Mitwelten *(Co-Worlds)*

Media Ecological Infrastructures for Biodiversity

Software Environment

This document describes the *Software Environment* developed for the research project MITWELTEN (*Co-Worlds*, www.mitwelten.org), deployed in the Basel area, funded by the Swiss National Science Foundation (2020-2024).

Contents

Overview

The figure below shows the system overview: measurement data is collected and recorded in the field using sensors, cameras and microphones, forwarded to the IoT and ML backend via local gateway, stored, analyzed and made available to apps and 3rd party services.

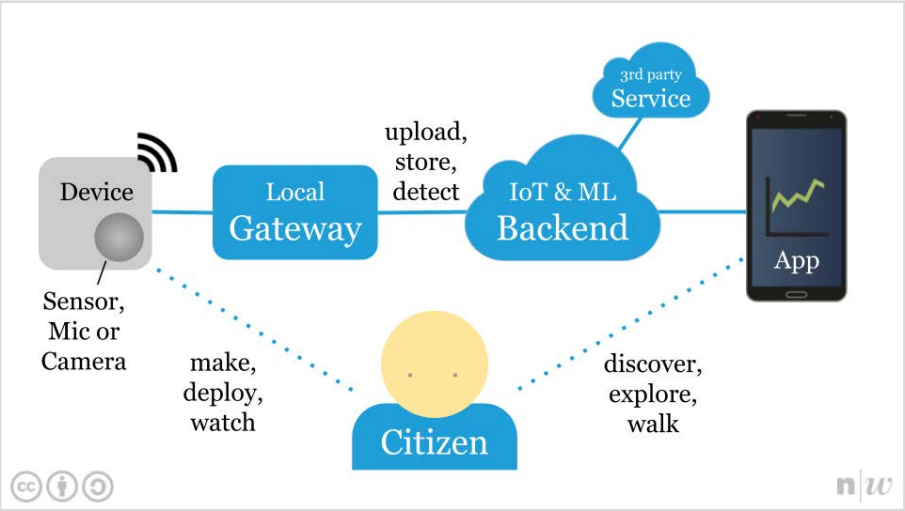

Fig. 1: *Mitwelten Software System overview*.

As an infrastructure for the experimental setups of the *Mitwelten* project, we developed a *Software Ecosystem* with components for sensor deployment (*Deploy*), monitoring (*Monitor*), ML-based analysis and detection (*Detect*), exploration (*Explore*), and map-based mediation (*Discover*). They are programmed in Python and Typescript, use a common Mitwelten API (*Data*), and are published as open source (see github.com/mitwelten). The ecosystem of services and custom, easy to use apps, supports the setup and maintenance of IoT sensor nodes, as well as the viewing, analysis and evaluation of the recorded data sets. The location aware web app *Walk* (github.com/mitwelten/mitwelten-walk-app) enables interactive discovery of datasets in the field.

Services

Data (Mitwelten API)

The *Data* software provides the *Mitwelten API* (data.mitwelten.org/api/v3/docs), the central repository of data, pictures and sound from the IoT devices (see Appendix B), ML analysis, research annotations and metadata like tags or information on deployments.

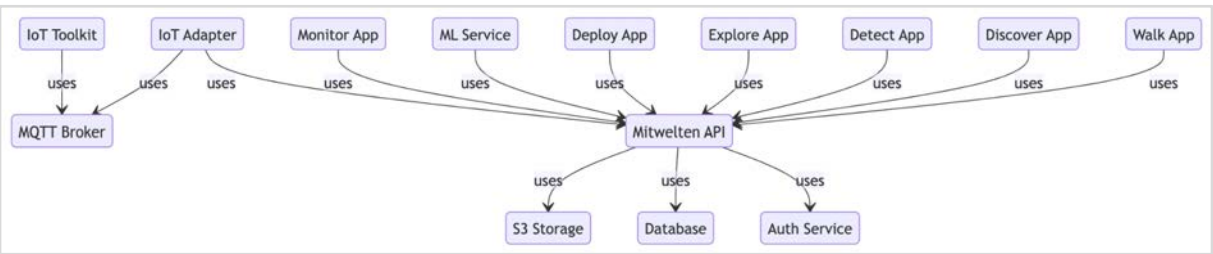

Fig. 2: The Mitwelten API in the center of the software ecosystem.

