RI SCALE

D5.2

Validation Plan for Use Case Scenarios

Status: Under EC Review

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	
Keywords	Validation Plan, Use Case Scenarios, Data Exploitation Platform, KPIs, Traceability, Interoperability

This deliverable presents the Validation Plan for the RI-SCALE Data Exploitation Platform (DEP). It defines the framework, methodology, and structure guiding the validation of both functional and non-functional requirements consolidated in D5.1. The plan establishes a clear link between the DEP development activities, the scientific and technical use cases, and the project's Key Performance Indicators (KPIs). Validation is organised around twelve use case scenarios that represent the full range of operational and scientific contexts within RI-SCALE. Each scenario defines specific test cases and acceptance criteria to ensure measurable and transparent verification of the platform's performance and interoperability. The deliverable also introduces the mechanisms for requirement-to-scenario mapping, traceability, and continuous update through Confluence and Jira, ensuring adaptability throughout the project lifecycle. This work provides the operational foundation for the validation cycles that will follow, leading to the reporting of results in D5.4 and D5.6.

Revision History

Version	Date	Description	Author/Reviewer
V 0.1	02/09/2025	First draft establishing the initial structure and content outline for D5.2.	Alexandros Psychas (JNP) Athanasia Spiliotopoulou (JNP)
V 0.2	20/09/2025	Initial contributions, including validation framework definition and use case structure.	Alexandros Psychas (JNP) Athanasia Spiliotopoulou (JNP)
V 0.3	06/10/2025	Integration of all validation scenarios, requirement mappings, and KPI alignment tables.	Alexandros Psychas (JNP) Athanasia Spiliotopoulou (JNP)
V 0.4	10/10/2025	Version for Internal Review	Alexandros Psychas (JNP) Athanasia Spiliotopoulou (JNP)
V 0.5	21/10/2025	Version with comments	Ville Tenhunen (EGI)

		and suggestions from reviewers	Martin Barisits (CERN)
V 1.0	24/10/2025	Final version for Submission	Alexandros Psychas (JNP) Athanasia Spiliotopoulou (JNP)

Document Description			
D5.2 – Validation Plan for Use Case Scenarios			
Work Package Number 5			
Document Type	Deliverable		
Document Status	Under EC Review	Version	1.0
Dissemination Level	Public		
Copyright Status	This material by the Parties of the RI-SCALE Consortium is licensed under a [Creative Commons Attribution 4.0 International License](https://creativecommons.org/licenses/by/4.0/).		
Lead partner	JNP		
Document Link	https://documents.egi.eu/document/4218		
DOI	https://zenodo.org/records/17494163		
Author(s)	• Alexandros Psychas (JNP) • Athanasia Spiliotopoulou (JNP)		
Reviewers	• Ville Tenhunen (EGI) • Martin Barisits (CERN)		
Moderated by:	• Matteo Agati (EGI)		
Approved by:	Technical Coordination Board		

Terminology / Acronyms	
Term/Acronym	Definition
AAI	Authentication and Authorization Infrastructure
AIFS	AI for Science
API	Application Programming Interface
ASC	Austrian Scientific Computing
BBMRI	Biobanking and Biomolecular Resources Research Infrastructure
BIA	Biomedical Image Analysis
CERRA	Copernicus European Regional ReAnalysis
CNN	Convolutional Neural Network
CORDEX	Coordinated Regional Downscaling Experiment
CPU	Central Processing Unit
CRC	Colorectal Cancer
CRMS	Credit Resource Management System
DEM	Digital Elevation Models
DoA	Description of Action
DEP	Data Exploitation Platform
EISCAT	EISCAT Scientific Association
ENES	European Network for Earth System Modelling
ESGF	Earth System Grid Federation
EURO	European Regional Domain (within CORDEX)
GPU	Graphics Processing Unit
GROQ	Generalized Representation for Object Query
HPC	High Performance Computing

ISR	Incoherent Scatter Radar
JPEG	Joint Photographic Experts Group
JSON	JavaScript Object Notation
KPI	Key Performance Indicator
KTH	Kungliga Tekniska Högskolan (KTH Royal Institute of Technology)
LLM	Large Language Model
LPU	Logical Processing Unit
LTU	Luleå Tekniska Universitet (Luleå University of Technology)
ML	Machine Learning
MVP	Minimum Viable Product
PDF	Portable Document Format
REA	European Research Executive Agency
RI	Research Infrastructure
RSREQ	RI-SCALE Requirements (Jira Project Key)
SUC	Scientific Use Case
TCB	Technical Coordination Board
TUC	Technical Use Case
VHR	Very High Resolution
WSI	Whole Slide Image

Table of Contents

List of Figures

List of Tables

Executive Summary

The present deliverable defines the Validation Plan for the Data Exploitation Platform (DEP) and its associated AI components within the RI-SCALE project. It builds directly upon the consolidated requirements and design considerations presented in D5.1, translating them into a structured and operational framework for validation. The plan defines how each requirement will be verified through measurable test cases, acceptance criteria, and use case–driven scenarios, ensuring that all developments are traceable, testable, and aligned with project KPIs.

The validation activities are organised around twelve use case scenarios, eight *Scientific Use Cases (SUCs)* and four *Technical Use Cases (TUCs)*, representing the scientific and technological domains covered by the project. These scenarios provide the backbone of the validation framework, capturing the interactions between end users, DEP components, and external systems. Each scenario is described using a consistent template, defining actors and roles, involved components, preconditions, validation steps, and expected outcomes. Following this structured process, the use case leaders produced the scenarios defined in this deliverable.

The deliverable introduces a scenario-based validation methodology, which connects individual requirements to concrete test activities. Requirements consolidated in D5.1[1] have been mapped to the relevant use cases, ensuring that validation is not only component-oriented but also reflects real operational workflows. Each validation step is linked to one or more acceptance criteria, allowing the project to systematically verify the implementation of functionalities and quality attributes such as performance, interoperability, and security. To maintain alignment between technical development and project objectives, the validation process is tightly integrated with the DEP architecture and the work of WP2, WP3, and WP4. It ensures that each major component, data lifecycle management, scalable AI frameworks, and access management technologies, is validated both individually and within end-to-end scenarios. The plan also supports traceability to project KPIs, providing a discrete link between system performance and the broader impact expected from the RI-SCALE initiative.

Validation will be conducted iteratively, following the platform's release roadmap. The first validation cycle will focus on the verification of baseline functionalities and interoperability, leading to the First Validation Report (D5.4). The second cycle will expand validation to include advanced and cross-domain features, culminating in the Final Validation Report (D5.6). Throughout both cycles, results and findings will be collected via Confluence and Jira, ensuring consistent documentation and traceability.

[1] https://zenodo.org/records/15755803

1. Introduction

Deliverable D5.2 defines the operational plan for validating the DEP and its supporting AI services within the RI-SCALE project. It provides the framework that will guide the verification of all system functionalities and quality aspects through structured validation activities carried out in WP5.

This document does not repeat the technical or requirement definitions already covered in D5.1. Instead, it incorporates those outcomes into a coordinated validation process that specifies how and when (in accordance with DEP Releases) the platform and its components will be tested, who is responsible for each activity, and which criteria will determine successful verification.

1.1. Scope and Purpose of the Deliverable

The primary purpose of D5.2 is to define a unified, transparent, and measurable framework for validating the DEP. It establishes the methods and organisational structure that will be used to verify the system's compliance with the requirements defined in D5.1 and to ensure that validation activities are systematically planned and executed across all project domains.

Beyond defining validation procedures, this deliverable also aims to bridge the DEP development work with the use case implementations, ensuring that technical components and scientific objectives evolve in parallel. The validation framework enables this by introducing structured scenario flows that connect platform functionalities to concrete use case goals, ensuring that both development integration and use case actuation proceed in a coordinated and traceable manner.

1.2. Relationship to other WPs and Deliverables

D5.2 is positioned between the requirement specification and the validation execution. It directly builds on the outcomes of D5.1 and provides the procedural foundation for D5.4 and D5.6, which will report the validation results.

Table 1: D5.2 Relationship to other WP and Deliverables

D5.2 Relationship to Other WPs and Deliverables		
WP / Deliverable	**Focus**	**Relationship to D5.2**
WP2–WP4 (D2.1, D3.1, D4.1)	Technical development of DEP components	Provide the software, data, and services to be validated according to this plan.
D5.1 – DEP Requirements and Design Considerations	Requirement consolidation and specification	Serves as the baseline for validation planning and defines the features to be verified.

D5.3 – First DEP Release / D5.5 – Second DEP Release	Platform implementation cycles	Each release will be tested against the validation plan and acceptance criteria established here.
D5.4 – First Validation Report / D5.6 – Final Validation Report	Execution and reporting of validation results	Build directly on the methodology and structure defined in this deliverable.
WP6 – Co-Design and Stakeholder Engagement	External input and user feedback	Provides complementary validation perspectives from RIs and external users.

1.3. Structure of the Deliverable

This document is organised as follows:

- **Section 2** presents the validation methodology and overall framework, describing the objectives, guiding principles, and processes that define how validation will be conducted across the project. It outlines the roles, tools, and procedures that ensure consistency and traceability throughout the validation lifecycle.
- **Section 3** provides an overview of the scientific and technical use case scenarios guiding validation, summarising their purpose, structure, and connection to the DEP components and requirements. It establishes the practical basis for scenario-driven validation.
- **Section 4** explains the requirement-to-scenario mapping and validation analysis, showing how the consolidated requirements from D5.1 are linked to specific validation steps and use case flows. It also analyses the coverage and alignment between system functionalities and user objectives.
- **Section 5** describes how validation results will be linked to project KPIs, detailing the traceability approach and defining how measurable outcomes will be used to assess project progress and impact.
- **Section 6** concludes with the next steps and outlines the transition toward validation execution and reporting. It defines the timeline, expected outputs, and the connection to upcoming deliverables D5.4 and D5.6.

2. Validation Planning Methodology and Framework

This section outlines the general approach and methodology guiding the validation activities. It defines the structure, principles, and coordination mechanisms that ensure the process is consistent, traceable, and aligned with the project objectives.

2.1. Validation Objectives

The plan sets out how the DEP and its components will be assessed in realistic conditions, ensuring that validation activities are transparent, measurable, and aligned with project goals. Scientific and technical use case scenarios provide the backbone of this framework, capturing flows of user actions, involved tools, and system components. By describing these scenarios in a consistent way, the project reduces ambiguity, steers development, and ensures that later validation can be carried out in a systematic manner.

The main objectives of the validation planning are:

1. **Establish a coherent validation framework**
   Define how validation will be organised, including scope, roles, preconditions, and reporting mechanisms. Ensure that the framework is actionable across work packages and feasible within project milestones.
2. **Scenario-based validation and alignment with technical developments**
   Use the eight scientific (SUC) and four technical (TUC) use case scenarios as the organising structure for validation. Each scenario provides a detailed description of actors, DEP components, preconditions, and flows of interaction. Requirements consolidated in D5.1 are mapped to these scenarios, ensuring that validation connects with specific system components requirements and end-to-end workflows developed in WP2, WP3, and WP4.
3. **Traceability to KPIs**
   Maintain a clear chain from project KPIs to requirements, from requirements to scenarios, and from scenarios to testable steps. In this way, validation outcomes provide measurable evidence of progress towards the project's expected impact.
4. **Definition of acceptance criteria and evidence collection**
   Translate scenario flows into precise acceptance criteria, such as performance thresholds, compliance checks, or usability expectations. Define how evidence will be collected and reported, ensuring transparency and comparability across validation activities.
5. **Iterative validation across project phases**
   Conduct validation in cycles aligned with project milestones. Early cycles will confirm the

feasibility of core functions, while later iterations will extend to advanced features and integrated workflows. This staged approach, reflected in D5.2 (Planning), D5.4 (First Report), and D5.6 (Final Report), ensures continuous feedback between use cases, development teams, and validation activities.

In summary, the validation planning provides a scenario-driven framework that ensures clarity, consistency, and traceability. It establishes how use case scenarios, requirement mapping, and acceptance criteria will be brought together to verify that the DEP evolves in line with user needs, system design, technical feasibility, and project KPIs.

2.2. Methodology and Scenario-Based Framework

The validation methodology in RI-SCALE is based on structured scenario flows that connect requirements, use case execution, and project KPIs. This scenario-driven approach ensures that validation is performed in realistic conditions, capturing both user interaction with the platform and the behaviour of underlying technical components.

The **use case scenarios** form the core of the validation framework. Each of the twelve scenarios (eight scientific and four technical) is described using a uniform template that includes:

- **Use Case ID and Name**: Unique identifier (SUC or TUC) and official title.
- **Description/Purpose**: Overview of the use case objective and relevance.
- **Actors and Roles**: Generic users (e.g. researcher, administrator), system functions, and external services.
- **System Components Involved**: DEP modules and WP2–WP4 services engaged in the scenario.
- **Preconditions**: Conditions that must be met before execution.
- **Scenario Flow**: Structured sequence of actions linking actors, tools, and data.
- **Expected Outcomes and Validation Reporting**: Acceptance criteria and measurable success conditions for each step.
- **Related Requirements**: Mapping of requirements from D5.1 that are validated through the scenario flow.

This template (full template presented in Annex I: Use Case Scenario Description Template) provides a clear and replicable structure for validation, avoiding ambiguity and ensuring traceability from requirements to outcomes.

To support transparency and project-wide collaboration, the process is managed through Jira and Confluence. Requirements are tracked as Jira issues, linked to the scenarios where they will be validated. Confluence pages are used to document scenario descriptions, actors, and flows, ensuring shared visibility across partners. Together, these tools provide the operational backbone for requirement mapping, version control, and reporting.

Validation planning incorporates roles and responsibilities already defined in the requirement lifecycle:

- **Use Case Leaders** define and refine scenarios' content (all the scenarios are presented in Section 3).
- **Technical teams (WP2–WP4)** ensure that system components and services are correctly represented.
- **Task 5.1 and WP5** oversee requirement mapping and traceability.
- **The Technical Coordination Board (TCB)** supervises the activity of the design, implementation and validation of the Data Exploitation Platform.

After the successful establishment of the scenario structure and content presented in this deliverable, responsibilities for validation execution are distributed as follows:

- **Use Case Leaders** coordinate and execute the validation activities within their assigned scenarios. They are responsible for running the defined tests, tracking progress against the acceptance criteria, and recording the results in the designated documentation tools.
- **Task 5.2 (scientific validation) and Task 5.3 (technical validation)** Leaders provide methodological guidance, ensuring that scientific validation and AI model evaluation follow a consistent process and share a common reporting structure.
- **Technical teams (WP2–WP4)** support the execution of the scenarios by ensuring that the necessary components and services are available, functioning, and correctly integrated into the validation workflow.

2.3. Integration with DEP Architecture and WP Contributions

Validation activities in RI-SCALE are directly linked to the architecture and development work of the DEP. The DEP is designed as a modular system, with contributions from the core technical work packages. The validation plan ensures that these contributions are not assessed in isolation but within end-to-end workflows, as represented by the scientific and technical use case scenarios.

The main areas of technical development, and their corresponding validation focus, are:

- **WP2 – Data Lifecycle Management**
  Validation covers the ability of the DEP to replicate, prepare, and manage datasets from

research infrastructures. This includes data discovery, integration of heterogeneous data holdings, and orchestration of resources to enable AI-driven processing.

- **WP3 – Scalable AI Solutions**
  Validation focuses on the model hub, execution of inference tasks, training and re-training of AI models, and integration of domain-specific frameworks. Scenarios ensure that these functions are not only operational but also interoperable with data and access components.
- **WP4 – Access Management Technologies**
  Validation addresses trust, security, and identity management services that enable the secure operation of the DEP. This also includes credit management, which allows monitoring of resource consumption and the enforcement of usage policies.

3. Use Case Scenario Overview

This section introduces the twelve use case scenarios that guide validation in RI-SCALE: eight scientific use cases (SUCs) and four technical use cases (TUCs), as defined in the Description of Action (DoA). Each scenario provides a structured description of user interactions, DEP components, and expected outcomes, ensuring that validation activities are carried out in a consistent and comparable way across domains (Annex I: Use Case Scenario Description Template). The development of these twelve use case scenarios required substantial coordination and iterative refinement across partners. Initial draft versions were prepared by the respective use case leaders, followed by an extensive internal review process to ensure consistency in roles, component references, and scenario flows. Particular attention was given to aligning the content with the official scenario template and the technical terminology introduced in the corresponding development deliverables (D2.1[2], D3.1[3], and D4.1[4]). Through several feedback rounds and validation checkpoints, each scenario was reviewed and approved by the technical leaders of WP2, WP3, and WP4. Following this coordinated review, the scenarios reached their final form and are presented in the sections below.

3.1. Scientific Use Case Scenarios (SUC-1 to SUC-8)

SUC1: High-resolution Downscaling of Climate Scenarios and Risk Trend Analysis in Agriculture

Use Case ID: SUC 1

Description/Purpose

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 the Coupled Model Intercomparison Project Phase 6 (CMIP6) and EURO-CORDEX, are provided at much coarser resolutions (10–100 km), which are insufficient for regional or local-scale decision-making. Achieving such fine resolutions with traditional physics-based climate models is practically infeasible, as it would require enormous computational resources and simulation times. In contrast, AI-based approaches provide a compelling solution: while they require significant resources during training, their inference phase is computationally efficient, enabling the generation of high-resolution climate data at scale.

[2] https://zenodo.org/records/16993600
[3] https://zenodo.org/records/16993122
[4] https://zenodo.org/records/16994162

To bridge this gap, the RI-SCALE project aims to develop a novel infrastructure for downscaling these 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, will be evaluated and fine-tuned within the project. These models will be evaluated and adapted in the context of well-established approaches commonly used in statistical downscaling.

