International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 12 Issue 02, February-2023

IoT Gateway Microservice

Jami Vivek, Janya N B, Naman Garg, Preetish Ranjan Chakraborty, Mahalakshmi Manasa P
Department of Computer Science Engineering
DSCE

Abstract **- A digital twin can be defined as a virtual model of an actual physical entity or object or process.**

The following three elements constitute a digital twin: a real-world entity in real space; the digital twin represented as a software; and data involving the linking together of the first two elements.

Digital Twins are providing immense business value by providing insights required to build better products or to improve performance and maintainability of a product in service and the way that product is manufactured.

This technology is key aspect in an organization's digital transformation strategy. Smart cities, wind turbines, self-driving cars and smart manufacturing (Industry 4.0) are few examples which are derived based on these technologies.

With advent of various technological evolutions (such as ML/Big Data, cloud computing, IIOT, Industrial metaverse, block chains etc) and integration of these technologies are enabling this transformation possible.

We intent to build IOT gateway service which enables collection, storing, processing of IoT data.

The combination of IOT and ML to predict live outcomes gives this project a greater scope.

It is used mainly for industry related use cases, for example continuous monitoring of the temperature and pressure to trigger an alert if a certain threshold is exceeded.

Keywords: IOT, Digital Twins; Machine Learning algorithms; Deep learning, IIOT, Cloud Computing, Blockchain;

INTRODUCTION

The main purpose of an IoT gateway is to bring together various devices in the Internet of Things and bridge the gap between them and the cloud. This involves facilitating communication between the devices and transforming the data they collect into useful information.

An authentic IoT gateway is equipped with communication technologies that connect end-devices like sensors and actuators to backend platforms for data, device, and subscriber management. Additionally, it has a computing platform that enables pre-installed or custom applications to manage aspects like data routing, device security, communication, and other crucial aspects of the gateway.

We intend to build an IOT Gateway of Microservices using MQTT and OPC-UA.

The applications include connecting multiple devices to one another and also to any external network like a cloud storage system, translating communication protocols between IOT devices that are manufactured by different companies and also operated differently, filtering raw data and passing on relevant information, mitigating security threats and edge intelligence and computing.

The objectives involve establishing IOT connection to physical devices or to IOT services, collection of live data from IOT sensors as raw data and their appropriate storage followed by live and historical data processing, constructing a Machine-Learning model based on processed data and establishing connection of ML model to the live data acquired from the sensors to predict relevant outcomes.

Intelligent devices require intelligent gateways, and that's where IoT gateways come into play. These gateways, also known as Internet of Things gateways, serve a similar purpose as traditional network gateways by facilitating communication between different technologies and acting as an access point for network data.

IoT gateways function as either devices or software and are positioned between edge systems such as internet-connected devices, sensors, and controllers, and external networks like the cloud. They convert the various protocols or languages utilized by smart devices, such as Wi-Fi, Bluetooth, Ethernet, MQTT, or serial ports, allowing these components to connect with systems that require their data. But what makes IoT gateways smart is their ability to collect sensor data from IoT devices, translate protocols, process the data, and send it out to the network all at the edge of the network.

Performing these tasks at the edge is crucial due to the increasing number of internet-connected devices and sensors, which generate a large volume of data.

IoT gateways play an important role in filtering out routine or insignificant information and only transmitting crucial or unusual data to avoid overloading the system. For example, a single IoT-connected office building could have hundreds or even thousands of sensors measuring temperature, light, air quality, and security systems, each producing data every second.
In this scenario, an IoT gateway would transmit temperature data to the cloud if it exceeds or falls below a pre-determined range, filtering out data for comfortable, room-temperature readings. Along with protocol translation, data filtering and processing, IoT gateways carry out other vital functions such as device connectivity, security, management, updates, and

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 12 Issue 02, February-2023

more. Some advanced IoT gateways even function as platforms for code that can analyse data and initiate actions, such as determining that the temperature is too high and then turning on the air conditioning.

[1] The Internet of Things (IoT) has allowed a wide range of devices to be connected in innovative ways, creating new services and opportunities. However, it also presents significant challenges for programmers.

This paper presents a vision for the future development of IoT systems by reimagining the core building block of IoT as a widely and precisely distributed "microservice," a concept already familiar in web service engineering, rather than as a "thing." The effectiveness of the technique hinges on changes that allow them to meet the main obstacles that IoT systems bring since they differ significantly from more recognised usage of microservice architectures. We contend that combining a microservice approach with other IoT system design techniques might have a mutually reinforcing effect.

