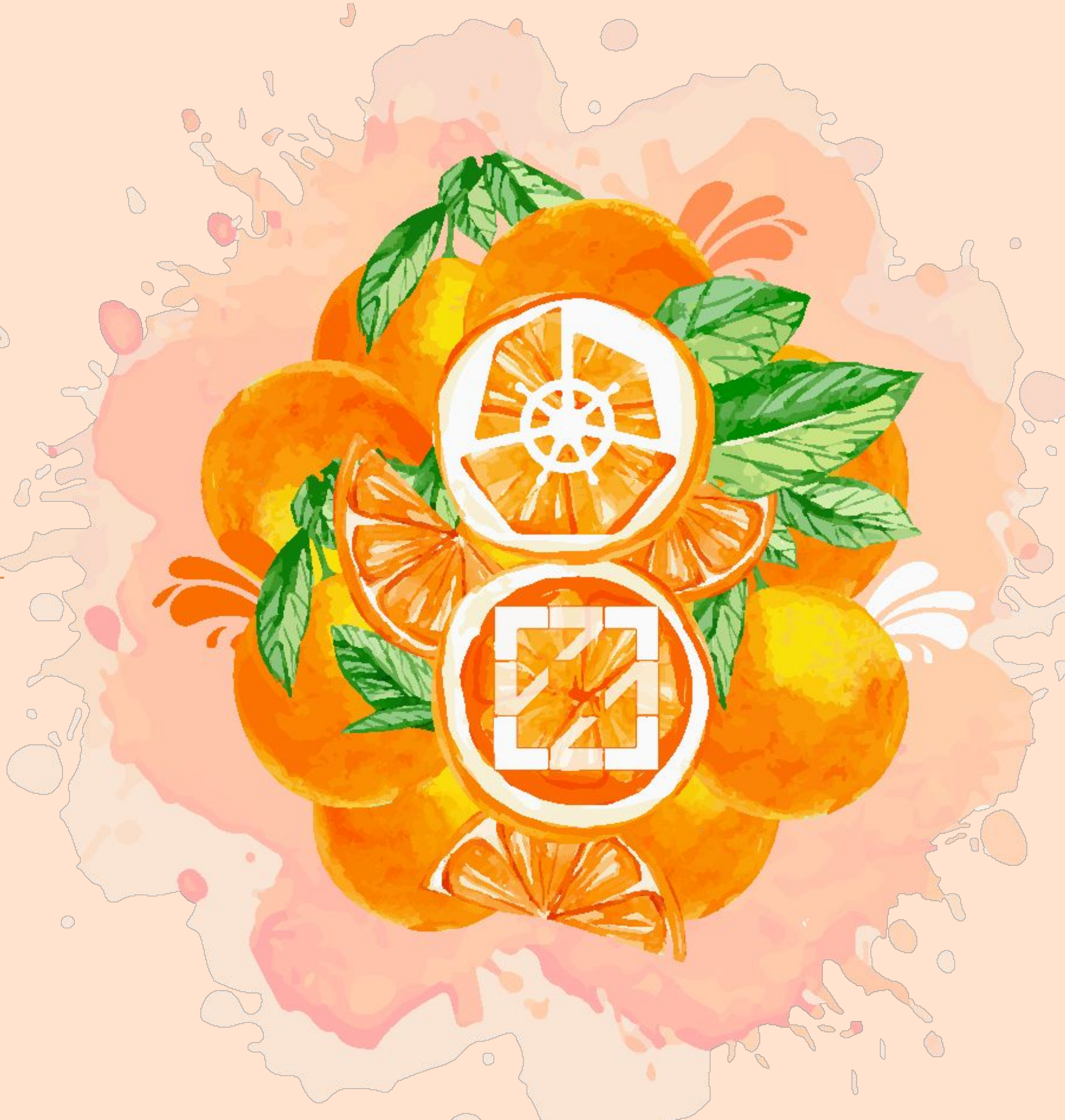
# KubeCon | CloudNativeCon

## Europe 2022

WELCOME TO VALENCIA



# Simplifying Service Mesh Ops with Flux and Flagger

Mitch Connors, Google
Stefan Prodan, Weaveworks

# Simplifying Service Mesh Ops with Flux and Flagger

**Mitch Connors**
Software Engineer
*Google*

**Stefan Prodan**
Software Engineer
*Weaveworks*

# Istio Makes Your Service Mesh Secure

Security is one of the leading motivations for adoption of Istio, with best-in-class features like:

- Zero Trust
- mTLS
- Certificate Rotation
- AuthN/AuthZ

But…

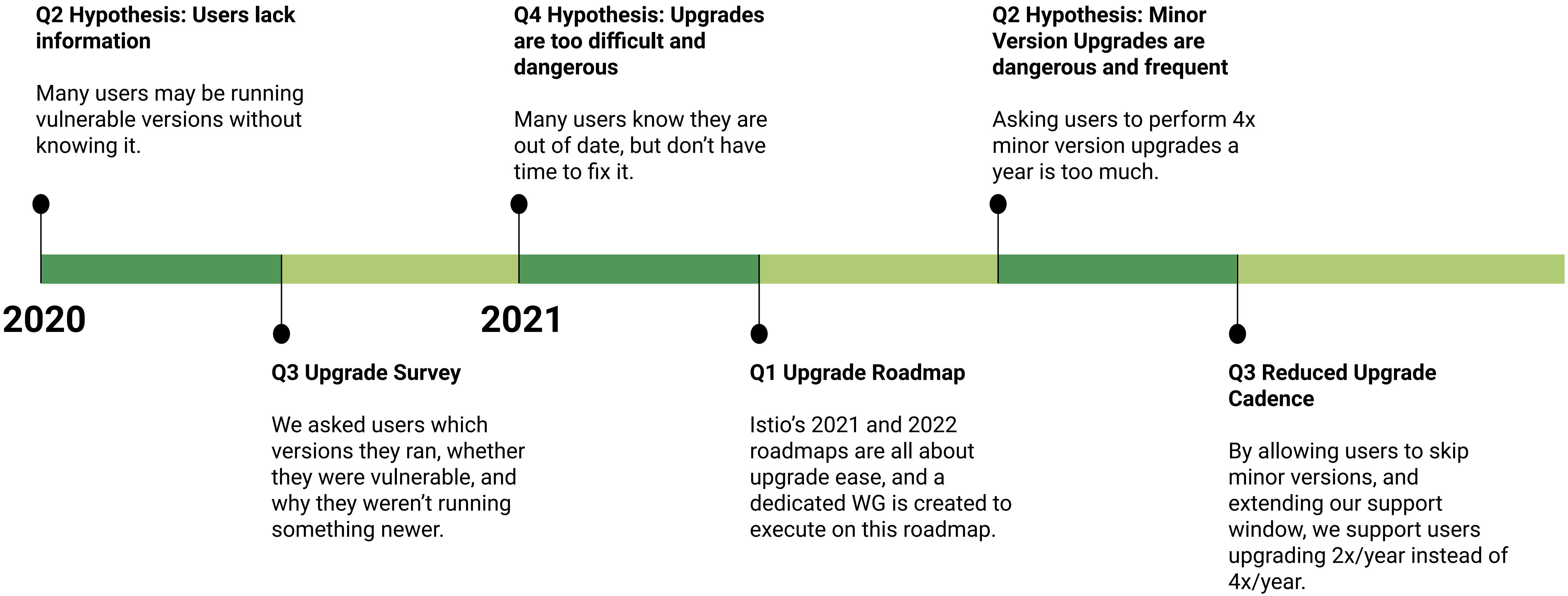
# 88%

Of Istio Installations are running known CVEs

# Agenda

- Why Aren’t Users Upgrading
- How Can GitOps Help
- Demo
- Takeaways

# Why Won’t They Upgrade?



**2020**

**Q2 Hypothesis: Users lack information**

Many users may be running vulnerable versions without knowing it.

**Q3 Upgrade Survey**

We asked users which versions they ran, whether they were vulnerable, and why they weren’t running something newer.

**2021**

**Q4 Hypothesis: Upgrades are too difficult and dangerous**

Many users know they are out of date, but don’t have time to fix it.

**Q1 Upgrade Roadmap**

Istio’s 2021 and 2022 roadmaps are all about upgrade ease, and a dedicated WG is created to execute on this roadmap.

**Q2 Hypothesis: Minor Version Upgrades are dangerous and frequent**

Asking users to perform 4x minor version upgrades a year is too much.

**Q3 Reduced Upgrade Cadence**

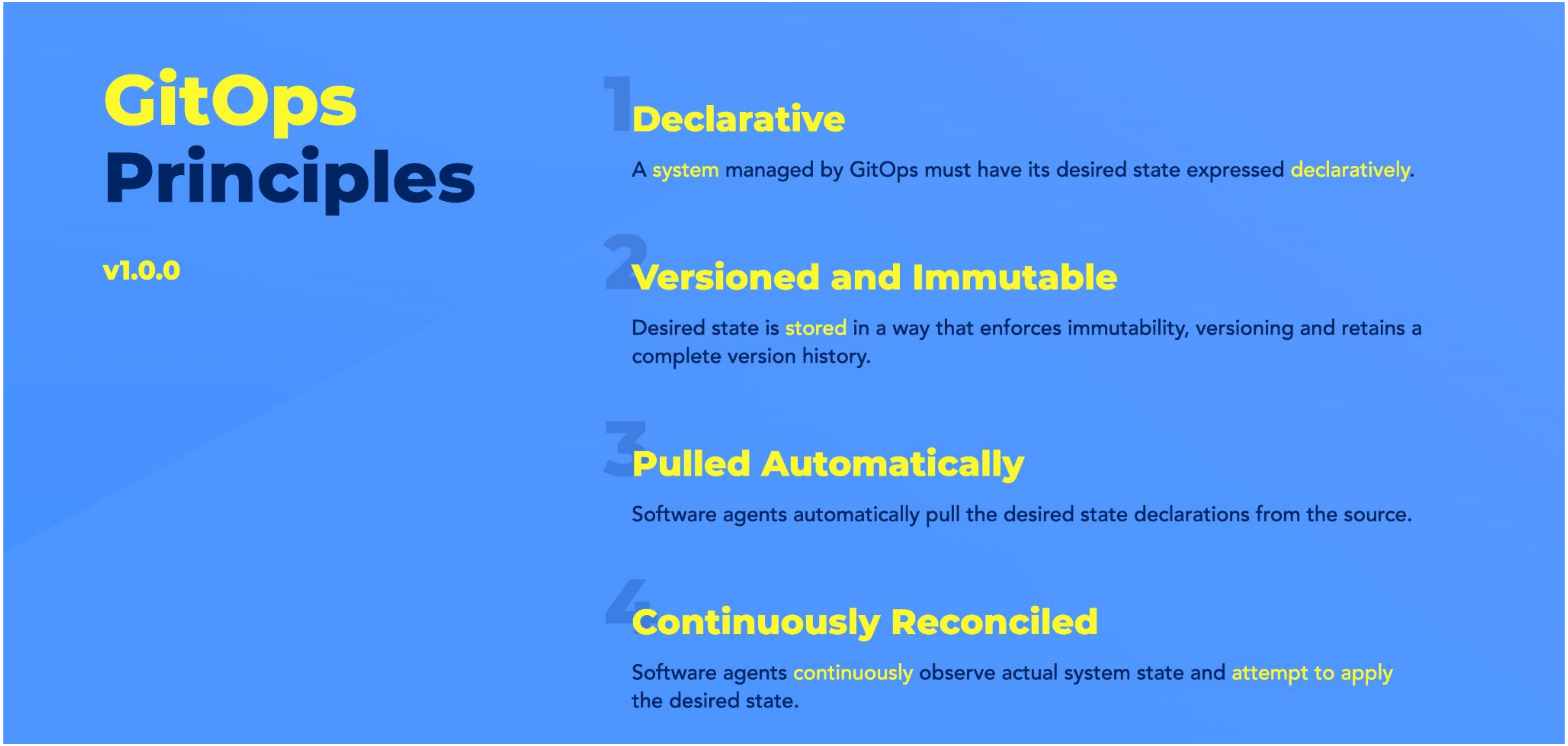
By allowing users to skip minor versions, and extending our support window, we support users upgrading 2x/year instead of 4x/year.

