SWAMP

SMART WATER MANAGEMENT PLATFORM

Project nº: 777112

WP2

D2.4 Final Data Collection and Storage Services

Editor: Juha-Pekka Soininen (VTT)

Author(s): Juha-Pekka Soininen (VTT), Kari Kolehmainen (VTT), Carlos Kamienski (ABC), Luca Roffia (UBO), Massimiliano Menghini (UBO), Ramide Dantas (UFPE)

Status – Version: 1.1

Date: 2.12.2020

Distribution – Confidentiality: Public

Code: 777112-SWAMP - D2.4 Final Data Collection and Storage Services

RNP

Disclaimer

Participant no.	Participant organisation name	Part. short name	Country
1 (European Coord.)	Teknologian tutkimuskeskus VTT	VTT	FI
2	Intercrop	ICRO	ES
3	University of Bologna	UBO	IT
4	Consorzio di Bonifica dell’Emilia Centrale	CBEC	IT
5	Quaternium	QUAT	ES
6 (Brazilian Coord.)	Federal University of ABC	ABC	BR
7	Centro Universitário da FEI	FEI	BR
8	Federal University of Pernambuco	UFPE	BR
9	LeverTech Tecnologia Sustentável	LEV	BR
10	Brazilian Agricultural Research Corporation	EMBR	BR

Document revision history

Date	Issue	Author/Editor/Contributor	Summary of main changes
20 August 2020	0.1	Juha-Pekka Soininen (VTT)	Preliminary document with standard template layout.
15 September 2020	0.2	Kari Kolehmainen (VTT)	Some amendments to the template
30 September 2020	0.3	Ramide Dantas (UFPE)	Added UFPE contributions.
05 October 2020	0.4	Carlos Kamienski (ABC)	Addition of platform service descriptions form GitLab
12 October 2020	0.5	Luca Roffia (UBO)	Added UBO contributions
28 October 2020	0.6	Massimiliano Menghini (UBO)	Added multispectral camera functions
11 November 2020	1.0	Juha-Pekka Soininen (VTT)	Final version
12 December 2020	1.1	Juha-Pekka Soininen (VTT)	Final fixes and formatting

Internal review history

Date	Reviewer	Summary of comments
12 November 2020	João Kleinschmidt (ABC)	Approved with comments and updates
23 November 2020	Hannu Tanner (VTT)	Approved with some fixes and remarks.

Table of contents

Abbreviations

AGROVOC	Agriculture vocabulary
API	Application Programming Interface
ARPAE	Advanced Research Projects Agency-Energy
C++	Programming language
DLS	Dynamic Light Scattering
DSM	Digital Spectrum Modulation
ETSI	European Telecommunications Standards Institute
FIWARE	Future Internet platform
GIS	Geographical Information System
GPIO	General Purpose Input Output
HAT	Hardware at Top
HTTP	Hypertext transfer protocol
ICT	Information and Communication Technologies
IMU	Inertial Measurement Unit
IoT	Internet of Things
IoTEE	IoT Entity Editor
IoTMA	IoT Management Agent
IR	Infrared
JSON	JavaScript Object Notation
MATOPIBA	Area in Brazil covering Maranhão, Tocantins, Piauí and Bahia states
MQTT	Message Queuing Telemetry Transport
NDVI	Normalized Difference Vegetation Index
NGSI	Next Generation Service Interfaces
NGSI-LD	Next Generation Service Interfaces Linked Data
NIR	Near Infrared
PCSE	Python Crop Simulation Environment
PDF	Portable Document Format
QUDT	Quantities, Units, Dimensions and Types [ontology]
RDF	Resource Description Framework
RDFS	Resource Description Framework Schema
REST	Representational State Transfer
RFID	Radio Frequency Identification
RPA	Robotic Process Automation
SEPA	SPARQL Event Processing Architecture

SOSA	Sensor Open Systems Architecture
SPARQL	Semantic Protocol And RDF Query Language
SSN	Semantic Sensor Network [ontology]
TIFF	Tagged Image File Format
VE	Virtual Entity
VRI	Variable Rate Irrigation
W3C	World Wide Web Consortium
WDA	Water Distribution Application
WiFi	Wireless Internet for Frequent Interface
WSplug	Weather Station plug

Executive Summary

This deliverable D2.4 Final Data Collection and Storage Services contains the final version of the SWAMP data collection and storage layer components, including descriptions and implementations of the following topics:

1. Communication and storage infrastructure services.
2. Drone as an IoT Service concept.
3. Virtual Entirty (VE) representations of the SWAMP data model.
4. IoT Services.

The deliverable has a short description of extensions from earlier deliverables D2.1 and D2.7. Major part of the deliverable are the actual design documents and descriptions of source codes that are stored in open access repositories. This document has links to the open source implementations in GitLab and other repositories, where the source codes and models exist.

1. Introduction

This document describes what SWAMP will have implemented at the end of the project. It describes the purpose of the source codes produced in the project and gives links to Gitlab or similar repository. The main aim has been to create a complete list of artefacts developed in SWAMP.

