# A Comparative Study of Microservices-based IoT Platforms

Badr El Khalyly[1], Abdessamad Belangour[2], Mouad Banane[3], Allae Erraissi[4]
Laboratory of Information Technology and Modeling
Hassan II University, Faculty of sciences Ben M'sik
Casablanca, Morocco

*Abstract*—**Internet of Things (IoT) is a set of technologies that aim at fitting together smart devices and applications to build an IoT ecosystem. The target of this kind of ecosystem is to enhance interaction between machines and humans through hardware to software binding while reducing cost and resource consumption. On the application level, IoT ecosystems were implemented by various technologies that all seek better interconnection, monitoring, and controlling of IoT smart devices. Among recent technologies, Microservices, which are a variant of the service-oriented architecture, are subject to great excitement. In fact, Microservices are an emerging technology built around Microservice paradigm which goal is to offer services with a small granularity, which exactly meets the distributed nature of IoT devices while maintaining a loosely coupled architecture between IoT components among other advantages. Efforts to build Microservice-based IoT platforms sooner emerged to take advantage of the numerous benefits of the Microservice paradigm to build scalable, interoperable, and dynamic ecosystems. The goal of this paper is to list these approaches, classify them and compare them using a Weighted Scoring Model (WSM) method. This involves, studying these platforms, establishing relevant criteria for comparison, assigning weights for each criterion, and finally calculating scores. The obtained results reveal the weaknesses and strengths of each of the studied platforms.**

***Keywords—IoT platforms; microservices; WSM method; IoT architecture; smart devices***

## I. Introduction

Internet of things is gaining widespread interest and popularity across almost everyday life domains ranging from smart homes, through smart cars or smart factories to smart cities. On the application level, which is our focus here, propositions and solutions are varying according to the interest point of research. Some have been dealing with domain-centric concerns while others are of general architecture purpose.

Among general-purpose architectures, there are Microservices based IoT platforms. Microservices are the next generation of service-oriented architecture designed with the idea of avoiding the hell of monolithic services through diminishing the size of exposed services and characterized with [1] (Componentization via Services, Organization around Business Capabilities, Providing of products, not Projects, Providing of smart endpoints and dumb pipes, Decentralization of Governance, Decentralization of Data Management, Automation of Infrastructure, Design for failure, Evolutionary Design).

The nature of Microservices matches exactly the needs and the requirements of IoT. Hence, many IoT platforms adopt Microservices for their implementations. While some knew how to maximize profiting from Microservices, others did less. Thereby, this paper proposes to make a comparative study of these approaches based on the Weighted Scoring Model (WSM) method. This paper starts our comparative study by extracting relevant criteria for comparison and provides a short description of each criterion (Section 4). These criteria are gathered from related work comparative studies (Section 2) together with other criteria they missed. Then, there is an application of the WSM method, which requires weight attribution to get final scores for each Microservices based IoT platform (Section 3). The obtained scores are represented using a spider chart and are well discussed (Section 5).

## II. Related Work

Many scientific works have as purpose to build an Internet of Things platforms and applications that are based on Microservices. Many researchers have made scientific efforts in this optic to implement the architecture style of Microservices. The authors propose solutions in different domains such as Smart City, Industry, Smart Vehicle, logistics, agriculture. The designs that are adopted in these approaches have been made considering functional and non-functional requirements. Functional requirements means a set of features that related to the domain to which the application is built. Non-functional requirements are a set of technical features that are mandatories in the application like (database management, authentication), and other options that aim to increase quality applications like (historical data, Data classifier).

Several works compare Internet of Thing platforms [2] [3] [4] [4] [5] [6] [7] [8]. All those comparative studies compared monolithic approaches of Internet of Things platforms. The advent of Microservice architecture style pushed the designer to propose many Microservice based Internet of Things platforms and solutions. However, it is relevant to have a study that assembles feature applications and demonstrates their different options and the domain of the function. In this work, there is a comparison of Microservice based Internet of Things efforts. This work is beginning by listing a set of criteria, classify the above listed platforms, and assign to each platform a value that corresponds to a specific criterion. Those criteria represent the features that characterize the platforms. Then, there is a proposition to evaluate other criteria like Interoperability–Customizability–Devops–Domain–Virtual

Objects. This work resides as well in the comparison between these platforms following the criteria. The Weighted Scoring Model is adopted to carry out this comparison, and get the work that is more favored for the most criteria and discuss the result of this comparison, and then this paper provides an opinion to increase the quality of each platform in future works.

Currently, the majority of IoT applications are deployed in the Cloud [9]. This method satisfies the need for IoT in terms of ubiquitous access, high availability, and scalability of computing performance and storage capacity. Also, the Cloud frees users from the specification of many details (eg platforms, dependencies, etc.) and simplifies the deployment and integration of services [10,11]. However, the cloud paradigm is a centralized IT model. This means that all data and requests must be transmitted to cloud data centers which represent centralized points in the heart of the network [12]. With the increase in the number of objects and the amount of data produced, this solution suffers from many problems. Transferring data from network edges to data centers can cause bottlenecks and high and unpredictable latencies. Thus, the QoS will be degraded [13]. Besides, many IoT applications are sensitive to latency, such as telemedicine, the Internet of vehicles, and security (e.g. intrusion detection) [9]. In addition, certain decisions can be made locally (i.e. at the edge of the network), without having to be transmitted to the Cloud [12]. On the other hand, several IoT applications require mobility support, geo-distributed deployment, and localization knowledge that the Cloud cannot efficiently provide [14].

