THE DEVELOPER'S CONFERENCE

TRILHA JAVA

Sábado 12/10

UNICAP
Recife / PE

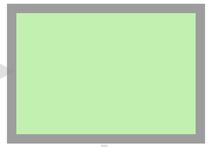
padrões essenciais de mensageria para integração de sistemas

helderdarocha
helder@argonavis.com.br

Por que integrar sistemas?

- Aplicações interessantes raramente vivem **isoladas**
  - **Sincronizar** emails, calendários, etc.
  - **Vincular** portal a um aplicativo de controle de estoque
  - **Integrar** com serviços e dados na nuvem & IoT
- **Comunicação** entre sistemas incompatíveis
- Não reinventar a roda
  - Reusar serviços que funcionam bem
  - Aproveitar pontos de integração

Desafios das soluções de integração



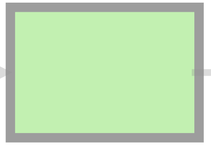
Redes **não são confiáveis**

Redes são **lentas**

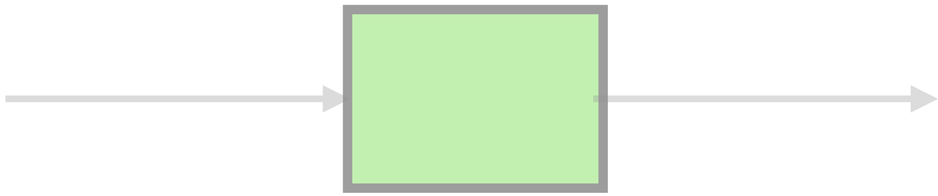
Aplicações são **diferentes**

Aplicações **mudam**

Soluções são **complexas**



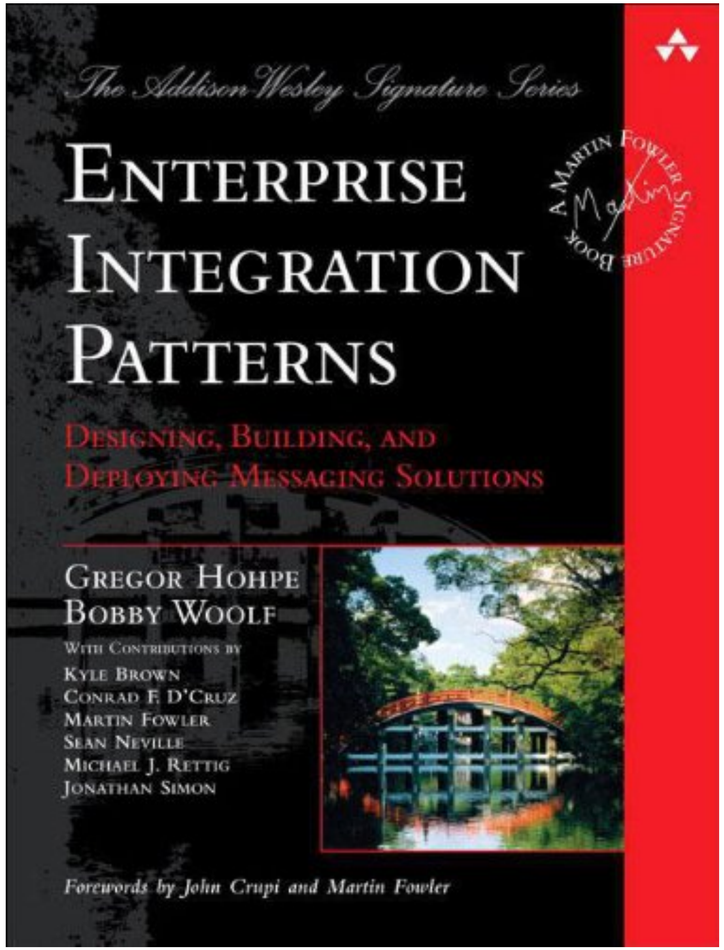
Padrões de design



- **Abstração** de alto nível
  - Nome / Icone
  - Contexto
  - Problema / Solução / Exemplo
  - Vocabulário
  - Notação
  - Padrões relacionados
  - Conseqüências da aplicação do padrão

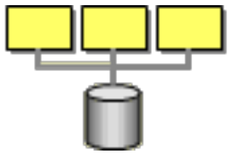
Padrões de integração

- 65 padrões de integração de sistemas
  - Aplicações em Java, .NET e outras plataformas
- 62 focam em soluções de **mensageria**
- Foco: minimizar o **acoplamento** entre componentes



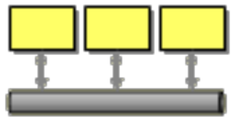
4 estratégias básicas de **integração**

Transferir **arquivos** entre aplicações (1)



Compartilhar um **banco de dados** (2)

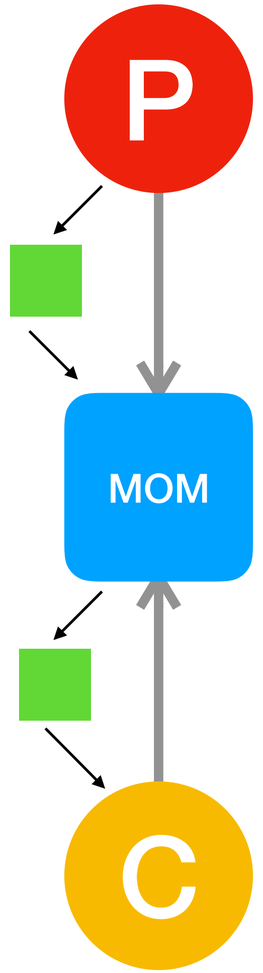
Usar interfaces + proxies **RPC** (3)



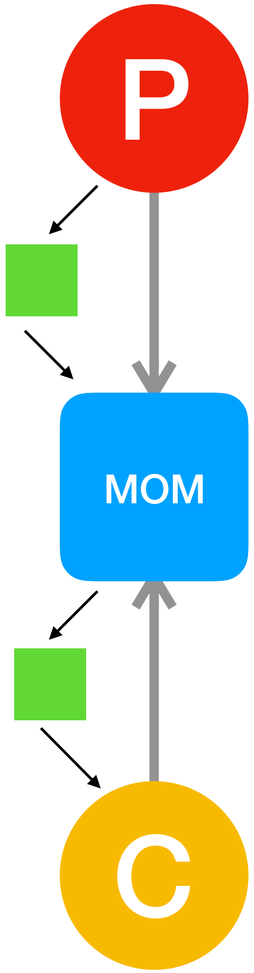
Usar **mensageria** (4)

O que é mensageria?

- Comunicação entre máquinas
  - Através de **mensagens** enviados em **canais** (filas) compartilhados entre máquinas
  - Remetente, **produtor** de mensagens
  - Destinatário, **consumidor** de mensagens
- Mensagem
  - Estrutura de dados (objeto, string, tipo, bytes)
  - **Cabeçalho**, metadados
  - Corpo, **payload**

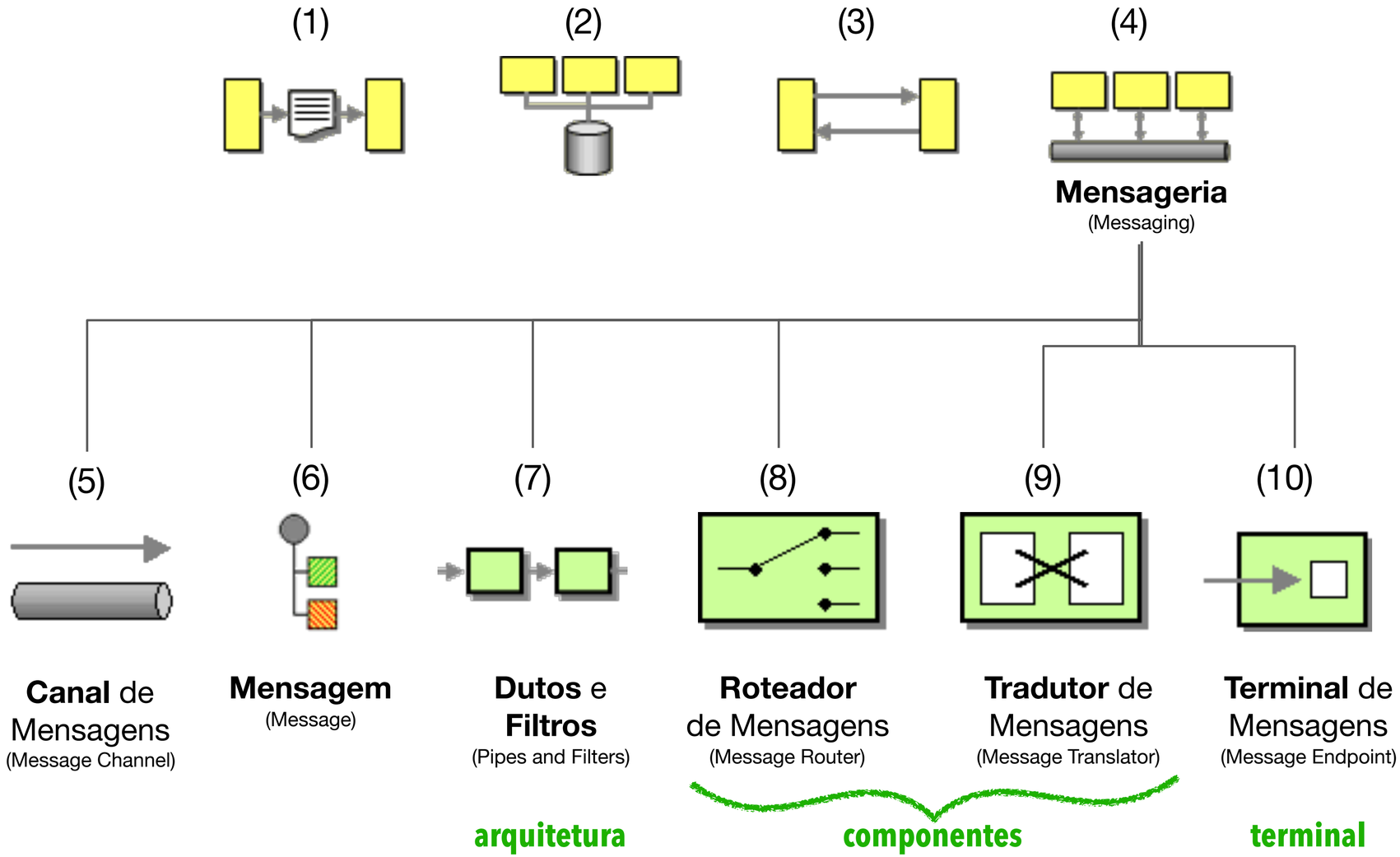


O que é um sistema de mensageria?



