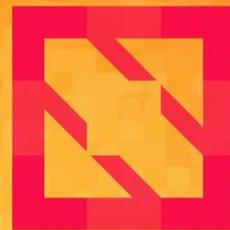




KubeCon



CloudNativeCon

North America 2019





KubeCon



CloudNativeCon

North America 2019

Meet the Gears Behind Kubernetes APIs: Introduction to SIG API-Machinery

Federico Bongiovanni - @fedebongio

Engineering Manager at Google - Co-Chair of SIG API-Machinery



Introduction to SIG API-Machinery



KubeCon



CloudNativeCon

North America 2019

Agenda

1. Introduction & SIG Overview
2. What do we own?
3. What do we not own?
4. Why it's so central and important (and hard)?
5. What did we do for you lately?
6. Roadmap / Working Groups
7. Recurrent Meetings & How to get involved

SIG Overview



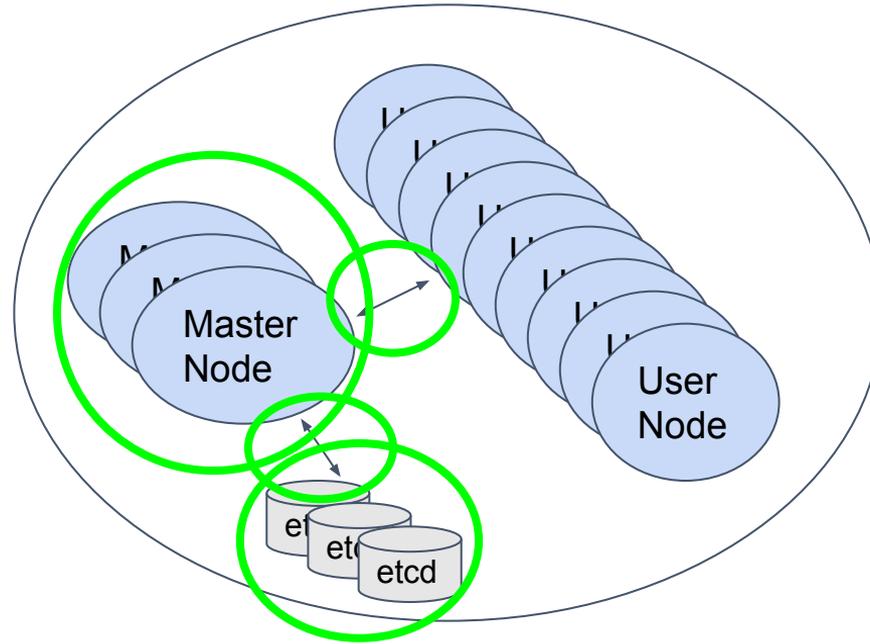
KubeCon



CloudNativeCon

North America 2019

Kubernetes Cluster



What runs in the master nodes is usually called the Kubernetes Cluster Control Plane



Is API Machinery == Kubernetes APIs?



No. . . .

**But it does own a few individual APIs,
which turn to use the same machinery
as the rest of the other individual APIs.**



API Machinery != Kubernetes APIs

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 Charter](#))

API Machinery = the machinery ("technical stack") used by the different Kubernetes APIs to be exposed and actuated, and the mechanisms to publish, process and extend them.

What do we own?