In Chafle et al. [15], the proposed approach aims to adapt defined service processes for composition and during execution by selecting web services as well as by adapting the abstract process to a concrete service process. The approach dissociates the functional and non-functional aspects in different prerequisites used. Thus, the approach makes it possible to react to changes at runtime based on abstract process templates and a directory of services. In Mosincat et al. [16] the proposed approach works on the selection of services for the maintenance of execution plans formalized in BPEL. To do this, the approach uses light execution monitoring systems to detect performance declines and to look for other services to be executed in the event of a fault in order to repair the execution. In Yu and [17], the approach considered works on matching services in terms of QoS needs in order to find services corresponding to the need formulated in particular in terms of QoS parameters with several parameters considered. In Guo et al. [18] an approach is proposed in order to carry out service selection on a larger scale over large areas of candidate services. The service selection process is thus divided into different stages involving a covering algorithm as well as a swarm particle algorithm using Spark. The spanning algorithm is used to reduce the initial search space and the particle algorithm is used to optimize the service selection in relation to the QoS parameters considered. In Guidara et al. [19,20], the proposed approach allows a selection of dynamic services for composition. The goal is to carry out selection actions dynamically to avoid major changes and mistakes in the execution of a pre-established plan. In addition, heuristics are proposed in order to improve the service selection process in the case of QoS constraints and also global and local time constraints. This process allows a cleaning of the space of the candidate services according to the global constraints and a better selection of services to obtain a quasi-optimal solution In Chattopadhyay et al. [21], the proposed approach works to realize an automatic selection of services web. First, an optimal solution search algorithm is proposed to build a solution satisfying the constraints.

QoS of the multi-objective optimization problem posed. Two heuristics are also proposed based on a beam search strategy [22] and a non-deterministic genetic sorting strategy. In Alsaryrah et al. [23], the proposed approach makes it possible to carry out an optimized path search among a set of candidate services for each task to be performed. The advantage of this approach is that it takes into account the energy consumed by the services which can be linked to connected objects. The approach also makes it possible to seek to optimize several parameters by normalizing these and by using a k-shortest path search algorithm. In Gronvall et al. [24], the proposed approach is based on a lightweight framework (SECREST) for composing REST services via the use of user guidance in a care network scenario. The composition is based on service templates that are instantiated into services according to the need. In Guinard et al. [25], the proposed approach aims to integrate REST services provided by connected objects into existing service architectures by integrating these services into existing service directories and by enabling the discovery of services. Other approaches that are based on the semantic Web such as: In [26], the proposed approach aims to achieve being able to determine if services correspond, in particular in terms of inputs/outputs based on semantic modeling of web services. In Kim et al. [27], the proposed platform makes it possible to perform service orchestration based on a user-centered approach as well as an ontology model in order to be able to orchestrate web services and connected objects. However, the management of QoS parameters is not integrated. In [28,29], the authors propose a system for managing semantic data in a Cloud / Big Data context, which can be easily used with a Spark query tool [30,31].

In Fähndrich et al. [32], the proposed approach makes it possible to find a correspondence in a specified process (BPMN) by finding the appropriate services using a semantic engine (based on OWL-S). A system has been designed to perform various stages from the specification of the process to the execution of the services. QoS parameters are not considered, however.

## III. Background

### A. IoT

Internet of Things environments (i.e., IoT: Internet of Things) are becoming more and more present in our contemporary societies. This paradigm was born from the idea of interconnecting different types of objects with each other through the Internet in order to improve the quality of life for humans. Several organizations have taken an interest in the IoT in order to define its different components according to various layered architectures but also the concepts and technologies which make it possible to deploy it.

The IoT is no longer a science fiction subject but evidence in our everyday life. Currently, a human has on average 2 connected digital or electronic devices enabling him to offer intelligent services [33]. The integration of IoT into the daily life of humans makes it possible to offer new applications improving the quality of life through the interconnection of objects. IoT is causing a revolution in the relationship of humans with technology. This revolution concerns various sectors of daily life inducing a change in habits and behaviors. Thus, the IoT has enabled the creation of connected refrigerators, connected shoes, connected electric meters, connected cameras, etc. In this context, the management of our main residence can be carried out remotely and even autonomously and intelligently. This remote and intelligent management can make human life more comfortable, easier, and even safer. Indeed, even critical areas, such as the medical field, are impacted by the integration of IoT technologies by connecting all types of objects. From now on, doctors will be able to diagnose their patients remotely, using sensors worn or implanted on patients. This technological advance improves vital services by optimizing performance and improving the quality of these services.

The IoT is responsible for the collection and/or creation of a large volume of data. This enormous volume of data, known by the term "Big Data" [34,35], allows, on the one hand, to have an incredible wealth in terms of information allowing the offer of advanced services. On the other hand, this volume of data creates new challenges to be considered such as securing, processing, and real-time accessibility of this data [36,37].

The IEEE, through the Webinar “Integrating the IoT and Cultural Heritage in the Smart City”, proposed architecture for IoT based on four layers (Interface layer, Service Logic layer, Data layer, and Resources layer) [38], Fig. 1. Each layer includes specific entities and elements to provide various functionality.

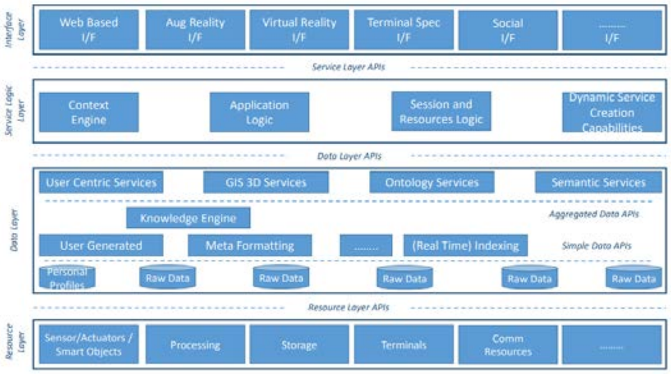

Fig. 1. Internet of Things Architecture of IEEE [38].

