KubeCon | CloudNativeCon

North America 2023

KubeCon | CloudNativeCon

North America 2023

Building Better Controllers

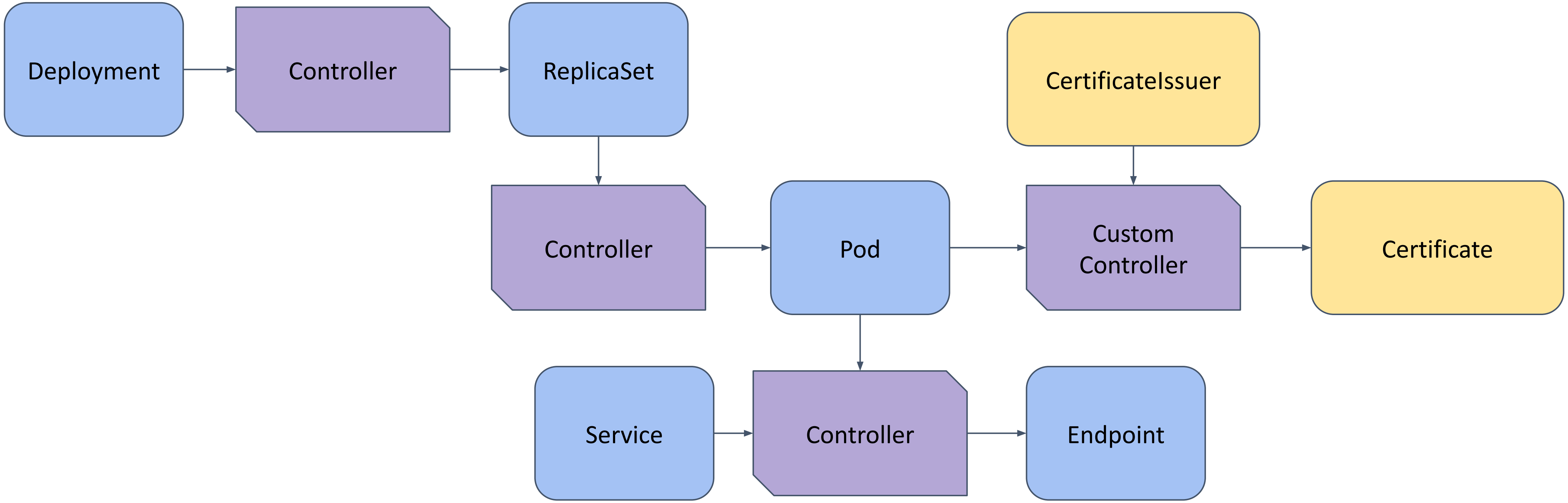
John Howard, Google

Overview

- What *are* controllers
- Writing controllers is really hard
- Challenges writing controllers for Istio
- A ~~better~~ different approach to writing controllers

What are controllers?

"Controllers are control loops that watch the state of your cluster, then make or request changes where needed. Each controller tries to move the current cluster state closer to the desired state."



Writing controllers is *hard*

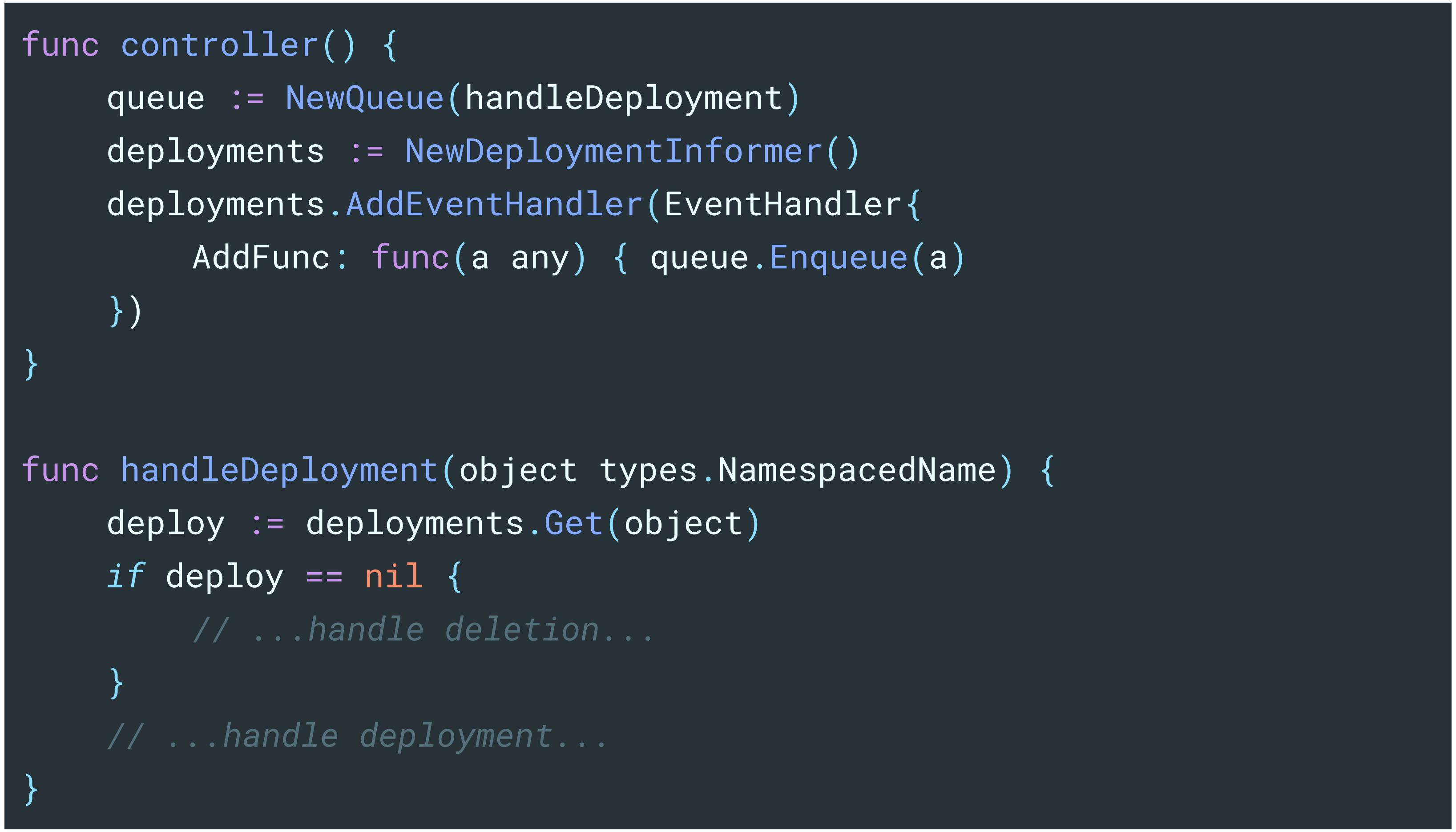
Most controllers are *very* low level.

