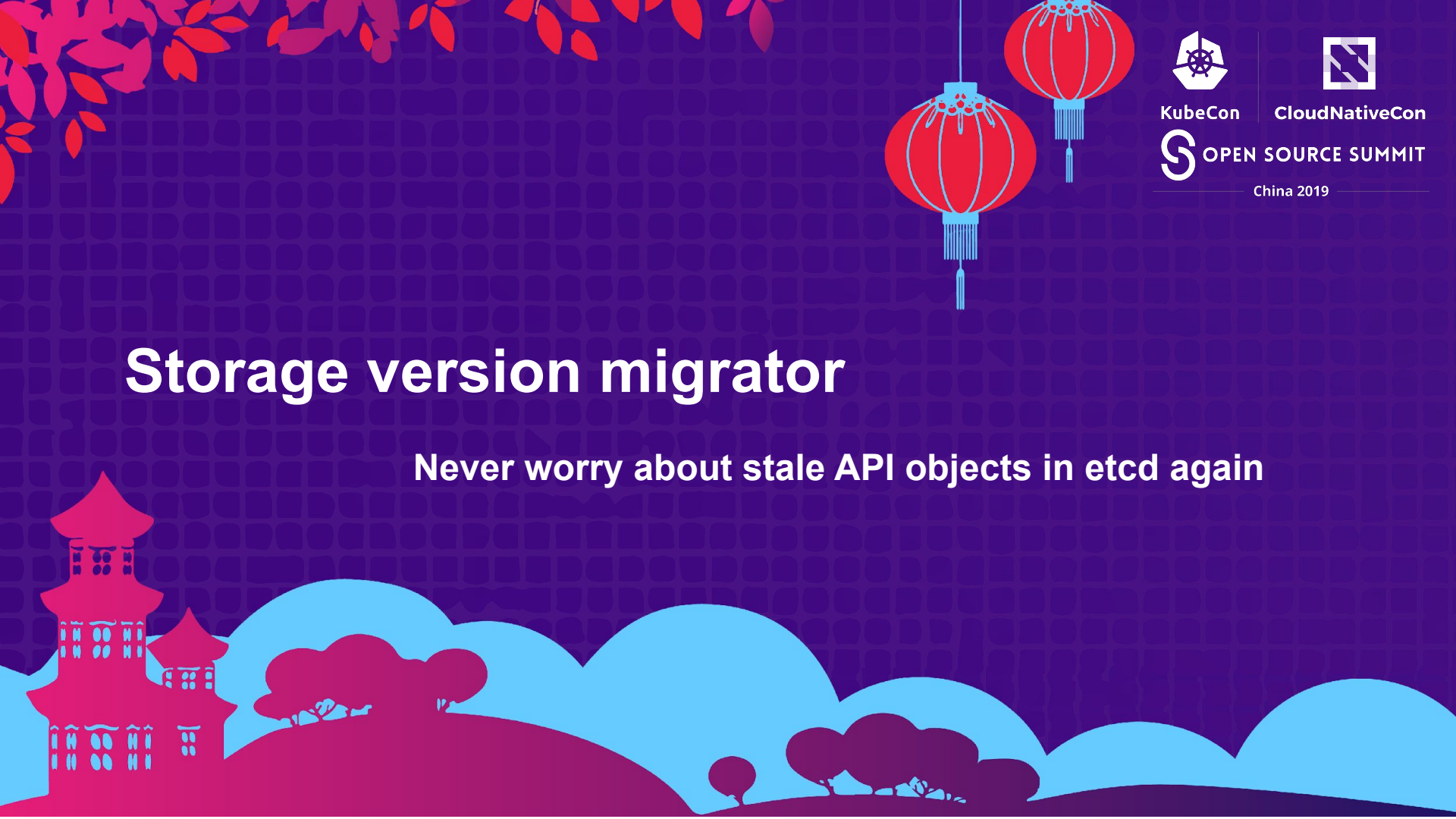


Storage version migrator

Never worry about stale API objects in etcd again

徐超 Chao Xu (caesarxuchao@github)

Senior Software Engineer at Google Kubernetes team. Active contributor to Kubernetes.

Agenda

How Kubernetes stores objects in etcd

Risks of stale objects in etcd

The storage migrator

Agenda

How Kubernetes stores objects in etcd

Risks of stale objects in etcd

The storage migrator

API Group & Version


```
$ curl 127.0.0.1:8080
{
  "paths": [

    "/apis/admissionregistration.k8s.io/v1beta1",
    "/apis/apiextensions.k8s.io/v1beta1",
    "/apis/apiregistration.k8s.io/v1",
    "/apis/apiregistration.k8s.io/v1beta1",
    "/apis/apps/v1",
    "/apis/apps/v1beta1",
    "/apis/apps/v1beta2",
    "/apis/authentication.k8s.io/v1",
    "/apis/authentication.k8s.io/v1beta1",
    "/apis/authorization.k8s.io/v1",
    "/apis/authorization.k8s.io/v1beta1",
    "/apis/autoscaling/v1",
    "/apis/autoscaling/v2beta1",
    "/apis/autoscaling/v2beta2",
    "/apis/batch/v1",
    "/apis/batch/v1beta1",
    "/apis/batch/v2alpha1",
    ...
  }
```

API Group/API Version


```
$ curl 127.0.0.1:8080
{
  "paths": [

    "/apis/admissionregistration.k8s.io/v1beta1",
    "/apis/apiextensions.k8s.io/v1beta1",
    "/apis/apiregistration.k8s.io/v1",
    "/apis/apiregistration.k8s.io/v1beta1",
    "/apis/apps/v1",
    "/apis/apps/v1beta1",
    "/apis/apps/v1beta2",
    "/apis/authentication.k8s.io/v1",
    "/apis/authentication.k8s.io/v1beta1",
    "/apis/authorization.k8s.io/v1",
    "/apis/authorization.k8s.io/v1beta1",
    "/apis/autoscaling/v1",
    "/apis/autoscaling/v2beta1",
    "/apis/autoscaling/v2beta2",
    "/apis/batch/v1",
    "/apis/batch/v1beta1",
    "/apis/batch/v2alpha1",
    ...
  }
```

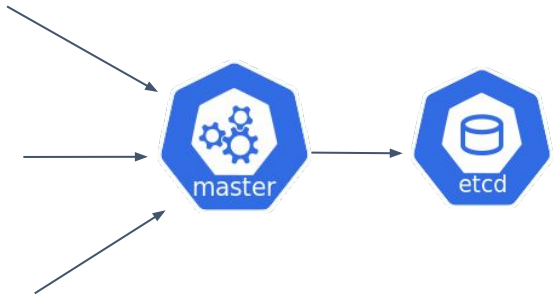
Multi-version RESTful API

RESTful API

/apis/batch/v1beta1/jobs

/apis/batch/v1/jobs

/apis/batch/v2alpha1/jobs



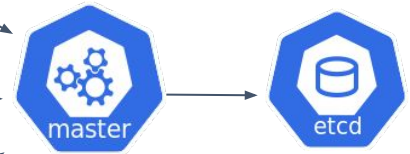
Multi-version RESTful API

RESTful API

GET /apis/batch/v1beta1/jobs/namespace/default/pi

GET /apis/batch/v1/jobs/namespace/default/pi

GET /apis/batch/v2alpha1/jobs/namespace/default/pi



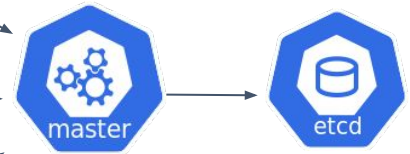
Multi-version RESTful API

RESTful API

GET /apis/batch/v1beta1/jobs/namespace/default/pi

GET /apis/batch/v1/jobs/namespace/default/pi

GET /apis/batch/v2alpha1/jobs/namespace/default/pi




```yaml
apiVersion: batch/v1
kind: Job
metadata:
  name: pi
spec:
  template:
  ...
```