We provide a concept for using microservice architecture in IoT systems in this article. We offer two case studies that fit into this architecture as well as a general model of a microservice's internal composition.

Microservices, however, are neither a free lunch nor a panacea. In order to successfully implement microservices, the application domain must be thoroughly understood in order to identify the service border, which may necessitate using extremely sophisticated methodologies or even the arts in challenging real-world application settings. Additionally, when a system evolves, new microservices frequently layer on top of older ones, and software-only microservices frequently depend on hardware, which might eventually result in dependency hell or increased service coupling. From the perspective of a 17 architect, the complexity is frequently still present but has now shifted from the intra-service domain to the interservice domain. Additionally, communication overhead tends to affect fine-grained services more, which reduces the architectural benefit.

Despite these drawbacks, we still think that microservice architecture has promise for the planning and development of an IoT system, mostly because of the latter's unique properties. This provides a summary of the features of IoT systems, relevant design issues, and microservice architectural strategies. Still, there are numerous trade-offs and difficult decisions to be made in order to properly use microservices in an IoT system. We have not addressed all of the unanswered questions in this study. For instance, does the API gateway present a scalability barrier and a single point of failure? Should we attempt to utilise one model to support several protocols, or should we establish a common service level protocol like HTTP to implement the model? Is it possible to utilise REST API everywhere? Who is responsible for assigning names and addresses to devices and services, or for providing them device and service identities? How can different sites work together?

Can we create a "bus" at the service level to transport events and other service requests across several API gateways? So how can we direct those communications to their intended recipients? There are frequently no satisfactory solutions, necessitating more research efforts.

[2] This paper discusses the design and development of new End-to-End Internet of Things (IoT) applications which require cloudification, virtualization, and softwarization of IoT devices and services. The increasing prominence of scalability, extensibility, and interoperability issues in emerging IoT services and platforms is also addressed. The paper covers the high degree of heterogeneity, information systems integration, gateways, and micro-granular architectures, as well as new challenges and solutions for flexible and scalable integration architecture models.

In future work, the NdR application architecture will be thoroughly executed to show its performance and efficiency. This will be carried out by utilizing the previously mentioned technologies in the architecture design. Measurements of forecast accuracy, energy savings for IoT devices, user goal achievement, average microservice response time, and other factors will be used to determine effectiveness. The effectiveness will be evaluated depending on how well the changes were made. The implementation must be expanded to take into account additional difficulties, such as testing and DevOps integration.

The development of early IoT technologies was hindered by limitations in infrastructure. However, the advent of data-centric and infrastructure-free IoT designs, applications, and services has led to a shift in these systems. The creation of new End-to-End IoT applications necessitates important components, such as the cloudification, virtualization, and softwarization of IoT devices and services. Issues with the forthcoming IoT services and platforms' scalability, extensibility, and interoperability are already gaining greater attention. It is advised in this study to use cloud-based microservices to address these issues. The Data Tweet IoT architecture and services that were previously constructed are now being enhanced with a high level of heterogeneity, information systems, integration of virtual IoT devices, gateways, and micro-scale architectural design. The paper looks into alternative techniques for integrating microservices.

The paper describes the effort to modernize the Data Tweet IoT architecture into a data-focused End-to-End IoT system built using microservices. The result is a proposed IoT service that is implemented in virtualized environments of both a Cloud and Edge server, offering a loosely connected, easily expandable, and repeatable solution. A functioning prototype of the complete architecture is available, with a roadside assistance scenario used to showcase its extensibility. The microservice-based architecture is being utilized to create an IoT testing environment for future advancements.

[3] There are many difficulties in designing Internet of Things systems because of heterogeneity, resource limitations,

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 12 Issue 02, February-2023

interoperability, etc. Even while microservice architectures (MSA) have become a popular approach for creating the next generation of IoT systems, they exacerbate these difficulties. This is due to the complexity of managing adaptation considerations that arise at several levels: IoT devices at the level of open and shifting contexts, resource limitations, etc. Microservices at the level of dynamic resource demands, and applications themselves at the level of changing user objectives. Studies have shown that traditional self-adaptation techniques are insufficient for microservice architecture (MSA) systems. Previous works have proposed adaptation at either the architectural or application level, but our proposed solution is a self-adaptive architecture specifically designed for microservice-based Internet of Things (IoT) systems.

