

A photograph of a camel race in progress. Several camels are running across a sandy track, with riders on their backs. The riders are wearing white shirts and dark pants, and some are wearing helmets. The camels are in various stages of their stride, and the background is a clear, light blue sky. The text is overlaid on the image.

Camel microservices with Spring Boot and Kubernetes

Claus Ibsen
@davsclaus

JPoint Moscow
April 2018

About me

- Senior Principal Software Engineer at Red Hat
- 10 years as Apache Camel committer
- Author of Camel in Action books
- Based in Denmark



Blog: <http://www.davsclaus.com>
Twitter: @davsclaus
Linkedin: davsclaus

System Integration



Figure 1.1 Camel is the glue between disparate systems.

Integration Framework





APACHE

Camel

PATTERN BASED INTEGRATION

Apache Camel, a powerful pattern-based integration engine with a comprehensive set of connectors and data formats to tackle any integration problem.



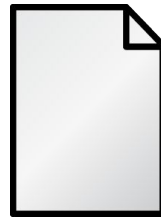
ENTERPRISE INTEGRATION PATTERNS

Build integrations using enterprise best practices.



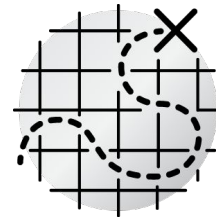
200+ COMPONENTS

Batch, messaging, web services, cloud, APIs, and more ...



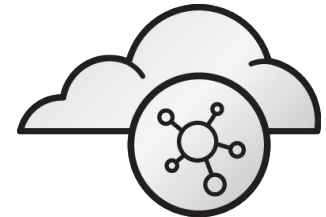
BUILT-IN DATA TRANSFORMATION

JSON, XML, HL7, YAML, SOAP, Java, CSV, and more ...



INTUITIVE ROUTING

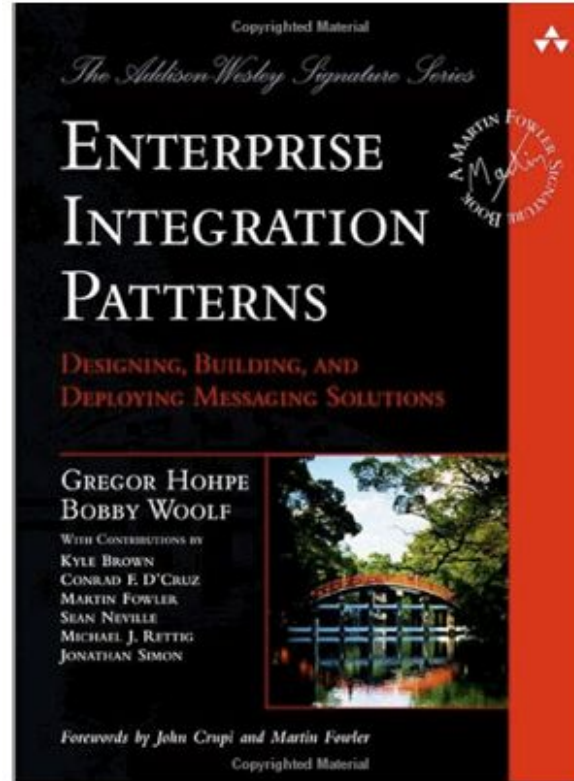
Develop integrations quickly in Java or XML.



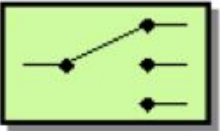

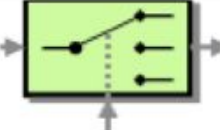
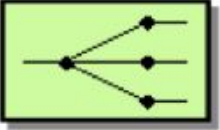
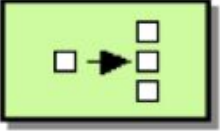
NATIVE REST SUPPORT

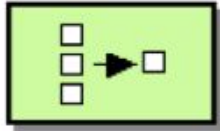
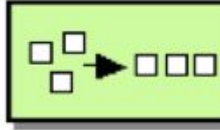
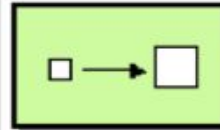
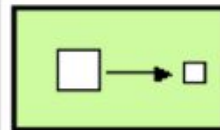
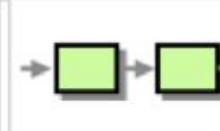
Create, connect, and compose APIs with ease.

Enterprise Integration Patterns

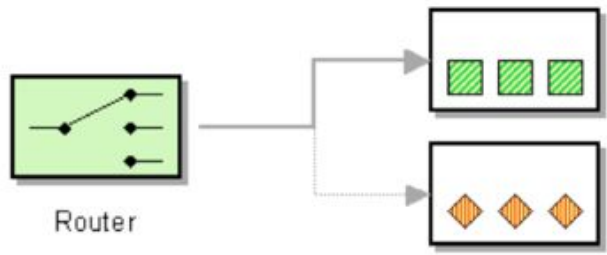


Enterprise Integration Patterns

| | |
|--|----------------------|
|  | Content Based Router |
|  | Message Filter |
|  | Dynamic Router |
|  | Recipient List |
|  | Splitter |

| | |
|--|-------------------|
|  | Aggregator |
|  | Resequencer |
|  | Content Enricher |
|  | Content Filter |
|  | Pipes and Filters |

Camel Routes



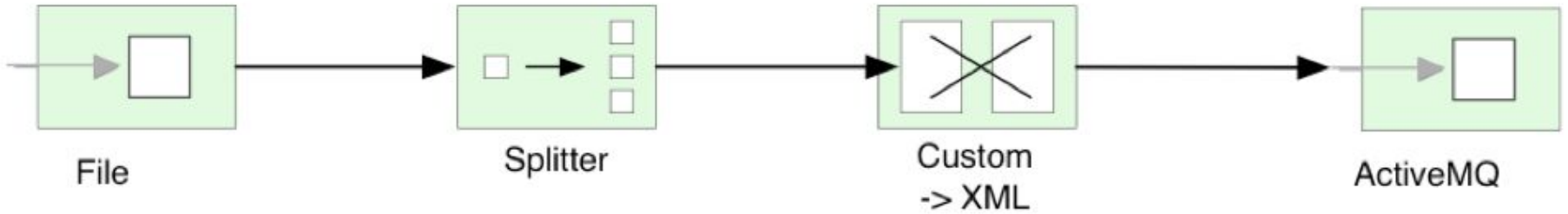
```
from("file:data/inbox")  
  .to("jms:queue:order");
```

Java DSL

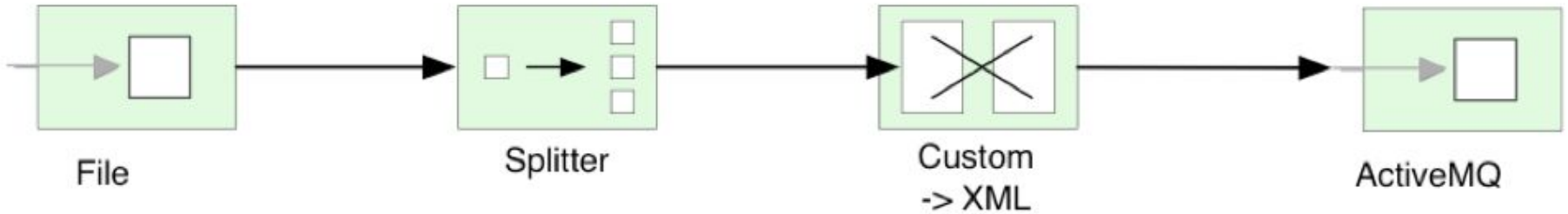
XML DSL

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


Camel Routes with Splitter

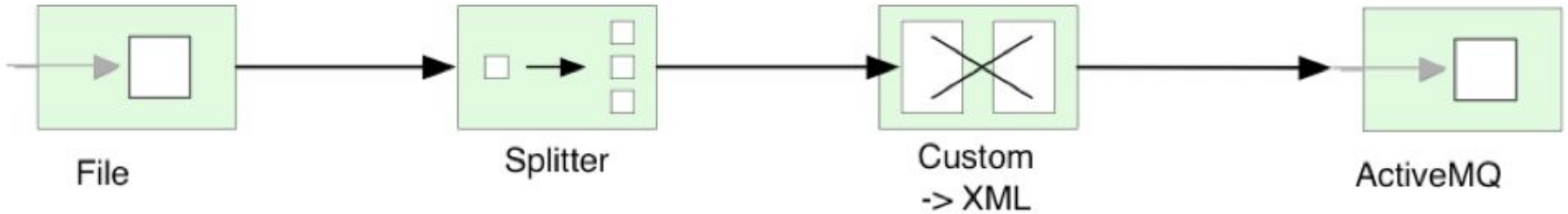


Camel Routes with Splitter



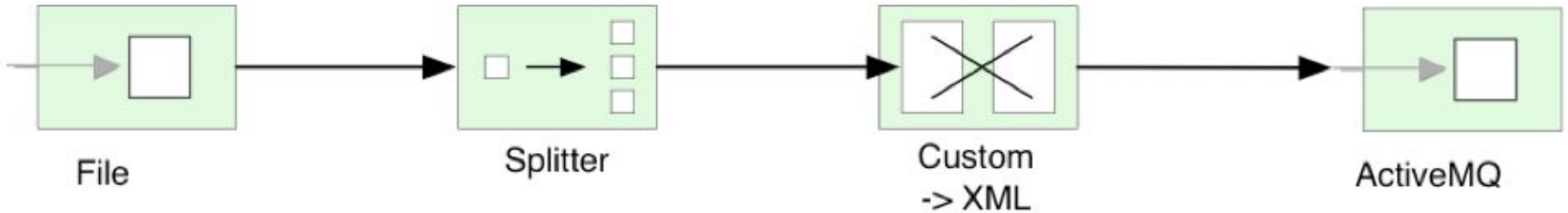
```
from("file:inbox")
```

Camel Routes with Splitter



```
from("file:inbox")  
    .split(body().tokenize("\n"))
```

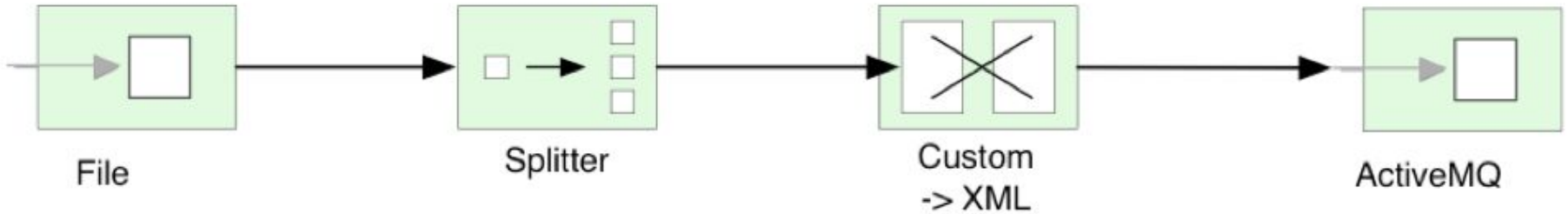
Camel Routes with Splitter



```
from("file:inbox")  
    .split(body().tokenize("\n"))  
    .marshal(customToXml)
```