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.

Actors and Roles

- **Data scientists:** primary actors initiating the use case, performing training of the AI model, and evaluating model outputs on a technical level.
- **Domain Expert**: is responsible for maintaining the downscaling algorithm in case of problems.
- **DEP Administrator:** provides HPC compute resources and components required to run the model.

System Components Involved

Table 2: SUC1 Software Components

List of UC Components and their Role	
Component Name	**Role/Involvement in this Scenario**
Data Orchestration Service	Transfers large-scale datasets across DEP components, provides access to climate projections and reanalysis data for model training, and allows data transfer to external resources
DEP Storage	Provides large-scale data storage for training data and output datasets
AI Model Hub	Provides the catalogue of available AI models together with model metadata, versioning, and compatibility information.
Data Discovery & Popularity Service	Allows downloading climate datasets through customizable queries, adapts and integrates the ESGF Search service, enabling users to efficiently locate data from the ESGF
ItwinAI	Provides an AI framework for hyperparameter optimization and ML-related developments
MLFlow/Jupyter	Allows model tracking during algorithm training and provides an interactive environment for data analysis

Preconditions

- Replication of the data on the Data Exploitation Platform.
- Access to HPC clusters to have enough computational resources available.
- Python environment (for the test and development of the AI model), such as itwinai proposed in WP3.
- All system components involved in the Use Case are deployed on the DEP and work as intended.
- All actors involved in the use case are registered in the DEP user management, and their access rights are set accordingly.

Expected Outcome / Validation Reporting

Table 3: SUC1 Scenario Flow and Test Cases

Use Case Testing and Validation Plan	
Scenario Step	**Test Case & Acceptance Criteria**
Step 1: DEP Administrator - provides - computational resources	Data Scientist accesses to the required computational resources for training and testing the ML (Machine Learning) model. Model training will be carried out on the HPC infrastructure. Before training the model, an approximate estimation of required resources (such as memory, GPU hours, and CPU) will be assessed.
Step 2: Data scientist - requests data transfer through - Data Orchestration Service	Data Orchestration Service receives a request for data transfer from data scientists, and the data transfer is initiated.
Step 3: Data Orchestration Service - transfers climate datasets from external data stores to **- DEP Storage**	Data Orchestration Service gives access to the reanalysis data needed as a gridded target dataset during the training phase. The data are now available for training and inference.
Step 4: Data Discovery & Popularity Service - downloads climate datasets through queries from ESGF and other climate data repositories required to **- DEP storage**	Climate simulations archived on the ESGF data platform are replicated by the Data Discovery & Popularity Service on the DEP storage. The data are now available for training and inference.
Step 5: Data Scientist - does data pre-processing, trains and tests AI models needed to perform the downscaling task through **- ItwinAI**	ItwinAI is deployed on the DEP with a dedicated interface for data scientists. Additional plugins will be developed to integrate downscaling capabilities, enabling model training and inference through ItwinAI tools.
Step 6: Domain expert - evaluates and validates model results through - **MLFlow/Jupyter**	MLFlow and Jupyter are deployed and working on the DEP, so that Data Scientists can validate the model results.
Step 7: Data scientist - registers validated model in - **AI Model Hub**	The final model is stored in the model hub, as well as the documentation to use it.
Step 8: Data Orchestration Service - moves final outputs	The results are displayed in the correct format and moved to the DEP Storage and external buckets, i.e. S3 buckets belonging to

to **- DEP Storage**	the Administrator of the use case (Hypermeteo).
Step 9: Data scientist **-** performs inference using the developed AI model available in - **AI Model Hub**	The model can now be used by End Users in order to infer downscaled climate projections over determined regions. New regions need retraining of the algorithm through the usage of the DEP.

SUC2: Smart Detection of Anomalies in Climate Data Usage

Use Case ID: SUC 2

Description/Purpose

In the Climate Science domain, the Earth System Grid Federation (ESGF) provides a federated data infrastructure for the Earth system modelling community for distributing and accessing large-scale climate data. In this context, 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. More specifically, 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.

The SUC2 will provide an ML-based model able to learn patterns associated with the usage of climate data from ESGF and predict and/or detect changes/anomalies. More specifically, the ML model will be trained on historical data usage and transfer logs collected from the ESGF nodes, then it will be applied to current data usage streams, for example, to identify the most used data in a given period or to react to the detection of high loads occurring in the data download streams.

Actors and Roles

- **Domain expert:** An authenticated and authorized ENES RI manager/software component in charge of detecting anomalies in data download streams or identifying trends or changes in download patterns.
- **DEP System:** Provides compute resources and components required to enable the use case.

System Components Involved

Table 4: SUC2 Software Components

List of UC Components and their Role	
Component Name	**Role/Involvement in this Scenario**
DEP computing site	Provides the computing infrastructure for the development, training, and deployment of ML models
AI Model Hub	Provides the catalogue of available ML models, including those used in the use case.
itwinai	Support ML model implementation.
MLflow	Model tracking during algorithm training.
AI Computing Framework	Deliver the scalable computing components of the DEP System.
ENES Data Popularity Service (WP2|T2.3)	Provides data usage and transfer logs related to the ESGF nodes, making them available through a set of APIs.

Preconditions

- The Domain Expert is authenticated and authorized.
- Access to OpenStack Cloud/HPC resources to get computational resources for running the ML model.
- Python ecosystem (e.g., containerized solution support) for running the ML model.
- An ML model trained exploiting WP3 components (e.g., **itwinai**) and made available in the **AI Model Hub**.
- Data usage information available from the ENES Popularity service.

Expected Outcome / Validation Reporting

Table 5: SUC2 Scenario Flow and Test Cases

Use Case Testing and Validation Plan	
Scenario Step	**Test Case & Acceptance Criteria**
Step 1: DEP computing site - provides - **computational resources** for running the ML model.	The Domain Expert gets access to the required computational resources for running the ML model.
Step 2: Domain Expert - gets access to a scalable data science environment (e.g., Jupyter-based) made available by the - **AI Computing Framework**	The AI Computing Framework provides user-friendly interfaces such as Jupyter Notebook so that the Domain Expert can easily access a fully configured environment.
Step 3: Domain Expert - discovers and selects a pre-trained AI model from the - **AI Model Hub**	A pre-trained ML model for anomaly/pattern detection/prediction is chosen and downloaded from the AI Model Hub to the Domain Expert workspace/environment.
Step 4: Domain Expert - retrieves data usage information provided by the - **ENES Data Popularity Service**	Data usage information is available for performing the inference step.
Step 5: Domain Expert - performs inference using the model retrieved from the - **AI Model Hub**	The system returns inference results.
Step 6: DEP System - sends the results from the model back to the - **Domain Expert**	The result is displayed in the correct format and matches the expected output for the input sample.
Step 7: Domain Expert - performs a downstream action on the - **DEP System**	The result is fed back into the DEP components to improve operational settings (e.g., data retrieval from ESGF).

SUC3: Intelligent Scheduling of Radar Observations and Experiment

Use Case ID: SUC 3

Description/Purpose

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.

The purpose of SUC3 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 is expected to yield useful data.

Actors and Roles

- **Domain Expert (Radar Operator)**: People whose task is to schedule the radar experiments. The primary purpose of Machine Learning models in the use case is to support their work.
- **System:** The software system itself (or an internal service/component), representing automated actions or backend processes that occur as part of the use case.
- **External Service:** Third-party services providing some of the necessary space weather and weather parameters through APIs and other channels.

System Components Involved

Table 6: SUC3 Software Components

List of UC Components and their Role	
Component Name	**Role/Involvement in this Scenario**
AI Model Hub	Serves the AI models used in the scenario.
AI Computing Framework	Provides the computing resources and supports the development, training, and deployment of ML models.
ML Models	Models responsible for predicting the presence/absence of environmental properties needed for various radar experiments
Expert-informed Logic	Component to connect the output of individual models into a suggestion about the viability of a radar experiment for a specific date
Data Orchestration Component	Responsible for the information flow between different components

Preconditions

- Availability of the real-time information necessary about space weather (and weather) conditions through **External Services**.
- Availability of a set of experiment requests from stakeholders (researchers) that specify what kind of operations they want, and the required environmental conditions.
- To be defined: the exact parts of the human operators' workflow that are to be supported by the AI, including the measures for validating the outcomes from these parts, and the range where these measures are acceptable.

Expected Outcome / Validation Reporting

Table 7: SUC3 Scenario Flow and Test Cases

Use Case Testing and Validation Plan	
Scenario Step	**Test Case & Acceptance Criteria**
Step 1: AI computing framework - provides - **computational resources**	The end user gets access to the required computational resources for running the ML model.
Step 2: The **Domain Expert (Radar**	The end user gets access to the requests, and the logic is

Operator) inputs the experiment request to the **Expert-informed logic**	able to process the input
Step 3: The **Expert informed logic** engages **ML models**	An expert validates the logic for engaging the necessary models
Step 4: The data orchestration component obtains data from the **External services**	The data is obtained from external services, and if the service is down, the necessary error flag is raised
Step 5: The data orchestration component feeds the data **to the ML models**	The ML model gains access to the data and is able to provide a prediction
Step 6: The **ML models** provide the predictions to the **Expert-informed logic**	The ML models return inference results (with a measurement error in the acceptable range, which is to be defined)
Step 7: The **Expert-informed logic** returns a suggestion to the **Domain Expert (Radar Operator)**	The expert-informed logic provides suggestions regarding the experiments to be scheduled (which potentially can mean no experiment to be scheduled) to the end user, in accordance with the expectations to be defined

SUC4: Space Debris and Anomaly Detection

Use Case ID: SUC 4

Description/Purpose

EISCAT AB (formerly EISCAT Scientific Association) 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 on 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.

With SUC4, we want to investigate how AI methods can be used to find specific events as well as anomalies, i.e. rare events or disturbances, in this data set. 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.

We hope to find phenomena in the data which have been missed in previous analyses, or which have possibly been wrongly categorised. Furthermore, we would like to see statistical information about 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 up and in real-time the phenomena and events 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). We will use the existing EISCAT data to identify events relating to the presence of space objects, establish occurrence statistics and find connections between these events, 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.

Actors and Roles

- **Domain Expert (Radar Analyst)**: Researcher, analyst, or scientist working with EISCAT radar data. The primary purpose of Machine Learning models in the use case is to support their work or that monitor the results on an automated process chain.
- **System:** The AI-driven detection and classification software. The software system itself (or an internal service/component), representing automated actions or backend processes that occur as part of the use case.
- **External Service:** Third-party services providing some of the necessary data from EISCAT Radar Infrastructure, and external space object databases/APIs parameters through APIs and other channels.

System Components Involved

Table 8: SUC4 Software Components

List of UC Components and their Role	
Component Name	**Role/Involvement in this Scenario**
Radar Data Acquisition	Ingests radar observations (historical and/or real-time) from the EISCAT infrastructure.
Data Storage	Central module for storing radar observations and metadata.

Feature Extraction Service	Derives model-ready tensors/features, i.e., cleans, formats, and prepares the data for AI model input.
Model Inference Service	Loads deployed models, performs classification/anomaly inference on incoming data, and produces event labels and scores.
Prediction Store	Persists inference results (labels, confidence, embeddings).
Historical Data Loader	Reads radar observations from the EISCAT infrastructure.
Labelling & Curation Tooling	Human-in-the-loop review, relabelling, and correction of ambiguous/novel events.
Training Compute Hub	Scalable compute (GPU/CPU) to train models.
Model Registry / AI Model Hub	Stores versioned models, signatures, and deployment artifacts.

Preconditions

- The researcher/analyst is authenticated in the system and has the necessary permissions to access radar data and analysis tools.
- Sufficient computational resources (CPU/GPU, memory, storage) are available for running model inference or training tasks.
- Availability of the radar data necessary for training and inference for the detection of anomalies and space debris.

Expected Outcome / Validation Reporting

Table 9: SUC4 Scenario Flow and Test Cases

Use Case Testing and Validation Plan	
Scenario Step	**Test Case & Acceptance Criteria**
Production Pipeline	
Step 1: Radar Data Acquisition – ingests radar observations from – **EISCAT infrastructure**	Radar stream is successfully ingested.

Step 2: Radar Data Acquisition – sends raw observations to – **Data Storage**	Raw observations are stored correctly; schema and metadata match specifications.
Step 3: Feature Extraction Service – retrieves raw observations from – **Data Storage**	Service retrieves requested data within the defined timeout.
Step 4: Feature Extraction Service – derives features and stores them in – **Data Storage**	Features are computed and stored correctly; feature schema matches training/inference requirements.
Step 5: Model Inference Service – loads trained model from– **Model Registry / AI Model Hub**	The inference service loads the correct model version.
Step 6: Model Inference Service – consumes features from – **Data Storage**	Features are read and parsed successfully.
Step 7: Model Inference Service – produces event labels and anomaly scores to – **Prediction Store**	Inference completes within a reasonable time per sample; classification accuracy achieves good performance.
Step 8: Prediction Store – persists inference results for – **Domain Expert**	Schema matches documentation.
Training Pipeline	
Step 1: Historical Data Loader – retrieves archived radar observations from – **Data Storage**	Historical datasets load correctly.
Step 2: Labelling & Curation Tooling – supports human review and relabelling by – **Domain Expert**	The user can view, relabel, and save events.
Step 4: Training Compute Hub – trains new models and sends them to – **Model Registry / AI Model Hub**	Training completes on the test dataset within allocated resources; the resulting model artifact is registered successfully.
Step 5: Model Registry / AI Model Hub – stores versioned models for – **Model Inference Service**	Model is versioned correctly and accessible.

SUC5: Colorectal Cancer Prediction with Explainable AI

Use Case ID: SUC 5

Description/Purpose

Histopathological whole-slide images (WSIs) are ultra-high-resolution scans of entire microscope slides, typically captured at 20× 40× magnification. A single WSI can reach about 100,000 × 100,000 pixels, billions of pixels per file, creating gigapixel images containing organ-level architecture, all the way down to individual cells.

In Graz, a large academic biobank has been digitising parts of its decades-old archive of tissue slides and assembling the resulting WSIs into comprehensive datasets, including a colorectal cancer dataset comprising nearly 10,000 cases. In this use case, we will utilise the colorectal cancer dataset, which is available through the BBMRI-ERIC Directory, alongside other datasets from more than 400 partner biobanks across Europe.

The availability of such large-scale WSI datasets allows deep-learning algorithms to probe tissue landscapes at cellular resolution and uncover micro-patterns beyond routine microscopy.
In Scientific Use Case 5, we will develop AI models to predict patient survival based on the histology of lymph nodes. 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.

Deep neural network models will be trained to regress an individual's survival from WSIs directly. 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.

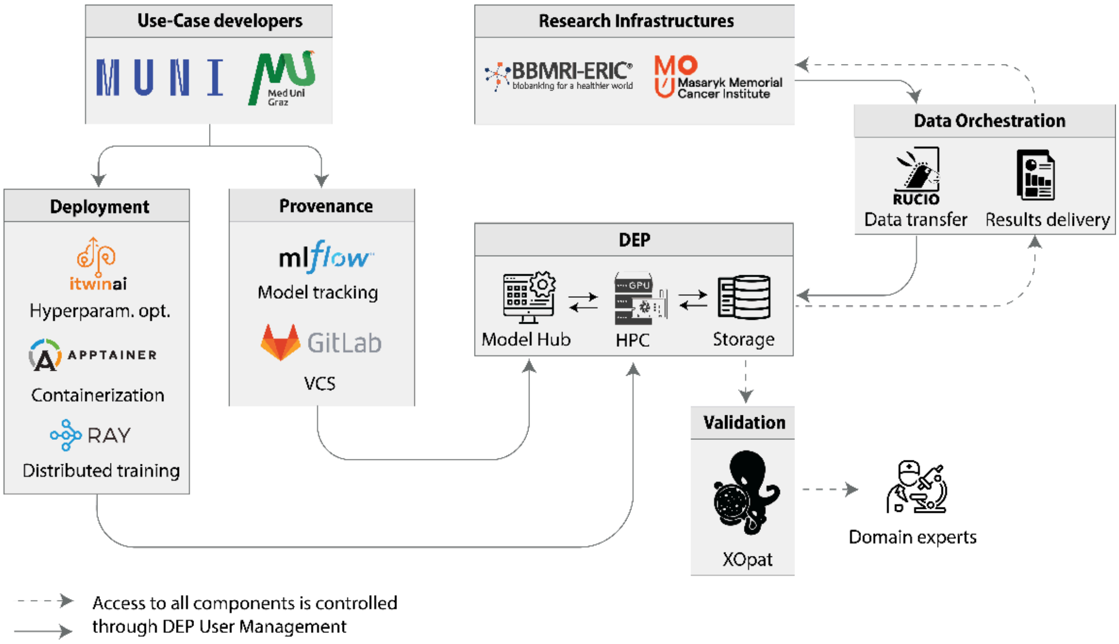

Figure 1: High-level Overview of the Interaction between System Components and Actors

Actors and Roles

- **Data Scientists:** Primary actors initiating the use case, performing training of the AI model, and evaluating model outputs on a technical level.
- **Domain Experts:** Pathologists with expert knowledge evaluate model outputs on a biological level.
- **RIs**: Provide the data for the scientific experiment.
- **DEP system**: Provides HPC compute resources and components required to run the model.

