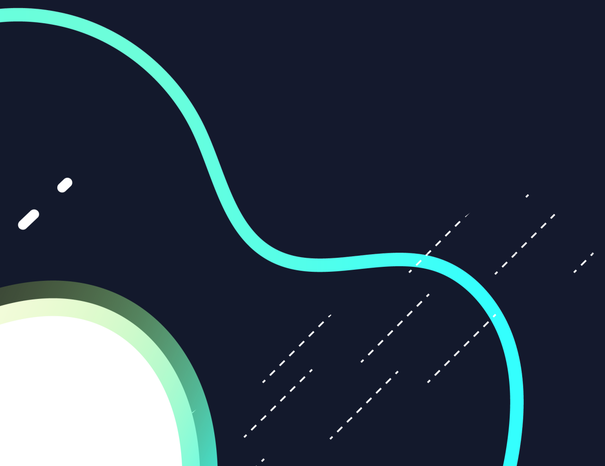
KUBERNETES UNDER THE HOOD

Richard Szabo
Duelbox
Co-Founder, Software Engineer

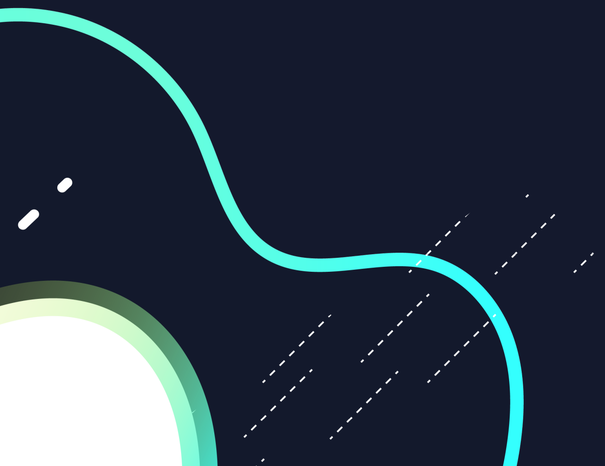
WHY BOTHER?



DUELBOX

WHY BOTHER?

Understand the basics of the infrastructure

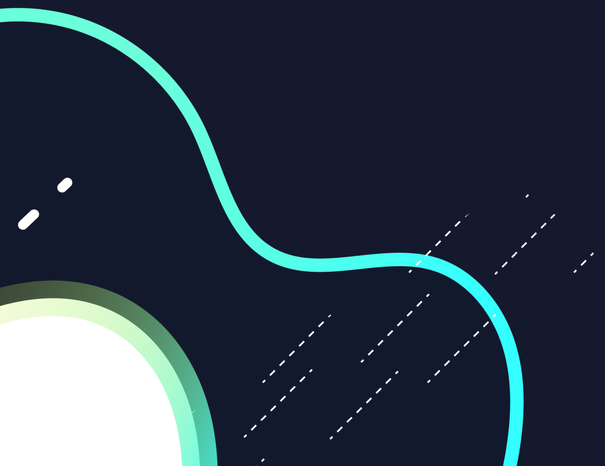


DUELBOX

WHY BOTHER?

Understand the basics of the infrastructure

Debugging / fixing issues



DUELBOX

WHY BOTHER?

Understand the basics of the infrastructure

Debugging / fixing issues

Helpful in design decisions

WHY BOTHER?

Understand the basics of the infrastructure

Debugging / fixing issues

Helpful in design decisions

Learn from patterns of a complex distributed system

WHY BOTHER?

