Supporting a Cloud Platform with Streams of Factory Shop Floor Data in the Context of the Intustry 4.0

Wael M. Mohammed, Borja Ramis Ferrer, Umer Iftikhar, Jose L. Martinez Lastra
FAST-Laboratory
Tampere University of Technology
PO Box 600, 33101 Tampere
{wael.mohammed, borja.ramisferrer, umer.iftikhar, jose.lastra}@tut.fi

Javier Hitado Simarro
ATOS Research & Innovation
ATOS
Bilbao, Spain
javier.hitadosimarro@atos.net

***Abstract*—Industry 4.0 is about the interconnectivity and digitalisation of industrial systems that need to be integrated in order to improve the efficiency of resources and, in turn, processes. Both research and commercial sectors are working towards addressing specific challenges, such as data modelling, collection and processing. A correct manipulation and interpretation of data is critical and, now, more difficult than ever due to the dramatic increment of the amount of data generated at different levels of enterprises. Ultimately, this research work presents a solution, integrated with an existing cloud-based platform, for collecting and processing real-time factory shop floor streams of data. Such solution is an IoT-based development, which consist on both IoT hub and gateway that permit the consumption and communication of device information. The required message exchange is done within state of the art technologies and protocols e.g., MQTT protocol and REST-based interface. The implementation of the solution is demonstrated through an industrial-based scenario.**

Keywords—Industry 4.0; Industrial Internet of Things; Industrial Cyber-Physical Systems; cloud computing; real-time data collection.

I. INTRODUCTION

The interconnectivity, digitalisation and computation of industrial resources and systems requires the implementation of new solutions, highly dependent on Information and Communication Technologies (ICT) [1]. Basically, the provision of equipment-specific interfaces that permits its connection with other systems is needed for handling and computing the generated shop floor data by other systems that might be located even in different geographic areas. In this context, the concept of collaborative networks emerged as one of the ways to provide a common platform for sharing information in the value network of enterprises [2].

Recently, a project named as Cloud Collaborative Manufacturing Networks (C2NET)[1] has been demonstrated and validated by the European Commission. The C2NET project is concerned on the integration of different parties and systems that operates for the same supply chain [3], [4]. More precisely, the integration is done for optimising the processes that are performed throughout the complete product value chain. The ultimate solution is a cloud-based platform that is capable of providing communication adapters, modules and applications that permits the interconnection of multiple interested parties. One of the major challenges was the provision of an approach for collecting and preparing factory shop floor data to the platform users. This is the part of the platform where the result of this research work is employed.

The approach presented in this paper permits the collection, processing and further analysis of factory data streams. More precisely, the data is generated by machines that, in turn, are connected to IoT-based devices, located at the so-called device layer, allowing the report of formatted data through the described ICT-based solution. Such provided solution consists, besides the devices, of two main IoT-based components: IoT hub and IoT gateway. In addition, the manuscript describes the required elements to be deployed at different layers: connector, data handling functionalities, communication and device layers. The required protocols, technologies and APIs are also described and explained. This solution has been integrated into the last and valid C2NET platform.

The rest of the manuscript is structured as follows: Section II presents some industrial practices in the scope of Industry 4.0. Then, Section III describes the FIWARE IoT hub, as existing work that has been used during this research work, and the architectural components required for developing, implementing and deploying the new approach and its integration with the C2NET solution. With objective of providing an illustrative example, this manuscript presents the implementation of the described approach and its deployment in an industrial-based scenario, the FASTory assembly line. Then, Section V presents the obtained results and discusses about the solution benefits. Finally, Section VI concludes the research work.

II. INDUSTRIAL PRACTICES TOWARDS THE INDUSTRY 4.0

The remarkable leap in the ICT domain permit for elevating the concept of manufacturing systems. Thanks to the high interoperability and fast computing, applications became more involved in systems integration [5], [6]. As an example, the concept of Cyber Physical Systems (CPS) and the Internet of Thing (IoT) among others [7], [8]. By combining these concepts and technologies or combining some of its' features

---

[1] http://c2net-project.eu/

new systems and approaches are presented [9]. In this context, the Industry 4.0 is one of the examples of that. In addition, the Reference Architecture Model for Industry 4.0 (RAMI4.0) highlighted on the generic level the lifecycle and hierarchy levels of the reference architecture. This section tends to highlight some of the main concepts that are featured in the Industry 4.0 paradigm [10]. More precisely, this section sheds the light on the connected factories and IIoT (Industrial IoT) concepts.

