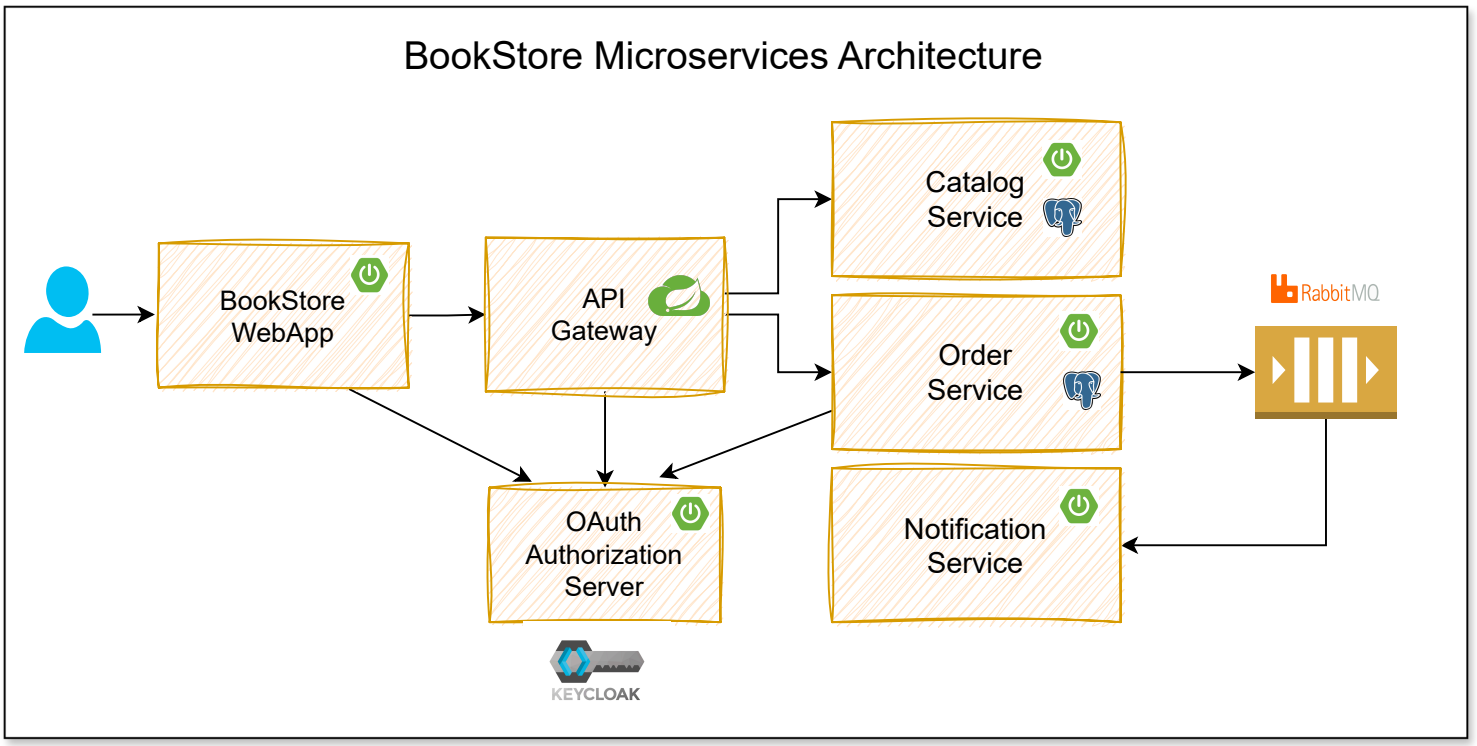


What is Microservices Architecture?

