



**D3.1**

# **AI Systems and Models Specification and Roadmap**

Status: Final

Dissemination Level: Public



Funded by  
the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement No 101188168.

## Abstract


### Key words

AI Computing Platform, Model Hub, Job Offloading, Provenance, AI in Environmental Sciences, AI in Life Sciences, Data Exploitation Platform

The Data Exploitation Platform (DEP) enables Research Infrastructures (RIs) to scale their AI applications across large-scale computing infrastructure, such as on cloud and High Performance Computing (HPC) systems, and enables scientists to train and/or run AI models at scale with RI scientific data. Work Package (WP) 3 plays a central role in enabling these capabilities by providing the technical solutions needed to integrate AI functionalities in the DEP.

This deliverable outlines the technical specifications of the AI solutions proposed in WP3 for the DEP, detailing their main features, planned developments and integrations. The document also presents user stories that illustrate scenarios of accessing the DEP for different kinds of users. These stories highlight the DEP access mechanisms and the role of the software solutions in the DEP. The AI applications and their compute and data requirements are also defined in this document. These requirements and the user stories guide the modular architecture of WP3, which enables flexible and customizable definition of workflows for DEP users. Finally, the document proposes the creation of testbeds, which form the methodological basis for realizing the technical implementations in the DEP.

Revision History			
Version	Date	Description	Author/Reviewer
V 0.1	14/07/2025	First Draft and Contributions	Rakesh Sarma (Juelich) Alex Krochak (Juelich) Daniele Spiga (INFN) Wei Ouyang (KTH) Sandro Luigi Fiore (UNITN) Saima Sharleen Islam(UNITN) Fabrizio Antonio (CMCC) Donatello Elia (CMCC) Teresa Zulueta-Coarasa (EMBL) Matthew Hartley (EMBL) Robert Harb (MUG) Carl-Fredrik Enell (EISCAT) Thomas Ulich (EISCAT) Miguel Santos (Neuraspace) Marta Guimarães (Neuraspace) Hakan Bayindir (TUBITAK) Ville Tenhunen (EGI) Gergely Sipos (EGI) Radoslava Kacová (MMCI) Tullio Degiacomi (HYP)
V 0.2	16/07/2025	First Draft ready for WP3 internal Revision	Rakesh Sarma (Juelich)
V0.3	23/07/2025	WP3 internal reviews provided to contributors	Rakesh Sarma (Juelich) Fabrizio Antonio (CMCC)
V0.4	28/07/2025	Refinements and quality control	Rakesh Sarma (Juelich)
V0.5	30/07/2025	Submitted to Reviewers	Rakesh Sarma (Juelich)
V0.6	22/08/2025	Submitted version to AMB and PO	Rakesh Sarma (Juelich)
V 1.0	29/08/2025	Final version for submission	Rakesh Sarma (Juelich)

Document Description			
D3.1 – AI Systems and Models Specification and Roadmap			
Work Package Number 3			
Document Type	Deliverable		
Document Status	Final	Version	1.0
Dissemination Level	Public		
Copyright Status	 <p>This material by Parties of the RI-SCALE Consortium is licensed under a <a href="https://creativecommons.org/licenses/by/4.0/">Creative Commons Attribution 4.0 International License</a>.</p>		
Lead partner	Juelich		
Document Link	<a href="https://documents.eqi.eu/document/4205">https://documents.eqi.eu/document/4205</a>		
DOI	<a href="https://zenodo.org/records/16993122">https://zenodo.org/records/16993122</a>		
Author(s)	<ul style="list-style-type: none"> <li>• Rakesh Sarma (Juelich)</li> <li>• Alex Krochak (Juelich)</li> <li>• Daniele Spiga (INFN)</li> <li>• Wei Ouyang (KTH)</li> <li>• Sandro Luigi Fiore (UNITN)</li> <li>• Saima Sharleen Islam (UNITN)</li> <li>• Fabrizio Antonio (CMCC)</li> <li>• Donatello Elia (CMCC)</li> <li>• Tullio Degiacomi (HYP)</li> <li>• Teresa Zulueta-Coarasa (EMBL)</li> <li>• Matthew Hartley (EMBL)</li> <li>• Robert Harb (MUG)</li> <li>• Carl-Fredrik Enell (EISCAT)</li> <li>• Thomas Ulich (EISCAT)</li> <li>• Miguel Santos (Neuraspace)</li> <li>• Marta Guimarães (Neuraspace)</li> <li>• Hakan Bayindir (TUBITAK)</li> <li>• Ville Tenhunen (EGI)</li> <li>• Gergely Sipos (EGI)</li> </ul>		
Reviewers	<ul style="list-style-type: none"> <li>• Thomas Geenen (ECMWF)</li> <li>• Alexandros Psychas (JNP)</li> <li>• Gergely Sipos (EGI)</li> </ul>		
Moderated by:	<ul style="list-style-type: none"> <li>• Matteo Agati (EGI)</li> </ul>		
Approved by:	Technical Coordination Board		

Terminology / Acronyms	
Term/Acronym	Definition
AI	Artificial Intelligence
BIA	BioImage Archive
CMIP	Coupled Model Intercomparison Project
CNN	Convolutional Neural Network
CRC	Colorectal Cancer
DEM	Digital Elevation Model
DEP	Data Exploitation Platform
DT	Digital Twin
EISCAT	EISCAT AB, formerly EISCAT Scientific Association
ENES	European Network for Earth System Modelling
ESGF	Earth System Grid Federation
DTE	Digital Twin Engine
GP	General Purpose
GPU	Graphical Processing Unit
HPC	High Performance Computing
HPO	HyperParameter Optimization
k8s	Kubernetes
ML	Machine Learning
MMCI	Masaryk Memorial Cancer Institute
MUG	Medical University Graz
RI	Research Infrastructure
SPE	Secure Processing Environment

SUC	Scientific Use Case
TRE	Trusted Research Environment
TRL	Technology Readiness Level
WP	Work Package
WSI	Whole-Slide Image

# Table of Contents

<b>Executive Summary.....</b>	<b>9</b>
<b>1. Introduction.....</b>	<b>10</b>
1.1. Scope of the Deliverable.....	11
1.2. Document Structure.....	11
<b>2. WP3 Architecture in DEP Landscape.....</b>	<b>12</b>
2.1. Data Exploitation Platform and User Stories.....	12
2.2. WP3 Architecture and its Purpose within DEP.....	16
<b>3. Technology Solutions in WP3.....</b>	<b>20</b>
3.1. itwinai.....	20
3.1.1. Component Status Overview.....	21
3.1.2. Identified Gaps and Developments planned in RI-SCALE.....	22
3.1.3. Planned integration with other components in WP3.....	23
3.2. interLink.....	24
3.2.1. Component Status Overview.....	25
3.2.2. Identified Gaps and Developments planned in RI-SCALE.....	26
3.2.3. Planned Integration with other Components in WP3.....	26
3.3. AI Model Hub.....	26
3.3.1. Component Status Overview.....	27
3.3.2. Identified Gaps and Developments planned in RI-SCALE.....	28
3.3.3. Planned Integration with other Components in WP3.....	29
3.4. BioImage.IO Chatbot.....	30
3.4.1. Component Status Overview.....	30
3.4.2. Identified Gaps and Developments planned in RI-SCALE.....	31
3.4.3. Planned Integration with other Components in WP3.....	32
3.5. yProv4ML.....	33
3.5.1. Component Status Overview.....	33
3.5.2. Identified Gaps and Developments planned in RI-SCALE.....	34
3.5.3. Planned Integration with other Components in WP3.....	34
<b>4. AI Applications in WP3.....</b>	<b>35</b>
4.1. Environmental Applications.....	35
4.1.1. SUC 1: High-resolution Downscaling of Climate Scenarios and Risk Trend Analysis in Agriculture.....	35
4.1.2. SUC 2: Smart Detection of Anomalies in Climate Data Usage.....	37
4.1.3. SUC 3: Intelligent Scheduling of Radar Observations.....	38
4.1.4. SUC 4: Space Debris and Anomaly Detection.....	40
4.2. Health and Life Science Applications.....	41
4.2.1. SUC 5: Colorectal Cancer Prediction with explainable AI.....	41
4.2.2. SUC 6: Synthetic Data for Computational Pathology.....	42
4.2.3. SUC 7: Foundational Models for Heterogeneous Biological Image Data.....	43

4.2.4. SUC 8: Generative AI-Powered Assistant for Data Discovery and Analysis.....	44
<b>5. Test and Integration Pilots.....</b>	<b>46</b>
5.1. Early Demonstrators and Ongoing Testbed Integrations.....	46
5.2. Roadmap for Future Setups.....	47
<b>6. Summary and Next Steps.....</b>	<b>49</b>
<b>References.....</b>	<b>51</b>
<b>Annexure.....</b>	<b>54</b>

## Table of Figures

- [Figure 1: DEP Architecture including all Components in RI-SCALE, with the "AI Lifecycle Management" Container highlighted with the Red Rectangular Box](#)
- [Figure 2: WP3 Architecture elaborating the Components in WP3 and their Dependencies](#)
- [Figure 3: Integration of AI Agents for the DEP in SUC 8](#)



# Executive Summary

The Data Exploitation Platform (DEP) enables Research Infrastructures (RIs) to scale their AI-based and data-driven applications on large-scale computing infrastructure and enables training and/or inference of AI models at scale with RI scientific data. Work Package (WP) 3 delivers the essential tools and solutions within the DEP for the AI functionalities. This deliverable provides the technical specifications of the software solutions that have been identified in the initial phase of the project. The specifications include a summary of the primary functionalities, the gaps in the technologies and the integrations that are planned among these components.

The DEP is formed by multiple containers developed by the technical WPs and use case providers in RI-SCALE. In this document, the modular architecture of the components in WP3 is presented and their interaction with the other containers in the DEP is elaborated. The WP3 architecture is based on a modular and flexible framework, which allows users to define their custom workflows. The computing framework in this architecture allows access to cloud and High Performance Computing (HPC) resources. The design of the architecture is complemented by user stories that describe scenarios for different kinds of DEP users and how they will access the DEP for their AI-centric data analysis workflows.

Besides, this document also presents the AI applications themselves and their compute and data requirements. These are used to formulate the specifications of the technical solutions that are proposed in WP3. Furthermore, to implement these AI solutions and the integrations, the document also highlights the methodology adopted in the form of testbeds. These testbeds or pilots provide the basis for development efforts and guide the roadmap to achieve the required technological solutions.

This document provides the foundation for the development and implementation of the AI-based services in the DEP, ensuring that the technical solutions meet the requirements of the RIs. Also, the document provides the DEP release timeline when each requirement shall be delivered, thereby supporting a coordinated implementation of the technical solutions.

# 1. Introduction

Artificial Intelligence (AI) technologies provide one of the core functionalities to the DEP in the RI-SCALE project. They provide RIs with scalable platforms for data exploration and utilization, enabling discovery and continuous learning. In this deliverable, details are provided on the AI technological solutions that will be developed and/or extended during RI-SCALE. The AI solutions will be developed by different associated partners in the project, with many of the developments based on outcomes of previous European and national projects.

The solutions will provide the necessary tools and technology for enabling AI-based functionalities in the DEP. This document also provides the functional and non-functional WP3 requirements in the [Annexure](#) (based on identified requirements in D5.1 - Data Exploitation Platform Requirements and Design Considerations [R1]), specifying the DEP release timeline when each requirement will be delivered. These technologies serve as the backbone for the “AI Lifecycle Management” container in the DEP (shown in [Section 2](#)). The primary functionalities provided by this container are:

- AI Computing Framework;
- Support for distributed training and inference;
- AI Model Hub;
- AI training/inference offloading;
- Hyperparameter optimization (HPO);
- Profiling and Performance Benchmarking;
- Provenance tracking;
- AI metrics logging;
- User Interfaces: Chatbot, Jupyter-like solutions.

The deliverable also discusses the AI applications involved in WP3 in Tasks 3.3 and 3.4. These tasks contribute to the scientific and technical use cases in WP5, which form the validation backbone for the DEP. WP3 focuses on developing AI applications in the environmental, health and life sciences domains, encompassing model development, training, inference, and deployment across the full MLOps lifecycle. In WP3, the AI software solutions developed in Tasks 3.1 and 3.2 work together with the applications to support all the steps to enable the Machine Learning (ML) workflows. The deliverable also discusses the data and compute requirements from the use-case perspective. These requirements guide the technical capabilities that the software solutions need to provide in order to drive the use case development. This is hence critical to the realization of the scientific and technical use cases in WP5.

To meet the RI-SCALE requirements, integrations among the AI solutions have to be developed during the project. The deliverable provides a summary of already achieved and planned integrations. Furthermore, testbeds will be used to implement and validate these integrations. These testbeds will support integrations not only among the technical solutions, but also across all the AI applications, ensuring interoperability and directly contributing to the realization of the Scientific Use Cases (SUCs).

Finally, the deliverable also discusses the overall architecture of the “AI Lifecycle Management” container within the DEP landscape. The components within this container are discussed in detail and their internal interactions are provided. Also, the workflows involving DEP developments from WP2 and WP4 are explored in this architecture. In practice, this will drive the entire pipeline of the DEP.

## 1.1. Scope of the Deliverable

This deliverable provides the specifications of the AI technical solutions, architecture and applications in WP3. The presented architecture illustrates the workflow between WP3 components and also the interactions with the other components in the DEP, which are developed in WP2 and WP4. Overall, the identified architecture serves as a blueprint for designing and implementing the AI-based workflows in the DEP. For the software solutions, the document describes their current status, the technological gaps and the planned integrations with the AI applications, which contribute to the SUCs. The deliverable provides a roadmap to realize these integrations. A conceptual DEP workflow (involving WP2 and WP4) is discussed; however, low-level details will become clearer in the later phases of the project.

## 1.2. Document Structure

The document is structured as follows: The WP3 architecture is presented in [Section 2](#), showing the workflow within WP3 and also to other containers in the DEP. User stories are provided to show DEP access scenarios. In [Section 3](#), details on the technology solutions developed in WP3 for delivering the AI capabilities in the DEP are provided. The AI applications are elaborated in [Section 4](#) along with their compute and data challenges. The proposed testbeds that will guide the WP3 methodology are presented in [Section 5](#). Finally, the main summary of the document and future steps of WP3 are provided in [Section 6](#).

## 2. WP3 Architecture in DEP Landscape

In this section, the WP3 architecture is presented, along with its contribution to the overall DEP landscape. In [Section 2.1](#), the functionalities offered by the DEP platform are elaborated in terms of the user interactions, by introducing different kinds of users and their goals of accessing the DEP. Based on these users, multiple user stories are presented in the form of scenarios. These describe in detail the steps and software components that are required to enable these scenarios. Then, in [Section 2.2](#), the overall WP3 architecture is shown, which takes into account the presented user stories.

### 2.1. Data Exploitation Platform and User Stories

DEPs enable RI data holdings to expand their services with ‘online data analysis’, by partnering with external compute facilities where data is replicated and served for user analysis.

