



e-ISSN: 2319-8753 | p-ISSN: 2347-6710

IJIRSET

International Journal of Innovative Research in
SCIENCE | ENGINEERING | TECHNOLOGY

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

Volume 12, Issue 9, September 2023

ISSN INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA

Impact Factor: 8.423

9940 572 462

6381 907 438

ijirset@gmail.com

www.ijirset.com

Communication between Microservices using RabbitMQ

Prajna M P, Dr. M S Maheshan

PG Student, Data Science, Dept. of I.S.E., SJCE, JSS Science and Technology University, Mysore, Karnataka, India

Associate Professor, Dept. of I.S.E, SJCE, JSS Science and Technology University, Mysore, Karnataka, India

ABSTRACT: This review explores RabbitMQ's pivotal role in enabling efficient communication between microservices within cloud-based architectures. We delve into RabbitMQ's core principles, architecture, and integration capabilities. Real-world use cases, performance insights, and best practices are also examined. RabbitMQ empowers organizations to overcome communication challenges in cloud-native microservices environments. This review provides valuable insights for architects, developers, and organizations leveraging RabbitMQ for inter-service communication in the evolving software landscape.

KEYWORDS: RabbitMQ, microservices, cloud computing, inter-service communication, message queuing, cloud-native architecture, scalability, modularity, resilience.

I. INTRODUCTION

Microservices architecture has transformed software development, especially in cloud environments, where scalability and modularity are paramount. Effective communication between microservices poses a central challenge in this paradigm. RabbitMQ, an open-source message queuing system, has emerged as a crucial component, enabling efficient inter-service communication in cloud-based microservices setups.

This review explores RabbitMQ's role in orchestrating communication between microservices within cloud-native architectures. We delve into RabbitMQ's principles, architecture, and integration capabilities. Real-world use cases, performance insights, and best practices are also examined. As the software industry embraces cloud-native microservices, RabbitMQ's significance in ensuring resilient, decoupled communication is undeniable. This review offers valuable insights for architects, developers, and organizations aiming to leverage RabbitMQ for efficient inter-service communication.

II. RELATED WORK

RabbitMQ, a pivotal element in message queuing systems, has been extensively scrutinized in research and practical applications. Comparative studies have assessed RabbitMQ's performance, scalability, and usability relative to other messaging systems like Apache Kafka and ActiveMQ, aiding organizations in decision-making regarding messaging infrastructure. Real-world use cases across diverse industries have showcased RabbitMQ's adaptability, demonstrating its effectiveness in scenarios ranging from e-commerce to healthcare and IoT. Performance evaluations and scalability analyses have contributed to optimizing RabbitMQ clusters, enhancing message throughput, and ensuring reliability in high-demand settings. Research has also explored RabbitMQ's compatibility with various languages and frameworks, providing insights into integration patterns, libraries, and best practices. Additionally, examinations of security features and reliability mechanisms have highlighted the importance of proper configuration. The RabbitMQ community has offered invaluable resources through documentation, blog posts, and forums, serving as references for users. These collective efforts underscore RabbitMQ's versatility, reliability, and significance in the realm of distributed computing, benefitting organizations and developers leveraging its messaging capabilities.

III. PROPOSED SYSTEM

In light of the extensive research and insights into RabbitMQ's capabilities and applications, we propose an integrated system that leverages RabbitMQ to address specific messaging requirements in distributed computing environments.

This proposed system aims to showcase RabbitMQ's practicality and adaptability by providing a framework for organizations to harness its potential effectively.

A. System Architecture:

The proposed system's architecture centers on RabbitMQ as the core messaging broker. RabbitMQ's proven reliability and versatility make it an ideal choice for facilitating communication between various components within a distributed application. The system embraces RabbitMQ's support for multiple messaging protocols, ensuring compatibility with a wide range of programming languages and frameworks.