Custom data transformation

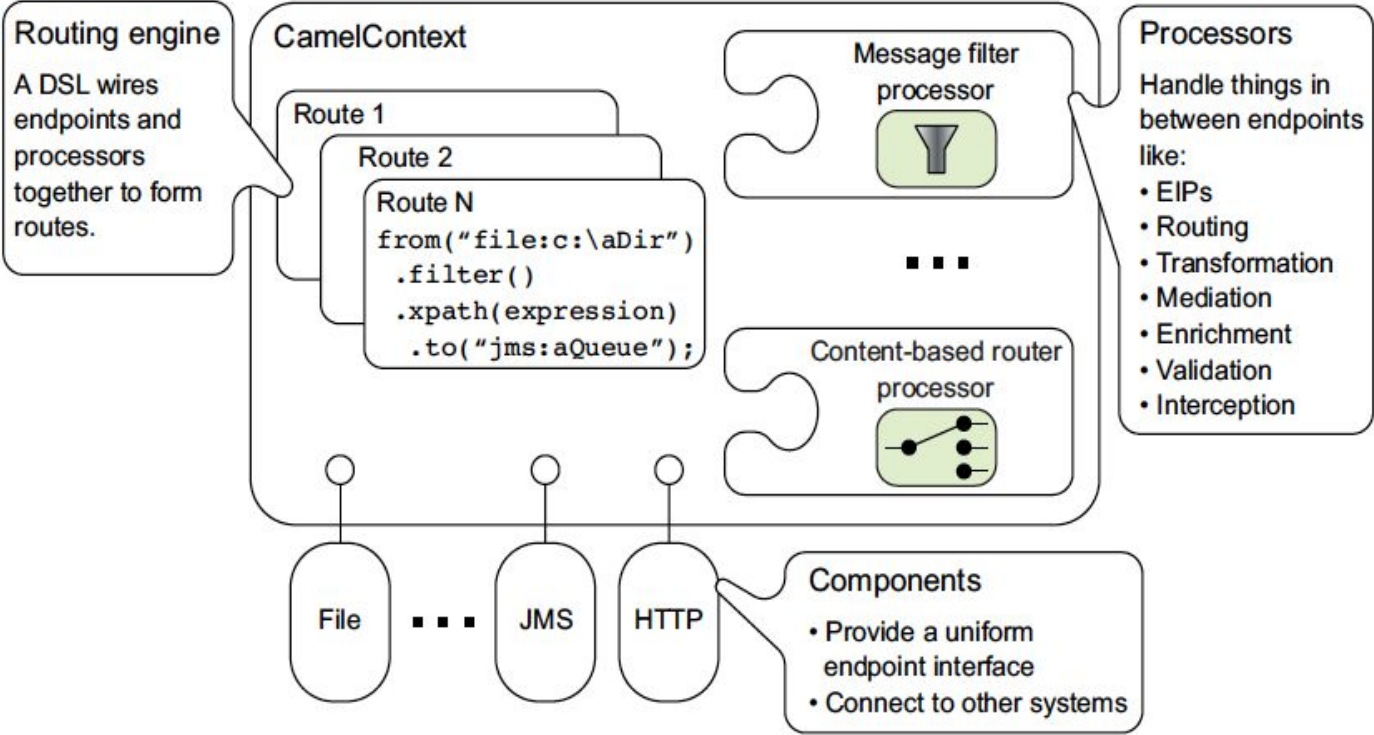
Camel Routes with Splitter



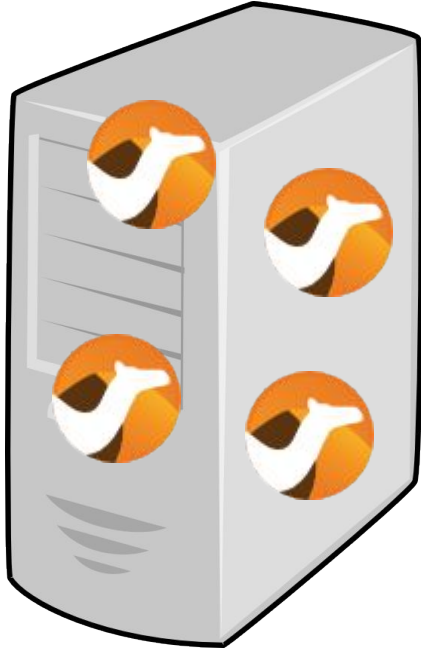
```
from("file:inbox")  
    .split(body().tokenize("\n"))  
    .marshal(customToXml)  
    .to("activemq:line");
```

Custom data transformation

Camel Architecture



Camel runs everywhere

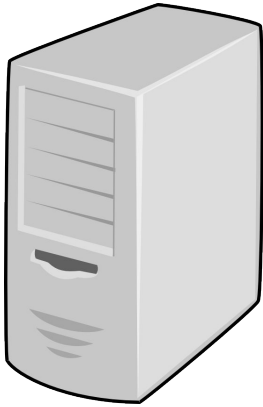


Application
Servers

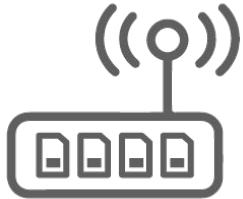


Linux
Containers

Camel connects everything



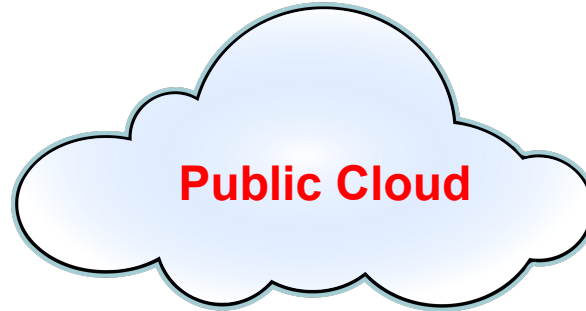
Enterprise Systems



IoT

- File
- FTP
- JMS
- AMQP
- JDBC
- SQL
- TCP/UDP
- Mail
- HDFS
- JPA
- MongoDB
- Kafka
- ...

- CoAP
- MQTT
- PubNub



Public Cloud

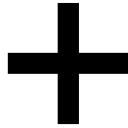
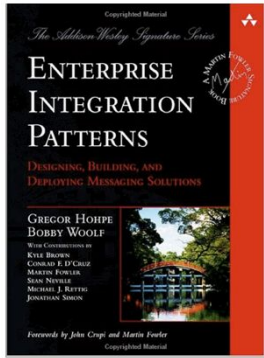
- AWS
 - S3
 - SQS
 - Kinesis
 - ...
- Google
 - BigQuery
 - PubSub
- Azure
 - Blob
 - Queue

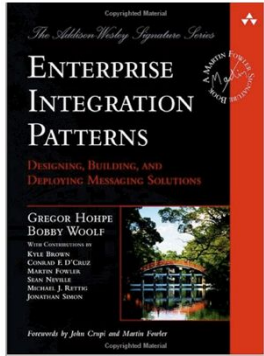
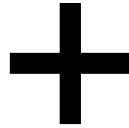
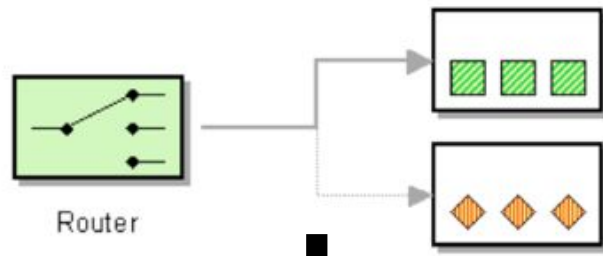


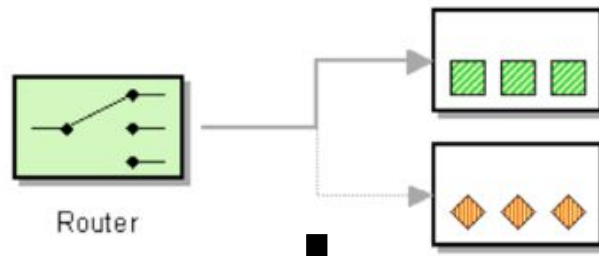
SaaS

- Box
- Dropbox
- Facebook
- LinkedIn
- Salesforce
- SAP
- ServiceNow







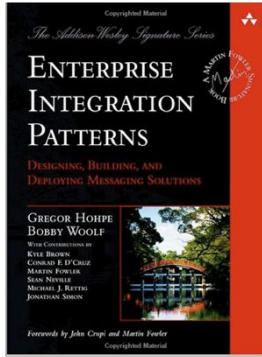
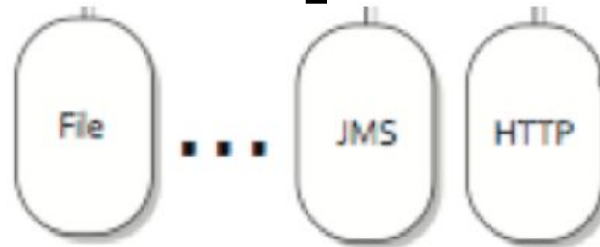


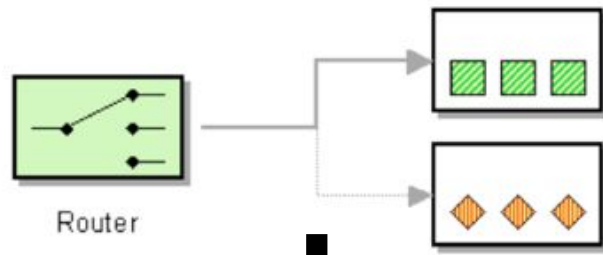
+

+



+



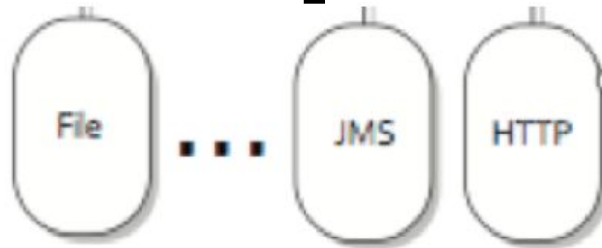


+

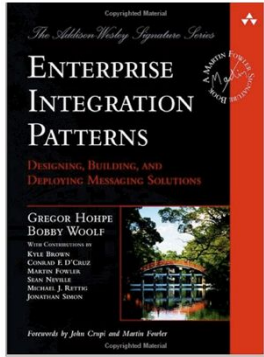
+



+



=



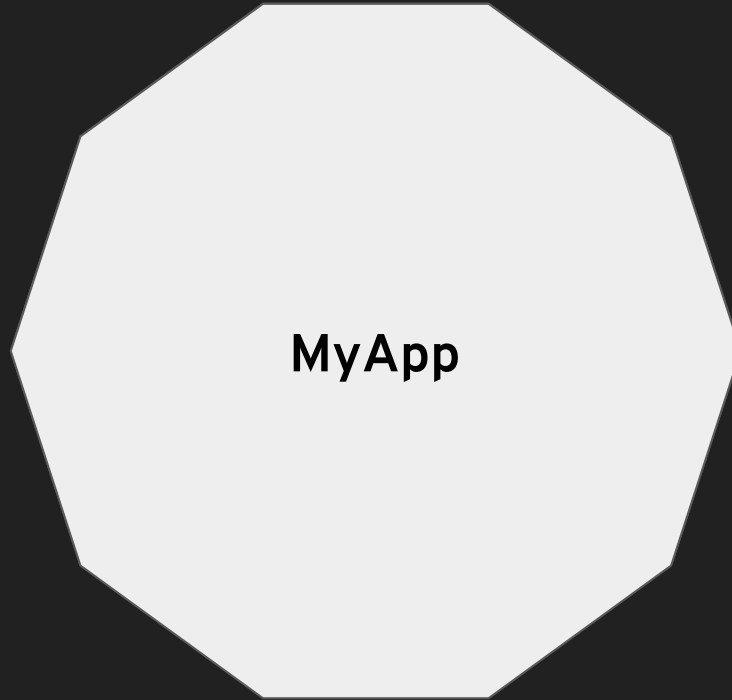


Who is using Camel?

