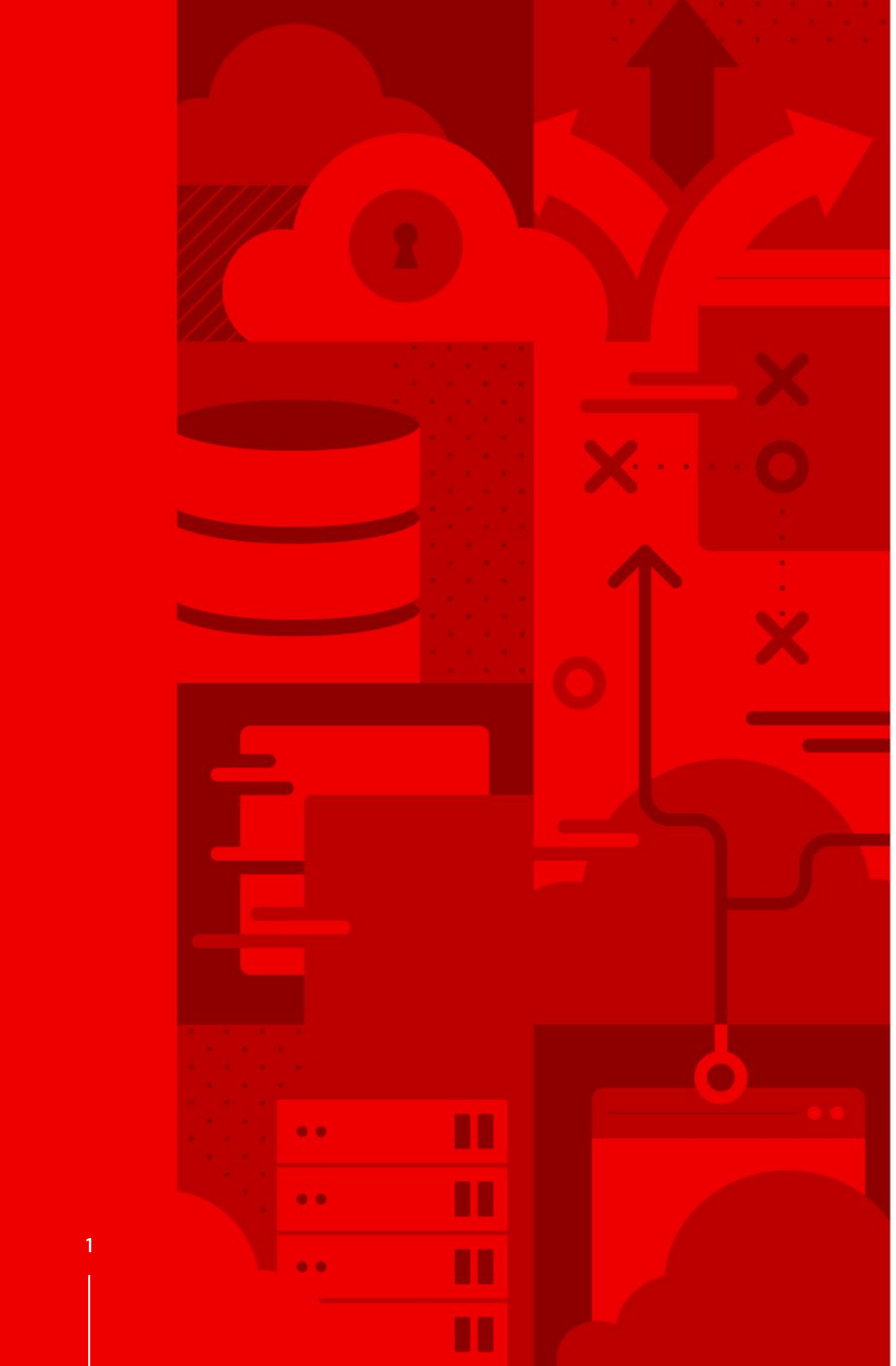


Systems Integration using Apache Camel

Mgr. Ivo Bek

Senior Product Manager

April 2023

Starting a new business

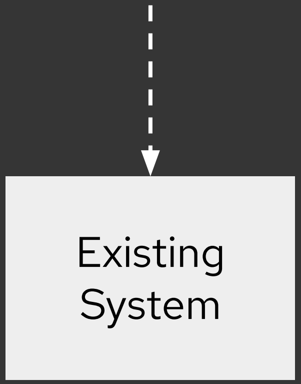
Which products / services would you choose today?

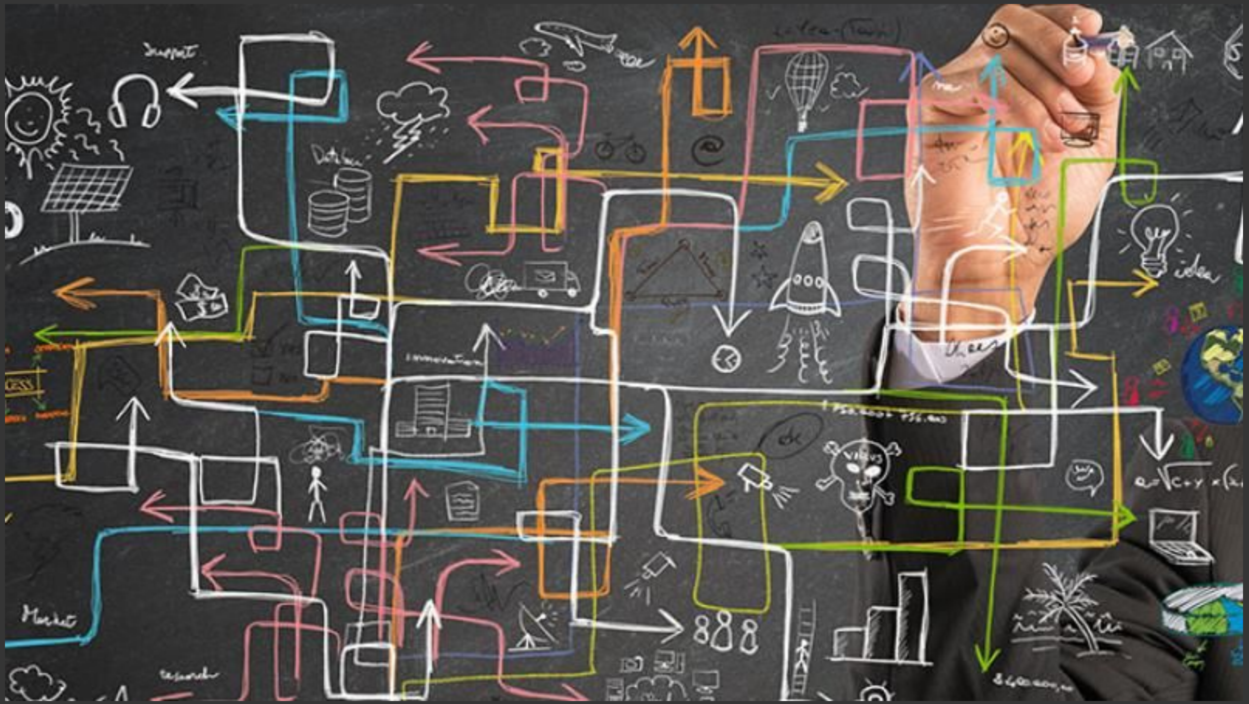
What next?

