KijaniSpace

Space-IoT Solution Box for Climate-Smart Agriculture in Africa

"Empowering Climate-Smart Agriculture with Copernicus and IoT Innovations for Smallholder Farmers in Africa"

Report D3.2

Development of KijaniBox

Beneficiary: Waziup

Document Reference: D3.2

Distribution: Public

Version: 2.0

Date: March 2026

Funded by the European Union

CONTRIBUTORS TABLE

Author(s)	Reviewer(s)
Corentin Dupont (IT21)	Ronald Ssembajwe (AIL)
František Zadražil (P4A)	Vivian Mwirigi (Waziup)

DOCUMENT REVISION HISTORY

Version	Date	Comments
1.0	30/03/2026	initial version
2.0	12/06/2026	Revision after reviewer's comments: 1. Added 2 paragraphs at the end of Section 2.2, to explain how the IoT data will be visualized and how the data privacy will be handled. 2. Added a new section 3.4. "IoT endpoints" to clarify the API with respect to the IoT data. 3. Added a paragraph in Section 4.2.3. "Waziup IoT platform back-end" to explain the mechanism regarding IoT data.

TABLE OF CONTENTS

1. Introduction

The KijaniSpace project represents a collaborative, innovation-driven initiative supported under the Horizon Europe programme (Project No. 101180225), coordinated by INNOTEC21 GmbH, and involving a consortium of thirteen beneficiaries from seven countries, including significant participation from Africa and Europe. The project's central goal is to enhance climate-smart agriculture practices in Africa by integrating Earth observation (EO) data from the Copernicus programme with Internet of Things (IoT) technologies, thereby contributing towards sustainable economic growth and green job creation.

The KijaniBox system (**hereafter Space-IoT Box**) is a platform that integrates geographic information systems, remote sensing, and Internet of Things from already tested EU initiatives and programmes for geospatial analytics and decision support. It is designed to facilitate the development of Copernicus downstream by developers from the Victoria Lake region. The platform exposes a comprehensive API layer that unifies access to existing solutions and enhances them with additional partner-provided services.

This deliverable details the development of the **Space-IoT Box** which comprises a dashboard (Section 1) and an API (Section 2). Section 3 details the development and technological choices. The platform will be used in WP4 (Talent Program & Innovation Program) and WP5 (downstream applications). The platform is currently deployed at https://app.kijanispace.eu/.

2. Space-IoT Box dashboard

This section presents the Space-IoT Box dashboard, the main user-facing component of the Space-IoT Box platform. The dashboard provides a unified environment for accessing, visualising, and interacting with geospatial information, Earth observation products, and IoT data relevant to climate-smart agriculture. Designed for developers, service providers, and other project stakeholders, the dashboard simplifies the exploration of datasets and services made available through the platform. It supports monitoring, analysis, and decision-making by combining map-based visualisation, data management features, and access to integrated Copernicus and partner-provided services.

The following subsections describe the main dashboard functionalities and illustrate the current implementation of the Space-IoT Box user interface.

2.1. Landing page

The Landing page section (Figure 1) welcomes the new users on the platform. Its different sections explain what the Space-IoT Box is and how to use it. Once you are ready, you can click on sign in. A classical sign-in page and registration page appears. Register with email and password. After successful registration login with your email and password.

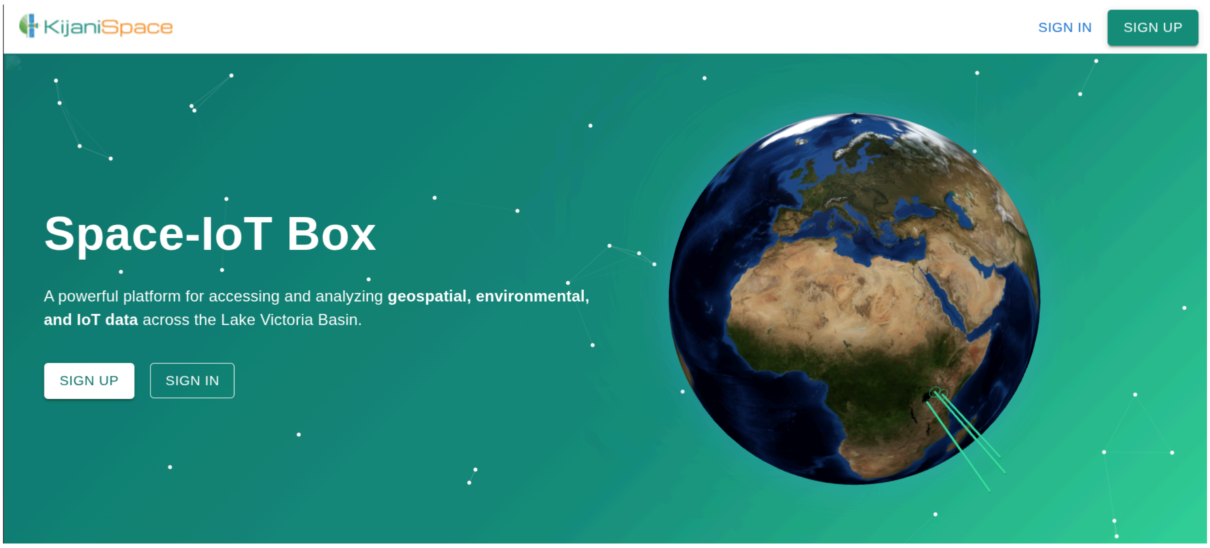


Figure 1: Space-IoT Box Landing page (1)