System Components Involved

Table 10: SUC5 Software Components

Use Case List of Components and their Role	
Component Name	**Role/Involvement in this Scenario**
DEP Storage	Large-scale data storage for training data
DEP Data orchestration	Transfer of large-scale datasets on the DEP
DEP User Management	Access control to the respective system components
DEP HPC provider	GPU computing infrastructure

AI Model Hub	Serves the AI models used in the scenario
ItwinAI	Hyperparameter optimization
Apptainer	Containerized deployment
Ray	Distributed training
MlFlow	Model tracking during algorithm training
GitLab	Version Control System
XOpat	Web-based viewer for WSIs

Preconditions

- The WSI data for the use case is available at the RI, and the data transfer interfaces between the RI and the DEP are set up.
- All system components involved in the Use Case are deployed on the DEP and work as intended.
- All actors involved in the use case are registered in the DEP user management, and their access rights are set accordingly.
- A sufficient amount of computing resources is available at the HPC provider.

Expected Outcome / Validation Reporting

Table 11: SUC5 Scenario Flow and Test Cases

Use Case Testing and Validation Plan	
Scenario Step	**Test Case & Acceptance Criteria**
Step 1: Data Scientist - requests data transfer through - **Data Orchestration component**	Rucio receives a request for the data transfer, and the data transfer is initiated.
Step 2: Rucio - transfers data from RI to - **DEP data storage**	Data are replicated over the DEP and are available for training and inference.
Step 3: Data orchestration component - sends notification	A notification to the Data scientist was sent.

about completed transfer to - **Data scientist**	
Step 4: Data scientist - initiates model training using - **ItwinAI / Apptainer / Ray**	ItwinAI / Apptainer / Ray are deployed on the DEP, they work as intended, and the respective interfaces for the data scientist to use them are provided.
Step 5: AI computing framework - provides - **HPC computational resources**	Model training can be initiated, and the HPC providers can deliver resources for large-scale training (32 to 64 GPUs simultaneously).
Step 6: Data scientist - evaluates model training results through - **MLFlow**	MLFlow is deployed and working on the DEP.
Step 7: Domain experts - validate algorithm outputs with - **XOpat**	XOpat is deployed on the DEP. The web-based interface can be reached, and access is controlled through the DEP AAI.
Step 8: Data scientist - registers validated model in - **Model Hub**	The model can be pulled from the model hub, and documentation about how to use the registered model exists.
Step 9: Results delivery mechanism - sends algorithm outputs back to - **RI**	The RI gets notified about the incoming transfer, and the results are sent correctly.

SUC6: Synthetic Data for Computational Pathology

Use Case ID: SUC 6

Description/Purpose

Histopathological whole slide images (WSIs) are ultra-high resolution scans of entire microscope slides, typically captured at 20× 40× magnification. A single WSI can reach about 100,000 × 100,000 pixels, billions of pixels per file, creating gigapixel images containing organ-level architecture, all the way down to individual cells.

In Graz, a large academic biobank has been digitising parts of its decades-old archive of tissue slides and assembling the resulting WSIs into comprehensive datasets, including a colorectal cancer dataset comprising nearly 10,000 cases. In this use case, we will utilise the colorectal cancer dataset, which is available through the BBMRI-ERIC Directory, alongside other datasets from more than 400 partner biobanks across Europe.

In Scientific Use Case 6, we will develop algorithms for generating synthetic histopathological images.

Synthesised histopathological images are valuable for computer scientists developing algorithms, as they can augment small or imbalanced datasets, improving model performance and generalisation. Moreover, since synthetic data is not tied to real patients, it enables researchers to work with medically relevant images without navigating complex legal and ethical approval processes.

Actors and Roles

- **Data Scientists:** Primary actors initiating the use case, performing training of the AI model, and evaluating model outputs on a technical level.
- **Domain Experts:** Pathologists with expert knowledge evaluate model outputs on a biological level.
- **RIs**: Provide the data for the scientific experiment.
- **DEP System**: Provides HPC compute resources and components required to run the model.

System Components Involved

Table 12: SUC6 Software Components

Use Case List of Components and their Role	
Component Name	**Role/Involvement in this Scenario**
DEP Storage	Large-scale data storage for training data
DEP Data orchestration	Transfer of large-scale datasets on the DEP
DEP AAI	Access control to the respective system components
DEP HPC provider	GPU computing infrastructure
AI Model Hub	Serves the AI models used in the scenario
ItwinAI	Hyperparameter optimization
Apptainer	Containerized deployment
Ray	Distributed training
MlFlow	Model tracking during algorithm training
GitLab	Version Control System
XOpat	Web-based viewer for WSIs

Preconditions

- The WSI data for the use case is available at the RI, and the data transfer interfaces between the RI and the DEP are set up.
- All system components involved in the Use Case are deployed on the DEP and work as intended.
- All actors involved in the use case are registered in the DEP AAI, and their access rights are set accordingly.
- A sufficient amount of computing resources is available at the HPC provider.

Expected Outcome / Validation Reporting

Table 13: SUC6 Scenario Flow and Test Cases

Use Case Testing and Validation Plan	
Scenario Step	**Test Case & Acceptance Criteria**
Step 1: Data scientist - requests data transfer through - **Data Orchestration component**	Rucio receives a request for the data transfer, and the data transfer is initiated.
Step 2: Rucio - transfers data from RI to - **DEP data storage**	Data are replicated over the DEP and are available for training and inference.
Step 3: Data orchestration component - sends notification about completed transfer to - **Data scientist**	A notification to the Data scientist was sent.
Step 4: Data scientist - initiates model training using - **ItwinAI / Apptainer / Ray**	ItwinAI / Apptainer / Ray are deployed on the DEP, they work as intended, and the respective interfaces for the data scientist to use them are provided.
Step 5: AI computing framework - provides - **HPC computational resources**	Model training can be initiated, and the HPC providers can deliver resources for large-scale training (32 to 64 GPUs simultaneously).
Step 6: Data scientist - evaluates model training results through - **MLFlow**	MLFlow is deployed and working on the DEP.
Step 7: Domain experts - validate algorithm outputs with - **XOpat**	XOpat is deployed on the DEP. The web-based interface can be reached, and access is controlled through the DEP AAI.
Step 8: Data scientist - registers validated model in - **Model Hub**	The model can be pulled from the model hub, and documentation about how to use the registered model exists.
Step 9: Results delivery	The RI gets notified about the incoming transfer, and the

mechanism - sends algorithm outputs back to - **RI**	results are sent correctly.

SUC7: Foundational Models for Heterogeneous Biological Image Data

Use Case ID: SUC 7

Description/Purpose

This use case focuses on developing and applying foundation models to interpret and categorise large, diverse collections of biological imaging data. As life sciences imaging datasets grow in size and complexity, understanding and searching the large collection of bioimage data stored at the BioImage Archive (BIA) can be difficult for end users. By training a large-scale model that learns meaningful representations (embeddings) of heterogeneous image inputs, the goal is to enable powerful downstream functionalities such as automated categorisation, similarity-based search, and extraction of derived measurements. The resulting model will enhance RI operations by improving data organisation and accessibility, ultimately increasing the value and usability of archived imaging datasets.

Actors and Roles

- **Data scientist:** Primary actor initiating the use case, performing training of the AI model, and evaluating model outputs on a technical level. This can be a BIA team member or a model developer.
- **RIs:** the BIA serves as the source of imaging datasets to be discovered and analysed.
- **DEP system:** provides HPC compute resources and components required to run the model.

System Components Involved

Table 14: SUC7 Software Components

Use Case List of Components and their Role	
Component Name	**Role/Involvement in this Scenario**
AI Model Hub	Serves the AI models used in the scenario.
DEP replicated data repository	Stores and serves data from the RIs inside the DEP
BIA data catalogue	Indexes, describes, and provides search and access to BIA data

DEP System	Software environment that allows users to interact with its capabilities.
BioEngine	Executes selected AI models on user-specified datasets using scalable compute resources; handles model inference, job scheduling, and resource management.
BIA data	Data transferred from the BIA to the DEP replicated data repository
Data Orchestration Service	System in charge of data life cycle management within the DEP
BIA S3 bucket	BIA cloud storage container

Preconditions

- The imaging data for the use case is available at the BIA, and the data transfer interfaces between the BIA and the DEP are set up.
- All system components involved in the Use Case are deployed on the DEP and work as intended.
- All actors involved in the use case are registered in the DEP AAI, and their access rights are set accordingly.
- A sufficient amount of computing resources is available at the HPC provider.

Expected Outcome / Validation Reporting

Table 15: SUC7 Scenario Flow and Test Cases

Use Case Testing and Validation Plan	
Scenario Step	**Test Case & Acceptance Criteria**
Step 1: Data scientist - discovers relevant BIA datasets from the - **BIA data catalogue**	Users can discover at least one relevant dataset in the catalogue.
Step 2: Data scientist - transfers BIA data to - **DEP replicated data repository**	Data are replicated over the DEP and are available for training and inference.

Step 3: DEP System - sends transfer information and confirmation to **- Data scientist**	Information about the transfer is sent to the end user, allowing users to visualise errors and when the data has been successfully transferred.
Step 4: Data scientist - uploads AI model architecture to - **AI Model Hub**	The model is successfully uploaded to the DEP AI Model Hub
Step 5: DEP System - sends upload confirmation message to - **Data scientist**	Information about the upload is sent to the end user, allowing users to visualise errors and when the model has been successfully uploaded.
Step 6: Data scientist - defines training pipeline for a model and a dataset using **- DEP System**	The end user should be able to define different steps in the pipeline, mainly datasets to use (and how to split them), any pre-processing necessary and the model that will be used for training. The system should also accept relevant parameters (e.g., batch size, learning rate, epochs). The system provides clear, actionable error messages for missing or invalid inputs.
Step 7: BioEngine - performs training using**- BIA data**	For training and fine-tuning models, GPU utilisation should exceed 90%. When fine-tuning segmentation models, accuracy must exceed 80% on a representative validation test for relevant metrics such as IoU or Dice[5].
Step 8: DEP System - presents performance metadata information to **- Data scientist**	The UI should display the correct metrics for the input model. Ideally, users can visualise performance metadata and errors live during training runs.
Step 9: Data Orchestration Service - exports the model artefacts, intermediate data, and performance metadata, to **- BIA S3 buckets**	The data orchestration service should periodically save the state of the model training (and related data). At the end of the transfer, all outputs must be present in the correct bucket and path
Step 10: DEP System - presents transfer information to **- Data**	Information about transfer is sent to end user, allowing users to visualise errors and when the data has been

[5] See Rahman & Wang (2016) and Milletari et al. (2016) for formal definitions of IoU and Dice metrics used in segmentation model evaluation.

scientist	successfully transferred.

SUC8: Generative AI-Powered Assistant for Data Discovery & Analysis

Use Case ID: SUC 8

Description/Purpose

This use case aims to develop and validate a generative AI-powered assistant to support data discovery and analysis within the BioImage Archive, addressing the increasing complexity of large-scale bioimaging datasets and AI tools. The assistant will help users—particularly non-experts—navigate the archive, retrieve relevant datasets, and perform common image analysis tasks (e.g., segmentation, cell counting) using integrated AI models such as Cellpose[6] and Segment Anything[7].

The assistant is built by integrating two key components: the **BioImage.IO Chatbot**, which provides a natural language interface for model and data discovery; and **BioEngine**, a container-based execution backend for scalable AI model inference and training. Through this integration, the assistant will transform user prompts into real-time computational actions within the DEP, offering a seamless workflow from data selection to result generation.

This use case demonstrates how the DEP enables large-scale, AI-powered analysis of public RI datasets by combining model repositories, execution infrastructure, and user-facing interfaces. It supports the overall goals of RI-SCALE by improving data accessibility, reducing technical barriers, and promoting the reuse of RI data through user-friendly, AI-driven services.

Actors and Roles

- **Data scientist**: A scientific researcher or imaging data analyst interacting with the assistant to discover datasets and perform bioimage analysis (e.g., segmentation, cell counting). May range from expert bioinformaticians to non-technical life scientists
- **DEP system**: The integrated DEP components that power the assistant—specifically including the BioImage.IO Chatbot (LLM-based user interface), BioEngine (model execution backend), and the DEP orchestration layer managing dataset access, compute scheduling, and result delivery.

[6] Stringer, C., Wang, T., Michaelos, M., & Pachitariu, M. (2021). *Cellpose: A generalist algorithm for cellular segmentation.* Nature Methods, 18(1), 100–106.
[7] Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., et al. (2023). *Segment Anything.* arXiv:2304.02643.

System Components Involved

Table 16: SUC8 Software Components

Use Case List of Components and their Role	
Component Name	**Role/Involvement in this Scenario**
AI Model Hub	Serves as the catalogue of available AI models (e.g. Cellpose, Segment Anything) and provides model metadata, versioning, and compatibility information used by the assistant.
BioImage.IO Chatbot	Acts as the generative AI interface enabling natural language interaction; interprets user queries, suggests relevant models/datasets, and orchestrates analysis workflows.
BioEngine	Executes selected AI models on user-specified datasets using scalable compute resources; handles model inference, job scheduling, and resource management.
DEP Orchestration Layer	Coordinates data access, model execution, and delivery of results across system components; ensures secure, reproducible operation within the DEP environment.
BioImage Archive	Provides the imaging datasets to be explored and analysed; serves as the primary data source integrated into the DEP for this use case.

Preconditions

- **The data scientist is authenticated** through the DEP's identity management system and has access permissions to the BioImage Archive datasets and model execution services.
- **The BioImage.IO Chatbot is deployed and integrated** with the DEP environment, with access to the AI Model Hub and BioEngine APIs.
- **BioEngine is running and properly configured** to execute the relevant AI models (e.g., Cellpose, Segment Anything) with GPU resources allocated via the DEP.
- **The BioImage Archive is indexed and accessible** through the DEP's orchestration layer, allowing programmatic search and retrieval of datasets.
- **The AI models intended for use in the scenario** (e.g., Cellpose, Segment Anything) are pre-registered in the AI Model Hub and executable through BioEngine.

Expected Outcome / Validation Reporting

Table 17: SUC8 Scenario Flow and Test Cases

Use Case Testing and Validation Plan	
Scenario Step	**Test Case & Acceptance Criteria**
Step 1. Data scientist – asks a question or analysis request via – **BioImage.IO Chatbot**	Chatbot responds to user input within 3 seconds and interprets the request correctly.
Step 2. BioImage.IO Chatbot – searches relevant datasets in – **BioImage Archive**	At least one relevant dataset is returned from BioImage Archive.
Step 3. Data scientist – selects dataset and confirms task via – **BioImage.IO Chatbot**	The dataset is correctly selected and passed to the backend.
Step 4. BioImage.IO Chatbot – selects appropriate model from – **AI Model Hub**	A valid AI model is selected based on the task (e.g. segmentation).
Step 5. BioImage.IO Chatbot – sends execution request to – **BioEngine**	A job is successfully submitted to BioEngine.
Step 6. BioEngine – runs model on selected dataset from – **BioImage Archive**	Model inference runs without error and completes within an acceptable time (e.g. <1 min for test data).
Step 7. BioEngine – returns result to – **BioImage.IO Chatbot**	Output is returned in the correct format (e.g. segmentation mask, cell count).
Step 8. BioImage.IO Chatbot – presents results to – **Data scientist**	The final result is displayed clearly in the chatbot interface.

3.2. Technical Use Case Scenarios (TUC-1 to TUC-4)

TUC1: Scalability on EuroHPC with Destination Earth

Use Case ID: TUC 1

Description/Purpose

The purpose of this use case is to evaluate and measure how the DEP performs when deployed on EuroHPC infrastructure, serving a computationally demanding AI model using large-scale datasets from Destination Earth (DestinE). The goal is to validate the scalability and performance of the DEP and its components when used for large data processing and model execution in an HPC environment.

Actors and Roles

- **Data Scientist:** Initiates and monitors the workflow, uploads datasets and models to the DEP, and analyses the performance results.
- **DEP Administrator:** Manages and supervises the DEP environment on the EuroHPC infrastructure and ensures access to the relevant DestinE data and models.
- **EuroHPC Infrastructure:** Provides the HPC resources for large-scale model execution and benchmarking.

System Components Involved

Table 18: TUC1 Software Components

List of UC Components and their Role	
Component Name	**Role/Involvement in this Scenario**
AI Model Hub/anemoi	Retrieves or registers AI models. In this UC, the model is expected to be pulled from Anemoi through the DEP.
DEP Replicated Data Repository	Stores and manages the datasets and intermediate outputs used during model execution.
AI Computing Framework (EuroHPC backend)	Provides the compute environment to run the AI model at scale.

Preconditions

N/A

Expected Outcome / Validation Reporting

Table 19: TUC1 Scenario Flow and Test Cases

Use Case Testing and Validation Plan	
Scenario Step	**Test Case & Acceptance Criteria**
Step 1: DEP Replicated Data Repository - retrieve training data from- **Anemoi input**	DestinE datasets will be made available to the DEP via the 'DestinE databridge', a cloud stack collocated with the EuroHPC systems.
Step 2: DEP Replicated Data Repository - retrieve model from - **Anemoi input**	DestinE AI model will be made available from Anemoi.
Step 3: DEP/EuroHPC - model will run at scale with different parallel frameworks - **Data Scientist**	The performance optimisation and scalability of the AIFS model will be tested in this setup.
Step 4: DEP/EuroHPC - model will run at scale with different parallel frameworks - **Data Scientist**	Use the AIFS machine learning model of DestinE, and digital twin data from DestinE, in a full DEP stack to be deployed on different (pre)exascale EuroHPC systems.

TUC2: Advanced Image Compression

Use Case ID: TUC 2

Description/Purpose

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. The purpose of the Technological Use Case 2 is to integrate JPEG 2000 image compression into the DEPs Data Preparation Pipeline. JPEG2000 compression could potentially increase data loading times during training and inference of algorithms.

Actors and Roles

- **Data scientists:** Initiate the JPEG2000 WSI compression on the DEP.

- **DEP system**: Provides computing resources to run the compression and storage for the compressed images.