Understand the basics of the infrastructure

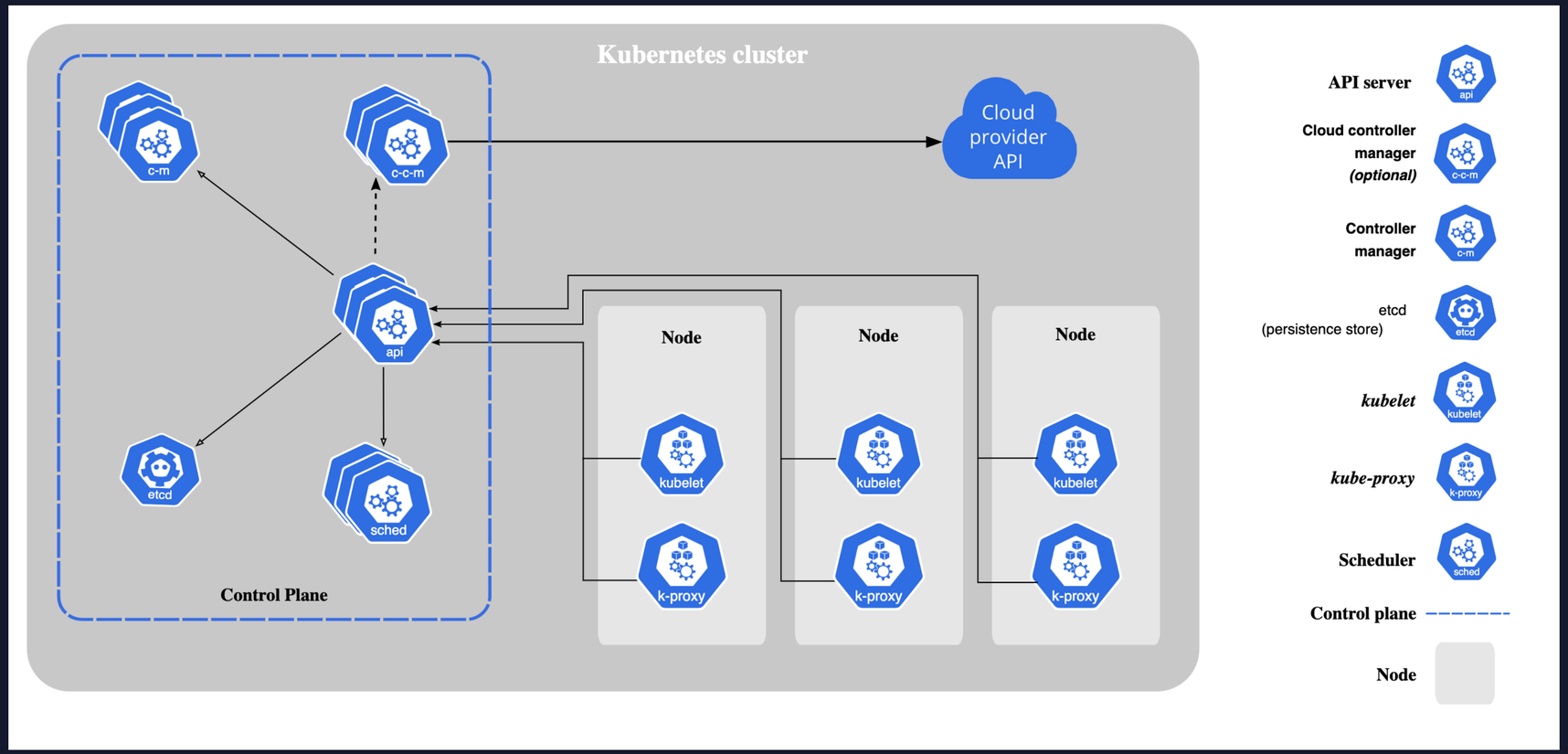
Debugging / fixing issues

Helpful in design decisions

Learn from patterns of a complex distributed system

Extensions, CRD

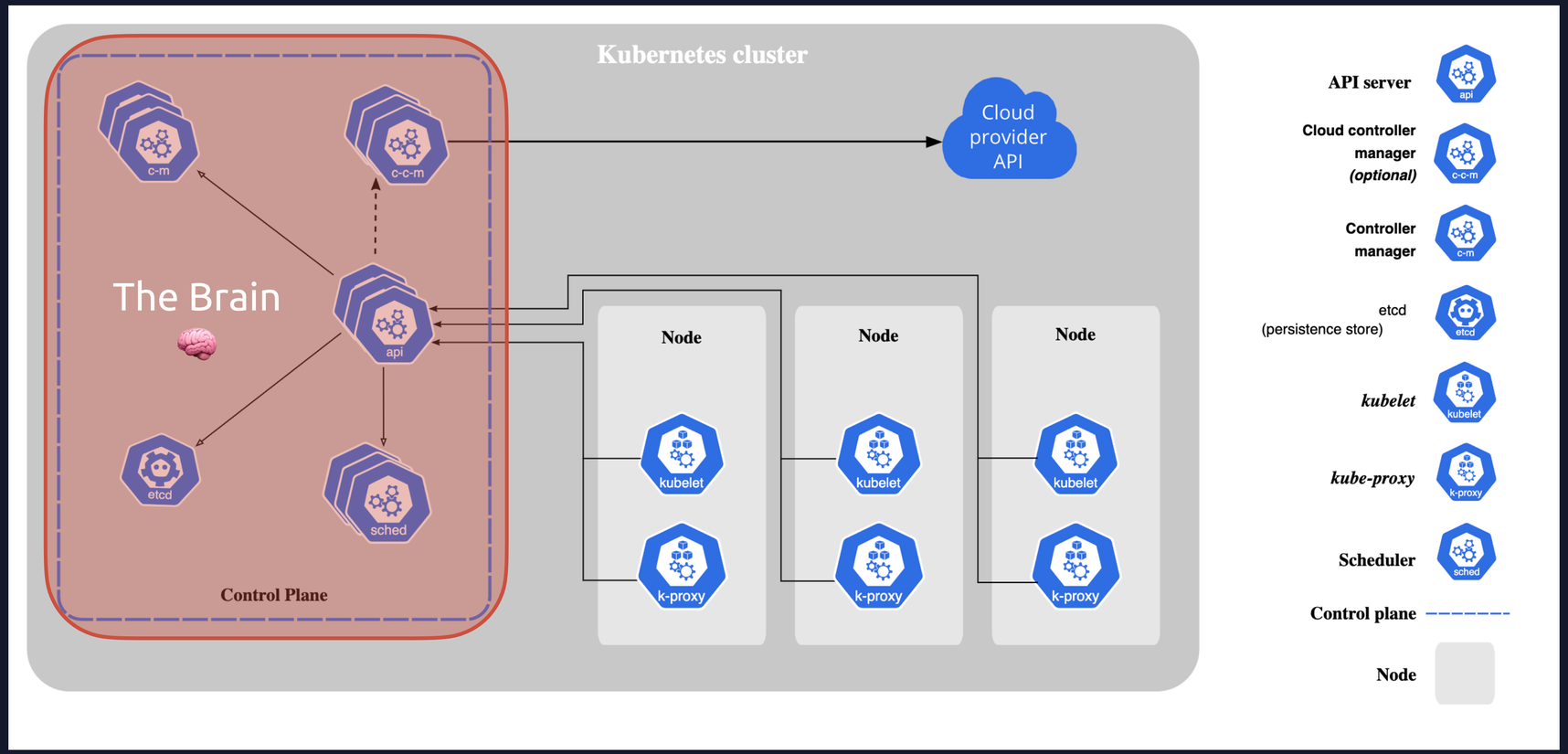
KUBERNETES ARCHITECTURE



https://kubernetes.io/docs/concepts/overview/components/

DUELBOX

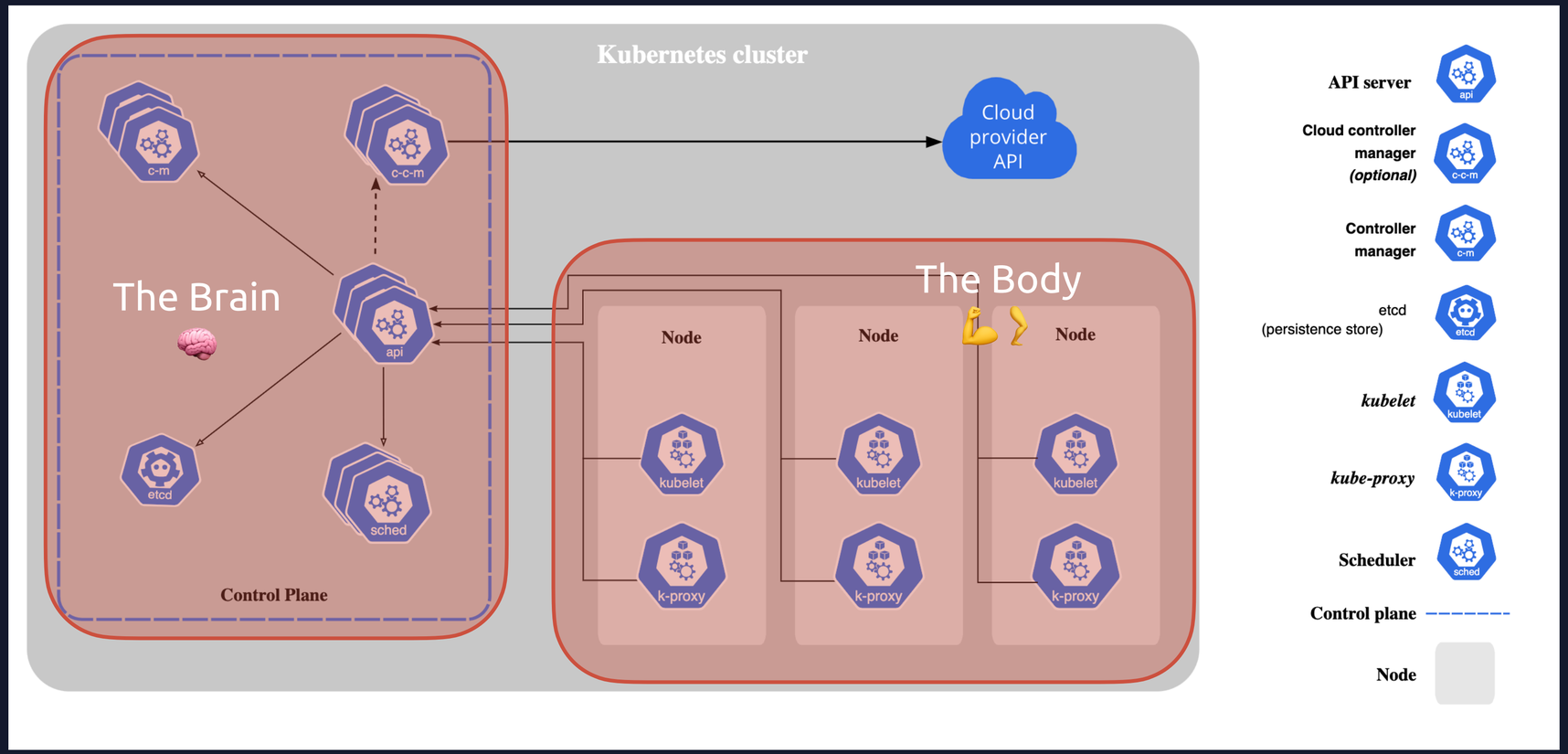
KUBERNETES ARCHITECTURE



https://kubernetes.io/docs/concepts/overview/components/

DUELBOX

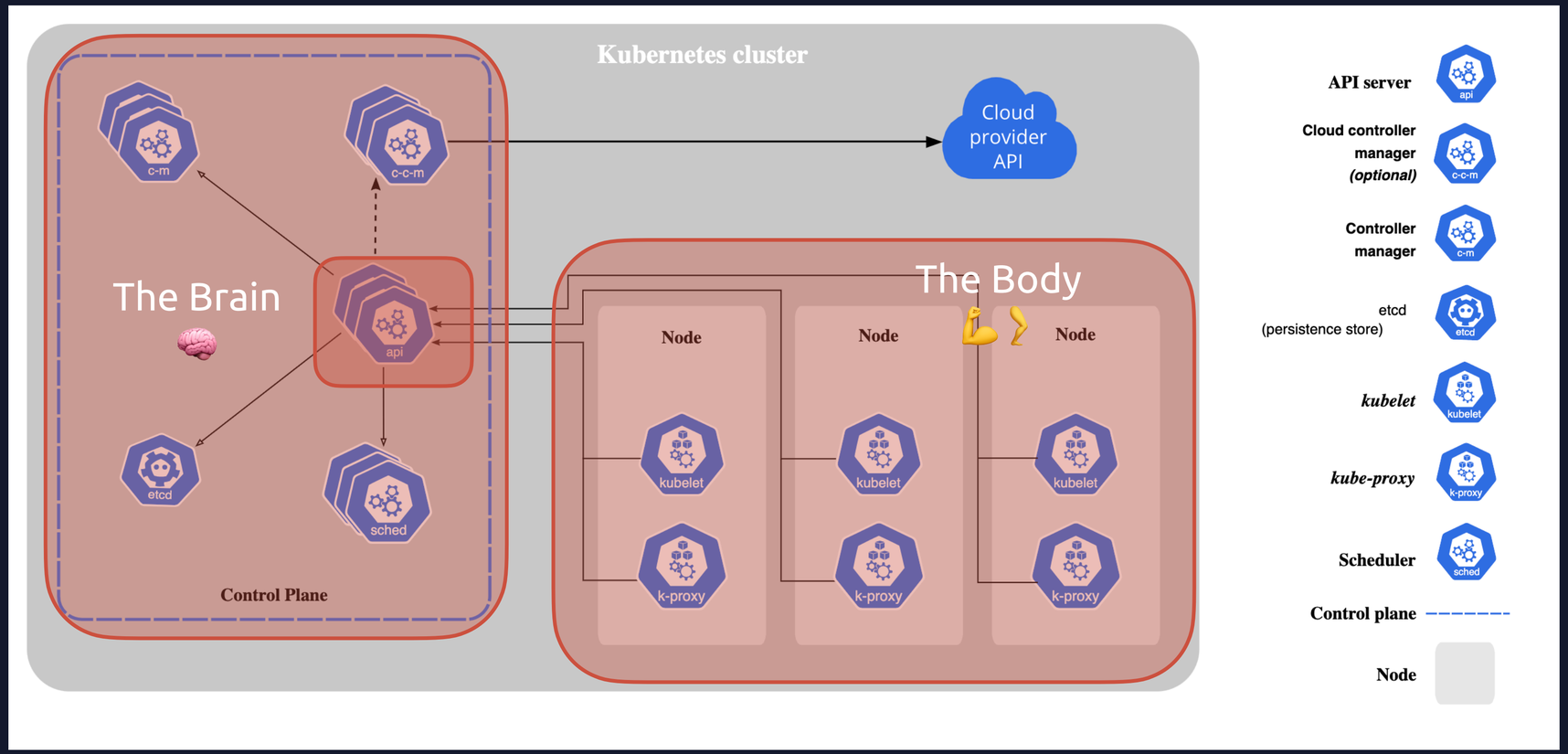
KUBERNETES ARCHITECTURE



https://kubernetes.io/docs/concepts/overview/components/

DUELBOX

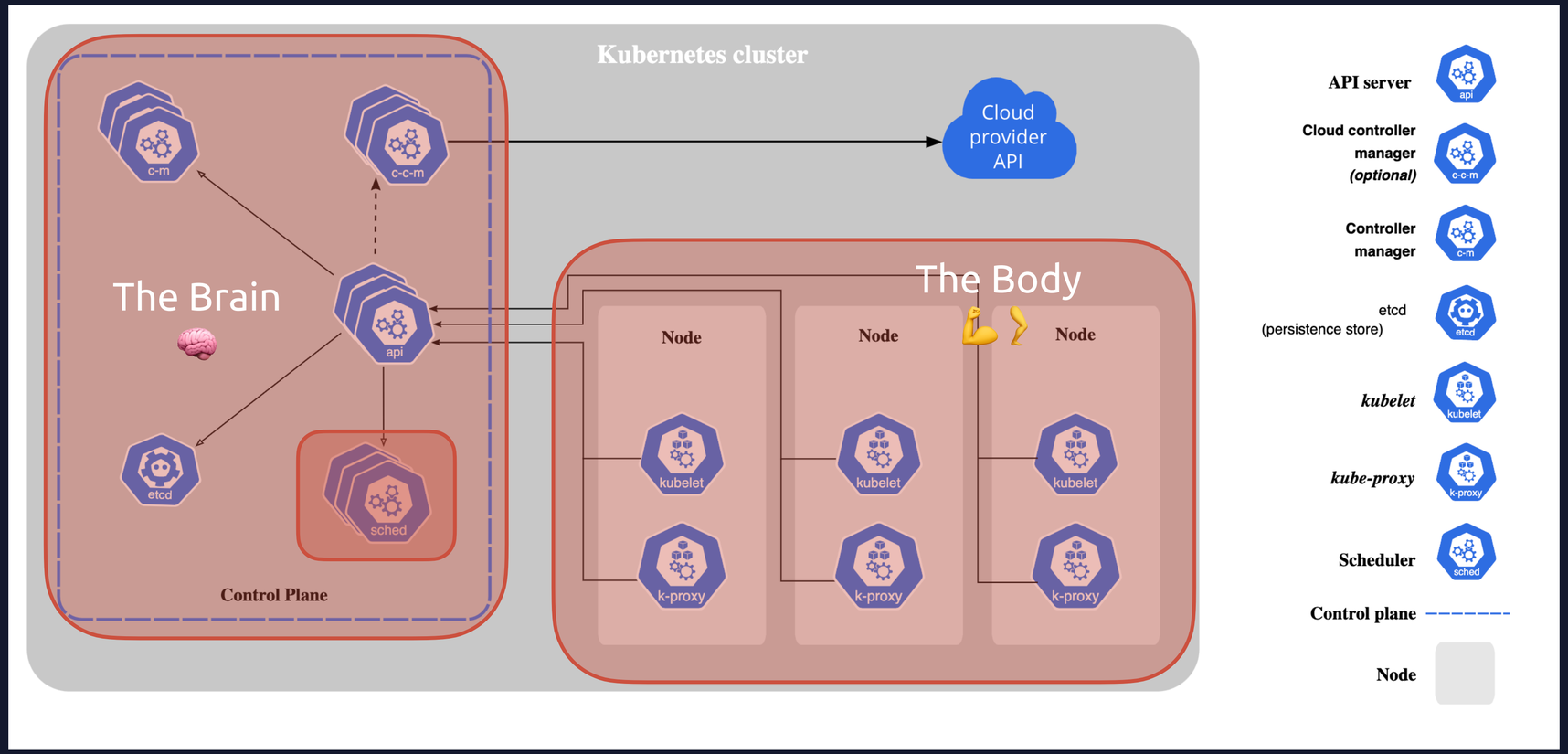
KUBERNETES ARCHITECTURE



https://kubernetes.io/docs/concepts/overview/components/

DUELBOX

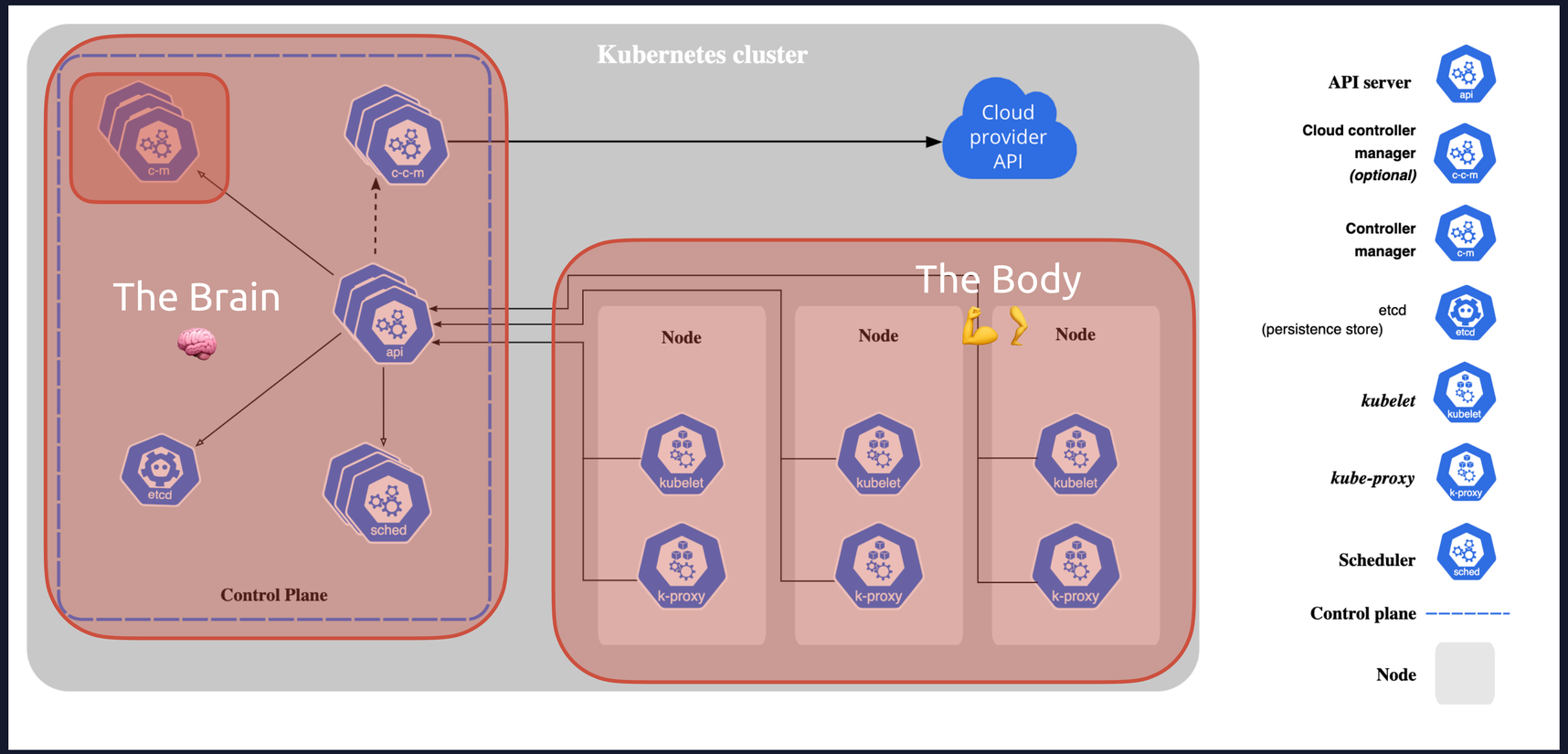
KUBERNETES ARCHITECTURE



https://kubernetes.io/docs/concepts/overview/components/

DUELBOX

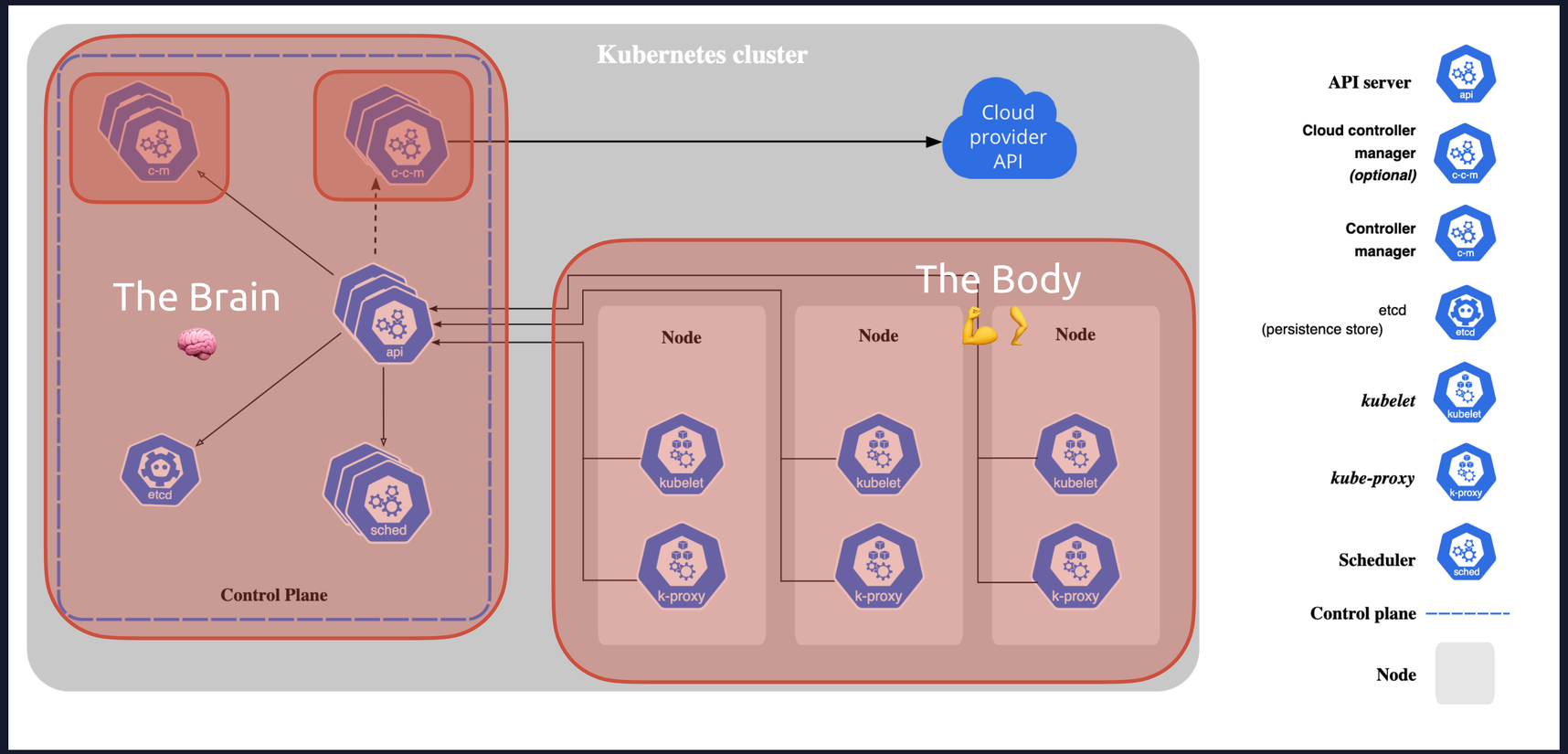
KUBERNETES ARCHITECTURE



https://kubernetes.io/docs/concepts/overview/components/

DUELBOX

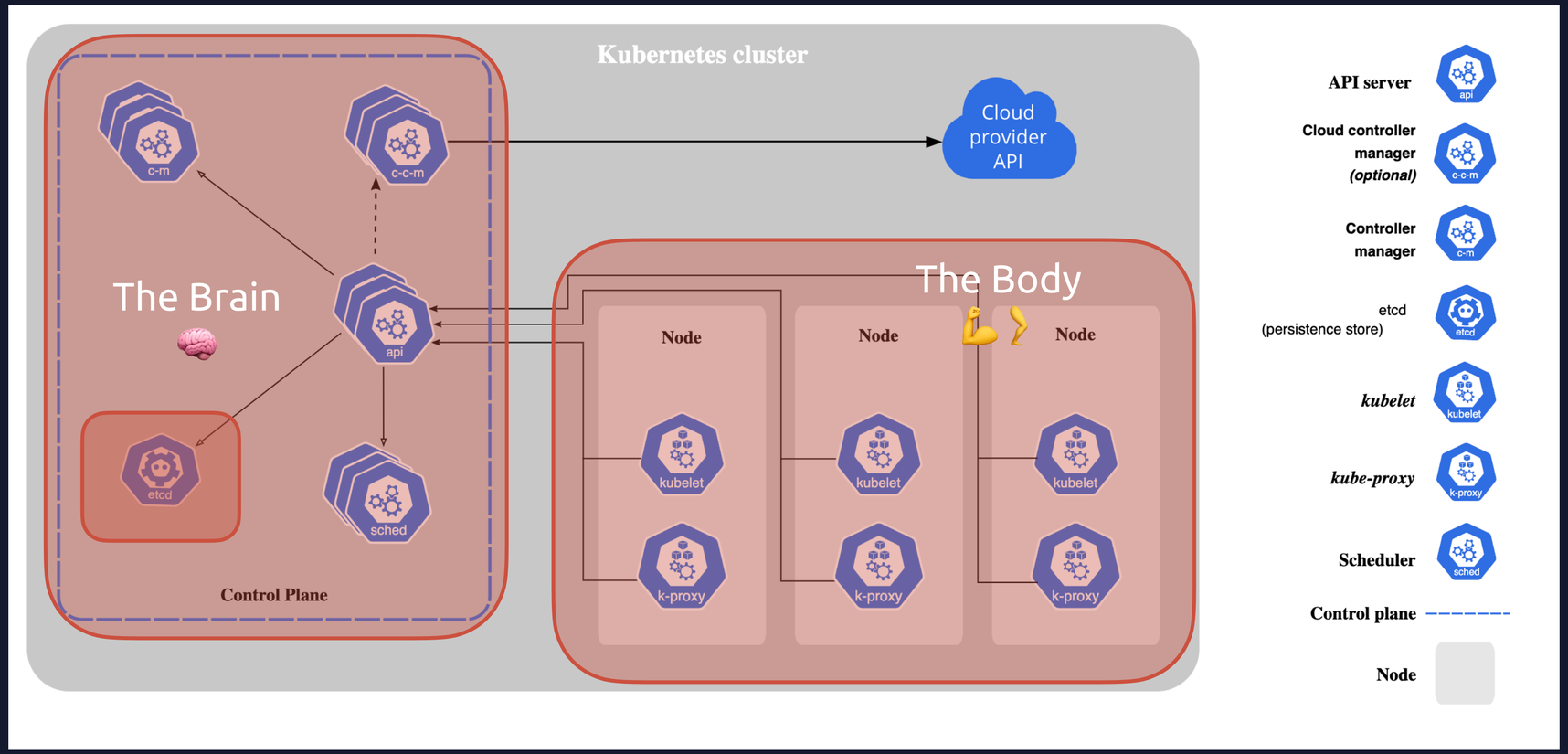
KUBERNETES ARCHITECTURE



https://kubernetes.io/docs/concepts/overview/components/

DUELBOX

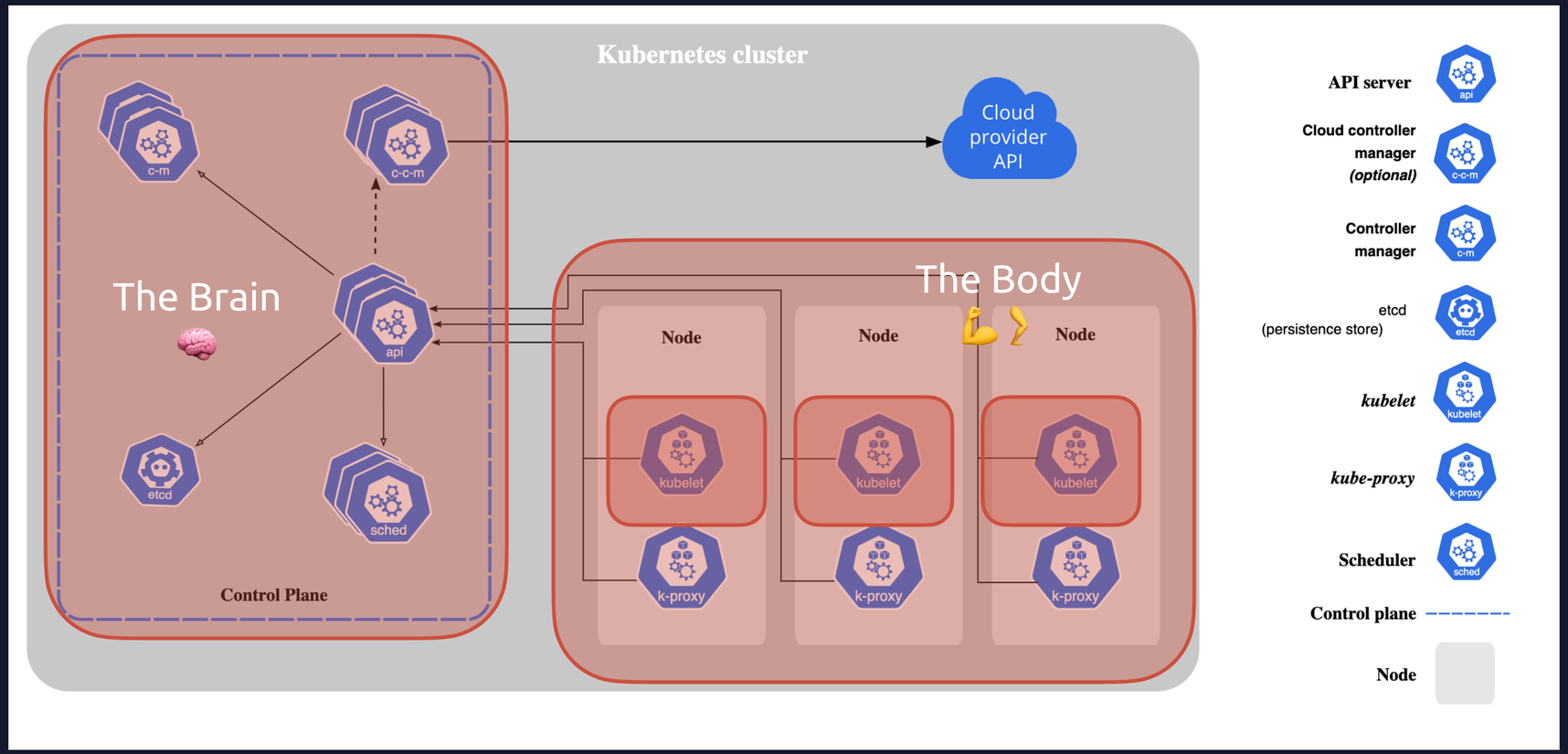
