trainitek

Implementing Modern Architecture

To be a successful architect, one needs to understand the pros and cons of different architectures and under what conditions they are applied in a project. In addition, a good knowledge of the development process and production support is also very valuable.

In this training, we will go on a journey starting with the definition of architecture and continuing with Domain-Driven Design, System Architecture, coding it, and finishing with running our Architecture in production. Participants will also learn about the pitfalls of misapplying architectural styles to certain problems. We'll also cover topics like Microservices, CQRS, Hexagonal Architecture, Event Sourcing, System stability, Development quality, and more.

Audience: Architects, Team Leads, Senior Developers
Duration: 3 days

- Day 1: Architecture and Architects, Domain-Driven Design, System Architecture, Documenting your Architecture
- Day 2: Communication, REST API, Coding your Architecture
- Day 3: Tactical DDD, Command Query Responsibility Segregation, Event Sourcing, Stability of your System, Securing your System, Monitoring your System

Format: 20% workshop / 80% lecture

Training program

1. Architecture and Architects
    a. What is “architecture”?
    b. What is good architecture?
    c. Who is a good architect?
    d. Architectural drivers

2. Domain-Driven Design
    a. Why use DDD?
    b. How to discover a Bounded Context?

3. System Architecture
    a. Monolith
    b. Microservices
    c. Modular Monolith
    d. How to make a choice?
    e. A path from Monolith to Microservices

4. Documenting your Architecture
    a. Sharing main decisions
    b. Visualizing architecture

5. Communication
    a. Synchronous and Asynchronous communication
    b. Commands vs Events
    c. Big and Small Events
    d. Message Naming
    e. Event Choreography and Orchestration
    f. Message Concurrency
    g. Message Processing Order
    h. Dealing with Errors
    i. Idempotent Consumers
    j. Outbox and Inbox

6. REST API
    a. General principles
    b. Task-based REST API
    c. Dealing with Errors: Problem Details

7. Coding your Architecture
    a. Project structure
    b. Applying Hexagonal Architecture principles
    c. Package structure
    d. Code quality automation

8. Tactical Domain-Driven Design
    a. Designing Aggregates and Value Objects
    b. Ensuring Invariants in Domain Model

9. Command Query Responsibility Segregation

10. Event Sourcing: advantages and challenges

11. Stability of your System
    a. Common failures and stability antipatterns
    b. Stability patterns

12. Securing your System

13. Monitoring your System
    a. Aspects of monitoring
    b. Metrics to expose