



Develop and Connect Cloud Native Applications

Red Hat RHAF - CAMEL
DEMO

DEMO use case

DEMO use case – Objectives

What CAMEL provides:

- OData and REST APIs
- Tooling for developing
- Orchestration: example of Integration Pattern implementation between two systems:
 - Multicast
 - Aggregation
- Deployment in OpenShift

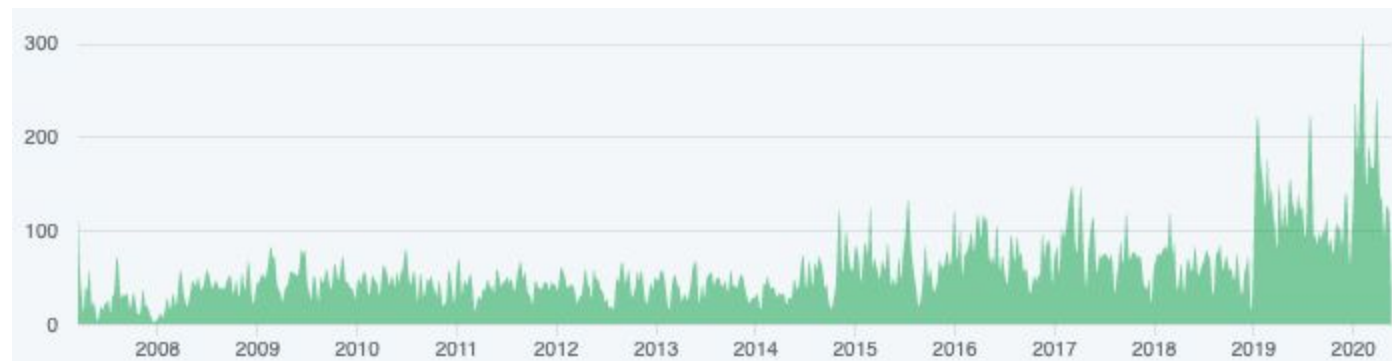


What is Apache Camel?

- ▶ Open source Swiss Army knife **framework** for integration
- ▶ 340+ **components** allow to talk to external systems like: Salesforce, AWS S3 Storage Service, etc.
- ▶ Enables more connectivity with **data formats** (JSON, XML, CSV, YAML, Avro, Protobuf...) and **protocols** including: AMQP, HTTP, SCP, etc.
- ▶ Transformation with more than 15 languages (JSONPath, XPath, SpEL, Groovy, Simple, XSLT...)
- ▶ Routes and **Enterprise Integration Patterns** (EIP) modeled for designing and developing integration solutions
 - Commonly used patterns like: Content-based router, Splitter, Aggregator, etc

Apache Camel Community

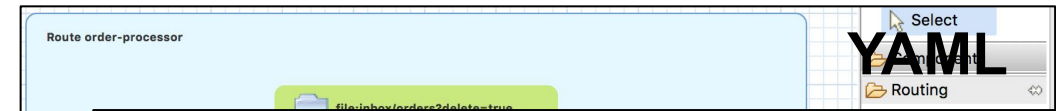
- ▶ The biggest and most active community for open source integration software
- ▶ 600+ unique contributors
- ▶ 3800+ pull requests (mostly from external contributors)
- ▶ 13+ years of development and one of the most active Apache projects
- ▶ The interface hasn't changed from the beginning
- ▶ 4.7k stars on Github in 2023



How to getting started?

- Camel supports multiple languages (Java, XML DSL, Java DSL, YAML DSL..)

CAMEL DSL

**YAML**

```
<from uri="file:inbox/orders?delete=true">
<choice>
  <when>
    <xpath>/order/product = 'widget'</xpath>
    <to uri="activemq:Orders.Widget">

```

XML

```
public class MyRoute extends RouteBuilder {
  from("file:inbox/orders?delete=true")
  .choice()
  .when().xpath("/order/product = 'widget'")
    .to("activemq:Orders.Widget")
  .otherwise()
    .to("activemq:Orders.Gadget");
}
```

JAVA

- Runtimes (Quarkus, Springboot)
- IDE → Visual Code
- Deployment on Standalone or on OpenShift (kubernetes)
- JBang or Maven plugins



Camel in Action

Camel routes



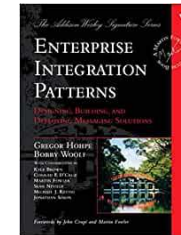
That could connect to ~~any~~
almost any system



