

Spring Integration



JTeam

Innovating J2EE Development

Roberto van der Linden

Agenda

- Background
- Goals and principles
- Main components

- Project ONS
- Demo
- Questions



JTeam

Innovating J2EE Development

Background

- Current version 1.0.3
- Message-driven architecture
 - Horizontal perspective
- Messaging and integration concerns handled by framework
- Wide variety of configuration options



JTeam

Innovating J2EE Development

Pipes and Filters Architecture

- Divide larger processing task into a sequence of smaller steps
- Filters
 - Independent processing steps
- Pipes
 - Connect steps by channels



Goals and principles

- Simple model for implementing complex enterprise integration solutions
- Facilitate asynchronous, message-driven behavior
- Components should be loosely coupled
- Separation of concerns between business logic and integration logic



JTeam

Innovating J2EE Development

Main components

- TNT mail example

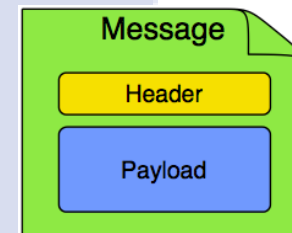
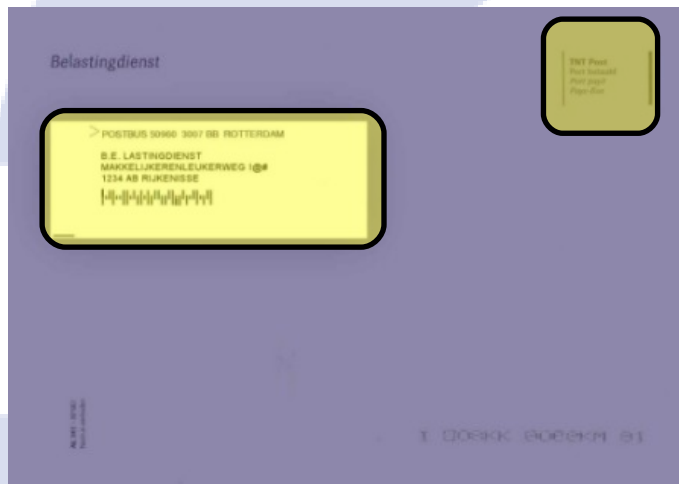


JTeam

Innovating J2EE Development

Main components

- Message
 - Generic container for data
 - Headers as key-value pairs



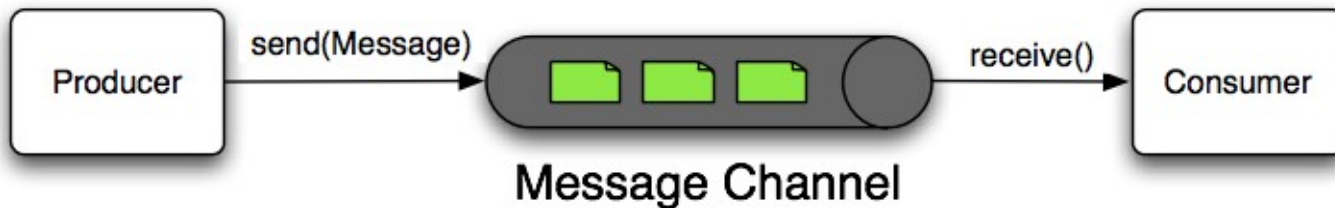
JTeam

Innovating J2EE Development

Main components

- Message Channel
 - PollableChannel
 - SubscribableChannel

```
<channel id="numberChannel"  
        datatype="java.lang.Number"/>
```



JTeam

Innovating J2EE Development



Main components

- **Message Endpoint**
 - Represents the "filter" of a pipes-and-filters architecture
 - Handles messages
 - Mapped to Message Channels



Splitter

- Split a message into multiple messages
- Send each message to its output channel

```
<splitter id="splitter"  
  ref="splitterBean"  
  method="split"  
  input-channel="inputChannel"  
  output-channel="outputChannel" />
```