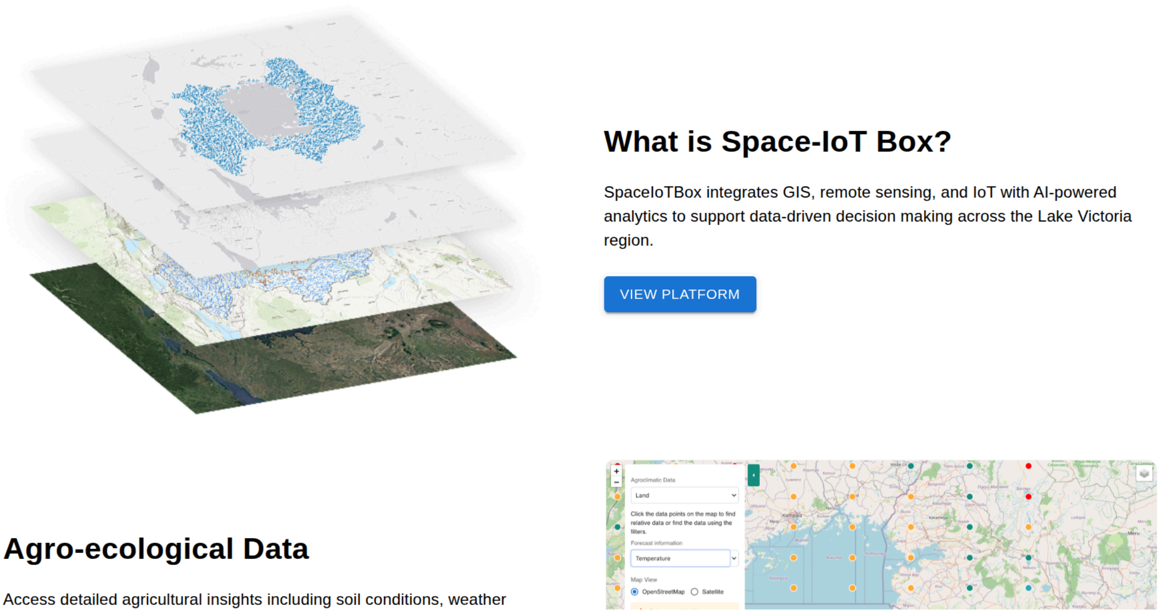


Figure 2: Space-IoT Box Landing page (2)

2.2. Agroclimatic Data

Once in, a screen similar to Figure 3 will appear. On the left-hand side, a vertical navigation menu allows users to access the platform's main features: Agroclimatic data, Satellite images, Data sources, Training, and IoT.

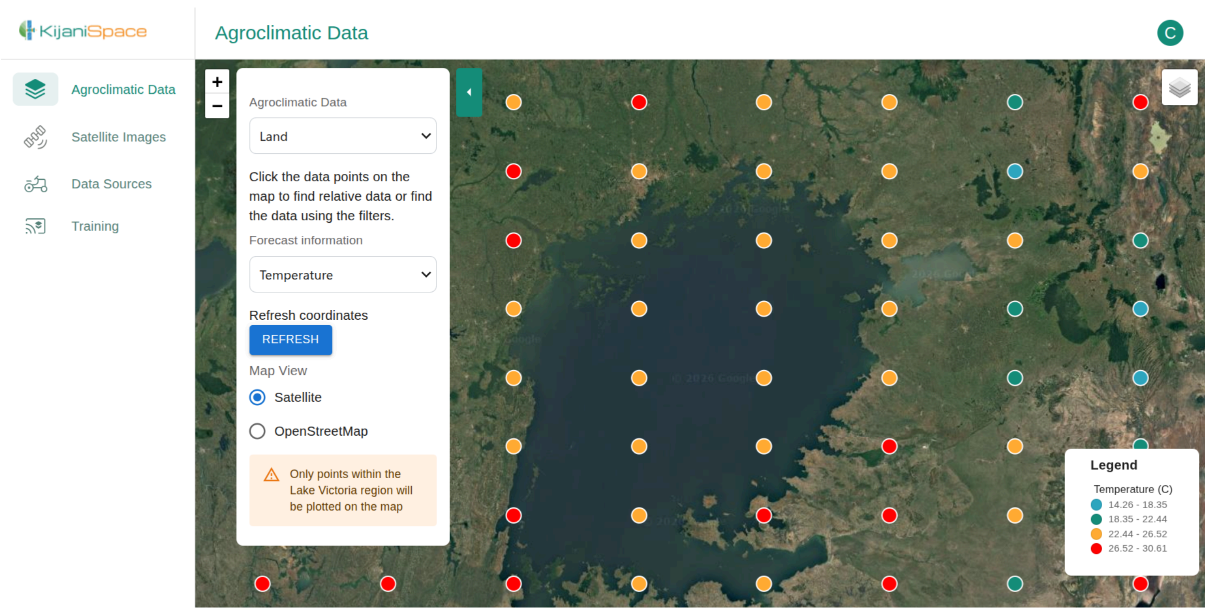


Figure 3: Agroclimatic section - Land data

The Agroclimatic data section allows users to explore and download historical and forecast data about the region. The screen displays a satellite view of the region. A floating menu allows you to choose between Land, and Water data. Once selected, you can choose which variable to display. For Land, the following are available:

- temperature
- felt temperature
- precipitation
- precipitation probability
- convective precipitation
- relative humidity
- wind speed
- wind direction
- soil temperature (0 to 10cm)
- soil moisture (0 to 10 cm)
- potential evapotranspiration
- evapotranspiration
- sensible heatflux
- skin temperature
- leaf wetness
- dew point temperature
- wet bulb temperature
- NDVI
- vegetation health index
- vegetation condition index

For Water, the following is available:

- water body percentage
- diffuse attenuation coefficient at 490 nm (monthly climatology)
- lake water temperature (monthly climatology)
- chlorophyll a concentration (8 day climatology)
- bathymetry depth

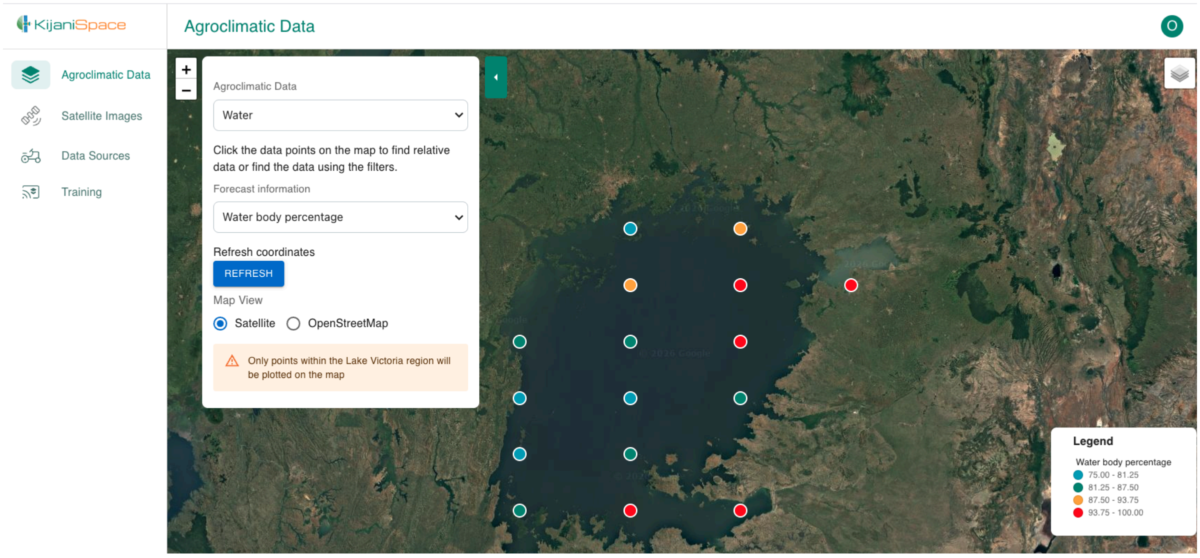


Figure 4: Agroclimatic section - Water data

Upon selecting one of these variables, the map will refresh and display a 10x10 grid showing the values around the lake. The color of the dots reflects the range of values displayed, from blue (lowest) to red (highest). A legend indicates the corresponding value ranges and unit of measurement. The user can move in the map and zoom in or out using the mouse. For example, he can zoom in on a specific farm and select the soil humidity in the menu.