The design utilizes machine learning to aid data-driven adjustments and manages adaptions across different levels through the use of different techniques such as: 1) a fog layer at the device level, 2) a service mesh at the microservice level, and 3) dynamic quality of service-aware service composition at the application level.

The proposed self-adaptive design for IoT systems utilizes microservices and will be further developed to demonstrate its efficiency and effectiveness. The implementation of the NdR application architecture will be carried out using techniques such as machine learning to support data-driven adjustments, a fog layer at the device level, a service mesh at the microservice level, and dynamic QoS-aware service composition at the application level. This will be accomplished by implementing the architecture using the aforementioned technologies. Measurements of forecast accuracy, energy savings for IoT devices, user goal achievement, average microservice response time, and other factors will be used to determine effectiveness. The effectiveness will be evaluated depending on how well the changes were made. As indicated in [17], the implementation must be expanded to take into account other issues as DevOps integration and testing.

[4] This paper presents a detailed review of a prototype implementation and performance measurement of edge-based IoT applications created using a microservices-based architecture. The design involves selecting and connecting the required microservices, each with a simple functionality and well-defined interfaces, to create temporal responsive applications. The edge server in the system serves two purposes: it administratively manages the IoT infrastructure and meets the latency and privacy needs of the applications. The study aims to understand the performance impact of using a microservices architecture for building IoT applications at the edge.

By implementing several microservices separately and independently, integrating them to form an IoT application, and maybe sharing microservices between many IoT applications that are operating concurrently to improve interoperability, we show the value of this architecture. Finally, we offer a thorough performance analysis that focuses on CPU and memory usage as well as application delay.

This paper presents the design and implementation of a microservice-based architecture for an edge server in the Internet of Things (IoT). The architecture is tested by building end-to-end IoT applications, which are created by choosing and connecting necessary microservices that provide simple functionalities. The system is divided into two parts: system containers and application containers. System containers are responsible for managing the IoT infrastructure and client interactions, while application containers use sensors and actuators to perform specific functions. The paper highlights the benefits of the architecture, including the independent implementation of various microservices, the creation of an IoT application by connecting these microservices, and the ability to share microservices between different IoT applications. Performance evaluation shows that significant cost savings can be achieved through the use of shared microservices.

[5] Future Internet of Things (IoT) systems are expected to advance through the use of edge computing. It uses calculations and application specific data available on the cloud. Amongst the IOT devices that generate data this methodology minimises both the amount of data transmitted by filtering out garbage data and the network latencies associated with programme execution.

In this paper, the authors present a case study of a distributed microservice architecture applied to providing mobility analysis as a service at the edge, utilizing Multi-Access Edge Computing (MEC) specifications.

In this paper, the implementation of microservices for edge computing was presented, and the potential for future advancements of distributed IoT applications or microservices was demonstrated through the use of edge computing for microservices. This paper supports a modular edge service architecture that is delivered to its respective components.

[6] Internet of Things is the next big industrial revolution. Microservices have curbed monolithic architecture with a more high - quality decentralized approach. Hence, a combination of IOT and microservices offers a powerful strategy and solves some of the biggest problems with monolithic architecture. This paper presents an agile methodology with IOT concepts and modularity.

This is applicable to a variety of fields, including ambient assisted living, among others. The implementation of this article offers a universal IOT microservice architectural solution. The writers adhere to the DevOps organisational development approach's recommendations for testing, deployment, and operational stages.

A more sophisticated software architecture that addresses other issues like cloud management and human-computer interaction could be able to apply the described method with success.
Therefore, in further research, the Micro IOT approach will be evaluated to see if it can successfully adapt to different types of structures.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 12 Issue 02, February-2023

[7] The Internet of Things' vision made it possible to create a vast range of services, applications, and ecosystems that were previously thought to be unfeasible. However,

This study highlights the challenges of lacking standardization in the Internet of Things and how it needs to be addressed. It is recognized that the IoT needs a scalable and performant solution to support a large number of users and data processing, which suggests the use of many small specialized services. The study also takes into account the presence of different systems in the IoT and proposes a microservice-based middleware design to link various IoT devices.

The inventive technical ideas and solutions that led to the fourth industrial revolution, underlines the necessity of integrating disparate devices into a single global ecosystem, but at the same time presents a number of unresolved issues. The importance of open platform design to address the challenge of fragmented closed systems in the Internet of Things is emphasized in this study. The need for scalable applications is emphasized as crucial for maximizing the use of IoT resources, and although there is no single solution to the standardization issue, the study highlights the importance of addressing it.