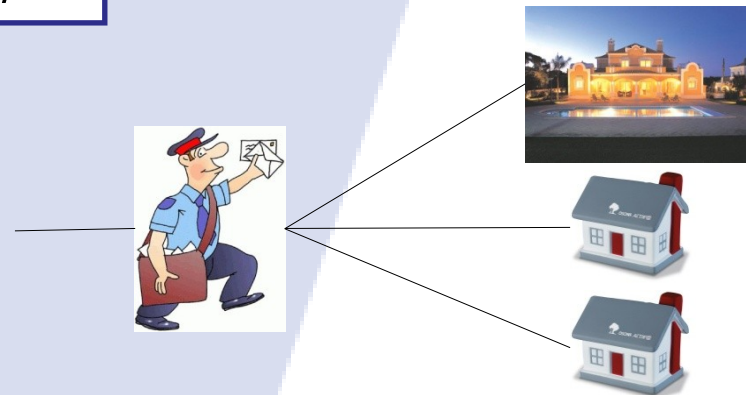
JTeam

Innovating J2EE Development

Router

- Routes a single message to the correct Channel.
 - PayloadTypeRouter
 - HeaderValueRouter

```
<router input-channel="input"  
        ref="somePojo"  
        method="someMethod"/>
```



JTeam

Innovating J2EE Development

Filter

- Used to decide whether a Message should passed along or dropped.
- Discard channel

```
<filter input-channel="input"  
  ref="selector"  
  method="someBooleanReturningMethod"  
  throw-exception-on-rejection="true"  
  discard-channel="rejectedMessages"  
  output-channel="output"/>
```



Transformer

- Transforms the Message to another type.
- Loose-coupling Message Producers and Message Consumers.

```
<transformer id="testTransformer"  
    ref="testTransformerBean"  
    input-channel="inChannel"  
    method="transform"  
    output-channel="outChannel"/>
```



JTeam

Innovating J2EE Development

Aggregator

- Mirror-image of the Splitter
- Receives multiple Messages and combines them into a single Message.
- CorrelationStrategy and CompletionStrategy

```
<aggregator id="completelyDefinedAggregator"  
  input-channel="inputChannel"  
  output-channel="outputChannel"  
  discard-channel="discardChannel"  
  ref="aggregatorBean" method="add"  
  completion-strategy="completionStrategyBean"  
  completion-strategy-method="checkCompleteness"  
  correlation-strategy="correlationStrategyBean"  
  correlation-strategy-method="groupNumbersByLastDigit"  
  timeout="42" send-partial-result-on-timeout="true"  
  send-timeout="86420000" />
```



Aggregator



JTeam

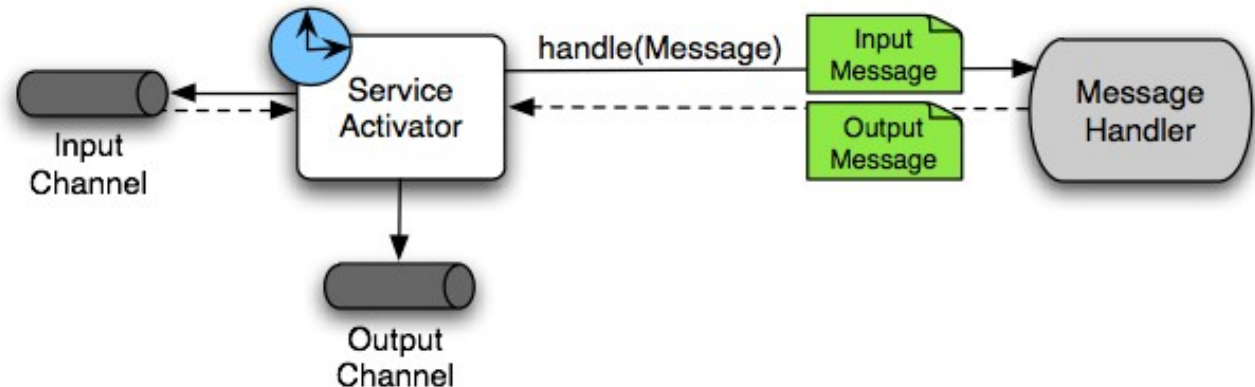
Innovating J2EE Development

Service activator

- Connecting any Spring-managed Object to an input channel.
- Plays the role of a service.

```
<service-activator input-channel="exampleChannel"  
                  output-channel="replyChannel"  
                  ref="somePojo"  
                  method="someMethod"  
>
```

```
/>
```



Channel Adapter

- Connecting a single sender or receiver to a Message Channel.
- Inbound/outbound-channel-adapter
- Out of the box support for JMS, File, HTTP, Web Services and Mail.