## B. Microservices

The first to describe architectures based on microservices are J. Lewis and M. Fowler [39]. This software architecture is a way of dividing a monolithic application into a set of small programs having only one precise task to execute (the service) and also being able to communicate with the other services using a protocol; common to all (usually an HTTP REST API). Microservice-based architecture can be seen as an extension of Service Oriented Architecture (SOA) [40], but with the addition of additional concepts, such as direct communication between services. Microservices are currently mainly used for Cloud applications and Internet of Things (IoT). Much of the literature, such as [41], recognizes many advantages to microservices, such as simpler development and maintenance compared to its monolithic equivalent, with for example a dedicated development team per microservice. This is because, over time, a monolithic application will grow and become more and more complex with the addition of new functionalities to the point of being complex and difficult to understand for a newcomer. In addition, this complexity and the historical code can also block the technologies used because of the changes that are too deep (a good example is banking applications, some of which are still in COBOL). Microservices, while remaining simple, do not have this problem and thus allow them to be more reactive to new technologies. Also, due to the relative simplicity of each service, an application based on microservices can be more easily scaled because only the limiting elements will need to be replicated instead of the complete application. But microservices do not only offer advantages, among the most significant drawbacks, for example, more complex application deployment and the need to implement a communication protocol between microservices to compose them.

## C. Microservice-based IoT Platforms

Many scientific research efforts proposed Microservice based internet of things implementation. These efforts can be classified into seven categories or domains that are:

- Smart Cities: A Smart city is an ecosystem that is rolled out in an urban emplacement, and uses different technologies to collect and process data. The gathering data is analyzed to control and manage traffic and transportation systems, energies, administration services, waste management, and fraud detection. In the literature, there is InterSCity [42], and DIMMER [43], InterSCity doesn’t suggest functional services like DIMMER.

- Smart Commerce: It is a restricted ecosystem that is implemented on the fog level. It is a local system that is rolled out in a set of local devices and servers, and meet the personal need in the stores. Smart commerce provides automation services for customers in the stores. In [51] researchers proposed a solution that is based on Microservices and blockchains and is deployed on AWS. The solution developers adopt shadows that are abstractions of physical objects and provided by AWS.

- Smart Farms: It is a system that is rolled out in the farms to help farmers to gain time and improve the quality of farm services. Author in [50] described a solution that is destinated to the agriculture domain and is based on Microservices. They deployed their solution on three-level: Fog, Edge, Cloud.

- Smart Cars: It is a set of services that aim to automate car components. Author in [49] talked about a Testbed

Architecture using Web of Thing. Their architecture is based on Microservices.

- Smart Factories: It is an ecosystem that is implemented in the factories that automate the production chain. Kontou [44] proposed a solution for Evolvable assembly systems that are based on Microservices. They described a solution on 3 levels: Edge, Fog, and Cloud. They used an ontology-based approach to organize the assembly process as Microservice.

- Smart Logistics: It is a set of services that are rolled out to process data to improve the supply chain quality. Author in [45] described an approach for the transportation planning that is based on Microservices and deployed on the cloud. They described its life cycle as well in experimental validation. Author in [47] proposed a solution for the safe transportation of dangerous products. This solution is based on Microservices that are organized following functional and non-functional requirements.

- Cross-domain or general-purpose: It is a generic system that is designed to any kind of Internet of Things ecosystem. Author in [46] proposed a four-layer model for deploying IoT Microservices in fog computing. In [48] authors adopted the concept of virtual object and web of the object that is an abstraction of a physical object, then they proposed Web of Objects Architecture for IoT Service Provisioning, and implemented a Functional Model of Microservices in Web of Objects Platform. In [52] designed a framework that is composed of four Microservices that contribute to providing Fault tolerance support.

## IV. Comparison of Microservice-based IoT Platform

Weighted Scoring Model is a methodology that helps choosing between compared objects based on a set of criteria [20].

To compare the works that are cited below in the Section 3, the WSM Method is applied [53]. The following steps can perform the application of this method:

- Step 1: Selection of criteria that constitutes platform features.

- Step 2: Assigning weights to group of criteria based on their importance.

- Step 3: Elaboration of a table that contains nominal values of each distribution for each criterion.

- Step 4: Elaboration of a table that contains weights for each criterion. The weight is a percentage. The weight total equals 100%.

- Step 5: For each platform, a calculation of the product score of weights and nominal values is elaborated.

### *A. Comparison Criteria*

The choice of comparison criteria is based on the set of Internet of Things studies. Some criteria from the works that are cited below are extracted. Those criteria are mostly the common characteristic that is shared between Internet of Things platforms:

- Data Management: This criterion shows if the platform can manage the gathering data obtained by things. This criterion is evaluated in Boolean value that demonstrates the availability of this feature/criterion for such a distribution.

- Data Processing: This criterion evaluates if the platform can process data gathering data obtained by things. This criterion is evaluated in Boolean value that demonstrates the availability of this feature/criterion for such a distribution.

- Internet of Things Integration: This criterion shows if the platform provides a hub or gateway that allows to bind things to the platform. This criterion is evaluated in Boolean value that demonstrates the availability of this feature/criterion on this distribution.

- Context Awareness: This criterion shows if the platform provides intelligence to the bound things. This criterion is evaluated in Boolean value that demonstrates the availability of this feature/criterion for such a distribution.

- Specific-domain Services: This criterion if the platform offers services that help facilitates life in the smart city. This criterion is evaluated by calculating the number of services that are specific to the domain to which the platform is built.

- Cloud: This criterion shows if the platform uses the services and features of the cloud. This criterion is evaluated in Boolean value that demonstrates the availability of this feature/criterion on the distribution.

- Scalability: This criterion shows if the platform is scalable. This criterion is evaluated in Boolean value that demonstrates the availability of this feature/criterion on the distribution.

- Security: This criterion shows if the platform is safe for users, this criterion is evaluated in Boolean value that demonstrates the availability of this feature/criterion on the distribution.

- Privacy: This criterion shows if the platform protected user data, this criterion is evaluated in Boolean value that demonstrates the availability of this feature/criterion on the distribution.

