



KubeCon



CloudNativeCon

North America 2023

Adopting Server-Side Apply in Knative

Dave Protasowski

Dave Protasowski



KubeCon



CloudNativeCon

North America 2023



vmware®

Staff Engineer



Serving Lead & TOC



dprotaso



lintinmybelly



dprotasowski

Agenda



KubeCon



CloudNativeCon

North America 2023

- Problems with Client-Side Apply
- Server-Side Apply Overview
- Knative Overview
- Where Server-Side Apply impacts Knative
- Status & Learnings

Problems with Client-Side Apply



KubeCon



CloudNativeCon

North America 2023

- Controllers and users make UPDATES to resources locally

Problems with Client-Side Apply



KubeCon



CloudNativeCon

North America 2023

- Controllers and users make UPDATES to resources locally

K8s API

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: demo
data:
  Speaker: Dave P
```

Problems with Client-Side Apply



KubeCon



CloudNativeCon

North America 2023

- Controllers and users make UPDATES to resources locally



```
apiVersion: v1
kind: ConfigMap
metadata:
  name: demo
data:
  Speaker: Dave P
  City: Chicago
```



```
apiVersion: v1
kind: ConfigMap
metadata:
  name: demo
data:
  Speaker: Dave P
  Conference: KubeCon NA
```

K8s API

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: demo
data:
  Speaker: Dave P
```

Problems with Client-Side Apply



KubeCon



CloudNativeCon

North America 2023

- Controllers and users make UPDATES to resources locally



```
apiVersion: v1
kind: ConfigMap
metadata:
  name: demo
data:
  Speaker: Dave P
  City: Chicago
```

kubectl apply -f

K8s API



```
apiVersion: v1
kind: ConfigMap
metadata:
  name: demo
data:
  Speaker: Dave P
  Conference: KubeCon NA
```

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: demo
data:
  Speaker: Dave P
  City: Chicago
```

Problems with Client-Side Apply



KubeCon



CloudNativeCon

North America 2023

- Controllers and users make UPDATES to resources locally



```
apiVersion: v1
kind: ConfigMap
metadata:
  name: demo
data:
  Speaker: Dave P
  City: Chicago
```

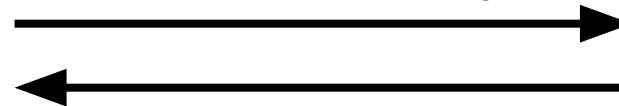


```
apiVersion: v1
kind: ConfigMap
metadata:
  name: demo
data:
  Speaker: Dave P
  Conference: KubeCon NA
```

K8s API

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: demo
data:
  Speaker: Dave P
  City: Chicago
```

kubectl apply -f



Problems with Client-Side Apply



KubeCon



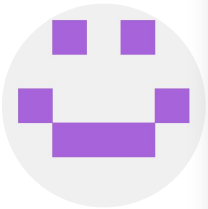
CloudNativeCon

North America 2023

- Controllers and users make UPDATES to resources locally



```
apiVersion: v1
kind: ConfigMap
metadata:
  name: demo
data:
  Speaker: Dave P
  City: Chicago
```



```
apiVersion: v1
kind: ConfigMap
metadata:
  name: demo
data:
  Speaker: Dave P
  Conference: KubeCon NA
```



K8s API

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: demo
data:
  Speaker: Dave P
  City: Chicago
```

Problems with Client-Side Apply



KubeCon



CloudNativeCon

North America 2023

- Controllers and users make UPDATES to resources locally



```
apiVersion: v1
kind: ConfigMap
metadata:
  name: demo
data:
  Speaker: Dave P
  City: Chicago
```



```
apiVersion: v1
kind: ConfigMap
metadata:
  name: demo
data:
  Speaker: Dave P
  Conference: KubeCon NA
```



kubectl apply -f

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: demo
data:
  Speaker: Dave P
  Conference: KubeCon NA
```

K8s API

Problems with Client-Side Apply






KubeCon



CloudNativeCon

North America 2023

- Controllers and users make UPDATES to resources locally
 -  Conflicts (HTTP 409)
 -  Retries Required
 -  Potential Data Loss

Problems with Client-Side Apply






KubeCon



CloudNativeCon

North America 2023

- Controllers and users make UPDATES to resources locally
 -  Conflicts (HTTP 409)
 -  Retries Required
 -  Potential Data Loss
 - ∞ Infinite Churn

Server-Side Apply



KubeCon



CloudNativeCon

North America 2023

Multiple appliers collaborating
to manage the fields of a
single object

Server-Side Apply



KubeCon



CloudNativeCon

North America 2023

Multiple appliers collaborating
to manage the **fields of a
single object**

structured schema & endpoints

```
apiVersion: v1
kind: Pod
metadata:
  name: nginx
  ...
spec:
  containers:
  - name: nginx
    image: nginx:1.25.2
    ports:
    - containerPort: 80
    ...
status:
  conditions: [...]
  containerStatuses: [...]
  ...
```

/api/v1/pods/nginx

/api/v1/pods/nginx/status

Server-Side Apply



KubeCon



CloudNativeCon

North America 2023

Multiple appliers collaborating
to manage the fields of a
single object



Server-Side Apply



KubeCon



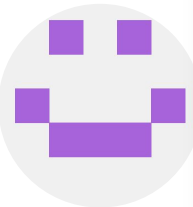
CloudNativeCon

North America 2023

Multiple appliers collaborating
to **manage** the fields of a
single object



```
apiVersion: v1
kind: ConfigMap
metadata:
  name: demo
data:
  City: Chicago
```



```
apiVersion: v1
kind: ConfigMap
metadata:
  name: demo
data:
  Conference: KubeCon NA
```


Server-Side Apply



KubeCon



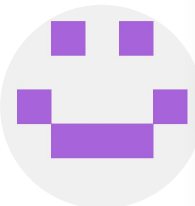
CloudNativeCon

North America 2023

Multiple appliers collaborating
to **manage** the fields of a
single object



```
apiVersion: v1
kind: ConfigMap
metadata:
  name: demo
data:
  City: Chicago
```



```
apiVersion: v1
kind: ConfigMap
metadata:
  name: demo
data:
  Conference: KubeCon NA
```

```
kubectl apply -f config.yaml
--server-side
--field-manager=$(whoami)
```

Server-Side Apply



KubeCon



CloudNativeCon

North America 2023

Multiple appliers collaborating
to **manage** the fields of a
single object



```
apiVersion: v1
kind: ConfigMap
metadata:
  name: demo
data:
  City: Chicago
```



```
apiVersion: v1
kind: ConfigMap
metadata:
  name: demo
data:
  Conference: KubeCon NA
```

```
kubectl apply -f config.yaml
--server-side
--field-manager=${whoami}
```



```
apiVersion: v1
kind: ConfigMap
metadata:
  name: demo
data:
  Speaker: Dave P
  City: Chicago
  Conference: KubeCon NA
```

Server-Side Apply



KubeCon

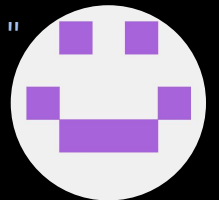


CloudNativeCon

Multiple appliers collaborating
to **manage** the fields of a
single object

```
kubectl get configmap test \  
-o yaml \  
--show-managed-fields
```

```
apiVersion: v1  
kind: ConfigMap  
metadata:  
  name: demo  
  managedFields:  
    - apiVersion: v1  
      manager: dave  
      operation: Apply  
      time: "2023-10-22T00:00:05Z"  
      fieldsType: FieldsV1  
      fieldsV1:  
        f:data:  
          f:City: {}  
  
    - apiVersion: v1  
      manager: dprotaso  
      operation: Apply  
      time: "2023-10-22T00:00:05Z"  
      fieldsType: FieldsV1  
      fieldsV1:  
        f:data:  
          f:Conference: {}  
    - ...  
data:  
  Speaker: Dave P  
  City: Chicago  
  Conference: KubeCon NA
```



Server-Side Apply



KubeCon



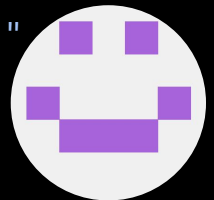
CloudNativeCon

- SIG API Machinery Deep Dive - Applying What We've Learned
 - <https://youtu.be/oiC2w1PVjrQ>
- Kubernetes Docs: Server-Side Apply
 - <https://kubernetes.io/docs/reference/using-api/server-side-apply>

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: demo
  managedFields:
    - apiVersion: v1
      manager: dave
      operation: Apply
      time: "2023-10-22T00:00:05Z"
      fieldType: FieldsV1
      fieldsV1:
        f:data:
          f:City: {}

    - apiVersion: v1
      manager: dprotaso
      operation: Apply
      time: "2023-10-22T00:00:05Z"
      fieldType: FieldsV1
      fieldsV1:
        f:data:
          f:Conference: {}

