

Conference Paper

When IoT Meets DevOps: Fostering Business Opportunities

Logica Banica¹, Persefoni Polychronidou², Magdalena Radulescu¹, and Cristian Stefan³

¹Faculty of Economics and Law, University of Pitesti, Targu din Vale Street, no. 1, 110040 Pitesti, Romania

²Department of Accounting and Finance, Central Macedonia Institute of Technology, Termia Magnisias, 62124, Serres, Greece

³Faculty of Engineering, Informatics and Geography, Spiru Haret University, 13 Ion Ghica Street, District 3, Bucharest, Romania

Abstract

The Internet of Things (IoT) is the new digital revolution for the near-future society, the second after the creation of the Internet itself. The software industry is converging towards the large-scale deployment of IoT devices and services, and there's broad support from the business environment for this engineering vision. The Development and Operations (DevOps) project management methodology, with continuous delivery and integration, is the preferred approach for achieving and deploying applications to all levels of the IoT architecture. In this paper we also discuss the promising trend of associating devices with microservices, which are further encapsulated into functional packages called containers. Docker is considered the market leader in container-based service delivery, though other important software companies are promoting this concept as part of the technology solution for their IoT customers. In the experimental section we propose a three-layer IoT model, business-oriented, and distributed over multiple cloud environments, comprising the Physical, Fog/Edge, and Application layers.

Keywords: Internet-of-Things, software technologies, project management, business environment Headings

Corresponding Author:

Logica Banica
 olga.banica@upit.ro

Received: 14 October 2018

Accepted: 1 November 2018

Published: 26 November 2018

Publishing services provided by
 Knowledge E

© Logica Banica et al. This article is distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use and redistribution provided that the original author and source are credited.

Selection and Peer-review under the responsibility of the EBEEC 2018 Conference Committee.

1. Introduction

Over the last few years, Internet of Things (IoT) has emerged as a cutting-edge technology aiming to solve the problems of a growing number of devices connected to the Internet and their requirements for simple services like authentication, data transmission/reception, communication with other devices, and so on.

 OPEN ACCESS

These services can be implemented as single-function software applications (microservices), which are scalable, relatively independent though they can collaborate, and can be deployed in the Cloud. These software components are subject to permanent updates and require a fast and continuous delivery and integration, thus they can be developed as small and medium-sized projects, using *DevOps* (Development and Operations) project management methodology.

Frequent and concurrent access to microservices slows down the performance of Cloud-based applications, which led to the idea of deploying them in another Cloud intermediate structure, called Fog/Edge computing.

In this context, microservices are placed at this level, accessible through the *Application Programming Interfaces (APIs)*, and are grouped into functional packages, called *Containers*.

In the following section, Literature Review, we focus on several of the current IoT approaches, technologies and models. The Methodology section will detail the above-mentioned concepts, focusing on the functions and ways of cooperating and integrating them into a three-layer IoT architecture. In the experimental part, we define a model for the business environment and we propose software implementations launched by IT companies, which can be used for deployment. The Conclusions section contains the final remarks regarding the significance of this new world of interconnected “things” and about the development pace, which will be at least as fast as the global Internet coverage.

2. Literature Review

The IoT research field is one which has not yet been clearly delimited, as it has many directions of impact (technology, living environment, social acceptance), and is in its early infancy, with rapid evolving trends and surprising day-to-day applications. The future will bring to attention many innovative solutions in this area.

The concept involves a wide range of possibilities in the business environment and means of improving the client usage experience. Smart cities, smart universities, smart devices, smart grids and energy management are several of the potential applications and services of the IoT. We don't currently have a generally accepted definition for the IoT concept, but there are a few available definitions of IoT as follows.

The Internet-of-Things has been defined in Recommendation ITU-T Y.2060 as a “global infrastructure for the information society, enabling advanced services by inter-connecting (physical and virtual) things based on existing and evolving interoperable information and communication technologies” [1].

In the special report Internet of Things, 2014, IEEE referred to the Internet of Things as: “A network of items—each embedded with sensors—which are connected to the Internet” [2].

IoT is expected to adopt major improvements especially at the core level in the near future, being able to manage more end-points such as mobile, sensors, actuators etc. that will require an increased level of integration between “things” and applications.

