KubeCon | CloudNativeCon

North America 2018

Life of a Kubernetes Watch Event

Haowei Cai (roycaihw@github), Google
Wenjia Zhang (wenjiaswe@github), Google

About us

Wenjia Zhang (@wenjiaswe)

Software Engineer in Google. She is an active contributor for Kubernetes SIG API Machinery and etcd open source projects.

Haowei Cai (@roycaihw)

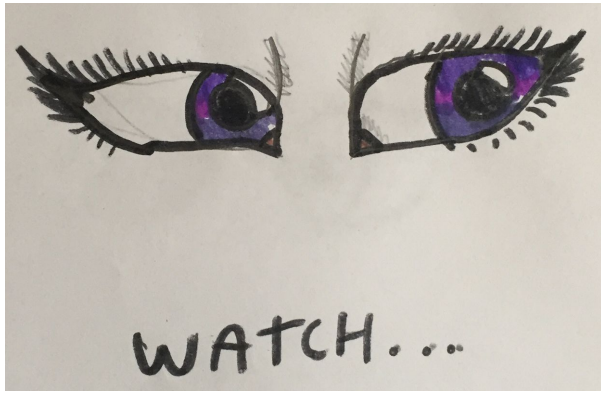
Software Engineer in Google. He is an active contributor for Kubernetes SIG API Machinery and client libraries.

Agenda

- **What** is a Kubernetes Watch Event?
- **Why** is Watch Event important for Kubernetes?
- **How** is the life of a Kubernetes Watch Event?
- **Key Takeaways**

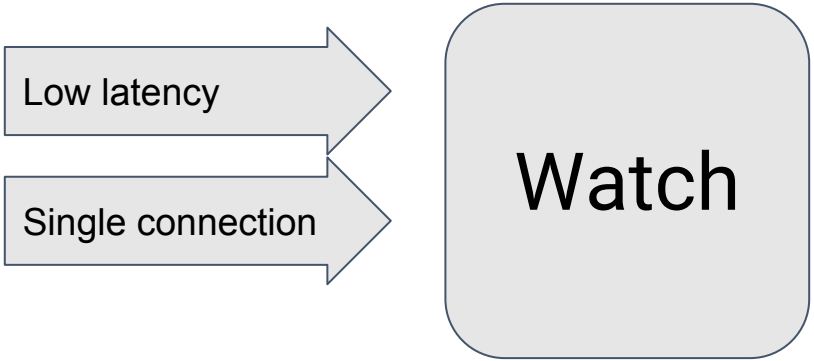
What is a Kubernetes Watch Event?

What is Watch?



Watch is an incremental change notification feed

Watch vs. Poll



Watch vs. Poll

Low latency

Single connection

Watch

Poll

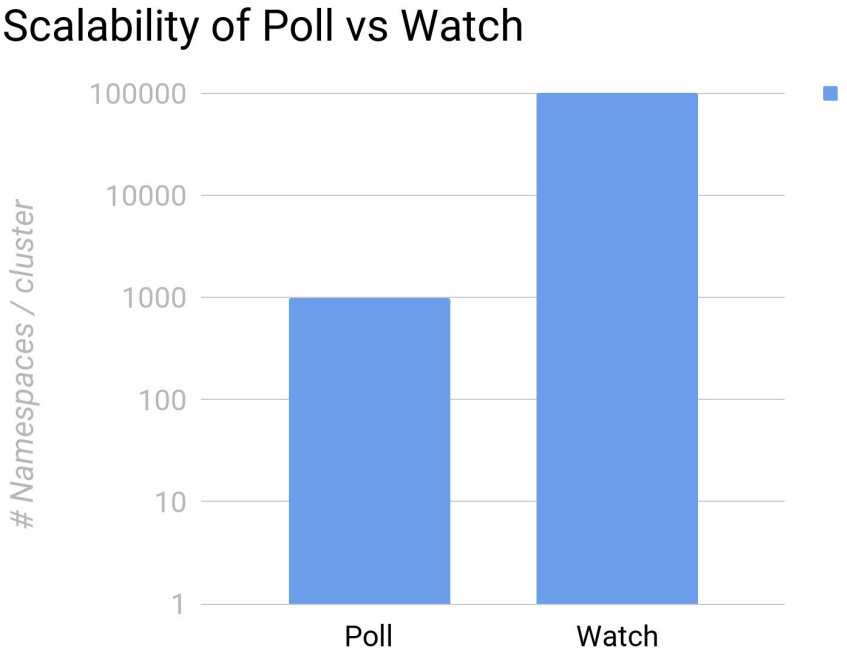
Extra load
Extra latency

Multiple connection

Watch vs. Poll

Kubelet on nodes:

- Previous: periodically poll kube-apiserver for secrets and configmaps
- Now: watch individual secrets
- OSS PR: [Kubelet watches necessary secrets/configmaps instead of periodic polling #64752]



What is Event?

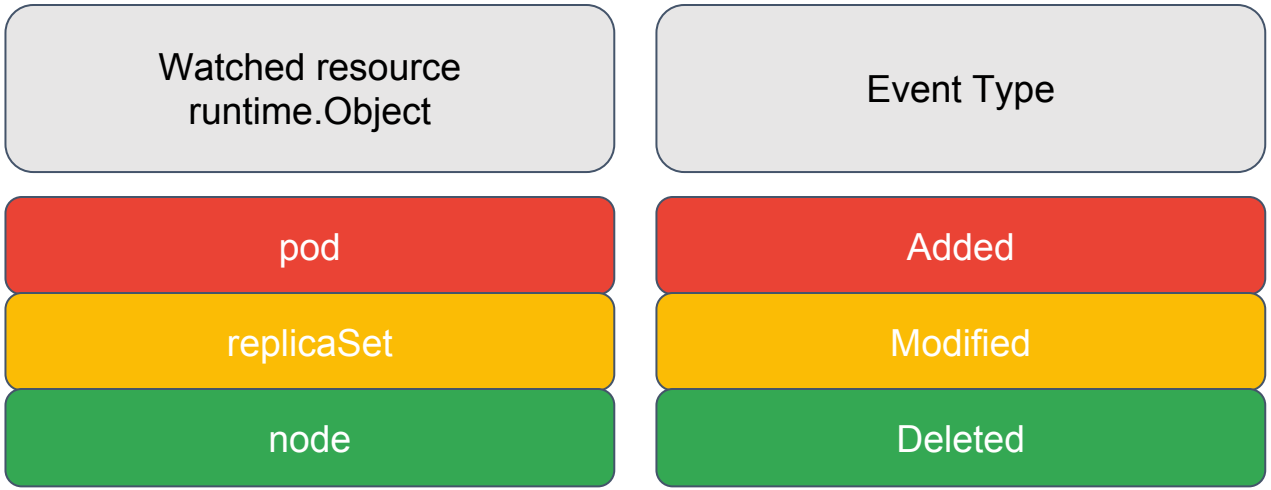
A single change to a watched resource

Watched resource
runtime.Object

Event Type

What is Event?

A single change to a watched resource



Why is Watch Event important for Kubernetes?

Kubernetes core design concept

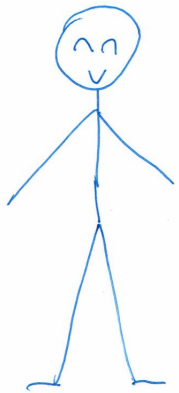
Level Triggering and Soft Reconciliation

Declarative configuration

Desired state

Declarative configuration

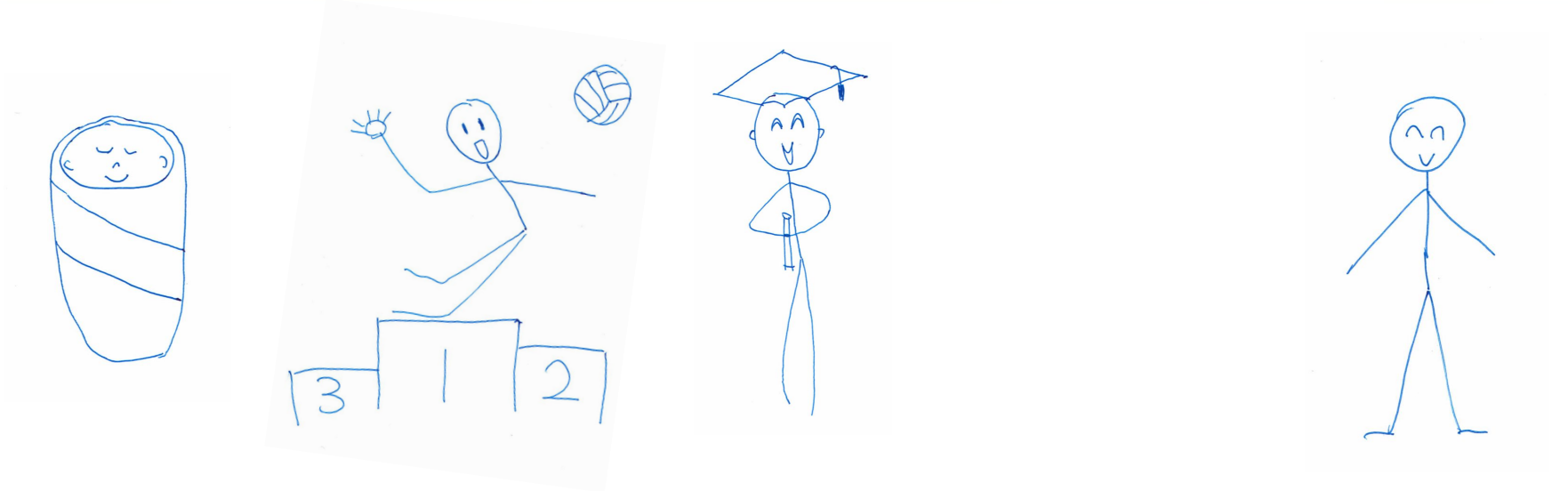
State is accumulation of events



Person1

State

State is accumulation of events



Person1, Added

Person1, Modified

Person1, Modified

. . .

Person1

Events

State

State is accumulation of events

Pod1

State

State is accumulation of events


```
apiVersion: v1
kind: Pod
metadata:
  name: constraintpod
spec:
  containers:
  - name: sise
    image: mhausenblas/simpleservice:0.5.0
    ports:
    - containerPort: 9876
    resources:
      limits:
        memory: "64Mi"
        cpu: "500m"
```

Adding label

Image version change:
0.5.0 -> 1.5.3

Pod1, Added

Pod1, Modified

Pod1, Modified

. . .

Pod1

Events

State

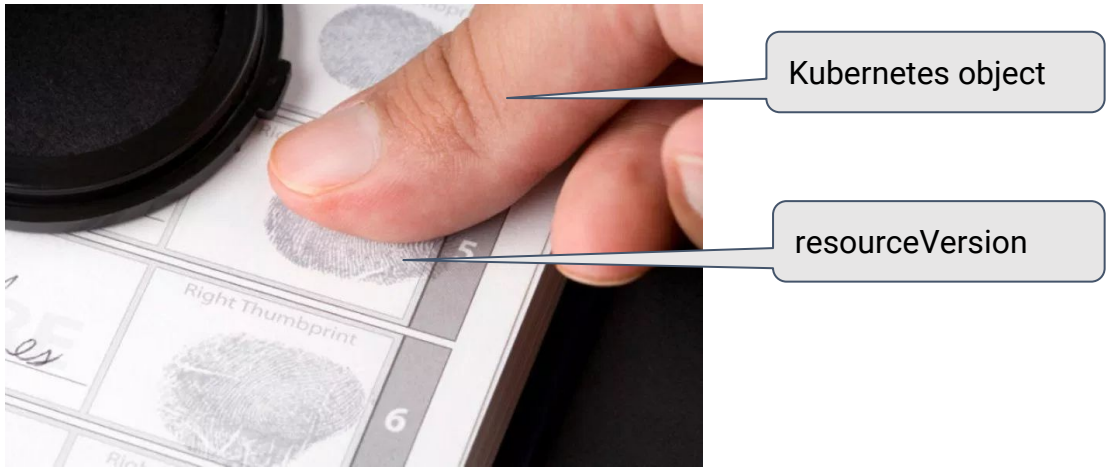
KubeCon | CloudNativeCon

North America 2018

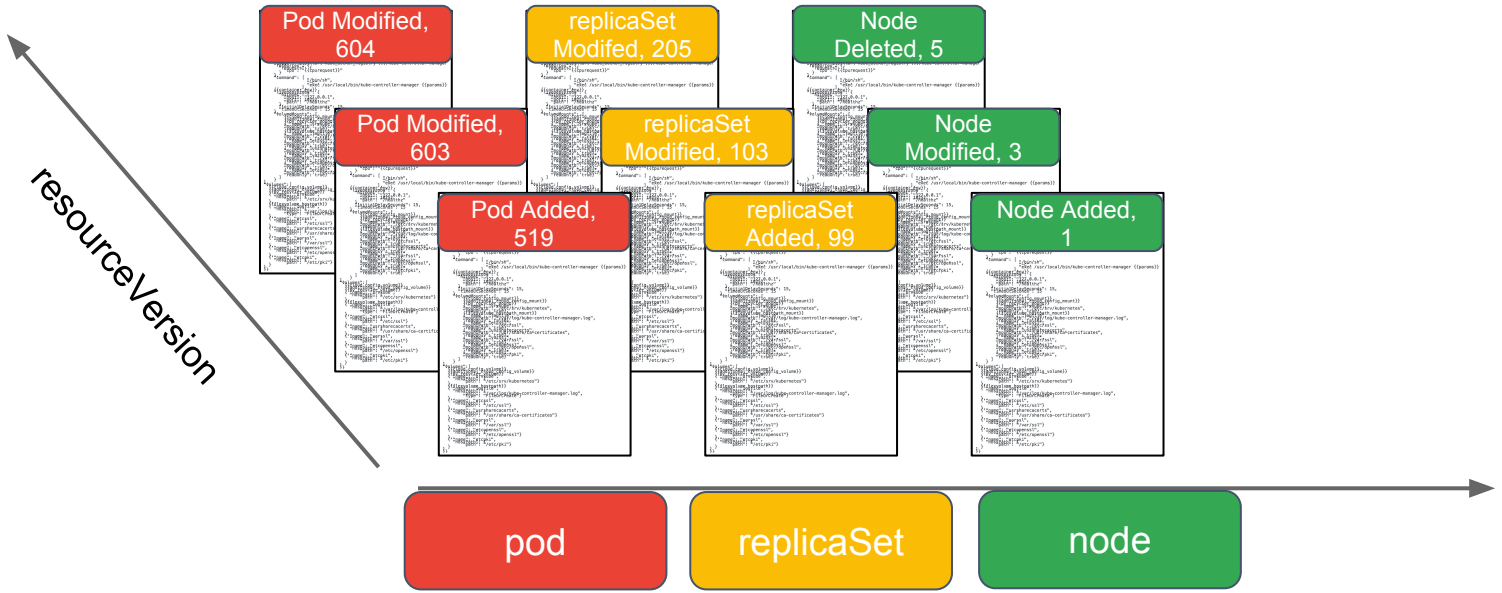
How is the life of a K8s watch event?

