An IoT Agriculture System using FIWARE

Pedro Corista, Diogo Ferreira, João Gão, João Sarraipa, Ricardo Jardim Gonçalves

CTS, UNINOVA, Dep.º de Eng.ª Electrotécnica, Faculdade de Ciências e Tecnologia, FCT, Universidade Nova de Lisboa
Caparica, Portugal

Abstract—The industrial panorama is evolving. Using IoT sensors and actuators it is possible to increase product's value, by controlling its production chain. Internet technologies such as FIWARE can provide the services that modern industry needs to process and evaluate sensor data to apply higher production standards, which increase product value. Although FIWARE enablers provide the tools to fulfil modern industry needs, there is an interoperability gap a gap between applications and enablers, as different enablers have different protocols and needs. This article describes an agriculture system developed using vf-OS (virtual factory Operating System), a platform that aims to become the bridge between applications and enablers, as it provides the means to interact with them. The developed system is composed of different applications, that use enablers (integrated using the vf-OS system), current context management and fruit quality theories, to control product quality during the whole fruit production chain.

Keywords—Agriculture; IoT; FIWARE; FoF; Sensor; Smart Farming; vf-OS

I. INTRODUCTION

With the technological evolution, there is a trend to store all possible data, including the one not used or not needed. These days, social-networks are well-known examples that store all data, for example, all conversations or the dates and geographical location of every login, which are not necessary or not used on the application's usability. These trends' importance is increasing, as the grounds for Factories of the Future are being developed requiring more and more handling of data. The future industry will tend towards flatter management structures with a more highly skilled and IT literate workforce that will be focused on improving the product's development and performance [1]. By developing smart connected products it is possible to create a competitive advantage, regarding the area where such products compete, as seen in [2]. IoT sensors and tools can be used to add and increase product value. Tools such as FIWARE enablers are used to enhance application that manage the product's chain, as they are internet technologies that allow supply chains to use virtualization dynamically in operational management processes[3]. Although enablers are efficient IoT assets, a uniform approach to use them does not still exist, as there exist several different enablers, with different interfacing protocols (eg, NGSI, REST, SOAP) and different installation needs. European researchers are developing vf-OS, a platform that will act as a bridge between enablers a service providers and applications as service consumers.

This paper proposes the vf-OS enablers related approach tested within an agriculture scenario. Although

people do not associate agriculture with industry, much of the work done since growing seeds to the harvest can be optimized using IoT concepts that already exist, applied in industry. By using sensors and processing the information generated by them it is possible to achieve a higher product control, following the same approach as in industry. Following this idea, the authors applied what has been addressed about vf-OS enabler integration to the agriculture industry a well. Strategies to address some agriculture issues are already being developed using FIWARE, as seen in [4]. This document provides an alternative to the existing approaches as it provides a homogeneous solution to address the enablers thus, in the next section some considerations about the enablers used to develop the proposed system, and the relation with vf-OS platform are explained. The third section provides the agriculture scenario and its problematic related to the definition of a FIWARE based system. Section four introduces the several applications that compose the system and provides an insight about how they work. The fifth section provides details about how the system works, regarding each application's role. It also illustrates how the used enablers interact with the developed system, using the vf-OS assets.

II. FIWARE

This section presents the concept of enablers, pointing out to the two most significant enablers used by the system that this paper addresses. Then, due to its direct relation, it also describes the FIWARE project, from which the enablers were defined. It also describes the vf-OS project and its influence to the proposed system definition.

A. The FIWARE Project

FIWARE is a project that aims to the creation of a core platform on the context of the Future Internet Private Public Partnerships (FI-PPP), that was part of FP7 (Seventh Framework Programme) of the European Commission [5]. The programme's final goal was to develop future internet technology which could be used in smart applications and it would contribute for the technological competitiveness and growth of Europe. Using FIWARE became possible to speed up of the design and implementation of architectures and platforms, as seen in [6]. This technology offers many open-source resources and their main components are Enablers. These components are services, developed either for global or specific use in different domains, as for example in manufacturing, agriculture, eHealth or energy domains, which can be used by smart applications to perform specific tasks. One of their main characteristics is to be reusable and composed by shared modules which have interoperable interfaces.

