UDENE Urban Development Explorations using Natural Experiments

D3.4: Operational Exploration Tool and Matchmaking Tool

The project has received funding from the European Commission's Horizon Europe Research and Innovation Programme under grant agreement No 101131190 with the EUSPA

Funded by the European Union

Deliverable Information			
Work Package	WP3		
Lead partner	Prof. Dr. Murat Özbayoğlu (TOBB)		
Author(s)	Hüseyin Pekkan (TOBB) Oğuz Deniz (TOBB) Serena Sauissi (TUNSA) Khalil Ben Said (TUNSA) Mohamed Rajhi (TUNSA) Mustafa Ali Türker (WEglobal)		
Due date	31 December 2024		
Version number	1.0 (Alpha)	Status	Draft
Dissemination level	Public		

Funded by the European Union

Document Information	
Project Number	101131190
Project Title	Urban Development Explorations using Natural Experiments
Acronym	UDENE
Start Date	1 January 2024
Duration	24 months
Call identifier	HORIZON-EUSPA-2022-SPACE
Topic	HORIZON-EUSPA-2022-SPACE-02-56 Strategic autonomy in developing, deploying and using global space-based infrastructures, services, applications and data 2022
Instrument	HORIZON-RIA
Project URL	www.udene.eu
EU Project Officer	Valeria CATALANO
Disclaimer	This document has been prepared within the scope of UDENE Project which received funding from the European Union's Horizon 2023 Research and Innovation Programme under grant agreement No 101131190. All information in this document is provided "as is" and no guarantee or warranty is given that the information is fit for any particular purpose. All the information in this document reflects only the Authors' view. Neither WG nor the EC is not responsible for any use that may be made of the information contained therein.

Funded by the European Union

Revision History			
Revision	**Date**	**Contributor(s)**	**Description**
0.1	1.12.2024	Mustafa Ali Türker	Table of Contents and the template
0.2	15.12.2024	Hüseyin Pekkan	Infrastructure Implementation
0.3	19.01.2025	Khalil Ben Said	Frontend
0.4	23.02.2025	Hüseyin Pekkan	Demonstration for urban experiments and API Updates
0.5	26.02.2025	Mohamed Rajhi	Matchmaking tool
1.0	28.02.2025	Mustafa Ali Türker	Final Edits
1.1	25.03.2025	Hüseyin Pekkan, Mustafa Ali Türker	Review Updates

Quality Control			
Role	**Date**	**Contributor(s)**	**Approved/ Comment**
Reviewer	1/20/2025	*BioSense institute (Nikola Obrenović, Maksim Lalić, Polina Maglevannaia, Marija Kopanja)*	Quality related comments provided
Reviewer	1/21/2025	*BioSense institute (Nikola Obrenović)*	Approved

Funded by the European Union

TABLE OF CONTENTS

Funded by the European Union

Funded by the European Union

1. Overview

1.1 Overall Objectives

The D3.4 deliverable is a key milestone in the UDENE (Urban Development Explorations using Natural Experiments) initiative. It highlights the development and release of tools that leverage GIS data cubes, AI-powered analysis, and interactive front-end components to support operational exploration and matchmaking activities across diverse domains.

1.2 Work Package Description

This work package focuses on defining urban development use cases by detailing the outcome variables, input parameters, and necessary datasets—thereby setting the foundation for the subsequent data provisioning in WP2. It will develop advanced algorithms that identify urban regions with similar characteristics based on comprehensive Copernicus data analysis and transparently present the causal effect estimations via an integrated Exploration Tool. Moreover, the package includes creating a Matchmaking Tool to connect users with a curated inventory of European space-sector providers, supported by secure API-driven user management, GIT-based version control, and deployment on the WEkEO DIAS (Data and Information Access Services).

1.3 Task Descriptions

Task 3.3 is dedicated to designing and building the user interfaces and back-end systems for the Exploration and Matchmaking Tools. The Exploration Tool will provide interactive visualizations and causal explanations—drawing from sensitivity analyses and models developed in earlier tasks—to support decision making in urban planning. Simultaneously, the Matchmaking Tool will use context-driven algorithms, including basic NLP techniques, to match users with relevant European EO-based service providers. The development process follows an agile, iterative approach (using the DSDM, Dynamic Systems Development Methodology) that emphasizes rapid prototyping, continuous user feedback, and incremental improvements, while also incorporating an automated reporting feature to log user interactions for future enhancements.

Funded by the European Union

2. Introduction

2.1 Project Overview

UDENE aims to revolutionize geospatial information systems (GIS) by integrating advanced data cube implementations, artificial intelligence (AI) backends, and intuitive front-end tools. These innovations facilitate seamless data exploration, analysis, and matchmaking across various sectors, ensuring scalability, interoperability, and user-centric design.

Exploration Tool:
This tool is designed to help users interactively explore and visualize complex maps and datasets gathered from satellites. In simple terms, it lets urban planners, researchers, and policymakers see detailed information about city environments in a clear and interactive way. This visual and analytical approach makes it easier to understand current urban conditions and predict the impact of new development ideas.

Matchmaking Tool:
The purpose of this tool is to connect users with the right data and service providers. It uses artificial intelligence to analyze what a user needs (for example, by extracting key terms from a project idea) and then automatically recommends the most suitable European Earth Observation (EO) service providers. This way, users don't have to manually search through vast amounts of data—they get personalized recommendations that save time and improve decision-making.

2.2 Tool Objectives

The Operational Exploration Tool and Matchmaking Tool are developed to:

- Provide a user-friendly platform for exploring and analyzing GIS datasets using advanced visualization techniques.
- Enable AI-powered matchmaking to connect users with relevant datasets, tools, or services based on their requirements.
- Support decision-making processes by offering tailored insights derived from multidimensional data cubes.
- Foster collaboration among stakeholders through shared access to comprehensive data and analytics.

2.3 Intended Users and Stakeholders

The tools are designed to cater to a diverse range of stakeholders, including:

- **Policy Makers:** To inform urban planning, environmental policies, and sustainable development strategies.
- **Researchers:** To leverage data cubes and AI analytics for scientific studies and experiments.
- **Industry Professionals:** To utilize GIS tools for resource management, logistics, and strategic planning.

Funded by the European Union

- **Educators and Students:** To access advanced geospatial tools for teaching and learning purposes.

3. Tool Features and Implementation

3.1 Exploration Tool

The primary goal of the Exploration Tool is to empower urban planners, researchers, and industry professionals by providing a dynamic platform for analyzing and visualizing complex geospatial data. This tool is designed to facilitate data-driven decision-making in urban development by integrating interactive visualizations, comprehensive querying capabilities, and advanced analytical features.

The Exploration Tool is a pivotal component of UDENE, enabling users to:

- **Interactive Data Exploration:** Users can navigate and visualize GIS data cubes interactively, with capabilities such as zooming, panning, and layer toggling.
- **Advanced Querying:** Incorporates a flexible query interface allowing complex searches based on spatial, temporal, and thematic criteria.
- **Visualization Options:** Supports multiple visualization formats, including heatmaps, 3D renderings, and temporal animations.
- **Customizable Dashboards:** Provides users with the ability to create and save personalized views and dashboards.

3.2 Matchmaking Tool

The Matchmaking Tool is an integral part of the UDENE Tool, designed to serve as a bridge between users and European Earth Observation (EO) products and service providers available on eoMALL (Earth Observation Mall). This tool enables users to seamlessly connect with relevant EO-based solutions that align with their specific development ideas and needs.

The system functions as an automated recommendation engine, ensuring that users are efficiently matched with the most suitable eoMALL service providers without manually searching through large datasets. The core objective of the matchmaking tool is to provide personalized, high-accuracy recommendations that optimize decision-making and enhance the user experience.

Once users enter their development idea and complete the required form fields, they initiate the matchmaking process by clicking on the "Run Experiment" button. At this point, the system automatically extracts keywords, analyzes the input, and matches it with relevant EO products and services listed on eoMALL.

After identifying the best matches, the system scores the EO products and service providers based on their relevance and similarity to the user's development concept. Finally, it displays only the top two most relevant matches directly on the main screen, ensuring that users receive the most optimal recommendations without being overwhelmed by excessive options.

The matchmaking tool follows a four-stage process:
Extracting Keywords from User Input (NLP)

Funded by the European Union

- Users enter their development idea and complete the required form fields.
- The system applies advanced NLP (Natural Language Processing) techniques, such as Llama + LangChain, to analyze the input and extract relevant keywords and themes.
- The NLP model ensures that extracted keywords reflect the user's true intent, enabling precise matches with eoMALL services.

For example, if a user enters:
"We want to build a park to mitigate the urban heat load."
The tool will extract keywords such as "park", "urban" and "heat load" to use as search keywords among the EO products available on eoMALL and identify the most relevant.

Data Collection & Matching with eoMALL Service Providers

- The system maintains a structured database of European EO products and service providers listed on eoMALL.
- The extracted keywords are compared with stored service provider profiles to find potential matches.
- If multiple providers offer similar services, the matching algorithm selects the most relevant ones based on predefined criteria.

Matching Algorithm & Scoring System

- The system uses Cosine Similarity and Elasticsearch to measure how closely an EO service matches the user's needs.
- Each identified EO service is scored based on relevance, with higher scores assigned to products that align closely with the user's description.
- Only the top two highest-scoring service providers are selected for display.

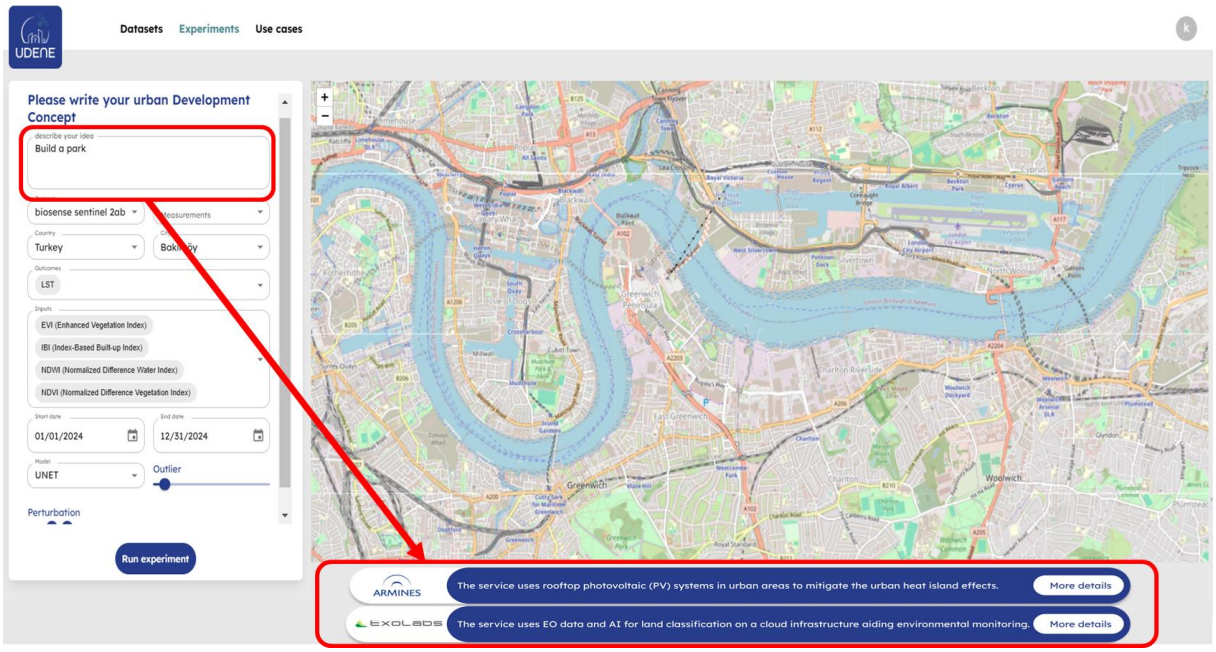
Integration into the UDENE System & User Interface

- The two most relevant EO products/services are presented directly on the main screen after the user clicks "Run Experiment."

The Matchmaking Tool leverages AI to facilitate:

- **User-Data Matching:** Suggests relevant datasets or services based on user-defined needs or behavior patterns, by extracting keywords from the urban develoıpment concept and its intended outcome that the user defines.
- **Collaborative Features:** Connects users with similar interests or overlapping objectives to encourage partnerships by providing links to the return results of the most likely products and services from the eoMALL database.
- **Feedback Integration:** Continuously improves recommendations by learning from user feedback and interactions (requirement for the M20 Beta version)

Funded by the European Union



3.2.1 User Feedback Integration

To ensure continuous improvement and personalization of the recommendation process, the system will incorporate a feedback loop. This mechanism will allow users to rate and provide input on the relevance and accuracy of the matched EO service providers. The feedback collected will be utilized to refine the matching algorithm and enhance future recommendations. In upcoming versions, user ratings and comments will be integrated into the machine learning models, ensuring that the recommendations become increasingly tailored to user needs over time.

3.3 Demonstration Cases

To illustrate the functionalities of the tools, we are developing predefined cases demonstrating their operational capabilities:

1. **Urban Planning:**
    - **Scenario:** A city planner needs to identify high-risk flood zones within urban boundaries.
    - **Demo Workflow:**
        - The Exploration Tool is used to overlay rainfall data and urban development layers.
        - Advanced querying highlights regions prone to flooding based on historical data trends.
        - The Matchmaking Tool suggests datasets for local hydrological models and connects the planner with experts in urban hydrology.
2. **Agricultural Monitoring:**
    - **Scenario:** A researcher studies the impact of drought on crop yield in a specific region.
    - **Demo Workflow:**
        - The Exploration Tool visualizes temporal NDVI data and overlays meteorological records.
        - Customizable dashboards display correlations between drought periods and vegetation health.

Funded by the European Union

        - The Matchmaking Tool identifies relevant datasets and processing workflows for detailed analysis.
3. **Biodiversity Conservation:**
    - **Scenario:** An ecologist maps wildlife corridors to support conservation strategies.
    - **Demo Workflow:**
        - The Exploration Tool displays species occurrence data alongside satellite land cover imagery.
        - Querying reveals patterns of habitat connectivity and fragmentation.
        - The Matchmaking Tool recommends best practices for conservation data analysis and connects the ecologist with collaborators in the field.

3.4 Use Case Demonstrations

1. **Serbia: Assessment of Urban Development Plans from Environmental Perspectives**
    - **Study Area:** Novi Sad, Serbia, located in the southern Pannonian plain.
    - **Objectives:**
        - Estimate pollutant emission changes caused by major transport infrastructure developments, such as the construction of a bypass and the creation of pedestrian zones.
    - **Details:**
        - Analyze the impact of building two new bridges as part of a city bypass on traffic flow and emissions.
        - Evaluate the effects of pedestrianizing the historical Petrovaradin core on traffic congestion and urban livability.
        - Leverage local traffic metrics and GIS datasets for simulation and decision-making support.

2. **Tunisia: Effect of a Linked Park System on Heat Load**
    - **Study Area:** Greater Tunis, Tunisia, situated along the Mediterranean coastline.
    - **Objectives:**
        - Mitigate urban heat islands (UHI) by implementing and enhancing a cohesive system of green spaces.
        - Assess the impact of park-based cooling strategies on Local Climate Zones (LCZs) and overall urban temperature reduction.
    - **Details:**
        - Quantify spatial and temporal variations in UHI reduction linked to park locations and vegetation types.
        - Evaluate seasonal and weather-dependent cooling efficiency of green spaces.
        - Use GIS tools to correlate park cooling indices with LCZ characteristics for urban planning insights.

3. **Turkey: Determination of Having a High-Rise District Effect in the Context of Earthquake Preparedness**
    - **Study Area:** Kadıköy-Ataşehir-Üsküdar District Municipalities of İstanbul Metropolitan Municipality, Turkey, located in the Marmara Region.
    - **Objectives:**

Funded by the European Union

- Conduct thorough assessments of earthquake-induced damage and loss in high-rise districts.
- Determine the rate of damage/undamaged buildings.
- Assess the economic impact of earthquakes in the district.
- Define in-situ variables to enable global search results for similar previous earthquake-affected areas.

- **Details:**
  - Analyze the impact of high-rise building districts on earthquake preparedness using GIS tools.
  - Evaluate scenarios for earthquake-resistant urbanization aligned with Sustainable Development Goal 11 (SDG11) for resilient cities.
  - Provide actionable insights for evidence-based land use planning and disaster risk reduction strategies.

3.5 Infrastructure Implementation

The UDENE project's infrastructure is built on a robust cloud-based architecture utilizing Amazon Web Services (AWS) to ensure scalability, reliability, and secure access to the exploration and matchmaking tools. This section details the technical implementation of the infrastructure supporting these tools.

AWS is a leading cloud computing platform worldwide. With its broad range of services and flexible infrastructure, AWS provides reliable, scalable, and cost-effective solutions for businesses and projects of all sizes. By offering a wide variety of services covering critical components such as computing, storage, networking, and security, AWS enables organizations to manage their infrastructure more efficiently compared to traditional data centers.

One of the key advantages of AWS is its high availability and low latency, achieved through globally distributed data centers. This is particularly beneficial for big data processing, real-time applications, and systems requiring high performance.

Moreover, AWS's ease of use and flexible pricing model allow organizations to dynamically scale resources based on demand. For instance, server capacity can be increased during high-traffic periods, while resources can be minimized during low-demand times to optimize costs.

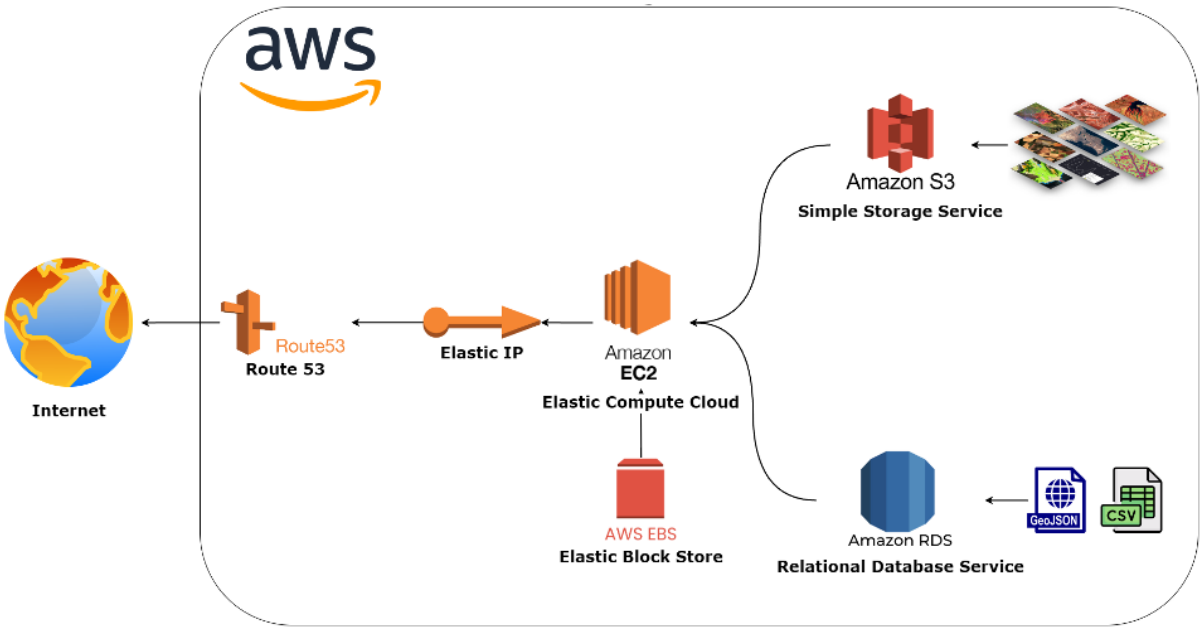
From a security perspective, AWS provides robust solutions, including data encryption, network security, access control, and compliance management. It holds international security certifications such as ISO 27001, SOC 2, HIPAA, and GDPR, making it a reliable choice for projects involving sensitive data.

In the UDENE project, AWS has been chosen to meet the scalability, reliability, and data security requirements of the system. The services provided by AWS allow the project to quickly adapt to changing needs, ensuring seamless and efficient infrastructure management.

3.5.1 Core Infrastructure Components

Funded by the European Union

The infrastructure is built around four primary AWS services, each serving specific functions in the system:



Compute Environment (EC2 Service)

The EC2 service provides a publishing environment to open the project to the outside world. It is a virtual machine with an operating system running on it. The EC2 service is a virtual machine that hosts an operating system and runs the Front-End, Back-End, and Data Cube source codes of the project while providing external access. For the operating system, Ubuntu Server was chosen due to its high performance, reliability, and extensive support network. Detailed information about the EC2 service and its configuration process is presented in the image below.

Amazon Machine Image (AMI)

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type
ami-0e9085e60087ce171 (64-bit (x86)) / ami-08ad90b57490ff3e3 (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

Description

Ubuntu Server 24.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (http://www.ubuntu.com/cloud/services).

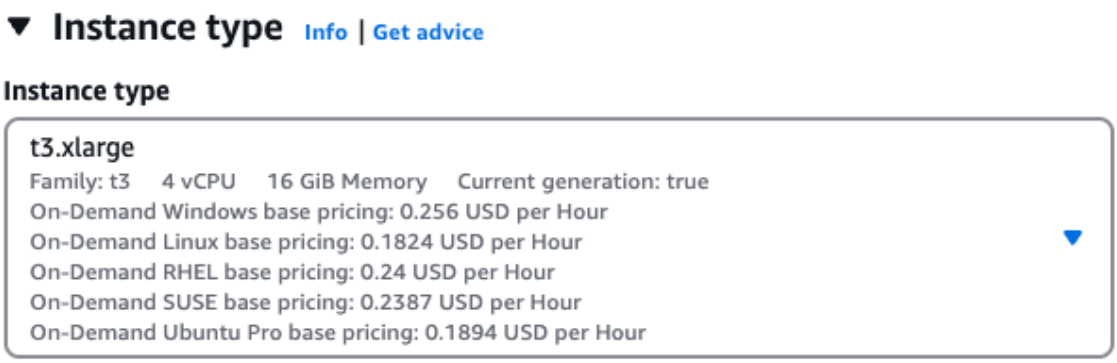
Canonical, Ubuntu, 24.04, amd64 noble image

Architecture	AMI ID	Username	
64-bit (x86)	ami-0e9085e60087ce171	ubuntu	Verified provider

Initially, the t3.xlarge server (4 vCPU, 16 GiB RAM) was selected as the instance type for the project. This choice was made to keep costs low while providing adequate performance during the development and testing phases. AWS offers a variety of server types tailored to different workloads. T3 servers are low-cost, flexible-performance instances that are ideal

Funded by the European Union

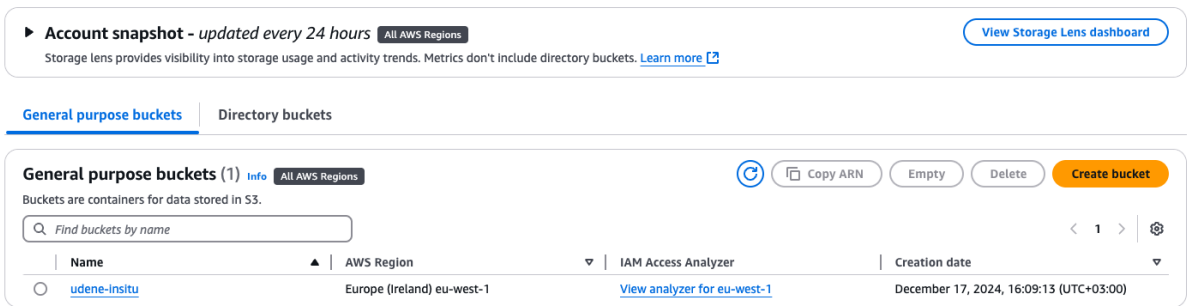
for development and testing environments. These servers provide burstable CPU performance, making them suitable for workloads with variable resource demands.