Fingerprint of kubernetes object: resourceVersion

A resourceVersion is valid on a single kind of resource across namespaces.



resourceVersion created with object change/event



resourceVersion created every time the resource is written


```
apiVersion: v1
kind: Pod
metadata:
  name: constraintpod
spec:
  containers:
  - name: sise
    image: mhausenblas/simpleservice:0.5.0
    ports:
    - containerPort: 9876
    resources:
      limits:
        memory: "64Mi"
        cpu: "500m"
```

Adding label

Image version change:
0.5.0 -> 1.5.3

Pod1, Added	Pod1, Modified	Pod1, Modified	. . .	Pod1
519	603	604		

resourceVersion changes every time the resource is written


```
apiVersion: v1
kind: Pod
metadata:
  name: constraintpod
spec:
  containers:
  - name: sise
    image: mhausenblas/simpleservice:0.5.0
    ports:
    - containerPort: 9876
    resources:
      limits:
        memory: "64Mi"
        cpu: "500m"
```

Adding label

Image version change:
0.5.0 -> 1.5.3

Pod1, Added	Pod1, Modified	Pod1, Modified	. . .	Pod1
519	603	604		

Life of a K8s watch event

Pod1, Added
resourceVersion: 519

Life of a K8s watch event

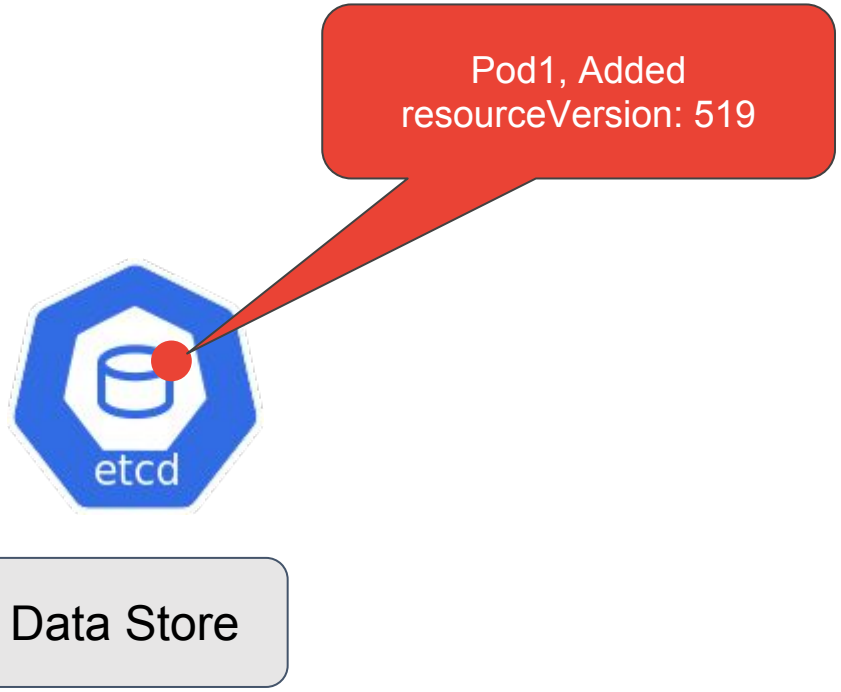
Pod1, Added
resourceVersion: 519

Schedule pods on nodes

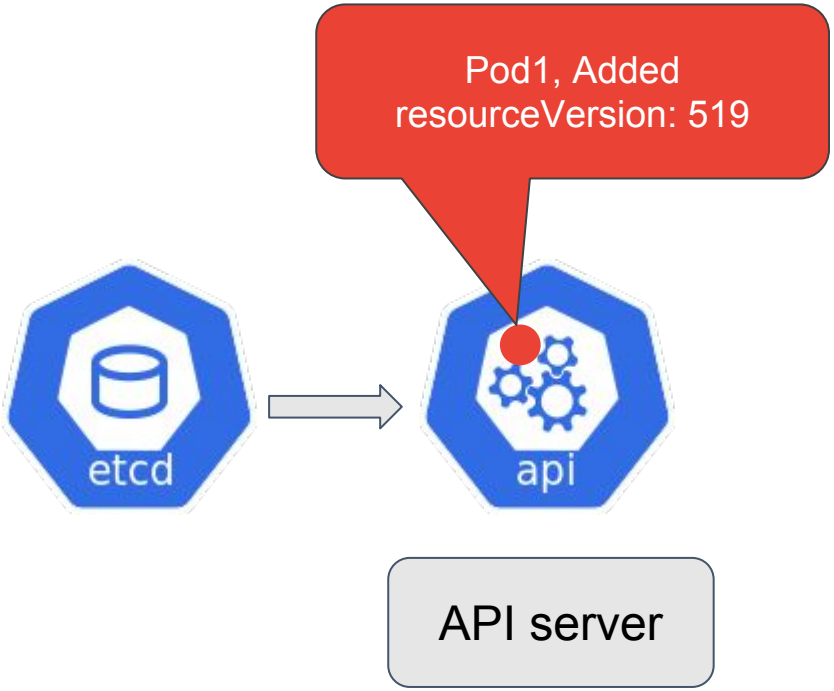
Runs controllers

Business logics

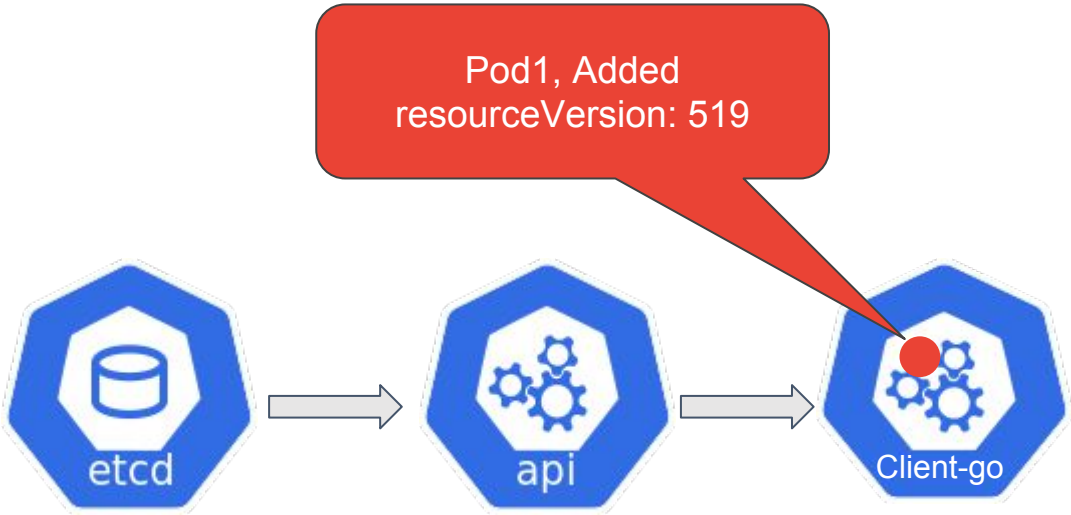
Life of a K8s watch event



Life of a K8s watch event

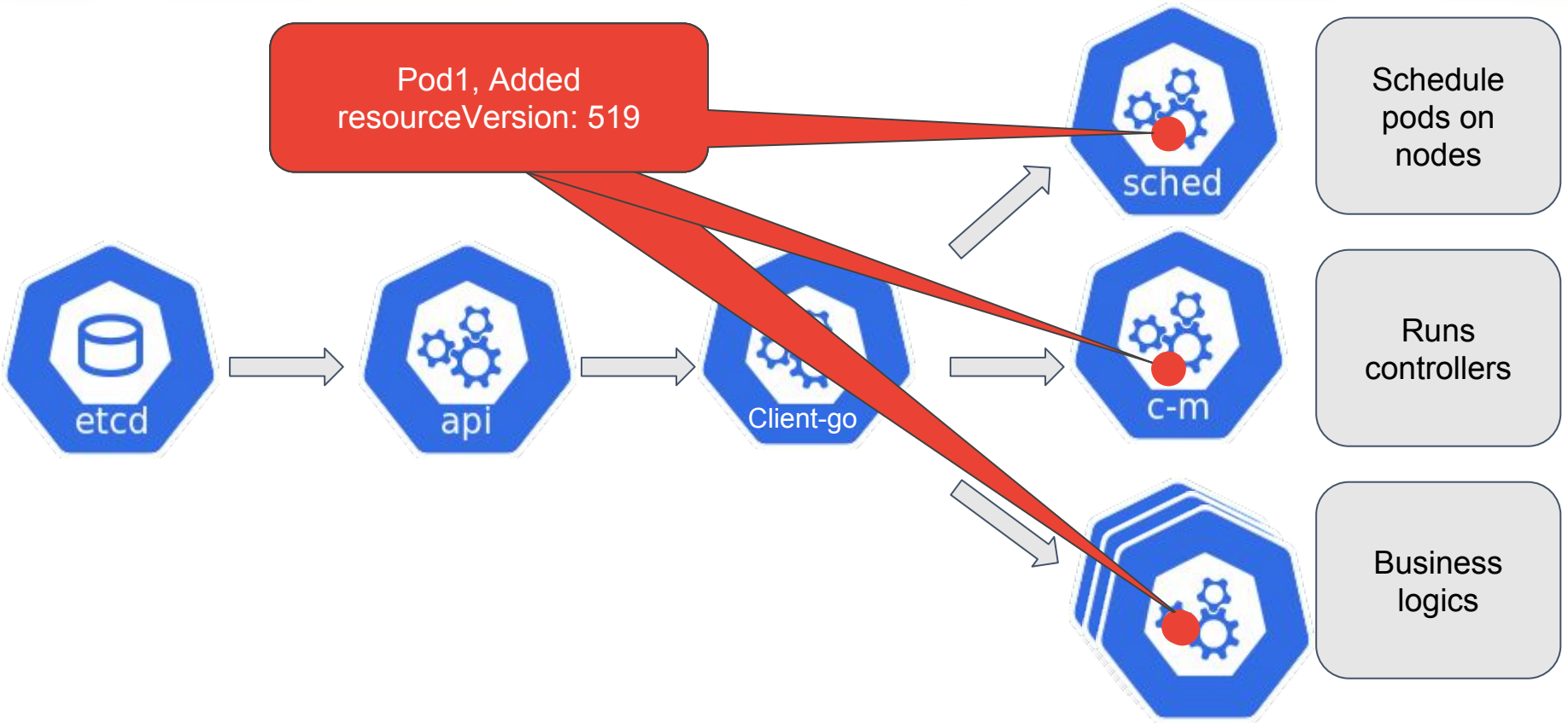


Life of a K8s watch event



Go clients for talking to a kubernetes cluster.

Life of a K8s watch event



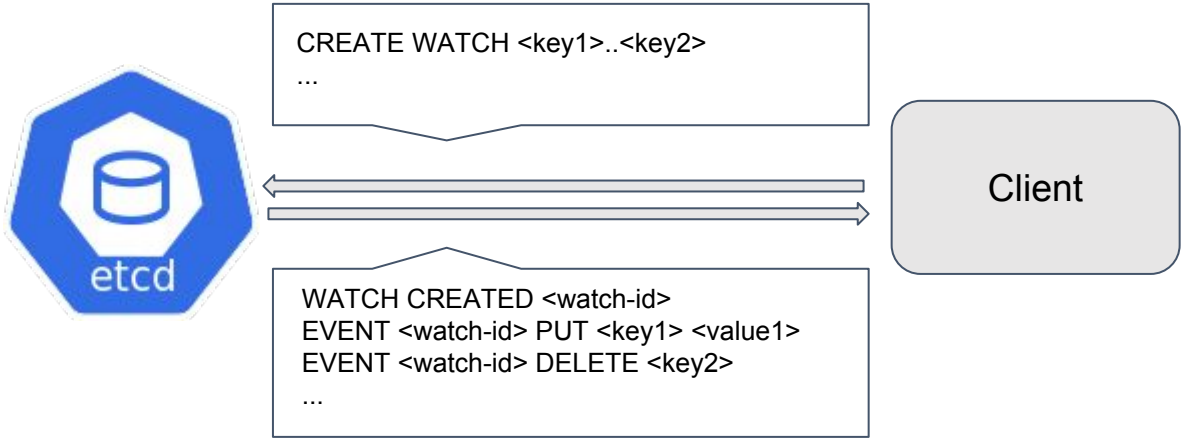
Watch Event in etcd

Watch in etcd

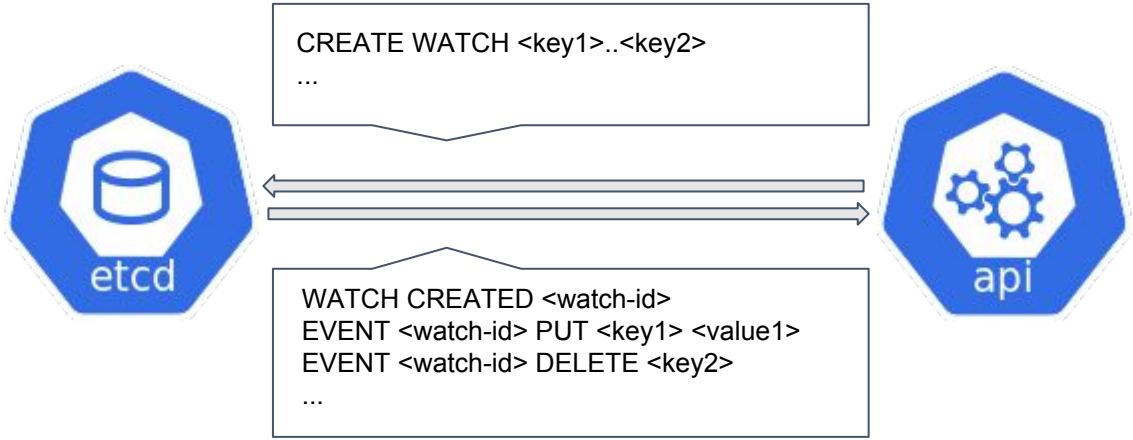
etcd **watch** feature provides an event-based interface for asynchronously monitoring changes to keys.

