D16.1 Plans for the testing and validation of the EOSC Beyond Pilot Nodes

16/03/2026

Abstract

This document presents a comprehensive testing and validation framework for the EOSC Beyond Pilot Nodes network, establishing a federated European Open Science Cloud (EOSC) infrastructure. The methodology encompasses three core components: a Minimum Viable Product defining baseline participation requirements, systematic testing across individual Nodes and integration points, and Scientific User Stories with acceptance criteria ensuring practical research value. Ten Pilot Nodes spanning diverse domains—from social sciences to structural biology, distributed computing to climate science—undergo rigorous evaluation against standardised criteria whilst accommodating domain-specific requirements. Validation operates at multiple levels: Node-level assessment of service registration, metadata provision, authentication integration, and interoperability standards; and integration-level verification of unified access across heterogeneous systems. Results demonstrate technical integration feasibility and confirm researcher value through reduced friction in accessing distributed resources. The best practices establish a replicable model for broader EOSC deployment, supporting the vision of a pan-European open science infrastructure serving all research disciplines.

Funded by the European Union

EOSC Beyond receives funding from the European Union's Horizon Europe research and innovation programme under grant agreement No. 101131875.

Document Description

D16.1 Plans for the testing and validation of the EOSC Pilot Nodes			
WP 16			
Due date	16/03/2026	**Actual delivery date:**	16/03/2026
Nature of document	Report	**Dissemination level**	Public
Document Status	Under EC review	**Version**	1.0
Lead Partner	CYFRONET		
Authors	Katarzyna Lechowska-Winiarz and Roksana Wilk (CYFRONET), Marta Gutiérrez David (EGI), John Shepherdson (CESSDA)		
Reviewers	Androniki Pavlidou (OpenAIRE), Carlos Brandt (EGI Foundation)		
Approved by	Diego Scardaci on behalf of TCB		
Document link	https://documents.egi.eu/document/4134		
DOI	https://zenodo.org/records/19004563		
Keywords	EOSC Federation, EOSC Pilot Node, EOSC Data Space, Service Integration, Network of Pilot Nodes, Integration Matrix, EOSC Core Innovation Sandbox		

Revision History

Issue	Date	Description	Author/Reviewer
V 0.1	07/11/2025	ToC	Katarzyna Lechowska-Winiarz (CYFRONET), Roksana Wilk (CYFRONET), Marta Gutierrez (EGI)
V0.2	13/11/2025	Added MVP and testing approach sections; completed CESSDA User Stories section	John Shepherdson (CESSDA)
V0.3	16/12/2025	Added document structure and executive summary sections; completed CESSDA Node testing and validation layer section	John Shepherdson (CESSDA)
V0.4	13/01/2026	Added Abstract; responded to comments in the CESSDA section.	John Shepherdson (CESSDA)
v.05	03/02/2026	Content from the service providers and the pilots	All relevant project partners
v.06	11/02/2026	Added missing sections and chapter intros. Unification of the chapters structure.	Roksana Wilk (Cyfronet)
v.1.0	**16/03/2026**	**Final**	**Andrea Anzanello (EGI)**

Table of content

Copyright and licence info

Tables

References

1. N. Fiore and D. Scardaci, ‘EOSC Beyond D5.3 EOSC Platform Architecture and Network of EOSC Nodes’, Zenodo, Jul. 2025. doi: 10.5281/zenodo.16566006.

Executive Summary

This deliverable presents the comprehensive testing and validation approach for the network of EOSC Beyond Pilot Nodes, documenting a critical phase in establishing a federated European Open Science Cloud infrastructure. The work on establishing a network of Pilot Nodes demonstrates how diverse research infrastructures and service and content providers (thematic) can be successfully integrated into a unified service ecosystem while maintaining their individual operational characteristics and serving their specific scientific communities.

The methodology established for this validation exercise centres on three key components:

- Defining a Minimum Viable Product (MVP) that sets baseline requirements for Pilot Node participation,
- Implementing a systematic testing and validation approach across both individual Nodes and integration points,
- Defining clear acceptance criteria for the Scientific User Stories to ensure the network of Nodes meets real-world research needs.

This framework provides both technical rigor and user-centered validation, ensuring that the network is not only technically sound but also of practical value to the scientific community.

Ten Pilot Nodes representing a broad spectrum of research domains will undergo thorough testing and validation. These Pilot Nodes span domains from social sciences (CESSDA) to structural biology (Instruct ERIC and CNB-CSIC), from distributed computing infrastructure (EGI and E-INFRA CZ) to domain-specific research environments in climate science (ENES), biodiversity (LifeWatch ERIC), food systems (METROFOOD-RI), and regional research initiatives (NI4OS and NFDI). Each Pilot Node will be evaluated against standardised criteria while accommodating the unique requirements and constraints of their respective scientific domains.

The validation process operates at multiple levels:

1. At the Node level, each Pilot Node has to demonstrate its capability in service registration, metadata provision, authentication and authorization integration, and adherence to interoperability standards.
2. At the integration level, testing will verify that the core service layer effectively mediates between heterogeneous systems, enabling unified access while respecting individual Node governance and technical architectures. The Scientific User Stories provide essential validation from the researcher perspective, confirming that the integrated infrastructure supports actual research workflows such as data discovery across multiple repositories, cross-domain data access, and seamless authentication across services.

The results will indicate whether or not there is successful technical integration across all Pilot Nodes, with each demonstrating functional connectivity to the core services layer and meeting baseline interoperability requirements. The acceptance criteria derived from

Scientific User Stories must be met, in order to confirm that the network delivers value to researchers by reducing friction in accessing distributed resources and enabling new forms of cross-disciplinary collaboration. Challenges identified during validation (including variations in metadata standards, authentication protocol implementations) will inform refinements to both technical specifications and governance processes.

The testing and validation process is intended to provide evidence that the federated EOSC model is viable and that diverse research infrastructures can effectively interoperate within a common framework. The successful validation of these Pilot Nodes will establish a foundation for broader deployment and offer a replicable model for onboarding additional Pilot Nodes. Documenting the lessons learned and best practices will help guide the next phase of EOSC expansion, supporting the vision of a cross-European open science infrastructure that serves researchers across all disciplines.

1. Introduction

1.1. Structure of the document

This document is organised into five main sections. Following the introduction, **Section 2** outlines the methodology, including the definition of a Minimum Viable Product for Pilot Nodes, the overall testing and validation approach, and pilot identity criteria. **Section 3** describes the Node registration testing and validation process. **Section 4** covers the core service integration layer and its testing framework. **Section 5** presents detailed testing and validation results for each of the ten Pilot Nodes, with each Pilot examined through both technical validation and Scientific User Story test and acceptance criteria. The document concludes with a forward-looking discussion of the next steps.

1.2. Definitions

The definitions in this table referring to the EOSC Nodes have been applied in this document in the context of a Pilot Node.

Definition	Description
EOSC Core Innovation Sandbox	The EOSC Beyond Innovation Sandbox is a pre-production environment where researchers, service providers, and Node operators can test and integrate the latest EOSC Core developments in a safe, controlled setting. It acts as a Federator Node for project pilots, offering Core Federating Capabilities
EOSC Resource	A digital entity with a persistent identity, a name (or title), a description, a publishing date, and associated with an EOSC Node. It acts as the foundational concept for services, data, publications, software, and other research resources.
Resource Provider	An organisation making a Resource available.
EOSC Node Architecture	A reference architecture that can be implemented by each Node of the Federation for the operation of their services and resources
EOSC Node	An entity complying with the EOSC Federation policies and legal framework, working at the local, national, regional, thematic or European level. An EOSC Node offers added value resources to the EOSC Federation and delivers federating capabilities in collaboration with other EOSC Nodes. Each EOSC Node has its autonomy, its own governance model and an offer in terms of resources. It operates its own platform, complying with the technical framework, and the EOSC Federation Platform architecture.

EOSC Beyond Pilot Network	The network of Pilot Nodes created by the project.
Onboarding (As in EOSC Federation Handbook)	Resources are onboarded in an EOSC Node when they are integrated into an EOSC Node. The provider selects the most appropriate Node based on the characteristics of the resource. For example, if a provider has a domain-specific data repository, it may choose a thematic Node that best represents and serves that domain. A resource can be onboarded to multiple Nodes.
EOSC Thematic Node	Digital infrastructure specialised for a specific scientific domain of research or with specific techniques.
EOSC National/Regional Nodes	An EOSC Node for a regional or national community.
EOSC Federating Capabilities	EOSC Federating Capabilities are the added-value capabilities offered by the EOSC Federation that allow all EOSC end-users and providers to exploit services, data and other resources in the Federation.
Node Core functions	The Node Core functions enable the basic operation of an EOSC Node. An initial reference implementation of these Node Core functions has been implemented by the EOSC EU Node. The Node Core functions can be implemented by an EOSC Node as functions of a platform, individual services, or acquired through the EOSC Federation (offered as-a-Service or using the technology of the reference implementation).
Node Exchange services	Services and other resources a Node shares with the EOSC Federation. They contribute to the collective EOSC Exchange.

Table 1: Definitions

2. Methodology

2.1. Minimum Viable Product for a Pilot Node

The Minimum Viable Product (MVP) for a Pilot Node represents the baseline configuration and functionality required for meaningful participation in the EOSC Beyond network. It defines the essential components, interfaces, and behaviours that must be in place for a Node to be recognised as a valid, operational entity within the network.

At the core of the MVP are a set of mandatory capabilities that demonstrate compliance with the fundamental architectural and interoperability principles of the EOSC Beyond ecosystem[1].:

- Node Home page: a landing/entry page for the Node's users must be available.
- Service Exposure: The Node must expose at least one service or dataset through standardised interfaces that can be registered and discovered via the EOSC Node Registry.
- Authentication and Authorisation Integration: The Node must support federated identity management compatible with the EOSC Beyond Access and Authentication Infrastructure.
- Operational Monitoring: Basic observability features - such as service health checks and uptime metrics - must be implemented to support automated testing and ongoing validation.
- Documentation and Configuration Metadata: Sufficient technical documentation and metadata must be provided to enable testing, integration, and reproducibility.
- The Node must be registered with the EOSC Node Registry.

Establishing the MVP provides a consistent baseline for all participants, ensuring that testing and validation activities are applied to Nodes that meet a minimum level of maturity and readiness. It also supports incremental development by allowing individual Nodes to evolve beyond the MVP through iterative enhancement, guided by results from the successive stages of testing and feedback gathered during integration and scientific validation.

2.2. The overall approach to testing and validation

The testing and evaluation framework follows a multilayered approach, designed to ensure comprehensive verification of functionality, interoperability, and scientific value. This approach has three complementary levels of assessment:

- Node Testing and Validation,
- Integration Testing and Validation (in particular integration with Federating Capabilities),
- Scientific User Stories and Acceptance Criteria.

Together, these layers establish a coherent methodology for confirming that the Pilot Node performs as intended, both in isolation and as part of the wider network, and meets or exceeds the MVP criteria.

2.2.1. Node Testing and Validation

Node testing and validation focus on the individual applications, services, and systems that constitute the Pilot Node, in particular on the Node Core capabilities. The objective of this activity is to verify that all internal components are correctly implemented, properly configured, and fully interoperable within the Node's local environment. Testing activities include functional, performance, and reliability assessments to confirm that each component operates according to specifications. This phase ensures that the Pilot Node functions as a

stable, coherent, and self-contained system capable of supporting subsequent stages of integration.

Node testing and validation process depends on the specific Node implementation and can vary a lot. For this reason, this process is further detailed directly in **Section 5** with a series of tests specific for the Core and Exchange services of each Node.

2.2.2. Integration Testing and Validation

Integration testing and validation extend the scope of assessment to the federation level, in particular in relation to the EOSC Core Federating Capabilities (AAI, Catalogues, Helpdesk, etc.). The primary goal of this phase is to confirm that the Pilot Node can be successfully integrated into the broader federation of Nodes, thereby enabling the exchange of data, services, and workflows across the distributed infrastructure. This includes verification of communication protocols, interoperability standards, authentication and authorisation mechanisms, and data-sharing procedures. Successful completion of this phase demonstrates that the Pilot Node effectively provides resources that can be consumed by the federated ecosystem and adheres to the overarching architectural and operational principles of the EOSC Pilot Node network. In doing so, it provides evidence that the Pilot Node meets the expectations and requirements of its targeted user community.

The EOSC Node Registry

The EOSC Node Registry plays a central role within the testing and evaluation framework, serving as the authoritative reference point for all Pilot Nodes that constitute the EOSC Beyond network. The registry provides an overview of each participating Node, via which the detailed metadata describing the services it offers, the endpoints through which those services are exposed, and other relevant operational characteristics can be obtained.

By maintaining this structured and up-to-date information, the Node Registry enables the systematic discovery, tracking, and verification of Nodes across the network. Many of the required Node-level and Integration-level tests can be executed programmatically, drawing directly on the information accessed via the registry.

This automated capability improves both the efficiency and consistency of the validation process, reducing manual intervention while ensuring that all Nodes are evaluated against uniform criteria. It also enhances scalability, allowing the testing framework to accommodate the progressive expansion of the EOSC Beyond network as additional Pilot Nodes are deployed and federated.

Node registration in the Node registry testing and validation process is detailed in **Section 3**. Integration testing and validation process against EOSC Core Federating Capabilities is presented in **Section 4**.

2.2.3. Scientific User Stories and Acceptance Criteria

Tests and related acceptance criteria have been defined for each scientific user story to perform their thorough validation. They are reported per each Pilot Node in **Section 5**.

3. Node registration testing & validation

This section defines the pre-integration and post-integration tests for the registration of a Node in the Node Registry. The objective is to verify that the Node Registry has valid information for each Node part of the federation. In order to define these steps the **EOSC Federation handbook** is used as the main guideline.

