

Integração Ágil com Camel

Integração tri-legal

Sobre

- **Otavio R. Piske**
 - Twitter: [@otavio021](https://twitter.com/otavio021)
 - E-mail: angusyoung@gmail.com
- **Meu trabalho**
 - Senior Software Engineer @ Red Hat
 - Committer @ Apache



Agenda

- Introdução
- Camel
- Camel K
- Kamelets
- Camel JBang
- Karavan
- Demo

Introdução

Desenvolvimento é divertido...

Que
maravilha!

Kubernetes
é da hora!

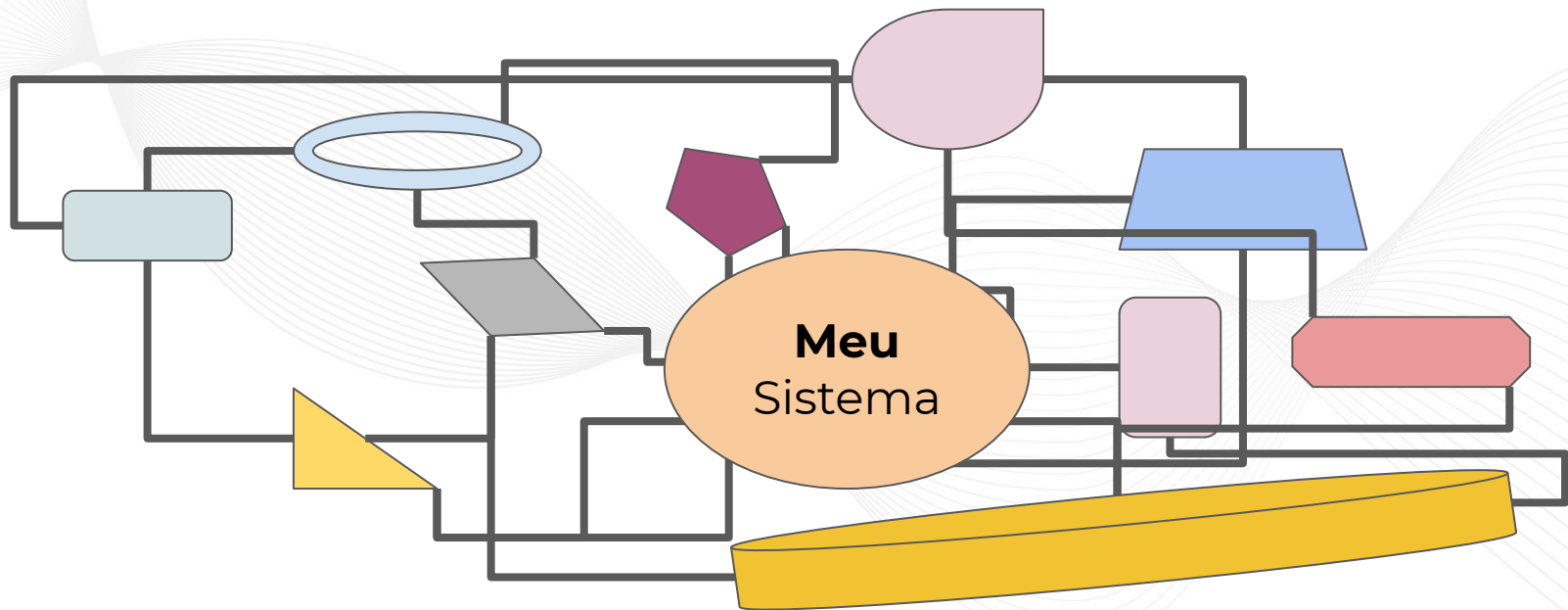


Amamos
**micro
serviços!**

Vamos usar
serverless!