A. Cloud Platforms for connected factories

The concept of the connected factories is advancing with the progress of the development of the interoperability level of the devices and systems [11]. In this context, several research works have been conducted in terms of defining the bases for such concept. Similar to social media platforms, researchers tend to apply the concept of collaborative networks for connecting factories and manufacturing systems for better productivity and the efficiency in general [5], [12], [13]. In fact, the C2NET project presented an approach for supporting end users with a collaborative platform for addressing the supply chain interactions by providing optimization and monitoring production, logistics and delivery plans [14].

On the other hand, cloud-based solution requires special measure regarding privacy and security of the end users' data. Thanks to the developed protocols and standards, cyber-attacks and threats can be reduced by using secured protocols, encryption and encoding techniques for i.e. HTTPS (Hypertext Transfer Protocol Secure), WSS (WebSocket Secure), and AES (Advanced Encryption Standard) [15]. With relation with RAMI4.0, connected factories and Cloud Platforms represents the upper level on the hierarchy axis [16]. To achieve this level of connectivity, data should be interchangeable and accessible through the different layers for more transparency and reachability [17].

B. Industrial Internet of Things

During the 80s of the previous century, the concept of smart and connected devices was envisioned in [18]. Later, and with evolvement of the networks, applications, devices and computers, light and mobile devices that are capable to exchange and process data has been introduced. In fact, these devices, which known as IoT devices, represent nodes in a distributed system that is capable of sharing data and knowledge through the web. In [19], the approach presents the usage of smartphones, which acts as IoT devices, to monitor the traffic in the roads.

As industrial domain exploits the advances in different fields, such as Computer Science and ICT, the IoT concept is not an exception. Recently, the term IIoT (Industrial IoT) and ICPS (Industrial CPS) have been introduced to address the implementation of web service enabled devices that are capable of exchange, analyse and process data in an industrial environment where the robustness, safety and durability is required [20], [21]. With this technology-merging, industrial and manufacturing systems become more flexible and technology-independent where web services work as bondage for the different systems regardless of the differences in the hardware and software implementation.

III. THE FIWARE IOT HUB

The FIWARE IoT Hub is a middleware component to support continuous real-time data collection from IoT based resources. In a typical IT infrastructure, it should be located at gateway level, between data producers (devices) and data consumers (data analytics). Deployable in low power devices, it is designed to support the following features:

- Interoperability, supporting well-known IoT protocols facilitating the communication between different systems.
- Scalability, adaptable to support different data loads situations
- Plug & Play, offered under a simple API, it is easy to use and install.

Figure 1 depicts the High-Level Architecture of the component, that it will be explained in detail in the following chapters.

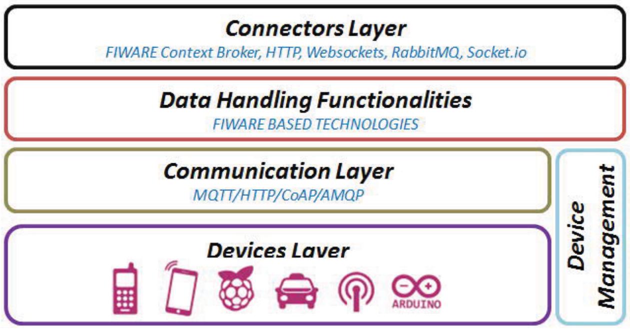

Figure 1. FIWARE IoT high-level architecture

A. Architectural components

A more detailed architecture, including the internal modules of the components is presented in the Figure 2. As the figure presents, FIWARE IoT is composed by a set of components with the aim of tackling three different issues: i) communication; ii) device management; iii) data handling. In terms of communication, it grants interoperability and adaptation between protocols and the matching between data producers and consumers. In regards of the management of devices, it supports the virtualisation of the devices enclosing their generic information as well as addressing their connectivity. With respect to data handling functionalities, it assures that the data obtained from the IoT world is pre-filtered before being passed to the data consumer, reducing the flow or quantity of inaccurate data.

