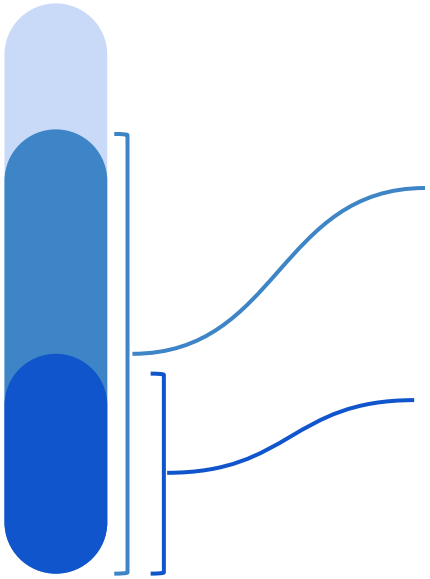
Agriculture is essential.
but...

Global annual fresh-water consumption



70% **Agricultural water consumption**

40% **Water wasted in agriculture***

*according to coolfarm.org

Every garden is unique. Its care should be too.

Unexpected weather events

Unknown local soil properties



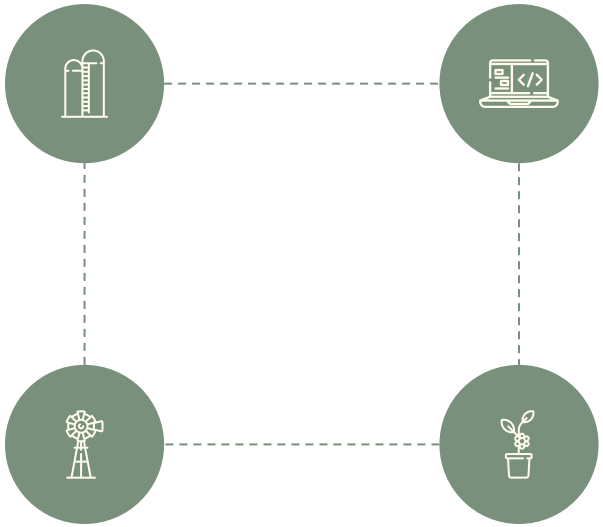
Human biases lead to irrigation errors

Nobody is perfect.
We know that.

FLoRaTech

Smart garden management

Monitoring of soil conditions and weather conditions



Automated irrigation by AI

Save water and optimize irrigation

Real-time notifications

Real-time garden monitoring and notifications

Complete modularity

Add and remove sensors easily, based on your needs

We hear you.
How can all this work?





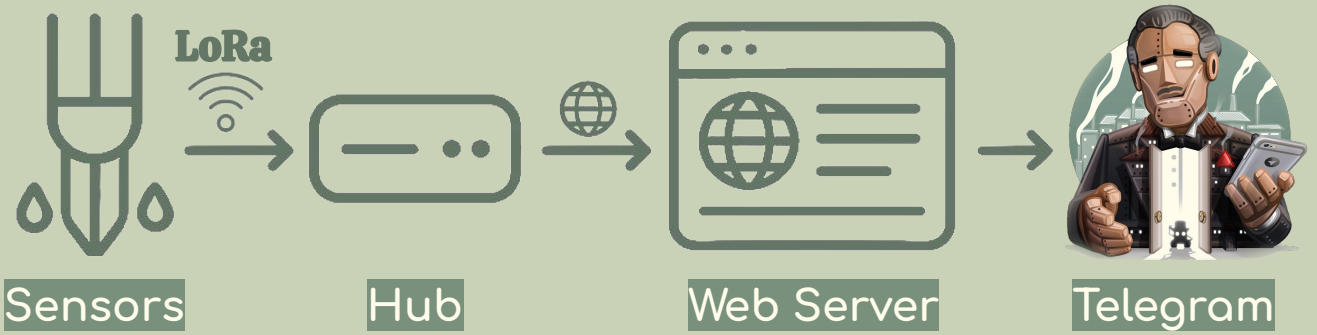
It works like magic.

