



# D4.2 INITIAL VERSION OF ENVISION PLATFORM

---

Project: Monitoring of Environmental Practices for Sustainable Agriculture

Supported by Earth Observation

Acronym: ENVISION



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 869366.

## Document Information

Grant Agreement Number	869366	Acronym	ENVISION
Full Title	Monitoring of Environmental Practices for Sustainable Agriculture Supported by Earth Observation		
Start Date	1 <sup>st</sup> September 2020	Duration	36 months
Project URL	<a href="https://envision-h2020.eu/">https://envision-h2020.eu/</a>		
Deliverable	D4.2 Initial version of ENVISION platform		
Work Package	WP4 – ENVISION service		
Date of Delivery	Contractual	M12	Actual M12
Nature	Report	Dissemination Level	PU
Lead Beneficiary	DRAXIS		
Responsible Author	Petros Gasteratos (DRAXIS)		
Contributions from	Stavros Tekes (DRAXIS), George Efthimiou (DRAXIS), George Letsos (DRAXIS), OCS		

## Document History

Version	Issue Date	Stage	Description	Contributor
D0.1	16/8/2021	Draft	Sent for review	DRAXIS
D0.2	30/8/2021	Draft	Review comments	OCS
F1.0	31/8/2021	Final	Revised based on review comments	DRAXIS

## Disclaimer

*This document and its content reflect only the author's view, therefore the EASME is not responsible for any use that may be made of the information it contains!*



## CONTENT

Executive Summary.....	5
1 Progress towards ENVISION Platform .....	6
1.1 Development methodology .....	6
1.2 Co-creation meetings .....	7
1.3 User experience.....	9
2 ENVISION Architecture .....	10
2.1 Additions to the initial architecture .....	10
2.1.1 System Components .....	10
2.1.2 Dataflow Diagram .....	13
2.1.3 Component Interface .....	14
2.1.4 Data Inputs/ Outputs .....	14
2.1.5 Application Interface.....	14
2.1.6 Security.....	15
2.2 Mapping of user requirements .....	15
2.3 ENVISION as an open-source platform .....	17
3 Tour to the ENVISION Platform .....	18
4 Next Steps.....	23
4.1 Features .....	23
4.2 Quality of information provided by the services.....	23

## LIST OF TABLES

Table 1: Sample table.....	<b>Error! Bookmark not defined.</b>
Table 2: Docker Containers along with the respective environments.....	11
Table 3: Analysis of the internal interfaces .....	14
Table 4: Input and output data.....	14
Table 5: Protocols for communication with external systems .....	14

## LIST OF FIGURES

Figure 1: Trello, ENVISION issue tracking system .....	7
Figure 2: Wireframe dashboard .....	7
Figure 3: Wireframe of the main view of a selected parcel .....	8
Figure 4: Dataflow Diagram.....	13

## Executive Summary

---

This document aims at describing the progress and accomplishments achieved in WP4 and it tries to give a tour to the ENVISION platform functionalities that were implemented. Furthermore, the deliverable includes a description of how the identified user requirements were addressed based on the implemented ENVISION functionalities. Finally, it offers an overview of the next steps that are envisioned right after the platform's delivery.

Specifically, the sections of this deliverables are:

**Section 1 – Progress towards ENVISION platform:** Contains the progress made by the technical team.

**Section 2 – ENVISION architecture:** Contains the traceability matrix table with regards to the identified user requirements and how they were addressed by the developed ENVISION functionalities. Furthermore, it illustrates modifications/ adaptations that have been performed in the initial architecture delivered in the D4.1 Architecture and Services Specifications report.

**Section 3 – Tour to ENVISION platform's functionalities:** Contains some user interfaces of the platform in order to give the reader a feeling of the environment and at the same time to demonstrate the main ENVISION functionalities.

**Section 4 – Next Steps:** Contains a brief description of the next steps that are planned after the release of the first version of the ENVISION platform.



## 1 Progress towards ENVISION Platform

---

ENVISION aims to develop a ready-to-market commercial platform of services, co-designed and co-created with PAs and CBs, fit to address their needs. In particular, ENVISION provides PAs and CBs with a robust and cost-efficient set of services allowing them to monitor continuously and systematically the agricultural land, shifting the focus from fragmented monitoring limited to specific fields and dates (or time window) to territory-wide and all-year-round monitoring. Acting as a trailblazer for organisations that monitor environmental- and climate- friendly agricultural practices stemming from CAP, ENVISION increases the effectiveness of risk-based analysis for the selection of farms for inspection; increases the efficiency and transparency of PAs and CBs' procedures when implementing the CAP; reduces the number of on-site visits by performing more targeted controls and reduces operational and administrative costs.

### 1.1 Development methodology

The Scrum<sup>1</sup> methodology was used for the development of the platform. All user requirements and needs identified during the user requirement phase as well as feedback received from the co-creation phase were collected and placed on an online tool (Jira<sup>2</sup>), creating sprint schedule in order to start analysing and working on the first sprints. The aim of this process is to assist the easy monitoring of the development tasks and better allocate the technical resources.

The technical team distributed the sprints as early as possible, in order to start building the Minimum Viable Product (MVP) for the co-production phase with the end-users to verify the offered functionalities.

Trello<sup>3</sup> was created in order to be used as an issue tracking system, where the business cases partners and the testers will report issues tracked during their use of the MVP (initial) version. This issue tracking issue will also be used during the business cases implementation phase in order to collect valuable feedback from the actual users.

---

<sup>1</sup> [https://en.wikipedia.org/wiki/Scrum\\_\(software\\_development\)](https://en.wikipedia.org/wiki/Scrum_(software_development))

<sup>2</sup> <https://www.atlassian.com/software/jira>

<sup>3</sup> <https://trello.com>

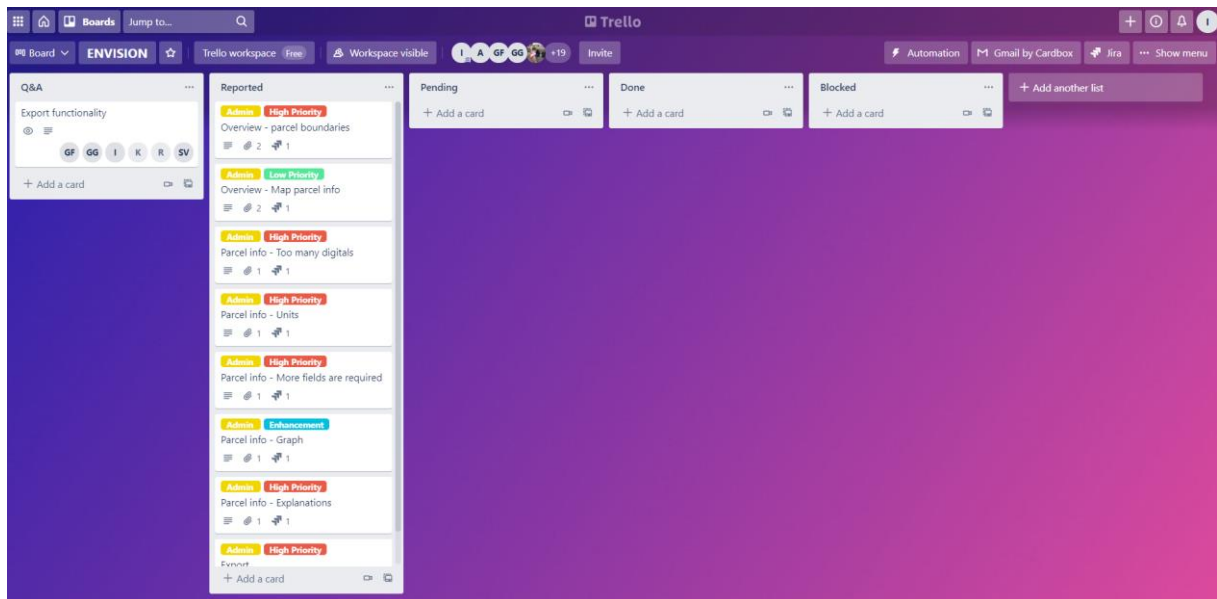


Figure 1: Trello, ENVISION issue tracking system

## 1.2 Co-creation meetings

During the user requirements and co-creation phases (in close collaboration with WP2 and WP3), the technical team (WP4) as well as the WP2 leader, the service providers and the end-users organized several meetings in order to collect valuable feedback on how to structure and design the platform and ameliorate the functionalities identified so as to collect all the needed information with regards to the ENVISION platform's operability, services' demonstration as well as feedback regarding the UI/UX. In order to assist these meetings, wireframes were designed by the technical team.