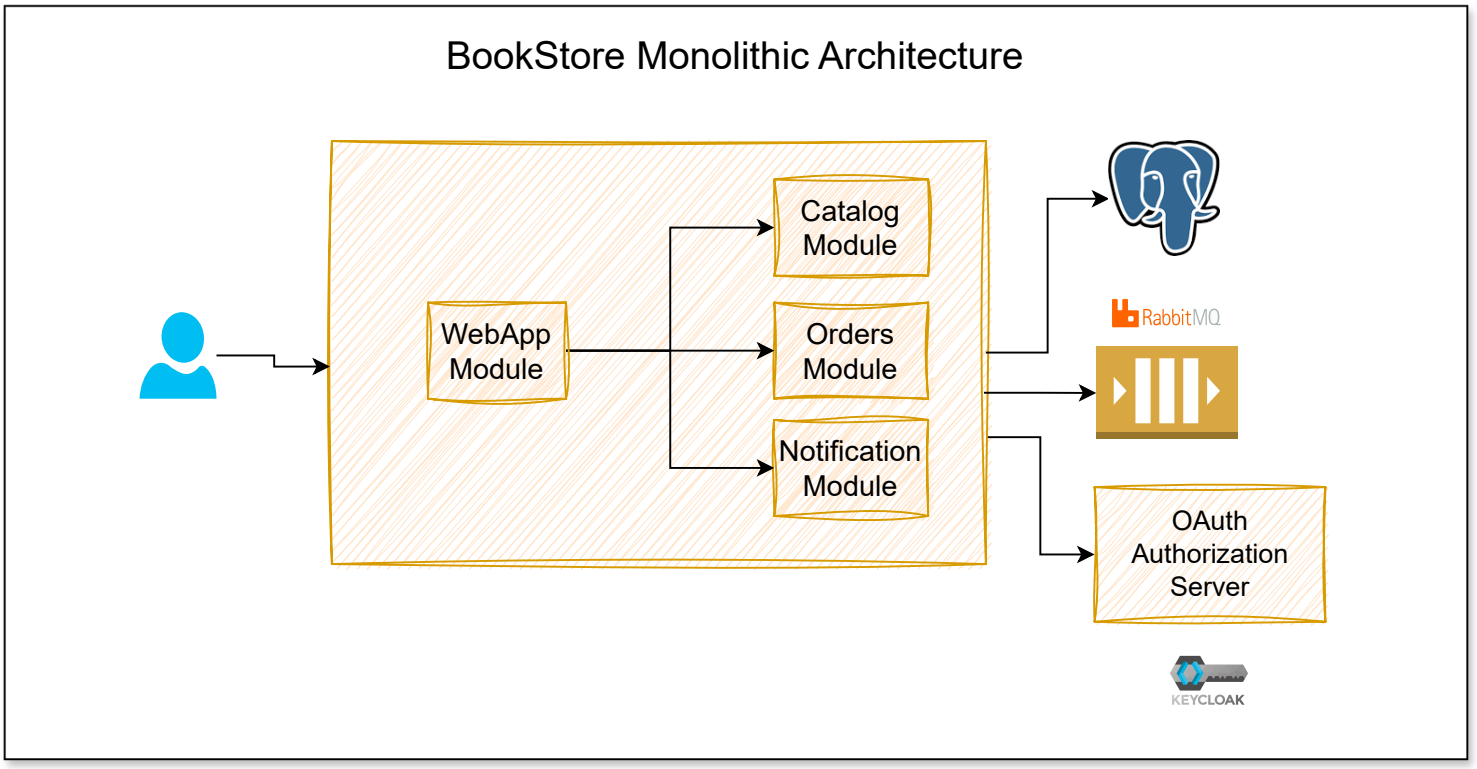
Microservices is an architectural style where large applications are composed of small services that provide specific business capabilities that can be deployed and managed independently.

Learning Objectives

- Building Spring Boot REST APIs
- Database Persistence using Spring Data JPA, Postgres, Flyway
- Event Driven Async Communication using RabbitMQ
- Implementing OAuth2-based Security using Spring Security and Keycloak
- Implementing API Gateway using Spring Cloud Gateway
- Implementing Resiliency using Resilience4j
- Job Scheduling with ShedLock-based distributed Locking
- Using RestClient, Declarative HTTP Interfaces to invoke other APIs
- Creating Aggregated Swagger Documentation at API Gateway
- Local Development Setup using Docker and Testcontainers
- Testing using JUnit 5, RestAssured, Testcontainers, Awaitility, WireMock

Additional Topics(Membership)

- Monitoring & Observability using Grafana, Prometheus, Loki, Tempo
- Kubernetes 101 course
- Deployment to Kubernetes