The challenges and limitations RIs face that DEP is designed to address are:

1. **Lack of on-site compute:** Limited compute and storage provisioning at RI data holding sites hinders data quality control, data product improvement (incl. FAIR-ification) and the widespread uptake of data by the broader user community for analysis;
2. **Large data downloads and data management:** Large datasets are cumbersome and time-consuming to download for an individual researcher; moreover, separate storage and compute systems may use different access control mechanisms;
3. **Complex software:** Installing and configuring software stacks for running environments for data science (e.g., with AI, Digital Twins (DTs), Trusted Research Environments (TREs) or Secure Processing Environments (SPEs)) presents a major barrier to users.

A DEP fundamentally acts as an extension of an RI, a new service that is connected to the RI data holding and offers online data analytical services for data processing. In the RI-SCALE project, DEPs are specifically designed to support AI-based analysis.

The main connections and main users of a DEP are:

1. The **end user** of a DEP is the main beneficiary. They want to explore and browse the relevant datasets from an RI to perform data analysis, replicate this data from the holding to the compute facility that operates the DEP environment, and choose a pre-configured, pre-trained AI model to analyse the data with inference runs. Typically, end users have direct access to the DEP, and they are authorised RI users who run the models on data and share the outputs with other authorised users;

2. **Model developers** create and deploy new AI models within the DEP. They either use off-the-shelf 3rd party models, or develop their own models, then train the models with RI data, and share the validated models via the DEP with the end users;
3. **DEP operator** deploys, configures and operates the DEP environment within a compute centre, and ensures its proper connections to external systems. These connections include links to the data holding(s), to AI-model stores and to identity management systems that are supported by the specific DEP installation.

Based on the project objectives, the main goals of the DEP are:

- Replicate and manage copies of big scientific data from RI repositories and Data Spaces to and on high-performance and cloud compute resources;
- Facilitate the use of AI applications for scalable data analysis;
- Support real SUCs with big scientific data and AI applications;
- Enable seamless access to users to resources and services across the entire value chain;
- Track and report resource and service consumption during the entire usage workflow;
- Increase the AI-based data exploitation and data mining capacity and resources of RIs.

Requirements defined in project deliverable D5.1 [R1] set numerous principles for the DEP, which are linked with these goals. These requirements also describe high-level activities which aided in the identification of the three user groups described above, and what they are expected to do with the DEP.

The activities for the **end user** are:

- Explore and flag relevant datasets from a research infrastructure or data space;
- Explore and choose a pre-configured and pre-trained AI model to analyse the data;
- Perform data analysis on the RI data;
- Export/share/use the results of data analysis.

Activities for **model developers** are of two types. Firstly, there are activities which are linked with new AI model development, and secondly, there are activities with existing models.

The activities with new models are to:

- Create a new AI model;
- Deploy the new AI model for training;
- Train a new AI model with RI data;
- Validate new model accuracy;
- Share the new validated model in one or multiple DEP(s).

The activities with existing models are to:

- Select an existing model for retraining;
- Associate the existing model and the training data;
- Train an existing or 3rd party AI model with the data;
- Validate an existing model's accuracy;
- Share the old validated model in one or multiple DEP(s).

The third identified user of the DEPs is the **DEP operator**. This role is responsible for the following activities:

- Deploy, configure and operate the DEP environment within the compute centre;
- Ensure DEP environment connection to external systems (AAI, AI model stores, data holdings, etc.);
- Ensure DEP's infrastructure availability and continuity;
- Manage DEPs' infrastructure incidents and service requests;
- Ensure infrastructure capacity for DEPs;
- Report DEP's resource usage.

For the various kinds of users identified above, the following sections elaborate on user stories focused on AI model development, operation or use for data analytics and other applications. These stories or scenarios build up the functionalities that need to be provided by the DEP. Here, specifically, the AI technologies that are developed in WP3 to provide solutions to the users are specified. It is important to mention that the examples are non-exhaustive and they are intended to provide instances where these functionalities are exploited through the DEP. The DEP responses are guided by the WP3 architecture described in [Section 2.2](#). Further, the WP3 software solutions (itwinai, interLink, yProv4ML, AI Model Hub) that are referenced in the scenarios below are described in [Section 3](#). These scenarios also involve DEP components from WP2 and WP4, which are elaborated in [D2.1](#) and [D4.1](#).

*Scenario 1 (End user): I want to use existing AI model(s) to analyze/benchmark my selected dataset from the RI.*

Steps	DEP response
Login to DEP with AAI-monitored credentials	Authenticates user via AAI (from WP4) and provides role-based access
Pull, search and filter, and flag one or more datasets for analysis	WP2 components (RUCIO and other developments) make the required data available from the cached source or newly replicated from RI to the compute center

Browse the catalogue of available AI model(s) in Model Hub and select the required model	AI Model Hub provides an interface to browse models released by developers and sends the required model(s) to the runtime environment provided by itwinai and interLink
Configure model inference parameters and execute the model run	itwinai provides an inference pipeline definition interface. This is provided to interLink, which launches the job in the compute facility, if user credit is available (given by AAI)
Analyze/benchmark, visualize and export results	Visualize and benchmark model performance, show training metrics and provenance information through itwinai/MLflow/yProv4ML through various interfaces. Export options include storage solutions provided by WP2

*Scenario 2 (End user): Reuse or reanalyze previously analyzed model and data*

Steps	DEP response
Login to DEP with AAI-monitored credentials	Authenticates user via AAI and provides role-based access
Reload some dataset and model	RUCIO makes the required data and model available from cache or short-term storage
Modify model inference parameters	itwinai provides model definition functionalities in the form of configuration files, where inference parameters can be edited
Run the defined pipeline and visualize the results	The defined pipeline is launched on the chosen infrastructure by interLink

*Scenario 3 (Model Developer): Develop and deploy a new AI model, and perform HyperParameter Optimization (HPO)*

Steps	DEP response
Login as a DEP user in the development model	AAI grants access to the development environment
Load required modules, tools and access training data	itwinai provides a development interface (e.g. Jupyter-like interfaces, CLI, etc.) in containerized or Python virtual environment; RUCIO provides access to the required training data for preprocessing and training
Build and train a new AI model, select a distributed training strategy, perform HPO and validate the model	itwinai provides training/validation definition to interLink, which launches the training/HPO job on the selected compute center, when credit is available (provided by AAI)
Package and register the final model	Registers the model in AI Model Hub

Share the model with users	AI Model Hub publishes a model with access control and versioning
----------------------------	---

*Scenario 4 (DEP Operator): Check provenance information, track resource usage, and compile performance statistics*

Steps	DEP response
Collect provenance logs and usage metrics	yProv4ML/itwinai provides provenance information, profiling, energy and performance metrics
Link usage to a specific project or users	Subject to privacy considerations, AAI provides job-specific user information
Generate performance and usage reports	Generates reports (e.g. profiling provided by itwinai) for administrative purposes and dissemination to stakeholders
Highlight bottlenecks or under-utilized resources	Administrators use information to advise developers to improve the performance of identified bottlenecks

These scenarios and the requirements gathered in deliverable D5.1 [R1] provide the basis for the development of the software solutions in WP3.

## 2.2. WP3 Architecture and its Purpose within DEP

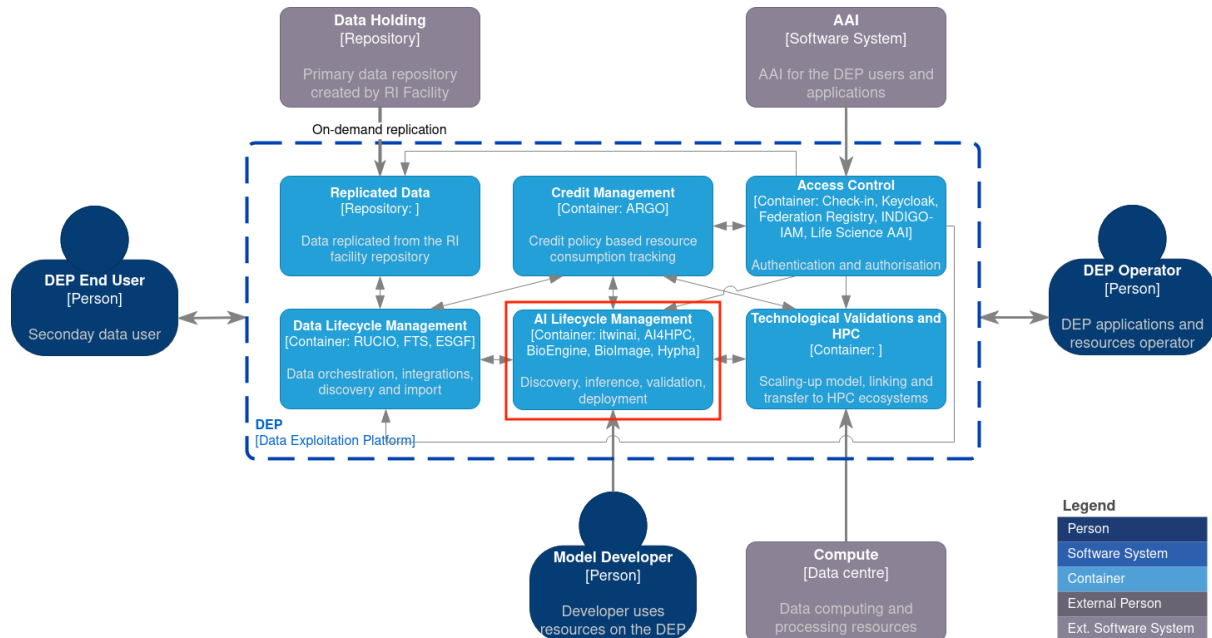
Here, the WP3 architecture is discussed in detail in a C4<sup>1</sup> model-based diagram. [Figure 1](#) shows the overall DEP architecture in the RI-SCALE project. As can be seen, the DEP consists of components provided by WP2, WP3 and WP4, which are:

- **Replicated Data (WP2):** The data that is replicated from the data holdings at the RIs is accessible through this container or repository;
- **Data Lifecycle Management (WP2):** This is the primary data orchestration and integration service, which allows other containers to access the replicated RI data;
- **Credit Management (WP4):** The user resource consumption on the compute and data centers is managed by this container;
- **Access Control (WP4):** This container provides the authentication mechanism for accessing the computing infrastructure;
- **AI Lifecycle Management (WP3):** This container provides the AI tools that will be used by the use-cases to run their ML workflows. This container is further discussed in detail below;

<sup>1</sup> C4 model: <https://c4model.com/>, accessed on 15.08.2025



- **Technological Validations and HPC:** The technological validations are carried out in the compute and data centers associated with the project. Besides, provisioning of EuroHPC resources is also expected to run some use cases.

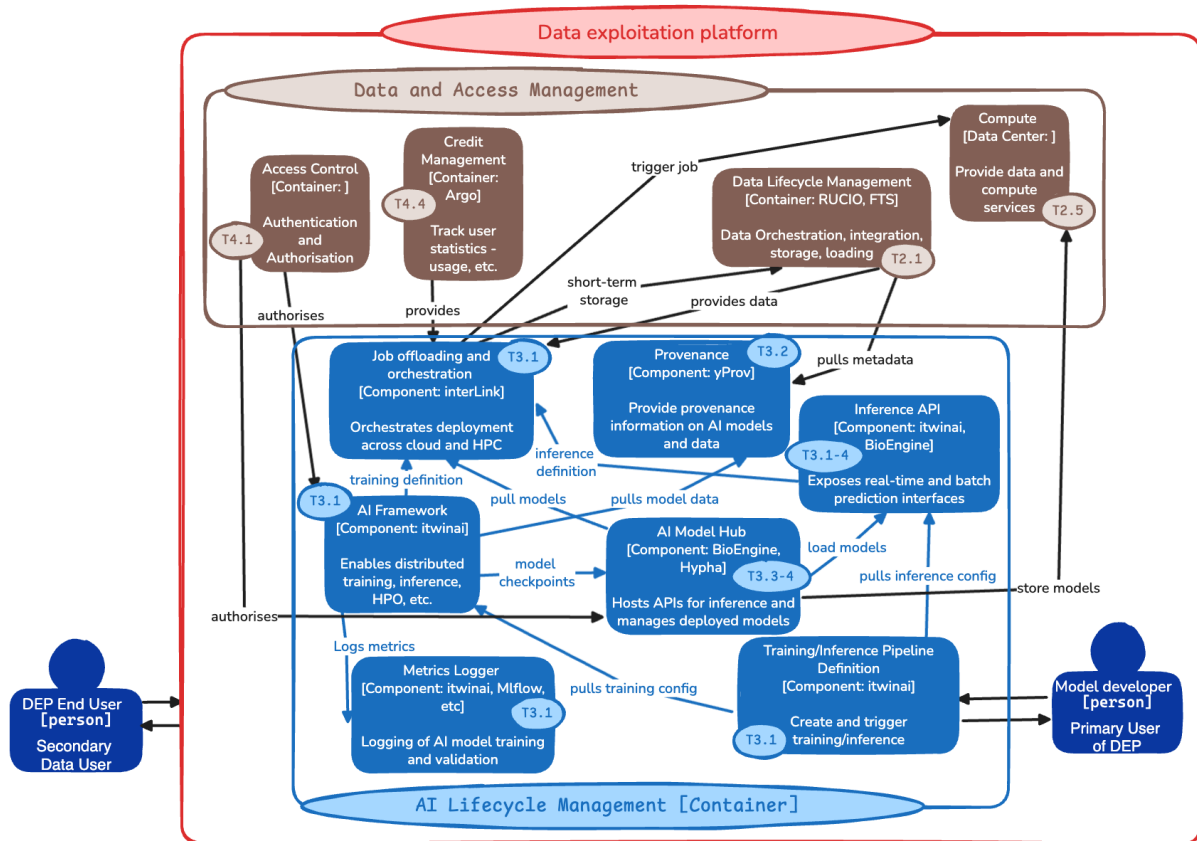


**Figure 1:** DEP Architecture including all Components in RI-SCALE, with the “AI Lifecycle Management” Container highlighted with the Red Rectangular Box

The detailed component view of the “AI Lifecycle Management” container is shown in [Figure 2](#). In the course of the project, necessary changes to this architecture might be adopted. The main components, their features and interdependencies that are envisioned are elaborated below:

- **AI Framework (itwinai):** This component is provided by itwinai (elaborated in [Section 3.1](#)), with features such as distributed training, HPO, etc. It provides model data to the “Provenance” component, logs metrics to “Metrics Logger” and creates model checkpoints in the “AI Model Hub”.
- **Job offloading and abstraction (interLink):** This component abstracts the offloading and triggering of ML training and inference workflows in WP3 to “Compute”, such as cloud and HPC infrastructure. Further details are provided in the definition of interLink in [Section 3.2](#). This component needs to be authorised by the “Access Control” container. Components such as “AI Model Hub”, “AI Framework”, and “Inference API” provide definitions for the jobs to be launched. For example, these definitions could also include storage of data through the “Data Lifecycle Management” container.
- **AI Model Hub (BioEngine, Hypha):** This component is provided by BioEngine and Hypha (defined in [Section 3.3](#)), which provides the model serving platform and deployment. It provides the required models to the “Inference API” component and stores models on the “Compute” container.

