BO·TT·EGA
IT minds

Training program:

Application and Systems Architecture – Architectural Patterns for application architects

Info:

Name:	**Application and Systems Architecture – Architectural Patterns for application architects**
Code:	**Arch-patterns**
Category:	Architectural patterns
Target audience:	architects
Duration:	3 days
Format:	50% lecture / 50% workshop

This training presents selected Design and Architectural Patterns in a practical and non-textbook approach, set in the context of designing web applications, platforms, systems, and frameworks. It includes practical examples drawn from real-world systems such as ERP, visual tools, distributed systems, and servers.

Participants will gain integrated knowledge of modern software engineering achievements, enabling them to create advanced systems. The discussed topics underpin modern frameworks and technologies, enhancing understanding and facilitating conscious usage. Techniques for combining patterns into higher-order structures are also presented.

The training is intended for designers and architects seeking to expand their competencies in standard architectural styles.

The program serves as a general framework; its full implementation exceeds 3 days. During the pre-training analysis, we select patterns beneficial for the team and focus on them during the training. Typically, we can address about 70% of the listed patterns within 3 days.

It's all about the content.

- Modern architectures (CQRS – supporting DDD)
- Possibility of basing examples on Kafka
- Matching solution class to problem class

BO·TT·EGA
IT minds

Training program

1. Documenting Architecture in the 4C Approach - Context, Containers, Components, Classes

1.1. Context

1.1.1. The context in which the system will operate

1.1.2. Main actors

1.1.3. Organization-level processes involving the system

1.1.4. Role of other systems

1.2. Containers

1.2.1. Technical containers in which the system will be deployed

1.2.2. Integration patterns

1.2.3. Communication protocols

1.3. Components

1.3.1. Components and Services in the SOA context

1.3.2. Canonical data exchanged between Services

1.3.3. API design

1.3.4. Single source of truth principle

1.4. Classes

1.4.1. Introduction to design techniques and patterns at the class level

2. System Architecture - Overview of Approaches

2.1. Modularization

2.1.1. Levels of separation

2.1.2. Techniques for integrating modules

2.2. Micro Kernel Architecture

2.3. Command + Command Handler

BO·TT·EGA
IT minds

BO·TT·EGA
IT minds

3.3.1. plugins

3.3.2. listeners

3.3.3. injecting extensions

3.3.4. aspect-oriented approach

4. Data access layer patterns

4.1. DAO

4.2. Repository

4.3. Active Record

4.4. Object-relational Mapping - using Java Persistence API (Hibernate)

4.5. ORM Issues

4.5.1. Paradigm mismatch

4.5.2. Object boundary mapping

4.5.3. Composition or aggregation - mapping

4.5.4. 3 approaches to optimistic locking

4.5.5. Versioning

4.5.6. Locking objects that serve as bases for others (READ)

4.5.7. Ensuring consistency of object graphs via root locking (WRITE)

5. Modularization

5.1. Separation of Bounded Contexts

5.1.1. Techniques for defining module boundaries

5.1.2. Techniques for module encapsulation

5.1.2.1. API

5.1.2.2. DTO

5.2. Module integration - approaches

5.2.1. Events

BO·TT·EGA
IT minds

8.6.1. Specification by Example - business goals

8.6.2. Flow - User Story

8.6.3. Automating UI interaction - Agents as abstraction over test scripts