- Adaptation: This criterion shows if the platforms can be adapted to updates and changes. This criterion is evaluated in Boolean value that demonstrates the availability of this feature/criterion on the distribution.

- Experimental Evaluation: This criterion shows if the platform is well evaluated on the real world; this criterion is evaluated in Boolean value that demonstrates the availability of this feature/criterion on the distribution.

- Free/Open Source: This criterion shows if other users can access and use the solution freely. This criterion is evaluated in Boolean value that demonstrates the availability of this feature/criterion on the distribution.
- Customizability: This criterion shows if other users can extend the existing code of the platform and develop new features.
- Agile Method [58]: This criterion shows if the designers respect agile methods.
- Virtual Object [57]: This criterion shows if the platforms provides an abstraction of things called Virtual Object.
- Technologies: This criterion shows the technologies that are used to build the platform.
- Functional Features: This criterion shows if the platform has features that do not belong to the technical scope.
- Syntactic Interoperability [54]: This criterion shows if the platform supports heterogeneous actors that communicate using different languages.
- Semantic Interoperability [55]: This criterion shows if the platform supports can exchange data with shared meaning.
- Cross-Platform Interoperability [56]: This criterion shows if the platform can exchange data with external platforms.
- Devops [59]: This criterion demonstrates if the platform respects the technologies regarding continuous development and integration.
- Domain: This criterion shows the number of domains that are covered by the platform.

### *B. Comparison Study*

Table I of nominal values is carried out. For each criterion, the value that corresponds to each platform is assigned. These values are extracted from studies that are cited in the next section.

### *C. Application of Weighted Scoring Model*

Following the application of WSM approach, the table below determines the score of each Microservice based Internet of Things platform. The assignment of weight percentages is realized following the importance of the criterion. Because of their mandatory requirement, the priority is given to this set of criteria: Data management – Internet of Things Integration – Context-awareness – Adaptation – Domain – Security - Scalability - Data Processing. To each criterion of them a weight of 6% is assigned. The second category of importance is given to the criteria: Cloud – Privacy. To each criterion of them a weight of 5% is assigned. The third category of importance is given to the criteria: - Experimental Evaluation - Virtual Object – Technologies - Syntactic Interoperability - Semantic Interoperability - Cross-Platform Interoperability. To each criterion of them a weight of 4% is assigned. This set of criteria is not such a big importance: Specific-domain Services - Functional Features - Free/Open Source – Customizability - Agile Method – Devops. To each criterion of them a weight of 3% is assigned. The total weight equals 100%. Table II shows the result.

TABLE I. NOMINAL VALUES OF CRITERIA CORRESPONDING TO THE PLATFORM

| | **InterScity [42]** | **Dimmer [43]** | **A Cyber-Physical Microservices An IoT-based Framework for Manufacturing Systems [44]** | **Smart ITS sensor for the transportation [45]** | **Leveraging Microservice Architecture for Next-Generation Iot Applications [46]** | **Real-Time HazMat Environmental Information System [47]** | **Microservices in web objects enabled IoT environment for enhancing reusability [48]** | **IoT and Microservices Based Testbed for Connected Car Services [49]** | **SmartHerd management [50]** | **Blockchain-based IoT Platform [51]** | **Microservices Architecture for Reactive and Proactive Fault Tolerance [52]** |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Data Management** | 1 | 1 | 1 | 1 | n/a | 1 | 1 | n/a | 1 | 1 | n/a |
| **Data Processing** | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | n/a | 1 |
| **IoT Integration** | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| **Context Awareness** | 1 | 1 | 2 | n/a | n/a | n/a | 1 | 1 | n/a | n/a | n/a |
| **Specific-domain Services** | n/a | BIM Services, GIS Services, SIM Services | n/a | n/a | n/a | GIS Services– Regulation– Alert – Transport Document- Atmosphere dispertion | n/a | Vehicle data processing | Heat detection, SmartAgri Page, Weather Data, Livestock mobility Monitorin | Shop Service, | n/a |

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | g, Animal welfare | | |
| **Cloud** | 1 | 0 | n/a | n/a | n/a | 1 | 1 | 1 | 1 | 1 | 1 |
| **Scalability** | 1 | 1 | n/a | n/a | | n/a | 1 | 1 | n/a | 1 | 1 |
| **Security** | 0 | 0 | n/a | n/a | | n/a | n/a | 1 | n/a | 1 | n/a |
| **Privacy** | 0 | 0 | n/a | n/a | | n/a | n/a | n/a | n/a | 1 | n/a |
| **Adaptation** | 1 | 1 | 1 | n/a | n/a | n/a | n/a | n/a | 1 | n/a | n/a |
| **Evolvability** | 1 | 1 | 1 | 1 | 1 | 1 | n/a | n/a | 1 | n/a | n/a |
| **Experimental Evaluation** | 1 | 1 | n/a | 1 | 1 | 1 | 1 | 1 | 1 | 1 | n/a |
| **Free/Open Source** | 1 | 1 | n/a | n/a | n/a | n/a | n/a | 1 | n/a | n/a | n/a |
| **Customizability** | 1 | 0 | 1 | n/a | 1 | n/a | 1 | n/a | n/a | n/a | n/a |
| **Agile Method** | 1 | 1 | n/a | n/a | 1 | 1 | 1 | n/a | n/a | 1 | n/a |
| **Virtual Object** | 1 | 0 | n/a | n/a | n/a | 1 | 1 | n/a | n/a | n/a | n/a |
| **Technologies** | PostGres - RabbitMq - Ocean cloud - Docker - REST- Redis- Ruby [6] | LinkSmart – SPARQL- MQTT | CPMS - UML4IoT– MDE – RDF - OWL | REST Python C++ - Shell | Devops – HTTP – JAVA SCRIPT – SSL – Azure | MQTT, Node, MongoDB, PostgreSQ, Qgis server, Angular, | JSON, XML, RDF, HTTP, REST | JWT, ITS-G5, NodeJs, Mongo DB, W3C | NoSQL, MQTT, Java, Docker, Kubernetes, IBM, Fog Computing, Edge Computing, 4G, WIFI. | Block Chain, EVM, JSON, Web3.js, AWS, MQTT, RPC, NAT | CEP, MQTT, REST |
| **Functional Features** | Resource Viewer - Data Collector - Resource Catalog - Actuator Controller- Resource Adaptor | Energy simulator - Energy efficiency - Context awareness- Historical storage - Semantic storage - Resource catalog - Message broker - Service catalog | n/a | n/a | n/a | Data Collection, Hazmat | Microservice Lookup & Discovery Task, Microservice management task, Microservice creation Task, Object Reusing Microservices, Microservice Listener, Authentication, Template Database, Registry Database | Experiment Submission, Report generation, New Service Addition, Discovery, TD Registration, Protocol Binding | Data Acquisition, Data classification, MQTT Publisher, Service Registry, Herd Classification, Alert Service, Local Visualization Webpage, Availability Monitoring, Save Data, NoSQL | Registration, Encryption, Rule Engines, | Predictive Fault Tolerance, Real-time Fault Tolerance |
| **Syntactic Interoperability** | 1 | 1 | 1 | n/a | n/a | n/a | 1 | 1 | n/a | n/a | 1 |
| **Semantic interoperability** | 1 | 1 | 1 | n/a | n/a | n/a | 1 | 1 | n/a | n/a | n/a |
| **Cross-Platform Interoperability** | 1 | 1 | 1 | n/a | n/a | n/a | 1 | n/a | n/a | n/a | n/a |
| **Devops** | n/a | n/a | n/a | n/a | n/a | n/a | n/a | n/a | 1 | n/a | |
| **Domains** | Smart city | Smart city | Industry | Logistic | | Logistic | | Vehicle | Agriculture | Commerce | |

