

[Open in app ↗](#)

♦ Member-only story

# Difference between SAGA and CQRS Design Patterns in Microservices?

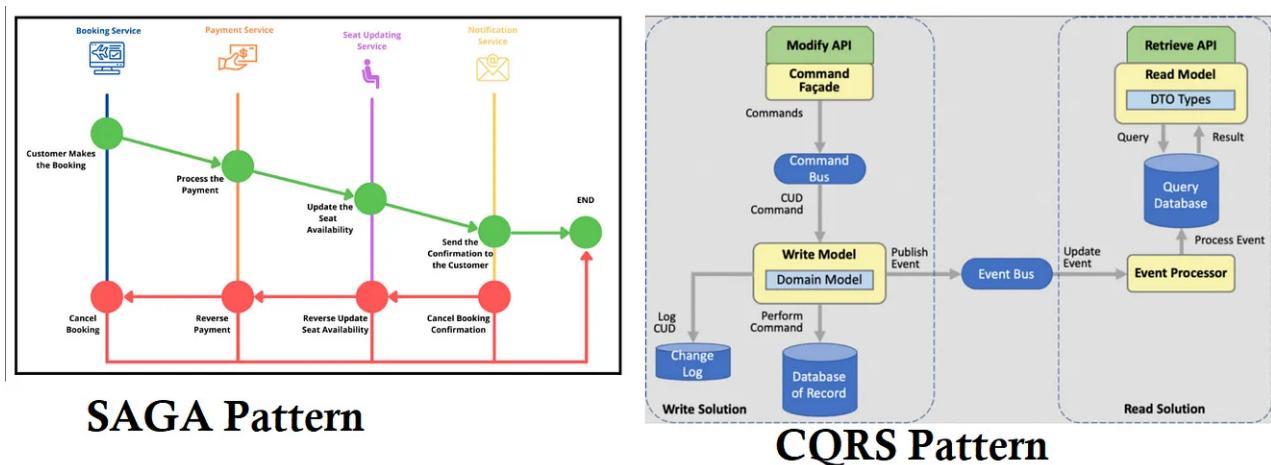
A popular Microservice interview question which you cannot miss



Soma · Following

Published in Javarevisited

6 min read · Mar 2

[Listen](#)[Share](#)[More](#)

Hello folks, if you have heard about SAGA pattern and CQRS pattern in Microservice world and wondering what are they and when to use them then you are at the right place. Difference between SAGA and CQRS Pattern is one of the common question on [Java interviews](#).

Both SAGA Pattern and CQRS are both popular patterns used in microservices architecture, but they serve different purposes.

The main difference between SAGA and CQRS is that SAGA focuses on the **transactional consistency** of the system while CQRS focuses on the **scalability and performance** of the system but there are much more to learn about them which you will find in this article.

In the past, I have mentioned about SAGA Pattern when I shared 50 Microservice Interview Questions for beginners and experienced developers and 10 essential Microservice design principles and in this article, In this article, I will share the difference between two and when to use SAGA and CQRS pattern in Microservices.

## Top 10 Microservice Design Patterns for Experienced Developers

10 essential Microservices design pattern to create robust, scalable, and reliable Microservices