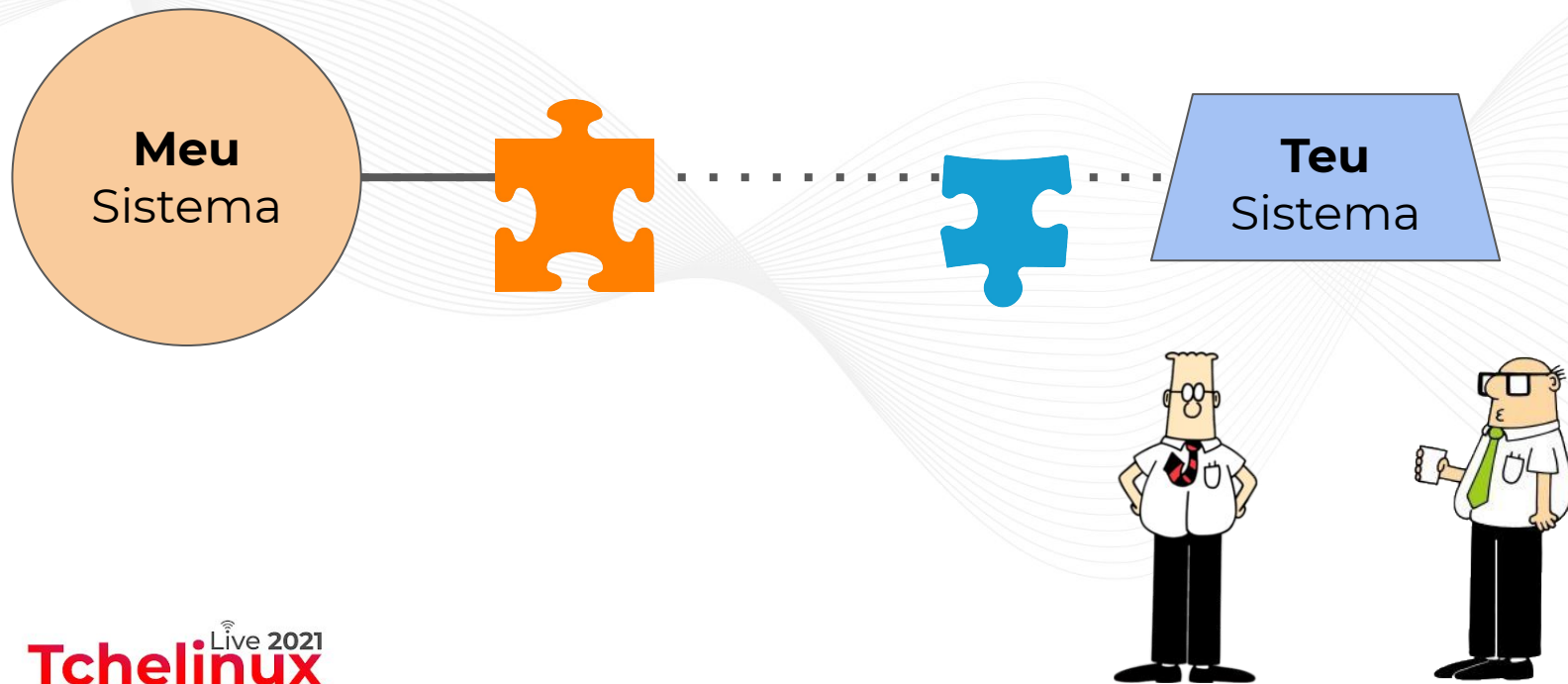
... quando você tem liberdade!

Mas então você percebe que ...

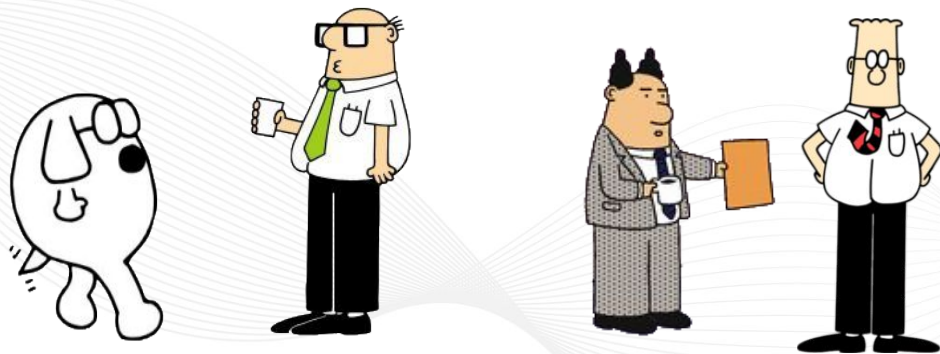


... que não está sozinho no mundo!

Mesmo com apenas 2 sistemas...



Você precisa resolver uma série de problemas



Falar o mesmo idioma

- Modelos de **comunicação** diferentes
- **Protocolos** ou **sistemas de mensageria** diferentes
- **Linguagens** ou **tecnologias** diferentes
- Sistemas **legado** incapazes de mudar



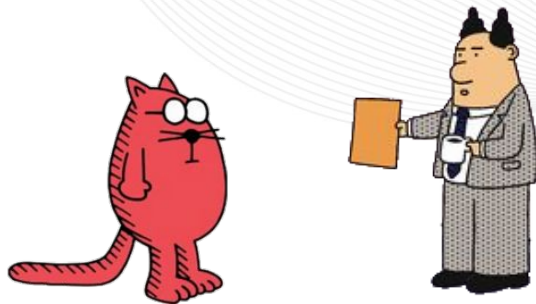
Um "contrato social"

- Domínios de **negócio** diferentes
- Padrões diferentes de acesso a **dados**
- Atender a **requisitos legais**