<table>
<tr><th colspan="2">Node registry</th></tr>
<tr><td colspan="2">Enrolling an EOSC Node in the federation requires a combination of legal, organisational and technical preparations to ensure interoperability and compliance with the EOSC Federation framework.
The following checklist outlines the requirements for a Node to register with the EOSC Node Registry (ENR) and successfully join the federation.
● Legal and Organisational Preparation
● Each Node needs to be represented by a legal entity called Hosting Organisation. Therefore a Hosting Organisation needs to be named for each Node.
● Nominate Key Personnel: Each Node must appoint specific individuals to mandatory roles, including a Coordinator, Operation Manager, Technical Coordinator, and Security Officer.
● Set up the EOSC Node Endpoint (ENE): The Node must operate a functional ENE API that lists its capabilities and metadata in a machine-actionable format.</td></tr>
<tr><td>Tests to be performed
Step 1: All the paperwork is in order.
Step 2: Test Node Endpoint responsiveness to queries.</td><td>Validation metrics
Step 1: All the paperwork is available.
Step 2: Node Endpoint responds successfully (true / false).</td></tr>
<tr><th colspan="2">Integration with Pilot Nodes</th></tr>
<tr><td colspan="2">To register with the EOSC Node Registry (ENR), the following metadata must be provided for the Node entry:
● Basic Identity: Name, logo, description, and the Node's website URL.
● Legal Identification: Name and acronym of the legal entity, including its ROR ID.
● Persistent Identifier (PID): A unique PID for the Node, typically obtained through integration with the EOSC PID Service.
● Endpoint Information: The URL for the Node Endpoint (ENE) API that returns metadata describing the Node's capabilities.
As soon as the Registration is complete the Node registry will list the aforementioned Node Metadata in its API.</td></tr>
<tr><td>Tests to be performed
Step 1: Validate the metadata for a specific Node registered in the Node Registry.</td><td>Validation metrics
Step 1: The API endpoint follows a specific pre-defined schema</td></tr>
</table>

Node registry	
Step 2: Validate the Node Endpoint and the capabilities it lists. Step 3: For each capability listed perform the equivalent validation as defined in **Chapter 4** below.	Step 2: The API responds with Node Metadata and its complete including Node Endpoint (ENE). Step 3: The Node Endpoints lists at least 2 Capabilities as AAI and Resource Catalogue capabilities are mandatory for each Node.
Chapter 4 provides further details on testing and validation of other capabilities.	

Table 2: Node registration tests in the Node Registry

4. Core service integration layer

This chapter defines the test specifications used to demonstrate and validate that each pilot component is correctly integrated with the EOSC Core Federating Capabilities. Its purpose is to provide a repeatable, measurable set of tests that prove functional and technical connectivity between pilots and Sandbox's core services. This will create the basis for the framework of Testing and Validation of Core Integrations in EOSC Beyond project.

For each core service there is a list of tests to be performed pre- and post-integration, along with the validation metrics to assess the correctness of the integrations.

4.1. AAI

This section defines the pre-integration and post-integration tests for the Authentication and Authorisation Infrastructure (AAI) integration of Pilot Nodes. The objective is to verify that Pilot Nodes can rely on the federated AAI layer for federated authentication and token-based access to protected services, in line with the **EOSC AAI Architecture 2025** and the **AARC Blueprint Architecture**.

AAI integration is evaluated against two capabilities:

- Federated Single Sign-On (SSO) for interactive access;
- Using tokens for API access across Pilot Nodes.

<table>
<tr><th colspan="2">AAI</th></tr>
<tr><td colspan="2">PRE-INTEGRATION TESTS
These tests verify that a Pilot Node is technically and operationally ready to integrate with the AAI Federation. The pre-integration tests focus on validating prerequisites such as the presence of an Infrastructure Proxy, support for required federation protocols and flows, correct handling of federated identity attributes, and documented token validation mechanisms. Successful completion of pre-integration tests demonstrates that the Pilot Node meets the baseline AAI readiness requirements and can proceed safely to federation-level integration.</td></tr>
<tr><td>Tests to be performed
1. Verify that the Pilot Node operates at least one Infrastructure Proxy (as defined in the AARC Blueprint Architecture [AARC-BPA-2019]) that can be connected as an OpenID Connect Relying Party.
2. Verify that a Community AAI (if present) can be connected as an OpenID Provider.
3. Verify support for OpenID Connect Discovery:</td><td>Validation metrics
1. At least one Infrastructure Proxy is deployed and can be connected as an OpenID Connect Relying Party.
2. Where applicable, a Community AAI is deployed and operates as an OpenID Provider.
3. OpenID Connect Discovery:
a. The Infrastructure Proxy supports dynamic consumption of OpenID Provider metadata; and
b. OpenID Provider metadata is reachable for all declared OpenID</td></tr>
</table>

a. as an OpenID Relying Party, to dynamically determine OpenID Provider metadata; and b. as an OpenID Provider (when a Community AAI is present), to publish provider metadata. 4. Verify support for the OpenID Connect Authorization Code flow with mandatory use of: a. Proof Key for Code Exchange (PKCE); and b. the nonce parameter (see OIDC-Core Section 3.1.2.1). 5. Verify that insecure OAuth/OIDC flows, in particular the Implicit Grant and any response type returning access tokens directly in the authorisation response, are prohibited. 6. Verify that SAML 2.0 is not used for inter-Node federation and is supported only for accessing legacy local Node services, where required. 7. Verify that the Pilot Node AAI can receive and process federated identity information, including: a. a public subject identifier; b. name and email attributes; c. group and role information; d. identity assurance information. 8. Verify that the Pilot Node AAI supports token validation via: a. OAuth 2.0 Token Introspection (RFC 7662); and b. OAuth 2.0 Proxied Token Introspection (AARC-G052).	Providers (i.e. Community AAIs) by concatenating the string `/.well-known/openid-configuration` to the Issuer. 4. Support for the OpenID Connect Authorisation Code flow with mandatory use of PKCE and the nonce parameter is documented. 5. No insecure OAuth/OIDC flows (including the Implicit Grant) are enabled for cross-Node federation. 6. SAML 2.0 endpoints, if present, are clearly scoped to legacy local services and are not used for inter-Node federation. 7. Attribute mappings and processing logic are defined and documented for: a. Public subject identifiers (see AARC-G026 and AARC-G056-DRAFT); b. Name and email attributes (see OIDC-Core and AARC-G056-DRAFT); c. Group and role information (see AARC-G069 and AARC-G056-DRAFT); d. Identity assurance information (see the REFEDS Assurance Framework and AARC-G056-DRAFT). 8. Support for OAuth 2.0 Access Token validation is documented, including: a. Token Introspection (RFC 7662), e.g. via the introspection endpoint advertised in OpenID Provider metadata; and b. Proxied Token Introspection (AARC-G052).
Integration with Pilot Nodes	
POST-INTEGRATION TESTS These tests validate the effective operation of the AAI once the Pilot Node has been connected to the AAI federation. The post-integration tests confirm that federated authentication, authorisation, and token-based access function correctly in practice, including Single Sign-On to Node's web services and access to protected APIs using OAuth 2.0 access tokens, and enforcement of authorisation decisions based on federated claims.	

<table>
<tr><td colspan="2">Post-integration results provide evidence that the Pilot Node can reliably participate in cross-Node access scenarios within the AAI federation.</td></tr>
<tr><td>Tests to be performed
1. Verify federated authentication to a Pilot Node service via the Node's Infrastructure Proxy using OpenID Connect.
2. Verify that Infrastructure Proxies and Community AAIs (when present) process and express group membership and role information according to AARC-G069.
3. Verify access to protected APIs or services using OAuth 2.0 Access Tokens issued by a federated Authorisation Server.
a. Verify Token Introspection according to RFC 7662, ensuring that token active state and any scopes or claims relevant to authorisation are correctly evaluated.
b. Verify Proxied Token Introspection according to AARC-G052, where token validation is mediated by the Node's Infrastructure Proxy.
4. Verify that authorisation decisions at the Pilot Node correctly enforce:
a. group and role information;
b. scopes;
c. identity assurance information (where applicable).</td><td>Validation metrics
1. Successful federated login to at least one Pilot Node service via the Node's Infrastructure Proxy.
2. Group membership and role information are conveyed and interpreted according to AARC-G069:
a. The Node's Infrastructure Proxy(ies) correctly process and propagate AARC-G069 group and role entitlements.
b. The Node's Community AAI(s) (when present), correctly release AARC-G069 group and role entitlements.
3. Successful access to at least one protected API offered by the Pilot Node using an OAuth 2.0 Access Token.
a. Token validation via RFC 7662: Access tokens can be validated through the RFC 7662 Token Introspection endpoint of the Node's Infrastructure Proxy, with the active state and relevant scopes or claims correctly returned for authorisation decisions.
b. Proxied Token Introspection (AARC-G052): Where access tokens are issued by another trusted issuer, token validation is performed via Proxied Token Introspection through the Node's Infrastructure Proxy.
4. Authorisation decisions for both SSO-based and token-based access flows are consistent with claims present in tokens, UserInfo or Token Introspection responses, including
a. Group membership and role information (entitlements claim
b. Scopes (scope claim)
c. Identity assurance information eduperson_assurance claim)</td></tr>
</table>

Table 3: AAI integration

4.2. Service Catalogue

This section defines the pre-integration and post-integration tests for the Service Catalogue Core Component used by Pilot Nodes in their own infrastructure. Service Catalogue integration is evaluated against three capabilities:

- Ability to onboard resources through the Providers Portal and the API;
- Test curation/EPOT operations cycle (approve pending Providers and Resources, update and disable onboarded entities);
- Onboarded resources and changes performed in Node's Service Catalogue appear in (federated) Front Office (Marketplace) listings.

<table>
<tr><th colspan="2">Service Catalogue</th></tr>
<tr><td colspan="2">PRE-INTEGRATION TESTS
Acceptance criteria for integration with the Service Catalogue are:
Technical Readiness
• API compliance verified: Pilot demonstrates API endpoints usage, protocols, and data formats defined by the Service Catalogue API and EOSC Profiles.
• Network accessibility confirmed: Endpoints are reachable, stable, and respond within acceptable latency thresholds.
• Version alignment: Pilot uses supported software versions, libraries, and schema definitions.
• Error handling implemented: API returns meaningful error codes and messages according to the Service Catalogue API specification.
Data Readiness
• Metadata completeness: Required metadata fields are populated and validated against the EOSC Profile definitions.
• Data quality checks: No malformed, missing, or inconsistent data in sample submissions.
• Interoperability validation: Data formats match the expected standards (JSON/XML).
Functional Readiness
• Core use-cases validated: Pilot demonstrates that it can perform the minimum required workflows (onboarding/curation) end-to-end.
• Pilot's EOSC Service Catalogue contains fully populated entries for all Profiles.</td></tr>
<tr><td>Tests to be performed
1. Test APIs for searching, listing and CRUD operations on onboarded Providers and Resources
2. Test/Verify onboarding through API and Providers Portal
3. Test onboarding of Profiles with missing mandatory fields</td><td>Validation metrics
1. Onboarding Provider/Resource through API appears in Providers Portal listing
2. Onboarding through Providers Portal successful
3. CRUD API calls return 200
4. CRUD API calls for specific (EPOT/curation) operations need elevated permissions.</td></tr>
</table>

4. Test curation/EPOT operations cycle (approve pending Providers and Resources, update and disable onboarded entities) 5. Check controlled vocabulary population entries.	5. Cross-check API and UI controlled vocabularies are in place containing the same items.
Integration with Pilot Nodes	
POST-INTEGRATION TESTS Pilots successfully integrated with the Service Catalogue include CESSDA and NI4OS. Both Nodes have demonstrated onboarding of their resources, with CESSDA also including an external Catalogue.	
Tests to be performed 1. Onboarding/updating a Provider/Resource through Providers Portal/API is propagated to a Front Office that uses the Node's Service Catalogue as a source of data.	Validation metrics 1. Onboarded resources and changes performed in Node's Service Catalogue appear in (federated) Front Office listings.

Table 4: Service Catalogue integration

4.3. Research Product Catalogue

The objective of the integration with the Research Product Catalogue is to make the research products of the Pilot Node available via the EOSC Discovery Hub. The Pilot Node must offer bibliographic metadata about its research products via one or more data sources compliant with the **EOSC IF Guidelines for Data Sources to onboard Research Products**. The integration path is supported by the **OpenAIRE PROVIDE Dashboard** and EOSC Beyond PROVIDE as-a-service, which offers functionalities for validation and registration of data sources in OpenAIRE.

Research Product Catalogue	
One or more data sources compliant with the **OpenAIRE Guidelines**. Requirements: ● Node is in the EOSC Node Registry with a PID ● Data source is registered in the EOSC Resource Catalogue with an assigned PID ● Data source is registered in **OpenAIRE EOSC Beyond PROVIDE** as-a-service, validated, and aggregated by OpenAIRE	
Tests to be performed 1. Search for research outputs of a Pilot Node in the Discovery Hub	Validation metrics 1. At least one data source and one research product (e.g. publication, dataset, software, other) are returned by the search

Integration with Pilot Nodes	
The CESSDA Pilot Node is successfully integrated with the Research Product Catalogue. The CESSDA Data Catalogue is registered, validated, aggregated in OpenAIRE and its research products are discoverable via the Discovery Hub.	
Tests to be performed 1. **Search for research outputs of CESSDA in Discovery Hub**	Validation metrics 1. The result list includes one data source (CESSDA Data Catalogue) and research products

Table 5: Research Product Catalogue

4.4. Front Office

This section defines the pre-integration and post-integration tests for the Front Office Core Component used by Pilot Nodes in their own infrastructure.

Front Office integration is evaluated against the following capabilities:

- Ability to discover resources through the Front office and its API
- Completeness and validation of the end-user facing fields of the resources profiles.
- Onboarded resources and changes performed in Node's Service Catalogue appear in (federated) Front Office listings.

Front Office	
PRE-INTEGRATION TESTS These tests verify that a Pilot Node is technically and semantically ready to be exposed through the EOSC Beyond Front Office. The Pilot must provide discoverable, machine-readable metadata and accessible service or resource endpoints compliant with EOSC Beyond requirements.	
Tests to be performed 1. Verify that the Pilot Node is registered in the EOSC Node Registry with a valid identifier. 2. Verify that the Pilot exposes at least one service or research product through a catalogue API compatible with EOSC Beyond. 3. Validate availability and correctness of metadata required for discovery (title, description, provider, access conditions, URL).	Validation metrics 1. Pilot Node is present in the Node Registry with a resolvable endpoint. 2. At least one service or research product from the Pilot is successfully harvested and indexed. 3. Search query via the Front Office returns the Pilot's resource in results. 4. Metadata completeness score ≥ minimum required fields (100% mandatory fields populated). 5. All referenced URLs respond successfully (HTTP 200).