No, it's just FLoRaTech

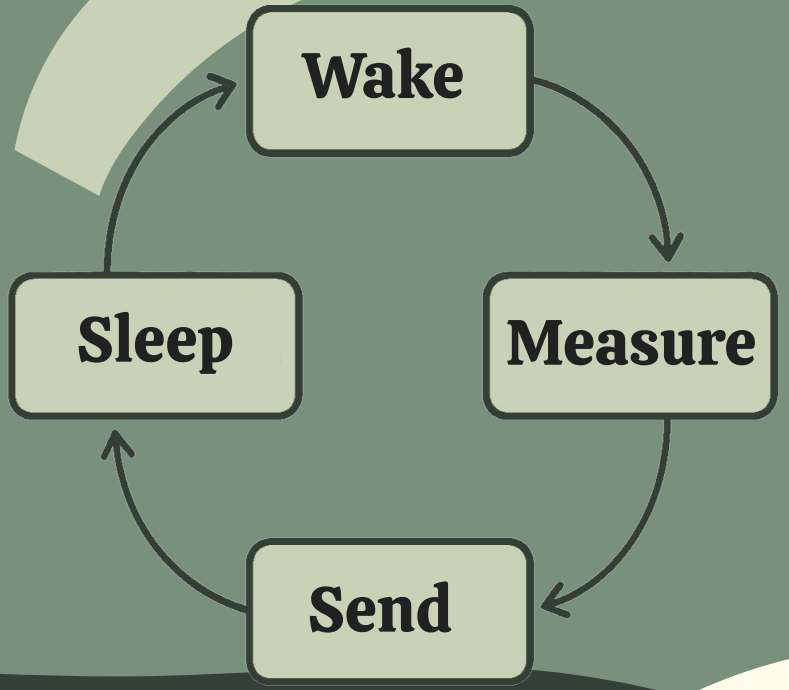
FLoRaTech

Smarter Water, Greener Future.

Architecture Diagram



Long-Lasting Sensors



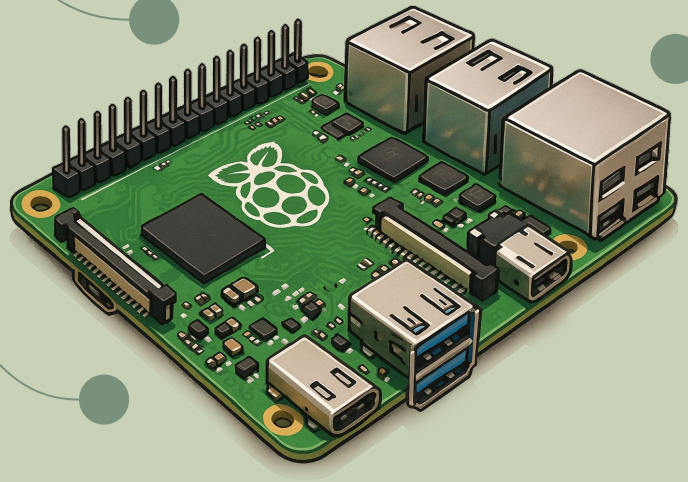
The Heart of the System

Safer

Controls when to irrigate to prevent harming due to climate conditions

Dynamic

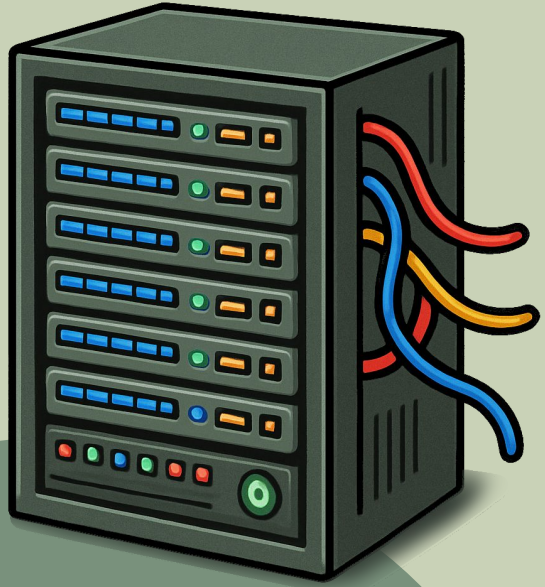
Dynamically configures the local infrastructure on the User needs



Efficient

Can operates while sipping power

The Brain



User in Control

With the Web Server's clear interface the user can manage and verify the system working correctly

Connected

Third party APIs allow precise weather forecast and geocoding

Powered by AI

An AI model precisely predicts how much water is required

Now let’s zoom in on how each piece is built and powered

Table of contents

01 **Sensors**
To sense moisture and actuate pumps

02 **Hub**
Connecting all sensors to the cloud

03 **Django**
A user friendly platform connecting everything

04 **Weather**
Predictive model and daily water computation

05 **Telegram bot**

01

Sensors

Low level components
and LoRa
communication

Garden



Hub



Sensor



Sensor



Sensor



Actuator

Why LoRa?

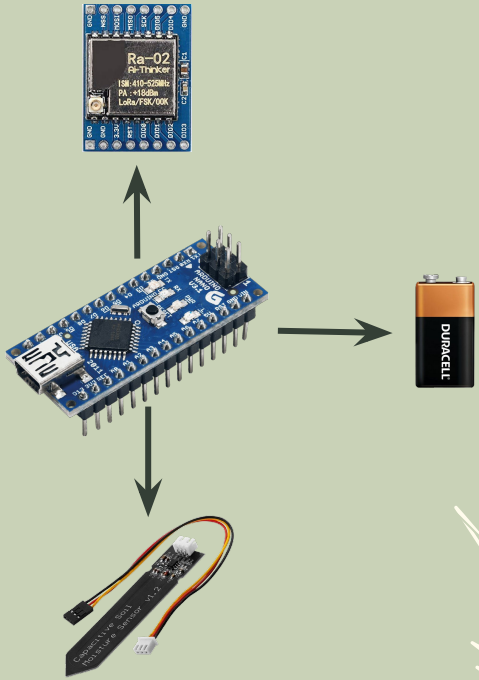


- Long Range of Transmission
- Low Energy Supply
- Affordable Components
- Good Signal Penetration
- "Private" Network

