

The background is a dark blue gradient representing a night sky. A large, bright white circle representing the moon is in the upper left. Several small white dots representing stars are scattered across the sky. A shooting star with a long white tail is in the middle right. In the foreground, there are dark blue silhouettes of rolling hills. On the left, a saguaro cactus stands on a hill. In the center, another saguaro cactus stands on a hill. On the right, a caravan of three camels is walking across a hill. In the bottom left corner, there are some light blue geometric shapes representing rocks or a small structure.

Low-code Visual Integration Design with Apache Camel Karavan

What is it all about?

❑ You are

- ❑ Solution architect
- ❑ Software developer
- ❑ Site Reliability Engineer

❑ You have to

- ❑ send metrics from my local applications to CloudWatch
- ❑ create a ticket in Redmine if there is an alert in SQS
- ❑ transfer all messages from our JMS brokers to Kafka

❑ You want to

- ❑ solve it fast
- ❑ make a solid solution
- ❑ have fun

We can help



Marat Gubaidullin

- ❑ Apache Camel Karavan developer
- ❑ Solution architect at Red Hat
- 🐦 @mgubaidullin



Claus Ibsen

- ❑ Apache Camel PMC
- ❑ Senior Principal Engineer at Red Hat
- 🐦 @davsclaus

Apache Camel



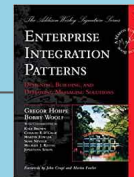
That could connect to **any**
almost any system



That can work on and off
the cloud



This is
Apache Camel

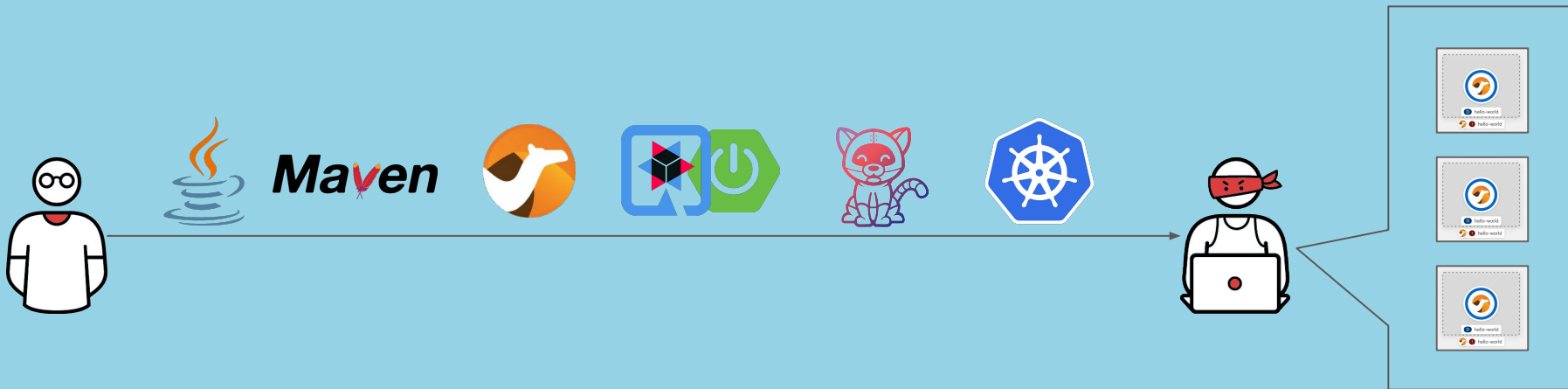


With support for known integration
patterns

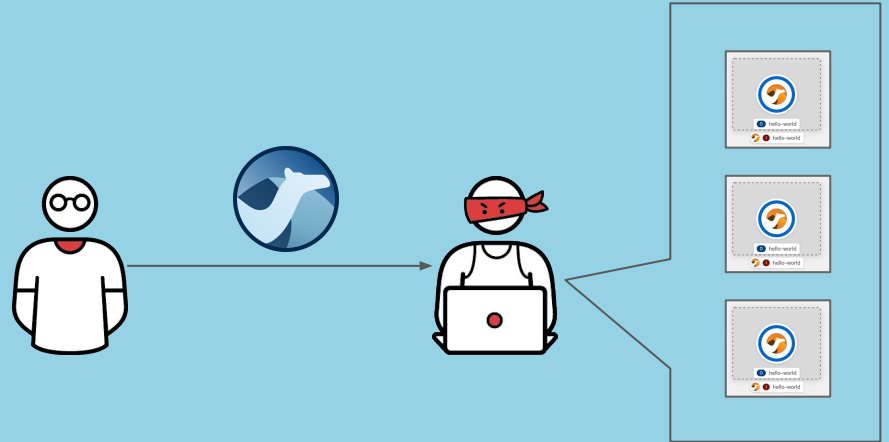
```
from("kafka:mytopic")  
.to("jms:cheese")
```

