
KubeCon
CloudNativeCon
North America 2023

KubeCon | CloudNativeCon

North America 2023

SIG API Machinery

Leila Jalali, Google
Stefan Schimanski, Upbound

Where are we on the journey?

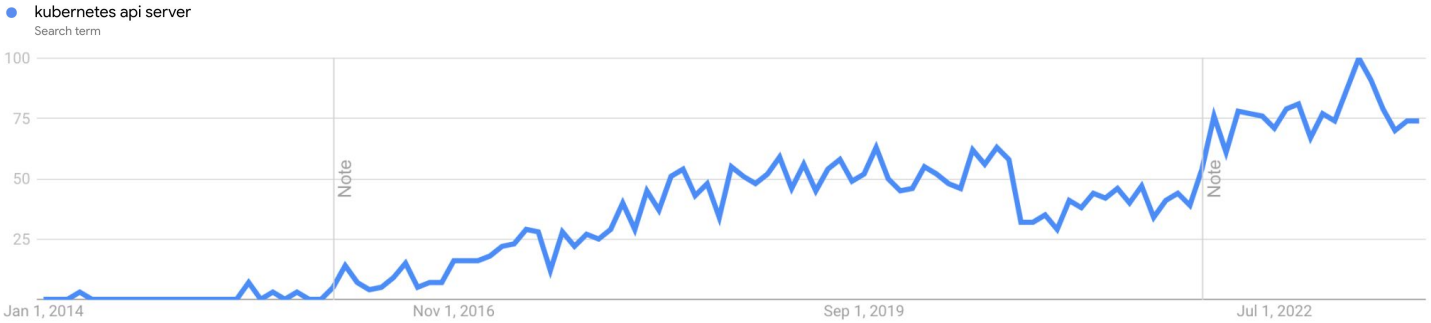
2014
Kubernetes introduced

2015
Kubernetes v1.0
Birth of CNCF

2018
K8s graduated CNCF

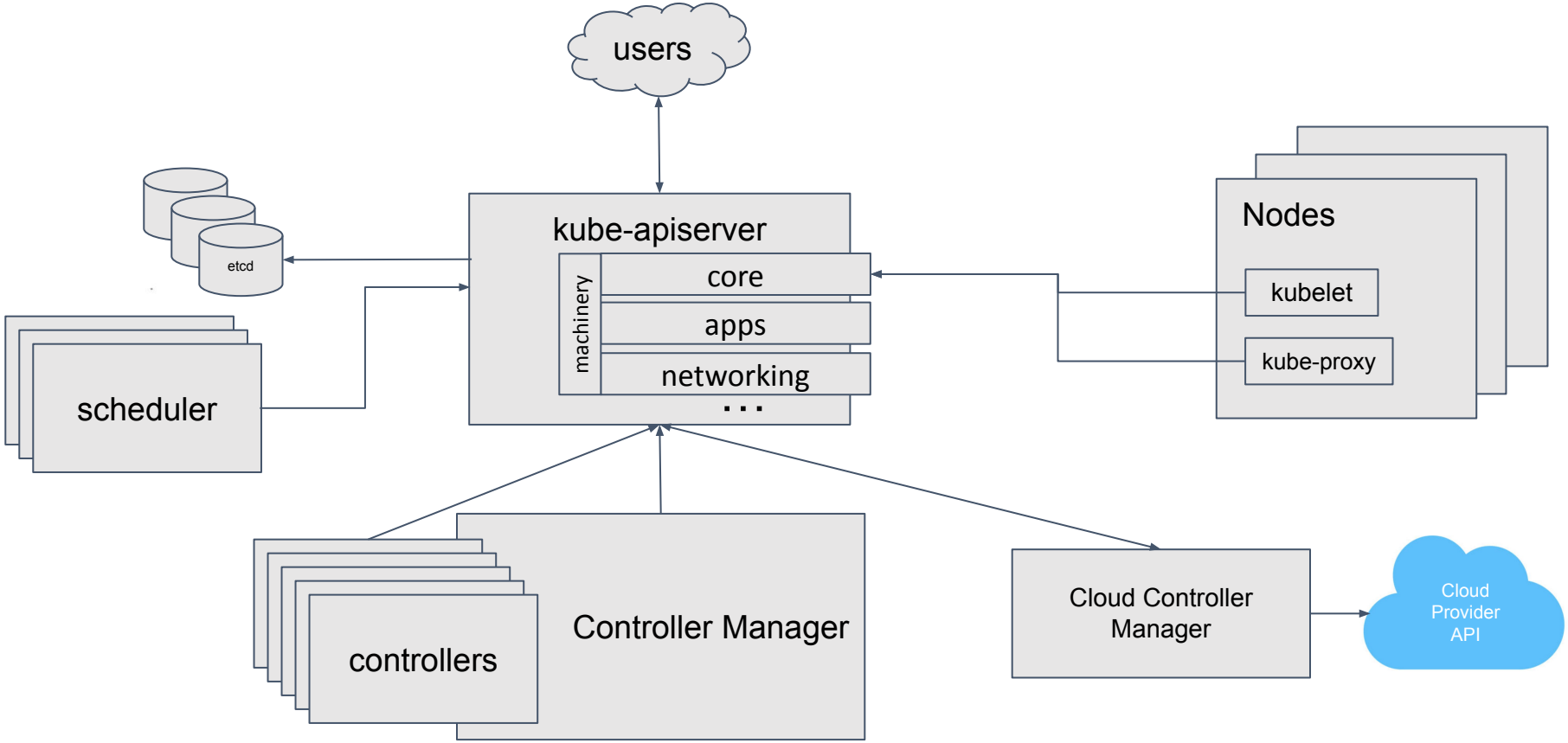
2022
multi-cloud and hybrid cloud

2023
96% of organizations are either using or evaluating Kubernetes

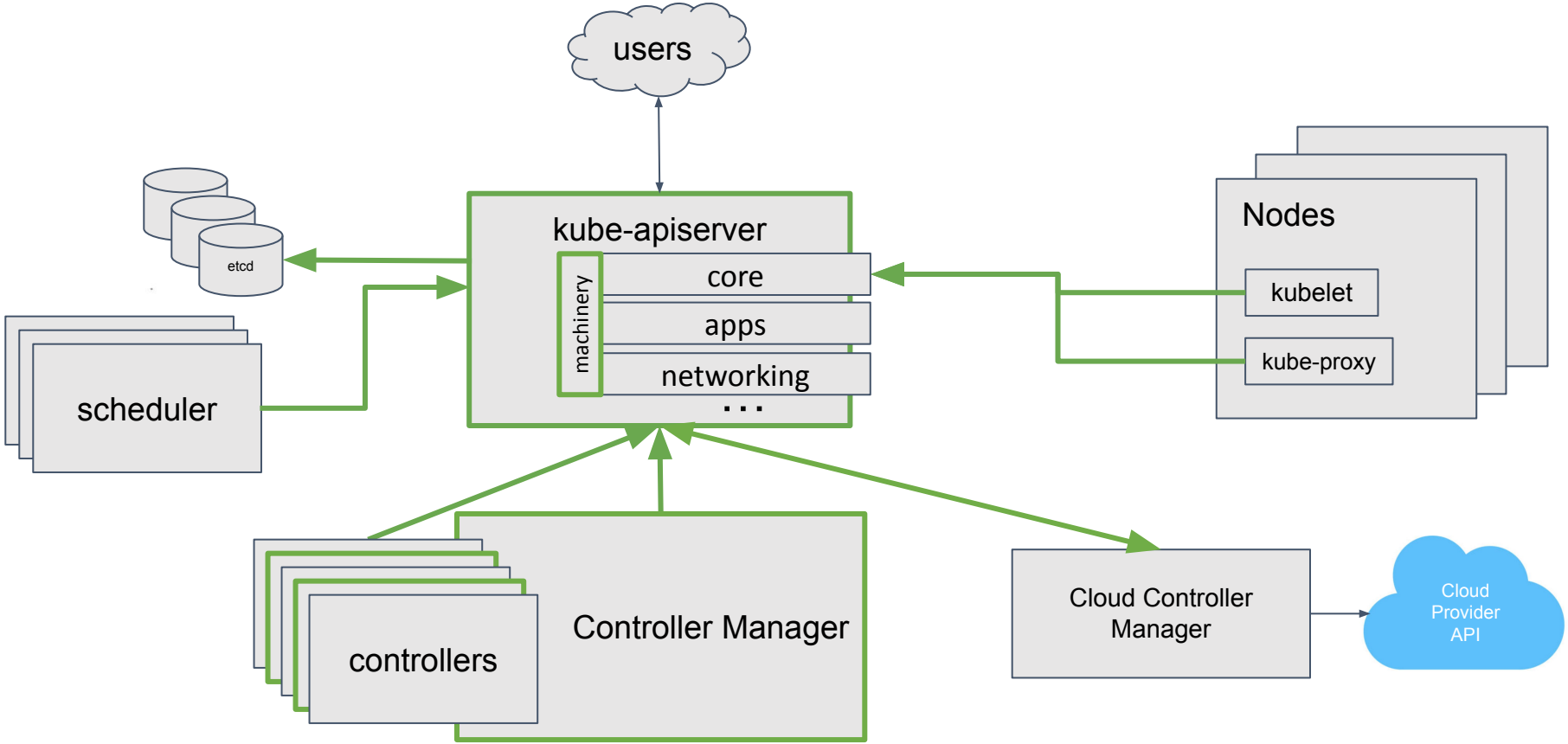


Interest over time

SIG Overview



SIG Overview



SIG Overview

SIG API Machinery is responsible for the development and enhancement of Kubernetes cluster control plane. The scope covers API server, persistence layer (etcd), controller manager, cloud controller manager, CustomResourceDefinition and webhooks. (SIG API Machinery Charter)

- What does the name API Machinery stand for?