IoT Adapter Service

The *IoT Adapter Service* (github.com/mitwelten/mitwelten-mqtt-relay) ingests IoT sensor data into the Mitwelten API, from the MQTT broker or from the TheThingsNetwork LoRa backend..

ML Service

The *ML Service* (github.com/mitwelten/mitwelten-ml-backend) analyses sound and pictures. Results are stored in the database via the Mitwelten API.

Authentication Service

The *Authentication Service* manages users and roles for access to the Mitwelten API and enables single sign on for all services and apps.

Storage

Database

The *Database* is a relational database, exposed via Mitwelten API, see Appendix C.

S3 Storage

The *S3 Storage* offers file storage, exposed via the Mitwelten API, see Appendix C.

MQTT Broker

The *MQTT Broker* acts as a temporary storage for IoT sensor data.

Web Apps

Monitor (status dashboards)

The *Monitor* software (github.com/mitwelten/mitwelten-21hs-p7-tgi-backend) allows users to monitor the device and backend system status, e.g. the amount of collected data.

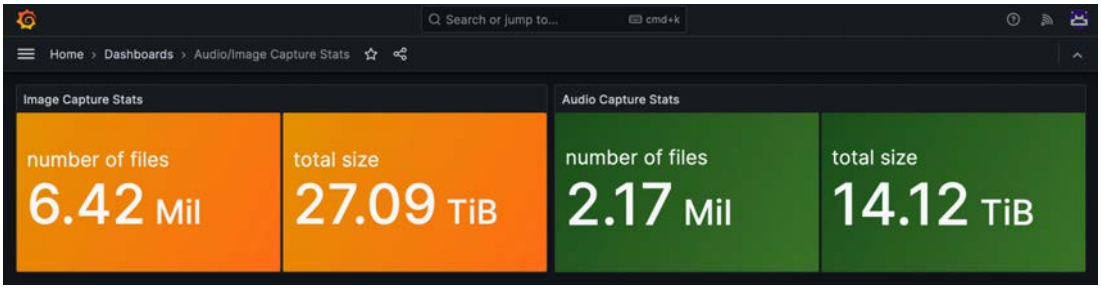

Fig. 3: A dashboard gives an overview of all stored audio and image files.

Fig. 4: A dashboard provides an overview of the stored audio and image files per sensor node and time interval and visualizes the amount of data using a spectral color code.

Another dashboard visualizes sensor activity using time-based diagrams so that data streams can be monitored online. This also allows the monitoring of the installations in the field, e.g. their battery status.