That can integrate with a
simple language. Such as
XML, YAML and Java

Learning path

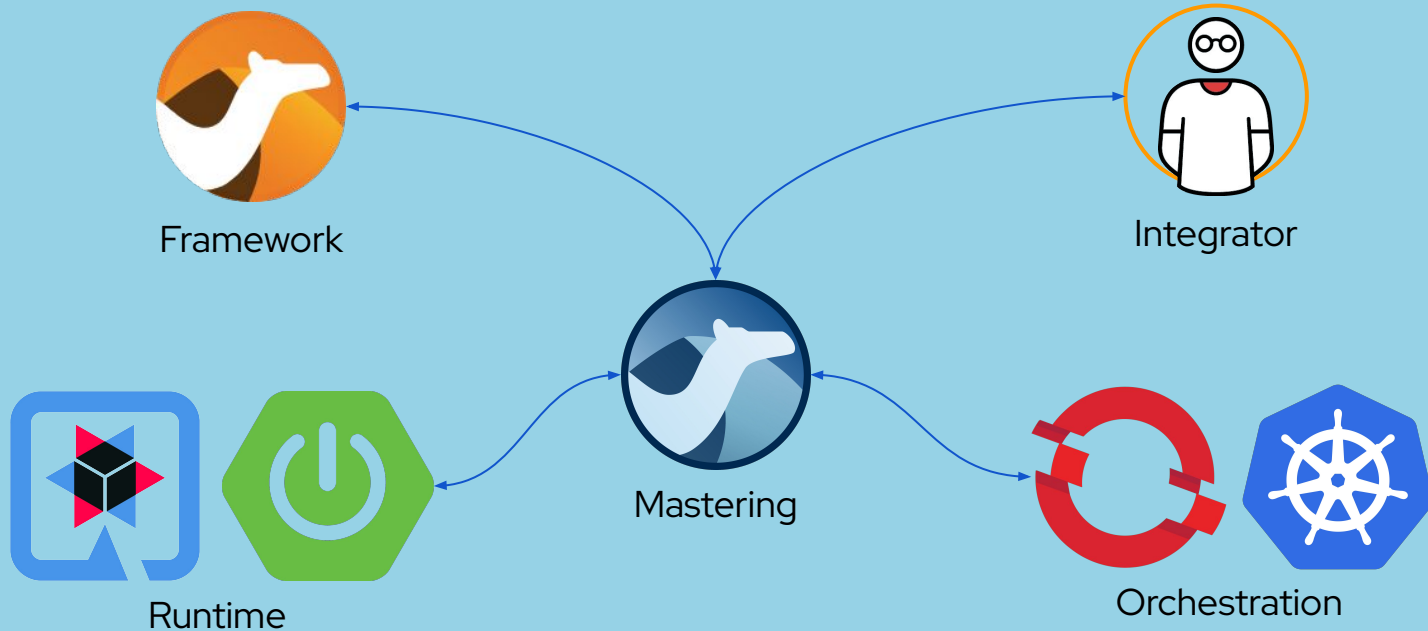


Learning path



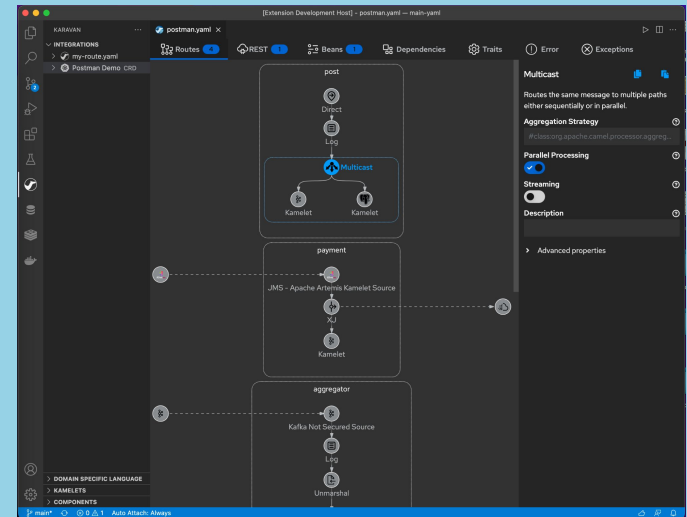
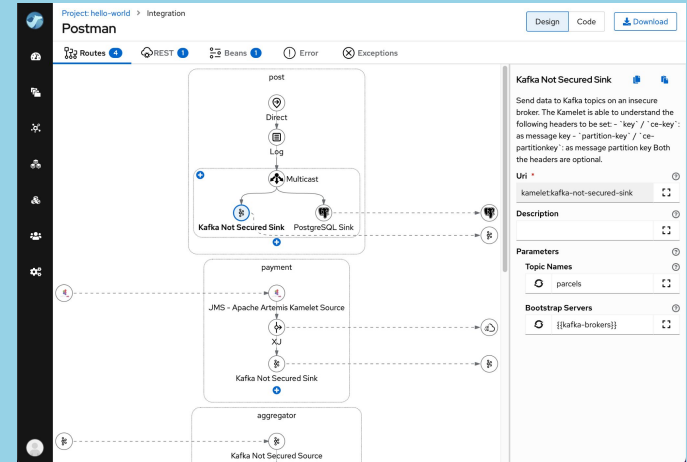
Apache Camel Karavan

Karavan is an Integration Toolkit for Apache Camel, which makes integration easy and fun through the **visualization** of pipelines, integration with **runtimes** and **package**, image **build** and **deploy** to kubernetes **out-of-the-box**.

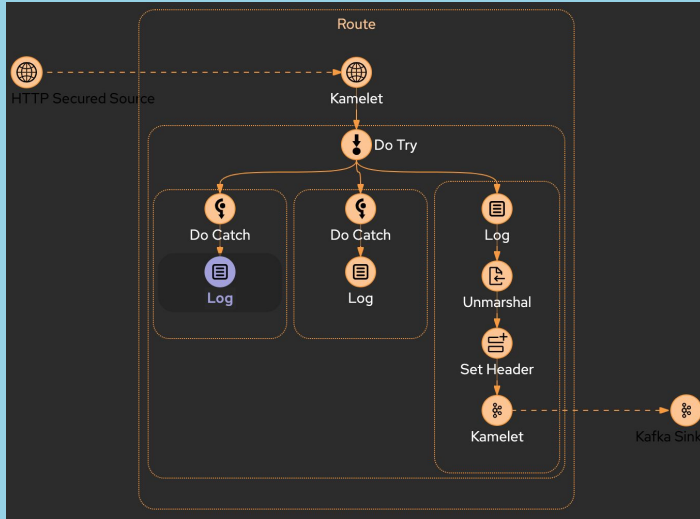


Designer

- ❑ **Visual** Integration Process development tool
- ❑ **Enterprise Integration Patterns**
- ❑ **REST DSL**
- ❑ Build-in catalogue of **Kamelets** and **Components**
- ❑ Configurable **DataFormats**



Collaboration



```
from("kamelet:http-secured-source")
    .doTry()
    .log("${headers}")
    .unmarshal().json()
    .setHeader("name", constant("claus"))
    .to("kamelet:kafka-sink")
    .doCatch(IOException.class)
    .log("${body}")
    .doCatch(CamelExchangeException.class)
    .log("${body}")
    .end();
```