Main primitive is the `Informer`, allowing us to subscribe to events:

```go
type EventHandler struct {
 AddFunc    func(obj any)
 UpdateFunc func(oldObj, newObj any)
 DeleteFunc func(obj any)
}
```

Writing controllers is *hard*

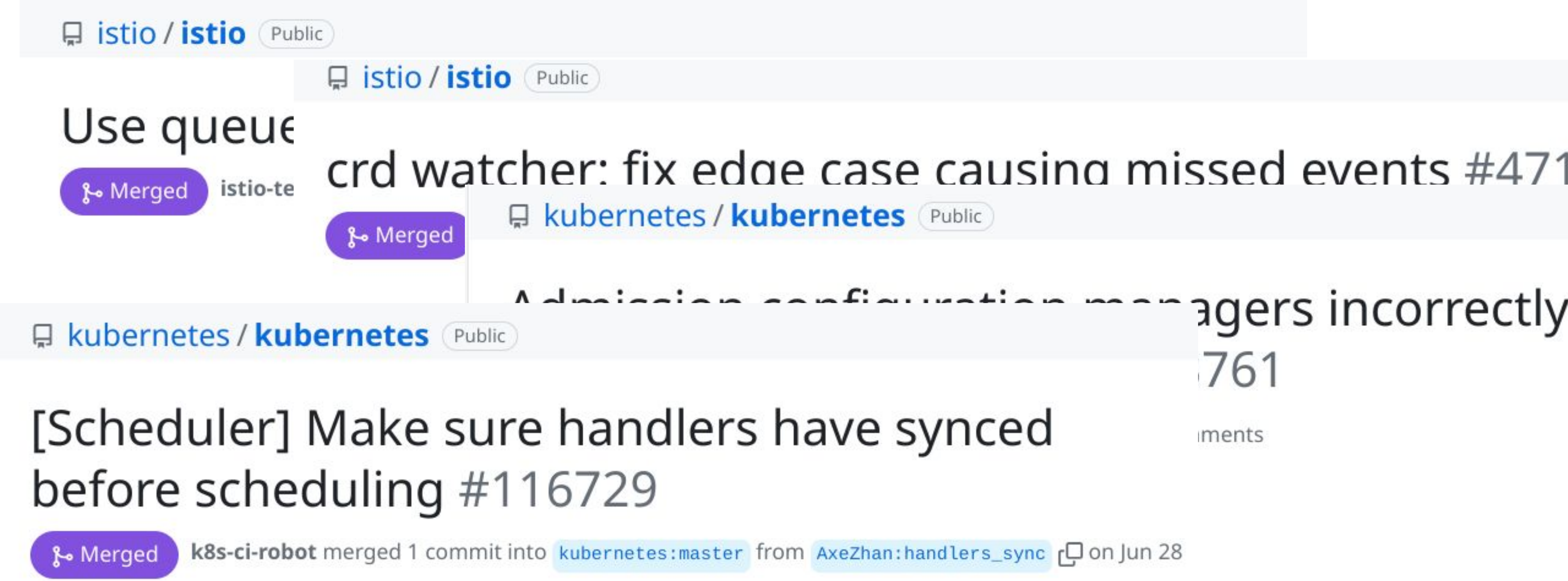
Real world code is 400 LOC for a sample



```go
func controller() {
    queue := NewQueue(handleDeployment)
    deployments := NewDeploymentInformer()
    deployments.AddEventHandler(EventHandler{
        AddFunc: func(a any) { queue.Enqueue(a)
    })
}

func handleDeployment(object types.NamespacedName) {
    deploy := deployments.Get(object)
    if deploy == nil {
        // ...handle deletion...
    }
    // ...handle deployment...
}
```