Respond to new needs and expectations

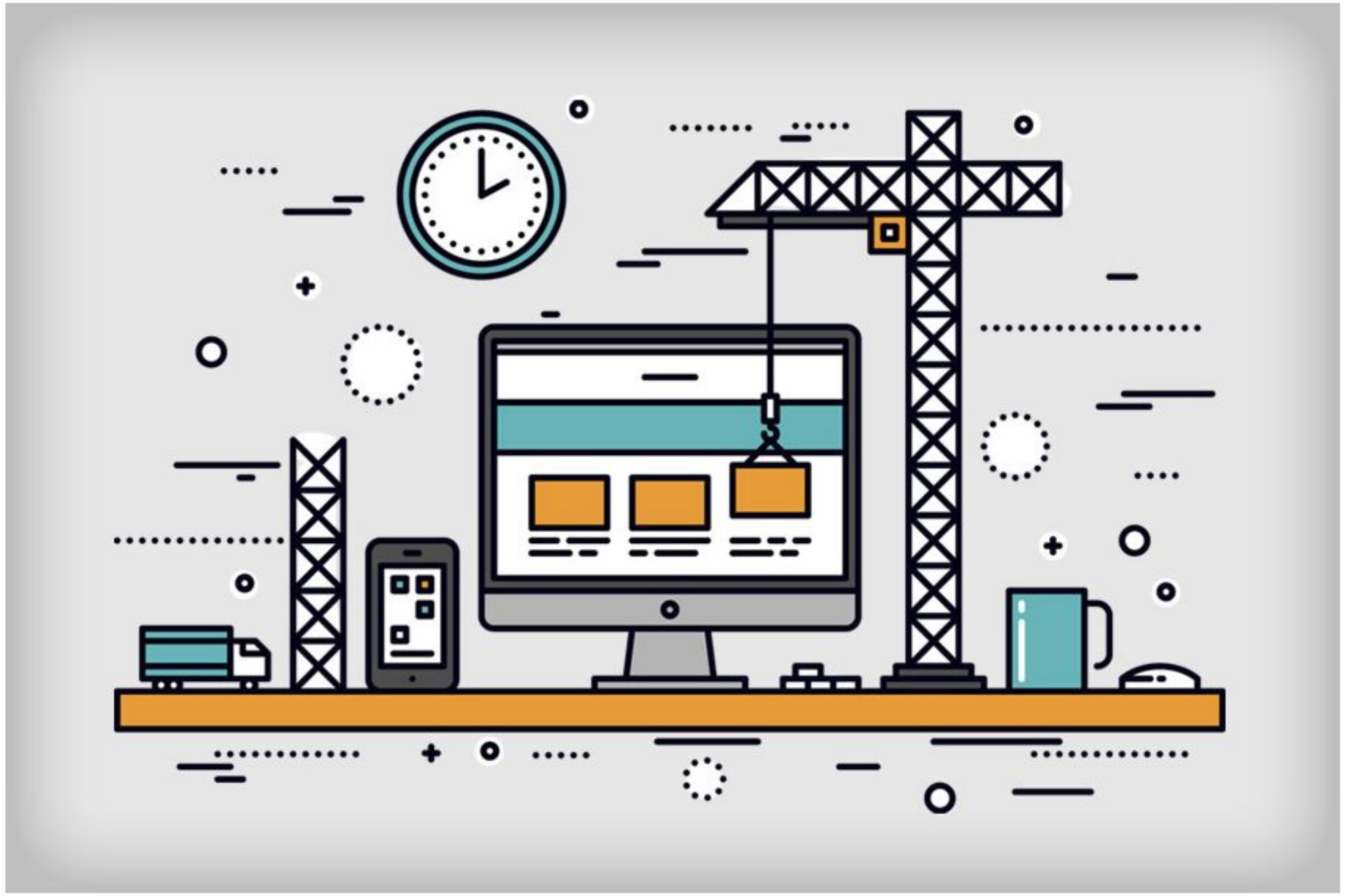
- AI, assistants, bots, ...
- Nanobots, robotics
- Quantum computing
- AR / VR
- Brain computer interfaces
- Space network - laser comms, interplanetary internet,
- Synthetic biology
- Bioengineering
- ....

HOW?


New capabilities
New architectures and approaches
New services
New experiences
Existing System



When would you rewrite parts of the system?



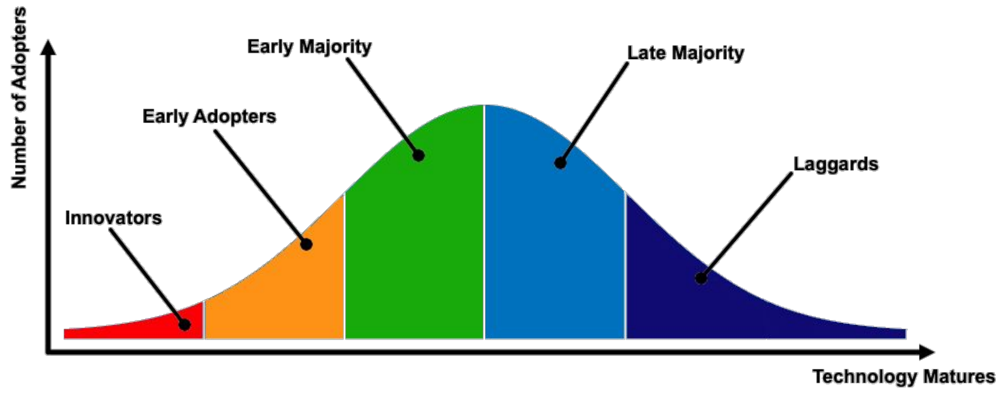
When would you rewrite parts of the system?

- Products and services are **out of support** or close to it
- No or very few people have **skillsets** for those products and services
- **Expensive** to run
- **Bad UX**
- Low **performance**
- ...

When would you keep parts of the system?

When would you keep parts of the system?

- It works
- Critical part of the system
- Products and services are still supported
- The technology is still widely used
- Replacement would cause expensive downtime
- Big risk of failing
- Rewriting would mean huge investments



To run and innovate our business

We integrate

… because the world is not getting any simpler

- Balance
- Low risk & High gain
- Do it smart using
  - Standard communication protocols
  - Standard data formats
  - Proofed integration patterns
- Stay agile and choose the best architectures for any given problem
  - Microservices, Service mesh
  - Event-driven
  - Hierarchical
  - Hexagonal
  - Gateway
  - Serverless
  - …

Red Hat

Evolution of Integration

Architecture

Point to Point

Direct connection between systems, application both internally and with external services

Enterprise Service Bus

Placing a centralized bus that integrate between loosely coupled services.