KUBERNETES ARCHITECTURE



https://kubernetes.io/docs/concepts/overview/components/

DUELBOX

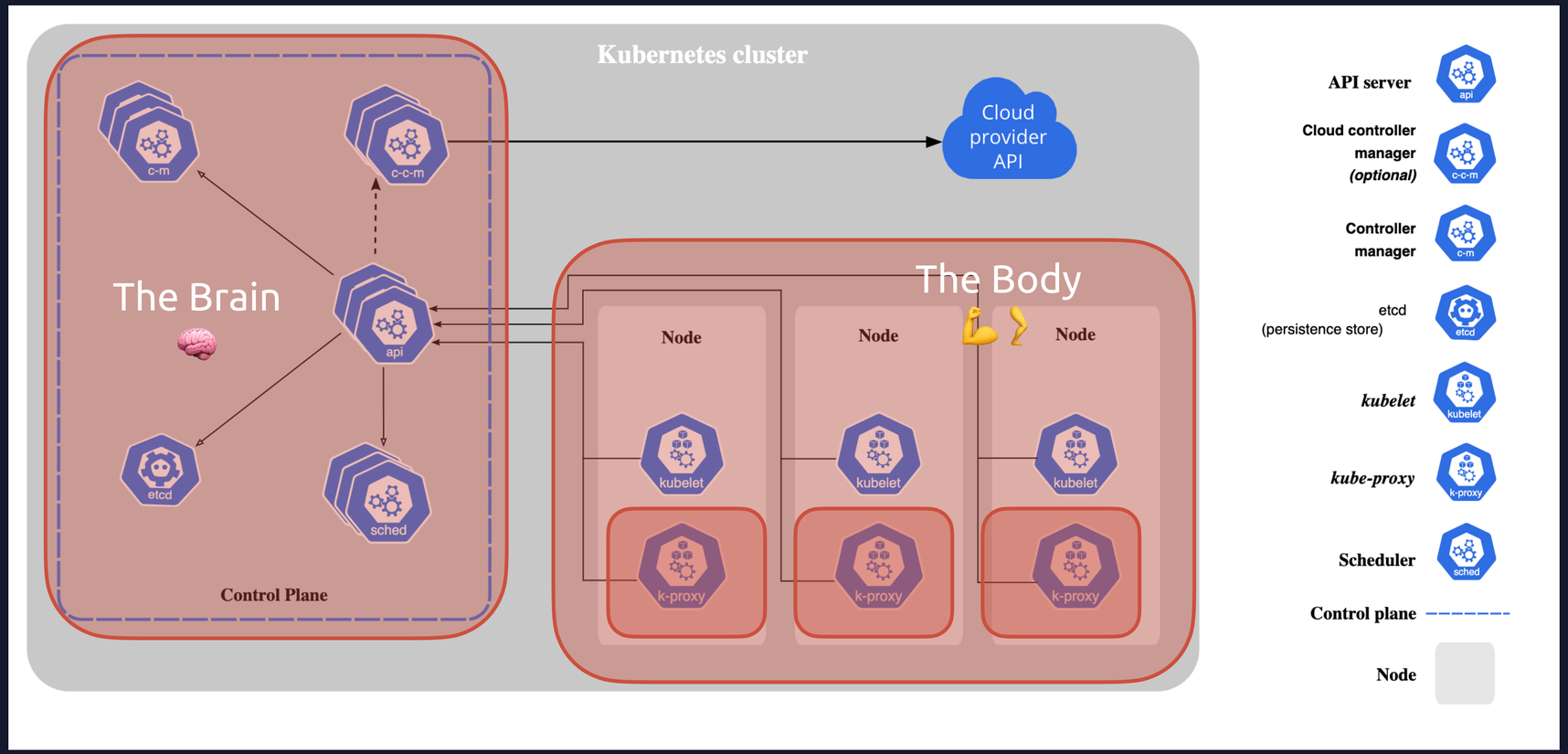
KUBERNETES ARCHITECTURE



https://kubernetes.io/docs/concepts/overview/components/

DUELBOX

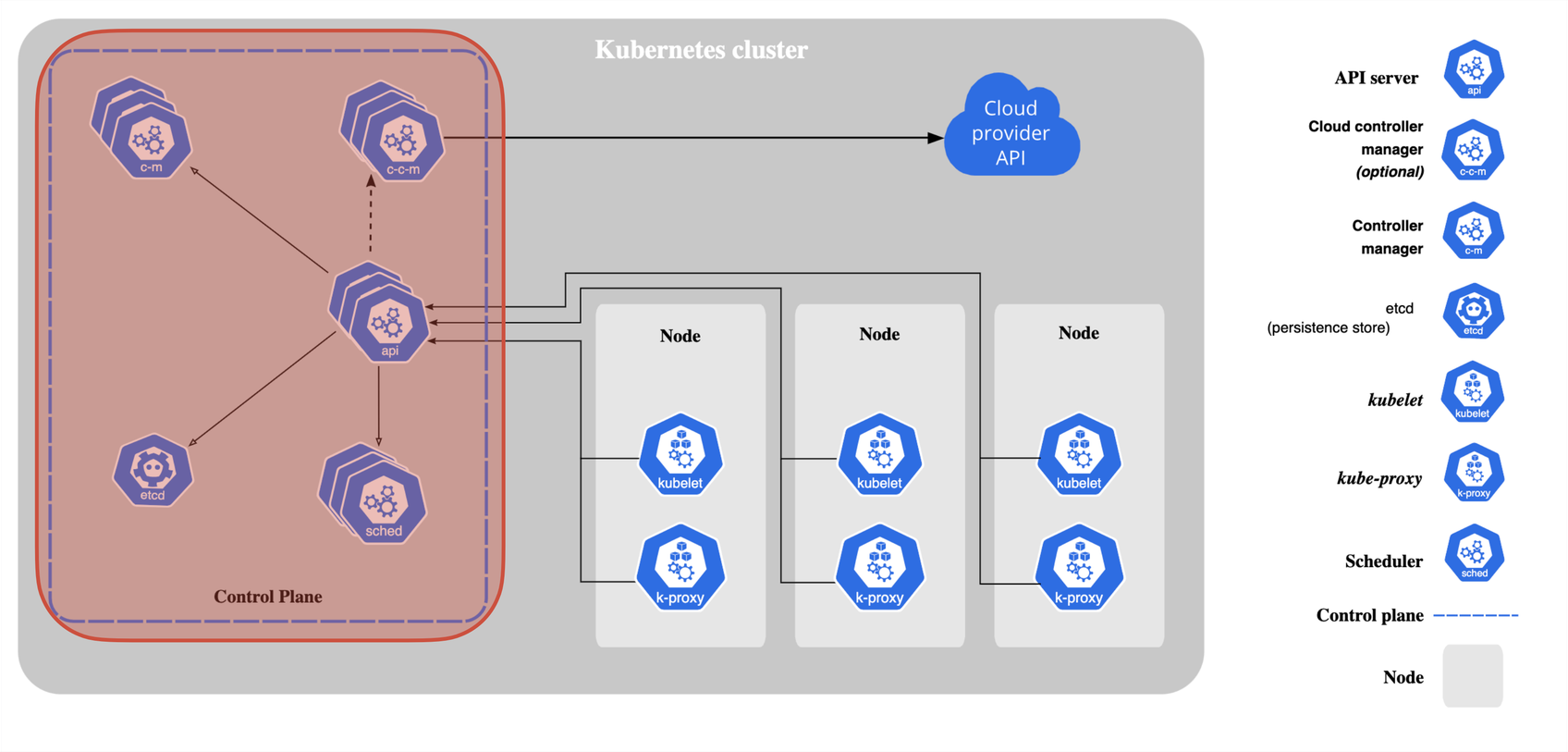
KUBERNETES ARCHITECTURE



https://kubernetes.io/docs/concepts/overview/components/

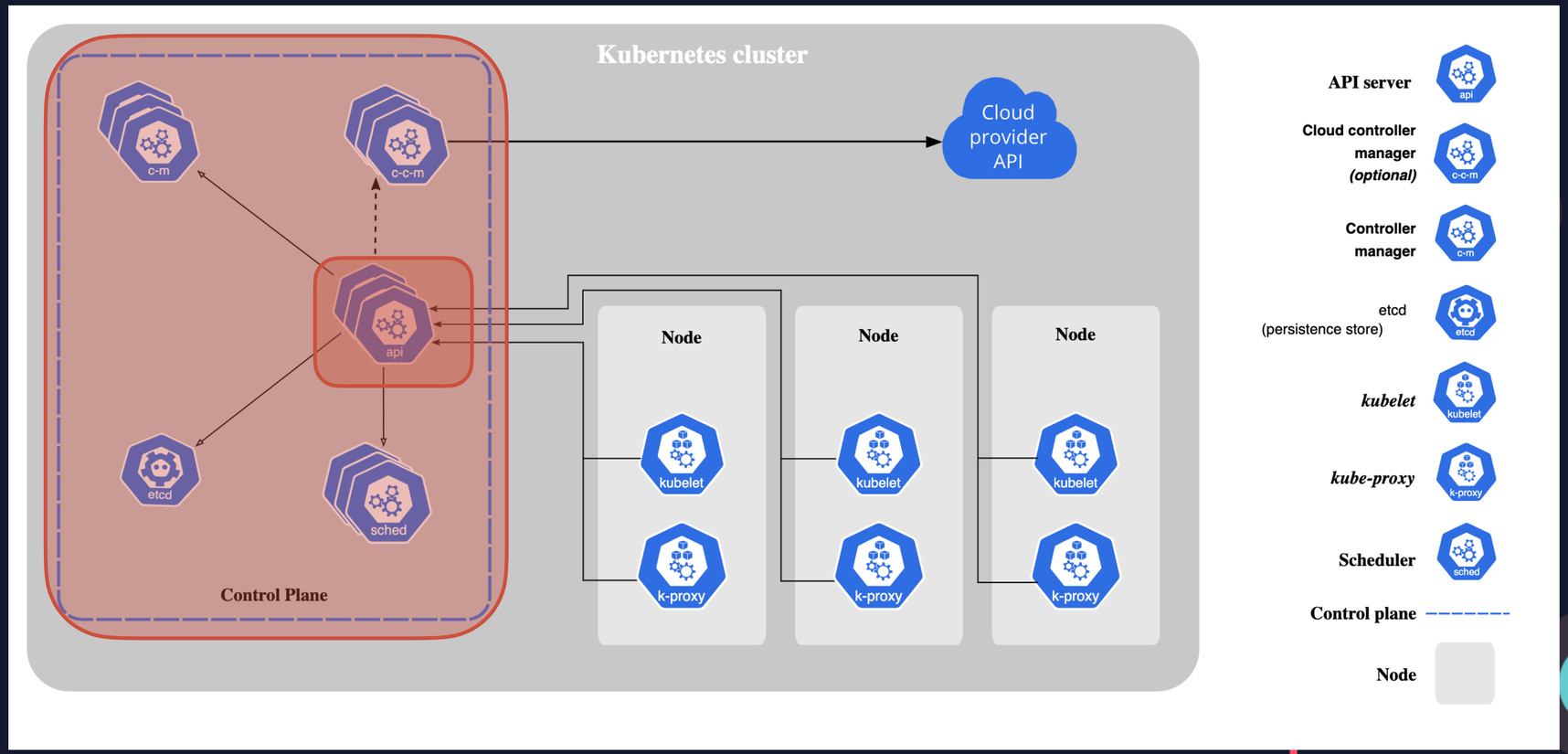
DUELBOX

CONTROL PLANE



DUELBOX

API SERVER

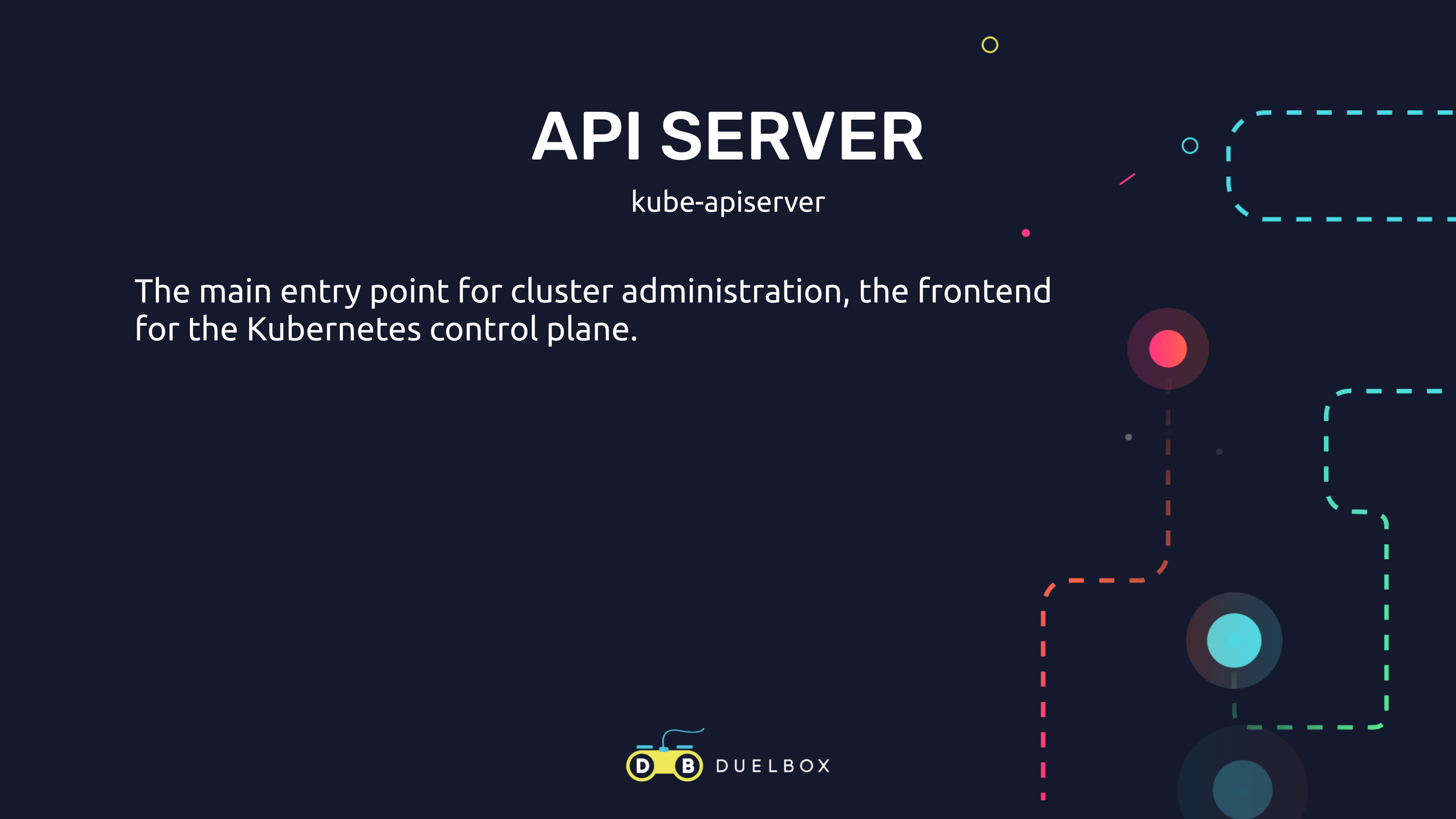


DUELBOX

API SERVER

kube-apiserver

The main entry point for cluster administration, the frontend for the Kubernetes control plane.



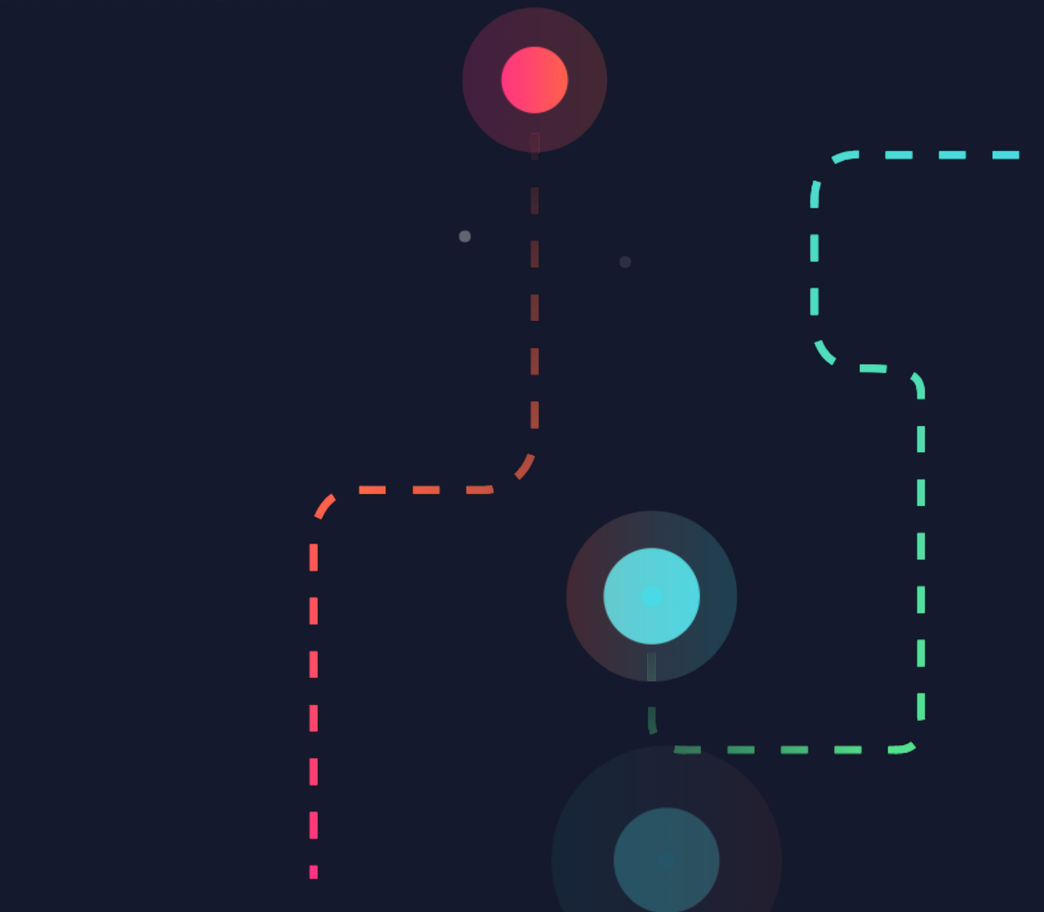
DUELBOX

API SERVER

kube-apiserver

The main entry point for cluster administration, the frontend for the Kubernetes control plane.

REST API



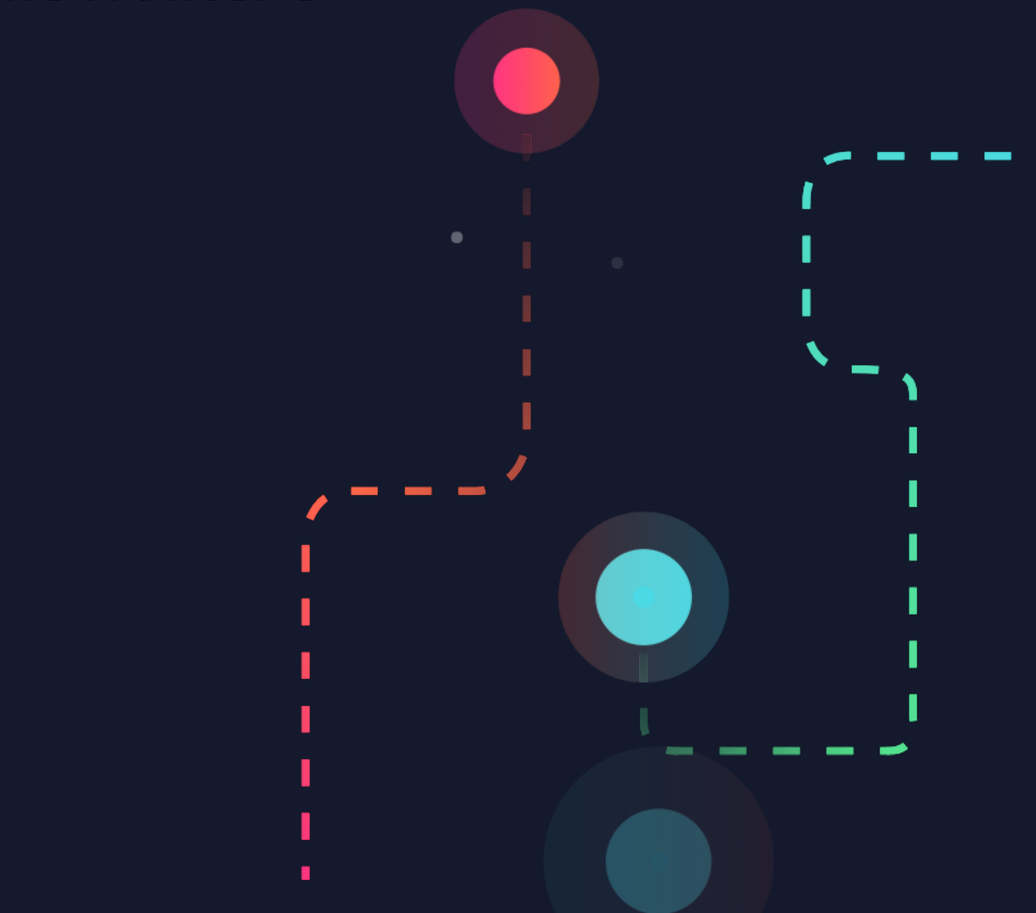
API SERVER

kube-apiserver

The main entry point for cluster administration, the frontend for the Kubernetes control plane.

REST API

Request validation



DUELBOX

API SERVER

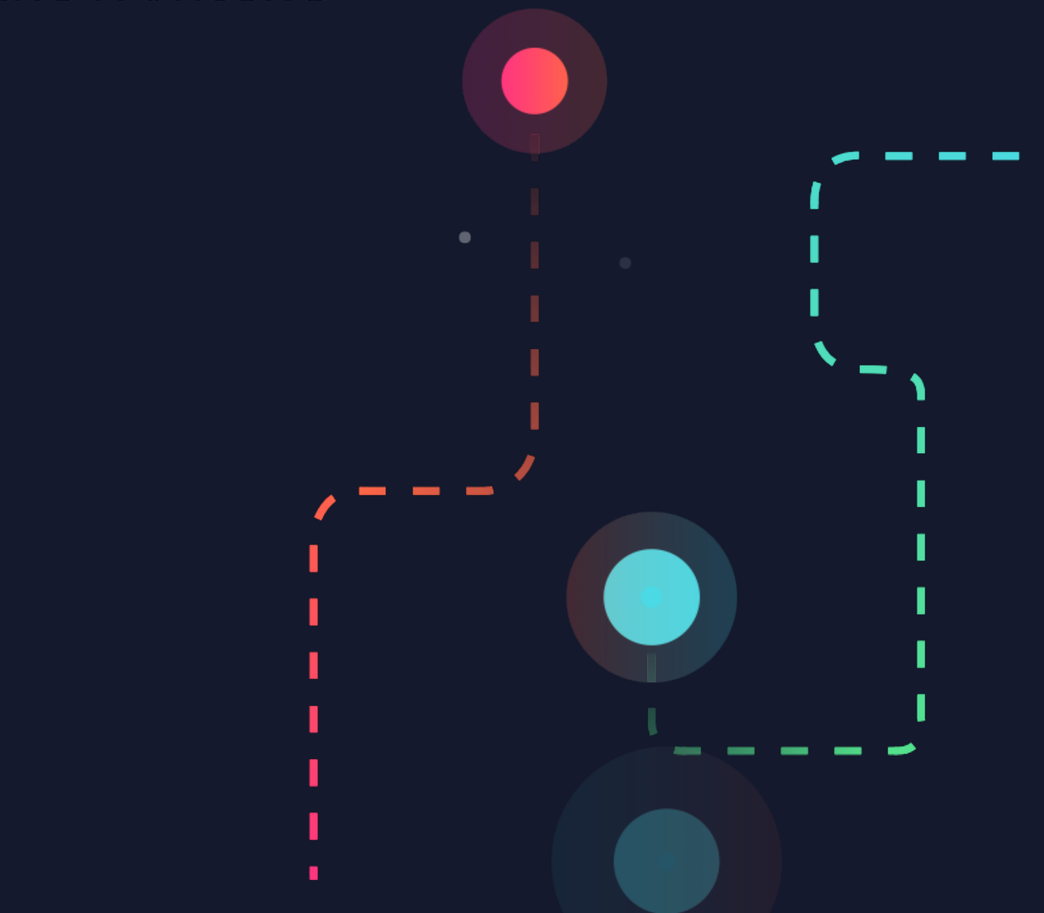
kube-apiserver

The main entry point for cluster administration, the frontend for the Kubernetes control plane.