In contrast, M5 servers are optimized for general-purpose workloads. They offer a balanced combination of compute power and memory, making them well-suited for large-scale applications or workloads requiring high performance. During the production phase, the servers will be scaled to M5 specifications to accommodate increased performance needs. The diagram below provides detailed specifications of the initial instance type used.
In the selected server model, Amazon's Pay-As-You-Go pricing policy has been adopted. Under this model, users are charged on an hourly basis depending on the server's usage time. It is possible to pause the server when it is not in use, allowing users to save costs. This pricing policy offers flexibility to users by enabling them to pay only for actual usage. However, the rate of cost increase will vary depending on the server's usage intensity and operating time. It is important to consider that costs may rise quickly in scenarios with high usage.

Data Storage (S3 Service)

Amazon Simple Storage Service (S3) is an object-based storage service offering high durability, scalability, and security. It provides a reliable solution for storing data that can be easily accessed, secured, and managed. S3 is widely used for various scenarios, including backup, archiving, and big data processing workflows. With its customizable storage classes, S3 allows for optimization based on cost and performance requirements. As part of the UDENE project, insitu data provided by project partners have been specifically configured for the S3 service. These datasets are organized in a structured format, optimized for easy access and fast processing. The structure on S3 not only meets the data requirements of other system components but also ensures secure storage of the data.

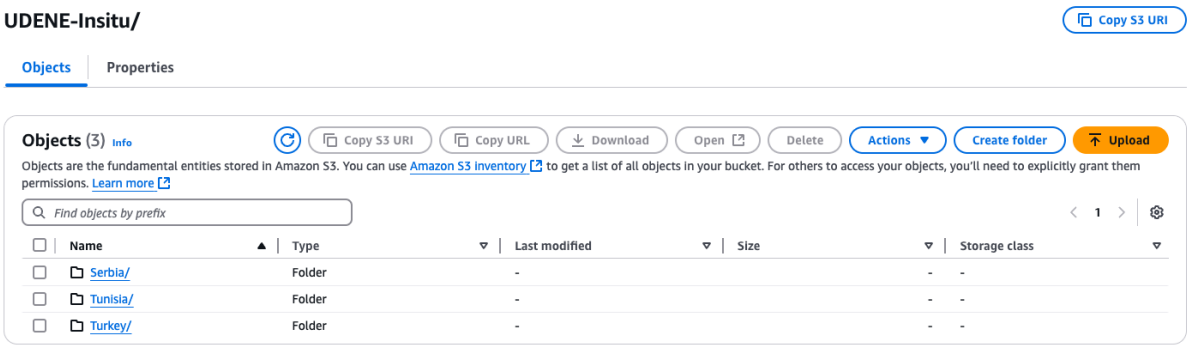


Funded by the European Union

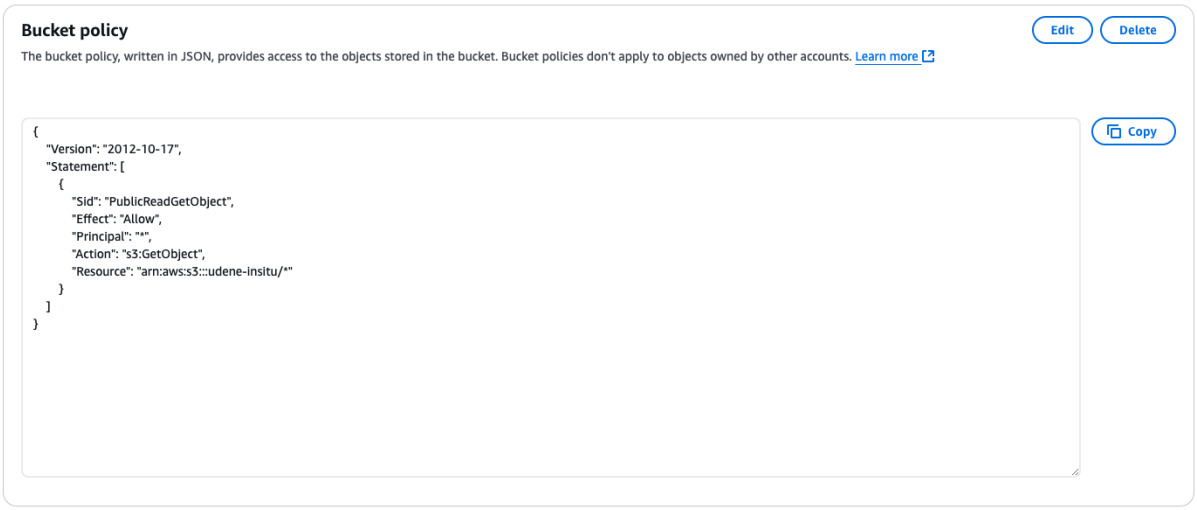
The S3 bucket created for the UDENE project is named udene-insitu. Access to this bucket is configured as "public read" to allow data access by both the developed Data Cube system and external Data Cube systems. However, in line with security best practices, direct "public write" access to the bucket is not granted. Data upload and update operations are performed securely through authorized systems and roles. Separate folders have been created within the bucket for the three partners responsible for the use case (Serbia, Tunisia, and Turkey). This folder structure facilitates data organization and allows each partner to easily access their own data. The folder structure is as follows:

- udene-insitu/Serbia
- udene-insitu/Tunisia
- udene-insitu/Turkey

This structure not only ensures logical separation and management of data but also simplifies access control and authorization processes.

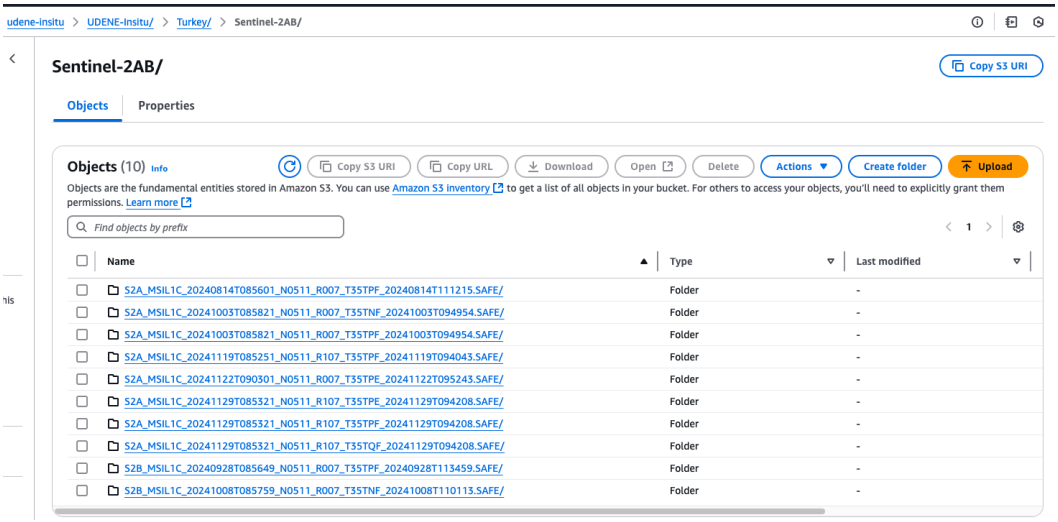


The policy applied to the udene-insitu S3 bucket grants public read access (including anonymous users) to all objects within the bucket. This configuration is designed to allow both the developed Data Cube system and other external Data Cube systems to freely access the relevant data and achieve integration. The policy allows the s3:GetObject action, enabling any user to access and download the data within the bucket. This access configuration is implemented due to the public nature of the data and, specifically, to facilitate the connection and integration of other external cubes with this data. This supports data sharing and interoperability between different platforms and systems.



Funded by the European Union

Within the udene-insitu S3 bucket, in addition to the original data provided by each partner,



there are also sample datasets located in the respective partner folders. These sample datasets consist of 10 images from the Sentinel-2A and Sentinel-2B satellites, covering the study areas of each partner (Serbia, Tunisia, and Turkey). These images have been specifically prepared to be compatible with the Data Cube format. Each image is separated into its spectral bands, and each band is provided in Cloud Optimized GeoTIFF (COG) format. The COG format is optimized for cloud-based access and processing, enabling efficient handling and visualization of large geospatial datasets. This configuration is designed to test and demonstrate the Data Cube system's integration and performance with various data sources.

The data residing in the udene-insitu S3 bucket can be accessed via the following addresses:

- s3://udene-insitu/UDENE-Insitu/ (For AWS CLI and SDKs)
- https://udene-insitu.s3.eu-west-1.amazonaws.com/UDENE-Insitu/ (For web browsers and HTTP clients)

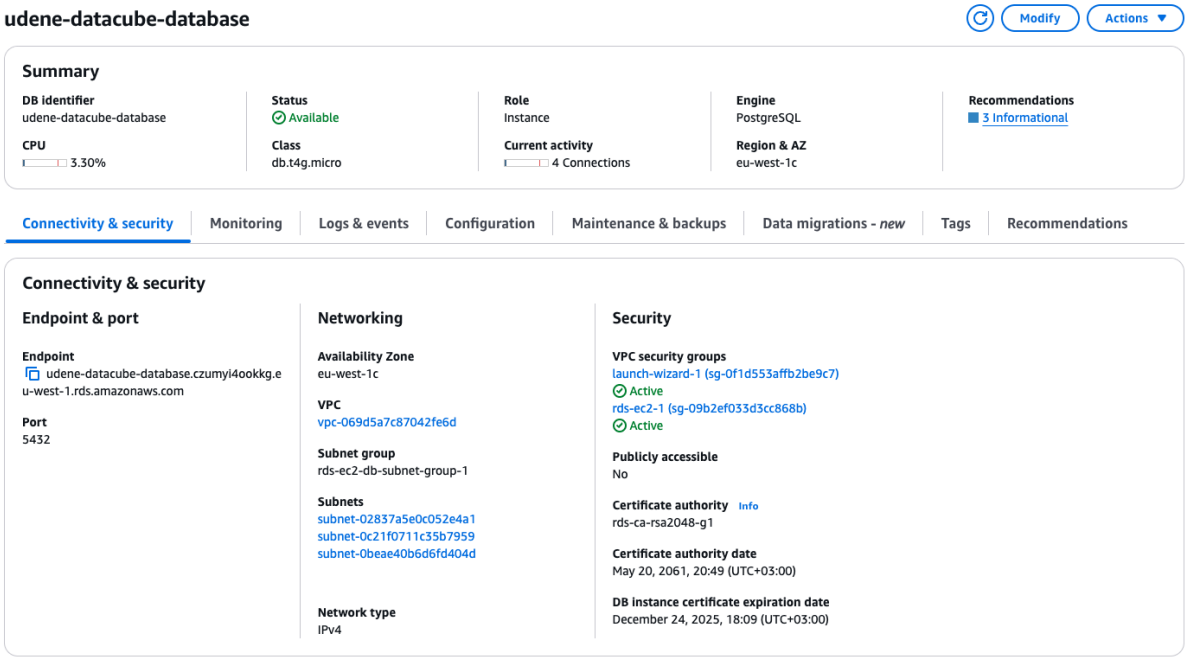
These addresses allow both the Data Cube system and other external systems, as well as authorized users, to access the data programmatically or directly. The data is readily available at the specified locations.

Database Services (RDS Service)

Amazon Relational Database Service (RDS) is a managed service offered by Amazon Web Services (AWS) that simplifies setting up, operating, and scaling relational databases in the cloud. RDS automates many of the complex tasks associated with database management, allowing users to focus on application development. This service manages operational tasks such as hardware provisioning, database setup, patching, backups, recovery, scaling, and monitoring. RDS supports various database engines, including MySQL, PostgreSQL, MariaDB, Oracle, SQL Server, and Amazon Aurora. Within this project, the PostgreSQL database engine has been selected for data storage and management. PostgreSQL is a widely used relational database management system known for its robust features, open-source nature, and extensive community support. Using PostgreSQL on RDS provides a scalable and reliable solution to meet high availability, security, and performance requirements.

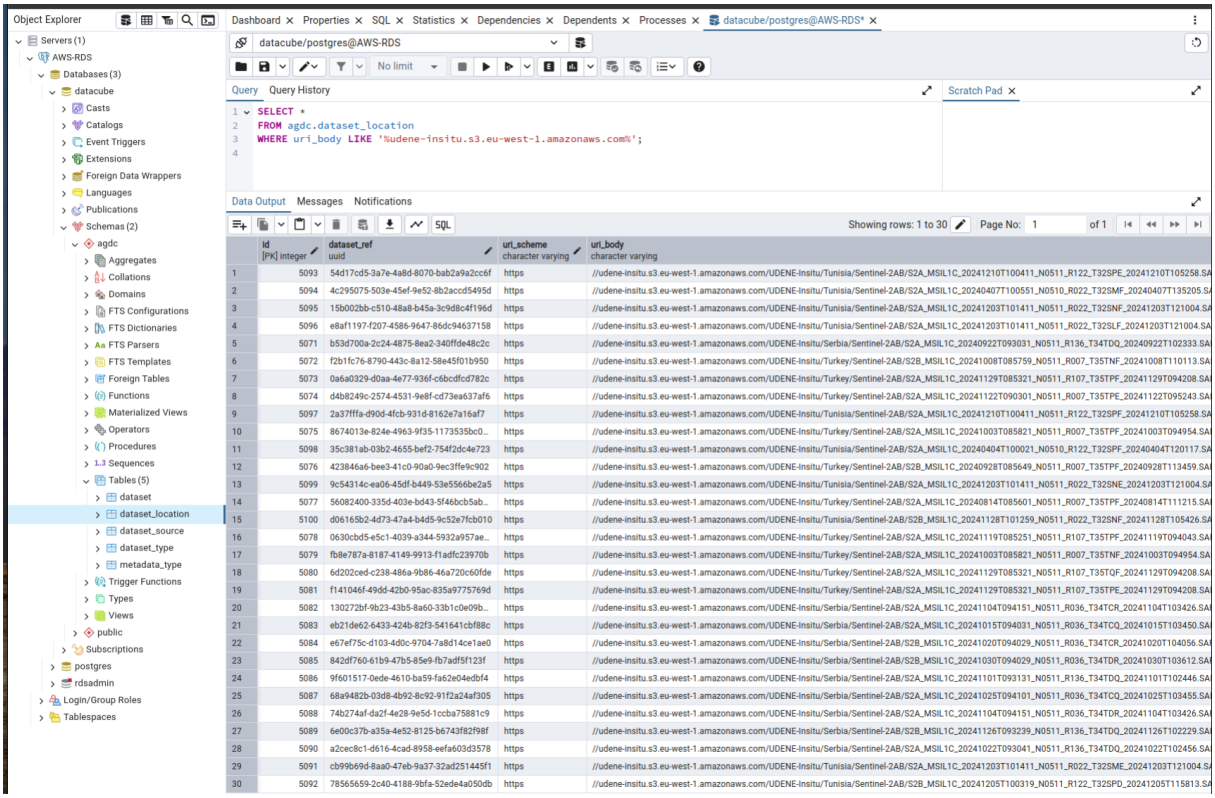
Funded by the European Union

In this project, the PostgreSQL database running on AWS RDS serves two primary purposes. Firstly, the developed Data Cube project requires a PostgreSQL database to meet its operational needs. All metadata and relational data indexed by the Data Cube system are stored in this database. This metadata contains comprehensive information 3about satellite imagery, sensor data, and other geospatial data, and is crucial for the efficient operation of the Data Cube. Secondly, geospatial data in vector format obtained from in-situ measurements is also stored in this database. PostgreSQL's PostGIS extension provides support for geospatial data types and functions, enabling efficient storage, querying, and analysis of this data. PostGIS offers the necessary tools for storing geographic objects, executing spatial queries, and performing geographic analyses. This allows both Data Cube metadata and in-situ vector data to be managed and analyzed in an integrated manner.



The PostgreSQL database hosted on AWS RDS forms the core data repository for the Data Cube project. In this database, the metadata of all data managed by the Data Cube system is indexed. This metadata contains comprehensive information about geospatial data obtained from various sources. For example, images from the Sentinel-2A and Sentinel-2B satellites uploaded to the S3 storage are indexed in this database. For each image, information such as acquisition date, geographic extent, cloud cover, band information, and file location are stored in the database. This indexing allows the Data Cube system to quickly and efficiently search through large volumes of satellite data and easily access the desired data. In addition, sample datasets from other data cubes supported within the project are also indexed in the database. This is done to demonstrate and test the Data Cube system's ability to integrate with different data sources. The database schema is designed to support different data types and sources and provides flexibility for future data integrations.

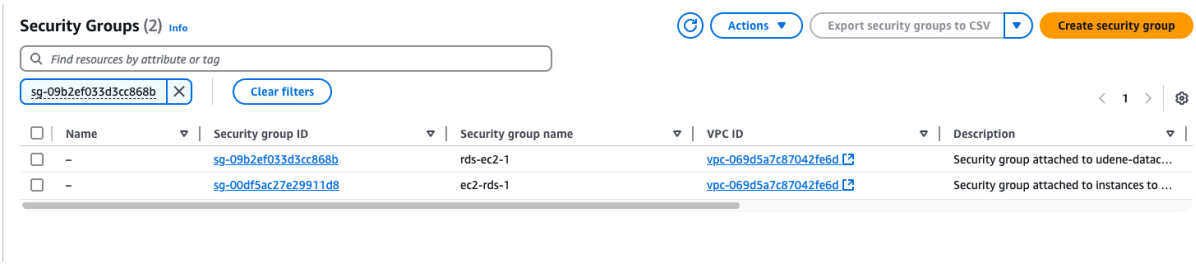
Funded by the European Union



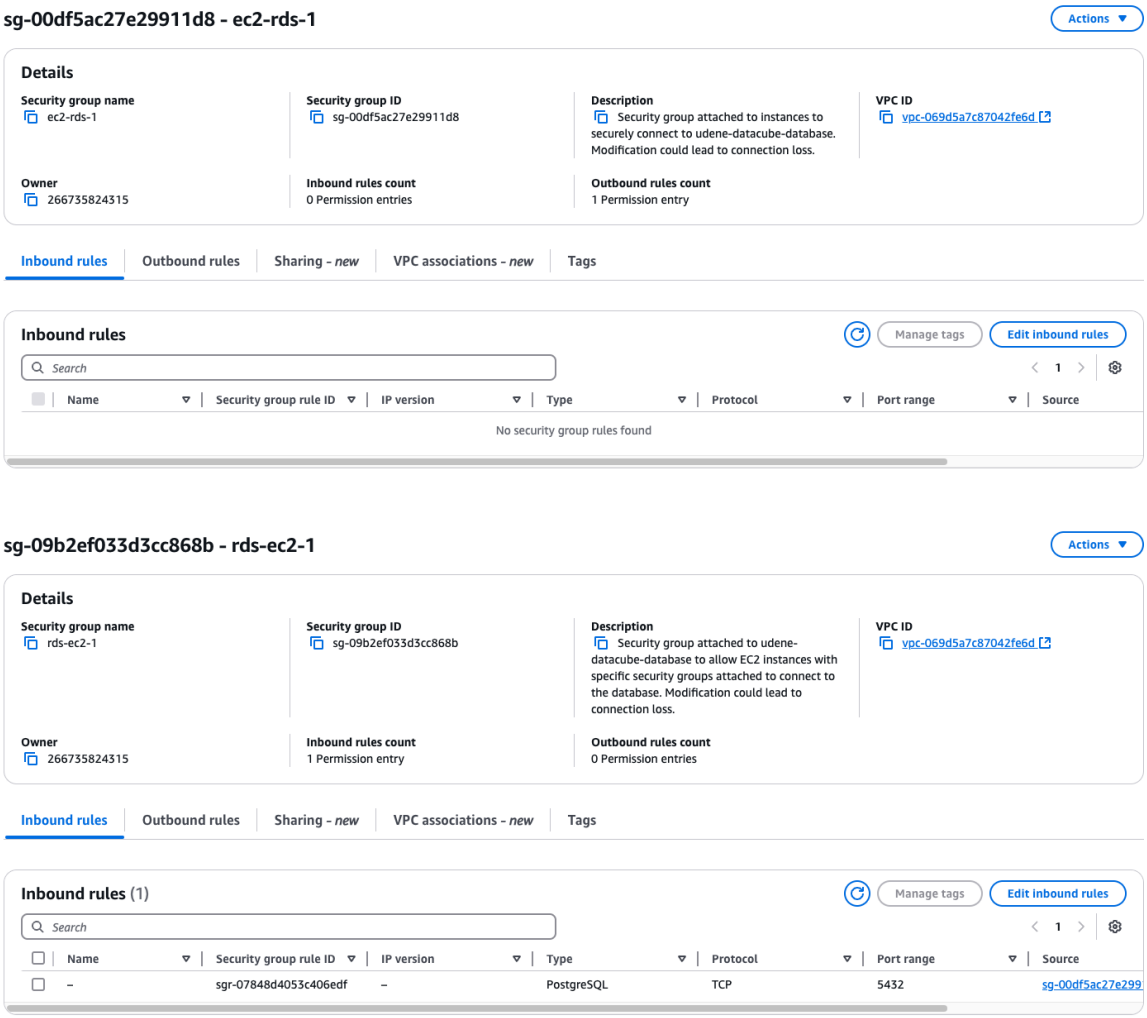
Specific steps were taken to ensure that the PostgreSQL database running on AWS RDS can securely connect to a pre-existing EC2 instance. In this connection process, the roles of the AWS Identity and Access Management (IAM) service are critical. IAM is a service used to securely manage access to AWS resources. In this context, an IAM role has been assigned to the EC2 instance. This role allows the EC2 instance to have access permission to the RDS database. The role's permissions are limited to include only the necessary database operations (e.g., connecting, querying, reading/writing data). This enhances security by following the principle of least privilege.

The connection steps are generally as follows:

1. Security Groups: Security groups have been configured for both the EC2 instance and the RDS database. Security groups are virtual firewalls that control incoming and outgoing traffic. The EC2 security group is configured to allow traffic to the RDS security group over a specific port (usually 5432 for PostgreSQL). Similarly, the RDS security group is set up to allow traffic only from the EC2 security group. This prevents unauthorized access.



Funded by the European Union



2. IAM Role Assignment: An IAM role was assigned when the EC2 instance was created or later. This role contains a policy that grants access to the RDS database. The policy specifies the ARN (Amazon Resource Name) of the database instance and the allowed actions.

