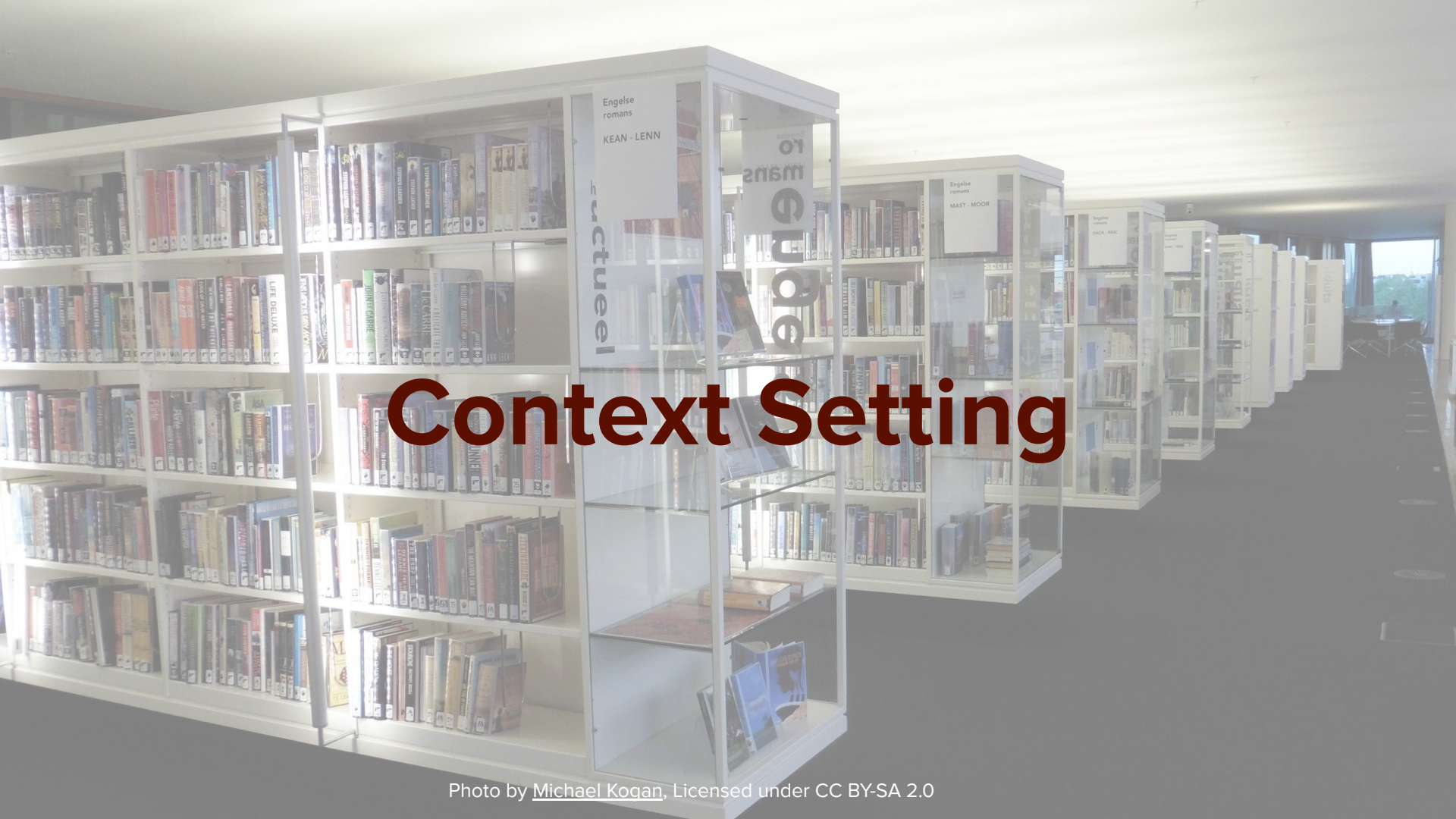
Knative

Bringing Serverless to K8s

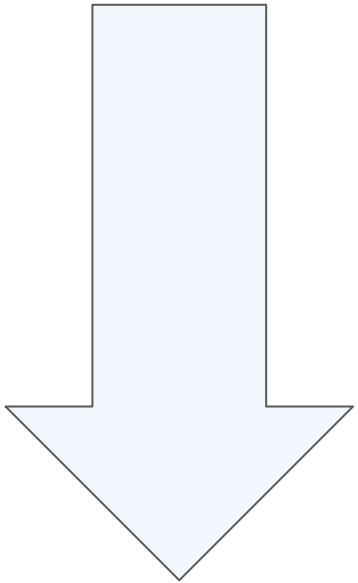
Andrew Ripka
@rippmn



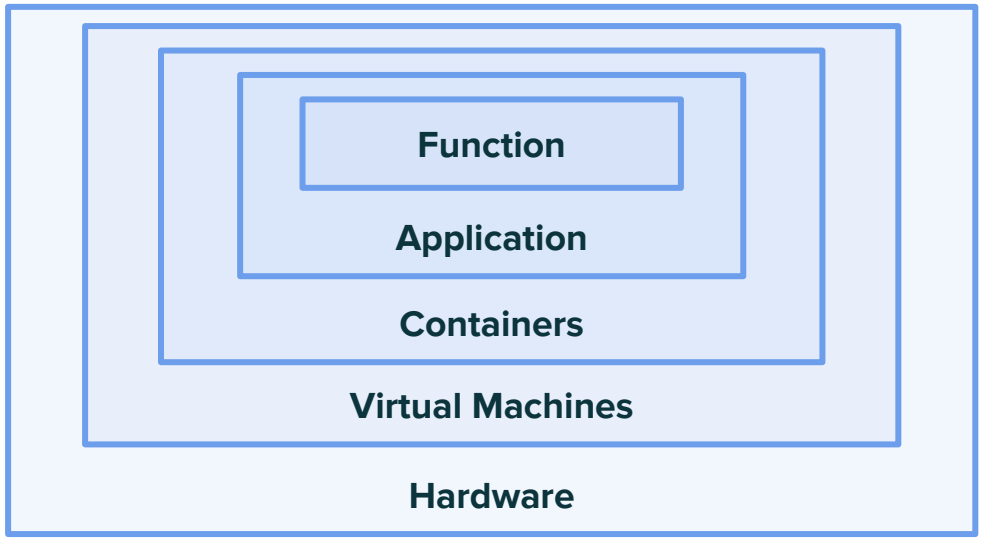
Context Setting

Platform Developer Abstractions

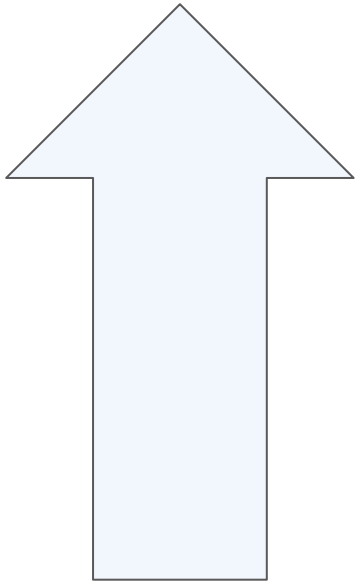
More Standards



More Flexibility



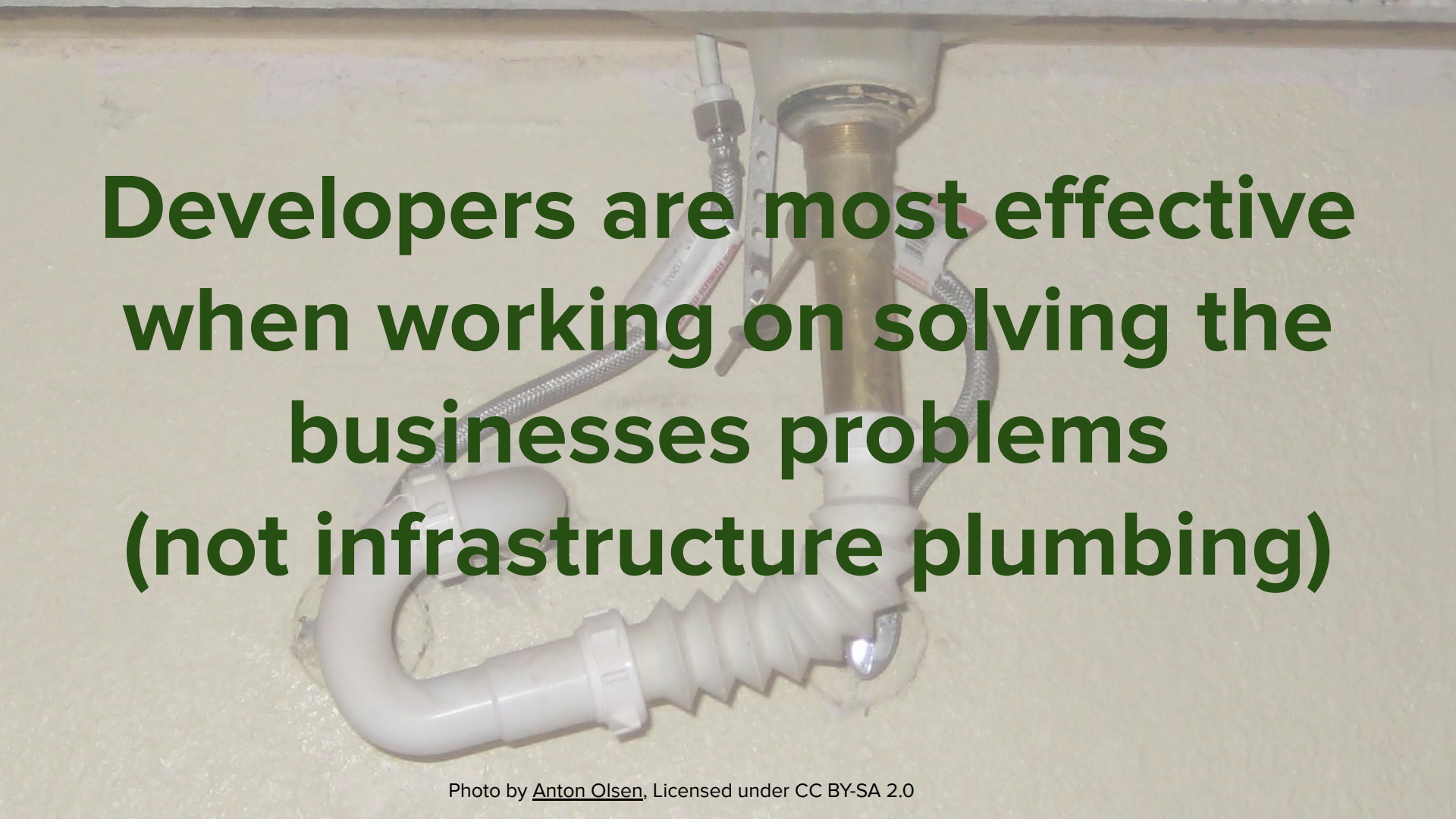
Higher Efficiency



Lower Efficiency

What is Serverless?

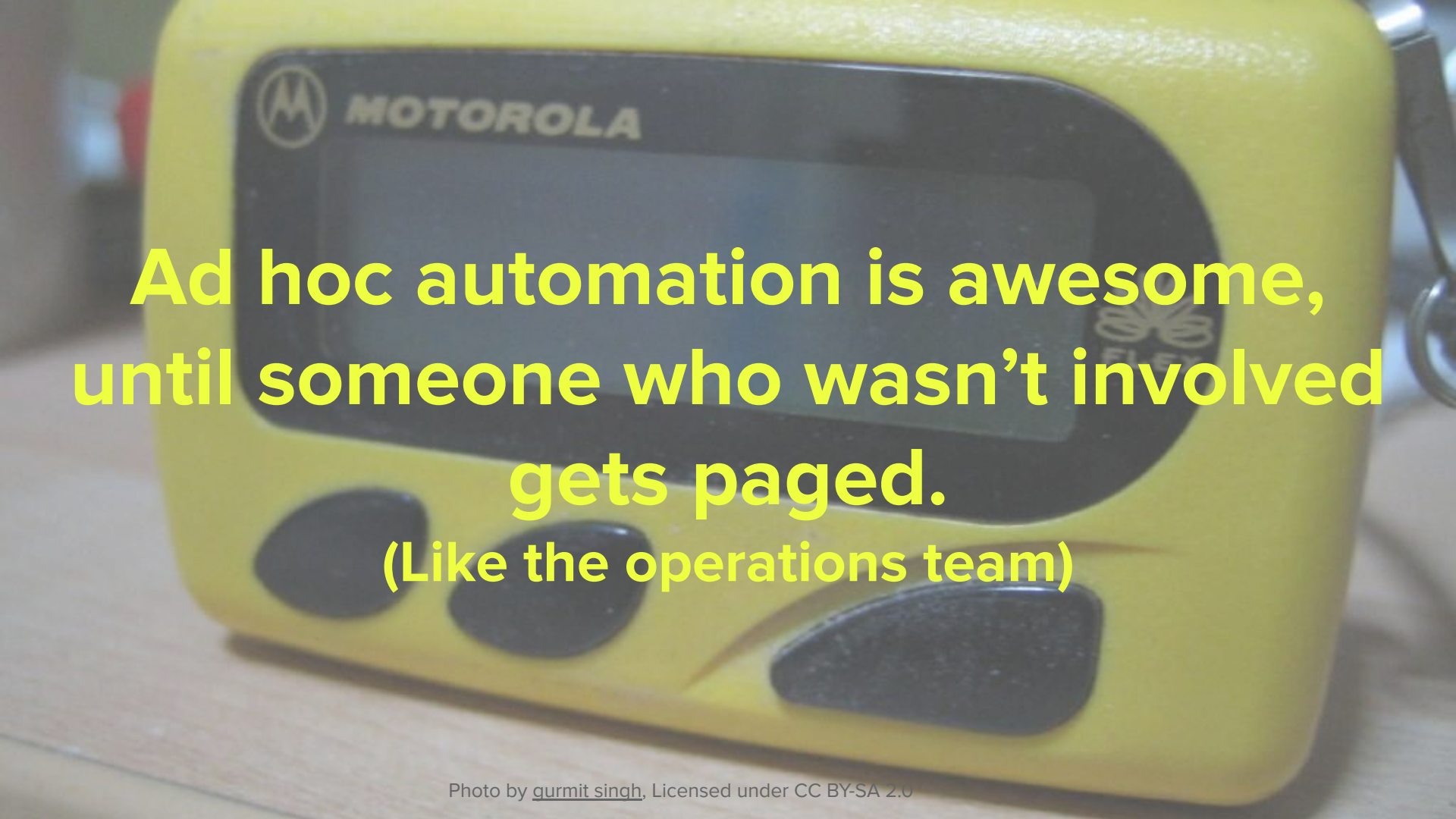
- Frees devs from details that aren't part of the core business logic of their code. (e.g. Infrastructure)
- Small single purpose functions that are invoked in response to events
- Scale to 0 when no events occurring



Developers are most effective when working on solving the businesses problems (not infrastructure plumbing)

If current platform isn't easy enough developers will build one themselves

With tools and constraints they chose...



