Getting Started with Point-to-Point Messaging

Jesper de Jong
SOFTWARE ARCHITECT

@jesperdj www.jesperdj.com

Overview

Demo application

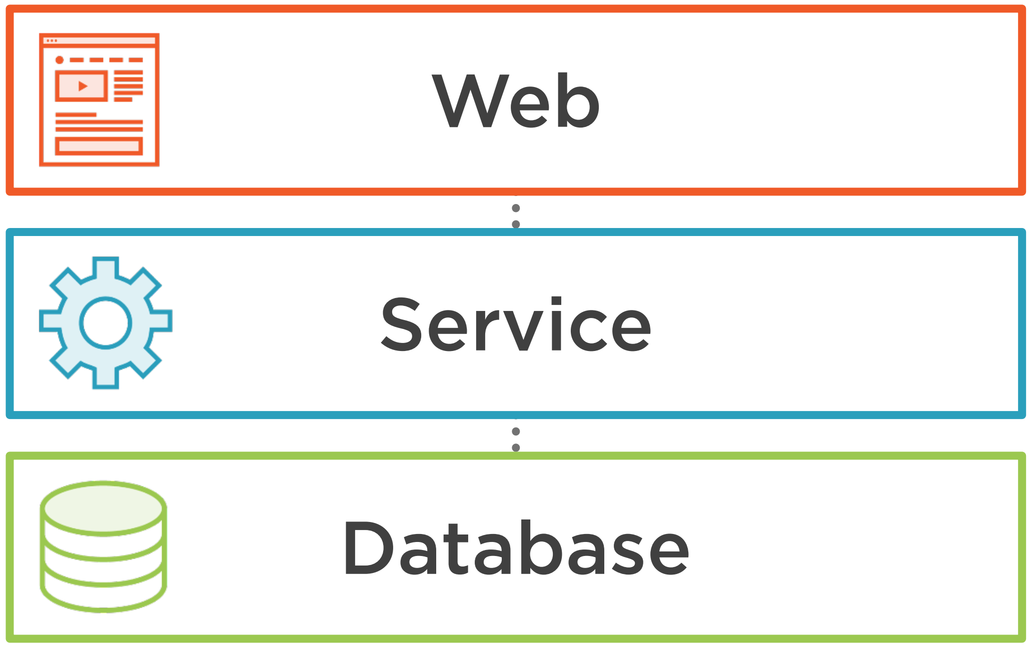
Adding Spring Integration

Message channel implementations

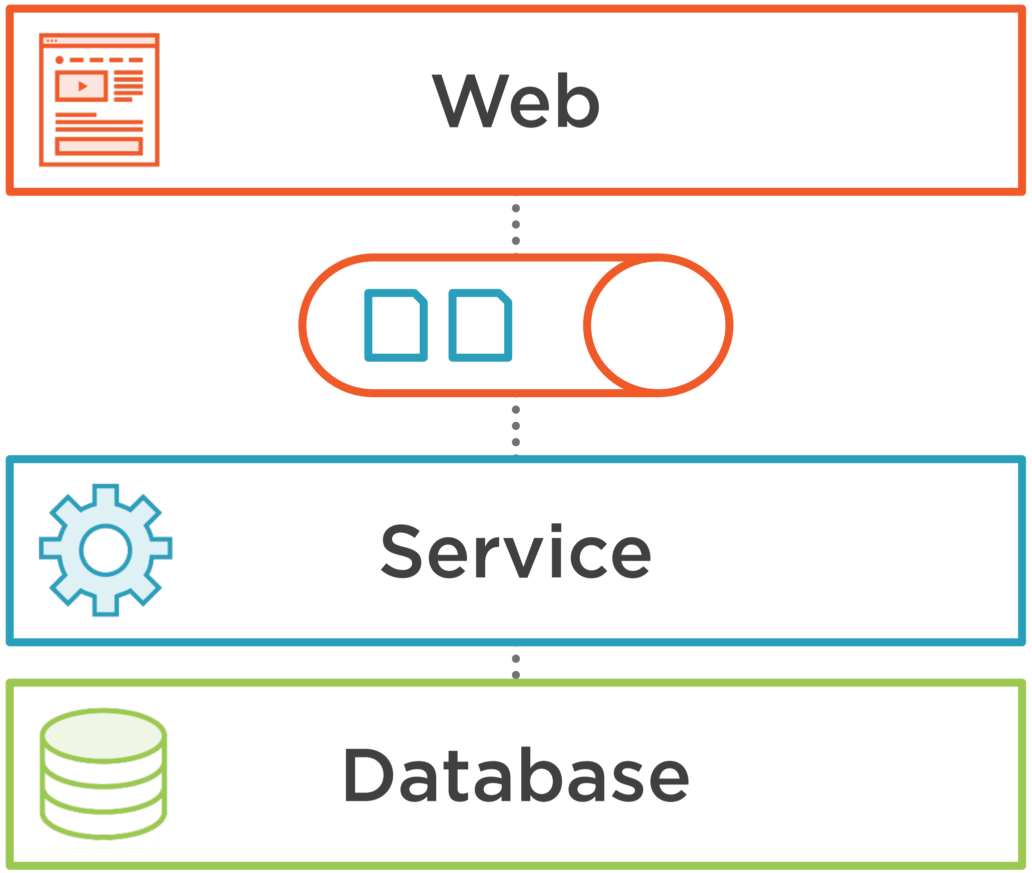
Setting up the Demo Application

Adding Spring Integration

Demo Application Architecture



Demo Application Architecture



Using Java Configuration

Spring Integration Configuration

XML Config

Spring Integration XML namespaces

Java Config

Spring Integration annotations

DSL

Spring Integration domain-specific config language

Working with the Service Activator

Using the Spring Integration DSL

Understanding Message Channels

Message Channel Implementations

Message channels

DirectChannel	QueueChannel
ExecutorChannel	RendezvousChannel
PublishSubscribeChannel	PriorityChannel

Interface MessageChannel

```java
public interface MessageChannel {
    boolean send(Message message);
    boolean send(Message message, long timeout);
}
```

Message Channel Implementations

Message channels

Subscribable channels

DirectChannel

ExecutorChannel

PublishSubscribeChannel

Event-Driven Consumer

Pollable channels

QueueChannel

RendezvousChannel

PriorityChannel

Polling Consumer

Interface SubscribableChannel

Event-Driven Consumer

```
public interface SubscribableChannel extends MessageChannel {
    boolean subscribe(MessageHandler handler);
