

Towards Smart Agriculture using FIWARE Enablers

Diogo Ferreira, Pedro Corista, João Gião, Sudeep Ghimire, João Sarraipa and Ricardo Jardim-Gonçalves
CTS, UNINOVA, Dep.º de Eng.ª Electrotécnica, Faculdade de Ciências e Tecnologia, FCT, Universidade Nova de Lisboa
Caparica, Portugal

{dma.ferreira, p.corista@campus.fct.unl.pt, {jgs, sud, jss, rg@uninova.pt}

Abstract— With the introduction of paradigms like Internet of Things, Cyber Physical Systems and Cloud Computing, Smart Factories are becoming a central part of today’s manufacturing systems. Even though there already are some solutions in the market the full potential for smart manufacturing hasn’t yet been achieved. In order to fulfil the gap European researchers are developing vf-OS, a platform that aims to be the future reference in future factories operating systems. In this work is presented some preliminary results regarding the modules related to IoT, event processing, situational awareness and data harmonization that are being researched in the scope of vf-OS to achieve holistic solution for industries, specifically targeting the agriculture domain.

Keywords— *agriculture; factories of the future; fiware; enabler; smart farming.*

I. INTRODUCTION (HEADING 1)

With the development of technology, new forms of agriculture are emerging in our society. The use of technology is growing in order to improve agriculture production and have higher profits. To achieve these goals, agricultural producers or distributors use Information Technology and Communication (ICT) to help to manage agriculture. Sensors can be used to enable real time monitoring food’s parameters, such as pH, temperature, earth’s moisture or oxygen flow. With these sensors and its connection with the Internet, it is possible to monitor all cultivations, even if they are apart, and predict and control its quality and how much food can be sold. With the knowledge of position, its values and seasonal care, tractors can supply plant’s daily needs, without human intervention. This sensor technology that makes farms more intelligent, with real-time data gathering, processing and analysis is called ‘smart farming’ [1].

Several plants production is increasingly threatened. Climate changes, insufficient available lands, air toxins are some problems that agriculture faces these days. One agricultural solution to address these problems are plant factories. They can be defined as horticulture greenhouses or automated system facilities that have artificially controlled environments in order to produce vegetables and seedling year-round [2]. By using the benefits of smart farming, it is possible to turn the entire growth process automated and leading economies in order to control cost, quantity and quality against the required harvest time.

All these features have been addressed by European Research Initiatives as the European Technology Platform on the “Future Manufacturing Technologies” (MANUFUTURE). This is a project which has the goal to implement research and

innovation methodologies to increase the rate of industrial transformation in Europe [3], where Agriculture Engineering Technologies (‘AET’), even being from a cross-sectorial domain, is taking an effort to be integrated in the future of the industry research [4].

With the uprising of technologies linked to industry 4.0 now it is possible to create virtual factories. Virtual factory is one digital representation of the real factory where actors can simulate and operate the normal operation of the factory, predict, find and solve malfunctions and take advantage of scalability. It is also possible to add new services to the virtual factory and perform impact analysis before applying them to the physical one. To create a global standard virtual factory operating system, the European Union under the Horizon 2020 Programme commissioned the creation of a Virtual Factory Open Operating System called vf-OS.

The vf-OS project aims to provide an economical multi-sided market platform with the aim of creating value by enabling interactions between different customer groups: Software Developers; Manufacturing Users; ICT Providers and Service Providers. In order to create value in such multi-sided platform, vf-OS will provide a range of services to the connected factory of the future to integrate better manufacturing and logistics processes. Additionally, Manufacturing Applications Store (vf-App) will be open to software developers who, using the free Open Applications Development Kit provided, will be able to quickly develop and deploy smart applications to address different aspects of manufacturing industries. Vf-OS solution will provide technology stack for device and legacy system integration, sensorial system virtualization, complex event processing, data harmonization and analytics, situational awareness etc. This work aims to propose a solution for the agriculture industry using FIWARE solutions to implement some core functionalities and being aligned with the proposed methodology formulated by vf-OS.

II. FACTORIES OF THE FUTURE AND INDUSTRIAL INTERNET OF THINGS

According to [5] factories will tend towards flat management structures with a more highly skilled and IT literate workforce focusing on more product design, optimization, monitoring and controlling of processes. These factories will rely on sensors and actuators to allow production monitoring and control systems in real-time. The information coming from the sensors will be correlated with digital models and simulations resulting on a high efficient situational analysis and task management. At the

same time the current trend of increasing development of technology used in factories and in industry in general aims to further improve the way robots and machines in industry are being used. Briefly speaking they are being more and more interconnected and optimized, which are specifically addressed by a branch of the Internet of Things (IoT) dedicated especially to the industry, the Industrial Internet of Things (IIoT). Simply put, such as [6] ascribes to [7], “IoT is an information network of physical objects (sensors, machines, cars, buildings, and other items) that allows interaction and cooperation of these objects to reach common goals”. The IoT has today many applications in different domains such as transportation, healthcare, smart homes and industrial environments [8]. When applied to core industries, it can be called IIoT and its main purpose is to transform the way field assets (e.g., machines or robots) connect and communicate within a factory or between factories, resorting to the use of sensors, advanced analytics, and intelligent decision making [9].