Ad hoc automation is awesome, until someone who wasn't involved gets paged.

(Like the operations team)

So you say you want a serverless platform

and you want to do it yourself...

Why not Kubernetes?

It’s building blocks

And an infrastructure abstraction

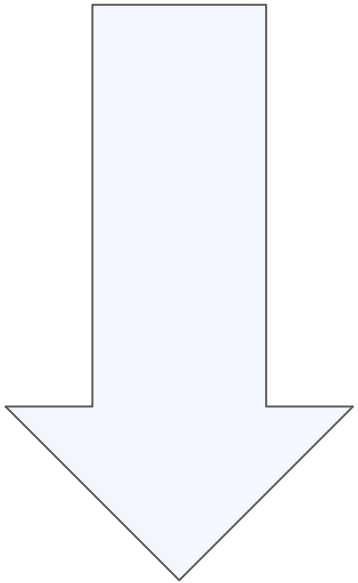
Dan Woods
@danveloper

3. You are building an application platform. A lot of people treat Kubernetes _as_ the platform, but it provides the foundational models for infrastructure, not applications. Higher-order modeling is required to translate the infra domain to the app domain.

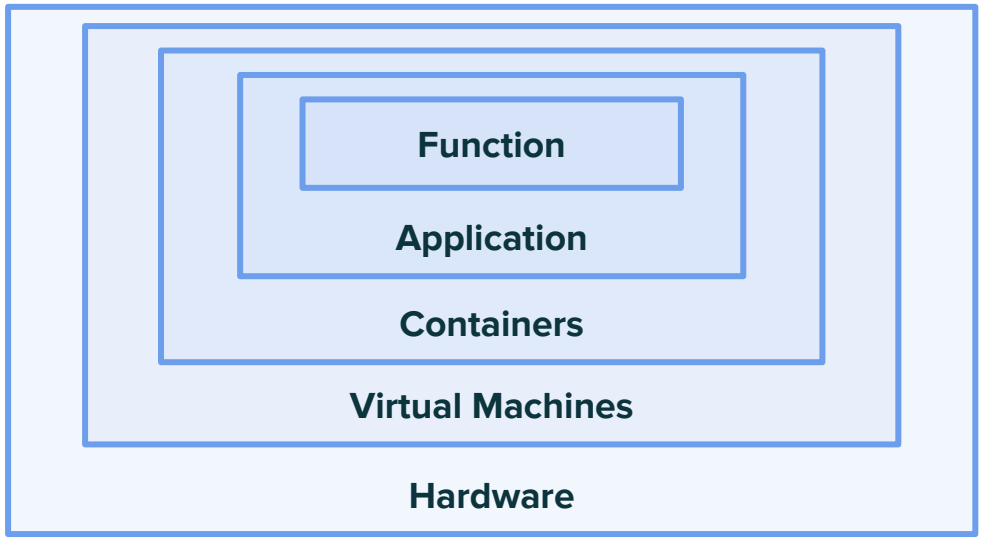
♡ 55 6:58 AM - Apr 19, 2019

Platform Developer Abstractions

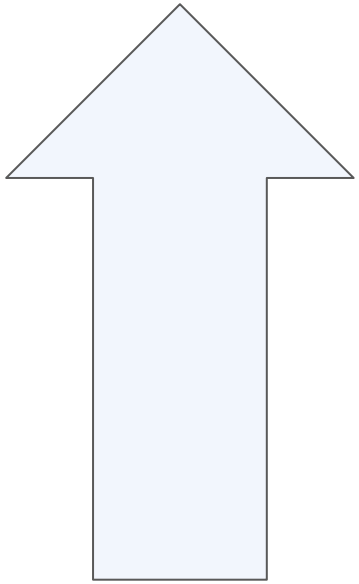
More Standards



More Flexibility



Higher Efficiency



Lower Efficiency

Coarse Grained Requirements

- Package to deployable
- Deploy and Run
- Map traffic and events to running handlers
- Conserve Resources