- Message Oriented Middleware
  - Mediator pattern (GoF) entre consumidores e produtores
  - Administra o sistema (**canais** e **conexões**)
- Etapas
  - **Criar** – produtor cria a mensagem e preenche com dados
  - **Enviar** – produtor adiciona mensagem a um canal
  - **Entregar** – sistema de mensageria transfere mensagem de uma máquina para a outra
  - **Receber** – consumidor lê a mensagem do canal
  - **Processar** – consumidor extrai os dados da mensagem

Padrões de Mensageria (4)

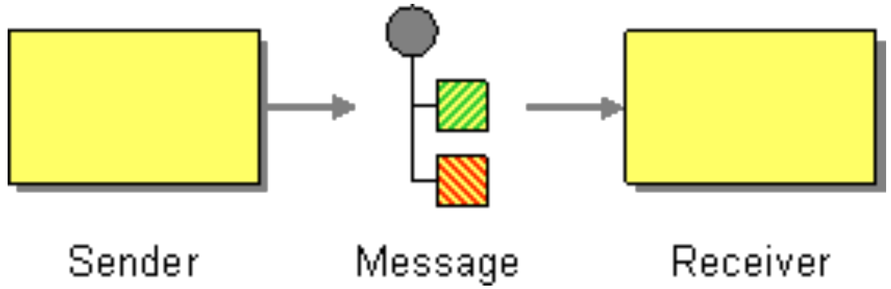


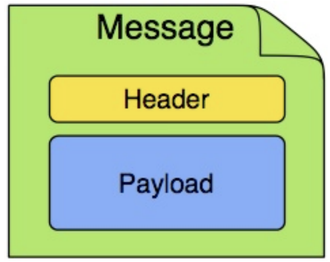
Message

1

Problema “Como é que duas aplicações conectadas por um canal de mensagens podem trocar informação?”

Solução “Empacote a informação dentro de uma **Mensagem (Message)**, um registro de dados que o sistema de mensageria pode transmitir através de um canal de mensagens.”





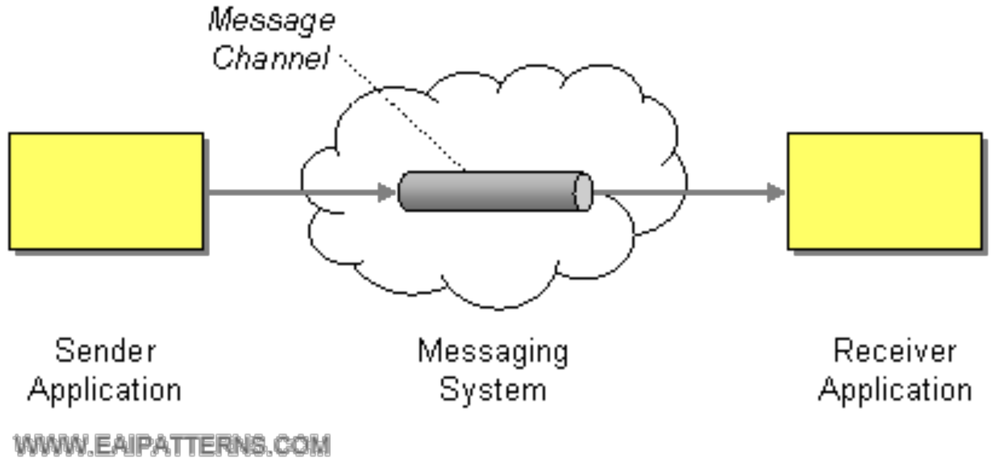
"Envelope"

Message Channel

2

Problema "Como pode uma aplicação comunicar-se com outra aplicação usando mensageria?"

Solução "Conectar as aplicações usando um **Canal de Mensagens (Message Channel)**, onde uma aplicação grava informação no canal e a outra lê do canal."



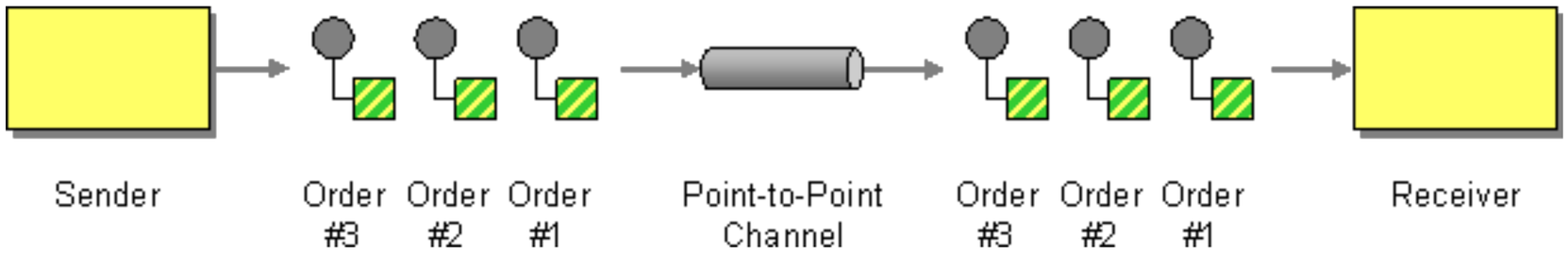
Fonte: eaipatterns.com

Point-to-point channel

3

Problema "Como o remetente pode ter certeza que apenas um destinatário irá receber a mensagem ou executar um comando?"

Solução "Envie a mensagem através de um **Canal Ponto-a-Ponto**, que garante que apenas um destinatário irá receber a mensagem."



Imagens: eaipatterns.com

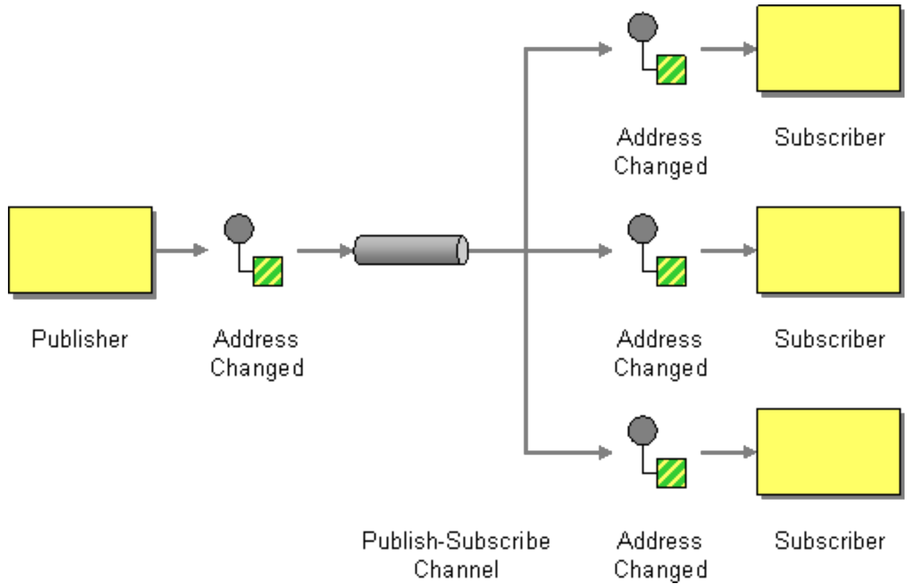
Tipo de Message Channel

Publish-subscribe channel

4

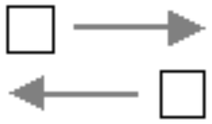
Problema "Como pode um remetente transmitir um evento a todos os destinatários interessados?"

Solução "Envie o evento para um **Canal de Difusão**, que entrega uma cópia do evento a cada destinatário."



