International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 05 | May 2020 www.irjet.net p-ISSN: 2395-0072

Microservices Architecture for Building Internet of Things (IoT) Applications

Rajayogi J R[1], Dr. Rajashree Shettar[2]

[1]*Department of CSE, RVCE, Bengaluru, Karnataka, India*
[2]*Professor, Department of CSE, RVCE, Bengaluru, Karnataka, India*

Abstract - *Internet of Things (IoT) has emerged as one of the booming technology in recent years due to its ability to transform everyday physical objects into smarter and intelligent by connecting them to internet, which in turn enrich our lives. From bulbs to refrigerators to parking spaces to houses, IoT has stormed every field and is moving swiftly towards multi-trillion dollar industry. To make IoT more reachable to every domain, the existing approach of application development needs to be replaced with more suitable and robust architecture style. The traditional Monolithic architecture is outdated and does not support complete use of modern technologies, which hinders the application development and business process with respect to IoT. So to cater these needs microservices architectural style was bought into picture, where independent services with their own individual capabilities are developed and combined together to make an application which is more scalable, reliable and flexible in terms of development and delivery. This paper mainly talks about the drawbacks of monolithic architecture style and how microservices architecture is more suitable for building Internet of Things applications using newer technologies.*

Key Words: **Microservices, Internet of Things, Monolithic, Software development, architecture style.**

1. INTRODUCTION

Advancement in communication technology and computation have made a significant contribution to the connected devices which is predicted to expand further to 30 billion devices [1] by the end of the year 2020. Internet of Things(IoT)[2] connects any device (including all cell phones, vehicles, home appliances and other wearable devices embedded with sensors and actuators) to the Internet so that these objects can exchange data on a network with each other. The IoT devices can collect data, transfer data, interact and learn from each other's to make decisions based on information received from the devices like Humans do.

In layman's terms Internet of things is all about taking things in the world and connecting to the internet which is an expansion of internet connection and communication between everyday objects and physical devices. With newer wireless networking devices, sensors, and actuators and better computing capabilities, IoT is storming its way into all the domains and areas. Sensors and actuators are the basic devices, which helps us to interact with the world. Sensors transfer data to the microcontroller, which in turn forward it to another device where it can be analyzed and used to make decisions.

Internet of Things is a fast-growing and trending industry, which is bringing a revolution in the way we live our everyday lives. IoT [2] is rapidly spreading to a number of areas, including personal health care, home automation, Agriculture, Industrial automation, environmental monitoring & smart mobility. IoT is a global network of connected devices. It will not only build machine-to-machine relationships, it will also make machine-to-human and peer-to-peer interactions easier. As a result, in a number of public and private contexts, more and more IoT systems are being introduced, slowly becoming familiar artefacts of everyday life.

With the constant advancement and evolvement of Internet of Things (IoT) [3] , the traditional monolithic architecture of application development cannot handle the complexity of IoT applications which leads to poor scalability, extensibility and maintainability. As contrary to the drawbacks of Monolithic traditional approach, a revised software architecture style - Microservices architecture can be used in IoT applications, which is lightweight, flexible and loosely coupled. In this architecture [4] where each service is loosely coupled and independent from each other are integrated together to make a single application. This architectural style is particularly suitable for Internet of Things domain as many devices are reliable on each other. With microservices this becomes easier as different services can be build, deployed and maintained independently.

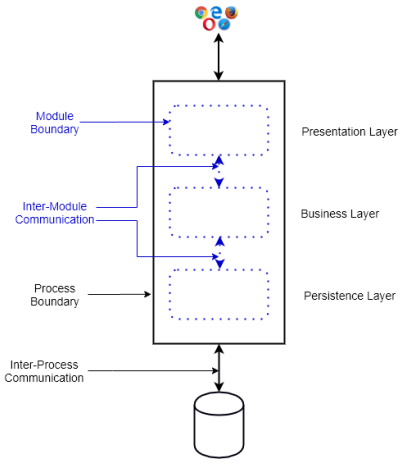

Fig -1: **Monolithic Web Application**

Since the beginning of Software Development (the 1950s), Software applications were developed as an individual system and deployed as an individual process. Fig. 1 is a classic example of Monolithic web application, where the entire application is split into multiple layers like persistence, business and presentation. The persistence layer, which contains the code, is responsible to access the underlying database. Business layer consists of logic to solve the business problem and presentation layer contains the code for handling user interaction e.g., Graphical user interface. So here, the complete software application is deployed as whole, i.e., all or nothing and the application runs as a single process.