Some efforts are being made to implement the IIoT in today’s industry, starting by addressing the standardization challenges, promoting open interoperability and the widespread usage of a common architecture. Despite all progress and improvement, there are still many ways to where IIoT can be further improved and optimized. According to [10] test-beds for smart production technologies (called experimental factories) have been created with the purpose of establishing interoperability guidelines and applying new IT technologies in existing automated systems, and thus demonstrate how technologies from different organizations can work together and support new innovations. However, still according to [10] has been no attempt to interconnect these experimental factories and allow them to flexibly adapt their production capabilities based on cross-site demands.

In order to manage the usage of all these smart embedded devices in industry, applications become necessary to better integrate real-time state of the physical world, and hence, provide services that are highly dynamic, scalable and efficient. To incorporate these functionalities principles of service oriented computing (SOC) it is necessary to adopt the paradigm of industrial systems. SOC is built upon services that provide autonomous, platform-independent, computational elements that can be described, published, discovered, orchestrated and programmed using standard protocols to build networks of collaborating applications distributed within and across organizational boundaries. This needs to be implemented in the scope of factory systems where infrastructures are composed of large numbers of networked, resource-limited devices and machines. Service orientation for modelling, representation and implementation of such systems can enable such devices to offer its functionalities and, at the same time, discover and invoke others functionalities offered by services of other devices dynamically and on-demand, as suggested by [11].

A. Important FoF Concepts

Based on the findings from the literature review, in [12] are presented the principles that are used to obtain design principles to factories of the future and that are relevant to this work’s main goal. These principles are interoperability, virtualization,

decentralization and real-time capability and are explained below.

Interoperability is the ability of two or more systems to communicate and exchange data, even if the languages and models used in their implementation are not the same [13]. This means that the concept of interoperability is an important factor on factories of the future because it will allow to enable communications between the CPS, the installation network and the people connected through it.

The virtualization of the factories of the future can be achieved using cyber-physical systems that are composed by a physical entity embedded with a cyber entity [14]. The cyber part contains a virtualization of the physical part so that it can replicate virtually any behaviour of the physical machine. Being a virtual/physical system allows the system to easily access and store information in the cloud. CPS also contain wireless embedded wireless devices [15] which allow the various machines to interact and communicate with each other and with the network.

III. FIWARE

FIWARE is a research driven project with the goal for creation of a core reference platform in the scope of Future Internet [16]. The project has delivered a stack of solutions for internet technology which can be used in smart infrastructures of different domains and that will contribute for the technological competitiveness and Europe’s growth.

FIWARE offers customized open-source resources through functional building blocks that are called Generic Enablers (GEs) that ease creation of smart Internet Applications and services. These GEs offer reusable and shared components, which provides a concrete set of APIs and interoperable interfaces that are in compliance with open specifications published for that GE [17].

The FIWARE GEs are divided into technical chapters, each one of them has its technological focus. For this paper, it will be analyzed the GEs from FIWARE chapters: “IoT Services Enablement” and “Data/Context Management”. IoT Services Enablement chapter provides the GEs to allow any physical object to become available, searchable, accessible, and usable context resources fostering Apps interaction with real-life objects. On the other hand, the Data/Context Management enablers ease development in applications that require management, processing and exploitation of context information as well as data streams in real-time data. Together, these enablers will help achieving functionalities to collect and analyze a large quantity of information. Additionally, FIWARE data/context management chapter looks forwards to develop GEs that provide an easier and faster way to create or integrate data analysis algorithms [18].

A. IoT Enablers examples

IoT enablers provide functionalities for connecting devices using standard IoT communication protocols and providing interfaces for the enablers at data/context layer. Basically, IoT GEs are spread over two different domains i.e. IoT Backend and IoT Edge. IoT Backend comprises the set of functions, logical resources and services hosted in a Cloud mainly focused on

management of IoT devices. While, IoT Edge is made of all on-field IoT infrastructure elements needed to connect physical devices to other applications. The GEs of interest are Protocol Adapter - MR CoAP and Backend Device Management - IDAS which fall in IoT Edge and IoT Backend domain respectively.