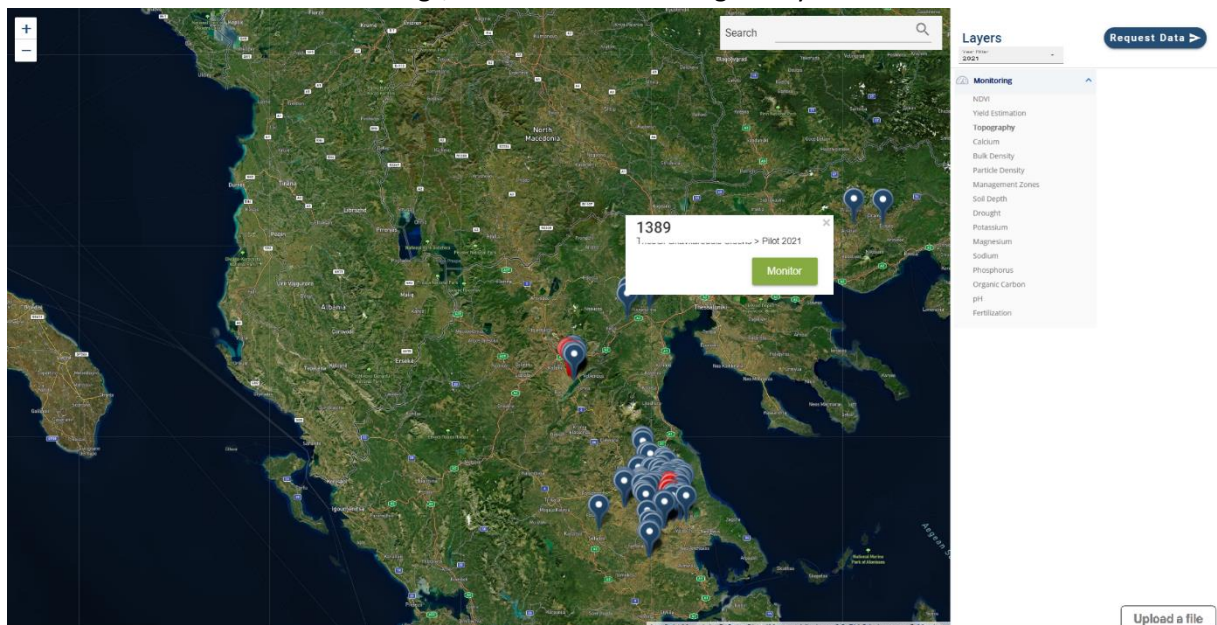


Figure 2: Wireframe dashboard

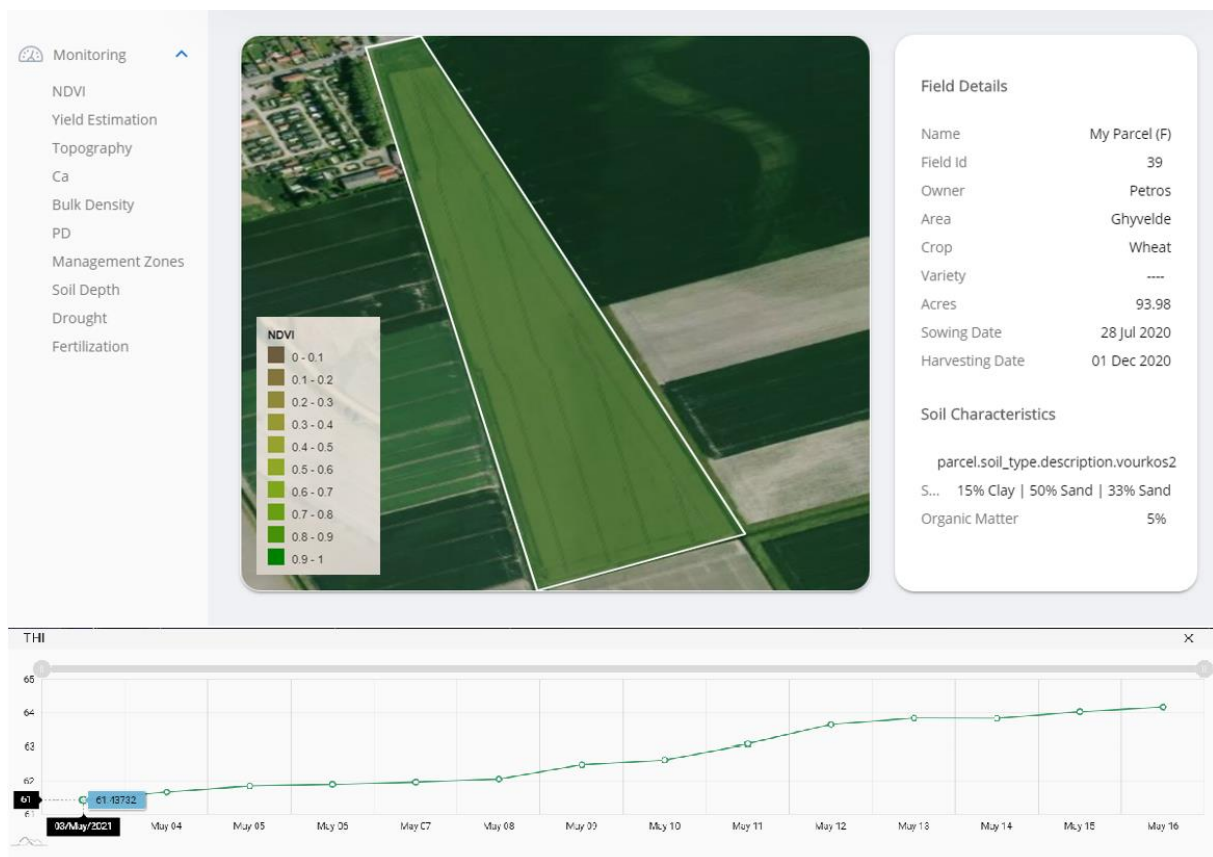


Figure 3: Wireframe of the main view of a selected parcel

Furthermore, the wireframes for the mobile application have been created to assist the workshops with the farmers and retrieve as much valuable comments/ feedback as possible.