System Components Involved

Table 20: TUC2 Software Components

List of UC Components and their Role	
Component Name	**Role/Involvement in this Scenario**
Data Preparation for Exploitation	Orchestrates the JPEG2000 compression
DEP Data storage	Contains the uncompressed images and stores the result of the compression
HPC compute infrastructure	Performs the computations necessary for the compression

Preconditions

- The WSI dataset to be compressed has been transferred to the DEP.

Expected Outcome / Validation Reporting

Table 21: TUC2 Scenario Flow and Test Cases

Use Case Testing and Validation Plan	
Scenario Step	**Test Case & Acceptance Criteria**
Step 1: Data Scientist - initiates JPEG2000 compression through - **Data Preparation for Exploitation**	The Data Preparation for Exploitation exposes the necessary interfaces to initiate the compression.
Step 2: Data Preparation for Exploitation - performs compression using - **HPC infrastructure**	The HPC infrastructure provides the necessary compute resources and performs the compression successfully.
Step 3: Data Preparation for	A notification that the compression has been completed is sent to

Exploitation - sends notification to - **Data Scientist**	the data scientist.

TUC3: Green Computing Improvement

Use Case ID: TUC 3

Description/Purpose

This use case investigates how the integration of GROQ Language Processing Units (LPUs) can reduce energy consumption while maintaining comparable performance in Data Exploitation Platforms (DEPs). GROQ hardware offers substantial energy efficiency advantages over traditional GPUs, particularly for inference tasks, due to its large internal memory (230MB) and optimized compute capacity. While GROQ LPUs do not necessarily deliver higher performance in model training, they excel in inference workloads by achieving similar performance levels with significantly lower energy usage.

The objective is to evaluate whether GROQ cards can reduce energy consumption and model execution time as an essential building block of DEPs, serving the AI model inference, i.e. operations stage. The evaluation is twofold: first, a targeted test of GROQ inference energy efficiency in a small 4-node cluster; second, large-scale testing through the GROQ API at a dedicated data centre. The tests focus on inference workloads, emphasizing energy savings and maintaining performance comparable to GPU-based systems.

Actors and Roles

- **Austrian Scientific Computing Server Administrator:** The individual responsible for managing and configuring the Austrian Scientific Computing (ASC) server infrastructure and the GROQ hardware for this study. This includes overseeing ASC server setup, resource allocation, and operational support during the evaluation.
- **Data Scientist:** A generic ASC user, such as a researcher or AI model developer, who initiates and conducts the evaluation scenario to examine energy efficiency and performance of GROQ hardware for inference workloads.
- **System:** The Austrian ASC server, which manages workload distribution, resource scheduling, and execution tracking for the GROQ hardware during the evaluation.

System Components Involved

Table 22: TUC3 Software Components

List of UC Components and their Role	
Component Name	**Role/Involvement in this Scenario**
GROQ Hardware	Specialized inference accelerators installed in the server, used for high-performance and energy-efficient execution.
GPU Hardware	Graphics processing units in the server, used as the baseline hardware for inference comparison.
ASC Server	The central server hosting the GROQ and GPU hardware, managing execution and communication with the researcher.
Server Infrastructure	The supporting resources of the ASC server including CPU, memory, storage, and networking.
Monitoring and Benchmarking Framework	Software tools used to measure, log, and compare inference performance and energy consumption for both GPU and GROQ hardware.
AI Models	Machine learning models uploaded by researchers to the ASC server for inference execution.
Input Data	Datasets uploaded by researchers to be processed by the AI models during inference tests.
Performance Metrics	The recorded results of inference tests, including throughput, latency, and efficiency values stored by the monitoring framework.
Summary Results	The aggregated benchmarking and performance comparison output sent from the ASC server to the researcher.

Preconditions

N/A

Expected Outcome / Validation Reporting

Table 23: TUC3 Scenario Flow and Test Cases

Use Case Testing and Validation Plan	
Scenario Step	**Test Case & Acceptance Criteria**
Step 1: Administrator - installs and connects **- GROQ cards to Austrian DataLab server**	GROQ hardware is physically connected and recognized by the server. Validation is confirmed if ASC server commands (e.g., lshw, groqview) detect the device correctly and test inference jobs can be scheduled.
Step 2: Administrator - configures **- GROQ hardware on the server**	Configuration settings for GROQ cards are applied successfully. Verified by ASC server logs showing correct hardware initialization and resource availability.
Step 3: Researcher - selects **- AI model for inference execution**	The AI model is locally available or retrieved by the researcher and loaded within 3 seconds. Model metadata, including version and integrity checksums, match expected values.
Step 4: Researcher - uploads AI model and deploys to the **-server**	Model and dataset upload completes successfully without corruption. Validated through file integrity checksums and accessible model metadata.
Step 5: Researcher - initiates inference execution on **- GPU hardware**	Inference tasks on the GPU are complete within 10 seconds per batch with no errors. Output results from the GPU and LPU are compared to verify they are identical or comparable, regardless of network training status.
Step 6: Monitoring and Benchmarking Framework - records **- GPU inference performance and energy consumption**	Performance metrics such as latency and power draw per inference are logged consistently for at least 100 sequential runs and/or across different batch and interval input scenarios, without missing data.
Step 7: Researcher - initiates inference execution on - **GROQ hardware**	Inference tasks on GPU, GROQ hardware, and (optionally) CPU complete within defined batch time limits with no errors. Outputs are compared for consistency, and performance metrics (latency, accuracy, power draw) are measured across

	at least 100 sequential runs, with all data logged. Performance from each hardware type is reported in the summary.
Step 8: Monitoring and Benchmarking Framework - records **- GROQ inference performance and energy consumption**	Latency and power consumption metrics are logged fully, covering all test samples consistently without gaps.
Step 9: Monitoring and Benchmarking Framework - compares **- GPU and GROQ performance metrics**	The framework generates a comprehensive comparison report on latency, throughput, and energy consumption with less than 5% missing data. Report completeness and accuracy are verified.
Step 10: ASC Server - sends - summary results to - **Data Scientist**	The researcher receives a performance summary report in PDF and JSON formats within 30 seconds of test completion. The report contains clearly labelled metrics, charts, and visual plots.

TUC4: Credit Management System

Use Case ID: TUC 4

Description/Purpose

The Credit Management System (CRMS) aims to provide a scalable, modular framework for managing resource usage and credit allocation within DEPs. It integrates multiple logical components through a RESTful API to achieve its core objectives:

- Collect comprehensive data on computational resource usage (e.g., processing power, storage, network transfers) and environmental impacts (e.g., energy use).
- Translate resource consumption data into credit values based on DEP-specific, policy-driven rules that account for factors like resource type, efficiency, and environmental impact.
- Distribute quotas as credits to projects in alignment with the DEP's capacity and governance policies, enabling fair and efficient resource access tailored to platform constraints and priorities.
- Empower administrators to detect resource trends, forecast DEP capacity, and track credit usage, promoting informed resource planning and ensuring operational clarity.

- Facilitate seamless communication with external components, such as the Authentication and Authorisation Infrastructure (AAI) and DEP orchestration layers, through a standardised RESTful API to ensure interoperability and streamlined operations.

This architecture aims to enable traceability, scalability, and compliance with sustainability and operational goals, laying the groundwork for an effective Minimum Viable Product (MVP) in the DEP ecosystem. The use case validation presented for the Credit Management System (CRMS) reflects the general design approach, intended functionality, and conceptual workflow of the system. Specific components, processes, and implementation details may be refined, updated, or modified during the development and integration phases to align with evolving technical requirements, architectural decisions, and project priorities.

Actors and Roles

- **DEP Administrator**: Responsible for internal operations in the DEP System.
- **Data Scientist**: Any end-user operation performed in the context of the DEP.
- **Domain Expert**: A UC-specific person with domain knowledge (e.g. "Domain Expert: doctor who assesses the validity of results").

Use Case Scenario #1

Data Scientist

The Data Scientist begins by discovering relevant datasets from a research environment or data space, then flags selected datasets for analysis within the processing environment. Next, the Data Scientist identifies and selects a pre-configured, pre-trained AI model suitable for the analysis and uses it to process the RI data. Finally, the Data Scientist retrieves the exported results of the data analysis.

Domain Expert (new AI Model)

The Domain Expert creates a new AI model, then deploys and trains it using RI data. After training, the Domain Expert validates the accuracy of the model and, once validated, shares it across one or multiple DEP instances.

Domain Expert (existing AI Model)

The Domain Expert selects an existing model for retraining and associates it with the relevant training data to train the existing or third-party AI model. After training, the Domain Expert validates the accuracy of the model and, once validated, shares it across one or multiple DEP instances.

System Components Involved

Table 24: TUC4 Software Components

List of UC Components and their Role	
Component Name	**Role/Involvement in this Scenario**
Resource Usage & Environmental Impact Tracking	This logical component collects metrics on CPU/GPU/storage/network usage and environmental indicators.
Service Infrastructure	The service infrastructure that supports resources, including CPU, memory, storage, and networking.
DEP AAI	DEP's Authorisation Framework.

Preconditions

- The interconnection between system components and external services is established through the CRMS's RESTful API, which provides a standardised interface for accessing, configuring, and exchanging data related to credit allocation, usage tracking, policy enforcement, and user management across the DEP environment.
- The DEP environment is operational and connected to the CRMS.
- A DEP Administrator has configured credit distribution and translation policies.
- AI models and datasets are available in the research environment. (Data Scientist perspective only)
- In the following scenario, the Data Scientist and the Domain Expert engage with the same underlying descriptive logic.

Expected Outcome / Validation Reporting

Table 25: TUC4 Scenario Flow and Test Cases

Use Case Testing and Validation Plan	
Scenario Step	**Test Case & Acceptance Criteria**
Step 1: Data Scientist – gets authorised – **DEP AAI**	DEP AAI successfully authorises the Data Scientist and grants access to DEP resources.
Step 2: Data Scientist – submits	After gaining access, the Data Scientist prepares and submits

a job – **Service Infrastructure**	a computational job through the DEP Service Infrastructure (e.g., HPC, Cloud). The submission includes details about the job configuration, required resources (CPU, GPU, memory, storage), and any related input datasets.
Step 3: Service Infrastructure – sends metrics – **Resource Usage & Environmental Impact Tracking**	At the end of the day, the Service Infrastructure sends the resource consumption metrics to the Resource Usage & Environmental Impact Tracking component, which collects the data and stores it in the CRMS.

Use Case Scenario #2

DEP Administrator

The DEP Administrator deploys, configures, and operates the DEP environment within the compute centre, ensuring its connection to external systems such as AAI, AI model stores, and data holdings. The Administrator retrieves reports on the DEP's resource usage, observes the infrastructure's availability, and sets quotas (e.g. credits) allocation for the DEP environment.

System Components Involved

Table 26: TUC4 Software Components

List of UC Components and their Role	
Component Name	**Role/Involvement in this Scenario**
Resource Usage & Environmental Impact Tracking	This logical component collects metrics on CPU/GPU/storage/network usage and environmental indicators.
Credits Translation Policy Management	This logical component applies policy rules to convert resource usage into credits.
Credits Distribution Policy Management	This logical component allocates quotas (i.e. credits) to projects based on DEP Administrator-defined policies.
Credits Allocation Registry	This logical component reports a detailed record of credit ownership, consumption, and related information.

Service Infrastructure	The service infrastructure that supports resources, including CPU, memory, storage, and networking.
DEP AAI	DEP's Authorisation Framework.

Preconditions

- The interconnection between system components and external services is established through the CRMS's RESTful API, which provides a standardised interface for accessing, configuring, and exchanging data related to credit allocation, usage tracking, policy enforcement, and user management across the DEP environment.
- The CRMS is deployed and operational in the DEP environment.
- The DEP environment is connected to the DEP AAI.
- The DEP Administrator is authenticated and authorised to configure CRMS translation and distribution policies.
- Service types (e.g., compute, storage) and related metric types (e.g., core-hours, GB*Hours) are defined in the system.

Expected Outcome / Validation Reporting

Table 27: TUC4 Scenario Flow and Test Cases

Use Case Testing and Validation Plan	
Scenario Step	**Test Case & Acceptance Criteria**
Step 1: DEP Administrator – gets authorised – **DEP AAI**	DEP AAI successfully authorises the DEP Administrator and grants access to administrative configuration functions.
Step 2: DEP Administrator – bootstraps Tier System of services – **Resource Usage & Environmental Impact Tracking**	The DEP Administrator correctly defines the tier structure, and it is reflected in the accounting module; services and metric definitions are outlined appropriately.
Step 3: DEP Administrator – validates capacity and unit cost – **Credits Translation Policy Management**	The DEP Administrator validates the capacity and unit cost of the service to set quotas (e.g., credits).

Step 4: DEP Administrator – sets credits distribution policies – **Credits Distribution Policy Management**	The DEP Administrator sets quotas (e.g. credits) per project, based on capacity and unit cost of the service.
Step 5: Service Infrastructure – sends metrics – **Resource Usage & Environmental Impact Tracking**	At the end of the day, the Service Infrastructure sends the resource consumption metrics to the Resource Usage & Environmental Impact Tracking component, which collects the data and stores it in the CRMS.
Step 6: DEP Administrator – checks service's capacity limit and unit cost – **Credits Translation Policy Management**	The system returns the current service (regarding the associated metric definition) unit cost and its capacity status, and indicates if the defined capacity threshold has been exceeded.
Step 7: DEP Administrator – checks service's available quotas – **Credits Distribution Policy Management**	The system returns the current service (regarding the associated metric definition) available quotas.
Step 8: DEP Administrator – reports consumption and credits usage portion – **Credits Allocation Registry**	The Credits Allocation Registry reports the consumption and corresponding credits usage portion allocated to the DEP resources to the DEP Administrator.

4. Requirement Mapping and Validation Analysis

This section presents the process and outcomes of mapping the consolidated requirements to the scientific and technical use case scenarios defined in Section 3. The first part reviews the updates introduced since D5.1, which mainly concern the assignment of time frames for validation and the explicit linkage of requirements to specific use cases and scenario flows. Building on this, the section provides an analysis of the mappings, showcasing the extent to which requirements are covered by the scenarios and highlighting their overall distribution across functional and non-functional categories as well as across WP. All the requirements and their updates can be found in Annex II: Updated RI-SCALE Requirements.

4.1. Requirement Updates

Since the publication of D5.1, the list of accepted requirements has undergone the planned updates to support the planning and execution of validation activities. The most notable addition concerns the **Time Frame** field, which defines when each requirement will be **validated**, rather than when it will be developed. While development and validation are closely related in time, this distinction ensures that validation milestones are realistically aligned with both component maturity and use case execution schedules.

Several minor refinements were also introduced to harmonize terminology, update dependencies, and clarify requirement ownership across the technical work packages. These refinements were necessary to maintain consistency with the progress of the DEP architecture, the use case developments, and the updated validation scenarios.

The assignment of requirements to the **1st** or **2nd DEP release** followed a structured decision logic. Four main factors were considered:

1. **Priority**: Based on the MoSCoW classification defined in D5.1, "Must" and "Should" requirements were generally included in the first release to ensure early validation of critical functionalities.
2. **Maturity**: Requirements depending on technologies or components still under active development were shifted to the second release to allow sufficient stabilization before validation.
3. **Dependencies**: When a requirement relied on another WP's component (e.g., WP3's AI frameworks depending on WP2's data orchestration services), its validation was scheduled only after the prerequisite component became available.

4. **Alignment with Use Case maturity**: For requirements derived from scientific or technical use cases, the readiness of the associated use case influenced the release planning. Use cases expected to reach operational readiness later in the project led to corresponding requirements being assigned to the second DEP release.

Together, these criteria ensured a balanced distribution of validation activities across the two releases, aligning technical feasibility with project priorities and ensuring that all core platform functionalities can be validated in realistic conditions.

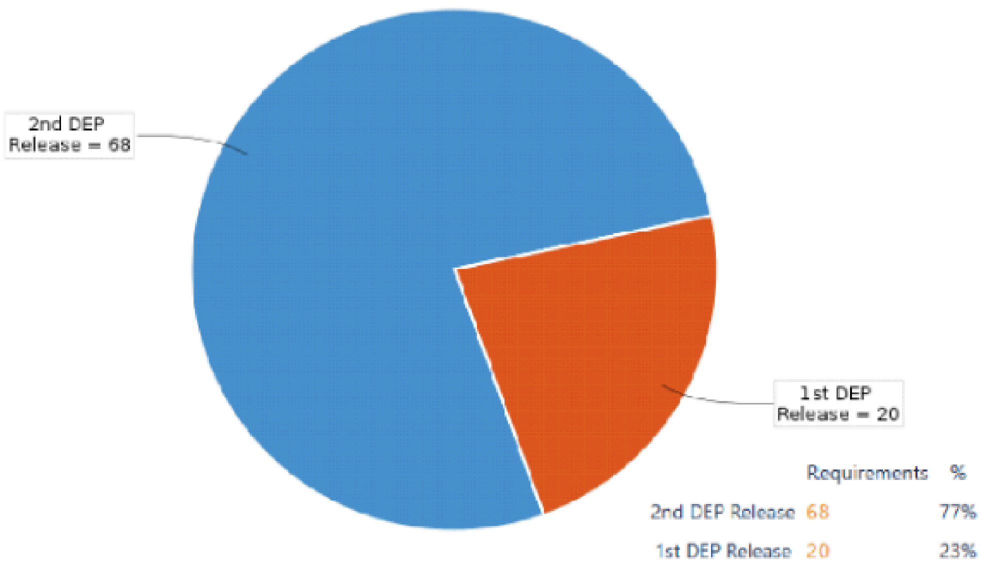

Figure 2: Requirement Distribution by DEP Release

The pie chart in Figure 2 illustrates the overall distribution of requirements across the two planned DEP releases. Out of the total accepted requirements, 68 (77%) are planned for the second DEP release, while 20 (23%) are assigned to the first release. This distribution reflects the project's phased implementation strategy, where the first release focuses on validating the core functionalities of the DEP, ensuring that foundational services related to data, AI, and access management are stable and interoperable. The second release encompasses the broader integration of advanced features and scientific use cases, which depend on the maturity of these initial components.

