

Course code: **ESB/F**

Course title: **Enterprise Service Bus illustrated by JBoss Fuse and Fabric8**

Days: 3

## Description:

### Course intended for:

The training is intended for Java programmers, system analysts and architects wishing to get familiar with the techniques of development of an ESB (Enterprise Service Bus).

### Course objective:

The training objectives include:

- Getting the participants familiar with the basic terms and standards, associated with system integration and ESB,
- Presentation of good and bad practices of integration, tools available on the market,
- Getting familiar with integration patterns and methods of their implementation in the ESB,
- Acquisition of practical skills associated with JBoss Fuse / Fabric8.

The training is provided in several variants –for programmers, for system analysts, architects. Depending on the training group, main emphasis is placed on theoretical or practical issues and implementation of integration solutions.

### Requirements:

The participants must have at least the basic Java programming skills, at least basic familiarity with WebService and the XML processing technology. Basic knowledge of Java EE is also recommended.

### Course parameters:

3\*8 hours (3\*7 net hours) of lectures and workshops (with a visible emphasis on workshops). During workshops, examples are implemented, which illustrate realization of the most popular integration patterns, using JBoss Fuse.

Group size: no more than 8-10 participants.

## Course curriculum:

### 1. Basic integration of enterprise applications

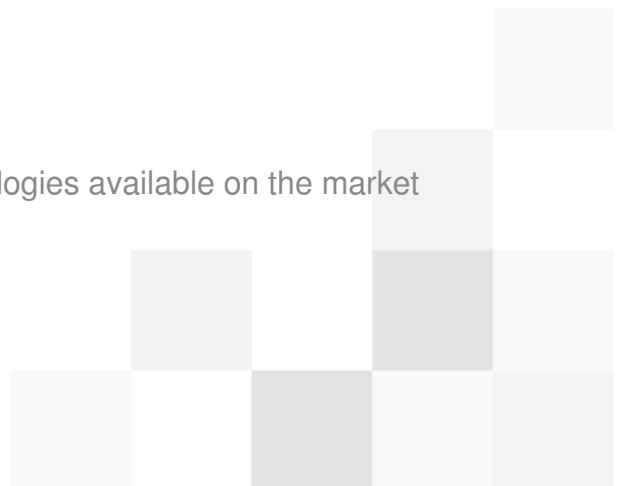
- Presentation of basic concepts and terms, associated with broadly understood integration: silos, SOA, EIP, MEP, EAI, ESB, service bus, data bus, ETL, EDA, CMD, MOM, integration adapter and broker, orchestration, synchronous/asynchronous/offline integration, JBI, SCA, WS-\* etc.
- Presentation of differences between the service bus, integration platform, integration broker, queue server etc.
- Pros and cons of implementation of an integration platform, the objective of such integration at a corporation, migration of existing systems to the "service bus"

### 2. Discussing of characteristics and functionalities provided by the ESB platform

- Location transparency
- Conversion of transport protocols
- Message transformation
- Message routing
- Message enhancement
- Service bus protection
- Monitoring and management
- Performance
- Interoperability
- Standardization

### 3. Review and comparison of tools and technologies available on the market

- Commercial and non-commercial



- Based on Java EE, .NET, other
- Mature and developing
- More and less popular
- According to other criteria

## 4. Discussing of JBoss Fuse platform

- Basic components and architecture
  - Apache Karaf (OSGi container – modularity, versioning, runtime)
  - Apache Camel (integration framework - routing, transformation, configuration)
  - Apache CXF (framework web service – SOAP and REST)
  - Apache A-MQ (JMS server)
  - Fuse Fabric (distributed deployments tool)
- Concepts and basic terminology for Fuse Fabric
  - Fabric, Fuse Application Bundle
  - Fabric Agent, Fabric Container, Fabric Profile
  - Fabric Ensemble, Fabric Server
  - Fabric Registry
- Concepts and basic terminology for Apache Camel
  - Routes
  - Endpoint
  - Camel Context
  - Camel Component
  - Message, Exchange
  - Processor



- DSLs: Java DSL, Spring DSL
  - The quality of services provided by JBoss Fuse (realization mechanisms: high accessibility, resistance to failures, load balancing, persistence, transactionability and security)
  - Similarities and differences in relation to other ESB platforms. Including reference and comparison to associated projects
    - JBoss Enterprise SOA Platform 5.x and JBoss ESB
    - JBoss Fuse Service Works
    - FuseSource ESB
    - Apache ServiceMix
  - Programming environment - Red Hat JBoss Developer Studio
  - Discussing of differences between the commercial version (JBoss Fuse) and the free version (Fabric8)
  - Monitoring and administration via Hawt.io Web console
5. Most frequently used integration patterns (EIP – Enterprise Integration Patterns) and their realization using JBoss Fuse platform
- Channel
  - Message
  - Services
  - Filter
  - Router
  - Transformer
  - Endpoint
  - Discussing of other popular patterns
6. Working with message management
- Message structure



- Message types and formats
- Transformation and conversion of messages
- Message validation
- Message persistence

## 7. Working with services

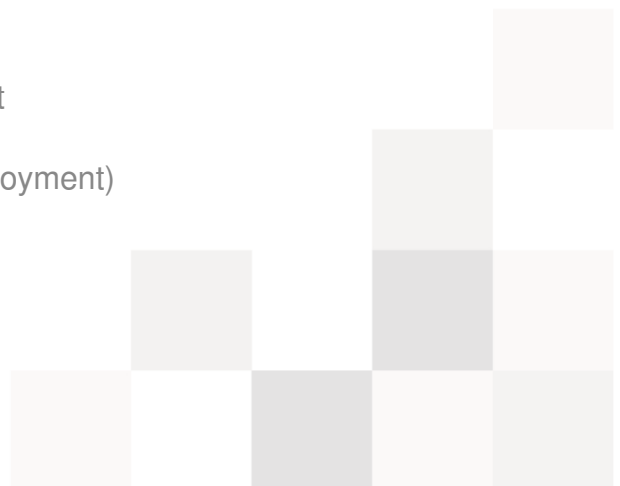
- Structure of service
- Service types
- Service contract
- Service configuration

## 8. Service arrangement and message routing on the bus

- Service registers and repositories
- Routing on ESB
- CBR - Content Based Routing
- Notifications

## 9. Service performance quality

- Service replication
- Clustering at the service and transport protocol level
- Repeating of messages
- Clustering at the level of services and transport protocols
- Repeating of messages
- Service monitoring and management
- Implementation of changes (hot deployment)
- Other



## 10. Protection of services

- Secure services on JBoss Fuse
- Authentication and authorization
- Data encoding on the bus

## 11. Management of errors and exceptions

- Repeating
- Compensation
- Withdrawal
- Transaction processing

## 12. Performance

- Tuning of service parameters (transport protocol, number of threads etc.)
- Cache
- Launching environment parameter adjustment (Java, application server, queue server, data bus etc.)

## 13. Testing services on the ESB

- Methods and tools supporting service testing (automation)

## 14. Advanced services on ESB

- The difference between integration flow, orchestration of services (BPEL) and a business process (BPM)
- Support for business processes (BPM) and orchestration of services (BPEL)
- Support for business principles (BRMS)
- Support for event stream processing (CEP)

## 15. Review of the most popular integration adapters

## 16. Good and bad practices in building of integration solutions based on ESB



- Frequently encountered practices
- Recommended project patterns and anti-patterns to be avoided
- Recommended communication protocols