# Humans are Bad at Repetitive, Monotonous Labor

New Hypotheses, Q1 ‘22

# Enter Gitops

- What is GitOps?
- What is Progressive Delivery?
- Flux and Flagger
- GitHub Actions



# GitOps Principles

v1.0.0

## 1 Declarative

A system managed by GitOps must have its desired state expressed declaratively.

## 2 Versioned and Immutable

Desired state is stored in a way that enforces immutability, versioning and retains a complete version history.

## 3 Pulled Automatically

Software agents automatically pull the desired state declarations from the source.

## 4 Continuously Reconciled

Software agents continuously observe actual system state and attempt to apply the desired state.

https://opengitops.dev/

# CNCF Flux Project

The Flux project aims to provide a complete **Continuous Delivery** platform on top of Kubernetes, supporting all the common practices and tooling in the field.

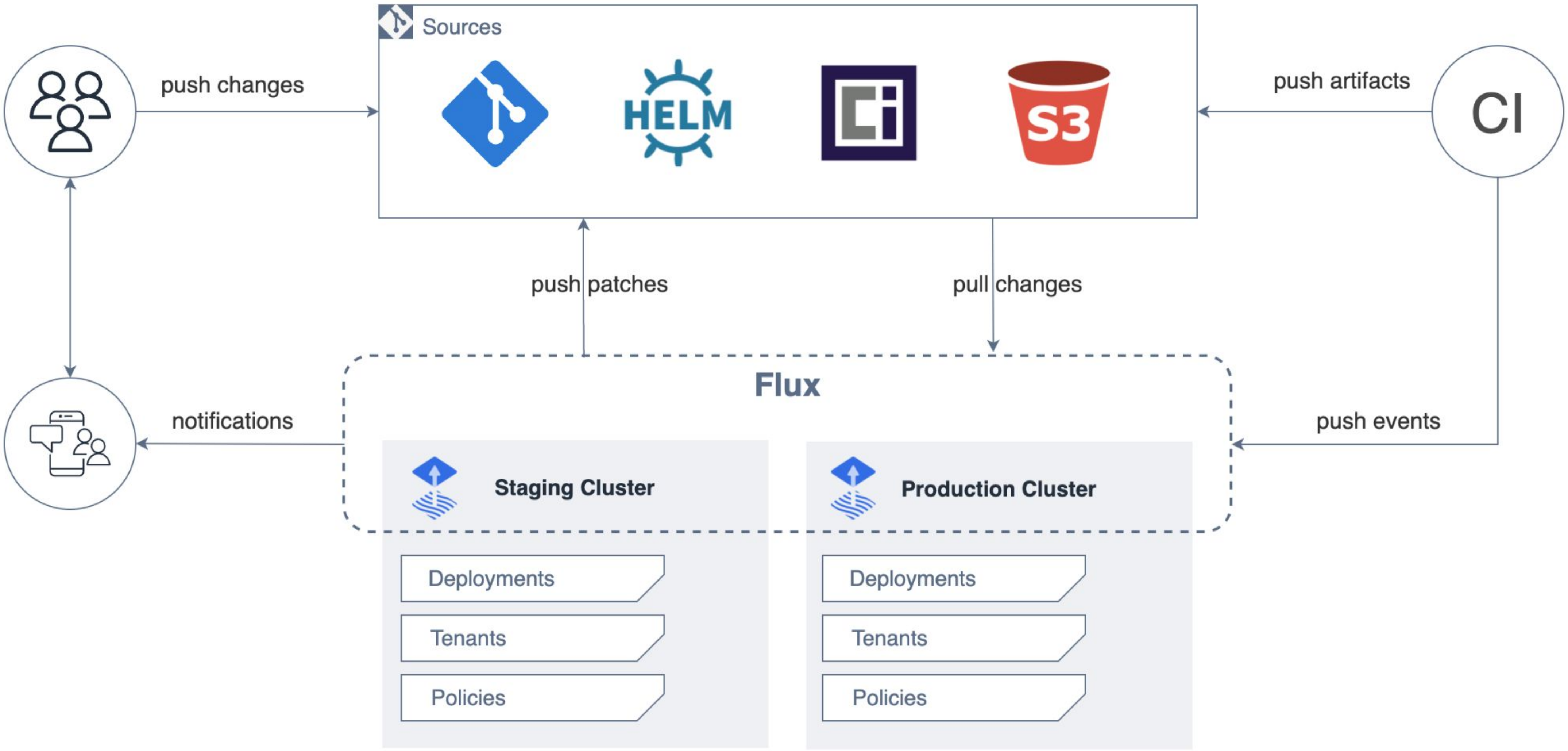
https://github.com/fluxcd/flux2

Flux v2 is powered by the **GitOps Toolkit**, a set of composable APIs and specialized tools for keeping Kubernetes clusters in sync with sources of configuration, and automating updates to configuration when there is new code to deploy.