    - ...
data:
  Speaker: Dave P
  City: Chicago
  Conference: KubeCon NA
```





- Kubernetes based open source building blocks for serverless
- Founded - 2018
- CNCF Incubating - 2022
- knative.dev

- **Serving**
 - Autoscales your workloads based on traffic - can scale to zero
- **Eventing**
 - Declaratively bind consumers and producers of events
- **Client (kn)**
 - Create resources interactively from the command line
- **Functions**
 - A programming model to simplify development

- **Serving**
 - Autoscales your workloads based on traffic - can scale to zero
- **Eventing**
 - Declaratively bind consumers and producers of events
- **Client (kn)**
 - Create resources interactively from the command line
- **Functions**
 - A programming model to simplify development

Knative Serving



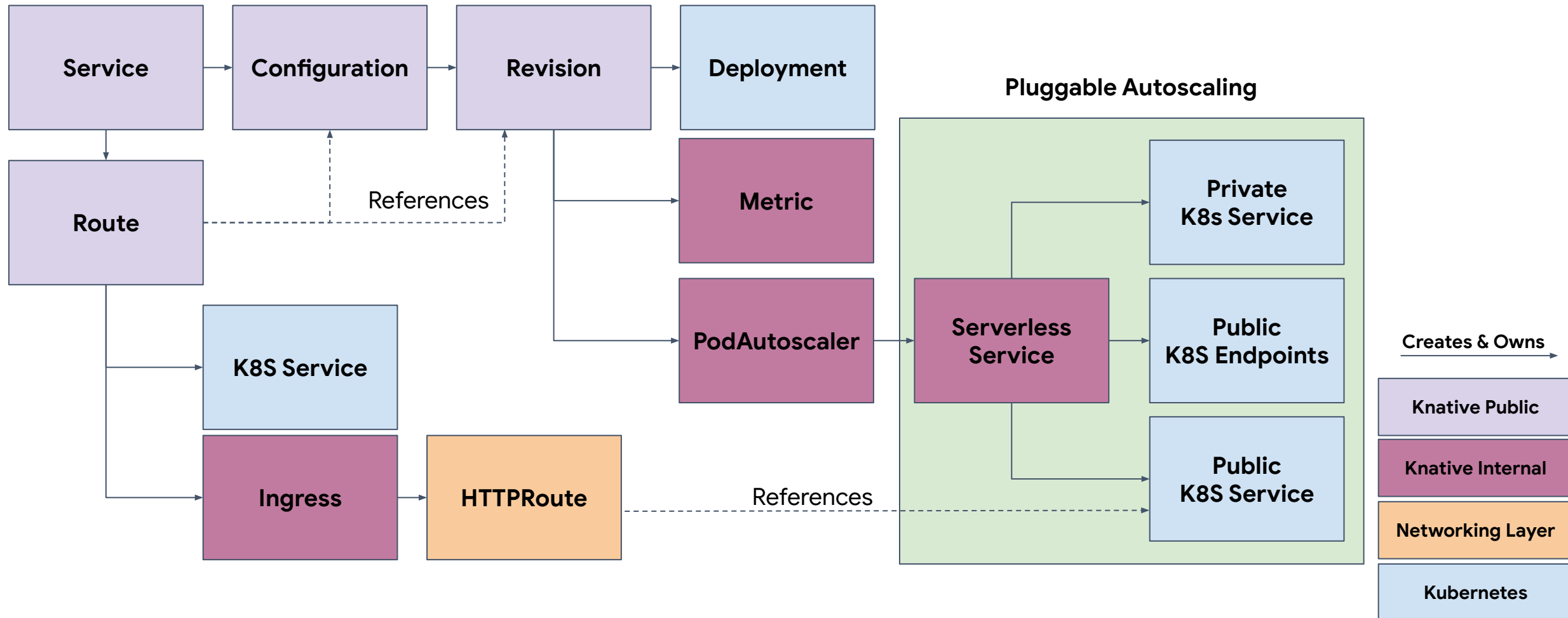
KubeCon



CloudNativeCon

North America 2023

Architecture - Resource Model