Imagens: eaipatterns.com

Tipo de Message

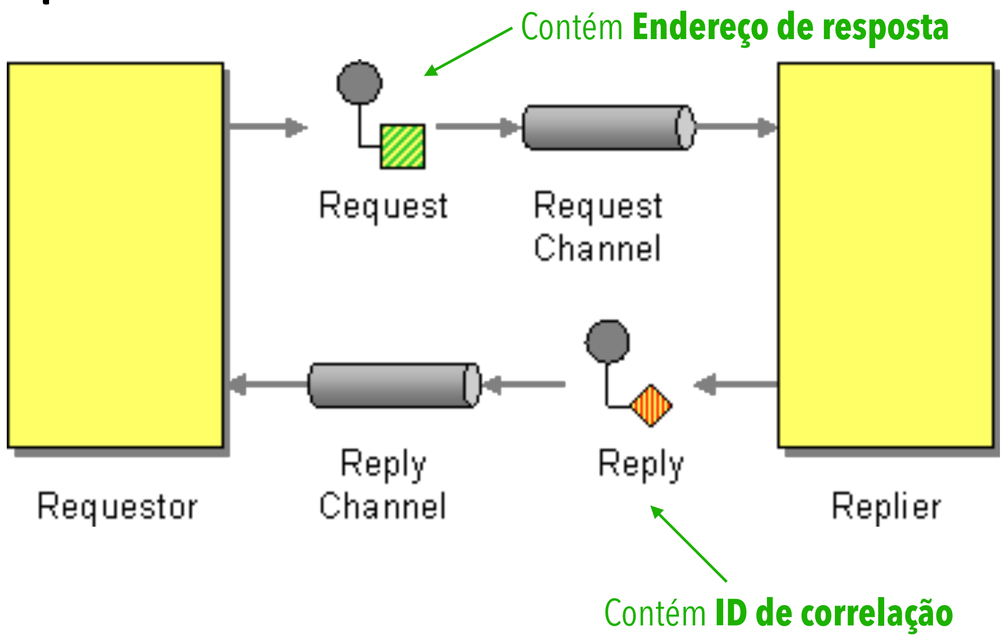


Request-Reply

5

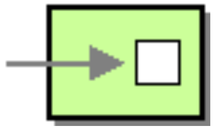
Problema "Quando uma aplicação envia uma mensagem, como ela pode obter uma resposta do servidor?"

Solução "Envie um **par de mensagens Requisição-Resposta**, cada uma no seu próprio canal"



Exchange
:getIn()
:getOut()

Imagens: eaipatterns.com

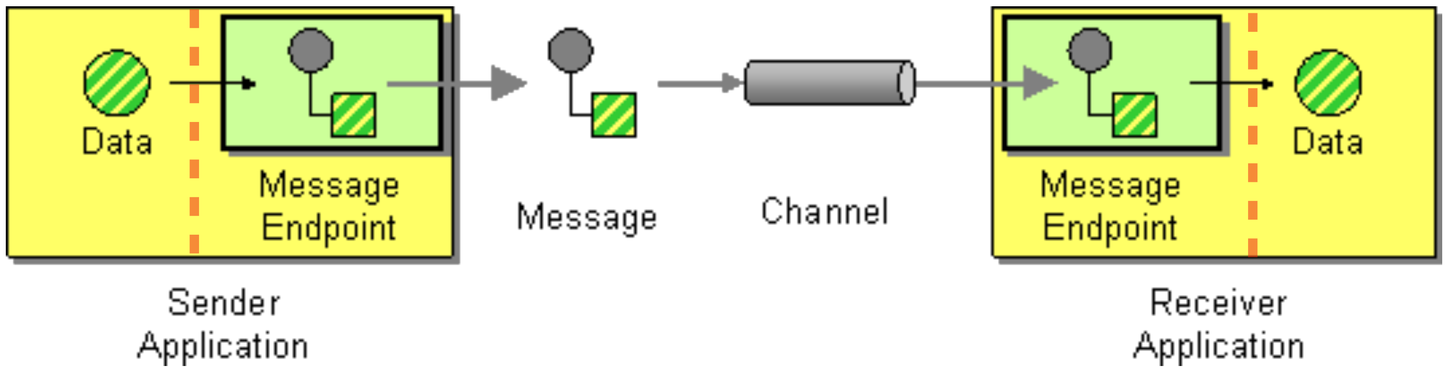


Message Endpoint

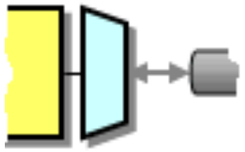
6

Problema "Como pode uma aplicação conectar-se a um canal de mensageria para enviar e receber mensagens?"

Solução "Conecte uma aplicaçao a um canal de mensageria usando um **Terminal de Mensagens**, um cliente do sistema de mensageria que a aplicação pode usar para enviar ou receber mensagens."



Imagens: eaipatterns.com



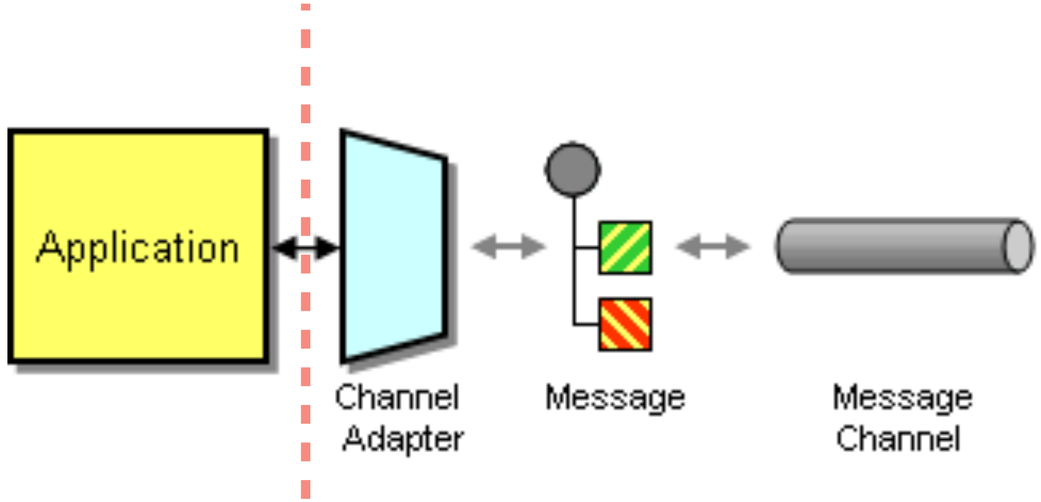
Tipo de Message Endpoint

Channel Adapter

7

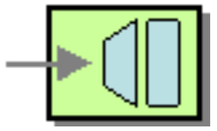
Problema "Como conectar uma aplicação ao sistema de messaging para que ela possa enviar e receber mensagens?"

Solução "Use um **Adaptador de Canal** que possa acessar a API da aplicação ou seus dados, e publique mensagens em um canal, ou que possa receber mensagens enviadas para um canal e chamar funcionalidades dentro da aplicação"



Imagens: eaipatterns.com

Tipo de Message Endpoint

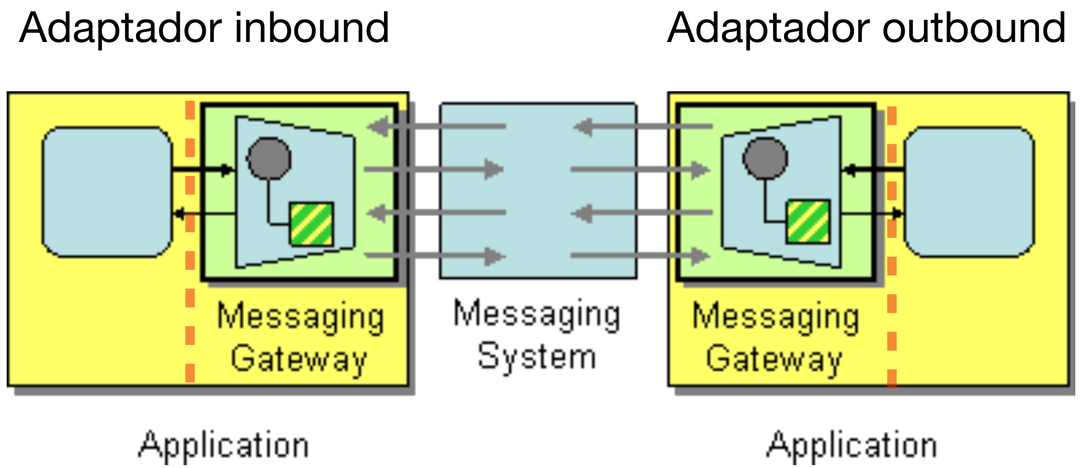


Messaging Gateway

8

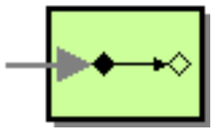
Problema "Como isolar o acesso ao sistema de mensageria do restante da aplicação?"

