

RESILIENCE
REALIZED



KubeCon



CloudNativeCon

North America 2021

Migrating to Service Mesh at Scale to Support Billions of \$ Transactions

@imrenagi & @girikuncoro from  **gopay**

Intro



 @imrenagi



 @girikuncoro

RESILIENCE
REALIZED



KubeCon



CloudNativeCon

North America 2021



gopay

Case Study



- Has **largest MAU** in
Indonesia since Q4 2017



- Has **largest MAU** in Indonesia since Q4 2017
- Accepted at **700,000+** online and offline merchants



- Has **largest MAU** in Indonesia since Q4 2017
- Accepted at **700,000+** online and offline merchants
- Has integrations with **28+** financial institutions



GoPay Engineering



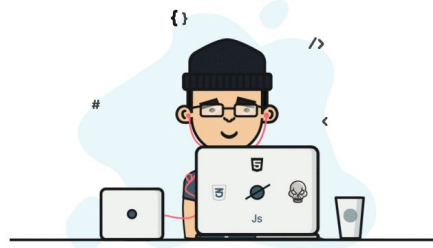
KubeCon



CloudNativeCon

North America 2021

- 230+ developers



GoPay Engineering



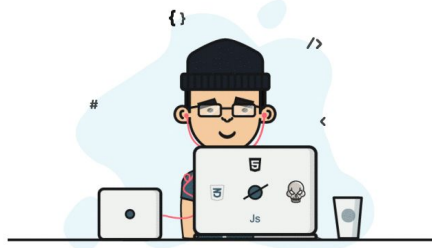
KubeCon



CloudNativeCon

North America 2021

- 230+ developers
- 30+ teams



GoPay Engineering



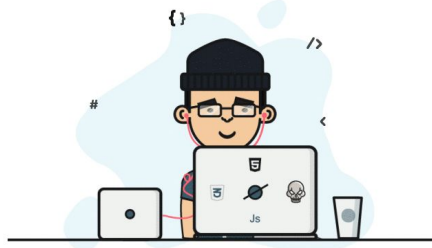
KubeCon



CloudNativeCon

North America 2021

- 230+ developers
- 30+ teams
- 30 Kubernetes clusters



GoPay Engineering



KubeCon



CloudNativeCon

North America 2021

- 230+ developers
- 30+ teams
- 30 Kubernetes clusters
- 3000+ deployments / week



Kubernetes Migrations

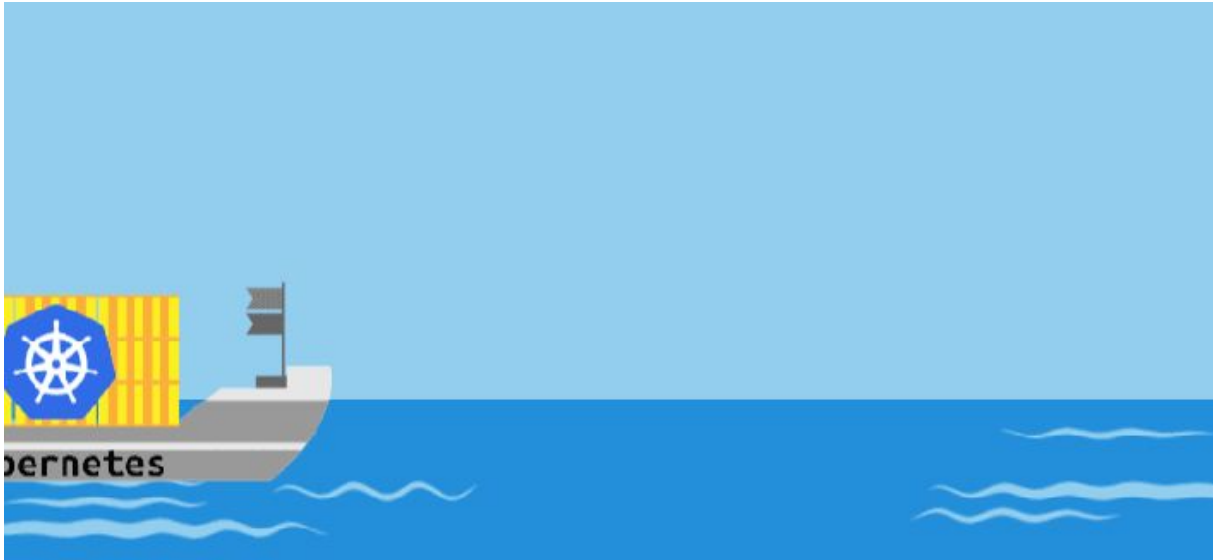


KubeCon



CloudNativeCon

North America 2021



Moving to Kubernetes in late 2018

Kubernetes Migrations

After 2 years ...

After 2 years ...

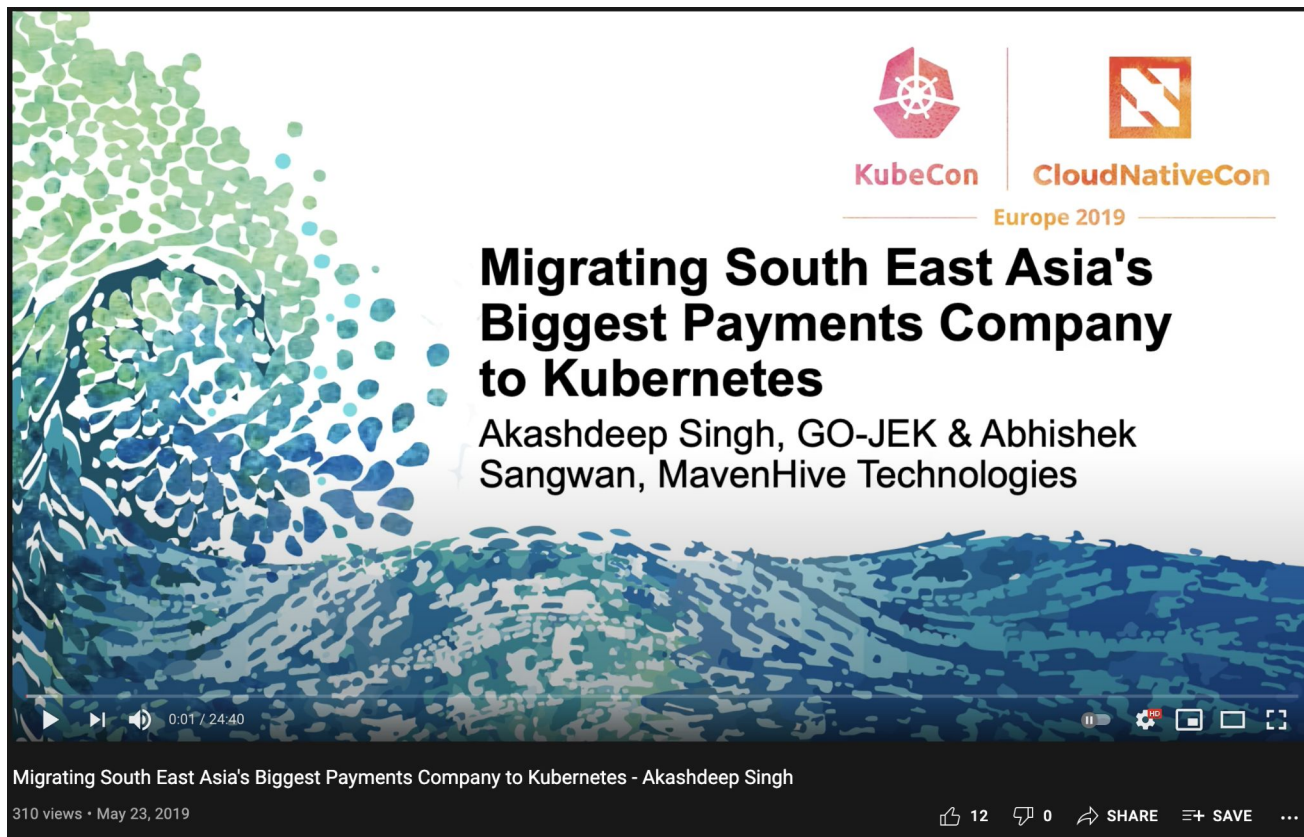
100% of services
in Kubernetes

Kubernetes Migrations

After 2 years ...