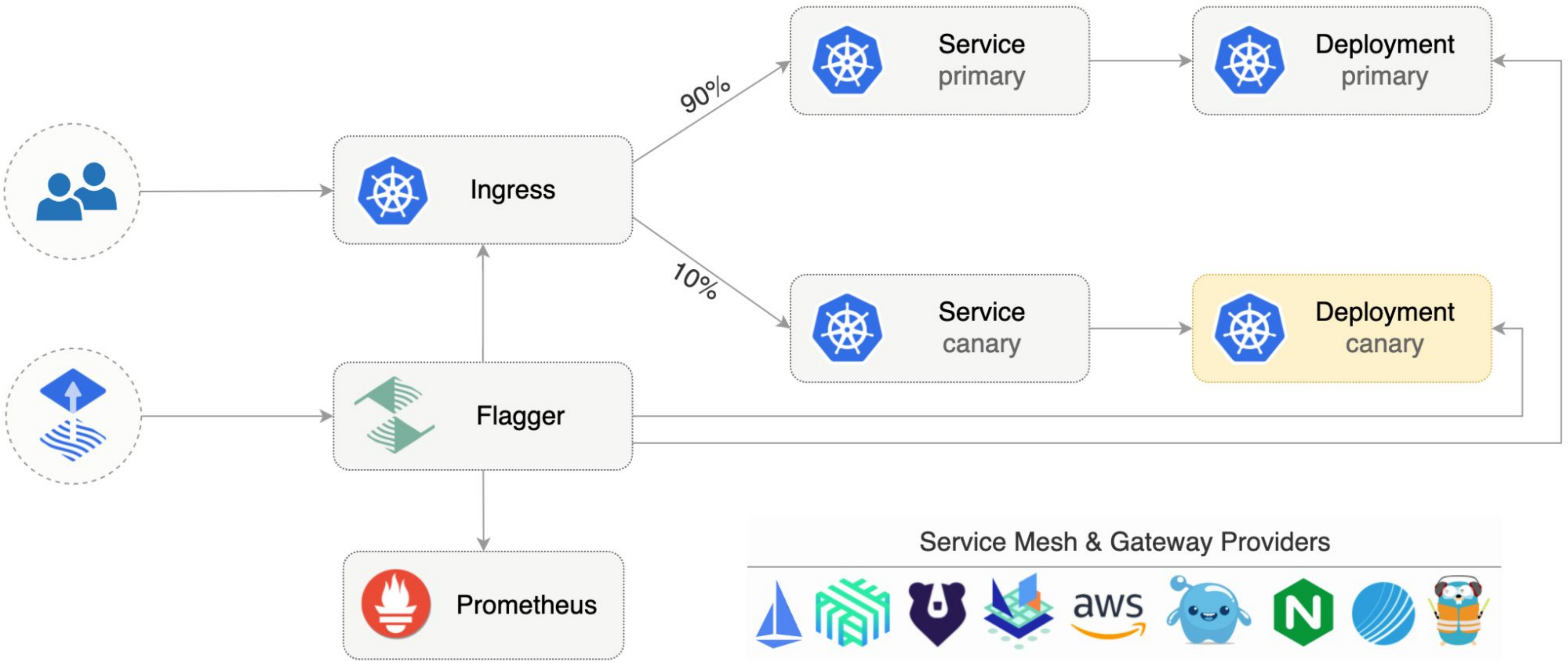
https://github.com/fluxcd/flagger

Flagger is a **Progressive Delivery** tool that automates the release process for applications running on Kubernetes. Flagger comes with a declarative model for **decoupling** the deployment of apps on Kubernetes from the release process.

# Flux - GitOps Continuous Delivery



# Flagger - Progressive Delivery



# It is turtles all the way down

Using Flux to keep your app up-to-date is great, but someone is going to have to upgrade Flux…

GitHub Actions to the rescue!

# Demo

Bootstrapping a New Cluster with gitops-istio

# GitOps Principles Keep Istio Up To Date

github.com/stefanprodan/gitops-istio

# Why Flagger?

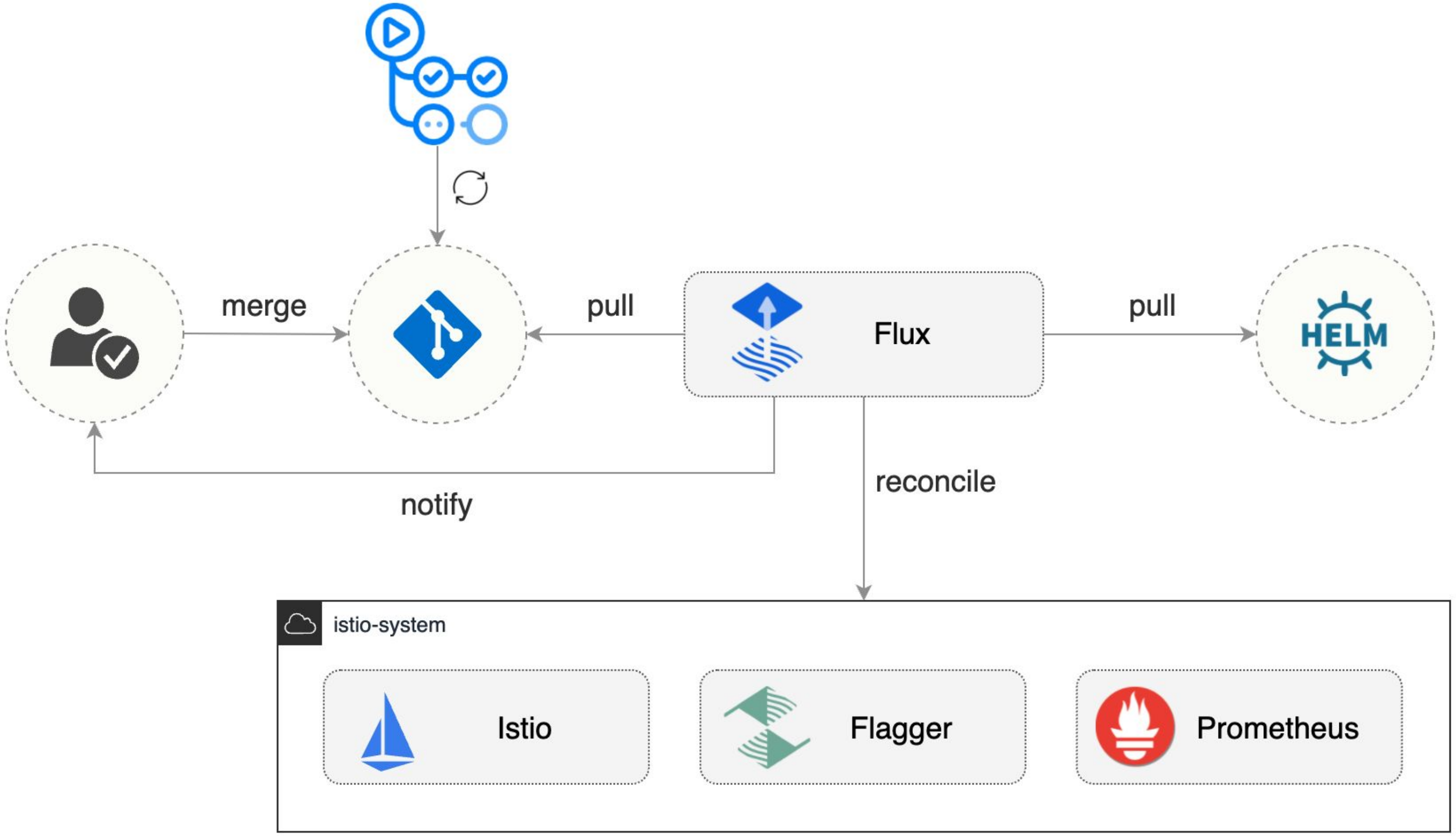
Excellent GitOps Solutions abound, but

- Flagger is the only Progressive Delivery API
- Stefan already did all the work!

