

SpringOne Platform by Pivotal.

Google Cloud Workshop: Serverless with Spring Cloud, riff, and Knative

Ray Tsang
Developer Advocate, GCP
@saturnism

Thomas Risberg
Staff Engineer, Pivotal
@trisberg



Ray Tsang

Developer Advocate
Google Cloud Platform

Java Champion

Contributing to: Spring Cloud GCP,
JHipster, JDeferred, ...

@saturnism saturnism.me

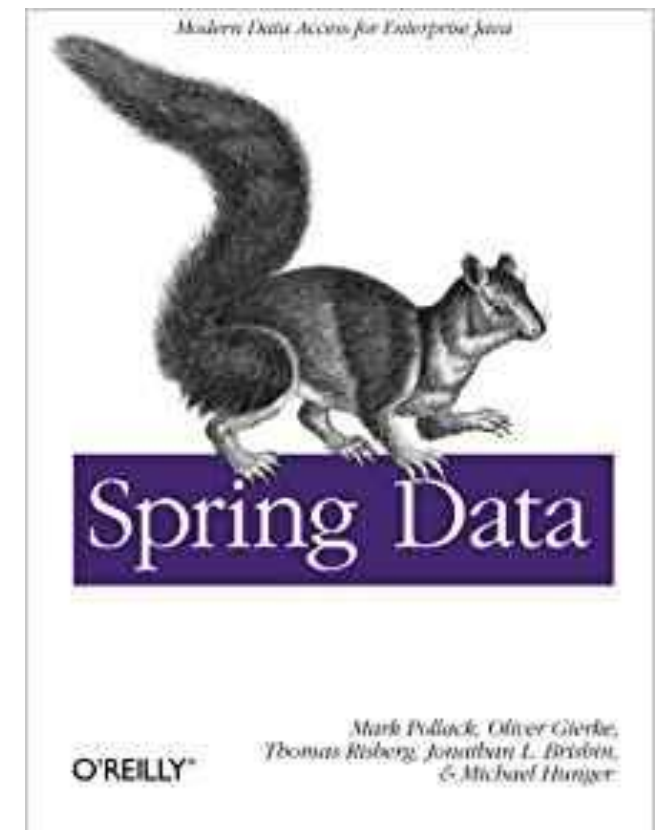
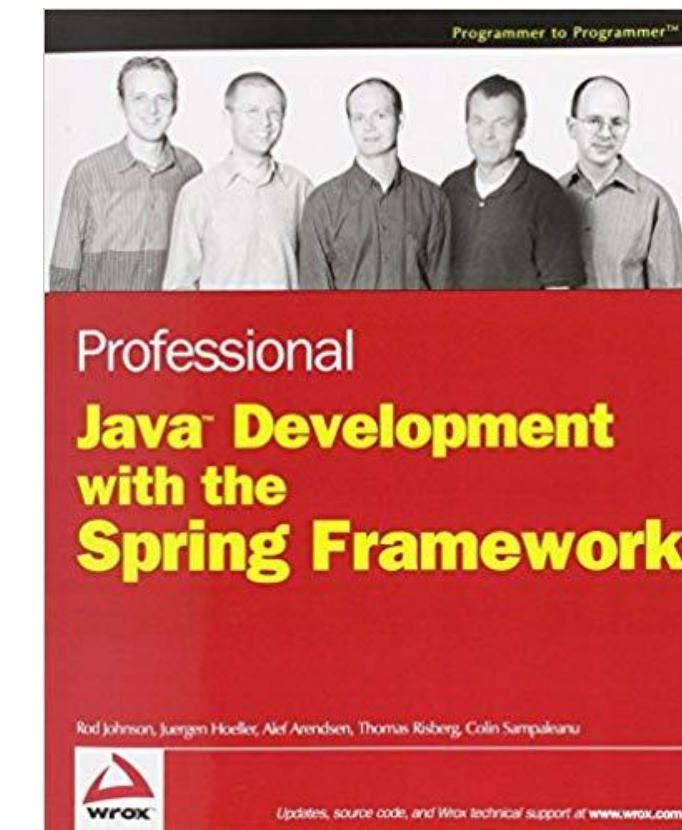


Thomas Risberg

Staff Engineer
Pivotal

Contributing to: riff, Knative, Spring Cloud,
Spring Data, and Spring Framework

@trisberg



Workshop Overview

The primary technologies we will use:

- Kubernetes / Google Kubernetes Engine (GKE)
- Spring Cloud Function
- riff
- Knative

In addition, nice Google Cloud Platform tools and starters:

- Jib - Build container images for Java applications without Docker
- Spring Cloud GCP - Idiomatic starters to use Google Cloud Platform services
- Vision API, Cloud Pub/Sub, Cloud SQL, ...

