# Cloud Services for Smart Farming: A Case Study of the Veracruz Almond Crops in Portugal

Filipe Fidalgo[1,2][0000-0001-7326-9957], Osvaldo Santos[1,2][0000-0003-0341-2839], Ângela Oliveira[1,2][1111-2222-3333-4444], José Metrôlho[1,2][0000-0002-7327-2109], Fernando Reinaldo[1,2][0000-0002-1225-3844], Antonino Candeias[1], Jorge Rebelo[3], Paulo Rodrigues[3], Rodrigo Serpa[3] and Rogério Dionísio[1,2][0000-0002-6810-2447]

[1] Polytechnic Institute of Castelo Branco, 6000-084 Castelo Branco, Portugal
[2] R&D Unit DiSAC, 6000-767 Castelo Branco, Portugal
[3] Veracruz, Idanha-a-Nova, 6060-188 Portugal
rdionisio@ipcb.pt

**Abstract.** Efficient use of resources is a critical factor in almond crops. Technological solutions can significantly contribute to this purpose. The VeraTech project aims to explore the integration of sensors and cloud-based technologies in almond crops for efficient use of resources and reduction of environmental impact. It also makes available a set of relevant and impactful performance indicators in agricultural activity, which promote productivity gains supported by efficient use of resources. The proposed solution includes a sensor network in the almond crops, the transmission of data and its integration in the cloud, making this data available to be consumed, processed, and presented in the monitoring and alerts dashboard. In the current state of the development, several data are collected by sensors, transmitted over LoRaWAN, integrated using AWS IoT Core, and monitored and analysed through a cloud business analytics service.

**Keywords:** Smart farming, AWS, Database, LoRaWAN, The Things Network.

## 1 Introduction

Population growth and climate change are the two major challenges for agricultural production in the 21st century. The pressure to produce more with fewer resources implies changes in production methods and in the efficient management of each method. The Food and Agriculture Organization of the United Nations (FAO) recommends that all agricultural sectors equip themselves with innovative techniques and tools, to adopt and take advantage of digital technologies [1].

Thus, there is a growing commitment to precision agriculture, where the objective is to optimize production per agricultural unit, using the most modern tools to achieve better results in a sustainable approach.

The development of wireless sensor network technologies has been opening doors to new ways of developing precision agriculture.

Currently, a multitude of monitoring systems and data generation are installed and operating on many farms in the Beira Interior region, in Portugal. However, such systems operate in a non-integrated way, so the potential of cross-information between image captures and processing, the use of agricultural production factors, the planning of water use, soil, evolution of crop productivity, pests and diseases, meteorology and human resources is not achieved [2].

Current solutions do not supply such level of integration to group multiple data sources and correlate it to extract useful information for supporting and providing decisions. In general, the available solutions provide only closed islands of data (rain sensors, tree counts, stem growth, etc.). Even sensors with a set of APIs available are not, in fact, able to integrate information useful to optimize the agricultural planning solution. Moreover, the amount of data, due to its complexity and multiplicity, is not being accompanied by processes that lead to comprehensive decision support systems for the farmers [3].

The objective of this work is to develop and implement a model and present a solution based on Cloud services platforms for capturing multiple data from different sources, as shown in Fig. 1, so that, through smart farming, precision agriculture, real time monitoring and business intelligence solutions, they can introduce increases in productivity and optimize the usage of resource in the almond cultivation process of the VeraCruz orchards in Portugal.