# GitOps for IstioD

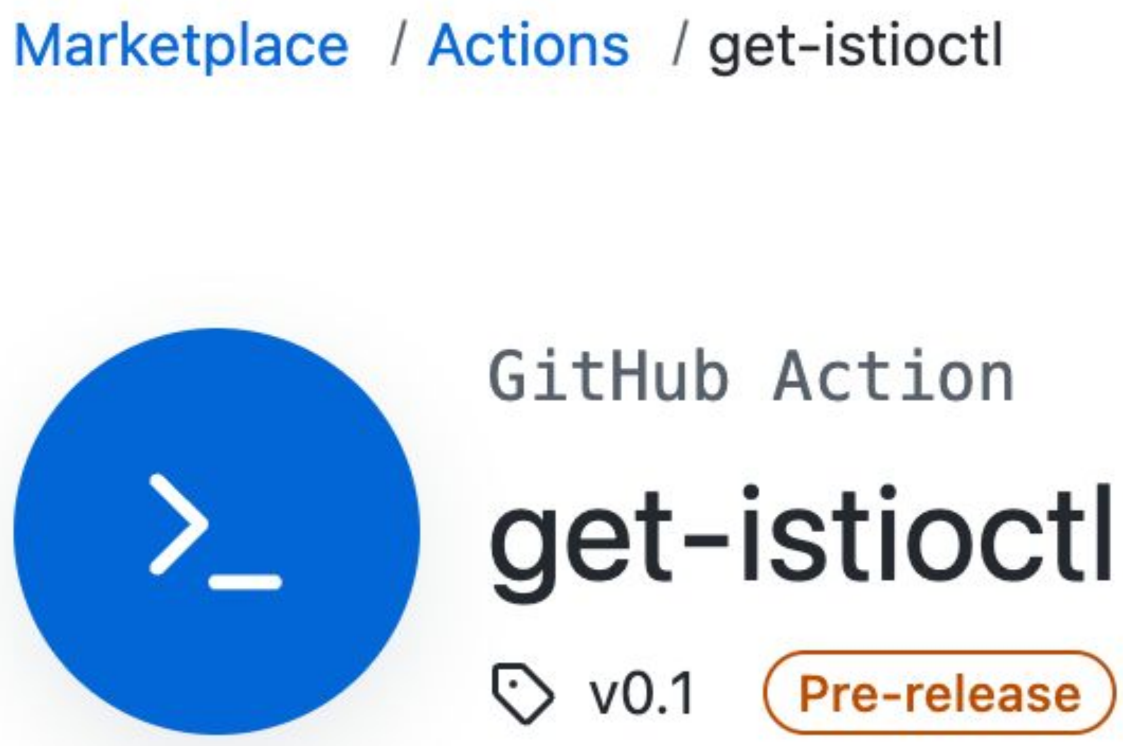
Flux for managing the Istio control plane and its addons from Git.

GitHub Actions for checking new Istio Versions, opening Pull Requests and running E2E tests in Kubernetes Kind.



https://github.com/stefanprodan/gitops-istio

# Launched: GitHub Actions for Istio



Marketplace / Actions / get-istioctl

GitHub Action

get-istioctl

v0.1 Pre-release

```
- uses: ./.github/actions/get-istioctl
  id: getit
  with:
    version: "1.11.*"
- name: Get the istioctl version
  run: |
    ./istioctl version --remote=false"
```

# Demo

Upgrading Istio Control Plane w/ GitHub Actions + Flux

## Advantages

- Control Plane maintained up-to-date within semver range
- Out-of-range updates trigger pull requests from GitHub Actions
- Full stack tested in kind e2e tests before production rollout
- Istioctl Analyze runs on every change

## Disadvantages

- Proxy Upgrades still uncontrolled
- Needs updates for Revision-based upgrades

# Terminology Warning

For this presentation, Proxy, Sidecar, and Data Plane all refer to the Envoy Sidecar Proxy provided by Istio.