As we have shown in a previous paper, microservices are essential to the development of IoT because they allow the development of modular applications, easy to modify and test, independently deployable and loosely-coupled services, intended to perform a single function [3]. A tightly-coupled architecture, having too many dependencies, would require more time and technological efforts to implement and change applications, in comparison with the microservice-based architecture that should be independent and not be constrained to deliver updates in cascade [4].

These microservices can be built using the support of DevOps methodology, which is characterized by continuous deployment, integration, testing, and delivery of the product [5].

Both the microservice architecture and the DevOps project management methodology are accepted as philosophies for software development, and their implementation is not related to specific tools or languages. Each microservice is targeted at a distinct, specific function, and it is developed, tested, and deployed independently, having autonomy from other services, but communicating with them as processes through API gateways [6].

Microservices work better in containers that encapsulate a running environment, playing the role of host for development, testing and deployment in the same time, and offering scalability and portability. Containers running on the same host are isolated from each other and changing a service will not influence other services, given that the APIs are kept the same [6].

Although containers are not a new technology and have been used before by Unix and Linux, currently Docker is the best known open container management framework which offers all the features necessary for development and deployment of microservices. Docker containers are usually transient and disposable, but they are suitable for long-running operations, such as web servers or application servers.

Rafael Benevides from RedHat company points out the benefits of using containers [7]:

1. They run processes in any environment because they encapsulate the application with all its necessary components (operating system, libraries, configurations and folders);
2. They are lightweight since they allocate the memory only for the main process.
3. They have a fast start up (as fast as a typical Linux process).

According to Petacchi, the use of microservices architectures with containers has also drawbacks to consider [8]:

1. Software creators are required to build a complex distributed system;
2. Testing process becomes more complicated;
3. Application architects must develop the communication between services;
4. The lack of distributed transactions makes the service management difficult;
5. The coordination between team members must be tight, as multiple services form a challenging working environment.

This paper is based on the idea that IoT can empower the business environment and introduces a hierarchical model that combines a series of modern technologies, free and open software that can be leveraged by enterprises with minimum costs and optimum efficiency.

3. Methodology

Research methodology should define the problem statement and answer a few questions which will lead to a viable IoT model. Many major software producers have already implemented this concept into their commercial offers for different industries, though the question is whether the business environment is ready for mass IoT adoption. In this regard, the implementation level depends on several factors, such as:

1. Managers do not have an IT development strategy or enough knowledge about the Internet-of-Things;
2. When the top management supports the deployment of this concept, the lack of financial support issue arises,

3. Is there a way to fulfill the requirements of small and medium-sized enterprises (SMEs), or even big companies, with free software solutions?

Our paper aims to find answers to several research questions:

1. which technologies can be used for supporting IoT?
2. what are the benefits of IoT implementation?
3. which are the steps that a company should go through to in order to join the IoT world?

Therefore, the objectives of this study are:

1. Highlighting the advantages of microservice-based IoT implementations, and their development with DevOps methodology, for the business environment;
2. Defining the above-mentioned concepts and identifying the existing technologies corresponding to their implementation,
3. Offering some examples of successful software products available on the IT market in this area.

3.1. Microservices and containers in Fog/Edge computing

Microservices can be developed and best deployed in container-defined operating environments, as they offer hardware memory, storage, connectivity and security capabilities and portability across different environments. Also, containers allow the application of DevOps methodology for software development.

Because containers include all packages required to make available a service requested by smart devices, it is recommended to use them for any IoT device. The fact that a container is isolated from the OS and from other software applications offers increased security, including to DDoS (distributed denial-of-service) attacks on the infrastructure.

Unlike hypervisor virtualization, which allows for many operating systems running simultaneously on a hardware platform, containers share the same operating system and create an environment for running the application, isolated from other applications.

According to RedHat organization, a Linux container is “a set of processes that are isolated from the rest of the system, running from a distinct image that provides all files necessary to support the processes” [9].

The platform on which Linux containers (LXC) are deployed can be a server, located in the Cloud or in the Fog/Edge layer, while maintaining the structure and operating conditions.