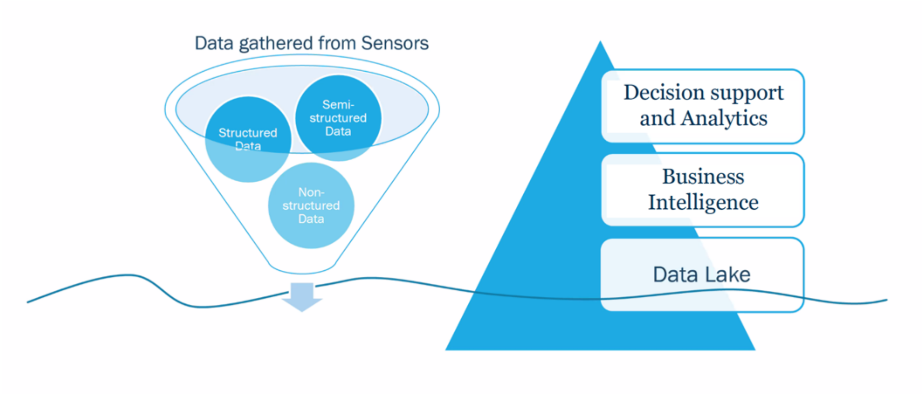


**Fig. 1.** Integration of several data source into a single decision support platform.

The use of Cloud platforms is a requirement for the development of an integrated decision support service for almond orchards. Given the variety of IoT cloud platforms currently available, difficulties may arise in the process of selecting the one that best suits the necessary requirements. A recent study was made in [4] and concluded that the platforms that best suited a set of defined criteria are commercial solutions, and that open-source platforms generally only provide basic functionalities. We compared the features of commercial IoT cloud platform currently available from the main players in the market: Amazon Web Services (AWS), Google Cloud Platform (GCP) and Microsoft Azure. The results are presented in Table 1.

From the analysis of Table 1, except for GCP that failed to comply with one requirement, the other two platforms have similar services and support and from that point of view, it is difficult to choose. Recent studies have compared and analyzed the performance of the platforms from simulations to identify which ones are more efficient when exposed to different usage situations [5-7]. The results show that AWS is a better performing cloud provider in serverless applications than Azure in constant workload. Thus, AWS is the elected choice to implement our solution.

**Table 1.** Comparative analysis between three commercial cloud platforms for IoT applications.

| Requirement \ Platform | AWS | GCP | Microsoft Azure |
|---|---|---|---|
| **Device management** | Yes | Yes | Yes |
| **Device simulation** | Yes | No | Yes |
| **Dashboards** | Yes | Yes | Yes |
| **Alerts and warning configuration** | Yes | Yes | Yes |
| **Development (SDK and programming languages)** | Yes<br>AWS IoT Device SDKs and AWS Mobile SDKs (C++, Java, Javascript, Python, Embedded C) | Yes<br>Node.js, Python, Go, Java, .NET, Ruby and PHP | Yes<br>Azure IoT Hub SDK (C, C#, Java, Node.js, Python)<br>Azure IoT Hub REST API (Go, PHP) |
| **Machine learning** | Yes | Yes | Yes |
| **Storage limits** | Scalable. According to the subscription. | Scalable. According to the subscription. | Scalable. According to the subscription. |
| **Communication protocols** | Yes<br>MQTT, MQTT over WSS – Websockets<br>HTTPS | Yes<br>MQTT, MQTT<br>HTTPS | Yes<br>MQTT and AMQP<br>MQTT and AMQP over WebSockets, HTTPS |
| **Security (Access levels and permissions, device authentication, encryption)** | Yes<br>Encryption TLS/SSL of the data transmission; Authentication with X.509 certificate; *endpoints certificate* | Yes<br>TLS 1.2, Support Authentication, access control, security tokens, encrypted connections, and digital certificates. | Yes<br>Support Authentication, access control, security tokens, encrypted connections, and digital certificates. |

Besides the presented introduction, this paper is organized as follows. Section 2 presents the methodology used and details about Architecture and Implementation. Finally, in Section 3 the conclusions and further work are presented.

## 2 Methodology

Considering the goals and requirements previously identified, a set of guidelines was identified for defining the architecture of the solution. Thus, it must include the installation of a sensor network in the almond's plantation, the transmission of data and its integration in the Cloud, and make these data available to be consumed, processed, and presented in a monitoring, recommendation, and alerts dashboard and analyzed using Business Intelligence algorithms.

