

**NAME: SAAD HUMAYUN**

**REG No: FA22-BSE-012**

**Title:** Architectural Problems and Solutions in Software Systems

## **Major Architectural Problems in Software Systems**

### **1. Transition from Monolithic to Microservices**

**Problem:** Traditional monolithic architectures consist of a single codebase handling all functionalities of a system. While this approach simplifies development in the early stages, it introduces significant challenges as the system grows. Scaling becomes difficult, deployment of updates impacts the entire system, and tightly coupled components lead to decreased development agility.

**Solution:** Transitioning to a microservices architecture addresses these challenges by breaking the system into smaller, independently deployable services. Each service is responsible for a specific functionality and communicates via APIs. For example, Netflix adopted a microservices approach to achieve better scalability and maintainability, enabling it to handle its vast global audience effectively.

### **2. Scalability Issues in Distributed Systems**

**Problem:** Distributed systems often encounter performance bottlenecks as user traffic increases, leading to issues such as slow response times and downtime. These problems arise due to uneven load distribution, inadequate resource allocation, and poor fault tolerance mechanisms.

**Solution:** Employ horizontal scaling to add more servers to the system, implement load balancers to distribute traffic evenly, and use distributed databases with sharding to handle large datasets. For instance, Twitter improved its backend by implementing data sharding and load balancing, enabling it to process billions of tweets daily.

### **3. Overuse of Shared Resources in Multi-Tenant Systems:**

**Problem:** Multi-tenant systems often suffer from contention for shared resources like databases or APIs, resulting in slow response times and potential system failures.

**Solution:** Implement resource isolation techniques such as per-tenant databases or rate limiting to ensure fair resource allocation. Shopify uses resource isolation to prevent high-traffic tenants from impacting the performance of others.

#### 4. Database Bottlenecks:

**Problem:** Centralized databases can become a bottleneck under high read/write loads, leading to slow query responses and system latency.

**Solution:** Introduce caching mechanisms such as Redis or Memcached to store frequently accessed data in memory. Additionally, database replication or sharding can distribute the load effectively. Facebook addressed database bottlenecks by using extensive caching layers with MySQL.

#### 5. Difficulty in Adopting Modern Technologies for Legacy Systems

**Problem:** Legacy systems are often built using outdated technologies, making it difficult to integrate modern tools or frameworks. This incompatibility can hinder scalability and innovation.

**Solution:** Use middleware or API layers to act as bridges between the legacy system and modern technologies. Walmart modernized its e-commerce platform by introducing APIs that allowed seamless integration with newer technologies.

---