There are two ways to implement Linux containers:

1. the original Linux container technology – LXC, offering an OS-level virtualization technology that allows multiple isolated virtual environments on a single control host;
2. Docker system, an extension and an improvement of LXC's capabilities, acting as a container engine, packaging the application and its dependencies in a virtual container running on a Linux server; Docker is recommended for microservices-based architectures.

Docker is a container system oriented for fast deployment of application environments. In order to leverage its power, one has to install it first – it comes already packaged for major Linux distributions. After installation (the command is specific to each distribution), the service must be started manually before use.

At the core of the Docker design philosophy there are two main concepts: the Containers and the Images. The image is built by manually defining the properties/commands required for the current running environment, including the download and configuration of specific apps. The next step is to actually build the container, and then generate the running script. Docker will execute this code and expose the application to the outside world on the indicated port. Containers can be started or stopped independently at any given time [10, 11].

Docker engine can be a viable platform for Fog computing, an intermediate level of infrastructure that enables decentralizing Cloud computing services and reduces the response time by moving processing power closer to the "things".

Figure 1 shows the Fog/Edge layer of an IoT model, based on microservices running in Docker containers, and on the paradigm of solution decentralization:

The Fog/Edge layer is the first line of processing data generated by the devices and is based on containerized applications (microservices) that are supported by Docker.

The Fog/Edge layer does not intend to be a replacement for the Cloud, as the requirements of the market involve also the processing of large quantities of data that calls for powerful Big Data platforms [12].

As the need for a common interface that distributed elements can access became obvious, the fog/edge layer was implemented in order to support their lifecycle. This way, devices communicate directly with the microservices that are deployed closer to

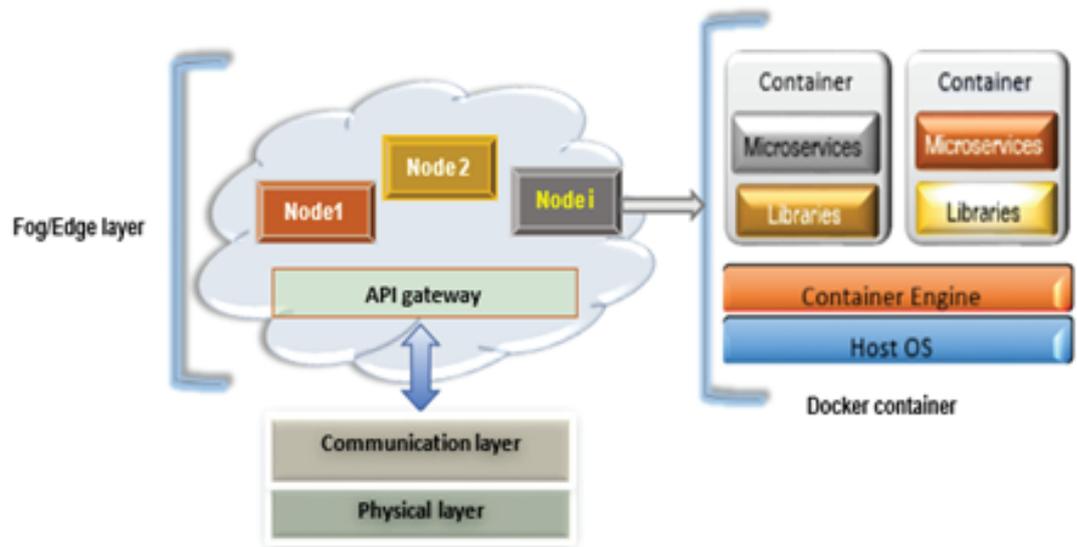


Figure 1: Including Docker containers in IoT model at Fog/Edge layer.

them, and use API requests via standard HTTP channels, the advanced features like SSL communication, authentication and load balancing can be easily supported [13].

3.2. Applying DevOps for IoT development solutions

Developing an IoT project for a business involves modularity and focus on granular software solutions that implement device-required functions, integrated into a microservice-based architecture [5].

These software modules can be implemented in any language (C ++, Java, Python etc.), though they require quick and easy updating, testing, deployment and monitoring that do not impact on the activity of other components, leading to the idea of Agile design / redesign by using the DevOps paradigm.

