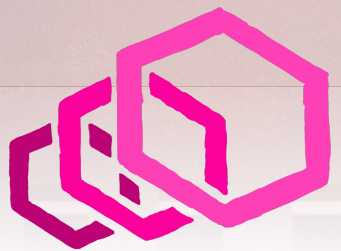
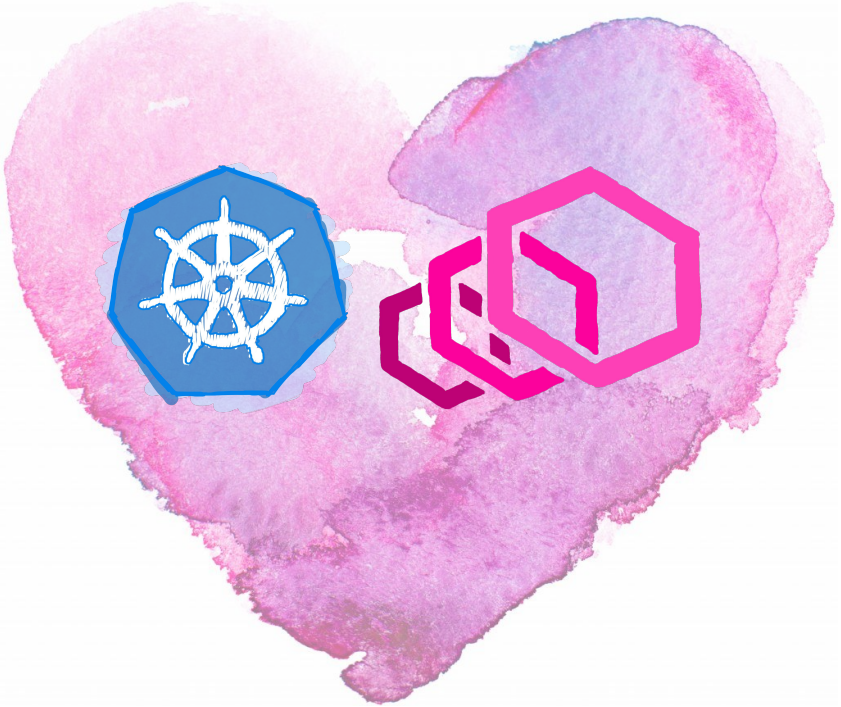
LITA CHO & TOM WANIELISTA

Service Mesh in Kubernetes: It's not that Easy



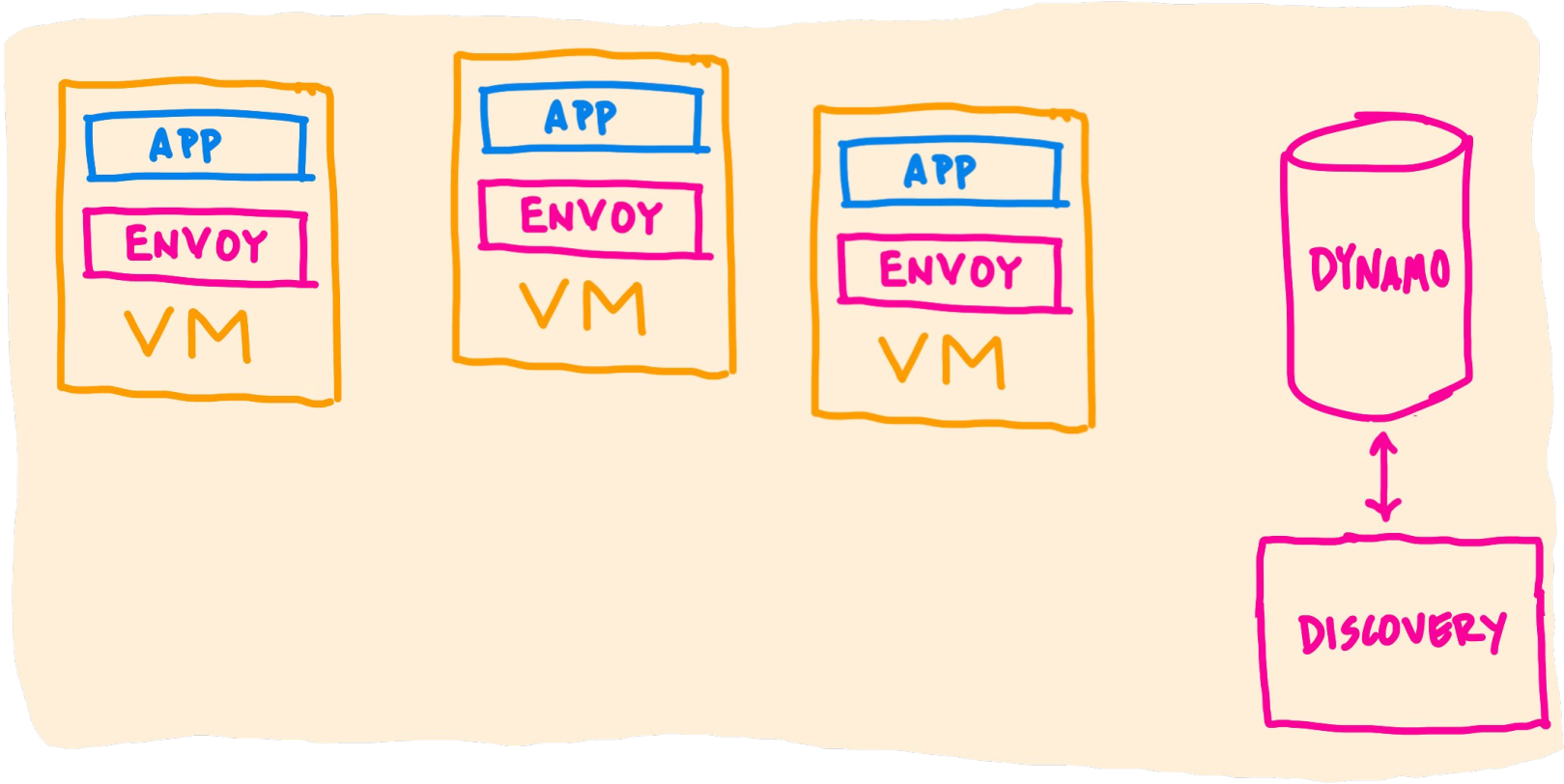
lyft

Agenda

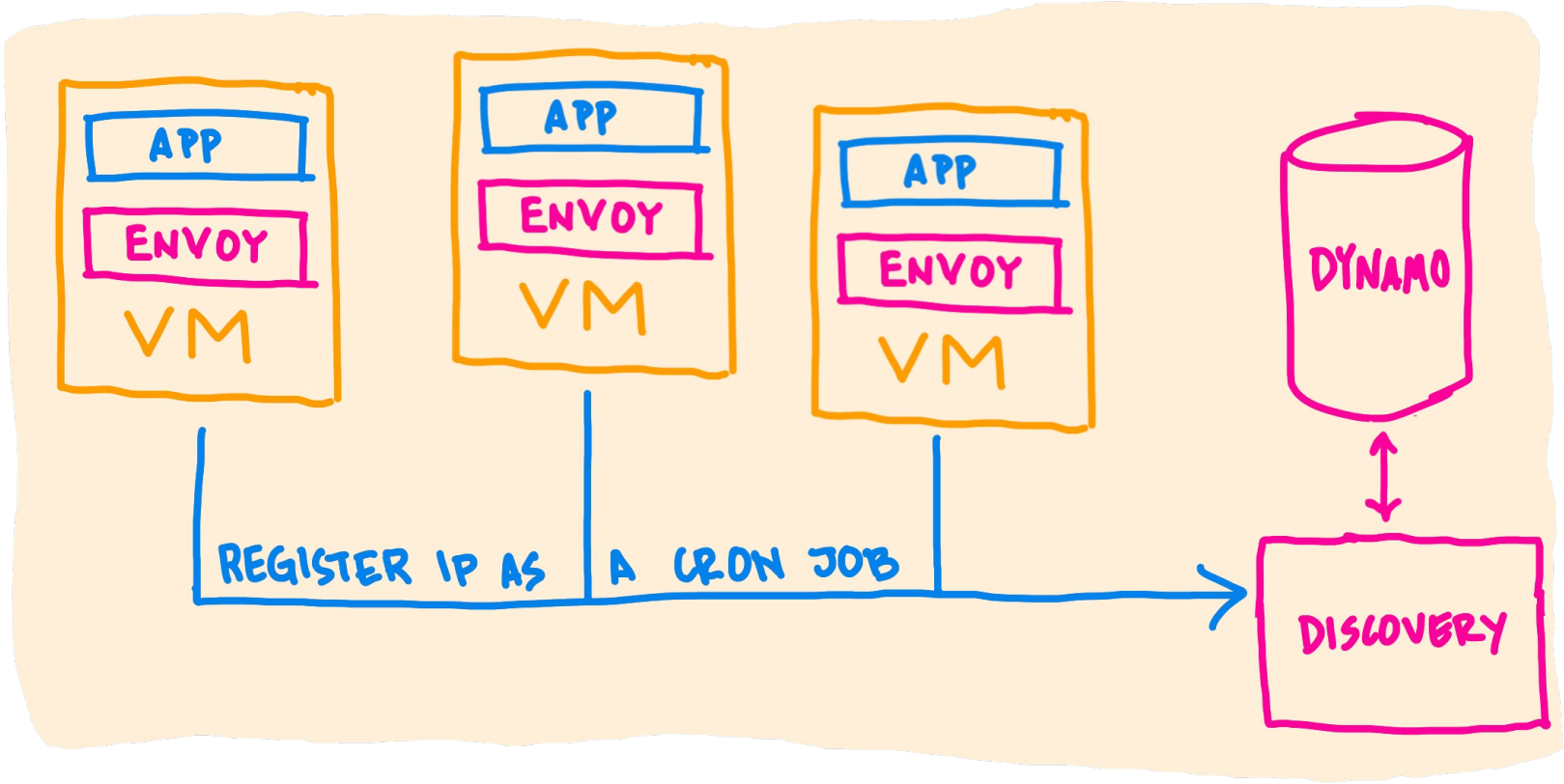


- Service Discovery
  - Legacy vs. Kubernetes
- Issues
  - Scale up
  - Scale down
- Future Work

