

interTwin

D3.1 Blueprint architecture, functional specifications, and requirements analysis first version

Status: FINAL

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 provides an architectural blueprint for the interTwin Digital Twin Engine, outlining its functional specifications, requirements analysis, and fundamental building blocks.

The blueprint is designed to be utilised by the interTwin project's technical Work Packages (WPs) to align with the requirements of associated subsystems. Additionally, it considers other relevant initiatives and projects to identify potential architectural components that can be incorporated within the interTwin context. The blueprint will ultimately serve as a conceptual model of the Digital Twin Engine, following its planned evolution and iterations through collaborative co-creation during the project.



Document Description			
D3.1 Blueprint architecture, functional specifications, and requirements analysis first version			
Work Package number WP3			
Document type	Deliverable		
Document status	FINAL	Version	1
Dissemination Level	Public		
Copyright Status	 <p>This material by Parties of the interTwin Consortium is licensed under a Creative Commons Attribution 4.0 International License.</p>		
Lead Partner	EGI		
Document link	https://documents.egi.eu/document/3930		
DOI	https://doi.org/10.5281/zenodo.8094251		
Author(s)	<ul style="list-style-type: none"> • Raul Bardaji Benach (EGI) • Andrea Manzi (EGI) • Ivan Roderó (EGI) 		
Reviewers	<ul style="list-style-type: none"> • Paul Millar (DESY) • Alexander Jacob (EURAC) 		
Moderated by:	<ul style="list-style-type: none"> • Sjomara Specht (EGI) 		
Approved by	Germán Moltó on behalf of TCB		



Revision History			
Version	Date	Description	Contributors
V0.1	08/03/2023	ToC	Andrea Manzi (EGI)
V0.2	24/05/2023	First version shared with Reviewers	Raul Bardaji Benach (EGI) Andrea Manzi (EGI) Ivan Rodero (EGI)
V0.3	01/06/2023	Version ready for internal review	Raul Bardaji Benach (EGI) Andrea Manzi (EGI) Ivan Rodero (EGI)
V0.4	23/06/2023	Version reviewed by internal reviewers	Paul Millar (DESY) Alexander Jacob (EURAC)
v0.5	26/06/2023	Version reviewed by TCB	Germán Moltó (UPV)
v0.6	26/06/2023	Version ready for QA	Raul Bardaji Benach (EGI) Andrea Manzi (EGI) Ivan Rodero (EGI)
V1.0	29/06/2023	Final	

Terminology / Acronyms	
Term/Acronym	Definition
DT	Digital Twin
DTE	Digital Twin Engine
ML	Machine learning
DoW	Description of Work

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



Table of Contents

1	Introduction	9
1.1	Scope	9
1.2	Document Structure.....	9
1.3	Definitions and Glossary	9
2	Requirements analysis for the architecture	12
2.1	Requirements analysis process description	13
2.2	Use cases' requirements.....	15
2.2.1	Lattice QCD Simulations - High Energy Physics	16
2.2.2	Detector simulation - High Energy Physics	18
2.2.3	Noise simulation - Radio Astronomy.....	20
2.2.4	VIRGO Noise detector DT - GW Astrophysics	21
2.2.5	Climate Change Future Projections of Extreme Events - Natural Hazards	23
2.2.6	Early Warning for Extreme Events - Natural Hazards	24
2.2.7	Climate Change Impacts of Extreme Events - Natural Hazards	26
2.3	Requirements Category Summary	28
3	Relation to existing initiatives	30
3.1	Destination Earth (DestinE)	30
3.1.1	DestinE components	31
3.1.2	DestinE DTE	32
3.1.3	Data Lake.....	33
3.1.4	Core Service Platform.....	35
3.1.5	Linking activities with DestinE.....	35
3.1.6	Technology exchange from DG-Connect	35
3.2	EOSC and its compute platform	36
3.2.1	EOSC Core.....	37
3.2.2	EOSC Exchange.....	37
3.2.3	EOSC Interoperability Framework (EOSC-IF).....	37
3.2.4	EOSC Compute platform (EGI-ACE Project)	38
3.3	ESCAPE	41
3.3.1	DIOS Architecture.....	41
3.4	C-SCALE	42
3.4.1	Federated Earth System Simulation and Data Processing Platform (FedEarthData)	43
3.4.2	Earth Observation Metadata Query Service (EO-MQS)	44
3.4.3	openEO Platform.....	44
3.5	Digital Twin Consortium	45
3.6	Gaia-X.....	46
3.7	EU Data Spaces	48
3.8	Summary of input to the blueprint architecture and DTE implementation	50
4	Digital Twin Engine Blueprint Architecture.....	52
4.1	Methodology	52
4.2	Conceptual Model.....	53



4.3	DTE engine users.....	55
4.4	Architecture Model Specification.....	56
4.4.1	System context diagram.....	56
4.4.2	Container diagram.....	57
4.5	DT applications	59
4.6	DTE Thematic Modules.....	59
4.7	DTE Core Modules.....	61
4.8	DTE infrastructure.....	62
5	Conclusion.....	64
6	References	65



Table of Figures

Figure 1 Iterative interTwin DTE Blueprint Architecture	15
Figure 2 DestinE Organisational Responsibilities	32
Figure 3 DestinE DTE Architecture	33
Figure 4 DestinE Data Lake Highlight	34
Figure 5 EOSC High Level architecture.....	36
Figure 6 EOSC Compute Platform functional block diagram	39
Figure 7 DIOS overview	42
Figure 8 C-Scale Architecture.....	43
Figure 9 openEO platform federation.....	45
Figure 10 Digital Twins System	46
Figure 11 Gaia-X High Level conceptual architecture.....	47
Figure 12 Common European Data Spaces.....	48
Figure 13 High-level overview of SIMPL capabilities and architecture layers.....	49
Figure 14 interTwin Digital Twin Engine conceptual model	54
Figure 15 interTwin Digital Twin Engine System context diagram	57
Figure 16 Container diagram of the DTE	58

Table of Tables

Table 1 Summary of input to the blueprint architecture and DTE implementation	50
---	----



Executive summary

This document provides an architectural blueprint for the interTwin Digital Twin Engine, outlining its functional specifications, requirements analysis, and fundamental building blocks.

The blueprint is designed to be used by the interTwin project's technical Work Packages (WP5, WP6 and WP7) to align with the requirements of the related software components. Additionally, it considers other relevant initiatives and projects to identify potential architectural components that can be incorporated within the interTwin context and identify where interoperability is desirable. The blueprint will ultimately serve as a conceptual model of the Digital Twin Engine, following its planned evolution and iterations through collaborative co-creation during the project.

The initial iteration of the blueprint emphasises the foundational technical elements of the Digital Twin Engine, based on the needs of selected communities, engagement with other pertinent digital twin initiatives, and the aspiration to create a widely accepted Digital Twin conceptual model that caters to a multitude of diverse applications. Moreover, the document offers a timeline and roadmap for the implementation of the Digital Twin Engine.

1 Introduction

1.1 Scope

The primary scope of this deliverable is to establish an overview of the Digital Twin Engine (DTE) Architecture blueprint. This blueprint aims to fulfil the requirements arising from a broad array of scientific Digital Twin (DT) domains while simultaneously addressing the harmonisation of access to various heterogeneous computation and storage providers delivering the computing capacity.

The science use cases will not only define the requirements but also play an integral role in co-constructing the blueprint. Their active involvement will ensure that the requirements align with the software components designed and developed by the interTwin project's technical Work Packages (WPs). This process will centre around a symbiotic relationship between the science use cases and the software design, effectively grounding the design in real-world applications. Additionally, the co-design process will consider other relevant initiatives and projects to identify potential architectural components that can be integrated within the interTwin context, and to pinpoint areas where interoperability is desirable. The blueprint, deeply rooted in the science use cases, will ultimately serve as a conceptual model for the DTE, guiding its planned evolution and iterations through iterative co-creation during the project.

The blueprint will be continually refined and extended throughout the project, with a new version of the associated deliverable scheduled for release at M17 (D3.4), and a final version due at M26 (D3.5).

1.2 Document Structure

The document is organised as follows:

- **Chapter 1** comprises the introduction, including definitions and a glossary related to Digital Twins.
- **Chapter 2** details the process used to collect and analyse requirements from the DT applications within the project.
- **Chapter 3** investigates existing initiatives associated with Digital Twins and Distributed Infrastructure, assessing their potential contributions to the project.
- **Chapter 4** outlines the blueprint architecture, guided by the principles of the C4 model.
- **Chapter 5** serves as conclusion to the deliverable.

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 sources of data such as sensors or models considering historical as well as real time observations. A



D3.1 Blueprint architecture, functional specifications, and requirements analysis first version

Digital Twin is a digital replica that replicates 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.

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 re-train 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



D3.1 Blueprint architecture, functional specifications, and requirements analysis first version

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 pre-processing, 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 for the architecture

To build a DTE that is efficient, reliable, scalable, and compatible with other European initiatives, a thorough analysis of the architectural requirements must be undertaken before its development. This chapter delves into various requirements that need to be considered when creating an effective platform capable of constructing and exploiting accurate, efficient, and powerful DTs.

This chapter first presents the methodology and process for requirements collection and analysis. It then describes the high-level requirements for each use case and provides a summary. It further lays the groundwork and the necessary building blocks for defining the blueprint architecture. This architecture will be further refined and expanded in subsequent iterations, offering use cases an architecture that can adhere to standard design principles as detailed in **Chapter 4**.

The first two subsections delve into the process, examining the specific use cases defined in the project and the requirements they impose on the digital twin engine. The process, driven by the use cases, aims at delivering a technology-agnostic architecture whose key functionalities are divided into thematic, core, and infrastructure modules as defined at project proposal stage.

The **DTE Thematic Modules** of the DTE encompass domain-specific modelling techniques, data sources, and algorithms. These capabilities enable the DTE to tackle the unique challenges posed by each scientific discipline.

The **DTE core modules** must offer horizontal capabilities to facilitate the creation and operation of data-intensive and compute-intensive DT applications. These include:

1. Advanced workflow composition capable of invoking other core module capabilities and linking them with domain specific thematic modules for various use case applications.
2. Real-time data acquisition and analytics that ensure high-performance data ingestion based on serverless computing and on-the-fly processing during data acquisition.
3. Data fusion processes that harmonise, align or merge datasets from different sources.
4. AI workflow and method lifecycle management to execute complex AI setups, including model creation, training, validation, verification, and uncertainty tracing for model quality.

As outlined in the Description of Work (DoW) of the project, the **DTE infrastructure** modules should offer federated data and compute resources for modelling and simulation tasks, which includes:

1. Federated data management of current and future data infrastructures, paving the way for an interoperable cloud of data services (i.e., a data lake).

2. Federated compute infrastructure that provides diverse on-demand processing capacities and supports workflow execution.
3. Orchestration of resources and workflow components for the efficient execution of complex tasks.
4. Federation services and Authentication and Authorization Infrastructure (AAI) to control access and monitor the utilisation of resources and data across multiple providers.

Through meticulous requirement analysis and blueprint architecture definition, the DTE has been designed to cater to the needs of various disciplines, thus providing a foundation for the development of interoperable digital twins.

2.1 Requirements analysis process description

The primary goal of this process is to compile and analyse the requirements for the creation of the interTwin DTE Blueprint Architecture. This Blueprint Architecture outlines a reference architecture that could be used to implement DT solutions for any research collaboration. To achieve this objective, we employ a pragmatic requirements analysis approach anchored in a co-design process. This approach provides a conceptual framework (the DTE blueprint architecture as per [Chapter 4](#)) and facilitates the prototyping of an interdisciplinary DTE.

This first iteration of the requirements analysis process leans on a co-design approach with DT developers from select use cases: High energy physics, Radio astronomy, Gravitational wave astrophysics, Climate research, and Environmental monitoring. These use cases push the boundaries in modelling and simulation using heterogeneous distributed digital infrastructures, advanced workflow composition, real-time data management and processing, quality and uncertainty tracing of models, data fusion, and analytics. Consequently, the co-design process is steered by the principles of open standards adoption and interoperability towards a DT blueprint architecture, realising the ambition of a common approach to DT implementation.

The specific steps followed in this process, whose results are elaborated upon in the subsequent sections of this deliverable, include:

1. **Requirements elicitation interviews:** Through video conferences, we identified the needs and expectations of the use cases for the DTE. These meetings involved the technical team members of individual use cases and project technical coordination, focusing on gathering information about functionalities, features, constraints, and other attributes that the DTE must meet. This process was pivotal in establishing a shared understanding of the requirements among stakeholders.
2. **High-level requirements distilling:** This step involved summarising and synthesising the information collected from the interviews for each use case, pinpointing key requirements and features that needed addressing in the DTE.
3. **Requirements consolidation:** Here, the requirements were scrutinised, and common issues identified. This step allowed the integration of similar

D3.1 Blueprint architecture, functional specifications, and requirements analysis first version

requirements across use cases, driven by the project's technical coordination team.

4. **Feedback gathering:** Upon establishing the consolidated requirements, they were shared with stakeholders for review and validation. The interTwin Face-to-Face technical meeting in Madrid on January 25-26, 2023¹, was instrumental in gathering valuable feedback from stakeholders.
5. **Feedback analysis and consolidation:** In this final step, the feedback received was analysed and integrated into the initial version of the requirements, culminating in an improved set of requirements. These requirements were then considered for the blueprint architecture and further exploited in Work Packages 5, 6, and 7.

This initial requirement analysis, discussed in [section 2.2](#), revealed that at this stage of the project, there are some uncertainties that need to be addressed through a refinement process. This refinement is an inherent aspect of the DTE iterative co-design process, as it allows for continuous improvement and adaptation based on evolving project needs and emerging technological advancements. For example, during the development of the DTE, new data management techniques or collaboration tools may arise, necessitating the integration of these innovations into the existing architecture. By incorporating an ongoing refinement process, the DTE blueprint can stay agile and responsive to the dynamic nature of research collaborations and the ever-changing technological landscape.

The process depicted in [Figure 1](#) enables interTwin to evolve the blueprint architecture through co-creation within user communities. This collaboration directly contributes to the targeted key exploitable results (KER1, KER2, KER3, KER4, KER5 and KER6) as follows:

- **KER1:** Interdisciplinary Digital Twin Engine - A software platform providing both generic and tailored functional modules for modelling and simulation to facilitate the development and deployment of Digital Twins addressing scientific problems in diverse domains.
- **KER2:** Interoperability Framework - Guidelines, Specifications, and Blueprint Architecture. The interTwin interoperability framework aligns technical approaches and promotes collaboration in modelling and simulation application development across scientific domains.
- **KER3:** Toolkit for AI workflow and method lifecycle management - AI-based methodologies to extract application sector-specific information from exabyte-scale research data in real time, thereby enhancing the efficiency and accuracy of simulation and modelling outputs.
- **KER4:** Quality Framework - Tools for automated quality measures and trust, development of standard quality mapping and indicators for effectively communicating differences in qualities of inputs and outputs from digital twins, addressing issues such as data and model pedigree, accuracy, and lack of knowledge.

¹ <https://indico.egi.eu/event/6004/>



- **KER5:** DTE federated infrastructure integrated with EOSC and EU Data Spaces - Federated distributed compute platform providing access to distributed data and integrating HTC, HPC, Cloud and Quantum Computing capabilities for processing.
- **KER6:** interTwin Open Source Community - The community of DT application developers, users, and operators responsible for the design, development, and maintenance of the DTE codebase.