1) Protocol Adapter- MRCoAP

The “Protocol Adapter -MRCoAP” GE serves in between of the registered devices (e.g. the patient sensor) and the gateway that offers the communication layer to the host environment. It supports data collection, sensor detection and connectivity. The particular GEi for instantiation is MR CoAP Protocol Adapter, which provides functionality to plug devices using on CoaP over 6LoWPan protocol. The protocol adapter is designed to work with IBMs Moterunner platform and communicates via 6LoWPAN and uses CoAP as application layer protocol. This GE is linked with the Gateway Device Management GE or the Data Handling GE [19].

2) Backend Device Management -IDAS

The “Backend Device Management -IDAS” GE allows communication between backend and devices (e.g. via the protocol adapter). This GE provides functionalities to connect physical devices to a FIWARE platform. Other functionalities of this GE includes manages IoT-related NGSI Context Entities, handles the connections and provides IoT Integrators with the ability of transforming devices specific Data Models into the Data Models defined at the NGSI level by different verticals (Smartcities, SmartAgrifood, Smartports, etc.). The GEi that can be utilized is Backend Device Management - IDAS that provides functionality to translate IoT-specific protocols into the NGSI context information protocol, which is the FIWARE standard data exchange model. You do not need this component if your devices or gateways natively support the NGSI API [20].

B. Data/Context Management Enablers examples

The Data/Context Management GEs provide functionalities that will ease development and the provisioning of applications that require management, processing and exploitation of context information as well as data streams in real-time and at massive scale. In this regard two of those enablers, are: FIWARE Publish/Subscribe Context Broker Generic Enabler - Orion Context Broker and FIWARE Short Time Historic (STH) – Comet and are described below.

1) FIWARE Publish/Subscribe Context Broker Generic Enabler - Orion Context Broker

The Orion Context Broker is an implementation of the Publish/Subscribe Context Broker GE used to develop a Data/Context Scenario through the NGSI9 and NGSI10 interfaces (Next Generation Services Interface).

In order to deal to physical devices through an Internet approach i.e. to create the so called IoT devices, FIWARE associated with OMA (Open Mobile Alliance) to create the NGSI concept which enables a transition from device-level information to Thing-level information and vice versa. In order to accomplish so, two interfaces were created NGSI9 and NGSI10. Both are RESTful API's via HTTP but with slight differences. While NGSI10 purpose is to exchange context

information, the NGSI9 is used to exchange information about the availability of context information.

Using both the described NGSI interfaces the FIWARE Orion Context Broker makes possible the creation of entities with all the listed information. Making it also possible to update that information and to subscribe to that information (be informed when a change has occurred). The FIWARE Orion Context Broker serves than an intermediate crossing point between the Context Producers and the Context Consumers, storing all the information in a Database (MongoDB is the database used by Orion by default) which can be accessed by both the Producers (to update or generate information) and the Consumers (to query the entities or be notified through the subscriptions).

2) FIWARE Short Time Historic (STH) – Comet

The Short Time Historic is a FIWARE component capable of managing (storing and retrieving) historical raw and aggregated time series information evolution in time of context data (i.e., entity attribute values) registered in an Orion Context Broker instance. Similarly, to the FIWARE Orion Context Broker, all the communications performed by the FIWARE STH make use of the NGSI9 and NGSI10 interfaces.

As stated, even though the FIWARE STH supports the storing and retrieval of raw context information, (the concrete entity attribute value which were registered in an Orion Context Broker instance over time), its main capability and responsibility is the generation of aggregated time series context information about the evolution in time of those entity attribute values. In order to keep track of the different values a certain attribute takes over time, the STH makes use of the Orion Context Broker subscription function, so that every time that attribute's value changes, the STH can be informed that the change occurred and to which value the attribute has changed into. With this information arriving every time a change of values occurs, the STH keeps a history of the changes in a database (MongoDB by default).

Using an HTTP RESTful API, external clients can query the available historical raw context information maintained by the FIWARE STH. By doing so the consumers get a list of the different values that attribute has taken over time and at what time the changes occurred. These results can be filtered to be shown, for example, the changes the occurred in the last hour or day, or the 10 last changes that happened, or from a specific date to another specific date, etc. On the other hand, if it is historical aggregated time series context information that the consumer is after, the FIWARE STH, also through an HTTP RESTful API, has many ways of interpreting the information and returning it through many different aggregations.

IV. IOT AS A SERVICE

The Internet of Things presupposes the interaction between physical world, objects and their virtual counterparts, allowing these objects to capture and send information to the Internet. Or as defined by [21], “the Internet of Things (IoT) consists of networks of sensors attached to objects and communications devices, providing data that can be analysed and used to initiate

automated actions”. In order to integrate all this physical objects with the digital world, it is necessary to find a way to make them accessible in the Internet, and therefore an Internet of Things as a Service approach is necessary.

