



# Enterprise Integration Patterns

Exemplified in Java Business Integration

## Part I - Introduction

Christoph Hartmann

“There is no simple answer for enterprise integration.”

Hohpe et al. *Enterprise Integration Patterns*, 2004, p. 4

# Agenda

3

- Part I - Introduction
  - Fundamental Integration Challenges
  - Integration Styles
  - Service-oriented Architecture
  - Enterprise Service Bus
- Part II - Introduction
  - Design Patterns
  - Java Business Integration
- Part III - Analysis
  - Client
  - Normalized Message Router
  - Service Engine
  - Conclusion

## Fundamental and Integration Challenges

4

- Networks are unreliable
- Networks are slow
- Any two applications are different
- Change is inevitable
  
- Software has to map the enterprise communication structure
- Complex implications
- Changes to legacy systems are not possible in most cases
- Only a few standards for integration exist (XML, XSL, Web Services)
- Enterprise Application Integration is a complex task

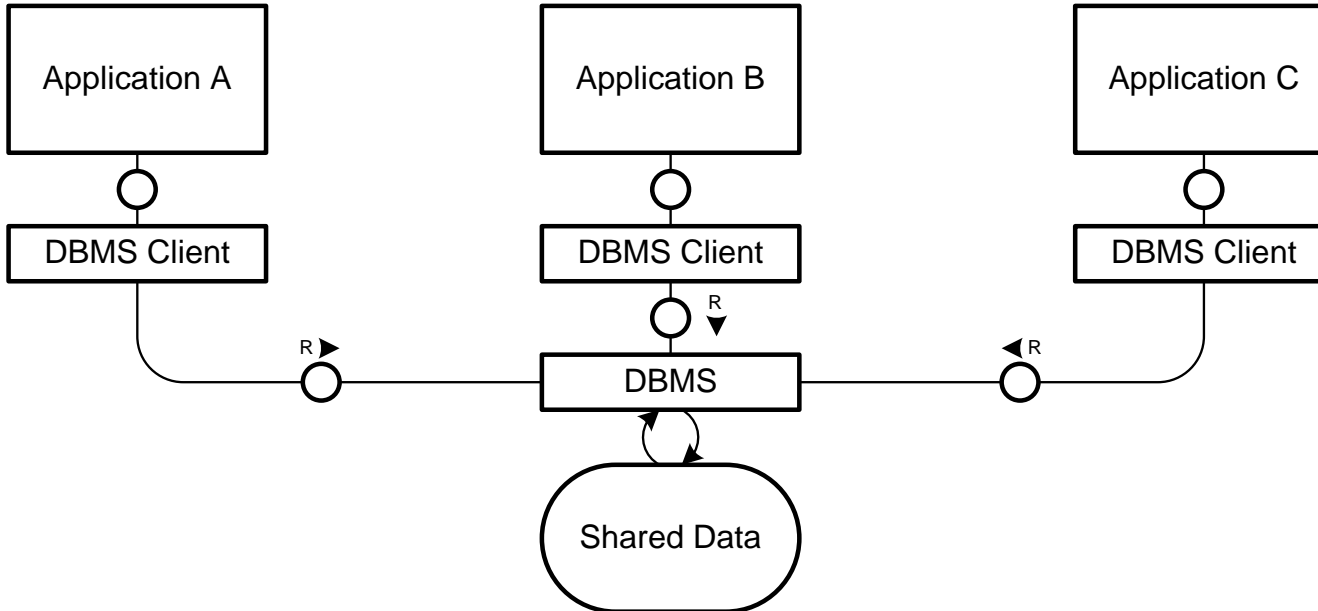
# Integration Styles

5

- File transfer



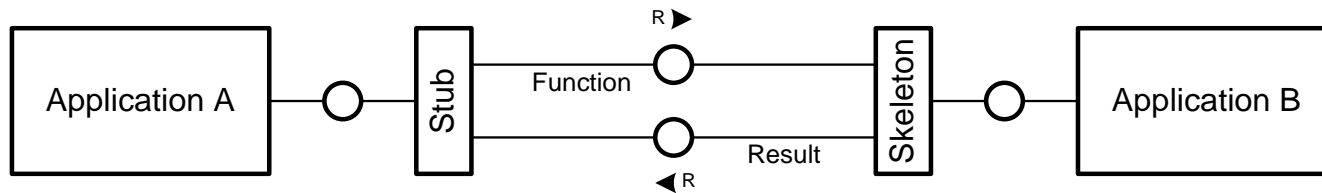