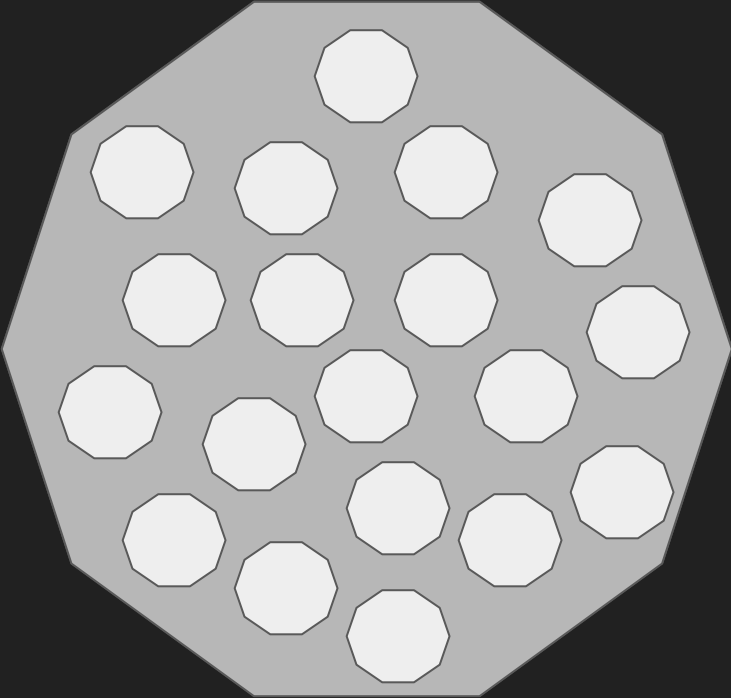


**What about Camel in
the Cloud?**

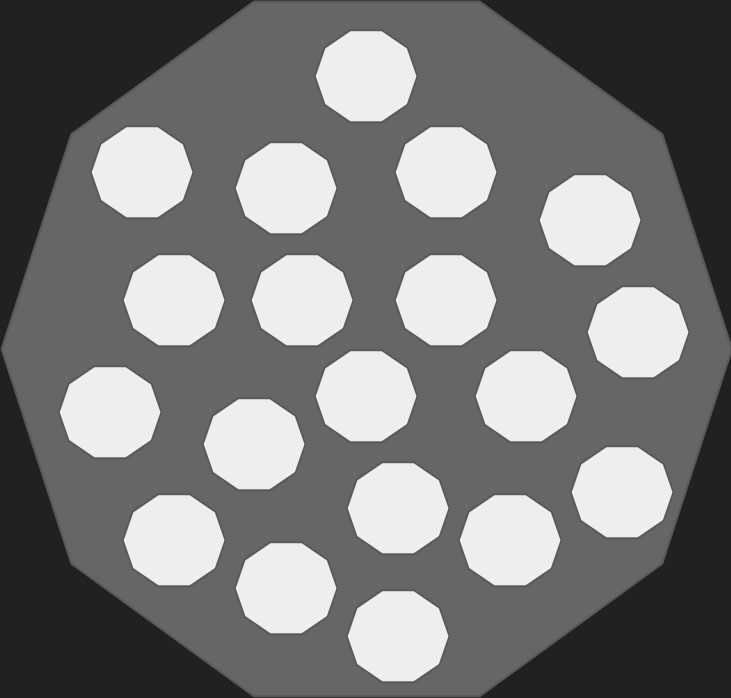
Monolith



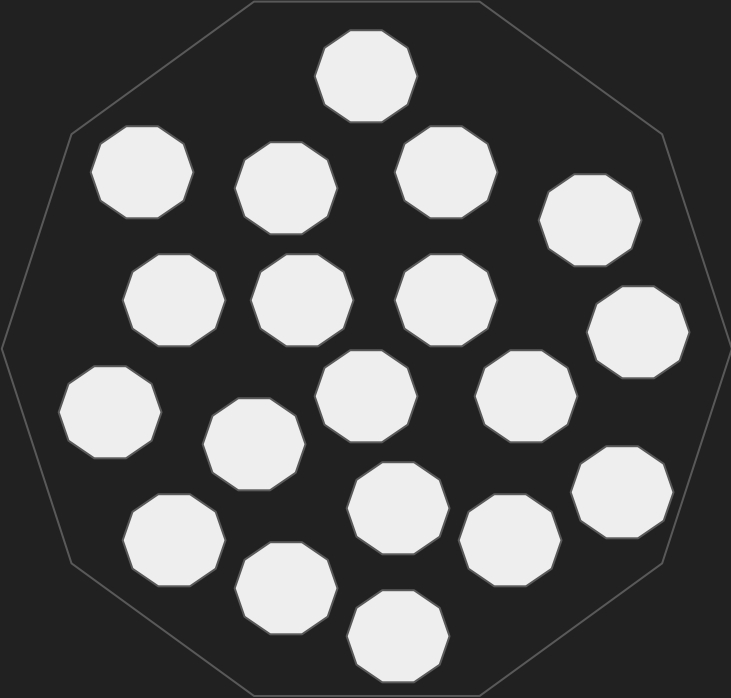
Microservices



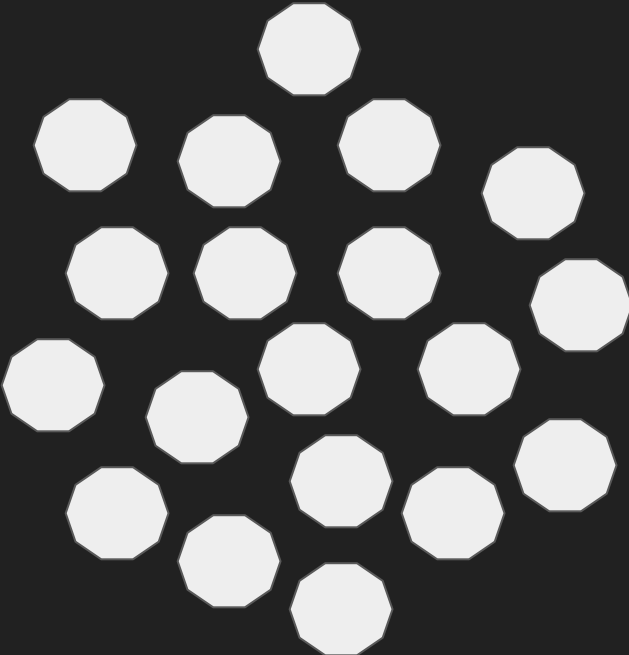
Microservices



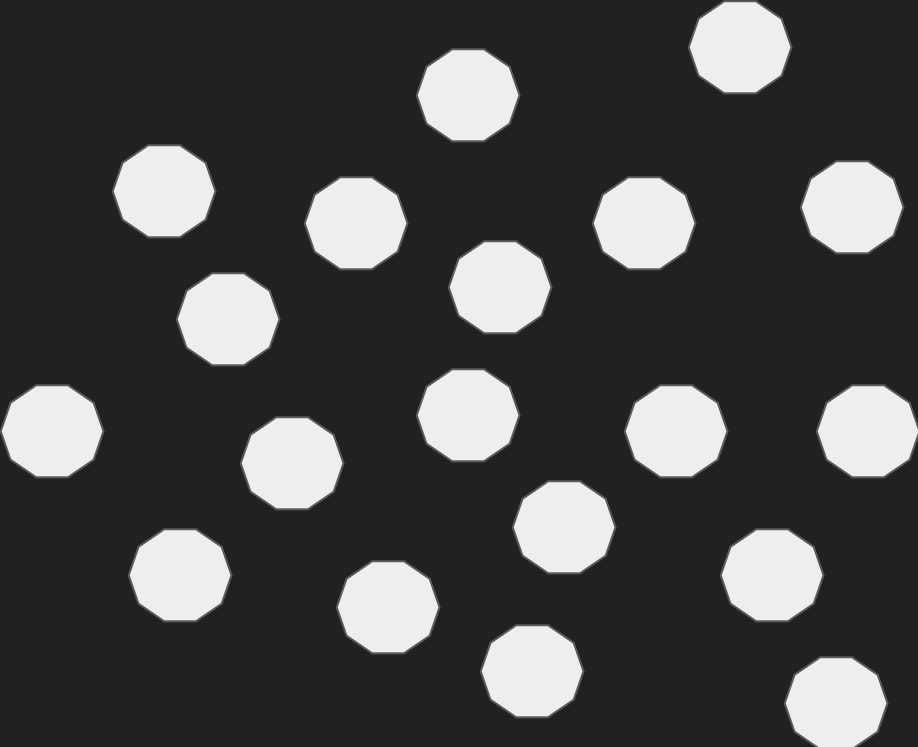
Microservices



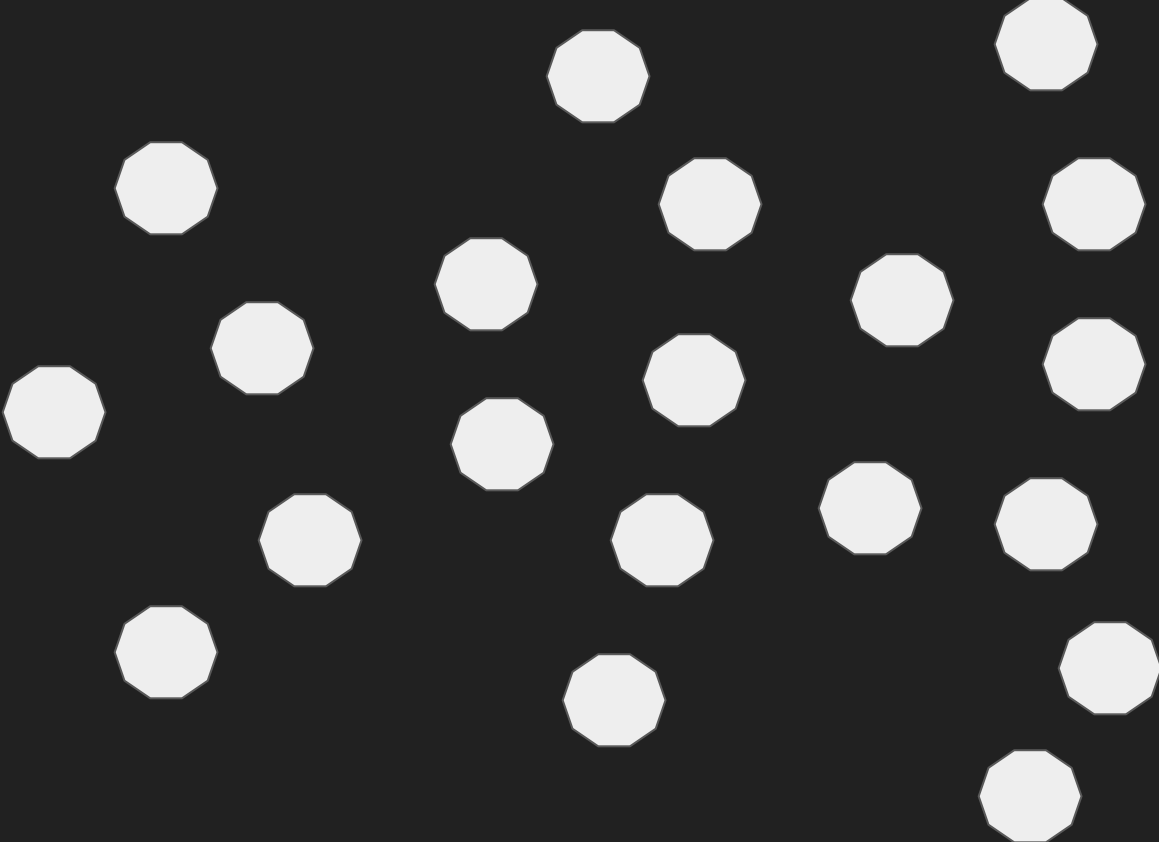
Microservices



