



KubeCon



CloudNativeCon



OPEN SOURCE SUMMIT

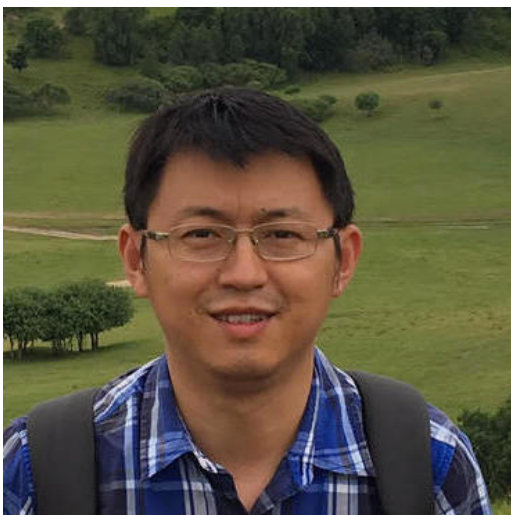
China 2019

Building Cloud Integration Application Through Knative and Apache Camel-K

Willem Jiang & Longchun Zhang



Speaker Introduction



Jiang Ning is the technical expert of Huawei, a member of the Apache Software Foundation, he worked on many Apache projects like Camel, CXF, ServiceMix and ServiceComb.



Longchun Zhang is the Technical Expert of PaaS CTO office of Huawei, is also senior architect in middle-ware domain, in charging of research, design, and implementation of gPaaS/iPaaS and middle-ware related services now.

Agenda



KubeCon



CloudNativeCon



OPEN SOURCE SUMMIT

China 2019

- The common user case of Smart park
- Why do we use Apache Camel
- How to use Camel-K and Knative to run Camel application
- Demo

