RI SCALE

D4.1

Access Management Systems Specification and Roadmap

Status: Final

Dissemination Level: Public

Funded by
the European Union

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

Abstract

Key words	Access Management Architecture, Credit Management System, Data Exploitation Platform

This deliverable articulates the architectural design, functional and non-functional requirements, and implementation roadmap for the Access Management Systems within the Data Exploitation Platform (DEP) ecosystem, as part of the RI-SCALE project. It establishes a robust framework to meet the needs of infrastructure providers, DEP end-users, model developers, and operators by ensuring secure, equitable, and environmentally sustainable access to computational resources. The document details two core modules: the Access Management Architecture and the CRedit Management System (CRMS). The Access Management Architecture integrates an Authorisation Framework using Open Digital Rights Language (ODRL) and Open Policy Agent (OPA) for fine-grained, policy-driven access control, an Interoperability Framework supporting federated and decentralised identity management for seamless cross-domain operations, and a Privacy and Consent Management subsystem to ensure compliance with privacy regulations. The CRMS enables scalable resource and credit management by collecting granular metrics on computational resource usage (e.g., CPU, GPU, storage, network transfers) and environmental impacts (e.g., energy consumption, CO_2 emissions), translating these into credit values using transparent, sustainability-focused policies like green-index discounts, and distributing credits equitably via a centralised registry. This deliverable provides a vital blueprint for achieving RI-SCALE's goals, enabling the DEP to integrate data and computation while addressing the intricate requirements of contemporary research infrastructures.

Revision History			
Version	Date	Description	Author/Reviewer
V 0.1	27/05/2025	First Draft	Nikolaos Triantafyllis (GRNET)
V 0.2	22/07/2025	First Draft Revision	Nikolaos Triantafyllis (GRNET)
V 0.3	28/07/2025	Submitted for Internal Review	Nikolaos Triantafyllis (GRNET)
V 0.3	01/08/2025	Review #1 completed	Bernd Saurugger (TUWien)
V 0.3	04/08/2025	Review #2 completed	Sandro Fiore (UNITN)
V 0.4	06/08/2025	Submitted version for Final Review	Nikolaos Triantafyllis (GRNET)
V 0.5	08/08/2025	Submitted version for TCB approval	Nikolaos Triantafyllis (GRNET)
V 0.5	15/08/2025	Review #3 completed	Ville Tenhunen (EGI)
V 0.5	22/08/2025	Review #4 completed	Gergely Sipos (EGI)
V 0.6	22/08/2025	Submitted version for Quality Check	Nikolaos Triantafyllis (GRNET)
V 1.0	29/08/2025	Submitted version to EC	Matteo Agati (EGI)

Document Description			
D4.1 - Access Management Systems Specification and Roadmap			
Work Package Number 4			
Document Type	Deliverable		
Document Status	Final	Version	1.0
Dissemination Level	Public		
Copyright Status	CC BY This material by the Parties of the RI-SCALE Consortium is licensed under a Creative Commons Attribution 4.0 International License.		
Lead partner	GRNET		
Document Link	https://documents.egi.eu/document/4201		
DOI	https://zenodo.org/records/16994162		
Author(s)	● Nikolaos Triantafyllis (GRNET) ● Nicolas Liampotis (GRNET) ● Konstantinos Georgilakis (GRNET) ● Antreas Kozadinos (GRNET) ● Fotios Basios (GRNET) ● Kostas Koumantaros (GRNET) ● Themis Zamani (GRNET) ● Francesco Giacomini (INFN) ● Federica Agostini (INFN) ● Enrico Vianello (INFN) ● Peter Lenyi (MU) ● Fernando Gonzalez Perez (EGI) ● Valeria Ardizzone (EGI) ● Neeraj Sharma (T-Systems) ● Lena Matsela (T-Systems) ● Tim Wittenberg (T-Systems) ● Cédric Crettaz (AS) ● Adrián Quesada Rodríguez (AS) ● Hakan Bayindir (TUBITAK)		
Reviewers	● Bernd Saurugger (TUWien) ● Sandro Fiore (UNITN) ● Ville Tenhunen (EGI) ● Gergely Sipos (EGI)		
Moderated by:	● Matteo Agati (EGI)		
Approved by:	Technical Coordination Board		

Terminology / Acronyms	
Term/Acronym	Definition
AAI	Authentication and Authorisation Infrastructure (AAI) is a service that enables authenticated and authorised access to resources (see [AARC-G045])
ABAC	Attribute-Based Access Control (ABAC) is an access control model where decisions are based on attributes of the subject, resource, action, and environment. (see also [NIST SP 800-162])
Access Token	A credential represented by a string, issued to a client by an authorisation server, and used to access protected resources. (see also [RFC6749])
AI	Artificial Intelligence
API	Application Programming Interface
AS	Authorisation Server (AS) is the server that issues access tokens to the client after successfully authenticating the resource owner and obtaining authorisation. (see [RFC6749])
BBMRI-ERIC	Biobanking and BioMolecular Resources Research Infrastructure – European Research Infrastructure Consortium. (see also [BBMRI-ERIC])
C4	A hierarchical model for visualising software architecture at four levels: Context, Container, Component, Code. (see also [C4-Model])
CI/CD	A development approach combining Continuous Integration (automated code integration and testing) and Continuous Delivery or Deployment (automated release of software).
Client	An application making protected resource requests on behalf of the resource owner and with its authorisation. The term "client" does not imply any particular implementation characteristics (e.g. whether the application executes on a server, a desktop, or other devices). (see [RFC6749])
CPU	Central Processing Unit
CRMS	CRedit Management System
DEP	Data Exploitation Platform
DID	A Decentralised IDentifier (DID) is a globally unique identifier that enables verifiable, self-sovereign digital identity.
EHDS	European Health Data Space
EOSC	European Open Science Cloud
FAIR	Findable, Accessible, Interoperable, and Reusable. These principles aim to make data easy to locate, access, integrate, and use by both humans and machines.

Gaia-X	Gaia-X is a European initiative that aims to establish a secure, transparent, and federated digital ecosystem based on European values. (see also [GAIA-X])
GDPR	General Data Protection Regulation (GDPR) is a European Union regulation that governs the processing, transfer, and protection of personal data of individuals in the EU and EEA. (see also [GDPR])
GPU	Graphics Processing Unit
GUI	Graphical User Interface
HPC	High-Performance Computing
HTTP	HyperText Transfer Protocol
IAM	Identity and Access Management (IAM) is a framework of policies, processes, and technologies for managing digital identities and controlling access to resources, ensuring that only authorised entities can access specific systems, data, or applications based on their roles and permissions. (see also [NIST-IAM])
JSON	JavaScript Object Notation (JSON) is a lightweight, text-based, language-independent data interchange format. (see also [RFC8259])
JSON-LD	JavaScript Object Notation for Linked Data (JSON-LD) is a lightweight syntax to serialise Linked Data in JSON. (see also [JSON-LD])]
JWKS	JSON Web Key Set (JWKS) is a JSON object that represents a set of public keys, typically used to verify the signatures of issued JWTs. (see also [RFC7517])
JWT	JSON Web Token (JWT) is a compact, URL-safe means of representing claims to be transferred between two parties. (see also [RFC7519])
MFA	Multi-Factor Authentication (MFA) is an authentication system that requires more than one distinct authentication factor for successful authentication. The three authentication factors are something you know, something you have, and something you are. (see also [NIST-MFA])
mTLS	Mutual Transport Layer Security (mTLS) is a mode of TLS in which both client and server authenticate each other using X.509 certificates, typically used for stronger authentication in OAuth 2.0 and service-to-service communication. (see also [RFC8705)
MVP	Minimum Viable Product
ODRL	The Open Digital Rights Language (ODRL) is a policy expression language used to define and manage rights and permissions for digital content and services. (see [ODRL-Model])
OIDC	OpenID Connect (OIDC) is an identity layer built on top of the OAuth 2.0 protocol [RFC6749], enabling clients to verify the identity of an end-user based on authentication performed by an authorisation server and to obtain basic profile information about the user in an interoperable and REST-like manner. (see [OIDC-Core])

OID4VCI	OpenID for Verifiable Credential Issuance (OID4VCI) is a standard that defines an OAuth-protected API for the issuance of verifiable credentials. (see also [OID4VCI])
OID4VP	OpenID for Verifiable Presentations (OID4VP) defines a mechanism on top of OAuth 2.0 that enables the presentation of verifiable credentials as verifiable presentations. (see also [OID4VP])
OPA	Open Policy Agent (OPA) is an open-source, general-purpose policy engine that allows for unified and context-aware policy enforcement across various cloud environments. (see [OPA])
Policy Decision Point - PDP	A component in an access control system that evaluates access requests against security policies and attributes, provided by a Policy Information Point (PIP), to make authorisation decisions (e.g., grant or deny access).
Policy Information Point - PIP	A component in an access control system that provides external information, such as user attributes, environmental data, or resource metadata, to a Policy Decision Point (PDP) to support authorisation decisions.
Rego	Policy language for Open Policy Agent
RESTful	Representational State Transfer
RI	Research Infrastructure
Subject	A person, organisation, device, hardware, network, software, or service. (see also [NIST-SP-800-63-3])
SIOPv2	Self-Issued OpenID Provider v2 (SIOPv2) is an OpenID Connect extension that enables a user to act as their own OpenID Provider to authenticate and present claims directly to Relying Parties. (see also [SIOPv2])
TLS	Transport Layer Security (TLS) is a cryptographic protocol that provides privacy and data integrity between two communicating applications. (see also [RFC8446])
VC	A Verifiable Credential (VC) is a digital credential that is tamper-evident and can be cryptographically verified. (see also [VC-Data-Model])
WP	Work Package
WORM	Write Once Read Many (WORM) is a data storage technology that allows data to be written once and read multiple times, ensuring data immutability and integrity for compliance and archival purposes.

Table of Contents

List of Figures

List of Tables

Executive Summary

This deliverable defines the **Access Management Systems** for the **Data Exploitation Platform** (**DEP**), comprising the **Access Management Architecture** for policy-driven access control, federated identity, and privacy compliance, and the **CRedit Management System** (**CRMS**) for tracking resource usage, environmental impacts, and credit translation and distribution.

This deliverable consolidates the architectural vision, functional and non-functional requirements, and implementation roadmap for the Access Management Systems within the DEP ecosystem. It focuses on addressing the needs of infrastructure providers, DEP end-users, model developers, and operators by defining a robust framework for secure, equitable, and sustainable resource access. The document outlines two core modules, namely, the Access Management Architecture and the CRedit Management System (CRMS).

The Access Management Architecture integrates an Authorisation Framework using Open Digital Rights Language (ODRL) and Open Policy Agent (OPA) for policy-driven access control, an Interoperability Framework for federated and decentralised identity management, and a Privacy and Consent Management subsystem for regulatory compliance. The CRMS tracks computational resource usage (e.g., CPU, storage, network) and environmental impacts (e.g., energy, Carbon Dioxide -CO_2- emissions), translates these into credits via sustainability-focused policies (e.g., green-index discounts), and distributes credits fairly through a centralised registry. The proposed architecture and phased implementation plan provide a clear path forward for adoption, validation, and future enhancements.

1. Introduction

1.1. Scope and Purpose of the Deliverable

The D4.1 deliverable aims to define the architectural design, functional and non-functional requirements, and implementation roadmap for the Access Management Systems within the Data Exploitation Platform (DEP) ecosystem as part of the RI-SCALE project. Its primary purpose is to establish a framework ensuring secure, equitable, and environmentally sustainable access to computational resources for DEP end-users, model developers, and operators. It provides a clear baseline to guide the development, implementation, and validation of these systems, aligning with RI-SCALE's goals of enhancing data access, AI-driven analysis, and resource management across distributed research infrastructures (RIs).

The deliverable focuses on two core modules:

- **Access Management Architecture**: Encompasses the Authorisation Framework (using Open Digital Rights Language (ODRL) and Open Policy Agent (OPA) for policy-driven access control), Interoperability Framework (supporting federated/decentralised identity management with standards like OpenID Connect (OIDC) and Decentralised Identifier (DID)), and Privacy and Consent Management subsystem (ensuring GDPR compliance).
- **CRedit Management System (CRMS)**: Takes into account computational resource usage (e.g., CPU, GPU, storage, network) and environmental impacts (e.g., energy, CO_2 emissions), that are stored in the system, translates these into credits using sustainability-focused policies (e.g., green-index discounts), and manages equitable distribution via a centralised registry.

1.2. Structure of the Deliverable

Section 2 describes the DEP as a solution for enabling secure, AI-driven analysis of distributed research data. It defines roles for end-users, model developers, and operators, outlines their interactions with access management systems, and introduces the CRMS to ensure fair, sustainable, and usage-based access to compute resources.

Section 3 outlines the access management systems within the DEP architecture, highlighting its modular design around data management, scalable AI, and robust access control. Key components include data orchestration, AI computing frameworks, and policy-based authorisation. The Access Management Systems enable federated identities, consent management, and fine-grained access control. Central to this is the CRMS, which tracks resource usage, applies sustainability policies, and allocates credits to ensure fair and efficient access across the DEP ecosystem.