The document has following relationships with other SWAMP deliverables:

- D1.4 SWAMP Architecture model describes how complete system describes the system structure and how it works
- D2.1 Communication and storage substrate describes how these services work together
- D5.3 Pilots infrastructure implementations describes what devices have been developed or deployed in pilots

The structure of the document is following. Chapter 2 presents the SWAMP platform services. Chapter 3 describes the drone and camera system parts. Chapter 4 presents the SWAMP applications. Chapter 5 describes the data and ontology models developed in SWAMP. Chapter 6 summarizes the open research data published in the Italian pilot.

All the data in this deliverable is available as open source in Gitlab or other similar service or usable with open licence.

2. Communication, storage, and IoT services in the SWAMP platform

2.1. IoT platform services

The SWAMP IoT Platform services are the set of software modules responsible for obtaining, storing, and calibrating the sensor data until it is ready to be consumed by other applications. The IoT platform services provide a layer of abstraction for the SWAMP Application Platform that directly consumes the sensor data from the layer below. The IoT Platform services are composed of third-party software and by applications developed within SWAMP.

2.1.1. IoT Distribution Package

The IoT Distribution Package provides an easy and practical way to instantiate the SWAMP IoT Platform locally. When downloading and executing just one command, the IoT Distribution Package will download and instantiate – as Docker containers – all the modules composing the SWAMP IoT Platform. On top of that, the IoT Distribution Package will populate Orion with the main entities of the SWAMP information model. This module provides third-party developers a kick-start to the SWAMP IoT Platform. The following modules compose the SWAMP IoT Platform:

- FIWARE
  - Orion Context Broker
  - QuantumLeap
- SWAMP
  - LoRaWAN UltraLight 2.0 IoT Agent
  - SWAMP COVID
- ChirpStack
  - Gateway Bridge
  - Network Server
  - Application Server
- Database
  - MongoDB
  - CrateDB
  - Redis
  - PostgreSQL

https://git.rnp.br/swamp-essentials/iot-platform/iot-distribuition-package

2.1.2. LoRaWAN Ultralight 2.0 IoT Agent

The IoT Agent is a FIWARE General Enabler responsible for translating non-NGSI data into NGSI format and sending it to Orion. Thus, it works as an entry point for IoT data in FIWARE-based IoT platforms. There is no official and stable IoT Agent that integrates with LoRaWAN, especially with modules from the ChirpStack LoRaWAN Server. Thus, we implemented an IoT Agent that receives messages from a LoRaWAN server – the ChirpStack Application Server – and sends them to Orion, in the NGSI format, mapping each LoRa device to the Orion entity. The LoRaWAN server uses the MQTT protocol to send messages to the IoT Agent and has the payload structured in the Ultralight 2.0 protocol. Like the official FIWARE IoT Agent, our IoT Agent uses MongoDB to store metadata about the device-entity mapping.

https://git.rnp.br/swamp-essentials/iot-platform/swamp-iot-agent

2.1.3. SEPA (SPARQL Event Processing Architecture)

The SWAMP platform integrates SEPA (SPARQL Event Processing Architecture) as enabling technology for Linked Open Data services. All the services and applications for the Italian pilot have been implemented on top of SEPA. SEPA is a publish-subscribe architecture designed to support information-level interoperability.

The architecture is built on top of a generic SPARQL endpoint where publishers and subscribers use standard SPARQL 1.1 Updates and Queries. Notifications about events (i.e., changes in the RDF knowledge base) are expressed in terms of added and removed SPARQL binding results since the previous notification.

https://github.com/arces-wot/SEPA

2.1.4. SEPA-based MQTT IoT agent

The service allows to collect data from multiple MQTT brokers and to map them according to the W3C SOSA ontology.

https://github.com/arces-wot/SEPA-MqttIotAgent

2.1.5. SWAMP Sensor Calibration

The soil probes measure the soil moisture using capacitance sensors in millivolts and then send the raw values to the SWAMP Platform. However, there is the need to calibrate those millivolts values to moisture percentiles. The SWAMP Sensor Calibration is a service that receives the millivolt messages from Orion and calibrates them according to the management zone where they were generated. Each message calibration consists of a polynomial function previously determined by the analysis of management zone soil. After the calibration, the SWAMP Sensor Calibration service sends the data back to FIWARE Orion.

https://git.rnp.br/swamp-essentials/iot-platform/swamp-calibration

2.1.6. Weather Station plug (WSplug)

The Weather Station collects current local weather data like air humidity, air temperature, precipitation, and other environmental data. The weather station used by SWAMP Project in Brazilian pilots is the VAISALA WXT520 that provides RS-232/RS-485 serial communication. The purpose of WSplug is obtaining data from the weather station via serial RS-232 communication and publishing it in Orion context broker using the NGSI protocol. Written in C++, the WSplug runs three threads: the first one opens the serial port stream and decodes the data sent by the weather station. The second publishes the decoded weather data in Orion using Http/NGSI protocol in a specified interval. The third thread provides an API for a health check and instant current data query.

https://git.rnp.br/swamp-essentials/iot-platform/wsplug

2.2. Application platform services

The SWAMP Application platform consists of services that enable field monitoring using the applications, irrigation planning and optimisation, irrigation execution, and interaction with CBEC water distribution system.