The Hardware – Sensors







The Hardware – Sensors

Arduino Nano

- Small
- Cheap
- Low Power Consuming

SX1278 Transceiver

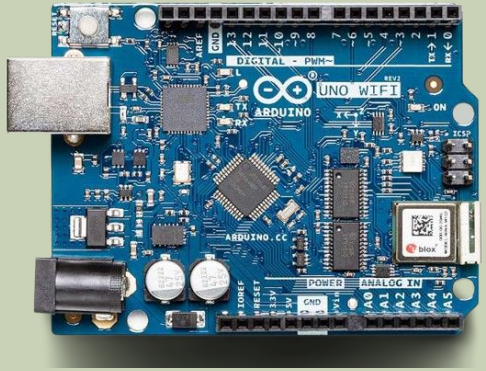
- Good Compatibility
- Resilient to Interferences
- Cheap
- Versatile

AZ Delivery Hygrometer Module V1.2

- Solid
- Simple
- Cheap
- Capacitive

The Hardware – Actuators







The Software

RadioLib

- Flexible - Highly Configurable
- Good Documentation with Working Examples
- Compatible with Everything
- Interrupt Function Support (better than LoRa.h)

- **CAD Functionalities**

The Software

LowPower.h

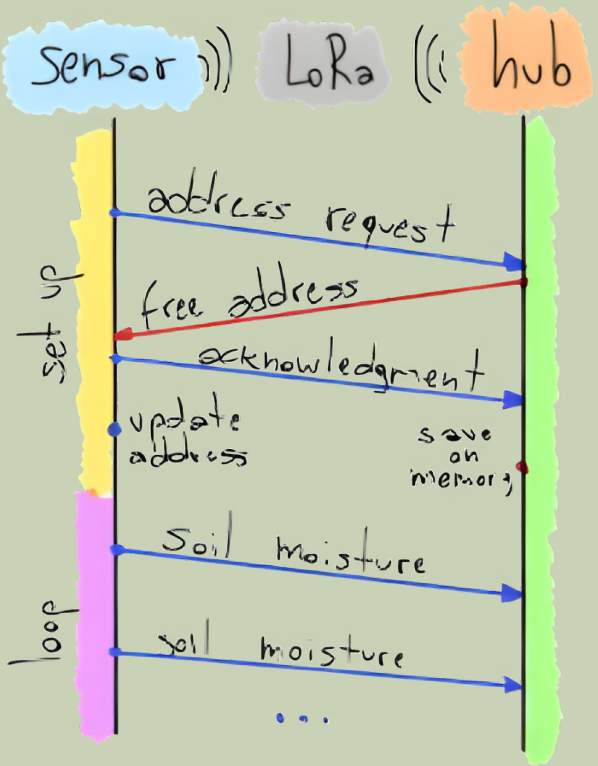
- Deep Sleep Mode
- Power Saving

EEPROM.h

- Read and Write on Arduino Register
- Keep Information after Shutdown or Reset

Sensor-Hub Pseudocode

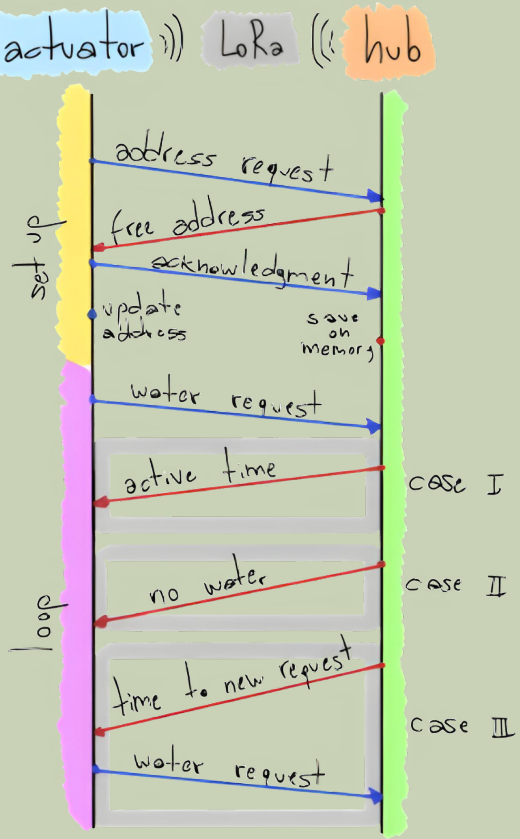






Actuator-Hub Pseudocode

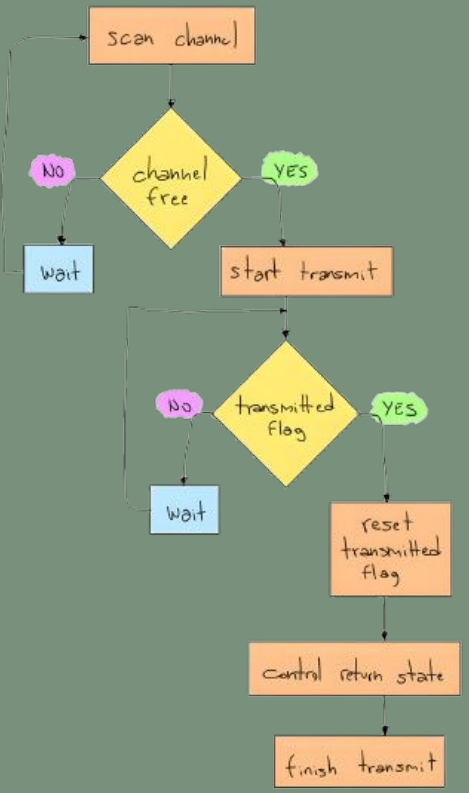






Channel Activity Detection

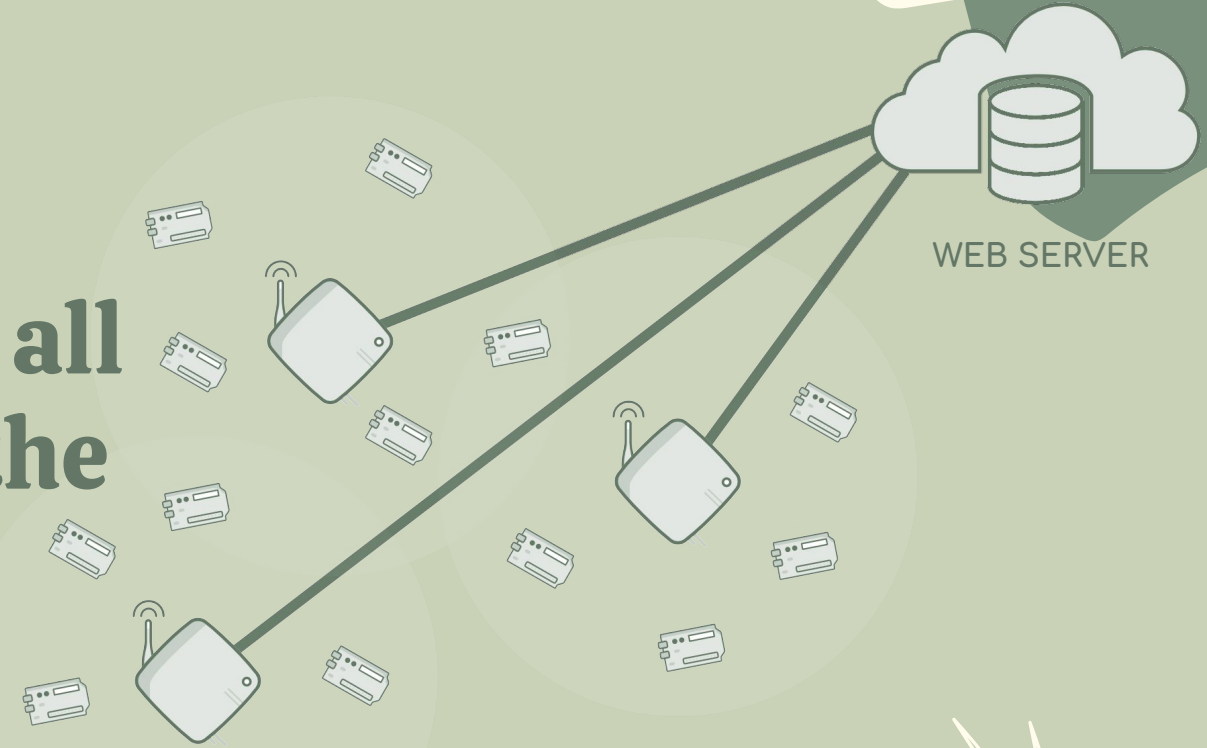
- Avoid Packet Collision
- Scan Preamble, Not Entire Packet
- Reduce Lost Packet



02 Hub

Raspberry Pi & LoRa

Connecting all sensors to the network...