- Is API Machinery != All Kubernetes APIs?
- API Machinery == the machinery used by different Kubernetes APIs to interact with the Kubernetes cluster, to be exposed and actuated, and the mechanics to publish, process, and extend them.

The Trouble With Project Names

What do we own?

- We provide stable core APIs to establish the permanent foundation for the rest of the K8s components to interact with.
- We enable and support other SIGs to be successful through the usage of our machinery

In scope

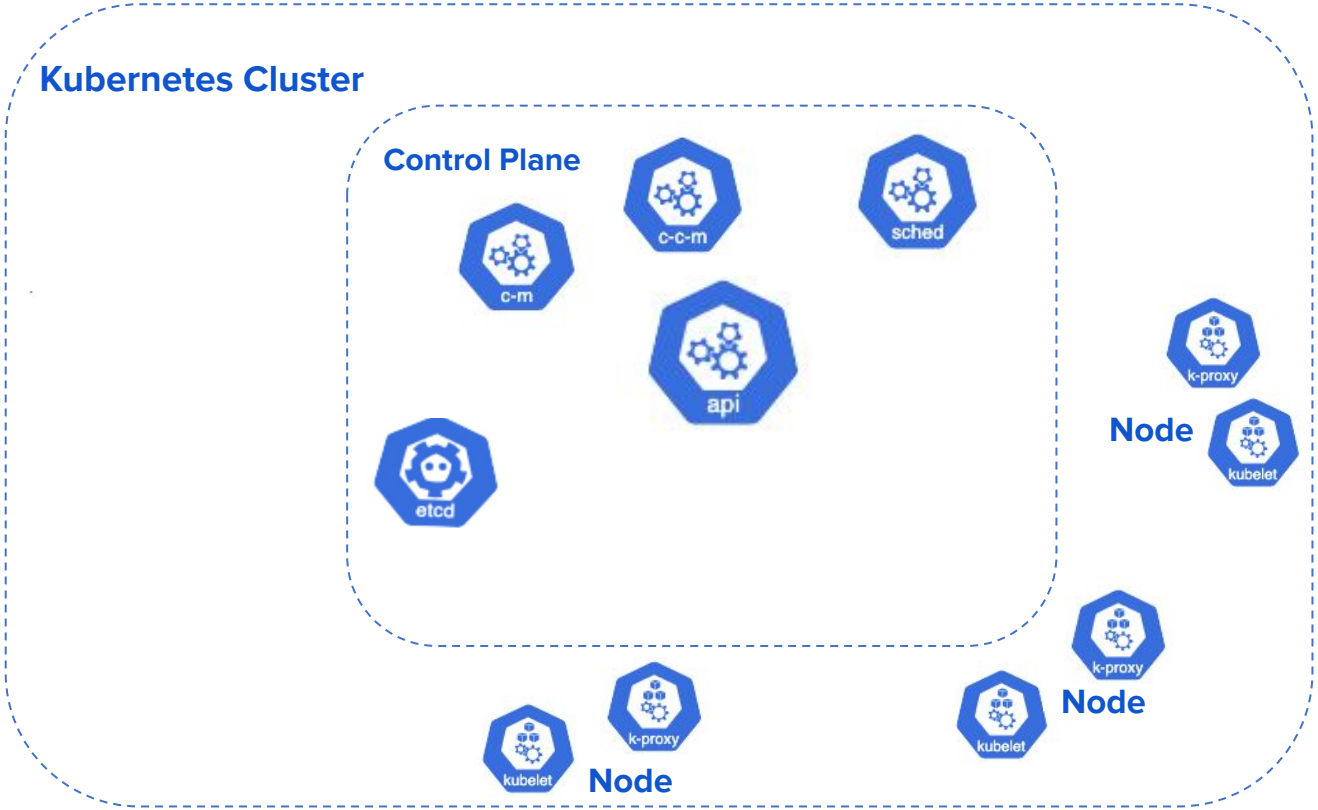
All aspects of

- How to read, modify, delete objects, including parsing, conversion, defaulting and validation
- Describing and extending the system: OpenAPI, Discovery, CRDs, Webhooks, client/informer libraries
- Maintaining a healthy system: controller-manager, garbage collection, namespace lifecycle
- The persistence layer (etcd, scalability SIGs)