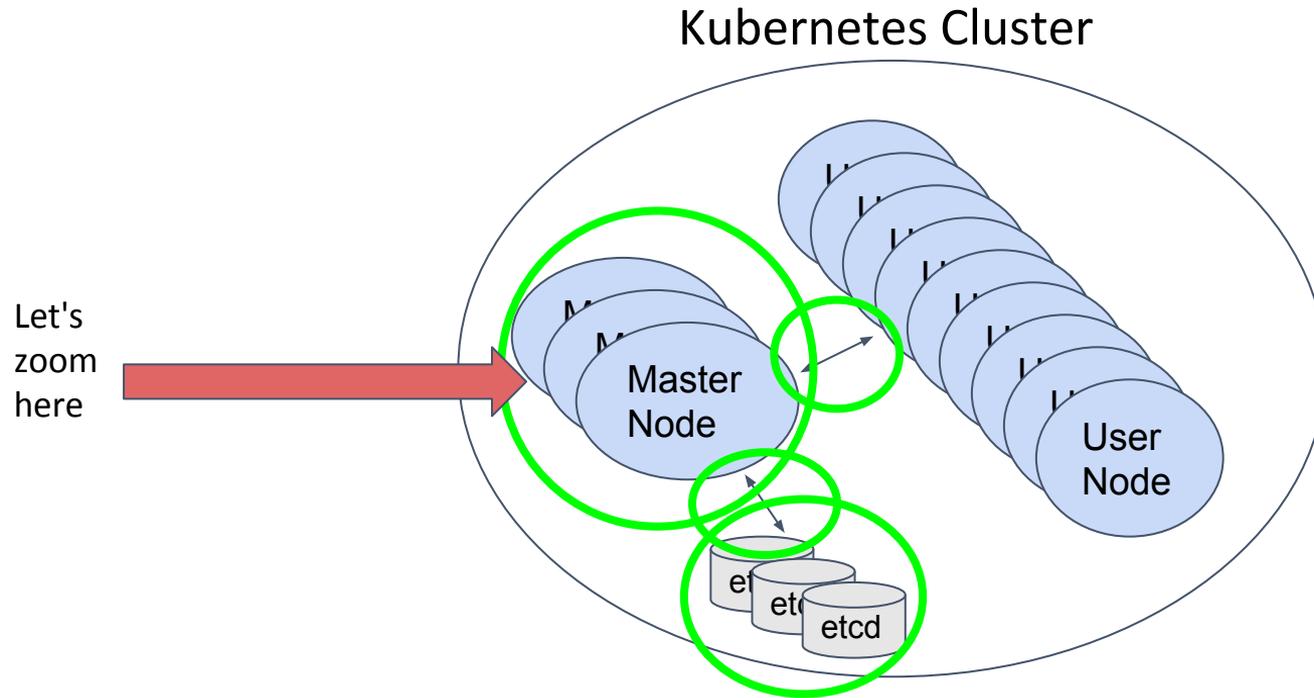
KubeCon



CloudNativeCon

North America 2019

Kubernetes Cluster Control Plane



What do we own?



KubeCon



CloudNativeCon

North America 2019

The formal and precise list of components can be found in our [charter](#):

In other words...

- kube-apiserver
 - ...and generic apiserver library
 - storage path (to etcd)
- Controller framework and some concrete controllers like
 - Garbage Collector
 - Namespace
- Client generators (and client-go)
- Serialization stack
- Object conversion and defaulting mechanisms
- **Extension mechanisms (CRDs and Admission Webhooks)**

In scope

Code, Binaries and Services

All aspects of

- API server
- API registration and discovery
- Generic API CRUD semantics
- Admission control
- Encoding/decoding
- Conversion
- Defaulting
- Persistence layer (etcd)
- OpenAPI
- The informer libraries
- CustomResourceDefinition
- Webhooks
- Garbage collection
- Namespace lifecycle
- Client libraries

Cross-cutting and Externally Facing Processes

Client library releases

What do we own?



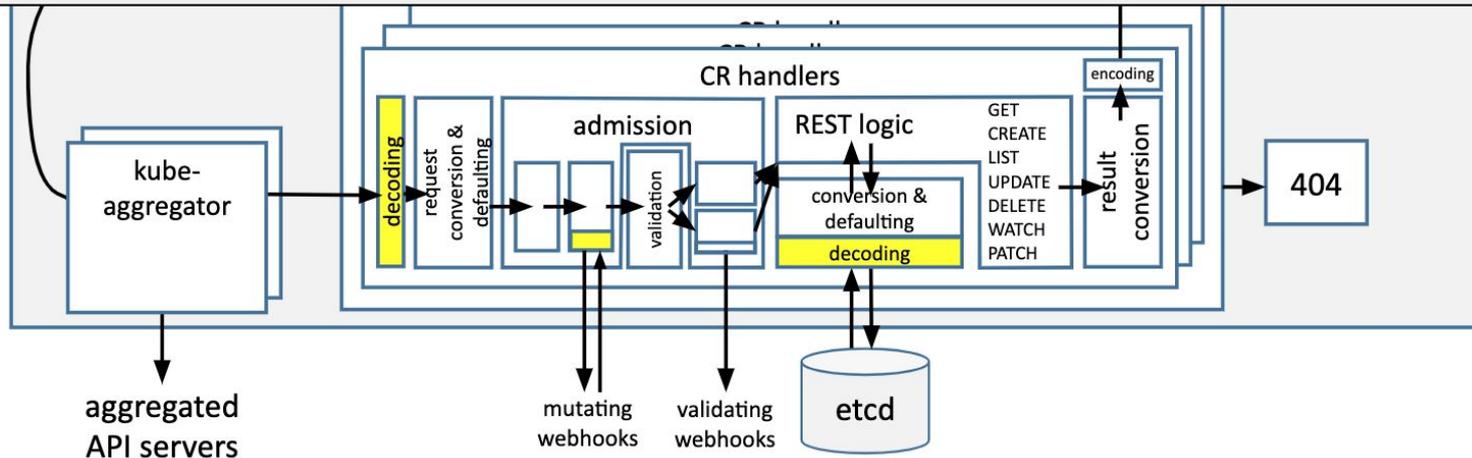
KubeCon



CloudNativeCon

North America 2019

Kubernetes (through API Machinery) allows you to extend Kubernetes APIs through CRDs and Admission Webhooks, and it will treat your new resources as first class citizens, leveraging all the power of Kubernetes



Did you know?