Writing controllers is *hard*



istio / istio Public

Use queue

Merged istio-te

istio / istio Public

crd watcher: fix edge case causing missed events #471

Merged

kubernetes / kubernetes Public

Admission configuration managers incorrectly

761

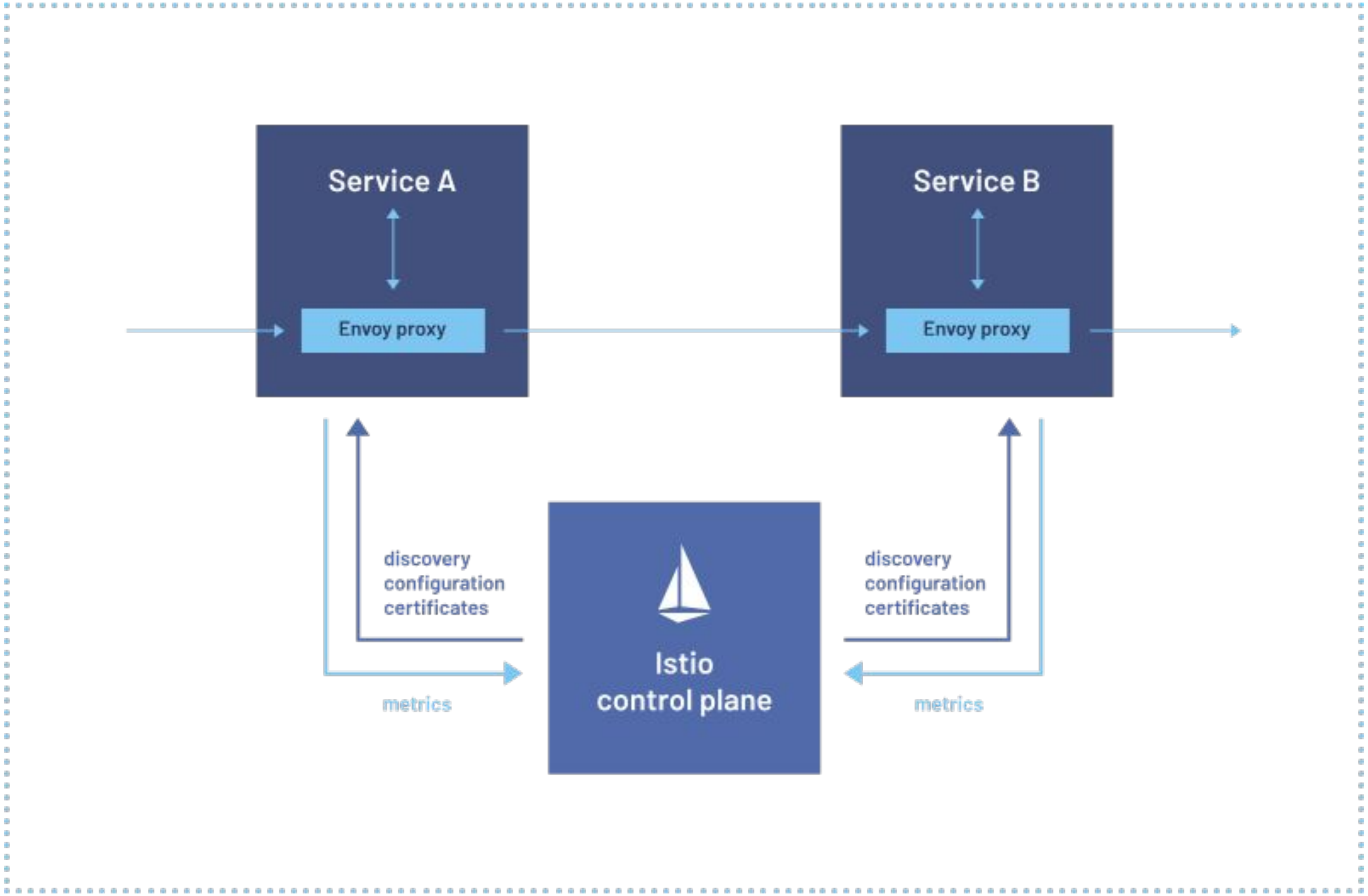
ments

kubernetes / kubernetes Public

[Scheduler] Make sure handlers have synced before scheduling #116729

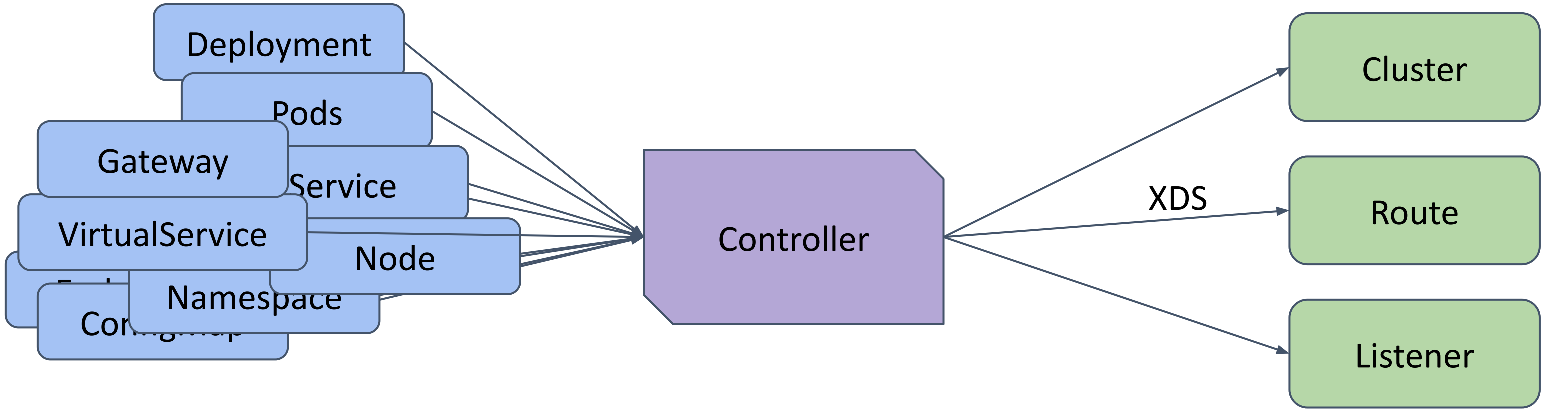
Merged k8s-ci-robot merged 1 commit into `kubernetes:master` from `AxeZhan:handlers_sync` on Jun 28

Example: Istio Service Mesh



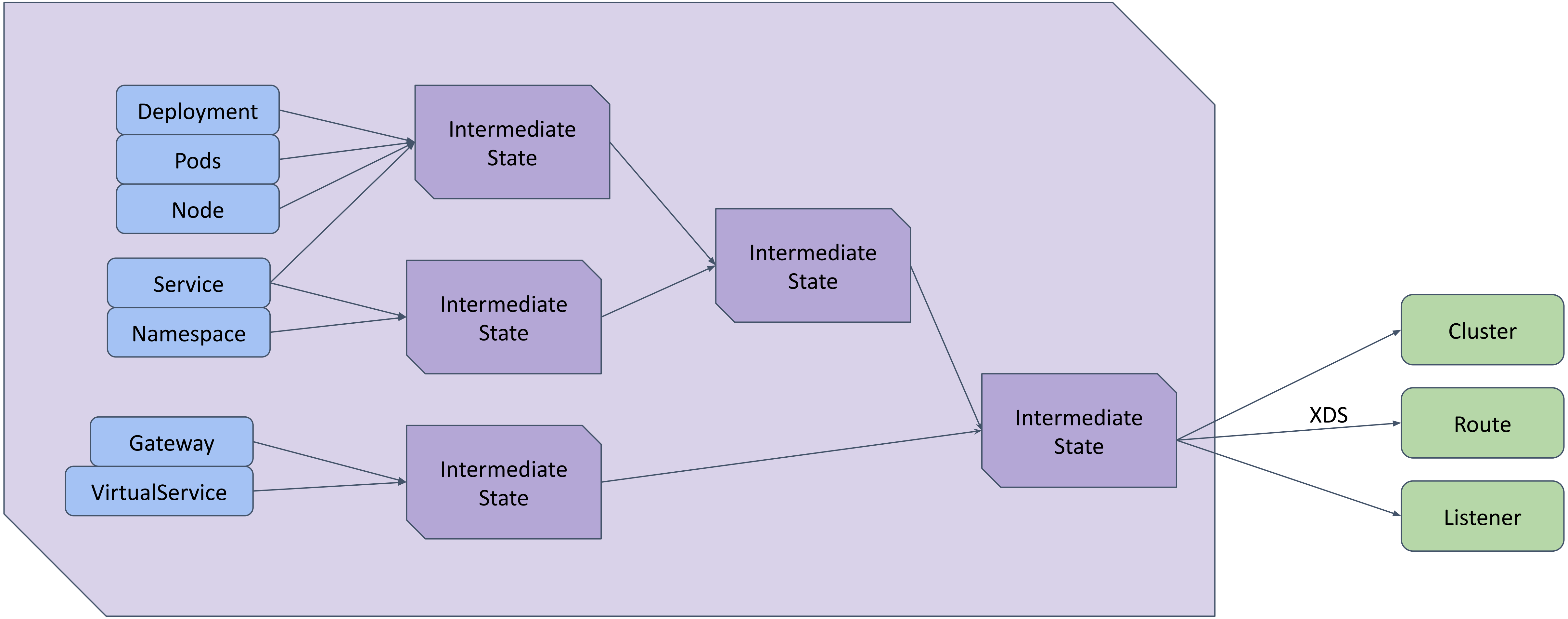
Istio Challenges

Istio is primarily one giant controller, but unique.