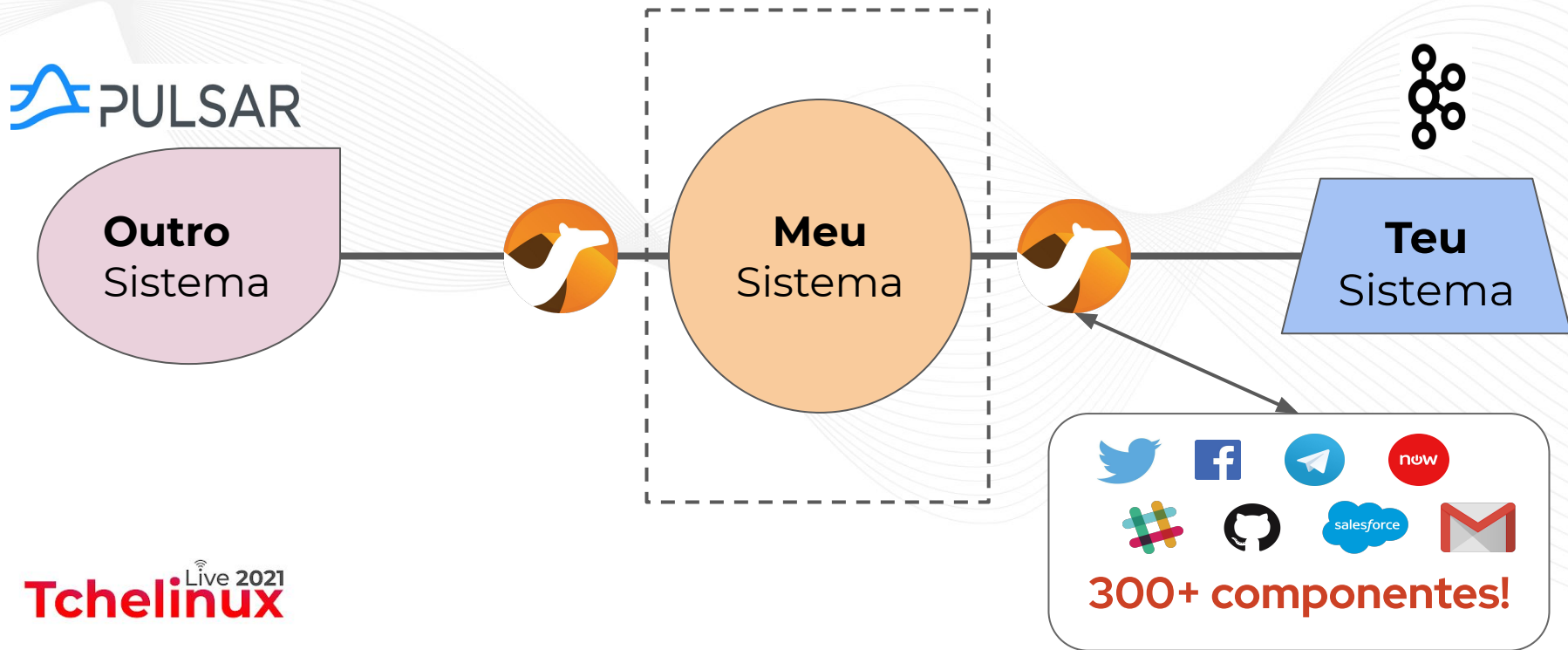
Gerenciar os riscos de integração

- E se a aplicação/rede ficar **indisponível**?
- Como podemos garantir **consistência**?
- Como podemos garantir **disponibilidade**?



Apache Camel

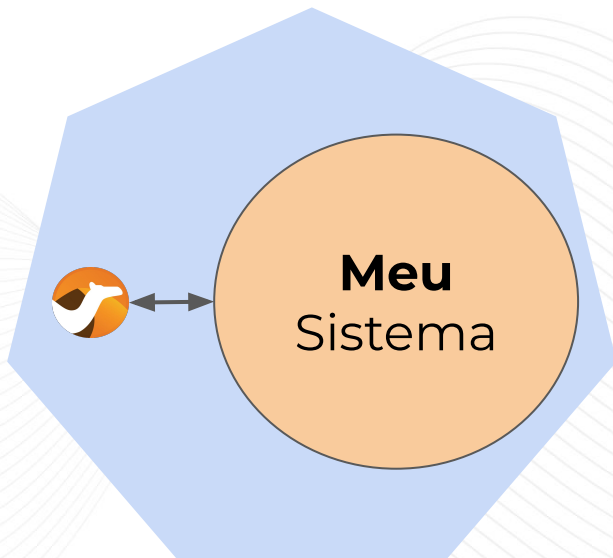
Apache Camel pode ter libertar



Encapsule a complexidade de uma integração

Apache Camel DSL (***route.java***)

```
from("pulsar://company/nsx/topic1")  
  .unmarshal().json()  
  .transform().simple("${body[data]}")  
  .to("rest:post:api1")
```



Apache Camel K

O que é o Apache Camel K?