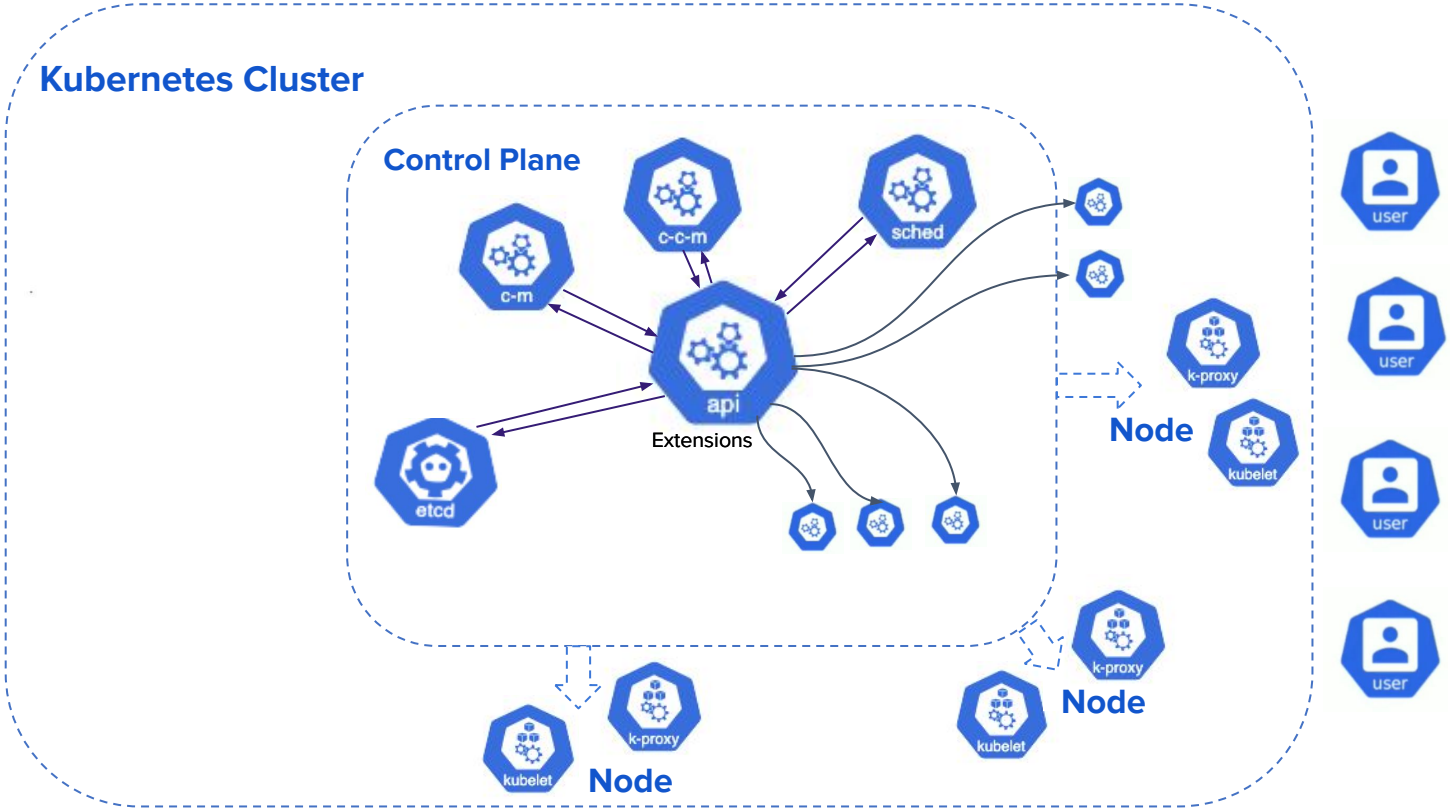
Out of scope

- All the individual Kubernetes APIs

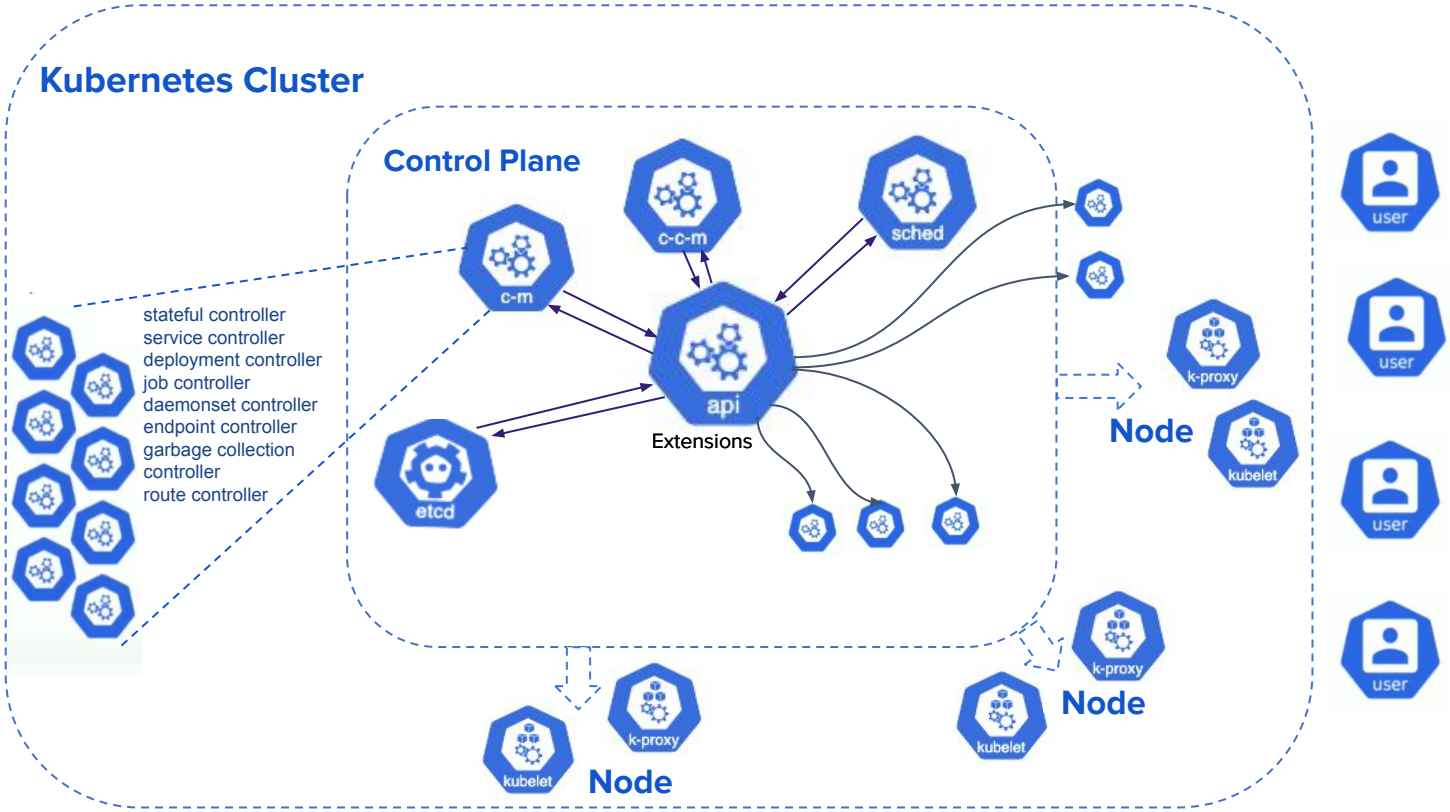
Why is it so important and complex?



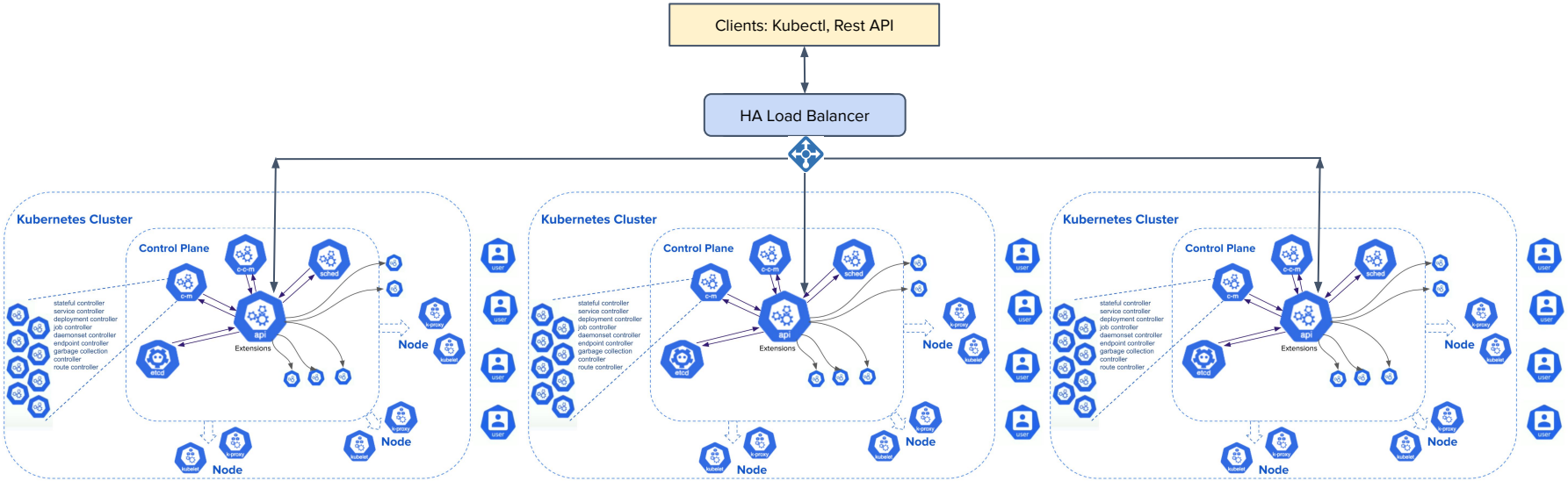
Why is it so important and complex?