In the lower or first layer of the component, where the communication with the resources is managed, resides the IoT Agent, responsible for the virtualisation of devices, converting them into Context Entities using the FIWARE data models[2]. This module offers a RESTful API to manage the full-cycle of the devices (register, retrieve, update and delete). At this level, and as depicted in Figure 2, several IoT protocols are

---

[2] https://www.fiware.org/data-models/

supported, each protocol has its specific agent. This subcomponent is designed following a micro service-oriented architecture, resulting in high modularity and scalability. Independently of the protocol chosen, the data model to be followed by the data sources has to be same, thus, facilitating data interoperability. For the realisation of this manuscript, the MQTT agent has been selected and used. When a device is registered, the necessary MQTT[3] topics are created; using those topics the devices are able to send data to the IoT Agent.

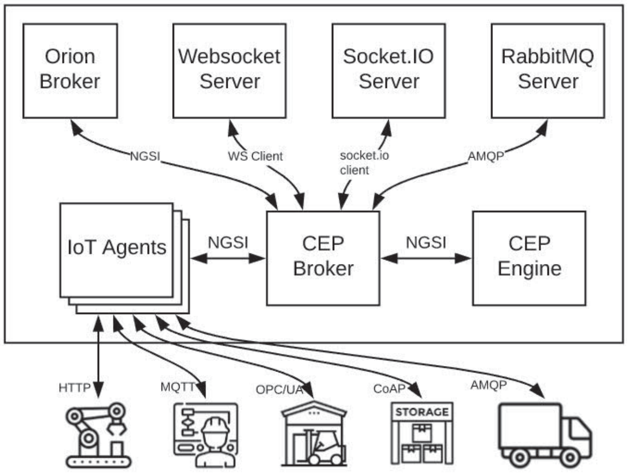

Figure 2. FIWARE IoT Component

To provide the data handling functionalities, the FIWARE IoT Hub offers a complex event-processing tool based on Esper[4]. This data management module is decomposed in two modules, a communication broker and the CEP engine itself. Consequently, the CEP can be isolated from the rest of the modules and so, from other irrelevant tasks. Both broker and CEP are communicated following the publish-subscribe pattern. Using FIWARE[5] based technologies assure the correct interoperability among the different modules of the FIWARE IoT Hub, as the components all the components involved use the NGSI API defined by OMA[6]. On top of the component, a different set of connectors are offered. These connectors support a wide range of technologies as FIWARE, socket.io[7], WebSockets[8] or rabbitmq[9]. The aim of these connectors is to feed with information the data consumers for further processing, as an example C2NET platform, with the correct data and with the expected format.

B. Integration of the IoT hub with the C2NET platform

To facilitate a bidirectional communication between C2NET platform and the FIWARE IoT Hub, it was followed the client/server communication pattern, implemented using Socket.IO. This event-driven technology enables real-time, bidirectional communication between a client and a server, where both components have a nearly identical API.

Socket.IO primarily uses the WebSocket protocol with polling as a fallback option, while providing the same interface. Although it can be used as a wrapper for WebSocket, it provides many more features, including broadcasting to multiple sockets, storing data associated with each client, and asynchronous I/O. Due to its nature, it supports the ability to implement real-time analytics, binary streaming and messaging among other functionalities. It is a custom real-time transport protocol implemented on top of other real-time protocols like WebSocket. It handles the connection transparently, automatically upgrading to WebSocket if possible. Consequently, this technology perfectly matches the main interoperability requirements between the FIWARE IoT Hub and C2NET platform, assuring the correct execution of the processes where both actors participate: firstly, the configuration of the FIWARE IoT Hub and secondly, the feeding of C2NET platform with data coming from IoT devices.

C. Integration with factory shop floor devices

IoT Gateway is developed to bridge the gap between the devices on the factory floor and the FIWARE IoT Hub. The gateway exploits the physical devices on the shop floor to publish the relevant data to the FIWARE IoT Hub. The gateway uses the publish-subscribe pattern to exchange the desired data, thus supporting both one-to-one as well as one-to-many communication. Therefore, it is possible to broadcast data from the connected devices to multiple subscribers against a specific topic, by making use of the MQTT[10] protocol. Moreover, the gateway is developed based on the REST[11] architecture. The gateway exposes specific REST endpoints, which the IoT hub can use to register, read, modify and un-register the devices and/or relevant events available against the device. In this context, Figure 3 presents sequence diagram of the steps involved for registering specific events against the desired device.