1) Model

The first step is the creation of a model through which the physical “things” can be found and accessed. According to [22], the research so far developed in this area, has focused mainly on “sensor descriptions and observation data modelling” that offer sensor measurement data services on the web.

2) Framework

The second step in the Internet of Things as a Service paradigm is the creation and usage of a Framework. In order to make the paradigm accessible to everyone, and to make sure all will benefit from it, an open service framework needs to be created. According to [23], even though there have already been made some IoT Frameworks, those were mainly created by big enterprises such as governments or companies and are mostly based on B2B (Business to Business) and B2G (Business to Government) business models. However, there are also open service platforms for IoT such as Cosm, former Pachube or EVERYTHING.

3) Mashup

The idea behind a “Mashup” is to create new content by reusing or recombining previous existing content from various sources. A Mashup is, therefore, a way to compose a new service from existing services and, “when applied directly to the Web domain, a Mashup is a Web-based application that is created by combining and processing on-line third party resources that contribute with data, presentation or functionality” [24].

Unlike regular Web Services that are provided in a specific domain and available at any time in the web, IoT devices, providing a service, are not always available and not always working in the same place, in addition to the number and variety of connected devices. To solve this problem, a solution is presented in [25], where an IoTaaS (IoT Mashup as a Service) is introduced. IoTaaS is defined as “a mashup of things, software, and computation resource”, and is presented as a cloud-based IoT mashup service model. In IoTaaS, thing is described as “any identifiable object which can have sensing and actuation services; software is an “assembly description of software components”; and computation resource is “a current computer model consisting of CPU, memory, disk, persistent storage, network, etc. How these components interact with the IoT world is depicted in Fig. 1.

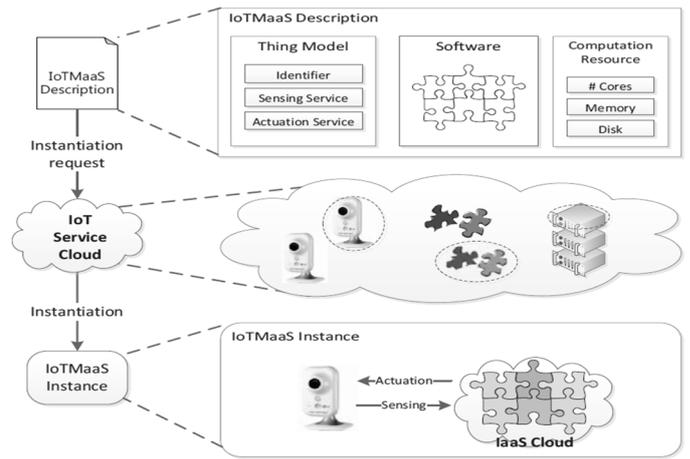


Fig. 1. IoTaaS concept, adapted from [25]

When applied to the Internet of Things, the IoTaaS whose concept is depicted in Fig. 1, allows every IoT device to provide its service disregarding which platform it works on or which protocol it uses to communicate, because regardless of this vast heterogeneity of devices they are all treated equally by the IoTaaS. Therefore, users and or developers can make use of all their IoT devices and enhance their functions resorting to this Mashup. Technically, the enablers from the FIWARE IoT enablement services viz. Protocol Adapter –MRCoAP and Backend Device Management- IDAS as discussed previously will provide core functionalities for realization of IoTaaS.

V. vf-OS

The European vf-OS Project aims to be an Open Virtual Factories Operating System, including a Virtual Factory System Kernel (vf-SK), a Virtual Factory Application Programming Interface (vf-API) and a Virtual Factory Middleware (vf-MW) specifically designed for the factories of the future. Positioning of these different components in layered view of vf-OS is as show in Fig 2. It can be noted that the App are the applications that will be developed to implement different functional/business requirements while other components provide core functionalities envisioned in general for manufacturing processes.



Fig. 2. High level view of different layers of vf-OS

This Open Framework will be able to manage network of collaborative manufacturing and logistics environment, and therefore enable humans, applications and devices to communicate and interoperate in the interconnected operative environment. Additionally, vf-OS will provide a set of Open Services, rooted in the cloud and instantiated at the vf-OS Platform, moving the industry from the device-centric to the user-centric paradigm. This Open Platform is to be linked by

strong and advanced ICT (like CPS, IoT Cloud-Models, M2M, etc.) in order to fulfil the actual need on the market for open services interoperability based on data exchange. Vf-OS will act as an intermediary between the applications of the factory and the factory devices, eventually enabling factory functionalities, services, devices etc. to be virtualized, executed and accessed in ubiquitous manner. Similarly, to a regular OS the vf-OS comprehends core functionalities, but mainly focused in a manufacturing environment. An analogy between some components of a regular Software OS and the vf-OS Manufacturing OS is presented in Table I (adapted from the vf-OS wiki glossary).

