Spring Integration: Getting Started

INTRODUCTION AND GETTING SET UP

Jesper de Jong
SOFTWARE ARCHITECT

@jesperdj www.jesperdj.com

Spring Integration: Getting Started

Spring Integration

Integrate many different kinds of systems without writing plumbing code

Separate integration code from business logic

Use Enterprise Integration Patterns in your Spring-based applications

Overview

Setting the scene

- What is “integration”?
- What is “asynchronous messaging”?

Enterprise Integration Patterns

Spring Integration

Practical points

- Prerequisites
- Where this course fits in
- Setting up your system

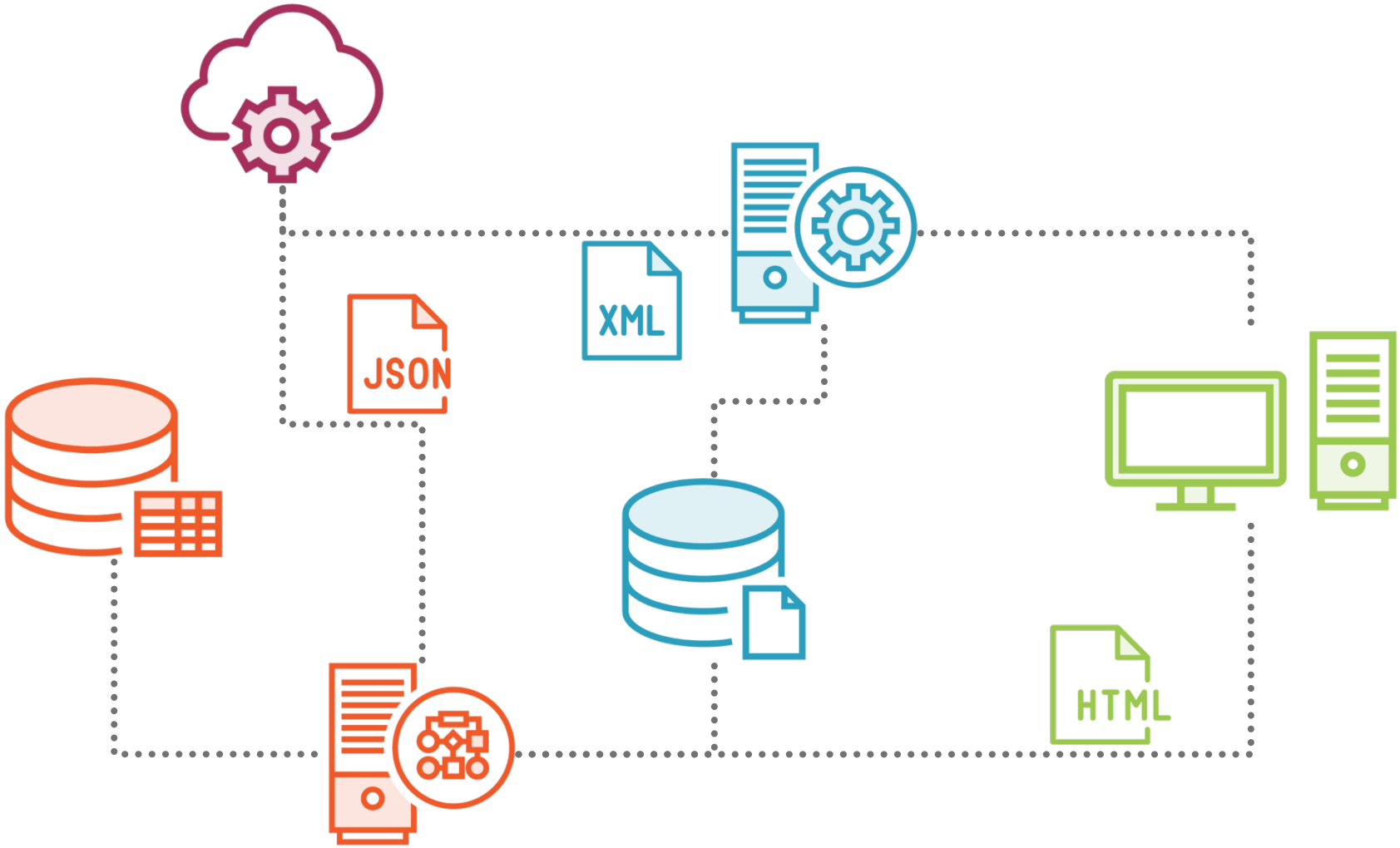
Setting the Scene

Enterprise Integration Patterns

Asynchronous messaging

Spring Integration

Setting the Scene



Design Patterns

General, reusable solutions
to common software design problems.

