

# INTEL-IRRIS

Intelligent Irrigation System for Low-cost Autonomous Water Control  
in Small-scale Agriculture



**Intel-Irris**



# WaziUp Technologies: Hardware, Software, Apps and Intel-IRRIS irrigation App



Intelligent Irrigation System for Low-cost Autonomous  
Water Control in Small-scale Agriculture

INTEL-IRRIS – PRIMA S2 2020 – PROJECT ID 1560



Felix Markwordt  
felix.markwordt@waziup.org  
Waziup e.V., Germany



# Structure of the Presentation

1. Introduction
2. WaziGate
  1. Hardware
  2. Our Software
  3. WaziCloud
3. WaziGate – Apps
  1. General information
  2. Jupyterlab
  3. Computer Vision
  4. Time Series Forecasting
4. Intel-Irris WaziApp (IIWA)
5. References

# Introduction

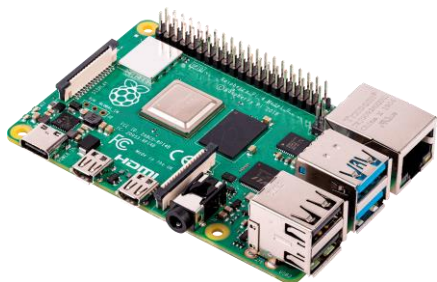
- My Name is Felix Markwordt



- Finished my master thesis about computer vision in late 2020
- I work for WaziUp since September 2021

# Hardware: WaziGate

- Raspberry Pi is a small single-board computer
- We use RPI 3 B+ & RPI 4



Raspberry Pi



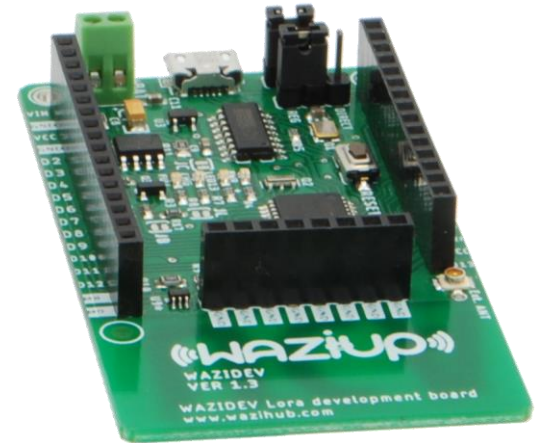
WaziHat



WaziGate

# Hardware: WaziDev

- Arduino pro mini with ATmega 328P
- 3.3V 8Mhz (low power consumption)
- Fitted for different applications:
  - WaziDev
  - WaziSense
  - WaziAct



WaziDev

# Our Software: WaziGate

- Read and display sensors and actuators values
- Connect to two different wifi, can act as a station and access point at the same time
- Remote management (not in the same sub-network, via cloud)
- Data upload with HTTP, MQTT or even SMS
- Possibility to mirror all your sensor data to a cloud
- you can host your own applications
- gateway can work without Internet connectivity

