D4.2 PoliRuralPlus Platform Beta

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement No 101136910

Document information

Key information	Data
Project reference number	101136910
Project acronym	PoliRuralPlus
Project title	Fostering Sustainable, Balanced, Equitable, Place-based and Inclusive Development of Rural-Urban Communities' Using Specific Spatial Enhanced Attractiveness Mapping ToolBox
Work package (WP) and deliverable number and name	WP4 - D4.2 – PoliRuralPlus platform Beta
WP leader and co-leader	Asplan Viak, Czech Center for Science and Society
Deliverable leader and key author(s)	Karel Charvat (CCSS), Runar Bergheim (AV)
Contributors and authors	Karel Charvat (CCSS), Stein Runar Bergheim (AV), Frantisek Zadrazil (CCSS), Otakar Cerba (CCSS), Raul Palma (PSNC), Aron Rynkiewicz (PSNC), John O'Flaherty (MAC), Daniel Molina (SINNO), Alejo Martín Arias Filippo (SINNO)
Peer reviewers	Tuula Löytty (SML), Marieta Okenkova (SUA), O'Flaherty (MAC),
P.R. approval date/version	Draft 0.1 | 27-04-2026 Draft 0.2 | 16-05-2026 Draft 0.3 | 23-06-2026 Final 1.0 | 30-06-2026
Document type	DEM — Demonstrator, pilot, prototype
For public dissemination	PUB - Public
Brief abstract	Deliverable D4.2 documents the Beta version of the PoliRuralPlus platform. It follows D4.1 Platform Design and reports the first integrated implementation of former PoliRural components with AI/DL services, Data Spaces and EOSC-oriented infrastructures. The deliverable describes the Beta architecture, implemented services, component status, data and metadata flows, pilot testing approach, limitations and the roadmap towards the final platform release.

Disclaimer

Document history

Version	Date	Main measures/changes	Responsible
0.1	27.04.2026	Initial Beta-oriented table of contents and report skeleton prepared from D4.1 structure.	Karel Charvat (CCSS)
0.2	16.05.2026	Added Data Cubes, GeoRAG and RAG-Knowledge Graph integration sections, including implications for EOSC/Data Spaces and cross-project reuse.	Karel Charvat (CCSS)
0.3	23.06.2026	Added extensive descriptions of remaining sections	Runar Bergheim (AV)
0.4	24.06.2026	**Peer-review comments:** D4.2 is broadly compliant with the Grant Agreement and consistent with the role of D4.2 as the Beta-stage deliverable between D4.1 Platform Design and D4.3 Platform Final. For D4.3 preparation, it is recommended that the roadmap in chapter 11 is complemented with an explicit evidence collection framework ensuring that pilot validation results, Open Call feedback, corrective actions and component maturity improvements are systematically captured during the final six months.	Tuula Löytty (SML)
1.0	30.06.2026	Final version of D4.2 PoliRuralPlus platform Beta for upload to the EU portal/SYGMA	Coordinator

Table of content

Abbreviations

Acronym	Title
AI	Artificial intelligence
API	Application Programming Interface
DL	Deep Learning
DIH	Digital Innovation Hub
EOSC	European Open Science Cloud
FSTP	Financial Support for Third Parties
GeoAI	Geospatial Artificial Intelligence
H4E	Hub4Everybody/Hub/PoliRuralPlus Hub
LLM	Large Language Model
MAA	Multi-Actor Approach
MCP	Model Context Protocol
OGC	Open Geospatial Consortium
RAG	Retrieval-Augmented Generation
RAP	Regional Action Plan
SSO	Single Sign-On
TRL	Technology Readiness Level
WP	Work Package
Data Cubes	Multidimensional data structure for organising spatial, temporal and thematic variables for analysis
DGGS	Discrete Global Grid System
GeoRAG	Geospatial Retrieval-Augmented Generation
KG	Knowledge Graph
RDF	Resource Description Framework
SPARQL	SPARQL Protocol and RDF Query Language
STAC	SpatioTemporal Asset Catalog

Figures and tables

Executive Summary

D4.2 PoliRuralPlus Platform Beta documents the first integrated Beta version of the PoliRuralPlus platform. It follows D4.1 Platform Design and shifts the reporting focus from architectural intention to implemented capability. The deliverable describes the operational Beta environment, the integration of former PoliRural components with new PoliRuralPlus services, the role of AI and GeoAI tools, the organisation of data and metadata flows, pilot-facing functionality, Open Call extensions, cross-project reuse, limitations and the roadmap towards the final platform release.

The Beta platform is presented as a modular and reusable digital ecosystem for supporting rural-urban development, foresight, stakeholder engagement, territorial analysis and Regional Action Plan preparation. It is not conceived as a single monolithic application, but as a set of interoperable components that can be combined according to different user, pilot and thematic requirements. This modularity is essential for long-term sustainability, because public authorities, rural innovation actors, advisors and external developers often require selected platform functions rather than a complete deployment.

The main platform components include Hub4Everybody, Raven, JackDaw, MapWhiteboard, the Multi-Actor Approach Tool, dashboards, metadata services, spatial data services, analytical tools and AI-assisted interaction mechanisms. Raven supports data and metadata ingestion, dataset lifecycle management, semantic descriptions, tool descriptors and metadata-based selection of suitable resources. JackDaw provides a GeoAI interaction layer capable of connecting natural-language user requests with spatial data, analytical services and decision-support outputs. Collaborative tools such as MapWhiteboard and the Multi-Actor Approach Tool support stakeholder interaction, spatial co-design and structured planning processes.

The Beta implementation demonstrates progress towards a service-oriented platform architecture. Data ingestion, metadata preparation, spatial data management, AI-assisted discovery, tool orchestration and user-facing interaction are increasingly connected through APIs, standardised metadata profiles and controlled service workflows. The use of MCP-based integration strengthens the ability of the platform to connect external tools, analytical services and agent-based components without hard-wired dependencies.

Pilot use and testing confirm that the platform addresses a broad set of territorial challenges. The core pilots apply the platform in areas such as circular economy governance, participatory foresight, AKIS coordination, rural events and community planning, cross-border attractiveness mapping, cultural heritage corridors, regional KPI monitoring and agricultural digitisation. These pilots provide the primary evidence for the usefulness of the Beta platform in real rural-urban planning and decision-support contexts.

The Open Call and FSTP activities extend the validation scope by involving external developers, regional actors and specialised use cases. These activities test the platform under different territorial, linguistic, infrastructural and domain-specific conditions. They provide important feedback on usability, localisation, performance, data quality, spatial accuracy, tool integration and user interaction. The findings from Open Call projects are treated as a structured source of improvement for the final release and not only as external demonstrations.

The report also demonstrates the relevance of the PoliRuralPlus platform beyond the project itself. Cross-project cooperation with FOCAL, BioClima, INTERESH, FAIR2Adapt and KijaniSpace confirms that the platform components

can be reused in climate services, biodiversity monitoring, soil-health innovation, FAIR data governance and international capacity-building contexts. This cooperation shows that tools such as JackDaw, Raven, Hub4Everybody and SensLog can operate as reusable digital assets within a wider European and international innovation ecosystem.

Data management is a central element of the Beta platform. The report documents the role of metadata, semantic enrichment, provenance, data quality and reuse conditions. Metadata is treated not only as a documentation layer, but as an operational mechanism supporting discovery, dataset suitability assessment, tool selection, access control, transparency and AI-assisted interpretation. Provenance and quality information are considered essential for trustworthy use of platform outputs in policy, planning and advisory contexts.

The Beta release also identifies several limitations and development needs. These include the need for stronger integration between components, improved user-interface consistency, better linguistic robustness, clearer separation between internal technical processes and user-facing outputs, more systematic metadata guidance, continuous testing, improved training material and stronger support for model builders. These limitations are not treated as failures of the Beta stage, but as concrete inputs for the final development phase.

The roadmap from Beta to final platform defines the next stage as a consolidation phase. The priority is to integrate pilot and Open Call feedback into the solution, improve component interoperability, establish continuous testing procedures, prepare training materials, support model builders, develop reproducible demonstrations, embed collaborative tools into real decision processes and move mature components towards higher Technology Readiness Levels. This roadmap provides a practical bridge between technical demonstration and operational readiness.

Overall, D4.2 shows that the PoliRuralPlus Beta platform has reached a level where it can support meaningful pilot interaction, external validation and cross-project reuse. At the same time, it identifies the work still required to transform the Beta platform into a stable, better documented, more integrated and more exploitable final release. The deliverable therefore provides evidence of implementation progress and a structured basis for the final stage of platform development.

1. Introduction

This chapter explains how the platform concept described in D4.1 has been translated into the operational Beta release documented in D4.2. D4.1 defined the architectural principles, component portfolio, interoperability assumptions and initial functional requirements of the PoliRuralPlus platform. D4.2 takes this design baseline as a starting point and documents the current state of implementation, integration, testing and pilot-facing use.

The main purpose of this chapter is to clarify the transition from planned architecture to implemented capability. The focus is no longer only on what the platform should contain, but on what has already been implemented, connected, exposed to users, tested by pilots, extended through Open Call/FSTP activities and prepared for further consolidation. This distinction is important because D4.2 is a demonstrator deliverable. It must provide evidence of working components, actual workflows, operational constraints and the remaining steps towards the final release.

The Beta platform represents a significant implementation step. It brings together former PoliRural assets, new PoliRuralPlus services, data and metadata management components, GeoAI functionality, collaborative tools, stakeholder-support instruments and pilot-specific configurations. The platform is deployed as a service-oriented ecosystem rather than as a single monolithic application. This enables different user groups to access selected tools according to their operational needs, technical capacity and territorial context.

At the architectural level, the Beta release confirms the modular direction defined in D4.1. Components such as Hub4Everybody, Raven, JackDaw, the Multi-Actor Approach Tool, MapWhiteboard, spatial data services, dashboards and analytical tools are treated as reusable building blocks. They can be combined into integrated workflows, but they can also be reused separately in other projects, pilots or institutional environments. This approach is essential for rural and regional innovation, where different territories require different combinations of data, analysis, stakeholder interaction and decision-support functions.

The Beta implementation also introduces a more operational interpretation of interoperability. Interoperability is not limited to formal compliance with standards. It also concerns the capacity of components to exchange data, metadata, service descriptions, access information and analytical outputs in ways that can be used by real users and external developers. Raven plays a central role in this respect by supporting data and metadata ingestion, semantic descriptions, dataset lifecycle management and machine-actionable information needed for discovery and tool selection. JackDaw uses this information to support AI-assisted interaction and controlled access to spatial and analytical services.

The transition from D4.1 to D4.2 also changes the role of users. In D4.1, users and pilots were mainly represented through requirements, intended scenarios and design assumptions. In D4.2, they become active validators of the platform. The nine core pilots test the platform in heterogeneous regional conditions, while Open Call/FSTP projects provide additional external validation, localisation and stress-testing. Their feedback is a major source of evidence for platform improvement and must be integrated into the final development stage.

The Beta release therefore has a dual role. First, it demonstrates implemented platform capabilities for the PoliRuralPlus project. Second, it provides a basis for a reusable digital ecosystem that can support future projects, public-sector use cases, rural innovation activities, advisory services and cross-project cooperation. D4.2 documents both aspects: the current Beta implementation and the pathway towards a more mature, better integrated and more exploitable final platform.

1.1 Scope of D4.2 and relationship to D4.1

D4.2 is the implementation-oriented continuation of D4.1 Platform Design. D4.1 defined the conceptual and technical baseline of the PoliRuralPlus platform, including the platform vision, modular architecture, intended components, interoperability principles and initial requirements. D4.2 does not repeat this baseline in full. Instead, it documents how the baseline has been translated into an operational Beta release and what evidence exists regarding implementation, integration, usability and pilot relevance.

The scope of D4.2 is practical and demonstrator-oriented. It describes the implemented Beta architecture, available services, data and metadata flows, AI and GeoAI components, collaborative tools, access mechanisms, testing approach, pilot use, Open Call/FSTP extensions, cross-project reuse and the roadmap towards the final release. The report also identifies limitations and unresolved issues, because these are necessary inputs for the final development phase.

The relationship between D4.1 and D4.2 can be understood through three layers. The first layer is the design baseline inherited from D4.1. This includes the modular platform concept, the need for interoperability, the integration of former PoliRural components and the objective to support rural-urban decision-making. The second layer is the implemented Beta capability documented in D4.2. This includes operational components, deployed services, metadata and data management mechanisms, GeoAI-based interaction, collaborative functions and pilot-facing workflows. The third layer consists of remaining development priorities for the final platform release, including stronger component integration, training material, continuous testing, support for model builders and progression of selected tools towards higher Technology Readiness Levels.

This separation between design baseline, Beta implementation and final-release priorities is important for transparency. It prevents the report from mixing design intentions with implemented functionality. It also makes clear which parts of the platform are already operational, which are available in a controlled Beta form, which require further integration and which remain planned for the final stage.

D4.2 also extends the perspective of D4.1 by documenting real use and validation. The Beta platform is not assessed only as a software artefact. It is assessed as a working support environment for users involved in rural policy, planning, foresight, stakeholder engagement, data interpretation and Regional Action Plan preparation. The report therefore focuses on whether components can support practical activities, not only on whether they exist technically.

Another important element of the D4.2 scope is the treatment of feedback. Core pilots and Open Call/FSTP projects are not described only as external users of the platform. They are considered validation environments that generate requirements for correction, improvement and further integration. Pilot feedback mainly informs the suitability of the platform for territorial decision-support workflows, while Open Call/FSTP feedback provides evidence on localisation, usability, robustness, extensibility and external developer uptake.

D4.2 also positions the Beta platform as a reusable ecosystem component beyond the immediate PoliRuralPlus context. The report documents cross-project cooperation and tool reuse in related initiatives such as FOCAL, BioClima, INTERESH, FAIR2Adapt and KijaniSpace. This is relevant because long-term impact depends on whether the developed components can be reused outside the original project configuration.

The scope of D4.2 can therefore be summarised as in Table 1 below;

Table 1: Scope summary table

Scope element	Role in D4.2	D4.1 layer / baseline connection
Implementation status	Documents which components and services are available in the Beta release	Layer 1: Design baseline inherited from D4.1 — confirms which intended platform components from the D4.1 design have been implemented or made available in Beta form.
Integration status	Describes how components interact through workflows, APIs, metadata and service connections	Layer 1 → Layer 2 transition — shows how the D4.1 modular architecture and interoperability principles have been translated into operational Beta integration.
Data and metadata flows	Explains how datasets, descriptions, provenance, quality and access conditions are handled	Layer 2: Implemented Beta capability — demonstrates operational data-management and metadata mechanisms derived from the D4.1 interoperability and data-flow principles.
Pilot use	Provides evidence from core pilot regions on practical platform use	Layer 2: Implemented Beta capability — validates that the platform is not only technically implemented, but usable in pilot-facing workflows.
Open Call/FSTP extension	Documents external validation, localisation and stress-testing of platform components	Layer 2 → Layer 3 transition — uses external validation to identify improvements, localisation needs and extensibility requirements for the final release.
Decision-support relevance	Assesses how the platform supports foresight, territorial analysis, stakeholder engagement and RAP preparation	Layer 1 and Layer 2 — links the D4.1 objective of supporting rural-urban decision-making with evidence of Beta-stage use in real decision-support contexts.
Cross-project reuse	Shows how PoliRuralPlus tools are transferred to related projects and wider ecosystems	Layer 3: Final-release and sustainability priorities — demonstrates the platform's potential as a reusable ecosystem beyond the original project configuration.
Limitations and roadmap	Identifies remaining gaps and defines priorities for the final platform release	Layer 3: Remaining development priorities — defines the consolidation work required for D4.3, including stronger integration, training, continuous testing, model-builder support and higher TRL progression.

In this way, D4.2 acts as a bridge between D4.1 and the final PoliRuralPlus platform. It confirms which parts of the original design have been implemented, provides evidence of practical testing, and defines the consolidation work required before the final release.

1.2 Beta release objectives and demonstrator boundaries

The Beta release has the objective of demonstrating that the PoliRuralPlus platform can already support meaningful operational use in pilot regions and related validation environments. It is not expected to be a complete final platform. Its role is to provide an integrated, testable and improvable environment where implemented components can be accessed, evaluated and refined before the final release.

1. The first objective is to demonstrate an operational service ecosystem. The Beta platform should show that key components are not only developed individually, but can function together within coherent workflows. This includes data ingestion through Raven, metadata preparation, spatial data publication, AI-assisted interaction through JackDaw, collaborative work through MapWhiteboard and the Multi-Actor Approach Tool, and access to dashboards or analytical services where relevant. The Beta release must therefore demonstrate integration, not only component availability.
2. The second objective is to support pilot-facing use. The platform must provide practical functionality for pilot managers, regional actors, stakeholders and technical partners. This includes access to relevant data, spatial visualisation, stakeholder coordination, AI-assisted interpretation, preparation of evidence for decision-making and support for Regional Action Plans. The pilot environment is essential because it tests whether the platform can operate under real territorial, institutional and thematic conditions.
3. The third objective is to enable external validation through Open Call and FSTP activities. These activities extend the testing environment beyond the core consortium. They allow external teams to adapt platform tools, validate specific components, test usability, identify limitations and propose extensions. This is important for the Beta stage because it provides evidence on how the platform behaves when used by actors who were not involved in its original design.
4. The fourth objective is to establish a reusable and extensible technical foundation. The Beta platform should confirm that generic components can be separated from project-specific configurations. Generic components include Hub4Everybody, Raven, JackDaw, SensLog-related integration, MapWhiteboard, the Multi-Actor Approach Tool, spatial data management services and metadata mechanisms. Project-specific configurations include local datasets, pilot vocabularies, thematic workflows, regional indicators and local user-interface settings. This separation is necessary for sustainability and exploitation beyond PoliRuralPlus.
5. The fifth objective is to collect evidence for final-stage improvement. The Beta release is a controlled implementation stage where limitations are expected. These limitations must be documented and transformed into development priorities. Feedback from pilots, Open Call projects, technical partners and model builders should be used to improve component integration, user interfaces, metadata completeness, tool orchestration, training material and validation procedures.

The demonstrator boundaries of the Beta platform must also be clearly stated. Not all platform functions have the same maturity level. Some components are operational and already used in pilots. Some are available as Beta prototypes requiring guided use or technical support. Some are integrated only partially and need further consolidation. Some workflows are still experimental and should be treated as inputs for future development rather than as final operational services.

The main demonstrator boundaries as shown in table 2 below, are the following:

Table 2: Demonstrator boundaries

Boundary area	Interpretation for the Beta release
Functional completeness	The Beta release includes implemented and testable components, but not all planned final functions are complete
Integration depth	Some components are well integrated, while others still require improved workflow connections
Data readiness	Data availability depends on pilot contributions, metadata quality, access rights, licensing and technical preparation
Metadata maturity	Metadata is operationally important, but completeness and consistency still require improvement
AI reliability	JackDaw and related AI functions provide useful interaction, but require further validation, linguistic improvement and better grounding in verified sources
User interface maturity	User-facing interfaces are usable for Beta testing, but require refinement based on pilot and Open Call feedback
Performance	Some workflows may require optimisation, especially for complex spatial queries or use under weak connectivity conditions
Training and documentation	Initial support can be provided, but complete training packages and model-builder guidance are priorities for the final stage
TRL status	Tools are at different maturity levels and require explicit assessment before final release

Within these boundaries, the Beta platform represents a major implementation milestone. It confirms that the PoliRuralPlus architecture can be instantiated as a working service environment and that its components can support real pilot activities. At the same time, it provides a structured basis for continuous testing, corrective actions, training preparation and maturity progression.

The final stage must therefore build directly on the Beta evidence. Comments from pilots and Open Call/FSTP projects should be integrated into the solution. Components should be connected more tightly into end-to-end workflows. Training material should be prepared for users, technical integrators and model builders. Collaborative tools should be embedded into decision processes. Continuous testing should become part of the development cycle. The most mature tools should be prepared for movement towards higher Technology Readiness Levels.

In summary, the Beta release is not the endpoint of the platform development. It is the operational validation stage that demonstrates implemented capability, exposes limitations and provides the evidence needed to prepare a stronger final platform. Its value lies in showing what works, identifying what must be improved and creating the technical and methodological basis for a sustainable PoliRuralPlus platform ecosystem.

1.3 Relationship to WP2, WP3, WP5, WP6 and WP7

The Beta platform does not stand alone. Its implementation in WP4 depends on continuous interaction with the other work packages. These links are not only administrative. They shape the platform's relevance, technical direction, validation logic and longer-term sustainability. D4.2 therefore situates the Beta release within a broader project system where requirements, methods, testing, external uptake and business planning all influence the platform outcome. (Figure 1)

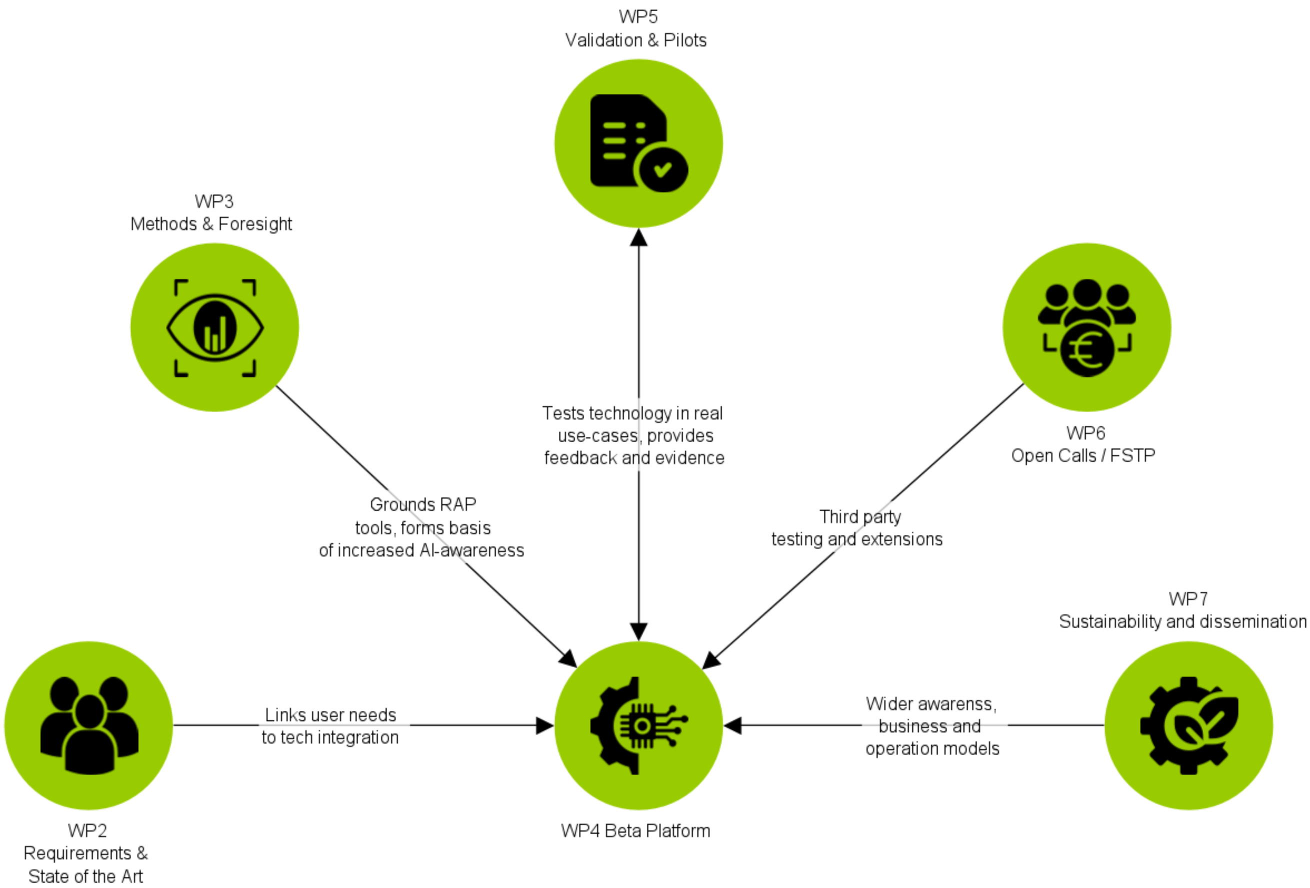

Figure 1: Cross WorkPackage Relation

WP2 is central because it connects user requirements with technical possibilities. In practical terms, WP2 helps identify what pilots and stakeholders need, and then relates these needs to the state of the art in digital tools and data services. This function is essential in a rural development context, where requirements are often diverse and not always expressed in technical language. For WP4, WP2 provides the basis for prioritising useful functions over purely experimental ones. It helps ensure that the Beta platform is shaped around actual decision-support and planning needs rather than around technology for its own sake.

WP3 has a different but equally important role. It contributes methodological frameworks, experimentation logic and substantive background for technology development. This includes support for foresight, assessment processes and other analytical tasks that the platform is meant to enable. In this way, WP3 helps define what the tools should do and how their outputs can be used meaningfully. For the Beta release, this relationship is important because implemented components need methodological grounding if they are to support Regional Action Plan development and broader rural policy processes.

WP5 is the main validation counterpart for WP4. Its role is critical because a Beta platform only becomes credible when it is tested in conditions that reflect intended use. WP5 provides the framework for testing the technology, gathering evidence on fitness for purpose, identifying weaknesses and documenting user experience. This feedback loop is necessary for moving from a technical demonstrator to a more stable and adoptable platform. In D4.2, the relationship to WP5 is therefore reflected in the way pilot access, validation arrangements and known limitations are documented.

WP6 adds an important external dimension. Through the Open Call and FSTP process, WP6 creates conditions for third parties to test, extend and build on PoliRuralPlus technology components. This is highly relevant for the Beta release because it allows the platform to be assessed not only by core project partners but also by actors with different needs and development approaches. Their contributions can expand the platform through concrete solutions and can reveal how well the building-block logic works in practice. For a modular platform, this kind of third-party engagement is a strong test of usability, interoperability and transfer potential.

WP7 is essential for long-term perspective. Even a technically strong Beta release has limited value if there is no realistic path towards sustainability, awareness and operational uptake. WP7 supports the development of business and operational models and contributes to dissemination and broader visibility of the project results. This influences WP4 in two ways. First, it encourages design and implementation choices that are maintainable beyond the project period. Second, it helps position PoliRuralPlus outputs in a wider ecosystem of users, projects and potential adopters. In this way, WP7 supports the transition from project-bound platform development to durable and reusable digital assets.

Taken together, these work package relationships show that the Beta platform is the result of coordinated project effort rather than isolated software development. WP2 anchors user needs; WP3 provides methodological substance; WP5 tests the results; WP6 opens the platform to third-party contribution and extension; and WP7 supports sustainability and uptake. D4.2 therefore presents the Beta implementation not only as a WP4 milestone, but as a shared project asset shaped by cross-work package dependencies and intended for continued use and development.

2. Beta Platform Architecture and Integration Approach

This chapter describes the Beta architecture (Figure 2) and the integration approach used to turn a set of independent tools and services into a coherent PoliRuralPlus platform environment. In the previous chapter, the focus moved from design intention to implemented capability. Here, that shift is made concrete through the structure of the running platform, the way services are connected, and the practical choices that shaped the Beta release.

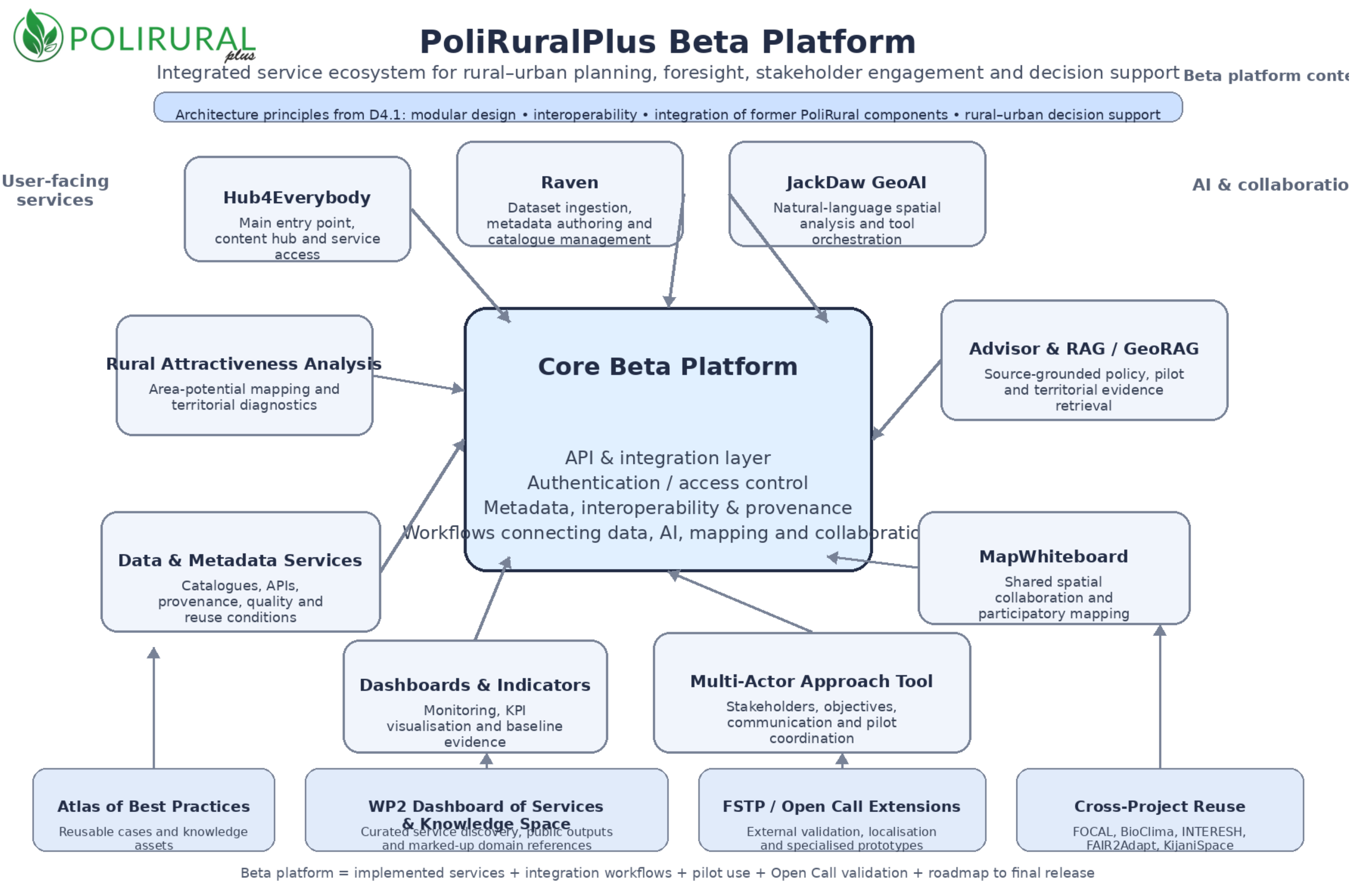

Figure 2: Beta Platform component Services

The implementation follows the modular logic already established earlier in the project, but the Beta version gives that logic a clearer operational form. The platform is not treated as one closed application. It is built as a connected environment where user-facing tools, backend services, data resources and AI-supported functions can interact through shared interfaces and controlled service flows. This approach supports both the project demonstrator and later reuse of individual parts.