Clicking on one of the dots, a graph appears. The graph is displaying the past, and forecasted values for the specific location and variable selected. For example, the graph in the following Figure shows the temperatures for the past day and the future 4 days in a location close to Mwanza, Tanzania. The graph can be zoomed-in and out using the tools at the top-right corner of the window.

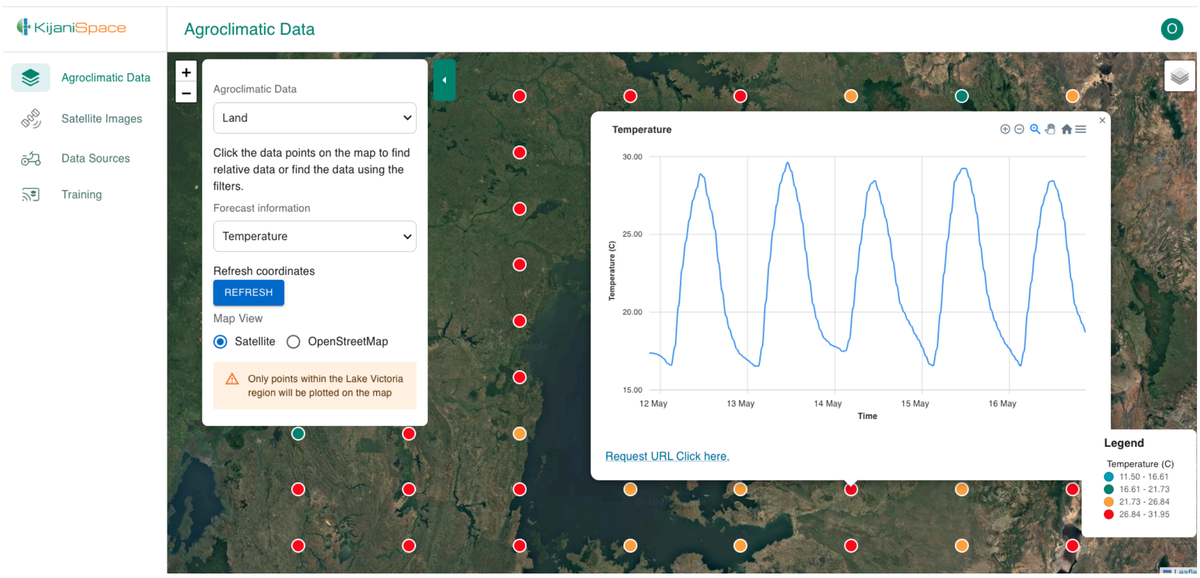
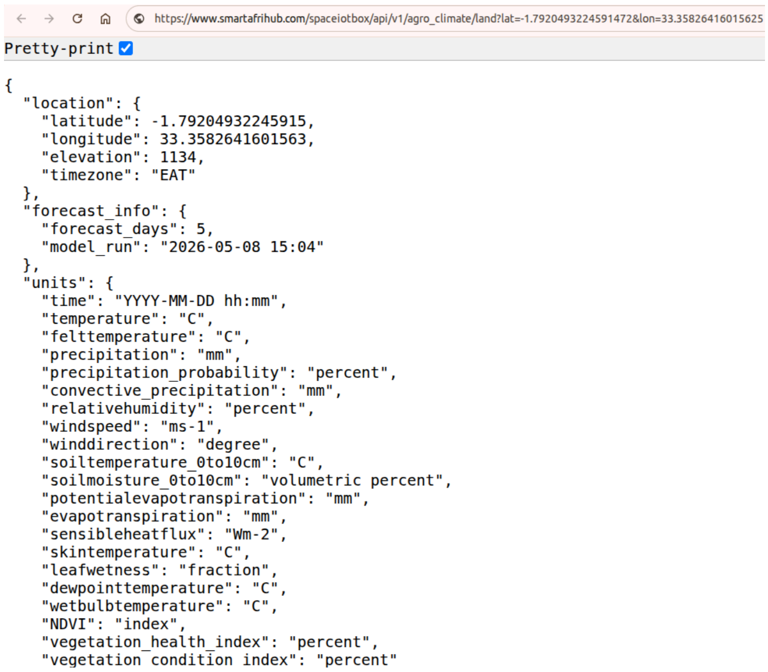


Figure 5: Agroclimatic section - Prediction graph

Finally, the data can be downloaded using the “Request URL click here” link at the bottom. This button gives you access to the REST API URL that the developers of KijaniSpace downstream apps can use (next Figure).



https://www.smartafrihub.com/spaceiotbox/api/v1/agro_climate/land?lat=-1.7920493224591472&lon=33.3582641601 5625

Pretty-print


```
{
  "location": {
    "latitude": -1.79204932245915,
    "longitude": 33.3582641601563,
    "elevation": 1134,
    "timezone": "EAT"
  },
  "forecast_info": {
    "forecast_days": 5,
    "model_run": "2026-05-08 15:04"
  },
  "units": {
    "time": "YYYY-MM-DD hh:mm",
    "temperature": "C",
    "felttemperature": "C",
    "precipitation": "mm",
    "precipitation_probability": "percent",
    "convective_precipitation": "mm",
    "relativehumidity": "percent",
    "windspeed": "ms-1",
    "winddirection": "degree",
    "soiltemperature_0to10cm": "C",
    "soilmoisture_0to10cm": "volumetric percent",
    "potentialevapotranspiration": "mm",
    "evapotranspiration": "mm",
    "sensibleheatflux": "Wm-2",
    "skintemperature": "C",
    "leafwetness": "fraction",
    "dewpointtemperature": "C",
    "wetbulbtemperature": "C",
    "NDVI": "index",
    "vegetation_health_index": "percent",
    "vegetation_condition_index": "percent"
```


Figure 6: Request URL used by KijaniSpace downstream app developers

Similarly, IoT data from the connected devices will appear on this map. Each IoT device will be represented as a dot and the historical data will be represented using the graph. The data is coming from the “WaziCloud” API. The IoT device management itself, including connectivity, ownership, and deployment will be handled on the WaziCloud platform[1].

On the WaziCloud dashboard, the user can decide if his devices will be “public” or not. If they are public, the data collected is available to anyone. In that case, the Space-IoT Box dashboard will display the device sensor value to anyone. If the device is set to “private”, only the owner of the device will be able to see the device on the KijaniBox dashboard[2].