- Lots of inputs/outputs
- Intermediate state not persisted in cluster
    - Or memory!
- Outputs large (megabytes), and written to many places

Istio Implementation

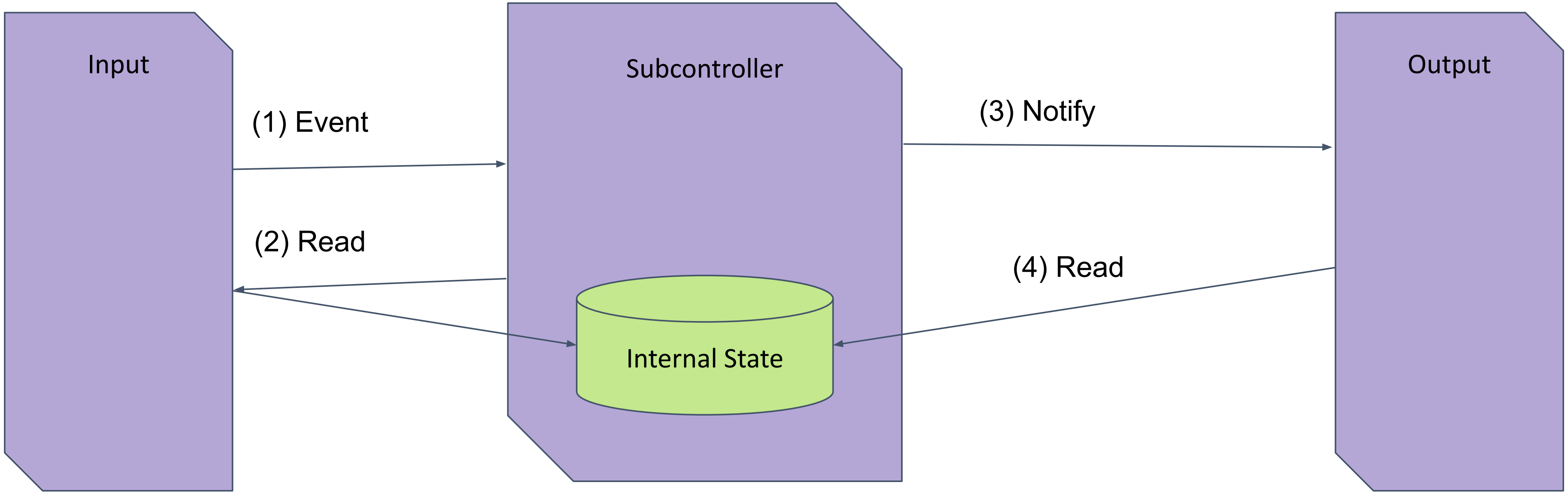


State management

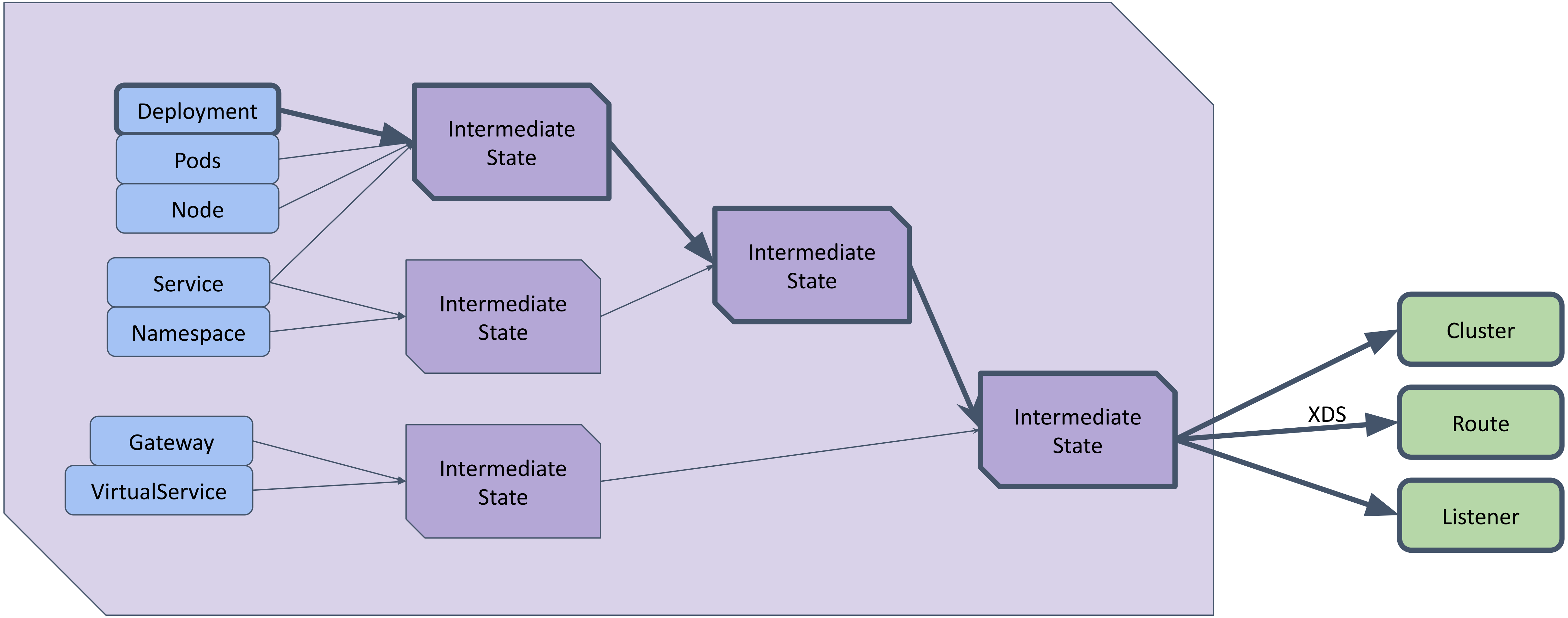
Majority of Istio codebase consists of:

- Receive event that something has changed.
- Reconcile internal state with the change.
- Notify dependencies something changed
- Dependency queries our new internal state