Revision (etcd) == resourceVersion (apiserver)

Watch event in etcd

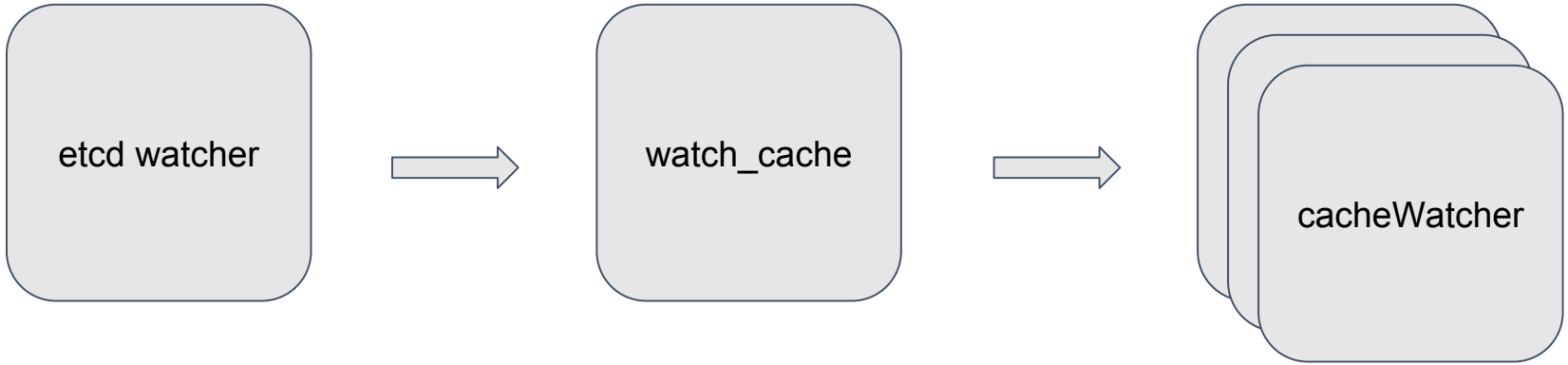


Watch event in etcd

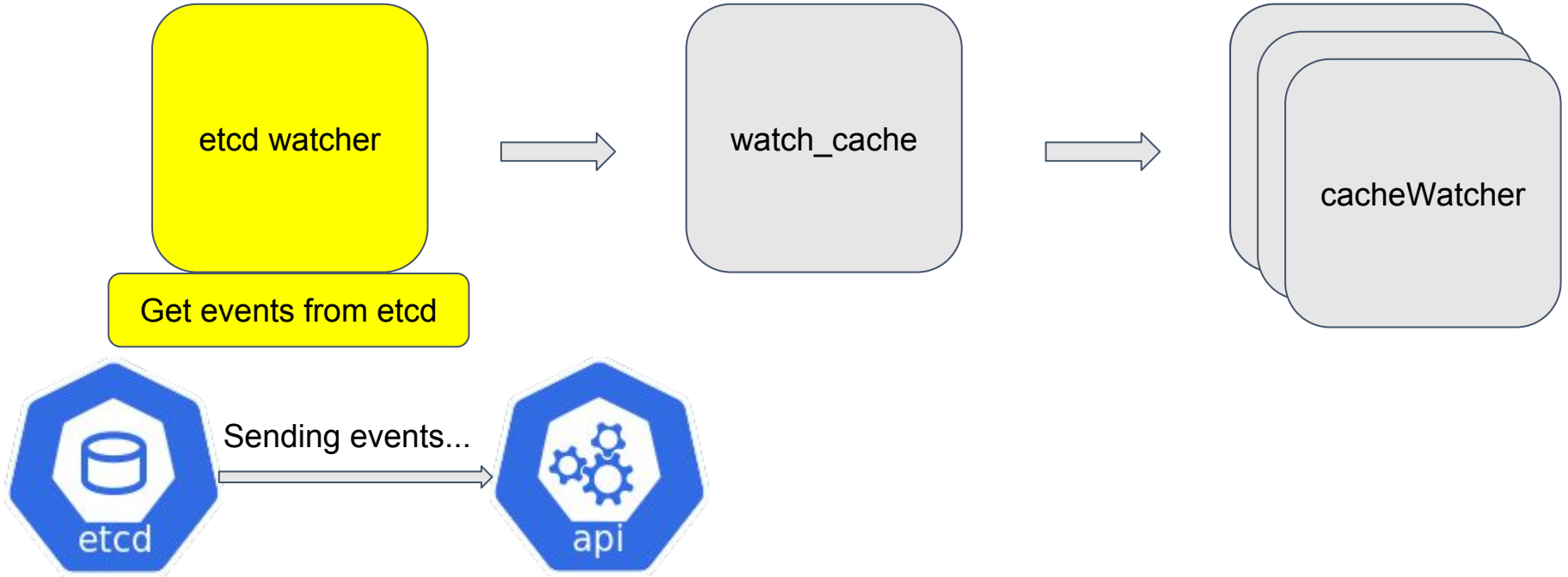


Watch Event in Kube APIserver

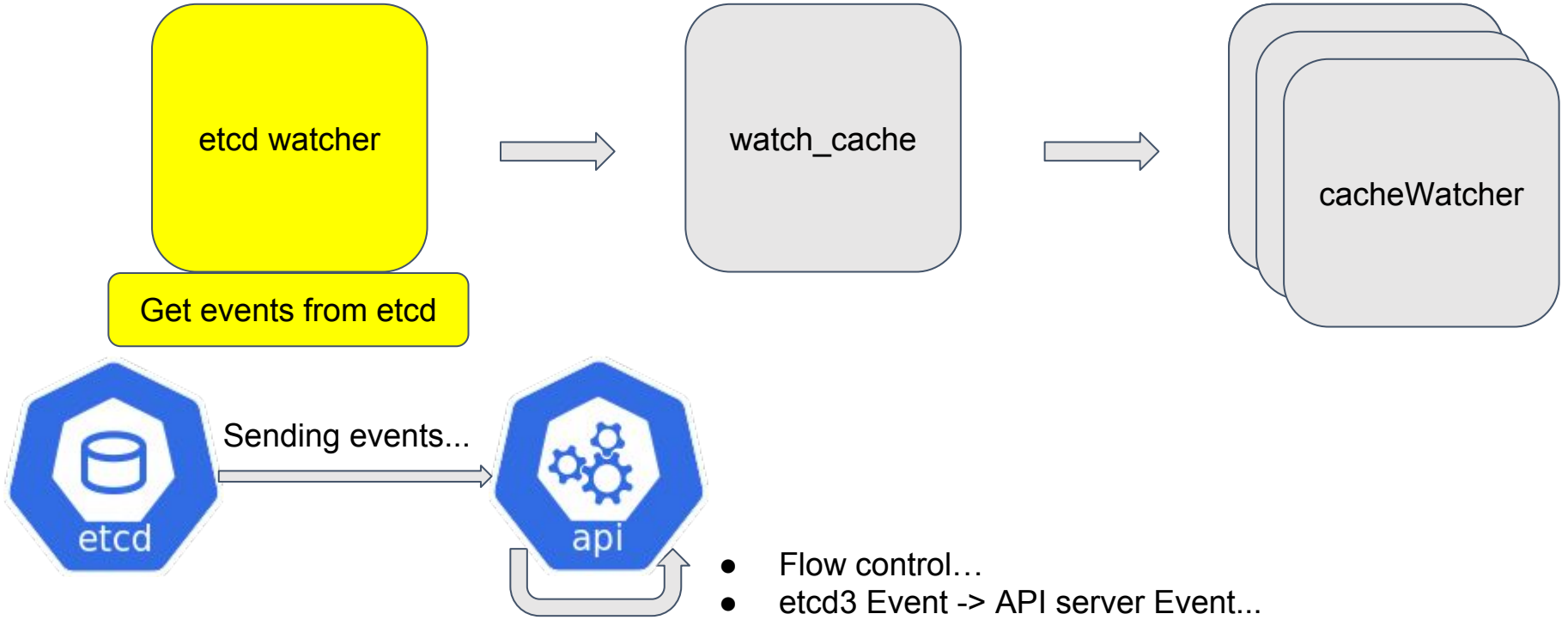
Watch Event in Kube APIserver



Watch Event in Kube APIserver

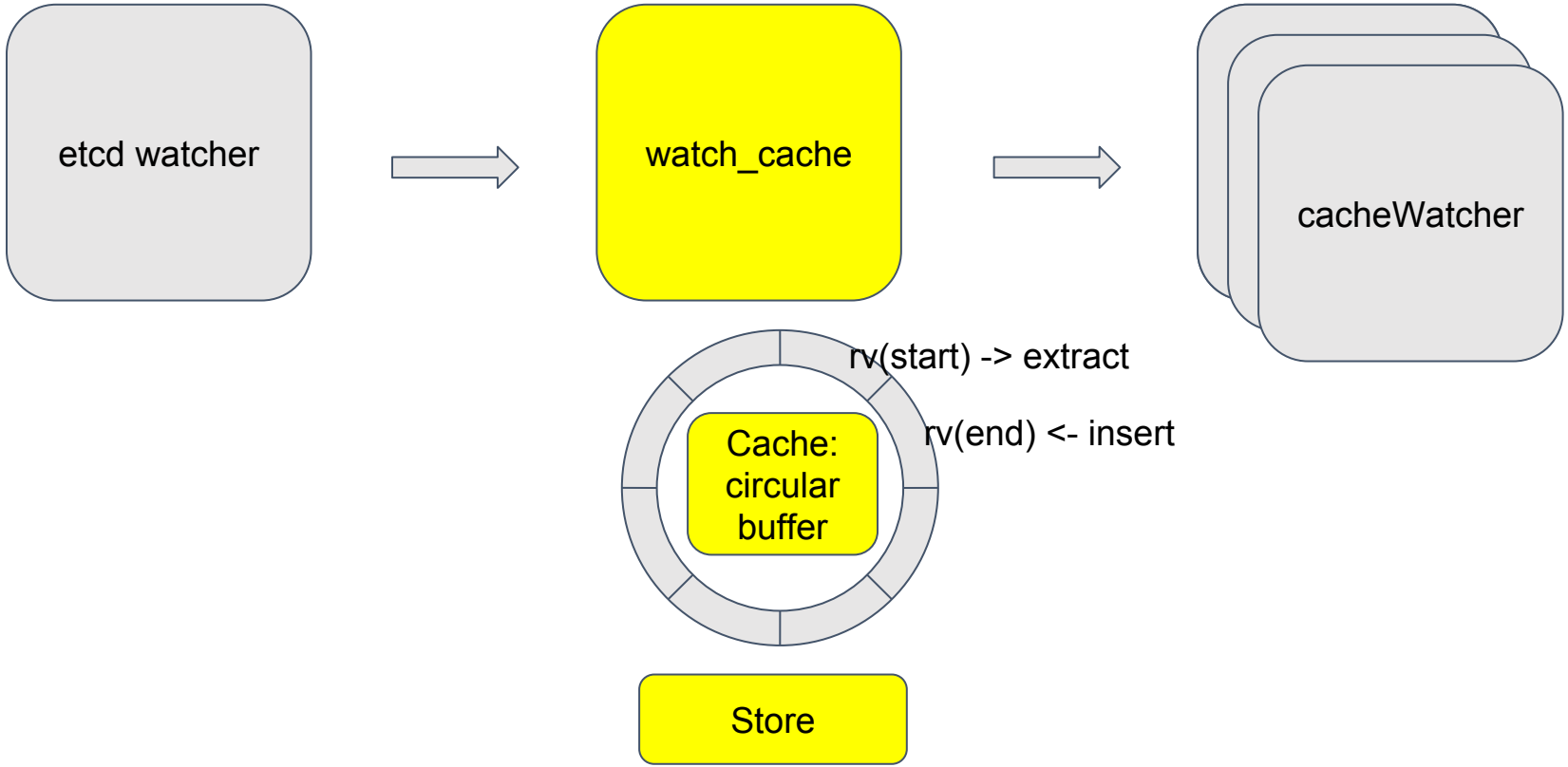


Watch Event in Kube APIserver

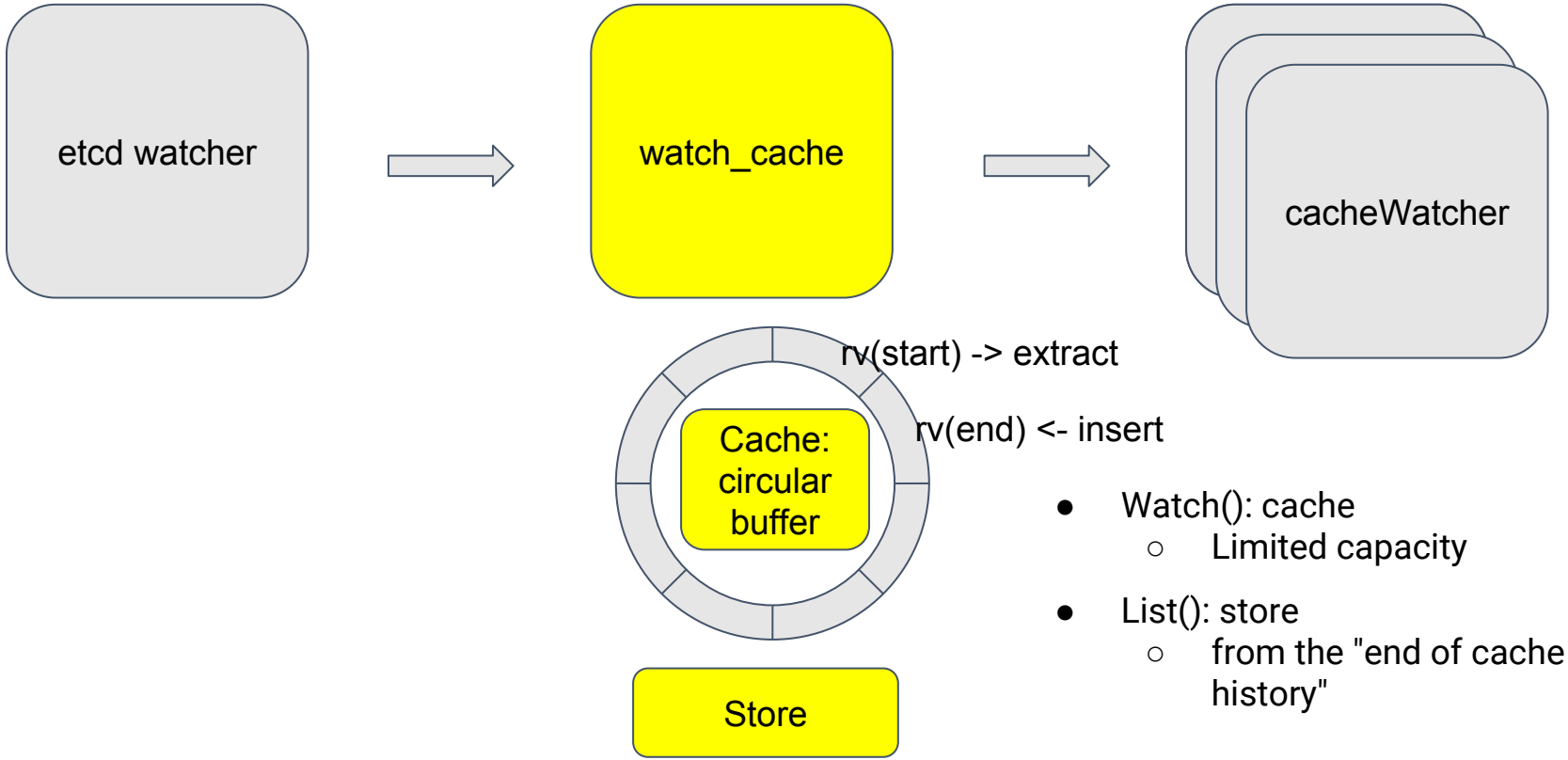


- Flow control…
- etcd3 Event -> API server Event...

Watch Event in Kube APIserver

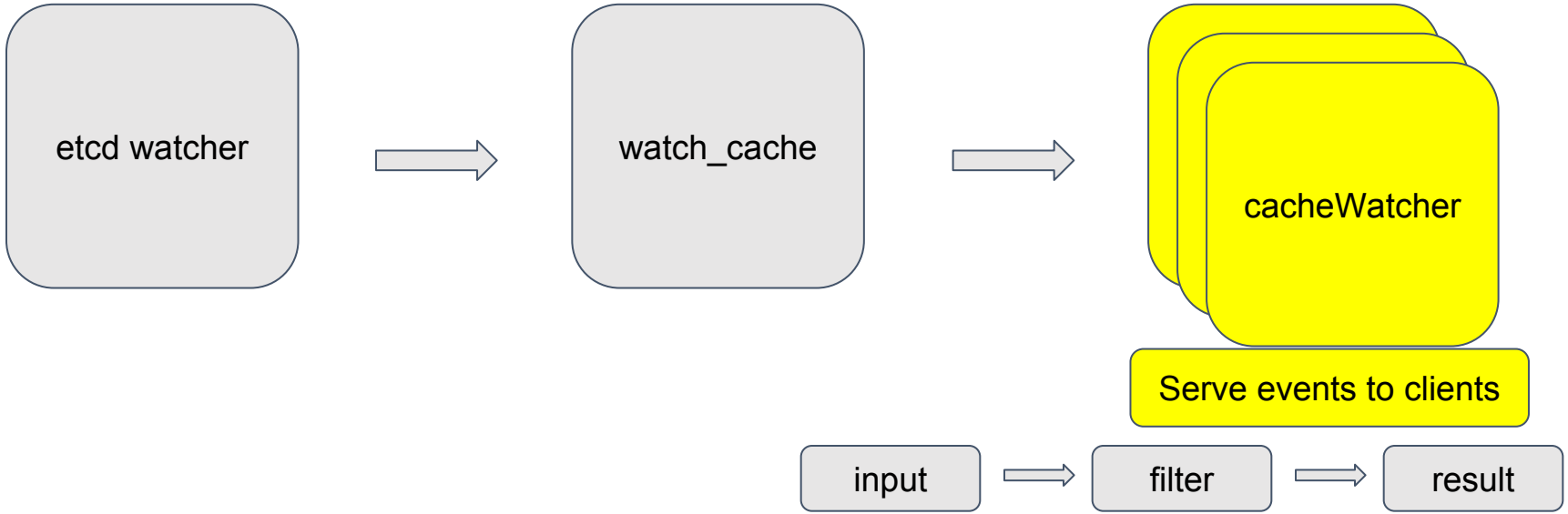


Watch Event in Kube APIserver



- Watch(): cache
  - Limited capacity
- List(): store
  - from the "end of cache history"

Watch Event in Kube APIserver



Watch Event in Client-go

What is Client-go?

Clientset
Dynamic Client
REST Client
Informer
...