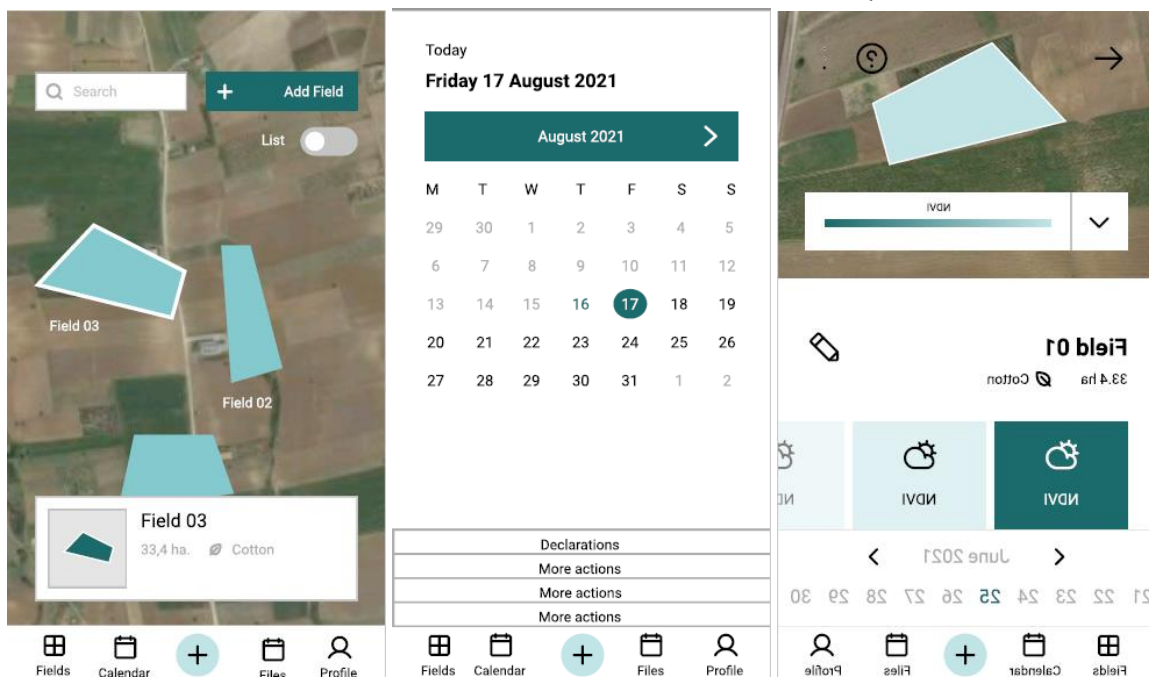


Figure 4: Mobile application wireframes



### 1.3 User experience

In order to develop a robust and solid platform, the users were engaged from the early stages of the development phase in order to better fit their needs. The platform should be easy to use, well organised and attractive, but in the same time functional and able to store, handle and serve different types of data. After going through the user requirements (D2.2), the wireframes were designed and based on the feedback received both from the service providers and the end-users, the technical team developed the initial (MVP) version of the platform.



## 2 ENVISION Architecture

---

The ENVISION architecture as it was presented in D4.1 Architecture and Services Specifications report was followed but with some changes.

### 2.1 Additions to the initial architecture

#### 2.1.1 System Components

The following Service-Oriented Architecture (SOA) approach is framed by technologies that handle the application development logic and the subsystems should be able to meet the needs for GIS and Web Mapping capabilities.

The data are stored in a geospatial database (for the case of vector files) and in the file system for the cases of raster files. A Central Authentication system controls the rules of user access to the application data. A Notification production subsystem is triggered by predefined conditions and is used to send emails to the users to inform them about potential issues.

More specifically, the components and technologies to be applied to the SOA approach are:

- **Geospatial Server:** The GeoServer allows the display of spatial information to the users. Implementing the Web Map Service (WMS), WMS-T standard, GeoServer can create maps in a variety of output formats. GeoServer is built on GeoTools, an open-source Java GIS toolkit. There is much more to GeoServer than nicely styled maps. GeoServer conforms to the Web Feature Service (WFS) standard, and WCS standard which permits the sharing and editing of the data that is used to generate the maps. GeoServer also uses the Web Map Tile Service (WMTS) standard to split your published maps into tiles for ease of use by web mapping and mobile applications. GeoServer is a modular application with additional functionality added via extensions.
- **Web Server:** For sending the Web UI of the platform as well the RESTful APIs, a NGINX server has been used. NGINX is an open-source software for web serving, reverse proxying, caching, load balancing, media streaming, and more. It started out as a web server designed for maximum performance and stability. In addition to its HTTP server capabilities, NGINX can also function as a proxy server for email (IMAP, POP3, and SMTP) and a reverse proxy and load balancer for HTTP, Transmission Control Protocol (TCP), and User Datagram Protocol (UDP) servers.
- **Authorisation Server:** A KeyCloak authorisation server has been used to enable OAuth2 and OpenID Connect functionalities to the platform in order to protect the resources from unauthorised use. KeyCloak is a standalone authorisation server that undertakes to maintain and apply user access rules to systems centrally, based on the industry standards for identity and access management for modern applications and services.
- **Geospatial Database:** PostgreSQL is a powerful, open-source object-relational database system that uses and extends the SQL language combined with many features that safely store and scale the most complicated data workloads. PostGIS is PostGIS is an Open Geospatial Consortium (OGC) compliant software used as an extender for PostgreSQL, which is a form of object-relational database. While PostGIS is free and open source, it is used in both commercial (e.g., ArcGIS) and open-source software (e.g., QGIS). PostGIS extends

capabilities of PostgreSQL to increase its management capabilities by adding geospatial types and functions to enhance spatial data handled within a relational database structure. The language of PostGIS is similar to SQL and allows spatial analysis and typical queries to be performed on spatial data with relative ease. This makes it a powerful backend for databases within larger software, helping projects to use SQL-like functionality to do more complex spatial analysis and queries.

- **Web Application:** The Web Application is a JavaScript Single Page Application (SPA) powered by Vue.js and Vuetify. Vue.js is a modern, component-based, web application building framework and Vuetify is a components library and framework used on top of Vue.js for building interactive web interfaces by providing a set of pre-made common and mature components. The Web Application is the interface between the platform components and the Paying Agencies (PAs) and Certification Bodies (CBs) users. It contains all the necessary web elements and tools to provide a friendly-user experience.
- **Mobile Application:** A Mobile Application will be used to allow farmers to access information about their parcels. The development of the mobile application will be based on hybrid technologies such as Vue.js together with Capacitor. Capacitor is an open-source runtime for building web-based applications that can run natively on Android, iOS and the Web.
- **Parcel Import & Product Import components:** Those two similar components are the interface for populating the platform with parcels and data products provided by the PAs and CBs and the service providers respectively. The interface is a HTTP/RESTful APO and is protected through the access token mechanism provided by the Authorisation Server.
- **GeoServer Protection component:** In order to protect the spatial data served by GeoServer from unauthorized use, a dedicated component guards the access to them by acting as a reverse proxy. Like the import components the GeoServer Protection component also makes use of the central Authorisation infrastructure to achieve the access control.
- **Notifications' component:** The notifications' component is responsible for generating and dispatching emails with notifications whenever necessary.
- **Web App and Mobile App Application Programming Interfaces (APIs):** Those two components are RESTful APIs acting as the interface between the Core of the platform, the Web Application and the Mobile Application respectively.

All components developed as Docker Containers. A container is a standard unit of software that packages up code and all its dependencies so the application runs quickly and reliably from one computing environment to another. A Docker container image is a lightweight, stand-alone, executable package of software that includes everything needed to run an application: code, runtime, system tools, system libraries and settings.

Due to the variety of subsystems, technologies and dependencies of the entire system have been split into Docker Containers and therefore, environments as described below:

*Table 1: Docker Containers along with the respective environments*

<b>PostgreSQL/PostGIS</b>	The main database of the platform. It stores parcel data, and products inserted to the platform
<b>GeoServer</b>	It serves OGC Services to the platform such as WMS tiles with parcels and data
<b>Django/Backend</b>	The core part of the platform consisting of various components

	and RESTful APIs called “Django Apps”
<b>Redis</b>	A Redis server for supporting the Django/Backend asynchronous processes
<b>SPA/Frontend</b>	The PAs and CBs web application/UI. (A Vue.js SPA built on top of the Django/Backend APIs)
<b>KeyCloak</b>	The central Authorization Server used across the system for protecting the resources and managing the users
<b>KeyCloak DB</b>	A dedicated PostgreSQL database required for the operation of the Authorization Server

By splitting the system as described above, the systems’ dependencies are isolated avoiding conflicts between libraries, services and different programming languages. In addition, using the Docker-Compose tool, the appropriate orchestration between the containers can be added in order to keep the whole system running as a unique service. Docker-Compose is a tool for defining and running mutli-container Docker applications.