[1] Dashboard.waziup.io
[2] https://app.wazilab.io/resources/waziup-e-v/wazicloud/docs/fg1DfNP7Les#devices

2.3. Satellite images

The next entry in the menu is “Satellite images”. This section gives access to raw satellite images that can be exploited by downstream application developers. The next Figure shows the page. Several satellite image thumbnails are displayed, one for each past day (when the data is available).

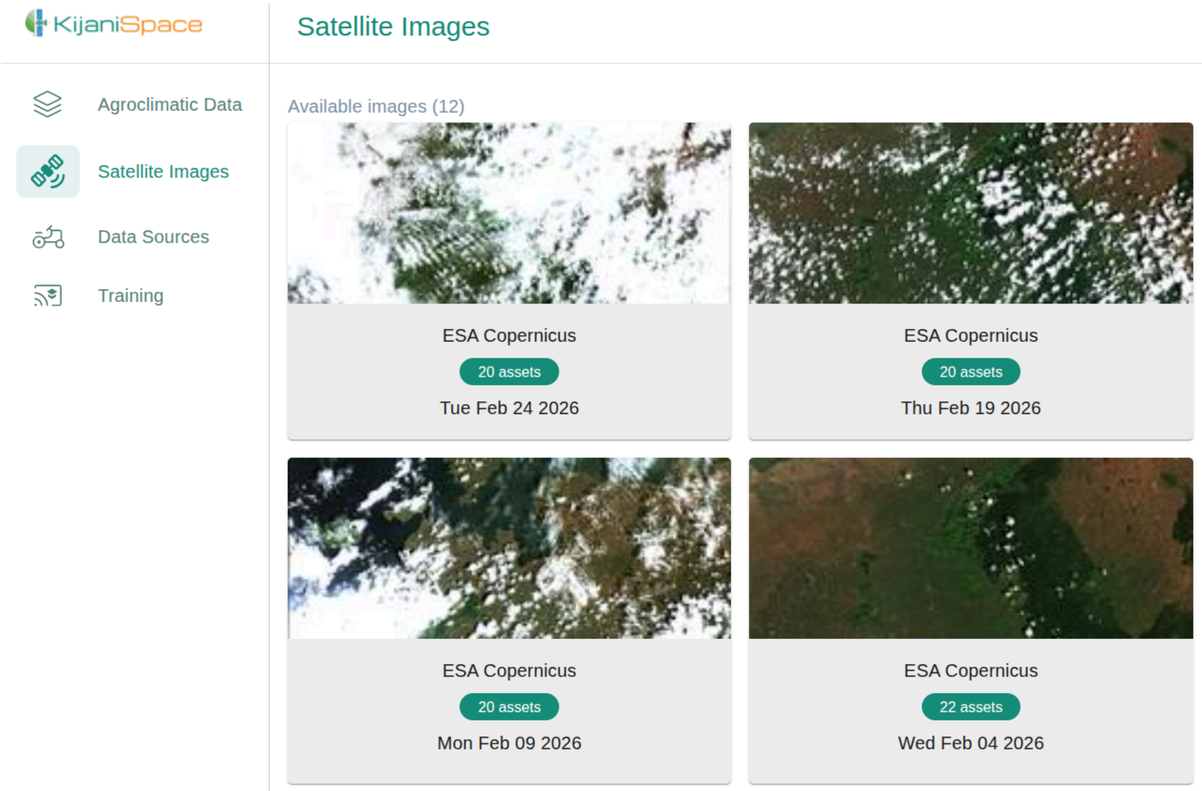


Figure 7: Space-IoT Box Dashboard UI: Satellite images Data section

Clicking on an image, a window appears (next Figure). It displays a list of downloadable assets. The assets are: NDVI and NDRE indices, TCI (True Color Images] and all the bands from raw Sentinel 2 scenes. The list of indices can be simply extended if needed in the future. The user can download each of them. Application developers can use the displayed REST API URL to integrate the image in their application.

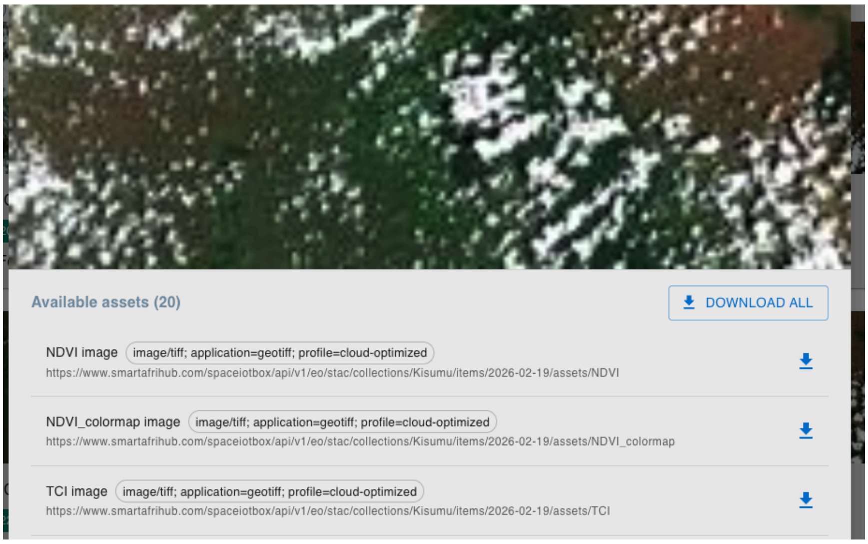


Figure 8: Space-IoT Box Dashboard UI: Satellite images download

2.4. Data sources

The "Data sources" menu entry gives the users and developers additional information about the source of the data and resources that they can use in their applications. The table (see next Figure) displays various information about each data source, such as title, description, subject, variables. The list of variables is an important indicator for the user looking for some specific information. Two "Action" icons link the user to the data source website for the first icon, and to the KijaniSpace training resources on WaziLab for the second icon.

