Apache Camel

Systémová Integrace

Pavel Macík
(pmacik@redhat.com)

24.9.2015

Co nás čeká

- Představení Apache Camel
- Komponenty Camelu
- Integrační návrhové vzory (EIP)
- Logování
- Zpracování vyjímek a chyb
- Testování aplikací pro Camel

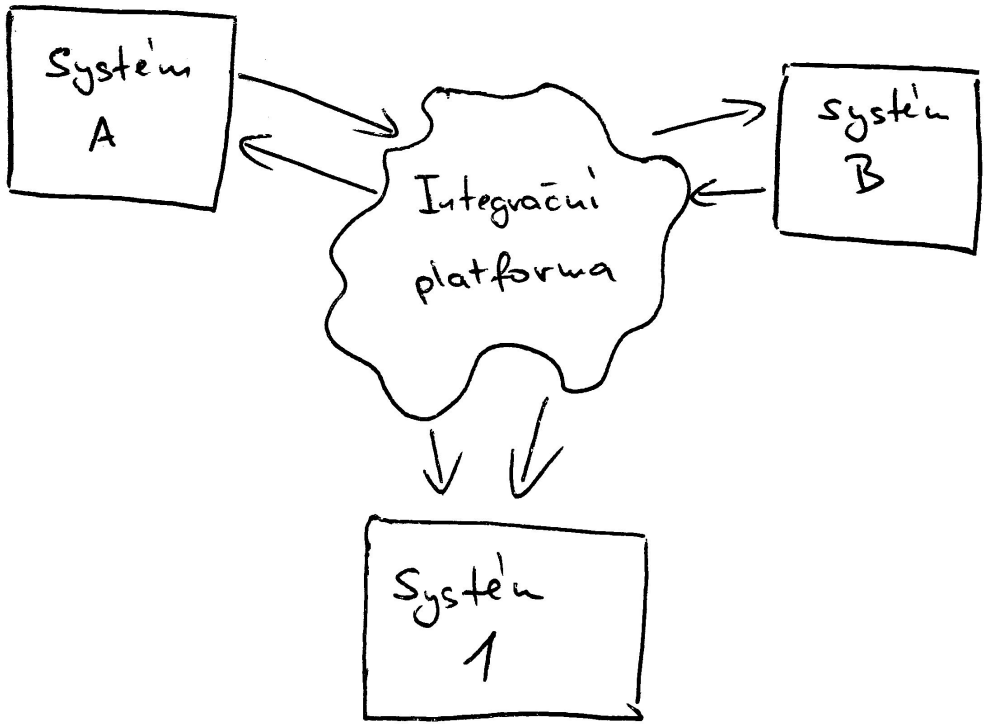
Představení Apache Camel

- Seznamte se - Camel
- Použití
- Route
- Exchange
- Message
- Architektura
- Vývoj aplikací v Camelu

Seznamte se - Camel

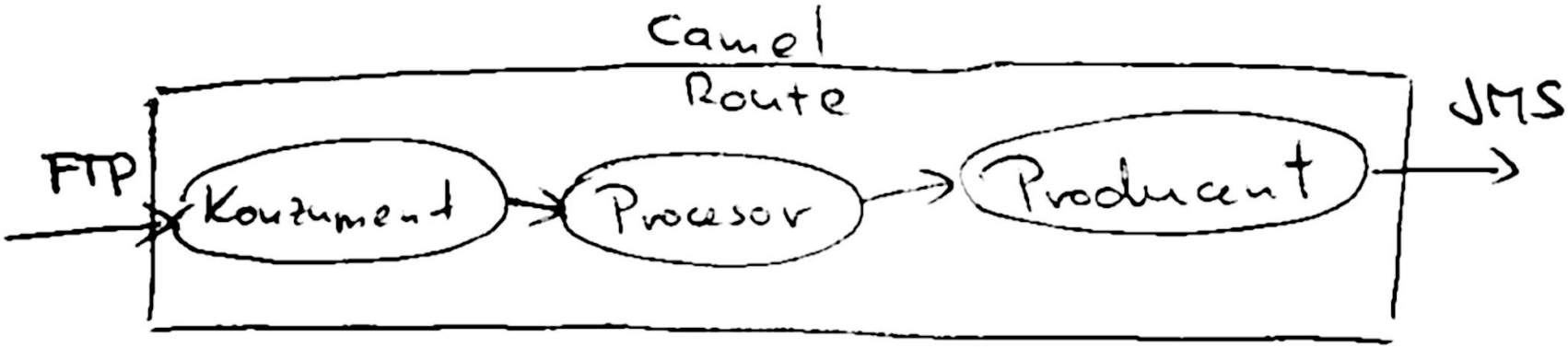
- Open-source systém pro integraci různorodých systémů
- Směrování a transformace zpráv cestujícími mezi systémy
- Implementován v jazyce Java
- Implementuje Integrační návrhové vzory (EIP = Enterprise Integration Patterns)