That can work on and off
the cloud



This is
Apache Camel

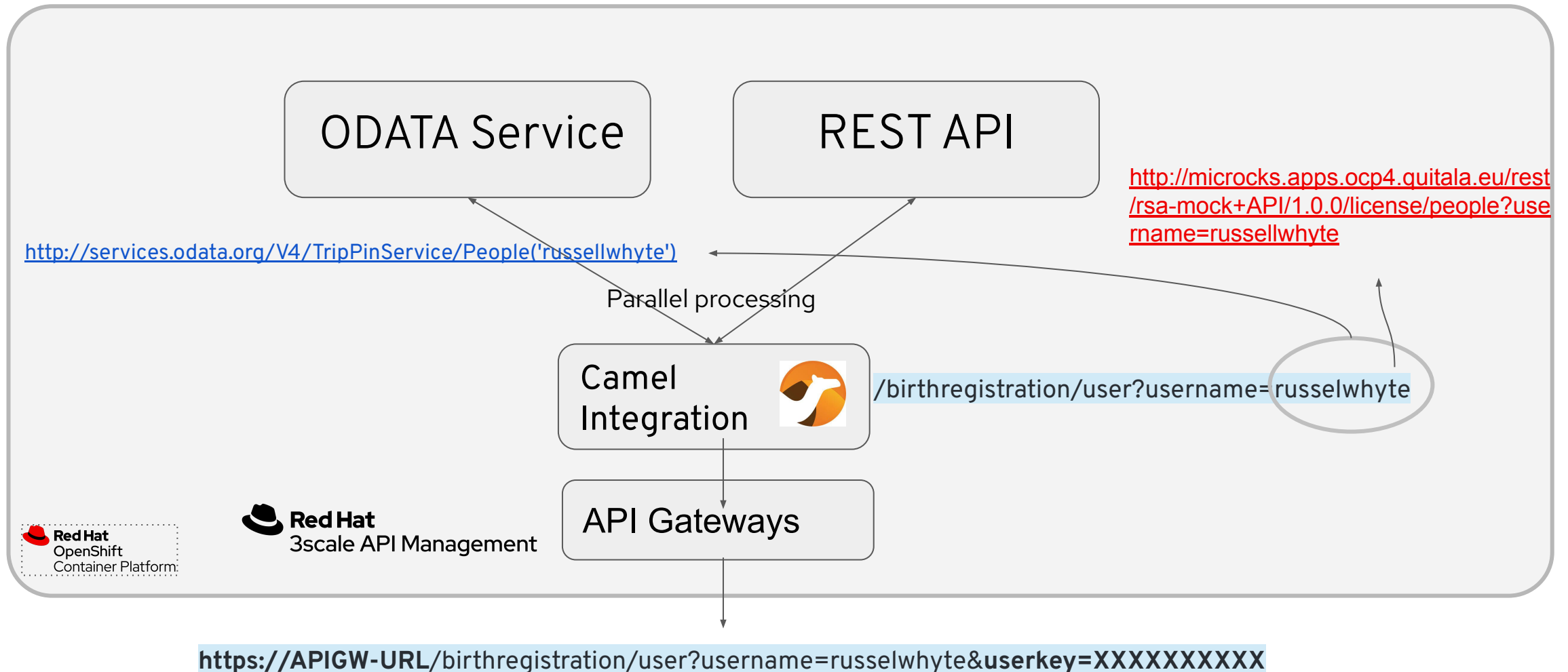


With support for known
integration patterns

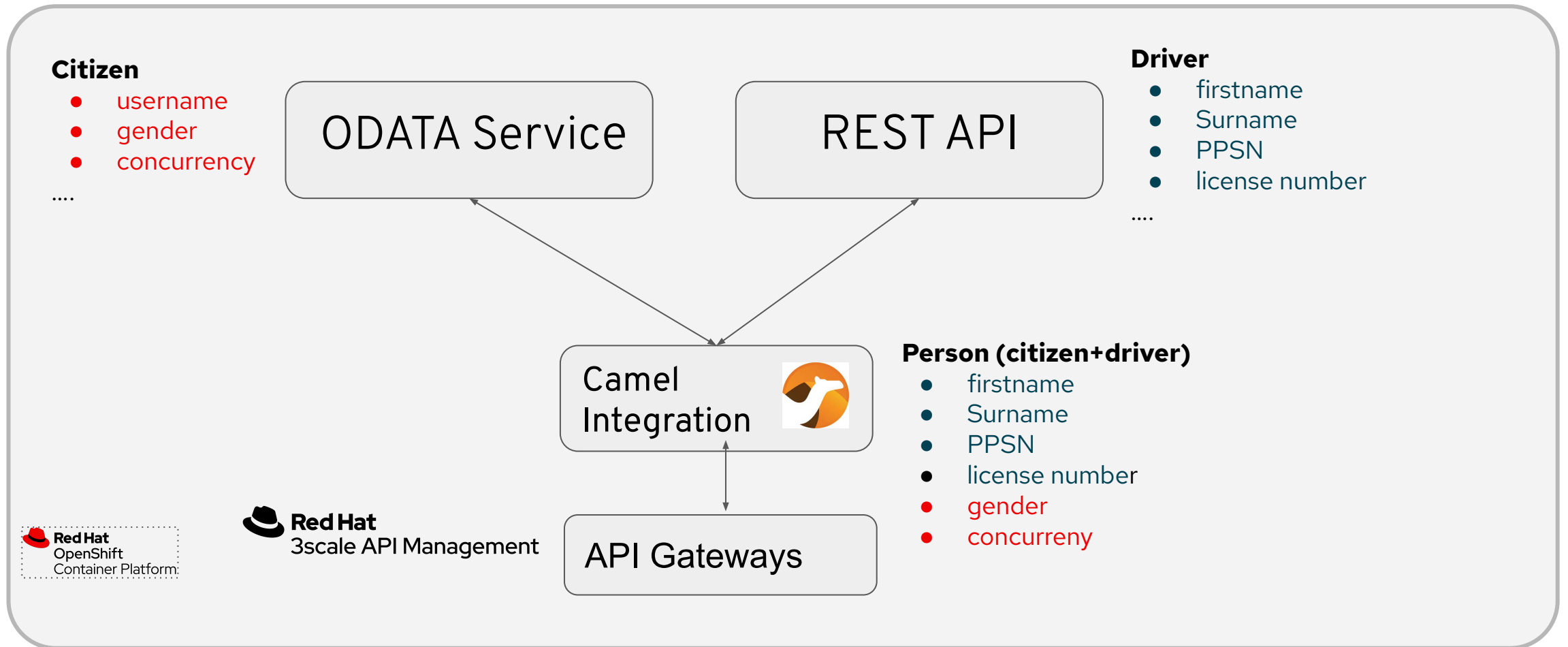
```
from("kafka:topic")
  .process {message}
  .to("s3bucket:endpoint")
  .to("http:endpoint")
```