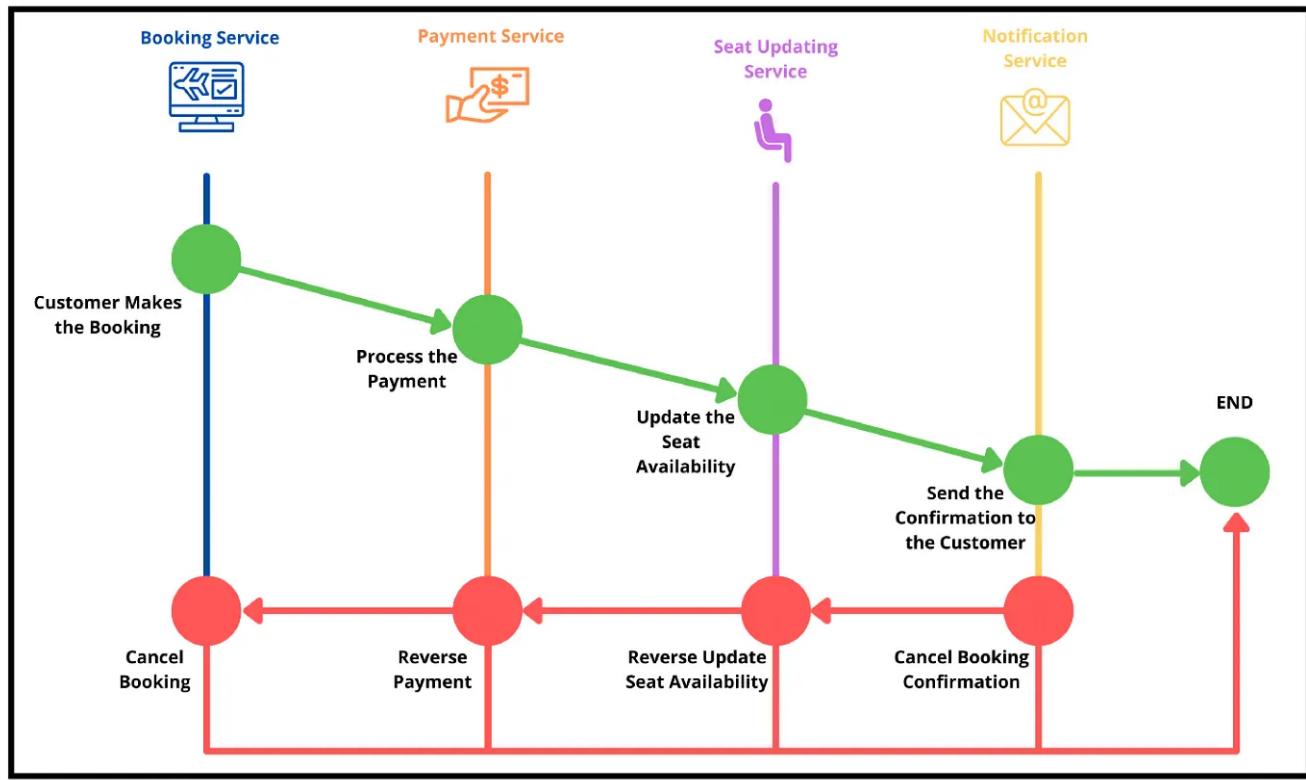
medium.com

### What is SAGA Pattern in Microservice Architecture?

The SAGA pattern is used to manage transactions across multiple services. It ensures that each service involved in a transaction commits or rolls back its part of the transaction, even if other services fail.

For example, in an e-commerce application when a user initiates an online purchase, and this involves multiple services like payment processing, order management, and shipping.

*The SAGA pattern can be used to ensure that if any of these services fail, the transaction is rolled back, and the user's money is refunded.*



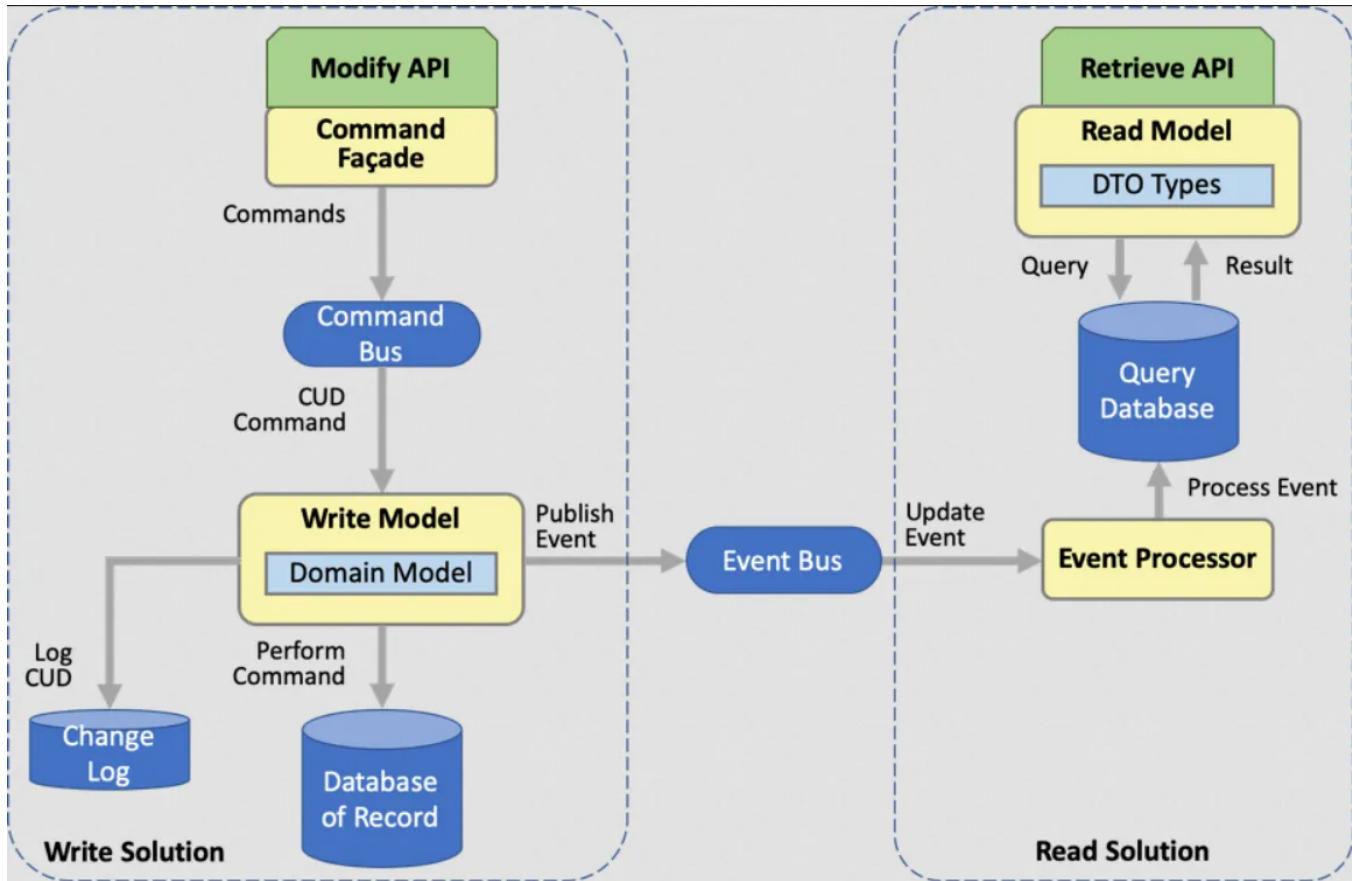
### What is CQRS Pattern in Microservices? What Problem does it Solve?

