Camel Kafka Connector

Tune Kafka to speak with (almost) everything

Speakers:
Andrea Tarocchi (@valdar)
Hugo Guerrero (@hguerreroo)

APACHECON @HOME
Spt, 29th – Oct. 1st

2020

Who are we?

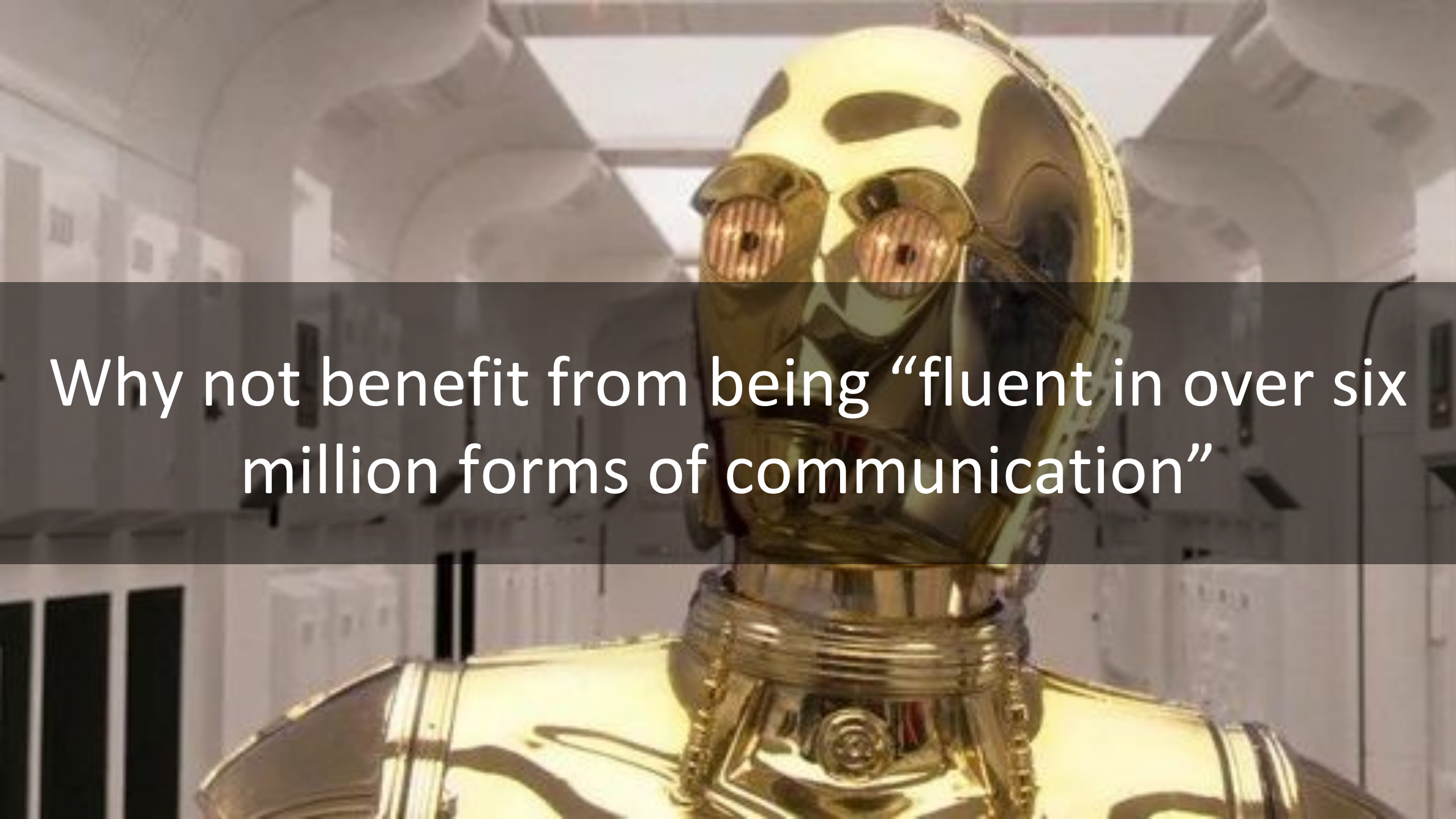
Hugo Guerrero (@hguerreroo)

- Developer Advocate @ Red Hat
- APIs & Messaging Specialist
- Food, Travel & History Enthusiast

Andrea Tarocchi (@valdar)

- Senior Software Engineer @ Red Hat
- Apache Camel committer
- https://blog.valdar.it


When you need to talk to “almost” everything


Why not benefit from being “fluent in over six million forms of communication”

Camel

Quick recap

APACHECON NA
Spt, 28th – Oct. 2nd

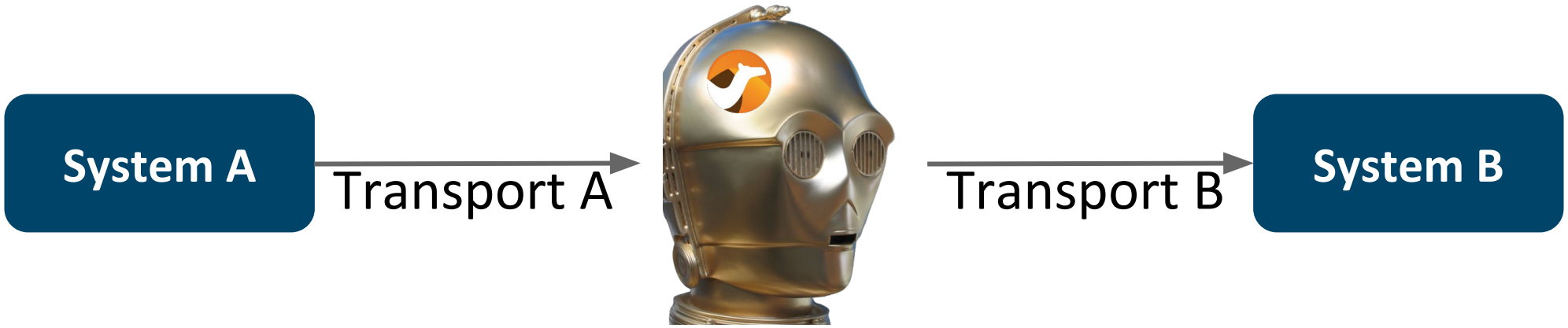
2020

What is Apache Camel?