← User Interface paradigm reflects Camel DSL →

Build and deploy

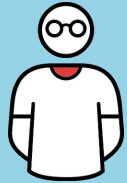
Local development environment

- ❑ VS Code extension
- ❑ Camel JBang

Cloud development environment

- ❑ Web application
- ❑ Kubernetes
- ❑ Tekton

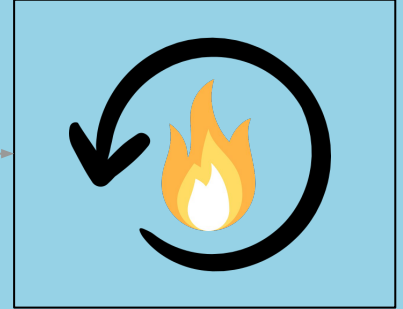
Local development environment



YAML/s
java, groovy,
properties

localhost

```
jbang camel@apache/camel run *
```



Camel JBang



- ❑ Camel **CLI**
- ❑ **Quickly run** Camel integration file(s)
- ❑ Automatic **dependency management** (no Maven or Gradle)
- ❑ All **DSLs** (java xml yaml groovy kotlin)
- ❑ **Hot reload** on source change
- ❑ **Export** to Spring Boot or Quarkus project (with Maven pom.xml)

Camel JBang



New in Camel 3.19

- ❑ **Control** running integrations
(list, start, stop, etc.)
- ❑ Control **Spring Boot** & **Quarkus**
Camel applications
- ❑ Runtime **statistics**
- ❑ **Top** routes and EIPs
- ❑ Show Camel route **source**
- ❑ View **catalog** of out of the box artifacts
- ❑ View **documentation**
- ❑ Launch **hawtio** web console

Camel JBang Demo

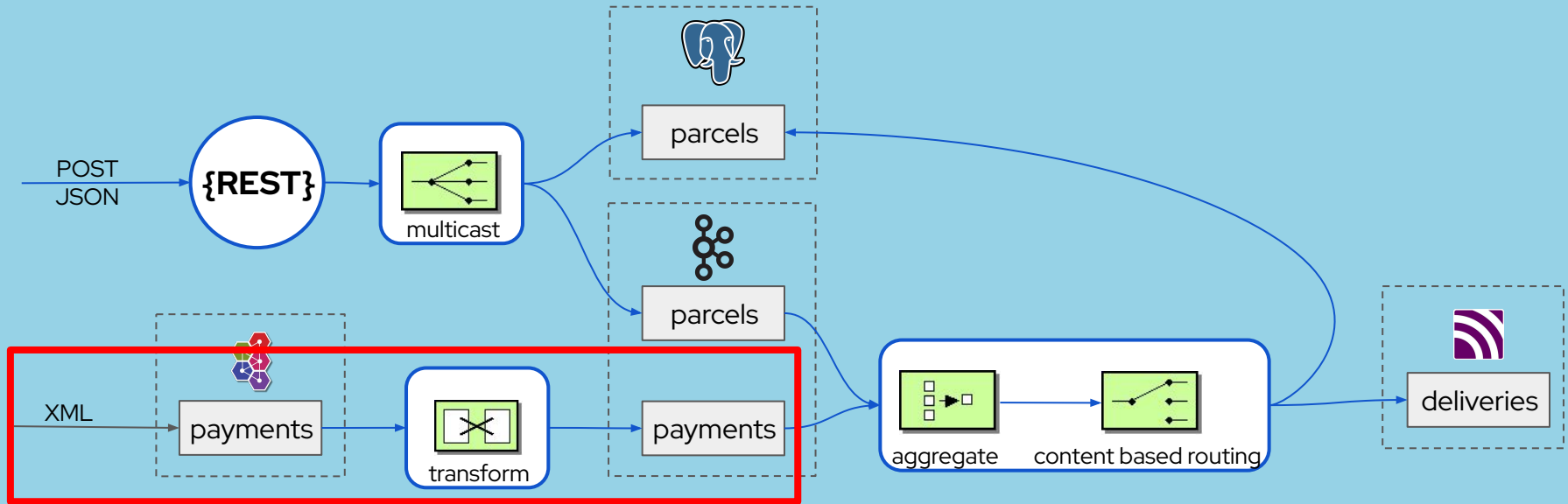


```
17% ██████████ 9.1 GB | P | □ -
~ > jbang -V
0.98.0
~ > camel -V
3.19.0-SNAPSHOT
~ > amel
```

```
ac2022.java
1 // camel-k: language=java
2
3 import org.apache.camel.builder.RouteBuilder;
4
5 public class ac2022 extends RouteBuilder {
6
7     @Override
8     public void configure() throws Exception {
9
10        // Write your routes here, for example:
11        from("timer:java?period={{time:1000}}").routeId("java")
12            .setBody()
13            .simple("Hello Camel from ${routeId}")
14            .log("${body}");
15    }
16 }
17
```