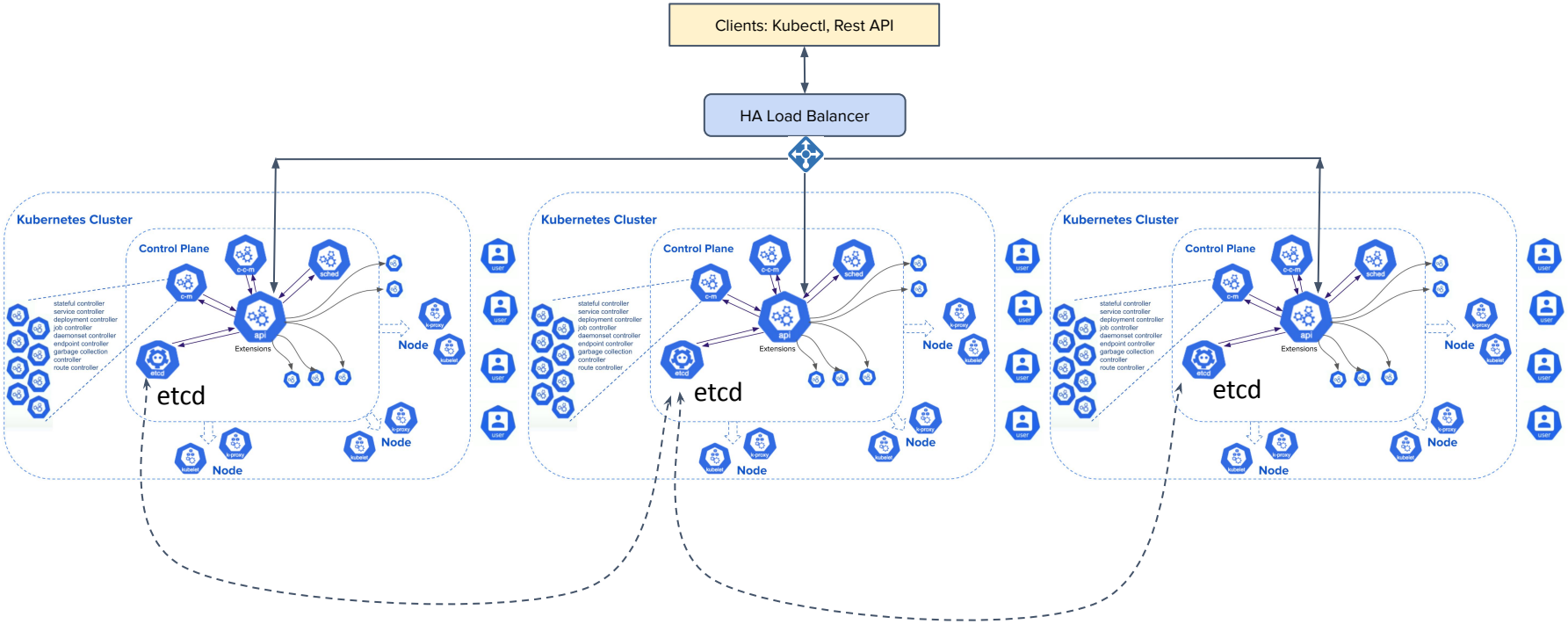
Why is it so important and complex?



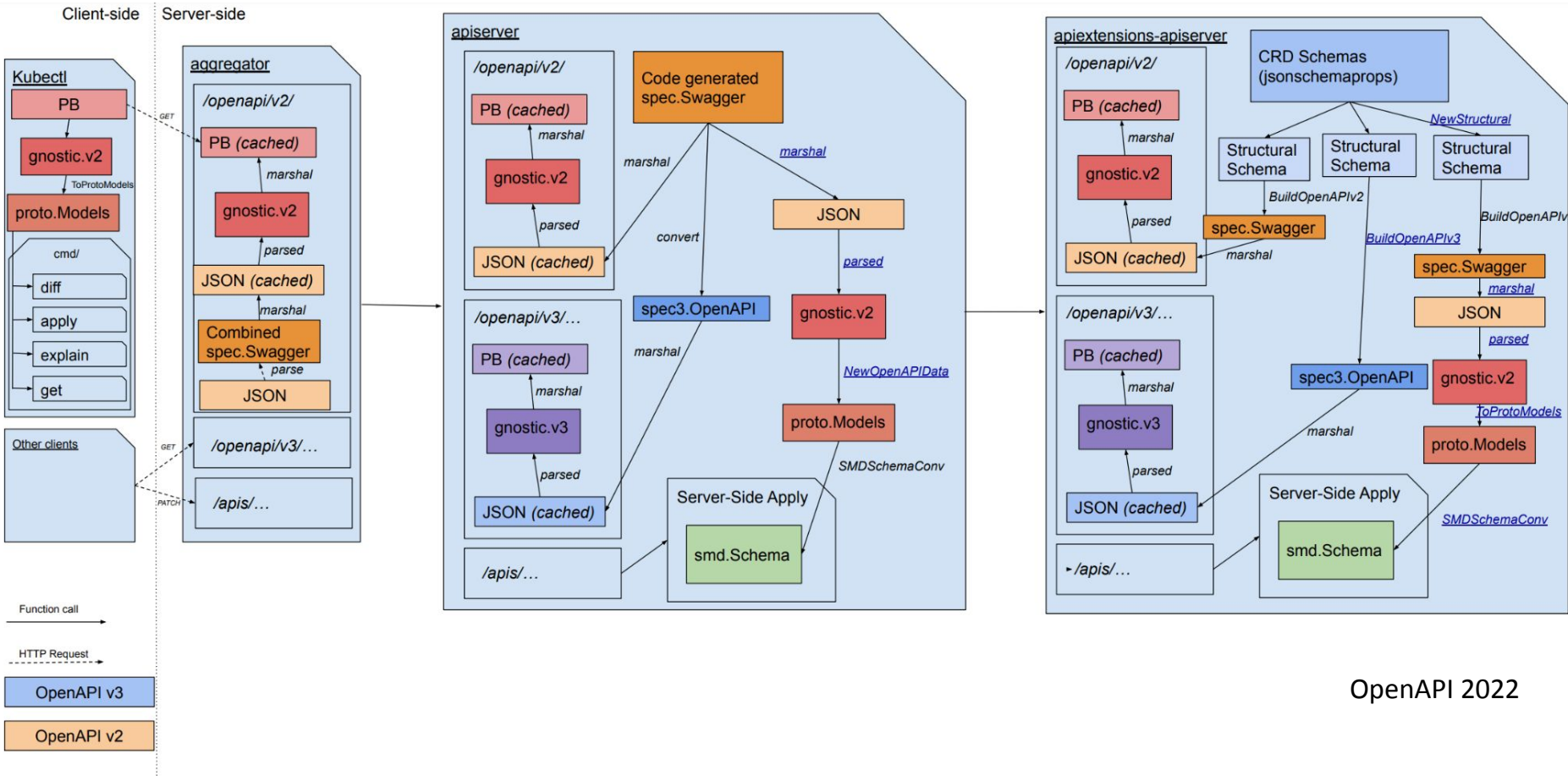
Why is it so important and complex?