3. Connection Setup: A connection to the RDS database was established from within the EC2 instance using PostgreSQL client tools (e.g., psql) or an application. For the connection, information such as the RDS database endpoint, port number, database name, username, and password are used. Thanks to the IAM role, the 5EC2 instance can securely connect to RDS without having to directly manage AWS credentials.

This configuration provides a secure and controlled connection between EC2 and RDS. Thanks to IAM roles, sensitive database credentials do not need to be stored on the EC2 instance, which reduces security risks.

3.5.2 Network Configuration

The infrastructure implements a comprehensive security framework to protect data while ensuring accessibility:

Network Access Control

- Elastic IP service provides a static IPv4 address (46.137.89.112) for consistent access
- Route53 service manages domain name resolution and external accessibility

Funded by the European Union

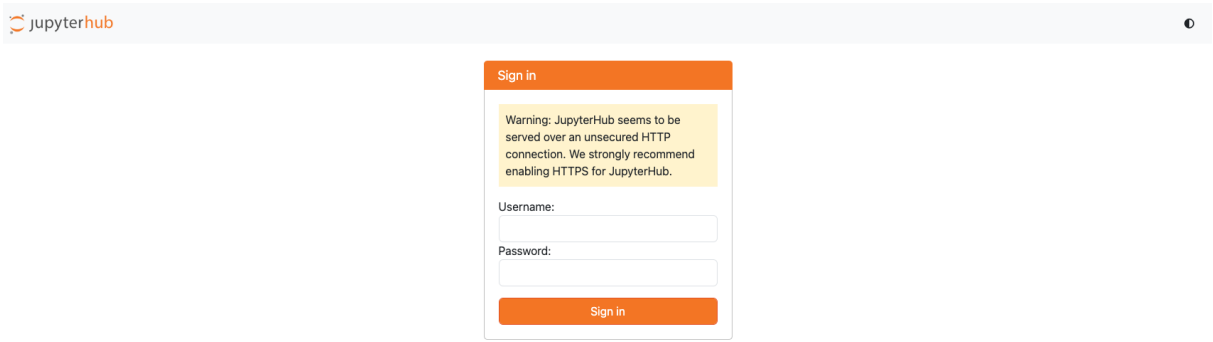
- Security groups control inbound and outbound traffic for both EC2 and RDS instances
- Implements specific port configurations:
    - Port 80: HTTP access to main server
    - Port 443: HTTPS secure connections
    - Port 3000: Front-end application access
    - Port 5000: Back-end API services
    - Port 5432: PostgreSQL database connections

3.5.3 Development Environment Integration

The infrastructure supports an interactive development environment enabling efficient tool development and testing:

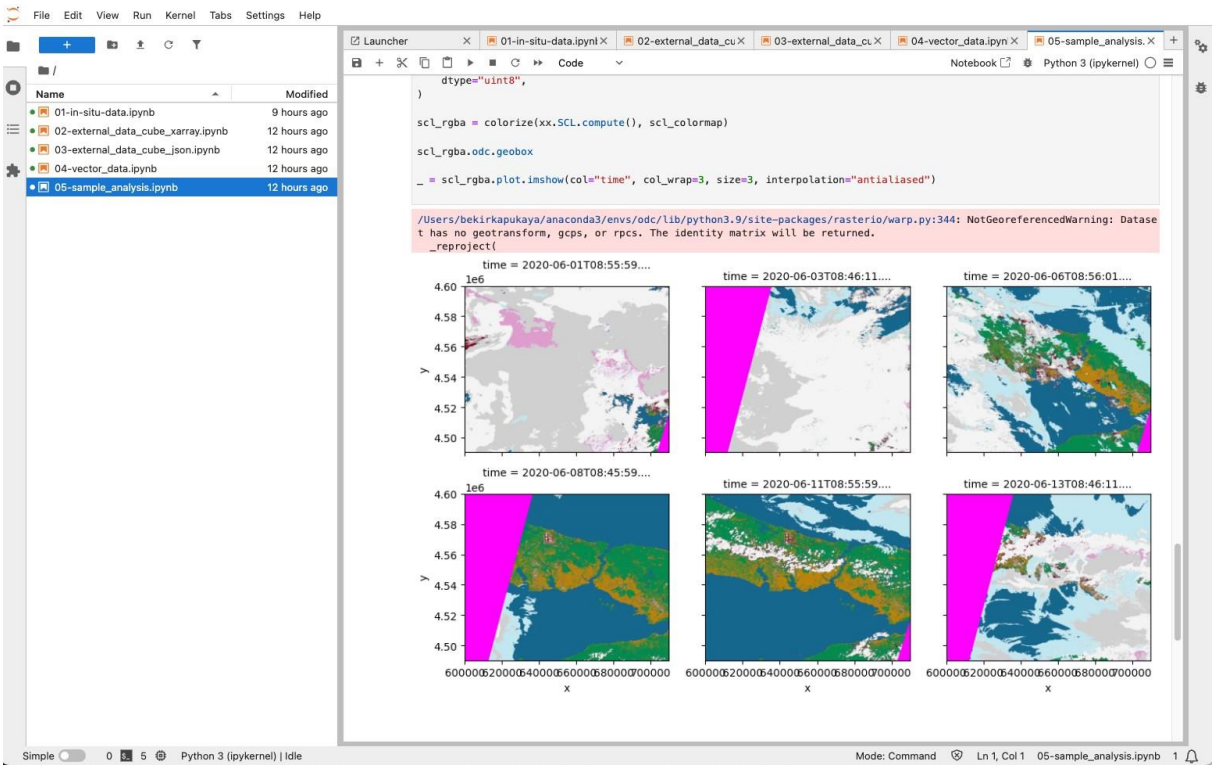
JupyterHub Integration

- Web-based Python development environment accessible via port 8000
- Pre-configured notebooks demonstrating tool capabilities and use cases
- Direct access to Data Cube functionalities and analysis tools
- Integrated with AWS services for seamless data access and processing

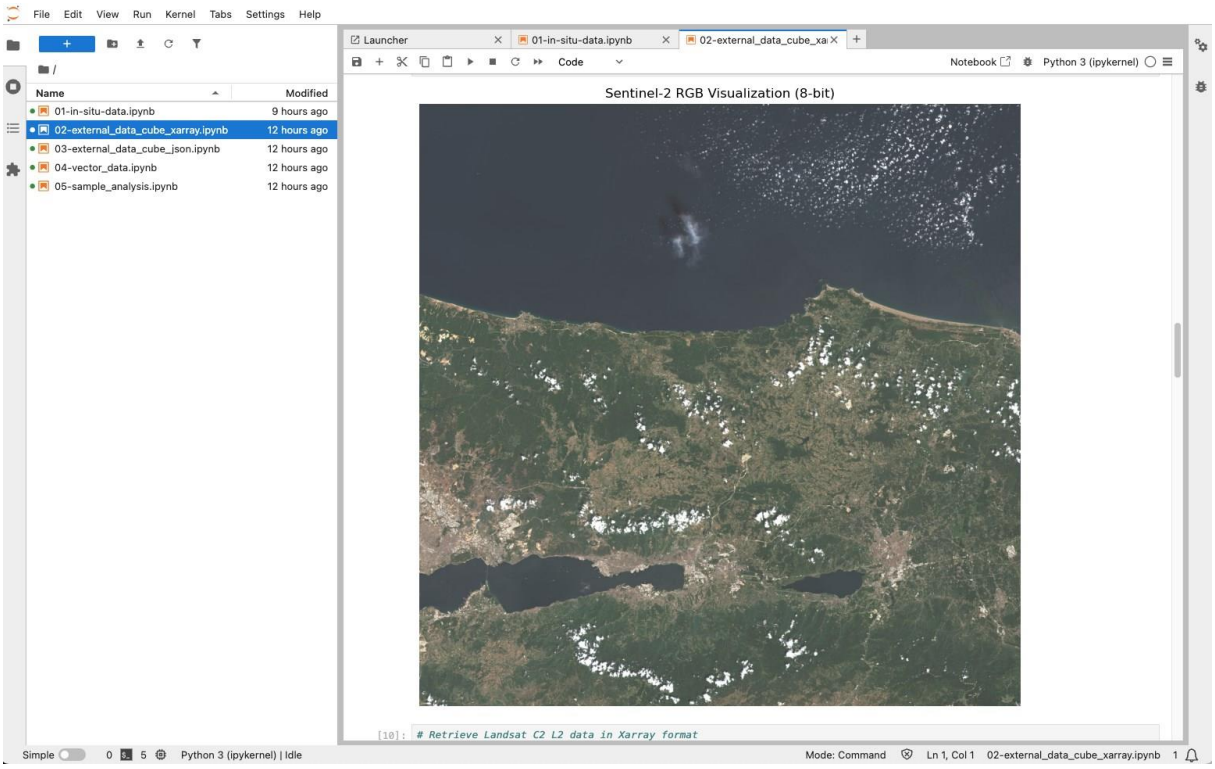


JupyterHub login interface, where users enter their credentials to access the shared Python development environment.

Funded by the European Union

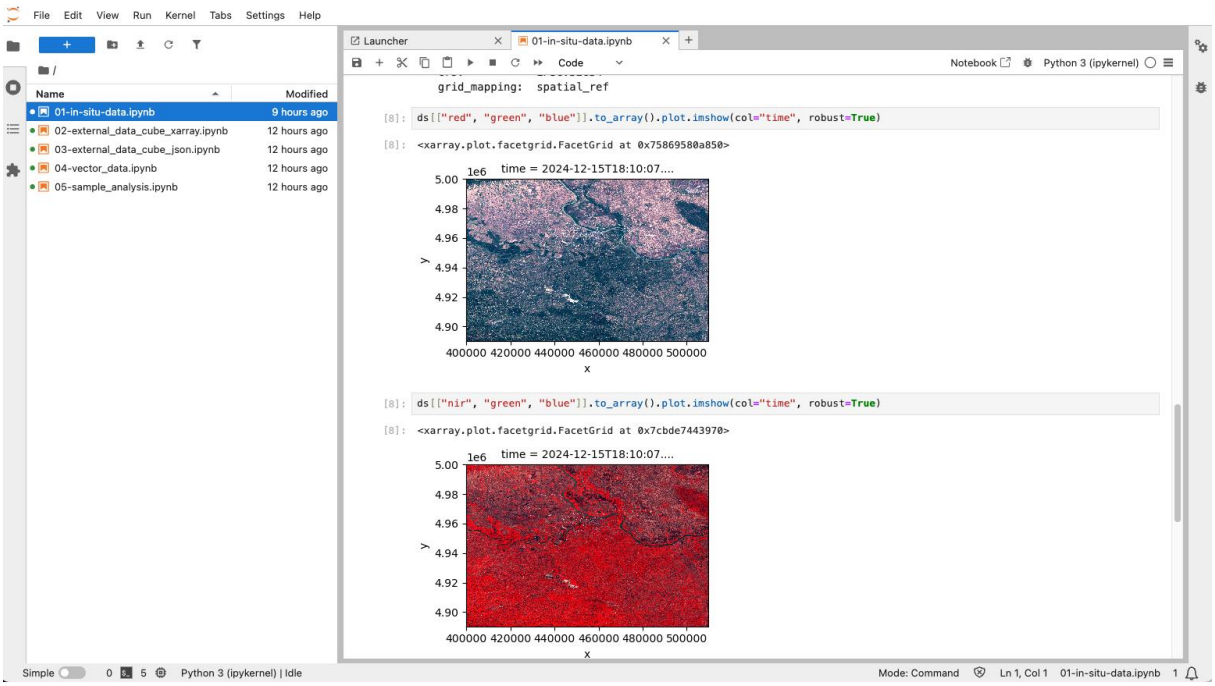


Here, the user is working within a Jupyter notebook that demonstrates loading and visualizing Sentinel-2 data.



An example notebook for an RGB composite of Sentinel-2 imagery from an external datacube.

Funded by the European Union



Here, two images illustrate different band combinations or indices (e.g., near-infrared, red, green) for enhanced vegetation detection from our datacube.

3.5.4 Scalability and Performance Considerations

The infrastructure is designed with scalability in mind, implementing several key features:

Resource Optimization

- Automated scaling capabilities to handle varying workloads
- Performance monitoring and analytics for resource utilization
- Cost-effective resource allocation through AWS's flexible pricing models
- Buffer capacity for peak usage periods during data processing and analysis

Funded by the European Union

3.5.5 Maintenance and Monitoring

Comprehensive monitoring and maintenance protocols ensure system reliability:

System Monitoring

- Real-time performance metrics tracking
- Automated alerts for system anomalies
- Regular backup procedures for critical data
- Continuous security assessments and updates

This infrastructure implementation provides a solid foundation for the Exploration Tool and Matchmaking Tool, ensuring reliable performance, secure data handling, and scalable operations across all use cases. The architecture's flexibility allows for future enhancements and adaptations as project requirements evolve.

3.5.6 Cost and Usage Overview

AWS offers various pricing models designed to provide flexibility and cost efficiency for different workloads. These models help optimize costs based on specific usage scenarios. The primary AWS pricing models include:

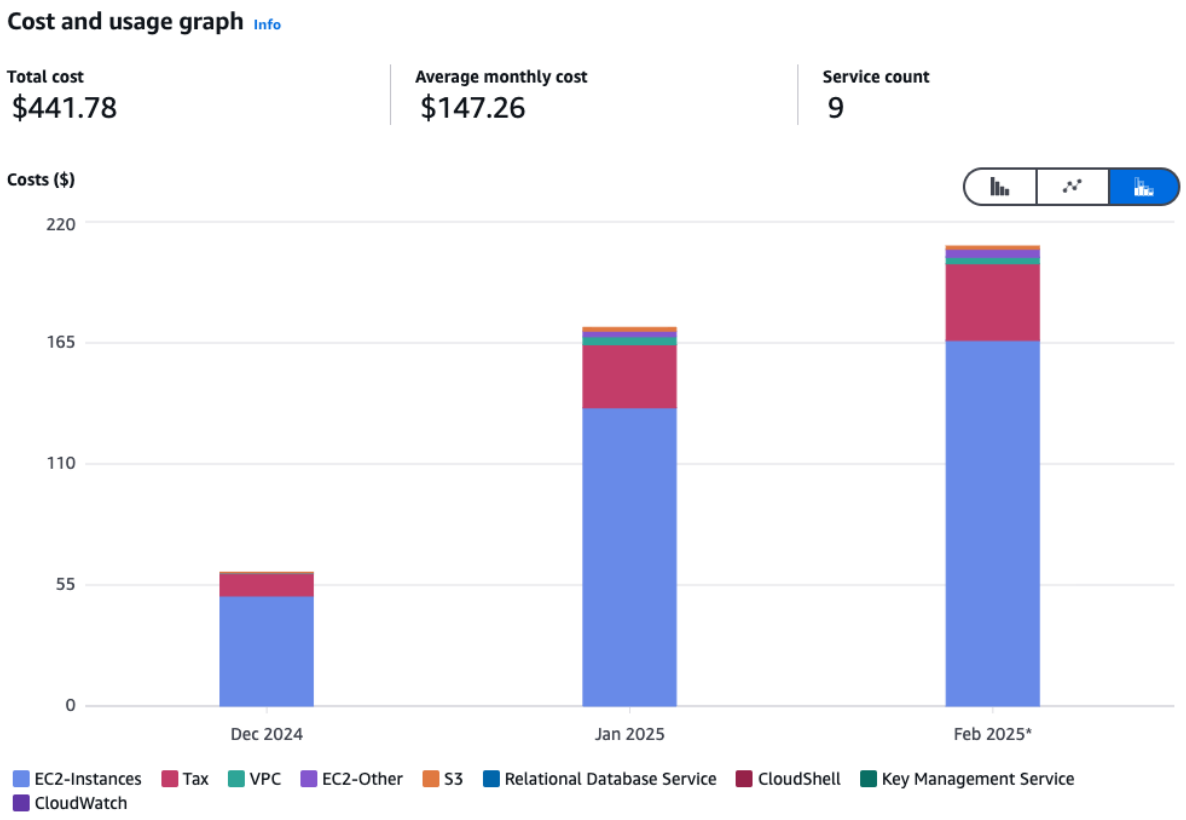
- On-Demand Pricing: Users pay only for the resources they consume without any long-term commitments.
- Reserved Instances: Users reserve resources for a fixed period (1 or 3 years) at a discounted rate.
- Spot Instances: Provides access to unused AWS capacity at lower costs.
- Savings Plans: Offers flexible long-term pricing discounts based on usage commitments.
- Pay-as-You-Go Model: Users are billed based on actual resource consumption, allowing for cost-effective scalability.

The UDENE project utilizes the Pay-as-You-Go pricing model. This approach ensures cost-efficient scalability by allowing AWS services (EC2, S3, RDS, etc.) to be billed based on actual usage. Due to the varying workload and data processing needs of the project, this model was chosen to provide flexibility and efficiency. With AWS's automatic scaling and resource management capabilities, additional resources can be provisioned as demand increases, while unnecessary costs are minimized during lower usage periods.

The bar chart below illustrates the total costs of all AWS services used during the period from December 2024 to February 2025. This visualization helps identify which services contribute the most to the overall expenses by displaying the proportion of each service's cost.

The EC2 service has the highest cost during this period, accounting for a significant portion of the total expenditure. Tax costs are the second largest expense, following EC2. Other services such as RDS, S3, EBS, and Route 53 also contribute to the overall costs but at lower levels.

Funded by the European Union



Additionally, the following table provides a detailed breakdown of the costs incurred for each AWS service from December 2024 to February 2025. It presents the monthly expenses for each service, enabling a deeper analysis of spending patterns. The fluctuations in EC2 and tax costs indicate variations in system usage intensity over time.

Cost and usage breakdown

Download as CSV

Find cost and usage data

1

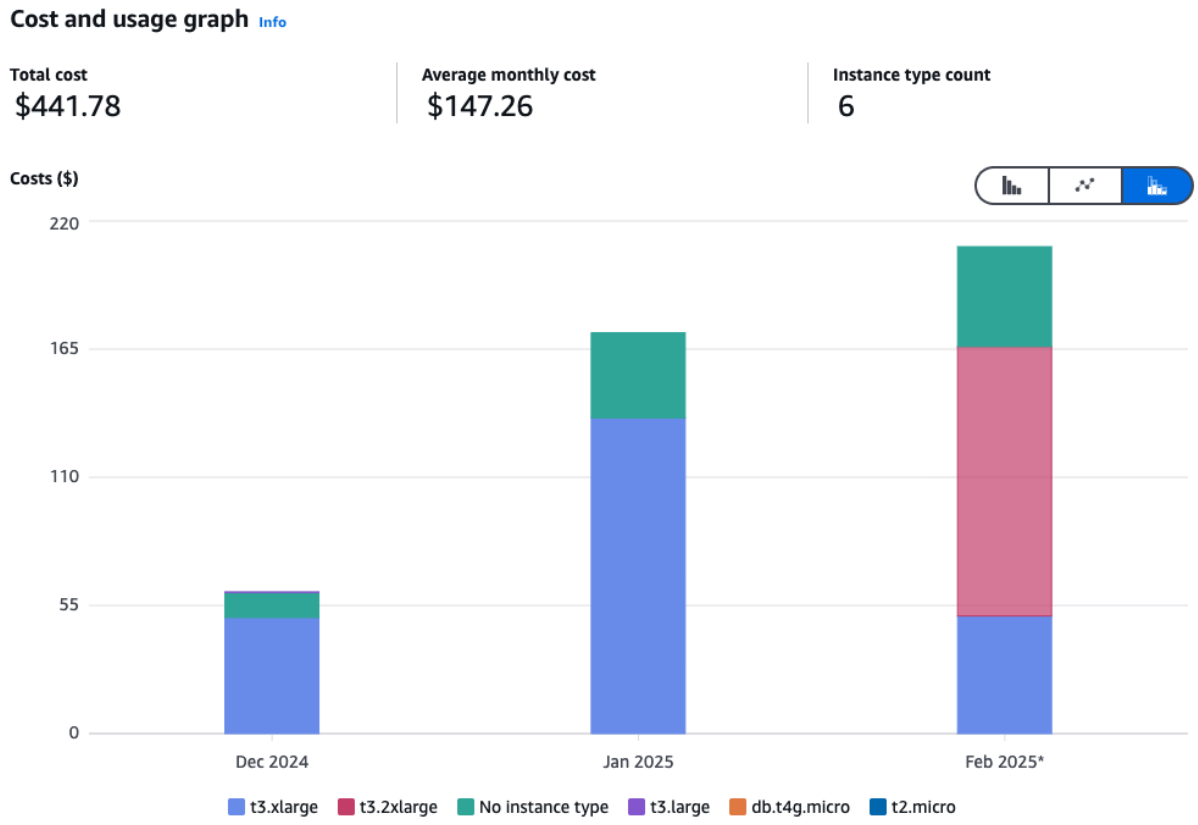
Service	Service total	December 2024	January 2025	February 2025*
Total costs	$441.78	$60.79	$172.02	$208.96
EC2-Instances	$351.70	$50.05	$135.51	$166.13
Tax	$73.62	$10.13	$28.67	$34.82
VPC	$6.96	$0.36	$3.63	$2.97
EC2-Other	$6.23	$0.02	$2.58	$3.63
S3	$3.28	$0.23	$1.63	$1.42
Relational Database Service	$0.00	$0.00	$0.00	$0.00
CloudShell	$0.00	$0.00	-	-
Key Management Service	$0.00	$0.00	$0.00	$0.00
CloudWatch	$0.00	$0.00	$0.00	-

The bar chart below illustrates the cost distribution of different EC2 and RDS instance types from December 2024 to February 2025. This visualization highlights cost variations due to changes in system resource requirements over time.

Funded by the European Union

- December 2024: The t3.xlarge instance type incurred the highest cost, as the system was initially deployed with this instance. The second-largest expense was associated with the db.t4g.micro instance used for database operations.
- January 2025: The t3.xlarge instance remained the highest-cost resource, followed by db.t4g.micro.
- February 2025: The t3.2xlarge instance became the most expensive instance due to an upgrade from t3.xlarge, which was no longer sufficient for the system's resource demands. Additionally, t3.large and db.t4g.micro contributed significantly to the overall costs.

These insights demonstrate how scaling decisions impact overall costs as the project evolves.



The table below provides a detailed breakdown of monthly costs per instance type for the period December 2024 - February 2025. By displaying the exact costs incurred by each instance type per month, the table offers a numerical representation of cost trends.

In particular, the cost increase in February 2025 for the t3.2xlarge instance resulted from the system's resource demands exceeding the capabilities of the previously used t3.xlarge instance. This dataset serves as a valuable reference for evaluating the long-term impact of instance type selection on cost management.