Enterprise Integration Patterns

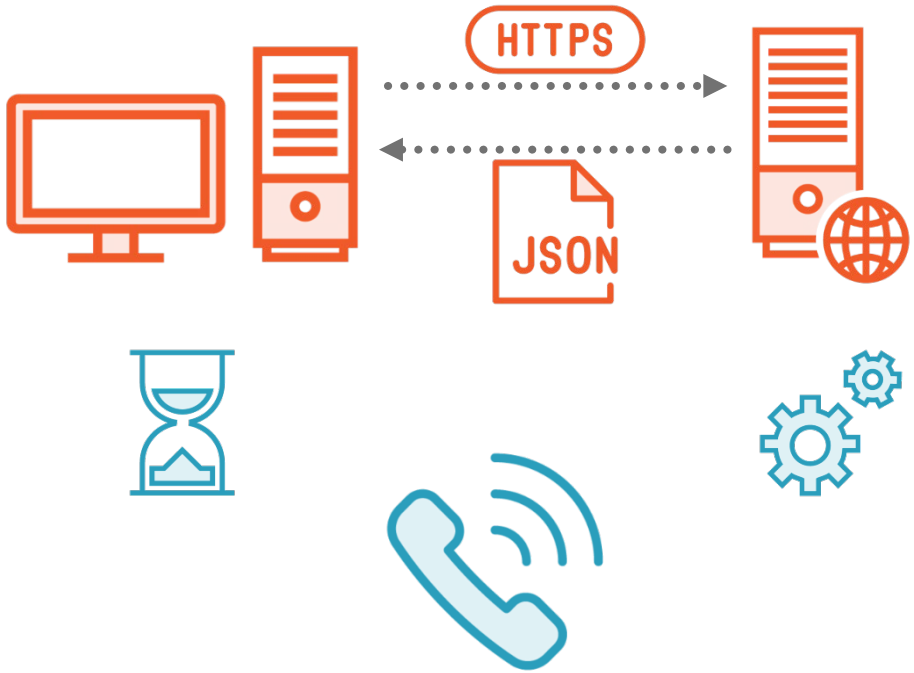
Designing, Building and Deploying Messaging Solutions

(Gregor Hohpe, Bobby Woolf, 2003)

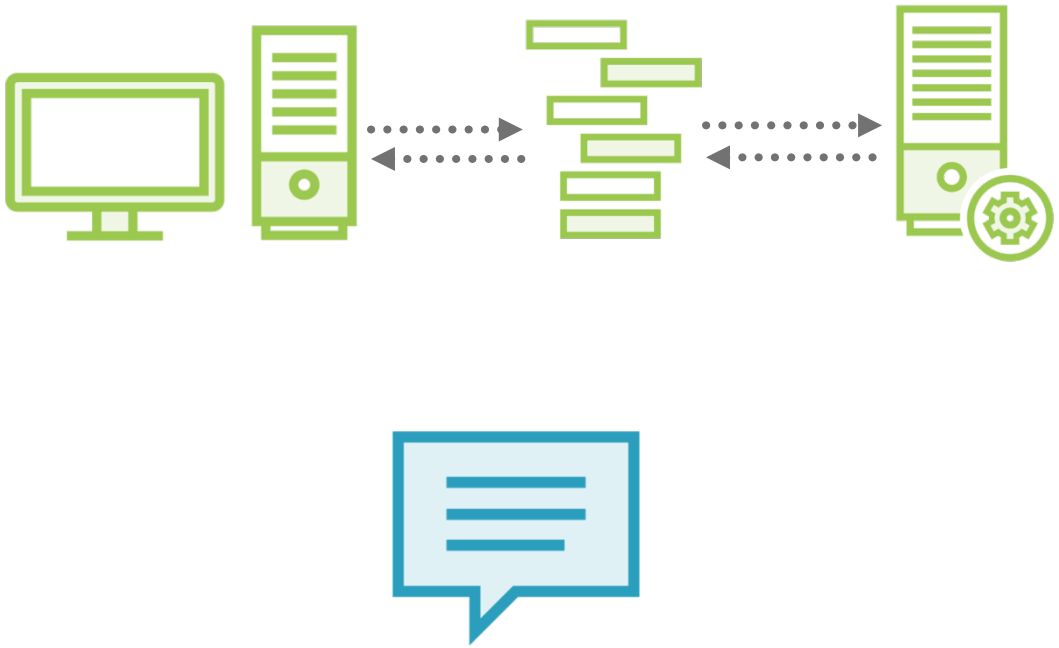
https://www.enterpriseintegrationpatterns.com