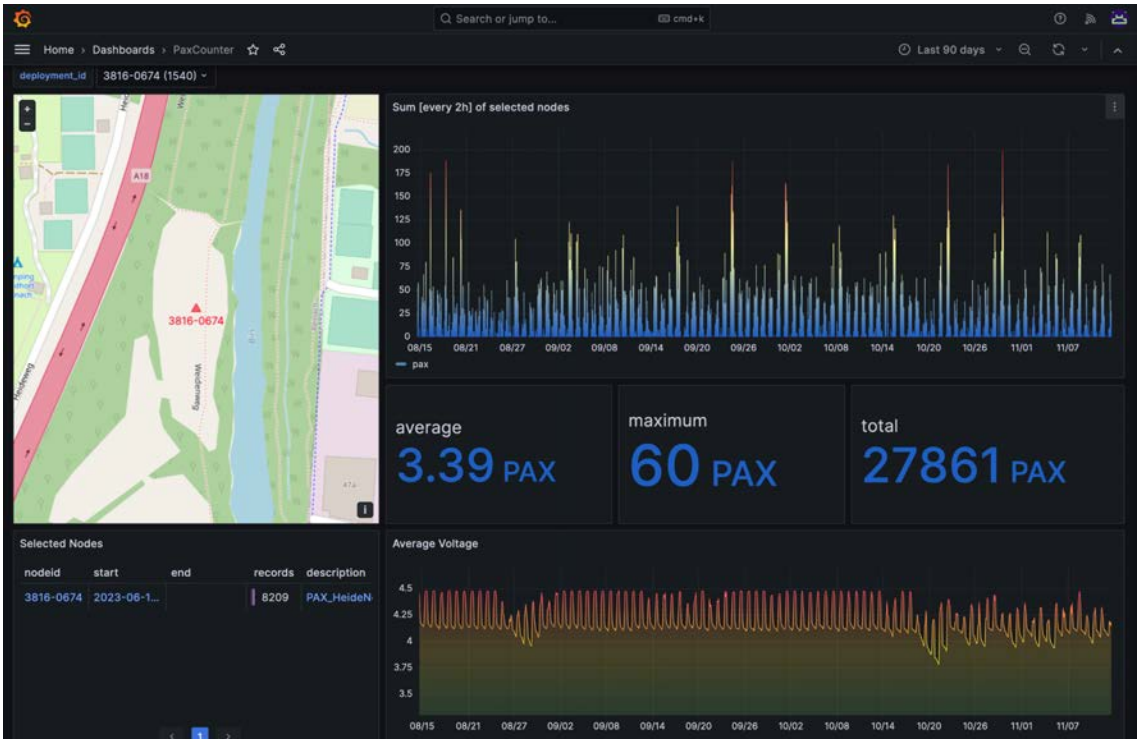

Fig. 5: This dashboard represents data collected by a Pax sensor. The node location is shown on a map and the date range can be chosen. It shows the Pax count and average voltage per time interval.

Detect (bird species visualization)

Detect (github.com/mitwelten/mitwelten-detect-app) is a web app to monitor the processing of audio datasets through our BirdNET pipeline. The dashboard also gives an overview of data inferred by the ML pipeline (all the detected bird species, independent of location and filter).

