KubeCon CloudNativeCon North America 2022
BUILDING FOR THE ROAD AHEAD
DETROIT 2022

SIG API Machinery advanced topics

David Eads, Red Hat (deads2k@)
Jeffrey Ying, Google (jefftree@)

Host: Federico Bongiovanni, Google (fedebongio@)

- ***The power and the Danger of Aggregated API Servers (David Eads)***
  we plan to explain the architecture around the Aggregated API servers in the Kubernetes API Machinery domain, and how they work chained together. What can you do with them and what you can't. More importantly we will go into concrete examples and recommendations on when to use it in concrete.

- ***OpenAPIv3 (Jeffrey Ying)***
  a powerlfull feature in Beta right now, and graduating to GA very soon. What is it good for? How can I use it? Advanced use cases, and GA Plan.

BUILDING FOR THE ROAD AHEAD

DETROIT 2022

OpenAPI V3

Jeffrey Ying, Google (@jefftree)

What is OpenAPI

The OpenAPI Specification defines a standard, language-agnostic interface to HTTP APIs allowing both human and computers to discover and understand the capabilities of the service

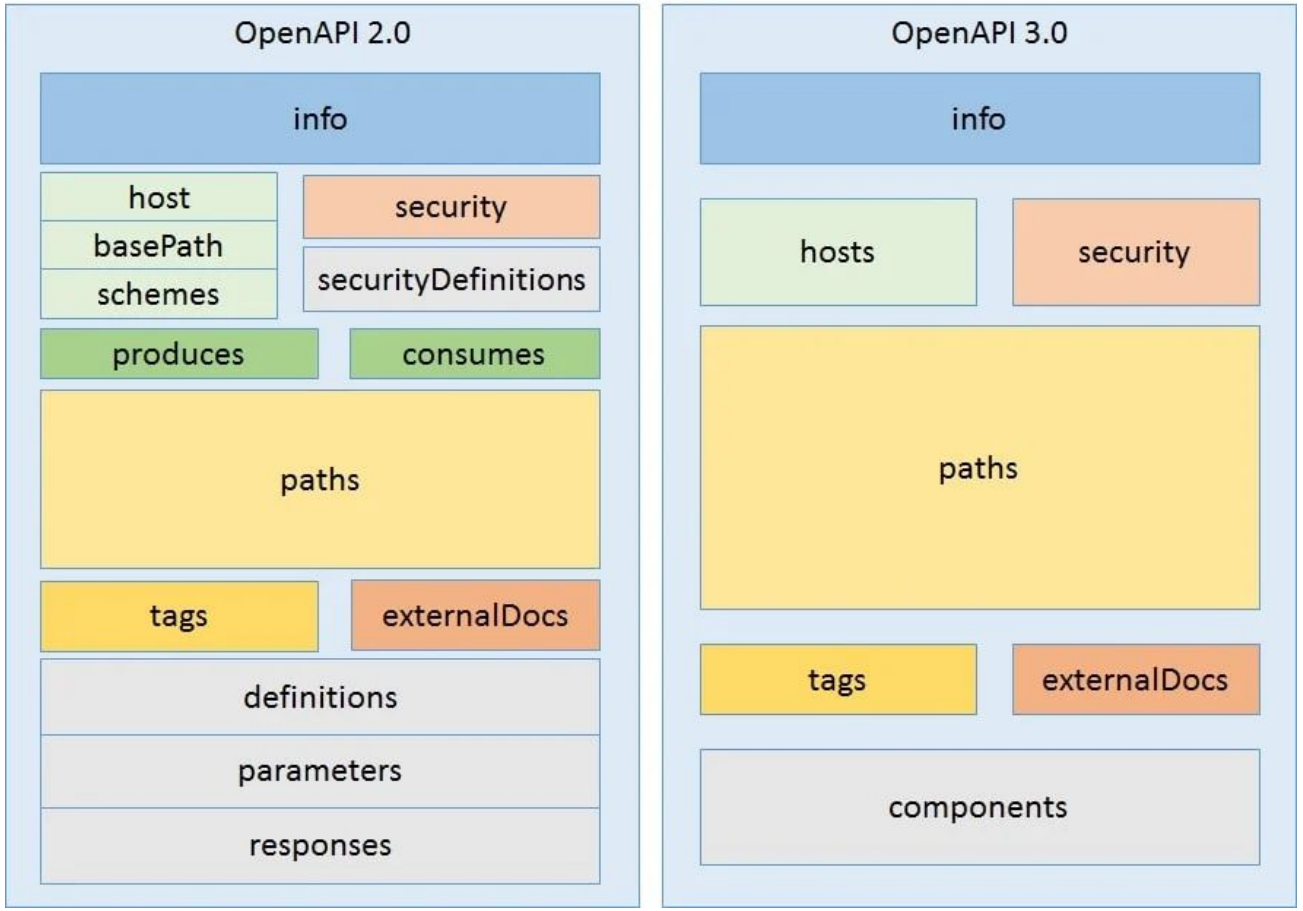
In Kubernetes:

- kubectl (explain)
- Autocompletion for UIs
- Generating documentation
- Generating clients


```
"post" : {
  "consumes" : [
    "*/*"
  ],
  "description" : "create a DaemonSet",
  "operationId" : "createAppsV1NamespacedDaemonSet",
  "parameters" : [
    {
      "in" : "body",
      "name" : "body",
      "required" : true,
      "schema" : {
        "$ref" : "#/definitions/io.k8s.api.apps.v1.DaemonSet"
      }
    },
    {
      "description" : "When present, indicates that modifications should not be persisted.
      "in" : "query",
      "name" : "dryRun",
      "type" : "string",
      "uniqueItems" : true
    },
    {
      "description" : "fieldManager is a name associated with the actor or entity that is
      "in" : "query",
      "name" : "fieldManager",
      "type" : "string",
      "uniqueItems" : true
    },
    {
      "description" : "fieldValidation instructs the server on how to handle objects in th
      "in" : "query",
      "name" : "fieldValidation",
      "type" : "string",
      "uniqueItems" : true
    }
  ],
  "produces" : [
    "application/json",
    "application/yaml",
    "application/vnd.kubernetes.protobuf"
  ],
  "responses" : {
    "200" : {
      "description" : "OK",
      "schema" : {
        "$ref" : "#/definitions/io.k8s.api.apps.v1.DaemonSet"
      }
    },
```

OpenAPI 2.0 vs 3.0

- Restructured document so that API definitions are easier to reuse
- Extended JSON Schema support
  - oneOf
  - anyOf
  - default
  - nullable



CRD OpenAPI V3 Structural Schema

Structural Schema (OpenAPI V3)

```
type: object
properties:
        foo: string
        default: "bar"

type: object
properties:
        intstringfield:
                anyOf:
                - type: integer
                - type: string
                x-kubernetes-int-or-string: true
```

Published Schema (OpenAPI V2)

```
type: object
properties:
        foo: string


type: object
properties:
        intstringfield:
                x-kubernetes-int-or-string: true
```

CRD OpenAPI V3 Structural Schema

Structural Schema (OpenAPI V3)

```
type: object
properties:
        fieldwithvalidation:
                type: integer
                anyOf:
                - minimum: 5
                  maximum: 10
                - minimum: 15
                  maximum: 20
```

```
type: object
properties:
        foo:
                type: object
                nullable: true
                properties:
                        a:
                                type: string
                        b:
                                type: integer
```

Published Schema (OpenAPI V2)

```
type: object
properties:
        fieldwithvalidation:
                type: integer
```

```
type: object
properties:
        foo:
                type: object
```

OpenAPI Endpoint

discoveryClient.OpenAPISchema()

/openapi/v2

OpenAPI V2 schema for all paths

discoveryClient.OpenAPIV3()

/openapi/v3

/openapi/v3/apis/apps/v1

OpenAPI V3 schema for only apps/v1 endpoints

/openapi/v3/autoscaling/v1

OpenAPI V3 schema for only autoscaling/v1 endpoints

OpenAPI Endpoint

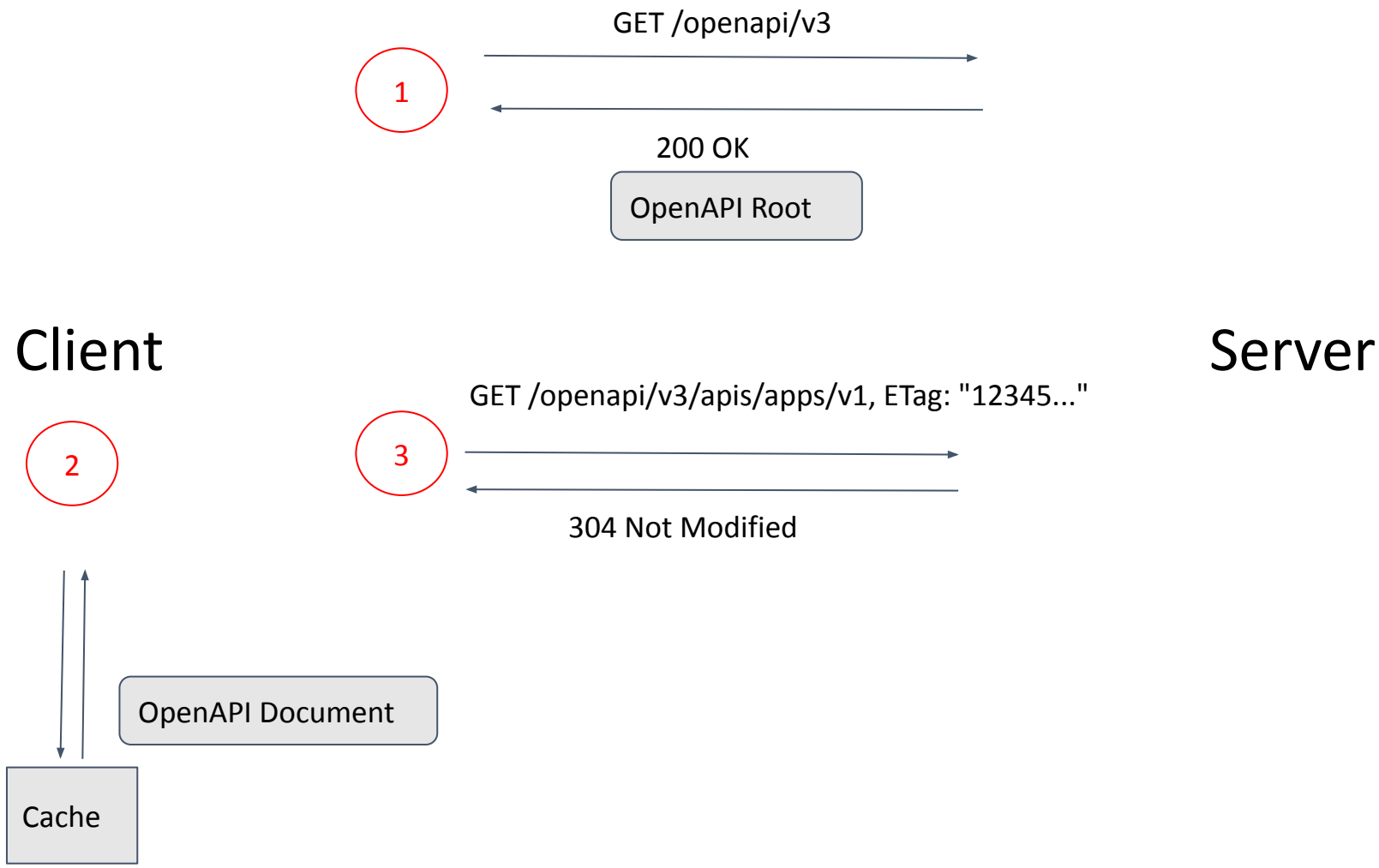
Size	• An aggregated OpenAPI V2 is on the magnitude of megabytes, increasing based on complexity and number of CRDs and aggregated apiservers • OpenAPI V3 publishes separate specifications for each group version
Incremental Update	• With OpenAPI V2, an update to a single resource will cascade into a recomputation of the entire schema. Leads to CPU and memory spikes • OpenAPI V3 only updates the OpenAPI for the corresponding group version
Complexity of Aggregation	• OpenAPI V2's aggregator will send a request to all aggregated apiservers periodically, download and merge the OpenAPI • OpenAPI V3's aggregator acts as a proxy

