

interTwin

D6.4 Final release of the DTE core modules

Status: Under EC Review
Dissemination Level: Public



Funded by the
European Union


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

Abstract

Key Words DTE, Core, development, integration

The deliverable includes the status of the development and integration of all WP6 software products available with the final release of the interTwin DTE. It includes a description of each component, link to source code and documentation and related release notes. A summary of the integration activity is also reported as conclusions.



Document Description			
D6.4 Final release of the DTE core modules			
Work Package number 6			
Document type	Deliverable		
Document status	Under EC Review	Version	1.0
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	LIP		
Document link	https://documents.egi.eu/document/3953		
DOI	https://zenodo.org/records/14778361		
Author(s)	<ul style="list-style-type: none"> • Isabel Campos (CSIC) • Germán Moltó (UPV) • Alexander Jacob (EURAC) • Pablo Orviz (CSIC) • Miguel Caballer (UPV) • Matteo Bunino (CERN) • Anna Elisa Lappe (CERN) • Jarl Sondre Saether (CERN) • Rakesh Sarma (FZJ) • Sandro Fiore (UNITN) • Estíbaliz Parceró (UPV) • Donatello Elia (CMCC) 		
Reviewers	<ul style="list-style-type: none"> • Renato Santana (EGI Foundation) • Mattias Schramm (TU Wien) 		
Moderated by:	<ul style="list-style-type: none"> • Andrea Anzanello (EGI Foundation) 		
Approved by	AMB		



Revision History			
Version	Date	Description	Contributors
v0.1	28/11/2024	ToC	Isabel Campos (CSIC)
v0.2	15/01/2025	Version ready for internal review	All authors
v0.3	22/01/2025	Internal Review	Renato Santana (EGI Foundation), Matthias Schramm (TU Wien)
v0.4	30/01/2025	Version ready for QA	All authors
v1.0	31/01/2025	Final	

Term/Acronym	Definition
Terminology / Acronyms	
AI	Artificial Intelligence
API	Application Programming Interface, aka programmatic interface of a computer system through which other computer systems can interact with it
CI/CD	In software engineering, CI/CD is the combined practices of continuous integration and continuous delivery
CLI	Command line interface
CWL	Common Workflow Language
DT	Digital Twin, a digital representation of an actual physical product, system or process that serves as the effectively indistinguishable digital counterpart of it for practical purposes, such as simulation, integration, testing, monitoring, and maintenance.
DTE	Digital Twin Engine, a platform to build DTs
GUI	Graphical user interface
JSON	JavaScript Object Notation
ML	Machine Learning is a branch of AI and computer science which focuses on the use of data and algorithms to imitate the way that humans learn, gradually improving its accuracy
REST API	API that conforms to the design principles of REST, or representational state transfer architectural style

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



Table of Contents

1 Introduction	7
1.1 Scope	7
1.2 Document Structure	7
2 Core Components	8
2.1 Components for Task 6.1	8
2.1.1 OSCAR	8
2.1.2 DCNiOS	10
2.1.3 openEO	11
2.1.4 Ophidia	12
2.1.5 yProv	14
2.2 COMPONENTS FOR TASK 6.2	16
2.2.1 SQAaaS platform	16
2.2.2 SQAaaS GitHub Actions	17
2.2.3 SQAaaS CLI	18
2.3 COMPONENTS FOR TASK 6.3	19
2.3.1 openEO	19
2.4 COMPONENTS FOR TASK 6.4	21
2.4.1 Infrastructure Manager	21
2.4.2 Big Data Analytics TOSCA templates	22
2.4.3 Configuration artefacts	23
2.5 COMPONENTS FOR TASK 6.5	24
2.5.1 itwinai	24
3 Summary of integration status	26
4 References	28

Table of Tables

<u>Table 1 - List of core components under release by task.</u>	<u>8</u>
---	----------



Executive summary

This report updates the Digital Twin Engine core services description presented in the deliverable D6.2 [R2] as the first version. The design considerations were addressed in previous deliverables D6.1[R1] and D6.3[R3].

Real time data ingestion and analysis is supported by OSCAR and DCNiOS. Workflow management is being developed in the frameworks of openEO API and the Ophidia tools. The yProv tool enhances workflows with data provenance information. Model quality and validation are supported by the evolution of the SQAaaS service towards automated validation.

The interplay between the infrastructure and the required big data analytics core services is dealt with by the Infrastructure Manager (IM), complemented by a set of TOSCA templates and configuration artefacts repositories. Several already existing tools and APIs are integrated here as a core component.

Support for AI-oriented workflows is being handled by itwinai, a user-facing component developed in intertwin from scratch. It consists of a python-based library that streamlines AI workflows, while integrating with HPC and cloud resources.

The current integration and testing strategy is also described in terms of integration between core tools as required by the different DT use cases.