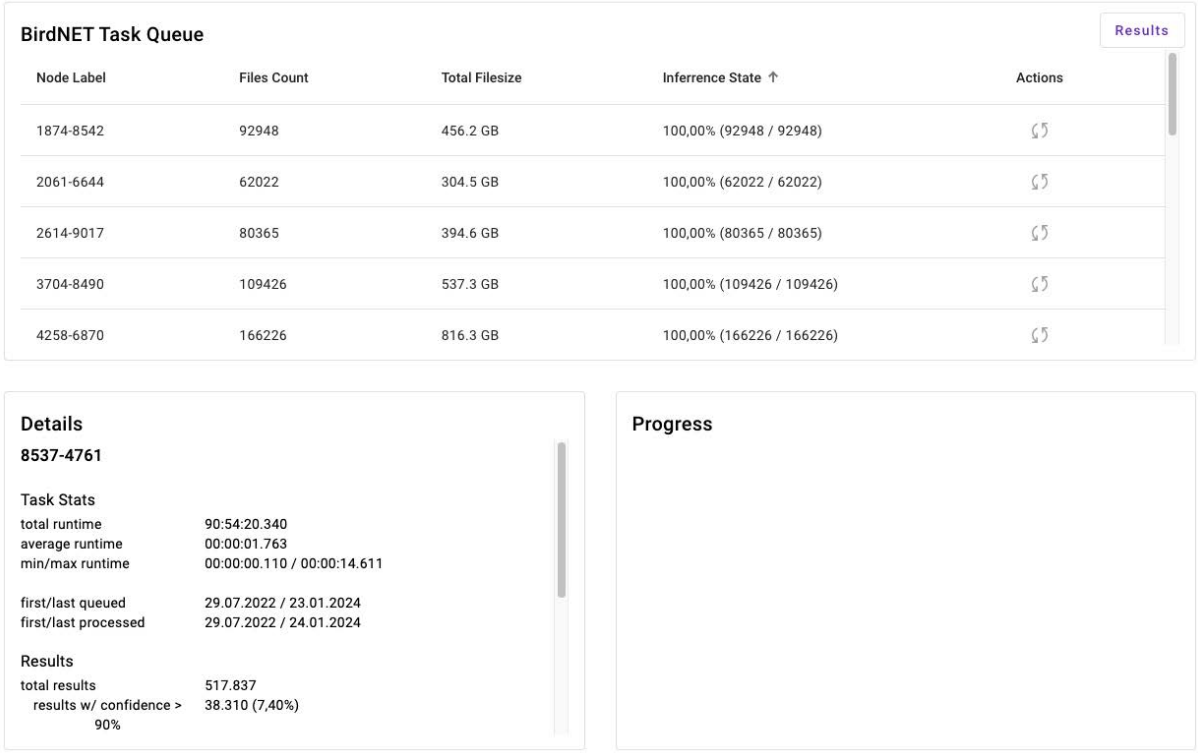

Fig. 6: This dashboard allows control and monitoring for ML processing of audio datasets.

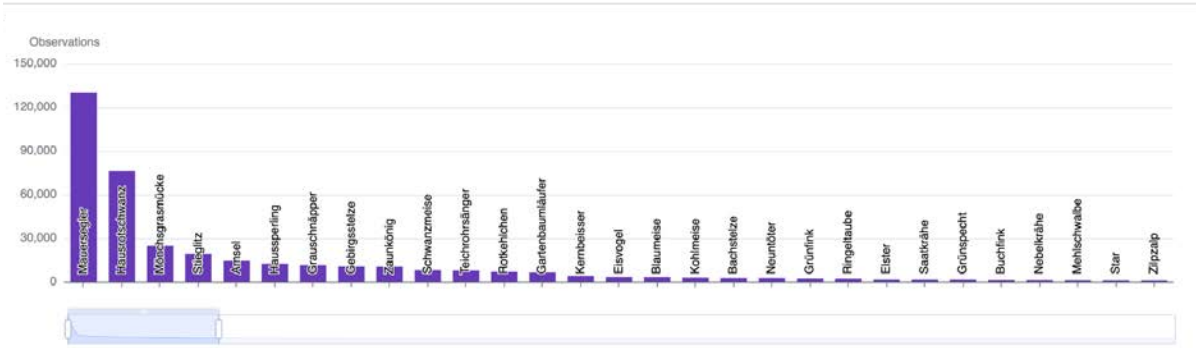

Fig. 7: This dashboard ranks all identified bird species according to the number of bird calls detected.

Explore (data exploration)

The *Explore* app (github.com/mitwelten/mitwelten-explore) is used for exploratory data analysis. Collected data sets or publicly accessible data streams can be selected, viewed, compared and annotated. The software also allows to validate identifications with citizen

science data collected through the web-based inventory Global Biodiversity Information Facility (www.gbif.org) and to put it into context with weather data collected from cantonal weather stations (e.g. wetter-binningen.ch).