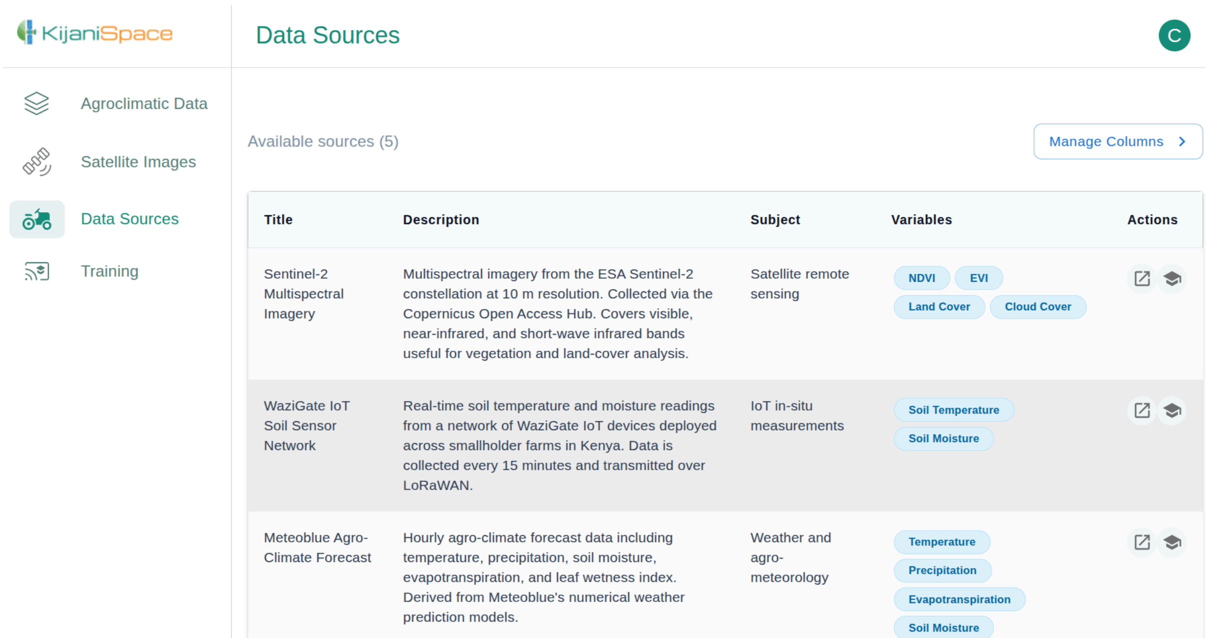


Figure 9: Space-IoT Box Dashboard UI: Data sources section

2.5. Training

The "Training" page shows the list of training resources prepared for the KijaniSpace project. It ranges from "Fundamentals of Spatial Data" to "Data-driven Precision Farming". Clicking on the links, the platform gives direct access to those training resources on WaziLab. The resources will be used in the KijaniSpace Talent Program, and also in the KijaniSpace Innovation Program. A user can obtain in-depth knowledge of different relevant topics, which are closely related to the Earth Observation, GIS and Internet of Things.

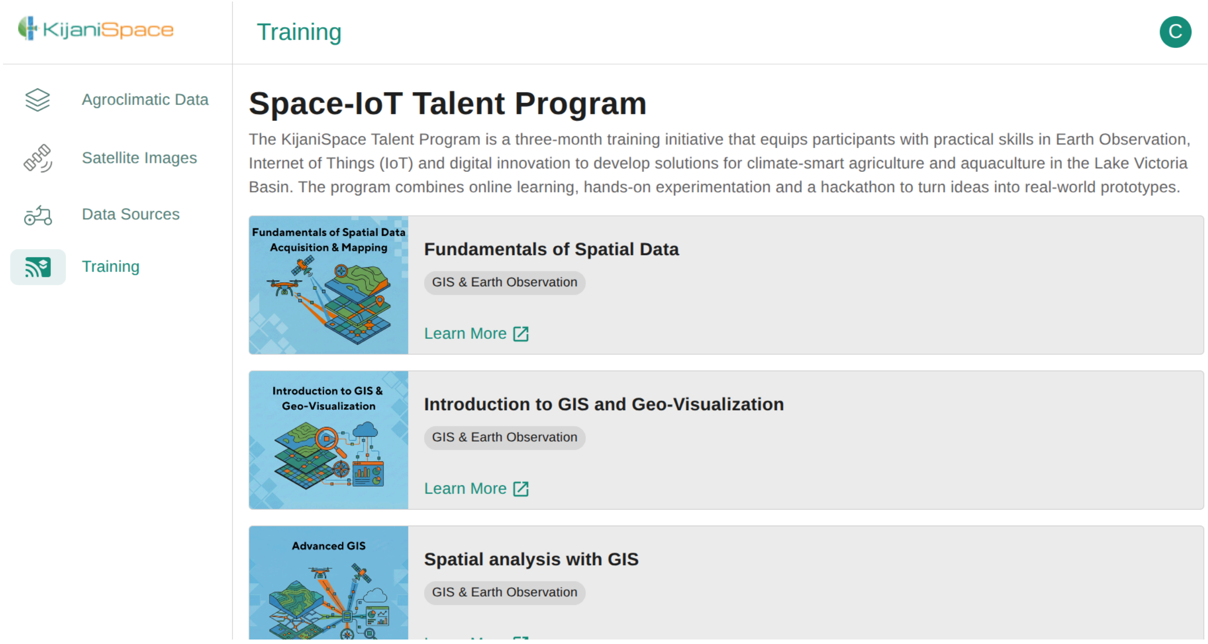


Figure 10: Space-IoT Box Dashboard UI: training section

3. API

The Space-IoT Box API is an important tool for Copernicus downstream application developers. It allows developers to easily access data (e.g. weather, agronomic data, satellite images) on the Lake region. The Space-IoT Box have the following high-level endpoints:

Table 1: Space-IoT Box API endpoints

Endpoint	Description
/api/v1/agro_climate	This endpoint provides a live time series about weather information, forecasts and agronomic data.
/api/v1/eo	Provide rasters of sentinel images and vegetation indices
/api/v1/datasources	This endpoint lists the data sources and describes them (e.g. URL, variables, resolution…). It is mainly used by the Space-IoT Box dashboard to display a list and a map display of the datasets.

The API uses URI-based versioning, where all endpoints are prefixed with /v1 (e.g. /v1/eo/stac, /v1/datasets). This approach makes the active version explicit in every request, allows multiple versions to coexist on the same server without conflict, and keeps client integrations stable.

The Space-IoT API is deployed at api.kijanibox.eu.

3.1. Agroclimatic endpoints

This endpoint provides weather information and forecasts. The API gives both historical data and forecasts.

Table 2: Space-IoT Box API Agroclimatic endpoints

Endpoint	Method	Query parameters	Description
/api/v1/agro_climate/land	GET	location (lat,lon)	Returns land-related agroclimatic time-series data. The query is for a single geographic location, combining historical observations and forecast data where applicable. The data returned is hourly. The forecast is returned for 5 days.
/api/v1/agro_climate/water	GET	location (lat,lon)	Returns water-related agroclimatic data. The query is for a single geographic location, combining historical observations and forecast data where applicable.