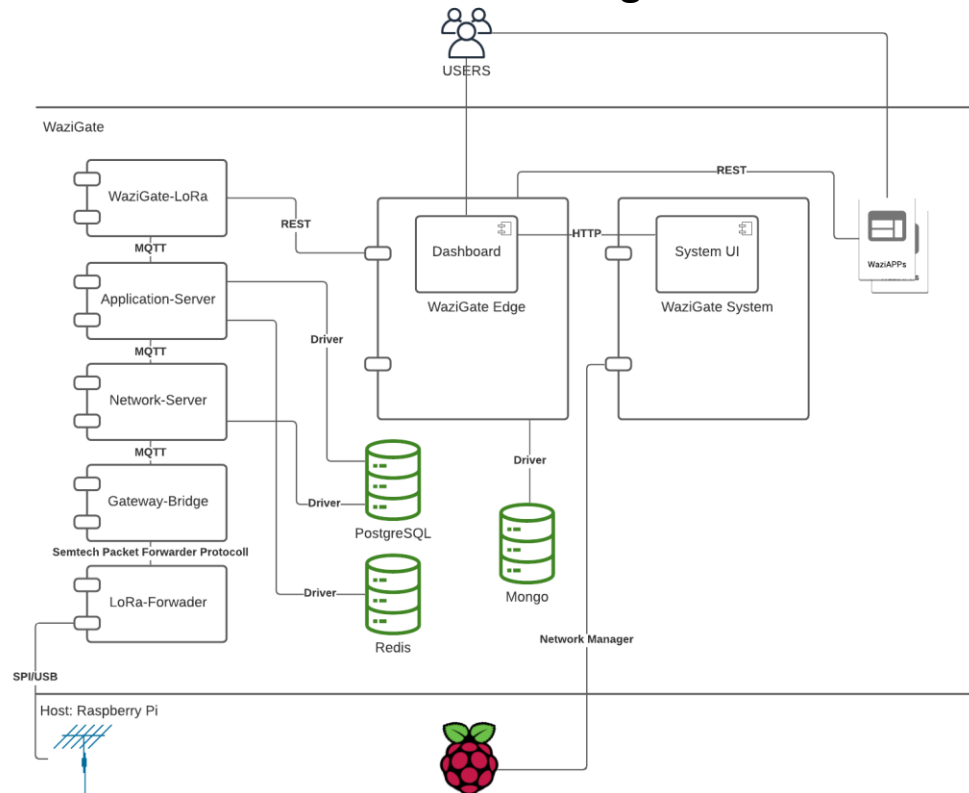
- Both WaziCloud and WaziGate provide an extensive API:
  - Allowing developers to connect their applications
  - Available in HTTP, MQTT and Web Sockets protocols

Actuators	
GET	/devices/{device_id}/actuators
POST	/devices/{device_id}/actuators
GET	/devices/{device_id}/actuators/{actuator_id}
DELETE	/devices/{device_id}/actuators/{actuator_id}
PUT	/devices/{device_id}/actuators/{actuator_id}/name
PUT	/devices/{device_id}/actuators/{actuator_id}/actuator_kind
PUT	/devices/{device_id}/actuators/{actuator_id}/value_type
PUT	/devices/{device_id}/actuators/{actuator_id}/value



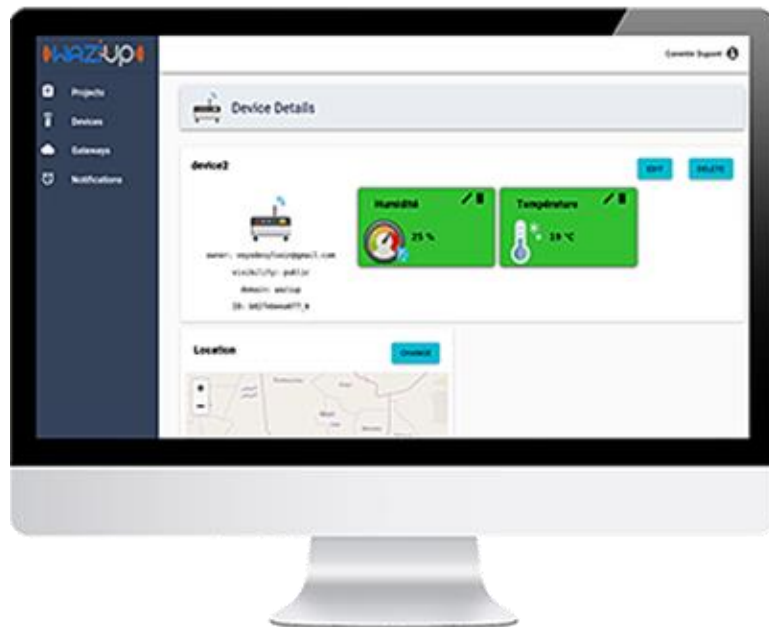
# Our Software: WaziGate

- Software of the gateway is setup as a microservice architecture
  - Independent components, that are to an extend interchangeable
- Frontend: ReactJS
- Backend: GO
- Database: MongoDB