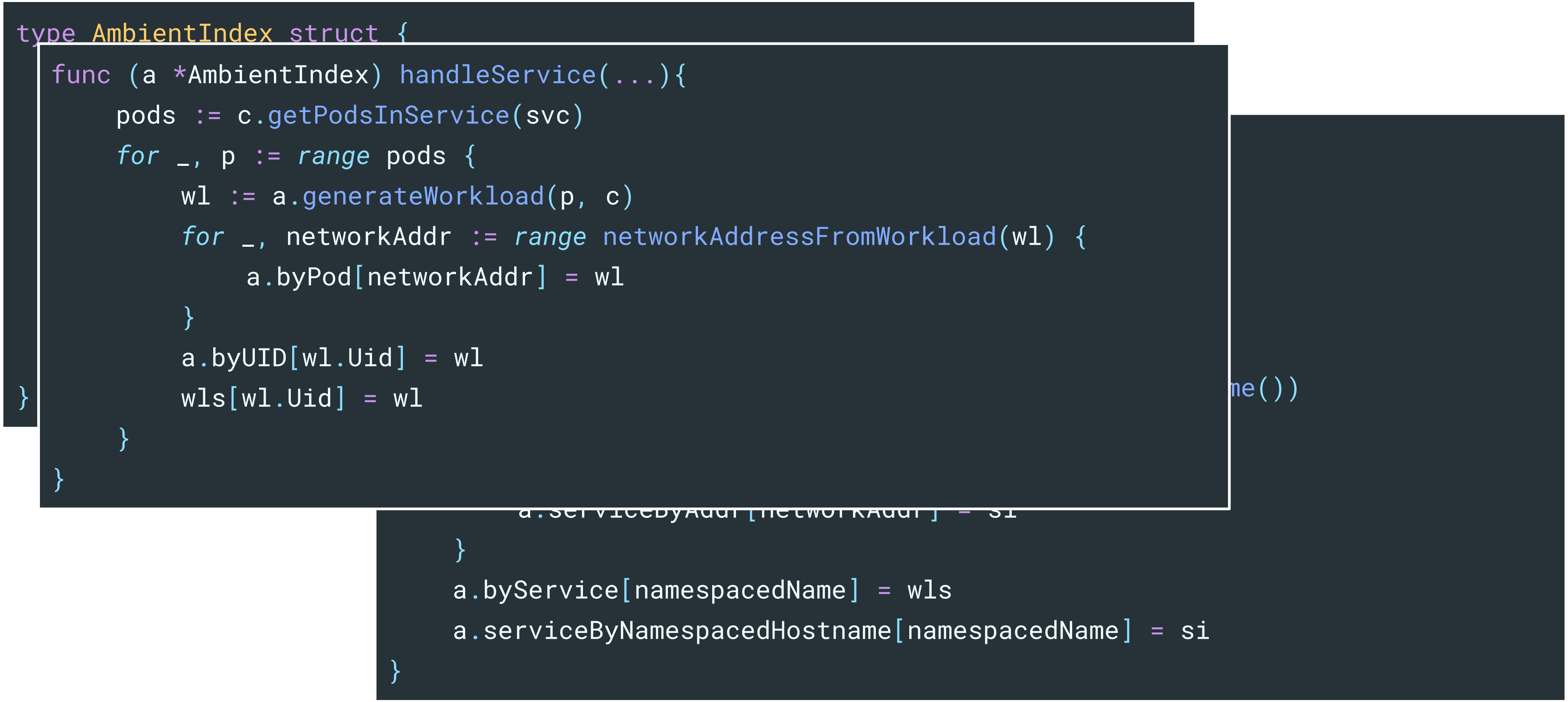
Most controllers utilize Kubernetes as intermediate state to get this for "free"



Event detection



Istio Implementation

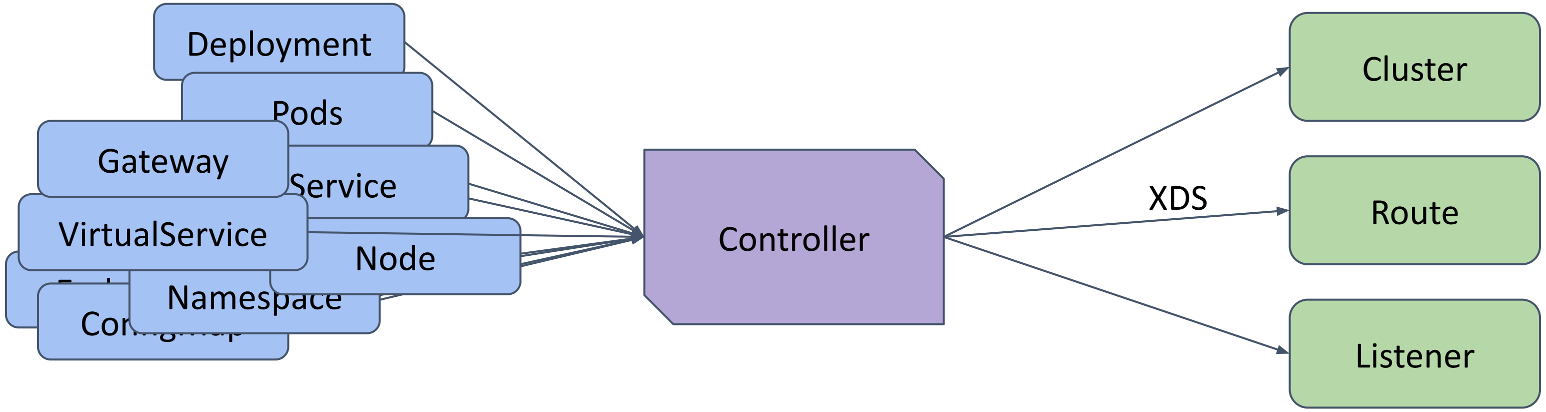


```
type AmbientIndex struct {
}
```

```
func (a *AmbientIndex) handleService(...){
    pods := c.getPodsInService(svc)
    for _, p := range pods {
        wl := a.generateWorkload(p, c)
        for _, networkAddr := range networkAddressFromWorkload(wl) {
            a.byPod[networkAddr] = wl
        }
        a.byUID[wl.Uid] = wl
        wls[wl.Uid] = wl
    }
}
```

```
me())
        a.serviceByAddr[networkAddr] = si
    }
    a.byService[namespacedName] = wls
    a.serviceByNamespacedHostname[namespacedName] = si
}
```

Istio


Deployment
Pods
Gateway
Service
VirtualService
Node
Namespace
Controller
XDS
Cluster
Route
Listener