The FIWARE IoT hub sends a JSON configuration to the gateway, which consists of the *deviceName*, *deviceId*, *deviceURL* that provides the endpoint, events that lists each concerned event and *mqttServer* to provide the endpoint of the server, and in this case, it is the FIWARE IoT Hub.

Once the IoT gateway receives the necessary data, it creates an identity for each device and thereafter, it keeps a track of the metadata of that device, e.g., its capabilities and attributes. Additionally, the gateway supports metadata with a description about the device capabilities. As an example, which events does the device reports and what are the respective units of each event. Moreover, the gateway keeps a track of the latest state of the device (whether it has connected or disconnected) and reports the FIWARE IoT Hub of the respective status. Once a device has been registered, the gateway subscribes against the requested events for the device and listens to the data change on them. As soon as the events data changes for the device, it gets informed to the gateway,

---

[3] http://mqtt.org/

[4] http://www.espertech.com/

[5] https://www.fiware.org/

[6] http://www.openmobilealliance.org/

[7] https://socket.io/

[8] https://en.wikipedia.org/wiki/WebSocket

[9] https://www.rabbitmq.com/

[10] https://www.iso.org/standard/69466.html

[11] http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm

which then forwards the data to the desired topic in MQTT server.

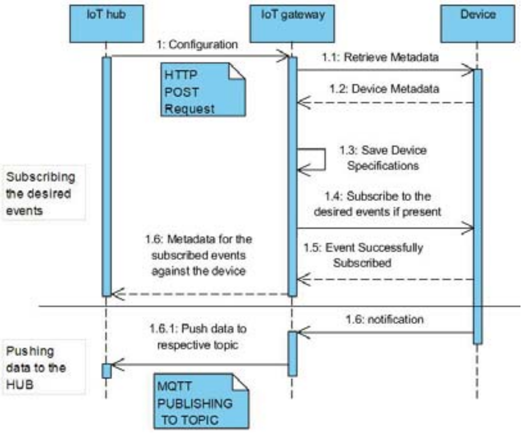

Figure 3. Registering specific events against a specific device diagram

IV. AN INDUSTRIAL SCENARIO FOR PROOVING THE CONCEPT: THE FASTORY LINE

As any research work, this approach has been tested and validated via a use case. In this section, the approach which has been presented in the previous section is validated in an educational and research use case. This section presents the use case, the implementation and configuration that took place in this validation.

A. The FASTory line

The Factory Automation Systems and Technologies Laboratory, known as FAST Lab, established in 2006 and located in Tampere University of Technology. As a research and educational facility, FAST Lab owns an assembly line, which was used for assembling mobile phones as depicted in Figure 4. The line was transformed from assembling real mobile phones to draw the phones that mimics the original purpose. With three models and three colours and three main parts (frame, screen and keyboard), the line is capable of producing 729 different products. Additionally, the FASTory line was retrofitted with web services enabled devices and to permit the usage of web services for invoking operations, receiving notifications and collect data form the line.

The FASTory line contains 10 Workstations that are equipped with two conveyors and one SCARA robot for drawing purposes, one workstation for adding new paper or removing the final product form the pallets and finally, a manual workstation for pelletizing the assembly line with a custom-made pallets the carries a paper as also depicted in Figure 4.

This research assembly line has been intensively used in research projects such as C2NET, eScop[12], ASTUTE[13], eSONIA[14]. Besides, this line has been used for educational purposes were student can interact with a real equipment that might face them in the future. Regarding this research work, the web services enabled devices has been exploited as IoT devices that are uploads the shopfloor data via the FIWARE IoT Hub that is presented in the previous section.

Figure 4. FASTory Assembly Line and the final product on a pallet

B. The FIWARE IoT Hub deployment

There are two main processes in which the FIWARE IoT Hub participates: i) the configuration of the FIWARE IoT Hub itself including the devices to be used, and ii) the feeding of C2NET platform with the correct data coming from the IoT devices. Both processes are presented in Figure 5.