- **Provenance** (yProv4ML): This component (described in [Section 3.5](#)) provides provenance information on the ML models. It receives data from the “AI Framework” component and “Data Lifecycle Management” container.
- **Metrics Logger** (itwinai, Mlflow, yProv4ML): This component is provided by a collection of different technological solutions, which enables a user to choose a logger of their choice. It logs various ML metrics, not only from the model perspective, but also looking at energy consumption, GPU utilization, etc.
- **Inference API** (itwinai, BioEngine): This component provides the necessary interfaces to launch inference pipelines. It loads models from the “AI Model Hub” and the inference definition from the “Training/Inference Pipeline Definition”.
- **Training/Inference Pipeline Definition** (itwinai): This component provides the configuration to launch training and inference of ML models. It provides a centralised solution to define hyperparameters, workflow steps and other job definitions.



**Figure 2:** WP3 Architecture elaborating the Components in WP3 and their Dependencies

In the context of the users, the “Model developer” directly interacts with the pipeline definition component to define their ML workflows. The “DEP End User” interacts with the DEP as a whole through an interface, which will be defined over the course of the project.

This architecture will evolve depending on the project requirements. In particular, to address the needs and developments from WP2 and WP4, the interactions will be further refined after discussions. Given the overall modular structure of the workflow and the components themselves, the needs of RIs can be incorporated for each component in WP3.

## 3. Technology Solutions in WP3

WP3 brings together a set of complementary technology components designed to provide a coherent foundation for deploying, executing, and managing AI-driven scientific workflows within the DEP. These components span the full lifecycle of AI applications, from workflow orchestration and scalable computation (itwinai, interLink) to model hosting, discovery, and execution (AI Model Hub, BioEngine), and from interactive, user-facing interfaces (BioImage.IO Chatbot) to provenance capture and reproducibility (yProv4ML). While each component addresses distinct technical challenges, they are designed to interoperate through standardised APIs, shared metadata models, and common deployment patterns, enabling them to be combined into end-to-end workflows. Together, they form a flexible and modular toolkit capable of supporting diverse scientific domains, while remaining adaptable to the evolving requirements of RI-SCALE use cases and the DEP architecture.

This section provides an overview of the identified technology solutions in WP3. Each component is briefly introduced with links to the source code and documentation, and component features. The identified gaps in the technology and developments planned within RI-SCALE are also provided. Finally, the planned integrations with other components in WP3 are presented.

### 3.1. itwinai

itwinai<sup>2</sup> is an open-source Python-based toolkit that provides a wide range of functionalities intended to accelerate AI and ML workflows. Although developed as the core module of a Digital Twin Engine (DTE) in the interTwin project<sup>3</sup> to deploy DTs for various scientific applications, itwinai is versatile and can support any generic ML application. However, as the library is developed to assist scientists in minimizing their AI development effort and tested across a multitude of scientific applications, it is suitable in the context of the RI-SCALE project. Itwinai also includes detailed documentation and allows easy adoption for new users, ensuring ease of deployment on diverse computing environments.

Itwinai has already been ported to and tested on various Tier 0/1 HPC and EuroHPC systems across Europe. This includes, besides others:

- LUMI supercomputer<sup>4</sup>,
- JUWELS Booster system<sup>5</sup>,
- Vega supercomputer<sup>6</sup>, and

<sup>2</sup> itwinai <https://itwinai.readthedocs.io/latest/>

<sup>3</sup> interTwin <https://www.intertwin.eu/>

<sup>4</sup> LUMI <https://www.lumi-supercomputer.eu/>

<sup>5</sup> JUWELS Booster <https://apps.fz-juelich.de/jsc/hps/juwels/booster-overview.html>

<sup>6</sup> Vega <https://izum.si/en/vega-en/>

- JUPITER Booster<sup>7</sup>, the first exascale supercomputer in Europe.

Furthermore, it has also been tested on cloud infrastructure, such as the JSC Cloud<sup>8</sup>, allowing users to choose among their preferred infrastructure.

### 3.1.1. Component Status Overview

The library is already at an advanced stage in its development process and is suitable for both advanced and new AI practitioners.

The main functionalities provided by itwinai are:

- **Distributed training and inference:** itwinai supports various distributed training frameworks such as PyTorch DDP<sup>9</sup>, Horovod<sup>10</sup>, DeepSpeed<sup>11</sup>, and TensorFlow MultiWorkerMirroredStrategy. Each of these frameworks can perform differently depending on the dataset, model, and infrastructure. The intention behind providing the user with this choice is to allow benchmarking of their use cases with different frameworks. A user can simply specify the required strategy in the input configuration file to deploy their model.
- **Logging:** itwinai provides built-in support for MLflow<sup>12</sup>, Weights & Biases<sup>13</sup>, and TensorBoard<sup>14</sup>. Similarly as is the case for the strategy above, the user can choose their preferred logger(s) in the input configuration file.
- **Modular workflows:** Users can easily plug and play components in their ML workflow definition in itwinai. For instance, users can replace and/or customize one or more steps in the ML pipeline definition, while still exploiting the features provided by the library.
- **Profiling AI training and inference:** In large-scale AI training and inference, tracking performance is essential in order to improve efficiency and to enable better utilization of compute resources. itwinai provides built-in support to profile ML runs and easily track lines in the code which lead to major bottlenecks. This was especially useful in the context of a use case in interTwin, where an ML training run achieved a speed-up of about 70% with the help of the profiler provided by itwinai.
- **HPO:** itwinai provides built-in support to perform HPO, which is enabled with the Ray<sup>15</sup> framework. Computationally, large-scale HPO is only possible on HPC resources by performing individual HPO trials on different workers. In itwinai, users can parallelize individual trials using data-parallel distributed training for individual trials.

<sup>7</sup> JUPITER Booster <https://www.fz-juelich.de/en/ias/jsc/jupiter/tech>

<sup>8</sup> JSC Cloud <https://apps.fz-juelich.de/jsc/hps/jsccloud/index.html>

<sup>9</sup> PyTorch DDP [https://docs.pytorch.org/tutorials/intermediate/ddp\\_tutorial.html](https://docs.pytorch.org/tutorials/intermediate/ddp_tutorial.html)

<sup>10</sup> Horovod <https://github.com/horovod/horovod>

<sup>11</sup> DeepSpeed <https://github.com/deepspeedai/DeepSpeed>

<sup>12</sup> MLflow <https://mlflow.org/>

<sup>13</sup> Weights & Biases <https://wandb.ai/site/>

<sup>14</sup> TensorBoard <https://www.tensorflow.org/tensorboard>

<sup>15</sup> Ray Tune <https://docs.ray.io/en/latest/tune/index.html>

- **Plugins:** itwinai also allows users to develop their use cases in plugins, which allow independent development of use cases. In the interTwin project, this has been widely used across a multitude of use cases. A list of the currently available plugins can be found on the documentation page<sup>16</sup>.
- **Support for containerized execution:** The itwinai packages are available in the form of Docker container images, allowing users to run it in Docker environments, both for production and deployment. This has already been useful in the context of many use cases, especially for the work with interLink (more details in the next section).

User Documentation: <https://itwinai.readthedocs.io/latest/>

Source code: <https://github.com/interTwin-eu/itwinai/tree/main>

Licence is Apache 2.0

### 3.1.2. Identified Gaps and Developments planned in RI-SCALE

In RI-SCALE, the itwinai package will be extended to include functionalities to allow the use cases to deploy their scientific applications. Based on the requirements gathered in deliverable D5.1 [R1], the gaps in technological solutions that need to be further provided by itwinai are identified. Based on these gaps and also the long-term vision for the itwinai package, the following developments are planned during the RI-SCALE project. This is non-exhaustive, and further features will be added depending on the scientific and technological requirements.

- **Support for model parallelism:** At present, itwinai primarily supports data-parallelism. With the DeepSpeed framework, certain model parallelism is possible, but this is somewhat limited. Since large-scale models such as for training foundational models could potentially require distribution among workers, development of model parallelism features is also planned in itwinai.
- **Support for training and fine-tuning foundational models:** The SUCs in RI-SCALE are also working on training and fine-tuning foundational models. Support to enable these developments will be extended in the itwinai package such that tailored modules for such model training and inference are available.
- **Development of User Interface:** In line with DEP architectural requirements, the user interface for itwinai will further be tailored to project needs. Based on further discussions within the technical work packages, this feature will be developed.
- **Integration with RI-SCALE compute and data providers and other EuroHPC sites:** itwinai will be integrated with the sites at TUBITAK and TU Wien to allow the development of use cases. Furthermore, depending on access (for example, with DestinE provided resources), other EuroHPC sites will also be integrated.

---

<sup>16</sup> itwinai plugins <https://itwinai.readthedocs.io/latest/getting-started/plugins-list.html>

- **Logging integration in DEP user management:** Additional features in the logging tool will be integrated, such as providing user-specific access control to logs.
- **Inclusion of additional HPO features and automation:** Depending on use-case requirements, additional features in the HPO module of itwinai will be developed, in particular with respect to automation of the trial runs.
- **Advanced post-processing routines and advanced visualization tools:** Further post-processing and visualization tools will be supported during the course of the project.
- **Testing of GROQ cards:** In order to support GROQ cards, itwinai will provide the necessary software layers to enable their deployment.
- **Compliance with data management and access solutions:** This feature development will be in collaboration with WP2 and WP4 to ensure proper functioning of the DEP.

### 3.1.3. Planned integration with other components in WP3

The main integrations of itwinai with other technical solutions identified in WP3 are:

- **interLink:** During the interTwin project, itwinai and interLink have already been integrated, where scientific applications in the project were tested on HPC systems, such as at the Vega supercomputer. In RI-SCALE, these integration test pilots will be utilized for deploying the use-cases on the computing infrastructures in the project through interLink.
- **AI Model Hub:** The integration of itwinai with the AI Model Hub, along with the BioEngine, will allow workflows defined in the itwinai configuration files to pull models and launch them on the selected infrastructure through the Inference API that will be developed during the project. Furthermore, the trained models will also have access to the Model Hub to create checkpoints and provide model releases.
- **BioImage.IO Chatbot:** The **BioImage.IO** Chatbot provides a state-of-the-art user experience for running inference on ML models, which has already been demonstrated in the use cases on life sciences. For itwinai, integrations will be planned to potentially allow exposing the itwinai frontend to the Chatbot. The details will be clearer over the course of the project.
- **yProv4ML:** Early demonstrators on integration of yProv4ML with itwinai have already taken place during the interTwin project. This will be further enhanced in the RI-SCALE project.

Other than that, the integrations with the AI applications (defined by Tasks 3.3 and 3.4) are envisioned by the development of test pilots (see [Section 5](#)).

## 3.2. interLink

interLink is an open-source service to enable transparent access to heterogeneous computing providers. It provides an abstraction for the execution of a Kubernetes pod on any remote resource capable of managing a Container's execution lifecycle.

The aim is to provide an open-source solution capable of extending the container orchestration de facto standard (Kubernetes) to support offloading to any type of resource provider (Cloud/HTC/HPC) transparently, where little to no knowledge is required by the end user. The key objective of interLink is to enable a Kubernetes cluster to send containers/pods to a "virtual" node. This node seamlessly manages the entire lifecycle of the user's applications, whether on a remote server or, preferably, within an HPC batch queue.

From a technical perspective, the interLink component extends the Kubernetes Virtual Kubelet solution with a generic API layer for delegating pod execution on ANY remote backend. Kubernetes Pod requests are digested through the API layer (e.g. deployed on an HPC edge) into batch job execution of a container.

The architecture is plugin-based, with a dedicated plugin for each supported backend.. For each API (VERB), the plugins perform specific operations based on what the actual backend is. Submitting a Pod to the cluster means the plugin will receive from interLink the list of all related Secrets, ConfigMaps, EmptyDirs and the description of the Pod itself. Utilizing this information, the plugin takes specific actions accordingly. Each plugin accepts the three standard outgoing calls described above the interLink API.

The plugins currently available are:

- **interlink-slurm-plugin:** A GO-based plugin to connect the Slurm-managed batch system to interlink;
- **interlink-kueue-plugin:** A Container plugin to connect Kueue to interlink;
- **interlink-htcondor-plugin:** A Python-based plugin to connect HTCondor-CE to interlink;
- **interlink-docker-plugin:** A Python-based plugin to connect any system with docker engine to interLink;
- **interlink-unicore-plugin:** A Python-based plugin to connect UNICORE API to interlink;
- **interlink-arc-plugin:** A Python-based plugin to connect the ArcCE gateway to interLink;

At the time of writing, interLink has already been integrated with several frameworks running on top of Kubernetes (k8s), including itwinai, in order to successfully exploit large-scale systems such as EuroHPC. In terms of computing sites, interLink has already been deployed in:

- **VEGA:** The first of eight peta and pre-exa-scale EuroHPC hosted in Slovenia.



- **Juelich:** JSC provides seamlessly integrated cloud and HPC resources through UNICORE middleware using interLink.
- **PSNC:** Provides access to a world-class e-Infrastructure for the scientific community, a specific research and development environment. In terms of plugin, this integration uses the SLURM one.
- **KBFI:** The cluster consists of around 8000 compute cores and a distributed storage facility with 3.8 PB of raw disk capacity, where interLink interacts with compute resources via the ARC-CE Compute model.
- **Leonardo at CINECA:** Both Booster and General Purpose (GP) Partitions are exploited via interLink deployed at the edge running with the Slurm plugin.

### 3.2.1. Component Status Overview

The development status of interLink is already at an advanced stage and has demonstrated the necessary flexibility. On one hand, it supports a variety of backends to offload computation; on the other hand, it is able to integrate distinct types of high-level services, ensuring their compatibility with a k8s-based solution for provisioning computational resources.

In summary, the main functionalities provided by interLink are:

- **Offload Kubernetes applications with tasks to be executed on HPC systems:** This feature focuses on Kubernetes applications that require HPC resources for executing tasks (AI training and inference, ML algorithm optimizations, etc.). These tasks might involve complex computations, simulations, or data processing that benefit from the specialized hardware and optimized performance of HPC systems.
- **Remote "runner"-like application for heavy payload execution requiring Graphical Processing Units (GPUs):** interLink is designed for applications that need to execute heavy computational payloads, particularly those requiring GPU resources. These applications can be run remotely, leveraging powerful GPU hardware to handle tasks such as model training, data analysis, or rendering.
- **Designed to ease the work required to include new remote providers:** interLink is designed to simplify the integration of new remote providers. Extending beyond HPC+SLURM involves creating simple web servers in the preferred language, where the provider can decide the proper way of managing the container execution lifecycle.

