D3.7 Report on software architecture concepts based on DestinE and interTwin

Status: Under EC Review

Dissemination Level: public

Funded by the European Union

Abstract	
Key Words	Interoperability, software architecture, Digital Twin engine

The document describes the work that has been done to work towards interoperability between DestinE and interTwin through their respective Digital Twin engine (DTE) software components, through the implementation of specific use-case and pilot studies. It includes results and lessons learned with respect to interoperability between the two DTE implementations and an outlook towards future developments and improvements. The document can serve as a guideline for other communities that could be onboarded in DestinE, complementing other initiatives towards evolving and expanding the Digital Twin ecosystem in DestinE.

Document Description

Report on software architecture concepts based on DestinE and interTwin

Work Package number 3

Document type	Deliverable		
Document status	Under EC Review	**Version**	1.0
Dissemination Level	Public		
Copyright Status	This material by Parties of the interTwin Consortium is licensed under a Creative Commons Attribution 4.0 International License.		
Lead Partner	ECMWF		
Document link	https://documents.egi.eu/document/3936		
DOI	https://zenodo.org/records/16966835		
Author(s)	• Thomas Geenen (ECMWF) • Adam Warde (ECMWF) • Paul Millar (DESY) • Björn Backeberg (DELTARES) • Raul Bardaji (EGI Foundation) • Muhammad Arabi Tayyab (Uni Trento)		
Reviewers	• Christian Pagé (CERFACS) • Daniele Spiga (INFN)		
Moderated by:	• Andrea Anzanello (EGI)		
Approved by	Andrea Manzi (EGI) on behalf of the interTwin TCB		

Revision History

Version	Date	Description	Contributors
V0.1	08/04/2025	ToC	Andrea Manzi (EGI Foundation), Thomas Geenen (ECMWF)
V0.2	09/07/2025	Main contributions collected and a first version created	Thomas Geenen (ECMWF), Adam Ward (ECMWF), Paul Millar (DESY), Bjorn Backeberg (DELTARES), Raul Bardaj, Sandro Fiore,(UNITN) Muhammad Arabi Tayyab (UNITN)
V0.3	21/08/2025	Reviewed version	Daniele Spiga (INFN), Christian Pagé (CERFACS)
V0.4	26/08/2025	TCB review	Andrea Manzi(EGI Foundation)
V0.5	27/08/2025	Version ready for QA	Thomas Geenen (ECMWF)
V1.0	28/08/2025	**Final**	

Terminology / Acronyms

Term/Acronym	Definition
DT	Digital Twin
DTE	Digital Twin Engine

Terminology / Acronyms:

- **https://confluence.egi.eu/display/EGIG**
- **https://destine.ecmwf.int/glossary/**

Table of Contents

List of Figures

Executive summary

Destination Earth (DestinE) and interTwin both aim to provide full Digital Twin systems, but they are distinctly different in the domains they target and the users and usage they support. DestinE aims to evolve towards an operational earth system Digital Twin system with a focus on specific impact sectors to support policymakers towards adaptation policies. interTwin aims to develop an environment where Digital Twin developers can quickly prototype and design Digital Twins for science in any scientific domain. Where these objectives converge, for instance when new scientific Digital Twin developments can be relevant for DestinE or when DestinE produces information that can be used by scientific Digital Twins, there is an opportunity for delivering added value when DestinE and interTwin reach a certain level of interoperability. The approach we embraced towards reaching this level of interoperability is through a selected set of use-cases where we recognize that we do not aim for the design of a generic interoperability framework but instead increase interoperability for selected pilot implementations. The main results of the work done in interTwin on interoperability with DestinE has been the demonstration of the deployment of a shared use-case between the two DTE implementations of DestinE and interTwin, where we worked towards the successful deployment of the same use-case in both environments and the implementation of the interoperability between the two different data lake concepts and specific implementations. The latter is more generic in nature but was implemented in support of the same use-case. We also report on the implementation of a pilot study for SIMPL in interTwin where we must mention explicitly that this was not implemented or tested in the DestinE system but only in the interTwin context. We do recognize that SIMPL could open further avenues to make Digital Twin systems more interoperable and therefore added this study to this deliverable.

1 Introduction

1.1 Scope

This deliverable gives an overview of the interoperability pilots that have been conducted to support initial efforts to connect the DestinE DTE components with relevant interTwin DTE components. The main focus has been on data handling component interfaces to support relevant use-cases in interTwin and DestinE. We did not explore, in the context of the interTwin project and this deliverable, deeper model integration between DestinE and interTwin Digital Twins or Digital Twin components, for instance through direct model coupling. Rather the focus has been on identifying interfaces for DestinE data consumption by interTwin use-cases and integration with the interTwin data lake.

1.2 Document Structure

This document is structured in the following way. **Section 2** gives an overview of the relevant DestinE and interTwin system components and architecture concepts with a focus on the DTE component. **Section 3** compares the DestinE and interTwin DTE architectures with the aim to identify potential interfaces for interoperability. **Section 4** gives an overview of the technical pilots. **Section 5** draws a summary.

2 Software architecture concepts based on DestinE and interTwin

In this section we introduce the main components of DestinE and interTwin and relevant subcomponents that we explored for interoperability with interTwin.

2.1 Destination Earth

Destination Earth or DestinE is a European Commission-funded initiative to develop a digital replica, or Digital Twin of our planet by 2030. This groundbreaking endeavor will facilitate greater understanding of our changing climate, as well as dangerous, extreme weather events. The development of DestinE will significantly help to support the European Commission's Green Deal, Data Strategy and Digital Strategy, complementing existing national and European efforts.

DestinE will create a step change in environmental prediction by operationalising the latest advances in numerical weather prediction and climate projections, digital technologies, supercomputers and artificial intelligence. To achieve this goal, three major European organizations, the European Centre for Medium-Range Weather Forecasts (ECMWF), European Space Agency (ESA) and the European Organization for the Exploitation of Meteorological Satellites (EUMETSAT), have combined their expertise and competencies. They implement DestinE together with over 100 partners throughout Europe and under the leadership of DG CNECT. Each entity is responsible for delivering a key component of the system; the Earth-System Digital Twins and the Digital Twin Engine are implemented by ECMWF, the Core Service Platform by ESA and the data lake by EUMETSAT.

DestinE is implemented through a strategic partnership with the European High-Performance Computing Joint Undertaking (EuroHPC JU) that provides access to its pre-exascale supercomputers.

2.2 DestinE main Components

The main components of DestinE are the Digital Twins of the Earth system, the Digital Twin Engine (both developed by ECMWF), the Core Service Platform (ESA) and the data lake (EUMETSAT)

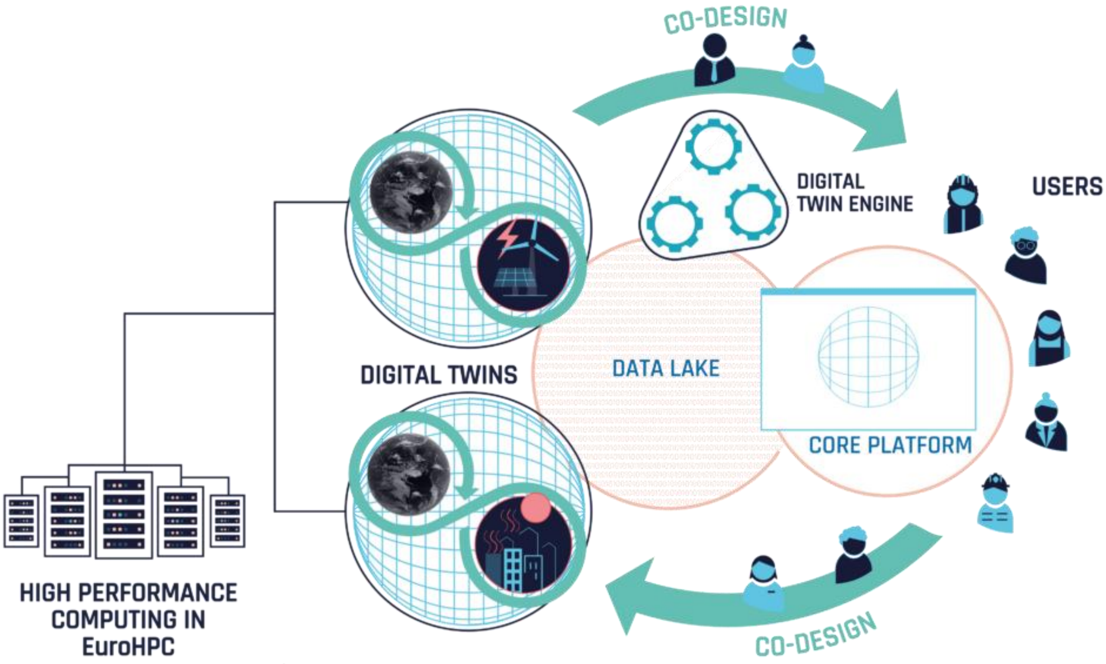

Figure 1 A graphical depiction of the relation between the main Components that together make up DestinE

DestinE Digital Twin Engine (DestinE DTE) [1]: Software-defined environment to operate DestinE's DTs and manage their corresponding control and data flows across distributed HPC and cloud computing resources. It provides a common system approach to a unified orchestration of Earth-system Digital Twins. Moreover, it creates a framework for the fusion of observations with Earth-system simulations and the integration of applications targeting specific impact sectors via selected use cases. The DT Engine enables the porting and optimization of codes, developing and managing the Digital Twin workflows, and provides the data handling and model interaction and interactivity capabilities that run on diverse HPC and cloud infrastructures including the Data Warehouse hosted on the data bridges. The DT Engine concept focuses on interoperability and interactivity and delivers the expected access agility and performance for the Digital Twins and associated data access.

DestinE data lake (DEDL) [2]: the DEDL provides discovery, access, and big data processing services according to a defined and evolving tailored to user needs DestinE data portfolio offered to DestinE users, including required data storage. It provides seamless access to datasets via GUIs or APIs to data in accordance with the DestinE Data Portfolio, regardless of data type and location. The DEDL big data processing allows near-data processing and by this conceptually supports ML/AI applications executed on the DEDL. The DestinE data lake federates with existing data holdings as well as complementary data from diverse sources like in-situ, socio-economic, or data-space data.