Microservices

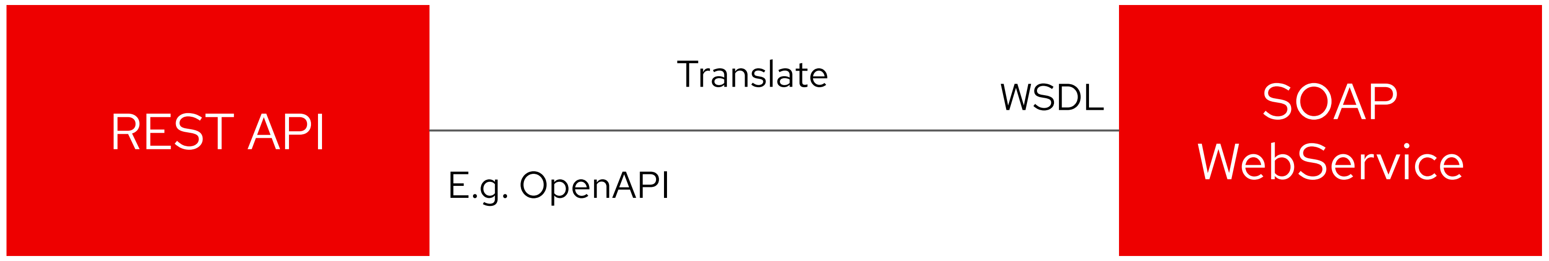
Fine grained distributed services, allowing faster turnover rate, more agile and flexible deployment model.

Serverless

Scale down to zero. Optimize Resource Usage. Avoid random, arbitrary workload prediction

What are some of the common integrations today?

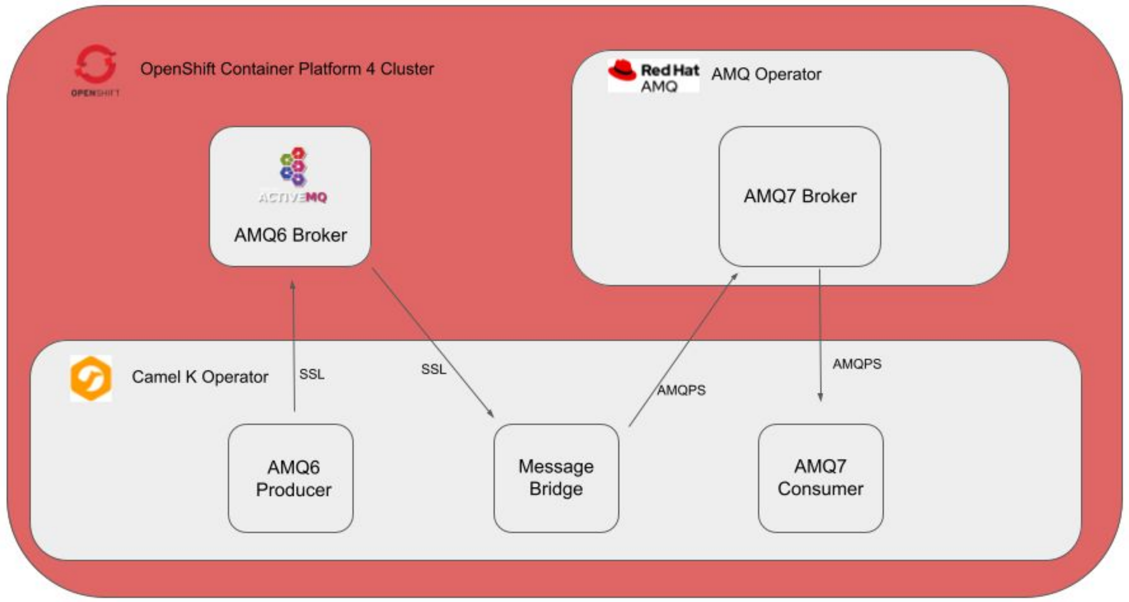
Expose legacy SOAP web services using REST API



What are some of the common integrations today?

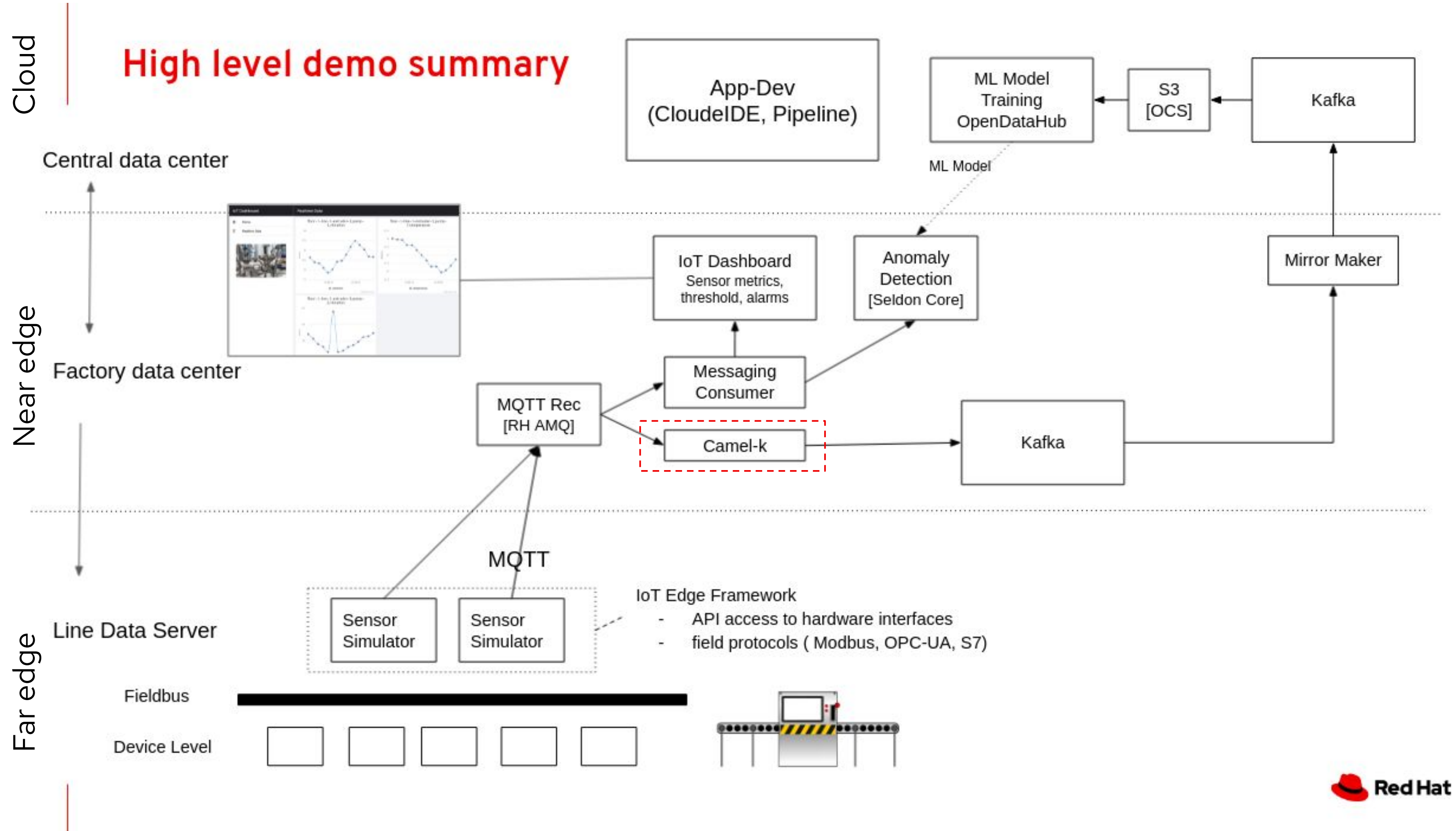
Bridge messages between different Broker technologies

