Precision Irrigation with Cost-effective and Autonomic IoT Devices using Artificial Intelligence at the Edge

D4.2

Integration of the OSIRRIS Irrigation system

Responsible Editor: Innotec21

Contributors: Waziup e.V.

Document Reference: D4.1

Distribution: Public

Version: 1.0

Table of Contents

1. Introduction

The OSIRRIS irrigation platform is an affordable, easily deployable and efficient system based on IoT and AI using Edge computing architecture. The core objective of this deliverable is to document the integration of the various components—ranging from soil sensors to edge gateways and physical actuators—into a cohesive, automated system. This integration allows the platform to move beyond simple data monitoring to autonomous irrigation management, where the system calculates the optimal water quantity and executes the irrigation schedule via hardware actuation.

2. Hardware integration

The hardware ecosystem of OSIRRIS is built around the WaziSense sensor nodes, the WaziAct actuator and the WaziGate edge gateway.

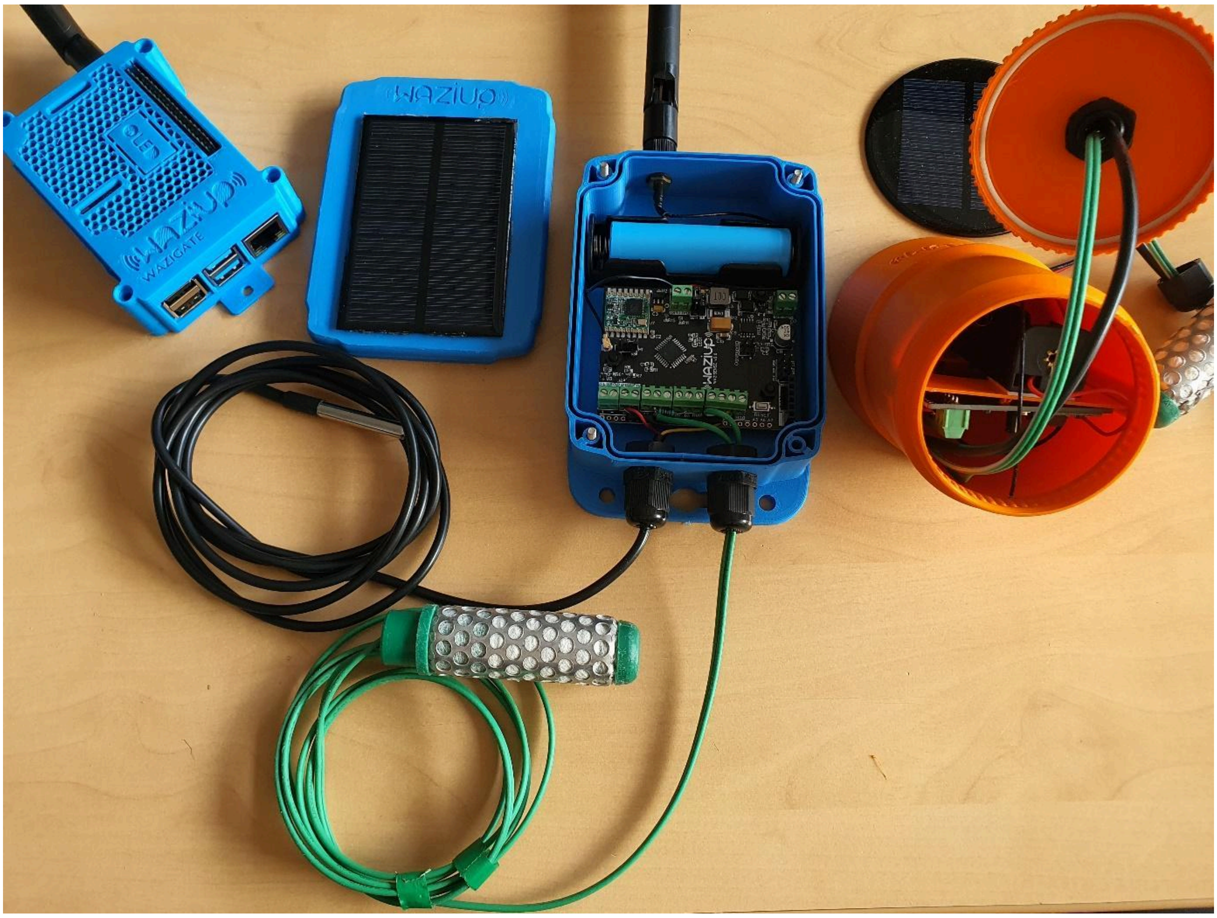

Figure 1: WaziGate (left), WaziSense that can be mounted to a pole (middle), WaziSense to be mounted inside pipe

The WaziGate is the local central entity of the system, it makes a connection to the sensor devices via LoRa and connects to the WaziCloud. It is based on a Raspberry PI, coupled with a single channel LoRa HAT with included RTC clock. Capsuled in a custom case to facilitate the components. It serves as the central hub of the OSIRRIS hardware ecosystem, acting as a versatile IoT LoRa Gateway designed for remote agricultural needs. It can manage up to 100 IoT sensor and actuator nodes simultaneously using a robust LoRa radio network with a communication range of up to 10-12 kilometers. Key features and functionalities of the WaziGate include:

- **Edge Computing Capacity:** Beyond simple data collection, the WaziGate possesses edge capacity, allowing it to host isolated WaziApps within Docker containers. This enables the gateway to run complex AI irrigation models locally, ensuring a responsive system that can function even without a continuous internet connection.
- **Microcontroller Optimization:** Explicitly states that the integration strategy delegates computationally intensive AI tasks to the WaziGate edge gateway. This preserves node battery life and reduces manufacturing complexity, allowing the ATmega328P microcontrollers to focus purely on data collection and swift LoRa transmission.