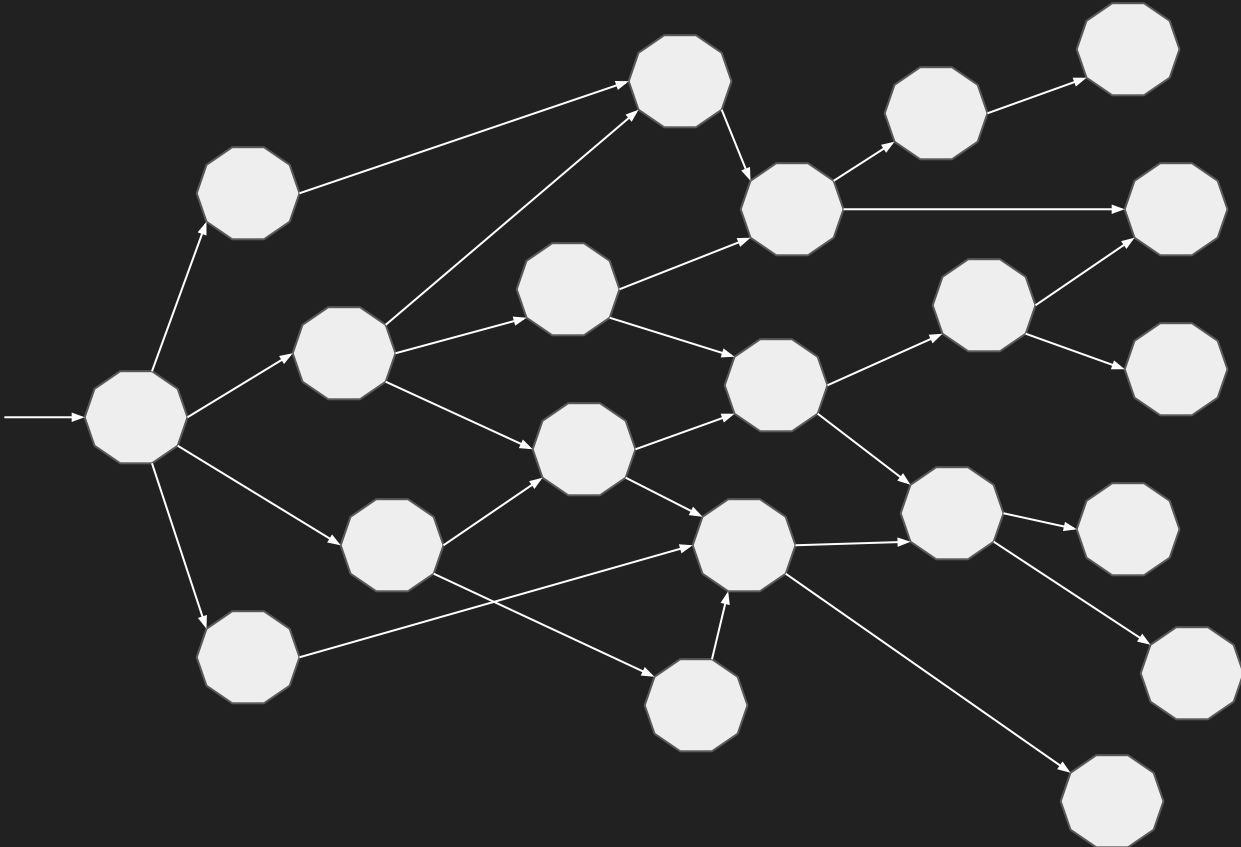
Microservices



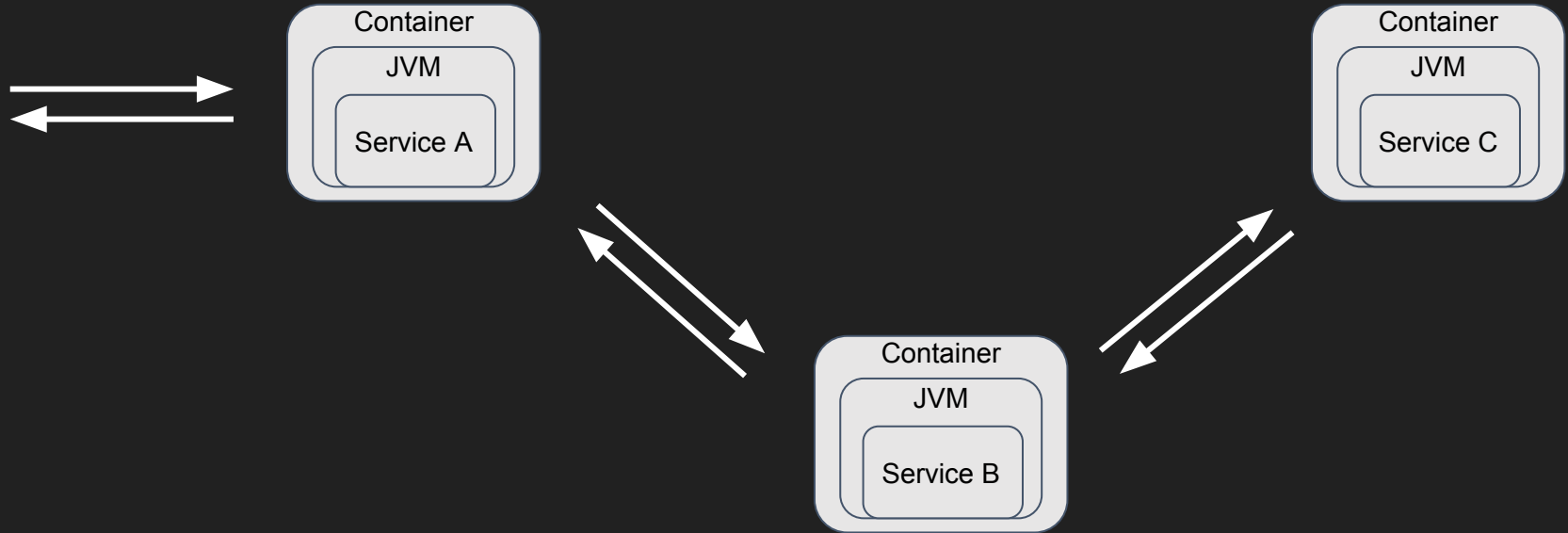
Microservices



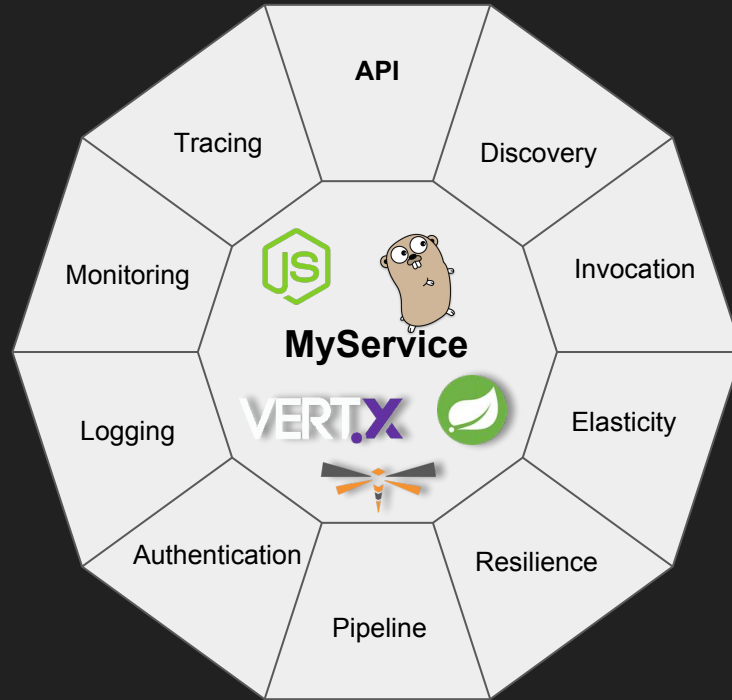
Network of Services



Microservices == Distributed Computing

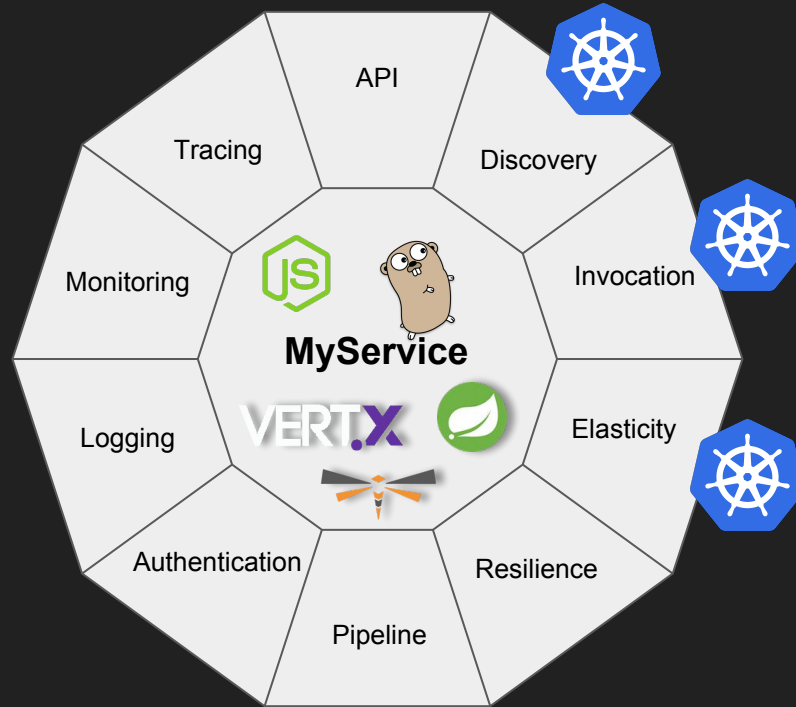


Microservices'ilities

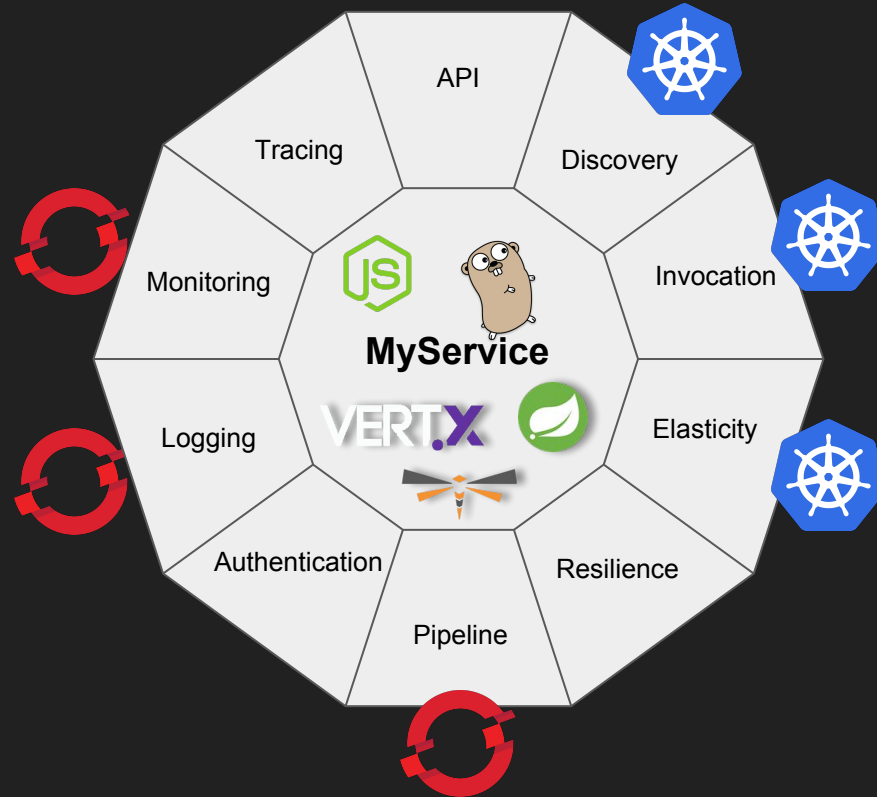