- Plenty of vendors and products
- Various message types and protocols
  - JMS
  - AMQP
  - MQTT
  - Kafka
  - CloudEvent



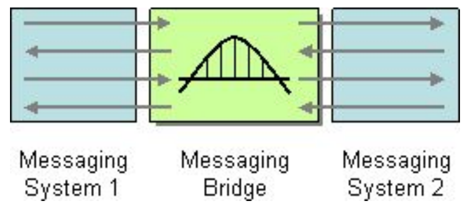
FACTORY EDGE SYSTEM

Example



machine condition monitoring based on sensor data in an industrial setting, using AI/ML

Camel as messaging bridge between MQTT and Kafka events

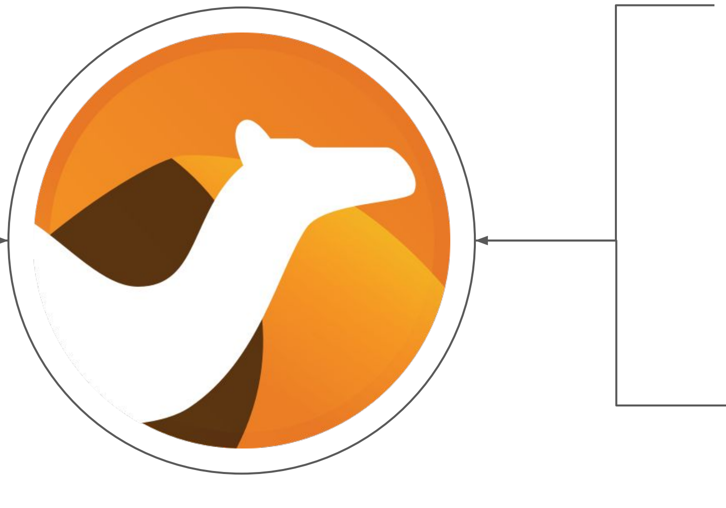


INTEGRATION PATTERNS

- Data handling
- Data model translation
- Data synchronisation
- Publish subscribe
- Routing
- Content based routing
- Data enrichment
- Protocol and transport translation
- Large object management
- Correlation
- Assured delivery
- Canonical data model
- Relationships/cross referencing
- Once only delivery
- Scatter gather
- Aggregation
- Store and forward
- Flow control/throttling
- Service exposure
- Failed event management
- Idempotence resolution
- Retry
- Health check
- Prioritization
- Optimistic/pessimistic locking
- Event sequencing
- Verb conversion
- Existence checking
- Compensation
- Batch processing (various)
- Command controller (various)
- Error handling (various)
- Composition (various)
- Orchestration (various)

$$\begin{tikzcd}\text{High-level vision of integration} \arrow[r, "\text{Enterprise Integration Patterns}"'] & \text{System implementation}\end{tikzcd}$$

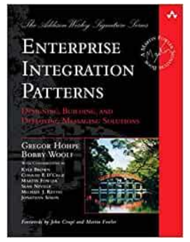
Apache Camel



That could connect to ~~any~~ almost any system

That can work on and off the cloud

This is
Apache Camel



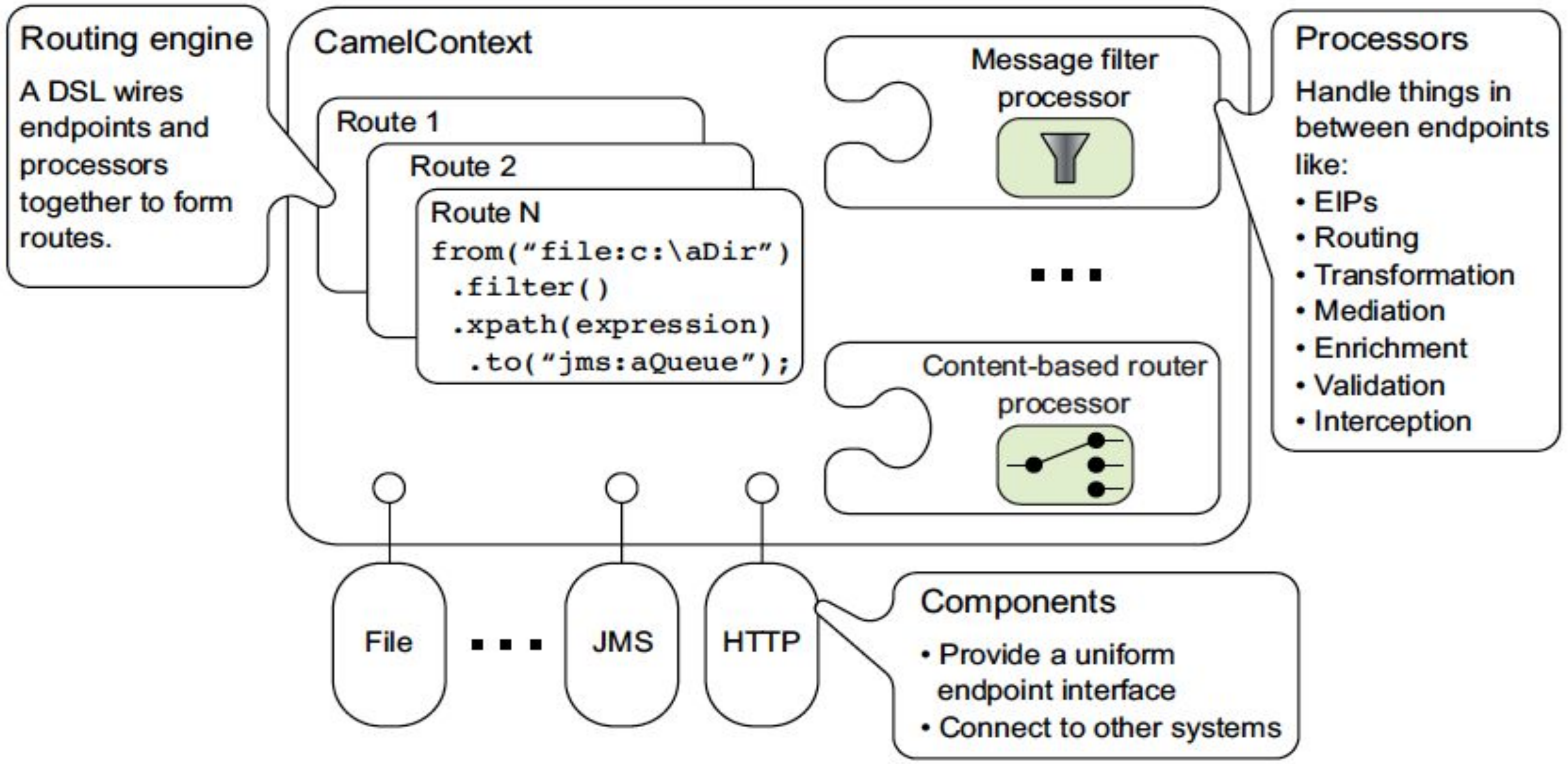
With support for known integration patterns

```
from("kafka:topic")
 .to("grpc:endpoint")
```