Knative Serving



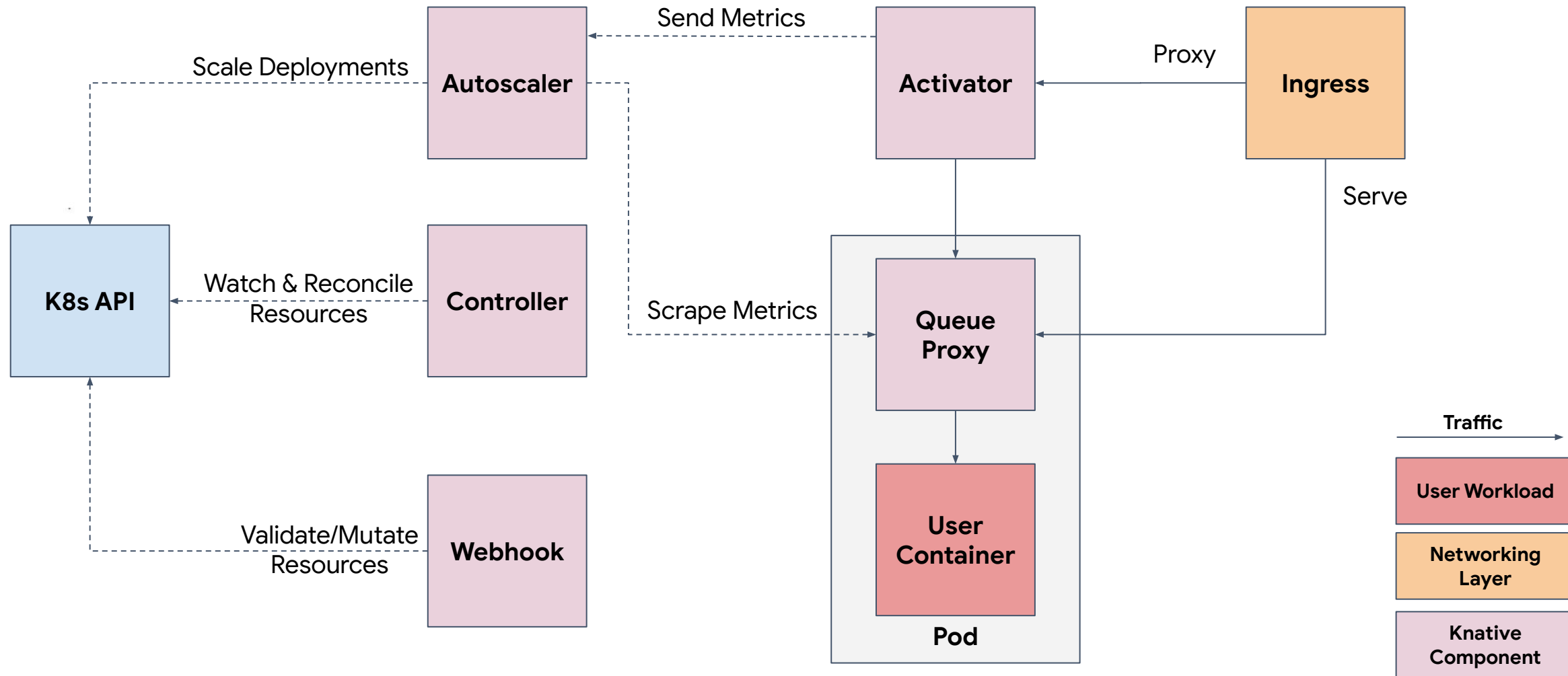
KubeCon



CloudNativeCon

North America 2023

Architecture - Components



Knative Controllers

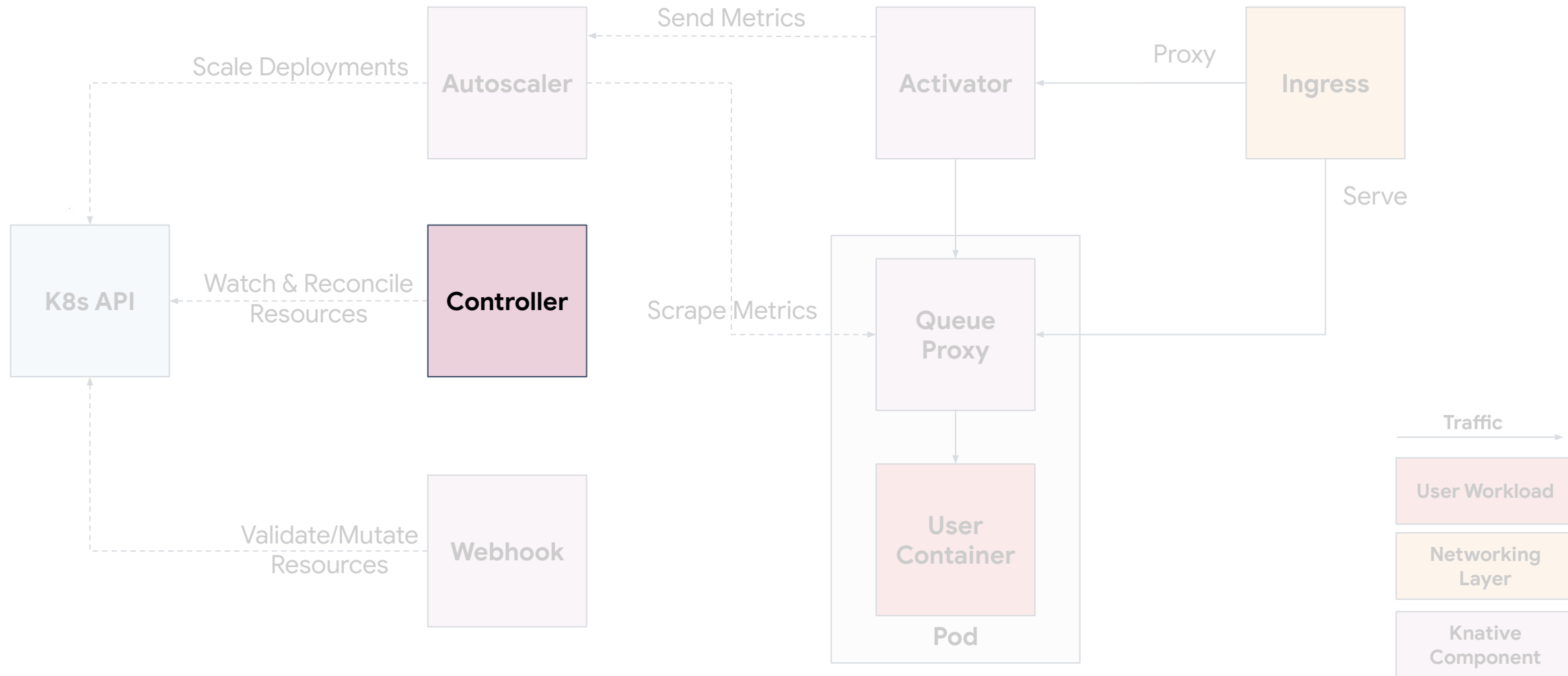


KubeCon



CloudNativeCon

North America 2023



Knative Controllers



KubeCon

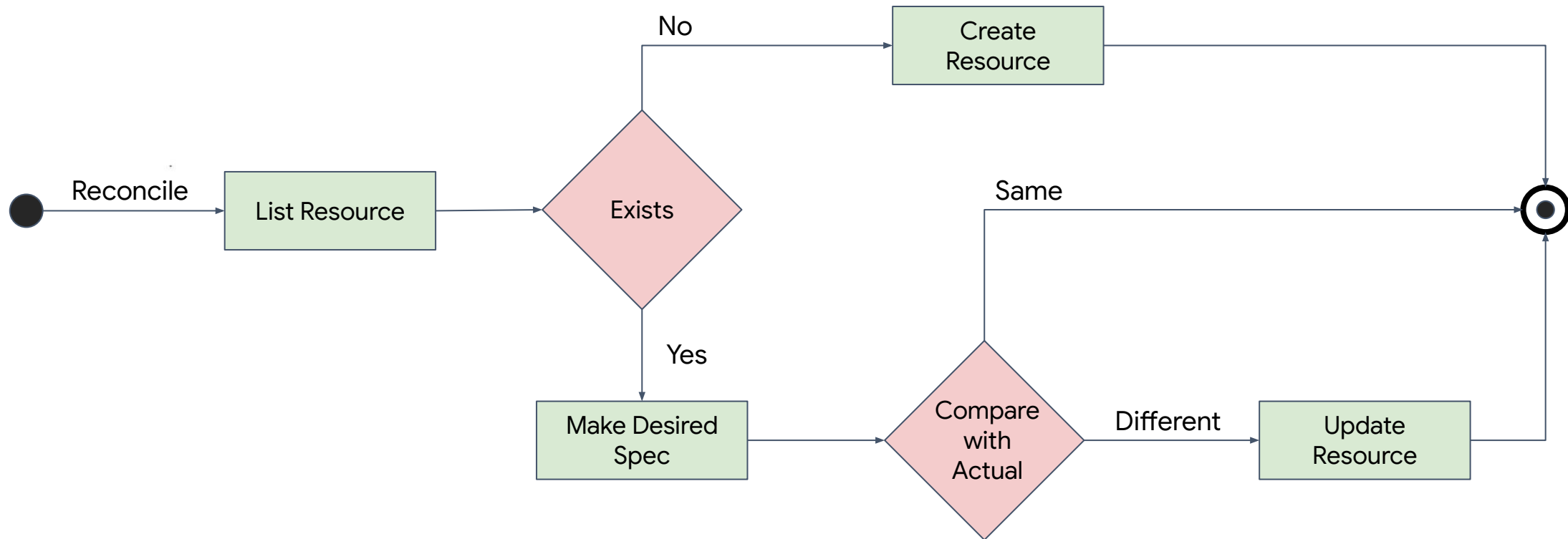


CloudNativeCon

North America 2023

- Level Based ([ref](#))
 - Look at current state and make changes
- Idempotent
 - same input \Rightarrow same output
- Reconstructive
 - for each reconcile, output is constructed from scratch

Control Flow



Knative Controllers



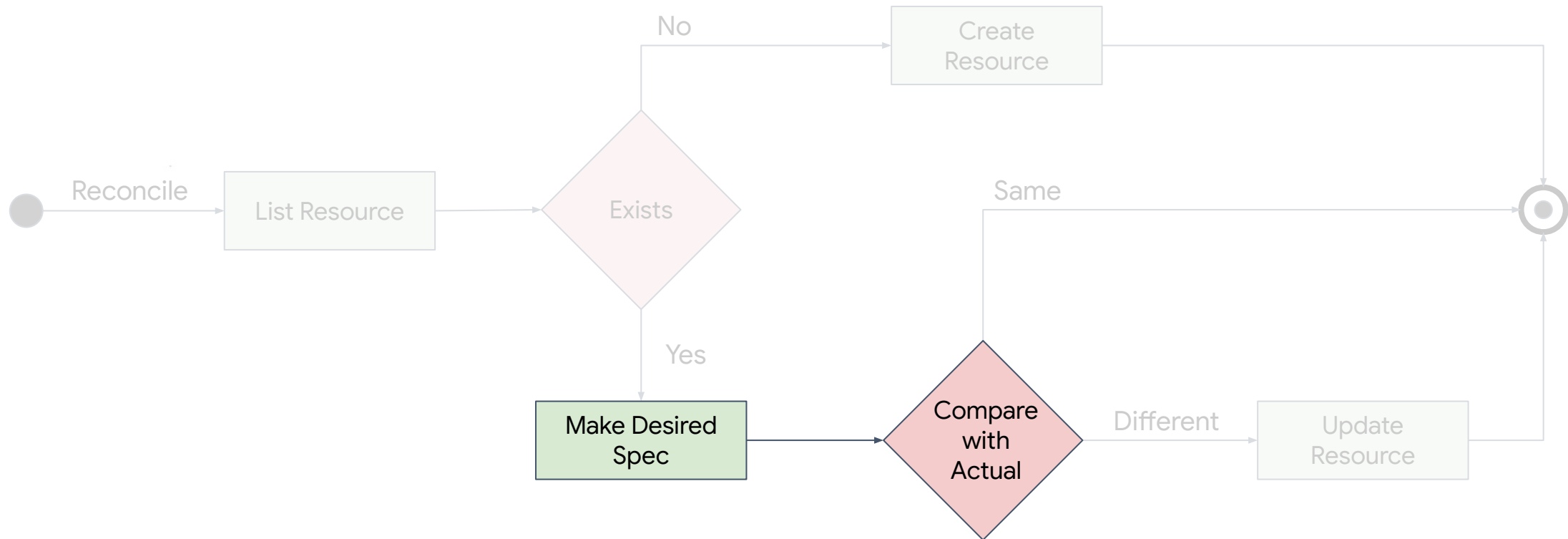
KubeCon



CloudNativeCon

North America 2023

Not using Server-Side Apply



Knative Controllers



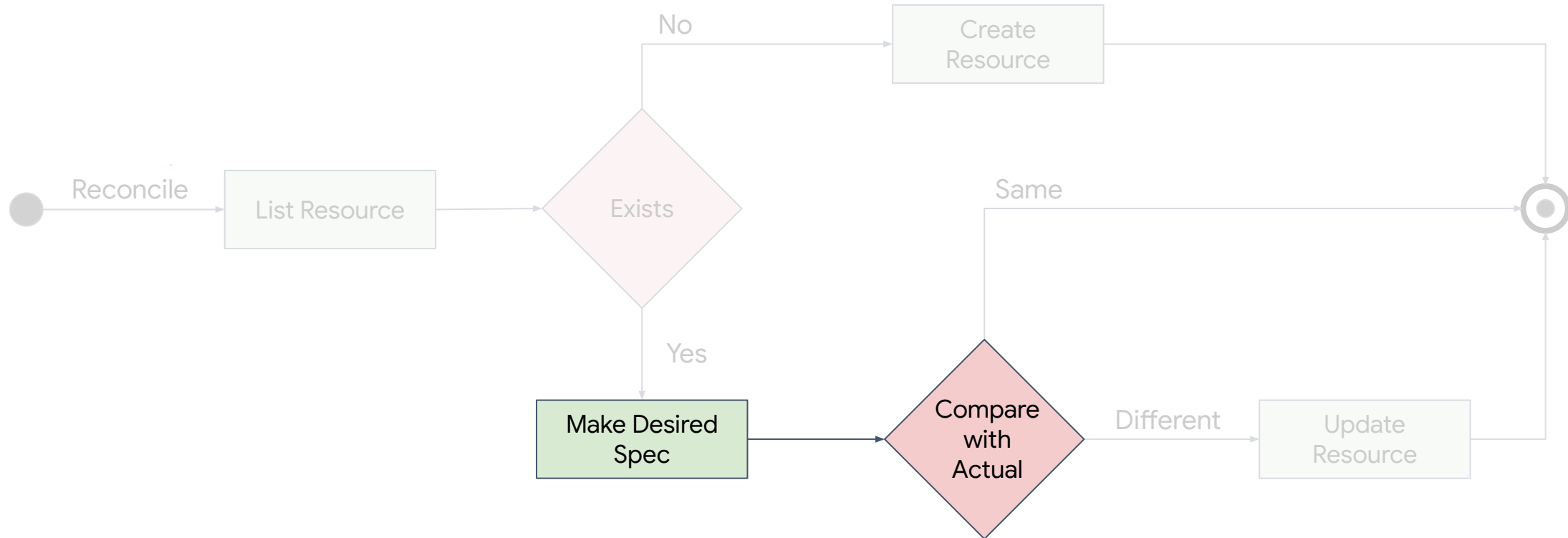
KubeCon



CloudNativeCon

North America 2023

Not using Server-Side Apply
but we have **multiple appliers!**



Knative Controllers



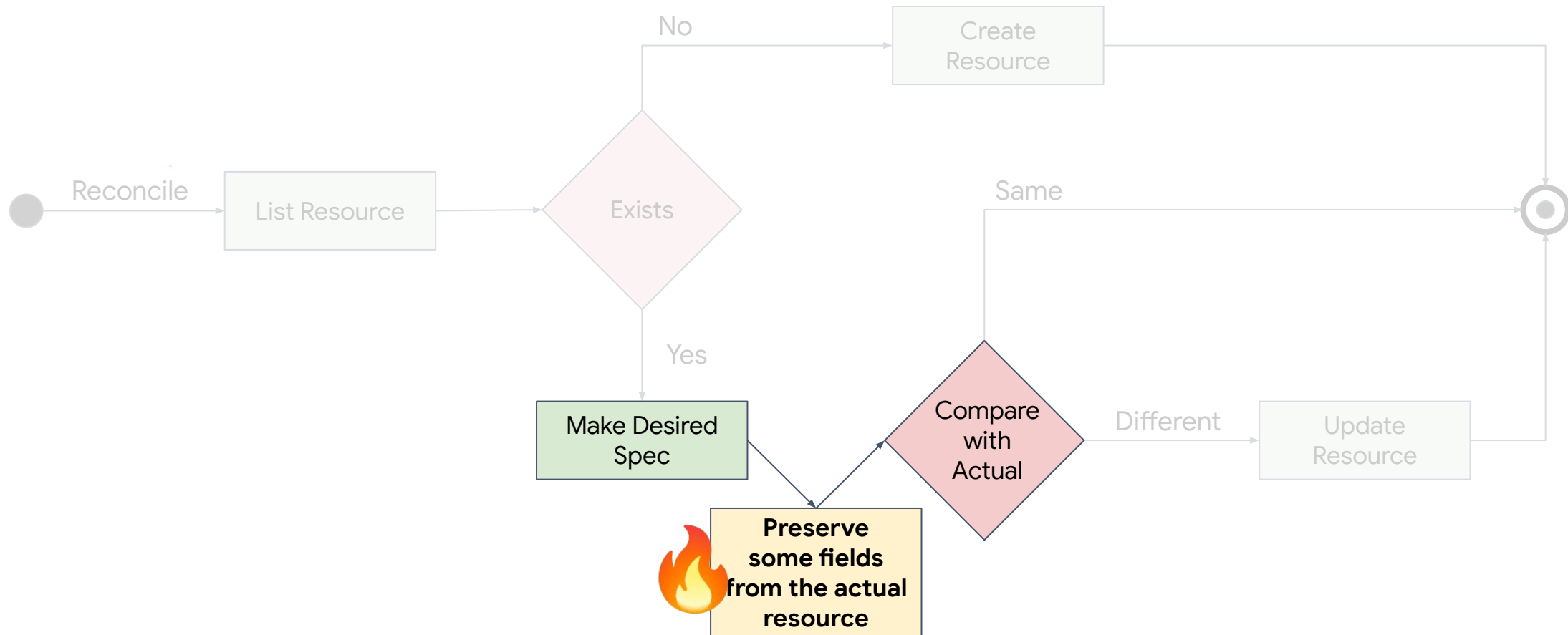