2.2.1. SWAMP API

A myriad of microservices constitutes the SWAMP Platform, each of them having a different API and automatic data exchange with other platform components. Furthermore, two FIWARE components are responsible for data management: Orion, which stores the current state of an entity, and QuantumLeap, which stores complete time series. The SWAMP API is a transparent interface with the SWAMP platform, providing a unified entry point for applications to access platform services. Thus, it facilitates the implementation of new applications, offering a richer alternative to the services provided by the FIWARE platform, including notifications. The API follows the REST architectural style, using HTTP and JSON as communication protocol and data format, for universal support. It offers endpoints for historical and current data and the data format of results follows the FIWARE NGSI conventions.

https://git.rnp.br/swamp-essentials/application-platform/swamp-api

2.2.2. SWAMP Data Fusion

The SWAMP Data Fusion is a service that implements algorithms for computing aggregate metrics summarized by hour and day over data coming from soil moisture sensors, weather stations, and weather forecast services. This service is needed for two reasons. Firstly, different data is produced at different rates and they must be equalized before being processed by the water need estimation models. Secondly, data must be summarized at the same granularity used by the water need estimation models, i.e., daily, hourly, or even at a different granularity. Data is consumed from FIWARE Quantum Leap, which provides offers an API to persist and query time-series data in a database. The computed data is stored on new entities on Orion that are in turn stored by Quantum Leap for maintaining a historical time series.

https://git.rnp.br/swamp-essentials/application-platform/swamp-data-fusion

2.2.3. Weather Forecast Collection

The Weather Forecast Collection is a set of services that collect data from Weather Forecast services and stores them within the SWAMP Platform. Currently, AccuWeather, OpenWeather, and Climatempo are being collected by this service, and stored in FIWARE Orion, so that a time series becomes available through Quantum Leap.

https://git.rnp.br/swamp-essentials/application-platform/weather-forecast

2.2.4. Soil Moisture Forecast

The Soil Model Forecast is a service that takes as input data from soil moisture probes and weather forecast information and uses a predictive model to predict the soil moisture of a future period. The predictive model is exchangeable and can be replaced by a customized model trained from data of a specific site, as long as this model has the same interface. The computed forecasts are stored in the platform for computing the water need.

https://git.rnp.br/swamp-essentials/application-platform/water-need

2.2.5. Irrigation Optimization Service

The Irrigation Optimization Service is tasked with the generation of irrigation plans for the farm. It uses as input the soil moisture forecasts (section 2.2.4) for the management zones, along with information about the irrigation system (e.g., method of irrigation) to generate a set of irrigation recommendations that comprise the new irrigation plan. Each recommendation refers to an irrigation zone (which typically matches a management zone, except for cases such as MATOPIBA with Centre Pivot irrigation) and describes the irrigation depth per time unit (usually hour) for the next days, according to the forecast provided. The service is triggered indirectly whenever new forecast estimates are available. More details on the optimization approach implemented can be found in deliverable D3.2 (Optimisation for Water Irrigation).

https://git.rnp.br/swamp-essentials/application-platform/swamp-water-optimization/

2.2.6. Application Platform Distribution Package

The Application Platform Distribution Package provides an easy and practical way to instantiate the SWAMP Application Platform locally. When downloading and executing just one command, this module will download and instantiate – as Docker containers – all the components comprising the SWAMP Application Platform, providing third-party developers a kick-start to the SWAMP Application Platform.

https://git.rnp.br/swamp-essentials/application-platform/application-distribution-package

2.2.7. SWAMP RPA

The SWAMP RPA (Robotic Process Automation) is a set of scripts for detecting the creation of new devices in FIWARE Orion and automating the creation of these new devices in FIWARE IoT Agent and ChirpStack LoRaWAN Server.

https://git.rnp.br/swamp-essentials/application-platform/swamp-rpa

2.2.8. Root Size Estimation

The plants can only absorb water up to the root zone. Therefore, when calculating the volume of water available to the plant, for some crops, it is necessary to know the depth of the roots. Furthermore, the growth of the roots depends on several aspects, such as the crop development stage, soil type, weather conditions, and crop management. To produce reliable estimates of the root size, we have implemented a service based on the PCSE crop simulator. This simulator is widely used in the literature, and implements several models for crop simulation. The implemented service consumes data from the platform (such as crop and soil type, soil humidity, and weather conditions) and computes the root size. The estimated root size is then stored back in the platform.

https://git.rnp.br/swamp-essentials/application-platform/root-size-estimation

2.2.9. Water Need Calculation

The Water Need Calculation is a service that receives the soil moisture forecasts generated by the service of the same name (section 2.2.4) and transforms them into a amount of water to be sprayed into the crop. In other words, it receives information of the forecast for the soil moisture (given in percentage) for a given time frame (e.g. days) in advance, and translates it into an amount of water that needs to be refilled in the soil to avoid both under-irrigation (given by the wilting point) and over-irrigation (given by the field capacity). This process also considers that the root system covers different depth (currently, SWAMP consider three depths) and that the water need calculation must consider all of them. A simple approach followed by the preliminary implementation of this component is to consider that once the field capacity of a certain depth is reached, the water percolates to an underlying depth.

https://git.rnp.br/swamp-essentials/application-platform/water-need-calculation

2.2.10. Irrigation service

This is a collection of services that depend on irrigation systems. In SWAMP we had two irrigation systems that were planned to be operated through SWAMP application platform: Cartagena IoT node-based system and MATOBIPA central pivot system.