1 Introduction

1.1 Scope

This deliverable summarises the development status of core components, relevant to implement the architecture of a Digital Twin Engine in the framework of interTwin.

1.2 Document Structure

Section 2 lists all core services that have been developed within Work Package 6 or that are still under development. It provides details on their functionalities, release notes and future plans. **Section 3** provides a summary of the integration status with the rest of the project Work Packages.

2 Core Components

The list of core components to be developed in Work Package 6 is shown in [Table 1](#). The design considerations are available in deliverable D6.3 ([R3](#)).

Table 1 - List of core components under release by task.

Task 6.1	Task 6.2	Task 6.3	Task 6.4	Task 6.5
<ul style="list-style-type: none"> - OSCAR - DCNiOS - openEO - Ophidia - yProv 	<ul style="list-style-type: none"> - SQAaaS - SQAaaS Github action - SQAaaS CLI 	<ul style="list-style-type: none"> - openEO 	<ul style="list-style-type: none"> - Infrastructure Manager (IM) - TOSCA templates - Configuration artefacts 	<ul style="list-style-type: none"> - itwinai

2.1 Components for Task 6.1

2.1.1 OSCAR

Component name	OSCAR https://oscar.grycap.net
Description	OSCAR is an open-source platform to support the event-driven serverless computing model for data-processing applications. It can be automatically deployed on multi-clouds, and even on low-powered devices, to create highly parallel event-driven data-processing serverless applications along the computing continuum. These applications execute on customised runtime environments, provided by Docker containers which run on elastic Kubernetes clusters.
Value proposition	Users can set up an OSCAR cluster themselves on any available cloud infrastructure. The automatically scalable cluster can be used to scale file-based on-demand processing (e.g. automatically when a file is uploaded to an object store such as MinIO). Furthermore, it allows communication via HTTP-based calls for programmatic interaction with auto-scaled, stateless, user-defined services.
Users of the Component	Scientific users require data-driven processing on multiple cloud back-ends. Non-expert users can

	interact via high-level web-based GUIs, while advanced users can use a command-line interface (CLI).
User Documentation	https://oscar.grycap.net/blog/
Technical Documentation	https://docs.oscar.grycap.net/
Responsible	GRyCAP-I3M-UPV - products@grycap.upv.es Contact point: Germán Moltó - gmolto@dsic.upv.es
License	Apache 2.0
Source code	https://github.com/grycap/oscar

2.1.1.1 Release notes

OSCAR has been adopted by the interTwin project to implement a generic framework for real-time data acquisition and processing that builds on event-triggered execution of workflows, to i) improve response times and minimise data transfers, and to ii) support new event sources.

For this release, Apache NiFi, an open-source system that supports powerful and scalable directed graphs of data routing, transformation, and system mediation logic, has been firmly integrated with OSCAR. The following aspects outline the recent updates and current implementations:

Data Lake integration:

- Using Apache NiFi, Amazon S3 + Amazon SQS is employed; its configuration is managed via DCNiOS (see [DCNiOS](#)).
- Using Apache NiFi, dCache is supported as a source of events; its configuration is managed via DCNiOS.

Real-time data acquisition:

- Using Apache NiFi, Kafka is supported as a source of events; its configuration is managed via DCNiOS.

Integration with other DTE components:

- interLink: The integration of the OSCAR component with the interLink component to offload computation to HPC has already been completed.
- itwinai, an AI pipeline management tool, has been integrated into OSCAR.
- JupyterHub has been integrated employing MinIO as a storage backend, seamlessly connecting the storage of both systems.

Use case integration:

- The integration of CERN's 3DGAN inference for particle simulation in the LH-LHC was successfully completed, demonstrating the use of itwinai, dCache, and interLink.
- The integration of DT-Flood (a digital twin for flood detection and flood impact estimation) by Deltares and TU Wien is about to be completed, enabling the offloading of computationally intensive steps in their CWL workflow to OSCAR.



D6.4 Final release of the DTE core modules

- The integration of EURAC's Drought Forecasting use case has been achieved by enabling the offloading of the computationally intensive steps in their workflow within OSCAR.

Software quality:

- The integration with the SQAaaS platform enables OSCAR to automatically perform quality assessments for each release, facilitated by a GitHub action on every pull request.

2.1.1.2 Future plans

OSCAR will be evolved to address the arising needs of the interTwin project, ensuring it remains robust and adaptable to future requirements with a focus on use case integration, improved notebook support, and data lake integration.