3.1.1. Query

The query contains only the location where the data is requested. The location is a geographical coordinate in latitude, longitude. Here is the format: lat=<lat>&long=<lon>

```JSON
Example query:
$curl
https://api.kijanispace.eu/v1/agro_climate/land?lat=42.90&lon=5.6
```

3.1.2. Return Output

The return structure contains a number of variables. Variable names are returned as keys (e.g. temperature, precipitation, windspeed). Units for each variable are provided in the top-level units object. Not all variables are available at all time resolutions. Here is the list of variables returned:

Soil & crop farming (forecasting up to 5 days):

- soiltemperature_0to10cm
- potentialevapotranspiration
- evapotranspiration
- soilmoisture_0to10cm
- sensibleheatflux
- skintemperature
- leafwetnessindex
- dewpointtemperature
- wetbulbtemperature

Soil & crop farming (no forecasting)

- NDV
- VHI (vegetation health index)
- VCI (vegetation condition index)
- GLOBCOVER (land cover)

Water & fish farming:

- water_body_frequency
- Water Temperature
- Water Turbidity
- Chlorophyll a concentration
- Bathymetry

Weather (forecasting):

- temperature
- felttemperature
- precipitation
- precipitation_probability

- convective_precipitation
- Uvindex
- Relativehumidity
- Windspeed
- winddirection

The returned data is a time series for the location.

```json
Example query:
$ curl
www.kijanispace.eu/api/v1/agro_climate/land?lat=42.90&lon=5.63
Response
{
  "metadata": {
    "modelrun_updatetime_utc": "2025-08-06 12:00",
    "name": "",
    "latitude": 48,
    "height": 560,
    "timezone_abbrevation": "CEST",
    "modelrun_utc": "2025-08-06 12:21",
    "longitude": 8,
    "utc_timeoffset": 2,
    "generation_time_ms": 13.342023
  },
  "units": {
    "soilmoisture": "volumetric percent",
    "temperature": "C",
    "transpiration": "mm",
    "sensibleheatflux": "Wm-2",
    "leafwetness": "fraction"
  },
  "data_day": {
    "time": [
      "2025-08-06T12:21:00+00:00",  "2025-08-06T13:21:00+00:00", …
    ],
    "potentialevapotranspiration": [3.5, 4.0, …],
    "soiltemperature_0to10cm_max": [19, 20, …],
    "wetbulbtemperature_mean": [13.8, 14, …],
    "leafwetnessindex": [43, 44, …],
    "soilmoisture_0to10cm_mean": [29, 30, …],
    …
   }
}
```

3.2. Earth observation (EO) endpoints

The EO endpoint is used to access satellite images. It provides standardised endpoints defined by the OGC STAC API[3][4]. The endpoint provides access to:

- Raw Sentinel-1 and Sentinel-2 data,
- True color images,
- NDVI, NDRE, EVI, and other vegetation indices.

3.2.1. Locations

Satellite images are grouped into locations. Location can be eg. “Kisumu”. This endpoint is the only non-standard STAC endpoint and is meant as a simplified list of available data.

Table 3: Space-IoT Box API EO endpoints

Endpoint	Method	Description
/api/v1/eo/locations	GET	Returns list of locations with data available. It includes a list of available data for each data product.

```
JSON
Request
GET https://www.smartafrihub.com/spaceiotbox/api/v1/eo/locations
Response

{
  "Kisumu": {
    "NDRE": {
      "dates": [
        "2026-01-05",
       ]
    },
    "NDVI": {
      "dates": [
        "2026-01-05",
      ]
    },
    "TCI": {
      "dates": [
        "2024-01-01",
        "2024-01-06",
```

[3] https://www.ogc.org/standards/stac/

[4] https://github.com/radiantearth/stac-api-spec

```
      ]
    },
    "raw_Sentinel2": {
      "dates": [
        "2024-01-01",
      ]
    }
  }
}
```

3.2.2. STAC Collections

Satellite images are grouped by collections. For example, Lake Victoria can be one collection. STAC Collection is the same as the location described in the previous chapter.

Table 4: Space-IoT Box API EO endpoints

Endpoint	Method	Description
/api/v1/eo/stac/collections	GET	Returns a FeatureCollection containing all collections (datasets) the API offers.
/api/v1/eo/stac/collections/{collectionId}	GET	Retrieves the metadata for a single collection (its description, spatial/temporal extent, available assets, etc.).

```
JSON
Request
GET https://www.smartafrihub.com/spaceiotbox/api/v1/eo/stac/collections
Response
{
  "collections": [
    {
      "id": "Kisumu",
      "title": "Kisumu",
        "description": "Collection of Copernicus data and its derived
products for the location of Kisumu",
      "extent": {
        "spatial": {
          "bbox": [
            [
              33.9072230525189,
              -0.989718108133729,
              35.7689860221686,
              -0.00292603289832081
```

```
            ]
          ]
        }
      },
      "links": [
        {
          "rel": "self",
                                                                    "href":
"https://www.smartafrihub.com/spaceiotbox/api/v1/eo/stac/collections/Ki
sumu"
        },
        {
          "rel": "items",
                                                                    "href":
"https://www.smartafrihub.com/spaceiotbox/api/v1/eo/stac/collections/Ki
sumu/items"
        }
      ]
    }
  ]
}
```


3.2.3. STAC Items

In the Space-IoT Box API, an item is a specific date (eg. “2026-02-24”). An item can then contain multiple assets. An asset is the actual data, typically a raster file (GeoTIFF, COG).

Table 5: Space-IoT Box API EO endpoints

Endpoint	Method	Description
/api/v1/eo/stac/collections/{collectionId}/items	GET	Returns a FeatureCollection of items belonging to the collection (supports paging, filtering via query parameters).
/api/v1/eo/stac/collections/{collectionId}/items/{itemId}	GET	Retrieves a single Item (the core STAC Item object).
/api/v1/eo/stac/collections/{collectionId}/items/{itemId}/thumbnail	GET	Shortcut to a small preview image (often provided by the thumbnail asset).
/api/v1/eo/stac/collections/{collectionId}/items/{itemId}/assets/{assetKey}	GET	Provides a redirect or direct download link to the binary asset (e.g., a GeoTIFF).

```
JSON
Request
GET
https://example.com/collections/sentinel-2-l2a/items?limit=2&bbox=-120,
35,-119,36

Response
{
  "type": "FeatureCollection",
  "features": [
    {
      "type": "Feature",
      "id": "S2A_20200101T103021_N0209_R108_T31TFJ_20200101T123456",
      "collection": "sentinel-2-l2a",
      "bbox": [13.5, 45.0, 14.5, 46.0],
      "geometry": { "type": "Polygon", "coordinates": [...] },
      "properties": {
        "datetime": "2020-01-01T10:30:21Z",
        "cloud_cover": 12.3
      },
      "assets": {
        "B04": {
          "href": "https://example.com/assets/S2A_..._B04.tif",
          "type": "image/tiff; application=geotiff",
          "title": "Red band"
        }
        // …other bands…
      },
      "links": [
                                     {     "rel":     "self",     "href":
"https://example.com/collections/sentinel-2-l2a/items/S2A_20200101T1030
21_N0209_R108_T31TFJ_20200101T123456", "type": "application/geo+json" }
      ]
    }
    // …more items…
  ],
  "links": [
                             {        "rel":        "next",        "href":
"https://example.com/collections/sentinel-2-l2a/items?cursor=abc123&lim
it=2", "type": "application/geo+json" }
  ]
}
```