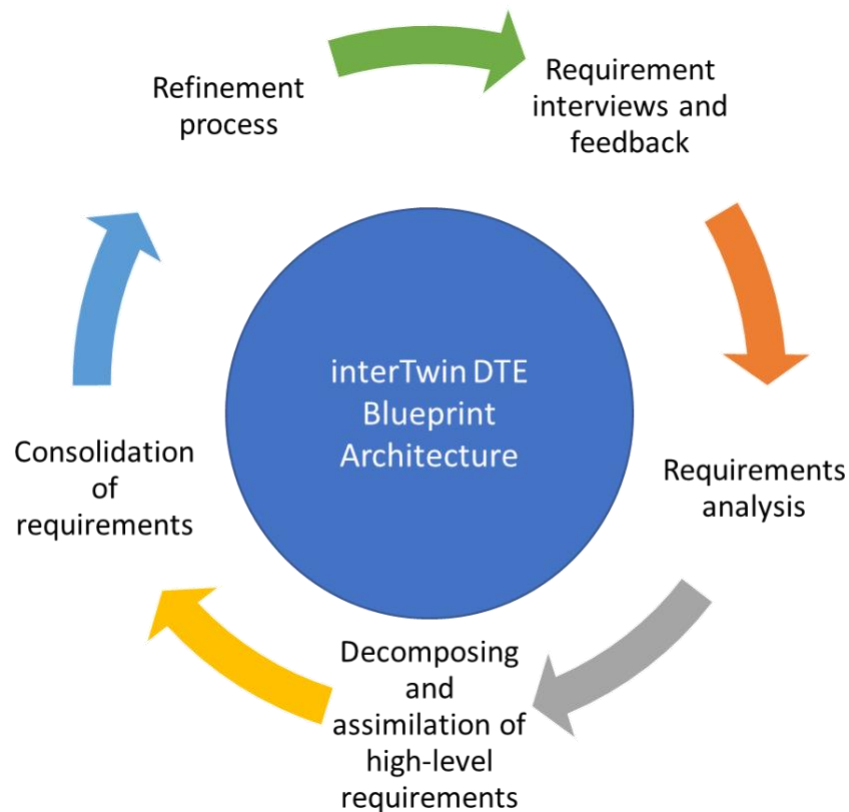


Figure 1 Iterative interTwin DTE Blueprint Architecture

By engaging with a diverse range of user communities and incorporating their invaluable input, interTwin can continuously refine and enhance the interoperability framework. This ensures its relevance and effectiveness in addressing the unique challenges faced by various scientific domains.

2.2 Use cases' requirements

The DTE is a multifaceted platform designed for deployment across various scientific disciplines, each presenting unique requirements. Therefore, understanding the specific needs and use cases for which the DTE will be utilised is essential. This section provides a high-level overview of the distinct requirements of the DT use cases, as derived from the interviews based on specific project-defined use cases. These include:

- High energy physics
 - Lattice QCD Simulations
 - Detector Simulation

D3.1 Blueprint architecture, functional specifications, and requirements analysis first version

- Radio astronomy
 - Noise Simulation
- Gravitational Wave (GW) Astrophysics
 - VIRGO Noise Detector DT
- Environmental monitoring
 - Climate Change Future Projections of Extreme Events such as storms and fire
 - Early Warning for Extreme Events such as floods and droughts
 - Climate Change Impacts of Extreme Events such as storms, fire, floods, and droughts

Examining the requirements of each use case provides insights into the specific capabilities and functionalities the DTE must embody to be effective in each scenario.

The requirements of each use case will be synthesised in the following subsections, all of which will adhere to the same format. Each subsection will first provide a brief summary of the use case's objective, followed by specific capabilities. Finally, key findings from interviews with the developers of each use case regarding their requirements will be summarised and categorised. This format provides a clear and organised presentation of the requirements for each use case, affording a comprehensive understanding of the necessary capabilities and functionalities the DTE must possess to be effective in each scenario.

The requirements for each use case will be sorted into the three main groups discussed previously (i.e., the thematic capabilities of the DTE, core capabilities of the DTE, and DTE infrastructure). In addition, the interview and requirement analysis process will help identify more detailed capabilities.

By categorising the requirements into these three groups, we can distinguish between the unique requirements of each use case and the core capabilities and infrastructure that bolster the overall functionality of the DTE. This approach facilitates the identification of areas for improvement, as well as potential synergies between different use cases, which could be leveraged to enhance the effectiveness of the DTE across all scientific disciplines.

2.2.1 Lattice QCD Simulations - High Energy Physics

Summary: The primary objective of the Lattice QCD Simulations Digital Twin use case is to create a digital representation of a system of quarks and gluons on a lattice, allowing scientists to simulate and predict their behaviour under extreme conditions. This Digital Twin could have significant applications in high energy physics research and potentially lead to new discoveries in the field.

Requirements elicitation: A summary of the interview process is provided below; however, more detailed information including technology requirements will be made available in D4.2 (M12)



D3.1 Blueprint architecture, functional specifications, and requirements analysis first version

Thematic Capabilities of the DTE:

- Accurate modelling of quantum chromodynamics for subatomic particles within a lattice.
- Support for investigating the behaviour of quarks and gluons under extreme conditions.

Core Capabilities of the DTE:

- Appropriate ML language and framework to support the desired ML models and tasks.
- Importance sampling and acceptance rate monitoring for ML models.
- ML validation through accept/reject methods.
- Hyperparameter optimization prior to training for small-sized problems.
- A workflow management system that involves minimal user intervention and a fixed number of iterations in the training process.
- Capability to handle various data formats, such as binary data, text data, and serialised data.

DTE Infrastructure:

- Input/output storage solutions that accommodate both local storage and HPC centres.
- No external databases required for training.
- Computing resources that utilise local and HPC centres for CPU, with support for parallel processing across multiple GPUs.
- Compatible OS and execution framework to support the project's requirements.

Requirement categories: The requirements' categories for this use case from the analysis of the requirements elicitation are summarised as follows:

Thematic Capabilities of the DTE:

1. The Digital Twin should accurately model the quantum chromodynamics of subatomic particles within a lattice.
2. The system should support the investigation of the behaviour of quarks and gluons under extreme conditions.

Core Capabilities of the DTE:

3. ML Language: Python.
4. ML Framework: PyTorch.
5. ML Models: Normalising flows used for importance sampling.
6. ML monitoring: Acceptance rate monitoring for normalising flows.
7. ML validation: Performed through accept/reject methods.
8. No ML retraining is required, but hyperparameters will be optimised prior to training for small-sized problems.



D3.1 Blueprint architecture, functional specifications, and requirements analysis first version

9. Workflow Tools: OS command line, Jupyter notebooks for monitoring only, separate batch scripts for training and measurements, no user intervention, and fixed number of iterations in the training process.
10. Data Formats: Binary data, text data, and pickle.

DTE Infrastructure:

11. Storage input/output: Local storage and HPC centres for both input and output.
12. Computing: Utilises local and HPC centres for CPU, with multi-GPU support for the entire computation, including training, configuration generation, and measurements.
13. OS and Execution Framework: Linux.

2.2.2 Detector simulation - High Energy Physics

Summary: The primary objective of the Detector Simulation LHC Digital Twin use case is to simulate the Large Hadron Collider's (LHC) detectors, addressing the complex multi-dimensional problem of detector simulation in high energy physics. The increasing sophistication and complexity of the detectors, the heightened precision required by cutting-edge experiments, the surge in data volume that needs to be processed, and the implementation of ever more advanced physics models expect a significant increase in annual CPU consumption in the coming years. The need for fast simulation methods comes from two facts: simulation takes a substantial part of computing resources, and calorimeters are the sub-detectors that usually are the most time-consuming.

Requirements elicitation: A summary of the interview process is provided below; however, more detailed information including technology requirements will be available in D4.2 (M12)

Thematic Capabilities of the DTE:

- Accurate simulation of the LHC detectors using generated input data.
- Transformation of output files into common formats compatible with other analysis frameworks.
- Generalisation of training data to reduce file sizes.

Core Capabilities of the DTE:

- Appropriate ML language, framework, and models to support the desired training process and model optimization.
 - Workflow engine for executing training and model operations.
 - Continuous monitoring and adjustment of the training process as needed.
 - Custom metric definition for model optimization.

DTE Infrastructure:

- Input/output storage solutions that accommodate local storage and object storage.
 - No external databases required for training.



D3.1 Blueprint architecture, functional specifications, and requirements analysis first version

- Computing resources that support HPC with MPI infrastructure for CPU and GPU.
- Compatible OS, containerization environment, and execution framework to support the project's requirements.
- Real-time data acquisition and processing capabilities, with offline post-processing for larger datasets when needed.
- Workflow tools for executing training steps, pre-processing files, and monitoring.
 - Support needed for ROOT and HDF5 data formats.

Requirement categories: The requirements' categories for this use case from the analysis of the requirements elicitation are summarised as follows:

Thematic Capabilities of the DTE:

1. Simulate the LHC detectors using generated input data from Monte Carlo simulations run through the GEANT4 toolkit.
2. Transform the specific format output files to other common formats to be used with other analysis frameworks.
3. Generalise the training data to abstract the structure of the input data, reducing file size significantly.
4. Optimise generative models using loss functions to produce simulated data closely resembling the Monte Carlo output.

Core Capabilities of the DTE:

5. ML Language: Python.
6. ML Framework: TensorFlow.
7. ML Models: GAN, Transformer, Flow, or Energy-based models for training.
8. Use of a workflow engine to execute the steps of the training and model operations.
9. Continuous monitoring of the training process, adjusting training parameters or stopping the training as needed.
10. Users can define custom metrics to optimise the model.

DTE Infrastructure:

11. Storage input/output: A persistent storage system with high-availability, durability to withstand potential data losses, and performance capabilities sufficient to handle substantial data volumes is required. Furthermore, a storage solution capable of providing durable, scalable storage for output data and also capable of accommodating input data when necessary is needed.
12. Computing: HPC with MPI infrastructure for CPU and GPU support.
13. OS and Execution Framework: Linux, containerization environment, and Jupyter notebooks.
14. Real-time Data Acquisition and Processing: Currently online processing, but post-processing might be done offline for larger datasets.



D3.1 Blueprint architecture, functional specifications, and requirements analysis first version

15. Workflow Tools: Use of workflow tools to execute all steps of training and running the model, offline pre-processing files (before feeding them into the ML model), and Jupyter notebooks for monitoring.
16. Data Formats: ROOT and HDF5.

2.2.3 Noise simulation - Radio Astronomy

Summary: The primary objective of the Noise Simulation for Radio Astronomy Digital Twin use case is to develop an ML-based system to analyse and filter data collected from radio telescopes, such as the Effelsberg and MeerKAT telescopes in Germany and South Africa, respectively, and future telescopes like the Square Kilometre Array (SKA) in South Africa and Australia.

Requirements elicitation: A summary of the interview process is provided below; however, more detailed information including technology requirements will be available in D4.2 (M12)

Thematic Capabilities of the DTE:

- Accurate noise simulations for radio astronomy instruments.
- Support for the classification of radio signal sources.
- Adaptability for use with current and future radio telescopes.

Core Capabilities of the DTE:

- Appropriate ML language and framework to support the desired ML models and tasks.
- Efficient ML training and deployment strategies.
- Workflow management tools to facilitate distributed training and multithreaded computation.

DTE Infrastructure:

- Storage solutions suited for small (<10 Mb) and medium-sized (< Gb) files.
- Database or online service for model history retrieval.
- Computing resources that support local CPU and GPU processing.
- Compatible OS and execution framework to support the project's requirements.

Requirement categories: The requirements' categories for this use case from the analysis of the requirements elicitation are summarised as follows:

Thematic Capabilities of the DTE:

1. The Digital Twin should accurately simulate noise for radio astronomy instruments.
2. The system should support the classification of radio signal sources.
3. The system should be adaptable for use with current and future radio telescopes.

Core Capabilities of the DTE:



D3.1 Blueprint architecture, functional specifications, and requirements analysis first version

4. ML Language: Python (SciPy, NumPy, and others).
5. ML Framework: TensorFlow, with additional modules implemented in C++.
6. ML Training: Distributed training on an HPC cluster using multiple GPUs and partial training to minimise data involvement.
7. ML Output: A text file with a list of time chunks and attributed classifications.
8. ML Deployment: Future implementation on FPGA firmware at the instrument side.
9. Workflow Tools: Distributed training on an HPC cluster, multithreaded computation, and Jupyter notebooks for monitoring and analysis.

DTE Infrastructure:

10. Storage: File-based storage suited for small and medium-sized files (GB).
11. Databases: A database or online service for retrieval of model history for offline analysis.
12. Computing: Local computing resources for CPU, with GPU support.
13. OS and Execution Framework: Linux, Singularity for containerization, and Jupyter notebooks.

2.2.4 VIRGO Noise detector DT - GW Astrophysics

Summary: The primary objective of the VIRGO Noise Detector Digital Twin use case is to enhance the precision of gravitational wave interferometers by using Generative Adversarial Networks (GANs) to develop a Digital Twin of the Virgo interferometer. The DT will simulate transient noise in the detector, allowing for the characterization of the noise and optimization of auxiliary channels to minimise false multi-messenger signals and denoise the signal in low-latency searches.

Requirements elicitation: A summary of the interview process is provided below; however, more detailed information including technology requirements will be available in D4.2 (M12).

Thematic Capabilities of the DTE:

- Accurate simulation of transient noise in the Virgo interferometer.
- Optimization of auxiliary channels for vetoing and denoising the signal in low-latency searches.

Core Capabilities of the DTE:

- Appropriate ML language and framework to support the desired ML models and tasks.
- Real-time processing of multiple parallel input streams, with pre-processing and model selection based on input data analysis.
- Buffering and retraining capabilities, with data discarded after retraining.
- Quality verification and monitoring for input stream processing and (re)training process.



D3.1 Blueprint architecture, functional specifications, and requirements analysis first version

- Workflow management tools to facilitate distributed training and multithreaded computation.
- Real-time data acquisition and processing capabilities.

DTE Infrastructure:

- Storage solutions suited for small and medium-sized files.
- Database or online service for model history retrieval.
- Computing resources that support local CPU and GPU processing.
- Compatible OS and execution framework to support the project's requirements.

Requirement categories: The requirements' categories for this use case from the analysis of the requirements elicitation are summarised as follows:

Thematic Capabilities of the DTE:

1. The Digital Twin should accurately simulate transient noise in the Virgo interferometer.
2. The system should optimise the use of auxiliary channels for vetoing and denoising the signal in low-latency searches.

Core Capabilities of the DTE:

3. ML Language: Python (SciPy, NumPy, and others).
4. ML Framework: TensorFlow, with additional modules implemented in C++.
5. ML Training: Real-time processing of multiple parallel input streams (up to 10), with pre-processing and model selection based on input data analysis.
6. ML Retraining: Buffering data for retraining based on volume, input data, or external triggers, and discarding buffered data after retraining.
7. ML Output: "Cleaned" data with vetoed or denoised noise, feeding into other pipelines in the project.
8. Quality Verification: Monitoring input stream processing and (re)training process, with sampling and a framework like TensorBoard for notifications and visualisation.
9. Workflow Tools: Distributed training on an HPC cluster, multithreaded computation, and Jupyter notebooks for data analysis and pre-processing.
10. Real-time Data Acquisition and Processing: Required for this project.

DTE Infrastructure:

11. Storage: File-based storage suited for small and medium-sized files (GB).
12. Databases: A database or online service for retrieval of model history for offline analysis.
13. Computing: Local computing resources for CPU, with GPU support.
14. OS and Execution Framework: Linux, Singularity for containerization, and Jupyter notebooks.



2.2.5 Climate Change Future Projections of Extreme Events - Natural Hazards

Summary: The Climate Change Future Projections of Extreme Events DT use case aims to develop a comprehensive solution for analysing and predicting extreme weather events such as storms and fires. This will be achieved by creating a DT that integrates climate data from various sources such as COPERNICUS², ESGF³, and IBTrACS⁴. The project will include a data processing suite for data manipulation, ensuring machine learning (ML) compliance, and delivering ML models for extreme event use cases (storms and fires) within a framework that employs Iterative Model Updating.

Requirements elicitation: A summary of the interview process is provided below; however, more detailed information including technology requirements will be available in D4.1 (M12).

Thematic Capabilities of the DTE:

- Analysis and accurate prediction of extreme weather events, such as storms and fires.
- Integration of climate data from various sources.

Core Capabilities of the DTE:

- Appropriate ML language and framework to support the desired ML models and tasks.
- Support for various Machine Learning (ML) models, including Convolutional Neural Networks (CNNs), which are primarily used for image and video processing tasks; Generative Adversarial Networks (GANs), a class of AI algorithms designed for unsupervised learning; and Graph Neural Networks (GNNs), also known as Convolutional Gaussian Networks (CGNNs), which are designed to process data structured as graphs.
- Use of multiple GPUs for training and parallel processing of data.
- Data pre-processing and visualisation.
- Robust workflow management and automation.

DTE Infrastructure:

- Storage solutions suited for small (<10 Mb) and medium-sized (< Gb) files.
- Computing resources that support HPC/HTC environments and HPC resource managers.
- Compatible OS and execution framework to support the project's requirements.
- Real-time data acquisition and processing capabilities.

² <https://www.copernicus.eu/en/access-datas>

³ <https://aims2.llnl.gov/search>

⁴ <https://www.ncei.noaa.gov/products/international-best-track-archive>



D3.1 Blueprint architecture, functional specifications, and requirements analysis first version

- Workflow management tools, such as Jupyter notebooks, CI/CD, and log management systems.
- The system can handle different common data formats, including NetCDF and CSV.

Requirement categories: The requirements' categories for this use case from the analysis of the requirements elicitation are summarised as follows:

Thematic Capabilities of the DTE:

1. The DT should accurately analyse and predict extreme weather events such as storms and fires.
2. The system should integrate climate data from various sources, such as COPERNICUS, ESGF, and IBTrACS.