4. Verify that the Pilot's services/resources are indexed by the Front Office search backend. 5. Check that service/resource endpoints referenced in metadata are reachable (HTTP 200). 6. Verify basic interoperability with EOSC AAI (where applicable), ensuring access conditions are correctly described.	6. Front-Office is able to Authenticate and Authorise a user based on their AAI account. Users attributes are fetched by the Front-Office
Integration with Pilot Nodes	
POST-INTEGRATION TESTS These tests confirm that the Pilot Node has been successfully integrated and is fully operational within the Front Office, enabling end users to discover and access its services through the Front Office.	
Tests to be performed 1. Execute keyword-based search queries in the Front Office and verify visibility of Pilot services/resources. 2. Validate filtering and faceting (e.g. by provider, resource type, domain) for Pilot content. 3. Verify that service/resource detail pages load correctly and display complete metadata. 4. Verify redirection from the Front Office to the Pilot's service landing page or access URL. 5. (If applicable) Validate user access flow consistency with declared access policies (open / restricted).	Validation metrics 1. Pilot services/resources appear in Front Office search results for relevant queries. 2. Metadata displayed in the Front Office matches the source catalogue metadata. 3. End users can successfully navigate from search results to the Pilot service/resource. 4. No broken links or metadata inconsistencies detected. 5. Successful manual validation by at least one end-to-end user journey (search → view)

Table 6: Front Office Core Component

4.5. Order Management

This section defines the pre-integration and post-integration tests for the Order Management Core Component used by Pilot Nodes in their own infrastructure.

Order Management integration is evaluated against the following capabilities:

- Ability to connect external Order Management Systems to EOSC Beyond's Order Management API
- Exposure of external service offers in EOSC Beyond's OM-related UIs
- The completion of the order management life-cycle
- Connection of the EOSC Beyond's OM to the external OMSs via the provided API

Order Management	
PRE-INTEGRATION TESTS These tests verify that a Pilot Node is ready to integrate with the EOSC Beyond Order Management service. The Pilot must be able to accept, process, and manage access requests originating from the EOSC Order Management in a controlled, traceable, and policy-compliant manner.	
Tests to be performed 1. Verify that the Pilot Node defines at least one service or resource with a clear access policy (open / restricted / ordered). 2. Verify that ordering is supported via a stable service endpoint or workflow exposed by the Pilot. 3. Validate that the Pilot provides sufficient metadata to support ordering (resource identifier, provider, contact point, access conditions). 4. Verify integration readiness with EOSC AAI for identifying the requesting user. 5. Verify that the Pilot is able to receive and interpret order requests generated by the Order Management service. 6. Check that order lifecycle states (e.g. submitted, pending, approved, rejected, completed) are supported or mapped.	Validation metrics 1. Pilot Node resources are flagged as "orderable" in the system. 2. Order request metadata is complete and correctly mapped to the Pilot service. (verification for test 2 and 3) 3. Authentication context (user identity, affiliation) is successfully propagated. 4. Pilot confirms ability to handle order requests without errors (verification for tests 5 and 6).
Integration with Pilot Nodes	
POST-INTEGRATION TESTS These tests validate that the Pilot Node is fully integrated with the EOSC Beyond Order Management service and that end-to-end ordering workflows function correctly from the user perspective.	

Tests to be performed	Validation metrics
1. Submit an order request for a Pilot service/resource. 2. Verify that the order is registered and visible in the Order Management System. 3. Validate order state transitions throughout the lifecycle (submission → processing → resolution). 4. Verify that the Pilot Node receives the order request and performs the expected action (manual approval, automated provisioning, or access enablement).Validate notification and status updates visibility to the end user.	1. Orders submitted via the Front Office appear correctly in the Order Management system. 2. Order lifecycle states are correctly updated and persisted. 3. Pilot Node confirms receipt and handling of the order. 4. End users can track the status of their requests in UserSpace. 5. At least one successful end-to-end order flow completed (request → approval/provisioning → access).

Table 7: Order Management Core Component

4.6. PID

This section defines the pre-integration and post-integration tests for the PID Service Core Component used by Pilot Nodes to mint and manage PIDs for their services. PID Service integration is evaluated against three capabilities:

- Ability to mint/create persistent PIDs
- Ability to resolve the PIDs created
- Ability to update the metadata of the PID record

PID	
PRE-INTEGRATION TESTS These tests are needed to verify that a Pilot Node is ready to integrate with the EOSC Beyond PID service.	
Tests to be performed • Validate that the PID namespace exists	Validation metrics • The PID namespace exists and contains the necessary technical information about the PID resolution service(s).
Integration with Pilot Nodes	

POST-INTEGRATION TESTS These tests validate that the Pilot Node is fully integrated with the EOSC Beyond PID service and that end-to-end creation - resolution function works correctly from the user perspective. These are tests that are done after integration.	
Tests to be performed 1. Validate the PID creation 2. Validate updating metadata in the PID record 3. Validate whether the PID resolution works	Validation metrics 1. PIDs are created and persisted in the PID service 2. Changes in the PID record are correctly stored. 3. The PID resolution works correctly on an existing PID record.

Table 8: PID Service Core Component

4.7. Service Monitoring

This section defines the pre-integration and post-integration tests for the Service Monitoring Core Component used by Pilot Nodes to monitor their own infrastructure and services. Service Monitoring integration is evaluated against three capabilities:

- Ability to collect and publish Availability and Reliability results for the services of a Node.
- Ability to collect and publish Status results for the services of a Node.
- Ability to define monitoring Profiles.

Service Monitoring
PRE-INTEGRATION TESTS These tests verify that a Pilot Node is ready to integrate with the EOSC Beyond Service Monitoring. The Pilot must either have a resource or service catalogue already integrated in the federation and request a tenant to be created for the pilot or onboard its services in the main EOSC Beyond Service Registry or Topology Tool and request a dedicated report to be created. Monitoring requires the following information source to be available. • Topology Source It can either be the Nodes Service Catalogue or a different topology source (CMDB). • Profiles[1] that define how to treat each endpoint listed in the topology source. They are: a. Metric profiles b. Aggregation profiles c. Thresholds profiles d. Operations profiles • Metrics defined for each type of service.

[1] https://argoeu.github.io/argo-monitoring/docs/category/profiles

Tests to be performed 1. Validate the topology Source 2. Validate the Profiles and Metrics are defined.	Validation metrics 1. Validate that the Topology lists at least one endpoint 2. Validate that the Profiles and Metrics are defined for all Metrics
Integration with Pilot Nodes	
These tests confirm the successful integration of the Pilot Node and its full operational status within Service Monitoring. This allows end users to access Availability and Reliability, as well as Status results for the services provided by the Node through the Service Monitoring Dashboard. Currently, MetroFood and ENES are integrated with the Monitoring service by adding their topology data as part of the EOSC Beyond Topology. They've requested a separate report for these services.	
Tests to be performed: 1. Validate that the web-api serves Availability & Reliability and status results for the Node. 2. Validate that the Node report is available on the UI.	Validation metrics 1. Web-API serves results for the Node 2. The Node Report is available in the UI.

Table 9: Service Monitoring Core Component

4.8. Service Accounting

This section outlines the pre-integration and post-integration tests for the Service Accounting Core Component. Pilot Nodes use this component to collect accounting data for their own services. The integration of Accounting for Services is assessed based on its ability to centrally publish and retrieve accounting data for any service on the Node.

Service Accounting	
PRE-INTEGRATION TESTS These tests verify that a Pilot Node is ready to integrate with the EOSC Beyond Accounting for Services. The Pilot must have a resource or service catalogue already integrated in the federation and request a project to be created for the pilot in the EOSC Beyond Accounting for Services system. Accounting requires the following information sources to be available. ● Node AAI (as part of the federation) ● Resource Catalogue (as part of the federation) ● EOSC Messaging events topic (optional to sync the topology)	
Tests to be performed 1. Validate that the Node's AAI is available	Validation metrics 1. Node's AAI endpoints are alive and responding. 2. The Service catalogue returns at least one service.

2. Validate that the Node's service catalogue is available.	
Integration with Pilot Nodes	
POST-INTEGRATION TESTS These tests confirm the successful integration of the Pilot Node and its full operational status within Accounting for Services. This allows end users to access Accounting results for the services provided by the Node through the Accounting Dashboard and API.	
Tests to be performed: 1. Validate that the Node's services publish accounting data to the system. 2. Validate that the api serves results for the Node. 3. Validate the existence of accounting data for the services of the Node on the dashboard.	Validation metrics 1. Nodes' services accounting data is available for the Accounting Service and the system can process it accordingly (data is available in the right format etc.) 2. API serves results for the Node's services 3. The Dashboard presents accounting data for the services of the Node.

Table 10: Service Accounting Core Component

4.9. Research Product Accounting

By integrating with the Research Product Accounting Service, a Pilot Node contributes to the monitoring of downloads and views of research products. Data sources of a Pilot Node that are already integrated with the EOSC Beyond **Research Product Catalogue** can also integrate with this service. The data source will communicate the downloads and views of its products to EOSC Beyond Accounting for Research Products (**OpenAIRE's UsageCounts service**). The UsageCounts service aggregates the information, leveraging on the duplicate detection mechanism of the OpenAIRE Research Graph, and delivers usage track reports compliant with the Counter Code of Practice v5.0 that include not only the views and downloads of the products via the data source itself, but also the views and downloads reported by other repositories that host other manifestations of the same research products.

The integration path is supported by the **OpenAIRE EOSC Beyond PROVIDE as-a-service Dashboard** and adapters for different repository platforms: **DSpace**, **ePrints**, **others** (generic tracker).

Research Product Accounting
To integrate with the EOSC Beyond Research Product Accounting service, a Pilot must be integrated with the EOSC Beyond **Research Product Catalogue**.

Managers of data sources of the pilots can activate the service following the instructions on EOSC Beyond PROVIDE as-a-service (OpenAIRE PROVIDE). Adapters that can be used for integration are also available via the EOSC Beyond Discovery Hub.	
Tests to be performed 1. UsageCounts reports are available for the data sources of the Node. It can be checked by the data source manager on PROVIDE or via the Sushi-Lite API (https://usagecounts.openaire.eu/resources#apis)	Validation metrics 1. The report is not empty.
Integration with Pilot Nodes	
No Pilot Nodes are integrated with the Research Product Accounting, yet.	

Table 11: Research Product Accounting Service

4.10.Helpdesk

This section describes the pre-integration and post-integration tests for the Helpdesk Core Component used by the Pilot Nodes in their own infrastructure. This integration enables seamless issue tracking and resolution across a distributed support environment through a dedicated adapter.

The Helpdesk Integration is evaluated on its capability to ensure bidirectional synchronization of tickets and status updates between the Pilot Node Helpdesk and the EOSC Beyond Helpdesk. Additionally, it is assessed on its ability to facilitate issue exchange between some of the EOSC Core Services and the EOSC Beyond Helpdesk, as well as to enable seamless direct data flow between the two integrated Pilot Node helpdesks.

Helpdesk	
These tests verify that a Pilot Node Helpdesk is ready to integrate with the EOSC Beyond Helpdesk service. The pre-conditions for a Pilot Node prior to integration are: 1) An operational Pilot Node helpdesk 2) A functional and supported REST API in the Pilot Node helpdesk	
Tests to be performed 1. Submit a sample ticket creation and update request directly using the REST API supported by the Node helpdesk	Validation metrics 1. Tickets can be created and updated via authenticated REST API calls with correct responses and data integrity 2. The Node helpdesk is available and capable of receiving, creating, and

2. Verify that the Node helpdesk is operational and capable of receiving and tracking integrations via emails and notifications 3. Verify the available authentication methods and confirm that they are functioning correctly.	tracking integration-related tickets via email and notifications 3. The selected or preferred authentication method successfully validates authorised credentials and consistently rejects invalid or unauthorised access attempts
Integration with Pilot Nodes	
The EOSC Beyond Helpdesk has been bidirectionally integrated with the Pilot Node helpdesks of CESSDA, METROFOOD RI, Instruct-ERIC, and LifeWatch ERIC. In addition, a bidirectional Node-to-Node Helpdesk integration between CESSDA and METROFOOD RI has been implemented, enabling data flow between the two Pilot Node helpdesks. Furthermore, the EOSC Beyond Helpdesk has been integrated with EOSC Core Services, such as the EOSC Service Catalogue Portal and the EOSC Front Office.	
Tests to be performed 1. Verify ticket creation by submitting a ticket in the corresponding group from the first helpdesk. 2. Verify accurate ticket field mapping and completeness, ensuring that synchronised tickets do not miss any mandatory fields 3. Verify ticket update functionality of the synchronised ticket (e.g., add an article or change the state of the ticket in the first helpdesk) 4. Verify bidirectional ticket synchronization between the two helpdesks by updating the ticket in the second helpdesk (e.g., change the state or priority of the ticket) 5. Verify that no duplicate tickets are created in the two integrated helpdesks 6. Check for cross-Node isolation of the synchronised tickets	Validation metrics 1. Ticket is successfully created in the corresponding group of the second helpdesk 2. All mandatory ticket fields are correctly mapped and synchronised between the two helpdesks 3. Ticket updates are reflected accurately in the second helpdesk 4. Changes performed in the second helpdesk are correctly reflected in the corresponding ticket in the first helpdesk 5. Only a single ticket exists in the corresponding group of each helpdesk, with no duplicates 6. Tickets from one Pilot Node Helpdesk do not appear in the other unrelated Pilot Nodes' helpdesks 7. Ticket creation and updates are synchronised within a minimal time difference

7. Check end-to-end integration and synchronization time for ticket creation and updates between the two helpdesks	

Table 12: Helpdesk Core Component

4.11.IF Registry

This section defines the pre-integration and post-integration tests for the IF Registry Core Component used by Pilot Nodes in their own infrastructure.

IF Registry integration is evaluated against three capabilities:

- Ability to onboard Interoperability Guidelines through the Providers Portal and the API
- Ability to link Services and Adapters from the Service Catalogue to Interoperability Guidelines
- Onboarded Guidelines performed in Node's IF Registry appear in (federated) Front Office listings.

IF registry
PRE-INTEGRATION TESTS Acceptance criteria for integration with the IF Registry are: **Technical Readiness** • API compliance verified: Pilot demonstrates API endpoints usage, protocols, and data formats defined by the IF Registry API and EOSC Integration Guideline and Adapter Profiles. • Network accessibility confirmed: Endpoints are reachable, stable, and respond within acceptable latency thresholds. • Version alignment: Pilot uses supported software versions, libraries, and schema definitions. • Error handling implemented: APi returns meaningful error codes and messages according to the IF Registry API specification. **Data Readiness** • Metadata completeness: Required metadata fields are populated and validated against the EOSC Interoperability Guideline and Adapter Profile definitions. • Data quality checks: No malformed, missing, or inconsistent data in sample submissions. • Interoperability validation: Data formats match the expected standards (JSON/XML). **Functional Readiness** • Core use-cases validated: Pilot demonstrates that it can perform the minimum required workflows (onboarding/curation) end-to-end.