## 2.1 Architecture

Four main modules of the architecture were identified, each of which includes several components (see Fig. 2):

1. Sensoring;
2. Integration;
3. KPI identification and generation;
4. Visualization and decision support dashboards.

The architecture must allow the integration of all data from the sensors and devices included in the "sensing network" in the farm. It should also allow the integration of various parameters from external data sources, such as data from farm machinery, data from weather stations, data from and pest and disease detectors, and data about the use of resources and on the characterization of production. This data must be integrated into a data model on the Cloud to allow scalability. Data is later accessed through API by the various application modules of the solution. This data model must be able to store the data and must allow continuous evolution.

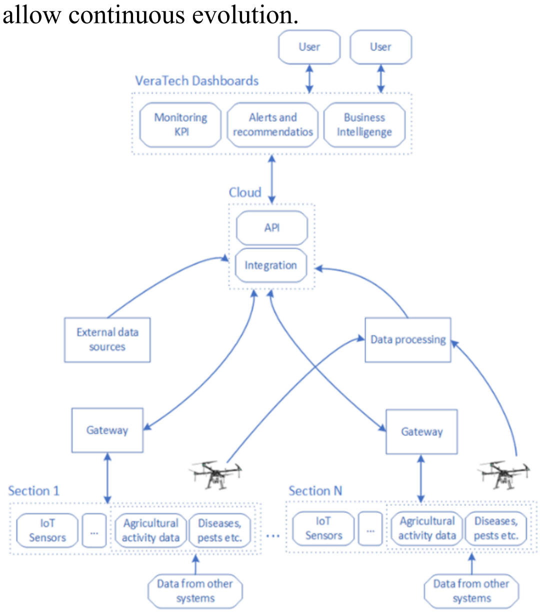


**Fig. 2.** An overview of the proposed architecture.

## 2.2 Implementation

Several wireless nodes are currently installed in the Veracruz Almond Crops, as shown in Fig. 3. Sensors are collecting information from soil, air, almond trees, and critical node parameters and sending data to a TTN (The Things Network) server using LoRaWAN.


**Fig. 3.** Location of the sensors and gateway on the crop field.

## Payload Encoder and Decoder

Sensing data from a wireless sensor network is an energy and time-consuming process. In the specific case of LoRaWAN sensor connected through the TTN server, for EU863-870 frequency band, the Regional Parameters 1.0.2 Rev B as used by the TTN community network, define duty-cycled limited transmissions to comply with the European Telecommunications Standards Institute (ETSI) regulations. As a rule-of-Thumb, the TTN fair access policy allows for at most 30 seconds uplink airtime and 10 downlink messages per sensor, per 24 hours [8].

For the case of the almond crop wireless device, for sensors with integer output with values greater than 255 (8 bits), the encoding process divides the variable into sections of one byte each, starting by extracting the least significant 8 bits, and proceeding with a bit-shift technique (8 bits) to extract the remaining sections of the variable. For a sensor with analogue (float) outputs, the values are initially multiplied by 100 to store information up to two decimal places, then the value is truncated to an integer, and then follow an approach like the one used for integers. For sensor that may output negative values (e.g., temperature), we set the most significant bit (bit 0) of the most significant byte to represent the value sign: 0 is negative and 1 is positive.

## AWS Integration

The TTN is integrated with the AWS IoT Core service, a highly scalable managed IoT broker that can connect billions of IoT devices with cloud applications and other devices [9]. This service is thus the data interface between the IoT devices and the developed applications.

Every time a sensor sends a message to the TTN, a copy of that uplink message is relayed to the AWS IoT Core. Each IoT device publishes its messages in a specific topic related to that device's type. Incoming messages to IoT Core are fed to a set of rules that filter the messages by subscribed topic, and then apply routing policies to them. These rules can be configured with direct connectors to other AWS services such as Dynamo DB, S3, Analytics and others. However, there is not a direct connector to relational databases like Aurora, thus an intermediate layer (Lambda functions) is being used to connect the rule to the Aurora database, as can be seen in Fig. 4.