Everything is Message
Starts with a Consumer (**from**)
Process or Integration pattern implementation
Establish target endpoints (**Tos**)

Demo Diagram - Service orchestration



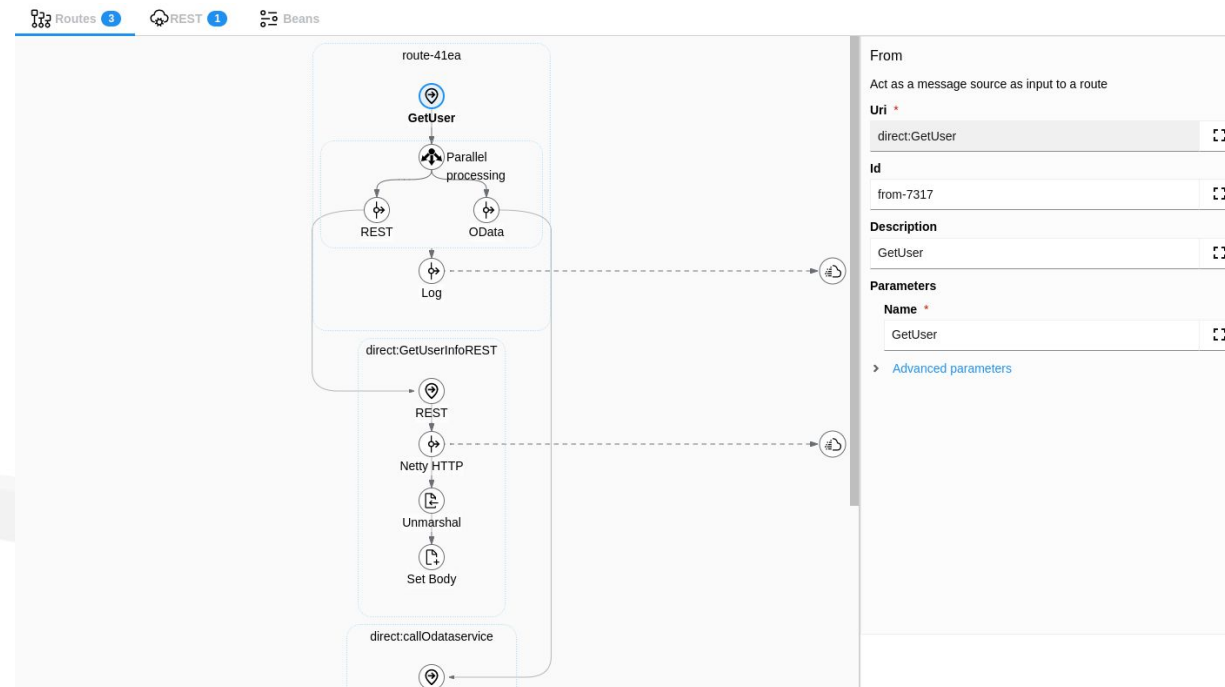
Demo Diagram - Data Model



How to getting started?

FAST PROTOTYPING

1. **Start with the YAML DSL and draw your integration routes with a Camel designer (Kaoto or Karavan).**
 - a. Visual Code plugins to create a design your camel routes.