Creating an Object

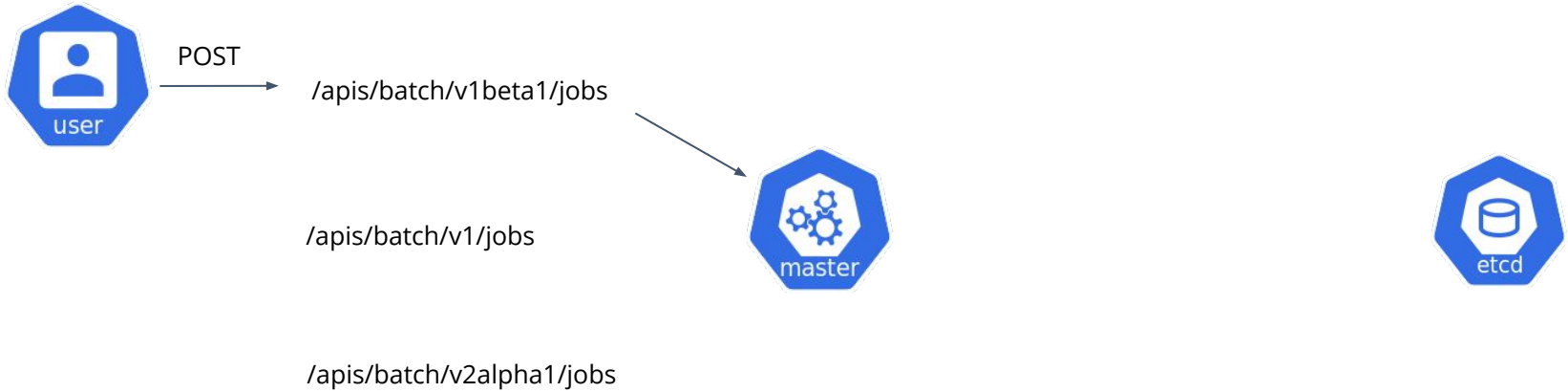
RESTful API

/apis/batch/v1beta1/jobs

/apis/batch/v1/jobs

/apis/batch/v2alpha1/jobs

Creating an Object



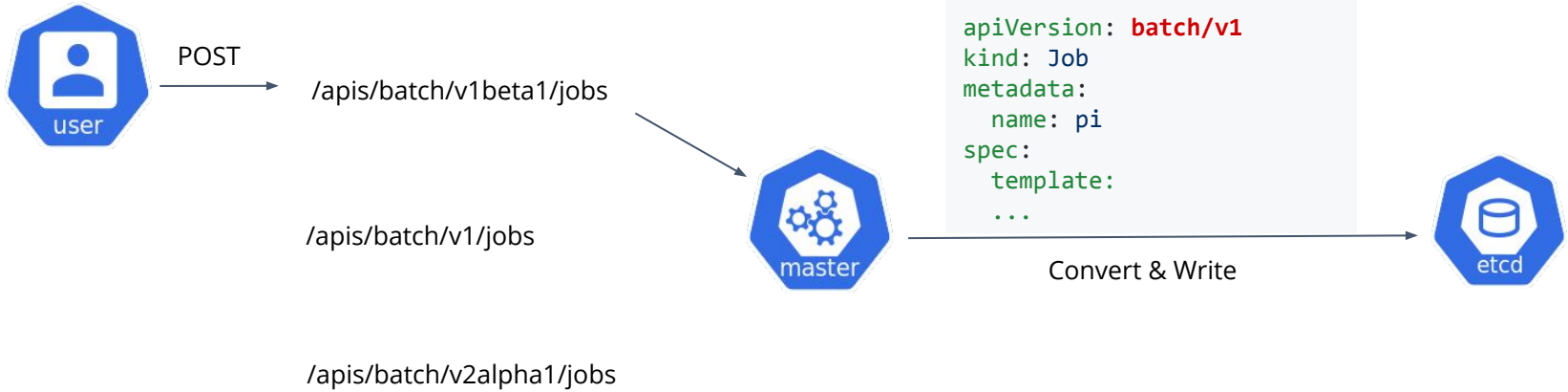
Creating an Object

API Version of the request must match the URL

POST https://localhost:8080/apis/**batch/v1beta1**/namespaces/default/jobs

```
apiVersion: batch/v1beta1
kind: Job
metadata:
  name: pi
spec:
  template:
    spec:
      containers:
      - name: pi
        image: perl
        ...
```

Creating an Object



Storage Version

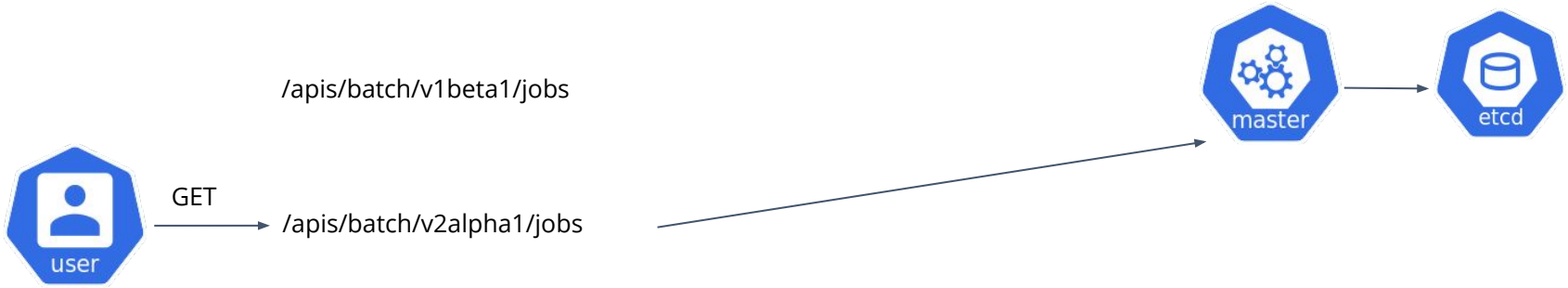
- Built-in resources: tied to API server version
- CRD: defined in CRD.Spec

Reading an Object

RESTful API

/apis/batch/v1/jobs

/apis/batch/v1beta1/jobs



Reading an Object

RESTful API

/apis/batch/v1/jobs

/apis/batch/v1beta1/jobs

/apis/batch/v2alpha1/jobs

Convert & Serve

```
// apiserver converts object
// to the requested version

apiVersion: batch/v2alpha1
kind: Job
metadata:
  name: pi
spec:
  template:
  ...
```

“Why”s

“Why”s

- Why does the API server support multiple versions of an API?

“Why”s

- Why does the API server support multiple versions of an API?

  Server-client compatibility

If only one version is supported...

v1.x

If only one version is supported...

If only one version is supported...

“Why”s

- Why does the API server support multiple versions of an API?

  Server-client compatibility

- Why does the API server convert objects to storage version before writing to etcd?

“Why”s

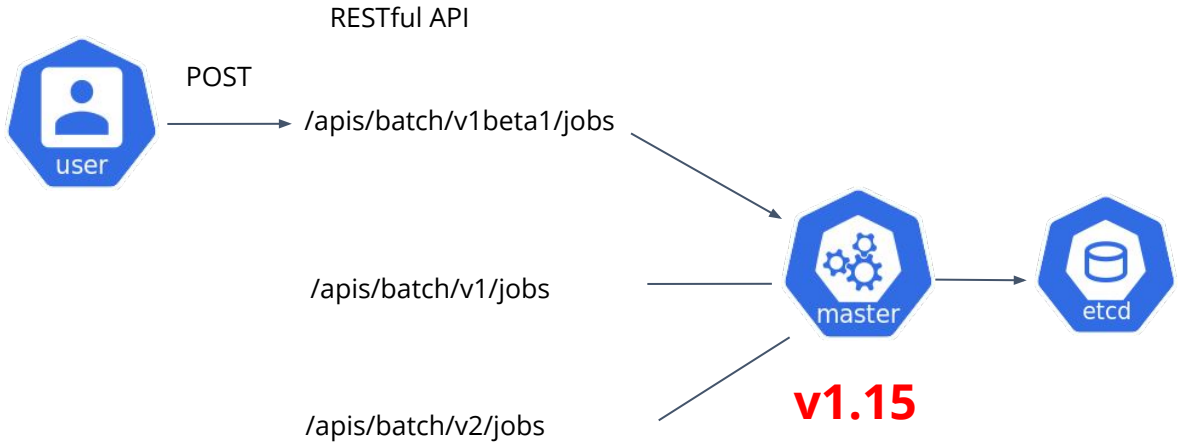
- Why does the API server support multiple versions of an API?