Solução "Use um **Gateway de Mensageria**, uma classe que encapsula chamadas especificas ao sistema de mensageria e expõe uma interface com métodos específicos ao domínio da aplicação"



Imagens: eaipatterns.com

Tipo de Message Endpoint

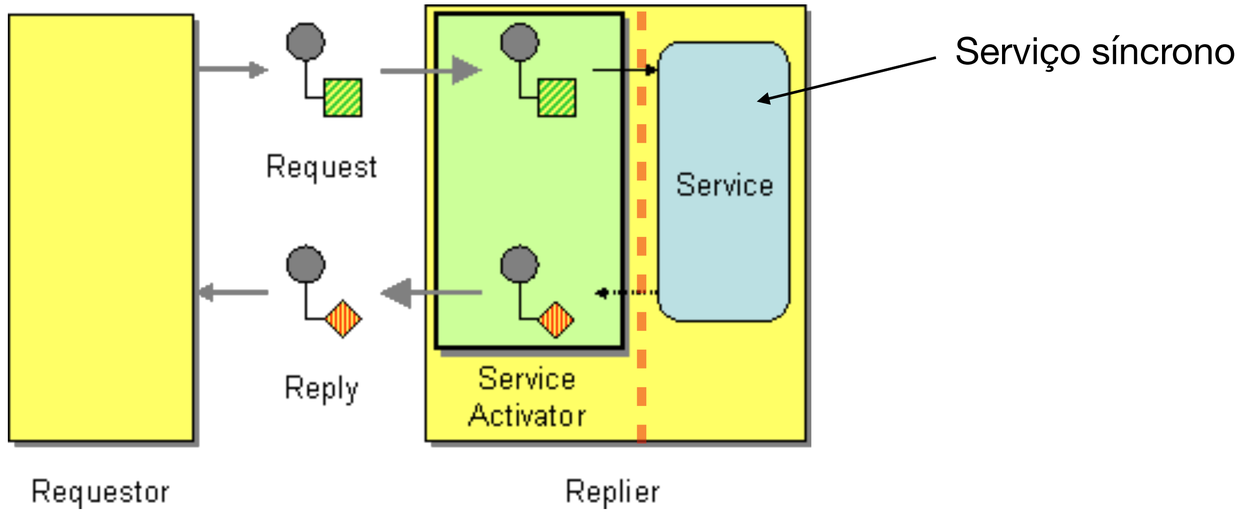


Service Activator

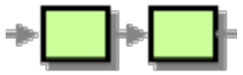
9

Problema "Como projetar um serviço que possa ser chamado de forma síncrona (sem usar mensageria) ou de forma assíncrona (usando tecnologias de mensageria)?"

Solução "Projete um **Ativador de Serviços** que conecte as mensagens do canal ao serviço"

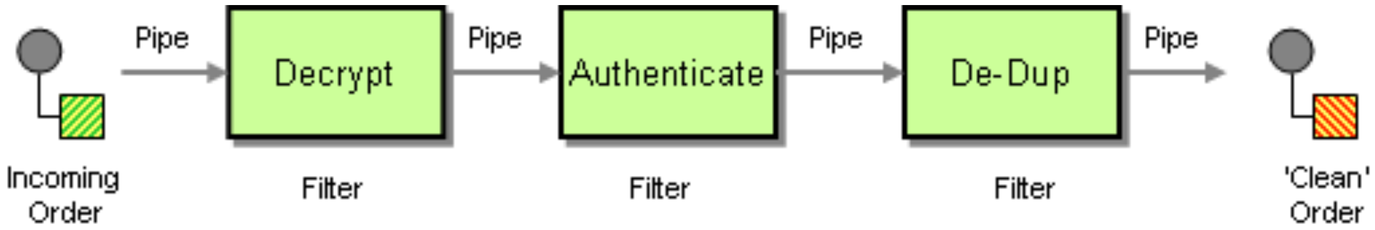


Imagens: eaipatterns.com



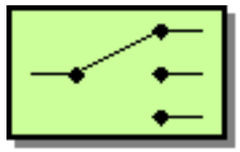
Arquitetura **dutos** e **filtros**

- Padrão de arquitetura (PEAA*)
  - Representa a **conexão** de **componentes** (filtros), em série, através de dutos (**canais**) que os conectam
  - Em vez de enviar a mensagem diretamente ao destinatário, pode-se **interceptá-la** em **etapas** intermediárias para validação do seu conteúdo, roteamento, transformação de dados, etc.
  - Cada **componente** age como filtro; cada **canal** age como duto.
  - Permite criar **rotas** (arquitetura de pipeline)



* Patterns of Enterprise Application Architecture (Martin Fowler)

Fonte: eaipatterns.com



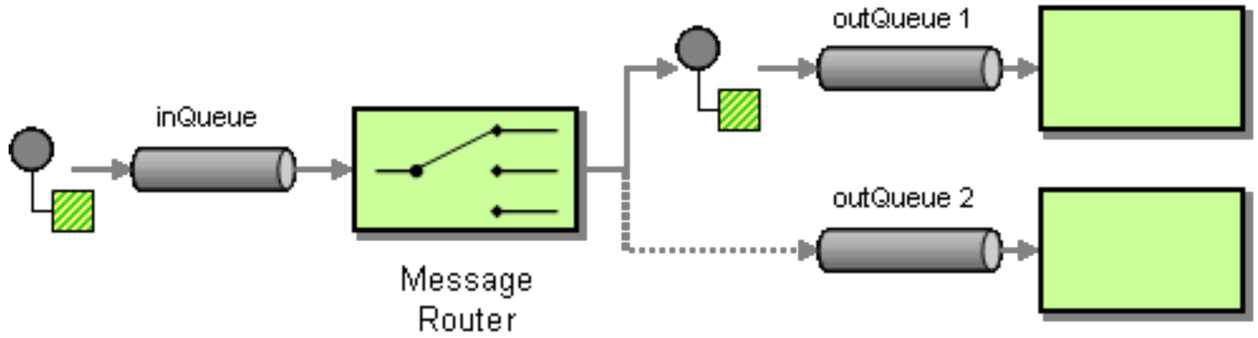
Componente D&F

Message Router

11

Problema "Como desacoplar passos individuais de processamento de forma que mensagens possam ser passadas para diferentes filtros dependendo de uma série de condições?"

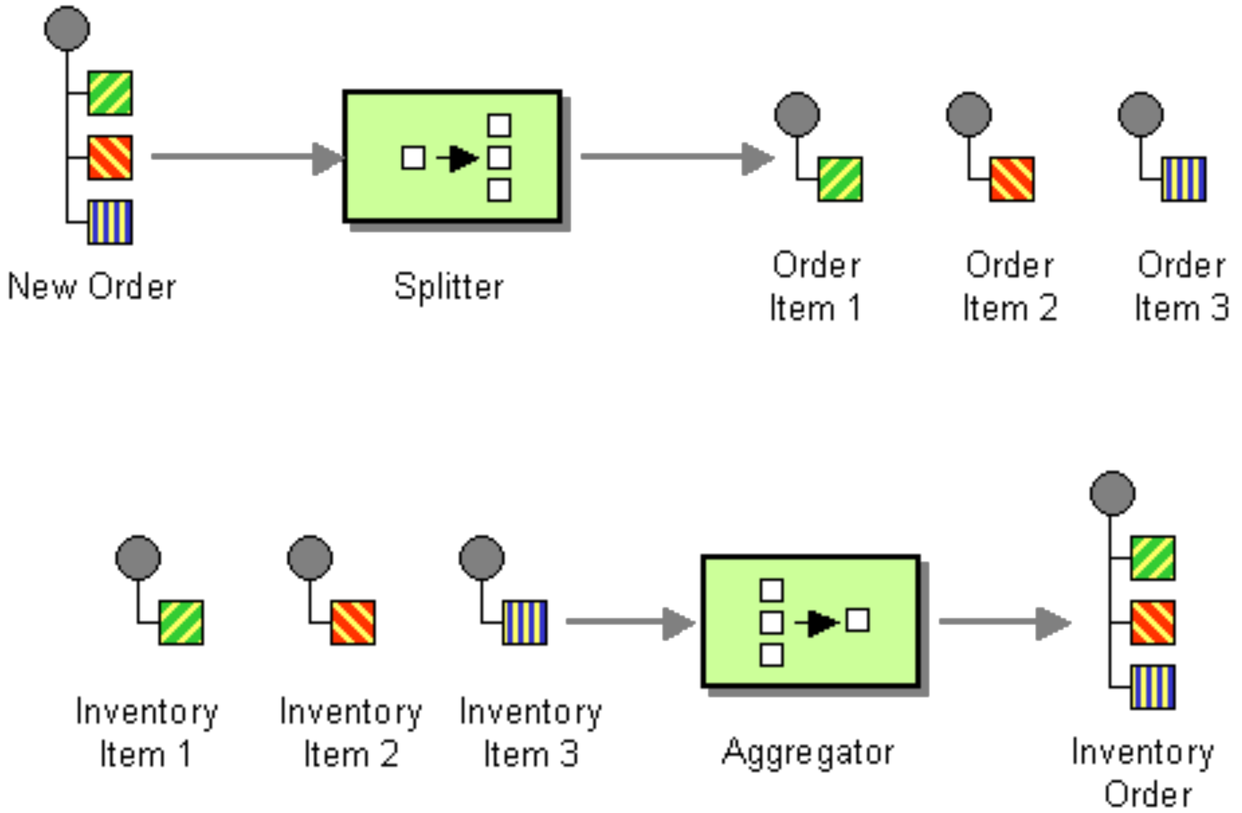
Solução "Use um filtro especial, um **Roteador de Mensagens** (Message Router) que consome uma Mensagem de um Canal de Mensagens **e** publique-a em outro Canal de Mensagens, dependendo de uma série de condições."



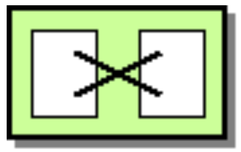
Condições para rotear: algoritmo estático ou dinâmico, conteúdo, eventos externos, etc.