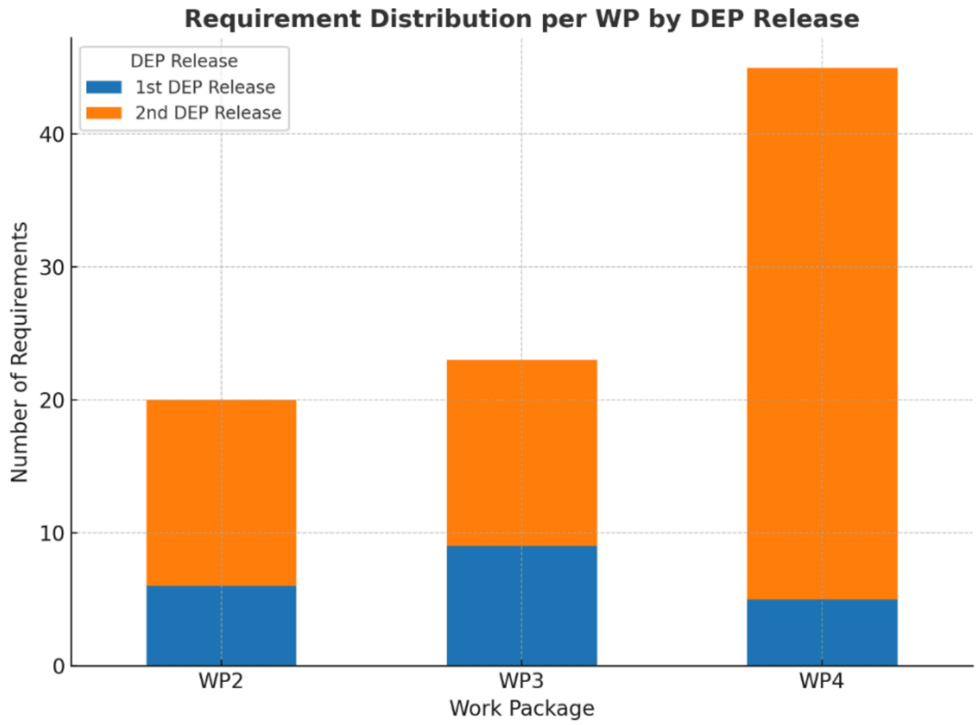

Figure 3: Requirements per Work Package by DEP Release

Figure 3 illustrates how requirements are distributed across WP2 (DEP Data Lifecycle Management), WP3 (Scalable AI Solutions), and WP4 (Access Management Technologies), split between the first and second DEP releases. From the chart, it is evident that WP4 holds a substantial share of requirements planned for the second DEP release, reflecting the sequential nature of DEP integration. Access, identity, and credit management functionalities depend on stable data, computation and AI layers from WP2 and WP3, which must be validated first. In contrast, WP2 and WP3 show a more balanced or front-loaded distribution toward the first release, aligning with the project's objective to make the data and AI layers operational early to support downstream validation in use cases.

This pattern mirrors the architecture outlined in the project DoA: WP2 establishes the foundational data layer, WP3 builds AI-driven analytics on top of it, and WP4 provides the overarching access and policy control framework that integrates all components.

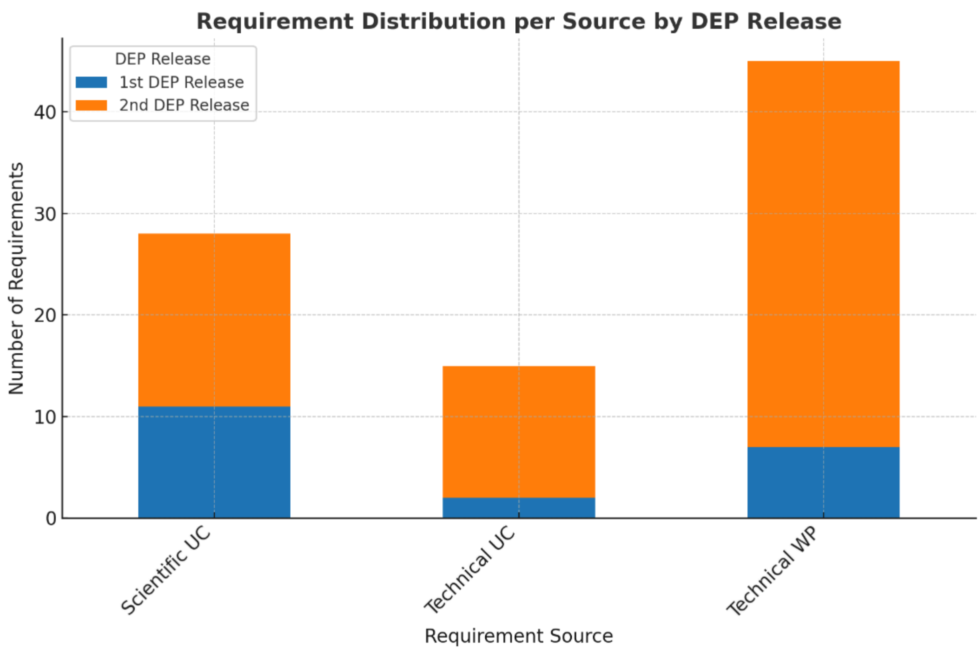

Figure 4: Requirement Source vs DEP Release

The chart in Figure 4 shows how requirements originating from different sources, SUCs, TUCs, and Technical Work Packages (WP2–WP4), are distributed across the two DEP releases. The data reveals that scientific use case requirements are primarily concentrated in the second DEP release, while most technical WP requirements are planned for the first release. This reflects the logical dependency between the two categories: technical requirements define the platform capabilities, while scientific use cases depend on those capabilities to execute their validation workflows.

4.2. Requirement-to-Scenario Mapping Approach

The mapping of requirements to use case scenarios represents a critical methodological step in ensuring that every accepted requirement can be validated through measurable and traceable activities. The objective is to establish a direct link between the project's functional and non-functional requirements, the DEP components that implement them, and the scientific and technical use cases that will exercise and verify their performance.

This mapping guarantees full traceability throughout the validation process, connecting each element in the chain from *requirement*-to-*use case scenario*-to-*DEP component*-to-*test-case/acceptance* criteria.

The approach follows several key principles:

1. **Comprehensive Coverage**: Every accepted requirement is linked to at least one validation use case scenario. This ensures that the entire set of requirements defined in D5.1 will be validated in the course of the two DEP releases.

2. **Realistic Mapping Logic**: Mapping is based strictly on whether the use case scenario actually exercises the component related to the requirement. Artificial or redundant associations are avoided. The goal is to achieve meaningful validation rather than numerical completeness.
3. **Differentiation Between Use Case Types**:
   - **Scientific Use Cases (SUCs):** Validate end-to-end workflows, domain-specific datasets, and the accuracy or performance of AI models operating on real data.
   - **Technical Use Cases (TUCs):** Focus on the DEP's cross-cutting capabilities, including data orchestration, scalability, interoperability, and access management.
4. **Multi-Mapping Flexibility**: The mapping process supports many-to-many relationships. A single requirement may be validated in multiple use cases, and a single use case may validate multiple requirements. This flexibility reflects the interconnected nature of the DEP architecture, where the same service (e.g., identity management or orchestration) may support several validation contexts.
5. **Dependency Awareness**: Dependencies between requirements and components are accounted for when assigning mappings. For example, a WP3 requirement related to model execution cannot be validated unless the data ingestion and orchestration services provided by WP2 are available in the corresponding use case flow. This approach preserves logical sequencing and technical realism.

Implementation in Jira

The mapping process is implemented and tracked in Jira through a dedicated field named "Validation Use Case", allowing each requirement (RSREQ Jira issue) to be linked to one or more SUC or TUC identifiers. The system ensures consistent traceability and provides a central repository for all mappings.

Each requirement also includes a Component field that identifies the relevant DEP subsystem (e.g., WP2, WP3, WP4). When performing the mapping, the component indicated in the requirement must align with the components described in the selected use case scenario. In cases where inconsistencies arise, mapping is reviewed jointly by the WP lead and the WP5 validation coordination team to maintain accuracy.

4.3. Analysis of Mappings and Scenario Coverage

This section presents the quantitative results derived from the mapping of requirements to scientific (SUC) and technical (TUC) use case scenarios. Building on the methodology outlined in Section 4.2, the analysis evaluates how effectively the accepted requirements are distributed across the use cases and work packages, providing an overview of validation coverage and balance.

The purpose of this analysis is to verify that all core functionalities of the DEP are represented within the validation portfolio and that overlaps or gaps are identified early. The figures included in this section summarise the current state of requirement coverage, highlight areas of concentration or dependency between WPs and use cases, and provide a basis for tracking validation progress in later project phases.

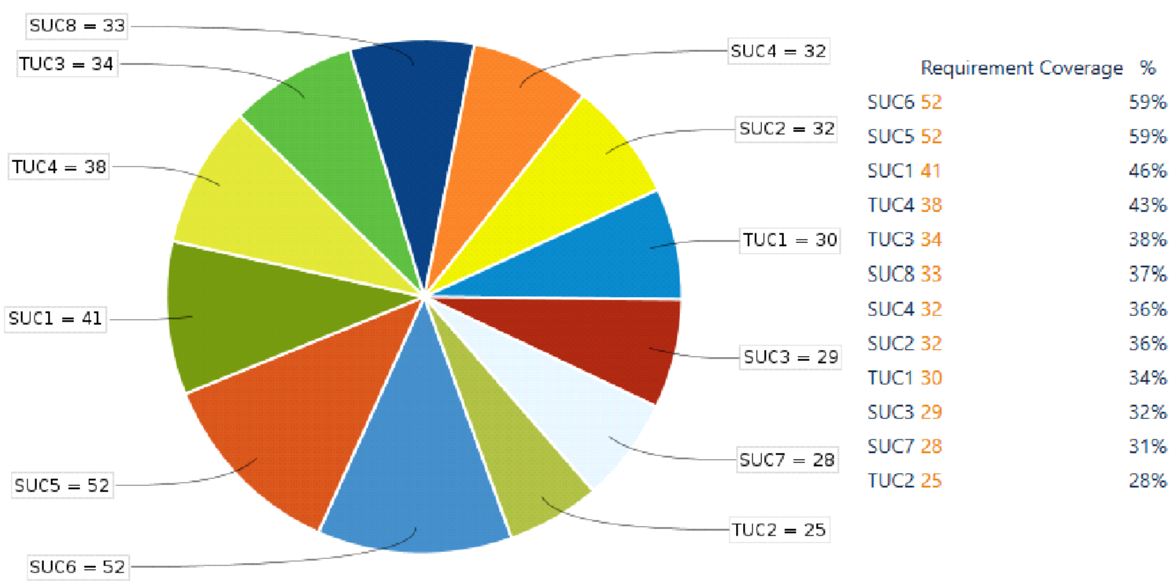

Figure 5: Requirement Coverage across Use Cases

The chart above (Figure 5) illustrates how the accepted requirements are distributed across the scientific (SUC) and technical (TUC) use cases, providing an overview of the current validation coverage. The results show that SUC5 and SUC6 present the highest coverage, each validating around 59% of the requirements, followed by SUC1 and TUC4, which account for approximately 46% and 43%, respectively. This strong representation reflects the maturity and completeness of these scenarios, which integrate multiple DEP components and serve as central validation points for both AI and data lifecycle functionalities.

Lower coverage levels are observed for SUC7, SUC8, and TUC2, which range between 28% and 31%. These use cases are more specific in scope and focus on targeted validation objectives rather than complete end-to-end pipelines.

It is important to mention that even the use cases with lower coverage still address a substantial portion of the overall requirements, ensuring that each contributes meaningfully to the overall validation process and collectively supports comprehensive coverage across the DEP ecosystem.

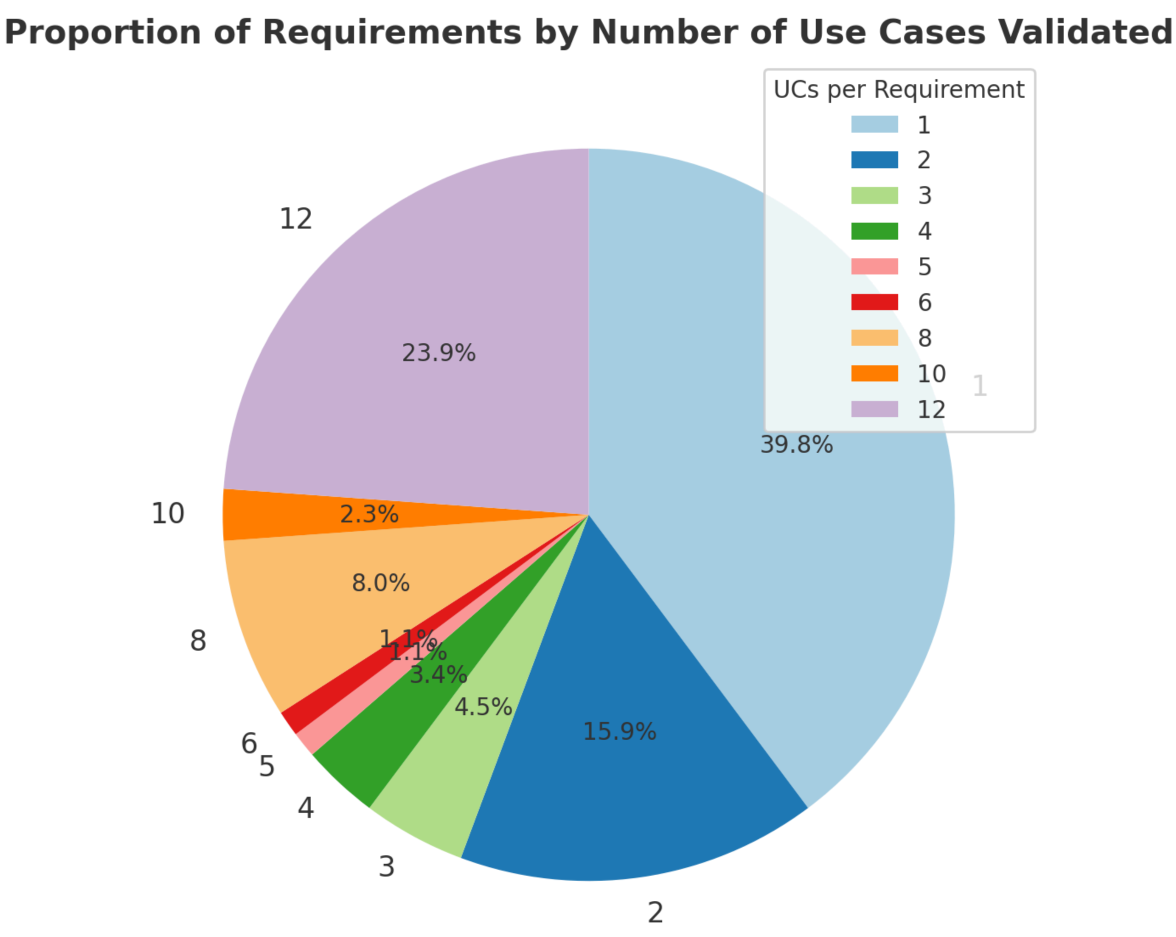

Figure 6: Requirement Proportional Validation Coverage

The chart (Figure 6) presents how many use cases are linked to each requirement, illustrating their validation coverage across the RI-SCALE portfolio. Most requirements (around 40%) are validated in a single use case, reflecting the presence of use case–specific developments that address the particular objectives or technical needs of each scenario. These requirements usually focus on tailored implementations within the data lifecycle or AI workflow of a given use case and are not intended for cross-domain validation.

A smaller portion (about 16%) is validated in 2 or 3 use cases, generally representing shared platform functions such as data orchestration or AI model deployment, which span multiple workflows but remain confined to particular validation contexts.

The most notable group, comprising roughly 24% of requirements, is validated across all 12 use cases. These are general, cross-cutting requirements primarily originating from WP4, including functionalities such as user authentication, access control, and credit management, which are invoked by every scientific and technical workflow. Alongside these, several WP2-related requirements, particularly those addressing data holding, replication, and computation orchestration, also appear across many use cases. Their inclusion reflects the foundational role of WP2 in enabling unified data access and processing within the DEP. This coverage demonstrates that many of the core DEP developments are intentionally designed to support multidisciplinary use cases, ensuring interoperability and reusability across scientific domains.