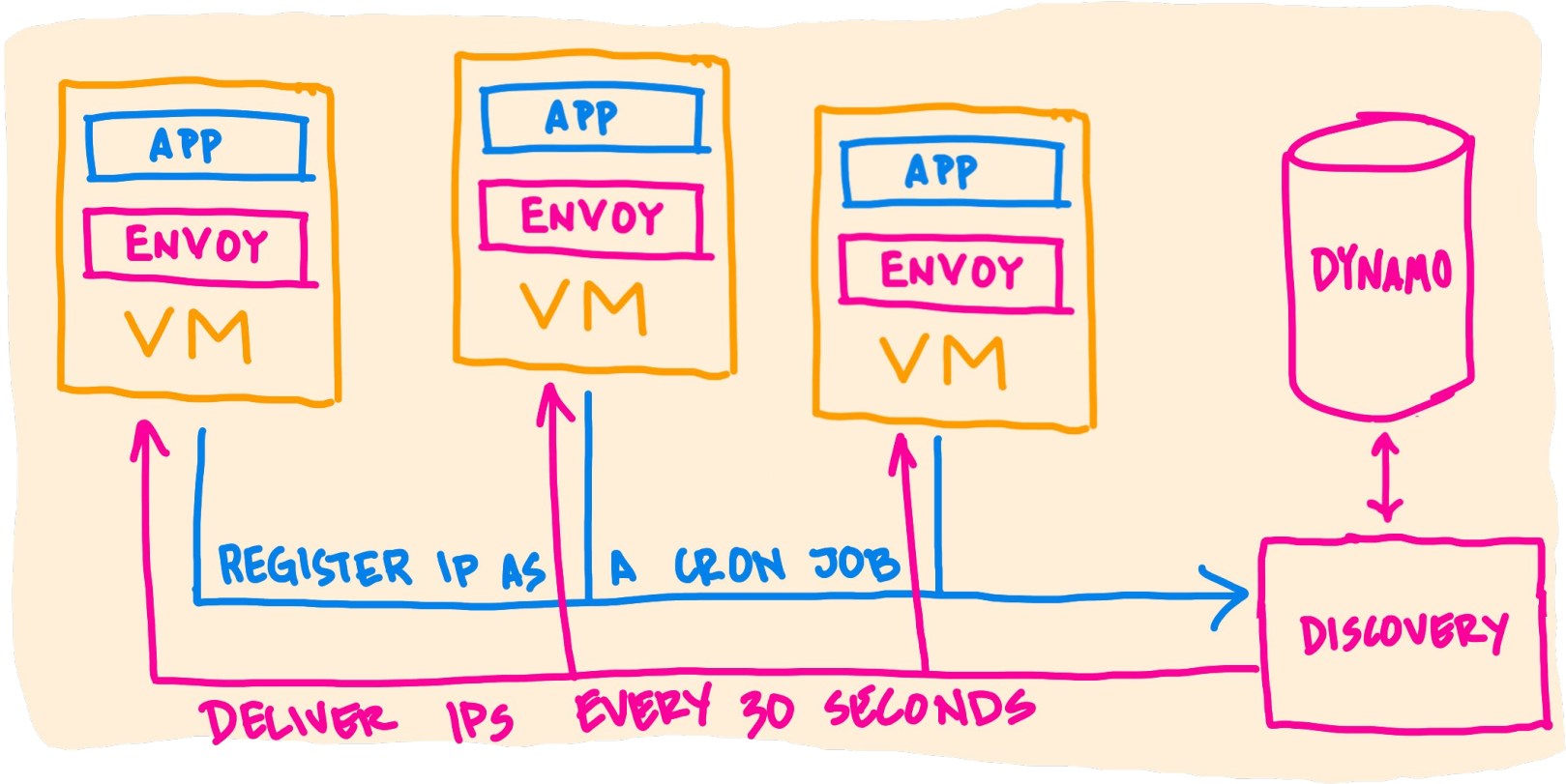
Pre-Kubernetes Service Discovery



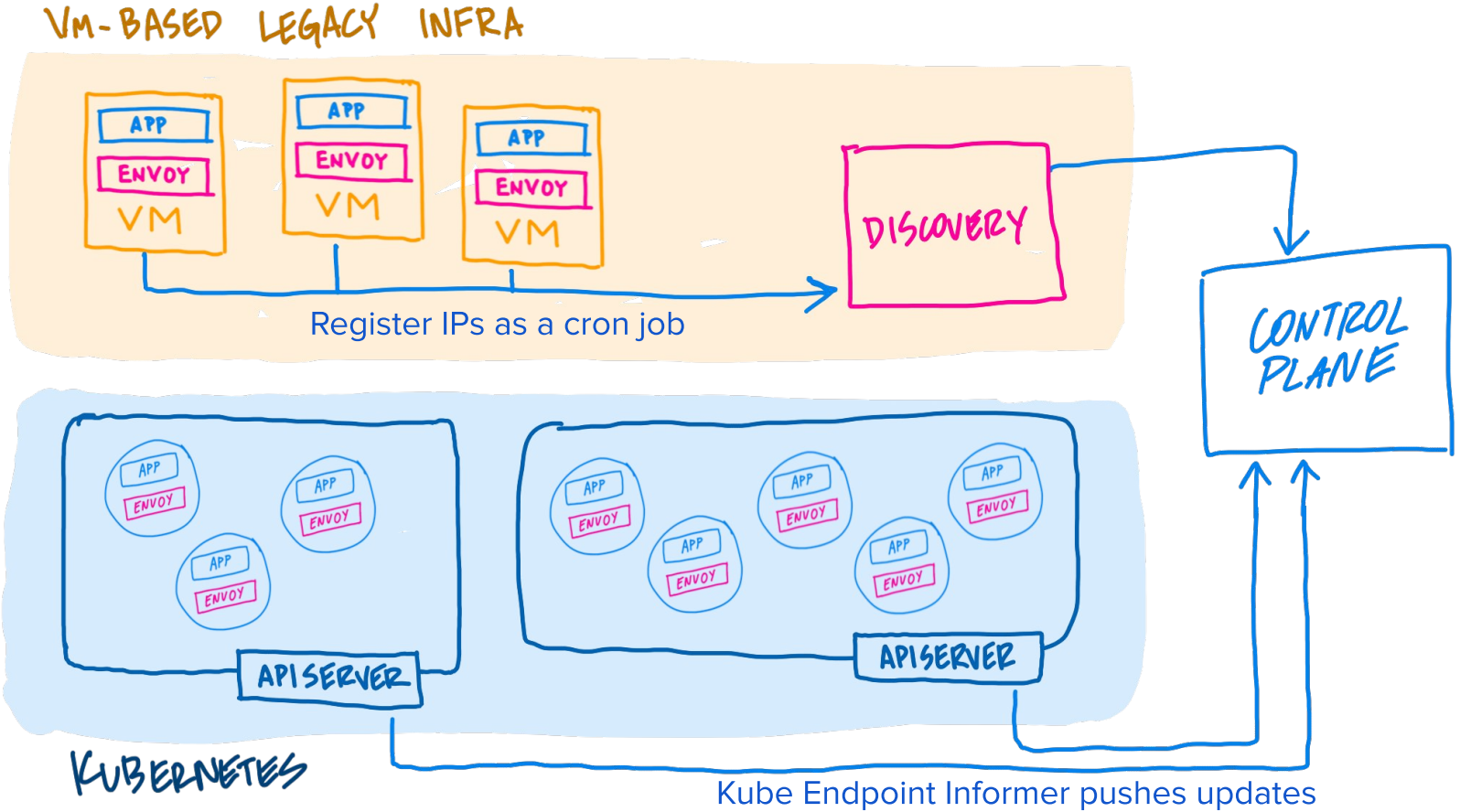
Pre-Kubernetes Service Discovery



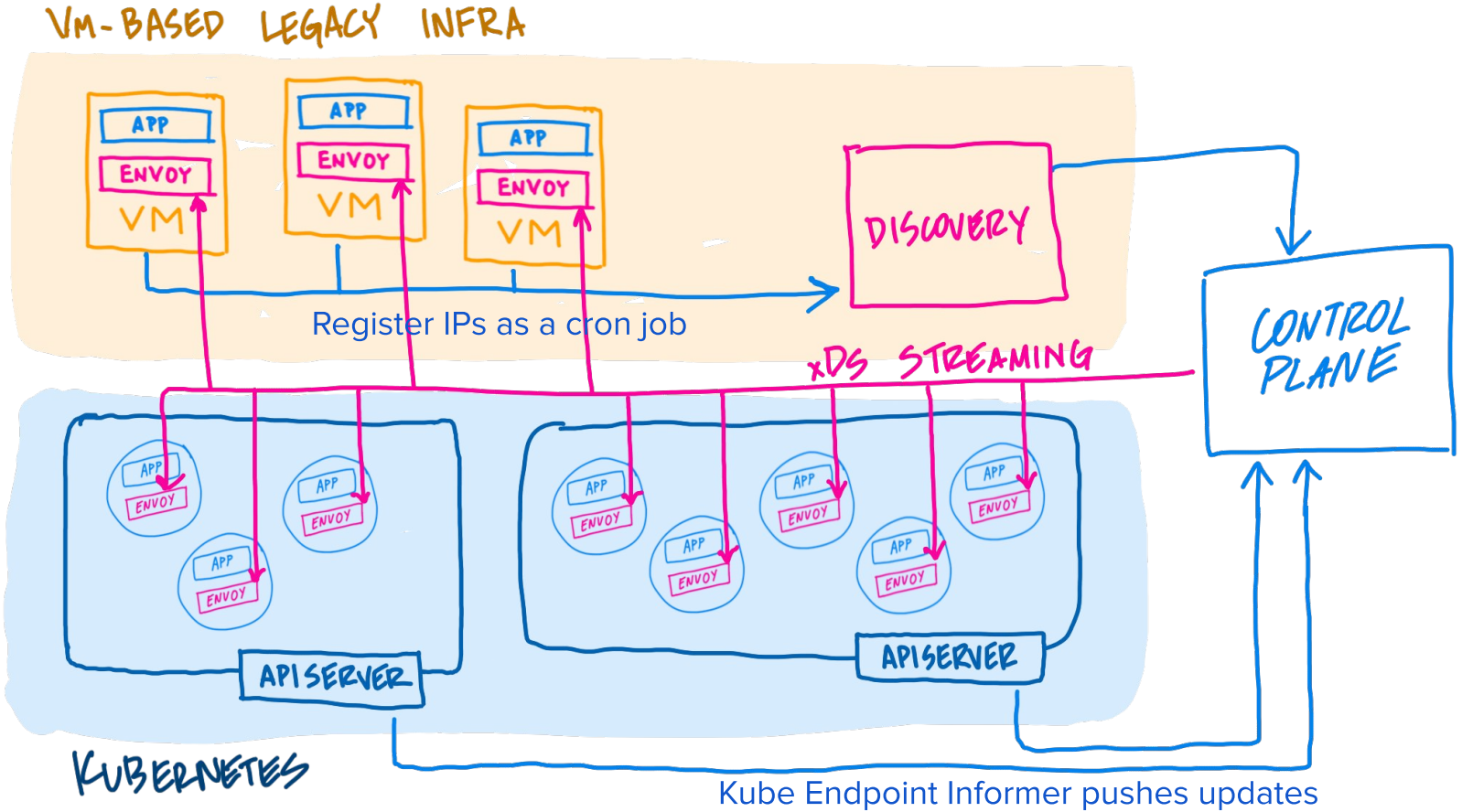
Pre-Kubernetes Service Discovery



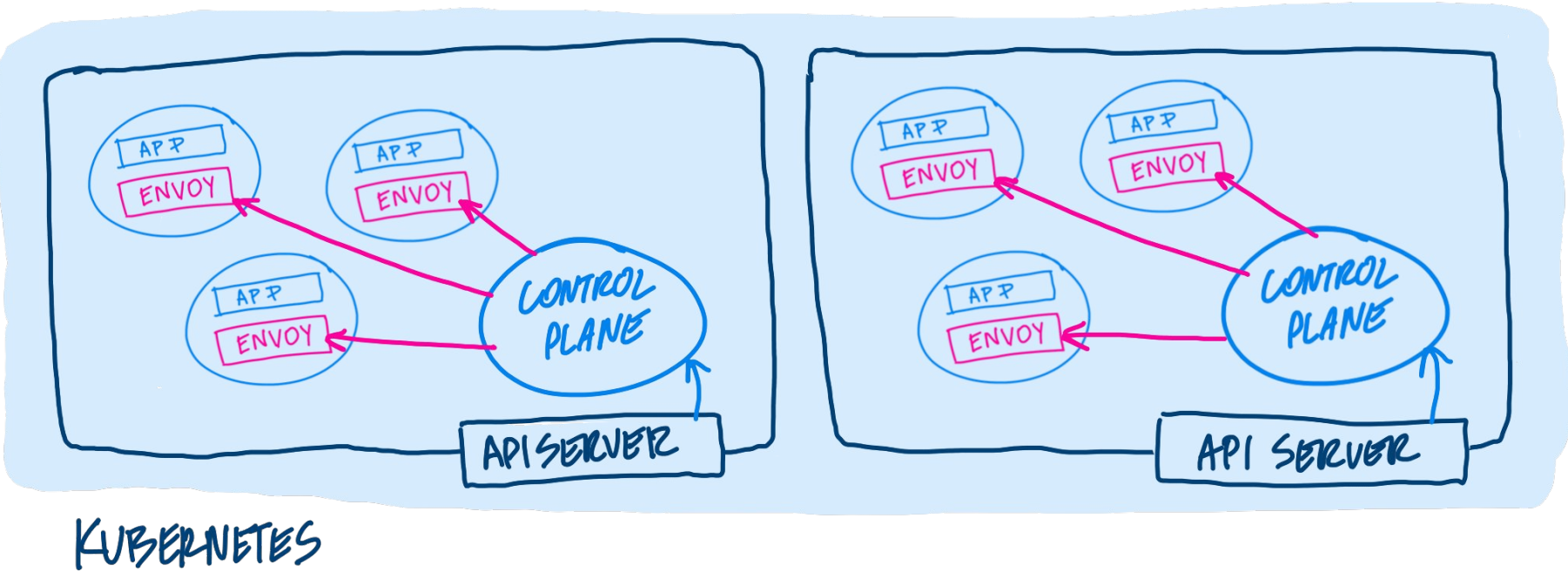
Hybrid Service Discovery



Hybrid Service Discovery



Final State


APP
ENVOY
APP
ENVOY
APP
ENVOY
CONTROL PLANE
API SERVER
APP
ENVOY
APP
ENVOY
APP
ENVOY
CONTROL PLANE
API SERVER
KUBERNETES

KEEPING IT SIMPLE

Initial Design

- Control Plane leverages Informers.
  - Uses WATCH APIs
- Kubernetes Endpoints object provide all IPs for a particular Service.
  - Provides old state and the new state. Either you can generate the diff, or in our case, just overwrite with the new state.
- Control Plane includes data from legacy Discovery API
- Moving to Streaming xDS APIs
  - Important since K8s changes very fast with immutable infrastructure.

Envoy CDS

Active Health Checking	<pre>"health_checks": [
	  {
	  "timeout": "2s",
	  "interval": "15s"
	  }
	…</pre>

Outlier Detection	<pre>"outlier_detection": {
	  "consecutive_5xx": 3,
	  "max_ejection_percent": 5,
	  "success_rate_minimum_hosts": 20,
	  "success_rate_request_volume": 50,
	  "success_rate_stdev_factor": 1900
	}</pre>