Použití



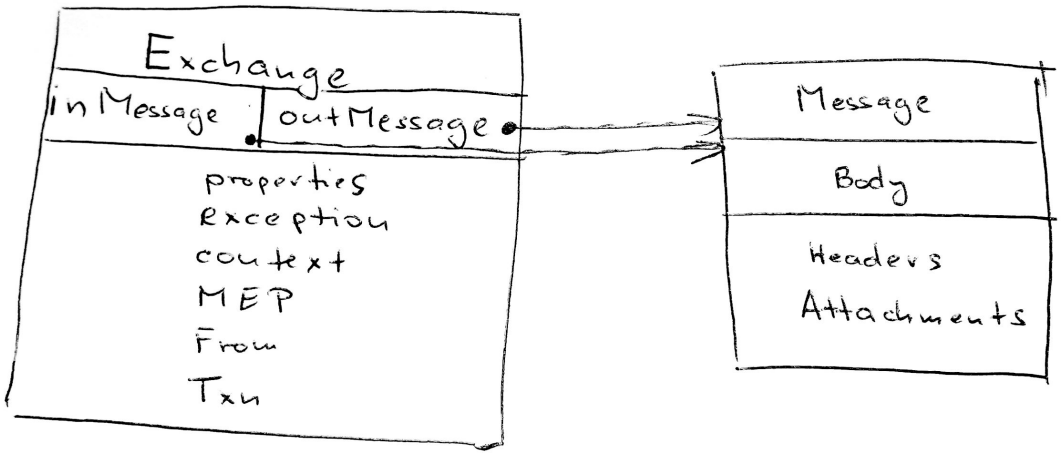
Camel Route

- Definice trasy pro cestu a zpracování zprávy



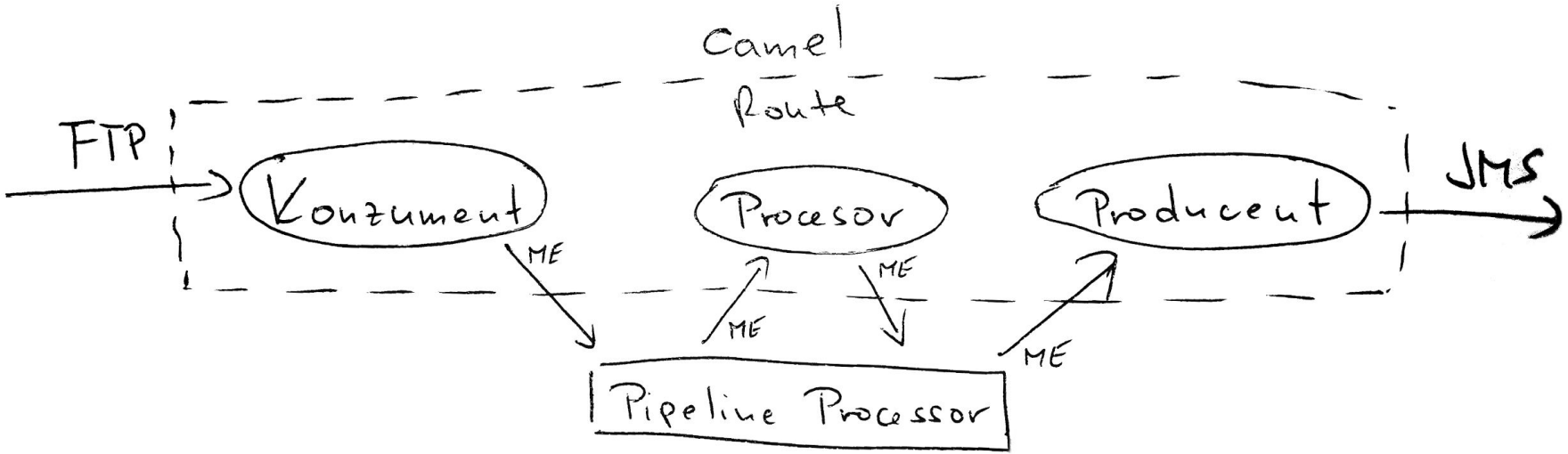
Message Exchange

- Exchange - datová struktura
  - informace o výměně zpráv
  - směrovací informace
  - vlastnosti trasy