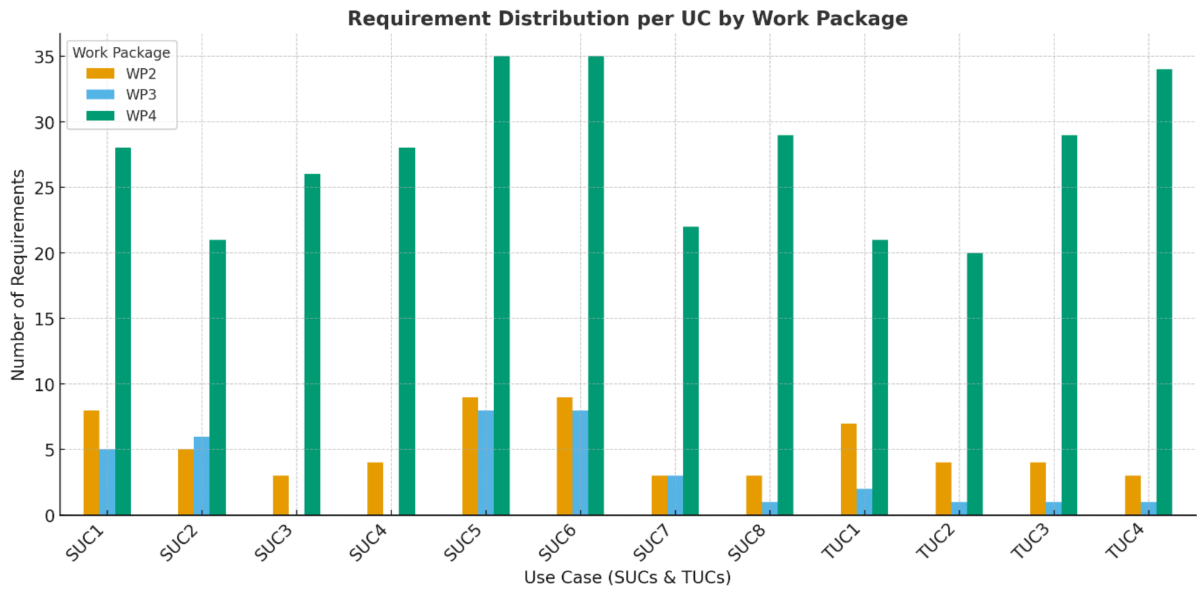

Figure 7: Requirement Distribution across WP's

Requirements from WP2 are validated across nearly all use cases (Figure 7) because they define the underlying data handling logic that enables the entire DEP to operate as a coherent system. These requirements address not only replication and orchestration but also how data are described, discovered, and moved efficiently across distributed research infrastructures. Their rationale focuses on ensuring that each use case (regardless of scientific domain) can transparently access and process datasets without being constrained by where those data are physically stored. This includes the automation of data staging and locality-aware computation, where workflows dynamically move processing tasks closer to data sources, reducing latency and improving resource efficiency.

Requirements from WP3 are distributed across a smaller set of use cases (Figure 7), reflecting the specialised nature of AI-related developments within the DEP. These requirements focus on enabling model training, execution, explainability, and reuse of AI assets, forming the analytical intelligence layer of the platform. Unlike WP2, which underpins all validation activities, WP3 contributes specific functionalities that become active only when the scenario involves machine learning or data-driven prediction. This explains their stronger presence in use cases such as SUC5 and SUC6, where AI methods are directly applied to complex biomedical and imaging datasets, and their more limited appearance in others that do not require model development or inference. The mapping pattern confirms that WP3 developments are not general-purpose services but targeted modules that extend the platform's capabilities to advanced analytics.

WP4 contains the highest number of requirements overall, which naturally gives it a commanding presence across all use cases. These requirements define the mechanisms that ensure secure, transparent, and accountable access to data and computational resources within the DEP. They cover functionalities such as identity federation, authorization, credential management, and usage accounting, which are essential to every workflow regardless of scientific domain. As discussed earlier and confirmed again in this diagram, WP4 also exhibits the most complete coverage across all SUCs and TUCs, reflecting the fact that its components are exercised in every validation scenario. This is consistent with its role as the platform's unifying operational layer, governing how users, data, and services interact. The dominance of WP4 in the mapping results, therefore, illustrates both its cross-cutting nature and its importance in maintaining the reliability, trust, and interoperability of the DEP as a whole.

5. Mapping Validation Outputs to KPIs

This section presents how the results of the validation activities will be linked to the KPIs. It defines the approach for translating validation evidence into measurable progress indicators, ensuring that each test activity contributes to assessing the achievement of RI-SCALE's overall objectives

5.1. Traceability Model

The validation activities in RI-SCALE are designed to provide measurable evidence of progress toward the project's KPIs. Traceability is achieved through a structured chain that links high-level project objectives, as defined in the DoA, to the concrete validation results produced by each scientific and technical use case.

This model integrates three main information sources:

1. The **DoA KPIs**, which define the measurable outcomes expected from the DEP, including the number of validated models, datasets, pilots, and participants.
2. The **use case scenario descriptions**, which outline the workflows, actors, and validation conditions under which measurable outputs will be produced.
3. The **accepted requirements**, which capture the technical and functional elements of the DEP that enable these KPIs to be fulfilled and assessed.

Traceability is established by mapping each requirement to the use case scenarios in which it is validated and then linking these scenarios to the relevant KPIs that they support. This creates a continuous line of evidence from the system functionality to the expected project results. For example, requirements defining data replication and orchestration contribute to the validation of dataset availability and size indicators, while those addressing AI model execution and explainability connect to metrics on model training and deployment.

The traceability framework focuses on the subset of KPIs that can be directly validated and monitored through requirement fulfilment and use case implementation. These KPIs represent measurable project outcomes that can be objectively demonstrated within the validation activities. Other KPIs, which refer to broader strategic, organisational, or outreach objectives, are tracked through complementary project mechanisms and fall outside the scope of this validation-based traceability model.

In the following section, the current mapping and status of the KPIs covered by this framework are reviewed, showing how the validation planning aligns with measurable project outcomes.

5.2. KPI Mapping and Status

This section presents the KPI–Validation UC Confusion Matrix (Table 28), which connects the project's KPIs to the validation use cases through which they will be demonstrated. Building on the traceability model defined in Section 5.1, this mapping provides a clear view of how each measurable objective from the DoA is supported by the DEP validation planning.

Each KPI is linked to the relevant scientific and technical use cases, identifying the validation pathways and the partners responsible for producing measurable evidence. The table also includes a short validation rationale describing how each KPI will be addressed during the implementation and pilot phases. Together, these mappings form the basis for monitoring progress and confirming the fulfilment of DEP-related objectives throughout the project's validation cycle.

Table 28: KPI-validation UC Confusion Matrix

KPI-validation UC Confusion Matrix		
KPI (ID-Name-Definition)	**Source of Validation**	**Planned Validation Method**
KPI#3: No. of AI frameworks/toolboxes offered within DEPs (2) AI frameworks that are available for RI operators and RI users for the DEP technology	SUC1, SUC2, SUC5, SUC6, SUC8	The validation of this KPI will be performed through the scientific use cases that plan to employ DEP's AI capabilities for model execution, management, and analysis. SUC1 and SUC2 apply AI methods for climate modelling and anomaly detection, while SUC5, SUC6, and SUC8 focus on biomedical and imaging applications. Across these scenarios, two complementary frameworks are planned for validation: the AI Model Hub, enabling registration and sharing of trained models, and the AI Computing Framework, supporting large-scale distributed training and inference.
KPI#4: No. of AI models offered within DEPs (15) AI models available for users in DEP AI repositories	SUC1, SUC2, SUC5, SUC6, SUC7, SUC8, TUC1, TUC3	This KPI will be validated through the use cases that involve the creation, adaptation, or reuse of AI models using the DEP's AI frameworks. SUC1 and SUC2 generate AI models for climate prediction and anomaly detection, while SUC5, SUC6, SUC7, and SUC8 produce models for biomedical imaging, pattern recognition, and generative discovery. The technical use

		cases TUC1 and TUC3 complement these efforts by validating the scalability, deployment, and interoperability of the trained models within the DEP environment.
KPI#5: No. of DEP installation pilots deployed (4) DEP installation reports	All	This KPI will be validated through the deployment of the four planned DEP installation pilots described in the DoA. These installations represent the operational environments where both scientific and technical use cases will execute their validation activities. Each pilot corresponds to a distinct deployment context, including EuroHPC, e-Infrastructure, and RI-based environments and together they provide the backbone for data access, computation, and AI model execution across all UCs.
KPI#6: Total size of datasets used in DEP pilots (2PB) Inventory of data sources successfully replicated and served from data holdings to DEPs	SUC1, SUC2, SUC3, SUC4, SUC5, SUC6, SUC7, SUC8, TUC2, TUC3	The validation of this KPI is based on the datasets employed across both scientific and technical use cases. The scientific UCs (SUC1–SUC8) collectively cover a wide spectrum of domains and data volumes: climate and atmospheric data (CMIP6, CORDEX, EISCAT) in the multi-terabyte range, biomedical and imaging datasets (WSIs, BioImage Archive) reaching tens of terabytes, and generative AI datasets potentially extending into the petabyte scale. The technical UCs, particularly TUC2 and TUC3, will further validate the handling, transfer, and orchestration of these datasets across distributed infrastructures, confirming DEP's scalability and data lifecycle performance. Combined, these validation activities are expected to reach the 2 PB target defined in the project KPIs.
KPI#7: No. of AI models trained in DEP pilots (8) AI models used for the validation use cases	SUC1, SUC2, SUC5, SUC6, SUC7, SUC8, TUC2, TUC3	This KPI will be validated through the use cases that plan to train or re-train AI models within the DEP environment. SUC1 and SUC2 will generate climate-related models for downscaling and anomaly detection, while SUC5 and SUC6 will train biomedical

		imaging and diagnostic models on whole-slide and microscopy data. SUC7 and SUC8 extend the training activities to advanced imaging and generative discovery contexts. The technical UCs, especially TUC2 and TUC3, ensure the trained models can be executed, scaled, and benchmarked on EuroHPC infrastructures. Together, these scenarios are planned to produce and validate at least eight trained models during the DEP pilot phase.
KPI#8: No. of use cases developed for DEP validation purposes (12) DEP validation use case reports	All	This KPI is addressed through the creation and detailed description of the twelve validation use cases presented in D5.2. These include eight scientific and four technical scenarios that define the structure, actors, components, and workflows required for DEP validation. While the KPI is not yet fulfilled, the work completed and documented in this deliverable provides the necessary framework to achieve it during the implementation and validation phases of the project.
KPI#9: No. of Data Spaces with which DEP is interoperable (3) Data Space interoperability validation reports	TUC1, TUC2, TUC3	This KPI will be validated through the technical use cases that focus on enabling and testing interoperability between the DEP and external data ecosystems. TUC1 and TUC2 implement and validate the data access, orchestration, and federation mechanisms that allow the DEP to communicate with external infrastructures.
KPI#11: No. of SMEs participating in the DEP validation (10) SMEs with active participation in the setup and performing of validation use cases	SUC1, SUC3, SUC4, TUC2, TUC3	(5) SME partners are currently involved in DEP validation activities (**Comprimato Systems**, **Fragmentix**, **Hypermeteo**, **JNP**, **Neuraspace**). They contribute to component development and use case execution across climate, space, and technical pilots. While the current number is five, additional SMEs will be engaged during the pilot phase to reach the KPI target of ten.
KPI#13: No. of academic institutes participating in DEP	SUC1–SUC8, TUC1–TUC3	(7) academic partners are directly contributing to DEP validation (**KTH**, **LTU**, **Masaryk University**, **Medical University of Graz**, **TU Wien**, **University of Trento**,

validation (8) Entities participating in technology validation		**Universitat Politecnica de Valencia**). Their participation spans climate, space, and biomedical use cases as well as orchestration and AI workflows. Combined with contributions from research infrastructures, this meets and slightly exceeds the KPI target of eight.

6. Conclusions and Next Steps

This deliverable marks the consolidated effort to establish the foundation for the validation phase of the RI-SCALE Platform. Building on the requirements and design considerations defined in D5.1, it translates the project's conceptual framework into a structured and operational validation plan. The purpose has not been to execute validation activities, but to ensure that subsequent testing and verification phases are grounded in a clear methodology that links system functionalities, use case objectives, and project KPIs. Through this work, the project has defined a unified process that connects platform development with use case actuation.

The validation framework presented in this deliverable reflects the collective input of technical, scientific, and coordination partners across RI-SCALE. It introduces a comprehensive structure of twelve validation scenarios, eight scientific and four technical, each representing distinct workflows, operational environments, and user interactions within the DEP ecosystem. These scenarios serve as the backbone for testing the platform's components, including data lifecycle management, scalable AI services, and access management technologies, both individually and in integrated end-to-end contexts. Their definition ensures that validation is not limited to component-level testing but extends to the real operational workflows that the platform is designed to enable.

A key outcome of this deliverable is the establishment of full traceability between the requirements consolidated in D5.1, the validation scenarios defined in this plan, and the project KPIs. This mapping ensures a consistent bridge between development and validation, enabling transparent tracking of how individual platform features contribute to scientific outcomes. The resulting framework ensures that the DEP's technical maturity and the use cases' scientific objectives evolve in synchrony.

The current level of integration achieved between the DEP architecture and the use case planning demonstrates that the project is ready to advance into the development and validation execution phase. The first iteration of the platform, incorporating initial functionalities from WP2, WP3, and WP4, will provide the necessary foundation for initiating test execution and collecting evidence of performance and interoperability. The use of Confluence and Jira as the central management tools ensures that validation activities, test cases, and results are fully traceable, version-controlled, and transparently accessible to all partners.

Looking forward, the next steps focus on the transition from planning to execution. The deployment of the first validation cycle will verify the baseline functionalities and interoperability of the DEP, leading to the First Validation Report (D5.4). This phase will build on the structured roles, responsibilities and UC scenarios established during planning, with Use Case Leaders coordinating and executing the scenario-based tests under the methodological guidance of Task 5.2 and Task 5.3 leaders, and with continued technical support from WP2–WP4 to ensure full component integration. Following this, validation test cases and acceptance criteria will be refined based on the feedback from the first DEP release(D5.3), ensuring that test coverage evolves alongside the platform's maturity. Continuous updates to Jira and Confluence tracking will preserve traceability and provide an auditable record of progress. Furthermore, input from WP6 and WP1 will remain integral to the process, with WP6 contributing external and stakeholder-driven perspectives that refine priorities and ensure alignment with the expectations of the wider research infrastructure community, and WP1 ensuring that all validation activities respect ethical, legal, and compliance frameworks throughout their execution.

Annex I: Use Case Scenario Description Template

This template was created to support the validation planning process in RI-SCALE by providing a consistent structure for documenting the 12 use case scenarios (4 technical and 8 scientific). Its purpose is to enable the systematic validation of both the **DEP components**, and the **requirements** consolidated in Deliverable D5.1. Through these scenarios, we ensure that the platform implementation addresses the needs and objectives defined by the use cases, and more broadly, aligns with the overall goals of the project.

Use Case ID: Use the following format to identify each use case:

- SUC-# for Scientific Use Cases
- TUC-# for Technical Use Cases

The numbering and classification (e.g. SUC-1, TUC-3) follow the official identifiers used in DoA. Example: SUC-3, TUC-2

Use Case Name: Use the official title of the use case as it appears in the DoA.

Description/Purpose: Provide a short overview of the use case, based on the corresponding description in the DoA. The goal is to give readers enough context about what the use case is trying to achieve, why it matters, and what general functionality it exercises within the DEP.

Actors and Roles

- Please make sure your UC scenarios use the agreed unified roles:
    - **DEP Administrator**: Responsible for internal operations in the DEP System.
    - **Data Scientist**: Any end-user operation performed in the context of the DEP.
    - **Domain Expert**: A UC-specific person with domain knowledge (e.g. "Domain Expert: doctor who assesses the validity of results").
- Define the role clearly and specify what they do in the context of your UC.
- Replace previously established roles with these unified roles.

Use Case Scenario Template

System Components Involved

List the major components or modules of the system that are engaged in this scenario.

Component Name	Role/Involvement in this Scenario
AI Model Hub	Serves the AI models used in the scenario.

Guidance: Include any relevant backend services, interfaces, tools, or platform features from WP2–WP4. Only include components that play a meaningful role in this specific scenario.

Preconditions

Any conditions or system state that must be true before this use case can begin (i.e. the researcher is authenticated and has selected a dataset to analyse).

Guidance: List any prerequisite state or required data. This helps set the starting context for the scenario.

Scenario Flow (Steps)

Describe the main sequence of actions in the scenario using structured triplets in the form: **(Subject)** - (Predicate) - **(Object)**

- Subject: An actor or system component already listed in the "Actors/Roles" or "System Components Involved" sections
- Predicate: A clear, active verb describing what is being done
- Object: Another actor, component, or data item that is the target of the action

Example:

1. **Data Scientist** -uploads dataset to - **AI Model Hub**
2. **AI Model Hub** - triggers validation using - **Model Validator**
3. **System**- sends confirmation message to - **End User**

Guidance:

- Stick to one action per line,

- Use consistent naming for roles and components (as defined in previous sections),
- Number the steps always.

Expected Outcome / Validation Reporting

This section formalises the **input of the use case partners** by linking each step of the scenario to concrete test cases and acceptance criteria. These will guide the validation planning for D5.2. For each step, define what needs to be tested, how success will be measured, and provide space for documenting results.

Scenario Step	Test Case & Acceptance Criteria	Validation Results
Step 2: AI Model Hub - performs inference on - **input**	The system returns inference results within 5 seconds and accuracy exceeds 85% on the benchmark set.	[To be filled during validation]
Step 3: DEP UI - sends result to - **End User**	The result is displayed in the correct format and matches the expected output for the input sample.	[To be filled during validation]

Guidance:

- **Scenario Step:** Reference the exact step number and description from the *Scenario Flow (Steps)* section.
- **Test Case & Acceptance Criteria**:
  - Define what will be tested at this step.
  - Be **precise** and **measurable** (e.g. accuracy thresholds, time limits, format checks).
  - Avoid vague outcomes like "should work" — define success clearly.
- **Validation Results:** Leave blank initially. This will be filled after validation execution or during reporting.