    boolean unsubscribe(MessageHandler handler);
}
```

Interface PollableChannel

Polling Consumer

```
public interface PollableChannel extends MessageChannel {

    Message<?> receive();

    Message<?> receive(long timeout);

}
```

Message Channel Implementations

Message channels

Subscribable channels	Pollable channels
DirectChannel	QueueChannel
ExecutorChannel	RendezvousChannel
PublishSubscribeChannel	PriorityChannel
Push	Pull
Non-buffering	Buffering

Subscribable Message Channels

Subscribable Message Channels

Subscribable message channels

DirectChannel

ExecutorChannel

PublishSubscribeChannel

Event-Driven Consumer Push Non-buffering

Subscribable Message Channels

Subscribable message channels

Unicasting dispatcher

- DirectChannel
- ExecutorChannel

Broadcasting dispatcher

- PublishSubscribeChannel

Subscribable Message Channels

Subscribable message channels

Unicasting dispatcher

DirectChannel

ExecutorChannel

Point-to-Point Channel

Broadcasting dispatcher

PublishSubscribeChannel

Publish-Subscribe Channel

Class DirectChannel

Point-to-Point Channel

Event-Driven Consumer

```
public boolean send(Message message) {
    // Get subscribed handler
    MessageHandler handler = ...;

    if (handler != null) {
        // Call handler
        handler.handleMessage(message);
        return true;
    }

    return false;
}
```

(Pseudo-code, not the actual implementation)