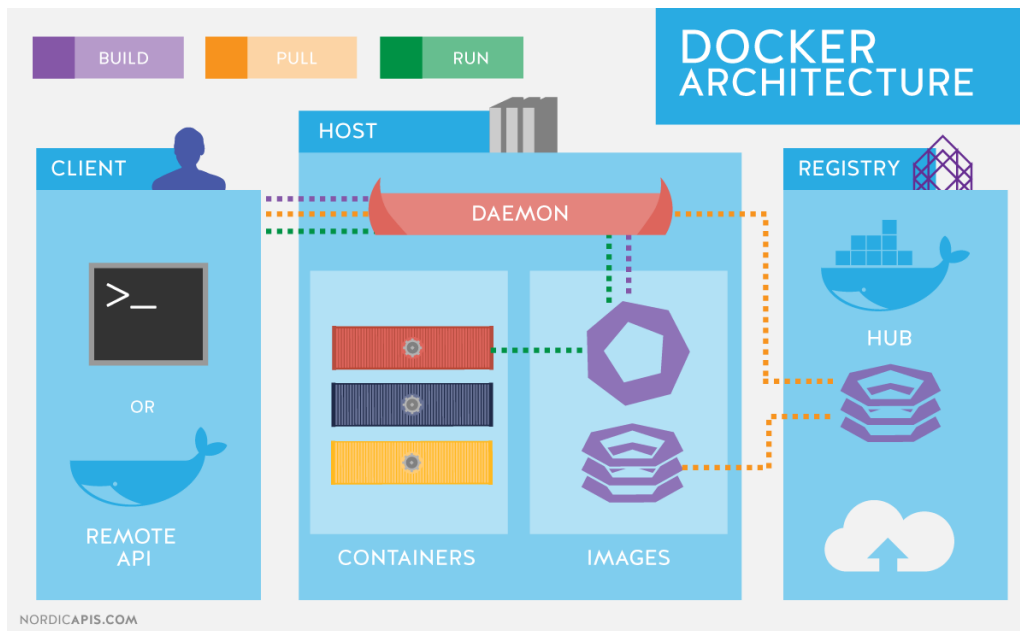
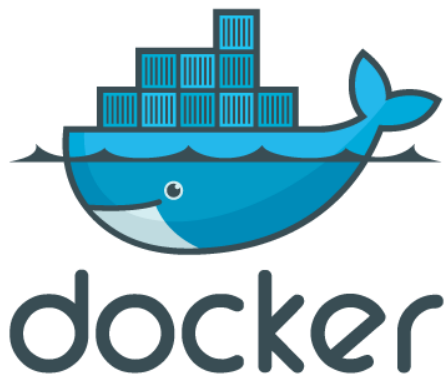
# Our Software: WaziCloud

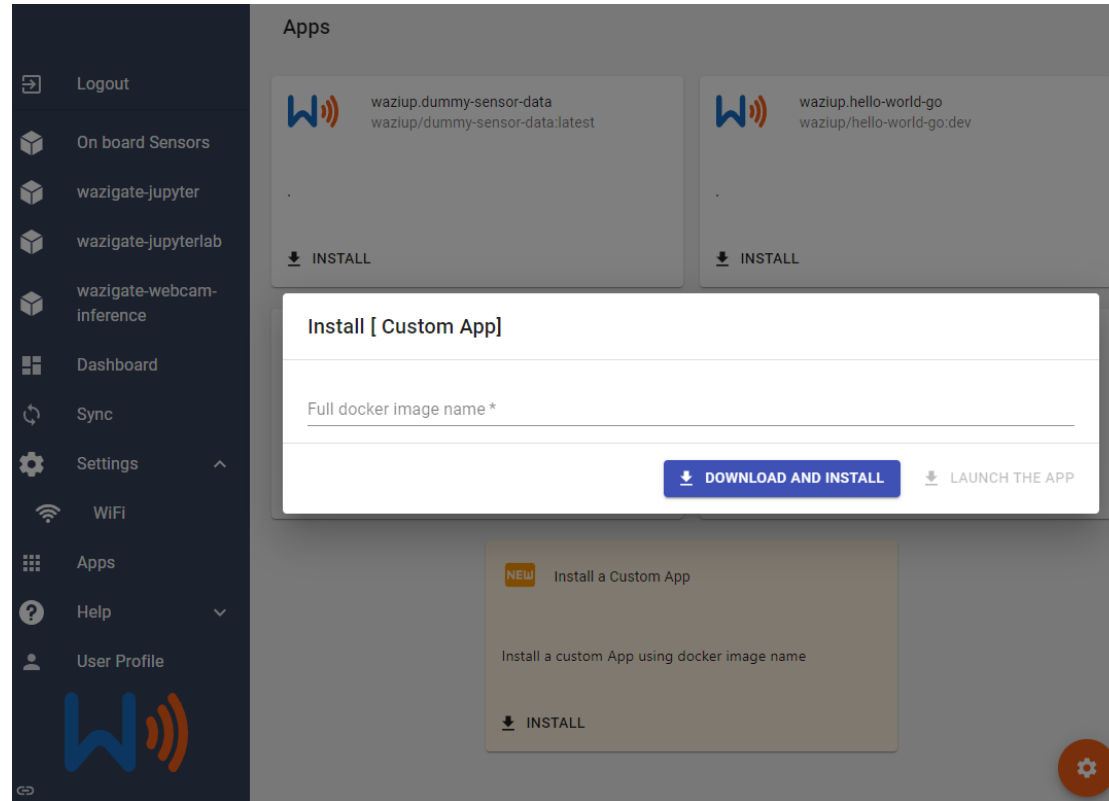
- Connect a physical gateway to the cloud, remote access to device
- Manage several gateways in one place
- Setup rules for actuation
- More features present, just an overview