REST API

Request validation

Scaled horizontally



API SERVER

kube-apiserver

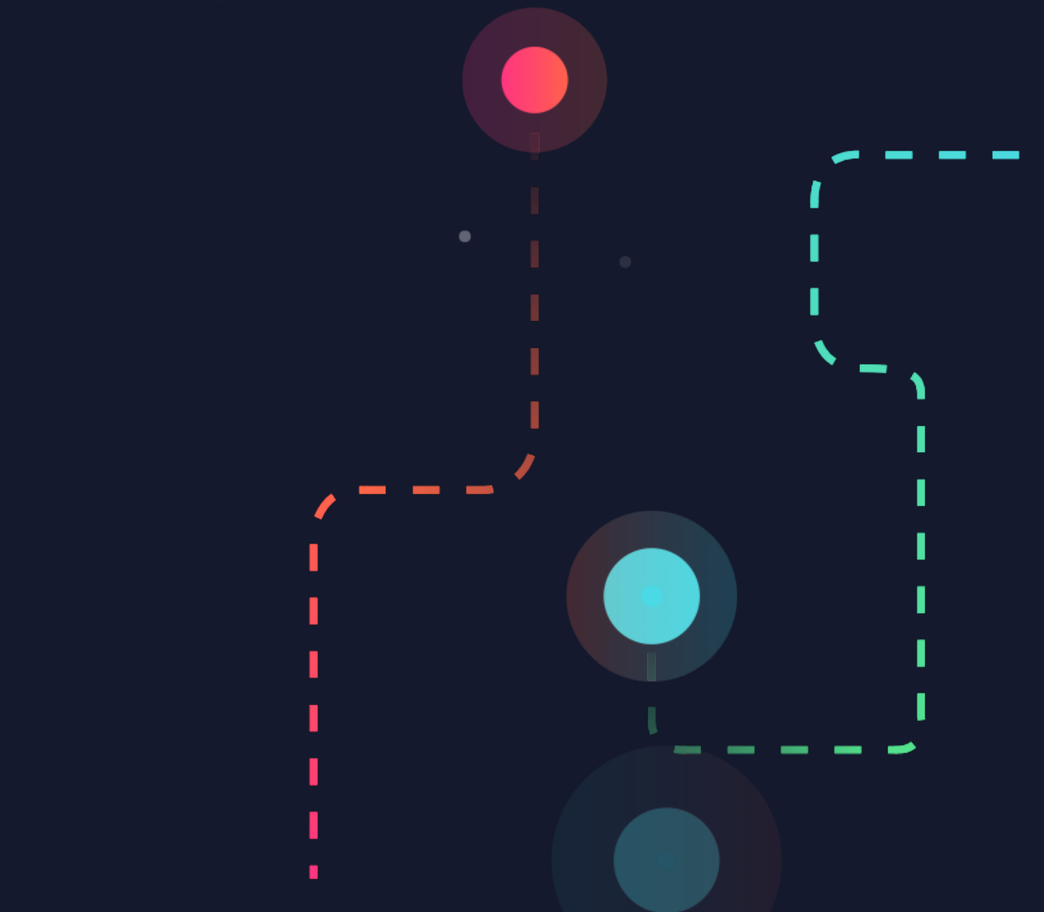
The main entry point for cluster administration, the frontend for the Kubernetes control plane.

REST API

Request validation

Scaled horizontally

Cluster endpoint with load balancer



DUELBOX

API SERVER

kube-apiserver

The main entry point for cluster administration, the frontend for the Kubernetes control plane.

REST API

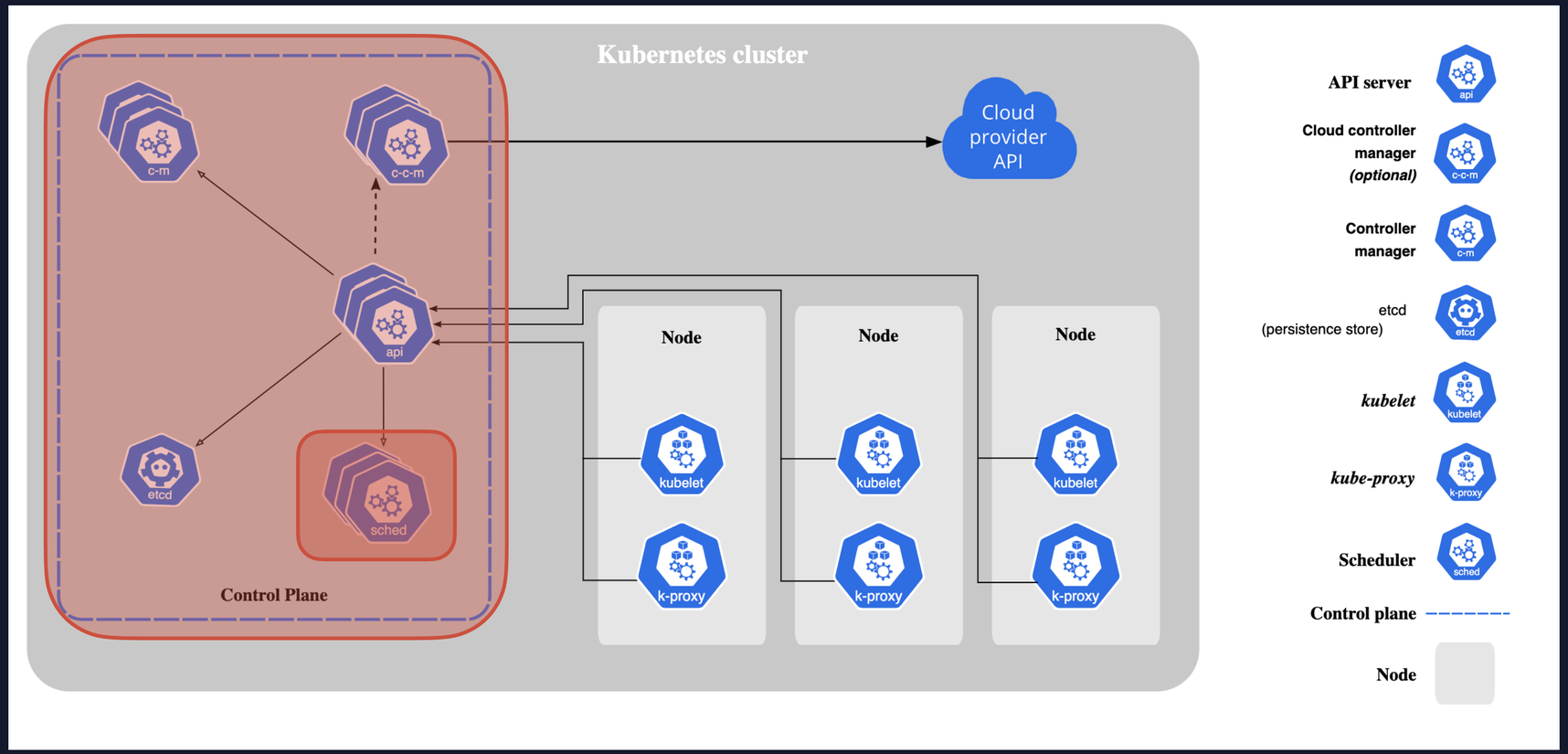
Request validation

Scaled horizontally

Cluster endpoint with load balancer

kubectl

SCHEDULER



DUELBOX

SCHEDULER

kube-scheduler

DUELBOX

SCHEDULER

kube-scheduler

Schedules Pods onto *appropriate* nodes of the cluster.

DUELBOX

SCHEDULER

kube-scheduler

Schedules Pods onto *appropriate* nodes of the cluster.

Matches supply and demand:

SCHEDULER

kube-scheduler

Schedules Pods onto *appropriate* nodes of the cluster.

Matches supply and demand:

Resource requirements / limits (CPU, memory)

SCHEDULER

kube-scheduler

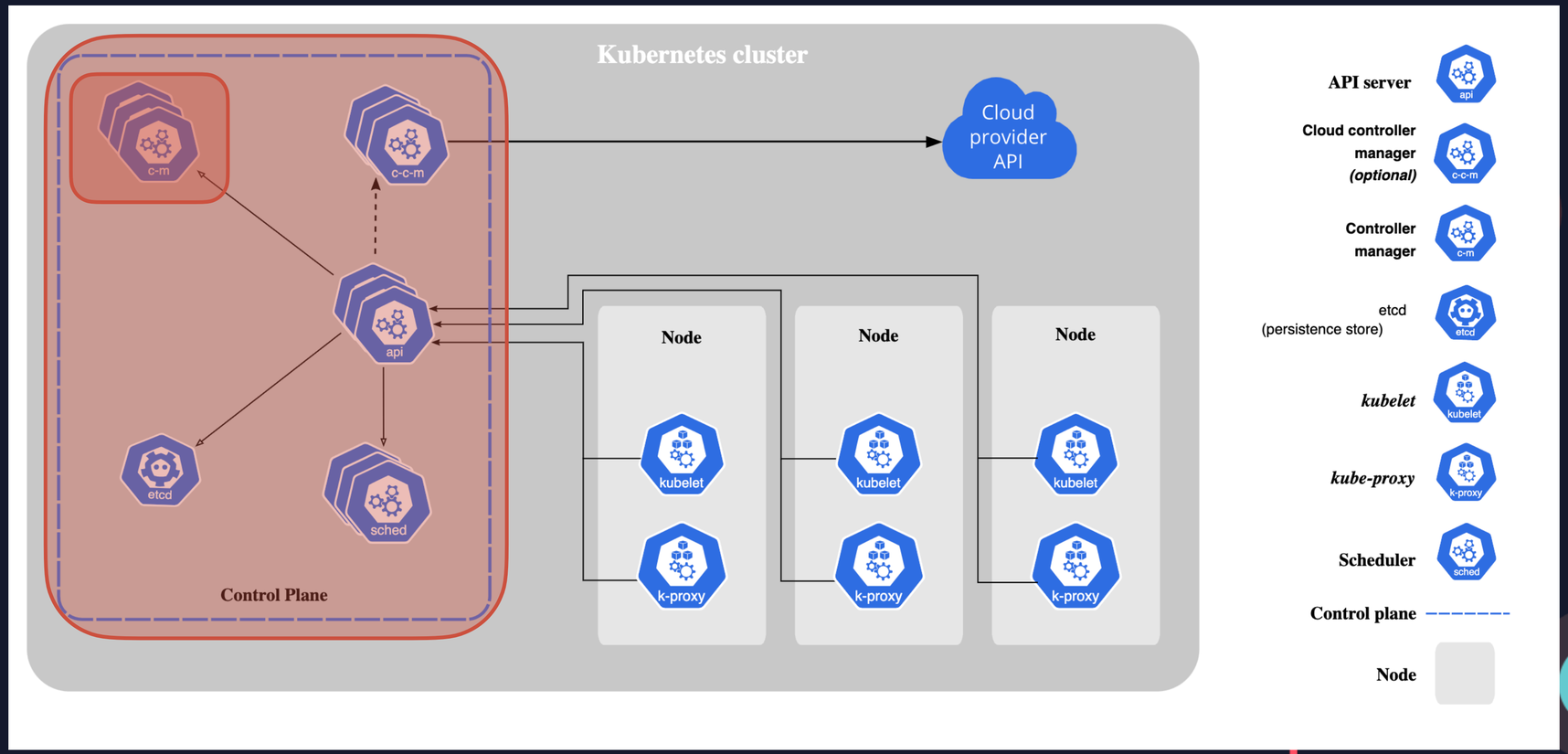
Schedules Pods onto *appropriate* nodes of the cluster.

Matches supply and demand:

Resource requirements / limits (CPU, memory)

nodeSelector / affinity / taints

CONTROLLER MANAGER

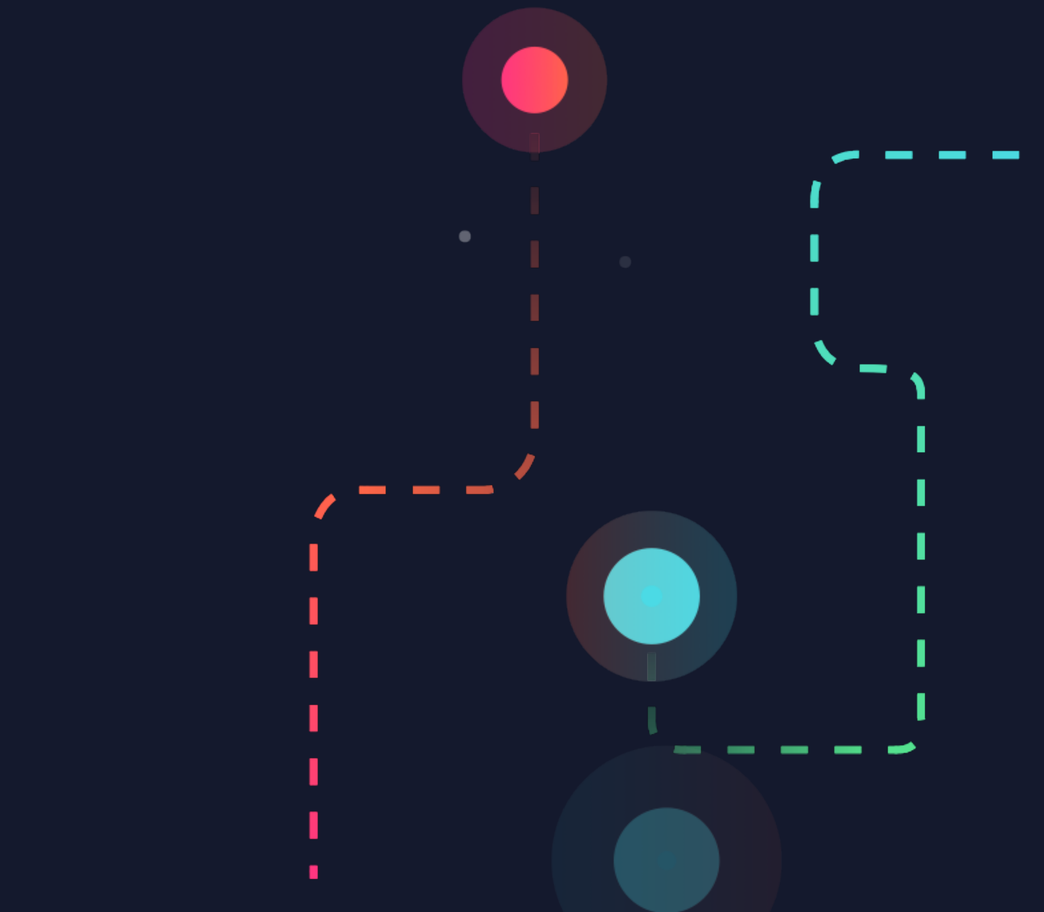


DUELBOX

CONTROLLER MANAGER

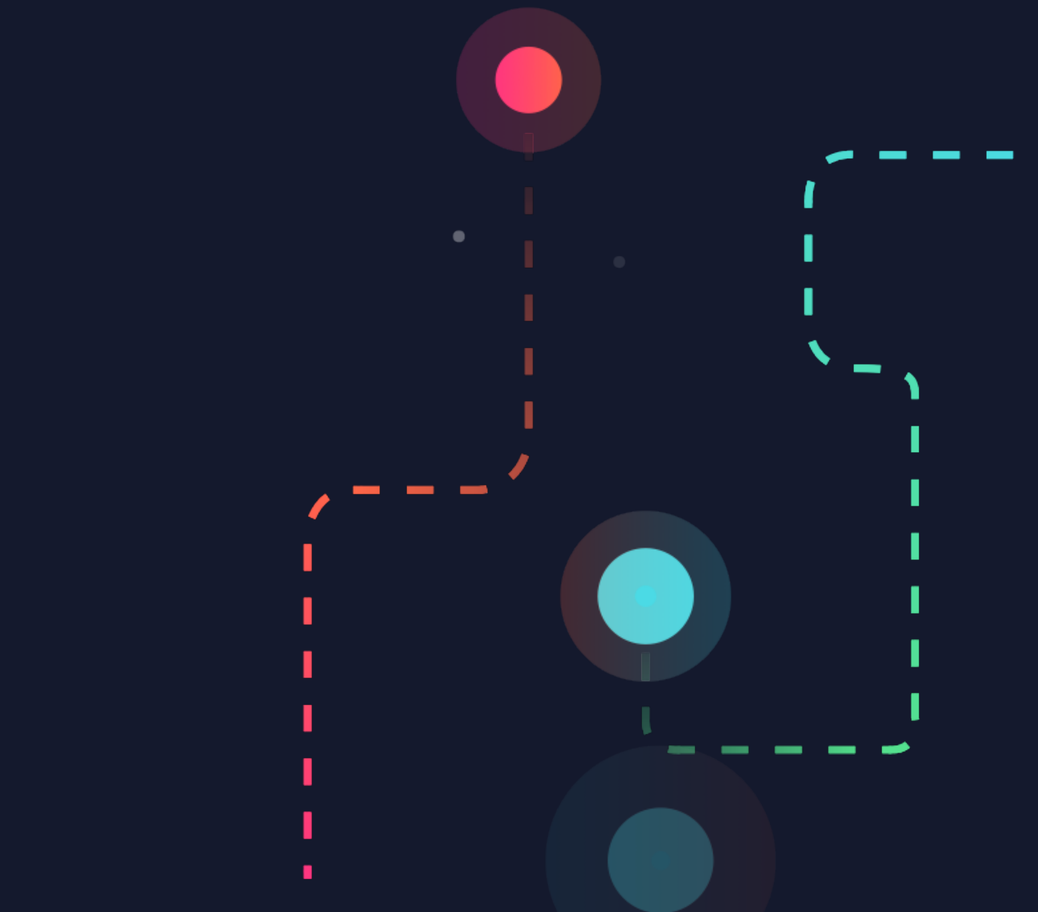
kube-controller-manager

Manages various controllers.



DUELBOX

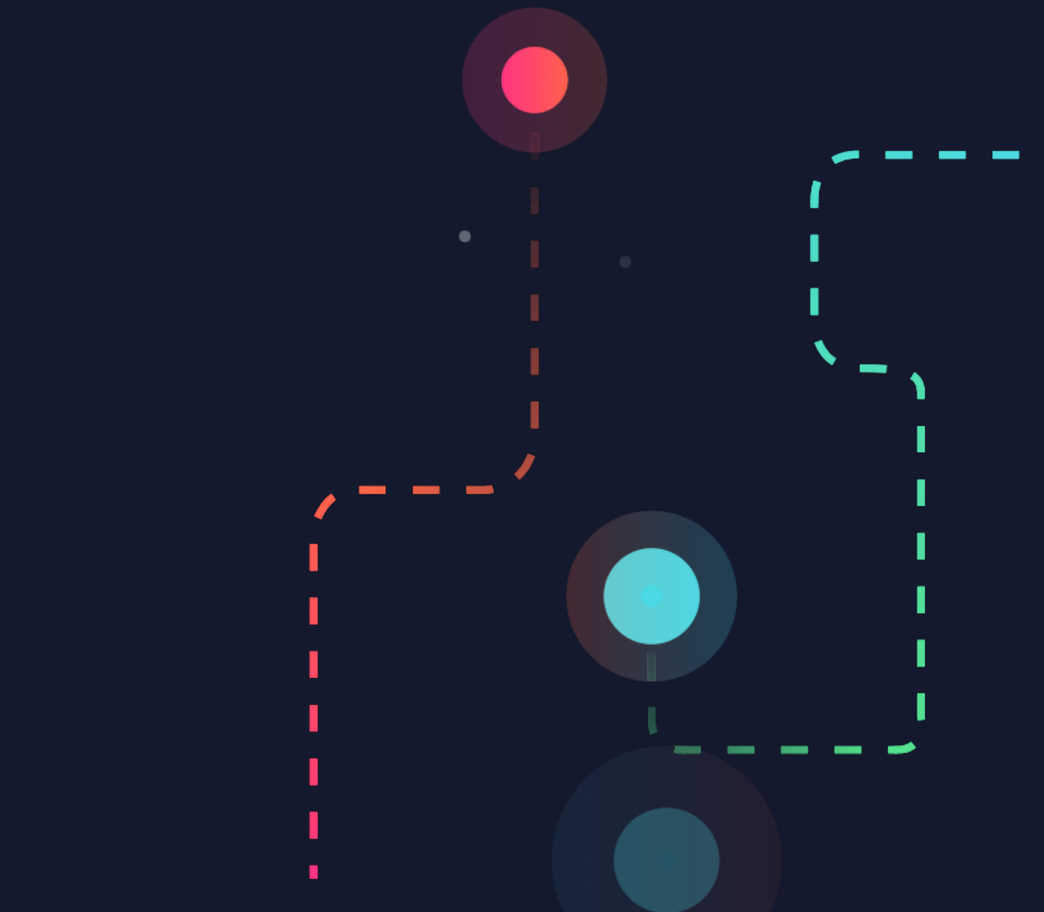
WHAT IS A CONTROLLER?



