Foundations of Kubernetes

The small ideas behind a big project

FOK

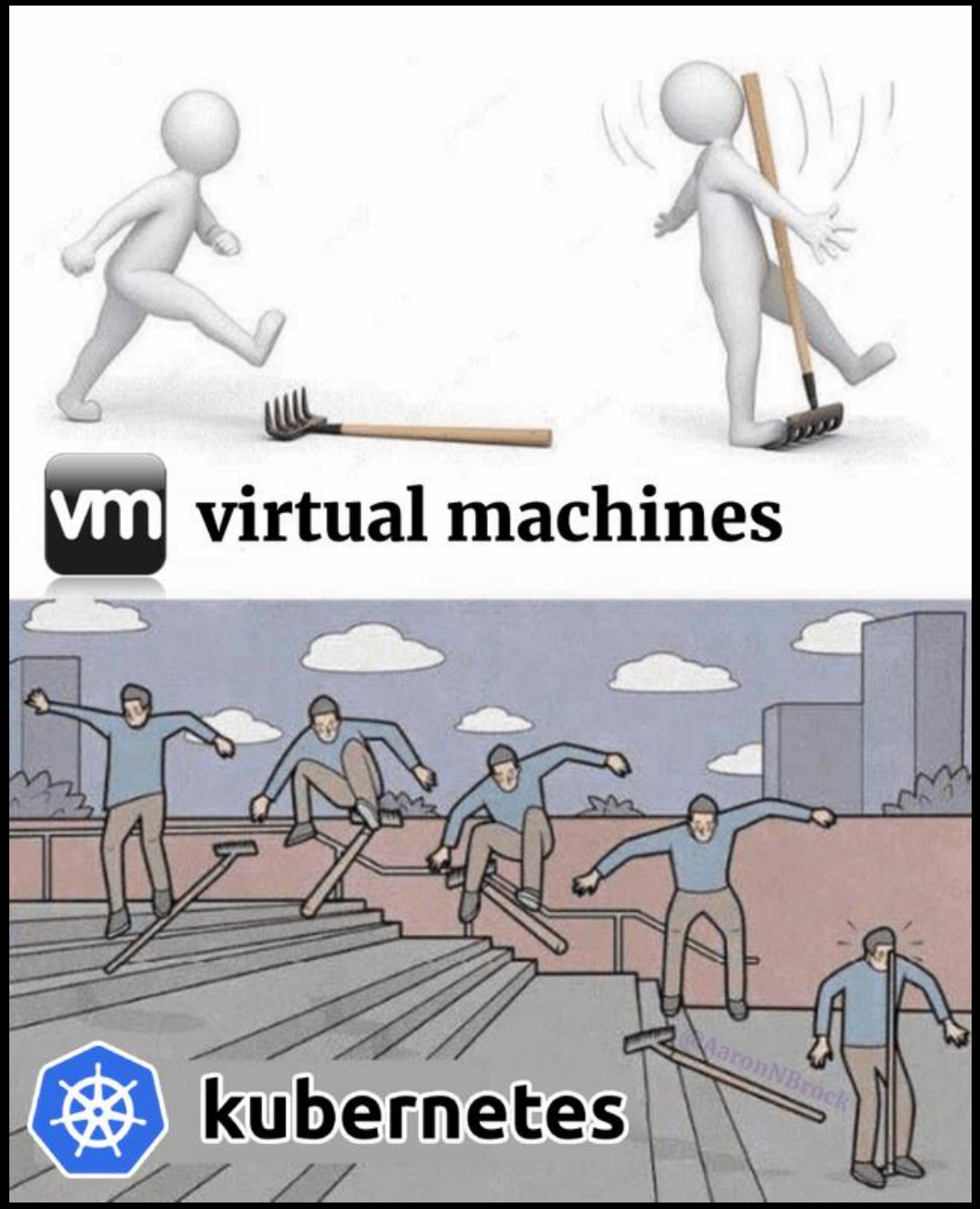
The small ideas behind a big project

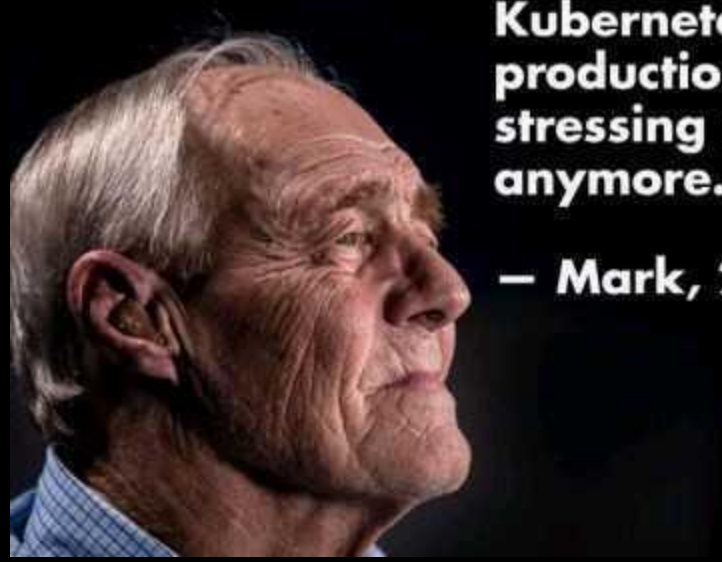
Kubernetes, also known as K8s, is an open-source system for automating deployment, scaling, and management of containerized applications.

It groups containers that make up an application into logical units for easy management and discovery. Kubernetes builds upon 15 years of experience of running production workloads at Google, combined with best-of-breed ideas and practices from the community.

https://kubernetes.io/

Why?


virtual machines
kubernetes



Running a bare metal Kubernetes cluster in production isn't stressing me out anymore.

— Mark, 22 years old

It is the second largest open source project in the world after Linux and is the primary container orchestration tool for 71% of Fortune 100 companies. Looking ahead, Kubernetes adoption shows no signs of slowing down – according to Gartner's "The CTO's Guide to Containers and Kubernetes" by 2027, more than 90% of global organizations will be running containerized applications in production.

Kubernetes Project Journey Report | CNCF

Goals

- Discuss the big ideas
- Be hands-on
- Think differently about Kubernetes

Non-Goals

- Explain containers
- Demonstrate application migrations
- Explore day 2 concerns
- Cover best practices

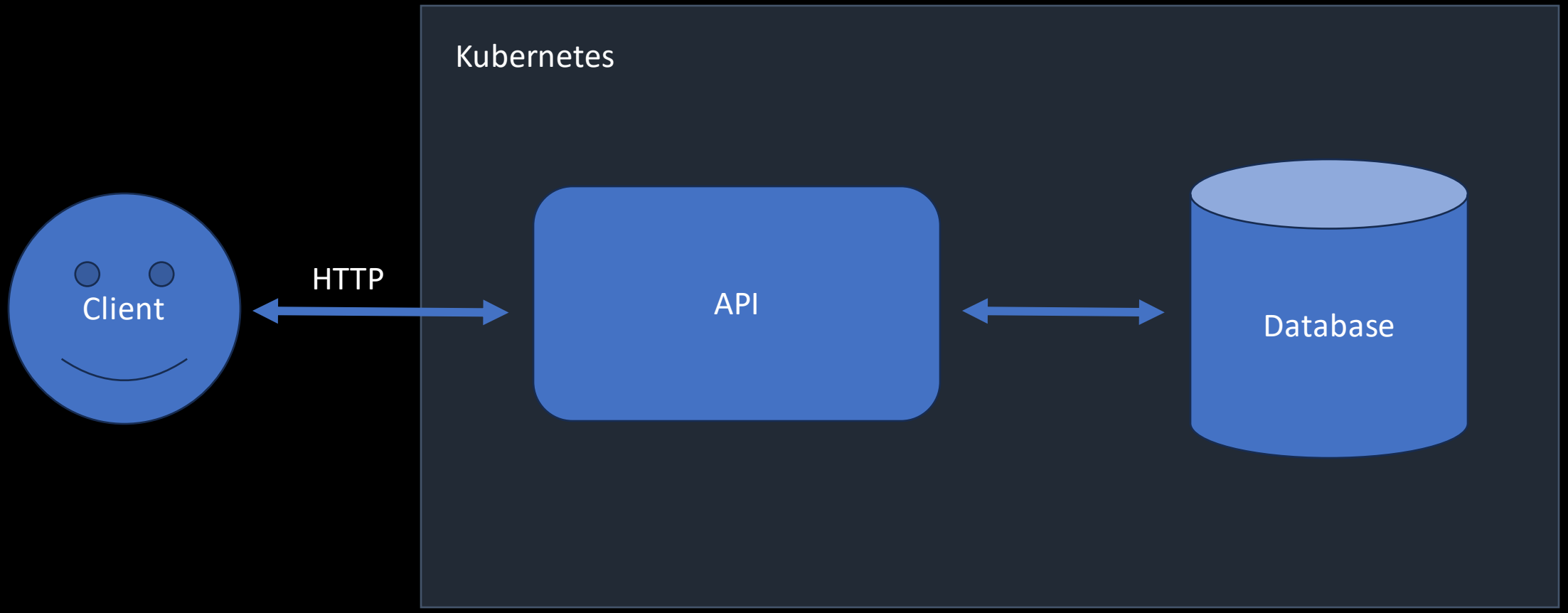
SwiftOnSecurity
@SwiftOnSecurity

One time I tried to explain Kubernetes to someone.
Then we both didn't understand it.

16:40 · 06/08/2019 · Twitter for iPhone

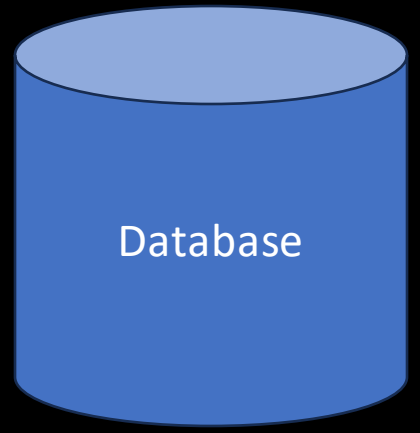
Kubernetes is an API

A CRUD JSON API


Kubernetes
Client
HTTP
API
Database

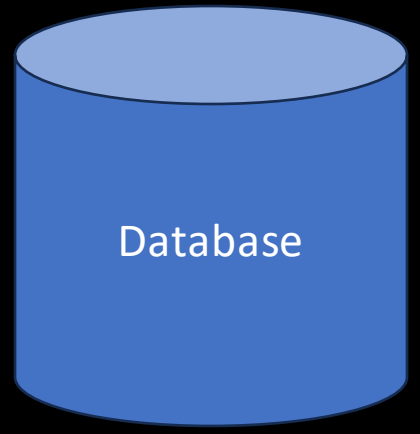
Resources

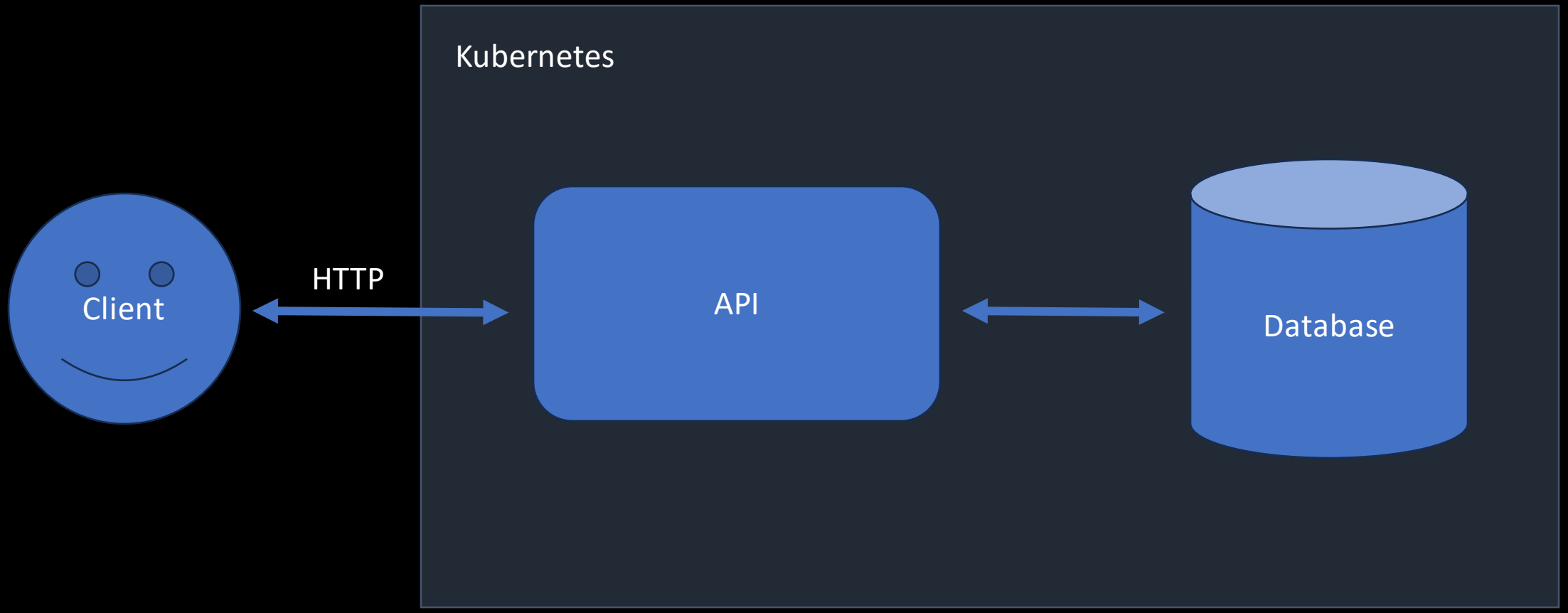
```
CREATE TABLE customer (
   ...
);

CREATE TABLE invoice (
   ...
);
```