# WaziGate - Applications

- All Apps run separate from the host operating system in an encapsulated docker container
  - Include all the software that is necessary



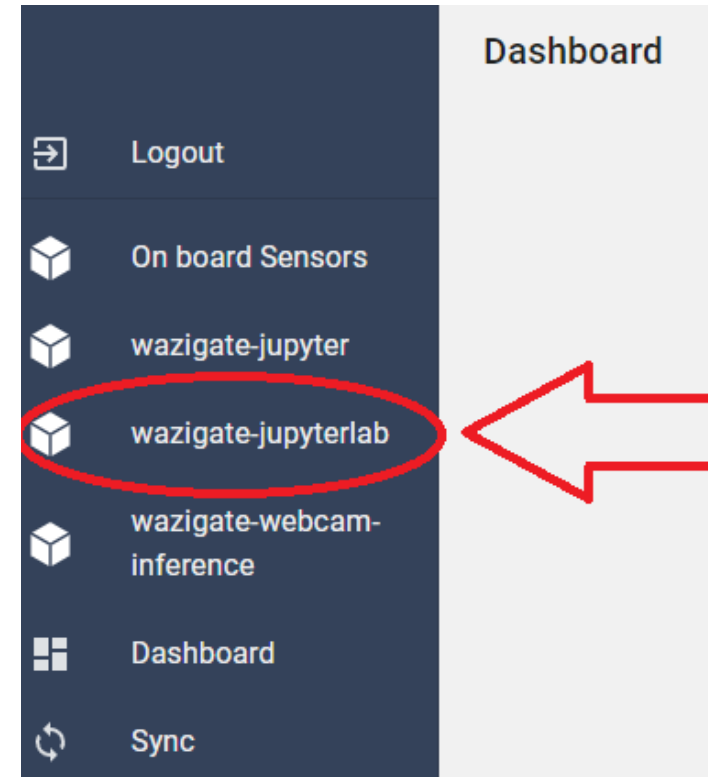
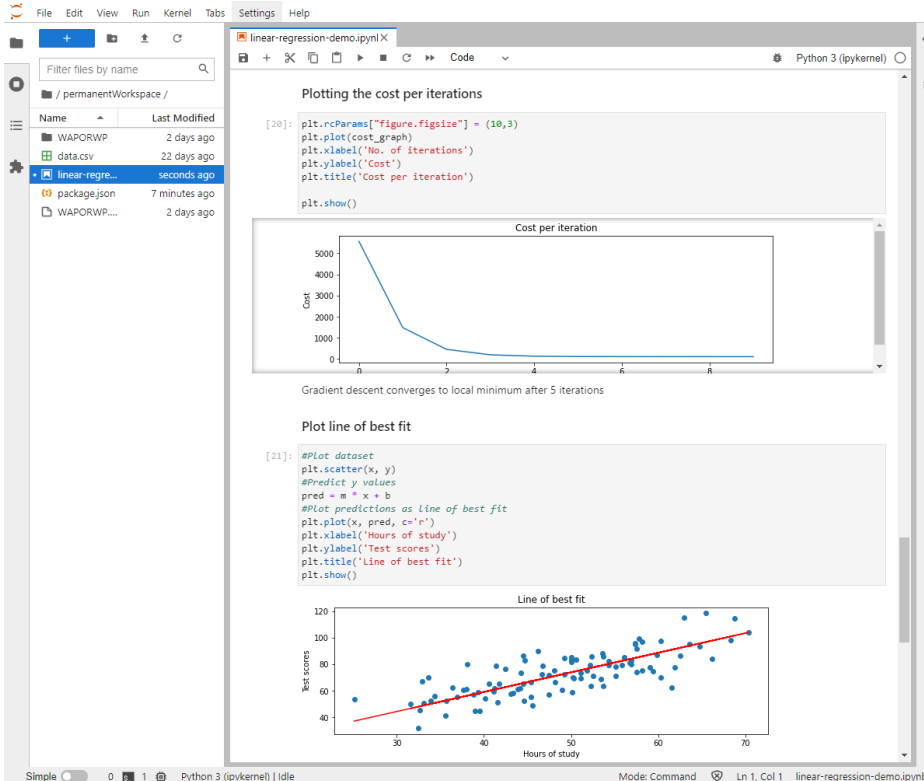