Integration defined in a simple language. Such as XML, YAML and Java

Apache Camel

Basics



Routing engine
A DSL wires endpoints and processors together to form routes.
CamelContext
Route 1
Route 2
Route N
Message filter processor
Content-based router processor
Processors
Handle things in between endpoints like:
• EIPs
• Routing
• Transformation
• Mediation
• Enrichment
• Validation
• Interception
File
JMS
HTTP
Components
• Provide a uniform endpoint interface
• Connect to other systems

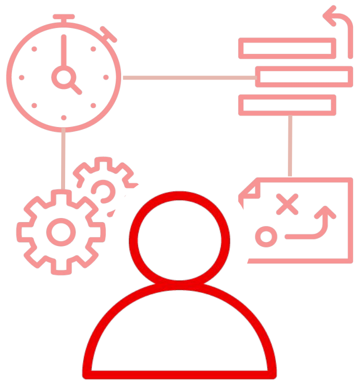
```
from("file:c:\aDir")
 .filter()
 .xpath(expression)
  .to("jms:aQueue");
```

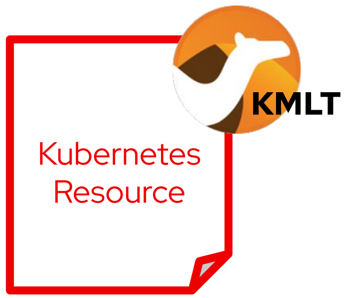
Kamelet Concepts

Kamelets are **use case driven**, they are Camel recipes encapsulating a well defined purpose

Use case story…

… implemented as a Kamelet





Types of Kamelets

Sources

Bring data from external systems to the platform

Sinks

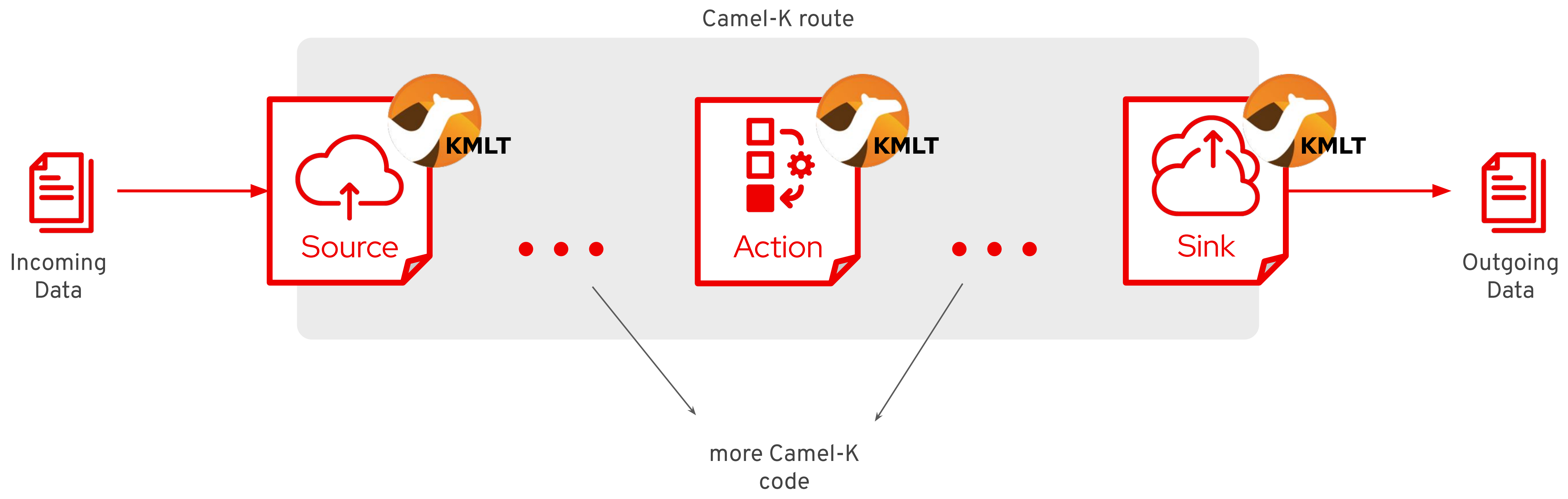
Forward data from the platform to external systems.