CAMEL K

Uma plataforma de integração leve,
nascido em Kubernetes, com
super-poderes serverless

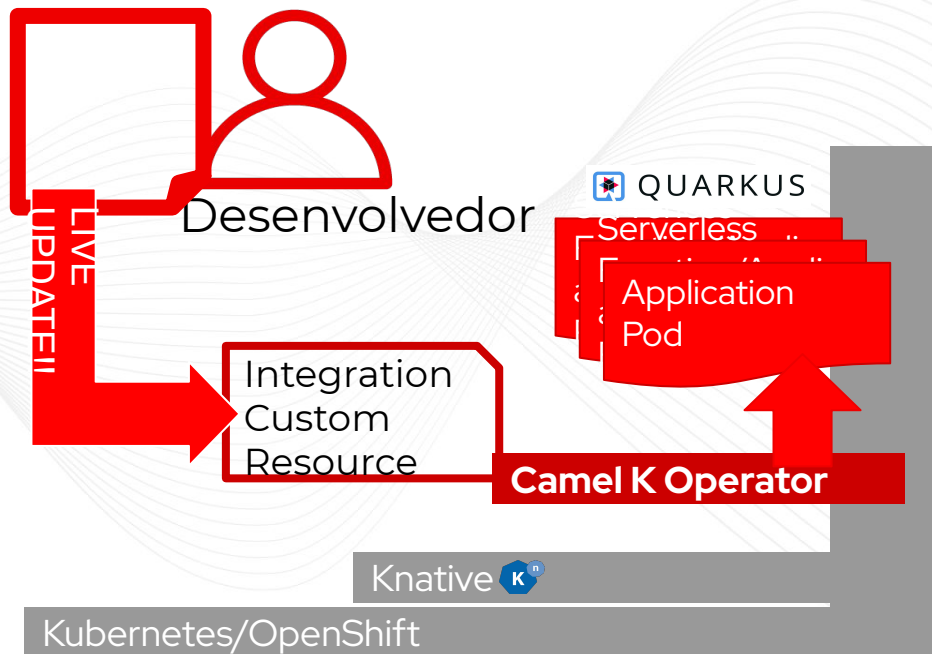
Ferramentas e Ecossistema

- Trabalhe usando o VS Code ou Eclipse Che
 - **Completação** de código e **destaque** de erros
 - Gerenciamento do **ciclo de vida** das integrações
 - Verificação de **status** e **logs**
- Ferramentas **abertas**
- Desenvolvimento **ativo**

Arquitetura do Camel K

Feito sob-medida para uma experiência de **desenvolvimento cloud-native**.

- "Live coding" na nuvem
- Gerenciamento de dependências built-in
- "Deployment" rápido, com updates incrementais
- Altamente customizável



Camel K para desenvolvedores

```
from("knative:channel/xxxx")  
  .transform()...  
  .to("kafka:topic")
```

1

Crie um
arquivo de
integração

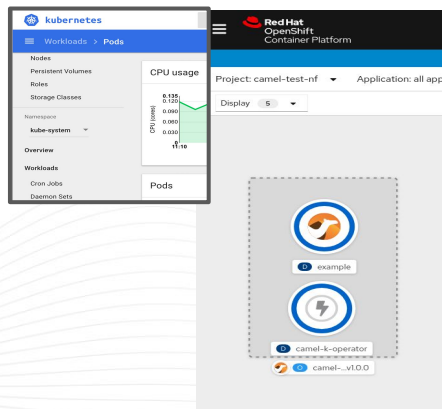
2

EXECUTE a
ferramenta
de linha de
comando

```
$ kame1 run integration.java
```

3

RODE
Serverless no
OpenShift ou
Kubernetes



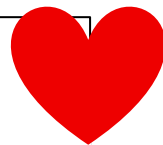
Camel K e Knative

Camel K adora Knative



CAMEL K

Uma plataforma de integração leve,
nascido em Kubernetes, com
super-poderes serverless



Não é bem o
logo oficial ...
mas
gostamos
desse aqui!

