



Microservice Architecture

Key Drivers & Barriers In Adopting Microservices



Agenda

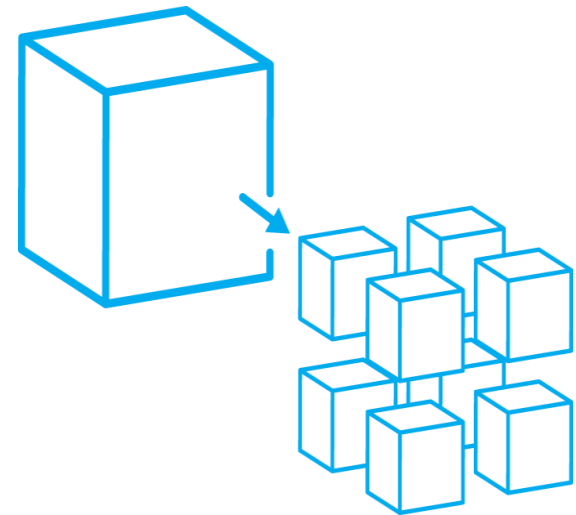
- What are Microservices ?
- Characteristics of Microservices
- Key Drivers & Barriers in adopting Microservices
- OES Microservice Architecture (Current State)
- OES Microservice Implementation (Case Study)
- Limitations of OES Microservice Architecture (Current State)
- OES Microservice Architecture (Future State)
- Q & A



What are Microservices ?

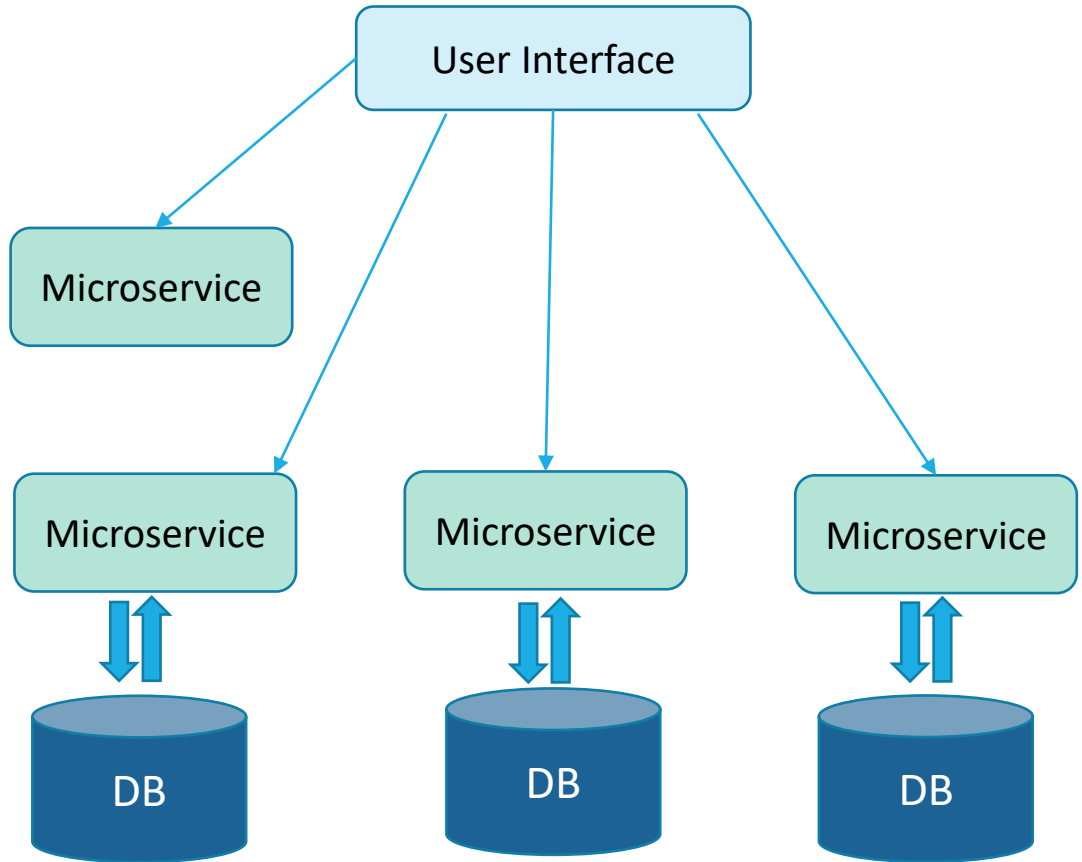
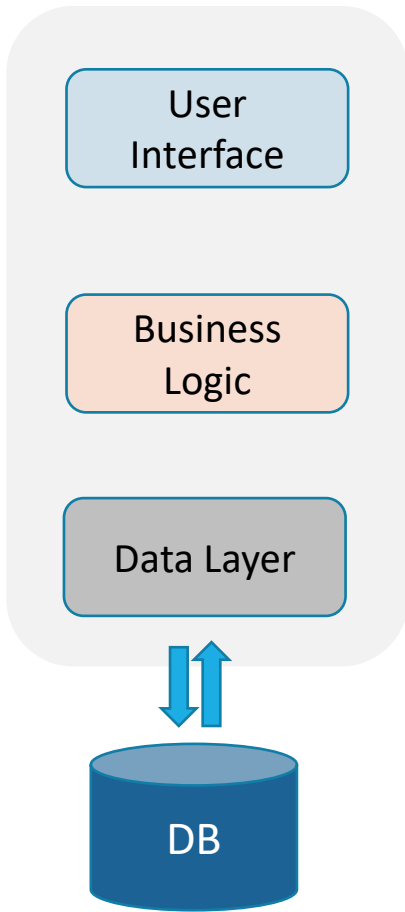
Microservices (aka microservice architecture) is an architectural style that structures an application as a collection of services that are:

- Independently deployable
- Loosely coupled
- Organized around business capabilities

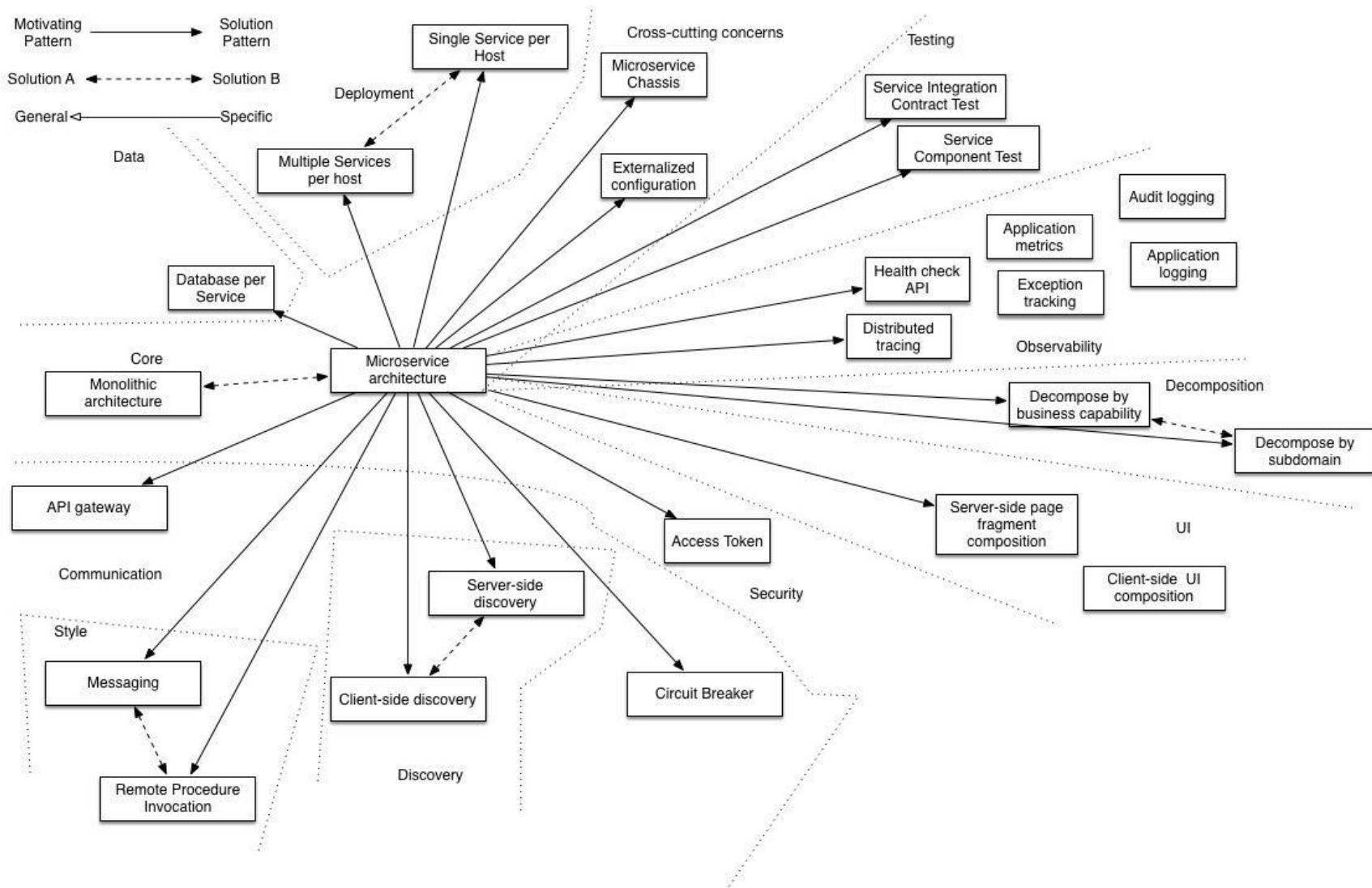


Monolithic Architecture

Microservices Architecture



Microservice Characteristics