TABLE III. TABLE OF WSM RESULTS

| Criterion | Weight | Requirement Score | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | [42] | [43] | [44] | [45] | [46] | [47] | [48] | [49] | [50] | [51] | [52] |
| **Data Management** | 6 | 6 | 6 | 6 | 6 | 0 | 6 | 6 | 0 | 6 | 6 | 0 |
| **Data Processing** | 5 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 0 | 6 |
| **Internet of Things Integration** | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 |
| **Context Awareness** | 6 | 6 | 6 | 6 | 0 | 0 | 0 | 6 | 6 | 0 | 0 | 0 |
| **Specific-domain Services** | 3 | 0 | 9 | 0 | 0 | 0 | 15 | 0 | 3 | 18 | 3 | 0 |
| **Cloud** | 5 | 5 | 0 | 0 | 0 | 0 | 5 | 5 | 5 | 5 | 5 | 5 |
| **Scalability** | 4 | 6 | 6 | 0 | 0 | 0 | 0 | 6 | 6 | 0 | 6 | 6 |
| **Security** | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 6 | 0 | 6 | 0 |
| **Privacy** | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 5 | 0 |
| **Adaptation** | 6 | 6 | 6 | 6 | 0 | 0 | 0 | 0 | 0 | 6 | 0 | 0 |
| **Experimental Evaluation** | 4 | 4 | 4 | 0 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 0 |
| **Free/Open Source** | 3 | 4 | 4 | 0 | 0 | 0 | 0 | 0 | 4 | 0 | 0 | 0 |
| **Customizability** | 3 | 4 | 0 | 4 | 0 | 4 | 0 | 4 | 0 | 0 | 0 | 0 |
| **Agile Method** | 3 | 4 | 4 | 0 | 0 | 4 | 4 | 4 | 0 | 0 | 4 | 0 |
| **Virtual Object** | 4 | 4 | 0 | 0 | 0 | 0 | 4 | 4 | 0 | 0 | 0 | 0 |
| **Technologies** | 4 | 28 | 12 | 20 | 16 | 20 | 24 | 20 | 20 | 40 | 32 | 12 |
| **Functional Features** | 3 | 15 | 24 | 0 | 0 | 0 | 6 | 24 | 18 | 30 | 9 | 6 |
| **Syntaxic Interoperability** | 4 | 3 | 3 | 3 | 0 | 0 | 0 | 3 | 3 | 0 | 0 | 3 |
| **Semantic Interoperability** | 4 | 3 | 3 | 3 | 0 | 0 | 0 | 3 | 3 | 0 | 0 | 0 |
| **Cross-Platform Interoperability** | 4 | 3 | 3 | 3 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 |
| **Devops** | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 0 | 0 |
| **Domain** | 6 | 6 | 6 | 30 | 6 | 30 | 6 | 30 | 6 | 6 | 6 | 30 |
| **Score** | 100 | 119 | 108 | 93 | 44 | 74 | 86 | 134 | 96 | 130 | 92 | 74 |

## V. DISCUSSION

According to the previous results, the work [48] is the most favored Microservice based platform. Most of the works concentrate their efforts in Data Management, Data Processing, and Agile Methods, which is relevant to a client who needs a completed product without the need to modify it. There is a considerable richness in most of the efforts. All the efforts implement IoT Integration Microservices. This result is reflected in the Multicriteria Radar Graph shown in Fig. 2. Only [42] [43] [49] are Open Source. A considerable effort must be made in Evolvability, Scalability, and Customizability. The customizability is important to allow other stakeholders to customize the Microservices following the needs. Context-awareness and Interoperability still a feature that encounters shortcomings. Many works do not take into consideration the Interoperability factor to connect and bind devices that work with heterogeneous protocols. There is a big lack of security, privacy, and Devops method that must be taken into consideration by researchers.

