Microservices Architecture - Quick Interview Guide

Client Layer

- **React Web App** + **Mobile PWA** → Frontend applications
- **JWT Authentication** → Secure token-based auth

API Gateway (Spring Cloud Gateway)

- **Entry point** for all client requests
- **Global CORS** → Cross-origin request handling
- **Rate Limiter** → Controls request frequency (Redis-backed)
- **Circuit Breaker** → Prevents cascade failures

Service Discovery

- **Eureka** → Service registry, tracks all microservices
- **Config Server** → Centralized configuration management

Business Microservices

- **Auth-service** → Login, registration, JWT validation
- **User-service** → User profiles, account management
- **Product-service** → Product catalog, inventory
- **Order-service** → Order processing, transactions
- **Recommender-service** → Personalized recommendations

Data Layer

- **PostgreSQL** → Primary database for each service
- **Redis** → Caching + rate limiting storage

Monitoring Stack

- **Prometheus** → Metrics collection
- **Grafana** → Dashboards and visualization
- **Zipkin/OTEL** → Distributed tracing

Key Interview Points

✅ **Independent scaling** of each service
✅ **Fault isolation** with circuit breakers
✅ **Security** through JWT + rate limiting
✅ **Observability** with full monitoring stack
✅ **Configuration management** centralized
✅ **Database per service** pattern

Interview Tip: This shows a **production-ready** microservices architecture with proper security, monitoring, and resilience patterns.