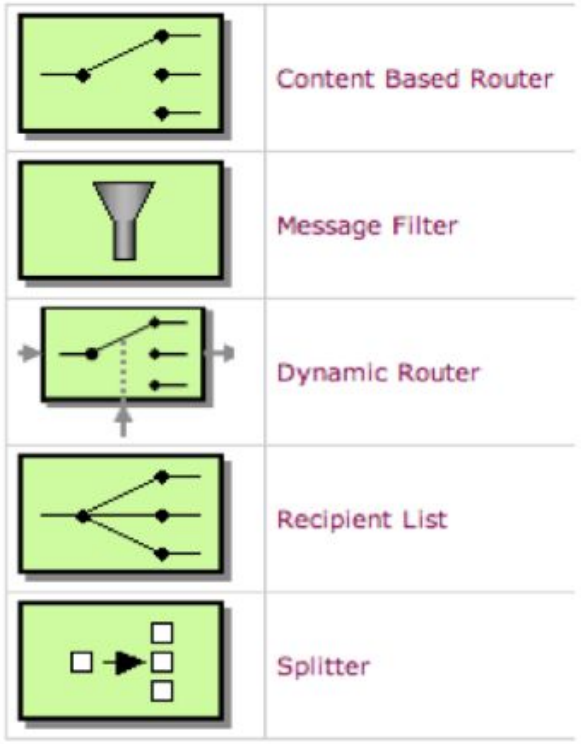
- **Open source** swiss knife **framework** for integration
- **350+ components, data formats and protocols** allow to talk to (almost) everything.
- Routes and **Enterprise Integration Patterns** (EIP) modeled for designing and developing integration solutions
- Very active project and community.