Core Capabilities of the DTE:

3. ML Language: Python (Xarray, PyCDO, NumPy, Pandas, SciKit Learn).
4. ML Framework: TensorFlow, Keras.
5. ML Models: Support for different ML/DL models (CNNs, GANs, CGNN/GNNs).
6. ML Training: Utilise multiple GPUs for training CNN-based models for storm prediction and GAN-based models for fire prediction.
7. Data Pre-processing: Required before training.
8. Visualisation: Graphical environment to display maps with cyclone centres and grid points with fire risk.
9. Parallel Processing: Capability to parallelize processes using multiple CPUs/GPUs.
10. Workflow Management: Robust workflow engine for managing stages and Cron jobs for task automation.

DTE Infrastructure:

11. Storage: File-based storage suited for small and medium-sized files (GB), approximately 300GB for storm data, 1TB for fire data, and 0.5-1TB for models.
12. Computing: HPC/HTC with HPC resource managers (LSF, Slurm) for model training, and exploration of cloud resources.
13. OS and Execution Framework: Windows, Linux, Docker or Singularity for containerization, and user visualisations with Python (artopy, matplotlib, bokeh).
14. Real-time Data Acquisition and Processing: Handled by openEO.
15. Workflow Tools: Jupyter notebooks, CI/CD, and log management systems.
16. Data Formats: NetCDF, CSV, and others.

2.2.6 Early Warning for Extreme Events - Natural Hazards

Summary: The primary objective of the Early Warning for Extreme Events Digital Twin project is to develop a comprehensive simulation and early warning system for high



D3.1 Blueprint architecture, functional specifications, and requirements analysis first version

impact hydrometeorological events, such as floods and droughts. By leveraging multidisciplinary observations and models, the project will improve the quality and accuracy of early warnings, enhancing the realism of impact warnings and response effectiveness.

Requirements elicitation: A summary of the interview process is provided below; however, more detailed information including technology requirements will be available in D4.1 (M12).

Thematic Capabilities of the DTE:

- Simulations that accurately predict high impact hydrometeorological events across all earth system components.
- Improved early warning quality and accuracy for extreme events like floods and droughts.

Core Capabilities of the DTE:

- Appropriate ML language and framework to support the desired ML models and tasks.
- Processing of input data from diverse sources and data formats.
- Time scale output information for early warning systems.
- Workflow management tools to facilitate development, documentation, monitoring, and event-based processing.
- Real-time data acquisition and processing capabilities.

DTE Infrastructure:

- Storage solutions suitable for objects and files. Database management such as an API for indexing and storing data as needed.
- Computing resources that support local, HPC, and cloud processing for both CPU and GPU tasks.
- Compatible OS and execution framework to support the project's requirements.

Requirement categories: The requirements' categories for this use case from the analysis of the requirements elicitation are summarised as follows:

Thematic Capabilities of the DTE:

1. The Digital Twin should accurately simulate high impact hydrometeorological events across all earth system components.
2. The system should provide improved early warning quality and accuracy for extreme events like floods and droughts.
3. The system should be able to link existing hydrological models with data driven ml models into hybrid modelling digital twins.

Core Capabilities of the DTE:

4. ML Language: Python (PyTorch) and tensorflow.
5. ML Models: Real-time data acquisition and processing.



D3.1 Blueprint architecture, functional specifications, and requirements analysis first version

6. ML Training: Processing of input data from diverse sources, such as Copernicus Climate Data Store, IBTrACS, and CMIP6 Models in NetCDF file format.
7. ML Output: Timely and accurate information for early warning systems.
8. Workflow Tools: openEO process graph-based workflows, Jupyter notebooks for development and documentation, event-triggered (user) monitoring processing, and event-based platform.
9. Real-time Data Acquisition and Processing: Data acquisition and processing managed through openEO API.
10. Data Formats: Support for various raster and vector data formats, including binary, text, and geospatial data formats such as geojson, geopackage, shape files, jpeg2000, geotiff, NetCDF, HDF (4&5) and Grib.

DTE Infrastructure:

11. Storage: Local file-based storage and object storage (S3) for input data, along with a combination of cloud storage, object storage (S3), and local file-based storage for output data.
12. Databases: Data indexed into STAC based catalogues or ingested into datacube engines such as open data cube or propriety database such as rasdaman or the FEWS database, with the possibility of storage in a linked archive.
13. Computing: Local, HPC, and cloud resources for both CPU and GPU processing.
14. OS and Execution Framework: Windows and Linux operating systems, with Docker or Singularity for containerization. User visualisations created using Python libraries such as matplotlib and cartopy.

2.2.7 Climate Change Impacts of Extreme Events - Natural Hazards

Summary: The primary objective of the Climate Change Impacts of Extreme Events Digital Twin project is to evaluate changes in the characteristics of climate extreme events and their impact using machine learning-based methodologies. The study aims to enhance our comprehension of the potential impacts of climate extremes in the future and offer support for future climate change mitigation and adaptation policies.

Requirements elicitation: A summary of the interview process is provided below; however, more detailed information including technology requirements will be available in D4.1 (M12).

Thematic Capabilities of the DTE:

- The Digital Twin should accurately evaluate the spatial extent, frequency of occurrence, time duration, and intensity of extreme events using global climate simulations.
- The system should assess the uncertainties using multiple greenhouse gas scenarios and simulation ensembles, employing climate indices to identify climate change impacts.

Core Capabilities of the DTE:



D3.1 Blueprint architecture, functional specifications, and requirements analysis first version

- ML Models: Different models may need to be trained for each domain (storms, wildfires, floods, droughts).
- ML Training: Train an ML model using GPUs and all grids for the area of interest, using all available history and computed indexes from input data.
- ML Retraining: Retrain the model every 5-10 years if necessary.
- Workflow Tools: Pre-processing should be done once, with the resulting data stored and reused.
- Data Formats: NetCDF files, REST APIs for filtering and searching data.

DTE Infrastructure:

- Storage: Storage space for climate simulations, a few hundred GB per simulation. One file per month per variable, with a file size of 25 MB per month per variable.
- Computing: GPU resources for training ML models.
- OS and Execution Framework: Not specified, but Linux is commonly used in the climate science domain. Visualisation services required.
- Real-time Data Acquisition and Processing: Data can be grid data from the Copernicus Climate Data Store using REST APIs, historical simulations of climate models from ESGF climate data infrastructure, or future projection data from ESGF climate data infrastructure.

Requirement categories: The requirements categories for this use case from the analysis of the requirements elicitation are summarised as follows:

Thematic Capabilities of the DTE:

1. Accurate evaluation of the characteristics of climate extreme events and their impact.
2. Assessment of uncertainties using multiple greenhouse gas scenarios and simulation ensembles.

Core Capabilities of the DTE:

3. Appropriate ML language and framework to support the desired ML models and tasks.
4. Training and retraining capabilities for different models based on each domain.
5. Workflow management tools to facilitate pre-processing, data storage, and reuse.
6. Data acquisition and processing capabilities that support multiple data formats.

DTE Infrastructure:

7. Storage solutions suitable for various types and sizes of climate simulations.
8. Database or online service for retrieval of model history for offline analysis, if needed.
9. Computing resources that support GPU processing for ML model training.
10. Compatible OS and execution framework to support the project's requirements, with visualisation services.



2.3 Requirements Category Summary

This section summarises the requirement categories identified in the first iteration of the process described in **Figure 1** for the definition of the interTwin DTE Blueprint Architecture as seen in **Chapter 4**. However, there are some uncertainties and requirement evolution that will be addressed in the next iteration.

Thematic Capabilities of the DTE: These requirements pertain to the specific domain knowledge and functionality required for each use case and are developed in deliverable D7.1 **[R1]** and D7.2. **[R2]**

Core Capabilities of the DTE: These requirements are common across all use cases and encompass the fundamental functionalities and features that the DTE must provide, regardless of the specific scientific domain. These capabilities include:

- **ML Language:** Specifies the programming language used for ML models in the use case.
- **ML framework:** Specifies the ML framework used for the use case.
- **ML other languages:** Specifies any secondary languages that are essential for the complete functionality of the ML models, even though they are not the main language used in the process.
- **ML models:** Specifies the need for ML models used for the use case.
- **ML monitoring:** Specifies the need for monitoring and logging of the ML models.
- **ML retraining:** Refers to the need of updating an existing ML model with new data. This process is often necessary because the performance of ML models can degrade over time. This degradation can be due to changes in the environment in which the model operates, a phenomenon referred to as concept drift.
- **ML event-based retraining:** Determines the necessity for retraining a Machine Learning model based on specific events or triggers, as opposed to adhering to a fixed schedule.
- **ML validation:** Specifies the need for validation of the ML models.
- **Real-time data acquisition and processing:** Refers to the need of continuous collection and analysis of data in real time, as opposed to batch processing, where data is collected over a certain period and then processed all at once. Parameters associated with online data processing, such as latency, have not yet been definitively established in this document. This is mainly due to the project still being in its early stages, and for many use cases, these parameters remain undefined. The examination and definition of these online data processing parameters, including latency, is an essential aspect of the project development and will be comprehensively addressed in future versions of the blueprint architecture.
- **Workflow tools:** Specifies the need of technological applications designed to help developers automate sequential tasks and streamline the processes involved in the execution of a DT.



D3.1 Blueprint architecture, functional specifications, and requirements analysis first version

- **Data formats:** Specifies that particular data formats and data structures are used in the use case.

DTE Infrastructure: These requirements pertain to the underlying infrastructure that supports the DTE and enables it to function effectively across all use cases. These requirements include:

- **Storage input:** Specifies the need for a storage medium used for the training data of the DT for an input use case, such as a file system.
- **Storage output:** Identifies the requirement for a storage medium, such as object storage, to store the Digital Twin (DT) and its associated output files from the use case.
- **Databases:** Specifies the need for database services, including relational databases, for the use case.
- **Computing GPU:** Specifies the need for GPU capabilities.
- **OS and execution framework:** Specifies the operating system and execution framework used for the use case.



3 Relation to existing initiatives

In the development of the interTwin project, a systematic analysis of the project requirements has been conducted. This is augmented by a review of related projects and initiatives that connect to the objectives that interTwin is designed to fulfil. Recognising synergies and potential reuse of concepts or technologies is an integral part of this process.

This chapter will delve into various projects that are engaged in creating digital twins, and those leveraging IT infrastructures to build distributed hybrid computing frameworks along with data analysis products.

Given that this is the initial iteration of the deliverable about the Blueprint Architecture and considering the initial stage of several projects and initiatives described in this chapter, the content presented here will be expanded in subsequent iterations of the deliverable as our engagement with these initiatives strengthens.

The purpose of this exploration is to leverage existing knowledge, learn from the challenges faced by these initiatives, and pinpoint opportunities for collaboration or integration. Understanding the broader landscape of similar projects and initiatives is crucial for strategically placing interTwin within this larger ecosystem, circumventing redundancy, and delivering a DTE that embodies effectiveness, efficiency, and innovation.

3.1 Destination Earth (DestinE)

Destination Earth, often referred to as DestinE, is an initiative committed to establishing a high-fidelity digital replica of our planet. This ambitious undertaking strives to deepen our comprehension of climate change impacts and environmental catastrophes, thereby equipping policymakers with robust tools to devise more effective responses.

DestinE aligns with the dual priorities of the European Commission - the digital transition and the climate transition - with the overarching goal of achieving carbon neutrality by 2050. As a 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, that incorporates real-time data from a broad array of sources, including climate monitors, meteorological sensors, atmospheric probes, and behavioural indicators.

This digital model of Earth is designed to serve a multitude of user groups, ranging from the public sector to scientific communities and private enterprises. By allowing the monitoring and simulation of natural and anthropogenic activities, the system paves the way for the development of various test scenarios aimed at predicting the consequences of climate change more accurately.

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

3.1.1 DestinE components

The overall goal of our project is to align its architecture with the one envisioned by DestinE. The DestinE architecture's construction is delegated to three distinct organisations, each entrusted with a different part of the architecture, as depicted in **Figure 2**.

- DestinE **Digital Twin Engine** (DestinE DTE)⁵: This component is under the management of the European Centre for Medium-Range Weather Forecasts (ECMWF).
- DestinE **Data Lake** (DEDL)⁶: This is managed by the European Organisation for the Exploitation of Meteorological Satellites (EUMETSAT). The DEDL serves as the repository for vast amounts of data required for the functioning of the DestinE system. It handles the collection, storage, and distribution of meteorological and other relevant data.
- DestinE **Core Service Platform** (DESP)⁷: The European Space Agency (ESA) manages this component. The DESP delivers the necessary resources and services to enable the efficient operation of the DestinE DTE and DEDL. It forms the backbone of the DestinE architecture by integrating various services, including data processing, visualisation, and analysis tools.

⁵ <https://digital-twin-engine.readthedocs.io/en/latest/>

⁶ <https://www.eumetsat.int/who-we-work/destine>

⁷ https://www.esa.int/Applications/Observing_the_Earth/Journey_to_Destination_Earth_begins



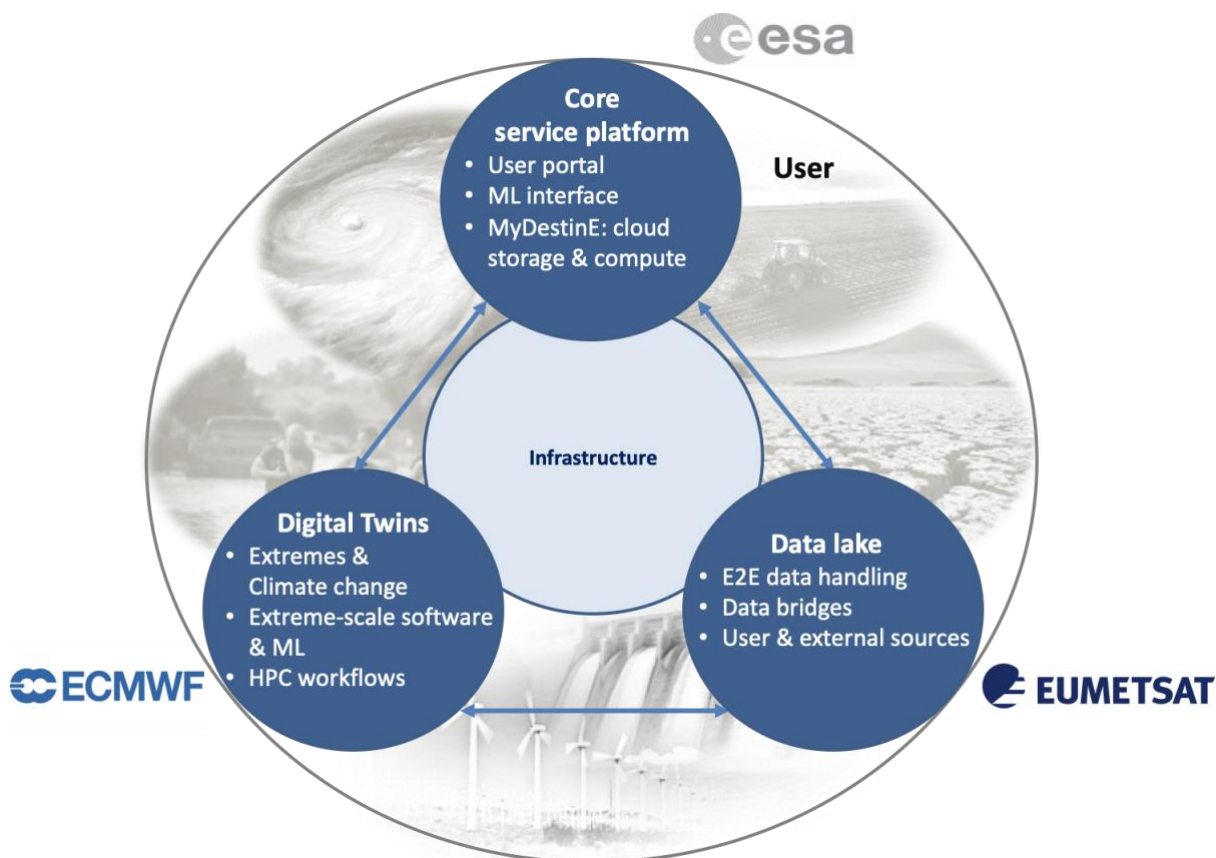


Figure 2 DestinE Organisational Responsibilities

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

3.1.2 DestinE DTE

The DestinE Digital Twin Engine (DTE) forms an important component of the DestinE platform, comprising the software infrastructure necessary 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 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 – Findable, Accessible, Interoperable, and Reusable. Some of the digital twin data also adheres to directives pertaining to the availability of public datasets.

