***SMART WATER SYSTEM***

IOT BASED PRECISION IRRIGATION FOR

AGRICULTURE

INTRODUCTION:

Agriculture is the biggest consumer of freshwater in the world, amounting to up to 70% of the total use, which makes the case for smart water management in order to guarantee water and food security to the world’s population. Irrigation systems and field application methods for the cultivation of crops play an important role therein. In an attempt to avoid loss of productivity caused by water stress (under-irrigation), farmers spray more water than needed (over-irrigation) and as a result not only productivity is challenged but also water and energy are wasted. Precision irrigation, in its turn, can use water more efficiently and effectively, avoiding both under-irrigation and over-irrigation. The smart management of water for precision irrigation in agriculture is essential for increasing crop yield and decreasing costs, while at the same time contributing to the environmental sustainability.

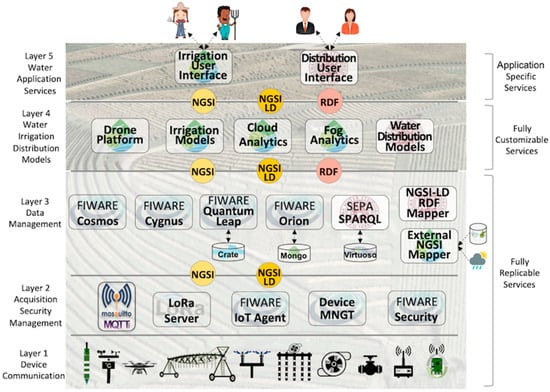
The Internet of Things (IoT) emerges as the natural choice for smart water management applications, even though the integration of different technologies required for making it work seamlessly in practice is still not fully accomplished. The emergence of IoT is a phenomenon that owes to the conjunction of several factors such as inexpensive devices, low-power wireless technologies, availability of cloud data centres for storage and processing, management frameworks for dealing with unstructured data from social networks, high-performance computing resources in commodity platforms, and computational intelligence algorithms to deal with this monumental amount of data

This paper presents the SWAMP project, its architecture, platform and pilots, as well as a scenario-based development process of derived systems. The SWAMP layered architecture considers three categories of services to ensure its replication and adaptability. Entirely replicable services deal with IoT services, storage services, and data analytics and machine learning. Fully customizable services deal with water data management issues that specialize generic analytic services into particular techniques for different types of irrigation and water distribution. Finally, application specific services require higher development effort since they serve particular farms.

CONCEPT AND OVERVIEW:

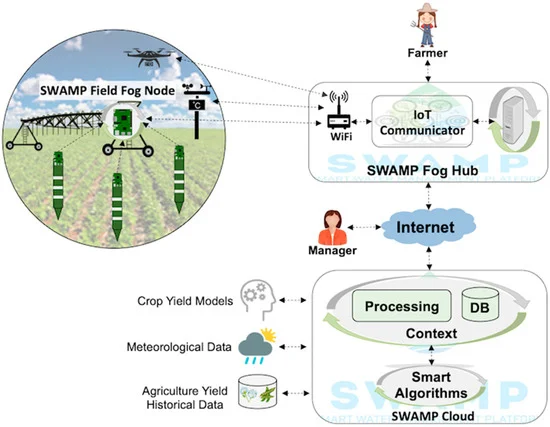
The SWAMP project is developing a high-precision smart irrigation system concept for agriculture. Within SWAMP, water management for agriculture is partitioned into three phases: water reserve, water distribution and water consumption. For Water Consumption SWAMP provides real-time responses for adapting irrigation as crop conditions change. On the other hand, changes in water distribution are performed in a longer timescale. Distribution and Consumption management systems are integrated, as water usage triggers water distribution. The management of water reserves is not considered.