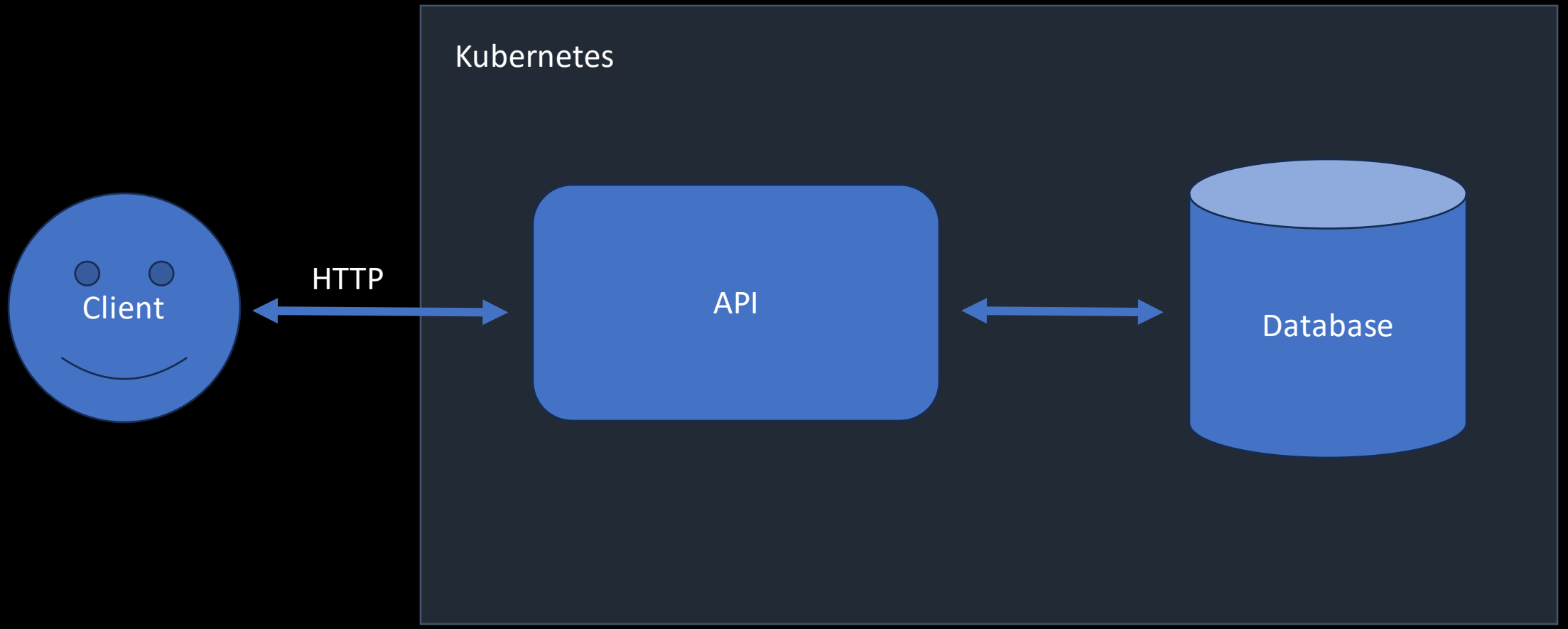


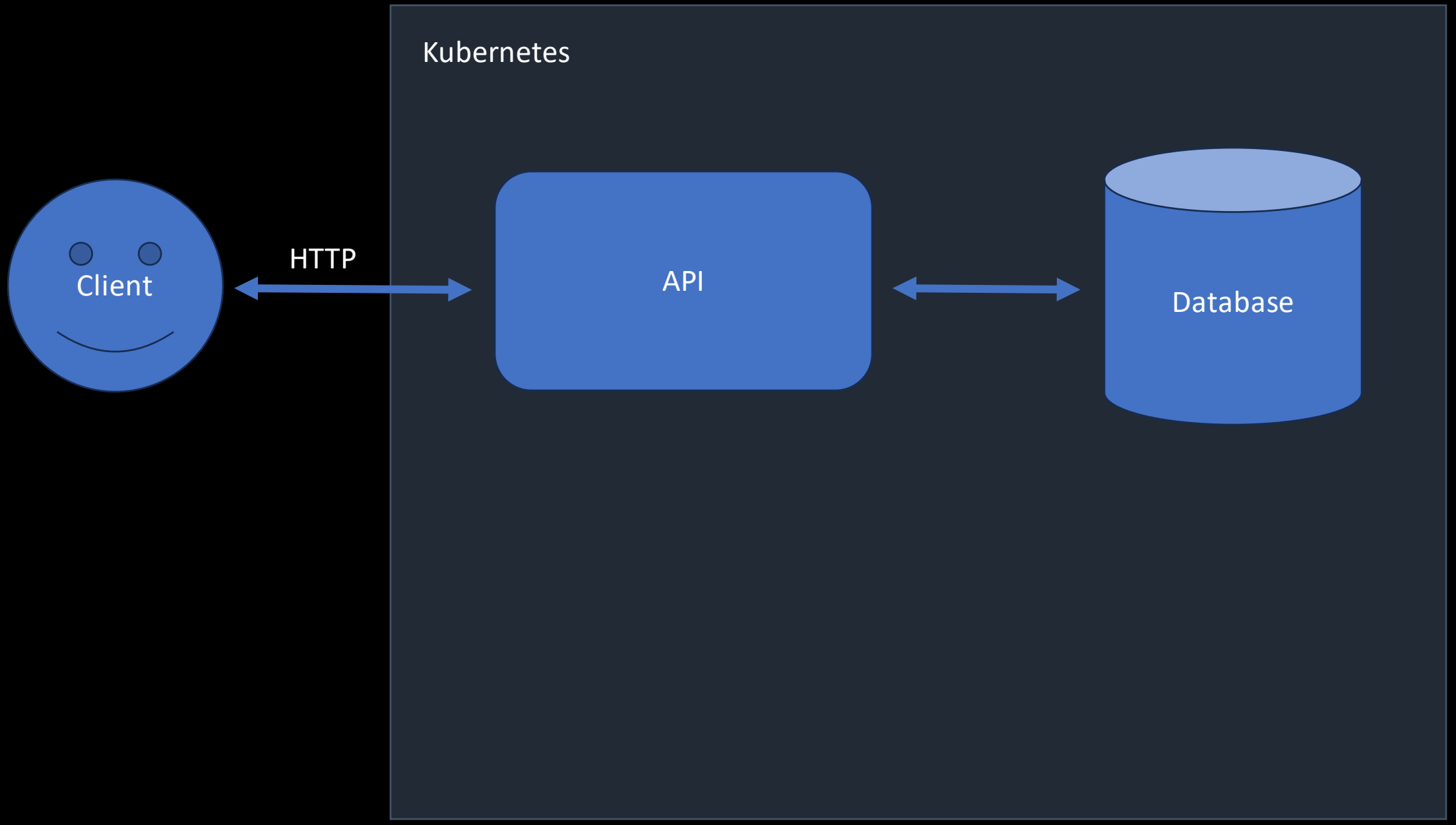
Resources

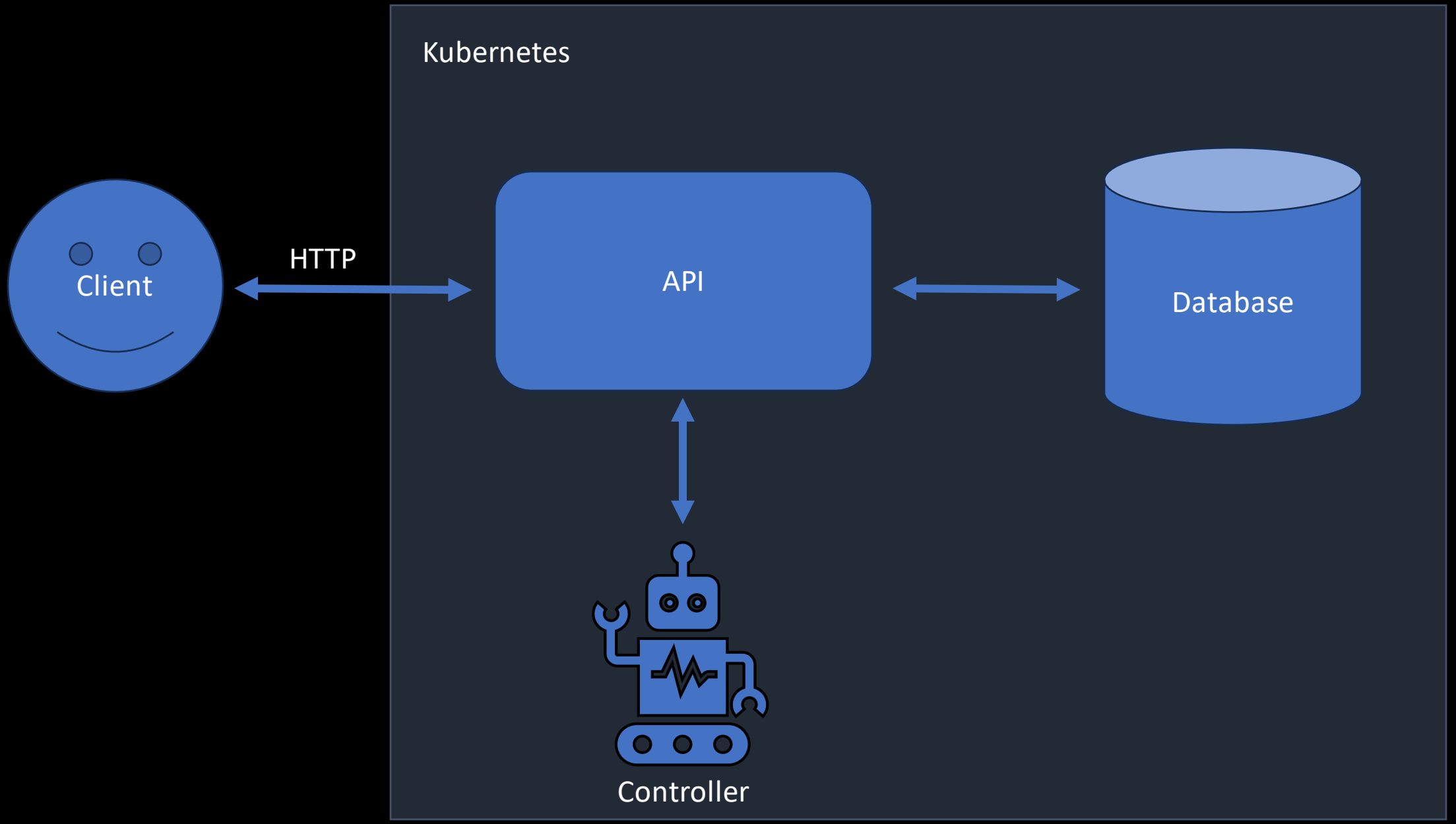
```
CREATE TABLE container (
   ...
);

CREATE TABLE firewall_rule (
   ...
);
```