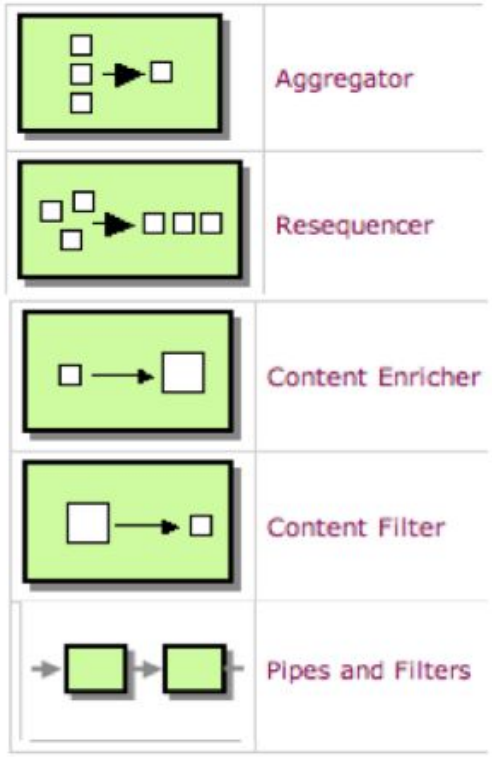
Source: https://camel.apache.org/

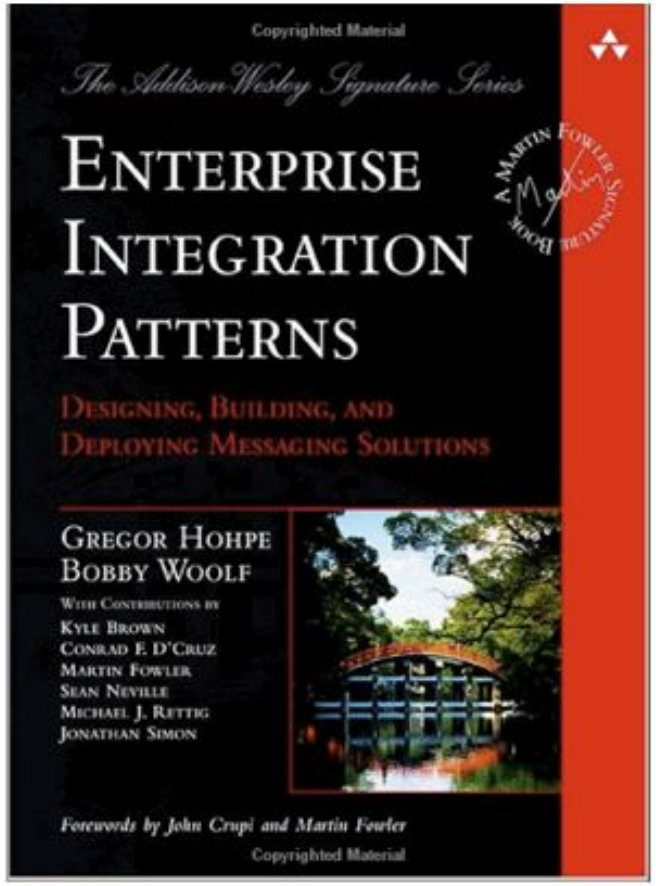
System Integration



Enterprise Integration Patterns



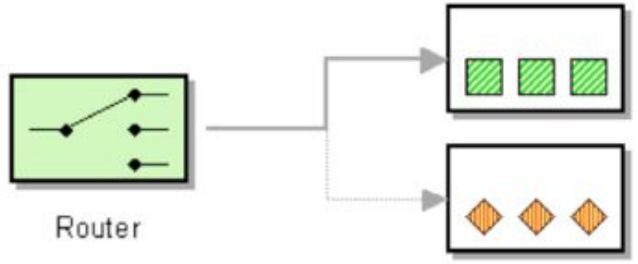




Camel Routes

```
from("file:data/inbox")
  .to("jms:queue:order");
```

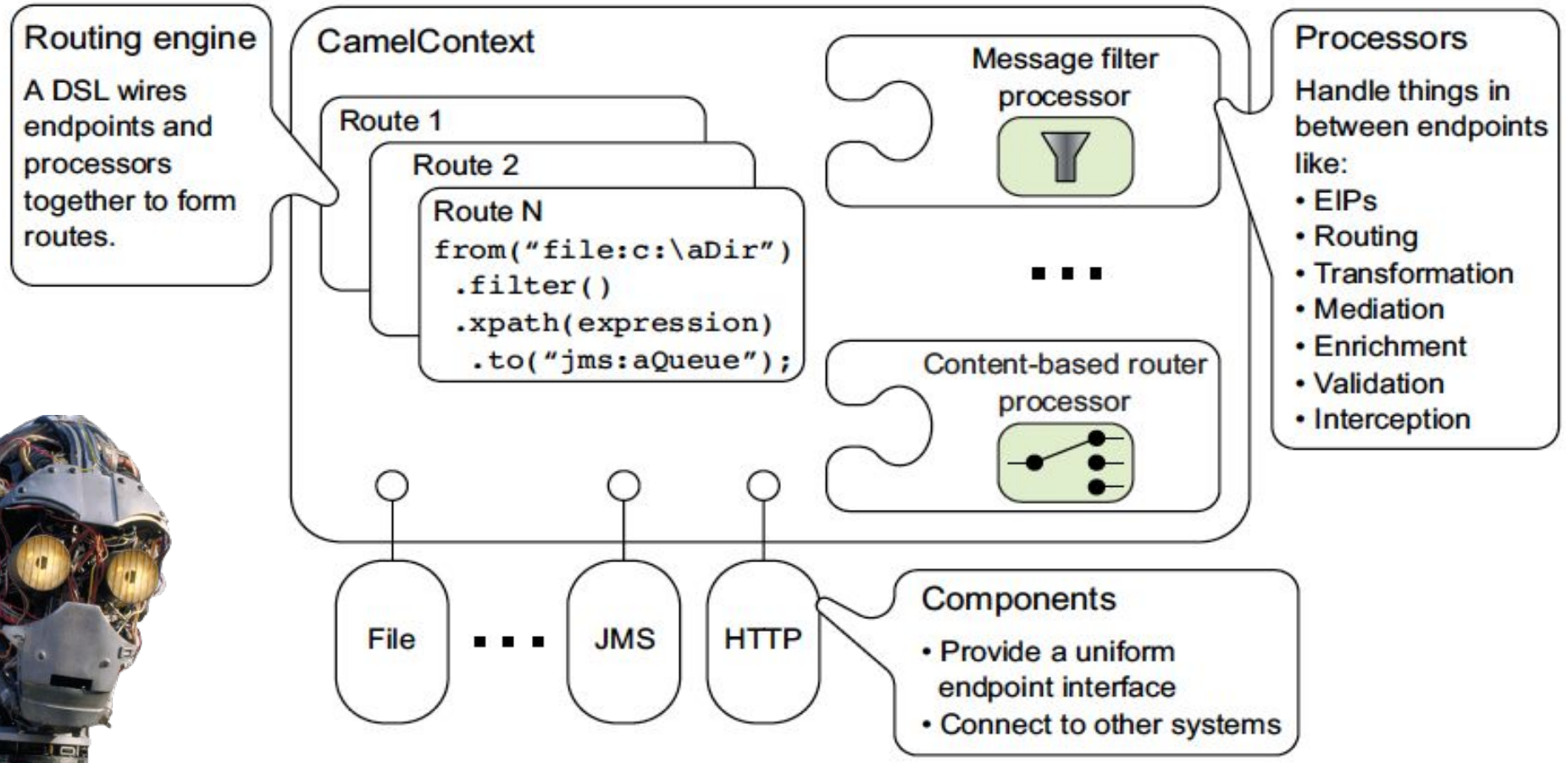
Java DSL

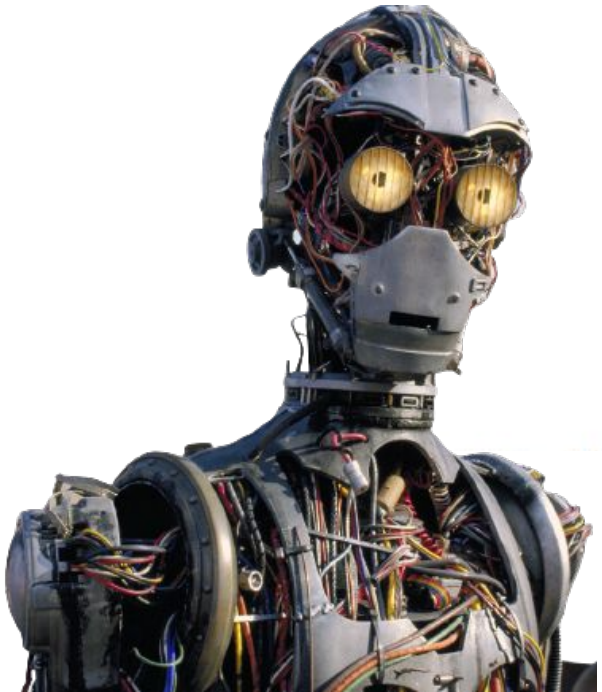


XML DSL

```
<route>
  <from uri="file:data/inbox"/>
  <to uri="jms:queue:order"/>
</route>
```