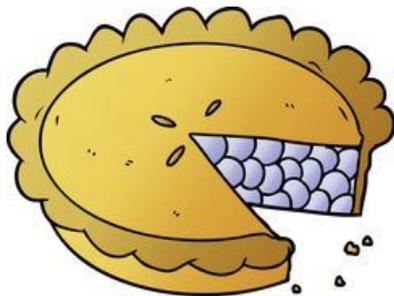


KubeCon



CloudNativeCon

North America 2019



...about **25%** of the Kubernetes code base (!) belongs to API Machinery?





KubeCon



CloudNativeCon

North America 2019

"**Kubernetes** is not just API-driven,
but is **API-centric**".

– *Brian Grant, Principal Engineer, Google*

What do we not own?

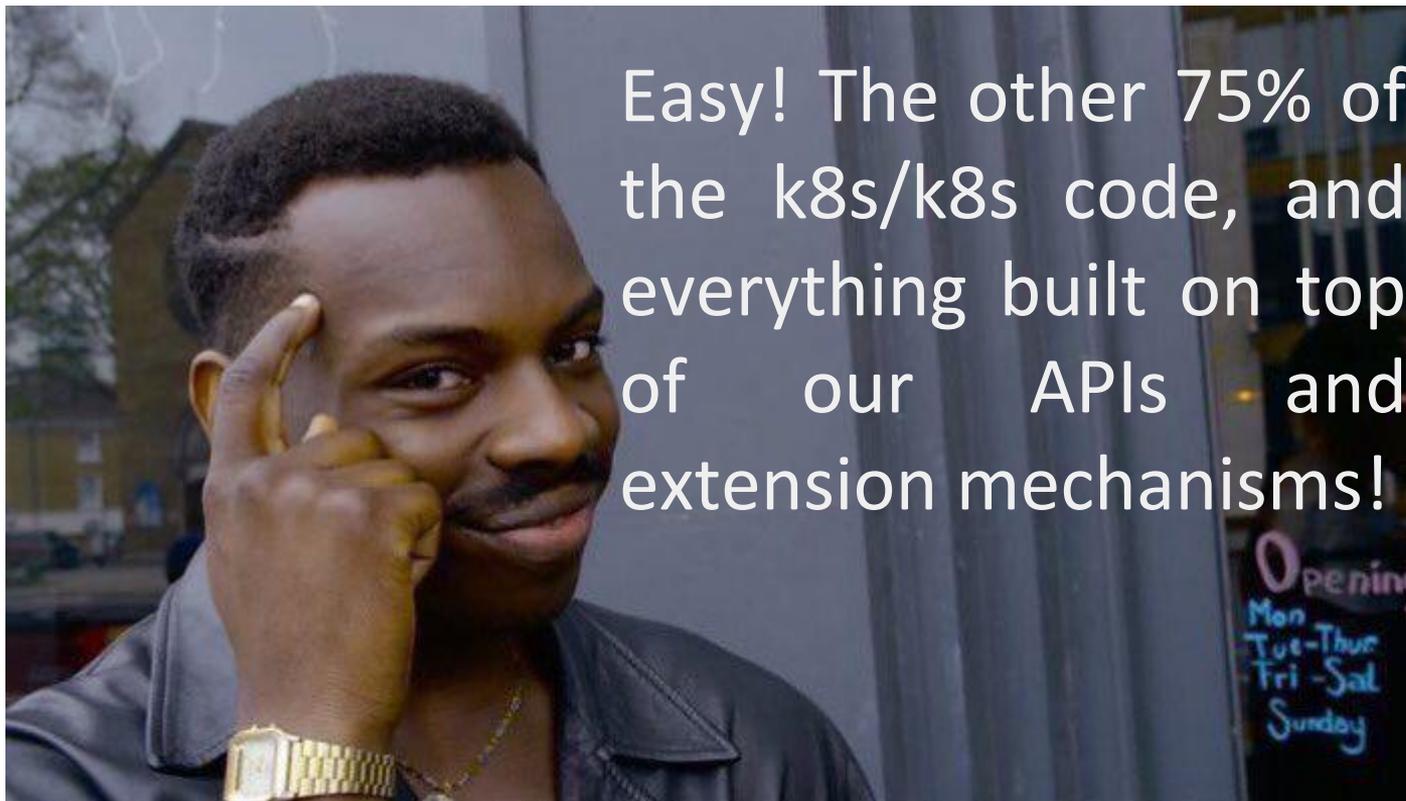


KubeCon



CloudNativeCon

North America 2019



What do we not own?