Almost 1,000 of apps
in Kubernetes

Kubernetes Migrations



The video player interface shows a presentation slide with the KubeCon and CloudNativeCon logos at the top, followed by 'Europe 2019'. The main title is 'Migrating South East Asia's Biggest Payments Company to Kubernetes', and the speakers are 'Akashdeep Singh, GO-JEK & Abhishek Sangwan, MavenHive Technologies'. The slide features a background illustration of a mountain range with a green dotted pattern on the left. The video player controls at the bottom show a progress bar at 0:01 / 24:40, and the video title 'Migrating South East Asia's Biggest Payments Company to Kubernetes - Akashdeep Singh' with 310 views from May 23, 2019. Engagement icons for likes (12), comments (0), share, save, and a menu are also visible.

KubeCon | CloudNativeCon
Europe 2019

Migrating South East Asia's Biggest Payments Company to Kubernetes

Akashdeep Singh, GO-JEK & Abhishek Sangwan, MavenHive Technologies

0:01 / 24:40

Migrating South East Asia's Biggest Payments Company to Kubernetes - Akashdeep Singh

310 views • May 23, 2019

12 0 SHARE SAVE ...

Source: <https://youtu.be/eYb--4iOSCY>

Istio Migrations

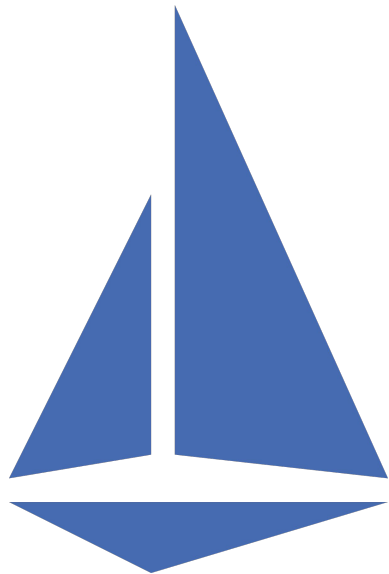


KubeCon



CloudNativeCon

North America 2021



Sailing with Istio since early 2020

Istio Migrations



 KubeCon |  CloudNativeCon
Europe 2020

Virtual

Migrating Transactions Worth Billions of \$ to Service Mesh With No Downtime

Mahendra Kariya & Shishir Joshi, Gojek

0:00 / 34:48

Migrating Transactions Worth Billions of \$ to Service Mesh With N... Mahendra Kariya & Shishir Joshi

178 views • Sep 4, 2020

6 1 SHARE SAVE ...

Source: <https://youtu.be/OY0CxiKbX2o>

Istio Migrations

After 1 year ...

After 1 year ...

< 2% of services
in Istio

Past Migrations: Lessons Learned



KubeCon



CloudNativeCon

North America 2021

Past Migrations: Lessons Learned

- **Steep learning curve** for developers

Past Migrations: Lessons Learned

- **Steep learning curve** for developers
- **Frictions** in keeping
standardized templates **up to date**

Past Migrations: Lessons Learned



KubeCon



CloudNativeCon

North America 2021

- **Steep learning curve** for developers
- **Frictions** in keeping
standardized templates **up to date**
- New approach **is not default**

Past Migrations: Lessons Learned

- Very high **migration overhead**

Past Migrations: Lessons Learned

- Very high **migration overhead**
 - **Exponentially** more services and edges

Past Migrations: Lessons Learned



KubeCon



CloudNativeCon

North America 2021

- Very high **migration overhead**
 - **Exponentially** more services and edges
 - **Complexity** with different use cases

Past Migrations: Lessons Learned



KubeCon



CloudNativeCon

North America 2021

- Very high **migration overhead**
 - **Exponentially** more services and edges
 - **Complexity** with different use cases
 - **Maintaining mixed state** of infrastructure