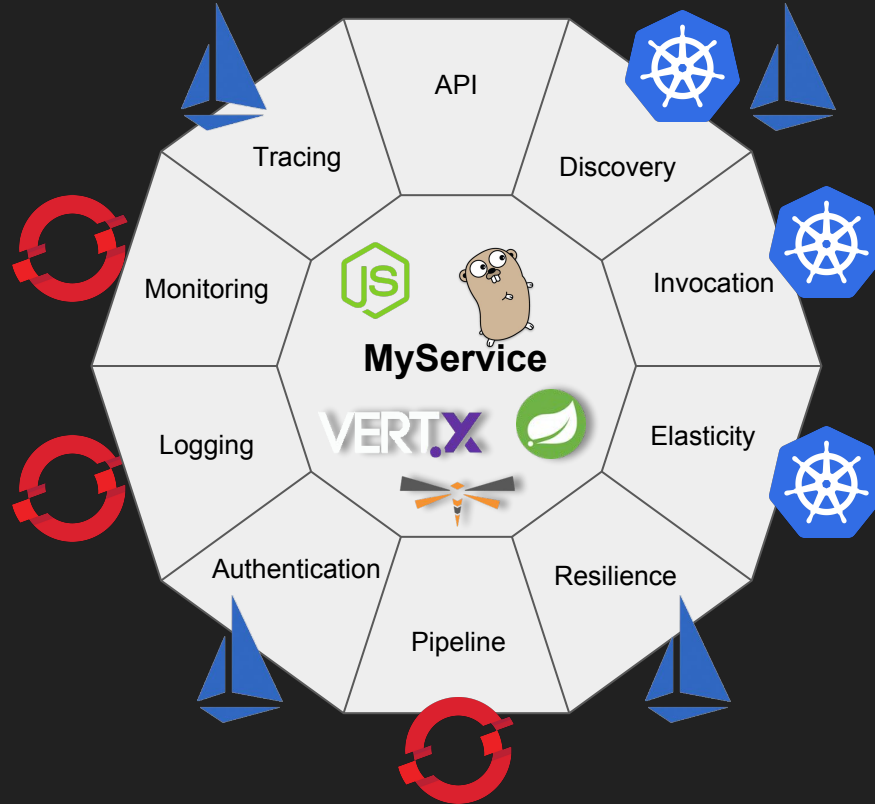
Microservices'ilities + Kubernetes



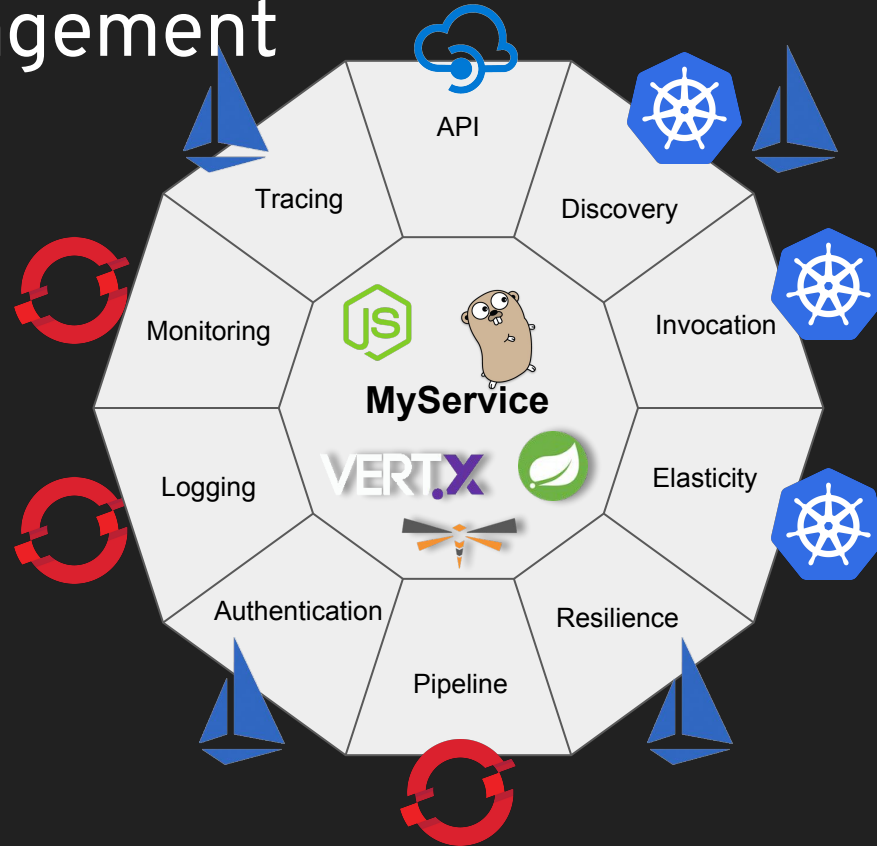
Microservices'ilities + PaaS



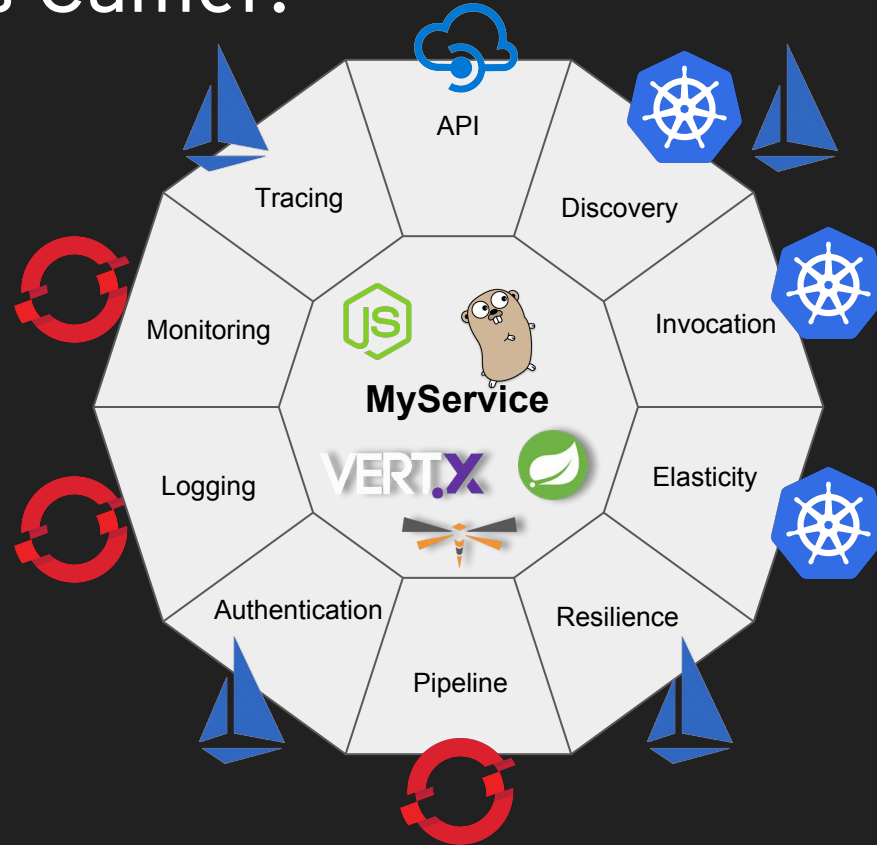
Microservices'ilities + Istio



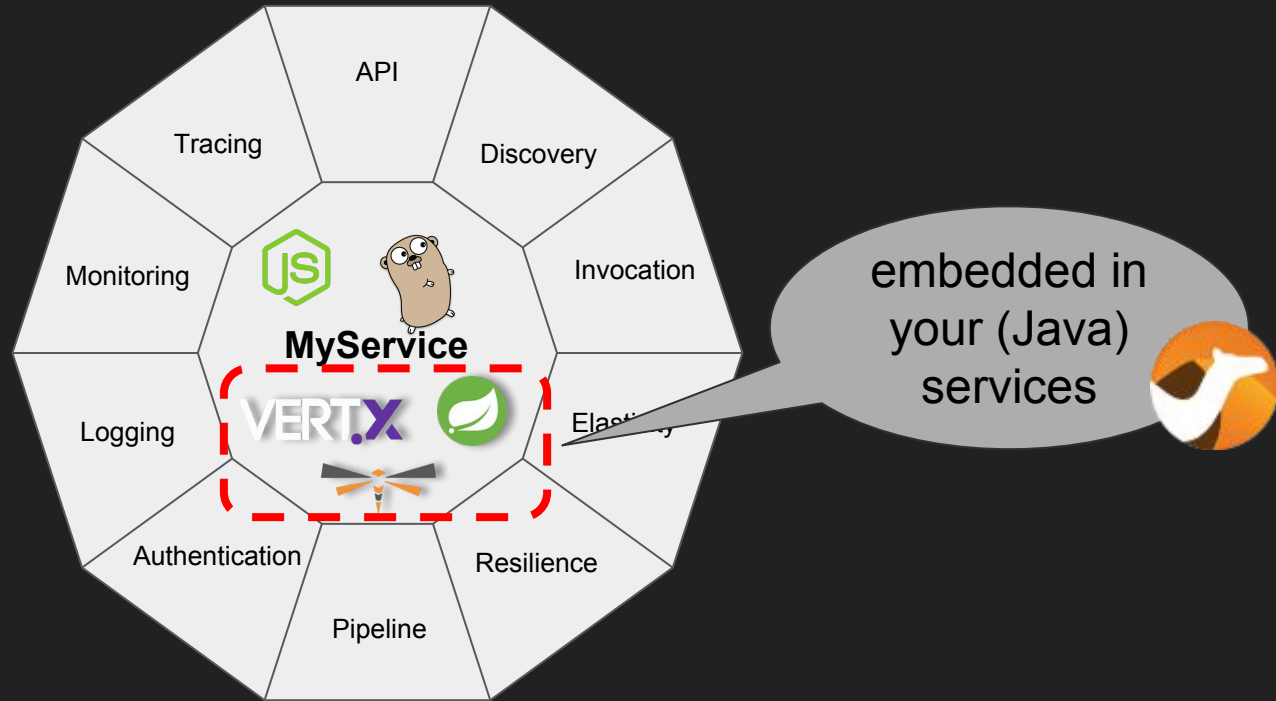
Microservices'ilities + API management



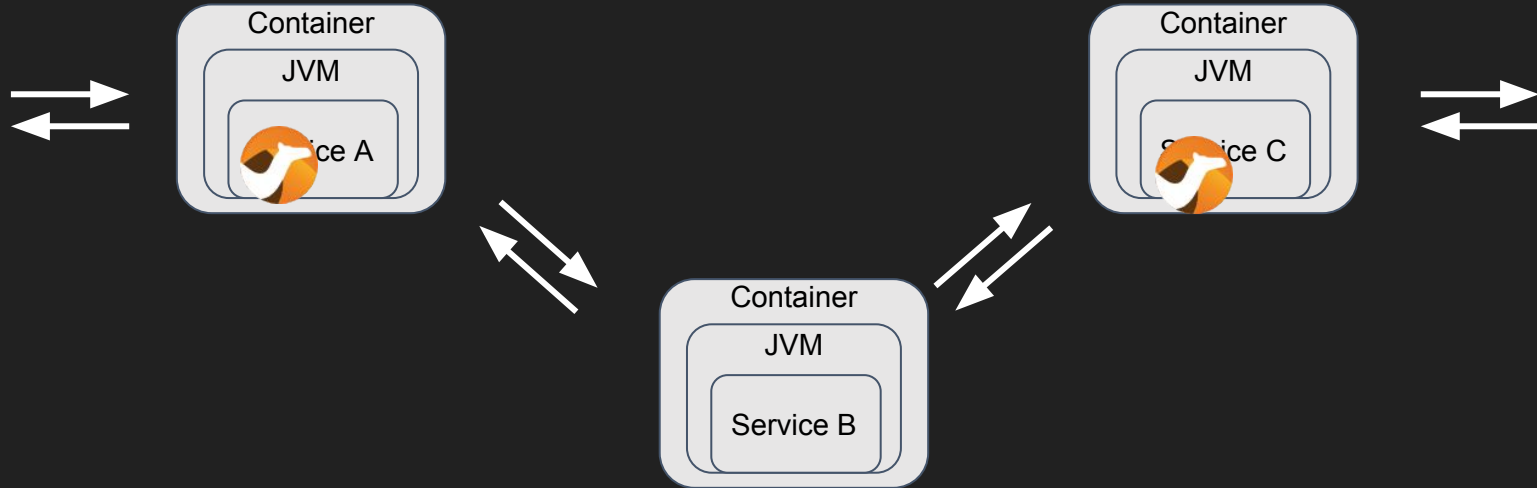
But where is Camel?