## 2.1.2 Dataflow Diagram

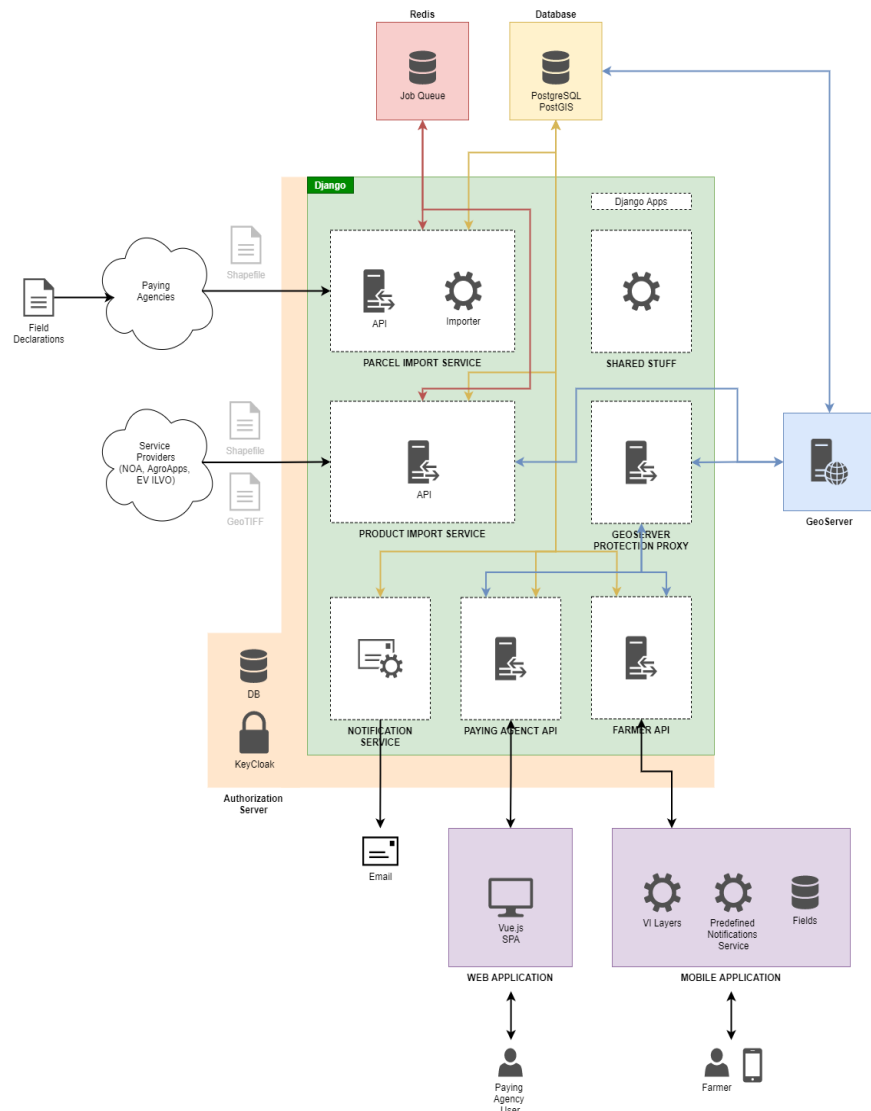


Figure 5: Dataflow Diagram

The above data flowchart presents the different components of the platform and the ways they communicate with each other. The data interconnection between PAs and CBs and service providers is performed via the Parcel Import and the Product Import RESTful interfaces.

The products produced by the execution of the models are a data source for the geospatial server which in turn offer OGC web services (WMS, WCS/ WMS-t) at the web level of the application.

Those OGC services are consumed by the Web Application component of the platform and especially through the OpenLayers-based web maps. OpenLayers is a powerful library for creating interactive maps for the web.

For the protection of the data and the platform processes from unauthorised usage, the access to the RESTful APIs is protected by Access Tokens powered by KeyCloak, the Authorisation Server of the ENVISION platform.

Finally, the farmer RESTful API will be the backend system that will drive the Mobile client applications that will be used by the farmers.

### 2.1.3 Component Interface

The table below presents an analysis of the internal interfaces used by the components as well as how they are interconnected.

*Table 2: Analysis of the internal interfaces*

<b>GeoServer</b>	Integrated with Postgres/PostGIS via Postgres Driver Integrated with Web via OGC services Integrated with Model's output via OS file system
<b>PostgreSQL</b>	Integrated with Django via ORM Integrated with GeoServer via Postgres/PostGIS Driver
<b>KeyCloak</b>	Integrated with Django via the python-keycloak library and custom code Integrated with the Web Application through the official Keycloak JavaScript library
<b>Web Application</b>	Integrated with Backend via RESTful API
<b>Mobile Application</b>	Integrated with Backend via RESTful API
<b>Notifications Component</b>	Integrated with Mail server via SMTP
<b>Django/Backend</b>	Integrated with Web Application via RESTful API Integrated with Mobile Application via RESTful API Integrated with PostgreSQL via ORM Integrated with GeoServer via HTTP Integrated with KeyCloak via the python-keycloak library and custom code Integrated with Redis via the Django-RQ library

### 2.1.4 Data Inputs/ Outputs

The following table reports the input and output data that the system receives and generates:

*Table 3: Input and output data*

<b>Vector files</b>	Shapefiles (.shp) provided by the Paying Agencies/ Certification Bodies
<b>Raster files</b>	GeoTIFF files from service providers as service products
<b>Rest Payloads</b>	JSON data for communication between components and external clients

### 2.1.5 Application Interface

The system provides capabilities for communication with external systems (third-parties) through the following protocols:

*Table 4: Protocols for communication with external systems*

<b>Application Programming Interface (API)</b>	Via RESTful APIs third party systems or developers will be able to consume systems functionalities
<b>OGC Services</b>	Via Common OGC services third party SDI or Desktop GIS software can consume Geospatial Products
<b>Postgres Connector</b>	Via Postgres connection third party Postgres Clients can gain access to system's database
<b>HTTP</b>	Via HTTP protocol service providers can push service's products into system



## 2.1.6 Security

The system communicates with the outside world permanently via secure HTTPs connections. The secure HTTP protocol provides privacy and data integrity between the communicating entities. In addition, the platform resources are protected by an OAuth2/ OpenID Connect infrastructure so that only authorised entities are able to access them.

From a system administration and maintenance perspective, the access to the system's servers, the containers of the components and the database data is achieved through SSH and SSH tunnelling with the use of private/ public keys.

On top of the above, a Firewall allows the access only to the necessary for the platform operation system ports and provides additional protection to the system.

## 2.2 Mapping of user requirements

The following table presents a requirements' traceability matrix with regards to the ones that refer to the platform development.

