Course Outline

Systems and Service Architecture - Microservices Architecture

Course Summary

Description

Building complex enterprise applications is very challenging in an ever-changing environment.

Changes in the integration between services or systems will trigger changes in the design of the system. This course covers how to design maintainable, fit to the size and purpose of your organization service and systems architecture and practical infrastructure.

This course covers common problems with building the landscape of distributed apps, starting from integration problem, infrastructure, and modern approaches, to solving complexity issues.

This course explains the pros and cons of each solution (e.g. traditional SOA vs Microservice architecture), available tools (mostly open-source) supporting the development and maintenance, etc..

Topics

- Quick overview of traditional approaches
- Design concepts and tradeoff
- Generic SOA
- Microservices and implementation of SOA - concepts
- Microservice protocols and tools
- Cloud and Auto Scalability
- Real problems

Audience

This course is designed for those wanting to learn how to design maintainable, fit to the size and purpose of your organization service and systems architecture and practical infrastructure.

Prerequisites

Before taking this course, a basic understanding of software and system engineering required. A good understanding of system development and integration is recommended. Some exposure to problems encountered during building complex enterprise solutions is recommended.

Duration

Two days

Course Outline

Systems and Service Architecture - Microservices Architecture

Course Outline

I. Quick overview of traditional approaches

A. Monolithic systems
B. System of Systems
C. Early service orientation (EAI, CORBA, etc...)
D. Early webservices (SOAP, etc...)
E. Service Oriented Architecture (SOA)
F. Microservices

II. Design concepts and tradeoff

A. Flexibility and Complexity tradeoff
B. Cohesion, Coupling
C. Hidden dependencies vs explicit dependencies
D. Small system vs Big System
E. Module/Component vs Service

III. Generic SOA

A. Benefit and Costs of SOA
B. Successes and failures in implementing SOA
C. Messaging and ESB
D. Infrastructure and tools supporting SOA

IV. Microservices and implementation of SOA - concepts

A. Do one thing and do it well
B. Microservice vs Service
C. DevOps
D. Continuous Deployment and Delivery
E. Lightweight protocols

V. Microservice protocols and tools

A. HTTP, JMS, AMQP, Websockets, JSON, etc...
B. Deployment
    1. Containers (Docker, K8N, LXC, etc...)
    2. Configuration Management (Ansible, etc...)
C. Monitoring and Management
D. Infrastructure

VI. Cloud and Auto Scalability

A. Microservice redundancy and fail-over
B. Performance scalability
C. Auto scalability
D. Implement: OpenStack, AWS, etc.

VII. Real problems

A. Complexity of ecosystem
B. Network Performance
C. Security
D. Deployment
E. Testing
F. Nano-services