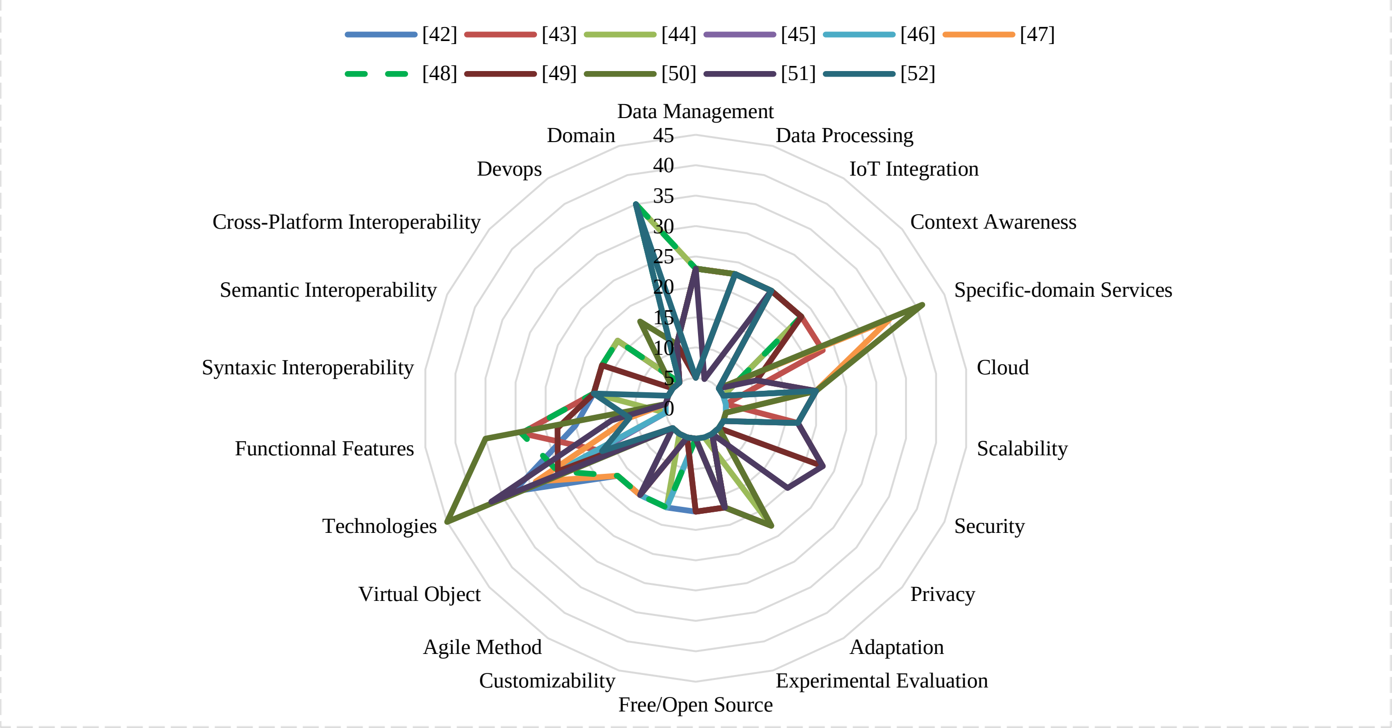

Fig. 2. Multicriteria Spider Graph.

## VI. Conclusion

In this paper, a comparative study of the Microservice based IoT platforms is presented. This work is started by identifying a set of relevant works that adopt Microservices and propose a platform that meets the need of different domains. Then a set of domains that are concerned by these works is enumerated. A set of criteria is identified as well upon which this comparison is based. Based on Weight Score Model method the scores for each of the studied platforms is obtained. These scores or results helped us establishing a general ranking between these platforms but also showed their inner strengths and weaknesses regarding each studied criterion. This study allows recognizing a set of fails that the Microservice based platform suffers from. The main problems that platforms face are contexted awareness and interoperability. The companies have to develop a platform that takes into consideration context awareness, so that they allow the ecosystem to have an intelligence aspect, such as voice recognition, biometrical identification, and another kind of acknowledgment. The developers have to develop an interoperable system that connects heterogeneous devices that use different communication protocols and work in different ecosystem domains. Scalability is a primordial factor that must be the object of scientific research to improve Microservice based Internet of Things platform. Lastly, the researchers and the designers have to adopt Devops culture, which is a set of tools that help improving product quality and reducing development costs, with providing continuous delivery and integration for IoT Designers and Developers.

## References

[1] Martin Fowler and James Lewis. Microservices, 2014. http://martinfowler.com/articles/microservices.html.

[2] Anthopoulos LG, Janssen M, Weerakkody V (2015) Comparing Smart Cities with different modeling approaches. Association for Computing Machinery (ACM), pp 525–528.

[3] Fernández-Vázquez A, López-Forniés I Analysis and comparison of smart cities initiatives.

[4] Nakhuva B, Champaneria PT (2015) STUDY OF VARIOUS INTERNET OF THINGS PLATFORMS. Int J Comput Sci Eng Surv 6:. https://doi.org/10.5121/ijcses.2015.6605.

[5] Solapure SS, Kenchannavar H (2016) Internet of Things: A survey related to various recent architectures and platforms available. In: 2016 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2016. Institute of Electrical and Electronics Engineers Inc., pp 2296–2301.

[6] Guth J, Breitenbücher U, Falkenthal M, et al (2016) Comparison of IoT Platform Architectures: A Field Study based on a Reference Architecture. https://doi.org/10.1109/CIOT.2016.7872918.

[7] Torán D Comparison of different Internet of Things platforms.

[8] Salami A, Yari A (2018) A framework for comparing quantitative and qualitative criteria of IoT platforms. In: 2018 4th International Conference on Web Research, ICWR 2018. Institute of Electrical and Electronics Engineers Inc., pp 34–39.

[9] H. Gupta, A. Vahid Dastjerdi, S.K. Ghosh, and R. Buyya. iFogSim : A Toolkit for Modeling and Simulation of Resource Management Techniques in Internet of Things, Edge and Fog Computing Environments. ArXiv eprints, June 2016.