Actions

EIPs, transformation, etc

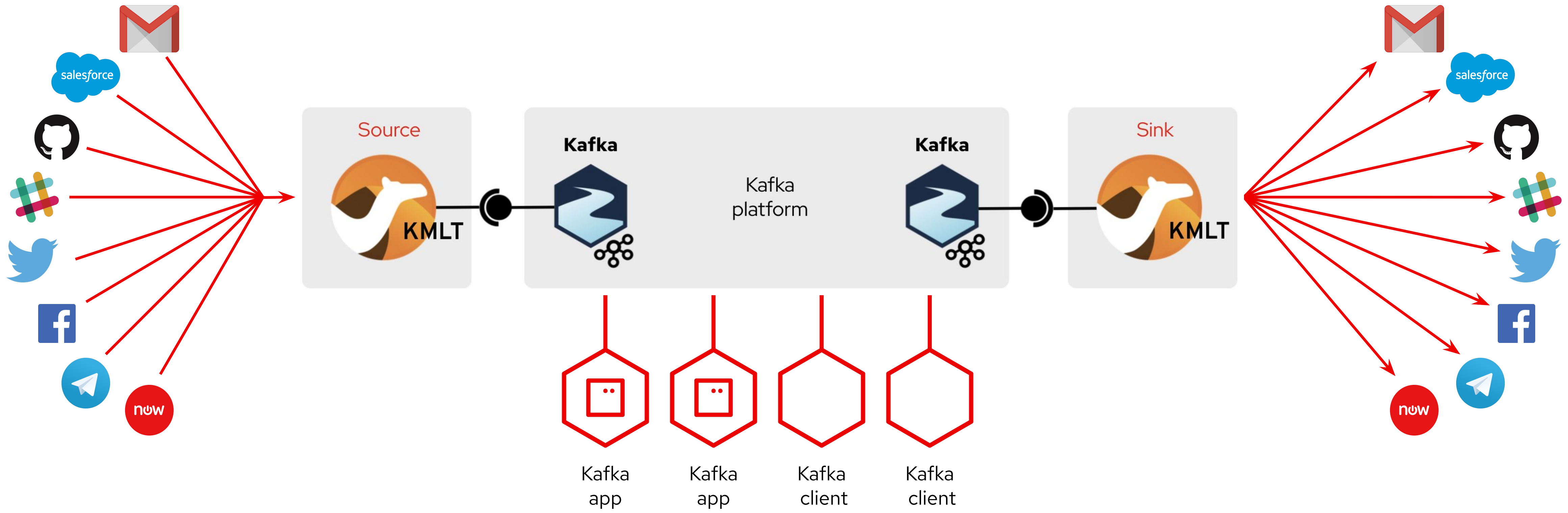
Camel with Kamelets

Use Kamelets where you need, as sources, sinks, or mid-flow actions

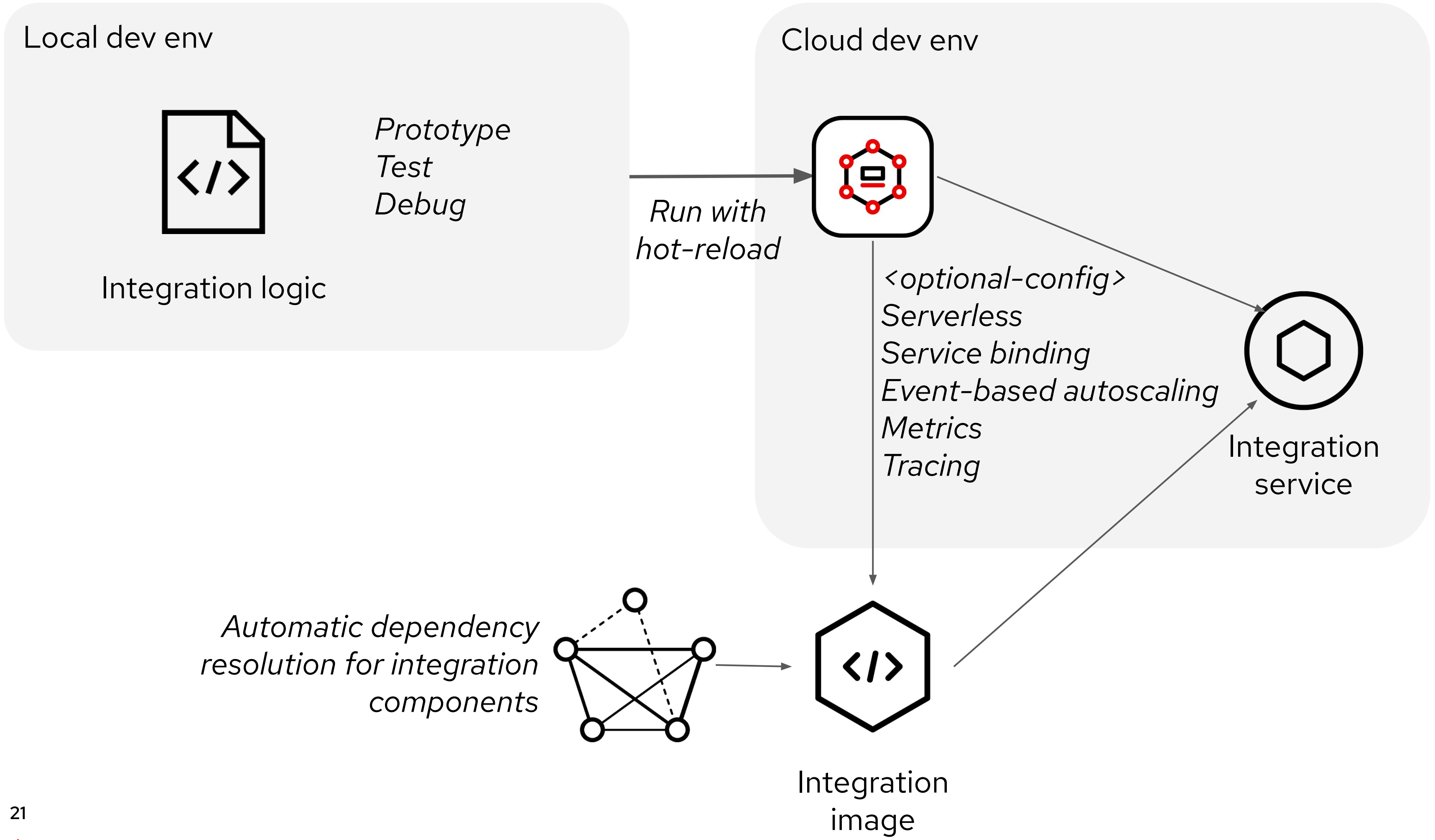


Kamelet Bindings

bind Kamelets to form a running integration unit



EXAMPLE CLOUD-NATIVE INTEGRATION JOURNEY

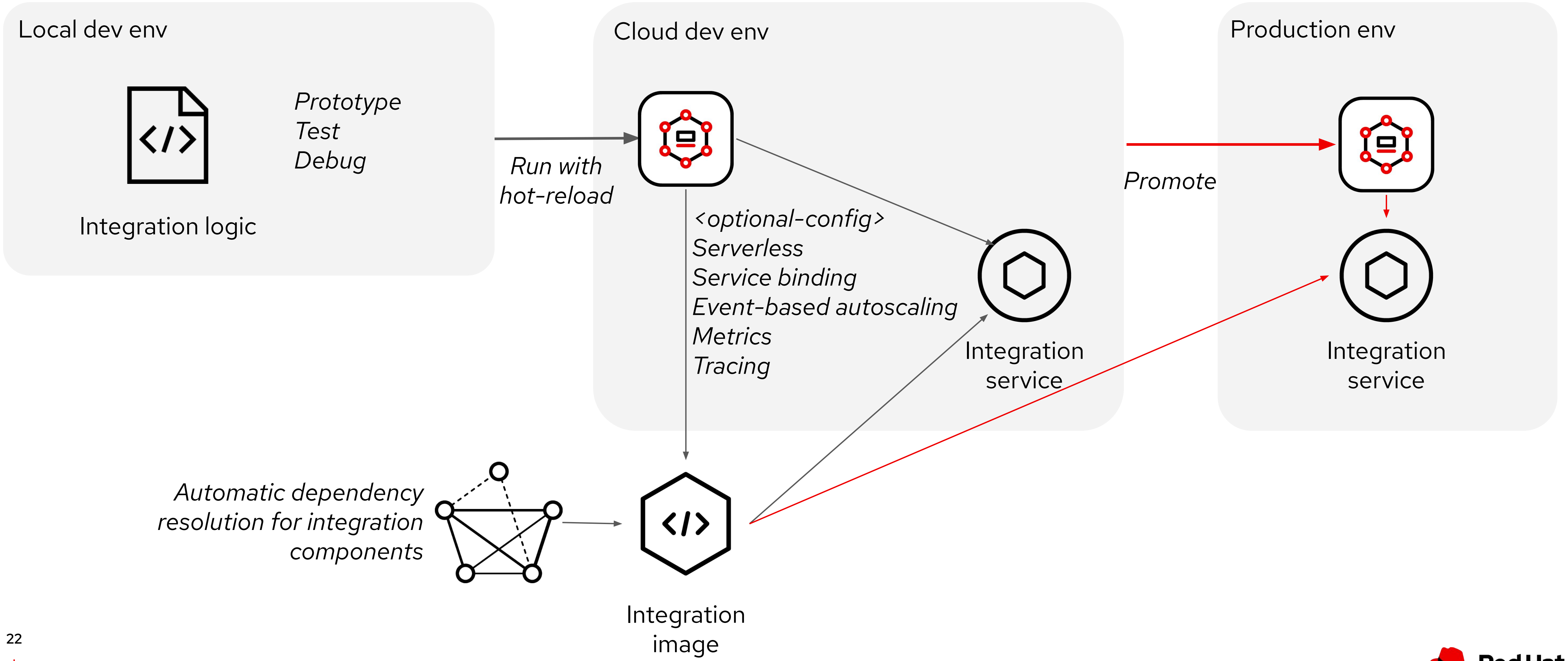


kafka-http.yam

```yaml
- from:
    uri: kafka:example-topic
    steps:
    - log:
        message: ${body}
        logging-level: INFO
    - to:
        uri: ${TARGET_URL}
```

```
> kamel run kafka-http.yaml --dev
```

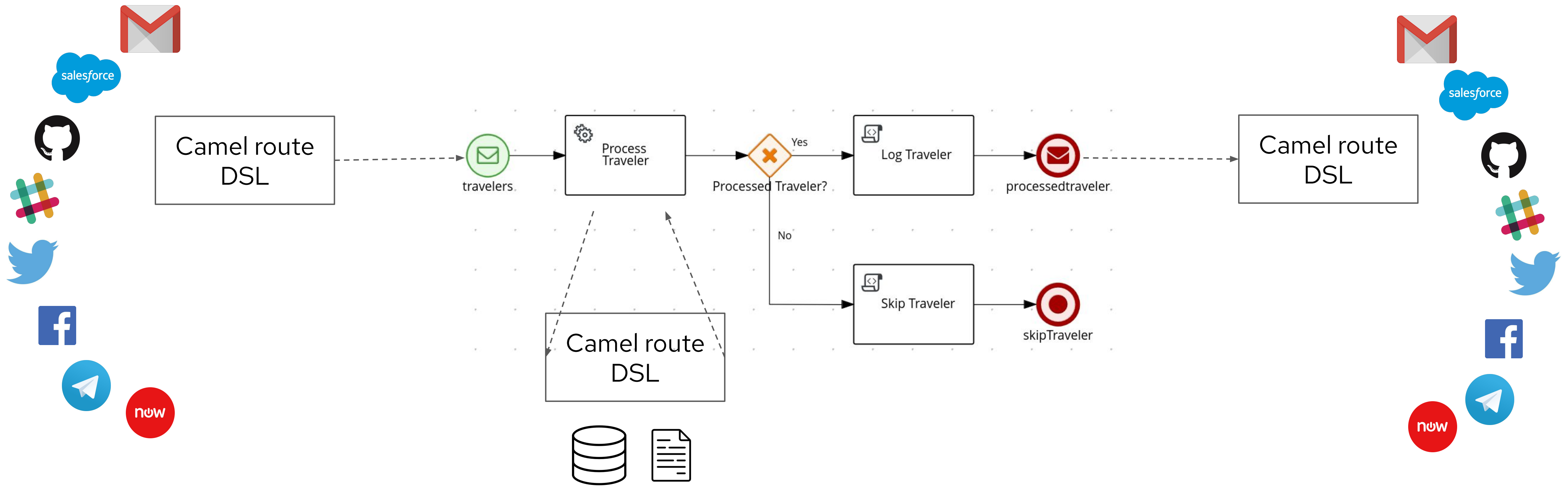
EXAMPLE CLOUD-NATIVE INTEGRATION JOURNEY



Camel in your projects

- Run integration service(s) in a local environment
- No programming but writing integration DSL

Routing, Transformation, Processing, Filtering, ...



Camel route example

HTTP -> Log

```
<?xml version="1.0" encoding="UTF-8"?>

<routes xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xmlns="http://camel.apache.org/schema/spring"