Finer Details

- Container creation and deployment automation
- Scale to zero and back
- Support revisions with traffic splitting and rollback
- Allows configurable event flows
- Gather metrics and telemetry from running deployments


K
n
Knative

Knative Codified Patterns

- Deploying a container
- Orchestrating source-to-URL workflows on Kubernetes
- Routing and managing traffic with blue/green deployment
- Automatically sizing workloads based on demand
- Binding running services to eventing ecosystems

Not a fully featured Function as a Service platform

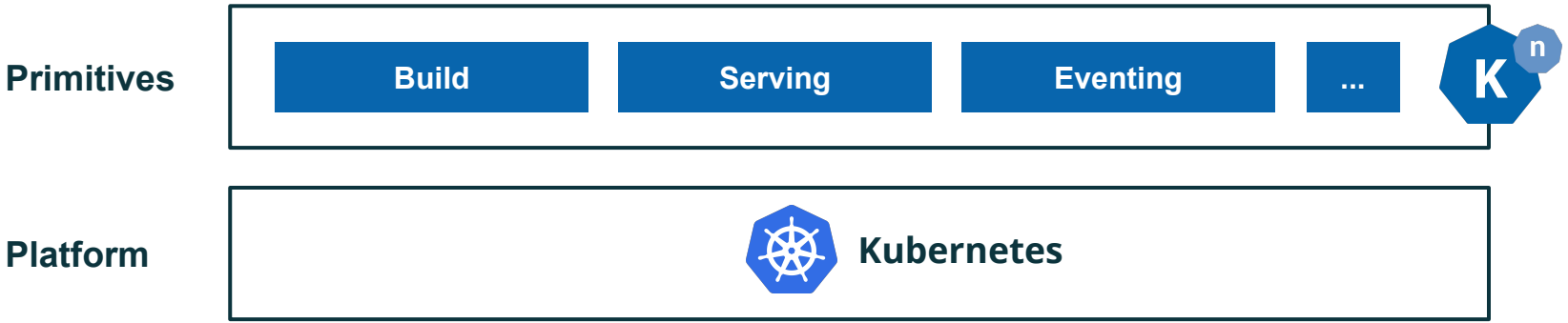
Intended to be used by to build FaaS or PaaS

Knative is...

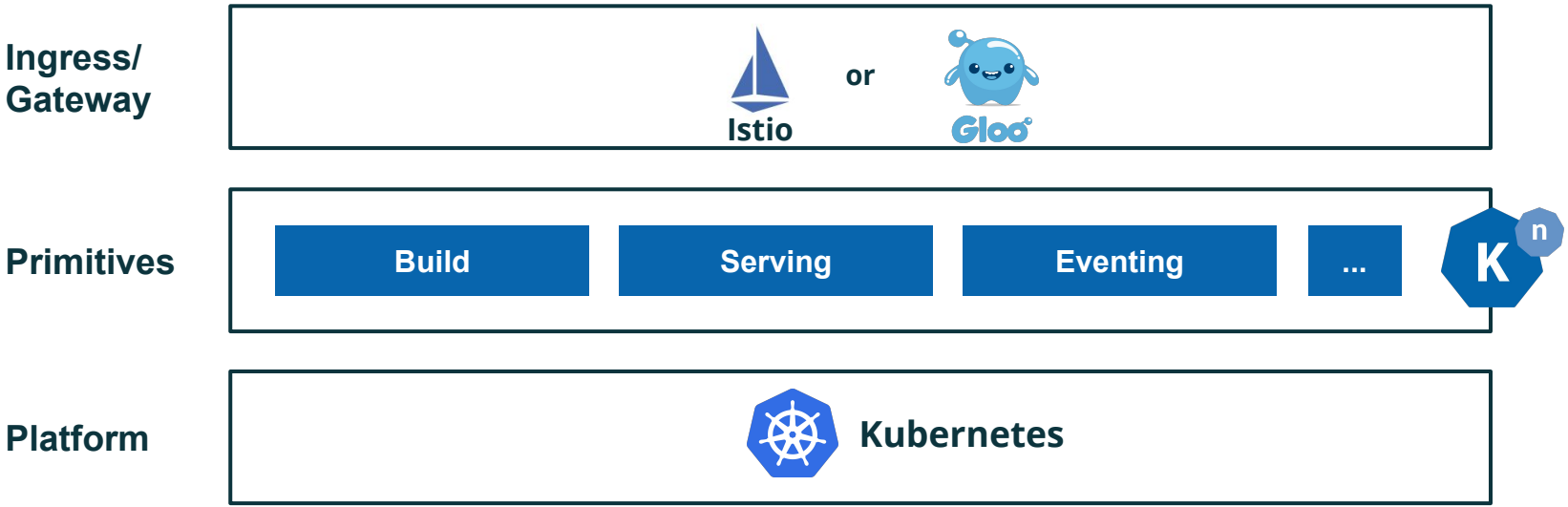
- An abstraction on top of Kubernetes
  - Deploys and runs containers
- Set of building blocks to construct your own FaaS/AaaS/CaaS
  - abstracts common tasks through custom Kubernetes API objects
- Middleware components essential to build modern, source-centric, and container-based applications that can run anywhere.