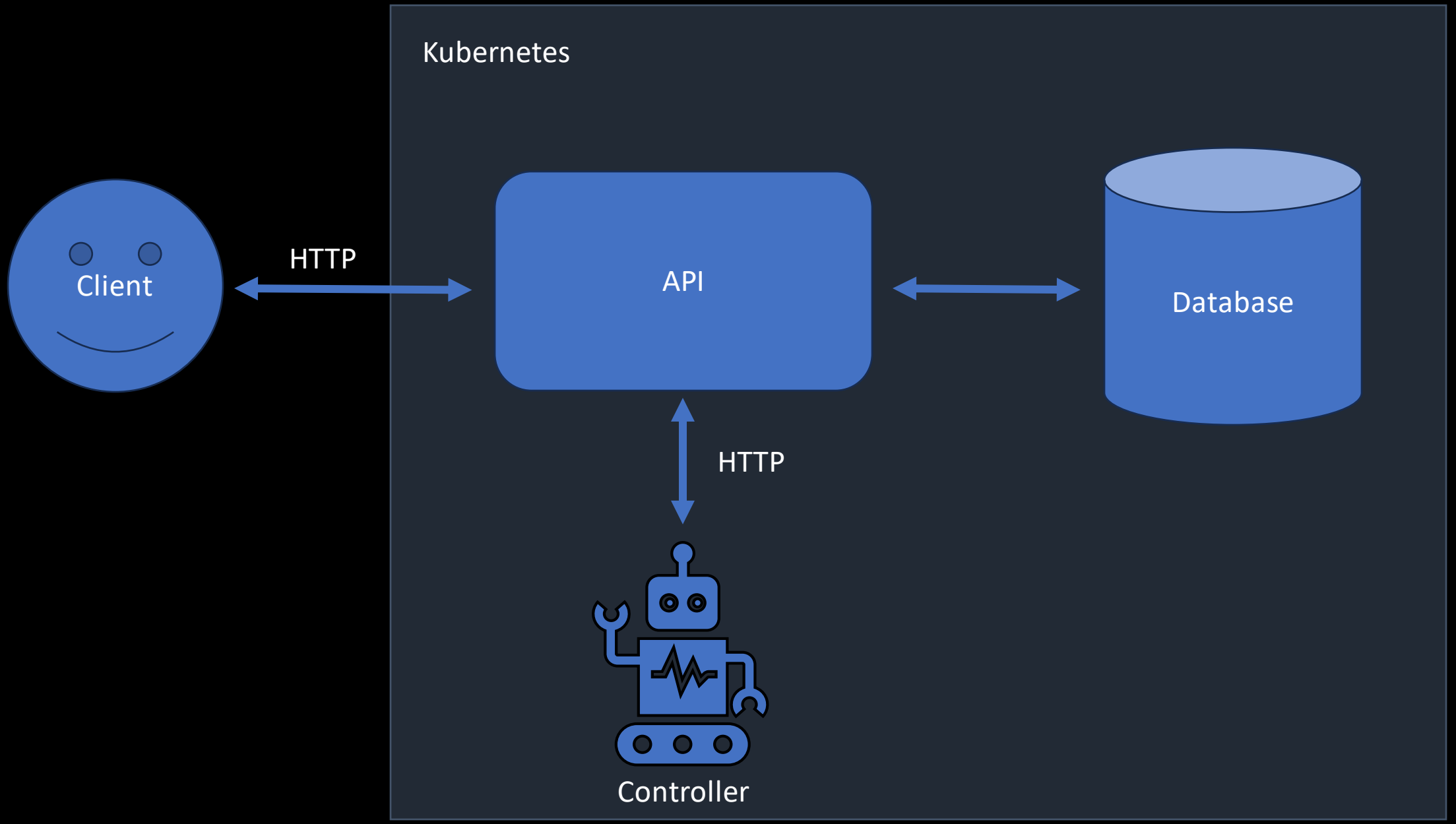



Kubernetes
Client
HTTP
API
Database

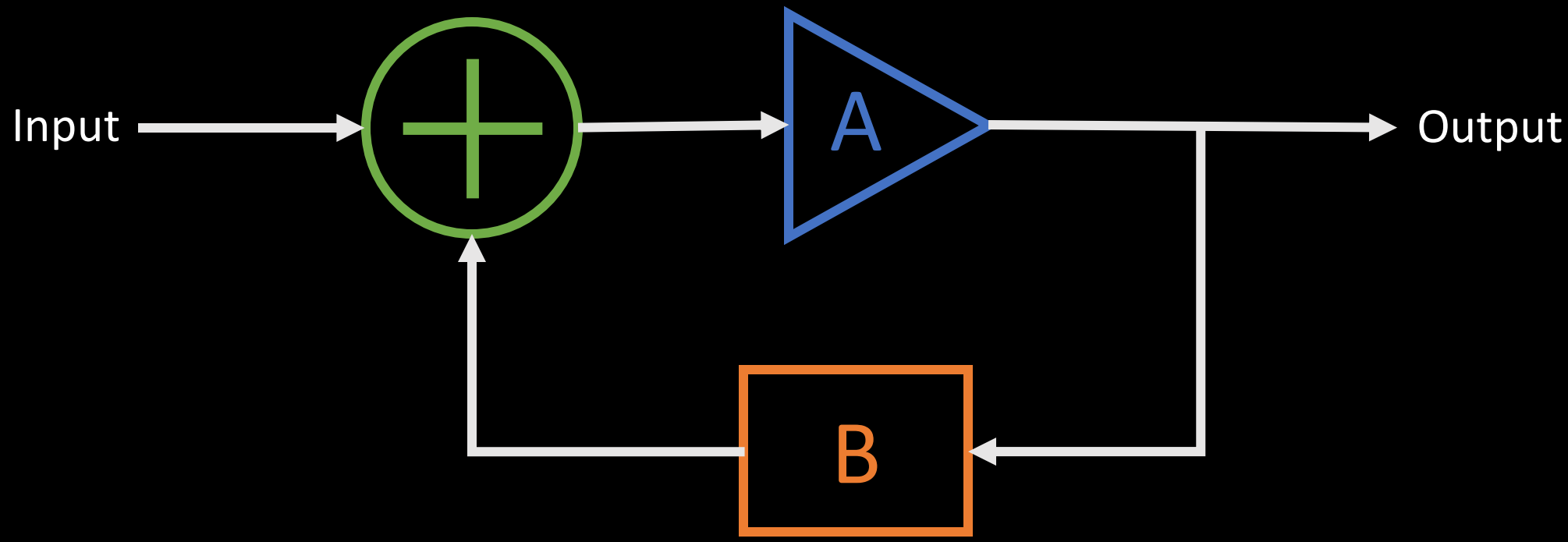

Kubernetes
Client
HTTP
API
Database


Kubernetes
Client
HTTP
API
Database

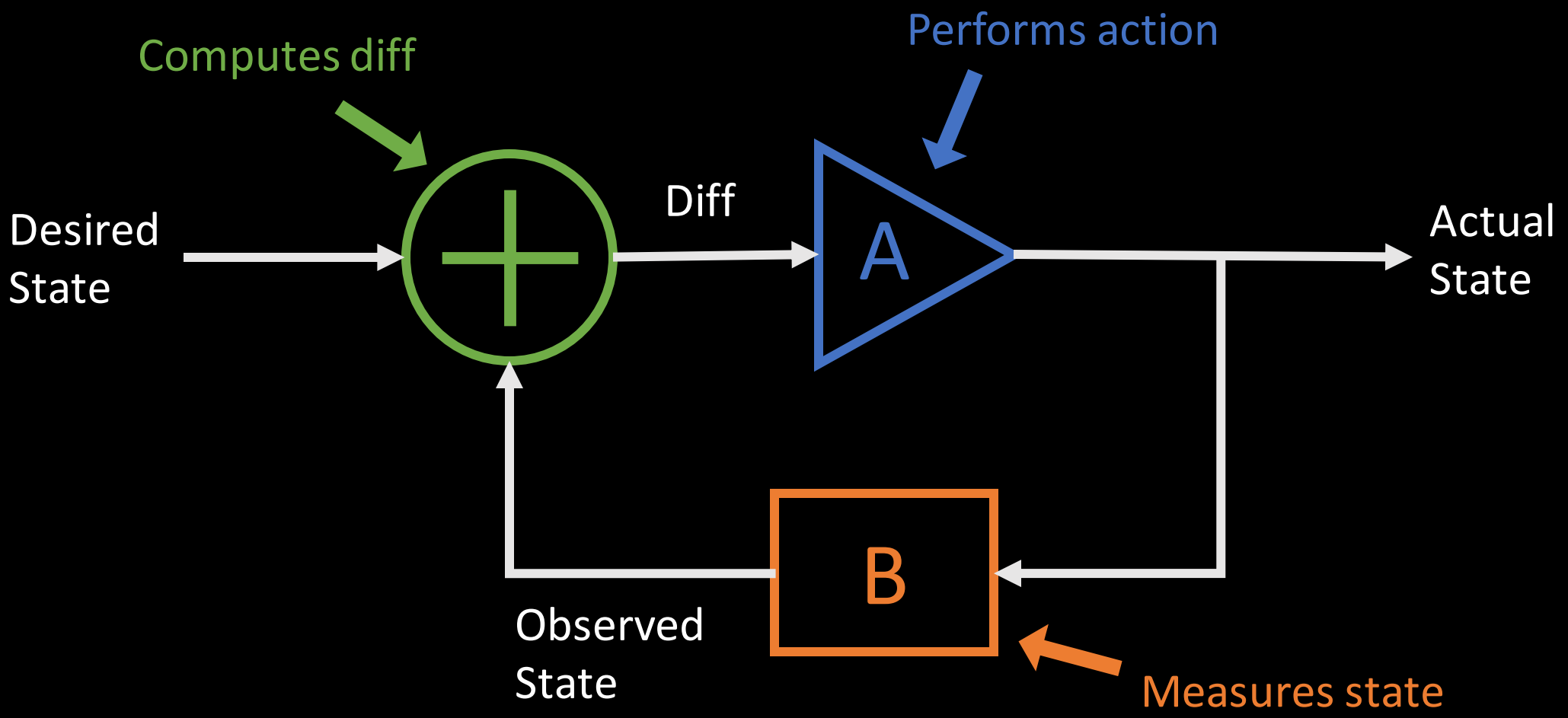

Kubernetes
Client
HTTP
API
Database
Controller


Kubernetes
Client
HTTP
API
Database
HTTP
Controller

Controller

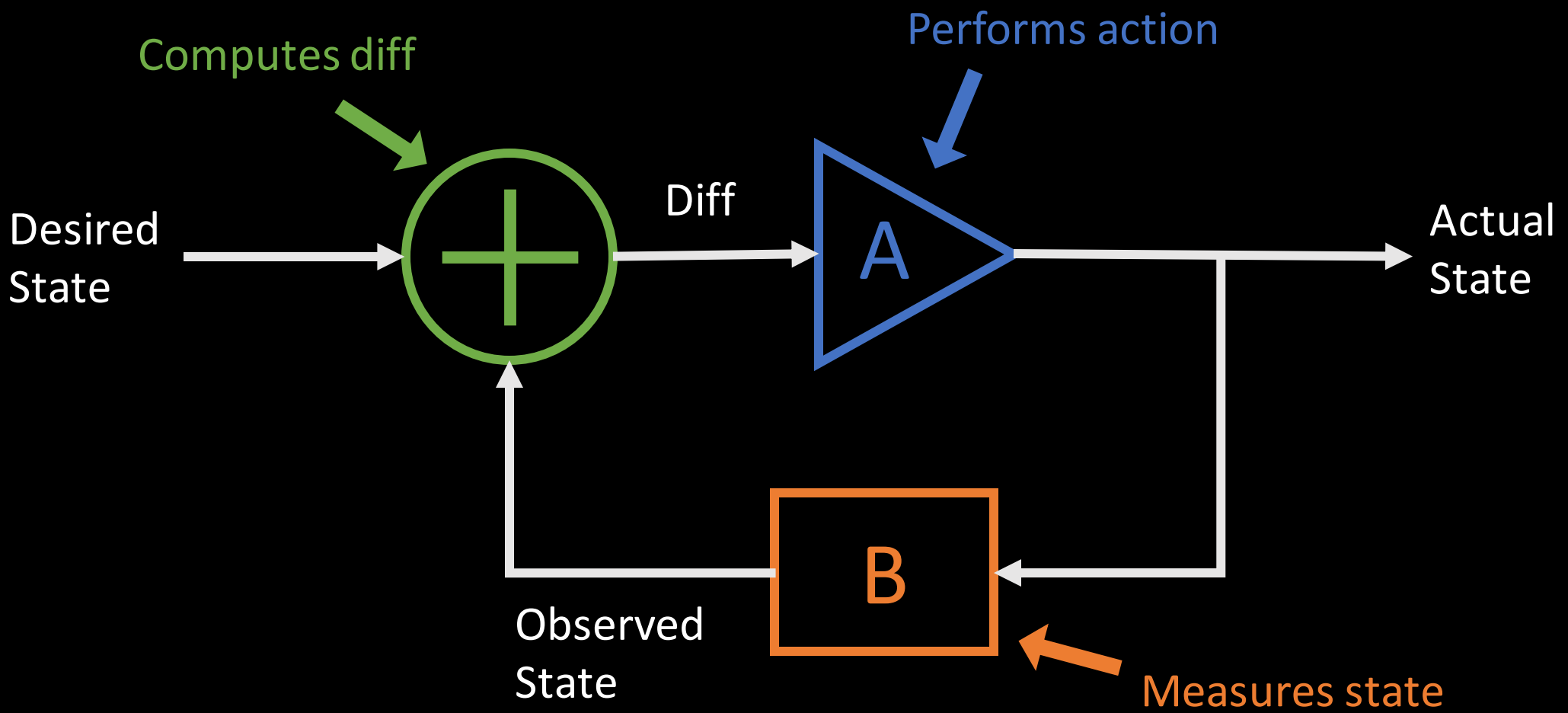


Controller


Computes diff
Performs action
Desired State
Diff
A
Actual State
B
Observed State
Measures state