Section 4 details the technical specifications of the Authorisation Framework, Interoperability Framework, Privacy and Consent Management, and the CRMS.

Section 5 outlines a phased deployment strategy for DEP's Access Management System by providing the implementation roadmap, where capabilities are prioritised across project milestones.

2. DEP User Stories Involving Access Management Systems

2.1. DEP Vision

The DEP, as outlined in the RI-SCALE project's Deliverable D5.1[1], is a framework designed to enhance the capabilities of RIs by addressing the challenges of managing, processing, and analysing large-scale, heterogeneous scientific data. It aims to bridge the gap between data generation and computational analysis, particularly for distributed RIs where data is generated across multiple facilities. The DEP facilitates seamless data access, AI-driven processing, and secure, scalable computation to support scientific and technical use cases, ensuring alignment with user needs and compliance requirements. Figure 1 illustrates a scenario where multiple DEPs connect to multiple data holdings, such as in the case of BBMRI-ERIC. This configuration supports RIs with multiple repositories and is backed by multiple compute centres. It is a complex setup, requiring intensive operational effort and sophisticated IT security and privacy configurations.

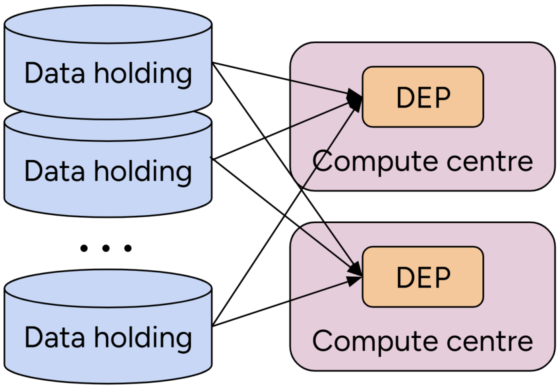

Figure 1: Multiple DEPs Linked to Multiple Data Holdings

The key RI challenges and limitations that DEP seeks to resolve are:

1. **Limited On-Site Compute**: Insufficient compute and storage resources at RI data holdings impede data quality control, FAIR-ification, and widespread data use for analysis;
2. **Large Data Handling**: Downloading large datasets is slow and complex for researchers, with differing access controls between storage and compute systems adding further challenges;
3. **Complex Software Setup**: Configuring data science environments (e.g., AI, Digital Twins, Trusted Research Environments, Secure Processing Environments) poses significant barriers for users.

[1] Psychas, A., Spiliotopoulou, A., Tenhunen, V., & Sipos, G. (2025). RI-SCALE_D5.1 – Data Exploitation Platform Requirements and Design Considerations (V1_Under EC Review). Zenodo. https://doi.org/10.5281/zenodo.15755803

2.2. Major Stakeholders in a DEP

The DEP serves as an extension of an RI, providing a new service linked to RI data holdings to enable online data processing, with a focus on AI-driven analysis. The primary users involved in a DEP include the following groups:

1. **End Users**: The end-users of a DEP serve as the primary beneficiaries. Their key objectives include discovering relevant datasets from an RI, transferring this data from storage to the DEP's compute facility, and selecting pre-configured, pre-trained AI models to perform analysis through inference runs. They execute models on the data and share the resulting outputs with other authorised users, facilitating collaborative research and data-driven insights;
2. **Model Developers**: They create and deploy new AI models within the DEP, either by utilising or advancing off-the-shelf third-party models or developing custom models from scratch. After training these models with RI data, they validate and share them through the DEP, making them accessible to end-users for analysis and inference;
3. **DEP Operators:** They are responsible for deploying, configuring, and maintaining the DEP environment within a compute centre. Their role includes establishing and managing critical connections to external systems, such as data repositories, AI model stores, and identity management systems, as required by the specific DEP implementation.

2.3. Overview of DEP's User Activities

Aligned with the project objectives, the primary goals of the DEP are to: replicate and manage large-scale scientific datasets from RI repositories and Data Spaces onto high-performance and cloud computing resources; facilitate scalable AI-driven data analysis; support real-world scientific use cases involving big data and AI applications; enable seamless user access to resources and services across the entire value chain; monitor and report resource and service consumption throughout the usage workflow; and enhance the AI-based data exploitation and mining capabilities of RIs. The DEP serves three main user roles, as described in Section 2.2.

2.3.1. User Stories for DEP End-Users

End-Users interact with the DEP to discover datasets, select AI models, analyse data, and export results, all within an integrated research environment.

- As a DEP End-User, I want to discover relevant datasets within a research infrastructure or data space so that I can find data suited to my research needs.
- As a DEP End-User, I want to flag specific datasets for analysis within the processing environment.

- As a DEP End-User, I want to discover available pre-configured and pre-trained AI models to find the best fit for my data analysis tasks.
- As a DEP End-User, I want to select from a set of pre-configured and pre-trained AI models to analyse the data efficiently.
- As a DEP End-User, I want to perform data analysis directly on the research infrastructure data to derive meaningful insights.
- As a DEP End-User, I want to export the results of my data analysis so I can use them in reports or further processing.

2.3.2. User Stories for DEP Model Developers

Model developers engage in two primary types of activities: those related to developing new AI models and those involving work with existing models.

2.3.2.1. Activities with New Models

- As a DEP Model Developer, I want to create a new AI model so that I can address novel research challenges.
- As a DEP Model Developer, I want to deploy the new AI model for training to begin the learning process on relevant data.
- As a DEP Model Developer, I want to train the new AI model using research infrastructure (RI) data to optimise its performance.
- As a DEP Model Developer, I want to validate the new model in terms of accuracy and performance to ensure it meets quality standards.
- As a DEP Model Developer, I want to share the validated model across one or multiple DEPs so others can benefit from my work.

2.3.2.2. Activities with Existing Models

- As a DEP Model Developer, I want to select an existing model for retraining to improve or adapt it for new data.
- As a DEP Model Developer, I want to associate the existing model and the training data to enhance its learning.
- As a DEP Model Developer, I want to train an existing or third-party AI model with new data to keep it relevant.
- As a DEP Model Developer, I want to validate the accuracy of an existing model after retraining to confirm improvements.

- As a DEP Model Developer, I want to share the validated, updated model across one or multiple DEPs to distribute its benefits.

2.3.3. User Stories for DEP Operators

DEP operators are responsible for deploying, managing, and maintaining the DEP infrastructure, ensuring seamless integration, high availability, and efficient resource usage.

1. As a DEP operator, I want to deploy, configure, and operate the DEP environment within the compute centre to ensure stable and reliable operation.
2. As a DEP operator, I want to establish and maintain connections between the DEP environment and the external systems (such as AAI, AI model stores, and data holdings) to enable seamless integration.
3. As a DEP operator, I want to monitor and ensure the availability and continuity of the DEP's infrastructure to maintain uninterrupted service.
4. As a DEP operator, I want to manage infrastructure incidents and handle service requests promptly to minimise downtime and resolve issues efficiently.
5. As a DEP operator, I want to oversee infrastructure capacity for DEPs to guarantee that resources meet demand.
6. As a DEP operator, I want to generate reports on DEP resource usage to support operational planning and optimisation.

2.4. Access Management Interactions

2.4.1. Access Management Workflow from a DEP End-User Perspective

Table 1: Storyline of the interaction of a DEP End-User with the Access Management Services

Sequence of Interactions Between a DEP End-User and the Access Management Services		
Steps	**User Action**	**Interaction with the Access Management Services**
1	The DEP End-User discovers relevant datasets from a research environment or data space.	The DEP End-User is authenticated via the DEP's integrated AAI or the RI AAI, depending on the access flow. Access to dataset metadata may require authentication and authorisation checks through the RI's AAI and policy enforcement by the DEP Authorisation Framework.

2	The DEP End-User flags datasets for analysis in the processing environment.	The Authorisation Framework validates whether the user has the appropriate rights to initiate processing on the selected datasets, applying relevant access control policies.
3	The DEP End-User discovers a pre-configured and pre-trained AI model to analyse the data.	Access Management Services validate the DEP End-User's permissions for model discovery, applying model-sharing policies defined in the DEP Authorisation Framework.
4	The DEP End-User chooses a pre-configured and pre-trained AI model to analyse the data.	The DEP Authorisation Framework evaluates policy rules to confirm the user is authorised to bind the selected model with the requested dataset for analysis.
5	The DEP End-User performs data analysis based on the RI data.	At execution time, the DEP End-User's access token is validated and attribute-based policies are evaluated by the Authorisation Framework, potentially retrieving additional context from the RI AAI or integrated Policy Information Points.
6	The DEP End-User retrieves the exported results of the data analysis.	Output access is enforced by the same policy engine. Access Management Services ensure only authorised users can access results; audit logs are recorded for accountability and traceability.

2.4.2. Access Management Workflow from a DEP Model Developer Perspective

2.4.2.1. Creating and Sharing a New AI Model

Table 2: Storyline of a DEP Model Developer's interaction with the Access Management Technologies during the creation of a new AI model

Sequence of Interactions Between a DEP Model Developer and the Access Management Services		
Steps	**User Action**	**Interaction with the Access Management Services**
1	The DEP Model Developer creates a new AI model.	Access Management Services enforces permissions for creating new models through the DEP Policy Engine. The developer must be authenticated and authorised to register new models.

2	The DEP Model Developer deploys a new AI model for training.	The DEP Authorisation Framework checks that the developer is authorised to execute training workflows, evaluating access policies linked to compute resources and input datasets.
3	The DEP Model Developer deploys a new AI model with RI data.	Federated access control policies are evaluated to ensure the developer can access and process RI-held data. The DEP may delegate policy enforcement to the RI AAI or enforce combined RI and DEP policies locally.
4	The DEP Model Developer validates the accuracy of the trained AI model.	Access Management Services verify that the user is permitted to access and evaluate training results.
5	The DEP Model Developer shares the validated model in one or multiple DEP(s).	Sharing actions are authorised based on model ownership and access-sharing policies. The DEP Authorisation Framework evaluates entitlements to ensure only eligible users or other trusted DEP environments gain access.

2.4.2.2. Activities with Existing Models

Table 3: Storyline of a DEP Model Developer's interaction with the Access Management Technologies when using existing AI models

Sequence of Interactions Between a DEP Model Developer and the Access Management Services		
Steps	**User Action**	**Interaction with the Access Management Services**
1	The DEP Model Developer selects an existing model for retraining.	The DEP Authorisation Framework validates whether the developer has access to view and reuse the selected model, based on sharing policies and entitlements.
2	The DEP Model Developer associates the existing model and the training data.	The policy engine evaluates access rules for both the model and the data. Attribute-based access control is applied to determine if the user can combine these assets for processing.
3	The DEP Model Developer trains an existing or third-party AI model with the data.	The Authorisation Framework ensures policy-compliant use of both model and data. Access decisions may rely on token claims and dynamic attributes retrieved via Policy Information Points.
4	The DEP Model Developer validates the accuracy of the trained AI model.	Access policies determine whether the developer can access evaluation outputs.

5	The DEP Model Developer shares the validated model in one or multiple DEP(s).	The DEP Policy Engine enforces policies for model sharing across DEPs, ensuring only authorised distribution and audit logging of the sharing action.

2.4.3. Access Management Workflow from a DEP Operator Perspective

Table 4: Storyline of the interaction of a DEP Operator with the Access Management Services

Sequence of Interactions Between a DEP Operator and the AMS		
Steps	**User Action**	**Interaction with the Access Management Services**
1	The DEP Operator deploys, configures, and operates the DEP environment within the compute centre.	The DEP Authorisation Framework (e.g., Policy Engine, Policy Authoring API, Policy Repository) is deployed and configured. The operator ensures that policy configurations reflect the access control needs of the DEP environment.
2	The DEP Operator ensures the DEP environment's connection to external systems (AAI, AI model stores, data holdings, etc.).	The operator integrates the DEP environment with external AAIs (RI and compute provider) and registers trusted Policy Information Points.
3	The DEP Operator reports DEP's resource usage.	Not directly handled by the Authorisation Framework; resource usage reporting is addressed by the Credit Management System (see Sections 2.5 & 4.4).
4	The DEP Operator ensures DEP's infrastructure availability and continuity.	The Authorisation Framework is monitored to ensure continued access control functionality; the operator ensures authorisation services remain responsive as part of broader infrastructure availability.
5	The DEP Operator manages DEP's infrastructure incidents and service requests.	Incident response involves reviewing or adjusting policy configurations and examining audit logs of policy decisions; all changes are recorded by the Authorisation Framework.
6	The DEP Operator Ensures infrastructure capacity for DEPs.	Access policies enforced by the Authorisation Framework reflect project quotas or resource constraints informed by the Credit Management System; the operator may update policies accordingly.