DestinE Core Service Platform (DESP) [3]: A user-friendly platform that provides a large number of users with evidence-based policy and decision-making tools, applications and services, based on an open, flexible, scalable and evolvable secure cloud-based architecture. DESP federates access to users' platforms, European cloud and HPC infrastructures and integrates access to an increasing number of Digital Twins as they become gradually available via related European Commission and, possibly, national efforts. The platform will employ novel digital technologies for providing data analytics, visualization, and Earth-system monitoring, simulation and prediction capabilities to its users. At the same time, it will allow users to customise the platform, integrate their own data and develop their own applications.

2.2.1 DestinE DTE

The architecture of the DTE, as illustrated in **Figure 2**, integrates multiple subcomponents that ensure seamless data flow and control across the system. A detailed description of the DTE is given in [**1,4–6**]

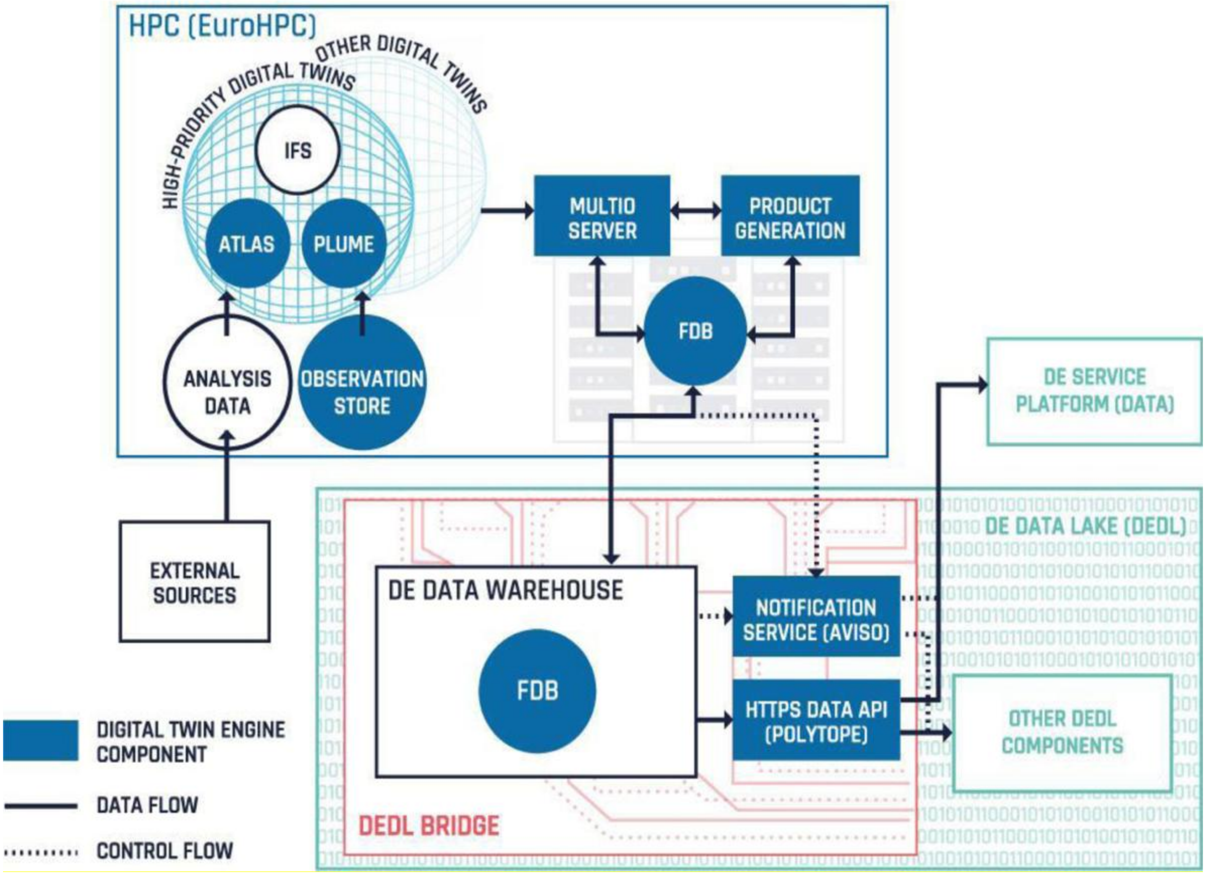

Figure 2 DestinE DTE high-level Architecture

The DestinE DTE data handling components, developed as opt-in modules, continuously evolve to stay aligned with the standards that govern data access and transformation, facilitating seamless interoperability with adapter hooks that are either provided as part of the DTE or developed by contributors to DestinE.

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 data and data handling services remain FAIR. Selected Digital Twin datasets also adhere to directives about the availability of public datasets.

DestinE through DTE implementations will contribute to developing and maintaining efficient data access methods. This includes providing hooks for connectors relevant to the community, DestinE relevant impacts sectors and selected use-cases. These implementations support interoperability with other tools (e.g., Climate Data Operators, Climate Data Store Toolbox [7]), community software platforms (e.g., Pangeo [8]), and infrastructure systems (Wekeo [9], European Weather Cloud [10], etc.) and in the context of this project, interTwin data handling services and the interTwin data lake.

In line with ECMWF's 2022 software strategy [11], all DTE data handling and processing components are openly developed. This encourages direct interaction with the community and supports and enables interoperability between standards, data formats, and APIs. Furthermore, DTE component development aims to leverage existing community software stacks and contribute back to these initiatives through open-source collaboration.

2.2.1.1 DestinE DTE Data access interfaces

The primary native data access interface provided by the DestinE DTE is Polytope [12]. Polytope acts as an interface for outputs from both the Extremes DT and the Climate DT stored in a semantically addressable key-value object storage (FDB) [13]. Polytope also provides feature extraction allowing users to request time series, vertical profiles, and arbitrary polygon cutouts from output data, rather than retrieving entire global fields in their raw format. This significantly reduces the amount of data transferred and speeds up access times to data. It also allows for server-side post-processing.

Polytope allows federated access to DestinE data from any of the EuroHPC systems running the Digital Twins, simplifying access for users and services.

An illustration of the usage of polytope for specific DestinE use-case pilot is presented in the section on the Deltares FloodAdapt use-case. There data is pulled from the Climate DT via polytope and used in the FloodAdapt application. Deltares could further use feature extraction to improve the performance of their application by lowering the amount of data retrieved and only targeting the area where data is required rather than pulling global data fields and subsequently performing post-processing on that data to extract the relevant area.

Polytope feature extraction calls return data in the CoverageJSON [14] format. CoverageJSON is an OGC community standard widely used in geospatial and meteorological communities. This allows further interoperability with non-climate and non-meteorological communities and integration with other open-source tools that can parse CoverageJSON.

CoverageJSON is a cloud-ready, compact and human readable format that contains several different feature types that correspond well to features from polytope, such as point series, paths, and vertical profiles. These features can then be interpreted by tools provided by ECMWF and other open-source tools for out of the box processing and visualisation.

As well as providing standard output formats for DestinE data, standard interfaces are also being developed in the context of DestinE and the DTE. A STAC (Spatiotemporal Asset Catalogs) [15] catalogue is provided for data from the Extremes and Climate DTs allowing users to interact and generate requests from this STAC catalogue.

The DestinE DTE implements the EDR (Environmental Data Retrieval) [16] interface in addition to the native polytope API to access DestinE data via polytope, allowing to leverage of the efficiency of polytope while using a more generic interface. EDR is an OGC standard and hence allows for standardised requests to geospatial data, it aligns with polytopes existing request features and the OGC CoverageJSON output format. This will allow users to easily integrate DestinE data into their existing workflows that already use EDR.

As EDR is a more generic standardised interface than the native polytope interface it could be easily integrated with openEO [17], a framework evolved in the interTwin project and used in several interTwin use-cases. This could be achieved by adding the Polytope EDR service as a backend to openEO. In this case not only DestinE Digital Twin data could be served but also other data sources such as the Copernicus Climate Data Store and other federated data sources from the DestinE data lake. These collections can easily be added to the EDR service and then served via openEO once it is added as a backend.

Polytope is also integrated with the Eumetsat-developed Harmonized data access (HDA) service that is deployed in the DestinE data lake, allowing users to access Digital Twin data using the HDA via polytope. This integrates access between the Extremes and Climate Dt data and the many federated datasets provided via the DestinE data lake. The HDA API is used for the data lake interoperability study, **section 4.1**.

2.2.2 DestinE data lake

The DestinE Data Lake (DEDL) provides discovery, access, and big data processing services. The DEDL big data processing allows near-data processing and by this conceptually supports ML/AI applications executed on the DEDL. The DestinE data lake federates with existing data holdings as well as with complementary data from diverse sources like in-situ, socio-economic, or data-space data, **Figure 3**. In the context of the interoperability pilots in interTwin with DestinE, integration with DestinE IAM services has been implemented to access DestinE data from the interTwin data lake.