Why is it so important and complex?

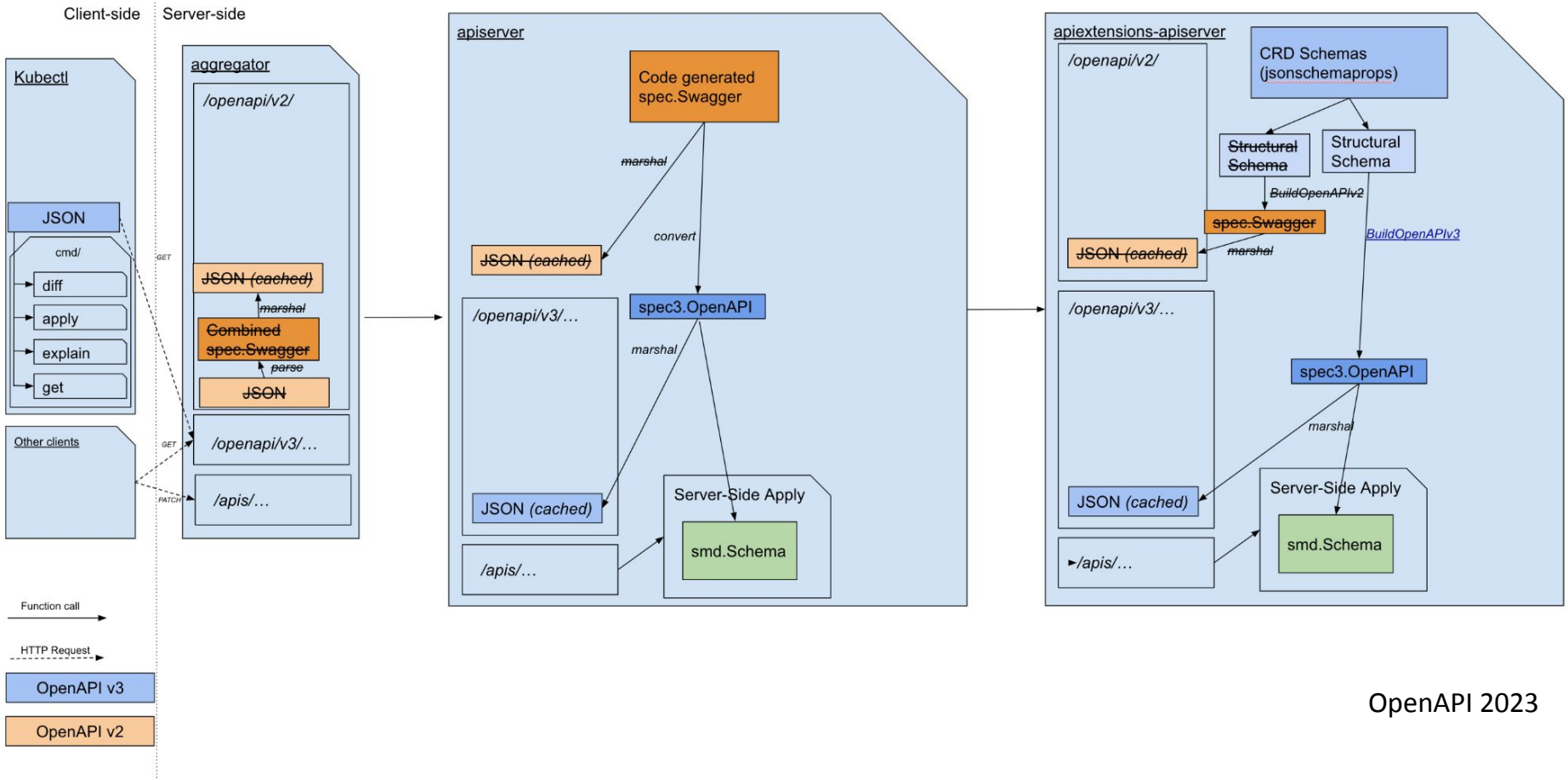


Why is it so important and complex?



OpenAPI 2022

Why is it so important and complex?

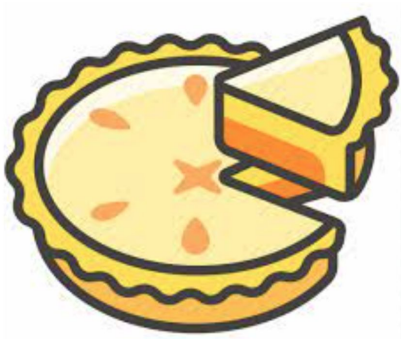


OpenAPI 2023

Code Ownership

25-40% code ownership

- 2 approaches with solid proxies:
  - SIG-labelled PRs vs file cataloguing



1.29 Enhancement Tracking

Title	Status	Stage	Type
Priority and Fairness for API Server Requests #1040	Tracked for Enhancements Freeze	Stable	Graduating
Transition from SPDY to WebSockets #4006	Tracked for Code Freeze	Alpha	Net New
Support paged LIST queries from the Kubernetes API #365	At Risk for Code Freeze	Stable	Graduating
Move Storage Version Migrator in-tree #4192	Tracked for Enhancements Freeze	Alpha	Net New
CRD Validation Expression Language #2876	At Risk for Code Freeze	Stable	Graduating
CBOR Serializer #4222	Tracked for Enhancements Freeze	Alpha	Net New
Allow informers for getting a stream of data instead of chunking #3157	Removed from Milestone	Beta	Graduating
CRD Validation Ratcheting #4008	At Risk for Code Freeze	Beta	Graduating
Declarative Validation #4153	Tracked for Enhancements Freeze	Alpha	Net New
CEL for Admission Control #3488	Tracked for Enhancements Freeze	Beta	Graduating

1.29 Enhancement Tracking

Title	Status	Stage	Type
Priority and Fairness for API Server Requests #1040	Tracked for Enhancements Freeze	Stable	Graduating
Transition from SPDY to WebSockets #4006	Tracked for Code Freeze	Alpha	Net New
Support paged LIST queries from the Kubernetes API #365	At Risk for Code Freeze	Stable	Graduating
Move Storage Version Migrator in-tree #4192	Tracked for Enhancements Freeze	Alpha	Net New
CRD Validation Expression Language #2876	At Risk for Code Freeze	Stable	Graduating
CBOR Serializer #4222	Tracked for Enhancements Freeze	Alpha	Net New
Allow informers for getting a stream of data instead of chunking #3157	Removed from Milestone	Beta	Graduating
CRD Validation Ratcheting #4008	At Risk for Code Freeze	Beta	Graduating
Declarative Validation #4153	Tracked for Enhancements Freeze	Alpha	Net New
CEL for Admission Control #3488	Tracked for Enhancements Freeze	Beta	Graduating

Performance

1.29 Enhancement Tracking

Title	Status	Stage	Type
Priority and Fairness for API Server Requests #1040	Tracked for Enhancements Freeze	Stable	Graduating
Transition from SPDY to WebSockets #4006	Tracked for Code Freeze	Alpha	Net New
Support paged LIST queries from the Kubernetes API #365	At Risk for Code Freeze	Stable	Graduating
Move Storage Version Migrator in-tree #4192	Tracked for Enhancements Freeze	Alpha	Net New
CRD Validation Expression Language #2876	At Risk for Code Freeze	Stable	Graduating
CBOR Serializer #4222	Tracked for Enhancements Freeze	Alpha	Net New
Allow informers for getting a stream of data instead of chunking #3157	Removed from Milestone	Beta	Graduating
CRD Validation Ratcheting #4008	At Risk for Code Freeze	Beta	Graduating
Declarative Validation #4153	Tracked for Enhancements Freeze	Alpha	Net New
CEL for Admission Control #3488	Tracked for Enhancements Freeze	Beta	Graduating