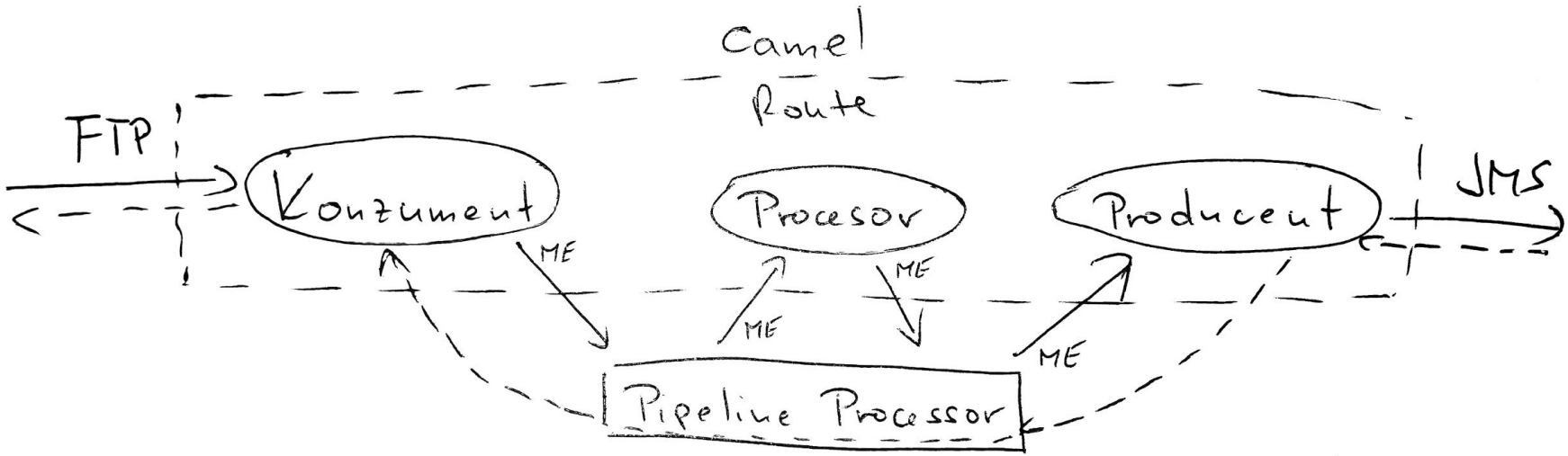
Message Exchange

- InOnly - pouze odeslání zprávy

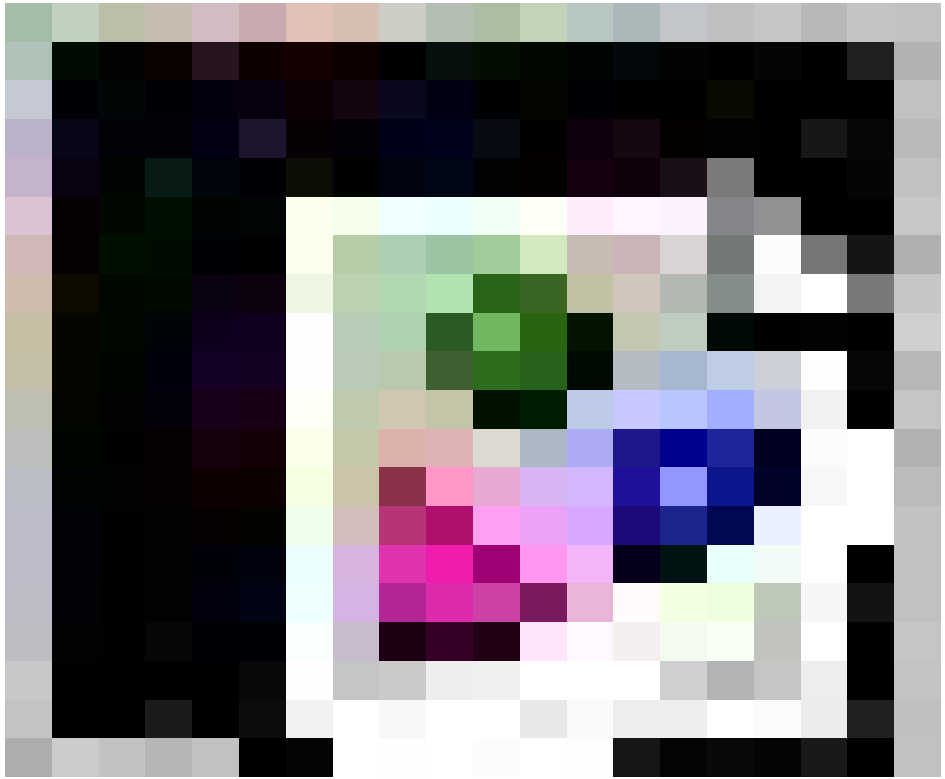


Message Exchange

- InOut - po odeslání zprávy očekávána odpověď



Architektura



Vývoj aplikací v Camelu

- Spring DSL
- Java DSL
- Maven - Demo

Spring DSL

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xsi:schemaLocation="
  http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd
  http://camel.apache.org/schema/spring http://camel.apache.org/schema/spring/camel-spring.xsd">
  <camelContext xmlns="http://camel.apache.org/schema/spring">
    <route>
      <from uri="file:src/data?noop=true"/>
      <choice>
        <when>
          <xpath>/person/city = 'London'</xpath>
          <log message="UK message"/>
          <to uri="file:target/messages/uk"/>
        </when>
        <otherwise>
          <log message="Other message"/>
          <to uri="file:target/messages/others"/>
        </otherwise>
      </choice>
    </route>
  </camelContext>
</beans>
```

Java DSL