Key Drivers & Barriers In Adopting Microservices

Key Drivers for Adopting Microservices :

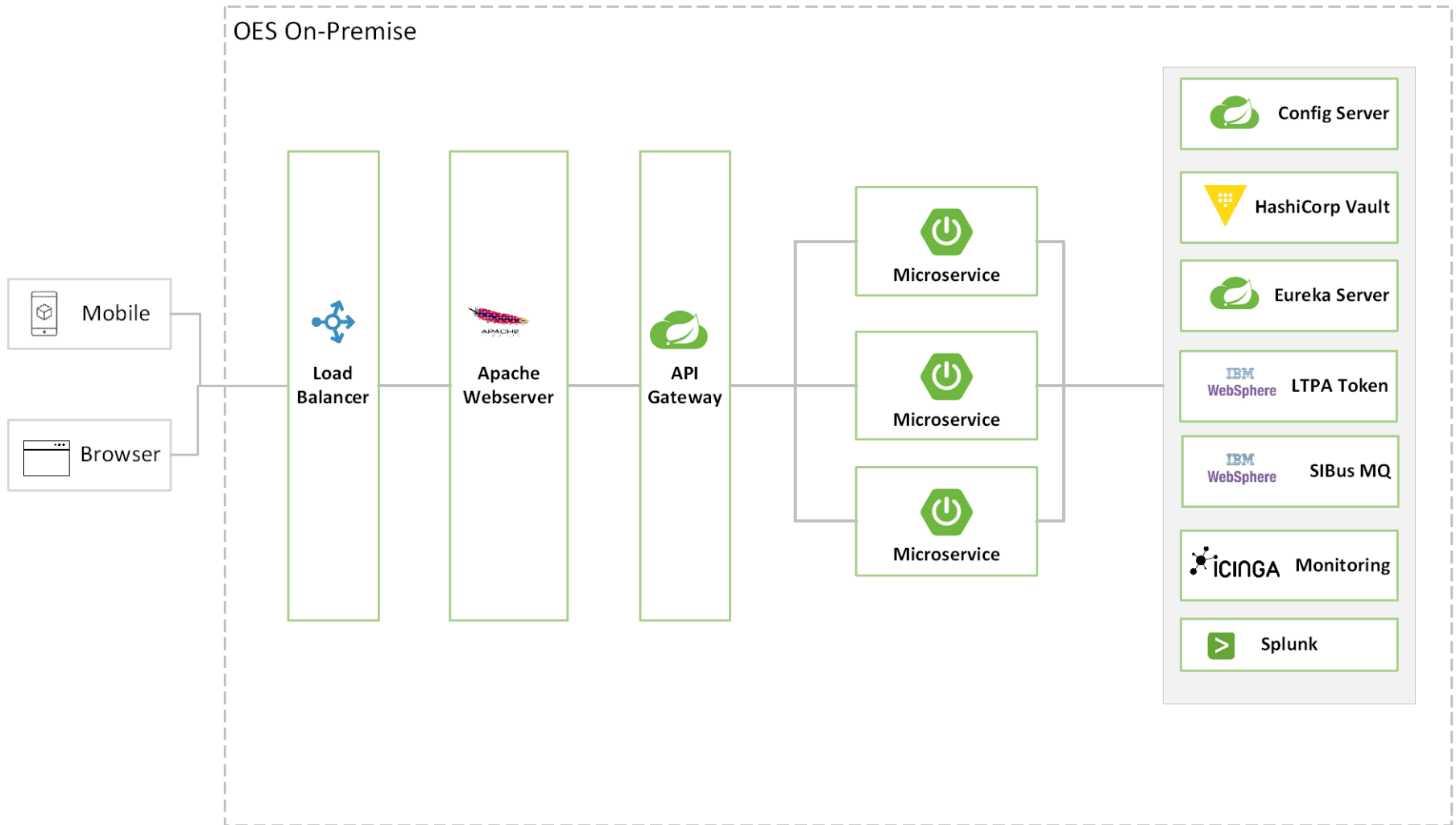
- Faster and simpler deployments
- High Reliability
- High Availability & Scalability
- Design autonomy