Knative



Knative Serving

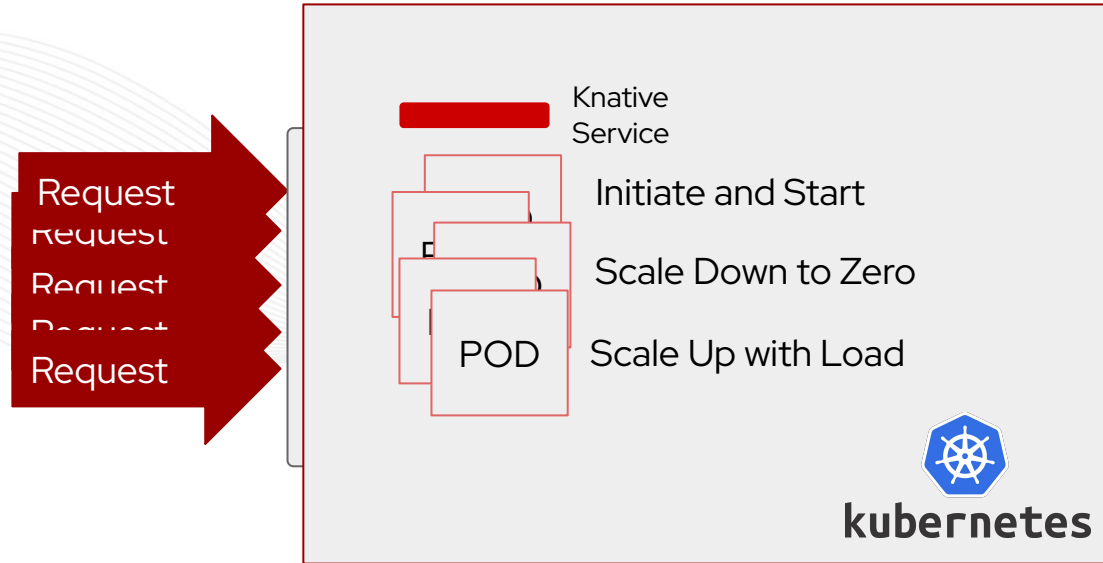
*Auto-scaling e
scale-to-zero*



Knative Eventing

Mensageria para
aplicações baseadas
em eventos

Knative



Kamelets

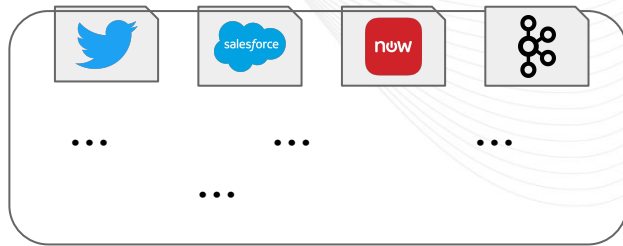
Kamelets

Significa: **Kamel** route snippets

Facilitar a escrita de integrações sem a necessidade de usar a DSL.



Catalogo (Objetos do Kubernetes)



Kamelets em detalhes

Outros casos de uso além de Kubernetes com Camel K:

- Camel Core
- Camel JBang
- Visual Tools for Camel K development

**Mais de 120
Kamelets
disponíveis
no catálogo**

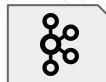
1 - Escolha a origem



2 - Configure os parâmetros

```
query=quarkus  
token=...
```

3 - Escolha o destino



Camel JBang

Camel + JBang -> Camel JBang



- Execução de códigos Java **auto-contidos**
- **Sem** necessidade de **compilar**



- **Expõe funcionalidades** do Camel Core
- Integrações simples **sem** precisar **escrever código**
- **Procure** por componentes e DSLs na documentação

Karavan Designer

Karavan

- Um construtor **visual** de integrações
- Voltado para **usuário comum**
- Integração com **Kamelets**
- Suporte preliminar a DSL de **padrões de integração**