Camel Architecture





Apache Camel Projects

Camel K

Camel on
Kubernetes & Knative

Camel

Swiss knife of
integration

Camel Quarkus

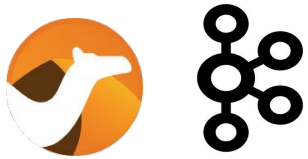
Optimized JVM & Native
compiled Java (GraalVM)

Camel Karaf

Camel on
Apache Karaf / OSGi

Camel Spring Boot

Camel on
Spring Boot



Camel Kafka Connector

Kafka Connector
with Camel

Source:https://camel.apache.org/projects/

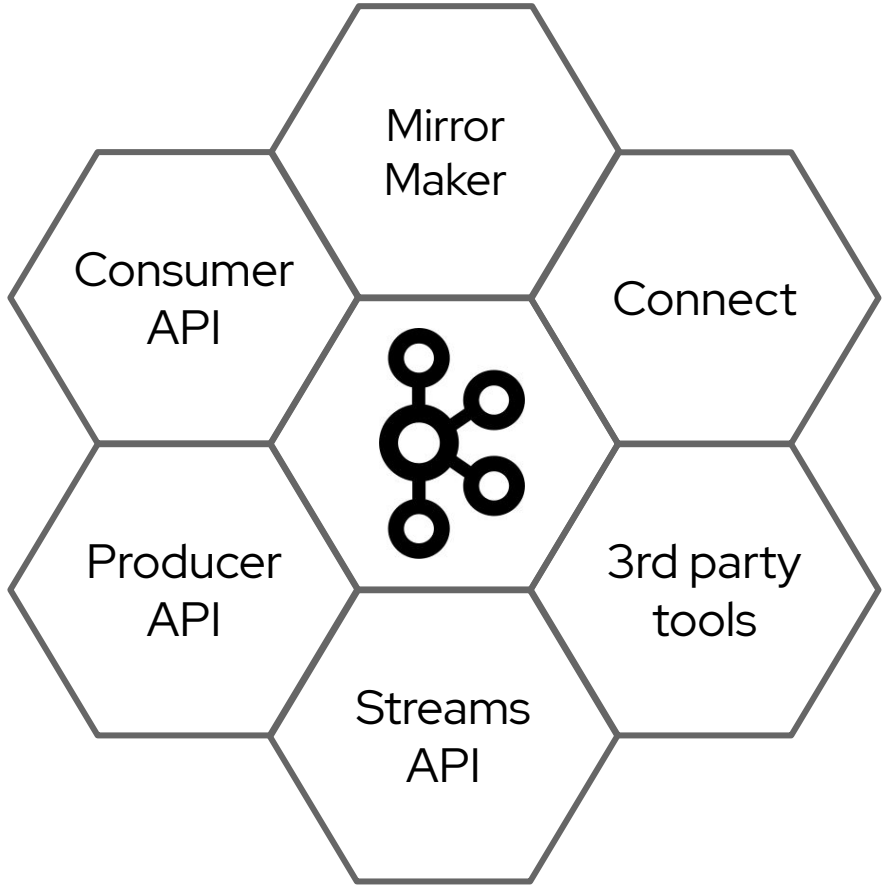
Kafka

Quick intro

APACHECON NA
Spt, 28th – Oct. 2nd 2020

What is Apache Kafka?

- Project originally created by LinkedIn
  - publish/subscribe messaging system
  - data-streaming platform
  - distributed commit log
- **Broader ecosystem** besides broker



Source: http://kafka.apache.org

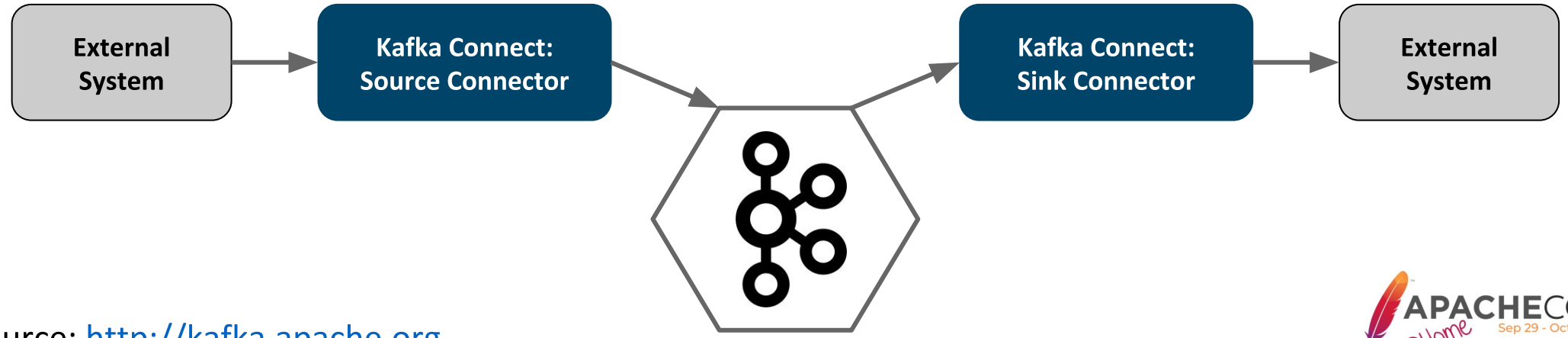
Kafka Connect

- Wraps around the Consumer and Producer APIs
- Framework for **transferring data between Kafka and other data systems**
- Facilitate **data conversion**, scaling, load balancing, fault tolerance
  - **Connector** plugins are deployed into Kafka connect
  - Well defined API for creating new connectors (with Sink/Source)
  - Apache Kafka itself includes only FileSink and FileSource plugins
  - Some **additional plugins** are available outside of Apache Kafka project

Source: http://kafka.apache.org

Why Kafka Connect?