Cache Busting

/openapi/v3

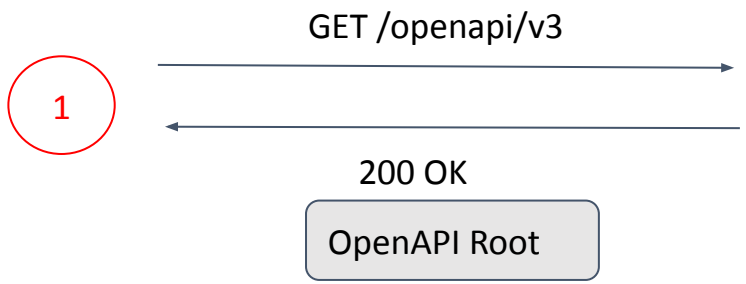

```
{
  "paths" : {
    "api/v1" : {
      "serverRelativeURL" : "/openapi/v3/api/v1?hash=90F7D..."
    },
    "apis/admissionregistration.k8s.io/v1" : {
      "serverRelativeURL" : "/openapi/v3/apis/admissionregistration.k8s.io/v1?hash=00C7F..."
    },
    "apis/apiextensions.k8s.io/v1" : {
      "serverRelativeURL" : "/openapi/v3/apis/apiextensions.k8s.io/v1?hash=7B58B..."
    },
    "apis/apps/v1" : {
      "serverRelativeURL" : "/openapi/v3/apis/apps/v1?hash=06F46..."
    },
    "apis/authentication.k8s.io/v1" : {
      "serverRelativeURL" : "/openapi/v3/apis/authentication.k8s.io/v1?hash=AF778..."
    },
    ...
}
```

OpenAPI with only ETags



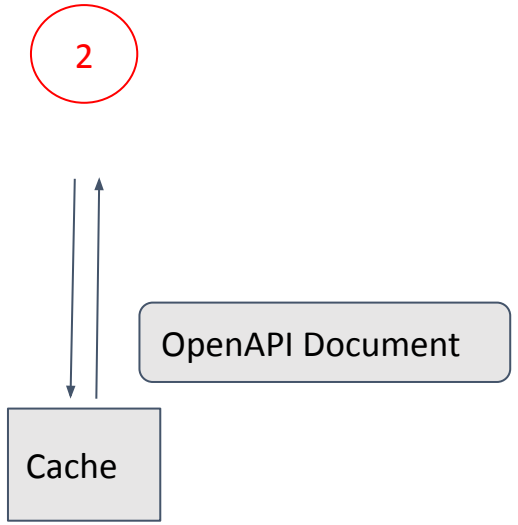
GET /openapi/v3/apis/apps/v1, ETag: "12345..."

OpenAPI with Cache Busting



Client

Server



Headers

Cache Control: immutable
Expires: 1 year

GET /openapi/v3/apis/apps/v1?hash=12345..., ETag: "12345..."

Future Work

- KEP 2896: OpenAPI V3 to GA
- KEP 3515: kubectl explain templating and OpenAPI V3 upgrade

BUILDING FOR THE ROAD AHEAD

DETROIT 2022

The Power and Danger of Aggregated APIServers

David Eads

Agenda

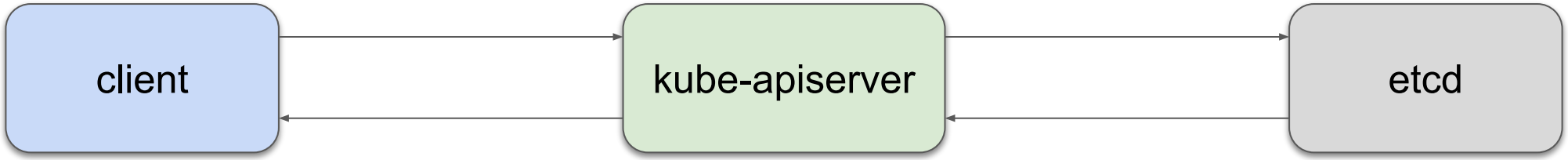
- **What is API Aggregation?**
  - Older, less known cousin of CRDs
  - How does it work?
  - How is access secured?
  - Where does authorization happen?
- **What cool things does API Aggregation allow?**
  - Binary storage format
  - No storage: Metrics server
  - Multiple implementations/Alternative storage: Prometheus-adapter
- **What bad things can happen?**
  - Inconsistent availability from HA masters
  - RESTMapping failures
    - impact on admission
    - Impact on garbage collection
    - Impact on namespace cleanup
  - Namespace cleanup cycles
- **Limitations**
  - Cannot stack behind CRDs

Agenda

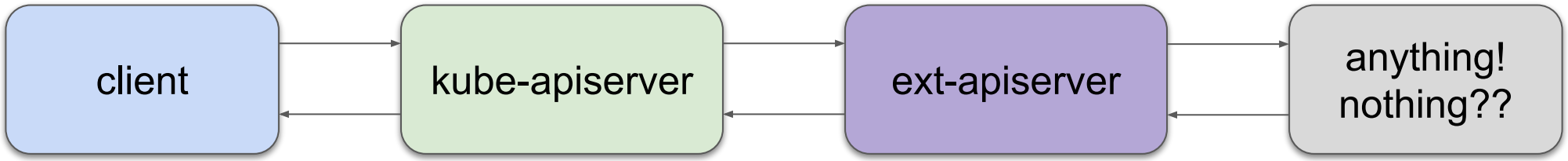
- **What is API Aggregation?**
  - **Older, less known cousin of CRDs**
  - **How does it work?**
  - **How is access secured?**
  - **Where does authorization happen?**
- What cool things does API Aggregation allow?
  - Binary storage format
  - No storage: Metrics server
  - Multiple implementations/Alternative storage: Prometheus-adapter
- What bad things can happen?
  - Inconsistent availability from HA masters
  - RESTMapping failures
    - impact on admission
    - Impact on garbage collection
    - Impact on namespace cleanup
  - Namespace cleanup cycles
- Limitations
  - Cannot stack behind CRDs