A central design principle in the Beta release is that integration should not force all components into one monolithic stack. In rural and regional development settings, this would often be impractical. Different users need different functions, and the technical context can vary strongly from one use case to another. The architecture therefore aims for a balance; it provides enough common structure for a coherent user experience, while keeping modules separable and reusable.

This balance also affects how the platform is framed beyond the project itself. The full PoliRuralPlus instantiation remains important for pilots and for the integrated demonstrator. At the same time, long-term value is expected to come from building blocks that can be combined in smaller configurations. The Beta architecture has therefore been designed both as a project system and as a source of reusable technical assets.

Another important development in this release is the introduction of a more agent-oriented integration method. This has changed how tools interact with the core conversational environment and how the system can be embedded into other clients. As a result, the architecture is now more flexible than in earlier planning assumptions, and less dependent on one specific chat front end.

2.1 Overall beta architecture

The Beta architecture is organised in several layers that work together in a clear service chain. At the top, the platform contains front-end applications that users interact with directly. These include the main conversational client, selected analytical interfaces and other pilot-facing applications. Below this sits an API and service layer that coordinates requests, manages workflows and connects applications to data, tools and AI services. (Figure 2)

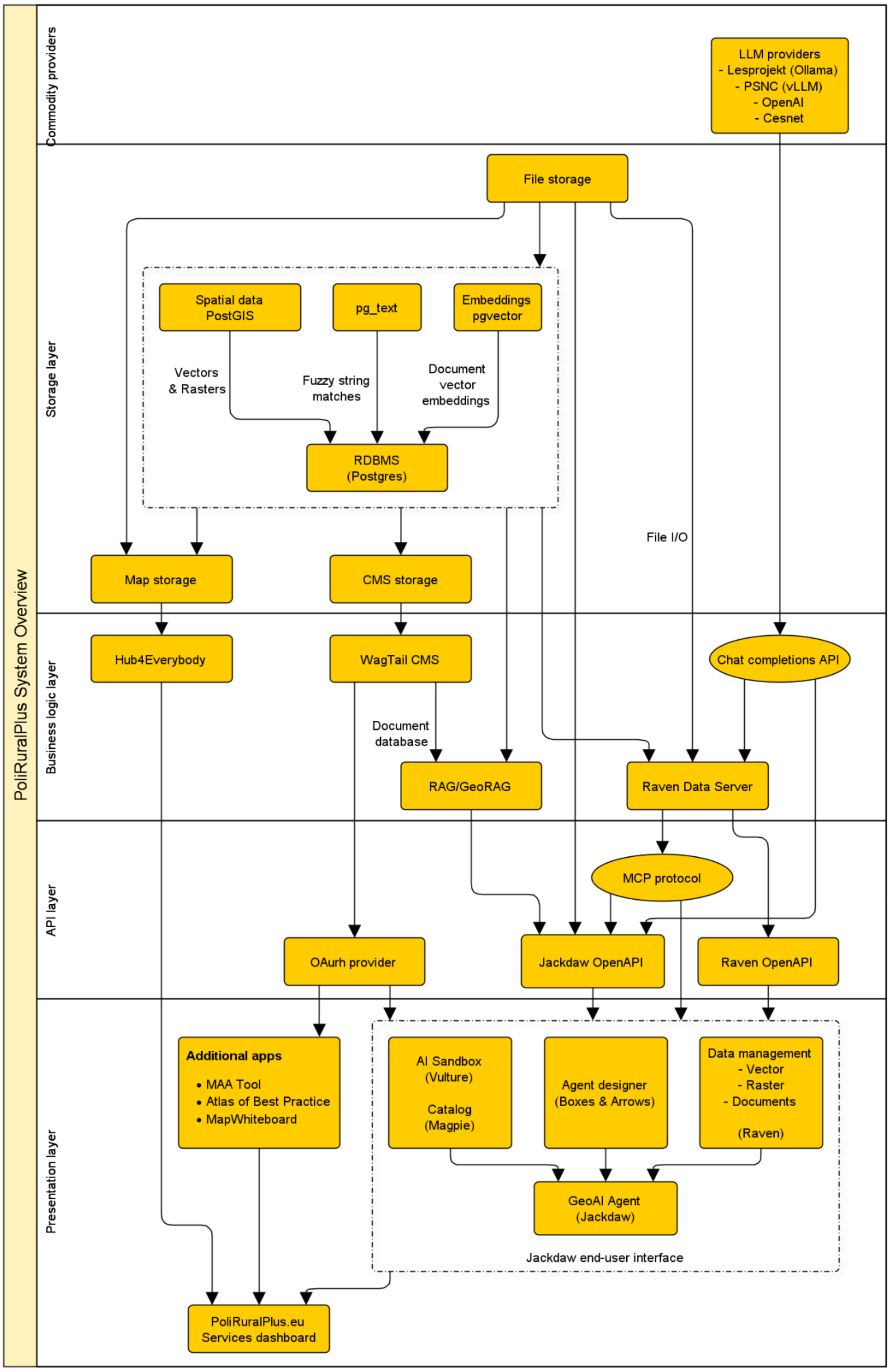

Figure 3: High level architecture showing interconnectivity between PoliRuralPlus components

Behind the API layer, the platform relies on a set of data stores and specialised services. These include structured databases, file-based resources, geospatial services and metadata-oriented cataloguing functions. Some services support standard application tasks, while others provide functions such as retrieval, geospatial processing or orchestration of AI-supported operations. This division helps keep the system manageable and allows services to evolve without forcing changes across the whole platform.

The architecture also includes an authentication layer and a deployment environment that support secure and reliable operation. Security and access management are not added as external features after the fact. They form part of the operational architecture and shape how requests reach services, how permissions are checked and how users move between modules. This creates a more stable basis for pilots and for future expansion.

A major change in the Beta release is the introduction of the MCP protocol for interactions between tools and the chat agent. This protocol was not available at the time of the earlier release planning, but it has since become important for interoperability in agent-based environments. Its inclusion changes the role of the conversational core. Instead of being tied to one dedicated client, the Jackdaw core can now act as a reusable component that can serve other chat agents and other interface contexts.

This is significant for two reasons. First, it reduces dependency on one front-end implementation and improves architectural flexibility. Second, it supports a broader market and research logic, where conversational capabilities are increasingly expected to work across tools rather than inside one fixed application. In practical terms, the Beta architecture is therefore built around a more open interaction model; user interfaces, tools and agent capabilities are connected through shared protocols rather than hard-wired dependencies.

The Beta platform also incorporates a knowledge-access and service-discovery layer developed around the WP2 Dashboard of Services, marked-up domain references, the PoliRuralPlus Advisor service and the wider PoliRuralPlus Knowledge Space. These elements connect the technical platform with the project's public knowledge outputs and make them usable as operational resources rather than static documents. The WP2 Dashboard of Services provides a structured entry point to services, tools, datasets and outputs relevant to rural-urban linkages, stakeholder analysis, territorial diagnostics and RAP development. Marked-up domain references enrich project and policy materials with thematic, spatial and methodological tags, improving discoverability and enabling more precise retrieval by AI-supported services. The Advisor service uses this curated and structured knowledge base to provide source-grounded support for policy questions, pilot analysis, foresight tasks and preparation of Regional Action Plans. Together, these components form the PoliRuralPlus Knowledge Space, which brings together all public project outputs, deliverables, pilot materials, methodological references and reusable knowledge assets. Within the Beta architecture, the Knowledge Space functions as an evidence layer for JackDaw, RAG/GeoRAG services, dashboards and pilot-facing workflows, ensuring that users can move from data and tools to contextual interpretation, documented evidence and decision-support outputs. This strengthens the role of the Beta platform as both a technical demonstrator and a knowledge infrastructure for rural-urban policy innovation.

2.2 Integration of former PoliRural components

The integration of former PoliRural components has been guided by practical usefulness rather than by the idea that every earlier asset must be reproduced in full. Some components remain directly relevant and have been reused or modernised. Others provide partial functions, data structures or interface logic that are now embedded into newer platform services. This has allowed continuity with previous work while avoiding unnecessary duplication.

In practice, the integration strategy is a balance between modularity and completeness. A fully integrated total platform makes sense in the project demonstrator, where users need to see how the parts work together in one environment. Outside that context, however, most use cases do not require the whole stack. Public authorities, regional actors and innovation intermediaries often need selected capabilities rather than a complete platform instance. The Beta architecture reflects this reality by supporting both combined use and selective reuse.

Several former PoliRural assets have therefore been connected in adapted form. This includes elements linked to the Digital Innovation Hub logic, Rural Attractiveness-related functionality and collaborative support mechanisms. In some cases, this means direct integration into the current platform. In other cases, the earlier component is no longer used as a visible stand-alone module, but its functions survive through wrapped services, tools, reusable data flows or background logic exposed through APIs.

This also means that not every earlier component has the same status in the Beta release. The MapWhiteboard, for example, is largely left out as a major stand-alone part of the current platform configuration. This does not suggest that such functionality lacks value. Rather, it reflects prioritisation decisions made during implementation, where resources were directed towards services with stronger immediate relevance for the Beta use cases and for the integrated demonstrator.

Other former capabilities have been brought into the platform through tools, ingestion pipelines and MCP-enabled interactions. This is an important point because integration is not limited to user interface composition. In many cases, the more sustainable strategy is to preserve useful functions at service level, where they can be called by other modules, used in automated workflows or made available to conversational agents. This gives the platform a more flexible inheritance from earlier PoliRural results and better aligns it with the building-block logic established for the Beta release.

2.3 AI integration layer and associated tech

The AI integration layer connects large language models, retrieval workflows and GeoAI-oriented functions to the wider platform through APIs, service wrappers and controlled execution paths. The main purpose of this layer is not to expose raw model access alone. It is to make AI functions usable within concrete platform tasks such as evidence gathering, question answering, summarisation, retrieval of relevant content and support for geographically informed analysis.

At the current stage, the offering relies in part on mainstream cloud APIs, most notably GPT services from OpenAI. These services provide strong baseline capability and are used where they offer a practical and stable path to user-facing functionality. For a Beta platform, this is an important consideration. T. The Beta release balances model independence with operational stability and pilot usability.

At the same time, the architecture does not depend only on commercial cloud provision. The Beta platform also opens for self-hosted model use through Ollama and Llama CPP, supported through CCSS infrastructure. This helps the project prepare for cases where local deployment, lower operating cost, stronger data control or experimental model selection are required. It also supports a more resilient architecture by avoiding complete reliance on one provider model.

In addition, experimental integrations are being pursued towards the LLM infrastructures of PSNC and CESNET. These connections extend the platform's technical options and support exploration of European research and infrastructure capacities. They are relevant not only for experimentation, but also for longer-term alignment with shared infrastructure approaches that may become more important as the market for AI services matures and cost conditions change.

This mixed strategy reflects a realistic assessment of the current AI landscape. Smaller local models are improving quickly and already show impressive performance on consumer-level hardware for many tasks. For this reason, edge and distributed processing are treated as credible future options rather than distant ideas. This matters because Future cost exposure associated with commercial cloud-based AI services remains uncertain and should be mitigated through self-hosted and European infrastructure options. Many commercial AI services are still operating under pricing conditions that may shift. By preparing cloud-based, self-hosted and distributed alternatives in parallel, the Beta platform reduces future risk and keeps the AI layer adaptable.

2.4 Data Spaces, EOSC and external infrastructure alignment

The Beta platform is aligned with European Data Spaces and EOSC-oriented principles in a pragmatic way (Figure 4). The objective is to make data resources discoverable, interoperable, reusable and accessible through machine-readable interfaces where possible. This includes attention to metadata, persistent access patterns and compatibility with external infrastructures that may support wider research and policy use. The aim is not only to display data in graphical interfaces, but also to make it usable in workflows driven by APIs, catalogue services and AI-assisted agents. (Figure 4).

In the Beta version, this alignment also includes preparation for Data Cube-oriented access to gridded and time-series information. This is relevant for rural and regional analysis because many policy questions depend on temporal change and spatial comparison rather than on one isolated dataset. The architecture therefore begins to support forms of data access that are better suited to repeated querying, indicator preparation and machine-supported analysis across larger territories and longer time spans.
The same logic applies to Earth observation resources. The platform is being prepared for STAC and OGC-based discovery of EO assets, alongside machine-actionable metadata that can serve GeoRAG and broader RAG workflows. This is important because AI-supported retrieval is only as useful as the metadata and access logic behind it. If catalogues, identifiers and service descriptions are weak, even advanced models cannot produce reliable support. For this reason, metadata quality and machine-readability have been treated as core integration concerns.

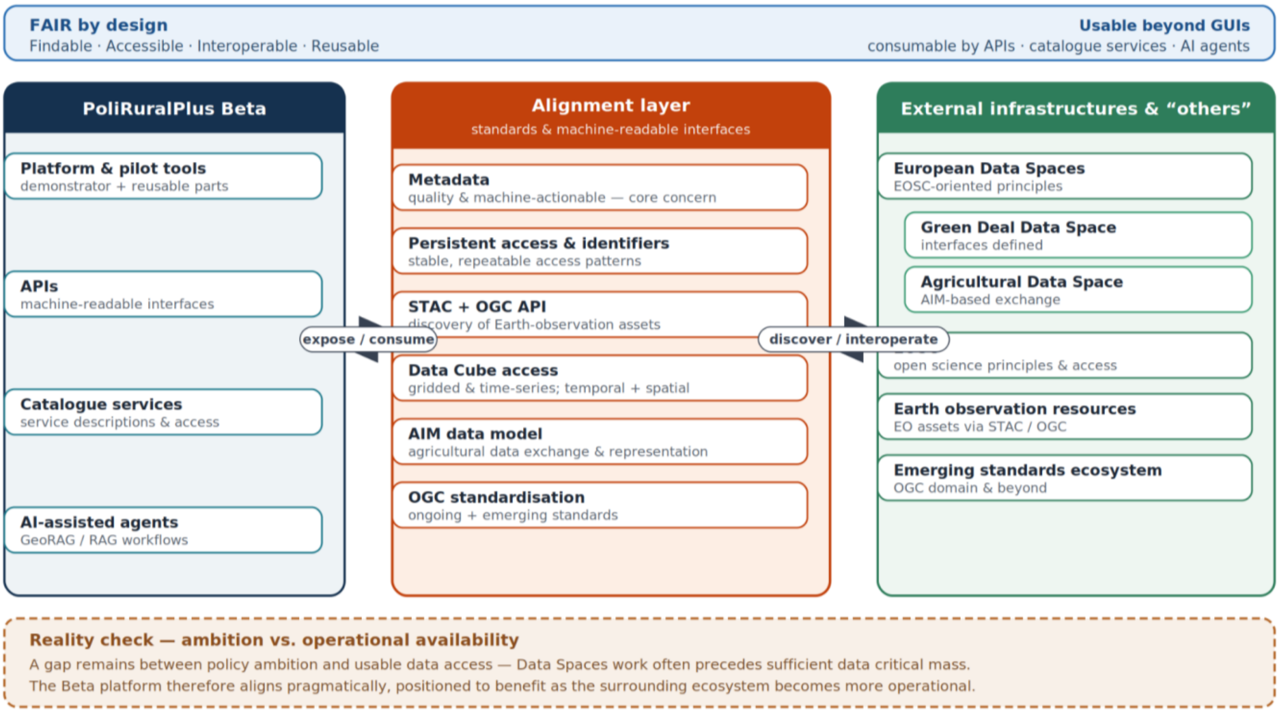

Figure 4: Data Spaces, EOSC and external infrastructure alignment

At the same time, the chapter should note a practical limitation in the current Data Spaces landscape. There remains a gap between policy ambition and operational availability. Considerable effort has gone into frameworks, governance concepts and organisational definitions, but in many cases broad and readily usable data access is still difficult to achieve. In that sense, work on Data Spaces can at times resemble an effort made before the data itself has reached sufficient critical mass. The Beta platform therefore aligns with these initiatives, but does so with a clear awareness of present constraints.

Even with these constraints, relevant interfaces have been defined towards the Green Deal and Agricultural Data Spaces. The project also uses AIM as a data model for agricultural data exchange and representation, alongside other relevant current and emerging standards. This includes ongoing standardisation work in the OGC domain. The result is not a claim of complete data-space maturity, but a practical positioning of the Beta platform so that it can work with current standards, benefit from future improvements and support more structured data exchange as the surrounding ecosystem becomes more operational.

2.5 Security, authentication and access control

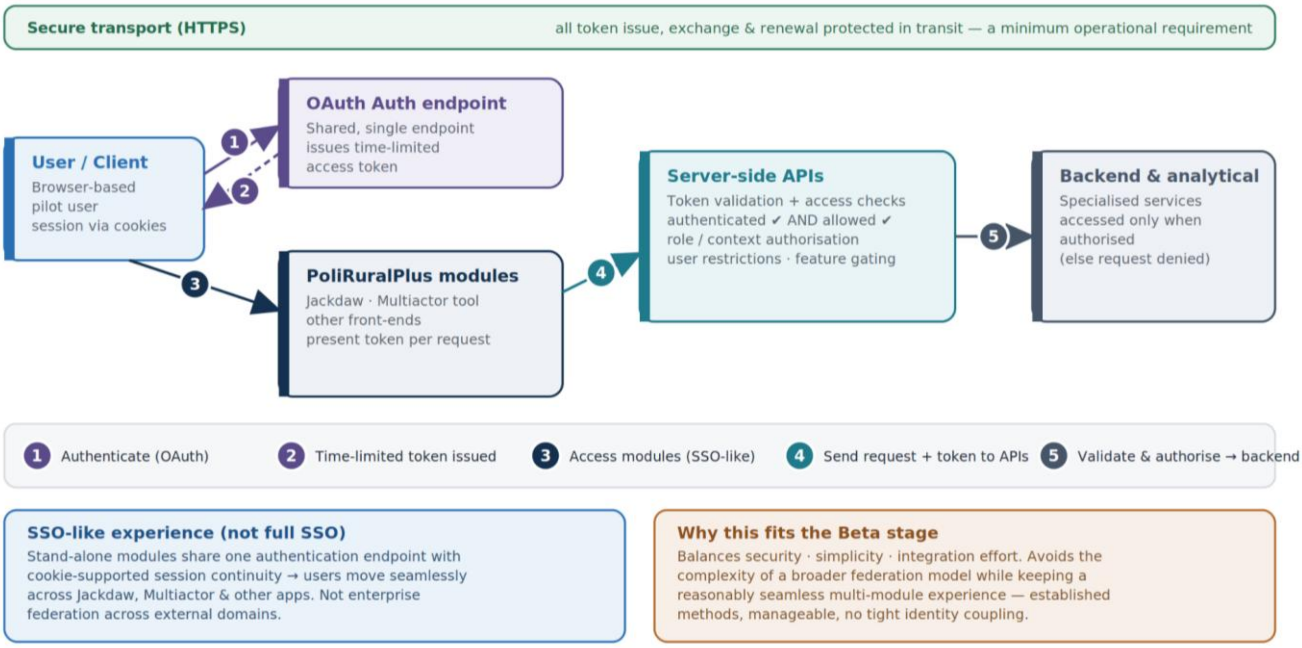

Figure 5: Security, authentication and access control

The Beta platform uses a conventional and well-understood trust model for security and access control. Authentication is handled through an OAuth-based flow, where a user obtains a time-limited access token that is then validated by server-side APIs for each relevant request. This approach supports secure communication without requiring each individual service to handle identity management in isolation. It also creates a common basis for user access across the modular environment. (Figure 5).

All key exchange, issue and renewal processes are handled over secure HTTP communication. This means that token handling and credential-related traffic are protected in transit, which is essential in a distributed service architecture. In a platform where multiple modules talk to shared APIs and tools, transport security is not a secondary concern. It is part of the minimum operational requirement for responsible deployment.

The authorisation logic is implemented at service level through token validation and access checks. This allows backend services to assess whether a request is both authenticated and allowed under the current role or context. In practical terms, this supports user restrictions, controlled access to features and the distinction between general pilot use and more specialised service access. As the Beta platform includes both user-facing modules and backend analytical services, this server-side control remains important.

No full single sign-on mechanism is envisaged as a formal feature of the solution. However, the platform does provide an SSO-like experience in practice because the stand-alone modules use the same authentication endpoint. This allows users to move between PoliRuralPlus applications, including Jackdaw and the Multiactor tool, with a more continuous session experience based on shared identity handling and cookie-supported access continuity. The result is not enterprise-grade federation across external domains, but a coherent and practical user experience within the platform's own application family.

This arrangement is suitable for the Beta stage because it balances security, simplicity and integration effort. It avoids the complexity of a broader federation model while still giving users a reasonably seamless way to access multiple modules. For a project platform serving pilots and demonstrator activities, this is an appropriate compromise. It secures the environment through established methods, keeps implementation manageable and supports the modular architecture without forcing every component into one tightly coupled identity solution.

3. Beta Components and Services

This chapter describes the main Beta components and services that make up the operational PoliRuralPlus environment. The previous chapter explained how the platform is structured and how the modules are connected. Here the focus moves one level deeper; each component is described in terms of what it does, how far implementation has progressed, what data it uses, what it returns to users or other services, how it is accessed, what it depends on, where the present limits are, and what will be developed next.

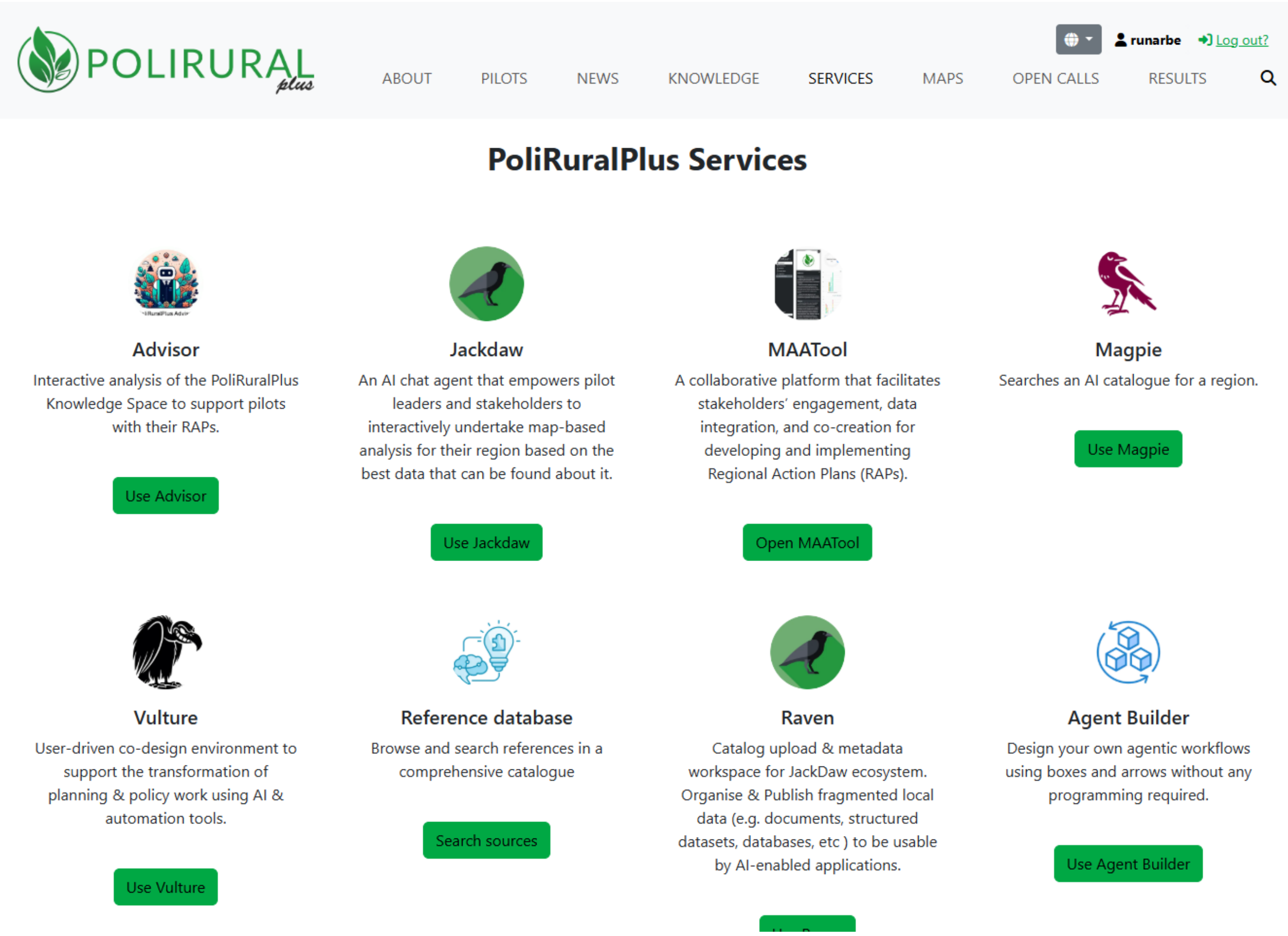

Figure 6: This shows the high-level services available from the PoliRuralPlus project

The Beta platform is not one single application with one fixed workflow. It is a set of connected services that support communication, evidence retrieval, data discovery, mapping, stakeholder management and analytical work. Some parts are visible to end users as web tools. Others work mainly as background services that support retrieval, metadata processing, spatial access or AI-assisted interactions. This modular approach remains important at component level as well. (Figure 6)

A clear shift between the earlier project stages and the current Beta release is the movement from narrow proofs of concept towards more reusable services. In several places, the work has moved away from hard-coded functions designed for one specific dataset or one isolated use case. In their place, the Beta release increasingly uses metadata-driven and API-based mechanisms that can be applied across different datasets and pilot contexts. This is especially visible in the data, retrieval and conversational services.

Another practical feature of the current release is that not all components have evolved in the same way. Some have expanded mainly through content growth and improved operational use. Others have advanced through stronger integration, better metadata, or support for more generic workflows. A few earlier functions remain stable with limited change because related needs are now addressed by newer modules. This uneven pattern is normal in a Beta platform that is being shaped by pilot use, implementation constraints and the emergence of new technical options.

The following sections present each Beta component using a common structure. This helps show not only the individual function of each module, but also how the modules fit into the wider service environment and where current development effort is being concentrated.

3.1 PoliRuralPlus Hub / Hub4Everybody evolution

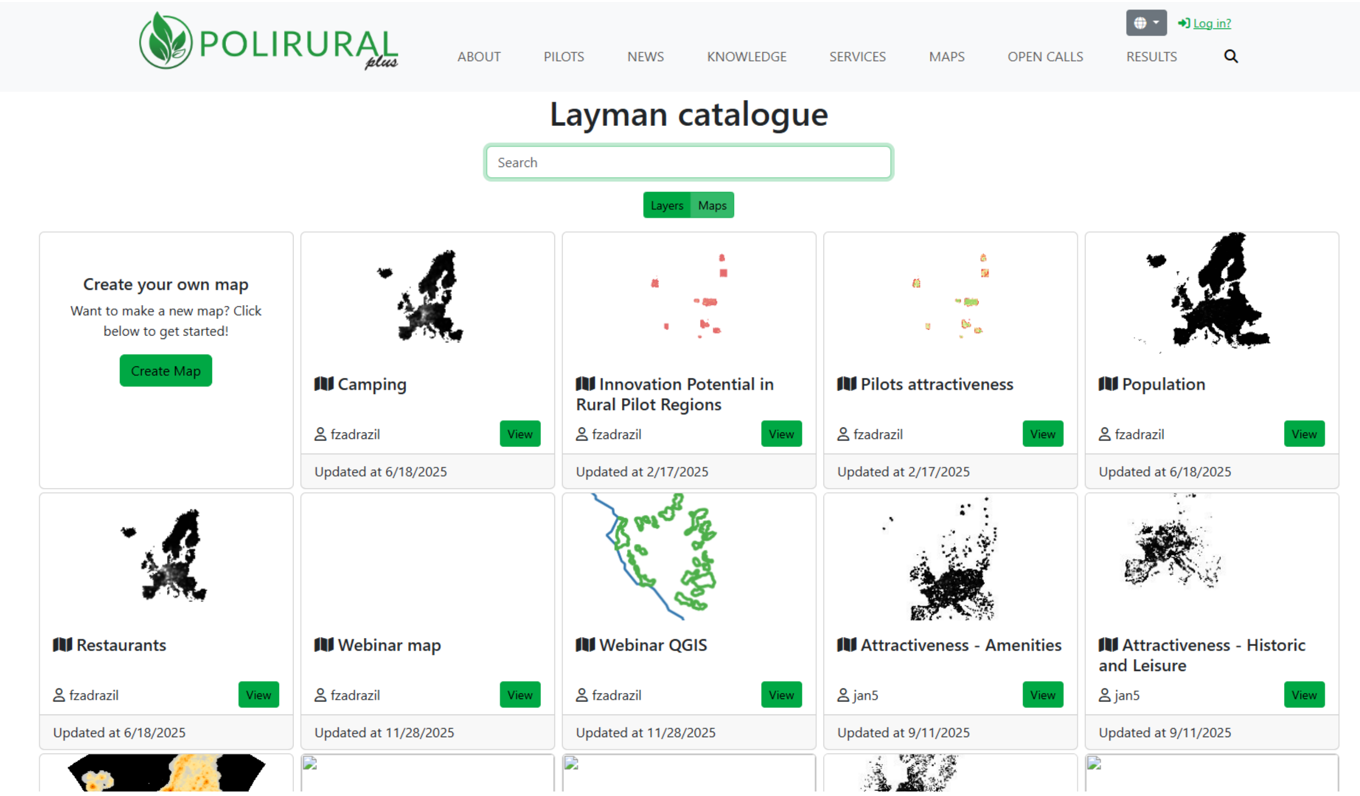

Figure 7: PoliRuralPlus Hub4Everybody Repository

The PoliRuralPlus Hub remains the main entry point to the platform environment. Its purpose is to provide a central place for communication, publication, document access and service entry. In practical terms, it works as the visible front door of the Beta environment, while also supporting content management and integration with selected backend functions. This role is especially important in a project where users do not approach the platform through one single analytical task, but through a wider mix of documents, results, services and pilot material. (Figure 7)

In implementation terms, the second reporting period has brought more change in content scope than in platform logic. The Hub has evolved mainly by expanding the amount and variety of available material. This includes project documents, outputs, library resources and service links that make the environment more useful as a working repository rather than only a demonstration shell. The underlying role of the Hub as a common access point has therefore become clearer through operational use rather than through radical redesign.