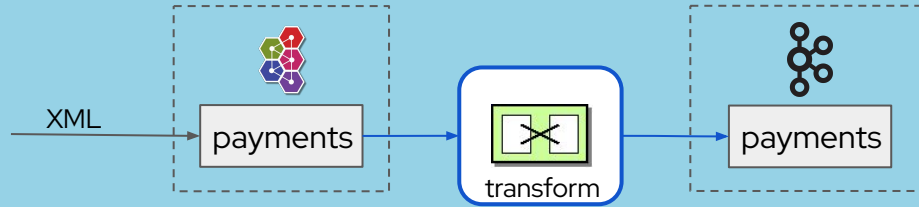
```
watch camel get
Every 2.0s: camel get
davsclaus-mini.local: Tue Sep 27 18:44:02 2022
```

Parcels Demo



1. REST Service to store **parcel** to database and publish to Kafka
2. Consume **payment** from JMS, Transform XML to JSON and publish to Kafka
3. Aggregate **parcel** and **payment**. If payed send **delivery** to MQTT, otherwise update database

Local Karavan Demo



- Consume payment **from** JMS
- ... then transform XML **to** JSON
- ... and publish **to** Kafka

<https://github.com/apache/camel-karavan/tree/main/karavan-demo/jms-to-kafka>

Parcels Demo (VS Code)

The screenshot displays the VS Code interface for editing a Camel route in a file named 'payment.yaml'. The main editor shows a route diagram for 'payment' with the following components:

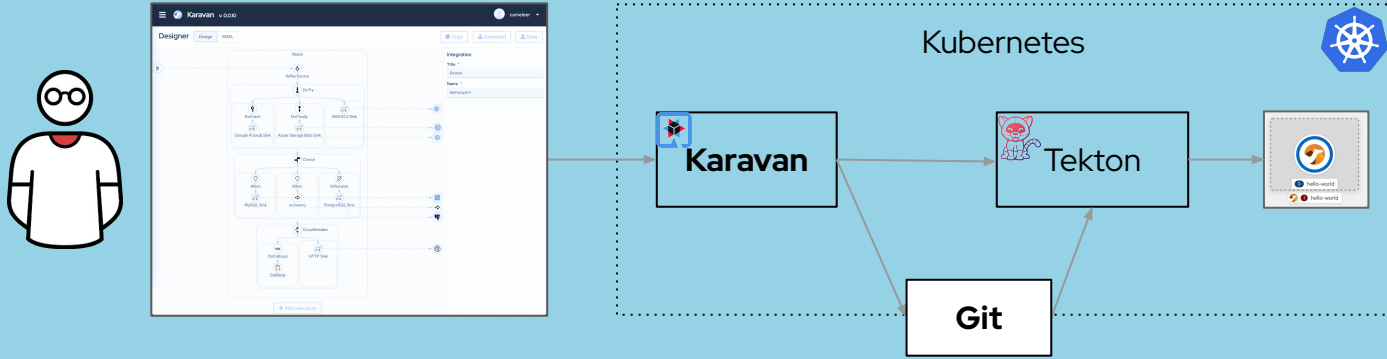
- JMS - Apache Artemis Kamelet Source**: The starting point of the route.
- XJ Identity Action**: An action that follows the source.
- Kafka Not Secured Sink**: The final destination of the route.

A dashed arrow indicates the flow from the source to the sink. Below the diagram is a '+ Create new route' button. The right-hand sidebar shows the configuration for the selected 'JMS - Apache Artemis Kamelet Source' component:

- Uri**: `kamelet:jms-apache-artemis-source`
- Id**: (Empty)
- Description**: (Empty)
- Parameters**:
 - Destination Type**: `queue`
 - Destination Name**: `payments`
 - Broker URL**: `tcp://localhost:61616`