Open Up The Material

PLEASE DO NOT JUMP AHEAD FOR INITIAL SETUP

bit.ly/spring-riff-lab

Lab 1 - Initial Setup - Follow Me!

Lab 2 - Bootstrap GKE, riff, Knative, Istio, ...

Introduction to Google Kubernetes Engine

- Fully managed Kubernetes cluster on Google Cloud Platform
- Integrated with Stackdriver Logging, Monitoring, and more
- Free Master Node! Fully managed and has SLA. Free :)
- Multi-zone deployment
- Node Autoscaling
- Node Auto-repair
- Managed master and node upgrades (automatic if needed)
- Schedule maintenance window (beta)

Lab 3 - Simple Function

Working with Spring Cloud Function

Serverless Functions the Spring way

- Promote the implementation of business logic via functions
- Abstract away the target runtime
- Support a uniform programming model across serverless providers
 - run standalone (locally or in a PaaS).
- Enable Spring Boot features on serverless providers
 - auto-configuration
 - dependency injection
 - metrics

Our first function

```
package functions;

import java.util.function.Function;

public class Upper implements Function<String, String> {

    public String apply(String name) {
        return name.toUpperCase();
    }
}
```

A serverless function by itself is ...

kind of pointless ... we need a way to deploy it and to invoke it

- include the function in an app
- invoke it via a standalone invoker app
- deploy it to a FaaS

A function invoker

```
package com.springdeveloper.demo.functioninvoker;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.cloud.function.deployer.EnableFunctionDeployer;

@SpringBootApplication
@EnableFunctionDeployer
public class FunctionInvokerApplication {

    public static void main(String[] args) {
        SpringApplication.run(FunctionInvokerApplication.class, args);
    }
}
```

A function app

```
package com.springdeveloper.demo.functionapp;

import java.util.function.Function;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.context.annotation.Bean;

@SpringBootApplication
public class FunctionAppApplication {

    @Bean
    public Function<String, String> uppercase() {
        return s -> s.toUpperCase();
    }

    public static void main(String[] args) {
        SpringApplication.run(FunctionAppApplication.class, args);
    }
}
```

Deploy it to a FaaS

```
riff function create java upper \  
--namespace demo \  
--git-repo https://github.com/trisberg/upper.git \  
--image springdeveloper/upper \  
--verbose
```

Lab 4 - Riff to Knative

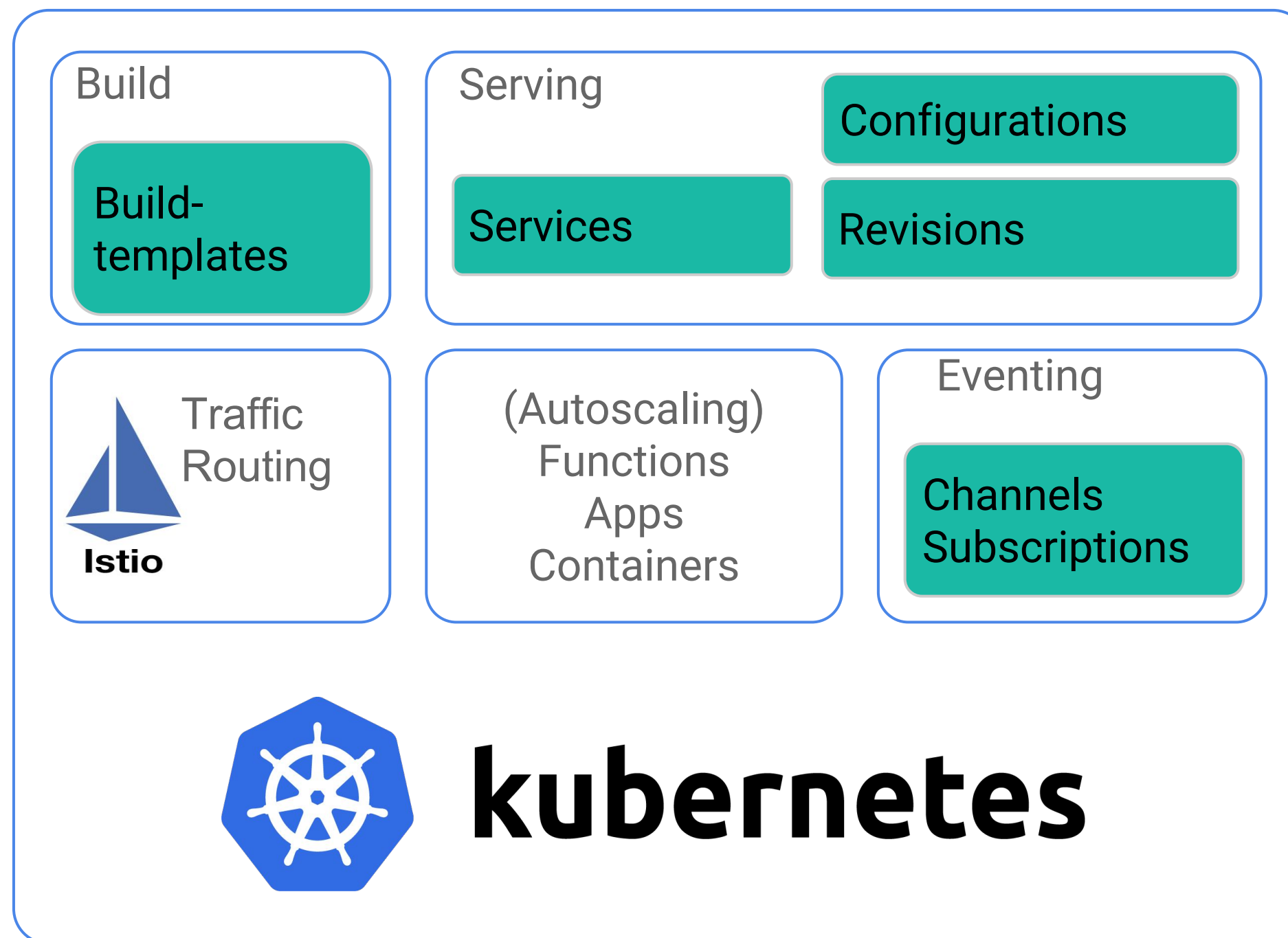
What is Knative?