User Story - Requirement	ENVISION Platform
As a Controller, I would like to receive data of crop type maps every two weeks from the middle of April to the middle of August (ideally mid-September)	ENVISION offers a two-week frequency for output through the smart sampling algorithm.
As a Controller, I would like grassland mowing and grazing layers every two weeks from June till November with more than 85% accuracy	ENVISION offers a two-week frequency for output through the smart sampling algorithm.
As a Controller, I would like to be able to mask layers of interest with information from ENVISION outputs, for example to check parcels which intersect with soil erosion results, or to link crop type maps with grassland mowing layers	ENVISION offers the possibility to visualise results derived from the EO services as layers on top of the map.
As an Organisation, we would like to be able to identify and distinguish between organic and conventional crop, and to monitor pesticide use on the declared plots because this is an important objective in many agri-environmental policies	ENVISION offers a distinct classification/ categorisation on the platform between organic and conventional parcels that have been imported. Each of them is coloured with a different colour based on its category (green- organic, purple – conventional). Furthermore, vegetation indices are provided to the end-users as a layer on top of the map as well as through a graph.
As an Organisation, we want to get ENVISION outputs per parcel, especially for information on yield of each crop	ENVISION offers the possibility to export the outputs.
As an Organisation, we need the performance of the system to be fast, to enable quick testing	The ENVISION platform and relevant operations (data processing) is reasonably quick, since they are based on a web application.
As an Organisation, we want the system to provide us with errors against legislation that we can communicate to farmers	ENVISION provides notification to the end-users through email.
As an Organisation, we need to be able to integrate	ENVISION offers to the end-users APIs in order to be able to

services in our own applications. It is important to us that the ENVISION toolbox features as many standards as possible and that the various outputs are downloadable or easy to share via APIs so that we can analyse them in our own existing systems (interoperability and potential to transfer/download data)	integrate their existing systems with the ENVISION platform.
As an Organisation, we need all our data to be stored in one place	All relevant outputs and data are stored in the ENVISION database for the ENVISION lifetime. Participating organisations are able to download outputs (i.e., shapefiles, csv files etc.), share via APIs or access the data storage online
As an Organisation, we want to have an idea of the accuracy of the output of a service through relevant indicators and sufficient documentation of the methodology, as well as to receive notifications when the accuracy degrades throughout the cultivation period	ENVISION provides to the end-users along with the outputs of the services and the relevant accuracy.
As an IT expert, I want the ENVISION platform to monitor itself and notify me if there is a problem, so I can be confident that everything is ok if I am not notified	A notification mechanism has been development in order to automatically inform the end-users for any issue that may occur. This mechanism is based on the defined thresholds that the service providers have set.
As an Administrator, I need to know when ENVISION services' outputs are not available so I can warn the respective farmers that they need to provide the relevant information themselves	Indications are given if values are "Not Available – N.A" for certain pixels or parcels.
As an Inspector, I want the results from ENVISION's remote monitoring services to be reliable and verifiable on the spot	ENVISION is a web-based application and can be used anywhere since there is an internet connection. Furthermore, ENVISION offers the possibility to download the output and print them.
As an Organisation, we need to receive outputs both as maps/layers and relevant tables/numeric information, as well as to receive time series of various indicators to study changes and emerging problems	ENVISION offers to the end-users both layers and graph, depending on the output of each service.
As an IT Expert, I want the toolbox to be installed on DIASes, or that DIASes offer the tools as a service so it is preinstalled there, accessed and even maintained by the DIAS	The ENVISION platform has not been installed on DIASes. However, the DataCube has been built on DIASes and ENVISION will be integrated with it.
As an IT Expert, I want good quality to characterise the ENVISION platform services, in terms of ease of use, security and interoperability	ENVISION has paid great attention to the security both of the services and the data handling.
As an Organisation, we would like to be able to visualise historic data and all relevant to a plot information on the platform, for as far back in time as possible	ENVISION stores historical data with regards to compliance results. However, vegetation and biophysical indices will not be kept in the system.



As an Inspector, I would like to see through the ENVISION platform what is important to check for each plot, according to a farmer's declaration. This is important as it will clarify the reason why certain parcels need to be checked according to the organisation's sample	ENVISION will sent notifications to the end-users with possibly breaches and the users are able to perform queries in order to retrieve easily and faster results for parcels compliance.
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

## 2.3 ENVISION as an open-source platform

The ENVISION platform is an open-source platform and will be licenced under the GNU General Public License (GPLv3).

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program--to make sure it remains free software for all its users<sup>4</sup>.

The ENVISION platform will be released under the version 3. The GPLv3 explicitly gives everyone permission to use code that has requirements like restrictiveness.

---

<sup>4</sup> <https://www.gnu.org/licenses/gpl-3.0.html>

### 3 Tour to the ENVISION Platform

This section details the first version of the ENVISION platform that will be used by the business cases partners to test and validate its provided services and functionalities. This first release of the platform is as complete as needed until the next defined version.

The ENVISION platform is accessible through the following link: <https://platform.envision-h2020.eu/>. The administrator (technical team) will generate accounts for each business case in order for the users to have access to the web application.

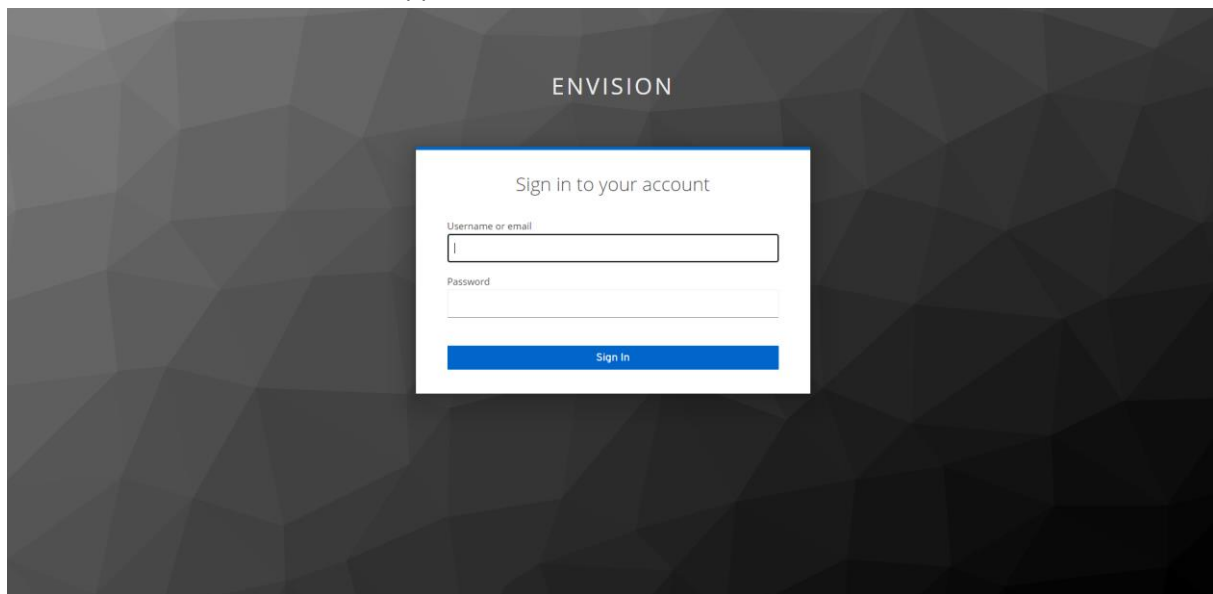


Figure 6: Log in

After the log in, the users land on the overview screen.

