of national research information system
European e-Infrastructures
providing generic data services

Figure 17. The Node based EOSC architecture.

The EOSC EU Node[45] is the first of the EOSC Federation and its construction is done through a 'Public Procurement project' that started in February 2024 and runs for 3 years (then extendable for a second phase). Discussions are still ongoing about all the other nodes that will complement the EOSC EU Node.

A "writing team" is working on the "EOSC Federation Handbook"[46] which intends to be the first document defining minimum requirements for EOSC Nodes, and specifying interoperability scenarios/interfaces for them. A mature draft is expected in November 2024.

The EOSC Tripartite Governance runs a questionnaire during summer 2024 to gather interest from countries, science groups and e-infrastructures concerning their ambition of establishing EOSC Nodes (called 'Candidate EOSC Nodes' at this stage). The key event for EOSC has been the EOSC Symposium[47] 2024, when the EU Node has been formally opened, updates about the Federation Handbook will be given, and further information about the candidate EOSC nodes will be given.

EOSC changed its architecture to the Node based concept in 2024. In this new landscape details are known only about the first node, the EU Node. This EU Node opened in October 2024, with the following capabilities:

- Services for researchers – These provide the 'real' services that researchers will be able to access:
- File Sync and Share: Enable automatic file syncing and secure sharing across locations and teams.

[45] EOSC EU Node: https://open-science-cloud.ec.europa.eu/

[46] EOSC Federation Handbook: https://eosc.eu/eosc-federation-handbook/

[47] EOSC Symposium 2024: https://eosc.eu/symposium2024/

- Interactive Notebooks: Create and share documents with real-time code execution.
- Large File Transfer: Streamline large file transfers online with added security and integrity.
- Virtual Machines (IaaS Cloud): Design and conduct experiments with flexibility while ensuring reproducibility.
- Cloud Container Platform: Deploy cloud-native containerised applications that can easily scale.
- Bulk Data Transfer: Move data effortlessly to data-intensive execution environments.
- Core Capabilities – These provide 'back-end' enabling services for the researcher facing services, as well as will serve external contributors (See next bullet list):
- EOSC AAI: A federated trust and identity management framework for EOSC.
- Resource Catalogues and Registry Services: Seamless connection and discovery of research objects and catalogues.
- Application Workflow Management: A workflow tool to compose and orchestrate infrastructure resources federated into EOSC.
- Monitoring and Accounting: Transparent monitoring and accounting information across EOSC.
- Helpdesk: Part of the Service Management System proposed for EOSC as a common framework for operationalised environments.

External contributors – The EU Node is expected to be able to integrate external contributors of the following types. Detailed information and rules on how to contribute will be available later:

- Repositories: Offer curated and FAIR research products with unique identifiers and minimum semantic interoperability (such as publications, data, software, etc.) that are as open as possible and as closed as necessary.
- Research Infrastructures: Contribute specialised knowledge, tools, applications, and datasets, particularly in niche or cutting-edge areas of science.
- Technology and Infrastructure Providers: Offer infrastructure and/or platform services following the cloud-based delivery model, that can form the European backbone of computational and data storage capabilities.
- Software Developers and IT Professionals: Develop the models, tools, and applications that facilitate data management, analysis, collaboration, and other essential research activities within the EOSC ecosystem.
- Industry Partners: Engage with the EOSC EU Node for collaborative research activities spanning across public and private sectors and for potentially bridging the gap between scientific research and industrialised innovation.

interTwin partners contribution to the new to the EU Node are:

- Federation capabilities that enable the whole EU Node to operate as a system of systems, meeting various qualitative and quantitative criteria defined by the European Commission.

- User facing services (most importantly Jupyter Notebooks) that enable users of the Node to define and run scientific applications that rely on the compute and storage resources of the EU Node.
- The Application Workflow Management is developed by UPV and relies on the TOSCA standards and same components reused in interTwin

Upon completing the interTwin Project, the interTwin DTE and some select DT applications will be onboarded into the new EOSC EU Node. Integrating these elements into the Node will depend on the maturity and development they achieve by the end of the project and the procedures for onboarding that are still under development at the time of writing the deliverable.

5.2. ESCAPE

ESCAPE[48] (European Science Cluster of Astronomy & Particle Physics ESFRI research infrastructures) is a Horizon 2020 project[49] that ended in January 2023. It aimed at addressing the Open Science challenges shared by ESFRI facilities and other pan-European research infrastructures in astronomy and particle physics.

ESCAPE developed solutions for the large data sets handled by the ESFRI facilities. These solutions delivered resulted in the following:

- connect ESFRI projects to EOSC, ensuring integration of data and tools;
- foster common approaches to implement open data stewardship;
- establish interoperability within EOSC as an integrated multi-messenger facility for fundamental science.

To accomplish these objectives, ESCAPE united astrophysics and particle physics communities with proven computing and data management expertise by setting up a data infrastructure beyond the current state-of-the-art to support the FAIR principles. These joint efforts resulted in a data-lake infrastructure as a cloud open-science analysis facility linked with the EOSC.

In addition to sharing some of the communities with ESCAPE (CERN and VIRGO), InterTwin could benefit from the data lake architecture (DIOS[50]) developed in the project.

The ESCAPE Data Infrastructure for Open Science (DIOS) is a federated data infrastructure that follows the FAIR data principles and provides a flexible and robust data lake to efficiently manage large volumes of data in terms of storage, security, safety, and transfer, with the basic orchestration machinery to make them accessible and be combined with high-quality data from different communities.

From the data orchestration/management point of view, DIOS comprises a bulk data transfer service and a storage orchestration service, allowing seamless access to a heterogeneous storage infrastructure. In particular, the File Transfer functionality is

[48] https://doi.org/10.3030/824064

[49] http://doi.org/10.13039/100010661

[50] https://projectescape.eu/services/data-infrastructure-open-science-dios

implemented by the FTS[51] service, and the Data orchestration is done by the Rucio[52] service, both of which were developed at CERN. FTS is a collection of servers and clients that allow for the automated scheduling and execution of remote file transfers. At the same time, Rucio is a software framework that provides functionality to organise, manage, and access large volumes of scientific data using customisable policies. **Figure 18** shows the overview of DIOS.

