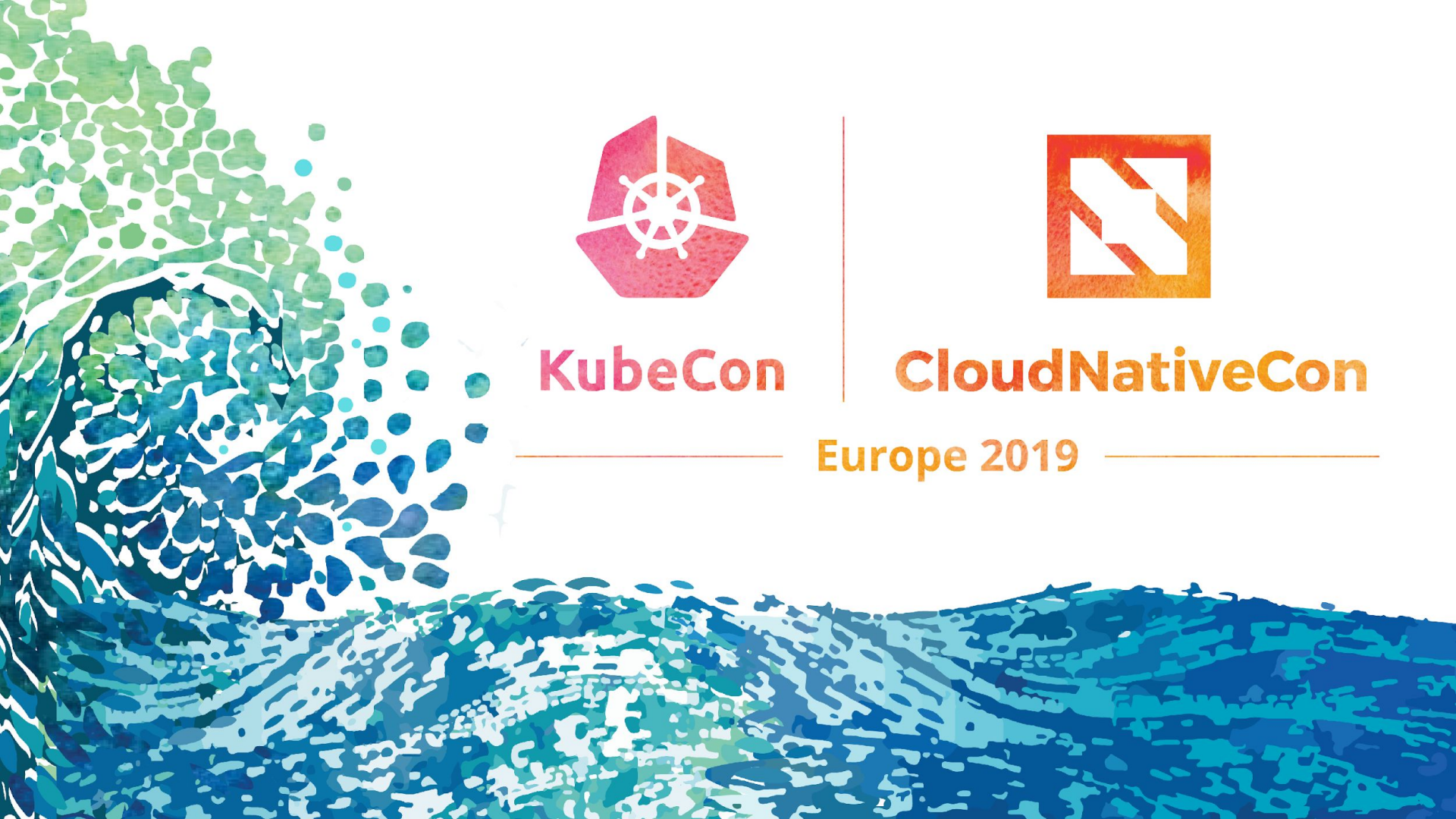