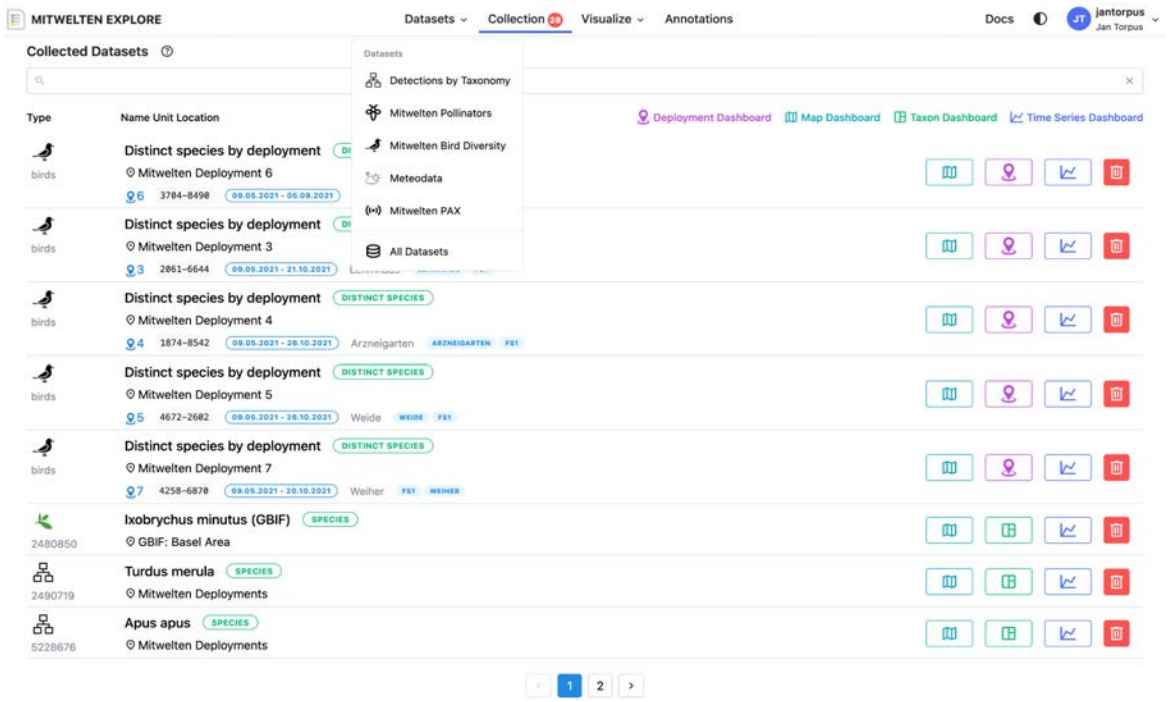

Fig. 8: This screenshot of the Explore interface shows how different types of datasets can be chosen from the dropdown menu Datasets, to become listed in the Collections. With the dropdown menu item Visualize different data visualization modes can be chosen and up to 7 datasets can be compared and correlated. With Annotations personal annotations can be assigned.

The location and duration of interest and the machine learning parameter confidence can be adjusted. Different types of visualizations, such as maps, graphs, or pie charts, make the data sets more accessible for exploratory data analysis.