- **Connectivity and Networking:** The device supports multiple connectivity options, including WiFi, 3G, and Ethernet. It establishes a local WiFi hotspot, allowing farmers to connect via a smartphone or PC to configure settings, view visualizations, and manage the system on-site.
- **Internal Connectivity:** The system uses LoRa communication with “Activation By Personalization” (ABP), hardcoding session keys into the firmware to ensure immediate connectivity upon startup without the need for a complex "join" procedure.
- **Data Management:** It includes an embedded database for local data storage and a web-based visualization module for real-time monitoring. If an internet connection is available, it can automatically synchronize data with WaziCloud for remote access.
- **Local Interface:** For immediate physical interaction, the WaziGate is often equipped with an OLED display mounted on a HAT (Hardware Attached on Top). This display provides quick insights into critical metrics like soil water tension, Volumetric Water Content (VWC), and the next calculated irrigation time. It can also generate a dynamic QR code to help users quickly join the local network and access the full application UI.
- **System Coordination:** In the OSIRRIS workflow, the WaziGate receives sensor data from the Smart Tensiometers via LoRa, stores it, and utilizes its internal AI engine to determine if the "over-dry" threshold has been reached, subsequently triggering the actuation of the irrigation controller.

The Smart Tensiometers, utilizing WaziSense V2 boards, collect soil water tension via a Watermark 200SS soil tensiometer and soil temperature data (DS18B20). They communicate wirelessly via LoRa technology, ensuring long-range connectivity even in challenging environments. The device is solar powered and has a lithium battery to ensure continued operation. A foundational integration strategy is to delegate all computationally intensive machine learning tasks to the WaziGate. This allows the ATmega328P microcontrollers on the sensor nodes to focus exclusively on data collection and LoRa transmission, significantly reducing manufacturing costs and extending field longevity.

Key features and functionalities of the WaziSense include:

- **Energy Autonomy:** The integrated WaziSense V2 features a built-in MPPT solar charge controller, achieving a 105-fold increase in deep-sleep power efficiency compared to its predecessor. This enables the device to operate for 181 days on a single charge of a standard 18650 battery without any solar input.