Cartagena service development was under development when COVID-19 lockdown came into effect. It was supposed to be an algorithm that creates IoT node control sequences based on irrigation plan and irrigation system model, and it writes status updates to virtual entities of IoT actuators in Orion. The IoT Agent then forwards the commands to nodes that subscribe to the entities.

In MATOPIBA case the irrigation system is controlled by irrigation prescription map that is sent to it. No actual commands are needed. As VRI central-pivot system is under procurement from non-SWAMP partner. The development of irrigation service is pending.

2.2.11. CRITERIA water need estimation

CRITERIA is a soil water balance model simulating one-dimensional water fluxes, crop development and crop water needs. The soil and crop parameters can be defined at different levels of detail. It requires as input daily agro-meteorological data: minimum and maximum air temperature, total precipitation and, if available, data of watertable depth to estimate the capillary rise. It is available as open source at: https://github.com/ARPA-SIMC/CRITERIA1D

CRITERIA has been integrated within the SWAMP platform to provide irrigation recommendations. The SWAMP CRITERIA services retrieve the required inputs from a SEPA instance and output the results back to SEPA.

https://github.com/arces-wot/CriteriaSWAMPService

2.2.12. ARPAE weather agent

The service collects weather observations and forecasts from the ARPAE open data. Data are available as 5 km x 5 km GRIB data. Collected data are transformed and then stored into a SEPA instance within the SWAMP platform.

https://github.com/arces-wot/gribtranslate

2.2.13. CBEC adapter

The service has been developed to collect data coming from CBEC, mainly related to irrigation requests. At the time of writing more than 15K requests have been collected and registered into the SWAMP platform. The service also allows collecting data from the CBEC sensor network, like pluviometers, water flow sensors and water levels.

https://github.com/arces-wot/swamp-cbec

2.2.14. Water distribution optimization service

The water distribution optimization service aims to provide the daily scheduling of the network operations.

The service takes as input:

- The structure of the network (number of channels, water travelling time for each channel, etc.);
- The considered time horizon (number of time intervals, time interval duration, number of operations, etc.);
- The water requests (preferred start time, volume requested, etc.).

As output, it provides the optimized scheduling.

https://github.com/vlatorre847/SWAMPOPT

2.3. Management platform services

The SWAMP Management System is a part of any SWAMP System focused on managing the components of the SWAMP IoT Platform (FIWARE, SEPA, LoRa Server) and the SWAMP Application Platform. The main components of the SWAMP Management System are the IoT Entity Editor (IoTEE), IoT Management Agents (IoTMA) and the IoT Entity Data Visualization.

2.3.1. IoT Entity Editor

The SWAMP IoT Entity Editor (IoTEE) offers to the SWAMP Management System the functions of Creating, Reading, Updating and Deleting (CRUD) Virtual Entities. The virtual entities are stored in FIWARE Orion, that in turn stores them in MongoDB. The IoTEE provides a user friendly GUI that prevents users to handle directly REST APIs using scripts.

https://git.rnp.br/swamp-essentials/management/swamp-iot-ee

2.3.2. IoT Entity Data Visualization

The IoT Entity Data Visualization service is the main dashboard of the SWAMP Management System that shows time series information of entities monitored by IoTMA agents. IoTMA stores the collected data from monitoring points in FIWARE Orion, which in turn is stored by QuantumLeap. The IoT Entity Data

Visualization service obtains the data from QuantumLeap and shows it to the users in a variety of ways. Using this interface, users can check whether devices are working properly in a visual and friendly manner.

https://git.rnp.br/swamp-essentials/management/swamp-dashboard

2.3.3. IoT Management Agents

The main functionality of an IoT Management Agent is to monitor and control the mist, fog, and cloud devices (physical/logical), and the software components running on them. Monitoring is basically a data collection of the functioning of hardware/software components, such as fault or performance. This information is stored in FIWARE Orion. Controlling means some actuation on the components, such as turning on/off some hardware equipment or software component and changing some configuration.

https://git.rnp.br/swamp-essentials/management/iotma

2.4. Communication and actuation services and devices

2.4.1. SWAMP Local Gateway

The SWAMP Local Gateway is implemented as a Linux service with Python language. Gateway provides access to wired and wireless sensors and actuators. Gateway provides the following sensor and actuator interfaces:

- LoRaWAN Sensors using LoRaWAN Gateway HAT
- LoRa Sensor using low level protocol
- Bluetooth sensors
- Power System sensor via Victron Energy interface
- Two interrupt based water meter sensors can be connected
- CanBUS interface for CanBUS sensors and actuators
- Two 12V latching solenoid actuators for irrigation valves

Gateway periodically collects and relays data from sensors to SWAMP platform as well as subscribes to relevant actuator entities for commands such as irrigation valve entities.

https://git.rnp.br/ctic/swamp/tree/master/VTT/SWAMP_LocalGateway

2.4.2. Gateway shield

The Gateway shield is a Raspberry Pi hat that provides power for raspberry as well interfaces to CanBUS and the irrigation valve control as well as water meter reading via Raspberry Pi GPIO. CanBUS interface is implemented with off-the-shelf component. The Gateway shield is designed so that it can be stacked with the Raksystems LoRaWAN gateway shield extending the available communication methods to LoRaWAN.

https://git.rnp.br/ctic/swamp/tree/master/VTT/SWAMP-GWShield

2.4.3. LoRaWAN sensors

Three LoRaWAN sensor nodes were developed, tested and installed: the EMBRAPA sensor node, the UNIBO Multisensor Node, and VTT sensor node.