The CQRS (*Command Query Responsibility Segregation*) pattern is used to separate the read and write operations of an application into different models. It allows for different models to be optimized for their respective tasks.

For example, consider a banking application where users can view their account balance, transaction history, and make transfers. The read operations (like view account balance) can be optimized for high availability and low latency, while the write operations (e.g., make a transfer) can be optimized for data consistency and accuracy.

*By separating the read and write operations, we can choose the appropriate storage and processing mechanisms for each.*

Here is how a CQRS pattern look like in Microservice architecture:



In summary, SAGA and CQRS are two different design patterns used in microservices architecture.

SAGA is used to manage transactions across multiple services, while CQRS is used to separate the read and write operations of an application into different models.

## Difference between SAGA and CQRS Design Pattern in Microservice Architecture?

Now that you know the basic difference between SAGA and CQRS Pattern let's see more difference between them. Sure, here is a tabular comparison between SAGA and CQRS patterns in microservices:

### 1. What problem they solve?

SAGA is a pattern used for managing a sequence of distributed transactions, where each transaction updates data in multiple services, while CQRS is a pattern that separates the read and write operations in a system, using separate models for each.

### 2. Functionality

SAGA is used to ensure data consistency across multiple microservices when a business transaction requires updates to multiple services, while CQRS pattern is

used to improve system scalability, performance, and availability by separating read and write operations and optimizing each separately.

### 3. Working

SAGA coordinates transactions using a series of compensating transactions that roll back changes made by the previous transaction if an error occurs. On the other hand, CQRS uses a different model for read and write operations. The write model is optimized for data modification, while the read model is optimized for querying and data retrieval.

### 4. Consistency vs Scalability

SAGA is a solution for maintaining data consistency in a distributed system, where data is spread across multiple services. On the other hand, CQRS is a solution for scaling up a system by separating read and write operations and using different models for each.

### 5. Complexity

SAGA Pattern can be complex to implement and manage, as it involves coordinating multiple transactions across multiple services., while CQRS can simplify the management of complex systems by separating the read and write operations and using optimized models for each.

### 6. When to use?

SAGA pattern is best suited for scenarios where multiple microservices need to be updated as part of a single business transaction. While CQRS is best suited for scenarios where read and write operations are heavily used and need to be optimized separately.

SAGA pattern is typically used in systems that **require strong transactional consistency**, such as financial systems, while CQRS pattern is used in systems that require high scalability and performance, such as social media platforms.

### 7. Implementation

SAGA pattern can be implemented using either the choreography or orchestration approach, while CQRS pattern is typically implemented using event sourcing.