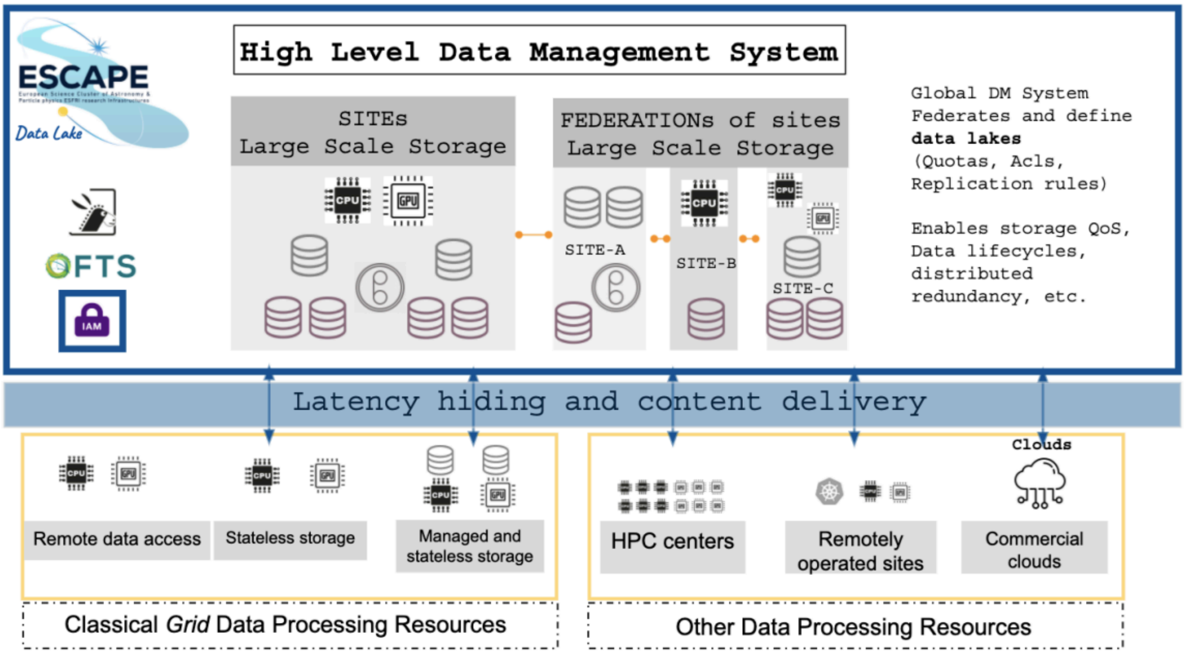

ESCAPE
Data Lake
FTS
IAM
High Level Data Management System
SITEs Large Scale Storage
CPU
GPU
FEDERATIONs of sites Large Scale Storage
SITE-A
SITE-B
SITE-C
Global DM System Federates and define data lakes (Quotas, Acls, Replication rules)
Enables storage QoS, Data lifecycles, distributed redundancy, etc.
Latency hiding and content delivery
Remote data access
Stateless storage
Managed and stateless storage
HPC centers
Remotely operated sites
Clouds
Commercial clouds
Classical Grid Data Processing Resources
Other Data Processing Resources

Figure 18. DIOS overview

The InterTwin DTE Data Management is based on DIOS, given the similarities in delivering a Federated Data management solution over a hybrid computing and storage infrastructure as interTwin is trying to build. The details of the DIOS concepts and components reused in interTwin are described in the deliverable D5.1[R4]

5.3. C-SCALE

C-SCALE is a Horizon 2020 project that aims to help European researchers, organisations, and activities by making Copernicus data, instruments, assets, and amenities simpler to find, access, and exchange. The project will be incorporated with the European Open Science Cloud (EOSC), so C-SCALE solutions may be easily incorporated into all other EOSC-supported research and development activities and procedures.

This integration joins various cross-/inter-disciplinary EOSC services, guaranteeing compatibility between distributed data catalogues, computing tools, and infrastructure. In doing so, the federation amplifies the EOSC Portal's service offer, providing

[51] https://fts.web.cern.ch/fts/

[52] https://rucio.cern.ch/

up-to-date research and enabling services to its users. It provides an open, clearly explained system for incorporating new service providers and application developers.
C-SCALE provides unique data resources and the Copernicus body of knowledge accessible to new audiences and user communities more user-friendly through the access via the EOSC portal. It provides a modular, open, and robust federation for discovering, processing, and exploiting Copernicus and, more generally, Earth observation data.
The C-SCALE project's architecture, as depicted in Figure 19, outlines the seamless integration of HTC, HPC resources, and IaaS cloud resources, all orchestrated through a PaaS model.
In **Figure 19**, SRAM (SURF Research Access Management) is a service designed to provide AAI for researchers and research support staff. The system is part of the services offered by SURF, the collaborative ICT organisation for Dutch education and research institutions. SRAM and EGI Check-In allow users a secure and simplified access point to the project's offerings. The integration of local and cloud storage solutions ensures that users have the necessary resources at their disposal for complex data processing tasks.
Moreover, the architecture features a C-SCALE Software repository, ensuring researchers can readily access the latest tools and applications.

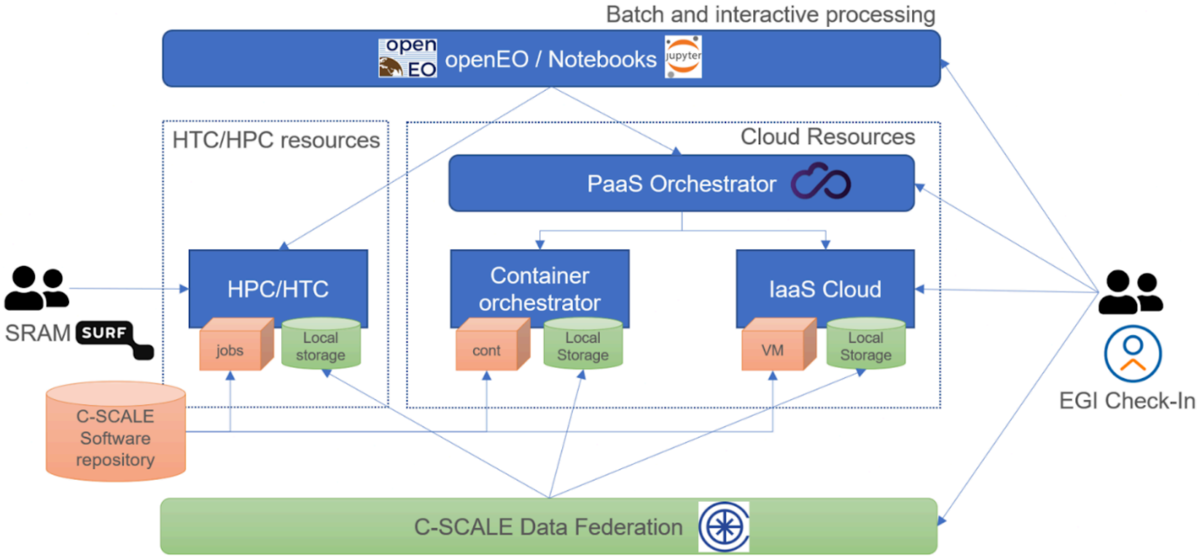


Figure 19. C-Scale Architecture

The following sections describe C-Scale components and services developed in the project that are relevant to the interTwin architecture blueprint: FedEarthData, EO-MQS, and the openEO API.