Knative (pronounced kay-nay-tiv) extends Kubernetes to provide a set of middleware components that are essential to build modern, source-centric, and container-based applications that can run anywhere: on premises, in the cloud, or even in a third-party data center.

(from <https://github.com/knative/docs>)

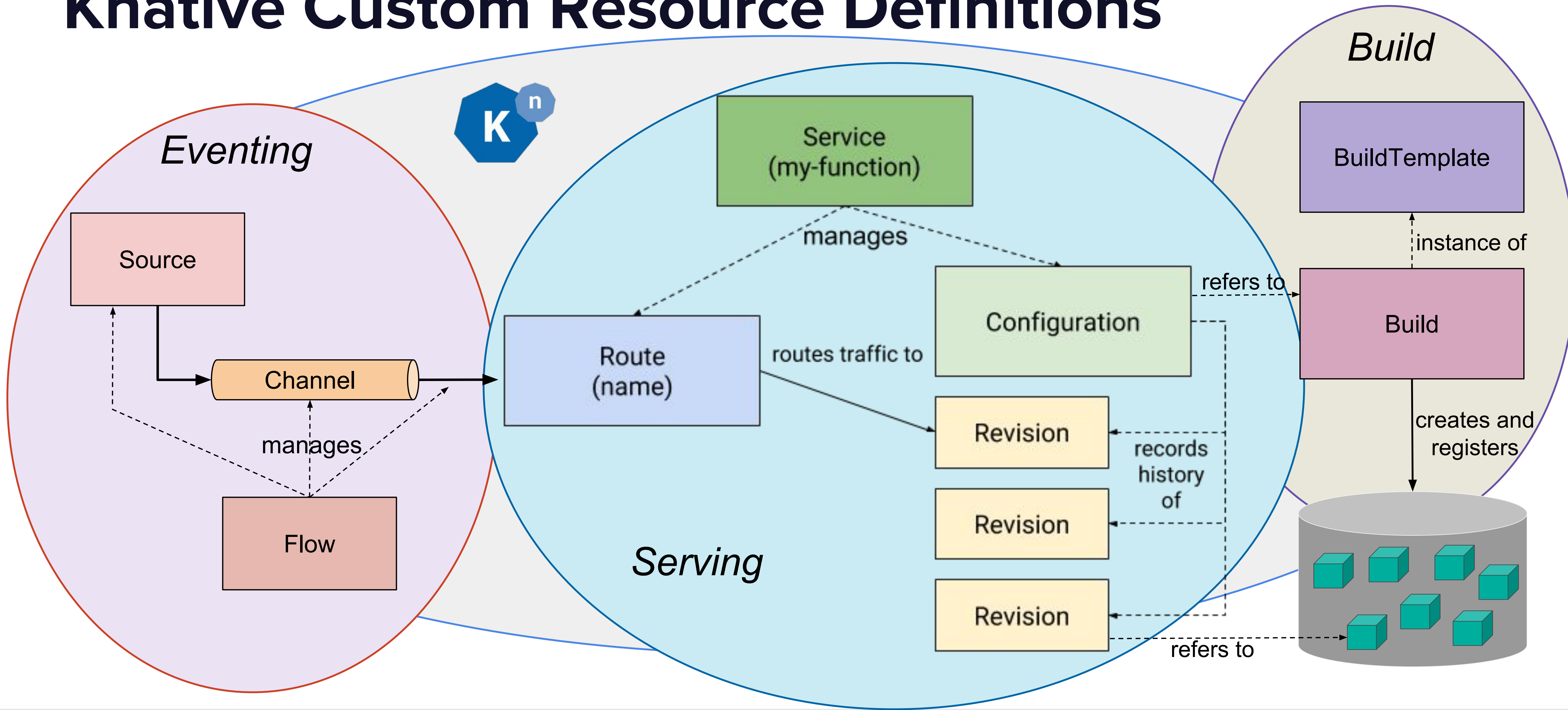
Overview of Knative

- A new OSS project initiated by Google
- Provides **build** for cloud builds, **serving** managing deployments and scaling, **eventing** for functions that handle events



<https://github.com/knative>

Knative Custom Resource Definitions