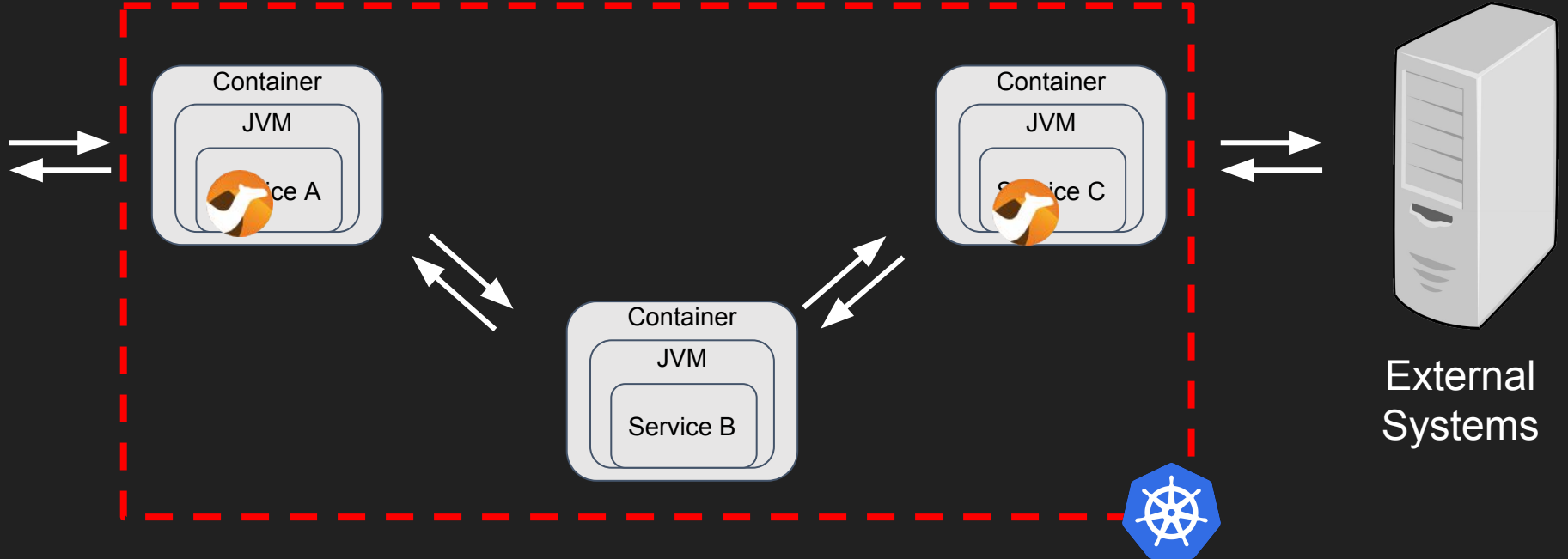
But where is Camel?



Microservices == Distributed Integration



Microservices == Distributed Integration



THE THREE PILLARS OF AGILE INTEGRATION

Key foundational capabilities needed by today's enterprises

DISTRIBUTED INTEGRATION

- LIGHTWEIGHT
- PATTERN BASED
- EVENT ORIENTED
- COMMUNITY SOURCED

FLEXIBILITY



CONTAINERS

- CLOUD NATIVE SOLUTIONS
- LEARN ARTIFACTS, INDIVIDUALLY DEPLOYABLE
- CONTAINER BASED SCALING AND HIGH AVAILABILITY

SCALABILITY



APIs

- WELL DEFINED, REUSABLE, AND WELL MANAGED END-POINTS
- ECOSYSTEM LEVERAGE

RE-USABILITY



Camel in the Cloud



Best Practice - Small in Size

- Camel is light-weight
 - (camel-core 4mb)
 - + what you need

- Single fat-jar via:



Best Practice - Stateless

- Favour stateless applications
- If state is needed:
 - Data-grid
 - camel-infinispan
 - camel-hazelcast
 - camel-ignite
 - ...
 - Storage
 - camel-sql
 - camel-jpa
 - camel-kafka
 - ...
 - Kubernetes
 - Stateful-set

Best Practice - Configuration Management

- Kubernetes ConfigMap
 - Inject via ENV
 - Inject via files
- Kubernetes Secrets
 - inject via ENV
 - Inject via files



```
// inject configuration via spring-style @Value  
@Value("${fallback}")  
private String fallback;
```



```
.simple( text: "{{fallback}}")
```

```
$ kubectl get cm -o yaml my-configmap  
apiVersion: v1  
data:  
  fallback: I still got no response  
kind: ConfigMap
```

Best Practice - Fault Tolerant

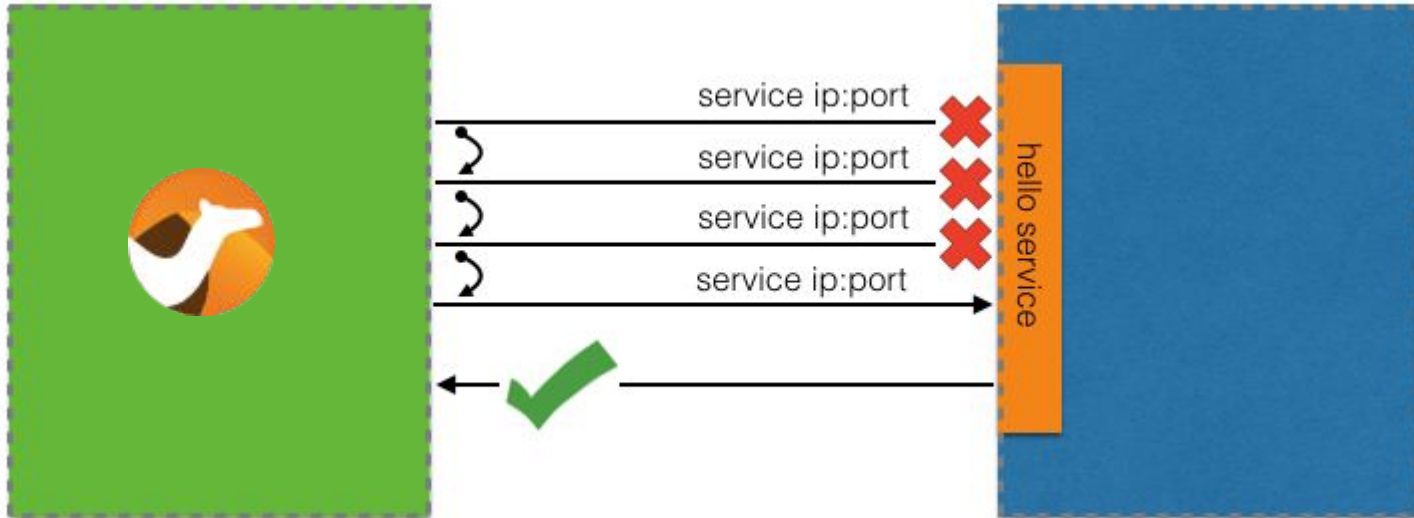
- Camel Retry
 - onException
 - errorHandler
- Camel Hystrix
 - circuit breaker



Best Practice - Fault Tolerant

- Camel Retry
 - onException
 - errorHandler

```
onException(Exception.class)  
    .maximumRedeliveries(10)  
    .redeliveryDelay(1000);
```