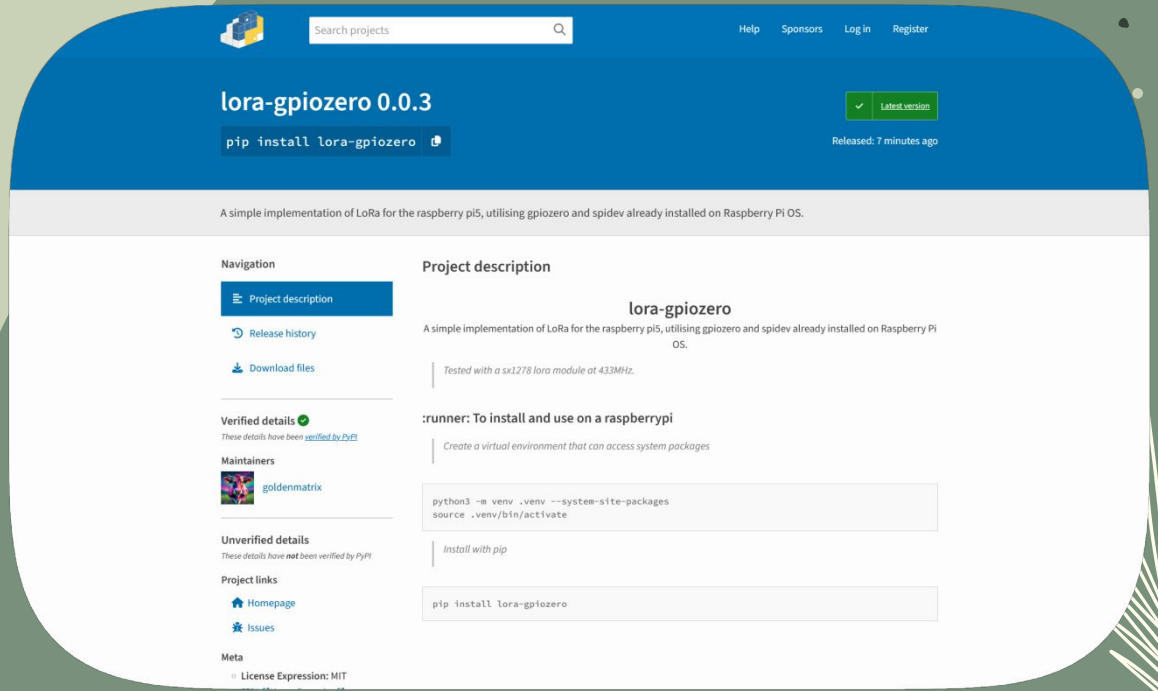
A Minor Hiccup

LoRa

Radio communication

GPIO Zero

Installed by default in the RaspberryPi OS



Under the Hood: asynchronous programming

Efficient Task Handling without sleep()

Traditional **sleep( ) function** can lead to performance bottlenecks by blocking the entire execution thread.

With asynchronous tasks and a simple multithreading implementation the main thread remains responsive, efficiently handling time-based callbacks without missing any incoming data.

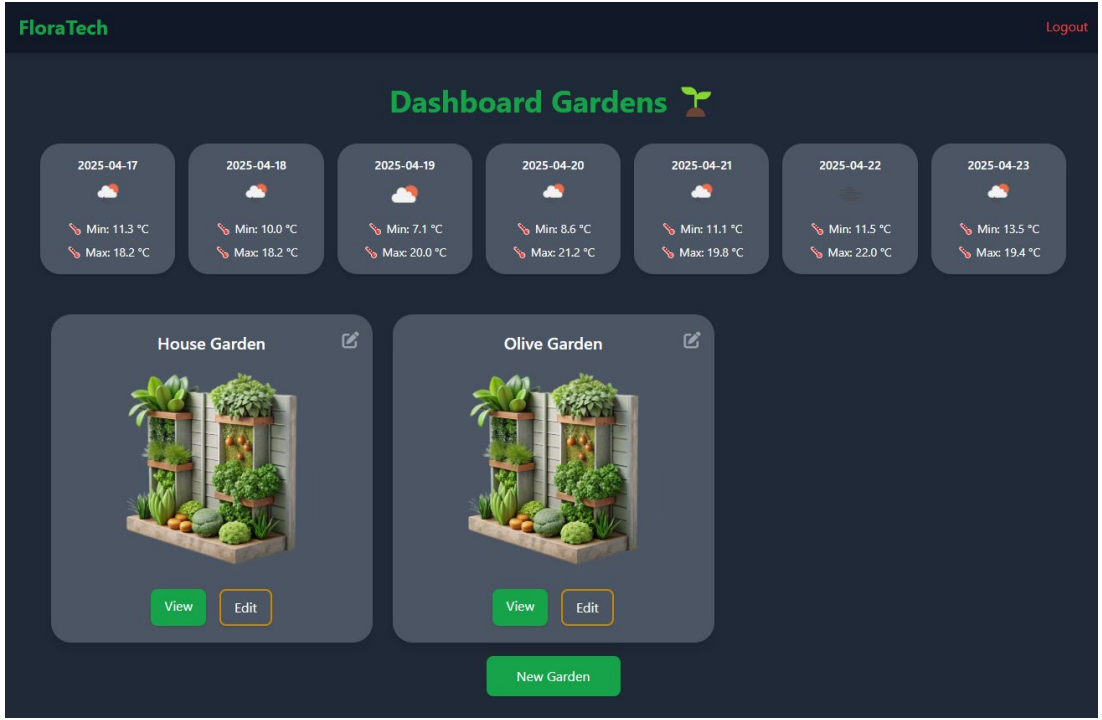


Django 03

1. Dashboard

Through the dashboard you can view the gardens you are managing and a preview of the weather forecast for the week.