Run app/function on Knative Serving

```
apiVersion: serving.knative.dev/v1alpha1
kind: Service
metadata:
  name: upper
  namespace: default
spec:
  runLatest:
    configuration:
      revisionTemplate:
        spec:
          container:
            image: springdeveloper/upper:0.0.1
```


Overview of riff

riff provides developers with a service for executing functions in response to events.

Features:

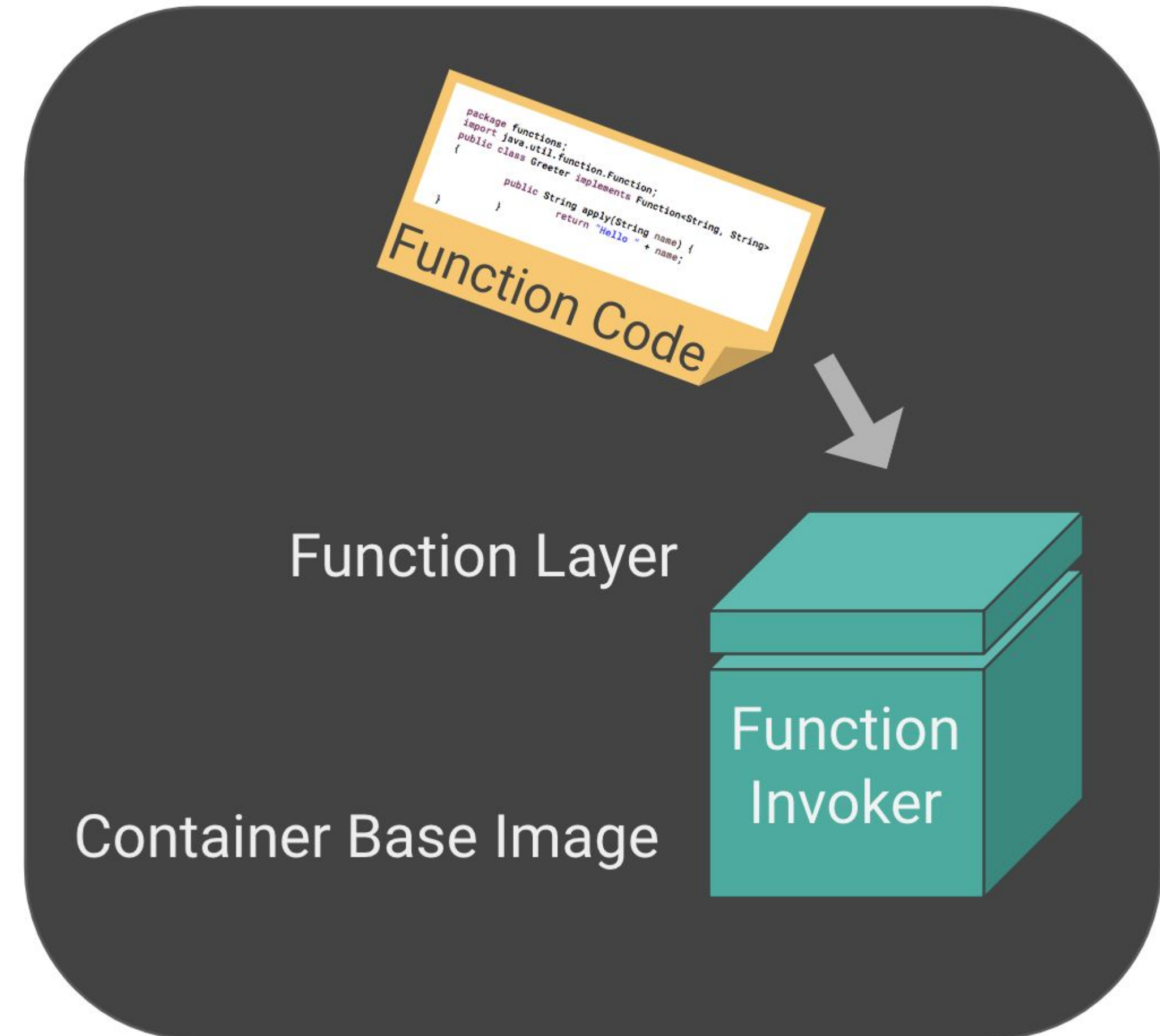
- Kubernetes-native -> now built on Knative
- polyglot
 - available: Java, node.js, shell commands
 - coming soon: python, ruby, go
- event streaming