1.29 Enhancement Tracking

Title	Status	Stage	Type
Priority and Fairness for API Server Requests #1040	Tracked for Enhancements Freeze	Stable	Graduating
Transition from SPDY to WebSockets #4006	Tracked for Code Freeze	Alpha	Net New
Support paged LIST queries from the Kubernetes API #365	At Risk for Code Freeze	Stable	Graduating
Move Storage Version Migrator in-tree #4192	Tracked for Enhancements Freeze	Alpha	Net New
CRD Validation Expression Language #2876	At Risk for Code Freeze	Stable	Graduating
CBOR Serializer #4222	Tracked for Enhancements Freeze	Alpha	Net New
Allow informers for getting a stream of data instead of chunking #3157	Removed from Milestone	Beta	Graduating
CRD Validation Ratcheting #4008	At Risk for Code Freeze	Beta	Graduating
Declarative Validation #4153	Tracked for Enhancements Freeze	Alpha	Net New
CEL for Admission Control #3488	Tracked for Enhancements Freeze	Beta	Graduating

SIG Leadership

SIG Chairs:

- David Eads (@deads2k), Red Hat
- Federico Bongiovanni (@fedebongio), Google

SIG Technical Leads:

- David Eads (@deads2k), Red Hat
- Joe Betz (@jpbetz), Google

How to get involved?

- Regular SIG meetings:
  - **SIG Meeting**: 60 min / every 2 weeks (***Wed***, *recorded*)
  - **PR and Bug triage**: 30 min / twice every week (**Tue** & **Thu**)
- Regular Working Group meetings (API Expression, Kubebuilder, CEL)
- Upcoming project-based mentorship program

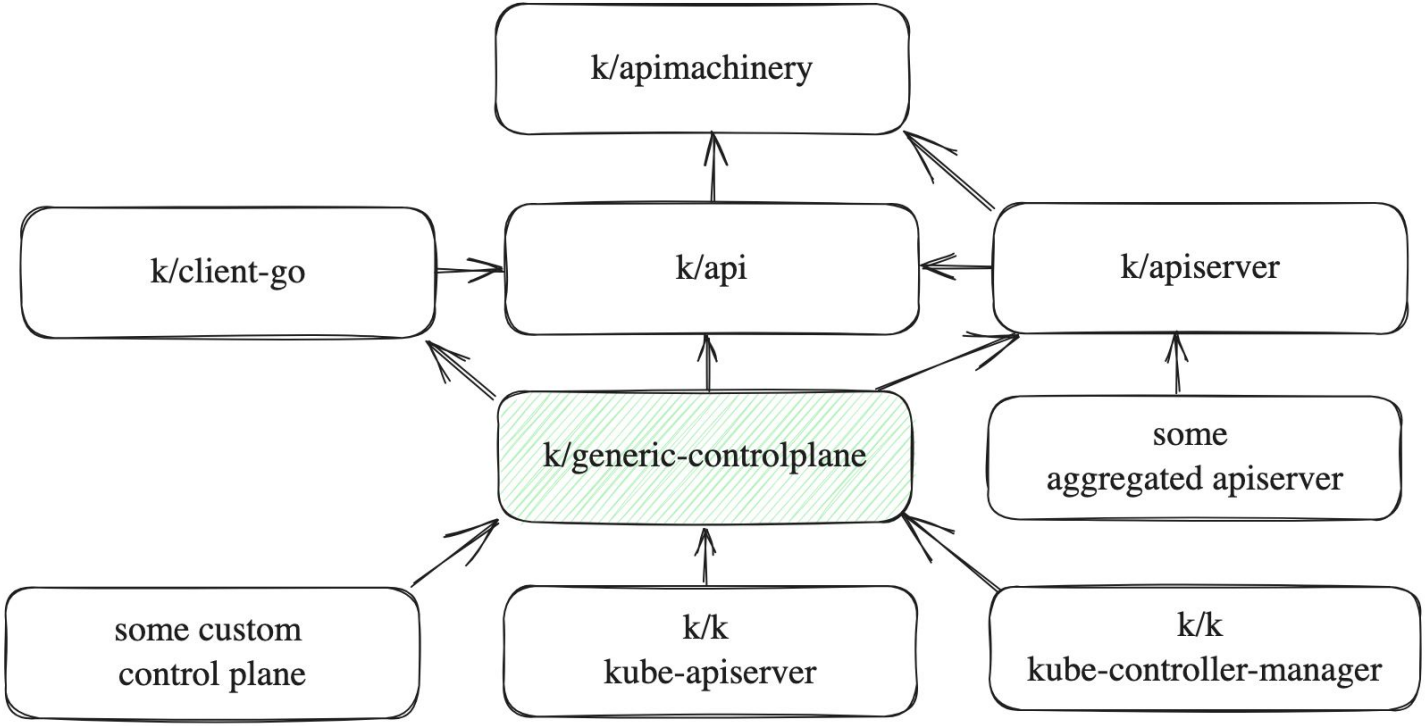
Some spotlights on topics under the radar.

Generic Control Plane

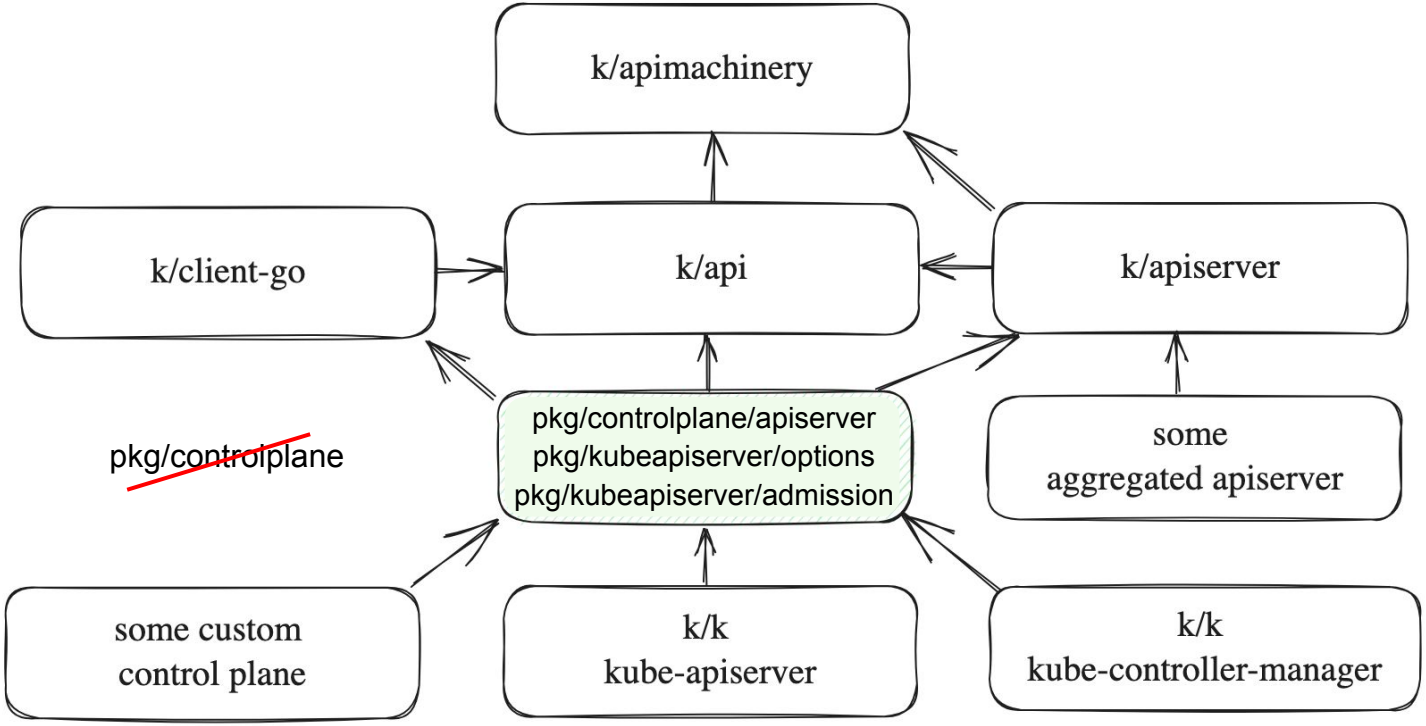
- https://github.com/kubernetes/enhancements/issues/4080

> A working kube-based control plane is more than just an apiserver component built on `k/apiserver`. It includes **standard resources** (depending on context namespaces, CRDs, RBAC, secrets, configmaps), and **standard controllers** (think of garbage collection, namespace deletion, etc.). *kube-apiserver* today is a bundle of those resources with container orchestration, *kube-controller-manager* equally for the corresponding controllers.
>
> **Separating the generic parts from container orchestration** will allow new use-cases building upon `k/apimachinery` and `k/apiserver`, while **keeping a unified codebase and ecosystem**, and by improving the factoring of *kube-apiserver* for easier maintenance due to less complexity by clear layering.