KubeCon



CloudNativeCon

North America 2023

Not using Server-Side Apply
but we have **multiple appliers!**



Knative Contention Points



KubeCon



CloudNativeCon

North America 2023



Knative Contention Points



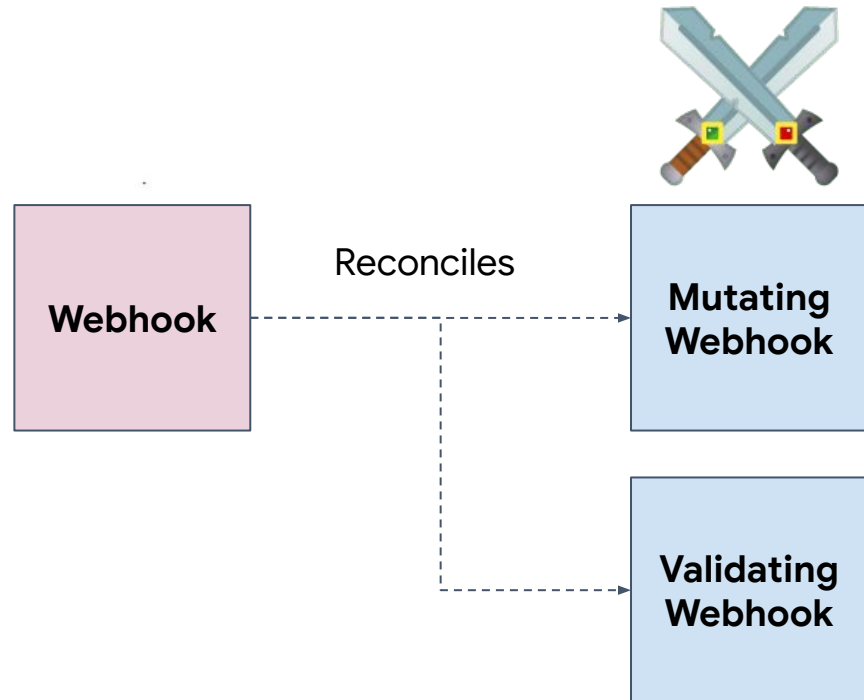
KubeCon



CloudNativeCon

North America 2023

- Our webhooks dynamically reconcile rules and clientConfig



Knative Contention Points



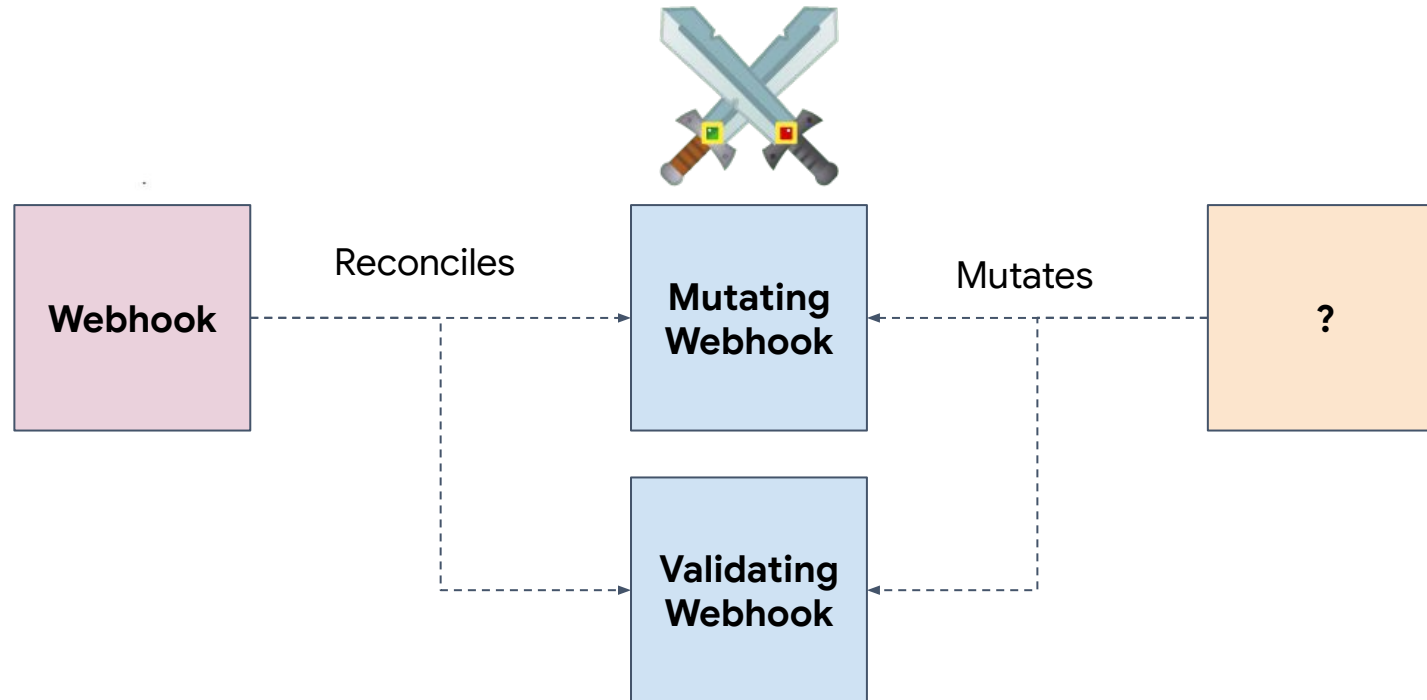
KubeCon



CloudNativeCon

North America 2023

- Our webhooks dynamically reconcile rules and clientConfig



Knative Contention Points



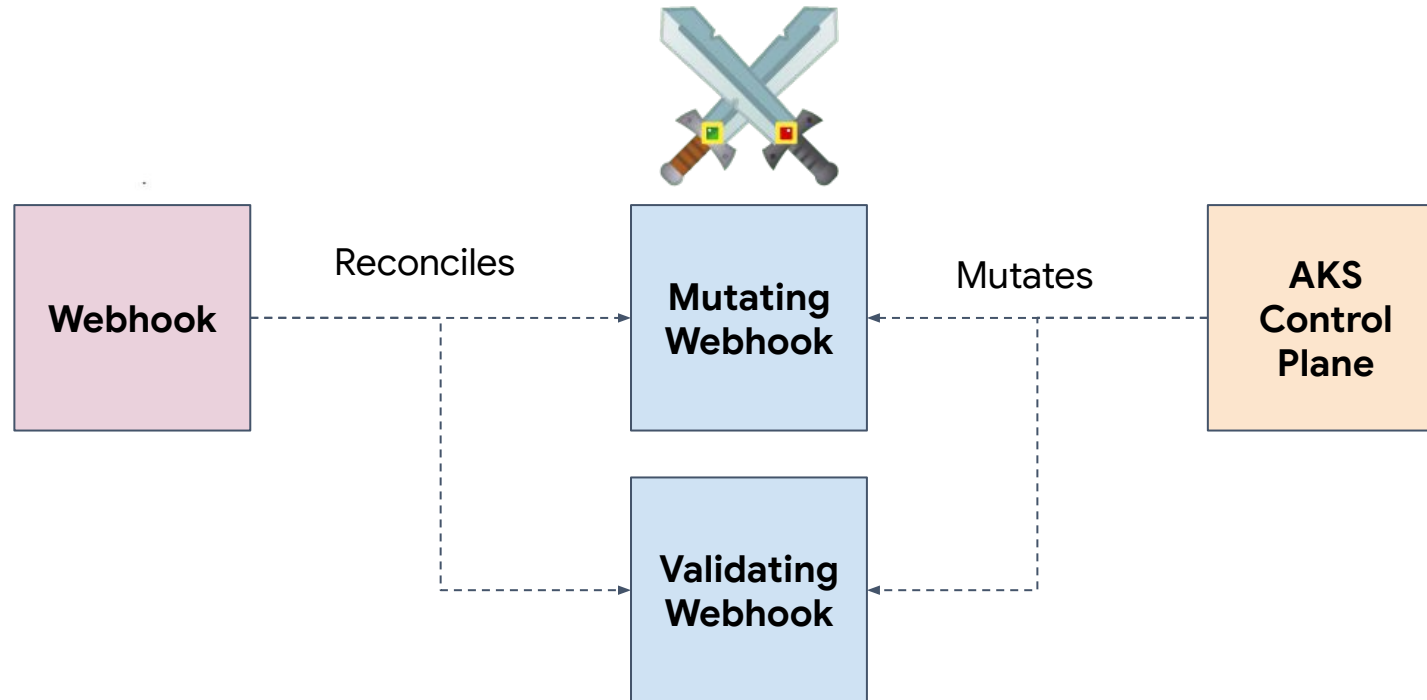
KubeCon



CloudNativeCon

North America 2023

- Our webhooks dynamically reconcile rules and clientConfig



Knative Contention Points



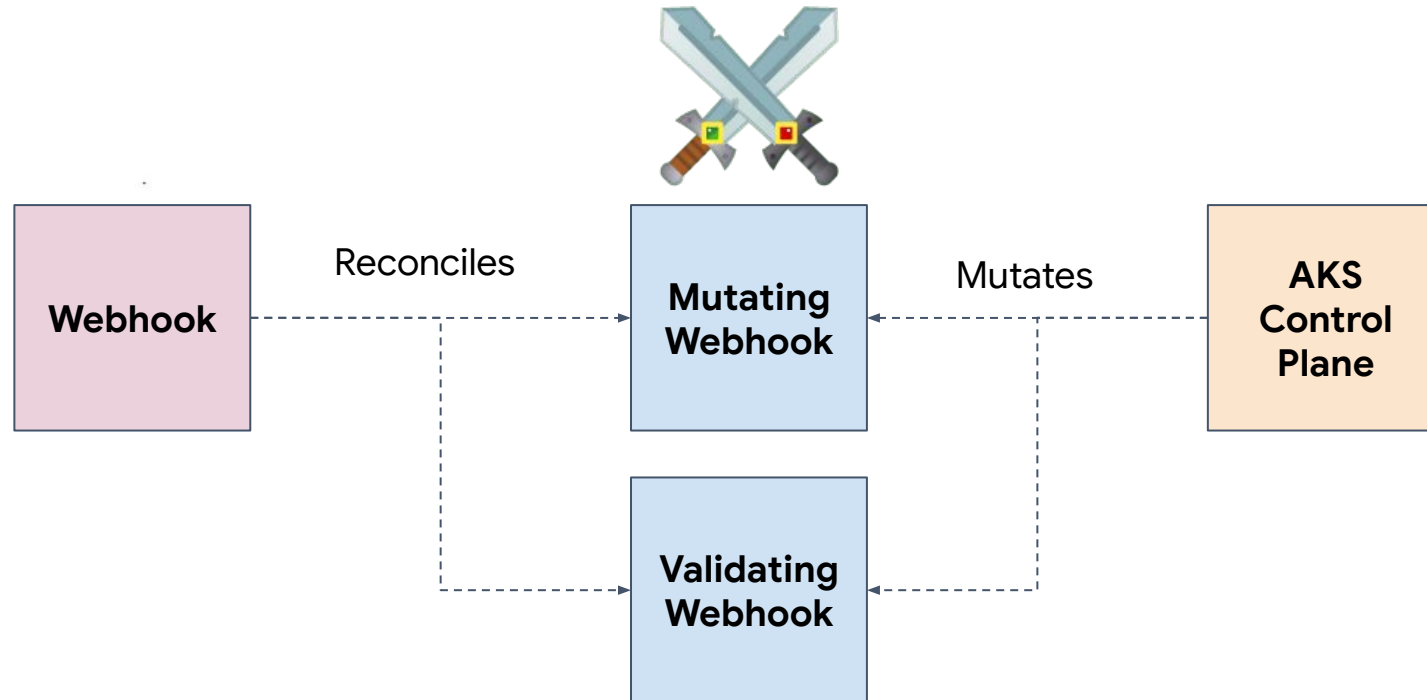
KubeCon



CloudNativeCon

North America 2023

- Our webhooks dynamically reconcile rules and clientConfig