Funded by the European Union

Cost and usage breakdown

Download as CSV

Find cost and usage data

1

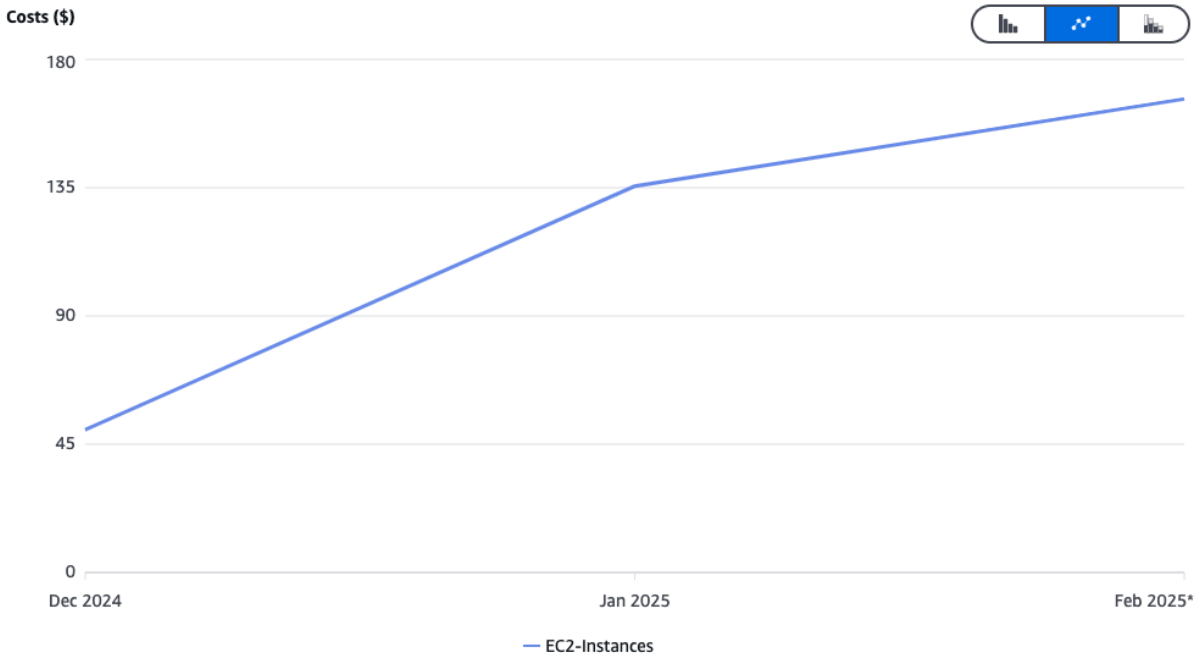
Instance type	Instance type total	December 2024	January 2025	February 2025*
Total costs	$441.78	$60.79	$172.02	$208.96
t3.xlarge	$235.81	$49.81	$135.51	$50.49
t3.2xlarge	$115.64	-	-	$115.64
No instance type	$90.08	$10.73	$36.51	$42.84
t3.large	$0.25	$0.25	-	-
db.t4g.micro	$0.00	$0.00	$0.00	$0.00
t2.micro	$0.00	$0.00	-	-

The graph below illustrates the cost distribution exclusively for the Amazon EC2 service from December 2024 to February 2025. Given that the total AWS cost for this period was $441.78, EC2 accounted for $351.70, making it the most significant contributor to overall expenses.

The average monthly EC2 cost was $117.23, and costs increased steadily over the three months due to growing resource demands. In particular, the transition to a more powerful instance type (t3.2xlarge) in February 2025 was a key factor in the rising EC2 expenses.

Cost and usage graph Info

Total cost	Average monthly cost	Service count
$351.70	$117.23	1



The graph below represents the tax costs associated with AWS services for the period from December 2024 to February 2025. Over this three-month period, a total of $73.62 was paid in taxes, with an average monthly tax cost of $24.54.

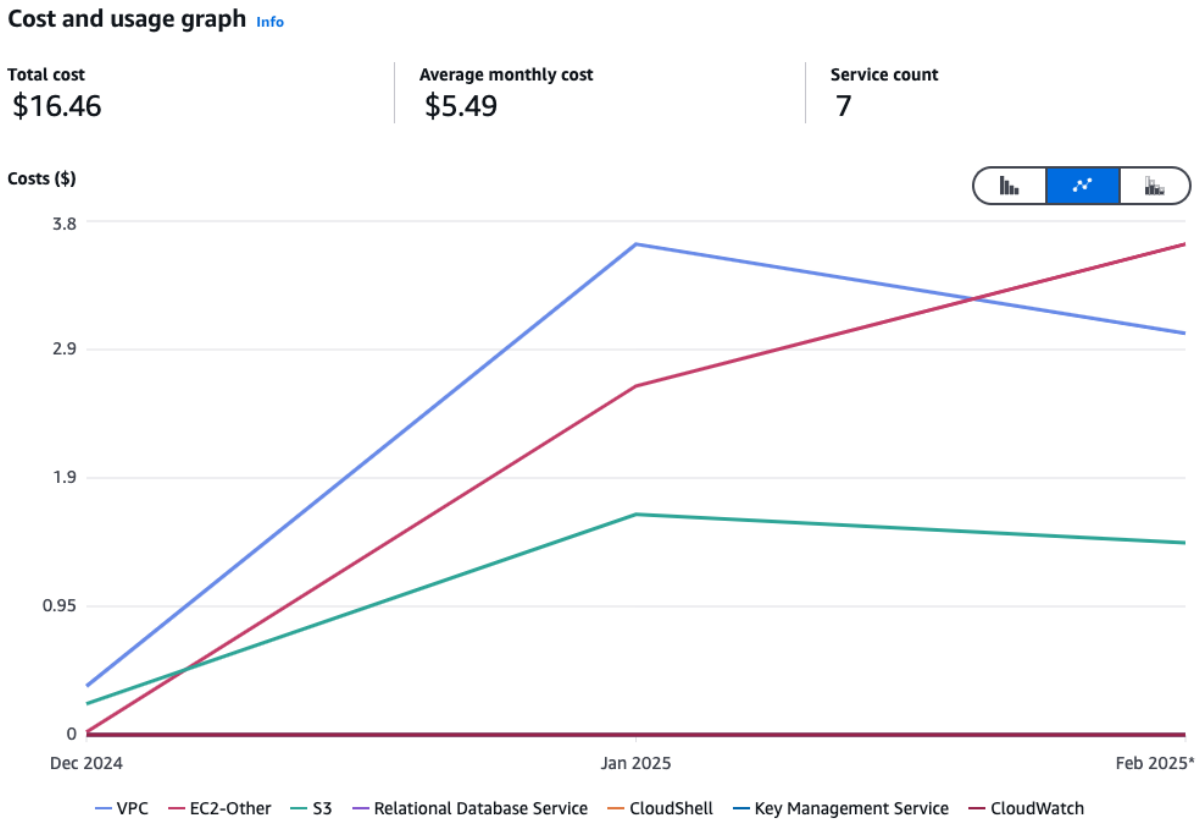
Funded by the European Union

Tax payments have steadily increased each month, primarily due to the overall rise in AWS service usage. In terms of total expenses, taxes ranked as the second-largest cost item after EC2 services.

The graph below illustrates the cost distribution of VPC, S3, RDS, CloudShell, CloudWatch, Key Management Service, and EC2 Other services for the period from December 2024 to February 2025. The total cost of these services over the three-month period amounted to $16.46, with an average monthly cost of $5.49.

The cost trend shows an increase from the first to the second month, followed by stabilization in the third month. This pattern reflects the implementation of cost-effective and resource-efficient strategies while configuring AWS services.

Funded by the European Union



3.6 Backend Implementation and API Services

3.6.1 System Architecture and Technical Foundation

The UDENE platform's backend architecture is meticulously designed to handle intricate geospatial data processing under specific infrastructure constraints. This section delves into the chosen implementation strategies, explores their technical impact, and elaborates on the key reasoning behind major architectural decisions.

Infrastructure Context and Constraints

The current deployment runs on an AWS t3.xlarge instance (4 vCPU, 16 GiB RAM) with Ubuntu Server. While sufficient for the alpha phase and testing, this setup imposes several operational boundaries that guided our architectural design:

1. Memory Management Limitations

- Operating under 16 GiB RAM demands strategic memory management techniques.
- Concurrent satellite image processing may consume more RAM than is available.
- CPU burst capabilities on t3.xlarge necessitate careful resource allocation.

2. Storage and I/O Challenges

- Limited local SSD capacity affects how temporary files are managed during processing.

Funded by the European Union

- Network bandwidth impacts data transfer rates between S3 and the processing server.
- Preventing I/O bottlenecks is critical during high processing loads.

Framework Selection and Rationale

FastAPI was chosen as the primary framework based on the following considerations:

1. Performance

- Asynchronous processing supports efficient concurrency.
- Built-in Pydantic ensures robust data validation with low overhead.
- FastAPI's lightweight footprint maximizes use of available resources.

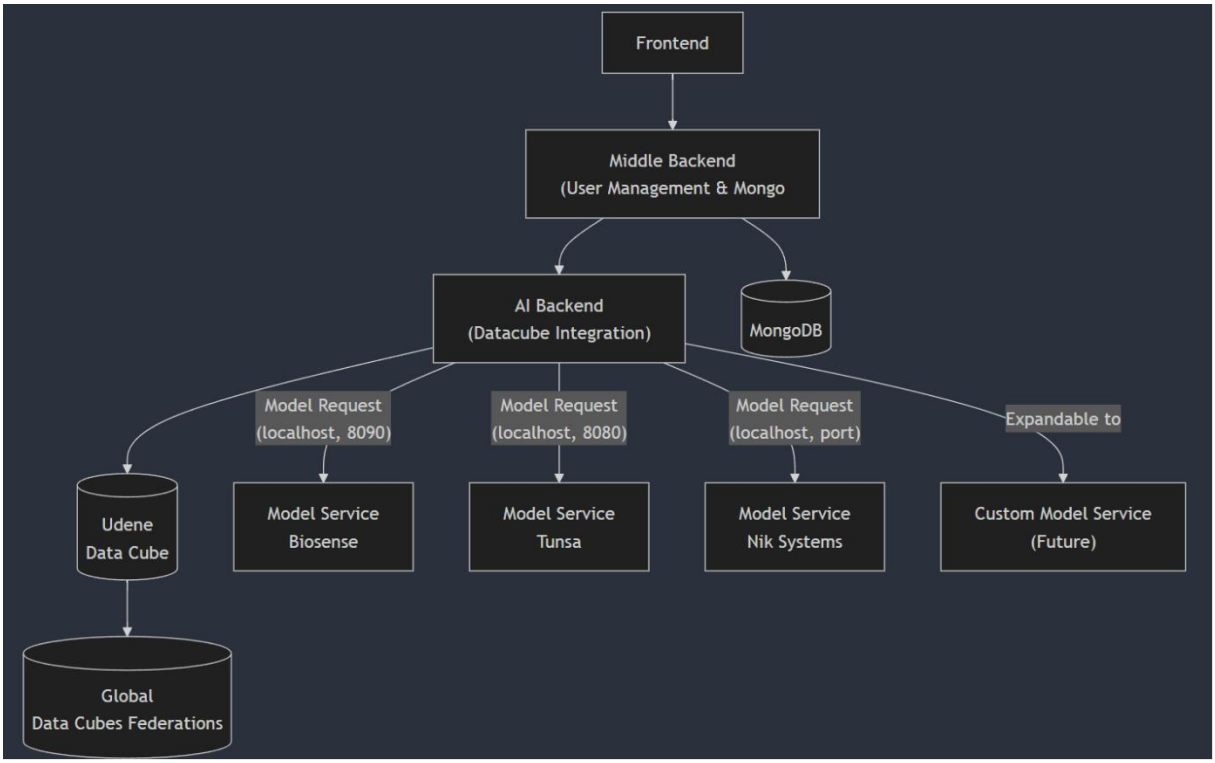
2. Development Efficiency

- Automatic OpenAPI documentation simplifies both testing and integration.
- Type hints and modern Python capabilities boost maintainability.
- Native async functions streamline complex geospatial tasks.

3. Integration Flexibility

- Smooth interoperability with Python's scientific libraries (e.g., numpy, xarray, dask).
- Straightforward machine learning integration (e.g., PyTorch).
- Flexible middleware options for customizing.

3.6.2 Core System Components



Funded by the European Union

Memory Management System

To handle resource constraints effectively, the platform implements a specialized memory management mechanism:

```
MAX_MEMORY_THRESHOLD = 1024  # MB
os.environ['GDAL_DISABLE_READDIR_ON_OPEN'] = 'EMPTY_DIR'
os.environ['CPL_VSIL_CURL_ALLOWED_EXTENSIONS'] = '.tif'

def check_memory_usage():
    """
    Monitors system memory usage and triggers garbage collection
when needed.
    Prevents memory overflows during large-scale data processing.
    """
    process = psutil.Process()
    memory_mb = process.memory_info().rss / 1024 / 1024
    if memory_mb > MAX_MEMORY_THRESHOLD:
        gc.collect()
        raise HTTPException(
            status_code=503,
            detail=f"Memory usage too high ({memory_mb:.2f}MB).
Please try with smaller chunks."
        )
```

Key benefits:

1. Memory Monitoring

- Constantly tracks RAM usage and proactively runs garbage collection.
- Allows graceful handling of memory spikes.

2. Resource Optimization

- Dynamically adjusts data chunking to match memory availability.
- Ensures timely freeing of memory after process completion.
- Minimizes risk of memory leaks with systematic cleanup.

3. Error Handling

- Provides controlled degradation under high load.
- Supplies clear error messages for troubleshooting.
- Tries automatic recovery where possible.

Funded by the European Union

Data Processing Pipeline

A chunk-based approach enables efficient handling of large geospatial datasets:

```
def get_optimal_chunk_size(data_shape):
    """
    Calculates optimal chunk size for data processing based on
system resources.
    Balances performance with memory constraints.

    Parameters:
        data_shape (tuple): Dimensions of the input data array

    Returns:
        tuple: Chunk size dimensions
    """
    total_pixels = np.prod(data_shape)
    target_chunk_size = 64 * 1024 * 1024  # 64MB
    pixels_per_chunk = target_chunk_size // (8 * 3)  # float64 with
3 bands

    chunk_side = int(np.sqrt(pixels_per_chunk))
    return min(chunk_side, data_shape[0]), min(chunk_side,
data_shape[1])
```

Advantages:

1. Adaptive Handling

- Automatically adjusts chunk size based on dataset specifics.
- Allows optimal memory usage for different data varieties.
- Processes inputs of varying sizes seamlessly.

2. Performance Gains

- Enables parallel processing for large datasets.
- Minimizes the memory overhead of big operations.
- Improves throughput by limiting data loaded in memory at once.

3.6.3 API Services and Endpoint Architecture

The UDENE API now offers a broad and modular set of endpoints that address diverse aspects of geospatial data processing, analysis, and machine learning integration. Built with FastAPI and leveraging modern asynchronous programming patterns, the API is organized into clear service categories that cater to system management, data processing, and machine learning functionalities.

Funded by the European Union

3.6.3.1 System Management Services

Health Monitoring Service
Endpoint: **GET /health**
Description: Performs real-time system checks and returns a JSON payload that indicates overall system health, the status of the Data Cube connection, and the total number of available products. This endpoint is optimized for automated health monitoring.

Product Discovery Services
Endpoint: **GET /products**
Description: Retrieves a detailed list of supported satellite products. Beyond just product names, this endpoint returns enriched metadata including short and full descriptions, default coordinate reference systems (CRS), spatial extents (as GeoJSON polygons), and overall bounds. This makes it useful for initial data exploration and for understanding the spatial coverage of each product.

Measurement Discovery Service
Endpoint: **GET /measurements/{product}**
Description: Lists the available measurements (bands) for a specified product. By accepting a product name as a path parameter, it helps users verify which bands or variables are available for further analysis.

3.6.3.2 Data Processing Services

Statistical Analysis Service
Endpoint: **POST /stats/{product}**
Input:

- Product identifier (path parameter)
- Time range (start and end dates provided in the request body)
- Optional list of bands/measurements

Output: Returns computed statistics (minimum, maximum, mean, and standard deviation) for each specified measurement. This service uses chunked processing to efficiently handle large datasets.

Image Composition Service
Endpoint: **POST /rgb/{product}**
Input:

- Product identifier (path parameter)
- Date range (via the request body)

Output: Produces an RGB composite image optimized for web display along with relevant processing metadata. The endpoint employs memory-efficient processing techniques and leverages Dask for handling large image data.

Band Processing Service
Endpoint: **POST /band/{product}/{band}**
Input:

- Product identifier and specific band name (path parameters)
- Time range (provided in the request body)

Funded by the European Union

Output: Returns the processed data for the requested band as well as statistical summaries (min, max, mean, and standard deviation), aiding in preliminary data visualization and analysis.

3.6.3.3 Machine Learning Services

Prediction Service

Endpoint: **POST /predict**

Input:

- JSON payload containing the product identifier, time range, and an optional list of measurements

Output: Provides prediction outcomes (e.g., land surface temperature maps) along with confidence levels and metadata detailing the processing steps. The service dynamically loads the UNet model to perform inference on preprocessed data from the Data Cube.

Similarity Analysis Service

Prediction on Spatial Windows with Dynamic ML Service Integration

Endpoint: `POST /predict_window/{product}`

`Inputs:`

- **`bbox`:** The spatial bounding box for the area of interest in EPSG:4326 coordinates.
- **`time_range`:** The start and end dates defining the temporal window.
- **`measurements`:** List of satellite bands required for processing.
- **`window_params`:** Parameters for extracting a specific window from the dataset (here, a window size of 50 is used, and only the central window is returned by default).
- **Query Parameters (`ml_service_url` and `ml_service_port`):** These allow dynamic specification of the ML service endpoint. In this example, the service is hosted at `http://ml.example.com` on port `8082`.

Description:

This endpoint retrieves a spatial window from the Data Cube, computes relevant spectral indices (such as NDVI, NDBI, NDWI, SAVI, IBI, and EVI) using predefined bands (B02, B03, B04, B08, B11, and B12), and then obtains predictions from an external machine learning service. Notably, the endpoint is designed to be highly dynamic:

- **Dynamic ML Service Configuration:**
  Users can override the default ML service by providing the ML service URL and port as query parameters. This allows the backend to flexibly interact with different ML service providers without requiring code changes. For example, if an alternative ML service is needed for testing or to support a new model version, one simply supplies the new service's URL and port in the request.
- **Flexible Data Processing:**
  The endpoint accepts input in a standard coordinate reference system (EPSG:4326), then internally transforms it to EPSG:32634 to align with the Data Cube's configuration. It then leverages existing utilities to extract windowed data and calculate spectral indices—ensuring that the data sent to the ML service is always appropriately preprocessed.

Funded by the European Union

- **Asynchronous Communication:**
  The service uses asynchronous HTTP calls (via aiohttp) to communicate with the specified ML service endpoint. This non-blocking design enables high throughput and responsiveness, even when multiple windows or large datasets are processed concurrently.
- **Robust Error Handling and Logging:**
  Comprehensive logging at each step—from request normalization and window data extraction to the asynchronous call to the ML service—ensures that any issues are captured and reported. If the ML service returns an error or is unreachable, the endpoint provides a clear HTTPException detailing the issue.

Key Dynamic Aspects:

1. **Configurable Service Endpoint:**
   By accepting `ml_service_url` and `ml_service_port` as optional parameters, this endpoint supports dynamic routing to any compatible ML service. This decouples the API from a single ML service provider and facilitates testing or gradual transitions between services.
2. **Reusable Processing Pipeline:**
   The endpoint reuses common functions for window data extraction and spectral index calculation, ensuring that regardless of the ML service used, the input is consistently prepared. This modular design promotes code reuse and simplifies integration.
3. **Asynchronous and Scalable:**
   Leveraging FastAPI's asynchronous capabilities and aiohttp for external requests, the endpoint can handle high loads and multiple concurrent predictions without performance degradation.

Endpoint: POST /run_ranged_experiments/{product}

Inputs:

- **bbox:** The spatial bounding box for the area of interest in EPSG:4326 coordinates.
- **time_range:** The start and end dates defining the temporal window.
- **window_params:** Parameters for extracting windows from the dataset (e.g., a window size of 50, which defines the spatial extent of each window).
- **PerturbedWindowRequest Fields:**
  - **selected_measurements:** A list of satellite bands (e.g., B02, B03, etc.) that are to be perturbed.
  - **perturbed:** A float value representing the ratio by which the selected bands are modified.
  - **outlier_ratio (optional):** A float (default 0.0) used to filter similar windows based on similarity scores.
- **Query Parameters (ml_service_url and ml_service_port):**
  These parameters allow dynamic specification of the ML service endpoints. For example, a client can specify that the ML service is hosted at http://ml.example.com on port 8082.

Funded by the European Union

Description:

This endpoint is designed to execute multiple experiments over a range of perturbed ratio values. Its dynamic nature makes it adaptable to various ML services and experimental configurations:

- **Flexible Data Processing:**
  The endpoint starts by normalizing the input bounding box and then constructs a window request using a standard set of required bands (B02, B03, B04, B08, B11, and B12). The request model extends the base `SpatialWindowRequest` with additional fields (as defined in the `PerturbedWindowRequest` class):
  1. **selected_measurements:** Determines which bands will have their values perturbed.
  2. **perturbed:** Specifies the multiplication factor applied to the selected bands.
  3. **outlier_ratio:** Sets a threshold used to filter similar windows returned by the ML service.
- After extracting windowed data from the Data Cube, the endpoint iterates over a calculated range of perturbed ratios (from `perturbed_end` to `perturbed_start` with a given step). For each window and each perturbation value:
  1. The selected bands are modified by the specified perturbed ratio.
  2. Spectral indices (such as NDVI, NDBI, NDWI, SAVI, IBI, and EVI) are calculated on the modified data.
- **Caching for Efficiency:**
  The endpoint is decorated with a caching mechanism (`@cache_response(expiration_time=36000)`) to store and quickly retrieve results for identical experiment requests within the specified cache expiration time. This improves performance when experiments are repeated or during iterative analysis.

3.6.3.4 Machine Learning Integration

Model services within UDENE are powered by a dynamic model management subsystem that ensures efficient resource utilization and robust performance. Key aspects include:

Dynamic Resource Allocation:
The system dynamically loads the UNet model and associated embeddings using the load_model_and_embeddings() function. This function selects the appropriate computation device (CPU, GPU, or specialized hardware such as MPS), ensuring that the model is always executed in an optimal environment. Once predictions are complete, resources are freed via unload_model_and_embeddings() to maintain memory efficiency.

Processing Efficiency:
The architecture supports batch processing of input data, which improves prediction speed and throughput. Normalized embeddings are precomputed and utilized for rapid similarity calculations. This approach minimizes computational overhead during similarity search operations.

Flexible Integration:
The ML endpoints are designed to interact seamlessly with both the internal data processing pipeline and external ML services (via asynchronous HTTP calls). This integration allows for

Funded by the European Union

real-time predictions, outlier detection, and similarity analysis, making the system highly adaptive to various urban development use cases.

Caching and Robustness:
A custom caching mechanism is implemented to store and quickly retrieve responses for frequently executed queries. This further enhances the performance of the ML services, particularly when processing large volumes of geospatial data.

3.7 Frontend

The UDENE exploration and matchmaking tool frontend is designed to handle multiple simultaneous requests with multiple sessions running concurrently to offer each user a custom one of one experience. This section delves into the chosen implementation strategies, stack used and elaborates on the key reasoning behind major architectural decisions.

