A Microservice-Based System for Industrial Internet of Things in Fog-Cloud Assisted Network

Fida Hussain Khoso  
Department of Basic Science  
Dawood University of Engineering & Technology  
Karachi, Pakistan  
fidahussain.khoso@duet.edu.pk

Abdullah Lakhan  
Computer Science and IT Department  
Benazir Bhutto Shaheed University  
Karachi, Pakistan  
abdullahrazalakhan@gmail.com

Aijaz Ahmed Arain  
Department of Computer Science  
Quaid-e-Awam University of Engineering, Science and Technology  
Nawabshah, Pakistan  
aijaz@quest.edu.pk

M. Ali Soomro  
Department of Computer Systems Engineering  
Quaid-e-Awam University of Engineering, Science and Technology  
Nawabshah, Pakistan  
kalakot2@gmail.com

Shah Zaman Nizamani  
Department of Information Technology  
Quaid-e-Awam University of Engineering, Science and Technology  
Nawabshah, Pakistan  
ahazaman@quest.edu.pk

Kelash Kanwar  
Department of Electronic Engineering  
Quaid-e-Awam University of Engineering Science and Technology  
Nawabshah, Pakistan  
kelashkanwar@quest.edu.pk

Abstract—Nowadays, the usage of the Industrial Internet of Things (IIoT) in practical applications has increased. The primary utilization is a fog cloud network, which offers different services, such as network and remote edges, at different places. Existing studies implemented the Service-Oriented Architecture (SOA) based on the fog-cloud network to run IIoT applications, such as e-healthcare, e-agriculture, renewable energy, etc. However, due to the applications' monolithic property, issues like failures, security, and cost factors occur, e.g. the failure of one service in SOA affects monolithic applications' performance in the system. With this motivation, this study suggests a microservice-based system to deal with the cost, security, and failure risks of IIoT applications in the fog-cloud system. The study improves the existing SOA systems for e-healthcare, e-agriculture, and renewable energy and minimizes the applications' overall cost. The performance evaluation shows that the devised systems outperform the existing SOA system in terms of failure, cost, and the deadline for all applications.

Keywords—IIoT; security; e-healthcare; SOA; microservice

I. INTRODUCTION

Real world applications of the digital transformation of technologies such as the Industrial Internet of Things (IIoT) possess many advantages [1]. Such applications belong to the e-agriculture, e-healthcare, e-transport, e-servicing, and augmented fields [2]. Cloud computing is an incipient archetype which offers services based on Service-Oriented Architecture (SOA) [3]. Generally, SOA comprises of different services which are monolithically managed by one centralized component [4]. SOA architecture capabilities extend to the fog computing, edge computing, and cloudlet data center to efficiently execute time and latency-sensitive applications [5]. However, fault tolerance, security, and cost of services in SOA architecture depend directly upon the centralized controller. For instance, if any specific service fails or is attacked, then the impact and cost will affect the entire system [6]. The microservice-based system is an innovative solution which offers autonomous service to run IIoT applications. Each microservice executes its application independently, and service failure has no impact on the other services during the process. All microservices are communicating with each other via an REST API. The generalization is a standard interface that exploits the REST API to create cross-platform for all microservices to run one application. Each application requires many microservices to run its tasks. For instance in e-health applications, blood pressure, heartbeat, and ECG tasks each require one microservice for execution [7]. However, a microservice-based system for IIoT has not been developed yet.

The current study extends the existing IIoT work [7] with new suggestions that improve the handling of cost, security, and failure issues of the system in fog-cloud networks. The current work is a survey paper. However, it proposes a solution
for those systems to improve applications' cost and security mechanisms. The paper makes the following contributions to the existing systems: It discusses the IIoT process based on the microservices system. It also discusses industry automation based on microservices with cost and security constraints and e-healthcare services based on microservices connected with the distributed fog-cloud network. Generally, all systems exploit microservice systems in a fog-cloud network with deadline, cost, security, and fault-tolerant constraints. The study solves the offloading and scheduling problems for the considered systems.