Knative architecture

Knative architecture



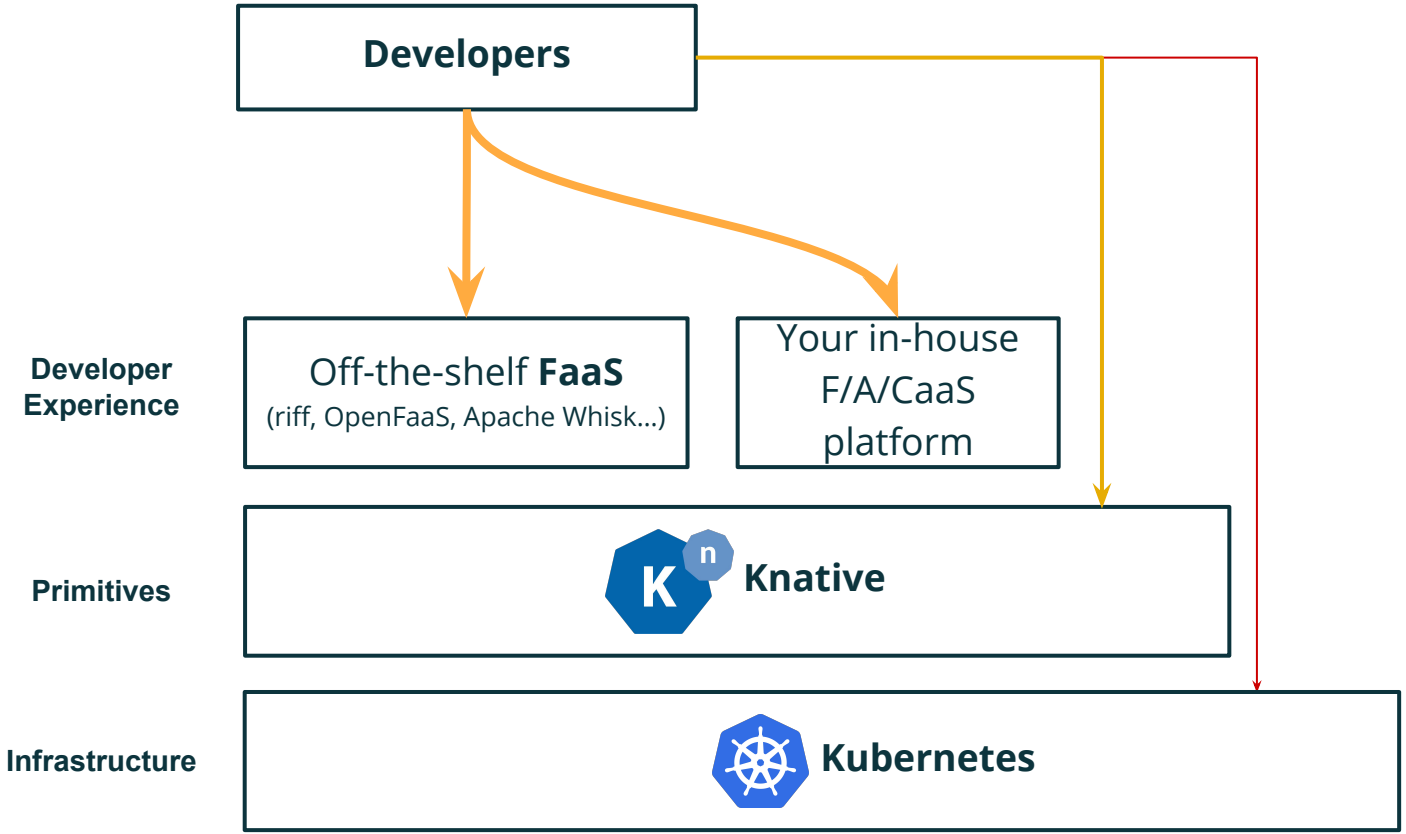
Knative architecture



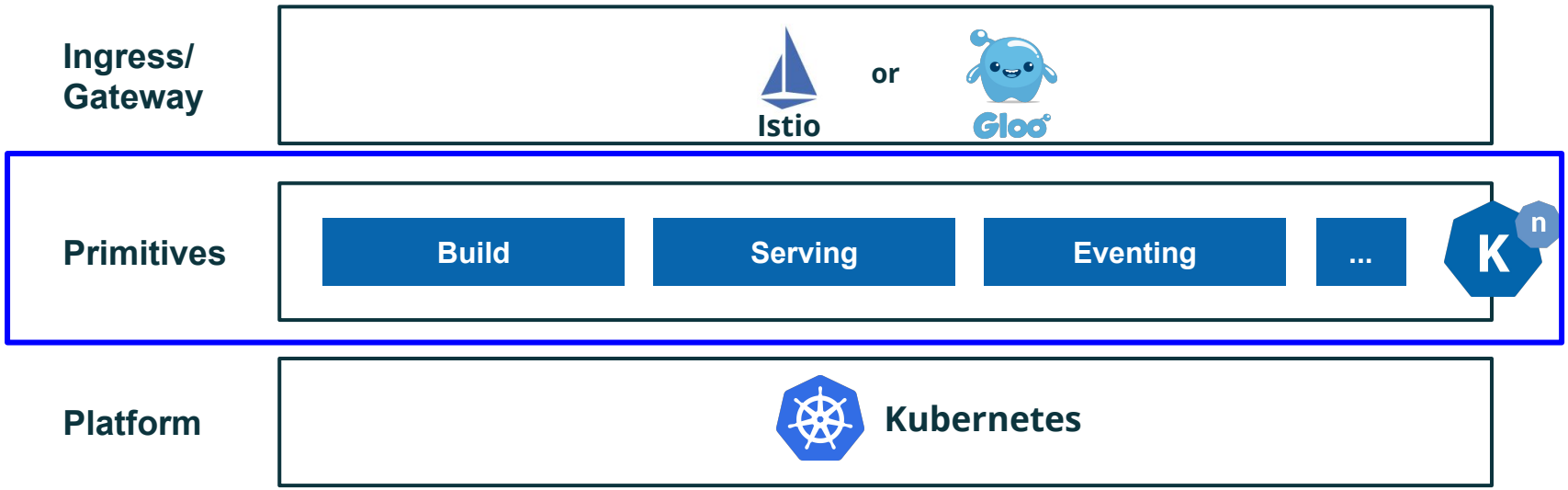
Audiences


Users and Systems (IoT) use applications developed and deployed by developers and hosted by operators
Istio
Knative
Kubernetes
Developers build and deploy apps
API
GitHub
Contributors develop and contribute code and docs to the OSS project
Operators deploy and manage Knative instances using Kubernetes API and tools
Platform Providers (such as Google Cloud Platform) provide underlying infrastructure

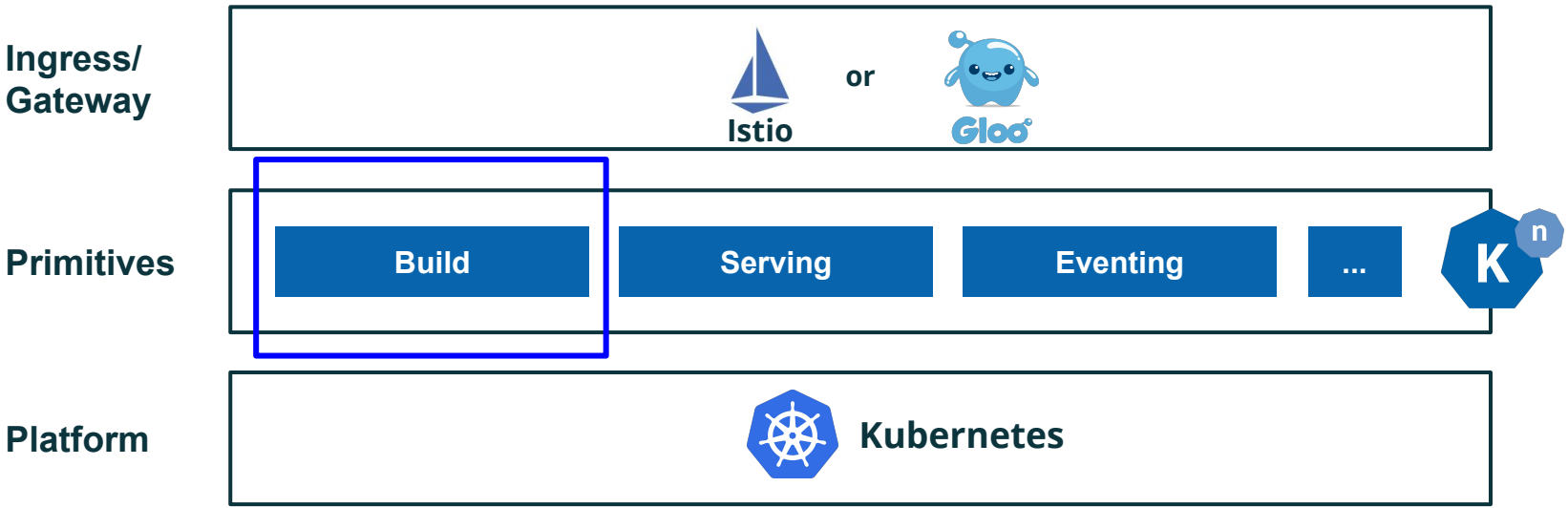
Developer Abstractions

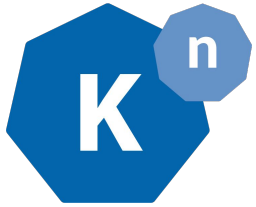


Knative Components



Knative Components





Build

Source-to-container build orchestration

Build

- Defines process that runs to completion and provide status
- Create container image from supplied source for deployment to Knative Serving
- Connects to secured resources via Kubernetes Service Account

Build Components

- Build is made up of Steps containing builders
- Builder - container image that accomplishes a task (discrete or composite)
  - One of the steps typically push to container registry

Build Components (Cont.)

- Source - data necessary for build in mounted volume
  - GCS
  - Git
  - Custom (container image)

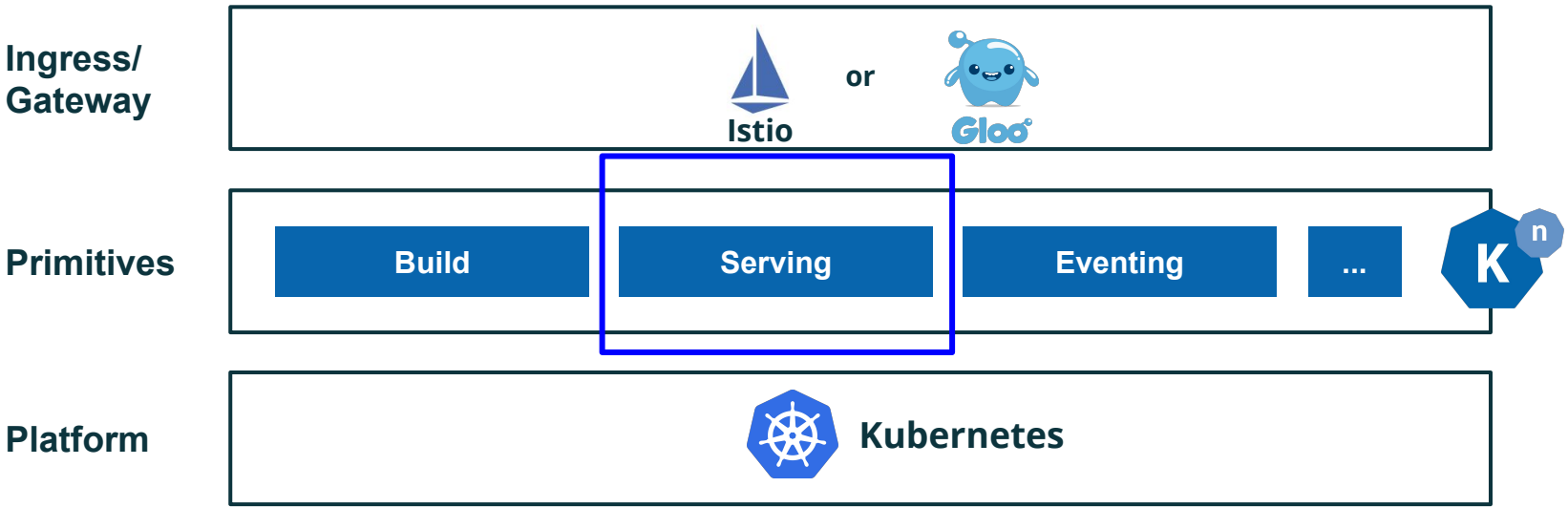
Build Components (Cont.)