Barriers to Adopting Microservices:

- Cost of Migration & ROI
- Legacy Systems Integration
- Cultural Shift
- Complexity
- Operational Overhead



OES Microservice Architecture (Current State)

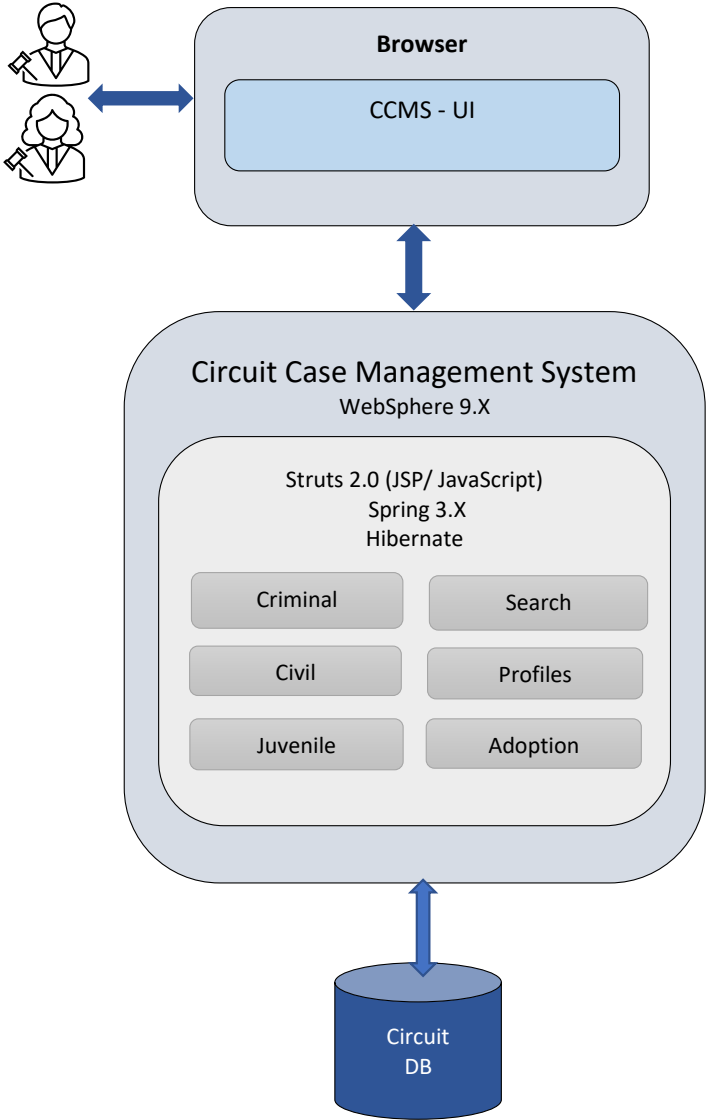


OES Microservice Platform (Current State)

- IBM Open Liberty Server
- Open JDK 11/17 (IBM Semeru Runtimes)
- Spring Cloud
 - ❖ Eureka Service Discovery
 - ❖ Spring Cloud API Gateway
 - ❖ Spring Cloud Config Service
 - ❖ Spring Boot
 - ❖ Spring Actuator
 - ❖ Spring Sleuth \ Micrometer
- Hashicorp's Vault (Secrets / Sensitive Data)
- Jenkins CI/CD
- SonarQube (Code Quality & Static Code Analysis)
- JFrog Xray (Vulnerability Scanning)
- Splunk (Distributed Logging & Analytics)