```
...
namespaceSelector:
  matchExpressions:
    - key: control-plane
      operator: DoesNotExist
...
```

Knative Contention Points



KubeCon



CloudNativeCon

North America 2023



Knative Contention Points

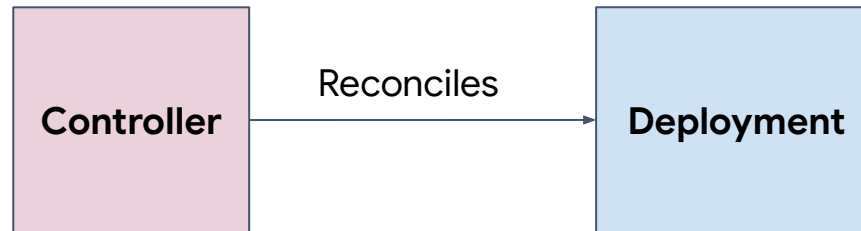


KubeCon



CloudNativeCon

North America 2023



Knative Contention Points

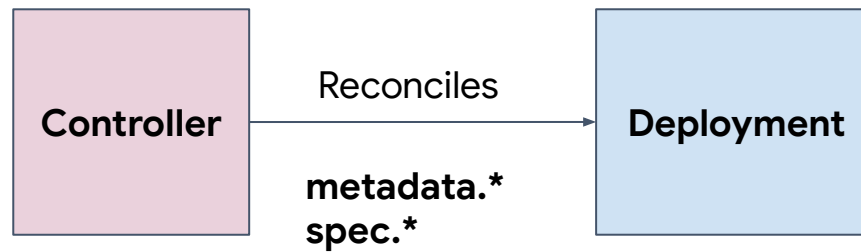


KubeCon



CloudNativeCon

North America 2023



Knative Contention Points

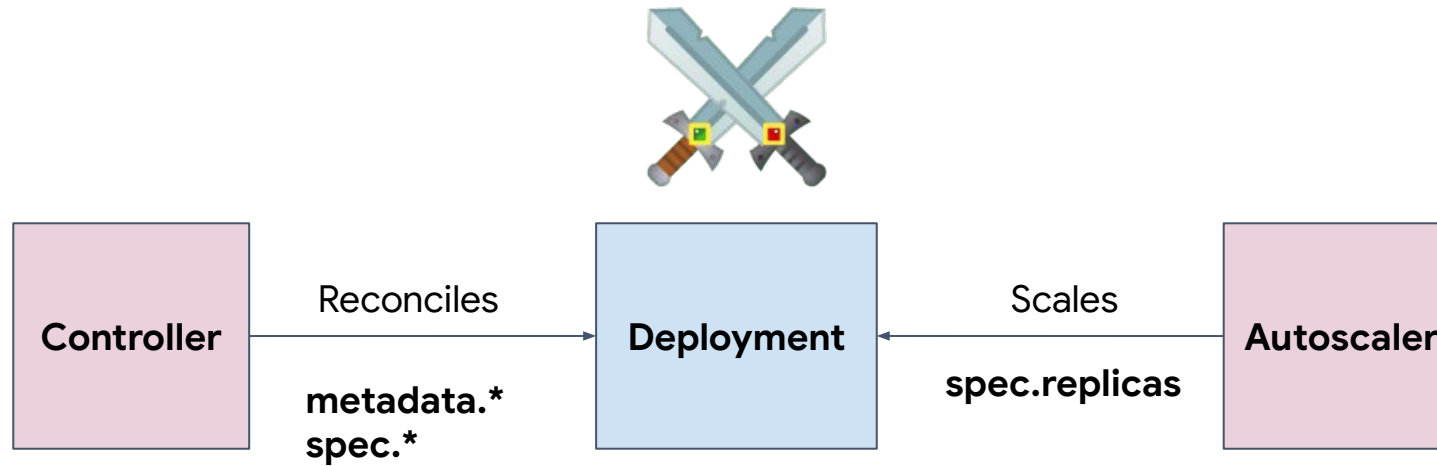


KubeCon



CloudNativeCon

North America 2023



Knative Contention Points

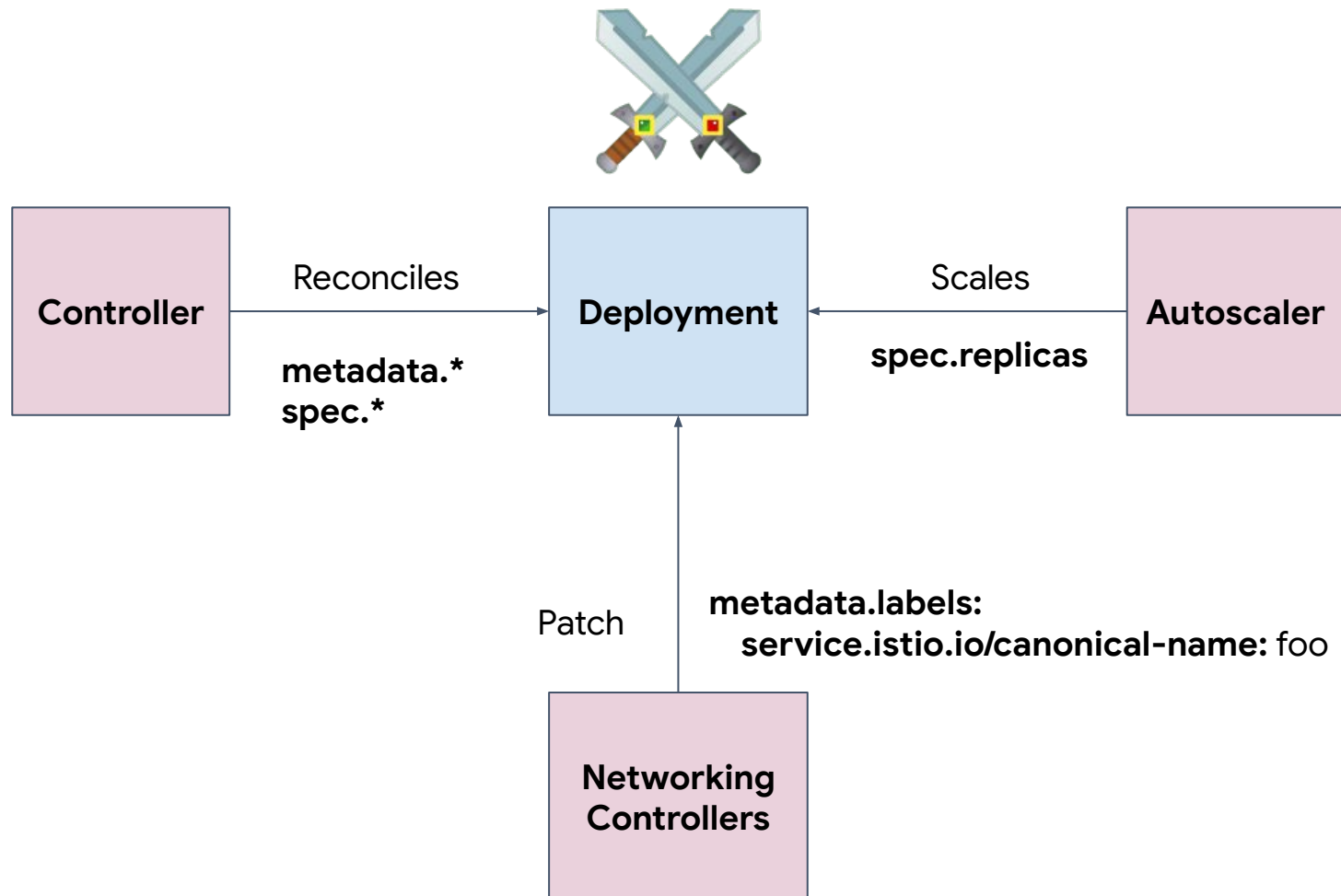


KubeCon



CloudNativeCon

North America 2023



Knative Contention Points

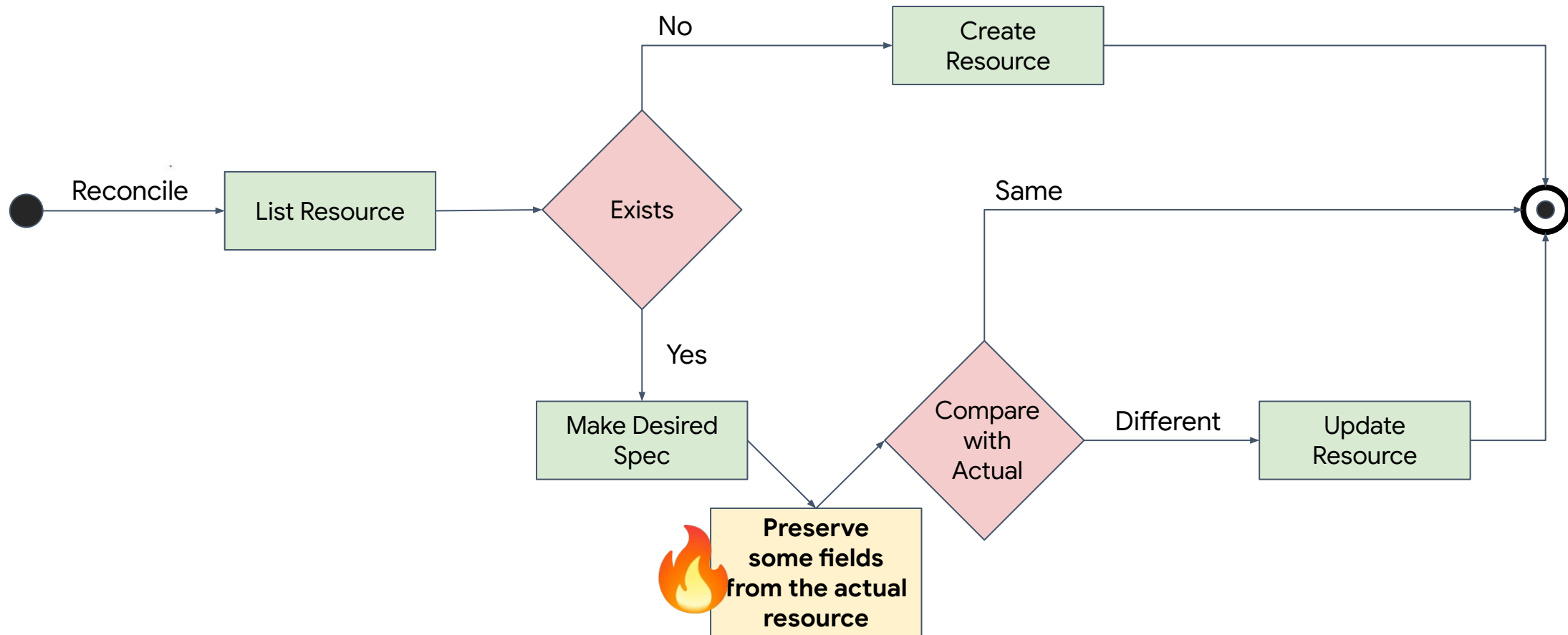


KubeCon



CloudNativeCon

North America 2023



Knative Contention Points



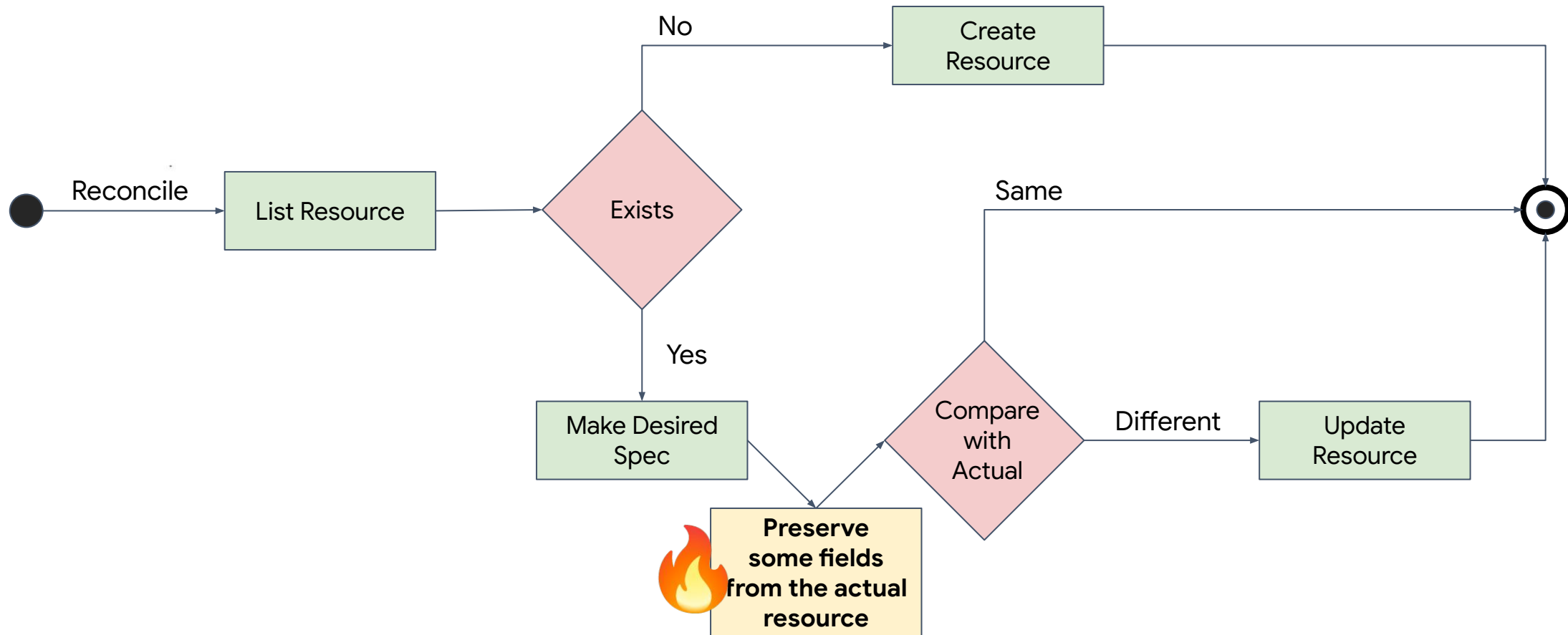
KubeCon



CloudNativeCon

North America 2023

- Do we still need Server-Side Apply?



Knative Contention Points



KubeCon



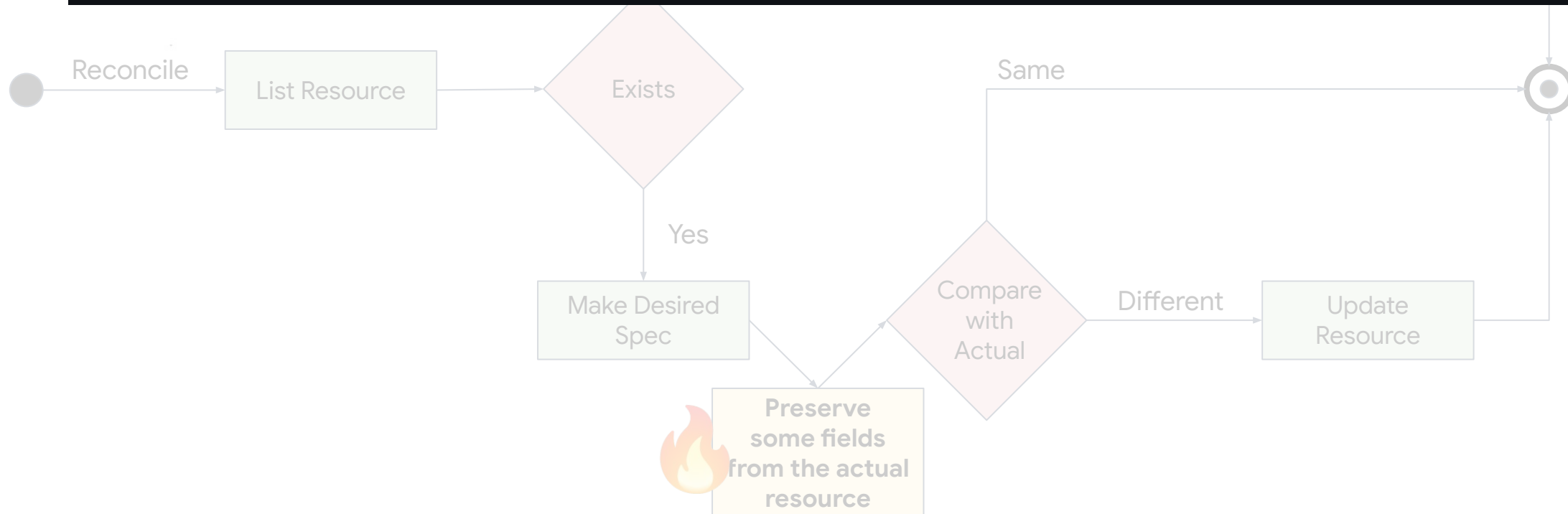
CloudNativeCon

North America 2023

Revision reconciler always updates the Kubernetes Deployment #13204



SaschaSchwarze0 opened this issue on Aug 12, 2022 · 15 comments



Knative Contention Points



KubeCon



CloudNativeCon

North America 2023

Revision reconciler always updates the Kubernetes Deployment #13204



SaschaSchwarze0 opened this issue on Aug 12, 2022 · 15 comments