5.3.1. Federated Earth System Simulation and Data Processing Platform (FedEarthData)

The Federated Earth System Simulation and Data Processing Platform (FedEarthData[53]), a key initiative by C-SCALE, represents a significant advancement in Earth Observation (EO) research. This platform is not just a collection of tools and services; it is an ecosystem that brings together a pan-European federation of data and computing infrastructures specifically designed for Copernicus services. This integration ensures that EOSC researchers have unparalleled access to a distributed network of data and compute providers, supporting a vast array of Earth System Simulation and Data Processing workflows.

FedEarthData stands out for its cloud-based data processing capability, enabling users to efficiently create and scale data processing pipelines. These pipelines are optimised to operate in execution environments near the data sources, enhancing processing efficiency. A notable feature of the platform is its use of Jupyter Notebooks and the openEO[54] API, which collectively offers an intuitive and user-friendly means to process a diverse range of Earth Observation datasets. These tools effectively integrate various datasets with modelling and forecasting workflows, leveraging specialised computational resources to provide comprehensive analytical capabilities.

A central component of FedEarthData is its robust Data Federation service. This service provides access to a substantial archive of EO data and facilitates the discovery and accessibility of data providers under the EOSC network. By ensuring that metadata databases are searchable and product storage is accessible, the Data Federation makes it easier for researchers to find and utilise the data they need. Furthermore, the platform is enriched with an extensive collection of Copernicus datasets, all managed by the FAIR principles. This collection is dynamic, with provisions for incorporating new datasets as requested by platform users.

In addition to these core services, C-SCALE has focused on delivering a seamless user experience. This is achieved by abstracting the complexities associated with resource provisioning and orchestration, allowing researchers to concentrate on their analytical tasks without the burden of managing the infrastructure. The FedEarthData platform, along with the Earth Observation Metadata Query Service (EO-MQS) and the openEO Platform service, forms the trio of primary services offered under the C-SCALE project, each playing a pivotal role in enhancing the capabilities of the European EO research community.

5.3.2. Earth Observation Metadata Query Service (EO-MQS)

The C-SCALE Earth Observation Metadata Query Service (EO-MQS[55]) significantly advances Earth Observation (EO) data accessibility. This service, developed as part of the C-SCALE project, simplifies discovering relevant scientific data from satellite data archives across Europe, specifically within the C-SCALE Data Federation. The EO-MQS

[53] https://open-science-cloud.ec.europa.eu/resources/services/21.11166%252FBHitqi

[54] https://openeo.org/

[55] https://eo-mqs.c-scale.eu/browser

stands out as an STAC-compliant API, serving as the central interface for querying and identifying Copernicus data distributed across various partners within the federation.
One of the key features of EO-MQS is its ability to make Copernicus data, which is distributed across numerous providers within the C-SCALE Data Federation, both discoverable and searchable. This is achieved through an STAC-compliant service that exposes all available collections within the federation via a single endpoint. This simplification of data access is crucial in enhancing the efficiency of research and analysis.
Furthermore, the EO-MQS's STAC compliance ensures that it aligns with the SpatioTemporal Asset Catalog (STAC) standard[56], a widely recognised specification in the EO community. This compliance facilitates interoperability and ensures the service integrates seamlessly with tools and applications designed to work with STAC APIs. As a result, users can easily redistribute their queries among the data providers in the federation, receiving a consolidated list of results that aggregates the responses from these providers.

5.3.3. openEO

The openEO[57] platform offers a federated cloud-based environment for Earth Observation (EO) data processing, addressing the need for large-scale EO data analytics. It implements the openEO API, facilitating access to Earth Observation datasets, including Copernicus Sentinel missions. The platform's federated architecture integrates multiple infrastructures (see **Figure 20**), all supporting the openEO API, which enables diverse research and large-scale EO data applications.

[56] https://stacspec.org/en

[57] https://openeo.cloud/

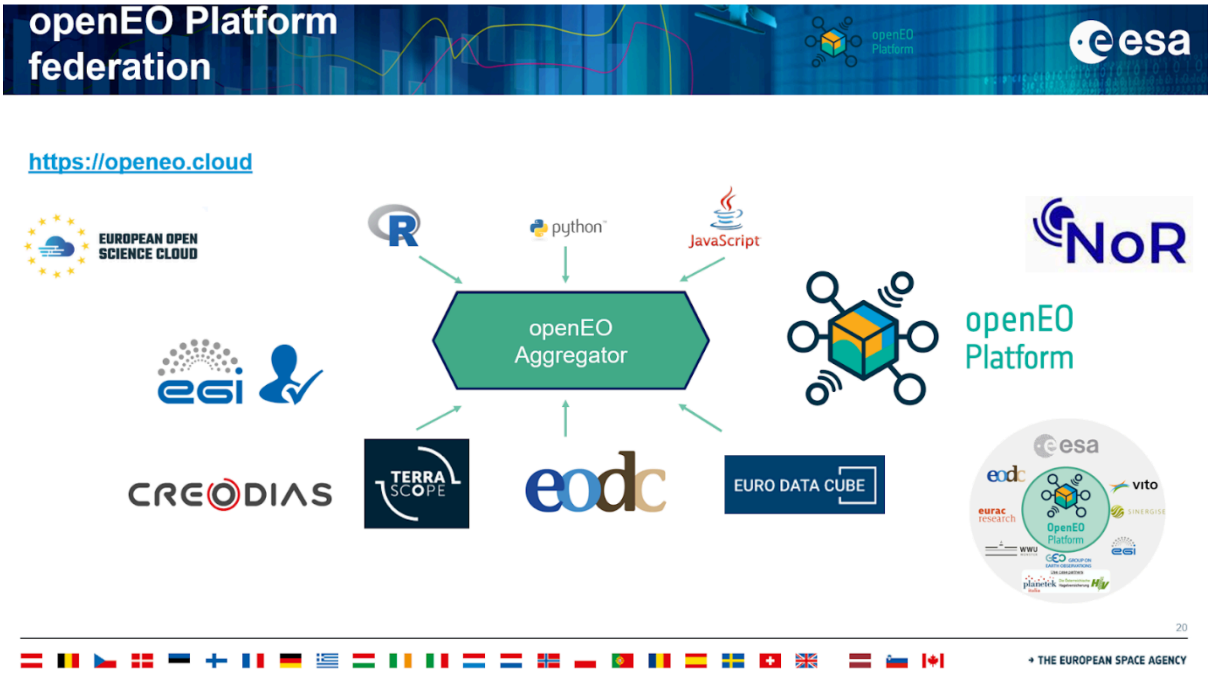


Figure 20, The federated architecture of the openEO platform, showing the integration of multiple infrastructures and the availability of different programming interfaces

Users benefit from intuitive R, Python, and JavaScript programming libraries for data processing. The platform's Jupyter notebooks offer interactive data engagement, while the openEO web editor provides a programming-less graphical interface. The collaboration among European experts in cloud operations and EO science ensures the platform's robustness, offering scalable solutions from individual pixels to continental expanses.