Function packaging with riff

functions are packaged as containers

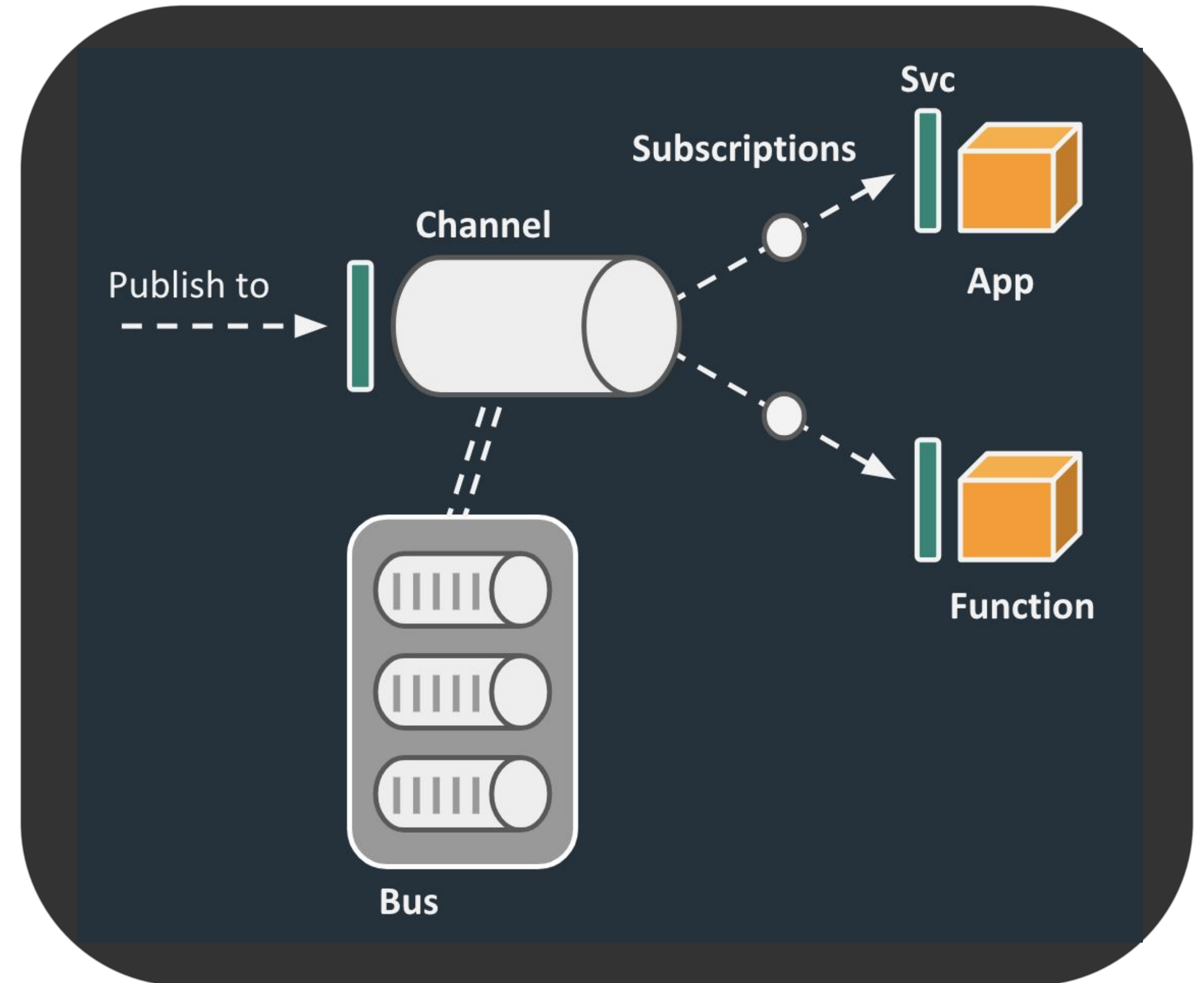
- Developers are responsible only for the business logic.
- Dependencies in base image layers can be managed independently.
 - simpler
 - more secure
 - more efficient