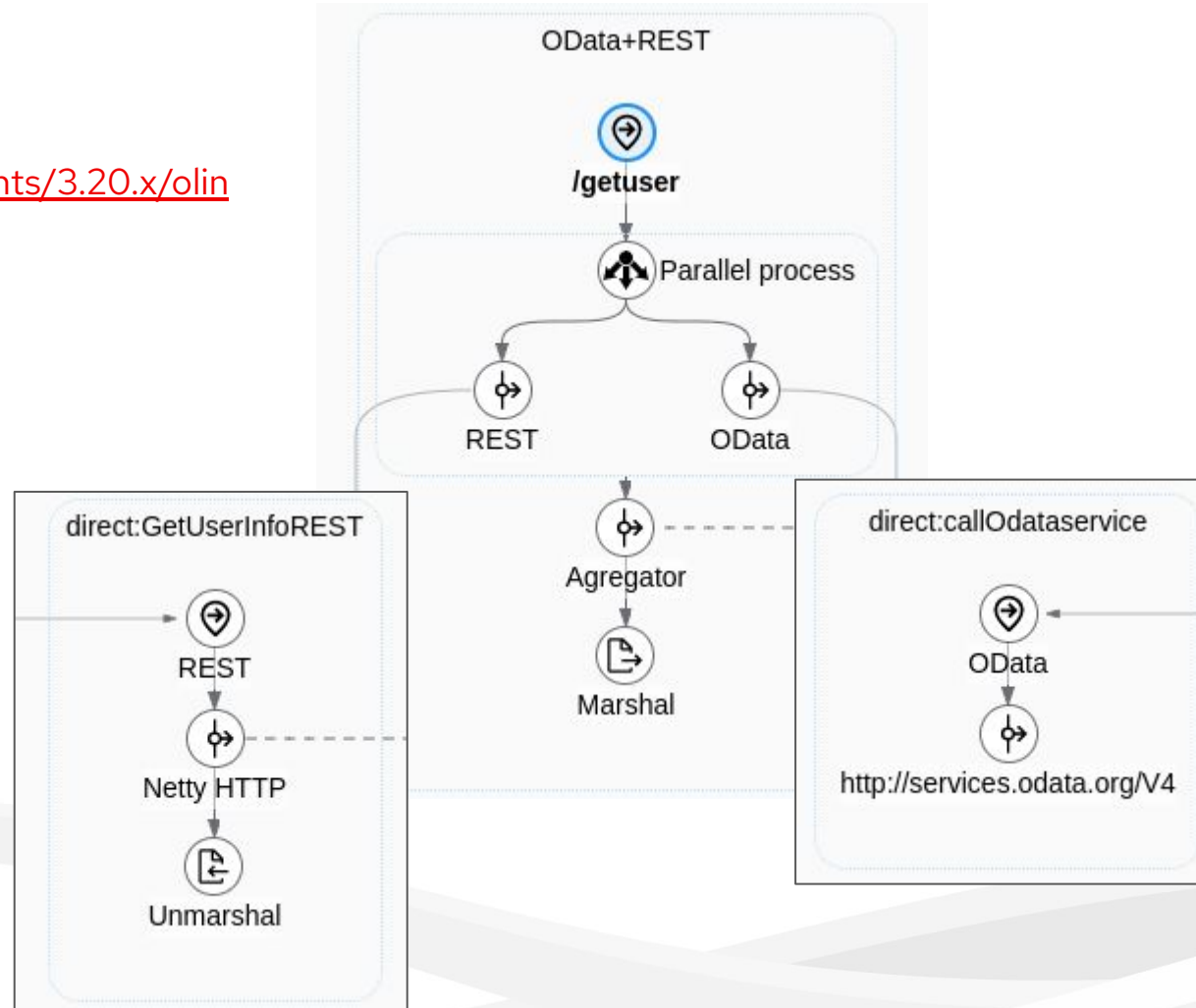


Camel Route

FAST PROTOTYPING

OData component:

<https://camel.apache.org/components/3.20.x/olingo4-component.html>



How to getting started?

FAST PROTOTYPING

2. Execute your routes with JBang and that's it. :-)

```
jbang -Dcamel.jbang.version=3.20.3 camel@apache/camel run * or
```

Camel JBang

- You can run Camel routes from any of the supported [DSLs](#) in Camel such as YAML, XML, Java, Groovy.
- When working with Camel JBang then dependencies are automatically resolved. This means that you do not have to use a build system like Maven or Gradle to add every Camel components as a dependency.
- Developer console: You can enable the developer console, which presents a variety of information to the developer.

How to getting started?