TABLE I. ANALOGY BETWEEN SOFTWARE OS AND VF-OS MANUFACTURING OS

Software Operating System	Virtual Factory Operating System (vf-OS)
Kernel	Virtual Factory System Kernel (vf-SK)
Processor, Memory, Internal Bus	Framework, Generic Enablers, Manufacturing Enablers
I/O	Virtual Factory Application Programming Interface (vf-API)
Interfaces, Peripherals, Device Drivers, APIs	Devices Drivers, APIs Connectors, Security & Data Access
File and Data handling	Virtual Factory Middleware (vf-MW)
Interfaces	Data Infrastructure Middleware, Data Storage, Data Harmonisation, Data Analytics

As described before, the vf-OS intends to be the reference system/platform for providing common services for manufacturing industry. This solution will be the part of system software in a real industrial. In order to accomplish this, some generic case scenarios have been established which can be addressed with the vf-OS Platform and its Smart Applications.

The current industrial environment faces some problems related to specific industries and respective sectors some of which are addressed by vf-OS use cases. The developed scenarios integrate both industrial and user scenarios as they propose to produce advanced technical solutions to some of the existing industrial scenarios by developing suitable applications, whereas the user scenarios are described following a standard methodology through well-defined objectives, processes, actuators and possible sets of data. The scenarios addressed by the vf-OS project capture the needs of different industrial sectors and process domains. Table II, briefly explains some of generic apps that vf-OS aims to develop that are equally applicable for the research domain of this paper.

TABLE II. EXAMPLES OF VF-OS GENERIC USE CASE APPS

Name	Description
vOrder	A platform accessible to the farmers who shipped goods in order for them to control and verify the transport conditions and the safe arrival of the products. Platform also accessible to the transport providers for them to be able to create and delete trucks which can be used by the farmers.
vfNegotiation	The buyer must choose, between all its resellers, the most desirable one, according to different factors. In most of the interactions, between one agent who produces the goods and the other agent who buys the

	goods, the purpose of the negotiation is to make the better negotiation choice, between several agents.
vFail	Identifies if all the requirements regarding the processes of fruit calibration and selection have been achieved. This tool will act whenever new sensor values are retrieved and will find all the context that affects them. This provides an extra insurance regarding the product quality that aims to help a producer to have a higher fruit quality level compared to other fruit producers
vProductMon	A farmer can submit a context and it's corresponding rules within the system which means that he can have control above every quality checkpoint within the system. Because vProductMon communicates with vFailure it will display any context evaluation in real-time.

VI. THE AGRICULTURE SCENARIO

In order to further explain the practical application of the vf-OS paradigm, let us consider a scenario in the domain of agriculture industry. This scenario is simply described as a real fruit production chain.

The process begins with the harvesting of various types of fruits by several different farmers. The collected fruit is then manually or automatically split according to different factors (type, size, weight, brand, quality, etc.), and stored in boxes grouped by the desirable factors. When a buyer seeks a buy with a specific type of fruit with specific features, he/she can make use of a buying application where he can find the sellers of the fruit with those specifications and from where he can emit an order to acquire the fruits taking into account the price of the fruit, the seller, the quality and so on. Once the buying order reaches the producer, the latter dispatches the stock as described in the order.

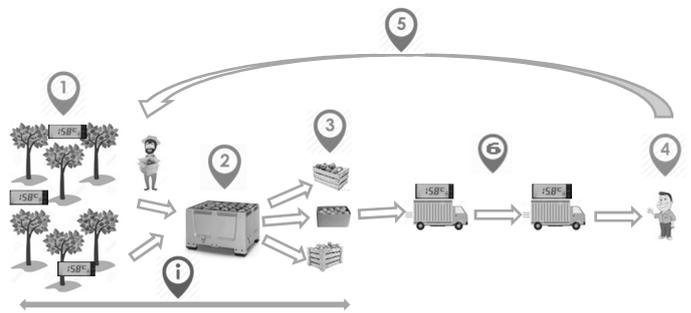


Fig. 3. Practical Scenario Illustration

All the steps are enumerated in Fig. 3 and are listed and briefly explained in Table III.

TABLE III. SCENARIO STEPS EXPLANATION

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 type of fruit, weight and farmer
5	Buying order
6	Monitored transport to ensure success delivery
i	Monitoring production and checking for failures

In the course of these steps different technologies can be used for harvesting, sorting, packing, tracking etc. The producer, while dispatching its fruits is able to control, in real time, all the steps taken by the fruit since the its harvest to the time the fruit is delivered to the buyer. Besides, he is always aware of any faults or failures occurring during the process as well as keeping track of the amount of fruit that is being harvested, shipped, or removed due to not being in conformity with the quality standards. At the end of this process, the buyer can have all information of what was bought, i.e. access to all information from the various process in the food chain from agricultural production to consumption (which is commonly called “from farm to fork”) [3].