Functions communicate via messaging

functions connect to topics

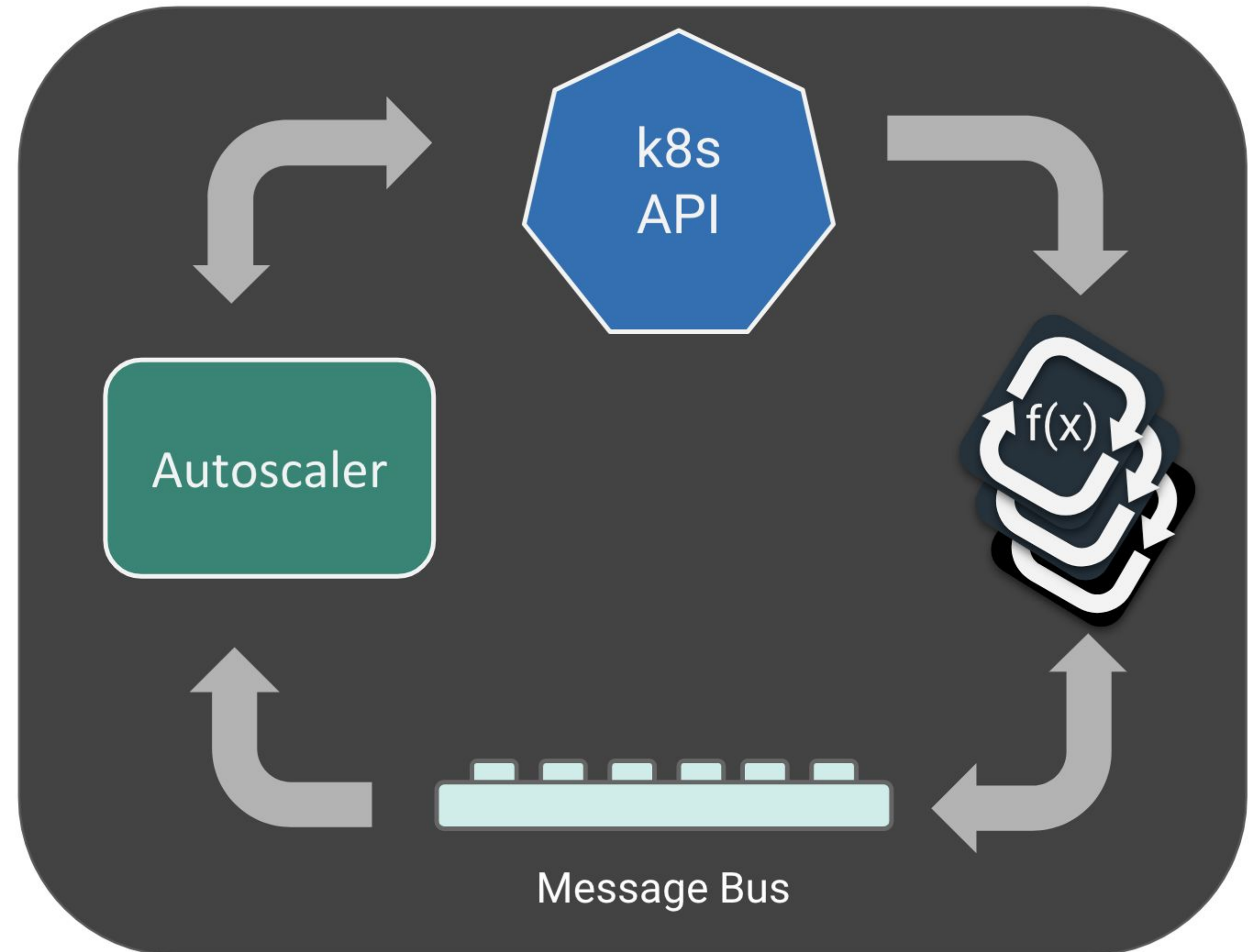
- pub/sub topics provide loose coupling.
- Enable message consumers and producers to operate without prior knowledge of each other.
- Enable a rich set of patterns beyond request/reply E.g. reactive streams.



Scaling functions with riff

functions scale with events

- Autoscaler:
 - interacts with K8s API
 - scales functions 0-1 and 1-N
 - monitors traffic



A bit of Caution

- Knative and riff are alpha releases
- Knative design is evolving, especially the Eventing portions
- riff will adapt to these changes but there might be some CLI commands that will change



Image by: <https://www.vecteezy.com/members/designious>

Lab 5 - Cat Not Cat

Spring Cloud GCP

Goals

Provide integration between GCP services and Spring

Meet developers where they are

Cut down on boilerplate code



+



Google Cloud Platform

Spring Boot Starters for...