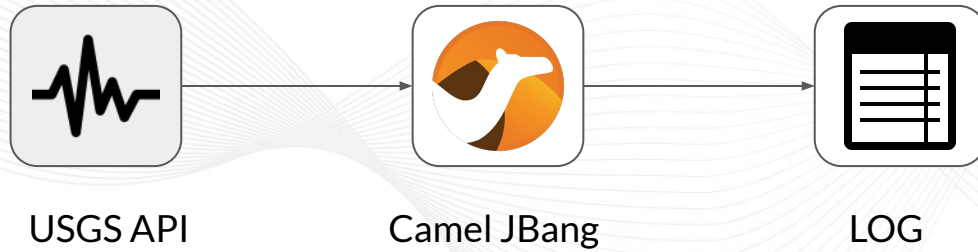
Demo

Demo 1: Camel JBang

- Ajuda
- Listar componentes
- Buscar componentes
- Rodar rotas simples



Demo 2: Earthquake source

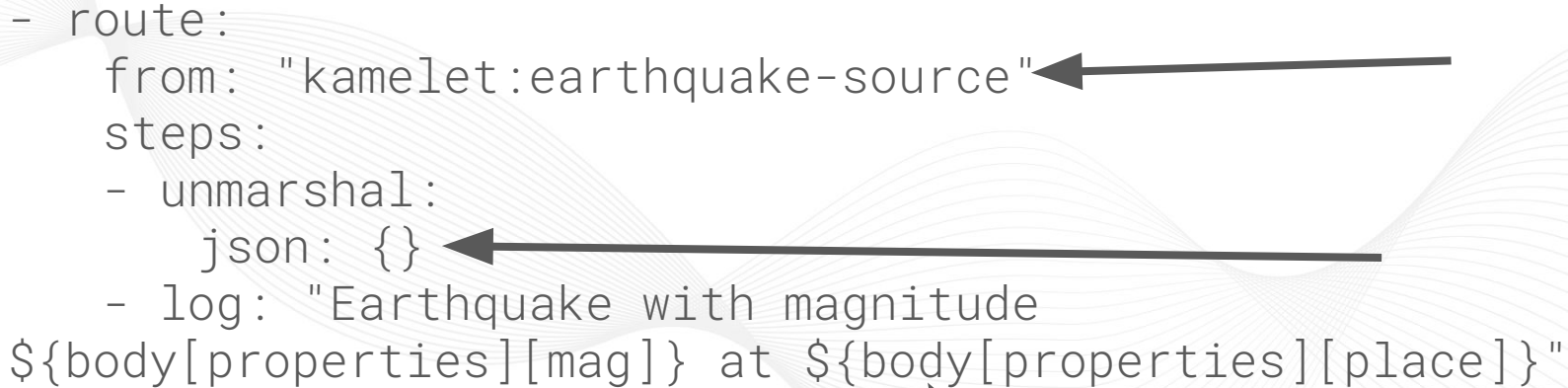


Demo 2: Earthquake source

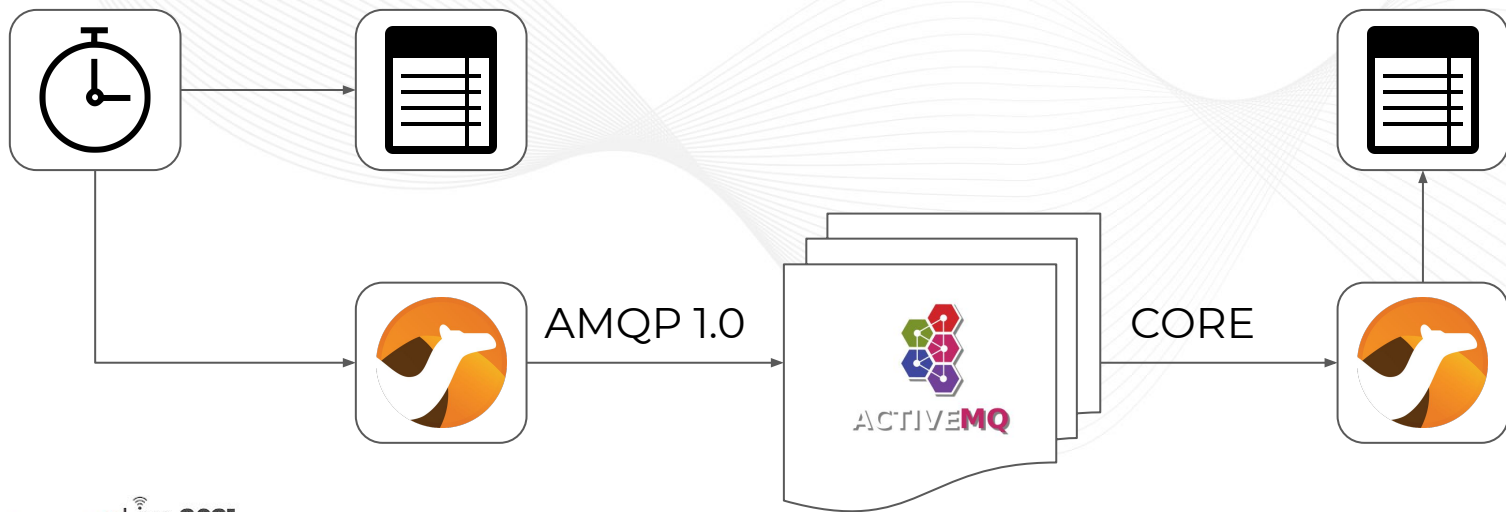
```
{ "type": "FeatureCollection", "metadata": { "generated": 1636120848000, "url": "https://earthquake.usgs.gov/fdsnws/event/1/query?format=geojson&updatedafter=2021-11-01T13%3A13%3A23.834&orderby=time-asc", "title": "USGS Earthquakes", "status": 200, "api": "1.12.3", "count": 1813 }, "features": [ { "type": "Feature", "properties": { "mag": 1, "place": "33 km SW of Alatna, Alaska", "time": 1633528948582, "updated": 1636063859663, "tz": null, "url": "https://earthquake.usgs.gov/earthquakes/eventpage/ak021ctnadw9", "detail": "https://earthquake.usgs.gov/fdsnws/event/1/query?eventid=ak021ctnadw9&format=geojson", "felt": null, "cdi": null, "mmi": null, "alert": null, "status": "reviewed", "tsunami": 0, "sig": 15, "net": "ak", "code": "021ctnadw9", "ids": "ak021ctnadw9", "sources": "ak", "types": ""
```