DUELBOX

WHAT IS A CONTROLLER?

Controllers are responsible for ensuring the components of the cluster are in the *desired* state.

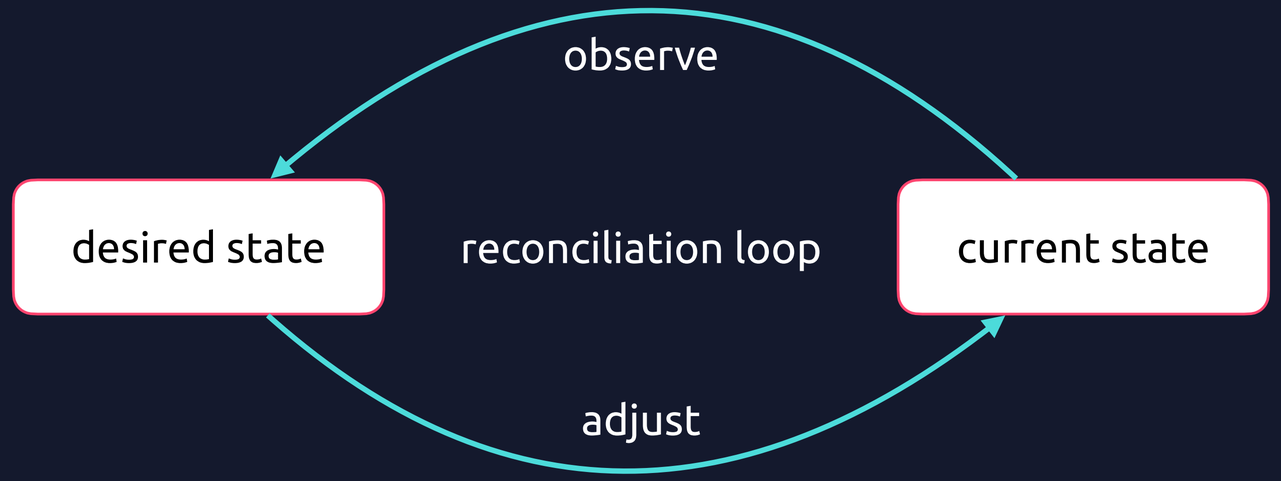


DUELBOX

WHAT IS A CONTROLLER?

Controllers are responsible for ensuring the components of the cluster are in the *desired* state.

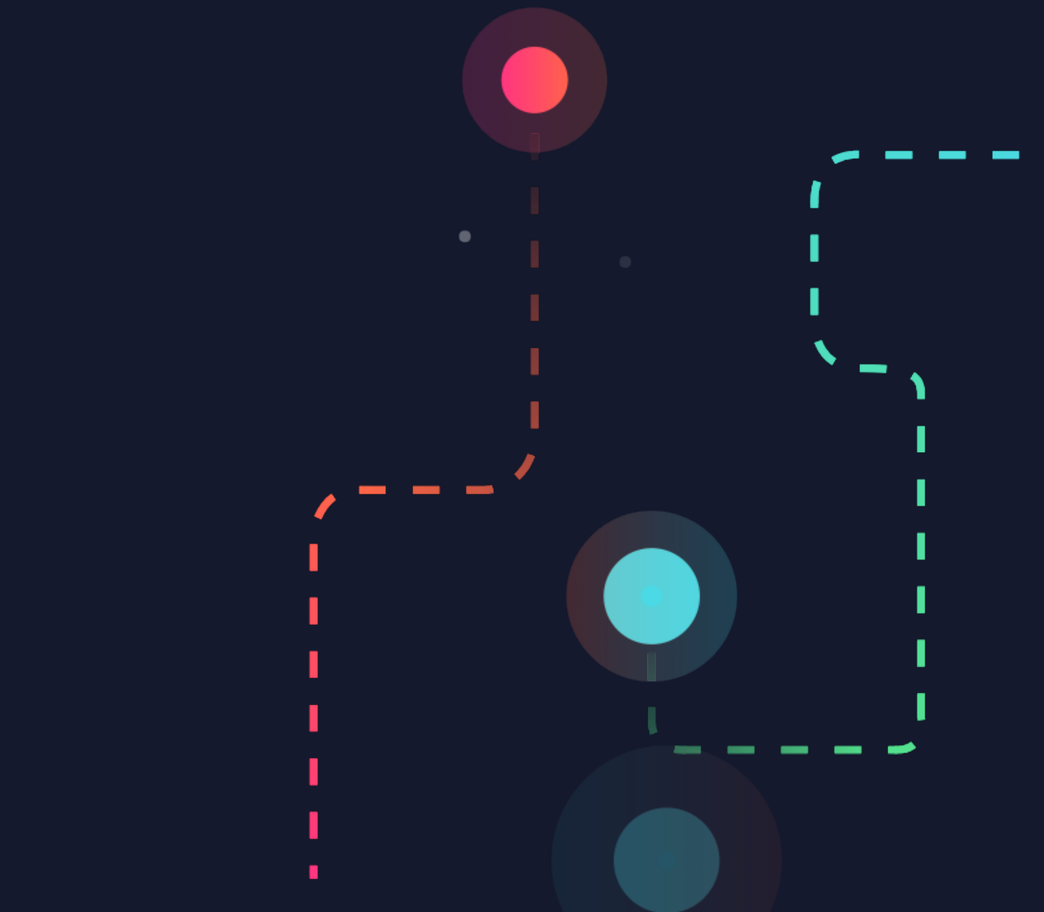
Reconciliation / control loop:



DUELBOX

CONTROLLER MANAGER

Manages various controllers.

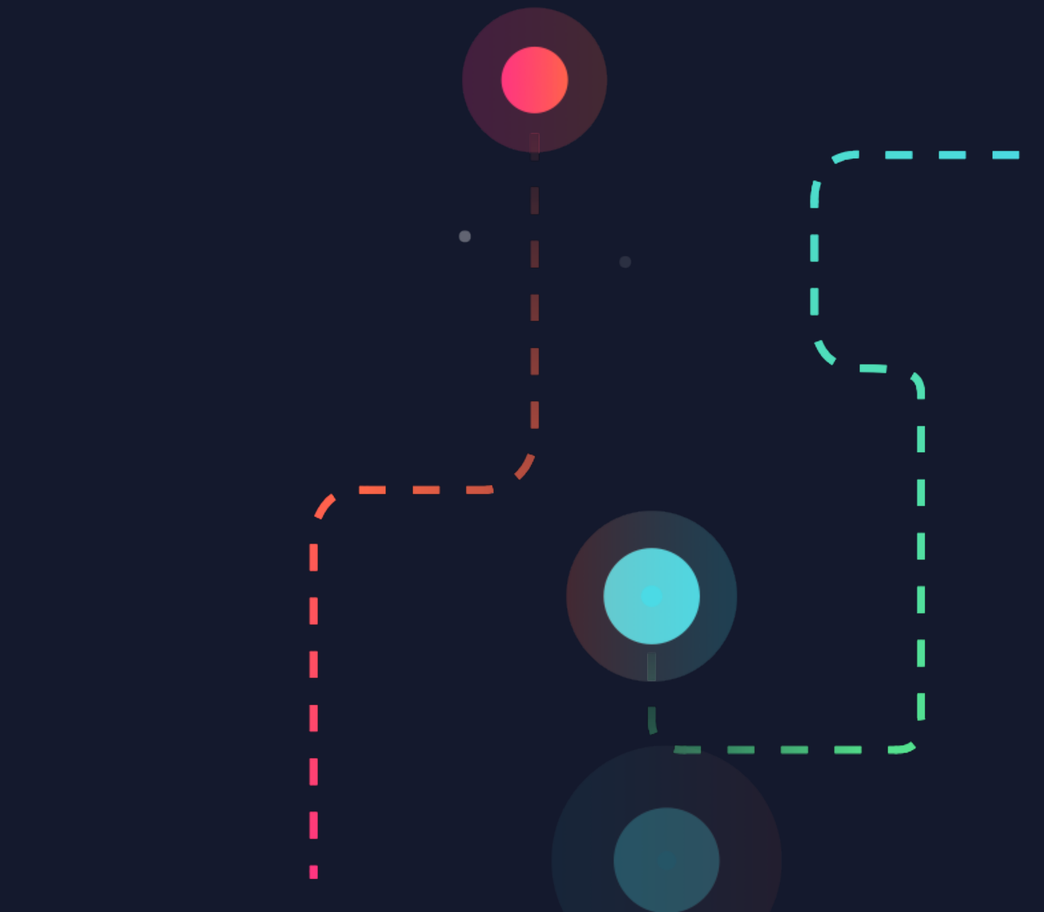


DUELBOX

CONTROLLER MANAGER

Manages various controllers.

ReplicaSet / Deployment controller

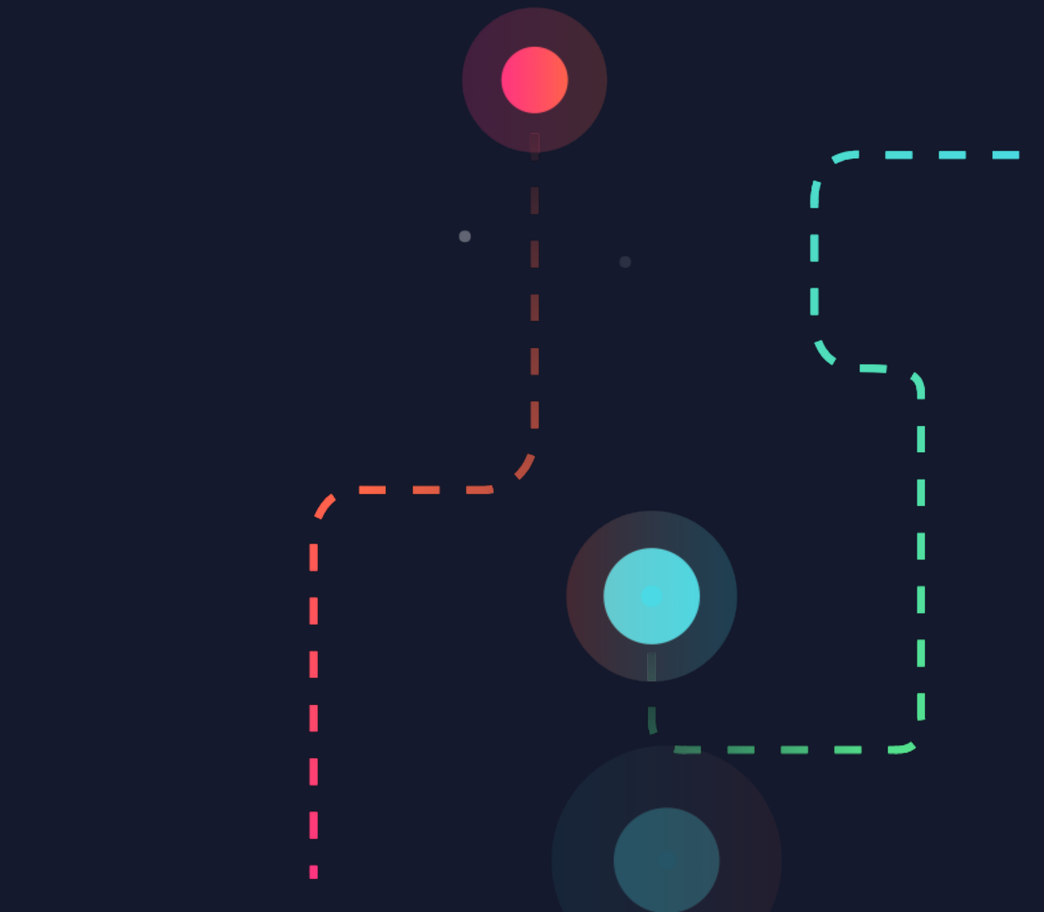


CONTROLLER MANAGER

Manages various controllers.

ReplicaSet / Deployment controller

Node controller



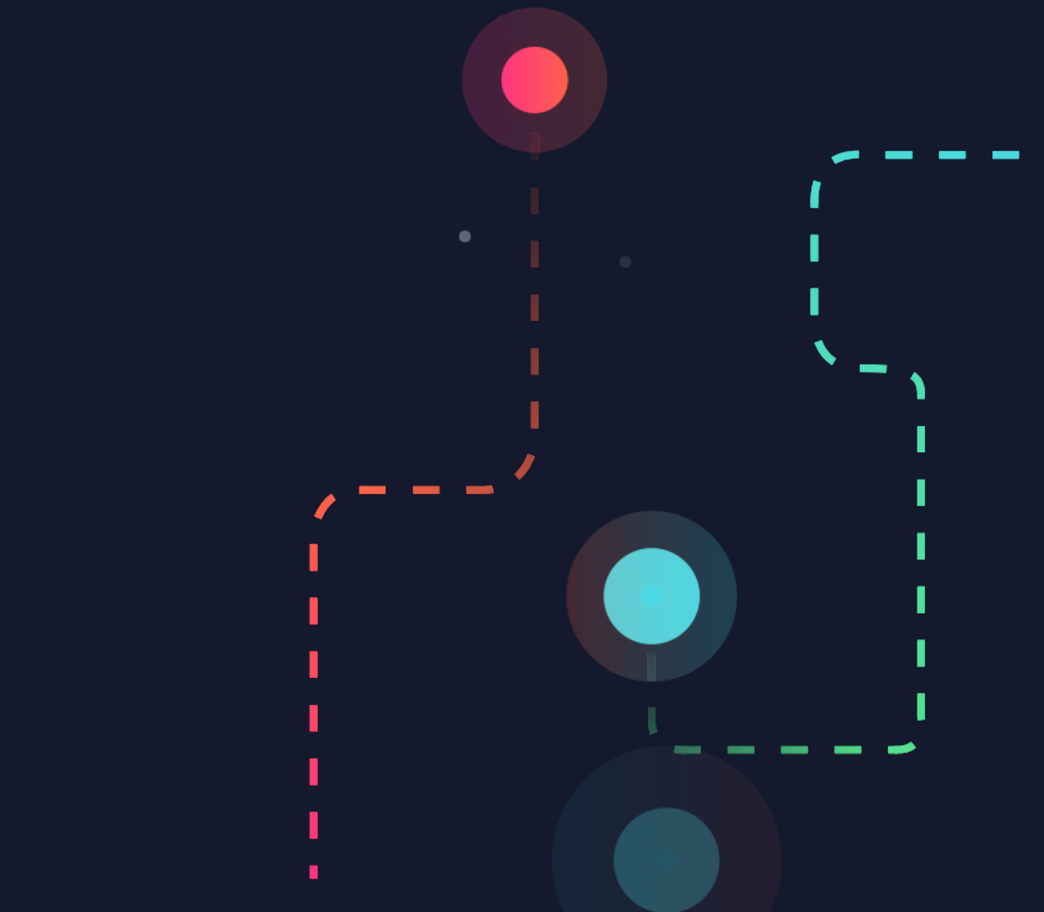
CONTROLLER MANAGER

Manages various controllers.

ReplicaSet / Deployment controller

Node controller

Endpoints controller



CONTROLLER MANAGER

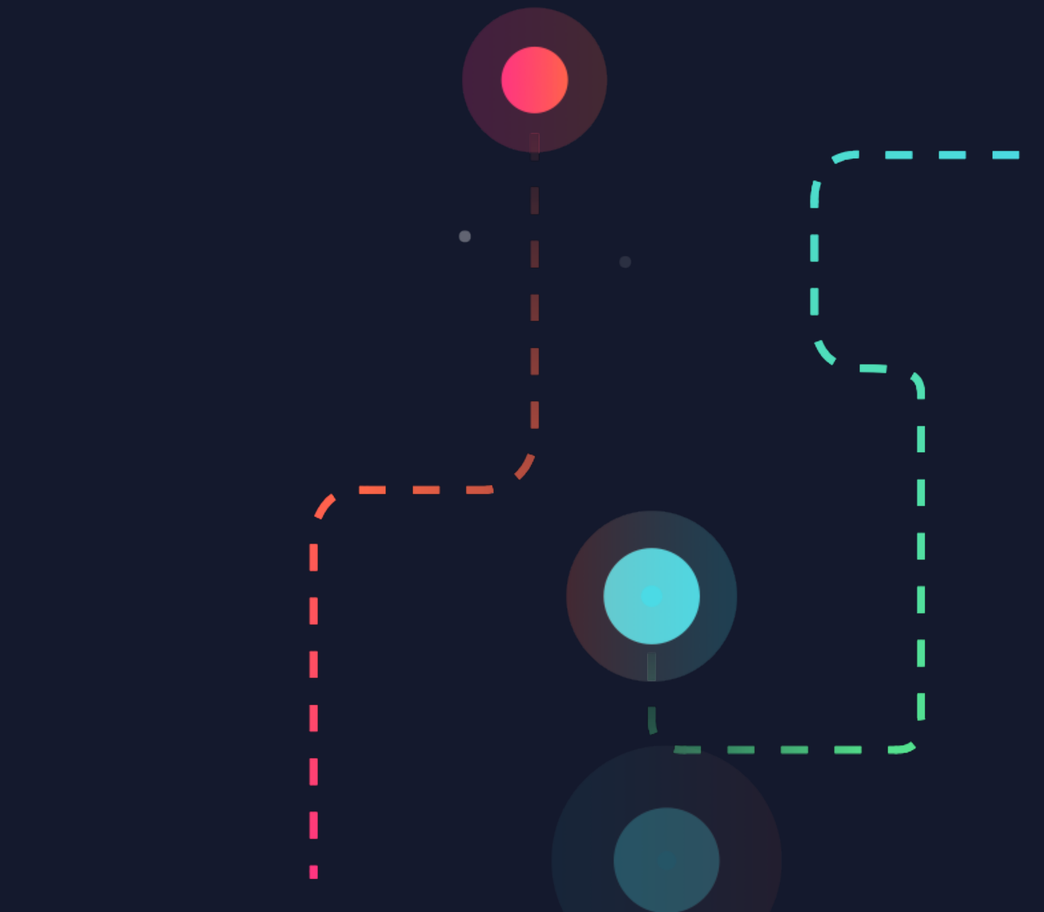
Manages various controllers.