- Shared database



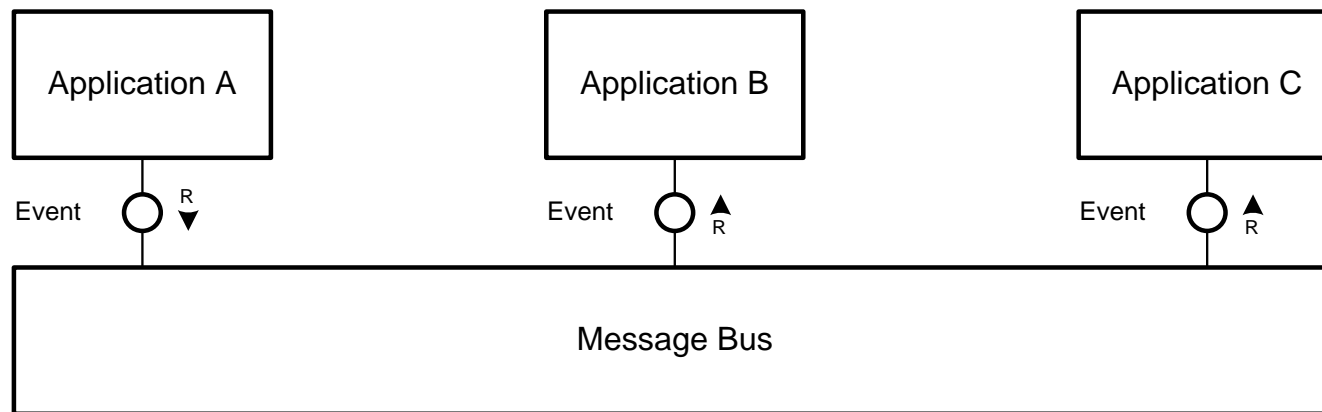
# Integration Styles

6

- Remote procedure invocation



- Messaging

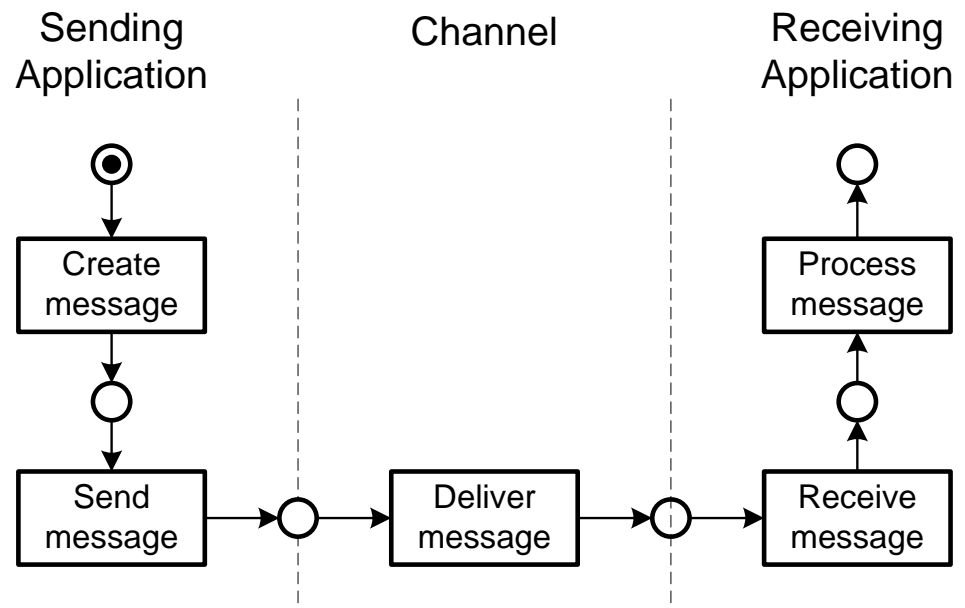


# Messaging

7

- Main steps: *Create, Send, Deliver, Receive, Process*

- Benefits of messaging
  - Remote communication
  - Variable timing
  - Throttling
  - Reliable communication
  - Disconnected operation
  - Mediation
  - Thread management
  - Platform/language integration
  - Asynchronous communication



# Service-oriented Architecture

8

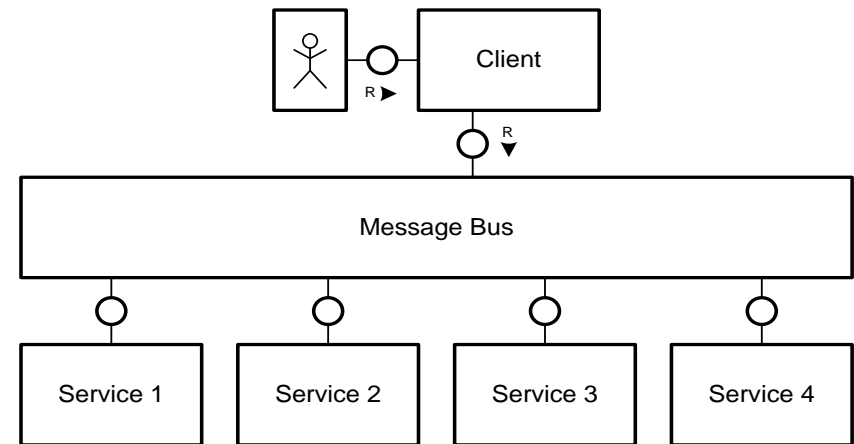
- Key concepts
  - Application front-end
  - Service
  - Service repository
  - Service bus
  - Message
  
- Characteristics
  - Reusable
  - Synchronously callable
  - Asynchronously callable
  - Shared enterprise wide

# Enterprise Service Bus

9

“An Enterprise Service Bus provides a Service-oriented Architecture approach to building composite applications.” *open esb*

- ESB combines
  - Messaging
  - Data transformation
  - Reliable routing
  - Web Services



- ESB provides an implementation infrastructure for SOA based applications
- Standard based integration platform
  - Often called “integration network”

Thanks for your attention.

## Bibliography

- Gregor Hohpe and Bobby Woolf, Enterprise Integration Patterns, Addison Wesley, 2004
- David A. Chappell, Enterprise Service Bus. Theory in Practice, O'Reilly, 2004
- Eric Pulier and Hugh Taylor, Understanding Enterprise SOA, Manning Publications, 2006



# Enterprise Integration Patterns

Exemplified in Java Business Integration

## Part II - Introduction

Christoph Hartmann

# Agenda

13

- Part I - Introduction
  - Fundamental Integration Challenges
  - Integration Styles
  - Service-oriented Architecture
  - Enterprise Service Bus
- Part II - Introduction
  - Design Patterns
  - Java Business Integration
- Part III - Analysis
  - Client
  - Normalized Message Router
  - Service Engine
  - Conclusion

“Each pattern describes a problem which occurs over and over again in our environment, and then describes a core of the solutions to that problem, in such a way that you can use that solution a million times over, without ever doing it the same way twice”

Alexander. et al. *A Pattern Language*. Oxford, 1977.