Best Practice - Fault Tolerant

- Camel Retry
 - onException
 - errorHandler

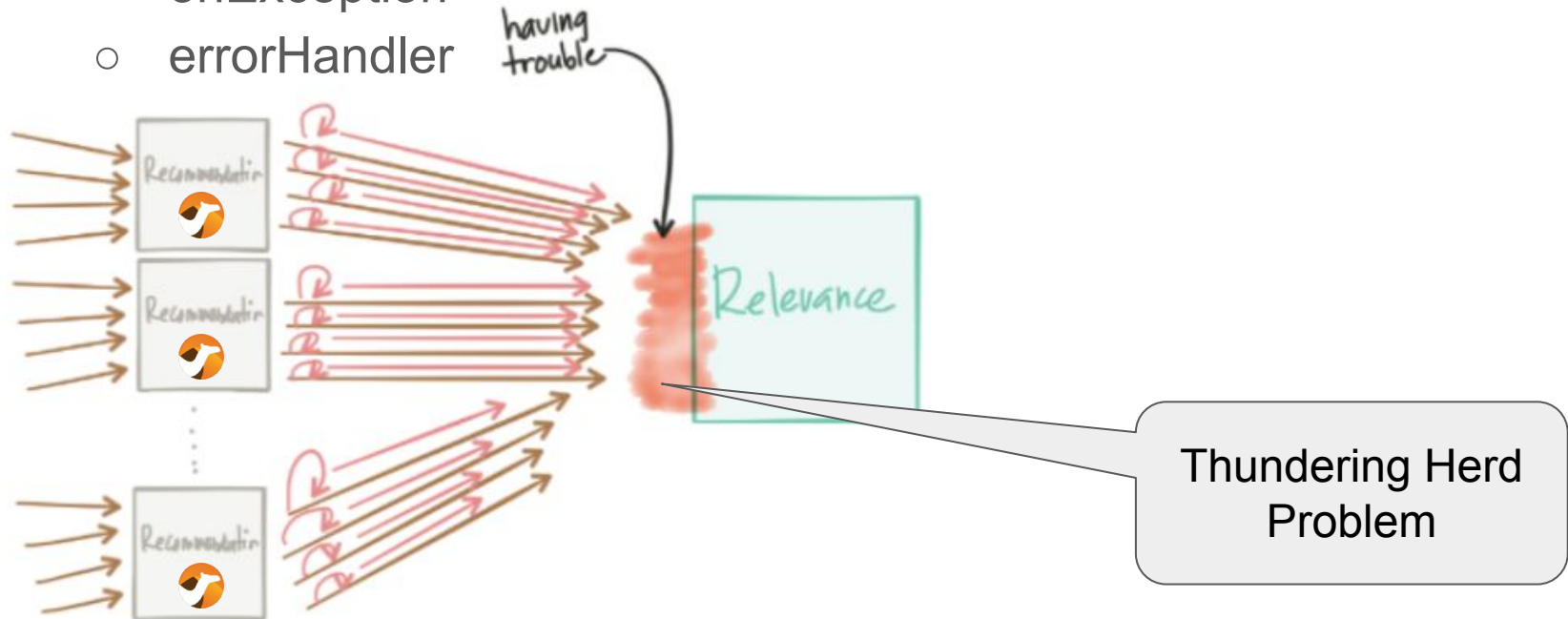


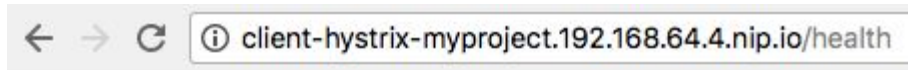
Figure by Christian Posta

Best Practice - Health Checks

- Health Checks
 - camel-spring-boot actuator
 - wildfly-swarm monitor

- Readiness Probe
 - Kubernetes

- Liveness Probe
 - Kubernetes

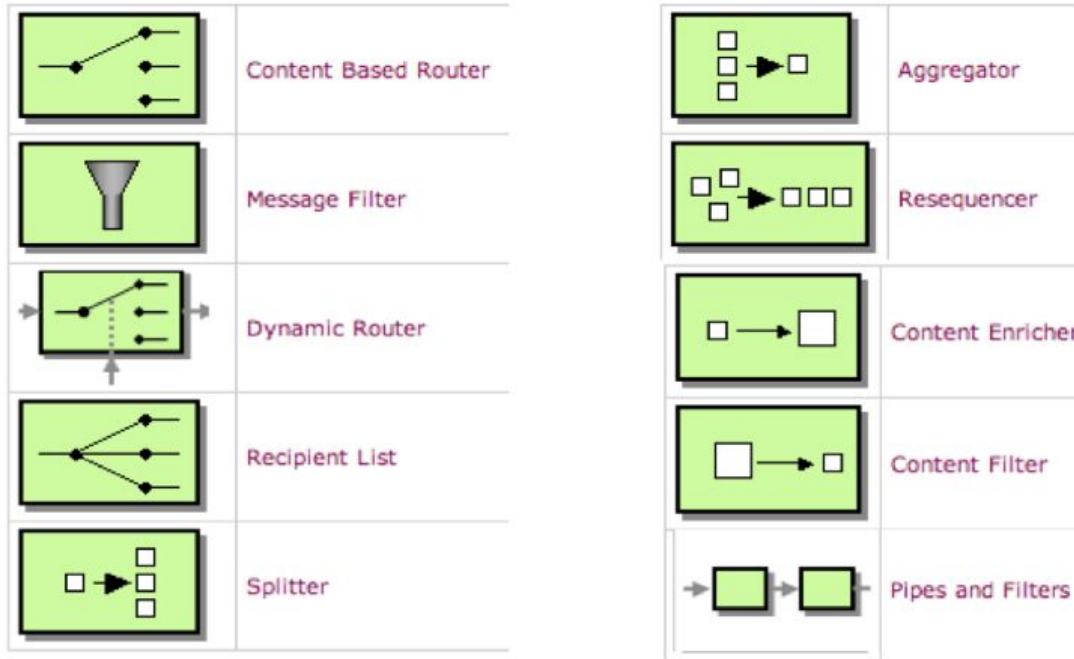


```
{
  status: "UP",
  - camel: {
    status: "UP",
    name: "camel-1",
    version: "2.20.2",
    contextStatus: "Started",
  },
  - camel-health-checks: {
    status: "UP",
    route:routel: "UP",
  },
  - diskSpace: {
    status: "UP",
    total: 19195224064,
    free: 5747757056,
    threshold: 10485760,
  },
}
```

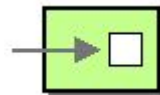


Best Practice - EIP Patterns

- Works anywhere



EIP Cloud Patterns



Service Call

Plugins

- Consul
- Etcd
- Kubernetes
- Ribbon
- Zookeeper

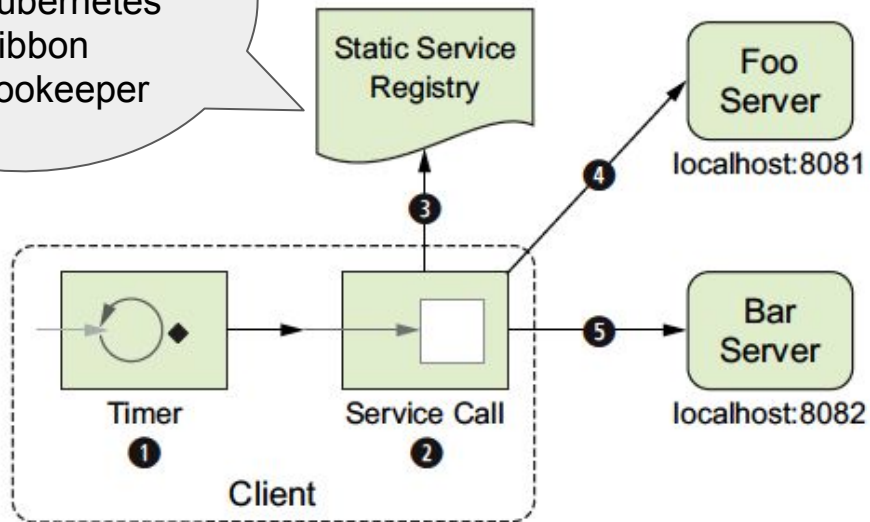
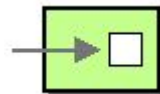


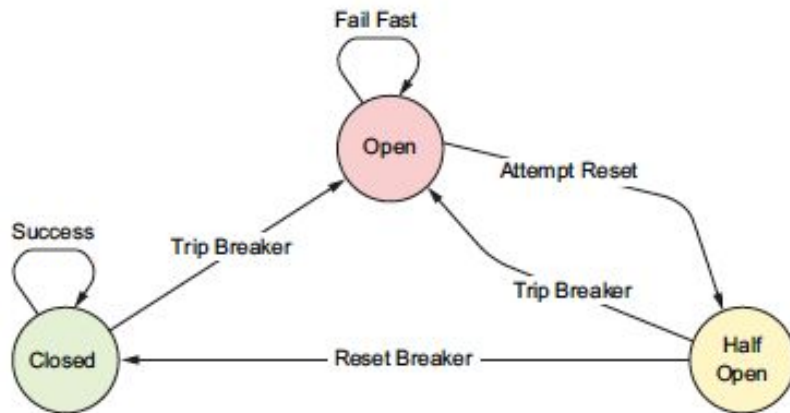
Figure 17.9 A timer ① triggers the Service Call EIP ② to call a clustered service. The physical locations of the service are looked up in the service registry ③. The service is then called in a round-robin fashion by calling either Foo server ④ or Bar server ⑤.

```
from("timer")  
    .serviceCall("hello-service");
```

EIP Cloud Patterns



Hystrix EIP



```
from("timer:foo")
  .hystrix()
    .to("http:myservice")
  .onFallback()
    .to("bean:myfallback")
  .end()
```