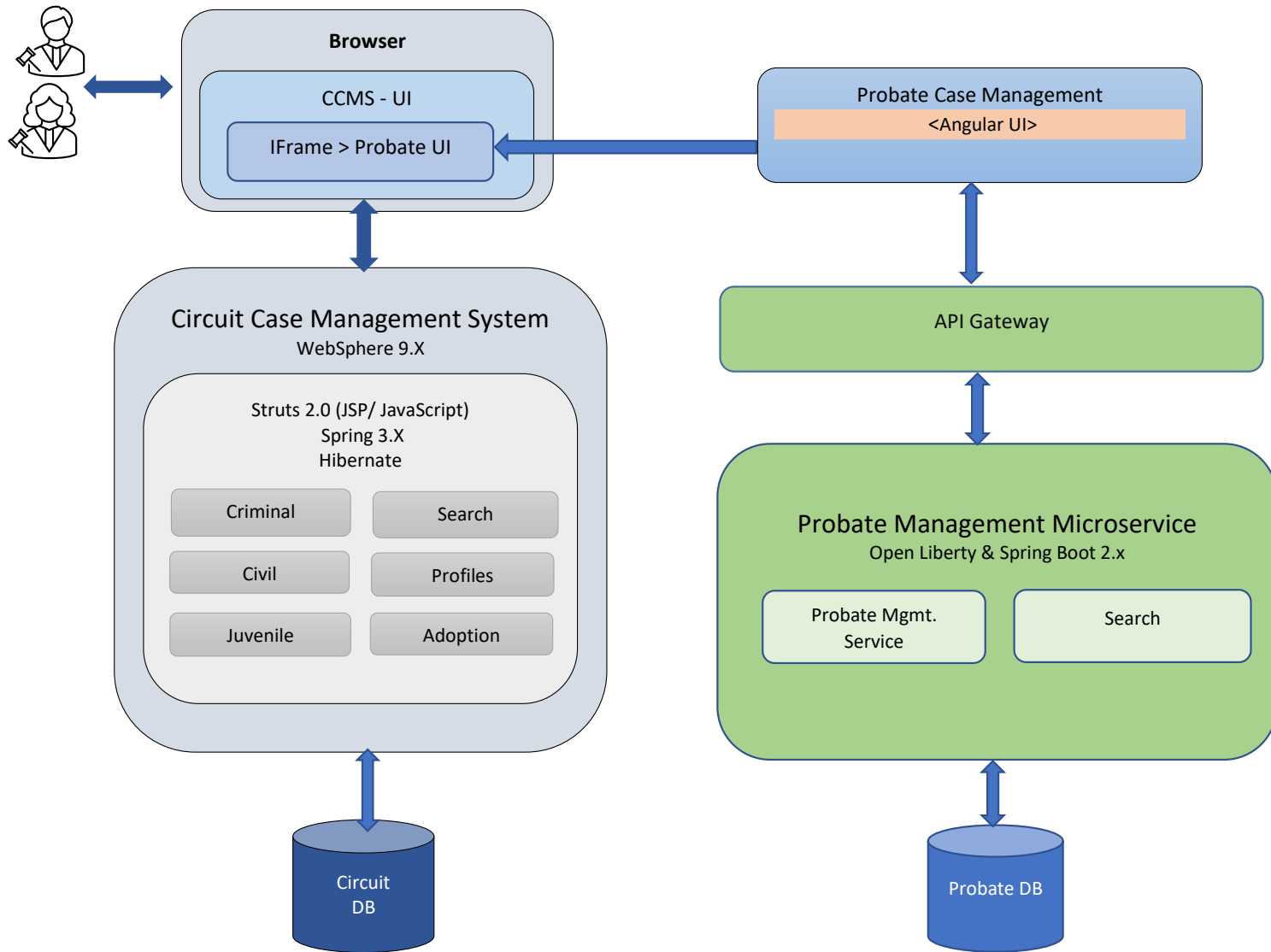
OES Microservice Adoption (Case Study)



Monolithic Architecture



OES Microservice Adoption (Case Study contd..)