Development, Testing and Deployment of monolithic application is easy as there is zero deployment dependencies and cross cutting concerns. However, this is limited to small applications with simple workflow as it has major drawbacks like lack of flexibility where this cannot be used to integrate new technologies when incorporated with monolithic. Apart from that, Scalability of monolithic application is difficult, development speed is very slow, Continuous development, integration is impossible, and this is difficult to use in Internet of things applications, as they are interdependent and complex in nature.

2. MICROSERVICES ARCHITECTURE

2.1 Motivation

Lately there has been rapid changes in software development landscape with introduction of some of the technologies [5] like Cloud Computing, Containerization (Docker) and Devops. Similarly, in the software development, agile method is widely in practice replacing the traditional waterfall model and many high productive, lightweight and easy to use programming languages have flooded the market. Along with this, computing also upgraded with GPU and multicore CPU and new database technologies like NewSQL, NoSQL have emerged replacing traditional databases. Monolithic architecture won't cope up with these technologies, as the software applications are getting complex. To completely use these new technologies a new software architecture style arose i.e., Microservices architecture.

2.2 Introduction

Microservices is an architectural style where an application is built as a collection of autonomous units which are independently deployable services communicating through lightweight protocols, coupled together to fulfil the business capabilities. In addition, it has many end-points like GUI, firmware and Quality analysis. IoT consists of huge deployment ecosystem with multiple applications, protocols, servers, and sensors, which requires more integration between applications, data, and devices. With Microservices this can be achieved seamlessly as it is based on loosely coupled architecture.

2.3 Basic IoT Architecture using Microservices

Microservices [6] are small, independent and loosely coupled services as shown in below Fig. 2 where each service is a separate codebase written in any programming language and managed by a small team. This team can deploy their service independently, rebuild and redeploy their service alone without any changes to the main application.

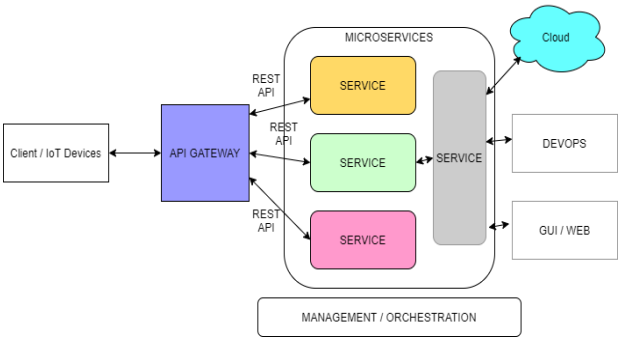

Fig -2: **Basic IoT architecture using microservices**

Each independent service is responsible for preserving its data, control, and external states unlike traditional models where a separate layer handles this. These services communicate between each other using well defined APIs (REST, HTTP). The best thing about microservices is that services compulsorily don't need to share same technology, Programming language, frameworks or libraries.

Besides these, there are some other common components in micro services architecture for building IoT applications such as

i. **Management/Orchestration,** which is mainly responsible for bringing up services and identifying failures and rebalancing the services (Kubernetes, Dockers).
ii. **Physical entity** consists of IoT devices/Physical devices like sensors, Mobiles or Computer Systems.
iii. **API Gateway** acts as an entry point for the clients, where client calls are redirected to appropriate services on the back end. Client calls are not directly called, instead forwarded by API gateway, which decouples clients from services.
iv. **Authorization technology**, which deals with security of microservices. One example is OAuth2 authentication, a well-known authorization protocol.
v. **Messaging protocols** are used for communication between different services. There are different protocols like synchronous (HTTP) or Asynchronous (AMQP), which can be used, based on requirements of the application.
vi. Cloud Infrastructure, which provides many services like storage, analytics, and visualization

to name a few. Advantage of cloud is that we can scale up resources horizontally whenever required.

vii. **DevOps** enables continuous integration and continuous delivery of different services due to change in requirement or functionality of the independent service.