The main inputs to the Hub are documents, publication records, content pages, media assets and structured references to project outputs or related resources. These inputs come from project partners and from editorial processes within the Hub environment. A key output is curated and searchable access to this material for platform users. In addition, the Hub now contributes directly to AI-supported retrieval; documents added to the PoliRuralPlus library are made available through an integration that feeds the retrieval layer behind Jackdaw. This means that the Hub is not only a publication surface, but also part of the evidence infrastructure.

The user interface is provided as a web-based content and access environment. It combines standard page-based navigation with repository-style access to documents and materials. On the service side, the Hub depends on content management functions, file storage, authentication, and the document ingestion pipeline that prepares content for retrieval use. Through these links, the Hub acts as an integration layer between user-facing publishing and the advisory or GeoAI functions described later in this chapter.

Current limitations are mainly linked to the editorial and organisational side of growth. As the amount of available content increases, the value of the Hub depends more strongly on consistent classification, metadata quality and content curation. Without this, a larger repository can become harder to navigate. The next development steps therefore concern continued content expansion, stronger harmonisation of metadata and taxonomy, and tighter operational links between Hub publication, retrieval services and pilot-facing knowledge access. (Figure 5)

3.2 Data and metadata services

The purpose of the data and metadata services is to support publication, discovery, cataloguing and structured access to the datasets and documents used by the Beta platform. These services provide the foundation for machine-readable data access, catalogue search, spatial service exposure and repository functions. They are important not only for direct user access, but also because other components depend on them for retrieval, tool selection, geospatial interaction and evidence traceability.

The implementation has changed significantly since the first proof-of-concept stage. Earlier demonstrations showed chat flows supported by hard-coded Python functions, where metadata was often embedded informally in code comments. This was sufficient to prove interaction logic, but it was not suitable for reusable and scalable operations. In the second release, much more emphasis has been placed on explicit, higher quality metadata. This shift allows the platform to move from fixed tool-data combinations towards generic tools that can be instantiated over a much broader range of datasets.

This change has led to the central role of the module called Raven. Raven handles the selection of relevant data resources and the dynamic instantiation of tools on top of those resources. Instead of building a narrow function for one predefined dataset, the system now relies on metadata and service descriptions that allow the same class of tool to be applied to different data sources. Inputs to this layer therefore include catalogue metadata, service descriptions, dataset identifiers, access URLs, spatial descriptors, and repository information for documents and structured data. Outputs include discoverable records, publishable services, resolved data references, and instantiated tool contexts that can be used by Jackdaw and related components.

Access is provided through APIs, metadata catalogues, repository interfaces and spatial service endpoints. Depending on the resource, users may also interact through web catalogue views or via chat-driven workflows that call these services indirectly. The dependencies are substantial; the data and metadata layer depends on structured catalogue models, repository storage, spatial publishing mechanisms, indexing services and access control. It also has strong links to standards-oriented work discussed earlier, including OGC-related discovery concepts and machine-readable service exposure.

The current limitations are typical for metadata-intensive environments. Genericity depends on consistent descriptions, and the value of tool instantiation depends on how complete and reliable the underlying metadata is. Heterogeneous source quality still creates uneven behaviour across datasets. The next development steps therefore focus on further metadata harmonisation, improved catalogue completeness, broader support for machine-actionable descriptions, and stronger operational links between Raven, retrieval pipelines and geospatial analysis services.

3.3 Jackdaw GeoAI chat agent

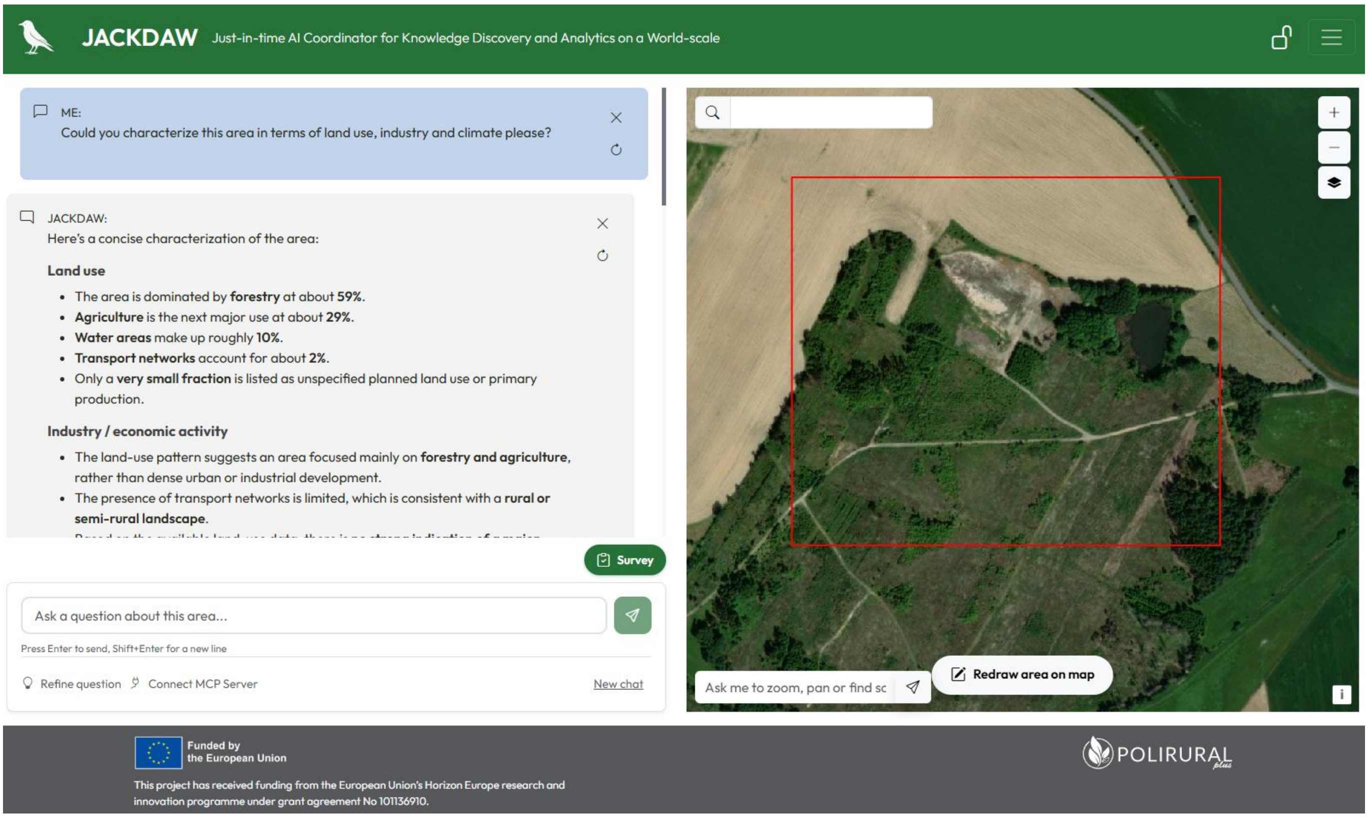

Figure 8: Screen shot from the main Jackdaw GeoAI Chat agent user interface

Jackdaw is the main spatially enabled chat agent in the Beta platform. Its purpose is to combine natural language interaction with geospatial context, metadata discovery and the invocation of analytical tools. This makes it more than a general chat interface. It is intended as a working assistant that can interpret a user question, identify relevant evidence sources, bring in map-related context, and trigger supporting operations when a plain language response alone is not sufficient. (Figure 8.)

The implementation status of Jackdaw in the Beta release reflects the broader architectural shift described in the previous chapter. The chat agent no longer depends only on tightly coupled internal functions. It now operates in a more open, service-oriented way, where tools and supporting resources can be reached through shared interfaces and agent-oriented interaction patterns. This makes the component more reusable and less bound to one front-end assumption. It also supports the move from isolated demos toward a platform service that can interact with multiple data and tool sources.

Jackdaw takes several types of input. These include user prompts, conversation context, retrieved documents, metadata records, map layers, service descriptions and, where relevant, structured spatial or analytical parameters. Its outputs are generated responses, source-grounded answers, references to documents or datasets, tool invocation results and, in some cases, geospatially informed follow-up suggestions. The component can therefore produce both conversational and action-oriented outputs, depending on what the request requires.

Users access Jackdaw through a conversational user interface. Behind that interface, the component depends on language model services, RAG services, Raven-driven metadata and tool selection, authentication, and access to geospatial data or analytical functions. In practice, Jackdaw also depends on the quality of the retrieval and metadata layers. If relevant sources are missing, weakly described or not exposed through usable service interfaces, the quality of the interaction declines. The chat experience is therefore only one visible part of a wider dependency chain.

The present limitations are linked to the maturity of connected sources and tools rather than to the conversational shell itself. Spatially enabled interaction remains sensitive to metadata completeness, service reliability and the difficulty of keeping evidence trails transparent when multiple tool calls are involved. The next development steps include continued improvement of evidence grounding, better exposure of source provenance, stronger connection to generic tool instantiation through Raven, and further maturation of GeoRAG-style interaction where textual and spatial evidence are handled together.

3.3.1 Technical overview of backend API services

The Jackdaw component forms the intelligent backend of the PoliRuralPlus platform, providing a RESTful and streaming API that enables natural language interaction with a curated set of geospatial, agricultural, climatic, and socio-economic data services. It is implemented in Python using the FastAPI framework and exposed as a versioned HTTP service.

Agent Architecture

At the core of Jackdaw is a conversational agent orchestrated using LangGraph, a directed state graph framework built on LangChain. The agent follows a structured reasoning loop: incoming user messages are first passed through a routing stage that classifies the user's intent and selects an appropriate subset of available tools (up to five per turn). The agent then invokes a large language model (LLM) — configurable to use OpenAI-compatible endpoints, including models hosted by project partners at PSNC — which reasons over the selected tools and the conversation history to produce a response or a set of tool calls.

Tool outputs are collected and fed back into the agent loop, which continues until the LLM determines that no further tool calls are needed. A dedicated GeoRAG subgraph handles the specialised case of geospatial dataset discovery and querying: it first identifies relevant datasets from the Raven geospatial catalogue using semantic matching, and then formulates spatial queries against those datasets to retrieve analysis-ready results.

Tool Ecosystem

Jackdaw exposes eighteen core analytical tools callable by the LLM. These cover the following thematic areas:

- Geospatial catalogue: Dataset discovery and attribute querying via the Raven API, supporting any dataset registered in the project's spatial data infrastructure.
- Climate and agro-climatology: Access to ERA5-Land historical data, WorldClim monthly climatology, IPCC RCP 4.5 climate projections for the 2050s, and short-term weather forecasts via Open-Meteo. Sixteen agro-climatic indicators are supported, including growing degree units, frost occurrence, and precipitation patterns.
- Land cover and land use: Classification statistics derived from the OLU_EU dataset, providing coverage percentages for user-defined areas of interest.
- Terrain: Elevation analysis (minimum, maximum, and average) from a digital elevation model.
- Short food supply chain (SFSC) assessment: A suite of four tools for assessing farm scale distribution, agricultural block size, LPIS-based crop suitability for direct sales, and local market demand potential.
- Demography and business: NUTS3-level population and business demography statistics sourced from Eurostat, including grid-based population data from the 2021 census.
- Points of interest and attractiveness: Counts and summaries of Smart POI categories, used for computing area attractiveness and innovation potential scores.
- Policy knowledge: Retrieval-augmented generation against a vector database of PoliRuralPlus policy documents and rural development knowledge.

All tools accept a spatial extent in WKT format, which is automatically populated from the user's current map viewport, ensuring that analysis is always spatially contextualised to what the user is viewing.

Extensibility via Model Context Protocol

Jackdaw implements the Model Context Protocol (MCP), an open standard for connecting LLM agents to external tool servers. System administrators can register MCP servers in a configuration file at startup, making their tools available to all users. Individual users may also connect personal MCP servers dynamically at runtime through a dedicated API endpoint. MCP tools are namespaced by server name and receive the user's current spatial context automatically, enabling third-party tools to participate seamlessly in geospatial workflows.

Routing and Skill System

To manage the complexity of a large tool catalogue, Jackdaw employs a lightweight routing mechanism: a single LLM call at the start of each agent turn classifies the user's intent and selects the most relevant tools, reducing noise and improving response quality. Complementing this is a skills system, in which domain-specific procedural playbooks (authored in YAML and Markdown) can be matched to recognised user tasks. When a skill is activated, it injects curated guidance — preferred tools, recommended steps, and anti-patterns to avoid — into the agent's system prompt, effectively encoding expert domain knowledge into the agent's behaviour without hard-coding it.

Streaming and Interruption

Responses are streamed to the client using server-sent events (SSE), allowing the frontend to display incremental progress as tool calls are initiated and completed. The agent supports an interruption mechanism whereby tools marked as requiring user approval pause execution and surface a confirmation request to the user before proceeding. This design allows sensitive or resource-intensive operations to remain under user control while preserving the conversational flow.

3.3.2 Technical overview of Jackdaw front-end service / user interface

The interface follows a two-panel layout: an interactive map occupies one half of the screen, while a conversational chat panel occupies the other. The workflow is area-first: the user begins by defining a geographic region of interest on the map — either by drawing a rectangle freehand or by using the place-name search to navigate to a known location — and then poses natural-language questions about that area in the chat. This spatial grounding ensures that all analytical queries are explicitly tied to a user-defined territory, which may range from a single farm plot to a regional administrative boundary.

Map Component

The map is built on OpenLayers and supports standard web-mapping interactions (pan, zoom). It includes a GeoNames-powered place-name search control, enabling users to navigate quickly to locations by name. A drawing tool allows the user to mark a rectangular area of interest; the resulting geometry is encoded in WKT (Well-Known Text, EPSG:4326) and attached to every subsequent chat message as spatial context.

A distinctive feature is a natural language navigation bar, powered by a locally-running large language model (Ollama). Users can type instructions such as "zoom in three steps", "pan right", "find Bratislava", or "draw a 5 km square here", and the model interprets the intent and issues the corresponding map operation via a structured tool-calling mechanism. This makes the map accessible to users who are unfamiliar with conventional GIS navigation controls.

Conversational AI Panel

The chat panel streams responses from a backend AI agent over a persistent conversation thread. While the agent is processing, the user sees live progress status updates. Once the response is complete, it is rendered as formatted text (Markdown), and any accompanying structured data — statistics, indicator tables, or time series — is presented as interactive HTML tables and charts immediately below the response.

The interface maintains thread continuity across multiple turns: the conversation history is preserved in a server-side checkpoint, so follow-up questions can reference prior context without re-sending the full history. Users can restart a conversation from any earlier message, which allows iterative refinement of the area or the line of inquiry.

Sample queries shipped with the interface illustrate intended use cases: characterising an area in terms of land use, industry, and climate; and assessing the suitability of an area for a specific crop. These serve both as onboarding guidance and as representative benchmark prompts for evaluation purposes.

Human-in-the-Loop Control

When the backend agent requires user input before proceeding — for example, to confirm a potentially significant data retrieval or processing step, or to have the user choose between alternative analysis strategies — the chat panel presents an approval card. This card lists the pending decision items, allows the user to inspect the parameters involved, and requires an explicit confirm or reject action before the agent resumes. This mechanism implements a controlled human-in-the-loop pattern that keeps the user informed of and in control over consequential agent actions.

External Tool Connectivity (MCP)

Advanced users and integrators can connect external data sources and tools to Jackdaw via the Model Context Protocol (MCP). A configuration modal allows one or more MCP server endpoints to be registered; the backend agent can then invoke tools exposed by those servers as part of its reasoning process. This makes the architecture extensible without requiring changes to the core application.

Research Survey

To support ongoing evaluation of the platform, Jackdaw includes an opt-in user survey. Participants select their professional role (e.g. farmer, policy maker, researcher, GIS specialist) and consent to having their conversation logs recorded anonymously for research purposes. Participation is managed with a per-session anonymous identifier and can be revoked at any time. This mechanism is used to gather real-world interaction data that informs iterative improvement of the agent's response quality and tooling.

3.3.3 Raven — Dataset Ingestion and Metadata Management Tool

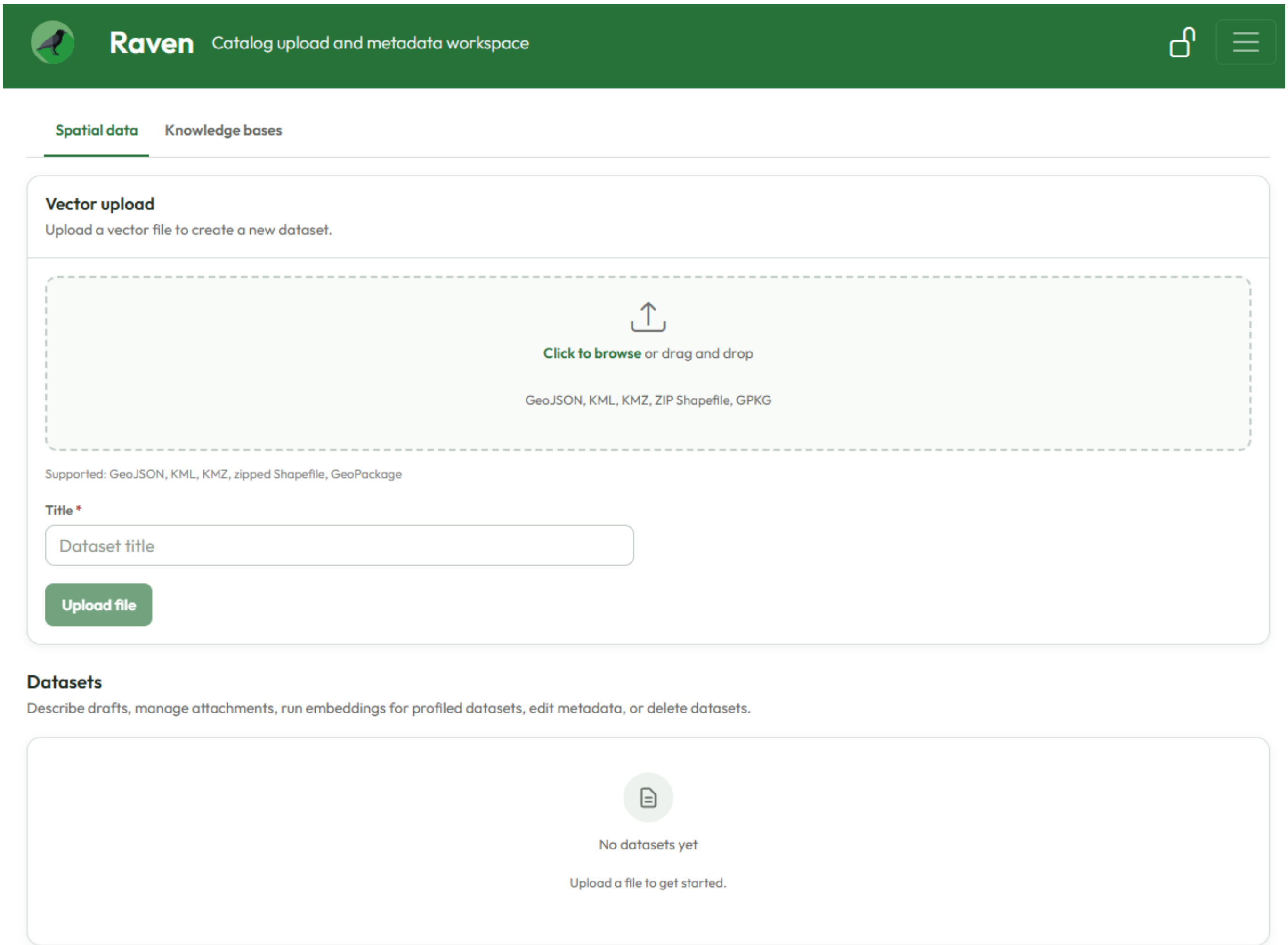

Figure 9: Raven user interface for upload of spatial datasets

Raven is the data ingestion and curation component of the platform, purpose-built for the workflow of preparing geospatial vector datasets for inclusion in the shared catalog. It is architecturally distinct from Jackdaw, the AI-assisted query interface, in that Raven addresses the supply side of the data pipeline: it is the tool through which data providers upload, describe, and govern their datasets, whereas Jackdaw serves data consumers by allowing natural-language exploration of the catalog that Raven populates. For an example of the user interface, see Figure 9.

The upload mechanism is built around a resumable, chunked transfer protocol. Large vector files are split into fixed-size binary chunks and transmitted in parallel (up to three concurrent threads) against a dedicated ingest endpoint. If a transfer is interrupted, the session can be resumed at a later point using the server-tracked list of already-received chunks, avoiding the need to re-transmit data that was successfully delivered. This design accommodates the large file sizes typical of high-resolution geospatial datasets and unreliable network conditions common in field or institutional contexts.

Following a successful upload, Raven transitions the dataset into a structured metadata authoring workflow. The data provider is prompted to supply a human-readable title and description, classification tags, temporal coverage (either as an explicit date range or explicitly marked as unknown), and per-column semantic descriptions. The description quality guidance in the interface explicitly encourages multilingual input to maximise downstream semantic search recall. Datasets remain in a draft state until the provider explicitly publishes them to the catalog.

Beyond initial publication, Raven supports ongoing dataset lifecycle management: metadata can be edited after publication, file attachments (e.g. methodology documents or data dictionaries) can be added or removed at any time, access permissions can be granted or revoked to other named users within the identity provider, and the underlying geometry layer can be replaced for datasets in eligible states without requiring a full re-ingest.

Backend API & AI Integration Layer

PRP Raven has its own API, a Python-based backend service providing geospatial dataset management, AI-assisted data discovery, and agent integration capabilities for the PoliRuralPlus platform. It is built on FastAPI and exposes two complementary interfaces: a REST API for direct client consumption and a Model Context Protocol (MCP) endpoint enabling integration with AI agent frameworks.

The service connects to a PostgreSQL database extended with PostGIS (for geospatial operations) and pgvector (for semantic vector search), supporting ingestion, profiling, and embedding of geospatial datasets. Uploaded datasets pass through a structured ingest pipeline — from initial upload through metadata enrichment and vector embedding via an external embedding API — resulting in datasets that are queryable both by structured SQL and by natural-language semantic search.

Web data acquisition is handled by an integrated extractor module combining DuckDuckGo-based web search and a Crawl4AI/Playwright headless browser crawler capable of breadth-first site traversal. These capabilities are exposed both as REST endpoints and as registered MCP tools, allowing AI agents to autonomously search and crawl the web as part of broader data discovery workflows.

The MCP tool registry is auto-discovered at startup, enabling modular extension without changes to the core application factory.

3.4 Magpie AI-enabled metadata/catalogue search

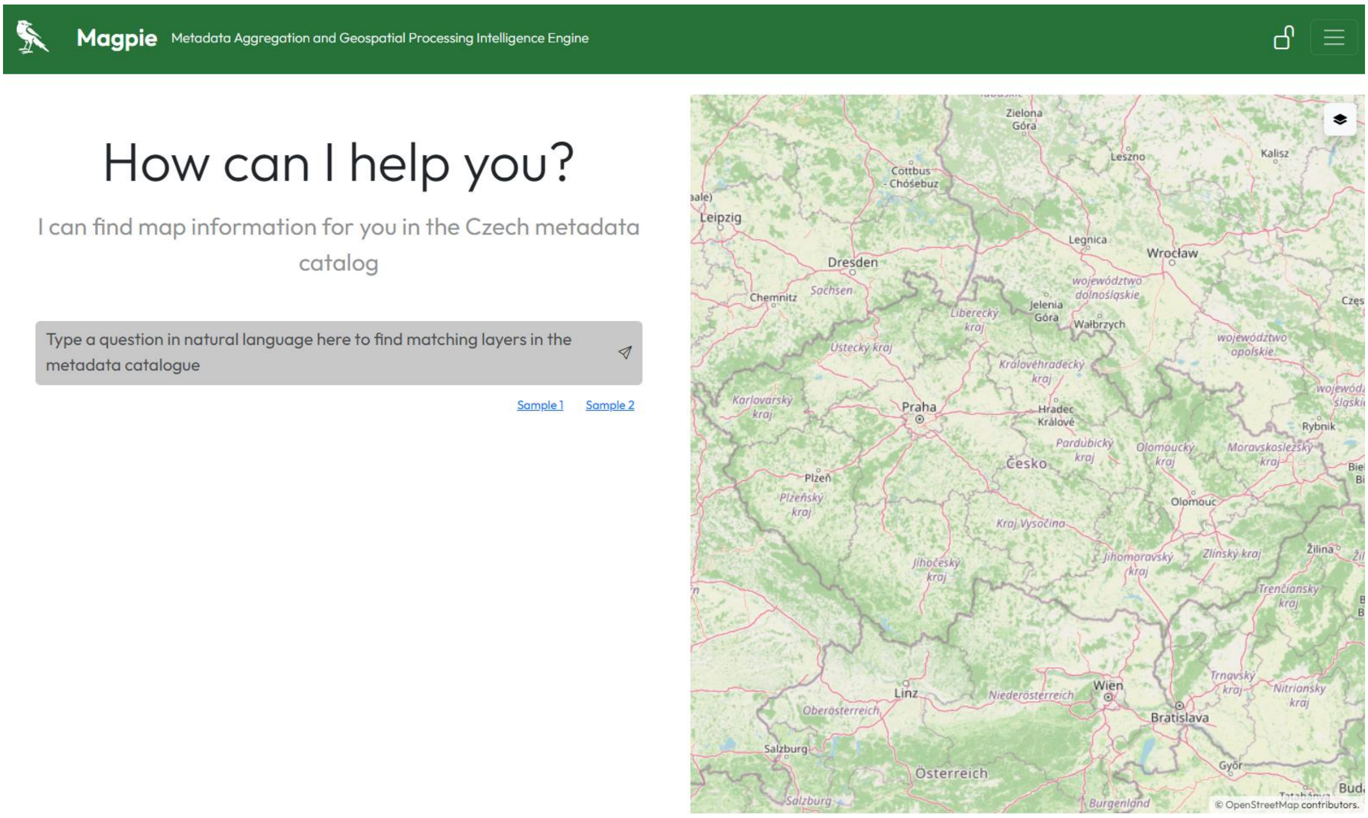

Figure 10: Example of the natural language search interface for the Czech metadata catalogue

Magpie was designed as the natural-language metadata and catalogue search component of the platform. Its purpose is to help users discover relevant datasets, metadata records and catalogue resources through language-oriented search rather than only through formal filtering interfaces. This includes links to OGC catalogue ideas and to broader data discovery contexts such as PSI, INSPIRE, Earth observation and emerging data-space resources. The underlying aim is to lower the access barrier to complex catalogues and make metadata more usable in practical policy and pilot settings (Figure 10).

In the current Beta release, Magpie has remained largely stable because much of its core function has been absorbed by Raven. This does not mean that the original need has disappeared. Rather, natural-language metadata discovery is now treated less as a separate visible product and more as part of the broader data-selection and tool-resolution workflow. The Beta architecture therefore preserves the function while shifting its operational centre into a more general module.

The inputs to this function are user queries, metadata records, domain vocabularies, catalogue indexes and service descriptions. Outputs include ranked metadata matches, inferred links to datasets or services, and selection cues that can be used by other components, especially conversational agents. In practical use, these outputs often appear through Jackdaw or through internal service calls rather than through a stand-alone Magpie interface. This reflects the wider integration logic of the Beta platform, where certain functions survive mainly as service capabilities.

Where a user interface is exposed, it is focused on search and discovery over metadata collections. At service level, the component depends on the same underlying catalogue and metadata infrastructure discussed in Section 3.2. It also depends on semantic clarity in metadata descriptions and on mappings between user language and technical catalogue fields. This is one reason why metadata quality has become more important in the second release than in the early proof-of-concept phase.

Current limitations arise from the limits of metadata itself. Natural-language search can improve discoverability, but it cannot fully compensate for incomplete records, weak keywords or inconsistent classification across domains. The next development path is therefore not likely to be a major stand-alone expansion of Magpie as an isolated product. Instead, development will continue through Raven and related discovery services, with better metadata alignment, stronger semantic enrichment and smoother integration into chat-driven and tool-driven workflows.