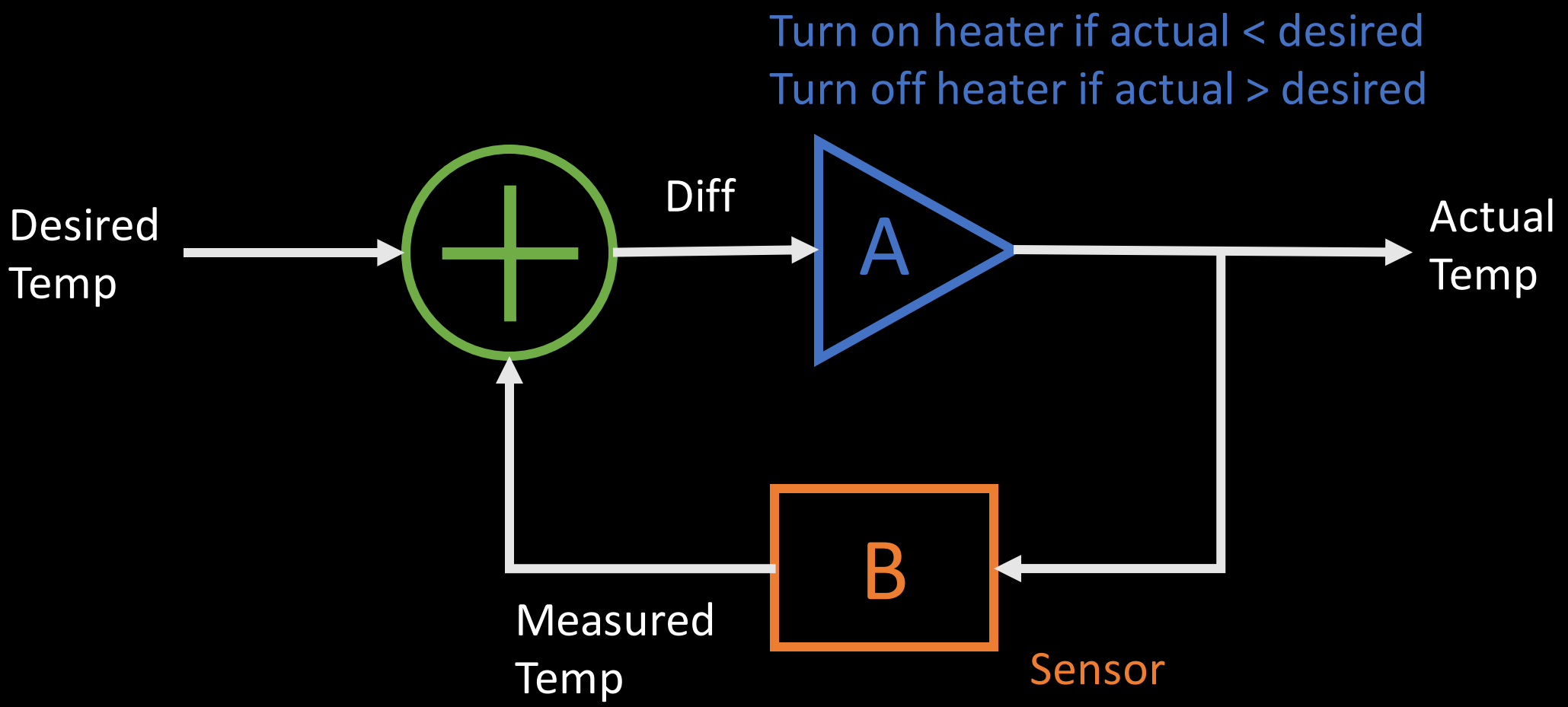
Controller

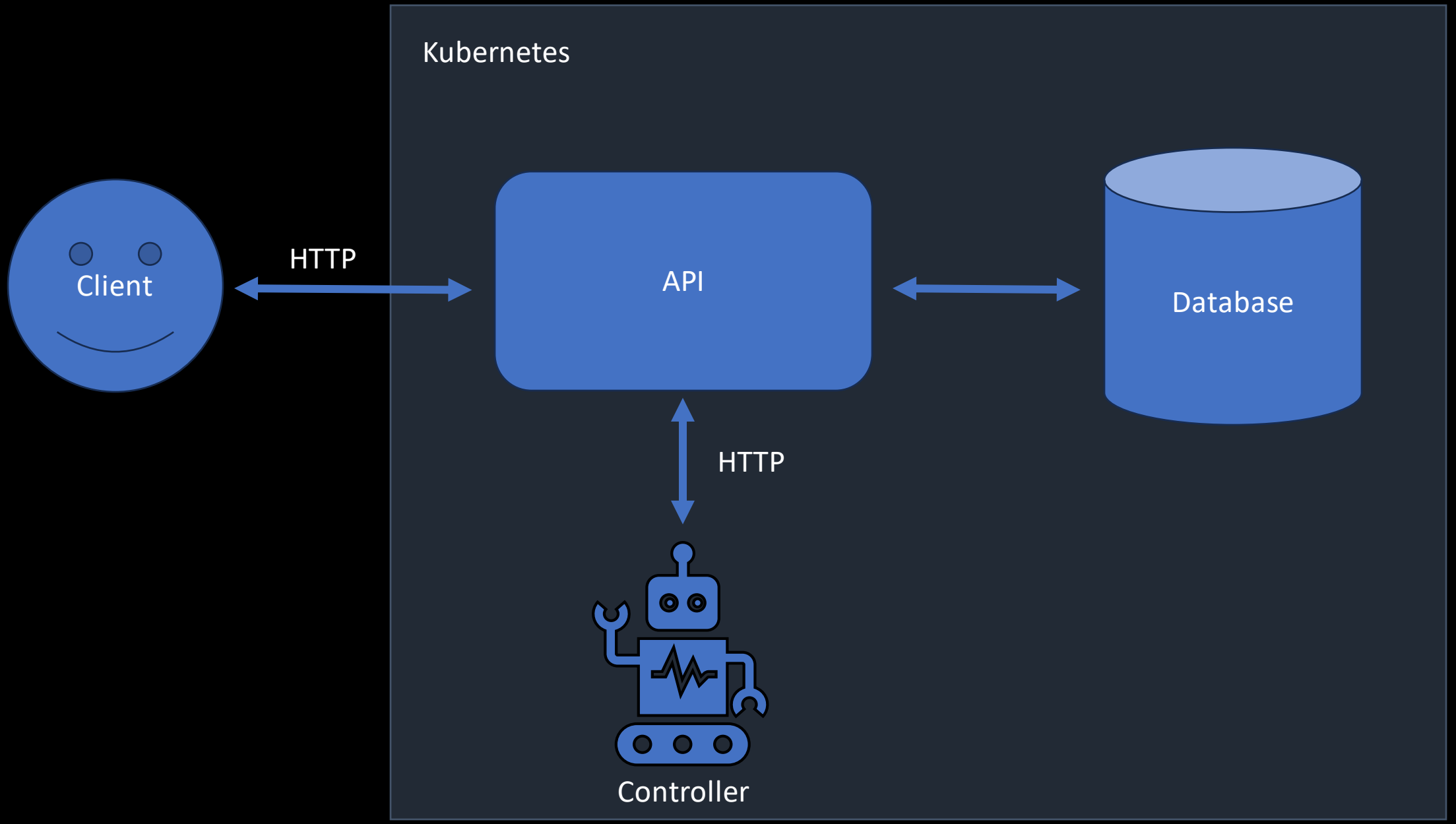
"Reconciliation"