ReplicaSet / Deployment controller

Node controller

Endpoints controller

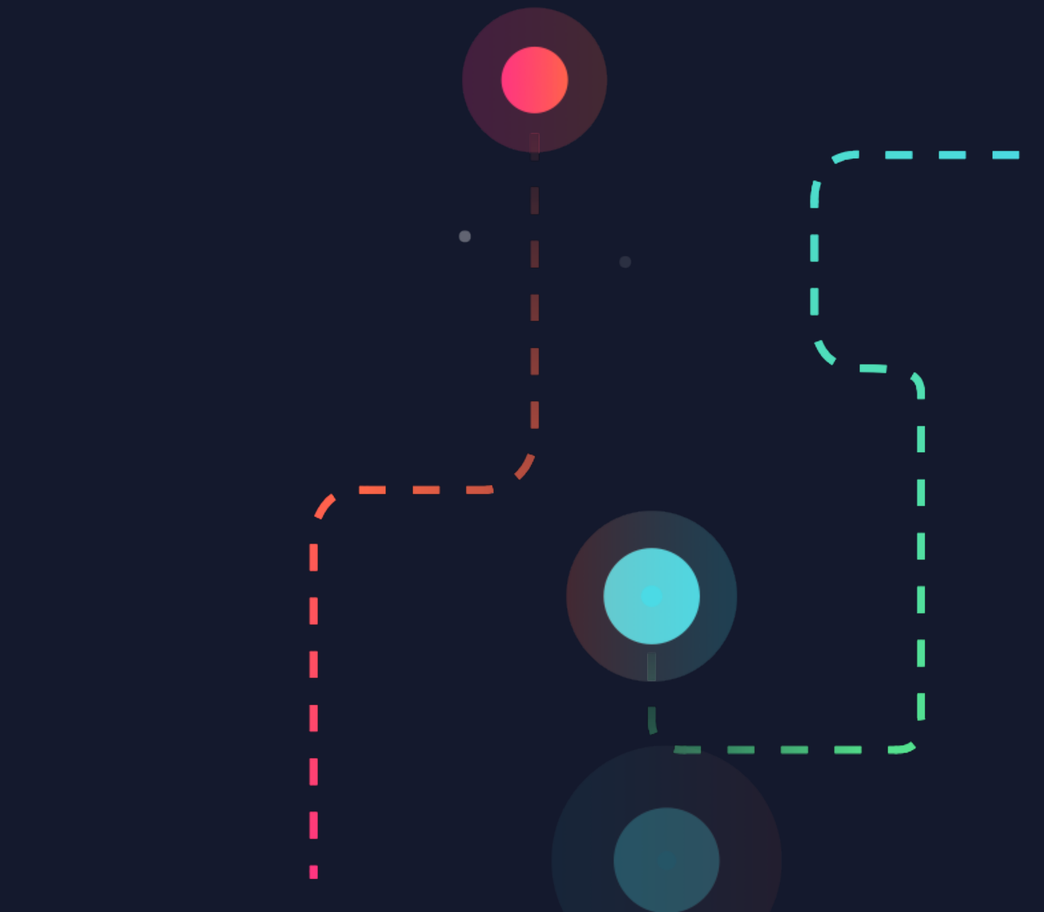
CRD controller



DUELBOX

+ CLOUD CONTROLLER MANAGER

Cloud-specific control logic:

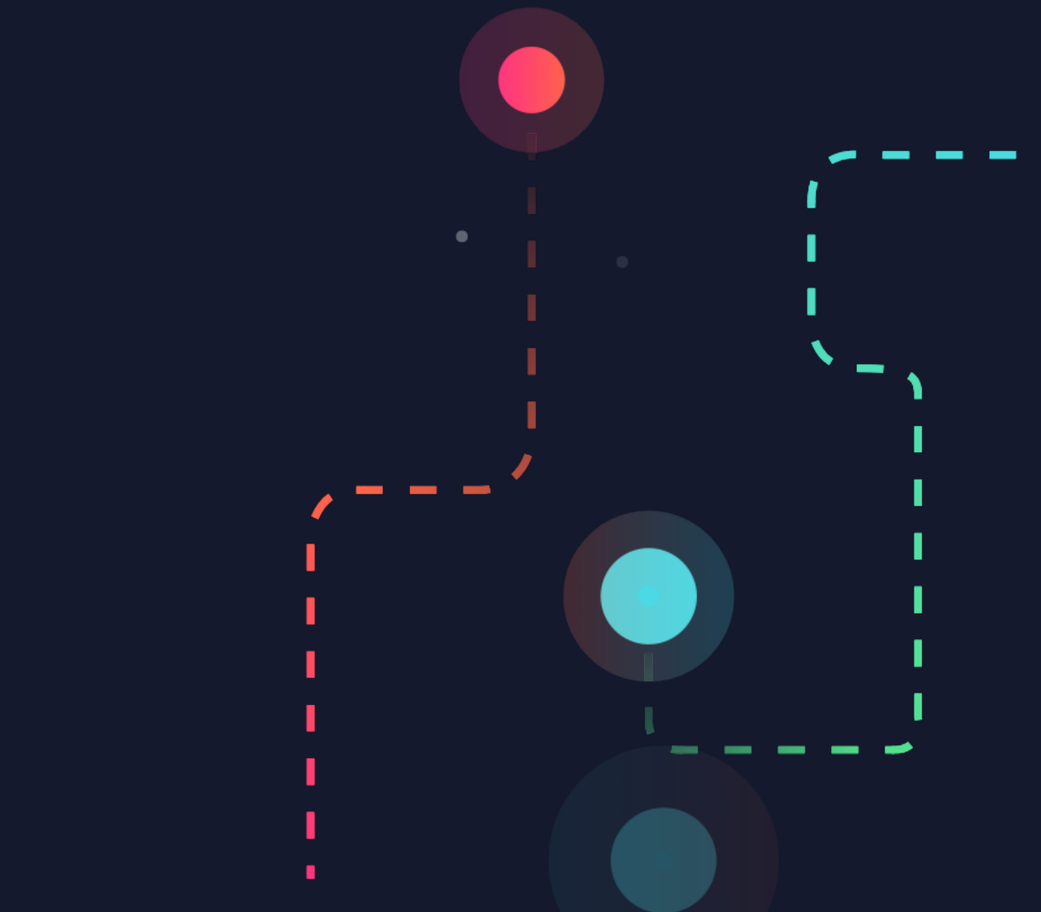


DUELBOX

+ CLOUD CONTROLLER MANAGER

Cloud-specific control logic:

Node-related information



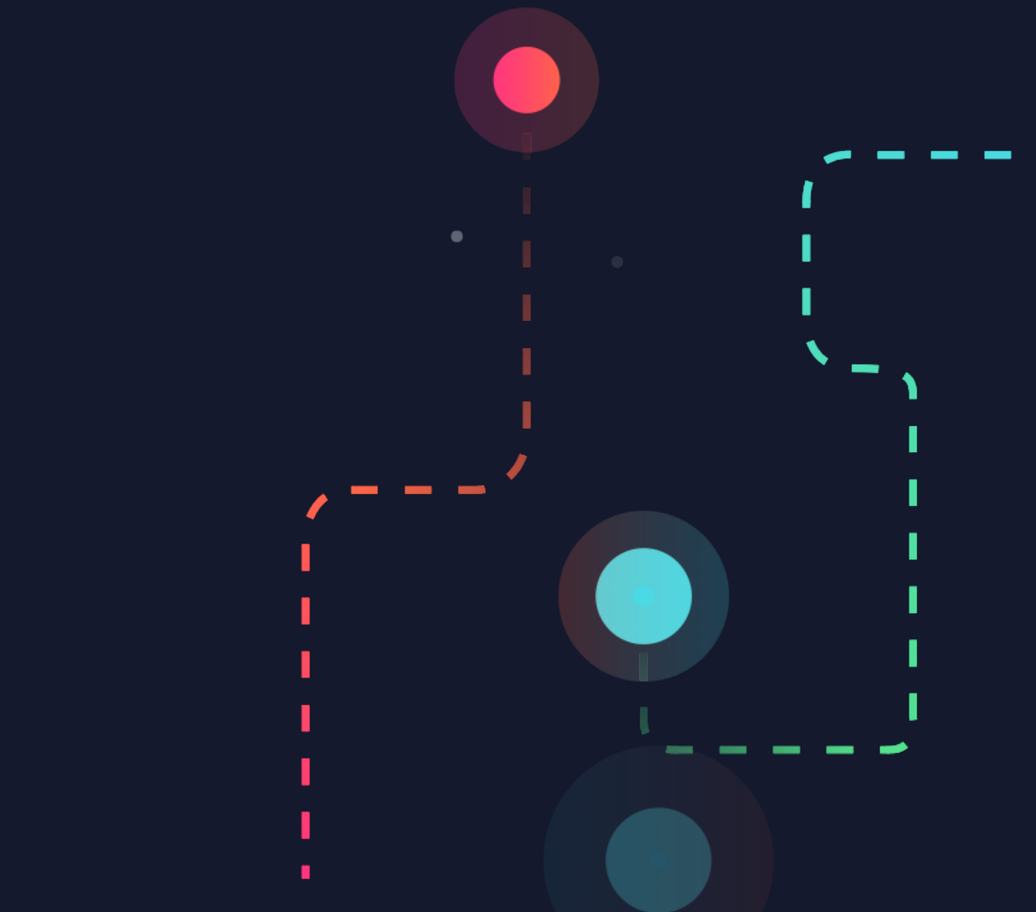
DUELBOX

+ CLOUD CONTROLLER MANAGER

Cloud-specific control logic:

Node-related information

Storage



DUELBOX

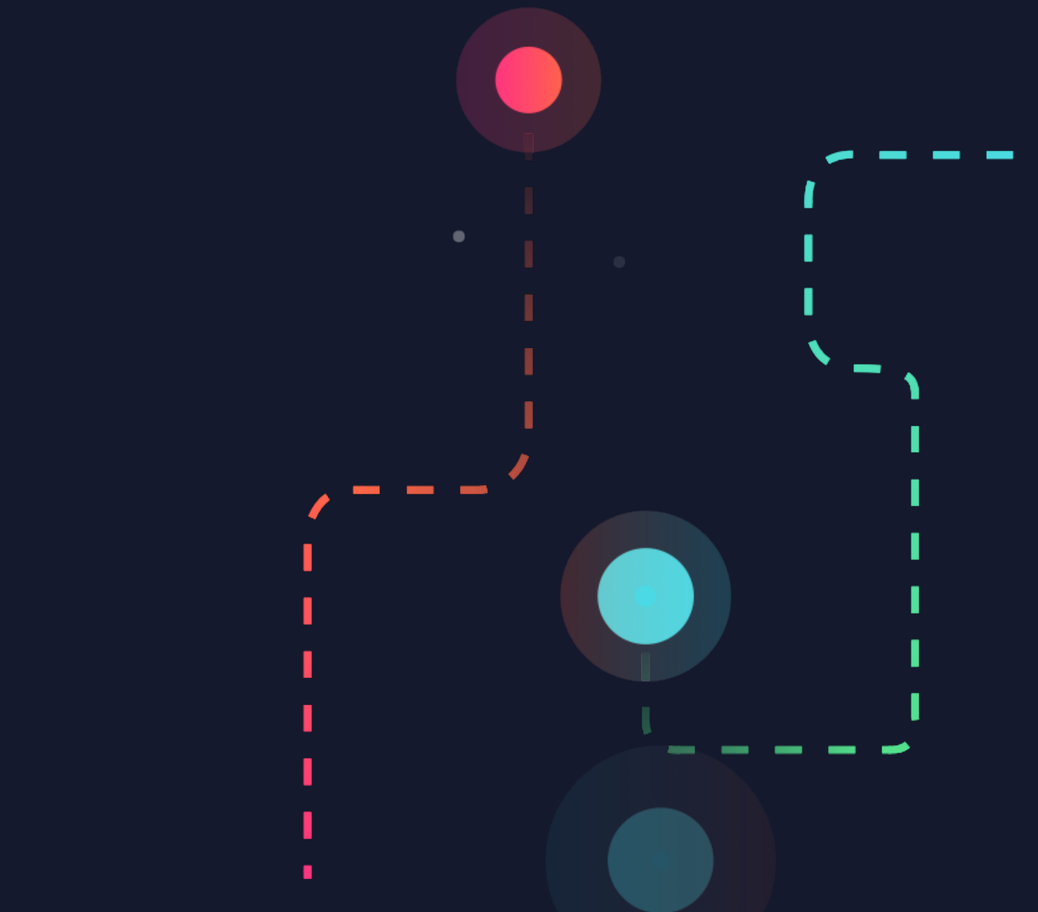
+ CLOUD CONTROLLER MANAGER

Cloud-specific control logic:

Node-related information

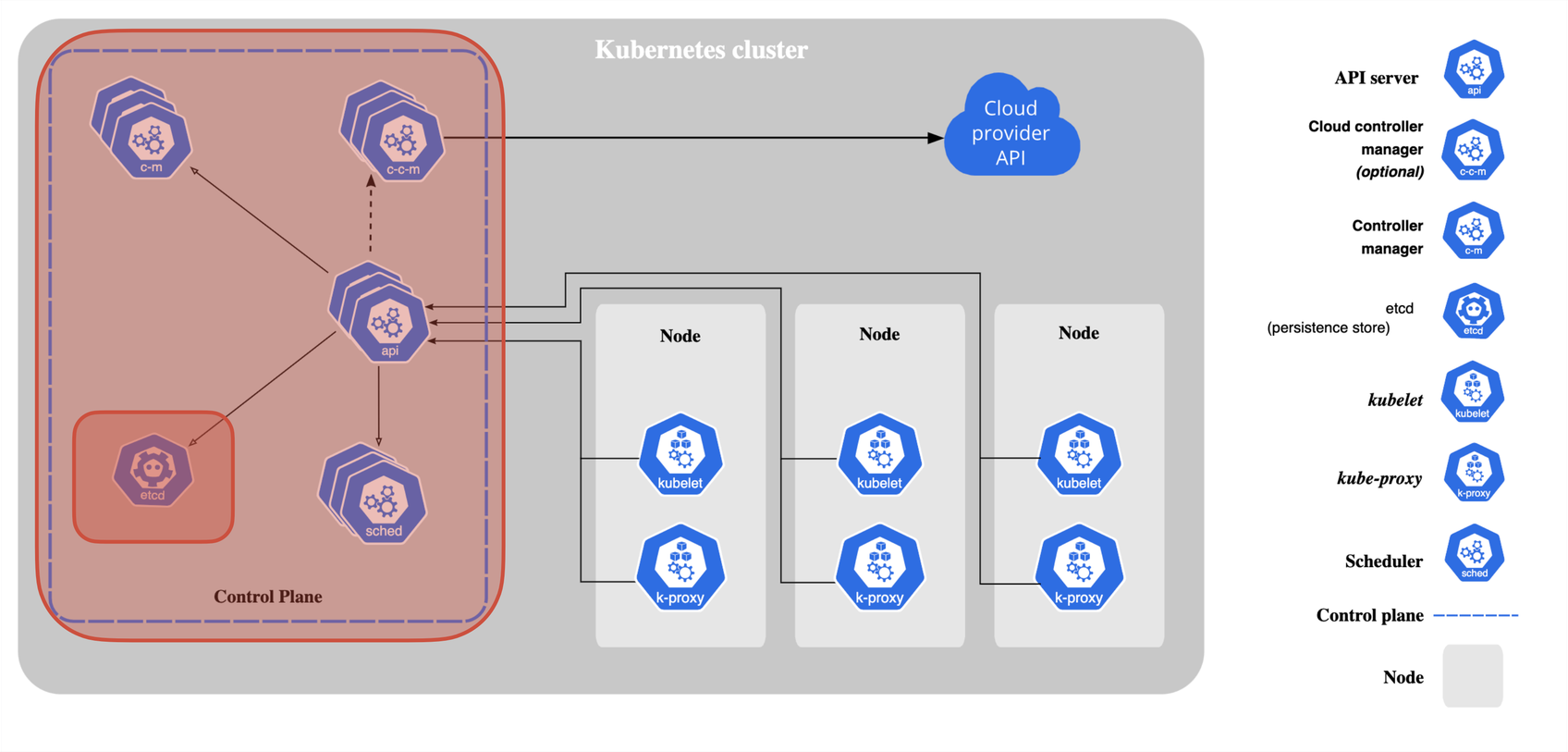
Storage

Services, Networking



DUELBOX

ETCD



DUELBOX

ETCD

The single source of truth – the database for storing all cluster data.

ETCD

The single source of truth – the database for storing all cluster data.

HA, distributed, key-value storage

ETCD

The single source of truth – the database for storing all cluster data.

HA, distributed, key-value storage

Kubernetes API objects: Pods, Services, Nodes, etc.

ETCD

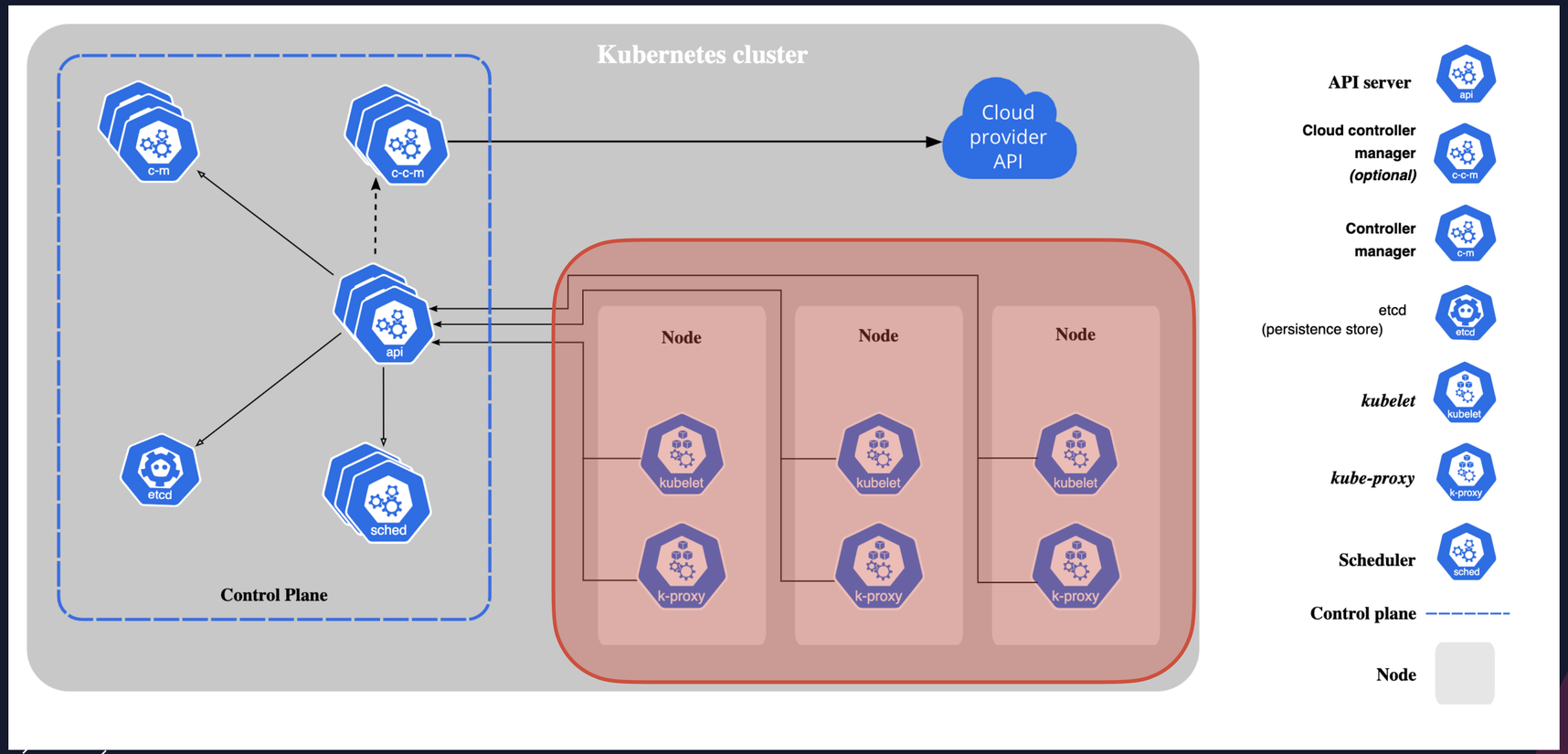
The single source of truth – the database for storing all cluster data.

HA, distributed, key-value storage

Kubernetes API objects: Pods, Services, Nodes, etc.

(Might be replaced with other storage engine: e.g. SQLite)

WORKER NODES



DUELBOX

KUBELET

The Kubernetes agent that runs on each node of the cluster.

KUBELET

The Kubernetes agent that runs on each node of the cluster.

Registers node with the API server.

KUBELET

The Kubernetes agent that runs on each node of the cluster.

Registers node with the API server.

Takes PodSpecs from the API server and ensures that containers of Pods are running and healthy.

KUBELET

The Kubernetes agent that runs on each node of the cluster.

Registers node with the API server.

Takes PodSpecs from the API server and ensures that containers of Pods are running and healthy.

Reports the status of the node itself.