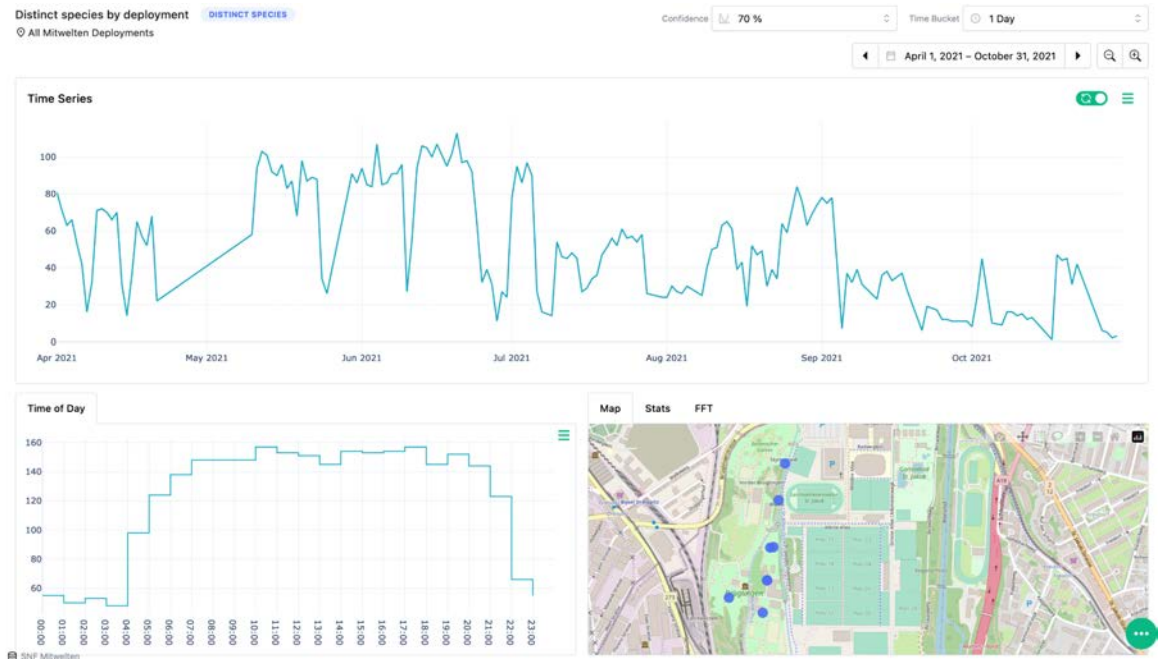

Fig. 9: This screenshot of the Explore interface shows a selected dataset by means of time-based diagrams and maps. Users can adjust their search by adapting time span and selecting the ML confidence factor.

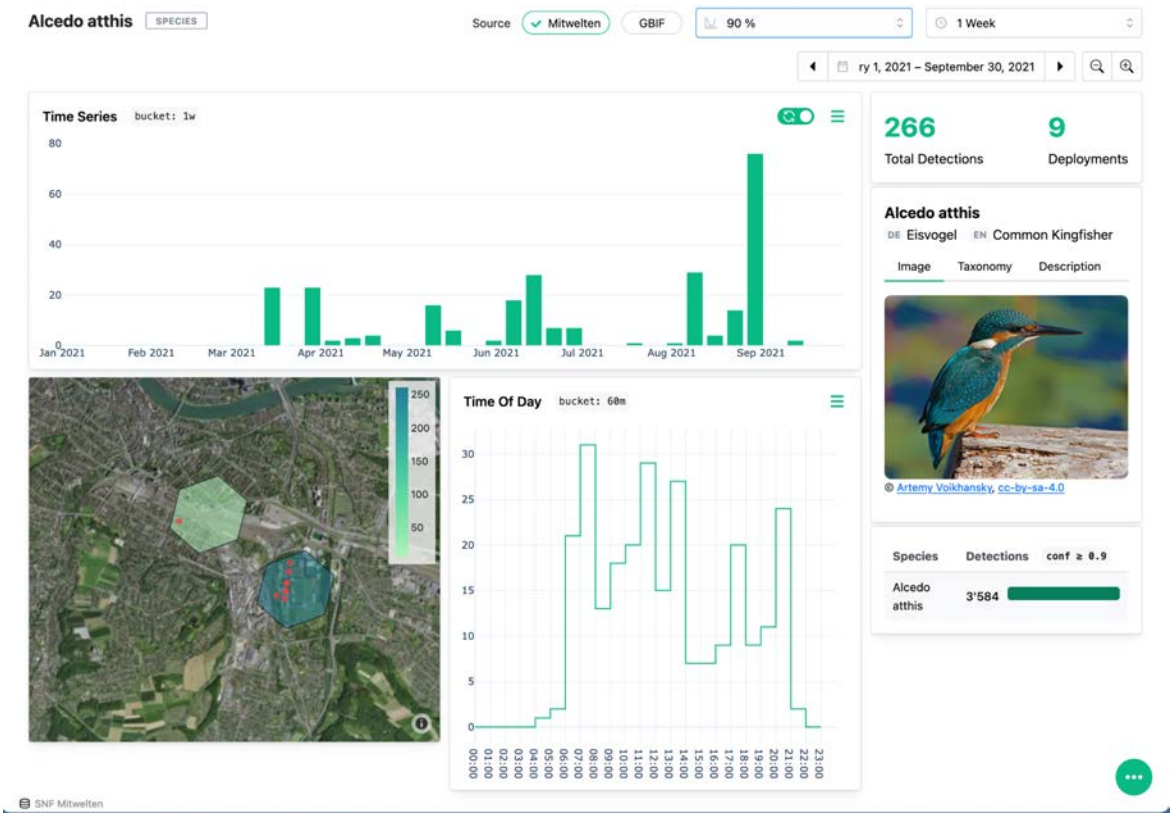

Fig. 10: The collected data can also be displayed according to taxonomies. This screenshot of the Explore interface shows the presence of the kingfisher in the case study Reinacher Heide from all audio-loggers during the timeframe January 1 to September 30, 2021, with ML confidence 90%. Also the presence per day time is displayed and general information about the species is provided. The button GBIF allows comparison with the citizen science reporting platform https://www.gbif.org.