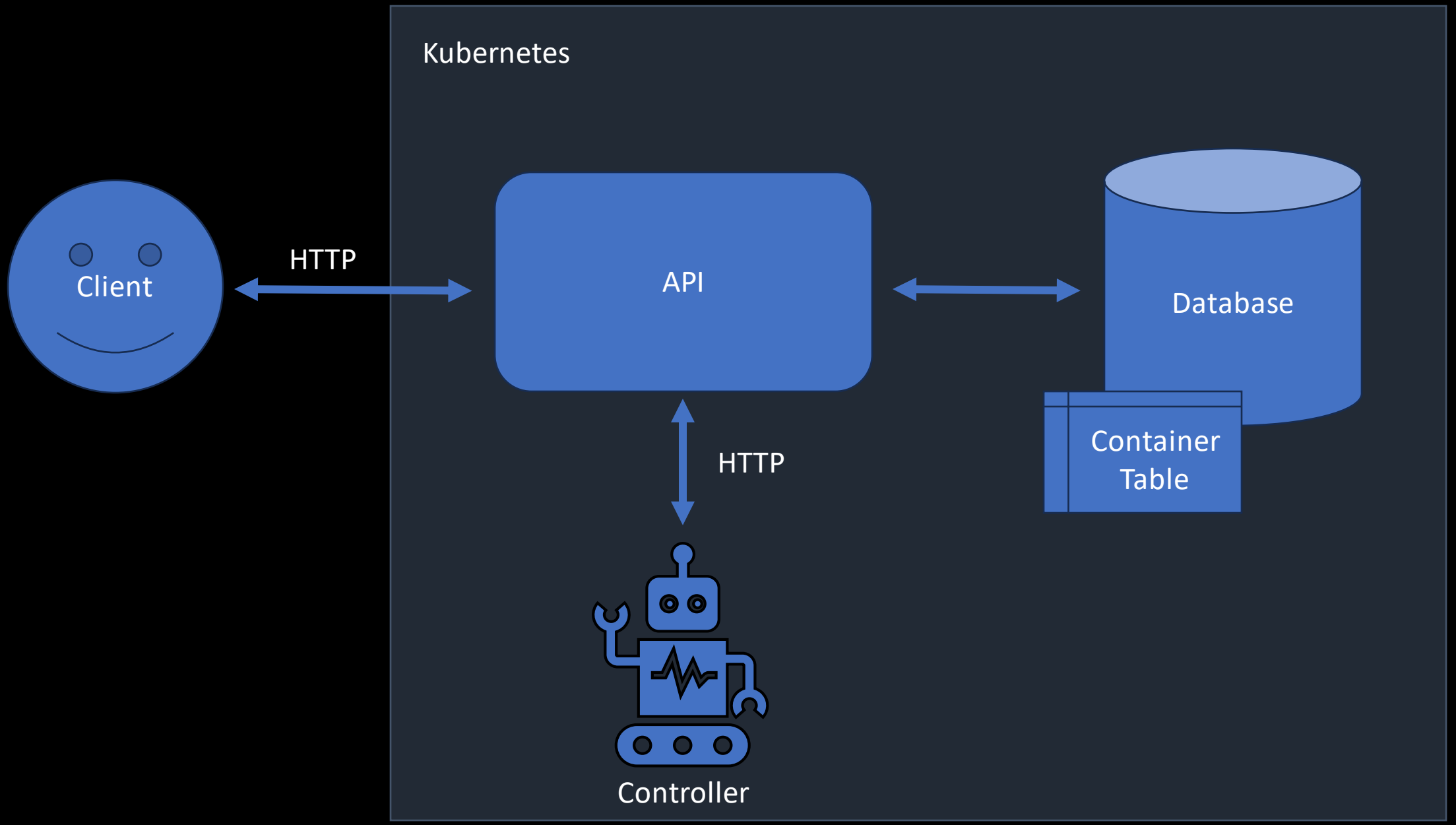


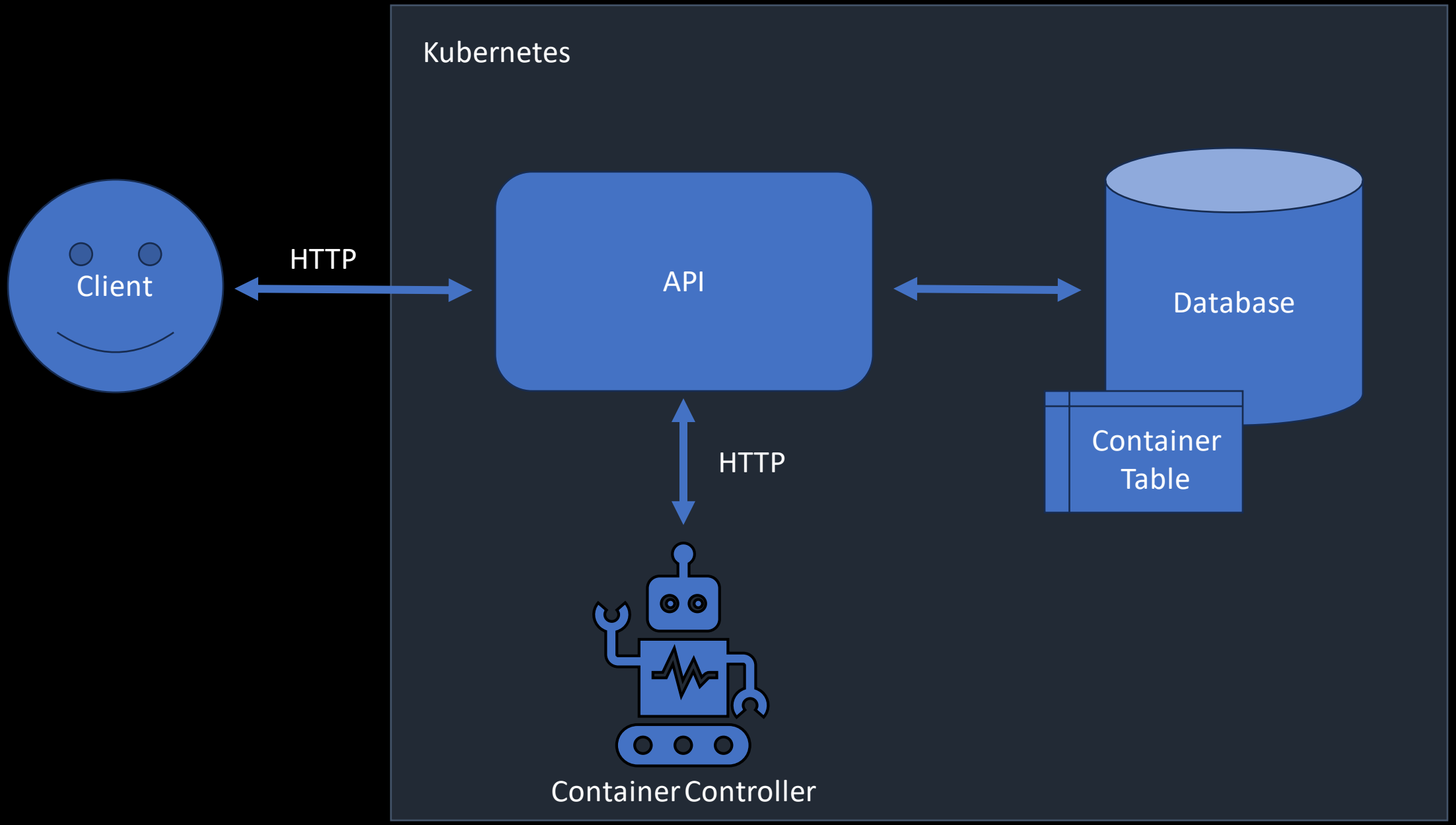
Controller

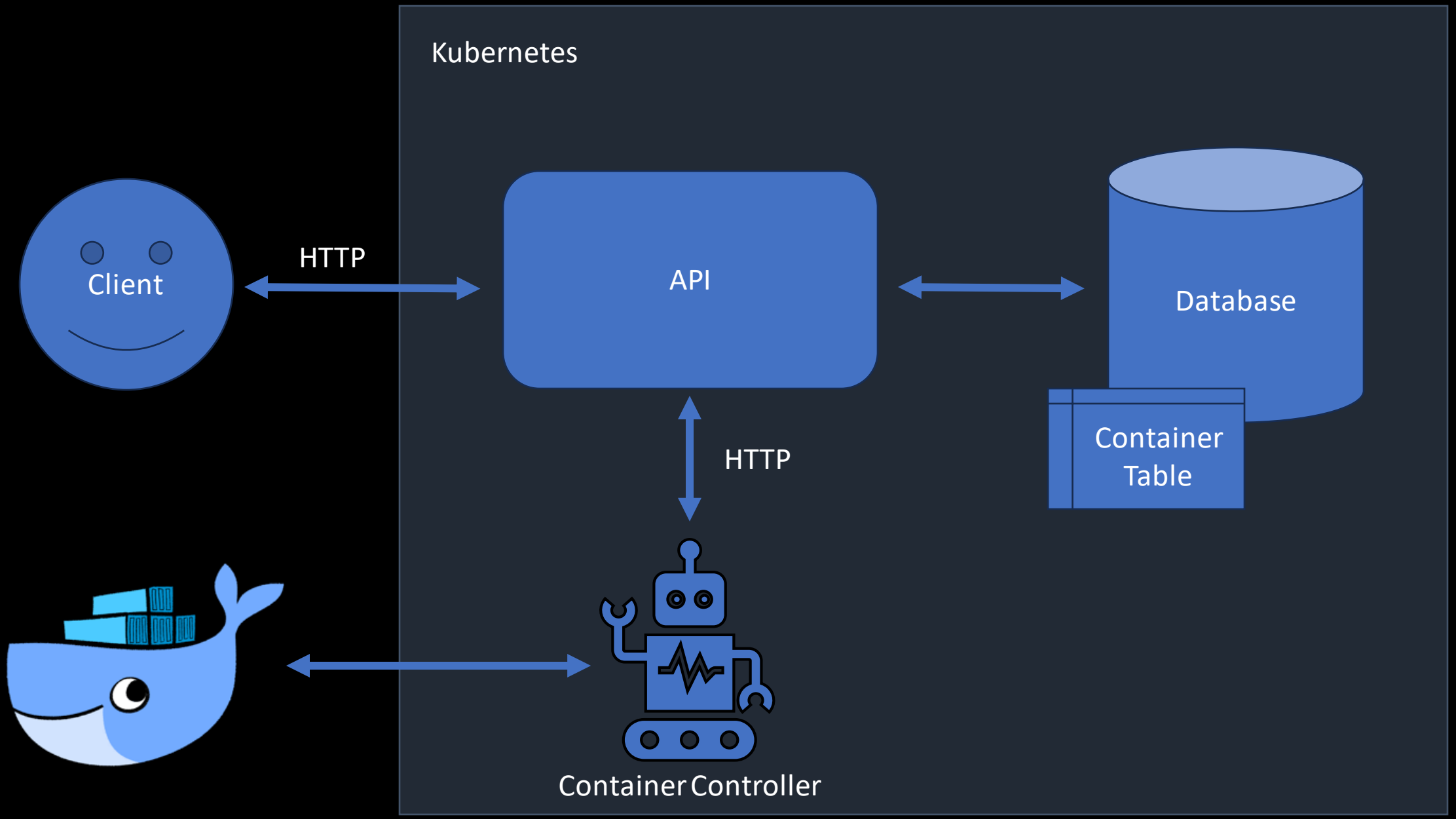
"Reconciliation"