Feature	GCP Service	Spring Framework
Authentication	Auto credential discovery. Service Accounts.	
Configuration	Cloud Runtime Config	Spring Cloud Config
Messaging	Cloud Pub/Sub	Spring Integration Spring Cloud Stream Spring Dataflow
Database	Cloud SQL Cloud Spanner	Spring Data Spring Data Spanner
Storage	Cloud Storage	Spring Resource
Logging	Stackdriver Logging	Logback
Trace	Stackdriver Trace	Spring Cloud Sleuth

Cloud Vision API

- Analyze what's inside of an image
- Feature detection such as
 - labels (is there a cat?)
 - text extraction
 - facial features and emotion
 - landmarks
 - logos
 - safe search
 - ...

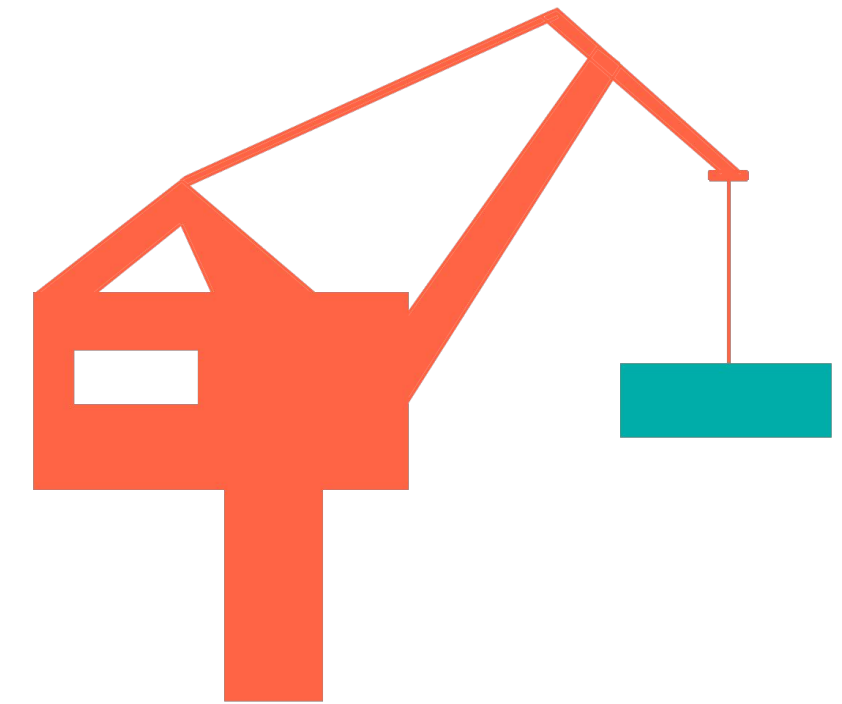


More Cloud ML APIs

- There are more ML APIs!
 - AutoML - Learn from your images
 - Video API - Extract information from video
 - Natural Language Processing - Parse sentence, understand sentiment
 - Speech to Text & Text to Speech
 - Translation
- BigQuery ML
 - ML within Data warehouse, with SQL query!

Jib

- Build container images without Docker
- Build directly to container image registry
 - e.g., DockerHub, Google Container Registry, ...
- Maven and Gradle plugin
- Smart layering for faster rebuilds and pulls
- Fast and repeatable builds
- Uses Distroless image by default, can use your own base image



Lab 6 - Event Bus

Event Bus

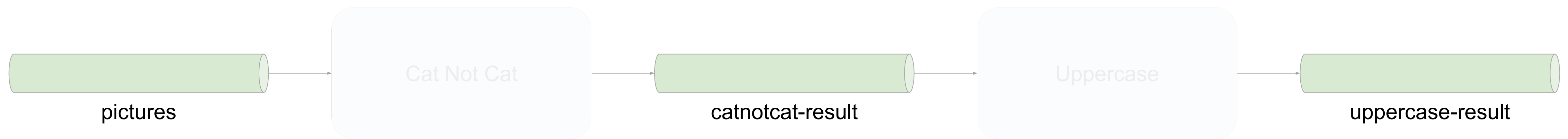
- Stub (In Memory)
 - No Persistence
 - No redelivery
- Kafka Bus
- Cloud Pub/Sub Bus