Past Migrations: Lessons Learned



KubeCon



CloudNativeCon

North America 2021

- **Unclear ownership of services**



What Developers Want

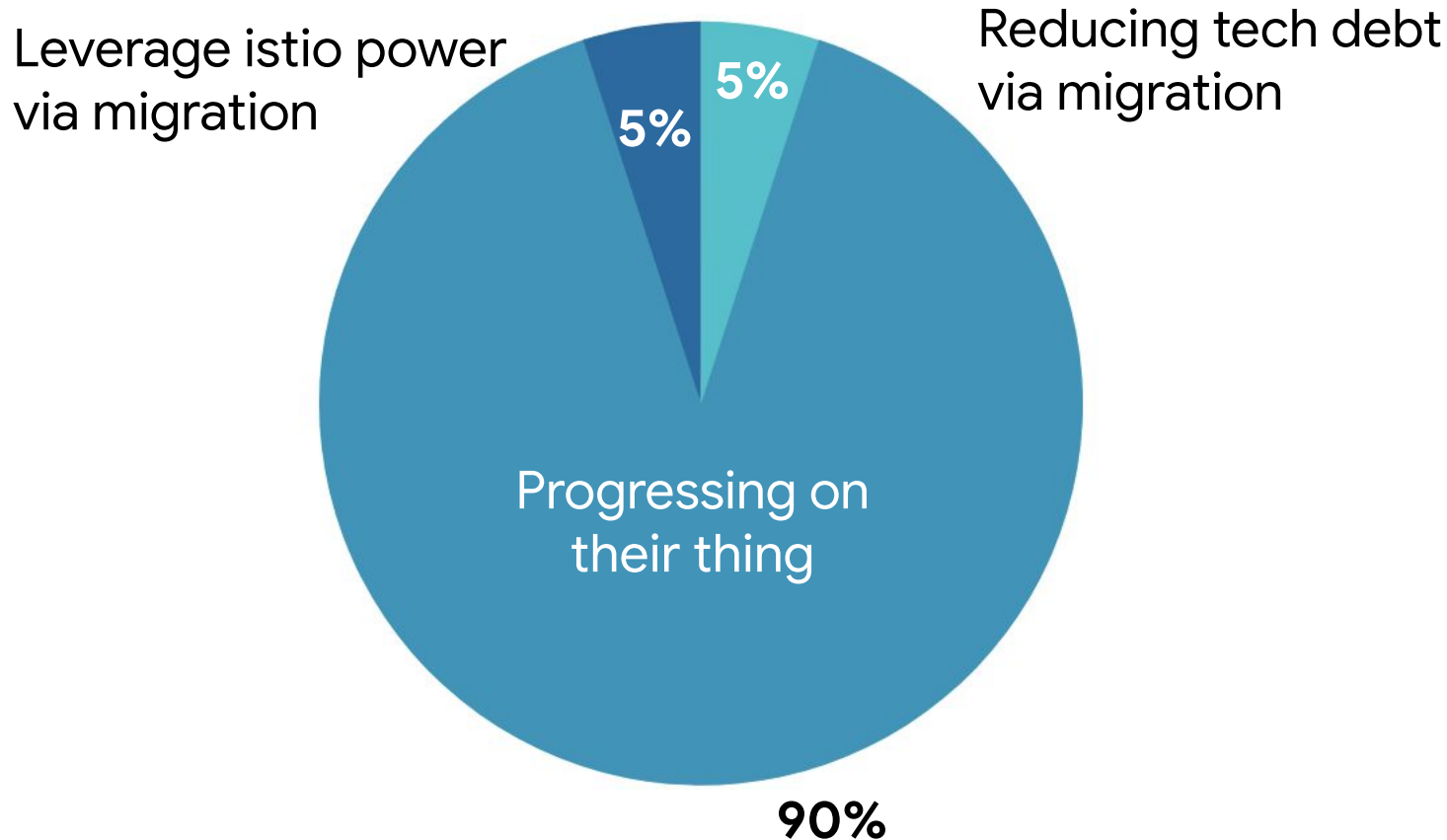


KubeCon



CloudNativeCon

North America 2021



What Developers Get

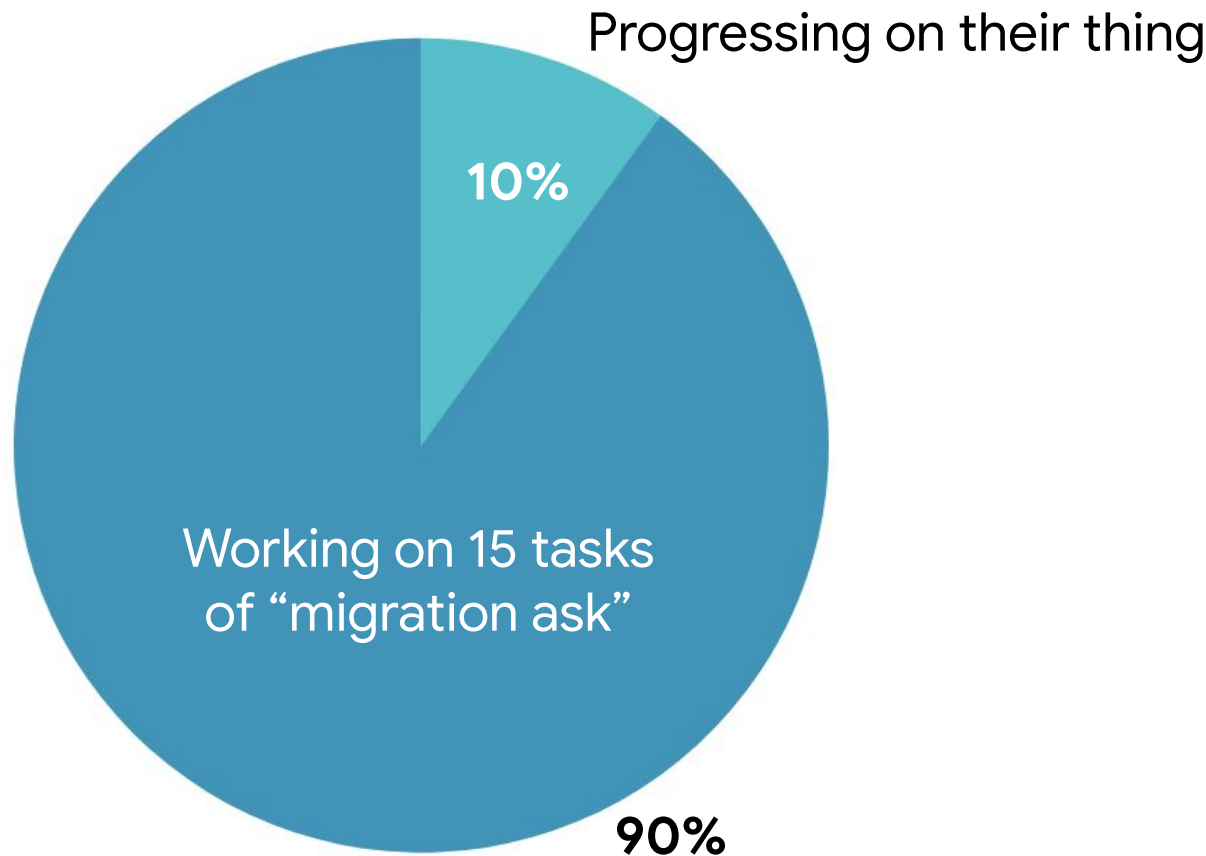


KubeCon



CloudNativeCon

North America 2021



Needs to Fix Migration Overhead



KubeCon



CloudNativeCon

North America 2021

Needs to Fix Migration Overhead

- Develop **abstractions** over migrating infrastructure

Needs to Fix Migration Overhead

- Develop **abstractions** over migrating infrastructure
- Make current migration easier

Needs to Fix Migration Overhead



KubeCon



CloudNativeCon

North America 2021

- Develop **abstractions** over migrating infrastructure
- Make current migration easier
- Avoid leaky abstractions

Needs to Fix Migration Overhead



KubeCon



CloudNativeCon

North America 2021

- Develop **abstractions** over migrating infrastructure
- Make current migration easier
- Avoid leaky abstractions
- Make **future migrations easier**