Goals

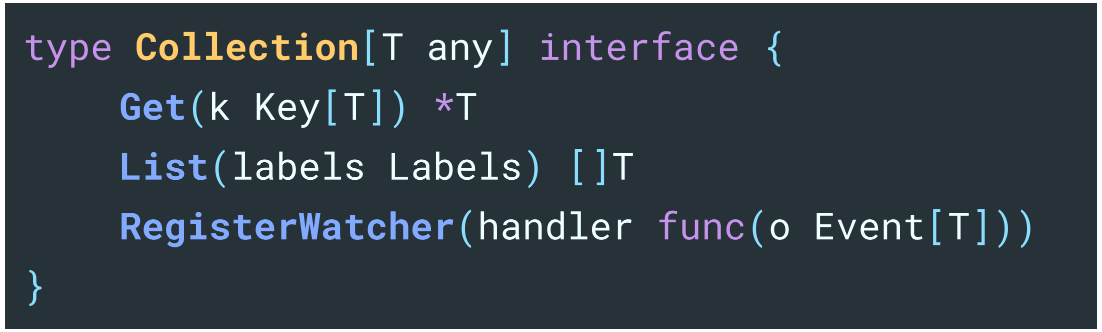
- Easy to write **correctly** and **efficiently**
  - The obvious implementation should be the correct and efficient one.
- **High level**
  - The controller should describe business logic, and shouldn't be concerned with low level state management.
- **Composable**
  - Controllers should be able to build upon each other, even if they are not persisted to Kubernetes.

Goals

DECLARATIVE APIS
IMPERATIVE CONTROLLERS

Implementation

Extremely simple core interface



```go
type Collection[T any] interface {
    Get(k Key[T]) *T
    List(labels Labels) []T
    RegisterWatcher(handler func(o Event[T]))
}
```

Implementation

Ecosystem of composable components built around this interface

Sources

- Collection from informer
- Collection from files
- Collection from in-memory objects
- Collection fetch from external state

Transformations

- Index
- Transformations
- Complex compositions

Outputs

- Write to Kubernetes
- Send over XDS
- Write to Cloud APIs
- Arbitrary event handler

Collection Creation

```go
ConfigMaps := FromInformer[ConfigMap]()
Config := NewCollection(func() Config {
    cfg := DefaultConfig()
    cms := []ConfigMap{
        // Fetch a single item from the ConfigMaps collection.
        FetchOne(ConfigMaps, filter.Name("config")),
        FetchOne(ConfigMaps, filter.Name("config-overrides"))
    }
    for _, c := range cms {
        cfg = cfg.Merge(c.data.Config)
    }
    return cfg
})
```