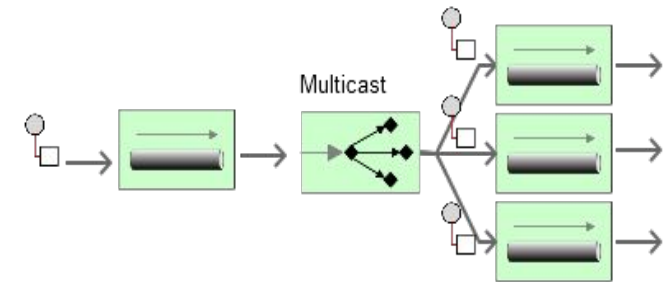
IMPLEMENTATION

3. Create a Camel Maven project

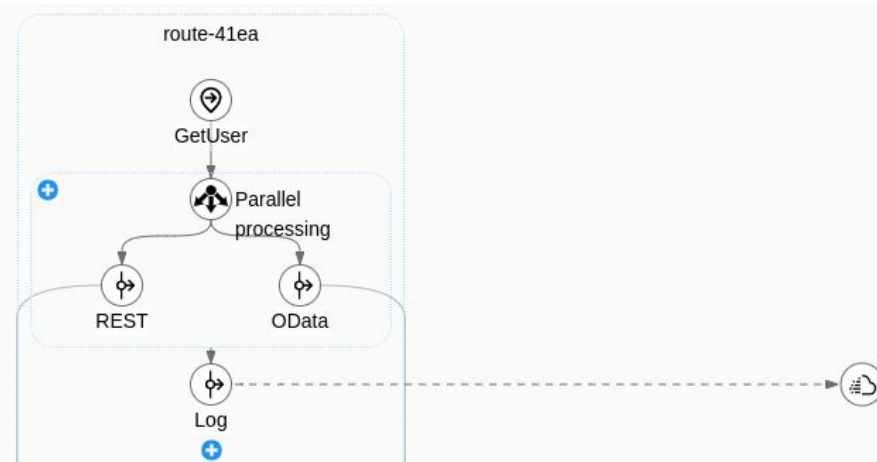
- I would recommend create your own or download an existing Camel project template :
<https://github.com/sgutierr/camel-templates/tree/master>
<https://github.com/mthirion/fuse-to-camel3-camelk/tree/main>
- You can get which maven dependencies are required in this file camel-jbang-run.properties and add them in your pom.xml.
- Copy the yam DSL and check in the application.properties file, if this attribute points to folder where are located the YAML DSL files: camel.main.routes-include-pattern = routes/*.yaml
- Maven:
 - mvn quarkus:dev

Orchestration

Multicast (Enterprise Integration Pattern)



Sending and processing the replies from the multicasts which happens concurrently. A difference with the Splitter pattern it **does not** split the message into several pieces

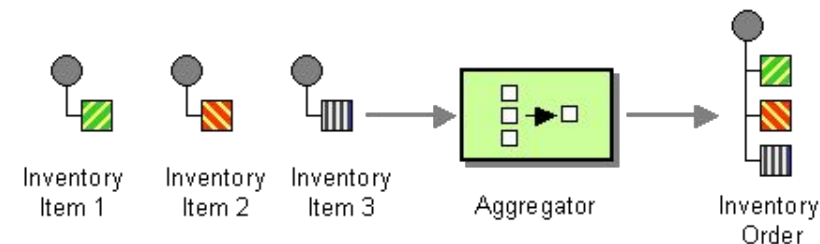


Parallelism

```
<!-- **** Service Orchestration **** -->
- route:
  id: route-41ea
  from:
    uri: direct:GetUser
    id: from-7317
    description: GetUser
  steps:
    - multicast:
      id: multicast-74d1
      aggregationStrategy: BirthAggregationStrategy
      parallelProcessing: true
      description: Parallel processing
      steps:
        - to:
          uri: direct:direct:GetUserInfoREST
          id: to-0126
          description: REST
        - to:
          uri: direct:direct:callOdataservice
          id: to-d118
          description: OData
```

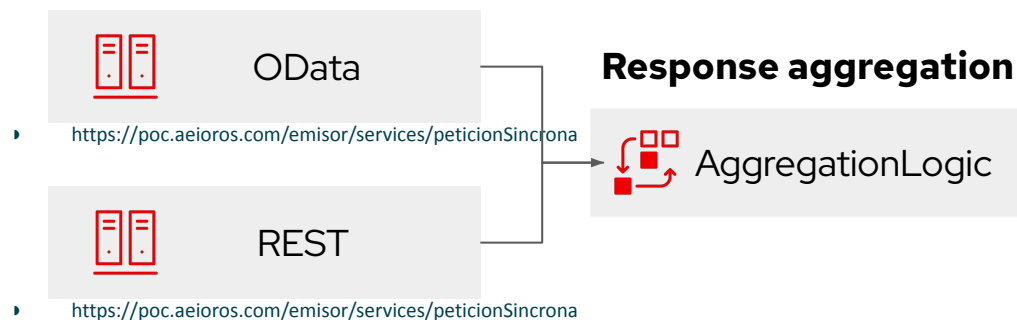
Orchestration

Aggregate (Enterprise Integration Pattern)



Response

The `AggregationStrategy` is used for aggregating the old, and the new exchanges together into a single exchange; that becomes the next old, when the next message is aggregated, and so forth.

