trainitek

Implementing Modern Architecture - Part 2

This training builds on the popular “Implementing Modern Architecture” training by going deeper into key topics and covering those that couldn’t fit into the first training. Over 3 days, participants will acquire tools applicable to their daily work as architects.

We will cover important subjects such as Architecture Modernization, Serverless architecture, versioning and evolution of synchronous and asynchronous APIs, and best practices in documenting your architecture beyond traditional diagrams. The course will also demonstrate how to approach testing complex systems, storing data effectively, and shaping an environment that enables teams to quickly and safely ship code to production. Additionally, we will explore how AI can assist architects in their daily tasks.

Through a series of tasks and discussions, participants will learn how to apply the tools, strategies, and methodologies presented during the training to the products they work on.

Audience: Architects, Tech Leads, Senior Developers

Duration: 3 days

- Day 1: Recap of the previous training, System Architecture, Communication, Documenting your Architecture
- Day 2: Storing Data, Coding your Architecture, Testing complex systems
- Day 3: Continuous Delivery for real systems, Deployment strategies, Database migrations, Leveraging AI in an Architect’s work

Format: 20% workshop / 80% lecture

Training program

1. Recap: Architecture, Architects, and Domain-Driven Design

2. System Architecture
    a. Quick Recap: Monolith, Modular Monolith, and Microservices
    b. Serverless Computing
    c. Strategies for Software Modernization

3. Communication
    a. Introduction to API Design
    b. Versioning of Sync and Async APIs
    c. Producer-side errors

4. Documenting your Architecture
    a. Documentation beyond C4

   b. Auto-generated documentation for APIs

5. Storing Data
   a. Types of Data Stores
   b. Applicability of different Data Stores
   c. Choosing a proper Data Store

6. Coding your Architecture
   a. Different strategies of Code Sharing
   b. Effective Code Reviews
   c. Pair Programming
   d. Mob Programming

7. Testing complex systems
   a. Types of Tests
   b. Shapes of Tests (Pyramid, Honeycomb, etc.)
   c. Contract Testing
   d. Acceptance Testing

8. Continuous Delivery for real systems
   a. Introduction to Continuous Delivery
   b. DORA metrics
   c. Branching strategies
   d. Pipelines best practices
   e. Versioning artifacts
   f. Inspecting artifacts
   g. Managing environments

9. Deployment strategies
   a. Zero downtime deployments
   b. Canary deployments
   c. Blue-green deployments

10. Database migrations

11. Leveraging AI in an Architect's work