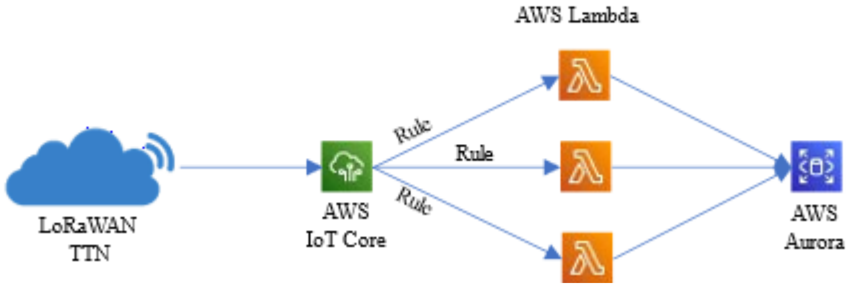


**Fig. 4.** Message routing path from TTN to Aurora database.

There are three different rules, one for each type of device. Currently the IoT network comprises three types of IoT sensors: Dragino devices (for temperature and humidity), soil devices (for temperature, humidity, and electrical conductivity) and water flow sensors (volume and alarms). Each rule connects to its own lambda function, which intercepts the messages, fetches some fields, and inserts them in the Aurora database. The lambda functions are written in Node.JS and perform some additional pre-processing, such as data quality checks, data format conversion and average calculation. These uplink messages are sent by the TTN in a specific JSON format [10], which includes multiple fields of the TTN itself, and specific fields that come from the IoT device's payload, embedded in the uplink_message section.

**Table 2.** Data fields fetched by the lambda functions.

| JSON element | Description |
|---|---|
| device_id | TTN end-device identifier |
| latitude, longitude, altitude | GPS coordinates of the device |
| received_at | ISO 8601 UTC time |
| rssi | Received signal strength indicator (dBm) |
| snr | Signal-to-noise ratio (dB) |
| airtime | Time-on-air, calculated by the network server |

Some of the message fields are common to all the sensors whereas other are device specific. Table 2 enumerates the common fields of the different devices, which are fetched by all the lambda functions and sent to the Aurora database. The dragino IoT devices, which publish message in the aws-dragino topic, send the additional fields shown in Table 3:

**Table 3.** Data fields specific of the dragino device.

| JSON element | Description |
| --- | --- |
| BatV | Battery voltage |
| Bat_status | Battery status |
| Hum_SHT | Humidity (internal probe) |
| TempC_SHT | Temperature (internal probe) |
| TempC_DS | Temperature (external probe) |

The soil sensors publish messages in the aws-milesight topic and have the following specific fields:

**Table 4.** Data fields specific of the soil sensor.

| JSON element | Description |
| --- | --- |
| ec | Electrical conductivity |
| humidity | Humidity |
| temperature | Temperature |
| battery | Battery level |

Finally, the water flow sensors publish messages in the aws-axiom topic and have the additional fields shown in Table 5.

**Table 5.** Data fields specific of the water flow sensor.

| JSON element | Description |
| --- | --- |
| volume | Water flow volume |
| reverseFlow, battery, leak, permanent, burst, dry, low-Temperature | Binary fields used to transmit alarm conditions |

A database represents a collection of data core for business, usually stored in digital form. For the work present in this study, Amazon Web Services (AWS) were used. AWS as DBaaS offers more than fifteen engines to support different data models: relational, key-value, document, in-memory, graph, time series, wide column, and ledger databases [11].

After a detailed analysis of the data involved, several relationships between them were identified that should be preserved, and due to the volume of data a relational database – Aurora Database was selected, which made it possible to create meaningful information by joining the tables. Additionally, relational databases allow users to use stored procedures. A stored procedure is an object that can be stored in a database for later use and can be used many times. Several stored procedures were created and were called by other AWS objects that passed them a set of parameters. Stored procedures evaluated these parameters and depending on the result obtained store the received data in specific tables keeping the referential integrity and strong data consistency. This stored data will be later used for creating dashboards (see Fig. 5) and key performance indicators (KPI) to support the organization's decisions. Quicksight [12] was the AWS service selected for creating dashboards with relevant visualizations for users to gain insights from the data.