[10] H. Atlam, R. Walters, and G. Wills. Fog computing and the internet of things : A review. Big Data and Cognitive Computing, 2(2), 2018. ISSN 2504- 2289. doi : 10.3390/bdcc2020010. URL http://www.mdpi.com/2504-2289/ 2/2/10.

[11] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli. Fog computing and its role in the internet of things. In Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing, MCC '12, pages 13–16, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1519-7. doi : 10.1145/2342509.2342513. URL http://doi.acm.org/10.1145/2342509.2342513.

[12] P. Hu, S. Dhelim, H. Ning, and T. Qiu. Survey on fog computing : architecture, key technologies, applications and open issues. Journal of Network and Computer Applications, 98 :27 – 42, 2017. ISSN 1084-

8045. doi : https:// doi.org/10.1016/j.jnca.2017.09.002. URL http://www.sciencedirect.com/ science/article/pii/S1084804517302953.

[13] S. Sarkar, S. Chatterjee, and S. Misra. Assessment of the suitability of fog computing in the context of internet of things. IEEE Transactions on Cloud Computing, PP(99) :46–59, 2015. ISSN 2168-7161. doi : 10.1109/TCC.2015. 2485206.

[14] S. Yi, C. Li, and Q. Li. A survey of fog computing : concepts, applications and issues. In Proceedings of the 2015 Workshop on Mobile Big Data, pages 37–42. ACM, 2015.

[15] Girish Chafle, Koustuv Dasgupta, Arun Kumar, Sumit Mittal et Biplav Srivastava. Adaptation in Web Service composition and execution. Proceedings - ICWS 2006 : 2006 IEEE International Conference on Web Services, pages 549–557, 2006.

[16] Adina Mosincat et Walter Binder. Automated maintenance of service compositions with SLA violation detection and dynamic binding. International Journal on Software Tools for Technology Transfer, vol. 13, no. 2, pages 167–179, 4 2011.

[17] Shih Yuan Yu, Chi Sheng Shih, Jane Yung Jen Hsu, Zhenqiu Huang et Kwei Jay Lin. QoS oriented sensor selection in IoT system. Proceedings - 2014 IEEE International Conference on Internet of Things, iThings 2014, 2014 IEEE International Conference on Green Computing and Communications, GreenCom 2014 and 2014 IEEE International Conference on CyberPhysical-Social Computing, CPS 20, no. iThings, pages 201–206, 2014.

[18] Xing Guo, Shanshan Chen, Yiwen Zhang et Wei Li. Service Composition Optimization Method Based on Parallel Particle Swarm Algorithm on Spark. Security and Communication Networks, vol. 2017, no. 1, pages 1–9, 2017.

[19] Ikbel Guidara, Imane Al Jaouhari et Nawal Guermouche. Dynamic Selection for Service Composition Based on Temporal and QoS Constraints. Dans 2016 IEEE International Conference on Services Computing (SCC), pages 267–274. IEEE, 6 2016.

[20] Ikbel Guidara, Nawal Guermouche, Tarak Chaari, Ikbel Guidara et Nawal Guermouche. Heuristic based Time-aware Service Selection Approach. 2017.

[21] Soumi Chattopadhyay et Ansuman Banerjee. QoS aware Automatic Web Service Composition with Multiple objectives. pages 1–15, 2018.

[22] Christoph Tillmann et Hermann Ney. Word reordering and a dynamic programming beam search algorithm for statistical machine translation. Computational linguistics, vol. 29, no. 1, pages 97–133, 2003.

[23] Osama Alsaryrah, Ibrahim Mashal et Tein Yaw Chung. Energyaware services composition for Internet of Things. IEEE World Forum on Internet of Things, WF-IoT 2018 - Proceedings, vol. 2018-Janua, pages 604– 608, 2018.

[24] E Gronvall, M Ingstrup, M Ploger et M Rasmussen. REST based service composition : Exemplified in a care network scenario. Visual Languages and Human-Centric Computing (VL/HCC), 2011 IEEE Symposium on, pages 251–252, 2011.

[25] Dominique Guinard, Vlad Trifa, Stamatis Karnouskos, Patrik Spiess et Domnic Savio. Interacting with the SOA-Based Internet of Things : Discovery, Query, Selection, and On-Demand Provisioning of Web Services. IEEE Transactions on Services Computing, vol. 3, no. 3, pages 223–235, 7. 2010.

[26] Z Wu, AH Ranabahu et K Gomadam. Automatic composition of semantic web services using process and data mediation. 2007.

[27] Youngjun Kim, Sanghum Lee et Ilyoung Chong. Orchestration in Distributed Web-of-Objects for Creation of User-Centered IoT Service Capability. Wireless Personal Communications, vol. 78, no. 4, pages 1965–1980, 2014.

[28] BANANE, Mouad, ERRAISSI, Allae, et BELANGOUR, Abdessamad. SPARQL2Hive: An approach to processing SPARQL queries on Hive based on meta-models. In : 2019 8th International Conference on Modeling Simulation and Applied Optimization (ICMSAO). IEEE, 2019. p. 1-5.

[29] BANANE, Mouad et BELANGOUR, Abdessamad. Towards a New Scalable Big Data System Semantic Web Applied on Mobile Learning. International Journal of Interactive Mobile Technologies (iJIM), 2020, vol. 14, no 01, p. 126-140.

[30] BANANE, Mouad et BELANGOUR: A Big Data Solution To Process Semantic Web Data Using The Model Driven Engineering Approach, International Journal of Scientific & Technology Research. vol. 9, no 02, 2020.

[31] Banane, Mouad, and Abdessamad Belangour. "A new system for massive RDF data management using Big Data query languages Pig, Hive, and Spark." International Journal of Computing and Digital Systems 9.2 (2020): 259-270.

[32] Johannes Fähndrich, Tobias Küster, F Johannes, K Tobias et Nils Masuch. Semantic Service Management and Orchestration for Adaptive and Evolving Processes. International Journal on Advances in Internet Technology, no. January, 2016.