- **Part of Apache Kafka** itself
- Distributed and scalable by default
- Automatic offset management
- Simple transformations
- Streaming / batch integration
- Easier and less error-prone than writing your own integrations

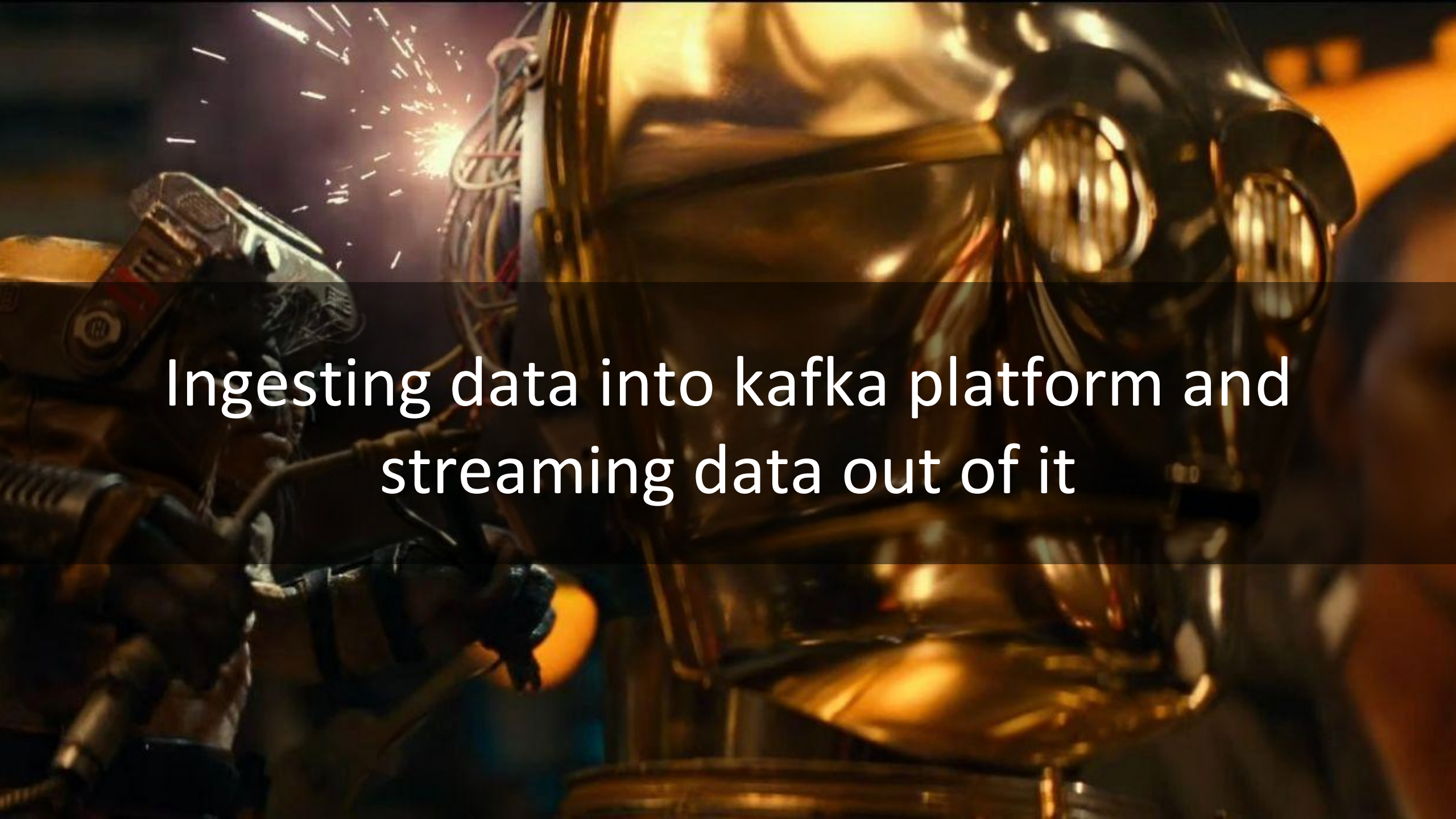


Source: http://kafka.apache.org

Camel Kafka Connector

APACHECON NA
Spt, 28th – Oct. 2nd

2020

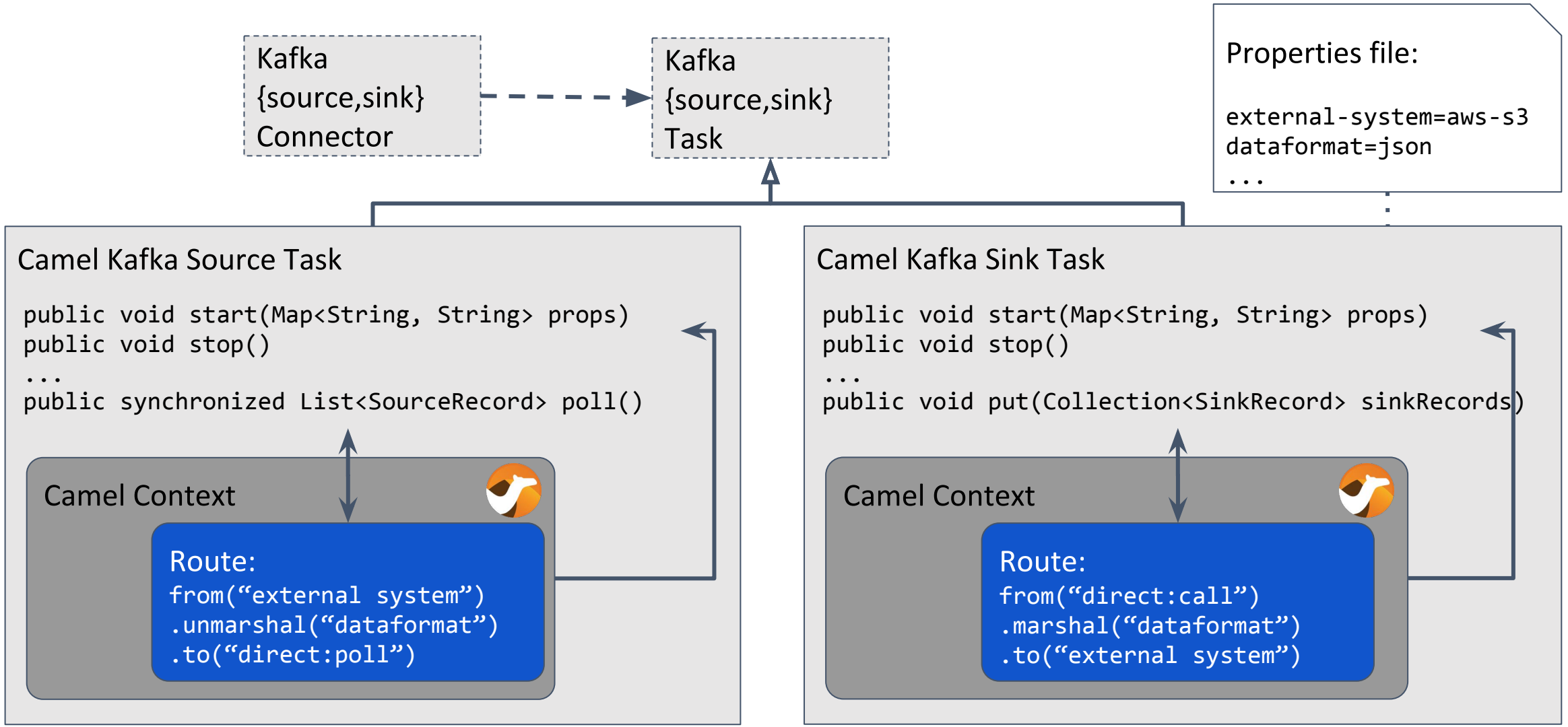

Ingesting data into kafka platform and
streaming data out of it

What is Camel Kafka Connector

- A pool of **Kafka Connectors** built on top of Apache Camel
- Reuses in a **simple way** most of the Camel components as Kafka sink and sources
- Creates a (**tiny**) layer between Camel and Kafka Connect
- Auto Generated documentation and connectors list
- Live as a sub-project of Apache Camel.

Source: https://camel.apache.org/camel-kafka-connector/latest/connectors.html

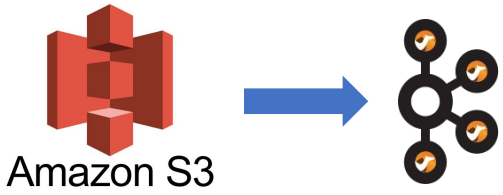
What is Camel Kafka Connector



Why use Camel Kafka Connector?

Ingesting data into kafka platform and streaming data out of it

- Consolidate events stored in kafka into a Mongodb instance for reporting purposes (Mongodb Sink)