```
// If the spec we want is the spec we have, then we're good.  
if equality.Semantic.DeepEqual(have.Spec, deployment.Spec) {  
    return have, nil  
}
```



This check will never return true. The reasons are the following:

- 1. Defaulting.** Knative sets up the deployment and does not set fields at various places which Kubernetes will default to some value. Those differences are always there. Examples are: the `Protocol` field in ports which gets set to `TCP`,

resource

Knative Contention Points

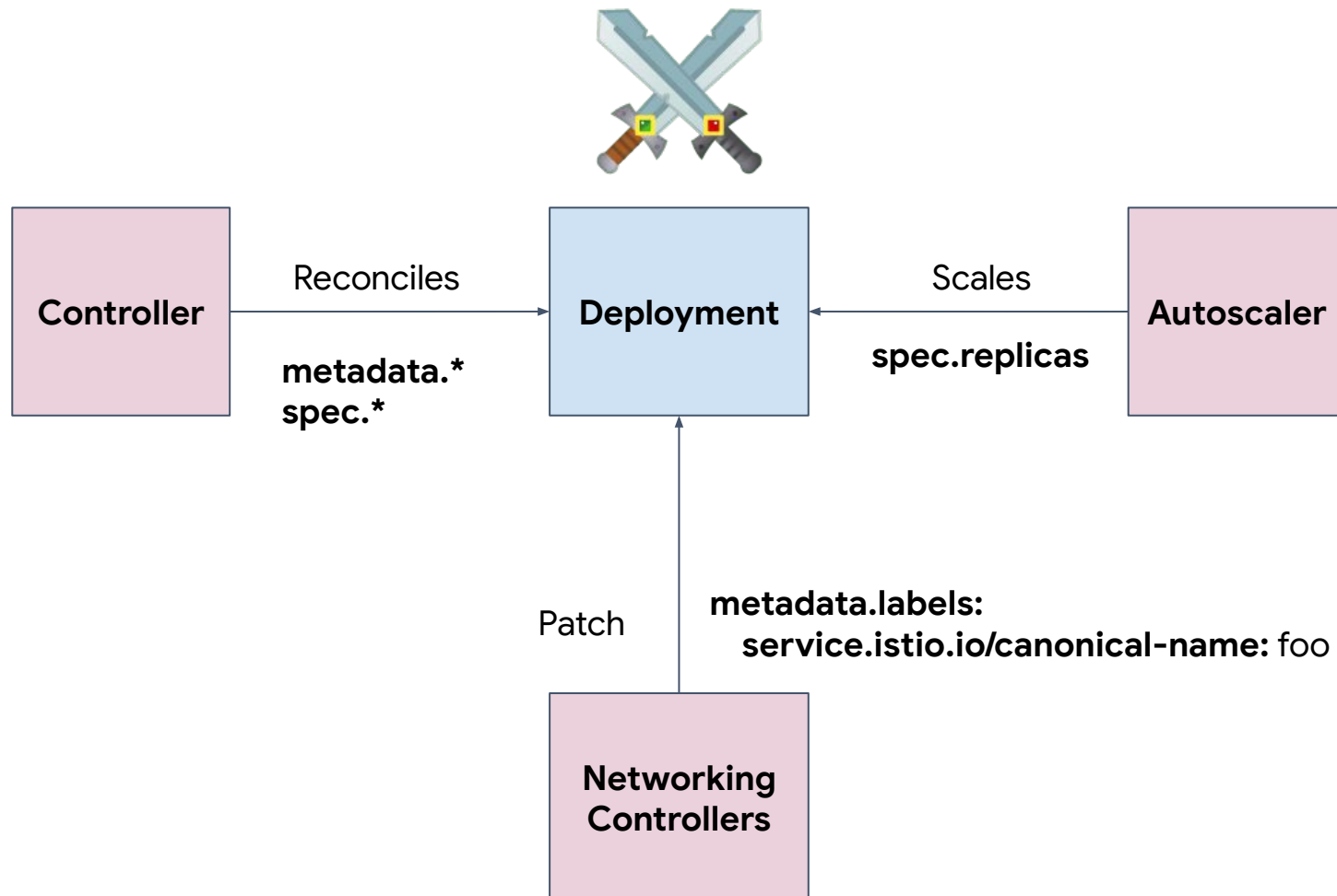


KubeCon



CloudNativeCon

North America 2023



Knative Contention Points

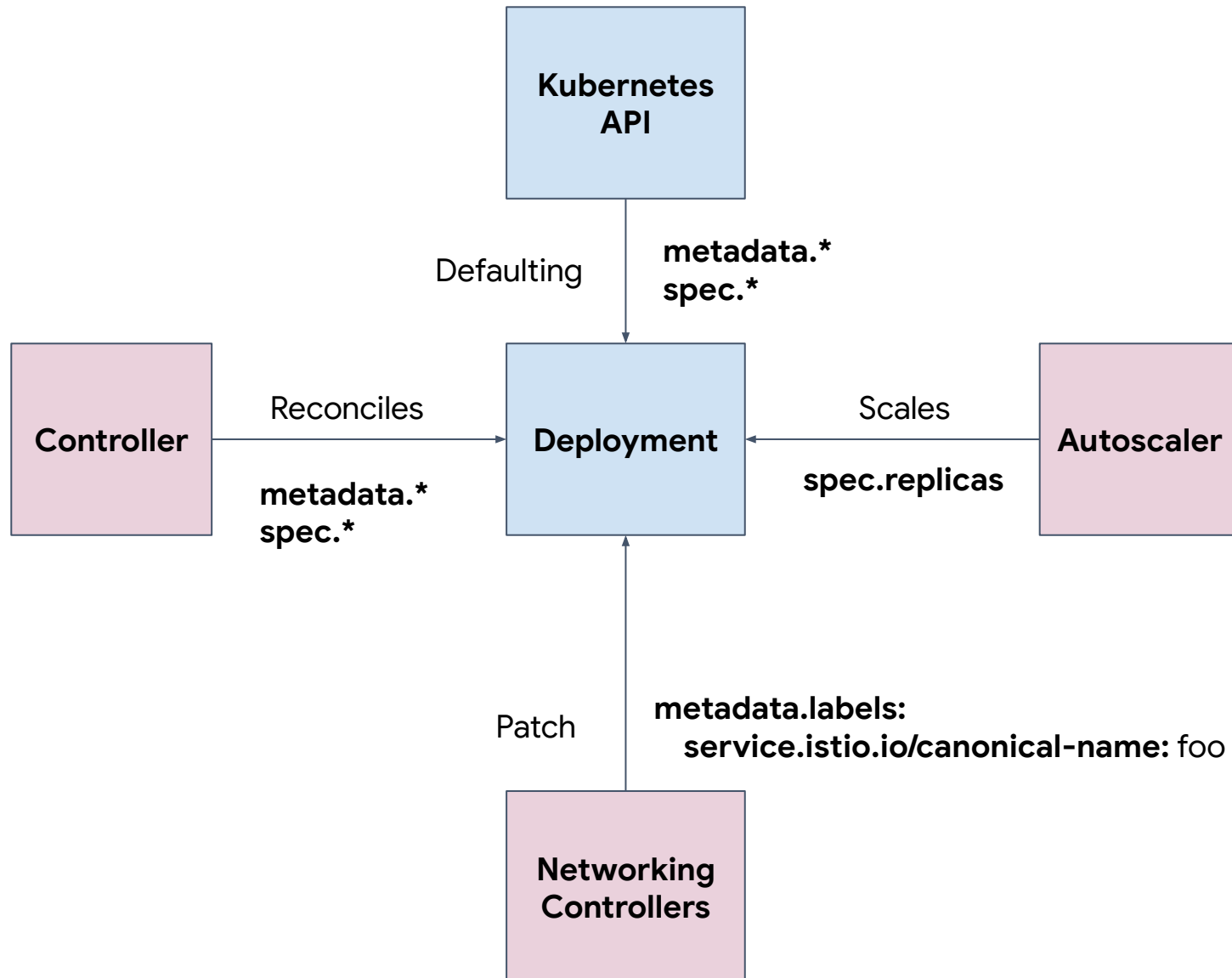


KubeCon



CloudNativeCon

North America 2023



Knative Contention Points

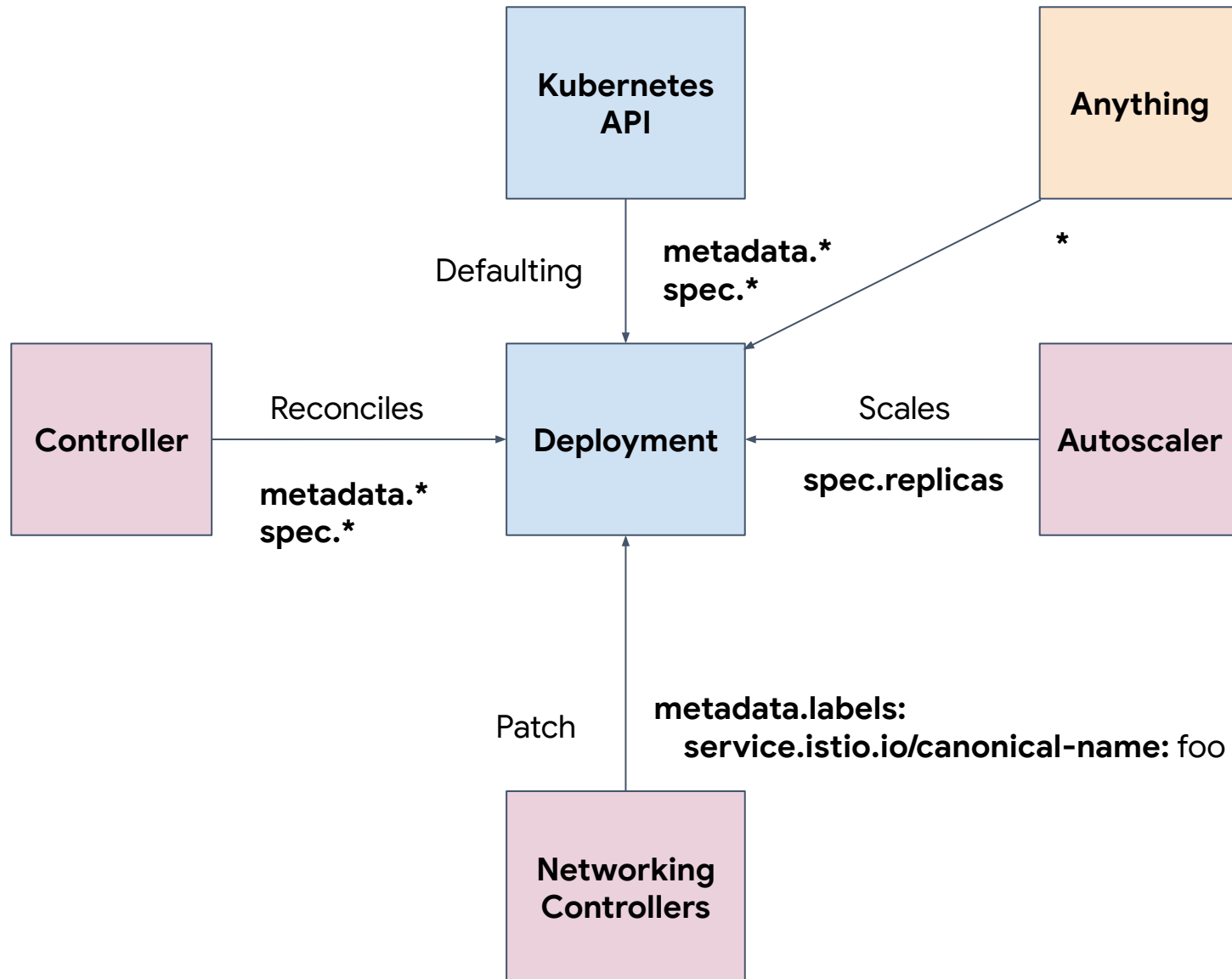


KubeCon



CloudNativeCon

North America 2023



Defaulting Contention



KubeCon



CloudNativeCon

North America 2023