# Istio Architecture Primer

Istio

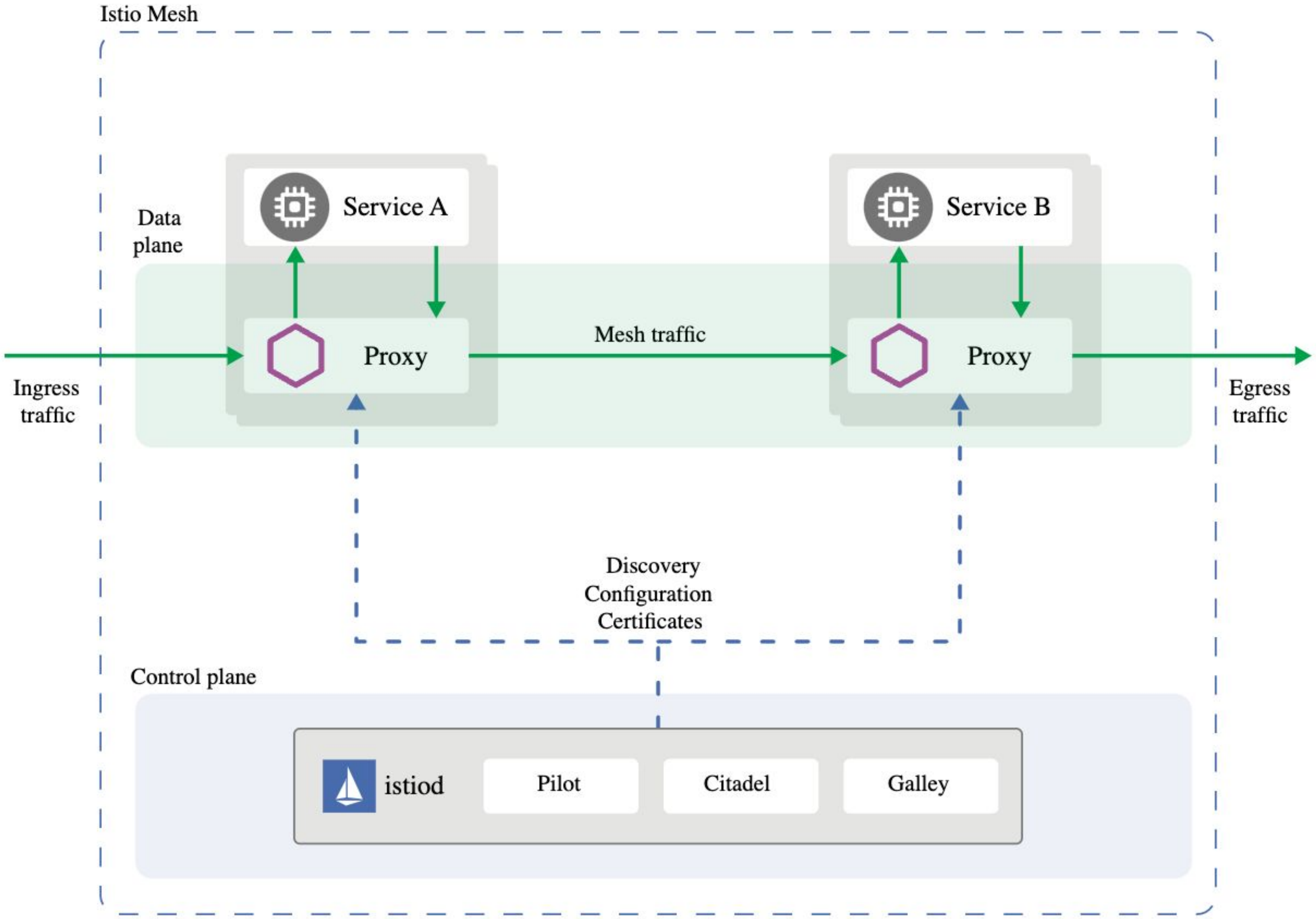
=

1x Control Plane

+

Data Plane

(N Proxies)

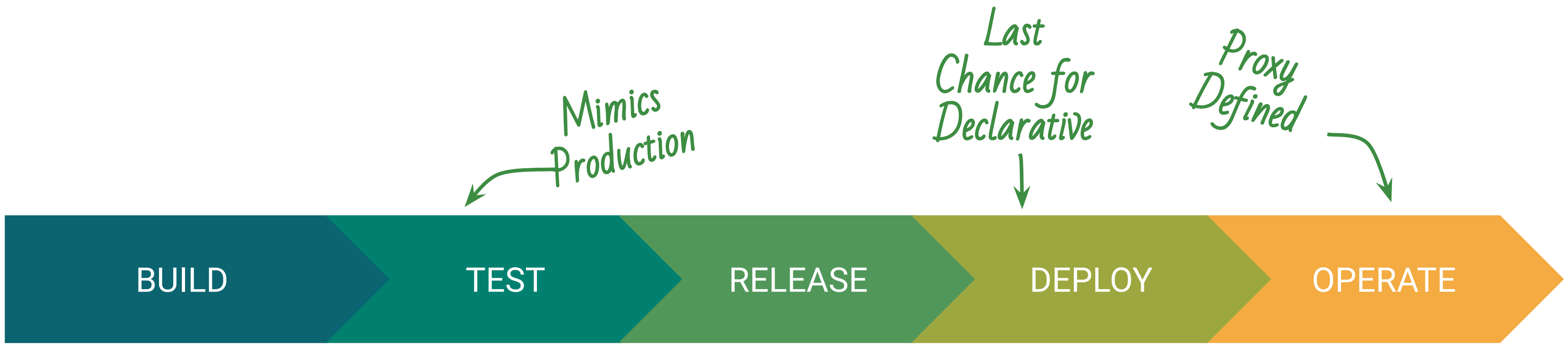


Istio Architecture

# The Problem with Proxies

Imperative Definition

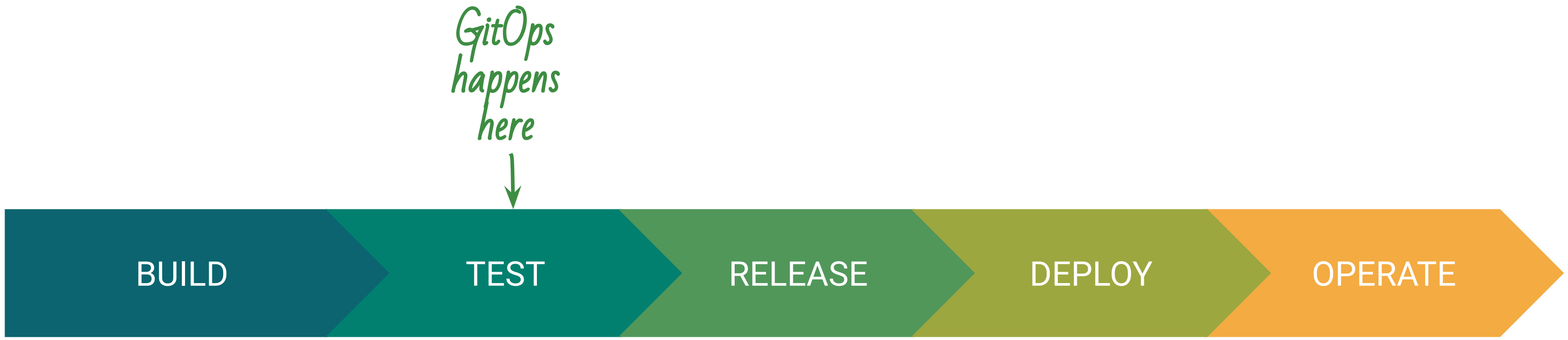
- Reduces Test Determinacy
- Updates Rollout Uncontrolled
- Frequent Global Restarts Required