Synchronous vs Asynchronous

Synchronous



Asynchronous



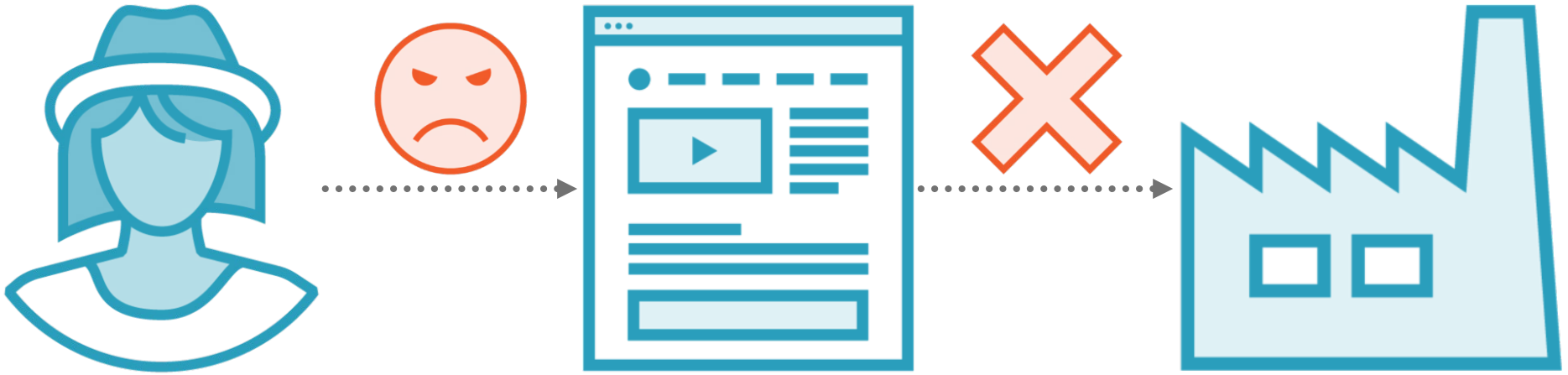
Asynchronous Advantages

More responsive system because sender does not have to wait

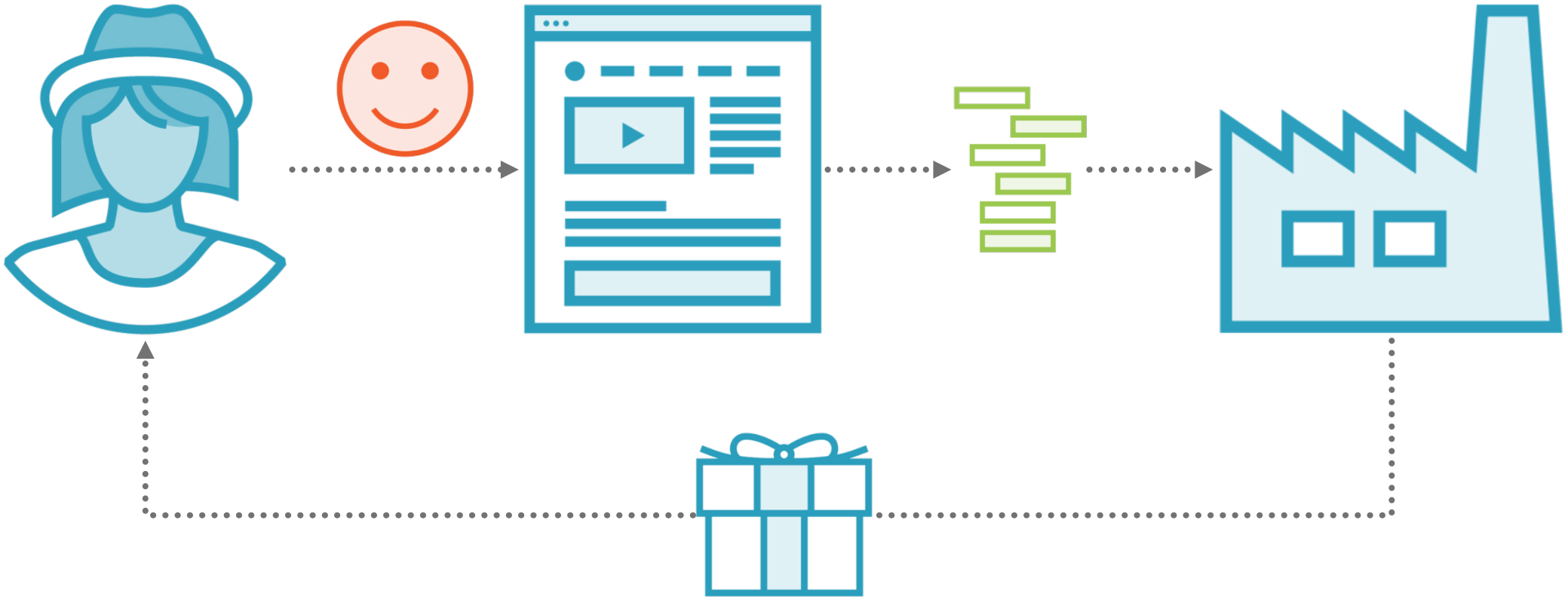
Message buffering makes the system resilient to failure