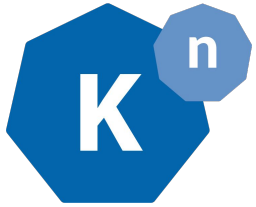
- Service Account - privileges to run build
- Volumes - non default volumes to use during steps
- Timeout - max time for resource allocation and steps

Build Templates

- Encapsulate build process for reuse and consistency
  - limited parameterization
- Similar to “compile” phase of Heroku and Cloud Foundry buildpacks
- Limited to namespace but can be shared across cluster as Cluster Build Template

Knative Components





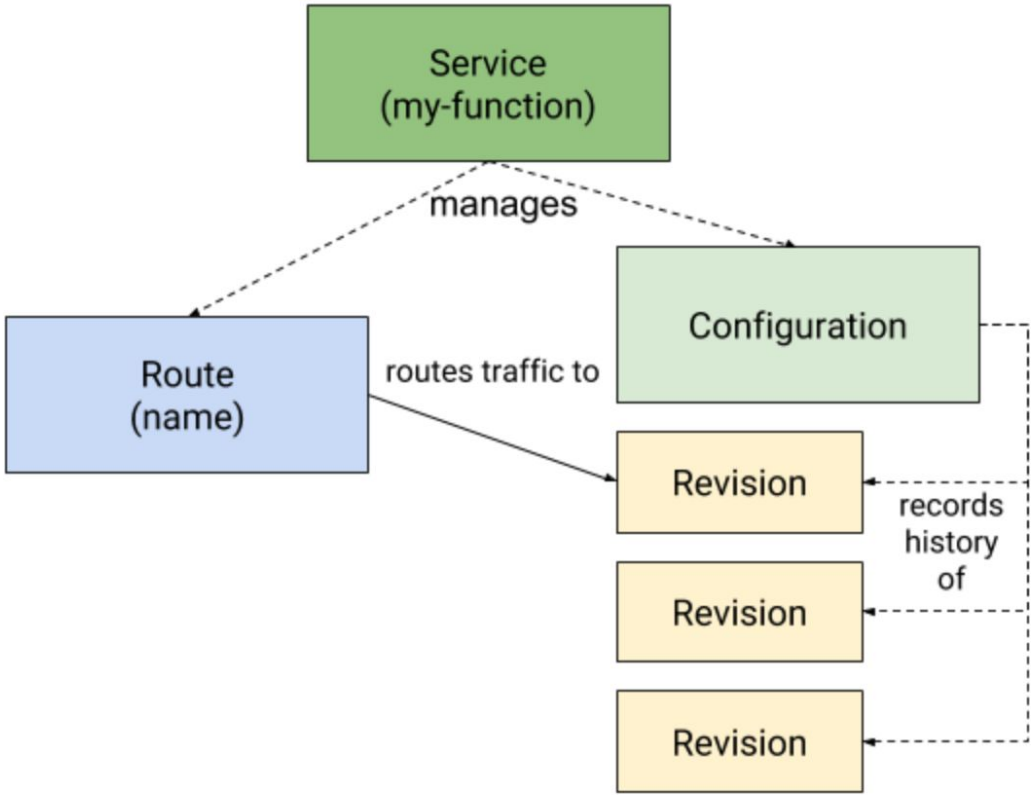
Serving

Request-driven serverless compute

Serving

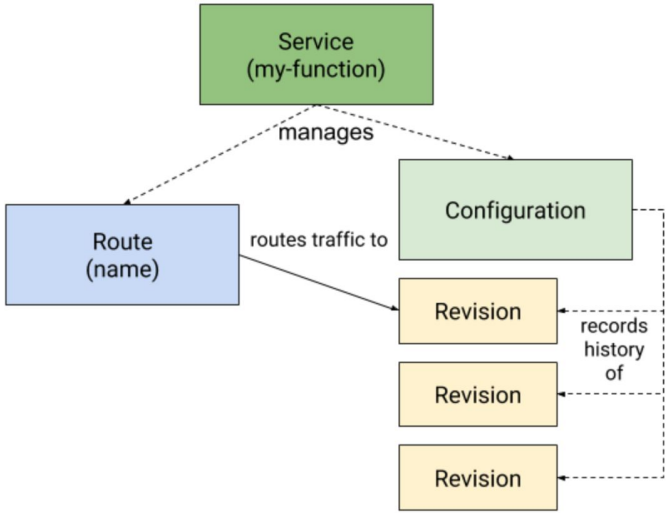
- Rapid deployment of serverless containers
- Programming for ingress gateway for network routing (Istio or Gloo)
- Point-in-time snapshots of deployed code and configurations (enabling rollback)
- Automatic scaling up and down to zero
- Observability through logs and metrics

Serving Components



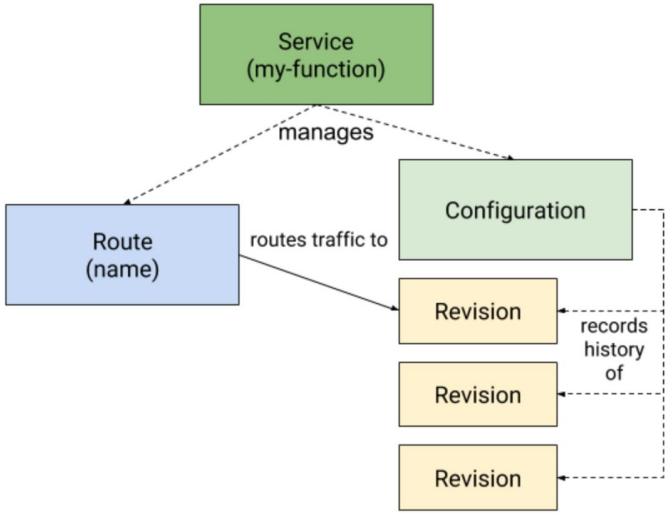
Route

- Maps HTTP routable endpoint to revision(s)
- Supports percentage based splitting of traffic



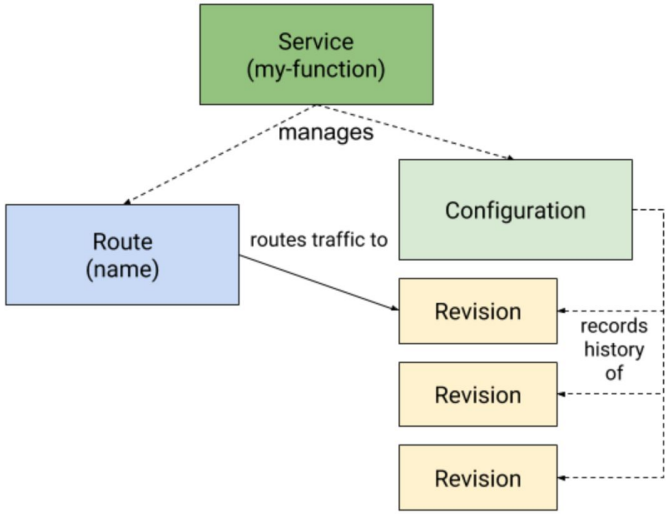
Revision

- Immutable code (as container or build) and config
- Generated by configuration changes
- Retired no route mapped (resources deleted)



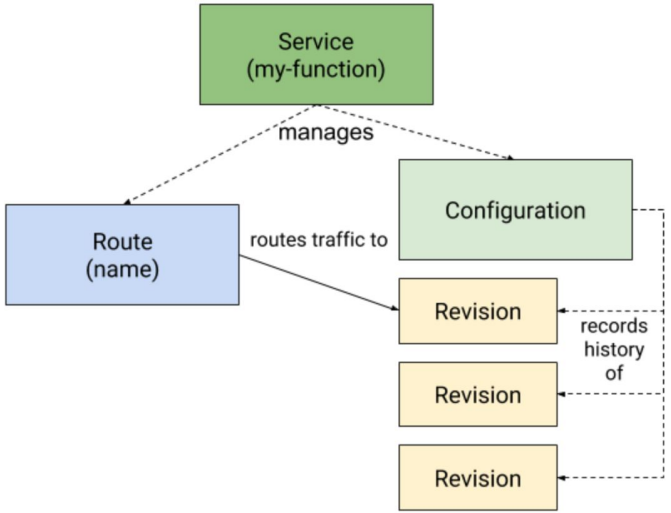
Configuration

- **Latest revision of code and config**



Service

- Set of routes and configurations/revisions
- Single abstraction of software component