- **Mechanical Integration:** To ensure durability in harsh agricultural environments, the electronics are integrated into a custom 3D-printed ASA (Acrylonitrile Styrene Acrylate) enclosure. This housing is designed to fit standard 75mm PVC sewage pipes, providing a globally accessible and UV-resistant platform that protects sensor cables from direct sunlight and animal interference.

The Irrigation Controller is a specialized irrigation controller, based on a WaziAct Arduino board, was integrated into the system. This controller is actuated via LoRa directly by the WaziGate. The irrigation controller is connected to a flow meter that measures the amount of the water that is given. The controller incorporates a relay to actuate the water pump and a flow meter to provide real-time feedback on water consumption. This physical integration ensures that the system can not only start a watering event but also monitor the exact volume of water delivered to the plants. The controller is connected in between the power outlet and the pump, it conveniently uses power from the power outlet of the pump which it operates.

Figure2: Pump and installed flow meter

Figure 3: Irrigation controller

3. Software integration

The software integration is centered around the edge-computing capabilities of the WaziGate gateway, which hosts locally executed applications that bridge the gap between sensor data acquisition, artificial intelligence (AI)-based decision making, and physical irrigation actuation. By executing the application directly on the gateway, the system enables autonomous operation, reduced latency, lower bandwidth consumption, and resilience against intermittent internet connectivity.

3.1 Edge Application Architecture

The software integration centers on a dockerized microservice architecture hosted on the WaziGate, ensuring that all AI dependencies (such as PyCaret, Pandas, and others) are isolated and easily manageable.

The Intelligent Decision Engine is composed of:

- Automated ML Workflow: The integration utilizes the PyCaret framework to automatically compare and tune over 24 models, including 19 regression algorithms (e.g., XGBoost, CatBoost) and 5 deep learning architectures (CNN, LSTM, GRU) to capture complex soil-water dynamics.

- Agronomic Data Fusion: Raw soil tension (cb) is automatically converted into Volumetric Water Content (VWC) using the Van Genuchten water retention curve approach. Farmers can select presets for sandy, loamy, or clay soils or input custom curves to match their field's unique composition.
- Multi-plot Management: To handle large-scale operations, the software utilizes threading and a task queue, allowing a single gateway to execute training and inference for multiple agricultural plots concurrently without system instability.

A dedicated irrigation intelligence application was developed and deployed on the WaziGate gateway. The application acts as the central orchestration layer for data collection, preprocessing, AI model execution, visualization, and irrigation control. Furthermore, the application is capable of managing multiple agricultural plots simultaneously within a single deployment.

The application performs the following tasks:

- Acquisition of sensor data from local smart tensiometers via LoRa communication
- Retrieval of meteorological forecast data from external APIs
- Synchronization and preprocessing of heterogeneous data sources
- AI-based irrigation prediction and threshold analysis
- Visualization of sensor values and prediction results
- Irrigation control via actuated pumps or valves
- Continuous retraining and prediction updates over dynamic timespans

The complete software workflow and data flow are illustrated in the infographic below.