II. RELATED WORK

Authors in [1] suggested the SOA aware framework to solve the IIoT mechanism issue of e-agriculture applications in a fog cloud network. The goal was to minimize the end to end latency of the applications. However, they did not consider the security and fault-tolerance of the applications. Fault-tolerant aware SOA is discussed in [2]. The goal was to minimize the failure risks and exploit the primary backup method to handle any SOA architecture failure. However, it consumed much more resources and had high cost during the process. The security-aware framework [3] handles application-level security in SOA architecture. Applications can offload tasks with secure data in the distributed network, however the authors did not consider failures and network and cloud security mechanisms. A hybrid security-aware system based on SOA architecture was suggested in [4]. The goal was to protect data from denial of service and brute force attacks to IIoT applications. The security and failure aware centralized SOA systems for monolithic and coarse-grained applications were studied in [5]. The objective was to improve resource utilization and minimize the application cost. However, the centralized control could fail anytime, and this would affect the entire system in the network. Cost-efficient offloading for healthcare applications based on SOA architecture by considering security and deadlines was investigated in [6]. All considered applications are monolithic and fine-grained and run their tasks distinctly under their deadline. SOA architecture offers services based on an RPC model where users are treated as thin clients and servers as thick clients. However, the centralized failure of any service affects the entire system. An e-agriculture and a body area sensor aware network based on SOA architecture was suggested and studied in [7, 8]. The goal was to optimize the service process with minimum end to end latency of the applications while offering 24/7 services with the robot and faultless services. However, there are many issues in the system. The SOA architecture selects the services from a published pool for consumer and producer and a hacker can publish services and quickly become part of SOA monolithic architecture. The container and function aware IIoT solutions were suggested in [9, 10] to handle the offloading and scheduling problem's transient failure and security issues. These studies achieved better results as compared to SOA architecture. However, the proposed systems have not matured yet for all kinds of IIoT applications.

All reviewed studies proposed systems for IIoT based on the SOA and container function mechanisms. Many of them considered different constraints such as security, failure, cost, and latency. However, failure in service due to hacking or resource-imbalance will impact the overall system performance. The suggested process was very costly and consumed a lot of resources. With the motivations mentioned earlier, this paper suggests microservice aware different IIoT system to minimize failure risk, cost, and security and meet the deadline of applications.

III. THE PROPOSED SOLUTION

The study proposes a different system based on microservice architecture instead of SOA in order to improve cost, failure, and security risk of the applications.

![Microservice based fog-cloud system](image)

Cloud computing is a distributed service provider which offers services at different rates to run any application on the network. Fog computing is an extension of cloud computing which brings cloud resources at the end of a wireless network. The main advantage of fog computing is the reduction of the end to end latency of an application. This study complements the existing architecture [15-24] which is based on SOA. The work creates a system which consists of three layers such as user-end, fog end and cloud end as shown in Figure 1. The user-end is connected with different computer systems that offload their data to the fog server via communication channels for further processing. The fog servers are located at the edge of the network and have short distance and minimum end to end latency during submission or offloading. The basic mechanism design is based on the remote process call which is a popular method allowing the communication channels between users and serves via stud and skeleton objects. The renewable IIoT aware network performs different tasks such as monitoring temperature, humidity, weather, etc. sensors which require microservices to run and save their data in the cloud as shown in Figure 1. Generally, these are real-time tasks,
working 24/7 and generating big data to the fog system. Due to the
time sensitive and security sensitive data, the initial process
will be done on the fog servers. After that, for persistent saving,
the fog node offloads all data to the cloud for further big data
analytic. The high volume of data with various types and
speed will be easily managed by the rich resource cloud
computing. A body-area network is connected with different
sensors for blood pressure, heartbeat, ECG, and online
monitoring tasks at the user level. All tasks offload their data
to the fog node which applies computation by using a unique
microservice to run each task under its deadline. Big data are
generated by the tasks to be offloaded to the remote cloud for
further analyzing.

The main advantage of the system is that the failure of one
microservice will not affect the entire system because they all
are working isolated to each other. Another advantage is that
microservices have execution charges. If one microservice fails
it will not charge for anything until and unless a task is finished
without completing its execution. The fault-isolation, security,
and autonomous execution make this system more efficient and
intelligent in comparison with all the existing SOA
architectures. This system selects microservices from the
published pool which is publically available with different
charges and time-slots. However, the considered system is
smart, if one microservice fails, the system will select another
without any violation of task deadlines.

IV. PERFORMANCE EVALUATION

The performance evaluation was conducted by
implementing the EdgeX Foundry microservice and SOA
architecture to run all IIoT applications. We conducted the
experiment in different cases as shown in Tables I-III. Table I
summarizes the security methods at the user level only and the
execution cost of the applications. Table II summarizes the
study of the failure mechanisms in the case only one system
failure and determines the execution cost of the applications.
Table III shows the results of the comparison based on deadline
methods at the application level only and the execution cost of
the applications.

The overall service cost of security of the monolithic
applications based on SOA architecture became high during the
process due to the centralized handling of all components.
However, microservices had low cost because they are run
inside a container and charged for their execution instead of the
resource provisioning of SOA which charged for all
applications. The primary backup and check pointing always
required a lot of resources in SOA architecture. If one node
fails, it may transfer a task from a service to another node. This
improves the overall performance of the applications, but with
high cost. The bottom line is that users pay for the failure costs
for their applications.

Generally, fog-cloud applications co-operatively work
together to achieve the goals of the applications. However, due
to the monolithic architecture, if a centralized component fails
or is attacked, it needs a lot of time to restart. In this way, the
deadline of and the overall performance of the applications will
be degraded. The deadline of applications directly depends
upon execution time and the availability of resources. The cost
of the applications by combining both failure and security
becomes high when IIoT uses the SOA architecture (Table IV).
The monolithic applications need a lot of resources for their
execution. However, the isolated microservice-based system
works efficiently for all IIoT in the fog-cloud network.