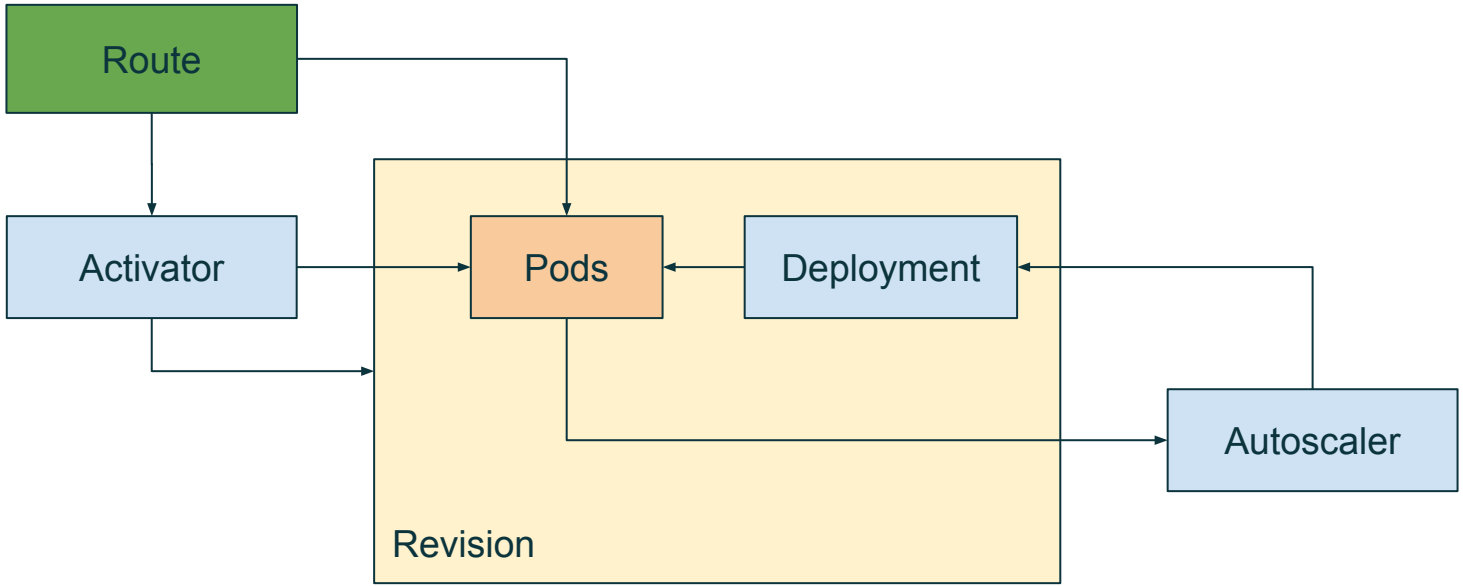
Autoscaling Goals

- Fast - scale from 0 to 1000 concurrent requests in < 30 seconds
- Light - System makes decision without operator intervention
- Make Everything Better - Custom components are short term until K8s scaling components improved

Pod States

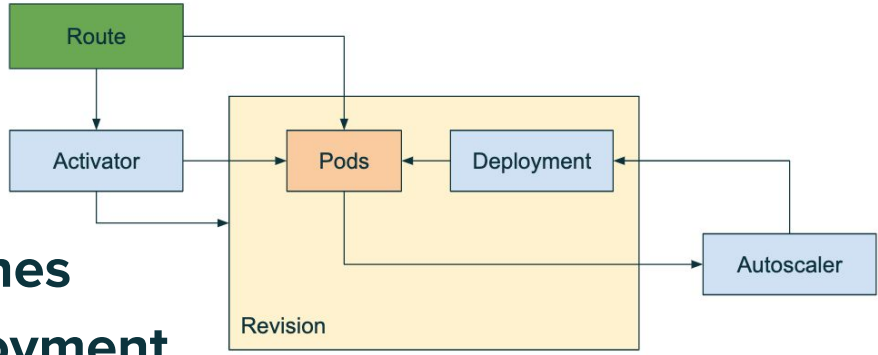
- Active - Handling current traffic
- Reserved - Scaled to 0
- Retired - No route mapped

Autoscaling Components



Autoscaling Components

- Knative Deployment - manages running revision
- Knative Service Autoscaler - watches revision metrics and advises deployment to scale increase/decrease pods
- Knative Activator - intercepts requests for revisions in reserved state, activates pods, forwards requests to activated pod



Autoscaling Modes

- Stable - 60 second window averages
  - calculated every 2 seconds
- Panic - 6 second window averages
  - Triggered when average is 2x desired
  - Increases only
  - Revert to stable after 60 seconds of stability
- Deactivation - average of 0 for configured threshold
  - Revision to reserved state
  - Default threshold 5 minutes
  - Minimum threshold 30 seconds

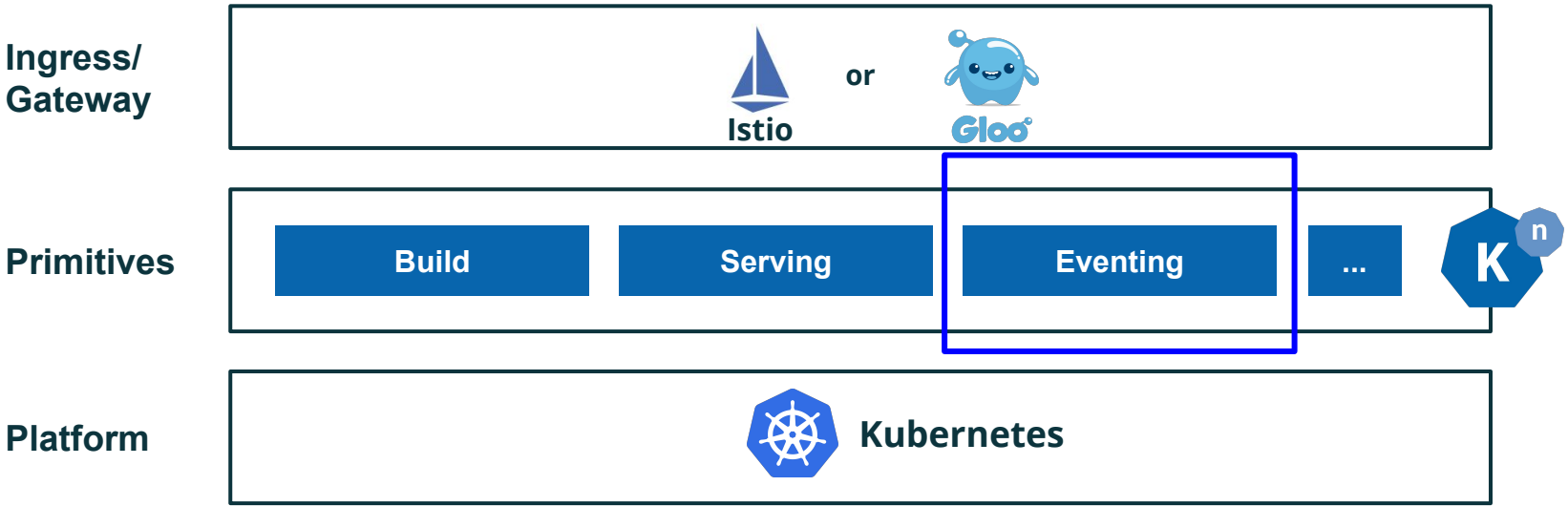
Autoscaling Classes

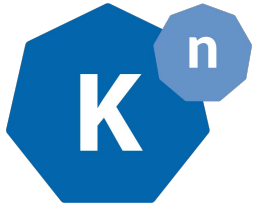
- Knative Autoscaler - concurrent requests per pod
  - default 100
- Kubernetes Horizontal Pod Autoscaler - CPU Percent
  - Default 80%

Logging And Metrics

- Knative collects logs using Fluentd
- Plugable logging/observability implementations
  - Elasticsearch and Kibana
  - Stackdriver
  - Custom logging plugin
- Provides Grafana metrics observability

Knative Components





Eventing

Declarative management and delivery of events

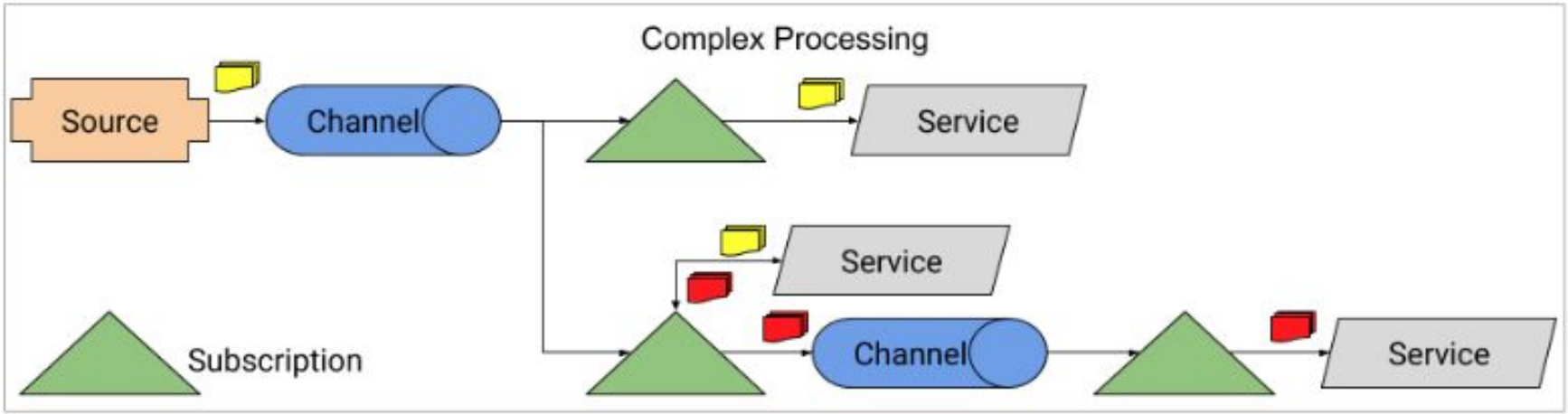
Eventing

- Composable primitives
- Late-binding loosely coupled event producers and consumers
- Consistent with CNCF Serverless WG CloudEvents specification