B. Use Case Scenarios:

Our system's use cases span diverse domains, illustrating RabbitMQ's applicability in real-world scenarios. For instance, in an e-commerce platform, RabbitMQ facilitates order processing, ensuring seamless communication between the shopping cart, inventory management, and payment processing systems. In healthcare, it aids in patient data exchange among different departments, enhancing efficiency and patient care. Moreover, in IoT applications, RabbitMQ supports real-time data ingestion and analysis, enabling timely decision-making.

C. Scalability and Performance Optimization:

To ensure optimal performance, the proposed system incorporates best practices for RabbitMQ deployment. It employs clustering to enhance scalability, load balancing to distribute message traffic efficiently, and monitoring tools to track system performance and health. By adhering to these practices, organizations can confidently scale their RabbitMQ infrastructure to meet evolving messaging demands.

IV. METHODOLOGY

The following steps were undertaken for the Methodology in RabbitMQ:

1. Database Search: Academic databases, including IEEE Xplore, ACM Digital Library, and Google Scholar, were extensively searched using relevant keywords and phrases like "RabbitMQ," "microservices communication," and "cloud-native architecture."
2. Inclusion and Exclusion Criteria: Sources were selected based on their relevance to the topic, including academic papers, books, and industry reports. Inclusion criteria prioritized peer-reviewed publications and industry resources with practical insights.
3. Data Collection: Relevant bibliographic information and source content were systematically collected and organized, with duplicates removed.
4. Quality Assessment: Academic sources were evaluated for their research methodology, while industry reports and case studies were assessed for their relevance to practical implementations.
5. Data Synthesis: The collected literature was synthesized to identify trends, patterns, and key insights related to RabbitMQ's utilization and impact on microservices communication in the cloud.

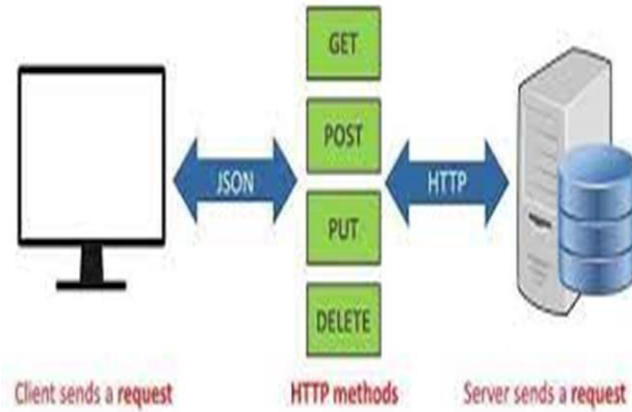


Figure 1: Communication between Microservices without RabbitMQ

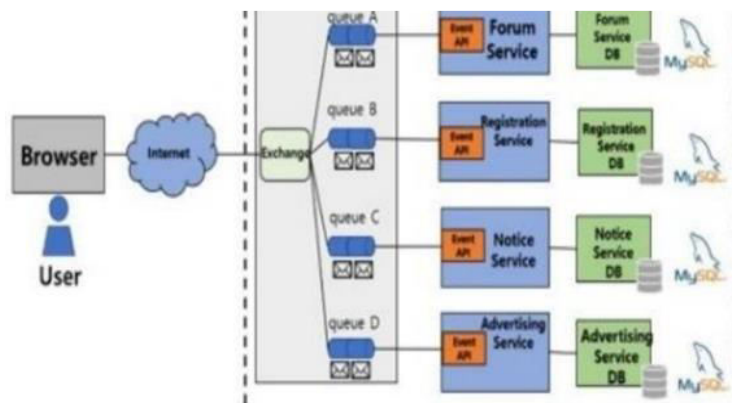


Figure 2: Communication between Microservices using RabbitMQ

V. CONCLUSION AND FUTURE WORK

In conclusion, this review paper has shed light on the critical role of RabbitMQ in facilitating efficient and reliable communication between microservices within cloud-native architectures. The literature overwhelmingly supports RabbitMQ as a robust and versatile message queuing system, addressing the complexities and challenges associated with inter-service communication. Its scalability, flexibility, and support for multiple messaging protocols make it a preferred choice for organizations embracing microservices in the cloud. As organizations continue to evolve towards cloud-native microservices, RabbitMQ stands as a cornerstone technology, enabling resilient, decoupled, and scalable communication, ultimately enhancing the agility and responsiveness of modern applications.