Smart Park User Case



KubeCon

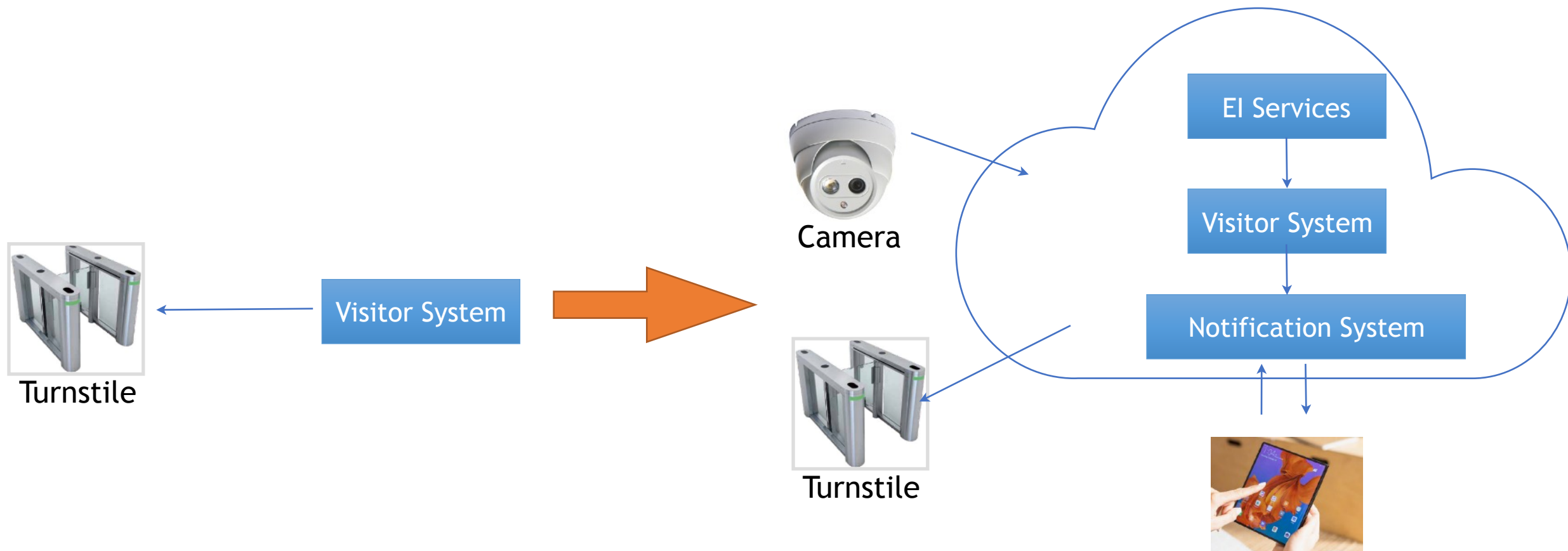


CloudNativeCon



OPEN SOURCE SUMMIT

China 2019



Smart Park User Case



KubeCon



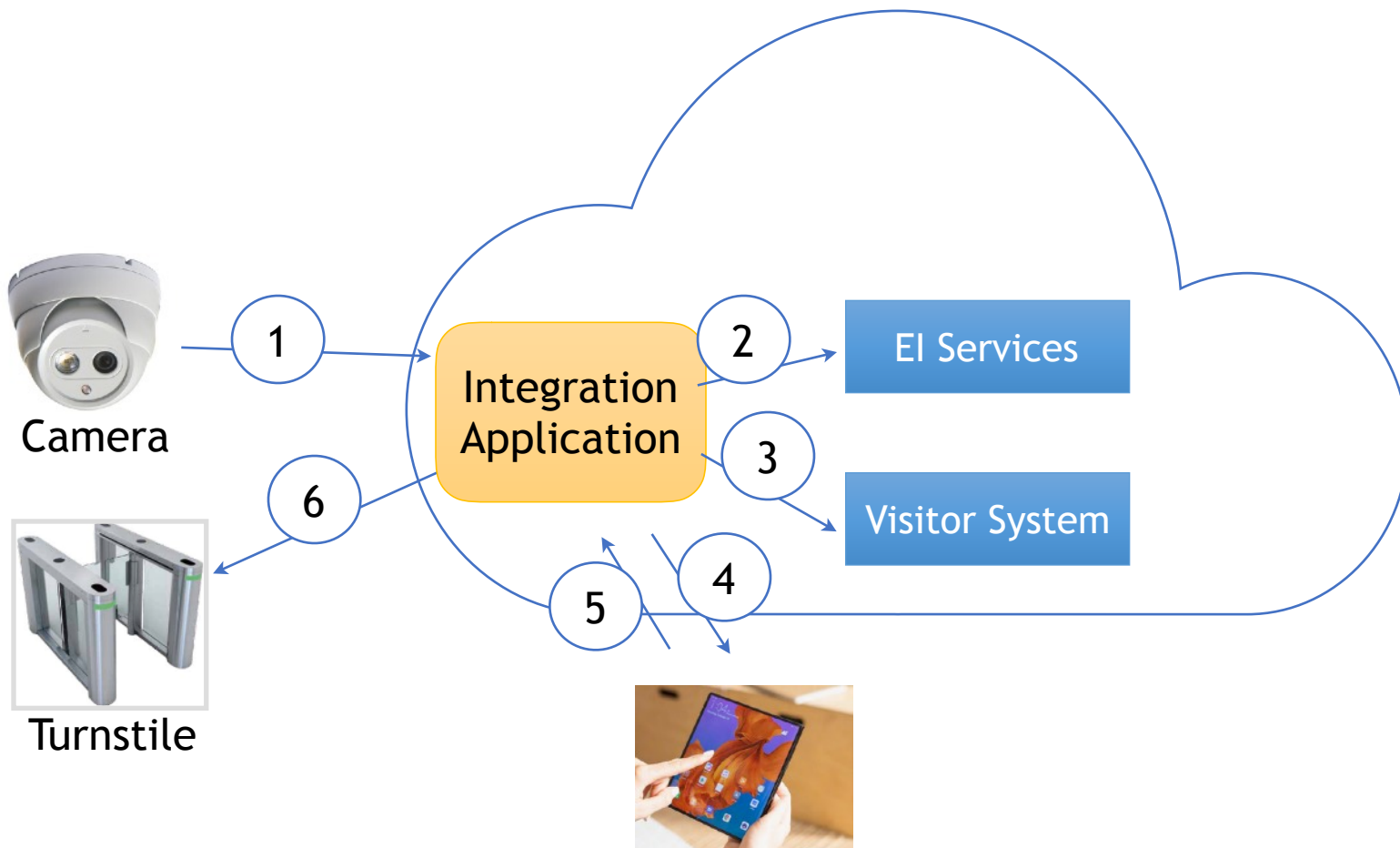
CloudNativeCon



OPEN SOURCE SUMMIT

China 2019

1. A new face event arrived face photo()
2. Recognize the face id by EI Services
3. Access control by Visitor System
4. Send message to the on duty mobile
5. Get order from the on duty mobile
6. Send the "Open" command to turnstile



Smart Park User Case



KubeCon

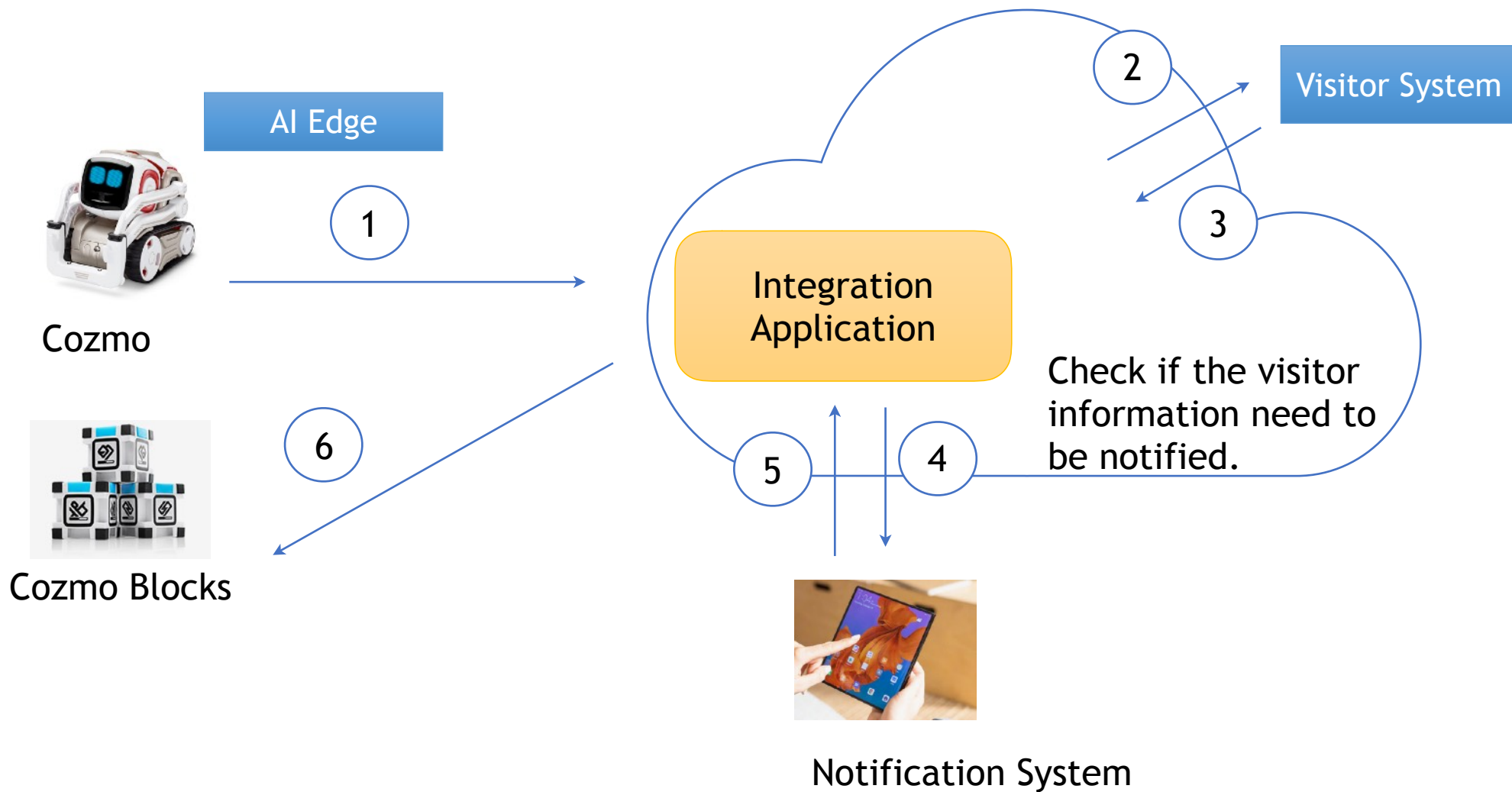


CloudNativeCon



OPEN SOURCE SUMMIT

China 2019



Why Apache Camel



KubeCon



CloudNativeCon



OPEN SOURCE SUMMIT

China 2019

- The Swiss knife of Integration
- Java-based integration framework
- Easy to run and integrate
- EIP implementation
- Powerful DSL
- 300+ Components
- Amazing Community, 12 + years development , still very active