1) Orion and STH

The main FIWARE assets are the enablers, which can be catalogued as generic and specific. Generic Enablers (GE) are designed for a general purpose and independent of any domain area. Specific enablers (SE) are made to be used on industrial areas such as manufacturing, media or agri-food. At the FIWARE website [7] GE and SE can be found, as well as their characteristics and the policy behind their development were. From the several available enablers, the two enablers used by the authors in this research work are presented below.

Orion Context Broker is a GE that is a middleware between consumers producers (e.g. sensors) and context consumers applications (e.g. applications that use sensor data) allowing to perform context management functionalities [8]. As a result, this GE can be used in a data/context scenario to update information, queries, registrations and subscriptions using NGSi9 and NGSi10 interfaces.

The NGSi (Next Generation Services Interface) interface was developed in order to integrate physical devices using an IoT approach to aid the transition from device-level information into thing-level information. NGSi9 standard is used to verify the context availability and NGSi10 is used to perform exchange of information [9].

Orion uses entities to successfully create a virtual representation of the real world where each entity has a unique identification and a type. As an example, an entity virtualization of a book can have as ID his title and type "book". Entities can have several attributes that describe its characteristics with a name, a type and a value. Following the previous example, a book chapter can be described as an attribute, where the chapter title would be the ID, the type would be "chapter" and the value would be the matching pages. One of the main advantages of using this type of relations is to be able to have structured data using generic models (entities) and add new associations (attributes) to these models. This feature is very useful in IoT applications, allowing to categorize all data in separated entities and attributes. All the querying functionalities become more efficient as the amount of data is reduced and it is possible to apply more filters using characteristics of the object to query.

TABLE I. FIWARE STH HISTORICAL AGGREGATION METHODS

Type of attribute	Aggregation method	Explanation
Numeric	Max	Maximum value
	Min	Minimum value
	Sum	Sum of all samples
	Sum2	Sum of the square value of all samples
String	occur	Occurrences of each textual value that attribute have had over the time

To take advantage of Orion's characteristics, it can be used another enabler, the STH. As it happens with Orion, this enabler makes use of the NGSi9 and NGSi10 interfaces and its main capabilities are related to the aggregation of entities information during a defined time. The STH accesses the subscriptions made through Orion and it is informed every time an entity value changes. This enabler also provides history of all the changes occurred in the database, represented by table 1.

With the integration of both presented generic enablers, a system can represent context and, at the same time create a context log. Using the log's information, such context recording can be processed, according to the system's needs.

B. vf-OS project influence

vf-OS – virtual factory Operating System – is an European Project with the objective to develop an Open Operating System for Virtual Factories. This project is specifically designed for the factories of the future and allows to integrate software resources such as FIWARE enablers. By accessing the enabler's capabilities, vf-OS provides the basic elements for developing complex applications. This approach allows helps building the applications from scratch and provides the user with a wide range of tools. The three main vf-OS components can be used by manufacturing and logistic users that aim to explore already created solutions and adapt them to their business area.

The industrial panorama will tend towards smart systems that use IoT sensors to perform real-time monitoring, as seen in [10]. The several applications within a factory will exchange data between them, even if the languages and models in their implementation are not the same, thus achieving interoperability [11]. Vf-OS allows to integrate several tools and resources (e.g. GE), so that it can operate within a network of collaborative manufacturing and logistic environment. By integrating resources and enabling the connection with applications, sensors and devices, it will allow to implement FoF (Factories of the Future) concepts in an IoT perspective to create faster and more efficient systems.