2.1.2 DCNiOS

Component name	DCNiOS
Description	DCNiOS (Data Connector through NiFi for OSCAR) is a new open-source command-line tool to easily manage the creation of event-driven data processing flows.
Value proposition	When files are uploaded to data storage, events are ingested by Apache NiFi, which can queue them up depending on the (modifiable at runtime) ingestion rate. Then, they are delegated for processing into a scalable OSCAR cluster, where a user-defined application, based on a Docker image, can process the data file.
Users of the Component	Technology Integrators
User Documentation	http://github.com/interTwin-eu/dcnios
Technical Documentation	http://github.com/interTwin-eu/dcnios
Responsible	GRyCAP-I3M-UPV - products@grycap.upv.es Contact point: Germán Moltó - gmolto@dsic.upv.es
License	Apache 2.0 License



Source code	https://github.com/grycap/dcnios
-------------	---

2.1.2.1 Release notes

DCNiOS, initially developed within the interTwin project to integrate dCache with OSCAR, has significantly evolved to support a myriad of data sources. This component now includes implementations for supporting Amazon S3 along with Amazon SQS and Kafka as source of events, as well as any generic element that can be created as an Apache NiFi ProcessGroup. Additionally, DCNiOS supports alterations that modify the received input data during the execution of the data flow.

DCNiOS uses the SQAaaS platform (see section 2.2.1) to perform quality assessments for each release, ensuring thorough evaluation and reliability.

2.1.2.2 Future plans

DCNiOS will continue to adapt to emerging requirements, ensuring it remains a versatile and reliable component within the interTwin project, with a focus on use case integration and the improvement of the documentation.

2.1.3 openEO

Component name	openEO
Description	openEO is an application programming interface (API) that supports i) the management of workflows, ii) job handling, and iii) linking to data sources and processing capabilities on compatible cloud platform providers in a standardised way.
Value proposition	openEO can be extended to support execution of containerized software packages as execution of specific processes following the OGC API processes approach. openEO has many implementations, but a specific set of them can be selected for handling workflows in interTwin.
Users of the Component	Digital Twin Developers
User Documentation	https://openeo.org/documentation/1.0/
Technical Documentation	https://openeo.org/documentation/1.0/developers/



Responsible	Eurac, EODC, Münster University, Alexander Jacob, Christoph Reimer, Brian Pondi alexander.jacob@eurac.edu
-------------	--

2.1.3.1 Release notes

For the interTwin project a specific deployment has been composed that i) based on the available open-source components under the openEO's GitHub repository and ii) allows the further integration of new component types into the workflows, based on the openEO process graphs. Those new types to be integrated are set up as the so-called application packages that are defined by the EOEPKA architecture¹ and that are used in the OGC standard² (e.g. as the OGC API processes). A first prototype has been developed integrating one specific processor in this way, based on the wflow hydrological model³. Furthermore, parts of the use case on flood mapping based on the Global Flood Mapping algorithm, have also been ported to openEO (see Task 7.5). Initial OSCAR and openEO integration has started to trigger Application Package processing from openEO in the OSCAR cluster.

2.1.3.2 Future plans

This first process can act as a template for other thematic modules, to be defined and set up the same way and then to be executed. Ideally, new components should also be developed to interact with the other core components (e.g. with the iTwinAI package; see Task 6.5) That could probably also fully be described in this way. Also, a connection to the OSCAR and DCNiOS event-driven triggering of workflows is foreseen, so that the event can trigger the execution of openEO process graphs. As a next step, a TOSCA template for this environment should be developed together with Task 6.4.

2.1.4 Ophidia

Component name	Ophidia/PyOphidia
Description	PyOphidia provides the Python bindings for Ophidia, a High-Performance Data Analytics framework
Value proposition	Ophidia framework is an open-source solution for the analysis of scientific multi-dimensional data, joining HPC paradigms and Big Data approaches. It provides an environment targeting High-Performance Data Analytics through parallel and in-memory data processing, data-driven task scheduling and server-side analysis. The framework supports the execution of complex analytics workflows in the form

¹ <https://eoepka.org/>

² <https://www.ogc.org/standards>

³ <https://deltares.github.io/Wflow.il/stable/>



	of DAGs of Ophidia operators. Integration with other Python libraries and tools (e.g., Jupyter notebooks) is supported through the PyOphidia module.
Users of the Component	Data scientists, DT developers
User Documentation	https://pyophidia.readthedocs.io/en/latest/
Technical Documentation	https://pyophidia.readthedocs.io/en/latest/installation.html
Responsible	CMCC, Ophidia dev team: ophidia-info@cmcc.it Contact point: Donatello Elia - donatello.elia@cmcc.it
License	GPLv3
Source code	PyOphidia: https://github.com/OphidiaBigData/PyOphidia Ophidia server: https://github.com/OphidiaBigData/ophidia-server

2.1.4.1 Release notes

The latest release of the Ophidia framework components, and in particular the Ophidia server and PyOphidia, includes all the extensions developed within the project. These extensions cover two main features: support for Common Workflow Language (CWL) based workflows and provenance tracking.