APACHE[®]
Camel

Integration work



KubeCon

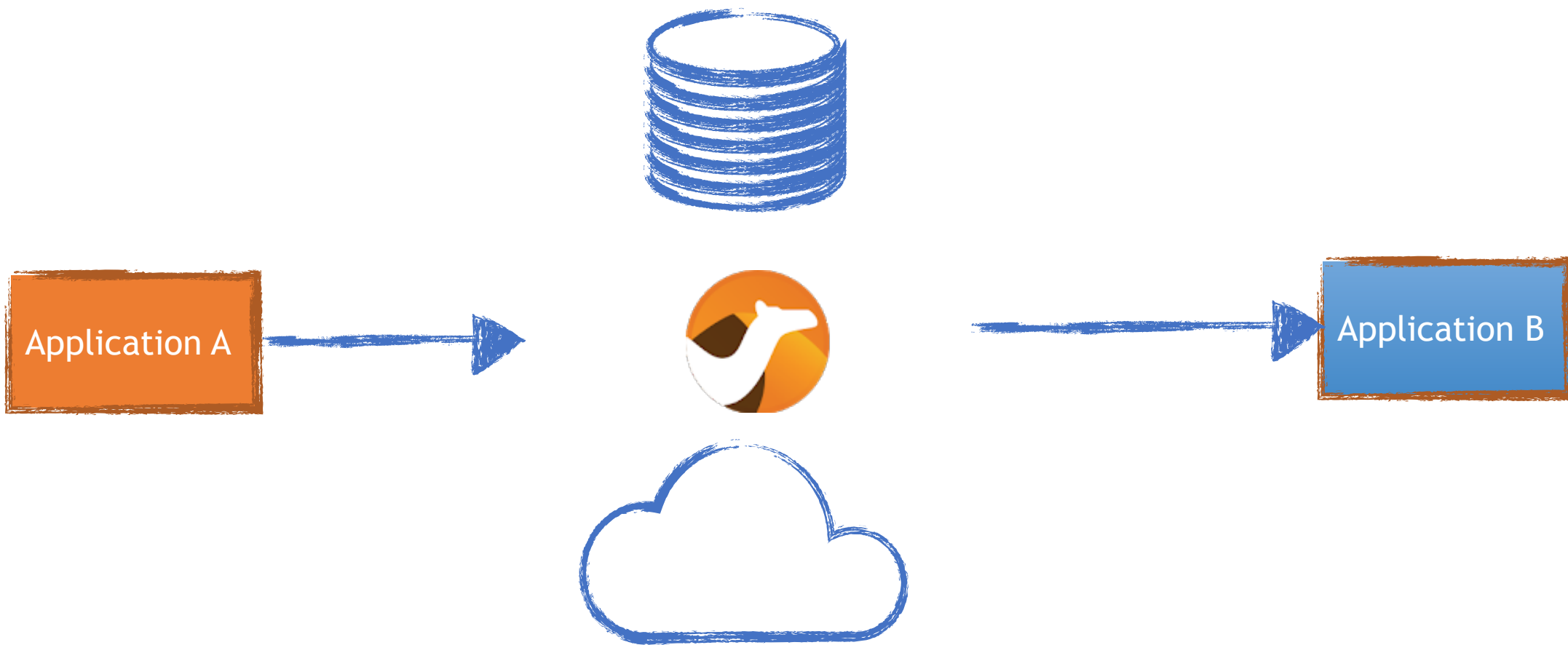


CloudNativeCon



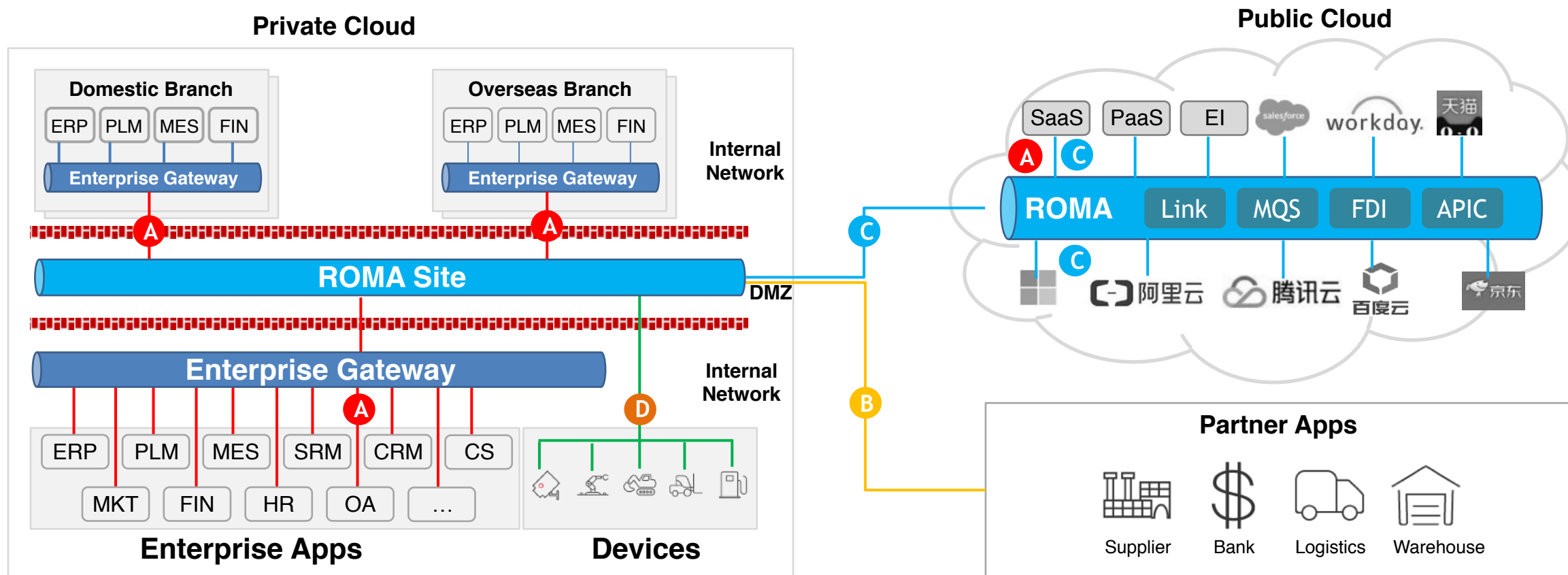
OPEN SOURCE SUMMIT

China 2019



```
from("ApplicationA").doSomething().to("ApplicationB")
```


ROMA Integration Platform



Enterprise Integration Pattern



KubeCon

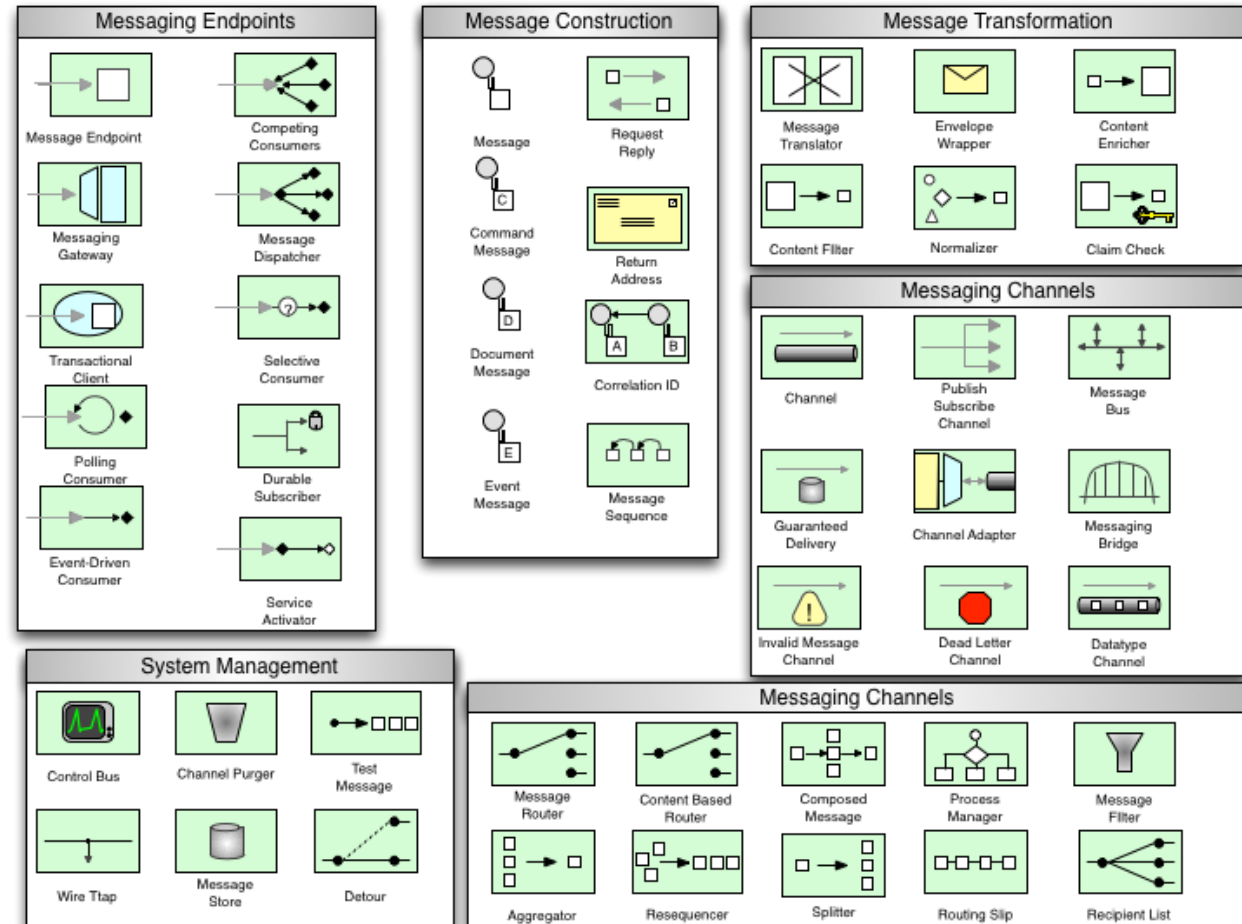
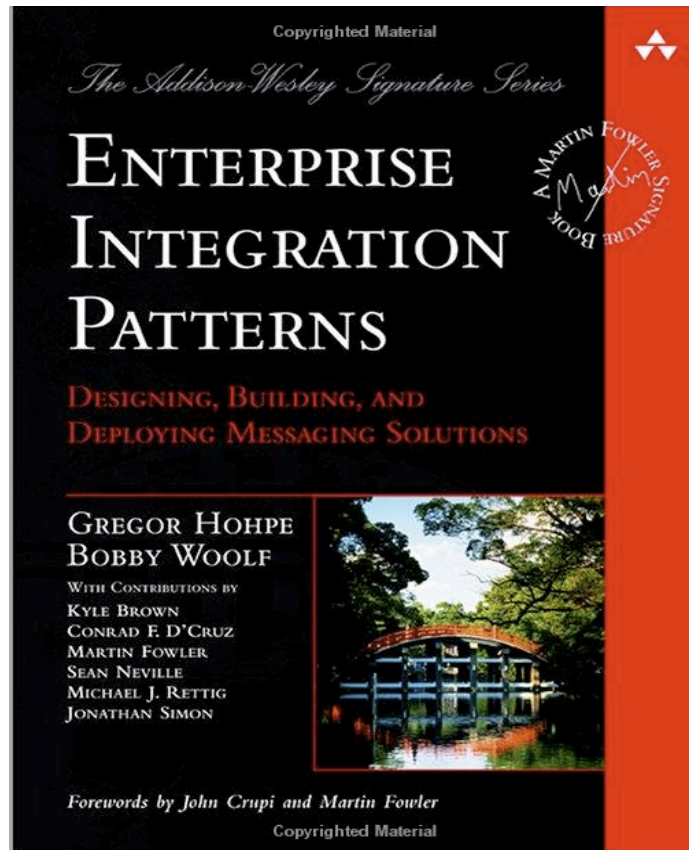


CloudNativeCon



OPEN SOURCE SUMMIT

China 2019



<https://www.enterpriseintegrationpatterns.com/patterns/messaging/>

Why Camel DSL



KubeCon