Where in the k/k codebase (eventually)



Where in the k/k code base (today)



What’s in

- CRDs
- namespaces
- secrets optional
- configmaps optional
- RBAC optional
- service accounts optional
- admission webhooks + policies optional
- quota optional
- aggregation, APIServices optional

Garbage Collection

Namespace Deletion

Quota

What’s in

- CRDs
- namespaces
- secrets
- configmaps
- RBAC
- service accounts
- admission webhooks + policies
- quota
- aggregation, APIServices

Garbage Collection

Namespace Deletion

Quota

Demo / Code

```go
    return aggregatorServer, nil
}

1 usage  Dr. Stefan Schimanski
func storageProviders(c controlplaneapiserver.CompletedConfig) ([]controlplaneapiserver.RESTStorageProvider, error) {
    return []controlplaneapiserver.RESTStorageProvider{
        withDisabledResources{
            disabled: map[string][]string{
                "v1": {"configmaps", "secrets", "serviceaccounts", "resourcequotas", "resourcequotas/status"},
            },   Schimanski, 05.11.23, 18:03 • Add cmd/sample-minimal-controlplane
            RESTStorageProvider: &corerest.GenericConfig{
                StorageFactory:              c.Extra.StorageFactory,
                EventTTL:                    c.Extra.EventTTL,
                LoopbackClientConfig:        c.Generic.LoopbackClientConfig,
                ServiceAccountIssuer:        c.Extra.ServiceAccountIssuer,
                ExtendExpiration:            c.Extra.ExtendExpiration,
                ServiceAccountMaxExpiration: c.Extra.ServiceAccountMaxExpiration,
                APIAudiences:                c.Generic.Authentication.APIAudiences,
                Informers:                   c.Extra.VersionedInformers,
            },
        },
        apiserverinternalrest.StorageProvider{},
        authenticationrest.RESTStorageProvider{Authenticator: c.Generic.Authentication.Authenticator, APIAudiences: c.Generic.Authentication.APIAudiences},
        flowcontrolrest.RESTStorageProvider{InformerFactory: c.Generic.SharedInformerFactory},
        coordinationrest.RESTStorageProvider{},
        eventsrest.RESTStorageProvider{TTL: c.EventTTL},
    }, nil
}

2 usages  Dr. Stefan Schimanski
type withDisabledResources struct {
    disabled map[string][]string
```

storageProviders(c controlplaneapiserver.CompletedConfig) ([]controlplaneapiserver.RESTStorageProvider, error)

```
# get https://vbom.ml/util?go-get=1
# get https://vbom.ml/util?go-get=1: Get "https://vbom.ml/util?go-get=1": dial tcp: lookup vbom.ml: no such host
go: github.com/rancher/wrangler@v0.8.5 requires
    sigs.k8s.io/cli-utils@v0.16.0 requires
    k8s.io/kubectl@v0.0.0-20191219154910-1528d4eea6dd requires
    vbom.ml/util@v0.0.0-20160121211510-db5cfe13f5cc: unrecognized import path "vbom.ml/util": https fetch: Get "https://vbom.ml/util?go-get=1": dial tcp: lookup vbom.ml: no such host
```

kubernetes > cmd > sample-minimal-controlplane > server > server.go 1198 Δ/1↑ 0↓ Blame: Schimanski 05.11.23, 18:03 209:15 LF UTF-8 Tab

What is a minimal Kube Control Plane?

Soon: generic, but extensible

$ go run ./cmd/**sample-generic-controlplane**

- **configmaps**.v1
- **resourcequotas**.v1
- **namespaces**.v1
- **secrets**.v1
- **serviceaccounts**.v1
- **events**.v1

- **apiservices**.apiregistration.k8s.io/v1
- **events**.events.k8s.io/v1

- **selfsubjectreviews**.authentication.k8s.io/v1
- **tokenreviews**.authentication.k8s.io/v1

- **selfsubjectaccessreviews**.authorization.k8s.io/v1
- **selfsubjectrulesreviews**.authorization.k8s.io/v1
- **localsubjectaccessreviews**.authorization.k8s.io/v1
- **subjectaccessreviews**.authorization.k8s.io/v1
- **clusterroles**.rbac.authorization.k8s.io/v1
- **rolebindings**.rbac.authorization.k8s.io/v1
- **roles**.rbac.authorization.k8s.io/v1
- **clusterrolebindings**.rbac.authorization.k8s.io/v1

- **certificatesigningrequests**.certificates.k8s.io/v1

- **mutatingwebhookconfigurations**.admissionregistration.k8s.io/v1
- **validatingwebhookconfigurations**.admissionregistration.k8s.io/v1
- **validatingadmissionpolicies**.admissionregistration.k8s.io/v1beta1
- **validatingadmissionpolicybindings**.admissionregistration.k8s.io/v1beta1

- **customresourcedefinitions**.apiextensions.k8s.io/v1

- **leases**.coordination.k8s.io/v1

- **prioritylevelconfigurations**.flowcontrol.apiserver.k8s.io/v1beta3
- **flowschemas**.flowcontrol.apiserver.k8s.io/v1beta3
- **storageversions**.internal.apiserver.k8s.io/v1alpha1

WIP: generic, not extensible, no RBAC

$ go run ./cmd/**sample-minimal-controlplane**

- **namespaces**.v1
- **events**.v1
- **apiservices**.apiregistration.k8s.io/v1
- **events**.events.k8s.io/v1
- **selfsubjectreviews**.authentication.k8s.io/v1
- **tokenreviews**.authentication.k8s.io/v1
- **leases**.coordination.k8s.io/v1
- **flowschemas**.flowcontrol.apiserver.k8s.io/v1beta3
- **prioritylevelconfigurations**.flowcontrol.apiserver.k8s.io/v1beta3
- **storageversions**.internal.apiserver.k8s.io/v1alpha1

- **your natively implemented APIs**

Tilt: not even a cluster

$ **tilt** api-resources

- **clusters**.tile.dev/v1alpha1
- **cmdimages**.tile.dev/v1alpha1
- **cmds**.tile.dev/v1alpha1
- **configmaps**.tile.dev/v1alpha1
- **dockercomposelogstreams**.tile.dev/v1alpha1
- **dockercomposeservices**.tile.dev/v1alpha1
- **dockerimages**.tile.dev/v1alpha1
- **extensionrepos**.tile.dev/v1alpha1
- **extensions**.tile.dev/v1alpha1
- **filewatches**.tile.dev/v1alpha1
- **imagemaps**.tile.dev/v1alpha1
- **kubernetesapplys**.tile.dev/v1alpha1
- **kubernetesdiscoveries**.tile.dev/v1alpha1
- **liveupdates**.tile.dev/v1alpha1
- **podlogstreams**.tile.dev/v1alpha1
- **portforwards**.tile.dev/v1alpha1
- **sessions**.tile.dev/v1alpha1
- **tiltfiles**.tile.dev/v1alpha1
- **togglebuttons**.tile.dev/v1alpha1
- **uibuttons**.tile.dev/v1alpha1
- **uiresources**.tile.dev/v1alpha1
- **uisessions**.tile.dev/v1alpha1