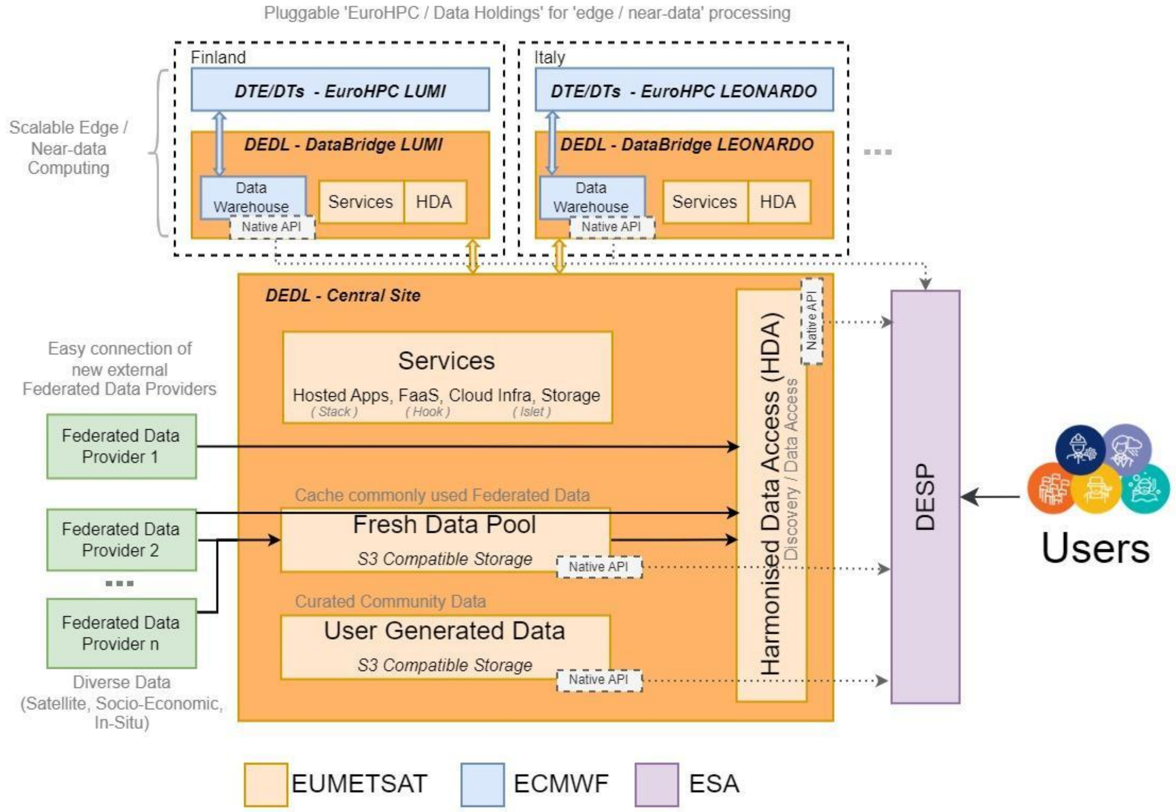

Figure 3 DestinE data lake high-level design [18]

2.2.3 DestinE Core Service Platform

ESA is responsible for developing the platform, which serves as a single access point for users of the DestinE ecosystem. The DESP integrates and operates an open ecosystem of services (also called DESP Framework) to support DestinE data exploitation and information sharing for the benefit of DestinE users and third-party entities. Figure 4 shows the high-level architecture of DESP opened to users in October 2024.

The DESP Framework includes essential services such as:

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

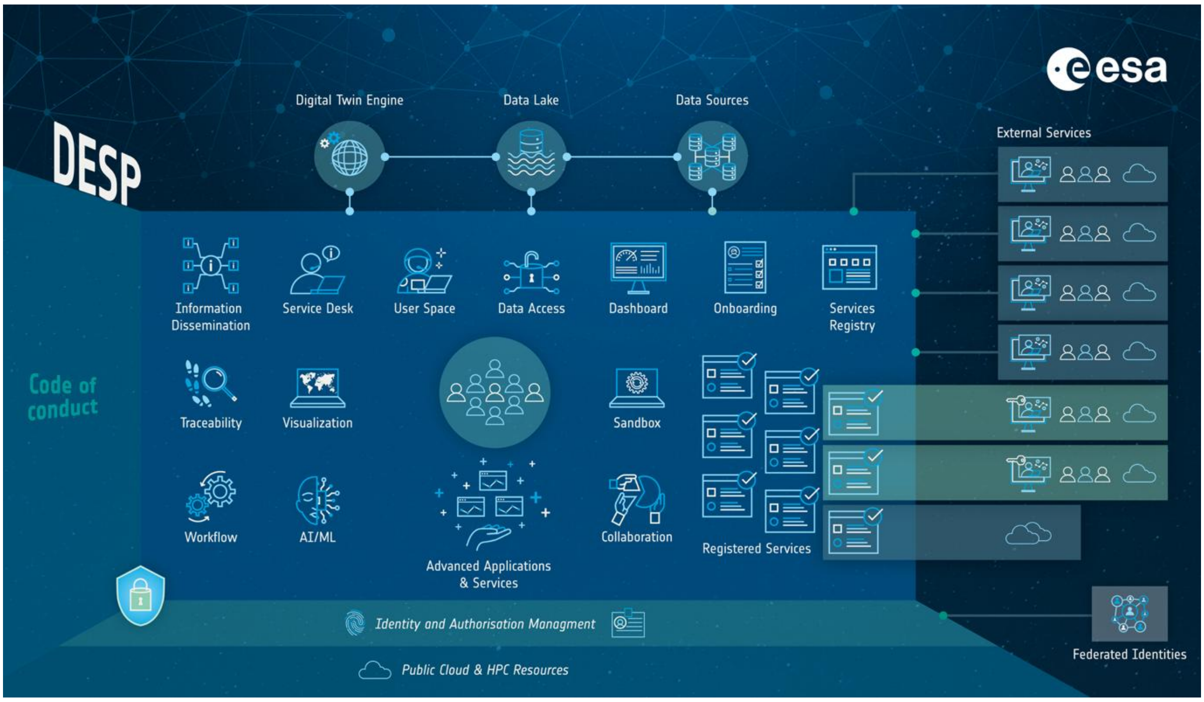

Figure 4 DESP high-level architecture

3 Architectures comparison

This section will compare the high-level DestinE and interTwin DTE architectures with the aim of identifying interfaces that can be exploited for interoperability but also to provide a more general architectural comparison. We found that at a high level the same architectural abstractions can be identified but that there is a key difference in focus between the two. DestinE has the aim to develop a Digital Twin of our planet by 2030, interTwin does not have as a primary goal the development of specific Digital Twins for a particular application domain but aims to provide a research & development environment for DTs or as they call it: An interdisciplinary Digital Twin Engine for science. As part of the project Digital Twins are developed but to develop domain specific (weather, climate, high energy physics and radio astronomy) thematic modules, that provide features, services and functionality in support of DT developments and deployments in these domains, arguably more with a view towards selected technology developments that could fit the DEDL and DESP environment within DestinE. Coupling aspects for DestinE DTE data consumption and interoperability with the workflow management system component of DestinE have been considered as well as interoperability with the interTwin data lake interfaces and tools, section 4.1.

3.1 interTwin Components mapping to DestinE Architecture

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

3.1.1 DestinE C4 Architecture

The diagram in Figure 5, illustrates the high-level C4 [19] system context for the DestinE architecture mapped on a similar design concept in interTwin. It is made up of four main subsystems:

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

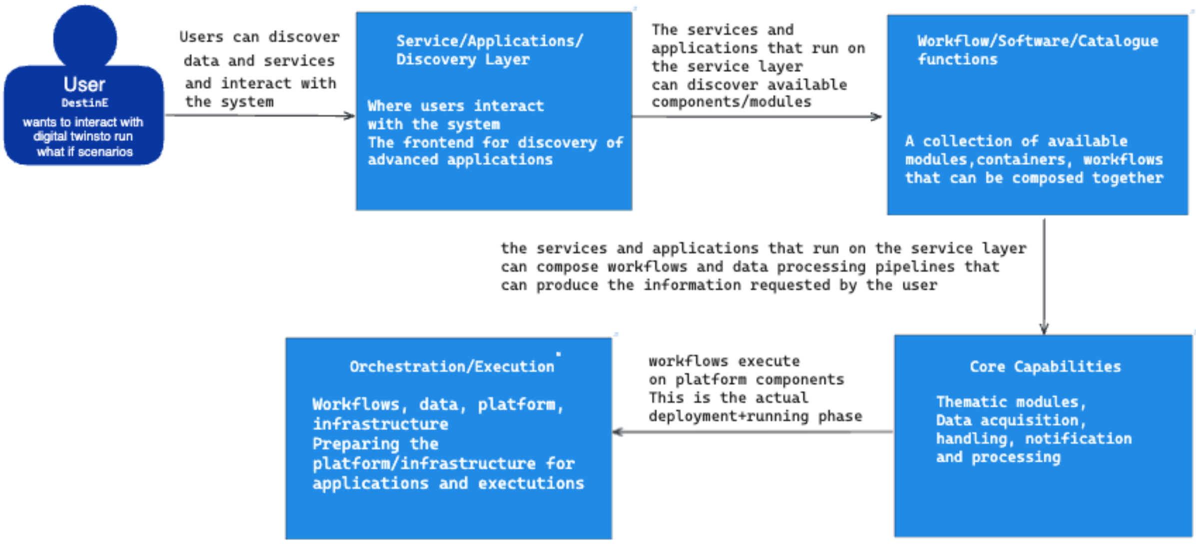

Figure 5 System Diagram for Destination Earth

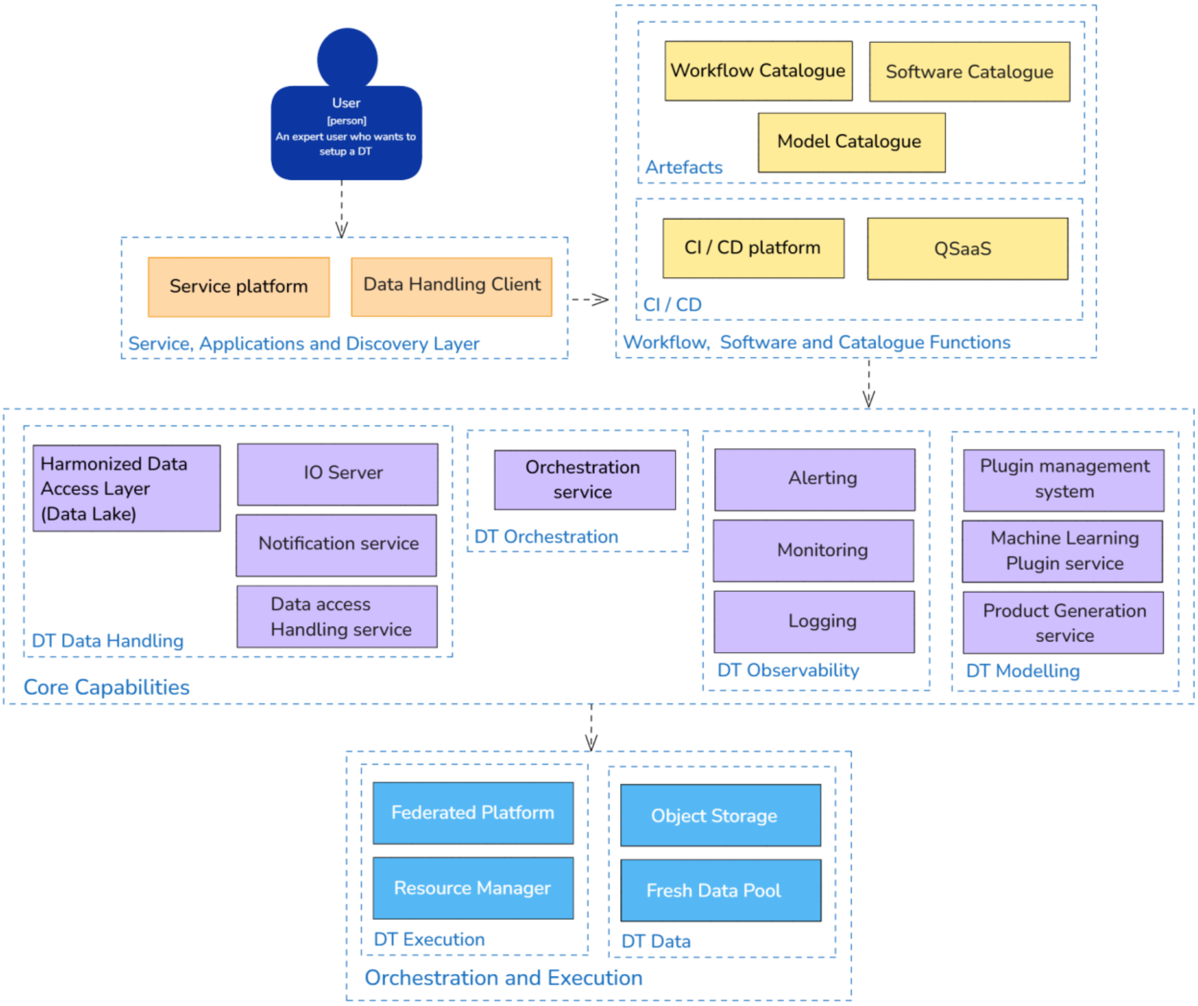

Figure 6 DestinE Container Diagram

Figure 6 Provides a more detailed container diagram of the DestinE implementation following a shared DestinE, interTwin architectural concept. Components that implement these functional capabilities that are already developed:

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

3.1.2 Initial Gap Analysis

To clarify the alignment and differences between interTwin and DestinE architectures, this section provides a mapping of components across both systems. In Figure 7, we used the main structure from Figure 6, which represents the DestinE container diagram, as a base. We then incorporated the corresponding components from interTwin to illustrate where the two architectures align and complement each other. This approach helps to visually map interTwin unique elements within the existing DestinE framework, providing a clearer view of shared functionalities and distinguishing features.

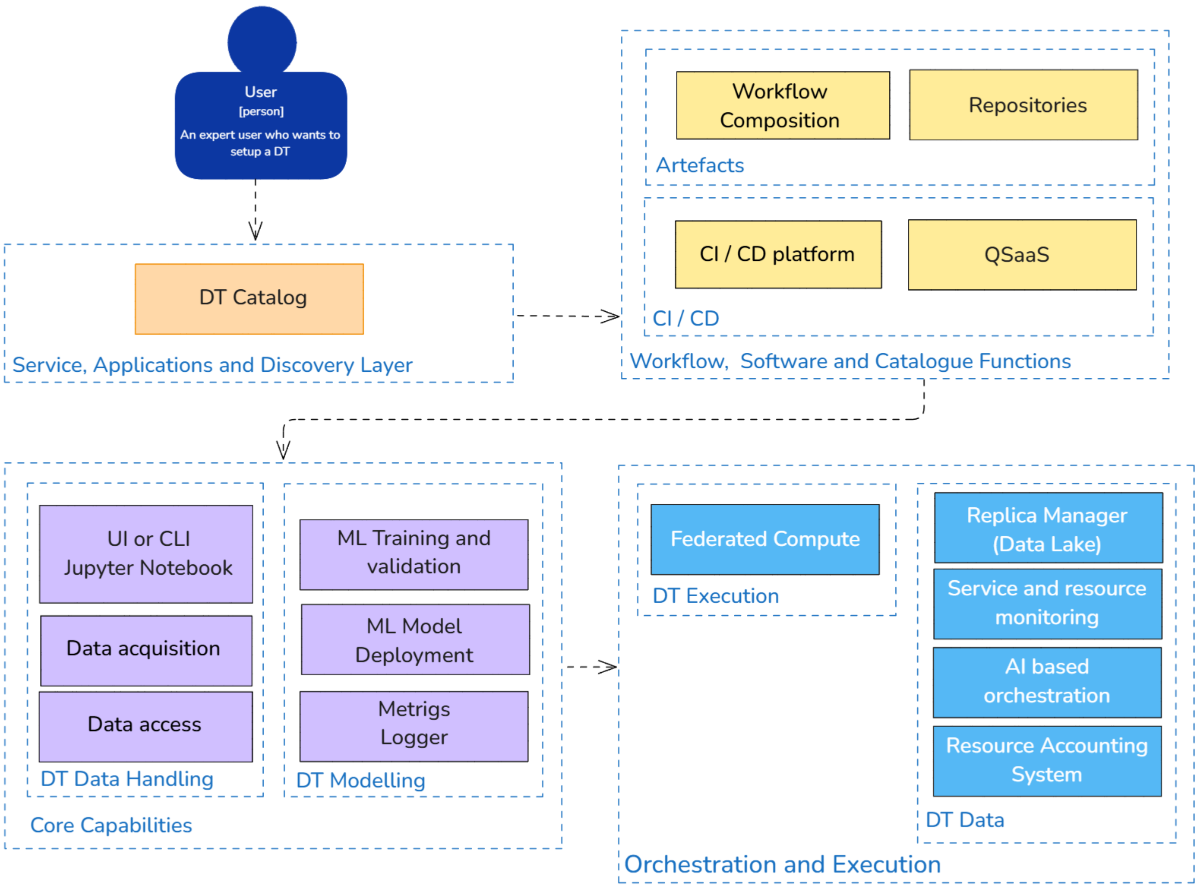

Figure 7 interTwin capabilities mapped to DestinE container diagram

The main layers in both architectures, such as the Service, Applications, and Discovery Layer, Workflow, Software, and Catalogue Functions, Core Capabilities, and Orchestration and Execution, establish a foundation for alignment. However, within each layer, specific components serve unique purposes aligned with each project's goals:

- Service, Applications, and Discovery Layer: interTwin DT catalog provides a streamlined (single) entry point for users to access Digital Twin interfaces, while DestinE Service Platform offers a range of diverse general and domain-specific functionalities, including computational resources tailored for end-users exploring DT scenarios.
- **Workflow, Software, and Catalogue Functions**: Both architectures include infrastructures for repositories and workflow management. interTwin, however, goes further by incorporating continuous integration and software quality assurance, whereas DestinE emphasizes repository management with a focus on continuous integration.
- **Core Capabilities**: Data handling and modelling are core to both architectures. DestinE introduces a thematic Data Harmonization Layer that supports specialized applications, whereas interTwin core capabilities are more generalized, allowing multidisciplinary DT applications across Earth systems.
- **Orchestration and Execution**: interTwin approach includes hybrid resource management, utilizing both cloud and high-performance computing (HPC), and

offloading mechanisms to optimize processing. DestinE incorporates batch systems to handle workflows, offering a more traditional orchestration structure. interTwin also includes an orchestration and deployment component that was not planned for DestinE.

4 Piloting Activities

The section summarizes piloting activities performed in interTwin to show the interoperability at the level of data transfer and access with DestinE. In particular:

- Bulk data transfer from DestinE DEDL using HDA
- DT data access via polytope for a coastal hazard DT
- SIMPL deployment in interTwin

4.1 Bulk data transfer from DestinE DEDL HDA

An exercise was undertaken, in the context of an interoperability pilot for data access, to generate a proof-of-concept workflow that shows data being transferred from Destination Earth's data lake (DEDL) to the interTwin data lake using the DEDL HDA interface [20]. The goal of this pilot was to provide a complete implementation demonstration, to show that user data taken from the DEDL HDA service may be used by interTwin workflows that are running on interTwin resources.

Although the pilot was successful, it also helped identify some improvement points where future developments could provide increased functionality and accelerated adoption. These points are listed at the end of this section.

The interTwin data lake is a federated data management solution that consists of a heterogeneous set of storage providers that, together, provide the data lake's storage capacity. The availability of data stored within the data lake is made possible by one or more replicas of files using the capacity of these storage providers. Each replica stores a complete copy of a file's data. If an additional replica is needed (for example, to provide optimized access for analysis running on nearby compute resources), then an additional replica is created on the targeted storage. This third-party data copying (TPC) is achieved using a centrally run File Transfer Service (FTS) [21] that optimizes shared resources (such as network bandwidth) to maximize throughput. The catalogue of available replicas is stored in another centrally run service called Rucio [22]; this service also organizes data into datasets (collections of files) and containers (collections of datasets and containers) to support managing large numbers of files. Rucio is also responsible for orchestrating replicas and deleting data that is no longer needed or if it needs to reclaim the capacity for storing new replicas. This technology stack has provided a proven operational solution since 2018 for managing petabytes of data comprised of millions of files.

To demonstrate the ability to transfer data from the DEDL infrastructure, files were chosen using the DEDL STAC [15] catalogue, although the process would have been the same for any user file that was not registered in the DEDL STAC catalogue.

The DEDL storage requires authentication before granting access to a file. This is true even for data that is generally accessible, i.e., data that any user with an account can access. The authentication scheme requires presenting an OAuth2 [23] access token when making the HTTP GET request, using the standard Authorisation header. This approach to authentication is broadly the same as with the interTwin data lake; however, the OAuth2 access tokens are issued by different OpenID providers (OPs): for DEDL storage, the tokens are issued by the DEDL OP and for the interTwin storage, the tokens are issued by the EGI Check-in (demo) service [24].