<table>
<tr><td colspan="2">

- Pilot's EOSC Interoperability Registry contains fully populated entries for all Interoperability Guidelines.

</td></tr>
<tr><td>Tests to be performed
1. Test APIs for searching, listing and CRUD operations on onboarded Providers and Resources
2. Test/Verify onboarding through API and Providers Portal
3. Test onboarding of Profiles with missing mandatory fields
4. Test curation/EPOT operations cycle (approve pending Providers and Resources, update and disable onboarded entities)
5. Check controlled vocabulary population entries.</td><td>Validation metrics
1. Onboarding Provider/Resource through API appears in Providers Portal listing
2. Onboarding through Providers Portal successful
3. System does not allow to submit a form without mandatory fields in place.
4. CRUD API calls return HTTP 200
5. CRUD API calls for specific (EPOT/curation) operations need elevated permissions.</td></tr>
<tr><td colspan="2">Integration with Pilot Nodes</td></tr>
<tr><td colspan="2">POST-INTEGRATION TESTS
Pilots successfully integrated with the IF Registry include CESSDA and NI4OS (as part of the Service Catalogue integration process). Although both Nodes have demonstrated onboarding of resources, no onboarding of Adapters or Interoperability Guidelines has taken place.</td></tr>
<tr><td>Tests to be performed
2. Onboarding/updating an Interoperability Guideline or Adapter through Providers Portal/API is propagated to a Front Office that uses the Node's IF Registry as a source of data.</td><td>Validation metrics
2. Onboarded Guidelines performed in Node's IF Registry appear in (federated) Front Office listings.</td></tr>
</table>

Table 13: IF Registry Core Component

4.12. Deployment Service

The EOSC Deployment Service, implemented through the Infrastructure Manager (IM), is an open-source service to deploy complex applications on multiple Cloud back-ends. It performs virtualised infrastructure provision, automatic software deployment, and lifecycle management of these applications. It supports the OASIS TOSCA (Topology and

Orchestration Specification for Cloud Applications) standard for application architecture description.

It is deployed as an API-centric server-side component as part of the EOSC Beyond Innovation Sandbox, available at **https://deploy.sandbox.eosc-beyond.eu**. The service is used by the EOSC Front Office so that users can easily list and deploy "Deployable Services" on cloud providers. A "Deployable Service" is a self-contained application or component defined in a TOSCA template that can be automatically provisioned, configured, and managed on cloud environments. These TOSCA templates can reference open datasets which are dynamically fetched from open data repositories (e.g. Zenodo) and staged in (or closer to) the provisioned virtualised infrastructure, via the EOSC Data Transfer Service and DataHugger.

Deployment Service	
These tests verify that a Pilot Node can register and deploy any Deployable Service needed to launch the infrastructures and services needed to support the scientific activities. It will require the creation of TOSCA templates to describe the required cloud topologies.	
Tests to be performed 1. At least one Deployable Service should be available through the EOSC Front Office: **https://search.userspace.sandbox.eosc-beyond.eu/search/deployable_service**	Validation metrics 1. Number of TOSCA templates created to support the scientific activities of the Pilot Nodes. 2. Number of Deployable Services made available through the EOSC Front Office.
Integration with Pilot Nodes	
The Deployment Service has been integrated with several Pilot Nodes: ● CNB-CSIC: Created a TOSCA template to deploy Scipion + SLURM with GPU support. ● ENES: Enhanced TOSCA templates to deploy the ENES platform, STAC storage system and yProv provenance tool. ● NFDI: Created a TOSCA template to deploy Jupyter Notebooks with the needed data pre-staged (see demo[2]) ● LifeWatch: Registered a Deployable Service in the EOSC Front Office that enables the deployment of the Jupyter Notebooks environment with the needed pre-staged datasets (see demo[3]). Currently available Deployable Service in the Front Office: ● LifeWatch: **jupyterhub**	

[2] https://youtu.be/UFEDu5cNQko
[3] https://youtu.be/twZRSILuWkU

<table>
<tr><td colspan="2">Currently available TOSCA templates:
NFDI: nfdi_demo.yml
Scipion: scipion.yaml
LifeWatch: jupyterhub_datamount.yml
ENES:
stac_vm.yaml
enes.yml
yProv.yml</td></tr>
<tr><td>Tests to be performed
TOSCA template available for the use case.
Deployment Service registered in the Service Catalog.
A Deployable Service is deployable by the Front Office.</td><td>Validation metrics
A use case has a designed and documented TOSCA template
A Deployment Service is discoverable in the Interoperability Registry and the Front Office
An instance of a Deployable Service is up and working after calling the function in the Front Office</td></tr>
</table>

Table 14: EOSC Deployment Service

5. Pilot Nodes test and validation

This chapter defines the means and measures to validate Pilot Nodes' integration with EOSC Beyond's core services and the integration of their exchange services with the EOSC Beyond Core Platform (the innovation sandbox). It provides the summary per Pilot Node for both manual and machine-actionable verification along with the validation metrics. In addition it outlines the plan to test and validate Pilots' scientific user stories against defined acceptance criteria.

5.1 CESSDA

The CESSDA Pilot Node, a virtual implementation within the EOSC Beyond piloting activities, uses a combination of existing CESSDA services and new EOSC Beyond federating capabilities to enhance the discoverability, accessibility, and interoperability of Social Science data across Europe.

It uses Core Services for the purposes shown below:

- AAI - Federated login to access services
- Discovery Hub - Discover relevant studies and instruments
- Helpdesk - Request support
- Research Product Catalogue - Make research outputs (publications, data, software) visible
- Service Catalogue - Onboard Exchange services
- Service Monitoring - Availability and reliability of Exchange services.

5.1.1. Node testing and validation layer

<table>
<tr><th colspan="2">CESSDA Pilot Node Core services</th></tr>
<tr><td colspan="2">The CESSDA Pilot Node is integrated with a number of core services: AAI, Discovery Hub, Helpdesk, Research Product Catalogue, Service Catalogue, Service Monitoring.</td></tr>
<tr><td>Manual tests to be performed - AAI
1. Is the CESSDA Pilot Node part of the community AAI?
2. Does a CESSDA user have SSO rights within the community?
3. Does a community user have SSO rights to CESSDA protracted resources?</td><td>Validation measure / metrics
1. The CESSDA IdP appears in the WAYF list for the community AAI.
2. A user with CESSDA credentials can use single sign-on to access a protected resource belonging to another Pilot Node.
3. A community user with non-CESSDA credentials can use single sign-on to access a protected resource belonging to the CESSDA Pilot Node.</td></tr>
</table>

Manual tests to be performed - Discovery Hub 4. Does the CESSDA Catalogue comply with the EOSC Resource Profiles 5. Are Exchange services from other networked Pilot Nodes visible in the CESSDA Discovery Hub? 6. Are CESSDA Exchange services visible in the Sandbox Discovery Hub?	**Validation metrics** 4. CESSDA resources appear in the Discovery Hub. 5. An Exchange service from another networked Pilot Node is visible in the CESSDA Discovery Hub. 6. A CESSDA Exchange service (e.g. CESSDA Data Catalogue) is visible and accessible from the Sandbox Discovery Hub.
Manual tests to be performed - Helpdesk 7. Is the CESSDA Helpdesk federated with the Sandbox Helpdesk, i.e. is a ticket created in the Sandbox Helpdesk and added to the CESSDA group visible in the CESSDA Helpdesk? 8.Is the federation bi-directional, i.e. are made in the CESSDA Helpdesk visible in the ticket in the Sandbox Helpdesk and vice versa?	**Validation metrics** 7. A ticket created in the Sandbox Helpdesk and added to the CESSDA group is visible in the CESSDA Helpdesk. 8. Updates made in the CESSDA Helpdesk are visible in the ticket in the Sandbox Helpdesk and vice versa.
Manual tests to be performed - Research Product Catalogue 9. Is the CESSDA Data Catalogue content harvested by the Research Product Catalogue?	**Validation metrics** 9. At least one Research Output is visible in the Sandbox Discovery Hub.
Manual tests to be performed - Service Catalogue 10. Does the CESSDA Service Catalogue act as a Back Office from the CESSDA Discovery Hub (and vice versa)? 11. 11.Does the CESSDA Service Catalogue act as a means to make CESSDA Exchange services visible in another networked Pilot Node's Discovery Hub?	**Validation metrics** 10. An Exchange service onboarded in the CESSDA Service Catalogue is visible in the CESSDA Discovery Hub. 11. An Exchange service onboarded in the CESSDA Service Catalogue is visible in another networked Pilot Node's Discovery Hub.

Manual tests to be performed - Service Monitoring 12. Are the CESSDA Exchange services being monitored?	**Validation metrics** 12. Availability and reliability statistics for the last 30 days are available for CESSDA Exchange Services (Data Catalogue, Vocabulary Service, Metadata Validator).
CESSDA Pilot Node Exchange Services	
CESSDA Pilot Node is bringing a list of services to the Network of Pilot Nodes and their integrations can be verified as defined below.	
Machine actionable tests to be performed - Data Catalogue 1. REST endpoint: https://datacatalogue.cessda.eu/api/DataSets/v2	**Validation metrics** 1. Run the following query: `curl -X 'GET' 'https://datacatalogue.cessda.eu/api/DataSets/v2/search?q=house&metadataLanguage=en' -H 'accept: application/json'` The response code must be 200 and the JSON-formatted results set should contain one or more entries.
Machine actionable tests to be performed - Vocabulary Service 1. REST endpoint: https://vocabularies.cessda.eu/v2	**Validation metrics** 1. Run the following query: `curl -X 'GET' 'https://vocabularies.cessda.eu/v2/codes/TopicClassification/4.0/en' -H 'accept: application/json'` The response code must be 200 and the JSON-formatted results set should contain one or more entries.
Machine actionable tests to be performed - ELSST 1. REST endpoint: https://thesauri.cessda.eu/rest/v1	**Validation metrics** 1. Run the following query: `curl -X 'GET' 'https://thesauri.cessda.eu/rest/v1/vocabularies?lang=en' -H 'accept: application/json'` The response code must be 200 and the JSON-formatted results set should contain one or more entries.
Machine actionable tests to be performed - Data Archiving Guide 1. Endpoint: https://dag.cessda.eu/	**Validation metrics** 1. Detect the keywords 'Data Archiving Guide' using an external monitoring tool. This shows that the service is available and the content is present.

Machine actionable tests to be performed - Data Management Expert Guide 1. Endpoint: https://dmeg.cessda.eu/	**Validation metrics** 1. Detect the keywords 'Data Management Expert Guide' using an external monitoring tool. This shows that the service is available and the content is present.

Table 15: CESSDA - Node testing and validation layer

5.1.2. Scientific user stories acceptance criteria

User stories	
CESSDA-SUS1: As a Researcher (Social Scientist) I want to use the same instrument/scale for my research that was used in other similar research projects so that I can easily make comparisons with other data later. **Core Services used**: AAI, Discovery Hub **User journey**: Search the federated Discovery Hub. Find results from similar research projects and take note of the instrument(s) used. Search for access to the instrument(s). Go to the ARIA catalogue and authenticate using the community AAI. Use the booking calendar to reserve a slot to use the instrument(s). Conduct the experiment and save the results. Analyse the results.	
Manual tests to be performed 1. Ensure a User can search for INSTRUCT ERIC services in the federated Discovery Hub. 2. Ensure a User can use SSO enabled by EOSC AAI to apply for access to restricted resources in ARIA catalogue.	**Validation metrics** 1. Researcher is able to find and book access to instrument
CESSDA-SUS2: As a Researcher (Social Scientist) I need access to data on women's political behaviour in Greece so that I can complete my report. **Core Services used**: AAI, Discovery Hub **User journey**: The User searches the federated Discovery Hub to try to find the data. The User finds some study descriptions in the federated CESSDA Data Catalogue and selects the studies of interest. The User wants access to data, so follows the link to the data publisher (EKKE). The User can download any Open data, but must apply for access to the Restricted datasets. They can use their existing credentials, as the EKKE catalogue is federated to the EOSC AAI, via the CESSDA IdP.	

Manual tests to be performed	**Validation metrics**
1. Ensure a User can use SSO enabled by EOSC AAI to apply for access to restricted resources in any CESSDA Service Providers' data catalogues that are federated via the CESSDA Pilot Node. 2. Ensure a User can search for CESSDA services in the federated Discovery Hub	1. A researcher is able to find relevant studies and access and download related data files.

CESSDA-SUS3: As a Policy Analyst, I need access to relevant datasets on climate change and the green transition (e.g. emissions, energy use, environmental behaviours) so that I can support evidence-based policy design and monitoring.

Core Services used: AAI, Discovery Hub

User journey: Search the federated Discovery Hub using keywords related to climate, energy, and the green transition. Apply basic filters (e.g. by geographic region, time period, data type) to identify datasets that are relevant for assessing progress towards climate and energy targets. Inspect the metadata to understand the main variables, coverage and update frequency of each dataset. Select a small set of key datasets (e.g. on emissions, renewable energy, and public attitudes or behaviours) that can be combined to inform a policy brief. For each selected dataset, follow the links from the Discovery Hub to the provider's access page, authenticate using the community AAI, accept the applicable terms of use, and obtain access. Download or access the data in a secure analysis environment, generate summary indicators and simple visualisations, and store both the derived outputs and the analytical notes for inclusion in policy documents or communication material.

Manual tests to be performed	**Validation metrics**
1. Ensuring user access via AAI. 2. Performing basic filtering in the Discovery Hub (e.g. by geographic area, time coverage, topic) to retrieve climate- and energy-related datasets. 3. Verifying that links from the Discovery Hub correctly resolve to the providers' access pages and that any required access steps (terms acceptance, login) can be completed.	1. Successful discovery and filtering of multiple climate/green-transition- related datasets covering the relevant region and period. 2. Successful completion of access workflows for all selected datasets, followed by download or secure access to the data. 3. Ability to produce a minimal set of indicators or visual summaries from the retrieved datasets that can be reused in a policy or journalism context.

<table>
<tr><td colspan="2">CESSDA-SUS4: As a PhD Student, I need access to longitudinal survey microdata on trust in political institutions (political parties, national parliament, national government) for Hungary and Greece, so that I can construct and analyse consistent time series of institutional trust indicators across the two countries.

