

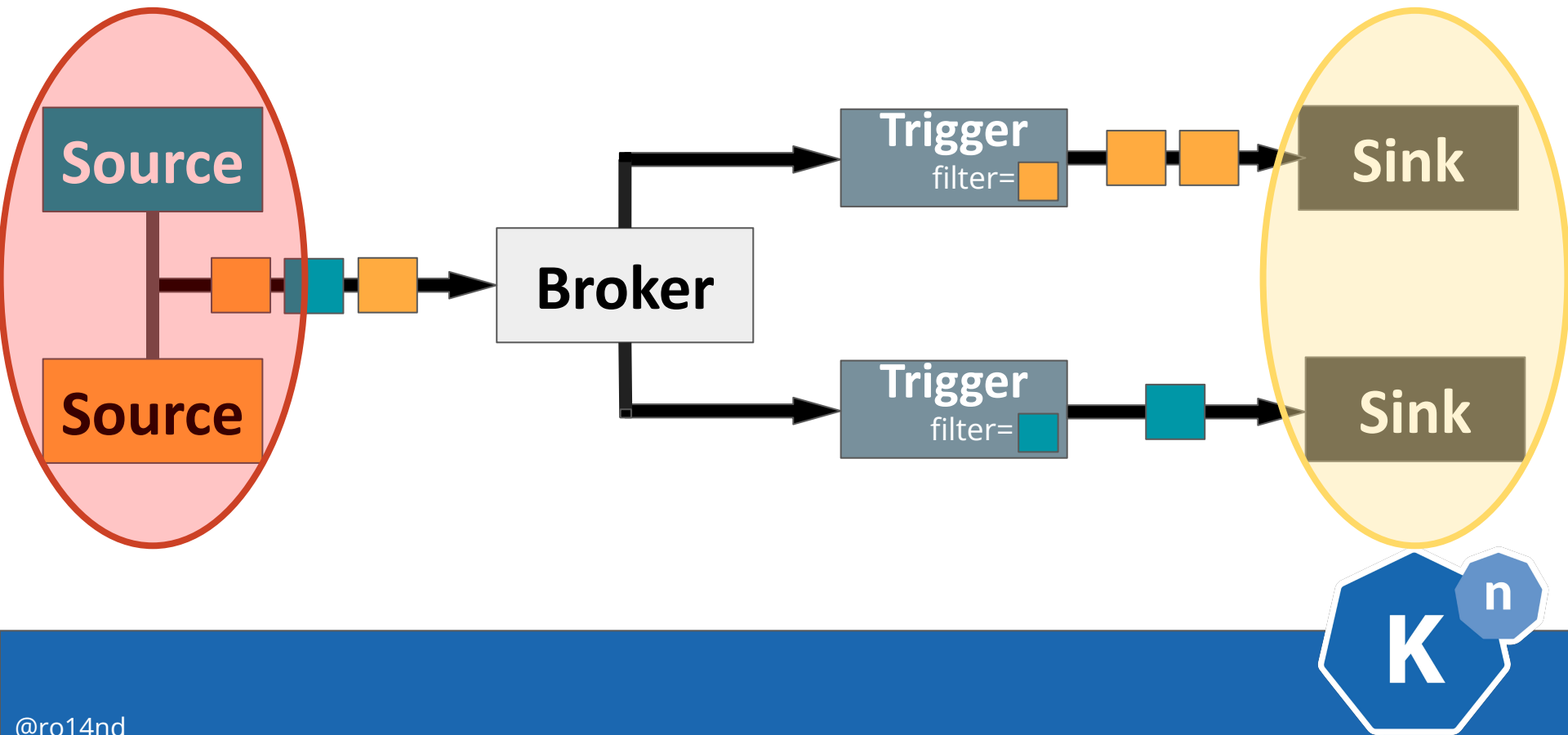
Knative and Kamelets

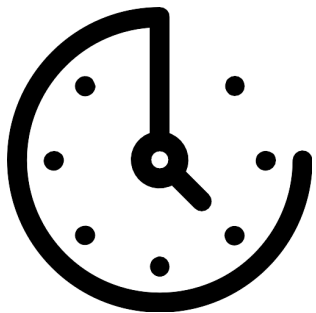
Roland Huß

Senior Principal Software Engineer, Red Hat
Knative Client WG Lead, Knative TOC member

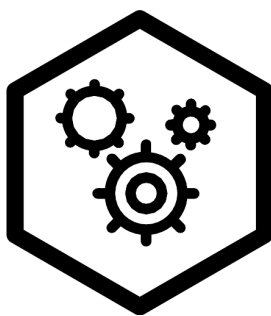


The Knative Event Mesh





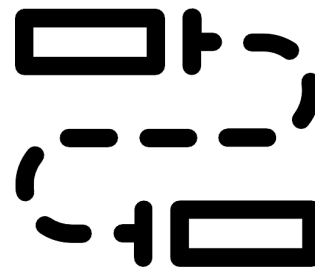
PingSource



ApiServerSource



ContainerSource



SinkBinding



Event Sources

- Entry point for CloudEvents into the Knative Eventing mesh
- Not difficult to implement but requires quite some boilerplate
 - Controller & CRDs for custom resources
 - **ContainerSource** and **SinkBinding** need custom images to provide functionality
- Discovery is difficult
 - EventRegistry (but currently not much used)
 - CRDs often not listable by standard users