Panic Routing	Turned on by default. Enabled after 50% of hosts fail healthcheck

Issues

Every deployment is a scale up and scale down

We need to propagate this information across the mesh in time

Let’s explore each separately

Let’s explore each separately

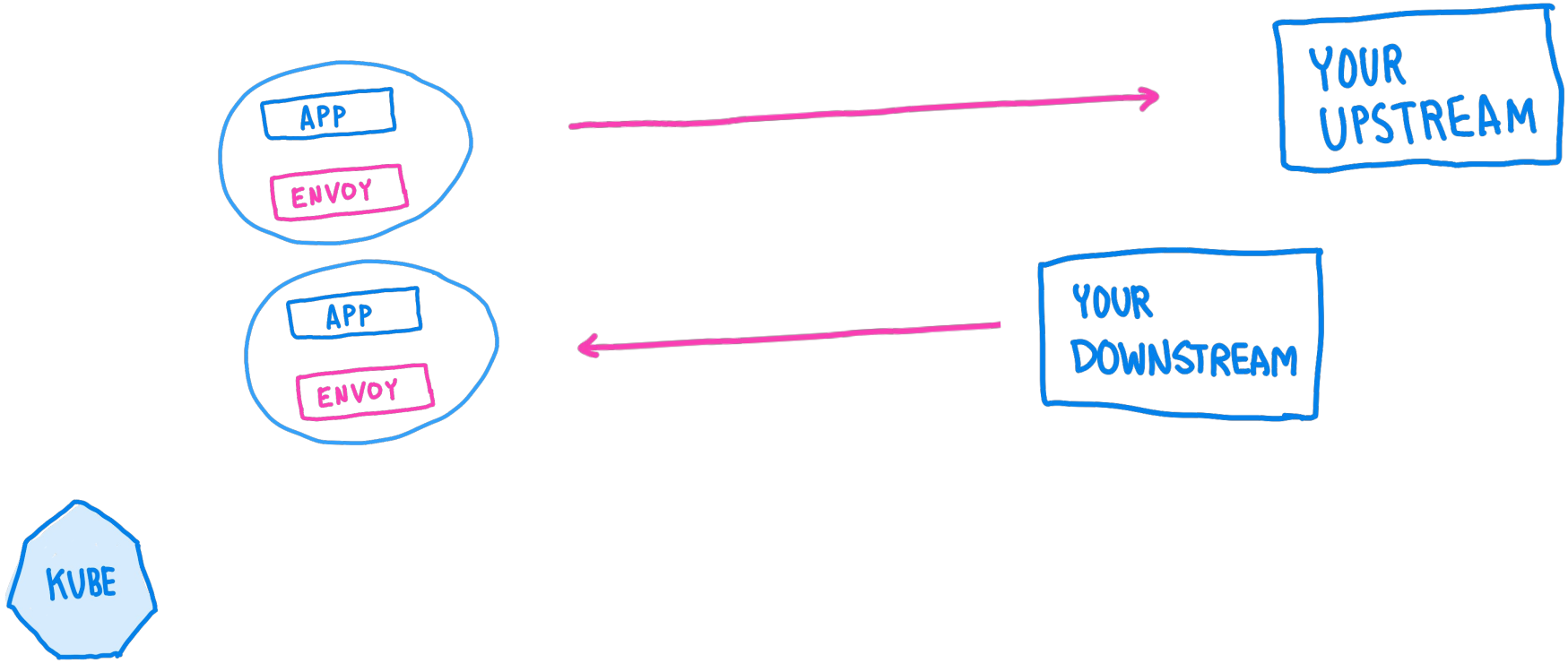
with an invitation to a party 

Scale Up

Scale Up: be polite and introduce yourself to everyone

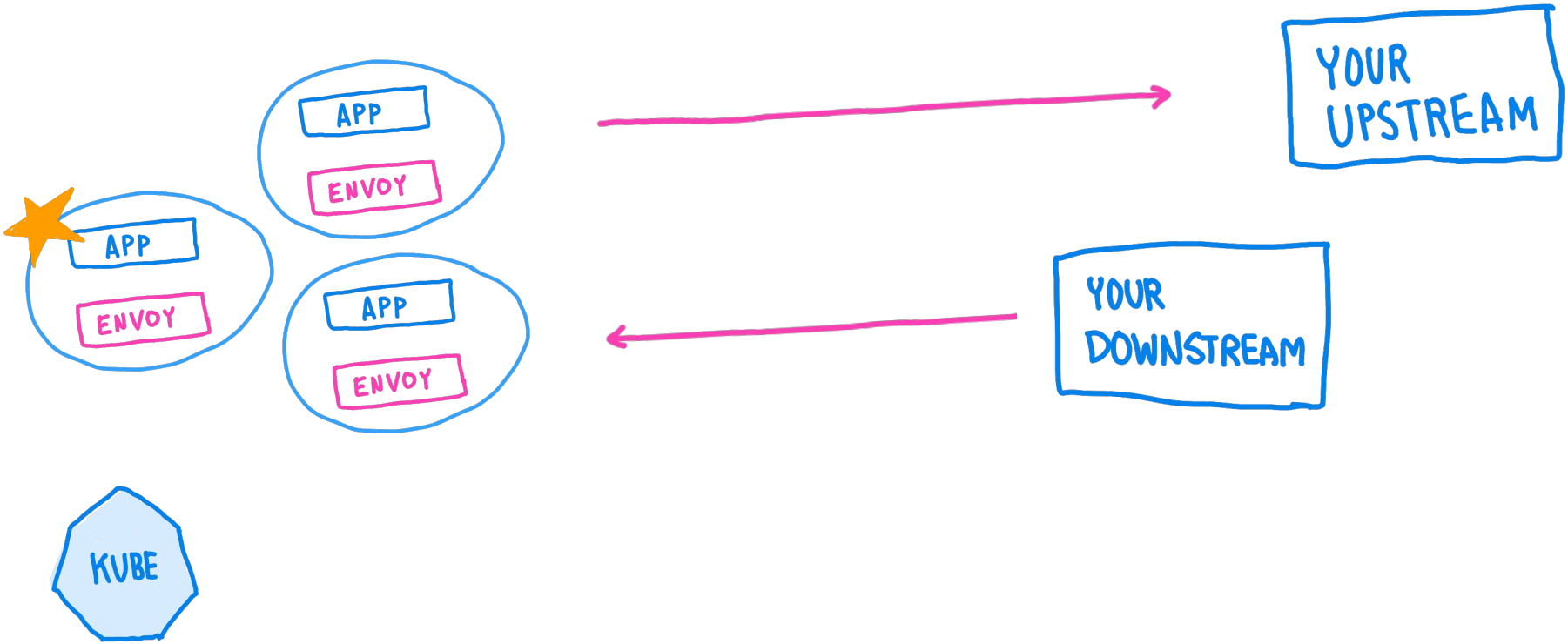
BE POLITE: SAY HELLO TO EVERYONE

Scale Up



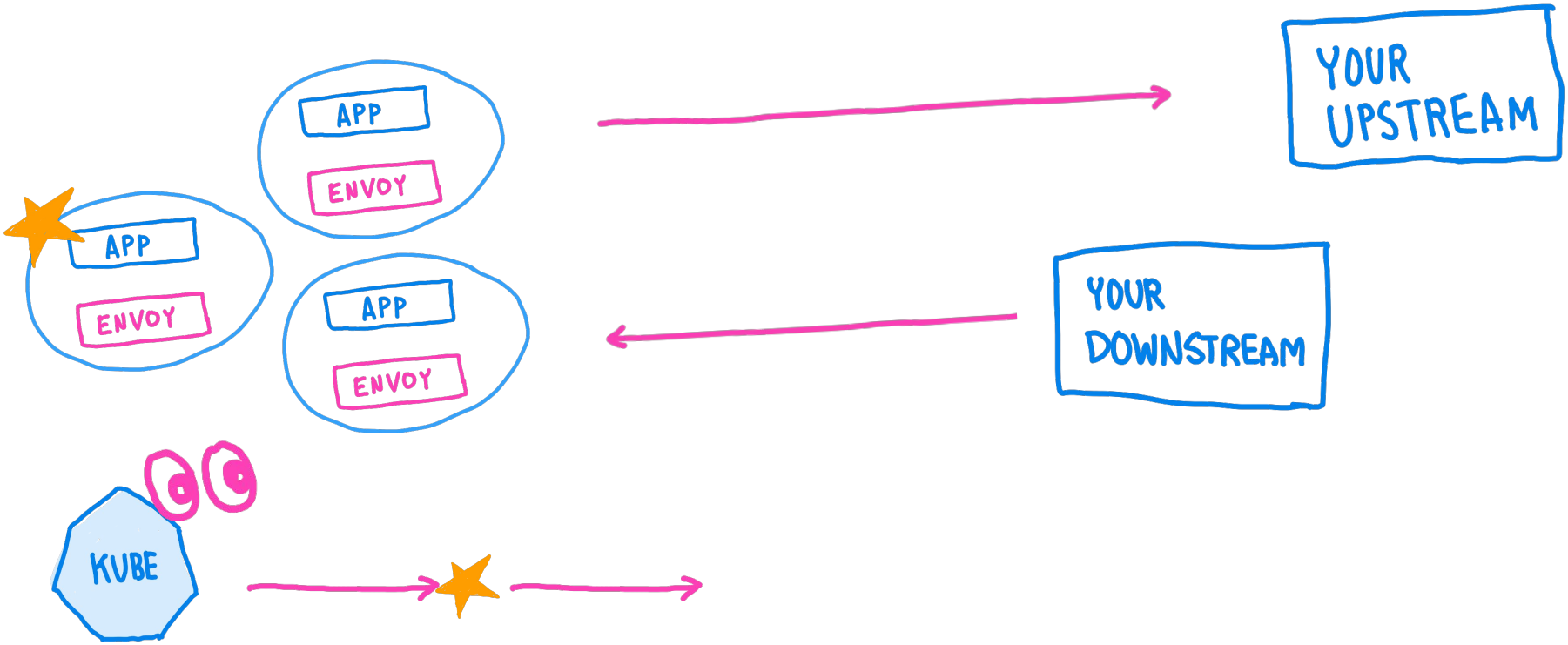
BE POLITE: SAY HELLO TO EVERYONE

Scale Up



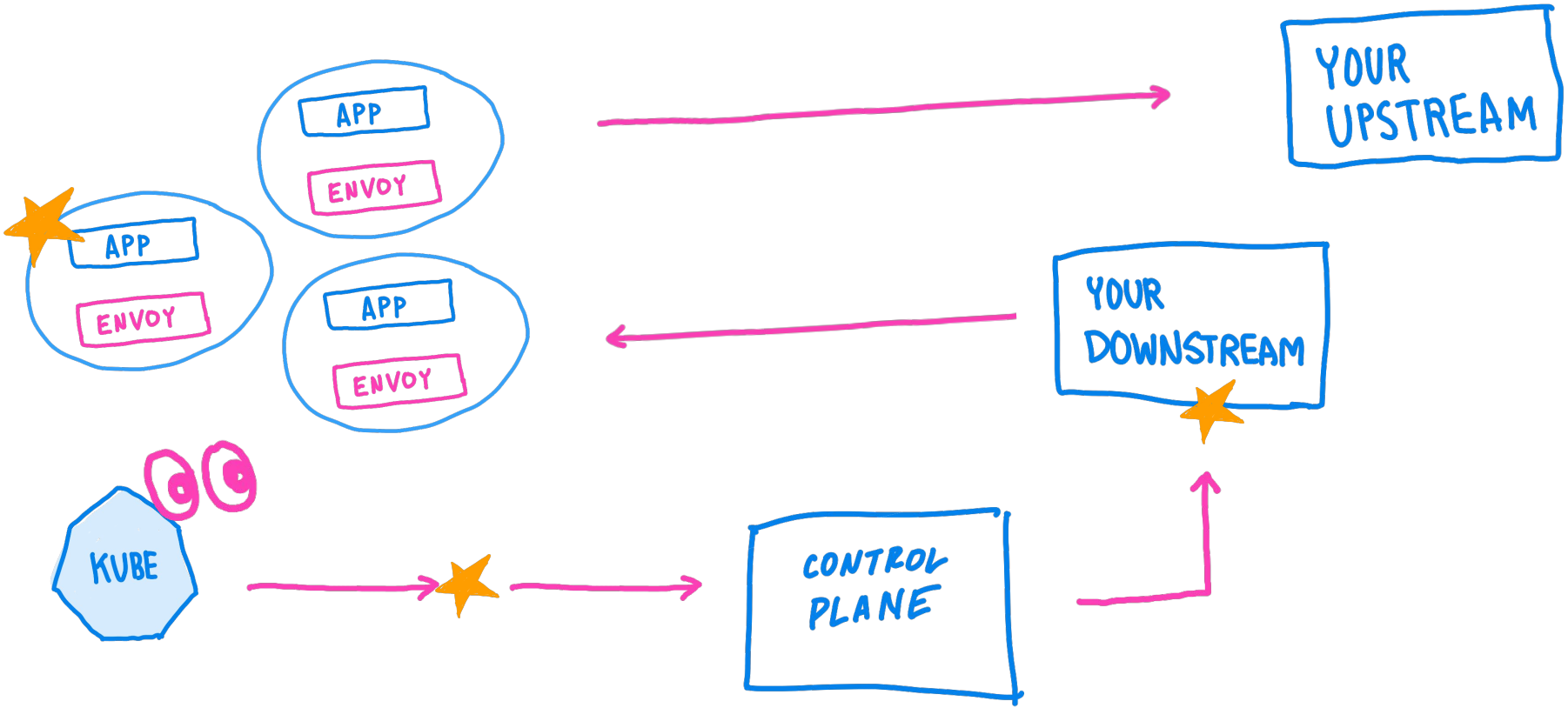
BE POLITE: SAY HELLO TO EVERYONE

Scale Up



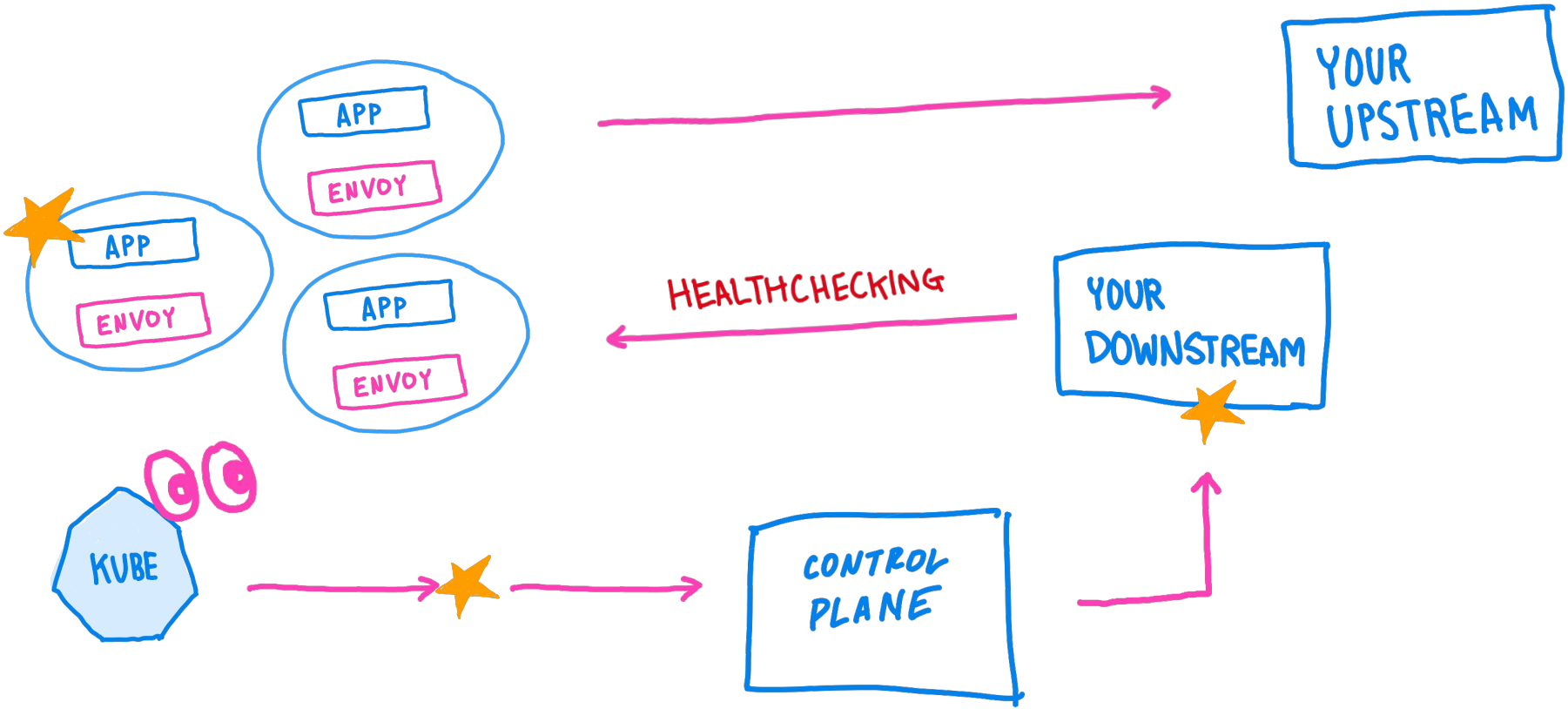
BE POLITE: SAY HELLO TO EVERYONE

Scale Up



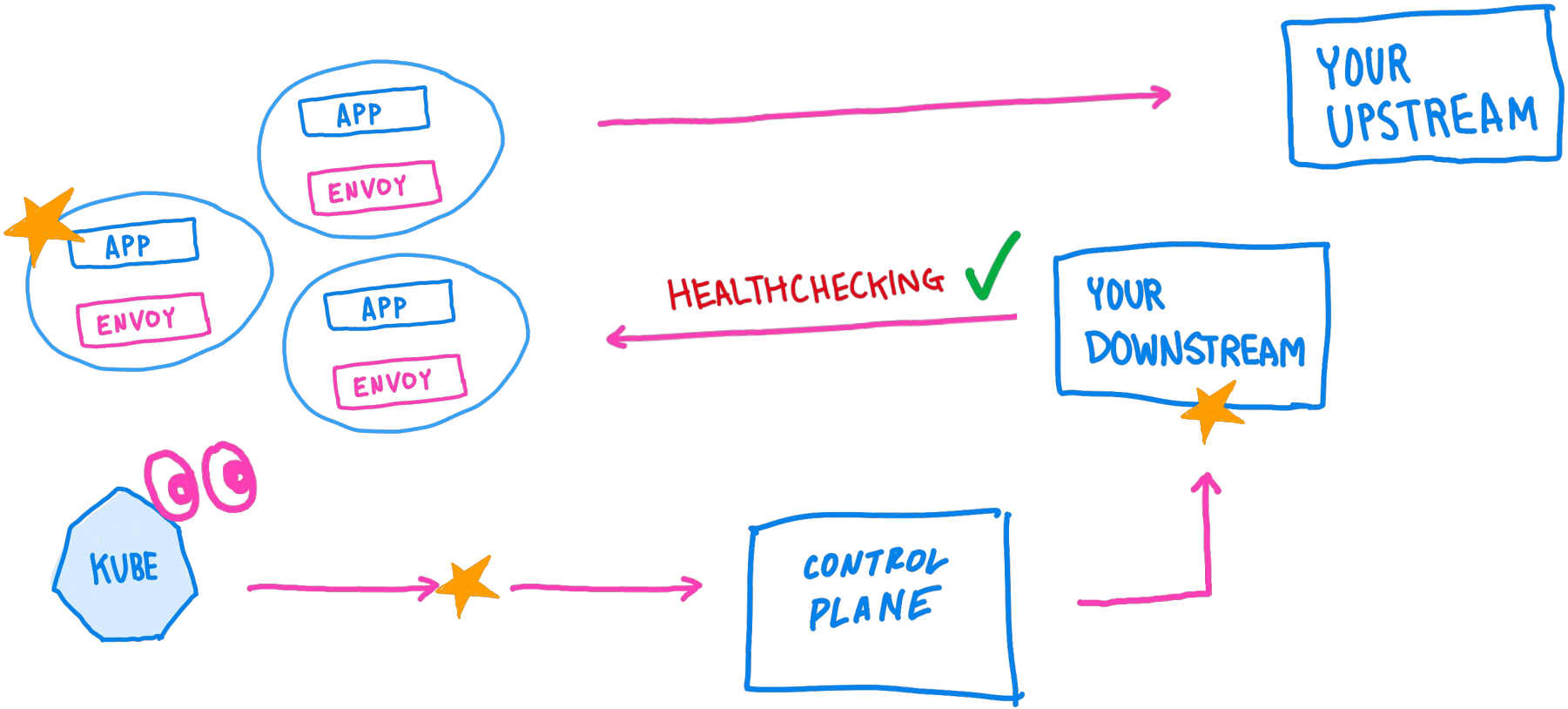
BE POLITE: SAY HELLO TO EVERYONE

Scale Up



BE POLITE: SAY HELLO TO EVERYONE

Scale Up



BE POLITE: SAY HELLO TO EVERYONE

Panic Routing

- Intended to prevent “thundering herd” effect.
- We have this turned on at 50%.
  - If 50% of the fleet is unhealthy, we will route requests to all hosts.
- **Issue:** A scale up >50% could cause unintentional panic routing.
  - New pods enter the mesh “unhealthy”, skewing the calculation.
  - Prevents us from taking advantage of Kubernetes’ faster scale-up.

BE POLITE: SAY HELLO TO EVERYONE

Scale Up Fix: Envoy Upstream

- Upstream: allow excluding hosts from load balancing calculations until initial health check (#6794) Thanks SnowP!
  - This works for both panic routing and priority spillover.
  - Has to be enabled via CDS

```
// CDS Configuration
bool ignore_new_hosts_until_first_hc = 5;
```

BE POLITE: SAY HELLO TO EVERYONE

Application & Envoy Startup

- Most applications at Lyft expect Envoy to be ready to serve traffic right away
- With our previous VM-based solution, Envoy was a long-lived process that was effectively always running.
  - Envoy is deployed in-place via hot-restarter
- In Kube, each new Pod means new Envoy
