International Journal of Core Engineering & Management
Volume-6, Issue-11, 2021 ISSN No: 2348-9510

BUILDING MICROSERVICES WITH CQRS AXON FRAMEWORK- CHALLENGES AND BEST PRACTICES

Hareesh Kumar Rapolu
hareeshkumar.rapolu@gmail.com

Abstract

The research paper has identified the methods to build microservices with the help of the CQRS Axon framework. Additionally, the paper has defined microservices and the Axon framework. It has explained the multiple protocols necessary to implement the framework. The basic focus of this assignment has remained the challenges and benefits of the framework. The document has listed several challenges and benefits based on contemporary business scenarios.

Keywords- Command, query, event-driven,event-sourcing, data synchronization.

I. INTRODUCTION

Command Query Responsibility Segregation (CQRS) is a cyber security architecture pattern. The basic usage of this service is to separate audio and typing operations through different models. It often collaborates with the Axon Framework in a microservice environment. Microservices are generally understood to be architectural approaches. These types of approaches are used to build applications that help to distribute and collaborate the code or program. The basic advantage of microservice is the combination so that the application will not break down. CQRS is generally used in microservice to enable independent interpretation of audio and typed models. For CQRS separation, a system uses the 'Application Logic= Command Model+Query Model' equation. This equation explains the parameters of the command model that handles written operations and the query model that handles audio operations. This can help minimize lock contention and enhance a particular system's performance under cyber traffic.

International Journal of Core Engineering & Management
Volume-6, Issue-11, 2021 ISSN No: 2348-9510

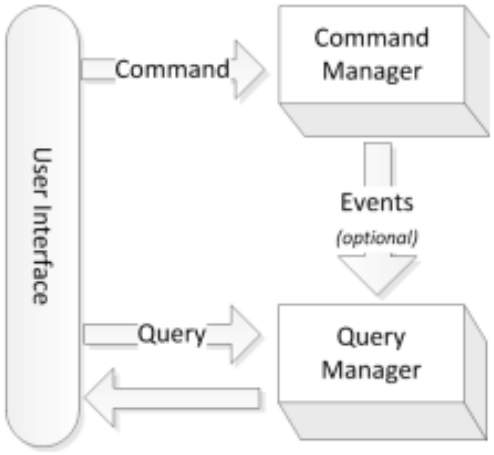

Figure 1: Core CQRS principles

II. MICROSERVICES

Microservices generally operate using the consistency model. This helps programmers to understand the scalability and availability of microservices. The equation used to ensure proper working of the consistency model is 'Consistency=Eventual Consistency≠Strong Consistency'. Microservices are generally used by cybersecurity architects or developers. It is a versatile platform and provides multiple benefits. Generally, this platform helps to understand the applications of the Java ecosystem by focused and independent methods. It further provides an event-sourcing mechanism . For this, the cyber security architect uses the equation, 'Current State=Σ(Past Events)'. In order to elaborate the equation, it can be stated that the events are stored sequentially and the summation of past events helps to understand the current state. It is also beneficial to use this platform through CQRS because the collaboration between reading and writing models helps to evaluate the current structure of the Java ecosystem. It is also helpful for the framework to combine coupled components to integrate their micro-operations.

The basic framework of microservice is different from traditional monolithic applications. On one hand, traditional applications help to understand the attributes of the Java ecosystem holistically. Alternatively, microservices which the CQRS framework communicates using different kinds of lightweight protocols. These protocols are generally termed REST (REpresentational State Transfer) or gRPC (general Remote Procedure Calls) . Besides the protocols, there is also an equation in place that helps in microservices communication. 'Service communication= Events+Commands/Queries' is the equation used. The events are the models for broadcasting changes while the commands are queries are communication techniques directly requested from the event store. Command units are used in terms of representing intentions to implement the required query or command handling. The unit is noted as 'CreateOrderCommand'. Contrary to the implementation of traditional monolithic frameworks, microservices working with the help of these protocols focus on smaller domains of business

International Journal of Core Engineering & Management
Volume-6, Issue-11, 2021 ISSN No: 2348-9510

systems. They focus on individual capability, development and beta testing regarding each component. It needs to be mentioned that microservices are combined loosely. The dichotomy of microservices enhances its benefits by providing system flexibility.