This article highlights some of the challenges in the Internet of Things (IoT) domain and presents a design for a microservice-based middleware to tackle these challenges. The aim of this architecture is to maintain the scalability of the system while ensuring compatibility across different devices, services, and communication protocols. The article also outlines a regression model data processing pipeline based on the combination of multiple data sources, built using the proposed middleware design.

[8] As IoT technology continues to advance, more and more businesses are looking to build IoT platforms in order to integrate and analyze data from various devices, with the goal of maximizing economic benefits. The IoT platform is designed and implemented in this article using the microservice architecture, spring framework, and docker containers as carriers for the microservices.

The Internet of Things (IoT) platform must be built by more and more businesses as a result of the ongoing development of IoT technology in order to combine and analyse various types of device data and increase economic advantages. The IoT platform is designed and implemented in this article using the microservice architecture, spring framework, and docker containers as carriers for the microservices. This article analyses and defines the IoT platform's needs, discusses the platform's primary functionalities and architectural layout, and illustrates how the platform might be used, for example, to detect partial discharges in electrical equipment.

Kafka is aware of the publish-subscribe model for message generation and consumption, and its

Figure 1 displays a schematic design of an architecture.

Following message production, the producer pushes the message to the broker in push mode while indicating the subject category. Each issue is split up into several parts and kept in several queues. The file offset is utilised as the message's special ID. The customer can then receive the appropriate message based on the subject. Furthermore, Kafka is set up to ensure reliability by configuring the necessary replicas and improve efficiency through sequential read and write operations.

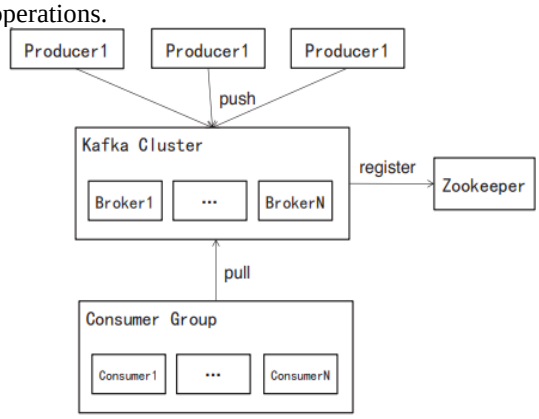

Fig: Architecture of the Kafka message pipeline

[9] By combining heterogeneous cloud, fog, and edge computing paradigms to cooperatively deliver a variety of smart services in our cities and communities, the Internet of Things (IoT) has seen significant technological advancements that make it possible to implement the blueprint for smart cities. In order to enable scalable and extensible services in massively distributed IoT systems, the microservices architecture has been proposed. Attractive characteristics like fine granularity and loose coupling make this possible. Recent studies have measured and analyzed the performance interference between microservices using cloud computing scenarios.

However, given the limitations of the edge device, such as compute usage and network bandwidth, they are not comprehensive for IoT applications. This study examines various edge computing platform microservice deployment policies. Comprehensive experimental results show how well the microservices perform and interact in benchmark scenarios. The microservices are developed as Docker containers.

This study proposes a set of deployment strategies for microservices and examines performance matrices. A simulation test is developed and tested on both a fog computing platform and an edge computing platform in order to assess the microservice deployment policy.
Based on the experimental findings, we draw the following conclusion:

- If we disregard the "one process per container" constraint in an edge computing scenario, operating many microservices in a single container is a viable alternative given the comparable performance.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 12 Issue 02, February-2023

The reasoning for the rule is that there may be an operational issue if we deploy several processes, such as microservices, in a single container. For instance, if one of the microservices requires upgrades, we could need to replace the entire container, which might be disastrous on a huge scale. Occasionally taking resource availability into account, we have no choice left.

• Except when they compete with other services of the same type inside a container, memory-intensive microservices are not significantly impacted. Disk I/O microservices are in a similar predicament, although both of these types have a big impact on CPU-intensive processes.

This is early research for an edge service for smart public safety built on a microservices architecture. Building a bigger edge monitoring system with 32 smart cameras is part of our ongoing work; this will allow for a more thorough analysis for richer insights.

[10] Many of the current industrial applications depend more and more on gathering data from Internet of Things (IoT) smart devices. This data collecting is significant because it may provide insightful information and facilitate quick, informed decisions. This enables companies to more quickly adapt to process changes, reduce downtime, boost production capacity, and enhance overall operational effectiveness. The problem is that the integration of microservices architecture and RESTful API composition may have a significant negative impact on many of these Industrial IOT(IIOT) applications.In addition, the dynamic of service-based settings and requirements is not taken into account by IIOT applications.