According to Lewis and Fowler, *microservices* represent the method for “designing software applications as suites of independently deployable services” [14].

The development of microservices by applying DevOps methodology ensures the continuous integration and the rapid delivery of “things” functionalities whenever it is necessary.

There are several factors specific to testing and deployment of microservices using DevOps:

1. IoT data impact in the management solution;
2. operating conditions complexity;

3. functionality of the new and older devices of the platform;
4. ensuring the latest updates for all devices.

The key element of the testing process is the simulation of the device behaviour and creating the virtual environment that can be used for software testing. Simulation refers to interfaces, API accessibility and creating training and test datasets [15].

Internet of Things deployments, based on microservices, containers and DevOps are currently some of the most popular topics in the IT world.

The division of IoT applications into tens and hundreds of microservices that are communicating with each other and the need to update devices according to technological evolutions of the market lead to a great importance level offered to automate the process of building / rebuilding, testing and deploying software by using DevOps.

Adopting continuous integration and delivery for microservices development cycle (the DevOps features) increases implementation speed and brings these changes in production as quickly as possible. Also, by working in DevOps, design teams can add new microservices without affecting the entire application [16].

As a continuous integration tool, it is recommended to use Jenkins, an open source application written in Java, for testing and troubleshooting when using a larger, real-time code base. Jenkins is an automation server that can use pre-defined scenarios for building, testing and deploying software applications. It can be installed via the native OS packages, Docker containers or can be run independently on any machine that has a Java Runtime Environment installed. Jenkins can be run as a standalone application or as a standard web application from a Java servlet container such as Apache Tomcat or JBoss.

Jenkins applications can be downloaded as Docker images, each of them being deployed in Docker as a container. A Docker image for any given application (like Jenkins application), can be deployed on any supported operating system (MacOS, Linux and Windows) or cloud service (AWS and Azure) which is also running Docker [17].

4. Managing an IoT Application for Business

According to a Gartner report from 2015 [18], there will be 21 billion connected IoT devices by 2020, but the effect of this increase will be seen in the transformation of the digital market, if the business environment will know how to manage this technological challenge.

A company that intends to deploy and integrate IoT into its operations, needs to tackle two important issues:

1. The internal communication between enterprise devices across the entire supply chain;
2. The access of customers and collaborators to devices, in accordance with the company's connectivity and security policies.

The dynamics and the interest for IoT are present especially in the distribution and retail divisions, supported by production, warehousing and transport sectors within the same globalized company [19].

From the perspective of a company that carries out production, warehousing, transportation and retail activities, we consider that the suitable architecture implements the specific functions required by devices at the Fog / Edge level, and performs the analytical, forecasting and simulation processes in the Cloud.

4.1. An IoT model in a retail and distribution company

Figure 2 presents the three-level architecture proposed by the authors for developing and implementing an IoT model in a retail and distribution company that involves a multi-stores and multi-warehouses structure.

1. At the first layer – Physical and Communication – are the physical objects connected in company stores, warehouses, transportation, that solve requests for connection/disconnection, authentication, sending/receiving information to/from the next level; the connectivity refers to the communicating with and between devices, reliable delivery of information across the network, implementation and translation of various protocols, security at the network level;
2. The nodes based on Docker container engine are placed at the Fog/Edge level, providing the functions attached to “things” from the entire company network by using microservices;
3. The third level - Cloud-based IoT platform - includes IoT's Big Data intelligent processing, consisting of analysis, filtering, and forecasting operations with business intelligence tools that lead to new information which underpin future decisions and management strategies.