3.7.1 Infrastructure Context and Constraints

The current deployment runs on a local virtual machine under VMware ESXSI (4 vCPU, 16 GiB RAM) with Ubuntu Server. This configuration will be sufficient even for the production version. However it is designed to be Scalable at will**.** The frontend is running under a docker container. Therefore the migration to a more robust server or demanding installation is at will.

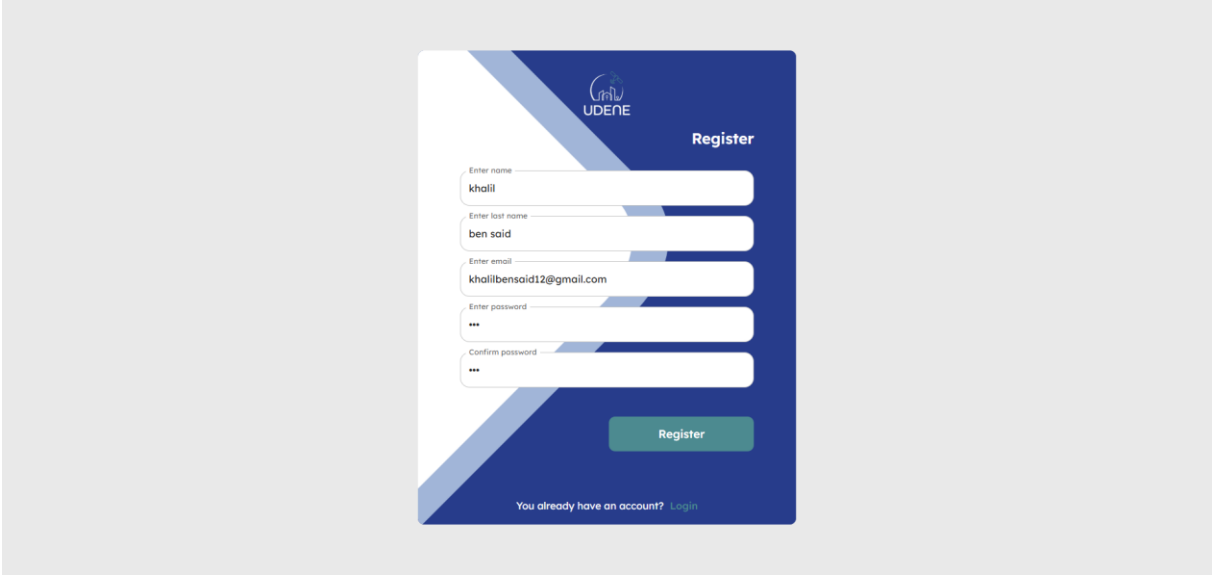
3.7.2 Framework Selection and Rationale

React ts was chosen as the primary framework based on the following considerations:

1. **Performance :**
    - The virtual DOM optimizes rendering performance by updating only the necessary parts of the UI. This is crucial for handling user interaction with sub page components.
    - A powerful State Management System. which enables us to store and share information across the application without returning to the backend for temporal use.
    - Typescript was used as strong error boundary
    - Reduces the number of event binding and focuses on action driven events.
    - Optimized css and styling methods to reduce loading overhead.
2. **Development Efficiency**
    - Component Reusability and Modularity
    - Utilize modern tools like Create React App for faster builds and seamless developer experiences.
    - Large community Support and high pre-built packages and functions.
    - Efficient Debugging and Testing due to open source custom DevTools
3. **Integration**
    - Supports client and server side rendering
    - Javascript based framework therefork fully compatible with leaflet-js and other javascript based maps
    - API consume based architecture.

Funded by the European Union

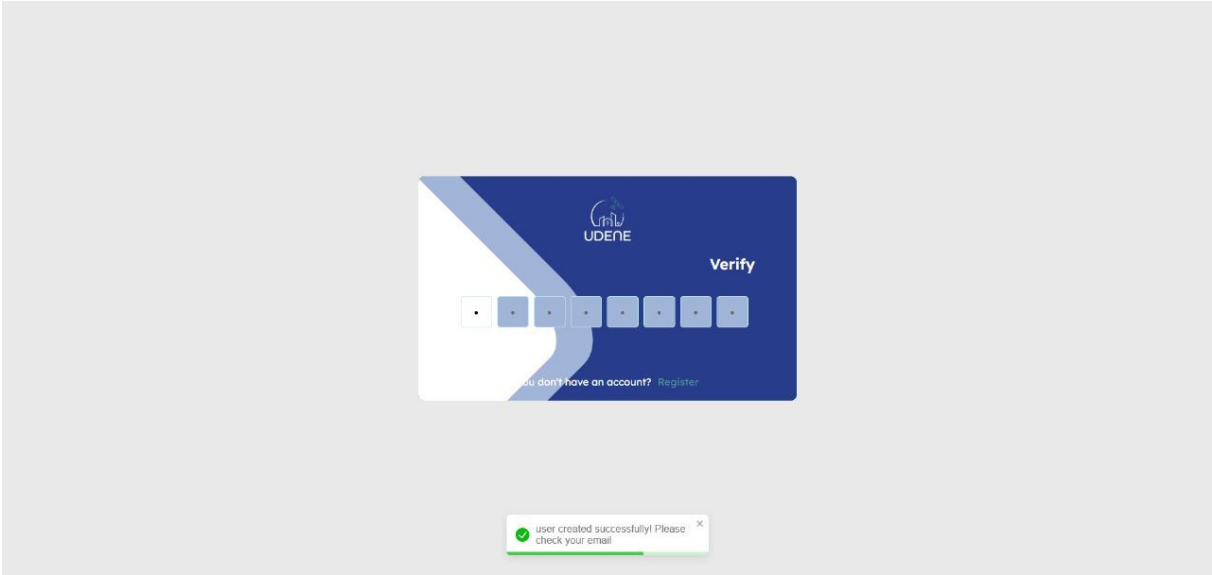
3.7.3 The Exploration and Matchmaking Tools Access via Account Creation



The registration page allows new users to create an account by providing essential credentials. It includes 4 main fields

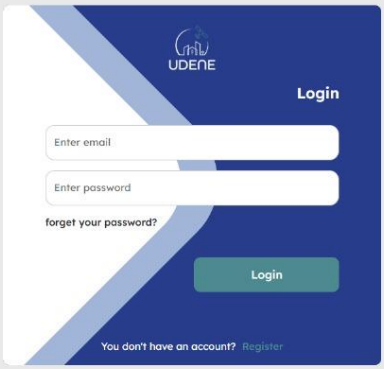
- Email: the user email that will be used to communicate with the user.
- Password: a secure password that must be at least 8 length and includes uppercase, lowercase , number and special characters
- Name
- Last name

After filling in the required fields, clicking the "Sign Up" button. The system will send an email to the user containing a verification code and redirect the user to the verification form.

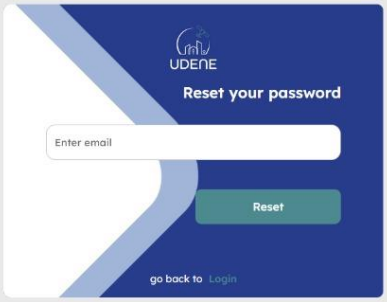


Upon entering the code to confirm his email address the system will grant the user access to the application, ensuring a secure and seamless onboarding experience.

Funded by the European Union



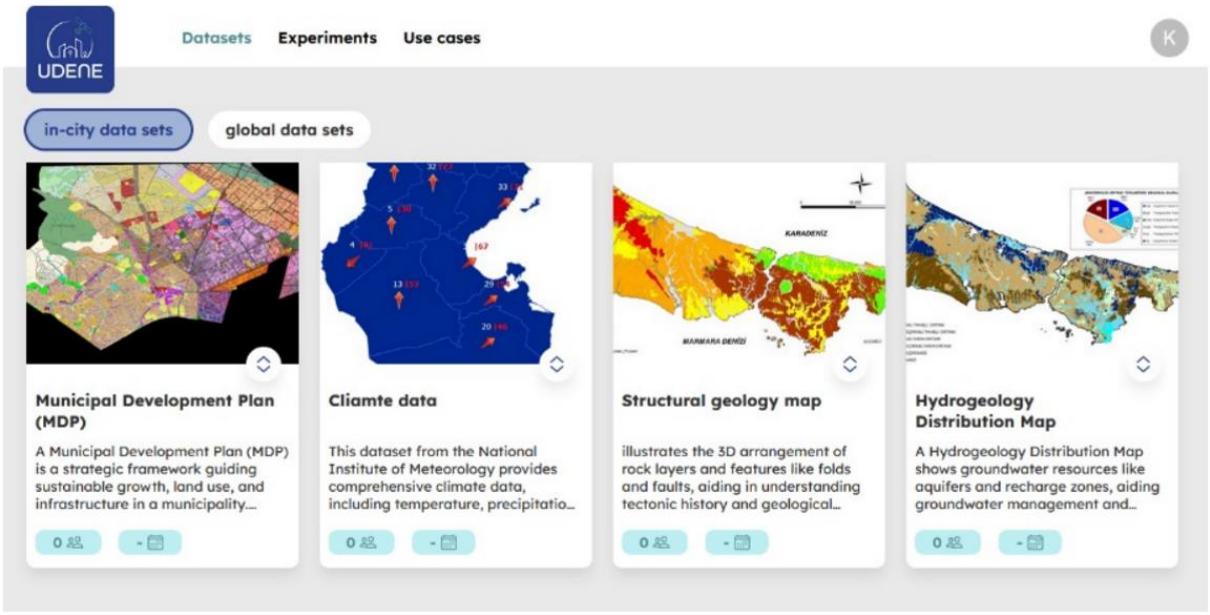
When you launch the application, simply enter your **email** and **password**, then click **Sign In** to access your account. This straightforward process ensures quick and secure login.



If you've forgotten your password, simply enter your **email address** on the password recovery page. You'll receive an email with a link to reset your password. Follow the instructions in the email to create a new password and regain access to your account. This secure process ensures you can quickly recover your account and continue using the application.

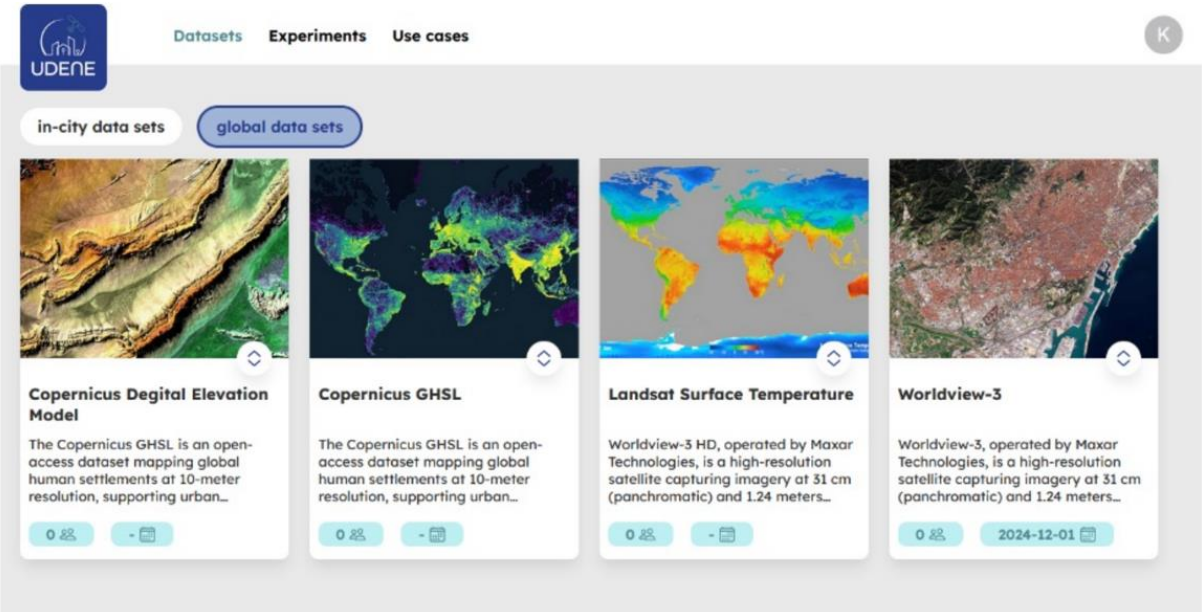
Funded by the European Union

3.7.4 The Datasets Tab



When you click the Dataset button in our application, the page provides access to two distinct categories of data. The first category focuses on in-situ datasets, which are datasets collected from ground-based observations or measurements. These datasets are designed to be fully interactive and user-friendly, allowing users to visualize the data directly within the application. Additionally, users can download the datasets for offline analysis or seamlessly integrate them into the Experiments Tool for processing and experimentation.

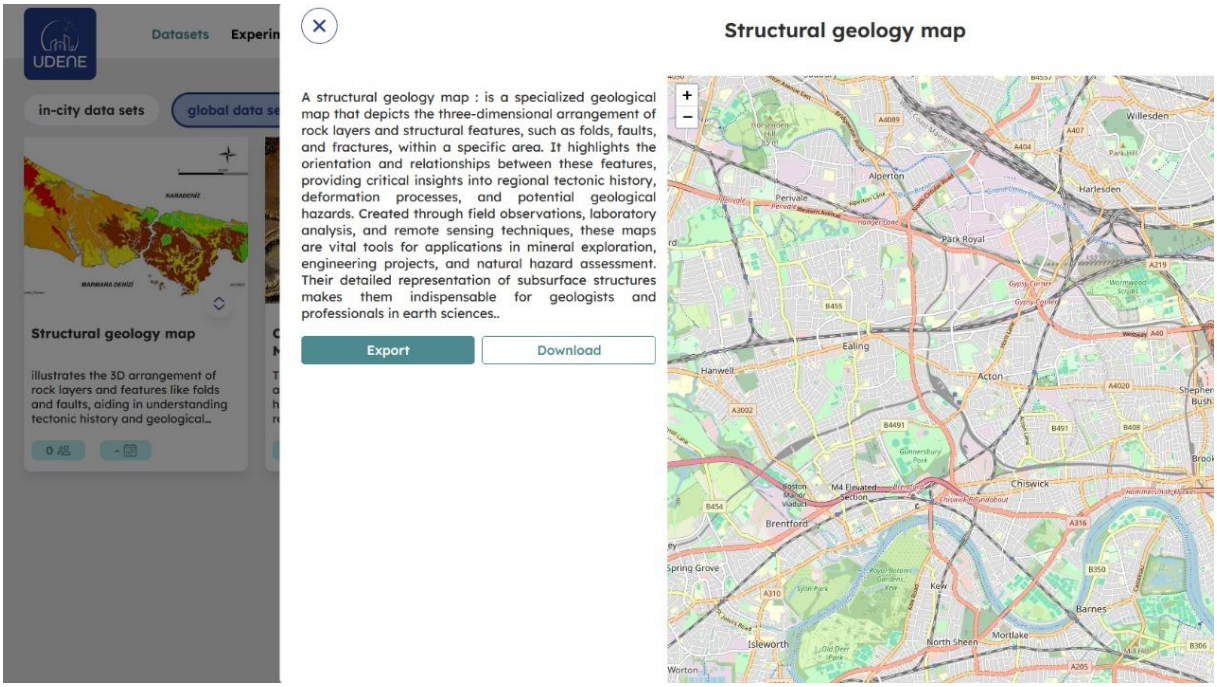
Each dataset is enriched with comprehensive metadata to ensure clarity and usability. This includes a detailed description of the dataset's purpose and contents, the number of users, contributors involved in its creation or curation, the exact date of acquisition, and any additional contextual information that might enhance the user's understanding of the dataset.



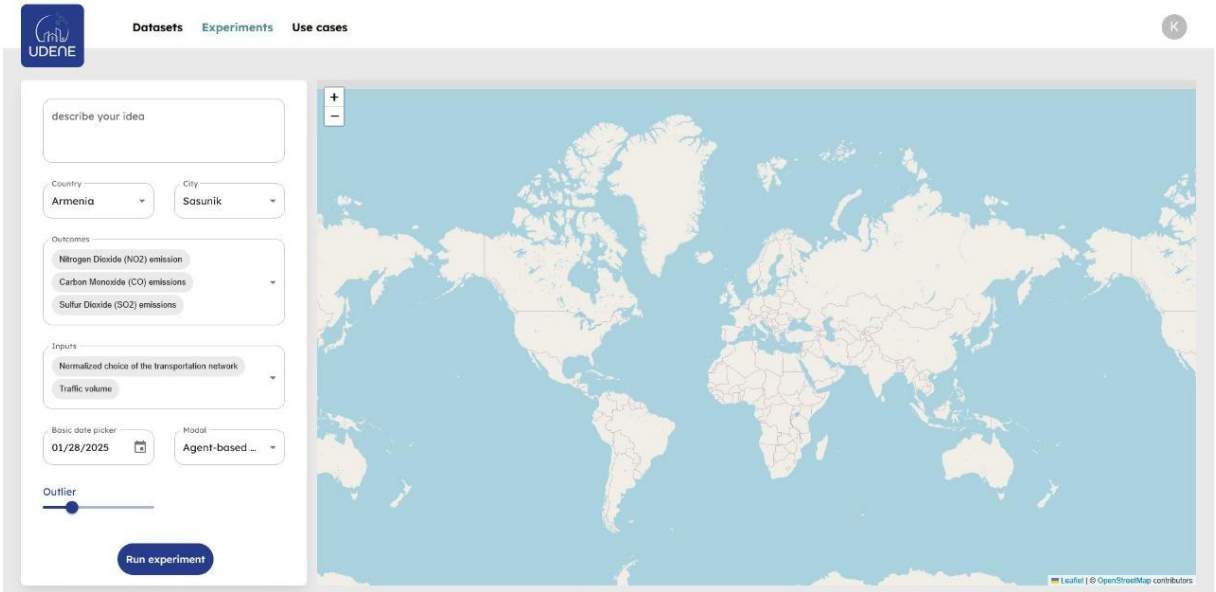
The second category is global datasets, which represent Earth observation datasets that are globally available. These datasets provide a broader perspective, leveraging satellite and remote sensing technologies to cover global-scale phenomena. Like the in-situ datasets, global datasets are designed to be fully interactive and user-friendly, enabling users to

Funded by the European Union

visualize the data directly within the application. Users can also download these datasets for offline analysis or integrate them seamlessly into the Experiments Tool for advanced processing and experimentation. Each global dataset is accompanied by comprehensive metadata, including a detailed description of the dataset's purpose and contents, the number of users, the date of acquisition, and additional contextual information to enhance understanding and usability.



When you click on any dataset, a new page will open featuring a detailed description, two buttons for exporting or downloading the data, and an interactive map that allows you to zoom in and out for closer inspection. This user-friendly interface ensures seamless access and exploration of the dataset.