```
package com.redhat.brq.integration;
import org.apache.camel.builder.RouteBuilder;
public class MyRouteBuilder extends RouteBuilder {
    public void configure() {
        from("file:src/data?noop=true")
            .choice()
                .when(xpath("/person/city = 'London'"))
                    .log("UK message")
                    .to("file:target/messages/uk")
                .otherwise()
                    .log("Other message")
                    .to("file:target/messages/others");
    }
}
```

Maven - Demo

- Spring DSL
  - `mvn archetype:generate -DarchetypeGroupId= org.apache.camel.archetypes -DarchetypeArtifactId= camel-archetype-spring -DgroupId=com.redhat.brq.integration -DartifactId=maven-camel-spring;`

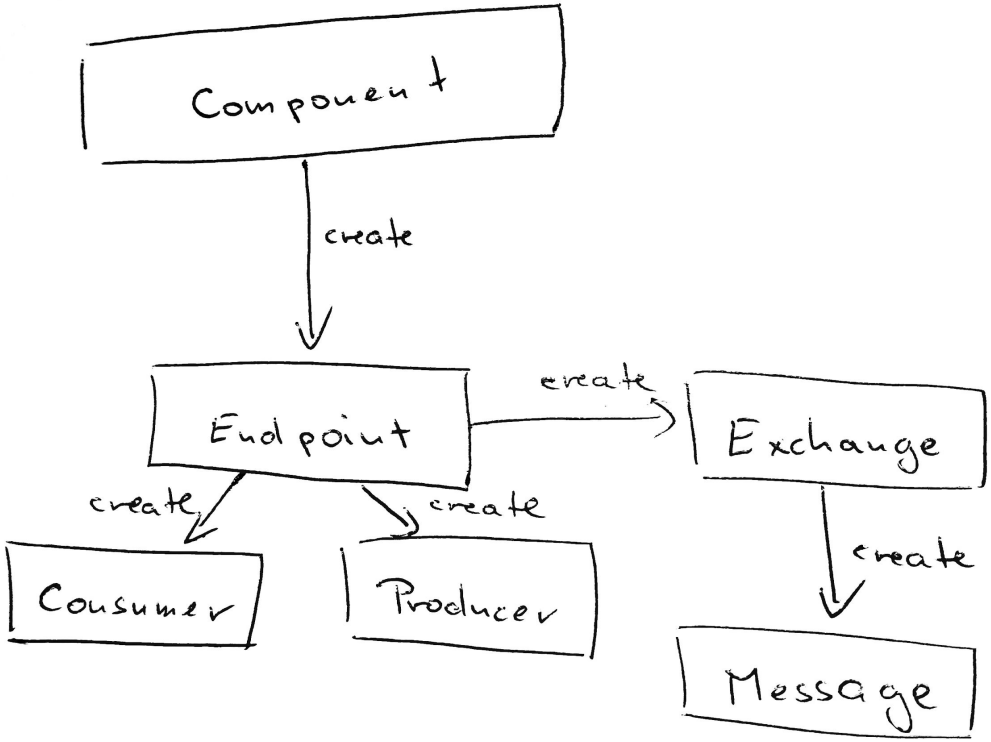
Maven - Demo

- Java DSL
  - `mvn archetype:generate -DarchetypeGroupId= **org.apache.camel.archetypes** -DarchetypeArtifactId= **camel-archetype-java** -DgroupId=com.redhat.brq.integration -DartifactId=maven-camel-java;`

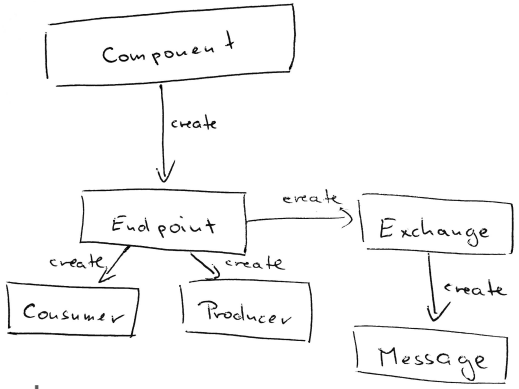
Komponenty Camelu

- Model komponenty
- Užitečné komponenty
  - Bean - Java objekt
  - File - soubory
  - Jetty - HTTP(s)
  - Restlet - JAX-RS
    - Rest DSL
  - ActiveMQ
    - MQTT
    - JMS

Model komponenty



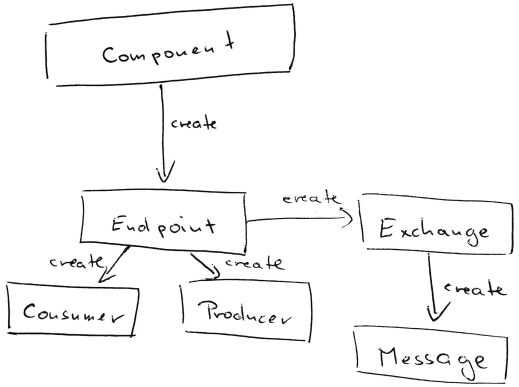
Model komponenty



- Komponenta
  - továtna na koncové body (endpointy)
- Koncový bod
  - reprezentuje začátek nebo konec kanálu zprávy z/do trasy Camelu
  - popsán prostřednictvím URI
    - `schema:context/path?options`
      - schema = identifikuje komponentu