User documentation: <https://interlink-project.dev/>

Developer documentation: <https://interlink-hq.github.io/interLink/docs/Developers>

Governance: <https://github.com/interlink-hq/interLink/blob/main/GOVERNANCE.md>

Source code: <https://github.com/interlink-hq/interLink>

Licence is Apache 2.0

### 3.2.2. Identified Gaps and Developments planned in RI-SCALE

In RI-SCALE, the interLink component will be extended to include functionalities needed to support new and advanced use cases needed to enabling scientific applications to effectively exploit computing capacity. In particular:

- **Networking adapters:** Enabling interLink, creating a network mesh (at the user namespace level) with the internal Pod overlay. It represents a possible game changer for hybrid clusters to support off-the-shelf AI frameworks.
- **AuthN/Z** interLink already integrates JWT-based flows. Building on top of this, a more fine-grained (group/scope-based) authorization mechanism is a task for development.
- **Data Management / Data Access:** Effective data access is a key to the efficient usage of resources. InterLink can offer handles to enhance the integration with data orchestration systems and input data caching mechanisms.
- **Monitoring and Accounting:** Based on current experience, a solid system to track any action executed on target providers via interLink is mandatory. Evolving the current implementation is key to a higher Technology Readiness Level (TRL).

Those developments will be driven by specific needs, and priorities will be adjusted based on requirements from scientific communities.

### 3.2.3. Planned Integration with other Components in WP3

The goal of interLink integration is to support any of the WP3 AI frameworks that run on top of k8s in order to successfully exploit computing resources not necessarily available locally.

- **itwinai:** The existing integration with itwinai will be exploited for this, where interLink will be enhanced following the requirements of the community using itwinai to define and manage AI pipelines.
- **AI Model Hub/BioEngine:** The AI Model Hub is already containerized and implements a k8s resources provisioning model. As such, it will be integrated and supported.

**AI Applications:** Other than that, the integrations with the AI applications (defined by Tasks 3.3 and 3.4) are envisioned by the development of test pilots.

## 3.3. AI Model Hub

The **AI Model Hub** is the central WP3 service for hosting, discovering, and executing AI models within the DEP. The model hub will be built based on existing work in the **AI4Life** project<sup>17</sup> and closely

<sup>17</sup> <https://ai4life.eurobioimaging.eu/>

integrated with the **BioImage Model Zoo**<sup>18</sup>. By integrating with the **Hypha**<sup>19</sup> framework, it integrates persistent storage, scalable execution, and lightweight application interfaces into a single platform. By consolidating these functions, it ensures that models are preserved with rich metadata, easily searchable, and directly usable in scientific workflows without extra deployment steps.

At its core, the **Artifact Manager**<sup>20</sup> stores models as versioned artifacts in S3-compatible object storage, with metadata indexed in a SQL database for fast, structured search. Metadata fields capture descriptive, technical, and provenance details, enabling precise queries by parameters such as model name, version, author, dataset, or license. Integrated access control ensures that sensitive models are only available to authorised users. For hosted execution, the hub connects to **BioEngine**<sup>21</sup>, a Ray-based distributed backend that retrieves models from the Artifact Manager and runs them in isolated, containerised environments. This setup guarantees reproducibility, supports CPU and GPU acceleration, and scales across local servers, HPC systems, or Kubernetes clusters. BioEngine exposes standard APIs for integration with desktop tools, notebooks, and other DEP services. The AI Model Hub also leverages Hypha's **serverless application framework**, allowing developers to build lightweight Python or JavaScript applications that provide graphical interfaces or automated workflows for hosted models. These applications can be accessed directly from the hub, lowering the technical barrier for end users.

By combining model storage, scalable execution, and user-friendly application interfaces, the AI Model Hub delivers a modular and interoperable foundation for AI-powered research in the DEP.

### 3.3.1. Component Status Overview

The core components of the AI Model Hub - the **Artifact Manager**, **BioEngine**, and the **serverless application framework** - are already in an advanced stage of development, with stable open-source implementations available through the Hypha and BioEngine projects. These components have been deployed and tested in production-like environments, primarily in the context of the BioImage Model Zoo and related research infrastructures, demonstrating their capability to support real-world AI model hosting and execution.

- The **Artifact Manager** is a mature Hypha service for storing and indexing models as versioned artifacts in S3-compatible object storage. It includes a relational metadata database for structured search and retrieval, as well as access control features to manage public and private content. This component is already in use for managing large model collections and has proven stable for both small and large-scale deployments.
- **BioEngine**, the Ray-based execution backend, is also in an advanced state, supporting containerised model execution with GPU acceleration and scalable task scheduling. It provides a standardised API for invoking models and has been integrated with multiple client

<sup>18</sup> <https://bioimage.io>

<sup>19</sup> <https://docs.amun.ai/>

<sup>20</sup> <https://docs.amun.ai/#/artifact-manager>

<sup>21</sup> <https://doi.org/10.5281/zenodo.14169671>

interfaces, including Jupyter notebooks and web applications. While it is already in use for bioimage analysis pipelines, targeted adaptations will be needed to support a broader set of AI model types and to align with DEP's infrastructure and orchestration requirements.

- The **serverless application framework** within Hypha enables the creation of lightweight, interactive applications for accessing and visualising model results. It supports both Python and JavaScript runtimes, making it straightforward to deploy custom user-facing tools that connect directly to hosted models. While the framework is fully functional, further work will be needed to adapt these applications for seamless integration within the DEP's user interface and to support the specific workflows of WP3 use cases.

Overall, the AI Model Hub components are technically mature and production-ready in their current domains. However, their deployment in the DEP will require integration work, interface harmonisation, and targeted extensions to meet the needs of WP3 applications and ensure smooth operation in the DEP environment.

User Documentation: <https://docs.amun.ai/#/getting-started>

Source code: <https://github.com/amun-ai/hypha>

Licence: MIT

### 3.3.2. Identified Gaps and Developments planned in RI-SCALE

The **AI Model Hub** (Artifact Manager + BioEngine + Hypha serverless apps) is technically mature for bioimage models from AI4Life, but several extensions are required for DEP-wide adoption across domains and infrastructures:

- **Scope and metadata generalisation:** Extend the current BioImage-centric schema into a domain-agnostic model card covering tasks, modalities, licenses, dataset references, runtime requirements, and provenance fields. Enhance the SQL-based index with richer search (facets, tags, aliases) so models are reusable and discoverable across multiple scientific domains.
- **Execution portability and heterogeneous compute integration:** Maintain BioEngine as the primary hosted inference backend while adding an interLink pathway for offloading workloads to HPC and cloud resources. This requires runtime compliance for different environments (e.g., Apptainer on HPC), adapters to interLink's plugin system, and mechanisms to propagate job status and results back to the Hub.
- **Workflow orchestration and provenance tracking:** Integrate tightly with itwinai so pipelines can pull Hub models for training or inference and push back packaged checkpoints as new versions. Standardise manifest hand-off to ensure reproducible runs, and feed execution metadata into yProv4ML/MLflow for lineage tracking, metrics collection, and basic benchmarking linked to datasets and outputs.

- **Access control, policy compliance, and data management alignment:** Connect Hub operations with WP4's policy-based authorisation to enforce project/group-level permissions, logging, and accounting. Support integration with WP2's data services so model runs can directly reference DEP dataset IDs, co-locate execution near data, or stream large datasets through tiling and chunked I/O.
- **User-facing interaction through serverless apps and LLM agents:** Harden the Hypha serverless app framework for safe, multi-tenant use, enabling lightweight Python/JS UIs (e.g., forms, visualisers, notebooks) attached to Hub models. Surface these apps directly in DEP interfaces and expose LLM-friendly OpenAPI/JSON-Schema endpoints so agents like the BioImage.IO Chatbot can search, configure, and invoke models in use cases such as SUC 8.
- **Operational robustness and scalability:** Improve caching, de-duplication of repeated runs, logging/observability, rate-limiting, and retry mechanisms. Support external weight fetching from sources like Hugging Face using the requester's credentials, and ensure large-artefact handling is efficient and resilient.

### 3.3.3. Planned Integration with other Components in WP3

The main planned integrations of the AI Model Hub with other technical solutions in WP3 are:

- **itwinai:** Potential interoperability between itwinai and the AI Model Hub could allow orchestration workflows defined in itwinai to invoke models hosted in the Hub through the BioEngine inference backend. This would enable more complex workflows, such as combining simulation components with AI-based analysis. Model execution endpoints may be referenced in itwinai configurations, allowing seamless selection of computational infrastructure for running inference tasks.
- **interLink:** The AI Model Hub may connect with interLink's resource brokering and workload distribution mechanisms to make better use of heterogeneous computing resources across the DEP. This would allow BioEngine-managed inference jobs to be scheduled on suitable HPC or cloud resources, improving scalability and efficiency without requiring manual resource selection by end users.
- **yProv4ML:** Model execution metadata from the AI Model Hub could be aligned with provenance and traceability formats supported by yProv4ML. This would provide consistent recording of model versions, input data references, and execution parameters, aiding reproducibility across workflows that span multiple DEP components.
- **Data Access and Delivery Services:** Where beneficial, AI Model Hub workflows could be adapted to operate directly on datasets accessible through DEP's data access layers, including support for streaming large image or numerical datasets. This would minimise data movement and enable more efficient execution of data-intensive models.

## 3.4. BioImage.IO Chatbot

The BioImage.IO Chatbot<sup>22</sup> is a conversational assistant designed to help researchers navigate the increasingly complex ecosystem of bioimaging tools, data resources, and analysis workflows. Originally developed for the BioImage Model Zoo<sup>23</sup>, it was built on GPT-4 and enhanced with retrieval-augmented generation (RAG) and tool execution capabilities. The chatbot connects users to a curated knowledge base drawn from community documentation, including resources like ImageJ, deepImageJ, and the BioImage Model Zoo—as well as external databases such as bio.tools and the Human Protein Atlas. It goes beyond static question-answering by dynamically generating Python code, running AI models via BioEngine, and even inspecting image data using vision capabilities. Within the context of RI-SCALE, the chatbot acts as a user-friendly access point to large-scale bioimaging resources and services, lowering technical barriers and making AI-driven analysis more approachable. Through its extension system, it can be embedded into different platforms or customized for specific use cases, offering a flexible interface that supports both discovery and hands-on data analysis. As part of the RI-SCALE effort to make data more accessible and actionable, the chatbot demonstrates how LLM-based systems can support researchers in working more effectively with scientific data.

### 3.4.1. Component Status Overview

The BioImage.IO Chatbot is in an advanced but evolving stage of development, having originated within the BioImage Model Zoo ecosystem and subsequently expanded to support more interactive and multimodal capabilities. It operates as a web-based conversational interface built on large language models (currently GPT-4), enhanced with retrieval-augmented generation (RAG) pipelines and tool execution modules. The chatbot is designed to operate with multiple assistant “personalities” (e.g., *Melman* for model discovery, *Bridget* for image analysis), each specialised for different aspects of bioimage research.

A public, production-grade instance is available at <https://bioimage.io/chat>, allowing users to explore bioimaging resources through natural language interaction. The backend integrates with curated domain knowledge bases - including documentation for ImageJ, deepImageJ, and BioImage Model Zoo entries - as well as external APIs such as **bio.tools** and the **Human Protein Atlas**. In addition to pure information retrieval, the chatbot can dynamically generate Python code, execute AI models via BioEngine, and perform basic image inspection through integrated vision model capabilities.

The system is modular and supports embedding into third-party portals, making it adaptable for different research infrastructures or institutional deployments. Extensions can be added to connect the chatbot with additional datasets, computational backends, or workflow engines. These features are enabled through a tool-calling architecture, which allows the chatbot to trigger remote services (e.g., running a segmentation model) and handle asynchronous responses.

<sup>22</sup> <https://github.com/bioimage-io/bioimageio-chatbot/>

<sup>23</sup> <https://bioimage.io>

Demo link: <https://bioimage.io/chat>

Source code: <https://github.com/bioimage-io/bioimageio-chatbot>

Licence: MIT

### 3.4.2. Identified Gaps and Developments planned in RI-SCALE

The **BioImage.IO Chatbot**[R9] is mature for bioimaging-oriented Q&A and guided discovery, but several extensions are needed for broader DEP adoption and interactive analysis:

- **Scope and interface generalisation:** Broaden beyond bioimage-specific intents to a domain-agnostic skill set where feasible (tasks, data modalities, model families). Provide clearer, machine-readable tool descriptions (OpenAPI/JSON-Schema) and stable function signatures so external services can be invoked reliably from chats.
- **Execution handoff and long-running jobs:** Add first-class support for asynchronous, resumable jobs (job IDs, status polling, partial results) so the chatbot can initiate analysis that runs on backend services and return when ready. Standardise request/response envelopes for tasks that route to the **AI Model Hub** (BioEngine) or other DEP services.
- **Data access alignment with DEP:** Enable selection and referencing of DEP datasets (IDs, access policies), lightweight preview/tiling for large images, and safe upload where permitted. Minimise data movement by preferring in-place processing and streaming pathways exposed by DEP data services.
- **Provenance, tracking, and reproducibility:** Emit structured run metadata (model/version, parameters, dataset references) for actions triggered by the chatbot, compatible with **yProv4ML/MLflow**, so results are traceable and comparable across sessions and users.
- **Safety, policy, and multi-tenancy:** Integrate with DEP AAI/authorisation to respect project- and role-based access when discovering models or launching runs. Add guardrails (rate limits, quota checks, input validation) for multi-user environments.
- **LLM/agent robustness and multimodality:** Improve tool-use reliability (fallbacks, retries, timeouts) and strengthen image-aware prompting for vision-in-the-loop tasks (e.g., ROI guidance, quick quality checks). Where helpful, expose small, scoped serverless UIs (Hypha apps) that the chatbot can open for parameter entry or result visualisation.
- **Operational observability:** Add logging and telemetry for tool calls (success/failure, latency), lightweight analytics for intent coverage, and configurable feature flags to enable incremental rollout within DEP sites.

These developments aim to keep the chatbot's role focused - conversational discovery and orchestration - while enabling dependable handoff to DEP services (AI Model Hub/BioEngine, data access, provenance) in a modular, standards-friendly way.

### 3.4.3. Planned Integration with other Components in WP3

As part of **SUC 8** and WP3, the BioImage.IO Chatbot will serve as a conversational access point to DEP's AI model hub and execution services. The main planned integrations are:

- **AI Model Hub / BioEngine:** The chatbot will connect directly to the AI Model Hub - backed by the Hypha Artifact Manager for model hosting and indexing - and to BioEngine for scalable inference on Ray clusters. This will enable users to search for models, review metadata, and trigger inference tasks (e.g., segmentation, classification) on selected datasets using natural language prompts. Support will include both interactive execution and submission of long-running jobs.
- **itwinai:** Potential integration with itwinai could allow chatbot-initiated model execution requests to be incorporated into digital twin workflows. This would enable models triggered from conversational queries to be part of broader simulation or predictive scenarios defined in itwinai.
- **yProv4ML:** Workflows launched from the chatbot may be connected to yProv4ML to capture provenance data, including model identifiers, execution parameters, and dataset references. This will enhance traceability and reproducibility for AI-powered analyses initiated via the chatbot.
- **interLink:** For cases where model execution needs to leverage heterogeneous computing resources, the chatbot could route jobs via InterLink, enabling execution on HPC, cloud, or other infrastructure while maintaining a unified user experience.