3.2.4. STAC Search

Table 6: Space-IoT Box API EO endpoints

Endpoint	Method	Description

/api/v1/eo/search	GET or POST	Implements the STAC Item Search operation. Allows arbitrary filtering on geometry, datetime, collections, ids, properties, and extensions (e.g., query, sortby). Supports pagination (limit, page, token).

```json
Request
GET
https://example.com/search?collections=sentinel-2-l2a&datetime=2020-01-01/2020-01-31&bbox=-120,35,-119,36&limit=5

Response
{
  "type": "FeatureCollection",
  "features": [ /* array of matching items */ ],
  "links": [
     { "rel": "self", "href": "https://example.com/search?...", "type": "application/geo+json" },
     { "rel": "next", "href": "https://example.com/search?token=xyz789", "type": "application/geo+json" }
  ],
  "context": {
    "matched": 1243,
    "returned": 10,
    "limit": 10
  }
}
```

3.3. Datasource endpoint

This endpoint lists the data sources and describes them (e.g. URL, variables, resolution…). It is mainly used by the Space-IoT Box dashboard to display a list of the datasets.

Table 7: Space-IoT Box API datasources endpoints

Endpoint	Method	Description
/api/v1/datasources/	GET	Get all data sources in a list
/api/v1/datasources/	POST	Insert a new data source
/api/v1/datasources/<datasource_identifier>	GET	Get a single data source
/api/v1/datasources/<datasource_identifier>	PUT	Modify a single data source
/api/v1/datasources/<datasource_identifier>	DELETE	Delete a single data source

/api/v1/datasources/<datasource_identifier>/icon	POST	Upload icon of the datasource for use in the frontend apps

3.3.1. Return data

A data source is composed of the following fields:

Table 8: Space-IoT Box API datasources endpoints

Name	Definition	Type
identifier	A unique reference to the record within the catalogue.	ID
title	A name given to the resource.	String
subject	A topic of the content of the resource.	String
description	An account of the content of the resource. Also mention where it comes from/how it was collected (Satellite/IoT)	String
source	A reference to the full metadata from which the present resource is derived (URL).	String
training_doc	A reference to KijaniSpace training resources (e.g. WaziLab) URL	String
variables	Closed list of variables (e.g. Soil Temperature, Precipitation, Evapotranspiration), see next section	List of Strings
publisher	An entity responsible for making the resource available.	String
date	The date of a creation or update event of the catalogue record.	String
format	The physical or digital manifestation of the resource.	String
language	A language of the intellectual content of the catalogue record.	String
coverage	The spatial and temporal extent or scope of the content of the resource.	String
domain	The main business domain: soil_farming|fish farming|weather	List of Strings
icon	Display icon for the dataset	image

3.4. IoT endpoints

This Section lists the endpoints for retrieving IoT data on the platform. Using those endpoints, the user can retrieve the data from their deployed devices. The management of the IoT devices themself is done on the Waziup IoT platform.

Table 7: Space-IoT Box API IoT endpoints

Endpoint	Method	Description
/api/v1/devices	GET	Get all user devices in a list
/api/v1/devices/{device_id}	GET	Get a specific device
/api/v1/devices/{device_id}/sensors	GET	Get the list of sensors from a device
/api/v1/devices/{device_id}/sensors/{sensor_id}	GET	Get a specific sensor from a device
/api/v1/devices/{device_id}/sensors/{sensor_id}/values	GET	Get the values from a sensor

The user should provide the *device_id* and the *sensor_id* in order to use the endpoints. Both of those ids are available on the Waziup IoT platform.

3.3.1. Return data

The devices endpoints return the following data structure:

JSON

```json
[
  {
    "gateway_id": "MyGW",
    "location": {
      "latitude": 5.36,
      "longitude": 4.0083
    },
    "domain": "waziup",
    "visibility": "public",
    "name": "My weather station",
    "id": "MyDevice",
    "sensors": [
      {
        "quantity_kind": "AirTemperature",
        "name": "My garden temperature",
        "id": "TC1",
        "meta": {},
        "unit": "DegreeCelsius",
        "sensor_kind": "Thermometer"
      }
    ],
    "meta": {},
    "actuators": [
      {
        "actuator_value_type": "bool",
        "value": true,
        "actuator_kind": "Buzzer",
        "name": "My buzzer",
        "id": "Act1",
```

```
        "meta": {}
      }
    ]
  }
]
400
```

The sensors endpoints return the following data structure:

```JSON
{
  "quantity_kind": "AirTemperature",
  "name": "My garden temperature",
  "id": "TC1",
  "meta": {},
  "unit": "DegreeCelsius",
  "sensor_kind": "Thermometer"
}
```

The sensor values are of the form:

```JSON
[
  {
    "sensor_id": "TC1",
    "device_id": "MyDevice90",
    "value": 25,
    "timestamp": "2016-06-08T18:20:27.873Z"
  }
]
```

4. Implementation

The implementation details have been provided in the previous deliverable D3.1. In this section, we provide an update.

4.1. Overview

The following schema depicts the overall architecture of the Space-IoT Box API using the Archimate notation. The API provides endpoints for agro-climatic and earth observation data and also a list of relevant data sources. Meteoblue and Plan4all own APIs are used behind the scenes to provide some of the data and administration of them. Meteoblue provides their comprehensive data sets of forecast and agro data. Plan4all uses their services for management of satellite imagery that are then served to the Space-IoT Box API and its end-users. Additionally, the platform provides links to the Waziup IoT platform, and WaziLab training center.