Imagens: eaipatterns.com

Splitter e Aggregator



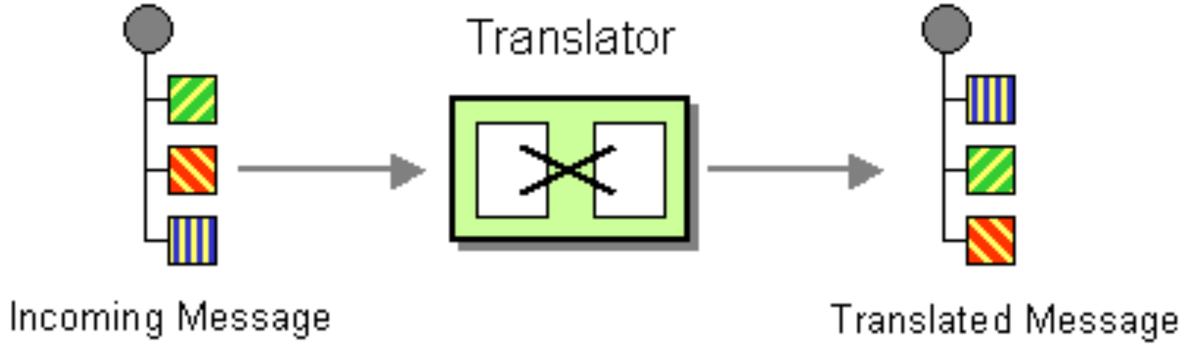
Imagens: eaipatterns.com



Message Translator

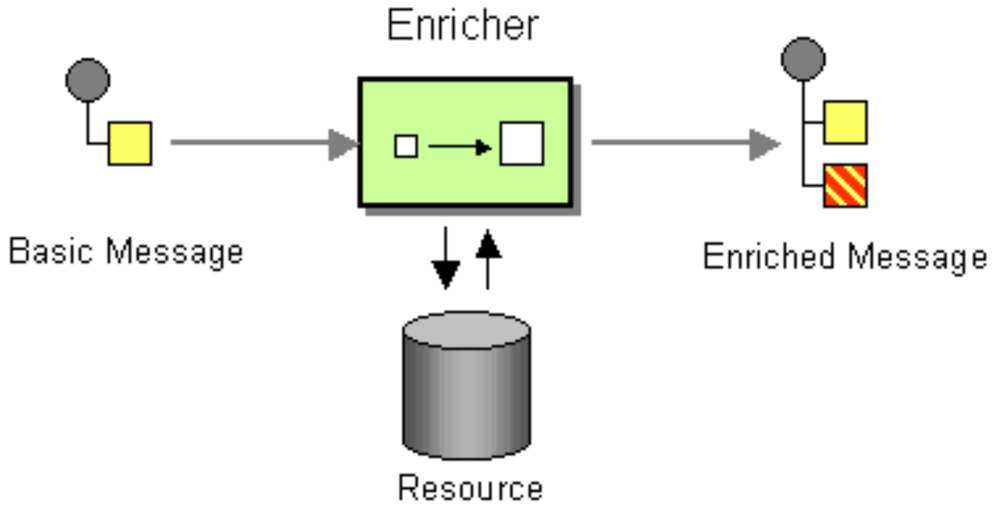
Problema "Como é possível realizar a comunicação entre sistemas que usam formatos de dados diferentes?"

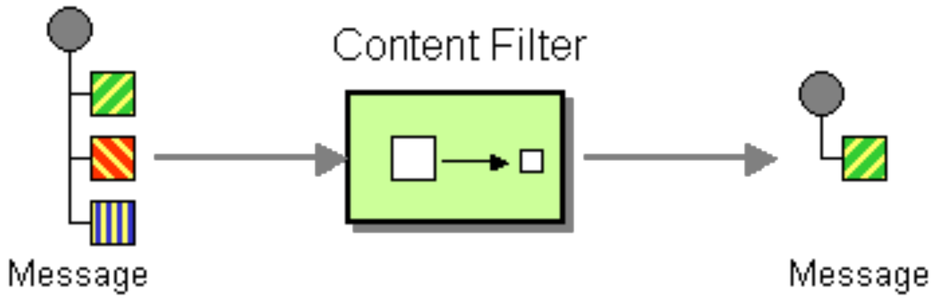
Solução "Use um filtro especial, um **Tradutor de Mensagens**, entre outros filtros ou aplicações para traduzir de um formato de dados para outro."



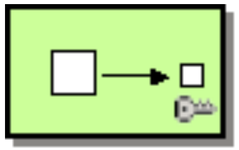
Imagens: eaipatterns.com

Content Enricher + Filter





Imagens: eaipatterns.com



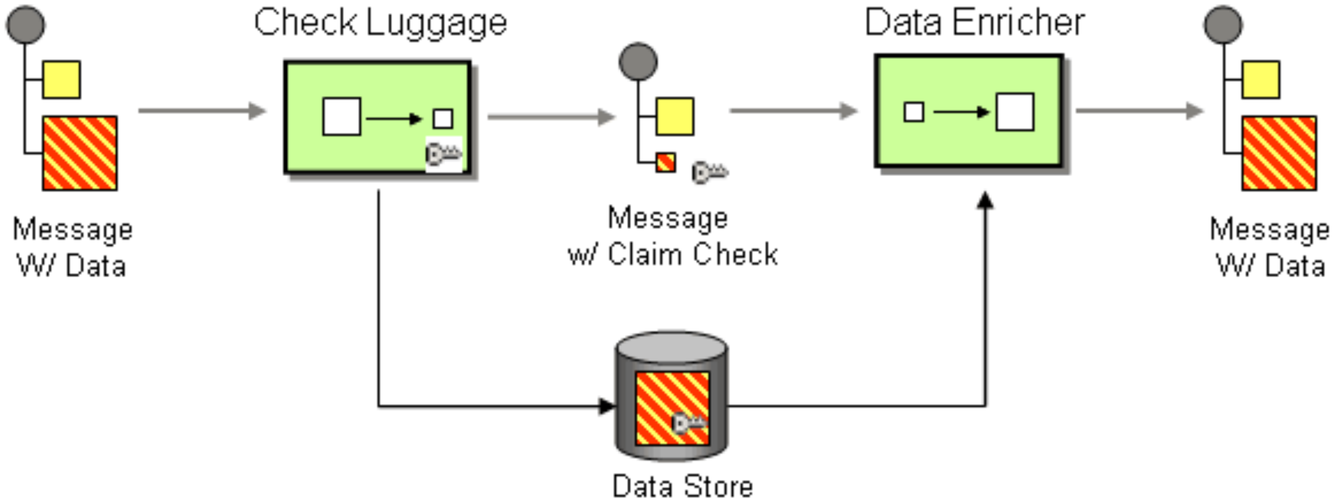
Tipo de Message Translator

Claim Check

15

Problema "Como podemos reduzir o volume de dados de uma mensagem enviada através do sistema sem sacrificar o conteúdo da informação?"

Solução "Guarde os dados da mensagem em um repositório persistente e passe um **Recibo de Bagagem** para os componentes seguintes. Esses componentes poderão usar o Recibo para recuperar a informação armazenada"



Imagens: eaipatterns.com

Como usar os padrões?

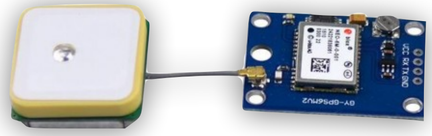
- Os padrões são **abstrações de alto nível** que permitem **descrever a solução** de um problema de integração
- Use os padrões para **descrever a arquitetura de rotas** de uma solução de integração
- Não existe solução única
  - Mesmos resultados podem ser alcançados com arquiteturas diferentes (+ conseqüências diferentes)

Problema #1: integrar IoT e webservice

Calcular a **diferença** entre uma temperatura medida e a temperatura média prevista para uma localidade.