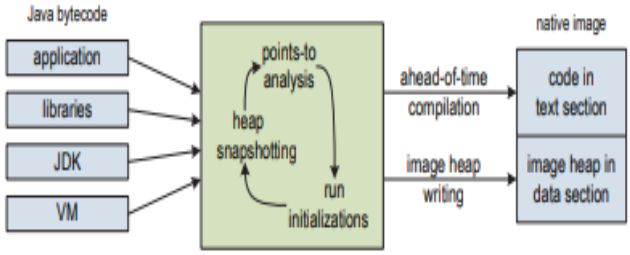

Figure 2: Java class sets

III. CQRS AXON FRAMEWORK

Axon Framework is a type of code or software that is built with the purpose and is open-sourced. The fundamental purpose of this Framework is to analyse the smaller domains of systems. It also helps in determining system flexibility by understanding the capability of the domains along with their combined attributes . It provides building blocks for modern-day applications using protocols life event-driven architecture (EDA). These protocols are majorly powered by domain-driven design (DDD), event sourcing and CQRS. With the help of this Framework, individuals can focus on business objectives and determine methods to fulfil them.
It is also helpful for system evaluation and updating. Another important point that needs to be mentioned regarding the Framework is that it genuinely reduces redundant complex software problems. The major benefits provided by the framework to their users are efficiency, reliability, haste accessibility and event-driven . Generally, event-driven ideas are generated through aggregate units defined through OrderAggregate. It also mitigates the hassle of beginning from square one. In addition to this, it is also an effective custom-designed JVM-based application that delivers results personalized for the business objectives.

IV. BEST PRACTICES OF BUILDING MICROSERVICES WITH CQRS AXON FRAMEWORK

Resilience and scalability

The Framework ensures proper scalability options for a system. An equation is used to understand the resilience and scalability of the framework. The equation is 'Resilience=Retries+CircuitBreakers+Saga Patterns'. In addition to this, optimised data protection provides performance optimisation through the CQRS mechanism . The command service generally uses a NoSQL database to separate return operations and query services implemented in reading operations.

International Journal of Core Engineering & Management
Volume-6, Issue-11, 2021 ISSN No: 2348-9510

Flexibility and maintainability

The separation of handling commands (writing operations) from handling queries (read operations) with the help of CQRS explains the responsibility of the platform. It has also been understood that the separation of query and command helps individuals to understand, evolve and maintain the system for a long period of time. In addition to this, singular microservice among CQRS architecture encompasses particular business domains. The particularity in question helps to simplify the updating and replacement of smaller focus domains without overhauling the entire system.

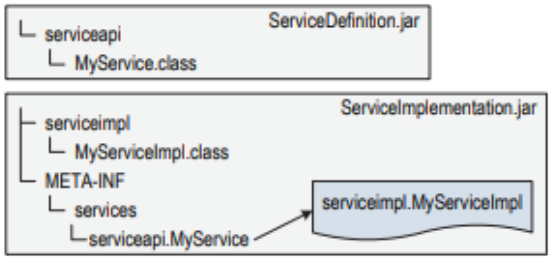

Figure 3: Event stocks

V. CHALLENGES OF BUILDING MICROSERVICES WITH CQRS AXON FRAMEWORK

Increased complexity

Since micro services focus on smaller domains of business it is often complex to implement. It is a Framework exclusively preferred by cybersecurity architects and coders. It needs separate types of commands, event sourcing, consistency mechanism and query path to be properly implemented. The event units are generally identified as Order Created Event. It is evident that without proper IT training, it will not be possible to implement CQRS.