- Consolidate events stored in kafka into an Elasticsearch instance for analytics purposes (Elasticsearch Sink)
- Ingest transactional log events to further process and aggregate them (files source of syslog source)




```
name=CamelAWSS3SourceConnector
connector.class=org.apache.camel.kafkaconnector.awss3.CamelAwss3SourceConnector
key.converter=org.apache.kafka.connect.storage.StringConverter
value.converter=org.apache.camel.kafkaconnector.awss3.converters.S3ObjectConverter

camel.source.maxPollDuration=10000

topics=test1

camel.source.url=aws-s3://camel-kafka-connector?autocloseBody=false

camel.component.aws-s3.access-key=xxxx
camel.component.aws-s3.secret-key=yyyy
camel.component.aws-s3.region=EU_WEST_1
```

No code, configuration only!


```
name=CamelAWSSQSSinkConnector
connector.class=org.apache.camel.kafkaconnector.awssqs.CamelAwssqsSinkConnector
key.converter=org.apache.kafka.connect.storage.StringConverter
value.converter=org.apache.kafka.connect.storage.StringConverter

topics=test1

camel.sink.path.queueNameOrArn=camel-1

camel.component.aws-sqs.access-key=xxxx
camel.component.aws-sqs.secret-key=yyyy
camel.component.aws-sqs.region=EU_WEST_1
```