https://github.com/kubernetes/client-go

- Go clients for talking to a kubernetes cluster
- Used by Kubernetes itself

What is Client-go?

Clientset
Dynamic Client
REST Client
Informer
...

https://github.com/kubernetes/client-go

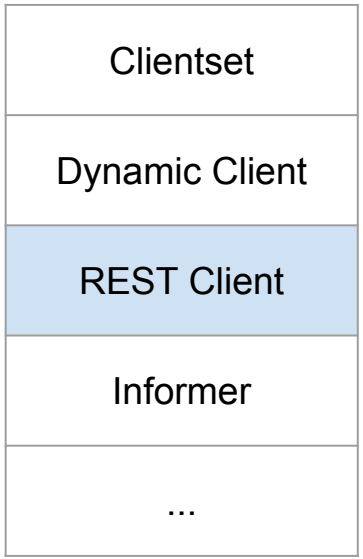
- Go clients for talking to a kubernetes cluster
- Used by Kubernetes itself

What is Client-go?

https://github.com/kubernetes/client-go

- Go clients for talking to a kubernetes cluster
- Used by Kubernetes itself

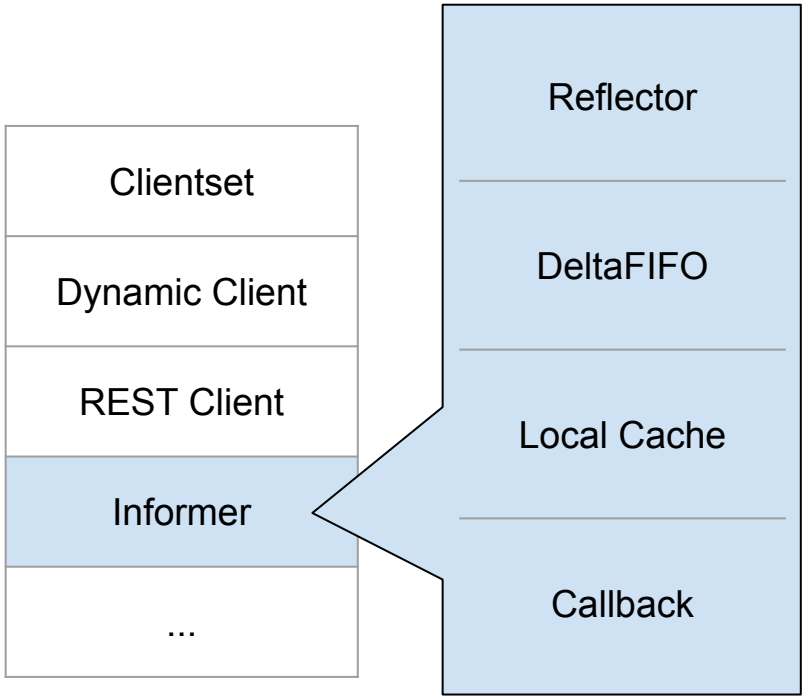
What is Client-go?



https://github.com/kubernetes/client-go

- Go clients for talking to a kubernetes cluster
- Used by Kubernetes itself

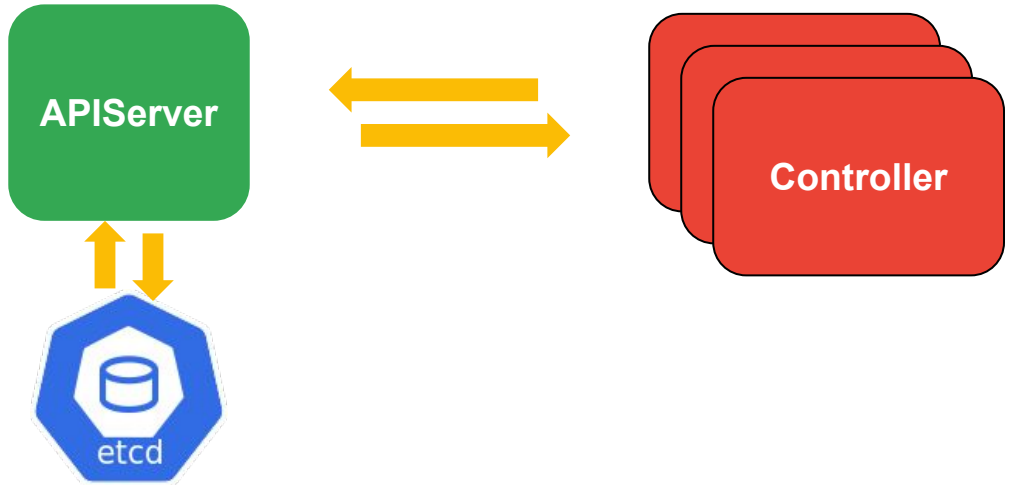
What is Informer?



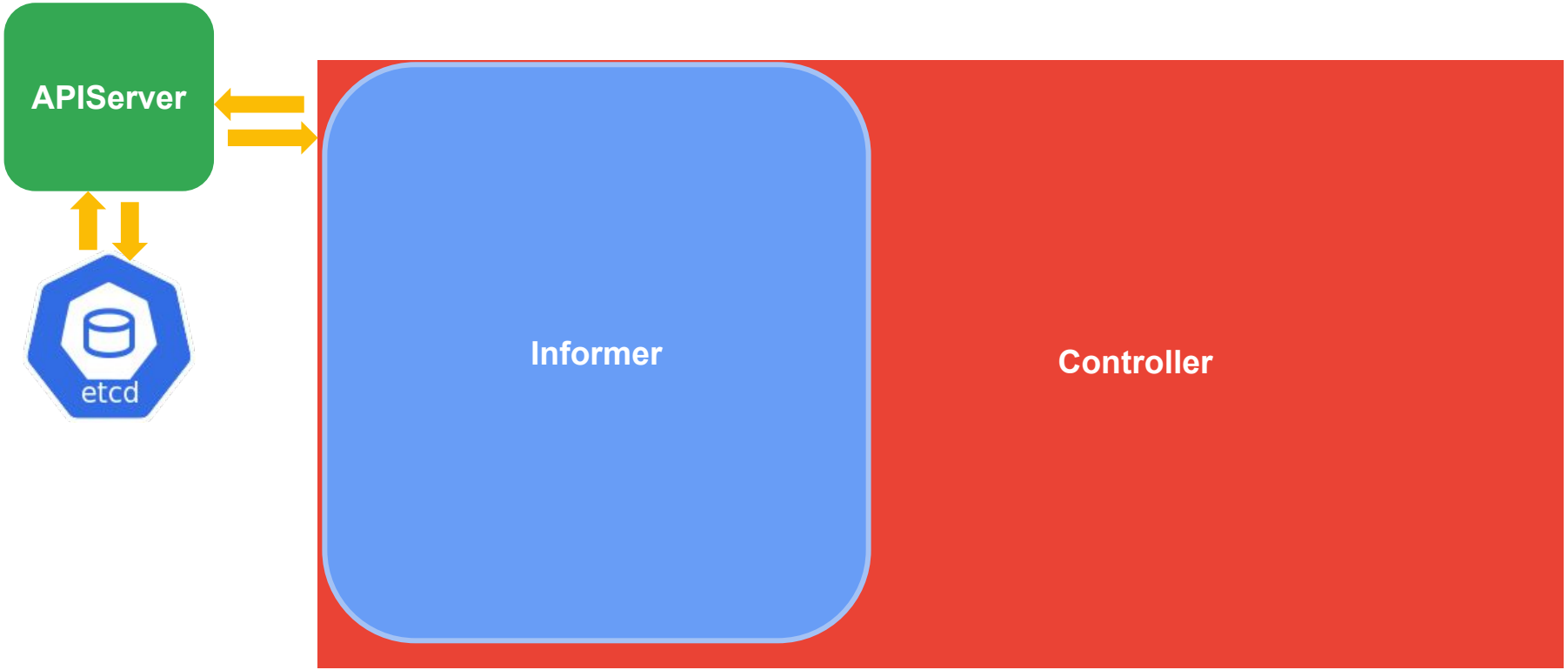
k8s.io/client-go/tools/cache
k8s.io/client-go/informers