Demo 2: Earthquake source

```
- route:  
  from: "kamelet:earthquake-source"  
  steps:  
    - unmarshal:  
      json: {}  
    - log: "Earthquake with magnitude  
    ${body[properties][mag]} at ${body[properties][place]}"
```

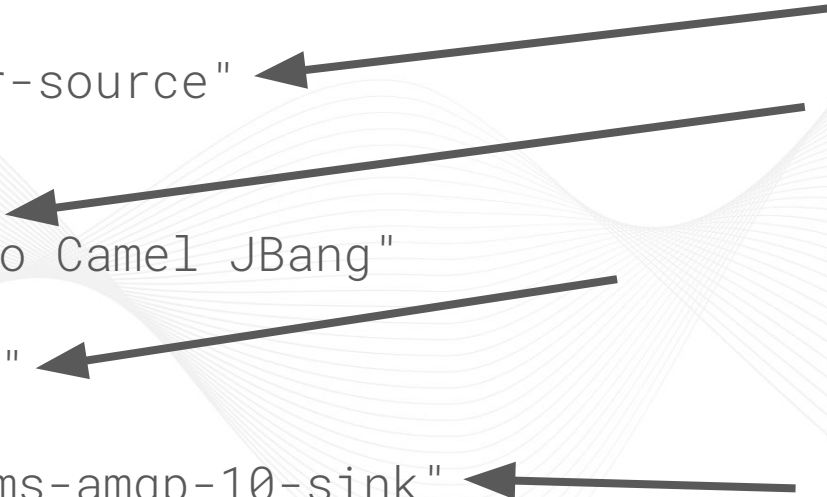
The diagram consists of three arrows. One arrow points from the right towards the 'from' field in the 'route' configuration. A second arrow points from the right towards the 'json' field in the 'unmarshal' step. A third arrow points from the bottom-left towards the 'log' field, specifically pointing to the first placeholder in the log message.

Demo 3: Local broker



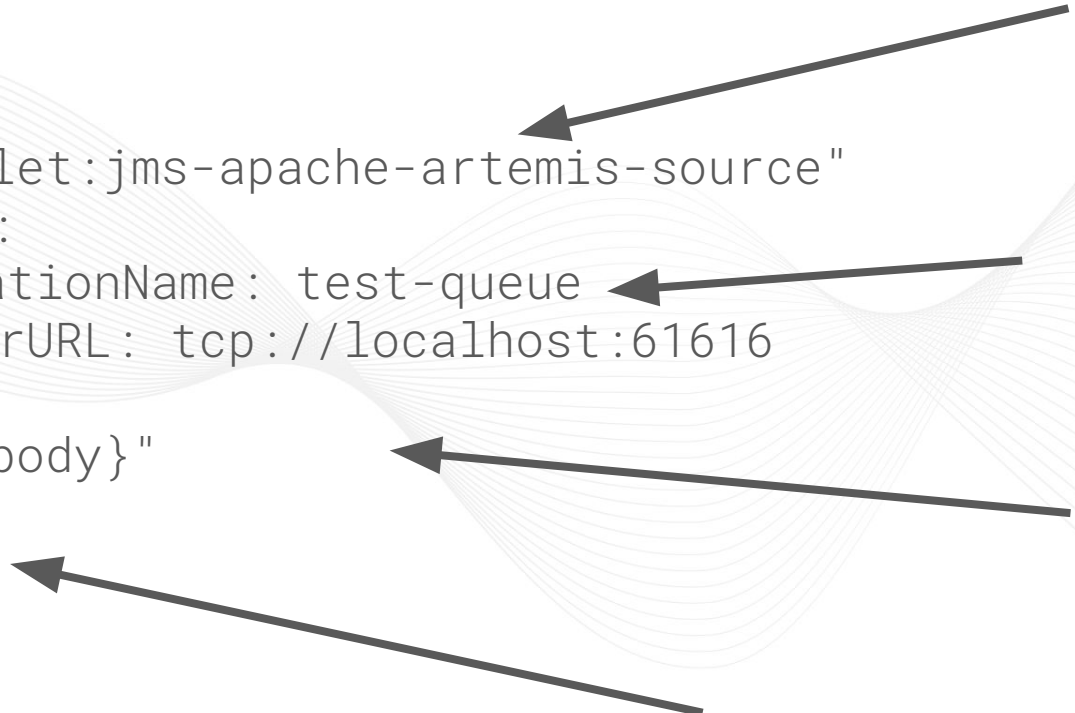
Demo 3: AMQP Sink

```
- route:
  from:
    uri: "kamelet:timer-source"
    parameters:
      period: 1000
      message: "Hello Camel JBang"
  steps:
    - log: "${body}"
    - to:
      uri: "kamelet:jms-amqp-10-sink"
      parameters:
        remoteURI: amqp://localhost:61616
        destinationName: test-queue
```