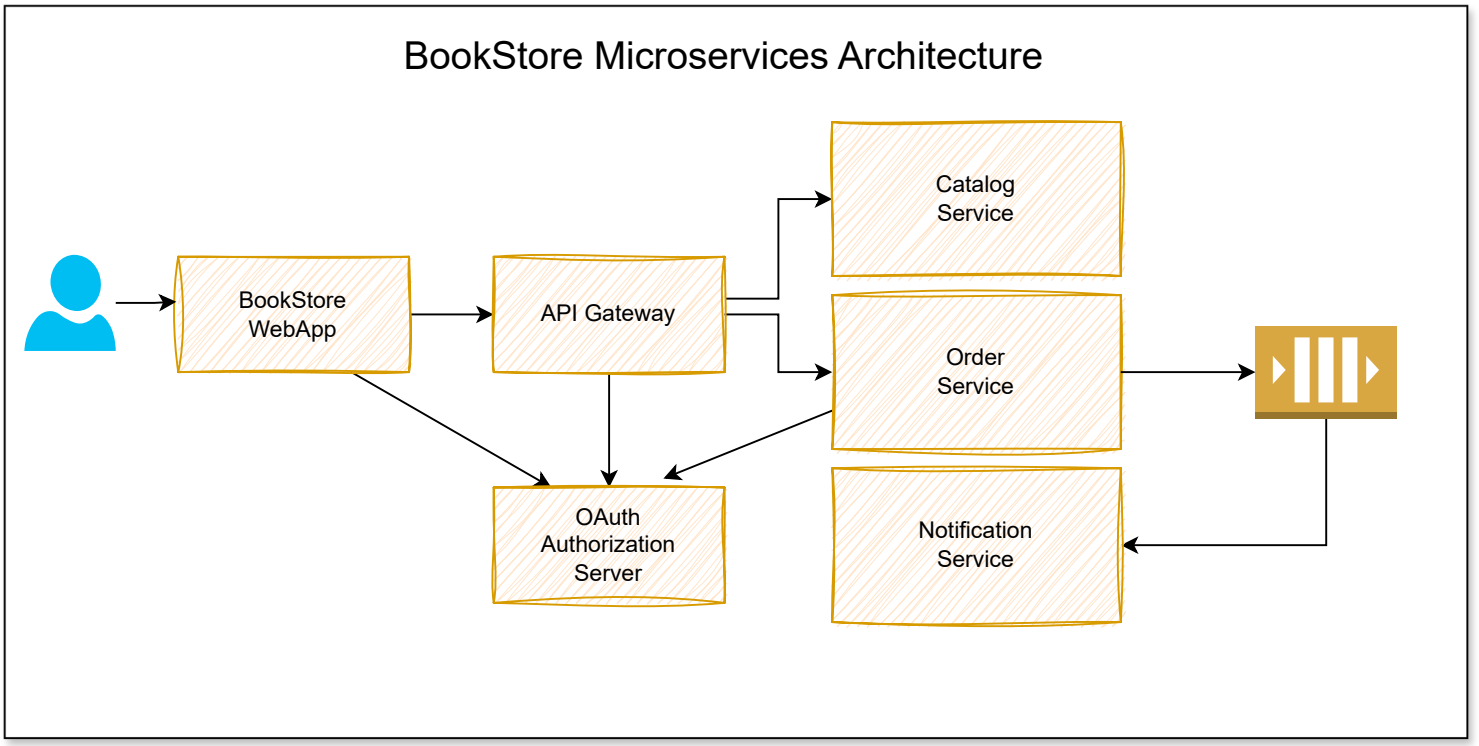
Monolithic Architecture

Pros

1. Simpler Development
2. Easier Testing & Debugging
3. Simpler Deployment

Cons

1. Difficult to scale sub-systems(modules)
2. Difficult to adopt new technologies
3. Higher chance to become big ball of mud