- Useful component for building event-oriented controllers
- Used by control plane controllers, kubelet, etc.
- Reflector used by kube-apiserver watch cache

Kubernetes controller workflow

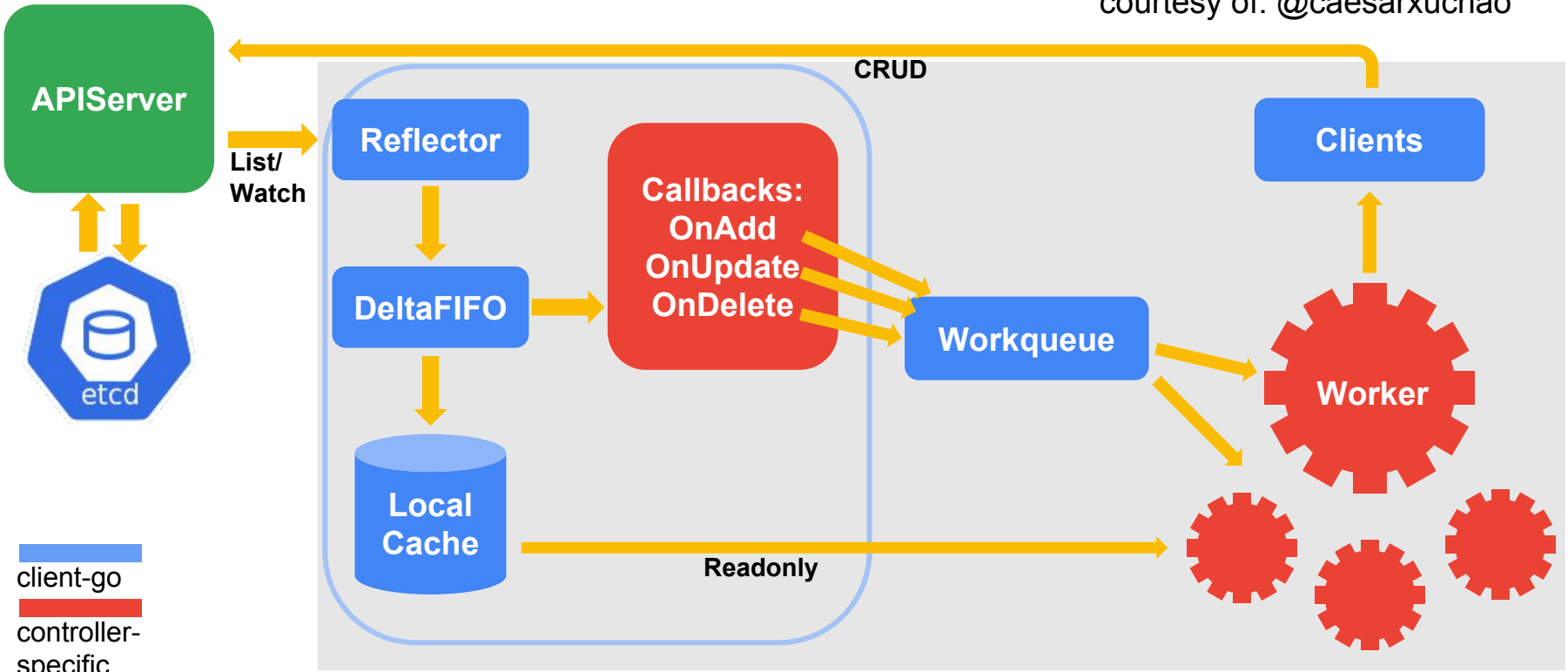


Kubernetes controller workflow

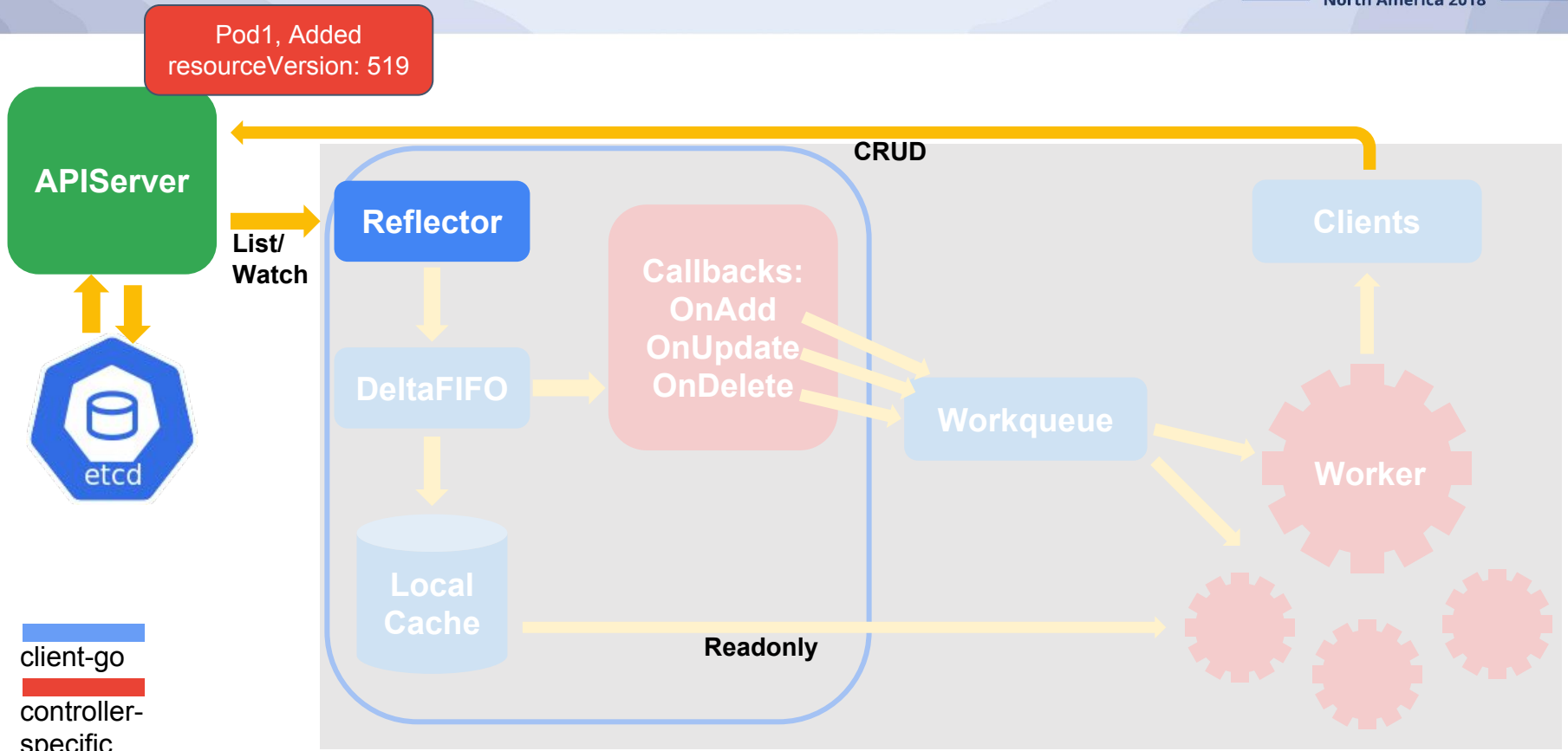


Kubernetes controller workflow

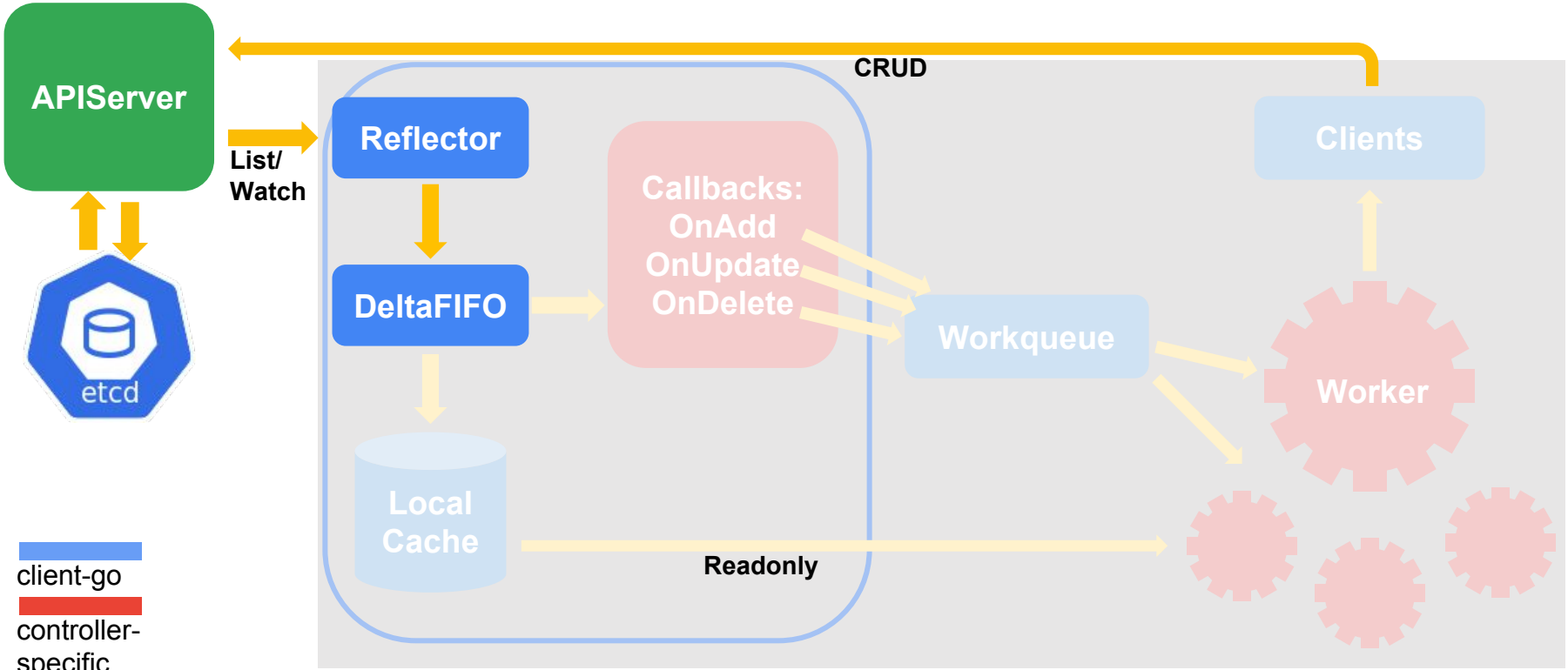
courtesy of: @caesarxuchao



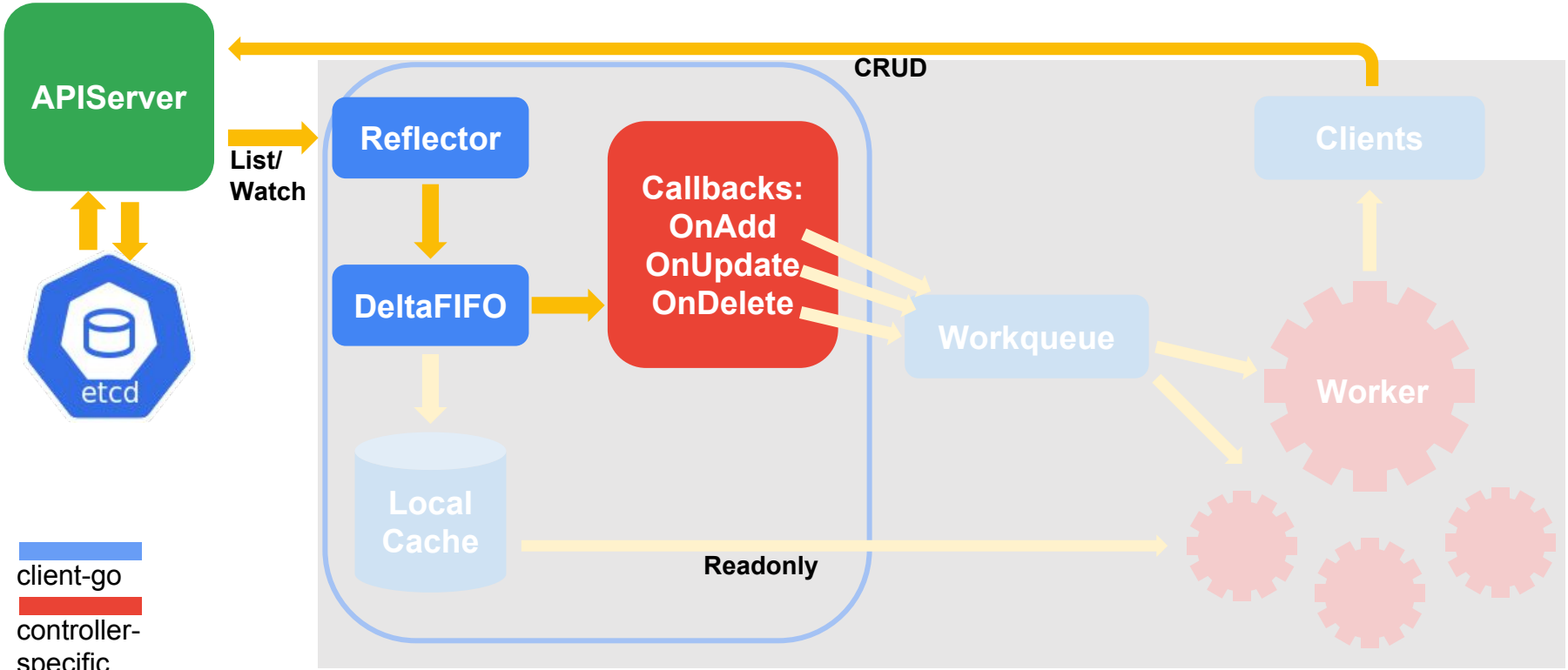
Kubernetes controller workflow



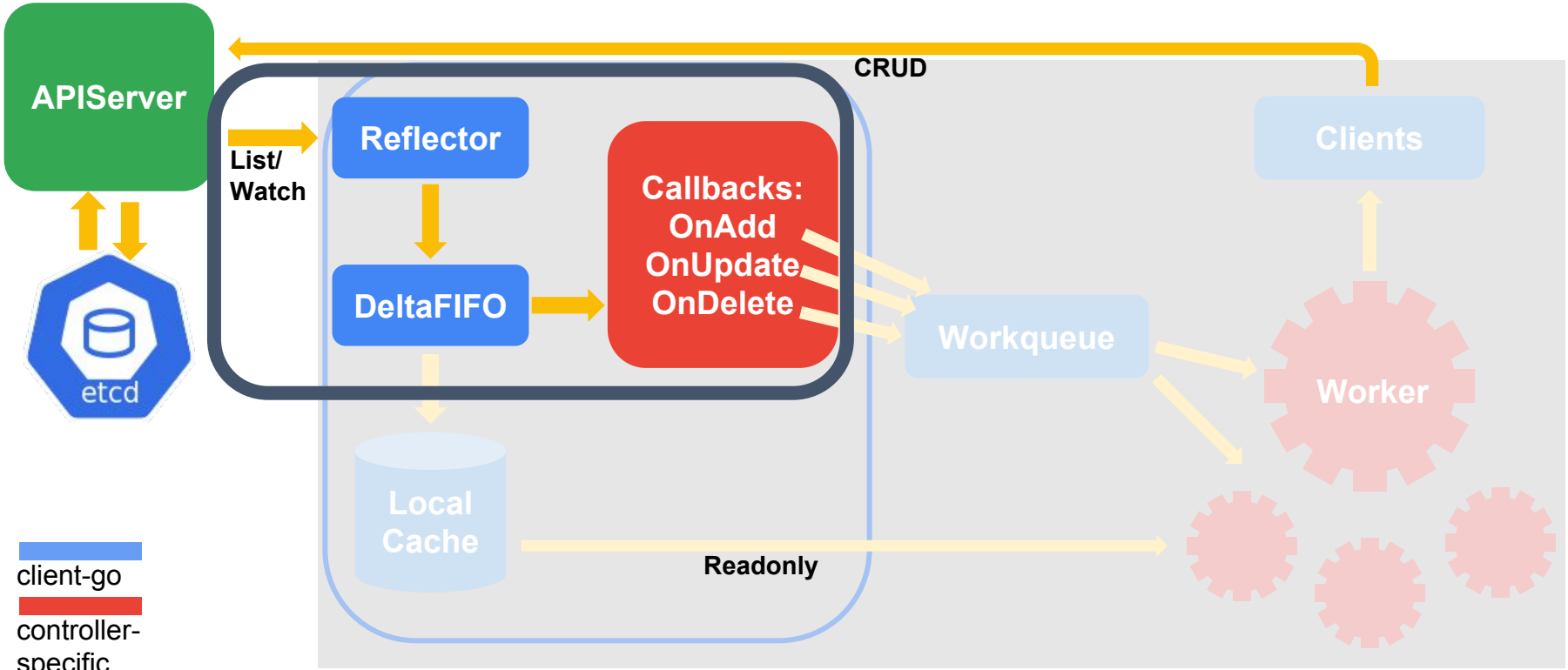
Kubernetes controller workflow



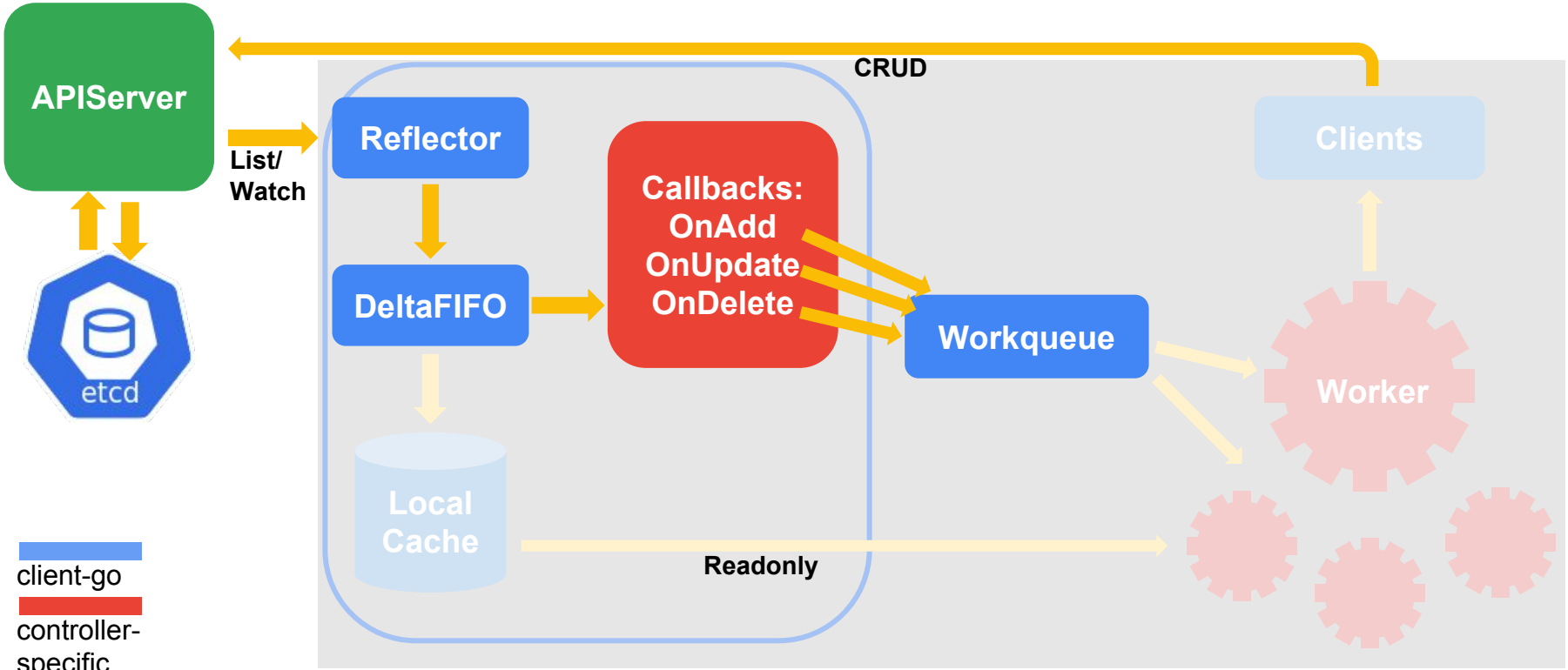
Kubernetes controller workflow



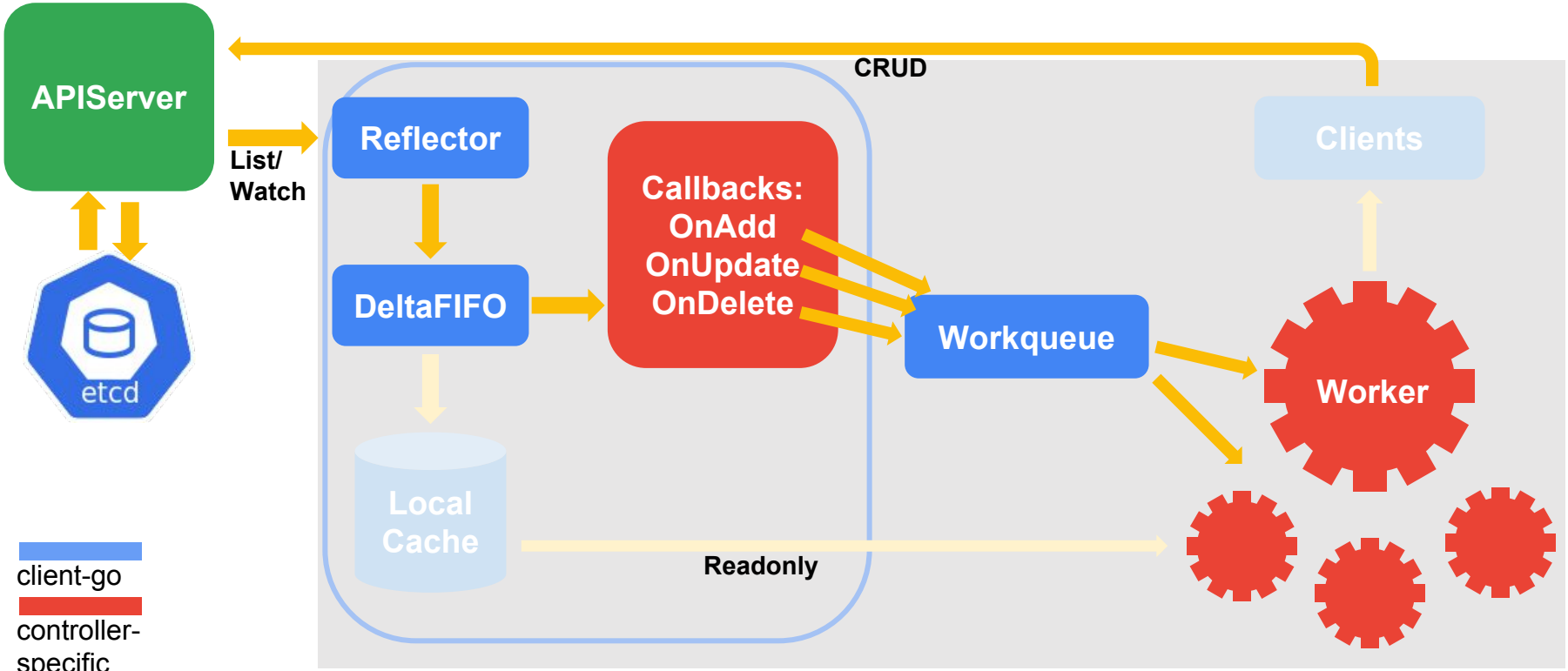
Kubernetes controller workflow



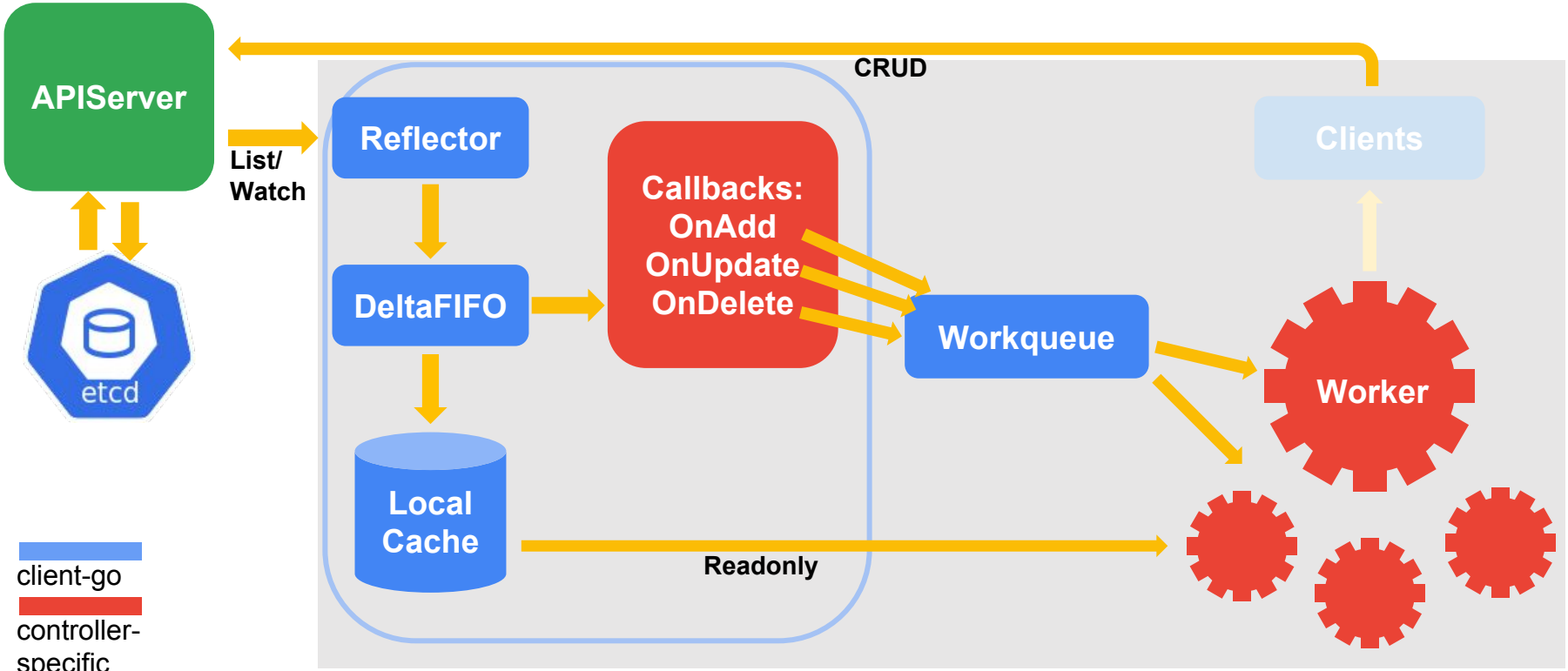
Kubernetes controller workflow



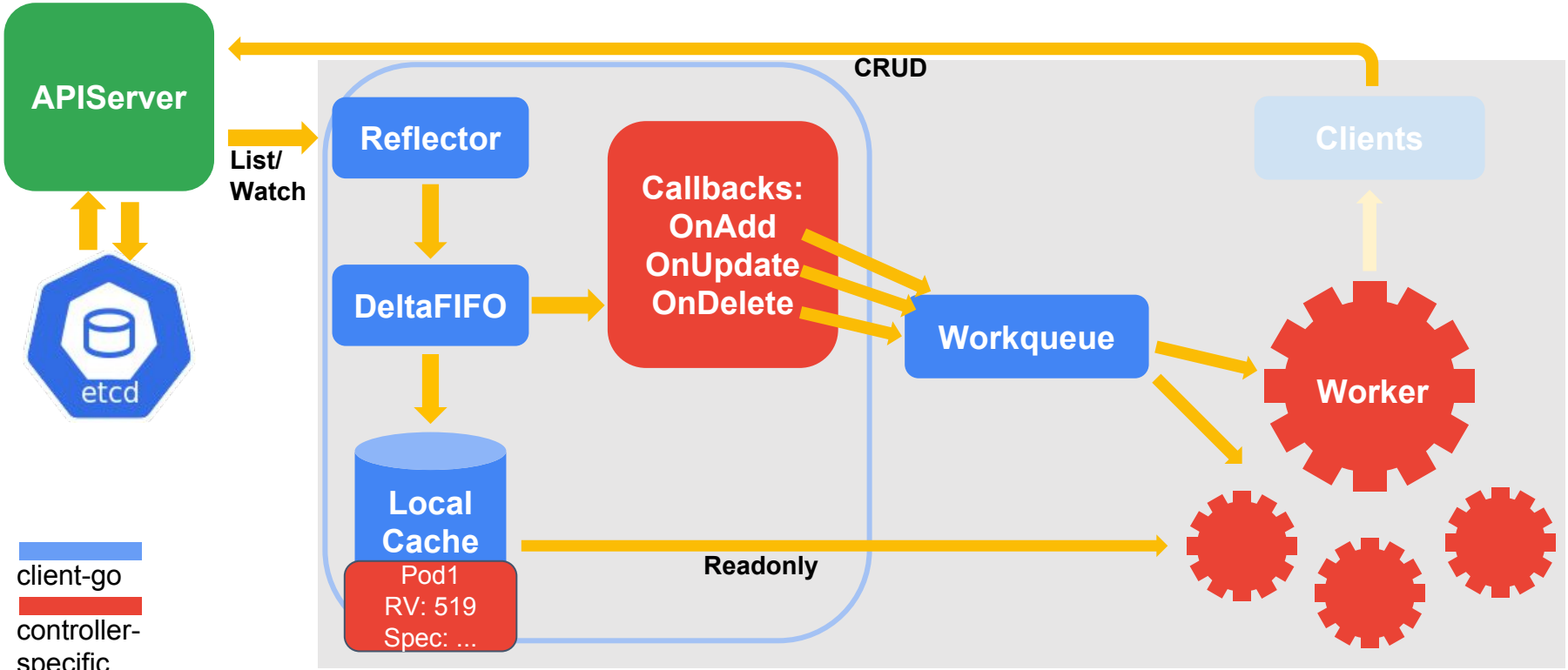
Kubernetes controller workflow



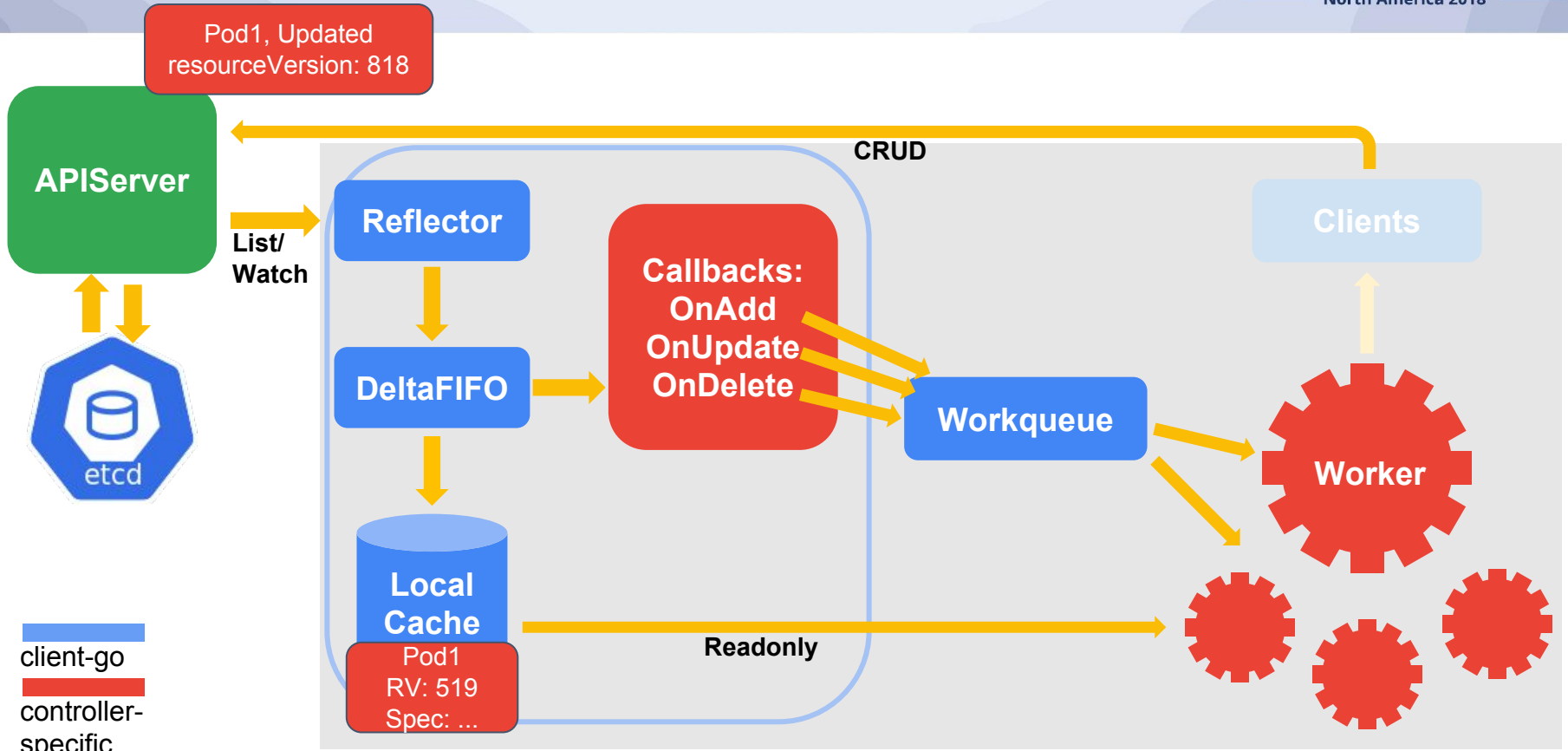
Kubernetes controller workflow



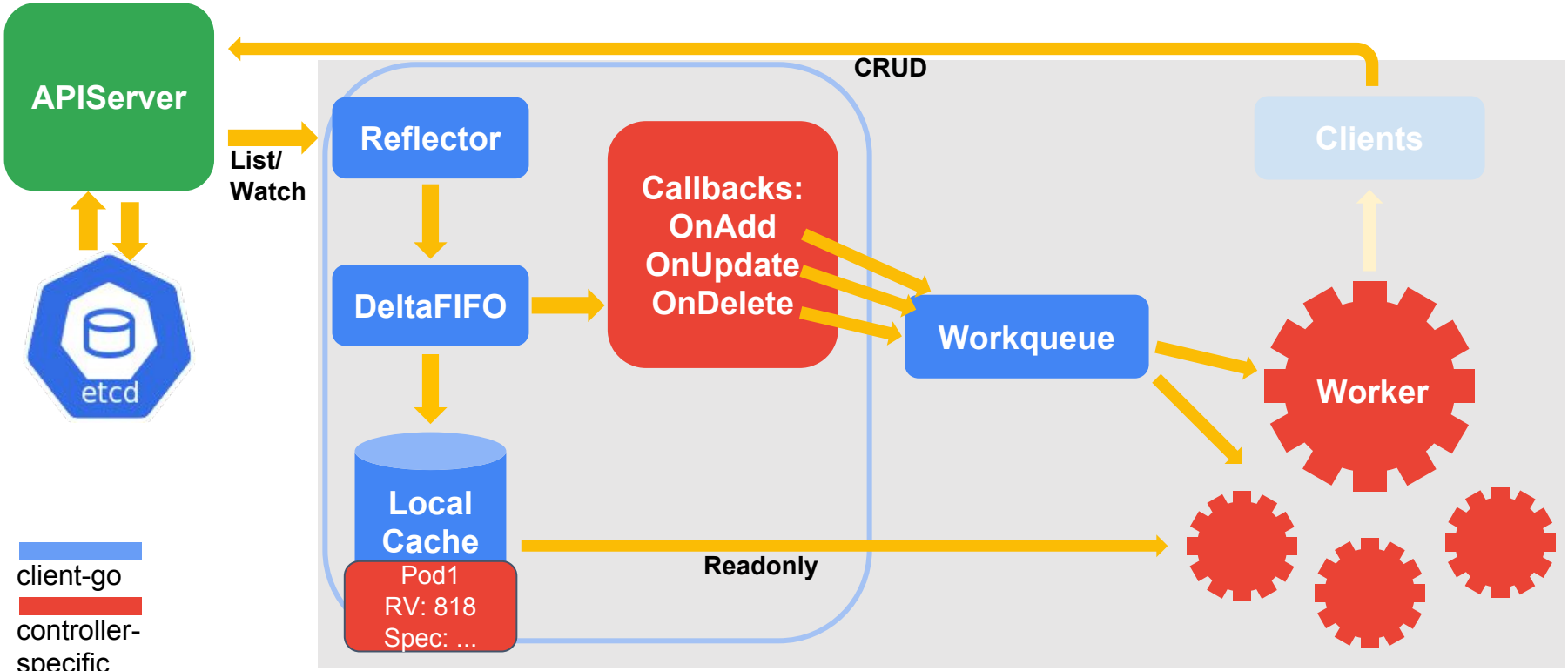
Kubernetes controller workflow



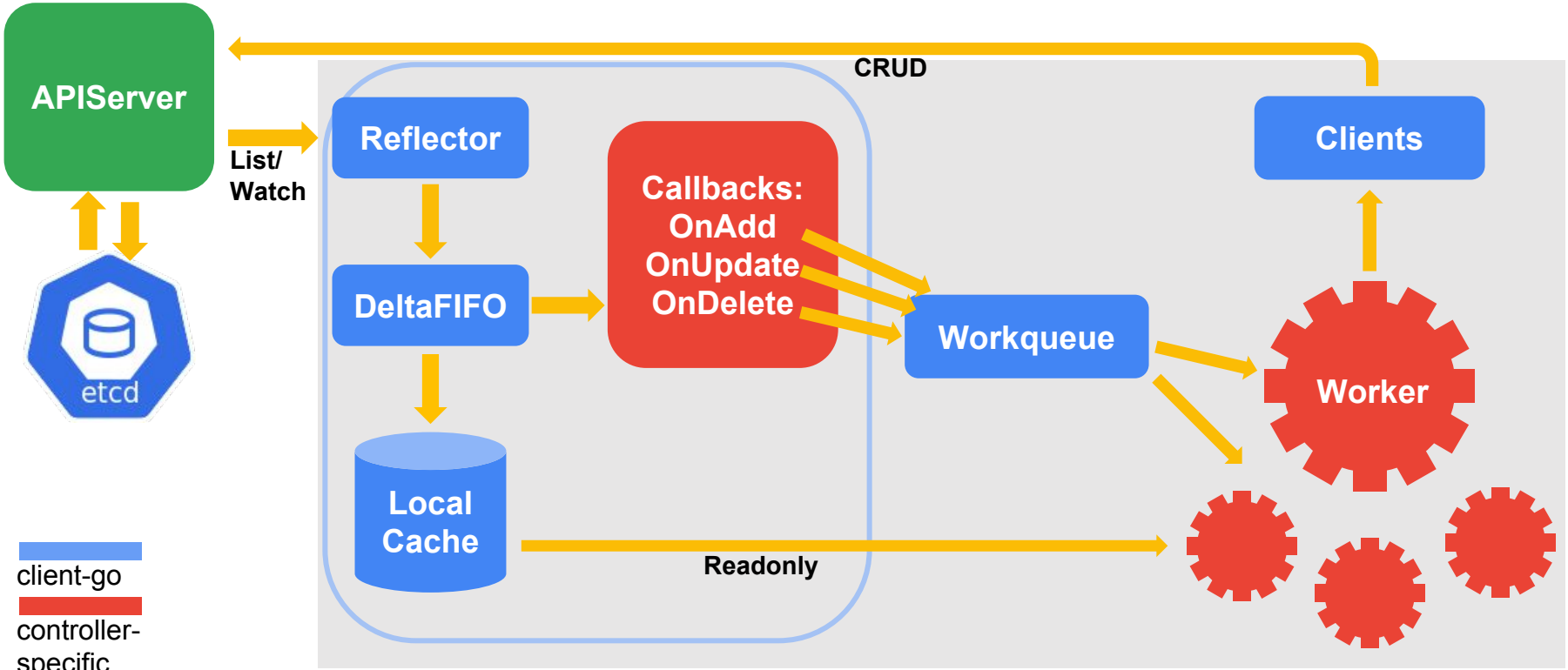
Kubernetes controller workflow



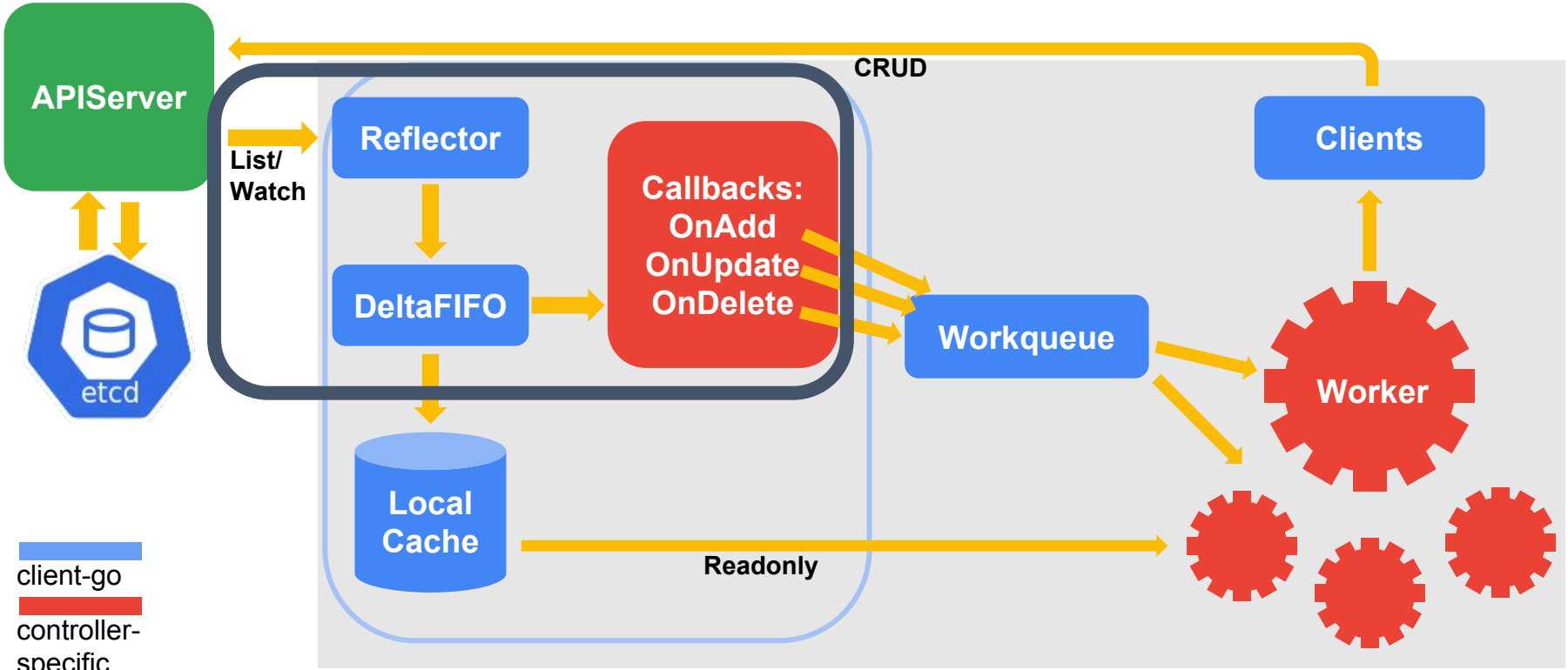
Kubernetes controller workflow



Kubernetes controller workflow



Kubernetes controller workflow



List and Watch

Pseudo code


```
1   func ListAndWatch() error {
2           list, err = listWatcher.List()
```

List and Watch

Pseudo code


```
1  func ListAndWatch() error {
2          list, err = listWatcher.List()
3          resourceVersion = list.GetResourceVersion()
```

Extract the actual ResourceVersion

List and Watch

Pseudo code


```
1  func ListAndWatch() error {
2          list, err = listWatcher.List()
3          resourceVersion = list.GetResourceVersion()
4
5          for {
6                  w, err = listWatcher.Watch(ListOptions{ResourceVersion: resourceVersion})
```

Start watch with latest ResourceVersion

List and Watch

Pseudo code


```
1  func ListAndWatch() error {
2          list, err = listWatcher.List()
