

Message Processing

Michael Hoffman
@mhi_inc



pluralsight 
hardcore dev and IT training

Message Processing

Messages

- ✓ Message EIPs
- ✓ Camel Message Structure
- ✓ Message Transformation

Case Study Project Updates

- ✓ Code Message Translation
- ✓ Test Message Translation

Message EIP

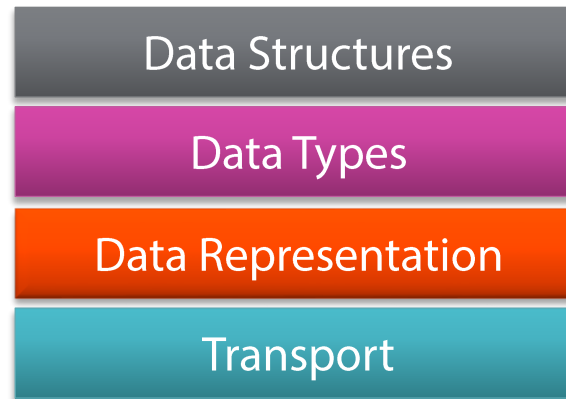
- **Structure**
 - Header
 - Body
- **Types**
 - Command Message
 - Document Message
 - Event Message
- **Format**
 - Canonical Data Model

Camel Message Model

- **Message**
 - Header
 - Body
 - Fault
- **Exchange**
 - Container for the inbound and outbound messages
 - Message Exchange Pattern (MEP)

Message Translation

- Message Translator EIP
- Translation Layers



Camel Message Translator

- **DSL transform method**
- **DSL process method**
 - Common Processor interface for processing
 - Process method, accepts Exchange
- **DSL bean and beanRef method**
 - POJO, Spring Bean

Camel DSL Transform

```
return new RouteBuilder() {  
    public void configure() throws Exception {  
        from("direct:foo").  
            transform(body().append("Hello World!"));  
    }  
}
```

Camel DSL Process

```
return new RouteBuilder() {  
    public void configure() throws Exception {  
        from("direct:foo").  
            process(new Processor() {  
                public void process(Exchange exchange) {  
                    exchange.getIn().setBody(  
                        exchange.getIn().getBody(String.class) +  
                        "Hello World!"  
                    );  
                }  
            });  
    }  
}
```