3.5 Vulture productivity sandbox

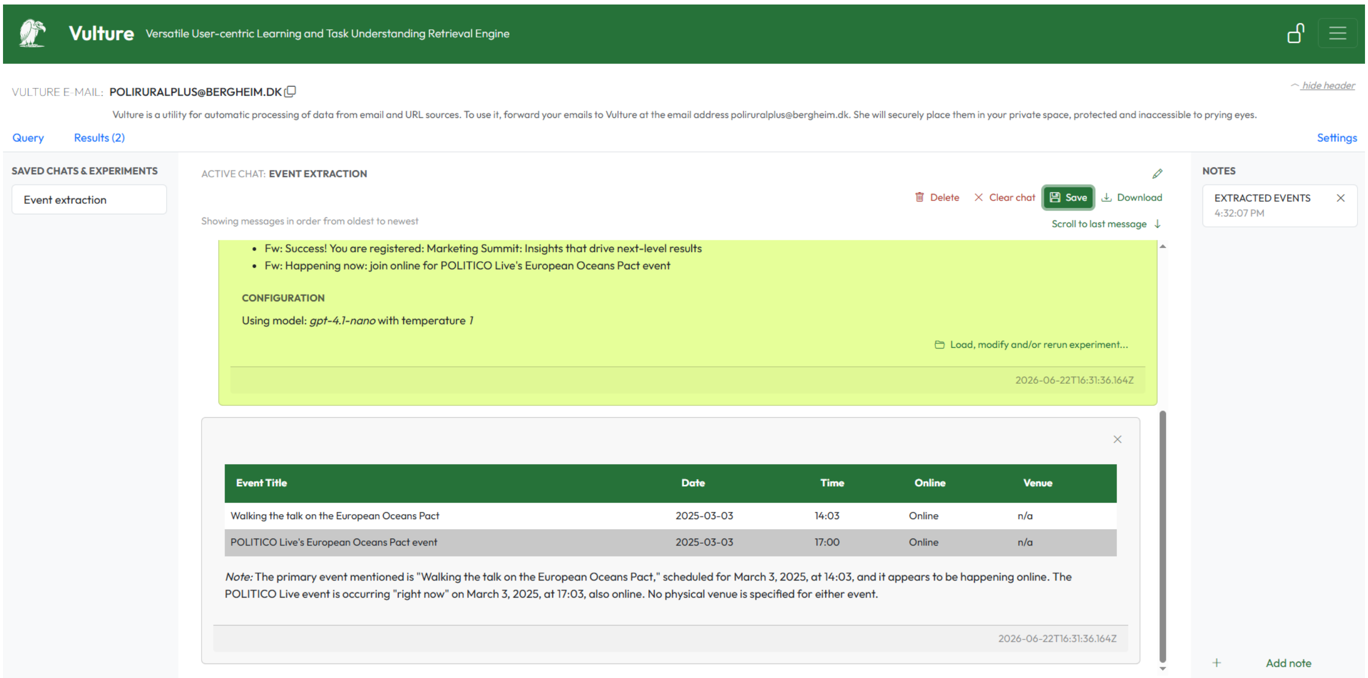

Figure 11: Sample screen from the Vulture module of Jackdaw

Vulture was created as a sandbox for AI-enabled productivity experiments. Its purpose is to provide a safe and flexible space for trying out workflows related to document processing, information streams and semi-structured inputs. In project terms, this kind of component is useful because not every emerging need should immediately become part of the stable platform surface. A sandbox allows fast experimentation, testing of formats and exploration of AI-assisted support tasks before deciding whether they deserve stronger integration (Figure 11).

In the present Beta stage, Vulture has remained largely stable and has become less central than originally expected. One reason is that parts of its intended role have been overtaken by tools such as OpenClaw and by more integrated AI workflows elsewhere in the platform. This is a common pattern in evolving technical environments; some experimental spaces continue to add value, while others become less decisive once more mature tools appear around them.

The inputs to Vulture typically include uploaded documents, copied text, semi-structured content, information feeds and other working materials that users want to transform, summarise, inspect or organise. Outputs depend on the specific experiment, but may include extracted information, summaries, structured representations, draft classifications or other AI-assisted intermediate products. The sandbox nature of the component means that outputs are often provisional and exploratory rather than formal platform evidence.

Access is usually provided through a web interface designed for experimentation rather than through a tightly controlled workflow. Dependencies include language model services, document parsing mechanisms and, where needed, access to external or internal content sources. Compared with more production-oriented services, the sandbox has lighter integration requirements, but this also means weaker guarantees in relation to reproducibility, provenance and long-term maintenance.

Its current limitations follow directly from this role. Vulture is not intended to carry the burden of core operational functions where auditability and repeatability are essential. As other components become more capable, the distinct role of the sandbox becomes narrower. The next development step is therefore likely to be selective maintenance rather than major expansion; useful experimental patterns can continue to be tested there, but stable and high-value functions should migrate into more structured platform services when they prove their relevance.

3.6 Boxes & Arrows Agent Builder

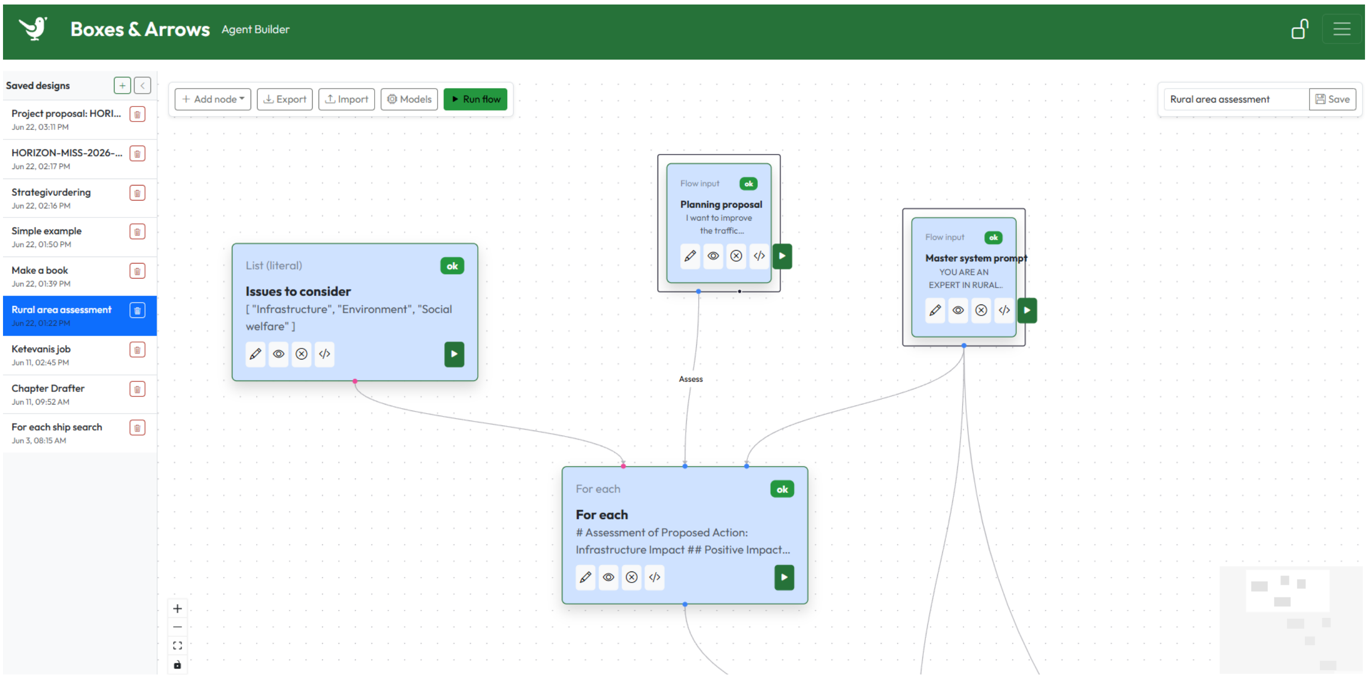

Figure 12: Screenshot from the agent builder user interface.

One challenge identified during the Beta use of Jackdaw is that users may treat chat agents as conventional search interfaces, asking one question and expecting one final answer. This interaction pattern does not fully exploit conversational agents, which can provide stronger results through iterative clarification, critical dialogue and structured follow-up (Figure 12).

In a chat interface backed by a retrieval-augmented generation engine, the agent identifies relevant information by comparing the user question, or an augmented version of it, against indexed document chunks. The RAG process typically returns a limited set of top-ranked chunks, often between 10 and 20. This is not equivalent to comprehensive analysis of the full corpus; it is a relevance-based match against a small subset of documents, with quality strongly dependent on the scope and precision of the user question.

If the user asks a specific question, the retrieved material may be suitable. If the question is broad and lacks clear scope or boundaries, the returned chunks may instead represent general statements with limited substance or analytical depth. For this reason, custom agents are needed to run structured conversations within domains where prior knowledge, predefined steps and quality checks can be specified.

The Boxes & Arrows Agent Builder module in Jackdaw therefore supports the visual construction of agent graphs that execute complex conversations. Each step can have its own instructions, background data or knowledge resources, and a defined output. The module supports branching, looping, merging and quality checks, enabling complex research and assessment tasks to be structured and partially automated. It also allows non-technical end users to configure their own agents and makes agentic AI functionality accessible to wider user groups.

3.7 PoliRuralPlus Advisor and RAG services

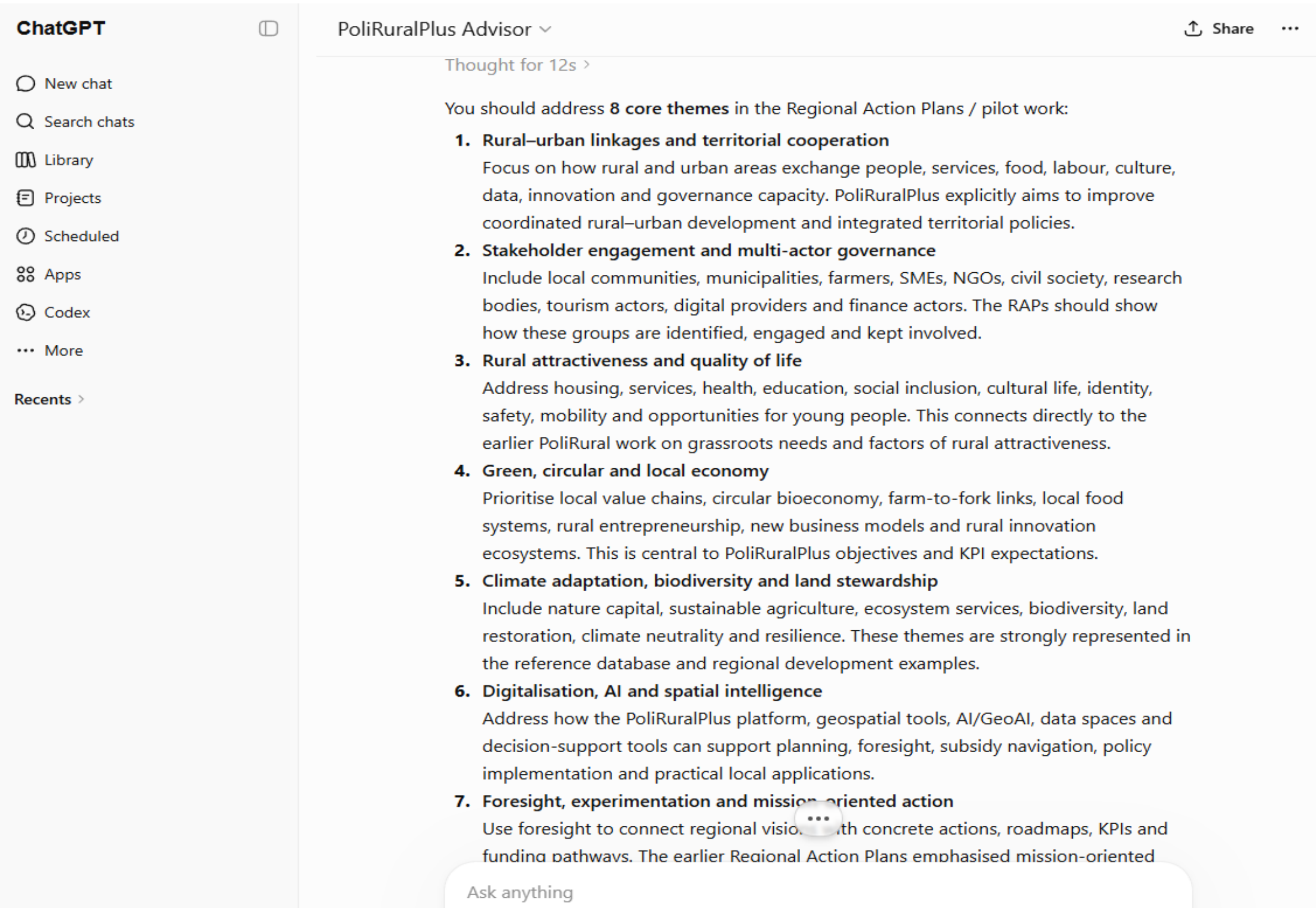

Figure 13: The Policy Advisor is a custom ChatGPT end-point augmented with relevant documents from the PoliRuralPlus project and other related Horizon Europe, national and regional policy documents

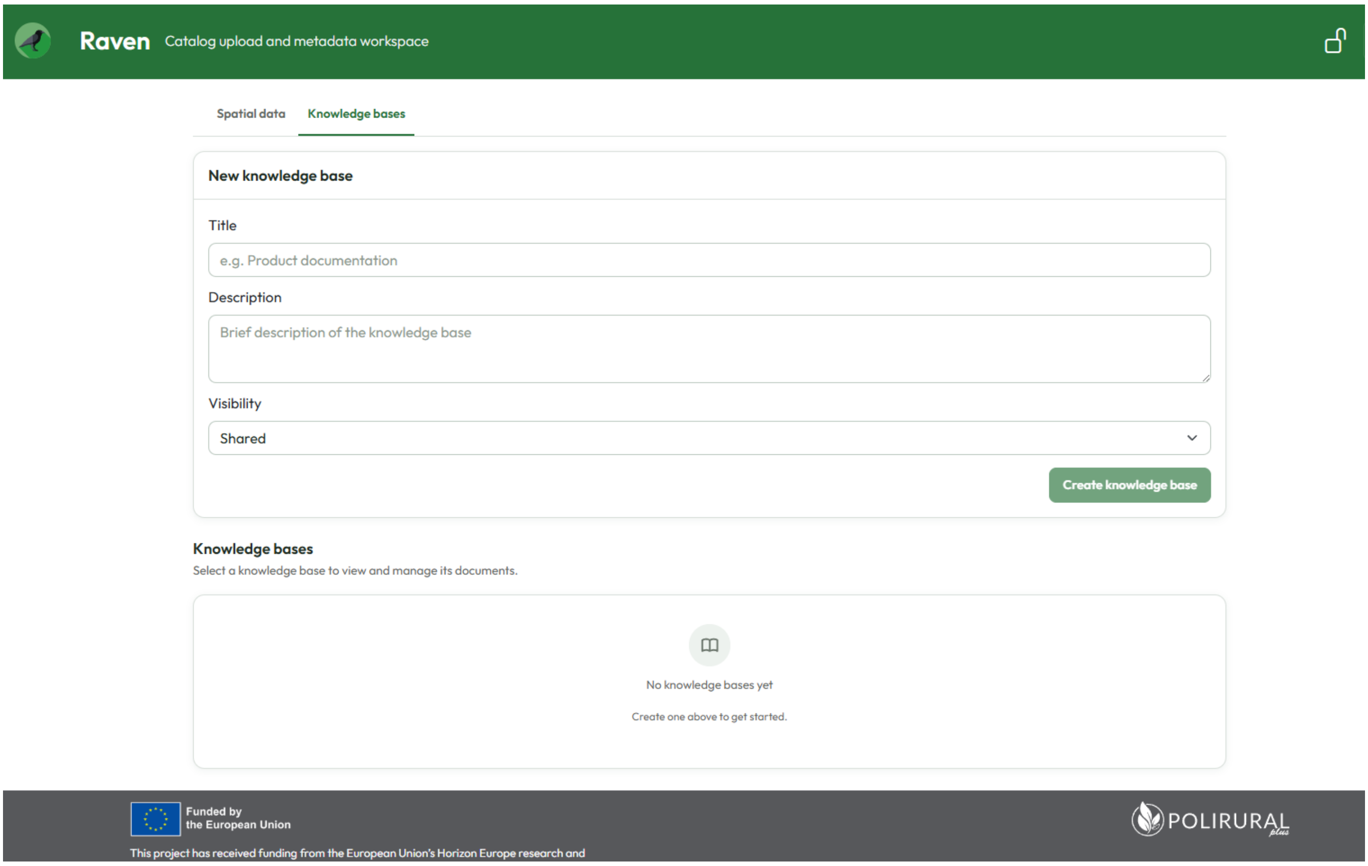

Figure 14: Users can add to, extend or build their own knowledge bases that go into the RAG that powers Jackdaw GeoAI chat agent. The two solutions both give access to the same underlying data volume.

The PoliRuralPlus Advisor (Figure 13) and its RAG services (Figure 14) form the main evidence retrieval layer for advisory use. The purpose of this component is to provide source-grounded support for policy brief preparation, RAP-related analysis, pilot evidence review and question answering over the project's own knowledge base. In the Beta release, the service operates as a private RAG for PoliRuralPlus, which means that retrieval is performed over controlled project-related corpora rather than over unrestricted public web content.

The implementation distinguishes between textual RAG and GeoRAG. Textual RAG retrieves and synthesises information from reports, policy documents, pilot material and related texts. GeoRAG extends this by bringing in spatial context, metadata records, map layers, indicators and analytical services. This distinction is important because many rural and regional development questions cannot be answered well from prose documents alone. They also depend on place, territory, comparability and access to structured spatial evidence.

Inputs to the service include ingested document corpora, chunked and indexed text, metadata records, user queries, retrieval parameters and, in the GeoRAG case, geospatial references and service-linked analytical context. Outputs include answer drafts grounded in retrieved evidence, document citations, source excerpts, reference trails and, where applicable, links to related spatial resources. One of the main objectives here is not simply to generate plausible text, but to support retrieval with evidence that users can inspect and reuse.

The Advisor is accessed through conversational interfaces and potentially through related service endpoints used by other modules. It depends on document preparation pipelines, indexing, embedding and retrieval infrastructure, access control, and the availability of well-managed source collections. It also depends on the Hub and other repository functions for content provision, and on the metadata layer for linking textual and spatial sources. A planned extension is the integration of RAG pipelines with Knowledge Graphs. This would support entity linking, semantic disambiguation, controlled vocabularies and more transparent evidence trails between user questions, retrieved sources, datasets and generated replies.

Current limitations concern both corpus quality and explainability. A private RAG can reduce noise, but its usefulness is still constrained by how complete and well-prepared the project corpus is. GeoRAG in particular requires stronger harmonisation between text retrieval, metadata discovery and geospatial service access. The next development steps are therefore to expand and clean the indexed corpus, improve citation-oriented retrieval, strengthen spatial evidence integration, and advance the planned Knowledge Graph connection so that generated answers can be traced more clearly back to entities, documents and datasets.

3.8 MapWhiteboard collaborative mapping

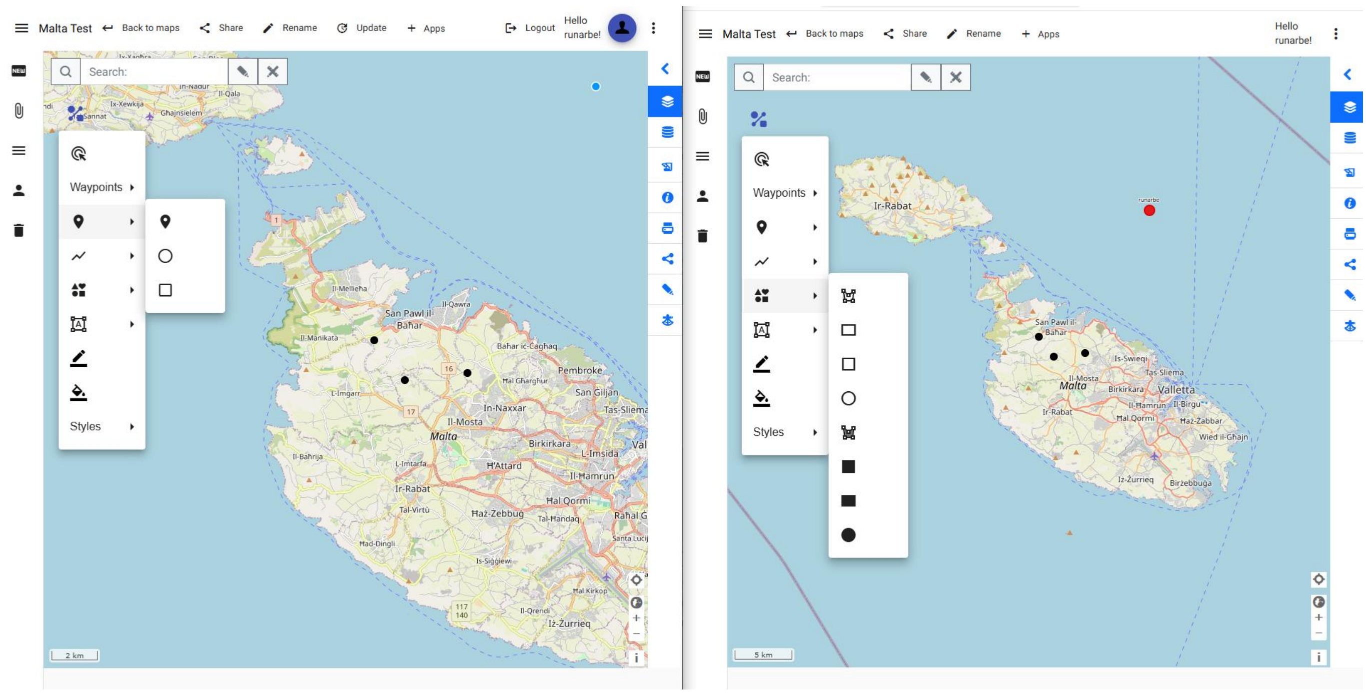

Figure 15: The MapWhiteboard application allows two or more connected map sessions to share a map view and see each other's cursors, and draw on the same map together.

MapWhiteboard (Figure 15) is the collaborative map-based workspace within the Beta platform. Its purpose is to support stakeholder engagement, spatial annotation and participatory planning through shared interaction with maps. In rural and regional development work, such a component is useful because many discussions are easier to structure around place-based references than around abstract documents alone. It gives participants a way to point to locations, exchange spatial interpretations and build simple shared map views.

The current Beta implementation is more focused and simpler than some earlier expectations around collaborative mapping. A new simplified instance has been developed and deployed. This instance allows cursor exchange between OpenLayers maps and feature exchange between OpenLayers VectorLayers. In practical terms, this creates basic shared navigation and shared object interaction, which are often the most immediately useful building blocks in collaborative map sessions.

Inputs to the component include map layers, vector features, user actions, annotations and live navigation events. Outputs include synchronised cursor position, exchanged features and a shared visual workspace where participants can inspect and discuss the same map context. While this may seem modest compared with a full collaborative planning system, these outputs provide an operational basis for joint exploration and lightweight participatory interaction without introducing unnecessary complexity.

The component is accessed through a web mapping interface and depends on OpenLayers, vector-layer handling, networked event exchange and the underlying authentication environment where restricted use is needed. It also has potential dependencies on other platform modules for shared identifiers, pilot context or linked resources. In future use, MapWhiteboard can support workflows that combine stakeholder participation, spatial discussion and retrieval of supporting evidence from other platform services. Current limitations are mainly linked to scope. The deployed instance offers focused collaborative exchange, but it does not yet represent a fully developed participatory planning suite with richer workflow and moderation functions. The next development steps would include deeper integration with pilot engagement processes, stronger links to evidence or annotation persistence, and clearer connection to modules such as the Hub, advisory services or the Multi-Actor Approach tool where map-based engagement could add operational value.

3.9 Multi-Actor Approach tool

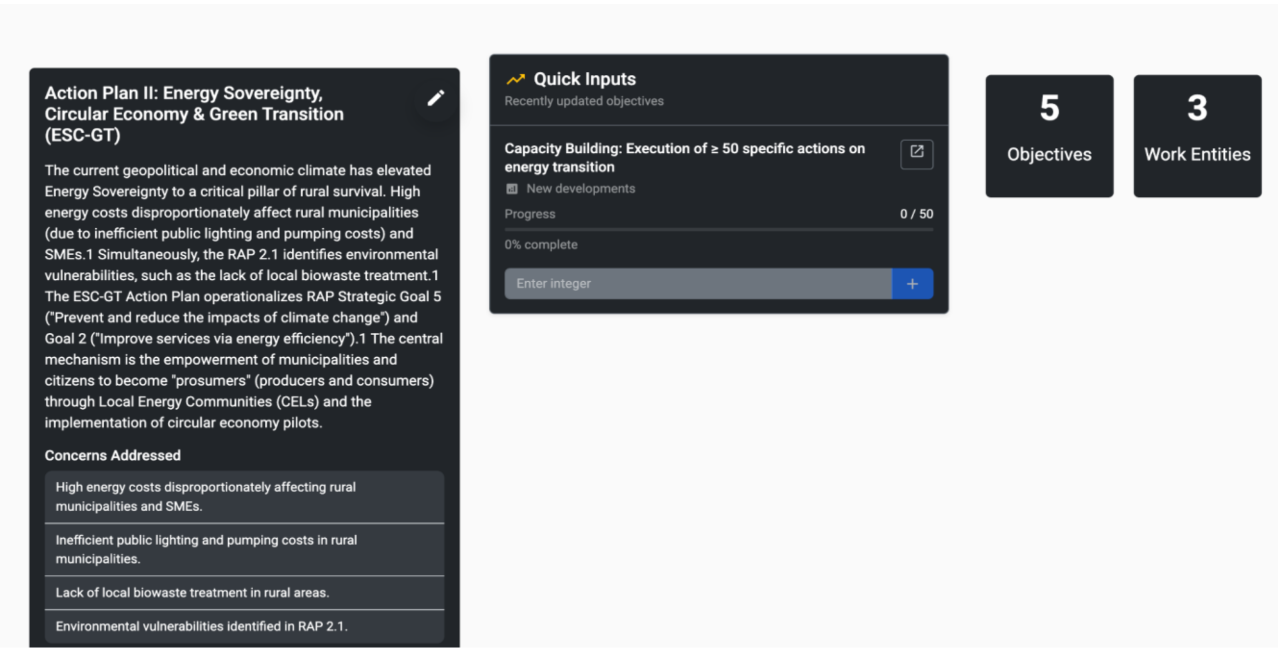

Figure 16: Screenshot from landing-page dashboard in MAA tool

The Multi-Actor Approach (MAA) tool is implemented in the Beta platform as the structured workspace used to organise pilot engagement around programmes, work packages, pilots, stakeholders, members, objectives, inputs and communication activities. In the PoliRuralPlus Beta instance, the tool is already configured for the PoliRuralPlus programme and mirrors the WP structure, including WP5 - Prepared Pilot by Stakeholder Multi-Actor Approach and nine pilot work entities. The dashboard view provides the operational evidence base for the Beta report, with the current workspace showing more than 300 users, 40 work entities, around 150 stakeholders, 49 objectives and 40 input records. For an example of the user interface of the MMA tool, see Figure 16.

The purpose of the component is to translate the Multi-Actor Approach from a methodological requirement into a traceable operational workflow. Pilot teams can register and classify stakeholders, assign roles and permissions, connect stakeholders and members to programmes or work entities, define objectives and sub-objectives, record indicators and inputs, and use configurable attributes to capture pilot-specific dimensions such as sector, organisation size or urban/rural profile. These records support WP5 stakeholder mapping, RAP preparation, pilot validation and feedback loops to WP3 and WP4.

The Beta implementation is delivered as a web application with an Angular front end and a NestJS/API back end. The user interface exposes programme, stakeholder, member, social-space, objective, work-entity, chart and communication views, while the API layer stores entities, roles, permissions, extra attributes and relationships between programmes, work entities, stakeholders, users, objectives and communications. Chart services can aggregate stakeholder distributions based on configured extra attributes, and the PoliRuralPlus workspace currently includes an example chart for the urban/rural stakeholder attribute.

Inputs for the tool include programme and pilot descriptions, stakeholder records, contact and organisation metadata, roles and permissions, configurable extra attribute values, objectives, indicators, input records and communication text. Outputs include searchable stakeholder and member registers, pilot/work-package views, objective progress tables, attribute-based distributions, communication history and structured evidence that can be reused in Beta testing reports, RAP documentation and stakeholder engagement summaries.

Integration with the wider platform is mainly through shared programme/work-entity identifiers, authentication, permission management and API-based access to MAA records. The component also provides a candidate evidence source for Hub publication, MapWhiteboard participation workflows and RAG/advisory services, because stakeholder, objective and pilot structures can be used as contextual metadata for later retrieval and reporting. The integrated Ramon AI assistant extends this direction by offering search and proposal workflows for work entities, users, stakeholders and objectives, with user confirmation before creation.

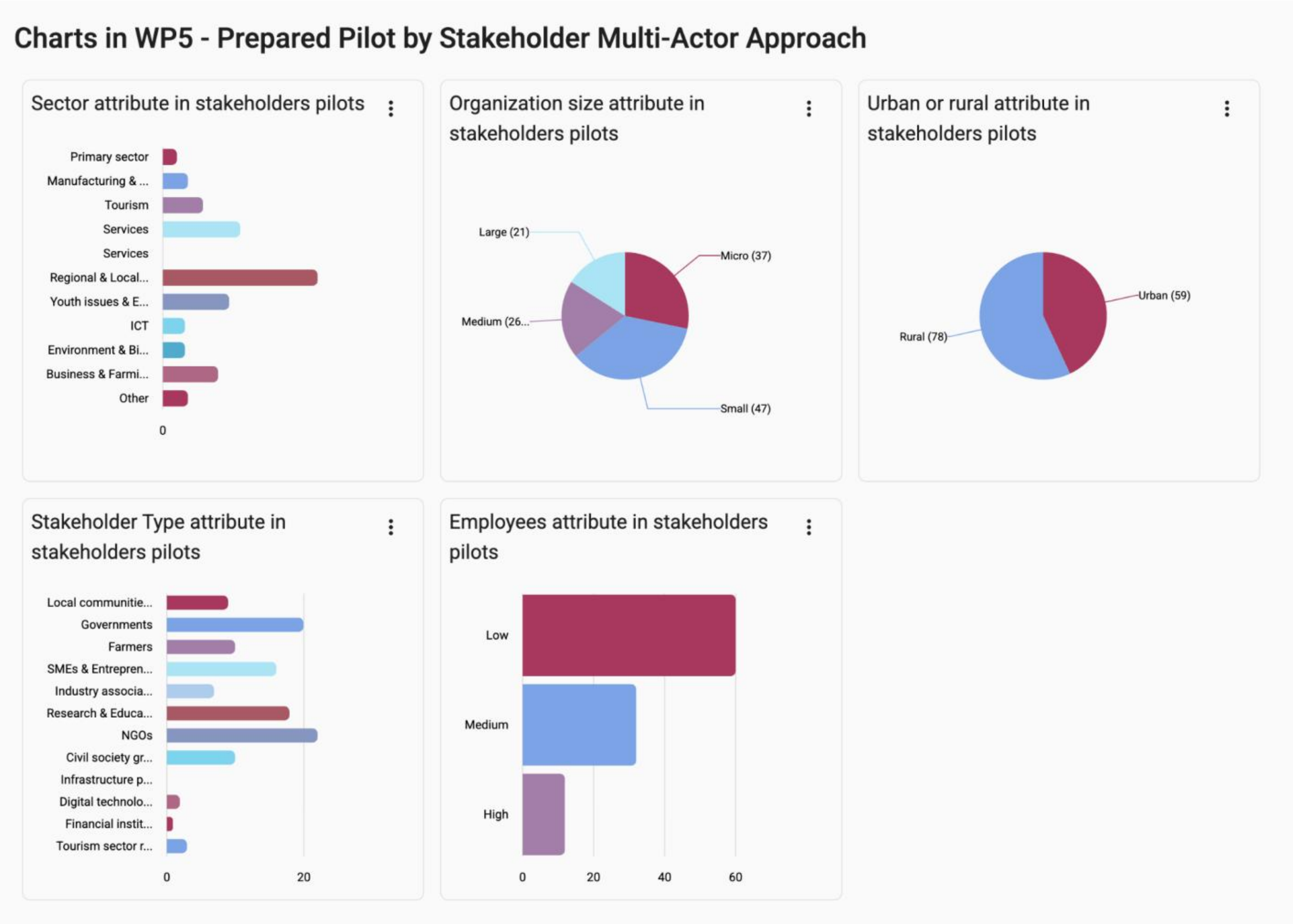

Figure 17: Dashboard for stakeholder engagement in MAA tool

Current Beta limitations are mainly related to completion and harmonisation of pilot data rather than to the core data model. In the inspected PoliRuralPlus workspace, social spaces are available in the navigation but are not yet populated at programme or WP5 level, and pilot-specific stakeholder assignments need to be completed consistently across the nine pilots. For D4.2, the MAA tool evidence is limited to the current Beta configuration, user/work-entity counts, available stakeholder and objective records, and observed needs for harmonisation across pilots. More comprehensive WP5 validation evidence will be collected during the final development phase and reported in D4.3. The remaining development step is therefore to finalise stakeholder mappings, define a common attribute vocabulary across pilots, populate social spaces and communication workflows, and prepare export/reporting functions so that WP5 validation evidence can be reused directly in D4.3 and in the final platform release.