- **Issue:** New Envoy may have not connected to the Control Plane yet!
  - Applications experience Envoy failures when making outbound requests.

BE POLITE: SAY HELLO TO EVERYONE

Kubernetes & Sidecars

- In Kube, all containers in a Pod are started together.
- We could have applications wait for Envoy but would require widespread code changes
  - As the infra group, we intended for the migration to Kubernetes to migrate applications unchanged.

BE POLITE: SAY HELLO TO EVERYONE

Kubernetes & Sidecars

- In Kube, all containers in a Pod are started together.
- We could have applications wait for Envoy but would require widespread code changes.
  - As the infra group, we intended for the migration to Kubernetes to migrate applications unchanged.

BE POLITE: SAY HELLO TO EVERYONE

Kubernetes & Sidecars

- Envoy
  - /ready admin endpoint serves as readiness
  - Be careful with initialization timeout
- We maintain a fork of Kubernetes with some patches:
  - At start, sidecars must be ready first before applications
  - At termination, applications exit first before sidecars
- Mostly follows KEP: sig-apps/sidecarcontainers.md
  - Working with Pinterest & the community to figure out a possible upstream solution.
  - https://github.com/lyft/kubernetes - branch: release-1.14.7-lyft

BE POLITE: SAY HELLO TO EVERYONE

Summary - Scale Up

- **Panic Routing**
  - Upstream Envoy fix prevents large scale-ups triggering unintentional panic routing.
- **Kubernetes' Sidecar Support**
  - Readiness means "I've connected to the control plane"
  - Our forked Kubernetes ensures Envoy is ready before the application starts.

BE POLITE: SAY HELLO TO EVERYONE

Summary - Scale Up

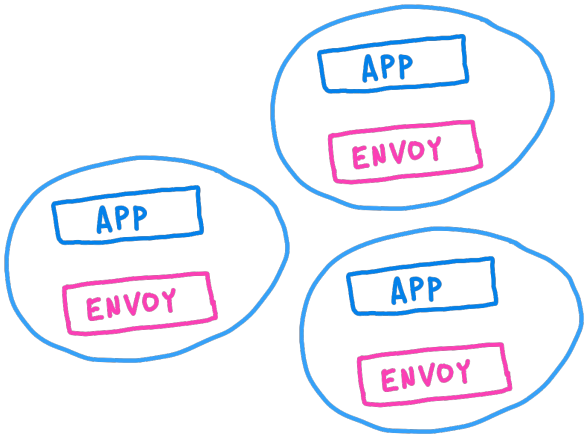
- **Panic Routing**
  - Upstream Envoy fix prevents large scale-ups triggering unintentional panic routing.
- **Kubernetes' Sidecar Support**
  - Readiness means "I've connected to the control plane"
  - Our forked Kubernetes ensures Envoy is ready before the application starts