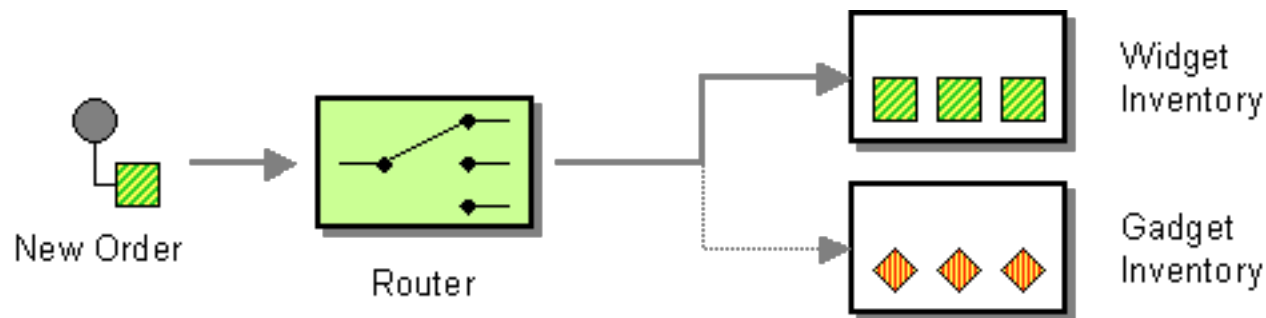


CloudNativeCon



OPEN SOURCE SUMMIT

China 2019

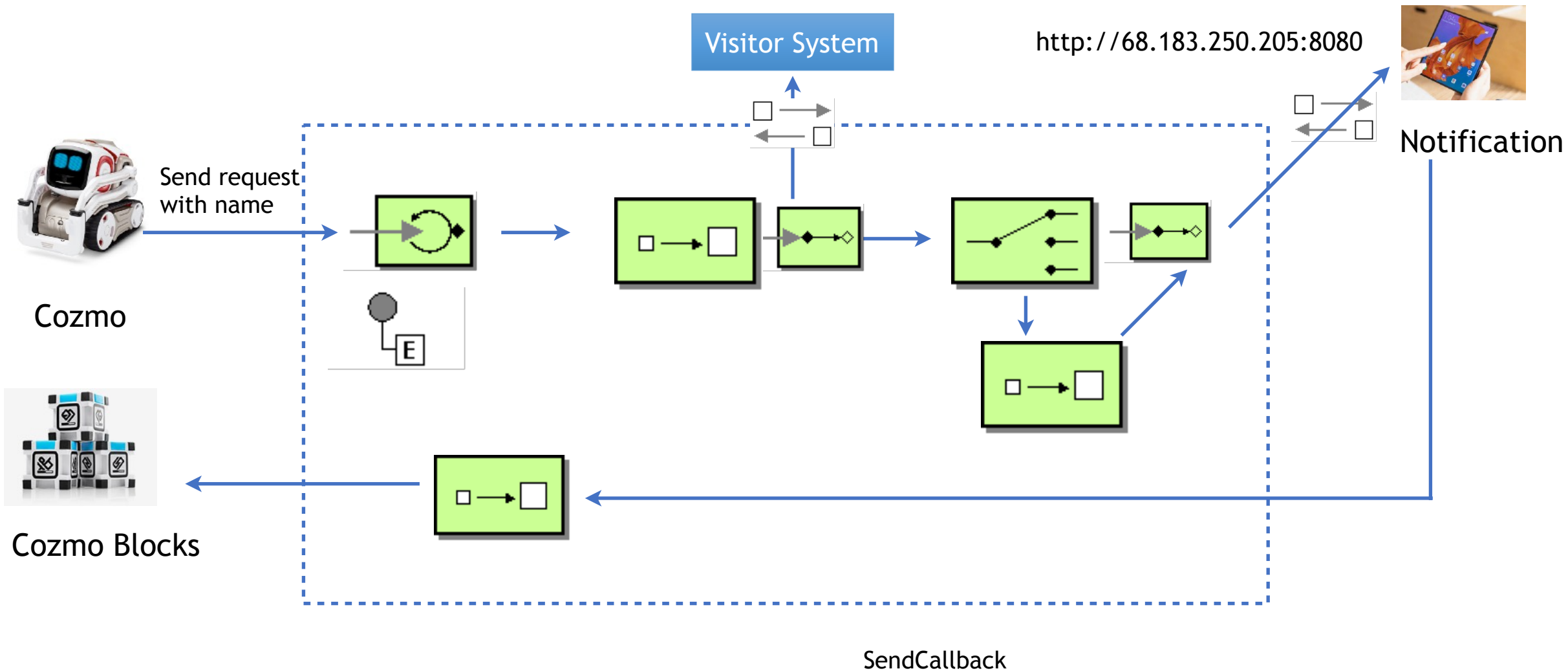


```
from("jms://inQueue")
  .choice()
    .when(header("foo").isEqualTo("bar"))
      .to("jms://widgetQueue")
    .otherwise()
      .to("jms://gadgetQueue")
```

```
<route>
  <from uri="jms://inQueue"/>
  <choice>
    <when>
      <xpath>$foo = 'bar'</xpath>
      <to uri="jms://widgetQueue"/>
    </when>
    <otherwise>
      <to uri="jms://gadgetQueue"/>
    </otherwise>
  </choice>
</route>
```