Core Services used: AAI, Discovery Hub

User journey: Search the federated Discovery Hub for cross-national survey datasets on political attitudes and institutional trust. Apply filters (e.g. by country and topic) to identify relevant longitudinal survey series that include Hungary and Greece. Inspect the metadata to understand which waves/years and trust variables are available. Realise that the time series for one country has gaps in certain years. Perform a new search in the Discovery Hub to identify additional, complementary survey datasets that also include trust in political institutions for those missing years. Take note of question wording and response scales to assess comparability. For all selected datasets, follow the links from the Discovery Hub to the respective providers' access pages, authenticate using the community AAI, agree to the data access conditions, and obtain the microdata. Finally, download or access the data in a secure analysis environment, harmonise variables across surveys, build the combined country–year time series, and store both the harmonised dataset and analysis scripts for further work.</td></tr>
<tr><td>Manual tests to be performed
1. Ensuring user access via EOSC AAI.

2. Performing results filtering in the Discovery Hub (e.g. by country and topic) to retrieve multiple longitudinal survey datasets on political trust.

3. Verifying that the links from the Discovery Hub correctly resolve to the dataset access pages and that access workflows (licence acceptance, terms of use) can be completed for each provider.</td><td>Validation metrics
1. Successful discovery and filtering of at least two complementary longitudinal survey sources that together provide political trust measures for Hungary and Greece over an extended period.
2. Successful completion of all access workflows, including agreement to licensing and data protection conditions, followed by download or secure access to the microdata.
3. Ability to identify comparable trust-in-institutions variables across surveys and to construct a combined, documented time series of institutional trust indicators that can be used in the PhD analysis.</td></tr>
</table>

Table 16: CESSDA - Scientific user stories acceptance criteria

5.2 CNB-CSIC

The CNB-CSIC Pilot Node serves as a specialised setup within the EOSC Beyond framework, established to respond to the challenges and possibilities linked to the Structural Biology community, more precisely, those related to cryo-electron microscopy (cryo-EM) data validation in terms of quality.

Throughout the cryo-EM data cycle, numerous steps involve data generation, beginning with sample preparation and acquisition at the cryo-EM facilities and culminating in the reconstruction of the volume map and atomic models that represent the nature of a macromolecule target. As the number of maps deposited in public databases determined by cryo-EM is quickly growing, it becomes crucial to complement cryo-EM data with quality-related data at the latest stage of the cryo-EM data cycle. Indeed, not all data have the same quality, although it is not easily discernible.

In this context, by integrating CNB-CSIC's existing software with the functionalities provided through various EOSC Core and Exchange services, either enhanced or newly developed under the EOSC Beyond umbrella, the cryo-EM Validation Report Service (VRS) will be offered to the structural biology community. This consists of a modular and open validation grading system that qualifies a structure map at six different levels depending on the information available to assess it. With this, users could upload and validate their data to a machine in the cloud where they could use the algorithms needed for its validation.

5.2.1 Node testing and validation layer

<table>
<tr><th colspan="2">Pilot Node Core services</th></tr>
<tr><td colspan="2">The CNB-CSIC Pilot Node will rely on core services of the EOSC Core Innovation Sandbox: AAI, Discovery Hub and Deployment service.</td></tr>
<tr><td>Tests to be performed
AAI
1. Verification that users authenticated through the EOSC AAI can access the Cryo-EM Validation Report Service.
Discovery Hub
2. Registration of the Cryo-EM Validation Report Service as a resource in the EOSC Beyond Sandbox Discovery Hub.
Deployment Service
3. Validation of the deployment in different infrastructures (IISAS cloud and CESNET cloud cluster environment).</td><td>Validation metrics
AAI
1. The CNB-CSIC Infrastructure Proxy is deployed and reachable at cnb-csic-proxy.pilot.eosc-beyond.eu.
Discovery Hub
2. Users can navigate from the Discovery Hub entry to the Cryo-EM Validation Report Service.
Deployment Service
3. The service operates in a single-Node deployment (IISAS cloud) and in a cluster deployment (CESNET cloud).</td></tr>
</table>

Pilot Node Exchange Services	
The CNB-CSIC Pilot Node offers the exchange services with EOSC Deployment Service.	
Tests to be performed 1. Check the VRS webpage (https://biocomp.cnb.csic.es/EMValidationService/) is up and running	Validation metrics 1. The webpage responds properly.
Tests to be performed 1. Check the AAI VRS and the domain (https://cnb-csic-proxy.pilot.eosc-beyond.eu/) works properly	Validation metrics 1. The user can login in the Discovery Hub for launching an instance of a cloud machine with Scipion.
Tests to be performed 1. Check the EOSC Deployment Service deployed the infrastructure needed and the validation is running	Validation metrics 1. The machine with Scipion and the necessary plugins for validation is up and running.
The Validation Report Service public catalogue	
Tests to be performed 1. The reports are available at https://biocomp.cnb.csic.es/EMValidationService/pub/	Validation metrics 1. An API for checking if an specific EMDB entry is validated, the way to verify is through https://biocomp.cnb.csic.es/EMValidationService/pub/?emdbID=[EMDBCODE]

Table 17: CNB-CSIC - Node testing and validation layer

5.2.2 Scientific user stories acceptance criteria

User stories	
CNB-CSIC-SUS-1: Validating cryo-EM maps and models in EOSC	
Tests to be performed 1. The structural biology researcher can access a cloud machine for uploading his/her data and validating	Validation metrics 1. The machine can be accessed.
CNB-CSIC-SUS-2: Validating exiting public cryo-EM data from EMDB and PDB	

Tests to be performed	Validation metrics
1. The validations of the existing map entries deposited at the Electron Microscopy Data Bank (EMDB) and their respective atomic models deposited at the Protein Data Bank (PDB) should be available	1. The EOSC catalogue should point to each of these entries, originally available at https://biocomp.cnb.csic.es/EMValidationService/pub/

Table 18: CNB-CSIC - Scientific user stories acceptance criteria

5.3 EGI Node

The EGI Pilot Node serves as a federated e-Infrastructure within the EOSC Beyond framework, bringing together 27 members of the EGI Federation, including national compute and storage providers, generic platform operators, and thematic service providers. It supports all scientific disciplines by offering integrated computing and data analytics services for research and innovation.

Within EOSC Beyond, the EGI Pilot Node provides dedicated capacity for transnational and cross-disciplinary access to services and tools, leveraging selected compute and data centres of the Federation. By integrating EGI's operational service portfolio — including cloud and high-throughput computing, container-based environments, workload management, data transfer and storage — with EOSC Beyond Core services (such as AAI and federation capabilities), the Pilot Node enables seamless, cross-border access to distributed computational and storage resources.

Through this integration, the EGI Pilot Node demonstrates how federated infrastructures can effectively interoperate with EOSC Beyond Core services to support international and cross-disciplinary research use cases.

5.3.1 Node testing and validation layer

Pilot Node Core services	
The EGI Pilot Node is integrated with a number of core services: AAI, Helpdesk, Service Catalogue, Service Monitoring	
Tests to be performed **AAI** • Is the EGI Pilot Node federated with the EOSC AAI?	**Validation metrics** • EGI users can use SSO to access EOSC services and resources offered by another Pilot Node. • Researchers from other scientific communities can use single sign-on to access services offered by the EGI

<table>
<tr><td></td><td>Pilot Node through the EGI Infrastructure Proxy.</td></tr>
<tr><td>Tests to be performed

Helpdesk
Is the EGI Helpdesk federated with the Sandbox Helpdesk?
Is the federation bi-directional?</td><td>Validation metrics
A ticket created in the Sandbox Helpdesk and assigned to the EGI group is visible in the EGI Helpdesk.
Any updates made in the EGI Helpdesk are visible in the ticket in the Sandbox Helpdesk and vice versa.</td></tr>
<tr><td>Tests to be performed
Service Catalogue
Does the EGI Service Catalogue act as a means to make EGI Exchange services visible in another networked Pilot Node’s Discovery Hub?</td><td>Validation metrics
An Exchange service onboarded in the EGI Service Catalogue is visible in another networked Pilot Node’s Discovery Hub.</td></tr>
<tr><td>Tests to be performed
Service Monitoring
Are the EGI Exchange services being monitored?</td><td>Validation metrics
Availability and reliability statistics for the last 30 days are available for EGI Exchange Services.</td></tr>
<tr><td colspan="2">Pilot Node Exchange Services</td></tr>
<tr><td colspan="2">The EGI Pilot Node offers the following exchange services:
EGI Check-in (AAI)
EGI Infrastructure Manager (compute)
EGI DataHub (storage)
EGI Cloud Compute (compute)</td></tr>
<tr><td>EGI CHECK-IN (AAI)
Automated tests to be performed:
Check the EGI Check-in OIDC discovery endpoint is up and running.
Check that a user can authenticate via EGI Check-in and obtain a valid access token.
Check that AAI integration from the Pilot Node domain works</td><td>Validation metrics
An API with a health endpoint (e.g., <code>https://<checkin-host>/.well-known/openid-configuration</code>) returns HTTP 200 and valid OIDC configuration.
A user can complete login and obtain tokens (access token + id token) from the OIDC token endpoint (HTTP 200), and <code>userinfo</code> returns expected</td></tr>
</table>

properly (login redirect, callback, session creation).	claims (e.g., `sub`, `email` when available). 3. A user can login to a Pilot Node protected service from `https://<pilot-Node-domain>/` using institutional credentials and access the protected resource without errors (HTTP 200 after authentication)
EGI Infrastructure Manager (compute) **Automated tests to be performed:** **1.** Check the Infrastructure Manager (IM) API is up and running. 2. Check that IM can deploy a minimal infrastructure (e.g., 1 VM) using a reference template and then tear it down.	**Validation metrics** **1.** An API health endpoint (e.g., `https://<im-host>/health`) returns HTTP 200. 2. For each deployment, an infrastructure id is created. The status can be checked via `https://<im-host>/infrastructures/<infra-id>` and reaches a terminal successful state (e.g., `configured`/`running`) within the defined timeout; teardown removes all resources.
EGI DataHub (storage) **Automated tests to be performed:** 1. Check the DataHub endpoint is up and running. 2. Check that a user can authenticate (via Check-in) and perform basic storage operations (upload/download/list/delete) on a test file. 3. Check that access control works (authorized user can read; unauthorized user is denied)	**Validation metrics** **1.** A health endpoint (e.g., `https://<datahub-host>/health`) returns HTTP 200 (or, if not available, a basic API/list endpoint returns HTTP 200). 2. A user can upload a file and retrieve it back successfully (HTTP 201/200). Integrity is verified via checksum match (e.g., MD5/SHA256 computed before upload equals checksum after download). 3. Authorized access returns HTTP 200; unauthorized access returns HTTP 401/403 for the same resource, consistently.
EGI Cloud Compute (compute) **Automated tests to be performed:** 1. Check the Cloud Compute API endpoint is up and running.	**Validation metrics** **1.** A health endpoint (e.g., `https://<cloudcompute-host>/health`) returns HTTP 200 (or an API

2. Check that a user can create a VM from a reference image/flavor and the VM becomes reachable	"list" call returns HTTP 200 and a valid response). 2. For each execution, a VM id is created. The VM status can be checked at `https://<cloudcompute-host>/instances/<vm-id>` and reaches `ACTIVE` (or equivalent) within the defined timeout; connectivity check (e.g., SSH) succeeds.

Table 19: EGI Node - Node testing and validation layer

5.3.2 Scientific user stories acceptance criteria

User stories
SUS1: Customised infrastructure for data analysis As a Researcher, I want to deploy customised compute infrastructure to analyse community datasets so that I can perform post-processing, validation and prepare results for publication. **Core Services used:** AAI, Resource Discovery Hub, Service Catalogue, Order Management, Deployment Service **Exchange Services:** EGI Cloud Compute **User journey** ● The Researcher identifies a dataset of interest in their community portal. ● The Researcher searches the federated Resource Discovery Hub and discovers EGI Cloud Compute. ● The Researcher selects the service and proceeds with the order. ● Using EOSC AAI (EGI Check-in), the Researcher authenticates via SSO. ● Through the Order Management system, the Researcher customises compute requirements (e.g., VM size, image, resources). ● The EOSC Deployment Service configures a custom template combining compute resources, datasets and (if required) a Virtual Research Environment. ● The infrastructure is deployed and the Researcher accesses the running environment to perform analysis. ● Results are generated and stored for further validation and publication.

Tests to be performed	**Validation metrics**
1. Ensure a User can search and discover EGI Cloud Compute in the federated Resource Discovery Hub. 2. Ensure a User can authenticate via EOSC AAI (EGI Check-in) using institutional credentials. 3. Ensure a User can customise compute parameters during the ordering process. 4. Ensure the Deployment Service successfully provisions a compute instance based on the selected template. 5. Ensure the deployed environment is accessible and operational for running analysis workloads.	1. Successful discovery of EGI Cloud Compute through the Resource Discovery Hub. 2. Successful authentication via EOSC AAI (SSO). 3. Technical parameters are available in the ordering UIs and passed further in the ordering process to the providers and ordering coordinators (if applicable) 4. Successful provisioning of compute infrastructure with a valid deployment ID and status reaching “running/active”. 5. Ability to execute analysis tasks within the deployed environment without infrastructure errors.

SUS2: Virtualised storage and data publication
As a Research Infrastructure facility, I want to provide virtualised storage capacity to researchers so that they can post-process experimental data and publish additional results.

Core Services used:
AAI, Resource Discovery Hub, Service Catalogue, Research Products Catalogue

Exchange Services:
EGI DataHub

User journey

- After collecting experimental data, the Researcher needs online storage for further analysis.
- The Researcher searches the federated Resource Discovery Hub and discovers EGI DataHub.
- The Researcher authenticates using EOSC AAI (EGI Check-in).
- Through the ordering process, a request is triggered (via EGI Helpdesk) to provision a dedicated virtual storage space.
- The storage space is configured with appropriate access permissions.
- The Researcher accesses the storage using supported clients (e.g., command line client or community tools).
- Files are uploaded and managed in the virtualised storage space.
- Metadata is entered and DOIs are assigned where applicable.

• Research outputs are made available for publication via the Research Products Catalogue (following OAI guidelines).	
Tests to be performed 1. Ensure a User can discover EGI DataHub in the federated Resource Discovery Hub. 2. Ensure a User can authenticate via EOSC AAI (SSO) to access storage services. 3. Ensure the ordering workflow successfully triggers provisioning of a virtual storage space. 4. Ensure a User can upload, download and manage files within the allocated storage space. 5. Ensure metadata entry and DOI assignment processes can be completed. 6. Ensure research outputs can be registered and exposed via the Research Products Catalogue.	**Validation metrics** 1. Successful discovery and access to EGI DataHub via the Resource Discovery Hub. 2. Successful authentication via EOSC AAI. 3. Provisioning of a dedicated virtual storage space with confirmed access permissions. 4. Successful upload and retrieval of files without data integrity errors. 5. Successful metadata registration and DOI assignment (where applicable).