For the former development, stronger integration of Ophidia workflows written in CWL within the PyOphidia modules has been achieved. Now CWL-based workflows can be directly imported through a method in the Python module and handled similarly to workflows written in the native Ophidia JSON-based format.

Concerning the latter, the support for tracking and generating provenance documents of workflows has been introduced in Ophidia. In particular, the Ophidia server has been extended to track and send back to the client modules additional information related to task execution. On the client side, PyOphidia has been extended with the capabilities: (i) to collect the information concerning the workflow execution at runtime, and (ii) to generate related provenance documents compliant with the W3C PROV standard.

Moreover, the capabilities for validating the workflow structure have also been improved and fully integrated into the related PyOphidia modules.

Besides code developments, notebooks with usage examples of the new features have been included in the PyOphidia package. The documentation is also being updated to describe the latest capabilities introduced (see table above).

2.1.4.2 Future plans

The focus will be on the integration of Ophidia/PyOphidia with DT applications from WP4, for example, those concerning climate-related extreme events. For example, the capabilities for handling CWL/JSON workflows will be further enhanced to better address requirements from the use cases. Moreover, integration with SQAaaS for the validation of Ophidia workflow syntax is envisioned.

2.1.5 yProv

Component name	yProv
Description	yProv is an open-source software ecosystem to support provenance management within scientific workflows. It relies on the W3C PROV family of standards, a RESTful interface and a graph database back-end based on Neo4J. The yProv web service (main component) is implemented in Python by using the Flask micro-framework which is based on the Jinja2 Template Engine and Werkzeug WSGI Toolkit. The service is domain-agnostic, though its primary case studies in the project come from the climate change domain (i.e. climate analytics workflows). The service aims at implementing the micro-provenance concept, to navigate within the provenance space across different dimensions (e.g., horizontal & vertical). yProv includes also the Command Line Interface and additionally, it delivers support for provenance tracking in AI, which adds extra capabilities in key and recurring use cases across different DTs.
Value proposition	Users can exploit the yProv service to manage (i.e. store, retrieve, explore, visualise) the provenance information associated with scientific datasets, thus getting a better understanding about specific datasets. The value proposition is about (i) stronger traceability, transparency, and trust (through a richer set of metadata) and (ii) multidimensional exploration/navigation of provenance metadata information (i.e., multi-level).
Users of the Component	Scientific users, both producers and consumers of datasets. End users can interact via the yProv RESTful API to manage (i.e., CRUD operations) the provenance information.



User Documentation	yProv service: https://github.com/HPCI-Lab/yProv/blob/main/README.md yProv CLI: https://github.com/HPCI-Lab/yProv-CLI/blob/main/README.md yProv4ML: https://github.com/HPCI-Lab/yProvML https://hpci-lab.github.io/yProv4ML.github.io/index.html
Technical Documentation	yProv service: https://github.com/HPCI-Lab/yProv/blob/main/README.md yProv CLI: https://github.com/HPCI-Lab/yProv-CLI/blob/main/README.md yProv4ML: https://github.com/HPCI-Lab/yProvML https://hpci-lab.github.io/yProv4ML.github.io/index.html
Responsible	University of Trento Contact point: Sandro Fiore (sandro.fiore@unitn.it)
License	GPLv3
Source code	yProv: https://github.com/HPCI-Lab/yProv/ yProv-CLI: https://github.com/HPCI-Lab/yProv-CLI yProvML: https://github.com/HPCI-Lab/yProvML

2.1.5.1 Release notes

yProv has been adopted in InterTwin to implement provenance support within scientific workflows, starting from some case studies identified in the environmental domains (i.e. climate data analytics workflows). Concerning the initial design, focusing on the core service, yProv delivers a software ecosystem which includes a service, libraries and tools. The current release has been integrated:

- in climate-related DTs for extreme events both at CMCC and UNITN;
- with itwinai (in collaboration with CERN);
- with IM for automated cloud deployment over Kubernetes clusters (in collaboration with UPV);
- with the SQAaaS platform through the available API (in collaboration with CSIC).



2.1.5.2 Future plans

yProv will evolve, during interTwin, to accommodate additional requirements. Ongoing and future activities include: (i) the integration of provenance tracking in openEO (now early stage); (ii) a new release of yProv fixing minor bugs that will be discovered; (iii) provenance support in the SQA process; (iv) stronger integration of metrics within the yProv libraries; (v) a first release of the yProv explorer, which will offer a graphical UI to navigate and inspect provenance documents; and finally, the integration with additional DTs.