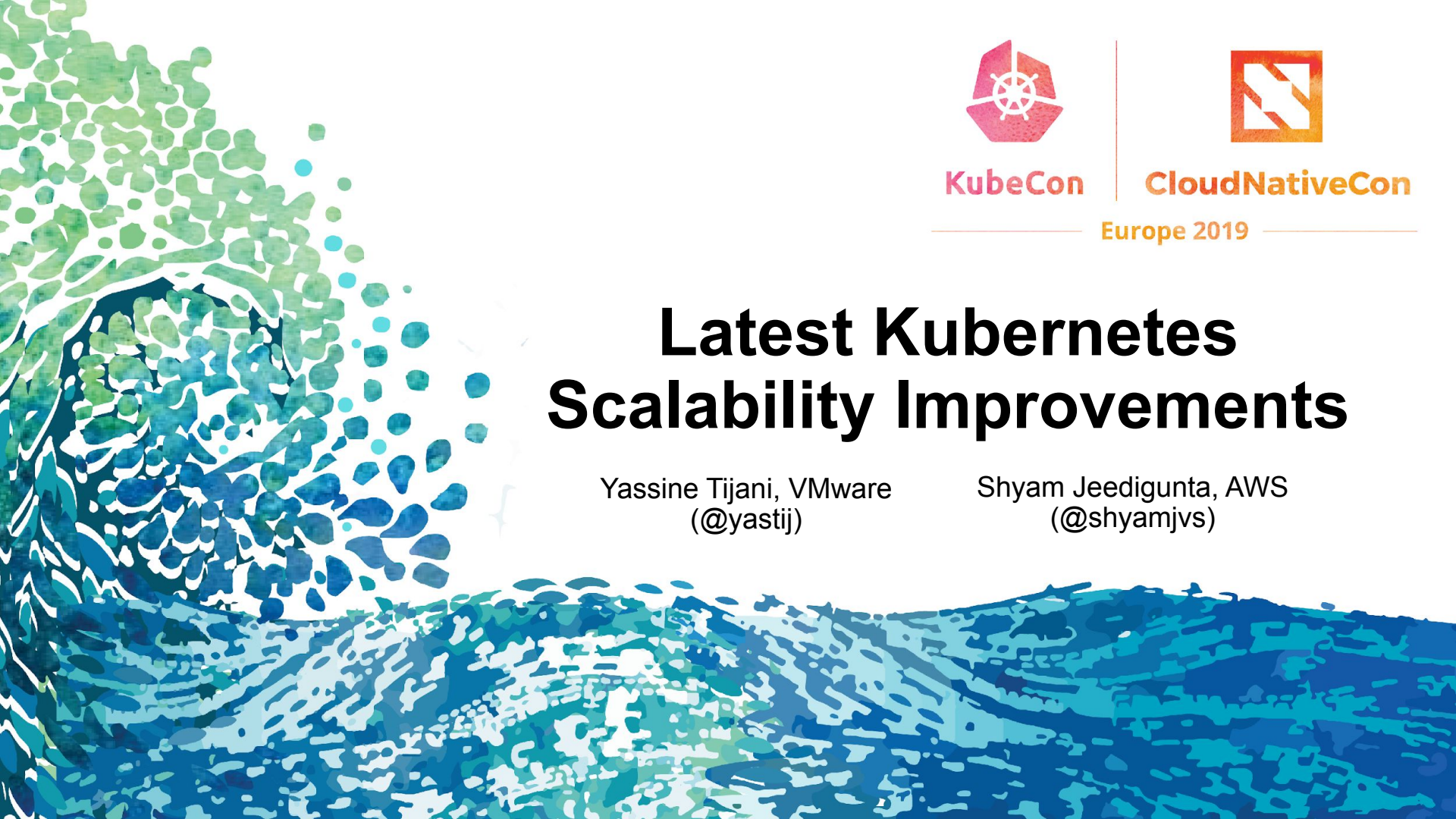