Table 20: EGI Node - Scientific user stories acceptance criteria

5.4 Instruct ERIC

The Instruct-ERIC Pilot Node aims to serve the Instruct-ERIC and structural biology community with the management of data they produce and making it FAIR. To support Instruct's researchers and facilities, the Instruct Node's management platform ARIA has been extended with storage brokering service, which provides a central point from which data storage can be provisioned from multiple providers, such as that supplied by EGI via the EOSC Polish Node. Facilities can choose to either directly interface with ARIA, or deploy Instruct FandanGO, a facility-side Laboratory Information Management System (LIMS) orchestration and data management tool, which also provides the functionality to interface with ARIA. Using these tools and services, the Instruct Node is able to store raw experimental data from Instruct research visits, which is linked to the researcher record held in ARIA. These data will then be published via EOSC Sandbox after an embargo period to reach a wider researcher base, and to make the data FAIR.

To improve federation with the wider EOSC landscape, the Instruct Node is also integrating its services with the EOSC AAI, Helpdesk, and Discovery Hub.

5.4.1 Node testing and validation layer

<table>
<tr><th colspan="2">Pilot Node Core services</th></tr>
<tr><td colspan="2">The Instruct-ERIC Pilot Node does not offer any of its own core services but is integrating with the AAI, Helpdesk and Discovery Hub core services.</td></tr>
<tr><td>AAI
Manual tests to be performed:
1. Instruct-ERIC Pilot Node is part of the federated login</td><td>Validation metrics
1. A user can access and apply for Instruct ARIA services and resources using EGI Check-in SSO as an IDP in ARIA IDSS AAI</td></tr>
<tr><td>Helpdesk
Manual tests to be performed:
1. Instruct-ERIC is part of the federated Sandbox Helpdesk
2. Helpdesk tickets arriving in the EOSC helpdesk can be picked up and continued in the Instruct helpdesk</td><td>Validation metrics
1. Tickets can be assigned to Instruct-ERIC Helpdesk under Group -> Pilot Nodes -> Instruct ERIC from Sandbox Helpdesk
2. Helpdesk tickets can arrive into the Sandbox Helpdesk via a redirection of a dedicated Instruct-ERIC email.</td></tr>
<tr><td>Monitoring
Manual tests to be performed:
1. Instruct ARIA's services are onboarded onto an EOSC monitoring platform</td><td>Validation metrics:
1. An ARGO deployment is able to successfully probe the availability of any ARIA service and return the correct service status</td></tr>
<tr><td>Service Catalogue
Manual tests to be performed:
1. Instruct-ERIC’s services are onboarded onto the EOSC resource catalogue
2. Instruct-ERIC’s resources are available on the Discovery Hub</td><td>Validation metrics
1. Instruct's services have been manually onboarded via the Provider Dashboard and can be seen on the Discovery Hub
2. Instruct services can be searched for and accessed via the EOSC Discovery Hub by any user</td></tr>
<tr><th colspan="2">Pilot Node Exchange Services</th></tr>
<tr><td colspan="2">The Instruct-ERIC Pilot Node is offering one exchange service and an adaptor:
● Instruct ARIA storage provisioning service
● FandanGO (community adaptor)</td></tr>
<tr><td>ARIA storage provisioning service</td><td>Validation metrics

Postman test pipelines succeed:</td></tr>
</table>

Automated tests to be performed: 1. The service can connect to at least one storage provider. 2. The service can validate the credentials of an ARIA user and return a list of available storage providers for their project . 3. A user can select a storage provider from the list and can retrieve valid credentials to connect to the provider. 4. Access controls are enforced to prevent a user who does not have valid credentials (e.g. ARIA account and project ID) from provisioning storage space from the provider. 5. Once embargo is lifted, a read-only public link to the data is generated. 6. The generated link is published on a public repository.	1. At least one EOSC storage provider is available when queried. 2. Given a selected provider, ARIA will provision the provider IDs, space IDs and access tokens required to connect to the EOSC Data Hub space. 3. The provisioned access tokens will only allow access to a unique directory within the EOSC Data Hub relevant to that user's project(s), and only if ARIA can prove they own that project. 4. The storage space and directory is inaccessible to anyone other than the user validated for that space by ARIA 5. The link is: a. Only generated after embargo is lifted b. Read-only (downloadable) c. Doesn't require authentication 6. OpenAIRE Graph gives a successful response upon upload of link and metadata.
Community adaptor: FandanGO **Automated tests to be performed:** 1. FandanGO can successfully connect to ARIA's storage provisioning service. 2. FandanGO can list available storage providers, and one can be selected. 3. FandanGO can request credentials to access storage on a provider from ARIA. 4. FandanGO can successfully connect to a storage provider using credentials obtained from ARIA. 5. FandanGO can upload data onto the storage provider.	**Validation metrics** **Code integration/unit tests succeed:** 1. ARIA gives a successful response and returns a valid bearer token, when a user logs in using FandanGO. 2. FandanGO presents valid storage options as provided by ARIA, given the user's credentials and project ID. 3. FandanGO can request storage provisioning for a selected storage provider from ARIA, and ARIA returns the valid credentials, including valid provider IDs, space IDs and access tokens to connect to the storage provider. 4. The storage provider gives a successful response when FandanGO connects with it using the credentials provided by ARIA.

	5. The user is able to perform actions on their files on the storage provider using FandanGO, including: a. Upload a file b. Locate their file c. Download their file d. Delete their file

Table 21: Instruct ERIC - Node testing and validation layer

5.4.2 Scientific user stories acceptance criteria

User stories	
INSTRUCT-SUS-1 A researcher visits a small Instruct facility to perform some structural biology research. Raw data is produced after the experiment and, after embargo, is then stored in a public repository which can then be accessed as future reference to other researchers.	
Automated tests to be performed: 1. A facility user can use Instruct ARIA's storage provisioning service to access information regarding an EOSC exchange storage provider service (e.g. EOSC Data Hub) and request relevant connection details. 2. A facility user can use Instruct FandanGO to connect to ARIA's storage provisioning service, select a EOSC exchange storage provider service (EOSC Data Hub), and set up a connection. 3. A facility user can use FandanGO to perform actions on their data stored in an EOSC exchange storage provider service, such as upload and download. 4. After embargo, a valid public link is successfully generated to access the stored data. 5. The public link is successfully published to a public repository	**Validation metrics** 1. Postman test pipelines succeed: ARIA will provide the provider IDs, space IDs and access tokens required to connect to the EOSC Data Hub. 2. Integration tests succeed: The facility user is able to successfully use FandanGO to provision data storage from the EOSC Data Hub and set up a local connection using the provisioned credentials. 3. Integration test scripts succeed: The user can upload their data to the EOSC Data Hub using FandanGO. 4. Integration test scripts succeed: The ARIA storage provisioning service successfully creates a read-only public link from the EOSC Data Hub after the embargo period has expired 5. Integration test scripts succeed: The ARIA storage provisioning service successfully publishes the public link to the OpenAIRE Graph as a Research Output, after embargo.

(e.g. OpenAIRE metadata graph) alongside relevant metadata. **Manual tests to be performed:** 6. Another user can find the public link on OpenAIRE and access the stored data.	6. Manual test succeeds: Published data can be seen on the OpenAIRE Graph and the link can be used to access the stored data in the EOSC Data Hub.

Table 22: Instruct ERIC - Scientific user stories acceptance criteria

5.5 E-INFRA CZ

e-INFRA CZ is a general-purpose computing infrastructure exposing multiple compute, storage and platform service endpoints to users federated through EOSC Beyond. It integrates with:

- AAI - Federated login to access services
- Discovery Hub - Discover relevant studies and instruments
- Service Catalogue - Onboard Exchange services
- Service Monitoring - Availability and reliability of Exchange services.

Among the host of services the following exchange services are given greater emphasis in the context of EOSC beyond, catering to the defined user stories:

- IaaS Cloud
- Managed Interactive Notebooks Platform
- National Repository Platform

5.5.1 Node testing and validation layer

The e-INFRA CZ Node consists of a number of services with different levels of priority, uptake and volume of usage. It is a mature infrastructure, in operation for years. It is subject to, and certified for, formal processes for IT Service Management (following mainly the FitSM framework) and Information Security Management (ISMS -- ISO/IEC 27001:2023).

Testing of services in the Node is highly automated, and exhaustive. At the time of this writing, the automated testing framework registers 115,342 distinct tests performed in an automated fashion on 2,222 hosts (not including the supercomputing branch of the infrastructure, which adds further to the overall total). Hence it is beyond the scope of this Deliverable to give a detailed listing of tests and services performed. However, several Core and exchange services are promoted in EOSC Beyond as a primary focus for integration: namely the IaaS cloud and a JupyterHub platform service in the Exchange category, and AAI, Helpdesk and Service catalogue in the Core category. Specific validation criteria can be given for those:

<table>
<tr><th colspan="2">Pilot Node Core services</th></tr>
<tr><td colspan="2">The EGI Pilot Node is integrated with a number of core services: AAI, Helpdesk, Service Catalogue, Service Monitoring</td></tr>
<tr><td>Tests to be performed

AAI
Is the e-INFRA CZ Pilot Node federated with the EOSC AAI?</td><td>Validation metrics
e-INFRA CZ users can use SSO to access EOSC services and resources offered by another Pilot Node.
Researchers from other scientific communities can use single sign-on to access services offered by the e-INFRA CZ Pilot Node through the EGI Infrastructure Proxy.</td></tr>
<tr><td>Tests to be performed

Helpdesk
Is the e-INFRA CZ Helpdesk federated with the Sandbox Helpdesk?
Is the federation bi-directional?</td><td>Validation metrics
A ticket created in the Sandbox Helpdesk and assigned to the e-INFRA CZ group is visible in the e-INFRA CZ Helpdesk.
Any updates made in the e-INFRA CZ Helpdesk are visible in the ticket in the Sandbox Helpdesk and vice versa.</td></tr>
<tr><td>Tests to be performed
Service Catalogue
Does the e-INFRA CZ Service Catalogue act as a means to make e-INFRA CZ Exchange services visible in another networked Pilot Node’s Discovery Hub?</td><td>Validation metrics
An Exchange service onboarded in the e-INFRA CZ Service Catalogue is visible in another networked Pilot Node’s Discovery Hub.</td></tr>
<tr><td>Tests to be performed
Service Monitoring
Are the e-INFRA CZ Exchange services being monitored?</td><td>Validation metrics
Availability and reliability statistics for the last 30 days are available for e-INFRA CZ Exchange Services.</td></tr>
<tr><th colspan="2">Pilot Node Exchange Services</th></tr>
<tr><td colspan="2">The EGI Pilot Node offers the following exchange services:
Cloud Compute (compute)</td></tr>
</table>

• JupyterHub (compute)	
Cloud Compute (compute) **Automated tests to be performed:** Check the Cloud Compute API endpoint is up and running.	**Validation metrics** A health endpoint at `https://compute.brno.openstack.cloud.e-infra.cz/` returns HTTP 200 (or an API "list" call returns HTTP 200 and a valid response).
JupyterHub (compute) **Automated tests to be performed:** Check the Cloud Compute API endpoint is up and running.	**Validation metrics** A health endpoint at `https://jupyter.e-infra.cz/hub/health` returns HTTP 200 (or an API "list" call returns HTTP 200 and a valid response).

Table 23: E-INFRA CZ - Scientific user stories acceptance criteria

5.5.2 Scientific user stories acceptance criteria

User stories enabled and promoted through EOSC beyond will be validated by a demonstration-based approach, that is, multimedia or audience-witnessed presentation of the workflow.

5.6 ENES

The ENES Pilot Node integrates ENES services with the next generation of EOSC Core capabilities to deliver an interoperable and scalable environment that enables AI-driven analysis, big data processing, and provenance tracking, fostering seamless access, innovation, and federation across scientific research domains.

5.6.1 Node testing and validation layer

Pilot Node Core services	
• **AAI** • **Helpdesk** • **PID Service**	
Tests to be performed **AAI** 1. The ENES Pilot Node is federated with the EOSC AAI.	**Validation metrics** 1. ENES users can use single sign-on to access EOSC services and resources offered by another Pilot Node.

	2. Researchers from other scientific communities can use single sign-on to access services offered by the ENES Pilot Node through the ENES Infrastructure Proxy.
Helpdesk 1. The ENES Helpdesk is federated with the Sandbox Helpdesk.	1. A ticket created in the Sandbox Helpdesk and assigned to the ENES group is visible in the ENES Helpdesk. 2. Any updates made in Sandbox Helpdesk are visible in the ticket in the ENES Helpdesk.
PID Service 1. Is the PID Service able to resolve a persistent identifier assigned to a provenance document?	1. A resolution test ran successfully on an existing PID record.
Pilot Node Exchange Services	
• **ENESLab** • **ENES Thematic Data Catalogue** • **Provenance Service** • **Provenance Explorer**	
Tests to be performed **ENESLab** 1. Access endpoint: https://ds.eneslab.pilot.eosc-beyond.eu/jupyter/hub	**Validation metrics** 1. The endpoint is reachable and the user is able to display the JupyterHub home page for performing the login to the data science environment through the EOSC AAI.
ENES Thematic Data Catalogue - STAC Browser 1. Access endpoint: https://catalogue.eneslab.pilot.eosc-beyond.eu	1. The endpoint is reachable and the user is able to display the STAC browser home page showing the content of the ENES data catalogue.
ENES Thematic Data Catalogue - STAC API 1. REST endpoint: https://api.eneslab.pilot.eosc-beyond.eu	1. Run the following query: `curl -X GET` https://api.eneslab.pilot.eosc-beyond.eu/collections The response code must be 200 and the JSON result should contain one

	or more entries (i.e., STAC collections in the ENES data catalogue). 2. Retrieve a token from https://aai-demo.egi.eu/token/ and run the following query: `curl -X POST https://api.eneslab.pilot.eosc-beyond.eu/collections/shared_collection/items -H "Content-Type: application/json" -H "Authorization: Bearer <token>" -d @stac_item.json` The response code must be 200: the STAC item is successfully added to the target collection.
Provenance Service 1. REST endpoint: http://yprov.disi.unitn.it:8004/docs	1. Run the following query: `curl -X GET http://yprov.disi.unitn.it:8004/documents` The response code must be 200 and the JSON result should contain one or more entries (i.e., provenance documents stored into the provenance service backend). 2. Retrieve a token from https://aai-demo.egi.eu/token/ and run the following query: `curl -X POST http://yprov.disi.unitn.it:8004/documents \ -H 'accept: application/json' \ -H 'Authorization: Bearer <token>'\ -H 'Content-Type: multipart/form-data' \ -F 'document_file=@prov_example.json;type=application/json'` The response code must be 200: the JSON document is successfully uploaded and a PID is assigned.