Fundamental principles include abstracting processing complexities, ensuring data integrity, and facilitating reproducibility. The platform supports use cases such as on-demand Analysis Ready Data for SAR and multispectral data, feature engineering, and time series analysis for change detection.

User identity management is streamlined with a single sign-on via EGI Check-In, making the platform accessible to researchers, developers, and EO specialists. It serves as an operational service for the European Space Agency and its Network of Resources (NoR), catering to private and academic sectors.

5.4. Digital Twin Consortium

The Digital Twin Consortium drives digital twin technology's awareness, adoption, interoperability, and development. It is dedicated to developing DT collaboratively with industry, academia, and government experts.

The consortium manages several working groups in various domains, organises events and establishes liaisons with several initiatives.

Some of the outcomes that could be input for interTwin are mainly related to definitions and glossaries that could help homogenise the landscape of DTs when defining the Blueprint architecture.

Figure 21 illustrates the architecture of a Digital Twin System, organised into several layers and essential components to represent, synchronise, and manage real-world data in a virtual environment. At the top layer are the applications, which include visualisation services, analytics, and other services that utilise the data and functions of the digital twin. The next layer, the integration service interfaces, provides the necessary interfaces for integrating different services. The virtual representation layer is responsible for creating and managing the virtual representation of real-world entities, combining stored representations, computational representations, and other structured and unstructured data. The following level is the IT platform, which supports the underlying technology, including software tools, platform APIs, orchestration, and fundamental components like computing, networking, and data storage. The security, trust, and governance layer is at the base of the architecture, ensuring that the system is secure, reliable, and compliant with necessary regulations, covering key aspects such as privacy, security, operational safety, resilience, and reliability. On the right, synchronisation mechanisms ensure that the virtual representation remains updated with real-world data through processes and technologies that guarantee data interoperability. Additionally, external data sources feed the digital twin with updated information. Finally, the natural world represents the physical entities and environments monitored and represented by the digital twin system. On the left, management and automation handle the tools and processes to manage and automate the system's operations. This layered organisation facilitates the efficient integration, representation, and synchronisation of complex data, ensuring the harmonious operation of the digital twin system.

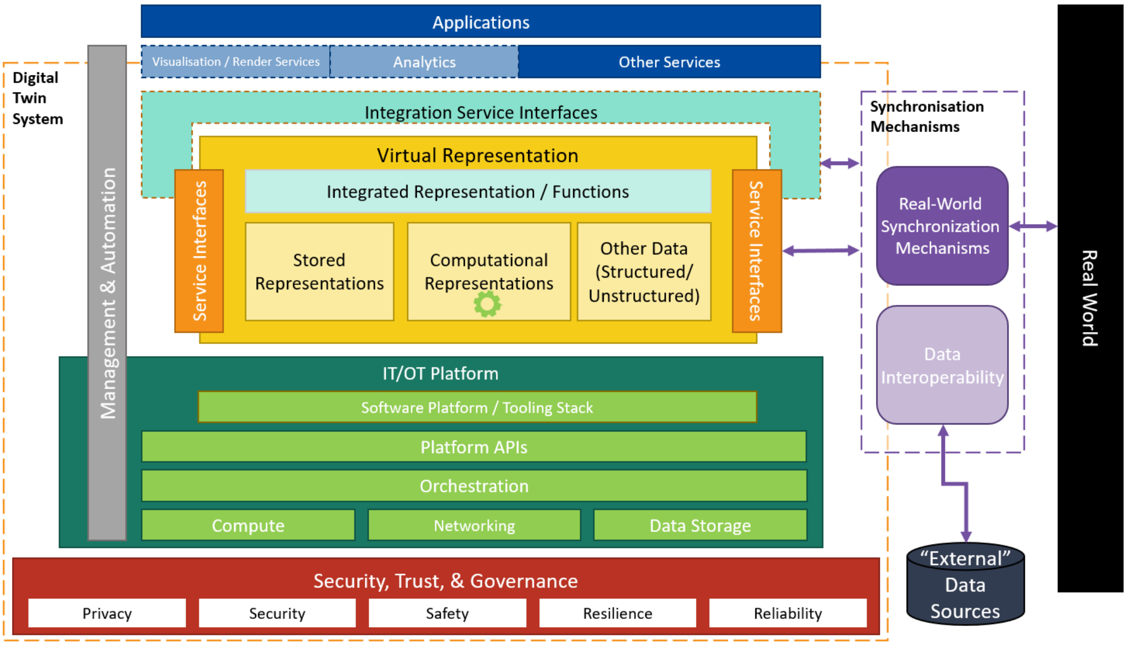


Figure 21. Digital Systems high-level architecture from DTC

5.5. EU Data Spaces

The European Strategy for Data Space initiative aims to create a single market for data that will ensure Europe's global competitiveness and data sovereignty. Common European data spaces will ensure that more data becomes available in the economy and society while keeping the companies and individuals who generate the data in control.

The EC is funding nine sectoral Data Spaces, as seen in **Figure 22**.