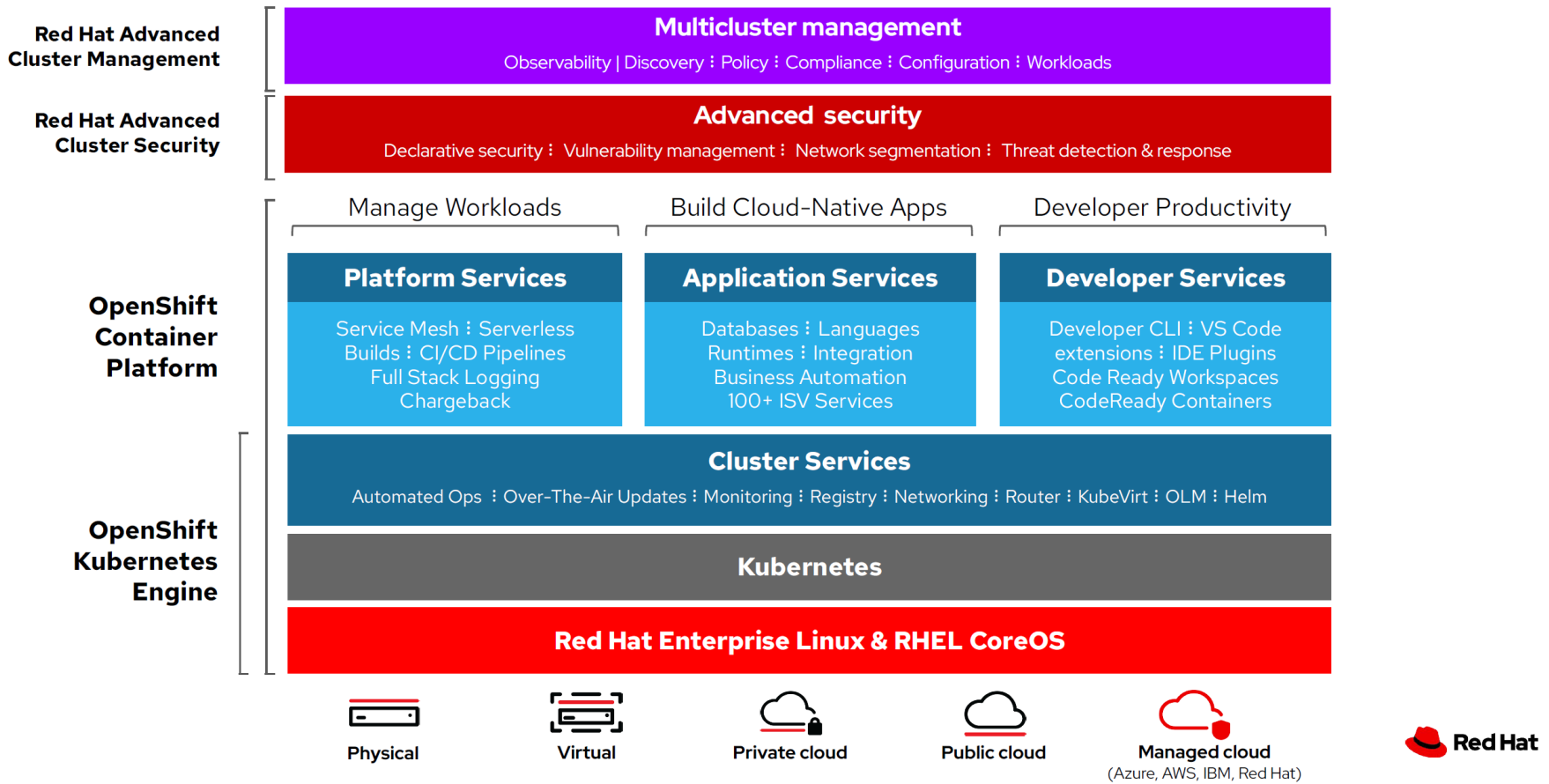
Limitations of Our Current MS Architecture

- **Isolation:** Hosting multiple microservices per host or VM can lead to potential dependency clashes
- **Dependency Management:** Managing dependencies between microservices becomes more manual and error-prone.
- **Deployment Complexity & Consistency:** Increased complexity of build & deployment pipelines.
- **Scaling Challenges:** Manual & less dynamic scaling
- **Resource Utilization:** Increased overhead and less efficient resource utilization

