Training program:

Microservices .NET Core - Part I: Application architecture

Info:

Name:	**Microservices .NET Core - Part I: Application architecture**
Code:	**NET-arch-ms1**
Category:	.NET Architecture
Target audience:	developers architects devops
Duration:	3 days
Format:	30% lecture / 70% ćwiczenia

After this workshop, you will get a vast understanding of building distributed applications using .NET Core. By following the good patterns and practices, covering the various aspects of event-driven approach, discussing in-depth topics such as clean architecture, asynchronous integration, internal communication, monitoring, distributed tracing, testing, deployment, containerization and orchestration, you shall find the practical solutions to the most common challenges when it comes to building the microservices.

Moreover, you will get to know how to make use of some of the most acknowledged open source tools being part of Cloud Native Computing Foundation in order to make your solution cloud-agnostic, meaning that it can run on-premise (bare metal) as well as in chosen cloud provider, avoiding any potential vendor lock-in.

Training program

1. Event Storming - discovering healthy boundaries

- 1.1. Process Level
  - 1.1.1. Discovering sub-domains
  - 1.1.2. Generic sub-domains based on business archetypes and analytical patterns
  - 1.1.3. Distilling Bounded Contexts
  - 1.1.4. Context Mapping
  - 1.1.5. Strategic integration
    - 1.1.5.1. Published Language
    - 1.1.5.2. Open Host
    - 1.1.5.3. Shared Kernel
    - 1.1.5.4. Anticorruption Layer
    - 1.1.5.5. Customer-Supplier
    - 1.1.5.6. Conformist
- 1.2. Tactical design
  - 1.2.1. Aggregates boundaries
    - 1.2.1.1. Antipatterns
  - 1.2.2. Policies

2. Foundations of microservices

- 2.1. The applicability of microservices
- 2.2. Architecture levels
- 2.3. Distributed monolith anti-pattern

3. DDD in practice

- 3.1. Aggregates and consistency boundaries

3.2. Repositories and application services

3.3. Domain events

4. Application architecture

4.1. Clean architecture

4.2. Implementation of a RESTful API

4.3. CQS and CQRS

4.4. Infrastructure layer

5. Event-drive architecture

5.1. Integration patterns

5.2. Message broker

5.3. Data consistency

6. Integration via events

6.1. Event subscription

6.2. Event publishing

7. Transactional messaging

7.1. Inbox pattern

7.2. Outbox pattern