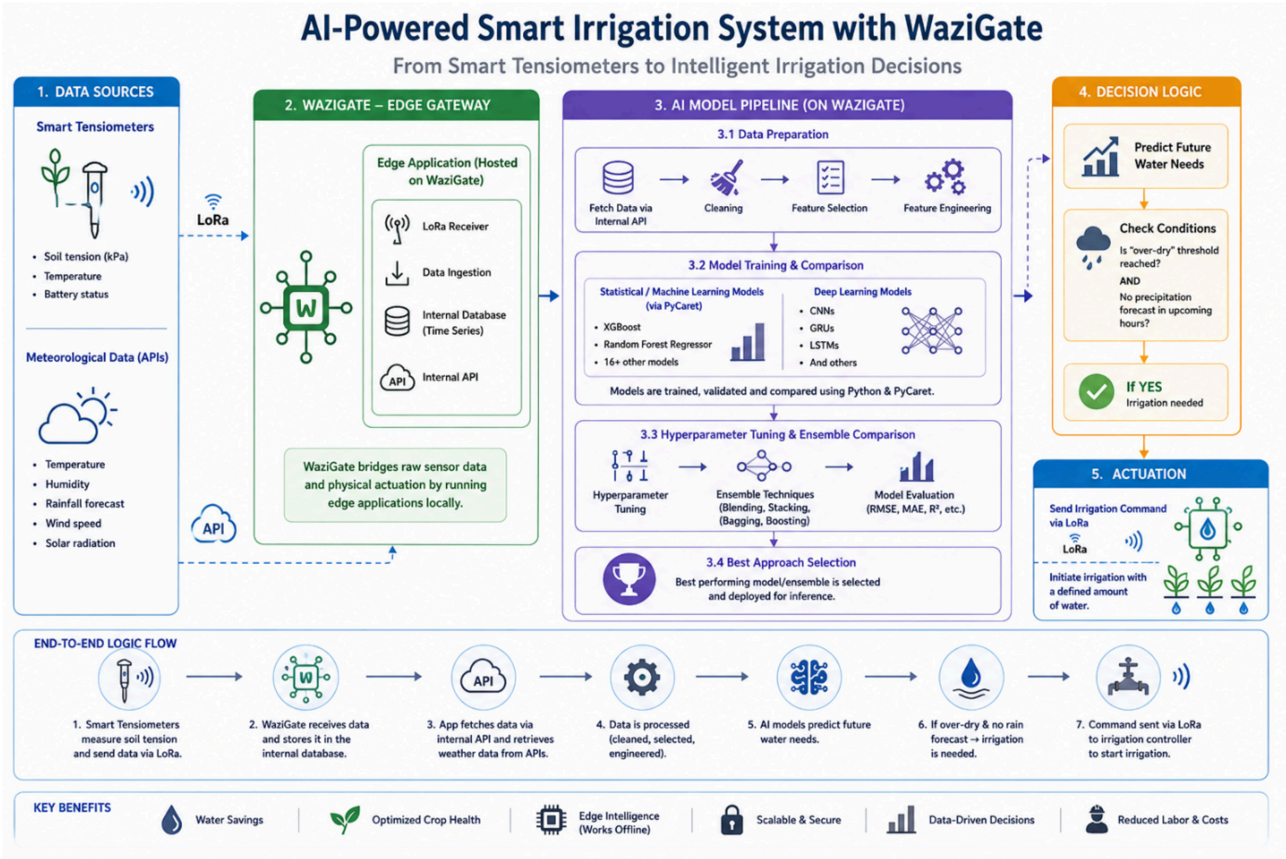

Figure 4: Logic/data flow of the application

The model training and all other calculations are encapsulated in threads that have a queue, tasks are being executed consecutively, this enables the application to handle several plots concurrently. Inside one thread are several subprocesses to facilitate the model creation in a save and resource capsulated way.

3.2 Sensor and Weather Data Integration

The system integrates two primary data sources:

Local Sensor Data

Smart tensiometers installed in agricultural plots continuously measure:

- Soil moisture / soil tension
- Soil temperature
- Sensor health indicators (e.g., battery status)

The sensor measurements are transmitted using LoRa communication to the WaziGate gateway, where they are stored in an internal MongoDB time-series database.

Meteorological Data

To complement local measurements, the system integrates weather forecast information from the external weather service:

[Open-Meteo](#)

The retrieved weather variables include:

- Air temperature
- Relative humidity
- Precipitation forecast
- Wind speed
- Solar radiation
- Additional environmental variables relevant to evapotranspiration and irrigation demand

The integration of weather forecasts allows the system to incorporate expected environmental conditions into irrigation decisions and avoid unnecessary watering before rainfall events.

3.3 Data Synchronization and Feature Engineering

Since sensor measurements and weather forecasts originate from heterogeneous sources with different temporal resolutions, a preprocessing pipeline was implemented to synchronize and harmonize the datasets.