2.5. Technological Use Case: Credit Management System

DEPs hosted on compute centres enable RIs to deliver scalable data analytics and AI-driven insights to a diverse user base, including scientists from academia and industry. Ensuring equitable access to HPC, cloud, storage, and AI resources while promoting environmentally sustainable usage across varied projects presents a significant challenge. To address this, a robust CRMS is essential to monitor resource consumption and environmental impact, translating these metrics into credits that support innovative DEP business models, such as virtual access funds. The CRMS aims to facilitate fair, transparent, and usage-based credit allocation, consumption, and enforcement across distributed research infrastructure services, ensuring equitable access to shared digital resources (e.g., compute, storage, and data services) for users, projects, and organisations while aligning with sustainability and operational goals.

The following user stories comprehensively outline the workflow for DEP End Users, Model Developers, and Operators as they interact with the CRMS within the DEP environment. These stories detail the sequence of actions and corresponding CRMS interactions, illustrating how the system should support resource allocation, credit management, and policy enforcement through its RESTful API, ensuring transparent, equitable, and sustainable access to computational resources across diverse research infrastructure services.

2.5.1. CRMS Workflow from a DEP End-User Perspective

The CRMS aims to facilitate DEP End-User interactions. It should be able to register users, assign initial credits, reserve credits for datasets and AI models usage, apply green-index discounts post-analysis, and finalise credit consumption asynchronously, ensuring transparent and sustainable resource access. Table 5 provides a detailed, step-by-step storyline of these interactions, illustrating how DEP End-Users engage with the CRMS at each phase of their workflow. It outlines user actions alongside the corresponding CRMS processes.

Table 5: Storyline of the interaction of a DEP End-User with the CRMS

Sequence of Interactions Between a DEP End-User and the CRMS		
Steps	**User Action**	**Interaction with the Credit Management System**
1	The DEP End-User discovers relevant datasets from a research environment or data space.	Upon the user's initial interaction with the DEP, after successful authorisation through the DEP's Authorisation Framework, the CRMS automatically registers them into the system, creating a user profile and assigning an initial fixed number of credits based on predefined credit distribution rules configured by a DEP Operator.

2	The DEP End-User flags datasets for analysis in the processing environment.	The CRMS is queried to translate the resource requirements of the flagged datasets into credit values, using predefined rules that translate resource requests into credits based on the compute centre’s policy. Credits are then reserved after verifying that the user’s available credit balance is sufficient to cover the estimated requested credits.
3-4	The DEP End-User discovers and chooses a pre-configured and pre-trained AI model to analyse the data.	The compute centre interacts with the CRMS components to verify that the user’s account has sufficient credit balance to cover the computational resources required for the selected AI model.
5	The DEP End-User performs data analysis based on the RI data.	When data analysis is finished, the CRMS applies a sustainability-focused green-index policy to evaluate resource usage, calculating and applying discount credits for environmentally efficient operations, which are then credited back to the user’s account, promoting sustainable resource consumption.
6	The DEP End-User retrieves the exported results of the data analysis.	At the next iteration, as part of the asynchronous resource usage harvesting process, the CRMS finalises the consumption of reserved credits based on actual resource usage, applies any credit consumption imbalance, and updates the user’s credit balance to accurately reflect the completed analysis and export activities.

2.5.2. CRMS Workflow from a DEP Model Developer Perspective

The CRMS aims to support DEP Model Developers in creating, training, validating, and sharing new or pre-existing AI models. For both new and existing models, the CRMS should automate user registration, assign initial credits, reserve credits for model deployment and training based on the compute centres’ policies, apply sustainability-focused policies like green-index discounts, and balance credit consumption asynchronously, ensuring transparent, efficient, and equitable resource management aligned with institutional and environmental goals. Tables 6 and 7 summarise the sequence of interactions between Model Developers and the CRMS, highlighting how credit-based operations are managed in alignment with institutional policies and compute centre configurations.

2.5.2.1. Creating and Sharing a New AI Model

Table 6: Storyline of the interaction of a DEP Model Developer with the CRMS.

Sequence of Interactions Between a DEP Model Developer and the CRMS		
Steps	**User Action**	**Interaction with the Credit Management System**
1	The DEP Model Developer creates a new AI model.	Upon the user's initial interaction with the DEP, after successful authorisation, the CRMS automatically registers them into the system, creating a user profile and assigning an initial fixed number of credits based on predefined credit distribution rules configured by a DEP Operator.
2-3	The DEP Model Developer deploys and trains a new AI model with RI data.	The CRMS interfaces with its components to verify and reserve credits for deploying the new model or associating the existing model with training data, using predefined rules for resource to credit translation based on the compute centre's policy.
4-5	The DEP Model Developer validates the accuracy of the trained AI model. Then, the DEP Model Developer shares the validated model in one or multiple DEP(s).	The CRMS tracks additional resource usage for validation tasks, reserves credits for computational resources that were used, and applies any applicable sustainability-focused policies, such as green-index discounts, to adjust credit consumption based on efficient resource usage. At the next iteration, as part of the asynchronous resource usage harvesting process, the CRMS finalises the consumption of reserved credits based on actual resource usage, applies any credit consumption imbalance, and updates the user's credit balance to accurately reflect the completed analysis and export activities.

2.5.2.2. Activities with Existing Models

Table 7: Storyline of the interaction of a DEP model developer with the CRMS

Sequence of Interactions Between a DEP Model Developer and the CRMS		
Steps	**User Action**	**Interaction with the Credit Management System**
1	The DEP Model Developer selects an existing model for retraining.	Upon the user's initial interaction with the DEP, after successful authorisation, the CRMS automatically registers them into the system, creating a user profile and assigning an initial fixed number of credits based on predefined credit distribution rules configured by a DEP Operator.

2-3	The DEP Model Developer associates the existing model and the training data, and trains an existing or 3rd party AI model with the data.	The CRMS interfaces with its components to verify and reserve credits for deploying the new model or associating the existing model with training data, using predefined rules for resource to credit translation based on the compute centre's policy.
4-5	The DEP Model Developer validates the accuracy of the trained AI model. Then, the DEP Model Developer shares the validated model in one or multiple DEP(s).	The CRMS tracks additional resource usage for validation tasks, reserves credits for computational resources that were used, and applies any applicable sustainability-focused policies, such as green-index discounts, to adjust credit consumption based on efficient resource usage. At the next iteration, as part of the asynchronous resource usage harvesting process, the CRMS finalises the consumption of reserved credits based on actual resource usage, applies any credit consumption imbalance, and updates the user's credit balance to accurately reflect the completed analysis and export activities.

2.5.3. CRMS Workflow from a DEP Operator Perspective

This section outlines the typical sequence of interactions between a DEP Operator and the CRMS as part of managing the DEP environment within a compute centre. It presents a step-by-step view of the Operator's responsibilities, from configuring service tiers and defining usage metrics to reporting resource consumption and managing capacity constraints. Table 8 summarises this workflow, highlighting how the CRMS supports operational continuity, usage accounting, and credit-based resource access across the DEP ecosystem.

Table 8: Storyline of the interaction of a DEP Operator with the CRMS.

Sequence of Interactions Between a DEP Operator and the CRMS		
Steps	**User Action**	**Interaction with the Credit Management System**
1-2	The DEP Operator deploys, configures, and operates the DEP environment within the compute centre. Then, the DEP Operator ensures the DEP environment's connection to external systems (AAI, AI model stores, data holdings, etc.).	The DEP Operator interacts with the CRMS to bootstrap the Tier System of the Services provided from the Compute Centre to the DEP environment. Then, the DEP Operator sets the capacity for each service and the unit cost of the associated metric types that the DEP users will be charged for (e.g., core hours).
3-4	The DEP Operator reports DEP's resource usage, and they ensure DEP's	The CRMS collects and processes resource usage data (e.g., CPU, GPU, storage, network transfers) and environmental impact metrics (e.g., kWh consumed, CO_2

	infrastructure availability and continuity.	emissions) to reflect accurate resource consumption for reporting purposes. An empty report could be an indicator of service unavailability.
5	The DEP Operator manages DEP's infrastructure incidents and service requests.	A new service is introduced to the DEP environment; the DEP Operator inputs the necessary information into the CRMS, including the metrics to be collected, their capacity, and the unit cost.
6	The DEP Operator ensures infrastructure capacity for DEPs.	A DEP End-User attempts to spend available credits to request resources, but the DEP Operator is notified by the CRMS that the capacity for the requested service has been fully utilised. As a result, the DEP End-User is unable to run their application.

3. Access Management Systems within the DEP Architecture

The DEP is a comprehensive platform that integrates RI data with advanced computational and AI capabilities. Its architecture is built around three core components - data management, AI processing, and access control - with a strong emphasis on scalability, security, and user-driven customisation. These foundational elements are implemented through the functional and non-functional requirements developed across Work Packages 2, 3, and 4.

- The Data Lifecycle Management (WP2) component encompasses several critical services for efficient data handling. The Data Orchestration Service manages secure data accessibility and transfer from research repositories to the DEP, utilising specialised tools while maintaining access control and provenance tracking. The Data Holdings Integration Framework enables high-speed data transfers to high-performance computing systems and supports optimisation services for caching and ingestion pipelines. Additionally, the Computing Site Integration Interface facilitates integration with compute resources, supporting various workflow execution methods across computing environments.
- The Scalable AI Solutions (WP3) component delivers advanced computational capabilities for AI-driven research. The AI Computing Framework enables large-scale model training and inference across diverse computing infrastructures, incorporating specialised tools for experiment tracking and performance monitoring. The AI for Health and Life Sciences module offers machine learning solutions for scientific analysis, supporting various research applications through tailored model architectures.
- The Access Management Technologies (WP4) component provides robust security and governance capabilities. The Policy-Based Authorisation Framework implements granular access control through standardised policy definitions, enabling sophisticated authorisation scenarios. The Interoperability Module for Data Spaces facilitates seamless integration with federated data ecosystems while maintaining secure identity management protocols. Additionally, the CRedit Management System (CRMS) governs resource allocation through a sustainable credit-based model that tracks usage patterns and environmental considerations.

In the architecture, as depicted in Figure 2, data replicated from RI repositories are processed in compute environments, where AI models and workflows interact with data lifecycle services under policy-driven access control. Core components - Authorisation Framework, CRMS, and federated AAI proxies (RI and Compute) - enable secure, auditable, and sustainable analytics across distributed infrastructures.

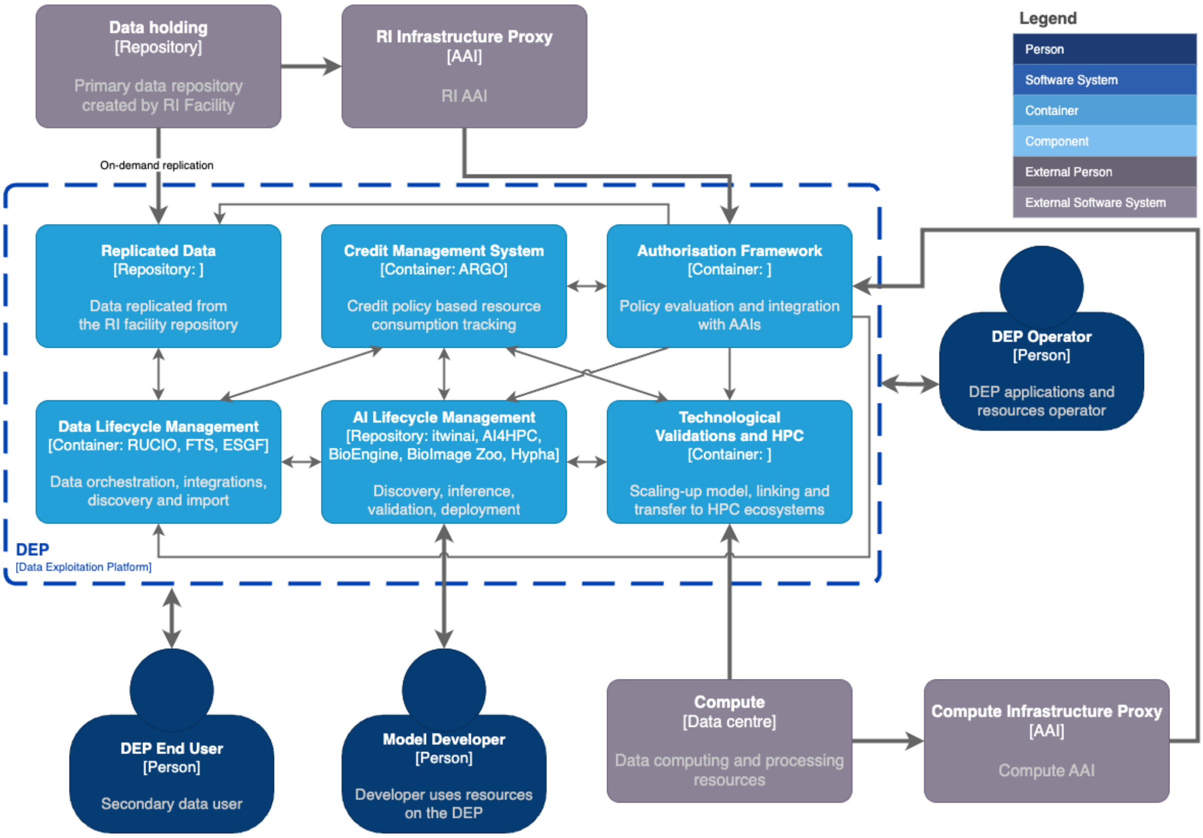
Data holding
[Repository]
Primary data repository created by RI Facility
RI Infrastructure Proxy
[AAI]
RI AAI
On-demand replication
Legend
Person
Software System
Container
Component
External Person
External Software System
Replicated Data
[Repository:]
Data replicated from the RI facility repository
Credit Management System
[Container: ARGO]
Credit policy based resource consumption tracking
Authorisation Framework
[Container:]
Policy evaluation and integration with AAIs
DEP Operator
[Person]
DEP applications and resources operator
Data Lifecycle Management
[Container: RUCIO, FTS, ESGF]
Data orchestration, integrations, discovery and import
AI Lifecycle Management
[Repository: itwinai, AI4HPC, BioEngine, BioImage Zoo, Hypha]
Discovery, inference, validation, deployment
Technological Validations and HPC
[Container:]
Scaling-up model, linking and transfer to HPC ecosystems
DEP
[Data Exploitation Platform]
DEP End User
[Person]
Secondary data user
Model Developer
[Person]
Developer uses resources on the DEP
Compute
[Data centre]
Data computing and processing resources
Compute Infrastructure Proxy
[AAI]
Compute AAI

