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
Microservice Characteristics

Motivating Pattern → Solution Pattern
Solution A ← Solution B

General Specific

Data

Multiple Services per host

Single Service per Host

Microservice Chassis

Cross-cutting concerns

Deployment

Externalized configuration

Microservice architecture

Testing

Service Component Test

Service integration Contract Test

Audit logging

Application metrics

Exception tracking

Application logging

Health check API

Distributed tracing

Observability

Decompose by business capability

Decomposition

Decompose by subdomain

UI

Server-side page fragment composition

Client-side UI composition

Access Token

Server-side discovery

Client-side discovery

Circuit Breaker

Remote Procedure Invocation

Messaging

Discovery

Communication

Style
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

Circuit Case Management System
WebSphere 9.X

Struts 2.0 (JSP/JavaScript)
Spring 3.X
Hibernate

Criminal
Search

Civil
Profiles

Juvenile
Adoption

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

- Red Hat Advanced Cluster Management
- Red Hat Advanced Cluster Security

OpenShift Container Platform

- Multicluster management
  - Observability
  - Discovery
  - Policy
  - Compliance
  - Configuration
  - Workloads

Advanced security
- Declarative security
- Vulnerability management
- Network segmentation
- Threat detection & response

Platform Services
- Service Mesh
- Serverless
- Builds
- CI/CD Pipelines
- Full Stack Logging
- Chargeback

Application Services
- Databases
- Languages
- Runtimes
- Integration
- Business Automation
- 100+ ISV Services

Developer Services
- Developer CLI
- VS Code extensions
- IDE Plugins
- Code Ready Workspaces
- CodeReady Containers

Cluster Services
- Automated Ops
- Over-The-Air Updates
- Monitoring
- Registry
- Networking
- Router
- KubeVirt
- OLM
- Helm

Kubernetes

Red Hat Enterprise Linux & RHEL CoreOS

Physical
Virtual
Private cloud
Public cloud
Managed cloud (Azure, AWS, IBM, Red Hat)
OES Microservice Architecture (Future State)
OES Microservice Architecture (Future State)
OES DevSecOps (Future State)
Q & A