Consistency management

The requirement of eventual consistency maintains a balance between both query and command. It is challenging to maintain the balance with proper concurrency and data replication. Another issue regarding consistency management is synchronization . It is essential to update data and commands. The proper updating of commands and queries leads to the reflection of accurate query results. It needs to be mentioned again that proper education is important for understanding these mechanisms. It will be difficult to conduct these operations with simple training.

VI. CONCLUSION

The implementation of microservices through the CQRS Axon framework is legit important in the contemporary business scenario. However, the basic difficulties faced are related to implementation. It is inherently used by cybersecurity architects and programmers. Therefore, the scope of the program is reduced because of the lack of proper education. Institutions require

continuous training programs and relevant reviews and checks to analyse the implementation of these techniques.

Abbreviations and acronyms

- **CQRS- Command Query Responsibility Segregation**
- **REST- REpresentational State Transfer**
- **gRPC- general Remote Procedure Calls**
- **EDA- Event-driven architecture**
- **DDD- Domain-driven design**
- **JVM- Java virtual machine**
- **NoSQL- Not only Structured Query Language**

Units

- **Command units- CreateOrderCommand**
- **Aggregate units- OrderAggregate**
- **Event units- OrderCreatedEvent**

Equations

- **Separation- Application Logic= Command Model+Query Model**
- **Consistency model- Consistency=Eventual Consistency ≠Strong Consistency**
- **Event sourcing- Current State=Σ(Past Events)**
- **Resilience and scalability- Resilience=Retries+CircuitBreakers+Saga Patterns**

REFERENCES

1. Dunie, D. Miers, J. Wong, M. Kerremans, K. Iijima, and P. Vincent, "Licensed for Distribution Magic Quadrant for Intelligent Business Process Management Suites," Jan. 2019. Available: https://b2bsalescafe.wordpress.com/wp-content/uploads/2019/09/gartner-magic-quadrant-for-intelligent-business-process-management-suites-january-2019-1.pdf
2. Wimmer et al., "Initialize once, start fast: application initialization at build time," Proceedings of the ACM on Programming Languages, vol. 3, no. OOPSLA, pp. 1–29, Oct. 2019, doi: https://doi.org/10.1145/3360610.
3. J. K. Krottje and A. van Ooyen, "A Mathematical Framework for Modeling Axon Guidance," Bulletin of Mathematical Biology, vol. 69, no. 1, pp. 3–31, Oct. 2006, doi: https://doi.org/10.1007/s11538-006-9142-4.
4. L. Coppolino, S. D'Antonio, G. Mazzeo, and L. Romano, "A Comparative Analysis of Emerging Approaches for Securing Java Software with Intel SGX," Future Generation Computer Systems, May 2019, Available: https://www.researchgate.net/profile/Giovanni-Mazzeo/publication/331845512_A_comparative_analysis_of_emerging_approaches_for

_securing_java_software_with_Intel_SGX/links/5ca70c774585157bd322fa78/A-comparative-analysis-of-emerging-approaches-for-securing-java-software-with-Intel-SGX.pdf

5. M. Ibrahim B and M. F. Hassan, "CONSTRUCTION OF CUSTOMIZABLE SOA SECURITY FRAMEWORK USING ARTIFICIAL NEURAL NETWORKS," JurnalTeknologi, vol. 12, no. 3, pp. 69–75, Oct. 2016, Available: https://journals.utm.my/jurnalteknologi/article/download/10023/5992
6. R. de Rooij, K. E. Miller, and E. Kuhl, "Modeling molecular mechanisms in the axon," Computational Mechanics, vol. 59, no. 3, pp. 523–537, Dec. 2016, doi: https://doi.org/10.1007/s00466-016-1359-y.
7. S. Barakat , "Monitoring and Analysis of Microservices Performance," Journal of Computer Science and Control Systems, vol. 10, no. 1, May 2017, Available: https://www.researchgate.net/profile/Saman-Barakat/publication/317690568_Monitoring_and_Analysis_of_Microservices_Performance/links/5b3208330f7e9b0df5cbd14c/Monitoring-and-Analysis-of-Microservices-Performance.pdf