Current industrial centralized systems are pushed to their limits, due to their lack abilities to deal with complex plans, control and changeable control and logistic systems [12]. By linking existing ICT technologies with the vf-OS platform, it is possible to create a decentralized system that overcomes the actual industry limitations. This means that the VF operating system will be the intermediary between applications and devices, being also responsible for managing the virtualization and data exchange of physical elements (hardware), as it would be made in a regular operating system.

Currently, vf-OS provides a framework that allows to integrate different enablers. This Enablers Framework (EF) works following the same principles as a kernel, as it implements the necessary services to access enablers' functionalities. The user can access the functionalities he

Steps	Explanation
1	Fruit harvest, controlled by IoT Devices
2	Fruit selection, separation (manually or automatically)
3	Farmer track of things (amount and quality of fruit, faults and failures, spare boxes, etc.)
4	Buyer selection of fruit type, weight and farmer
5	Buying order
6	Monitored transport to ensure success delivery
i	Monitoring production and checking for failures

needs directly from the framework. Whenever a new user request is performed (it can be done using IoT protocols such as REST or NGSI), it is redirected to the desired enabler using methods analogous to proxy communication component (Figure 1).

This is the platform that the authors intend to use to accomplish the system for the proposed agriculture scenario. Additionally, such solution will include FoF concepts and approaches to the agriculture domain.

Fig. 1. An excerpt of vf-OS platform

III. AGRICULTURE SCENARIO

Modern agriculture uses sensors to grant high product quality to reward farmers with higher profits. The usage of sensors can also be applied in several parts of the product chain increasing the product quality. This sensor technology approach, that makes farms more intelligent, with real-time data gathering, processing and analysis is called 'smart farming' [13].

Fig. 2. Practical scenario illustration [14]

This scenario, illustrated by Figure 2, and detailed in table 2, represents a fruit product chain that deals with the product from harvest to client delivery.

TABLE II. PRATICAL SCENARIO EXPLANATION

After fruit is harvested by the farmer, it is separated (manually or mechanically) and submitted to quality control to confirm that the desired standards have been achieved. The several strains of fruits are split according to the desired characteristics and made available to the client. From the moment that the product has been made available, clients can decide which products to choose according to their needs. To perform this, it was developed a buyer's application that allows, not only to view the farmers who are registered in the platform, but also their products, based on features such as fruit strains and measurements. When a client purchases a product, farmer receives a request and chooses whether to accept it or not. If he accepts the request, the product will be shipped. The transportation phase is also monitored with sensors, to inform the client the fruit's transportation's conditions (Figure 2).

During the entire scenario, it is possible to use several technologies according to the necessary processes, such as harvest, transport, etc. Regardless the process used, it is possible to be aware of the product situation analysis in real time by using the available sensors. It is an improvement to add sensors to control the process, not only to control the product's quality but also determine the product's price, using specific sensors (e.g. associated weather forecast).

To fully achieve the potential of this scenario, there are additional processes that would bring more information to qualify the fruit's quality, such as quality monitoring, packing, monitoring during transportation or product request.

IV. VAPPS FOR AGRICULTURE DOMAIN

Taking into consideration that vf-OS platform supports the development of IoT applications, the scenario presented above was divided into vApps, each one of them with a specific task, as illustrated on [14] and represented by activities interaction in Figure 3.

Fig. 3. Functionalities interaction of the agriculture scenario

vFail and vProductMon reason the harvest phase, based on sensor data from the farmer's plantation. To process such reasoning over the sensor information the user must provide rules and context to the system. VOrder

monitors the fruit quality during transportation. VFNegotiation provides an online platform where the fruits can be sold.

The vApp “vfHarvest” is still in the implementation phase because it requires specific agricultural efforts in order to attach distinct sensors in the plantation. To solve this issue, for the current time, this module provides simulated sensor data. Such simulations used XMPP-based wireless sensors representations. Extensible Messaging and Presence Protocol (XMPP) is a protocol used for real-time information exchange on the internet. By applying that protocol to wireless sensors, it is possible to exchange retrieved data between the user and a group of sensors as seen in [15] and consequently, in this case, have a realistic simulation of a farmer’s crop with distinct sensors.