<https://camel.apache.org/eip.html>

Camel Routes



Why Camel



KubeCon



CloudNativeCon



OPEN SOURCE SUMMIT

China 2019

```
public void configure() throws Exception {  
    restConfiguration().component("restlet").host("0.0.0.0").port("9080");
```

RestFront

```
    rest().get("/visitor/{name}").to("direct:visitorCheckIn");
```

```
    from("direct:visitorCheckIn")
```

VisitorCheckinService

```
        .transform().simple("{\\"visitorName\\":\\"${header.name}\\\"}")
```

```
        .removeHeaders("*")
```

```
        .setHeader(Exchange.HTTP_METHOD, constant(org.apache.camel.component.http4.HttpMethods.POST))
```

```
        .setHeader(Exchange.CONTENT_TYPE, constant("application/json"))
```

```
        .to("http4://" + visitorServiceAddress + "/visitors/checkIn")
```

```
        .transform().jsonpath("$.visitorName")
```

```
        .to("direct:notifyService");
```

```
    from("direct:notifyService")
```

NotifyService

```
        .choice()
```

```
            .when().simple("${body} == \"Alex\"")
```

```
                .transform().simple("VIP ${body}")
```

```
            .end()
```

```
        .setHeader("message").simple("Welcome ${body} to robot world")
```

```
        .transform().simple("{\\"message\\":\\" ${body} is here.\\"}")
```

```
        .setHeader(Exchange.HTTP_METHOD, constant(org.apache.camel.component.http4.HttpMethods.POST))
```

```
        .setHeader(Exchange.CONTENT_TYPE, constant("application/json"))
```

```
        .to("http4://" + notificationServiceAddress + "/notification/notify")
```

```
        .setBody().simple("${headers.message}")
```

```
        // We need to clean up the header to send out the message
```

```
        .removeHeaders("*");
```

```
}
```


How to run Camel Application



KubeCon



CloudNativeCon



OPEN SOURCE SUMMIT

China 2019

- Spring Boot Application with Camel Route
- Need to add the dependencies of Camel component
- Build the java artifacts and images
- Push the images into docker registry
- Run the Application from the docker registry

What if we update the Camel Route?

SOA to Cloud Native



KubeCon



CloudNativeCon

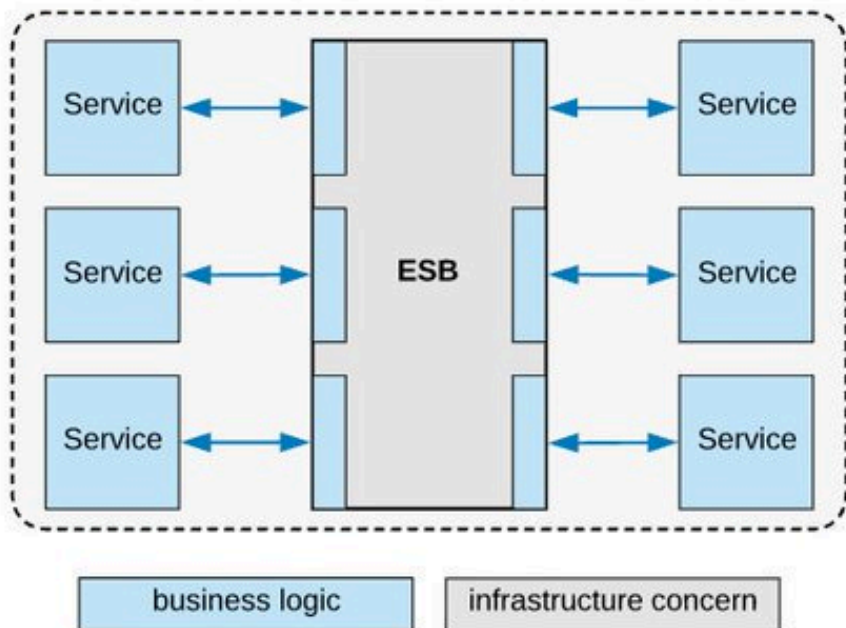


OPEN SOURCE SUMMIT

China 2019

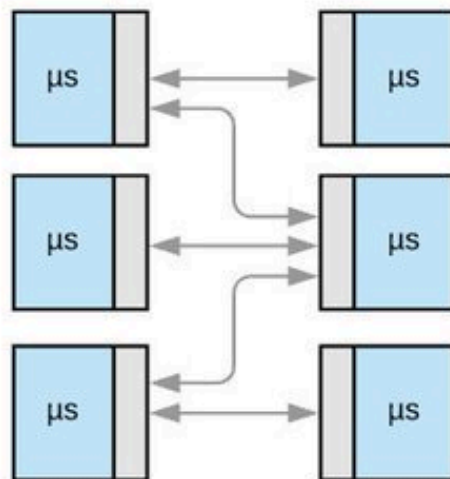
Service Oriented Architecture

(Smart pipes, dumb endpoints)