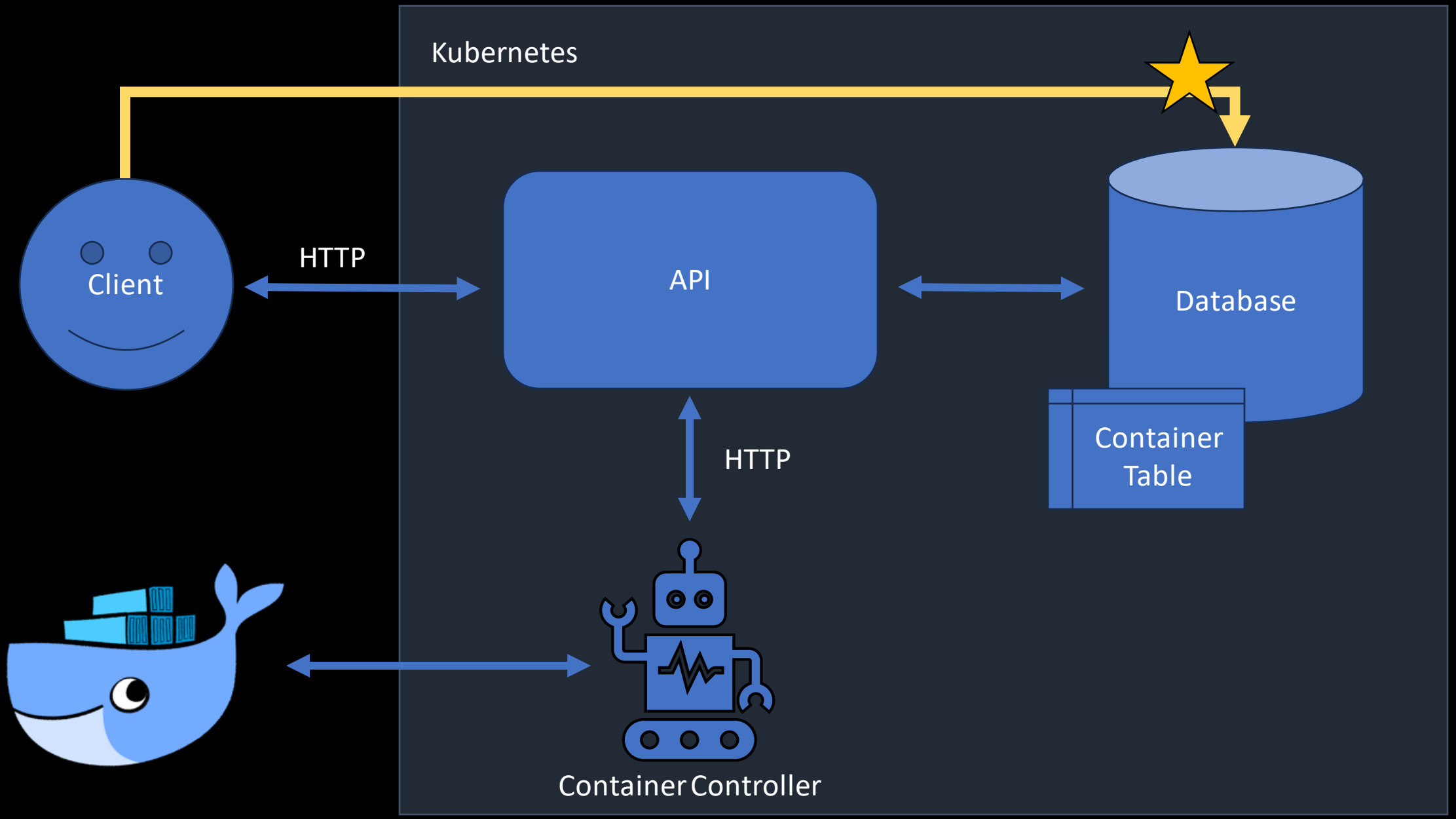


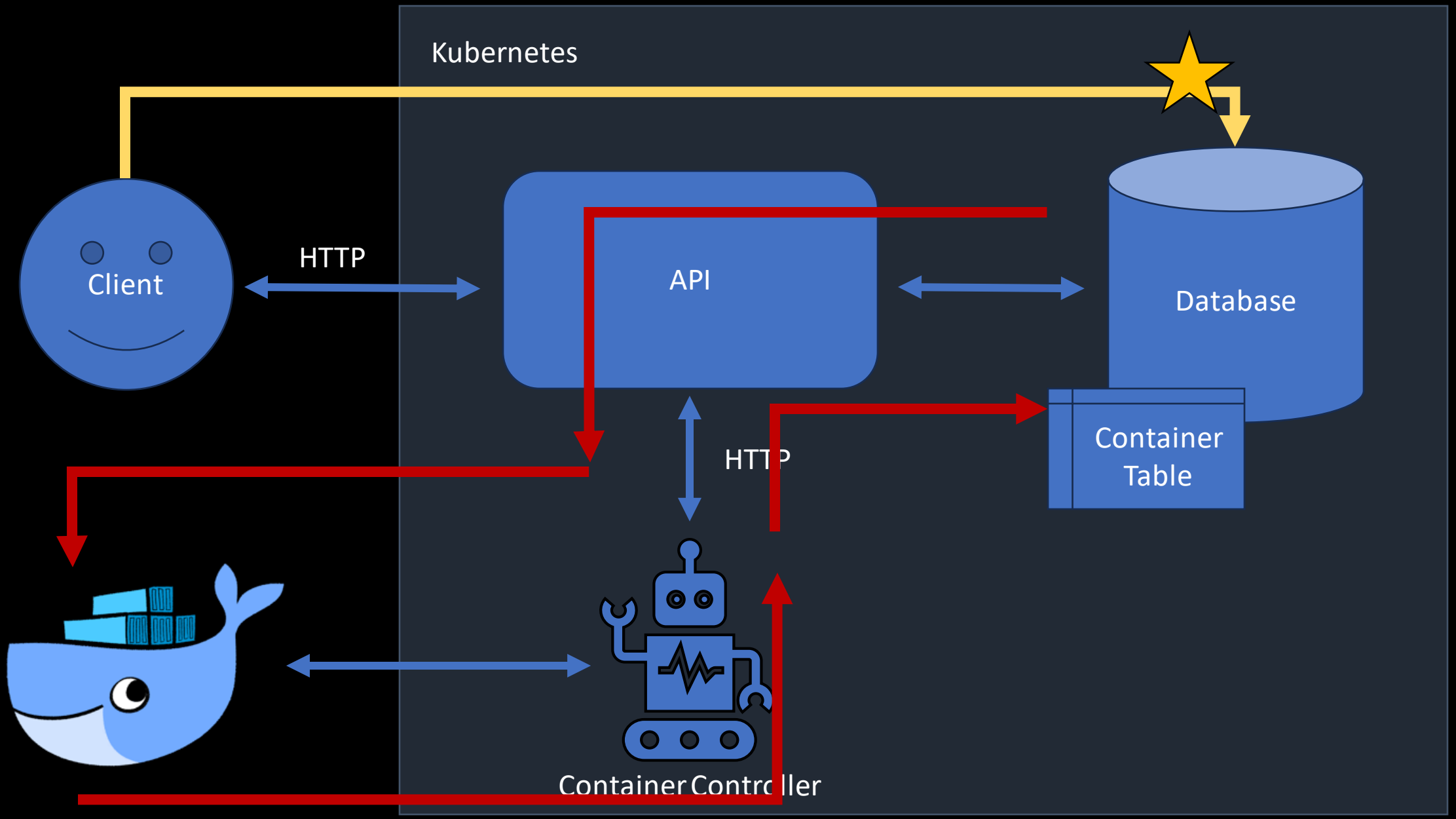

Kubernetes
Client
HTTP
API
Database
HTTP
Controller


Kubernetes
Client
HTTP
API
Database
Container
Table
HTTP
Controller

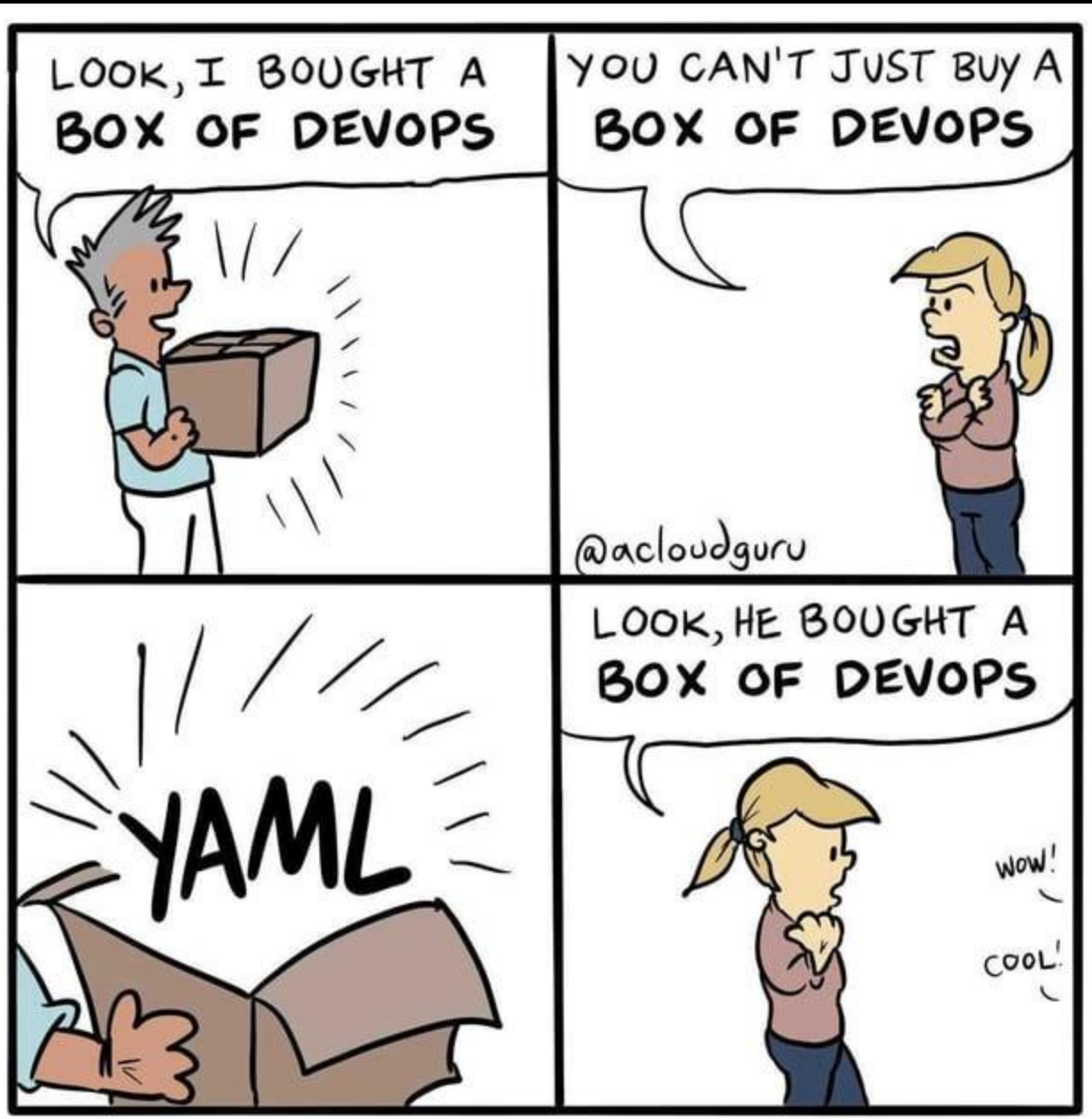

Kubernetes
Client
HTTP
API
Database
Container
Table
HTTP
Container Controller

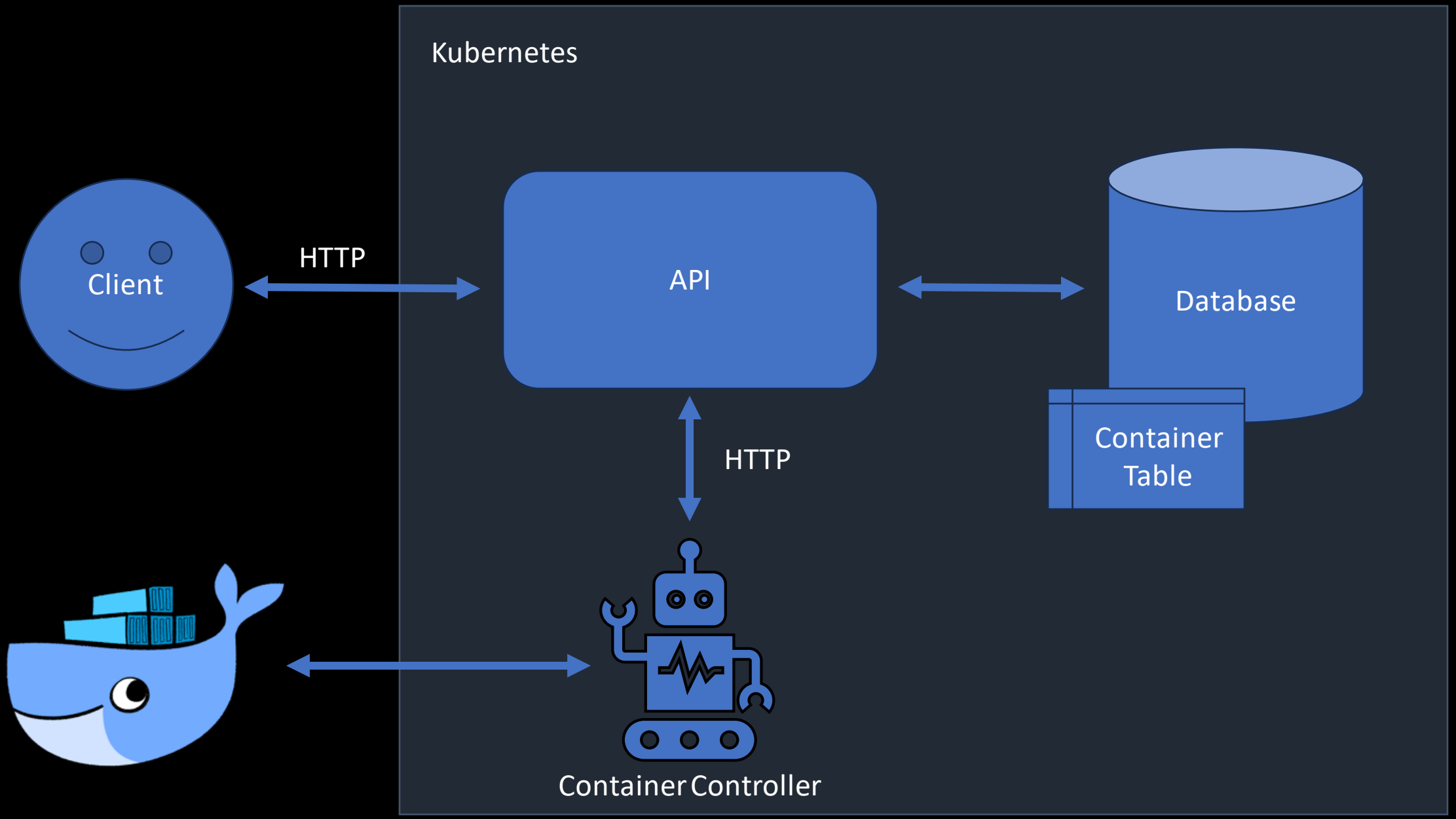

Kubernetes
Client
HTTP
API
Database
Container
Table
HTTP
Container Controller

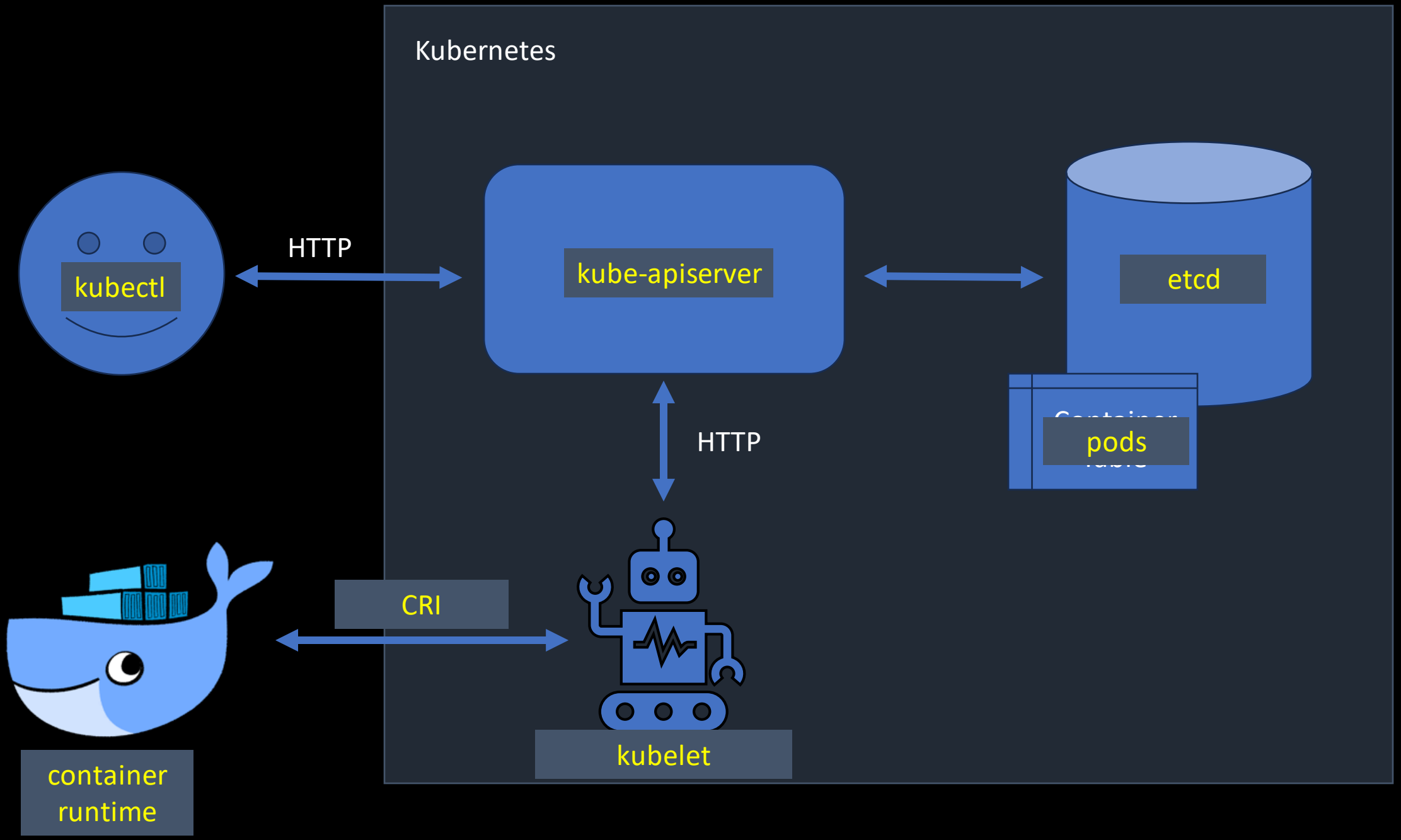

Kubernetes
Client
HTTP
API
Database
Container
Table
HTTP
Container Controller

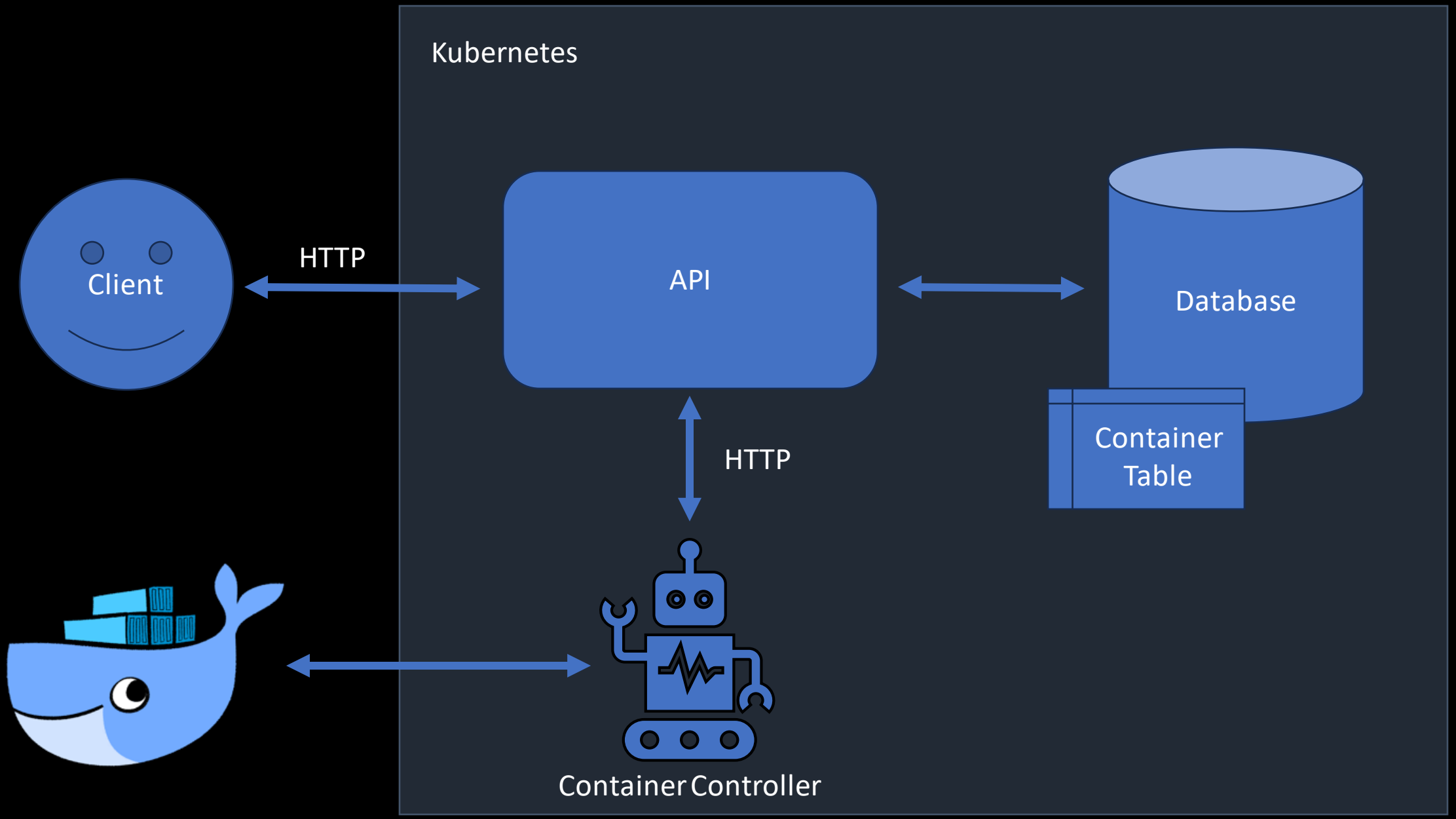

Kubernetes
Client
HTTP
API
Database
HTTP
Container
Table
Container Controller

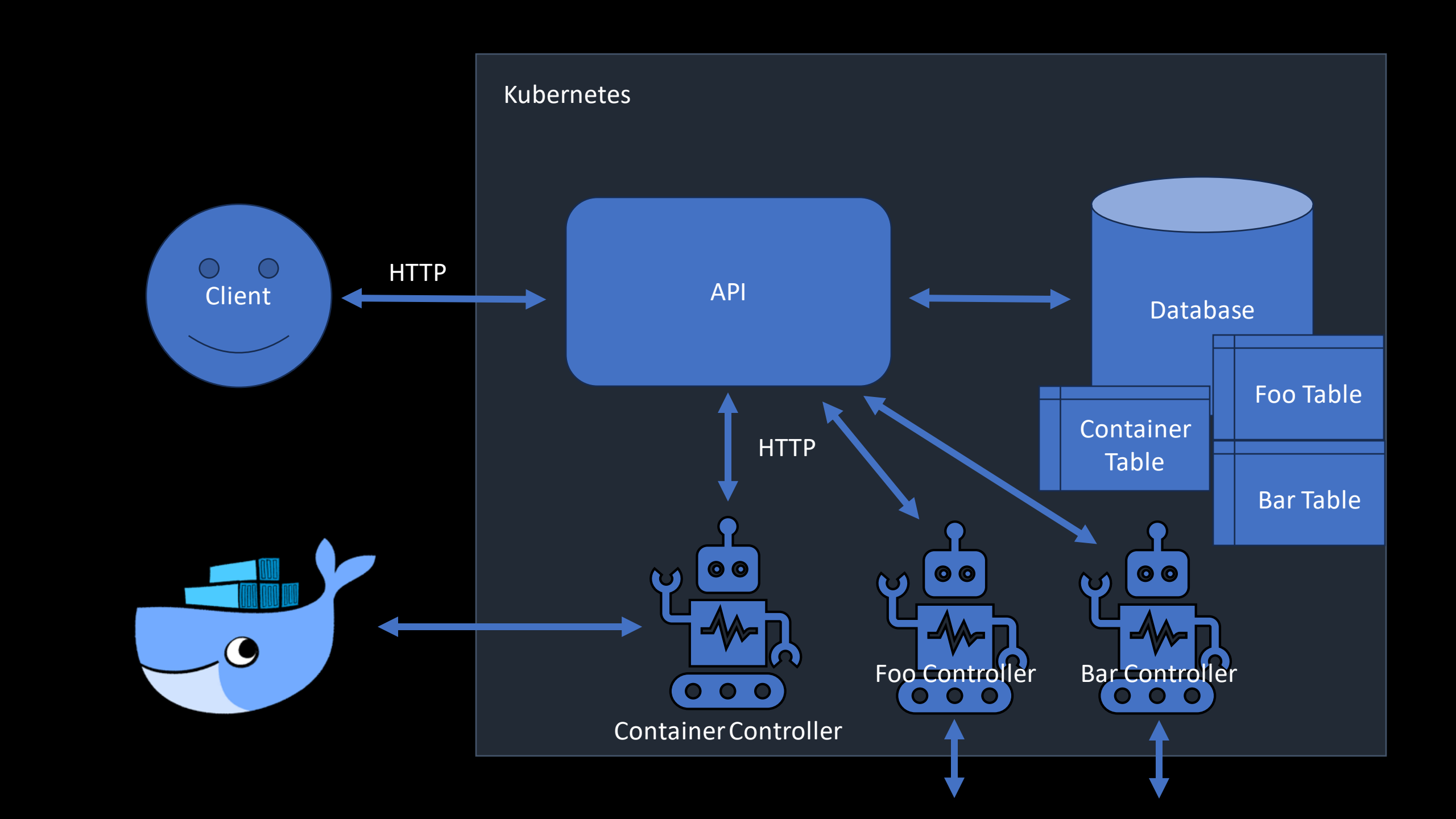
Demo: Simplenetes

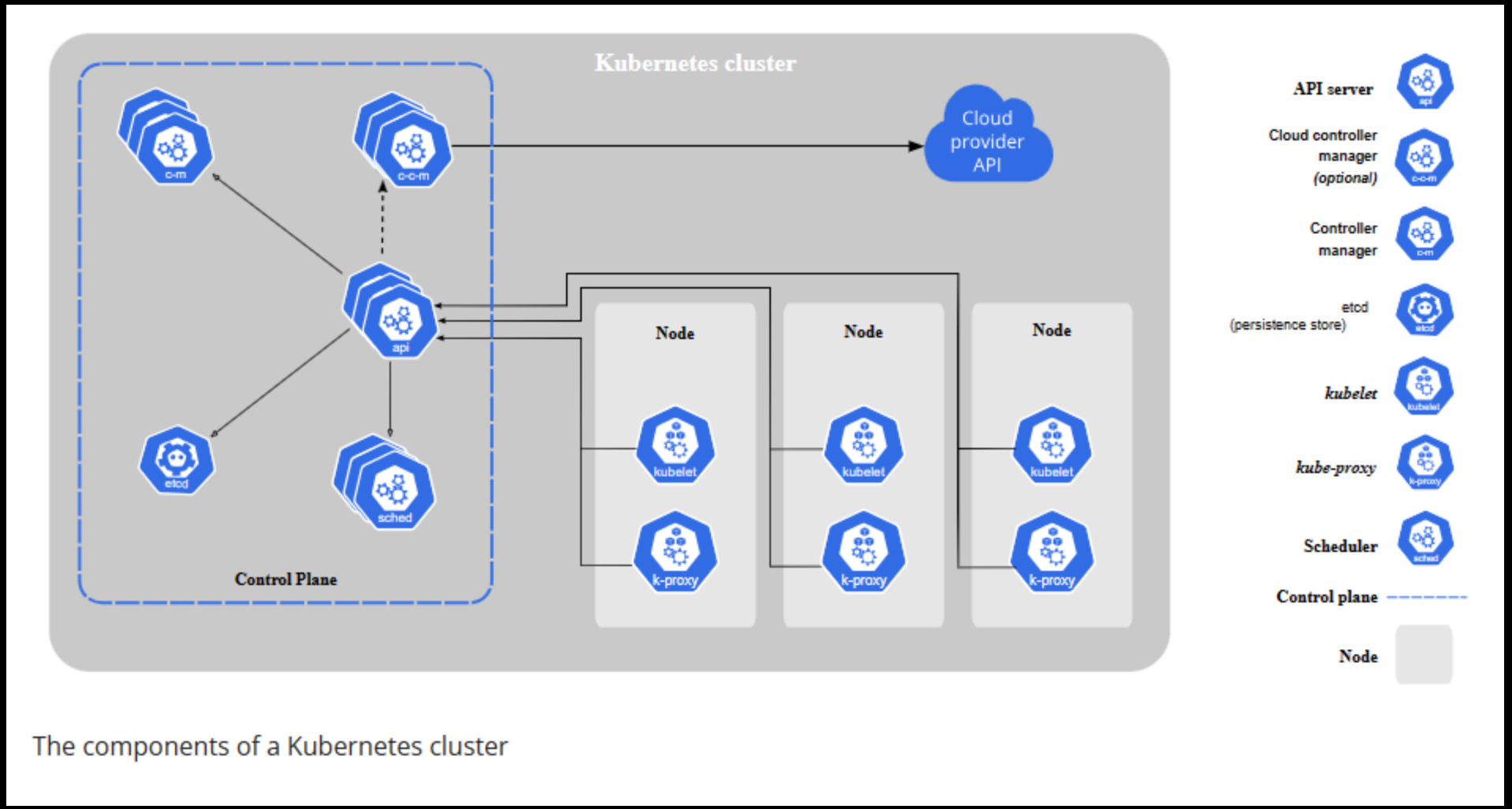

LOOK, I BOUGHT A BOX OF DEVOPS
YOU CAN'T JUST BUY A BOX OF DEVOPS
@acloudguru
YAML
LOOK, HE BOUGHT A BOX OF DEVOPS
WOW!
COOL!


Kubernetes
Client
HTTP
API
Database
Container
Table
HTTP
Container Controller