Detailing the Fog/Edge level, we mention the several main directions for tracking IoT devices:

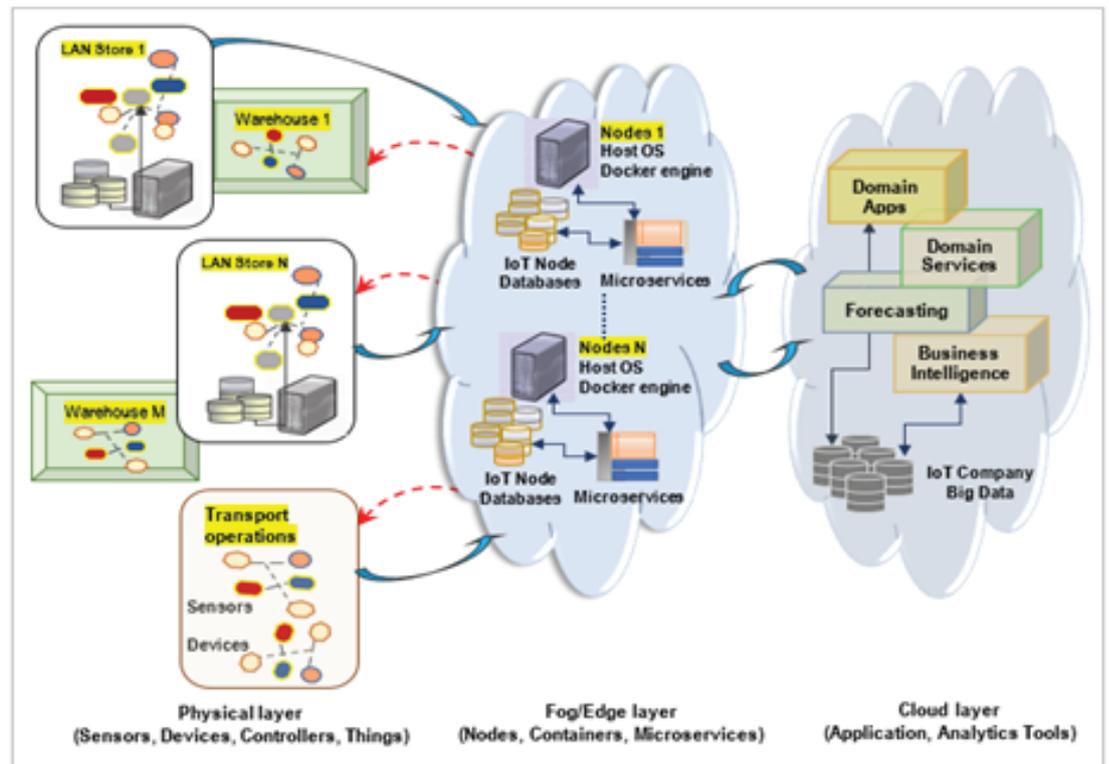


Figure 2: The IoT architecture for a retail and distribution company

1. Within the Company's smart stores, the entire shopping trips of customers will be analyzed and stored, so the merchants understand their future behavior. In addition, successive monitoring of the customer traffic will create the digital marketing based inside or outside the stores.
2. Sensors and devices placed at warehouse level allow full-time monitoring of product stocks in real-time, and implicitly collaborate with stores to meet customers' shopping requests and online sales opportunities. The next level will be to directly check the stocks by the mobile devices of customers.

Transporting goods from delivery locations to warehouses and shops in an efficient way is one of the goals of intelligent transport applications in retail and distribution. IoT will optimize the transport by tracking the route in advance and avoiding possible impediments (weather, accidents, maintenance).

5. Results and Discussions

Big software companies offer a whole range of IoT solutions on the IT market, but we consider that the business environment is not yet ready to implement such platforms [19].

In order to evaluate the benefits of this type of application, a suitable alternative is the use of simulation software tools [20] such as:

1. Azure IoT Device Simulation is a virtual environment aimed at testing the IoT elements and communication ways in complex scenarios, by avoiding the costly investment in dedicated hardware for rapid prototyping [21];
2. IoTify allows the users to deploy large installations of IoT, thus having lots of devices simulated with all their traffic, and this empowers the users to efficiently debug the projects before the project hits the market [22];
3. NetSim is another way to create various testing patterns, by defining large pools of nodes that interact in a convincing way over the virtual network [23];
4. IBM's Bluemix offers the IoT enthusiasts a sandbox to test their ideas and manage the infrastructure via a web-based interface and offers developers useful insight on how the apps behave in different use cases. There are functions for data processing and storage, and even social sites interaction [24];
5. iFogSim from CloudBus is aimed at the Fog layer devices behaviour simulation. These are the host for self-contained apps that concentrate data from child nodes and process it briefly [25].