<https://github.com/apache/camel-karavan/tree/main/karavan-demo/jms-to-kafka>

Cloud development environment



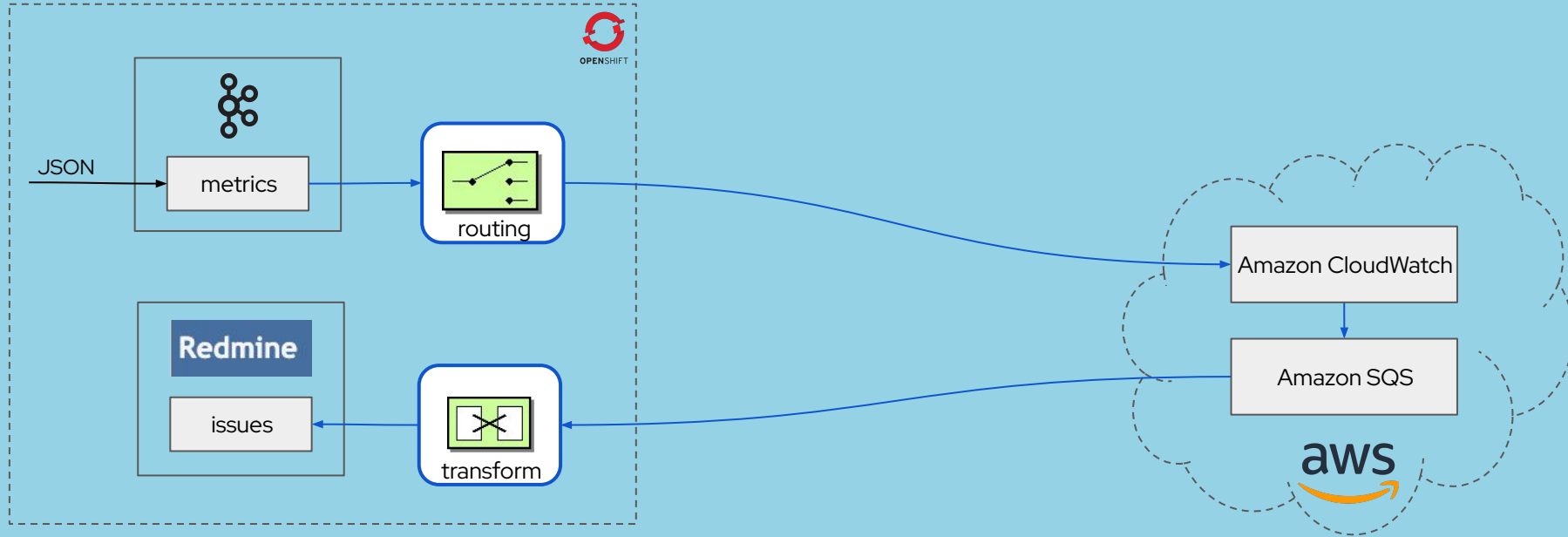
Project in Karavan

- ❑ Zero or multiple YAMLS
- ❑ Zero or multiple Java/Groovy files
- ❑ application.properties

Build and Deploy

- ❑ On Run Karavan creates Pipeline
- ❑ Run using pre-configured Pipeline
(Karavan provides default Pipeline, however you can replace it)

Demo



1. Receive metrics **from Kafka** topic and publish **to CloudWatch**
2. Consume messages **from SQS** and create an issue **in Redmine**

1 Year Anniversary



Aug 4, 2021: First UI prototype pushed to GitHub



Oct 4, 2021: Karavan become **Apache** Camel Karavan

Contributors 13

A grid of 13 circular icons representing contributors. The first row contains 7 icons: 5 profile pictures and 2 logos (GitHub and a red devil). The second row contains 6 icons: 4 profile pictures and 2 logos (a purple and white checkerboard pattern and a red devil).

...

Karavan Preview

Apache Camel | 1,918 installs

☆ 158 stars

👁 25 watching

🍴 44 forks



Oct 4, 2022: Karavan on **ApacheCon** North America

Roadmap Karavan



- ❑ Visual **debugging** and **tracing**
- ❑ **Monitoring** and **dashboards**
- ❑ Camel **Error Handler**



- ❑ **Propagation** between environments (test, prod)
- ❑ Kubernetes **Operator**
- ❑ **Role-Based Access Control**



- ❑ **Web-based IDE** (Gitpod, DevSpaces, GitHub Codespaces, etc)
- ❑ **Data mapping** (AtlasMap?)

Roadmap Camel JBang



- Coloured** output
- Show **logs**
- List **inflight** or **blocked** messages



- SOAP** generator
- Dependency downloader switch to Maven Aether
- Easy to switch between **Camel versions**



- Native** CLI binary



Hanbing Wang | WDOArts

Thanks