The DEDL provides a python script that obtains a DEDL-OP-issued access token from the user's credentials (username and password). This script obtains an access token from the DESP-OP and uses this token to authenticate against the DEDL-OP to obtain the desired DEDL-OP-issued access token. This last step produces both an access token and a refresh token. The latter is ignored by the python script, but this was adjusted and used to populate an oidc-agent account. Creating such an oidc-agent account allows for generating DEDL-OP-issued access tokens on demand, without requiring the user to present their credentials.

Having the ability to generate a DEDL-OP-issued access token, a client can request a file's data. For optimization, the data transfer should be site-to-site, without the client first downloading the data and subsequently uploading it. The HTTP-TPC protocol [25] was used to achieve this. The protocol is triggered by issuing an HTTP COPY request that targets the destination (some storage within the interTwin data lake) with instructions to fetch data from the DEDL storage along with the DEDL-OP-issued access token. Manually triggering transfers using the curl command showed that such transfers are possible.

As a final extension, the transfer was repeated using the interTwin FTS service. A request was submitted to transfer files, using FTS' ability to accept two distinct tokens: a DEDL-OP-issued token for the source storage and an EGI-Check-in token for the destination storage. This request was accepted by the FTS service and handled in the usual fashion: no changes were needed to support the transfers.

Through this, we were able to provide a proof-of-concept demonstration that data may be copied from DEDL storage into interTwin storage. This pilot explicitly did not explore interfaces with the EuroHPC parallel filesystem.

4.1.1 Identified limitations and points for future work

Although FTS was able to transfer the files without issues, there is currently no support in Rucio for generating storage-specific access tokens. Rucio assumes that the same token may be used for all storage. In the interoperability use-case, two distinct tokens are needed when transferring data from DEDL storage to interTwin storage (both tokens are short-lived).

Rucio requires that registered files have known checksums. STAC currently does not provide such checksum information about STAC assets. This prevents easy registration of STAC-registered files without downloading the asset to discover the file's checksum value.

These limitations are being communicated to subsequent projects, to allow these points to be addressed in the future.

4.2 Deltares DT data access via Polytope

4.2.1 Summary of the pilot case Deltares DT data access via Polytope

For the interTwin project, we explored the integration with DestinE from a use-case (Digital Twin) on flood risk mitigation and adaptation. In the DestinE Generic Adaptation Modelling Framework Flood Risk Management project (DE372 Lot 1), Deltares developed a prototype web application for the purpose of demonstrating the Generic Adaptation Modelling Framework for scoping and co-developing an adaptation modelling system, or Digital Twin, for flood risk management[1].

The prototype webapp, based on the FloodAdapt frontend and implemented using Solara[2], was designed to simulate flood scenarios, assess economic impacts, and evaluate adaptation measures for decision-makers engaged in climate adaptation planning. It integrates tools such as SFINCS (Super-Fast INundation of CoastS) for rapid flood simulations, Delft-FIAT for economic impact assessments, HydroMT for automated model building and adjustment and the FloodAdapt backend for scenario management, which are all interTwin thematic modules.

SFINCS, Delft-FIAT, FloodAdapt and the solara-based webapp are published under the following repositories:

- **https://sfincs.readthedocs.io/en/latest/index.html**
- **https://deltares.github.io/Delft-FIAT/latest/**
- **https://deltares-research.github.io/FloodAdapt/**
- **https://github.com/destination-earth/DestinE_Deltares_DE372a_FloodRiskDemo**

Figure 8 illustrates the workflow as it is executed on the interTwin DTE. A DT Developer uses Jupyter Notebooks to produce and edit a configuration file which in turn leverages HydroMT and FloodAdapt to set up the SFINCS, WFLOW, Delft-FIAT, and RA2CE models (see hyperlinks above for reference). This model set-up process can be executed iteratively until the user is happy with the quality of the models. The necessary data to set up the models is stored in the DestinE DEDL data lake.

Once the models have been created, a Scientist or DT User defines specific scenarios and runs what-if scenarios through a scenario file, also prepared in Jupyter Notebooks. The Scenario Workflow Execution allows users to iteratively explore and refine various flood risk scenarios using the same modelling components. The workflows are packaged as docker containers and executed via OSCAR [26] services[3]. For computationally intensive

[1] https://arcg.is/1LeDaK1

[2] https://solara.dev/

[3] https://www.interTwin.eu/article/core-dte-module-oscar

workflows, OSCAR offloads these to HPC infrastructures via interLink[4]. The results of the simulations are returned to the JupyterHub instance, where users can visualise and interpret results directly in the Jupyter Notebook.

The prototype flood risk management application developed under DE372 and used in this context to implement and demonstrate interoperability between DestinE and interTwin, only uses the SFINCS, Delft-FIAT, HydroMT and FloodAdapt modules. It is executed as a stand-alone application and is not tightly integrated with the interTwin core modules, OSCAR and interLink.

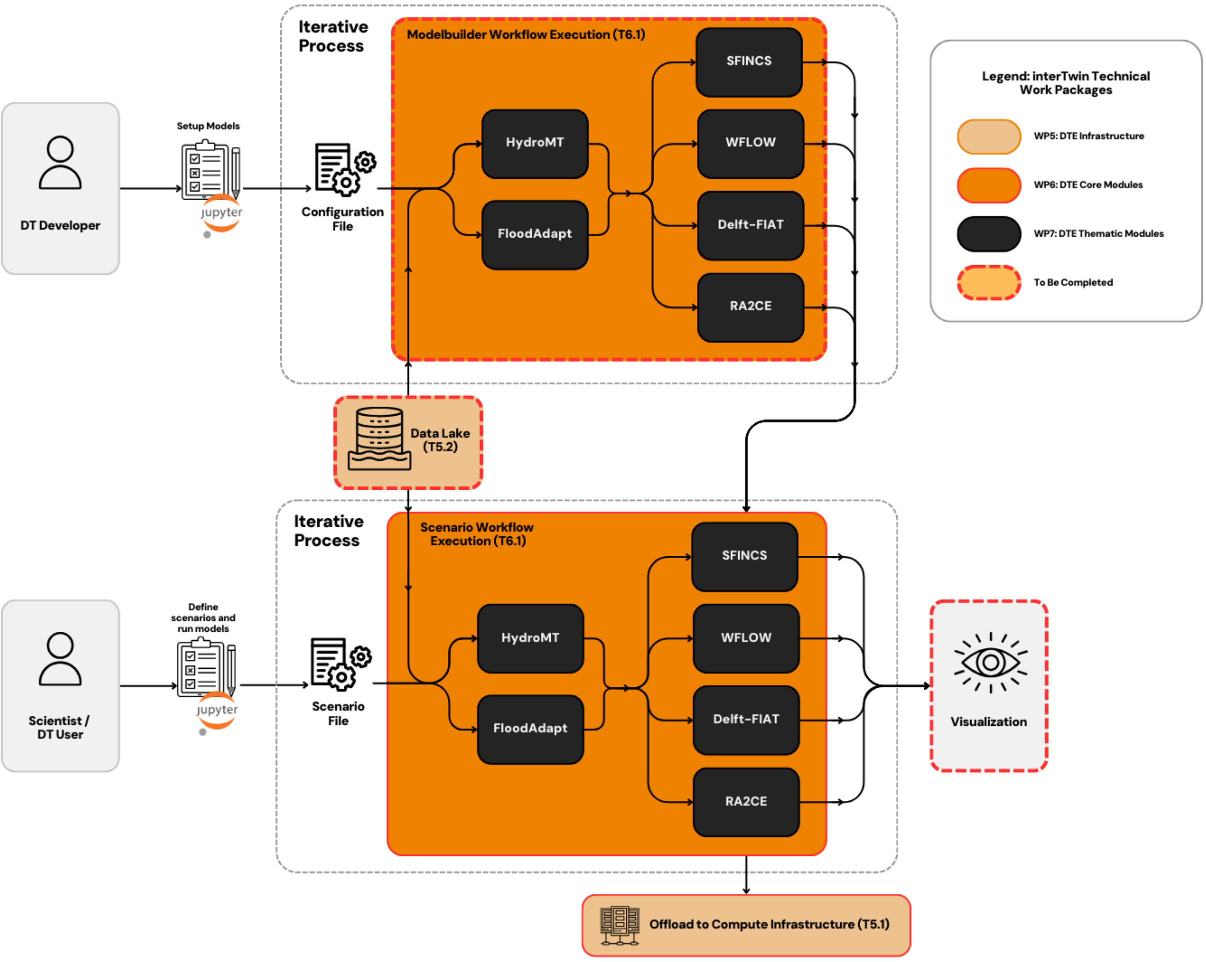

Figure 8 Workflow architecture for a Digital Twin (DT) for flood risk management using the interTwin Digital Twin Engine (DTE)

In the context of the tech transfer and technology engagement activities we have plotted the Floodadapt solution in the DestinE landscape, **Figure 8**.