KubeCon



CloudNativeCon

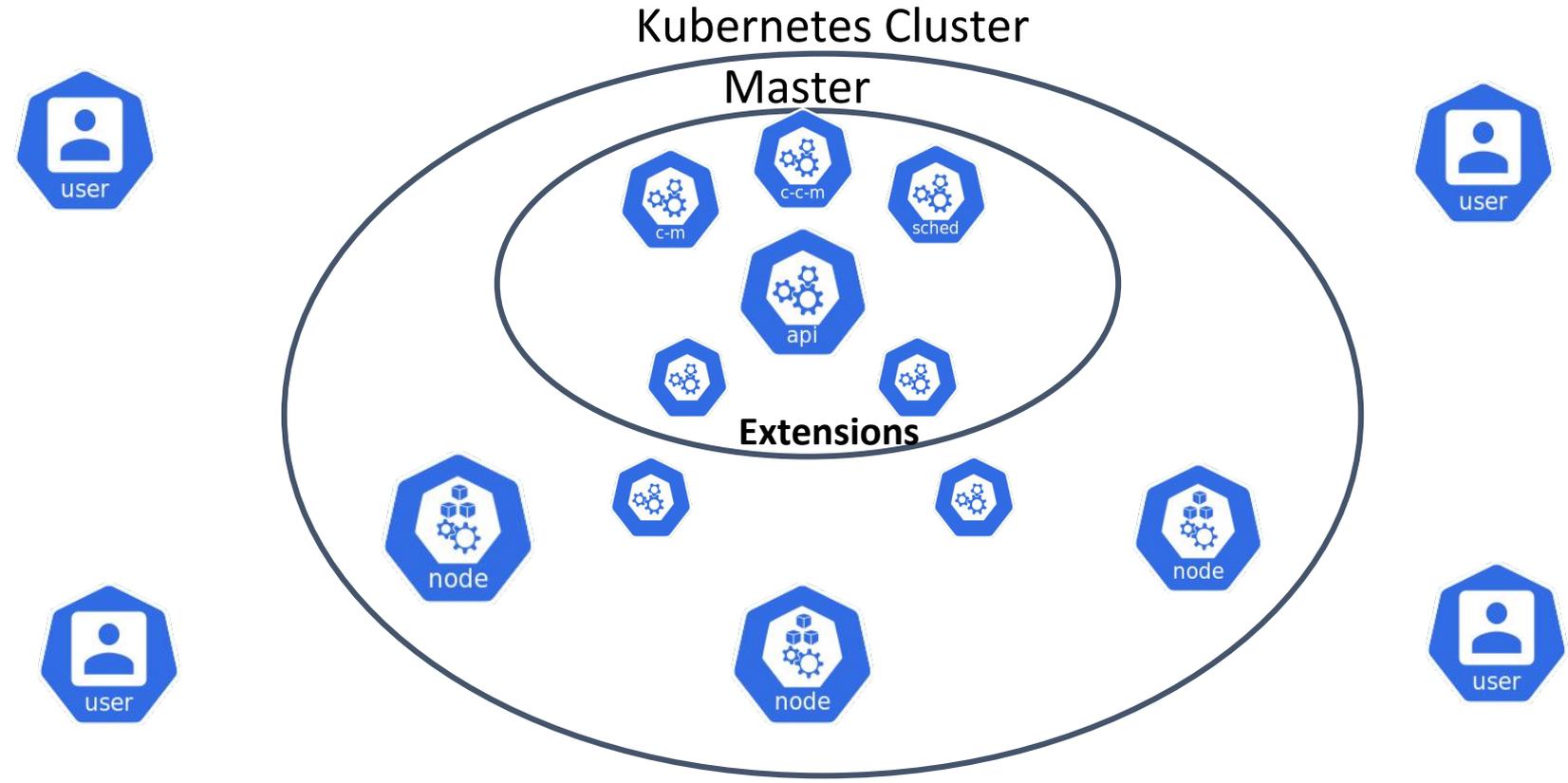
North America 2019

API Machinery = the generic machinery ("technical stack") used by the different Kubernetes APIs to be exposed and actuated, and the mechanisms to publish, process and extend them.