# Shifting Left

Declarative Definition

- All environments use same proxy
- Proxy Upgrades according to Application Rules
- Rollbacks are as simple as git revert

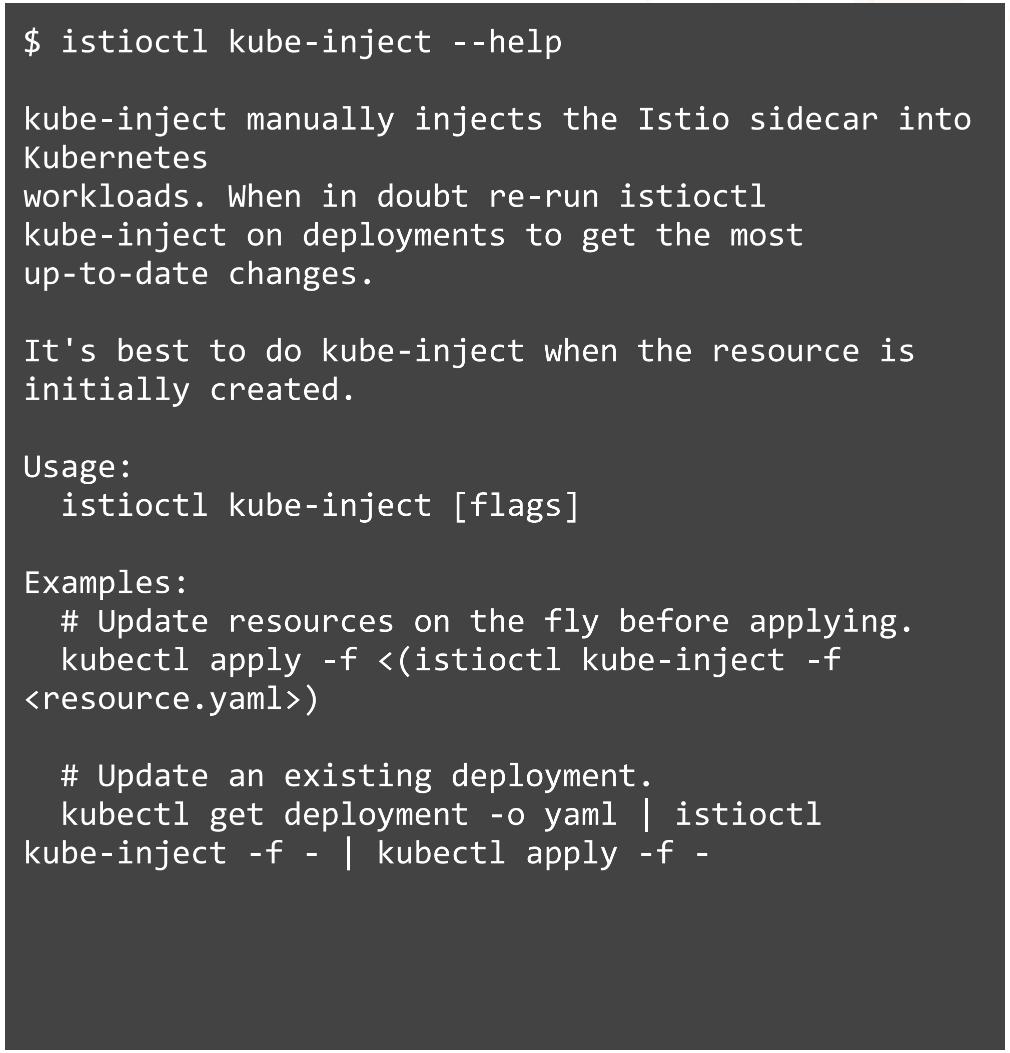


# How to Define your Sidecar in Git

Disable Sidecar Injection

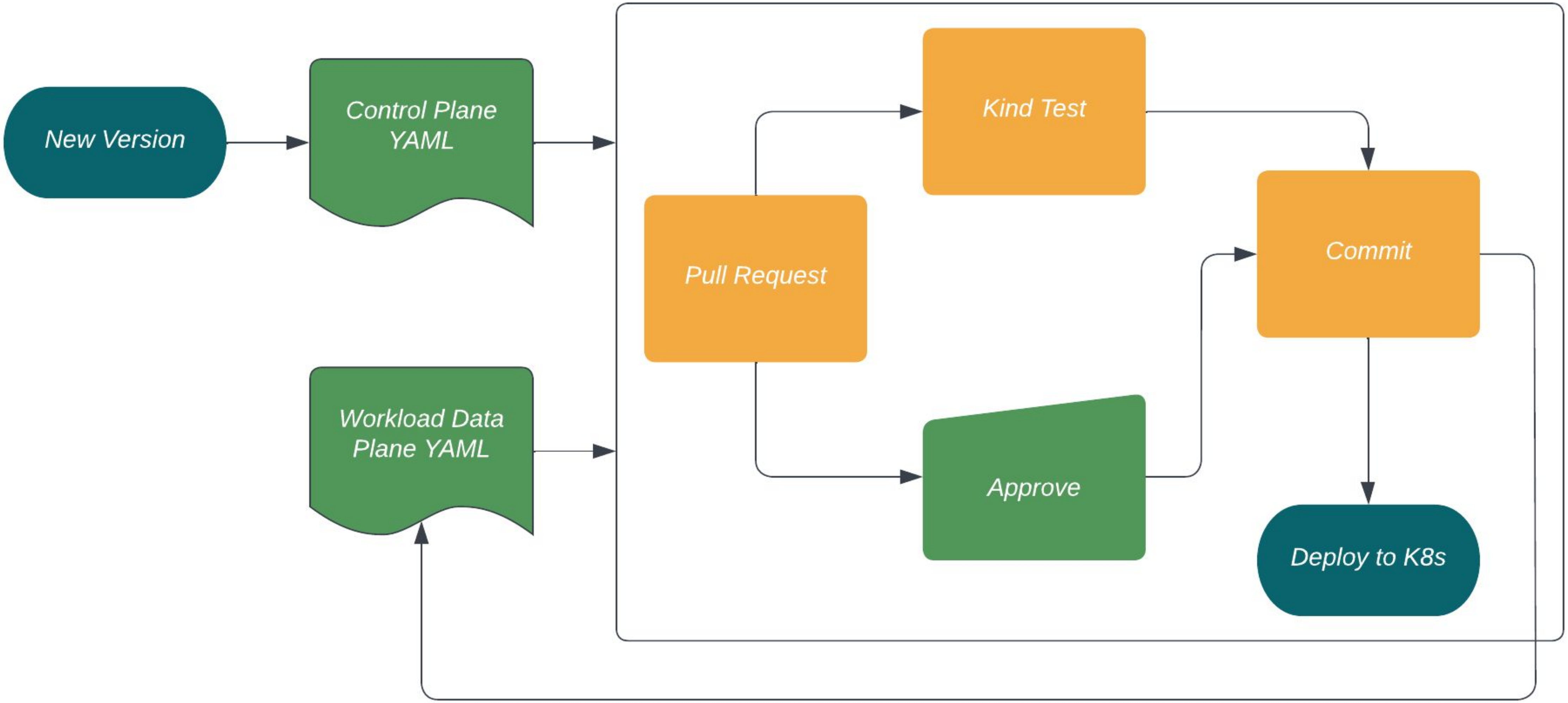
Run kube-inject from GitHub Actions

Deployment.yaml
+
istioctl kube-inject
=
Deployment.yaml w/sidecar



```
$ istioctl kube-inject --help

kube-inject manually injects the Istio sidecar into
Kubernetes
workloads. When in doubt re-run istioctl
kube-inject on deployments to get the most
up-to-date changes.

It's best to do kube-inject when the resource is
initially created.

Usage:
  istioctl kube-inject [flags]

Examples:
  # Update resources on the fly before applying.
  kubectl apply -f <(istioctl kube-inject -f
<resource.yaml>)