Abstraction: GoPay Example



KubeCon

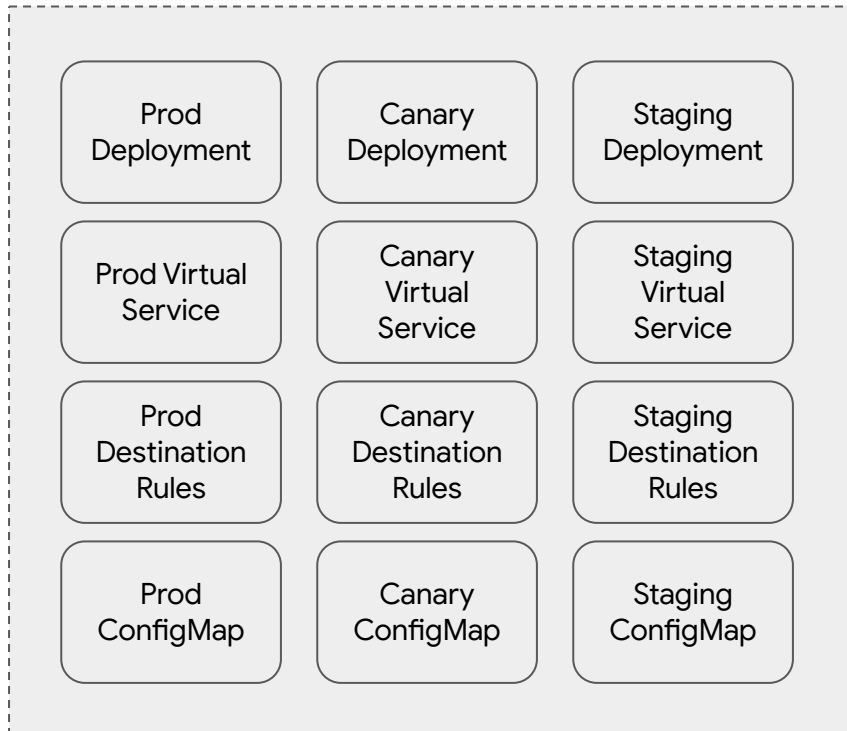


CloudNativeCon

North America 2021

Abstraction: GoPay Example

Kubernetes / Istio Config Files



kubectl apply



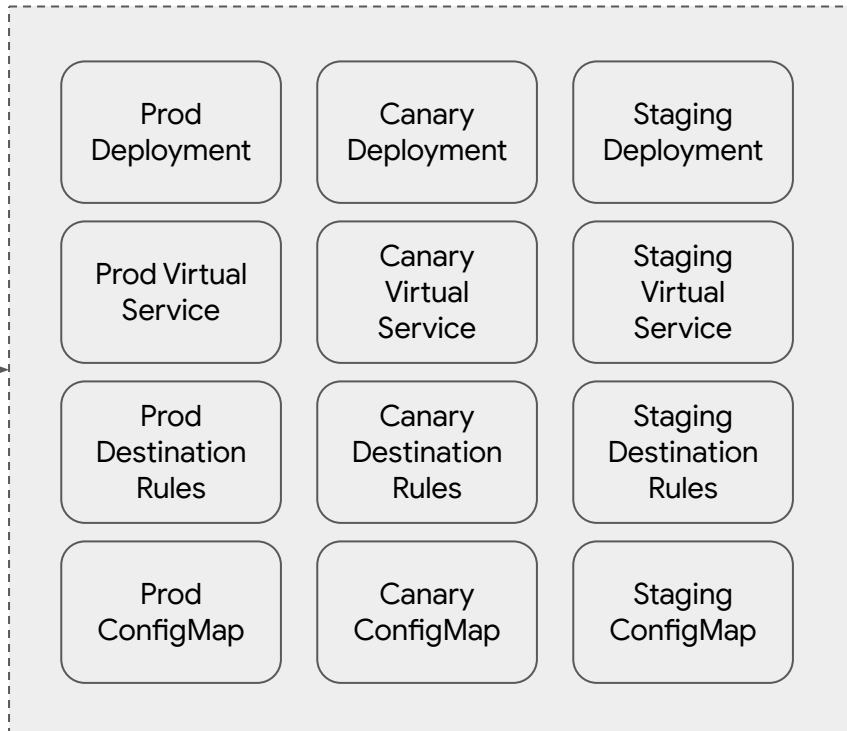
Abstraction: GoPay Example

Helm Templates



helm
render

Kubernetes / Istio Config Files



kubectl apply



Abstraction: GoPay Example

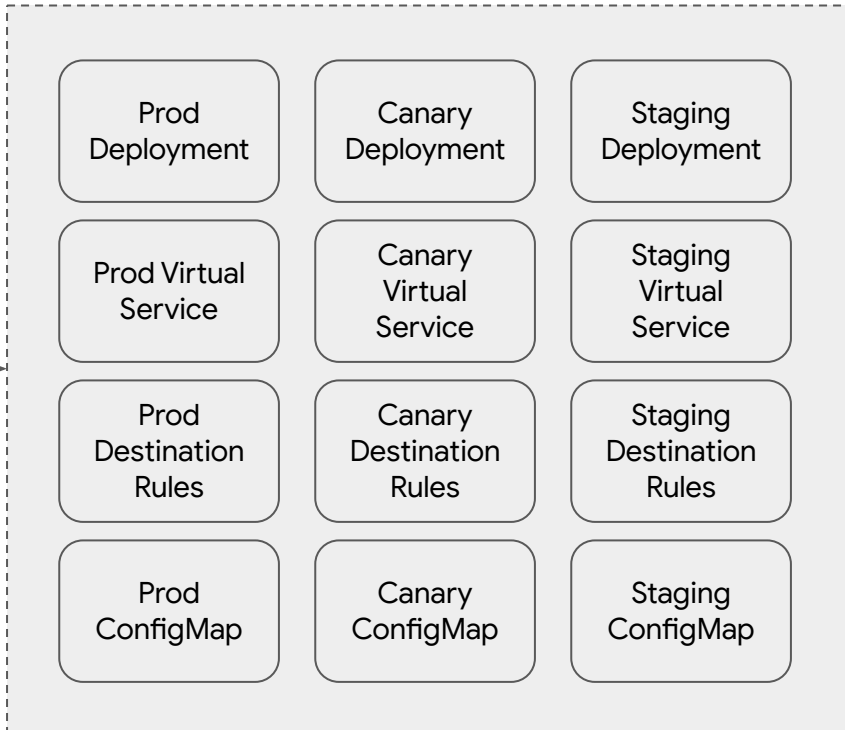
Helm Templates



Abstractions

helm
render

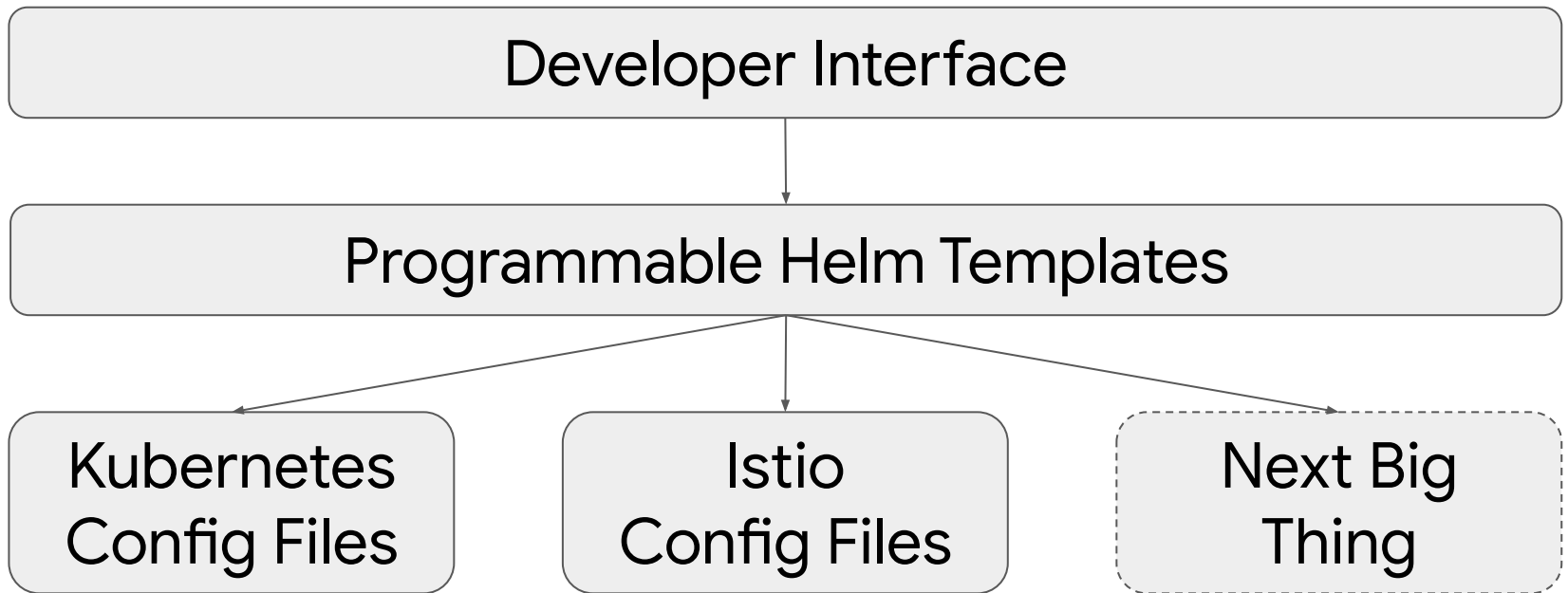
Kubernetes / Istio Config Files



kubectl apply



Better Abstraction?