Figure 2: High-level DEP Architecture extended with Access Management subsystems

3.1. Access Management Architecture

The DEP, illustrated in Figure 2, adopts a modular Access Management Architecture that integrates several distinct components:

- The **Authorisation Framework** (Section 4.1), responsible for defining and enforcing access policies using ODRL, is evaluated at runtime via the Open Policy Agent (OPA).
- The **Interoperability Framework** (Section 4.2) enables federated and decentralised identity integration, supporting both institutional and self-managed credentials.
- The **Privacy and Consent Management** subsystem (Section 4.3) governs user consent and token lifecycle management in line with privacy regulations.

Together, these components support multiple identity flows (federated and decentralised), fine-grained and auditable access control, consent-based credential usage, and cross-domain interoperability. Where relevant, access decisions may also take into account project-level constraints and resource usage information informed by the CRMS.

3.2. Credit Management System

The 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, carbon emissions).
- Translate these metrics into predefined credit values using transparent, policy-driven rules that incorporate sustainability factors like efficiency-based discounts or capacity thresholds.
- Ensure fair and configurable distribution of credits to users and projects, aligning with governance policies.
- Serve as the central database, maintaining a detailed record of credit ownership, consumption, and related information.
- Facilitate secure, standardised communication with external systems and clients, enabling seamless interaction and data exchange.

This architecture aims to ensure traceability, scalability, and compliance with sustainability and operational goals, laying the groundwork for an effective Minimum Viable Product (MVP) in the DEP ecosystem.

4. Technical Specifications

This chapter provides detailed technical specifications for the Access Management System developed under Work Package 4 (WP4) of the RI-SCALE project. The specifications cover four subsystems: Authorisation Framework (Task 4.1), Interoperability Framework (Task 4.2), Privacy and Consent Management (Task 4.3), and Credit Management System (Task 4.4). Each subsection addresses functional and non-functional requirements, ensuring a secure, interoperable, privacy-preserving identity and access management for DEPs.

4.1. Authorisation Framework

The Authorisation Framework, depicted in Figure 3, implements advanced policy-based access control, supporting fine-grained authorisation, near real-time policy evaluation, and interoperability with external systems.

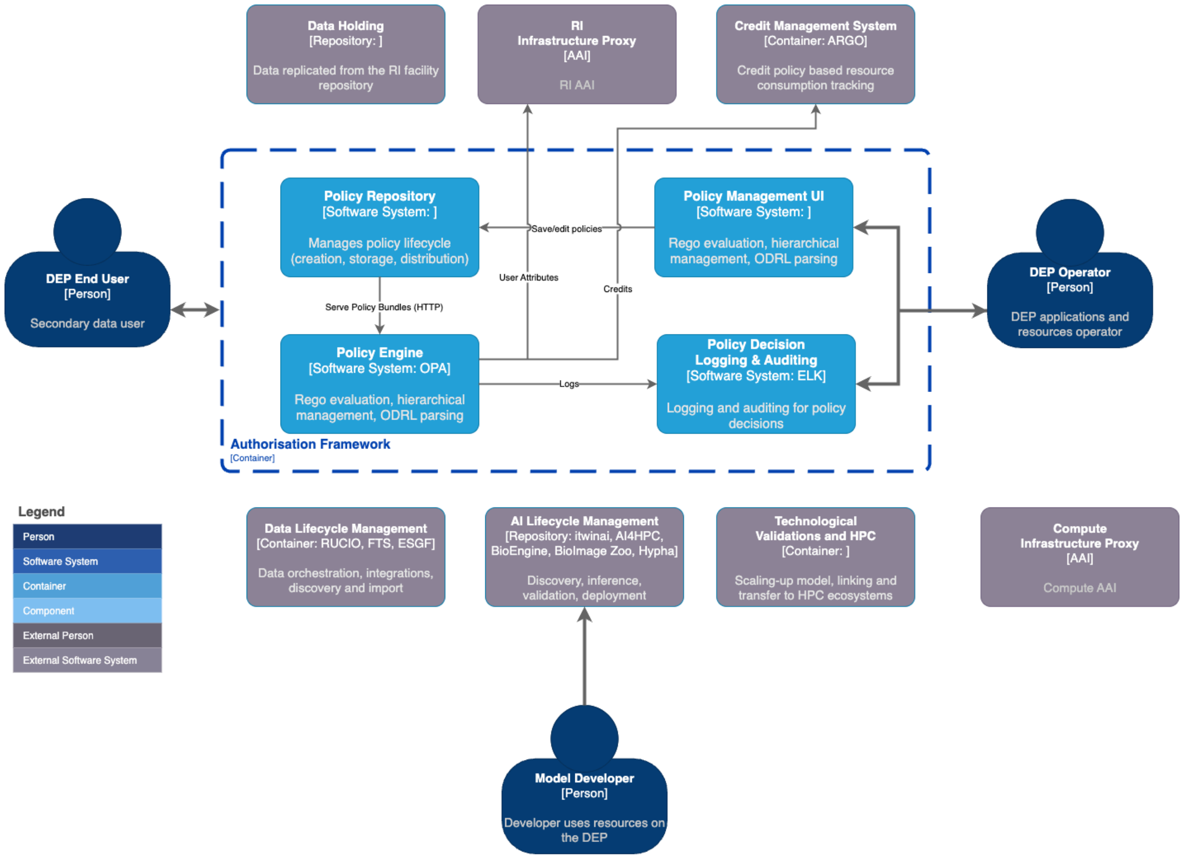
Data Holding
[Repository:]
Data replicated from the RI facility repository
RI
Infrastructure Proxy
[AAI]
RI AAI
Credit Management System
[Container: ARGO]
Credit policy based resource consumption tracking
Policy Repository
[Software System:]
Manages policy lifecycle (creation, storage, distribution)
Save/edit policies
User Attributes
Credits
Policy Management UI
[Software System:]
Rego evaluation, hierarchical management, ODRL parsing
DEP End User
[Person]
Secondary data user
Serve Policy Bundles (HTTP)
DEP Operator
[Person]
DEP applications and resources operator
Policy Engine
[Software System: OPA]
Rego evaluation, hierarchical management, ODRL parsing
Logs
Policy Decision
Logging & Auditing
[Software System: ELK]
Logging and auditing for policy decisions
Authorisation Framework
[Container]
Legend
Person
Software System
Container
Component
External Person
External Software System
Data Lifecycle Management
[Container: RUCIO, FTS, ESGF]
Data orchestration, integrations, discovery and import
AI Lifecycle Management
[Repository: itwinai, AI4HPC, BioEngine, BioImage Zoo, Hypha]
Discovery, inference, validation, deployment
Technological
Validations and HPC
[Container:]
Scaling-up model, linking and transfer to HPC ecosystems
Compute
Infrastructure Proxy
[AAI]
Compute AAI
Model Developer
[Person]
Developer uses resources on the DEP

Figure 3: High-level Authorisation Framework Architecture

The framework aligns with the Access control (Trust and Identity management) functional pillar of the overall DEP architecture described in Deliverable D5.1 (Section 2.6). This framework incorporates the Policy Repository, Policy Management UI, Policy Engine, and Policy Decision Logging to support policy authoring, evaluation, and auditing. Policies are expressed in ODRL and draw on user attributes, credits, and contextual data from the RI and the Credit Management System. This architecture enforces access control across DEP services, including Data Lifecycle Management, AI Lifecycle Management, and HPC resources.

4.1.1. Functional Specifications

The following subsections describe the functional specifications of the Authorisation Framework.

4.1.1.1. Policy Language and Engine Design

The framework uses a standardised policy language and a robust policy engine to enable complex access control rules, dynamic evaluation, and hierarchical policy management, addressing requirements RSREQ-71, RSREQ-72, and RSREQ-73.

4.1.1.1.1. Policy Language Specifications

The Authorisation Framework implements a policy-based access control model that separates policy authoring from enforcement. Policies are written in a high-level, interoperable language, ODRL and evaluated at runtime by a scalable and extensible policy engine, OPA, using Rego. This architecture enables access decision-making that can incorporate dynamic attributes from the Identity and Access Management (IAM) and external data sources.

Standardised Language

The framework adopts ODRL as the canonical format for authoring and storing access control policies. ODRL supports a machine-readable JSON-LD format and provides the expressive capacity to model permissions, prohibitions, and obligations. Defining a custom ODRL application profile may be required to support DEPs.

Expressive Capabilities

ODRL is used to define fine-grained rules based on:

- User Attributes, such as Identity Assurance, Groups and Role entitlements (e.g., expressed according to AARC-G069), Organisation Affiliation, and Nationality.
- Authentication context, such as whether multi-factor authentication (MFA) requirements are satisfied (e.g., based on the REFEDS MFA Profile).
- Resource metadata, such as data sensitivity levels (e.g, public, restricted), ethics committee approvals, or data access policies.
- Usage conditions, such as time restrictions or intended use.
- Credit consumption thresholds (linked to the Credit Management System described in Section 4.4).
- Resource capabilities, as an alternative to identity-based attributes, will also be considered. These include delegated or scoped permissions directly embedded in the access token.

Syntax and Structure

Policies are structured as JSON-LD documents using ODRL terms, including:

- Permission, prohibition, and duty for access control rules,
- target to reference protected resources,
- assignee to identify subjects or groups (directly or via attributes like entitlements),
- constraint to apply conditional logic (e.g., purposes, location, credit quotas).

Execution Model

Although policies are authored in ODRL, runtime evaluation is performed by Open Policy Agent (OPA) using the Rego policy language. The system supports two integration approaches: either by using ODRL policies directly as structured input into Rego rules, or by translating ODRL to Rego syntax before deployment. The choice between the two approaches remains flexible, allowing for future evaluation of trade-offs in terms of maintainability, performance, and expressiveness.

Interoperability

The use of ODRL ensures compatibility with external systems and ecosystems that rely on standardised vocabulary, including Gaia-X, EHDS, and Verifiable Credentials (VCs). Policies can be exported, versioned, and validated independently of their enforcement engine.

Example ODRL Policy

```
{
 "@context": "http://www.w3.org/ns/odrl.jsonld",
 "uid": "http://example.org/policy:project-x-mfa",
 "type": "Set",
 "permission": [
  {
   "target": "https://data.deps.eu/dataset/abc123",
   "assignee": "urn:example:aai.example.org:group:project-x:role=member",
   "action": "read",
   "constraint": [
    {
     "leftOperand": "acr",
     "operator": "eq",
     "rightOperand": "https://refeds.org/profile/mfa"
    }
   ]
  }
 ]
}
```

4.1.1.1.2. Policy Engine Specifications

Runtime Evaluation

The policy engine is based on OPA, a general-purpose decision engine that uses the declarative Rego language to evaluate access policies. At runtime, OPA receives an access request including subject, resource, action, and environment, and returns a permit or deny decision based on loaded policies and contextual data.

Dynamic Attribute Evaluation

OPA evaluates access decisions based on claims included in the access request. These typically originate from tokens issued by the IAM (e.g., OAuth 2.0 access tokens) and may include group entitlements, identity assurance, and authentication context. The responsibility for validating the token, whether through signature verification or OAuth 2.0 Token Introspection (RFC 7662, see Section 4.1.4), usually lies with the resource or enforcement component interfacing with OPA. However, in some cases, OPA may be configured to retrieve additional attributes from Policy Information Points (PIPs), such as credit balances or externally derived claims, when required for policy evaluation.