The Quality of Service (QoS) characteristics of RESTful APIs and microservices must be considered in order to get past these challenges, especially since these characteristics could change over the course of their existence. In this work, we provide a quality-aware microservices architecture that constantly assesses how these services are doing in terms of providing the necessary functionality. Results of experimental validation are provided in this study, along with an examination of the theories discussed.

In this article, we introduce the framework for managing the quality of service for microservices (mQoSM). In preparation for potential integration with IIOT applications, mQoSM, a middleware that is QoS-aware, assesses the overall quality of microservices.The likelihood of having a stable and long-lasting microservices architecture is considerably increased by the inclusion of non-functional attributes for microservices. The suggested approach has demonstrated its value and efficacy in discriminating between microservices that have comparable functionality and in including mQoS characteristics as part of the selection criterion.

The client's capacity to identify pertinent mQoS factors is necessary for selecting the best microservices.

There appears to be a need to weight each quality factor in relation to the significance or magnitude that it bestows upon ranking microservices based on mQoS parameters in order to account for various mQoS circumstances. In upcoming work, we intend to expand our mQoSM middleware to incorporate a ranking mechanism that will allow applications (or clients) to find pertinent microservices in addition to monitoring their behavior over time.

REFERENCES

[1] Duo Lu; Dijiang Huang; Andrew Walenstein; Deep Medhi, " A Secure Microservice Framework for IoT ", 2017 IEEE Symposium on Service-Oriented System Engineering (SOSE) **DOI:** 10.1109/SOSE.2017.27.

[2] Soumya Kanti Datta; Christian Bonnet, "Next-Generation, Data Centric and End-to-End IoT Architecture Based on Microservices" 2018 IEEE International Conference on Consumer Electronics - Asia (ICCE-Asia)**DOI:** 10.1109/ICCE-SIA.2018.8552135

[3] Martina De Sanctis; Henry Muccini; Karthik Vaidhyanathan, "Data-driven Adaptation in Microservice-based IoT Architectures ", 2020 IEEE International Conference on Software Architecture Companion (ICSA-C), 2018 **DOI:** 10.1109/ICSA-50368.2020.00019

[4] Khaled Alanezi; Shivakant Mishra, "Utilizing Microservices Architecture for Enhanced Service Sharing in IoT Edge Environments ", IEEE Access (Volume: 10), 2022 **DOI:** 10.1109/ACCESS.2022.3200666

[5] Teemu Leppanen; Claudio Savaglio; Lauri Loven; Tommi Jarvenpaa; Rouhollah Ehsani; Ella Peltonen; Giancarlo Fortino; Jukka Riekki, "Edge-Based Microservices Architecture for Internet of Things: Mobility Analysis Case Study", 2019 IEEE Global Communications Conference (GLOBECOM) **DOI:** 10.1109/GLOBECOM38437.2019.9014273

[6] Edwin Cabrera; Paola Cárdenas; Priscila Cedillo; Paola Pesántez-Cabrera, "Towards a Methodology for creating Internet of Things (IoT) Applications based on Microservices ", 2020 IEEE International Conference on Services Computing (SCC) **DOI:** 10.1109/SCC49832.2020.00072

[7] Tomislav Vresk; Igor Čavrak, "Architecture of an interoperable IoT platorm based on microservices ", 2016 39th International Convention on Infrmation and Communication Technology, Electronics and Miroelectronics (MIPRO) **DOI:** 10.1109/MIPRO.2016.7522321

[8] Xiaoyan Sun; Yun Liang; Hui Huang, " Design and Implementation of Internet of Things Platform based on Microservice and Lightweight Container", 2020 IEEE 9th Joint International Information Technology and Artificial Intelligence Conference (ITAIC) **DOI:** 10.1109/ITAIC49862.2020.9339144

[9] Qian Qu; Ronghua Xu; Seyed Yahya Nikouei; Yu Chen, "An Experimental Study on Microservices based Edge Computing Platforms", IEEE INFOCOM 2020 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS) **DOI:** 10.1109/INFOCOMWKSHPS50562.2020.9163068

[10] Eyhab Al-Masri, "Enhancing the Microservices Architecture for the Internet of Things", 2018 IEEE International Conference on Big Data (Big Data) **DOI:** 10.1109/BigData.2018.8622557