Fixing Migration Overhead

- Standardize on **90% of usecases**
- Migrate under **abstraction layer**

RESILIENCE
REALIZED



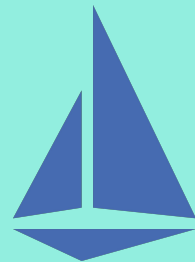
KubeCon



CloudNativeCon

North America 2021

Abstracting Istio Infrastructure through Developer Platform



GoPay.sh: Our Developer Platform



KubeCon



CloudNativeCon

North America 2021

Ownership

Service Delivery

Service mesh by default

Third Party Integration

Application Ownership

- Who owns what?
- No namespace restriction

Application Ownership



KubeCon



CloudNativeCon

North America 2021

STRATEGIES

- Onboard application as team resource

Application Catalog	
Name	Stream
██████████-service	Financial
██████████-service	Ops & De
██████████-adapter	Financial
██████████n-adapter	Emoney
Super ██████████-worker	Ops & De

Application Ownership



KubeCon



CloudNativeCon

North America 2021

STRATEGIES

- Deployment to new team namespace

Release Catalog (████████)

Release Status - All ▾

Pod - All ▾

Stream - All ▾

Type - All ▾

Env - All ▾

Search by release name

Name	Application	Type	Stream	Pod	Env	Cluster	●	Last Deployed	Status
████████-gsh	████████-service	service	FN████████	Dis████████	staging	████████	✓	5 Oct 2021 01:09pm	ONBOARDED
████████-worker-	████████-v2	worker	Dis████████	Dis████████	production	████████	✓	4 Oct 2021 03:17pm	ONBOARDED

Application Ownership



KubeCon



CloudNativeCon

North America 2021

STRATEGIES

- Ownership transfer

Transfer Ownership

From Stream

Source stream that owns the current application.

To Stream *

Destination stream which will own the current application after ownership transfer

☐ I have discussed and asked approval from the destination stream as the new owner of [redacted] **service** with all associated releases. I understand that this will transfer the ownership from [redacted] and there will be no changes with deployment namespace.

Transfer App and Releases

Helm Chart Headache

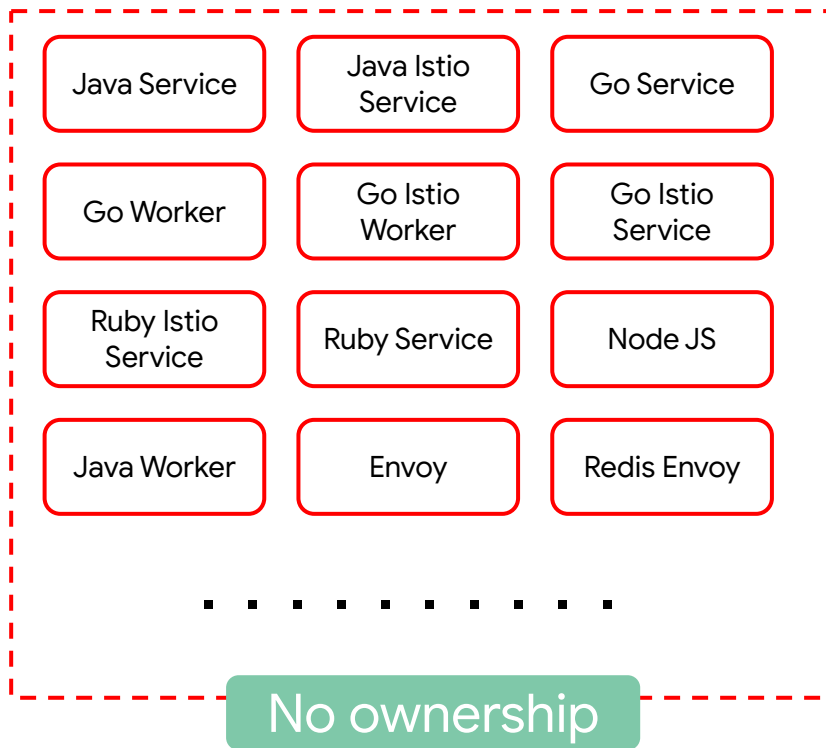


KubeCon



CloudNativeCon

North America 2021



- One language, multiple charts

Helm Chart Headache

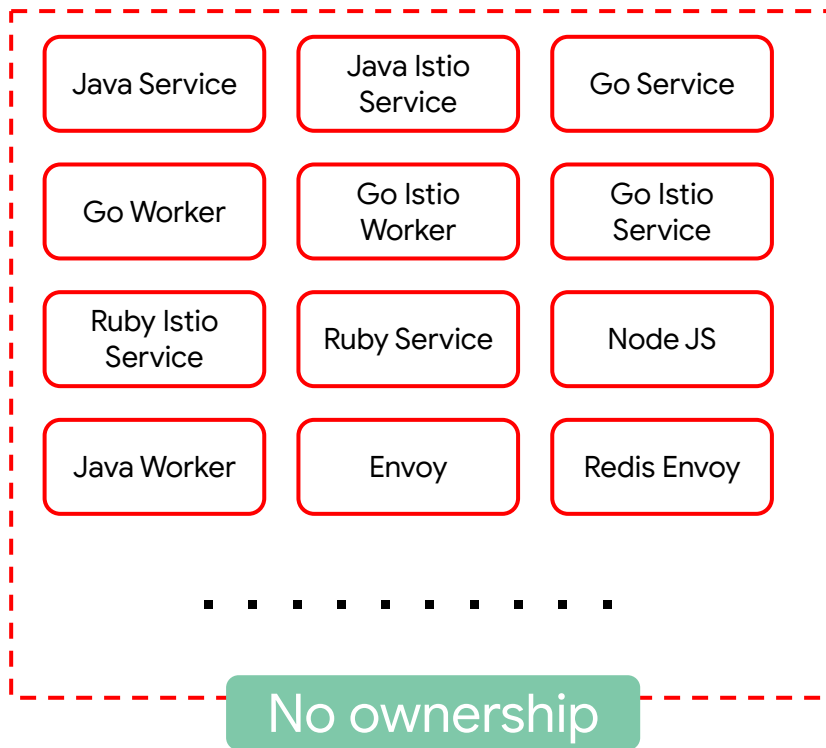


KubeCon



CloudNativeCon

North America 2021



- One language, multiple charts
- Not standardized and No ownership

Helm Chart Headache

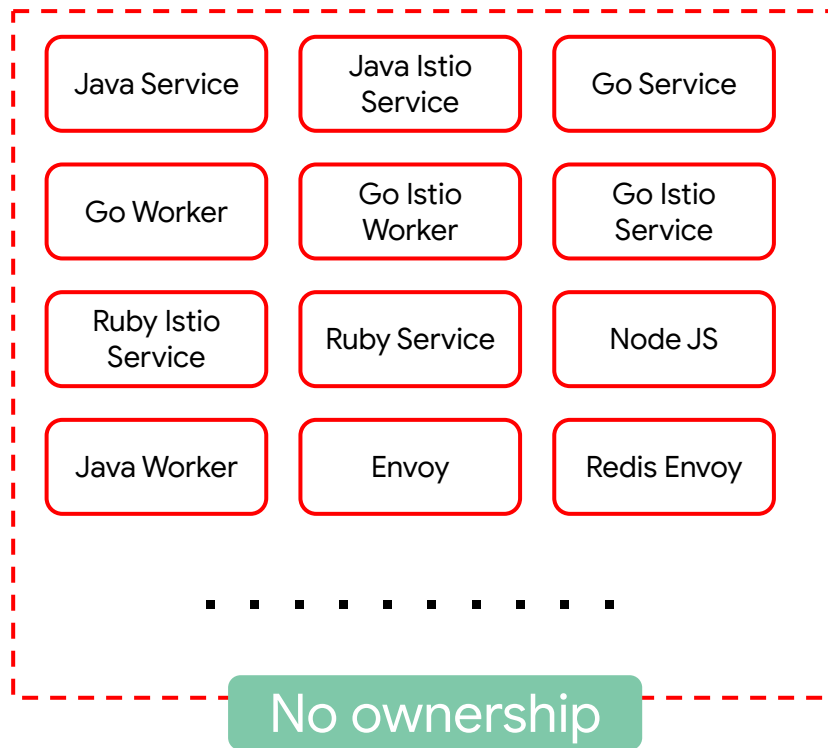


KubeCon



CloudNativeCon

North America 2021



- One language, multiple charts
- Not standardized and No ownership
- Kubernetes upgrade blocker