Kubernetes
kubectl
HTTP
kube-apiserver
etcd
HTTP
pods
CRI
container runtime
kubelet


Kubernetes
Client
HTTP
API
Database
Container
Table
HTTP
Container Controller


Kubernetes
Client
HTTP
API
Database
Container Table
Foo Table
Bar Table
HTTP
Container Controller
Foo Controller
Bar Controller



The components of a Kubernetes cluster

[Kubernetes Components | Kubernetes]


kubernetes
11004
kubernetes
4+
11004
@datamattsson
IDEAS INCLUDED
Pod
Service
StatefulSet
DaemonSet
Deployment

Pod

The controller manages groups of containers

(like docker compose)

Service

The controller manages network routing rules and DNS records

Deployment

The controller manages pod replicas and rolls out changes to pods

PersistentVolumeClaim

The controller provisions storage for a pod

Ingress

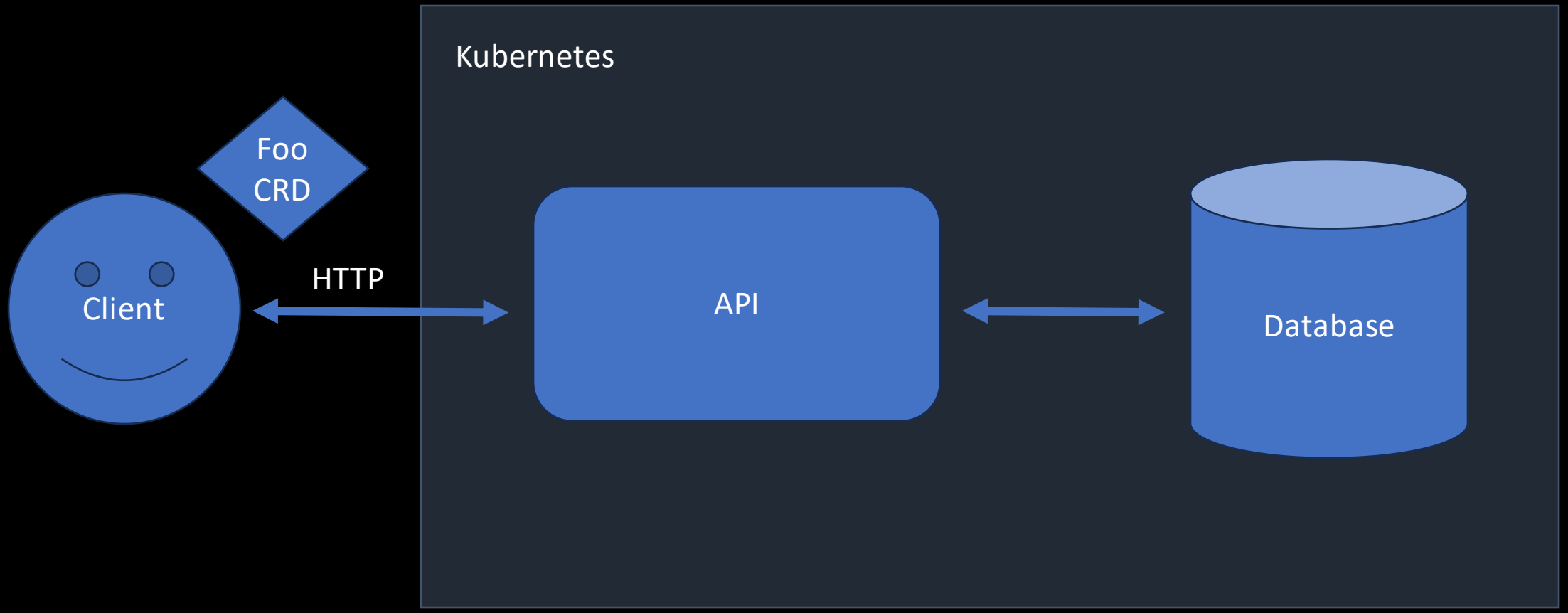
The controller configures the application that routes traffic received from outside the cluster

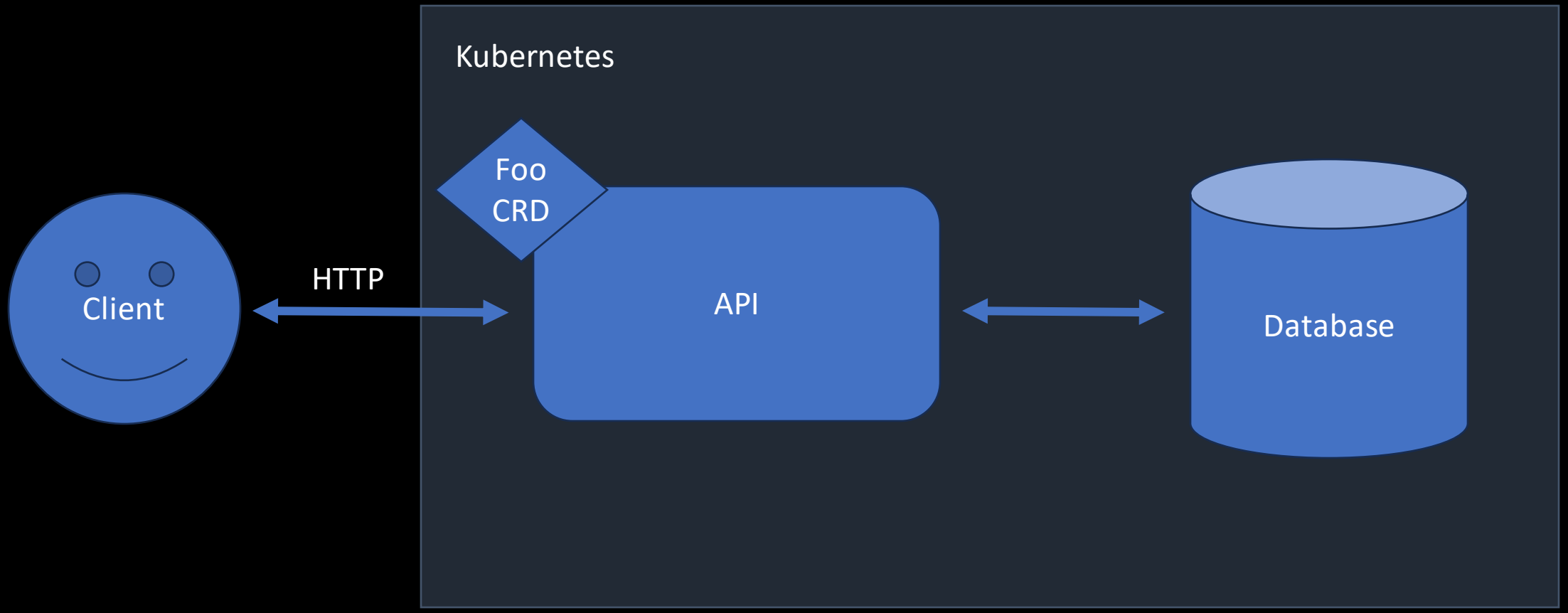
(example: nginx config)