The screenshot displays the WaziGate Applications interface. On the left is a dark sidebar with navigation options: Logout, On board Sensors, wazigate-jupyter, wazigate-jupyterlab, wazigate-webcam-inference, Dashboard, Sync, Settings, WiFi, Apps, Help, and User Profile. The main content area is titled 'Apps' and shows two installed applications: 'waziup.dummy-sensor-data' and 'waziup.hello-world-go'. A modal window titled 'Install [ Custom App]' is open, featuring a text input field for 'Full docker image name \*' and two buttons: 'DOWNLOAD AND INSTALL' and 'LAUNCH THE APP'. Below the modal, a 'NEW' badge is visible next to the text 'Install a Custom App', followed by the instruction 'Install a custom App using docker image name' and an 'INSTALL' button. A settings gear icon is located in the bottom right corner of the interface.

# WaziApps - Jupyterlab



- Enables developers to use kernel-backed documents
- Available languages: Markdown, Python, R, LaTeX, ...

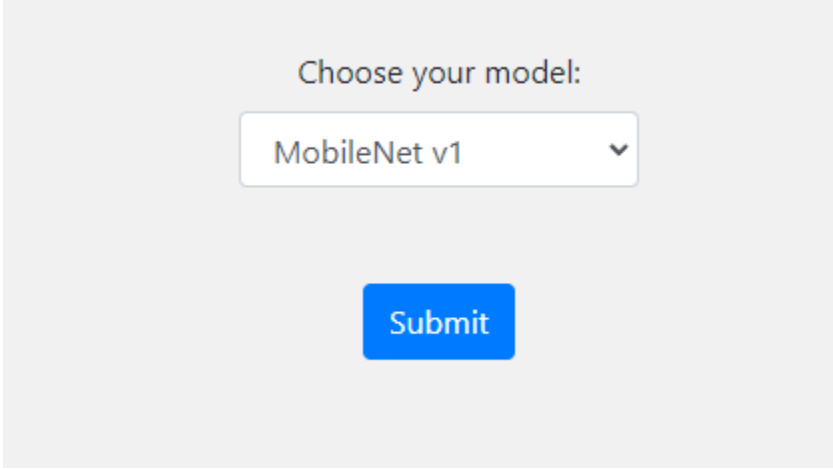


# WaziApps – webcam inference app

- Perform visual detection
- At the moment: two different models included
  - trained, using the COCO dataset
- We want to extend our use cases for this application
  - E.g. decease detection, crop health

# WaziApps – webcam inference app

- A user can select a model to perform the inference with



Choose your model:

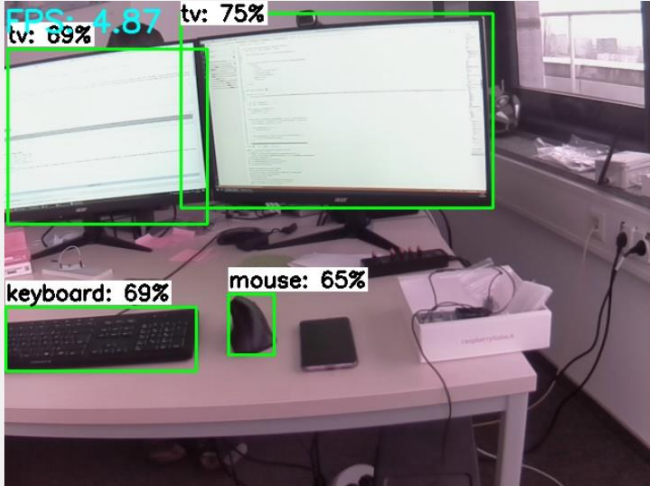
MobileNet v1 ▼

Submit

# WaziApps – webcam inference app

- [Logout](#)
- [On board Sensors](#)
- [wazigate-jupyter](#)
- [wazigate-jupyterlab](#)
- [wazigate-webcam-inference](#)
- [Dashboard](#)
- [Sync](#)
- [Settings](#)
- [Apps](#)
- [Help](#)
- [User Profile](#)

coco\_ssd\_mobilenet\_v1



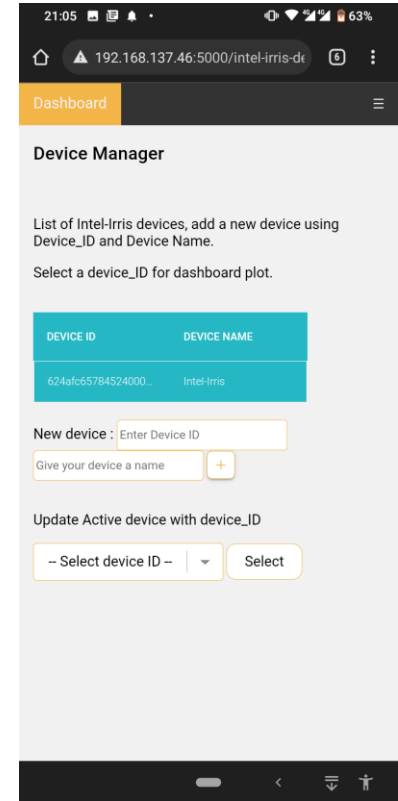
Class	Occurrences	Timestamp last occurrence	Class	Occurrences	Timestamp last occurrence
cup	1	2022-02-03 15:01:30	tv	2	2022-02-03 15:02:04
tv	2	2022-02-03 15:02:04	mouse	2	2022-02-03 15:02:04
mouse	2	2022-02-03 15:02:04	keyboard	1	2022-02-03 15:02:04
keyboard	1	2022-02-03 15:02:04			



- App for Wazigate to estimate generic future values of sensors readings
- Users can specify a range of readings of a sensors: to be included as training data to create a model
- Model can directly be created on Raspberry Pi
- Best model is found according to different metrics on evaluation data

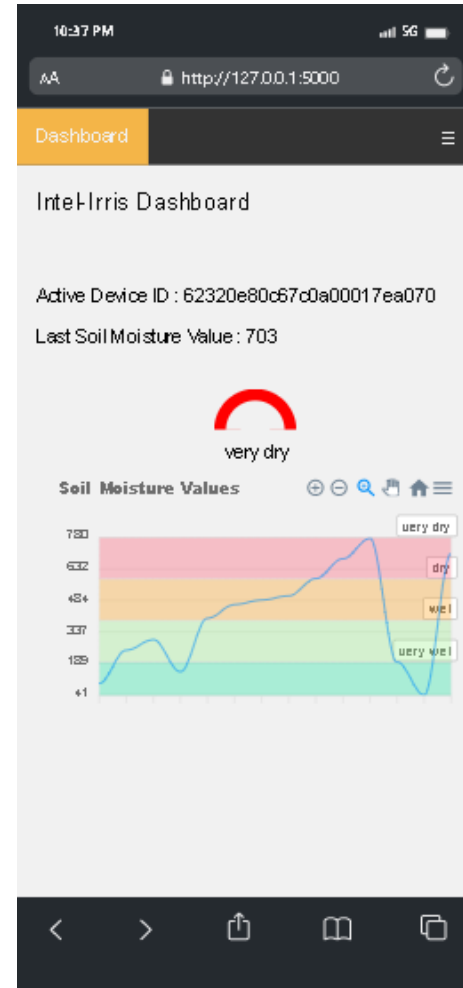
# Intel-Irris WaziApp (IIWA)