KubeCon
CloudNativeCon
Europe 2019



KubeCon | CloudNativeCon

Europe 2019

Latest Kubernetes Scalability Improvements

Yassine Tijani, VMware
(@yastij)

Shyam Jeedigunta, AWS
(@shyamjvs)

Background

Kubernetes *STARTED SCALING* to large clusters a while ago

Bigger clusters gained popularity

Usage patterns exposed some newer bottlenecks

So...

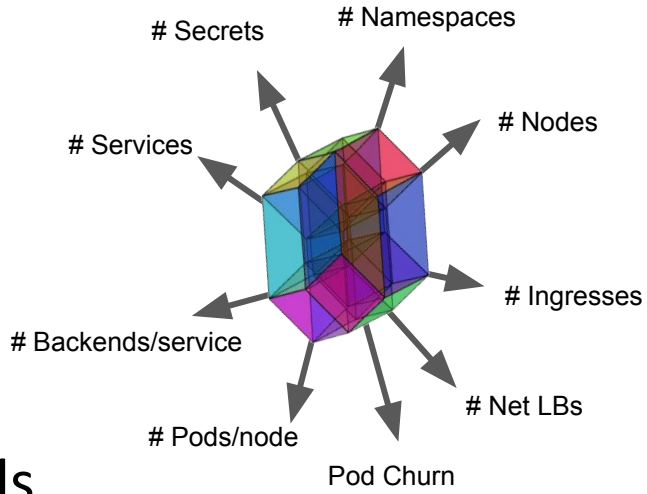
Background

We moved to scalability definition 2.0,

“Scalability is a multi-dimensional problem”
(for more see this past talk - https://sched.co/GrXy)

And…

Started working on improving various bounds



So.. What did we improve?

Too many node revisions

Problem:

- Too many node revisions in large clusters due to node heartbeats
- Made worse if too many images or volumes on the node
- Etcd disk fills up, causing NoSpace alarm
- Writes can’t happen anymore

Dimensions that suffer:

- #Nodes
- #Images/node
- #Volumes/node

Too many node revisions

How we solved?

- Split node objects from heartbeats
- Split node status updates from heartbeats
- Use new lease API to signal node heartbeats
- Continue using node status update also as liveness signal
- [Long-term] Reduce node-status update frequency to 1m

Feature availability:
1.13 (alpha)
1.14 (beta)

Node Lease API


```
apiVersion: coordination.k8s.io/v1
kind: Lease
metadata:
  creationTimestamp: 2019-04-16T13:12:35Z
  name: node-foo
  namespace: kube-node-lease
…
…
spec:
  holderIdentity: node-foo
  leaseDurationSeconds: 40
  renewTime: 2019-05-03T15:19:32.136799Z
```

heartbeats

```
status:
  …
  conditions:
    -       lastHeartbeatTime: "2019-05-05T18:30:46Z"
            lastTransitionTime: "2019-05-05T18:30:46Z"
            message: NodeController create implicit route
            reason: RouteCreated
            status: "False"
    -       lastHeartbeatTime: "2019-05-05T18:31:15Z"
            lastTransitionTime: "2019-05-05T18:30:46Z"
            message: kubelet is posting ready status. AppArmor enabled
            reason: KubeletReady
            status: "True"
            type: Ready
    …
  images:
    -       …
  volumesInUse
    -       …
  volumesAttached
    -       …
```

node status

Kubelet polling configs

Problem:

- Kubelet periodically polls secrets/configmaps it needs
- Can lead to many `GET secret/configmap` API calls
- Can eat away significant chunk of apiserver request queue
- Caching and reducing poll frequency, used as stopgaps

Dimensions that suffer:

- #Nodes
- #Secrets/node
- #Configmaps/node

Kubelet polling configs

How we solved?

- Switch kubelet to watch individual secrets/configmaps

Feature availability:

Enabled till 1.12.6 (but disabled from 1.12.7 due to golang bug)
Enabled till 1.13.4 (but disabled from 1.13.5 due to golang bug)
Enabled from 1.14 (with bug fixed by updating golang to 1.12)

Note: The bug is with kubelet TCP streams exhaustion if there are many (~250) configmaps/secrets needed by it

Scheduling performance

Problems:

- Scheduling throughput is low on large clusters:
  - ~80/s in 2k-node cluster
  - ~30/s in 5k-node cluster
- Scheduling throughput is very low when using pod anti-affinity:
  - < 5 pods/min in 5k-node cluster

Dimensions that suffer:

- Pod churn
- #Nodes

Scheduling performance

How we solved?

- Score only a percentage of nodes that were found feasible
- Improvement on the computing of affinity, splitted into phases
  - Find all pods that matches affinity/anti-affinity terms
  - Check then the topology matching of these pods
- Pod scheduling Latency improved (Available in 1.14)
  - improving the way we snapshot schedulers' cache

Events overload

Problem:

- Multiple scalability issues over the time
- Improve client-side filtering
- Improve UX
- Dimension to improve:

#API calls

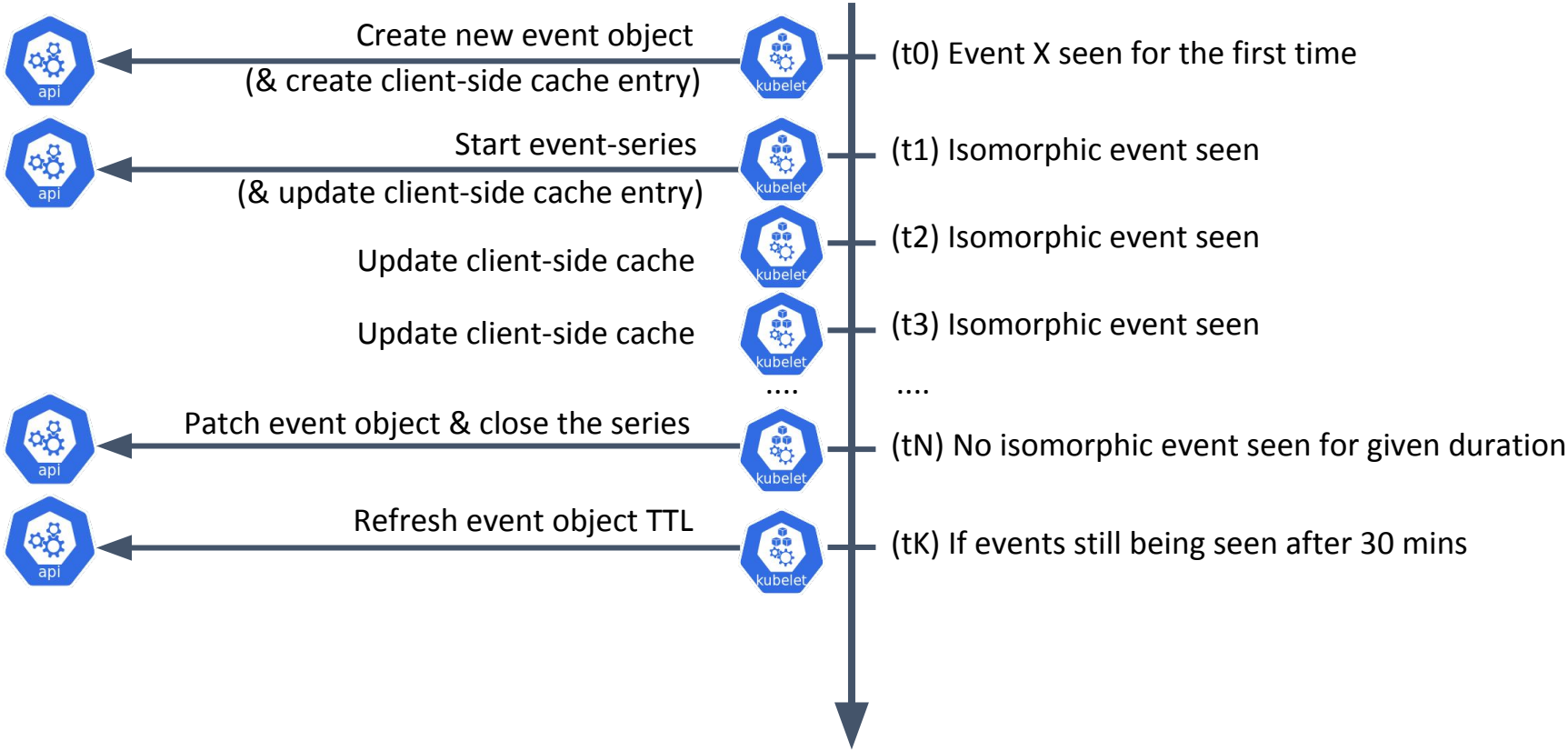
Events Scalability

How we solved?

- New Event API
- New deduplication logic that makes use of the new API
  - Concept of isomorphic Events
  - Avoid aggregation using an Event object

Reporting controller:	Kubelet
Reporting Instance:	node-foo
Regarding:	pod-bar
Related:	daemonset-foo
Type:	Warning
Action:	FailedCreatePodSandBox
Reason:	cannotAssignIP
Note:	Failed to create sandbox : ...