- Designers should reuse solutions that have worked in the past
  - Patterns were not invented but they were discovered
  
- Patterns have four essential elements
  - Pattern name
  - Problem
  - Solution
  - Consequences
  
- Support the decision of design alternatives
  
- The pattern value consists in helping to communicate ideas to others

- An open and flexible way to integrate applications
- Implementation of an Enterprise Service Bus
- Try to avoid problems that are known from classic Enterprise Application Integration (EAI) frameworks
  
- Specifies:
  - Plug-in components
  - Message Routing
  - Component life cycle
  - Component administration
  - Component packaging
- Is based upon WSDL 2.0

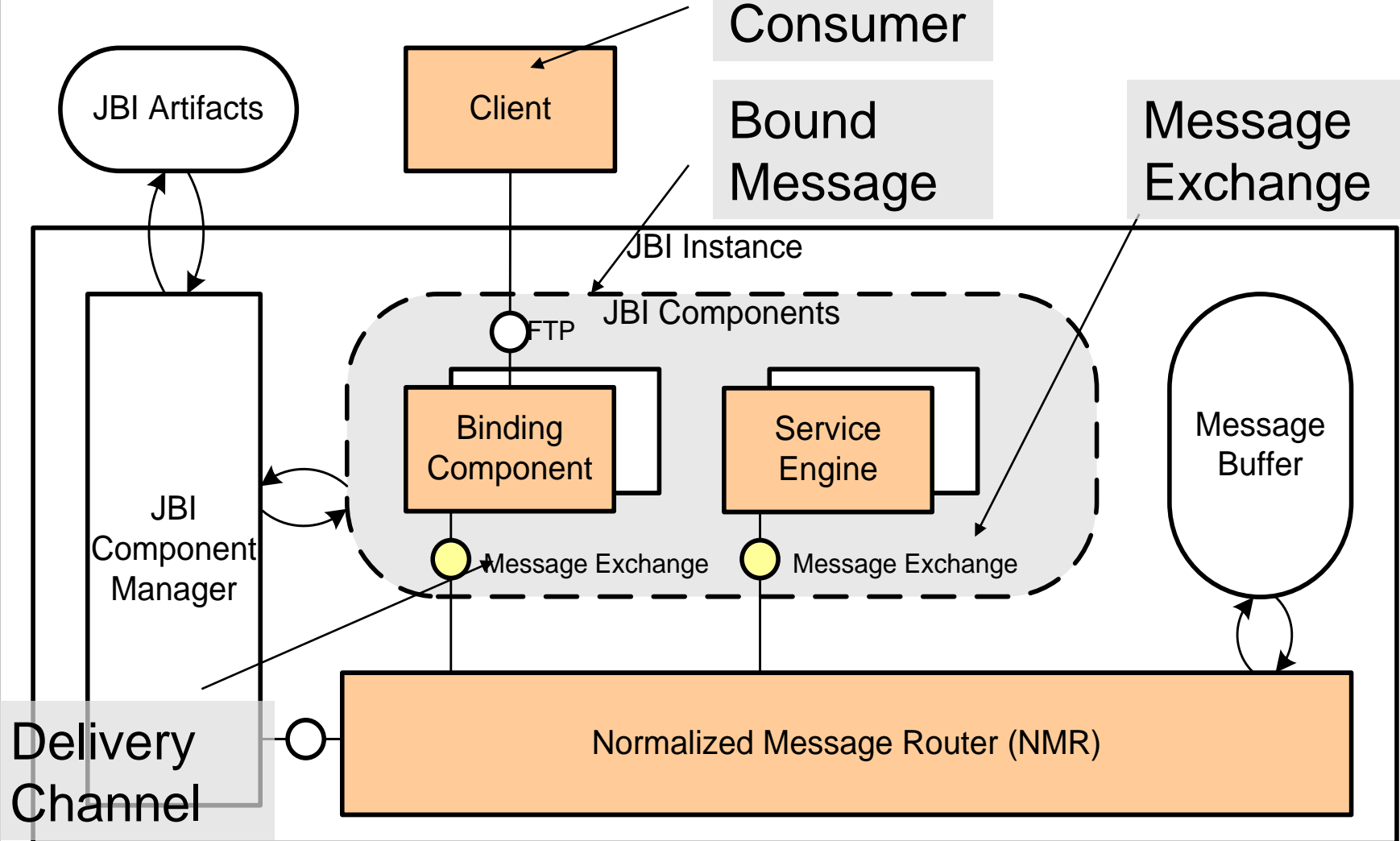
# JBI Architecture

Service Consumer

Bound Message

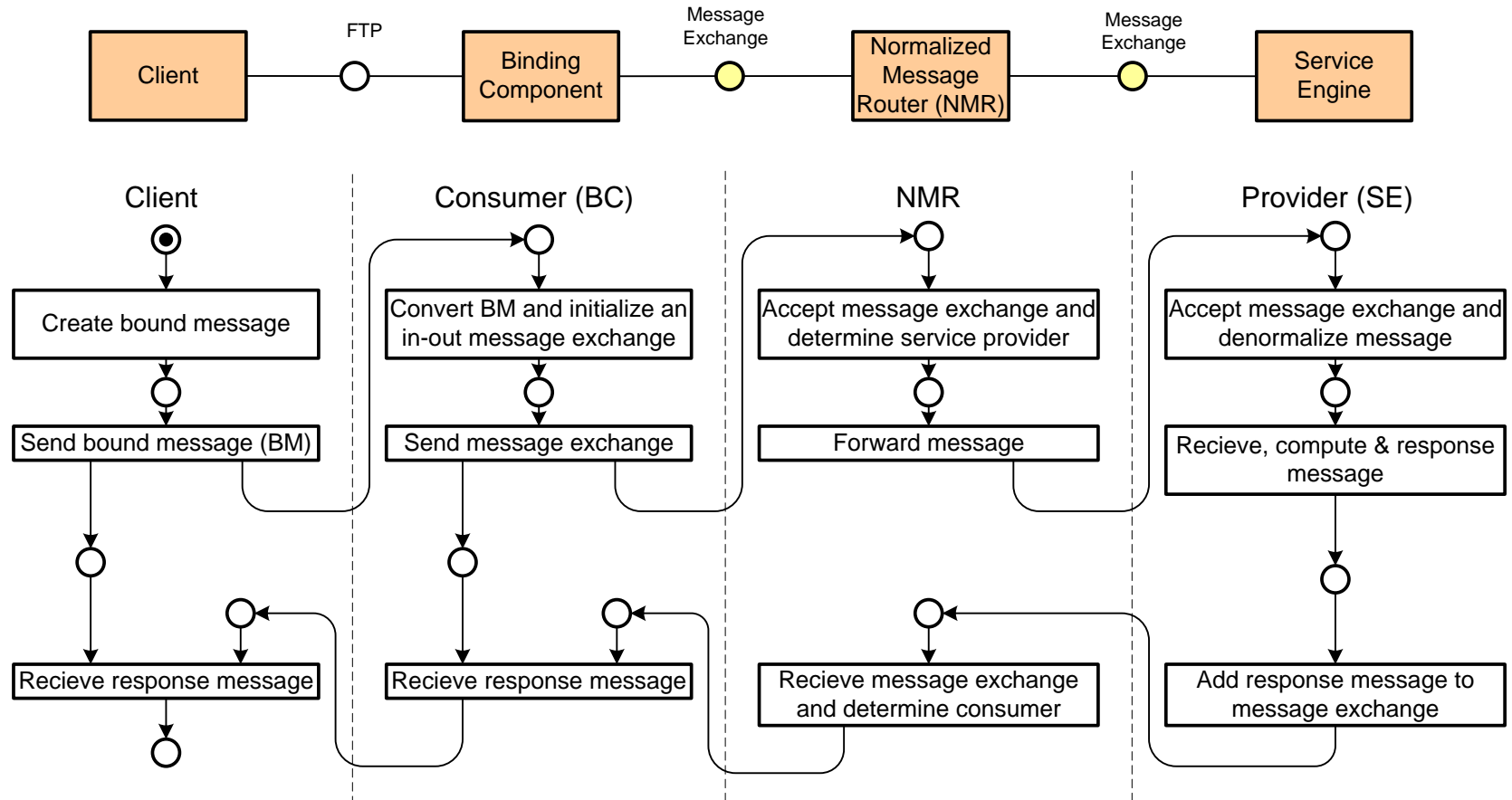
Message Exchange

17



# Message Exchange (Request-Response)

18