Services as Event Consumers

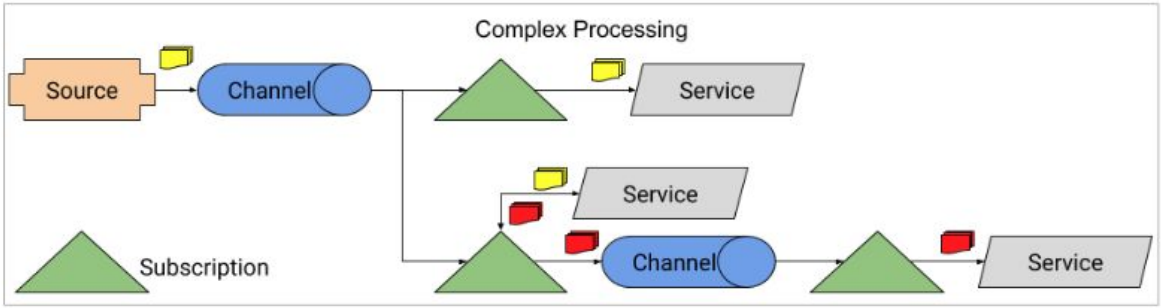
- Select and target specific subsets of the events from their producers
- Create new applications without modifying producer or consumer

Eventing Components



Source

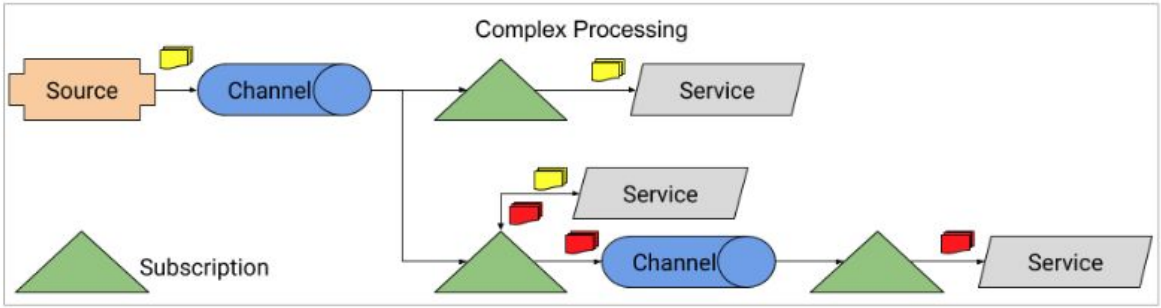
Register interest in events from other systems and publish to a sink (K8s events, Kafka, GitHub, GitLab, BitBucket, custom sources)



Service

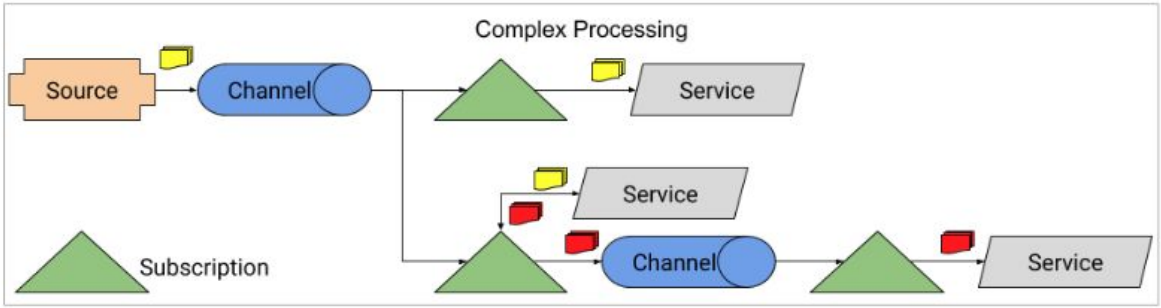
Consumer of events

- Addressable acknowledge delivery
- Callable - return 0 or 1 events



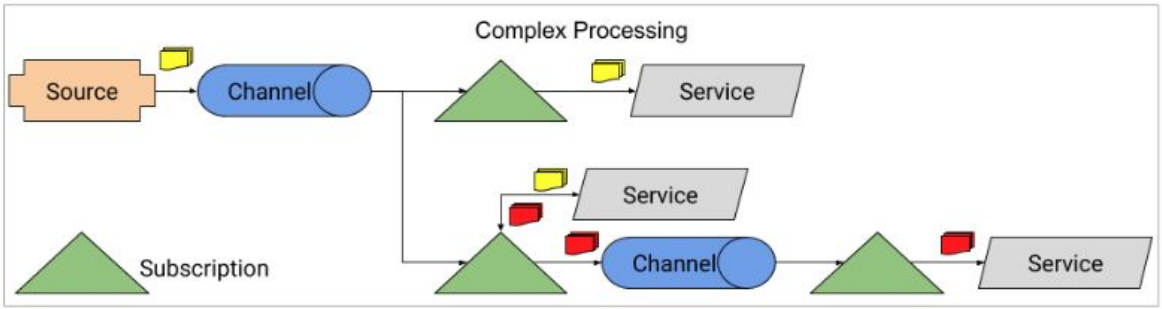
Channel

- Event forwarding persistence
- Plugable ClusterChannelProvisioner implementation for message providers