```
apiVersion: v1
kind: Pod
metadata:
  name: nginx
spec:
  containers:
  - name: nginx
    image: nginx:1.25.2
    ports:
    - containerPort: 80
```

kubectl apply -f -

Defaulting Contention



KubeCon



CloudNativeCon

North America 2023

```
apiVersion: v1
kind: Pod
metadata:
  name: nginx
spec:
  containers:
  - name: nginx
    image: nginx:1.25.2
    ports:
    - containerPort: 80
```



```
apiVersion: v1
fieldsType: FieldsV1
fieldsV1:
  f:spec:
    f:containers:
      k:{"name":"nginx"}:
        .: {}
      f:image: {}
      f:imagePullPolicy: {}
      f:name: {}
      f:ports:
        .: {}
        k:{"containerPort":80,"protocol":"TCP"}:
          .: {}
          f:containerPort: {}
          f:protocol: {}
      f:resources: {}
      f:terminationMessagePath: {}
      f:terminationMessagePolicy: {}
    f:dnsPolicy: {}
    f:enableServiceLinks: {}
    f:restartPolicy: {}
    f:schedulerName: {}
    ...
```

kubectl apply -f -

Defaulting Contention



KubeCon



CloudNativeCon

North America 2023

```
apiVersion: v1
kind: Pod
metadata:
  name: nginx
spec:
  containers:
  - name: nginx
    image: nginx:1.25.2
    ports:
    - containerPort: 80
```

```
kubectl apply -f -
--server-side
```

Defaulting Contention



KubeCon



CloudNativeCon

North America 2023

```
apiVersion: v1
kind: Pod
metadata:
  name: nginx
spec:
  containers:
  - name: nginx
    image: nginx:1.25.2
    ports:
    - containerPort: 80
```



```
apiVersion: v1
fieldsType: FieldsV1
fieldsV1:
  f:spec:
    f:containers:
      k:{"name":"nginx"}:
        .: {}
      f:image: {}
      f:name: {}
      f:ports:
        .: {}
        k:{"containerPort":80,"protocol":"TCP"}:
          .: {}
          f:containerPort: {}
```

kubectl apply -f -
--server-side

Defaulting Contention



KubeCon



CloudNativeCon

North America 2023

```
apiVersion: v1
fieldsType: FieldsV1
fieldsV1:
  f:spec:
    f:containers:
      k:{"name":"nginx"}:
        .: {}
      f:image: {}
      f:imagePullPolicy: {}
      f:name: {}
      f:ports:
        .: {}
      k:{"containerPort":80,"protocol":"TCP"}:
        .: {}
        f:containerPort: {}
        f:protocol: {}
      f:resources: {}
      f:terminationMessagePath: {}
      f:terminationMessagePolicy: {}
    f:dnsPolicy: {}
    f:enableServiceLinks: {}
    f:restartPolicy: {}
    f:schedulerName: {}
    ...
```