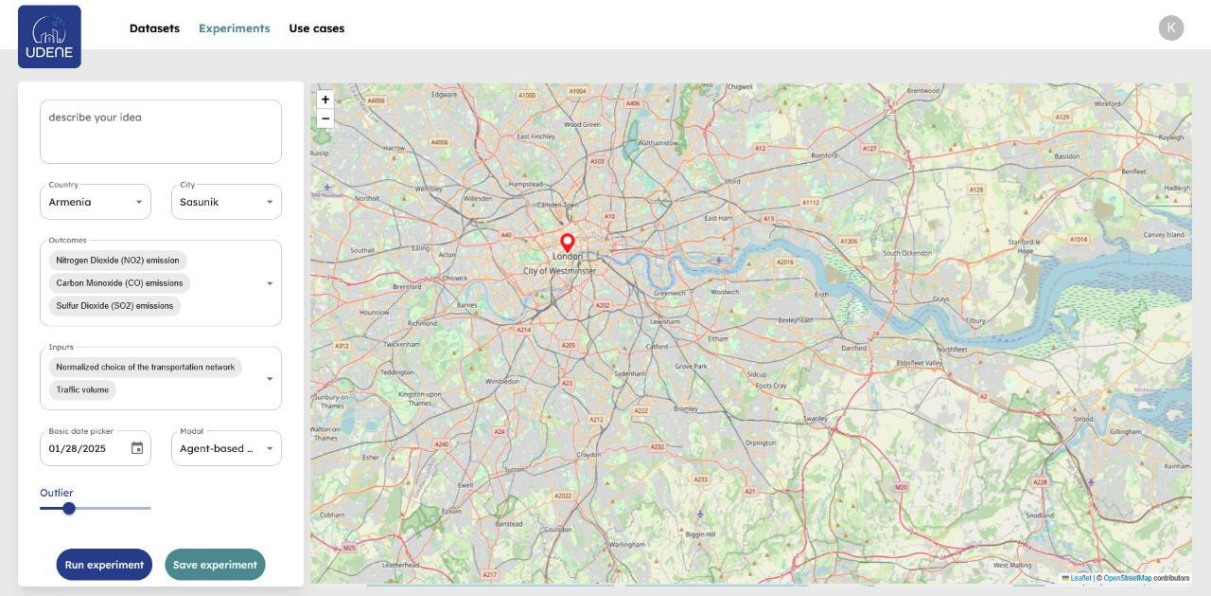


Funded by the European Union

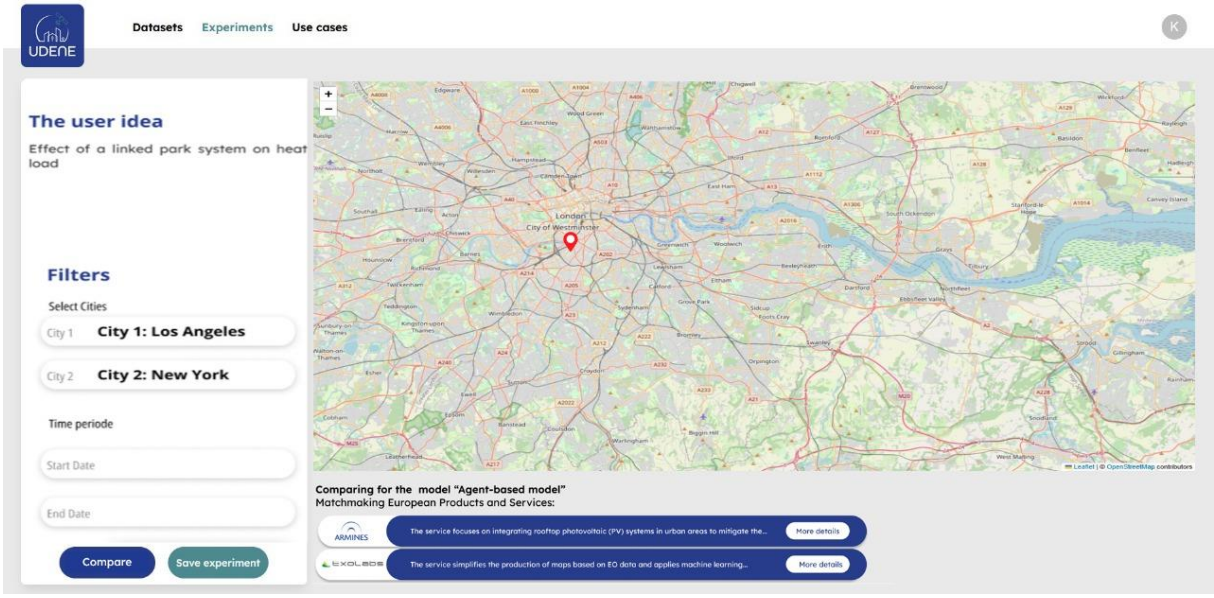
3.7.5 The Experiment Tab

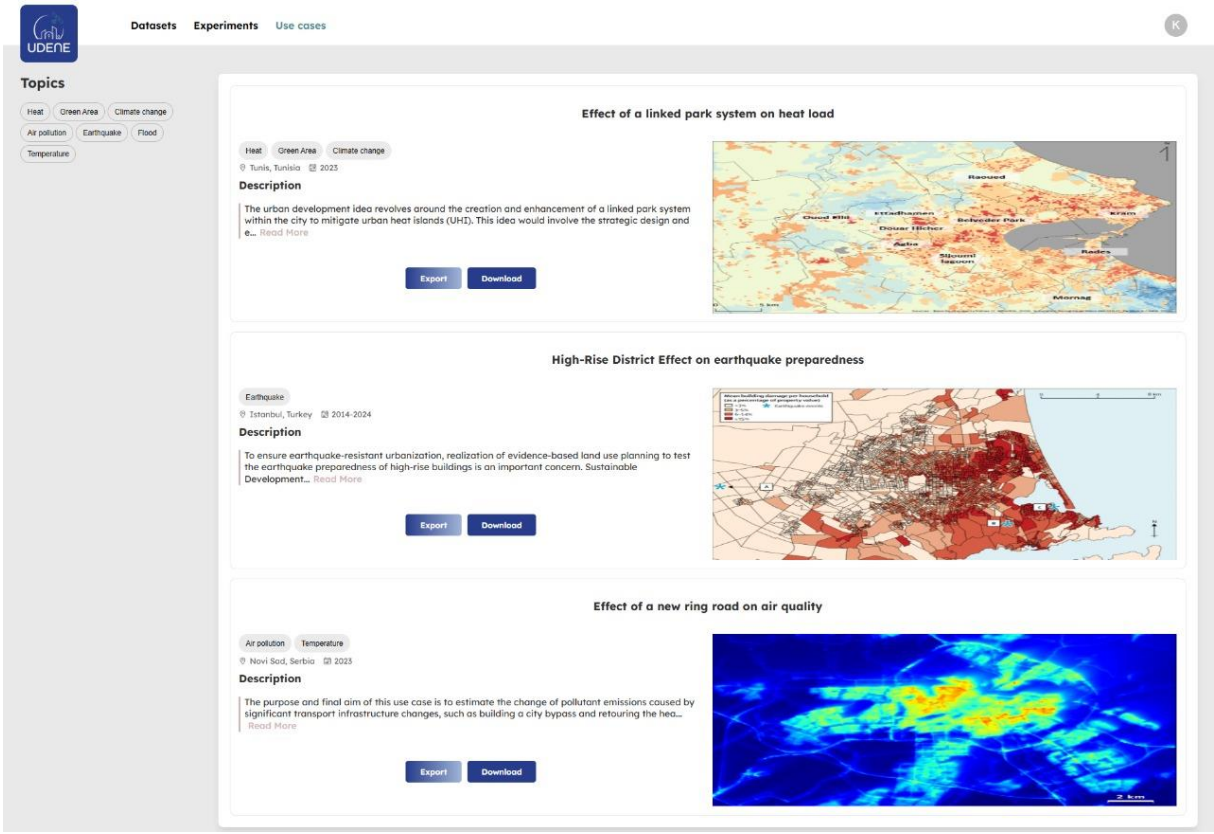
The Experiment tab is the dedicated space where users can test and evaluate their urban development ideas in an interactive and structured manner. The process is designed to guide users step-by-step, ensuring clarity and usability. Here's how it works:

- **Describe the Development Idea:** At the top-left of the page, users begin by providing a detailed description of their urban development idea in a text box. This helps contextualize the experiment and document the purpose of their test.
- **Select Area of Interest:** Users then choose the country and city of their area of interest. This is accompanied by the option to specify the time period for the experiment, ensuring that the analysis is tailored to a specific temporal and spatial context.
- **Define Variables:**
  Outcome Variables: Users select the variables they want to measure as the result of the experiment from a pre-prepared list of outcome variables.
  Input Variables: Users choose one or more input variables from another predefined list. These inputs represent the factors influencing the outcome variables.
- **Choose a Model:** From a list of available models, users select the one most suitable for their experiment. This allows for flexibility in testing different modeling approaches. Additionally, users specify the percentage of outliers to account for in the analysis.
- **Run the Experiment:** Once all parameters are set, users click the Run Experiment button to execute the analysis. The system processes the selected inputs and outputs, applying the chosen model to generate results.



Funded by the European Union





3.7.6 The Use Cases Tab

The Use Case page serves as a centralized hub where users can explore a list of use cases, organized by topics, to gain insights and inspiration for their own projects. The layout and functionality are designed for ease of navigation and usability. Here's how it works:

Topics List: On the left side of the page, users will find a categorized list of topics. This enables them to filter and browse use cases that align with their specific area of interest, ensuring a focused and streamlined exploration by clicking on the topic. The selected topic will be colored and highlighted to indicate which topics are being considered in the listing

Funded by the European Union

Use Case List: The main section of the page displays the list of use cases relevant to the selected topic. Each use case is accompanied by a comprehensive description, which includes:

- The date of acquisition of the data or results.
- The city or location associated with the use case.
- The relevant topics covered by the use case.
- Any additional contextual information that enhances the user's understanding of the use case and its potential application.

Export and Download: Users can export or download the use cases for offline use, allowing for further analysis, sharing, or integration into their own projects.

3.8 Demonstration: A Natural Experiment for Urban Cooling

3.8.1 Overview

In this demonstration, we showcase how our Operational Exploration Tool can support data-driven decision-making in urban development. The experiment simulates a real-world scenario where a landscape architect from Bakırköy—a bustling district in Istanbul—evaluates a proposal to build a park at the closed Atatürk Airport. The goal is to assess whether increasing green space can effectively cool down the area, a critical insight for urban planners facing the challenges of urban heat islands.

Experiment Scenario

Persona:
The Landscape Architect from Bakırköy Municipality

- **Challenge:** Bakırköy, with its dense urban fabric and coastal influences, experiences high temperatures in summer.
- **Idea:** Build a park to increase green areas, thereby reducing the local Land Surface Temperature (LST).
- **Assumption:** The creation of a park leads to enhanced vegetation (captured by indices such as EVI and NDVI), which in turn lowers LST.

3.8.2 Methodology

1. **Data Acquisition & Preprocessing**
    - **Data Source:** Retrieve Sentinel-2 satellite images covering Bakırköy for the entire year 2024.
    - **Preprocessing Steps:**
        - Align and crop the images to focus on the area around the closed Atatürk Airport.
        - Compute key geospatial indices that characterize the urban environment:
            - **EVI (Enhanced Vegetation Index)**
            - **NDVI (Normalized Difference Vegetation Index)**
            - **NDWI (Normalized Difference Water Index)**
            - **SAVI (Soil-Adjusted Vegetation Index)**
            - **IBI (Index-Based Built-up Index)**
            - **NDBI (Normalized Difference Built-up Index)**

Funded by the European Union

        - Normalize the data to prepare it for model input.

2. **Model Input & Prediction**
    - **Input Variables:** The computed indices, which represent the urban properties relevant to the park proposal.
    - **Outcome Variable:** LST (Land Surface Temperature) is used as a proxy for the "coolness" effect.
    - **Model Framework:**
        - A regression-based model, with the UNET architecture approximating the regression for faster, high-resolution segmentation.
        - **Why UNET?**
            - It offers high-resolution segmentation, crucial for identifying small-scale urban structures.
            - Skip connections in UNET preserve fine spatial details, enabling precise mapping of temperature gradients.
3. **Outlier Detection & Perturbation Testing**
    - **First Stage:**
        - Identify regions similar to Bakırköy based on the selected indices. This step reduces the search space by focusing on areas that share comparable urban characteristics.
    - **Second Stage:**
        - Among the similar regions, pinpoint "outlier" areas where the predicted LST is significantly lower than what the model would normally expect.
        - **Purpose:** Such outliers may indicate naturally occurring cooling effects—possibly due to existing green spaces—that validate the landscape architect's hypothesis.
    - **Additional Research Parameters:**
        - *Outlier Detection* flags unusual temperature readings.
        - *Perturbation Testing* introduces small variations in input values to assess the robustness of model predictions.
4. **Experiment Execution & Evaluation**
    - **Running the Experiment:**
        - Execute the workflow, feed the preprocessed inputs into the UNET model.
        - Visualize the results on a map to highlight both similar regions and the identified outliers.
    - **Evaluation Metrics:**
        - Use validation metrics such as Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) to assess model accuracy.
        - Analyze the spatial distribution of LST to correlate areas of increased green space with lower temperatures.

3.8.3 Expected Outcomes

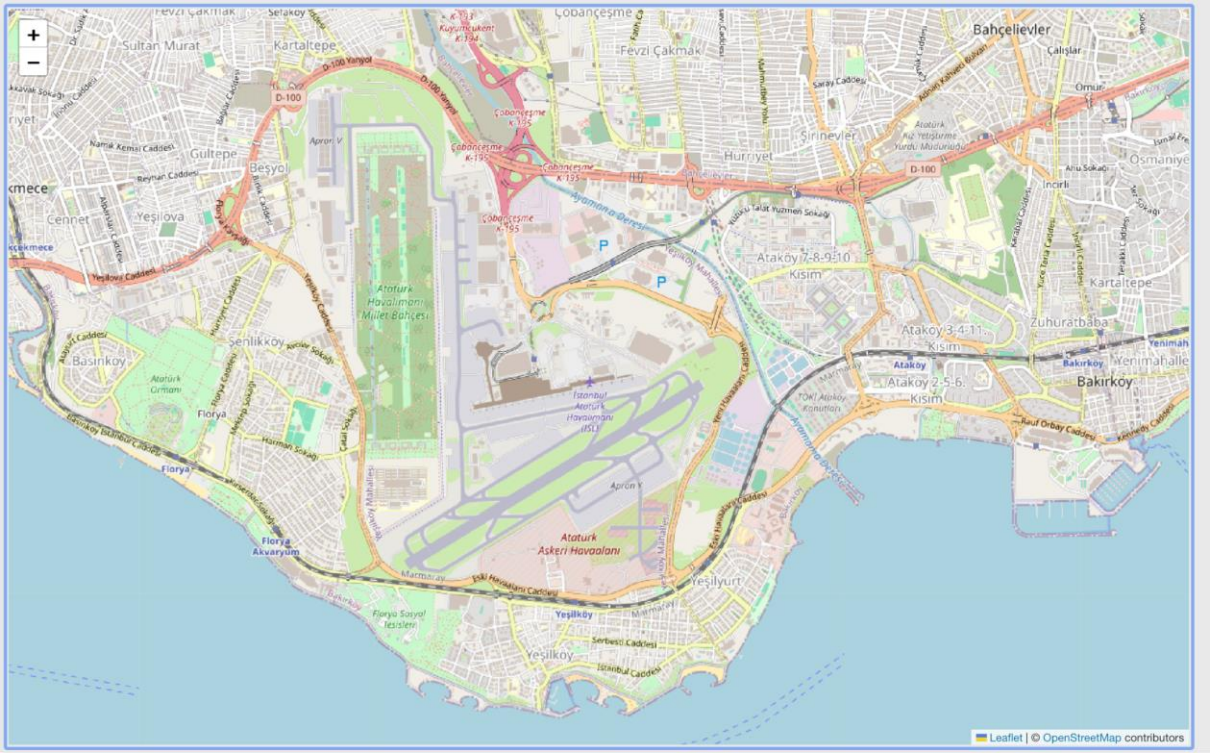
- **Urban Heat Island Effects:**
  Regions in Istanbul with dense construction and sparse vegetation show higher LST, validating the need for intervention.
- **Vegetation Impact:**
  Areas with higher EVI or SAVI values exhibit lower LST, suggesting that increased green space (as would be created by a new park) could lower temperatures.

Funded by the European Union

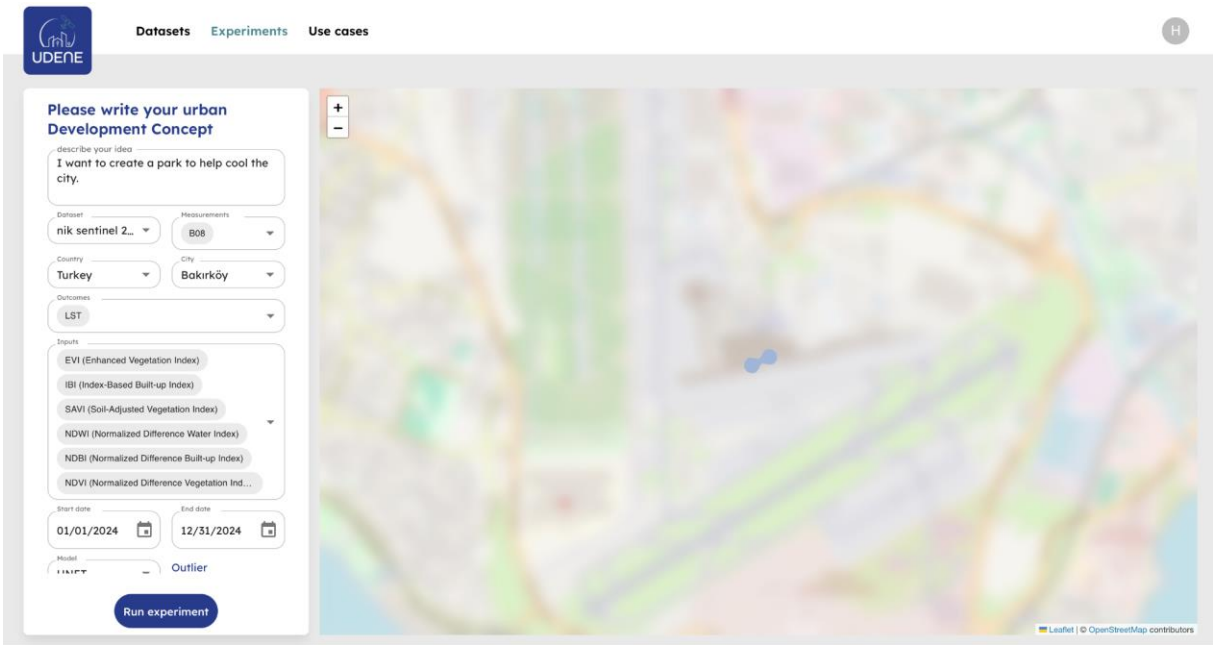
- **Role of Water Bodies:**
  NDWI analyses indicate that proximity to water correlates with cooler surface temperatures.
- **Anomalies and Natural Experiments:**
  The model's outlier detection has surfaced natural experiments both in Istanbul and in San Francisco/California, confirming that even in dense urban areas, localized cooling can occur under the right conditions. For Istanbul, these findings support the landscape architect's proposal to develop a park that leverages similar cooling mechanisms.

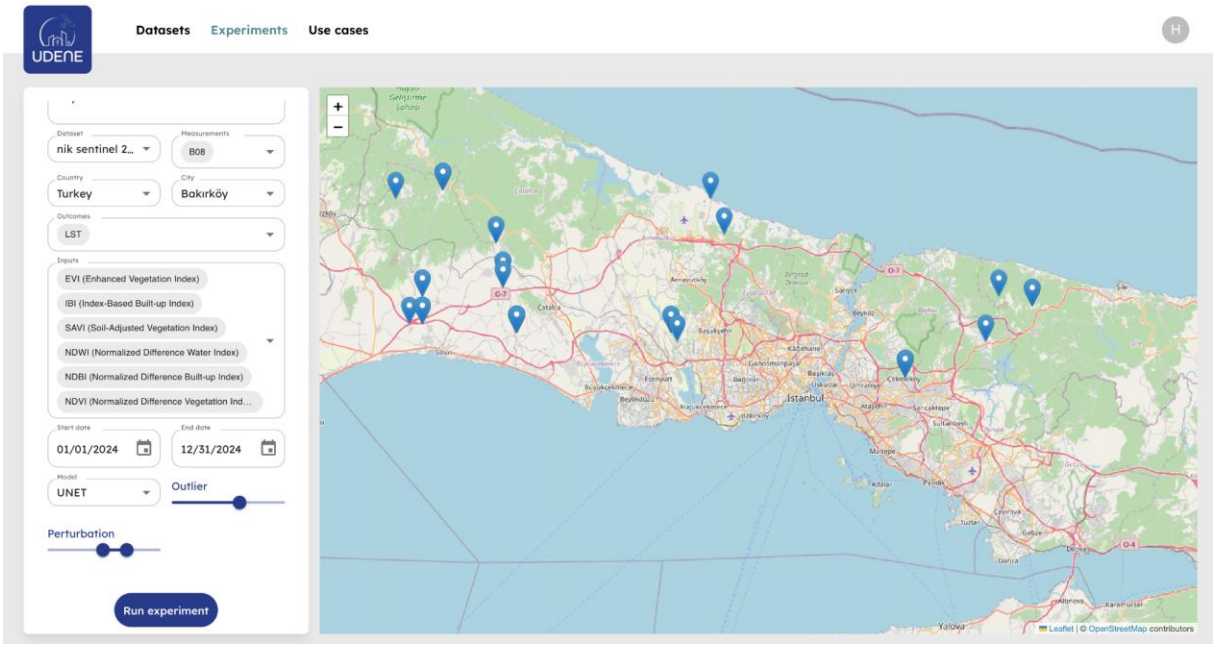
3.8.4 Integration with the Application

- **User Workflow:** A step-by-step guide where the landscape architect:
  - Describes the urban development idea.
  - Selects Bakırköy as the area of interest.
  - Configures input indices and runs the experiment.
  - Reviews the visual outputs and interprets the results to decide on the feasibility of the park project.

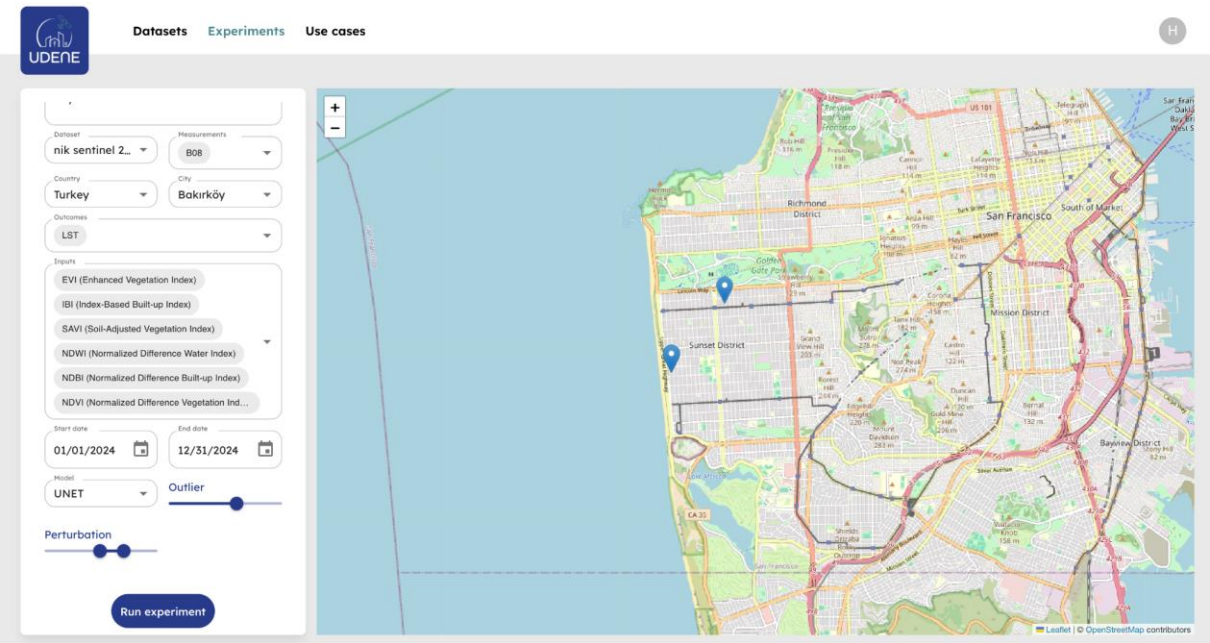


Funded by the European Union





Funded by the European Union



3.8.5 Conclusion

This demonstration effectively illustrates how our tool leverages satellite imagery, deep learning (via UNET), and advanced analytics to transform raw Earth Observation (EO) data into actionable insights for urban planning. By identifying natural experiments and outlier regions, urban developers are empowered to make evidence-based decisions—ultimately fostering sustainable urban development.

Funded by the European Union

4. Example Queries

The similarity finding endpoint in your backend system serves as a critical component for identifying locations with similar characteristics based on multi-band remote sensing data. This endpoint processes an input **.npy** file containing several geospatial indices and compares it against pre-computed embeddings to return the most similar locations. These embeddings are stored in a DataCube hosted on AWS, which allows efficient querying across cached embeddings from various parts of the world.

The similarity search leverages a cached embedding strategy, where embeddings for the most common regions are precomputed and stored in the DataCube. This approach ensures faster and more scalable similarity searches across large datasets. The endpoint retrieves embeddings from the DataCube and performs similarity calculations to find the best matching locations worldwide.

4.1 /find_similar Endpoint Workflow and Process

The /find_similar endpoint follows a structured process that involves multiple steps, from receiving the input file to returning the most similar locations. Below is a detailed explanation of each step.