+ CONTAINER RUNTIME

Docker

containerd

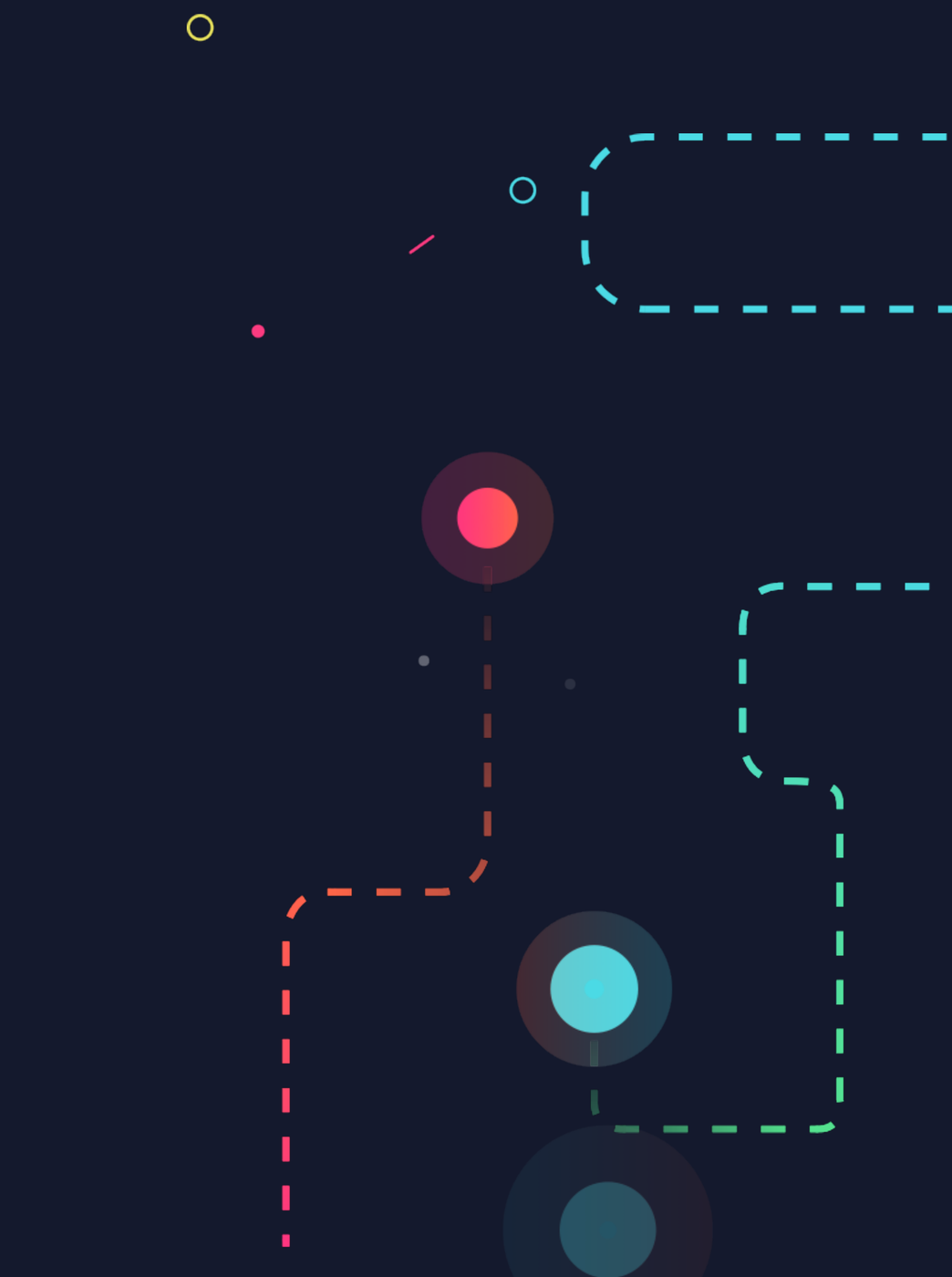
CRI-O

DUELBOX

PROXY

kube-proxy

A network proxy running on each node.



DUELBOX

PROXY

kube-proxy

A network proxy running on each node.

Maintains network rules to allow network communication to Pods from inside or outside of the cluster.

PROXY

kube-proxy

A network proxy running on each node.

Maintains network rules to allow network communication to Pods from inside or outside of the cluster.

Key component for Services.

+ OTHER

DNS

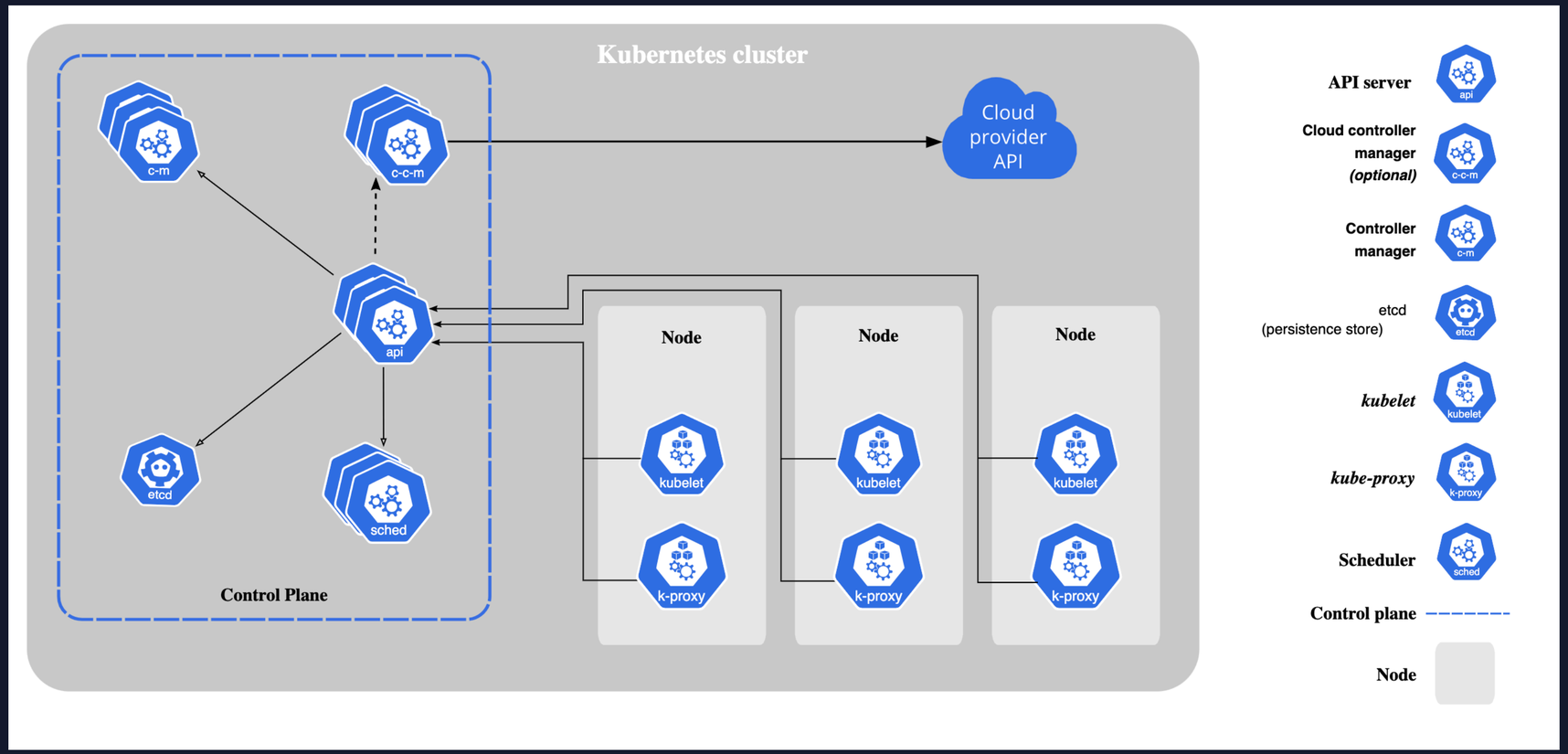
Web UI (Dashboard)

