BO·TT·EGA
IT minds

Training program:

Modern architecture of web applications - Microservices, REST, noSQL (Java/.NET)

Info:

Name:	**Modern architecture of web applications - Microservices, REST, noSQL (Java/.NET)**
Code:	**arch-ms-workshop-modern**
Category:	Microservices workshop
Target audience:	architects developers
Duration:	3 days
Format:	50% lecture / 50% workshop

The training is intended for designers and architects, who want to find their way in a world of new technologies for creating light, and at the same time ultra-efficient and scalable web applications.

It's all about the content.

- Modern architectures (CqRS – supporting the DDD)
- Technology integration and a holistic approach
- Choosing the class of solution for the class of a problem

BOTTEGA IT minds

Training program

1. REST

- 1.1. A good understanding of important aspects of http
- 1.2. The REST API philosophy
  - 1.2.1. Standards
  - 1.2.2. Best practices – selection for a context
- 1.3. Correct interpretation of SOA
  - 1.3.1. Canonical or domain model
- 1.4. Goal: a well-designed rest covers both web application and mobile technologies at the same time

2. Microservices and CqRS architecture

- 2.1. Assumptions
  - 2.1.1. Designing with failures in mind
  - 2.1.2. Evolutionary approach
  - 2.1.3. Decentralization of data management
  - 2.1.4. Endpoint and Pipe
- 2.2. Strategies for refactoring monolithic systems
- 2.3. How to determine a boundary of services
  - 2.3.1. Bounded Context with the DDD approach
  - 2.3.2. Anti-pattern: Nanoservice
- 2.4. Acceptance testing of services

3. System architecture

- 3.1. Integration of services
- 3.2. Command-query Responsibility Segregation
  - 3.2.1. Write stack – domain model

BOTTEGA IT minds

- 3.2.1.1. Building Blocks DDD
- 3.2.1.2. Unit testing of domain logic
- 3.2.1.3. Event Sourcing
- 3.2.2. Read stack – denormalization in order to optimize readings

4. Scalable distributed systems

- 4.1. Domain events
- 4.2. Event orchestration, Saga model
- 4.3. Queues
  - 4.3.1. Solution selection
  - 4.3.2. Optimization
  - 4.3.3. Patterns
    - 4.3.3.1. Event Broker
    - 4.3.3.2. Event Bus
- 4.4. Approach to the Eventual Consistency

5. noSQL

- 5.1. When it’s worth it and for what
- 5.2. MongoDB/RavenDB
- 5.3. CAP theorem in practice
- 5.4. Designing models for reading

6. Continuous Integration and Continuous Deployment

- 6.1. Enabling the CD from the perspective of architecture and code
- 6.2. Tools