



ARCHITECTURAL CHALLENGES AND SOLUTIONS

Hassan Ali Mashwani

FA22-BSE-067

PREPARED BY

TO
(Sir) Mukhtiar Zamin

December 31, 2024

Project AutoTrack - Fleet Management System

Issue:

Delayed updates of vehicle locations, leading to inaccurate tracking and inefficient route optimization.

Architecture Flaw:

- Used a single-threaded process to handle location updates from all vehicles, creating a bottleneck.
- Relied on a relational database that struggled with high-frequency writes.

Solution:

- **Migrated to an Event-Driven Architecture:** Used a message queue (e.g., Kafka) to handle high-frequency location updates asynchronously.
 - **Switched to NoSQL Database:** Used a time-series database (e.g., InfluxDB) optimized for high-write scenarios.
 - **Implemented Real-Time Processing:** Used stream processing tools (e.g., Apache Flink) for real-time analytics and route optimization.
-

Project StreamFlex - Video Streaming Platform

Issue:

High latency in video playback during peak hours, causing buffering and user dissatisfaction.

Architecture Flaw:

- Centralized content delivery architecture resulted in high server load.
- No mechanism for adaptive streaming, leading to inefficiencies in bandwidth usage.

Solution:

- **Adopted a Content Delivery Network (CDN):** Distributed video content to edge servers closer to users, reducing latency.
 - **Implemented Adaptive Bitrate Streaming (ABR):** Allowed the system to adjust video quality based on the user's bandwidth in real time.
 - **Deployed Load Balancers:** Balanced traffic across servers to prevent overloads.
-

Project HealthSync - Telemedicine Platform

Issue:

Poor video call quality and frequent disconnections during consultations.

Architecture Flaw:

- Relied on a single region data center for all video call processing, causing latency for users in distant regions.
- Insufficient monitoring and auto-scaling capabilities to handle spikes in demand.

Solution:

- **Moved to Multi-Region Deployment:** Deployed video processing servers in multiple regions to reduce latency.
 - **Integrated WebRTC for Video Calls:** Improved call quality with peer-to-peer connections.
 - **Enabled Auto-Scaling:** Configured cloud resources to scale dynamically during peak demand.
-

Project EduPro - Online Learning Platform

Issue:

The system frequently crashed during live classes due to high concurrent user load.

Architecture Flaw:

- The monolithic application hosted all services (e.g., video streaming, chat, file sharing) together, leading to resource contention.

Solution:

- **Transitioned to Microservices Architecture:** Separated services like video streaming, chat, and file sharing into independent microservices.
 - **Added Circuit Breakers:** Prevented cascading failures during spikes by isolating failing services.
 - **Implemented Horizontal Scaling:** Allowed the platform to scale out by adding more server instances during high traffic.
-

Project SafePay - Digital Wallet System

Issue:

Fraudulent transactions occurred frequently, and transaction processing was slow.

Architecture Flaw:

- No real-time fraud detection mechanism.
- Centralized database caused delays in transaction verification.

Solution:

- **Integrated Fraud Detection System:** Implemented a machine learning-based fraud detection system to analyze transactions in real time.
 - **Adopted Event Sourcing:** Logged all transaction events in an event store for real-time processing and auditability.
 - **Decentralized Database Architecture:** Used regional databases to reduce latency and ensure high availability.
-