      - context/path = identifikuje umístěná zdroje nebo cíle zprávy
      - options = nastavení vlastností komponenty

Model komponenty

- Konzument (from)
  - zdroj, z kterého je zpráva přijata
- Producent (to)
  - cíl, do kterého je zpráva odeslána



Komponenta Bean

- Umožňuje použít metody Java objektů pro zpracování zpráv
- URI pattern:
  - `bean:beanId[?options]`
- Užitečné vlastnosti
  - method - jméno metody, která se má zavolat

Komponenta Bean

- Java objekt
  - ```
    public class MyBean {
        public void process(String msg) {
            System.out.println("Zprava: " + msg);
        }
    }
    ```
- Camel
  - ```
    from("file:src/data?noop=true")
        .bean(new MyBean(), "process");
    ```

- Více informací na http://camel.apache.org/bean.html

Komponenta File

- Umožňuje přístup k souborovému systému
- URI pattern:
  - `file:directoryName[?options]`
- Užitečné vlastnosti
  - noop - pokud je true, soubory po zpracování nejsou přemístěny ani smazány
  - include - regulární výraz omezující jména souborů, které se mají zpracovat
  - exclude - regulární výraz označující soubory, které se mají ignorovat
- Více informací na http://camel.apache.org/file2.html

Komponenta Jetty

- Poskytuje koncové body založené na HTTP
- URI pattern:
  - `jetty:http://hostname[:port][/resourceUri][?options]`
- Více informací na http://camel.apache.org/jetty.html

Komponenta Restlet

- Poskytuje koncové body založené na REST HTTP
- URI pattern:
  - `restlet:restletUrl[?options]`
  - `restletUrl = protocol://hostname[:port][/resourcePattern]`

Komponenta Restlet

- Restlet