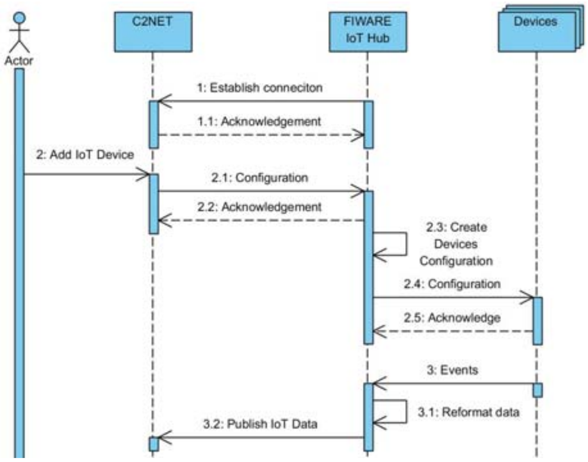

Figure 5. FIWARE IoT Processes

As a precondition, the user has to inform all the necessary properties to configure the FIWARE IoT Hub and Devices using the C2NET User Collaboration Portal (UCP). Once the user has registered the Hub in the platform, the process is as follows:

1. FIWARE IoT Hub tries to establish a connection providing a SSL client certificate to be identified. If the certificate is valid, the connection is authorised and established.
2. Using the UCP, the user can launch the configuration process. At this point, the FIWARE IoT Hub receives the configuration message to configure itself. The configuration message is composed by two parts: one specific for the FIWARE IoT Hub and other for the specific configuration of the Devices.
3. If the configuration of the FIWARE IoT Hub has been successfully executed, it will send the Devices their specific configuration. Among others, in this configuration it is included the specific MQTT topics to be used by the devices.

---

[12] http://www.tut.fi/escop/
[13] http://www.tut.fi/astute/www.astute-project.eu/index.html
[14] http://www.esonia.eu/

4. Using the MQTT topics created, the Devices will send their measurements and received at the FIWARE IoT Hub. Applying defined data handling techniques on the data, the FIWARE IoT Hub will redirect it to C2NET platform for further processing.

V. Results and Discussion

Once that the approach for supporting a cloud platform in order to collect and process streams of factory shop floor has been, first, introduced and, second, demonstrated within a use case, this section presents a set of results and a discussion derived from the performed research work. In fact, as the authors of this work belong to different type of organisations i.e., university and organisation research and innovation centre the achieved work has been positively evaluated from different viewpoints.

```
[{
    "name": "deviceWorkstation7",
    "type": "JSON",
    "value": {
        "device": {
            "id": "D7",
            "url": "http://192.168.7.2"
        },
        "event": [{
                "id": "Z2_Changed"
            }, {
                "id": "Z3_Changed"
            }
        ],
        "mqtt": {

            "mqttServer": "mqtt://130.206.116.69:1883",
            "topic": {
                "topic_command_req": "/C2NET/command/request",
                "topic_mult_attrs": "/C2NET/workstation7/attributes",
                "topic_command_resp":
                "/C2NET/workstation7/command/response",
                "topic_single_attr": "/C2NET/ws7/attributes/+"
            },
            "nsgiId": "nsgi_D1",
            "attribute": {
                "name": "AP1",
                "type": "double"
            },
            "valuePath": "notificationBody.payload.PalletID"
        }
    }
}]
```

Figure 6. JSON configuration

A. *Monitoring the factory shop floor data from the cloud*

The interaction of the IoT Gateway with the IoT Hub will be explained in this section. The FIWARE IoT Hub is responsible for sending a JSON configuration to the IoT Gateway, which is used by the gateway to configure the devices and corresponding events on its side. Figure 6 shows the configuration that was sent to the gateway. The used configuration shows that the Hub is interested in monitoring Zone 2 and Zone 3 of the Work Station 7 on the FASTory line. Besides, the configuration received from the FIWARE IoT Hub contains the following information: i) The device URL on the factory floor ii) Array of events to be subscribed against the device iii) MQTT server configuration for connectivity iv) Topic to send the respective event data.