3          resourceVersion = list.GetResourceVersion()
4
5          for {
6                  w, err = listWatcher.Watch(ListOptions{ResourceVersion: resourceVersion})
7                  if err {
8                          if err.IsError("connection refused") {
9                                  sleep(time.Second)
10                                 continue
11                         }
```

Most likely apiserver is not responsive.

List and Watch

Pseudo code


```
1   func ListAndWatch() error {
2           list, err = listWatcher.List()
3           resourceVersion = list.GetResourceVersion()
4
5           for {
6                   w, err = listWatcher.Watch(ListOptions{ResourceVersion: resourceVersion})
7                   if err {
8                           if err.IsError("connection refused") {
9                                   sleep(time.Second)
10                                  continue
11                          }
12                          HandleError(err)
13                          return nil
14                  }
```

Watch closed normally (EOF) or unexpected error

List and Watch

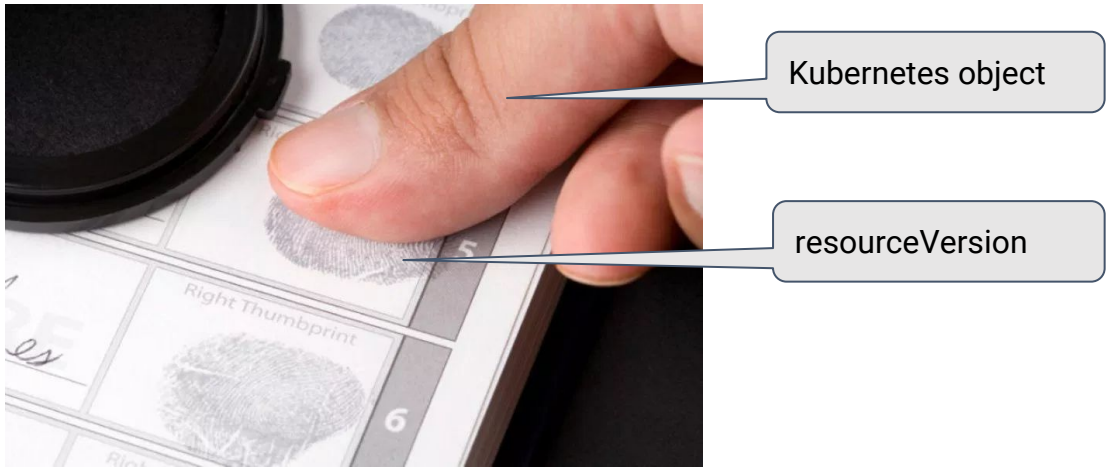
Pseudo code


```
1  func ListAndWatch() error {
2          list, err = listWatcher.List()
3          resourceVersion = list.GetResourceVersion()
4
5          for {
6                  w, err = listWatcher.Watch(ListOptions{ResourceVersion: resourceVersion})
7                  if err {
8                          if err.IsError("connection refused") {
9                                  sleep(time.Second)
10                                 continue
11                         }
12                         HandleError(err)
13                         return nil
14                 }
15                 watchHandler(w, &resourceVersion)
16         }
17 }
```

watchHandler watches w and keeps *resourceVersion up to date

Fingerprint of kubernetes object: resourceVersion