  Server-client compatibility

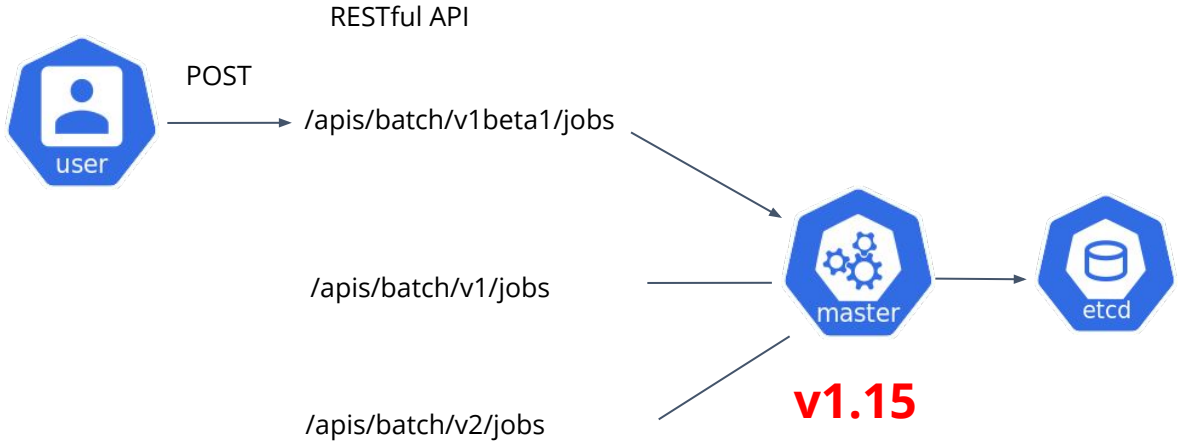
- Why does the API server convert objects to storage version before writing to etcd?

  Old server - new server compatibility

What's the storage version for?



What's the storage version for?

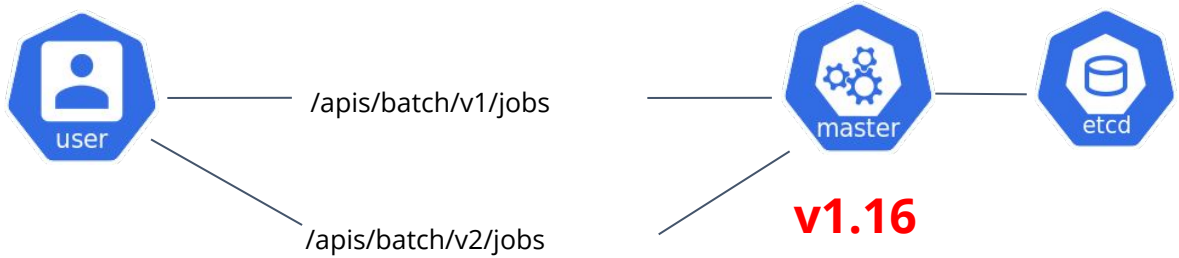


Etcd key: *registry/jobs/default/pi*

```yaml
apiVersion: batch/v1beta1
kind: Job
metadata:
  name: pi
spec:
  template:
  ...
```

What's the storage version for?

RESTful API

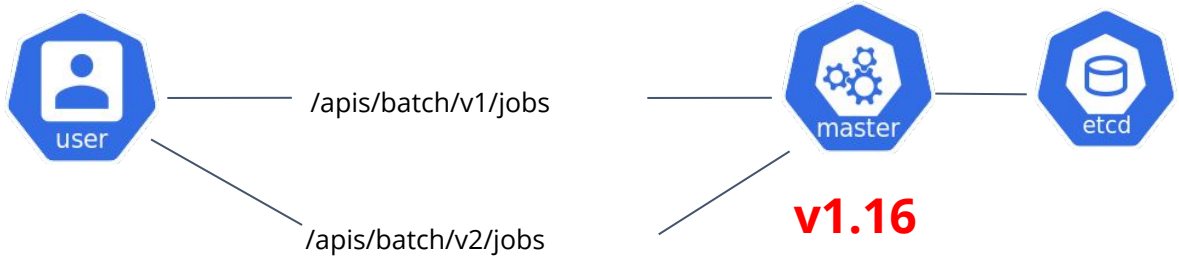


Etcd key: *registry/jobs/default/pi*

```yaml
apiVersion: batch/v1beta1
kind: Job
metadata:
  name: pi
spec:
  template:
  ...
```

What's the storage version for?

RESTful API

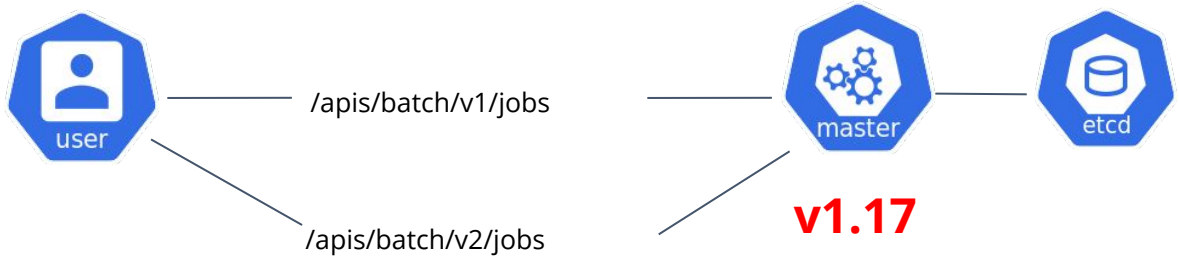


Etcd key: *registry/jobs/default/pi*

```yaml
apiVersion: batch/v1
kind: Job
metadata:
  name: pi
spec:
  template:
  ...
```

Storage version changes

RESTful API

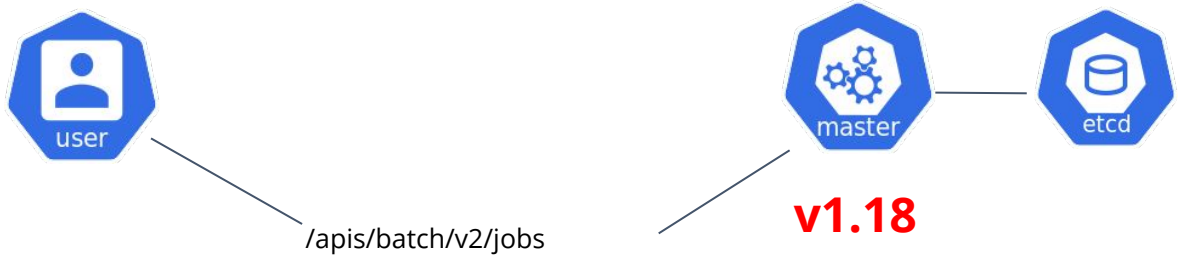


Etcd key: *registry/jobs/default/pi*

```yaml
apiVersion: batch/v2
kind: Job
metadata:
  name: pi
spec:
  template:
  ...
```

Storage version changes

RESTful API



Etcd key: *registry/jobs/default/pi*

```
apiVersion: batch/v2
kind: Job
metadata:
  name: pi
spec:
  template:
  ...
```