Receiver processes messages at its own rate

Asynchronous Advantages



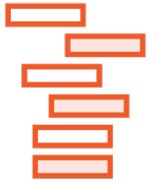
Asynchronous Advantages



Spring Integration

An implementation of Enterprise Integration Patterns based on the Spring Framework

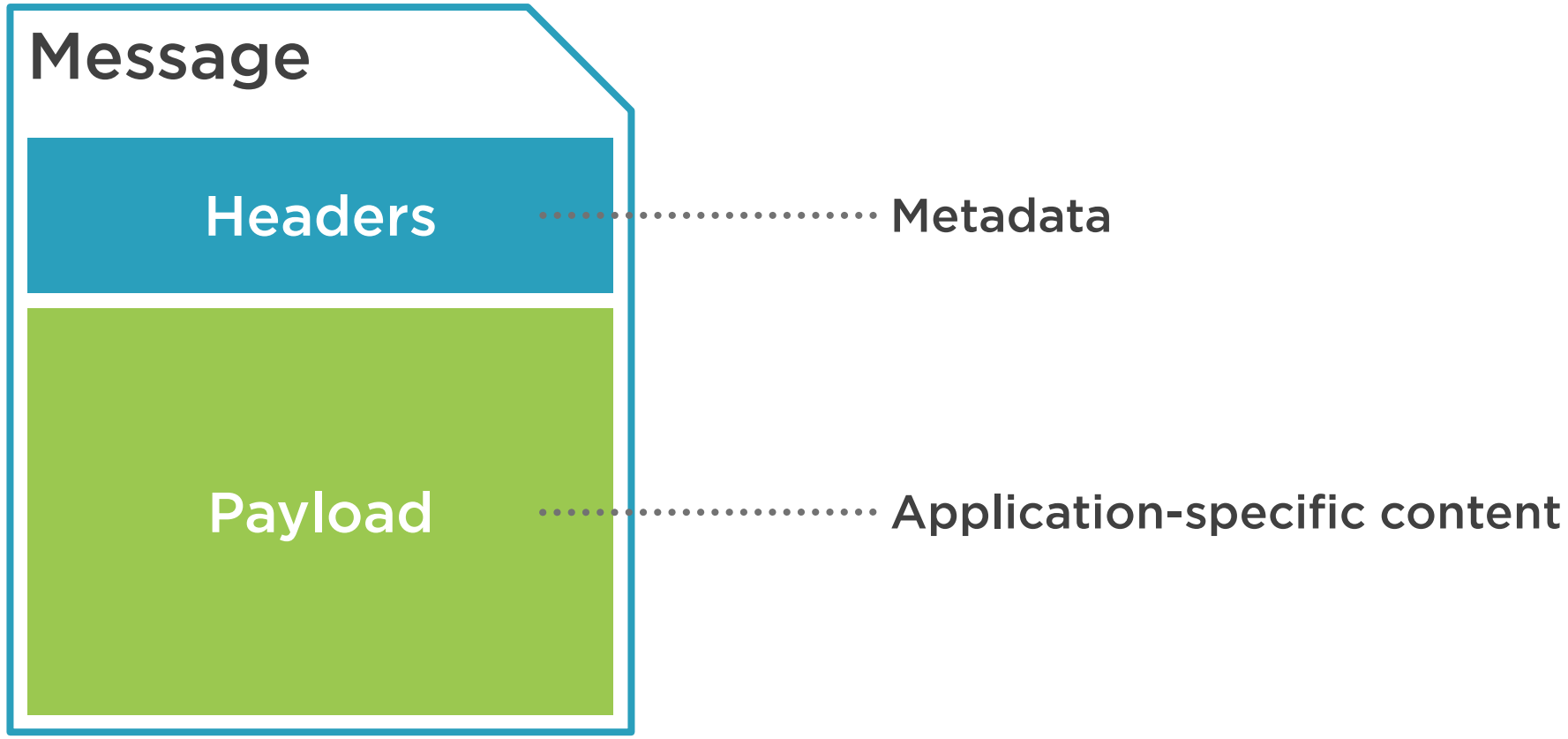
Familiar Spring Framework programming model