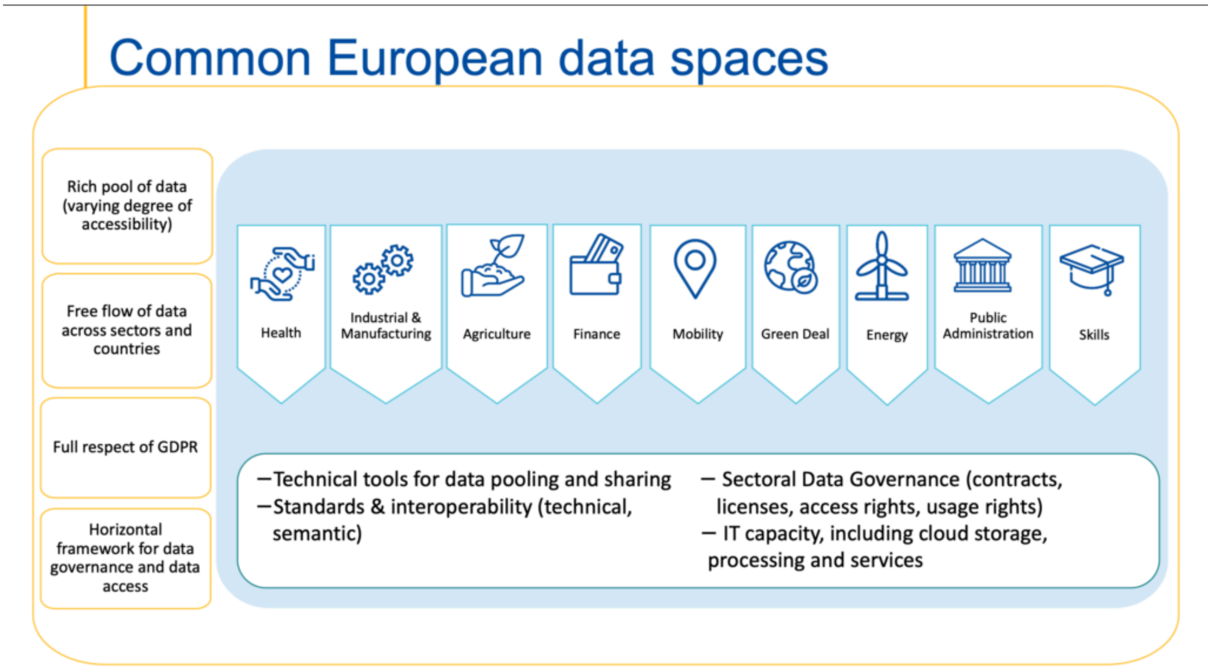
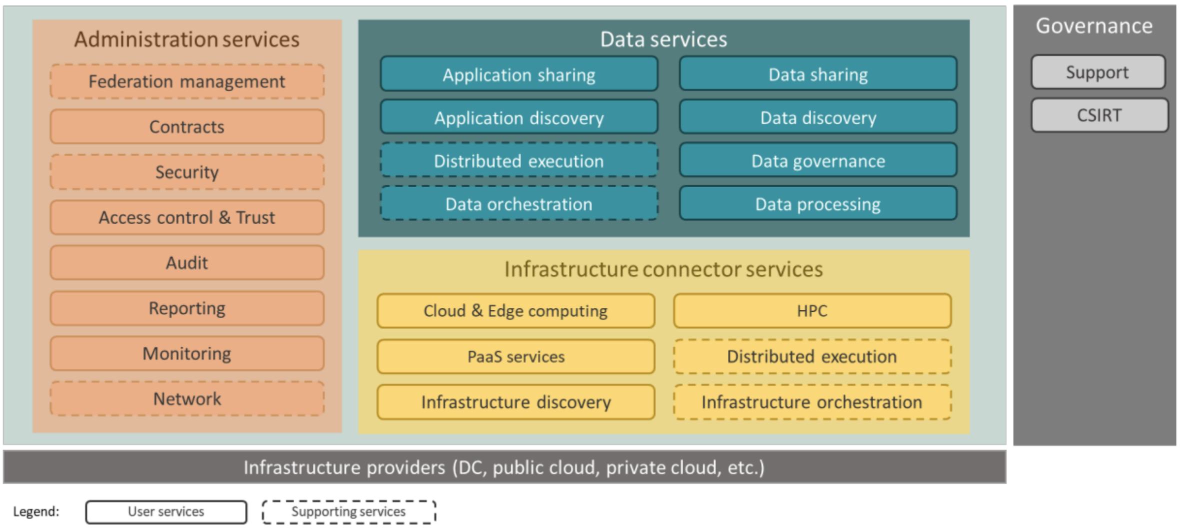

Common European data spaces
Rich pool of data (varying degree of accessibility)
Free flow of data across sectors and countries
Full respect of GDPR
Horizontal framework for data governance and data access
Health
Industrial & Manufacturing
Agriculture
Finance
Mobility
Green Deal
Energy
Public Administration
Skills
–Technical tools for data pooling and sharing
–Standards & interoperability (technical, semantic)
– Sectoral Data Governance (contracts, licenses, access rights, usage rights)
– IT capacity, including cloud storage, processing and services

Figure 22. Common European Data Spaces

As depicted in **Figure 23**, the EC has defined the procurement and the technical foundation for all those data spaces via the SIMPL framework[58].



[58] https://digital-strategy.ec.europa.eu/en/policies/simpl

Figure 23. High-level overview of SIMPL Open capabilities and architecture layers

SIMPL has been organised following this structure:

- **SIMPL-Open**: The open-source software stack, as envisaged in the preliminary study and shown in Figure 23, over which tenderers can elaborate their proposal.
- **SIMPL-Labs**: A pre-installed demonstration/playground environment where third parties (typically sectoral data spaces in their early stages of inception) can experiment with the deployment, maintenance, and support of the open-source software stack before deploying it for their own needs.
- **SIMPL-Live**: Several instances of the SIMPLE-Open software stack in the form of customised production environments for sectoral data spaces where the European Commission plays an active role in their management.

The high-level roadmap for the implementation of SIMPL is as follows:

- A Minimum Viable Platform will be released by the end of 2024.
- In parallel and starting as early as possible in 2025, the open testing environment (SIMPL-Labs) will be made available for stakeholders to experiment with.
- Progressively onboard and integrate use cases, helping them adjust SIMPL to their specific needs (without compromising its generic nature).
- The roadmap foresees significant new releases every six months.

interTwin integration with SIMPL will be evaluated as the first MVP implementation will be delivered, mainly to understand the type of data that could be made available for interTwin DTE and use cases.

5.6. TECH-01-01-2021 Projects

interTwin has been running activities of cross fertilisation and alignment of glossaries with projects funded under the same EC call, in particular BioDT[59] and DT-Geo[60]. We analysed in the following sections the architectures of the project to show the similarities and gaps with interTwin.

5.6.1. BioDT

The Biodiversity Digital Twin (BioDT) project aims to push the limits of our understanding of biodiversity dynamics by developing DT prototypes that provide advanced modelling, simulation and prediction capabilities. The developed DTs exploit the LUMI Supercomputer[61] and employ FAIR data combined with digital infrastructure, predictive modelling and AI solutions.

The BioDT Technical Platform architecture adopts a modular, layered approach with a strong focus on sustainability. It relies on existing services that are not solely dependent on the BioDT project itself, thereby ensuring long-term viability. The platform is designed to integrate HPC resources, cloud infrastructure, and specialised services that

[59] https://biodt.eu/

[60] https://dtgeo.eu/

[61] https://lumi-supercomputer.eu/

cater to the needs of biodiversity digital twins. The components of this architecture are depicted in **Figure 24**, and each plays a distinct role in supporting the platform's goals.