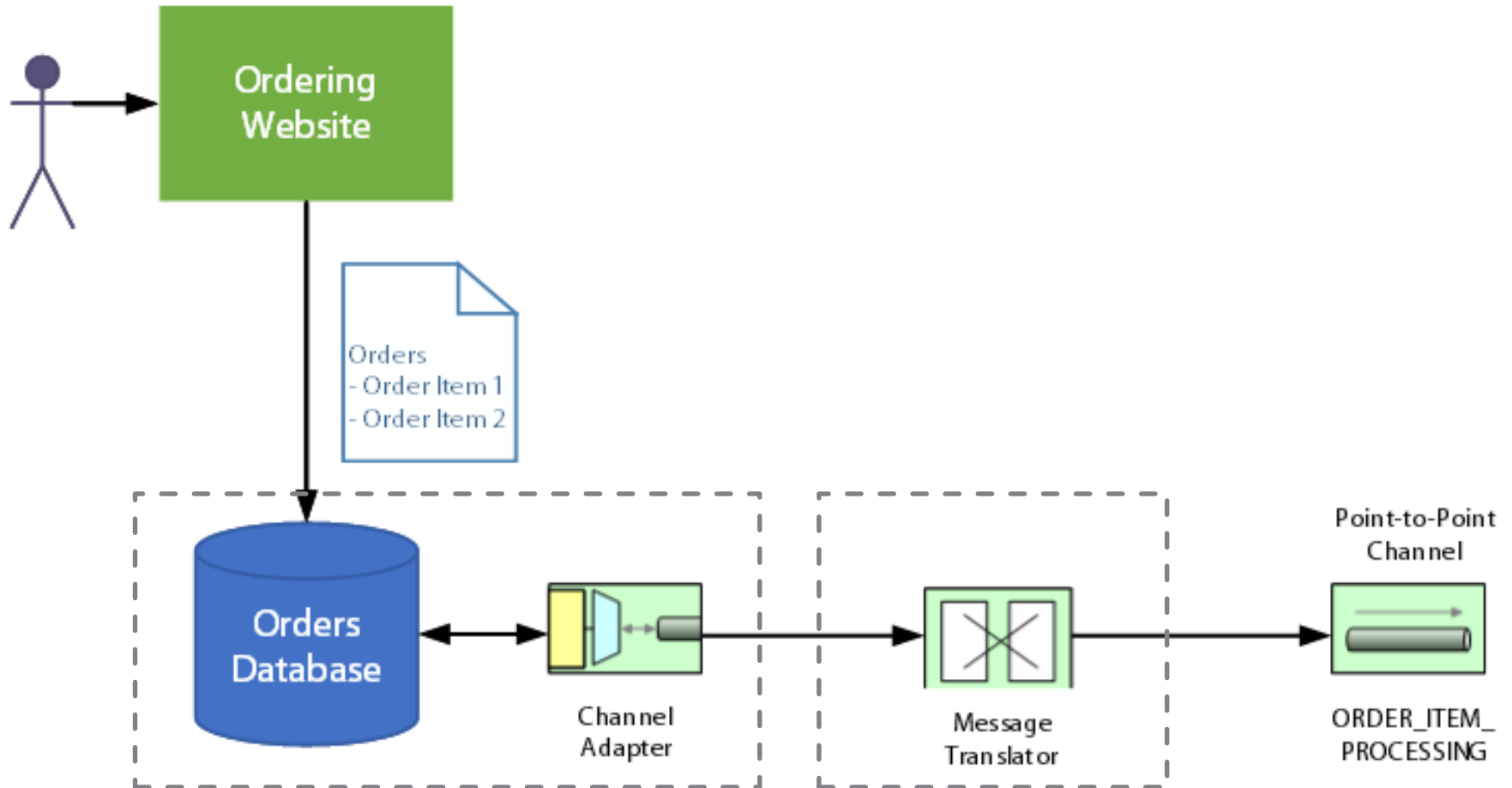

Camel DSL BeanRef

```
return new RouteBuilder() {  
    public void configure() throws Exception {  
        from("direct:foo").  
        beanRef("myBeanName", "doTranslationMethod");  
    }  
}
```

Bean Integration

- **Registry**
 - Access beans from the Spring Context
- **Bean Integration Types**
 - Annotation-based Spring bean routing
 - Spring remoting
 - Bean component
- **Bean Binding**
 - Automatic binding
 - Method parameter definition in Bean Component URI
 - Method signature annotation

Case Study – Order Item Processing Route



Message Format

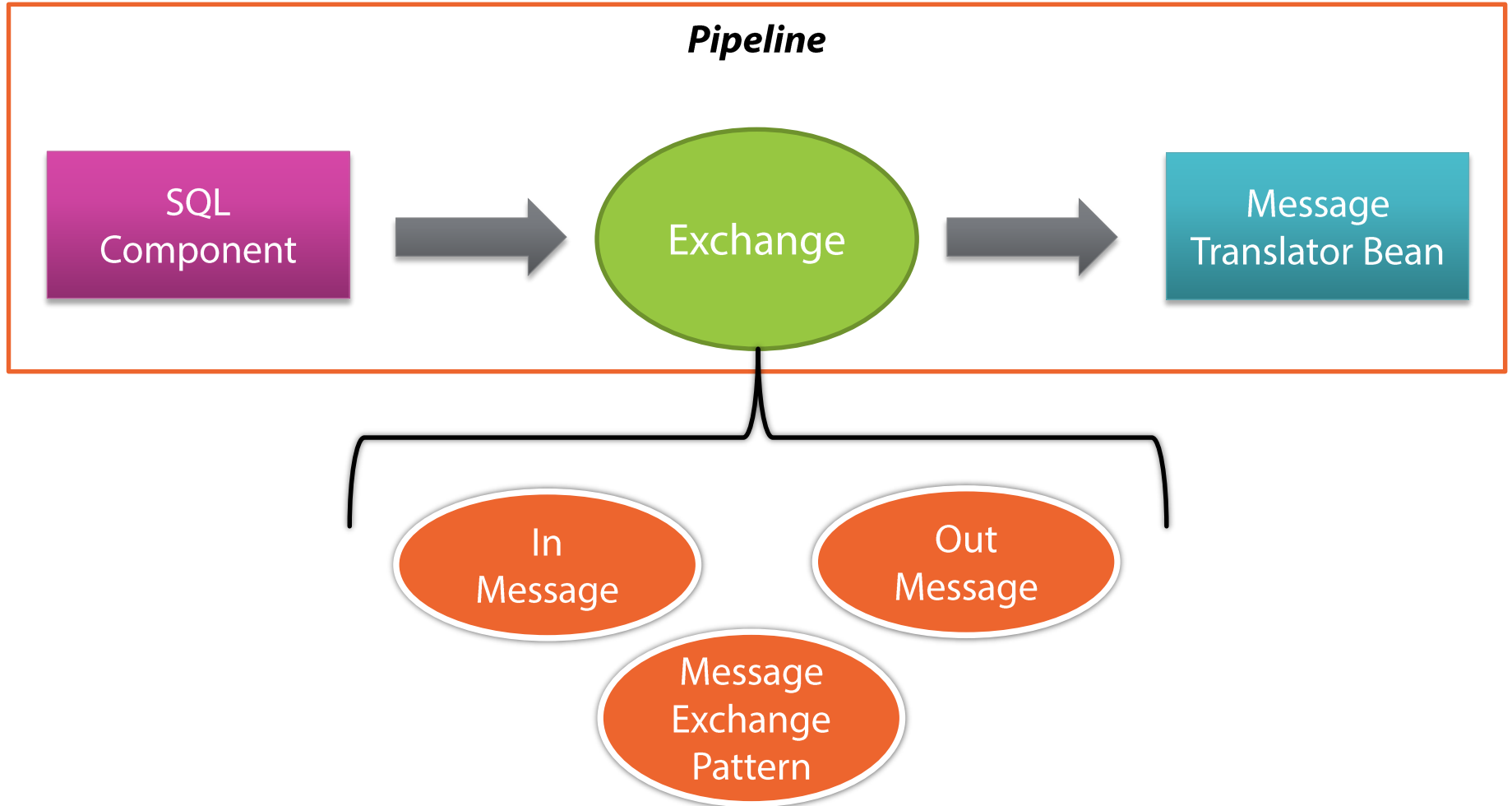
- Discovered as part of routing design
- Based on consumer
- Camel type conversion

```
[INFO] com.pluralsight.orderfulfillment.order -  
Exchange[  
  ExchangePattern: InOnly,  
  BodyType:  
    org.springframework.util.LinkedCaseInsensitiveMap,  
  Body: {id=2}  
]
```