Discover (data visualization)

The *Discover* app (github.com/mitwelten/mitwelten-discover-app) shows results of the field studies in an easily understandable way to the public. Users can select the data of interest by location, time span, sensor type, and type of medium. Besides the automatically geolocated representation of sensor data, it includes backend functions that can be used to upload information, e.g., about the cultural and operational context of the location, snippets of interviews displaying standpoints of the involved actors (e.g. gardeners, rangers, environmentalists, recreationists, etc.) and correlations amongst those actors, in different media formats such as text, sounds, and images.

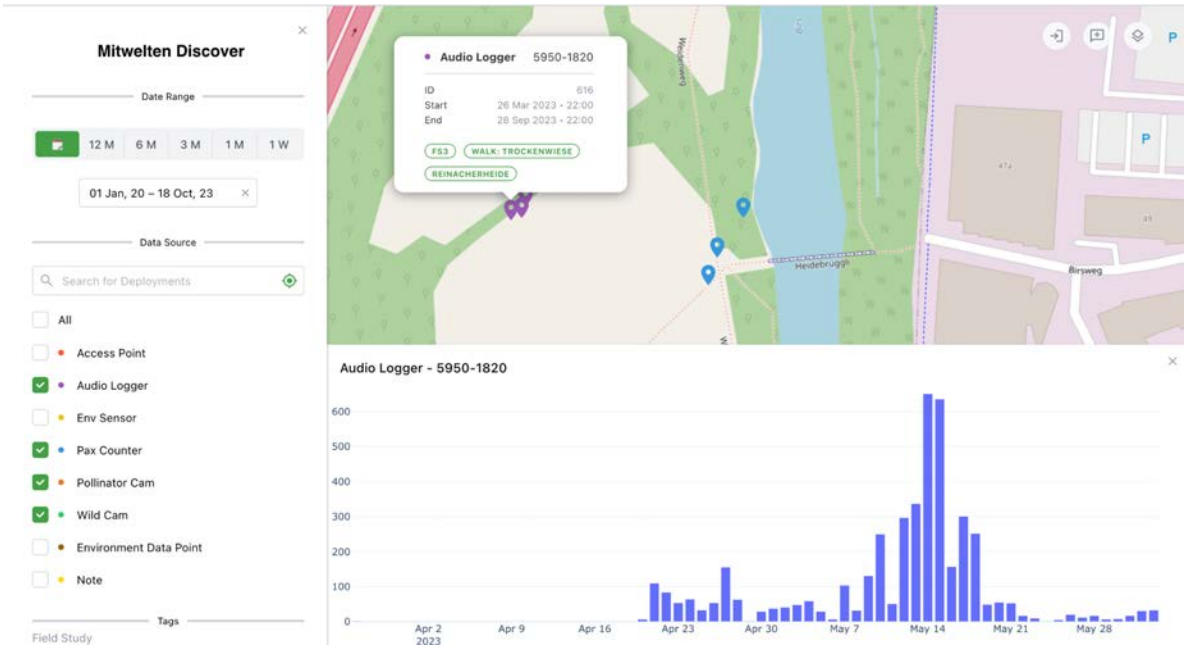

Fig 11: The Discover interface enables the selection of the desired data stream filter, the map-based selection of the sensor node, the selection of the time span and the visualization form.

Mobile Apps

Progressive Web Apps

Our mobile apps are PWA (developer.mozilla.org/en-US/docs/Web/Progressive_web_apps) which do not require app store approval for installation but still allow accessing smartphone features like GPS once a user gives permission.