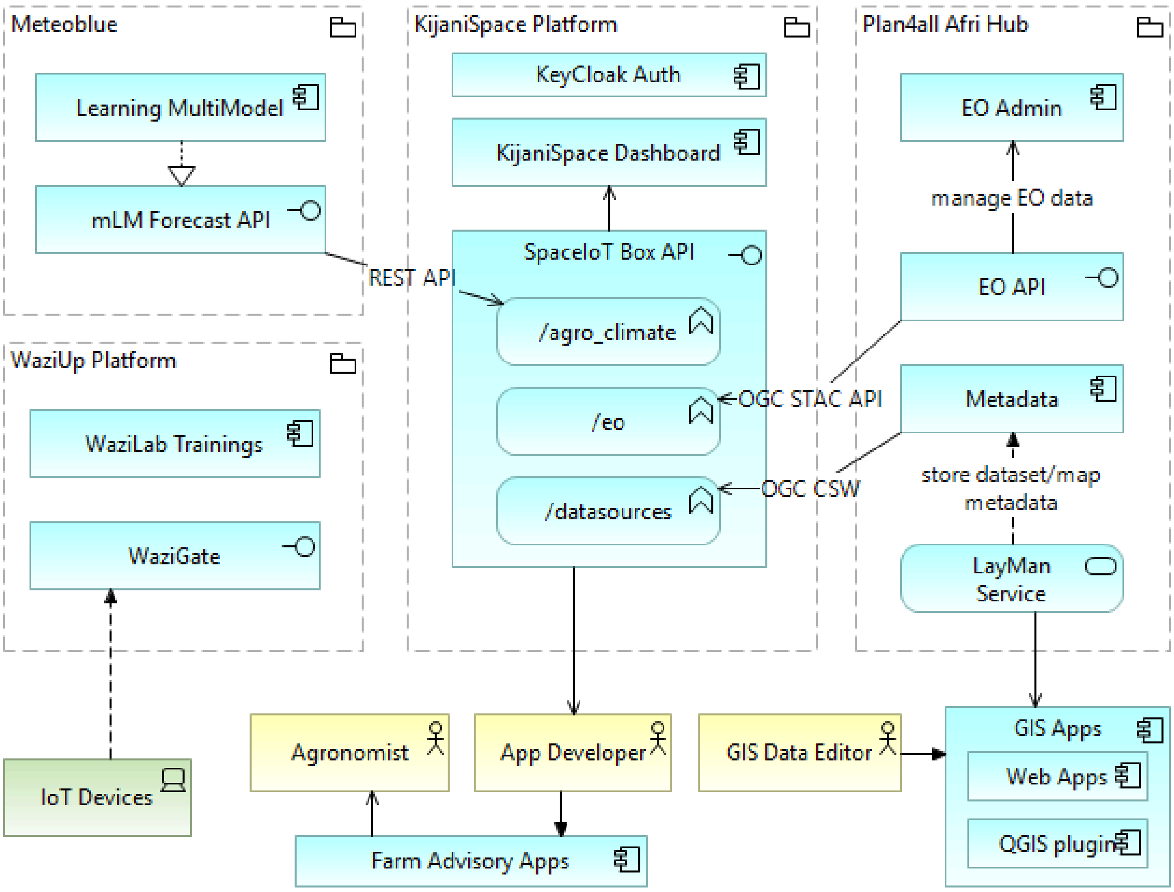


Figure 11: Overview of the KijaniSpace API and back-end services

4.2. Back-ends

The agroclimatic dashboard page and API uses the Meteoblue weather API[5][6]. The API is restricted to the lake region.

4.2.1. Meteoblue back-end

All queries to the API are point-based and can provide forecast data for every place world wide. Most packages have multiple differing variants which control the time resolution. For example the basic-1h package contains hourly data, whereas the basic-day package contains daily data.

- Basic package (weather data): hourly
- Agro package: hourly
- water data: daily

The following Table shows the parameters used to query the Meteoblue back-end.

Table 9: Query parameters on Meteoblue back-end:

Parameters	Values description
forecastDays	integer [0 .. 5] To limit the number of days which are returned set this query-parameter to the number of days which your application requires. If this parameter is not set the API returns as many forecast-days as it can, which might be more than 5 days. The resulting time-range of history_days and forecast_days must contain at least a day.
lat	number [-90 .. 90] Examples: lat=47.5596008300781 Latitude of point of interest, in degrees (WGS 84). Must be specified if no station id is given.
lon	number [-180 .. 180] Examples: lon=7.58860015869141 Longitude of point of interest, in degrees (WGS 84). Must be specified if no station id is given.

4.2.2. Info4Agro back-end

All satellite images are provided by the Info4Agro backend (provided by partner P4A)[7].

Data gets processed at the https://www.smartafrihub.com/eo-api/info which provides full functionality of the EO API.

[5] https://www.meteoblue.com/en/weather-api/dataset-api/
[6] https://docs.meteoblue.com/en/weather-apis/introduction/overview
[7] https://info4agro.com/eo-api/docs

4.2.3. Waziup IoT platform back-end

In order to let the users manage their IoT devices, the Space-IoT Box is giving access to the user to the Waziup IoT platform (WaziDev, WaziGate & WaziCloud).

The Space-IoT Box reads the IoT devices list of the user from the WaziCloud API. For this, it uses an access token, based on the user authentication. The device list is returned as a JSON structure, with the location of the device and its attached sensors. The sensor's values are then fetched. This data will be made available on the Space-IoT Box own API and displayed on the map.

4.3. Technology choices

The following Table gives the technology choices for the Space-IoT Box.

Table 10: Space-IoT Box technology choices

Technology	Comment/purpose	Benefits	Rationale
React + Typescript	React version 19. UI framework for state management, components and compiling dashboard	Easy to use, scalable. Safe for big projects and amazing long term support. Easy integration with libraries	Consistent UI library, scalable and handles big projects well. Long term support tools. TypeScript improves safety, scales with big projects well
Tanstack router	Routing library to dashboard pages	Type safety, easy file based routing, caching	Easy routing across components. File based routing
MUI	Component library for react	Amazing components, ease of use and integrations. UI complecent.	Based on Google's Material Design principles, known for its clean, modern aesthetics, and emphasis on user experience through meaningful motion and vibrant colors.
React query	API fetching library	Caching, easy state management, easy state synchronization across components	makes fetching, caching, synchronizing and updating server state in your web applications a breeze.

5. Conclusion

This deliverable has presented the development of the KijaniSpace "Space-IoT Box", including its dashboard, API, and main implementation choices. The platform provides a unified environment for accessing agroclimatic data, Earth observation products, data source information, training resources, and IoT-related services, while also exposing developer-friendly APIs to support the creation of Copernicus downstream applications for climate-smart agriculture in Africa.

The current implementation demonstrates that the Space-IoT Box is operational and ready to support the activities of WP4 and WP5. By combining Copernicus-based EO data, partner services, and IoT-oriented integration in a single platform, KijaniBox establishes a solid technical foundation for innovation, capacity building, and the development of practical services for smallholder farmers and other stakeholders in the Lake Victoria region. Further improvements will continue in the next project phases based on user feedback and application needs.

ACKNOWLEDGEMENT

This project has received funding from the European Union's Horizon Europe research and innovation program under grant agreement No 101180225

DISCLAIMER

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect host of the European Union or European Research Executive Agency (REA) Neither the European Union nor the granting authority can be held responsible for them.