Policies are authored in ODRL, as outlined in Section 4.1.1.1. At runtime, evaluation by the Policy Engine (OPA) can follow one of two implementation strategies:

1. **Translation-based execution** – ODRL policies are translated into Rego rules before deployment. This simplifies evaluation but requires a translation mechanism to preserve the original policy semantics.
2. **Data-driven evaluation** – The ODRL policy is passed as structured JSON-LD input to the Policy Engine. Rego rules then dynamically interpret the ODRL structure to determine authorisation outcomes.

WP4 will explore both strategies during implementation, maintaining ODRL as the canonical authoring format while allowing flexibility in how policies are enforced at runtime.

The following examples demonstrate the two evaluation strategies described above, using logic equivalent to the example ODRL policy presented in Section 4.1.1.1.

A. Translated Rego (example output of a policy translation layer)

```
package dep.authz

default allow = false

allow {
 input.token.entitlements[_] == "urn:example:aai.example.org:group:project-x:role=member"
  input.resource.id == "https://data.deps.eu/dataset/abc123"
  input.action == "read"
  input.token.acr == "https://refeds.org/profile/mfa"
}
```

This Rego policy reflects a direct translation from an ODRL policy to native Rego syntax.

B. Evaluation of ODRL as structured input

```
package dep.authz

default allow := false

allow if {
  input.action == data.action
  input.resource.id == data.target
  some constraint in data.constraint
  input.token.acr == constraint.acr
  some entitlement in input.token.entitlements
  entitlement == data.assignee
}
```

In this case, the ODRL policy remains in JSON-LD form and is evaluated dynamically using Rego rules that navigate the policy's structure (passed as data).

Hierarchical Management

To support multi-layered governance models, policies may be organised hierarchically:

- DEP-level global rules,
- RI-specific overrides,
- Dataset-level constraints.

Integration

OPA instances can be embedded within the IAM (for centralised enforcement) or queried by protected services directly (for decentralised enforcement). This supports:

- OPA-hidden mode, where services rely on the IAM to enforce authorisation,
- OPA-exposed mode, where services directly call OPA for fine-grained decisions.

Both modes may be used depending on the architectural and trust boundaries of the DEP[2].

[2] INFN has operational experience with the OPA-hidden mode in INDIGO-IAM. Exposed mode should also be supported and can be secured via TLS (see OPA documentation).

4.1.1.2. User Interface for Policy Authoring

An intuitive web-based user interface (UI) allows administrators to author, edit, and manage access control policies, abstracting the complexity of the policy language. While the UI improves usability, it is not a mandatory requirement (RSREQ-82). DEP deployments may rely exclusively on programmatic API-based policy provisioning.

4.1.2.1. UI Specifications

Interface Design:

- The UI should follow a modular design, providing views for policy creation, editing, validation, and history.
- Built using a responsive framework compatible with desktop and mobile environments.

Dynamic Policy Management:

- Enables administrators to create and edit policies using modular templates without needing to manually write ODRL.
- Provides bulk import/export capabilities for policies in JSON ODRL format.
- Provides versioning, rollback functionality, and tracking changes.
- Validation rules should detect malformed logic and conflicting constraints before deployment.

User Experience:

- The interface offers inline guidance (tooltips, documentation links) to help administrators understand policy constructs.
- A visual policy tree allows administrators to navigate hierarchical structures.
- Highlights potential conflicts using pre-deployment checks.

Interoperability:

- Supports policy export and import in standardised ODRL JSON format.
- Pulls real-time user and authentication attributes from integrated IAM systems.

Implementation:

- UI is implemented as a standalone web application integrated with the policy backend via secure REST APIs.
- Administrative actions require user authentication and authorisation via OIDC to ensure secure and controlled access based on group memberships.
- Authored policies are expressed in ODRL. At runtime, they are either:
    - Parsed and evaluated directly as structured input by Rego rules.

        - In this case, the OPA REST API is used for policy evaluation via decision queries (/v1/data/...).
    - Translated to Rego syntax prior to deployment to OPA.
        - In this case, the OPA REST API is used to manage policies (/v1/policies/...), and access can be secured using JWT-based authentication, with access rights derived from IAM-issued group claims and enforced via Rego rules.

4.1.1.3. Logging and Auditing Mechanisms

The framework implements logging and auditing to ensure traceability and compliance with regulatory requirements (e.g., GDPR), as per RSREQ-74. Audit logs include:

- Access request events (timestamp, subject ID, resource ID, decision outcome, policy ID).
- Administrative actions (policy creation/modification/deletion, login attempts).
- Integration events (attribute retrieval, token validation errors).

Logs are structured (JSON) and compatible with external log management systems (e.g., Elastic Stack).

The primary scope of the logging and auditing mechanism is within the boundaries of the Policy Evaluation Engine. It focuses on recording events related to access decisions, policy processing, and interactions with Policy Information Points (PIPs) performed by the engine.

4.1.1.4. IAM Integration and Token Handling

The framework integrates with IAM systems using OIDC for user authentication and attribute retrieval, supporting OAuth 2.0 Token introspection and offline Access Token Validation.

In addition to parsing identity information from tokens, the framework can retrieve additional contextual or user attributes from Policy Information Points (PIPs). This allows external sources, such as the Credit Management System, to influence authorisation decisions dynamically.

OPA can be deployed in two modes:

- Proxy-integrated (hidden from services): In this model, the AAI proxy invokes the policy engine (OPA) before issuing tokens or forwarding requests. End services do not need to be aware of OPA.
- Externally integrated (OPA exposed to services): In this model, services directly query OPA or a PDP for real-time decisions (e.g., for resource-specific authorisation).

4.1.1.4.1. OAuth 2.0 Token Introspection

Tokens issued by the IAM Authorisation Server can be introspected by the policy engine to verify their status and retrieve claims.

Example OAuth 2.0 Token Introspection (RFC 7662) result (partial)

```
{
  "active": true,
  "iss": "https://aai.example.org",
  "iat": 1756665180,
  "exp": 1756668780,
  "sub": "user-123@aai.example.org",
  "entitlements": [
    "urn:example:aai.example.org:group:project-x:role=member"
  ],
  "acr": "https://refeds.org/profile/mfa"
}
```

4.1.1.4.2. Validation of OAuth 2.0 Access Tokens with Embedded Claims

For self-contained JWT tokens, access policies can be evaluated based on embedded claims. The system supports:

- Signature verification using known keys (JWKS).
- Claim extraction for attributes such as groups and roles, identity assurance, and authentication context.

4.1.1.4.3. Policy Information Point (PIP) Integration

The framework supports pluggable PIPs that supply additional attributes at policy evaluation time. Each PIP defines:

- A supported schema (e.g., credit_balance)
- Query method (e.g, HTTPS REST)
- Caching and timeout rules
- An optional trust model or authentication mechanism (e.g., client credentials, mTLS)

Example policy using PIP-enriched attributes

```
package dep.authz
allow {
  input.token.scope[_] == "compute"
  data.pip.credit_balance[input.token.sub] > 100
}
```

4.1.2. Non-Functional Specifications

The Authorisation Framework is designed to meet the operational requirements of modern, scalable, and secure infrastructures. Non-functional characteristics are described qualitatively to accommodate diverse deployment environments and evolving performance expectations.

- **Scalability:** The system shall support horizontal scaling to accommodate increasing numbers of users, policy rules, and concurrent access evaluations across multiple services or tenants (see RSREQ-83).
- **Performance:** The framework shall provide near real-time policy decisions under typical operational load conditions to support interactive and automated use cases (see RSREQ-84).
- **Availability:** The framework shall be suitable for integration in high-availability environments. Deployments should ensure continuity of service during routine operations and maintenance.
- **Observability:** The system shall expose operational and security metrics (e.g., policy evaluation rates, error conditions, attribute resolution activity) to support integration with common monitoring solutions.
- **Security and Isolation:** Policies shall be evaluated in a controlled environment that preserves policy integrity and prevents data leakage between evaluation contexts.

4.2. Interoperability Framework

This section presents the task T4.2 Interoperability Framework for Federated Identity Access Management that supports decentralised identity solutions. The objective of this task is to provide a solution that should work seamlessly across different platforms and comply with European data management guidelines, such as Gaia-X. As illustrated in Figure 4, the Interoperability Framework should enable verifiable identification, dynamic trust models, and federated authentication processes, while maintaining privacy and scalability. It outlines user roles, identity management, trust infrastructure, secure data access, and system operations in alignment with standards such as Gaia-X, EHDS, and EOSC.

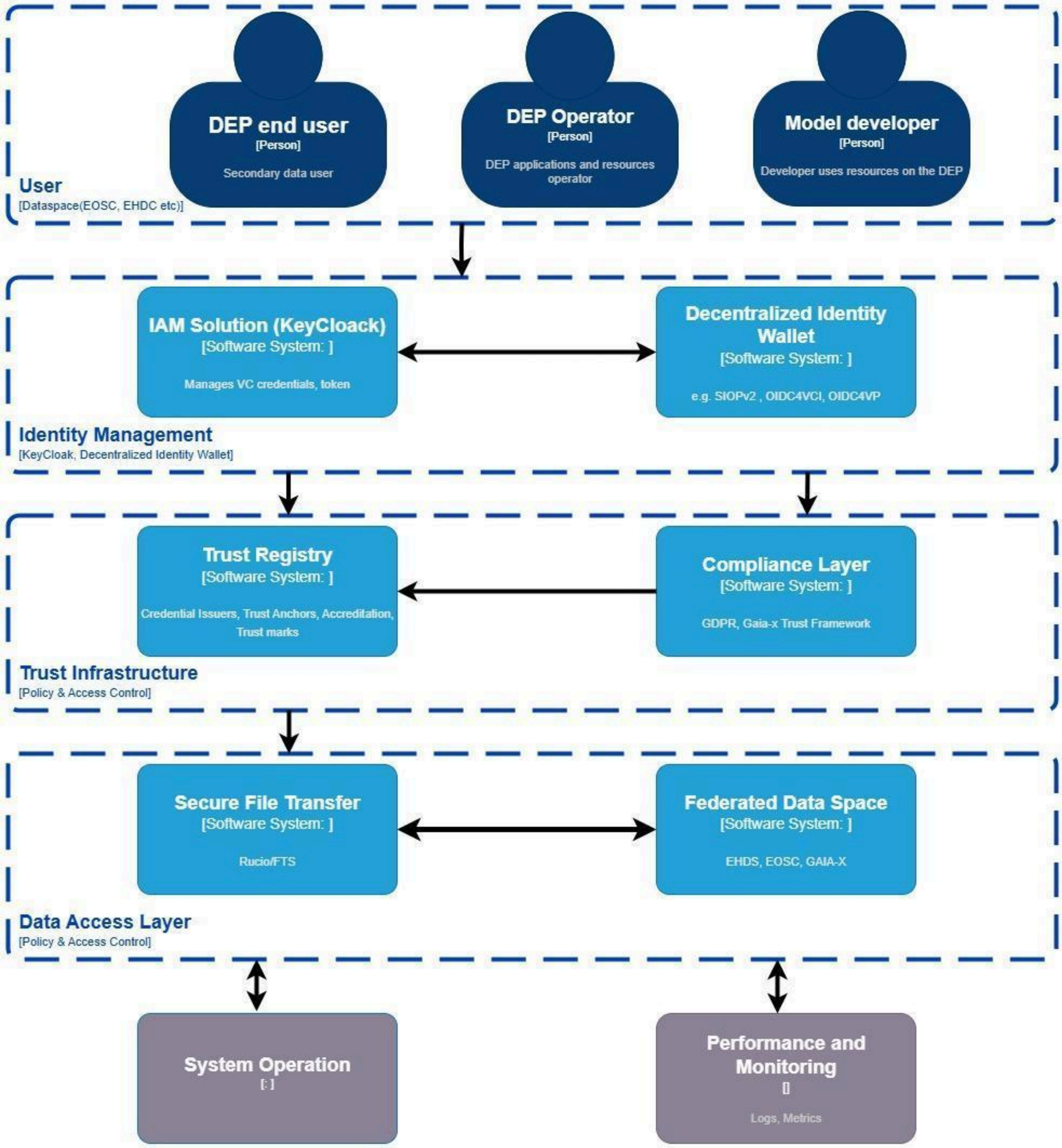

Figure 4: High-level Interoperability Framework Architecture of a trusted and federated data exchange ecosystem

4.2.1. Functional Specifications

The Interoperability Framework will provide an access management system that supports decentralised identity and access management. We will incorporate several techniques and standards to ensure that identities are verifiable, reliable, and comply with the EU regulations.

Below are some of the objectives that will be part of the framework:

- The Interoperability Framework will use a decentralised identity framework that supports both OIDC and DID resolution, allowing users to authenticate with either institutional credentials or self-managed identities.
- The Interoperability Framework will handle the issuance and verification of verifiable credentials using established standards like OID4VCI, SIOPv2, and OID4VP.
- A policy-based access control mechanism using ODRL and Attribute-Based Access Control (ABAC) will be implemented to make sure access decisions are finely tuned.
- A federated catalogue integration will be available to help find data and metadata using standardised schemas.
- The Interoperability Framework will include a trust registry to manage credential issuers and verifiers compliant with Gaia-X, ensuring trust evaluations can adapt across domains.