[4] https://www.interTwin.eu/article/infrastructure-component-interlink

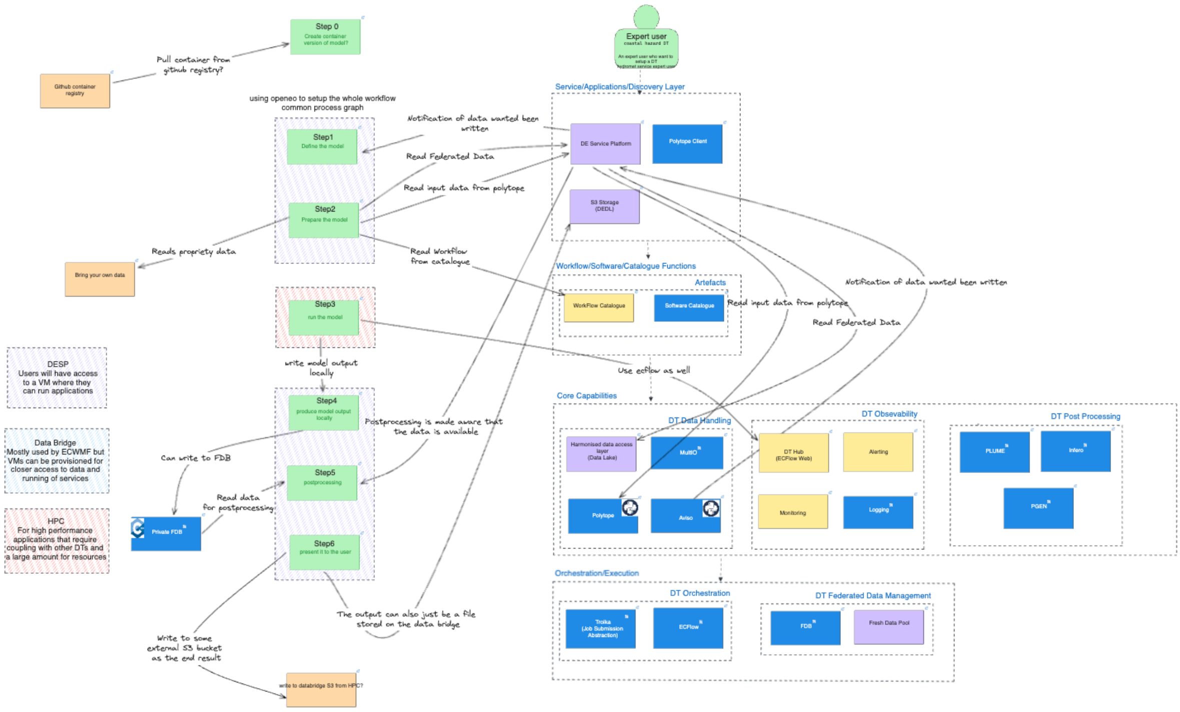

Figure 9 The Floodadapt solution is plotted on the DestinE DTE landscape, using some of the available DestinE DTE services and features

The flood risk management application has been analysed for integration with DestinE and it was found that its main components can be easily deployed with DEDL edge services on the DestinE data bridge infrastructure because they have been containerised to support infrastructure-agnostic deployment.

In terms of data requirements, the Polytope API service is used to download data from DestinE Climate Adaptation Digital Twin (Climate DT). For the data to be used in the flood risk management application it must be interpolated to a regular grid. The downloading and interpolating of the data are time-consuming, adversely affecting the interactivity of the application. As a workaround a Jupyter Notebook is provided with an example of how to download and process ClimateDT data for use with the demonstrator, and users are advised to do this before using the demonstrator to configure and run what-if scenarios.

Presently the Climate DT delivers the following data that are relevant for flood modelling:

- Wind Data: High-resolution wind vectors (U- and V-components at 10m above ground) that are vital for driving offshore wave and surge models.
- Precipitation Data: Total precipitation datasets, which are converted into hourly rates for use in hydrodynamic models.
- Atmospheric Pressure Data: Mean sea level pressure, which is crucial for simulating storm surges and tidal variations.

Note that water levels that are dynamically consistent with the Climate DT (i.e. water levels from a model that forced with Climate DT data) are not yet available, this is being partially addressed in a follow-up contract (DE374 “Global tide and storm surge forecasts for climate resilience and maritime optimisation"), where 4-day forecasts of water levels from the Global Tide and Surge Model[5], forced with DestinE Extremes DT will be provided as a pilot service. This pilot service could be expanded to the Climate DT to provide future projections of water levels.

The data requirements could be satisfied by calls to HDA and from Islet storage services. Specifically, the Earth Data Hub service[6] combined with near data compute resources could be explored as solutions to the time-consuming download and interpolate limitation mentioned previously.

Figure 9 illustrates the main potential integration points. Not only the consumption of DestinE data is foreseen but also tighter integration with the workload manager, via a CWL [27] to ecflow [28] interface. At the same time ideas have been exchanged between Deltares, interTwin and ECMWF on how the interTwin data lake components could be used from the DEDL (**section 4.1**), further using data access components like Polytope, HDA and Earth Data Hub to access DT data but also allow the flood risk management application output data to feed into the DEDL and/or the interTwin data lake. The flood risk management application could be deployed with some of the interTwin modules within edge services provided by the DEDL or directly on the DESP. This could use CWL and/or interface with OSCAR. Future DTE developments will directly support data catalogue STAC interfaces for DestinE DT data. In addition, interfacing with data handling components like Rucio and FTS could be explored further.

In summary, the flood risk management Digital Twin demonstrates a practical and flexible implementation that aligns well with the broader goals of the DestinE and interTwin initiatives. While currently operating as a stand-alone prototype on DestinE, the modular architecture, containerised deployment, and integration with OSCAR could in the future support tighter interoperability. Ongoing and future work should enhance data accessibility, computational scalability, and integration with DestinE services such as Earth Data Hub and ecflow. Enabling broader cross-infrastructure interoperability would support engaging a broader stakeholder community exploring the Digital Twin ecosystem for climate adaptation and flood resilience.

[5] https://publicwiki.deltares.nl/spaces/GTSM/pages/201033074/Global+Tide+and+Surge+Model

[6] https://platform.destine.eu/services/service/earth-data-hub/

4.3 Proof-of-Concept of Data Space Implementation with SIMPL Open, in interTwin

The primary goal of this initiative was to discover and demonstrate a Proof-of-Concept (PoC) for deploying a data space architecture using an open-source software framework called SIMPL Open [29]. The implementation aimed to support secure and interoperable data sharing based on international Data Spaces (IDS) [30] and GAIA-X [31] principles. The PoC was planned to carry out the following activities. First, to understand and assess the architecture and modular components of SIMPL Open. Second, to set up the SIMPL Open software stack on a Linux-based environment using a Kubernetes environment (Minikube). Likewise, deploying necessary services such as ArgoCD , cert-manager , Vault, Redpanda, Kafka, ELK, and other services for data spaces [32–37]. Moreover, to investigate the basic integration, authentication, and orchestration across components within Kubernetes.

4.3.1 SIMPL Open Overview

SIMPL Open is organized as a multi-layered data space architecture which is typically composed of an infrastructure layer, core services layer, data services layer and security & identity layer. The infrastructure layer is the base layer with Kubernetes, Helm, and Minikube. The core services layer includes the main components such as ArgoCD, Vault, Kafka, Redpanda, and ELK stack (Elasticsearch, Logstash, and Kibana) used for data streaming, authentication and observability. The data services layer comprises Kafka and Redpanda for data flow and a PostgreSQL [38] stack with operators for data persistence. Security and identity layer uses cert-manager, HashiCorp Vault for secure secret management. This layered approach offers reusability and flexibility but also entails vigilant configuration and orchestration, mostly in cloud-native environments.

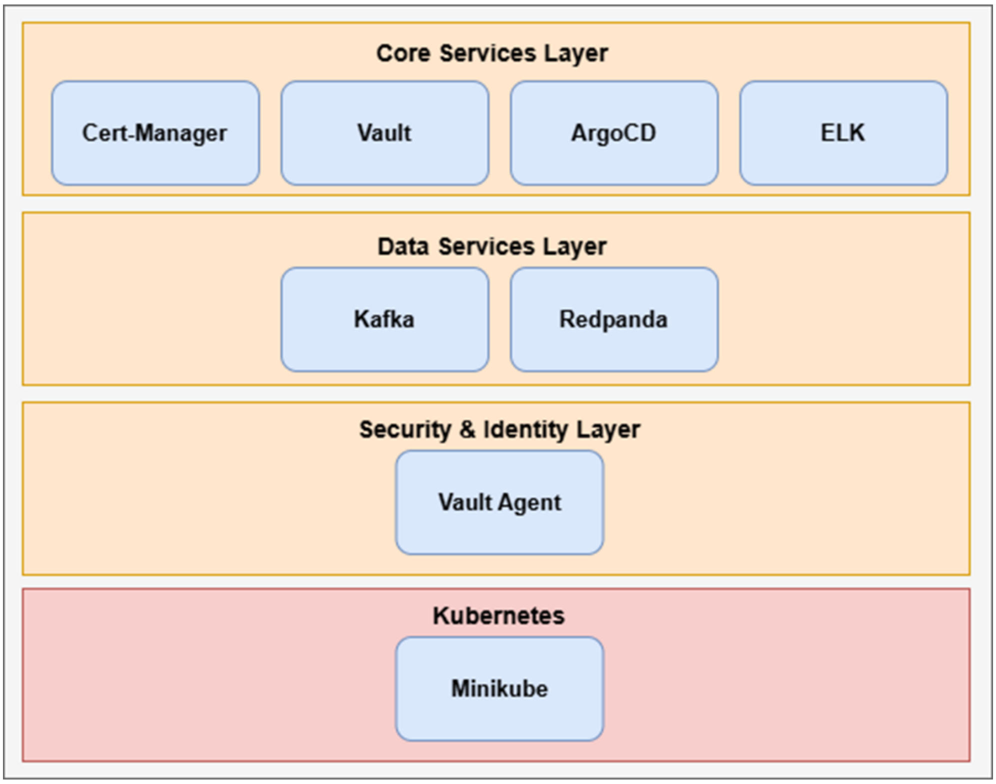

Figure 10 SIMPL Open (core) services running inside Kubernetes cluster

4.3.2 Activities and Key Implementations

In the context of the interTwin project, the Proof-of-Concept (PoC) implementation of data spaces using the SIMPL platform was carried out to examine its potential for building interoperable and secure data environments. The deployment was carried out on a Linux-based environment, with infrastructure resources (such as CPU, memory, storage, etc.) and container orchestration capabilities. Because the SIMPL Open (software stack) has substantial installation requirements, these resources were necessary particularly due to its modular architecture and reliance on multiple services within a Kubernetes environment.

4.3.2.1 ArgoCD Application setup

During the SIMPL Open deployment, we set up ArgoCD to manage and monitor application configurations declaratively through GitOps [39]. Two main applications that were configured and deployed within the common namespace are common and common-deployer. Common-deployer was accountable for provisioning the shared infrastructure components required by other apps while common included core services like Kafka, Vault, and other necessary micro-services as defined by the SIMPL Open control plane. These applications were defined in Git and visualized in the ArgoCD UI as shown in **Figure 11**below.