3.10 Rural Attractiveness / area-potential analysis

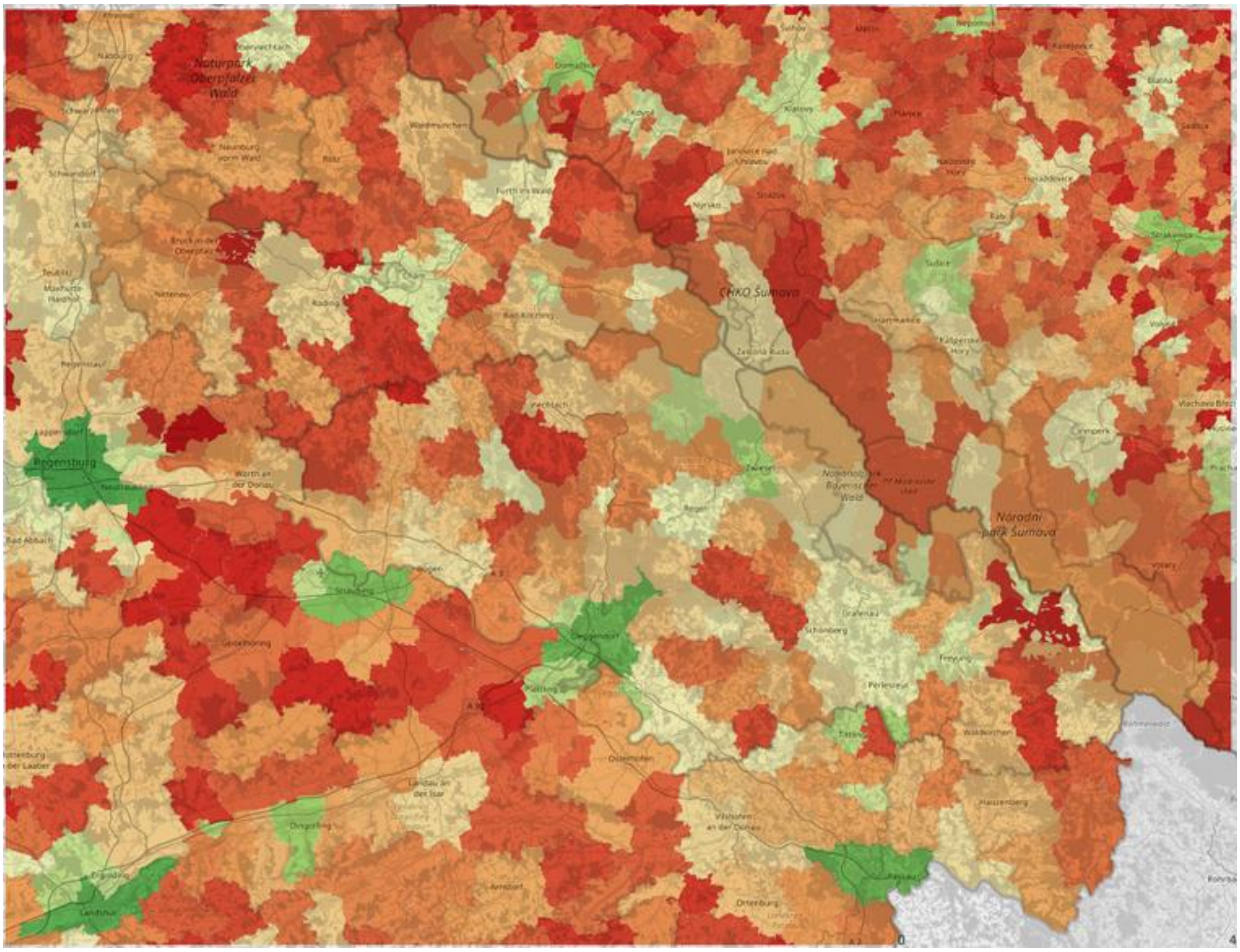

Figure 18: Regional Attractiveness Score (innovation potential) in Czech-Bavaria border

The Data-driven Innovation Potential Assessment tool (Figure 18, above) is implemented as the analytical engine for mapping and quantifying innovation capacity within the cross-border region of southwestern Bohemia and eastern Bavaria. Led by partners Plan4all and the Deggendorf Institute of Technology, the tool provides an objective evidence base for regional development. The current analytical framework processes layers of innovation infrastructure and socio-economic data, moving beyond traditional administrative boundaries to offer a high-resolution view of the territory's attractiveness.

The purpose of the component is to translate complex regional data into a data-driven assessment of attractiveness, creating a traceable link between raw indicators and strategic development. By processing data in a regular 1x1 km grid, the tool identifies spatial distribution of potential without the methodological bias inherent in aggregated administrative units. This allows stakeholders to pinpoint specific locations for Food Innovation & Rural Entrepreneurship, evaluate Digital Skills & AI Competence, and support AI-powered grant assistance for rural actors. The workflow is designed to build a "critical mass" for regional transformation by connecting the real spatial potential with the necessary funding and infrastructure.

The technical implementation is delivered as a scalable and repeatable analytical framework. The system architecture integrates heterogeneous data sets—including transport networks, research centers, and universities—into a unified processing layer. Advanced statistical methods are applied to these layers to quantify innovation potential at the grid level. The user interface provides specialized visualization modules that allow for multi-scale analysis, from a detailed local grid to the broader cross-border context of Central Europe, supporting specific segment analyses such as agrotourism or high-tech hub identification.

Inputs for the tool include statistical indicators, geographic coordinates of innovation infrastructure (universities, research institutes), transport network data, and socio-economic metadata. Outputs include the consolidated Attractiveness Index, high-resolution thematic maps, municipal-level potential summaries, and targeted reports for specific sectors like the agri-food economy. These outputs provide searchable spatial evidence that can be reused in strategic investment plans, territorial impact assessments, and regional development roadmaps.

Integration with the wider innovation ecosystem is achieved by serving as the primary data pillar for emerging technology hubs and regional authorities. The component acts as a candidate evidence source for cross-border platforms, providing the objective foundation for coordinating investments between Czech and Bavarian entities. By aligning infrastructure data with private sector needs, the tool facilitates the creation of new value chains and supports long-term sustainability and climate resilience in the regional economy.

Current limitations are primarily focused on the ongoing population of specific segment analyses and the continuous refinement of the heterogeneous data layers. While the 1x1 km grid provides high precision, the next development step involves further harmonizing cross-border statistical indicators and expanding the AI-based advisory services for subsidy assistance. Future iterations will focus on scaling this methodology to other Central European regions and finalizing the integration of real-time digital skill assessment metrics to further support the regional agri-food workforce (Figure 19).

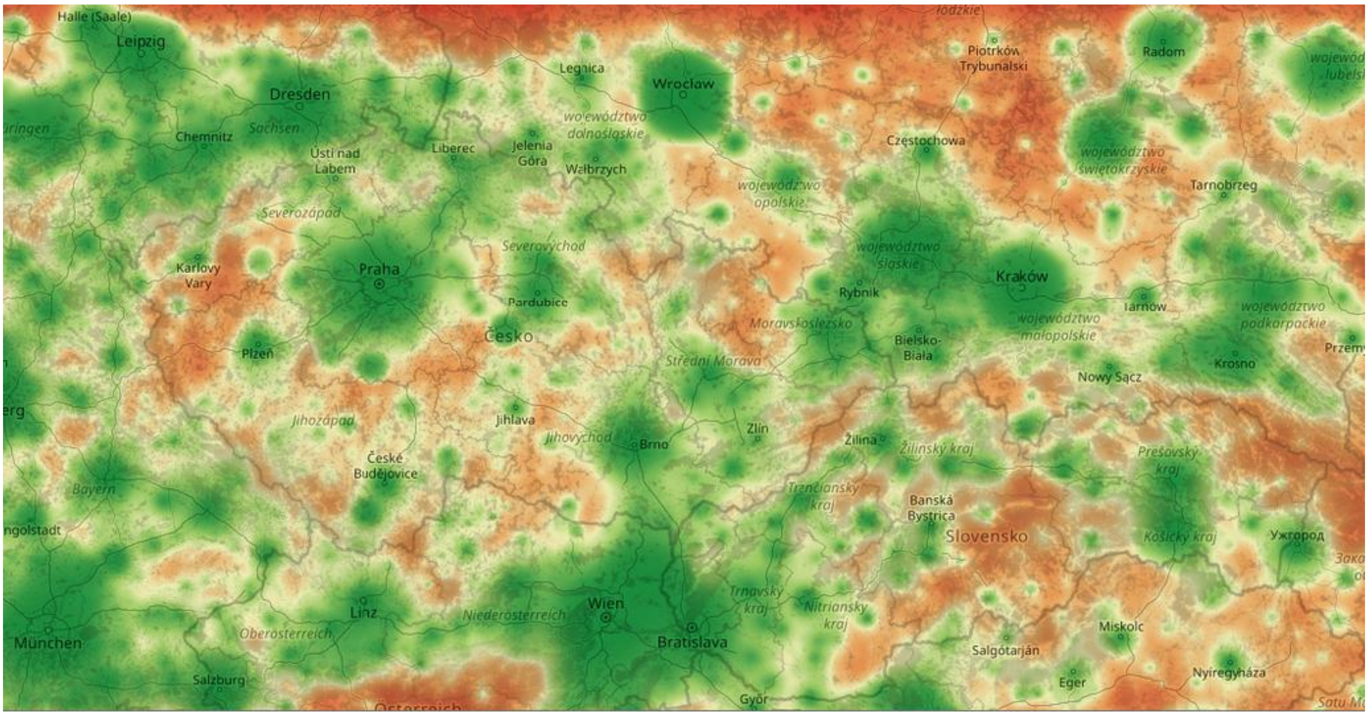

Figure 19: Innovation potential in Central Europe.

3.11 Atlas of Best Practices and e-market linkages

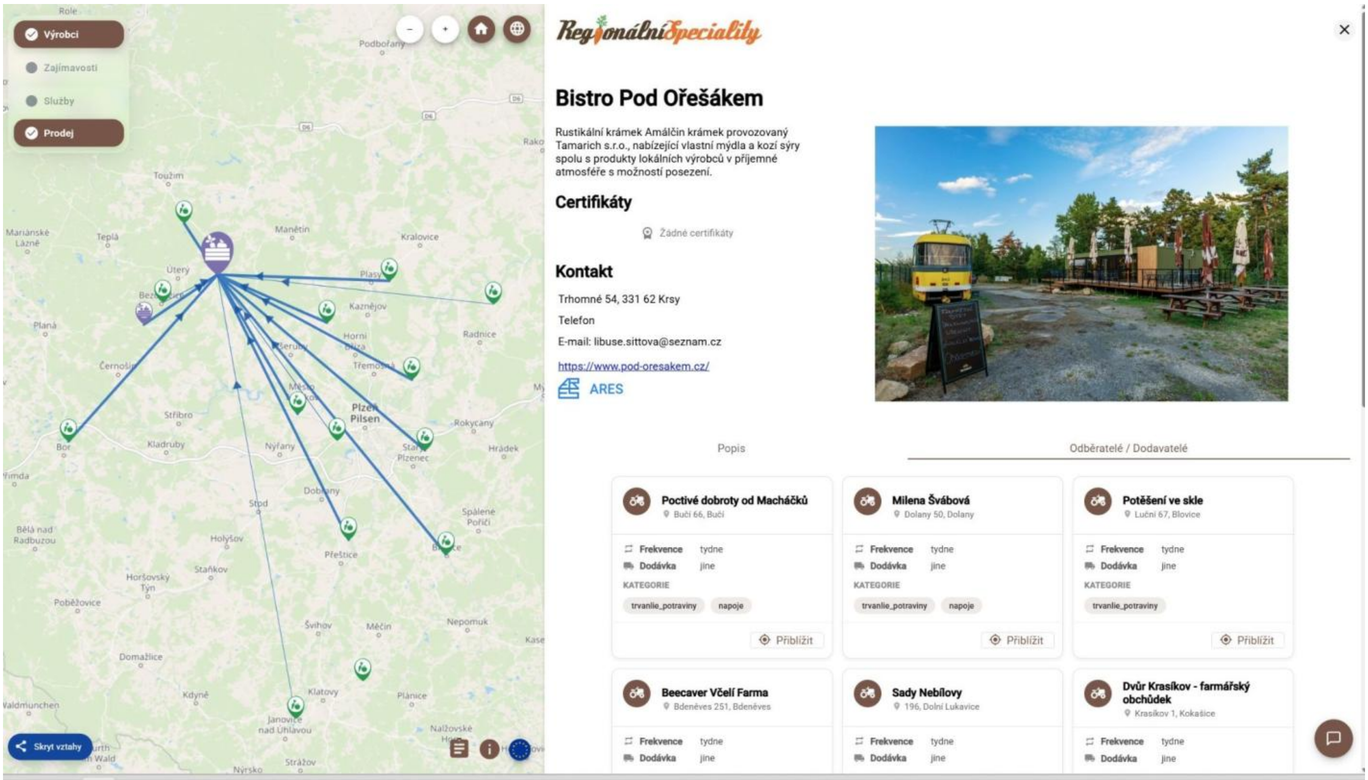

Figure 20: Atlas of Best Practices

The Atlas of Best Practices (Figure 19) serves as a structured repository and presentation environment for examples, cases and transferable practices relevant to the project's rural and regional development themes. Its purpose is to make practical experience visible and reusable, not only as a collection of static case descriptions, but as an organised resource that can support pilot learning, comparison and future adaptation. In the current period, the Atlas has grown substantially in content, which has strengthened its value as a knowledge resource across the Beta environment.

The main implementation progress in the interim period has been the availability of a substantial amount of new content. This growth matters because the usefulness of an atlas-style component depends heavily on the breadth and comparability of the material it contains. Without sufficient content, the structure may be technically sound but operationally weak. The present release has therefore improved the practical relevance of the Atlas by expanding the knowledge base that users can browse and draw upon.

At the same time, development has also included changes to the supporting editorial and presentation logic. Editorial flows have been improved so that content preparation and publication can be handled more consistently. Data model adjustments have been introduced to better reflect the structure of best-practice entries and related information needs. The presentation layer has also been improved, which supports clearer access to the material and makes the Atlas more usable as a front-facing resource rather than only as a back-office repository.

Inputs to the Atlas include case descriptions, structured metadata, thematic classifications, editorial updates and, where applicable, links to market-relevant or regional-food-related resources. Outputs include searchable and browsable best-practice entries, improved thematic presentation and a body of material that can be reused in pilot experimentation, reporting and advisory contexts. The mention of e-market linkages indicates a related role in connecting best-practice knowledge with local-food or regional-market tools where such links are relevant for pilot use. In this sense, the Atlas can serve both as a knowledge repository and as a bridge towards more practice-oriented local value-chain experimentation.

Current limitations are mainly linked to the maturity of these linkages and the need for further clarification of feature scope. While the Atlas content base has clearly expanded and the editorial and presentation functions have improved, the integration of best-practice content with e-market or local-food tool logic still requires further operational definition. The next development steps are therefore to continue content curation, stabilise the adjusted data model, refine editorial workflows further, improve the presentation layer where needed, and specify more clearly how e-market linkages should function in support of pilot experimentation and local or regional market-oriented use cases.

4. Data Management, Knowledge Base and Interoperability

This chapter documents how data, documents, metadata and knowledge resources are managed within the Beta platform .

4.1 Integrated data sources and pilot data flows

A major effort has gone into expanding the database that supports the Beta workflows. This work combines pan-European sources with much more detailed local data. The aim is to show the value of data at both ends of the scale. European-scale data helps to analyse wider regional challenges. It also helps to identify linkages and development potential that may be used in less advantaged or developing regions. Local data serves a different purpose. It supports hyper-local analysis, such as the choice of technical or engineering solutions, and more detailed decisions in daily agricultural management, including field operations and crop treatments.

At the pan-European level, the work has focused mainly on raster data. These datasets are useful for continuous spatial analysis across large areas. At the local level, high-resolution vector data is equally important. It allows more precise representation of parcels, infrastructure, and other site-specific features. The overall approach therefore combines raster layers with supplementary statistical data, mainly from Eurostat and national statistical agencies. Particular attention has also been given to datasets that express the positional advantages of locations, for example through centrality grids and related accessibility indicators.

For continent-wide coverage, several key sources have been integrated. OpenStreetMap provides a broad base of spatial information. This includes service-related points of interest such as places for sleeping, eating, leisure activities, and sightseeing. It also includes infrastructure layers, notably roads, rail lines, and train stations. These data are important for accessibility analysis and for understanding how rural locations connect to services and transport networks.

Additional continent-wide point data comes from SPOI, which offers categorized Smart Points Of Interest. This supports a more structured interpretation of service availability and functional land use. Climate-related information is drawn from WorldClim and ERA5. WorldClim contributes long-term climate surfaces, while ERA5 adds both climate and weather data. Together, these sources support analyses of environmental conditions, seasonal patterns, and climate-related constraints or opportunities.

Eurostat and GISCO provide another essential part of the European data layer. They supply the basic statistical and administrative units used throughout the analysis, including NUTS regions and local administrative units. They also provide infrastructure data such as ports and airports, as well as statistical indicators on population, health, and business demography. Population data is used both in grid form and at NUTS3 level, which allows the workflows to move between more aggregated regional views and finer spatial assessment where needed.

At the national level, the Beta workflows make use of more detailed thematic and administrative datasets. In the Czech Republic, this includes forestry datasets that are relevant for land use and resource management analysis. LPIS data adds parcel-level agricultural information, while cadastre data provides a formal spatial reference for land ownership and boundaries. The Czech Registry of Territorial Identification, Addresses and Real Estates (RÚIAN) contributes further official territorial and address-related information. These national datasets are important because they bring a level of operational detail that is not available from European sources alone.

At the regional level, the workflows also incorporate highly specific sources developed or provided within the pilots. In Central Greece, this includes detailed climate data accessed through APIs. In addition, datasets uploaded directly by pilot regions are used where relevant. These regional contributions are especially important for testing real data flows and for demonstrating how the platform can support local decision-making under real conditions.

Taken together, these integrated data sources show the intended scope of the Beta workflows. The system is not built around a single type of data or a single spatial scale. It is designed to connect broad European context with detailed local evidence. This is necessary because rural development problems rarely sit neatly at one level. Strategic planning needs large-scale context, while implementation depends on local precision. The data integration work reflects that reality.

4.2 Metadata and semantic enrichment

This section describes how metadata and semantic enrichment are used in the PoliRuralPlus Beta platform to support dataset ingestion, catalogue discovery, service selection and tool interoperability. The focus is not on establishing a separate semantic integration layer, but on making metadata operational across platform components. Metadata is treated as an active technical resource that supports data providers, catalogue users, AI-assisted services, analytical tools and external integrations.

In the Beta architecture, Raven is the main component responsible for dataset ingestion, metadata authoring and dataset lifecycle management. It addresses the supply side of the data pipeline by enabling data providers to upload, describe, classify and govern geospatial datasets before they become available to other platform services. This distinguishes Raven from user-facing query and analysis components, which consume the metadata prepared through Raven in order to support discovery, selection and interpretation.

The Raven workflow starts with dataset upload and continues with structured metadata authoring. Data providers are asked to define a human-readable title and description, thematic classification tags, temporal coverage and per-column semantic descriptions. The possibility to provide multilingual descriptions is important for improving semantic search recall and for supporting regional users who may not use the same technical terminology as data publishers. Datasets remain in a draft state until they are explicitly published, which allows the provider to review and improve metadata quality before catalogue exposure.

Semantic enrichment in this context means improving the interpretability and machine usability of metadata records. This includes the addition of thematic tags, spatial and temporal scope, domain-specific terminology, column-level descriptions, links to methodology documents or data dictionaries, and information on access conditions. These elements make datasets more discoverable and also allow other tools to determine whether a dataset is relevant, valid and suitable for a specific analytical task.

Raven also supports the ongoing lifecycle of datasets. After publication, metadata can be updated, supporting documents can be attached or removed, access permissions can be granted or revoked, and the underlying geometry layer can be replaced where this is technically permitted. This is important because metadata is not static. Data quality, ownership, access conditions, spatial coverage, temporal validity and methodological documentation may change during platform operation. The Beta platform therefore treats metadata management as a continuous process rather than as a one-time publication step.

At backend level, Raven is implemented as a Python-based service with a dedicated API for geospatial dataset management, AI-assisted data discovery and integration with external agents or applications. The service exposes REST interfaces for direct client consumption and an MCP endpoint for controlled integration with agent-based workflows. It uses PostgreSQL with PostGIS support for spatial data handling and pgvector support for semantic similarity search. This enables uploaded datasets to be queried through structured database operations as well as through natural-language-oriented discovery workflows.

Metadata is also used for tool and service selection. Each analytical service or external tool can be described through a minimal machine-actionable descriptor. This descriptor should include the tool identifier, name, purpose, domain tags, spatial scope, temporal scope, input schema, output schema, dependencies and known limitations. Such metadata allows the platform to determine which tools are applicable for a specific user request or territorial context. For example, a tool valid only for agricultural parcels should not be offered for an urban cadastral query, while a service with limited temporal coverage should not be selected for a request outside its valid period.

A practical metadata structure (see Table 3 below) for tool and service selection may include the following fields:

Table 3: Abstract description of metadata for tool and service selection

Metadata field	Purpose
Tool or service identifier	Stable internal reference used for routing and invocation
Name	Human-readable label shown to users or other platform components
Purpose	Functional description used to match the tool with user needs
Domain tags	Thematic categories such as rural development, land use, agriculture, forestry, mobility or socio-economic analysis
Spatial scope	Geographic area where the dataset, tool or service is valid
Temporal scope	Time period, update frequency or validity conditions
Input schema	Required input parameters, including geometry, attributes or temporal filters
Output schema	Expected structure of the returned result
Dependencies	Required datasets, database views, services or external APIs
Limitations	Known constraints related to resolution, quality, uncertainty, licence, cost or applicability

This metadata-driven selection process is essential for safe and efficient use of platform tools. It reduces the risk that user-facing services will apply unsuitable datasets or expose irrelevant tools. It also reduces complexity for conversational and analytical components, because only a limited and contextually relevant subset of available resources needs to be considered for a given task.

The same metadata layer supports Magpie-type catalogue discovery. In the current Beta release, the original Magpie function is not treated as a fully separate product, because much of its role has been absorbed into Raven and related discovery services. However, its core purpose remains relevant: helping users find datasets, metadata records and catalogue resources through natural-language-oriented search rather than only through formal filters. The metadata records prepared in Raven therefore provide the basis for more accessible catalogue search, especially for non-specialist users and regional stakeholders.

Metadata is also consumed by Jackdaw and other AI-assisted components. Jackdaw uses metadata records, service descriptions and tool descriptors to identify relevant sources, formulate structured requests and provide source-grounded responses. In this workflow, metadata acts as a control layer. It informs the system what data exist, where they are valid, how they can be accessed, what constraints apply and which processing tools are appropriate. The quality of the user interaction therefore depends directly on the completeness, clarity and consistency of the underlying metadata.

Semantic enrichment also supports provenance and transparency. Dataset descriptions, methodology attachments, temporal validity, ownership information, access rights and processing limitations provide the basis for explaining where an output comes from and under which conditions it should be interpreted. This is particularly important for policy, planning and Regional Action Plan support, where users need to understand not only the final answer or map output, but also the evidence base and constraints behind it.

For interoperability, metadata should be aligned with established geospatial and data catalogue practices wherever feasible. This includes the use of consistent spatial references, temporal descriptions, dataset identifiers, licence statements, access conditions, controlled vocabularies and service-level descriptors. The aim is not to impose a heavy metadata burden on data providers, but to define a pragmatic profile that is sufficient for discovery, reuse, tool selection and auditability within the Beta platform.

The main limitation remains metadata quality. Natural-language search, semantic matching and automated tool selection cannot compensate fully for incomplete descriptions, missing temporal coverage, weak thematic classification or unclear access conditions. For this reason, the Beta platform should include guidance for metadata authors and should progressively validate metadata completeness during the ingestion process. Priority should be given to fields that directly affect discovery, applicability and trust: title, description, spatial coverage, temporal coverage, thematic tags, licence, ownership, update status, methodology documentation and known limitations.

In summary, metadata and semantic enrichment in the PoliRuralPlus Beta platform are implemented as practical operational mechanisms. They support dataset ingestion through Raven, catalogue search through Magpie-type functionality, tool and service selection through metadata descriptors, and evidence-grounded interaction through

Jackdaw and related services. This approach makes metadata an active part of platform execution and not only a documentation layer attached to datasets after publication.

4.3 API layer and interoperability standards

This section describes the API layer and the interoperability standards implemented in the platform. The focus is on practical integration. It covers the APIs used to expose services, the documentation approach, and the data and geospatial standards that support exchange between components. It also includes newer interface patterns used for large language model services.

At platform level, authentication is handled through OAuth. This is provided by the Wagtail CMS layer. In practice, this gives a standard and widely understood mechanism for controlled access to platform services. It supports integration with external tools while keeping user and service access manageable within the overall architecture.
The general API framework is built with Python FastAPI. This provides the basis for implementing service endpoints in a consistent way. It also supports OpenAPI specifications and Swagger UI. As a result, the platform does not only expose APIs, but also documents them in a machine-readable and human-readable form. This is important for maintenance, reuse, and external integration. It reduces ambiguity for developers and makes testing easier during implementation and deployment.

The platform relies on open formats for data exchange. These include JSON, GeoJSON, and WebCSV, alongside other simple tabular formats where needed. The intention is to avoid tightly coupled or proprietary data structures. This matters in a rural innovation context, where systems often need to interact across organisational and technical boundaries, and where lightweight exchange formats remain the most practical option.

The implemented open web standards include JSON, XML, WebCSV, and CSV. These formats serve different purposes. JSON is the main format for modern web APIs. XML remains relevant where legacy services or structured document exchange are involved. CSV and WebCSV provide a simple route for tabular data transfer and inspection. Together, these standards give a workable baseline for interoperability with both newer and older systems.

For geospatial interoperability, the platform supports a set of open geospatial formats and protocols. This includes OGC protocols for spatial data access and querying, such as WMS, WFS, CS-W, and WCS. These standards are still central where map services, feature access, catalogue metadata, or coverage data need to be exchanged between systems. In addition, the platform uses GeoJSON, KML, and GeoCSV. This combination reflects the reality of geospatial integration: no single format is sufficient across all use cases. Some tools expect OGC services, while others work better with file-based or API-based exchange formats.

An important newer element is the use of emerging de facto protocols for LLM-based integration. In particular, the Model Context Protocol (MCP) is used as an interface between the Raven back-end and Jackdaw. This creates a cleaner separation between the back-end services and the agent layer. The practical benefit is that the back-end can be used more independently of the specific agent employed. It also supports more flexible deployment and integration options. This is useful in a fast-moving technical area where interface stability is often weaker than in conventional web services.

The platform also engages with current de facto standards for LLM service access. The main reference point is the OpenAI Chat Completions API pattern, which has become a practical interoperability layer across many tools and providers. This enables connection to OpenAI / ChatGPT, but the same interface approach is also used for other infrastructures. These include a private cloud environment, referred to here as CCSS, where intermediate-sized models are hosted on more realistic and controllable infrastructure. In parallel, interfaces are provided to science cloud resources that host larger open and self-hosted models, including environments such as PSNC and CESNET.

Across these LLM-related interfaces, there is a clear emphasis on sovereignty. This is not only a policy preference. It is a technical and operational concern. The ability to switch between commercial APIs, private cloud deployments, and science cloud infrastructures reduces dependency on a single provider. It also gives more control over data handling, deployment choices, cost exposure, and long-term service continuity. For a European project context, this is especially relevant.

4.4 Provenance, data quality and reuse conditions

This section documents how the PoliRuralPlus Beta platform records and manages data provenance, update frequency, quality constraints, access rights, licensing and reuse conditions. These elements are essential for ensuring that datasets, documents, services and analytical outputs can be interpreted correctly and reused responsibly. In the Beta platform, provenance and reuse information are not treated only as administrative metadata. They are part of the operational data management framework and directly affect publication, discovery, tool selection, access control and user trust.

The platform works with heterogeneous resources, including geospatial datasets, pilot data, documents, catalogue records, external web resources, analytical outputs and service descriptions. These resources differ significantly in origin, structure, update frequency, spatial and temporal coverage, quality and legal status. For this reason, each resource must be accompanied by sufficient descriptive, technical and legal information. Without such information, the platform may still be able to store or display the resource, but its use in evidence-based decision-making, policy support or analytical workflows would be limited.