As a consequence, the JSON response is sent against the received configuration, which is shown in Figure 7. As mentioned earlier, the IoT gateway has been created based on the REST architecture and therefore, the respective http codes are sent as part of the http response. For example, the response configuration signifies that a device has been successfully added and so on the requested events as well. After, the device and the corresponding events have been registered, the IoT gateway starts listening to the events changes on the factory floor. Since we are interested in monitoring whether a pallet has arrived or left a particular zone, therefore, we keep a track through IDs of those pallets. If a pallet has left a specific zone, this event will be notified by an ID containing a value of -1. However, this specific ID will notify an incoming pallet at a specific zone. For the configuration sent by the FIWARE IoT Hub, any pallet leaving Zone 2 will be notified by sending -1 to the following topic *C2NET/ws7/attributes/Z2_Changed.*

```
{
    "D7": {
        "code": 201,
        "info": {
            "deviceId": "D7",
            "links": {
                "self": "http://192.168.1.111:3000/devices/D7",
                "device": "http://192.168.7.2",
                "events": "http://192.168.7.2/rest/events"
            },
            "connection": "connecting..."
        },
        "events": {
            "Z2_Changed": {
                "code": 201,
                "info": {
                    "eventID": "Z2_Changed",
                    "links": {
                        "self": "http://192.168.1.111:3000/devices/D7",
                        "device": "http://192.168.7.2",
                        "events": "http://192.168.7.2/rest/events"
                    },
                    "topic": "C2NET/ws7/attributes/Z2_Changed"
                }
            },
            "Z3_Changed": {
                "code": 201,
                "info": {
                    "eventID": "Z3_Changed",
                    "links": {
                        "self": "http://192.168.1.111:3000/devices/D7",
                        "device": "http://192.168.7.2",
                        "events": "http://192.168.7.2/rest/events"
                    },
                    "topic": "C2NET/ws7/attributes/Z3_Changed"
                }
            }
        }
    }
}
```

Figure 7. JSON response

Similarly, an arriving pallet at Zone 2 will be notified by sending the Pallet ID to the mentioned topic in the MQTT server. As soon as the data is sent to the FIWARE IoT Hub, it verifies the data and thereafter, publish it to the C2NET platform. This configuration mentioned in the example is only for two zones for workstation 7 on the FASTory line. In order to track the pallet movement on the assembly line, every zone against each workstation must be subscribed. Thereafter, every change occurring on the line would be easily monitored and published to the cloud platform.

B. *Discussion*

The FIWARE IoT Hub is a software solution that aims at supporting the digitisation of manufacturing companies, focused on the integration of new IoT devices. Based on technologies like Nodejs and Java, it is used to help companies on the process of gathering data from shop floor devices and to move it to more complex computing environments, like cloud computing. Thanks to those new cloud environments, terms like predictive maintenance, supply chain optimisation or advance visualisation are becoming more and more often. This middleware component aims at breaking interoperability barriers, being a link between data producers and consumers, opening isolated shop floor data to a wide range of applications and services, obtaining a clear added-value on the huge amount of data generated by the resources.

Main benefits reside in the simplicity of use of the component, as well as that it is capable to run on low power devices systems. This clearly facilitates the access to this technology to manufacturing companies who do not have the capacity or do not want to expend much economical efforts in terms of IT infrastructures, such as SMEs, who usually have limitations in this aspect. Finally, it is important to highlight the real-time related manner in this research such latency which might be raised in such research. In this context, this research focused on the concept acceptability where the performance needs to be studied further in next related research work.

VI. CONCLUSION

As discussed in the manuscript, the new changes to be adopted by industries seeking for the implementation of the Industry 4.0 vision require an investment on ICT-based solutions that permit the collection, processing and analysis of data. In fact, there is a clear challenge not only in the handling of huge volumes of data but on the understanding and integration of a vast variety of formats that data may be presented. This is caused by the heterogeneity of systems that are connected under the same platform.

In the scope of Industry 4.0, industrial IoT emerged as the alternative to interconnect different types of devices that may both provide and consume information from industrial systems remotely and, ideally, without spending much time on derived tasks, such as configuration. In this context, this manuscript presents solution that is demonstrated though a specific industrial-based scenario. Furthermore, the solution has been deployed in a cloud-based solution that has been validated by the European Commission throughout the C2NET project. For the future work, this research can be extended to cover more protocols and standards that are used in the IIoT devices. Additionally, this research can be extended to address the performance issues in the presented approach.

ACKNOWLEDGEMENT

The research leading to these results has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement n° 636909, correspondent to the project shortly entitled C2NET, Cloud Collaborative Manufacturing Networks.

REFERENCES

[1] L. N. Hyseni and A. Ibrahimi, "Comparison of the cloud computing platforms provided by Amazon and Google," in *2017 Computing Conference*, 2017, pp. 236–243.

[2] N. Govindarajan, B. R. Ferrer, X. Xu, A. Nieto, and J. L. M. Lastra, "An approach for integrating legacy systems in the manufacturing industry," in *2016 IEEE 14th International Conference on Industrial Informatics (INDIN)*, 2016, pp. 683–688.