The SWAMP Architecture is divided into five layers

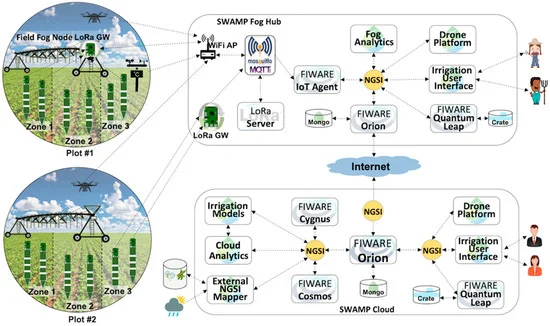


* Layer 1: Device & Communication: a variety of sensor and actuator technologies to acquire soil (e.g., moisture), plant (e.g., growing stage) and weather (e.g., air temperature), as well as LPWAN communication technologies are abstracted in this layer. The SWAMP pilots use commercial sensors as well as a homemade multiparametric sensor probe. Also, commercial drones have been used to take images but a specific drone is under development since one of the partners is a drone maker. A complete description of the sensing infrastructure is outside the scope of this paper.
* Layer 2: Data Acquisition, Security & Management: protocols and software components for data acquisition are the key characteristic of this layer, in addition to security and device management functions. The FIWARE IoT Agent GE also belongs to this layer as it translates the internal FIWARE data representation in JSON from/to devices.
* Layer 3: Data Management: contains software components in charge of data storage, processing and distribution based on FIWARE and SEPA SPARQL engine. A mapper between FIWARE JSON NGSI and SEPA RDF data models also belong to this layer, as well as a mapper from external data sources, such as historical agriculture yield databases and weather forecast services. A distributed infrastructure composed of cloud servers and fog nodes work together for dealing with massive amounts of data and make it available to the upper layers.
* Layer 4: Water Irrigation & Distribution Models: traditional agriculture models for estimating plant water needs using images generated by drones (crop-based approach) and using soil sensors for determining soil moisture (soil-based approach) belong to this layer. Optimization models and techniques for water distribution based on plant water needs are essential whenever collective networks replace individual water sources. Also, computational intelligence (e.g., machine learning) works together with traditional models or in place of them.
* Layer 5: Water Application Services: irrigation services that make sense to farmers and water distributors via user interfaces.

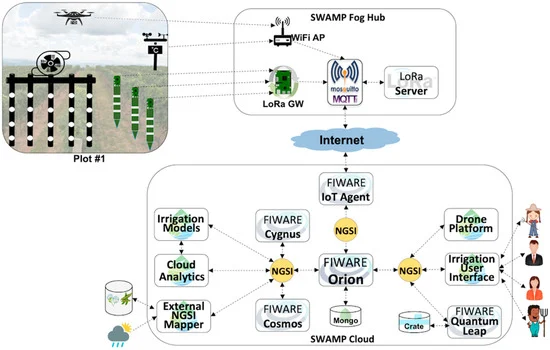
SYSTEM DEPLOYMENT SCENARIOS:



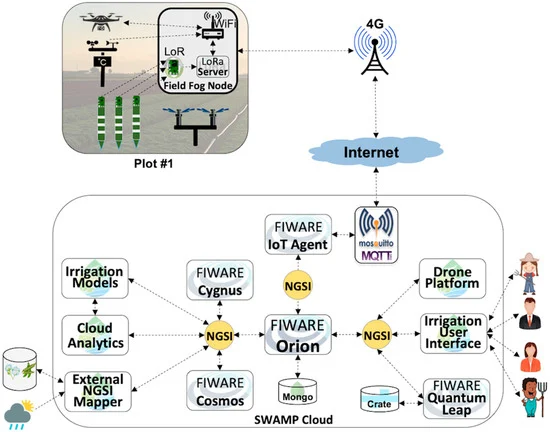
SWAMP Baseline Scenario including SWAMP Cloud and SWAMP Fog (further organized into SWAMP Fog Hub and SWAMP Field Fog Node).

 . 

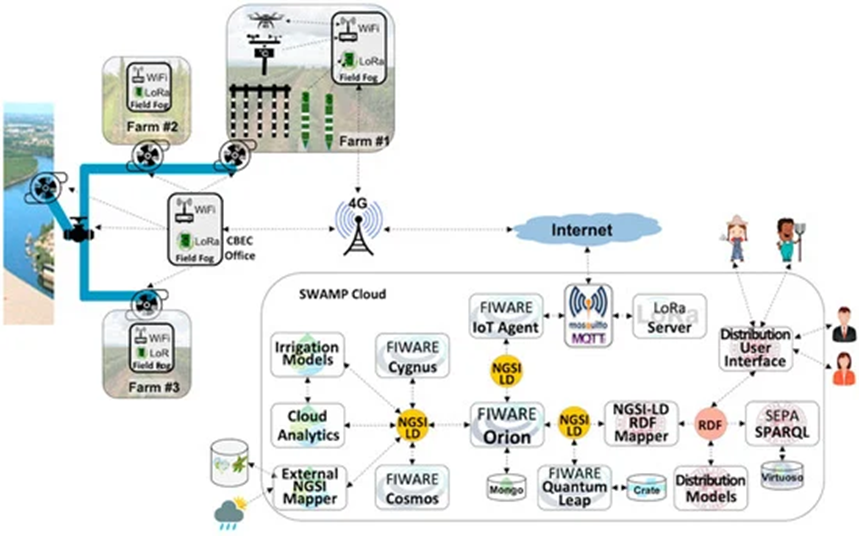
 MATOPIBA Scenario represented in the FIWARE-based SWAMP Platform



Guaspari Scenario represented in the FIWARE-based SWAMP Platform.