       xsi:schemaLocation="
           http://camel.apache.org/schema/spring
           https://camel.apache.org/schema/spring/camel-spring.xsd">

   <route id="hello">
       <from uri="platform-http:/hello"/>
       <setBody>
           <simple>Hello ${body} from Camel</simple>
       </setBody>
       <log message="${body}"/>
   </route>

</routes>
```

Summary

- Technologies evolve and businesses need to adapt effectively through integration in order to innovate and stay competitive
- Apply well defined practices and patterns to integrate services and systems
- Use standard communication protocols, messaging and data formats
- Stay agile to combine and use mix of architectures
- Apache Camel is the most popular integration framework, open source and free, bookmark it

CAMEL LAB

Installation

- JBang - `curl -Ls https://sh.jbang.dev | bash -s - app setup`"
- Camel JBang - `jbang app install camel@apache/camel`

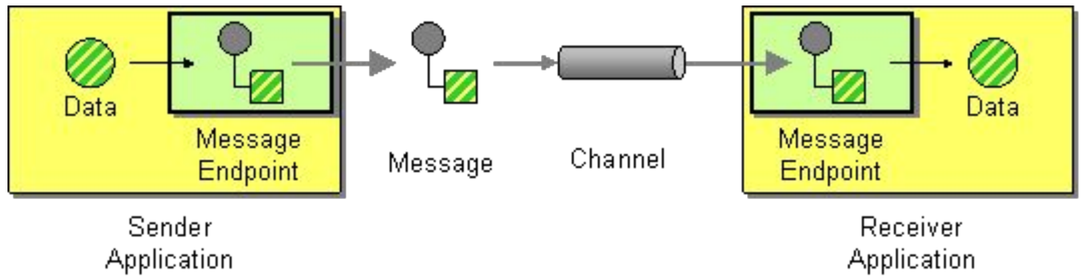
- Run and Verify

```
$ camel version

Camel JBang version: 3.Y.Z
```