## Java and Spring Interview Preparation Material

Before any Java and Spring Developer interview, I always use to read the below resources

## Grokking the Java Interview

[Grokking the Java Interview: click here](#)

I have personally bought these books to speed up my preparation.

You can get your sample copy [here](#), check the content of it and go for it

[Grokking the Java Interview \[Free Sample Copy\]: click here](#)



*If you want to prepare for the Spring Boot interview you follow this consolidated ebook, it also contains microservice questions from spring boot interviews.*

## Grokking the Spring Boot Interview

You can get your copy here – [Grokking the Spring Boot Interview](#)



That's all about the **difference between SAGA Pattern and CQRS design Patter in Microservices architecture**. The key difference between SAGA and CQRS is that SAGA focuses on the **transactional consistency** of the system while CQRS focuses on the **scalability and performance** of the system

It's also important to note that both patterns can be used together in a microservices architecture to achieve different goals.

In fact both SAGA and CQRS patterns can be used together in a microservice architecture to achieve both strong transactional consistency and high scalability and performance. However, this can add additional complexity to the system, so make sure you analyse the situation and pros and cons before using SAGA or CQRS Pattern.

By the way, these are just a two of the [many microservice design patterns](#) that are available which you can learn on your own, I will also be sharing more patterns in future. .

And, if you are not a Medium member then I highly recommend you to join Medium so that you can not only read these articles but also from many others like Google Engineers and Tech experts from FAANG companies. You can [join Medium here](#)

**Join Medium with my referral link - Soma**

Read this and every story from thousands of great writers on Medium). Read more from great writers on Medium like...

medium.com

## My Other Articles you may like to read

### Difference between Horizontal Scalability vs Vertical Scaling in System Design and Microservices

Hello folks , one question which I have been asked repeatedly on many System design and Java Developer interviews are...

medium.com

### Top 15 Side Hustles for Programmers and Developers for Additional Income in 2023

These are the 10 most effective ways for programmers to create additional income streams and passive income in 2023

medium.com

### 20 Advanced Core Java Interview Questions for Experienced Developers of 5 to 10 Years

These are difficult and advanced core Java questions for 5 to 10 years Experienced Java developers