Implication to API Server Upgrades

Objects encoded in storage version => Safe upgrade

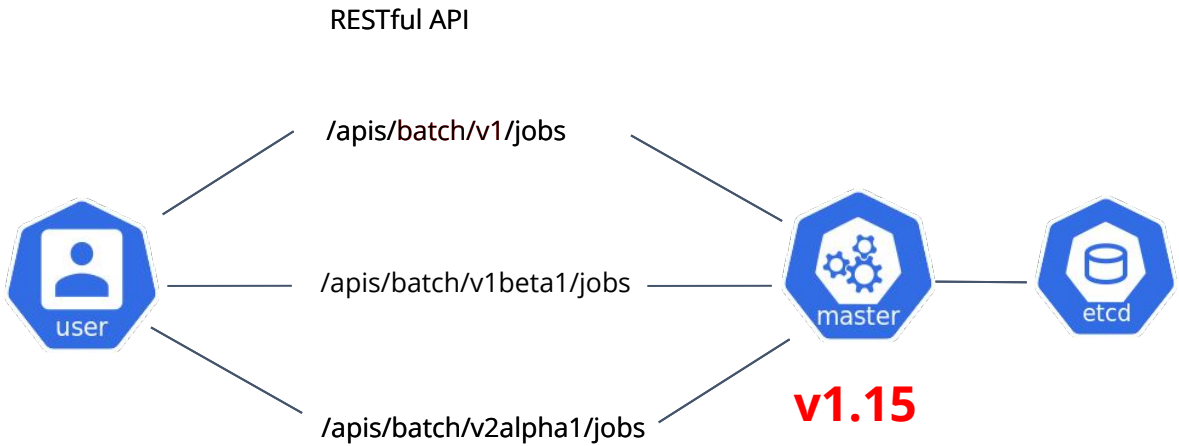
Agenda

How Kubernetes stores objects in etcd

Risks of stale objects in etcd

The storage migrator

Stale Objects

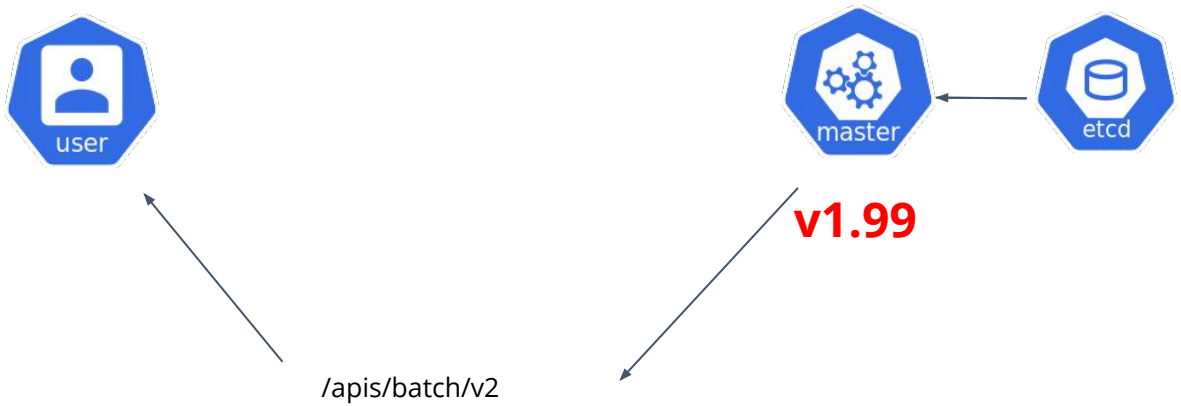


Etcd key: *registry/jobs/default/pi*

```yaml
apiVersion: batch/v1
kind: Job
metadata:
  name: pi
spec:
  template:
  ...
```

Stale Objects

RESTful API



Internal error: *batch.Job: no kind "Job" is registered for version "batch/v1"

Etcd key: *registry/jobs/default/pi*

```
apiVersion: batch/v1
kind: Job
metadata:
  name: pi
spec:
  template:
  ...
```

Agenda

How Kubernetes stores objects in etcd

Risks of stale objects in etcd

The storage migrator

Storage Migrator in One Line

A control loop that makes sure persisted API objects are encoded in their respective storage versions.

Highlights

Deployed via kubectl

Migrations API: Kubernetes-style

Resilient to failures

Vendor-agnostic

Deploying Storage Migrator


```
$ git clone
git@github.com:kubernetes-sigs/kube-storage-version-migrator.git

$ cd kube-storage-version-migrator

$ make local-manifests

$ kubectl apply -f manifests.local
```

Migration API


```yaml
apiVersion: migration.k8s.io/v1alpha1
kind: StorageVersionMigration
metadata:
  name: jobs.batch-cglnt
  namespace: kube-storage-migration
spec:
  resource:
    Group: batch
    Resource: jobs
    Version: v1
  continueToken: AL043vER
status:
  conditions:
  - lastUpdateTime: "2019-06-13T23:51:54Z"
    status: "True"
      type: Succeeded
```

Migration API


```yaml
apiVersion: migration.k8s.io/v1alpha1
kind: StorageVersionMigration
metadata:
  name: jobs.batch-cglnt
  namespace: kube-storage-migration
spec:
  resource:
    Group: batch
    Resource: jobs
    Version: v1
  continueToken: AL043vER
status:
  conditions:
  - lastUpdateTime: "2019-06-13T23:51:54Z"
    status: "True"
      type: Succeeded
```

Migration API


```yaml
apiVersion: migration.k8s.io/v1alpha1
kind: StorageVersionMigration
metadata:
  name: jobs.batch-cglnt
  namespace: kube-storage-migration
spec:
  resource:
    Group: batch
    Resource: jobs
    Version: v1
  continueToken: AL043vER
status:
  conditions:
  - lastUpdateTime: "2019-06-13T23:51:54Z"
    status: "True"
      type: Succeeded
```

Checking Migration Status

Wait for migration to complete before upgrading/downgrading API server:

```
$ kubectl wait --all --for=condition=Succeeded \
Storageversionmigrations.migration.k8s.io \
--namespace=kube-storage-migration
```