Concerning the interest of Romanian and Greek companies in IoT, our studies revealed an increase of participation in European research projects on the Internet of Things and the development of partnerships and networking between the academic and business community through the development of IoT models.

Greece participates in 4 of the 7-approved research and development projects at pan-European level, drawing a total of € 3.36 million of Community co-financing, while total Community co-funding for research and development is € 51.5 million [26]. In the past years, Greek companies have made some digital investments in their software and hardware stock but have not yet extensively adopted leading technologies like the Internet of Things, cloud computing and big data analytics [27].

In Romania an important project called "Ecosystem for Research, Innovation and Development of IoT Based Products and Services to Sustain an Interconnected Society", started in 2017, project co-financed by the European Fond for Regional Development in the framework of the Operational Programme for Competitiveness 2014-2020, having the University Politehnica of Bucharest as a main contractor and subcontractors representing the business community interested in the development of innovative

products and services in the IoT sphere and intelligent cities [28]. The total project value is 15.901.500 lei (approximately € 3.4 million).

Another initiative that will foster the development of the IoT-centric society is the European Commission's project called Wifi4EU, which will offer more than 120 million euros for deployment of free WiFi hotspots in public places. These are intended to support future IT services like smart parking, inter-vehicle communication, smart grids and public utility management. This project is currently in the voucher-request phase, which will end on May 15, 2018 [29].

6. Conclusions

This paper aims to answer some questions related to the implementation of IoT in the business environment.

We have mentioned in the previous section how IoT devices placed at physical layer are to be integrated, the fact that they exchange information and take intelligent decisions through simple functions, implemented as microservices. We also demonstrated the need to insert a Fog/Edge layer, consisting of nodes placed closer to the "things" and accessed through API calls. In these nodes, we recommend installing the Docker container engine, in order to isolate the microservices running in containers from each other and to include all required components that allow them to function independently from the operating system.

Frequent service upgrades and the need for continuous development and integration have led to the utilization of the DevOps methodology for software applications, from physical level and evolving toward gateways and Fog devices as a unique method of developing and managing IoT systems.

Our proposal involves a 3-tier IoT model that integrates and creates a close relationship between all these new technologies in a retail and distribution company.

Even if not all managers are ready to step up to the Internet of Things, it is beneficial to their businesses and this can be proven with IoT simulators made available by software companies, some of them being free software tools.

By analyzing the state of IoT implementation in Greece and Romania, we noticed that the academic environment initiated collaboration with the business world through structural funds projects (financed by the EU), aiming to prove the viability and the advantages of this opportunity. This paper described briefly the emerging technologies of IoT, proposed the generic structure for an IoT model, and described possible future simulation applications and some projects in the domain. Our future work will focus on

a comparative analysis of several simulation platforms; will select such a lab application and run tests and evaluate metrics to establish the advantages or drawbacks of its implementation in the business environment.

References

- [1] The official website of International Telecommunication Union (ITU), 2012. Recommendation ITU-T Y.2060: Overview of the Internet of things, <http://www.itu.int/en/ITU-T/gsi/iot/Pages/default.aspx>.
- [2] The official website of IEEE Institute, 2014. Special Report: The Internet of Things, <http://theinstitute.ieee.org/static/special-report-the-internet-of-things>
- [3] Banica, L., Stefan, C., Hagi, A. (2017). Leveraging the Microservice Architecture for Next-Generation IoT Applications, *Scientific Bulletin – Economic Sciences*, Volume 16/ Issue 2, pp. 26-32.
- [4] Babaria, U. (2018). Internet of Things - Why IoT Development Needs Microservices and Containerization, <http://sandhill.com/article/why-iot-development-needs-microservices-and-containerization/>
- [5] Kappelman, E. (2017). Why microservices and DevOps are a match made in heaven, <https://www.packtpub.com/books/content/why-microservices-and-devops-are-match-made-heaven>.
- [6] Ma, J. (2016). Building Modern Applications with Microservices: Part 1, <https://www.mongodb.com/blog/post/building-modern-applications-with-microservices-part-1>
- [7] Benevides, R. (2016). 10 things to avoid in docker containers, <https://developers.redhat.com/blog/2016/02/24/10-things-to-avoid-in-docker-containers/>
- [8] Petacchi, D. (2017). Microservices and containers: the new architecture paradigm, <http://www.sdggroup.com/en/find-insights/blog/microservices-and-containers-new-architecture-paradigm>.
- [9] The official website of Red Hat (2018). What's a Linux container?, <https://www.redhat.com/en/topics/containers/whats-a-linux-container>
- [10] Ismail, B., I., Goortani, E., M., Ab Karim, M., B., Tat, W., M. et al. (2015). Evaluation of Docker as Edge Computing Platform, DOI: 10.1109/ICOS.2015.7377291, <https://www.researchgate.net/publication/281445982>
- [11] The official website of Docker, 2018. What is Docker, <https://www.docker.com/what-docker>.