medium.com

Microservices

Java

Programming

Software Engineering

Software Development



Following

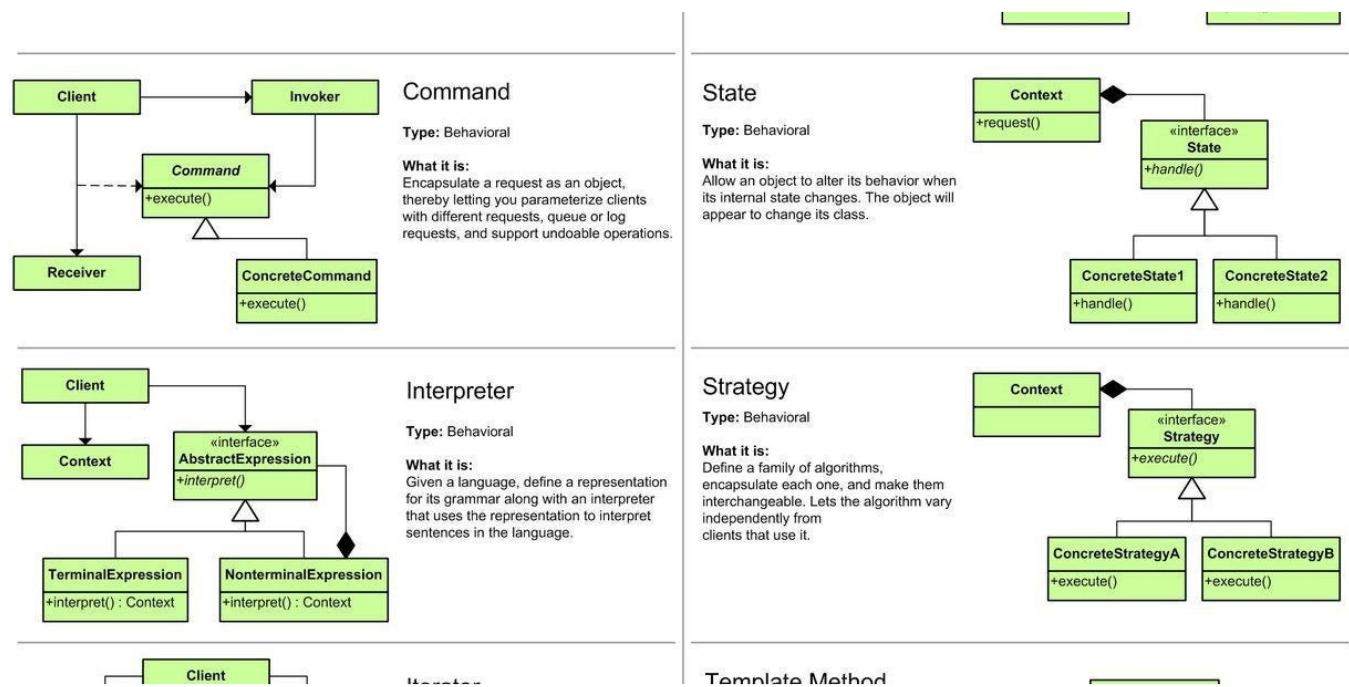


# Written by Soma

4.1K Followers · Editor for Javarevisited

Java and React developer, Join Medium (my favorite Java subscription) using my link [👉](https://medium.com/@somasharma_81597/membership)  
[https://medium.com/@somasharma\\_81597/membership](https://medium.com/@somasharma_81597/membership)

## More from Soma and Javarevisited



 Soma in Javarevisited

## 10 Object Oriented Design Patterns Every Programmer Should Learn + Design Pattern Cheat Sheet

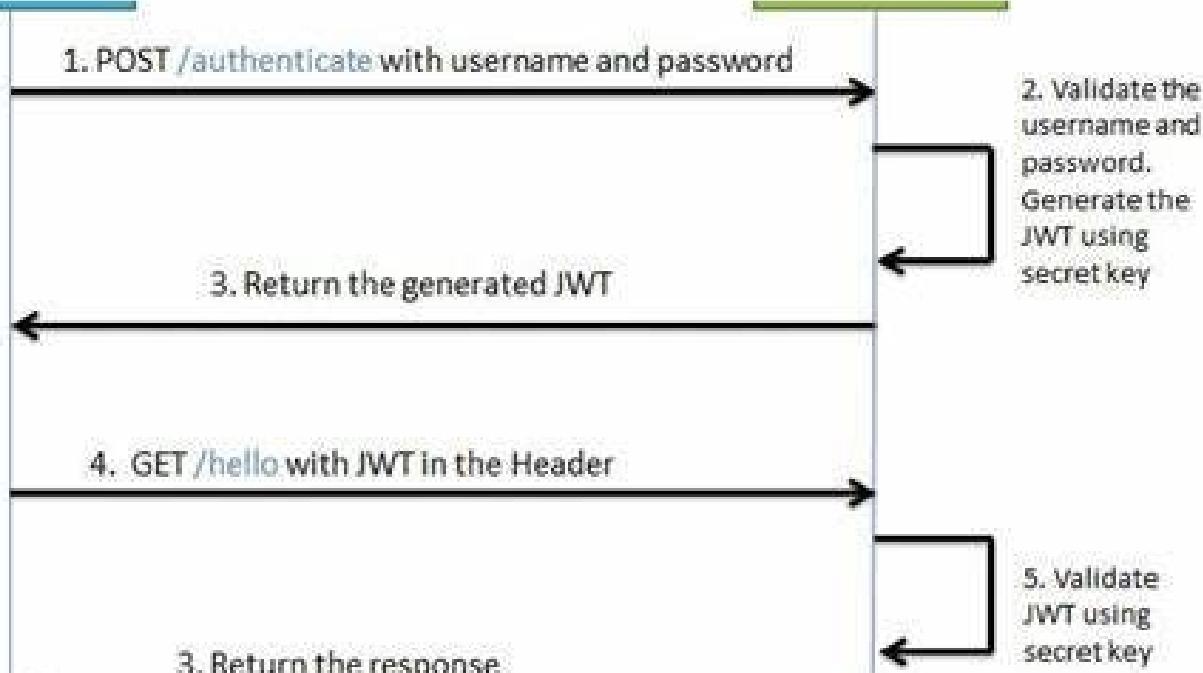
Design patterns are key for writing cleaner code and knowing them opens your mind

👉 · 18 min read · Jun 26

 220 



...