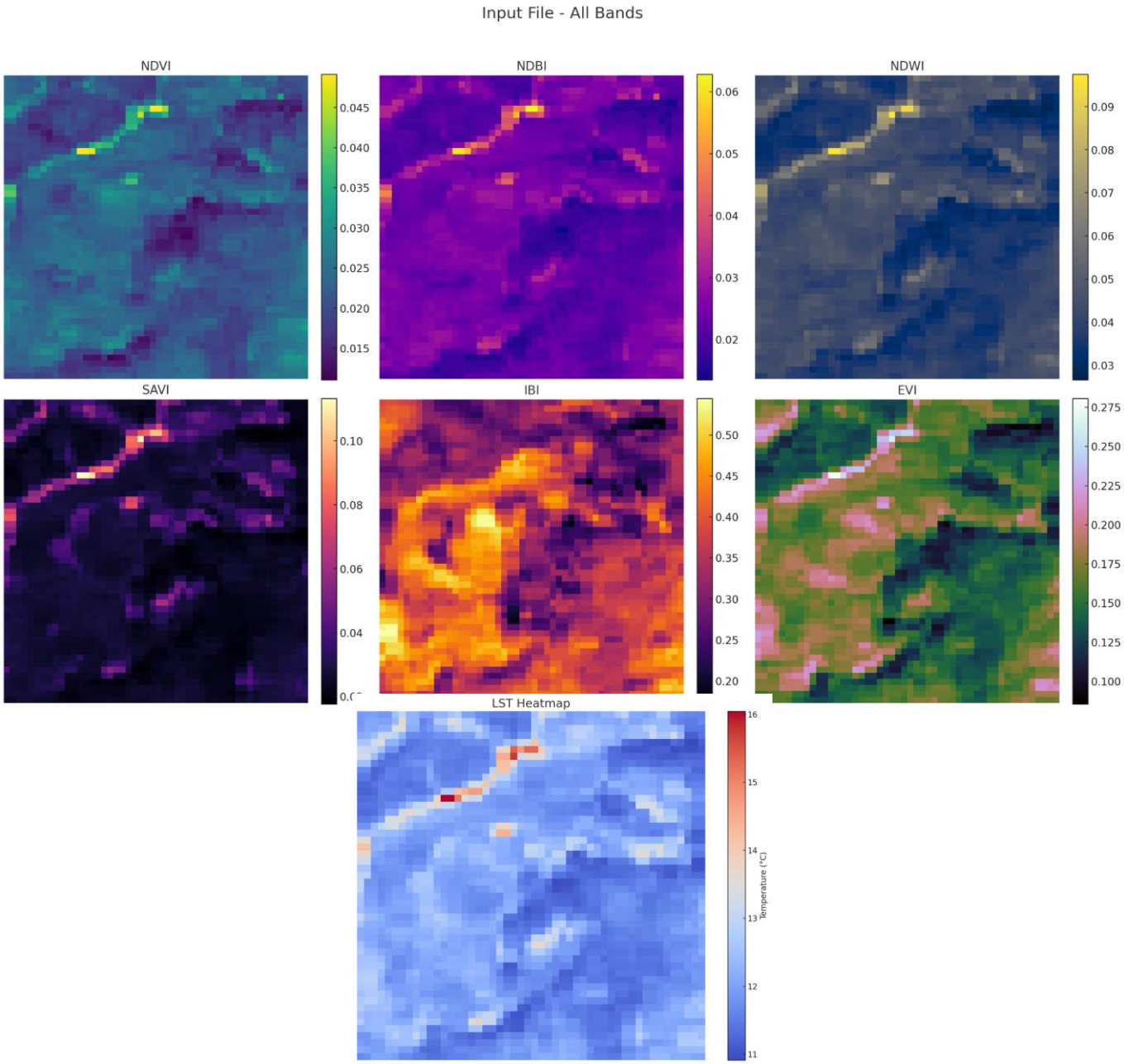
1. **Input File**

The process begins with an input .npy file that contains six geospatial indices and a corresponding **Land Surface Temperature (LST)** heatmap. The input file represents a specific location on Earth, encoded as a multi-dimensional array.

Band Name	Description
NDVI	Normalized Difference Vegetation Index - Indicates vegetation density
NDBI	Normalized Difference Built-up Index - Measures built-up areas
NDWI	Normalized Difference Water Index - Highlights water bodies
SAVI	Soil-Adjusted Vegetation Index - Accounts for soil brightness in vegetation detection
IBI	Index-Based Built-up Index - Enhances built-up area detection
EVI	Enhanced Vegetation Index - Improves vegetation monitoring, especially in dense areas

Funded by the European Union

The input file also includes a **Land Surface Temperature (LST)** map, which represents the temperature of the Earth's surface at the given location.



2. Embedding Calculation and DataCube Retrieval

The input file is processed by a pre-trained deep learning model to compute its **embedding**. An embedding is a compact, high-dimensional vector representation that captures the unique features of the input file. Instead of comparing this embedding with all possible locations globally, the system queries the **AWS-hosted DataCube** to retrieve relevant cached embeddings.

The **DataCube** acts as a centralized storage system that holds embeddings for the most common parts of the world. These cached embeddings are precomputed and updated periodically to ensure that the system remains efficient and accurate.

Funded by the European Union

Steps in Embedding Retrieval:

1. The system queries the DataCube to find the **best matching region** based on the input embedding.
2. Once the relevant region is identified, the system retrieves the corresponding cached embeddings from the DataCube.
3. The input embedding is compared against the cached embeddings to find the most similar and dissimilar locations within that region.

3. Similarity Calculation

To identify similar locations, the endpoint computes the cosine similarity between the input file's embedding and the cached embeddings retrieved from the DataCube. These cached embeddings represent precomputed, high-dimensional representations of popular cities and places stored in our AWS storage. They encapsulate various semantic and geographic characteristics derived from historical data and user interactions. By pre-calculating embeddings for the most requested locations worldwide, we can quickly and efficiently compare the input embedding against these representations. This not only speeds up the process by reducing real-time computation but also leverages detailed, consistent feature sets for each location, ensuring that the similarity comparisons are both fast and meaningful.

In addition to using cached embeddings for popular locations, our system is designed to calculate embeddings in real time for places that aren't already stored. This functionality is currently under development, and once fully implemented, it will further enhance our ability to handle unique or less-common queries efficiently.

Cosine similarity is used because it provides a robust, efficient, and scale-invariant measure of similarity between high-dimensional embeddings. In more detail:

- **Direction over Magnitude:** Cosine similarity focuses solely on the angle between vectors, which means that even if the embeddings have different magnitudes due to varying data scales or normalization, the similarity measurement remains consistent. This is crucial for tasks like comparing semantic content, where the direction of the embedding (i.e., the pattern of values) is more significant than its overall length.
- **Robustness in High Dimensions:** In high-dimensional spaces, traditional distance metrics (like Euclidean distance) can become less informative due to the curse of dimensionality. Cosine similarity, by measuring orientation rather than distance, tends to perform better in these settings.
- **Computational Efficiency:** Calculating cosine similarity is relatively inexpensive compared to other methods, which is advantageous when comparing a large number of embeddings quickly, such as in real-time location similarity searches.
- **Wide Adoption in Machine Learning:** It is a standard choice in many fields including natural language processing, recommendation systems, and clustering tasks, where it has consistently shown to capture meaningful relationships between data points.

By computing the cosine similarity between the input file's embedding and cached embeddings, the endpoint can effectively identify similar locations based on the directional alignment of their vector representations, regardless of the differences in scale.

Cosine Similarity Formula:

Funded by the European Union

$$\text{Cosine Similarity} = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|}$$

Where:

- **A** and **B** are the embedded vectors whose dot product is taken
- ‖**A**‖ and ‖**B**‖ are their magnitudes

The endpoint then ranks the locations based on their similarity scores and returns the **top N similar files**.

4. Results

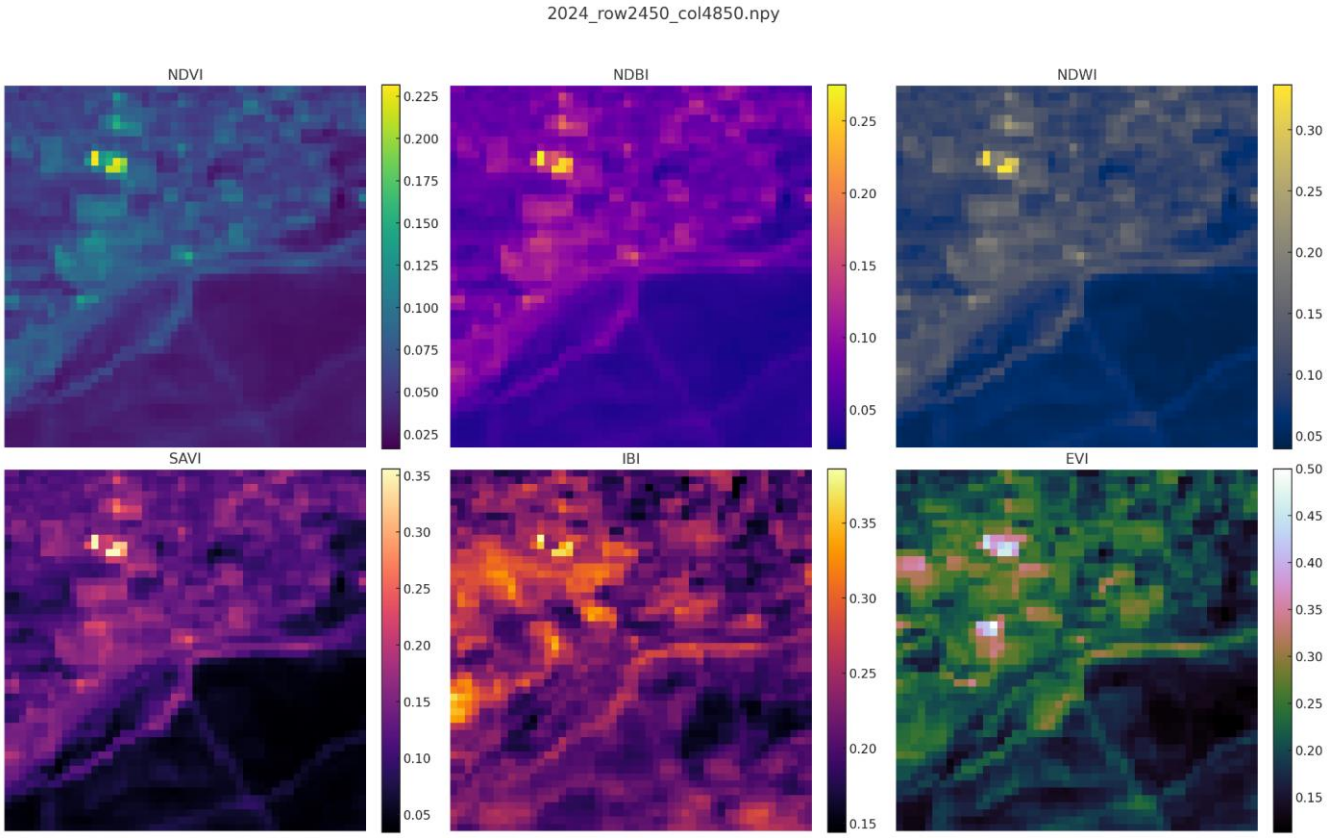
The endpoint returns a list of the **most similar locations**, along with their file paths and similarity scores. Additionally, the endpoint can return the **most dissimilar locations** by sorting the similarity scores in ascending order.


```
{
    "input_file": "2013_row50_col450.npy",
    "similar_files": [
        "2013_row1400_col4050.npy",
        "2014_row1400_col5600.npy",
        "2014_row1600_col6250.npy"

    ],
    "dissimilar_files": [
        "2017_row950_col2700.npy",
        "2017_row1000_col2700.npy",
        "2017_row900_col2700.npy"
    ]
}
```

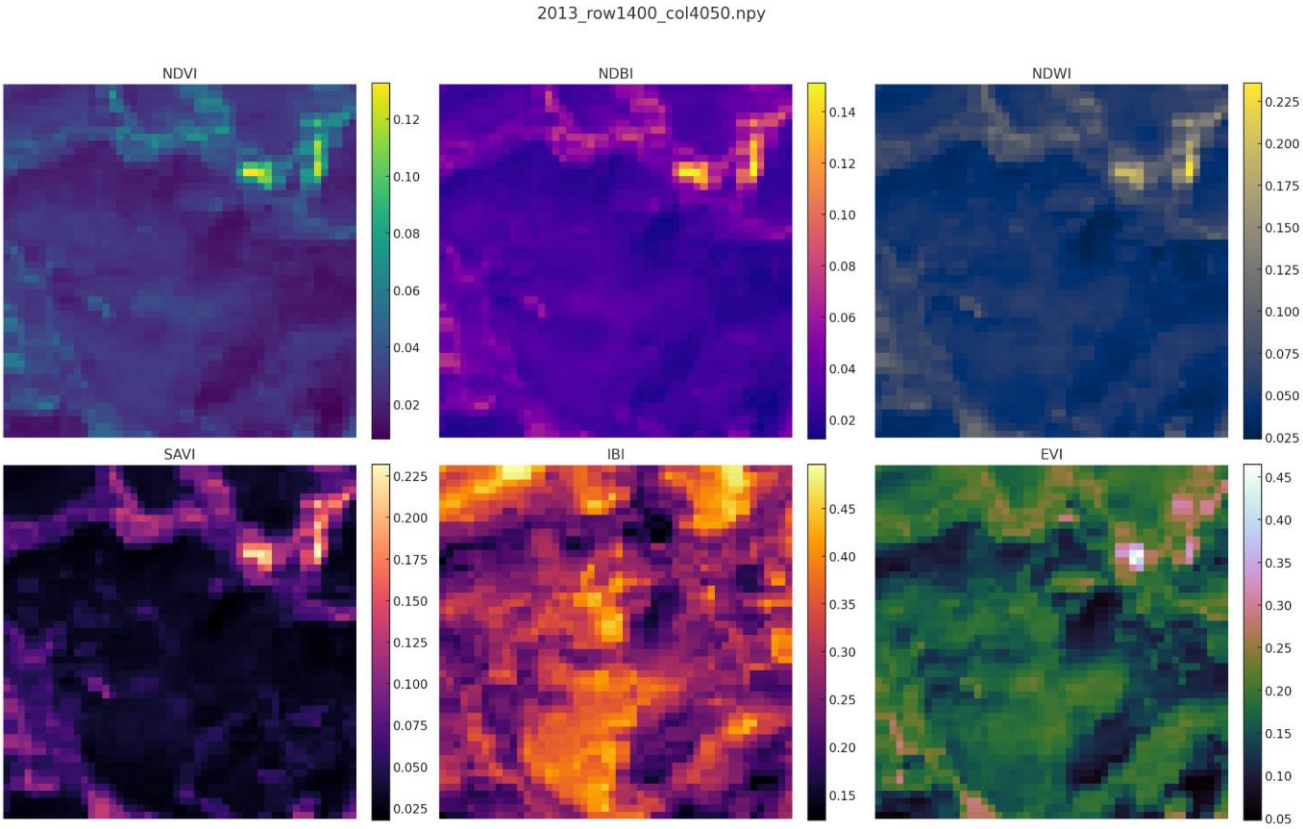

Funded by the European Union

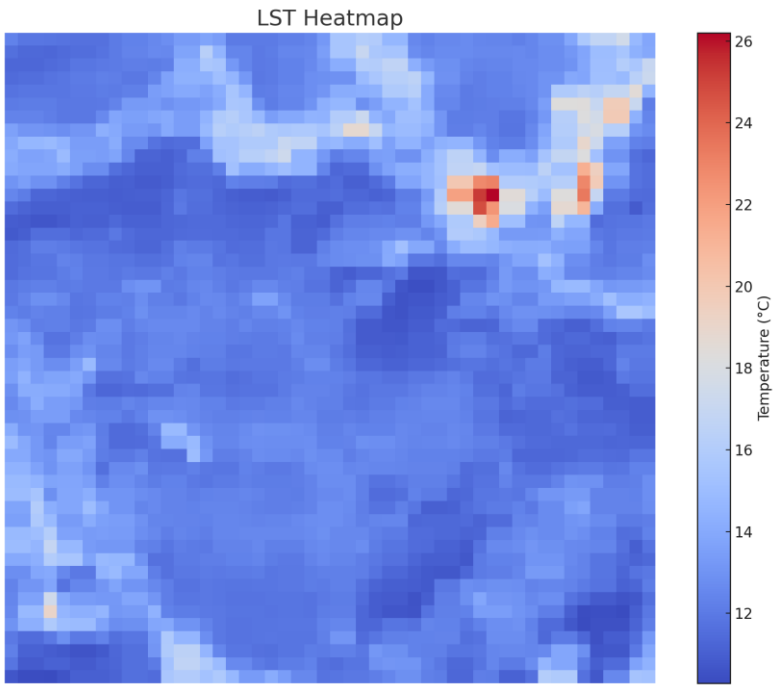
Example Output:



Funded by the European Union

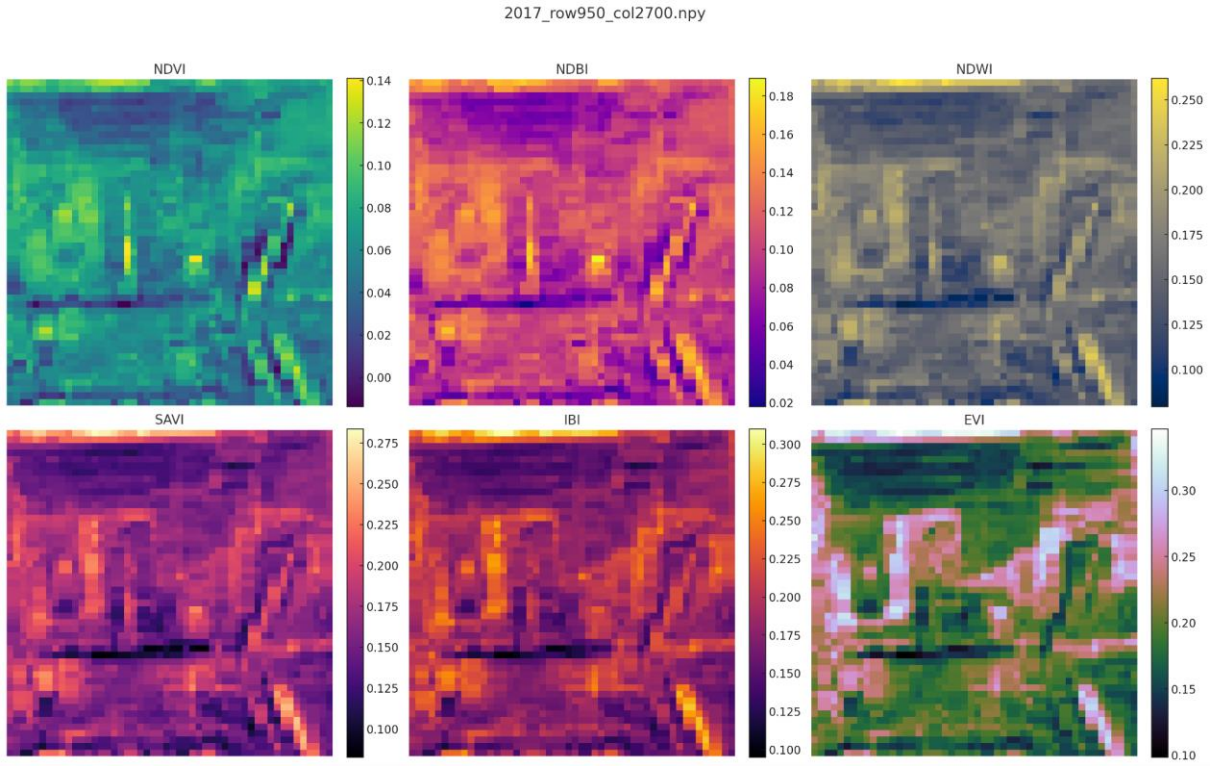
The below figure presents the output from a similarity analysis endpoint, categorizing datasets into similar and dissimilar groups based on computed similarity scores. For instance, the dataset “2013_row1400_col4050.npy” is identified as a similar match. It contains detailed geospatial indices that characterize the location’s features—such as vegetation, built-up areas, and water presence. These indices offer quantitative insights into the similarities between locations, empowering urban planners and researchers to quickly identify areas with analogous characteristics for further analysis and decision-making.

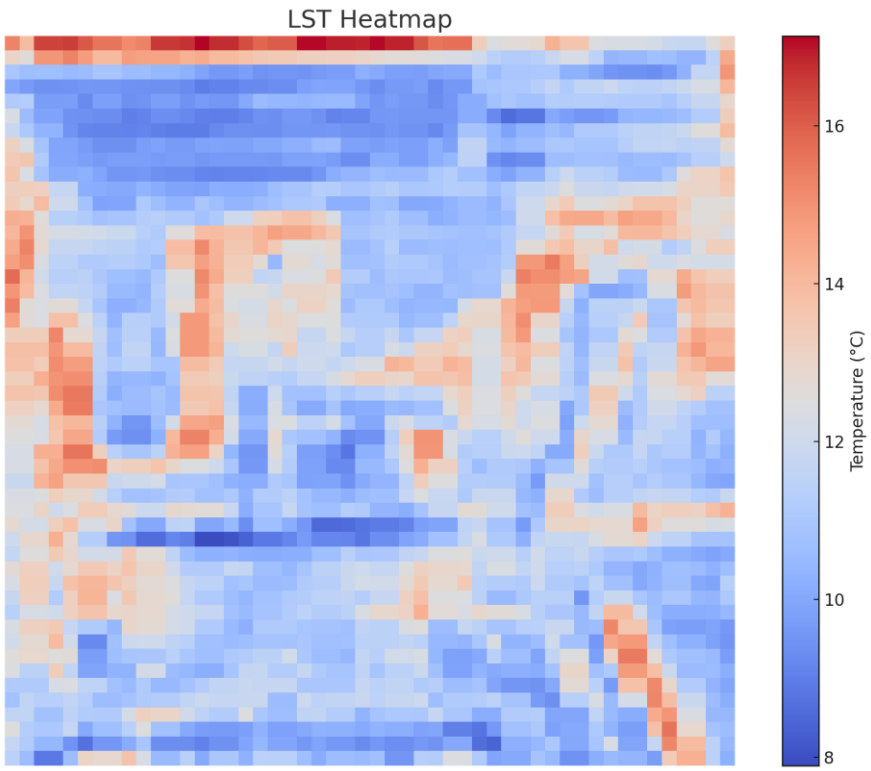




Funded by the European Union

The example below illustrates a dissimilar output. The dataset “2017_row950_col2700.npy” represents a location with geospatial indices that significantly diverge from the reference, highlighting its distinct characteristics compared to similar areas.





Funded by the European Union

4.2 /predict-and-find-similar Endpoint Workflow and Process

The primary goal of this endpoint is to provide an **interactive and dynamic tool** for urban planners, environmental scientists, and decision-makers. It allows users to:

1. **Explore Geospatial Scenarios**: Modify input band values to simulate hypothetical changes in vegetation, urbanization, water presence, and more.
2. **Observe LST Impact**: See how changes in the input indices affect the predicted **Land Surface Temperature (LST)**.
3. **Find Geospatial Matches**: Identify locations with similar or contrasting characteristics under the modified input scenario.

By combining LST prediction and similarity search, this endpoint enables planners to understand the potential outcomes of urban development or environmental changes and locate similar areas to inform decision-making.

How It Works

1. **Interactive Input**: Users can tweak values in the six indices (**NDVI, NDBI, NDWI, SAVI, IBI, EVI**) to simulate various geospatial scenarios.
2. **Dynamic LST Prediction**: The system dynamically predicts the new **LST map** based on the updated input indices.
3. **Similarity Search**: The tool searches the DataCube for locations with embeddings similar to the modified input scenario.