The first one was developed in order to manage the EMBRAPA Soil Moisture Probe. The second LoRaWAN interface can manage up to six general-purpose METER[1] sensors such as soil moisture sensors (TEROS12 or GS3), temperature sensors and water table sensors (HYDROS). The VTT Sensor node is three layer soil probe.

[1] https://www.metergroup.com/

EMBRAPA Sensor Node

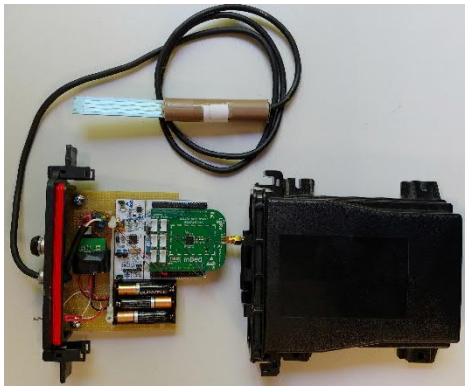

FIGURE 1. EMBRAPA SOIL MOISTURE SENSOR NODE

The EMBRAPA Sensor Node is powered by a STMicroelectronics NUCLEO board based on the STM32L1 microcontroller and a Semtech SX1272 LoRa shield to reach the LoRaWAN network[2]. The STM32L1 controls a power management module to supply the EMBRAPA soil moisture sensor only during measurements and, through the integrated ADC, samples the analog soil moisture data from the sensor. It has been installed in June 2019 and it is still running with a battery percentage of 79% thanks to its low power profile.

UNIBO Multisensor Node

FIGURE 2. UNIBO SOIL PROBE SENSOR.

In order to be compliant with the specifications about low energy consumption and low costs of maintenance, the node is powered by a STM32L0 microcontrollers and the LoRa protocol is used for the communication between the server and the node thanks to the integration between the microcontroller and the LoRa transceiver made by Murata (CMWX1ZZABZ-091). METER sensors are managed through a custom circuit for sensor supply and they communicate through the DDI serial protocol.

Within the Italian pilot, both nodes are installed in the Bertacchini's Farm in Correggio, Reggio Emilia and they are reachable from three different LoRaWAN Servers: the first one provided by The Things Network[3], the second powered by Chirp Stack[4] and the third through LEPIDA RetePAIoT[5]. Data from sensors (e.g., soil

[2] https://lora-alliance.org/

[3] https://www.thethingsnetwork.org/

[4] https://www.chirpstack.io/

[5] https://www.retepaiot.it/

moisture, soil temperature, soil conductivity and water table) are forwarded by servers to the SWAMP Platform in JSON format and they are available for users through the Water Distribution Application (WDA) (http://mml.arces.unibo.it/swamp/wda).

Soil Moisture sensor (VTT)

The Three-layer soil sensor is a capacitive sensor element that measures soil moisture and temperature up to 15cm depth in 5cm increments. Sensor also has conductive sensing element. Sensor cannot be used as a stand-alone element but requires electronics to read it. Sensor has a Texas Instruments ADS1115 24-bit Analog to Digital Converter that can be read via I2C bus.

https://git.rnp.br/ctic/swamp/tree/master/VTT/3LayerSensor

2.4.4. Sprinkler Valve Actuators

The Sprinkler Controller is a communication board that is able to control a latching solenoid irrigation valve. Communication with the controller is done via CanBUS interface. The CanBUS interface is an off-the-shelf component commonly available. Solenoid controlling is done via H-Bridge so that polarity of the solenoid voltage can be changed enabling on-off latching. The Controller is based on Microchip 32-bit microcontroller and the firmware is implemented with the Harmony development environment.

https://git.rnp.br/ctic/swamp/tree/master/VTT/SWAMP-SprinklerController

3. Drone as an IoT service

3.1. SWAMP drone implementation

The SWAMP Drone is a Pixhawk-based medium-sized drone capable of carrying varying payloads. The Drone was developed to carry multispectral cameras, RFID Reader, and Gateway electronics. Drone is composed of off-the-shelf components built around custom frame designed by Quaternium.

https://git.rnp.br/ctic/swamp/tree/master/Quaternium

3.2. Drone Gateway

The SWAMP Drone Gateway is a python software that acts as a drone mounted gateway and relays data from drone and sensors to SWAMP cloud. Drone gateway act as gateway and translator between SWAMP cloud platform and drone flight computer. Drone gateway subscribes to command topics published by the Drone entity in the SWAMP platform and updates its status to the SWAMP platform. Similarly, the Drone gateway acts as a gateway for RFID sensors that the on-board RFID reader can read. RFID can be swapped to Bluetooth or other wireless interface based on the application. Gateway uses underlying platform services to access internet either via 4G network of WiFi.