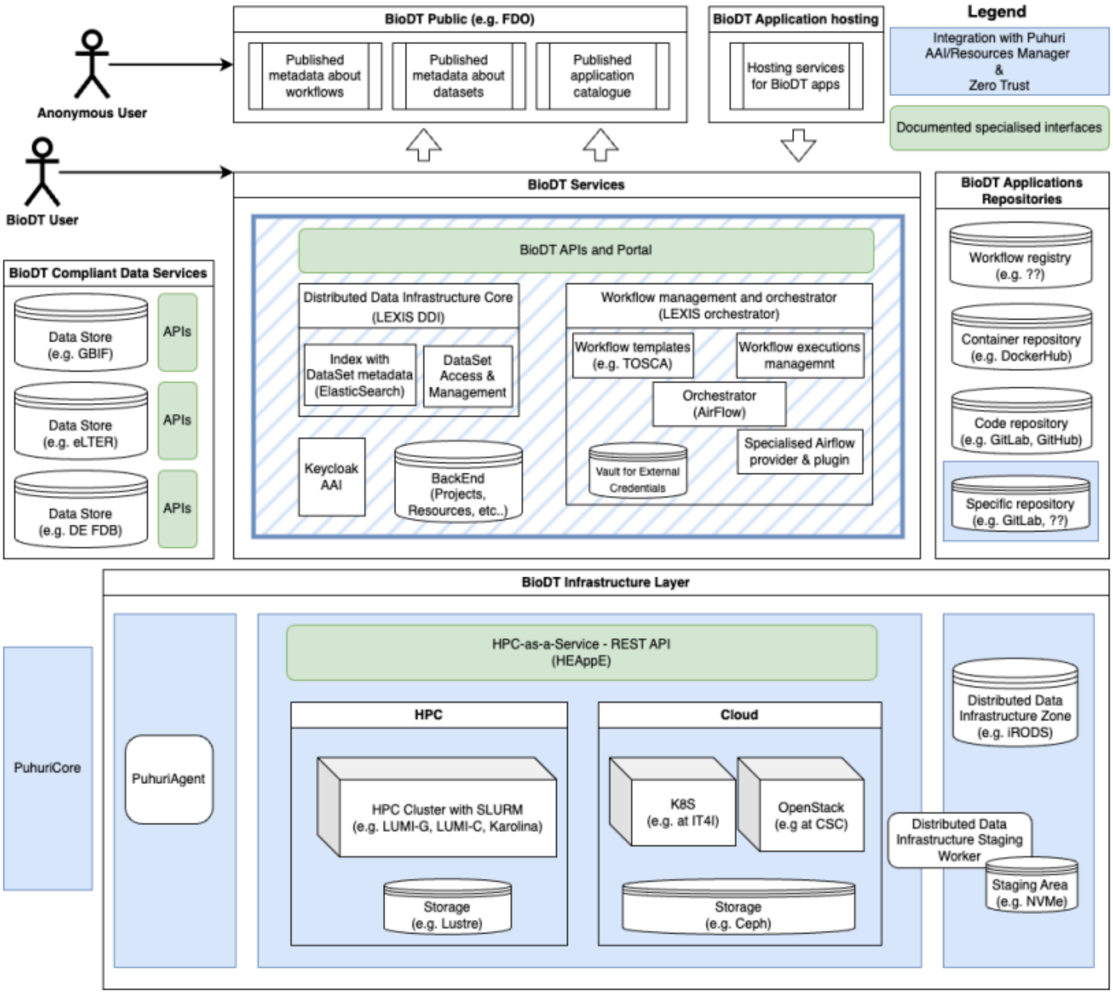

BioDT Public (e.g. FDO)
Published metadata about workflows
Published metadata about datasets
Published application catalogue
BioDT Application hosting
Hosting services for BioDT apps
Legend
Integration with Puhuri AAI/Resources Manager & Zero Trust
Documented specialised interfaces
Anonymous User
BioDT User
BioDT Services
BioDT APIs and Portal
Distributed Data Infrastructure Core (LEXIS DDI)
Index with DataSet metadata (ElasticSearch)
DataSet Access & Management
Workflow management and orchestrator (LEXIS orchestrator)
Workflow templates (e.g. TOSCA)
Workflow executions managemnt
Orchestrator (AirFlow)
Specialised Airflow provider & plugin
Vault for External Credentials
Keycloak AAI
BackEnd (Projects, Resources, etc..)
BioDT Compliant Data Services
Data Store (e.g. GBIF)
Data Store (e.g. eLTER)
Data Store (e.g. DE FDB)
APIs
APIs
APIs
BioDT Applications Repositories
Workflow registry (e.g. ??)
Container repository (e.g. DockerHub)
Code repository (e.g. GitLab, GitHub)
Specific repository (e.g. GitLab, ??)
BioDT Infrastructure Layer
PuhuriCore
PuhuriAgent
HPC-as-a-Service - REST API (HEAppE)
HPC
HPC Cluster with SLURM (e.g. LUMI-G, LUMI-C, Karolina)
Storage (e.g. Lustre)
Cloud
K8S (e.g. at IT4I)
OpenStack (e.g at CSC)
Storage (e.g. Ceph)
Distributed Data Infrastructure Zone (e.g. iRODS)
Distributed Data Infrastructure Staging Worker
Staging Area (e.g. NVMe)

Figure 24. BioDT Technical Architecture

Infrastructure Layer: At the core of the BioDT architecture is the infrastructure layer, which includes both HPC clusters and cloud computing resources. HPC clusters are equipped with batch schedulers like SLURM and are configured to handle specialised applications. These clusters are complemented by cloud systems that provide greater flexibility for tasks such as data pre/post-processing and visualisation services. BioDT utilises the LEXIS OpenStack cloud, which supports on-demand Kubernetes clusters. Both HPC and cloud systems are backed by high-performance storage solutions, such as Lustre and Ceph, ensuring rapid data access and transfer during computations.

BioDT uses an HPC-as-a-Service model, leveraging middleware solutions such as HEAppE[62]. This middleware abstracts the complexities of HPC access, providing job management, user authentication, file transfer, and other necessary functions through a REST API. The integration of HEAppE middleware simplifies the process for modellers and specialists, allowing them to run their models with minimal modification, while maintaining strict security policies on HPC usage.

[62]

Data Staging: One of the key challenges addressed by the BioDT architecture is the movement and management of large datasets. Data staging zones are implemented within the computer centres to facilitate efficient data transfer between tasks and workflows. These zones utilise iRODS[63] (Integrated Rule-Oriented Data System), which enables secure and fast data movement between different stages of computation. By caching frequently used datasets in the staging zone, the architecture optimises workflow performance while reducing network bandwidth usage.

BioDT Services: At a higher level, the platform provides a range of services and APIs that allow users to manage their data, execute computations, track workflow executions, and ensure reproducibility. The LEXIS Platform serves as the foundation for these services, integrating with Puhuri, a system that manages global resource allocation and authentication. Users are able to upload, download, and publish data through the platform, while the backend service ensures that all workflow executions and application parameters are tracked centrally.

The workflow management system in BioDT is designed to handle complex workflows that involve multiple tasks. These tasks are typically represented as Directed Acyclic Graphs (DAGs) and orchestrated using tools like Apache Airflow. For users, this orchestration process is simplified, with the technical details of task scheduling abstracted away, allowing them to focus on the high-level definitions of their workflows.

Security: Security is a critical aspect of the BioDT architecture. The platform adheres to a "zero trust" security model, where no user or service is inherently trusted. Access to resources and services is tightly controlled using the OAUTH2 protocol, and Keycloak is used to manage identity and access control. Additionally, Puhuri's authentication and resource allocation system is integrated with BioDT to ensure that users have the appropriate credentials to access global HPC resources. Sensitive data, such as those involved in specific use cases like disease modelling, are protected by additional security measures, including encryption and access vaults for storing credentials.