# Enterprise Integration Patterns

Exemplified in Java Business Integration

## Part II - Analysis

Christoph Hartmann

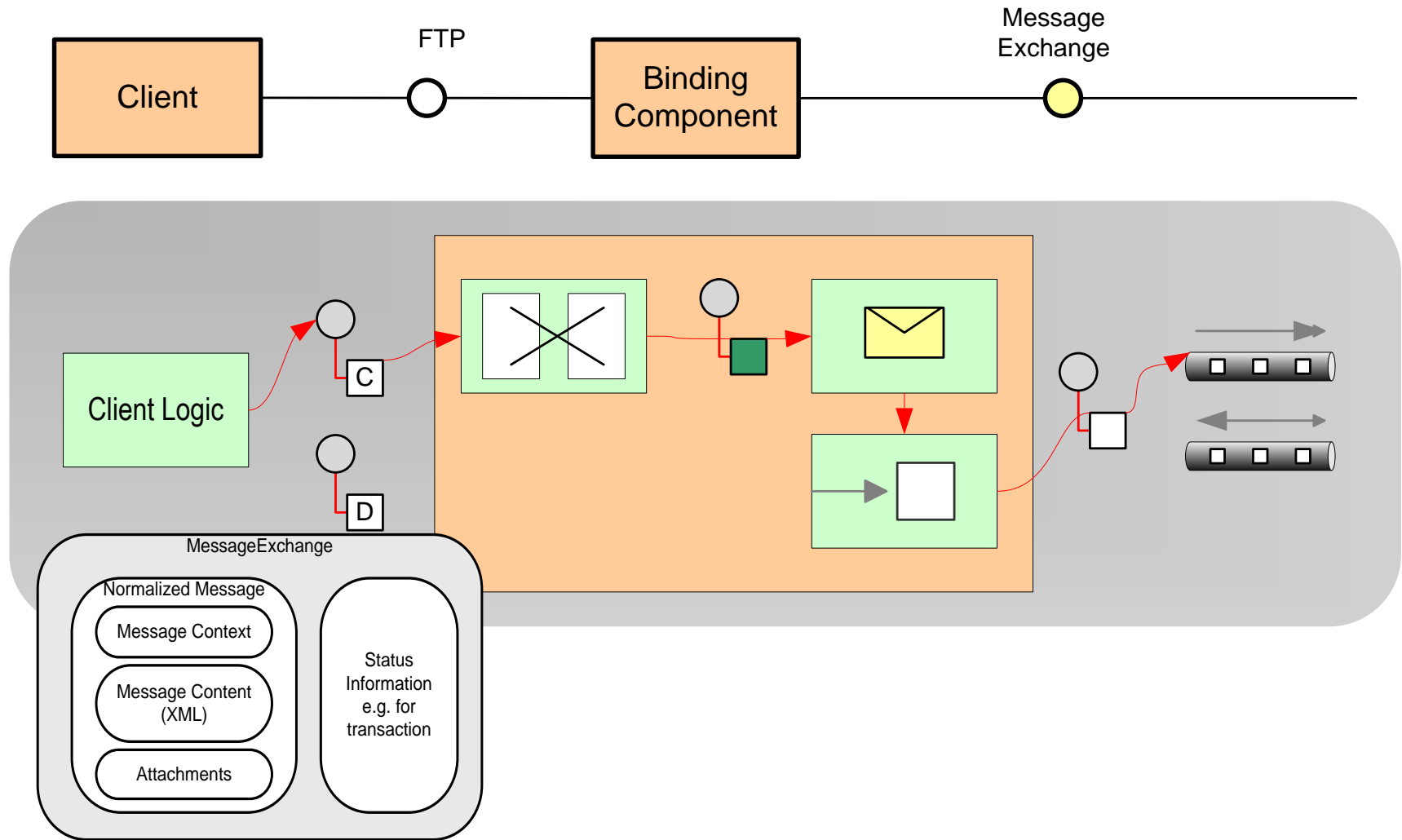
# Agenda

20

- Part I - Introduction
  - Fundamental Integration Challenges
  - Integration Styles
  - Service-oriented Architecture
  - Enterprise Service Bus
- Part II - Introduction
  - Design Patterns
  - Java Business Integration
- Part III - Analysis
  - Client
  - Normalized Message Router
  - Service Engine
  - Conclusion