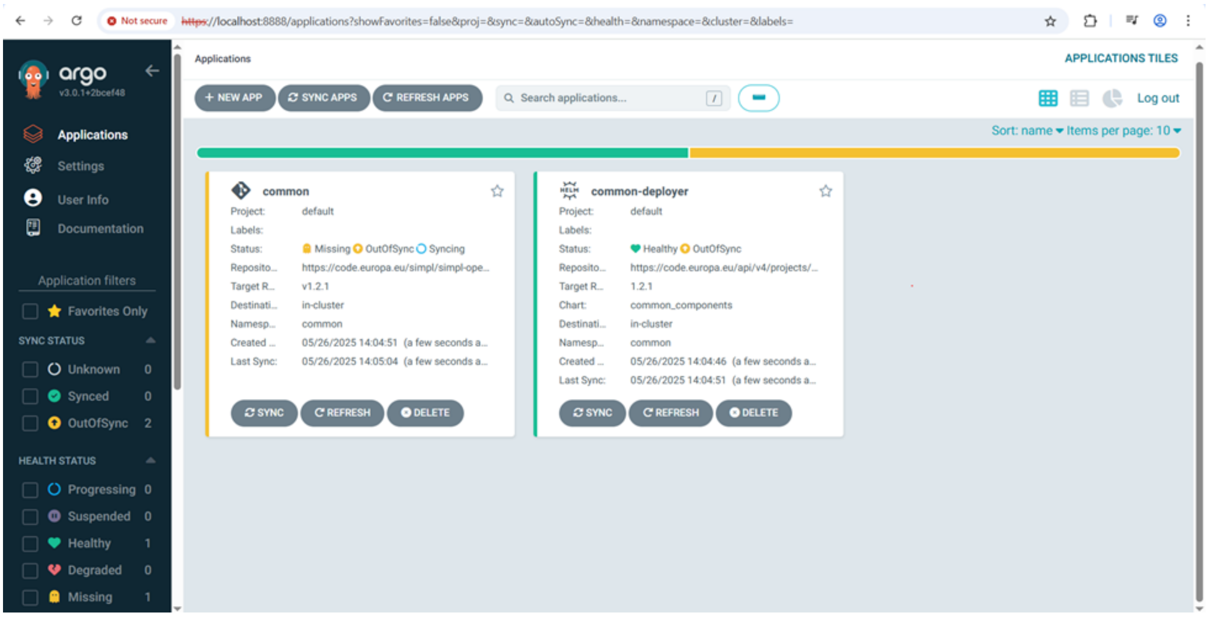

Figure 11 ArgoCD UI snapshot during SIMPL Open PoC deployment, showing the status of the common and common-deployer applications

4.3.2.2 ArgoCD Application Definition: common-deployer

To orchestrate the deployment of core infrastructure services in the common namespace of SIMPL Open, the common-deployer application was defined and registered in ArgoCD. This application was configured to deploy the common-components Helm chart [40] from the SIMPL Open Helm repository hosted on GitLab instance [41].

The configuration included details such as the chart version, namespace targets, cluster issuer, Vault access configuration, Kafka topic settings, and monitoring tools integration. ArgoCD monitored this configuration for changes and enabled GitOps-based synchronization to ensure reproducibility and consistency across deployments.

The following YAML [42] snippet shows the trimmed ArgoCD Application manifest used for common-deployer:

```
apiVersion: argoproj.io/v1alpha1

kind: Application

metadata:

  name: common-deployer

  namespace: argocd

spec:
```

```
  project: default
  source:
repoURL: 'https://code.europa.eu/api/v4/projects/951/packages/helm/stable'
targetRevision: '1.2.1'
    chart: common_components
helm:
            values: |
            values:
            branch: v1.2.1
            project: default
            namespaceTag: common
            domainSuffix: int.simpl-europe.eu
            argocd:
            appname: common
            namespace: argocd
            cluster:
            address: https://kubernetes.default.svc
            namespace: common
            issuer: dev-prod
            kubeStateHost:kube-prometheus-stack-kube-state
metrics.devsecopstools.svc.cluster.local:8080
        hashicorp:
            service:"http://vault-common.common.svc.cluster.local:8200"
            secretEngine: dev-int
```

```
            role: dev-int-role
            monitoring:
            operator:
            managedNamespaces: "common,authority,dataprovider,consumer"
            kafka:
            topic:
            autocreate: true
  destination:
server: 'https://kubernetes.default.svc'
namespace: common
  syncPolicy:
automated:
            prune: false
            selfHeal: false
syncOptions:
- CreateNamespace=true
```

4.3.2.3 Accessing Remote Server

Since the server was headless and accessed remotely, ArgoCD web UI was exposed via a NodePort service on the Ubuntu host [43]. To securely access the UI from a remote client, we configured an SSH tunnel using PuTTY [44], mapping the NodePort (e.g., localhost:8080) to the remote ArgoCD service on the server. This method allowed us to use a local browser to visualize and manage the common and common-deployer ArgoCD applications while maintaining command line interface (CLI) control over the same cluster via the PuTTY shell interface.

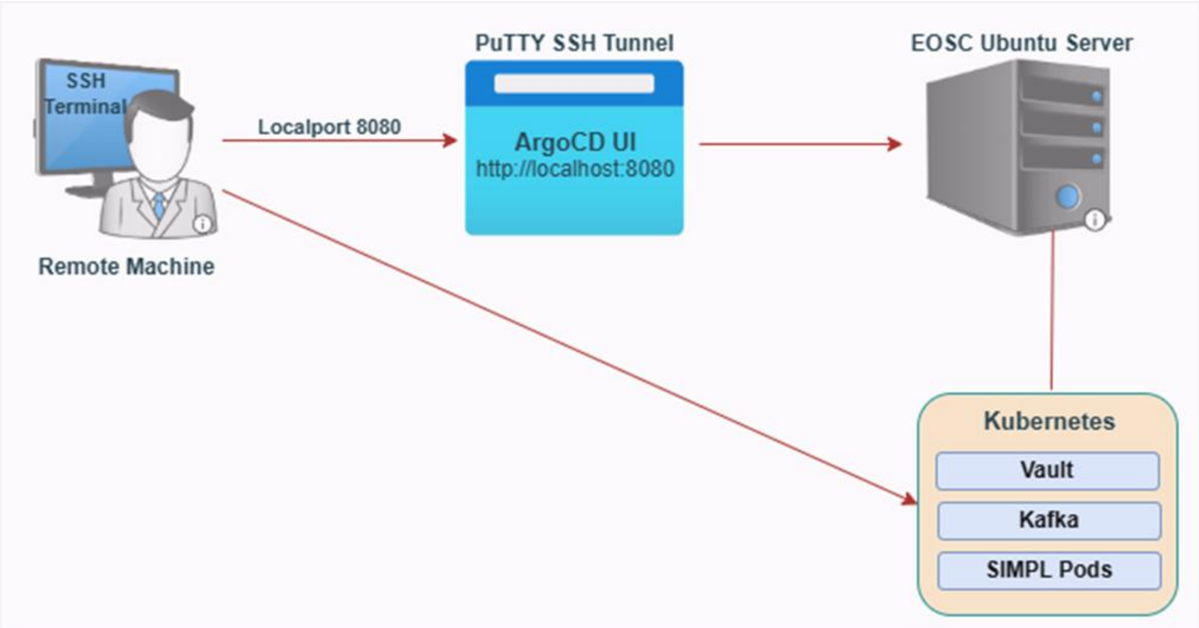

Figure 12 Secure remote access setup for SIMPL Open PoC using SSH tunnelling to ArgoCD UI and command-line tools

The initial activities involved preparing the infrastructure with suitable memory (16 GB recommended) and storage capacity to house the deployment of containerized services. First, we set up the virtual machines and the Kubernetes cluster, then we moved forward with the deployment of the essential SIMPL Open components, which encompass the Identity & Access Management (IAM) layer, the Data Connector layer, and the Registry layer. Additionally, we investigated the integration with Vault and Kafka, to secure data access and facilitate event-driven data exchange.

4.3.2.4 Verifying Application Runtime Status

Upon registering and synchronizing the ArgoCD application for the common components, we verified the deployment status of the individual pods in the common namespace. This validation step was essential to ensure that all services such as Kafka brokers, Vault, and supporting containers were properly launched and healthy. By executing the command "*kubectl get pods -n common –w*", below is a snapshot of the pod status output at a stable point in the deployment lifecycle:

NAME	READY	STATUS	RESTARTS	AGE
confluent-operator-common-5d658f54bc-5fwkt	0/1	CrashLoopBackOff	6867 (2m51s ago)	26d
eck-operator-common-0	1/1	Running	3556 (35m ago)	26d
elastic-elasticsearch-es-node-0	0/2	Pending	0	25d
elastic-elasticsearch-es-node-1	0/2	Pending	0	25d
elastic-elasticsearch-es-node-2	0/2	Pending	0	25d
filebeat-beat-filebeat-4w556	1/1	Running	7 (4h20m ago)	25d
heartbeat-beat-heartbeat-7c84996fd7-qm2bq	1/1	Running	0	25d
kafka-0	1/2	Running	3144 (7m7s ago)	18d
kafka-1	1/2	Running	3078 (14m ago)	18d
kafka-2	1/2	CrashLoopBackOff	3136 (2m53s ago)	18d
kraftcontroller-0	0/1	Running	798 (5m21s ago)	26d
kraftcontroller-1	0/1	Running	688 (12m ago)	26d
kraftcontroller-2	0/1	Running	664 (15m ago)	26d
logstash-beats-ls-0	0/2	Pending	0	25d
metricbeat-beat-metricbeat-x9k59	1/1	Running	16 (11m ago)	25d
redpanda-749b645cc8-dtqtp	0/1	CrashLoopBackOff	10763 (3m53s ago)	26d
vault-common-0	0/1	Running	13 (4m31s ago)	25d
vault-common-agent-injector-768cf5f787-2ffd4	0/1	CrashLoopBackOff	6675 (3m35s ago)	25d
kafka-2	1/2	Running	3137 (18d ago)	18d
kraftcontroller-1	1/1	Running	688 (12m ago)	26d

Figure 13 Showing the Service Availability in the Common Namespace

4.3.2.5 Challenges Faced and Lessons Learned

The implementation of the SIMPL Open PoC within the interTwin framework revealed some important challenges, particularly in orchestrating its complex service stack on Kubernetes. A major issue encountered was the management of the container lifecycle and the interdependencies of strongly connected components such as Vault, Kafka, and various SIMPL Open modules. While containerization is designed to facilitate deployment, the dynamic nature of pods, especially concerning init containers and Vault-agent injection, demanded careful tuning.