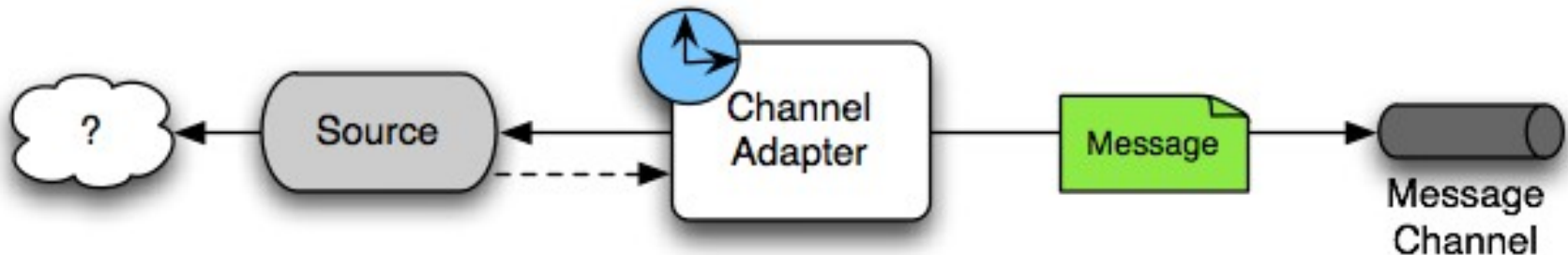
JTeam

Innovating J2EE Development

Channel Adapter

- Inbound-channel-adapter

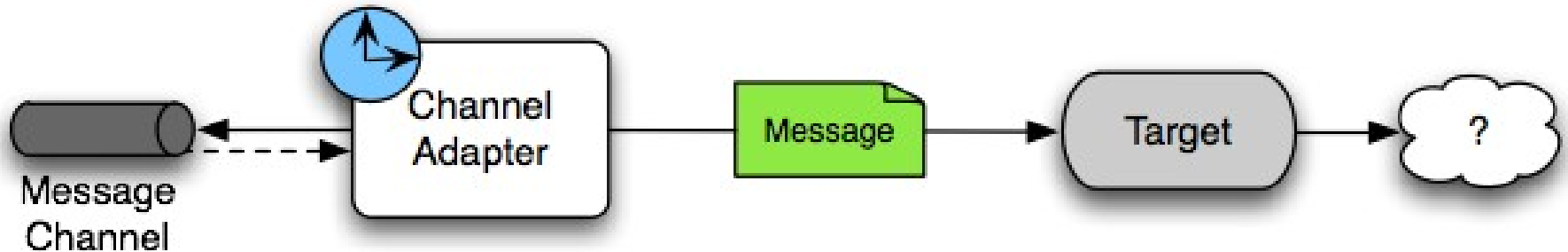
```
<inbound-channel-adapter ref="source1"  
                          method="method1"  
                          channel="channel1">  
  <poller>  
    <interval-trigger interval="5000"/>  
  </poller>  
</inbound-channel-adapter>
```



Channel Adapter

- Outbound-channel-adapter

```
<outbound-channel-adapter ref="source2"  
    method="method2"  
    channel="channel2">  
</outbound-channel-adapter>
```



Message Handler Chain

- Delegating to a chain of other handlers
- Several handlers connected in fixed, linear progression
- Simpler configuration

```
<chain input-channel="input" output-channel="output">  
  <filter ref="someSelector" throw-exception-on-rejection="true"/>  
  <header-enricher error-channel="customErrorChannel">  
    <header name="foo" value="bar"/>  
  </header-enricher>  
  <service-activator ref="someService" method="someMethod"/>  
</chain>
```