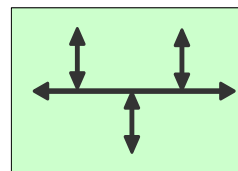
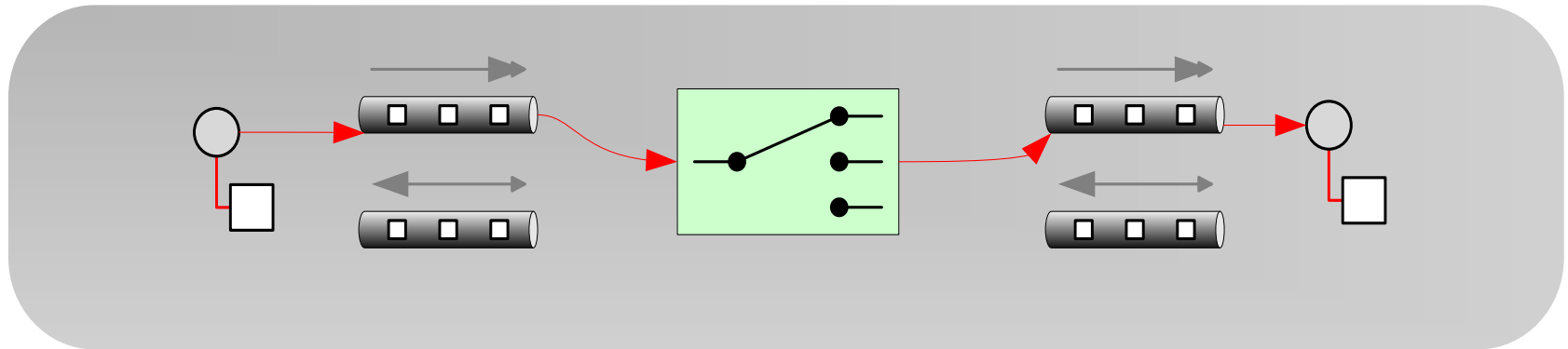
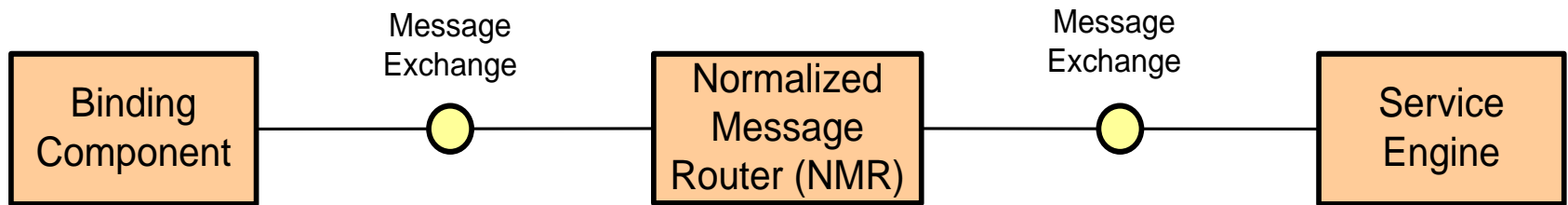
# Analysis - Client