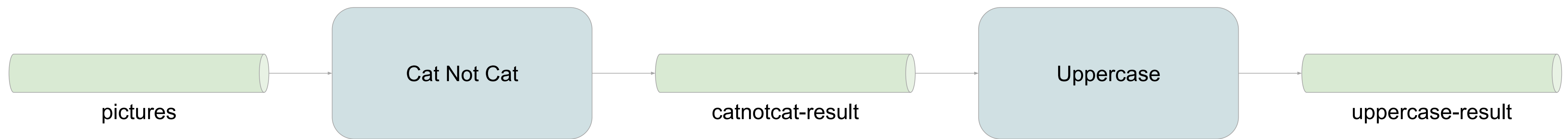
Cloud Pub/Sub

- Global messaging middleware
- Fully managed, highly available
- At-least once delivery
- One to Many
- Topic
- Subscription

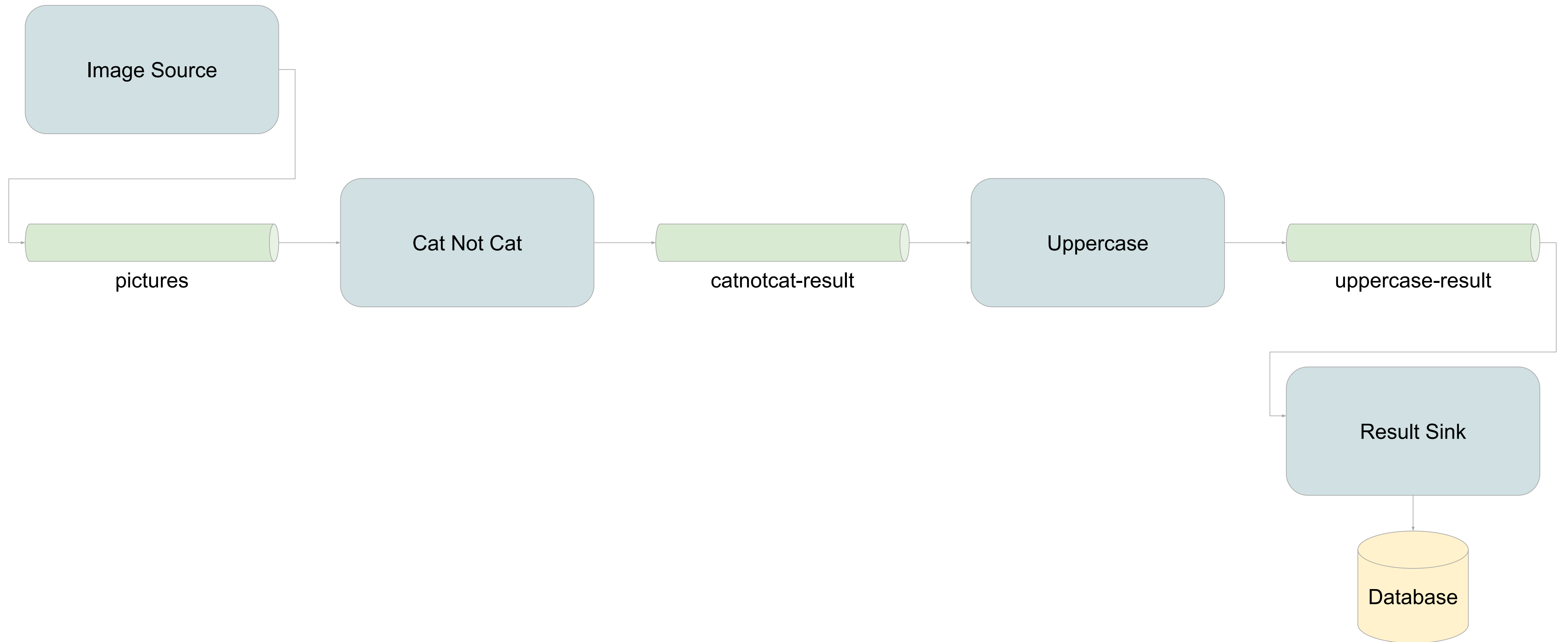


Lab 7 - Uppercat Lite





Lab 8 - Uppercat End to End



Cloud Spanner

- Horizontally scalable, strongly consistent, globally distributed, relational, database!
- Automatic, synchronous replication for high availability
- SQL Queries
- 99.999% SLA for global configuration



Lab 9 - Custom Domain Name

Lab 10 - Autoscaling

SpringOne Platform by Pivotal.

> Stay Connected.

Related Sessions

"Introducing Knative", Mark Fisher - Wednesday 2:00 pm - 3:10 pm

"Knative and riff for Spring Developers", Dave Syer - Wednesday 3:20 pm - 3:50 pm

"riffing on Knative", Scott Andrews & Eric Bottard - Thursday 11:50 am - 12:20 pm

"Google Cloud-Native Architecture with Spring Cloud GCP", Ray Tsang - Thursday 10:30 am - 11:40 am

"Spring, Functions, Serverless and You", Nate Schutta - Wednesday 4:20 pm - 5:30 pm



#springone