![Table I: Security Mechanisms](image)

<table>
<thead>
<tr>
<th>Reference</th>
<th>Security</th>
<th>Cost</th>
<th>Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1, 2]</td>
<td>RSA</td>
<td>10$</td>
<td>Monolithic</td>
</tr>
<tr>
<td>[3, 4]</td>
<td>RSA</td>
<td>20$</td>
<td>Monolithic</td>
</tr>
<tr>
<td>[5, 6]</td>
<td>MD5</td>
<td>17$</td>
<td>Monolithic</td>
</tr>
<tr>
<td>[7, 8]</td>
<td>SHA</td>
<td>23$</td>
<td>Monolithic</td>
</tr>
<tr>
<td>[9, 10]</td>
<td>CRC32</td>
<td>27$</td>
<td>Coarse-grained</td>
</tr>
<tr>
<td>Approved</td>
<td>RSA</td>
<td>25$</td>
<td>Fine-grained</td>
</tr>
</tbody>
</table>

![Table II: Failure Mechanisms](image)

<table>
<thead>
<tr>
<th>Reference</th>
<th>Fail</th>
<th>Cost</th>
<th>Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1, 2]</td>
<td>Primary backup</td>
<td>10$</td>
<td>Monolithic</td>
</tr>
<tr>
<td>[3, 4]</td>
<td>Primary backup</td>
<td>20$</td>
<td>Monolithic</td>
</tr>
<tr>
<td>[5, 6]</td>
<td>Check pointing</td>
<td>17$</td>
<td>Monolithic</td>
</tr>
<tr>
<td>[7, 8]</td>
<td>Check pointing</td>
<td>23$</td>
<td>Monolithic</td>
</tr>
<tr>
<td>[9, 10]</td>
<td>Check pointing</td>
<td>27$</td>
<td>Coarse-grained</td>
</tr>
<tr>
<td>Approved</td>
<td>Isolated</td>
<td>25$</td>
<td>Fine-grained</td>
</tr>
</tbody>
</table>

![Table III: Deadline](image)

<table>
<thead>
<tr>
<th>Reference</th>
<th>Deadline</th>
<th>Cost</th>
<th>Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1, 2]</td>
<td>Missed</td>
<td>40$</td>
<td>Monolithic</td>
</tr>
<tr>
<td>[3, 4]</td>
<td>Missed</td>
<td>70$</td>
<td>Monolithic</td>
</tr>
<tr>
<td>[5, 6]</td>
<td>Average</td>
<td>85$</td>
<td>Monolithic</td>
</tr>
<tr>
<td>[7, 8]</td>
<td>Reached</td>
<td>75$</td>
<td>Monolithic</td>
</tr>
<tr>
<td>[9, 10]</td>
<td>Reached</td>
<td>99$</td>
<td>Coarse-grained</td>
</tr>
<tr>
<td>Approved</td>
<td>Reached</td>
<td>25$</td>
<td>Fine-grained</td>
</tr>
</tbody>
</table>

![Table IV: Overall SOA and Microservice System Performance](image)

<table>
<thead>
<tr>
<th>Ref.</th>
<th>Security</th>
<th>Cost</th>
<th>Fail</th>
<th>Applications</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1, 2]</td>
<td>No</td>
<td>100$</td>
<td>No</td>
<td>E-agriculture</td>
</tr>
<tr>
<td>[3, 4]</td>
<td>Yes</td>
<td>200$</td>
<td>Yes</td>
<td>E-health</td>
</tr>
<tr>
<td>[5, 6]</td>
<td>No</td>
<td>150$</td>
<td>Yes</td>
<td>E-health</td>
</tr>
<tr>
<td>[7, 8]</td>
<td>Yes</td>
<td>300$</td>
<td>Yes</td>
<td>E-health</td>
</tr>
<tr>
<td>[9, 10]</td>
<td>Yes</td>
<td>500$</td>
<td>Yes</td>
<td>Industry automation</td>
</tr>
<tr>
<td>Proposed</td>
<td>Yes</td>
<td>30, 50, 70$</td>
<td>Yes</td>
<td>ALL</td>
</tr>
</tbody>
</table>

V. CONCLUSION

This study suggests the microservice based system for IIoT
applications instead of the SOA architecture in order to
minimize cost, failure effects, and risk. Simulation results show
that the proposed microservice based system can enhance the
performance of the applications in comparison with the
monolithic SOA architecture. In future work, the Internet of
Vehicle Things and mobility aware microservices in distributed
fog-cloud networks will be considered. The study will design a
security system based on blockchain-enable network in order to
avoid any kind of attack on the system.

REFERENCES

based architecture for secure and trustworthy operations in the industrial
Internet of Things,” Journal of Industrial Information Integration, vol.