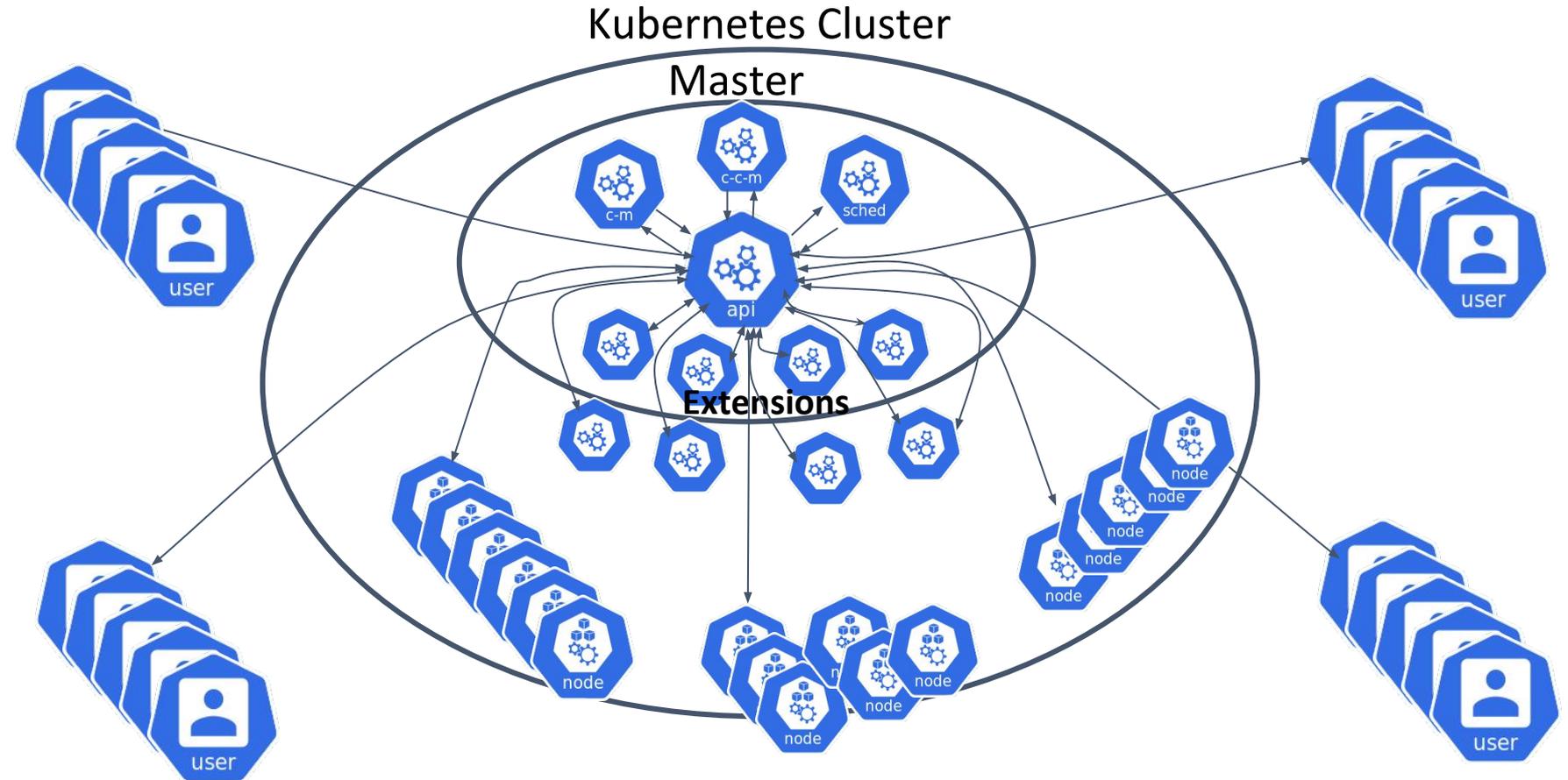
So with that said, we are not the owners of:

- All the individual Kubernetes APIs (each API is owned by it's own SIG or WG)
- We don't own most controllers (each Controller has a concrete owner which is not us)
- We usually don't own the operational characteristics of your cluster (support questions)
- We don't own your particular API review :)
 - ...unless you're trying to set up a pattern for everyone.
- "downward facing" extension mechanisms and frameworks

Why it's so central and important (and hard)?



Why it's so central and important (and hard)?



Why it's so central and important (and hard)?