Class ExecutorChannel

Point-to-Point Channel **Event-Driven Consumer**

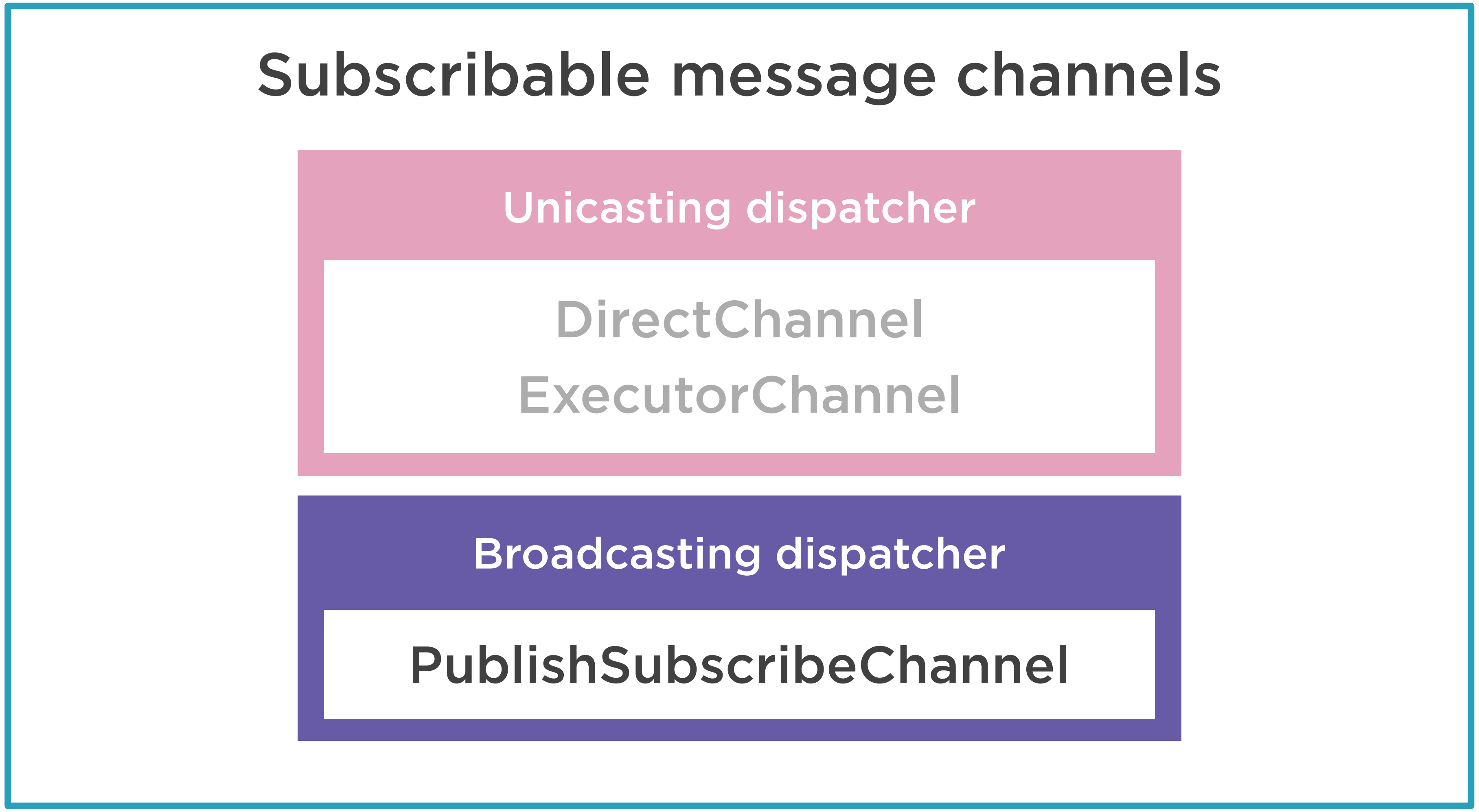
```
public boolean send(Message message) {
    // Get subscribed handler
    MessageHandler handler = ...;

    if (handler != null) {
        // Call handler using executor
        executor.execute(() ->
            handler.handleMessage(message));
        return true;
    }

    return false;
}
```

(Pseudo-code, not the actual implementation)