4.2.2. Non-Functional Specifications

Below are non-functional requirements that focus on information security, performance, risk management, and regulatory compliance processes.

1. In the Interoperability framework, all user information will be encrypted with TLS 1.3 during transit with AES-256.
2. In the Interoperability framework, every user request will use secure tokens that are revocable based on the user request and will reduce the trust level or withdraw user consent.
3. The interoperability framework will incorporate privacy features with selective disclosure or zero-knowledge proof for minimum data exposure.
4. Traceability and incident response will be supported by audit log generation using OpenTelemetry and storage in immutable backends like Grafana Loki, which are WORM-enabled.
5. The Interoperability framework will adopt modern orchestration and deployment approaches, which support modularity, scalability, and automation. These approaches enable declarative configuration, version-controlled deployments, and integration with CI/CD pipelines.

4.3. Privacy and Consent Management

This section presents the initial work done related to the Task T4.3 “Privacy-first compliance measures for federated Identity Access Management”. The objectives of this task are to assess the privacy compliance of proposed Identity Access Management (IAM) systems, to improve the confidentiality of sensitive data by applying technical solutions, and finally, to ensure that the consent and the information of the users are well managed in the context of federated IAM systems.

The following figure illustrates the expected architecture of the privacy and consent management:

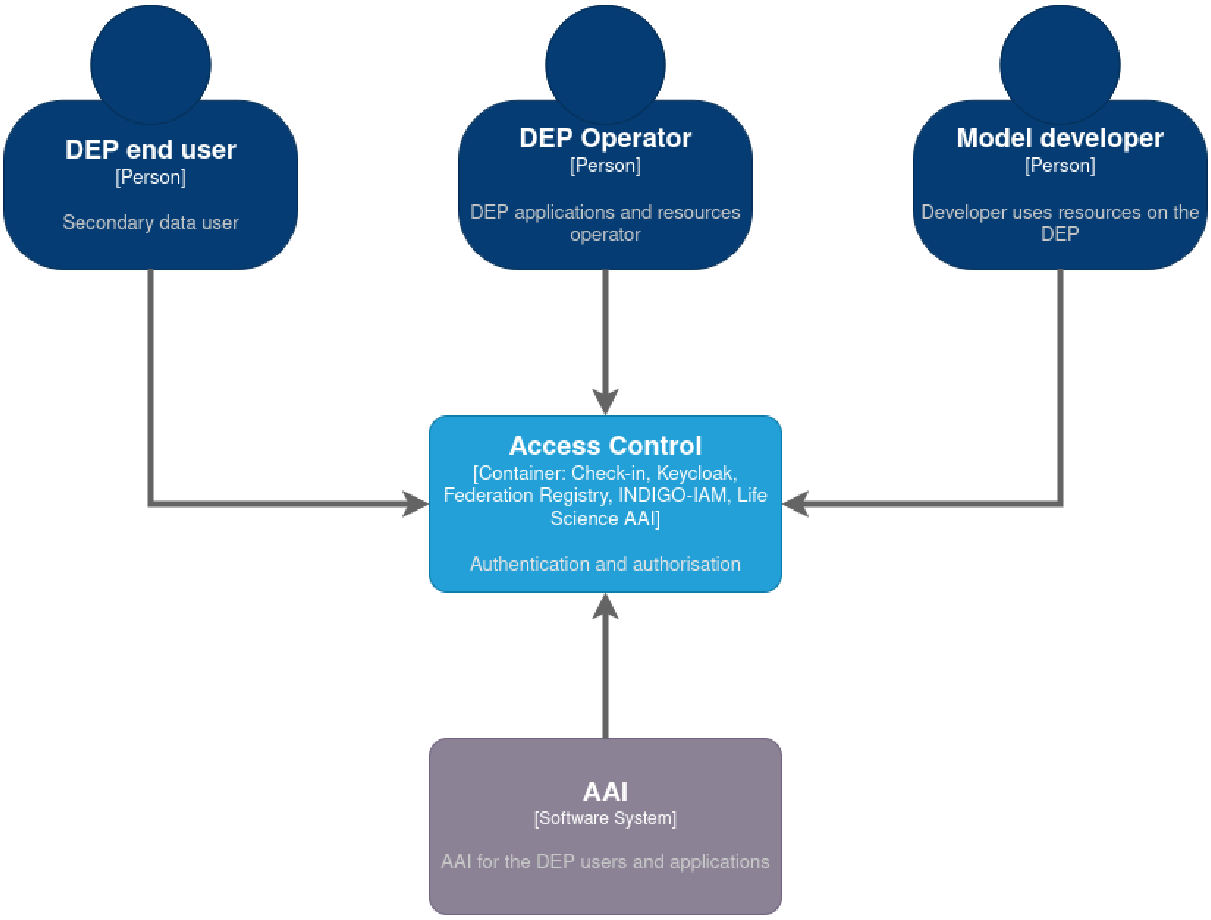
DEP end user
[Person]
Secondary data user
DEP Operator
[Person]
DEP applications and resources operator
Model developer
[Person]
Developer uses resources on the DEP
Access Control
[Container: Check-in, Keycloak, Federation Registry, INDIGO-IAM, Life Science AAI]
Authentication and authorisation
AAI
[Software System]
AAI for the DEP users and applications

Figure 5: Privacy and Consent Management Architecture

The access control is applied for each type of user as displayed in Figure 5. The DEP end-user is considered a secondary data user; in parallel, the DEP operator is responsible for the management of the DEP applications and resources. Finally, a model developer is using the resources available on the DEP. Each type of user gains the authentication and the authorisation from the access control, which depends on the AAI system.

Access control is a core subsystem of the AAI in the DEP, implemented using platforms such as Keycloak or INDIGO-IAM. Most importantly, different kinds of users, such as a DEP end-user, a DEP operator, or a model developer, interact with the access control system to gain access to DEP resources.. Compliance with legal obligations and implementation of cybersecurity requirements will follow the detailed specifications outlined below.

4.3.1. Functional Specifications

Privacy and consent management require collecting the user's consent and transparently informing each user. To achieve this, an efficient and reliable dedicated framework should be in place, using different technical solutions to guarantee the privacy of the information provided by the users. Among these technical measures, the creation and revocation of secure tokens represent a good solution for the federated IAM systems. Of course, all the actions linked to the management of the secure tokens should be registered in a proper way.

Based on the objectives of the privacy and consent management, different types of requirements and related specifications were elaborated for future implementation in the RI-SCALE. There are three categories of specifications, which are described in this part of this deliverable.

First of all, the functional specifications are based on the following requirements explaining how the implementation of the privacy and consent management should be done to be compliant with the privacy regulations, notably the GDPR:

- A user interface should be available to obtain the information given by the users. Typical information can be the organisation names and countries linked to the users.
- The same user interface should be able to display the information intended for the users. Indeed, the users have the right to be informed transparently.
- The user interface should collect the consent of each user. Based on the consent given or not by each user, actions can be executed to ensure that the will of the user is respected.

4.3.2. Non-Functional Specifications

Several non-functional specifications were elaborated based on the non-functional requirements associated with privacy and consent management. The corresponding list is below:

- A secure token is transmitted for each request to an IAM system. It means that the secure token is mandatory in each HTTP request sent to a given IAM system. This implies that the user authentication and authorisation is in place and implemented by the different IAM systems.
- The personal data should be encrypted in transit. To ensure cybersecurity and privacy, encryption is used during the exchange of data with the IAM systems. This second specification complements the secure token.
- Each secure token should be revoked properly. It means, for example, that the validity period of a given secure token should be checked regularly.
- Logs are put in place to collect all the actions related to the user’s consent and the revocation of the secure tokens. This permits a better audit in case of cybersecurity incidents or data breaches.
- The personal data should be encrypted at rest. It will guarantee privacy and reduce the risks associated with cybersecurity.

4.3.3. Legal Specifications

As detailed in the deliverable D1.3 - Ethics Requirements and Processes, the RI-SCALE project is surrounded by a complex legal framework that determines the compliance context for all project partners. In the specific case of the Privacy and Consent Management framework delineated in this deliverable, careful consideration and alignment must be ensured concerning the following specifications to uphold data subject rights:

- Lawful basis for processing: Personal data processing in the DEP must have a clearly identified and documented lawful basis as described in GDPR Articles 6 and 9.
- Informed, granular and transparent consent: Whenever data processing activities are based on consent, the GUI should provide comprehensive consent management solutions, including clear, concise and easily understandable information (e.g. on the purposes of data processing, types of data being processed, user rights (as detailed in GDPR Arts. 15-22), and how to exercise those rights) to data subjects (prior to obtaining their consent), the capability of recording consent provision by users (in a freely given, specific and unambiguous manner), and mechanisms to withdraw consent as easily as was granted originally. The GUI should allow users to consent separately to different data processing activities and ensure adherence to the principle of purpose limitation.
- Data protection by design and by default: The system should implement technical and organisational measures to minimise data collection, secure the data, and ensure that, by default, personal data is not processed unnecessarily.
- Record keeping and auditability: The system should maintain secure and auditable logs and clear documentation to demonstrate compliance with relevant regulatory frameworks (see D1.3, section 4.3), including the GDPR obligations detailed in this section.
- Compliance with national legislation: The framework of the system should be developed in a way that can be tailored with specific and/or additional privacy and security requirements applicable to the project partners, particularly as defined by national authority guidance and/or best practices of relevance to sensitive personal data handling.

4.4. Credit Management System

This section outlines the architectural design of the CRedit Management System (CRMS), detailing how it achieves its core functional goals: (1) collecting resource consumption and environmental impact metrics, (2) transparently converting usage data into credit values, taking into account green-indexing and resource's capacity, and (3) enabling policy-driven mechanisms for credit distribution and conversion. The design ensures end-to-end traceability from resource utilisation to credit distribution while maintaining compliance with sustainability and governance requirements. Building upon the defined functional and non-functional specifications, this design aims to provide a clear and actionable blueprint for development, ensuring the system not only meets its core objectives but also operates with high scalability, performance, and data reliability. By providing the internal structure of each component and its interactions with both other components and external systems, we lay the groundwork for an MVP credit management solution within the DEP ecosystem.

The overall architecture of the CRMS, as depicted in the C4 Container View[3] (Figure 6), is composed of several interconnected components:

- **Resource Usage & Environmental Impact Tracking:** This component is tasked with gathering raw data on resource consumption and environmental impact across the DEP. It collects usage metrics (e.g., core hours) and environmental indicators (e.g., energy consumption) for resources such as GPUs, CPUs, storage systems, network transfers, and AI models.
- **Credits Translation Policy Management:** This component processes raw data from the Resource Usage and Environmental Impact Tracking component and applies predefined policies to translate usage and environmental impact costs into a predefined credit system. It ensures that all resource consumption is converted into credits using consistent and transparent rules, while also taking into account the capacity of each resource as defined within its specific metric types.
- **Credits Distribution Policy Management:** This component manages the transfer of generated credits to users and projects, ensuring distribution is carried out in alignment with the predefined policies and rules.
- **Credits Allocation Registry:** This component serves as the central database, maintaining a detailed record of credit ownership, consumption, and related information. It tracks users, projects, and the credits they hold, spend, or receive discounts on, ensuring accurate and transparent accounting across the system.
- **Application Programming Interface (API):** The API acts as the main interface for interacting with the CRMS, enabling seamless communication between internal components and external users or systems. It provides access to the credit allocation registry, supports the management of credit distribution and translation policies, and allows tracking of resource usage and environmental impact data.

This modular design promotes maintainability, scalability, and independent development of each component, while the API ensures secure and standardised access to CRMS functionalities.