D3.1 Blueprint architecture, functional specifications, and requirements analysis first version

As part of its role, ECMWF 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 data handling and processing components in the DTE 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.

The architectural overview of the DTE is depicted in **Figure 3**.

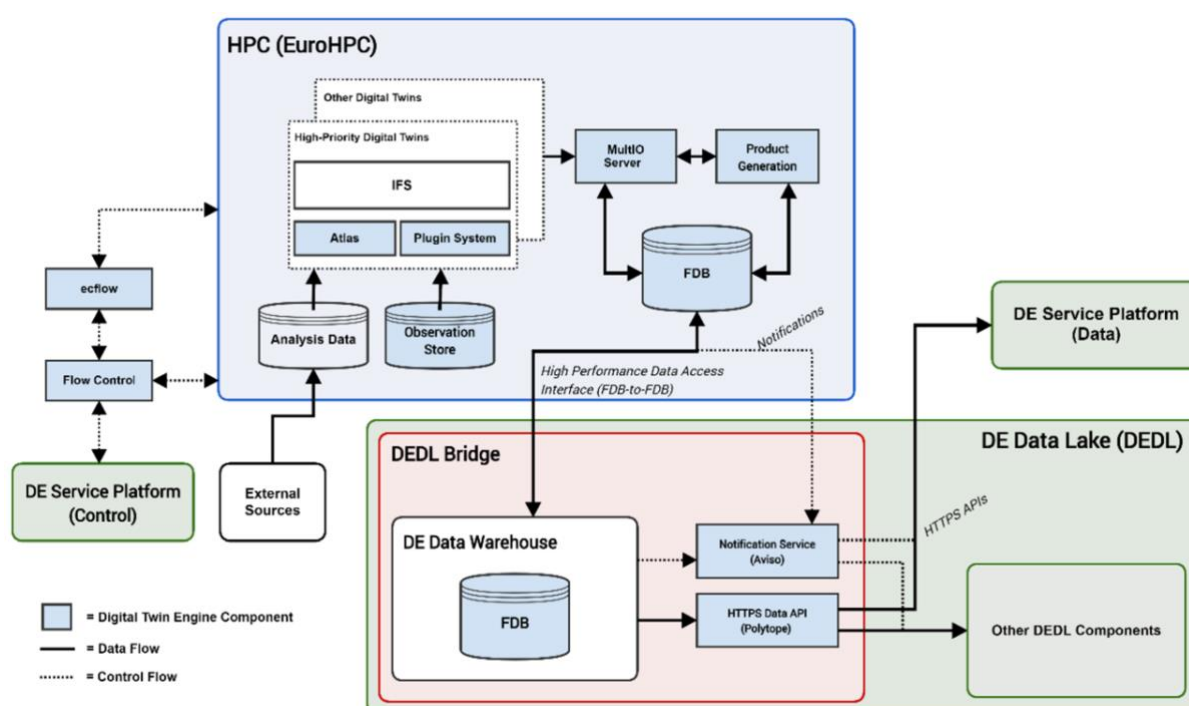


Figure 3 DestinE DTE Architecture

This snapshot offers a representation of the complex structure underlying the DestinE DTE, underscoring its essential role within the broader DestinE initiative.

3.1.3 Data Lake

End-to-end responsibility for the design, establishment, testing, operations, and procurement of the multi-cloud DestinE Data Lake and Data Warehouse has been entrusted to EUMETSAT. As one of the core services, the data lake will accommodate a vast diversity of data spaces. This will include data from EUMETSAT's own Earth observation satellite systems, as well as data from the European Copernicus Sentinel missions, ESA missions, and ECMWF.



D3.1 Blueprint architecture, functional specifications, and requirements analysis first version

This assortment of data will act as the input for the burgeoning digital twin engines, AI, and machine learning algorithms that will generate what is referred to as 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 be initially available via the core service platform and directly accessible in subsequent phases.

Figure 4 provides a visual representation of the DestinE Data Lake's structure and its key role within the overall DestinE initiative.

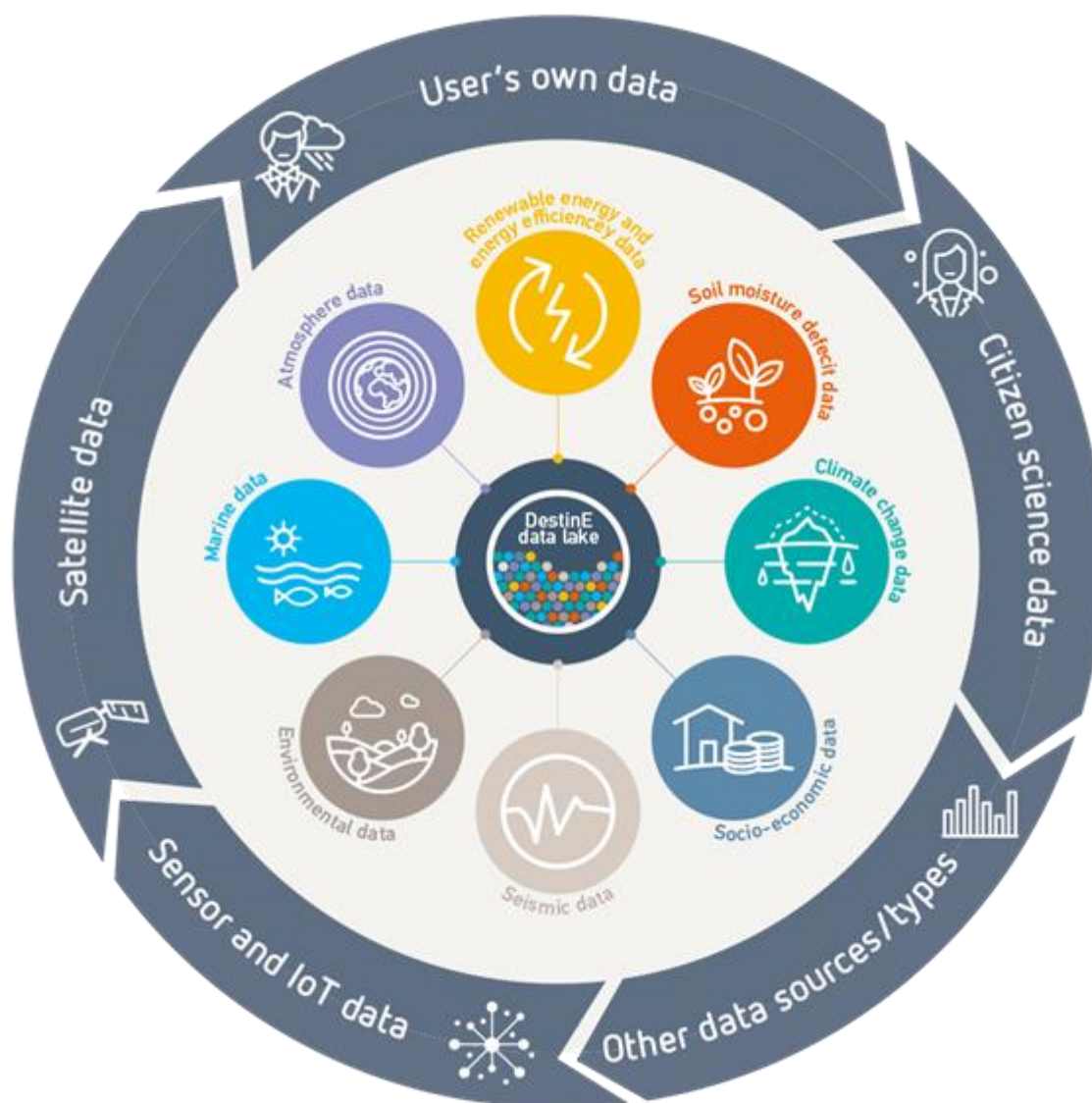


Figure 4 DestinE Data Lake Highlight

3.1.4 Core Service Platform

ESA is responsible for developing the platform serving as a single access point to users to the DestinE ecosystem. DestinE Core Service Platform (DESP) integrates and operates an open ecosystem of services (also referred to as DESP Framework) to support DestinE-data exploitation and information sharing for the benefit of DestinE users and Third-Party entities. The DESP Framework includes essential services such as user identification, authentication, and authorization service; infrastructure as a service with storage, network, and CPU/GPU capabilities; data access and retrieval service, in particular from the DestinE Data Lake operated by EUMETSAT, as it is the backbone for the data generated by ECMWF's Digital Twin Engine; data traceability and harmonisation services; basic software suite service for local data exploitation; data and software catalogue services; and 2D/3D data visualisation service.

3.1.5 Linking activities with DestinE

ECMWF is leading a task (T3.2) in interTwin which has the goal to align the architecture choices of interTwin with what destinE is building. The task aims to study and provide suitable APIs so that the interTwin data becomes accessible to the entire DestinE user base. This includes facilitating the harmonised data access, data fusion and analyses of a wider range of DTs with DestinE external data. As a result, the task should provide a TRL6 proof-of-concept for

- ingesting DestinE DT data for constraining 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

3.1.6 Technology exchange from DG-Connect

Starting from March 2023 an activity of technology exchange between interTwin, DestinE and other projects building DTs and DTE (like DT-Geo⁸, BioDT⁹ and EDITO-Infra¹⁰) has been coordinated by DG-Connect and at the time of writing the deliverable, two meetings have been organised in Bruxelles to discuss mainly coordinated activities between the projects and DestinE and possible alignment on architecture concepts. The analysis of the project's architecture and possible reuse of concepts and architectural choices will be included in the next version of the architecture deliverable D3.2 due in January 2024.

⁸ <https://dtgeo.eu/>

⁹ <https://biodt.eu/>

¹⁰ <https://edito-infra.eu/>



3.2 EOSC and its compute platform

The European Open Science Cloud (EOSC)¹¹ is an environment for hosting and processing research data to support EU science.

The ambition of the European Open Science Cloud (EOSC) is to provide European researchers, innovators, companies, and citizens with a federated and open multi-disciplinary environment where they can publish, find and re-use data, tools and services for research, innovation, and educational purposes.

The EOSC enables a step change across scientific communities and research infrastructures towards

- seamless access
- FAIR management
- reliable reuse of research data and all other digital objects produced along the research life cycle (e.g., methods, software, and publications)

EOSC ultimately aims to develop a Web of FAIR Data and services for science in Europe upon which a wide range of value-added services can be built. These range from visualisation and analytics to long-term information preservation or the monitoring of the uptake of open science practices.

The EOSC High Level architecture¹² is depicted in **Figure 5**.

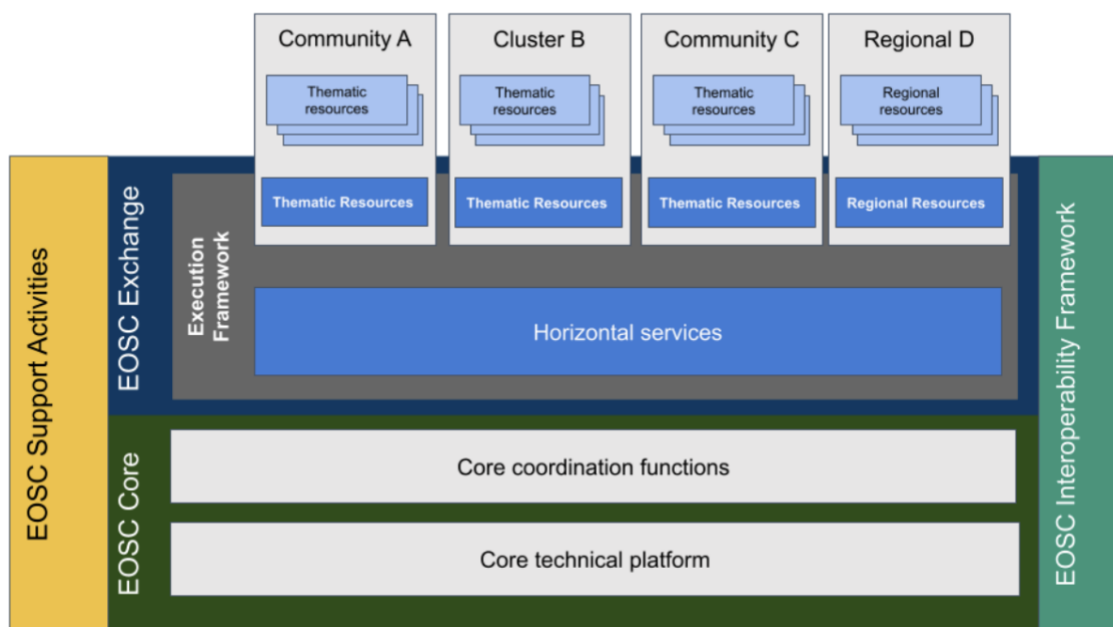


Figure 5 EOSC High Level architecture

¹¹ <https://digital-strategy.ec.europa.eu/en/policies/open-science-cloud>

¹² <https://eosc-portal.eu/sites/default/files/EOSC%20Future-WP3-EOSC%20Architecture%20and%20Interoperability%20Framework-2021-12-22%5B17%5D%5B6%5D-2.pdf>

As seen in **Figure 5**, the main pillars in EOSC are the EOSC Core, the EOSC Exchange and the EOSC Interoperability Framework

3.2.1 EOSC Core

The EOSC Core services include the technical core platform and the coordination functions needed for the operations. Among the core services we can mentioned the AAI, the Helpdesk, the accounting, and Monitoring, and the EOSC Portal¹³ which provides a gateway to information and resources in EOSC. It is maintained by the EOSC Future¹⁴ project that links together other research portals, resources, and services to respond to the needs of a wide range of researchers. The service provision is strengthened by projects funded in INFRAEOSC-07 call: EGI-ACE¹⁵, DICE¹⁶, OpenAIRE Nexus¹⁷, C-SCALE¹⁸ and Reliance¹⁹.

The interTwin DTE will be onboarded in the EOSC Portal depending on the level of maturity reached at the end of the project, together with some of the DT applications.

3.2.2 EOSC Exchange

Set of services storing and exploiting FAIR data and encouraging its reuse. Examples of services included in the EOSC-Exchange are those that store, preserve or transport research data as well as those that compute against it. Horizontal services which are used by multiple Clusters of research communities belong to the Exchange. The possible inclusion of services from interTwin in the EOSC Exchange is one of the outcomes of the project. The Quality framework to be delivered as interTwin core module is an example of such service to be included in the EOSC Exchange.

3.2.3 EOSC Interoperability Framework (EOSC-IF)

The EOSC Future Project is working on the EOSC-IF²⁰ to support interoperability of the various elements of EOSC. This interoperability and composability are twofold:

- internal to EOSC-Core in order to make it operational and to allow communication between components inside the Core
- external to the EOSC-Core to facilitate EOSC providers in:
 - onboarding resources in the EOSC-Exchange (EOSC resource catalogue) and

¹³ <https://eosc-portal.eu/>

¹⁴ <https://eoscfuture.eu/>

¹⁵ <https://doi.org/10.3030/101017567>

¹⁶ <https://doi.org/10.3030/101017207>

¹⁷ <https://doi.org/10.3030/101017452>

¹⁸ <https://doi.org/10.3030/101017529>

¹⁹ <https://doi.org/10.3030/101017501>

²⁰ <https://eosc-portal.eu/eosc-interoperability-framework>



D3.1 Blueprint architecture, functional specifications, and requirements analysis first version

- integrating resources with the EOSC-Core added-value services (such as Order Management, Monitoring, Accounting, Helpdesk).

Interoperability guidelines are being created with the goal to help Resource Providers to integrate within research infrastructures and with the EOSC-Core (where the EOSC-Core components are also interoperable). The EOSC-IF builds upon this existing foundation, creating an overarching framework that encompasses EOSC-Core and the interfaces necessary to accommodate links to community interoperability frameworks.

interTwin from one side will try to reuse some of the EOSC interoperability guidelines in order 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 guideline to be followed is the EOSC AAI guidelines, which in turn is built on the AARC guidelines²¹. While an example for guidelines, contributions from interTwin could be guidelines for the AI Workflow management.

3.2.4 EOSC Compute platform (EGI-ACE Project)

The EOSC Compute Platform, delivered by EGI-ACE²², is a free at-the-point-of-use, distributed computing environment. The Platform is built on a hybrid infrastructure composed of cloud computing resources, High-throughput computing (HTC) sites and High-Performance Computing (HPC) centres. It empowers users with higher-level services to ease the setup and operation of complex workflows, applications, containers, virtual research environments and data spaces on top of the hybrid infrastructure.