Cluster-level logging

RECAP

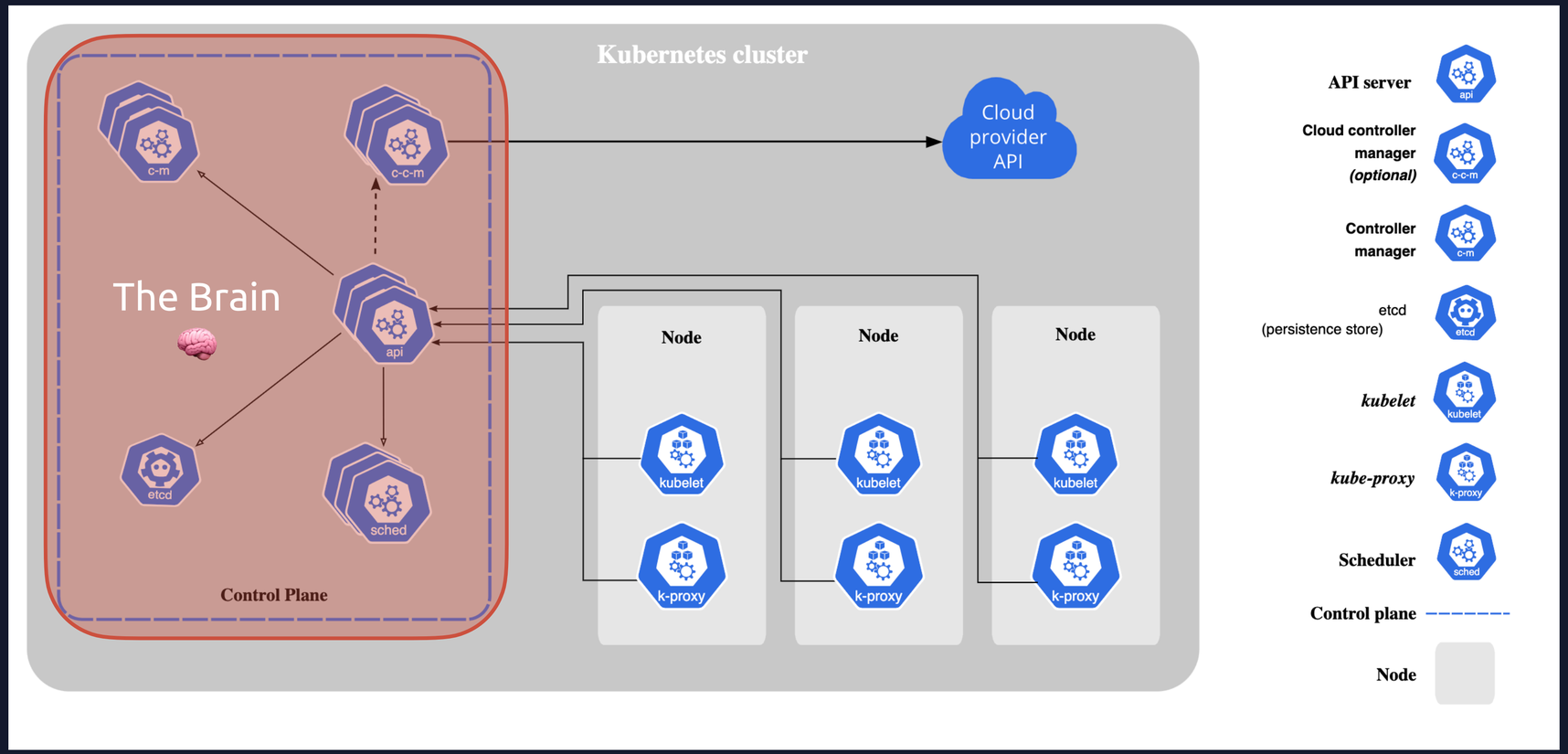
DUELBOX

KUBERNETES ARCHITECTURE



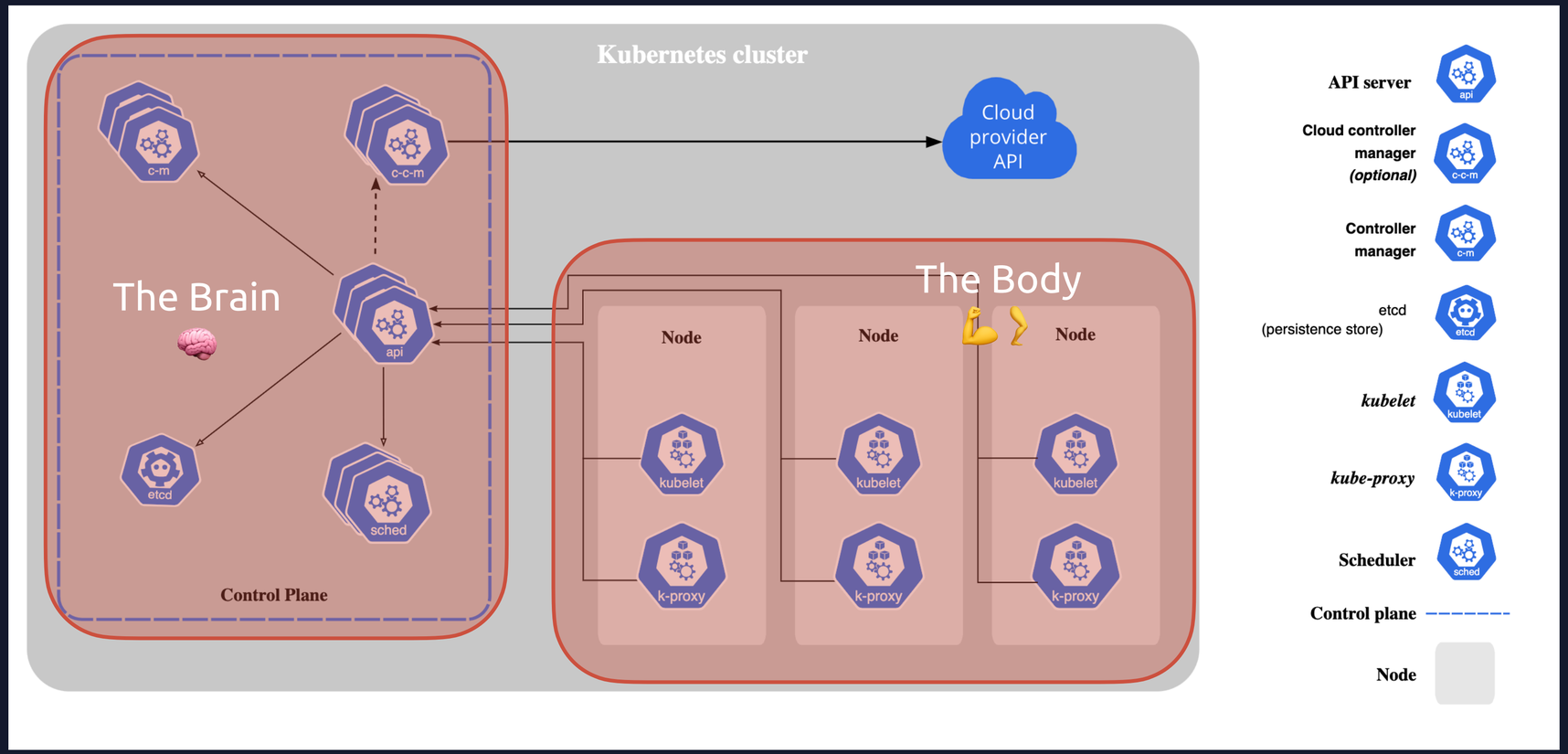
DUELBOX

KUBERNETES ARCHITECTURE



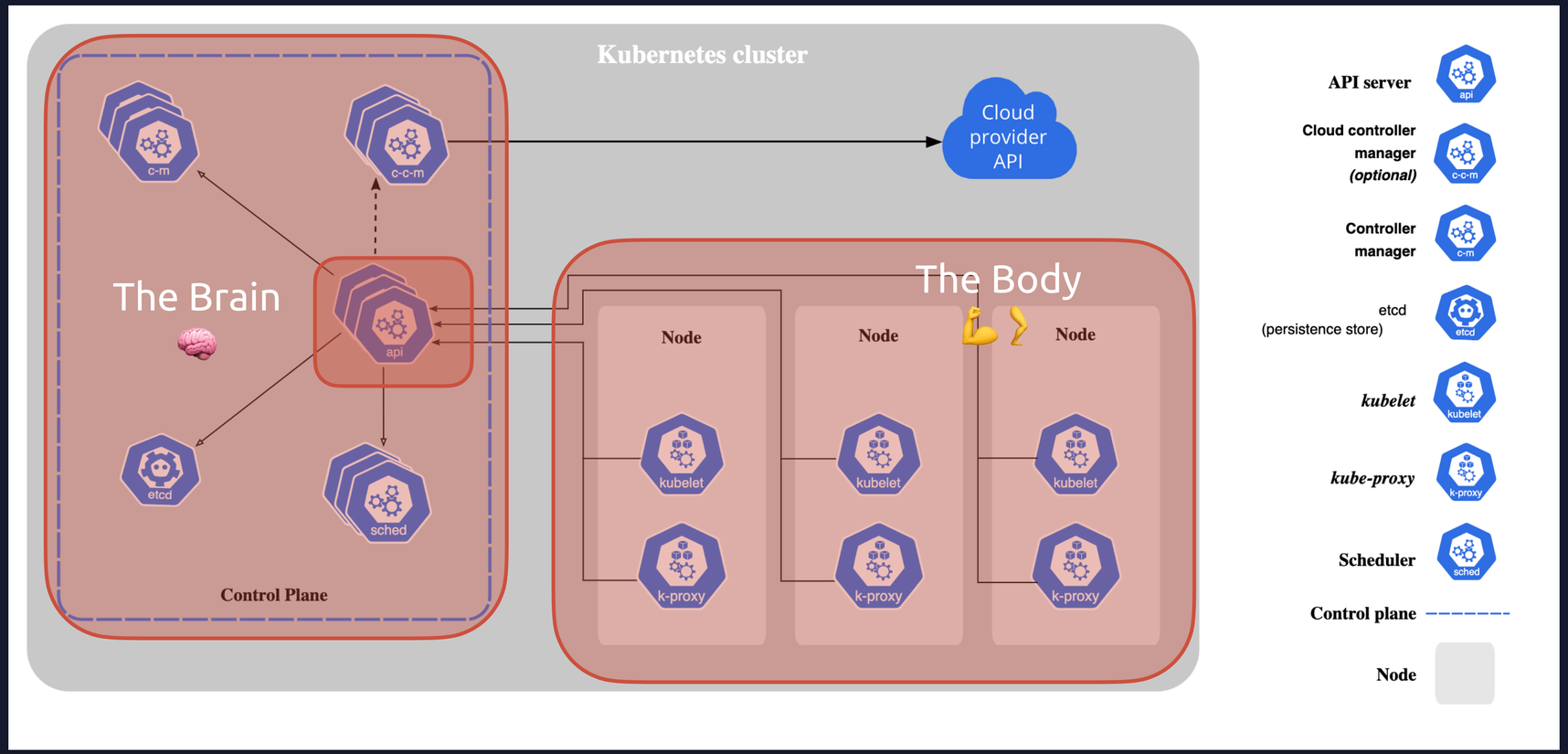
DUELBOX

KUBERNETES ARCHITECTURE



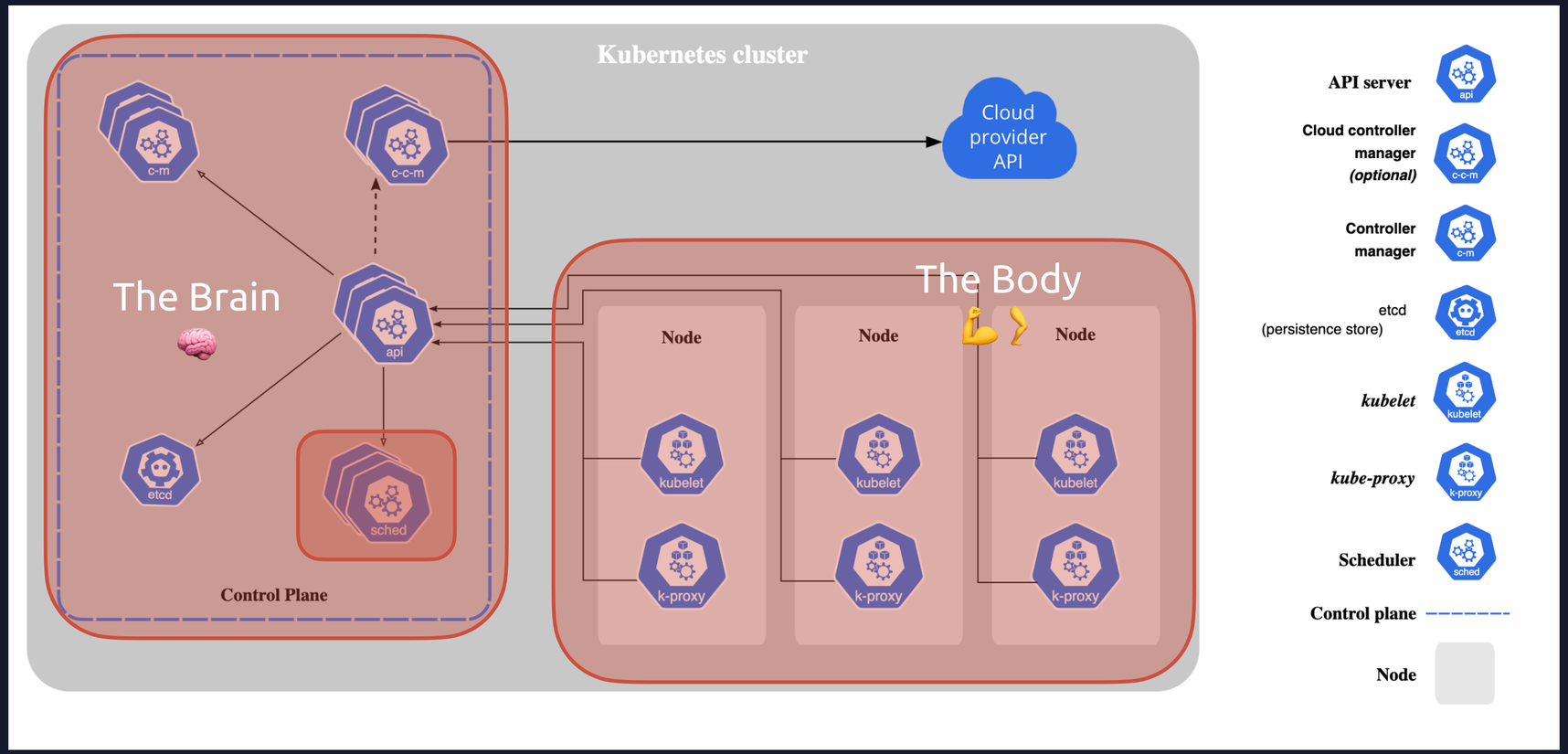
DUELBOX

KUBERNETES ARCHITECTURE



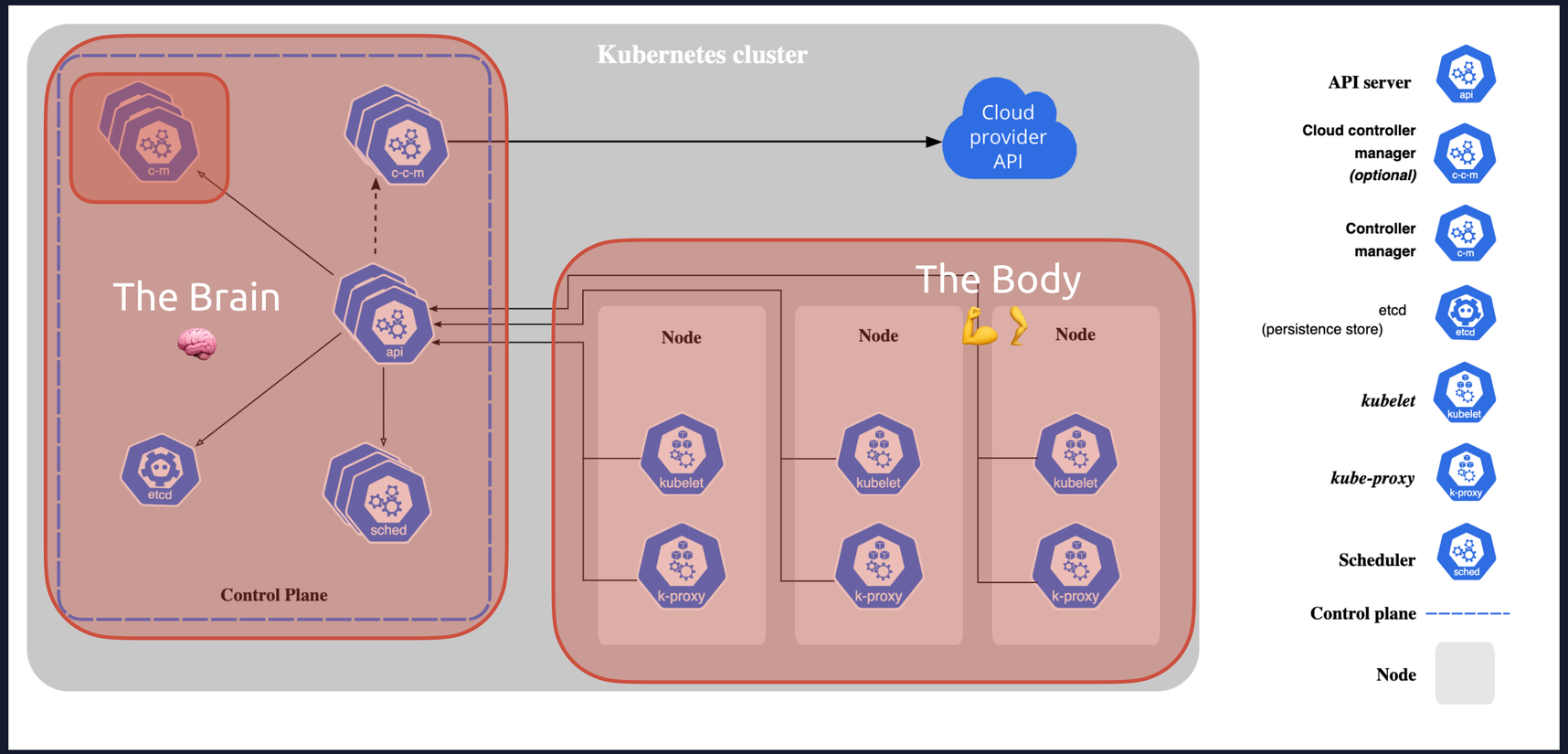
DUELBOX

KUBERNETES ARCHITECTURE



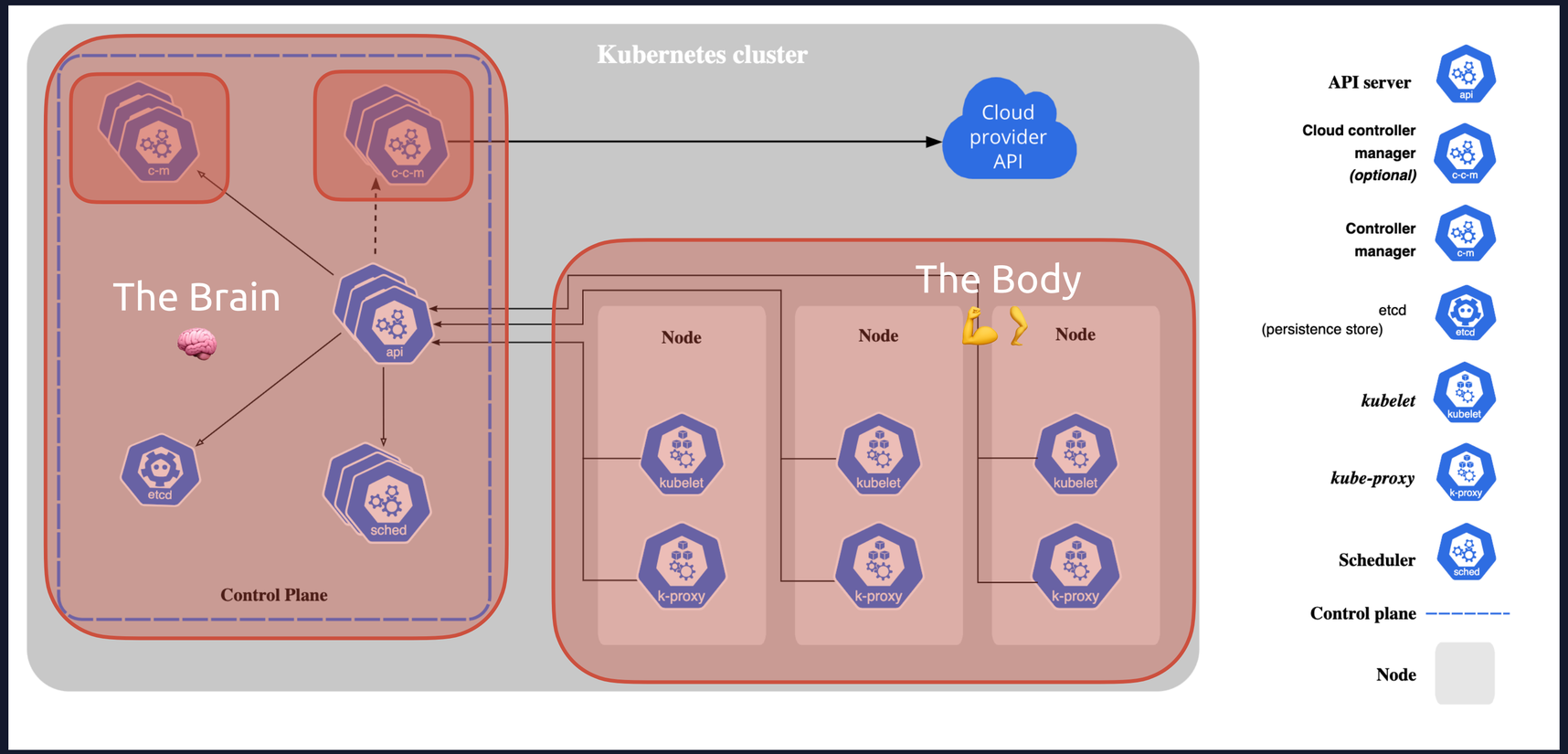
DUELBOX

KUBERNETES ARCHITECTURE



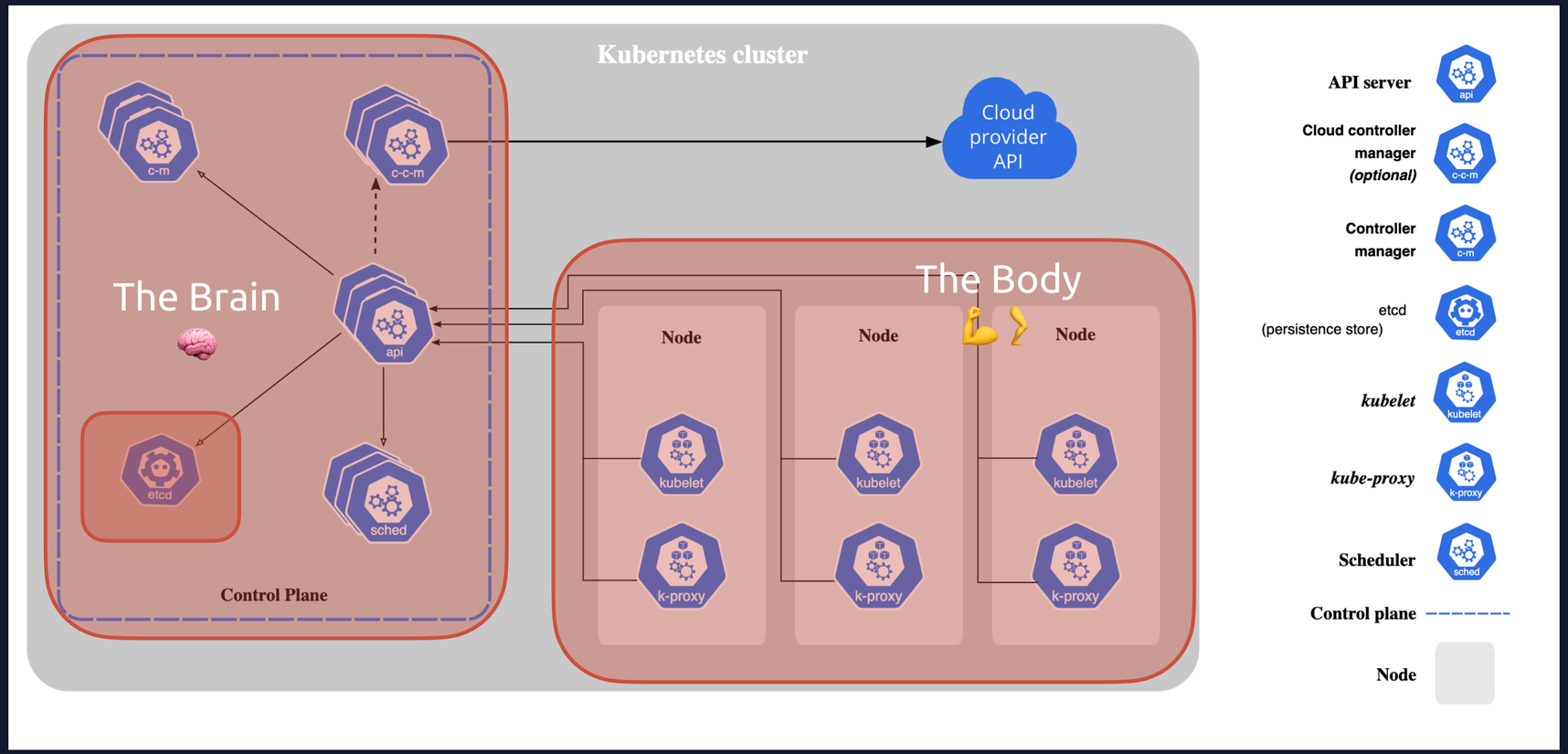
DUELBOX

KUBERNETES ARCHITECTURE



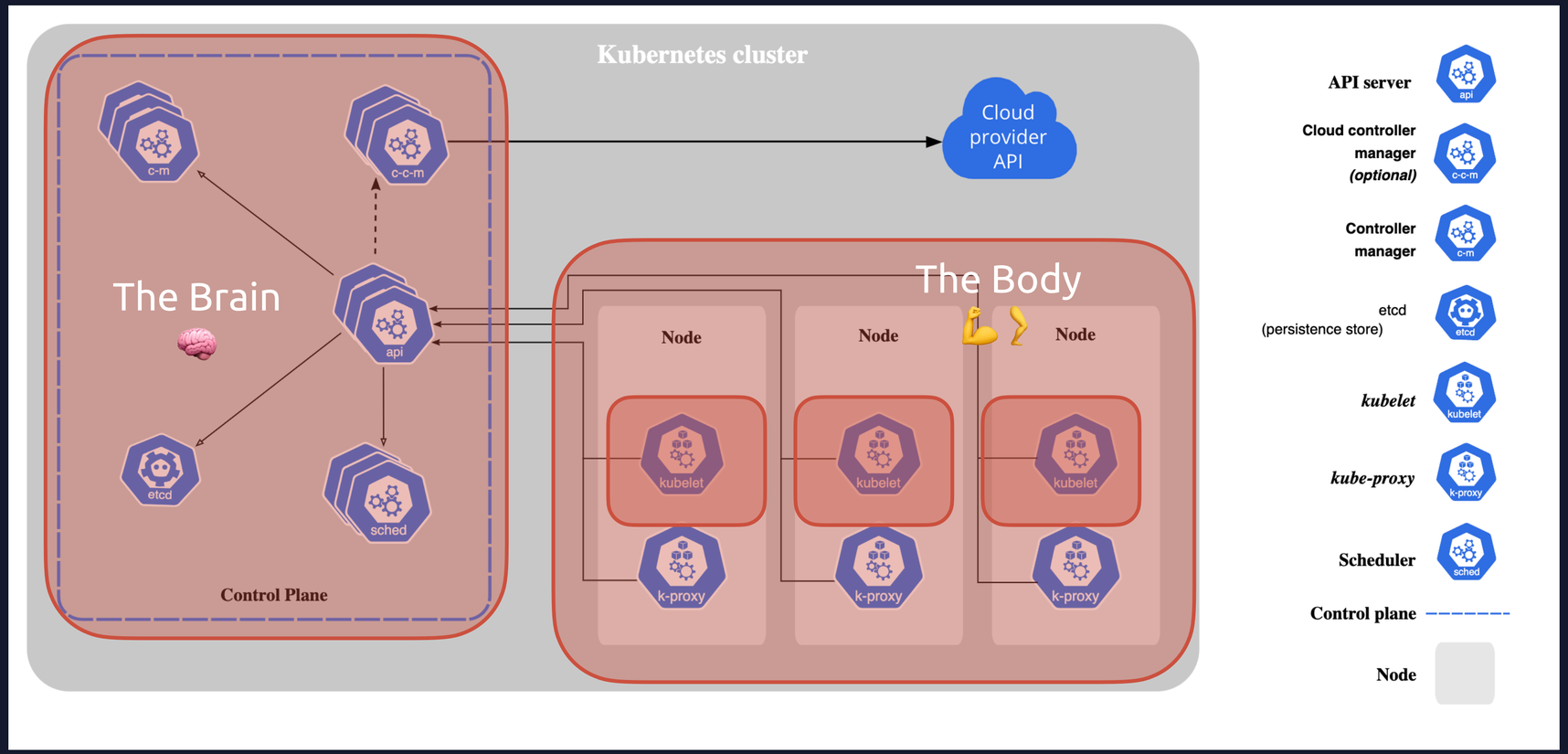
DUELBOX

KUBERNETES ARCHITECTURE



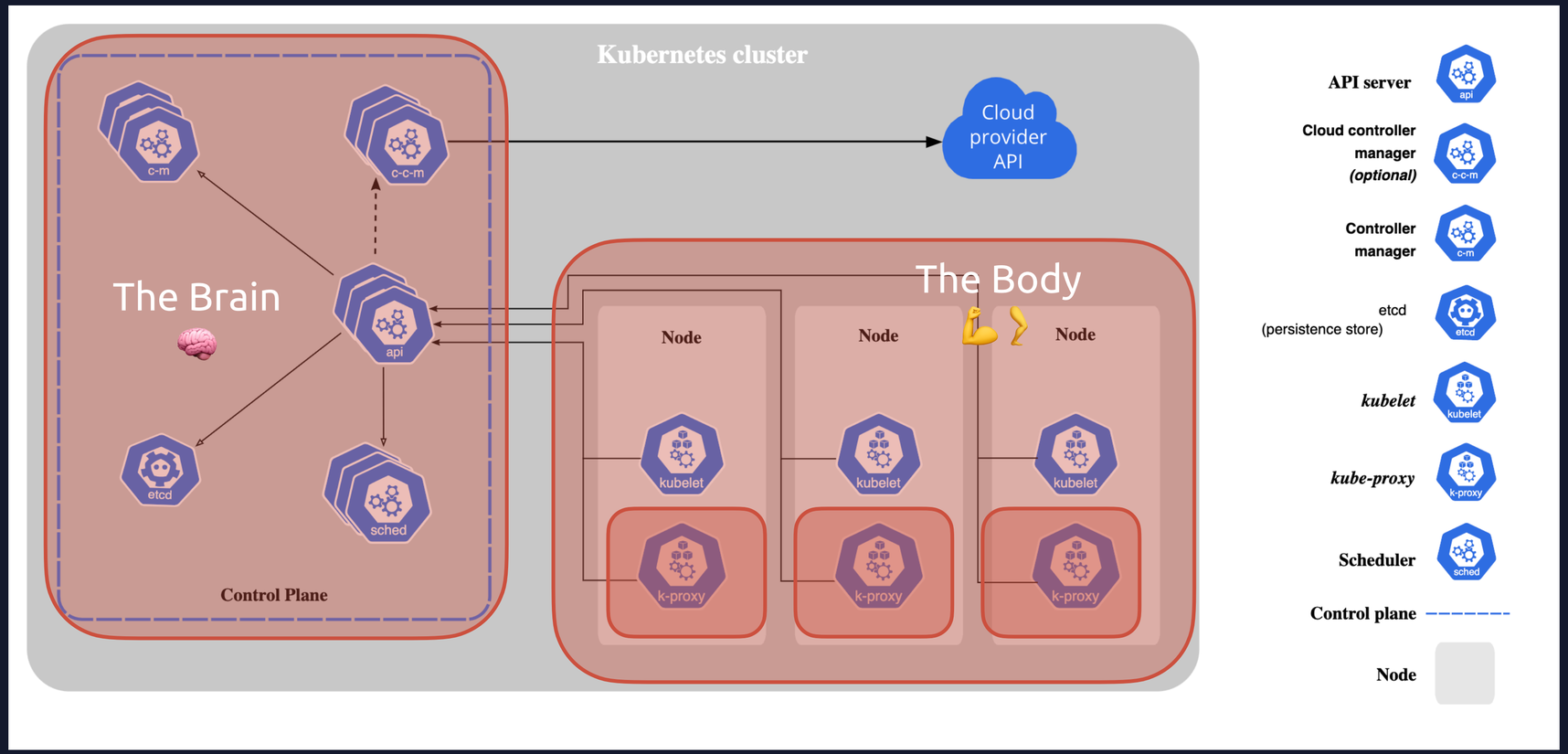
DUELBOX

KUBERNETES ARCHITECTURE



DUELBOX

KUBERNETES ARCHITECTURE



DUELBOX

THANK YOU!

DUELBOX