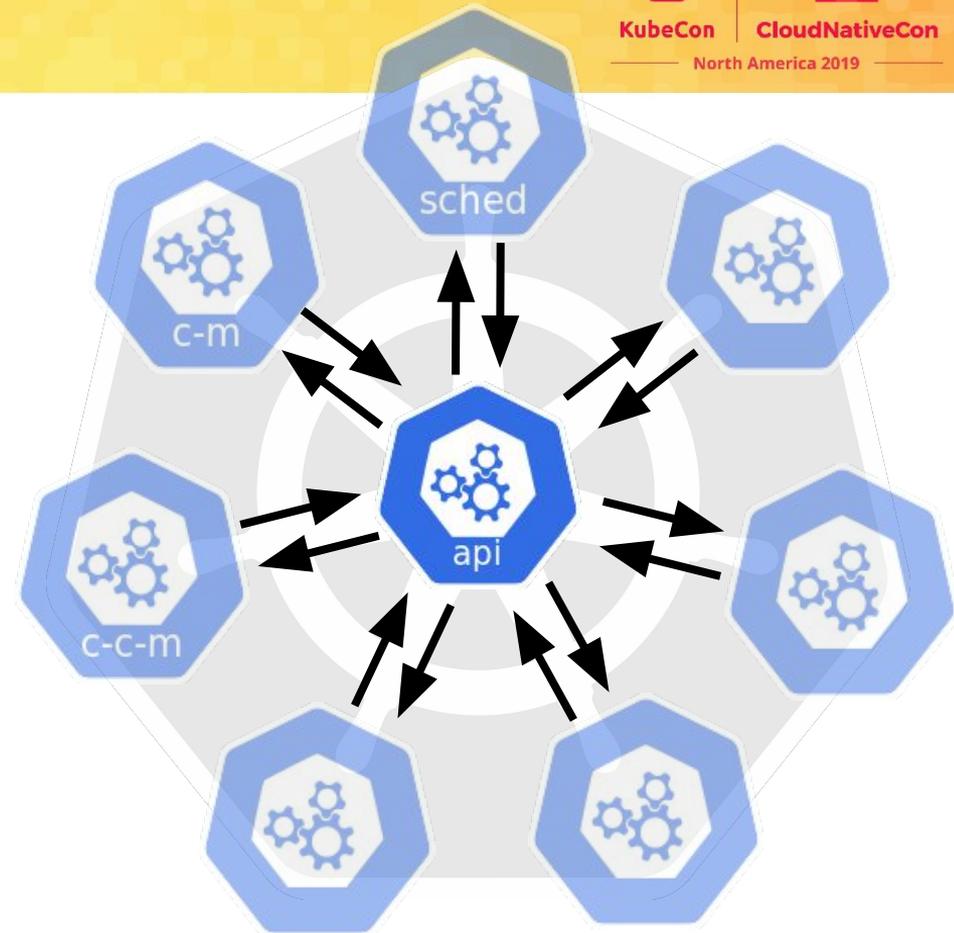
KubeCon



CloudNativeCon

North America 2019

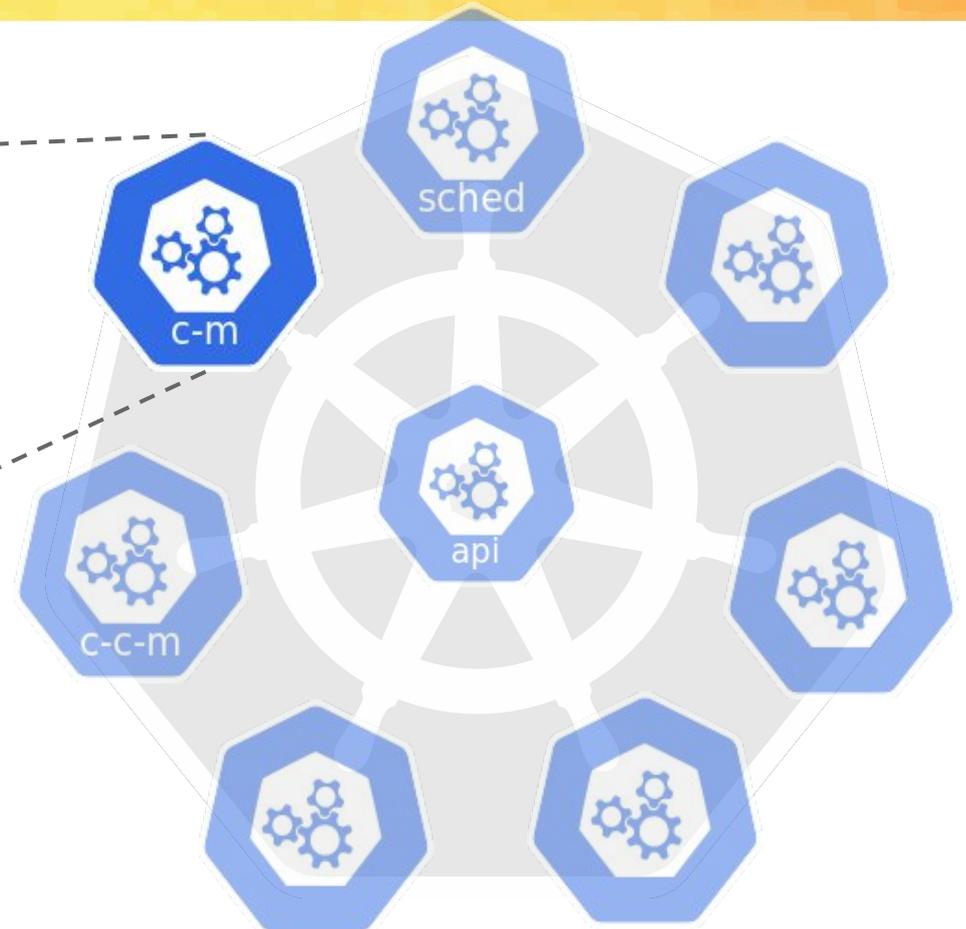
Kube API Server



Why it's so central and important (and hard)?