Connect business logic components using asynchronous messaging

Understanding Enterprise Integration Patterns

Message


Message
Headers
Metadata
Payload
Application-specific content

Message Channel

Point-to-point channel

One sender, one receiver

Example: Queue

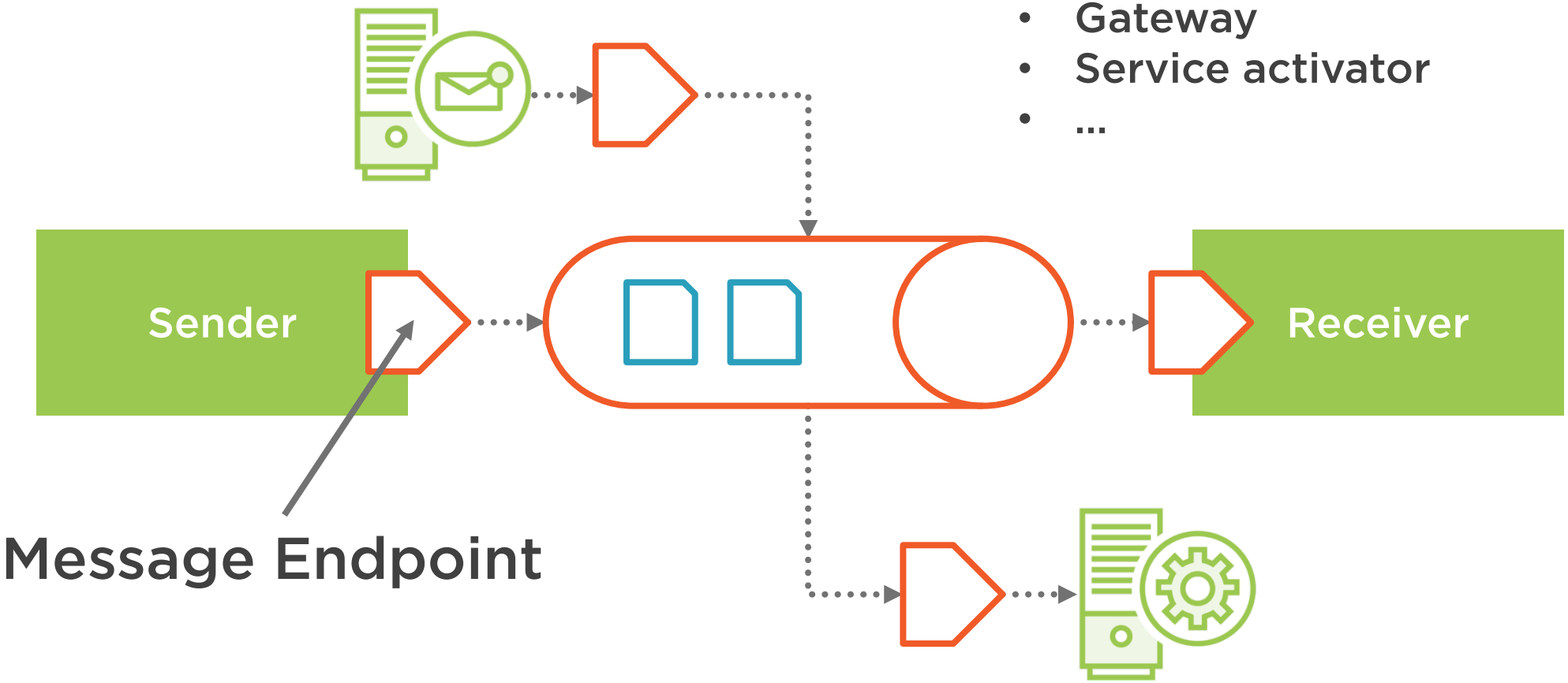
Publish-subscribe channel

One sender, multiple receivers

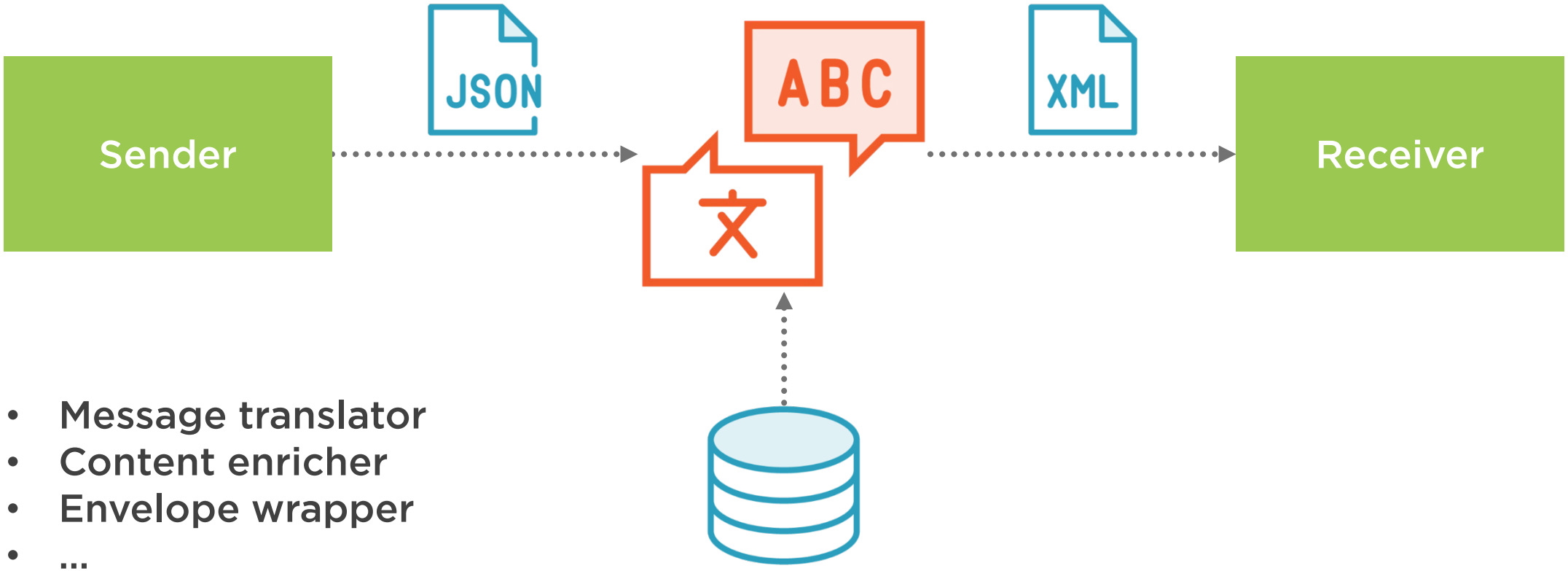
Example: Event notifications

Message Endpoint

- Channel adapter
- Gateway
- Service activator
- ...