A. vFail and vProductMon

The vFail and vProductMon applications were developed using the model of a context based framework depicted in Figure 4. This framework was based on the work from [16],[17] and [18] with a layered architecture.

Fig. 4. High level frameworks architecture

The acquisition layer is responsible for receiving the sensor data.

Sensor Data Harmonization Layer is responsible for getting the data from the previous layer and store it in the Sensor Data Repository. The sensor data is submitted for harmonization to be coherent with the framework’s data format. These procedures are mandatory to have an efficient query reasoning in the next layer.

Context Aggregation/Reasoning layer reasons the stored sensor data, based on the context and rules information available in the repository. The evaluation result is then sent to the upper layer.

The Application layer is the User Interface (UI) of this framework that is responsible for creating and deleting Rules and Context elements as well as to know what are the reasoning results.

vFail uses the first three layers of the framework presented on Figure 4. This vApp is used to verify what is

the context accordingly to the data received on the Acquisition Layer. The context is defined by the rules stored in the repository which target specific sensors. In this scenario, this vApp can verify if the fruit harvest matches the expected amount, if it is correctly calibrated or if there are enough boxes to package it.

vProductMon represents the Application Layer as the UI of the framework presented above. Within this application, a farmer can formulate rules for a context, to evaluate a specific case. It is also possible to verify if the requirements regarding a specific set of sensor information have been achieved. This application also allows to share information, regarding fruit evaluation, with vFNegotiation, when necessary.

B. vfNegotiation

vfNegotiation is an vApp developed with the intent of being used by sellers (farmers in this scenario) wanting to get his products out for sale and buyers looking for products to buy.

A farmer that uses this application can add or update new fruits to his stock (each defined by a fruit name, fruit breed and size), set the stock amount or the price for a specific fruit.

The buyer can search for the fruit provider based on several criteria. The application provides the buyer with two different search methods, a manual search, where a list of farmers able to sell the searched fruit is presented and the buyer can choose one of them, or an automatic search, where the buyer chooses the filters and only the best seller who meets the requirements is presented.

C. vOrder

This vApp is able to be used either by a farmer wanting to expedite his products or a transport manager who provides the farmers with the necessary trucks to transport the sold products. This can also be used by a buyer who wants to keep track of his purchased goods’ transport conditions.

The vOrder Application enables the creation of virtual trucks representations, that integrate data from real sensors, which exist inside such trucks. After the creation of the trucks in the system and, added to the farmer’s fleet, they can be tracked by the user allowing to evaluate a particular trip.

V. IMPLEMENTATION

The developed vApps integrate generic enablers and provide a solution for the presented scenario (Section III). Considering its needs, the vApps will integrate ORION and STH enablers. Since they provide the necessary functionalities for all applications that connect to them, they can remove complexity from the vApps or other core components. ORION is used to define and reason rules for the context and to process sensor data. This GE is used by VproductMon to formalize rules and contexts and by Vfail, to apply the formalized to specific data. vfNegotiation uses it as well to define and process buyers

orders and finally vOrder stores and processes sensor data through it. STH is used to aggregate and process sensor datasets. vOrder uses this GE to combine sensor information and calculate probabilistic values, such as average value, variance and standard deviation.

Figure 5 illustrates the developed system architecture, showing the several applications and GEs that may be connected to it. All the applications, that use these mentioned enablers, can be deployed in different machines, granting modularity. This means that farmers and traders (buyers) can use the applications on different facilities. The integration of each GE is assured using the vf-OS proxy communication method as described in Figure 1. When an application needs to use an enabler, it contacts the vf-OS platform, that provides access to the necessary GE's services. The communication between the enablers and the proxy component is made using different standard IoT protocols (depending on the enabler), such as NGSI and REST, achieving an interoperable system.