- [12] Abdelshkour, M. (2015). IoT, from Cloud to Fog Computing, <https://blogs.cisco.com/perspectives/iot-from-cloud-to-fog-computing>.
- [13] Athreyas, S. (2017). Microservices its use and its features, <http://www.techjini.com/blog/microservices/>
- [14] Lewis, J., Fowler, M. (2014). Microservices - a definition of this new architectural term, <https://martinfowler.com/articles/microservices.html>.
- [15] Venkateswaran, V. (2016). DevOps - A catalyst for IOT solution development, http://www.infosysblogs.com/engineering-services/2016/12/devops_a_catalyst_for_iot_solutions.html
- [16] Linthicum, D. (2016). What's the Connection Between IoT and DevOps?, <https://www.rtinsights.com/iot-and-devops-security-integration-deployment/>
- [17] The official website of Jenkins (2018). <https://jenkins.io/doc/>
- [18] The official website of Gartner (2015). Gartner Says 8.4 Billion Connected "Things" Will Be in Use in 2017, Up 31 Percent From 2016, <https://www.gartner.com/newsroom/id/3598917>, 2017
- [19] Banica, L., Rosca, D., Radulescu, M., Hagi, A. (2017). Internet-of-Things – A Layered Model for Business Environment, *Annals of "Dunarea de Jos" University of Galati, Fascicle I. Economics and Applied Informatics, Years XXIII – no3/2017*, pp. 47-53
- [20] Dean, M. (2017). 6 best IoT simulators for PC, <https://windowsreport.com/iot-simulators/>
- [21] Microsoft Azure website (2017). Device Simulation walkthrough, <https://docs.microsoft.com/en-us/azure/iot-suite/iot-suite-device-simulation-explore>
- [22] The official website of Iotify - Developer Docs (2018). Develop IoT applications with Virtual Devices, <https://iotify.help/index.html>
- [23] Tetcos website (2018). NetSim - Internet Of Things, <https://tetcos.com/netsim-iot.html>
- [24] IBM Cloud website (2018). IBM Bluemix, welcome to DevOps – Continuous Delivery, <https://console.bluemix.net/devops/try-toolchains>.
- [25] Gupta, H., Dastjerdi, A., V., Ghosh, S., K., Buyya, R. (2017). iFogSim: A toolkit for modeling and simulation of resource management techniques in the Internet of Things, Edge and Fog computing environments, *Software: Practice and Experience (SPE)*, Volume 47, Issue 9, Pages: 1275-1296, <https://doi.org/10.1002/spe.2509>
- [26] National Documentation Center (2016). The Internet of Things in Greece: New possibilities for funding, partnerships and networking accessed on 10/04/2018, available at <http://www.ekt.gr/el/news/19489>

- [27] Hellenic Federation of Enterprises (2017). Digital Greece: The path to growth communications industry digital state, accessed on 10/04/2018, http://www.sev.org.gr/Uploads/Documents/50550/Communications_Industry.pdf
- [28] Operational Programme for Competitiveness 2014-2020 (2016). <http://www.fonduri-ue.ro/poc-2014>, <https://aimas.cs.pub.ro/projects/>.
- [29] European Commission – WiFi4EU Portal (2018). Free Wi-Fi for Europeans, <https://www.wifi4eu.eu/#/home>.