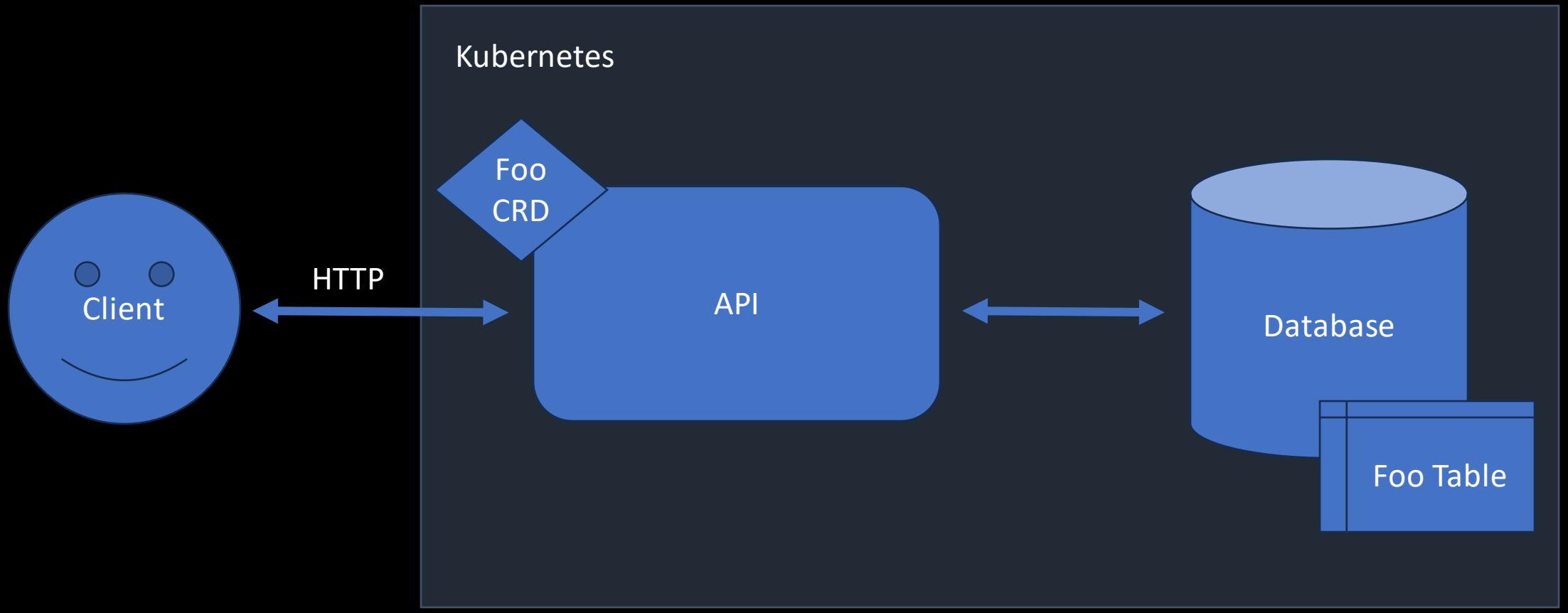

AH YES, MY FAVORITE
KUBERNETES RESOURCE
EMPLOYMENT

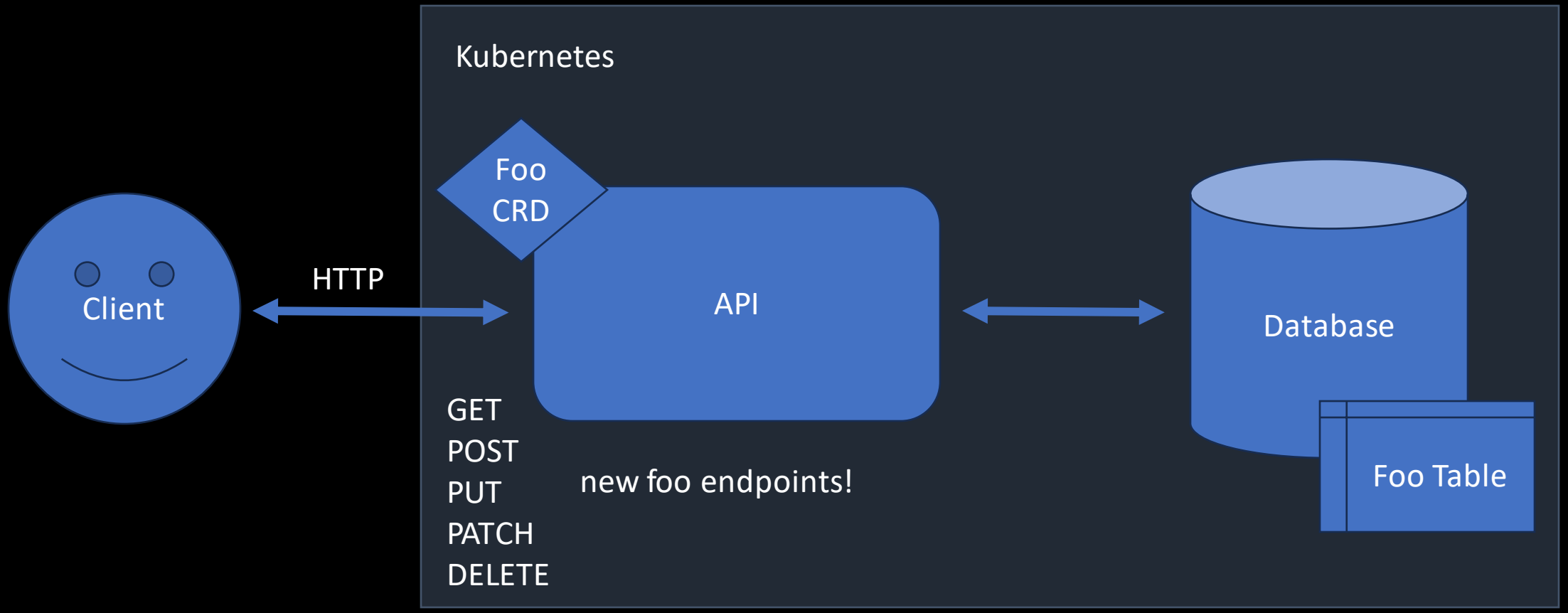
CustomResourceDefinition (CRD)

The controller creates a table in the database and exposes CRUD API endpoints for a new resource

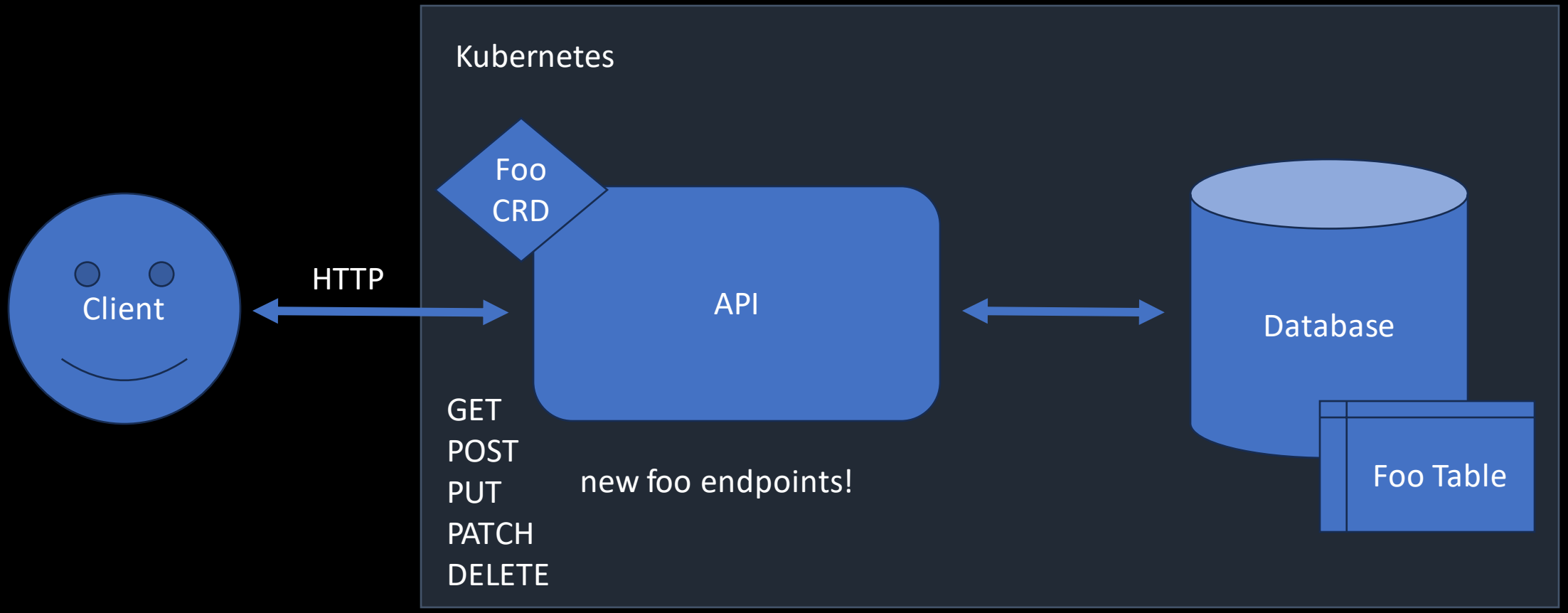

Kubernetes
Foo
CRD
Client
HTTP
API
Database


Kubernetes
Foo
CRD
Client
HTTP
API
Database


Kubernetes
Foo
CRD
Client
HTTP
API
Database
Foo Table

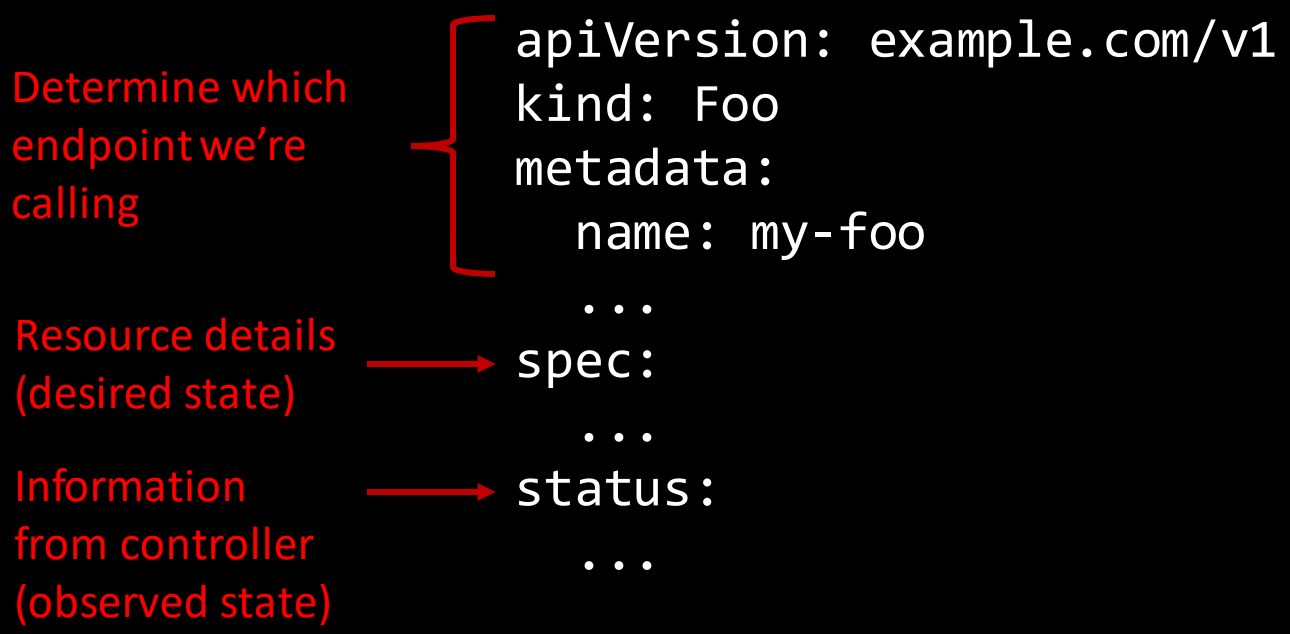

Kubernetes
Foo
CRD
Client
HTTP
API
Database
Foo Table
GET
POST
PUT
PATCH
DELETE
new foo endpoints!

The Kubernetes API is dynamic -- it can change at runtime!



OpenAPI v3 Schema

OpenAPI-Specification/versions/3.0.0.md at main · OAI/OpenAPI-Specification (github.com)


Determine which endpoint we're calling
Resource details (desired state)
Information from controller (observed state)
apiVersion: example.com/v1
kind: Foo
metadata:
name: my-foo
...
spec:
...
status:
...

Operator = CRD + Controller

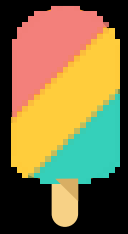
Example Operators

- cert-manager
- kube-prometheus-stack
- postgres-operator

Demo: Dice Roll Operator

The most complicated random number generator!

Kubernetes as a platform vs. Kubernetes as an API | Containers (amazon.com)



Crossplane

The Big Ideas

- Kubernetes is an API
- Clients provide the desired state
- Controllers perform actions so that actual state matches desired state
- The API can change at runtime

What’s next?

- Learn more resources
- Deploy some apps
- EKS, AKS, GKE, …
- Operator frameworks


source allies