Provenance Explorer 1. Access endpoint: https://explorer.yprov.disi.unitn.it/	1. The endpoint is reachable, and the user is able to display the Provenance Explorer home page for loading and exploring a new provenance document.

Table 24: ENES - Scientific user stories acceptance criteria

5.6.2 Scientific user stories acceptance criteria

User stories	
ENES-SUS-1: Perform interactive analysis and visualisation on scientific data As a scientist, I want to get access to computing resources so that I can run interactive analysis, AI/ML-based applications, and visualisation on scientific data. **Core Services Used**: AAI, Discovery Hub **User Journey**: The scientist browses the Discovery Hub to explore the data and services offered by the ENES Pilot Node. More specifically, the scientist finds that the ENES Thematic Data Catalogue provides a set of ready-to-use, variable-centric collections, including climate projections and observations from well-known initiatives such as the Coupled Model Intercomparison Project. The scientist also discovers that ENESLab offers a comprehensive data science environment supporting interactive analysis, data visualization, and provenance tracking as well as a convenient way to query the catalogue and access the available datasets. Through the EOSC AAI, the scientist can log in to a ready-to-use Jupyter environment that provides a rich data science software stack together with the compute and storage resources required to access, analyse, and visualise the datasets of interest.	
Tests to be performed	**Validation metrics**
1. Ensure a scientist is able to discover and access the ENES data catalogue and the ENESLab through the federated Discovery Hub. 2. Ensure a scientist can log in to the ENESLab through the EOSC AAI and access the needed resources.	1. A scientist is able to browse the Discovery Hub and find the relevant ENES services. a. All ENES services onboarded to the Discovery Hub 2. A scientist is able to access the ENESLab, query the ENES catalogue and perform some analysis on the selected data. a. Successful access to the Data Science Environment

<table>
<tr><td></td><td>via EOSC AAI (100% successful login)
b. Proper access to storage and computational resources according to user roles and permissions (computational environment made available to the users in a few minutes with no manual intervention)
c. Correct execution of data analysis workflows (e.g. notebooks, scripts) using integrated datasets and delivery of the expected output products.
d. Positive feedback about usability reported by test users.</td></tr>
<tr><td colspan="2">ENES-SUS-2: Manage provenance information associated with scientific workflows
As a scientist, I want to run my analytics workflow (achieved in ENES-SUS-1) and publish and/or manage provenance information at different levels of granularity.

Core Services Used: AAI, Discovery Hub, PID Service

User Journey: From the Discovery Hub, the scientist finds that the ENES Pilot Node also offers services for provenance management and exploration.
After running an experiment in the ENESLab, the scientist gets a provenance document compliant with the W3C PROV family of standards and uploads it to the provenance service through the RESTful APIs exposed by the service. A PID is assigned to the provenance document. Authentication and authorization aspects are still handled through the EOSC AAI. The scientist can also visualise and explore the provenance graph using the provenance explorer and extract insights from the traceability documentation.</td></tr>
<tr><td>Tests to be performed

1. Ensure a scientist is able to discover and access the ENES provenance services through the federated Discovery Hub.

2. Ensure a scientist can upload a new provenance document to</td><td>Validation metrics

1. A scientist is able to browse the Discovery Hub and find the ENES provenance services.
a. All ENES services onboarded to the Discovery Hub.

2. A scientist is able to upload a provenance document using the token introspection mechanism provided through the EOSC AAI</td></tr>
</table>

the provenance service using the EOSC AAI.	a. 100% successful authenticated requests 3. A scientist can download a provenance document via the provenance service: a. 100% successful download documents. 4. A scientist is able to visualise and explore provenance documents through the provenance explorer. a. Positive feedback about usability reported by test users.

Table 25: ENES - Scientific user stories acceptance criteria

5.7 LifeWatch ERIC

LifeWatch ERIC is a European Research Infrastructure providing e-Science research facilities for investigating biodiversity and ecosystem functions. Its main focus is to enable data-driven, reproducible, and collaborative research. Within EOSC Beyond, it integrates its services into a Pilot Node to improve the interoperability, discoverability and accessibility of environmental data and services across Europe.

5.7.1 Node testing and validation layer

LifeWatch ERIC Pilot Node Core services	
The LifeWatch ERIC core services used within this user story are: • AAI - Single-Sign-On (SSO) integrated with the EGI CheckIn authentication - TRL9 • LifeWatch ERIC Metadata Catalogue – discovery of relevant resources via user interface, APIs and OAI-PMH protocol - TRL8 • HelpDesk – user support via user interface and APIs - TRL8	
Tests to be performed	**Validation metrics**
1. SSO test: verify that the LifeWatch ERIC Pilot Node is part of the EOSC AAI.	1. The EGI Check-In AAI is included in the LifeWatch ERIC SSO. A user can hence access resources in both LifeWatch and EOSC environments with the same credentials.
1. Endpoint accessibility test: verify that the LifeWatch ERIC Metadata Catalogue and	1. Catalogue and HelpDesk endpoints are accessible and responsive, returning successful responses during testing.

HelpDesk endpoints are reachable and operational.	
1. Ticket creation test: verify that support tickets can be created from both the LifeWatch ERIC environment and the federated EOSC environment (bidirectional federation).	1. Support tickets created (and updated) in the EOSC environment are successfully created and visible in the LifeWatch ERIC environment and vice versa.
1. Catalogue onboarding test: verify that the LifeWatch ERIC Metadata Catalogue is onboarded in the EOSC Discovery Hub.	1. Users can search for and access LifeWatch ERIC resources directly via EOSC Discovery Hub.
Pilot Node Exchange Services	
LifeWatch ERIC Metadata Catalogue	
Tests to be performed	**Validation metrics**
1. Catalogue's REST endpoint: https://metadatacatalogue.lifewatch.eu/doc/api/index.html	1. The LifeWatch ERIC Metadata Catalogue can be accessed via APIs in both XML and JSON formats with a 200 response code. Some examples below: a. Get all catalogue's records: `curl -X GET "https://metadatacatalogue.lifewatch.eu/srv/api/records" -H "accept: application/json"` b. Get all services: `curl -X GET "https://metadatacatalogue.lifewatch.eu/srv/api/records?from=1&hitsPerPage=100&type=service" -H "accept: application/json"` c. Get all datasets: `curl -X GET "https://metadatacatalogue.lifewatch.eu/srv/api/records?from=1&hitsPerPage=100&type=`

	`dataset" -H "accept: application/json"`

Table 26: LifeWatch ERIC - Node testing and validation layer

5.7.2 Scientific user stories acceptance criteria

User stories	
SUS-1 Analysis of alien species incidence across ecosystems: As a researcher, I want to analyse biodiversity patterns, focusing on the presence and distribution of alien species across different ecosystems. ● The LifeWatch service is fully discoverable through the EOSC Discovery Hub. ● Service metadata schema is compliant with the EOSC schema. ● Required datasets are harvested through OAI-PMH and appear in the EOSC Knowledge Graph. ● Users can authenticate via EOSC AAI using a single sign-on (SSO) flow to access both the LifeWatch ERIC Metadata Catalogue and the EGI Cloud Compute environment. ● The service is deployed and executed automatically through the Deployment Service. ● The Order Management System supports execution requests and successfully triggers automated service execution directly from the EOSC Discovery Hub interface. ● The HelpDesk integration allows support tickets to be created either directly from the EOSC environment or from the LifeWatch ERIC environment. ● The service produces consistent and reproducible results when executing it. ● External test users can complete the full service end-to-end and report positive usability feedback.	
Tests to be performed	**Validation metrics**
1. Federated discovery test: validate that the service is findable in the EOSC Discovery Hub.	1. Searchability confirmed using relevant keywords (e.g., “alien species”); metadata record accessible without errors. 2.
3. Metadata schema compliance test: validate that metadata conforms to EOSC schema requirements.	1. Automated validation checks pass for 100% of mandatory and recommended fields; compliance with OpenAIRE Guidelines guarantees automatic onboarding into EOSC via OpenAIRE.

2. Dataset harvesting test (OAI-PMH → EOSC Knowledge Graph).	1. 100% of dataset metadata harvested successfully and visible in EOSC Knowledge Graph.
2. AAI authentication test: verify SSO access.	1. 100% successful authentication across test executions; users can login in LifeWatch ERIC, EOSC Discovery Hub and EGI environment with the same credentials.
2. Deployment Service integration test: automated provisioning of the workflow environment.	1. Computational environment required for service execution created automatically in < 10 minutes with minimal technical intervention.
2. Order Management System test: trigger execution from Discovery Hub	1. Request for service execution submitted successfully; users can further monitor the request and get feedback about the status directly from the EOSC UI.
2. HelpDesk integration test: support ticket initiation and routing.	1. Ticket creation functional from both EOSC and LifeWatch environments; correct routing to LifeWatch ERIC technical team.
2. Workflow execution test: alien species incidence analysis.	1. Service runs successfully in the EGI Cloud and delivers expected outputs.
2. Reproducibility test: repeat the whole workflow, from service discoverability to execution, multiple times.	1. Stable output across multiple runs (variance ≤ 1%).
2. User validation test: 6 external scientists test the service.	1. ≥ 90% of test users complete workflow end-to-end without blocking issues and report positive usability.

Table 27: LifeWatch ERIC - Scientific user stories acceptance criteria

5.8 NFDI

The NFDI Node is operated by the German National Research Data Infrastructure and its 27 consortia. The main audience are German scientists and their collaboration partners that care for data as a common good. The NFDI Pilot Node integrates with the EOSC Beyond sandbox and partners to provide their annotated data records and services to the community.

5.8.1 Node testing and validation layer

Pilot Node Core services	
• AAI - Single-Sign-On (SSO) integrated with the EGI CheckIn authentication - TRL8 • NFDI Pilot Node service catalogue – discovery of user-accessible resources via user interface & APIs - TRL6 • HelpDesk – user support via Email & web interface - TRL8 • Node endpoint - TRL 3, to be deployed	
Tests to be performed	Validation metrics
1. SSO test: verify that the NFDI Pilot Node is part of the EOSC AAI and login is possible	1. Users can authenticate by means of the NFDI community AAI.
2. Endpoint accessibility test: verify that the NFDI Service Catalogue and HelpDesk endpoints are reachable and operational.	1. Catalogue and HelpDesk endpoints as well as websites (if applicable) are accessible and responsive, returning successful responses during testing. They are monitored and will be reported offline in case of maintenance or issues to be taken care of.
2. Ticket creation test: verify that support tickets can be created by the NFDI community	1. Support tickets can be created (and updated) in the helpdesk and are accessible to the responsible admins and respective users.
2. Catalogue onboarding test: verify that the NFDI Service Catalogue is onboarded in the EOSC Discovery Hub.	1. Users can search for and access NFDI resources directly via the EOSC Discovery Hub.
2. Catalogue's APIs test: verify that the main API endpoints are up and running.	1. The NFDI Catalogue can be accessed via APIs and the test gets a 200 response code.
2. Node endpoint: EP is available and lists Nodes' capabilities according to a schema.	1. The EP's URL is available and returns HTTP 200 together with a valid metadata document.

Table 28: NFDI - Node testing and validation layer

5.8.2 Scientific user stories acceptance criteria

User stories	
Alex Synapse, scientist, is demonstrating how they find services to use with his team for their scientific work. They first discover the Node's User Space and find a dataset with a DOI in the public-data portal after which they deploy a compute infrastructure via EGI's infrastructure manager. A JupyterHub is then started in the federated cloud and the data from the catalogue is copied and analysed there.	
Tests to be performed	Validation metrics
1. Check that the NFDI (meta)-data catalogue is discoverable and accessible	1. Find the catalogue in the EOSC discovery hub or in the NFDI Pilot Node service catalogue by searching for "Public Data", https://public-data.desy.de is online and responds with HTTP 200
2. Check that a dataset with a minted and production-level DOI is available	1. A published dataset is available on the public-data landingpage in the NFDI service catalogue and can be accessed from there (https://public-doi.desy.de)
2. Check that the EGI infrastructure manager is findable, available and that authorised access is permitted	1. The IM responds with HTTP 200 and login with the eosc-beyond VO in EGI-Checkin is successful
2. Create a new JupyterHub cluster via a TOSCA template in the IM and provide the previously selected DOI for the dataset to be downloaded	1. After the wait time, the IM presents the user with access credentials and a URL, the JuypterHub is accessible and the requested dataset is available for analysis.

Table 29: NFDI - Scientific user stories acceptance criteria

5.9 NI4OS

The NI4OS Pilot Node within the EOSC Beyond project serves as a strategic Regional Pilot Node dedicated to integrating South East Europe's research infrastructures into the wider European Open Science Cloud ecosystem. By aggregating national resources and acting as a real-world testbed, the Node validates new EOSC technologies at a regional level, ensuring that local research needs are met while accelerating the adoption of a unified, machine-composable open science environment across the continent.

5.9.1 Node testing and validation layer

The NI4OS Pilot Node is now ready for federation and has already been registered in the EOSC Beyond Node Registry. The table below outlines the core and exchange services available on the Node. All core services are built on EOSC Beyond sandbox components, which are deployed independently (outside of the sandbox). This means that the probes developed for sandbox testing can be reused for monitoring the NI4OS core service performance. Conversely, exchange services are specific to each Node, and probes for these services are expected to be created by the respective service providers.

Pilot Node Core services	
The following core services were deployed: ● AAI ● Helpdesk ● Service catalogue ● Front office	
Tests to be performed	Validation metrics
AAI 1. Login/logout	1. Users can login and be validated through the NI4OS AAI service, integrated with eduGAIN
Helpdesk 1. Create ticket 2. Track ticket status 3. Update ticket status by the service provider	1. The user can create a ticket 2. The user can track the ticket resolution progress 3. Service providers can take actions based on the ticket and update the ticket status
Service catalogue 1. API of the service catalogue operational 2. External data quality test via API	1. Number of services, providers, etc., retrieved through the API 2. Various checks of the data quality based on data retrieved via API
Front office 1. Access front office 2. Search front office 3. Access service through front office	1. User can access the front office 2. User can search the front office 3. User can access service through the front office
Pilot Node Exchange Services	

The following exchange services services were deployed: • PARADOX cluster • FINKI Cloud • IPB Code repository service • Gaussian API service • Schrödinger API service • NI4OS-Europe repository service	
Tests to be performed	**Validation metrics**
1. PARADOX cluster accessible	1. Accessing the cluster via the command line, testing the batch system, job submission, software stack, etc.
1. Pull code from IPB core repository	1. Code successfully pulled from the repository to the PARADOX cluster
1. FINKI Cloud accessible	1. FINKI Cloud is accessible through the web interface, relevant virtual machine images are available, etc.
1. Gaussian API/Schrodinger API	1. The Gaussian/Schrodinger APIs are available and functional.
1. NI4OS repository	1. The NI4OS repository is accessible via the web interface, and the dataset download and upload functionalities are operational.