Scale Down

Scale Down: Say bye to everyone before you leave

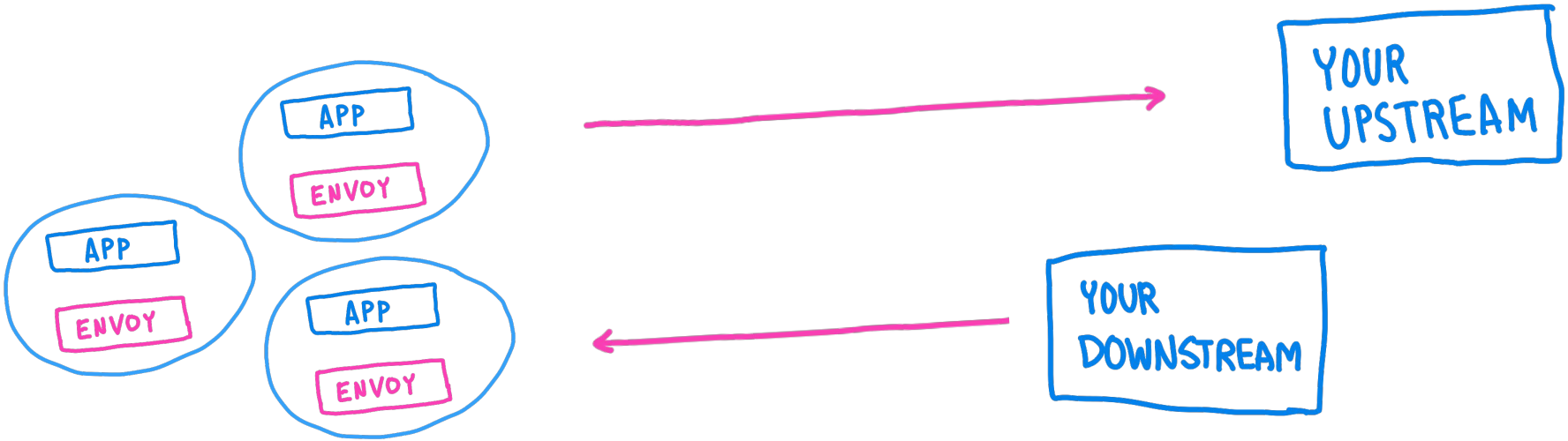
SAY BYE TO EVERYONE BEFORE YOU LEAVE

Scale Down



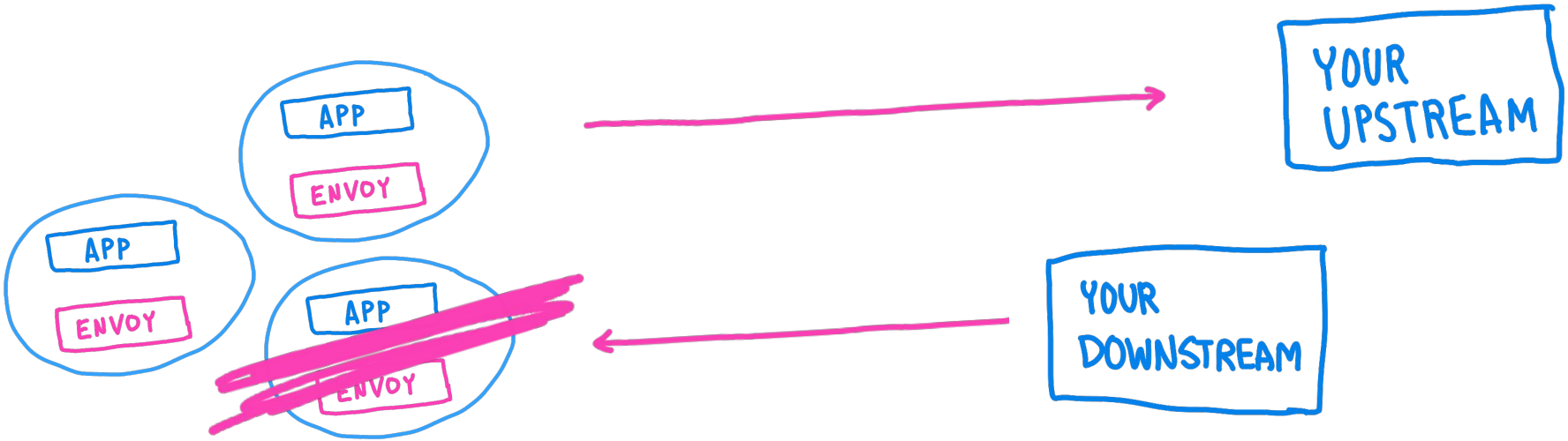
SAY BYE TO EVERYONE BEFORE YOU LEAVE

Scale Down



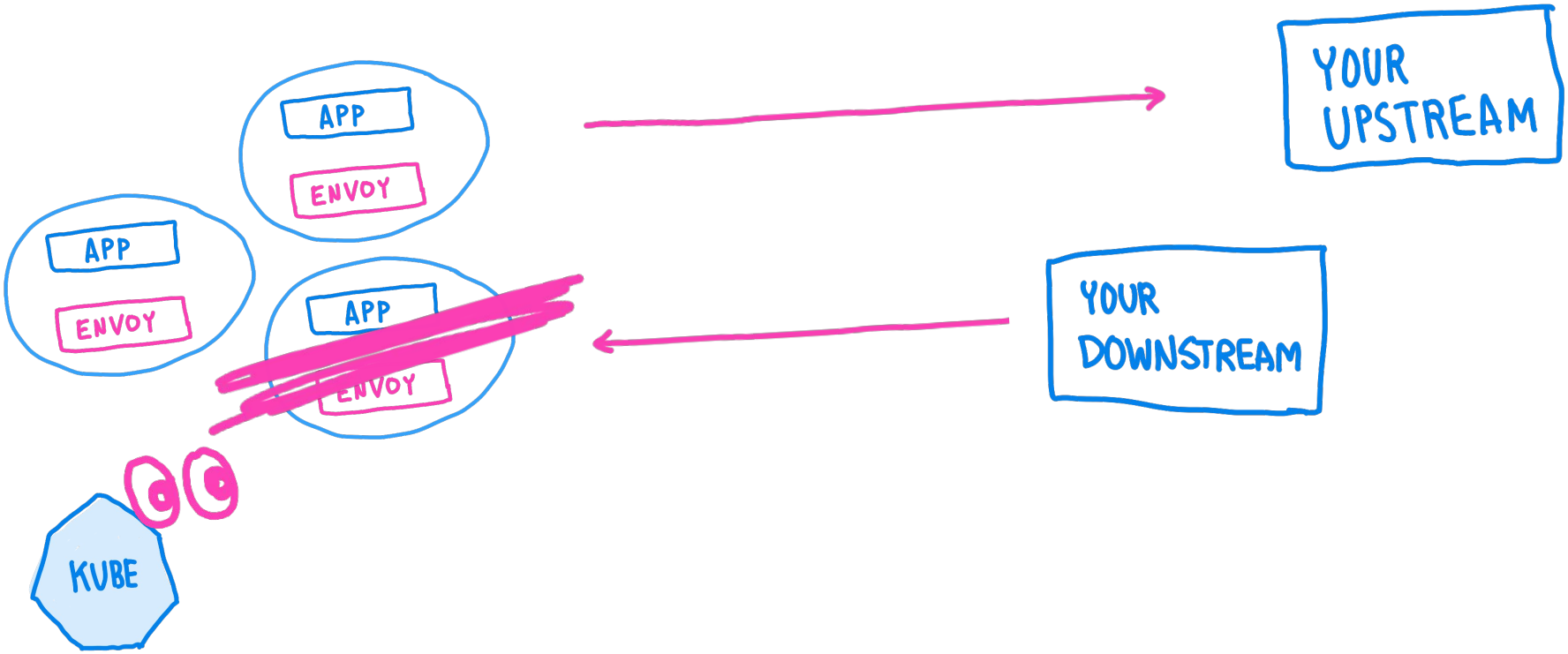
SAY BYE TO EVERYONE BEFORE YOU LEAVE

Scale Down



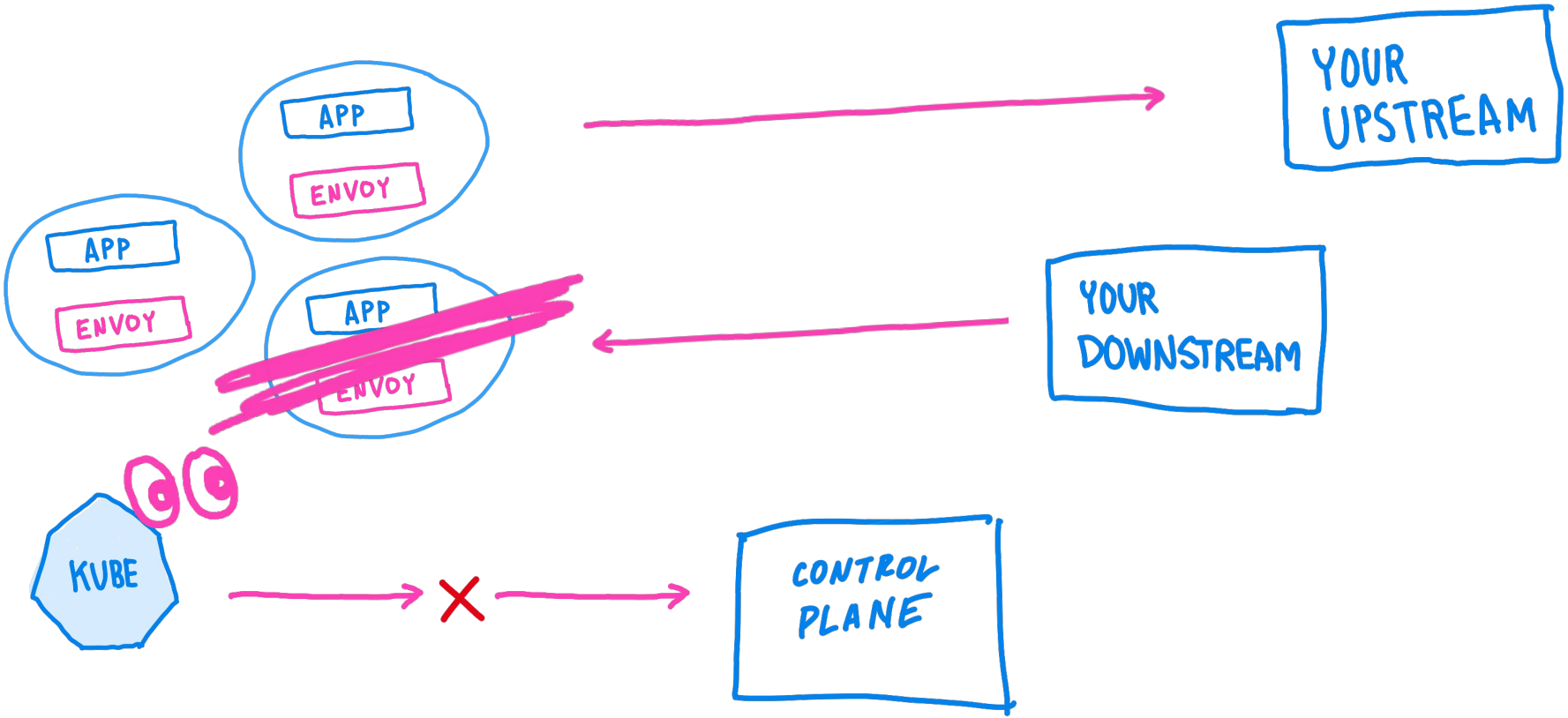
SAY BYE TO EVERYONE BEFORE YOU LEAVE

Scale Down



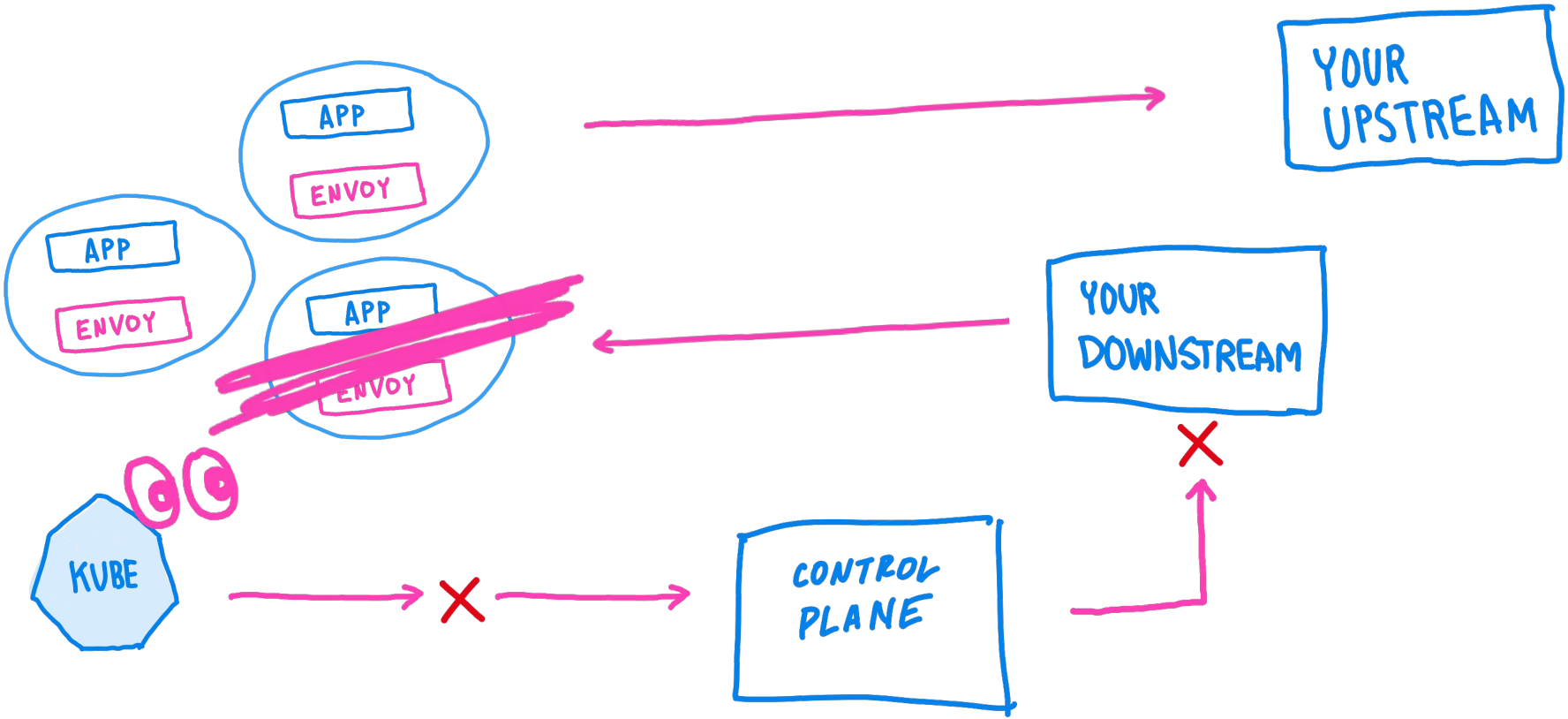
SAY BYE TO EVERYONE BEFORE YOU LEAVE

Scale Down



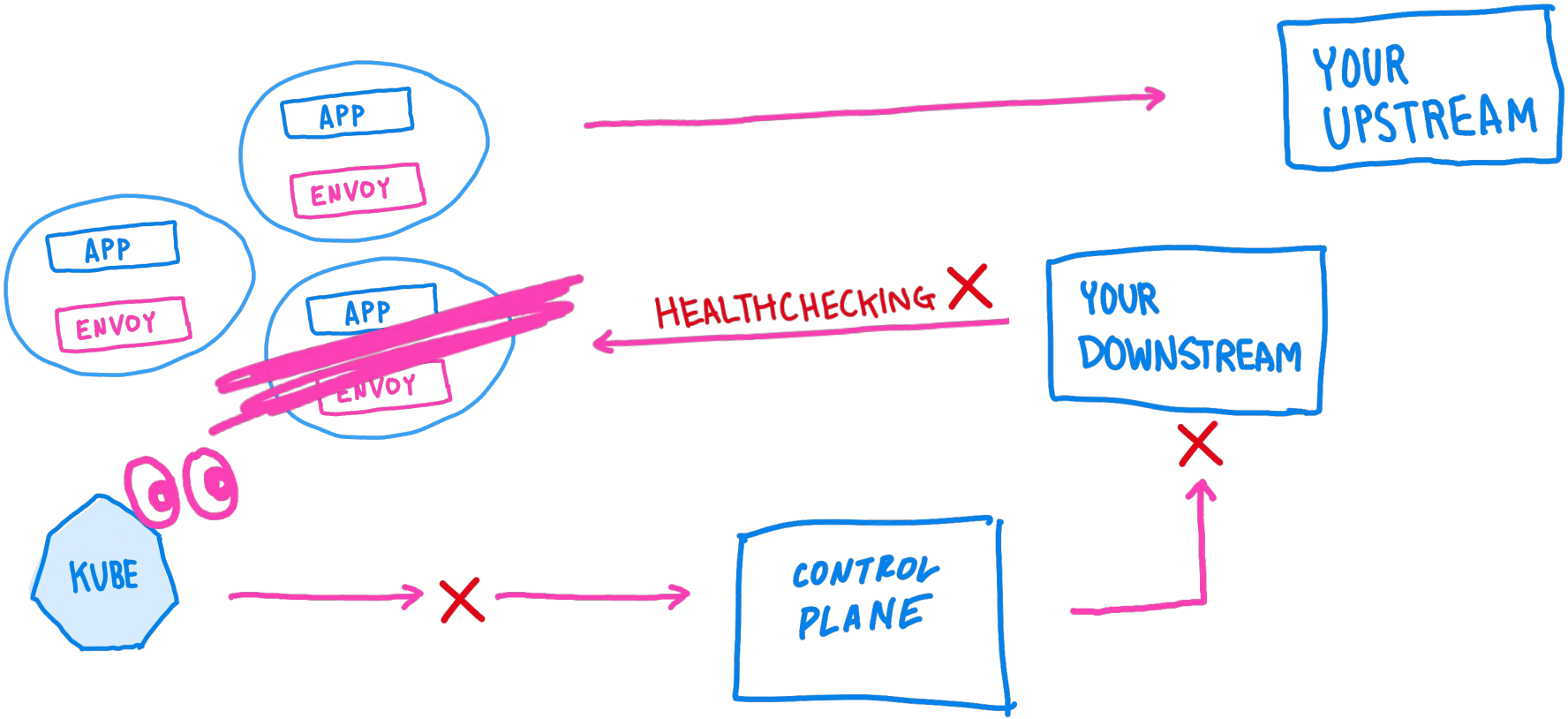
SAY BYE TO EVERYONE BEFORE YOU LEAVE

Scale Down



SAY BYE TO EVERYONE BEFORE YOU LEAVE

Scale Down



SAY BYE TO EVERYONE BEFORE YOU LEAVE

Undead Hosts

- Pods would be terminated, but their downstreams would keep these “undead” hosts in local discovery.
  - This caused service owners to be paged for having too many unhealthy hosts.
  - **The mesh was not converging correctly.**

SAY BYE TO EVERYONE BEFORE YOU LEAVE

Scale Down Issues

- Envoy protects from byzantine failures against the xDS Control Plane
  - Doesn't remove hosts from "local discovery" until they:
    - Fail healthchecks
    - Are absent from the Control Plane's EDS response
- ***In order!***
- We had this happen out of order, **accumulating stale entries until another update.**
- With xDS streaming, **you could have garbage IP addresses depending on update timing.**