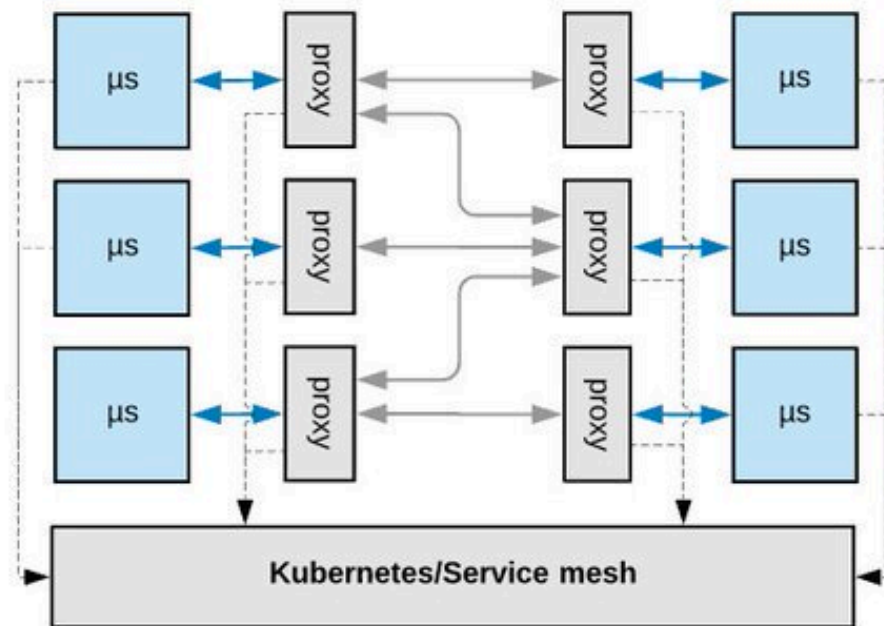
Microservices Architecture

(Smart endpoints, dumb pipes)



Cloud Native Architecture

(Infrastructure focused smart platform, business logic focused smart services)



<https://www.infoq.com/articles/microservices-post-kubernetes>

Camel-K



KubeCon



CloudNativeCon



OPEN SOURCE SUMMIT

China 2019

- Lightweight Integration platform
- Running Camel Application natively on Kubernetes.
- Designed for **serverless** and **microservice architectures**.
- Leveraging the **Operator SDK**
- Building Application on top of **Knative Eventing**

<https://github.com/apache/camel-k>

K8s Operator



KubeCon



CloudNativeCon



OPEN SOURCE SUMMIT

China 2019

```
from("gate:xxx").to("kafak:topic");  
...  
from("kafak:topic").transform(...)
```



Customer
Resource Define



```
kind: Integration  
apiVersion: camel.apache.org/  
v1alpha1  
metadata:
```

```
  name: my-integration
```

```
spec:
```

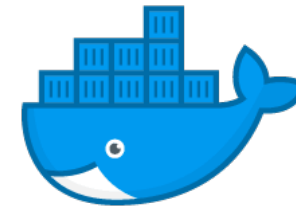
```
  sources:
```

```
    - name: source.java
```

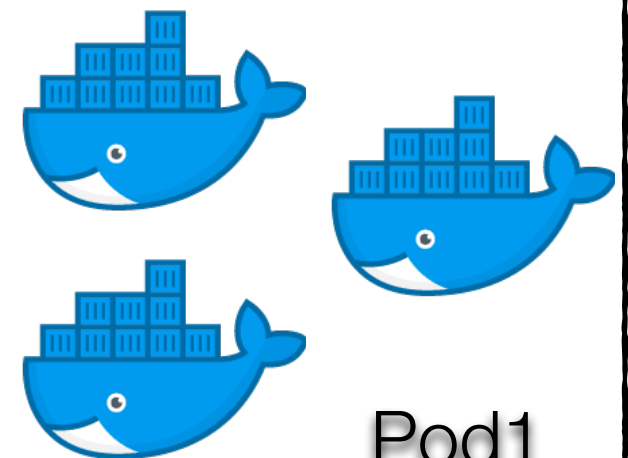
```
    content:
```

```
from("gate:xxx").to("kafak:topic");  
...
```