Subscribable Message Channels



Pollable Message Channels

Pollable Message Channels

Pollable message channels

QueueChannel
RendezvousChannel
PriorityChannel

Polling Consumer **Pull** **Buffering**

QueueChannel

Buffers messages in an in-memory queue

Unbounded capacity

RendezvousChannel

Zero-capacity queue

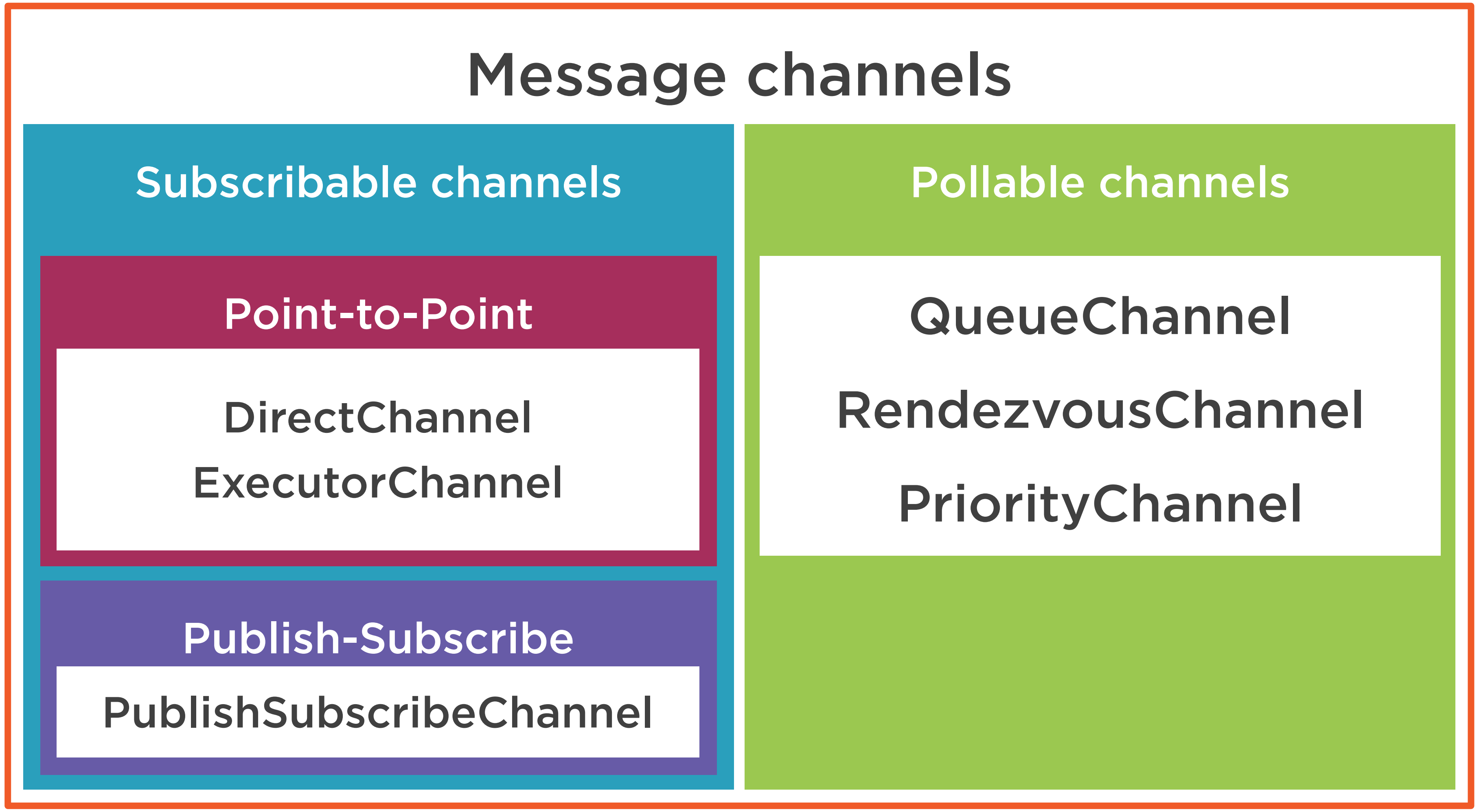
Blocks until sender and receiver meet

PriorityChannel

Buffers messages in a priority queue

Default ordering by “priority” header

Message Channel Implementations



Summary

Demo application

Spring Integration

- Configuration using XML, annotations, domain-specific language

Service Activator

Message channel implementations

- Subscribable and pollable channels
- Point-to-point and pub-sub channels