The application automatically performs:

- Temporal synchronization of sensor and weather observations
- Resampling to consistent time intervals
- Missing value handling and outlier reduction
- Data cleaning and normalization
- Feature extraction and feature engineering
- Feature selection to identify the most informative predictors

Examples of engineered features include lagged soil measurements, rolling averages, environmental gradients, and weather forecast indicators that improve prediction performance.

3.4 Artificial Intelligence Pipeline

The system integrates a comprehensive machine learning workflow executed using Python and the PyCaret framework. A total of 24 predictive models are evaluated:

Statistical and Machine Learning Models (19 Models)

The implemented statistical and machine learning approaches include:

- XGBoost
- Random Forest Regressor
- Decision Trees
- Gradient Boosting
- Extra Trees
- Ridge Regression
- Lasso Regression
- Elastic Net
- K-Nearest Neighbors
- Support Vector Regression
- AdaBoost
- LightGBM

- CatBoost
- Additional regression-based and ensemble approaches

These models are automatically trained, validated, and compared using standardized workflows.

Neural Network Models (5 Models)

In addition to conventional machine learning, several neural network architectures are evaluated, including:

- Convolutional Neural Networks (CNN)
- Gated Recurrent Units (GRU)
- Long Short-Term Memory networks (LSTM)
- Hybrid recurrent architectures
- Additional deep learning approaches

The neural network models are designed to better capture temporal dependencies and nonlinear relationships in soil-water dynamics.

3.5 Model Evaluation and Optimization

After training, all candidate models are systematically evaluated using meaningful performance metrics to ensure robust prediction quality.

The evaluation process includes metrics such as:

- Root Mean Square Error (RMSE)
- Mean Absolute Error (MAE)
- Coefficient of Determination (R^2)
- Prediction stability and generalization performance

Based on these evaluation criteria, the three most promising models are selected for further optimization.

Subsequently, a hyperparameter tuning procedure is performed to maximize predictive accuracy and robustness. Different optimization strategies are applied to identify the best-performing parameter configurations.

Following optimization, an ensemble model is created from the three best-performing tuned models. Ensemble techniques improve prediction robustness by combining multiple predictive perspectives and reducing individual model bias.

3.6 Dynamic Retraining and Prediction Cycle

To account for seasonal variations and changing environmental conditions, the system continuously updates its predictive capabilities.

The application:

- Retrains models at dynamically defined intervals
- Adapts to newly available sensor data
- Updates prediction behavior according to environmental changes
- Performs irrigation predictions every 6 hours

This adaptive learning approach ensures that irrigation recommendations remain accurate under changing climatic and soil conditions.

3.7 Decision Logic and Irrigation Actuation

The irrigation decision-making process is based on predicted soil conditions and weather forecasts.

The AI models analyze historical soil tension and meteorological variables to predict future irrigation demand and determine when a predefined "over-dry" threshold will be reached.

The system evaluates the following logic:

1. Predict future soil water conditions
2. Estimate the next optimal irrigation time
3. Determine whether the over-dry threshold will be exceeded
4. Check precipitation forecasts for the upcoming hours
5. Suppress irrigation if sufficient rainfall is expected
6. Trigger irrigation if dry conditions are confirmed

If irrigation is required, the application transmits a LoRa command to an irrigation controller connected to an actuated valve or pump. The irrigation amount is determined according to the calculated water demand and executed automatically.

3.8 User Interface and Visualization

To improve transparency and usability, the WaziGate user interface provides integrated visual monitoring features. The dashboard enables users to:

- Display raw sensor measurements in interactive charts
- Visualize synchronized datasets
- Monitor AI model outputs and prediction results
- Compare model performance
- Observe irrigation recommendations and system decisions
- Manage multiple plots within a single application interface

This visualization layer allows users to understand system behavior and validate irrigation recommendations in real time.