Deploy (deployment manager)

The *Deploy* app (github.com/mitwelten/mitwelten-deployment-manager/) supports the deployment of the *IoT Toolkit* sensor devices. When a device is set up for a field experiment, a deployment record is created that assigns a label to the device, metadata such as type, geolocation, time span, tags, and a short description of the location can be added manually.

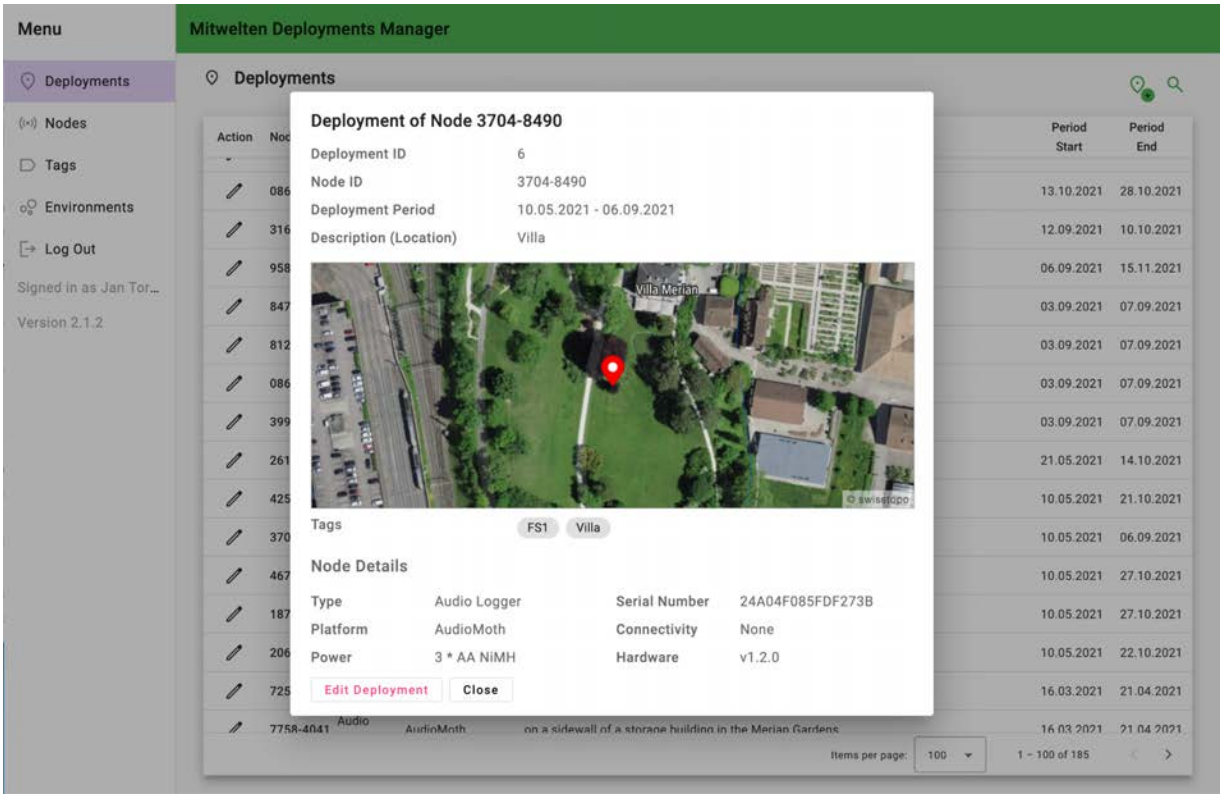

Fig. 12: The Deploy interface for sensor logging (example of audio-logger in the Merian Gardens).

Walk (interactive discovery)

The *Walk* app (github.com/mitwelten/mitwelten-walk-app) enables interactive discovery of datasets in the field.

Schweizerischer Nationalfonds

Fachhochschule Nordwestschweiz
Hochschule für Gestaltung und Kunst

Fachhochschule Nordwestschweiz
Hochschule für Technik