2.4 Advantages of Microservices architecture

The difference between microservices and monolithic architecture [7] is that microservices compose a single application from many smaller, loosely coupled services, while the monolithic is an approach of a large, tightly coupled application. Advantages of using microservices architecture over traditional monolithic architecture are as follows:

i. **Independent components** – all the services can be developed, deployed and updated independently which allows parallel development and saves lot of time and cost for an organization.

ii. **Application scaling** – As services are separate, required resources for the particular service can be scaled at appropriate times as compared to whole application which further impact cost savings.

iii. **Improved fault isolation –** If a single service fails, it will not affect the whole application. A bug in one service will not affect other services or whole application and using better log management will help debug the bug and fix it.

iv. **Easier understanding –** As services are split into smaller components, the complexity of the application is reduced which helps in better understanding and management.

v. **Removes technology barriers –** each service can be developed by using different technology, framework, library and programming language. This decision is taken by the team to choose the technology that makes sense for their service.

vi. **Fast releases –** This architecture allows faster releases as services size is smaller and independent. This can be achieved when combined with better practices and tools such as Agile and Devops.

vii. **Decreases Development Cost and Time –** Time and cost are two major factors an organization look into when building an application, microservices architecture reduces the time of development, as services are build, deployed and managed independently which in turn reduces the cost.

2.5 Drawbacks of Microservices architecture

Microservices Architecture has also its price and a fair share of disadvantages. Some of them are as follows:

i. **Distributed Systems Complexity –** As services are distributed designing, integrating and keeping track of all the service can be challenging especially when the number of independent services are more.

ii. **Operational Complexity –** As services are independent, some of the operations like testing, logging and monitoring has to be done individually for each service, which increases the operational complexity.

iii. **Data Sharing and Data Consistency –** When there is requirement to share data between different services to fulfil the business goal, data consistency and security challenges come into picture, which adds as extra burden for design and development.

iv. **Stability of microservices –** individual services are developed and well tested, once these services are integrated they behave differently and they may interact in unforeseen ways.

v. **Communication and security –** as the services are independent, communication between them is an overhead which needs to be handled by developers separately. Apart from that, security concerns arise, as there are multiple endpoints, which can be accesses by the clients.

3. CONCLUSIONS

In a monolithic architecture, everything is bundled together. If one of the service needs an update or whole application needs an upgrade, the entire application needs to be updated, this is not the case in microservices architecture. Speaking of IoT applications, which are generally complex and involves more components, use of monolithic architecture is generally discouraged. Hence, microservices architecture can be used in building IoT applications which enables using newer technologies and most importantly reduces the development cost and time of an organization.

As technology is fast paced, a new architecture style can come in future, which can overcome the drawbacks of the microservices architecture. Apart from this, custom architecture styles are designed by various organizations for building specific applications. Even Hybrid architectures styles are also used by some organizations to build better applications based on the business needs and requirements of the application.

ACKNOWLEDGEMENT

This study and research is carried out in R.V. College of Engineering, Bengaluru, Department of Computer Science and Engineering, under the guidance of Dr. Rajashree Shettar. I also thank our Head of the Department Dr. Ramakanth Kumar P and the Principal of the institution, Dr. K N Subramanya for providing us all the required facilities and suitable environment to successfully complete this research.

REFERENCES

[1] D. Evans, "The internet of things how the next evolution of the internet is changing everything (april 2011)," White Paper by Cisco Internet
Business Solutions Group (IBSG), 2012.

[2] Rafiullah Khan,Sarmad Ullah Khan, Rifaqat Zaheer and Shahid Khan, "Future Internet: The Internet of Things Architecture, Possible Applications and Key Challenges", 2012 10th International Conference on Frontiers of Information Technology journal , 17-19 Dec. 2012, Islamabad, India, pp. 122-132

[3] L. Sun, Y. Li and R. A. Memon, "An open IoT framework based on microservices architecture," in China Communications, vol. 14, no. 2, pp. 154-162, February 2017.

[4] A. Sill, "The Design and Architecture of Microservices," in IEEE Cloud Computing, vol. 3, no. 5, pp. 76-80, Sept.-Oct. 2016.

[5] K. Bakshi, "Microservices-based software architecture and approaches," 2017 IEEE Aerospace Conference, Big Sky, MT, 2017, pp. 1-8.

[6] D. Lu, D. Huang, A. Walenstein and D. Medhi, "A Secure Microservice Framework for IoT," 2017 IEEE Symposium on Service-Oriented System Engineering (SOSE), San Francisco, CA, 2017, pp. 9-18.

[7] "Design of Microservice Architecture for IOT", International Journal of Emerging Technologies and Innovative Research (www.jetir.org), ISSN:2349-5162, Vol.5, Issue 12, page no.737-740, December-2018,