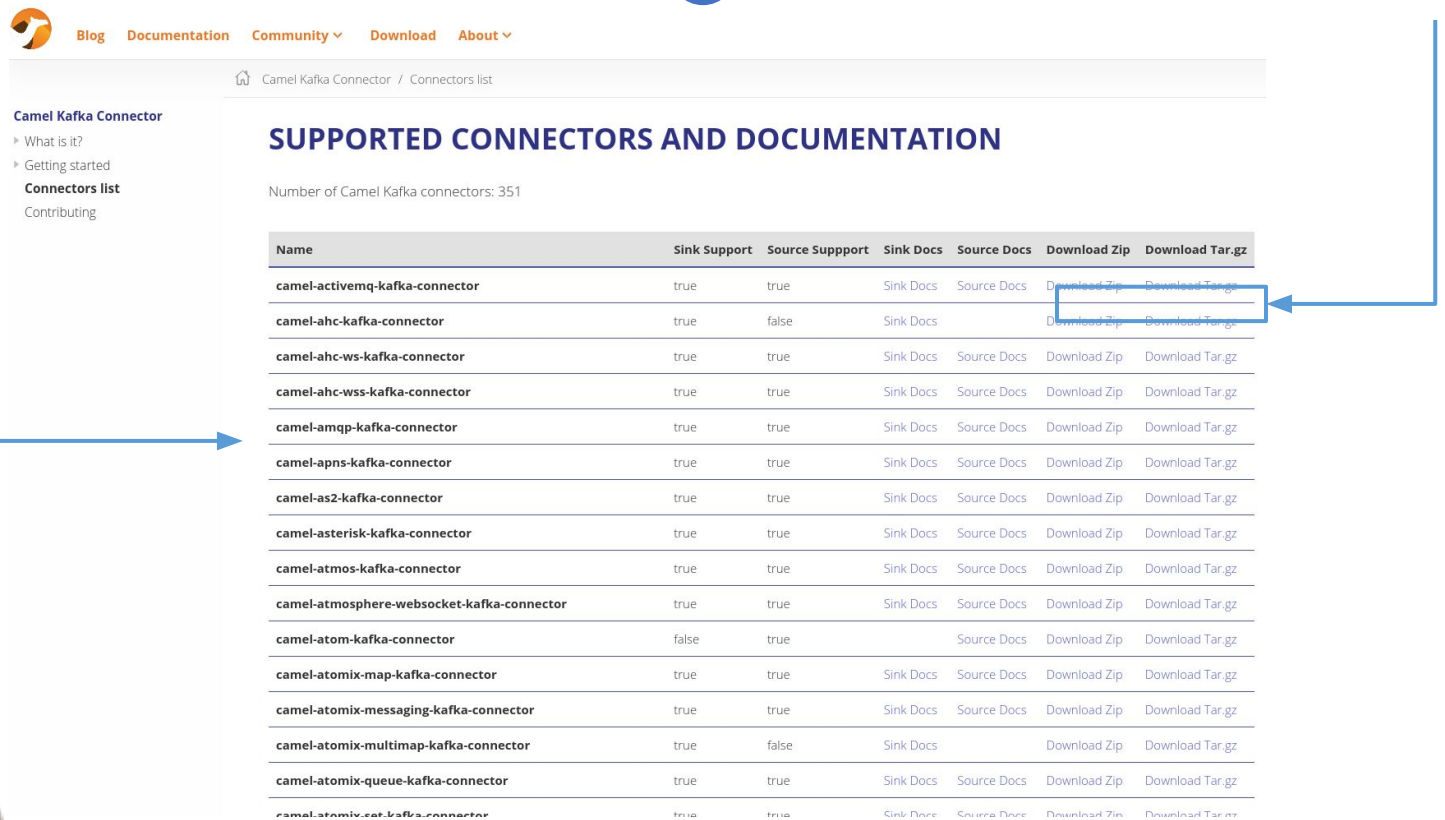
Source: https://github.com/apache/camel-kafka-connector-examples

Getting Started

Downloading the connectors

1 The connectors list, browse you should

2 Download the package compressed file



Source: https://camel.apache.org/camel-kafka-connector/latest/connectors.html

Photo: "Yoda" (CC BY-SA 2.0) by Hannaford

Getting Started (bare metal)

1. Unzip the file:

unzip camel-aws-s3-kafka-connector-0.2.0-package.zip

```
camel-aws-s3-kafka-connector
├── LICENSE.txt
├── NOTICE.txt
├── README.adoc
├── aws-java-sdk-core-1.11.714.jar
├── aws-java-sdk-kms-1.11.714.jar
├── aws-java-sdk-s3-1.11.714.jar
├── camel-api-3.2.0.jar
├── camel-aws-s3-3.2.0.jar
├── camel-aws-s3-kafka-connector-0.2.0.jar
...
├── commons-codec-1.14.jar
├── commons-logging-1.2.jar
├── httpclient-4.5.12.jar
├── httpcore-4.4.13.jar
├── ion-java-1.0.2.jar
├── jackson-annotations-2.10.3.jar
├── jackson-core-2.10.3.jar
├── jackson-databind-2.10.3.jar
├── jackson-dataformat-cbor-2.10.3.jar
├── jmespath-java-1.11.714.jar
├── joda-time-2.8.1.jar
└── slf4j-api-1.7.30.jar
```

2. Configure the connector:

```
name=CamelAWSS3SourceConnector
connector.class=org.apache.camel.kafkaconnector.awss3.CamelAwss3SourceConnector
key.converter=org.apache.kafka.connect.storage.StringConverter
value.converter=org.apache.camel.kafkaconnector.awss3.converters.S3ObjectConverter

camel.source.maxPollDuration=10000

topics=test1

camel.source.url=aws-s3://camel-kafka-connector?autocloseBody=false

camel.component.aws-s3.access-key=xxxx
camel.component.aws-s3.secret-key=yyyy
camel.component.aws-s3.region=EU_WEST_1
```

3. Run the AWS S3 connector:

$KAFKA_HOME/bin/connect-standalone.sh $KAFKA_HOME/config/connect-standalone.properties examples/CamelAWSS3SourceConnector.properties

Source: https://camel.apache.org/camel-kafka-connector/latest/try-it-out-locally.html

Getting Started (Container Image)

1 Create a container image from the Kafka Connect base image

```
FROM registry.redhat.io/amq7/amq-streams-kafka-24-rhel7:1.4.0
USER root:root
COPY ./my-plugins/ /opt/kafka/plugins/
USER 1001
```