This scenario demands different types of functionalities such as transportation tracking, packing status tracking, environmental monitoring of transportation condition, create buying order, dynamic notifications of production/delivery status etc. These types of functionalities are provided by vf-OS apps. In Fig. 4 it is presented the connections between the real scenario and the applications that answer to the generic Use Case Scenarios. All applications are connected in the generic use case scenario.

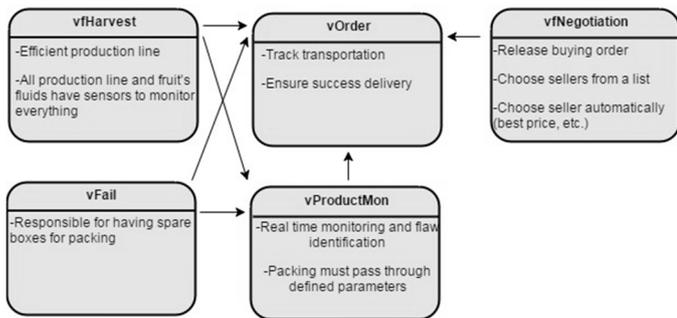


Fig. 4. Use Case Scenarios Applications applied to the Practical Scenario

Comparing with ancestral or traditional agriculture approaches, where the entire farming and harvesting are handmade and it is not possible to get explicit information from the plantation’s conditions, the methodologies presented in this scenario will bring an automatic and better overview of all the agriculture supply chain that will facilitate its improvement. Nevertheless, this scenario could be always enhanced with the addition of new types of sensors and greater controlling procedures, which may include data mining that aggregated with a recommender component, would bring a more autonomous and efficient function to the proposed system.

VII. IMPLEMENTATION

Using the FIWARE Generic Enablers and other technologies it was possible to create answers to the Use Case Scenarios applications presented in the vf-OS project, always bearing in mind an answer to the described practical scenario. Four FIWARE GEs were used (MRCoAP, Orion Context Broker, STH – Comet and IDAS).

In Orion’s model we can represent several entities associated with resources that are accessed by services. The entities allow creation of model of a sensor, along with the value retrieved by

the physical device. By querying features on this enabler and applying certain filters on sensors’ values we can collect data from the sensors as necessary for specified scenarios. Fig. 5 shows a very simple case of querying data collected by an entity which happens to be a sensor.

In the use case scenario, the STH-Comet is used to process the information stored within the repository, comparing it with time periods. It allows to aggregate information retrieved in a particular period of time which allows to drop meaningless information. It also allows the system to perform several sample captures within a defined timestamp. The use of these enablers provides competitive testing options by providing its own nodes and networks. The sets of tools provided allow users to access and use high level functions/applications that otherwise would have to be built from scratch.

```

{
  "contextElement" : {
    "type" : "sensor",
    "isPattern" : "false",
    "id" : "temperaturesensor1",
    "attributes" : [
      {
        "name" : "valor",
        "type" : "int",
        "value" : "3"
      }
    ]
  },
  "statusCode" : {
    "code" : "200",
    "reasonPhrase" : "OK"
  }
}
  
```

Fig. 5. Orion query result for a specific sensor

The MRCoAP enabler is used for connecting various low powered devices that follow CoAP communication protocol. This class of devices includes sensor nodes and actuators that can be used for collecting data from the floor and passing actions to be undertaken by devices respectively. Note that if the devices are capable of communicating with internet protocols then they don’t need to use the MRCoAP enabler. IDAS enabler is used for management of different devices deployed on the floor. IDAS also acts as the interface between the devices and orion context broker enabler. In this case the devices are the data producers and orion is the data distributor which are consumed by different vApps.