These integrations will be designed to remain modular and standards-based, ensuring the chatbot can act as a flexible orchestration layer for SUC 8 and potentially other WP3 scenarios.

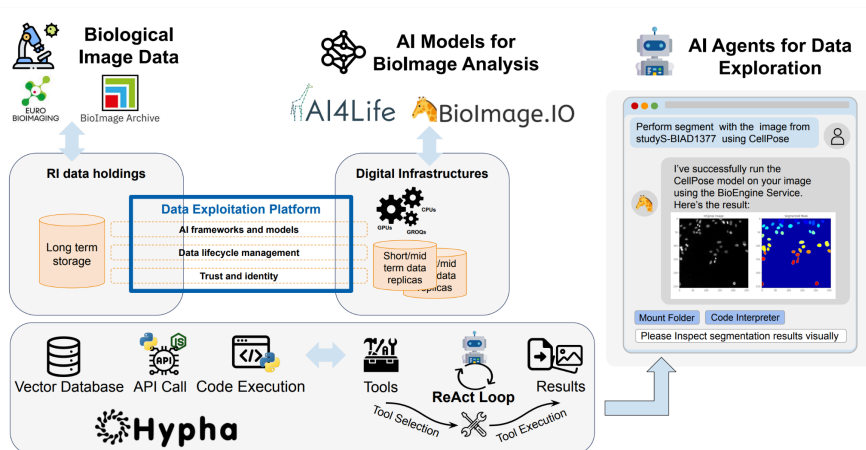


Figure 3: Integration of AI Agents for the DEP in SUC 8



## 3.5. yProv4ML

yProv4ML is a Python library designed with the objective of efficiently handling large-scale ML experiments. Its modular architecture and data model ensure that provenance information can be managed effectively, regardless of the complexity or size of the ML project. Since the library saves provenance data in the JSON format, the interoperability between different ML tools and platforms is enhanced, as users are allowed to choose their preferred data analysis method. This facilitates the sharing and comparison of provenance information across diverse research groups and projects, which in turn fosters collaboration and advances the state of the art in ML research. Provenance documents generated by yProv4ML are stored within the yProv store service, another component of the yProv ecosystem [R2], that will support provenance management within the RI-SCALE project.

### 3.5.1. Component Status Overview

yProv4ML has been developed in accordance with the specifications set forth by MLFlow, which allows for seamless integration of provenance tracking into existing workflows. yProv4ML offers a set of function calls that enable users to track metrics and parameters that are useful for the subsequent analysis of the training process and embed them within interoperable provenance documents.

This library separates the collected information into three modalities: artefacts, parameters and metrics. The first category identifies any file or output used in subsequent phases of the workflow; parameters represent one-time logged values utilized during the training phase; metrics relate to information that is updated during the training process (i.e., energy efficiency, power consumption, and GPU usage).

The main modules of yProv4ML are:

- **Main module:** which includes all of the functionalities of the library and allows for context declaration and shutdown.
- **Energy module:** it contains all utility functions to save energy-related metrics.
- **System module:** contains directives to save information related to the system.
- **Time module:** it contains helper functions to manipulate and save information.

The user and technical documentation are available at: <https://github.com/HPCI-Lab/yProvML> and <https://hpci-lab.github.io/yProv4ML.github.io/index.html>.

The source code is available at <https://github.com/HPCI-Lab/yProvML>.

The license is GPLv3.

### 3.5.2. Identified Gaps and Developments planned in RI-SCALE

According to the use case requirements listed in D5.1 [R1], here is a set of gaps and developments planned during the RI-SCALE project.

The gaps identified in the yProv4ML component relate to the following areas:

- **Scaling Metrics:** Depending on the number of epochs and steps, the set of metrics in training processes can be challenging to manage. Developments in this area will focus on (i) decoupling the provenance metrics from the process description and (ii) identifying more efficient data formats to store metrics as well as (iii) embedding compression techniques.
- **Experiment-level provenance view:** ML experiments can be quite articulated and include multiple runs. A cohesive view of all of them is missing from a provenance perspective, which, of course, could provide a more consistent view of the entire ML experiment. Developments in this area relate to fully implementing the yProv4ML data model, which already defines the *experiment* concept and its integration with RO-Crate<sup>24</sup>.
- **GUI:** Navigating, analyzing and exploring yProv4ML provenance documents can be beneficial for end users as it allows drilling down into the ML process, understanding metrics in depth, comparing different runs, etc. Developments in this area relate to a graphical interface that can fill this gap.
- **Ecosystem integration and interoperability:** yProv4ML will be further integrated with tools and standards in the area (e.g., MLflow, yProv store service) to provide users with better support in the development of AI models.
- **Metrics:** Improving metrics management across the ML training process will be addressed throughout the project lifetime.

### 3.5.3. Planned Integration with other Components in WP3

Planned integration with other WP3 components includes:

- **itwinai:** This will involve further extending and strengthening the integration with itwinai (early-stage development done in interTwin). This has been realized through the logger mechanism in itwinai, which makes the integration of yProv4ML transparent as one of the “logger” libraries available to the end users. The activity will extend to the new yProv4ML developments that will be addressed during the project.
- **AI Model Hub:** Integration with the model hub framework will be performed, which involves at least MLflow and the yProv store components. This activity will connect yProv4ML documents with PIDs and artifacts (i.e., AI models) that will be stored on the model hub. It will also provide a better link to MLflow through the implementation of the *experiment* concept foreseen in yProv4ML.

<sup>24</sup> RO-Crate <https://www.researchobject.org/ro-crate/>

## 4. AI Applications in WP3

This section summarizes the AI applications that drive the technical developments in WP3. These consist of the environmental and health/life science applications. Each is presented with a summary of the involved SUC and the associated compute and data challenges.

### 4.1. Environmental Applications

AI is playing a pivotal role in environmental science, enabling the extraction of insights from increasingly complex and voluminous datasets. However, unlocking the full potential of AI in this domain requires not only access to high-quality data but also compute resources and dedicated models tailored to the environmental monitoring and analysis challenges.

In the scope of RI-SCALE, domain-specific AI models will be integrated into the DEP framework and trained on big data to assist RI operators and users, thus enhancing their ability to monitor, understand, and manage environmental systems in a more informed, timely and effective way. The environmental applications include four SUCs targeting two thematic RIs from Environmental sciences: European Network for Earth System Modelling (ENES) and European Incoherent Scatter Scientific Association (EISCAT). These use cases include:

- High-resolution downscaling of climate scenarios and risk trend analysis in agriculture;
- Smart detection of anomalies in climate data usage;
- Intelligent Scheduling of Radar Observations and Experiments;
- Space Debris and Anomaly Detection.

Further details are provided in the following subsections, which provide an overview of each AI application along with the related compute and data challenges aimed to be tackled.

#### 4.1.1. SUC 1: High-resolution Downscaling of Climate Scenarios and Risk Trend Analysis in Agriculture

##### Use Case Overview:

The agricultural and insurance sectors critically depend on high-resolution climate data (spatial resolution of a few hundred meters) for accurate risk assessment, operational planning, and long-term strategy development. However, currently available climate projections, such as those from CMIP6<sup>25</sup> (Coupled Model Intercomparison Project Phase 6) and EURO-CORDEX, are provided at much coarser resolutions (10–100 km), which are insufficient for regional or local-scale decision-making.

<sup>25</sup> <https://pcmdi.llnl.gov/CMIP6/>

**WP3 Developments / AI Approach:**

In the RI-SCALE project, the aim is to develop a novel infrastructure for downscaling the coarse-resolution climate datasets into high-resolution maps suitable for sectoral applications. This infrastructure will leverage AI and statistical methods, with a particular focus on Convolutional Neural Networks (CNNs). State-of-the-art CNN-based models, based on prior work (such as [R7] and [R8]), will be evaluated and fine-tuned within the project. These models will address two different approaches commonly used in statistical downscaling of climate scenarios: "perfect prognosis" and "super-resolution".

The technical implementation will exploit various regional reanalysis datasets (e.g., CERRA, VHR\_REA\_IT), depending on the geographical region, as predictands, and climate projection datasets as predictors. In addition, Digital Elevation Models (DEMs) and terrain features will be incorporated to improve the accuracy of the high-resolution outputs.

The infrastructure will support large-scale data access and processing from ESGF and Copernicus repositories, including CORDEX and CMIP6 simulations, and will enable the training of AI models required for downscaling. The development pipeline will include distributed training, hyperparameter optimization, and inference, carried out on GPU-based systems, as well as model benchmarking against classical statistical downscaling methods (e.g., quantile mapping) to evaluate performance across multiple metrics (accuracy, computational cost, scalability). Model development and testing will initially focus on two or three pilot sub-domains to contain computational costs.

A core objective will be to validate and exploit the high-resolution outputs in real-world settings, particularly within risk trend analysis in agriculture, to demonstrate the added value of high-resolution data in these sectors.

**Compute and Data Challenges:**

The development of the downscaling models presents significant data and computational challenges. Climate projections and reanalysis datasets from ESGF and Copernicus sources can exceed 100 TB, depending on the number of models processed and on the domain and variables used. This requires extensive pre-processing, storage management, and data reduction strategies. Model training will be carried out in Python/Conda environments using frameworks such as PyTorch, TensorFlow, or Keras, with distributed training on GPU clusters and hyperparameter optimization. Based on prior studies, training similar CNN-based architectures may require hundreds to thousands of GPU hours, depending on model complexity, input domain size, and resolution targets. Different architectures (such as DeepSD) will be tested, fine-tuned, and benchmarked against classical downscaling techniques (e.g., quantile mapping) in terms of accuracy, training time, and scalability. Containerized workflows may be adopted during the development.

## 4.1.2. SUC 2: Smart Detection of Anomalies in Climate Data Usage

### Use Case Overview:

In the Climate Science domain, community efforts like CMIP<sup>26</sup> represent very relevant large-scale global experiments initiatives, which have led to the development of the Earth System Grid Federation (ESGF) [R3], one of the largest collaborative data efforts in Earth system science. ESGF consists of a federation of autonomous data nodes, distributed across several countries and united by common standards, protocols and interfaces. Data, including simulations, observations and reanalysis, is hosted at multiple sites worldwide and served through local data and metadata services. Therefore, monitoring this large distributed infrastructure has become a very challenging topic over the years.

### WP3 Developments / AI Approach:

ML techniques can help enhance the operational reliability of the ESGF infrastructure, providing valuable information about data access patterns, transfer activities, and user interactions across the distributed network of ESGF nodes. By applying anomaly detection algorithms, it would potentially be possible to automatically identify irregular behaviours, such as sudden drops in data access, unexpected traffic spikes, or incomplete data transfers, that may signal underlying technical issues or failures in the data delivery process.

Moreover, integrating ML into the ESGF data usage monitoring system could contribute to uncovering trends in data usage and detecting changes in download patterns, which is particularly relevant for the climate science community. Understanding how CMIP data is accessed and utilized through the ESGF infrastructure can help optimize resource allocation, improve data dissemination strategies, and support long-term planning for future model intercomparison projects.

### Compute and Data Challenges:

The ESGF Data Statistics service [R4] is a core component of the ENES RI that takes care of collecting, analyzing, and reporting a comprehensive set of data usage metrics and data archive information across the ESGF infrastructure. More specifically, the service continuously gathers fine-grained information on data access, transfers, and usage across the globally distributed ESGF infrastructure. The considerable amount of data usage information retrieved from each ESGF data node (~1TB of historical data from January 2018 to February 2025) requires efficient storage solutions capable of handling heterogeneous, high-volume log data. To this purpose, a series of data warehouse systems have been designed to collect and archive usage logs, allowing for scalable ingestion, fast querying and seamless integration with ML workflows aimed at anomaly detection, trend analysis and usage pattern forecasting.

---

<sup>26</sup> <https://wcrp-cmip.org/>

Moreover, preprocessing and harmonizing such logs, as well as training the corresponding ML models, could demand significant computational resources needing access to both GPUs and CPUs. This is due not only to the volume of the collected data, but also to the need for efficient filtering and aggregation, temporal alignment, hyperparameter tuning, iterative optimization, and performance model validation. All of this proves to be important for ensuring timely detection of anomalies or changes in data usage, and for enabling the deployment of a robust, responsive monitoring system supporting the research infrastructure operations.

In addition, transparency, reproducibility, and traceability represent essential aspects for ensuring that model results can be understood, verified, and trusted. In this regard, it would be worth tracking provenance (including artefacts, parameters and performance metrics) throughout the entire ML pipeline in order to enable future exploration, thus allowing infrastructure managers and scientists to understand how models were trained, which data were used, and under which conditions.

### 4.1.3. SUC 3: Intelligent Scheduling of Radar Observations

#### Use Case Overview:

EISCAT AB operates high-power ionospheric research radars (incoherent scatter radars, ISR) in Northern Fenno-Scandinavia and on Svalbard, which provide detailed information of the atmosphere and ionosphere from 70 km altitude upwards, even as far away as the Moon. Currently, the new tri-static, phased-array EISCAT\_3D radar is being deployed. Other than the legacy radars, EISCAT\_3D will be fully remotely controlled.

All radars will operate on request by the EISCAT users. In practice, a researcher requests radar time and specifies what kind of radar operations they want and what kind of space weather (environmental) conditions are required. For EISCAT\_3D, eventually EISCAT will decide when to run the requested observations, and - in case of competing requests - which request to prioritise.

The specification of radar operations, resulting in a “radar experiment”, consists of information such as beam pointing direction (azimuth, elevation), range extent, range resolution, timing, and, for multi-beam experiments, the schedule and order of beams to cycle through. Furthermore, for tri-static experiments involving the two remote receiver sites, the user also specifies the altitude resolution of the common volumes (beam overlaps) for which to compute wind velocities as 3D vectors.

The environmental conditions for a radar experiment include space weather parameters such as solar wind density and velocity, direction of the interplanetary magnetic field, solar activity, geomagnetic activity, as well as terrestrial weather parameters, mostly whether or not it’s cloudy or clear, and the elevation of the Sun as well as elevation and phase of the Moon, which are important to define light, twilight, and dark conditions.

**WP3 Developments / AI Approach:**

The purpose of SUC 3 is to define an ML process, which analyses the current and near-future environmental situation as well as the user-specified observational parameters and makes a suggestion for operations, i.e. which experiment to execute for the best possible outcome. The process can also reject all experiments when none are expected to yield useful data.

