Course duration

- 2 days

Course Benefits

- Understand and work with monoliths.
- Learn to design, develop, and integrate microservices.
- Learn common design patterns.

Course Outline

1. Breaking Up Monoliths – Pros and Cons
    1. Traditional Monolithic Applications and Their Place
    2. Disadvantages of Monoliths
    3. Developer's Woes
    4. Architecture Modernization
    5. Architecture Modernization Challenges
    6. Microservices Architecture is Not a Silver Bullet!
    7. What May Help?
    8. In-Class Discussion
    9. Summary
2. Microservices
    1. What is a "Microservice"?
    2. Unix Analogy
    3. Principles of Microservices
    4. Services within an SOA vs Microservices
    5. Properties and Attributes of Microservices
    6. Benefits of Using Microservices
    7. The Two-Pizza Teams
    8. Beware of Microservices Cons
    9. Anti-Pattern: Nanoservices
    10. The Twelve-Factor App Methodology
    11. The Select Factors
    12. Serverless Computing
    13. Microservices – Operational Aspects
    14. Summary
3. Microservices Architecture Defined
    1. The Microservices Architecture
    2. SOA Promises and Expectations
    3. Microservices Architecture vs SOA
    4. The ESB Connection

Phoca PDF

   5. Microservices Architecture Benefits
   6. Microservices Architecture Choices and Attributes
   7. Example: On-Line Banking Solution Based on MsA
   8. Distributed Computing Challenges
   9. Replaceable Component Architecture
   10. The Actor Model
   11. MapReduce Distributed Computing Framework
   12. Hadoop's MapReduce Word Count Job Example
   13. What Can Make a Microservices Architecture Brittle?
   14. 4+1 Architectural View Model
   15. Summary

4. Containerization Systems for Microservices
   1. Infrastructure as Code
   2. Why Not Just Deploy My Code Manually?
   3. What is Docker
   4. Docker Containers vs Traditional Virtualization
   5. Docker is a Platform-as-a-Service
   6. Docker Integration
   7. Docker Services
   8. Docker Application Container Public Repository
   9. Container Registries
   10. Your Own Docker Image Registry
   11. Starting, Inspecting, and Stopping Docker Containers
   12. One Process per Container
   13. The Dockerfile
   14. Kubernetes
   15. What is OpenShift
   16. Summary

5. Commonly Used Patterns
   1. Why Use Patterns?
   2. Performance-Related Patterns
   3. More Performance-Related Patterns
   4. Pagination vs. Infinite Scrolling - UX Lazy Loading
   5. Integration Patterns
   6. More Integration Patterns
   7. The Service Mesh Integration Pattern
   8. Mesh Pros and Cons
   9. Service-to-Service Communication with Mesh
   10. Resilience-Related Patterns
   11. Summary

6. API Management
   1. API Management Defined
   2. The Traditional Point-to-point Integration Example
   3. It Raises Some Questions …
   4. The Facade Design Pattern
   5. API Management Conceptual Diagram
   6. Complimentary Services for Microservices

7. What Else is Needed?
8. The Driving Forces
9. API Management Offerings
10. The Mashery API Management System Overview
11. AWS API Gateway Call Flow
12. Summary

7. Designing and Implementing Microservices
    1. Two Types of IT Projects
    2. What is In Scope for a Robust Microservices Design?
    3. Scoping Your Microservice via the Bounded Context
    4. Scoping Your Solution's Microservices Architecture
    5. External / Shared and Internal Service Models
    6. General Architectural and Software Process Organizational Principles
    7. Loose Coupling, the OOD Perspective
    8. Crossing Process Boundary is Expensive!
    9. Cross Cutting Concerns
    10. More Cross Cutting Concerns
    11. To Centralize or Decentralize Client Access?
    12. Decentralized Client Access
    13. Centralized Client Access
    14. The Facade Pattern
    15. The Facade Service Conceptual Diagram
    16. The Naked Objects Architectural Pattern
    17. When to Use Naked Objects Pattern
    18. Dealing with the State
    19. How Can I Maintain State?
    20. Micro Front-ends (a.k.a. MicroUI)
    21. How can MicroUI Help Me?
    22. Your Clients Are Diverse
    23. The "Rich Client" - "Thin Server" Paradigm
    24. The "Rich Client" - "Thin Server" Architecture
    25. RIA as a Driving Force to Turn the "Thin Server" into a Set of Microservices
    26. Design for Failure
    27. Managing Failures Effectively
    28. The Immutable Infrastructure Principle
    29. Implementing Microservices
    30. JAX-RS
    31. Microservice-Oriented Application Frameworks and Platforms
    32. Embedding Databases
    33. Embedded Java Databases
    34. Summary
8. Microservices Integration
    1. One Common Observation
    2. The “One Service - One Host” Deployment
    3. Things to Consider when Integrating
    4. Technology Options
    5. The Data Exchange Interoperability Options

Phoca PDF

6. The Correlation ID
7. Enterprise Integration Patterns
8. Asynchronous Communication
9. Benefits of Message-Oriented Middleware (MOM)
10. Asynchronous Communication Models
11. Message Brokers
12. A Message Broker Diagram
13. Asynchronous Message Consumption Patterns
14. Popular Messaging Systems
15. Challenges of Managing Microservices
16. Options for Managing Microservices
17. In-Class Discussion
18. Summary

9. Working with Data in Microservices
    1. Monolithic Databases
    2. The Traditional Two-phase Commit (2PC) Protocol
    3. Table Sharding and Partitioning
    4. The CAP Theorem
    5. Mechanisms to Guarantee a Single CAP Property
    6. The CAP Triangle
    7. Eventual Consistency
    8. Handling Transactions in Microservices Architecture
    9. The Event-Driven Data Sharing Diagram
    10. The Saga Pattern
    11. The Saga Log and Execution Coordinator
    12. The Saga Happy Path
    13. A Saga Compensatory Request Example
    14. In-Class Discussion
    15. The Need for Micro Databases
    16. Migrating Data from Existing Databases (Breaking up the Monolith Database)
    17. One Data Migration Approach
    18. One Data Migration Approach (Cont'd)
    19. In-Class Discussion
    20. Command Query Responsibility Segregation (CQRS)
    21. The CQRS Communications Diagram
    22. A Word of Caution
    23. The Event Sourcing Pattern
    24. Event Sourcing Example
    25. Applying Efficiencies to Event Sourcing
    26. Summary
10. Robust Microservices
    1. What Can Make a Microservices Architecture Brittle?
    2. Making it Resilient – Mechanisms
    3. Techniques and Patterns for Making Your Microservices Robust
    4. Fail Fast or Quiesce?
    5. Synchronous Communication Timeouts / Retries
    6. Asynchronous Communication Timeouts / Retries

Phoca PDF

7. In-Class Discussion
8. The Circuit Breaker Pattern
9. The Circuit Breaker Pattern Diagram
10. The Bulkhead Pattern
11. Factor IX of the 12 App Methodology
12. Feature Enablement
13. Designing for Test and Failure
14. Making Microservices Testable
15. Test for Failure
16. Continuous Testing and Integration
17. Continuous Release and Deployment
18. SLAs
19. Where and What to Monitor
20. Logging and Monitoring
21. Summary

Class Materials

Each student will receive a comprehensive set of materials, including course notes and all the class examples.

Class Prerequisites

Experience in the following *is required* for this Microservices class:

- Foundational knowledge of programming and software design principles.