SAY BYE TO EVERYONE BEFORE YOU LEAVE

Scale Down Issues

Discovery Status	Health Check OK	Health Check Failed
Discovered	Route	Don’t Route
Absent	Route	Don’t Route / Delete

SAY BYE TO EVERYONE BEFORE YOU LEAVE

Scale Down Issues

Discovery Status	Health Check OK	Health Check Failed
Discovered	Route	Don't Route
Absent	Route	Don't Route / Delete

SAY BYE TO EVERYONE BEFORE YOU LEAVE

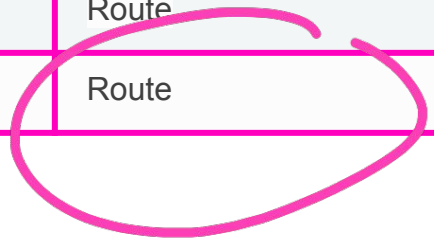
Scale Down Issues

Discovery Status	Health Check OK	Health Check Failed
Discovered	Route	Don't Route
Absent	Route	Don't Route / Delete

SAY BYE TO EVERYONE BEFORE YOU LEAVE

Scale Down Issues

Discovery Status	Health Check OK	Health Check Failed
Discovered	Route	Don't Route
Absent	Route	Don't Route / Delete



SAY BYE TO EVERYONE BEFORE YOU LEAVE

Envoy Upstream

- Upstream: do not stabilize host when failed by EDS (#6714)
  - Allows control plane to forcibly remove host from mesh
    - Control Plane can send DRAINING state first to forcibly remove host from mesh
- Upstream: handle health check fail after removal (#6765)
  - Removes ordering requirement for health checking
  - This change makes envoy store the state of the last discovery update

**Thanks mklein for implementing both of these code changes!

SAY BYE TO EVERYONE BEFORE YOU LEAVE

Scale Down Issues

Discovery Status	Health Check OK	Health Check Failed
Discovered	Route	Don’t Route
Absent	Route	Don’t Route / Delete

