Mildain Solutions

Microservices architecture and design workshop

Audience: technology leaders, architects and experienced developers who want to learn the microservice architecture. The architecture and design exercises are on paper and are independent of any particular technology stack.

Duration: 3 days, private on-site

By the end of the course, you will have learnt about:

- Why Microservices?
- Deconstructing your existing system and rec into a more efficient platform.
- Creating Microservices using messaging
- Creating Microservices using events
- using Event Sourcing to handle complex State
- Using Process Managers for complex transactional event systems
- Service patterns: Gateways, Adapters, Pipelines, Choreography
- Continuous Delivery
- Applying practical Domain Driven Design (DDD) and Command Query Responsibility Segregation (CQRS)

What you'll learn, and how you can apply it

The essential characteristics of the microservice architecture, its benefits and drawbacks, and when to use it

- Distributed data management patterns
- Effective microservice testing strategies
- Deployment options for microservices
- Strategies for refactoring a monolithic application to a microservice architecture

And you'll be able to:

- Architect an application as a set of microservices

- Use sagas to maintain data consistency
- Implement queries that span services
- Test microservices
- Refactor a monolith to services

Who should attend

This course is aimed at Developers, Admins and Software Architects who wish to gain a strong understanding of Microservices architecture before embarking on a development and deployment project.

Prerequisites

Delegates should have some understanding of systems architecture and design. Experience of administering Linux or Windows servers may also be useful. Some basic editing of code my occur during the course, so experience with a language such as Java, C#, C++ would be advantageous.

Course syllabus

Day 1

- Understanding change in a system
- Designing for change in a system
- APIs vs Microservices
- Designing a delivery pipeline for Microservices

Day 2

- RPCish Microservices and why they hurt you
- Composing Microservices using Messaging
  - Brokers and Point to Point
  - Schemas
  - Identify and defining Message Protocols
  - Messaging State Machines

- Composing Microservices using Events
    - Event Driven Architectures
    - Patterns for System Choreography
    - Stream Processing to materialize views of the system
    - Upgrading of views
- Event Sourcing
    - The Aggregate Root
    - System Commands
    - Understanding System State as an audit log
    - Combining with Stream Processing
    - Snapshotting

Day 3

- Transactions using Process Managers
- Event based testing techniques
- Cloud Deployment
- Containerization for deve process and deployment