The Platform supports diverse data processing and analysis use cases. Thanks to EC and national funds, it provides free at-the-point-of-use services with user support and training for research infrastructures, communities, projects, and the long tail of science. E-infrastructure providers joining the EOSC Compute Platform can benefit from the simplified integration with EOSC, streamlined user access handling and scalable resource allocation mechanisms, and various financial incentives.

²¹ <https://aarc-project.eu/guidelines/>

²² <https://www.egi.eu/project/egi-ace/>



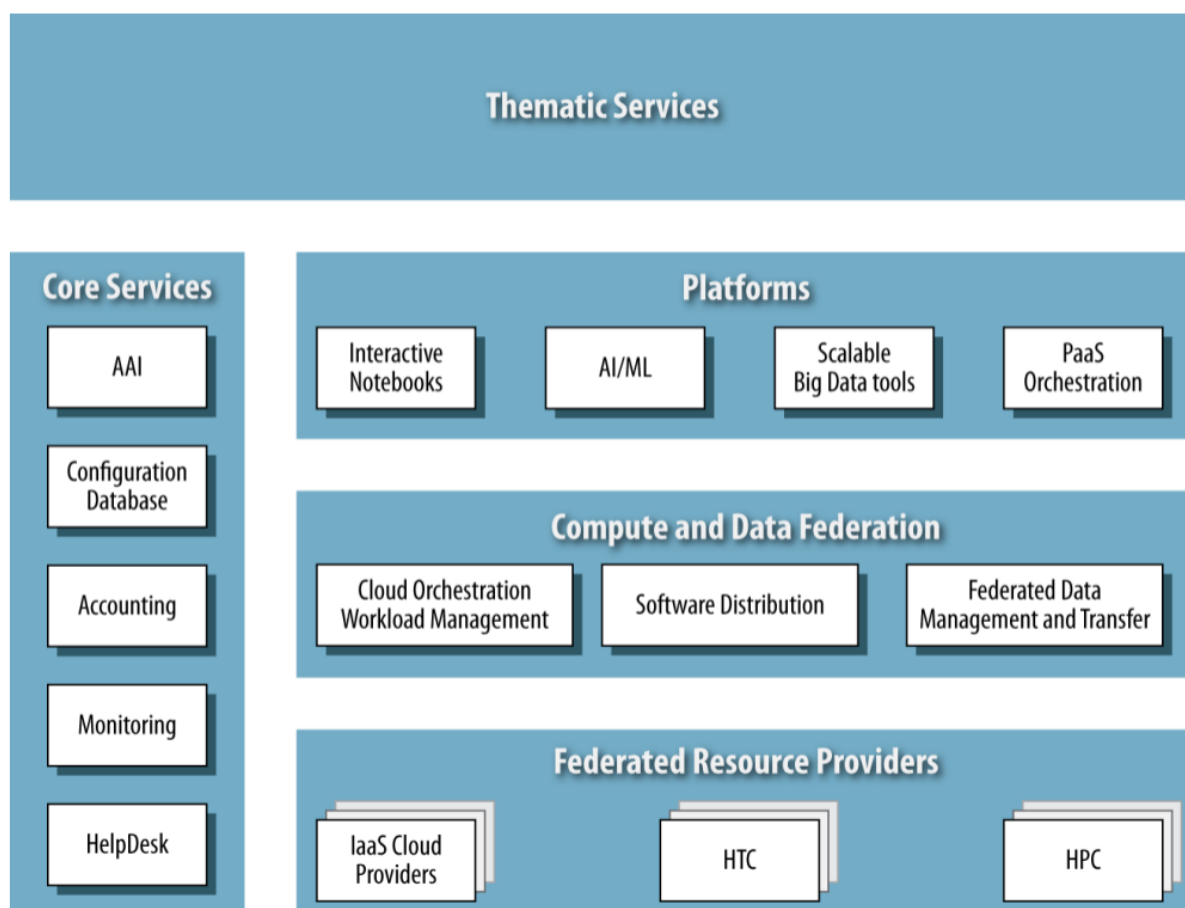


Figure 6 EOSC Compute Platform functional block diagram

The EOSC Compute Platform architecture is organised in functional blocks, as shown in **Figure 6** at the bottom of the architecture, the Federated Resource Providers deliver a hybrid infrastructure for hosting research applications and data. Different types of providers are considered in the EGI-ACE technical architecture:

- IaaS Cloud Providers provide access to Virtual Machine-based computing with associated Object and Block storage.
- HTC and HPC provide access to large, shared computing systems for running computational jobs at scale.

The Core Services pillar provides the necessary functionality to assist services of all other areas with the integration into the Federation. These services support the operation of the EOSC Compute platform and integrate and interoperate with the EOSC Core. Examples of Core Services are:

- Configuration Database, which contains detailed information about the whole infrastructure of the Federation.
- Accounting, which tracks the use of resources over time.
- Monitoring, which oversees the status of resources and provides an insight on their availability.

D3.1 Blueprint architecture, functional specifications, and requirements analysis first version

- AAI, which is implemented via EGI Check-in, EGI's Authentication and Authorisation service, is a key component of the architecture that enables a single identity manager to be used across all the layers and services of the EOSC Compute platform. This ensures a unified mechanism for controlling access permissions to resources.
- Helpdesk, which provides human support to end-users.

Other services are also included in this area, ranging from other non-technical services, coordination activities such as Operations Management, and Security and Incident Response. The Compute Federation services orchestrate the execution of user workloads in the Federated Resource Providers. These support the exploitation of data locality by moving computation close to the data and facilitate application portability with the support for a diverse range of computing platforms (Cloud IaaS, HTC, HPC) and the interaction with software distribution tools (such as VM images, container images or plain binaries).

There are three services considered in the architecture:

- Hybrid cloud orchestration, for the deployment of custom virtual infrastructure over multiple IaaS cloud backends;
- Workload Manager, for the scheduling and execution of jobs in the federated resource providers (both cloud and HTC/HPC); and a
- Software distribution, for making software available at the Federated Resource Providers (e.g., as VM images).

The Data Federation services support the exposure of discoverable datasets and the staging of data in and out of the EOSC Compute Platform Cloud. The Federated Data Management services control the raw storage capacity offered by the Federated Resource Providers to deliver data products that, by using the Data Transfer service, can be transferred between the different EGI-ACE providers and potentially to external data repositories.

The Platforms service layer provides generic added-value services to exploit the compute and storage resources of EGI-ACE, which can be easily reused by different communities to build thematic end-user services. These platforms rely on the existing Compute Federation and Data Federation services to access the Federated Resource Providers and deliver Interactive Notebooks as a tool for interactive analysis of data, PaaS Orchestration to facilitate the deployment of complex applications, AI/ML to support the models and algorithms required by some scientific fields, and Scalable Big Data Tools that can be reused in several research disciplines.

Finally, Thematic Services is built on top by combining services from all these areas to bring simulation, machine learning and data analytics capabilities that are tailored to the needs of a specific research domain. These data spaces and tools focus on data exploitation.

The link between interTwin and EGI-ACE are various. Starting from the project coordination by EGI Foundation and some common partners (e.g., UPV, DESY, INFN etc). Common partners also mean common technologies between the two projects. Based



on the successful experience in EGI-ACE, some technologies will be reused in the context of interTwin, such as the PaaS Orchestrator and the Infrastructure Manager. Finally, the DTE infrastructure will also use some of the cloud providers that are federated in the EGI Fedcloud.

3.3 ESCAPE

ESCAPE²³ (European Science Cluster of Astronomy & Particle physics ESFRI research infrastructures) is a Horizon 2020 project²⁴ ended in January 2023, which aimed at addressing the Open Science challenges shared by ESFRI facilities (SKA, CTA, KM3Net, EST, ELT, HL-LHC, FAIR) as well as other pan-European research infrastructures (CERN, ESO, JIVE) in astronomy and particle physics.

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

- 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 expertise in computing and data management by setting up a data infrastructure beyond the current state-of-the-art in support of the FAIR principles. These joint efforts resulted in a data-lake infrastructure as a cloud open-science analysis facility linked with the EOSC.

interTwin, apart from sharing some of the communities with ESCAPE (CERN and VIRGO), could benefit from the data lake architecture developed in the project, and named DIOS²⁵.

3.3.1 DIOS Architecture

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 is composed of a bulk data transfer service and a storage orchestration service, which allows access to seamlessly access a heterogeneous storage infrastructure. In particular the File transfer functionality is implemented by the FTS²⁶ service and the Data orchestration by the

²³ <https://doi.org/10.3030/824064>

²⁴ <http://doi.org/10.13039/100010661>

²⁵ <https://projectescape.eu/services/data-infrastructure-open-science-dios>

²⁶ <https://fts.web.cern.ch/fts/>



Rucio²⁷ service both developed at CERN. FTS is a collection of servers and clients that allow for the automated scheduling and execution of remote files transfers, while Rucio is a software framework that provides functionality to organise, manage, and access large volumes of scientific data using customisable policies. **Figure 7** shows the overview of DIOS.

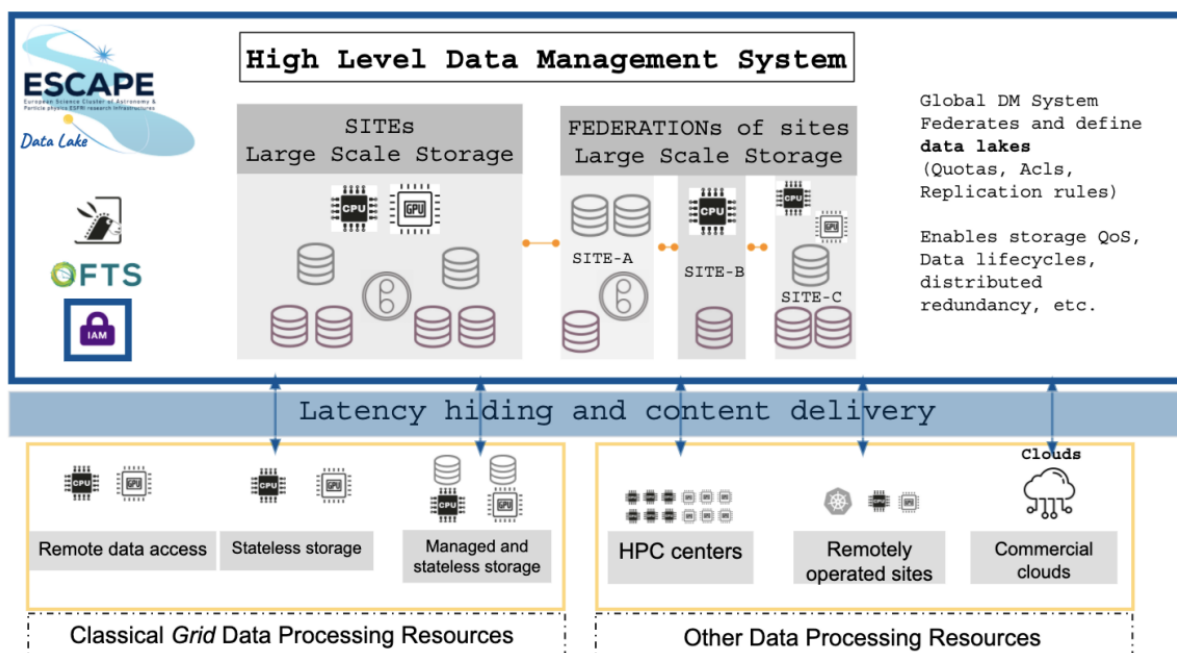


Figure 7 DIOS overview

interTwin DTE Data Management could be based on DIOS given the similarities to deliver 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 deliverable D5.1 [R1].

3.4 C-SCALE

C-SCALE²⁸ is a Horizon 2020²⁹ project with the aim of helping 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 service offer of the EOSC Portal, providing up-

²⁷ <https://rucio.cern.ch/>

²⁸ <https://doi.org/10.3030/101017529>

²⁹ <http://doi.org/10.13039/100010661>



D3.1 Blueprint architecture, functional specifications, and requirements analysis first version

to-date research and enabling services to its users. It gives an open, clearly explained system for incorporating new service providers and application developers.

C-SCALE makes unique data resources and the Copernicus community's body of knowledge accessible to new audiences and user communities more user-friendly through the EOSC portal. It provides a modular, open, and robust federation for discovering, processing, and exploiting Copernicus and, more generally, Earth observation data.

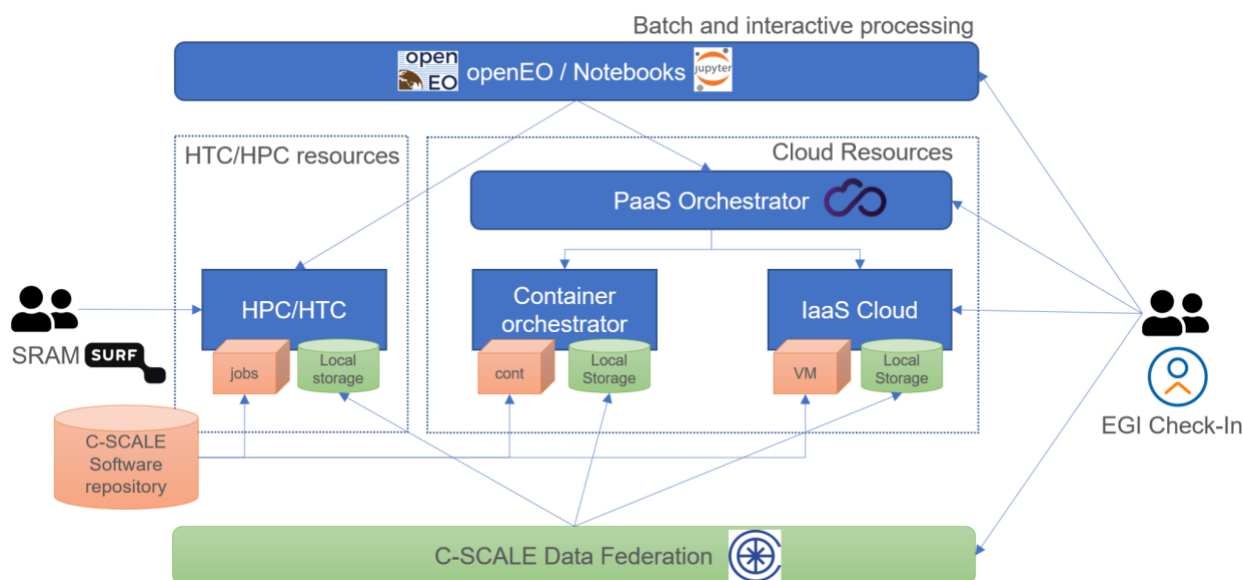


Figure 8 C-Scale Architecture

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

3.4.1 Federated Earth System Simulation and Data Processing Platform (FedEarthData)

The Federated Earth System Simulation and Data Processing Platform³⁰ built by C-SCALE provides a distributed infrastructure of data and compute providers to support the execution of Earth System Simulation and Data Processing workflows at scale to EOSC researchers.

It offers a flexible cloud-based data processing capacity to create and scale data processing pipelines that run on optimised execution environments near the data. Jupyter Notebooks and openEO³¹ API offers a user-friendly and intuitive processing of a wide variety of Earth Observation datasets on these computing providers, including the ability to integrate these data with modelling and forecasting workflows leveraging specialised compute resources.

³⁰ <https://marketplace.eosc-portal.eu/services/eosc.egi-fed.fedearthdata>

³¹ <https://openeo.org/>



Providers of the Copernicus Data Processing Platform already count with an extensive collection of Copernicus datasets, managed according to the FAIR principles, and may be further extended with new datasets requested by users of the platform.

3.4.2 Earth Observation Metadata Query Service (EO-MQS)

The C-SCALE Earth Observation Metadata Query Service (EO-MQS)³² makes Copernicus data distributed across providers within the C-SCALE Data federation discoverable and searchable. The EO-MQS is a STAC³³-compliant service that exposes all collections available within the federation on a single endpoint. Through a search interface, users' queries are redistributed among the data providers and a consolidated list of results is returned.

Thanks to the rich ecosystem that has evolved around STAC and the growing list of tools that can interact with STAC APIs, working with the EO-MQS is straightforward. Data providers maintaining their data assets in STAC catalogues can easily integrate them into the EO-MQS to increase their discoverability.