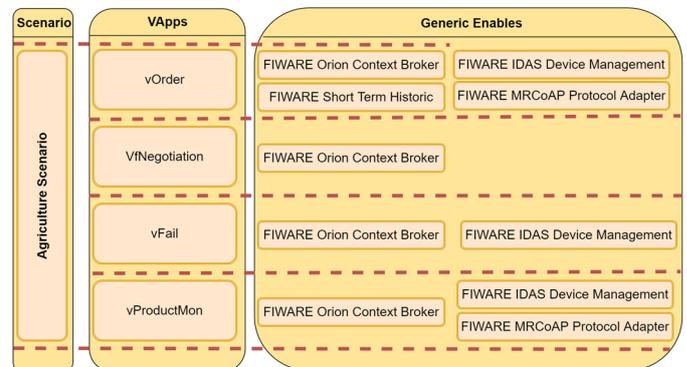


Fig. 6. Scenario + VApps + Generic Enablers

By utilizing these different enablers, the vApps as described can be implemented. Fig 6 shows the mapping with various GEs that will be used for implementing the apps. For instance, vOrder app provides functionalities for tracking transportation and ensures successful delivery. Firstly, to track transportation devices like GPS, speedometer etc. needs to be connected to the truck and eventually connected to the Internet. These functionalities can be achieved by using IDAS and MRCoAP GEs. Then, to find the current status of the delivery, Orion and Comet will process the data. Orion can also be utilized to generate notifications of different events that can be detected by time series analysis functionality provided by STH-Comet. As an example of the implementation that is already being developed, Fig. 7 shows a functionality of the vOrder-based application.

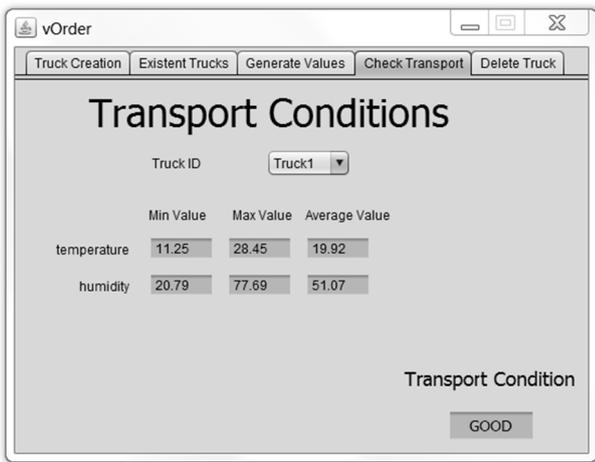


Fig. 7. vOrder application functionality example

The same mentioned in Fig. 7 presents the layout of the application during normal operation of one of its tasks, the Checking Transport Conditions to be precise. Making use of this vOrder application's functionality the user can check in real time the transport conditions of the goods being carried by a truck. These include the values that truck's sensors have read and the maximum, minimum and average value of the last trip. Besides the numeric information, the application also shows a qualitative evaluation of the transportation conditions at the checking time.

VIII. CONCLUSIONS

Next generation solutions for smart industries should provide functionalities for machine-to-machine (M2M) communication, advanced human-machine-interaction, and data analytics through the device management. Additionally, it is necessary to provide plug-and-play connections based on a standardised toolkit among the different Virtual Factory I/O devices (sensors and actuators). In such case, the device manager will be responsible for establishing an efficient integration between factory devices and apps, enabling an efficient response to changing designs at the virtual factory level, and allowing a scalable platform through plug-and-play capabilities. Concepts from IoT as service will be very important to achieve flexible and scalable IoT based solutions.

On this regard vf-OS captures the notion of a manufacturing "operating system" and makes full use of the "concepts" of a classical software operating system such as that on a desktop computer and provides the necessary tools to realise it either in-cloud or on-premise. This manufacturing approach can enable a whole new level of flexibility and scalability in the manufacturing domain. In this paradigm, to support the core functionalities of vf-OS, FIWARE provides technical solutions that provide implementations for IoT service enablement, complex data processing, context management etc.

The research work presents possible solution that could be a successful strategy for innovation on smart applications for agriculture industry that will support collaborative production and logistics processes in value chains. The proposed solution will provide efficient and innovative cloud-enabled tools to collaborate and optimize logistics assets, involving the whole supply chain in the agriculture industry which is explained with the scenario being tested and within the scope of this research work. A range of services to enable connected fields of the future to integrate better production and logistics processes will provide space for independent developers to develop smart applications and create value over the processes and data from selected industrial domain. These smart applications will provide functionalities such as demand forecast, planning, supply, manufacturing, distribution, storage, and replacement / recycling etc. in all the stages of production, delivery and post-production processes. In order to validate the proposed solution, we have planned to develop all the apps as mentioned above and experiment with some real cases of agriculture industries within the scope of the project virtual factory operating system (vf-OS).

ACKNOWLEDGMENT

The research leading to these results has received funding from the European Union H2020 Programs under grant agreements No 611312 vf-OS, as well as from FCT - Fundação para a Ciência e Tecnologia, research unit CTS - Centro de Tecnologia e Sistemas - reference number UID/EEA/00066/2013.