Agenda

How Kubernetes stores objects in etcd

Risks of stale objects in etcd

The storage migrator

- User-facing highlights
- The internals

Controllers

Migration Trigger Controller

API Server

Migrator Controller

Fetches discovery docs

Checks storage version changes

Time

Controllers

Migration Trigger Controller

API Server

Migrator Controller

Fetches discovery docs

Checks storage version changes

Server Upgrades

Time

Controllers

Migration Trigger Controller

API Server

Migrator Controller

Fetches discovery docs

Checks storage version changes

Server Upgrades

Fetches discovery docs

Checks storage version changes

Time

Controllers

Migration Trigger Controller

API Server

Migrator Controller

Fetches discovery docs

Checks storage version changes

Server Upgrades

Fetches discovery docs

Checks storage version changes

POST *Migrations*

Time

Controllers

Migration Trigger Controller

API Server

Migrator Controller

Fetches discovery docs

WATCH *Migrations*

Checks storage version changes

Server Upgrades

Fetches discovery docs

Checks storage version changes

POST *Migrations*

Time

Controllers

Migration Trigger Controller

API Server

Migrator Controller

Fetches discovery docs

WATCH *Migrations*

Checks storage version changes

Server Upgrades

Fetches discovery docs

⋮

Checks storage version changes

POST *Migrations*

Migrations from the Watch channel

Migrates resources whose storage versions change

Time

Controllers

Migration Trigger Controller

API Server

Migrator Controller

Fetches discovery docs

WATCH *Migrations*

Checks storage version changes

Server Upgrades

Fetches discovery docs

⋮

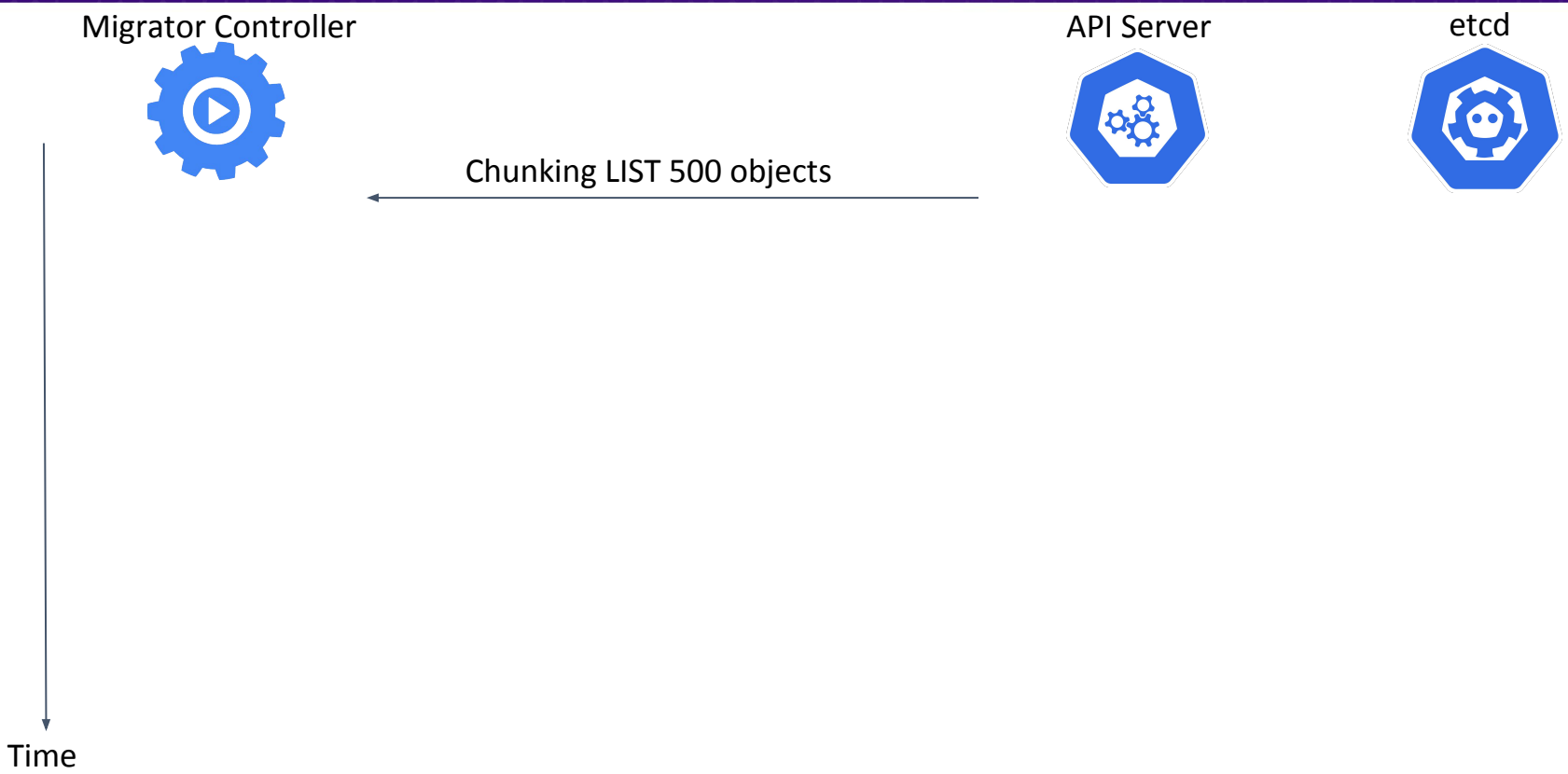
Checks storage version changes

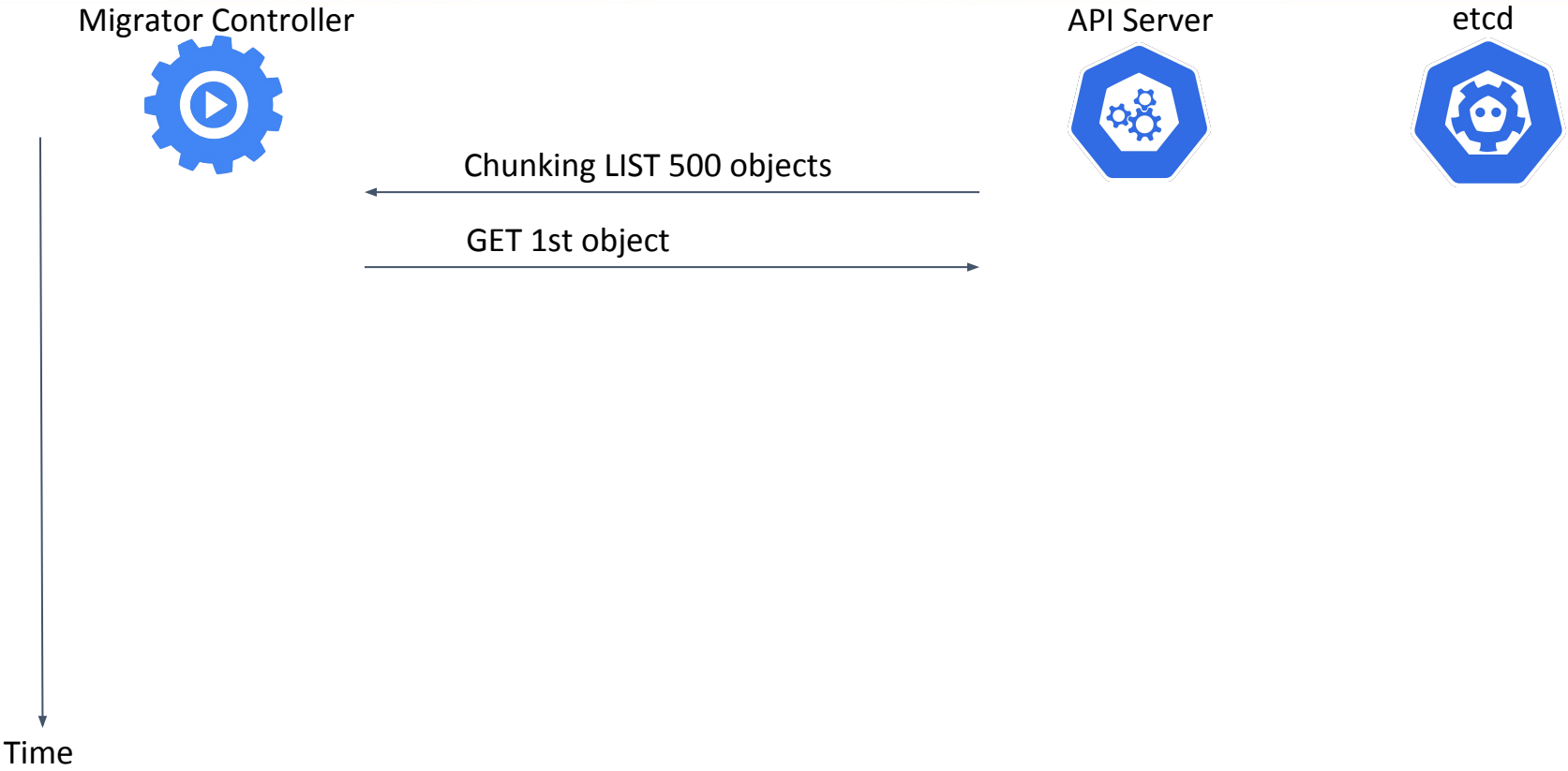
POST *Migrations*

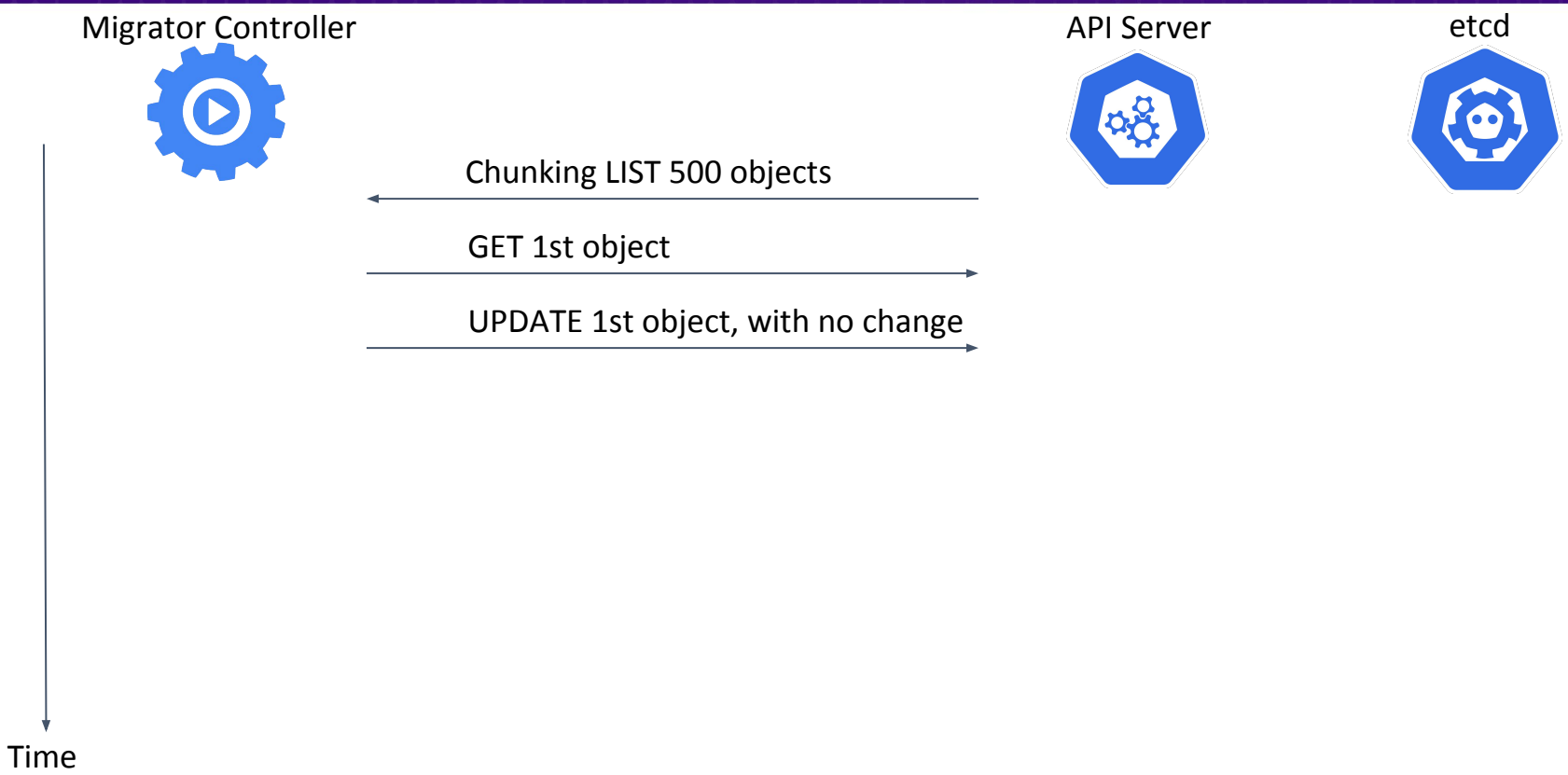
Migrations from the Watch channel

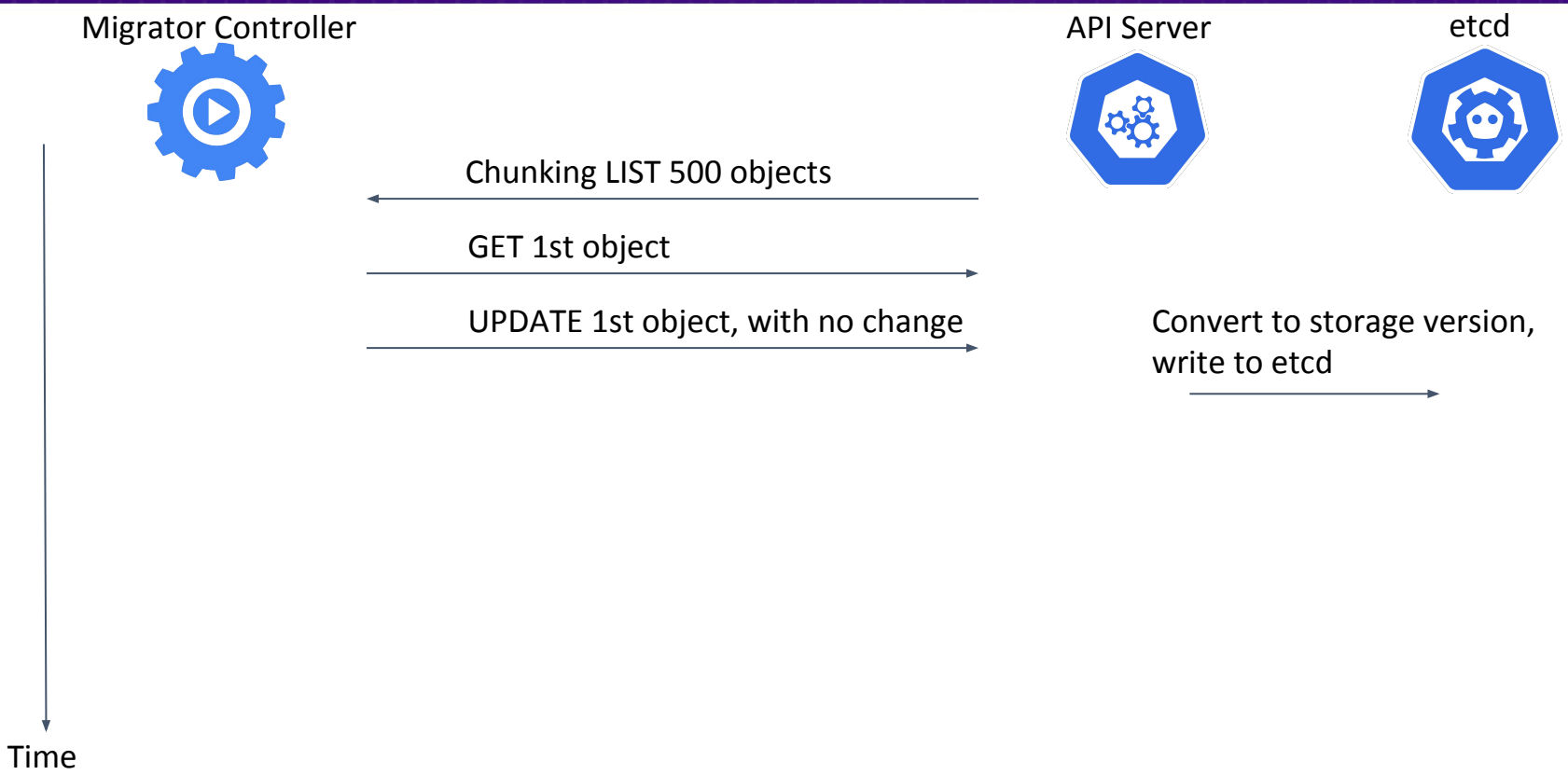
Migrates resources whose storage versions change