**Fig. 5.** Section of the dashboard showing temperature and humidity values.

## 3 Conclusions

This paper presented a case study on the use of Cloud services in the scope of precision agriculture that combines the use of state-of-the-art technological innovations to support the decision-making for the promotion of better working techniques, promotion of better use of natural resources and the protection of the environment in the Beira-Baixa region of Portugal. With the rapid growth in technology, there is a continuous opportunity to manage agriculture data and present it in dashboards to allow an interactive analysis of data, enabling stakeholders to obtain insights to support their decisions. Activating IoT-based alarms can prevent damage in cultures, like a flood in productive areas. Systems accuracy can improve by choosing reliable sensors which fit in the appropriate cloud services.

## Acknowledgment

The research leading to these results has received funding from the Portugal2020 (13/SI/2020) under grant agreement No. 113287 [VERATECH]. The authors would like to thank all VERATECH partners for their contributions.

## References


1. S. I. Hassan, M. M. Alam, U. Illahi, M. A. Al Ghamdi, S. H. Almotiri and M. M. Su'ud, "A Systematic Review on Monitoring and Advanced Control Strategies in Smart Agriculture," in IEEE Access, vol. 9, pp. 32517-32548, 2021, doi: 10.1109/ACCESS.2021.3057865.
2. Jahanzad E, Holtz BA, Zuber CA, Doll D, Brewer KM, Hogan S, et al. (2020) Orchard recycling improves climate change adaptation and mitigation potential of almond production systems. PLoS ONE 15(3): e0229588. https://doi.org/10.1371/journal.pone.0229588
3. J. V. Y. Martnez, A. F. Skarmeta, M. A. Zamora-Izquierdo and A. P. Ramallo-Gonzlez, "IoT-based data management for Smart Agriculture," 2020 Second International Conference on Embedded & Distributed Systems (EDiS), 2020, pp. 41-46, doi: 10.1109/EDiS49545.2020.9296443.
4. Oliveira, L., & Silva, F. (2020). Análise comparativa de plataformas baseadas em Cloud para o desenvolvimento de aplicações IoT. In Anais Estendidos do XXXVIII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos, (pp. 257-264). Porto Alegre: SBC. doi:10.5753/sbrc_estendido.2020.12427
5. Marcin Copik, Grzegorz Kwasniewski, Maciej Besta, Michal Podstawski, and Torsten Hoefler. 2021. SeBS: a serverless benchmark suite for function-as-a-service computing. In Proceedings of the 22nd International Middleware Conference (Middleware '21). Association for Computing Machinery, New York, NY, USA, 64–78. https://doi.org/10.1145/3464298.3476133
6. D. Ustiugov, T. Amariucai and B. Grot, "Analyzing Tail Latency in Serverless Clouds with STeLLAR," 2021 IEEE International Symposium on Workload Characterization (IISWC), 2021, pp. 51-62, doi: 10.1109/IISWC53511.2021.00016.
7. Pascal Maissen, Pascal Felber, Peter Kropf, and Valerio Schiavoni. 2020. FaaSdom: a benchmark suite for serverless computing. In <i>Proceedings of the 14th ACM International Conference on Distributed and Event-based Systems</i> (<i>DEBS '20</i>). Association for Computing Machinery, New York, NY, USA, 73–84. https://doi.org/10.1145/3401025.3401738
8. "Airtime calculator for LoRaWAN." https://avbentem.github.io/airtime-calculator/ttn/eu868, las accessed 2022/05/22.
9. AWS IoT Core Homepage, https://aws.amazon.com/iot-core/, last accessed 2022/05/22.
10. The Things Stack Data Formats Reference, https://www.thethingsindustries.com/docs/reference/data-formats/, last accessed 2022/05/22.
11. Amazon Web Services - Databases, "AWS Cloud Databases," https://aws.amazon.com/products/databases/.
12. Amazon Web Services, "AWS QuickSight," https://aws.amazon.com/quicksight/.