2 Point to the new container image

```
apiVersion: kafka.strimzi.io/v1beta1
kind: KafkaConnect
metadata:
  name: my-connect-cluster
  annotations:
    strimzi.io/use-connector-resources: "true"
spec:
  #...
  image: my-new-container-image
```

3 Create Connector Instance

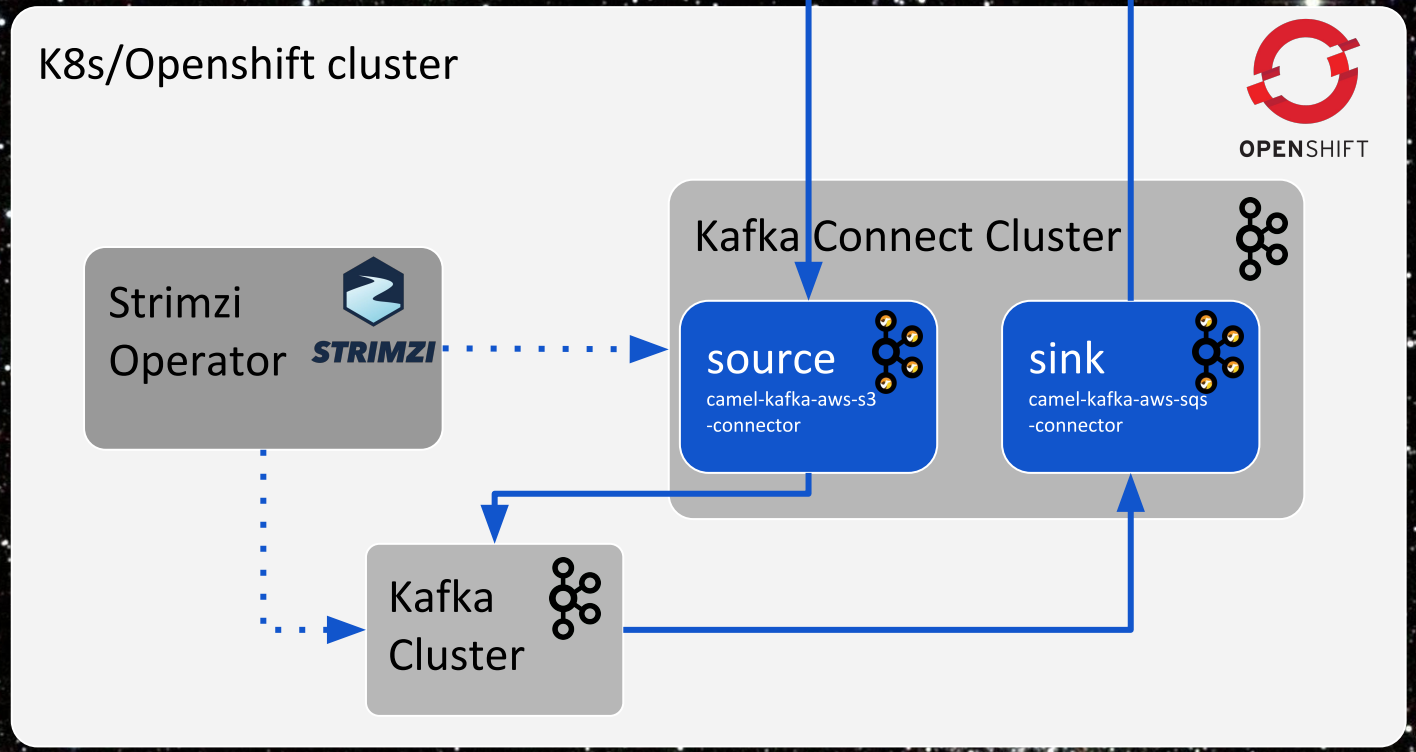
```
apiVersion: kafka.strimzi.io/v1alpha1
kind: KafkaConnector
metadata:
  name: s3-source-connector
  namespace: atarocch-ckc
  labels:
    strimzi.io/cluster: my-connect-cluster
spec:
  class: org.apache.camel.kafkaconnector.awss3.CamelAwss3SourceConnector
  tasksMax: 1
  config:
    key.converter: org.apache.kafka.connect.storage.StringConverter
    value.converter: org.apache.camel.kafkaconnector.awss3.converters.S3ObjectConverter
    topics: s3-topic
    camel.source.path.bucketNameOrArn: camel-connector-test
    camel.source.endpoint.autocloseBody: false
    camel.source.maxPollDuration: 10000
    camel.component.aws-s3.configuration.access-key: xxx
    camel.component.aws-s3.configuration.secret-key: xxx
    camel.component.aws-s3.configuration.region: xxx
```

Source: https://camel.apache.org/camel-kafka-connector/latest/try-it-out-locally.html

Demo “architecture”

AWS S3

AWS SQS



DEMO

TIME!

APACHECON NA
Spt, 28th – Oct. 2nd

2020

Apache Camel Kafka what's next?

Focus on needed improvements:

- Support for handling offset (save and resume) in sources connectors.
- Better error handling integrated with camel.
- Increase the number of integration tested covered connectors.

Apache Camel Kafka Connector

Takeaways

- Combines the features of two great Apache projects
  - Experience and maturity of the Apache Camel project with enterprise integration
  - Simplicity and distributed nature of Kafka Connect
- Existing Kafka Connect users get a lot of new options and integrations
- Existing Camel users get jump-start into the Kafka world

Apache Camel Kafka Connector

Some useful links

- https://github.com/apache/camel-kafka-connector
- https://camel.apache.org/camel-kafka-connector/latest
- https://camel.zulipchat.com
- https://twitter.com/apachecamel @ApacheCamel

Thank you!

Questions?

APACHECON NA
Spt, 28th – Oct. 2nd 2020