ESP8266
Microcontrolador WiFi

DS18B20
Sensor de temperatura



Neo6mV2
Módulo GPS

Envia dados via WiFi e protocolo **MQTT**:
`00027401235489 1466388`

27,4°C, Latitude -23,5489, Longitude -46,6388

Comparar temperaturas

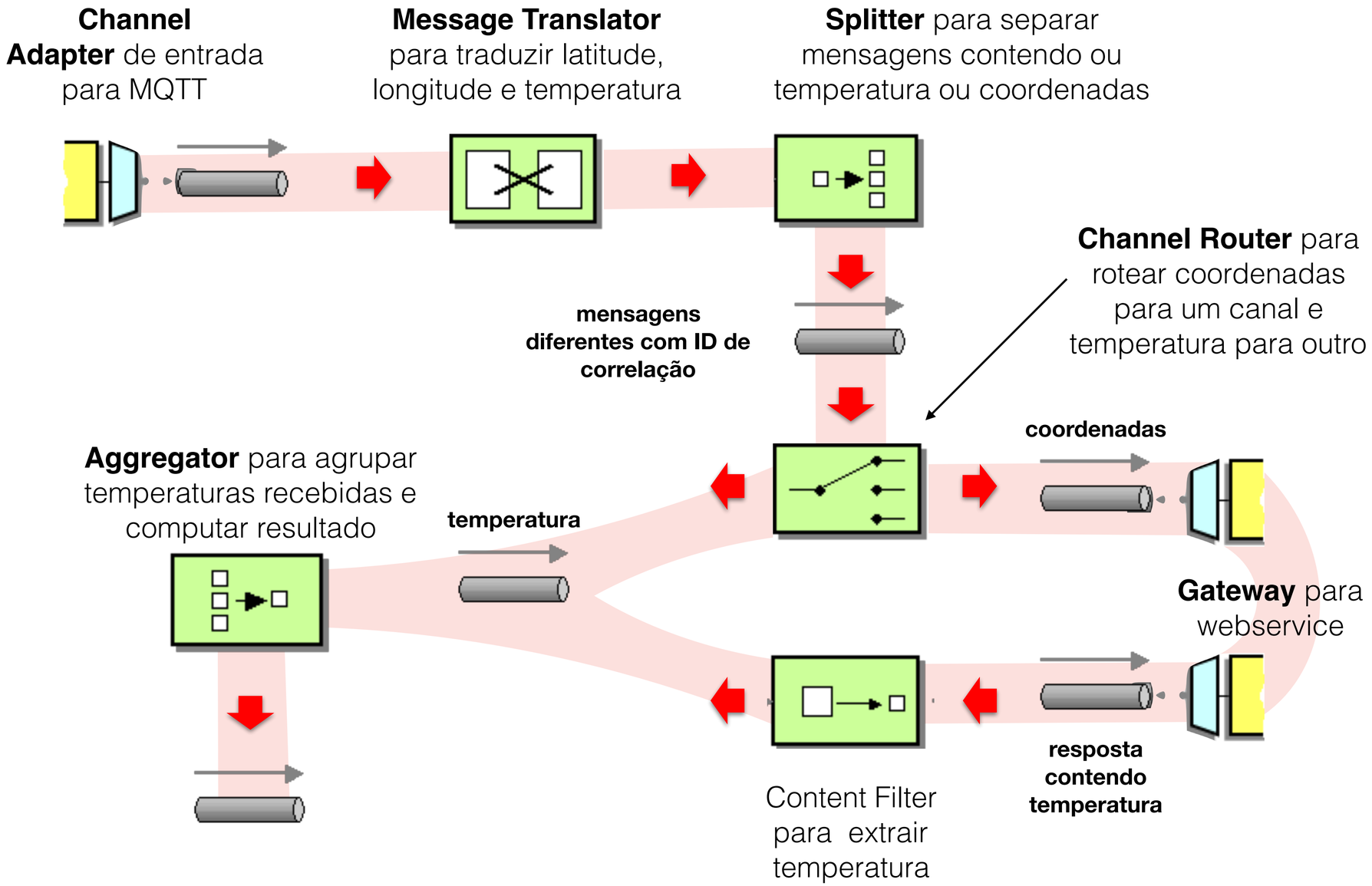
Usar coordenadas

Web Service

http://tempo.xyz/previsao.php?lat=**-23,5489**&lon=**-466388**&format=xml

```
<dwml ...>
   <temperature type="average" units="Celsius">
        <value>24.6</value>
   </temperature>
</dwml>
```

Possível solução para problema #1



Problema #2: integrar Twitter com página Web

Criar página Web contendo **tweets** selecionados em tempo real.

- Obter todas as mensagens com hashtag **#TheDevConf** ou **#TDC2019** no Twitter, periodicamente (polling)
- **Processar** apenas mensagens que tiverem **links**
- **Estruturar** o conteúdo das mensagens identificando **#hashtags**, **@usuários**, **links** e **remetentes**
- **Separar** mensagens que tratam de "**Java**" das demais mensagens
- **Disponibilizar** (arquivo de log e página Web) com **duas** listas de mensagens (Java e outras)

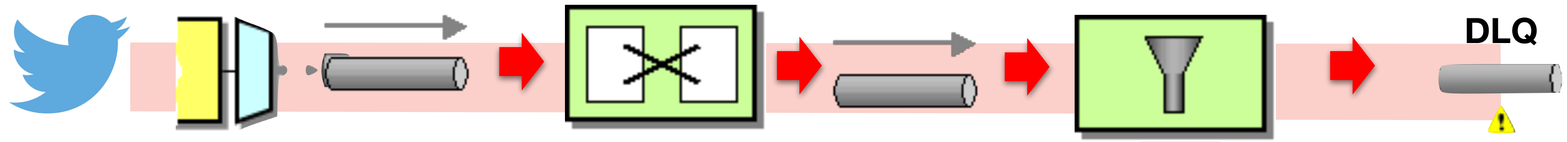
Possível solução para problema #2

Channel Adapter de entrada para Twitter

Translator para extrair dados do Tweet e guardar em uma mensagem simples

Message Filter para jogar fora mensagens que não têm links

DLQ

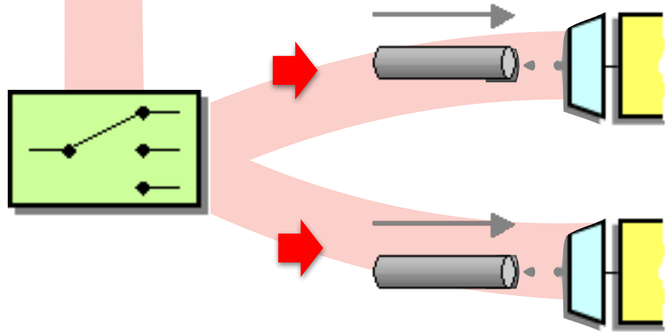


Translator para converter mensagem em fragmento HTML

Enricher para adicionar cabeçalho com assunto da mensagem

Router para rotear mensagens sobre Java para canal diferente

Channel Adapter para streams, arquivos ou serviços Web



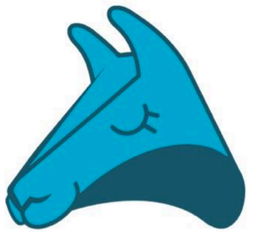
Como implementar?

- Construir solução caseira usando Java e JMS
- Usar um **framework** que implemente padrões de integração de sistemas
- **Alternativas** populares para Java:

Spring Integration

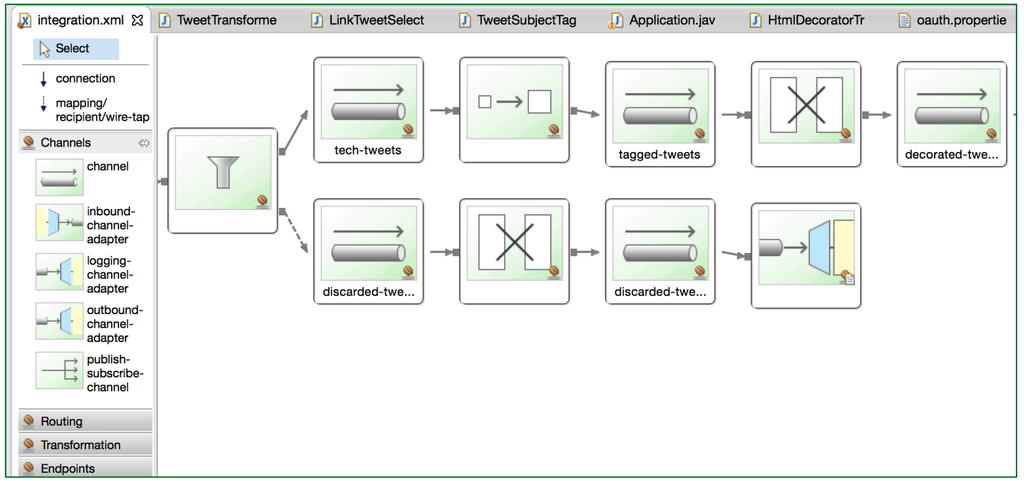
Apache **Camel** (+Spring Boot)

Alpakka (Akka streams)