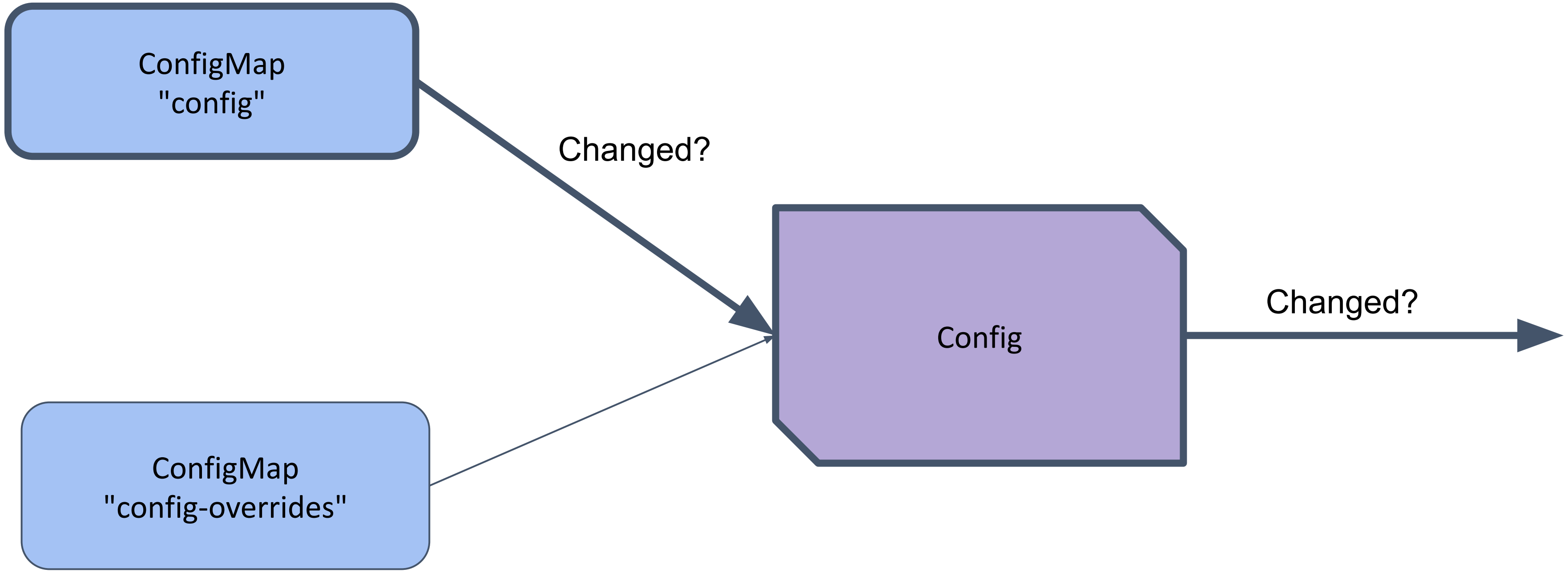
From Kubernetes Informer

Derived from various inputs

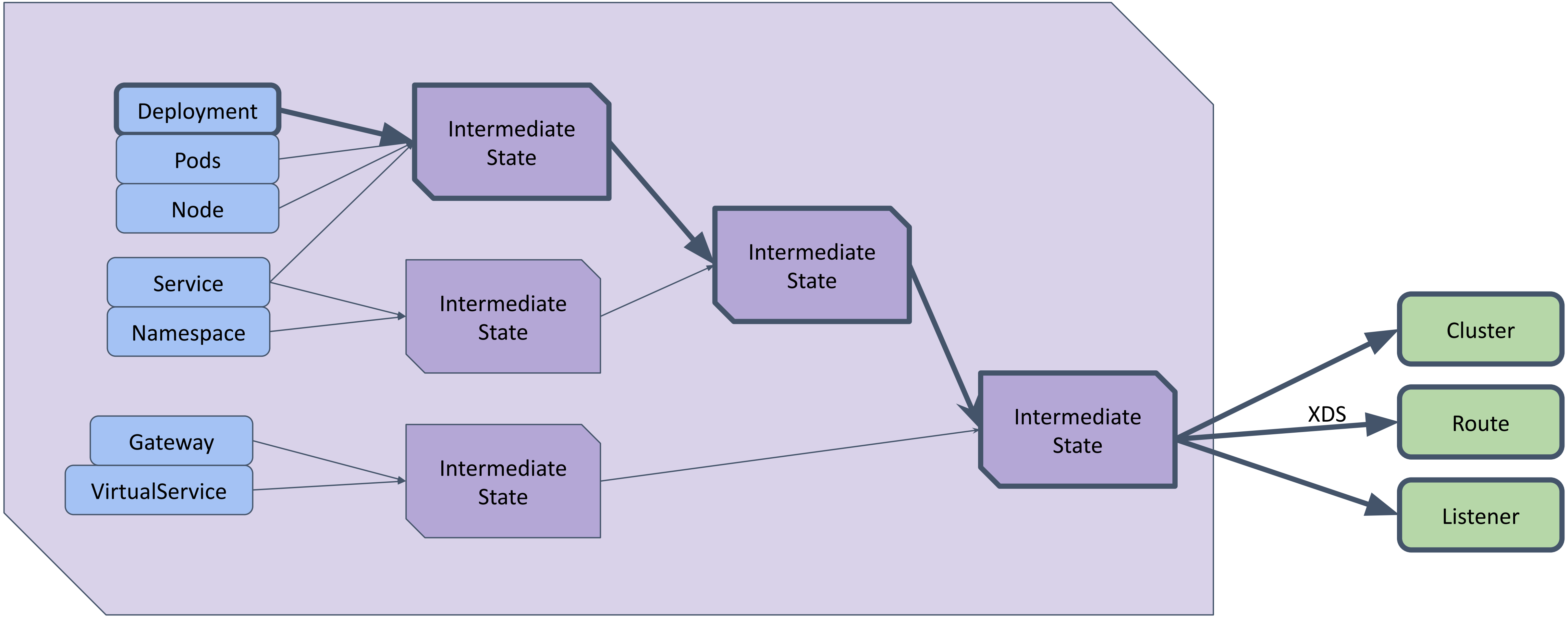
`Config` is automatically updated when the input `ConfigMaps` change

`Config` watchers notified any time `Config` changes

Collection Creation



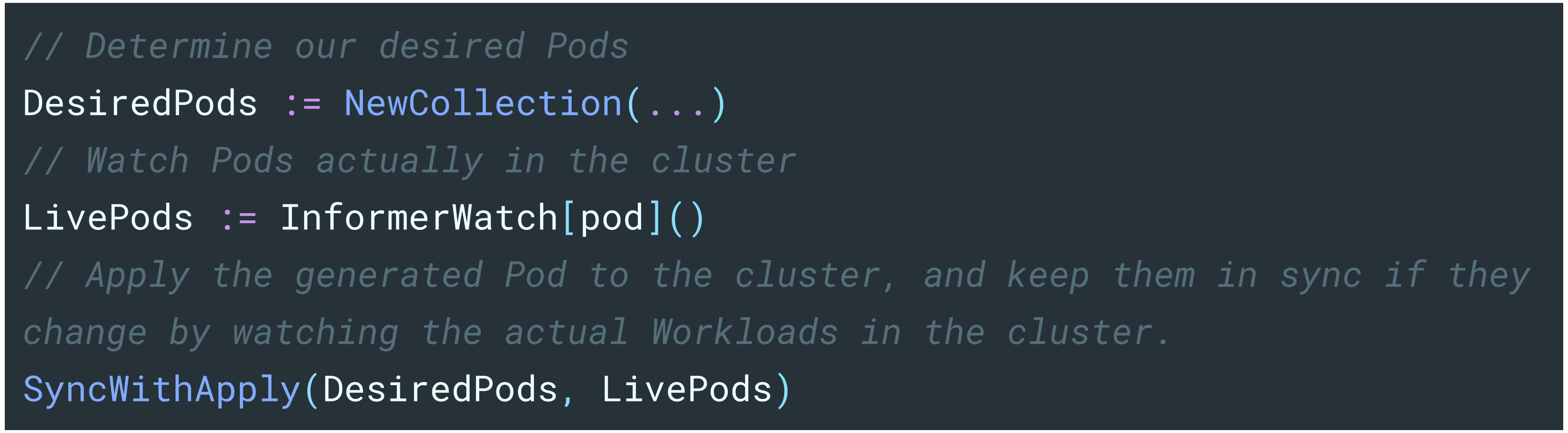
Event detection



Index Creation

```go
IpIndex := CreateIndex[Pod, string](Pods, func(p Pod) []string {
    return pod.Spec.PodIPs
})
// Now we can do IpIndex.Lookup(podIP)
```

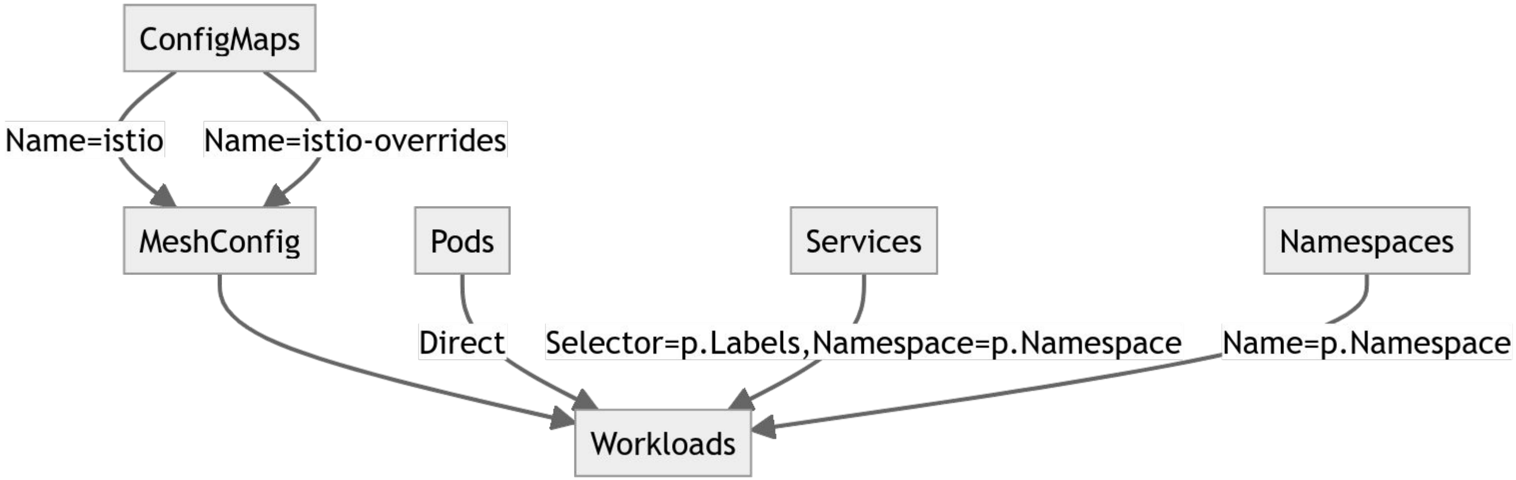
Outputs: Kubernetes



```
// Determine our desired Pods
DesiredPods := NewCollection(...)
// Watch Pods actually in the cluster
LivePods := InformerWatch[pod]()
// Apply the generated Pod to the cluster, and keep them in sync if they
change by watching the actual Workloads in the cluster.
SyncWithApply(DesiredPods, LivePods)
```