  # Update an existing deployment.
  kubectl get deployment -o yaml | istioctl
kube-inject -f - | kubectl apply -f -
```

# Current Workflow



# Demo

Upgrading Istio Data Plane w/ GitHub Actions + Flux

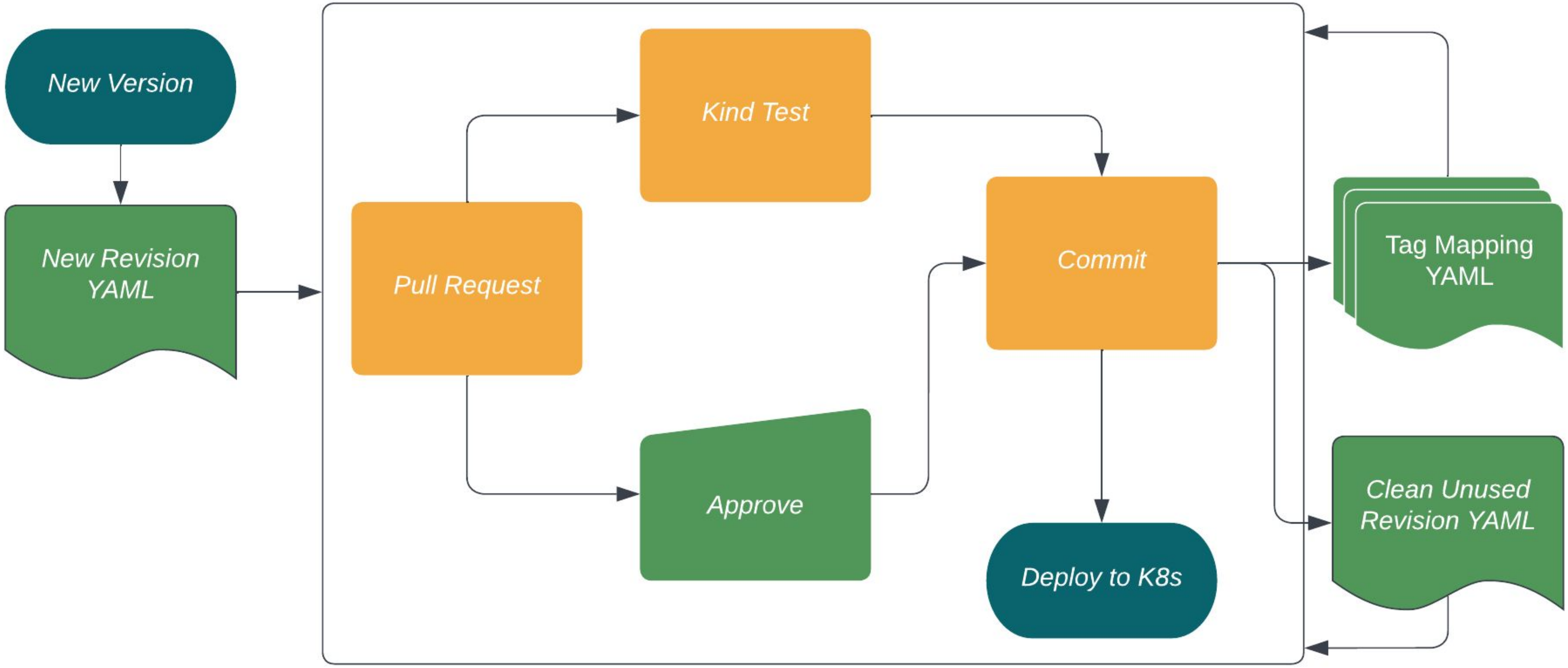
# Advantages

- Proxy maintained up-to-date within semver range
- Side effect: Canaries rollout out updates to proxy, as well as application
- Automated Canary rollback

# Disadvantages

- Doesn’t respect Revisions
- Rollouts to multiple clusters must be manually coordinated

# Future Workflow



# Key Takeaways

Automate Your Istio Upgrades

OR

Pay a vendor to upgrade Istio

OR

Budget engineering time to upgrade Istio

# Additional Resources

https://cloud.google.com/anthos/service-mesh

https://fluxcd.io

https://www.weave.works/product/gitops-enterprise/

KubeCon | CloudNativeCon

Europe 2022

THANK YOU!

github.com/stefanprodan/gitops-istio