Example Use Case

Urban Planning Scenario

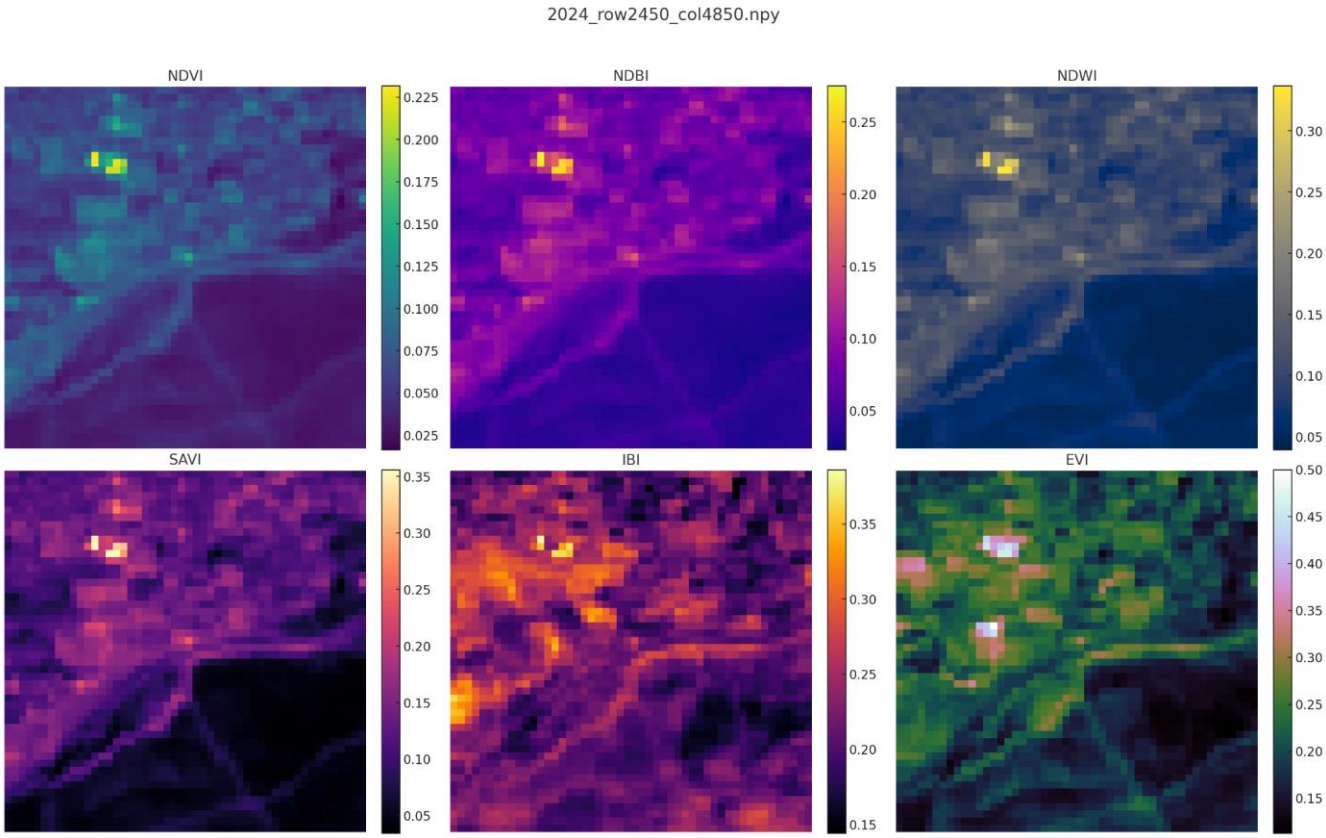
- **Initial Situation**: An urban planner is studying the effects of converting a semi-vegetated area into a fully urbanized zone.
- **Steps**:
    1. The planner reduces **NDVI** and increases **NDBI** values in the input.
    2. The endpoint predicts the resulting LST based on these changes.
    3. The planner identifies locations worldwide with similar urbanized characteristics to benchmark the changes.
- **Outcome**: The planner uses the LST predictions and similar location data to make informed decisions about heat mitigation strategies, such as green infrastructure or reflective building materials.

Benefits

- **Flexibility**: Users can explore the effects of hypothetical scenarios by adjusting band values.
- **Precision**: LST predictions provide localized insights into temperature changes.
- **Global Comparison**: Similarity search enables comparisons across diverse regions worldwide.
- **Scalability**: The DataCube architecture ensures the tool can handle large-scale geospatial datasets efficiently.

Funded by the European Union

Input:



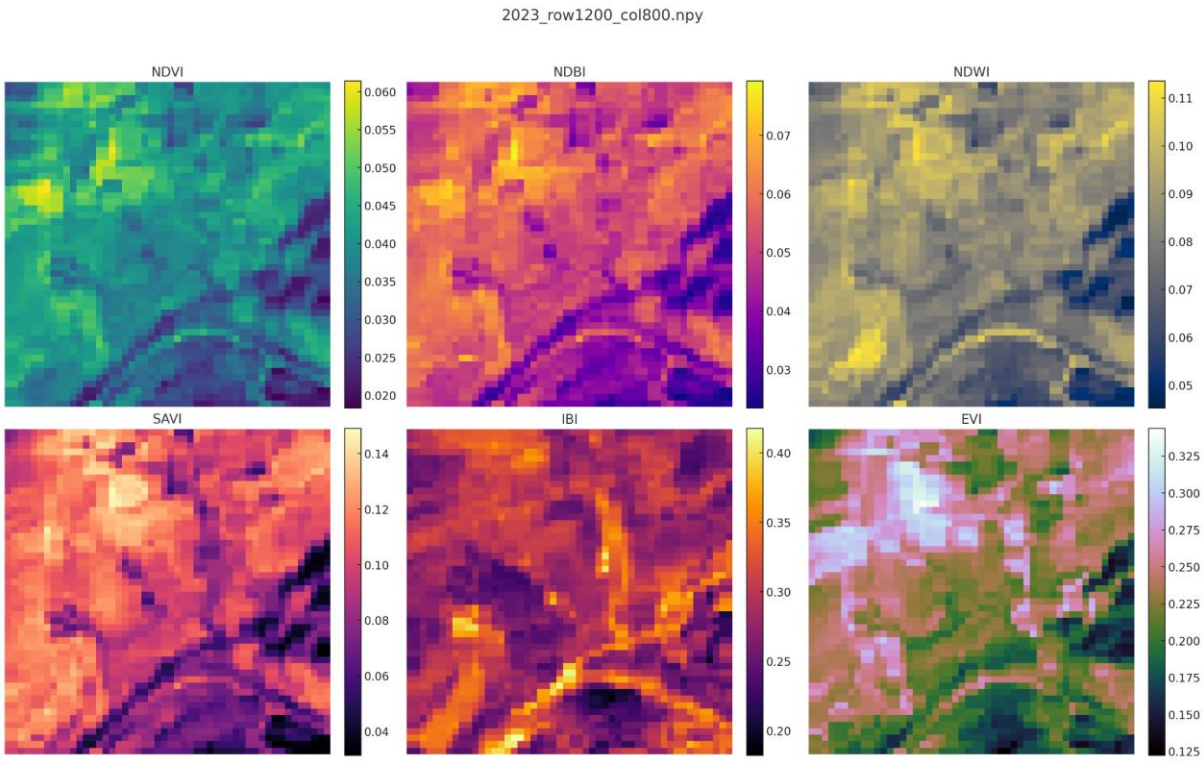
Here, the visuals display the input data from the queried location, which the model uses for subsequent analyses and predictions.

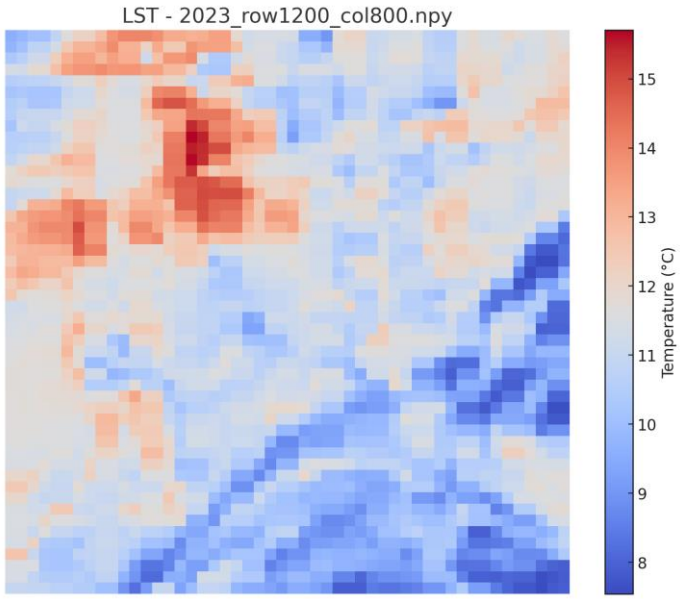
Output:

```
{
            "file_path": "2023_row1200_col800.npy",
            "similarity_score": 0.9891654849052429
}
```

This output indicates a strong match between the queried location and the dataset "2023_row1200_col800.npy," below with a similarity score of 0.989. In other words, the features of this matched dataset closely resemble those of the queried location. We use this approach not only to identify similar areas but also to handle outlier searches by pinpointing regions that deviate from expected patterns.

Funded by the European Union



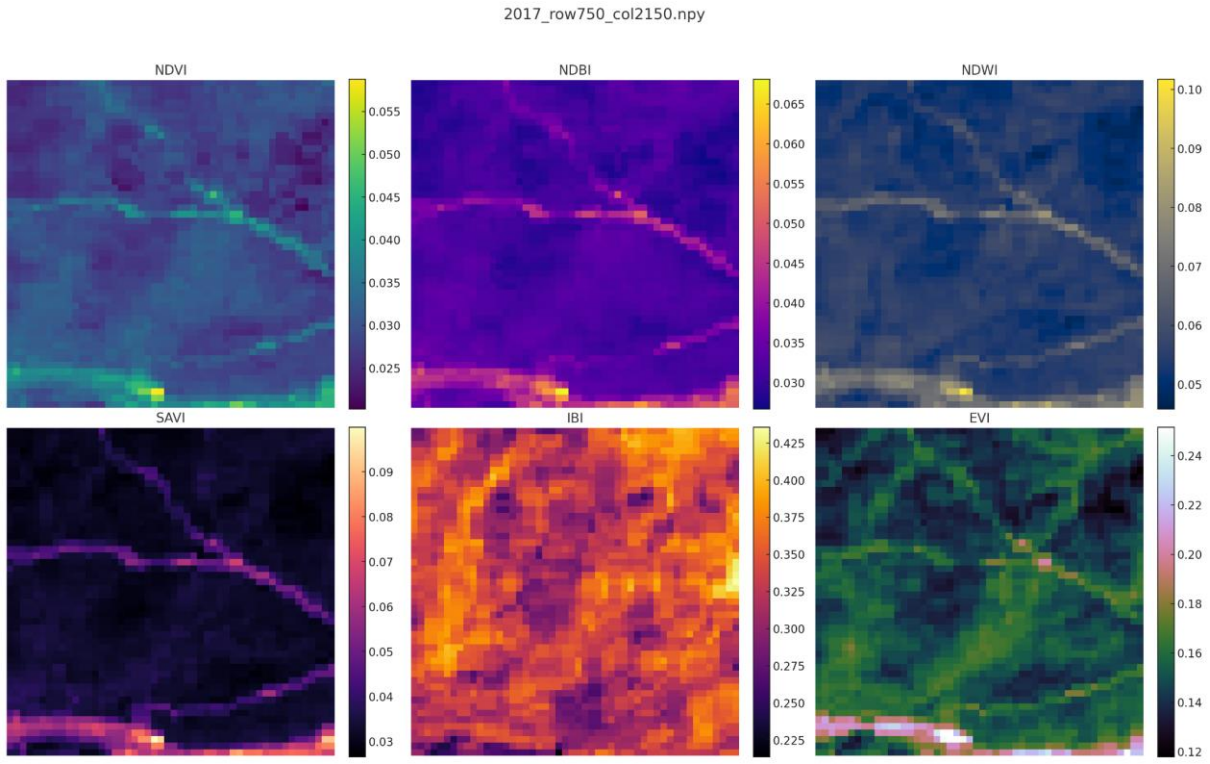


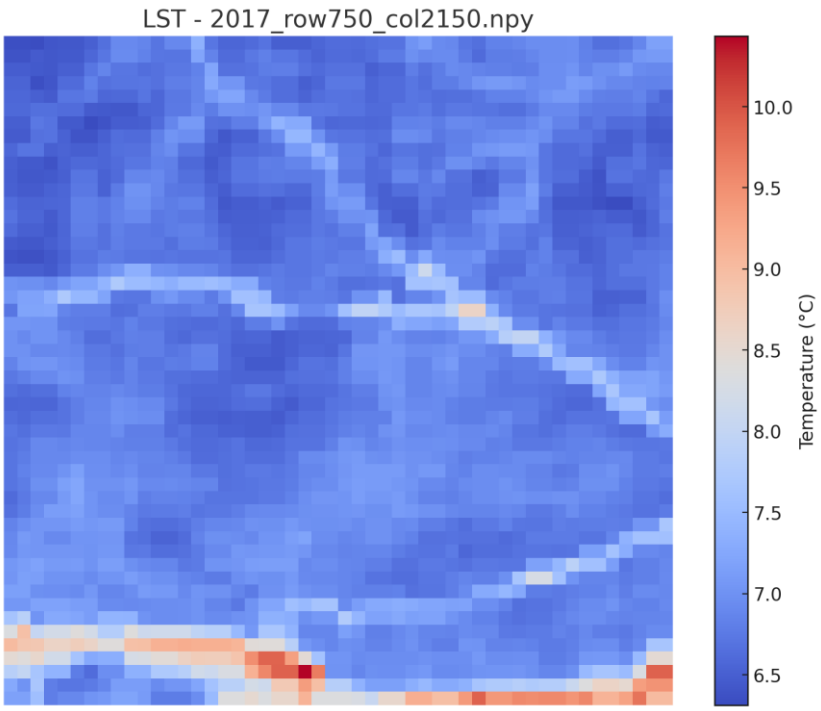
Funded by the European Union

Another Alternative Output:

This alternative output represents another strong match. The dataset "2017_row750_col2150.npy" has a similarity score of 0.9794186949729919.


```
{
            "file_path": "2017_row750_col2150.npy",
            "similarity_score": 0.9794186949729919
}
```



Funded by the European Union

5. Validation and Testing

5.1 Quality Assurance Protocols

To ensure the reliability and robustness of the Operational Exploration Tool and Matchmaking Tool, comprehensive quality assurance protocols were implemented during development. Key aspects include:

- **Functional Testing:** Verifying that each tool functions as intended across all features, including querying, visualization, and matchmaking.
- **Integration Testing:** Ensuring seamless interaction between backend, frontend, and AI-driven components.
- **Usability Testing:** Evaluating user experience to optimize tool design for a diverse audience, ranging from policymakers to researchers.
- **Performance Testing:** Assessing tool response times and scalability under varying workloads and data sizes.

5.2 Testing Framework for Backend API Services

5.2.1 Overview

The testing framework for the UDENE platform's backend API services is currently being integrated to ensure the system's reliability, accuracy, and robustness. We are actively developing integration tests, error handling tests, and machine learning endpoint tests to validate various functionalities. These tests are being written using the **pytest** framework and will soon be executed with the FastAPI TestClient to simulate requests and validate responses.

How to Run Tests

To execute the existing test suite:

1. Navigate to the project root directory.
2. Run the following command to execute all tests:
   pytest --markers integration
3. For detailed output with steps, use:
   pytest -v --steps

5.2.2 Integration Tests

Integration tests verify that the various API endpoints function as expected when interacting with other parts of the system. The following key endpoints were tested:

1. Health Check Endpoint

The `/health` endpoint performs a real-time system check, verifying that the backend services, including the DataCube connection, are operational. The test checks the endpoint's response status and verifies that the returned JSON contains a "healthy" status.

Funded by the European Union

2. Products Endpoint

The `/products` endpoint retrieves a list of available products from the DataCube. The test ensures that the endpoint returns a valid list of products and that the response status is 200.

3. Measurements Endpoint

The `/measurements/{product}` endpoint retrieves available measurements (bands) for a specified product. The test ensures the endpoint returns valid data and that the response status is 200.

4. Band Statistics Endpoint

The `/stats/{product}` endpoint calculates statistical measures (e.g., mean, max, min) for specified bands over a time range. The test verifies that the endpoint correctly handles the request and provides the expected output or appropriate error codes when data is unavailable.

5. RGB Composite Endpoint

The `/rgb/{product}` endpoint generates an RGB image composite from satellite data within a specified time range. The test ensures that the endpoint processes the request correctly, returns a valid response, and verifies the integrity of the generated RGB data.

5.2.3 Error Handling Tests

Error handling tests validate that the system gracefully handles incorrect or invalid input by returning appropriate error messages and status codes. These tests ensure the robustness of the API. Additionally, the endpoints created for the rapid implementation of the exploration tool are still under development.

1. Invalid Product Test

This test verifies that the `/measurements/{product}` endpoint returns a 404 status code when an invalid product is requested. This ensures that the system provides clear feedback when a user requests non-existent data.

2. Invalid Date Range Test

This test validates that the `/stats/{product}` endpoint returns a 422 status code when provided with an invalid date range (e.g., an end date that precedes the start date). The system should detect and reject such erroneous inputs.

3. Missing Required Fields Test

This test checks that the `/predict` endpoint returns a 422 status code when required fields are missing from the request body. The test ensures that the system enforces input validation.

Funded by the European Union

5.2.4 Machine Learning Endpoint Tests

Machine learning endpoint tests validate the functionality of endpoints that leverage machine learning models for predictions and analysis.

1. Prediction Endpoint Test

The `/predict` endpoint uses a machine learning model to generate predictions based on input satellite data. The test ensures that the endpoint processes the request correctly and returns prediction results, including metadata about the model and input data.

2. Find Similar Images Endpoint Test

The `/find-similar` endpoint identifies images similar to the input data based on a similarity analysis. The test verifies that the endpoint returns a list of similar images along with relevant metadata.

3. Combined Prediction and Similarity Search Endpoint Test

The `/predict-and-find-similar` endpoint combines prediction and similarity analysis into a single request. The test ensures that both functionalities are executed correctly, and the results are returned in a unified response format.

5.3 Alpha Release Testing Outcomes

The alpha release testing phase involved:

1. **Internal Testing:**
    - Conducted by the development team to identify and resolve bugs and inconsistencies.
    - Focused on core functionalities, including data visualization and matchmaking workflows.
2. **Stakeholder Testing:**
    - Early access provided to key stakeholders in Serbia and Tunisia to gather feedback on usability and relevance.
    - Test cases included predefined scenarios such as:
        - Estimating pollutant reductions in Novi Sad due to traffic rerouting.
        - Analyzing cooling effects of linked park systems in Greater Tunis.
3. **Feedback Integration:**
    - Suggestions from stakeholders were categorized into usability, functionality, and performance improvements.
    - Iterative updates addressed critical issues identified during testing.

5.3.1 Detailed Outcomes and Performance Metrics (Upcoming)

In the next phase of testing, we will enhance the transparency of our validation process by including comprehensive outcomes from our integration, error handling, and machine learning endpoint tests. Specific details will include:

Funded by the European Union

- **Integration Test Outcomes:**
  Detailed quantitative results for each API endpoint will be provided, including response times, throughput, and success rates. These metrics will allow us to evaluate performance against established benchmarks and highlight the system's robustness under varying operational conditions.
- **Error Handling Metrics:**
  We will document error rate statistics, such as the frequency and types of errors encountered during testing. Metrics will include the percentage of correctly handled invalid requests and system recovery times, providing insights into the resilience and fault tolerance of the API.
- **Machine Learning Endpoint Evaluation:**
  For endpoints utilizing machine learning (e.g., prediction and similarity searches), we will report performance indicators such as precision, recall, F1 scores, and inference times. This evaluation will measure the effectiveness and efficiency of the deployed models.
- **User Satisfaction and Usability Scores:**
  In collaboration with stakeholders and early users, we will integrate quantitative and qualitative feedback. Planned metrics include user satisfaction ratings, ease-of-use scores, and detailed feedback summaries to guide further refinements.

5.4 Validation Checklist

A high-level validation checklist was utilized to standardize quality control processes:

Validation Area	Checklist Item	Status
Functional Testing	Query and visualization features operational	☑ Complete
Integration Testing	Backend and frontend communication validated	☑ Complete
Usability Testing	Feedback from stakeholders incorporated	☐ Complete
Performance Testing	Tool scalability tested under peak conditions	☐ Ongoing
AI Ethics Compliance	Alignment with GDPR and ethical guidelines	☑ Complete

Funded by the European Union

6. Future Work and Recommendations

6.1 Planned Enhancements

Based on feedback from the alpha release testing and stakeholder inputs, the following enhancements are planned:

- **Advanced Analytics Features (M19 – Based on 1st Open Call Proposals):**
  - Incorporation of predictive modeling capabilities for environmental and urban planning scenarios.
  - Enhanced AI-driven recommendations based on dynamic data inputs.
- **Expanded Dataset Integration (M22 – Based on Open Call winners' demands):**
  - Integration of additional local and global datasets to broaden tool applicability.
  - Development of APIs for seamless data exchange with external platforms.
- **User Interface Refinements (M20 – Beta Version):**
  - Improved customization options for dashboards and visualizations.
  - Simplified workflows to enhance usability for non-technical stakeholders.

6.2 Scaling and Deployment

- **Geographic Expansion:**
  - Extend use case coverage to additional cities and regions with similar urban and environmental challenges.
  - Collaborate with international partners to validate tools in diverse geographic and climatic conditions.
- **Platform Scalability:**
  - Optimize backend infrastructure to handle increased user loads and larger datasets.
  - Incorporate cloud-based solutions for enhanced performance and accessibility.

6.3 Stakeholder Engagement

- **Workshops and Training:**
  - Organize hands-on workshops for stakeholders to maximize tool adoption and effective utilization.
  - Develop comprehensive training materials, including user guides and video tutorials.
- **Feedback Mechanisms:**
  - Establish continuous feedback channels to refine tool functionalities post-deployment.
  - Integrate user suggestions into iterative development cycles.

Funded by the European Union

6.4 Long-term Vision

- **Sustainability Focus:**
  - Leverage insights from use cases to promote sustainable urban and environmental practices.
  - Collaborate with policymakers to align tool functionalities with long-term regional goals.
- **Open-Source Contributions:**
  - Explore opportunities to release portions of the tool's codebase as open-source, fostering community-driven innovation.

Funded by the European Union

7. Conclusion

The D3.4 deliverable represents a significant milestone in the UDENE project, showcasing the successful alpha release of the Operational Exploration Tool and Matchmaking Tool. These tools exemplify the project's commitment to leveraging advanced GIS data cubes, AI-driven analytics, and user-centric design to address complex urban and environmental challenges.

Through predefined use cases in Serbia and Tunisia, the tools will demonstrate their capacity to:

- Support data-driven decision-making by providing actionable insights.
- Facilitate stakeholder collaboration through intuitive interfaces and AI-powered matchmaking.
- Advance sustainable urban development and environmental conservation initiatives.

Key outcomes of the alpha release include:

- Validation of core functionalities, including data visualization (e.g., heatmaps, 3D renders) and querying capabilities (e.g., spatial and temporal filtering). Metrics such as response time, accuracy of data layers, and user interaction feedback were rigorously tested to ensure tool reliability.

- Identification of enhancements to be addressed in subsequent development phases. Planned enhancements include the integration of predictive analytics for better decision-making, expanding dataset compatibility with real-time data streams, refining user interfaces for improved accessibility, and enhancing the scalability of backend systems to support larger datasets and user bases.

Funded by the European Union