[3] W. M. Mohammed, B. R. Ferrer, J. L. Martinez, R. Sanchis, B. Andres, and C. Agostinho, "A multi-agent approach for processing industrial enterprise data," in *2017 International Conference on Engineering, Technology and Innovation (ICE/ITMC)*, 2017, pp. 1209–1215.

[4] W. M. Mohammed, B. R. Ferrer, L. Jose, and M. Lastra, "Configuring and visualizing the data resources in a cloud-based data collection framework," in *2017 International Conference on Engineering, Technology and Innovation (ICE/ITMC)*, 2017, pp. 1201–1208.

[5] M. P. Papazoglou and A. Elgammal, "The manufacturing blueprint environment: Bringing intelligence into manufacturing," in *2017 International Conference on Engineering, Technology and Innovation (ICE/ITMC)*, 2017, pp. 750–759.

[6] S. Iarovyi, W. M. Mohammed, A. Lobov, B. R. Ferrer, and J. L. M. Lastra, "Cyber-Physical Systems for Open-Knowledge-Driven Manufacturing Execution Systems," *Proc. IEEE*, vol. 104, no. 5, pp. 1142–1154, May 2016.

[7] B. R. Ferrer and J. L. M. Lastra, "An architecture for implementing private local automation clouds built by CPS," in *IECON 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society*, 2017, pp. 5406–5413.

[8] H. R. Faragardi, H. Fotouhi, T. Nolte, and R. Rahmani, "A Cost Efficient Design of a Multi-sink Multi-controller WSN in a Smart Factory," in *2017 IEEE 19th International Conference on High Performance Computing and Communications; IEEE 15th International Conference on Smart City; IEEE 3rd International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, 2017, pp. 594–602.

[9] H. Jayathilaka, C. Krintz, and R. M. Wolski, "Detecting Performance Anomalies in Cloud Platform Applications," *IEEE Trans. Cloud Comput.*, vol. PP, no. 99, pp. 1–1, 2018.

[10] G. Zaayman and A. Innamorato, "The application of simio scheduling in industry 4.0," in *2017 Winter Simulation Conference (WSC)*, 2017, pp. 4425–4434.

[11] M. M. Hassan and E.-N. Huh, *Dynamic Cloud Collaboration Platform*. New York, NY: Springer New York, 2013.

[12] W. M. Mohammed *et al.*, "Generic platform for manufacturing execution system functions in knowledge-driven manufacturing systems," *Int. J. Comput. Integr. Manuf.*, vol. 0, no. 0, pp. 1–13, Nov. 2017.

[13] B. Andres, R. Sanchis, and R. Poler, "A Cloud Platform to support Collaboration in Supply Networks," *Int. J. Prod. Manag. Eng.*, vol. 4, no. 1, pp. 5–13, Jan. 2016.

[14] A. Katasonov *et al.*, "An approach to production scheduling optimization a case of an oil lubrication and hydraulic systems manufacturer," in *2017 International Conference on Engineering, Technology and Innovation (ICE/ITMC)*, 2017, pp. 1123–1130.

[15] "FIPS 197, Advanced Encryption Standard (AES)," p. 51.

[16] K. Suri, J. Cadavid, M. Alferez, S. Dhouib, and S. Tucci-Piergiovanni, "Modeling business motivation and underlying processes for RAMI 4.0-aligned cyber-physical production systems," in *2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2017, pp. 1–6.

[17] V. C. M. Leung and M. Chen, Eds., *Cloud Computing*, vol. 133. Cham: Springer International Publishing, 2014.

[18] "The 'Only' Coke Machine on the Internet." [Online]. Available: https://www.cs.cmu.edu/~coke/history_long.txt. [Accessed: 03-Apr-2018].

[19] P. Mohan, V. N. Padmanabhan, and R. Ramjee, "Nericell: rich monitoring of road and traffic conditions using mobile smartphones," 2008, p. 323.

[20] B. R. Ferrer, W. M. Mohammed, E. Chen, and J. L. M. Lastra, "Connecting web-based IoT devices to a cloud-based manufacturing platform," in *IECON 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society*, 2017, pp. 8628–8633.

[21] J. Lee, B. Bagheri, and H.-A. Kao, "A Cyber-Physical Systems architecture for Industry 4.0-based manufacturing systems," *Manuf. Lett.*, vol. 3, pp. 18–23, Jan. 2015.