- smart irrigation application that focuses on processing of soil parameters from a sensor
- sensor type, plant, soil, irrigation, and weather are also taken into consideration during the processing



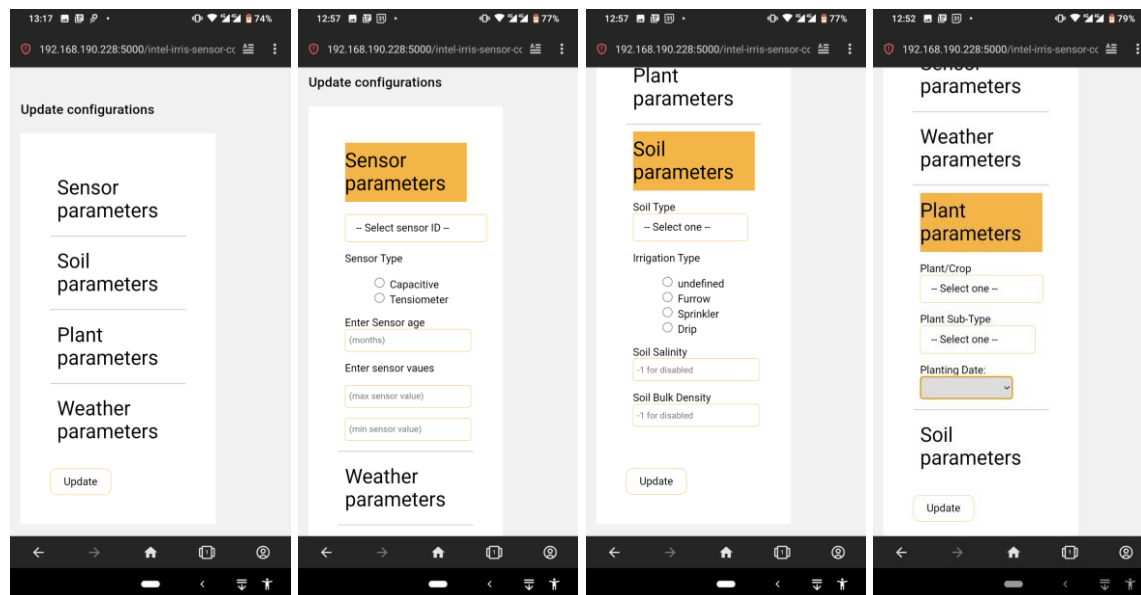
# Intel-Irris WaziApp (IIWA)

- ⦿ A line plot displays the values including a color-coded intervals that define the sensor value:
  - ⦿ very wet, wet, dry, very dry
- ⦿ Insights are presented in form of colored indicators



Parameters such as that can be adjusted:

- sensor id
- sensor type
- sensor age
- Region
- soil type
- irrigation type
- plant/crop type
- planting date
- soil salinity
- bulk density



The image displays four sequential screenshots of the Intel-Irris WaziApp (IIWA) 'Update configurations' screen. Each screenshot shows a different section of the configuration form highlighted in orange:

- Screenshot 1:** The 'Sensor parameters' section is highlighted. It includes a dropdown menu for 'Sensor ID' and an 'Update' button.
- Screenshot 2:** The 'Sensor parameters' section is highlighted. It includes radio buttons for 'Sensor Type' (Capacitive, Tensiometer), an input field for 'Enter Sensor age (months)', and input fields for 'Enter sensor vaues' (max and min values).
- Screenshot 3:** The 'Soil parameters' section is highlighted. It includes a dropdown for 'Soil Type', radio buttons for 'Irrigation Type' (undefined, Furrow, Sprinkler, Drip), and input fields for 'Soil Salinity' and 'Soil Bulk Density'.
- Screenshot 4:** The 'Plant parameters' section is highlighted. It includes a dropdown for 'Plant/Crop', a dropdown for 'Plant Sub-Type', and a dropdown for 'Planting Date'.

# Intel-Irris WaziApp (IIWA)

- From the Sensor Configuration page, a user can select a sensor ID and view its current configuration
- identify the current configuration to make necessary changes or validate if they are correct



**Thank you for your attention!**

**Any questions? Feel free to ask.**