A resourceVersion is valid on a single kind of resource across namespaces.



Recap: resourceVersion

Everything has a ResourceVersion:

- Changes every time when you write to the storage
- Individual API object (e.g. a Pod) has ResourceVersion
- For a list of API objects (e.g. a PodList)
  - The entire list has a ResourceVersion
  - Each API object in list items has ResourceVersion

The ResourceVersion of the top-level list is what should be used when starting a watch to observe events occurring after that list was populated.

Recap: resourceVersion

- ListOption in **List** Request
  - Unspecified: etcd
  - RV>0: the result is at least as fresh as given RV
  - RV=0: APIServer cache (stale read: #59848)

Recap: resourceVersion

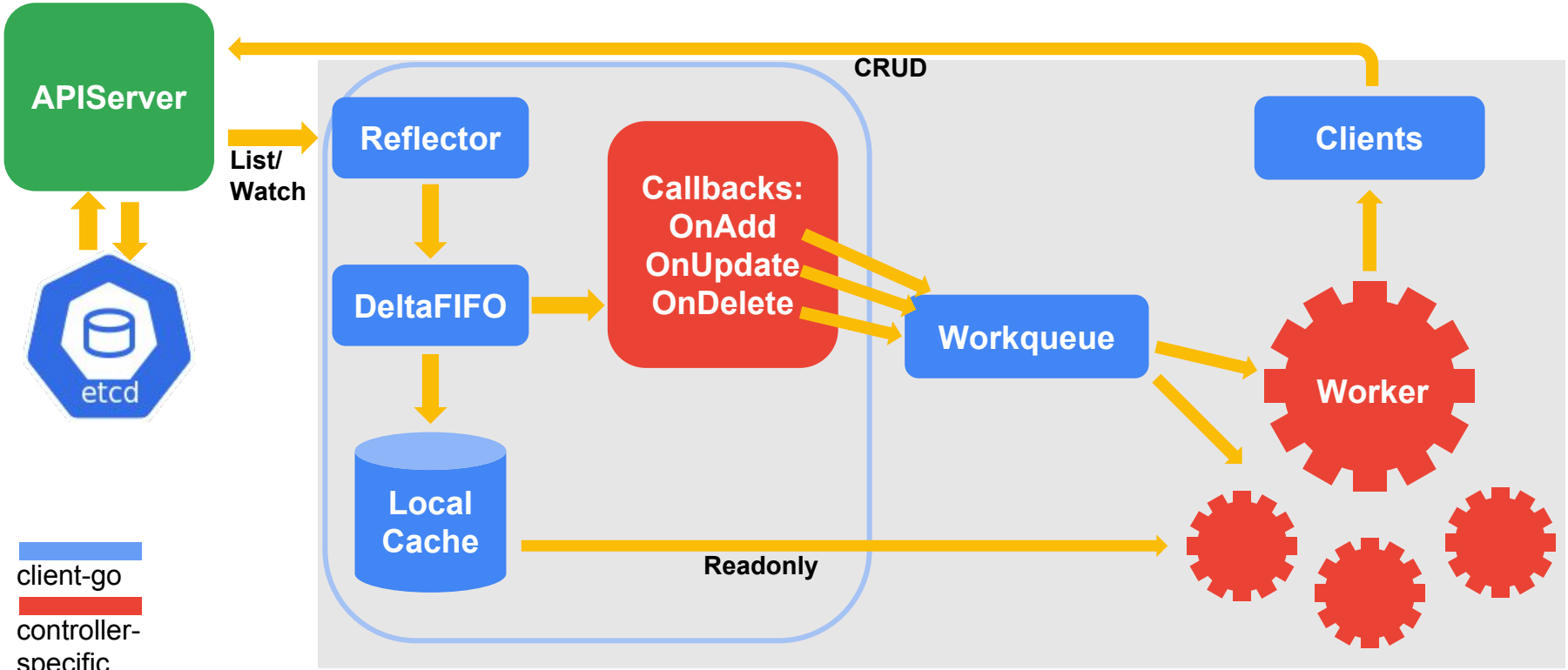
- ListOption in **Watch** Request
  - Unspecified: unspecified time point
  - RV=0: the result is an "ADDED" event for every existing object followed by events for changes that occur after the watch was established
    - (main reason: backwards compatibility-- #13910)
  - Best practice: always specify last listed/watched RV

KubeCon | CloudNativeCon

North America 2018

Watch Event on kube-scheduler, kube-controller-manager, kublet…

Kubernetes controller workflow



Mini Scheduler

Mini Scheduler

Watches:

- Node
- Pod

```
136    // New returns a Scheduler
137    func New(client clientset.Interface,
138            nodeInformer coreinformers.NodeInformer,
139            podInformer coreinformers.PodInformer,
```

Mini Scheduler “business” logic

Pseudo code

```
func InitEventHandlers(PodInformer) {
        // scheduled pod cache
        PodInformer.AddEventHandler(
                FilteringResourceEventHandler{
                        FilterFunc: func(obj interface{}) bool {
                                return IsScheduled(obj.Pod())
                        }
                        Handler: {
                                OnAdd: addPodToCache,
                                OnUpdate: updatePodInCache,