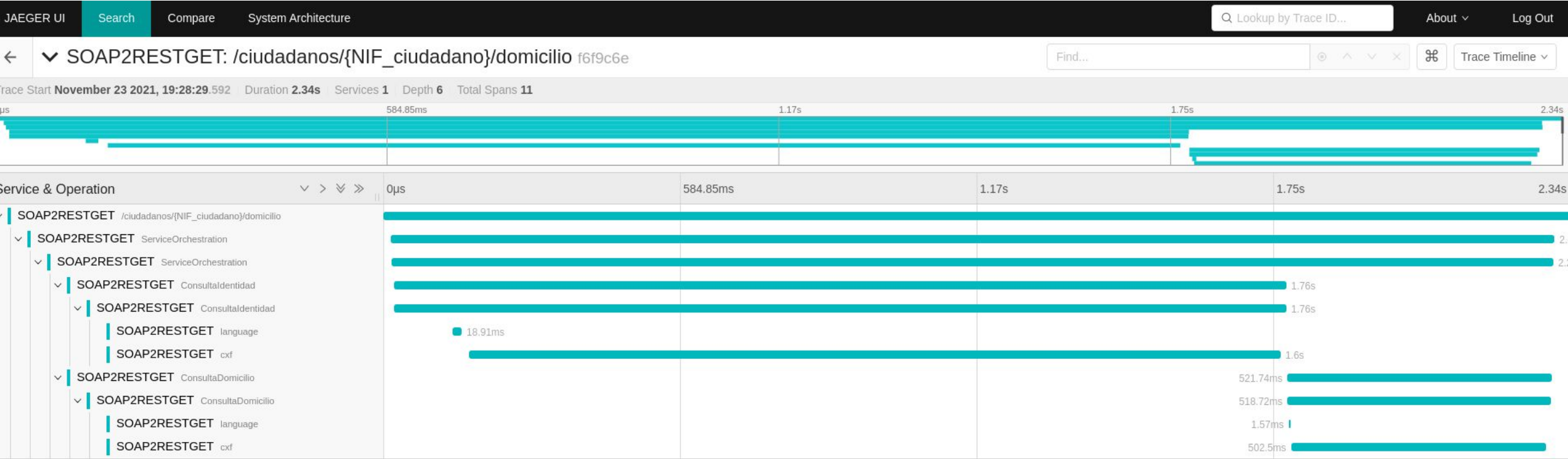


```
<!-- ***** Service Orchestration ***** -->
- multicast:
  id: multicast-74d1
  aggregationStrategy: BirthAggregationStrategy
  parallelProcessing: true
  description: Parallel processing
  steps:
```

Orchestration

Example of execution time WITHOUT Parallelism

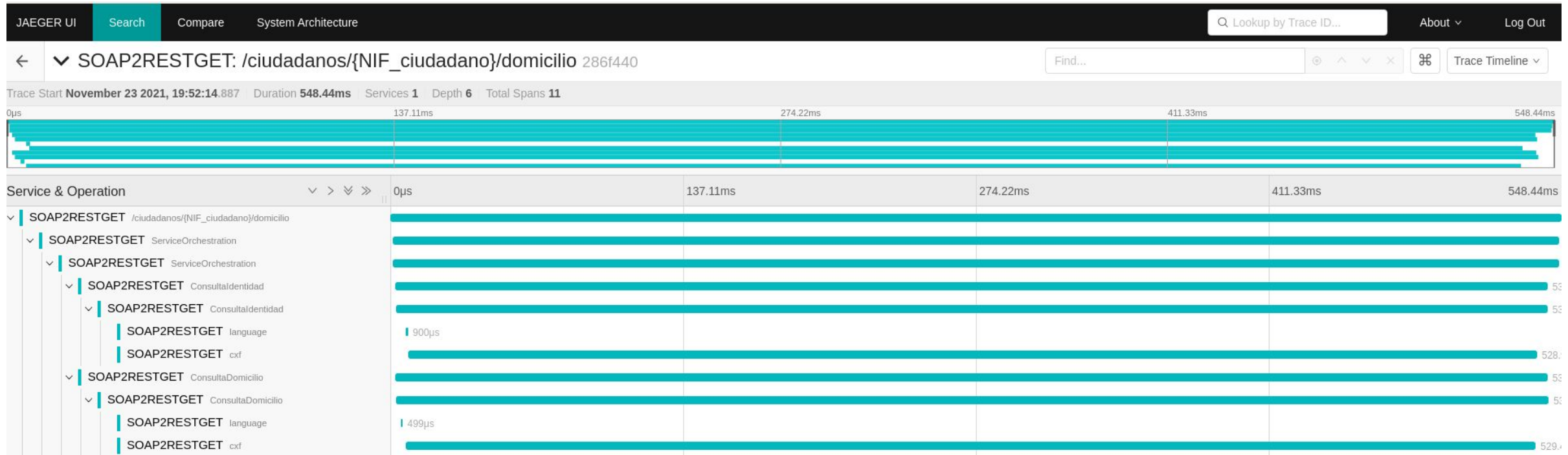
GET http://soap2rest-appdev-poc.apps.ocp4.quitala.eu/ciudadanos/PETICIONPOCR000165/domicilio



Orchestration

Example of execution time WITH Parallelism

GET <http://soap2rest-appdev-poc.apps.ocp4.quitala.eu/ciudadanos/PETICIONPOCR000165/domicilio>



Deployment

There are two Options:

- Standalone (virtual machine, bare metal, container)
- OpenShift (Kubernetes)