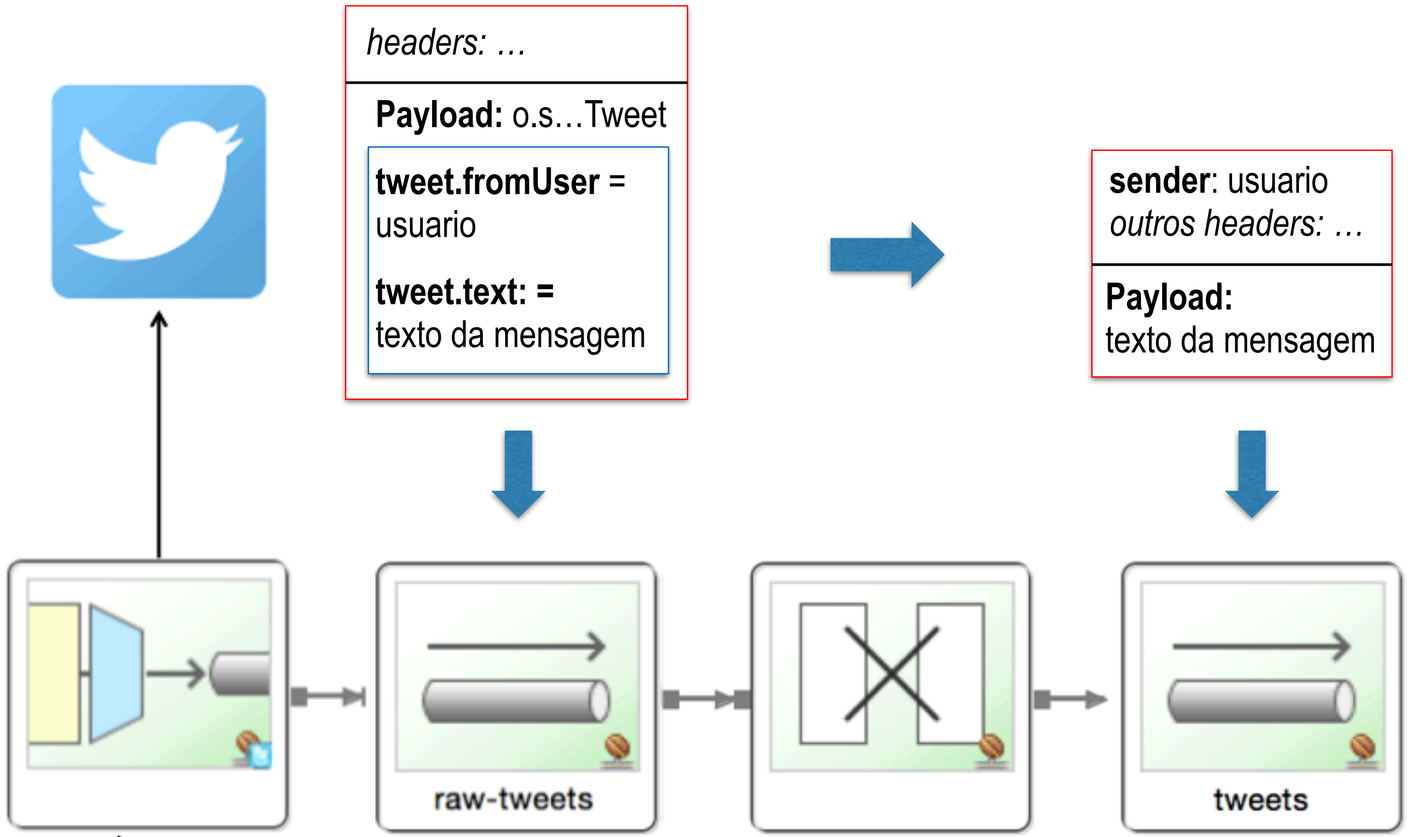


Spring Framework

- Spring é uma **plataforma Java** construída sobre conceitos **injeção de dependências** (DI) e aspectos
- Componentes Spring vivem em um **container** e são interligados pelo framework usando **DI**
- **Spring Integration** - infraestrutura para mensageria que implementa principais padrões
- Ferramentas para a plataforma Eclipse: **STS**



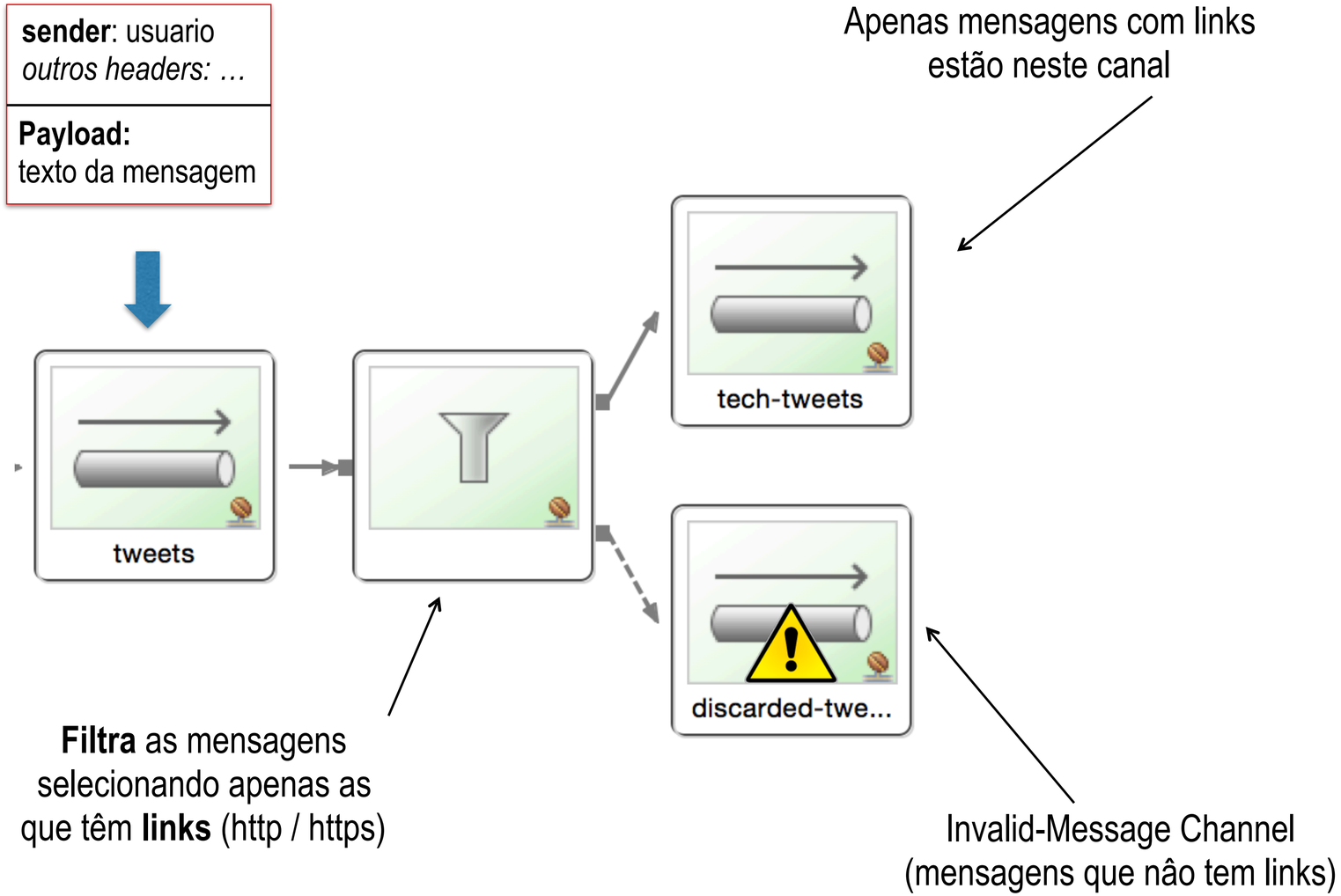
Inbound Channel Adapter



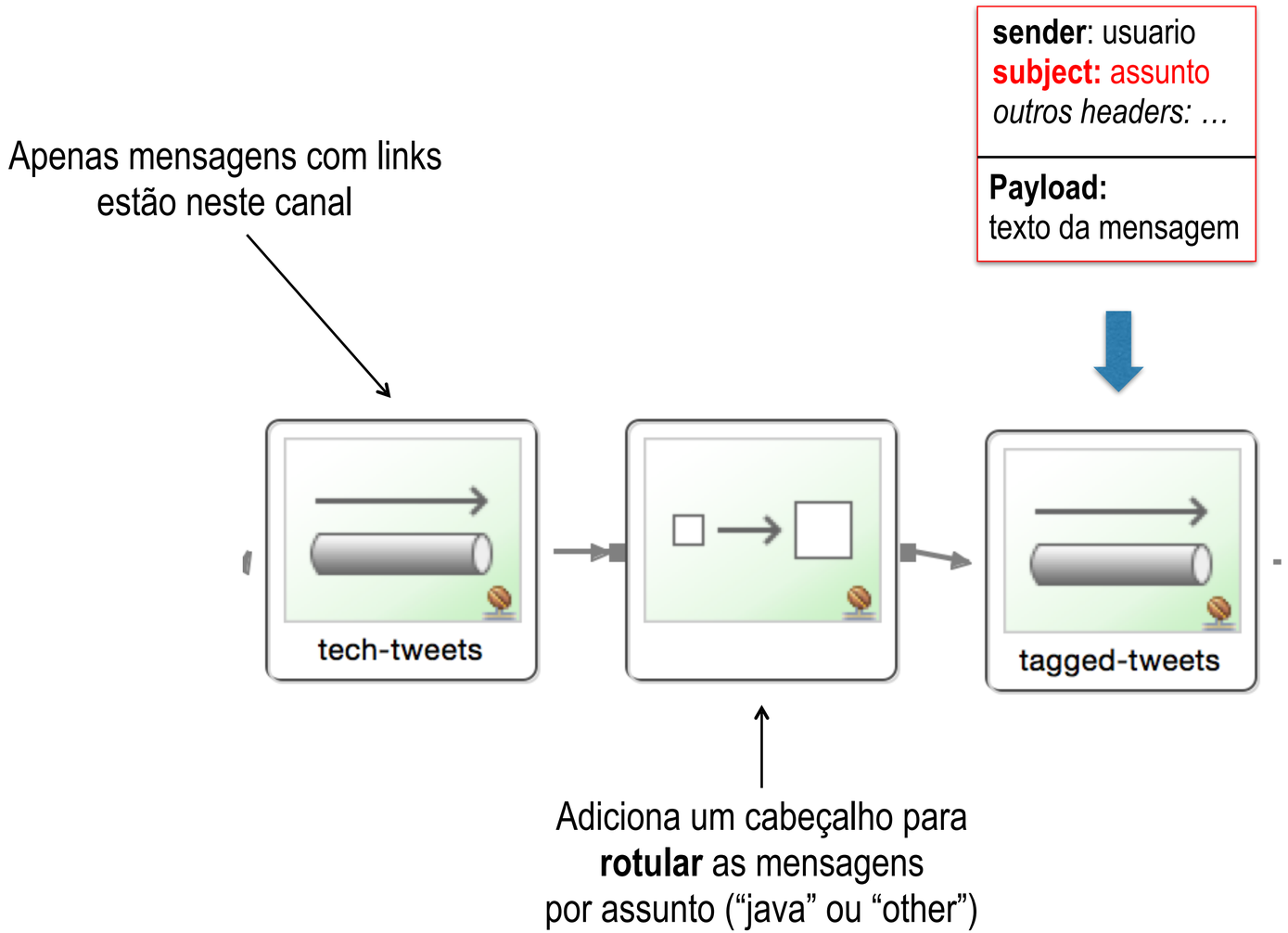
Faz query no **Twitter**:
"`#TheDevConf OR #tdc2019`"
Põe mensagens no canal **raw-tweets**