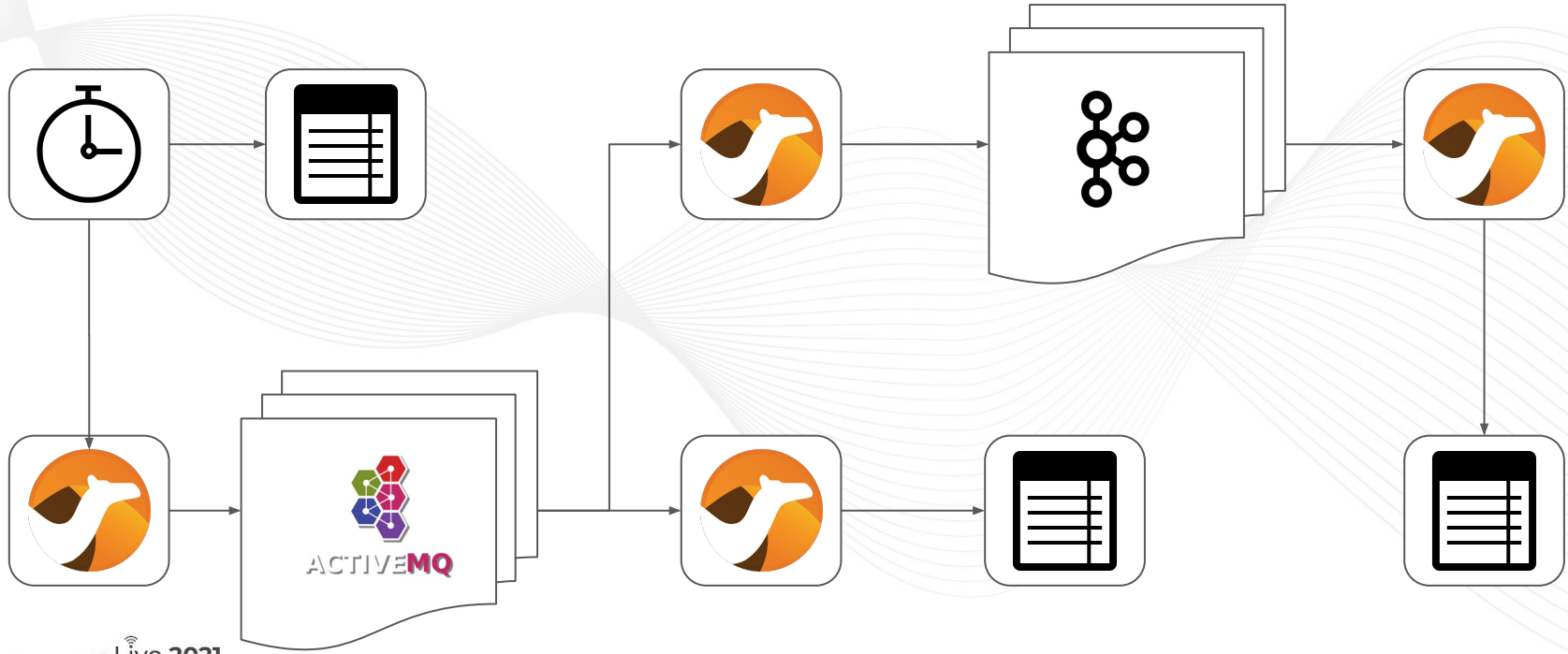


Demo 3: AMQP Source

```
- route:  
  from:  
    uri: "kamelet:jms-apache-artemis-source"  
  parameters:  
    destinationName: test-queue  
    brokerURL: tcp://localhost:61616  
  steps:  
    - log: "${body}"
```



Demo 4: Local Broker with remote Kafka



Comentários Finais



Obrigado

1. Use
 - a. <http://camel.apache.org>
2. Participe
 - a. <https://camel.apache.org/community/>
3. Contribua
 - a. <https://github.com/apache/camel>
 - b. <https://github.com/apache/camel-k>
 - c. ...
4. Promova!