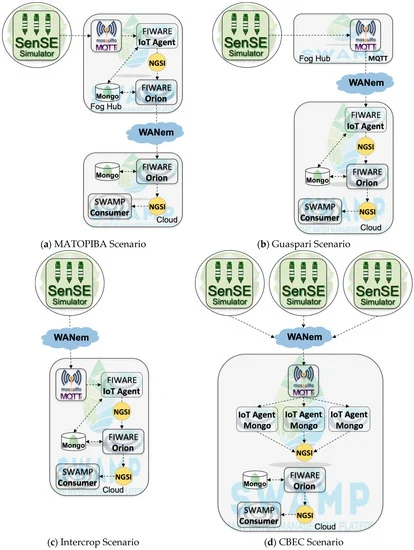
Intercrop Scenario represented in the FIWARE-based SWAMP Platform.



CBEC Scenario represented in the FIWARE-SEPA-based SWAMP Platform.

DESIGN AND RESEARCH METHODS:

In order to analyse the performance and scalability of the four SWAMP pilot scenarios, a FIWARE-based IoT testbed was designed. This involves obtaining sensor data values up to the point where they are transparently consumed by an application that can be deployed in different cloud and fog configurations. The key differences between the scenarios are the placement of the components in the cloud or fog.,



The additional components are:

* Sensor Simulating Environment: It is an open-source large-scale IoT sensor data generator able to abstract real devices and to model different complex scenarios, such as smart farms. The tool is a traffic workload generator that emulates heterogeneous sensors representing tens of thousands of IoT devices sending data simultaneously via MQTT. Although the sensors are synthetic, the traffic is real;
* Mosquitto MQTT Broker: Eclipse Mosquitto is an open source MQTT message broker;
* MongoDB: a document-oriented NoSQL database, serving as the default Orion storage;
* Consumer: a special purpose web application that subscribes in Orion and receives sensor data from the probes.
* WANem Network Emulator: emulates the Internet connection for the assessment of the impact of network parameters between the place where the data is generated (in the farm,) and the place it is processed (in the cloud).

Regardless of the location of components in the cloud or fog, the sequence of processing steps and data flow is always the same, from source ( to destination (Consumer); (1) Sgenerates sensor data and sends it to mosquitto; (2) IoT Agent receives data from Mosquitto, stores it in MongoDB, translates it to NGSI, and sends it to Orion; (3) Orion receives NGSI data from IoT Agent, updates entity values, stores them in MongoDB, and sends them to Consumer; (4) Consumer receives data from Orion and computes the elapsed time since it was generated by SenSE (a timestamp is embedded in the message and physical machines are synchronized by NTP). For the MATOPIBA scenario, there is an additional step, because of the cascading Orion solution, where the Cloud Orion has to subscribe to the fog Orion and the messages are then transferred from one to the other.

A lab testbed emulated the scenario for the experiments. Both fog and cloud were implemented using virtual machines (VM) in OpenStack. The following standard Amazon AWS VM configurations were deployed: cloud VM equivalent to a t2.medium instance (2vCPU—4 GB of RAM) and the fog VMs (both fog field node and fog hub) equivalent to a t2.small instance (1vCPU—2 GB RAM). The cloud was composed of 6 VMs. Two different physical machines were used, with the following configuration: Intel(R) Xeon(R) CPU E3-1240 V2 @ 3.40 GHz—8 cores and 8 GB of RAM.