Provenance describes where a resource comes from, who provided it, how it was generated, when it was created or updated, and whether it has been modified during ingestion or processing. For primary datasets uploaded by project partners or pilot actors, provenance should include at least the data provider, responsible organisation, original data source, date of creation, date of upload, date of last update, processing history and contact point. Where a dataset is derived from another source, the original source and transformation steps should also be recorded. This is particularly important for spatial datasets, where reprojection, geometry correction, attribute harmonisation or filtering may influence later analysis.

In Raven, provenance is linked to the dataset ingestion and lifecycle workflow. When a data provider uploads a dataset, the resource is first kept in a controlled state before publication. During this phase, the provider defines the title, description, thematic tags, temporal coverage and column-level descriptions. Supporting documents, such as methodology notes, data dictionaries or field survey protocols, can be attached to the dataset. These elements create the basic provenance package that other components can use when presenting, searching or processing the data.

The Beta platform should distinguish between several provenance levels. The first level is source provenance, which identifies the original creator, provider and ownership of the resource. The second level is ingestion provenance, which records when and how the resource entered the platform. The third level is processing provenance, which documents changes applied after upload, including geometry replacement, attribute cleaning, format conversion, enrichment or service publication. The fourth level is usage provenance, which records how the dataset has been used by platform services, analytical workflows or pilot applications where such tracking is technically and legally appropriate.

Update frequency is another critical metadata element. Some resources are static, such as historical maps, policy documents, reports or completed survey datasets. Others are periodically updated, for example statistical indicators, land-use layers, environmental monitoring datasets or administrative registers. A third category includes dynamic or near-real-time resources, such as sensor feeds, external service responses or operational datasets. The platform should therefore record whether a resource is static, irregularly updated, periodically updated or dynamically refreshed. Where possible, the expected update interval should be specified, for example daily, weekly, monthly, annual or event-based.

Clear information on temporal validity is required to prevent incorrect interpretation. A dataset may be technically available in the platform but no longer valid for current decision-making. Conversely, historical datasets may be highly valuable when used for long-term trend analysis, baseline reconstruction or comparison with current conditions. For this reason, temporal coverage and update status should be recorded separately. Temporal coverage describes the period represented by the data, while update status describes whether the dataset is current, historical, superseded, experimental or under revision.

Data quality information should be documented in a pragmatic and transparent way. The Beta platform does not certify all uploaded data as authoritative or fully validated. Instead, it should expose the known quality characteristics and limitations so that users and tools can decide whether a resource is suitable for a given purpose. This includes spatial resolution, positional accuracy, attribute completeness, thematic accuracy, temporal resolution, scale of capture, data gaps, uncertainty, methodological limitations and validation status.

For geospatial datasets, quality information should include at least the coordinate reference system, geometry type, spatial extent, geometry validity, scale or resolution, known topology problems and completeness of key attributes. For tabular data, the relevant elements include attribute definitions, units of measurement, missing values, outliers, coding schemes and consistency of identifiers. For textual and documentary resources, quality assessment should focus on source reliability, publication date, author or institution, version, language, completeness and relevance to the intended domain.

Quality constraints should be described explicitly and not hidden from users. Examples include incomplete territorial coverage, outdated source data, inconsistent administrative boundaries, limited temporal coverage, low spatial resolution, uncertain classification, missing attributes, unclear methodology or restrictions on interpretation. In decision-support applications, these constraints are as important as the data values themselves. A map, indicator or analytical result can be misleading if its uncertainty and applicability limits are not visible.

The platform should also support a simple quality status classification (see Table 4, below). This may include the following categories:

Table 4: Quality status classification for data sources

Quality status	Meaning
Validated	Dataset has been checked by the provider or relevant expert and is suitable for declared use cases
Partially validated	Dataset has been checked for basic consistency, but some limitations remain
Experimental	Dataset is used for testing, prototyping or demonstration and should not be treated as authoritative
External source	Dataset originates from an external service or catalogue and quality depends on the original provider
Historical	Dataset is not current but is relevant for retrospective analysis or baseline comparison
Superseded	Dataset has been replaced by a newer version and should be used only for reference or reproducibility

Reuse conditions define how a resource may be accessed, processed, shared and republished. In the Beta platform, this is relevant for open data, partner-provided data, pilot-restricted data, third-party datasets and derived outputs. Each dataset should therefore include a licence or access statement. Where a standard licence is available, it should be recorded explicitly. Where no standard licence exists, the metadata should at least specify whether the resource is open, project-internal, restricted to named users, available for research use only, or subject to approval by the data owner.

Reuse conditions should cover both legal and practical aspects. Legal conditions include ownership, licence, attribution requirements, redistribution rights, commercial-use restrictions and personal or sensitive-data limitations. Practical conditions include format, access method, API availability, required authentication, expected stability, citation requirements and technical restrictions. This is particularly important when data are consumed by other tools, because automatic processing must not bypass legal or organisational constraints.

Access control is part of the reuse framework. Raven supports permissions at dataset level, allowing access to be granted or revoked for named users within the identity management environment. This enables the platform to support both open publication and controlled sharing. It is important that access permissions are not confused with reuse rights. A user may be technically allowed to access a dataset, but the licence may still restrict redistribution, publication, commercial use or use outside the project context.

Derived outputs require special attention. When an analytical product, map, indicator or report is generated from one or more source datasets, the reuse conditions of the source data may affect the reuse conditions of the derived product. The Beta platform should therefore preserve references to input datasets and processing steps where feasible. This supports reproducibility and helps avoid situations where outputs are shared more openly than the underlying data allow.

A practical metadata profile for provenance, quality and reuse should include the following fields:

Table 5: Abstract overview of provenance, quality and re-use metadata

Metadata field	Description
Resource identifier	Stable internal identifier for the dataset, document, service or output
Resource title	Human-readable name of the resource
Data provider	Organisation or person providing the resource to the platform
Original source	Primary source from which the resource originates
Contact point	Responsible contact for questions on content, quality or reuse
Creation date	Date when the original resource was created, if known
Upload date	Date when the resource was ingested into the platform
Last update	Date of the most recent update or replacement
Update frequency	Static, irregular, periodic, dynamic or specified interval
Temporal coverage	Period represented by the content of the resource
Spatial coverage	Geographic area represented by the resource
Processing history	Description of transformations applied during or after ingestion
Quality status	Validated, partially validated, experimental, external, historical or superseded
Known limitations	Main constraints affecting interpretation or use
Licence	Formal licence or usage statement
Access level	Open, project-internal, restricted, private or approval-based
Reuse conditions	Attribution, redistribution, commercial-use and publication conditions

Citation or attribution text	Preferred reference when the resource is reused
Sensitive-data flag	Indication of personal, confidential, commercially sensitive or otherwise protected content

This profile should be implemented in a proportionate way. Not all resources will have complete information at the time of ingestion, especially when they come from external providers or pilot contexts. However, the platform should clearly distinguish between “unknown”, “not applicable” and “not provided”. This avoids false assumptions and supports later metadata improvement.

Provenance and quality information also have direct operational value for AI-assisted and analytical services. When a user asks a question, the system should prefer resources that are relevant, current, sufficiently described and valid for the requested territory or time period. If a dataset is experimental, outdated or restricted, this should influence how it is selected and how outputs are presented. In this sense, provenance and quality metadata act as a control mechanism that reduces the risk of inappropriate data use.

For pilot work and Regional Action Plan support, transparency of evidence is essential. Users need to know whether an output is based on official statistics, partner-provided pilot data, external web information, expert interpretation or experimental analysis. The platform should therefore make provenance visible wherever outputs are used for planning, foresight, stakeholder engagement or policy discussion. This does not require excessive technical detail in the user interface, but it does require clear references to data sources, update status and limitations. In the run-up to D4.3, a testing and validation plan will be created in cooperation with WP2 and 5 to ensure that the remaining effort is spent where it will make the greatest impact.

The main implementation challenge is consistency. Data providers may describe datasets with different levels of detail, different terminology and different assumptions about quality and reuse. The Beta platform should address this through metadata guidance, controlled fields where appropriate, and validation checks during ingestion. Priority should be given to fields that directly affect trust and reuse: source, provider, temporal coverage, update status, licence, access level, known limitations and quality status.

In summary, provenance, data quality and reuse conditions form a core part of the PoliRuralPlus Beta data management framework. They support responsible publication, controlled access, transparent interpretation, lawful reuse and evidence-based operation of platform services. The objective is not only to store datasets, but to make their origin, validity, limitations and reuse conditions explicit enough for reliable use by pilots, project partners, external tools and future adopters.

4.5 Data Cubes, GeoRAG and RAG-Knowledge Graph integration

The Beta platform will progressively introduce a Data Cube-oriented data access pattern for multidimensional geospatial resources. This is relevant for Earth observation, climate, land-use, meteorological, socio-economic and sensor-derived datasets where the analytical unit is not a single file, but a structured combination of spatial, temporal and thematic dimensions. Data Cubes will support more efficient querying, subsetting, aggregation and comparison of indicators needed for rural-urban foresight, territorial diagnostics and Regional Action Plan preparation.

GeoRAG will extend classical document-oriented Retrieval-Augmented Generation by incorporating spatial retrieval and geospatial reasoning into the question-answering process. Instead of retrieving only text fragments, the GeoRAG workflow should be able to identify relevant territories, metadata records, map layers, STAC assets, OGC services, Data Cube variables and analytical endpoints. This approach is essential for questions where the answer depends on location, scale, time period, spatial relation, indicator value or comparison between territories as shown in Table 6, below.

Table 6: Status of Data Cube, GeoRAG and Knowledge Graph integration in the D4.2 Beta platform

Integration element	Purpose in D4.2 Beta	Main implementation evidence to collect	Expected extension toward final platform
Data Cubes	Access to multidimensional spatial, temporal and thematic data for indicators and scenarios.	List of datasets, dimensions, access services, sample queries and screenshots.	Operational connection to analytical workflows, dashboards and AI agents.
GeoRAG	Spatially-aware retrieval from metadata, map layers, STAC assets, OGC services and analytical outputs.	Test queries, retrieved spatial resources, response examples and limitations.	Improved geospatial reasoning, ranking and explanation of retrieved evidence.
RAG + Knowledge Graphs	Semantic grounding of RAG outputs through entities, relations, vocabularies and controlled concepts.	Initial graph model, vocabularies, entity-linking examples and source-to-answer traces.	More robust multilingual advisory services and transparent provenance of generated answers.

5 Technical implementation environment and deployment model

This chapter describes the technical implementation environment and deployment model used during the Beta operation of the project services. The deployment workflow is based on common development in GitHub, continuous integration using GitHub Actions, manual management of sensitive configuration through GitHub Secrets, and deployment to a private OpenStack cloud environment hosted by the Czech partners in the project.

The overall model is intended to support a practical and reproducible deployment process while keeping operational complexity at a manageable level. The approach combines containerised services, controlled build pipelines, clear separation between source code and configuration, and a shared OAuth2-based security model for trusted communication between project components.

5.1 Development environment and repositories

Development is organised around shared GitHub repositories. Each technical component is maintained in one or more repositories depending on its functional scope, ownership, and deployment needs. The repositories contain the application source code, Docker build definitions, infrastructure-related deployment scripts, automated test definitions, and documentation required by technical partners.

GitHub is used as the common collaboration platform for source code management. Standard Git workflows are applied, where development is carried out in feature branches and changes are integrated into the main development branch through pull requests. Pull requests provide a common mechanism for code review, discussion, traceability, and quality assurance before changes are merged.

Issue tracking is also handled through GitHub. Issues are used to describe bugs, technical tasks, feature requests, deployment problems, and integration needs between partners. Where appropriate, issues are linked to pull requests so that implementation work can be traced back to the corresponding technical requirement or operational problem.

Versioning follows a practical release-oriented model. Stable versions are identified through Git tags or release branches, depending on the maturity of the component. During Beta operation, versioning is particularly important because deployed services may be consumed by several other entities in the project. Each deployment should therefore be traceable to a specific repository revision, container image version, and configuration state.

Collaboration between technical partners is supported by common repository conventions. These include consistent naming of branches, clear commit messages, pull request templates where relevant, and README files describing build, configuration, and local development procedures. This makes it easier for partners to understand how a service is developed, tested, and deployed, even when they are not the primary maintainers of that component.

The development environment is designed to be reproducible. Developers can run the services locally using Docker or Docker Compose, with local configuration files used for development-only settings. Sensitive credentials are not stored in the repository. Instead, local developers provide their own credentials through local environment files or development-specific secret stores that are excluded from version control.

5.2 Containerisation and deployment model

The services are containerised using Docker. Each service is packaged as an independent container image containing the application code, runtime dependencies, and service-specific startup configuration. This approach provides a consistent execution environment across local development, continuous integration, test deployment, and Beta operation.

The deployment target is a private cloud environment hosted by the Czech partners in the project. The environment provides the virtual computing, networking, and storage infrastructure required to operate the project services. On top of this infrastructure, the deployment model may use virtual machines, Docker-based service hosts, or orchestration mechanisms depending on the needs and maturity of each component.

The standard deployment model separates services according to their technical responsibility. A typical service stack may include:

- application API services;
- worker services for asynchronous processing;
- database services or externally managed database endpoints;
- vector database services;
- document conversion or ingestion services;
- embedding or semantic processing services;
- reverse proxy and routing components;
- monitoring and logging components.

The separation of services makes it possible to update, restart, scale, and monitor components independently. For example, a document conversion worker can be scaled separately from an API service if the workload is dominated by file ingestion. Similarly, a semantic processing service can be allocated more compute resources than a lightweight metadata API.

Container images are built automatically through GitHub Actions. When code is merged into the relevant branch, the CI workflow runs the defined test and build steps. If these steps complete successfully, the workflow builds a production-ready Docker image and publishes it to the configured container registry. The image is tagged with a version or commit identifier so that each deployment can be traced back to the exact source code state.

Deployment to the OpenStack-hosted environment is performed through controlled GitHub Actions workflows. These workflows may be triggered by changes to selected branches, by tagged releases, or manually by an authorised project developer. Manual triggering is useful during Beta operation, where deployments may need to be coordinated with integration testing, demonstrations, or partner availability.

The deployment process retrieves the required container image from the registry and updates the running service on the OpenStack infrastructure. The exact update mechanism depends on the service type and hosting configuration, but the general principle is that deployment should be repeatable, traceable, and based on versioned container images rather than manual modification of running servers.

The architecture is designed to allow horizontal scaling where appropriate. Stateless API services and asynchronous worker services can be replicated when workloads increase. More specialised components, such as databases or vector stores, require more careful scaling and backup strategies, but they are still deployed as clearly separated services with explicit configuration and operational responsibilities.

5.3 Service configuration and operational workflow

Service configuration is separated from application source code. This is a key principle of the deployment model, because the same container image should be usable in different environments, such as local development, integration testing, Beta operation, and demonstration deployments. Runtime configuration is provided through environment variables and deployment-specific configuration files.

Non-sensitive configuration values include service URLs, public API endpoints, database hostnames, feature flags, logging levels, model names, queue names, and operational parameters. These values are managed as part of the deployment configuration and injected into the containers at startup.

Sensitive configuration values are handled through GitHub Secrets. This includes API keys, database passwords, service credentials, OAuth2 client secrets, deployment keys, and registry credentials. Secrets are entered manually into the GitHub repository or organisation settings by authorised maintainers. They are not committed to the source repository and are not included in container images.

During deployment, GitHub Actions reads the required secrets and passes them securely to the build or deployment process. The secrets are used only where needed, for example to authenticate against a container registry, connect to the OpenStack environment, configure service credentials, or provide OAuth2 client configuration. This approach keeps secret management simple while maintaining a clear separation between code and sensitive operational data.

The CI/CD workflow consists of the following standard stages:

1. A developer creates a feature branch and implements a change.
2. The change is submitted as a pull request in GitHub.
3. The pull request is reviewed by relevant technical partners.
4. Automated checks are executed through GitHub Actions.
5. After approval, the change is merged into the target branch.
6. GitHub Actions builds and tags the Docker image.
7. The image is pushed to the configured container registry.
8. A deployment workflow updates the relevant service in the OpenStack environment.
9. The service is restarted or replaced using the defined deployment procedure.

10. Logs, health checks, and service endpoints are reviewed after deployment.

For Beta operation, deployments should be coordinated and documented. Each deployment should record the deployed service, image version, target environment, responsible partner, and any relevant configuration changes. This makes it easier to identify the source of operational issues and to roll back to an earlier version if needed.

5.4 Shared authentication

Security between project components is based on an OAuth2 workflow using a common circle-of-trust model. Tokens are issued by the OAuth2 provider in the PoliRuralPlusHub environment. Participating services and entities trust this provider and use its issued tokens to authenticate API calls and control access to protected resources.

In this model, a user or service first obtains an access token from the trusted provider. The token is then included in requests to participating APIs, normally as a bearer token in the HTTP Authorization header. Receiving services validate the token before processing the request. Validation may include checking the token signature, issuer, audience, expiry time, scopes, and roles.

This provides a common security model across the project. Instead of each service maintaining its own isolated user and credential system, services rely on the shared OAuth2 provider and apply local authorisation rules based on the claims contained in the token. This simplifies integration between project entities and supports secure service-to-service communication.

The circle-of-trust model is especially important because several independently developed services may need to interact during Beta operation. For example, a front-end application, document ingestion service, semantic search API, and project hub component may all participate in a single workflow. By using tokens issued by the trusted provider, each component can verify the identity and permissions of the caller in a consistent way.

5.5 Observability

Operational monitoring is based on a combination of application logs, container logs, health endpoints, and infrastructure-level monitoring in the OpenStack environment. Each service should expose a basic health check endpoint where appropriate. This endpoint can be used by deployment scripts, reverse proxies, or monitoring tools to verify that the service is running and able to respond.

Logging should be structured where possible and should include sufficient information to diagnose operational issues without exposing sensitive data. Logs should identify the service, request context, error category, and relevant technical details. Access tokens, passwords, API keys, and personal data should not be written to logs.

5.6 Development procedures

During Beta operation, technical partners should monitor the following aspects:

- service availability;
- failed deployments;
- container startup errors;

- authentication and authorisation failures;
- API response times;
- queue lengths for asynchronous workers;
- database and vector database connectivity;
- storage consumption;
- CPU, memory, and where relevant GPU usage.

Where distributed workflows involve several services, traceability is important. A request identifier or correlation identifier should be propagated between services where feasible. This makes it easier to follow a single operation across the API layer, worker services, document conversion, embedding generation, and data retrieval.

The operational workflow should remain simple and transparent. The preferred approach is to use GitHub as the main control point for development, review, automation, and deployment. GitHub Actions provides the repeatable automation layer, GitHub Secrets provides the controlled secret store, Docker provides consistent service packaging, and the Czech-hosted OpenStack private cloud provides the operational infrastructure for Beta deployment.

6. Pilot Use and Testing of the Beta Platform

This chapter documents how the PoliRuralPlus Beta platform has been used and tested across the nine core pilot regions. It focuses on pilot onboarding, pilot-specific service configurations, early operational feedback, usability observations and corrective actions arising from pilot implementation. The chapter is limited to core pilot implementation and does not address external validation activities.

6.1 Testing methodology and pilot onboarding

The Beta testing methodology follows an operational validation logic. Each pilot is treated as a heterogeneous testbed with different thematic priorities, governance arrangements, data availability and user capacities. Testing therefore assesses whether the platform can be configured for different territorial contexts rather than whether all pilots follow one uniform workflow.

Pilot onboarding covered five main steps: identification of the regional planning challenge; selection of relevant Beta services; registration or configuration of pilot workspaces and user roles; preparation or ingestion of pilot data and metadata; and iterative testing through meetings, workshops, demonstrations and practical planning tasks.

The main services tested in the pilot context included Hub4Everybody as the web and collaboration entry point, MAATool for structured multi-actor coordination, dashboards for indicator monitoring, MapWhiteboard for participatory spatial work, Jackdaw for plain-language GeoAI interaction, Raven for data and metadata ingestion, and selected advisory or analytical services for pilot-specific evidence synthesis.
The MAATool provided a structured coordination layer for the multi-actor work. In the Beta configuration, the PoliRuralPlus workspace includes more than 300 registered users, 40 work entities, approximately 150 classified stakeholder profiles, 49 specified objectives and 40 input progress records. These figures show that the Beta platform has moved beyond isolated tool demonstrations and is already used as a coordination environment for pilot and work package activities.

The testing methodology assessed functional correctness, ease of onboarding, data and metadata readiness, interoperability between components, language and interface usability, responsiveness under realistic network conditions, relevance of outputs for planning tasks, and the degree to which users could understand the origin and limitations of results.

6.2 Pilot-specific use cases and early feedback

The nine core pilots applied different combinations of platform services. This diversity demonstrates whether the Beta architecture can support selective reuse of tools rather than only a single end-to-end platform deployment. Table 7 below summarises the main pilot configurations and early feedback.

Table 7. Core pilot regional deployments and thematic applications

Pilot and territory	Core services deployed	Spatial and thematic focus	Operational milestones and early feedback
Pilot 1 - Monaghan (IE)	Dashboard; MAATool; Hub4Everybody	Circular economy governance; construction and demolition waste tracking; RAP coordination.	Standardised circularity indicators and mapping of waste producers and municipal recycling centres support regional circularity monitoring and policy enforcement.
Pilot 2 - Slovakia	Dashboard; MapWhiteboard; Policy Options Explorer; Hub4Everybody	Participatory foresight; rural-urban boundary definition; Slovak Vision 2040 coordination.	Regional roundtables with the Ministry of Interior and the Slovak CAP Network supported policy-platform linkage and spatial monitoring in the national planning context.
Pilot 3 - Central Greece	MAATool; Jackdaw GeoAI; Hub4Everybody	AKIS coordination; agrifood-tourism connectivity; digital-skills development.	The platform strengthened regional advisory capacity and helped structure cooperation among producers, technology providers, researchers and local innovation actors.
Pilot 4 - Italy	Dashboard; MapWhiteboard; Hub4Everybody	Community engagement; participatory local governance; rural-urban innovation planning.	Collaborative map work and vector drawing supported municipal dialogue on local territorial development and community engagement.
Pilot 5 - Mallusjoki (FI)	MAATool; Advisor services; Jackdaw; Raven metadata services	Rural events ecosystem; community branding; zoning and energy diagnostics.	Municipal-level spatial analysis combined open geospatial datasets and local planning needs for the rural events and activities ecosystem.
Pilot 6 - Czechia-Germany cross-border	Jackdaw GeoAI; Hub4Everybody; IoT and AI community tools	Cross-border attractiveness mapping; spatial connectivity; Digital Commons infrastructure.	High-resolution 1 x 1 km attractiveness grids supported comparison and coordination across southwestern Bohemia and eastern Bavaria.
Pilot 7 - Spain	Jackdaw GeoAI; MAATool; MapWhiteboard; Hub4Everybody	Cultural heritage corridor governance; rural-urban economic linkages; participatory analysis.	Roman archaeological parks at Segobriga, Valeria and Ercavica were mapped as a coordinated heritage corridor supporting rural-urban economic linkages.
Pilot 8 - Vidzeme (LV)	Dashboard; Hub4Everybody; data and AI community tools	Regional KPI monitoring; inter-municipal business support; evidence-based governance.	Standardised regional development metrics and indicator workshops supported local entrepreneurship and business-support monitoring.

Pilot and territory	Core services deployed	Spatial and thematic focus	Operational milestones and early feedback
Pilot 9 - Malta	MAATool; Dashboard; Hub4Everybody	Agricultural digitisation; youth-led local innovation; precision-farming capacity building.	Training and regional asset mapping supported local agricultural digitisation and capacity-building activities in a fragmented land context.

Across the pilots, the main feedback confirms the relevance of combining structured stakeholder management, spatial visualisation, indicator monitoring and conversational access to geospatial knowledge. The table also shows why the platform must remain configurable: some regions primarily require governance and stakeholder tools, while others need spatial diagnostics, local data ingestion or AI-assisted interpretation.

6.3 Usability, performance and functional observations

The main positive usability observation from the pilot context concerns the combination of map-based interaction with plain-language dialogue. This combination supports users who need territorial evidence but do not work routinely with advanced GIS environments. It also helps connect spatial evidence with planning questions and stakeholder discussions.

The MAATool proved useful for transforming multi-actor engagement into structured, traceable workflows. It supports stakeholder registration, classification, role allocation, objective management and progress tracking. The Ramon AI assistant reduced administrative effort by extracting stakeholders and objectives from pilot documents for later user validation.

Raven contributed to pilot testing by providing a dataset and metadata abstraction layer. Its role is important because pilots need to register or describe local datasets, clarify their spatial and temporal validity, and expose them to other platform tools without manual code modifications.

Dashboards and indicator services provided a stable entry point for monitoring and baseline description. MapWhiteboard supported collaborative spatial interpretation, especially where pilot activities required discussion of boundaries, corridors, land-use issues, infrastructure or local assets. Hub4Everybody provided the common environment for communication, visibility and access to platform-related resources.

The most important pilot limitations relate to heterogeneity of local data, uneven metadata completeness, different levels of user technical capacity, and the need to keep outputs understandable for non-technical stakeholders. These limitations confirm the need for metadata guidance, simple user interfaces, clear source references and progressive onboarding support.

6.4 Issues, limitations and corrective actions

Pilot testing identified a limited number of recurrent issues that should be addressed before wider-scale deployment. These are not treated as failures of the Beta platform, but as evidence for the final development roadmap.

- Improve guidance for pilot data preparation, especially metadata completeness, spatial coverage, temporal coverage, licences and known limitations.
- Maintain configurable service bundles, because pilots differ in their need for dashboards, stakeholder tools, spatial analysis, conversational support and metadata services.
- Optimise map interaction and performance for practical pilot use, including large spatial layers and rural connectivity constraints.
- Strengthen multilingual and place-name handling for pilot territories where users interact in national or regional languages.
- Ensure that user-facing outputs include clear source references, data validity information and limitations relevant to planning decisions.

The iterative co-development approach applied in PoliRuralPlus enables pilots, technical partners and stakeholders to continue contributing operational observations, usability feedback, governance insights and workflow-improvement recommendations. This process supports maturation of the Beta platform towards a more interoperable, user-oriented and territorially adaptable platform ecosystem.

7. Contribution to Decision-Making and Foresight

This chapter explains how the Beta platform contributes to decision-support, foresight and Regional Action Plan preparation in the core pilot context. The contribution is based on the combined use of structured stakeholder information, spatial evidence, metadata-supported data discovery, dashboards, conversational analysis and participatory planning tools.

7.1 Support to RAP development and territorial analysis

Regional Action Plans require a combination of territorial evidence, stakeholder priorities, scenario discussion and practical implementation pathways. The Beta platform supports this process by making relevant data and knowledge easier to discover, by allowing spatial and thematic evidence to be interpreted collaboratively, and by supporting the translation of analytical findings into planning questions and action-oriented discussions.

Dashboards and indicator services support monitoring and baseline description. MapWhiteboard supports collaborative interpretation of spatial patterns. Jackdaw supports user-oriented access to geospatial and policy evidence. Raven and metadata services improve the availability and reliability of local datasets. MAATool connects the analytical layer with stakeholders, roles, objectives and progress records.

For RAP development, the most important platform contribution is therefore not one isolated analytical model. It is the integration of several functions into a workflow where evidence, actors, territorial priorities and implementation logic can be managed together.

7.2 Collaborative stakeholder engagement

Collaborative stakeholder engagement in PoliRuralPlus is operationalised through a combination of participatory methods and WP4 digital services. WP5 coordinates stakeholder mapping, workshops, RAP discussions, bilateral meetings and pilot onboarding activities, while WP4 tools provide the digital environment for structured coordination, spatial collaboration and evidence sharing.

Table 8. Evidence supporting collaborative stakeholder engagement in the D4.2 Beta platform

Engagement function	Relevant platform component or activity	Contribution to decision-making and foresight
Stakeholder coordination	MAATool and Ramon AI assistant	Stakeholders, objectives, roles and progress records are structured in a common workspace, reducing manual coordination effort.
Spatial collaboration	MapWhiteboard, dashboards and map-based services	Users can jointly discuss territories, edit or interpret spatial information and connect map views with planning tasks.
Evidence access	Jackdaw, Advisor services and metadata-driven discovery	Non-specialist users can ask questions and access territorial evidence through natural-language interaction.

Engagement function	Relevant platform component or activity	Contribution to decision-making and foresight
Pilot participation	Pilot workshops, roundtables, demonstrations, trainings and regional meetings	Engagement is tested in practical regional settings and connected to RAP preparation and local implementation needs.
Communication and uptake	Hub4Everybody and project dissemination channels	Pilot knowledge, events and reusable outputs are made visible for wider uptake and exploitation.

The evidence shown in Table 8 above, demonstrates that collaborative stakeholder engagement in PoliRuralPlus extends beyond isolated consultation activities. It operates as a multi-actor pilot ecosystem combining participatory governance, collaborative mapping, AI-supported analytical workflows, dissemination activities and structured stakeholder coordination.

A key observation is the diversity of stakeholder participation across the pilots. Public authorities, businesses, academia, NGOs, citizens, farmers and community actors contribute jointly to pilot activities, strengthening regional ownership, legitimacy and practical relevance. The Multi-Actor Approach therefore functions as an operational coordination framework connecting pilots, stakeholders, digital services and evidence-based planning processes.

7.3 AI-supported evidence synthesis and scenario exploration

AI-supported evidence synthesis contributes to foresight by helping users formulate planning questions, retrieve relevant evidence, compare territorial situations and explore possible development pathways. In the Beta platform, this function is provided mainly through Jackdaw, Advisor-type services and metadata-driven access to datasets and documents.

The Beta results show that AI-supported outputs are most useful when they are treated as structured assistance rather than as automatic decisions. Outputs should be interpreted with reference to data provenance, territorial scope, temporal coverage and known limitations. This is particularly important where results are used for policy prioritisation, investment planning or public consultation.

For the final release, AI-supported evidence synthesis should be governed by four principles: metadata-based selection of valid resources; explicit spatial and temporal scope; clear communication of uncertainty and limitations; and human oversight for strategic or sensitive decisions.