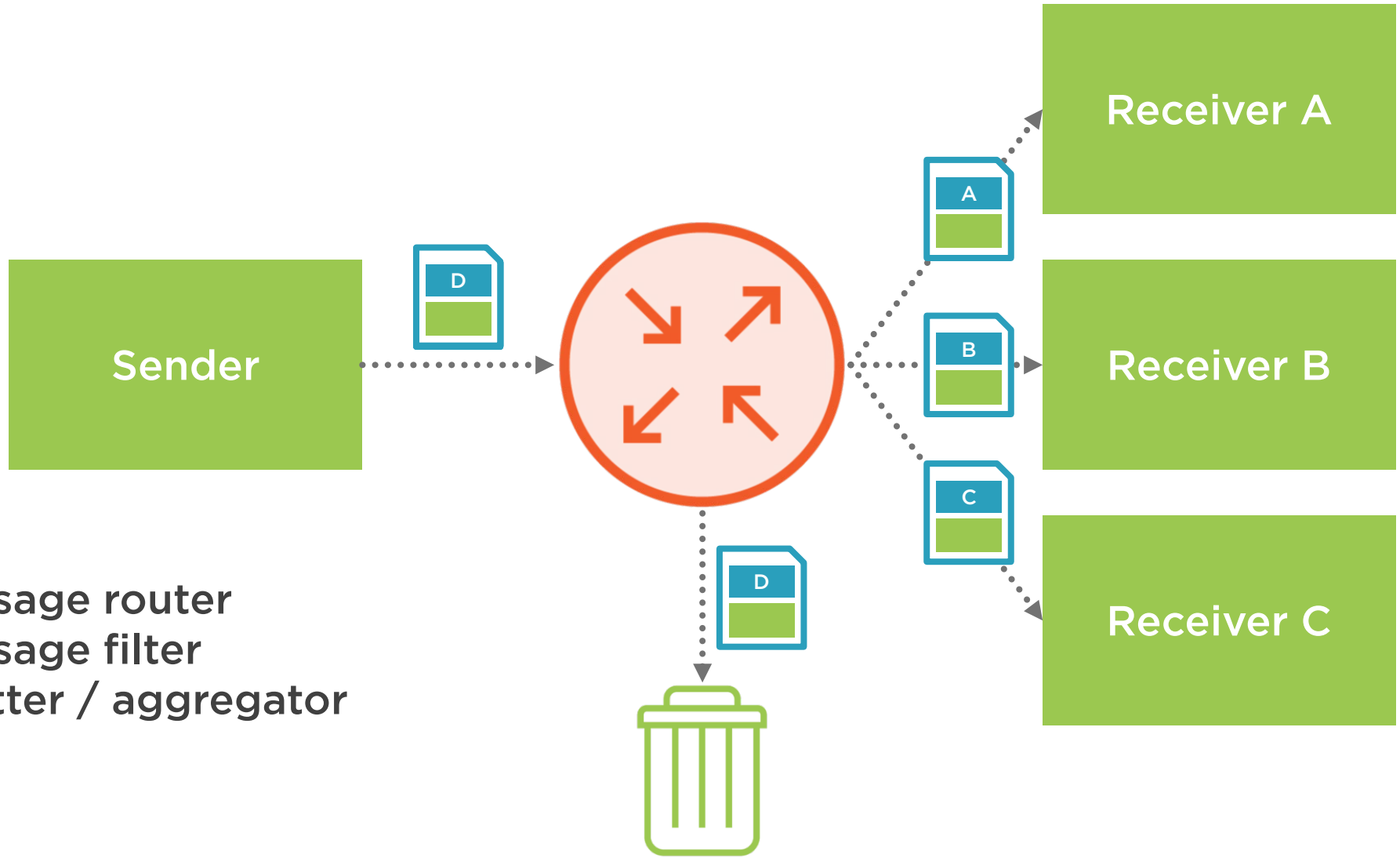


Message Transformation



- Message translator
- Content enricher
- Envelope wrapper
- ...

Message Routing



- Message router
- Message filter
- Splitter / aggregator
- ...