Controller Manager

- statefulset controller
- service controller
- deployment controller
- job controller
- daemonset controller
- endpoint controller
- garbage collection controller
- route controller



Why it's so central and important (and hard)?



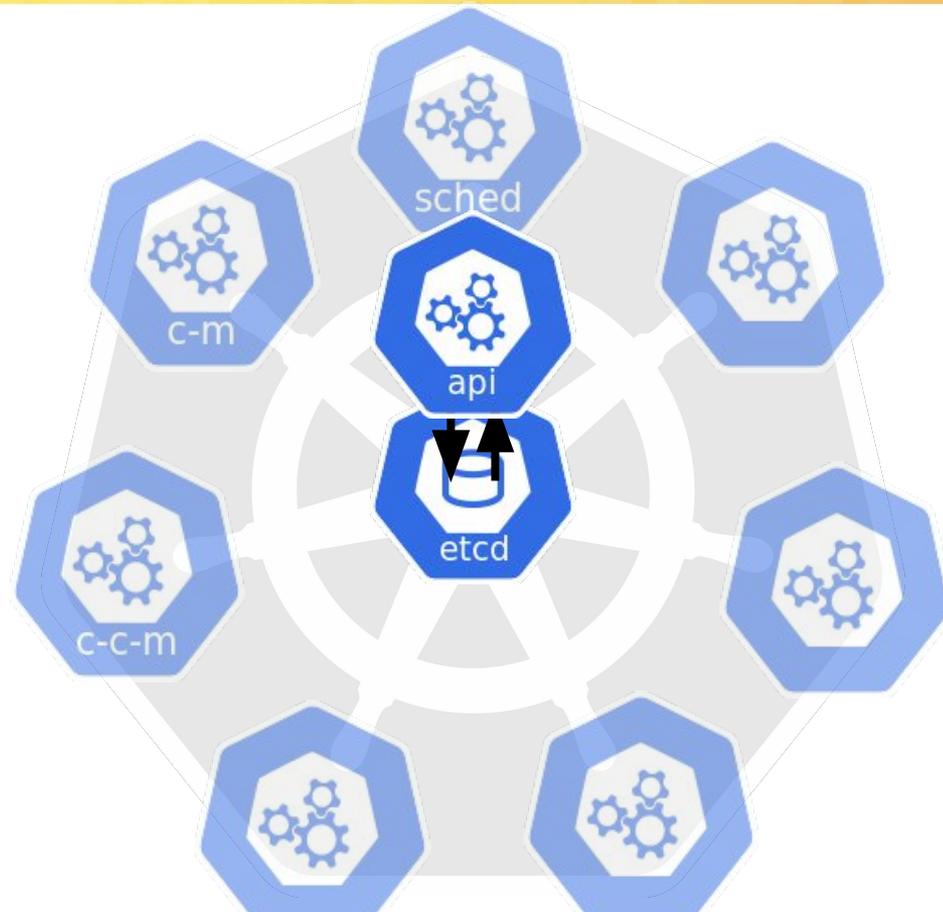
KubeCon



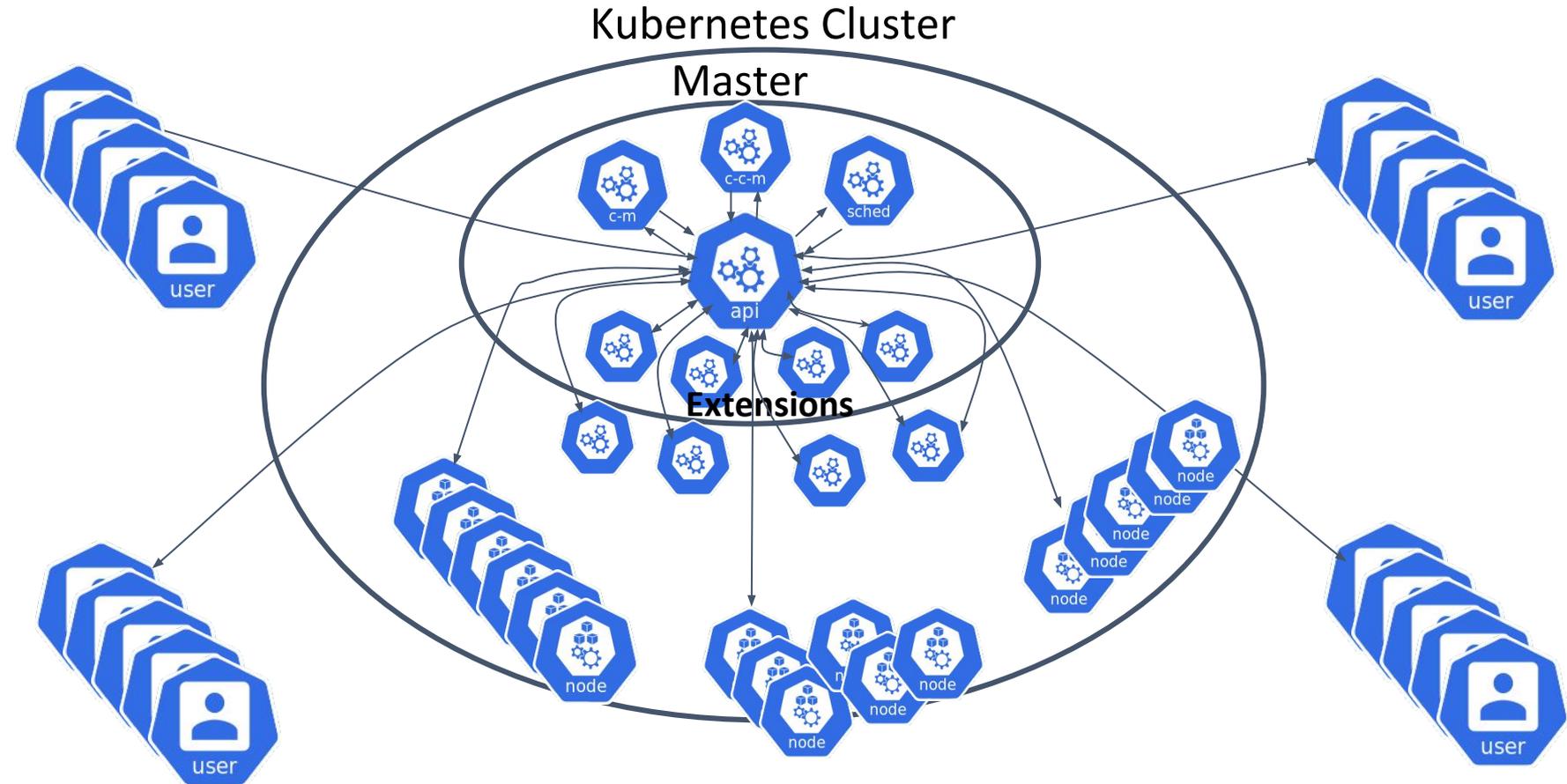
CloudNativeCon

North America 2019

etcd



Why it's so central and important (and hard)?



Why it's so central and important (and hard)?



KubeCon



CloudNativeCon

North America 2019



What did we do for you lately?



KubeCon



CloudNativeCon

North America 2019

- "Extensibility" to GA/Stable in [1.16](#)
 - CRD promoted to GA in 1.16
 - Admission Webhooks promoted to GA in 1.16
 - "Defaulting" promoted to to Beta - only available in v1
- Server-side apply maturity - promoted to beta
 - [Please try it and provide feedback](#)
- (related) Storage Migrator to Beta
- (related) Network Proxy to Alpha (towards the removal of SSH tunnels)
- (related) Better and more stable metrics for the future
- (related) helped build, release and upgrade to **etcd 3.4** (huge improvements)
- Contributed to increase ~11% addition of conformance tests

Roadmap



KubeCon



CloudNativeCon

North America 2019

- API server proxy support / SSH Tunnels deprecation
 - Trying for beta in 1.18 - see [KEP](#)
- Server-side apply (dedicated working group)
 - Looking for feedback for another beta