Distributed Data Infrastructure: As the BioDT platform is intended to operate across multiple data centres, it requires a robust Distributed Data UInfrastructure (DDI). The LEXIS DDI system, built on top of iRODS, provides a unified and reliable way to access data stored across different storage technologies. This system ensures that data remains accessible and secure, regardless of its location. Metadata about datasets is indexed using ElasticSearch, allowing for quick querying and retrieval of information.

Application Repositories and Hosting: To ensure the reproducibility and openness of BioDT workflows, the platform includes various repositories for storing workflows, containers, and code. WorkflowHub[64], DockerHub[65], and GitHub[66] are used to store workflows, containers, and code related to the BioDT use cases. For sensitive or confidential content, the platform supports private repositories, ensuring that the data remains secure while still being accessible to authorised users.

BioDT Public Interface: While much of the BioDT platform is reserved for registered users, certain services are available to the public. These include metadata about

[63] https://irods.org/

[64] https://workflowhub.eu/

[65] https://hub.docker.com/

[66] https://github.com/

workflows, datasets, and applications, which can be browsed without requiring a login. This public-facing aspect of the platform ensures that BioDT remains transparent and accessible to a broader community.

5.6.2. DT-GEO

The **DT-GEO** project aims to develop a prototype DT for geophysical extremes, providing users and researchers with a dedicated technological infrastructure for running workflows and simulations. The DT-GEO infrastructure has been extended using resources and services from **FENIX**, a federated infrastructure that offers cloud-based services, interactive computing, active data repositories, and data archival services. The DT-GEO FENIX infrastructure high level picture is shown in **Figure 25**.

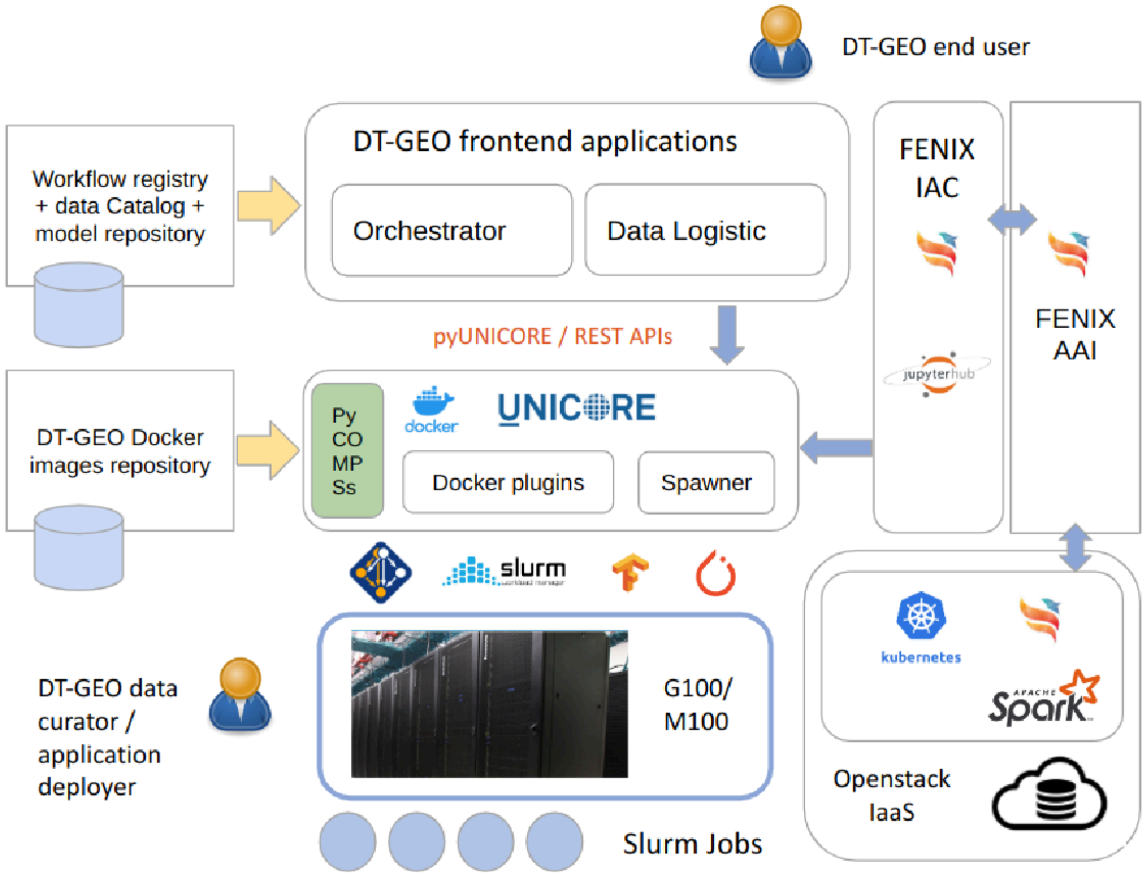


Figure 25. The DT-GEO FENIX infrastructure high level picture

The DT-GEO technical platform follows a modular Service-Oriented Architecture (SOA), designed to provide access to scalable computing and storage resources. This includes cloud-based services with support for containerization and AI applications. The infrastructure integrates services like the CINECA G100 HPC cluster and the OpenStack cloud, allowing users to create virtual machines (VMs) and run scientific applications. These components are critical for supporting the project's workflows and simulations.

Infrastructure Layer: At the core of the DT-GEO platform is the infrastructure layer, which includes HPC clusters like G100 and cloud resources from the FENIX federation. DT-GEO users can authenticate via SSH to access the G100 cluster and run containerized applications using environments like Docker or uDocker. In addition, the OpenStack cloud infrastructure allows users to create VMs and manage computational resources flexibly.

Interactive Computing Service (IAC): Allows users to submit jobs and run applications through a web interface based on JupyterHub, enabling the execution of scientific workflows without needing to directly manage the technical details of HPC resources.

Data Management: The DT-GEO infrastructure includes active data repositories located close to computational resources, allowing for fast and efficient access to data required for simulations. Users can also store and archive large datasets in the S3 object storage service provided by FENIX, ensuring the availability of data for future analysis or retrieval.

Workflow Orchestration: The workflow management system in DT-GEO is designed to handle complex workflows through tools like PyCOMPSs. These tools enable parallel application execution across multiple computing nodes, integrating cloud and HPC resources. Users interact with the platform via a web portal or CLI, simplifying scheduling and executing tasks.

Security: Security is a critical aspect of the DT-GEO infrastructure, managed through the FENIX AAI. This system ensures secure access to services using authentication protocols such as OpenID Connect and OAuth2. Additionally, users can seamlessly authenticate to FENIX resources across multiple sites within the federation using local identity providers (IdPs).