**Compute and Data Challenges:**

The core data challenges for SUC 3 include the correct definition of the requirements and access to the relevant resources. This is preceded by the selection of the parameters, which will be available to the radar users to specify their experiment.

These parameters largely belong to two groups: (1) required environmental conditions and (2) required radar setup. These groups can each be split into two sub-categories: environmental conditions may refer to (1.1) space weather or (1.2) terrestrial weather, and the radar setup typically consists of (2.1) beam geometries, i.e. number of beams and pointing directions, and (2.2) radar signal coding, which defines temporal and spatial resolution and extent of the experiment. While the choices for (2) are well defined, the choices for (1) need to be specified, and user feedback should be obtained.

With the choices defined, data products need to be identified to match the requested choices. Note that most of these data products come from external providers, so access policies and technical interfaces need to be clarified accordingly.

These two challenges are mutually dependent: radar users cannot be offered to make a choice of environmental conditions without corresponding data products to check the current state, while the choices requested by users will, in turn, determine which data products are required to make decisions.

The main task of SUC 3 is for an AI process to evaluate whether a specific radar experiment should be conducted on a given day. However, the next challenge is consequently to decide if the result of the experiment would be better on the following day, i.e. allow for limited forecasting of environmental parameters.

The computational challenges for SUC3 arise primarily from the need to evaluate complex, time-dependent data in a dynamic operational setting. The system must ingest and harmonise a wide range of environmental data streams (both from terrestrial and space weather sources), often in real time and with possible varying data formats and latencies. To assess whether an experiment should be run now or postponed, the AI process must include short-term forecasting capabilities and reason over predicted conditions, which adds computational load and uncertainty handling. The decision logic must support prioritisation among competing requests, balancing scientific value against timing and resource availability, all while remaining responsive to new inputs. Additionally, the system

must learn from outcomes over time, adapting its recommendations based on past experiment success. This requires infrastructure for continuous learning and the integration of feedback.

#### 4.1.4. SUC 4: Space Debris and Anomaly Detection

##### Use Case Overview:

EISCAT AB has operated incoherent scatter radars in Northern Fenno-Scandinavia since 1981, and thereby accumulated a vast archive of near-Earth space observations. These data have been actively studied, and EISCAT can look back to nearly 2500 papers published in international scientific journals. However, sometimes new phenomena are detected, which earlier were disregarded or misinterpreted. Furthermore, statistical studies of the occurrence of particular phenomena are, in practice, a lot of work of manually browsing quick-look plots.

##### WP3 Developments / AI Approach:

With SUC 4, AI methods will be investigated to find specific events as well as anomalies, i.e. rare events or disturbances, in this dataset. This requires classification of known events and then searching for non-classified structures in the data, which can include, but is not limited to, anthropogenic space objects as well as meteoroids traversing the radar beams.

The aim is to uncover phenomena in the data which have been missed in previous analyses or have been possibly wrongly categorised. Furthermore, it is intended to extract statistical information from the observations. Such a tool can be trained on the existing data, and using it to study the archive will already facilitate new science. However, the method will then be applied to future radar data to automatically flag and identify the phenomena and events in real time as they happen.

With the advent of “New Space” (i.e. miniaturisation leading to ever smaller satellites, as well as rapidly increasing numbers of satellites), overcrowding of orbits increases the risk of collisions, thereby creating more space debris, which eventually can lead to cascading, i.e. causing a “chain reaction” (Kessler Syndrome). The existing EISCAT data will be used to identify events related to the presence of space objects, establish occurrence statistics and find connections between these events, and then identify anthropogenic objects that can threaten sustainable space use, while producing sets of tracks (range, range rate, and radar cross-section measurements) to be used for orbital determination.

##### Compute and Data Challenges:

SUC 4 is an exercise in data analysis. EISCAT has accumulated an archive of radar data reaching back to 1981. The archive is fully digital, but the interface for the use of the data for AI processes must be specified and implemented. The data formats have to be implemented as well, as these might have changed over the years. While radar experiments have evolved and changed over time, it is intended to use exclusively standard incoherent scatter data (high-level data) for this use case. Among them, the most important are the vertical profiles of electron density, as well as electron and ion temperature. These vary over time only in altitude extent, as well as temporal and spatial resolution.



One particular challenge related to radar experiment definition is that every measured profile is associated with the radar beam direction (azimuth, elevation), which can change in so-called scanning experiments. Any analysis process needs to be able to recognise this and treat scanning experiments differently from fixed-beam experiments.

Finally, the developed process has to be applied to future data to detect phenomena as they occur. However, the format of the future data is yet unspecified. Therefore, the process should have a data interface into which it is easy to feed data. This would also allow applying the process to other, unrelated incoherent scatter radars from our collaborators.

As mentioned, the AI process must handle decades of radar data collected under varying experimental setups and evolving data formats. Training AI models on such a large and varied dataset is computationally demanding, both in terms of storage and processing power, especially when accounting for changing data formats, resolutions, and experimental configurations over time. Furthermore, the system must be designed for efficient inference to allow real-time application on future radar streams, even though those data formats are not yet defined. This calls for a flexible and modular architecture that can generalise across both historical and new data.

## 4.2. Health and Life Science Applications

DEPs are expected to have a wide impact on biological and medical imaging. The scientific image datasets stored at repositories such as the BBMRI-ERIC or the BioImage Archive (BIA) are large and complex, and, in the case of medical imaging, they can contain sensitive patient information that cannot be shared openly. AI methods have the potential to help analyse, categorise, and understand these complex collections of images, and ultimately provide insights, both in fundamental scientific research and for health benefits through clinical application. The DEPs present the perfect opportunity to do this at scale. In Task 3.4, generative, foundational, and multimodal AI methods will be developed and trained to address challenges related to health and life science applications.

### 4.2.1. SUC 5: Colorectal Cancer Prediction with explainable AI

#### Use Case Overview:

Since lymph nodes are the first anatomical checkpoint in metastatic spread, their microarchitecture and immune-cell composition may contain prognostic signals that routine histopathology overlooks. In SUC 5, algorithms will be developed for patient-survival prediction from lymph-node whole-slide images (WSIs) of colorectal cancer (CRC).

#### WP3 Developments / AI Approach:

Deep neural network models will be trained that regress an individual's survival directly from WSIs. By analysing the resulting attention maps and feature-attribution scores, specific microscopic structures will be identified, whose presence or absence systematically correlates with longer or

shorter survival times. Such image-derived biomarkers, once validated, could refine adjuvant therapy decisions and advance the understanding of CRC progression at the biological level.

Development will proceed in close collaboration with clinical experts from the Medical University Graz (MUG) and the Masaryk Memorial Cancer Institute (MMCI). Their domain knowledge is essential for several stages of the pipeline: curating high-quality lymph-node annotations, reviewing attention maps to determine whether highlighted regions correspond to plausible histological features, guiding the selection of clinically meaningful evaluation metrics, and assessing the clinical actionability of the resulting risk scores. Interim model outputs will be discussed with experts, and final models will be tested against established benchmark datasets.

#### **Compute and Data Challenges:**

For the implementation of this use case, approximately 90,000 lymph-node WSIs extracted from the BBMRI-ERIC CRC cohort [R5] will be utilized. To enable a multi-centre assessment of generalisability, an additional collection of lymph-node slides will be obtained from the MMCI and ingested through the same DEP workflow.

WSIs scanned at large magnifications typically reach resolutions around  $100,000 \times 100,000$  pixels, with individual files ranging from 2 GB to 5 GB; the BBMRI set alone therefore occupies hundreds of terabytes. Handling image data of this scale requires substantial storage capacity, high-throughput pipelines, and extensive multi-GPU compute resources. These requirements are met through the DEP's data-orchestration layer (Rucio + FTS) for secure transfer, the data-preparation service for format harmonisation and optional JPEG 2000 compression, and the itwinai framework for distributed training across clustered GPUs, with checkpoints and metadata captured in the DEP model hub. Additionally, an instance of the web-based WSI viewer XoPat will run within the DEP, allowing clinicians and data scientists to inspect raw slides, attention maps, and synthetic outputs directly in the browser, thereby eliminating the need to transfer terabyte-scale datasets to local machines.

### **4.2.2. SUC 6: Synthetic Data for Computational Pathology**

#### **Use Case Overview:**

In SUC 6, diffusion-based generative modelling will be investigated as a means to mitigate the data-access constraints imposed by patient-privacy laws and the corresponding institutional policies governing high-resolution WSIs. Since histological slides encode uniquely identifiable cellular patterns, these regulations mandate that pathology images held by RI repositories remain within secure computing environments, limiting their availability for external method development and validation.

#### **WP3 Developments / AI Approach:**

Training state-of-the-art diffusion models on these protected data will allow the creation of *synthetic* WSIs whose image statistics faithfully mirror cellular architecture, staining variability, and

artefacts while containing no patient-specific information. Such privacy-preserving surrogates can be shared without legal impediment, thereby expanding access to realistic training materials and benchmark sets. Moreover, a systematic analysis of the generator’s latent space and attention mechanisms provides a novel, data-driven avenue for identifying the histomorphological motifs most characteristic of colorectal cancer tissue, offering complementary insights into tumour organisation and variability.

The use case has two primary objectives. The first is the generation of entire WSIs at full scanning resolution, employing cascaded or patch-assembly diffusion schemes to preserve both the global slide layout and fine cellular detail. The second objective is conditional synthesis, in which the diffusion process is steered, for example, by spatial semantic maps that encode tissue classes or regions of interest, enabling the creation of slides with user-specified structural composition. Pursuing these directions will yield privacy-preserving data that are simultaneously anatomically realistic and experimentally controllable, broadening the scope for downstream method development and validation.

#### **Compute and Data Challenges:**

For the implementation of SUC 6, about 200 TB of colorectal cancer WSIs obtained from the BBMRI-ERIC repositories are planned for use. A major challenge is posed by the fact that state-of-the-art image generation models are typically designed for images ranging only up to a few megapixels. In contrast, each pathology slide reaches the gigapixel range ( $\approx$  approximately  $100,000 \times 100,000$  pixels, 2–5 GB per file). Adapting these models, therefore, poses both algorithmic and computational challenges: the global context must be preserved across full WSIs, and latent spaces must accommodate orders of magnitude more information. Therefore, the development of the generative model is expected to require about 75,000 GPU hours on 64–128 GPUs.

These demands are addressed through the same DEP components used in SUC 5. Slides are staged via the data-orchestration layer into DEP storage, harmonised by the data-preparation service (with optional JPEG 2000 compression), and then streamed into a distributed diffusion workflow executed by the itwinai framework. Checkpoints and hyperparameters are versioned in the DEP model hub, while access remains restricted by the platform’s policy-based AAI layer. An XoPat viewer instance hosted on the DEP will facilitate structured user studies with pathologists, allowing them to inspect real and synthetic slides side by side, assess visual fidelity, and provide feedback without transferring gigapixel data outside the controlled environment.

### **4.2.3. SUC 7: Foundational Models for Heterogeneous Biological Image Data**

#### **Use Case Overview:**

The BIA is EMBL-EBI’s data resource for biological images, and it hosts over 800TB of image data. However, biological images are very diverse; they can be multidimensional, produced by different imaging tools, come in different formats, and are typically acquired to address questions in varied

scientific disciplines. As a result, users of the BIA may find it difficult to make sense of the plethora of data available in the repository. A promising path forward is to create foundation models that understand and represent heterogeneous image data, providing BIA users with better data categorisation and search.

#### **WP3 Developments / AI Approach:**

To address the above challenges, a foundational model that will enable better discoverability, organisation and reuse of imaging data will be developed. The DEPs will be exploited to train the model on a curated subset of data from the BIA, taking data from multiple different database entries. In parallel, existing natural image segmentation models will be fine-tuned on heterogeneous datasets, and the models will be benchmarked using task-specific evaluation datasets. The model outputs and resulting metadata will then be transferred back to the BIA, where the model embeddings will be used to support downstream use cases, such as categorisation and similarity search, and derived measurements such as cell dynamics and morphology.

#### **Compute and Data Challenges:**

To support the development and deployment of a large-scale biological imaging foundation model, the proposed project requires access to computational resources, including 10,000 GPU hours on 64 A100 GPUs or equivalent infrastructure. The work will involve training and fine-tuning models on a dataset of approximately 100TB, leveraging both domain-specific and natural image segmentation models such as SAM [R6] (~1.3GB). Given the size of the datasets, distributed training and inference using Ray may be needed in this workflow. All environments will be containerized using Apptainer or similar to ensure reproducibility and compatibility with HPC systems.

### **4.2.4. SUC 8: Generative AI-Powered Assistant for Data Discovery and Analysis**

#### **Use Case Overview:**

The complexity associated with biological images is exacerbated by the variety of available tools for image analysis. BIA users sometimes struggle to find the most appropriate tool for a specific task or image type. Generative chat engines could revolutionise the way BIA users search through and interpret vast datasets.

#### **WP3 Developments / AI Approach:**

For this use case, a generative AI-powered assistant which is embedded directly within the BIA interface will be developed. This assistant will integrate advanced AI models for both efficient data retrieval and complex image analysis - such as segmentation, classification, and cell morphology assessments - while offering an intuitive, natural-language-based interface. Users will be able to locate relevant datasets, ask analytical questions, and execute detailed workflows (e.g. segmentation or quantification) without leaving the platform. By dynamically generating Python code, executing AI models, and returning results in interpretable formats, the assistant lowers technical barriers for non-expert users and enables more inclusive access to BIA's growing data landscape.

The assistant builds on the foundation established in SUC 7, adding a generative layer for dataset interaction and analysis. It acts as an accessibility layer to the BIA, supporting semantic search, exploratory queries, and on-demand execution of models such as Segment Anything, Cellpose, and the BioImage.IO Chatbot, see Section 3.4. The assistant will also provide guidance for selecting tools appropriate to specific image types and tasks, helping to reduce the manual support burden for RI operators. Developed jointly by Euro-BioImaging and KTH, and integrated tightly with the DEP infrastructure, this system aims to democratise bioimage analysis while strengthening the role of Euro-BioImaging as a key access point for data and services in the European research landscape.

**Compute and Data Challenges:**

A flexible, GPU-accelerated infrastructure is required, which is ideally built on k8s rather than traditional HPC setups using job schedulers like Slurm. Approximately 12,000 GPU hours across 8 GPUs will be needed to support the development, fine-tuning, and inference of large-scale ML models - including transformers, LLMs, and computer vision models ranging from 50 million to over 100 billion parameters. The raw dataset volume is expected to be between 10–100 TB, with additional capacity required for intermediate data, model outputs, and user-generated analyses. To support scalable experimentation, distributed training, profiling, and hyperparameter tuning will be orchestrated using Ray clusters deployed on k8s.