Translator: Extrai dados de objeto **Tweet** e cria novo objeto **Message**

Message Filter



Content Enricher

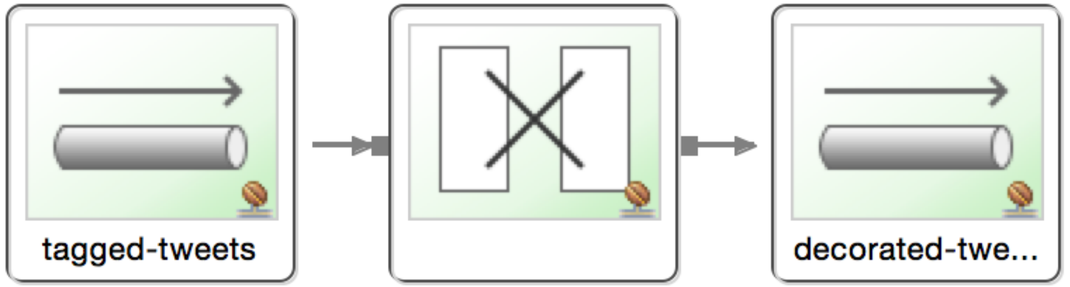


Message Translator

sender: usuario
subject: assunto
outros headers: …

Payload:
texto da mensagem

Transforma o texto em fragmento HTML
(links em <a href>, hashtags em , etc.



sender: usuario
subject: assunto
outros headers: …

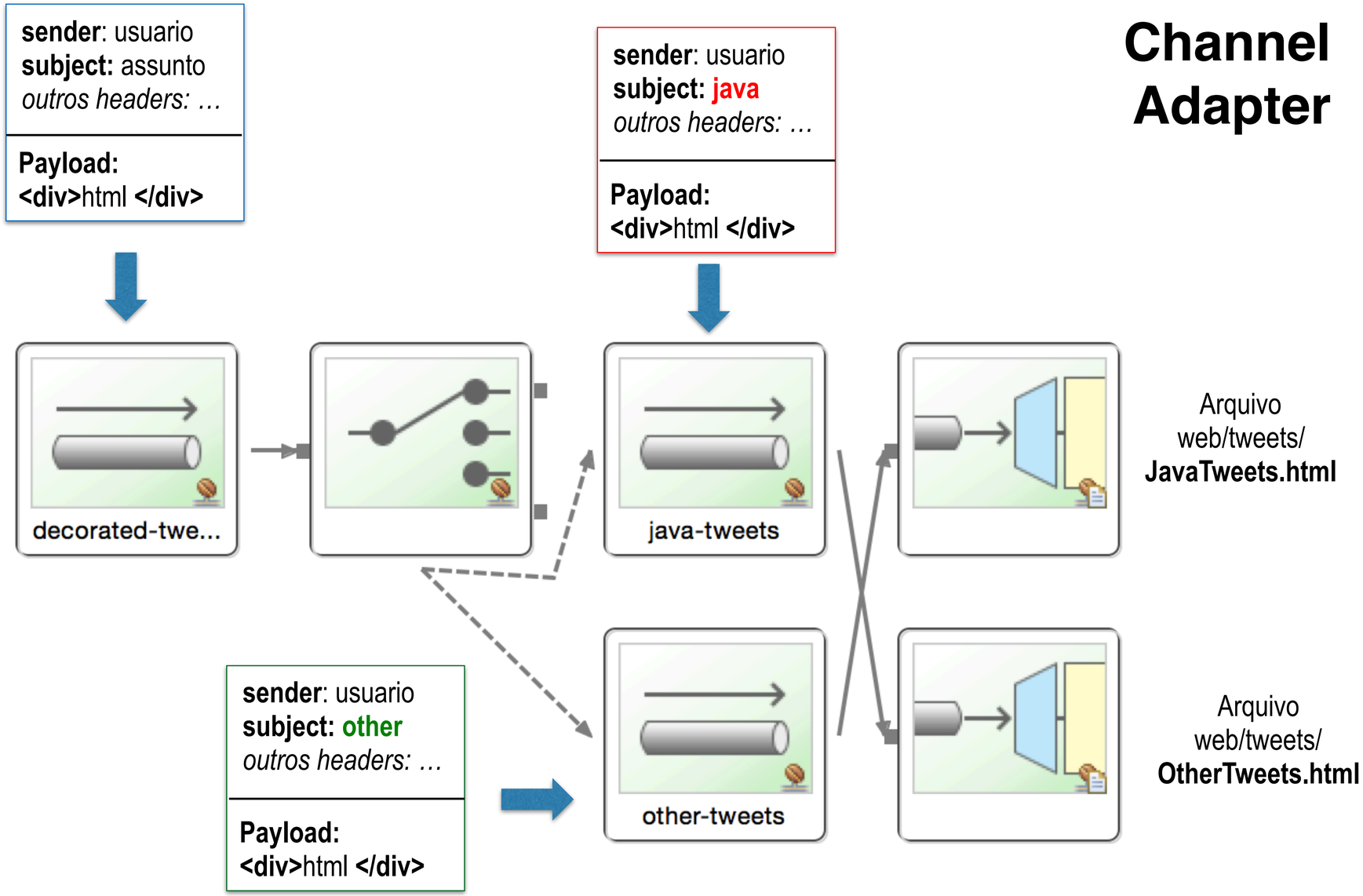
Payload:
** **html da mensagem** **

Palestra de @helderdarocha sobre EIP patterns e SpringIntegration #TheDevConf https://pic.twitter.com/eIcFIYAo8T

** **usuarioPalestra de @helderdarocha sobre EIP patterns e SpringIntegration <span class="hashtag”>#TheDevConf https://pic.twitter.com/eIcFIYAo8T** **

Message Router

+ Outbound **Channel Adapter**



Resultado

JavaTweets.html

OtherTweets.html

Ajax

#TheDevConf Twitter feeds

localhost:8080

#thedevconf and #tdc2019 tweets

Java tweets (contain Java-related strings)

Fabicanedo @TheDevConf voluntária hoje na trilha de Java, coordenada pela @psanrosa13 entre outros, só vamos pessoal que a pro… https://t.co/HsNWk7ucjW

rafaelvcunha @thedevconf São Paulo 2019 #DevLife #Java #JavaDeveloper #Developer #TheDevConf #TDC #TDC2019 #beard #bearded… https://t.co/FJJjSs6hii

rafaelvcunha @thedevconf São Paulo 2019 #DevLife #Java #JavaDeveloper #Developer #TheDevConf #TDC #TDC2019 #beard #bearded… https://t.co/FJJjSs6hii

Other tweets

DiogosGuedes "O Menino da Inovação em sua Nuvem Voadora, manda um salve pra todos!" ✌ (risos). Na Palestra de Migração p/ Clo… https://t.co/GnrHvJTjD6

Dionei_Piazza 🎯 Um dia fantástico ontem no The Developers Conference aqui em SP. Palestrei, compartilhei, aprendi e cresci!… https://t.co/a70cSPUAb5

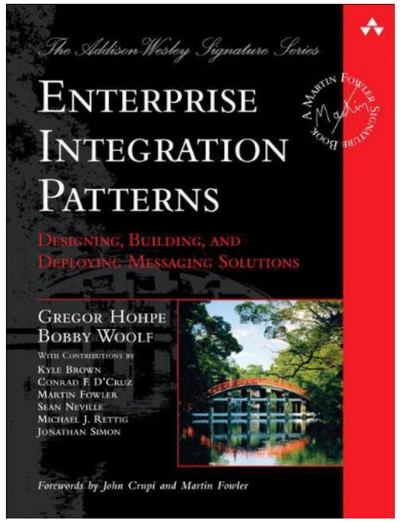
myLIMS Primeiro dia do evento #TheDevConf com a equipe Labsoft! Prontos para otimizar nossos processos ;) #labsoft https://t.co/JsmgV3qMra

educostadev Excited to the start of the #TheDevConf SP/Brazil (The Developer Conference) https://t.co/8L4Dx0J7tK

pasimoes Deck da minha apresentação no #TheDevConf São Paulo 2019 sobre @CloudNativeFdn #Cloud #cloudnative #cncf… https://t.co/9NItWfDDfQ

Conclusões

- **Mensageria** é uma estratégia eficiente para integrar aplicações que não foram projetadas para trabalhar juntas
- É **difícil** usar mensageria
- Padrões de mensageria promovem a **separação de responsabilidades** complexas (processamento, roteamento, etc.) facilitando a descrição da arquitetura de uma **solução** de integração
- Existem **produtos** que implementam os principais padrões, facilitando a implementação de soluções



Referência

Gregor Hohpe, Bobby Woolf, et al

Enterprise Integration Patterns

http://www.eaipatterns.com/

E-book + código

https://github.com/helderdarocha/EIP-Course

THE DEVELOPER'S CONFERENCE

padrões essenciais de mensageria para integração de sistemas

Baixe esta palestra (depois do TDC) em

http://www.argonavis.com.br/download/tdc_2019_eip.html

+Links para outros recursos, código, referências

https://github.com/helderdarocha/tutorial-messaging-patterns

www.summa.com.br

helder da rocha

helder@summa.com.br