Events Scalability



watch restart cost

Problem

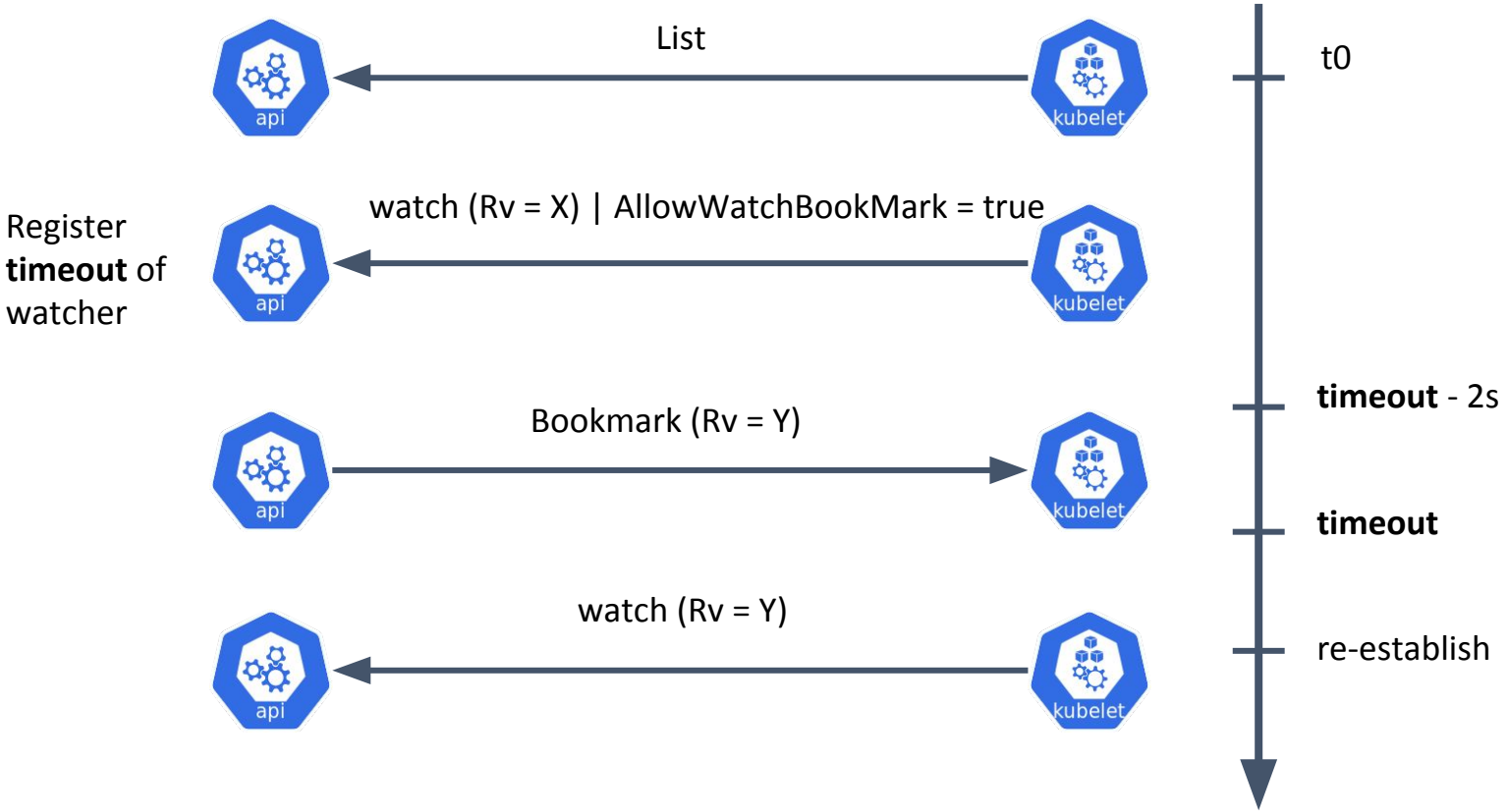
- Restarting watches overload the apiserver
- Users tend to get the famous “watch of v1.Foo ended with: too old resource version”

watch bookmark

Solution

- K8s 1.15 introduces WatchBookmark alpha feature to let clients know which resourceVersion they can use for watch
- Introduces a new Event type watch called “Bookmark”
- watch bookmark is backward compatible
- Dimension: no. of nodes

watch bookmark



watch bookmark

- There’s no guarantee that clients will receive a bookmark
- In practice it happens 2s before watch timeout
- Benchmark shows 40x improvement on event processing when re-establishing watch connections

What we plan to do next

Endpoint API

Improvements:

- One object per Endpoint
- Non-pod Endpoint
- Ready field for endpoint


KubeCon
CloudNativeCon
Europe 2019
Thank you!