Overview

CRD Flow

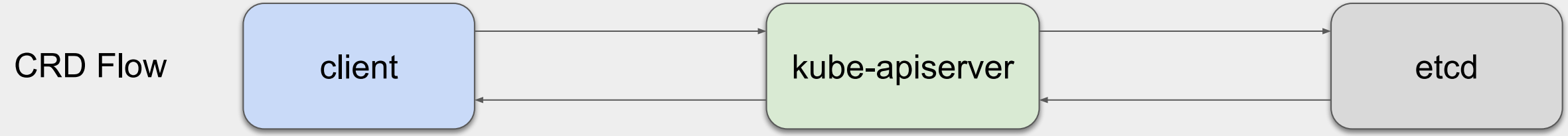


Aggregated API Server Flow

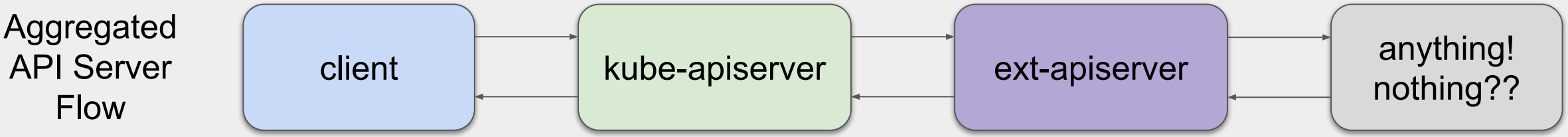


Overview

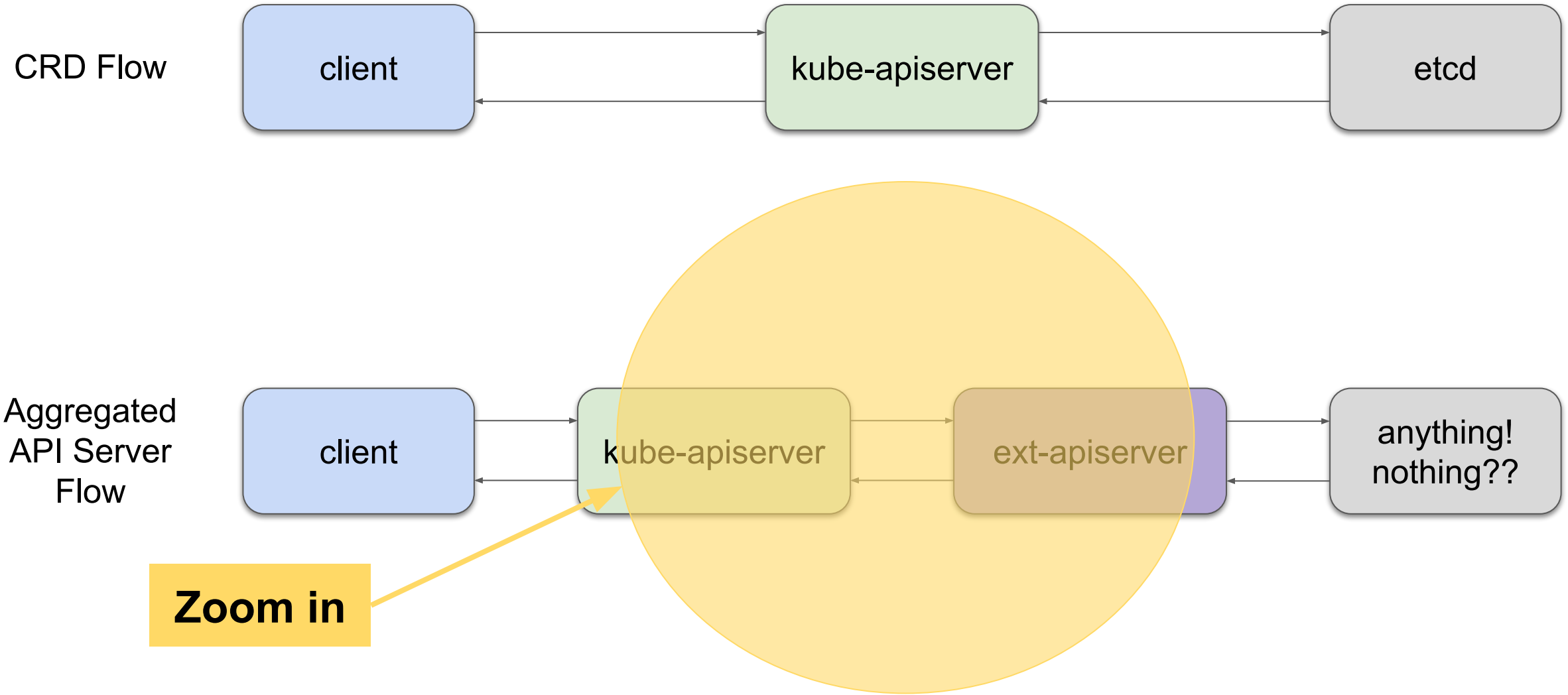
Constrained, but safe



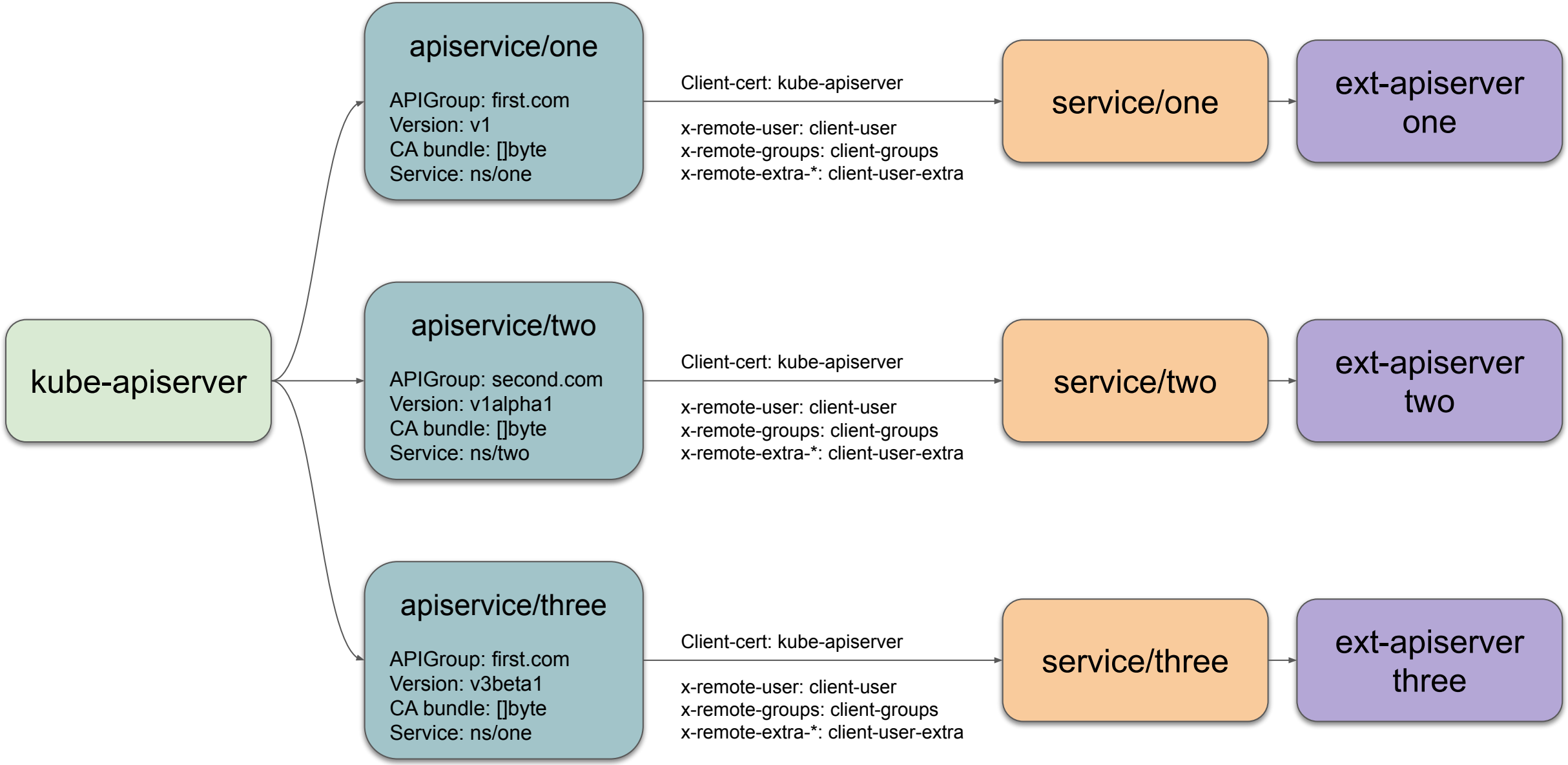
Freedom, with risk



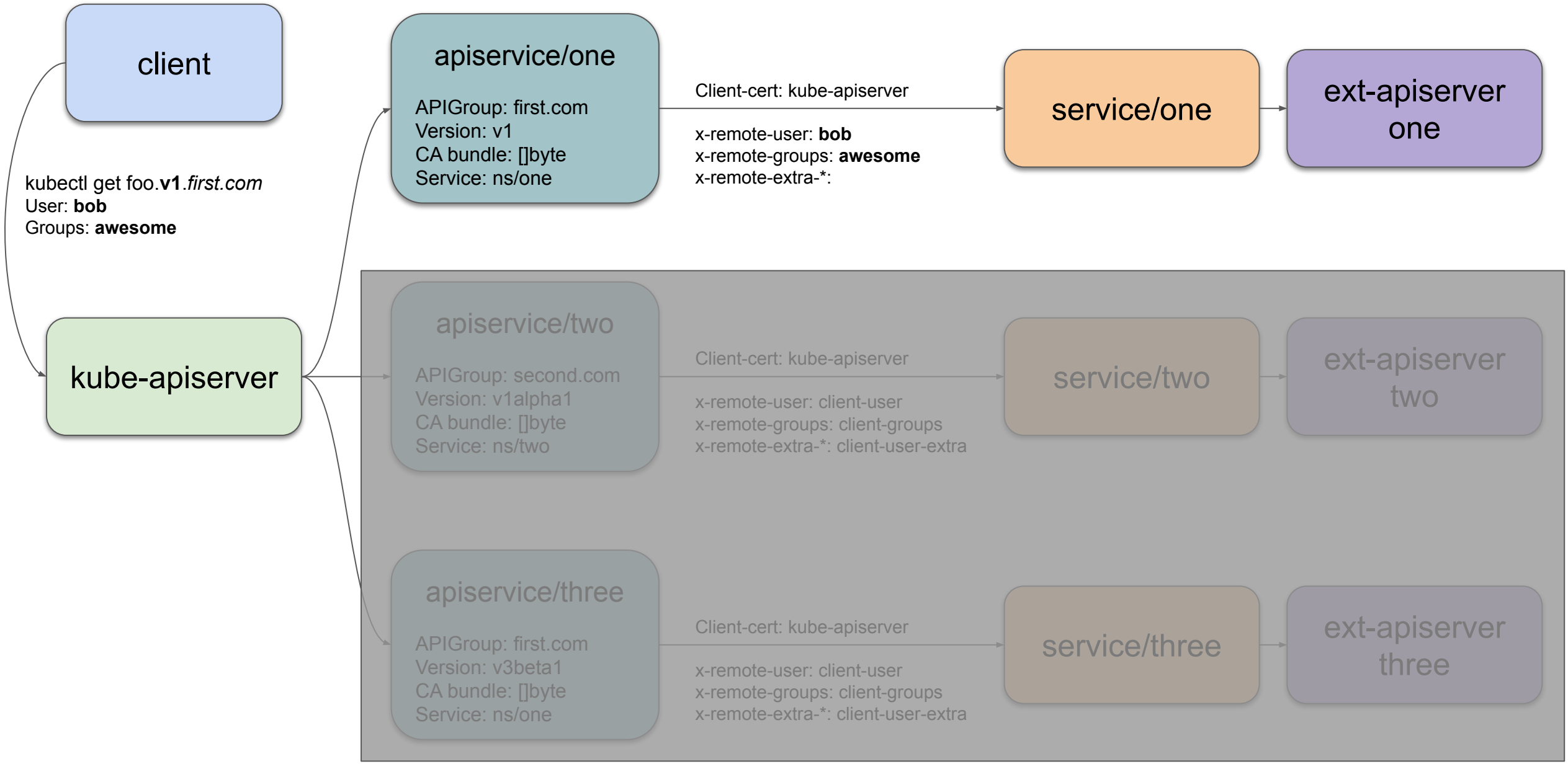
Overview


CRD Flow
client
kube-apiserver
etcd
Aggregated API Server Flow
client
kube-apiserver
ext-apiserver
anything! nothing??
Zoom in