EIP Cloud Patterns

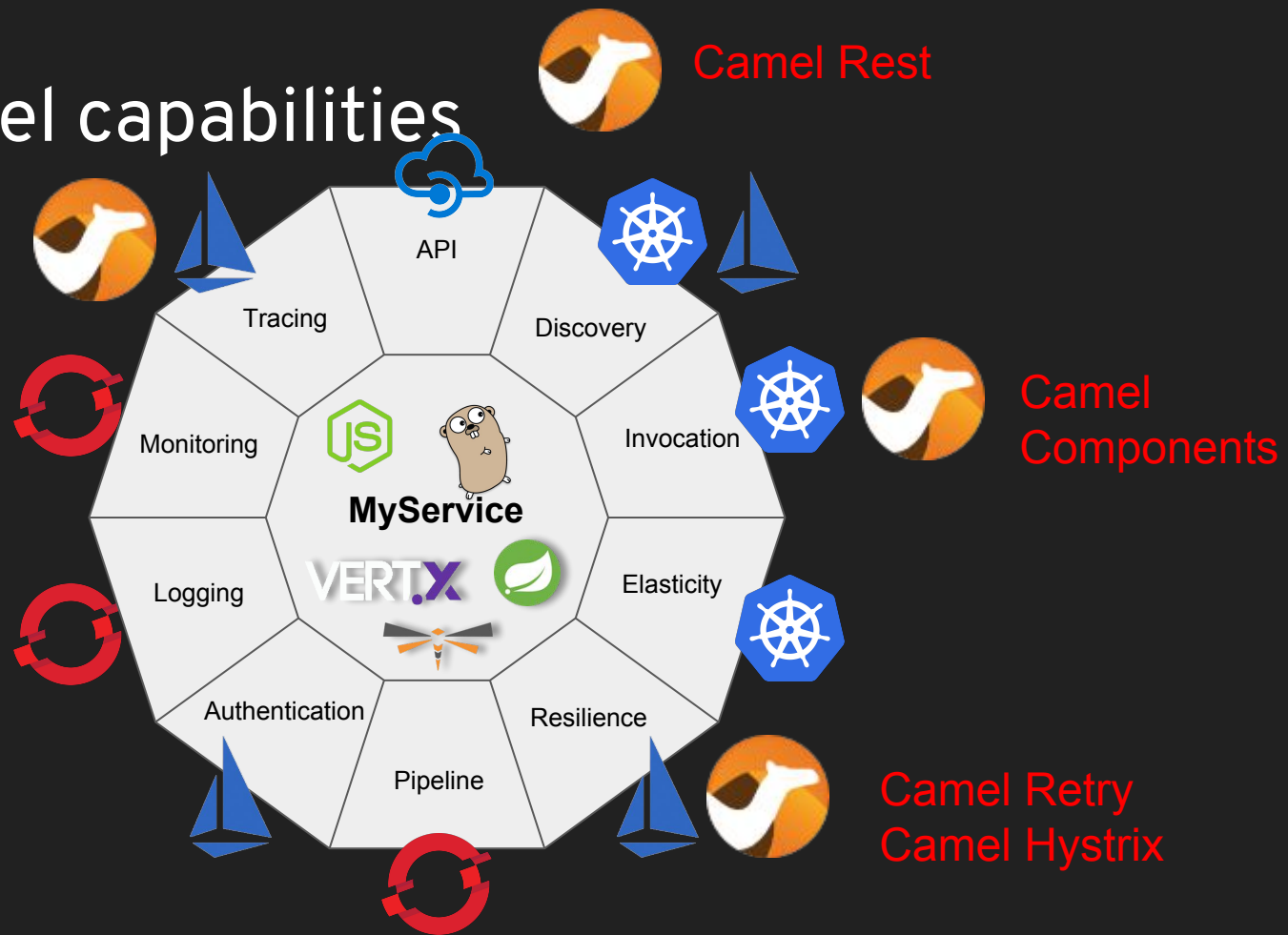


Distributed
Tracing

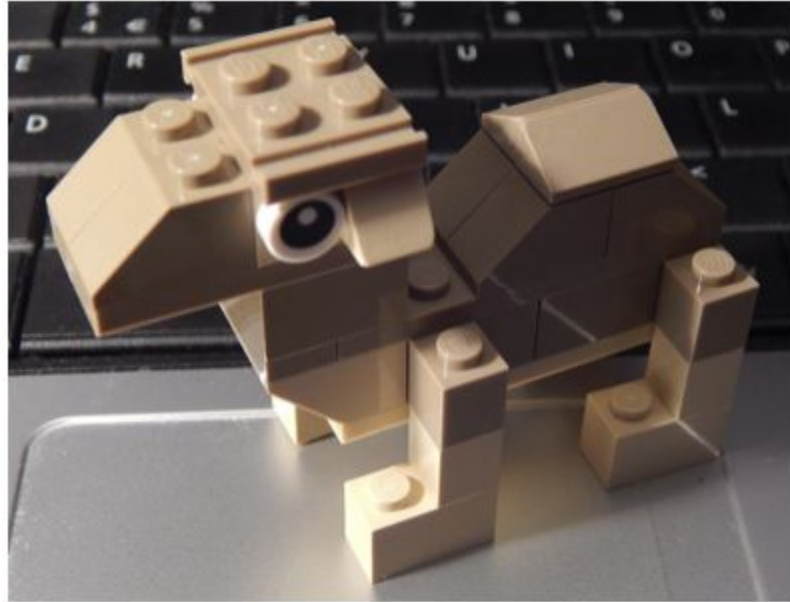


Usable Camel capabilities

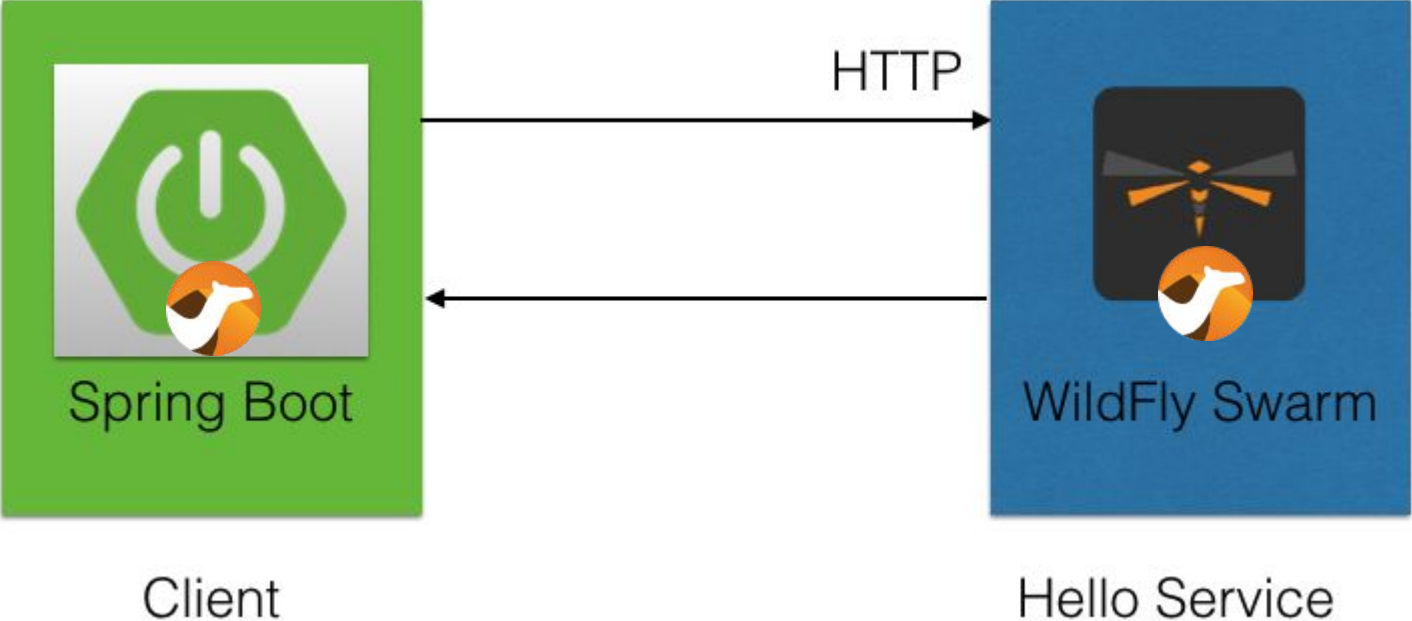
Camel Zipkin
Camel OpenTracing



Demo Time



Basic Demo

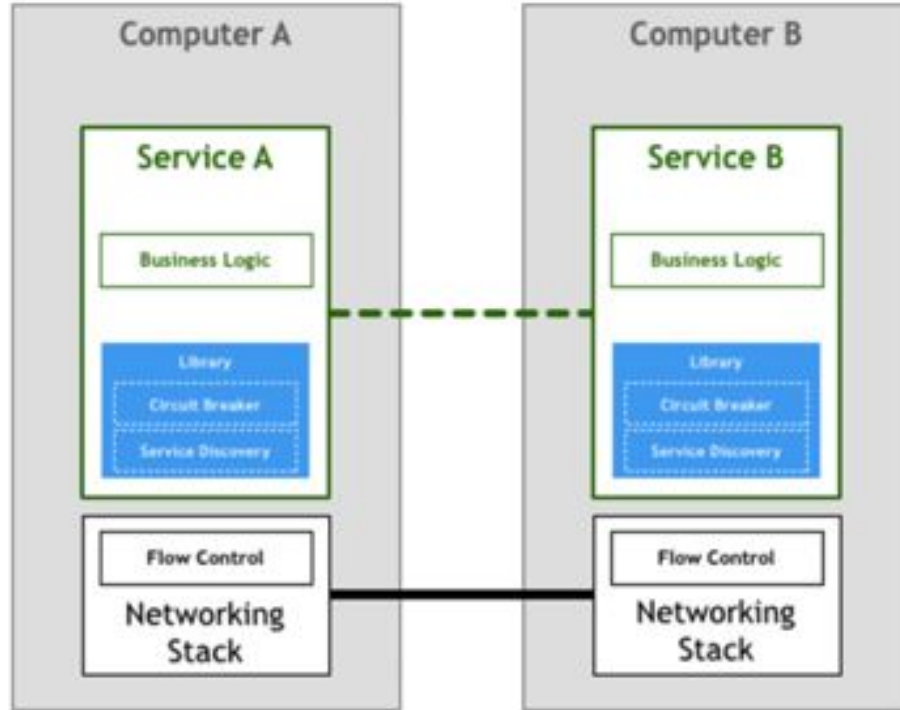


Tip of the iceberg

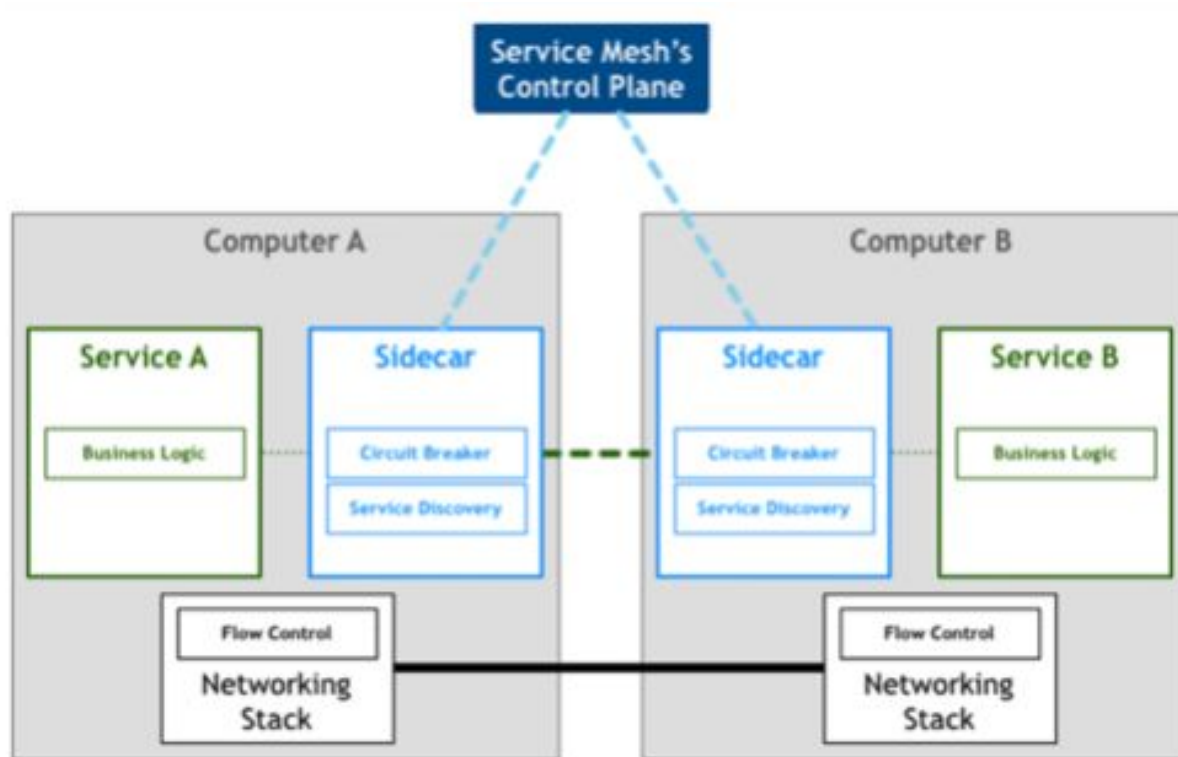


Figure by Bilgin Ibryam

Service Mesh



Service Mesh

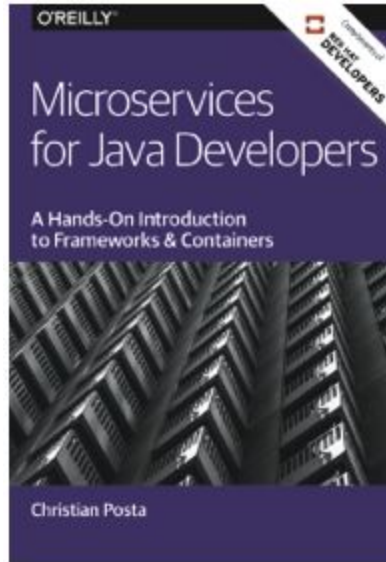


Service Mesh Webinar



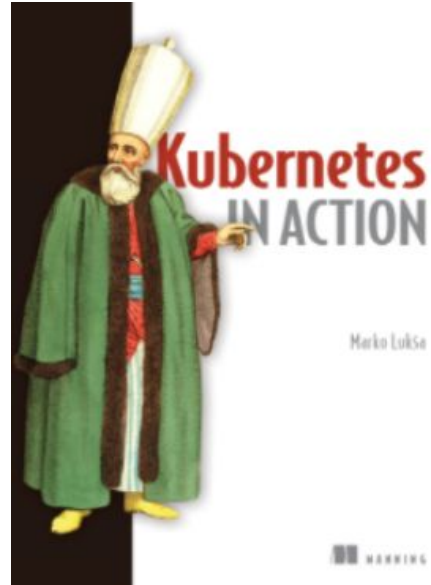
<https://developers.redhat.com/video/youtube/YQLOcjvbo9s>

Free book



<http://developers.redhat.com/promotions/microservices-for-java-developers>

Not so free book



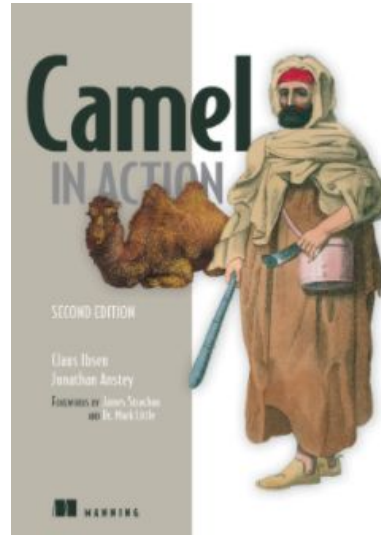
<https://www.manning.com/books/kubernetes-in-action>

Not so free book

- Discount code (39%):

came139

(ordering from Manning)



<https://www.manning.com/books/camel-in-action-second-edition>

More Information

- Slides and Demo source code:
<https://github.com/davsclaus/camel-riders-in-the-cloud/tree/moscow>
- Apache Camel website:
<http://camel.apache.org>
- Best "What is Apache Camel" article:
<https://dzone.com/articles/open-source-integration-apache>
- My blog:
<http://www.davsclaus.com>
- DevNation Webinars:
<https://developers.redhat.com/devnationlive>

Q & A