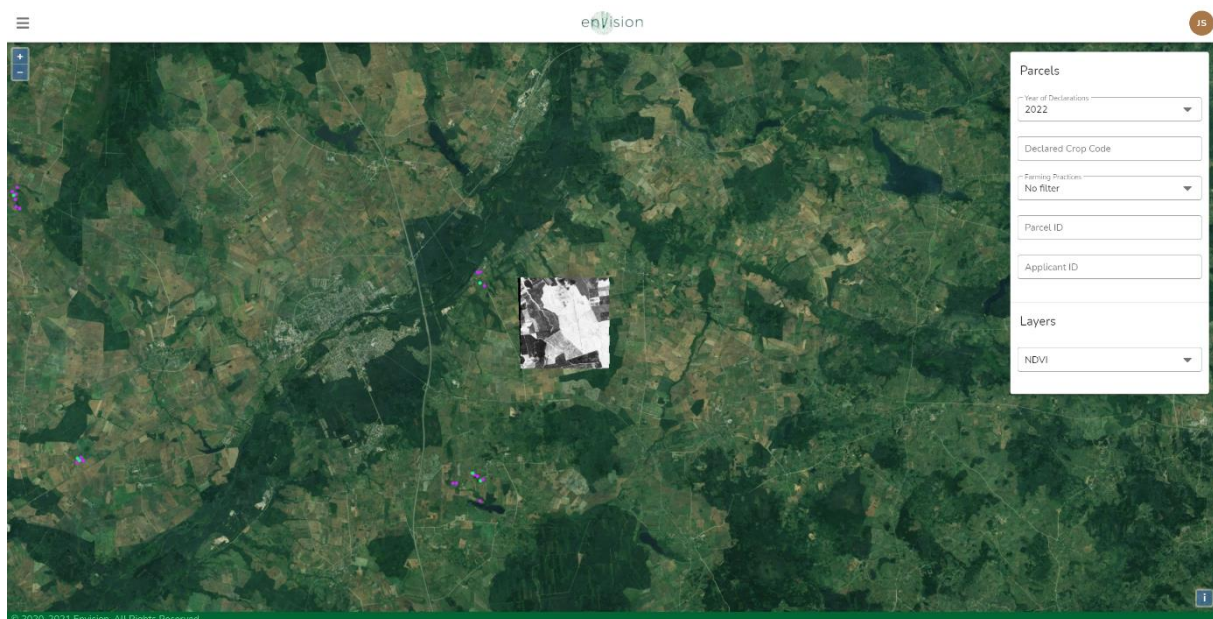
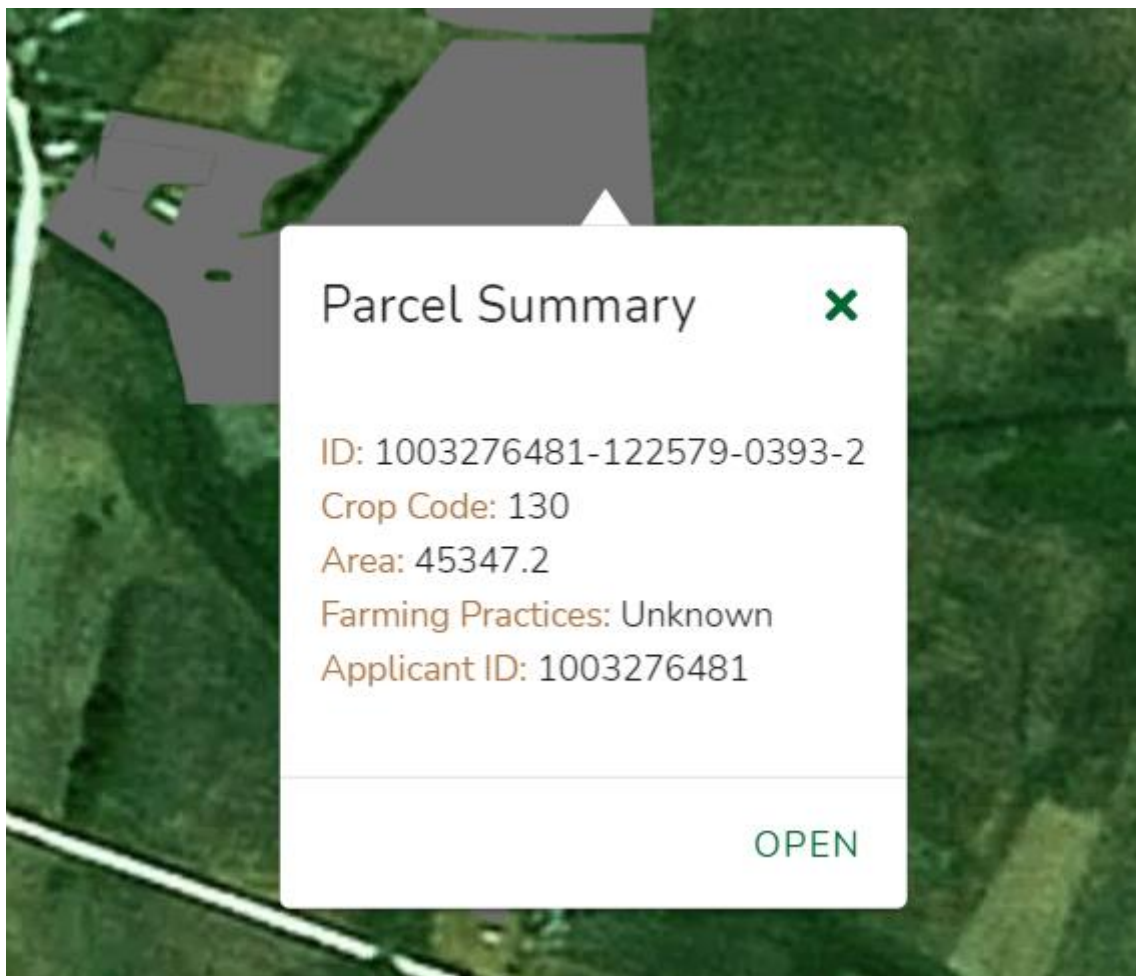


Figure 7: Overview

In the overview, the users are able to view all the imported parcels of the Area of Interest (AOI). If in the imported information is the categorisation (organic, conventional) filled in, then the parcels are coloured differently (green – organic, purple – conventional, grey – not defined). In that way, the users are able to identify easily and quickly the type of the parcel. Furthermore, in the overview, the users are able to visualise layers on top of the map and perform specific queries to retrieve information with regards to misclassifications, mowing events and burnt mapping. The users are also able to see previously imported declarations by selecting the preferred year and they can search a parcel either through a declared crop type or through a parcel and application ID. Finally, the users can filter the parcels and choose to see only the ones declared as organic or conventional.



*Figure 8: Parcel information*

When the users click on a parcel, then they can see the above information (ID, Crop Code, Area, Farming Practices and Applicant ID). Through the OPEN button, the users are directed to the monitoring screen of the selected parcel.

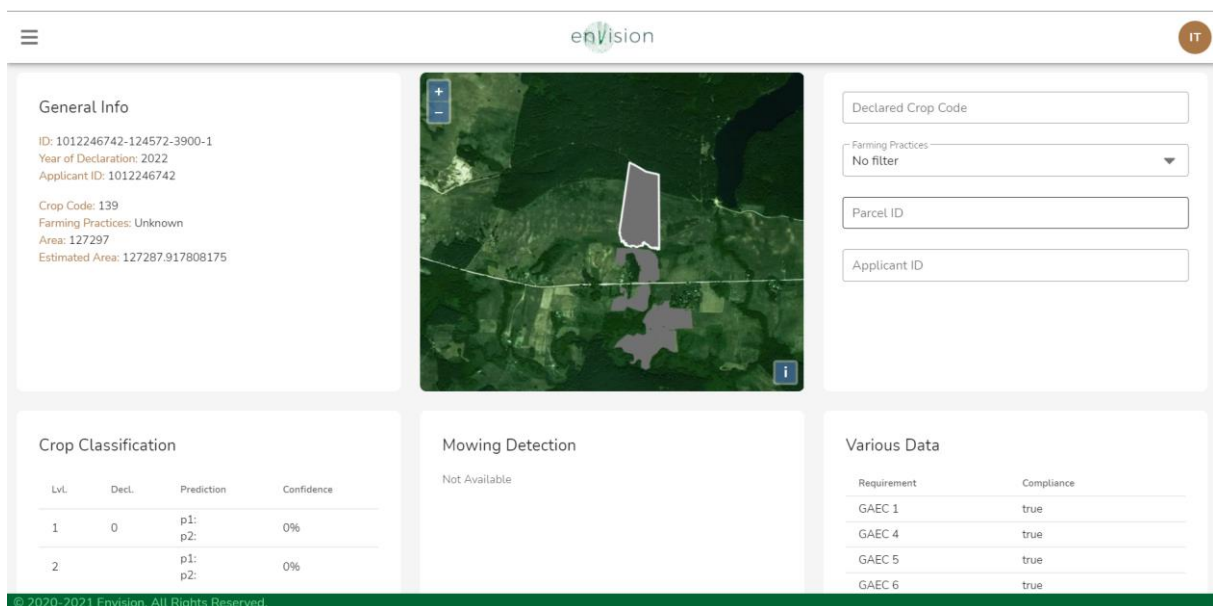


Figure 9: Parcel view

In the parcel view screen, the users can see general information for the parcel as well as the outcomes of the ENVISION services.