Maven plugin to deploy Camel on OpenShift

This plugin is added in your code - POM.XML

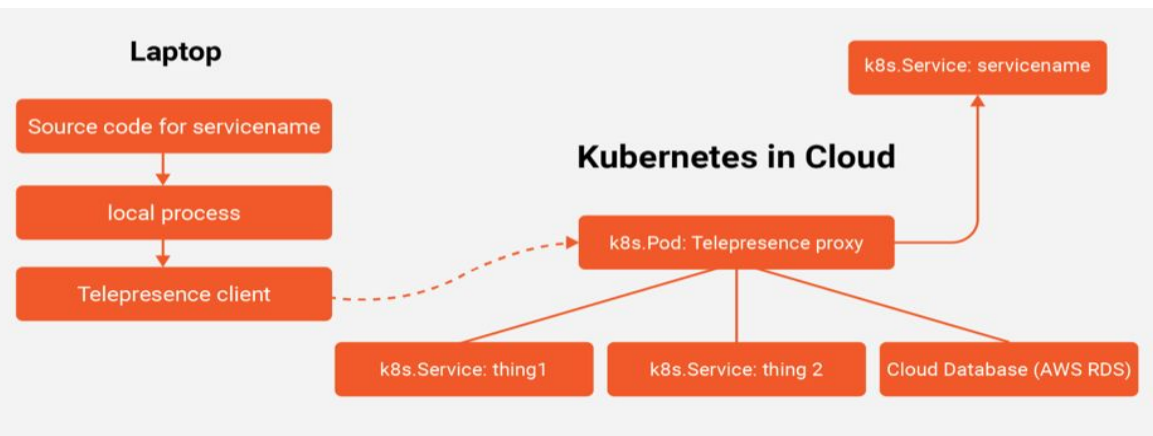


<https://www.eclipse.org/jkube/docs/openshift-maven-plugin>

- JKube is the new maven plugin based on fabric8-maven-plugin
- Built on top of **Kubernetes Maven Plugin** provides OpenShift specific features.
- Dealing with S2I images and hence inherits its flexible and powerful configuration.
-

```
mvn install -D skipTests -Popenshift
```

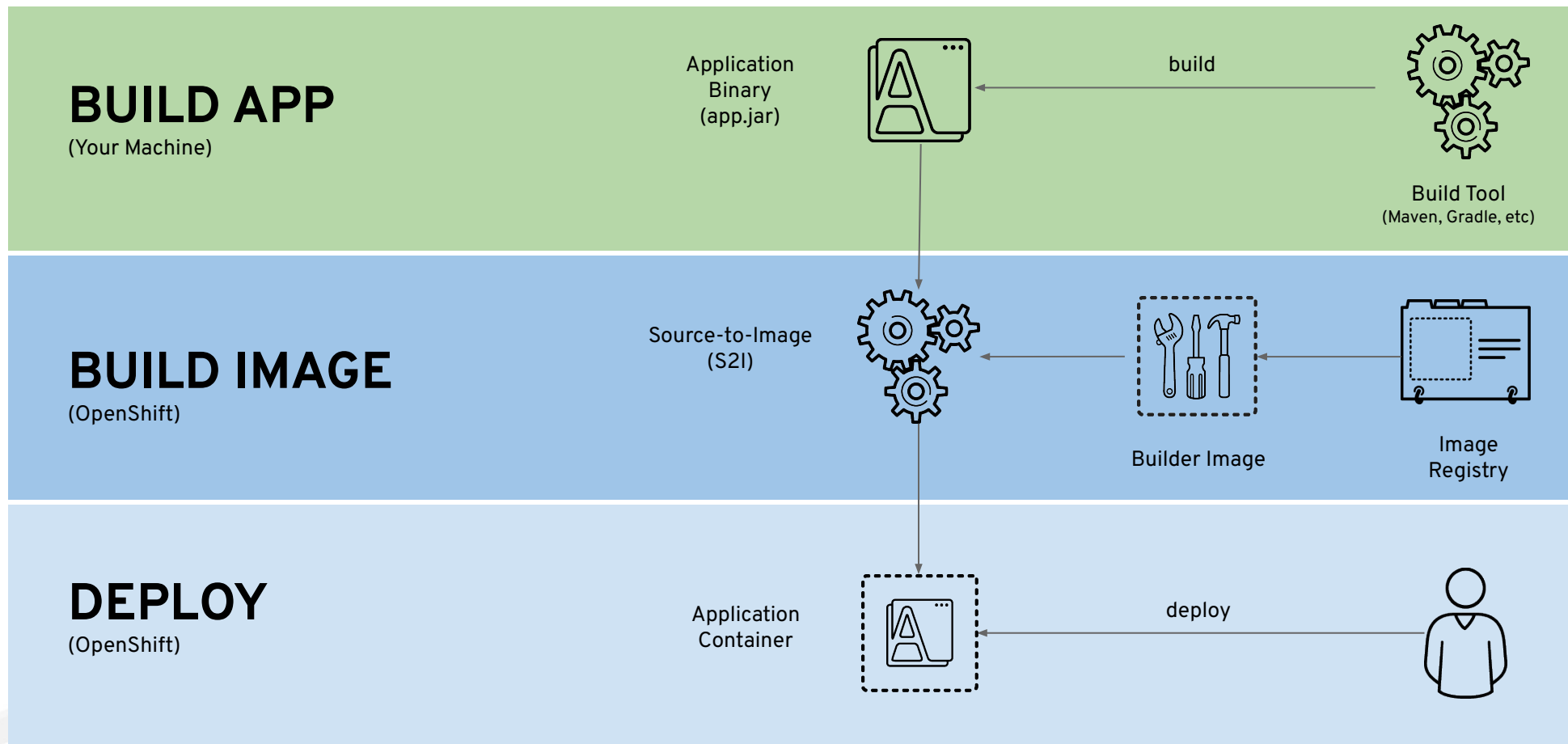
```
mvn clean package -D skipTests -Popenshift  
-Dquarkus.kubernetes.deploy=true  
-Dquarkus.kubernetes-client.trust-certs=true  
-Dquarkus.openshift.route.expose=true
```



Deployment on OpenShift – How does it works?