2.2 COMPONENTS FOR TASK 6.2

2.2.1 SQAaaS platform

Component name	SQAaaS https://sqaaas.eosc-synergy.eu
Description	Platform for quality assessment and awarding of multiple digital objects (source code, services, data)
Value proposition	Provide a module for quality validation within the interTwin's DTE
Users of the Component	DTE developers & users
User Documentation	https://docs.sqaaas.eosc-synergy.eu https://indigo-dc.github.io/jenkins-pipeline-library
Technical Documentation	https://github.com/eosc-synergy/sqaaas-api-spec
Responsible	Pablo Orviz < orviz@ifca.unican.es > Samuel Bernardo < samuel@lip.pt > David Arce < darce@i3m.upv.es >
License	GPL-3.0-only
Source code	https://github.com/eosc-synergy/sqaaas-api-server https://github.com/eosc-synergy/sqaaas-web https://github.com/indigo-dc/jenkins-pipeline-library



2.2.1.1 Release notes

The SQAaaS API server is now in version 3.2.3 including many new features. Also the Jenkins library is in version 2.1.1.

The main enhancements include:

- Individual triggering of QA checks, as required by the integration with WfMSs (i.e. performing a given QA task as individual steps in the workflow definition. This feature is available through the API path:
 - /pipeline/assessment?run_criteria_workflow_only=true
- Support to scale up the SQA testing to the Google Kubernetes Engine (GKE)

2.2.1.2 Future plans

The next steps for T6.2 are:

- Integrate the SQAaaS platform with the WfMS standards and specific technologies being supported by T6.1, such as CWL, and use cases namely from Extreme events (T4.5) and Lattice-QCD (T4.1).

2.2.2 SQAaaS GitHub Actions

Component name	SQAaaS assessment GitHub Action
Description	Trigger SQAaaS quality assessment service from GitHub actions
Value proposition	Integrate interTwin's GitHub organisation with the SQAaaS platform for software quality (incl. workflow and model code)
Users of the Component	DTE developers & users
User Documentation	https://github.com/EOSC-synergy/sqaas-assessment-action https://github.com/EOSC-synergy/sqaas-step-action
Technical Documentation	https://github.com/eosc-synergy/sqaas-gh-action https://github.com/EOSC-synergy/sqaas-step-action
Responsible	Pablo Orviz < orviz@ifca.unican.es >
License	GPL-3.0-only
Source code	https://github.com/EOSC-synergy/sqaas



	as-assessment-action https://github.com/EOSC-synergy/sqaaas-step-action
--	--

2.2.2.1 Release notes

The current release includes two GitHub actions (**sqaaas-assessment-action** and **sqaaas-step-action**) that enable the automated assessment of source code, including workflow and model code, by triggering the SQAaaS platform. More precisely, the main action (sqaaas-assessment-action) is in charge of interacting with the SQAaaS API, running the appropriate HTTP requests to conduct the source code assessment. As an output of this action, a summary containing the quality criteria being analysed is provided, and, in the event that a certain level of these criteria has been fulfilled, the corresponding digital badge that recognizes those achievements.

The current version (2.4.1) was released in March 2024. This action triggers the quality assessment of a source code repository. Improvements include the inclusion of quality badges and compliance with REUSE. Copyright and licensing is difficult, especially when reusing software from different projects that are released under various different licenses. **REUSE** provides a set of recommendations to make licensing easier. They also make it easier for a computer to understand how your project is licensed.

As a complement, the step action (sqaaas-step-action) adds the capability to define customised steps as part of the evaluation of a quality criterion within the SQAaaS source code assessment. This is required, for instance, for the testing criteria, where diverse testing frameworks might be used (e.g. Python's pytest). Additionally, this action serves the purpose of covering pre/post requirements that might be needed as part of the quality criteria validation. An example could be setting up the environment as a 'pre' condition before proceeding with the actual testing process (e.g. following the Python example: conda, virtualenv, etc.). The current release is 1.3.2 (March 2024) and includes fixes on containers gathering features.

2.2.2.2 Future plans

Extend the main GitHub action ([sqaaas-assessment-action](https://github.com/EOSC-synergy/sqaaas-assessment-action)) to cope with other types of quality assessments currently provided by the SQAaaS platform, in particular the validation of FAIR principles for data.

2.2.3 SQAaaS CLI

Component name	SQAaaS CLI
Description	Command line interface to interact with SQAaaS API from scripts
Value proposition	Integrate interTwin's workflows within



D6.4 Final release of the DTE core modules

	pipeline steps, creating assessments, getting outputs, checking pipeline status
Users of the Component	DTE developers & users
User Documentation	https://gitlab.a.incd.pt/eosc/eosc-synergy/sqaaas-cli/-/tree/cli-2.6.0
Technical Documentation	https://gitlab.a.incd.pt/eosc/eosc-synergy/sqaaas-cli/-/tree/cli-2.6.0
Responsible	Samuel Bernardo < samuel@lip.pt > Pablo Orviz < orviz@ifca.unican.es >
License	GPL-3.0-only
Source code	https://gitlab.a.incd.pt/eosc/eosc-synergy/sqaaas-cli/-/tree/cli-2.6.0

2.2.3.1 Release notes

The SQAaaS CLI is a work in progress, to answer the custom assessment required for DTE workflow integration into the SQAaaS platform. The CLI approach provides a way to launch commands from the workflows and run the required steps.