The assistant's backend must meet demanding requirements for responsiveness, fault tolerance, and efficient data access under multi-user workloads. To that end, the system will be tightly integrated with the DEP's federated data infrastructure - including Rucio and FTS for dataset distribution, and containerized preprocessing pipelines for format harmonization and compression. Model serving and orchestration will be handled via a dedicated model hub running on Kubernetes, equipped with S3-compatible object storage, public IP routing, and ingress configuration for secure and scalable inference APIs. All services will be containerized to ensure reproducibility, ease of deployment, and compatibility across DEP nodes. These compute and data capabilities are critical to enable real-time AI interactions over large bioimage datasets and to deliver a robust assistant experience accessible directly from the BIA interface.

## 5. Test and Integration Pilots

Testbeds will be established to pilot and evaluate the WP3 solutions, building on the architectural definition in [Section 2](#) and incorporating updates from ongoing project discussions. These testbeds will be the key to collecting feedback from the use case partners such that the overall architecture can be further refined to address their needs. The vision in this respect is that these testbeds will provide the necessary playground that will actually enable the fruitful process of co-design.

The implementation of these testbeds will rely on the computing resources provided by TUBITAK and TU Wien. Of course, these resources may be supplemented by additional providers who, during the course of the project, will serve as early adopters and contribute with in-kind resources. In this context, the use case from DestinE may contribute by providing access to one of the EuroHPC systems.

From TUBITAK, an OpenStack-based cloud platform will be provided, which shall be backed by a small HPC cluster dedicated to the cloud system. Both the cloud and HPC systems will include GPU-enabled servers. While the cloud platform is aimed to provide continuous operation both in CPU- and GPU-based workloads, the HPC system will allow users to offload jobs that require more processing power and/or multiple servers. The HPC system will be SLURM-based and will be designed as an “accelerator” for time-limited tasks which need higher performance and more resources. The cloud computing platform will allow users to build their server infrastructures for extended workloads and services and serve them to external consumers. For the Cloud/HPC continuum, either SLURM native REST-based methods or more sophisticated software designed for this aim, such as interLink, can be used.

### 5.1. Early Demonstrators and Ongoing Testbed Integrations

In order to bootstrap the piloting activities that are key to the success of the WP3 architecture integration and enhancement, activities will be initiated, similar to those of the interTwin project. In particular, the planned testbeds can benefit from the fact that two components of Task 3.1, namely itwinai and interLink, have already demonstrated a basic integration. Similarly, early-stage integration between itwinai and yProv4ML has already been proven, which defines an initial link and solid foundation for future activities between Task 3.1 and Task 3.2.

itwinai has already been used in many scientific and technical use-cases, such as fast particle detector simulation at CERN, lattice QCD physics at ETH Zürich, tropical cyclone detection for CMCC, and many others. All the integrated use-case workflows are kept at the main itwinai repository and are freely available to anyone. A full list of use-cases currently integrated together

with the associated documentation can be found in the footnote<sup>27</sup>. In particular, it is relevant to note here the CMCC use case (defined in [Section 4.1](#)) that can already benefit from the experience gained from these earlier activities.

To start these developments and experiment on the RI-SCALE use-cases, TUBITAK resources on OpenStack could be used, and a k8s cluster will be deployed for itwinai. The cluster will be equipped with a Virtual kubelet, and an interLink instance dedicated to RI-SCALE will be deployed in order to enable the offloading to the HPC part of TUBITAK. This integrated testbed will be the first initial demonstrator and will be made available for testing as well as to support the other WP3 services that can benefit from such a model. In the future, testbeds will also be implemented with the TU Wien resources, which will be exploited for implementing the use cases from health and life science applications.

## 5.2. Roadmap for Future Setups

In the following months, the implementation towards a complete testbed to provide an integrated DEP platform will include the following main steps. The steps are grouped based on the DEP release timeline, where the steps are expected to be completed.

### DEP 1st Release (in M12)

- Extend other services from Tasks 3.1 and 3.2 to integrate with access to both Cloud and HPC resources, i.e. via interLink;
- Initial demonstration of use case integration through the testbed with WP3 software components;
- Initial demonstration of integration with services and components from WP2 and WP4, particularly AAI and Data Management (i.e. RUCIO). This is especially important to realize the DEP workflow;
- Initial demonstrator of integration of AI scaling framework (itwinai) with the model repository (such as the model hub) for model loading/uploading to the use case repository, as well as the yProv ecosystem (in particular Prov4ML);
- Initial demonstration of the AI Model Hub provides basic model listing and hosting.

### DEP 2nd Release (M24)

- Update the use case integration with the testbed;
- Update the integration of services and components from WP2 and WP4;
- Extend the testbed to TU Wien HPC and potentially additional providers, e.g. EuroHPC;

<sup>27</sup> [https://itwinai.readthedocs.io/latest/use-cases/use\\_cases.htm](https://itwinai.readthedocs.io/latest/use-cases/use_cases.htm)

- Update to integration of AI scaling framework (itwinai) with the model repository (such as the model hub) and the yProv ecosystem (in particular yProvStore and PID service).
- Include any early adopters not only as a scientific community but also as a resource provider, e.g. GROQ.

#### DEP 3rd Release (M36)

- Final integration of use cases;
- Final integration of services and components from WP2 and WP4;
- Leverage the existing BioImage.IO chatbot/agent and expand its functionality for the AI computing framework and DEP as a whole;
- Final integration of provenance support for AI model documentation through the yProv GUI, to advance the user's experience regarding provenance visualization and exploration, as well as AI models' interpretability;
- Explore the scalability of the testbeds to enable RIs to exploit large-scale compute and data infrastructure.

These testbeds and integration pilots will contribute to the implementation of the requirements defined in D5.1 [R1]. The functional and non-functional requirements from WP3 are provided in the [Annexure](#). In this table, the DEP release where these requirements will be delivered is specified. Depending on how critical a requirement is for the DEP and the associated use case, the associated release is prioritized



## 6. Summary and Next Steps

This deliverable provides the foundation for the development of AI-based technical solutions in the DEP. The AI capabilities are key to the RIs and allow them to tackle the data and compute challenges in the SUCs. The software solutions presented in this deliverable provide the basis for enabling these use cases in the DEP. It should be noted here that the connection of WP3 to the technical use cases is realized with the identified requirements (mostly non-functional). Considering these requirements, the developments in the software components are planned, and milestones in the roadmap are proposed.

This document presented user stories based on various kinds of DEP users (end user, model developer and DEP operator). These stories present workflows for executing user-specific tasks and the corresponding DEP responses for each of the steps in a workflow. Here, the AI-focused user stories were presented. The DEP responses highlighted the exploitation of various components of the “AI Lifecycle Management” container, presented in the WP3 architecture. The presented architecture is modular and flexible, allowing users to make customized definitions of their workflows. The “AI Lifecycle Management” container interacts with other containers in the DEP, which are the technical solutions developed in WP2 and WP4. At this stage of the project, these cross-WP interactions are not known in detail. However, the conceptual workflow that is presented in this deliverable is expected to be consistent with the final adopted solution.

The presented software solutions include components that bring together various functionalities in the “AI Lifecycle Management” container. These components provide the basis for the specification of the AI systems in the DEP. For each of these components, gaps were identified based on requirements gathered in D5.1 [R1]. Furthermore, the AI applications were also presented in this document. The main objective of these applications, the associated AI modelling scenarios and the compute and data challenges for each application were identified. These applications drive the requirements and guide the architectural design of WP3. The technology gaps and the technical challenges provided by the applications guide the developments that are still needed for the software solutions in order to be deployed in the DEP. Overall, these components bring together all the essential elements needed in an MLOps lifecycle. In the next steps, these individual components need to be integrated to deliver the “AI Lifecycle Management” container. The flexible nature of this container allows each of the components to retain their core functionalities.

Finally, the proposed testbeds and roadmap for WP3 were presented, which will contribute to the integration among the technical solutions and the use cases. These provide the methodological basis and the playground for the implementation of the AI technical solutions. These testbeds will be deployed in the next weeks in the infrastructure provided by RI-SCALE partners. Furthermore, discussions with other technical WPs (WP2 and WP4) will guide the overall implementation of the DEP. This will possibly involve workshops, hands-on sessions and focused pilots with other WPs. The

identified roadmap provides a timeline based on the DEP releases, where the testbeds and integrations will be provided.

Overall, the deliverable provides the detailed specification of the AI-based components that form the DEP. The technical solutions have already demonstrated promising performance across a wide range of applications and across diverse infrastructure. This deliverable presented the basic building blocks of these solutions that will contribute to the DEP and enable the RIs to scale up their scientific and technical use cases.

# References

Reference	
No	Description/Link
R1	Psychas, A., Spiliotopoulou, A., Tenhunen, V., & Sipos, G. (2025). RI-SCALE_D5.1 – Data Exploitation Platform Requirements and Design Considerations (V1_Under EC Review). Zenodo. <a href="https://doi.org/10.5281/zenodo.15755803">https://doi.org/10.5281/zenodo.15755803</a>
R2	Padovani, G. et al. (2024). A software ecosystem for multi-level provenance management in large-scale scientific workflows for AI applications. SC24-W: Workshops of the International Conference for High Performance Computing, Networking, Storage and Analysis, Atlanta, GA, USA, pp. 2024-2031. <a href="https://doi.org/10.1109/SCW63240.2024.00253">https://doi.org/10.1109/SCW63240.2024.00253</a>
R3	Cinquini, L. et al. (2014). The Earth System Grid Federation: An open infrastructure for access to distributed geospatial data. Future Generation Computer Systems, vol. 36, pp. 400–417. <a href="https://doi.org/10.1016/j.future.2013.07.002">https://doi.org/10.1016/j.future.2013.07.002</a>
R4	Fiore, S., Nassisi, P., Nuzzo, A., Mirto, M., Cinquini, L., Williams, D. & Aloisio, G. (2019). A Climate Change Community Gateway for Data Usage & Data Archive Metrics across the Earth System Grid Federation. In Proceedings of the 11th International Workshop on Science Gateways (IWSG 2019)", vol. 2975 of CEUR Workshop Proceedings, p. 6, CEUR, Ljubljana, Slovenia, 12-14 June 2019. URL: <a href="https://ceur-ws.org/Vol-2975/paper5.pdf">https://ceur-ws.org/Vol-2975/paper5.pdf</a> .
R5	BBMRI ERIC Colorectal Cancer Cohort. URL: <a href="https://www.bbMRI-eric.eu/scientific-collaboration/colorectal-cancer-cohort/">https://www.bbMRI-eric.eu/scientific-collaboration/colorectal-cancer-cohort/</a>
R6	Kirillov, A. et al. (2023). Segment Anything. Preprint at <a href="https://arxiv.org/abs/2304.02643">https://arxiv.org/abs/2304.02643</a> .
R7	Vandal, T., Kodra, E., Ganguly, S., Michaelis, A., Nemani, R. & Ganguly, A. (2017). DeepSD: Generating High Resolution Climate Change Projections through Single Image Super-Resolution. <a href="https://doi.org/10.48550/arXiv.1703.03126">https://doi.org/10.48550/arXiv.1703.03126</a> .
R8	Baño-Medina, J., Manzanar, R., & Gutiérrez, J. M. (2020). Configuration and intercomparison of deep learning neural models for statistical downscaling, Geosci. Model Dev., 13, 2109–2124. <a href="https://doi.org/10.5194/gmd-13-2109-2020">https://doi.org/10.5194/gmd-13-2109-2020</a> .
R9	Lei, W., Fuster-Barceló, C., Reder, G. et al. BioImage.IO Chatbot: a community-driven AI assistant for integrative computational bioimaging. Nat Methods 21, 1368–1370 (2024). <a href="https://doi.org/10.1038/s41592-024-02370-y">https://doi.org/10.1038/s41592-024-02370-y</a>



# Annexure

The annexure contains the WP3-specific requirements submitted in Deliverable 5.1 - Data Exploitation Platform [R1] requirements and design considerations. Additionally, in the tables below, the release date for these requirements is also specified.

## Functional Requirements

Requirement Jira Key	Requirement Source	Description	Rationale	Component that fulfils it	Priority	Release Date
<a href="#">RSREQ-10</a>	[ITT] Internal – Technical Team	The AI computing framework must support large-scale offloading of AI training and inference seamlessly across cloud and HPC infrastructure.	This ensures that the AI framework in the DEP is readily portable, which allows RIs to deploy it in their local machine, cloud provider or a large HPC center. It contributes directly to KPI#3, KPI#4 and KPI#7.	AI Computing Framework (WP3-T3.1)	MUST	M24
<a href="#">RSREQ-12</a>	[ITT] Internal – Technical Team	The DEP must have a user-friendly interface that allows for seamless application and deployment for the use cases.	The value of a technology is determined by how effectively it is used in practice. Functionality and utility of the DEP, delivered by the Technical Team, must be sustained (project duration and beyond) and widespread (use-cases) application by the Scientific and Technical Use Cases and the	AI Computing Framework (WP3-T3.1)	SHOULD	M36



			associated end users.			
<a href="#">RSREQ-13</a>	[ITT] Internal – Technical Team	The AI Model Hub component must provide REST APIs and a basic Web UI for core model lifecycle operations (upload, discovery, versioning, retrieval) using Hypha and MLflow as a backend. It must enable the storage and API-based retrieval of key provenance metadata (e.g., creator, date, dataset reference, parameters/environment, license) with each model version, and offer interfaces to initiate model benchmarking.	Fulfills the task requirements for delivering an AI Model Hub capable of storing, serving, and benchmarking models, incorporating a provenance model, and leveraging Hypha and MLflow. This supports model sharing, reproducibility, transparency, and trustworthiness, contributing to project goals and potential KPIs related to model management and usage.	AI Model Hub (WP3-T3.2)	MUST	M24
<a href="#">RSREQ-14</a>	[ITT] Internal – Technical Team	The AI Model Hub must integrate with the central RI-SCALE Authentication and Authorisation Infrastructure (AAI), anticipated to be the Policy-Based Authorization Framework (WP4-T4.1), to enforce access control. Authorization decisions for all Hub functionalities	Directly addresses the task requirement to establish Authentication/Authorisation mechanisms for enforcing model access policies. This ensures secure and governed access to AI models within the Hub, aligning with project security requirements	AI Model Hub (WP3-T3.2)	MUST	M24



		and resources must be governed by policies managed within this AAI.	and potential KPIs for controlled resource access and compliance.			
<a href="#">RSREQ-15</a>	[ISU] Internal – Scientific Use Case (Scientific UC)	The platform must support foundational model training and inference execution on data from the BioImage Archive	This functionality is essential for Scientific use case 7 (Foundational models for heterogeneous biological image data). Also, to train foundation models for heterogeneous image data, as required on T3.4; It directly contributes to KPI#7 (No. of AI models trained in DEP pilots) and KPI#8 (No. of use cases developed for DEP validation).	AI for Health and Life Sciences (WP3-T3.4)	MUST	M24
<a href="#">RSREQ-16</a>	[ISU] Internal – Scientific Use Case (Scientific UC)	The platform must enable fine-tuning models on data from the BioImage Archive	This functionality is essential for Scientific use case 7 (Foundational models for heterogeneous biological image data). Also, to fine-tune models for image classification, segmentation and anomaly detection, as required on T3.4, it directly contributes to KPI#8 (No. of use cases developed	AI for Health and Life Sciences (WP3-T3.4)	MUST	M24



			for DEP validation).			
<a href="#">RSREQ-17</a>	[ISU] Internal – Scientific Use Case (Scientific UC)	The platform must support the development, training, and deployment of ML models to learn patterns in data usage and transfer failures, and predict changes/anomalies as required for the use case in T3.3.	This functionality is needed for implementing the use case in WP3.3 as well as validating the use case in WP5.2. It contributes KPI#7 (No. of AI models trained in DEP pilots) and KPI#8 (No. of use cases developed for DEP validation). Moreover, it is linked to KPI#4 (No. of AI models offered within DEPs).	AI Computing Framework (WP3-T3.1)	MUST	M24
<a href="#">RSREQ-20</a>	[ISU] Internal – Scientific Use Case (Scientific UC)	The platform must support the development, training, and deployment of ML models (probably CNNs) to perform statistical downscaling of climate projections at higher spatial resolution.	This functionality is needed for the implementation of one of the scientific use cases in WP3.3, and for the validation of the use case in WP5.2. It contributes to KPI#7 (No. of AI models trained in DEP pilots) and KPI#8 (No. of use cases developed for DEP validation). Moreover, it is linked to KPI#4 (No. of AI models offered within DEPs).	AI for Environmental Science (WP3-T3.3)	MUST	M24
<a href="#">RSREQ-26</a>	[ISU] Internal – Scientific Use Case	The DEP should integrate with the Itwinai workflow orchestration	Hyperparameter optimization is essential for developing AI models.	AI Computing Framework	SHOULD	M36