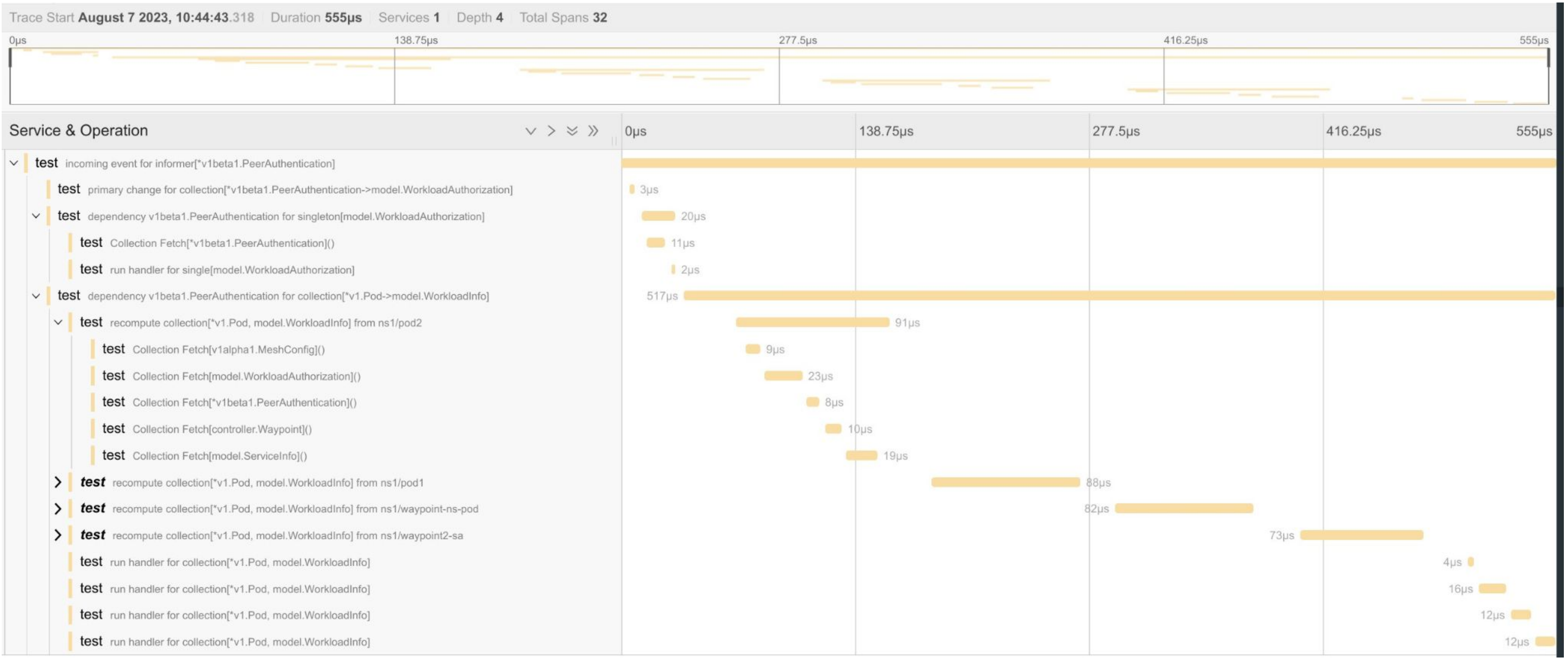
Debugging

Automatically generated architectural diagrams



Debugging

Auto-tracing instrumentation



Trace Start **August 7 2023, 10:44:43**.318 Duration **555μs** Services **1** Depth **4** Total Spans **32**

Service & Operation	0μs 138.75μs 277.5μs 416.25μs 555μs
test incoming event for informer[*v1beta1.PeerAuthentication]	
test primary change for collection[*v1beta1.PeerAuthentication->model.WorkloadAuthorization]	3μs
test dependency v1beta1.PeerAuthentication for singleton[model.WorkloadAuthorization]	20μs
test Collection Fetch[*v1beta1.PeerAuthentication]()	11μs
test run handler for single[model.WorkloadAuthorization]	2μs
test dependency v1beta1.PeerAuthentication for collection[*v1.Pod->model.WorkloadInfo]	517μs
test recompute collection[*v1.Pod, model.WorkloadInfo] from ns1/pod2	91μs
test Collection Fetch[v1alpha1.MeshConfig]()	9μs
test Collection Fetch[model.WorkloadAuthorization]()	23μs
test Collection Fetch[*v1beta1.PeerAuthentication]()	8μs
test Collection Fetch[controller.Waypoint]()	10μs
test Collection Fetch[model.ServiceInfo]()	19μs
test recompute collection[*v1.Pod, model.WorkloadInfo] from ns1/pod1	88μs
test recompute collection[*v1.Pod, model.WorkloadInfo] from ns1/waypoint-ns-pod	82μs
test recompute collection[*v1.Pod, model.WorkloadInfo] from ns1/waypoint2-sa	73μs
test run handler for collection[*v1.Pod, model.WorkloadInfo]	4μs
test run handler for collection[*v1.Pod, model.WorkloadInfo]	16μs
test run handler for collection[*v1.Pod, model.WorkloadInfo]	12μs
test run handler for collection[*v1.Pod, model.WorkloadInfo]	12μs

Testing

Lots of opportunities to improve testing

- Testing pure functions is much easier than controllers
- Unified framework gives more possibilities
  - Automated fuzz testing?

Where are we?

Demoware!

- Everything here is actually implemented as a prototype!
- I have a branch replacing ~50% of Istio with this model
- … but deploying to production is another matter

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

Questions?

Catch me at "Service Mesh Battle Scars"
In W176 at 5:25pm