Enterprise Integration Patterns

Message

Message Channel

Message Endpoint

Message Transformation

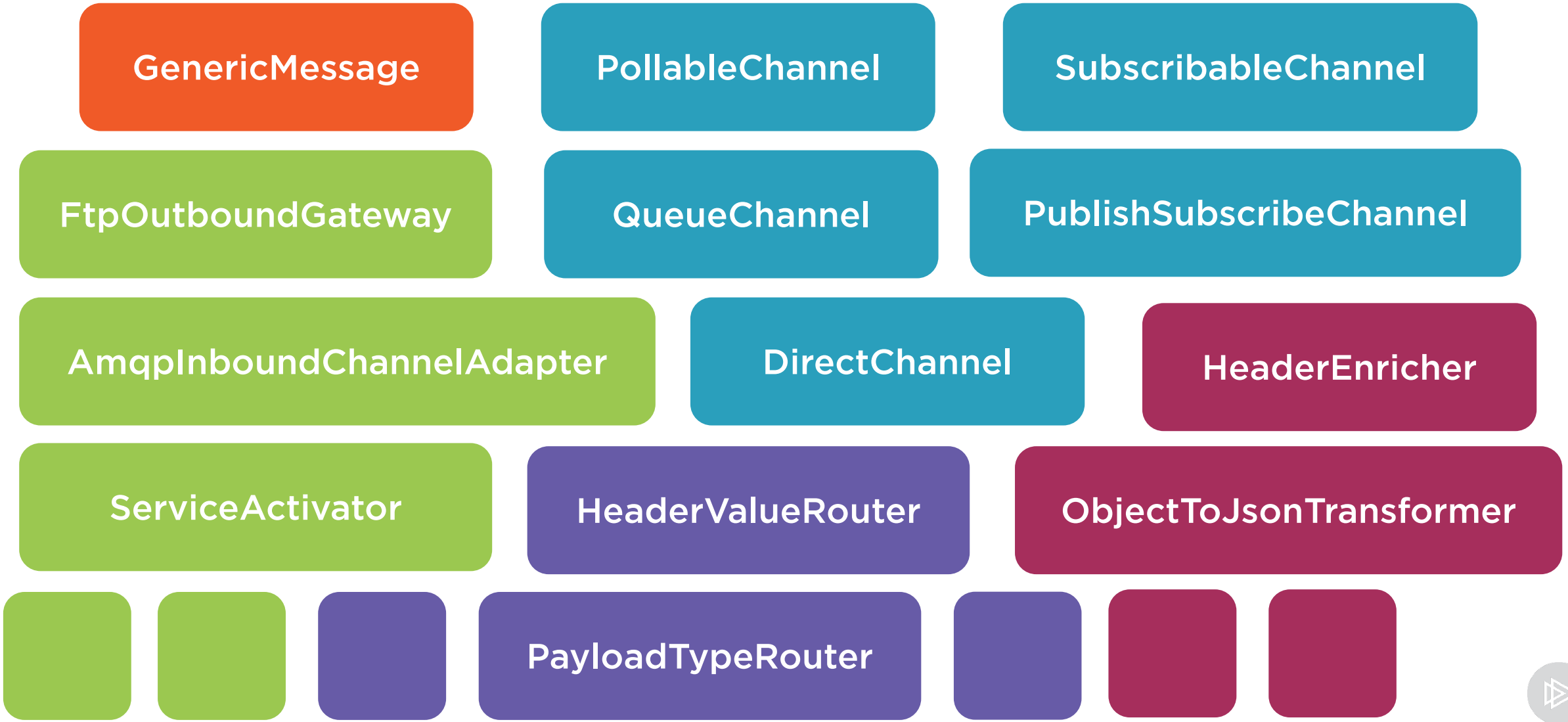
Message Routing

Introducing Spring Integration

Spring Integration

An implementation of the Enterprise Integration Patterns based on the Spring Framework.

Spring Integration



Spring Integration Configuration

XML Config

Spring Integration XML namespaces

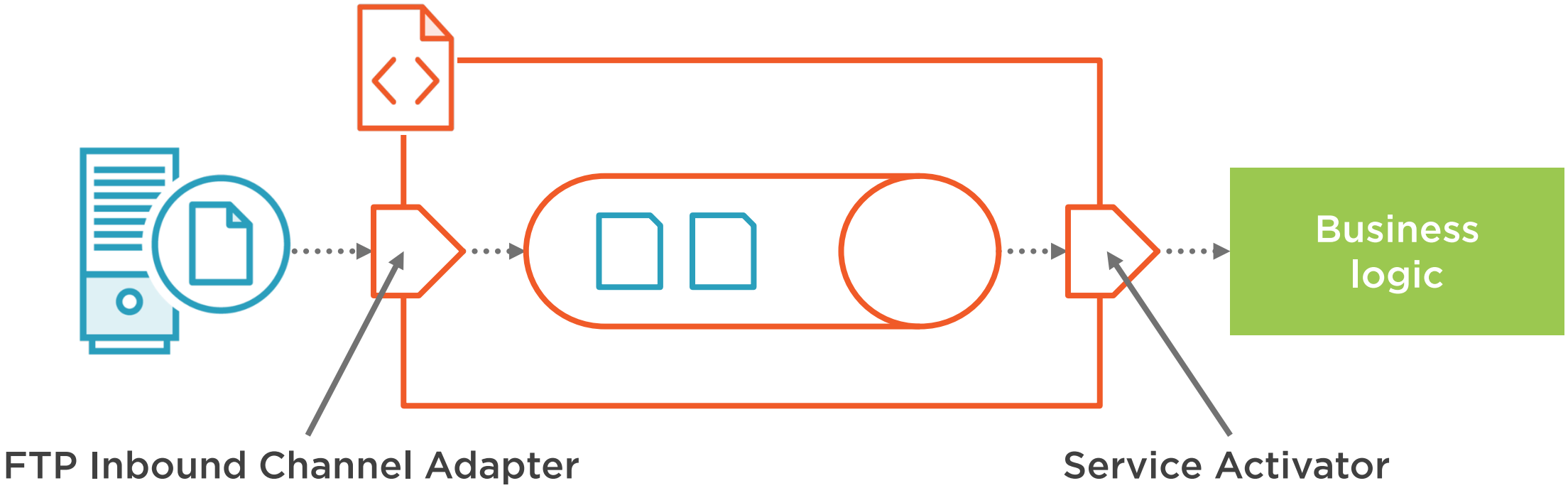
Java Config

Spring Integration annotations

DSL

Spring Integration domain-specific config language

Declarative Configuration



More Information

https://spring.io/projects/spring-integration

Prerequisites and Learning Path

Prerequisites

- Spring Framework
- Spring Web MVC
- Spring Data JPA
- Spring Boot

Spring Integration Learning Path

Setting up Your System

Setting up Your System

Java Development Kit

- https://oracle.com/javase
- https://adoptopenjdk.net

Apache Maven

- https://maven.apache.org

Your favorite IDE

RabbitMQ

- https://www.rabbitmq.com

Summary

Enterprise Integration Patterns

Asynchronous messaging

Core concepts

- Message
  - Payload and headers
- Message Channel
  - Point-to-point
  - Publish-subscribe
- Message Endpoint
- Message Transformation
- Message Routing

Spring Integration