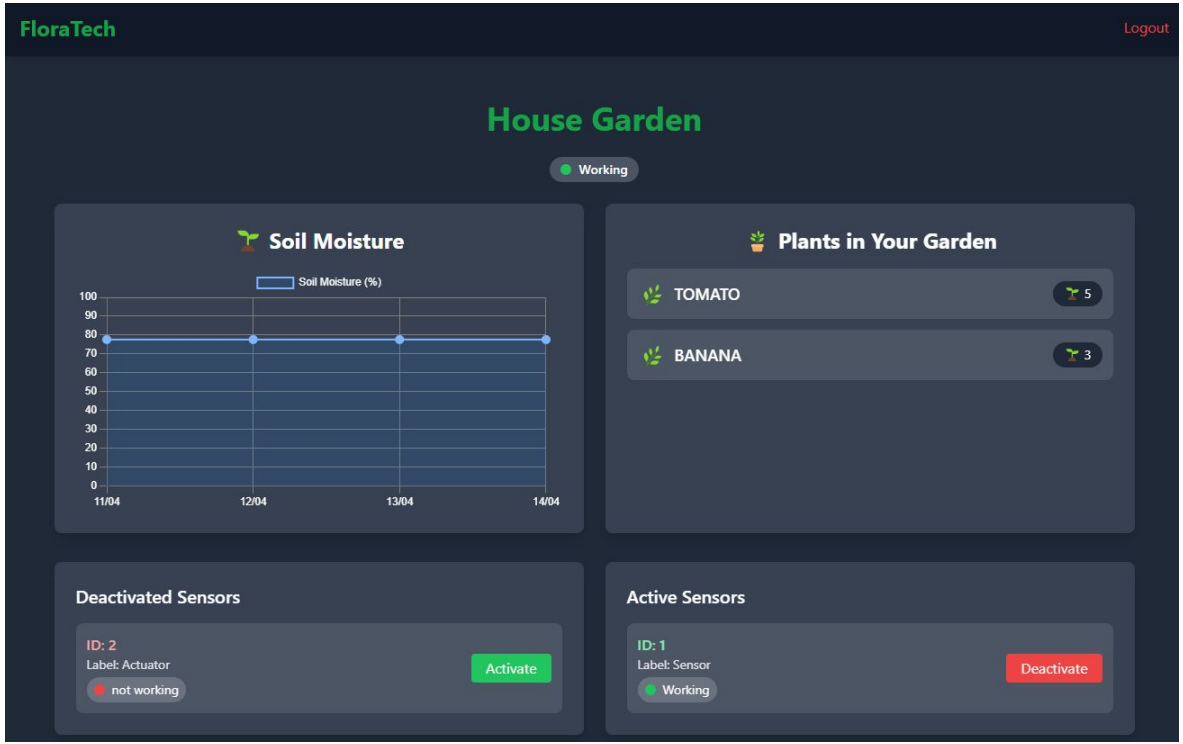
You can change the parameters of the garden such as the crops, the location, etc.



2. Garden

Through the garden page you can see various statistics, like soil moisture and crops in the garden.

You can also choose which sensors to activate and deactivate.



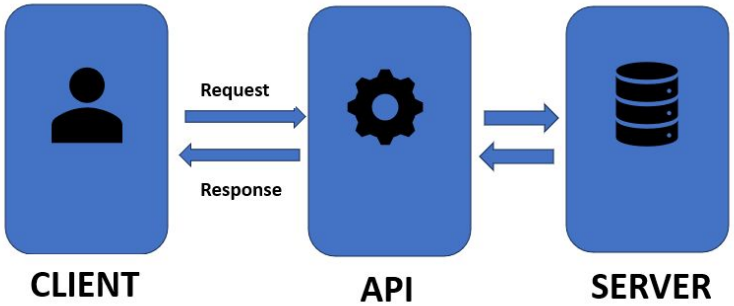
3. API

This is where the core of the communication between the Hub and the Web Server is located.

API calls made from the hub send data in json format that are processed by the web server.

If necessary, the web server obtains the necessary information via third-party APIs.

Finally the web server processes the data and returns a response to the hub.



4. Models and DB

The database contains a digital and simplified representation of the garden.

It's managed by django and is structured in the following way for each user. (see next slide)



4. Models and DB

Raspberry

At the time of configuration, the raspberry is manually entered into the database by our team and binded to an account.

It can manage multiple gardens.

Garden

Every garden is binded to a raspberry.

The garden model store the soil moisture data detected by the sensors and the various settings like geodata and crops.