Microservices Architecture

Pros

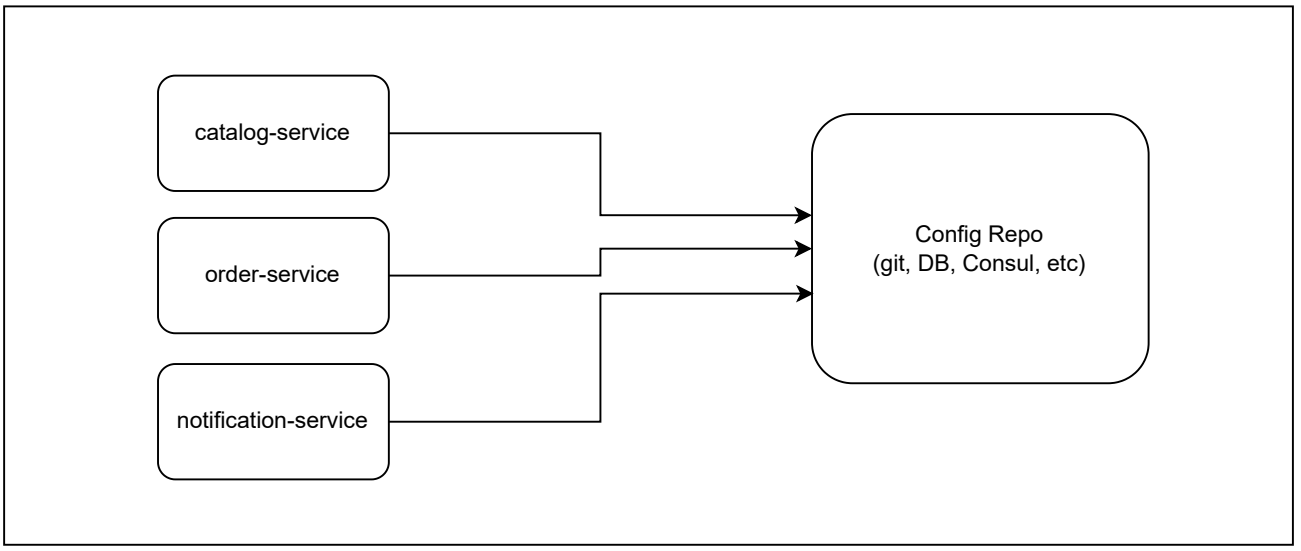
1. Can scale individual services
2. Smaller codebases easy to reason about
3. Easy to adopt newer technologies if needed
4. Less dependency on other team deliverables

Cons

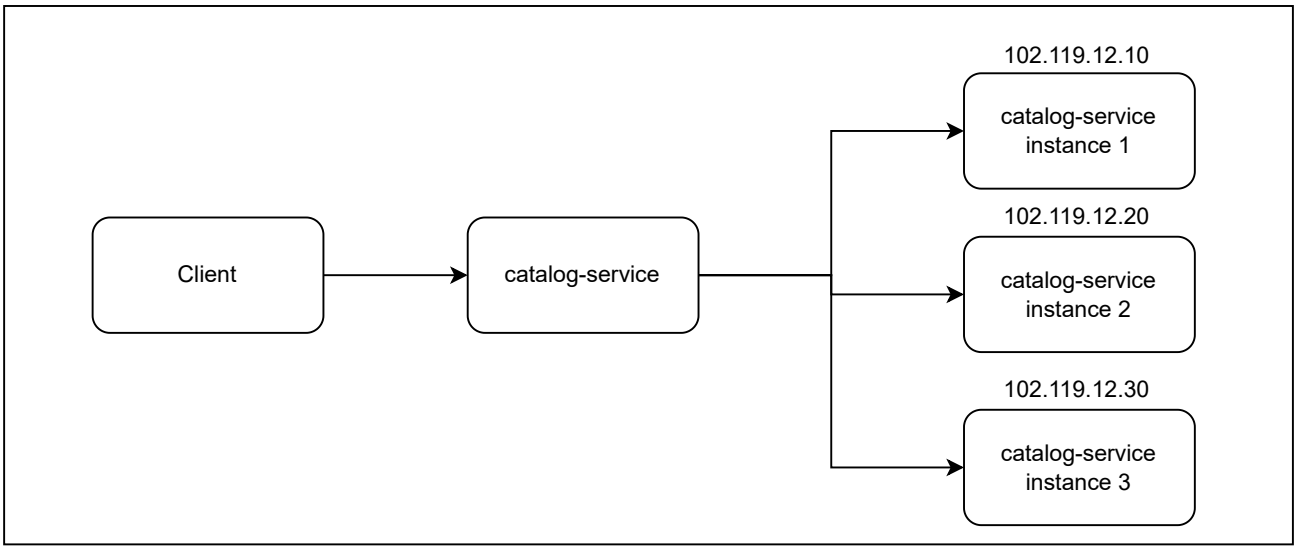
1. Difficult to build & manage distributed systems
2. Difficult to test & debug
3. Complex deployment process
4. Performance Issues

Spring Cloud vs Kubernetes

Spring Cloud Config Server



Service Registry (Eureka, Consul)



Kubernetes