```
public void configure()  {
   from("restlet:http://localhost:8080/rest-end?restletMethods=POST" ).log("POST: ${body}");
}
```

- REST DSL:

```
public void configure()  {


  rest("/rest-end").consumes("text/plain").produces("text/plain")
        .post().to("direct:post");

  from("direct:post").log("POST: ${body}");
}
```

Komponenta ActiveMQ

- ActiveMQ
  - Systém poskytující kompletní služby týkající se zpracování zpráv
  - Poskytuje různá rozhraní, např.
    - MQTT
    - JMS
- Více informací na http://activemq.apache.org/

Komponenta ActiveMQ - JMS

- URI pattern:
  - `activemq:[queue:|topic:]destinationName`
- Více informací
  - http://camel.apache.org/activemq.html
  - http://camel.apache.org/jms.html

Komponenta ActiveMQ - MQTT

- MQTT
  - Protokol pro komunikaci mezi různými zařízeními prostřednictvím zasíláním zpráv
  - de-facto standard pro IoT
- Zapnutí MQTT u ActiveMQ brokeru
  - `<transportConnector name="mqtt" uri="mqtt://localhost:1883" />`
- URI pattern:
  - `mqtt://name[?options]`
- Více informací na http://camel.apache.org/mqtt.html

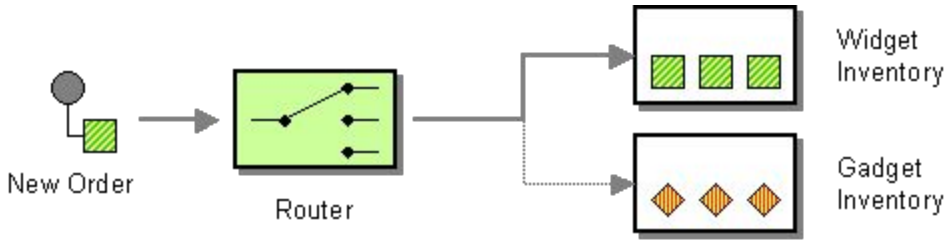
Integrační návrhové vzory (EIP)

EIP = Enterprise Integration Patterns

- Content based routing
- Splitter
- Multicast
- Aggregator
- Transformace zpráv
- http://camel.apache.org/eip.html
- http://www.enterpriseintegrationpatterns.com/

Integrační návrhové vzory

- Content based routing
  - Směrování zpráv na základě jejího obsahu



- Více informací na http://camel.apache.org/content-based-router.html

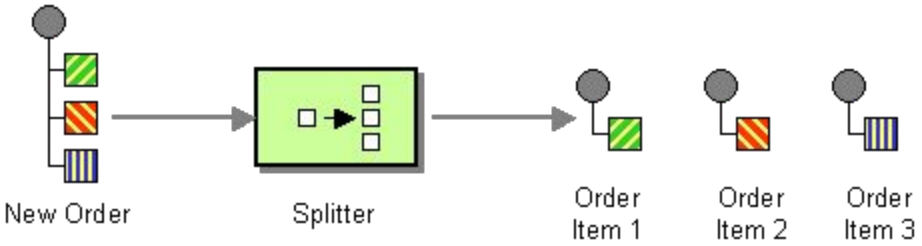
Integrační návrhové vzory

- Content based routing