21



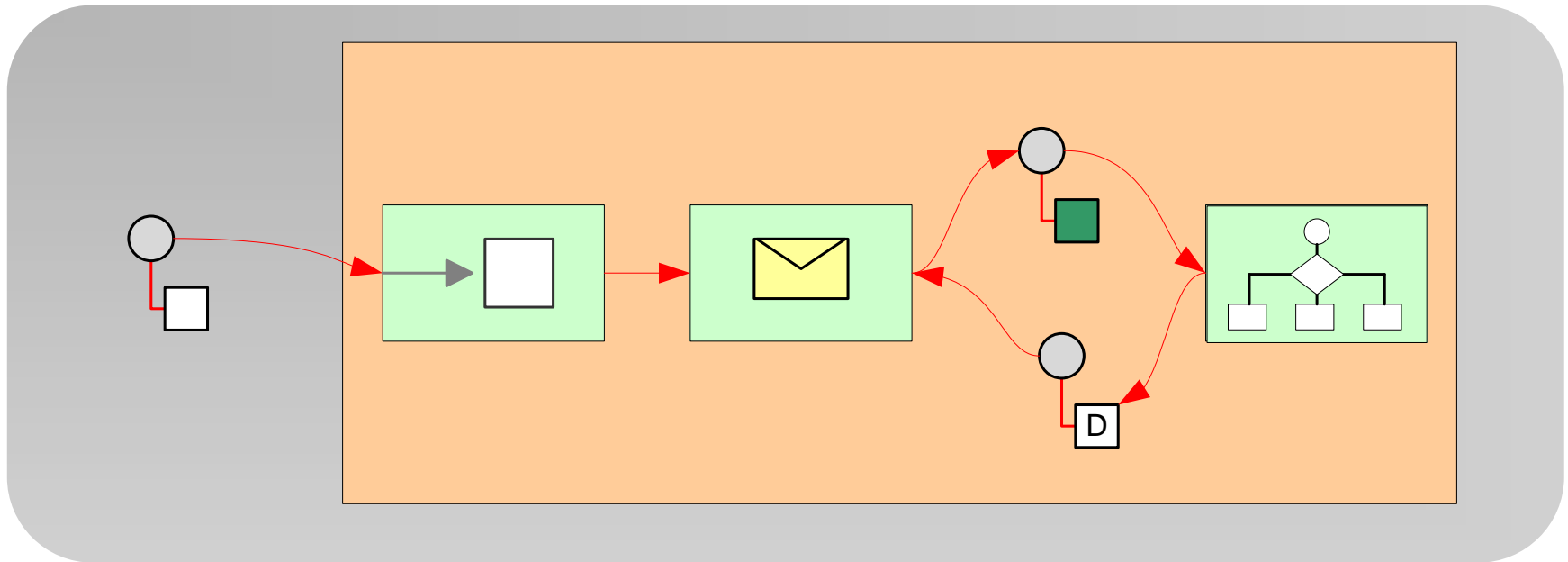
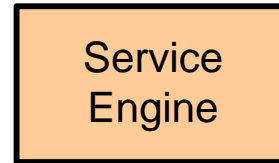
# Analysis – Normalized Message Router

22



# Analysis – Service Engine

23



## Conclusion

24

- Patterns can be discovered in real world scenarios
- Finding patterns can decrease development time
- Patterns improve the communicate with colleagues
  
- Only integration patterns were presented
- Other patterns are:
  - OO patterns
  - Enterprise application architecture patterns
  - SOA pattern
- Granularity of patterns is questionable

## Bibliography

25

- Alexander. et al. *A Pattern Language*. Oxford, 1977.
- E. Gamma et al. *Design Patterns – Elements of Reusable Object-Orientated Software*. Addison-Wesley, 1995.
- M. Fowler. *Patterns of Enterprise Application Architecture*. Addison-Wesley, 2002
- G. Hohpe and B. Woolf. *Enterprise Integration Patterns - Deigning, Building, and Deploying Messaging Solutions*. Pearson Education, 2004.
- D. A. Chappell. *Enterprise Service Bus - Theory in Practice*. O'Reilly, 2004.
- Ch. Hartmann. *Einführung in Java Business Integration. Seminar Systemmodellierung*, 2006.
- A. Sterkin. *Teaching Design Patterns*.
- JBI Specifiction JSR 208

Thanks for your attention.