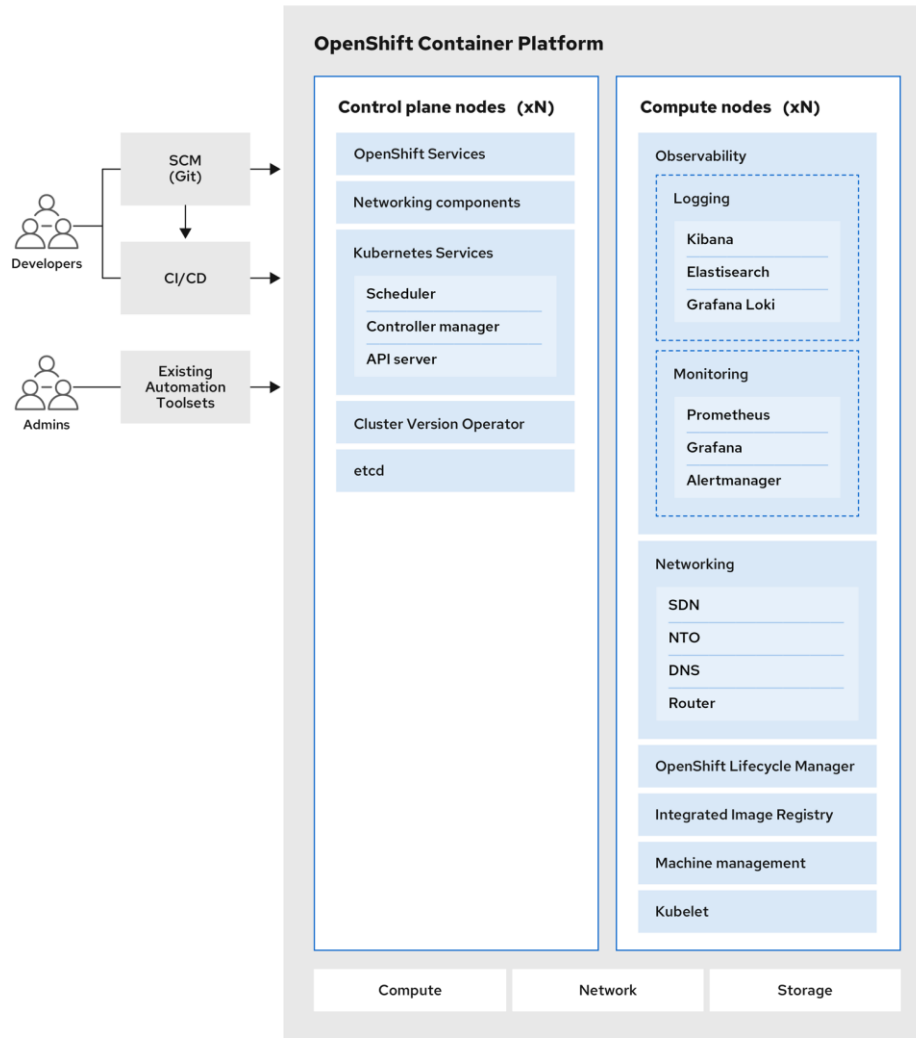


OES Microservice Architecture (Future State)

OpenShift Container Platform



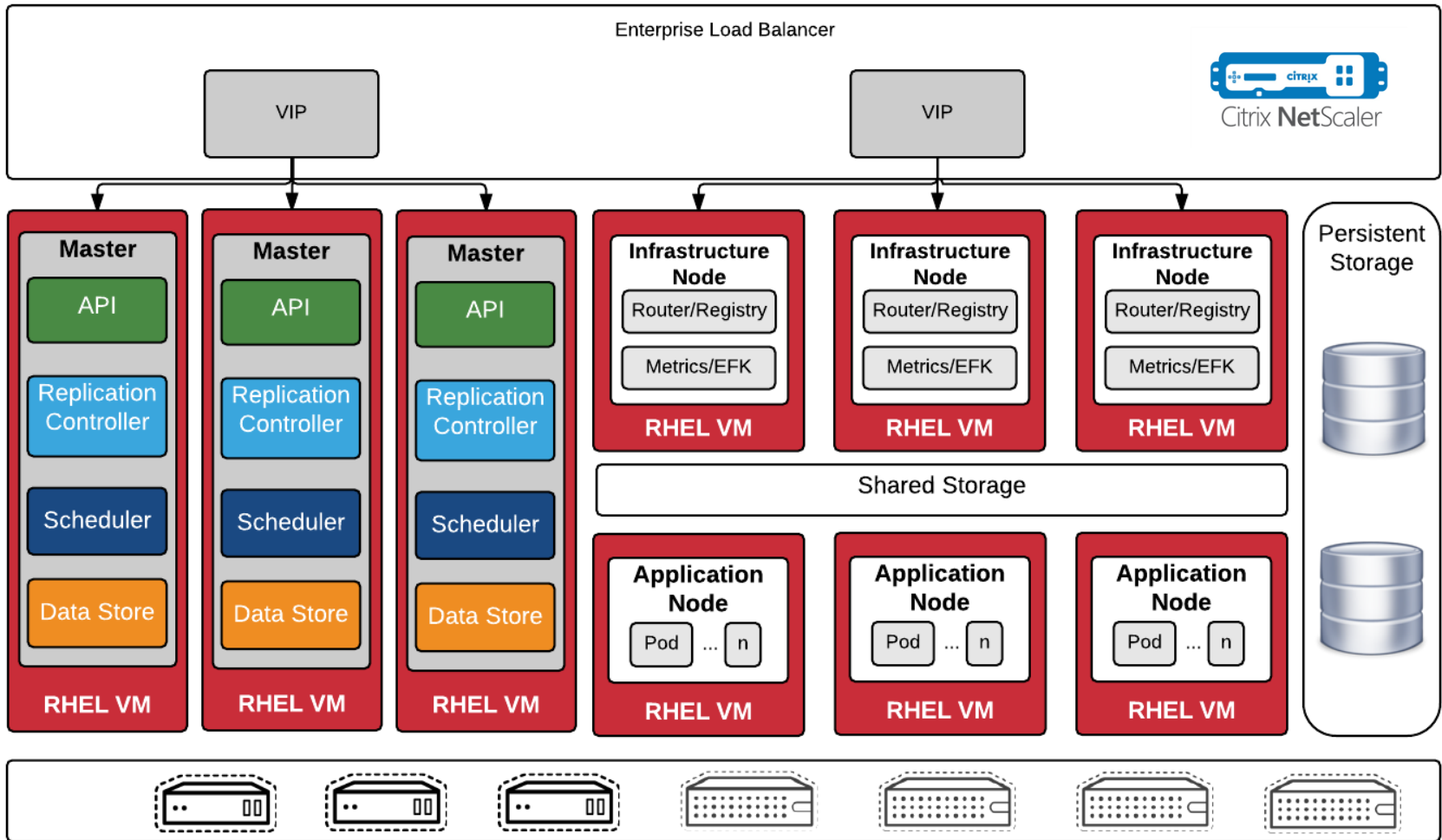
OES Microservice Architecture (Future State)