[3] https://c4model.com/

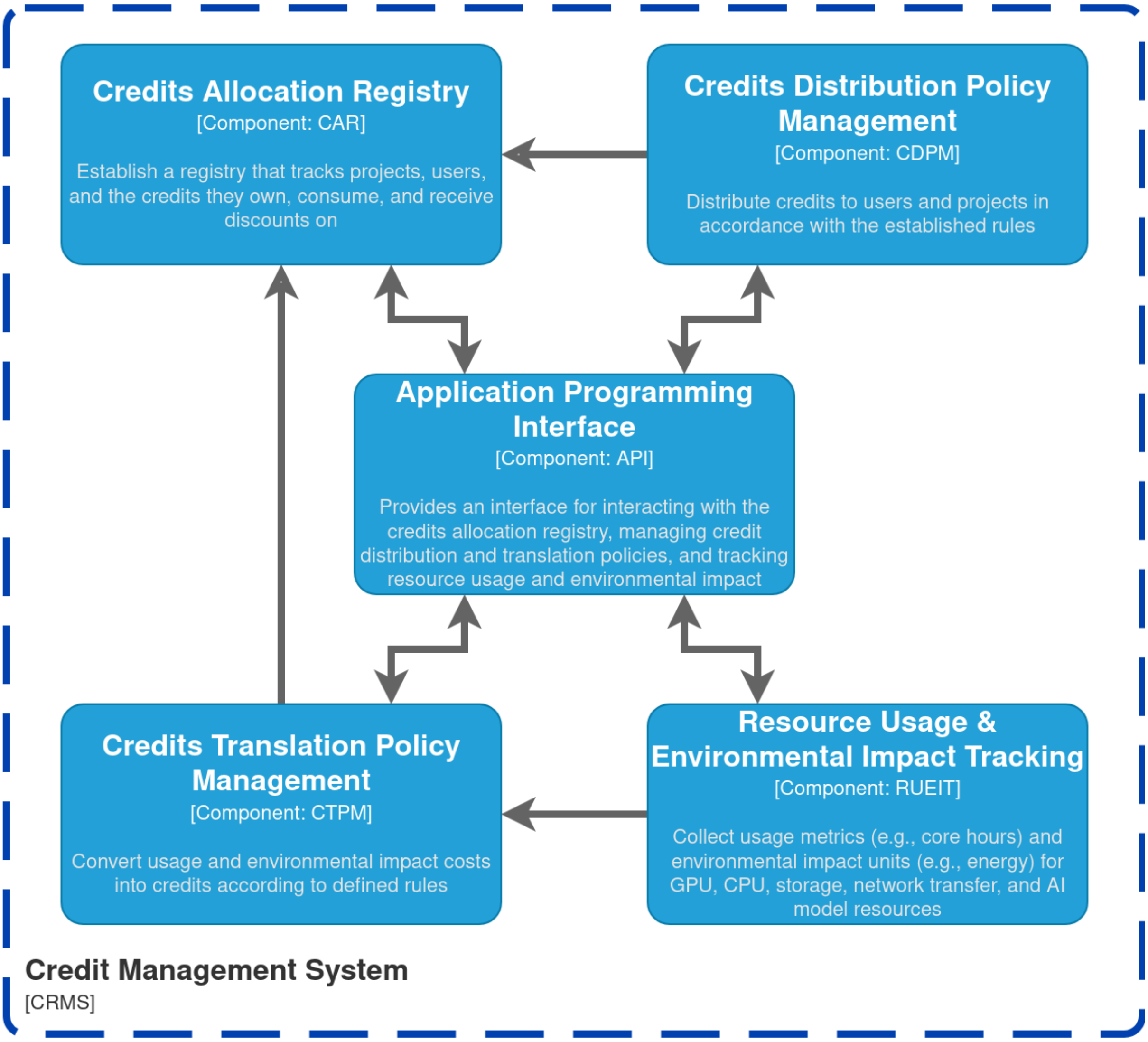
Credits Allocation Registry
[Component: CAR]
Establish a registry that tracks projects, users, and the credits they own, consume, and receive discounts on
Credits Distribution Policy Management
[Component: CDPM]
Distribute credits to users and projects in accordance with the established rules
Application Programming Interface
[Component: API]
Provides an interface for interacting with the credits allocation registry, managing credit distribution and translation policies, and tracking resource usage and environmental impact
Credits Translation Policy Management
[Component: CTPM]
Convert usage and environmental impact costs into credits according to defined rules
Resource Usage & Environmental Impact Tracking
[Component: RUEIT]
Collect usage metrics (e.g., core hours) and environmental impact units (e.g., energy) for GPU, CPU, storage, network transfer, and AI model resources
Credit Management System
[CRMS]

Figure 6: Overview of the CRedit Management System (CRMS) architecture, showing components for resource tracking, credit translation, distribution, and allocation, all communicating via an API

4.4.1. Functional Specifications

The functional specifications articulate the essential capabilities and operational requirements of the CRMS, defining what the system does to meet the needs of DEP End Users, Model Developers, and Operators. These specifications outline the system's core functionalities, ensuring precise tracking of resource usage across computational and environmental metrics, transparent and equitable credit allocation, and robust policy management for credit translation and distribution.

4.4.1.1. Resource Usage & Environmental Impact Tracking

The Resource Usage & Environmental Impact Tracking component is responsible for the comprehensive collection, aggregation, and standardisation of resource usage data and the associated environmental impact metrics across the DEP. It acts as the foundational input mechanism for the CRMS by ensuring transparency and accountability in how resources are consumed, which is important to translate into meaningful credits.

The functionality of this component can be summarised through the following key operations:

- **Usage Metric Collection**: The component gathers detailed data on resource usage across GPUs, CPUs, storage volumes, network transfers, and AI model executions. These metrics include core hours, memory consumed, volume read/write, and data transferred.
- **Environmental Impact Capturing**: For each resource type, the system is able to collect environmental units such as kWh consumed or estimated CO_2 emissions based on defined green metrics.
- **Asynchronous Harvesting**: Data is recorded with configurable temporal resolution (e.g., hourly, daily), enabling a time-based usage trend report.
- **Per-User and Per-Project Attribution**: All usage is capable of being tagged with user and project identifiers, facilitating accurate cost attribution, quota tracking, and reporting.
- **Cross-Boundary Project Support**: By employing node-level multi-tenancy authorisation mechanisms, the system ensures precise isolation and attribution of resource consumption to specific tenants – such as organisations, institutions, or projects – even in complex, distributed, and federated DEP environments. This approach enables secure, accountable, and scalable usage tracking across diverse administrative domains.
- **Standardisation**: All data collected is mapped to standardised schemas and formats to ensure compatibility and consistency across multiple DEP providers.

4.4.1.2. Credits Translation Policy Management

The Credits Translation Policy Management component is responsible for converting raw resource usage and environmental impact data into credit values. It applies well-defined and transparent policies to ensure consistent credit computation across various DEP installations, user groups, and resource types, while allowing each DEP to tailor the specific rules according to its unique operational and policy requirements. By incorporating sustainability-focused translation mechanisms - like green-index weighting - this component ensures that resource consumption aligns closely with institutional policies, funding frameworks, and environmental objectives.

The functionality of this component can be summarised through the following key operations:

- **Policy Engine & Rule Configuration**: Implements and enforces transparent credit translation policies while providing configurable rule sets that allow each DEP installation to tailor credit computation to its specific operational requirements, governance models, sustainability, green indices, and policy frameworks. This mechanism lays the foundation for introducing virtual access funds and other incentive-based programs to promote justifiable and equitable resource usage.

- **Usage and Impact Data Processor**: Converts raw resource consumption and environmental impact metrics into credit units.
- **Sustainability Integration Layer**: Incorporates sustainability-driven factors - such as green-index weighting - into credit calculations, enabling mechanisms like credit discounts for environmentally responsible user behaviour. It ensures that resource consumption is translated into credits based on consistent, transparent rules. This supports efficient resource utilisation and promotes fair credit allocation aligned with organisational sustainability goals.
- **API-Based Rule Submission**: Provides a unified interface for submitting, updating, and managing credit translation rules programmatically, ensuring streamlined integration with external systems and administrative tools.
- **Adaptive Policy Enforcement Across User Groups and Resources**: Ensures consistent policy application across diverse user groups, resource types, and DEP environments.
- **Policy Versioning and Management**: Supports version control of policies to manage updates and track changes over time.

4.4.1.3. Credits Distribution Policy Management

The Credits Distribution Policy Management component governs the equitable, transparent, and sustainable allocation of credits across the DEP. This critical module enforces predefined governance rules while supporting multiple allocation methodologies to accommodate diverse use cases.

The functionality of this component can be summarised through the following key operations:

- **User-Based Credit Distribution**: Credits are allocated to individual users or groups. They may periodically be replenished (e.g., quarterly) and may include contributions from external sources, such as funding agencies, to support a variety of project requirements or individual users. Credit policies could factor in metrics like resource consumption (e.g., CPU hours, storage) and environmental impact (e.g., energy usage), to encourage fairness and sustainability by rewarding efficient resource usage.
- **Rule-Based Allocation**: Credit assignment should be governed by predefined policies that take into account priorities such as project importance and sustainability goals. Default wildcard rules should ensure baseline access for all users, while targeted policies should automatically allocate credits to specific users or projects to align with strategic objectives. Special cases, such as milestone-based refills, should also be supported.
- **Resource Capacity Management**: Defines and manages the measurable limits of each resource based on its specific metric types (e.g., CPU time, storage, bandwidth). By establishing clear capacity boundaries, the system ensures accurate accounting of resource

usage and prevents overconsumption. These capacity definitions also serve as a reference point for normalising credit allocation.

- **Structured Credit Request Process**: Users or projects can request additional credits through a clear process; submission via the DEP interface, review by managers or automation, and approval with defined terms.
- **Flexible Assignment Methods**: Credits can be assigned manually for tailored needs or automatically via rules for efficiency. This flexibility adapts to varying workloads and Research Infrastructure (RI) setups, ensuring scalability and responsiveness.
- **API-Based Rule Submission**: Provides a unified interface for submitting, updating, and managing credit translation rules programmatically, ensuring streamlined integration with external systems and administrative tools.

4.4.1.4. Credits Allocation Registry

The Credits Allocation Registry acts as the central repository for tracking all credit-related information. It establishes a comprehensive registry of projects and users, detailing their credit ownership, consumption, and any applied discounts.

The functionality of this component can be summarised through the following key operations:

- **Project and User Registry**: The component should establish a registry that maintains associations between users, projects, and their respective credit information. It should accurately track credits owned, consumed, and any applicable discounts (e.g., green-index discounts), providing a clear view of credit balances.
- **Credit Reservation**: Credits should be reserved in advance before being formally charged to a user or project, following a model similar to a bank account. This approach accounts for asynchronous resource usage reporting or reduced credit consumption, such as when a discount is applied or a task completes earlier than expected.
- **Data Consistency:** It serves as the centralised and authoritative source for all credit allocation data, ensuring integrity and consistency across the CRMS, and ensuring that all credit-related transactions are accurately reflected.
- **API Interaction:** The component will interact with the Application Programming Interface (API) to allow other components or clients to access and update credit allocation data.

4.4.1.5. CRMS Application Programming Interface

The **Application Programming Interface (API)** serves as the central communication layer for the CRMS, enabling seamless interaction between its various components and external systems. It provides a standardised and secure interface for managing credit operations and accessing relevant data.

The functionality of this component can be summarised through the following key operations:

- **Unified Interaction Interface**: The API would offer a unified RESTful interface for seamless interaction with the Resource Usage and Environmental Impact Tracking, Credits Translation Policy Management, Credits Distribution Policy Management, and Credits Allocation Registry. It would enable external systems and clients, including DEP End Users viewing credit information and DEP Operators analysing resource consumption, to securely query, update, and manage credit data, resource usage metrics, and credit translation and distribution policies using standardised data formats and robust error handling.
- **Resource Tracking Interface**: The API facilitates the collection and submission of data for tracking resource usage and environmental impact from the Compute and other relevant systems.
- **Policy Management Interface**: It offers an interface for managing credit distribution and translation policies, allowing DEP Operators to configure and update rules for credit generation and distribution.
- **Secure Access**: The API will integrate with Access Control (AAI) to ensure secure and authenticated access for all interactions, controlling which users and systems can perform specific operations.

4.4.2. Non-Functional Specifications

Beyond its core functionalities, the CRMS must adhere to critical non-functional requirements to ensure its robust, efficient, and reliable operation within the DEP ecosystem. These specifications address the system's performance, scalability, and data integrity.

4.4.2.1. Scalability

The CRMS must be built to scale effectively with growing data volumes and user activity. Specifically:

- The system should be capable of processing and storing high volumes of usage and environmental impact data originating from multiple DEP operators. and a vast number of users. This includes accommodating concurrent data streams and batch processing requirements.
- The system must efficiently manage and store extensive records of user and project information within the Credits Allocation Registry, accommodating data from multiple DEP users and projects.
- The system should support scaling to accommodate a growing number of cross-national and international DEP projects. Its architecture will allow for seamless expansion as new projects and collaborations are introduced, without degrading overall system performance. This ensures the system can efficiently manage increasing data loads and user bases as DEP adoption expands.

4.4.2.2. Performance

Efficient and timely processing of data is paramount for a responsive user experience and smooth interoperability. Specifically:

- The CRMS must support low-latency processing and recording of usage and environmental metrics, enabling rapid data updates to ensure accurate tracking for users and projects operating across diverse DEP environments.
- The system will generate credit reports quickly for typical queries. This enables DEP operators, model developers, and end-users to swiftly access information regarding credit balances, consumption, and distribution, enhancing usability and supporting prompt decision-making.

4.4.2.3. Data Reliability

Ensuring the integrity and availability of critical usage and credit data is fundamental to the trustworthiness and continuity of the CRMS.