8. Cooperation with FSTP/Open Call Partners

This chapter documents how partners selected through the PoliRuralPlus Open Calls and Financial Support to Third Parties contribute to the Beta platform. The FSTP activities are treated as a separate extension and validation mechanism for testing, localising and improving the platform under real operational conditions.

The Open Call structure adds three complementary functions to the project. Mobilise activities support community-led ideation and digital literacy. Develop activities support localised software prototypes and service customisation. Enhance activities support targeted validation and stress-testing of Jackdaw and related platform components.

8.1 Role of FSTP partners in Beta validation and extension

FSTP partners contribute to the Beta platform by validating services with new user groups, identifying usability and data limitations, connecting local datasets and sector-specific knowledge, configuring tools for domain-specific purposes, and generating practical recommendations for the final release. Table 9 below summarises the main Open Call cases described in the current evidence base.

Table 9: Open Call and FSTP validation and extension cases

Call strand	Project or activity	Territory	Platform services or assets used	Validation or extension contribution
Mobilise	RURBAN Challenge / Smart Agritourism Challenge	Central Greece	Jackdaw; MapWhiteboard; Attractiveness Explorer	Community-led ideation for agrifood-tourism concepts; seven teams prepared proposals and selected concepts received mentoring.
Develop	DASK	Kythera and Antikythera, Greece	Localised Jackdaw; digital gateway; marketplace and knowledge base	Digital-terroir agronomic assistant, training support and product listing for olive groves, apiculture, herbs, wine and local products.
Develop	MALTESE	Malta	Jackdaw; MapWhiteboard; rural-urban e-market	Multilingual decision support, circular agri-business training and short food supply-chain concepts for rural youth, farmers and educators.
Develop	Avoin Map	Mallusjoki, Finland	Energy Map; Zoning Map; Activity Map and Calendar	Open-source planning services for building-energy profiles, carbon sinks, land-use editing, event data and community-building revitalisation.

Call strand	Project or activity	Territory	Platform services or assets used	Validation or extension contribution
Enhance	AURA	Central Macedonia, Greece	Jackdaw usability and planning validation	Stress-tested planning use cases and documented latency, map-editing, linguistic, spatial-grounding and transparency constraints.
Enhance	SYLVAN-AI	Netherlands	Jackdaw for forestry and land management	Validated plain-language forest-management interaction using satellite, protected-area and cadastral layers.
Enhance	J-Verse	Cyprus	Jackdaw; Open-Meteo; PATROLIoT sensors	Compared model-based Jackdaw outputs with Open-Meteo and local sensor telemetry, showing stronger alignment with regional API data than point-level measurements.
Enhance	e-Mooji	Germany, Scotland and Spain	Jackdaw for pasture and landscape management	Mapped pasture boundaries, livestock movement and carrying-capacity information into farmer-oriented advisory outputs.
Enhance	FloodAI+	Portugal	Jackdaw for hydrological risk communication	Translated flood-hazard maps, satellite water-level information and vulnerability indicators into localised risk advisories and collected workshop feedback.

The table shows that FSTP projects do not only demonstrate existing tools. They expose the operating conditions under which the tools must function: local language, local datasets, sector-specific workflows, non-specialist users, weak connectivity and specific trust requirements.

8.2 Collection of FSTP partner needs and technical requirements

FSTP partner needs were collected through proposal documentation, mid-term reporting, technical consultations, prototype demonstrations and direct interaction with platform partners (see Table 10 below). The Brussels consultation meetings on 24-25 March 2026 provided a structured peer-review environment where FSTP teams presented progress, integration needs and validation findings.

Table 10. Examples of FSTP technical requirements and validation findings

Project	Main requirement or data need	Integration and validation status	Implication for the Beta platform
AURA	Slow or unstable rural networks; Greek place names; map-area editing; non-technical user expectations	Structured personas and walkthroughs identified latency, input sensitivity, spatial mismatch and interface transparency issues.	Improve caching, vector tiles, area-editing workflows, linguistic normalisation and user-facing explanation of internal processes.
J-Verse	Open-Meteo comparison; PATROLIoT temperature and humidity telemetry; environmental validation	Ten-day comparison indicated close Jackdaw alignment with API weather data and larger deviation from local point-level sensors.	Clearly distinguish regional model-scale responses from local sensor observations and document the validity level of each source.
DASK	Local agricultural knowledge, marketplace data, farmer training needs and Kythera-specific conditions	Localised Jackdaw as an agronomic assistant in a digital gateway for remote island communities.	AI guidance should be configured with local agronomic context and linked to actual user capacity and data availability.
MALTESE	Bilingual support, youth training, fragmented land, local producers and circular agri-business workflows	Combined Jackdaw, MapWhiteboard and e-market functions during training and dissemination activities.	Maintain multilingual decision support and connect spatial planning, circular-business design and marketplace workflows.
Avoin Map	Energy, zoning, carbon, photogrammetry, local event and community-building data	Delivered Energy Map, Zoning Map and Activity Map/Calendar with local planning and drone/laser-scanning inputs.	Support local datasets and domain-specific planning functions through documented metadata profiles and exportable spatial outputs.
e-Mooji	Pasture boundaries, livestock movement, carrying capacity and farmer feedback	Validated pasture and landscape-management advice across several European landscapes.	Local livestock databases must be semantically described and linked to reliable spatial boundaries and management assumptions.
FloodAI+	Flood hazard maps, satellite water-level data, vulnerability indexes and stakeholder risk communication	First workshop collected feedback from regional participants and tested simplified flood-risk communication.	Translate technical hazard layers into concise, localised and actionable messages for civil protection, farmers and municipalities.

The requirement collection process should record a common set of information for each FSTP solution: territorial challenge, target users, platform services used, input datasets, metadata requirements, integration pattern, hosting assumptions, language needs, data protection constraints, expected outputs, testing method, user feedback and required support from WP4 or other platform partners.

8.3 Open Call solutions as extensions of the PoliRuralPlus platform

Open Call solutions can be grouped into three extension patterns. Community-led ideation extensions use the platform to support concept development and collaboration. Localised service prototypes adapt platform assets to concrete territorial challenges. Validation and stress-testing extensions examine whether Jackdaw and related services remain useful, understandable and reliable in specialised domains.

From a platform perspective, these Open Call activities extend PoliRuralPlus in four technical directions: local data onboarding, domain-specific user interfaces, metadata-driven service selection, and practical validation of conversational spatial analysis. They also provide evidence for the final platform roadmap (see Table 11 below) by identifying required improvements in language processing, map interaction, performance, local data integration and user-facing transparency.

Table 11: Open Call validation comments and corrective actions

Feedback theme	Open Call observation	Corrective action for final platform
Performance and rural connectivity	AURA and other field-oriented tests reported delays for complex spatial queries under weak or unstable connections.	Introduce stronger client-side caching, lighter map payloads, vector tiles and clearer progress/status messages.
Map interaction and area editing	Users had difficulties modifying selected areas when workflows required redrawing polygons from the beginning.	Allow direct modification of selected polygons and make area-selection tools more visible.
Linguistic robustness	Greek place names, accents, spelling variations and morphology affected query interpretation.	Add local gazetteers, fuzzy matching, accent-tolerant normalisation and multilingual prompt handling.
Local data and sensor scale	J-Verse showed that regional meteorological API responses may differ from point-level physical sensors.	Expose data scale, source type and spatial/temporal validity before presenting results as local evidence.
User-facing transparency	Technical terms such as backend component names or database operations can increase cognitive load.	Encapsulate internal processes and present concise user-facing explanations with source and limitation notes.

8.4 Support package provided by WP4 and platform partners

WP4 and platform partners provide a support package that enables FSTP partners to use the Beta platform effectively and to provide technically useful feedback (see Table 12 below). The support package remains modular, because different FSTP projects use different platform assets and have different levels of technical capacity.

Table 12: Support package for FSTP and Open Call partners

Support area	Content of support	Expected platform benefit
Technical access and orientation	Access to selected Beta services; explanation of architecture, APIs, MCP options and hosting assumptions.	FSTP partners can test components without treating the full platform as a monolithic system.
Data and metadata support	Guidance on Raven metadata, spatial validity, licences, temporal coverage and known limitations.	Local datasets become reusable and less risky for downstream tools.
Tool configuration and testing	Support for Jackdaw, MAATool, MapWhiteboard, Hub4Everybody and domain-specific service configuration.	Testing is connected to real territorial workflows and not only to demonstrations.
Usability and evidence collection	Templates for recording user tasks, misunderstood prompts, latency, missing functions and interface barriers.	Feedback can be translated into platform roadmap actions.
Communication and exploitation linkage	Coordination with WP6/WP7 for visibility, dissemination and sustainability assessment.	FSTP outputs can be positioned as reusable components or documented pilot-specific extensions.

8.5 Integration, testing and feedback workflow

The recommended workflow for integrating FSTP outputs into the Beta platform consists of seven stages (see Table 13 below). The workflow is intended to keep prototype activity connected to the consolidated platform roadmap.

Table 13. Recommended integration, testing and feedback workflow

Step	Workflow stage	Main action
1	**Onboarding and scope confirmation**	Clarify territorial problem, target users, platform services and expected outputs.
2	**Data and metadata review**	Check licences, coverage, quality, access conditions and required supporting documents.
3	**Integration pattern definition**	Select existing services, Raven ingestion, Jackdaw configuration, API/MCP connection or soft-linked prototype.

Step	Workflow stage	Main action
4	**Test deployment or configuration**	Prepare the minimum operational environment for validation with users.
5	**User testing and evidence collection**	Collect feedback on usability, language, performance, relevance, trust and decision-support value.
6	**Issue logging and corrective action**	Translate observations into technical, methodological or documentation tasks.
7	**Synthesis for final platform**	Identify reusable assets, transferable methods, remaining limitations and sustainability conditions.

This workflow ensures that Open Call projects contribute to the Beta platform as structured validation and extension cases. It also supports the transition from dispersed prototype activity to a consolidated platform roadmap. The most important feedback themes for the final release are linguistic robustness, performance under rural connectivity conditions, improved map editing, stronger local data integration, clearer user-facing communication and better metadata-driven control of tool and dataset selection.

9. Cross-Project Cooperation and Reuse of Tools

This chapter documents how PoliRuralPlus tools, data services and AI/GeoAI components are reused, tested and further developed in cooperation with related European and international projects. The purpose is to demonstrate that the PoliRuralPlus Beta platform is not a closed, project-specific system, but a reusable technical and thematic ecosystem supporting rural-urban development, climate adaptation, biodiversity monitoring, soil-health innovation, FAIR data governance and international capacity building.

The cooperation described in this chapter is based on a practical distinction between generic platform assets and project-specific configurations. Generic assets include Hub4Everybody, JackDaw, Raven, SensLog, MapWhiteboard, spatial data management services, metadata catalogues, AI-assisted discovery functions, geospatial analysis services and MCP-based tool integration. These components can be reused across domains and regions. Project-specific configurations include local datasets, thematic models, regional action plans, local vocabularies, pilot-specific indicators, domain-specific workflows and user-interface adaptations.

This distinction is important for the long-term exploitation of the Beta platform. The value of PoliRuralPlus is not limited to the nine core pilots or to the immediate project reporting period. The platform provides reusable digital building blocks that can be configured for different domains, including climate services, biodiversity monitoring, soil-health management, agricultural advisory services, rural tourism, regional innovation, international development cooperation and evidence-based public administration.

Cross-project cooperation also supports the technical validation of the platform. Each related initiative tests a different part of the PoliRuralPlus architecture. FOCAL strengthens the climate and Earth observation dimension. BioClima extends the platform towards biodiversity, climate and sensor-based ecological monitoring. INTERESH provides a soil-health and AKIS-oriented use case for participatory innovation and advisory workflows. FAIR2Adapt contributes advanced FAIR data, provenance and semantic interoperability practices. KijaniSpace validates the transferability of the platform beyond the European institutional and infrastructure context.

Together, these projects show that the Beta platform can operate as a shared technological ecosystem. They also provide feedback on requirements that are relevant for the final platform release, including metadata quality, semantic enrichment, edge or local deployment, multilingual interaction, local sensor integration, interoperability with data catalogues, user-centred tool orchestration and trustworthy AI-supported evidence generation.

9.1 Reuse and transfer between PoliRuralPlus and FOCAL

FOCAL provides an important test case for cooperation between the PoliRuralPlus platform and advanced climate and Earth observation workflows. The project focuses on making climate data and climate services more accessible at local level through workflow management, high-performance computing infrastructure and AI-supported analysis. Its thematic focus on urban planning and forestry is complementary to the rural-urban foresight and territorial planning orientation of PoliRuralPlus.

The cooperation between PoliRuralPlus and FOCAL is based on mutual transfer of geospatial modelling concepts, climate data access patterns and AI-ready data management practices. FOCAL contributes advanced processing methods for Earth observation and climate data. These include the reconstruction of cloud-affected Sentinel-2 optical imagery using complementary Sentinel-1 radar data, and super-resolution approaches for land surface temperature using combinations of meteorological model outputs and Landsat observations. These methods are relevant for producing more complete, spatially detailed and temporally consistent agroclimatic and environmental indicators.

From the PoliRuralPlus perspective, the main value of this cooperation lies in the operational use of these advanced climate and Earth observation products through user-facing platform components. The role of PoliRuralPlus is not to duplicate specialised climate-processing infrastructures. Rather, the platform provides mechanisms for discovery, interpretation and reuse of such outputs through metadata catalogues, geospatial services, GeoAI interaction and decision-support workflows.

FOCAL's use of STAC and OGC API – Records is particularly relevant for the D4.2 Beta platform. These standards support the catalogue-based discovery of spatio-temporal datasets and allow the transformation of climate and Earth observation resources into AI-ready, machine-actionable assets. This enables JackDaw and related GeoAI services to identify relevant datasets, select suitable analytical services and provide users with territorially specific answers based on documented data sources.

In practical terms, this means that a forestry manager, urban planner or regional authority can select an area of interest and ask a natural-language question about climate risks, vegetation status, drought exposure or thermal conditions. The platform can then use metadata, catalogue records and available analytical services to locate the relevant FOCAL data resources, execute or call the appropriate geospatial processing function, and return a contextualised answer. This workflow extends the value of static geospatial portals by connecting data discovery, analysis and explanation in a single interaction chain.

The cooperation with FOCAL therefore contributes to D4.2 in four main ways. First, it provides evidence that the PoliRuralPlus platform can consume advanced external climate and Earth observation products. Second, it supports the development of reusable Data Cube and catalogue-access patterns. Third, it strengthens GeoAI-based interaction with climate and environmental data. Fourth, it demonstrates how specialised research outputs can be translated into operational planning support for local and regional users.

9.2 BioClima and biodiversity/climate-oriented extensions

BioClima extends the PoliRuralPlus ecosystem towards biodiversity and climate-oriented monitoring. Its focus on integrated biodiversity and climate observation is directly relevant for nature-positive territorial planning, ecosystem resilience assessment and environmental decision support. The cooperation is based on the integration of biodiversity variables, climate variables, land-use information, sensor observations and AI-supported spatial analysis.

The main thematic contribution of BioClima is the extension of the platform towards Essential Biodiversity Variables and Essential Climate Variables. These variables provide a structured way to describe ecosystem condition, biodiversity pressures, habitat changes and climate-related environmental dynamics. Their integration into the PoliRuralPlus platform supports a broader interpretation of rural and peri-urban development, where attractiveness, investment, land use and infrastructure planning need to be assessed together with ecological sensitivity and climate resilience.

Hub4Everybody provides a flexible environment for the publication, sharing and contextualisation of BioClima-related data and outputs. It can support the dissemination of data resources, methodological descriptions, training materials and stakeholder-facing information. In combination with spatial data services and metadata catalogues, the platform enables biodiversity and climate-related resources to be made discoverable and reusable for territorial analysis.

Open Land Use and other land-cover data sources are relevant in this cooperation because they provide the spatial basis for assessing changes in landscape structure and land-use patterns. Comparisons between Open Land Use, Corine Land Cover and other reference layers can support the identification of landscape change, habitat fragmentation, ecological pressure and potential areas for restoration or nature-based interventions.

SensLog adds an important in-situ observation dimension. It supports the management of sensor data from static and mobile devices and can publish observations through standardised web services. This is relevant for monitoring microclimatic conditions, soil-related parameters and local environmental pressures. In the BioClima context, such data can help to complement satellite observations and land-use layers with local measurements, thereby improving the interpretation of ecological stability and habitat conditions.

JackDaw can act as a semantic and analytical orchestrator in this configuration. By combining spatial layers of protected areas, land-use analyses, biodiversity indicators, climate variables and sensor observations, the system can support questions related to ecological corridors, environmental pressures and the potential localisation of nature-based measures. Users can interact with the map interface and ask questions about the possible impact of planned infrastructure, land-use change or local development interventions on surrounding ecosystems.

For D4.2, BioClima provides evidence that the PoliRuralPlus Beta platform can be extended beyond rural attractiveness and foresight into biodiversity and climate-sensitive planning. It also confirms that the same platform components can support environmental monitoring, ecological interpretation and stakeholder communication where data sources are heterogeneous and include both Earth observation and in-situ measurements.

9.3 INTERESH and rural innovation knowledge exchange

INTERESH provides a complementary use case for the reuse of PoliRuralPlus tools in soil-health innovation, territorial cohesion and agricultural advisory workflows. The project focuses on strengthening soil-health management through regional Living Labs, participatory design and practical agronomic measures. Its orientation towards soil, water, fertility, advisory support and regional action planning creates a direct link with the multi-actor and decision-support functions of PoliRuralPlus.

The cooperation with INTERESH builds on participatory planning methods developed in PoliRural and PoliRuralPlus. The Multi-Actor Approach is particularly important in this context. Soil-health innovation cannot be implemented only as a technical process; it requires coordination among farmers, advisors, researchers, public authorities, regional development actors and local communities. The MAA tool supports this coordination by structuring stakeholder profiles, objectives, roles, interactions and progress records.

From a technical perspective, INTERESH can reuse Hub4Everybody, SensLog, Info4Agro-related services and JackDaw-based knowledge retrieval. This creates a practical infrastructure for collecting, visualising and interpreting data on soil properties, weather, vegetation status, agronomic practices and regional planning needs.

The platform can support both technical users, such as agronomists and advisors, and non-technical stakeholders who need accessible explanations of soil-health measures and regulatory requirements.

A key opportunity is the integration of GeoAI-supported advisory workflows with LPIS data, satellite indicators and agricultural regulations. Through this approach, agricultural advisors and AKIS actors can ask natural-language questions related to specific parcels, soil conditions, fertilisation, irrigation or compliance obligations. The platform can then retrieve relevant local data, regulatory information and analytical outputs and present them as operational advice.

This cooperation demonstrates how PoliRuralPlus can move from general rural-urban planning support towards more specialised agricultural advisory services. It also supports the development of workflows where regional action plans are informed by locally grounded evidence, participatory stakeholder processes and AI-assisted interpretation of complex technical information.

For D4.2, INTERESH provides methodological and technical evidence for the reuse of PoliRuralPlus components in Living Lab environments. It also strengthens the link between platform development, AKIS, soil-health policy, sustainable farming practices and regional innovation governance.

9.4 FAIR2Adapt and FAIR data/metadata alignment

FAIR2Adapt is directly relevant to the metadata, provenance and trustworthiness dimensions of the PoliRuralPlus Beta platform. The project focuses on transforming climate and socio-economic data into actionable knowledge for climate adaptation strategies. Its strong alignment with EOSC and FAIR Digital Object principles makes it an important cooperation case for improving the machine-actionability, traceability and reuse of data and knowledge resources.

The cooperation contributes to PoliRuralPlus by strengthening the platform's approach to FAIR data governance. This includes structured metadata, persistent identifiers, provenance documentation, semantic descriptions of variables, human validation of extracted claims, and transparent publication of evidence. These elements are essential for reducing the risk of unsupported AI-generated outputs and for ensuring that AI-assisted responses can be traced back to reliable sources.

A relevant pattern from FAIR2Adapt is the use of RO-Crate packaging with JSON-LD metadata. Such packages can describe datasets, models, workflows, documentation and related digital objects in a structured and interoperable way. For the PoliRuralPlus platform, this approach is relevant when preparing data and analytical outputs for reuse in EOSC-oriented infrastructures or when documenting the evidence chain behind AI-assisted results.

The I-ADOPT framework is relevant for semantic descriptions of observed variables. It helps define variables in a way that is clearer, more consistent and more machine-readable. This is important for cross-project cooperation because terms such as soil moisture, vegetation index, land surface temperature, flood exposure, biodiversity pressure or climate risk may be used differently across projects unless supported by structured semantic descriptions.

The FAIR2Adapt approach to claim extraction and human validation is also relevant for JackDaw and other AI-assisted components. Scientific or policy-relevant claims extracted from documents should not be treated as reliable automatically. They require provenance, expert review and, where appropriate, publication as independently citable and verifiable knowledge objects. This supports a more robust evidence model for AI-assisted decision support.

Natural-language search tools such as Pythia provide another relevant cooperation pattern. They show how users can query semantically enriched research objects and climate adaptation knowledge without requiring expert knowledge of formal metadata structures. This is consistent with the direction of JackDaw, where the platform aims to make complex spatial, policy and scientific resources accessible through natural language while preserving evidence traceability.

For D4.2, FAIR2Adapt contributes a methodological reference for machine-actionable metadata, FAIR assessment, provenance, semantic enrichment and trustworthy AI grounding. It strengthens the argument that PoliRuralPlus should not only provide access to data, but also document the evidence chain behind data, models, outputs and recommendations.

9.5 KijaniSpace and Africa-oriented deployment perspectives

KijaniSpace provides an important test case for the international transferability of PoliRuralPlus platform components. The project focuses on climate-smart agriculture and aquaculture in Africa, especially in the Lake Victoria Basin across Kenya, Tanzania and Uganda. Its Space-IoT Solution Box integrates Copernicus data with local IoT technologies to support smallholder farming, fisheries and local innovation ecosystems.

The relevance of KijaniSpace for PoliRuralPlus lies in the validation of platform components outside the European policy and infrastructure environment. Many assumptions valid in European projects cannot be transferred automatically to African contexts. Internet connectivity may be limited, data availability may be uneven, institutional arrangements may differ, and end users may require offline, mobile-first and multilingual access. These conditions create a useful stress test for platform robustness and adaptability.

Hub4Everybody can support KijaniSpace through hybrid deployment configurations, where local instances or adapted portals can be used for training, communication, data publication and community support. Mobile GIS clients and offline-capable workflows are relevant for field data collection, especially where connectivity is intermittent. This deployment logic supports the wider exploitation of PoliRuralPlus components in regions where centralised cloud-based access is not always sufficient.

Sensor integration is another important area of cooperation. KijaniBox devices can provide local measurements such as soil moisture, water temperature, water quality or other agro-environmental parameters. These observations can be managed through open IoT platforms and connected with SensLog-type services. This makes it possible to combine Earth observation, local sensor data and community-collected information in operational advisory workflows.

JackDaw can be adapted in KijaniSpace as a multilingual and simplified advisory interface. Such adaptation requires more than translation. The assistant must respect local crops, farming practices, aquaculture conditions, institutional constraints, advisory capacity and user literacy levels. Support for languages such as Swahili and simplified interaction patterns is therefore relevant for inclusive access to spatial and environmental intelligence.

Capacity building is also central. The use of e-learning, mapping modules, training materials and innovation programmes can help local partners, young innovators and start-ups create minimum viable products based on space data, IoT data and local needs. This confirms that the platform can support not only technical analysis but also knowledge transfer and innovation ecosystem development.

For D4.2, KijaniSpace provides evidence that PoliRuralPlus components can be transferred beyond the EU context if they are adapted to local infrastructure, language, governance and capacity-building conditions. It also supports the sustainability argument by showing that the platform can be reused in international cooperation and development-oriented innovation.

9.6 Lessons for sustainability and exploitation beyond PoliRuralPlus

The cross-project cooperation described in this chapter confirms that the long-term value of the PoliRuralPlus platform depends on its ability to operate as an open and replicable ecosystem. Sustainability cannot be based only on maintaining one project instance. It requires reusable components, clear documentation, modular deployment, standardised interfaces, transparent licensing, metadata quality, and a separation between generic software assets and project-specific configurations.

The main generic assets include Hub4Everybody, spatial data management components, metadata catalogues, HSLayers-NG-based map clients, SensLog, MapWhiteboard, JackDaw, Raven and related MCP-based service integration mechanisms. These assets can be reused across different domains because they are not tied to one specific pilot. They provide functions for content management, e-learning, geospatial publication, metadata management, sensor integration, collaborative mapping, natural-language interaction and AI-assisted spatial analysis.

Project-specific configurations include local datasets, regional action plans, thematic indicators, local models, pilot vocabularies, domain-specific ontologies, regional data licences, user-interface customisations and locally defined workflows. These configurations are necessary for practical deployment but should remain separable from the generic platform core. This separation makes the platform easier to maintain and replicate.

Cross-project cooperation also demonstrates that exploitation should not be limited to one commercial pathway. Several complementary pathways are relevant. These include open-source reuse, research infrastructure integration, public-sector deployment, advisory-service integration, DIH-based training and support, specialised service provision, international development cooperation, and domain-specific extensions for agriculture, biodiversity, forestry, climate adaptation and regional planning.

The Open Call and FSTP projects provide additional implementation evidence, but their detailed validation results are documented separately in Chapter 8. In Chapter 9, their relevance is primarily strategic: they confirm that the same platform assets can be localised by external actors and applied to different territorial challenges. This supports the broader exploitation logic of PoliRuralPlus as a reusable platform ecosystem.

A key lesson is that the platform must remain technically modular but organisationally supported. Reusable software components have limited value without onboarding material, metadata guidance, examples, deployment instructions, testing templates, user training and maintenance arrangements. D4.2 therefore needs to treat sustainability as a combination of software architecture, documentation, community uptake and cross-project reuse.

Another lesson concerns data sovereignty. The use of MCP-based service integration, local deployment options and controlled metadata-driven tool selection supports scenarios where sensitive data must remain inside institutional environments. This is relevant for public administrations, paying agencies, regional authorities and research infrastructures that cannot expose all data to external cloud services.

In summary, the sustainability of the PoliRuralPlus platform depends on maintaining a stable open core, supporting modular extensions, documenting reusable workflows, and continuing cross-project cooperation beyond the Beta release. The evidence from FOCAL, BioClima, INTERESH, FAIR2Adapt and KijaniSpace confirms that the platform has the potential to remain relevant beyond the original project boundary.

9.7 Cross-project Data Cube, GeoRAG and semantic knowledge cooperation

Modern rural planning, agricultural advisory services and climate adaptation require a transition from static map visualisation to dynamic, context-aware decision-support systems. Cross-project cooperation around PoliRuralPlus contributes to this transition through the combination of three technical patterns: Data Cubes, GeoRAG workflows and semantic knowledge resources. Together, these patterns create a Geospatial Knowledge and Service Layer that allows language models to operate as controlled orchestrators of verifiable spatial analysis rather than as isolated text generators.

Data Cubes provide efficient multidimensional access to large time series of satellite observations, climate variables and derived environmental indicators. They are relevant for questions involving change over time, anomaly detection, spatial comparison, vegetation condition, land surface temperature, drought indicators, land-use dynamics or climate exposure. FOCAL contributes to this pattern through advanced Earth observation processing, including cloud-gap mitigation and super-resolution methods. KijaniSpace adds requirements for the use of such data under conditions of limited infrastructure and high relevance for agriculture and aquaculture.

GeoRAG connects natural-language interaction with spatial discovery and analysis. In a typical workflow, a user defines or selects an area of interest and asks a territorial question. The platform then uses metadata, spatial context and tool descriptors to retrieve relevant datasets, identify applicable analytical services, execute or call required spatial operations and provide an answer grounded in data and documented sources. This is different from general textual retrieval because the response must be valid for a specific place, scale and time period.

Semantic knowledge resources provide the conceptual and evidential context needed to interpret spatial results. These resources may include policy documents, scientific publications, agronomic manuals, climate adaptation models, biodiversity descriptions, regulatory constraints and validated claims. FAIR2Adapt contributes advanced approaches to provenance, research objects, semantic variable descriptions and validated nanopublications. BioClima contributes to biodiversity and climate-related semantic relationships. INTERESH contributes soil-health and advisory knowledge linked to agricultural practice and regulation.

The Model Context Protocol is important for controlling this orchestration. Each analytical service can be exposed as a separate tool with a clear description of its purpose, input requirements, output structure, domain applicability, spatial validity and limitations. Raven can support the registration and description of these datasets and tools.

JackDaw can then pre-filter the available tools according to user intent, location, domain and data availability. This reduces unnecessary tool exposure and helps prevent inappropriate or irrelevant service calls.

This approach also helps address the problem of context overload in LLM-based systems. Instead of sending large volumes of documents, datasets or service descriptions to the language model, the system selects only the relevant resources and tools. The LLM is then used to interpret, coordinate and explain results, while authoritative calculations and data access remain within specialised services.

A typical future workflow could involve a user asking whether a specific parcel is suitable for a crop or restoration measure. The platform would first identify the spatial context and relevant parcel geometry. It could then retrieve time-series vegetation indicators and temperature anomalies from Data Cubes, local soil or weather observations from SensLog, land-use and administrative constraints from geospatial services, and applicable regulatory or advisory information from semantic knowledge resources. The final answer would be generated only after these sources are retrieved and interpreted in relation to the selected territory.

For D4.2, this cross-project cooperation provides an architectural and empirical basis for a reusable AI-ready platform pattern. It demonstrates how Data Cubes, GeoRAG, semantic metadata, sensor data, MCP services and JackDaw-based interaction can be combined into a coherent service layer for evidence-based spatial intelligence.

9.8 Summary matrix of cross-project cooperation and contribution to D4.2

The following table (see Table 14 below) summarises the main cooperation links relevant for Chapter 9. It identifies the associated PoliRuralPlus tools or services, the type of cooperation and the expected contribution to the D4.2 Beta platform documentation.