Figure 10: Parcel General Information

The General Information that are available to the users are:

- ID: The declared id of the parcel
- Year of declaration
- Applicant ID
- Crop Code: The declared crop code
- Farming Practices: The declared type of farming practices (organic or conventional)
- Area: The declared area
- Estimated Area: The area that is estimated from the ENVISION platform

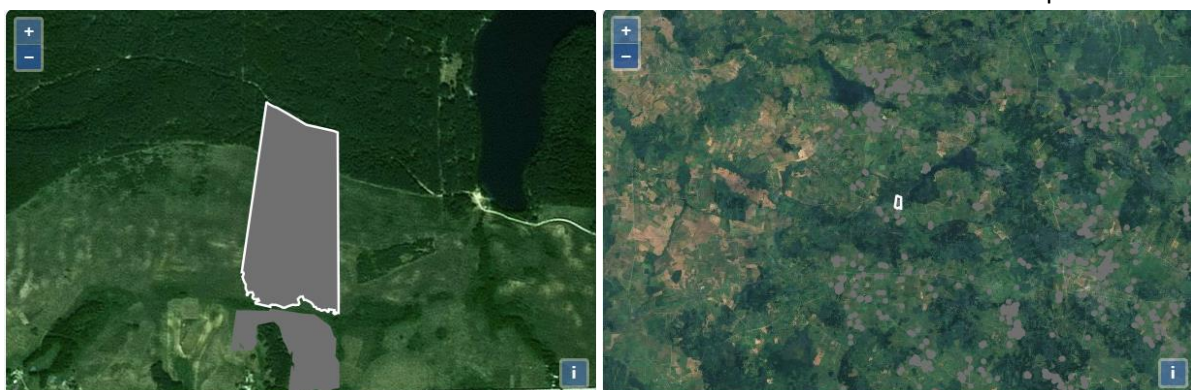


Figure 11: Parcel view in the map

The map in this screen is responsive. This means that the users are able to see the parcel in zoom but they are also able to zoom out and see the provided results in a wider area.



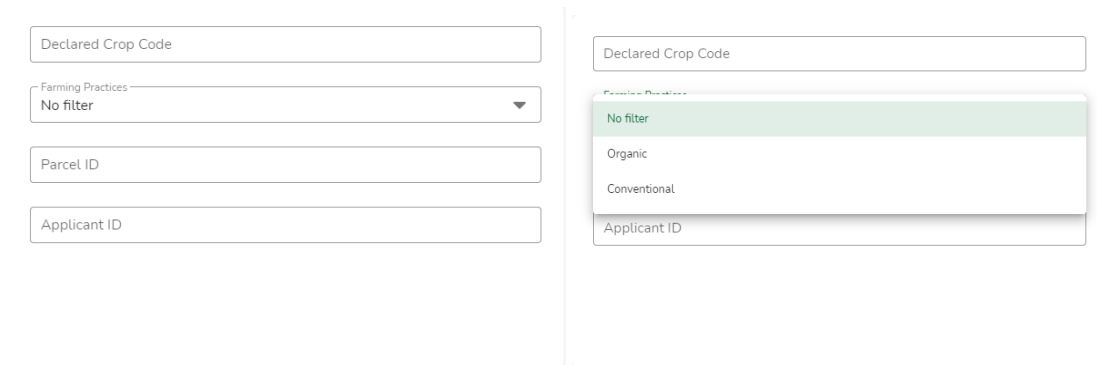


Figure 12: Filters

The users are able to search for a parcel through the filters depicted in the above figure. In addition, the users have the possibility to view several layers, derived from the EO-based services, on top of the map.

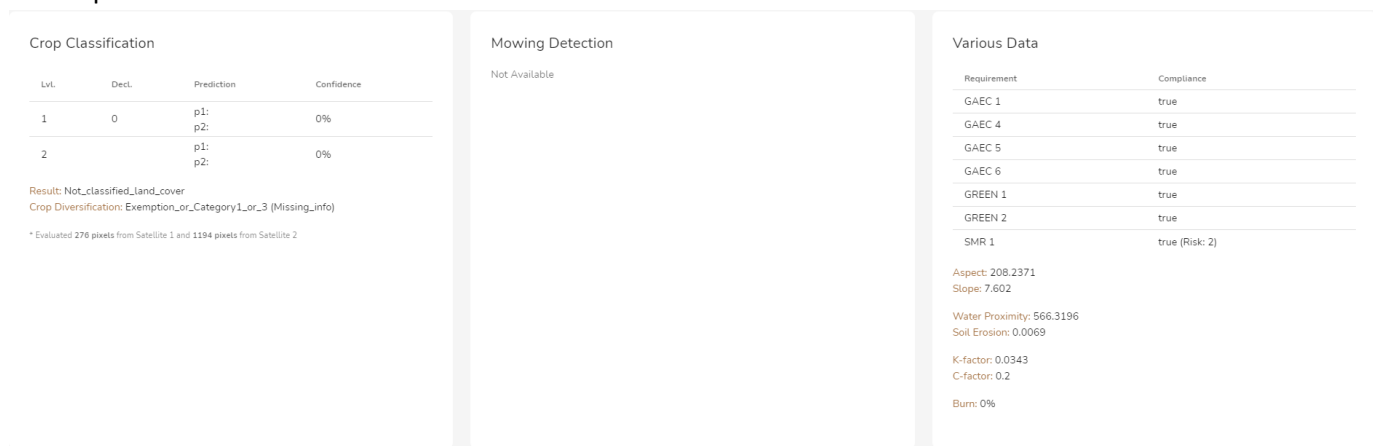


Figure 13: ENVISION Services

The ENVISION services with regards to CAP compliance are presented in the above figure. The users are able to see the results derived from the algorithms along with the respective accuracy. If a service does not have any result to demonstrate, then a message “Not Available” appears to the respective box.

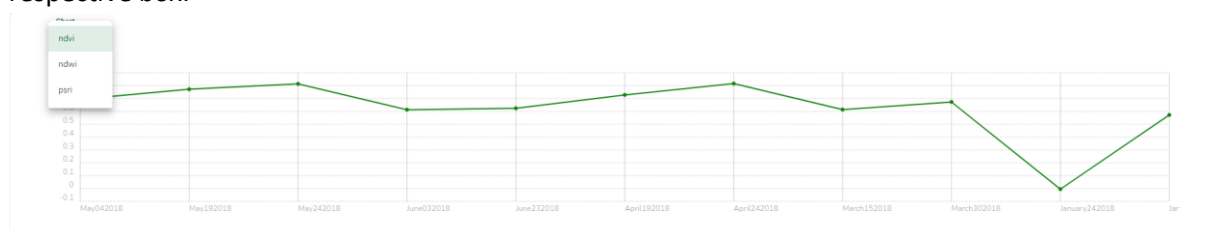


Figure 14: Graph

Furthermore, the users are able to see time-series of crop growth monitoring indices (NDVI, NDWI, PSRI, Yield Estimation).