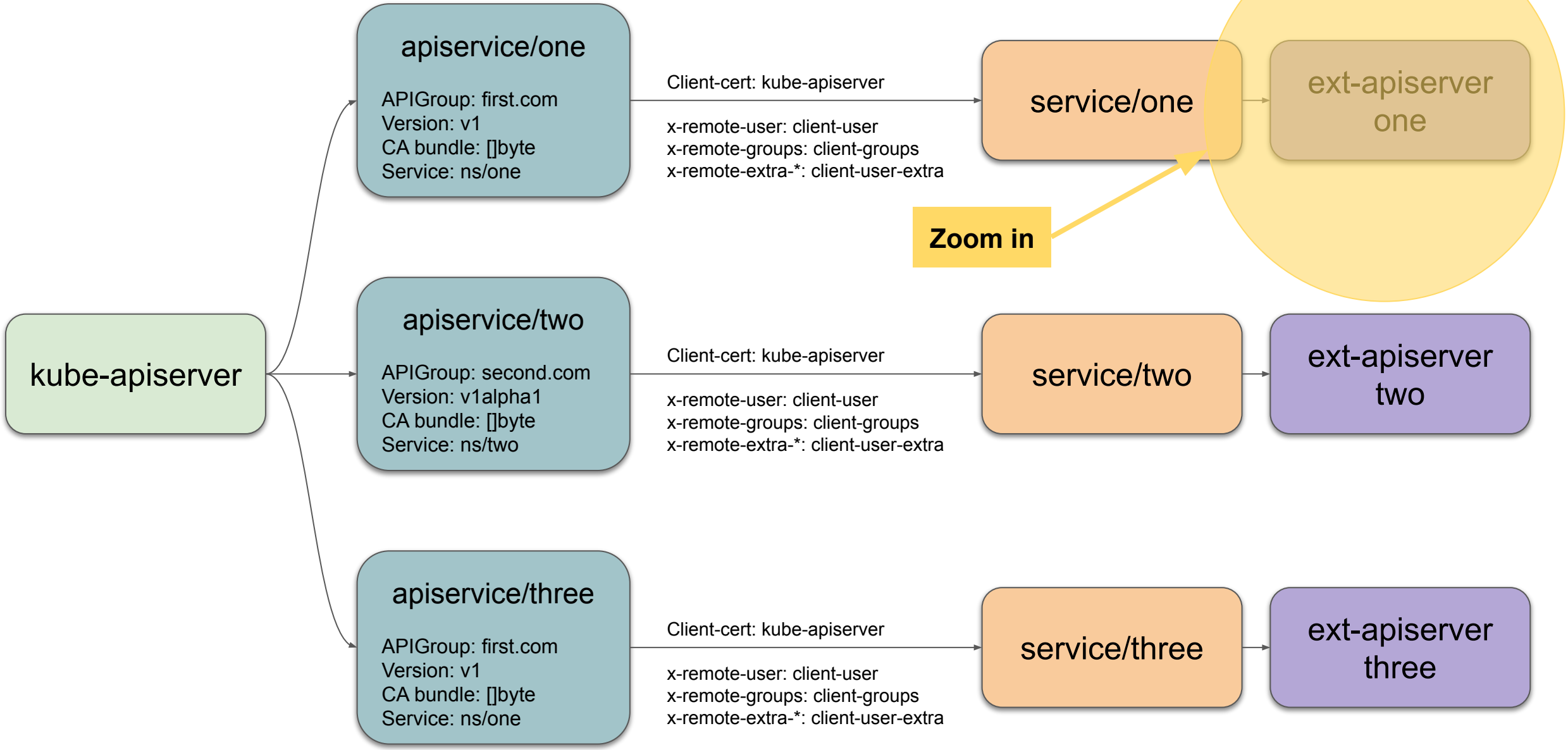
kube-apiserver routing



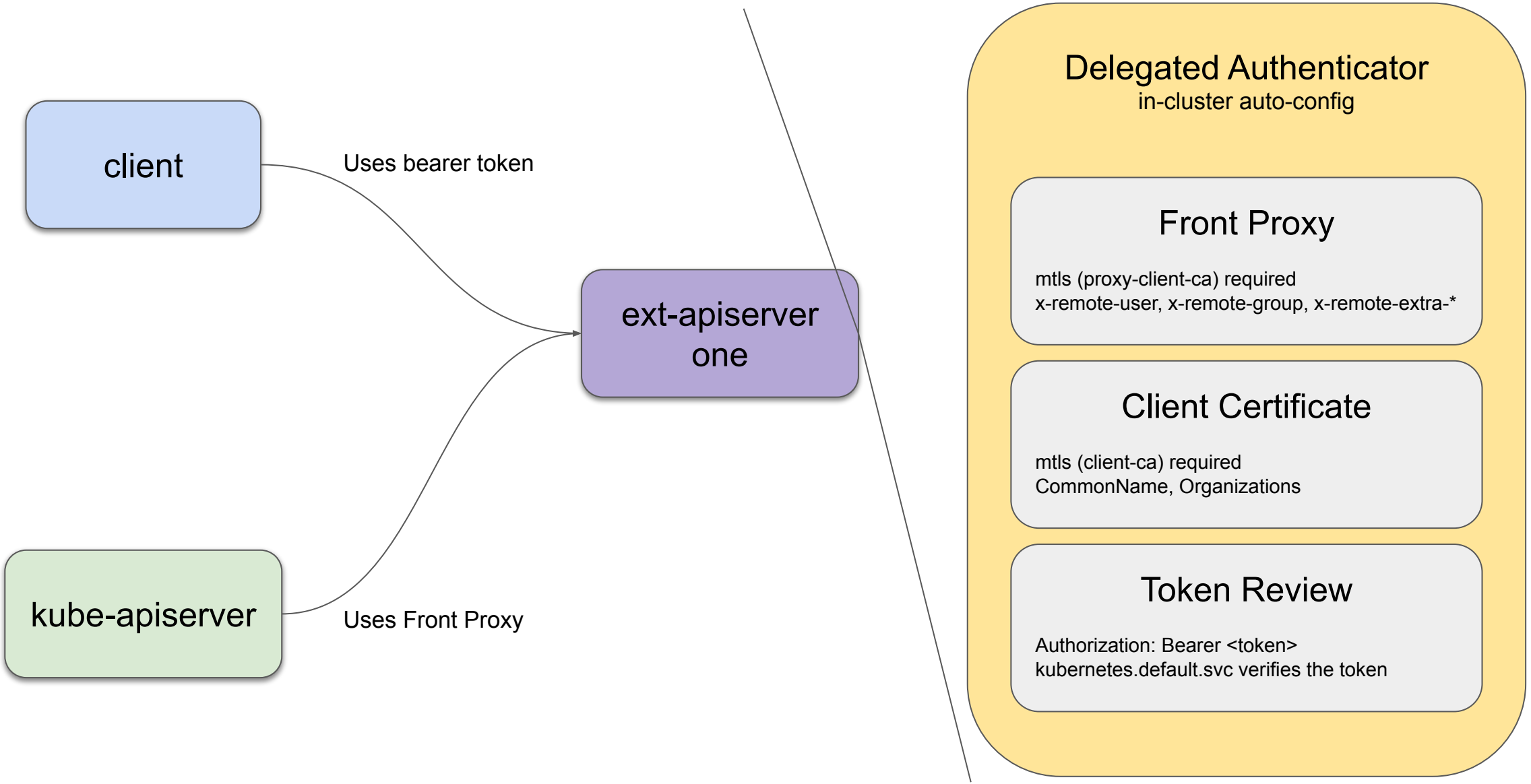
kube-apiserver routing example 1



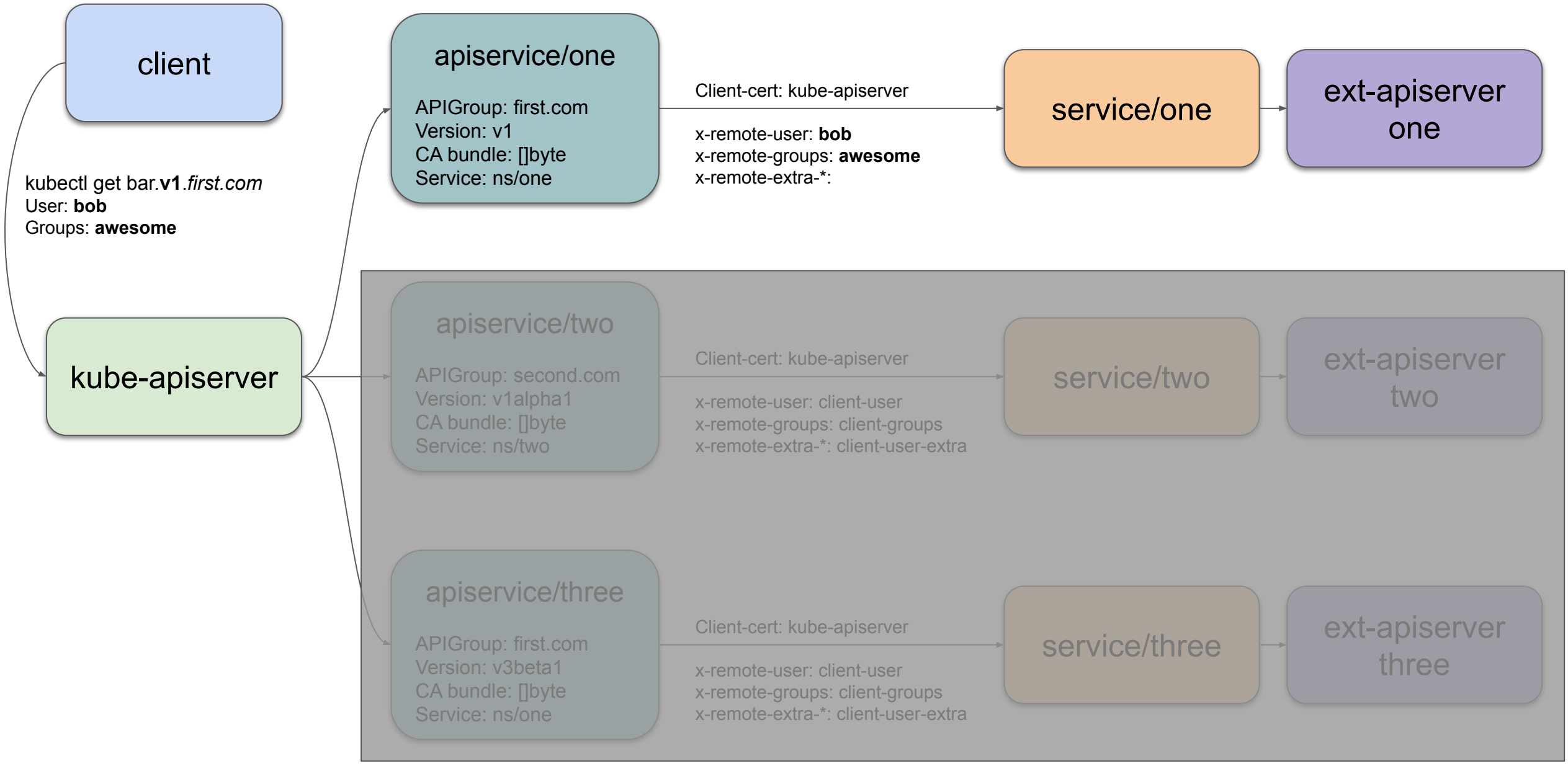
kube-apiserver routing



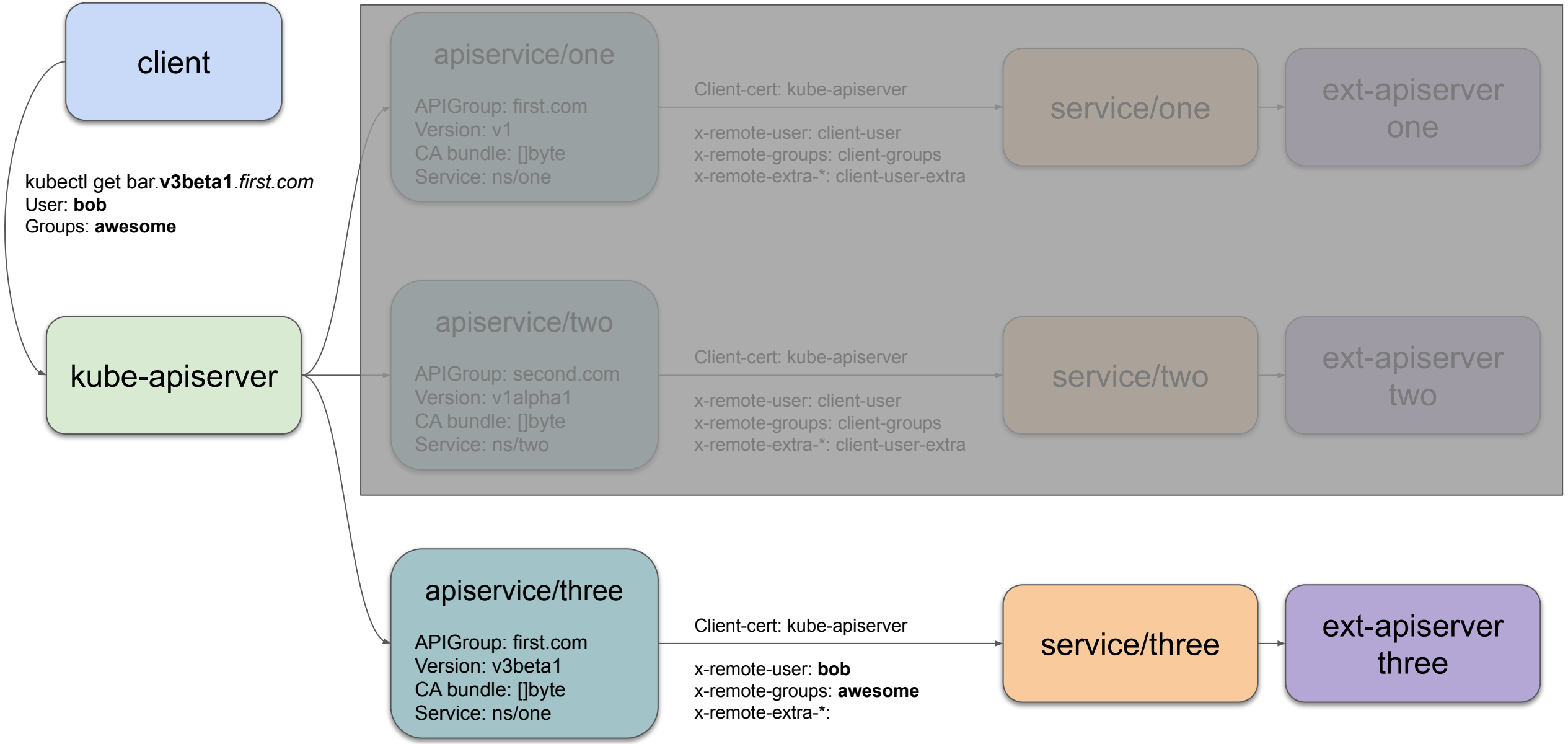
ext-apiserver authentication



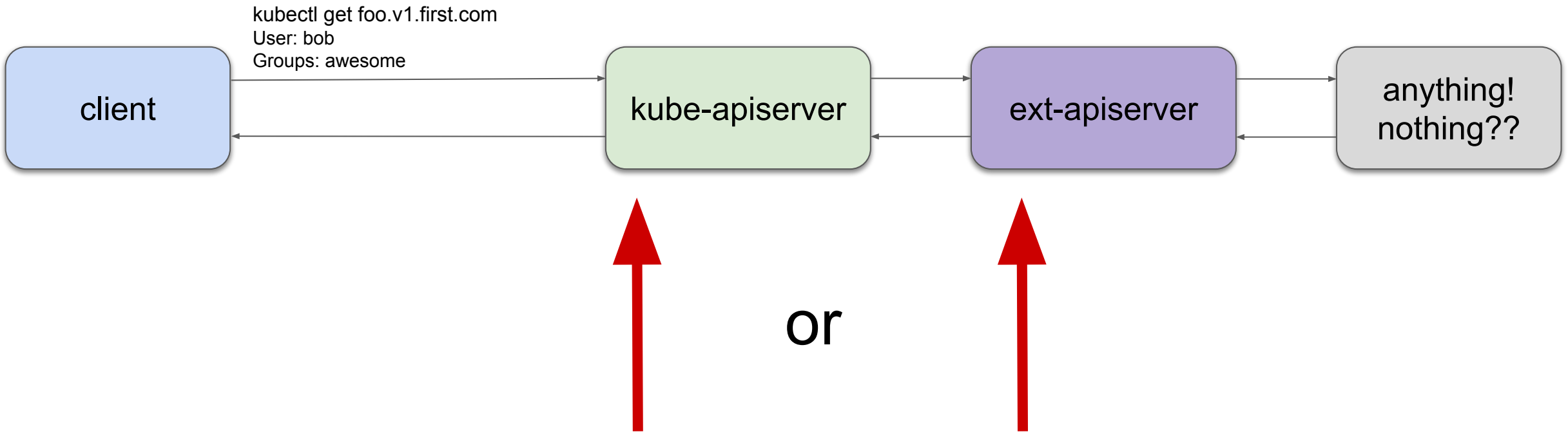