Fig. 5. High-level system view

A. Applications demonstration

This section demonstrates how the developed applications work. It also illustrates some steps that a user needs to perform to achieve the functionalities described in section 4. The demonstration cases presented below match the points i, 2 and 4 from the presented scenario

(that is illustrated in Figure 2).

Fig. 6. vProductMon - Create a rule

The first step to monitor the production (step i of the scenario) is to define a rule for a context using vProductMon. All rules are composed by four parts: the context to verify (Under Context); the identification (Name); the rule's formalization (Condition); and the target (Sensor Target). Figure 6 presents a rule example that defines that a fruit must be between 50mm and 60mm to have a S size. Once the rule is registered, the application makes a call on the enabler (Orion) to record the information. This is accomplished by showing it to the user. Then, during the fruit calibration, the targeted physical sensor measures the fruit's size and sends the value to the system. Finally, the system reasons the data based on the rule's condition querying the amount of fruit per size. The next example shows how to verify the amount of fruit that was successfully calibrated (step 2 of the scenario). Figure 7 shows the verification of sensor data (vProductMon), using the previously defined rule. From the figure, it is possible to verify the number of fruits under the defined rule.

Fig. 7. Amount of calibrated fruit (vProductMon)

Figure 8 shows the amount and type of fruit that a farmer can sell, using vfNegotiation. Comparing Figure 7 and Figure 8, it is possible to verify that the farmer can use vProductMon to make the fruit available to sell on vfNegotiation.

Fig. 8. Fruit submitted for sale(vfNegotiation)

Finally, the last example shows how a customer can buy fruit using vfNegotiation. The process of choosing and buying a fruit is depicted in Figure 9. First the buyer can search for orange sellers (list obtained through ORION) and check their prices (left side of Figure 9). Then, through a price filter, the buyer can reduce the number of producers accordingly to the chosen parameters.

Fig. 9. Selecting fruit based on farmer (vfNegotiation)

B. Vf-OS as implementation enhancer

As seen in the demonstrations, the developed system applications grant the farmer the means he needs to control and sell his products and also gives the buyer an opportunity to buy specific fruits. Figure 5 shows a practical approach about how the developed applications integrate the enabler's services, using the vf-OS EF asset. Following the same logic, any enabler can be integrated as illustrated in figure 1. The added value of using vf-OS platform is related to the possibility to integrate open source components, following certain conditions.

vf-OS aims to become a business multi-side platform enabling value by creating interactions between external producers and consumers. This approach provides an open, participative infrastructure for these interactions and sets governance conditions for them[19]. Developers can use the platform to find the enablers they need to develop the applications and integrate them. The platform provides embraced functionalities to application's specific domains, which the application owners may need, avoiding a certain amount of software cost.

VI. CONCLUSIONS

The evolving of the industrial panorama leads to the need to apply factories of the future concepts to value products. This article proposed a system that allows to monitor all the necessary steps, within the fruit product chain. The proposed system relies on applications built using vf-OS functionalities and FIWARE enablers resources. By using FIWARE enablers, it is possible to create light applications, as the GE may provide the necessary complex functions to satisfy the core software needs. The access to the necessary GE is made through the vf-OS platform. The vf-OS platform acts as a proxy that routes the application request (through the necessary protocol) to the necessary GE. Thus, it can be used as a core component, to handle any application requests in relation to enablers, regardless the scenario in which that application was developed. By using such an open platform it is possible for small companies stakeholders to achieve high quality standards, that otherwise would be not possible due to the costs of developing everything from scratch. Vf-OS acts as a booster by allowing small and medium companies to achieve competitive products that may integrate open internet tools (enablers).

VII. ACKNOWLEDGEMENT

The research leading to these results has received funding from the European Union H2020 Program under grant agreement No. 723710 "Virtual Factory Open Operating System" (vf-OS).