 - Multi-stage beta discussions - see [discussion](#).
- [Priority and fairness for API requests](#)
 - Trying to land as alpha in the 1.18 release--just missed 1.17.
 - API PR [83671](#)
 - Start of library helpers PR [83665](#)
- CRD work continues
 - Keep improving Kube-Builder (dedicated working group)
 - [Immutable fields](#)
 - Binary transport format being investigated - no fixed date
- [Storage Migrator to GA](#)
- [28 Open KEPs tagging API Machinery](#)

Meetings and Working Groups



KubeCon



CloudNativeCon

North America 2019

Regular SIG meetings:

- SIG Meeting: 60 min / every 2 weeks
- PR and Bug triage: 30 min / twice every week (Tue - Thu)

Regular Working Group meetings:

- Working Group Apply: 30 min / every 2 weeks
 - Working Group Kubebuilder and SDK: 60 min / monthly meeting
1. *Come join us, just listen, see what we do, how we work, learn from the experts*
 2. *There will be more organized ramping up programs are coming in 2020*



Introduction to SIG API-Machinery



KubeCon



CloudNativeCon

North America 2019

Chairs/Leads

- [@deads2k](#) (Co-Chair and Tech Lead, Red Hat)
- [@lavalamp](#) (Tech Lead and *emeritus* Co-Chair, 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>
- Slack channel: <https://kubernetes.slack.com/messages/sig-api-machinery>
- Youtube Playlist: <https://www.youtube.com/playlist?list=PL69nYSiGNLP21oW3hbLyjji4XhrwKxH2R>
- Mail Group: <https://groups.google.com/forum/#!forum/kubernetes-sig-api-machinery>



Thank you!