Acorn: no cluster, not even kube-apiserver

$ **acorn** kube kubectl api-resources

- **configmaps**.v1
- **secrets**.v1
- **subjectaccessreviews**.authorization.k8s.io/v1

- **accountevents**.account.manager.acorn.io/v1
- **accountquotas**.account.manager.acorn.io/v1
- **appcleanuppolicies**.account.manager.acorn.io/v1
- **imagerolebindings**.account.manager.acorn.io/v1
- **projectquotas**.account.manager.acorn.io/v1
- **projectrolebindings**.account.manager.acorn.io/v1
- **quotatotals**.account.manager.acorn.io/v1
- **quotausages**.account.manager.acorn.io/v1
- **workspaces**.account.manager.acorn.io/v1
- **quotarequests**.admin.acorn.io/v1

- **acornimagebuilds**.api.acorn.io/v1
- **apps**.api.acorn.io/v1
- **builders**.api.acorn.io/v1
- **computeclasses**.api.acorn.io/v1
- **containerreplicas**.api.acorn.io/v1
- **credentials**.api.acorn.io/v1
- **devsessions**.api.acorn.io/v1
- **events**.api.acorn.io/v1
- **imageallowrules**.api.acorn.io/v1
- **images**.api.acorn.io/v1
- **infos**.api.acorn.io/v1
- **jobs**.api.acorn.io/v1
- **projects**.api.acorn.io/v1
- **regions**.api.acorn.io/v1
- **secrets**.api.acorn.io/v1
- **volumeclasses**.api.acorn.io/v1
- **volumes**.api.acorn.io/v1

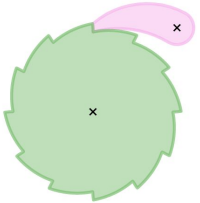
What is a minimal Kube Control Plane?

Where do we want to go as Kube project?

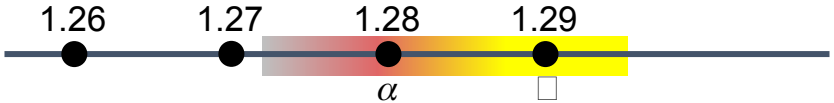
Ratcheting Validation

- https://github.com/kubernetes/enhancements/issues/4008

> **Modifying a value validation** on a CRD today means that **you risk breaking the workflow of all your users**, this high price to pay limits adoption, and degrades the Kubernetes user experience: we are prevented from shifting validation logic left.
>
> Goals
>
> - Remove barriers blocking CRD authors from widening value validations
> - Remove barriers blocking CRD authors from tightening value validations
> - Do this automatically for all CRDs installed into clusters with the feature enabled



Work by Alex Zielenski, so all the credits to him for starting this important work.



The Challenge

```yaml
apiVersion: apiextensions.k8s.io/v1
kind: CustomResourceDefinition
...
  schema:
    properties:
      spec:
        properties:
          replicas:
            type: number
          ip:
            type: string
```

```yaml
$ cat object.yaml
apiVersion: group/v1
kind: Foo
spec:
  replicas: 2.5
  ip: “1.2.3.4.5”

$ kubectl apply -f object.yaml
```

The Challenge

```yaml
apiVersion: apiextensions.k8s.io/v1
kind: CustomResourceDefinition
...
  schema:
    properties:
      spec:
        properties:
          replicas:
            type: integer
            minimum: 0
          ip:
            type: string
            format: ipv4
```

```yaml
$ cat object.yaml
apiVersion: group/v1
kind: Foo
spec:
  replicas: 2.5
  ip: “1.2.3.4.5”

$ kubectl apply -f object.yaml
```

The Challenge

```yaml
apiVersion: apiextensions.k8s.io/v1
kind: CustomResourceDefinition
…
  schema:
    properties:
      spec:
        properties:
          replicas:
            type: integer
            minimum: 0
          ip:
            type: string
            format: ipv4
```

```
$ cat object.yaml
apiVersion: group/v1
kind: Foo
spec:
  replicas: 2.5
  ip: “1.2.3.4.5”

$ kubectl edit foo
$ kubectl annotate foo x=y
$ kubectl edit # remove a finalizer
```

But CEL…

```
apiVersion: apiextensions.k8s.io/v1
kind: CustomResourceDefinition
…
  schema:
    properties:
      spec:
        properties:
          replicas:
            type: integer
            x-kubernetes-validations:
            - rule: "double(int(oldSelf))!=oldSelf || double(int(self))==self"
              message: only integers are allowed
            - rule: “oldSelf < 0 || self >= 0”
              message: must be zero or positive
```

But CEL…

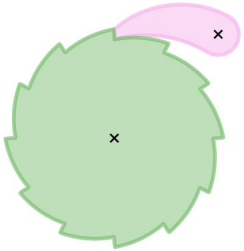
```
apiVersion: apiextensions.k8s.io/v1
kind: CustomResourceDefinition
…
  schema:
    properties:
      spec:
        properties:
          replicas:
            type: integer
            x-kubernetes-validations:
            - rule: "double(int(oldSelf))!=oldSelf || double(int(self))==self"
              message: only integers are allowed
            - rule: “oldSelf < 0 || self >= 0”
              message: must be zero or positive
```

these only apply on update

New: Automatic Ratcheting Validation

```
apiVersion: apiextensions.k8s.io/v1
kind: CustomResourceDefinition
...
  schema:
    properties:
      spec:
        properties:
          replicas:
            type: integer
            minimum: 0
          ip:
            type: string
            format: ipv4
```

```
apiVersion: group/v1
kind: Foo
spec:
  replicas: 2.5
  ip: “1.2.3.4.5”
```



Each box is “ratcheted”:

1. Validate **green box** only if **replicas** changes.
2. Validate **blue box** only if **ip** changes.

But CEL…sure – New: OptionalOldSelf


```yaml
apiVersion: apiextensions.k8s.io/v1
kind: CustomResourceDefinition
…
  schema:
    properties:
      spec:
        properties:
          replicas:
            type: integer
            x-kubernetes-validations:
            - rule: >
                (oldSelf.hasValue() && double(int(oldSelf.value())) != oldSelf.value()) ||
                double(int(self))==self"
              message: only integers are allowed
              optionalOldSelf: true
            - rule: "(oldSelf.hasValue() && oldSelf.value() < 0) || self >= 0"
              message: must be zero or positive
              optionalOldSelf: true
```

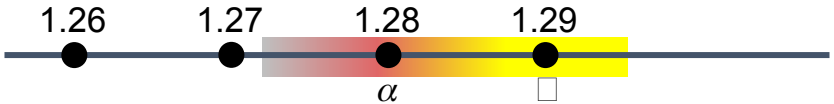
These also apply on Create!

Ratcheting Validation

- https://github.com/kubernetes/enhancements/issues/4008

> **Modifying a value validation** on a CRD today means that **you risk breaking the workflow of all your users**, this high price to pay limits adoption, and degrades the Kubernetes user experience: we are prevented from shifting validation logic left.
>
> Goals
>
> - Remove barriers blocking CRD authors from widening value validations
> - Remove barriers blocking CRD authors from tightening value validations
> - Do this automatically for all CRDs installed into clusters with the feature enabled

Work by Alex Zielenski, so all the credits to him for starting this important work.



KubeCon | CloudNativeCon
North America 2023

Thanks!

Meet us on Thursday, 12pm-3pm during
“Meet the Kubernetes Contributor Community”

Time for questions!