                                OnDelete: deletePodFromCache,
                        }
                }
        )

        // unscheduled pod queue
        PodInformer.AddEventHandler(
                FilteringResourceEventHandler{
                        FilterFunc: func(obj interface{}) bool {
                                return !IsScheduled(obj.Pod())
                        }
                        Handler: {
                                OnAdd: addPodToSchedulingQueue,
                                OnUpdate: updatePodInSchedulingQueue,
```

- SchedulingQueue for pods waiting to be scheduled
- PodCache for scheduled pods
- NodeCache for existing nodes

kube-scheduler


```
// New returns a Scheduler
func New(client clientset.Interface,
        nodeInformer coreinformers.NodeInformer,
        podInformer coreinformers.PodInformer,
        pvInformer coreinformers.PersistentVolumeInformer,
        pvcInformer coreinformers.PersistentVolumeClaimInformer,
        replicationControllerInformer coreinformers.ReplicationControllerInformer,
        replicaSetInformer appsinformers.ReplicaSetInformer,
        statefulSetInformer appsinformers.StatefulSetInformer,
        serviceInformer coreinformers.ServiceInformer,
        pdbInformer policyinformers.PodDisruptionBudgetInformer,
        storageClassInformer storageinformers.StorageClassInformer,
        recorder record.EventRecorder,
        schedulerAlgorithmSource kubeschedulerconfig.SchedulerAlgorithmSource,
        opts ...func(o *schedulerOptions)) (*Scheduler, error) {
```

pkg/scheduler/scheduler.go

Watches:

- Node
- Pod
- PV
- PVC
- RC
- RS
- Stateful set
- Service
- PDB
- Storage class

Key Takeaways

- A Kubernetes Watch Event is an efficient resource change notification
- Watch Event is the key to Kubernetes level triggering and soft reconciliation concept
- Watch is trustworthy and efficient
- Use Informer! Don’t misuse Watch!

Thanks!
Enjoy Seattle!

?