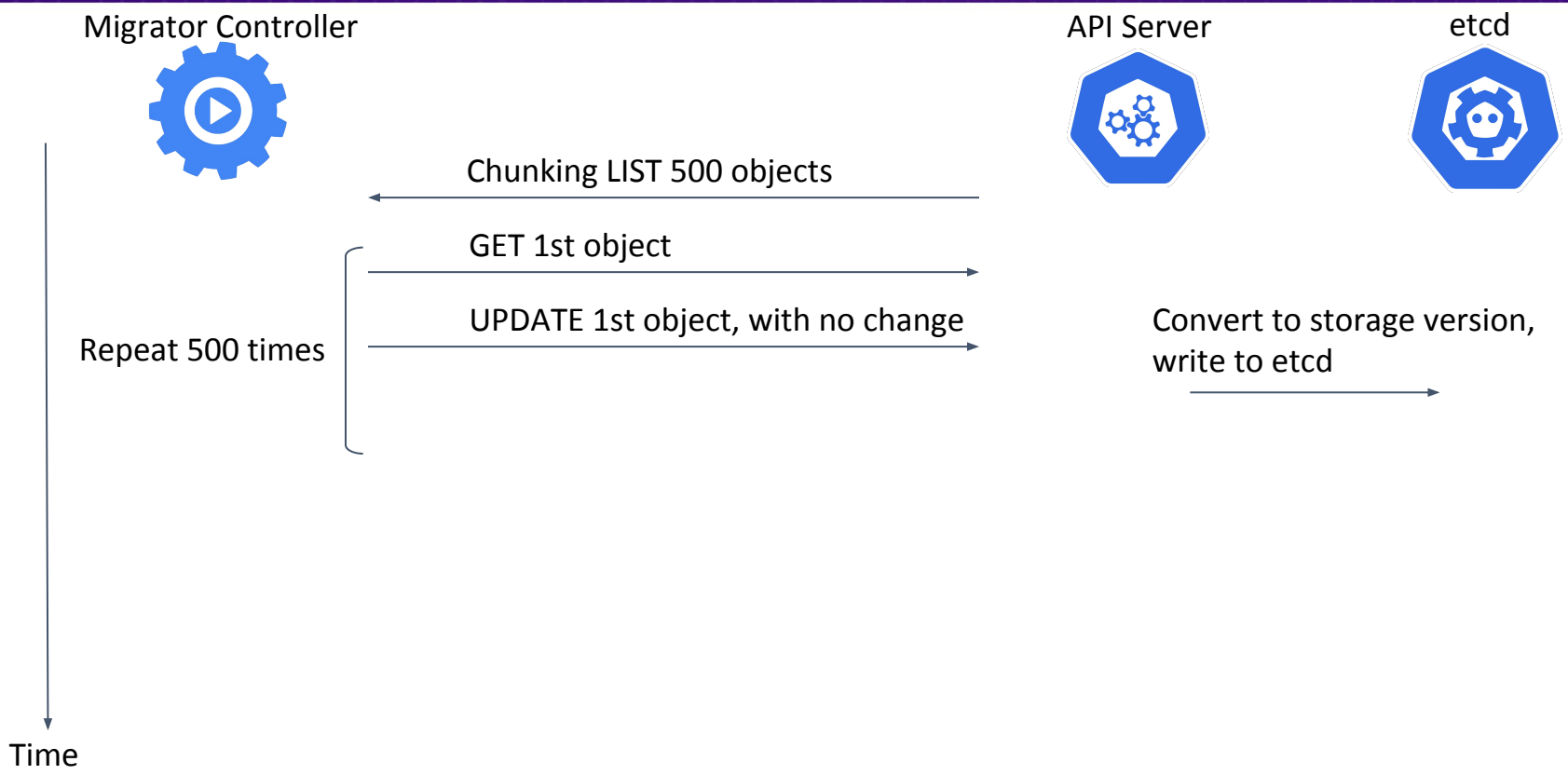
Time

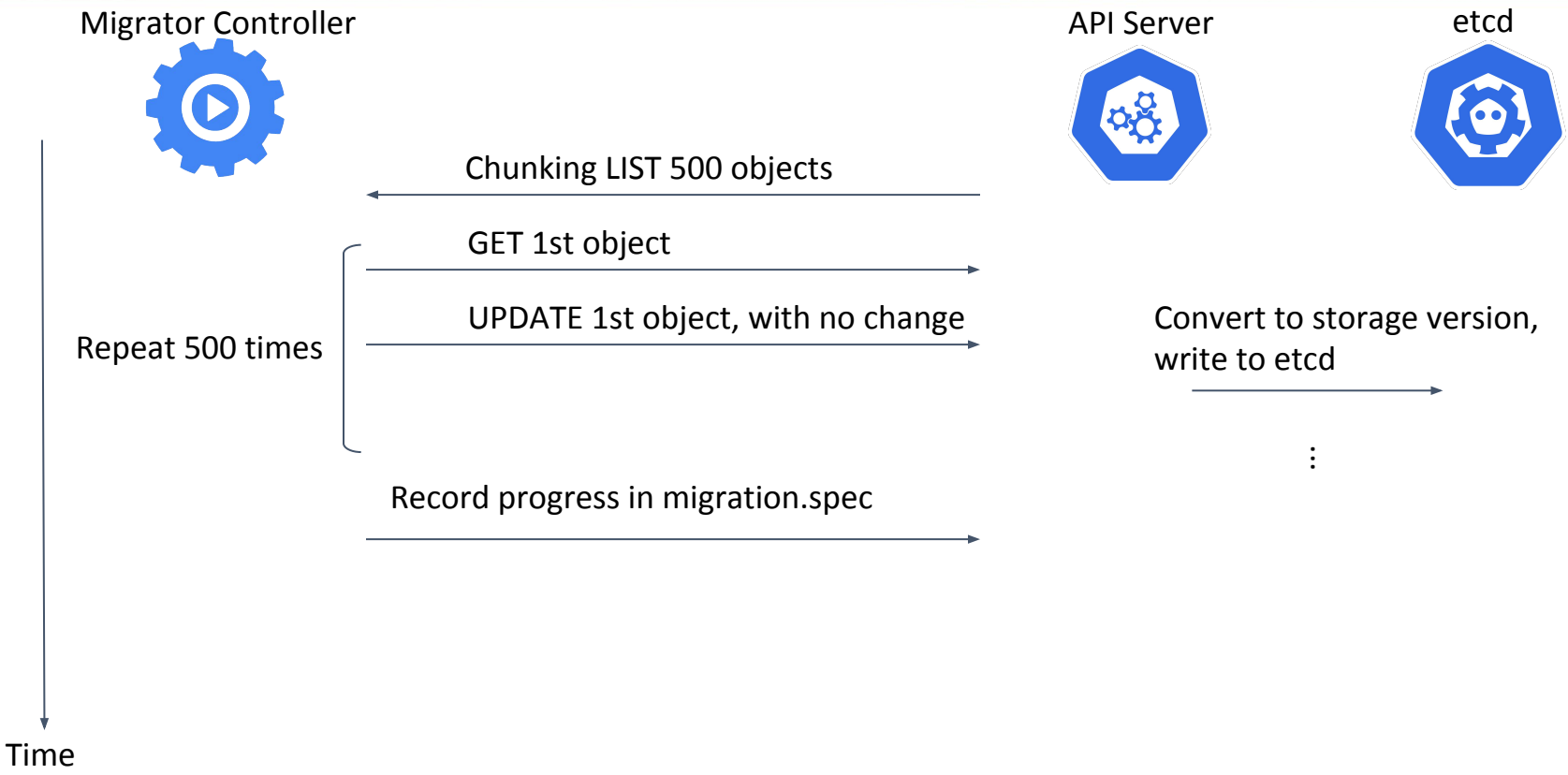

Migrator Controller
API Server
etcd
Chunking LIST 500 objects
Time

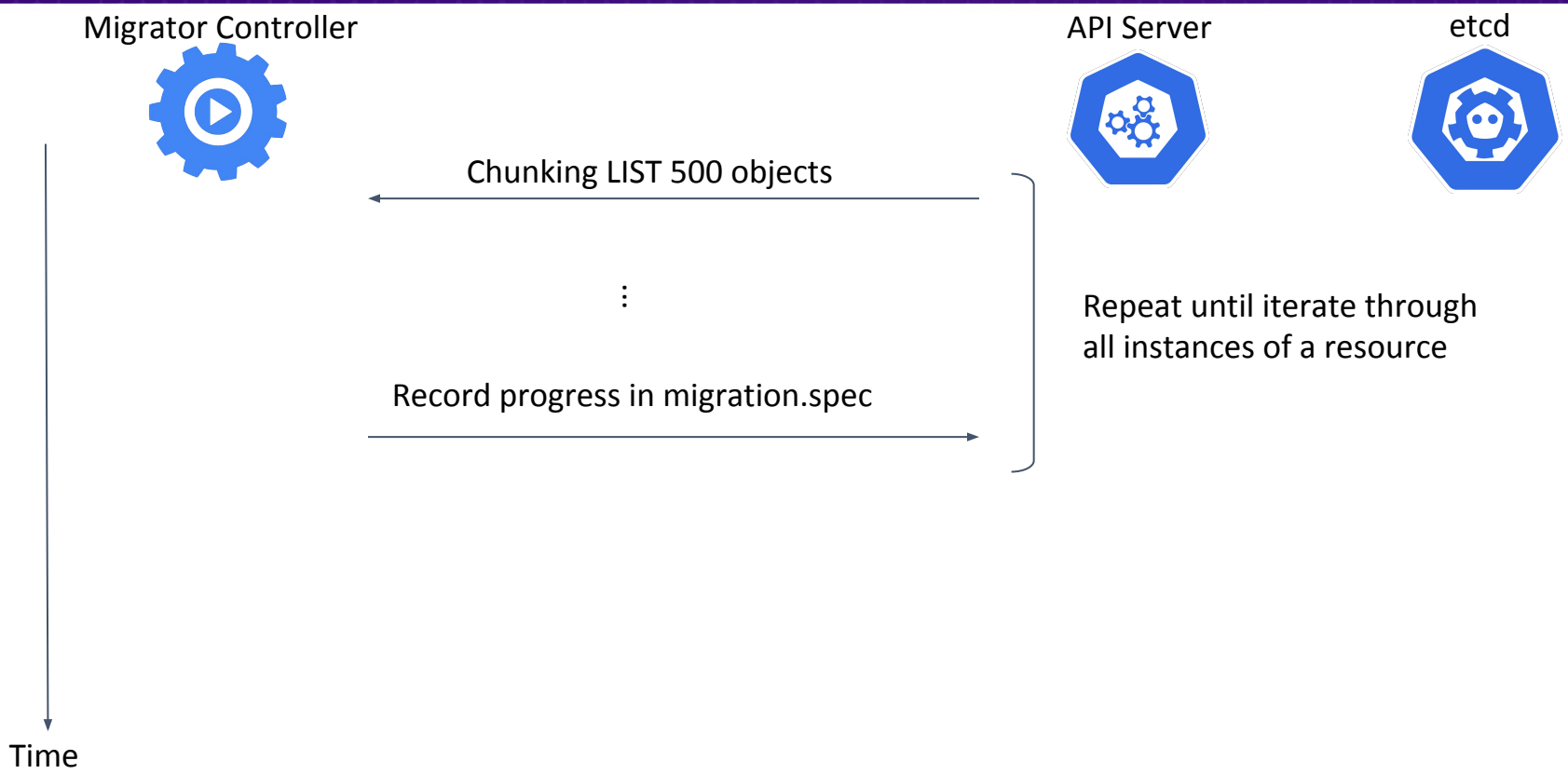

Migrator Controller
API Server
etcd
Chunking LIST 500 objects
GET 1st object
Time

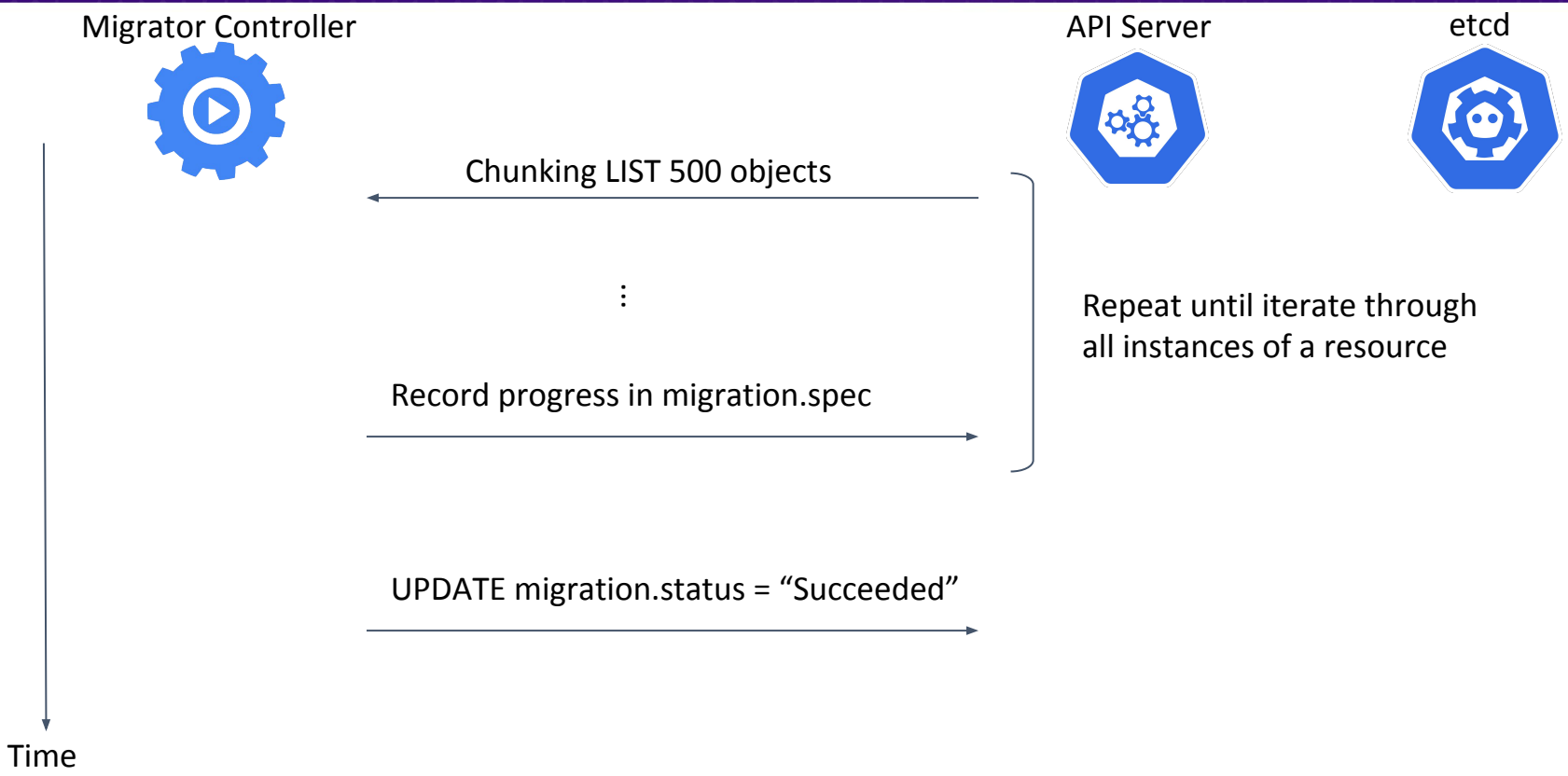

Migrator Controller
API Server
etcd
Chunking LIST 500 objects
GET 1st object
UPDATE 1st object, with no change
Time


Migrator Controller
API Server
etcd
Chunking LIST 500 objects
GET 1st object
UPDATE 1st object, with no change
Convert to storage version,
write to etcd
Time


Migrator Controller
API Server
etcd
Chunking LIST 500 objects
GET 1st object
UPDATE 1st object, with no change
Repeat 500 times
Convert to storage version,
write to etcd
Time


Migrator Controller
API Server
etcd
Chunking LIST 500 objects
GET 1st object
UPDATE 1st object, with no change
Repeat 500 times
Convert to storage version,
write to etcd
Record progress in migration.spec
Time


Migrator Controller
API Server
etcd
Chunking LIST 500 objects
Record progress in migration.spec
Repeat until iterate through all instances of a resource
Time


Migrator Controller
API Server
etcd