Create your first integration service

- Platform HTTP /hello -> Log “Hello ${body} from Camel”

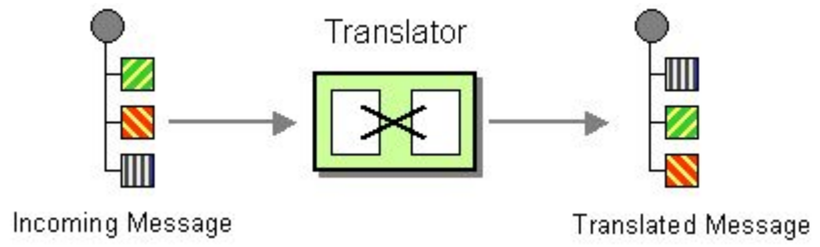


```
$ camel init hello.xml
$ camel run hello.xml
$ curl -X POST http://0.0.0.0:8080/hello -H "Content-Type:
text/plain" -d "Ivo"

Hello Ivo from Camel
```

Read json property

- Platform HTTP /hijson -> Log “Hi ${body[name]} from Camel”



```
<setHeader name="Content-Type">
    <constant>application/json</constant>
</setHeader>
<unmarshal>
    <json/>
</unmarshal>

$ curl -X POST http://0.0.0.0:8080/hijson -H "Content-Type:
application/json" -d "{\"name\":\"Ivo\"}"

Hi json Ivo from Camel
```

Prepare DB

- SQLite

```
$ sqlite3 data.db
CREATE TABLE users (
   id INTEGER PRIMARY KEY,
   name TEXT NOT NULL,
   UNIQUE("name")
);
```

application.properties

```
camel.beans.myDataSource = #class:org.sqlite.SQLiteDataSource
camel.beans.myDataSource.url =
jdbc:sqlite:/home/ibek/git/pv207/data.db

camel.component.sql.dataSource = #bean:myDataSource
```

Integrate with DB

- SQLite
- Platform HTTP /users POST -> **SQL Insert**

```
<convertBodyTo type="java.util.Map"/>
<to uri="sql:INSERT INTO users (name) VALUES (:#name)" />
<setBody>
    <simple>User ${body[name]} added successfully</simple>
</setBody>

$ curl -X POST http://0.0.0.0:8080/users -H "Content-Type:
application/json" -d "{\"name\":\"Ivo\"}"

User Ivo added successfully
```

```
$ camel run integration.xml application.properties --deps=org.xerial:sqlite-jdbc:3.41.2.1 --dev
```

Integrate with DB

- SQLite
- Platform HTTP /users/{user} GET -> **SQL Select**

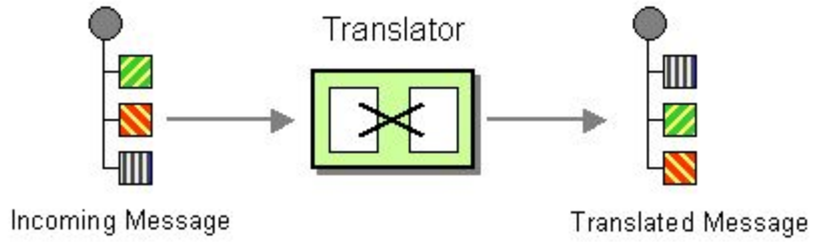
```
<to uri="sql:select * from users where name = :#user?outputType=SelectOne" />
<marshal>
     <json library="Jackson" />
</marshal>

$ curl http://0.0.0.0:8080/users/Ivo

{"id":1,"name":"Ivo"}
```

Transform json

- JQ - remove id property



```
<setBody>
    <jq>
       del(.id)
     </jq>
</setBody>

$ curl http://0.0.0.0:8080/users/Ivo

{"name":"Ivo"}
```

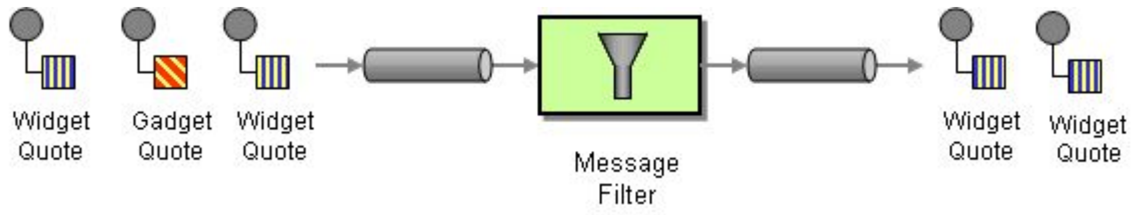
Direct routes

The Direct component provides direct, synchronous invocation of any consumers when a producer sends a message exchange.

This endpoint can be used to connect existing routes in the same camel context.

```
<from uri="direct:addUser"/>
<to uri="direct:addUser" />
```

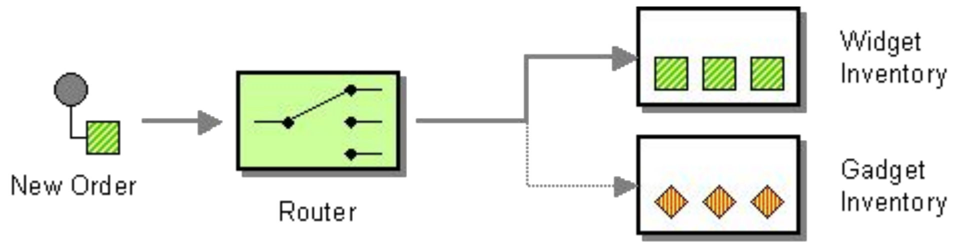
Filter EIP



```
<filter>
    <simple>${body[name]} !regex '.*-bot'</simple>
    <log message="Filtered body: ${body}"/>
    <to uri="direct:addUser" />
</filter>

$ curl -X POST http://0.0.0.0:8080/filteredusers -H "Content-Type: application/json" -d
"{\"name\":\"John-bot\"}"
```

Choice EIP



```
<choice>
    <when>
          <simple>${body[total]} &lt; 100</simple>
          <log message="Sending to standard processing department"/>
    </when>
    <otherwise>
          <log message="Sending to expedited processing department"/>
    </otherwise>
</choice>

$ curl -X POST http://0.0.0.0:8080/order -H "Content-Type: application/json" -d
"{\"total\":50}"
```

Learn more about Camel

- https://camel.apache.org
- More integration pattern examples at: https://camel.apache.org/components/3.20.x/eips/enterprise-integration-patterns.html
- More components to integrate with: https://camel.apache.org/components/3.20.x/

Thank you

Red Hat is the world's leading provider of enterprise open source software solutions. Award-winning support, training, and consulting services make Red Hat a trusted adviser to the Fortune 500.

linkedin.com/company/red-hat

youtube.com/user/RedHatVideos

facebook.com/redhatinc

twitter.com/RedHat