REFERENCES

- [1] A. Kamilaris, F. Gao, F. X. Prenafeta-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.
- [2] J. Hirama, "The History and Advanced Technology of Plant Factories," *Environ. Control Biol.*, vol. 53, no. 2, pp. 47–48, 2015.
- [3] European Commission, *Factories of the future - Multi-annual roadmap for the contractual PPP under Horizon 2020*. 2013.
- [4] European Commission, *Factories of the Future PPP Strategic Multi-annual Roadmap - EUR 24282*. 2010.
- [5] D. J. W. Keith Ridgway, Chris. W. Clegg, "The factory of the future," p. 53, 2013.
- [6] S. Jeschke, C. Brecher, T. Meisen, D. Özdemir, and T. Eschert, "Industrial Internet of Things and Cyber Manufacturing Systems," in *Industrial Internet of Things: Cybermanufacturing Systems*, S. Jeschke, C. Brecher, H. Song, and D. B. Rawat, Eds. Cham: Springer International Publishing, 2017, pp. 3–19.

- [7] L. Atzori, A. Iera, L. Atzori, A. Iera, and G. Morabito, "The Internet of Things : A Survey The Internet of Things : A survey," *Comput. NETWORKS*, no. November 2014, 2010.
- [8] A. Whitmore, A. Agarwal, and L. Da Xu, "The Internet of Things — A survey of topics and trends The Internet of Things — A survey of topics and trends," no. April 2014, 2016.
- [9] S. Wang, J. Wan, D. Li, and C. Zhang, "Implementing Smart Factory of Industrie 4.0 : An Outlook," vol. 2016, 2016.
- [10] J. Jung, K. Watson, and T. Usländer, "Design of Smart Factory Web Services Based on the Industrial Internet of Things," no. Iic, pp. 5941–5946, 2017.
- [11] S. Karnouskos, O. Baecker, and L. M. S., "Integration of SOA-ready Networked Embedded Devices in Enterprise Systems via a Cross-Layered Web Service Infrastructure," 2007.
- [12] T. Hermann, M.; Pentek, "Desingn principles for industrie 4.0 scenarios: a literature review," no. 1, p. 15, 2015.
- [13] 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.
- [14] S. Wen and G. Guo, "Control and resource allocation of cyber-physical systems," *IET Control Theory Appl.*, vol. 10, no. 16, pp. 2038–2048, 2016.
- [15] B. Bordel Sánchez, R. Alcarria, D. Sánchez de Rivera, and A. Sánchez-Picot, "Predictive algorithms for mobility and device lifecycle management in Cyber-Physical Systems," *EURASIP J. Wirel. Commun. Netw.*, vol. 2016, no. 1, p. 228, 2016.
- [16] L. Tuominen, "Future Internet in the European Union - Case FI-WARE Future Internet in the European Union - Case FI-WARE," 2013.
- [17] A. García Vázquez, P. Soria-Rodriguez, P. Bisson, D. Gidoïn, S. Trabelsi, and G. Serme, "FI-WARE Security: Future Internet Security Core," 2011, pp. 144–152.
- [18] FIWARE, "Overall FIWARE Vision," 2015. [Online]. Available: http://forge.fiware.org/plugins/mediawiki/wiki/fiware/index.php/Overall_FI-WARE_Vision. [Accessed: 20-May-2017].
- [19] FIWARE, "Protocol Adapter - MR CoAP," 2016. [Online]. Available: <https://catalogue.fiware.org/enablers/protocol-adapter-mr-coap>. [Accessed: 25-May-2017].
- [20] FIWARE, "Backend Device Management - IDAS," 2016. [Online]. Available: <https://catalogue.fiware.org/enablers/backend-device-management-idas>. [Accessed: 25-May-2017].
- [21] D. Lake, A. Rayes, and M. Morrow, "The internet of things," *Internet Protoc. J.*, vol. 15, no. 3, pp. 10–19, 2012.
- [22] M. Bauer, E. Martinbauerneclabeu, and S. Meissner, "Service Modelling for the Internet of Things," in *Proceedings of the Federated Conference on Computer Science and Information Systems*, 2011, pp. 949–955.
- [23] J. Kim and J. Lee, "OpenIoT : An Open Service Framework for the Internet of Things," in *2014 IEEE World Forum on Internet of Things (WF-IoT)*, 2014, pp. 89–93.
- [24] A. Koschmider, V. Torres, and V. Pelechano, "Elucidating the Mashup Hype: Definition , Challenges , Methodical Guide and Tools for Mashups," in *Proceedings of the 2nd Workshop on Mashups, Enterprise Mashups and Lightweight Composition on the Web (MEM 2009) held in conjunction with 18th International World Wide Web Conference (WWW 2009), April 20th, 2009, Madrid, Spain, 2009*, p. 8.
- [25] J. Im, S. Kim, and D. Kim, "IoT Mashup as a Service: Cloud-based Mashup Service for the Internet of Things," in *2013 IEEE International Conference on Services Computing*, 2013, pp. 462–469.