https://git.rnp.br/ctic/swamp/tree/master/VTT/SWAMP_DroneGateway

3.3. NDVI camera service

The NDVI Camera is a low-cost trial implementation of a multispectral camera using off-the-shelf components. Camera is based on two IMX377 Imaging units, where one has been modified to record near infrared spectrum (NIR). Modification is done by removing IR cut filter from the sensor and adding IR pass filter on front of the camera. This enables camera module to capture only NIR band of the spectrum while the other camera captures RGB band. For NDVI calculation, only red and NIR bands are required.

https://git.rnp.br/ctic/swamp/tree/master/VTT/NDVI_Camera

3.4. Pix4DMapper Multispectral Camera Processing

The 3D model reconstruction and NDVI maps are obtained by analysing the multispectral images using the software Pix4DMapper. This photogrammetric software can generate Index map needed to calibrate the parameters of the field model and it is also possible to monitor the state of health of the crops.

To have the highest levels of accuracy for each analysis, the SWAMP template for image analysis has been defined. The table below describes the main parameters of the SWAMP template:

SWAMP Template Settings		
1. Initial Processing	Keypoints Image Scale:	Full
	Quality Report:	Yes
	Correspondance:	Aerial Grid or Corridor
2. Point Cloud and Mesh	Image Scale:	1 (Original image size, Slow)
	Density:	High (Slow)
	Minimum Number of Matches:	6
	Classify Point Cloud	Yes
2.1 3D Texured Mesh	Generate 3D Texured Mesh:	Yes
	Settings:	High Resolution

3. DSM Orthomosaic and Index	Raster DSM, GeoTIFF:	Yes
	Orthomosaic GeoTIFF:	Yes
	Titles e KML of Google Maps:	Yes
3.1 Index Calculator	Correction Type:	Camera Sun Irradiance and Sun Angle using DLS IMU

In this section are reported the outputs of the data analysis of the missions performed within the project. The data are available at the following links:

https://git.rnp.br/ctic/swamp/tree/master/UniBo/2019 Missions

The data analyses are treated in detail in Deliverable 5.2 in section 2.1 (Reggio Emilia pilot)

The following table shows the file types present in the analysis folder:

Pix4d Mapper Outputs Data		
File Format	Software	Use
NDVI Index Map		
.tif (GEOTIFF)	Global Mapper ArcGIS AutoCAD	Visualization Color editing Analysis
.kml	G IS	Visualization Analysis
.pdf	PDF Reader	Visualization
.shp	GIS	Agriculture applications
Reflectance Map Geo TIFF		
.tif (GEOTIFF)	Global Mapper ArcGIS Quantum GIS	Index Map Generation
Grid Index Map Shapefile		
.shp	Farm Works	Agriculture applications

4. SWAMP applications

4.1. SWAMP Farmer Application

The SWAMP Farmer Application is an Android application intended to be the access point to the platform for the final user. It provides field monitoring functionalities via chart and satellite views, showing current and past soil moisture readings. Through the App, the farmer can also visualize, modify, and apply the irrigation plans generated by the platform services. The App accesses the platform through the SWAMP API described in section Error! Reference source not found.. More details on the Farmer App can be found in deliverable D4.2.

https://git.rnp.br/swamp-essentials/application-platform/swamp-farmer-app

4.2. SWAMP web interface

The SWAMP web interface is an experimental environment based on examples presented in the FIWARE tutorials. It uses the same basic structure as FIWARE tutorials, and has been built using Node.js. It can be seen as an alternative to the SWAMP Farmer Application, and can be used as an access point to the platform for the final user. However, currently it does not contain the same range of functionalities.

FIGURE 3: SCREEN CAPTURE FROM THE SWAMP WEB INTERFACE.

https://git.rnp.br/ctic/swamp/tree/master/VTT/SWAMP_web_interface.

4.3. Gatekeeper application

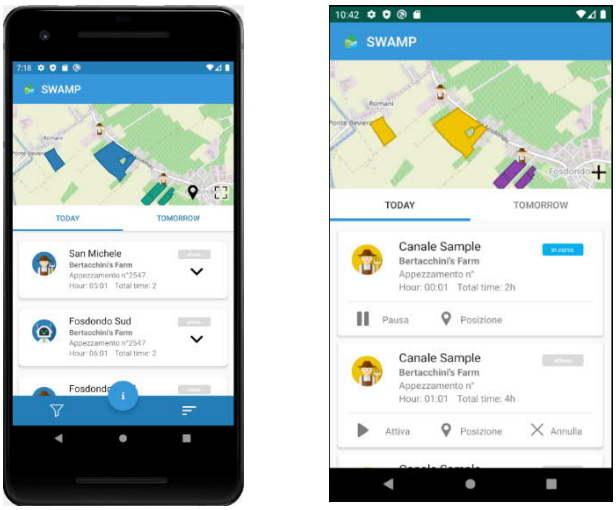

FIGURE 4. GATEKEEPER APPLICATION

This application aims to support the water delivery process carried on by gatekeepers. Through the application, the gatekeeper has a general overview of the scheduled irrigations and he can manage the processing of each one. The application provides a daily plan of the irrigation requests to be satisfied and allows the gatekeeper to give feedback to the platform of the gate operations that have been performed.