Configuration

- Namespace support

```
<beans xmlns="http://www.springframework.org/schema/beans"  
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
  xmlns:integration="http://www.springframework.org/schema/integration"  
  xmlns:file="http://www.springframework.org/schema/integration/file"  
  xsi:schemaLocation="http://www.springframework.org/schema/beans  
    http://www.springframework.org/schema/beans/spring-beans-2.5.xsd  
    http://www.springframework.org/schema/integration  
    http://www.springframework.org/schema/integration/spring-integration-1.0.xsd  
    http://www.springframework.org/schema/integration/file  
    http://www.springframework.org/schema/integration/file/spring-integration-file-1.0.xsd">  
  ...  
</beans>
```

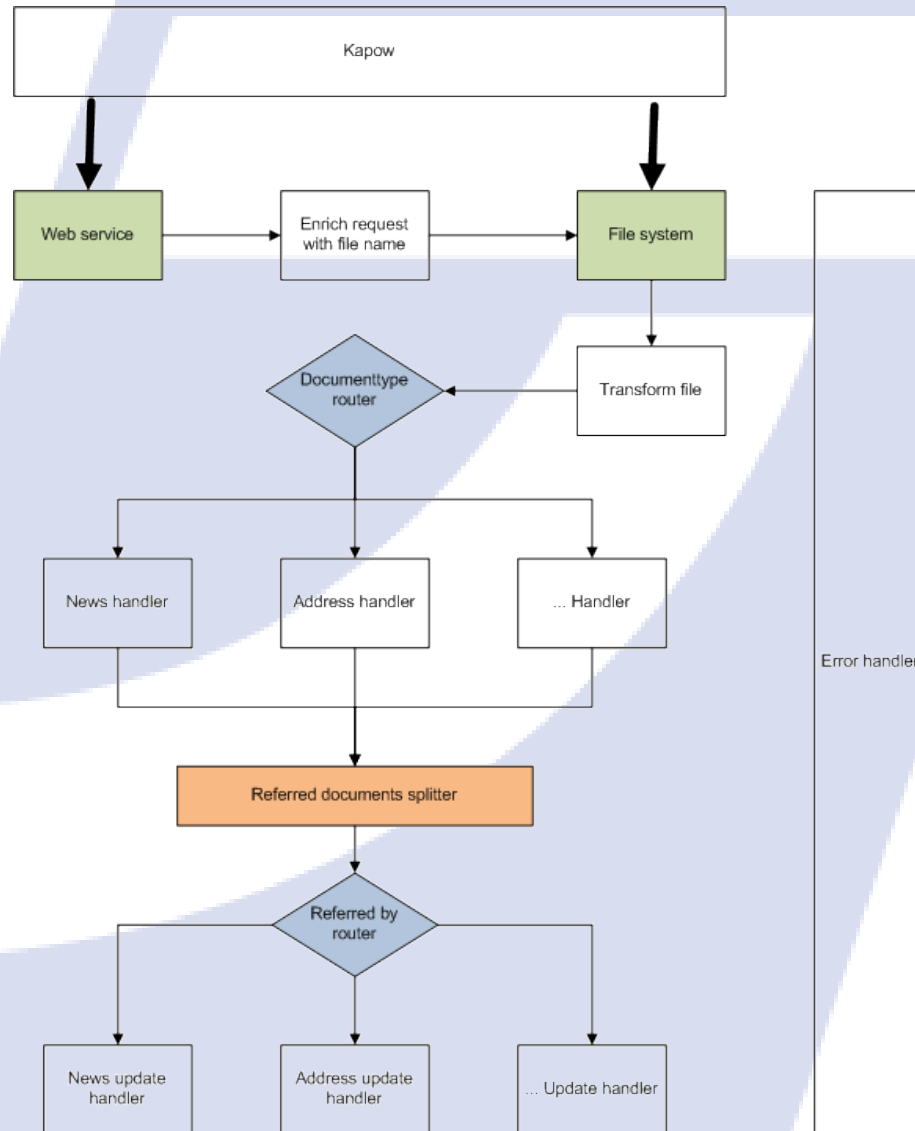


Configuration

- Annotation support
 - @MessageEndpoint
 - @Transformer
 - @Router
 - @Splitter
 - @Aggregator
 - @ServiceActivator
 - @ChannelAdapter
 - @Headers



Project



JTeam

Innovating J2EE Development

Web Service

- Use of namespace
- Send a message to a channel upon receiving a Web Service invocation
- Inbound-gateway

```
<ws:inbound-gateway id="marshallingGateway"  
    request-channel="requestChannel"  
    marshaller="someMarshaller"  
    unmarshaller="someUnmarshaller"/>
```



Web Service

- Provides more flexibility for other applications to connect with the importer tool.
- Prevents the file being read before it is completely written to the file system.
- Returns response message.



