trainitek

Implementing Modern Architecture

To be a successful architect, one needs to know 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 project support in production is also very valuable.
In this course, participants will learn about the pitfalls of misapplying an architecture style and applying it to the wrong problem. 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

- 1 day: Architecture and Architects, System Architecture, Communication
- 2 day: Domain Driven Design, Coding your architecture, REST API
- 3 day: Command Query Responsibility Segregation, Event Sourcing, Documenting your Architecture, Continuous Integration and Continuous Deployment, Stability of your System, Securing your System, Monitoring your System

Format: 40% workshop / 60% lecture

Training program

1. Architecture and Architects
    a. What is "architecture"?
    b. What is good architecture?
    c. Who is a good architect?
2. System Architecture
    a. Monolith
    b. Microservices
    c. Modular Monolith
    d. How to make a choice?
    e. A path from Monolith to Microservices
3. Communication
    a. Synchronous and Asynchronous communication
    b. Commands vs Events
    c. Big and Small Events
    d. Message Naming
    e. Event Versioning
    f. Messaging in Monolith
    g. Event Choreography and Orchestration
    h. Message Concurrency
    i. Message Processing Order
    j. Dealing with Errors

    k. Idempotent Consumers
    l. Two-phase Commit for sending Messages?
4. Domain Driven Design
    a. Why use DDD?
    b. How to discover a Bounded Context?
5. Coding your Architecture
    a. Project structure
    b. Package structure
    c. Designing Aggregates and Value Objects
    d. Choosing ID types (UUID, Long, etc.)
    e. Ensuring Invariants in Domain Model
    f. Separate Behavior and Persistence
    g. Do Exceptions help?
    h. Applying Hexagonal Architecture principles
    i. Code quality automation
    j. Code reviews
6. REST API
    a. General principles
    b. Task-based REST API
    c. Dealing with Errors: Problem Details
7. Command Query Responsibility Segregation
8. Event Sourcing: advantages and challenges
9. Documenting your Architecture
    a. Sharing main decisions
    b. Visualizing architecture
10. Continuous Integration and Continuous Deployment
    a. Versioning
    b. Automation
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
    c. Control in production
    d. Distributed tracing