Sensor

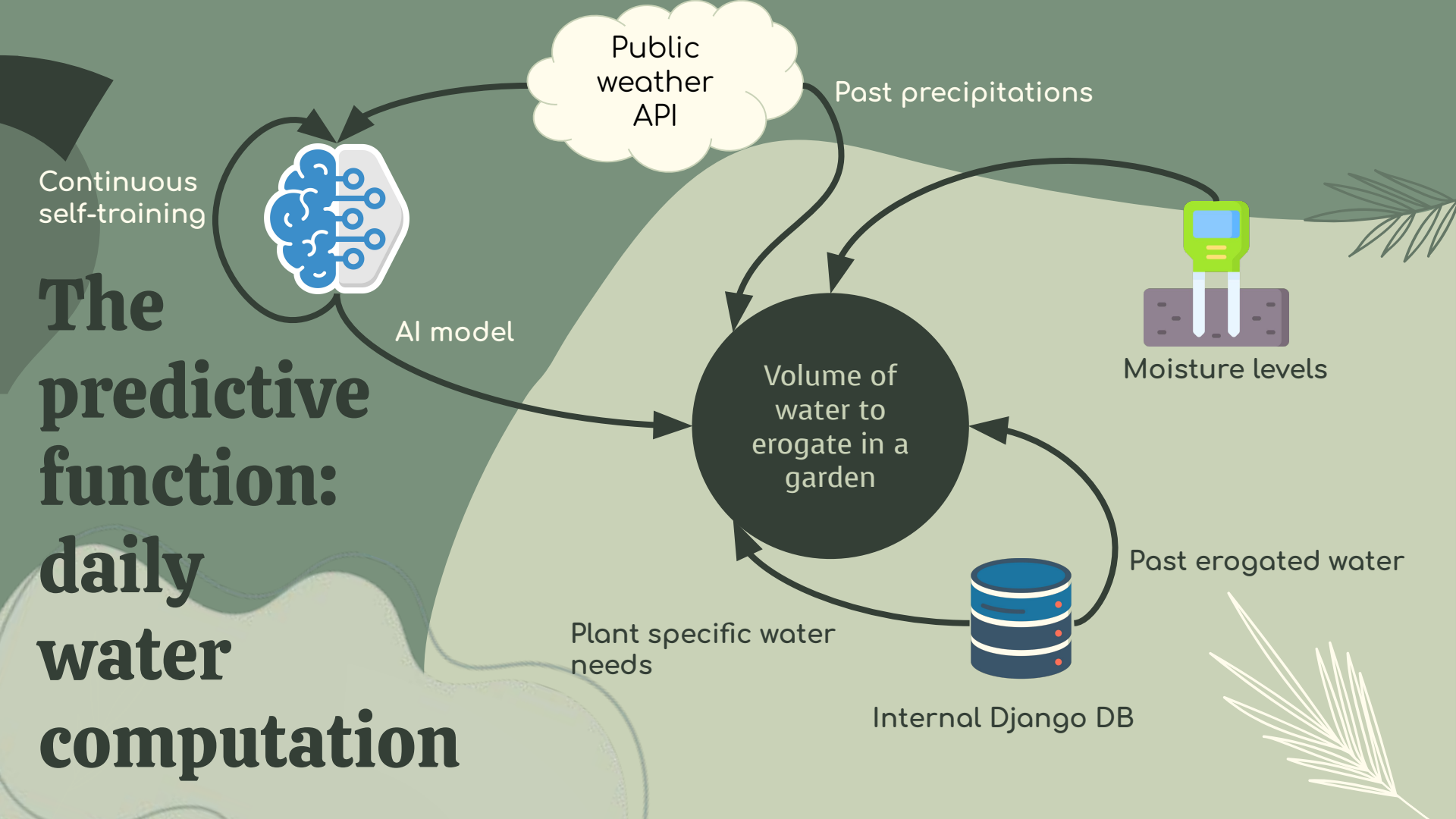
Every sensor is binded to a garden but can be unbound and relocated in a different garden.

Every sensor is stored as a “soil moisture sensor” or a “pump actuator”

04 Weather

Predictive model and
daily water computation

The predictive function: daily water computation

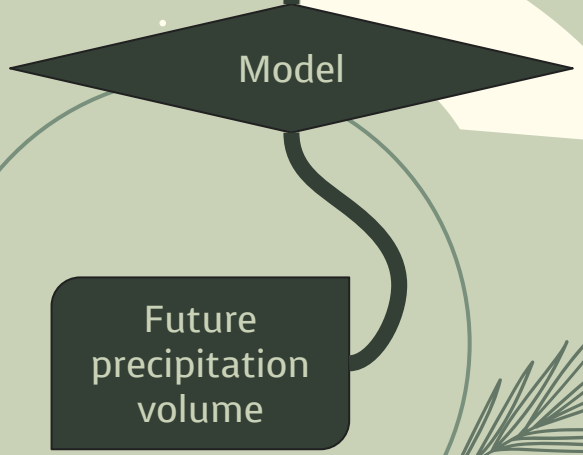


The AI model:

Input data

API data for day x in locality A
[weather_code, temperature_max, wind_speed_max, daylight_duration, temperature_min, wind_gusts_max, et0_fao_evapotranspiration, wind_direction_dominant, …]

Average of the moisture data for location A



Model used: SGDRegressor

Predicted value: The quantity of expected precipitations for the x future day (mm/m^2)

Training data: the model was trained gathering data from a public meteo api.