REFERENCES

- [1] D. J. W. Keith Ridgway, Chris. W. Clegg, "The factory of the future," p. 53, 2013.
- [2] J. E. H. Porter, Michael E., "How Smart , Connected Products Are Transforming Competition," Harvard Bus. Rev. 92, vol. 11, no. November, pp. 64–88, 2014.
- [3] C. N. Verdouw, J. Wolfert, A. J. M. Beulens, and A. Rialland, "Virtualization of food supply chains with the internet of things," J. Food Eng., vol. 176, pp. 128–136, 2016.
- [4] QUHOMA, "QUHOMA project- Developing a QUALitative HOrticulture MArketplace and community based on FIWARE tools." [Online]. Available: <http://tporganics.eu/quhoma-project-developing-a-qualitative-horticulture-marketplace-and-community-based-on-fiware-tools/>. [Accessed: 18-Apr-2018].
- [5] L. Tuominen, "Future Internet in the European Union - Case FI-WARE Future Internet in the European Union - Case FI-WARE," 2013.
- [6] M. Fazio, A. Celesti, F. G. Marquez, A. Glikson, and M. Villari, "Exploiting the FIWARE cloud platform to develop a remote patient monitoring system," Proc. - IEEE Symp. Comput. Commun., vol. 2016–Febru, pp. 264–270, 2016.
- [7] FIWARE, "Generic Enablers." [Online]. Available: <https://catalogue.fiware.org/enablers>. [Accessed: 04-Jan-2018].
- [8] Fiware Docs, "Readthedocs Fiware-Orion," 2017. [Online]. Available: <https://catalogue.fiware.org/enablers/publishsubscribe-context-broker-orion-context-broker>. [Accessed: 05-Jun-2017].
- [9] Fiware.org, "FIWARE NGSI-9 API Core (v1) · Apiary," 2017. [Online]. Available: https://forge.fiware.org/plugins/mediawiki/wiki/fiware/index.php/NGSI-9/NGSI-10_information_model. [Accessed: 05-Jun-2017].
- [10] "The Fourth Industrial Revolution Gains Momentum," Trends Mag., no. June 2016, 2016.
- [11] V. Fortineau, T. Paviot, and S. Lamouri, "Computers in Industry Improving the interoperability of industrial information systems with

description logic-based models — The state of the art,” *Comput. Ind.*, vol. 64, no. 4, pp. 363–375, 2013.

- [12] J. Schuhmacher and V. Hummel, : Decentralized Control of Logistic Processes in Cyber-physical Production Systems at the Example of ESB Logistics Learning Factory., vol. 54. Elsevier B.V., 2016.
- [13] A. Kamilaris, F. Gao, F. X. . Prenafta-Boldu, and M. I. Ali, “Agri-IoT: A Semantic Framework for Internet of Things-enabled Smart Farming Applications,” in *Internet of Things (WF-IoT)*, 2016 IEEE 3rd World Forum on, 2016.
- [14] J. S. and R. J.-G. Diogo Ferreira, Pedro Corista, João Gião, Sudeep Ghimire, “Towards Smart Agriculture using FIWARE Enablers.” *ICE/IEEE Conference 2017*, 2017.
- [15] A. Hornsby, P. Belimpasakis, and I. Defee, “XMPP-based wireless sensor network and its integration into the extended home environment,” pp. 794–797, 2009.
- [16] S. K. Mostéfaoui and B. Hirsbrunner, “Towards a Context-Based Service Composition Framework,” no. June, pp. 42–45, 2003.
- [17] K. G. & F. A. H. Niels Olof Bouvin , Bent G. Christensen, “HyCon: A framework for context-aware mobile hypermedia,” vol. 4568, no. June, 2017.
- [18] M. S. B. and S. M. I. Herma Van Kranenberg, “A Context Management Framework for Supporting Context-Aware Distributed Applications,” no. August, pp. 67–74, 2006.
- [19] N. M. Pedro Corista, Joao Gao, Joao Sarraipa, Raquel Almeida, Oscar Garcia Perales, “Enablers Framework: an approach to develop applications using FIWARE,” *Enterp. Interoperability*, 2018.