	(Scientific UC)	<p>framework to support HPO for AI models. Users should be able to define HPO tasks as part of their model training workflows and execute them on DEP compute infrastructure.</p> <p>The HPO framework should allow users to:</p> <ul style="list-style-type: none"> <li>• Define a search space for hyperparameters (e.g. learning rate, batch size, ...)</li> <li>• Choose from standard optimization strategies, such as grid search, random search, and Bayesian optimization.</li> <li>• Specify the objective metric to optimize (e.g. validation accuracy, AUC).</li> <li>• Set constraints on resource usage, including maximum number of trials, parallel runs, or GPU-hour budgets.</li> <li>• Enable early stopping of</li> </ul>	<p>Supporting early stopping helps minimize waste of computing resources by stopping poorly performing trials early. This will be used for Use Case 5 and Use Case 6, and contributes to KPI03: No. of AI frameworks/toolboxes offered within DEPs.</p>	(WP3-T3.1)		
--	-----------------	--	---	------------	--	--





		<p>unpromising trials based on intermediate results to save computational resources. Supported strategies should include Median stopping rule, Asynchronous Successive Halving (ASHA), and Hyperband.</p> <ul style="list-style-type: none"> <li>Automatically deploy training jobs for each HPO trial to the DEP, without requiring manual submission by the user.</li> </ul>				
<a href="#">RSREQ-27</a>	[ISU] Internal – Scientific Use Case (Scientific UC)	<p>The DEP should support the use of MLflow to track AI model training workflows executed within the platform. Users should be able to log metadata about each training run directly to an MLflow tracking server provided by the DEP. This tracking should include:</p> <ul style="list-style-type: none"> <li>Hyperparameters, such as learning rate, batch size, and number of epochs.</li> </ul>	<p>MLflow provides an established mechanism for capturing and organising metadata during model training, which is essential for reproducibility, comparison of experiments, and responsible model development. For WSI-based AI workflows, where models may require significant tuning and iteration, the ability to track all aspects of training is</p>	AI Computing Framework (WP3-T3.1)	SHOULD	M36



		<ul style="list-style-type: none"> <li>Performance metrics, such as accuracy, AUC, loss, or task-specific indicators.</li> <li>Artifacts, such as trained model files, logs, and visualisations.</li> <li>Environment metadata, including the Git commit hash of the training code and the container image identifier used for the run.</li> <li>Dataset references, including internal DEP dataset IDs.</li> </ul> <p>The tracking functionality should be accessible programmatically from within the training code (e.g. via the mlflow Python client) and operate seamlessly with training jobs launched on all DEP compute sites. Access to the web interface of MIFlow should be controlled through the user management of the DEP. It should be asserted that users only have access to training</p>	<p>critical. This requirement supports model development in Use Cases 5 and 6, and contributes to KPI03: No. of AI frameworks/toolboxes offered within DEPs.</p>			
--	--	---	--	--	--	--



		logs and metadata of projects they are associated with.				
<a href="#">RSREQ-28</a>	[ISU] Internal – Scientific Use Case (Scientific UC)	<p>The AI Model Hub should let users launch inference on any supported data modality via web UI or API. Inputs include dataset IDs, a chosen model, and run-time parameters (e.g., resolution, tile size). Workloads are sharded into parallel jobs (e.g., via itwinai) to scale on large datasets. The service must honour DEP’s user-management and ACL rules, ensuring users access only authorised data and models. Results are cached and reused whenever the same model, parameters and data recur, regardless of project or requester. Foundation models are integrated, and external ones requiring Hugging Face keys use the caller’s key at run time.</p>	Simple and scalable WSI inference through the AI Model Hub is a key component of the DEP and essential for a good user experience. Foundation models will be used in Use Case 5, and the requirement contributes to KPI04: Number of AI models offered within DEPs.	AI Model Hub (WP3-T3.2)	MUST	M36



<a href="#">RSREQ-31</a>	[ISU] Internal – Scientific Use Case (Scientific UC)	<p>In the DEP, algorithms should be developed for survival prediction based on Whole Slide Image (WSI) data. These algorithms should estimate patient outcomes using visual patterns in histopathological slides. The models should be implemented using reproducible workflows and make use of the training and orchestration tools provided by the DEP compute infrastructure. The developed models should produce predictions that are interpretable, allowing validation and assessment by pathologists.</p> <p>To ensure scientific relevance and robustness, validation should be performed:</p> <ul style="list-style-type: none"> <li>On publicly available survival prediction benchmarks to demonstrate the novelty and competitiveness of the</li> </ul>	<p>Developing survival prediction algorithms within the DEP highlights how RI data can be leveraged to address clinically relevant research challenges. The colorectal cancer use case demonstrates this approach in practice and contributes directly to RI-SCALE Use Case 5 and KPI#4 (No. of AI models offered within DEPs). Furthermore, the potential identification of novel biomarkers through model interpretation and validation may result in new scientific insights, contributing to KPI21: No. of peer-reviewed scientific publications and KPI22: No. of research outputs.</p>	AI for Health and Life Sciences (WP3-T3.4)	SHOULD	M36
--------------------------	--	---	--	--	--------	-----



		<p>developed algorithms against the current state of the art.</p> <ul style="list-style-type: none"> <li>On the colorectal cancer cohort available within the DEP, which includes WSIs of lymph node tissue, provided by MUG and MMCI, with the aim of supporting the potential identification of novel biomarkers.</li> </ul>				
<a href="#">RSREQ-32</a>	[ISU] Internal – Scientific Use Case (Scientific UC)	<p>Algorithms should be developed for generating synthetic histopathological images using generative AI models, such as diffusion-based approaches. These models should be trained on real pathology data available within the DEP.</p> <p>To ensure scientific relevance and output quality, validation of the generated synthetic images should include:</p>	<p>This requirement contributes to use case 6 and supports KPI#4 (No. of AI models offered within DEPs), KPI#21 (No. of peer-reviewed scientific publications), and KPI#22 (No. of research outputs).</p>	AI for Health and Life Sciences (WP3-T3.4)	SHOULD	M36



		<ul style="list-style-type: none"> <li>Privacy risk assessment, e.g. membership inference testing, to confirm that synthetic images do not expose identifiable patient information.</li> <li>A user study with pathologists, who will review synthetic images and assess their realism, diagnostic plausibility, and fitness for research or training.</li> </ul>				
<a href="#">RSREQ-40</a>	[ITU] Internal – Technical Use Case (Technical UC)	AIFS should be able to use at least some of the distributed ML frameworks on at least one of the EuroHPC systems that DestinE is currently using, Lumi (CSC), Leonardo (Cineca), MN5 (Bsc)	<p>The technical use case states: Evaluate and measure the scale at which the DEP can work on a EuroHPC machine while serving a challenging AI model with large data from DestinE.</p> <p>Hence, AIFS should be able to use at least some of the distributed ML frameworks on at least one of the EuroHPC systems that DestinE is currently using, Lumi (CSC), Leonardo (Cineca), MN5 (Bsc)</p>	AI Computing Framework (WP3-T3.1)	MUST	M12



<a href="#">RSREQ-87</a>	[ISU] Internal – Scientific Use Case (Scientific UC)	The DEP must enable to send any analysis outputs and model metadata back to the BIA. Model outputs and metadata must be returned to the BIA using a consistent schema so they can be used to support automated data categorisation and similarity search within the RI	This functionality is essential for Scientific use case 7 (Foundational models for heterogeneous biological image data). It directly contributes to KPI#8 (No. of use cases developed for DEP validation).	AI for Health and Life Sciences (WP3-T3.4)	MUST	M24
<a href="#">RSREQ-94</a>	[ISU] Internal – Scientific Use Case (Scientific UC)	The assistant should enable users to perform basic bioimage analysis tasks—such as segmentation or object detection—through a conversational interface. Users should be able to upload images or select existing datasets, request a supported analysis (e.g., “Segment cells using CellPose”), and receive both visual and downloadable results.	Many users accessing RI imaging data do not have the technical expertise to deploy AI models or run analysis workflows independently. This requirement aims to provide a simplified interface for triggering standard analysis tasks, improving usability and promoting broader adoption of AI tools in image-based research.	AI Model Hub (WP3-T3.2)	SHOULD	M24
<a href="#">RSREQ-95</a>	[ISU] Internal – Scientific Use Case (Scientific UC)	The project must provide an ML-based model for enabling prediction and/or detection of	This functionality is needed for implementing the use case in WP3.3 as well as validating the use	AI for Environmental Science	MUST	M24



		changes/anomalies in the usage of climate data from ESGF (Earth System Grid Federation). ML models can be effectively used to learn from historical data usage and transfer log patterns associated with usage and anomalies. The trained model will then be applied to current data usage streams, for example, to identify the most used data in a given period or react based on anomaly detection of high loads in the data downloads.	case in WP5.2. It contributes KPI#7 (No. of AI models trained in DEP pilots) and KPI#8 (No. of use cases developed for DEP validation). Moreover, it is linked to KPI#4 (No. of AI models offered within DEPs).	(WP3-T3.3)		
<a href="#">RSREQ-96</a>	[ISU] Internal – Scientific Use Case (Scientific UC)	The platform must support the development, training, and deployment of ML models to learn spatial patterns in climate projections and predict high-resolution climate scenarios as required for the use case in T3.3.	This functionality is needed for implementing the use case in WP3.3 as well as validating the use case in WP5.2. It contributes KPI#7 (No. of AI models trained in DEP pilots) and KPI#8 (No. of use cases developed for DEP validation). Moreover, it is linked to KPI#4 (No. of AI models offered within DEPs).	AI Computing Framework (WP3-T3.1)	MUST	M36





## Non-Functional Requirements

Requirement Jira Key	Requirement Source	Description	Rationale	Component that fulfils it	Priority	Release Date
<a href="#">RSREQ-11</a>	[ITT] Internal – Technical Team	The AI computing framework should be able to integrate all RIs and scale efficiently (>80%) to more than 100 GPUs.	This benchmark demonstrates the versatility of the AI framework and its large-scale training capabilities. It contributes directly to KPI#4 and also to KPI#6, to enable training with large-scale datasets.	AI Computing Framework (WP3-T3.1)	SHOULD	M36
<a href="#">RSREQ-19</a>	[ISU] Internal – Scientific Use Case (Scientific UC)	The ML model should be efficient enough to be potentially applied for real-time detection of anomalies in data usage streams when deployed in the infrastructure.	Real-time prediction can support early detection of changes in data usage patterns. This could allow RI managers to deal with high loads in data download patterns and react to potential issues connected with data transfer failures in a timely manner.	AI for Environmental Science (WP3-T3.3)	COULD	M36
<a href="#">RSREQ-29</a>	[ISU] Internal – Scientific Use Case (Scientific UC)	Users of the DEP should be able to automatically deploy an instance of the WSI visualization toolkit XOpac. The viewer is deployed in an Apptainer container. In this	User experience is improved through seamless interaction of WSIs and algorithm results directly in the DEP platform. Having an in-platform viewer is essential, as it	AI Computing Framework (WP3-T3.1)	SHOULD	M36



		<p>container, directories for the following data should be mounted: WSIS that the user has access to, the algorithm output of the user, and a directory where the viewer saves annotations. The algorithm output should be writable by compute jobs while the viewer is running, enabling real-time monitoring of results during training. The viewer's web interface is exposed through a URL, where access is controlled through the user management of the DEP.</p> <p>XOpat enables users to:</p> <ul style="list-style-type: none"> <li>• Zoom, pan, and navigate in gigapixel-sized WSIs with minimal latency by streaming only the visible viewport instead of full images.</li> <li>• Create pixel-level annotations on WSIs, which can be used as training data for algorithms.</li> </ul>	<p>is impractical to download terabyte-scale datasets to local machines for visualisation. Moreover, it ensures sensitive data remains within the secure DEP environment while still allowing users to interact with it effectively. XOpat will be used in the scientific validation (Task 5.2) of Use Cases 5 and 6, and contributes to KPI03: No. of AI frameworks/toolboxes offered within DEPs.</p>			
--	--	---	---	--	--	--



		<ul style="list-style-type: none"> <li>Visualize algorithm outputs on WSIs, such as heatmaps or segmentations.</li> </ul>				
<a href="#">RSREQ-86</a>	[ITU] Internal – Technical Use Case (Technical UC)	Advanced image compression for histopathology Whole Slide Images should be investigated. The goal is to enhance DEP storage and transfer efficiency without harming diagnostic value or model accuracy. In particular, JPEG2000 should be evaluated, focusing on the effects of AI training, inference, and processing speed.	Histopathological images, particularly WSIs, require large storage and are expensive to transfer across infrastructures. Efficient compression reduces system load and speeds up data access, while preserving image quality is essential for clinical validation and trustworthy AI development. This technical use case contributes to optimising the DEP's performance and supports sustainability, interoperability, and usability across health science domains.	AI for Health and Life Sciences (WP3-T3.4)	SHOULD	M12
<a href="#">RSREQ-102</a>	[ITU] Internal – Technical Use Case (Technical UC)	The GROQ cards being tested need to be accessible and usable via appropriate interfaces and/or software elements	To use GROQ cards in inference, we assume the software layers need to be adapted.	AI Computing Framework (WP3-T3.1)	SHOULD	M24