3.4.3 openEO Platform

The openEO platform³⁴ is a platform service implementing the openEO API in a EO cloud platform federation (see figure 9) and provides intuitive programming libraries to process a wide variety of Earth Observation datasets including the Copernicus satellite missions such as the sentinels. This large-scale data access and processing federation is performed on multiple infrastructures, which all support the openEO API. This allows use cases from explorative research to large-scale production of EO-derived maps and information. Users can interact with the data and platform capabilities through hosted Jupyter notebooks. Additionally, they can engage with programming-less graphical web applications like the openEO web editor. Client libraries are provided in multiple programming languages catering to the needs of various research communities from Python over R to JavaScript. A single sign-on mechanism based on OIDC provided by EGI Check-In allows for an efficient identity management of users. Currently, a wide variety of researchers, developers and EO data specialists are using the Platform for their needs. It provides an operational service in the context of the European Space Agency and their Network of Resources (NoR) and is available for both the private sector and the research community.

³² <https://eo-mqs.c-scale.eu/browser>

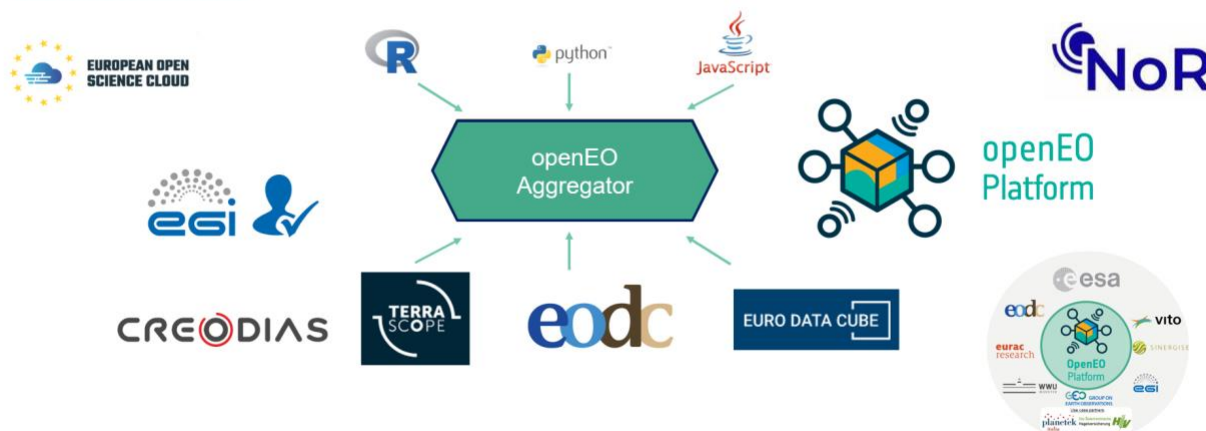
³³ <https://stacspec.org/en>

³⁴ <https://openeo.cloud/>





<https://openeo.cloud>



THE EUROPEAN SPACE AGENCY

Figure 9 openEO platform federation

3.5 Digital Twin Consortium

The Digital Twin Consortium³⁵ drives the awareness, adoption, interoperability, and development of digital twin technology. Through a collaborative partnership with industry, academia, and government expertise, the Consortium is dedicated to the overall development of digital twins.

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 DigitalTwins when defining the BluePrint architecture.

The definition³⁶ of a Digital Twin from the consortium:

- **A digital twin is a virtual representation of real-world entities and processes, synchronised at a specified frequency and fidelity.**
- **Digital twin systems transform business by accelerating holistic understanding, optimal decision-making, and effective action.**
- **Digital twins use real-time and historical data to represent the past and present and simulate predicted futures.**

³⁵ <https://www.digitaltwinconsortium.org>

³⁶ <https://www.digitaltwinconsortium.org/initiatives/the-definition-of-a-digital-twin/>



- **Digital twins are motivated by outcomes, tailored to use cases, powered by integration, built on data, guided by domain knowledge, and implemented in IT/OT systems.**

Together with the Digital Twin definition, a Glossary³⁷ terms have been also defined to be considered when comparing different initiatives dealing with Digital Twins and Digital Twins Engines/Systems.

In particular the following Digital Systems high-level architecture is defined.

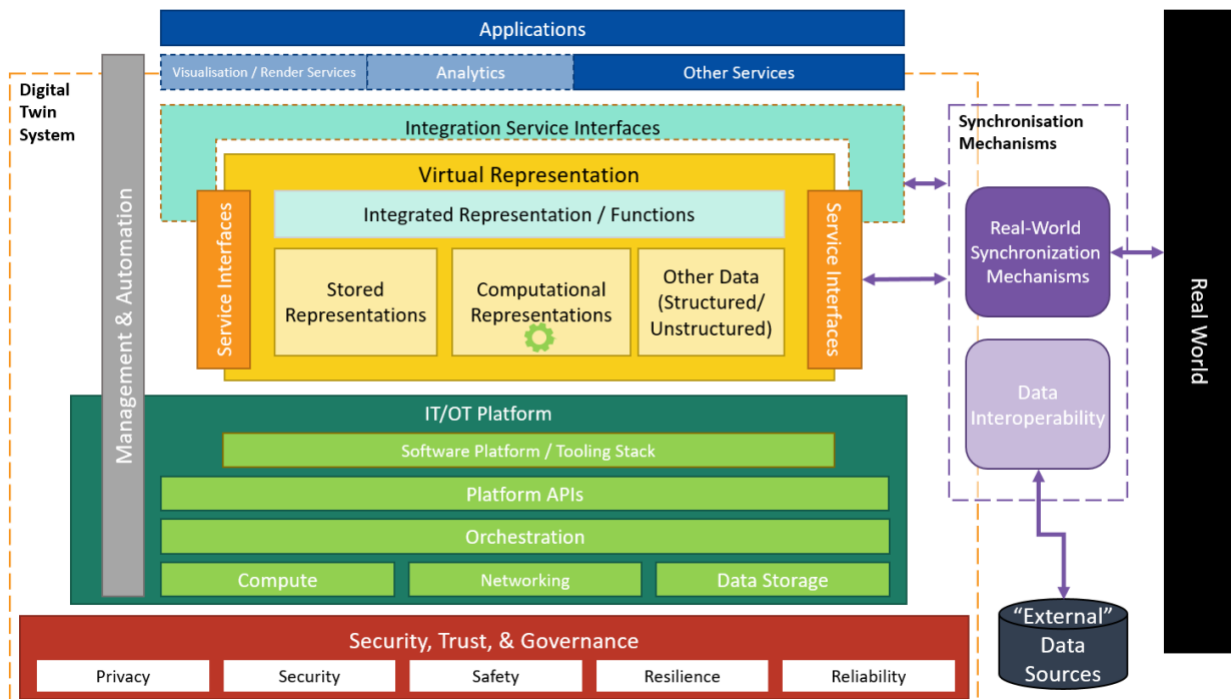


Figure 10 Digital Twins System

The participation to the consortium and therefore WGs (in particular the one on research and academia) is something to be considered by the project and will be discussed in WP2 activities.

3.6 Gaia-X

The goal of Gaia-X³⁸ is to build a secure and federated data infrastructure that stands for European values, digital sovereignty of the data owners, interoperability of different platforms and open source. Within this ecosystem, it will be possible to provide, share, and use data within a trustworthy environment. Thus, spurring innovation and creating added value for the data economy to all who share data.

Gaia-X started as an initiative by the former German Minister of Economic Affairs Peter Altmaier and his French counterpart in 2019. In addition to this project on secure data sharing, the Franco-German collaboration also involved cooperation in the field of

³⁷ <https://www.digitaltwinconsortium.org/glossary/>

³⁸ <https://gaia-x.eu/>



D3.1 Blueprint architecture, functional specifications, and requirements analysis first version

Artificial Intelligence (AI), and a joint approach to a European data infrastructure that will preserve and expand Europe's digital sovereignty.

In early 2022, the initial implementation of Gaia-X started with the launch of the first data spaces and related services (e.g., Mobility Data Space).

The latest architecture document³⁹ shows the Gaia-X framework which is being implemented (with some of the components available and under integration) based on the four pillars as shown in **Figure 11**.

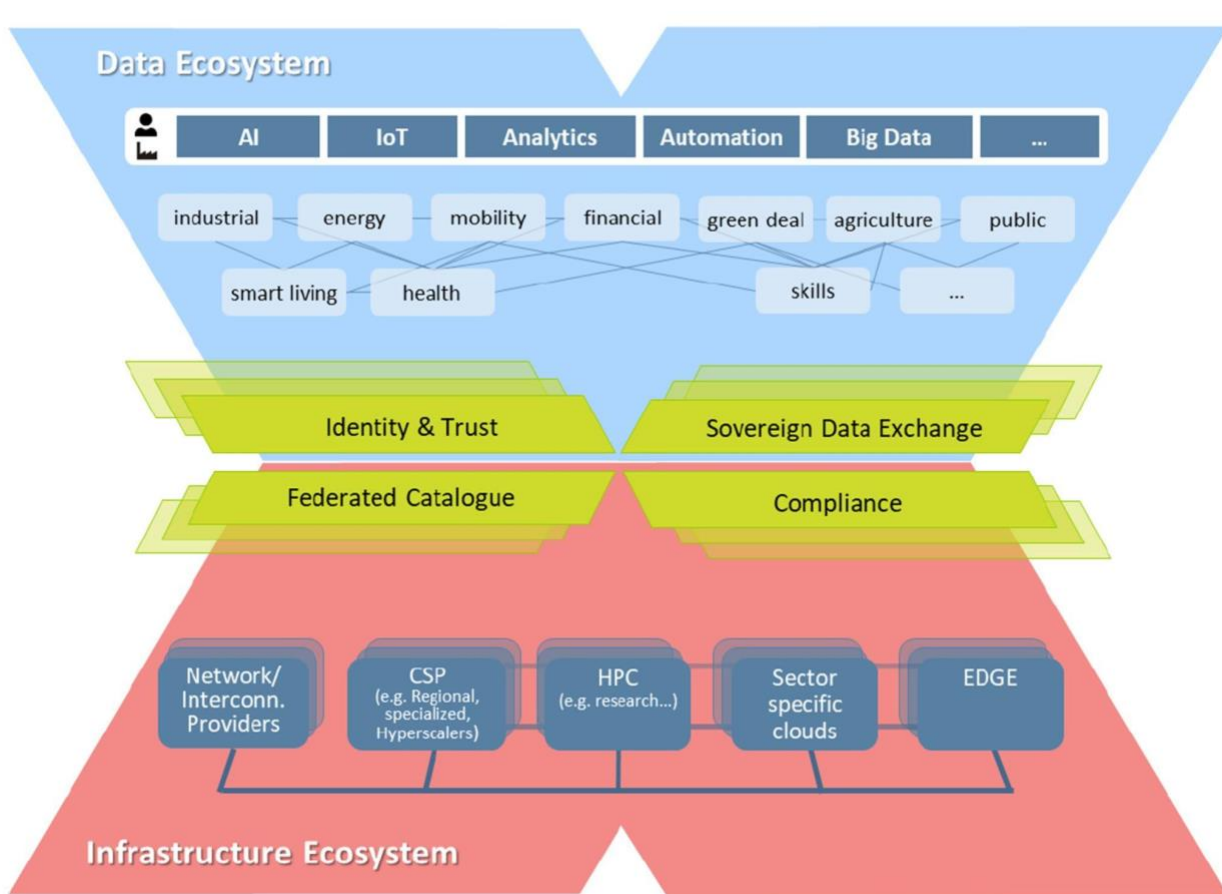


Figure 11 Gaia-X High Level conceptual architecture

The activity of Gaia-X is also driven by WGs where members of the Gaia-X ASBL can participate and contribute to. For instance, both EGI Foundation and INFN as Gaia-X members are contributing to Architecture and AAI WG.

Possible evolution of the framework could be the reuse of some of the components within the SIMPL framework being procured by EC (see next section).

³⁹ <https://docs.gaia-x.eu/technical-committee/architecture-document/22.10/>

3.7 EU Data Spaces

The European Strategy for data⁴⁰ Space initiative aims at creating 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 for use 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 12**.

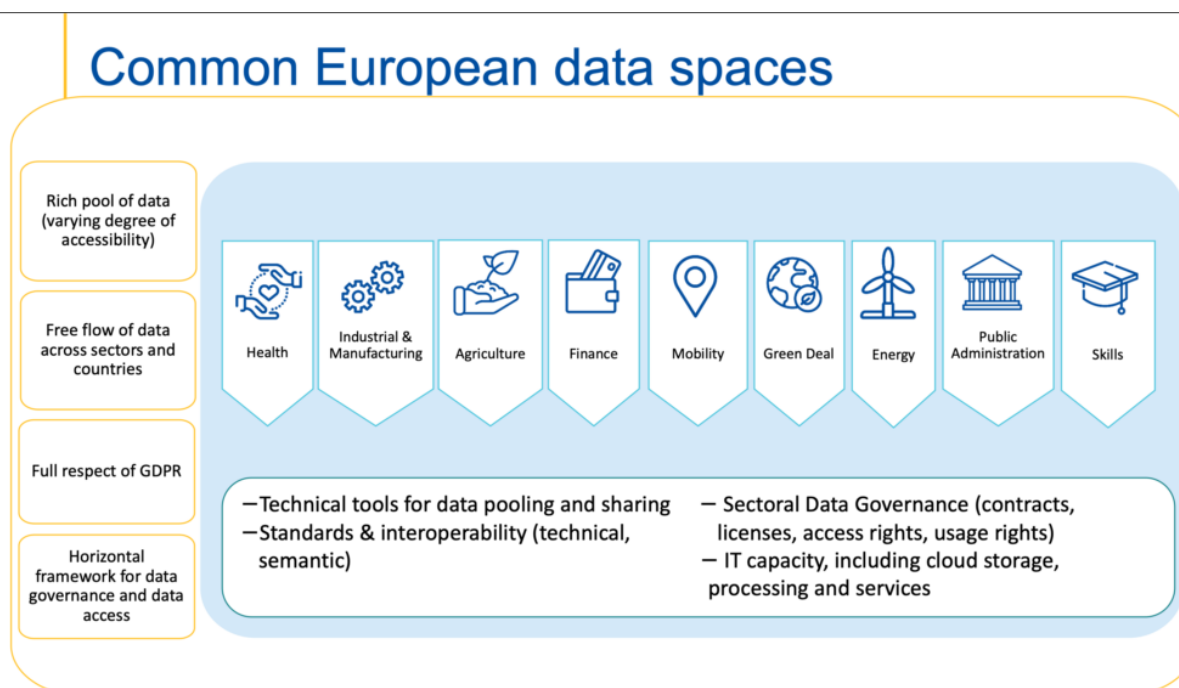


Figure 12 Common European Data Spaces

The technical foundation for all those Data Spaces is currently under procurement by the EC via the SIMPL⁴¹ framework, which is depicted in **Figure 13**.

⁴⁰ <https://digital-strategy.ec.europa.eu/en/policies/strategy-data>

⁴¹ <https://digital-strategy.ec.europa.eu/en/news/simpl-cloud-edge-federations-and-data-spaces-made-simple>

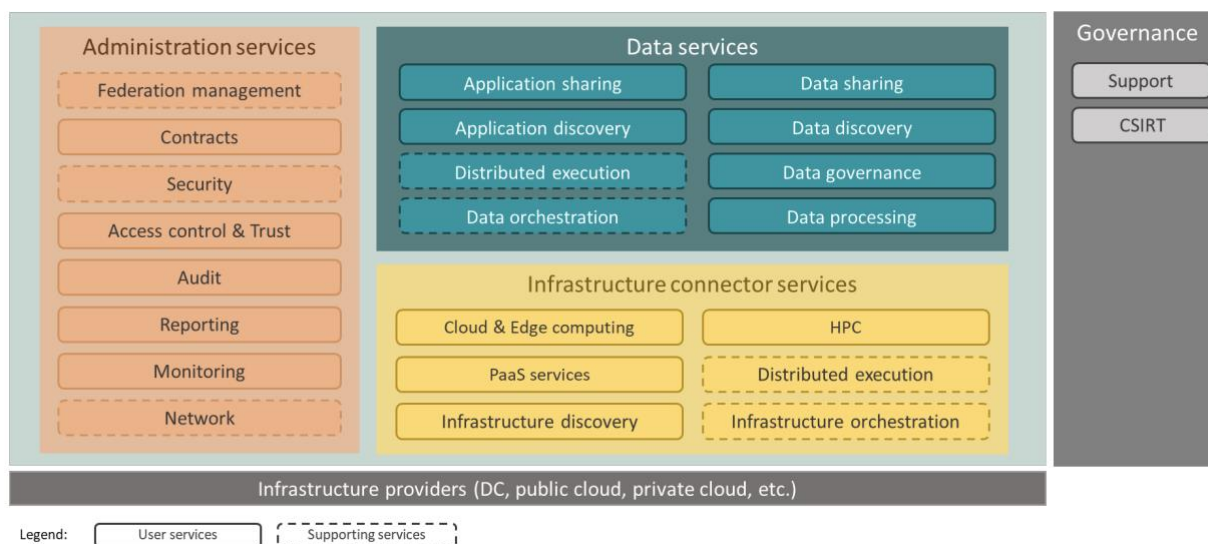


Figure 13 High-level overview of SIMPL capabilities and architecture layers

SIMPL has been organised following this structure:

- **SIMPL-Open.** The open-source software stack as envisaged in preliminary study and shown in **Figure 13**, over which tenderers will be able to elaborate their own proposal.
- **SIMPL-Labs.** A pre-installed demonstration/playground environment where third parties (typically sectoral data spaces in their early stages of inception) can experiment the deployment, maintenance, and support of the open-source software stack before deploying it for their own needs.
- **SIMPL-Live.** Several instances of the SIMPL-Open software stack in the form of customised production environments for sectoral data spaces where the European Commission itself plays an active role in their management.