Furthermore, although ArgoCD offered a user-friendly interface for GitOps-oriented deployment and synchronization, effective real-time debugging of malfunctioning pods still required a command-line interface and a deep understanding of Kubernetes networking, volumes, and readiness/liveness probes. The deployment of services like the Metrics Server to enhance clarity also presented challenges. Despite numerous efforts, the service frequently did not pass health checks due to issues with Transport Layer Security (TLS) and API access, adversely affecting node monitoring.

Through these activities, we extended extensive operational knowledge in cloud-native deployment strategies, container security over HashiCorp Vault, and infrastructure scaling. These challenges highlighted the importance of tight integration between Identity and Access Management) IAM, registry, and connector services in a data-space architecture. The PoC not only discovered challenges associated with real-world deployment but also demonstrated the necessity for vigilant orchestration, monitoring, and alternative strategies in distributed systems. It must be noted here that this pilot did not investigate the connection to EuroHPC environment or HPC environments in general that could pose additional challenges that are not explored in this study.

5 Summary

This deliverable illustrates, through use-case implementations, the potential and actual working examples of interoperability between DestinE and interTwin. Several key elements of interTwin will be carried forward in future projects like for instance RI-Scale or further developed by CERN as part of their FTS and Rucio operational environments and in contributions to the CERN openlab activities. It could also be adopted by the wider community, interlink is an example of such uptake [45] opening the path for those communities to be included in DestinE with reduced effort. We found that in general the DTE architecture concepts of both DestinE and interTwin share a common high-level design. From this we can imply that in general these address the relevant needs, and the use of open standards and reference implementations support the vision to be interoperable with the DestinE DTE with a reasonable amount of effort. Obviously DestinE is expanding its interfaces to tailor for a wider scope of communities, where the EDR interface and CoverageJSON output formats serve as an example of this activity. DestinE ensures that these more generic interfaces are backed by performant backends that leverage bespoke implementations that take advantage of a deep understanding of data layouts and access patterns that have been built over the last decades. The same concept holds for ecflow and a CWL interface where CWL can potentially serve a wider range of communities while ecflow ensures close embedding in DestinE and existing operational numerical weather prediction (NWP) activities.

6 References

Reference	
No	**Description / Link**
R1	THE DIGITAL TWIN ENGINE n.d. https://stories.ecmwf.int/the-digital-twin-engine/ (accessed August 12, 2025).
R2	DestinationEarth DataLake n.d. https://data.destination-earth.eu/ (accessed August 12, 2025).
R3	DestinE Platform – Your gateway to a sustainable future n.d. https://platform.destine.eu/ (accessed August 13, 2025).
R4	Wedi N, Bauer P, Sandu I, Hoffmann J, Sheridan S, Cereceda R, et al. Destination Earth: High-Performance Computing for Weather and Climate. Comput Sci Eng 2022;24:29–37. https://doi.org/10.1109/MCSE.2023.3260519
R5	Wedi N, Sandu I, Bauer P, Acosta M, Andersen RC, Andrae U, et al. Implementing digital twin technology of the earth system in Destination Earth. Journal of the European Meteorological Society 2025;3:100015. https://doi.org/10.1016/J.JEMETS.2025.100015
R6	Geenen T, Wedi N, Milinski S, Hadade I, Reuter B, Smart S, et al. Digital twins, the journey of an operational weather prediction system into the heart of Destination Earth. Procedia Comput Sci 2024;240:99–109. https://doi.org/10.1016/J.PROCS.2024.07.013
R7	C3S: Climate Data Store and Tool Box | ECMWF n.d. https://www.ecmwf.int/en/elibrary/80558-c3s-climate-data-store-and-tool-box (accessed August 25, 2025).
R8	Pangeo: A community for open, reproducible, scalable geoscience n.d. https://pangeo.io/ (accessed August 25, 2025).
R9	Copernicus and Sentinel data at your fingertips - WEkEO n.d. https://wekeo.copernicus.eu/ (accessed August 25, 2025).
R10	EWC | The European Weather Cloud n.d. https://europeanweather.cloud/ (accessed August 25, 2025).

R11	Software Strategy and Roadmap 2023-2027 | ECMWF n.d. https://www.ecmwf.int/en/elibrary/81334-software-strategy-and-roadmap-2023-2027 (accessed August 13, 2025).
R12	Hawkes J, Manubens N, Danovaro E, Hanley J, Siemen S, Raoult B, et al. Polytope: Serving ECMWFs Big Weather Data 2020. https://doi.org/10.5194/EGUSPHERE-EGU2020-15048
R13	Smart SD, Quintino T, Raoult B. A Scalable Object Store for Meteorological and Climate Data 2017;8. https://doi.org/10.1145/3093172.3093238
R14	OGC CoverageJSON Community Standard n.d. https://docs.ogc.org/cs/21-069r2/21-069r2.html (accessed August 13, 2025)
R15	The STAC Specification n.d. https://stacspec.org/en/about/stac-spec/ (accessed August 13, 2025).
R16	OGC API - Environmental Data Retrieval Standard n.d. https://docs.ogc.org/is/19-086r6/19-086r6.html (accessed August 13, 2025).
R17	openEO n.d. https://openeo.org/ (accessed August 25, 2025).
R18	Schick M, Puechmaille D. PRESENTATION TITLE ECMWF-DESTINATION EARTH DATA LAKE DESTINATION EARTH n.d.
R19	Vazquez-Ingelmo A, Garcia-Holgado A, Garcia-Penalvo FJ. C4 model in a software engineering subject to ease the comprehension of UML and the software. IEEE Global Engineering Education Conference, EDUCON 2020;2020-April:919–24. https://doi.org/10.1109/EDUCON45650.2020.9125335
R20	Harmonised Data Access (HDA) — Destination Earth Data Lake 0.0.1 documentation n.d. https://destine-data-lake-docs.data.destination-earth.eu/en/latest/dedl-discovery-and-data-access/Use-of-Harmonized-Data-Access/Use-of-Harmonized-Data-Access.html (accessed August 13, 2025).
R21	User Tools · FTS3 Documentation n.d. https://fts3-docs.web.cern.ch/fts3-docs/docs/cli.html (accessed August 13, 2025).
R22	Barisits M, Beermann T, Berghaus F, Bockelman B, Bogado J, Cameron D, et al. Rucio: Scientific Data Management. Comput Softw Big Sci 2019;3. https://doi.org/10.1007/S41781-019-0026-3
R23	OAuth 2.0 — OAuth n.d. https://oauth.net/2/ (accessed August 13, 2025).

R24	EGI Federation - Service - Check-in n.d. https://www.egi.eu/service/check-in/ (accessed August 25, 2025).
R25	Arora A, Guiang J, Davila D, Würthwein F, Balcas J, Newman H. 400Gbps benchmark of XRootD HTTP-TPC 2023. https://doi.org/10.1051/epjconf/202429501001
R26	Introduction - OSCAR Documentation n.d. https://docs.oscar.grycap.net/latest/ (accessed August 25, 2025).
R27	Crusoe MR, Abeln S, Iosup A, Amstutz P, Chilton J, Tijanić N, et al. Methods included. Commun ACM 2022;65:54–63. https://doi.org/10.1145/3486897
R28	Managing work flows with ecFlow | ECMWF n.d. https://www.ecmwf.int/en/elibrary/80182-managing-work-flows-ecflow (accessed August 13, 2025).
R29	Home | Simpl Programme n.d. https://simpl-programme.ec.europa.eu/ (accessed August 13, 2025).
R30	International Data Spaces Association· GitHub n.d. https://github.com/International-Data-Spaces-Association (accessed August 13, 2025).
R31	Overview - Gaia-X Architecture Document - 22.10 Release n.d. https://docs.gaia-x.eu/technical-committee/architecture-document/22.10/overview/ (accessed August 13, 2025).
R32	ELK Stack + Kafka End to End Practice — Log Consolidation with ELK Stack 1.2 documentation n.d. https://elastic-stack.readthedocs.io/en/latest/e2e_kafkapractices.html (accessed August 25, 2025).
R33	Apache Kafka n.d. https://kafka.apache.org/downloads (accessed August 25, 2025).
R34	High-Performance Streaming Data Platform | Redpanda n.d. https://www.redpanda.com/ (accessed August 25, 2025).
R35	HashiCorp Vault | Identity-based secrets management n.d. https://www.hashicorp.com/en/products/vault (accessed August 25, 2025).

R36	HashiCorp Vault | Identity-based secrets management n.d. https://www.hashicorp.com/en/products/vault (accessed August 25, 2025).
R37	Argo CD - Declarative GitOps CD for Kubernetes n.d. https://argo-cd.readthedocs.io/en/stable/ (accessed August 25, 2025).
R38	PostgreSQL: The world's most advanced open-source database n.d. https://www.postgresql.org/ (accessed August 25, 2025).
R39	Beetz F, Harrer S. GitOps: The Evolution of DevOps? IEEE Softw 2022;39:70–5. https://doi.org/10.1109/MS.2021.3119106
R40	Helm n.d. https://helm.sh/ (accessed August 25, 2025).
R41	The most-comprehensive AI-powered DevSecOps platform n.d. https://about.gitlab.com/ (accessed August 25, 2025).
R42	The Official YAML Web Site n.d. https://yaml.org/ (accessed August 25, 2025).
R43	NodePort :: The Kubernetes Networking Guide n.d. https://www.tkng.io/services/nodeport/ (accessed August 25, 2025).
R44	Download PuTTY: latest release (0.83) n.d. https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html (accessed August 25, 2025).
R45	interLink - Kubernetes to Everything | interLink n.d. https://interlink-project.dev/ (accessed August 12, 2025).