Document Message

- Sends a data structure as a message
- Point-to-Point Channel
- Message type

Message Processing



Exchange In Message

- **Generated by consumer endpoint**
- **Travels from consumer endpoint to producer endpoint**
- **Message type**

Exchange Out Message

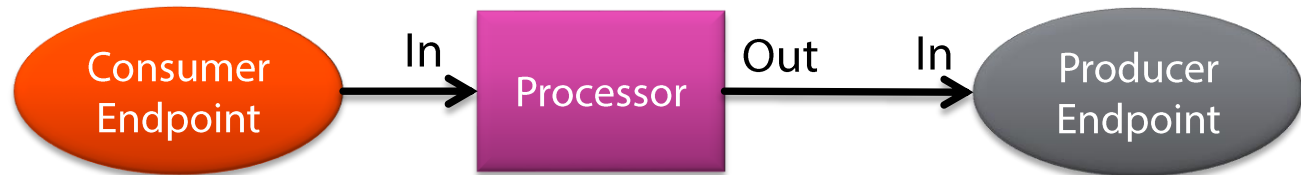
- Generated as a result of processing
- Travels from producer endpoint to consumer endpoint
- Message type

Message Exchange Pattern

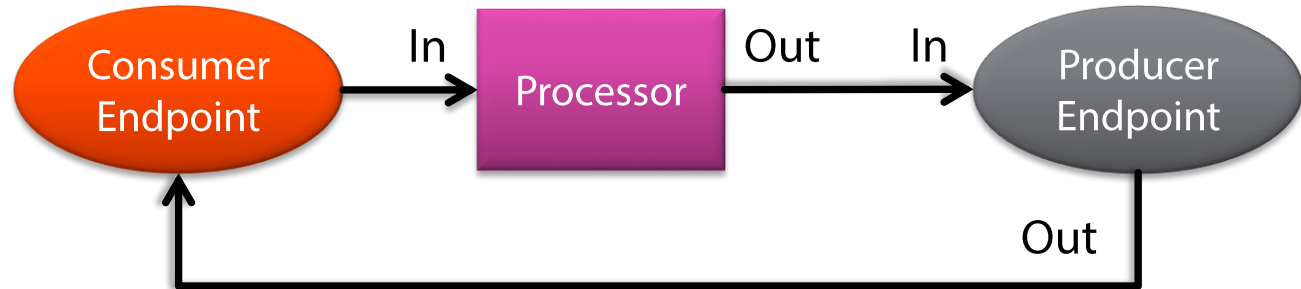
- **Pattern types**
 - InOnly
 - InOut
- **Determined by the consumer endpoint**
- **Realized by the producer endpoint**
- **Configurable through DSL**

Pipeline

In-Only MEP



In-Out MEP



Summary

Messages

Camel Route
Implementation