 Kishore Karnatakpu in Javarevisited

## How to secure Spring Boot with JWT Authentication and Authorization

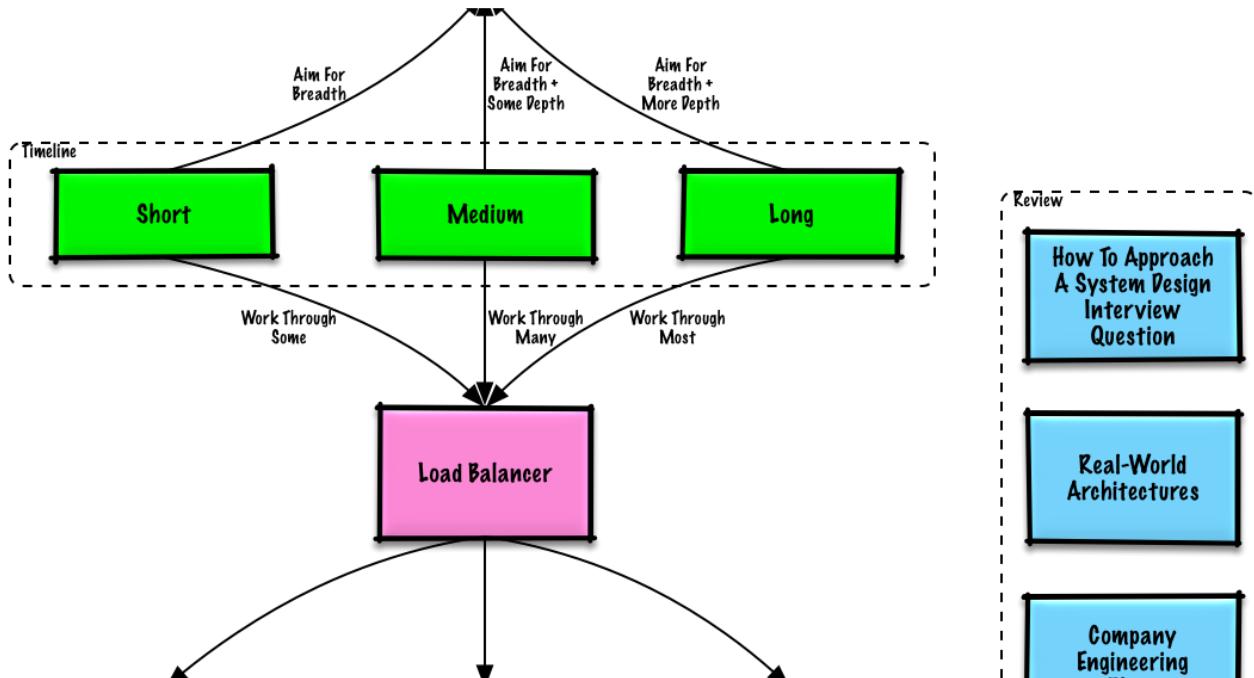
Hello learners, here we are going to learn about spring security implementation with spring boot3.0 and JWT. This is in continuation with...

4 min read · Apr 19

 200  16



...