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

- A Minimum Viable Platform released by the *beginning of 2024*.
- In parallel and starting as early as possible in 2024, the open testing environment (SIMPL-Labs) will be made available for stakeholders to experiment with.
- Progressively on-boarding and integration of use cases, helping them to adjust SIMPL to their specific needs (without compromising its generic nature).
- The roadmap foresees major new releases every 6 months.

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

3.8 Summary of input to the blueprint architecture and DTE implementation

The previous sections have described and analysed some of the prominent initiatives which are giving input to both the definition of the interTwin blueprint architecture and the actual DTE implementation. Most of them have been already identified at proposal stage and included in the DoA and some of them are not in a design or implementation phase that could at this stage be used as input for interTwin, therefore will be analysed in subsequent versions of this deliverable. **Table 1** summarises the information collected.

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

initiatives / Projects	Relevance for interTwin	Actions
DestinE	ECMWF partner of interTwin, DT of Extremes	Planned piloting activities in the context of T3.2 and technological exchange (both ways)
EOSC	EOSC Portal onboarding of services, EOSC IF	Adhere to EOSC IF in some relevant areas (e.g., AAI), contribute to the EOSC IF by implementing guidelines for DTE
EGI-ACE	Implement EOSC Computing platform	Extend the DTE infrastructure with providers coming from the EGI Federation and EOSC Computing platform
ESCAPE	ESCAPE Data Lake Blueprint	Adoption of the ESCAPE data lake Blueprint and services in interTwin
C-SCALE	Access of Copernicus data federation, possible technology exchange (openEO, EO-MQS based on STAC)	Understand from partners in interTwin part of the C-SCALE project (EODC, LIP, DELTARES, etc.) the data access and technology contributions
openEO platform	Implements data access and processing federation based on openEO API and common	Understand from partners in interTwin part of the openEO platform project

D3.1 Blueprint architecture, functional specifications, and requirements analysis first version

	process-graph definition	(EODC, EURAC, WWU, TU Vienna) the data access and technology
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 next version of the Blueprint architecture
GAIA-X	Cloud and Sovereign data federation in Europe	Participation to technical WGs and architecture alignment
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 2024
TECH-01-2021 Projects	"Sister projects" funded in the same call as interTwin	Analysis of architectures and synergies to be put in place also thanks to DG-Connect driven initiative.



4 Digital Twin Engine Blueprint Architecture

In this chapter, we will provide a comprehensive overview of the key components of a Digital Twin Engine and how they work together to create a powerful and efficient system. First, it introduces the methodology, the targeted conceptual model and the categorization of users using the system. Then an overall view of the architecture is presented followed by the DTE building blocks and components.

4.1 Methodology

The description of the architecture of the Digital Twin Engine follows the C4 methodology [R5], which is an easy way to describe complex software systems. As described by Brown⁴²:

- The adoption of agile methodologies has led to a reduction in the production of software diagrams. In the instances they are made, they frequently exhibit a lack of clarity and understanding.
- The structured levels of the C4 diagrams offer varying degrees of abstraction, each catering to a distinct audience.
- The C4 model avoids misunderstandings in diagrams, incorporates a significant amount of text, and provides a key or legend to decipher the notations used.

The C4 model represents a framework used for visualising the architecture of a software system. It breaks down the architecture into four different levels: Context, Container, Component, and Code.

- **Context** corresponds to the high-level view of a system, showing how it interacts with other systems, software, and users. It depicts the overarching layout of a software system, outlining its boundary and how it communicates with external entities.
- **Container** diagrams go a level deeper, breaking down the system into separate high-level technology-specific sections or containers. Each container typically runs a part of the software system, and can include elements such as web servers, databases, desktop applications, or mobile apps.
- **Component** diagrams provide an even more detailed view. They break down each container into smaller parts, called components, representing separate functional areas of the system.
- **Code** corresponds to the lowest level in the C4 model. It involves actual implementation details, with the code itself presented using UML class diagrams or similar representations.

⁴² <https://www.infoq.com/articles/C4-architecture-model/>



D3.1 Blueprint architecture, functional specifications, and requirements analysis first version

The C4 model provides a framework through which the structure of a software system may be readily understood.

Given the scope of this document, we will primarily focus on the Context and Container abstractions of the C4 model. More granular details, including Component level information, are covered in other deliverables of this project [R1], [R2], [R3], [R4].

4.2 Conceptual Model

The blueprint architecture provides an overall framework for the Digital Twin Engine (DTE), defining its boundaries and its interfaces with the surrounding environment based on building blocks.

The Digital Twin Engine is designed to be an open-source integrated platform based on open standards, APIs and protocols that offers the capability to integrate with application-specific Digital Twins. Its functional specifications and implementation are based on a co-designed interoperability framework and conceptual model of the DT for research.

The blueprint architecture is an instrument to introduce a level of abstraction that is sufficient to meet the requirements of the use cases for which different DTs are designed. As illustrated in **Figure 14** (interTwin DTE conceptual model), the interTwin DTE is organised in three functional areas: Infrastructure, Core and Thematic capabilities.



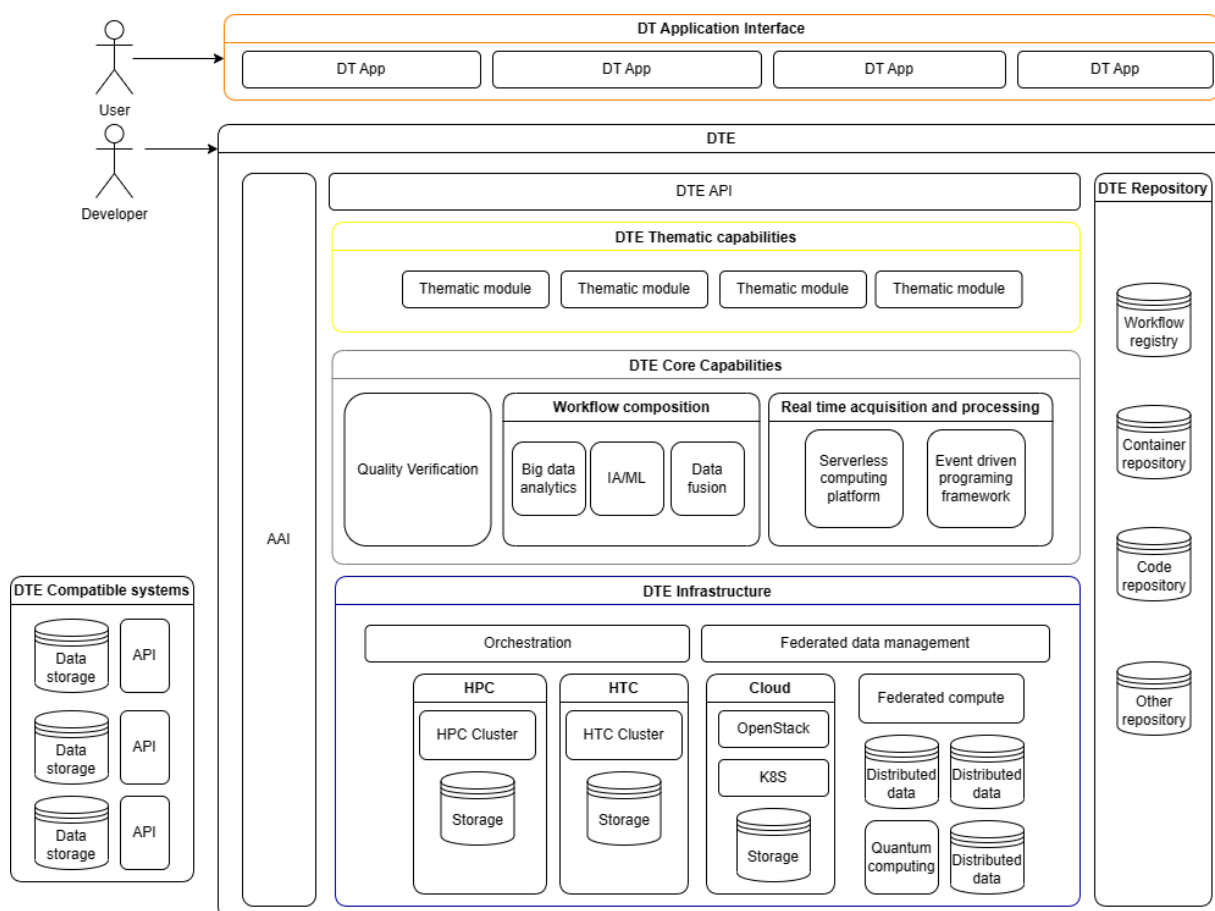


Figure 14 interTwin Digital Twin Engine conceptual model

Based on open-source software components, 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 also outside the project use cases. (SMEs, industry, and evidence-based policy makers). This will require a progressive shift of DT applications from the development of in-house solutions towards an increasing 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 have to integrate with application-specific data and compute facilities. It needs to integrate with current/future infrastructures from national to pan-European level, such as the European Open Science Cloud. The DTE infrastructure is sustained by investments in digital infrastructures.
- **PaaS and SaaS provisioning:** The DTE needs to provide a Platform-as-a-Service (PaaS) layer for development of custom applications and creating a user work environment integrating relevant data to be accessed by the modelling and simulation tasks and Software-as-a-Service (SaaS) layer for consuming the functionalities of the digital twins as dedicated services.

D3.1 Blueprint architecture, functional specifications, and requirements analysis first version

This system encompasses two primary user groups: Digital Twin (DT) developers and DT Users, the latter being primarily scientists using existing DT applications for their research or policy makers.

DT developers engage with the interTwin Platform as a Service (PaaS), developing DT applications and occasionally thematic modules that offer new domain-specific tools and best practices. These are subsequently translated into workflows by an underlying composition tool. The applications crafted by DT developers are in response to researchers' needs and are made accessible to users as Software as a Service (SaaS).

DT users have the option to select a pre-made DT application and connect it to their specific use case, or they can re-train it if necessary. The interactions between the DT Users and the DT application are facilitated through a user interface provided by the specific DT application.

Part of the DTE architecture includes an infrastructure DT provider that offers computational resources, storage, and a connection with the physical twin existing in the real world. This connection may occasionally require the use of intermediate infrastructures, such as satellites or particle detector sensors, typically situated near the use case. These intermediate infrastructures are necessary to filter, pre-process, or buffer the real-time data collected from the physical twin before it is processed by the infrastructure DT provider.

The security of digital twin models is essential to prevent a range of potential risks. These include unauthorised access, which could lead to manipulation or theft of data; threats to availability, which could cause disruptions in operations; durability risks, which could result in loss of data or system failures; and performance risks, which could adversely impact the efficiency and effectiveness of the digital twin's simulations. Consequently, robust measures are required to safeguard against these varied types of security threats, as a compromised digital twin can lead to significant issues, both in the virtual and real world.

The Authentication and Authorization Infrastructure (AAI) is key for both user and providers, and represents an integral part of the DTE infrastructure, as discussed in **Section 4.8**. Key features of AAI include: data protection, integrity of simulations and workflows, trust, and compliance, and securing interconnected systems (i.e., infrastructure).

4.3 DTE engine users

As introduced above, the interTwin Digital Twin Engine will serve mainly three categories of users:

- **DT developers** interact with interTwin DTE, seen as PaaS, developing DT applications and occasionally thematic modules tailored to the needs of specific user communities.
- **DT Users** access the DTE as a SaaS 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 Provider** provides computational resources, storage, and eventual connection with the physical twin living in the real world.

Given this categorization, the document describes how DT developers and DT users interact with the system using the C4 model. However, the interactions of DT providers are not depicted in the current C4 diagram, as they were not discussed during the initial iteration and interviews with the project partners.

4.4 Architecture Model Specification

This section, Architecture Model Specification, will describe the architecture of the Digital Twin Engine (DTE) using the C4 Model as a framework. The C4 Model provides a structured approach to visualise the static structure and architecture of the system at different levels of detail.

In **section 4.4.1**, the System Context diagram will be introduced, which illustrates how the DTE interacts with users and other systems in its environment. It sets the foundation for understanding the role of the DTE within a larger ecosystem.

Following the System Context diagram, **section 4.4.2** will present the Container Diagram. This diagram provides a high-level technology-centric view of the software system, dividing the system into containers, such as server-side applications, databases, and client-side applications.

The purpose of this section is to provide a clear, technical understanding of the DTE's architecture, outlining its core components and their interactions. The details provided in this section will be critical in guiding the technical implementation of the system, as well as informing how users interact with the Digital Twins.

4.4.1 System context diagram

The Context Diagram for the Digital Twin Engine (DTE), presented in **Figure 15**, is composed of four main blocks:

1. **DT Applications:** This block represents the software applications which developers use to create, manage, and configure the Digital Twins (DTs), and which DT Users use to execute the DTs for their research purposes.
2. **DT Thematic Modules:** The Thematic Modules form the second block, providing specific functionalities that DT Applications use to execute their processes and achieve their objectives. These modules cater to specialised requirements, and they can vary based on different application areas or themes. Developers have access to these modules to customise the DTs according to specific needs.
3. **DT Core Capabilities:** This block encompasses essential functionalities necessary for maintaining the integrity and efficiency of the DTE. Core capabilities include quality control (QC) and verification functions, workflow generation capabilities, and data acquisition and processing using Machine Learning (ML). Developers can interact with



D3.1 Blueprint architecture, functional specifications, and requirements analysis first version

these core functionalities to fine-tune the operation of DT Applications, ensuring that the applications perform as intended.

4. **DT Infrastructure:** The final block represents the underlying data infrastructure and server orchestration capabilities. This includes storage, processing, and management of data related to DTs, as well as the orchestration of servers to ensure efficient execution of all operations in the DTE.

Together, these four blocks constitute the overall System Context of the DTE, demonstrating how different elements of the system interact with each other and with users, both developers and users.