Release 2.6.0 reflects the SQAaaS production schema is not supporting the custom assessment yet. But it answers the quality-checking task, providing the commands to create an assessment, get assessment outputs, run pipelines, and get pipeline status.

2.2.3.2 Future plans

Complete the developments towards the custom assessment (CA), that will answer the DTE requirements for the model validation. Two use cases will be finalized: trigger assessment from GitHub and from a workflow step. The new features will comprehend the configuration (from a file, template, or web interface), register, and uniquely identify the CA from a URI representation.

2.3 COMPONENTS FOR TASK 6.3

The Core components for data fusion are partially aligned with the core components for the workflow management tools like openEO.

2.3.1 openEO

Component name	openEO
----------------	--------



D6.4 Final release of the DTE core modules

Description	openEO is an application programming interface (API), supporting management of workflows and handling of jobs, as well as linking to available data sources and processing capabilities in a harmonised way – front-end users don't need knowledge on the data structure.
Value proposition	The openEO syntax is already used for several specific predefined processes, dealing with a fusion of raster data. Those use cases can be further extended for integrating vector data and possibly data coming as output from different models from both the physical and data-driven domain as well as preparing data in a harmonised way for ingestion into such models.
Users of the Component	Digital Twin Developers
User Documentation	https://openeo.org/documentation/1.0/
Technical Documentation	https://openeo.org/documentation/1.0/developers/ , https://processes.openeo.org/
Responsible	Eurac, EODC, Münster University, Alexander Jacob, Christoph Reimer, Brian Pondi alexander.jacob@eurac.edu

2.3.1.1 Release notes

openEO-based processes for fusion of raster data have been already used in interTwin after the first release for the preparation of various sources of raster data in the environmental use cases, specifically dealing with early warning of extreme flood and drought events. TU Wien and Eurac are using those process libraries for monitoring flood and drought events; Deltares is currently evaluating their suitability for its own processes.

Based on application packages concept also developed in Task 6.1, a common workflow for preparing data building on components such as raster2stac and hydroMT has been set up to prepare data, that is then given to specific models such as wflow or LSTM networks or downscaleML setup using itwinai from T6.5 for evaluation.



2.3.1.2 Future plans

Activities started in the second year of the project in testing data access solutions provided by WP5, such as teapot and RUCIO and how they can work together with high-level API access through STAC and openEO, will be completed.

2.4 COMPONENTS FOR TASK 6.4

2.4.1 Infrastructure Manager

Component name	Infrastructure Manager (IM)
Description	The Infrastructure Manager (IM) is an open-source Infrastructure as Code (IaC) tool that provides both an XML-RPC and REST API to receive virtual infrastructure provision requests for their deployment on IaaS cloud back-ends. These requests may come from a web-based graphical user interface such as the IM-Dashboard, through the IM command-line interface (CLI) or via an HTTP-based client.
Value proposition	Users can self-provision any kind of complex virtual infrastructure on whichever cloud infrastructure they can access.
Users of the Component	Scientific users, requiring Big Data Analytics tools to be deployed on different Cloud back-ends.
User Documentation	https://imdocs.readthedocs.io/en/latest/
Technical Documentation	https://imdocs.readthedocs.io/en/latest/
Responsible	GRyCAP-I3M-UPV - products@grycap.upv.es Contact point: Miguel Caballer - micafer1@upv.es
License	GPL 3.0
Source code	https://github.com/grycap/im

2.4.1.1 Release notes

IM was adopted in interTwin for the deployment of the Big Data Analytics tools. This component has been maintained and bug-fixed, but no additional development has been made.



2.4.1.2 Future plans

IM is used in interTwin for the deployment of customized virtual infrastructure, according to the user requirements, mainly, for the Big Data Analytics tools. Maintenance and bug fixing will be provided.