- Too much infra related information

script:

```
- gopay-cd render ${PROJECT_NAME}
  --team-name ${TEAM}
  --chart-name ${RUBY_CHART_APP_NAME}
  --chart-version ${HELM_CHART_RUBY_APP_VERSION}
  --dest-cluster ${CLUSTER_NAME}
  --app-config ${APP_CONFIG_PATH}
  --set image.bucketName=${IMAGE_NAME}
  --set image.tag="${APP_VERSION}-${CI_COMMIT_SHA}"
  --namespace ${NAMESPACE}
  --set service.enabled=true
  --set istio.enabled=false
- gopay-cd deploy ${PROJECT_NAME}
  --team-name ${TEAM}
  --dest-cluster ${CLUSTER_NAME}
```

only:

```
- master
```


Deployment Abstraction



KubeCon



CloudNativeCon

North America 2021

Type *

The type of the service.

GRPC Service

Name *

Name of the service. Ensure the name doesn't collide with existing Deployment objects or ArgoCD app in this cluster

Port *

Port of the service.

9000

Run Command *

The run command used to start up the worker.

/usr/bin/java

Migration Command

The migration command used to perform database migration.

/usr/bin/java

-classpath

Yggdrasil

+ Add Yggdrasil Path

You must add application configuration here. Add one or more Yggdrasil path used to configure your application release.

Edit

Remove

Specification *

Select Specification

Select CPU and memory specification used to spin up your application release as Pods in this cluster. If you are curious the difference between request and limit, read the [Kubernetes docs](#).

Type: general-purpose.small

Requests

CPU: 100m

Memory: 256Mi

Limits

CPU: 200m

Memory: 512Mi

Autoscaling

Enable auto scale feature



Replication Size *

Select Replication Size

Select how many replicas of Pod do you want your application release to be running in this cluster. If autoscaling enabled, you only need to configure the minimum and maximum replicas.

Min. size: 2

Max. size: 3

Deployment Abstraction

Centralized

Machine
Spec

Replica and
Autoscaling

Run
Command

Application
Config

Generic
Chart

Maintained by
DevEx team

render

Deployment

Service

HPA

Configmap

Virtual
Service

Canary
Destination
Rules

Deployment Context
(Staging, Prod, Canary,
Rollback)



argo

sync



Adopting Service Mesh



KubeCon



CloudNativeCon

North America 2021

Adopting Service Mesh

- Client side load balancing and service discovery

Adopting Service Mesh

- Client side load balancing and service discovery
- Get rid of in-house envoy control plane (**tech debt**)

Adopting Service Mesh

- Client side load balancing and service discovery
- Get rid of in-house envoy control plane (**tech debt**)
- Utilizing better traffic split for canary

Adopting Service Mesh



KubeCon



CloudNativeCon

North America 2021

- Client side load balancing and service discovery
- Get rid of in-house envoy control plane (**tech debt**)
- Utilizing better traffic split for canary
- Making app codebase more lightweight



KubeCon



CloudNativeCon

North America 2021

make the new approach

THE DEFAULT

examples:

- Centralized **deployment configurations**
- Created/Onboarded services are automatically deployed to **new istio enabled namespaces**
- Create application generator that scaffold minimal setup using **new approaches**

Create New Application

Application is a group of services and workers which share the same codebase.



Note

This new application wizard is used if you want to build project from scratch. This will create new Gitlab repository for you, along with minimal started code, gitlab-ci.yml, and namespace. If you already have existing repository, please choose onboard existing repository.

[Onboard Existing Repository](#)

1

Basic Information

2

Select Owner

3

Add Service, Worker or Cron

4

Ready

Start with filling out basic information about your new application.

Name *

Unique name of the application, lowercase-with-dashes.

Description *

Description of the application.

Tags *

Tag the application with flow name and anything else you think can be useful for filtering and discovery later.

+ Add tags

Next



KubeCon



CloudNativeCon

North America 2021

abstraction

ISTIO DEPLOYMENT

```
gopaysh deployment create foo-service  
--cluster id-01  
--env production  
--canary  
--image ${IMAGE}
```

canary

Canary
Deployment

Canary HPA

Virtual
Service
Canary 5%Canary
Configmapcanary
rollbackVirtual
Service
Prod 100%

rollout

Production
DeploymentProduction
HPAVirtual
Service
Prod 100%Production
Configmap

Third Party Integration

All teams are trying to provide their solutions separately

- monitoring and logging
- Infra cost management
- stateful component provisioning
- private or public domains

Addons Integration



KubeCon



CloudNativeCon

North America 2021

STRATEGIES

- Enrich service capabilities

Add a New Addon

1 Select Providers

2 Information Detail

Provider *

Addon provider

GoPay.sh Addons

Kind *

Addon kind

Select Addon Kind

In-Cluster Domain

Domain accessible by other apps within same Kubernetes cluster, can be from inside or outside Istio service mesh

Out-Cluster Domain

Domain accessible from outside of the cluster, but same VPC network and via user VPN from your laptop

Public Domain

Domain accessible from public internet

Partition Manager

Partition manager to perform table partition on your database using db_commons library

Redis Envoy

Addons Integration



KubeCon



CloudNativeCon

North America 2021

STRATEGIES

- Enrich service capabilities
- Open Service Broker Spec

* Service Routing

URL routes used for redirecting requests coming to your service via FQDN defined above

HTTP Header



Use HTTP header(s) if you want to route requests whose specific HTTP header to your service

+ Add Item

* Path

/payment

Path to match. e.g. `/users` or just `/`. Can also be regex if you choose regex type. This value is case sensitive

* Matcher

prefix

How to match a given path string. Choose prefix for prefix-based match, choose exact for exact string match, choose regex for ECMAScript style regex-based match

+ Add Item

Addons Integration



KubeCon



CloudNativeCon

North America 2021

STRATEGIES

- Enrich service capabilities
- Open Service Broker Spec
- Addon provider can modify service behaviour if necessary

UI Plugin Integration



KubeCon



CloudNativeCon

North America 2021

STRATEGIES

- Integrating custom UI components to the portal
- Make it effective by providing tutorials and documentation



KubeCon



CloudNativeCon

North America 2021

RESILIENCE

REALIZED

Istio (Re)Rollout





KubeCon



CloudNativeCon

North America 2021

How do we migrate programmatically?