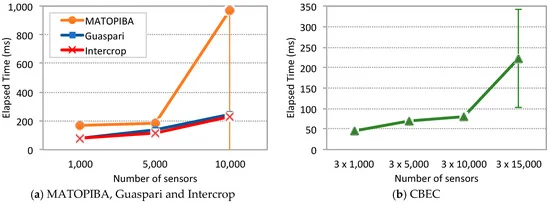
Three categories of metrics were used in the reported experiments:

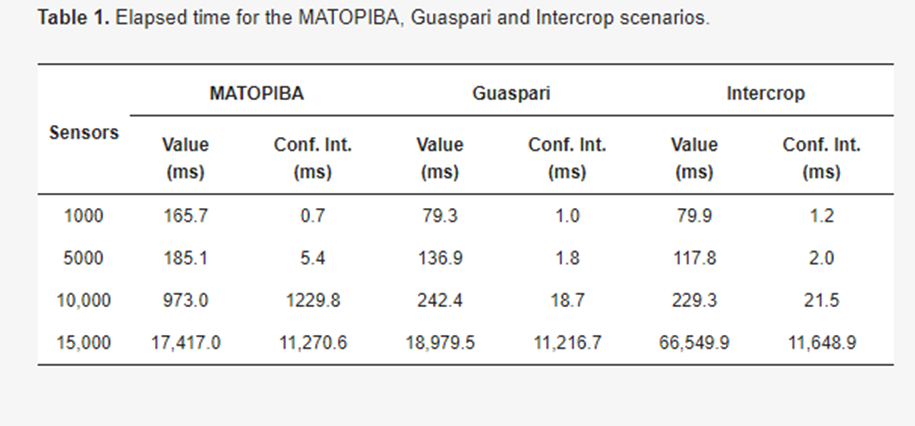
* Application metric: The elapsed time is the average time taken since a sensor data point is generated by Sense until the Consumer receives it. This metric represents how long it takes for sensor data to be available to any subscribed application.
* System metrics: CPU and RAM usage per Docker container, which allows observing each application, collected every 5 s.
* Experiment metrics: The duration of the experiment given by the number of replications (because some components crashed after some time) and the number of received messages are used to understand how experiments unfolded.

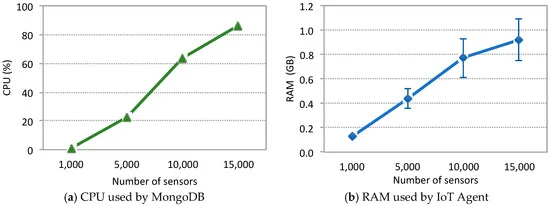
The experiments involved a large number of sensors, sending data every 10 min. The scenarios were executed with four different workloads determined by the number of sensors sending messages simultaneously. For the MATOPIBA, Guaspari and Intercrop scenarios, 1000, 5000, 10,000 and 15,000 sensors were used. For the CBEC scenario, the workload was tripled, total 3000, 15,000, 30,000 and 45,000 sensors. Each experiment took 1 min and was replicated 30 times, totalizing 16 h of running experiments. Asymptotic confidence intervals were calculated with a 99% confidence level.

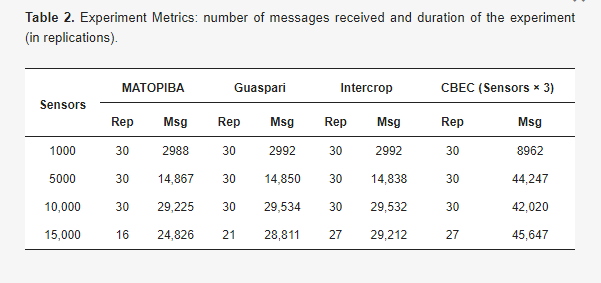
The configuration of WANem captured characteristics of a connection from a farm to a cloud based on a simple experiment that obtained the network parameters by pinging a public cloud using a 4G connection, which resulted in a connection of 10 Mbps with 45 m/s of delay and 5 m/s of jitter.

RESULT:









CONCLUSION:

The emergence of IoT is a phenomenon that owes to the conjunction of several factors and now starts to become real with huge effort both in research and business areas. In this context, the SWAMP project develops IoT-based methods for smart water management in precision irrigation, and pilots them in Italy, Spain, and Brazil. This paper introduced the SWAMP architecture, pilots and deployment scenarios for the four pilots using FIWARE as the underlying IoT platform.

A performance analysis of key FIWARE components personalized for each SWAMP pilot scenario was undertaken to understand the scalability limits of the system. The results show that this platform might be able to deal with the performance requirements of our pilots, even though requiring specially designed deployment configurations and the re-engineering of some components to provide higher scalability using less computational resources. Particularly, our experiments showed that MongoDB is CPU greedy, which negatively impacts system performance.

SWAMP is an ongoing project and therefore there are multiple paths for future work. Some examples are improving the platform deployment scenarios, reporting the overall working of the SWAMP approach in the pilots, including the experience with irrigation models and analytics and more advanced performance analysis.