2.4.2 Big Data Analytics TOSCA templates

Component name	Big Data Analytics TOSCA templates
Description	As a set of TOSCA templates to deploy Big Data Analytics tools
Value proposition	TOSCA templates enable the description, in a cloud-agnostic way, of the virtual infrastructures needed in the available Big Data Analytics tools.
Users of the Component	TOSCA template developers.
User Documentation	https://confluence.egi.eu/display/interTwin/TOSCA+Templates
Technical Documentation	https://confluence.egi.eu/display/interTwin/TOSCA+Templates
Responsible	UPV
License	Apache 2.0
Source code	https://github.com/grycap/tosca/tree/main/templates

2.4.2.1 Release notes

In the previous release, the following templates were created:

- **[KubeFlow](#)**: Template to deploy the Kubeflow machine learning (ML) workflows platform on top of Kubernetes.
- **[Airflow](#)**: Template to deploy the Apache workflows system on top of Kubernetes.
- **[CernVMFS](#)**: Install CernVMFS on a VM and mount a list of CernVM-FS repositories specified by the user.
- **[Kafka](#)**: Deploy Kafka distributed event streaming platform on top of a Kubernetes cluster.
- **[MLFlow](#)**: Deploy the MLFlow platform to manage the ML lifecycle in a single VM, with the possibility to store the artefacts in an external S3 (or MinIO) storage system.

In this release, the following templates have been created:

D6.4 Final release of the DTE core modules

- **yProv**: Deploy the yProv provenance service on top of a Kubernetes cluster using the [yProv helm chart](#).
- **openEO**: Deploy openEO on top of a Kubernetes cluster using the openEO argo [Helm chart](#).
- **STAC**: Deploy STAC catalog using PostgreSQL backend.
- **Horovod**: Deploy a Horovod cluster following [this install docs](#), launching 1 Front-end and a set of WN (with GPU) and another set of WNs (without GPU). In the case of GPU nodes, it installs the NVIDIA drivers and the NCCL 2 library. It creates a "horovod" user that can access passwordless SSH to all the nodes. It also installs NFS to share the /home directory from the FE to all the WNs.
- **EOEPCA ADES**: Installs ADES on top of a Kubernetes cluster. Using the [following documentation](#). It deploys the Processing profile deploying MinIO and the ZOO-Project DRU.
- **Ophidia**: Installs a Jupyter-Ophidia-based environment on top of a Kubernetes Cluster.

WP5 and WP6 members have tested the templates before the release, and plans for the testing with DT use cases have been established.

2.4.2.2 Future plans

Some of the templates are in an early stage (STAC, openEO and Ophidia) and need to be correctly tested by users with experience using these tools to validate the functionality of the deployed infrastructure. Other templates are more mature but may need some additions to improve them.

2.4.3 Configuration artefacts

Component name	Configuration artefacts
Description	Ansible playbooks and roles and any other artefact needed in the deployment of the selected Big Data Analytics tools. Those ansible playbooks are used by the TOSCA templates defined in section 2.4.2 to perform the actual deployments on cloud nodes.
Value proposition	TOSCA templates refer to these artefacts to enable the configuration of the Big Data Analytics tools using the Ansible tool.
Users of the Component	TOSCA template/Ansible playbook developers.
User Documentation	-
Technical Documentation	-
Responsible	UPV



D6.4 Final release of the DTE core modules

License	Apache 2.0
Source code	https://github.com/grycap/tosca/tree/main/artifacts

2.4.3.1 Release notes

These are the artefacts created to configure the templates defined in the previous section:

In the previous release:

- [KubeFlow](#), [Kafka on docker compose](#), [Kafka on Kubernetes](#), [CernVMFS](#) and [MLFlow](#), see [section 2.4.2.1](#) for the description of the components.

Added in this last release:

- [openEO Kubernetes Backend](#), [STAC with docker-compose](#), [Horovod Cluster](#) (two artefacts, one for the front-end and another for the working nodes) [EOEPCA ADES](#) and [Ophidia](#); see [section 2.4.2.1](#) for the description of the components.

2.4.3.2 Future plans

Some of the templates are in an early stage (STAC and openEO) and need to be correctly tested by users with experience, using these tools to validate the functionality of the deployed infrastructure. Other templates are more mature but may need some additions to improve them. Finally, further templates need to be created (e.g. for the ENES system).

2.5 COMPONENTS FOR TASK 6.5

2.5.1 itwinai

Component name	itwinai
Description	itwinai is a Python library that streamlines AI workflows, while reducing engineering complexity. It seamlessly integrates with HPC resources, making workflows highly scalable and promoting code reuse. With built-in tools for hyper-parameter optimization, distributed machine learning, and pre-trained ML models, itwinai empowers AI researchers. It integrates smoothly with Jupyter-like GUIs, enhancing accessibility and usability. It also supports containerized execution of the library, allowing ease of deployment.



D6.4 Final release of the DTE core modules

Value proposition	Different interfaces are provided to lower the entry barrier for users coming from different fields of expertise: Python API, command line interface, and high-level GUI. itwinai provides out-of-the-box state of the art AI tools. It encourages code reuse, to further simplify and streamline the development of ML workflows, on top of seamless integration with HPC resources.
Users of the Component	DT developers, ML engineers and researchers.
User Documentation	https://itwinai.readthedocs.io/
Technical Documentation	https://itwinai.readthedocs.io/
Responsible	CERN, FZ Juelich. Contact points: Matteo Bunino - matteo.bunino@cern.ch , Rakesh Sarma - r.sarma@fz-juelich.de
License	Apache 2.0
Source code	https://github.com/interTwin-eu/itwinai

2.5.1.1 Release notes

The itwinai library has been adopted in interTwin for the definition of Advanced AI workflows. Up-to-date release notes can be found in the GitHub repository⁴.