Features: continuous self-training.

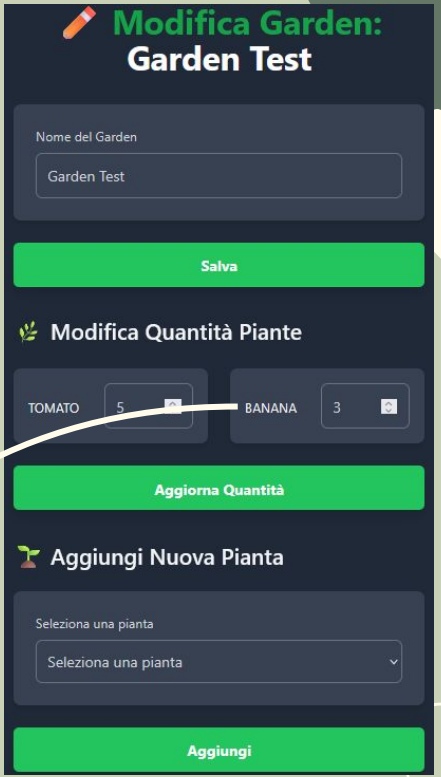
```
self.model = joblib.load(os.path.join(settings.BASE_DIR, 'model_parameters.pkl'))
```

```
self.model.partial_fit(df_api_transf,preds)
```

Garden specific data

Other elements that are taken into account:

- The typology of the plants into the garden
- Eventual anomalies in the values of the moistures
- Historical data of both natural and artificial water erogation
- Weekly temperature average

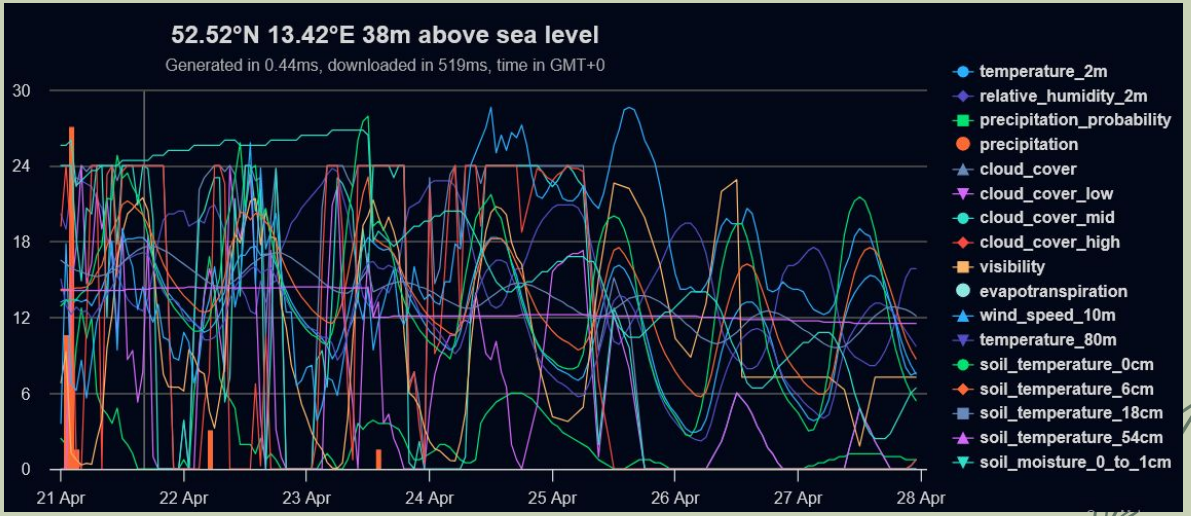


CROP TYPE	SOIL TYPE	REGION	TEMPERATURE	WEATHER CONDITION	WATER REQUIREMENT
BANANA	DRY	DESERT	10-20	NORMAL	8.75
BANANA	DRY	DESERT	10-20	SUNNY	10.25
BANANA	DRY	DESERT	10-20	WINDY	9.65
BANANA	DRY	DESERT	10-20	RAINY	0.75
BANANA	DRY	DESERT	20-30	NORMAL	9.85

Public weather API







Telegram Bot 05

Telegram Bot

A digital assistant that informs you about the status of the garden and notifies you in case of emergency.



Set up via webhook and embedded into the django ecosystem



Possible Updates & Improvements

Possible updates



Compute vision system

Possible updates



Dedicated Mobile App

Possible updates



Chat GPT API

Possible updates



Green Network



Thank You

floratech.leonardonels.com