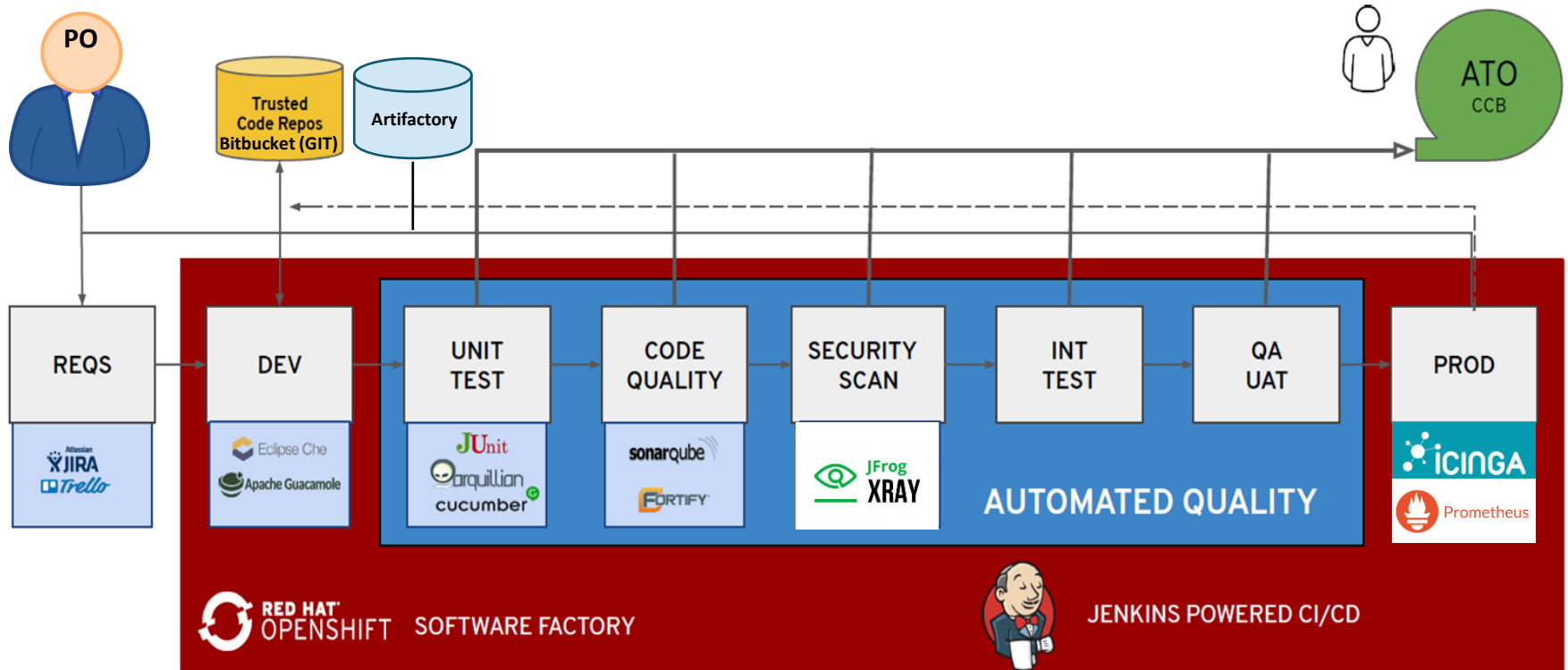
277_OpenShift_1022



OES Microservice Architecture (Future State)



OES DevSecOps (Future State)



Q & A