```
from("file:src/data?noop=true")
    .choice()
        .when(xpath("/person/city = 'London'"))
            .log("UK message")
            .to("file:target/messages/uk")
        .otherwise()
            .log("Other message")
            .to("file:target/messages/others");
```

Integrační návrhové vzory

- Splitter
  - rozdělí zprávu na menší části a ty pak posílá samostatně



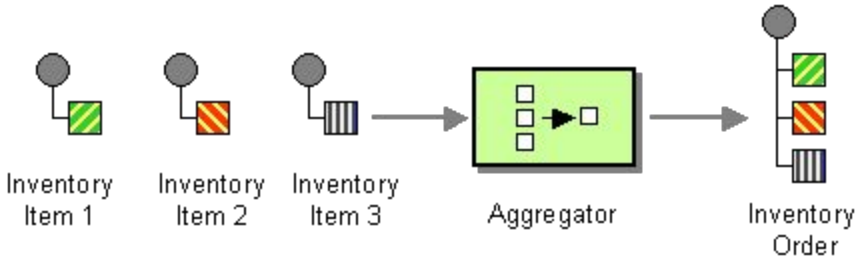
- Více informací na http://camel.apache.org/splitter.html

Integrační návrhové vzory

- Multicast
  - Umožňuje rozeslat stejnou zprávu na více koncových bodů a v každém z nich ji zpracovat jinak.
- Více informací na http://camel.apache.org/multicast.html

Integrační návrhové vzory

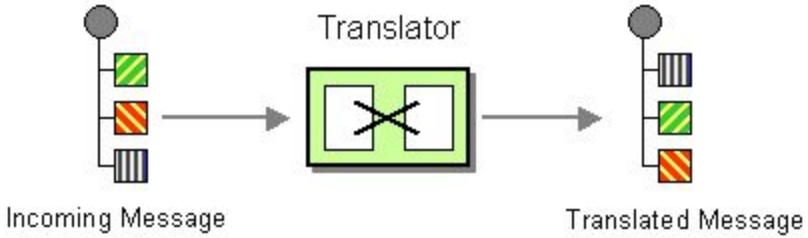
- Aggregator
  - Slučuje více zpráv do jedné na základě agregační strategie



- Více informací na http://camel.apache.org/aggregator.html

Integrační návrhové vzory

- Transformace zpráv
  - Transformuje příchozí zprávu a pošle ji dál



Integrační návrhové vzory

- Transformace zpráv - Simple
  - ```
    from("file:src/data?noop=true")
                .transform(simple("Odpoved: ${body}")).to("file:target");
    ```
    ```

Integrační návrhové vzory

- Transformace zpráv - CSV
  - Umožňuje transformaci záznamů z CSV souboru do Java objektů a zpět.
- Java objekt
  - ```
    @CsvRecord(separator=";", crlf="UNIX")
    public Class Person {
        @DataField(pos = 1)
        private String name;
        @DataField(pos = 2)
        private String city;
    }
    ```
- Camel
  - ```
    BindyCsvDataFormat bindy = new BindyCsvDataFormat("com.redhat.brq.integration");
    from("file:src/data?noop=true")
        .unmarshal(bindy).log("${body}");
    ```

Integrační návrhové vzory

- Transformace zpráv - CSV
  - Více informací na http://camel.apache.org/bindy.html

Integrační návrhové vzory

- Transformace zpráv - JAXB (XML)
  - Umožňuje transformaci XML souboru do Java objektu a zpět.
- Java objekt
  - ```
    @XmlRootElement(name = "person")
    @XmlAccessorType(XmlAccessType.FIELD)
    public class Person {
        @XmlAttribute
        private String user;
        @XmlElement
        private String firstName;
        @XmlElement
        private String lastName;
        @XmlElement
        private String city;
    }
    ```
    ```

Integrační návrhové vzory

- Transformace zpráv - JAXB (XML)
- MyBean
  - ```
    public class MyBean {
        public void process(Person p) {
            System.out.println(p.toString());
        }
    }
    ```
- Camel
  - ```
    JaxbDataFormat jaxb = new JaxbDataFormat("com.redhat.brq.integration");
    from("file:src/data?noop=true")
        .unmarshal(jaxb)
        .bean(new MyBean(), "process");
    ```

Logování v Camelu

- Komponenta, která umožňuje logovat na různých úrovní důležitosti přímo z jednotlivých tras
- URI pattern:
  - `log:loggingCategory[?options]`
- Více informací na http://camel.apache.org/log.html

Zpracování vyjímek a chyb

- Camel
  - `onException(MyException.class).to("file:target/exceptions");`
- Více informací na http://camel.apache.org/exception-clause.html

Testování aplikací pro Camel

- CamelTestSupport (JUnit)
- PaxExam

Děkuji za pozornost