- The CRMS should implement robust data redundancy mechanisms to prevent the loss of resource usage, environmental impact metrics, and credit data. This includes strategies such as replication, backups, and distributed storage solutions to safeguard against hardware failures, data corruption, or other unforeseen events, ensuring continuous operational continuity and data availability.
- The CRMS should employ reliable locking mechanisms to ensure consistency and integrity during credit transactions. These mechanisms will prevent race conditions, double-spending, and data anomalies in concurrent transaction scenarios.

5. Implementation Roadmap

Several IAM stacks are used by participating RIs and Compute Providers (OpenStack and HPC) in RI-SCALE. These IAM technologies are already in production and integrated into existing services, each with its own capabilities and integration requirements. Table 9 provides a summary of the system's key capabilities and features, serving as input to the implementation roadmap in Table 10.

Table 9: Summary of the Capabilities

Capability Descriptions			
Framework / Component	Capability	Description	Provided by
Authorisation Framework	Policy Authoring in ODRL	Create and manage access policies using the ODRL model.	DEP Access Policy API
	Policy Evaluation via OPA	Evaluate access requests at runtime using OPA, which returns allow/deny decisions.	OPA
	Policy Execution Model	Support ODRL-to-Rego translation or direct evaluation of structured ODRL input within OPA, depending on implementation/deployment preferences.	OPA *(via Rego or structured ODRL with IAM claims)*
	Dynamic Attribute Evaluation	Evaluate claims from tokens or retrieve them from external sources (PIPs).	OPA + IAM *(INDIGO IAM, Keycloak/EGI Check-in, Perun AAI/LS AAI)*
	OPA Deployment Model	Supports IAM-enforced (OPA-hidden) and service-enforced (OPA-exposed) integration patterns depending on trust boundaries and implementation/deployment preferences.	OPA + IAM *(integration model depends on IAM deployment — either OPA-hidden or OPA-exposed)*
	Visual Policy Editor (UI)	Web-based interface for administrators to create, edit, and validate policies using structured templates and visual tools.	DEP Access Policy UI
	Policy Audit Logging	Log authorisation decisions and admin actions in a structured format.	OPA + IAM + logging stack *(OPA*

			logs decisions, IAM logs auth events)
Interoperability Framework	DID & VC Support	Enables issuing, presenting, and verifying decentralised credentials (DID/VC) using OID4VCI, SIOPv2, OID4VP.	Keycloak (VC plugins), Wallets, Verifiers
	Trust Registry Integration	Integrates with the Gaia-X Trust Framework to validate credential issuers and verifiers.	Gaia-X Trust Anchor Registry
	OIDC-DID Resolution	Maps institutional identities (OIDC) to decentralised identifiers (DIDs) for dual-mode authentication.	Identity Resolver Service
	Federated Ecosystem Interop	Supports credential and trust policy exchange with EOSC, EHDS.	Interop APIs + VC Middleware
	ABAC + ODRL Policy Enforcement	Makes access decisions based on trust level, credential metadata, and identity attributes using policy engines.	OPA + IAM (e.g., Keycloak + DEP Access Policy)
	ABAC + ODRL Policy Enforcement	Makes access decisions based on credential metadata, trust scores, and user attributes using dynamic policies.	OPA, Keycloak + DEP Access Policy UI, ODRL → Rego mappers
	Token & Consent Management	Manages secure, revocable tokens linked to user consent, TTL, and access purpose tracking.	IAM (Keycloak), Consent UI, VC metadata policies
	Security & Privacy Controls	Ensures TLS 1.3/mTLS in transit, AES-256 at rest, and applies ZKPs and selective disclosure for privacy.	Envoy Gateway, Vault, etc
	Audit & Observability	Logs all credential and token actions for traceability, with real-time monitoring and tamper-proof audit trails.	OpenTelemetry, Grafana Loki, SIEM Integration
	Performance & Scalability	Supports sub-second AuthN/AuthZ with scalable microservices and local credential metadata caching.	Kubernetes, Redis, Async VC validation pipelines

Privacy and Consent Management	Management of the user's consent and the privacy enforcement	Collection of consent by the users and related personal data, logging, and audits of the actions associated with the access tokens.	IAM (e.g., Keycloak + DEP Access Policy) + clients accessing IAM
	Management of tokens, including their revocation	Management of the access token, in particular, involves the revocation of tokens.	IAM (e.g., Keycloak + DEP Access Policy) + clients accessing IAM
	Audit logs	Tracks the actions linked to the user's consent and the token creations and revocations in the logs.	IAM (e.g., Keycloak + DEP Access Policy) + clients accessing IAM
	Encryption	Encryption in transit and at rest.	IAM (e.g., Keycloak + DEP Access Policy) + clients accessing IAM
Credit Management System	Resource Consumption Tracking	Tracks detailed metrics (e.g., core hours, memory, storage, network transfers) for DEP resources like GPUs and CPUs, ensuring accurate monitoring of usage for fair credit allocation and transparency.	Resource Usage & Environmental Impact Tracking, logical component
	Environmental Impact Harvesting	Collects environmental metrics (e.g., kWh, CO_2 emissions) from DEP resource usage, enabling sustainability assessments and paving the way for green policy support like efficiency-based discounts.	Resource Usage & Environmental Impact Tracking, logical component
	Credit Translation based on unit cost and sustainability policies	Converts resource usage and environmental data into credits using unit costs and green-index policies, ensuring consistent, transparent, and sustainable credit calculations.	Credits Translation Policy Management logical component
	Credit distribution based on the resource's capacity and DEP	Allocates credits to users and projects based on resource capacity and DEP policies, supporting fair distribution and flexible methods like periodic or milestone-based assignments.	Credits Distribution Policy Management logical component

	policy-driven rules		
	Project- and User-Level Credit Ownership and Usage Tracking	Maintains a database of credit allocations and usage for users and projects, ensuring accurate, transparent tracking and compliance with allocation policies.	Credits Allocation Registry logical component
	Scientific Validation and Iterative Refinement of Credit Models	Validates and refines credit models using usage data and feedback, ensuring fair, accurate, and adaptable credit calculations aligned with DEP and sustainability goals.	The orchestration of CRMS components and their interaction with external clients
	Feedback for Future Plans	Document feedback from usage and performance data to refine access policies and virtual access fund allocations, ensuring the DEP ecosystem evolves to meet user needs and environmental objectives.	The scientific validation through iterative refinement

Table 10 maps each capability, as described above, to its corresponding implementation timelines, highlighting how the different AAI stacks and CRMS components will be progressively integrated into the DEP architecture.

Table 10: Access Management Systems' Implementation Roadmap

Implementation Roadmap				
Capability	Fulfilled Requirements	Priority (MoSCoW)	Timeline (M12: 1st DEP release M24: 2st DEP release M36: End of project)	Notes
Authorisation Framework				
Policy Authoring in ODRL	RSREQ-36, RSREQ-43, RSREQ-71, RSREQ-72	Must, Must, Must, Must	By M12	Enables structured authoring and storage of ODRL policies
Policy Evaluation via OPA	RSREQ-36, RSREQ-73, RSREQ-84,	Must, Must,	By M12	Core runtime decision mechanism for access control

	RSREQ-85	Must, Must		
Policy Execution Model	RSREQ-72, RSREQ-73	Must, Must	By M24	Flexibility to support Rego translation or structured ODRL input
Dynamic Attribute Evaluation	RSREQ-73, RSREQ-85	Must, Must	By M12	Evaluates claims from tokens or via PIPs for context-aware decisions
OPA Deployment Model	RSREQ-54, RSREQ-73, RSREQ-83, RSREQ-84, RSREQ-85, RSREQ-97	Should, Must, Must, Must, Must, Should	By M24	IAM-embedded or service-side enforcement, depending on integration needs
Visual Policy Editor (UI)	RSREQ-82	Could	By M36	Not essential; UI support improves usability for admins
Policy Audit Logging	RSREQ-74	Must	By M36	Tracks decisions, actions, and events
Interoperability Framework				
Decentralised Identity Protocols	RSREQ-47, RSREQ-55, RSREQ-57, RSREQ-85	Must, Should, Could, Must	By M24	Support for SIOPv2, OID4VCI, and OID4VP for federated and self-sovereign identity integration
IAM Extensions	RSREQ-48, RSREQ-53, RSREQ-57, RSREQ-85	Must, Must, Could, Must	By M24	Extend IAM systems (e.g., Keycloak) to support verifiable credentials and decentralised flows
Trust Infrastructure	RSREQ-49, RSREQ-50, RSREQ-57, RSREQ-66	Must, Must, Could, Should	By M36	Implement decentralised trust anchors, accreditation, and Gaia-X-compliant trust registries
Federated Ecosystem Interop	RSREQ-53, RSREQ-54, RSREQ-57	Must, Must, Could	By M36	Integrate with EOSC, EHDS, GAIA-X, and validate secure VC-based data exchange using Rucio/FTS and standard APIs
Policy & Access Management	RSREQ-55, RSREQ-60, RSREQ-62,	Should, Must, Should	By M36	Enforce ODRL/ABAC policies with OPA, ensure interoperability

	RSREQ-34			with standard protocols, and maintain a modular IAM architecture
Token & Consent Management	RSREQ-52, RSREQ-65, RSREQ-67, RSREQ-68, RSREQ-69, RSREQ-70	Must	By M36	Provide secure, revocable tokens and user interfaces to manage consent, TTL, and purpose-limited credential usage
Security & Privacy	RSREQ-56, RSREQ-58, RSREQ-64, RSREQ-104	Must, Must, Must, Should	By M36	Use TLS 1.3/mTLS for transit, AES-256 for rest, ZKPs for privacy, and ensure full GDPR-compliant data protection
Audit, Performance & Scalability	RSREQ-59, RSREQ-84, RSREQ-63	Must, Must, Must	By M24	Enable sub-second AuthN/AuthZ, scalable microservices, and full OpenTelemetry logging for traceability and audit
Privacy and Consent Management				
Token management	RSREQ-52, RSREQ-69, RSREQ-70, RSREQ-55	Must, Must, Must, Should	By M24	Management of secure tokens used within IAM
User Interface for consent and privacy management	RSREQ-65, RSREQ-67, RSREQ-68	Must, Must, Must	By M24	Collection and display of users' information
Encryption in transit	RSREQ-56	Must	By M24	For personal data in transit
Encryption at rest	RSREQ-104	Should	By M24	For personal data at rest
Credit Management System				
Resource Consumption Tracking (1)	RSREQ-44, RSREQ-80,	Must, Should	By M12	Resource Consumption Accounting for CPU, GPU, and Storage
Resource Consumption Tracking (2)	RSREQ-44, RSREQ-79, RSREQ-80, RSREQ-81	Must, Should, Should, Should	By M24	Accounting for data transfers for the AI frameworks

Environmental Impact Harvesting	RSREQ-75, RSREQ-79, RSREQ-80, RSREQ-81	Must, Should, Should, Should	By M24	Environmental-Impact accounting of DEP resources, aiming to support sustainability goals
Credit Translation based on unit cost and sustainability policies	RSREQ-76	Must	By M24	Converts usage and environmental data into credits using unit costs and green policies
Credit distribution based on the resource's capacity and DEP policy-driven rules	RSREQ-105	Must	By M36	Allocates credits to users and projects based on resource capacity and DEP policies
Project- and User-Level Credit Ownership and Usage Tracking	RSREQ-105, RSREQ-81	Must, Should	By M36	Maintains a database for credit allocations and usage, ensuring transparency and policy compliance
Scientific Validation and Iterative Refinement of Credit Models	RSREQ-77	Must	By M36	Scientific Validation through consumption pilots
Feedback for Future Plans	RSREQ-78	Should	By M36	Recommendations for further development and adoption of virtual access

6. Conclusion

This deliverable provides a comprehensive specification and roadmap for the Access Management Systems within the DEP ecosystem as part of the RI-SCALE project. By defining the Access Management Architecture and the CRedit Management System (CRMS), this document establishes a robust framework for secure, equitable, and sustainable access to computational resources across distributed RIs. The Access Management Architecture, encompassing the Authorisation Framework, the Interoperability Framework, and the Privacy and Consent Management subsystem, addresses the critical needs of secure and privacy-preserving access for DEP end-users, model developers, and operators. The CRMS complements this by enabling transparent tracking of resource usage and environmental impacts, translating these into credits through sustainability-focused policies, and ensuring fair credit distribution via a centralised registry.

The modular and scalable design of these systems ensures flexibility and adaptability to diverse RI environments, supporting seamless integration with existing infrastructure and compliance with European data management frameworks and regulations, such as the Gaia-X and the GDPR, respectively. The phased implementation roadmap outlined in Section 5 prioritises key capabilities, providing a clear path for development, validation, and deployment across project milestones. By addressing the needs of infrastructure providers, end-users, model developers, and operators, the Access Management Systems lay a strong foundation for enhancing AI-driven data analysis, fostering equitable resource access, and promoting sustainability within the DEP ecosystem.

This deliverable serves as a blueprint for advancing the RI-SCALE project's objectives, ensuring the DEP's ability to bridge data and computation while meeting the complex demands of modern research infrastructures.