kube-apiserver routing example 2



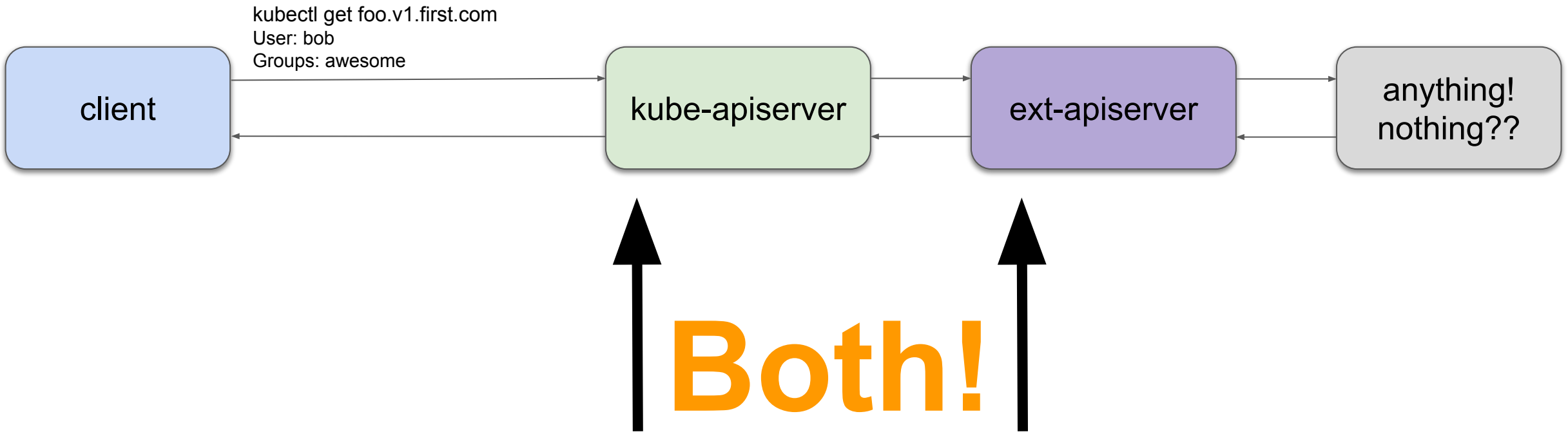
kube-apiserver routing example 3



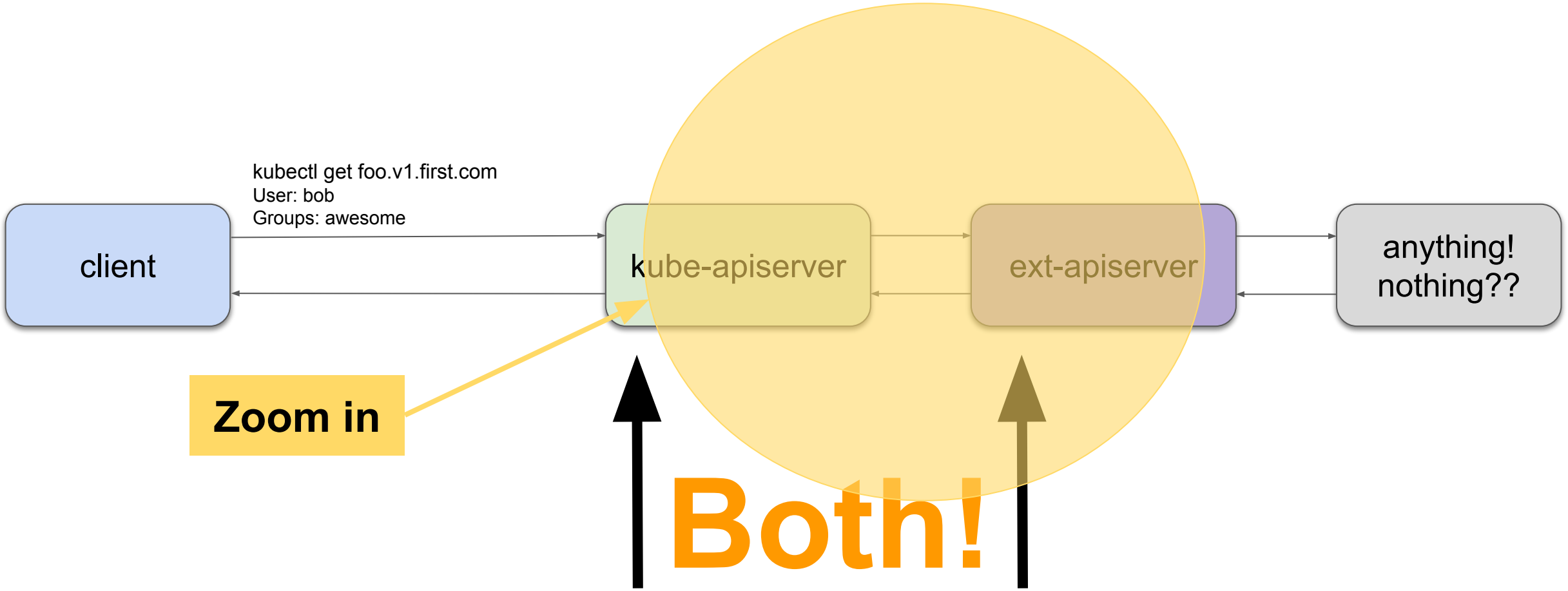
Authorization?



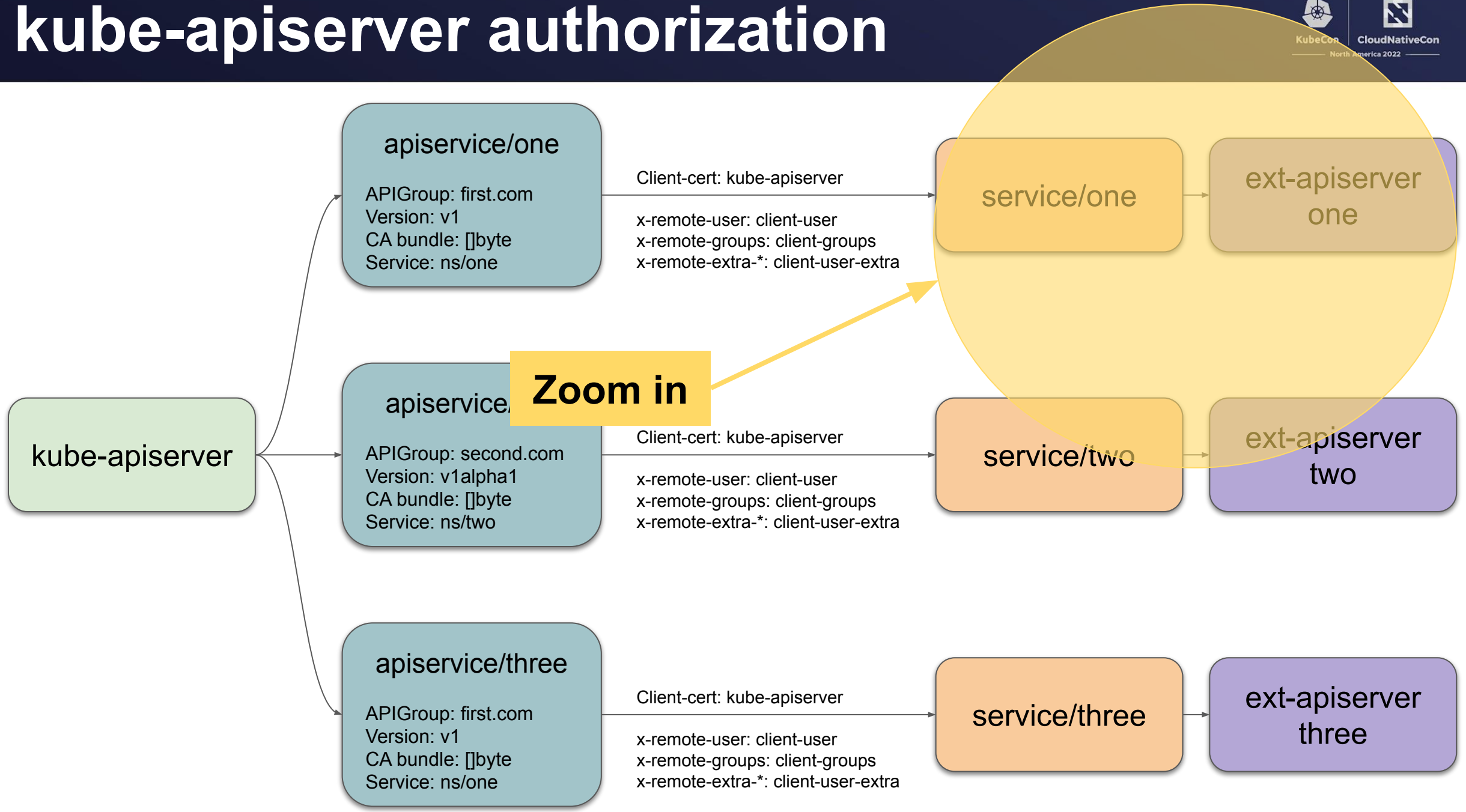
Authorization?



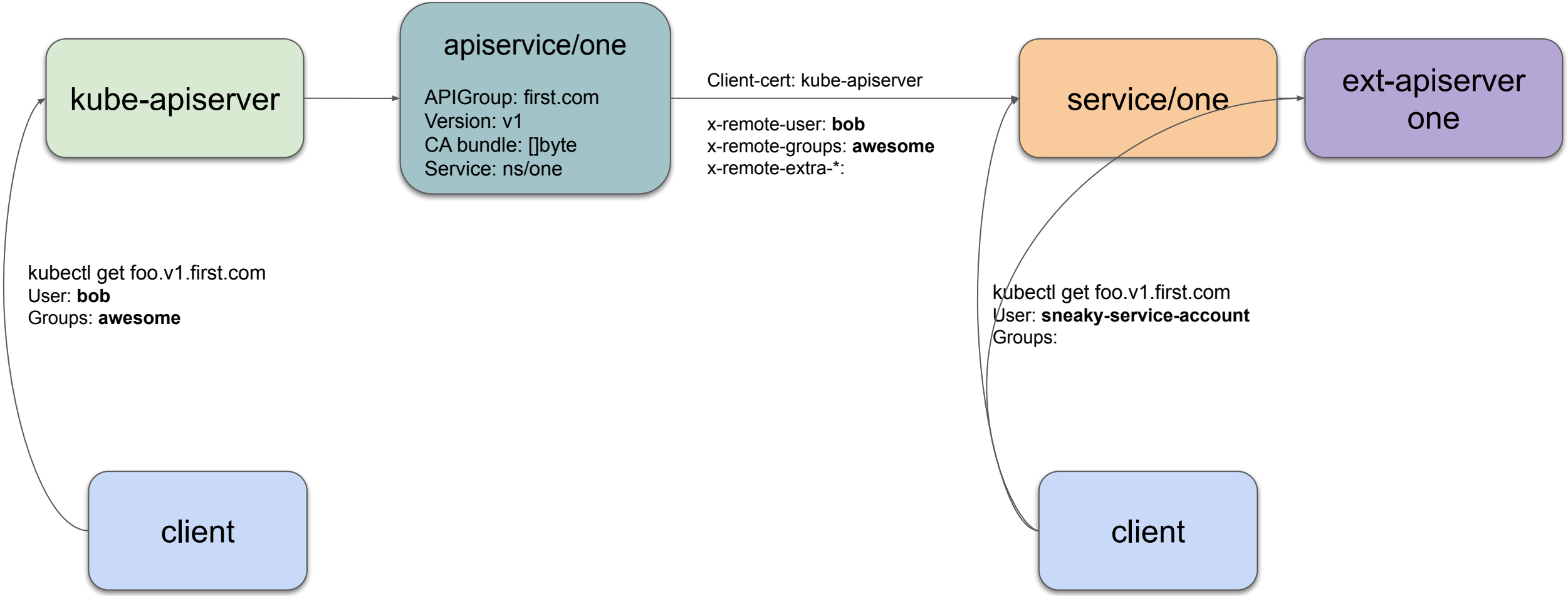
Authorization?