Annex II: Updated RI-SCALE Requirements

Functional Requirements

WP2 Functional Requirements

Requirement Jira key	Description	Rationale	Component That Fulfils It	Time Frame (Validation)	Validation Use Case
RSREQ-101	Infrastructure to operate GROQ card in a testing/proof of concept environment	To test inference on GROQ TPU cards, they need to be run in an HPC environment	Computing Site Integration Interface (WP2-T2.5)	2nd DEP Release	TUC3
RSREQ-24	The DEP should provide a centrally managed Git server based on GitLab to support version control of training code, configuration files, and related project resources. This Git server should: • Be accessible from all compute sites of the DEP. • Be integrated with the DEP's user management system, ensuring that access control (e.g. project-based visibility, permissions, and roles) is centrally	Collaborative AI model development within the DEP requires reliable, accessible, and secure version control for code and associated artifacts. A GitLab server provides users with a standard and widely adopted platform for collaborative development. The AI Model Hub needs a centralised repository to collect the code of tracked models. CI	Computing Site Integration Interface (WP2-T2.5)	1st DEP Release	SUC5, SUC6

	managed and aligned with the user's project membership and roles in the DEP. ● Be suitable for both interactive use (via GitLab web interface) and scripted operations (via Git over SSH/HTTPS and CI/CD API integrations). ● Support continuous integration (CI) pipelines for running automated jobs on the DEP compute infrastructure, including support for GPU-enabled CI runners to allow automated testing to be executed as part of version-controlled workflows.	support, especially with GPU capabilities, further enhances automation and reproducibility of AI development. This requirement supports development in Use Cases 5 and 6, and contributes to KPI03: No. of AI frameworks/toolboxes offered within DEPs.			
RSREQ-25	The DEP should provide a centrally managed container registry to support container-based workflows across all compute sites. The service should support Apptainer (formerly Singularity) containers. Examples of suitable solutions include Harbor or GitLab Container Registry. The registry should:	Reliable container management is essential for reproducible, portable, and secure execution of AI workflows in the DEP. The AI Model Hub should use the container registry to execute tracked models. Furthermore, this directly supports	Computing Site Integration Interface (WP2-T2.5)	2nd DEP Release	SUC5, SUC6

	• Be accessible from all DEP compute sites. • Allow users to upload, version, and share container images associated with their projects, with access controlled via the DEP's user management system. • Ensure Apptainer containers can be executed on all supported compute infrastructures.	collaborative development in use cases 5 and 6.			
RSREQ-38	The DEP needs to be able to copy data onto at least one of the EuroHPC systems that DestinE is currently using, Lumi (CSC), Leonardo (Cinceca), MN5 (Bsc)	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. Hence the DEP needs to be able to copy data onto at least one of the EuroHPC systems that DestinE is currently using, Lumi (CSC), Leonardo (Cinceca), MN5 (Bsc)	Computing Site Integration Interface (WP2-T2.5)	2nd DEP Release	TUC1
RSREQ-39	The DEP needs to be able to spawn virtual kubelets onto at least one of the	The technical use case states: Evaluate and measure the scale	Computing Site Integration	2nd DEP Release	TUC1

	EuroHPC systems that DestinE is currently using, Lumi (CSC), Leonardo (Cinceca), MN5 (Bsc).	at which the DEP can work on a EuroHPC machine while serving a challenging AI model with large data from DestinE. Hence the DEP needs to be able to spawn virtual kubelets onto at least one of the EuroHPC systems that DestinE is currently using, Lumi (CSC), Leonardo (Cinceca), MN5 (Bsc)	Interface (WP2-T2.5)		
RSREQ-89	Implement data transfer mechanisms and protocols at the computing sites	Needed to get data from data-owners to the computing infrastructure	Computing Site Integration Interface (WP2-T2.5)	2nd DEP Release	All
RSREQ-90	Implement federated authentication / authorisation protocol at the compute sites	Need to give data owners secure access to computing sites	Computing Site Integration Interface (WP2-T2.5)	1st DEP Release	All
RSREQ-106	The platform must allow to download climate datasets through customizable queries from different data platforms, such as those within the Copernicus Climate Data Store or through the ESGF data platform.	This functionality is needed for extracting the data needed for training the ML model and for inference of downscaled data for use case WP3.3.	Data Discovery & Popularity Service (WP2-T2.3)	2nd DEP Release	SUC1

RSREQ-18	The platform should support access to usage statistics on ESM data (i.e., CMIP) through customizable queries, allowing also for basic aggregation functions.	This functionality is needed for (i) extracting the data needed for training the ML model and (ii) for feeding the trained model with the data for inference for the use case in WP3.3. It contributes to KPI#4 (No. of AI models offered within DEPs), KPI#7 (No. of AI models trained in DEP pilots), and KPI#8 (No. of use cases developed for DEP validation).	Data Discovery & Popularity Service (WP2-T2.3)	2nd DEP Release	SUC2
RSREQ-45	The Data Popularity service must provide a way for extracting the most downloaded ESM data (e.g., CMIP) from the ENES research infrastructure.	This functionality is needed for properly driving the data ingestion pipeline and supporting the DEP caching mechanism. It contributes to KPI#6 (Total size of datasets used in the DEP pilots).	Data Discovery & Popularity Service (WP2-T2.3)	2nd DEP Release	SUC1
RSREQ-100	The storage systems in RI-SCALE: • RI Data holdings (long-term storage). • Storage @ E-Infrastructure for dataset access	Without that support, the authentication of storage operations will be extremely difficult and fall outside the trust network built in WP4.	Data Holdings Integration Framework (WP2-T2.4)	2nd DEP Release	All

	• Must support OIDC token authentication based on the work prepared in WP4.				
RSREQ-21	The platform should support access to CMIP climate projections and reanalysis data.	This functionality is needed for extracting the data needed for training the ML model and for inference of downscaled data for use case WP3.3.	Data Holdings Integration Framework (WP2-T2.4)	1st DEP Release	SUC1
RSREQ-42	Newly generated DestinE training data should be retrieved to the DEP to make it available for AIFS training runs on at least one of the EuroHPC systems that DestinE is currently using, Lumi (CSC), Leonardo (Cinceca), MN5 (Bsc)	There is also a performance component here, 100 Gb/s roughly between DEP and DestinE data holdings, data space. Hence, DEP should have enough storage available for the AIFS training datasets and transfer 50-100 TB in a few minutes (100 Gb/s) to at least one of the EuroHPC systems that DestinE is currently using, Lumi (CSC), Leonardo (Cinceca), MN5 (Bsc)	Data Holdings Integration Framework (WP2-T2.4)	2nd DEP Release	TUC1

RSREQ-22	The DEP should provide a robust and secure process for orchestrating the transfer of Whole Slide Images (WSIs) from Research Infrastructure (RI) data holdings into the DEP. The entire orchestration should be declaratively defined and user-configurable, while abstracting away the complexity of backend transfer and storage operations. The orchestration process should: • Enable RIs to define project-scoped staging rules, specifying which WSIs are to be transferred, using unique identifiers recognised across both the RI and the DEP. • Verify successful and complete transfer of datasets before making them available to the user. • Support the transfer of associated metadata alongside WSIs, and provide functionality to resynchronise metadata	Transferring large-scale, sensitive WSI data from RI holdings into the DEP must be handled through a controlled and reliable process. Allowing RIs to define project-level dataset visibility ensures that governance and data access policies are enforced at the source. Automated transfer of WSIs will be used for the scientific use cases 5 and 6.	Data Orchestration Service (WP2-T2.1)	2nd DEP Release	SUC5, SUC6

	independently, allowing RIs to update or extend metadata (e.g. annotations, genetic data) without requiring the WSIs to be re-transferred. • Track which WSIs have already been transferred to the DEP to avoid redundant transfers when multiple projects request the same images. • Support the definition of a data lifespan, allowing RIs or project owners to specify an expiry period after which the transferred data should be automatically deleted from the DEP. • Provide logging capabilities for all involved stakeholders, ideally accessible via a web interface, to monitor the status of data transfers (e.g., when a transfer was initiated, completed, or failed).				

RSREQ-23	In addition to enabling the transfer of datasets from RIs into the DEP for analysis, the DEP should provide a Results Delivery Mechanism that allows research outputs - such as annotations, processed datasets, and derived results - to be automatically transferred back to the originating RI at a predefined point in time, such as the end of a project. All involved stakeholders should be able to monitor the status of these transfers, e.g. through a web interface. The specific research outputs to be returned should be jointly defined at the beginning of the project or in the usage agreement between the DEP user and the corresponding RI. The Results Delivery Mechanism should then automatically initiate the transfer of outputs at a scheduled date, e.g., project end.	Returning research outputs to the RI ensures that data products generated through DEP-based analysis are preserved, traceable, and available for future reuse. This closes the research data lifecycle and reinforces the role of RIs as long-term stewards of scientific data. It also aligns with the FAIR data principles - ensuring that outputs are Findable, Accessible, Interoperable, and Reusable.	Data Orchestration Service (WP2-T2.1)	1st DEP Release	SUC5, SUC6

RSREQ-91	EISCAT L1 and L2 (raw and spectral) data MUST be made accessible, following access rules as specified in the other SUC4 Requirements. EISCAT is using EGI Checkin and Perun to manage access authN/authZ. The existing download portal (portal.eiscat.se) is not very suitable for API download, so a WebDAV or similar service to be used with Rucio/FTS SHOULD be deployed. Any used tools MUST implement data access authZ through supported protocols. Existing portal retrieves VO group membership information by OIDC to EGI Checkin- this is the preferred protocol. Analysed parameters (Level 3 data) are free for use, though and can preferably be filtered and retrieved using the Madrigal web services and Python client services.	Data accessibility	Data Orchestration Service (WP2-T2.1)	1st DEP Release	SUC4
RSREQ-92	Model outputs must be returned to the S3 buckets of Hypermeteo, so that they can be used to provide climate services.	This functionality is required in order to exploit the data produced by Scientific Use Case	Data Orchestration Service (WP2-T2.1)	2nd DEP Release	SUC1

		1 (downscaling of climate projections).			
RSREQ-33	During the data orchestration from RIs to the DEP, datasets must be prepared in a format that ensures they are ready for use in AI workflows and data analysis. This step acts as a bridge between raw data held at RIs and downstream exploitation in the DEP. For instance, this preparation should: • Perform data conversion, if needed, e.g. convert WSIs to DICOM format. • Perform pseudo-anonymisation, if required by the RI, such as the removal of patient-identifying metadata. Use case providers together with RIs should provide documentation specifying the acceptable data and metadata formats for their datasets, such that they are compatible with the DEP.	This requirement ensures that all data entering the DEP is consistently prepared and standardised, regardless of its origin or modality. Standardisation during data preparation is essential for enabling seamless integration with DEP tools, reproducibility of AI workflows, and interoperability across scientific use cases. It also reduces technical overhead for users and developers by ensuring that data is analysis-ready upon arrival.	Data Preparation Module (WP2-T2.2)	1st DEP Release	SUC5, SUC6, TUC2

WP3 Functional Requirements

Requirement Jira key	Description	Rationale	Component That Fulfils It	Time Frame (Validation)	Validation Use Case
RSREQ-10	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 them in their local machine, cloud provider or a large HPC centre. It contributes directly to KPI#3, KPI#4 and KPI#7.	AI Computing Framework (WP3-T3.1)	1st DEP Release	SUC1, SUC2, SUC5, SUC6
RSREQ-12	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 associated end users.	AI Computing Framework (WP3-T3.1)	2nd DEP Release	SUC5, SUC6
RSREQ-17	The platform must support the development, training, deployment of ML models to learn patterns in data usage and transfer failures and	This functionality is needed for implementing the use case in WP3.3 as well as validating the use case in WP5.2. It contributes to	AI Computing Framework (WP3-T3.1)	1st DEP Release	SUC2

	predict changes/anomalies as required for the use case in T3.3.	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).			
RSREQ-26	The DEP should integrate with the Itwinai workflow orchestration framework to support hyperparameter optimization (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. The HPO framework should allow users to: ● Define a search space for hyperparameters (e.g. learning rate, batch size, ...). ● Choose from standard optimization strategies, such as grid search, random search, and Bayesian optimization.	Hyperparameter optimization is essential for developing AI models. Supporting early stopping helps minimize waste of compute 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.	AI Computing Framework (WP3-T3.1)	2nd DEP Release	SUC1, SUC2, SUC5, SUC6

	• Specify the objective metric to optimize (e.g. validation accuracy, AUC). • Set constraints on resource usage, including maximum number of trials, parallel runs, or GPU-hour budgets. • Enable early stopping of unpromising trials based on intermediate results to save computational resources. Supported strategies should include Median stopping rule, Asynchronous Successive Halving (ASHA), and Hyperband. • Automatically deploy training jobs for each HPO trial to the DEP, without requiring manual submission by the user.				

RSREQ-27	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: ● Hyperparameters, such as learning rate, batch size, and number of epochs. ● Performance metrics, such as accuracy, AUC, loss, or task-specific indicators. ● Artifacts, such as trained model files, logs, and visualisations. ● Environment metadata, including the Git commit hash of the training code and the container image identifier used for the run. ● Dataset references, including internal DEP dataset IDs.	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 critical. This requirement supports model development in Use Cases 5 and 6 and contributes to KPI03: No. of AI frameworks/toolboxes offered within DEPs.	AI Computing Framework (WP3-T3.1)	2nd DEP Release	SUC1, SUC2, SUC5, SUC6

	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 MLflow should be controlled through the user management of the DEP. It should be asserted that users only have access to training logs and metadata of projects they are associated with.				
RSREQ-40	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 (Cinceca), MN5 (Bsc)	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. 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	AI Computing Framework (WP3-T3.1)	1st DEP Release	TUC1

		currently using, Lumi (CSC), Leonardo (Cinceca), MN5 (Bsc)			
RSREQ-96	The platform must support the development, training, 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)	1st DEP Release	SUC1
RSREQ-20	The platform must support the development, training, 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)	1st DEP Release	SUC1

RSREQ-95	The project must provide an ML-based model for enabling prediction and/or detection of 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 logs 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.	This functionality is needed for implementing the use case in WP3.3 as well as validating 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)	1st DEP Release	SUC2
RSREQ-15	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)	AI for Health and Life Sciences (WP3-T3.4)	2nd DEP Release	SUC7

		and KPI#8 (No. of use cases developed for DEP validation).			
RSREQ-16	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 for DEP validation).	AI for Health and Life Sciences (WP3-T3.4)	2nd DEP Release	SUC7
RSREQ-31	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	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	AI for Health and Life Sciences (WP3-T3.4)	2nd DEP Release	SUC5

	infrastructure. The developed models should produce predictions that are interpretable, allowing validation and assessment by pathologists. To ensure scientific relevance and robustness, validation should be performed: • On publicly available survival prediction benchmarks to demonstrate the novelty and competitiveness of the developed algorithms against the current state of the art. • 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.	potential identification of novel biomarkers through model interpretation and validation may result in new scientific insights, contributing to KPI#21: No. of peer-reviewed scientific publications and KPI#22: No. of research outputs.			
RSREQ-32	Algorithms should be developed for generating synthetic histopathological images using	This requirement contributes to use case 6 and supports KPI#4 (No. of AI models offered within	AI for Health and Life Sciences (WP3-T3.4)	2nd DEP Release	SUC6

	generative AI models, such as diffusion-based approaches. These models should be trained on real pathology data available within the Dept. Ensure scientific relevance and output quality, validation of the generated synthetic images should include: • Privacy risk assessment, e.g. membership inference testing, to confirm that synthetic images do not expose identifiable patient information. • A user study with pathologists, who will review synthetic images and assess their realism, diagnostic plausibility, and fitness for research or training.	DEPs), KPI#21 (No. of peer-reviewed scientific publications), and KPI#22 (No. of research outputs).			
RSREQ-87	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	This functionality is essential for Scientific use case 7 (Foundational models for heterogeneous biological image data). It directly contributes to	AI for Health and Life Sciences (WP3-T3.4)	2nd DEP Release	SUC7

	automated data categorisation and similarity search within the RI	KPI#8 (No. of use cases developed for DEP validation).			
RSREQ-13	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.	Fulfils 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)	1st DEP Release	SUC2, SUC5, SUC6
RSREQ-14	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)	1st DEP Release	TUC4

	and resources must be governed by policies managed within this AAI.	and potential KPIs for controlled resource access and compliance.			
RSREQ-28	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.	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)	1st DEP Release	SUC5, SUC6
RSREQ-94	The assistant should enable users to perform basic bioimage analysis tasks - such as segmentation or object	Many users accessing RI imaging data do not have the technical expertise to deploy AI models or	AI Model Hub (WP3-T3.2)	2nd DEP Release	SUC8

	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.	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.			