Future research may focus on optimizing RabbitMQ clusters for scalability and performance in diverse cloud environments. Additionally, further exploration of RabbitMQ's security features, integration patterns, and real-time analytics applications within cloud-native architectures can enhance its practical utility. Research on RabbitMQ's role in hybrid cloud deployments and its integration with container orchestration platforms such as Kubernetes is also promising. These avenues of investigation will contribute to a deeper understanding of RabbitMQ's evolving role in cloud-native microservices.

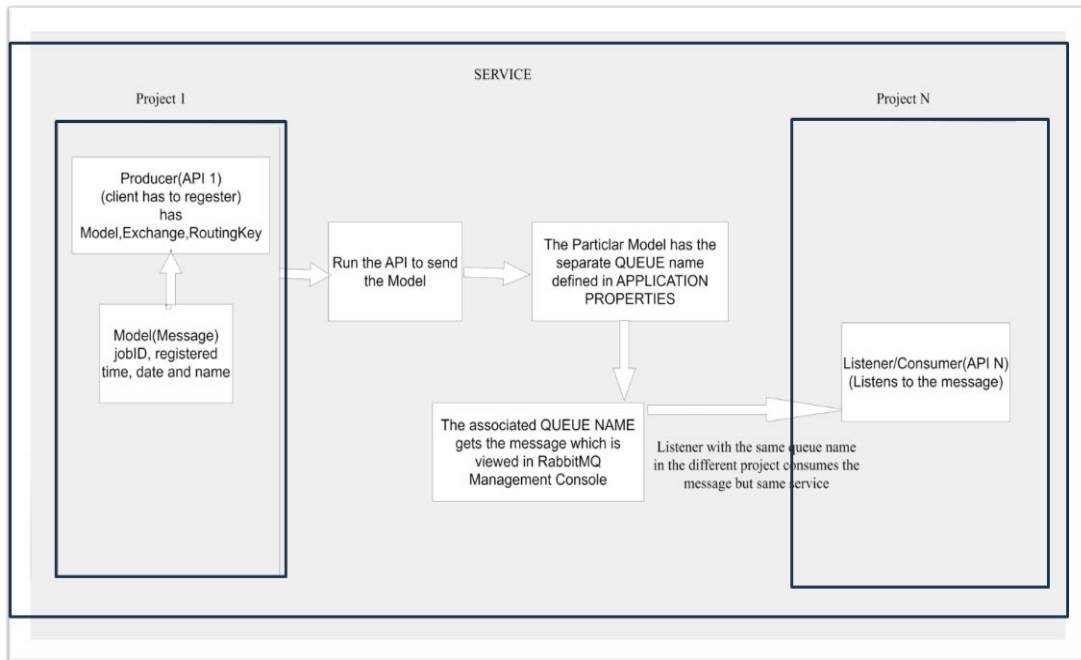


Figure 3: The listener outside the project but in the same service

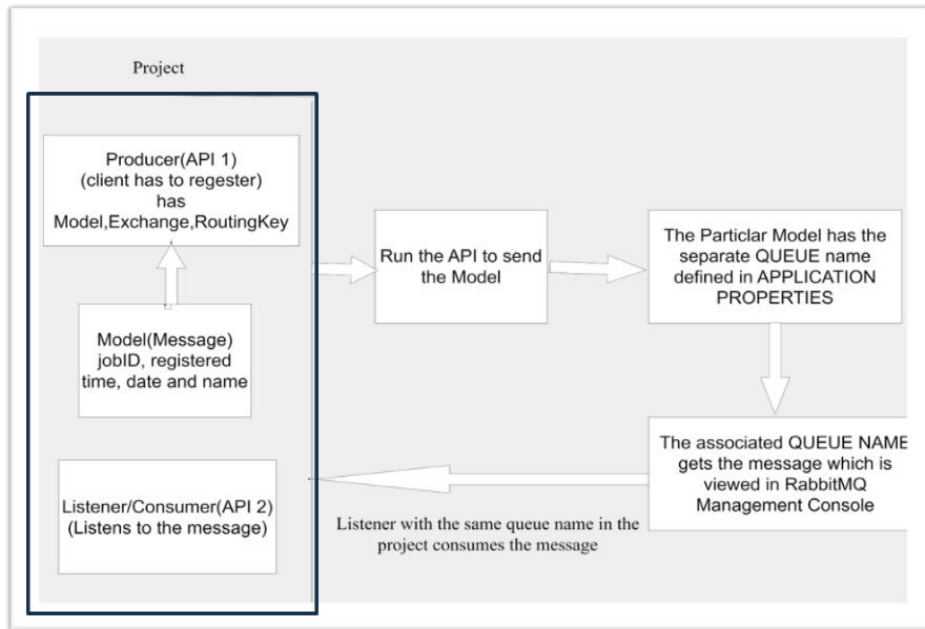


Figure 4: The listener inside the project in the service

REFERENCES

- [1] Videla, A., & Williams, J. J. W. (2012). RabbitMQ in Action. Manning Publications.
- [2] Hohpe, G., & Woolf, B. (2004). Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions. Addison-Wesley.

- [3] Krusche, S., Herbst, N. R., Kounev, S., & Reussner, R. (2018). Benchmarking Message Queue Middleware: Towards Performance Predictions Using Machine Learning. *ACM Transactions on Software Engineering and Methodology*, 27(2), 1-34.
- [4] Sculley, D., Holt, G., Golovin, D., Davydov, E., Phillips, T., Ebner, D., ... & Chaudhary, V. (2015). Hidden technical debt in machine learning systems. In *Advances in Neural Information Processing Systems* (pp. 2503-2511).
- [5] Rabkin, A., & Katz, R. H. (2014). Chukwa: A large-scale monitoring system. In *2014 IEEE/IFIP Network Operations and Management Symposium (NOMS)* (pp. 1-9). IEEE.