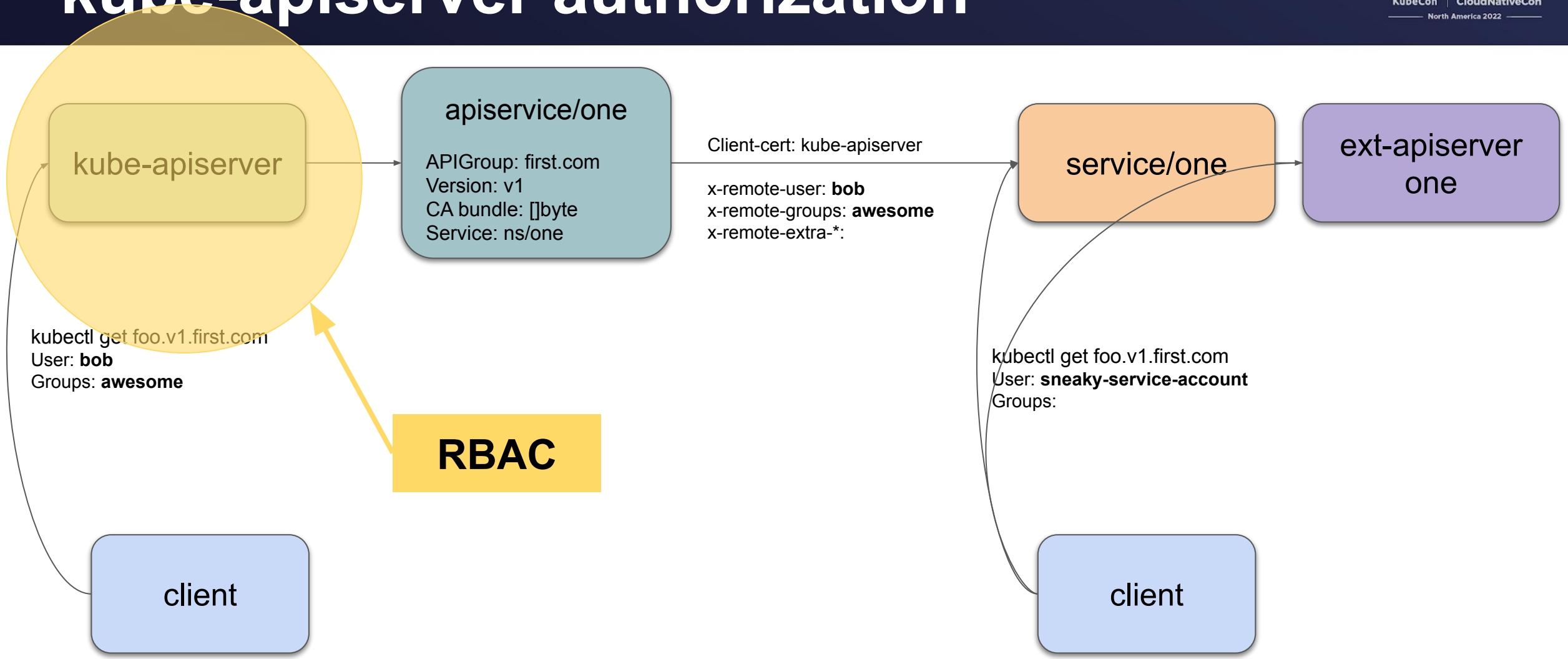
kube-apiserver authorization



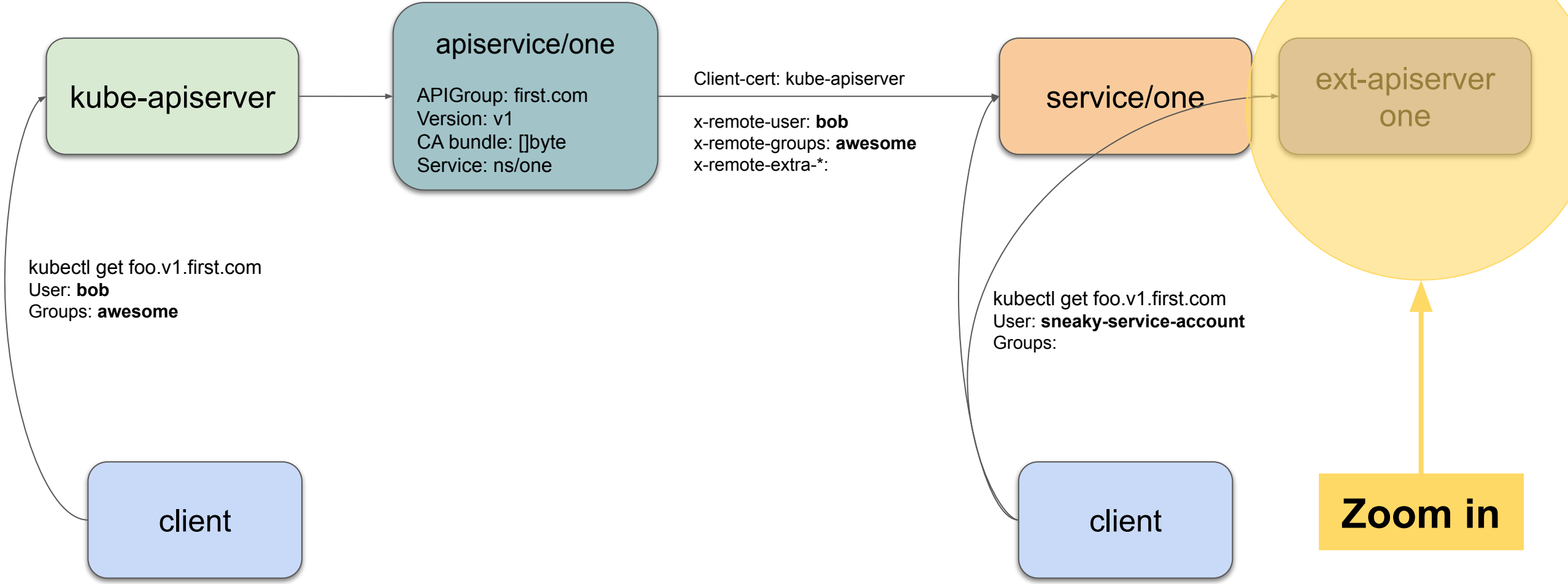
kube-apiserver authorization



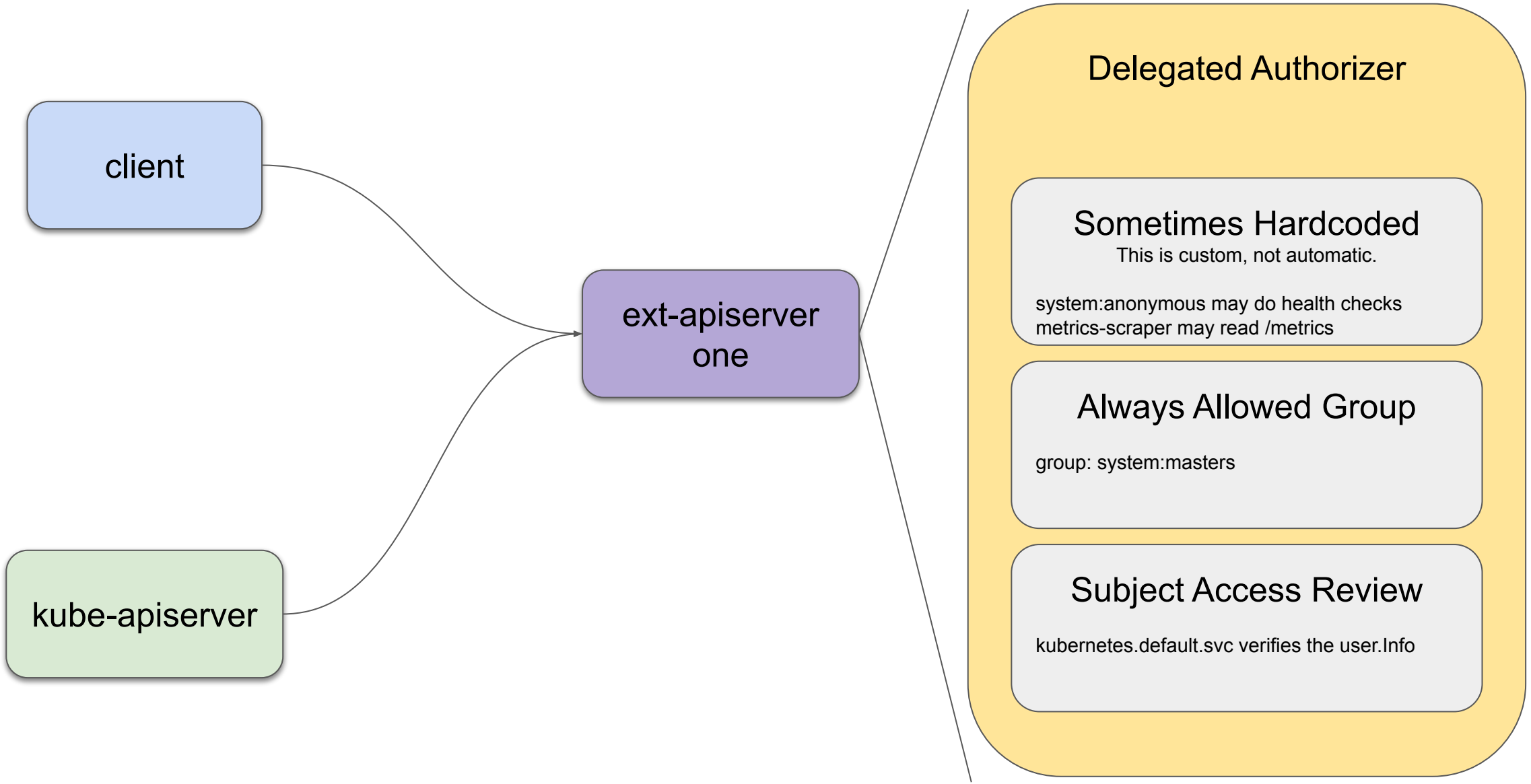
kube-apiserver authorization



ext-apiserver authorization



kube-apiserver routing

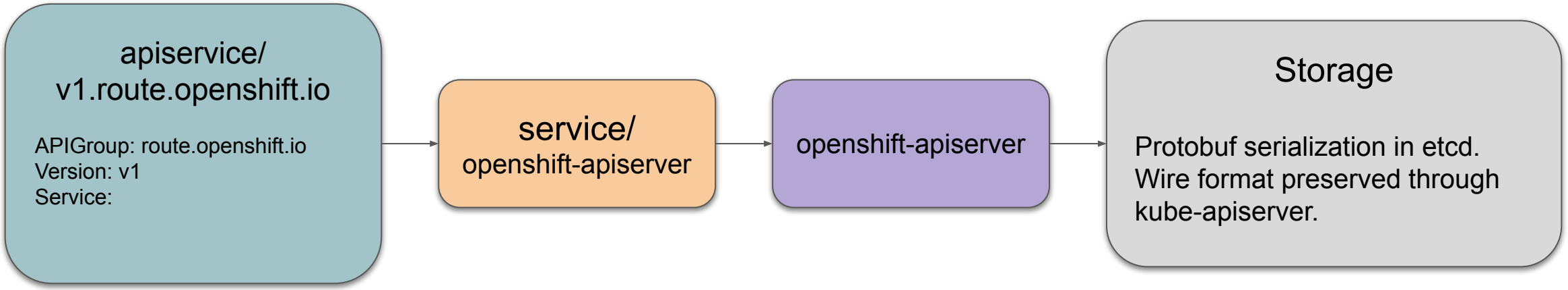


Agenda

- What is API Aggregation?
  - Older, less known cousin of CRDs
  - How does it work?
  - How is access secured?
  - Where does authorization happen?
- **What cool things does API Aggregation allow?**
  - **Binary storage format**
  - **No storage: Metrics server**
  - **Multiple implementations/Alternative storage: Prometheus-adapter**
- What bad things can happen?
  - Inconsistent availability from HA masters
  - RESTMapping failures
    - impact on admission
    - Impact on garbage collection
    - Impact on namespace cleanup
  - Namespace cleanup cycles
- Limitations
  - Cannot stack behind CRDs

