

Software Design And Architecture

Syed Ghazi Raza

FA22-BSE-082

Problem Scenario:

A ride-hailing application needs to assign rides to thousands of drivers in real-time, but the central server struggles with high traffic, causing delays and inefficiencies.

Solution (Using Software Design and Architecture):

To address this, the **Microservices Architecture** can be applied:

1. Divide the System into Microservices:

- Split the application into independent microservices like Ride Matching, Payment Processing, Driver Management, and Notification Services.

2. Load Balancing:

- Deploy a load balancer to distribute traffic evenly among instances of microservices.

3. Event-Driven Architecture:

- Use message brokers (e.g., Kafka or RabbitMQ) to decouple services and manage real-time ride requests.

4. Scalability:

- Each microservice can scale independently. For instance, during peak hours, scale the Ride Matching microservice dynamically.

5. Caching:

- Use caching mechanisms (e.g., Redis) to store frequently accessed data like available drivers to reduce server load.

6. Fault Tolerance:

- Implement fallback strategies and retries to handle failures in individual microservices without affecting the entire system.

Outcome:

This architecture ensures the system remains responsive, scalable, and efficient, even during traffic spikes.

Software Design And Architecture

5 more problem scenarios

1. E-Commerce Scalability Issue

An e-commerce platform experiences performance bottlenecks during flash sales, where thousands of users simultaneously add items to their carts, causing delays in inventory updates and payment processing.

2. University Management System

A university management system struggles with handling real-time course registrations, leading to conflicts in seat allotments for popular courses during peak times.

3. Smart Traffic Management

A smart traffic management system designed to reduce congestion in a city cannot process real-time data from thousands of IoT devices (like cameras and sensors), leading to outdated traffic signals.

4. Online Video Streaming

An online video streaming platform faces buffering and latency issues during live events with millions of viewers accessing the same content simultaneously.

5. Banking System Security

A banking system is vulnerable to fraudulent transactions because its current system cannot verify the origin of requests or detect patterns in real time, leading to data breaches.