Onboard Existing Release



1

Basic Information

2

Release Detail

3

Others

Environment

Environment type

staging

Cluster

The Kubernetes cluster where the service, worker or cronjob is going to run

Existing Kubernetes Namespace

The Kubernetes namespace where the service, worker or cronjob is currently running

default

Release Type

Select a release type

GRPC Service

Release Name

Enter existing release name.

foo-service

- Devs specify the service name and its existing location
- Backend calls kubernetes API to get service's infra configurations
- Devs review is required to ensure service metadata is onboarded properly



KubeCon



CloudNativeCon

North America 2021

What do we migrate programmatically?

- Configuration updates (ex: charts, k8s namespace)
- Sidecar injection
- Changing CI/CD system, etc



KubeCon



CloudNativeCon

North America 2021

Triggering Deployment

PRODUCTION

Please make sure this config exists in your `.gitlab-ci.yml`

```
foo-service:id-01:production:deploy:
  image: [REDACTED]/gopaysh:latest
  stage: production
  script:
    - gopaysh deployment create foo-service
      --cluster id-01
      --env production
      --image ${IMAGE}
  only:
    - master
  when: manual
  tags:
    - [REDACTED]
```

- Auto generated gitlab ci job for deployment
- After service is onboarded, there will be two deployment running

Comms



KubeCon



CloudNativeCon

North America 2021

Comms: Within Istio Mesh

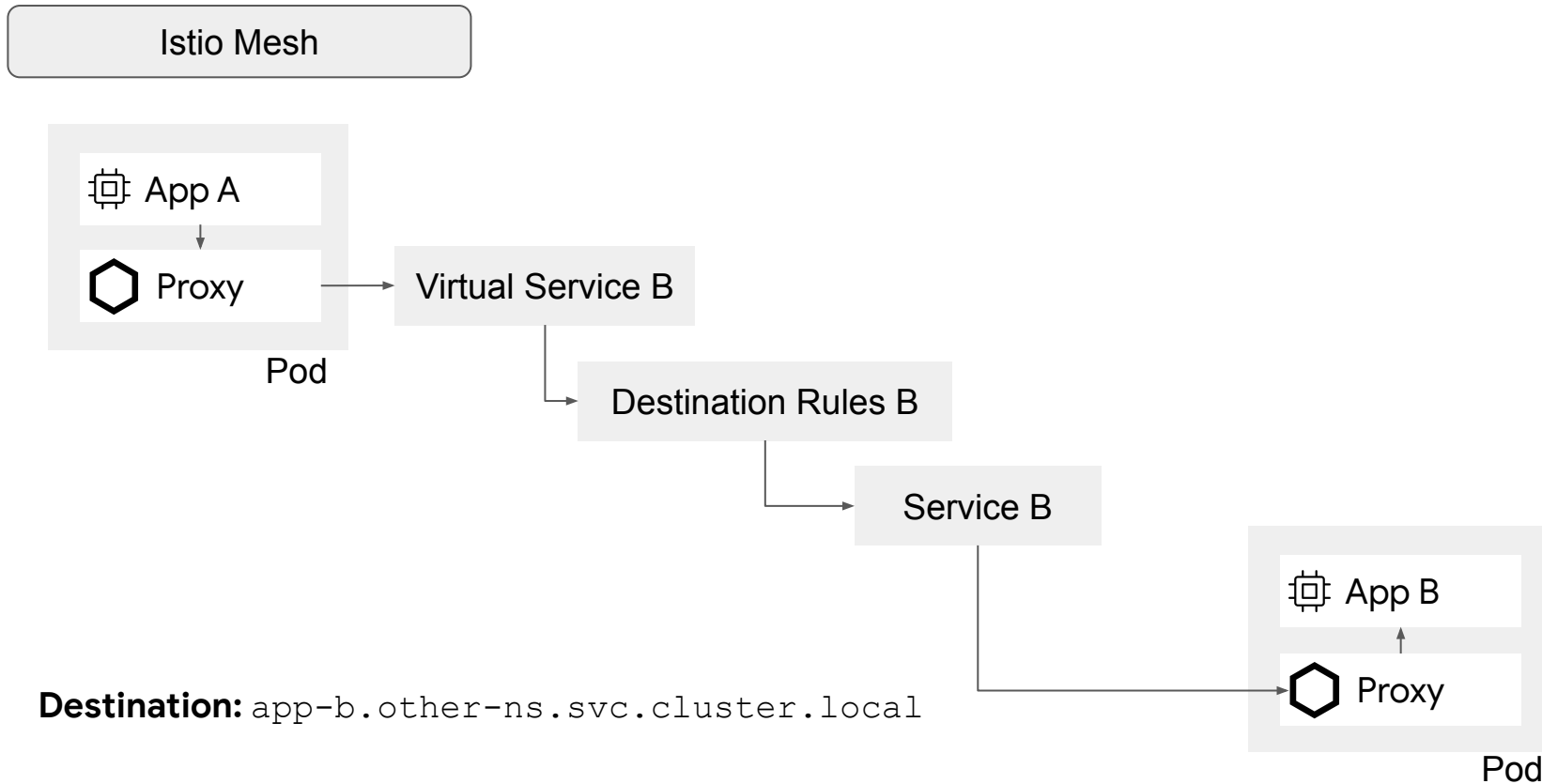


KubeCon

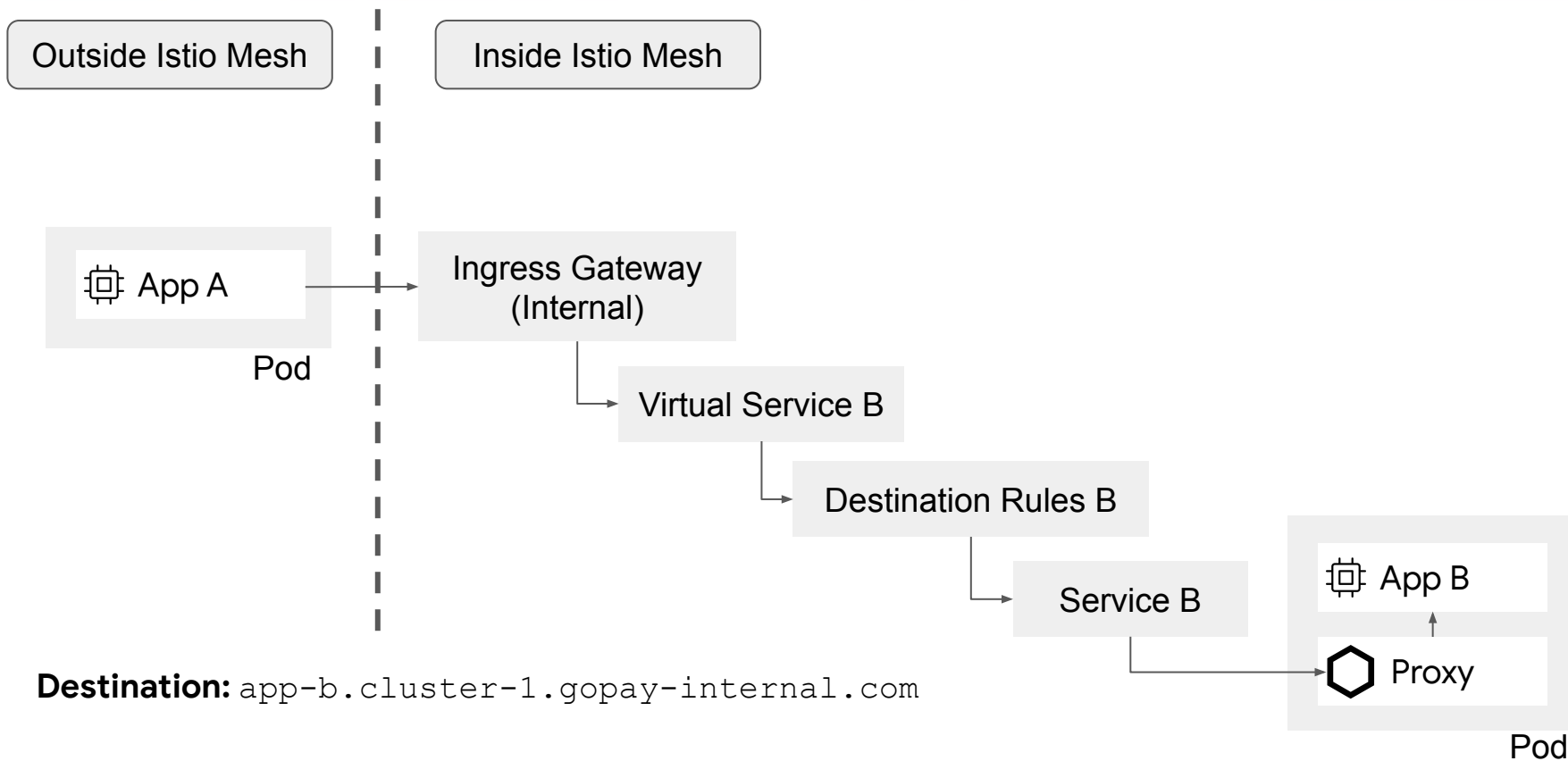


CloudNativeCon

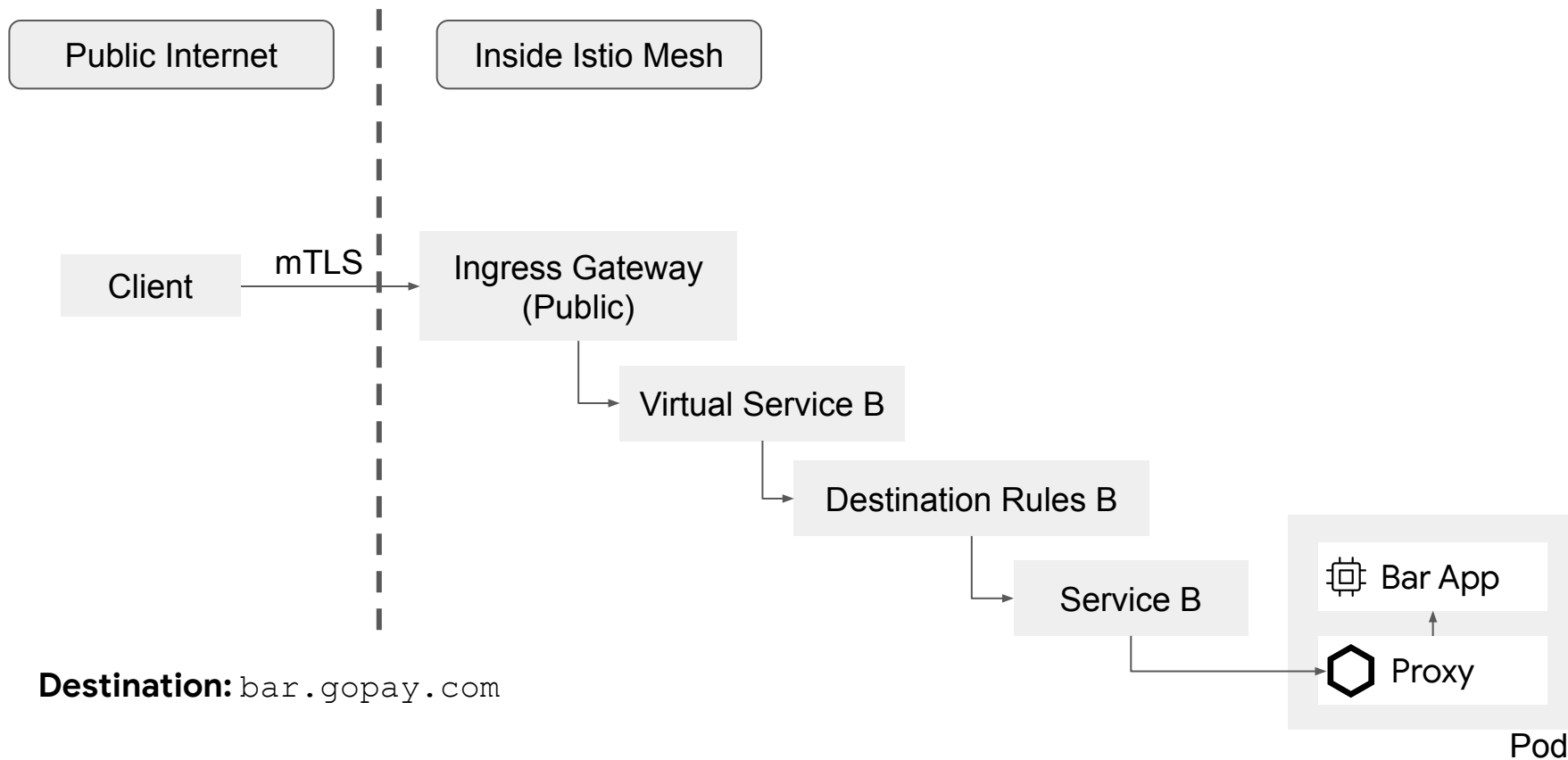
North America 2021



Comms: From Outside Mesh



Comms: From Public



Migration strategy: 3 stages



KubeCon



CloudNativeCon

North America 2021

Migration strategy: alpha rollout

**Pick several
least critical services**

**Assist devs to onboard
their services and to
update their clients**