1. Choose a Runtime
2. Create a project
3. Add dependencies
4. Build docker image
5. Deploy



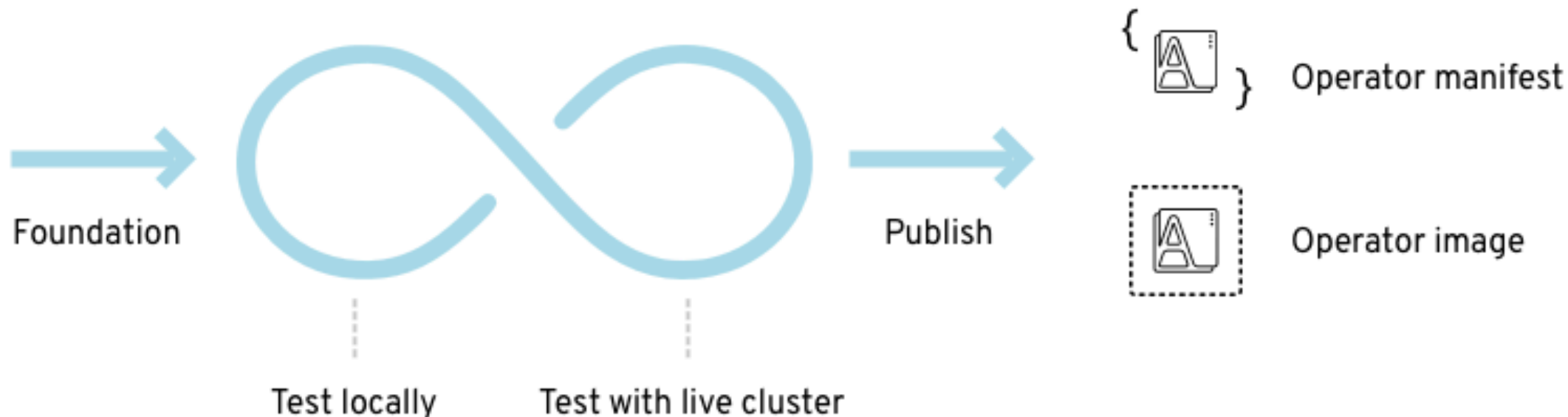
Pod1



Pod1

Operator SDK

Operator SDK *Build, test, iterate*



Knative



KubeCon

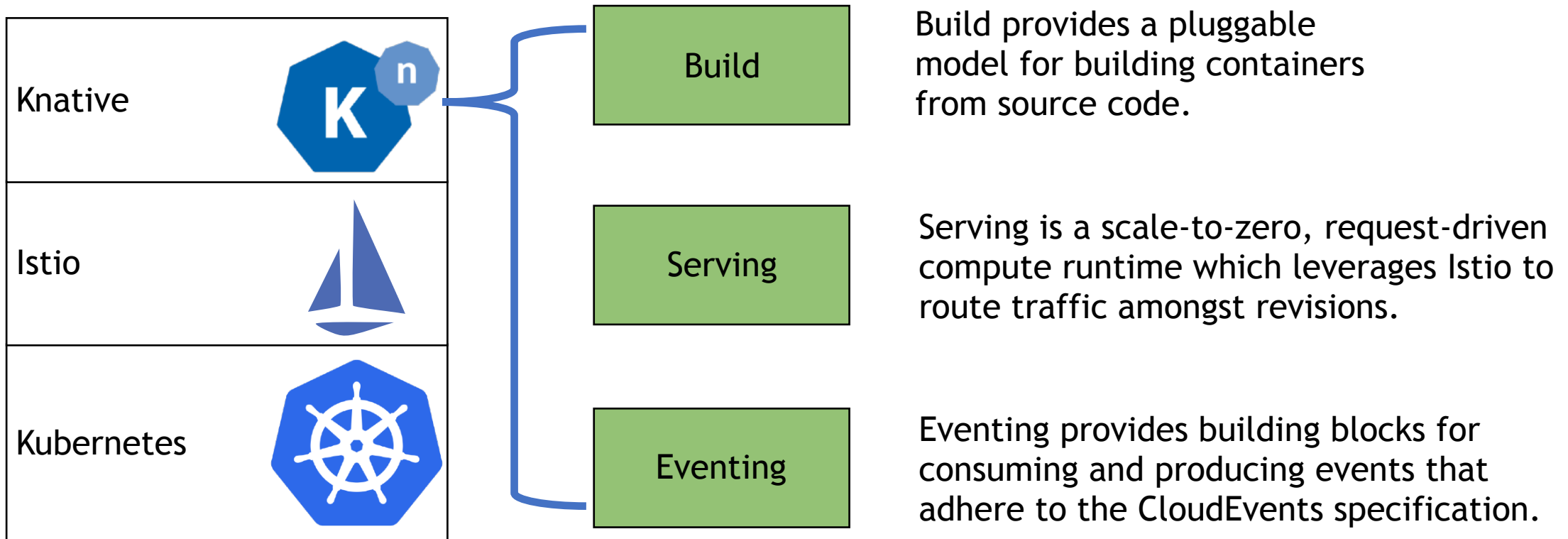


CloudNativeCon



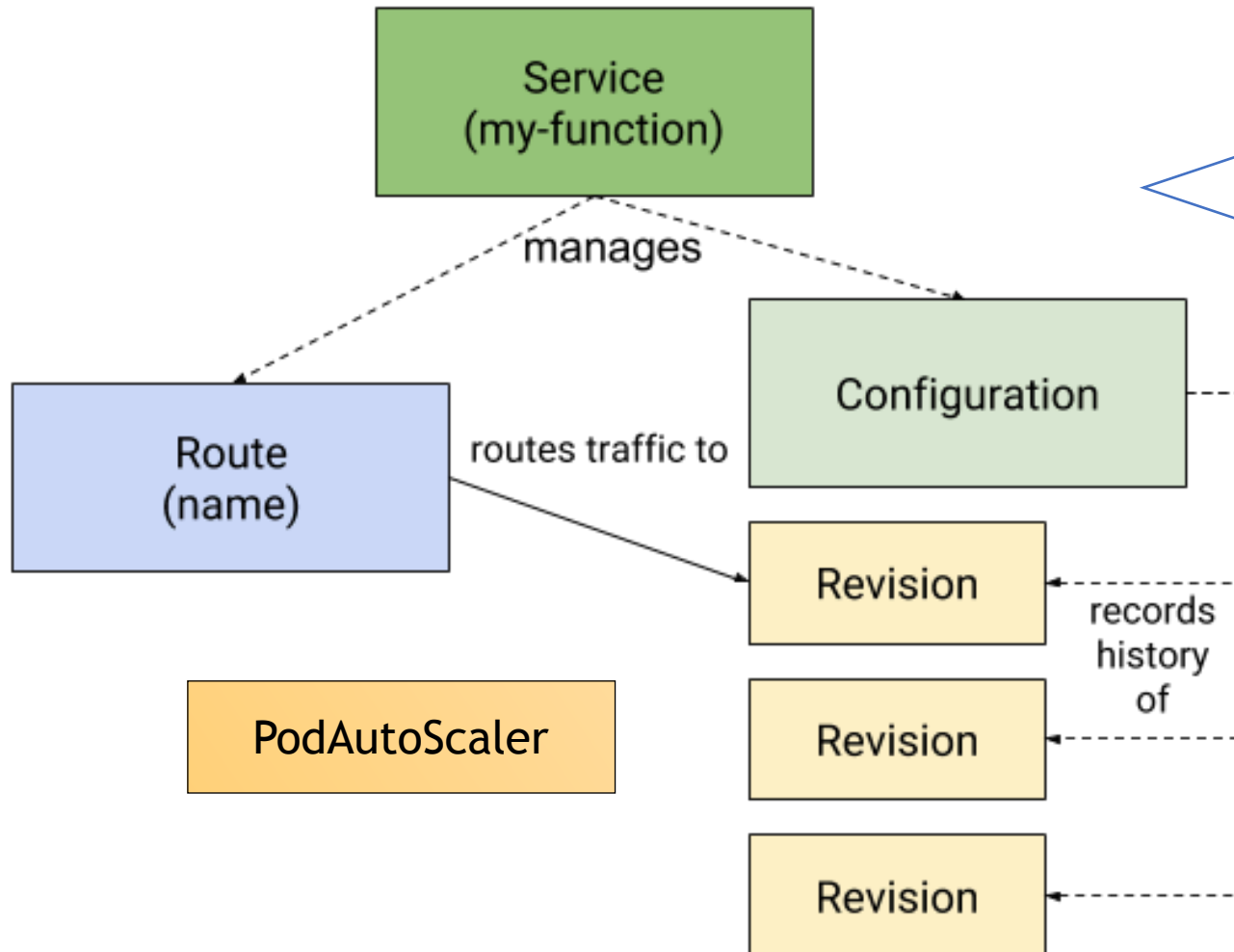
OPEN SOURCE SUMMIT

China 2019

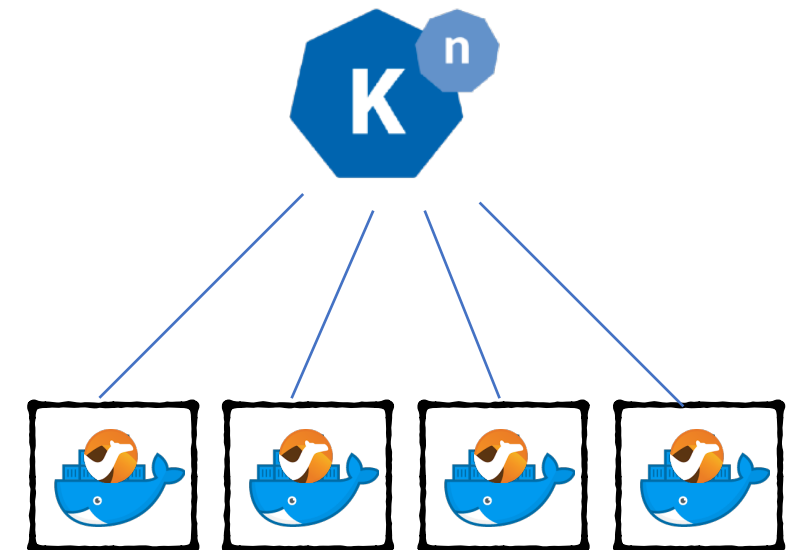