Table 14: Matrix showing cross-project cooperation leading up to D4.2

Project / initiative	Relevant PoliRuralPlus tool or service	Type of cooperation	Expected contribution to D4.2
FOCAL	Hub4Everybody, metadata services, STAC/OGC discovery, Data Cubes, JackDaw and GeoRAG workflows	Reuse and integration of climate, Earth observation and geospatial processing concepts	Evidence of technological interoperability, catalogue-based access to climate and EO resources, and transfer of advanced EO analysis into user-facing decision support

Project / initiative	Relevant PoliRuralPlus tool or service	Type of cooperation	Expected contribution to D4.2
BioClima	Open Land Use, land-cover analysis, SensLog, JackDaw, geospatial services and environmental metadata	Extension towards biodiversity and climate-oriented monitoring, including EBV/ECV-related use cases	Evidence for environmental and nature-positive territorial planning scenarios, combining EO data, land-use layers and in-situ sensor measurements
INTERESH	Hub4Everybody, MAA tool, SensLog, Info4Agro-related services, JackDaw knowledge retrieval and advisory workflows	Knowledge exchange, stakeholder coordination, soil-health innovation and AKIS-oriented advisory support	Methodological and technical evidence for Living Lab operation, participatory planning, soil-health workflows and parcel-level advisory services
FAIR2Adapt	Metadata services, provenance documentation, FAIR data practices, RO-Crate concepts, semantic variable descriptions and validated knowledge objects	Alignment with FAIR data, semantic interoperability, provenance and trustworthy AI grounding	Evidence for machine-actionable and auditable data governance, improved metadata quality and more reliable AI-supported evidence synthesis
KijaniSpace	Hub4Everybody, JackDaw, SensLog-compatible sensor integration, mobile GIS clients, training modules and local deployment patterns	International deployment, capacity building and climate-smart agriculture/aquaculture in Africa	Evidence of transferability, robustness and scalability beyond the EU context, including offline, multilingual and infrastructure-constrained deployment conditions
Cross-project Data Cube / GeoRAG / semantic cooperation	Data Cubes, GeoRAG workflows, semantic metadata, MCP servers, Raven tool descriptors and JackDaw orchestration	Shared technical design pattern developed and validated across cooperating projects	Architectural evidence for a reusable, AI-ready geospatial knowledge and service layer supporting controlled, provenance-aware spatial intelligence

Overall, Chapter 9 demonstrates that the PoliRuralPlus Beta platform is already functioning as a reusable ecosystem of tools, services and methods. The cooperation with FOCAL, BioClima, INTERESH, FAIR2Adapt and KijaniSpace confirms that its components can be transferred across domains, adapted to different territorial contexts and connected to external infrastructures. This is a central finding for D4.2, because it positions the Beta release not only as an internal project milestone, but also as a foundation for longer-term exploitation and cross-project innovation.

10. Risk Management, Ethics, Security and Sustainability

The Beta platform combines AI-supported services, geospatial data, stakeholder information, external tools and FSTP extensions. Risk management (see Table 15 below) therefore covers technical reliability, data protection, responsible AI, cybersecurity, operational sustainability and exploitation readiness.

Table 15: Risk assessment matrix for the most significant risks

Risk	Likelihood (1-3)	Impact (1-3)	Risk Score	Mitigation Strategy and Description
Inconsistent Metadata Quality (Leading to poor search/tool selection)	High (3)	High (3)	9	Implement strict metadata guidance for all contributors. Prioritize validation checks during the Raven ingestion process, focusing on mandatory fields like temporal and spatial coverage.
Interoperability Failure between Components (Lack of seamless interaction across modular tools)	Medium (2)	High (3)	6	Enforce adherence to the Model Context Protocol (MCP) for all service integrations. Develop comprehensive integration test suites that cover cross-module workflows end-to-end.
Data Governance and Provenance Gaps (Inability to trace source/limitations of outputs)	Medium (2)	High (3)	6	Mandate the logging of four levels of provenance (Source, Ingestion, Processing, Usage) for all derived assets. Update user interfaces to prominently display evidence trails alongside any output.
Poor User Adoption by Non-Technical Stakeholders (Complexity overwhelms end-users in pilot settings)	High (3)	Medium (2)	6	Focus on creating high-level, pre-defined "reference workflows" and standardized user guides. Develop role-specific training material emphasizing outcome over functionality.
Over-reliance on Proprietary or Unstable AI Models (Platform function degrades if external APIs change/fail)	Medium (2)	High (3)	6	Maintain the hybrid AI strategy: integrate and test both commercial cloud APIs and self-hosted, open-weight models (e.g., Ollama). Implement robust fallback mechanisms for critical functions.

Risk	Likelihood (1-3)	Impact (1-3)	Risk Score	Mitigation Strategy and Description
Scope Creep during Integration (Attempting to build a monolithic solution instead of modular one)	Medium (2)	Medium (2)	4	Strictly adhere to the "modular building block" concept documented in D4.1, allowing explicit technical and operational separation between core services when integrating new Open Call features.

The risk score is determined by multiplying the assigned Likelihood (1=Low, 2=Medium, 3=High) by the Impact (1=Low, 2=Medium, 3=High). Higher scores indicate higher priority risks that require immediate mitigation focus.

10.1 Technical and operational risks

The main technical and operational risks for Beta concern the stability, interoperability and practical usability of a modular service ecosystem. Inconsistent metadata quality may reduce the effectiveness of search, discovery and tool selection, particularly where datasets lack reliable temporal, spatial or thematic descriptions. Interoperability failure between components is also a material risk, as Beta depends on seamless interaction between independently developed services, tools and workflows. These risks will be mitigated through strict metadata guidance, validation checks during the Raven ingestion process, adherence to the Model Context Protocol (MCP), and integration testing across representative end-to-end workflows. Scope creep will be controlled by maintaining the modular building-block approach, ensuring that new services and Open Call components can be integrated without forcing the platform into a monolithic architecture.

10.2 Data protection, responsible AI and security considerations

Beta addresses risks related to data governance, provenance, privacy, access control and responsible use of AI-generated outputs. A key concern is that users may be unable to trace the origin, processing history or limitations of data and model-derived results, which could undermine trust and accountability. This will be mitigated by enforcing provenance logging at four levels: source, ingestion, processing and usage. User interfaces should expose these evidence trails clearly, allowing users to inspect the basis for outputs and understand relevant uncertainties. Responsible AI safeguards will include transparency on model limitations, appropriate human oversight, clear distinction between evidence and interpretation, and fallback mechanisms where proprietary or unstable AI services become unavailable or unsuitable for a given task.

10.3 Sustainability and exploitation of Beta services

The sustainability of Beta services depends on whether the platform can remain maintainable, useful and affordable beyond the immediate project period. Risks include excessive dependence on unstable proprietary AI APIs, insufficient uptake by non-technical users, and the creation of services that are technically impressive but difficult to exploit in real operational settings. The project will therefore maintain a hybrid AI strategy, combining commercial APIs with self-hosted open-weight models where appropriate, and will prioritise high-value reference workflows that demonstrate clear benefits for target users. Exploitation should focus on reusable services, role-specific workflows, documented interfaces and operational guidance that allow partners, advisors and other stakeholders to adopt or extend Beta components after the project.

11. Roadmap from Beta to Final Platform

This chapter defines the development path from the current Beta release towards the final PoliRuralPlus platform. The roadmap is based on the technical status of the Beta components, the experience from core pilot regions, the feedback collected through Open Call and FSTP activities, and the need to prepare the platform for more stable operational use. The final project phase must therefore be treated not only as a period for completing software development, but as a consolidation phase in which the platform is technically integrated, tested, documented, demonstrated and prepared for broader uptake.

The central objective for the final release is to move from a functioning Beta environment towards a coherent, reusable and validated platform ecosystem. The Beta version has already demonstrated that core components can support pilot interaction, AI-assisted analysis, metadata-driven data discovery, collaborative work and territorial decision-support tasks. The final release must now transform this capability into a more mature and better integrated operational environment.

A strong priority is the systematic integration of comments from pilots and Open Call/FSTP projects into the platform solution. These comments should not remain only as evaluation observations in Chapters 6, 7 and 8. They must be translated into development backlog items, corrective actions, interface improvements, training needs and validation scenarios. Pilot feedback should mainly guide the improvement of core territorial decision-support workflows, while Open Call feedback should guide robustness, localisation, usability and extensibility of platform components.

The roadmap is organised around five main directions:

1. consolidation and integration of platform components;
2. continuous testing and validation with pilots, Open Call partners and technical users;
3. preparation of training material and support mechanisms;
4. demonstration of workflows using model builders and configurable decision-support scenarios;
5. progression of key tools towards higher Technology Readiness Levels.

11.1 Remaining development priorities

The remaining development priorities are focused on making the final platform more integrated, more reliable and easier to use by pilots, external developers, model builders and decision-support practitioners. The next stage must reduce fragmentation between components and provide clearer operational pathways for users moving from data ingestion to analysis, collaborative interpretation and decision-making.

A first priority is the stronger integration of the core components. The final platform should provide smoother interaction between Hub4Everybody, Raven, JackDaw, MapWhiteboard, MAATool, dashboards, spatial data services, analytical tools and training resources. Users should not experience these components as isolated applications. Instead, they should be able to move between data discovery, map interaction, stakeholder coordination, AI-supported interpretation and reporting through a more coherent workflow.

Raven should be further consolidated as the main data and metadata ingestion component. The final stage should improve its stability, metadata guidance, dataset lifecycle management and integration with analytical tools. Raven metadata should be used not only for catalogue publication, but also for tool selection, access control, provenance tracking, dataset suitability assessment and preparation of AI-assisted workflows.

JackDaw should be further developed as the GeoAI interaction and orchestration layer. The priority is not only to improve natural-language interaction, but also to ensure that JackDaw is consistently connected to reliable metadata, spatial services, validated analytical tools and controlled execution paths. Feedback from pilots and Open Call validators should be used to improve linguistic robustness, spatial query reliability, transparency of responses, handling of local place names, and reduction of unnecessary technical details in user-facing outputs.

A second priority is continuous component testing. Testing should not be organised only as a one-time final validation exercise. It should operate as a continuous cycle in which each component is tested after relevant changes, user feedback is recorded, issues are classified, and corrective actions are followed until closure. The testing process should cover functional behaviour, usability, performance, metadata completeness, interoperability, security, access control, AI-response quality and suitability for real decision-support contexts.

A third priority is preparation of training material. The final release must be accompanied by training resources for different user groups. General users need introductory materials explaining platform access, basic navigation, map interaction, search functions and interpretation of AI-supported outputs. Pilot managers need guidance on preparing use cases, uploading data, organising stakeholder workflows and reporting results. Technical users need documentation on APIs, metadata profiles, deployment options and integration of external services. Model builders need specific material explaining how to configure workflows, connect tools, define inputs and outputs, validate models and prepare reusable demonstrations.

A fourth priority is the preparation of demos with model builders. The platform should include a set of reproducible demonstration scenarios showing how configurable workflows can be created and used in real decision-support processes. These demonstrations should cover, for example, rural attractiveness analysis, stakeholder mapping, regional action plan support, local agricultural advisory workflows, environmental risk interpretation, accessibility analysis, and collaborative planning. The demos should show not only individual tools, but complete workflows from data ingestion to interpretation and decision discussion.

A fifth priority is the integration of collaborative tools into decision processes. MapWhiteboard, MAATool and other collaborative components should be positioned as practical instruments for structured decision-making, not only as supporting or visualisation tools. They should help users document stakeholders, define objectives, discuss scenarios, map spatial evidence, record comments, and connect technical analysis with policy or planning decisions. This requires better workflow integration between collaborative mapping, multi-actor coordination, AI-supported evidence synthesis and Regional Action Plan preparation.

A sixth priority is the progression of tools towards higher Technology Readiness Levels. The final phase should identify which components are ready to move from prototype or Beta status towards more stable operational maturity. This does not mean that every component must reach the same maturity level. However, each major tool should have a clearly documented status, known limitations, testing evidence, deployment requirements, user documentation and exploitation pathway. Components with strong pilot or Open Call validation should be prioritised for TRL advancement.

While the validation thus far has been case-by-case based and qualitative in nature, the remaining validation activities for the final release of the platform will follow the more comprehensive approach of systematic consultation with WP5, leading up to the release of D4.3 at the end of the project.

The following table (Table 16) summarises the main remaining priorities for the final platform phase.

Table 16: Remaining development priorities

Priority area	Required action before final release	Expected result
Integration of pilot and Open Call feedback	Convert feedback into development backlog, corrective actions, validation cases and training needs	Platform improvements are directly grounded in real user and validator experience
Component integration	Improve workflow links between Hub4Everybody, Raven, JackDaw, MAATool, MapWhiteboard, dashboards and analytical services	Users experience the platform as a coherent decision-support environment
Raven consolidation	Improve metadata guidance, lifecycle management, tool descriptors and data suitability checks	Data and metadata become operational inputs for discovery, tool selection and provenance
JackDaw consolidation	Improve spatial query handling, linguistic robustness, tool orchestration and clarity of user-facing answers	GeoAI interaction becomes more reliable, transparent and useful for non-technical users
Continuous testing	Implement iterative testing of functionality, usability, performance, interoperability and AI-output quality	Platform quality is improved progressively rather than only at the final review stage
Training materials	Prepare role-specific guidance for users, pilots, technical integrators and model builders	Adoption barriers are reduced and external users can work with the platform more independently
Model-builder support	Prepare demos, workflow templates and configuration examples	Model builders can create, adapt and validate reusable decision-support workflows

Priority area	Required action before final release	Expected result
Collaborative decision processes	Integrate MAATool, MapWhiteboard and related tools into planning, foresight and RAP workflows	Collaborative tools become part of the decision process rather than separate applications
TRL progression	Document maturity, limitations, deployment conditions and validation evidence for each component	Key platform tools move towards higher readiness and exploitation potential

11.2 Integration roadmap for the final release

The integration roadmap for the final release should be organised as a structured sequence of technical, validation and capacity-building actions. The final phase should not add complexity without consolidation. The primary objective is to connect existing components better, improve their reliability and demonstrate their practical use through tested workflows.

The first step is backlog consolidation. Comments from pilots, Open Call projects, technical partners and WP validation activities should be reviewed together and classified into clear categories: functional defects, usability issues, performance limitations, missing documentation, missing training material, integration gaps, metadata gaps, AI-response limitations, and requirements for new or improved workflows. This classification should create a manageable implementation backlog for the final platform phase.

The second step is component integration. This should focus on practical user pathways. For example, a dataset uploaded through Raven should be discoverable through catalogue search, usable by JackDaw, available in map-based interaction, and connected to relevant analytical tools where access conditions allow. Similarly, outputs from spatial analysis should be reusable in collaborative discussion, stakeholder workflows and decision-support documentation.

The third step is preparation of reference workflows. The final platform should include a limited number of well-documented workflows that demonstrate the integrated use of components. These workflows should be selected according to pilot relevance, Open Call validation results and exploitation potential. They should be sufficiently concrete to be demonstrated and sufficiently generic to be reused in other regions.

The fourth step is continuous testing. Each reference workflow should be tested from the perspective of at least three user categories: pilot or domain user, technical platform user and model builder. Testing should document whether the workflow can be executed, whether outputs are understandable, whether required data are available, whether metadata are sufficient, whether access rights are correctly handled, and whether the result can support decision-making.

The fifth step is training and support. Training material should be prepared in parallel with testing, because testing reveals where users need explanation, examples or simplified procedures. Training material should include short user guides, technical notes, video or slide-based demonstrations, model-builder instructions, metadata templates, testing checklists and examples of good practice. These materials should be maintained as living resources until the final release.

The sixth step is demonstration and validation with model builders. Model builders should be trained to configure workflows, connect tools, define inputs and outputs, and prepare controlled demonstrations. This is important for sustainability because platform uptake will depend not only on end users, but also on intermediaries able to configure the platform for new domains, new pilots and new institutional contexts.

The seventh step is TRL-oriented assessment. At the end of the final platform phase, each major component should be assessed in terms of maturity, stability, documentation, validation evidence, deployment readiness and exploitation potential. The objective is to demonstrate measurable progress from Beta functionality towards higher readiness.

The roadmap (Table 17) can be structured as follows.

Table 17: Phasing of remaining WP4 activities

Roadmap phase	Main activities	Main outputs
Phase 1: Feedback consolidation	Review comments from pilots and Open Call/FSTP projects; classify issues and requirements; define corrective actions	Prioritised final-stage development backlog
Phase 2: Technical integration	Improve links between data ingestion, metadata, AI interaction, spatial services, dashboards and collaborative tools	More coherent platform workflows and reduced component fragmentation
Phase 3: Reference workflow preparation	Select and document demonstrator workflows relevant for pilots, Open Calls and exploitation	Reusable workflow templates and demonstration scenarios
Phase 4: Continuous testing	Test components and workflows with users, technical partners and model builders; record issues and improvements	Iterative validation evidence and corrected platform functions
Phase 5: Training and documentation	Prepare user manuals, model-builder guidance, metadata templates, demo material and technical integration notes	Training package for final release and post-project uptake

Roadmap phase	Main activities	Main outputs
Phase 6: Model-builder demonstrations	Train model builders; prepare configured demos; validate workflow configuration procedures	Demonstrated capacity to configure and replicate platform workflows
Phase 7: TRL progression and final assessment	Assess component maturity, limitations, deployment readiness and exploitation potential	Final component status matrix and evidence of TRL advancement

This roadmap should be implemented as an iterative process. Integration, testing, training and demonstration should proceed together. When a pilot or Open Call comment identifies a weakness, the response should include not only technical correction, but also improvement of guidance, testing scenarios and, where appropriate, model-builder training material.

11.3 Recommendations for WP4/WP5/WP6/WP7 alignment

The final platform phase requires close coordination between WP4, WP5, WP6 and WP7. The roadmap cannot be delivered by technical development alone. It requires a structured link between implementation, pilot validation, Open Call integration, training, dissemination and exploitation.

To ensure that the transition from the Beta platform (D4.2) to the final platform release (D4.3) is evidence-based, the final six-month period will use a structured evidence collection approach (see Table 18 below). The purpose is to ensure that technical improvements, pilot experiences, Open Call/FSTP contributions and exploitation activities are captured in a traceable way and can demonstrate final platform maturity.

Table 18: Evidence indicators for progress from beta to final platform (D4.3) and major contributing WP.

Evidence category	Required evidence towards D4.3	Main contribution
Implementation maturity	Component status matrix, release notes, resolved issues and remaining limitations	WP4
Integration readiness	End-to-end workflow demonstrations, interoperability tests, deployment documentation	WP4
Pilot validation	Pilot use cases, user feedback, decision-support relevance, RAP contribution evidence	WP5
Open Call/FSTP contribution	Validation findings, localisation needs, lessons learned, improvements integrated into the platform	WP6
AI transparency and trustworthiness	Source-to-answer examples, provenance information, limitations and responsible AI considerations	WP4/WP5

Evidence category	Required evidence towards D4.3	Main contribution
Uptake readiness	Training material, user documentation, demonstration cases, exploitation pathways	WP7

WP4 should lead the technical consolidation of the platform. Its main responsibility is to improve component integration, maintain the technical backlog, implement corrective actions, prepare deployment and integration documentation, and coordinate the maturity assessment of platform components. WP4 should also ensure that the final architecture remains modular, standards-based and reusable beyond the project.

WP5 should provide structured pilot validation. Its role is to ensure that core pilot feedback is translated into clear requirements and testing evidence. WP5 should help define pilot-specific validation scenarios, organise user testing, document decision-support relevance, and confirm whether collaborative and AI-supported workflows are usable in real territorial planning contexts.

WP6 should coordinate the integration of Open Call and FSTP feedback. Its role is to ensure that Open Call projects are not treated only as external demonstrations, but as a source of concrete validation findings, localisation requirements, extension patterns and technical stress tests. WP6 should work with WP4 to identify which Open Call outputs should remain project-specific and which should be generalised as reusable platform components or workflows.

WP7 should prepare the sustainability, dissemination and exploitation pathway. Its role is to transform the final platform outputs into understandable value propositions for public authorities, rural innovation actors, advisory services, Digital Innovation Hubs, research projects and external developers. WP7 should also coordinate training dissemination, demonstration events and exploitation narratives based on the final TRL status of platform components.

The following coordination model is recommended.

Table 19: Coordination model for WP-alignment leading up to completion of project

Work package	Main responsibility in the final phase	Required contribution to Chapter 11 roadmap
WP4	Technical consolidation, integration, deployment, documentation and TRL assessment	Maintain backlog, improve component integration, prepare technical documentation and final component status

WP5	Pilot validation and decision-support relevance	Provide pilot feedback, validation scenarios, user testing results and evidence of decision-process integration
WP6	Open Call/FSTP integration and extension assessment	Provide Open Call feedback, extension patterns, localisation issues and recommendations for reusable components
WP7	Sustainability, dissemination, training uptake and exploitation	Prepare exploitation messages, training dissemination, demo events and pathways for post-project adoption

A specific coordination mechanism should be established for the final stage (see Table 19 above). It should include a shared issue register, a component maturity matrix, a training-material checklist, a demo-scenario list and a validation-evidence tracker. These instruments should be regularly updated and used to ensure that the final release is not only technically complete but also tested, documented and ready for use.

The integration of collaborative tools into decision processes should be treated as a cross-work-package priority. WP4 should ensure technical integration; WP5 should validate use in pilot decision contexts; WP6 should collect evidence from Open Call adaptations; and WP7 should package the results for dissemination and exploitation. This will help demonstrate that collaborative mapping, stakeholder management and AI-supported analysis can become part of operational planning and policy-support workflows.

The final release should therefore be evaluated against practical maturity criteria. The platform should be considered ready for final delivery when core components are integrated, main pilot and Open Call comments have been addressed or documented, training material is available, model-builder demos can be executed, collaborative tools are linked to decision processes, and component maturity has progressed towards higher TRL.

In conclusion, the roadmap from Beta to final platform is a transition from technical demonstration to operational readiness. The final phase must convert validation feedback into concrete improvements, strengthen integration between components, provide structured training and support, enable model builders to configure reusable workflows, embed collaborative tools in decision-making, and move the most mature tools towards higher Technology Readiness Levels. This will ensure that the PoliRuralPlus platform is not only delivered as a project output, but prepared as a sustainable, reusable and exploitable digital ecosystem.

12. Conclusions

D4.2 confirms that the PoliRuralPlus platform has progressed from the design baseline described in D4.1 towards an operational Beta environment. The platform is no longer described only as an intended architecture or a portfolio of planned services. It is now presented as an implemented and testable ecosystem combining data ingestion, metadata management, GeoAI interaction, spatial analysis, collaborative tools, stakeholder coordination, pilot workflows and external extensions.

The main achievement of the Beta stage is the establishment of a modular platform structure. The platform integrates former PoliRural assets with new PoliRuralPlus components and provides a basis for reuse across different territorial and thematic contexts. This modular architecture is appropriate for rural-urban innovation because regional authorities, pilot managers, advisors, public actors and local communities do not all require the same technical stack. The Beta platform therefore offers a configurable set of services that can be adapted to different decision-support needs.

A second conclusion is that metadata has become a core operational element of the platform. Metadata supports not only catalogue discovery, but also dataset interpretation, tool selection, access management, provenance tracking and AI-assisted workflows. Raven plays a central role in this process by supporting data ingestion, metadata preparation and dataset lifecycle management. The quality of metadata directly affects the reliability of downstream services, including JackDaw, dashboards, analytical tools and collaborative decision-support workflows.

A third conclusion concerns the role of JackDaw and GeoAI interaction. The Beta platform demonstrates that natural-language interfaces can support access to complex spatial and thematic resources, especially when they are connected to structured metadata, validated datasets and controlled analytical services. However, the validation process also shows that AI-based interaction must remain grounded in reliable data, clear provenance, appropriate tool selection and transparent communication of limitations. The final platform phase must therefore continue improving spatial query reliability, linguistic robustness, handling of local context and clarity of user-facing responses.

The core pilots confirm the relevance of the platform for territorial planning and Regional Action Plan support. The platform has been tested in different governance and thematic settings, including circular economy, participatory foresight, AKIS coordination, community planning, cross-border attractiveness analysis, cultural heritage development, KPI monitoring and agricultural digitisation. These pilot experiences show that the platform can support evidence-based discussion, stakeholder coordination and spatially informed decision-making.

The Open Call and FSTP activities provide an additional validation layer. They demonstrate that external actors can adapt, stress-test and extend the PoliRuralPlus ecosystem. Their feedback is particularly important because it identifies practical constraints that may not be visible in core platform development alone. These constraints include usability issues, performance limits, linguistic sensitivity, local data availability, integration needs and expectations from non-technical users. The final development phase must systematically integrate these findings into the platform backlog and corrective-action process.

The report also confirms the strategic importance of collaborative tools. MapWhiteboard, the Multi-Actor Approach Tool and related mechanisms should not be treated only as optional communication functions. They are necessary for connecting evidence, stakeholders, scenarios and decisions. In the final stage, these tools should be more directly embedded into decision workflows, including Regional Action Plan preparation, stakeholder consultation, participatory mapping, scenario discussion and validation of policy options.

Cross-project cooperation demonstrates that the platform has value beyond PoliRuralPlus. Cooperation with FOCAL, BioClima, INTERESH, FAIR2Adapt and KijaniSpace shows that the same technical assets can support climate services, biodiversity monitoring, soil-health innovation, FAIR data governance and international development cooperation. This confirms the potential of the platform as a reusable digital ecosystem rather than a project-specific demonstrator.

The Beta stage also identifies clear limitations. Component integration must be strengthened, user pathways must be simplified, training material must be completed, model-builder support must be improved and continuous testing must be institutionalised. The final platform release should therefore focus less on adding isolated new functions and more on consolidation, reliability, documentation, workflow integration and readiness for exploitation.

A further conclusion is that Technology Readiness Level progression must be treated explicitly. Some components are already sufficiently mature for broader testing and reuse, while others remain experimental or require additional validation. The final report should therefore document component maturity, deployment requirements, validation evidence, known limitations and exploitation potential. This will help distinguish between tools ready for operational uptake and tools requiring further research or development.

The main recommendation for the final stage is to treat pilot and Open Call feedback as direct input to platform improvement. Feedback should be converted into backlog items, test cases, interface corrections, metadata improvements, training resources and demonstration scenarios. This will ensure that the final platform is not only technically delivered, but also improved through real user experience.

The second recommendation is to prepare a coherent training and demonstration package. Users, pilot managers, technical integrators and model builders require different levels of documentation. Training material should therefore include user guides, metadata templates, technical integration notes, workflow examples, model-builder instructions and reproducible demos. This is essential for sustainability because platform reuse depends on the capacity of external actors to understand, configure and operate the tools.

The third recommendation is to maintain the modular and standards-based direction of the platform. The use of APIs, geospatial standards, metadata profiles, MCP-based service integration and reusable components creates a practical basis for sustainability. This direction should be reinforced in the final release, especially where the platform is expected to operate in different institutional, technical and territorial environments.

In conclusion, D4.2 demonstrates that the PoliRuralPlus Beta platform is a functional and reusable basis for spatially enabled rural-urban decision support. It provides implemented components, pilot validation, Open Call extension pathways, cross-project reuse evidence and a clear roadmap to the final release. The final stage must now focus on integration, testing, training, model-builder support, collaborative decision workflows and TRL advancement. If these priorities are addressed, the PoliRuralPlus platform can move from a Beta demonstrator towards a sustainable and exploitable digital ecosystem for rural-urban innovation, climate adaptation, biodiversity-sensitive planning and evidence-based territorial development.

Annex 1 Component status matrix

Component	Beta status	Main function	Integration point	Responsible partner	Next action
PoliRuralPlus Hub / H4E	To be completed	Central access, content, repository and communication services	SSO, CMS, document/data services	To be completed	Document current deployment and access pathway
Jackdaw	To be completed	GeoAI chat and spatial decision support	API layer, metadata services, analytical tools	To be completed	Add implemented workflows and screenshots
Magpie	To be completed	AI-enabled catalogue and metadata search	OGC/metadata services, external catalogues	To be completed	Document supported catalogues and test queries
Vulture	To be completed	AI productivity sandbox	Document and information stream processing	To be completed	Define Beta use cases and limitations
MapWhiteboard	To be completed	Collaborative mapping and spatial annotation	Hub and pilot workflows	To be completed	Document pilot testing scenario

Annex 2 Beta testing template

Test item	Description
Tester / pilot region	To be completed
Component tested	To be completed
Test scenario	To be completed
Input data used	To be completed
Expected result	To be completed
Observed result	To be completed
Issue severity	Low / medium / high
Corrective action	To be completed
Responsible partner	To be completed

Annex 3 FSTP cooperation and cross-project reuse template

This annex should be completed during the Beta validation phase. It collects structured evidence from FSTP/Open Call partners and from related projects using or extending PoliRuralPlus platform components.

For each case, the report should record: cooperating partner or project, user need, reused PoliRuralPlus component, required data or metadata, integration mechanism, support provided by WP4, testing evidence, issue identified and planned follow-up.

Specific attention should be given to reusable Data Cube access patterns, GeoRAG retrieval workflows, RAG-Knowledge Graph integration, FAIR metadata, semantic vocabularies and evidence that the same component can be transferred across several project contexts.

Annex 4 References

- Horizon Europe, EU Grants, AGA - Annotated Grant Agreement, EU Funding Programmes 2021-2027.
- PoliRuralPlus Grant Agreement, GA No 101136910.
- PoliRuralPlus, D2.1 Stakeholders Identification
- PoliRuralPlus, D2.3 Rural-urban data analysis, Annex B: Cognitive Walkthrough, SUS-based usability assessment and service testing results.
- PoliRuralPlus, D3.1 RAP Methodology
- PoliRuralPlus, D4.1 PoliRuralPlus platform design
- PoliRuralPlus, D5.1 Prepared pilot by Stakeholder Multi-Actor Approach
- PoliRuralPlus, D5.2 Regional Action Plans draft
- PoliRuralPlus, D6.2 Open Call Package
- PoliRuralPlus, D7.1 Comprehensive handbook, guidelines and materials for communication and dissemination
- PoliRuralPlus, D7.2 Practice Abstracts – batch 1
- PoliRuralPlus, D7.3 Policy Brief No. 1
- PoliRuralPlus, internal M18 Technical/Periodic Report, Reporting Period 1
- PoliRuralPlus, Communication – implementation and monitoring dataset
- PoliRuralPlus, Dissemination – implementation and monitoring dataset
- PoliRuralPlus website, Rural Pilot webpages
- PoliRuralPlus website, Open Calls and FSTP project pages
- PoliRuralPlus website, Services / tool pages: MAATool, MapWhiteboard, JackDaw, Advisor/RAG, Hub4Everybody
- PoliRuralPlus website, Knowledge Base and webinar materials
- Related project materials: FOCAL, BioClima, INTERESH, FAIR2Adapt and KijaniSpace cooperation references