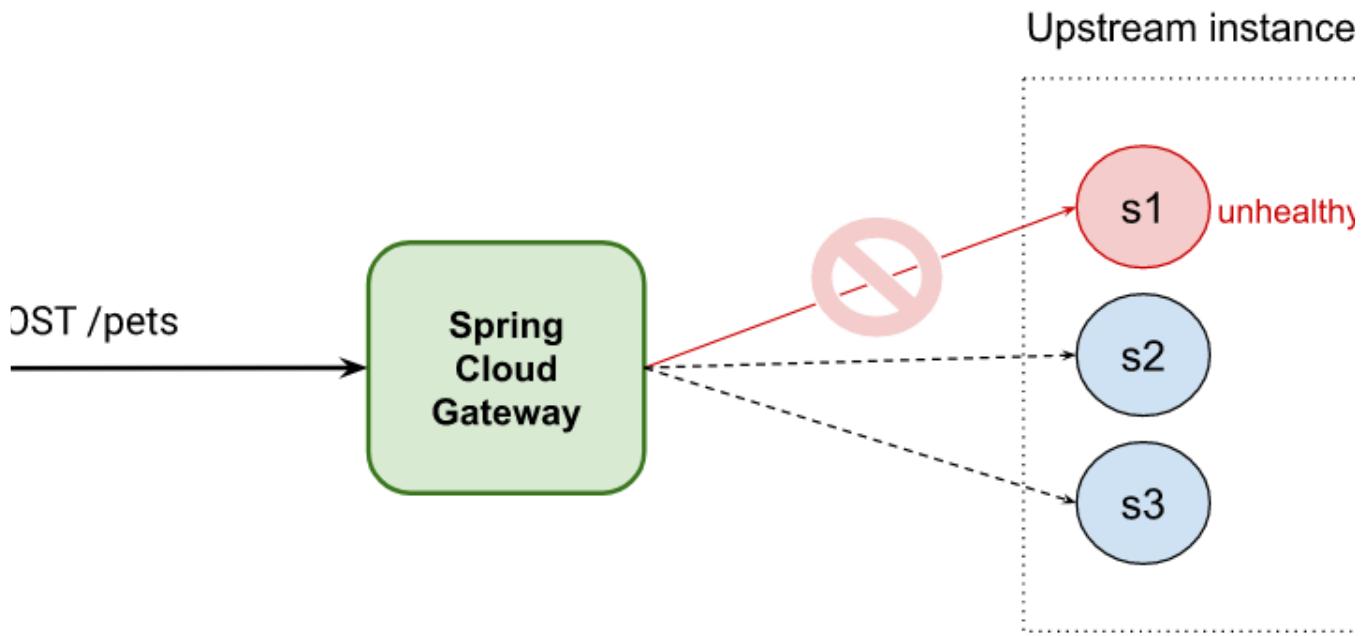
 javinpaul in Javarevisited

## 10 Best System Design Courses for Coding Interviews in 2023

These are the best online courses to learn System Design from Udemy, edX, and Educative for coding interviews in 2023.

18 min read · Oct 22, 2020

1K 1



Soma in Javarevisited

## Top 20 Spring Cloud Interview Questions with Answers for Java Developers

Common Spring Cloud Questions for Java Interviews with answers

· 9 min read · Jul 23

184

See all from Soma

See all from Javarevisited

## Recommended from Medium



 Amit Himani

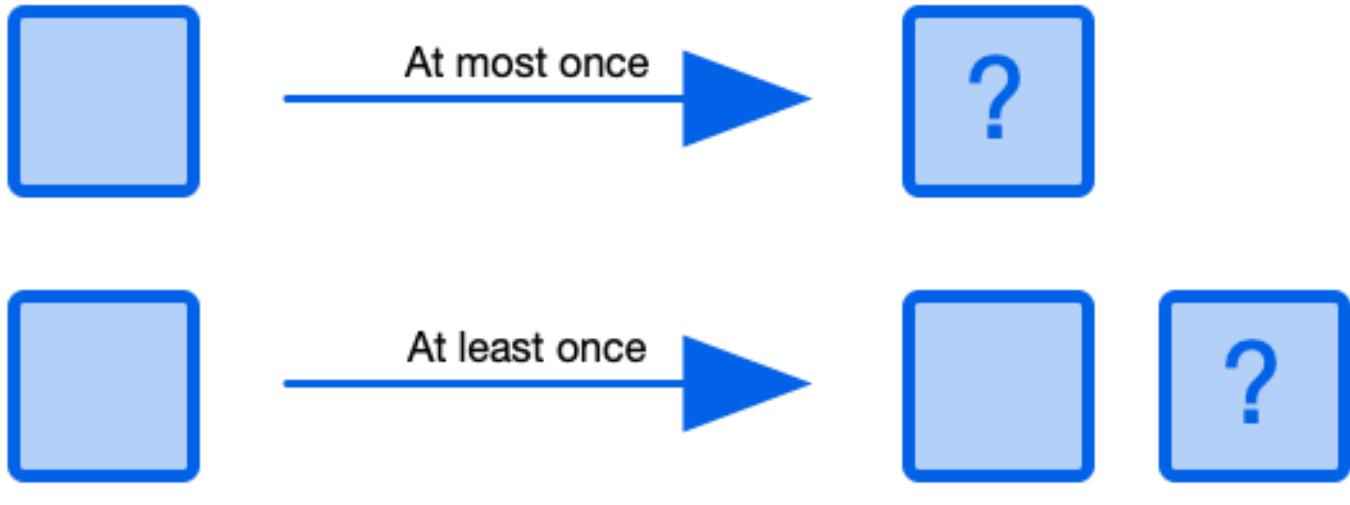
### Distributed Transaction in Spring Boot Microservices

Distributed transactions in microservices refer to transactions that involve multiple microservices, each handling a part of the...

★ · 5 min read · Feb 17

 49  3



Andy Bryant

## Processing guarantees in Kafka

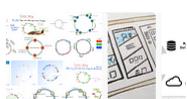
Each of the projects I've worked on in the last few years has involved a distributed message system such as AWS SQS, AWS Kinesis and more...

21 min read · Nov 16, 2019

1.91K    5

...