<https://knative.dev/>

Knative Service With Camel-K



Camel-K



AutoScale Min ... Max
AutoScale Target

Knative Event



KubeCon

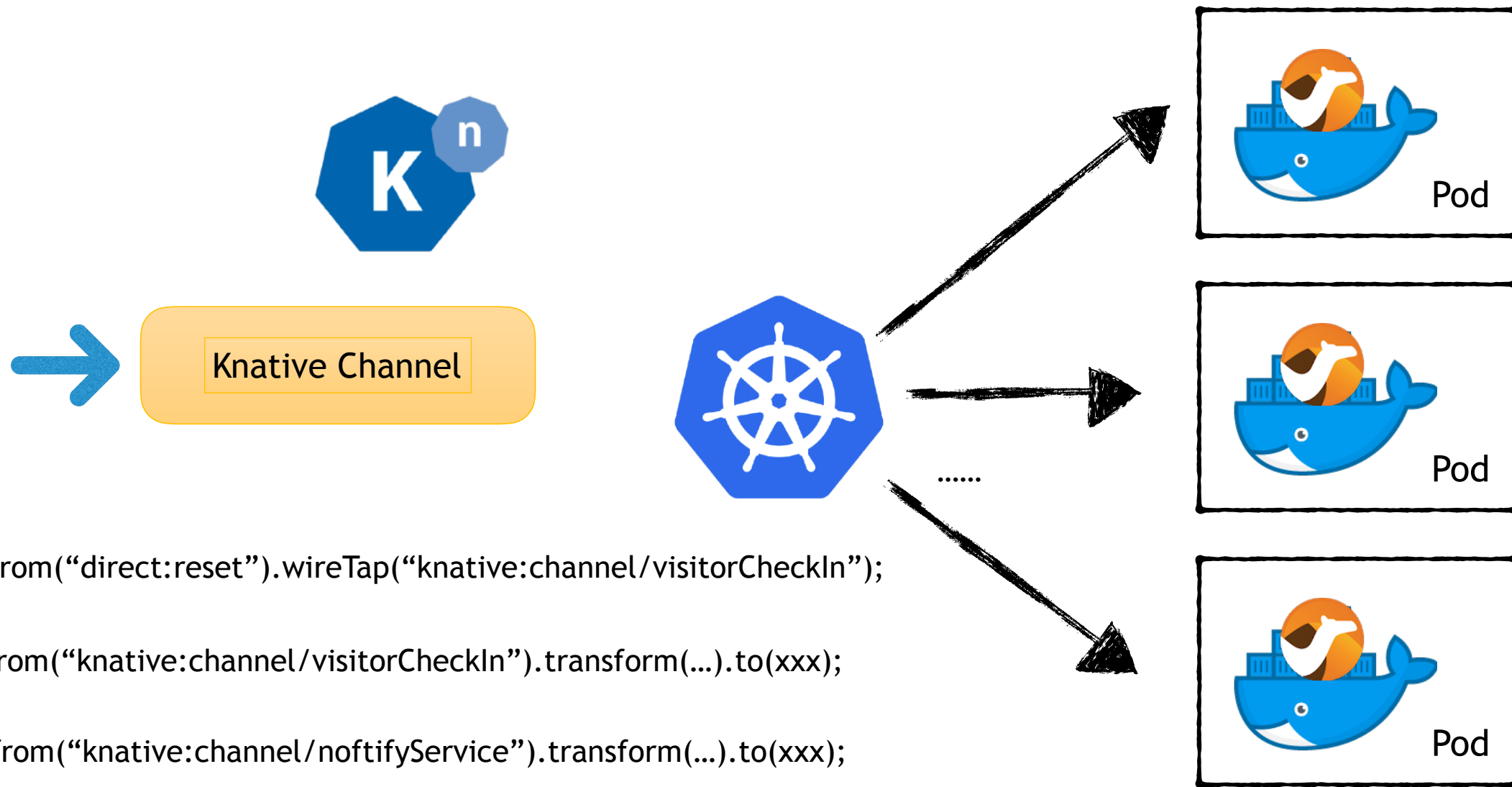


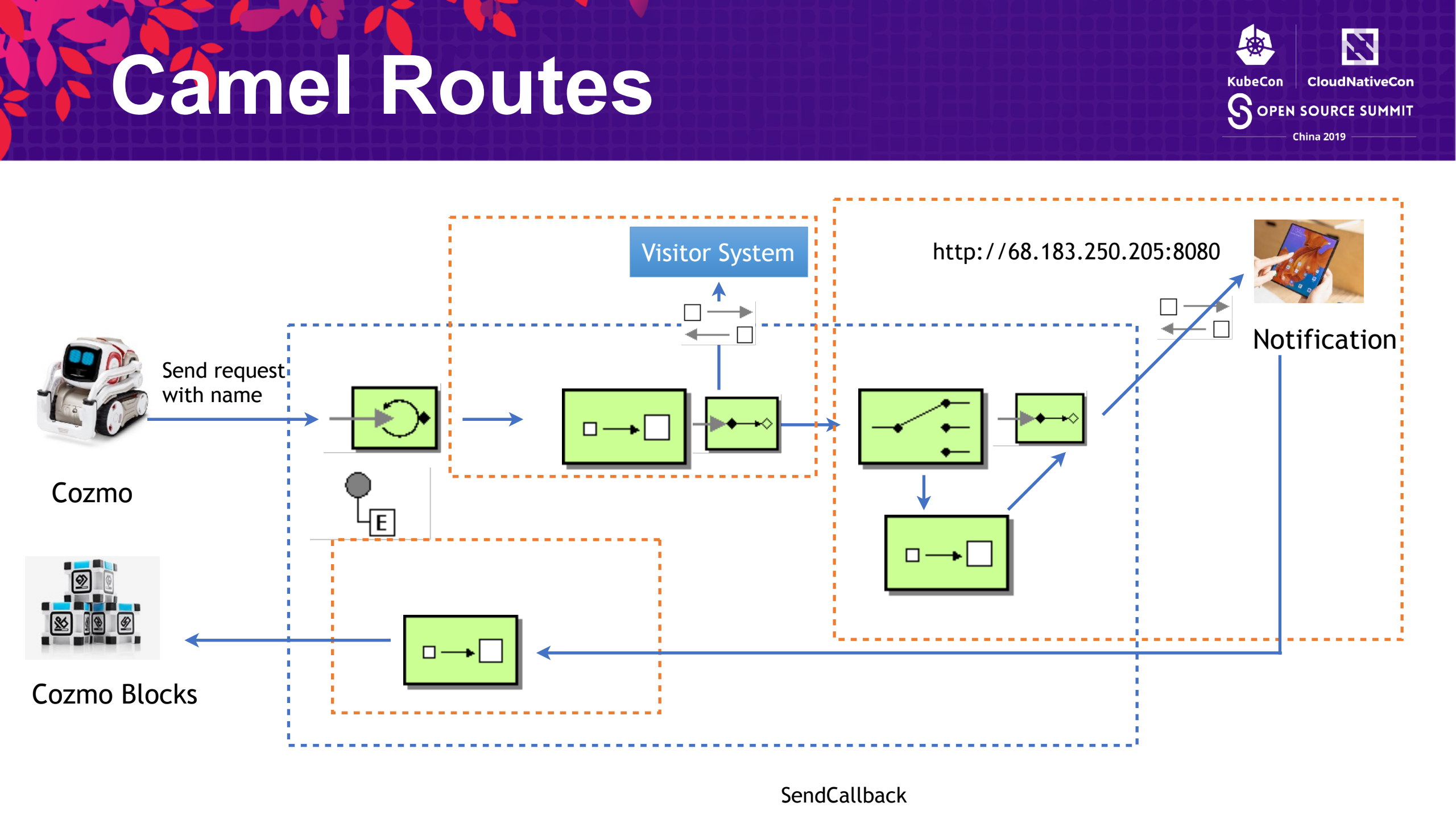
CloudNativeCon



OPEN SOURCE SUMMIT

China 2019







Kubecon Camel 技术讨论群



<https://github.com/WillemJiang/smart-park-demo>



KubeCon



CloudNativeCon

OPEN SOURCE SUMMIT

China 2019