Binary Deployment

Artefactos pre-construidos, OpenShift empaqueta y despliega



Additional slides

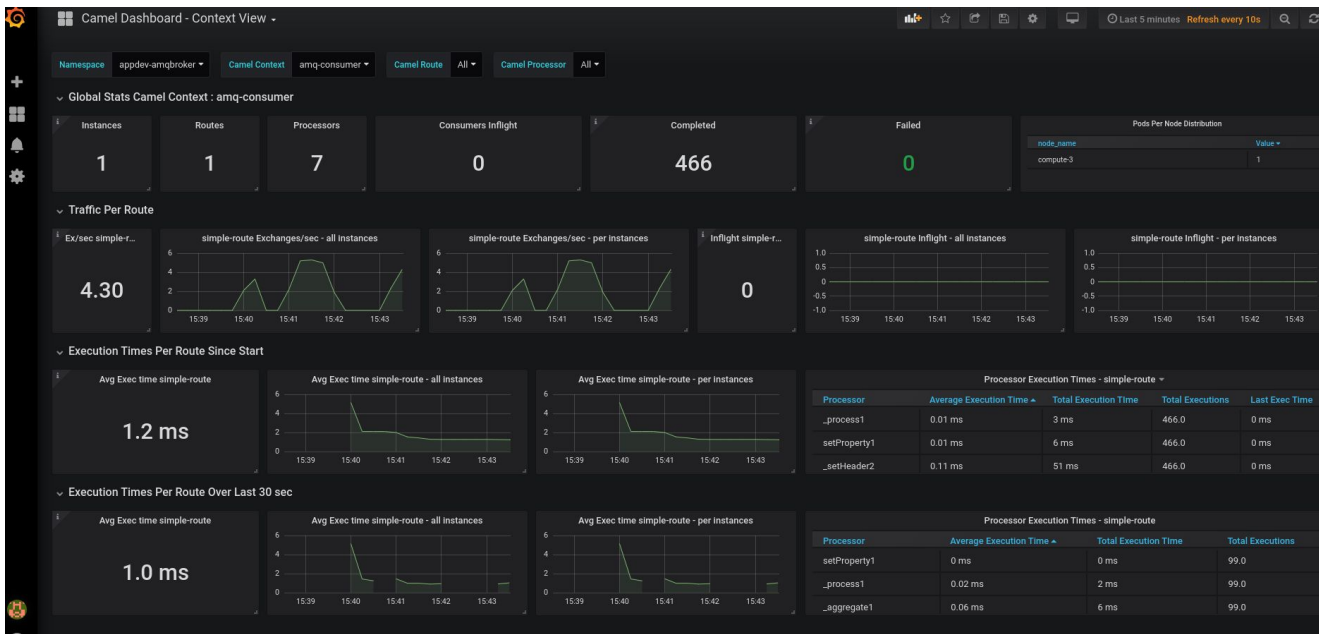
Jaeger



Prometheus/Grafana

Monitoring service resources on OpenShift

https://access.redhat.com/documentation/en-us/red_hat_fuse/7.9/html/fuse_on_openshift_guide/get-started-admin#prometheus-openshift



Grafana

3scale API Developer Portal

Discover, subscribe and access to the API Catalog

The screenshot shows the 3scale API Developer Portal interface. At the top, there is a navigation bar with a logo on the left, links for "DOCUMENTATION" and "PLANS" in the center, and a "SIGN IN" link on the right. Below the navigation bar is a large hero section with a background image of a person's hands using a laptop and a tablet. The text "API portal" is centered above the image. A blue semi-transparent box on the left side of the hero section contains the text "Fulfilling your clients' aspirations in a digital world". Below the hero section is a footer area with three columns of content: "Register" with a right arrow icon and the text "Register to the developer portal to use our APIs"; "Get your API key" with a key icon and the text "Use your API key to authenticate and report the calls you make"; and "Create your app" with a code icon and the text "Start coding and create awesome applications with the Echo API".

API portal

Fulfilling your clients' aspirations in a digital world

Register
➔ Register to the developer portal to use our APIs

Get your API key
🔑 Use your API key to authenticate and report the calls you make

Create your app
</> Start coding and create awesome applications with the Echo API

APIcurio Service Registry

Discover, subscribe and access to the API Catalog

- Datastore for standard event schemas and API designs
- Handles following data formats:
 - Apache Avro
 - JSON Schema
 - Protobuf (protocol buffers)
 - OpenAPI
 - AsyncAPI
 - GraphQL
 - Kafka Connect Schema
- Drop in replacement for Confluent Schema Registry



QUARKUS



APIcurio Service Registry

Use Cases



Schema Registry

Schema registry for Kafka serializers/deserializers.



API Designs

API specification registry for API consumers.



Shared Data Types

Shared data types (schemas) across API and Event driven architectures.

OpenShift Logging - Log aggregation

Leveraging OpenShift services



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