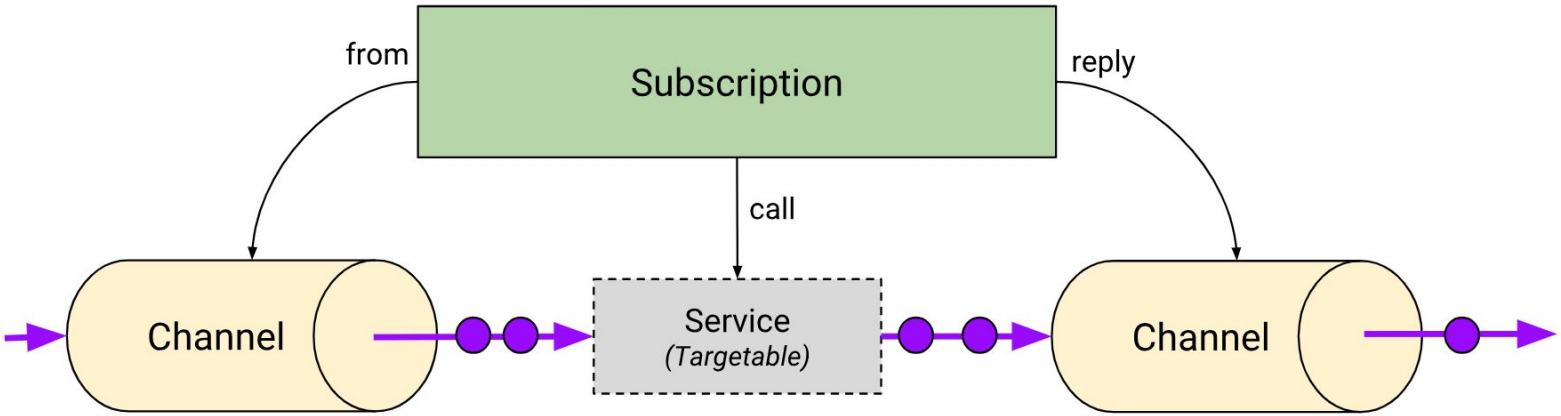


Subscription

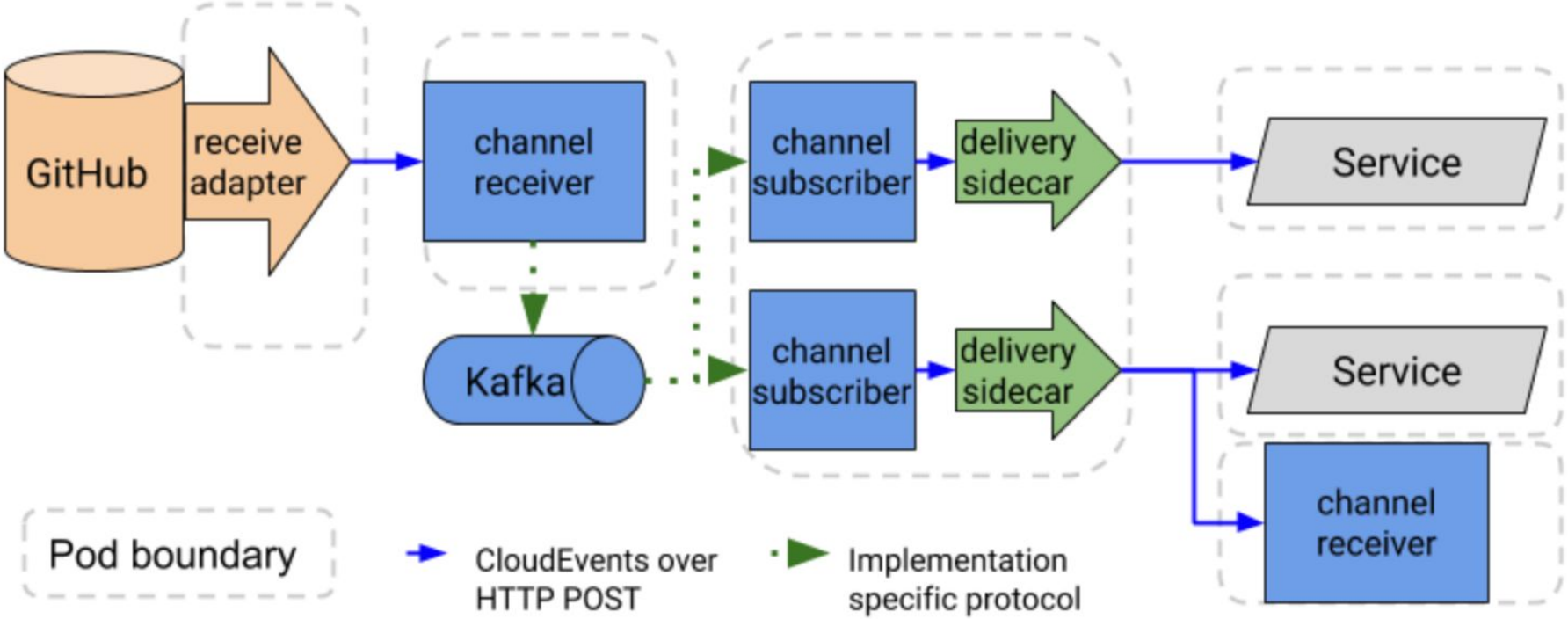
Means to declare interest in events from a channel



Channels & Subscriptions

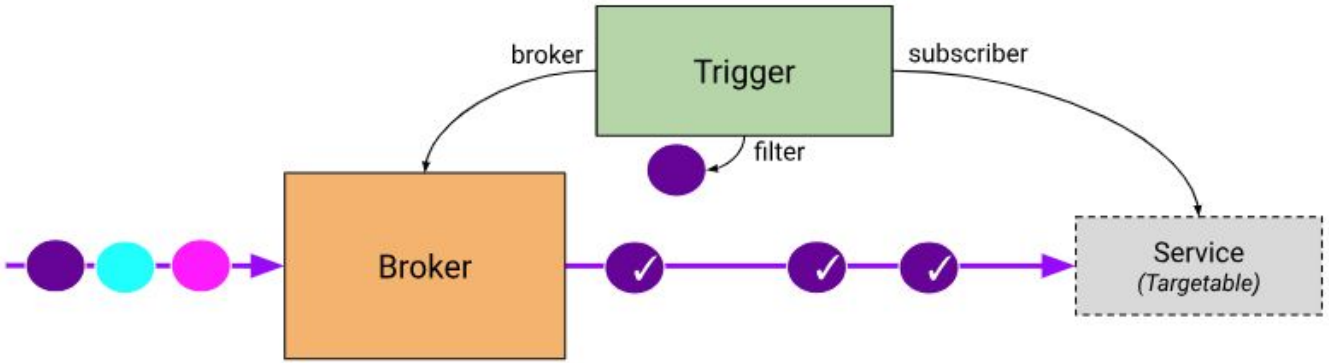


Data Plane

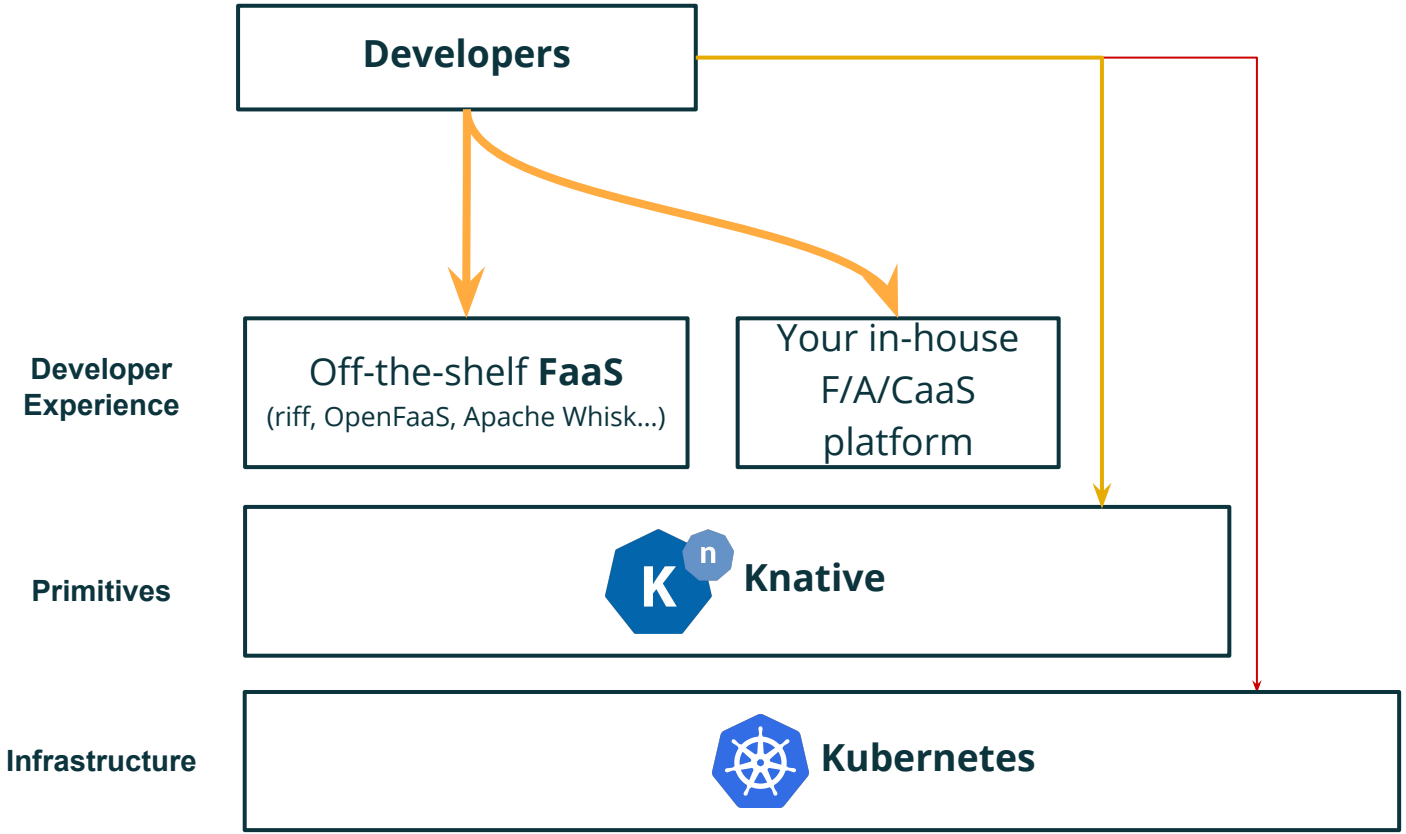

GitHub
receive adapter
channel receiver
Kafka
channel subscriber
channel subscriber
delivery sidecar
delivery sidecar
Service
Service
channel receiver
Pod boundary
CloudEvents over HTTP POST
Implementation specific protocol

Eventing Components (New as of Knative 0.5)

- Broker - mesh of events sent to interested subscribers
  - Use a channel template to create internal channels
- Trigger - declares subscription to events from a broker
  - Can be used to filter events based upon type



Developer Abstractions



Questions???

Where to go to learn more

- http://knative.dev
- https://github.com/knative/
- https://starkandwayne.com/blog/introduction-to-knative/
- https://medium.com/@pczarkowski/introduction-to-knative-b93a0b9aeeef
- https://www.ibm.com/cloud/learn/knative
- https://cloud.google.com/knative/

Config Sample Links

- Build
  - Build - https://github.com/GoogleCloudPlatform/knative-build-tutorials/blob/master/docker-build/README.md
  - Build Template - https://github.com/knative/build-templates/blob/master/kaniko/kaniko.yaml
  - Build with template - https://github.com/knative/build-templates/tree/master/kaniko
- Serving
  - Deploy and routing examples - https://github.com/knative/docs/blob/master/docs/serving/samples/blue-green-deployment.md
- Eventing
  - Channel - https://github.com/gswk/knative-eventing-demo/blob/master/channel.yaml
  - Subscription - https://github.com/gswk/knative-eventing-demo/blob/master/subscription.yaml
  - Broker - https://github.com/knative/eventing/tree/master/docs/broker
  - Trigger - https://github.com/knative/docs/blob/master/docs/eventing/samples/gcp-pubsub-source/trigger.yaml