```
apiVersion: v1
fieldsType: FieldsV1
fieldsV1:
  f:spec:
    f:containers:
      k:{"name":"nginx"}:
        .: {}
      f:image: {}
      f:name: {}
      f:ports:
        .: {}
      k:{"containerPort":80,"protocol":"TCP"}:
        .: {}
        f:containerPort: {}
```

Resource + API Changes



KubeCon



CloudNativeCon

North America 2023

- Serving API ~**2018**
- Server-Side Apply GA ~**2021**

Resource + API Changes

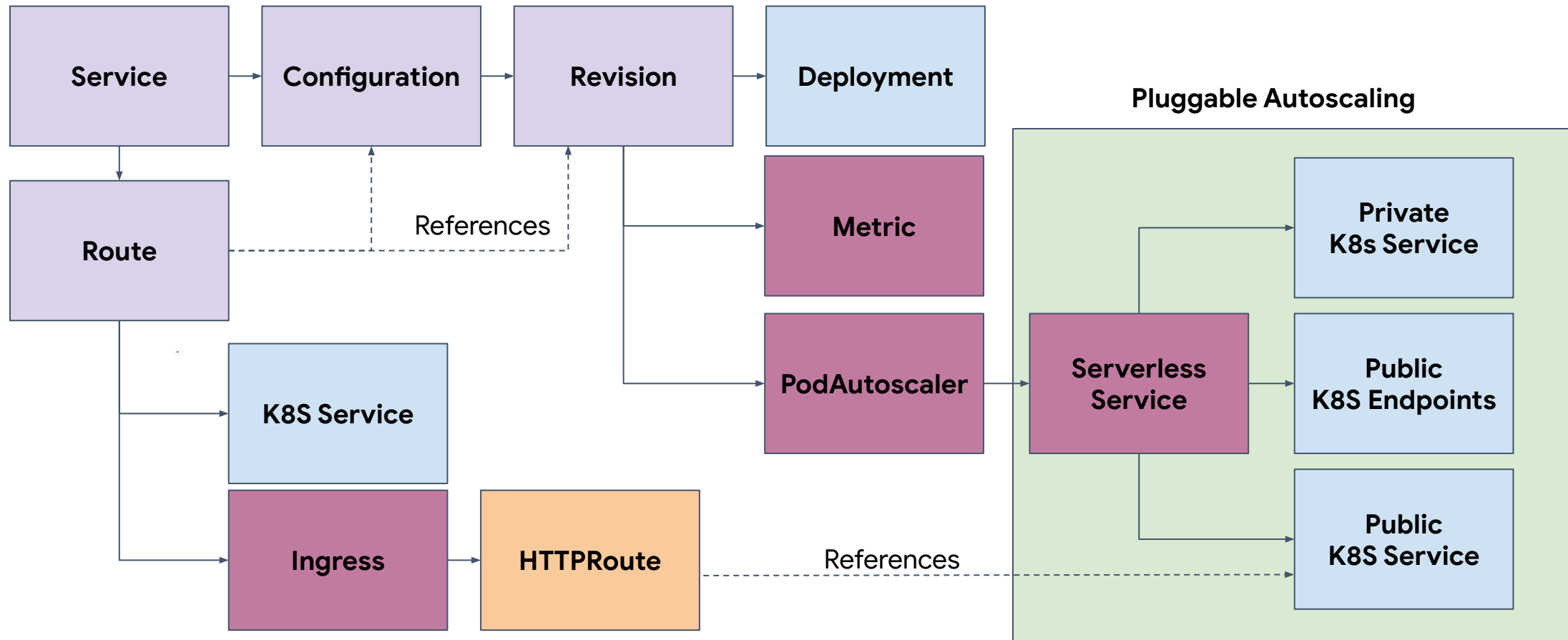


KubeCon



CloudNativeCon

North America 2023



Resource + API Changes

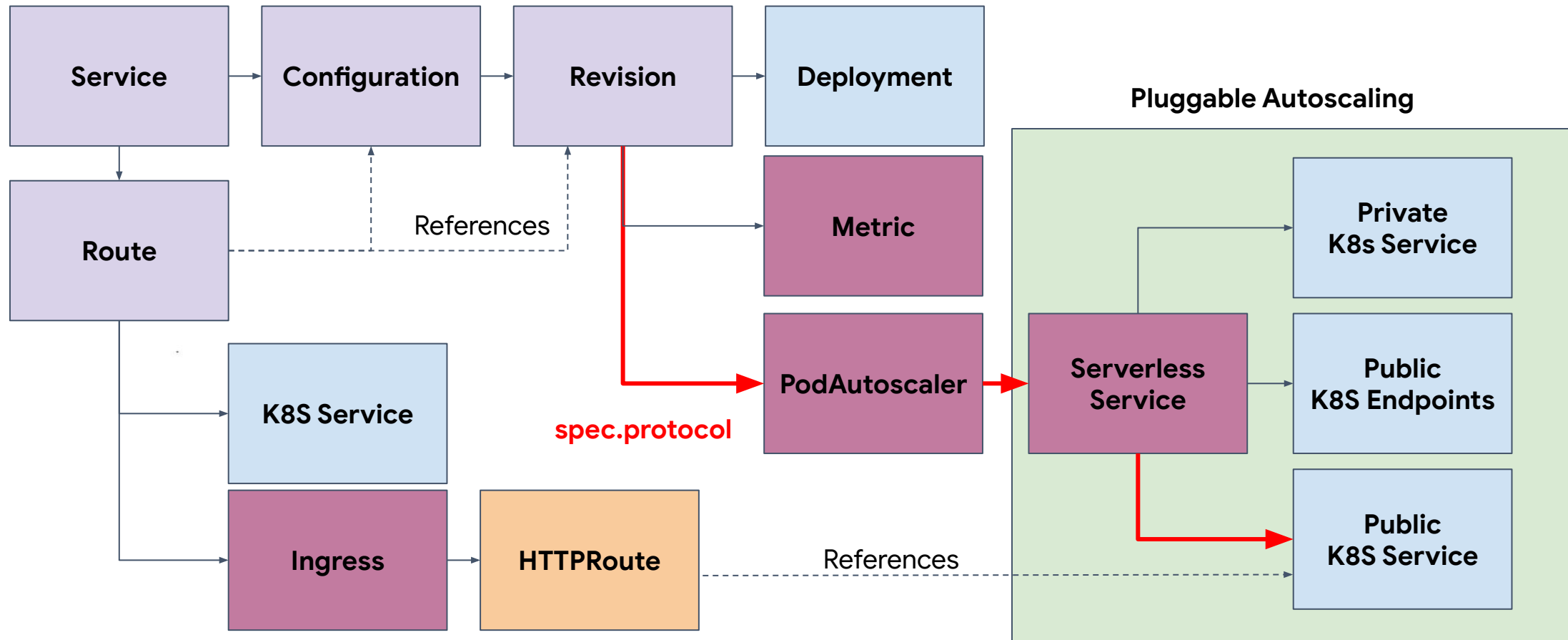


KubeCon



CloudNativeCon

North America 2023



Resource + API Changes

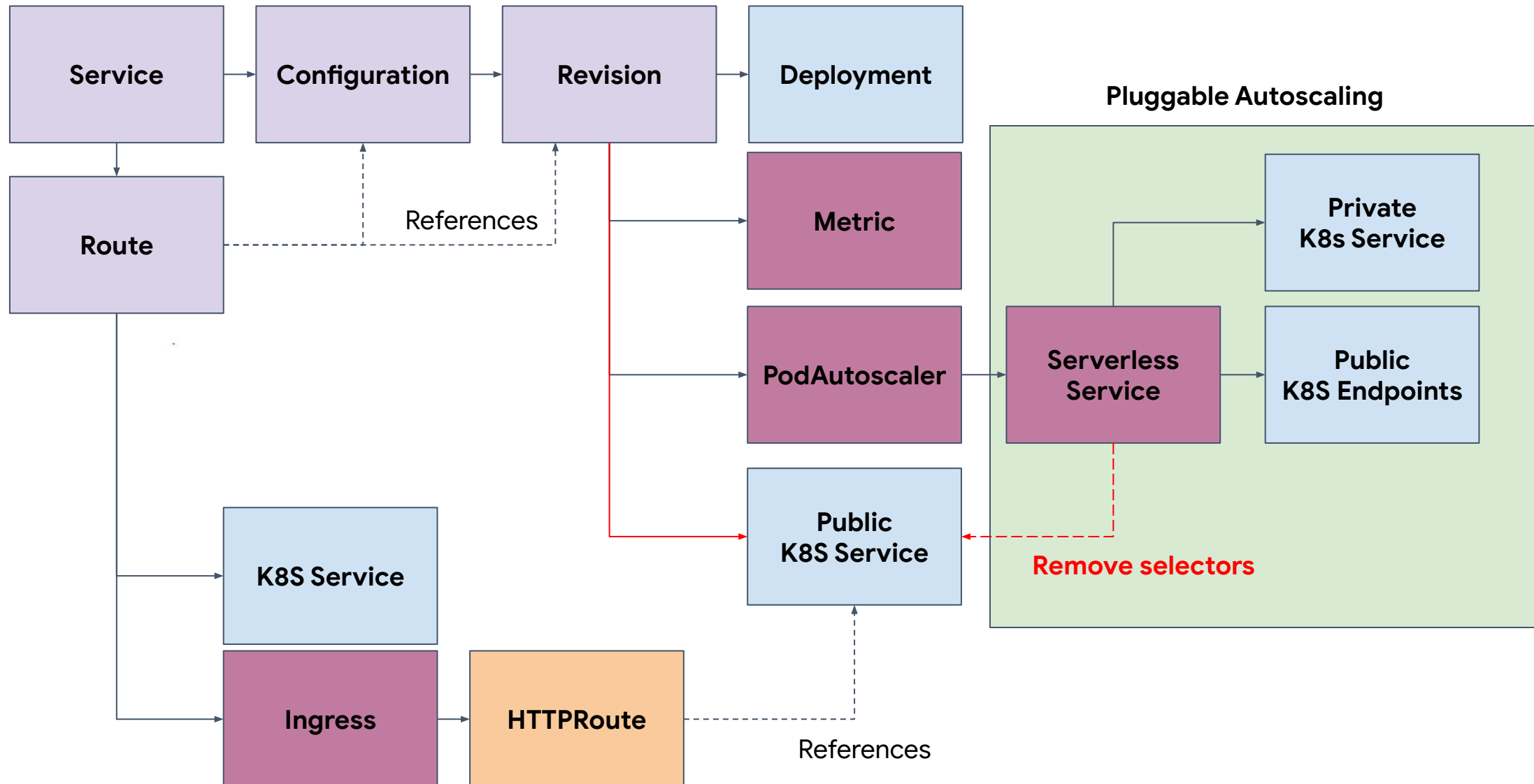


KubeCon



CloudNativeCon

North America 2023



Implementing SSA in Knative



KubeCon



CloudNativeCon

North America 2023

- Still prototyping in Serving



Implementing SSA in Knative



KubeCon



CloudNativeCon

North America 2023

- Still prototyping in Serving
- Rough Plan
 - Webhooks
 - Deployment Reconciliation
 - Autoscaler
 - Compare Performance
 - Knative CRDs





- GO Types - ApplyConfiguration
 - `k8s.io/client-go/applyconfigurations`
 - `k8s.io/code-generator`
 - `applyconfiguration-gen`

- GO Types - ApplyConfiguration
 - `k8s.io/client-go/applyconfigurations`
 - `k8s.io/code-generator`
 - `applyconfiguration-gen`
- Update Clientsets with Apply() methods
 - `--apply-configuration-package`

- GO Types - ApplyConfiguration
 - `k8s.io/client-go/applyconfigurations`
 - `k8s.io/code-generator`
 - `applyconfiguration-gen`
- Update Clientsets with Apply() methods
 - `--apply-configuration-package`
- Tooling needs a bit of work with Custom Resources

- Example:

<https://github.com/kubernetes-sigs/gateway-api/pull/2537>



- Unit Testing
 - Knative tests our reconcilers without spinning up K8s locally
 - Use client-go fakes

- Unit Testing
 - Knative tests our reconcilers without spinning up K8s locally
 - Use client-go fakes
- SSA support for fakes is in progress
<https://github.com/kubernetes/kubernetes/pull/118211>

Learnings



KubeCon



CloudNativeCon

- Migrating from Update to Apply
- With same manager **requires forcing conflict**

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: demo
  managedFields:
    - apiVersion: v1
      manager: dave
      operation: Update
      time: "2023-10-22T00:00:05Z"
      fieldsType: FieldsV1
      fieldsV1:
        f:data:
          f:City: {}
    - ...
data:
  Speaker: Dave P
  City: Chicago
  Conference: KubeCon NA
```





- K8s blog post suggests extract flow for controllers
 - **get** the existing object
 - extract your existing desires
 - make modifications, and re-**apply**



- K8s blog post suggests extract flow for controllers
 - **get** the existing object
 - extract your existing desires
 - make modifications, and re-**apply**
- *can still trigger excess updates*

- my current thinking to elide updates
 - generate apply config
 - **get** the existing object
 - extract your existing desires
 - prune your apply config based on what fields you own
 - **apply** if the pruned config != existing config

To Be Continued...



KubeCon



CloudNativeCon

North America 2023

- Work is ongoing in Knative and upstream K8s
- Big thank you to all the K8s maintainers landing such an amazing feature
 - API-Machinery Maintainer Talk
 - <https://sched.co/1R2my>
 - Get involved!
 - SSA Maintainers have ideas on future improvements
 - <https://docs.google.com/document/d/1Qdw24Kx2O5YXyoGkzTqmLSA4tD5g50QOVu1dh0yxDEo/edit>



PromCon
North America 2021



**Please scan the QR Code above
to leave feedback on this session**