WP4 Functional Requirements

Requirement Jira key	Description	Rationale	Component That Fulfils It	Time Frame (Validation)	Validation Use Case
RSREQ-105	• Distribute credits to users based on their calculated resource consumption and environmental impact. • Apply predefined rules to assign credits, prioritizing sustainable practices and efficient usage. • Provide a structured process for requesting additional credits (Credit Request > Approval	Distributing credits based on resource consumption and environmental impact, with predefined rules and structured request processes, promotes sustainable practices, ensures fair and transparent allocation, and supports flexible management through manual and automated methods while	Credit Management System (WP4-T4.4)	2nd DEP Release	All

	Workflow > Post approval actions). • Support both manual assignment by authorized managers and automated distribution based on rules or quotas. • Allocation metadata recording(such as validity period and source pool).	tracking metadata for accountability.			
RSREQ-44	• Collect usage metrics for GPU, CPU, storage, network transfer, and AI model resources. • Record usage units for each resource type at a specified granularity level in the time domain. • Support tracking for cross-national and international DEP projects. • Usage is captured on a per-user and per-project basis.	Accurately collecting and recording usage metrics for GPU, CPU, storage, network transfer, and AI model resources is essential for effective resource management, operational transparency, and cost accountability, especially in large-scale, multi-tenant environments. Capturing these metrics at a specified time granularity ensures the ability to analyse usage trends, detect	Credit Management System (WP4-T4.4)	2nd DEP Release	All

		anomalies, and optimize resource allocation. Tracking usage on a per-user and per-project basis supports detailed billing, quota enforcement, and audit trails, which are critical for compliance, especially in regulated or international contexts.			
RSREQ-75	• Capture 'environmental impact units' for each resource type (e.g., energy consumption, carbon footprint). • Integrate with external environmental impact data sources or models to enrich metrics. • Provide standardized environmental impact metrics across all DEP providers.	Measuring environmental impact is critical to encourage sustainable usage and align with global sustainability goals (e.g., EU Green Deal). Standardized metrics ensure consistency across providers, addressing the challenge of varying environmental data formats.	Credit Management System (WP4-T4.4)	2nd DEP Release	All

RSREQ-76	• Convert usage and environmental impact units into a unified credit system. • Define rules for credit allocation based on resource consumption and environmental impact, and policy-based at the organizational level. • Support configurable credit models to encourage environmentally friendly usage and targeted to each individual DEP.	Converting usage and environmental impact into a unified credit system, with rules defined at the organizational level and configurable models, promotes sustainable practices and supports diverse DEP business needs, such as virtual access funds, by aligning incentives with green computing goals.	Credit Management System (WP4-T4.4)	2nd DEP Release	TUC3, TUC4
RSREQ-77	• Enable validation of usage and environmental impact metrics through scientific use cases. • Support iterative feedback loops to refine credit models based on validation results.	Validation ensures the accuracy and reliability of usage and environmental metrics and credits, addressing the challenge of ensuring trust in the system. Iterative feedback supports continuous improvement, aligning with user and provider needs.	Credit Management System (WP4-T4.4)	2nd DEP Release	TUC4

RSREQ-78	Documented feedback shall be provided to refine access policies and virtual access fund allocations.	Feeding experiences into future provisioning plans ensures the DEP ecosystem evolves based on real-world usage and performance data, addressing the challenge of aligning infrastructure and policies with actual user needs and environmental goals.	Credit Management System (WP4-T4.4)	2nd DEP Release	TUC4
RSREQ-47	Enable processing of decentralized identities using protocols such as SIOPv2, OID4VCI, and OID4VP.	Supporting decentralized identity protocols like SIOPv2, OID4VCI, and OID4VP enhances user privacy by enabling self-managed identities, ensures interoperability with global standards, and strengthens security through cryptographic verification, reducing fraud and fostering trust across ecosystems.	Interoperability Module for Data Spaces (WP4-T4.2)	2nd DEP Release	All
RSREQ-48	Extend IAM systems (e.g., Check-in/Keycloak) to support WP5 distributed identity use cases.	Supports pilot use cases in WP5 (use case needs).	Interoperability Module for Data Spaces (WP4-T4.2)	2nd DEP Release	All

RSREQ-49	Implement a decentralized trust infrastructure including trust anchors, accreditation, and trust marks.	Ensures a trusted environment across systems (technical limitation).	Interoperability Module for Data Spaces (WP4-T4.2)	2nd DEP Release	All
RSREQ-50	Ensure GAIA-X compliance for trust relationships and credential issuer verification.	Addresses compliance requirements with GAIA-X (KPI, technical limitation).	Interoperability Module for Data Spaces (WP4-T4.2)	2nd DEP Release	All
RSREQ-53	Enable technical interoperability with initiatives such as EHDS, EOSC, and GAIA-X.	Ensures alignment with the external ecosystem (use case need, KPI).	Interoperability Module for Data Spaces (WP4-T4.2)	2nd DEP Release	All
RSREQ-54	Test and validate interoperability for secure data transfers using Rucio and FTS.	Validates secure data exchange mechanisms (technical limitation).	Interoperability Module for Data Spaces (WP4-T4.2)	2nd DEP Release	SUC5, SUC6
RSREQ-55	Define and enforce access and usage policies using the ODRL standard.	Provides flexible data governance (use case need).	Interoperability Module for Data Spaces (WP4-T4.2)	2nd DEP Release	All
RSREQ-57	Support integration with federated data spaces to test interoperability with external entities.	Enables advanced data sharing scenarios (use case need).	Interoperability Module for Data Spaces (WP4-T4.2)	2nd DEP Release	All
RSREQ-36	Download/access of raw/spectral EISCAT data (Levels 1/2) must follow EISCAT embargo rules, i.e. accessed data can only be processed and data can be shared/distributed only according to embargo rules set by the	Compliance with the EISCAT Data Policy.	Policy-Based Authorization Framework (WP4-T4.1)	1st DEP Release	SUC4

	EISCAT agreements. This applies mainly to measurements newer than 3 years; older archived data will be free for use in the project. (See present EISCAT Portal, which implements Checkin/Perun OIDC client / using OIDC Claims)				
RSREQ-71	Support data access policies based on: • Identity assurance (e.g. verified user identity). • User attributes (e.g. groups & roles, organisation affiliation, nationality). • Data attributes (e.g. sensitivity level, ethics committee approval, data access policy). • Credit capacity (e.g. user quotas/limits).	The requirement for extended data access policies is critical to delivering a secure, compliant, and flexible authorisation system. By enabling policies based on identity assurance, user and data attributes, and credit capacity, the system ensures fine-grained access control, regulatory compliance, and efficient resource management.	Policy-Based Authorization Framework (WP4-T4.1)	2nd DEP Release	All

RSREQ-72	Provide a standardised, expressive policy language that supports the creation of complex access control rules.	The requirement for a standardised, expressive policy language is essential to support the creation of complex access control rules, enabling fine-grained authorisation based on diverse criteria, such as user attributes (e.g. nationality, organisation), data attributes (e.g. sensitivity level, acceptance of ethics committee approval, data access policy), and credit capacity. A standardised language ensures consistency, interoperability, and maintainability, allowing to develop, test, and update policies efficiently.	Policy-Based Authorization Framework (WP4-T4.1)	1st DEP Release	SUC1, SUC2, SUC4, SUC5, SUC6, SUC7, SUC8, TUC2, TUC3, TUC4
RSREQ-73	Provide a policy engine that supports: • Dynamic evaluation of access policies in real-time (or near real-time).	A policy engine with dynamic evaluation and hierarchical management is essential for efficient authorisation. Dynamic evaluation ensures real-time	Policy-Based Authorization Framework (WP4-T4.1)	2nd DEP Release	SUC1, SUC2, SUC4, SUC5, SUC6, SUC7, SUC8, TUC2, TUC3, TUC4

	• Hierarchical management of policies (e.g., parent-child policy structures or precedence rules).	access decisions, while hierarchical management simplifies policy organisation and maintenance.			
RSREQ-74	Enable logging of policy decisions for traceability. Provide auditing capabilities to track access control decisions and policy evaluations. Log collection and querying should be done using an industry-standard solution.	Logging and auditing of policy decisions are essential for ensuring traceability and accountability in the authorisation system. Logging enables to record access control decisions, supporting troubleshooting, security incident analysis, and compliance with regulatory requirements (e.g. GDPR). Auditing capabilities allow tracking of policy evaluations, ensuring transparency and verifiability of access decisions.	Policy-Based Authorization Framework (WP4-T4.1)	2nd DEP Release	SUC1, SUC5, SUC6, SUC7, SUC8
RSREQ-82	Provide a user interface that allows administrators to author, edit, and manage access control policies easily. Such interface would ideally be dynamic enough to support the	An intuitive user interface is valuable for enabling administrators to manage complex access control policies	Policy-Based Authorization Framework (WP4-T4.1)	1st DEP Release	All

	introduction of new policies without the need for excessive new development.	with ease, abstracting the policy language's complexity.			
RSREQ-97	The AI compute framework, provided by T3.1, should be allowed to be deployed with relative ease on at least some of the EuroHPC systems. This has to do with both T3.1 and AAI from WP4.	TBA	Policy-Based Authorization Framework (WP4-T4.1)	2nd DEP Release	TUC1
RSREQ-65	A user interface should be available to get the information from the users, like organisation names and countries.	Collection of data from the users.	Privacy-First Compliance Layer (WP4-T4.3)	2nd DEP Release	SUC5, SUC6
RSREQ-67	A user interface should show the information intended for the users.	Obligation to inform the users.	Privacy-First Compliance Layer (WP4-T4.3)	2nd DEP Release	SUC1, SUC3, SUC4, SUC5, SUC6, SUC8, TUC3, TUC4
RSREQ-68	A user interface should collect consent from the users.	Obligation to get the users' consent.	Privacy-First Compliance Layer (WP4-T4.3)	2nd DEP Release	SUC1, SUC3, SUC4, SUC5, SUC6, SUC8, TUC3, TUC4

Non-Functional Requirements

WP2 Non-Functional Requirements

Requirement Jira key	Description	Rationale	Component That Fulfils It	Time Frame (Validation)	Validation Use Case
RSREQ-41	Provide enough storage space and throughput to serve data for AIFS training in a timely manner	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. Hence, DEP should have enough storage available for the AIFS training datasets and transfer 50-100 TB in a few minutes (100 Gb/s) to at least one of the EuroHPC systems that DestinE is currently using, Lumi (CSC), Leonardo (Cinceca), MN5 (Bsc)	Computing Site Integration Interface (WP2-T2.5)	2nd DEP Release	TUC1

RSREQ-46	The Data Popularity service should provide an easy-to-use interface (e.g., UI Dashboard or API-based) to support the extraction of information about the most downloaded data in an easy and quick manner.	The ESGF (Earth System Grid Federation) Data Statistics service continuously collects information about data usage from the ESGF data nodes. Therefore, providing an easy way for querying and retrieving information from the service is key to efficiently and effectively supporting the DEP data lifecycle management.	Data Discovery & Popularity Service (WP2-T2.3)	2nd DEP Release	SUC1, SUC2

WP3 Non-Functional Requirements

Requirement Jira key	Description	Rationale	Component That Fulfils It	Time Frame (Validation)	Validation Use Case
RSREQ-102	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)	2nd DEP Release	TUC3
RSREQ-11	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	AI Computing Framework (WP3-T3.1)	2nd DEP Release	TUC1

		enable training with large-scale datasets.			
RSREQ-29	Users of the DEP should be able to automatically deploy an instance of the WSI visualization toolkit XOpat. The viewer is deployed in an Apptainer container. In this 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. XOpat enables users to: ● Zoom, pan, and navigate in gigapixel-sized WSIs with minimal latency by streaming	User experience is improved through seamless interaction of WSIs and algorithm results directly in the DEP System. Having an in-platform viewer is essential, as it 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.	AI Computing Framework (WP3-T3.1)	2nd DEP Release	SUC5, SUC6

	only the visible viewport instead of full images. • Create pixel-level annotations on WSIs, which can be used as training data for algorithms. • Visualize algorithm outputs on WSIs, such as heatmaps or segmentations.				
RSREQ-19	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)	2nd DEP Release	SUC2
RSREQ-86	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	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	AI for Health and Life Sciences (WP3-T3.4)	2nd DEP Release	TUC2

	AI training, inference, and processing speed.	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.			

WP4 Non-Functional Requirements

Requirement Jira key	Description	Rationale	Component That Fulfils It	Time Frame(Validation)	Validation Use Case
RSREQ-79	• Handle high volumes of usage data from multiple DEP providers and users. • Support scaling to accommodate growing numbers of cross-national projects.	Scalability is necessary to manage increasing data loads as DEP adoption grows, addressing the technical challenge of processing large-scale, distributed usage metrics without performance degradation.	Credit Management System (WP4-T4.4)	2nd DEP Release	TUC4
RSREQ-80	• Process and record usage and environmental metrics with low latency. • Fast credits report generation for typical queries.	Low latency ensures a responsive system, improving user experience for providers and end users. Rapid report generation enhances functionality and	Credit Management System (WP4-T4.4)	2nd DEP Release	TUC4

		maintains smooth interoperability with other connected modules.			
RSREQ-81	• Implement data redundancy to prevent loss of usage or credit data.	Redundancy protects against data loss, ensuring operational continuity.	Credit Management System (WP4-T4.4)	2nd DEP Release	TUC4
RSREQ-34	Data to be used as input for scheduling decisions. Requires selecting the data needed, identifying their sources, investigating access methods and access policies.	We cannot make decisions without data to base these decisions on.	Interoperability Module for Data Spaces (WP4-T4.2)	2nd DEP Release	SUC3
RSREQ-58	Ensure secure identity handling and data transfer using mechanisms like end-to-end encryption and zero-knowledge proofs.	Protects sensitive data and ensures confidentiality (KPI, compliance needs).	Interoperability Module for Data Spaces (WP4-T4.2)	2nd DEP Release	All
RSREQ-59	The system must scale to handle a growing number of identities, credentials, and interoperability use cases.	Ensures system performance under load (technical limitation).	Interoperability Module for Data Spaces (WP4-T4.2)	2nd DEP Release	All
RSREQ-60	Use standard interfaces and protocols such as OpenID, OAuth 2.0, and W3C DIDs to ensure compatibility.	Promotes long-term maintainability and integration (technical limitation).	Interoperability Module for Data Spaces (WP4-T4.2)	2nd DEP Release	All

RSREQ-62	Ensure a modular system architecture that allows easy updates and integration of new protocols.	Reduces maintenance effort and supports future requirements (technical limitation).	Interoperability Module for Data Spaces (WP4-T4.2)	2nd DEP Release	All
RSREQ-63	All identity and credential transactions must be auditable and logged for traceability.	Ensures accountability and compliance (KPI, compliance needs).	Interoperability Module for Data Spaces (WP4-T4.2)	2nd DEP Release	All
RSREQ-64	Ensure compliance with GDPR regarding data minimization, purpose limitation, and user consent.	Addresses legal and regulatory obligations (compliance needs).	Interoperability Module for Data Spaces (WP4-T4.2)	2nd DEP Release	SUC5, SUC6, SUC7, SUC8, TUC3, TUC4
RSREQ-66	Fulfil the technical and policy requirements defined by EHDS, GAIA-X, and EOSC.	Supports project visibility and external cooperation (KPI, use case needs).	Interoperability Module for Data Spaces (WP4-T4.2)	2nd DEP Release	All
RSREQ-103	For data on the DEP access policies from the RIs should be enforced. For instance, it should be ensured that each transfer is associated with the relevant project context and that only authorised users and processes can initiate or access the transferred data.	Ensuring that data on the DEP can only be accessed by users with the respective rights is necessary to account for policies defined by the RIs.	Policy-Based Authorization Framework (WP4-T4.1)	2nd DEP Release	SUC5, SUC6

RSREQ-43	The data access model for DestinE should be respected; this probably means federation with the DestinE AIM system (customized Keycloak)	DestinE has a strict data access policy that needs to be respected by any component that holds that data and makes it available to third parties. We can also envision a setup where the data is maybe not kept in the DEP, but only the orchestration is handled by the DEP; still, access control should be in place and compliant with DestinE access policies. Hence, DEP should have enough storage available for the AIFS training datasets and transfer 50-100 TB in a few minutes (100 Gb/s) to at least one of the EuroHPC systems that DestinE is currently using, Lumi (CSC), Leonardo (Cinceca), MN5 (Bsc)	Policy-Based Authorization Framework (WP4-T4.1)	1st DEP Release	TUC1
RSREQ-83	The authorisation solution must scale to handle large numbers of users, policies, and access requests efficiently	Scalability is essential to ensure the authorisation solution manages growing users, policies, and requests efficiently. It allows	Policy-Based Authorization Framework (WP4-T4.1)	2nd DEP Release	SUC5, SUC6

		to maintain performance under high loads.			
RSREQ-84	The policy engine should evaluate policies dynamically with low latency to support real-time access control decisions	Low-latency policy evaluation is essential to ensure the policy engine delivers real-time (or near real-time) access control decisions.	Policy-Based Authorization Framework (WP4-T4.1)	2nd DEP Release	SUC5, SUC6
RSREQ-85	The authorisation solution should integrate seamlessly with existing systems and frameworks for identity management, data storage, and access control. It must support interoperability with common protocols such as OpenID Connect, OAuth2, and SAML, as well as frameworks like ODRL and Verifiable Credentials (VCs) to enable fine-grained, standards-based policy enforcement across distributed environments.	Interoperability enables the authorisation solution to integrate with existing identity, storage, and access systems, leveraging current infrastructure and ensuring consistent, scalable policy enforcement.	Policy-Based Authorization Framework (WP4-T4.1)	2nd DEP Release	SUC5, SUC6, TUC1
RSREQ-104	The personal data should be encrypted at rest.	Cybersecurity: data transmission	Privacy-First Compliance Layer (WP4-T4.3)	1st DEP Release	SUC1, SUC3, SUC4, SUC5, SUC6, SUC8, TUC3, TUC4

RSREQ-52	A secure token is transmitted for each request to the IAM system.	Authentication and authorization.	Privacy-First Compliance Layer (WP4-T4.3)	2nd DEP Release	SUC1, SUC3, SUC4, SUC5, SUC6, SUC8, TUC3, TUC4
RSREQ-56	The personal data should be encrypted in transit.	Cybersecurity: data transmission	Privacy-First Compliance Layer (WP4-T4.3)	2nd DEP Release	SUC1, SUC3, SUC4, SUC5, SUC6, SUC8, TUC3, TUC4
RSREQ-69	Each token should be revoked in a proper way.	Authentication and authorization.	Privacy-First Compliance Layer (WP4-T4.3)	2nd DEP Release	SUC1, SUC3, SUC4, SUC5, SUC6, SUC8, TUC3, TUC4
RSREQ-70	The actions related to consent and to token revocation should be incorporated in the logs.	Authentication and authorization.	Privacy-First Compliance Layer (WP4-T4.3)	2nd DEP Release	SUC1, SUC3, SUC4, SUC5, SUC6, SUC8, TUC3, TUC4