Table 30: NI4OS - Node testing and validation layer

5.9.2 Scientific user stories acceptance criteria

This table aims to identify the tests and validation metrics for each step of the user story.

User stories
The main steps of the NIO4OS User Story are as follows: • Simulation development – our first step sets up a development environment where researchers can create a simulation of interest. For instance, they can use PARADOX cluster for simulations written in lower-level languages or FINKI Cloud for high-level languages. They also get access to our IPB code repository service, which allows them to commit their in-progress code and test the latest version locally. • Input dataset preparation – in this step, researchers prepare the input dataset to be used in the simulation. The requirements for this step vary depending on

<table>
<tr><td colspan="2">the type of simulation and may involve using an external service, such as Gaussian API or Schrödinger API, for input dataset creation, or retrieving the dataset from a repository like the NI4OS-Europe repository service.
• Simulation configuration preparation involves setting up the config file that will be submitted with the simulation. This file determines how the simulation is initialised. The complexity of the simulation might decide whether we generate the config file using an external service.
• Testing and benchmarking – in this step, the researcher can test the simulation in a production-like environment. A small amount of HPC or cloud resources is needed for this step.
• Production runs – once testing is complete, the researcher gains access to the HPC resource to run a full simulation and store the resulting datasets.
• Analysis of the results – in this last step of the user story, researchers access the Cloud resource from which they can retrieve datasets obtained in the previous step and analyse the results in the provided development environment.</td></tr>
<tr><td>Tests to be performed</td><td>Validation metrics</td></tr>
<tr><td>1. Compile the pulled code
2. Submit a job to the cluster</td><td>1. Successful compilation of the pulled code
2. Successful submission of a job to PARADOX cluster</td></tr>
<tr><td>1. Create a VM on the FINKI cloud
2. Start the VM on the FINKI cloud</td><td>1. Successful creation of the VM
2. Successful starting of the VM</td></tr>
<tr><td>1. Gaussian API test through
2. Schrodinger API test through</td><td>1. Test the Gaussian API with sample values through Swagger https://gaussian.chem-api.finki.ukim.mk/swagger-ui/index.html?configUrl=/api-docs/swagger-config
2. Test the Schrodinger API with sample values through Swagger https://schrodinger.chem-api.finki.ukim.mk/swagger-ui/index.html?configUrl=/api-docs/swagger-config#/gaussian-controller/GPrepRemote</td></tr>
<tr><td>1. Ni4OS repository accessible</td><td>1. Successful access to sample dataset at the repository.</td></tr>
</table>

Table 31: NI4OS - Scientific user stories acceptance criteria

5.10 METROFOOD-RI

METROFOOD-RI is a European Research Infrastructure promoting metrology in food and nutrition, providing high-level services to support research on food quality, safety, and

nutrition. It aims to enable reliable, interoperable, and reusable food data and services for the food and health research communities. Within EOSC Beyond, METROFOOD-RI contributes to the Pilot Node, integrating its domain-specific data catalogues and services with the EOSC Platform to support interoperability and accessibility of food-related research services.

5.10.1 Node testing and validation layer

<table>
<tr><th colspan="2">Pilot Node Core services</th></tr>
<tr><td colspan="2">

- AAI
- Service Catalogue
- Helpdesk
- Service Monitoring

</td></tr>
<tr><td>

Tests to be performed - AAI

1. Verify whether Instruct-ERIC is included in the federated login.
2. Verify whether a community user has SSO access to METROFOOD protected resources.

</td><td>

Validation metrics

1. Metrofood is integrated with EGI check-in.
2. A community member with Metrofood or EGI credentials can use single sign-on to gain access to a protected resource within the Metrofood Pilot Node.

</td></tr>
<tr><td>

Tests to be performed - Helpdesk

1. Check whether METROFOOD is included in the federated Sandbox Helpdesk.

</td><td>

Validation metrics

1. Tickets are created in Metrofood Helpdesk only in this moment.

</td></tr>
<tr><td>

Tests to be performed - Service Catalogue

1. REST endpoint: https://api.catalogues.eosc.metrofood-services.eu/resources
2. REST endpoint: https://catalogues.eosc.metrofood-services.eu/shop/153

</td><td>

Validation metrics

1. Run the following query: `curl -X 'GET' 'https://api.catalogues.eosc.metrofood-services.eu/resources' -H 'accept: application/json'`
2. The response code must be 200 and the JSON-formatted results set should contain one or more entries.
3. Run the following query: `curl -X 'GET' 'https://catalogues.eosc.metrofood-services.eu/shop/153'`

</td></tr>
</table>

	`-H 'accept: application/json'` The response code must be 200 and the JSON-formatted results set should contain one object entry
Tests to be performed - Service Monitoring 1. Are the Metrofood services being monitored?	**Validation metrics** 1. Availability and reliability statistics for the last 30 days are available for Metrofood catalogues: https://beyond.ui.devel.mon.argo.grnet.gr/eoscbeyond/METROFOOD
Pilot Node Exchange Services	
• Data catalogue	
Tests to be performed - Data Catalogue 1. REST endpoint: https://api.catalogues.eosc.metrofood-services.eu/resources	**Validation metrics** 1. Run the following query: `curl -X 'GET' 'https://api.catalogues.eosc.metrofood-services.eu/resources' -H 'accept: application/json'` The response code must be HTTP 200 and the JSON-formatted results set should contain one or more entries.

Table 32: METROFOOD RI - Node testing and validation layer

5.10.2 Scientific user stories acceptance criteria

User stories
SUS1: As a food chemist, I want to perform metabolomic analysis of food products to evaluate freshness, flavor, and environmental influences on quality, so that I can identify key metabolic markers, assess nutrient variations, and detect potential contaminants. **Core Services used:** AAI, Service Catalogue, Marketplace/Resource Discovery Hub, METROFOOD-RI Data Catalogue, Helpdesk **User journey:** Search the federated Resource Discovery Hub. Find METROFOOD-RI services and datasets. Go to the METROFOOD-RI Data Catalogue and authenticate using EOSC AAI. Search for access to the service and submit a request via the METROFOOD-RI Helpdesk. Search the Resource Discovery Hub for suitable

<table>
<tr><td colspan="2">compute services. Request access to EGI Cloud Compute using EOSC AAI. Run metabolic analyses and save the results</td></tr>
<tr><td>Tests to be performed
1. Ensure a user can search for and find METROFOOD-RI services and datasets in the federated Resource Discovery Hub.
2. Ensure a user can access the METROFOOD-RI Data Catalogue and authenticate using EOSC AAI.
3. Ensure an authenticated user can submit a service access request via the METROFOOD-RI Helpdesk.
4. Ensure a user can search for and identify suitable compute services in the Resource Discovery Hub.
5. Ensure a user can request and obtain access to EGI Cloud Compute using EOSC AAI.
6. Ensure a user can run analysis using the accessed data and compute resource and save results to an agreed storage or output location.</td><td>Validation metrics
1. Researcher can discover METROFOOD-RI services and datasets via the federated Resource Discovery Hub.
2. Researcher can authenticate via EOSC AAI and access the METROFOOD-RI Data Catalogue.
3. Researcher can submit a service request via the METROFOOD-RI Helpdesk and receives a request confirmation.
4. Researcher can identify a suitable compute service via the Resource Discovery Hub.
5. Researcher can gain access to EGI Cloud Compute via EOSC AAI and confirm that compute service is active.
6. Researcher can run analysis and successfully store or export the results for further interpretation.</td></tr>
<tr><td colspan="2">SUS2: As a researcher, I want to combine food composition and consumption data with socio-economic indicators across European regions, so I can better understand disparities in dietary quality and support evidence-based food policies.

Core Services used: AAI, Service Catalogue, Marketplace/Resource Discovery Hub, METROFOOD-RI Data Catalogue, Helpdesk

User journey: Search the METROFOOD-RI Data Catalogue. Authenticate using EOSC AAI. Search the federated Resource Discovery Hub for socio-economic datasets. Request access to the CESSDA Data Catalogue. Encounter interoperability issues during analysis. Submit a support request via the EOSC Helpdesk. Integrate the datasets once the alignment issues are resolved and run the analysis.</td></tr>
<tr><td>Tests to be performed
1. Ensure a user can search for and find relevant datasets in the</td><td>Validation metrics
1. Researcher can discover relevant datasets via the METROFOOD-RI Data Catalogue.</td></tr>
</table>

METROFOOD-RI Data Catalogue. 2. Ensure a user can authenticate using EOSC AAI and access the METROFOOD-RI Data Catalogue. 3. Ensure a user can search for socio-economic datasets in the federated Resource Discovery Hub. 4. Ensure a user can request and obtain access to the CESSDA Data Catalogue using EOSC AAI. 5. Ensure a user can submit a support request via the EOSC Helpdesk for cross-infrastructure alignment support. 6. Ensure the Helpdesk supports coordination between METROFOOD-RI and CESSDA support teams. 7. Ensure a user can integrate the datasets and successfully run the analysis.	2. Researcher can authenticate via EOSC AAI and access the METROFOOD-RI Data Catalogue. 3. Researcher can identify socio-economic datasets via the federated Resource Discovery Hub. 4. Researcher can request and obtain access to the CESSDA Data Catalogue. 5. Researcher can submit a support request via the EOSC Helpdesk and receive a request confirmation. 6. Dataset alignment issues are resolved via cross-infrastructure support coordination between METROFOOD-RI and CESSDA. 7. Researcher can successfully integrate datasets and run the analysis.

Table 33: METROFOOD RI - Scientific user stories acceptance criteria

6. Next Steps

Building on the testing and validation of the ten Pilot Nodes using the approach documented in this deliverable, the EOSC Beyond project can enter a critical phase of consolidation, expansion, and operationalisation. The following actions have been identified to advance the federated infrastructure toward broader deployment and sustained operation, based on the anticipated outcomes of the testing and validation phase.

6.1. Standardisation and Documentation

The testing and validation process may reveal areas where greater standardisation would benefit both existing and future Nodes. In which case the main activities should include developing comprehensive onboarding guidelines that codify the lessons learned from Pilot Node validation, establishing clearer metadata schema specifications with domain-specific extensions to accommodate diverse research communities, and creating reference implementations for common integration patterns. Technical documentation will need to be enhanced to support self-service onboarding where feasible, reducing the barrier to entry for new Pilot Nodes while maintaining quality standards.

6.2. Governance Framework Refinement

The experience with the Pilot Nodes may highlight the need for more detailed governance arrangements, particularly around service level expectations and data protection compliance across jurisdictions. If that is the case, then work should proceed to formalise the governance model, clearly defining roles and responsibilities for Pilot Node operators and core service providers. This may include establishing working groups focused on technical interoperability, user experience, and policy harmonisation to ensure that governance evolves alongside technical capabilities.

6.3. Scalability and Performance Optimisation

If the number of integrated Pilot Nodes grows, the core service integration layer must be assessed for scalability and performance under increased load. Plans should include conducting stress testing with simulated traffic patterns representing larger Pilot Node populations, optimising service discovery and metadata aggregation mechanisms, and implementing caching and load balancing strategies to maintain responsiveness. Monitoring and observability systems should be enhanced to provide early warning of performance degradation and to support proactive capacity planning.

6.4. Expansion to Additional Nodes

If the current network of Pilot Nodes demonstrates technical viability, recruitment is expected to begin for a second wave of Nodes representing additional research domains and geographical regions. Priority should be given to domains not yet represented in the Pilot Nodes network and to Nodes that can demonstrate strong user communities and

commitment to open science principles, for a plurality of information and to include more user stories and scenarios.

6.5. User Experience Enhancement

Scientific user stories validated through the Pilot Node network provide a foundation, but continuous improvement requires ongoing engagement with the research community. Planned activities could include conducting user surveys and include focus groups to identify pain points and desired features and creating user-facing documentation and tutorials that demonstrate cross-Node workflows.

6.6. Knowledge Sharing and Community Building

The success of a federated model depends on active community participation and knowledge exchange. Plans could include organising regular technical forums where Pilot Node operators can share experiences and solutions, publish case studies and success stories that demonstrate the value of integration, establish a community support platform where Pilot Nodes can seek assistance from peers, and presenting findings at international conferences to attract new participants and gather feedback from the broader research infrastructure community.

6.7. Technical Roadmap Development

Based on Pilot Node network validation and evolving user needs, a detailed technical roadmap for the rest of the projects and beyond should be developed (within the Project's scope and objectives/existing constraints) through consultation with Pilot Node operators, user representatives, and technical experts. This roadmap should guide EOSC beyond and follow up actions on prioritising enhancements to core services, identifying areas where emerging technologies such as artificial intelligence and machine learning can add value, and planning for integration of new resource types beyond those currently included.

The successful validation of the Pilot Nodes, based on the testing and validation approach documented in this deliverable will demonstrate that the federated EOSC vision is achievable. These next steps can provide a pathway from proof-of-concept to operational infrastructure, ensuring that the momentum established through the Pilot Node phase translates into lasting impact for the European research community.

Acronyms

Acronym	Meaning
AMoS	ARGO Monitoring Service
AMS	ARGO Messaging Service
BSD	Berkeley Software Distribution
CC0	Creative Commons Zero (Public Domain Dedication)
DS	Deployment Service
EOSC	European Open Science Cloud
GPL	General Public License
GUI	Graphical User Interface
HPC	High-Performance Computing
IF	Interoperability Framework
IS	Integration Suite
JupyterHub	Multi-user server for Jupyter notebooks
MIT	Massachusetts Institute of Technology (license)
OAI-PMH	Open Archives Initiative Protocol for Metadata Harvesting
Papermill	Tool for parameterising and executing Jupyter Notebooks
PID	Persistent Identifier
REST API	Representational State Transfer Application Programming Interface
SQL	Structured Query Language
TOSCA	Topology and Orchestration Specification for Cloud Applications