2.5.1.2 Future plans

Future extensions to itwinai in terms of features and functionalities will be based on adoption requirements identified through integration efforts with new use-cases. Furthermore, AI-centric pipelines defined with tools such as Kubeflow⁵ will be continuously adopted in the course of the project. Integration with the workflow composition tool from Task 6.1 is currently in progress.

Other efforts include the integration of new use cases and refinement of use cases that have already been integrated. The integration of use case on Lattice QCD (T4.1) is currently in progress. The use cases of Radio Astronomy (T4.3) and High-energy Physics (second version, T4.2) are planned in the last period of the project. For already integrated use-cases, namely Virgo (T4.4), tropical cyclones detection and wildfires prediction (T4.5), Droughts early warning in Alpine region (T4.6), and Extreme events characterization (T4.7), additional feature enhancements provided by itwinai will be consolidated in the final versions.

⁴ <https://github.com/interTwin-eu/itwinai>

⁵ <https://www.kubeflow.org/>



3 Summary of integration status

The project's final year is dedicated to finalising use case integration. From the perspective of the core services, the situation is as follows.

OSCAR is evolving to address the arising needs with a focus on use case integration, improved notebook support, and data lake integration. DCNiOS, initially developed within the interTwin project to integrate dCache with OSCAR, has significantly evolved to support a myriad of data sources.

A particularly interesting integration case is with the openEO API library. As a recent development, a connection to the OSCAR event-driven triggering of workflows is in the making, so that events can trigger the execution of openEO process graphs. As a next step, a TOSCA template for this environment should be developed together with Task 6.4.

The Ophidia framework joins HPC paradigms and Big Data approaches at the user level. The latest release of the Ophidia component in particular, the Ophidia server and PyOphidia, cover two main features: support for CWL-based workflows and provenance tracking using yProv. The capabilities for CWL/JSON workflows will be further enhanced to better address use case tasks execution (i.e., support for non-Ophidia tasks). Moreover, integration with SQAaaS for validating workflow syntax is envisioned. yProv will be evolved during the last part of the project in order to accommodate additional requirements, such as provenance support in the SQA process;

From the model validation perspective, the strategy is to extend the current type of assessments (custom assessments) to deal with specific requirements on model validation and data quality of the existing use cases. In particular, those participating in T6.2 and T6.5. Finalization of the SQAaaS integration platform with the WfMS technologies being supported by T6.1, such as CWL. Supporting automated FAIR validation, event-based, on Data Lakes in the roadmap of the upcoming developments.

The Infrastructure Manager is the project choice for users to provision infrastructure in multi-cloud infrastructures. TOSCA templates enable the user needs description, in a cloud-agnostic way, of the virtual infrastructures needed in the available Big Data Analytics tools, and are used in conjunction with the Infrastructure Manager to self-provision resources. A number of them have been developed to support several usage models to construct Digital Twins. The work will continue following a similar strategy. Some of the templates are in an early stage (STAC and openEO) and need to be correctly tested by users with experience using these tools to validate the functionality of the deployed infrastructure.

itwinai is a Python library that streamlines AI workflows while reducing engineering complexity. Different interfaces are provided to lower the entry barrier for users coming from different fields of expertise: Python API, command line interface, and a high-level GUI. itwinai provides out-of-the-box state-of-the-art AI tools and encourages code reuse, to further simplify and streamline the development of ML workflows, on top of seamless integration with HPC resources. Future extensions to itwinai in terms of features and functionalities will be based on the adoption of requirements, identified



D6.4 Final release of the DTE core modules

through integration efforts with use cases. The next step is integrating itwinai with the LatticeQCD use case.



4 References

Reference	
No	Description / Link
R1	<p>interTwin D6.1 Report on requirements and core modules definition</p> <p>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.</p> <p>DOI. https://doi.org/10.5281/zenodo.10417153</p>
R2	<p>interTwin D6.2 First release of the DTE core modules</p> <p>Isabel Campos, Germán Moltó, Alexander Jacob, Pablo Orviz, Miguel Caballer, Matteo Bunino, Alexander Zoechbauer, Sandro Fiore</p> <p>DOI. https://doi.org/10.5281/zenodo.10224213</p>
R3	<p>interTwin D6.3 Updated report on requirements and core modules definition</p> <p>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.</p> <p>DOI. https://zenodo.org/records/13709618</p>