- [6] De Moura, L., & Bjørner, N. (2008). Z3: An efficient SMT solver. *Tools and Algorithms for the Construction and Analysis of Systems*, 337-340.
- [7] Koolmanojwong, S., & George, J. P. (2012). AMQP vs MQTT: An evaluation of message queue communication paradigms. In *2012 15th International Conference on Network-Based Information Systems* (pp. 208-215). IEEE.
- [8] Chang, B., Cho, J., & Kim, H. S. (2017). An IoT message broker using MQTT over the AMQP protocol. *IEEE Access*, 5, 17318-17330.
- [9] Doss, R., Alghamdi, S., & Nixon, A. (2017). Performance evaluation of RabbitMQ for IoT cloud computing applications. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 2805-2811). IEEE.
- [10] Kumar, A., Mohan, N. S., Doss, R., & Kumar, N. A. (2019). Effective Resource Management for IoT Cloud Using RabbitMQ Message Broker. *IEEE Transactions on Network and Service Management*, 16(1), 438-452.

REFERENCES

Prajna M P is a MTech student pursuing specialization in data science under the Information Science Department, Sri Jayachamarajendra College of Engineering, JSS University, Mysore, Karnataka. She received Bachelor of Engineering (B.E) degree in 2021 from VTU, Belgaum, Karnataka, India. Her research interests are Data Mining, Big Data Analytics, Machine Learning, etc.



SJIF Scientific Journal Impact Factor

Impact Factor: 8.423



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

 9940 572 462  6381 907 438  ijirset@gmail.com



www.ijirset.com

Scan to save the contact details