## Lists



### General Coding Knowledge

20 stories · 192 saves



### It's never too late or early to start something

13 stories · 66 saves



### Stories to Help You Grow as a Software Developer

19 stories · 261 saves



### Coding & Development

11 stories · 97 saves



 Rupert Waldron

## Create a non-blocking REST Api using Spring @Async and Polling

Get the great asynchronous, non-blocking experience you deserve with Spring's basic Rest Api, polling and @Aysnc annotation.

12 min read · Feb 26



...

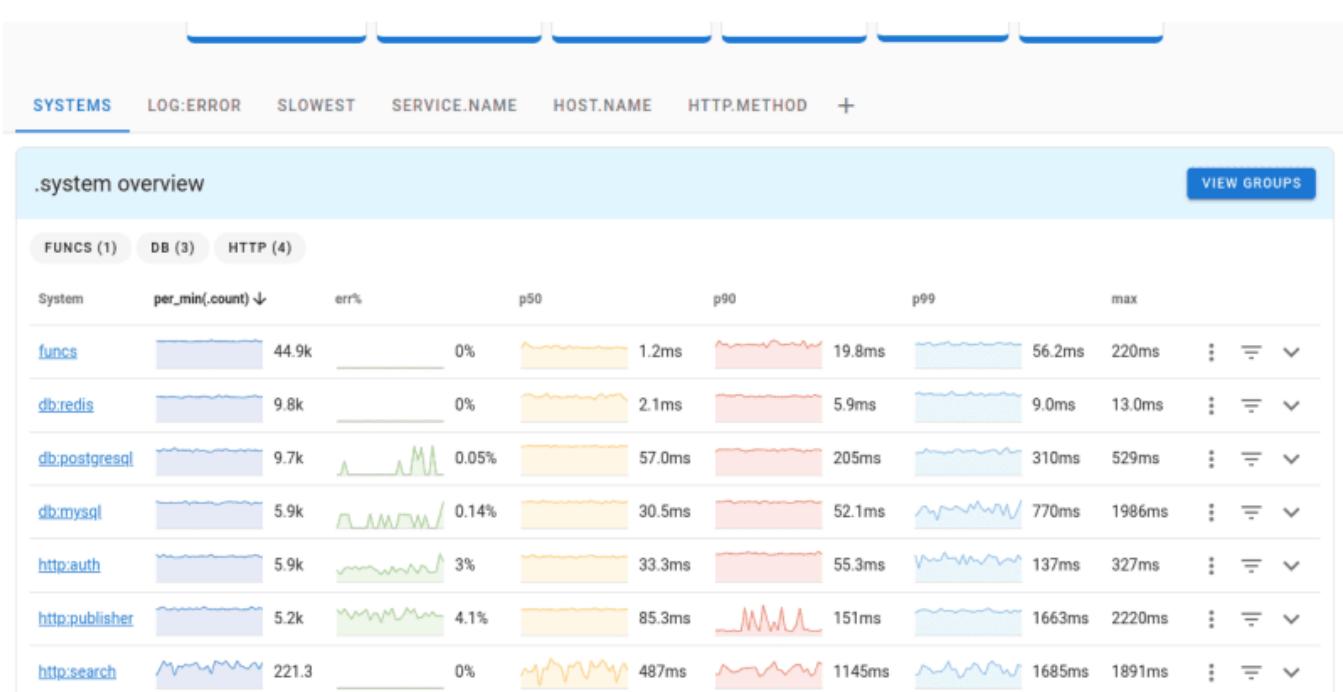


 Ashish Pandey in Excited Developers

## Java Multithreading—All you need to know

The only blog that you will ever need. Let's get started

13 min read · Jun 1



## Monitoring Spring Boot with OpenTelemetry

Integrating OpenTelemetry with Spring Boot allows you to capture distributed traces and other telemetry data from your application...

4 min read · Aug 2



The image shows the cover of a book or article titled "JAVA interview: TOP MOST ASKED QUESTIONS 2023". The title is in large, bold, yellow and white letters. Below the title is a subtitle "TOP MOST ASKED QUESTIONS 2023". At the bottom of the cover, there are three company logos: Infosys (blue text), Wipro (colorful circular logo), and Tata Consultancy Services (TATA logo with text). The background of the cover is light gray.

Syed Habib Ullah

## Top Most Asked Java Interview Questions at Accenture, Infosys, Capgemini, Wipro, Cognizant...

As Java is one of the most popular programming languages used in software development today, it's no surprise that Java skill is highly...

9 min read · Apr 1

30 2

+ ...

See more recommendations