https://github.com/arces-wot/Dugarolo

4.4. Water distribution application

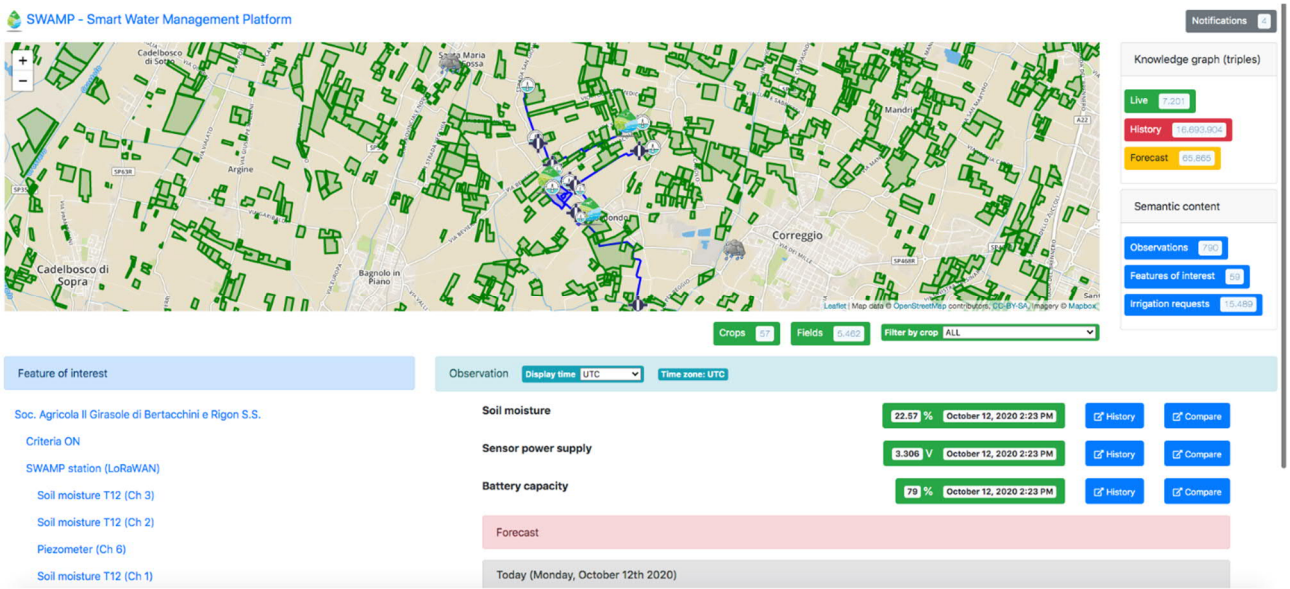

FIGURE 5. WATER DISTRIBUTION APPLICATION

The application allows monitoring real-time data as well as historical ones. Monitor data include sensor data, weather forecasts, irrigation requests, and outputs of the CRITERIA water need estimation model. The application is based on the SEPA component of the SWAMP platform.

https://github.com/arces-wot/SEPAview/tree/swamp

4.5. Sensor calibration application

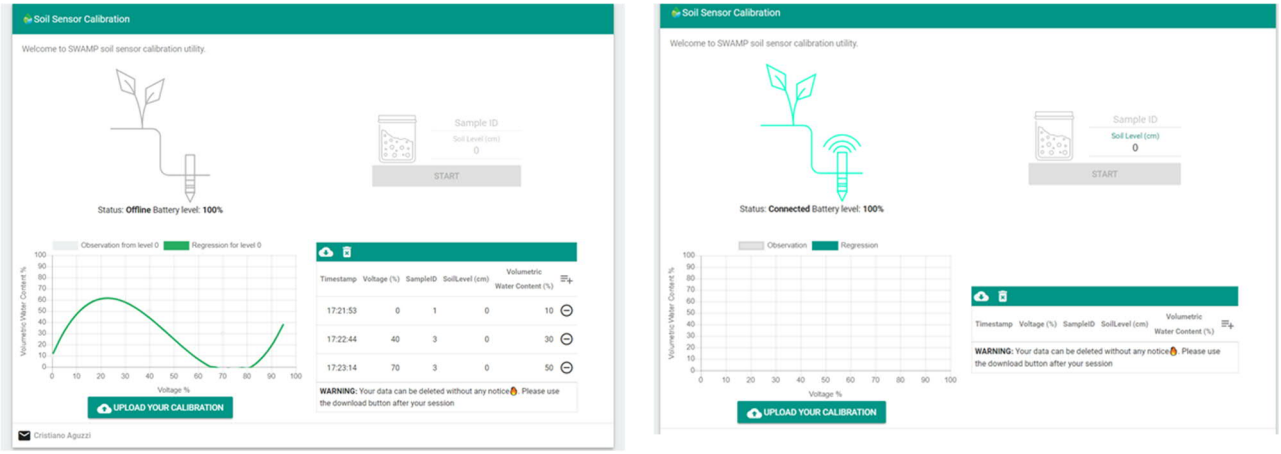

FIGURE 6. SENSOR CALIBRATION APPLICATION.