Chunking LIST 500 objects
Record progress in migration.spec
Repeat until iterate through all instances of a resource
UPDATE migration.status = “Succeeded”
Time

Timeline

Beta: 2019 Q2

GA: 2019 Q3

https://github.com/kubernetes-sigs/kube-storage-version-migrator

Agenda

How Kubernetes stores objects in etcd

Risks of stale objects in etcd

The storage migrator

BACKUP

Takeaways

How Kubernetes stores objects in etcd

Risks of stale objects in etcd

The storage migrator

- User's perspective
- The internals

Bonus: useful meta APIs used by the storage migrator

Non-consistent Chunking LIST


```
apiVersion: v1
kind: PodList
metadata:
    resourceVersion: 10245
    continue: ENCODED_CONTINUE_TOKEN
    ...
```

Non-consistent Chunking LIST

Continue token is returned with 410 error:

```
apiVersion: meta.k8s.io
kind: Status
metadata:
    continue: ENCODED_CONTINUE_TOKEN
code: 410
status: Failure
reason: Expired
...
```

RemainingItemCount in Chunking LIST Response


```
apiVersion: v1
kind: PodList
metadata:
    continue: ENCODED_CONTINUE_TOKEN
    remainingItemCount: 10245
     ...
```

Storage Migrator Roadmap

Unfreeze API Removal. Removing O(10k) lines of code.