Migration strategy: alpha rollout



KubeCon



CloudNativeCon

North America 2021

- **Understanding devs behaviour** during migration
- **Gathering feedbacks**, bugs, and identifying missing features
- Feedbacks are used to do UI **improvement and automation**

Migration strategy: beta rollout

**Make each teams
choose at least one
service to onboard**



Observe how devs
perform onboarding and
migration

Migration strategy: beta rollout

- Ensuring most of the team onboarding use cases are covered
- Discovered networking issues

Migration strategy: wide rollout

Create several phases of migration program

Team owns migration
end-to-end and partners
with leadership and devs
to finish

Migration strategy: wide rollout

- Speed up migration
- Monitoring migration completeness

Results

After only 4 months ...

After only 4 months ...

28% of services
in Istio



After only 4 months ...

> 50% of teams
in Istio



What's Next



KubeCon



CloudNativeCon

North America 2021

What's Next

- Introduce more Istio power to developers

What's Next



KubeCon



CloudNativeCon

North America 2021

- Introduce more Istio power to developers
 - **Traffic mirroring** for very low risk to test in prod
 - **Service graph** to help understand better
 - Decouple **rate limiting & circuit breaker** from code
 - **Multi cluster** capability for failover

Takeaways



KubeCon



CloudNativeCon

North America 2021

- Develop **abstractions** over infrastructure
- Make new approach the **default**
- **Iterate on migration** to ensure its fully validated, enabled, and finished

References

- [Melanie Cebulla, Infrastructure Migration at Scale](#)
- [Spotify, Dev Portal Adoption Metrics](#)

Thanks to GoPay.sh Team



KubeCon



CloudNativeCon

North America 2021



RESILIENCE
REALIZED



KubeCon



CloudNativeCon

North America 2021

Thank you LA!

@imrenagi & @girikuncoro from  **gopay**