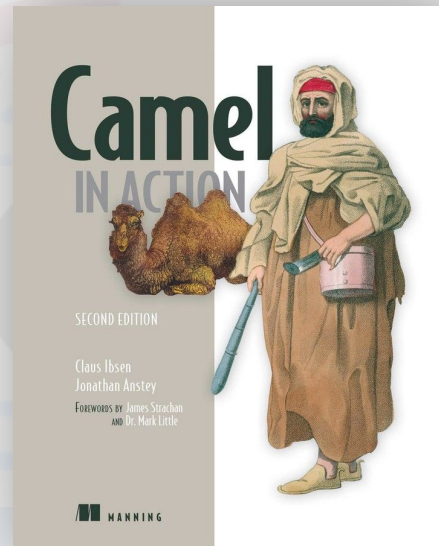
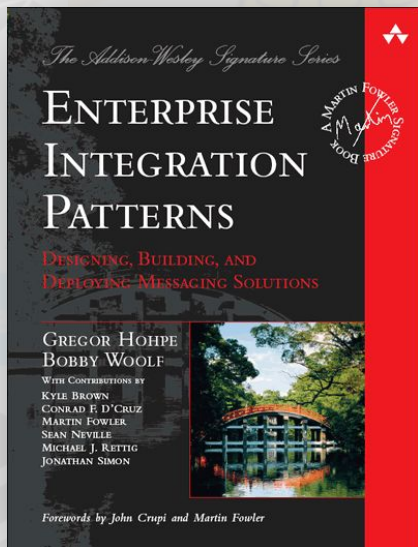
Is there an alternative to enable sources ?







APACHE® Camel



Apache Camel and Friends

- Apache Camel
 - Swiss knife of Enterprise Application Integration
 - 340+ components and connectors
 - Open Source, Java-based library and runtime
 - >10 years of development - still one of the most active Apache projects
- Camel-K
 - Camel on Kubernetes
 - Introduces CRD Integration for defining the integration route
- Kamelets
 - Off-the-shelve Camel route snippets with parameters
 - Opinionated packaging of Camel Components
 - Catalog with 70+ Kamelets, constantly growing



Camel Route

```
from("twitter-search:#kduck?accessToken=1234")  
  .marshal()  
  .json()  
  .to("knative:event/default");
```



Camel-K Integration

Integration

```
spec:
  configuration:
    - type: property
      value: 'accessToken = 1234'
    - type: property
      value: 'keywords = #kduck'
  flows:
    - route:
        from:
          steps:
            - uri: twitter-search-source/source
              to: knative:event?kind=Broker
        id: binding
```



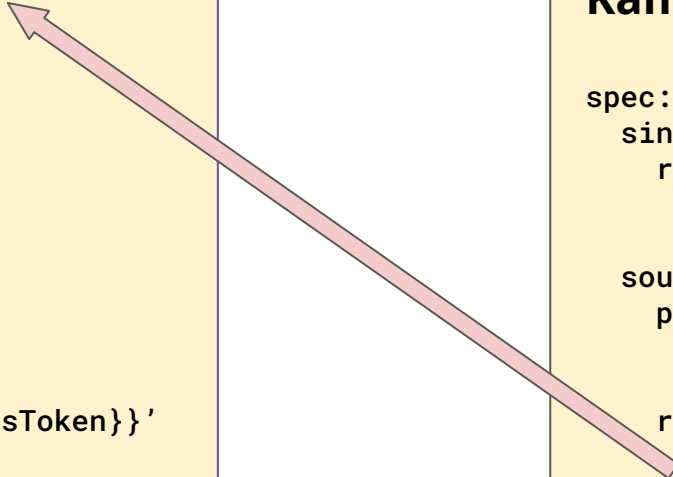
Kamelet and KameletBinding

Kamelet

```
meta:  
  name: twitter-search-source  
spec:  
  definition:  
    properties:  
      accessToken:  
        type: string  
      keywords:  
        type: string  
  
  template:  
    from:  
      parameters:  
        accessToken: '{{accessToken}}'  
      steps:  
        - marshal:  
            json: {}  
        - to: kamelet:sink  
      uri: twitter-search:{{keywords}}
```

KameletBinding

```
spec:  
  sink:  
    ref:  
      kind: Broker  
      name: default  
  source:  
    properties:  
      accessToken: 1234  
      keywords: '#kduck'  
    ref:  
      kind: Kamelet  
      name: twitter-search-source
```



Demo

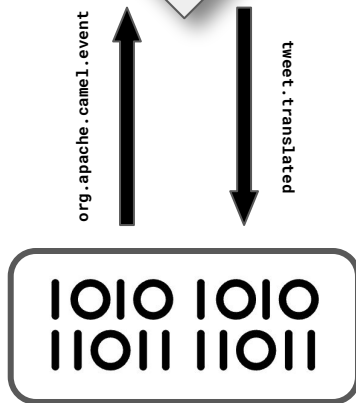
Function for translating
to random language



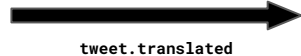
Google Cloud
Translation



Twitter Search #kduck



Knative Broker



Slack Post #knativecon-kamelet-demo



Kamelets & Knative: Outlook

- New Knative client features
 - kn-sink-kamelet plugin
 - Typed options, dynamically inferred from Kamelet
 - Support for Secrets and ConfigMaps
- More Kamelets Sinks and Sources
 - Ideally: All 300+ Camel components
- Integration with KEDA
 - Scale down to zero for Kamelets that have a KEDA Scaler



Thank You!