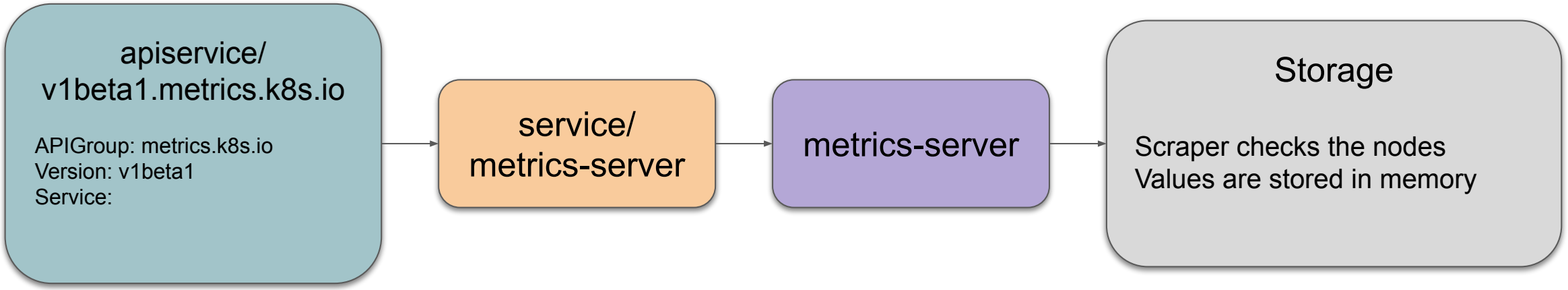
Binary storage: openshift-apiserver

github.com/openshift/openshift-apiserver



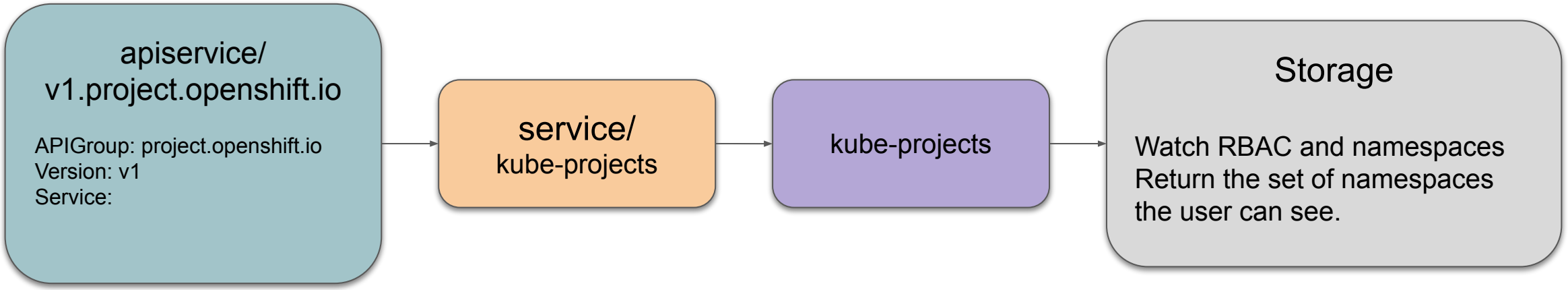
No storage: metrics server

github.com/kubernetes-sigs/metrics-server



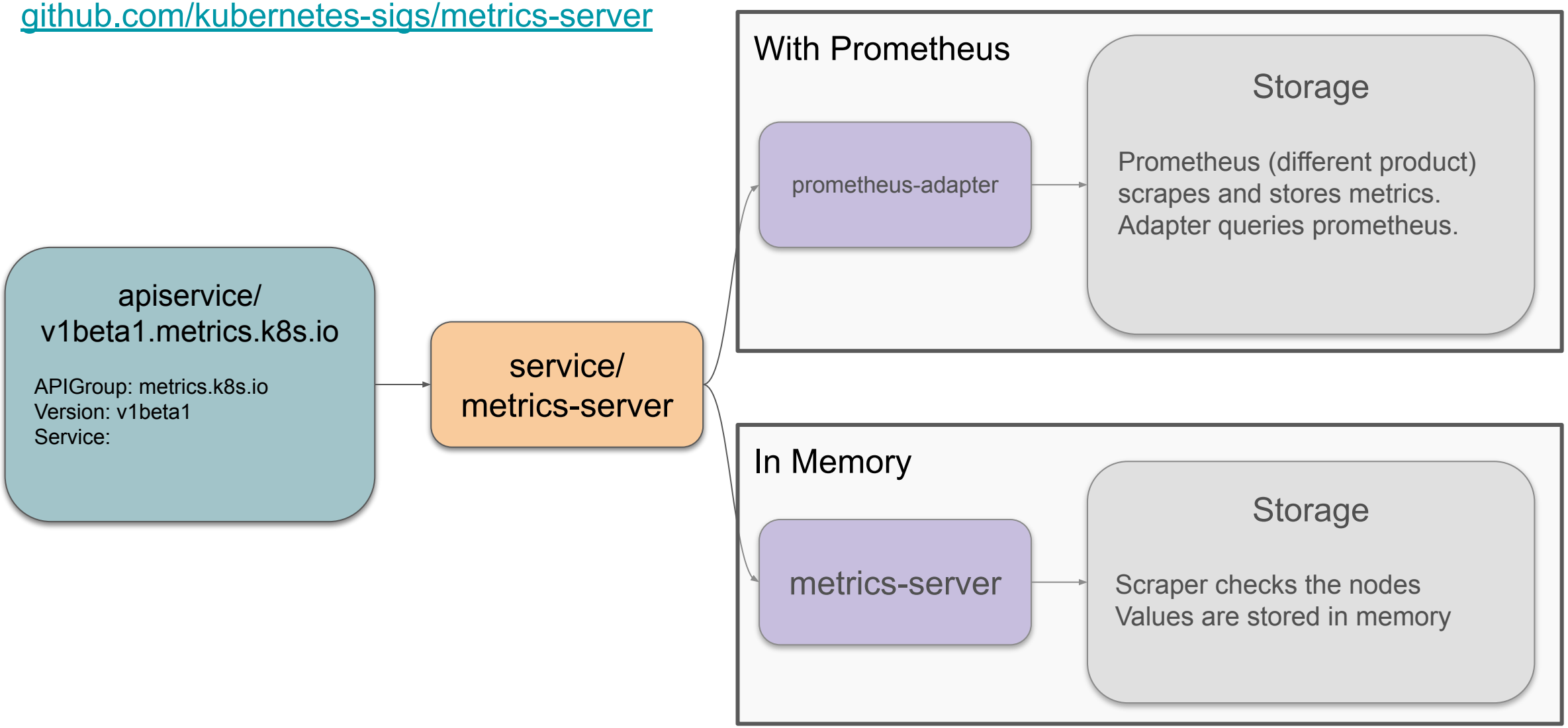
No storage: kube-projects

github.com/openshift/kube-projects (created as a POC)



Multiple implementations: metrics

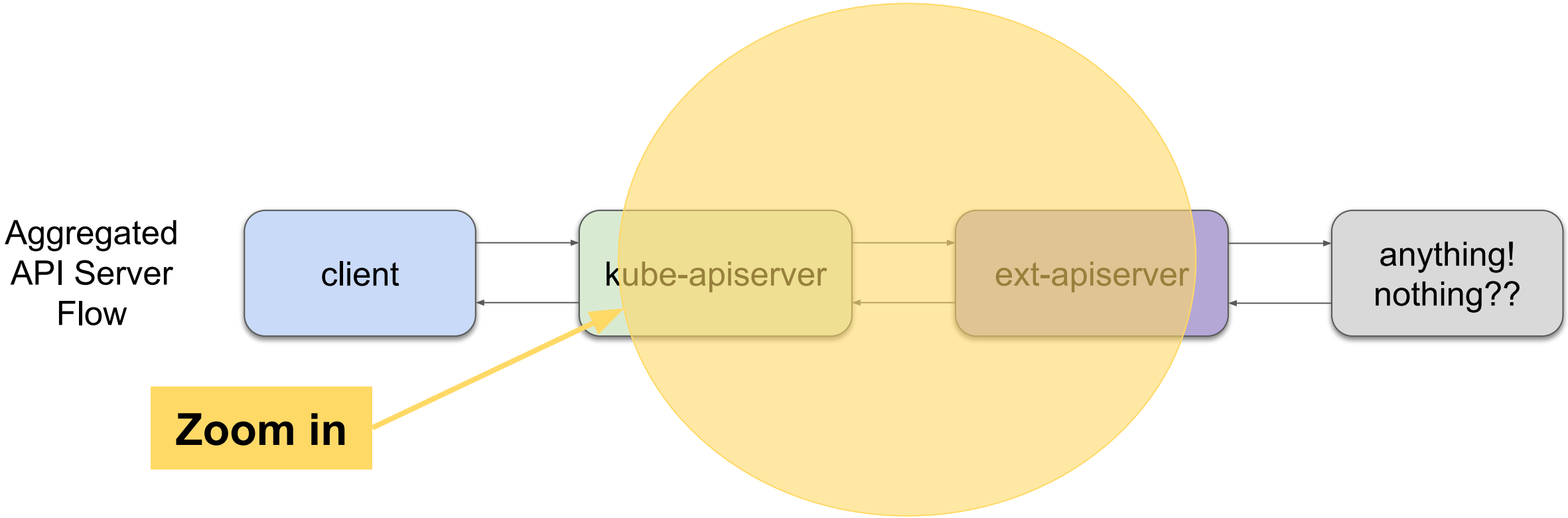
github.com/kubernetes-sigs/prometheus-adapter
github.com/kubernetes-sigs/metrics-server



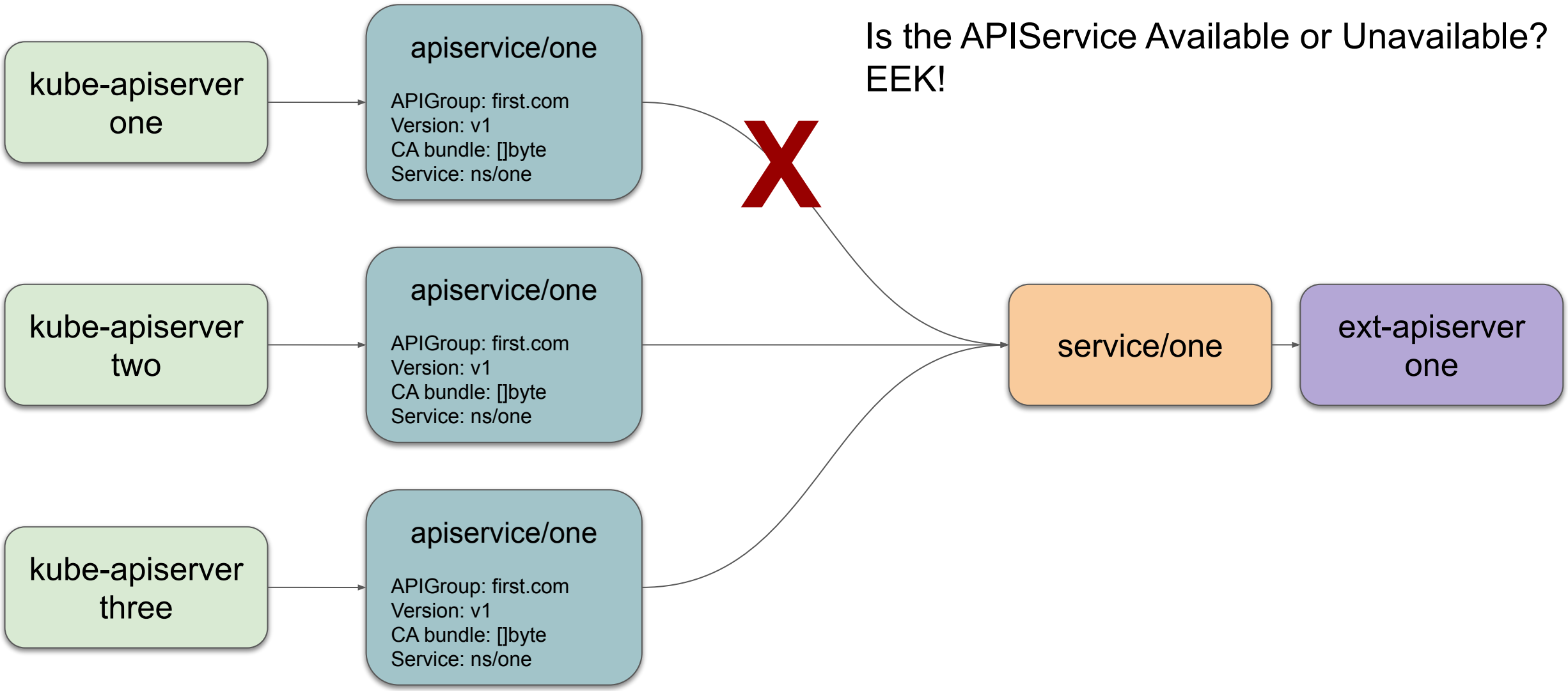
Agenda

- What is API Aggregation?
  - Older, less known cousin of CRDs
  - How does it work?
  - How is access secured?
  - Where does authorization happen?
- What cool things does API Aggregation allow?
  - Binary storage format
  - No storage: Metrics server
  - Multiple implementations/Alternative storage: Prometheus-adapter
- **What bad things can happen?**
  - **Inconsistent availability from HA masters**
  - **RESTMapping failures**
    - **impact on admission**
    - **Impact on garbage collection**
    - **Impact on namespace cleanup**
  - **Namespace cleanup cycles**
- Limitations
  - Cannot stack behind CRDs