A simple Web-based calibration tool for a soil moisture sensor. The application collects raw sensor readings which can be linked by the user to the actual soil moisture value. The application computes the calibration curve that can be saved for later use.

https://github.com/relu91/SWAMPCalib

5. Data model and Virtual Entity representations

5.1. SWAMP data model (ontology)

The SWAMP platform is built on top of FIWARE and thus it inherits the FIWARE data model aligned with the ETSI NGSI-LD specifications, which allows representing and retrieving context data according to a meta-model grounded on RDF/RDFS. At the same time, the SWAMP platform integrates SEPA (see Section Error! Reference source not found.) as Linked Open Data enabling technologies. The SWAMP ontology has been developed to enable the integration of both FIWARE and Linked Data ecosystems. On the one hand, Linked Data agents and services can interact with FIWARE using the Linked Data standard protocols and formats, and on the other hand, FIWARE benefits from the interoperable data sources made available through the Linked Open Data cloud. Because of that, the SWAMP ontology has been framed within the ontology proposed by NGSI-LD and it includes the following ontologies:

- NGSI-LD: https://www.etsi.org/deliver/etsi_gs/CIM/001_099/006/01.01.01_60/gs_CIM006v010101p.pdf
- QUDT: http://www.qudt.org/
- SSN/SOSA: https://www.w3.org/TR/vocab-ssn/
- AGROVOC: http://agrovoc.uniroma2.it/agrovoc/agrovoc/en/
- Schema.org

https://git.rnp.br/ctic/swamp/blob/ad35af2f9ecd43c9457c01ef5db027cbef0be827/ontology/swamp.owl

5.2. SWAMP Information Model (Gitlab-model)

The SWAMP Information Model is a repository group that encompasses all the projects that store the SWAMP data model. The repositories store entities in the NGSI structure adopted by FIWARE, each entity is stored in a single JSON file. There is a repository for the template of all entities composing the SWAMP data model, and a repository for each of the SWAMP pilots containing specific pilot data (e.g. number and location of management zones).

https://git.rnp.br/swamp-essentials/swamp-information-model

6. Open research data

The SWAMP platform provides a SPARQL endpoint for an open and interoperable data access:

http://mml.arces.unibo.it/swamp/dlod/

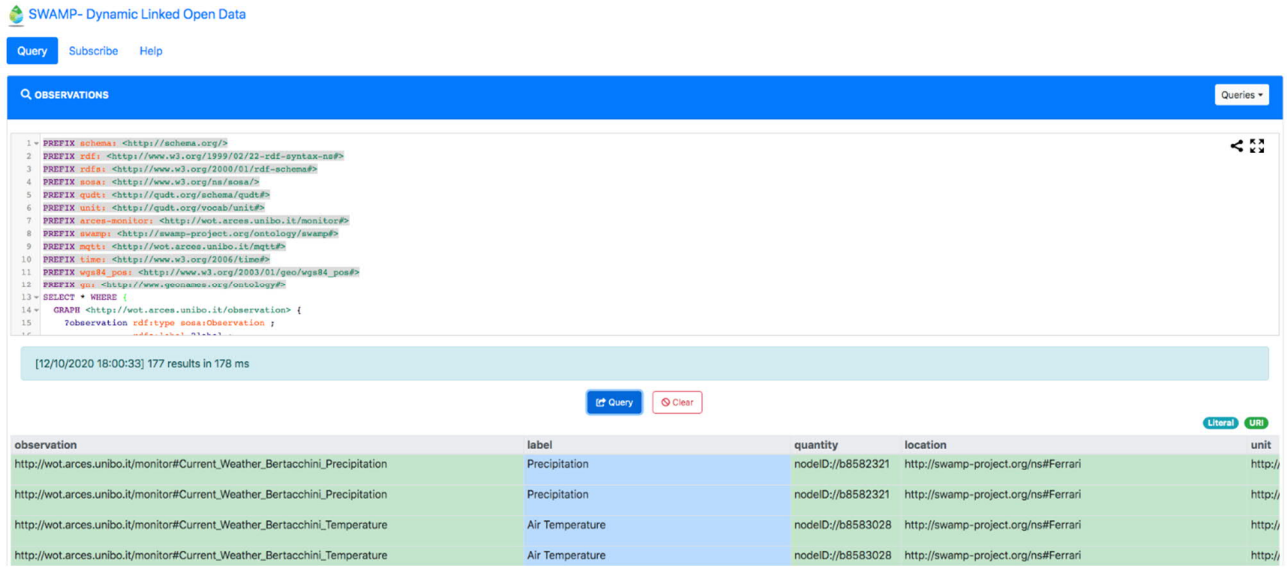

FIGURE7. SCREENSHOT FROM OPEN DATA INTERFACE.

Information interoperability is granted by the SWAMP ontology. Thanks to SEPA is also possible to subscribe and receive notifications on changes in the query results.

7. Summary

The document summarises the implementations of SWAMP platform services and other components in SWAMP pilots that have been developed in the project. All the developed services and models are available as open source through the SWAMP Gitlab or similar open source repository. The design documentation of developed devices are also available through the links provided.

The developed services have been tested and integrated as a part of the SWAMP platform.