[33] A. Nordrum, « Popular Internet of Things Forecast of 50 Billion Devices by 2020 Is Outdated », IEEE Spectrum, 2016. Disponible sur : https://spectrum.ieee.org/tech-talk/telecom/internet/popular-internet-of-thingsforecast-of-50-billion-devices-by-2020-is-outdated.

[34] A. Erraissi, B. Mouad and A. Belangour, "A Big Data visualization layer meta-model proposition," 2019 8th International Conference on Modeling Simulation and Applied Optimization (ICMSAO), Manama, Bahrain, 2019, pp. 1-5. doi: 10.1109/ICMSAO.2019.8880276.

[35] Erraissi Allae, et Abdessamad Belangour. « A Big Data Security Layer Meta-Model Proposition ». Advances in Science, Technology and Engineering Systems Journal 4, nº 5 (2019). https://doi.org/10.25046/aj040553.

[36] Erraissi, Allae, and Abdessamad Belangour. "An Approach Based On Model Driven Engineering For Big Data Visualization In Different Visual Modes." International Journal of Scientific & Technology Research.

[37] Fatima Kalna, Allae Erraissi, Mouad Banane, Belangour “A Scalable Business Intelligence Decision-Making System in The Era of Big Data” International Journal of Innovative Technology and Exploring Engineering 2019. https://doi.org/10.35940/ijitee.L3251.1081219

[38] « Webinars - IEEE Internet of Things ». Disponible sur : https://iot.ieee.org/education/webinars.html.

[39] Martin Fowler. Microservices a definition of this new architectural term, 2014.

[40] Kevin Khanda, Dilshat Salikhov, Kamill Gusmanov, Manuel Mazzara, and Nikolaos Mavridis. Microservice-based iot for smart buildings. In Advanced Information Networking and Applications Workshops (WAINA), 2017 31st International Conference on, pages 302–308. IEEE, 2017.

[41] Chris Richardson. Microservice architecture, 2017.

[42] Del Esposte ADM, Kon F, Costa FM, Lago N InterSCity: A Scalable Microservice-based Open Source Platform for Smart Cities.

[43] Krylovskiy A, Jahn M, Patti E (2015) Designing a Smart City Internet of Things Platform with Microservice Architecture. In: Proceedings - 2015 International Conference on Future Internet of Things and Cloud, FiCloud 2015 and 2015 International Conference on Open and Big Data, OBD 2015. Institute of Electrical and Electronics Engineers Inc., pp 25–30.

[44] Thramboulidis K, Vachtsevanou DC, Solanos A Cyber-Physical Microservices An IoT-based Framework for Manufacturing Systems.

[45] Herrera-Quintero LF, Banse K, Vega-Alfonso J, Venegas-Sanchez A (2016) Smart ITS sensor for the transportation planning using the IoT and Bigdata approaches to produce ITS cloud services. In: 2016 8th Euro American Conference on Telematics and Information Systems, EATIS 2016. Institute of Electrical and Electronics Engineers Inc.

[46] BANICA L, STEFAN C, HAGIU A (2017) Leveraging The Microservice Architecture For Next-Generation Iot Applications. Sci Bull - Econ Sci 16:26–32.

[47] Cherradi G, El Bouziri A, Boulmakoul A, Zeitouni K (2017) Real-Time HazMat Environmental Information System: A micro-service based architecture. In: Procedia Computer Science. Elsevier B.V., pp 982–987.

[48] Jarwar MA, Kibria MG, Ali S, Chong I (2018) Microservices in web objects enabled IoT environment for enhancing reusability. Sensors (Switzerland) 18:. https://doi.org/10.3390/s18020352.

[49] Kanti Datta S, Khan MI, Codeca L, et al IoT and Microservices Based Testbed for Connected Car Services.

[50] Taneja M, Jalodia N, Byabazaire J, et al (2019) SmartHerd management: A microservices-based fog computing–assisted IoT platform towards data-driven smart dairy farming. Softw - Pract Exp 49:1055–1078. https://doi.org/10.1002/spe.2704.

[51] A Blockchain-based IoT Platform Integrated with Cloud Services. https://www.researchgate.net/publication/334506644_A_Blockchain-based_IoT_Platform_Integrated_with_Cloud_Services. Accessed 25 Nov 2019.

[52] Power A, Kotonya G (2018) A Microservices Architecture for Reactive and Proactive Fault Tolerance in IoT Systems. In: 19th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, WoWMoM 2018. Institute of Electrical and Electronics Engineers Inc.

[53] Weighted-Scoring Model. http://www.appliedmanagement.com/downloads/. Accessed 4 Dec 2019.

[54] Jovellanos C (2003) Semantic and syntactic interoperability. In: Proceedings of the 4th ACM conference on Electronic commerce - EC ’03. ACM Press, New York, New York, USA, p 266.

[55] Hosseini M, Dixon BE (2016) Syntactic Interoperability and the Role of Standards. In: Health Information Exchange: Navigating and Managing a Network of Health Information Systems. Elsevier Inc., pp 123–136.

[56] Pérez-Berenguer D, García-Molina J (2016) Cross-platform interoperable component for course analytics. In: ACM International Conference Proceeding Series. Association for Computing Machinery, pp 887–892.

[57] Pascual Espada J, Sanjuán Martínez O, Pelayo G-Bustelo BC, Cueva Lovelle JM (2011) Virtual Objects on the Internet of Things. Int J Interact Multimed Artif Intell 1:23. https://doi.org/10.9781/ijimai.2011.144.

[58] Impact of Agile Methodology on Software Development Process. https://www.researchgate.net/publication/255707851_Impact_of_Agile_Methodology_on_Software_Development_Process. Accessed 18 Nov 2019.

[59] Jabbari R, Ali N Bin, Petersen K, Tanveer B (2016) What is DevOps? A systematic mapping study on definitions and practices. In: ACM International Conference Proceeding Series. Association for Computing Machinery.