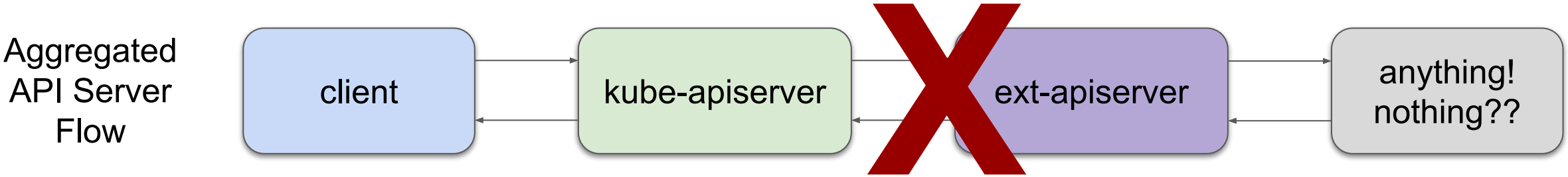
HA kube-apiservers



HA kube-apiservers, with disruption

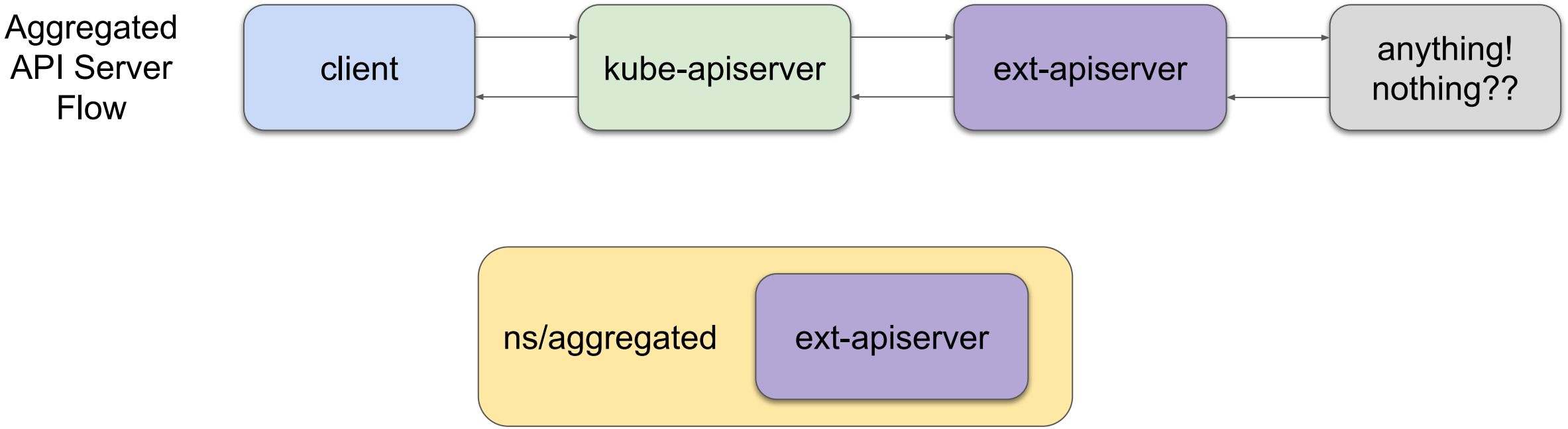


RESTMapping Failure

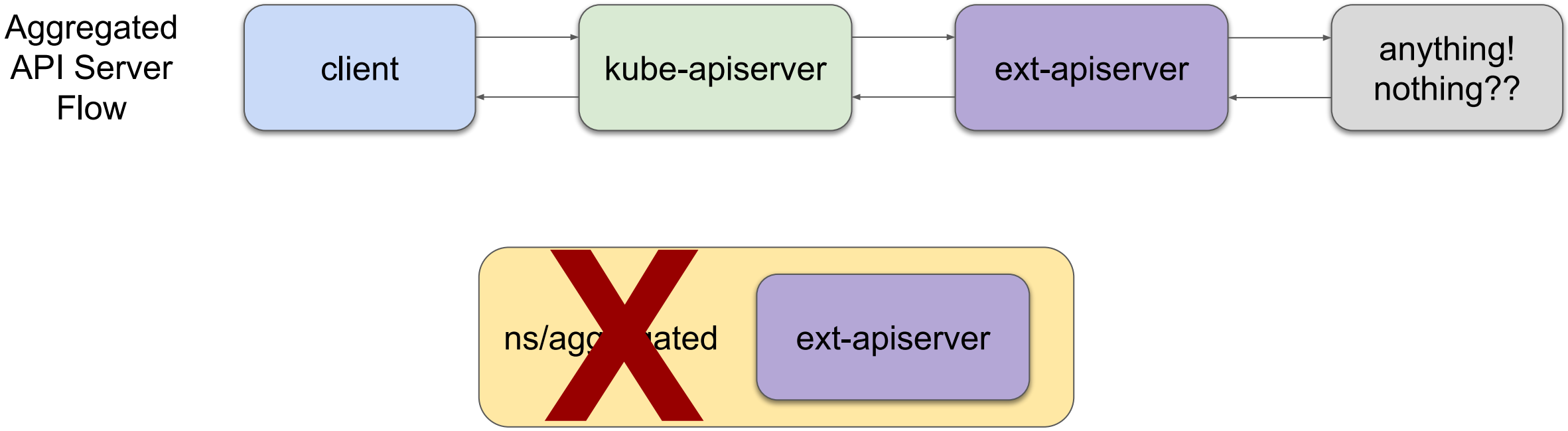


- This causes API discovery (which APIs are available) to fail!
- Discovery is used to know what resource a Kind matches.
  - Kinds are the serializations like Pod, resources are the URLs like api/v1/pods.
  - Same Kind can match multiple resources, see Scale.autoscaling
- kubectl get/create/apply foo.aggregated-group will fail
- OwnerReferences are Kinds
  - No discovery means GC does not know how to build the graph
  - Admission protection from blocking owner deletion will fail
- Namespace cleanup cannot get the full list of API resources to remove
  - Namespaces are stuck finalizing and won't delete

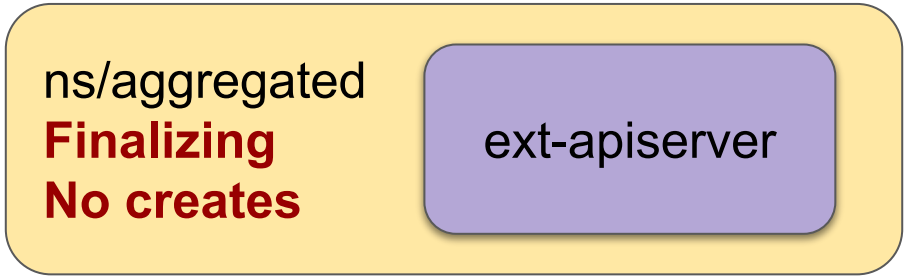
Namespace deletion cycle



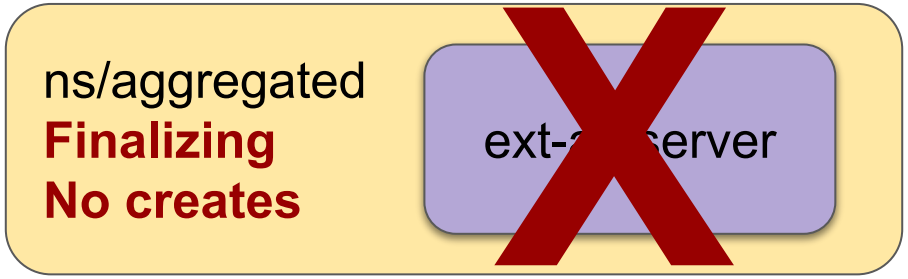
Namespace deletion cycle



Namespace deletion cycle



Namespace deletion cycle



Namespace deletion cycle

ns/aggregated
Finalizing
No creates

- Namespace lifecycle controller gets stuck because it cannot determine the list of resources to delete in namespace/aggregated
- The controller cannot progress without the ext-apiserver, the ext-apiserver cannot start until the controller is finished.
- If all other resources have been cleaned (see namespace/status) and your know for sure that it should be recreated, the finalizer can be manually cleared to allow progress
  - OpenShift operators do this when certain managed namespaces are deleted

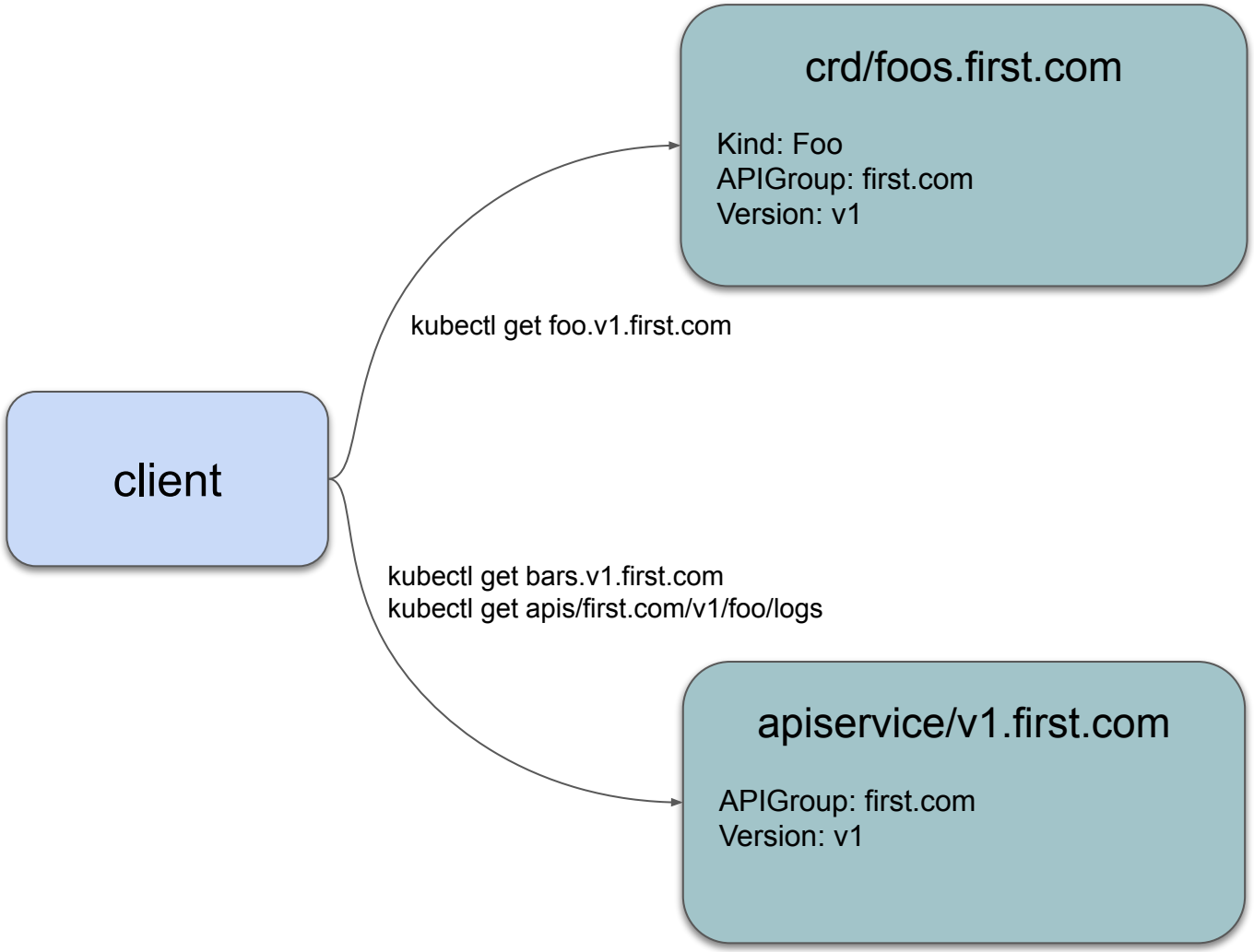
Agenda

- What is API Aggregation?
  - Older, less known cousin of CRDs
  - How does it work?
  - How is access secured?
  - Where does authorization happen?
- What cool things does API Aggregation allow?
  - Binary storage format
  - No storage: Metrics server
  - Multiple implementations/Alternative storage: Prometheus-adapter
- What bad things can happen?
  - Inconsistent availability from HA masters
  - RESTMapping failures
    - impact on admission
    - Impact on garbage collection
    - Impact on namespace cleanup
  - Namespace cleanup cycles
- **Limitations**
  - **Cannot stack behind CRDs**

CRD + APIService? No.

- **This does NOT work**
- Attempted for subresources
- Attempted for some special and some not-special types in the same group.

- Maybe someday?



Back to Fede

KubeCon CloudNativeCon North America 2022
BUILDING FOR THE ROAD AHEAD
DETROIT 2022

SIG API Machinery advanced topics

David Eads, Red Hat (deads2k@)
Jeffrey Ying, Google (jefftree@)

Host: Federico Bongiovanni, Google (fedebongio@)

Meetings and Working Groups

Regular SIG meetings:

- SIG Meeting: 60 min / every 2 weeks (11 am PST, Wednesday)
- PR and Bug triage: 30 min / twice every week *1pm PST Tuesday/9:30am PST Thursday*

(join the mailing list to get the invites!)

Regular Working Group meetings:

- Working Group API Expression: 60 min / every 2 weeks
- Working Group Kubebuilder and SDK: 60 min / monthly meeting

Where to find us?

- Mail Group: https://groups.google.com/forum/#!forum/kubernetes-sig-api-machinery
- Slack channel: https://kubernetes.slack.com/messages/sig-api-machinery

Presenters today

- @deads2k (Co-Chair and Tech Lead, Red Hat)
- @jefftree (Kubernetes Contributor, Google)
- @fedebongio (Co-Chair, Google)

Useful links

- Home page: https://github.com/kubernetes/community/tree/master/sig-api-machinery
- SIG Charter: https://github.com/kubernetes/community/blob/master/sig-api-machinery/charter.md
- Youtube Playlist: https://www.youtube.com/playlist?list=PL69nYSiGNLP21oW3hbLyjjj4XhrwKxH2R

BUILDING FOR THE ROAD AHEAD

DETROIT 2022

SIG API Machinery advanced topics

Thank you!