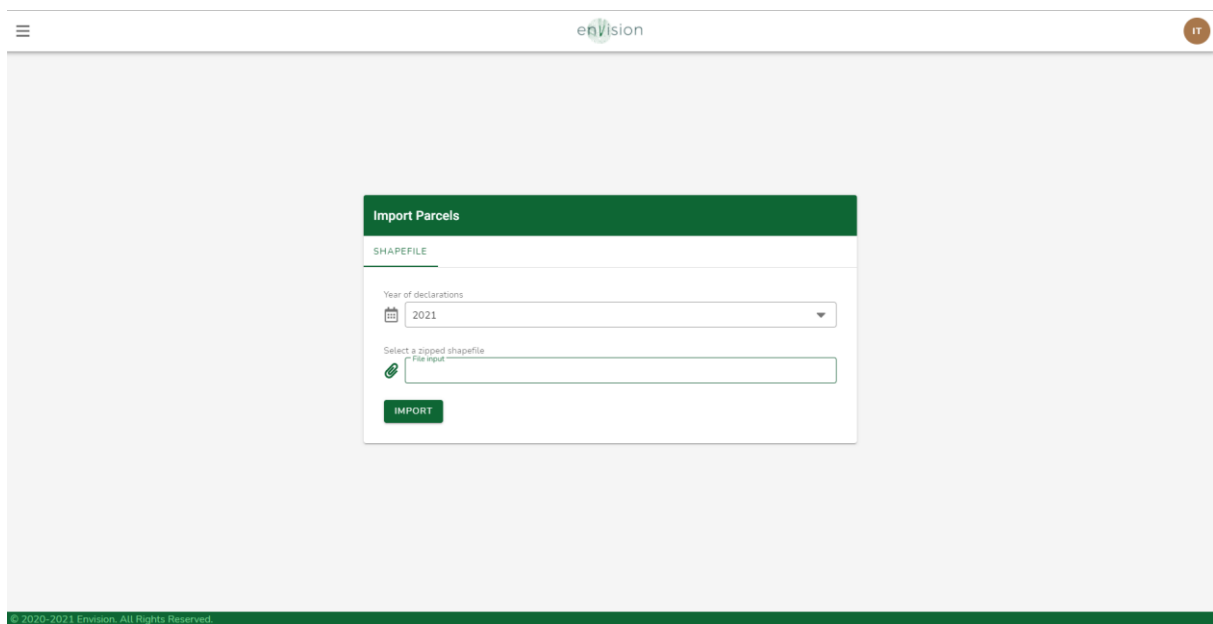


Figure 15: Import functionality

Up-to-now, the users are able to import the parcels through shapefiles and correlate them with the year of the declaration. In that way, the data are stored in the database based on the year declaration and gives to the users to opportunity to easily retrieve historical data.

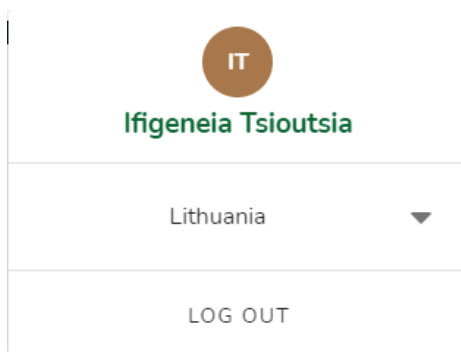


Figure 16: Use case assignment

Each user is assigned to a specific use case (i.e. Lithuania). However, the platform offers the possibility to assign more than one cases in a specific user (SuperUser). This process is implemented only for the administrator of the platform (the technical team).

## 4 Next Steps

---

Our plan is to continue improving the platform both from a feature perspective, but also from a perspective of performance and quality of provided information and user experience. This version is the initial one and does not include the integration with the service providers and any external system that will be used during the business cases implementation phase. This version was built in order to collect valuable feedback and better understand the users' needs and processes. Our aim is to build on top of this version and further improve it the next months in order to deliver a ready-to-market solution.

In addition, following the meetings with the farmers and developers and the feedback that will be collected, the mobile application and the Add-on Development tool will be built during the next months.

### 4.1 Features

There are some features that the ENVISION platform would good to have, but they were considered not to be critical for this period. Basically, these features were derived from the co-creation phase and will be implemented in the next version of the platform. Examples of these features are to integrate the platform with existing systems such as Geoserbja and the Geoservices of Flanders and add the export functionality. Further improvements will be communicated to the technical team after the platform will be released and be available to the end-users to test it.

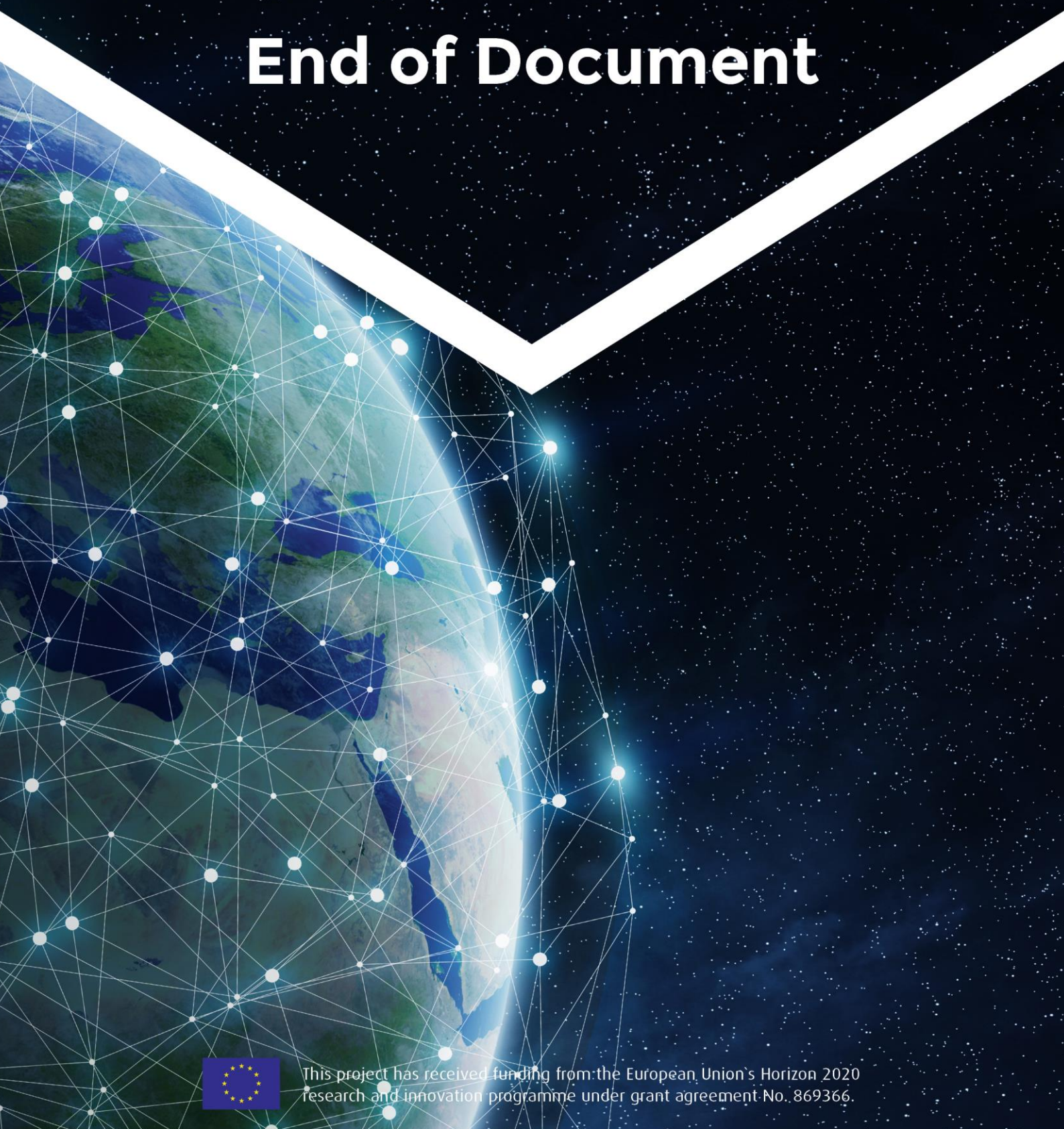
### 4.2 Quality of information provided by the services

During the business cases implementation phase, the quality of services will be assessed and changes/ adaptations/ improvements in the implementation of services might arise. WP4 will be in close collaboration with WP2, WP3 and WP5 in order to collect the required information, evaluate and validate the services and decide on the new approaches that should be followed, if it is needed, in order to improve the information provided by the current services.





# End of Document



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 869366.