SAY BYE TO EVERYONE BEFORE YOU LEAVE

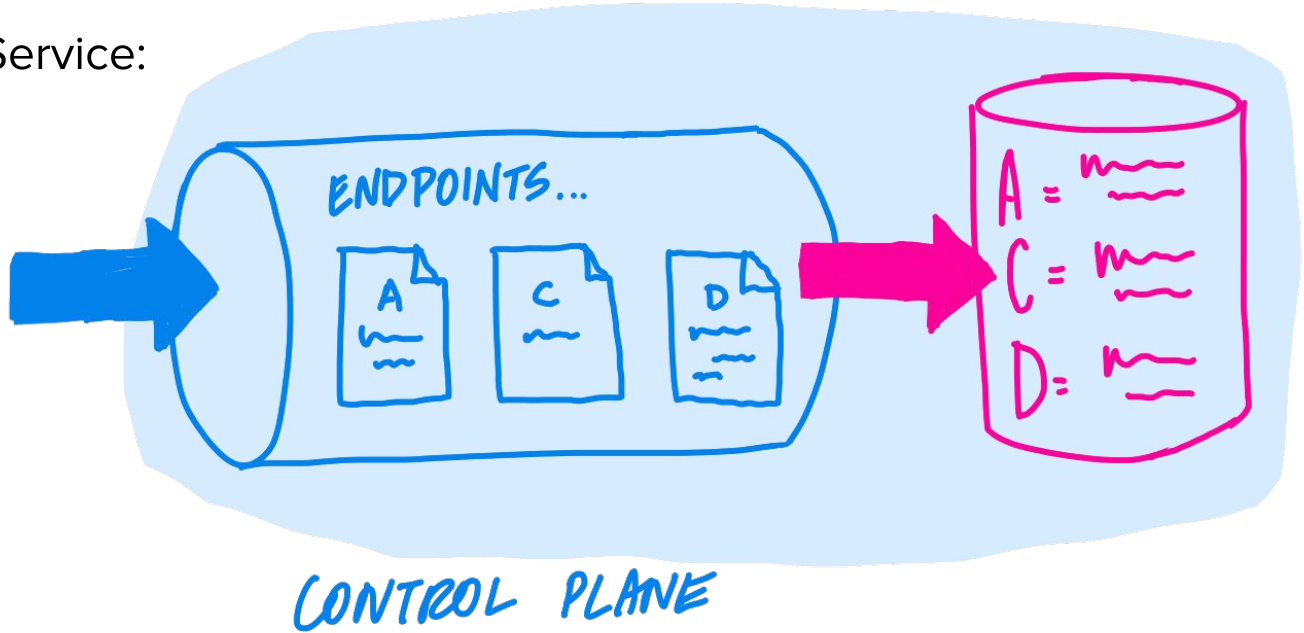
Even removing ordering, still seeing 503s

- Even though Envoy can handle out of order healthchecks, we still saw 503s.
  - We can rely on retries, but, we wanted to do better!
- This is because Endpoints are deleted at the same time the healthcheck fails.
- **Issue: How do we notify the mesh that a host is is going away to avoid unnecessary 503s?**

SAY BYE TO EVERYONE BEFORE YOU LEAVE

Kubernetes Endpoints

- Kubernetes Endpoints object tracks IPs in a Service.
- Informer is pretty easy to set up, every update contains all IPs for a given Service:



SAY BYE TO EVERYONE BEFORE YOU LEAVE

Kubernetes Endpoints

- Endpoints they are **immediately** removed once readiness fails.
  - Not much time to send a DRAINING update.
  - There could be a race**, causing small amount of 503s** when a pod is terminated ***before*** **the EDS removal** makes it to downstreams, but before active health check.

SAY BYE TO EVERYONE BEFORE YOU LEAVE

Kubernetes Endpoints

- Pods are the next logical alternative:
  - **A Pod's Termination Grace Period provides the window to send a DRAINING update.**
  - **Pod updates are not Service-level, they are individual updates. Yuck!**

SAY BYE TO EVERYONE BEFORE YOU LEAVE

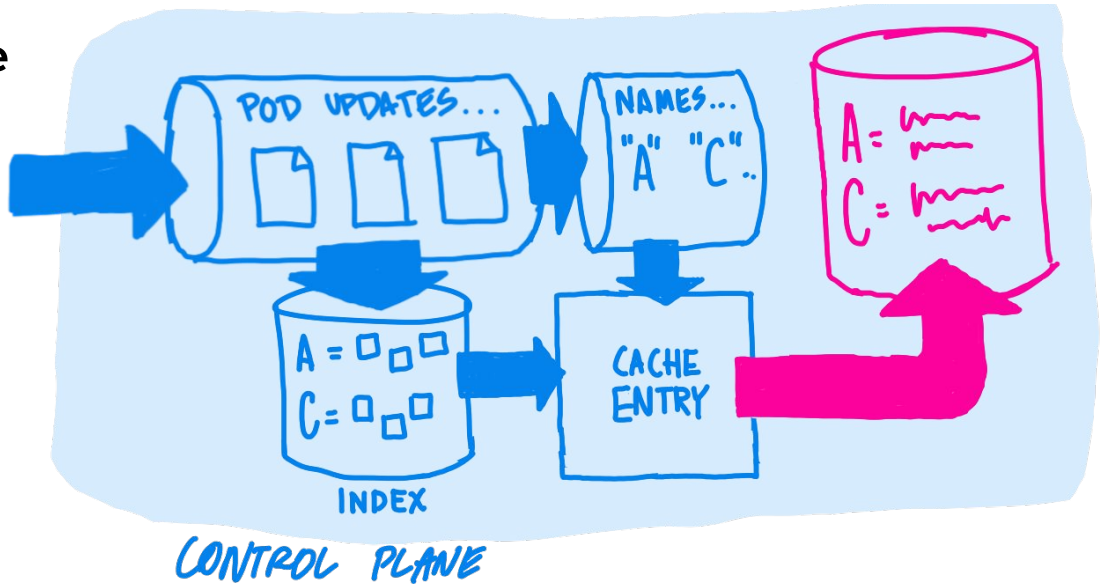
Kubernetes Endpoints

- Pods are the next logical alternative:
  - **A Pod's Termination Grace Period provides the window to send a DRAINING update.** 
  - **Pod updates are not Service-level, they are individual updates. Yuck!** 

SAY BYE TO EVERYONE BEFORE YOU LEAVE

Kubernetes Indexer for Pods

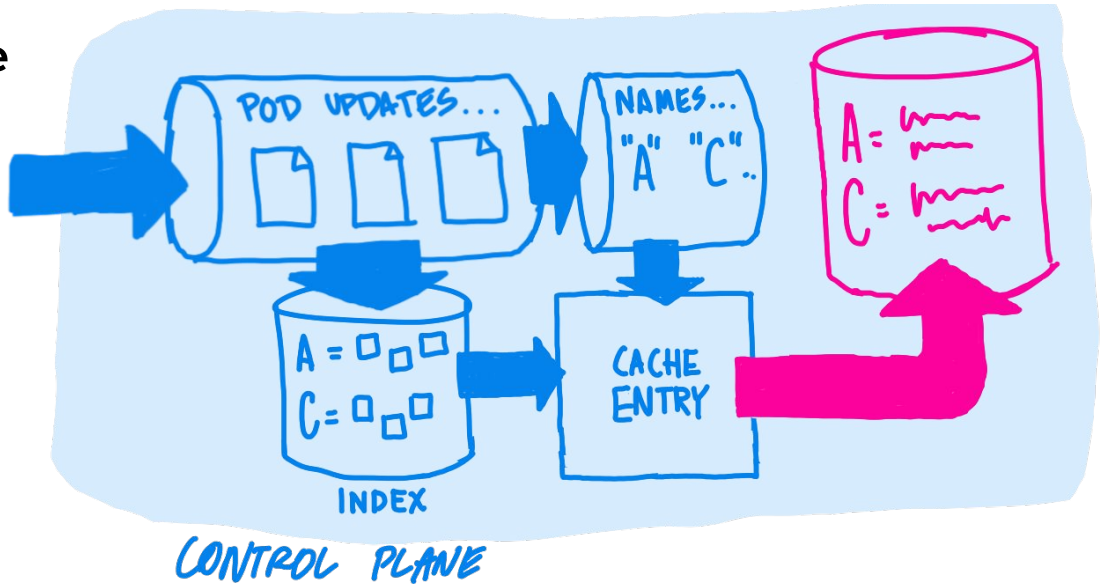
- Indexers allow you to bucket the watched objects in arbitrary ways.
- All of our Pods are labeled by service name, therefore:
- **We index by service name: get all pods for a given service in one update!**



SAY BYE TO EVERYONE BEFORE YOU LEAVE

Kubernetes Indexer for Pods

- Indexers allow you to bucket the watched objects in arbitrary ways.
- All of our Pods are labeled by service name, therefore:
- **We index by service name: get all pods for a given service in one update!**



SAY BYE TO EVERYONE BEFORE YOU LEAVE

Summary - Scale Down

- **Envoy can handle out-of-order discovery and healthcheck data**
  - Control plane can force host removal by sending a DRAINING update.
  - Envoy can resolve local discovery without healthcheck failing first
- **Control Plane monitors Pods**
  - Control Plane updates downstreams the moment a Pod is scheduled to be terminated.
  - Pods can safely drain requests before they exit.

SAY BYE TO EVERYONE BEFORE YOU LEAVE

Scale Down

- **Envoy can handle out-of-order discovery and healthcheck data**
  - Control plane can force host removal by sending a DRAINING update.
  - Envoy can resolve local discovery without healthcheck failing first
- **Control Plane monitors Pods**
  - Control Plane updates downstreams the moment a Pod is scheduled to be terminated.
  - Pods can safely drain requests before they exit

In Summary

PARADIGM SHIFT

Summary

- Paradigm shift moves most changes from Node-level to the mesh topography.
  - Take advantage of Kubernetes: quick scale up/down
- Few things needed tweaking:
  - Envoy upstream improvements to work with streaming
  - Kubernetes' Sidecar Ordering
  - Monitoring Pods, not Endpoints

THE FINAL FRONTIER?

Future Work

- Envoy runs as a sidecar on every Pod — Expensive!
  - Envoy as DaemonSet seems very interesting to us. But complex!
- Right now we have one big mesh, we’re looking into splitting them up
  - AWS Availability Zone
- Batch updates with big scaling up and down events in our control plane.
- Splitting envoy control plane / mesh
  - Envoy OSS moving to federation

Thank You!

Thank You!

Now we’re inviting you to a party  ...

Join us for some local beer, wine, and tacos!

Lyft Happy Hour

Date: Tuesday, Nov 19
Time: 7pm-10pm
Where: Thorn Barrio Logan (1745 National Avenue, San Diego, CA 92113)

RSVP: https://lyft-kubecon.splashthat.com/ (you can also register at the door)