JTeam

Innovating J2EE Development

Reading files

- Used the inbound-channel-adapter
- Added filter, so only XML files are read

```
<bean id="patternMatchingFileListFilter"  
class="org.springframework.integration.file.PatternMatchingFileListFilter">  
    <constructor-arg value=".*\.xml"/>  
</bean>
```

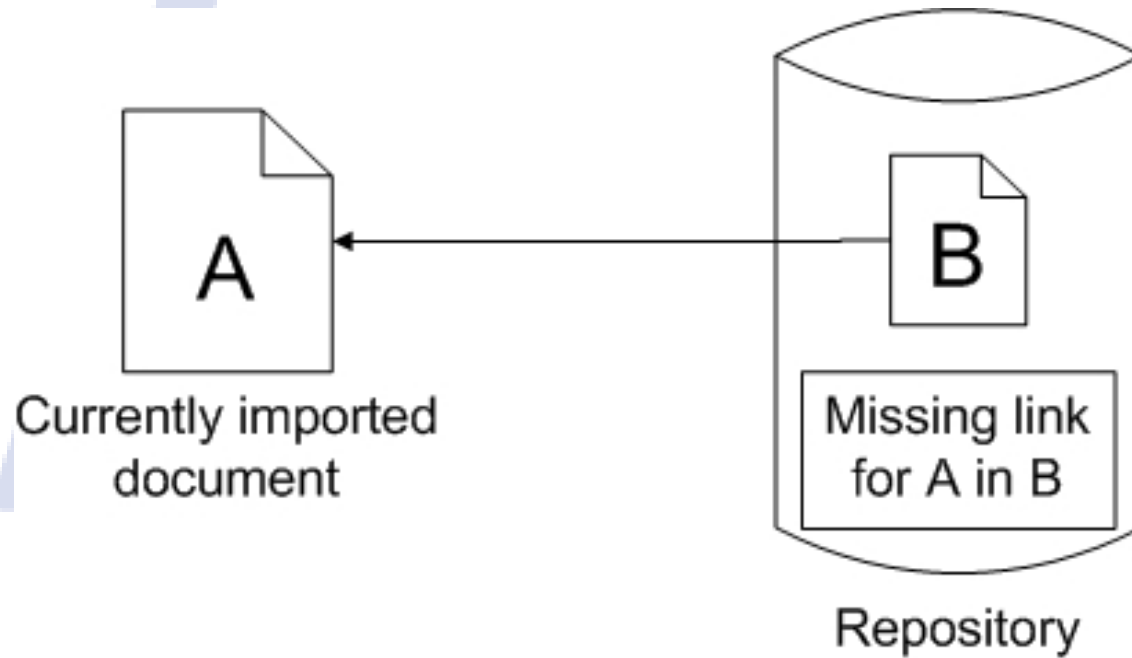


JAXB Un/Marshaller

- Spring provides interfaces
 - Marshaller
 - Unmarshaller
- JAXBMarshaller implement both in one class.
- XSD files
 - To validate XML
 - Generate objects



Update process



JTeam

Innovating J2EE Development

Error handling

- Critical error – write file to other directory
- Put the payload on a **channel**
- **Transformed** payload to xml
- Used the **outbound-channel-adapter** to write xml to directory.



Error handling

- Non critical errors
- Introduced error collector
- Collect errors during the whole process of importing one document
- Mail errors after document is imported



Demo



JTeam

Innovating J2EE Development

Conclusion

- Simple configuration
- Easy to understand
- Well documented
- Proven to work well in a relatively large project.
- Support in other projects (Axon framework)



JTeam

Innovating J2EE Development

Questions



JTeam

Innovating J2EE Development