Public Access: While most of the DT-GEO platform is reserved for registered users, certain metadata about workflows, datasets, and applications are publicly accessible. This ensures transparency and promotes collaboration within the broader research community.

5.7. Summary of input to the blueprint architecture and DTE implementation

The previous sections have described and analysed some prominent initiatives that provide input to define the interTwin blueprint architecture and the actual DTE implementation. Most have already been identified at the proposal stage and included in the DoA. Some are not in a design or implementation phase that could, at this stage, be used as input for interTwin and, therefore, will be analysed in subsequent versions of this deliverable. The following table summarises the information collected.

Table 1 - Summary of input to the blueprint architecture and DTE implementation

initiatives / Projects	Relevance for interTwin	Actions
EOSC	Onboarding of services, EOSC Interoperability Framework, Reuse of technologies	Adhere to EOSC IF in some relevant areas (e.g. AAI) and contribute to the EOSC IF by implementing guidelines

		for DTE and integration with EOSC EU Node The Interoperability with the EOSC EU Node deployment technology is used also in interTwin (Infrastructure Manager by UPV)
EGI-ACE	Implement the EOSC Computing platform	Extend the DTE infrastructure with EGI Federation Computing platform providers.
ESCAPE	ESCAPE Data Lake Blueprint	Adoption of the ESCAPE data lake Blueprint and services in interTwin
C-SCALE	Access to Copernicus data federation, possible technology exchange (openEO, EO-MQS based on STAC)	Understand the data access and technology contributions from partners in the interTwin part of the C-SCALE project (EODC, LIP, DELTARES, etc.).
openEO platform	Implements data access and processing federation based on openEO API and standard process-graph definition	Reuse of technologies and expertise from openEO platform partners in the interTwin project (EODC, EURAC, WWU, TU Wien).
Digital Twin Consortium	Definitions and Digital Twin glossaries, Working group on Digital Twins for Research and Academia	Mapping of the concepts, such as the Digital Twin systems, into the next version of the Blueprint architecture
EU Data Spaces and SIMPL	Access to Sectoral Data space data via integration of the SIMPL framework	Analysis of the first version of the SIMPL MVP in 2025 (delayed due to late procurement process)
TECH-01-2021 Projects	“Sister projects” funded in the same call as interTwin, glossary alignment and technology exchange	Analysis of architectures and synergies to be put in place also thanks to DG-Connect driven initiative and Trilateral agreement.

6. Conclusions

This document outlines the DTE's architecture blueprint, and this section provides a summary of its design, components, and potential applications.

Blueprint Architecture Overview: The DTE's architecture is designed to be modular and scalable, allowing for flexibility in adapting to different scientific applications and needs. This modular approach facilitates updates and enhancements, helping the DTE stay aligned with ongoing technological developments.

Key Components and Functions: Each component of the DTE, including its data management system, real-time processing capabilities, and AI/ML integration, plays an important role. These components work together to create an efficient platform. The data management system focuses on ensuring accurate and reliable data handling, while AI/ML integration supports forecasting and informed decision-making.

Integration with External Initiatives: The DTE's design is compatible with global digital initiatives, such as DestinE, which extends its applicability across various scientific fields. This alignment supports the DTE's relevance in the evolving landscape of digital twin technologies.

Addressing Challenges and Future Enhancements: The DTE addresses a range of scientific use cases but also faces challenges related to evolving requirements. Ongoing adaptations will be needed to meet future demands.

The Road Ahead: The DTE is designed to evolve in step with technological advancements. Its flexible foundation supports continuous innovation, ensuring its long-term relevance in digital twin technology.

References

Reference	
No	**Description / Link**
R1	**VOMS, an Authorization System for Virtual Organizations.** R. Alfieri et al. In Grid Computing. AxGrids 2003. Lecture Notes in Computer Science, vol 2970. Springe DOI: 10.1007/978-3-540-24689-3_5
R2	**interTwin D4.2 First Architecture design of the DTs capabilities for High Energy Physics, Radio astronomy and Gravitational-wave Astrophysics** Andrea Manzi, Levente Farkas, Kalliopi Tsolaki, Sofia Vallecorsa, Sara Vallero, Massimiliano Razzano, Javad Komijani, Yurii Pidopryhora, & Isabel Campos. (2023). DOI: https://doi.org/10.5281/zenodo.10417138
R3	**interTwin D4.1 First Architecture design of the DTs capabilities for climate change and impact decision support tools** Muhammad Usman Liaqat, Mariapina Castelli, Donatello Elia, Gabriele Accarino, Davide Donno, Sandro Fiore, Bjorn Backeberg, Matthias Schramm, Christian Pagé, Frederique de Groen, Albrecht Weerts, Kathryn Roscoe, & Atef Ben Nasser. (2023/) DOI: https://doi.org/10.5281/zenodo.10417135
R4	**D5.1 First Architecture design and Implementation Plan (Final)**

	Diego Ciangottini, Paul Millar, Liam Atherton, Marica Antonacci, Daniele Spiga, Andrea Manzi, Renato Santana, David Kelsey, Adrian Coventry, & Shiraz Memon. (2023). DOI: https://doi.org/10.5281/zenodo.8036983
R5	**interTwin D6.1 Report on requirements and core modules definition (Final)** Isabel Campos, Donatello Elia, Germán Moltó, Ignacio Blanquer, Alexander Zoechbauer, Eric Wulff, Matteo Bunino, Andreas Lintermann, Rakesh Sarma, Pablo Orviz, Alexander Jacob, Sandro Fiore, Miguel Caballer, Bjorn Backeberg, Mariapina Castelli, Levente Farkas, & Andrea Manzi. (2023). DOI: https://doi.org/10.5281/zenodo.8036987
R6	**interTwin D7.1 Report on requirements and thematic modules definition for the environment domain (Final).** Michele Claus, Alexander Jacob, Björn Backeberg, Frederique de Groen, Joost Buitink, Roel de Goede, Donatello Elia, Gabriele Accarino, Sandro Fiore, Christian Pagé, Matthias Schramm, Bernhard Raml, & Christoph Reimer. (2023). DOI: https://doi.org/10.5281/zenodo.8036991
R7	**interTwin D7.2 Report on requirements and thematic modules definition for the physics domain first version (Final)** Kalliopi Tsolaki, Sofia Vallecorsa, David Rousseau, Isabel Campos, Yurii Pidopryhora, Sara Vallero, Alberto Gennai, & Massimiliano Razzano. (2023). DOI: https://doi.org/10.5281/zenodo.8036997
R8	**interTwin D3.4 DTE blueprint architecture, functional specifications and requirements analysis second version** Raul Bardaji, Andrea Manzi, Ivan Rodero, Thomas Geenen, Adam Warde. (2024). DOI: https://doi.org/10.5281/zenodo.10650440