To lower the barrier for agricultural users, the system integrates several multimodal interfaces for real-time status monitoring:

- Gateway OLED Display: An integrated screen on the WaziGate provides immediate insights into soil tension, VWC, and the next suitable irrigation time.
- QR Code Connectivity: The gateway dynamically generates a QR code that allows farmers to join the local WiFi hotspot and access the full web-based dashboard on their smartphones instantly, bypassing manual network setup.
- Automated Alerts: The system is integrated with a notification engine that sends Email and SMS alerts for status updates like "Irrigate Now," "Irrigation Scheduled," or maintenance warnings.

4. WaziApp deployment

Deployment of the OSIRRIS software is handled through a modern, containerized approach to ensure stability and ease of installation in the field.

- **Docker Containers:** The OSIRRIS irrigation application is dockerized, allowing it to run in isolated microservices on the WaziGate. This architecture prevents conflicts between different applications and simplifies the deployment process, because every requirement is included in the container.
- **Easy Configuration:** The configuration is done in the settings menu and is internally managed through JSON files, allowing for the easy adjustment of parameters like soil types and threshold levels without modifying the core code.
- **CI/CD:** The build process is integrated into a CI/CD pipeline for automated building of the docker container, afterwards functional testing is conducted and the application is deployed on a test system. The last step is to automatically push the image to dockerhub.

5. Actuation

Actuation is the final step in the OSIRRIS "sense-decide-act" loop, moving from data surveillance to physical resource management.

- **Closed-Loop Surveillance:** The WaziAct controller performs independent surveillance of the irrigation process. Once the WaziGate triggers the pump via LoRa, the WaziAct monitors real-time pulses from a flow meter and automatically stops the pump when the pre-calculated water volume is reached regardless of subsequent LoRa connectivity.
- **Pump Control:** The WaziAct module is the primary interface for automation, controlling the pump via the integrated relay.
- **Precision Delivery:** During the Tunisia installation, the integration of the flow meter allowed the system to perform closed-loop control. When the AI model determines an irrigation event is necessary, the Gateway actuates the pump. The system continuously monitors the flow meter data; once the anticipated water amount is reached, the relay is triggered to stop the pump, preventing water wastage and ensuring optimal plant health.
- **Remote Calibration:** A key integration feature is the ability to remotely calibrate the flow meter via LoRa. Users can update the conversion factor (pulses per liter) through the WaziGate dashboard to accommodate changes in pipe diameter or equipment without physical access to the device.

6. APIs

APIs serve as the "connective tissue" for the OSIRRIS system:

- **WaziGate API:** This internal API enables communication between the WaziApp, the local database, and the WaziAct hardware control module.

- **External Weather APIs:** The system utilizes the Open-Meteo API to fetch historical and forecasted meteorological data (e.g., evapotranspiration, rainfall, and solar radiation), which significantly improves the accuracy of soil moisture predictions.
- **WaziCloud API:** For remote monitoring, the WaziGate synchronizes data to WaziCloud, allowing farmers to access metrics and system status from any location.
- **Irrigation Prediction API:** The application itself also has an API, that exposes and provides all relevant information like raw sensor data, combined and engineered data, predictions, thresholds and configurations. This way preset configurations can be prepared to automatically configure the application. Multimodal interfaces like an OLED could also display e.g. threshold timestamps about next irrigation events, to give a quick overview.

7. Delivery/Distribution

The system is designed for localized deployment in mind, to minimize costs and logistical complexity. Nevertheless, the system for example, the use of standard PVC sewage pipes for device casing allows parts to be sourced locally in regions like Tunisia. The final integrated solution is delivered as a pre-configured WaziGate image, which can be flashed onto an SD card for rapid setup on-site deployment.

8. Conclusion

The integration of the OSIRRIS system represents a significant advancement in autonomous precision irrigation. By combining low-cost sensors, edge-based AI models, and robust actuation hardware like the Tunisia irrigation controller, OSIRRIS provides farmers with a powerful tool to optimize water usage, reduce manual labor, and enhance crop yields through data-driven decision-making.