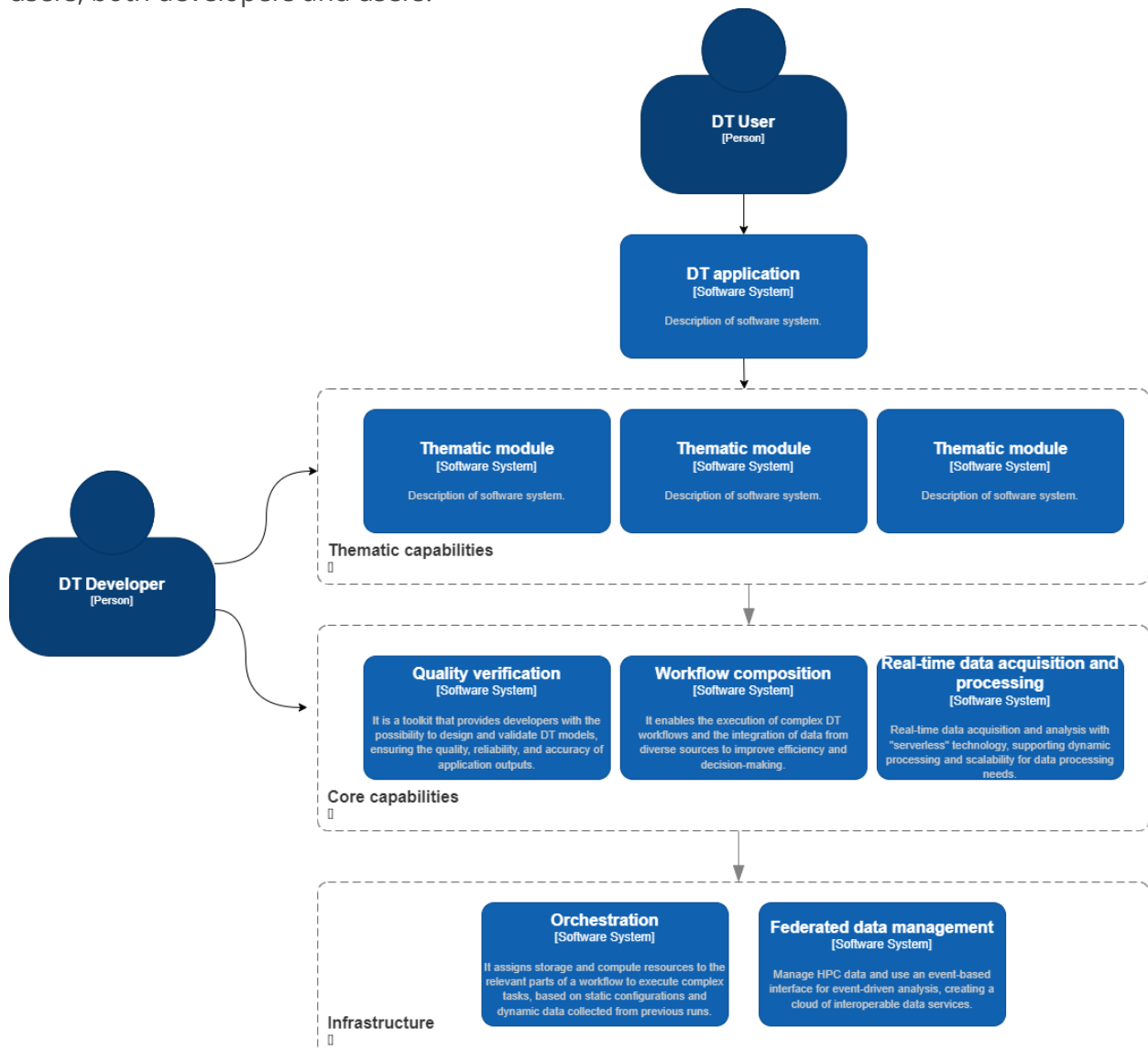


Figure 15 interTwin Digital Twin Engine System context diagram

4.4.2 Container diagram

This section elaborates on the container diagram that serves as a high-level blueprint for the architecture of the DTE. This diagram (Figure 16) is derived from an analysis of the preliminary requirements interviews with various project partners. It provides a



D3.1 Blueprint architecture, functional specifications, and requirements analysis first version

technology-agnostic system overview, revealing how server-side applications, databases, and client-side applications interconnect to deliver functionality.

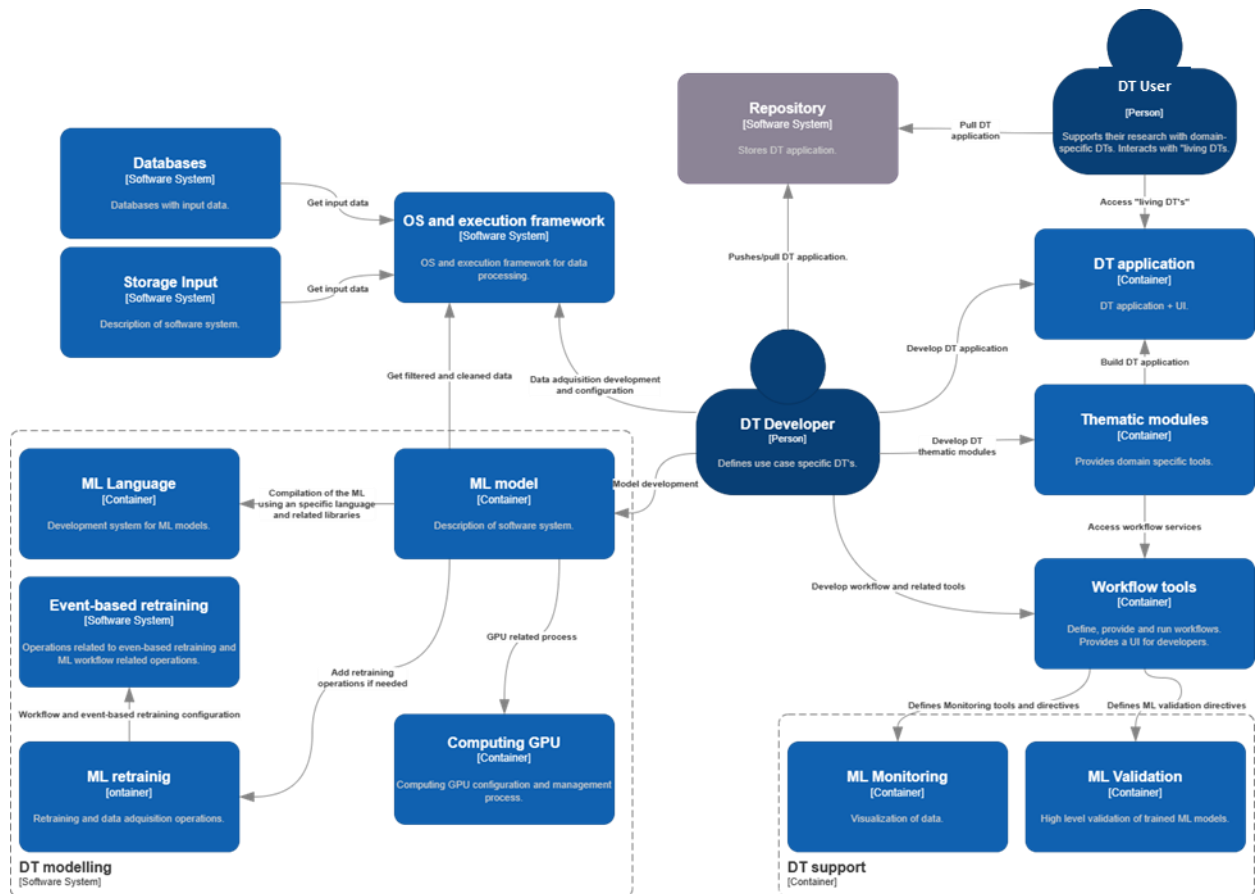


Figure 16 Container diagram of the DTE

Two primary user types interact with the DTE: the DT User and the DT Developer.

The DT Users interact with the Repository and DT Application. The Repository acts as a catalogue where the DT User can select and access the required DT Applications. Once selected and instantiated, these applications are used directly by the DT User within the DTE system, providing the necessary interface to simulate and analyse the behaviour of physical counterparts in a virtual environment.

On the other hand, the DT Developer is responsible for creating and maintaining key system components. The Developer accesses the Repository to perform push/pull requests for their developments, thereby maintaining the application catalogue that the DT User interacts with. They are also the creators of the DT Application and can develop thematic libraries and modules within the Thematic Modules container. They configure workflows in the Workflow Tools container and can also create and configure the ML Model. Additionally, they may interact with the OS and Execution Framework for specific configuration tasks or to facilitate particular computational operations, reflecting their broader role within the system.

Key components of the DTE, like the Thematic Modules, Workflow Tools, and ML Model, interact seamlessly to ensure the DT Applications function smoothly. Thematic Modules supplement the DT Application with specialised functionalities, enabling it to serve a



broad range of use cases. Meanwhile, Workflow Tools facilitate the management of various operational sequences within the DTE, assisting the Thematic Modules and interacting with the ML Model and DT Support for model monitoring and validation services.

The ML Model is a critical component that leverages computational resources from the Computing GPU container to execute machine learning operations. It is also connected to the ML Retraining container, which houses functionalities for model retraining, thus ensuring that the model stays up to date with the latest data and maintains high performance. This retraining can be event-based, a process facilitated by the Event-Based Retraining container. The ML Language container supplements the ML Model by providing the appropriate programming language for model development and execution.

The OS and Execution Framework acts as a bridge for data flow and system operations. It facilitates the DT Developer's interactions with various system components and enables data from the Databases and Storage Input containers to reach the other parts of the system.

This diagram represents the initial interpretation of the DTE's architecture and will be updated as the project evolves and as more data and requirements are gathered from project partners. It serves as a fluid representation of the system's architecture, adaptable to changes in project needs and technological advances.

This iterative approach ensures that our DTE remains flexible and scalable, ready to meet the evolving project requirements and harness future technological advancements.

4.5 DT applications

A DT application is a specific implementation of a Digital Twin, i.e., a digital replica of living or non-living physical entity. The DT applications are the consumers of the capabilities offered by the interTwin DTE and introduce the use case specific requirements analysed in **Chapter 2**. The reference use cases drive the technical roadmap of the interTwin.

The coordination and obtaining of the technical requirements necessary for the development of the case studies has been carried out in Work Package 4 - Technical co-design and validation with research communities with the seven selected use cases as described in **Chapter 2**.

4.6 DTE Thematic Modules

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

D3.1 Blueprint architecture, functional specifications, and requirements analysis first version

One of the key strengths of the interTwin DTE Thematic modules is their versatility. They can be applied to a wide range of scientific disciplines, including high energy physics, radio astronomy, astroparticle physics, climate research, and environmental monitoring. They are designed to be easy to use and flexible, making them an ideal tool for researchers in many different fields. Examples of how the interTwin DTE Thematic modules can be used in these fields are shown below.

In high energy physics, the interTwin DTE Thematic modules can be used to analyse data from particle accelerators and related experimental devices. They can be used to 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 interTwin DTE Thematic modules can be used to analyse data from radio telescopes. They can be used to identify patterns and correlations in the data, which can help researchers understand the properties of celestial objects and the structure of the universe.

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.

In climate research, the interTwin DTE Thematic modules can be used to analyse data from weather and climate models. They can be used to 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 be used to analyse data from environmental sensors. They can be used to identify patterns and correlations in the data, which can help researchers understand the state of the environment and predict potential environmental hazards.

Additionally, the project has already selected a few key modules that will be necessary for the DT applications to function properly.

Lattice QCD simulations and data management uses 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 the generation of Lattice QCD configurations.
- Module to include Quantum enhancement of the ML-configuration generation algorithms. These modules will be tested in the available Quantum Computing testbeds of the consortium.

Noise simulation for radio astronomy uses a module of ML methods for analysing time series (RNNs, LSTMs, neural ODEs) 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 GAN-based model and optimization 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.

4.7 DTE Core Modules

The DTE Core Modules (i.e., data- and compute-intensive capabilities) offer a range of capabilities to facilitate the creation and operation of DT applications. These modules provide advanced workflow composition, data fusion, real-time acquisition and data analytics, AI workflow and method lifecycle management, and validation, verification, and uncertainty tracing for model quality.

Advanced workflow composition allows for the execution of DT workflows that can invoke other Core module capabilities as illustrated with connections in **Figure 14**. An interdisciplinary processing graph serves as a link and API for the supported workflow engines and is used by various DT applications, ensuring a common user experience, and facilitating the integration of discipline-specific tools. Additionally, data fusion capabilities are provided to describe data with common metadata, enabling data from diverse observational systems and models to be analysed in combined workflows.

Real-time acquisition and data analytics deliver high-performance data ingestion by applying the serverless computing paradigm to DTs. This module expands the advanced workflow composition module to trigger on-the-fly processing upon data acquisition. Data processing tools are integrated with workflow management and operate at low- and high-scale with distributed and/or replicated storage for supporting higher-level data stores and data consumption services, on-line analytical processing for 3D datacubes, and general-purpose batch processing.

Data Fusion implements and integrates processes for merging datasets from different sources, including linking and visualisation of observational and modelled data, and the harmonisation of different types of observational data, such as gridded datasets with vector-based datasets like point streams of data from ground stations.

AI workflow and method lifecycle management is a customizable toolkit that enables complex AI setups. It offers capabilities such as model training, model evaluation, distributed training framework, distributed hyperspace optimization, a portfolio of 'base' models and tailored models, and support for data/label assimilation in model creation and updates.

Finally, validation, verification, and uncertainty tracing for model quality is a toolkit that provides developers with the possibility to design and validate DT models. It supports modelling, simulation, data analysis and uncertainty quantification phases to guarantee quality and reliability of the DT applications outputs. The toolset is meant to be offered as "Model Validation as a Service," enabling customizations of best practices and standard quality measures for scientific disciplines and applications. It applies DevOps practices to automate the process.

4.8 DTE infrastructure

The DTE infrastructure provides the federated data and compute resources involved in modelling and simulation tasks. Key functionalities across use case applications and components include:

- Federated data management federates existing and future data infrastructures by building a cloud of data services, referred to as data lake, that is interoperable with various repository types and include data transfer, caching tools and cataloguing systems. Through this module the DTE can manage HPC data ingress/egress, data transfer and use an event-based interface for event-driven analysis frameworks.
- Federated Compute Infrastructure delivers on-demand processing capacity and federated HTC, HPC and Quantum Computing resources that are accessible at pan-European level through cloud interfaces that support both batch workflow execution and interactive analysis.
- Orchestration executes complex tasks by matching the best storage and computing resources to the relevant parts of a workflow. Based on static configurations (site description, network connections, cost, etc.) and dynamic data collected in previous runs, it can choose data sources and compute resources and orchestrate data transfer tasks before launching the analysis.



D3.1 Blueprint architecture, functional specifications, and requirements analysis first version

- Federation services and Authentication & Authorisation Infrastructure (AAI) provide the ability to monitor utilisation of resources and data across multiple suppliers. Federated AAI permits the use of existing institutional credentials and other forms of user identification when accessing internal and external services of the DTE in an interoperable manner. Policies govern access channels and trust and identity management in the distributed multi supply infrastructure.



5 Conclusion

The interTwin Digital Twin Engine (DTE) blueprint architecture has been designed and analysed in the initial nine months of the project, with a keen focus on the requirements derived from the specific Digital Twin (DT) applications. The development and evolution of the architecture were guided by the C4 methodology, providing clear insights into the Containers, Components, Connectors, and Code that make up the DTE.

The DTE is an open-source integrated platform capable of incorporating application-specific Digital Twins. Its design is centred around the principles of standard-based integration, extensibility, modularity, scalability, and sustainability. It offers two primary levels of service - Platform-as-a-Service (PaaS) for DT developers and Software-as-a-Service (SaaS) for DT users. This allows it to cater to a wide range of scientific domains, industries, and user groups.

DT developers utilise the PaaS capabilities of the interTwin platform to create DT applications and thematic modules that offer new domain-specific tools. These applications are then translated into workflows and made accessible to DT Users via the SaaS layer.

DT users, typically scientists, can select a pre-existing DT application and modify it to their specific use case, if required. This interaction is facilitated by the user interface provided by the specific DT application. Additionally, an Authentication and Authorization Infrastructure (AAI) will be implemented as an integral part of the DTE infrastructure to ensure the security and integrity of the digital twins, offering features such as data protection, trust, and compliance, and securing interconnected systems.

The interTwin DTE blueprint architecture represents an initial interpretation of the system's structure, which is subject to continuous updates and iterations throughout the project. As more data and requirements are gathered from the project partners, the architecture is expected to evolve, making it adaptable to changes in project needs and capable of incorporating future technological advancements. This iterative approach ensures the system remains flexible and scalable, poised to meet the evolving project requirements and leverage future technological advancements.

Ongoing efforts will be put into understanding and implementing actions that will benefit both the blueprint definition and the related implementation. This includes an analysis of interTwin-related initiatives/projects that will influence the DTE architecture and development as part of the project's WP3 and particularly T3.1 task.



6 References

Reference	
No	Description / Link
R1	Diego Ciangottini, Paul Millar, Liam Atherton, Marica Antonacci, Daniele Spiga, Andrea Manzi, Renato Santana, David Kelsey, Adrian Coventry, & Shiraz Memon. (2023). D5.1 First Architecture design and Implementation Plan (V1 Under EC review). Zenodo. DOI: https://doi.org/10.5281/zenodo.8036983
R2	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). interTwin D6.1 Report on requirements and core modules definition (V1 Under EC review). Zenodo. DOI: https://doi.org/10.5281/zenodo.8036987
R3	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). interTwin D7.1 Report on requirements and thematic modules definition for the environment domain (V1 Under EC review). Zenodo. DOI: https://doi.org/10.5281/zenodo.8036991
R4	Kalliopi Tsolaki, Sofia Vallecorsa, David Rousseau, Isabel Campos, Yurii Pidopryhora, Sara Vallero, Alberto Gennai, & Massimiliano Razzano. (2023). interTwin D7.2 Report on requirements and thematic modules definition for the physics domain first version (V1 Under EC review). Zenodo. DOI: https://doi.org/10.5281/zenodo.8036997
R5	C4 model in a Software Engineering subject to ease the comprehension of UML and the software. A. Vázquez-Ingelmo, A. García-Holgado and F. J. García-Peñalvo, 2020 IEEE Global Engineering Education Conference (EDUCON), Porto, Portugal, 2020. DOI: 10.1109/EDUCON45650.2020.9125335

