

Curious Case of Streaming Platform on Kubernetes



@gamussa

#devoops

@confluentinc







#devkafkaops



<https://twitter.com/kelseyhightower/status/963413508300812295>



Kelsey Hightower ✓

@kelseyhightower

Following



Kubernetes can only meet stateful workloads half way and I lack the expertise to manage a production configuration of Kafka, RabbitMQ, or Postgres on static infrastructure, let alone a Kubernetes cluster.

9:06 AM - 13 Feb 2018

18 Retweets 71 Likes



2



18



71





Vik Gamov via @QuanBot

Чат, для [моего доклада](#) очень хочется понять целевую аудиторию. Помогите с опросом!

■ 28% (46) Я использую Kubernetes in production СЕГОДНЯ



■ 40% (65) Я считаю, что использовать Kubernetes in production СЕГОДНЯ хорошая идея



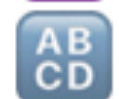
■ 9% (15) Я запускаю Stateful Workloads (Databases, Kafka, etc) в Kubernetes в production СЕГОДНЯ



■ 13% (22) Я считаю, что запускать Stateful Workloads в Kubernetes в production СЕГОДНЯ хорошая идея



■ 6% (11) Kubernetes/VM зло, только Bare metal, только hardcore!



Можно выбрать вариантов: 4

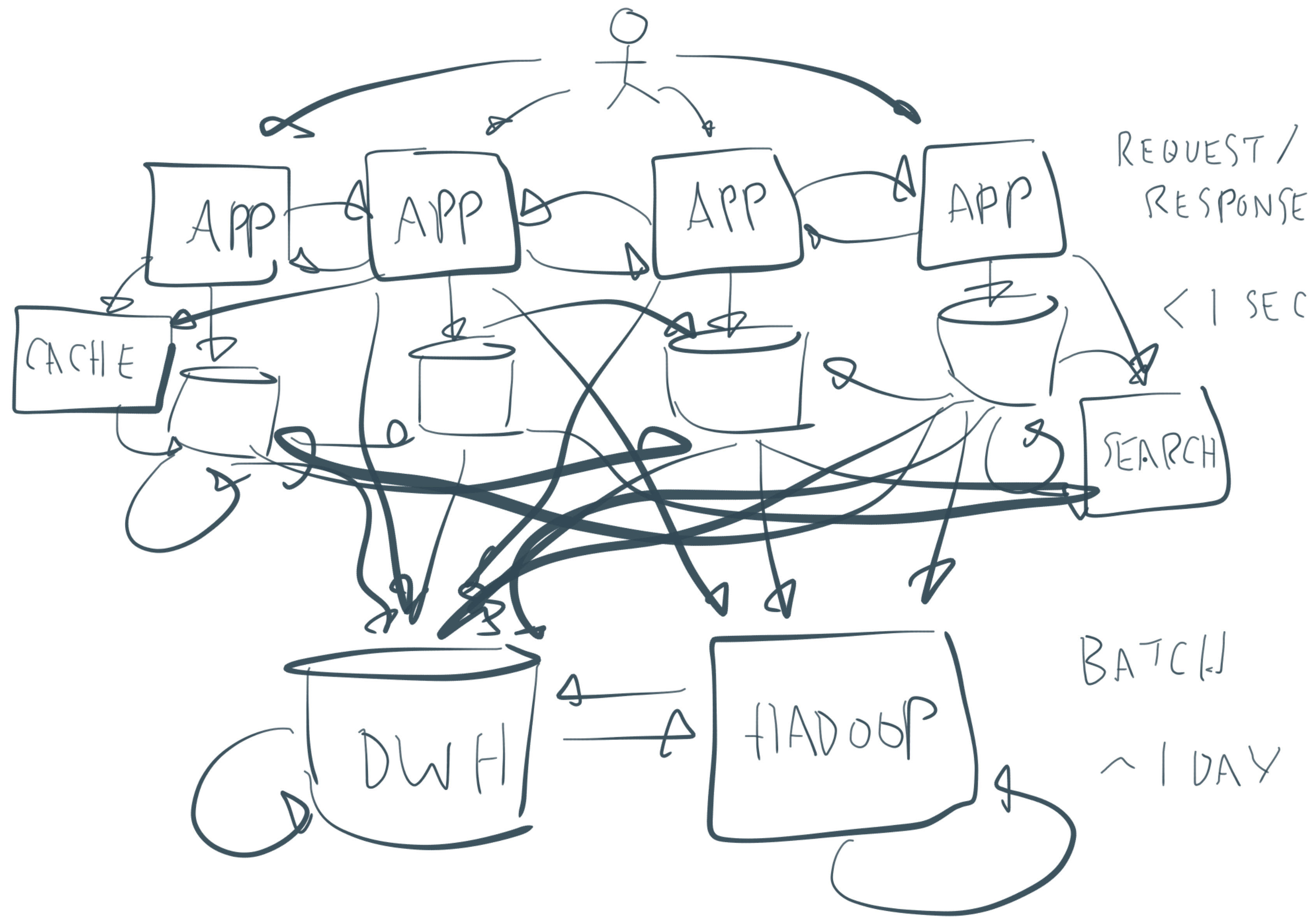


159 (123) - всего голосов

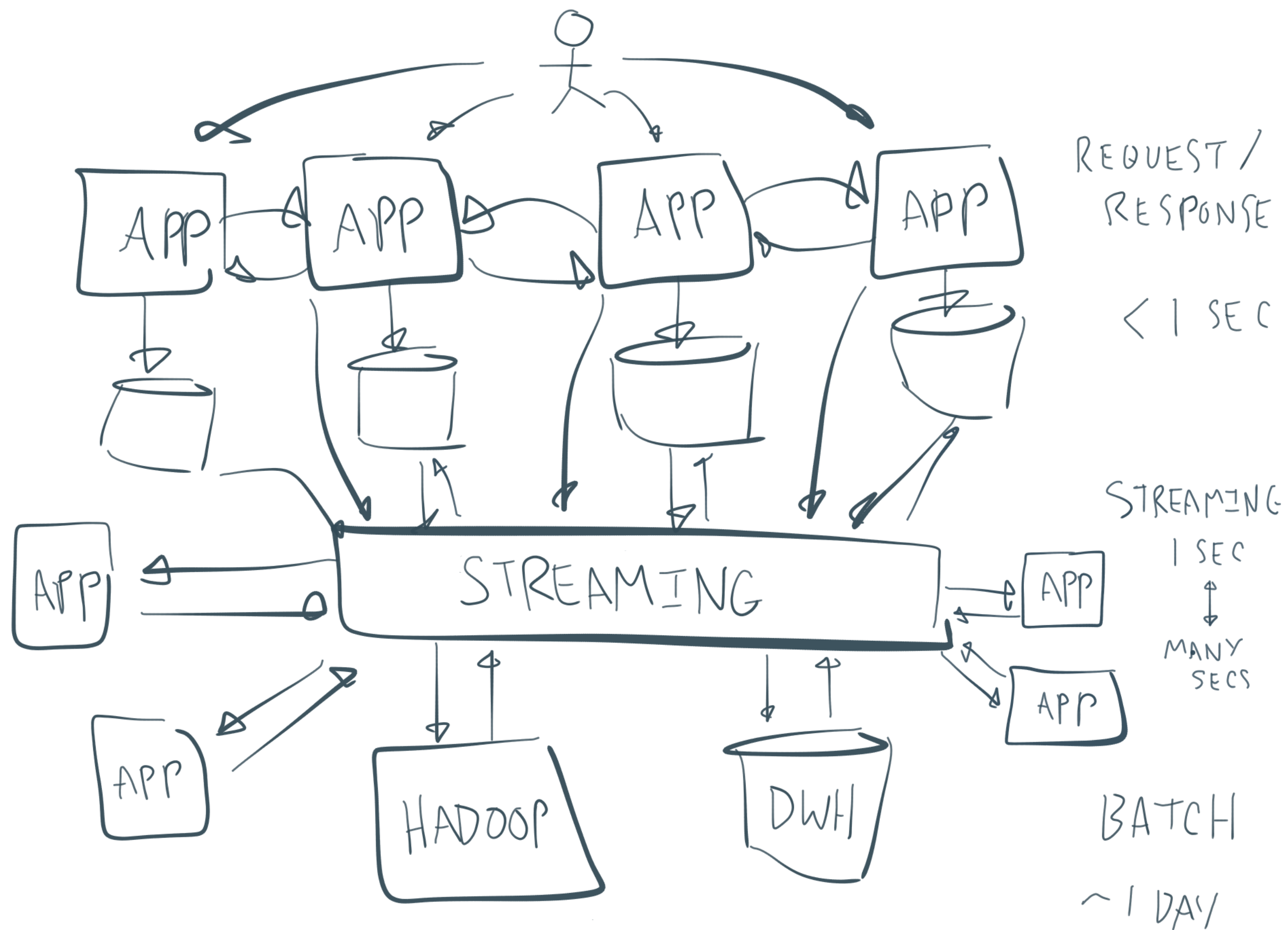


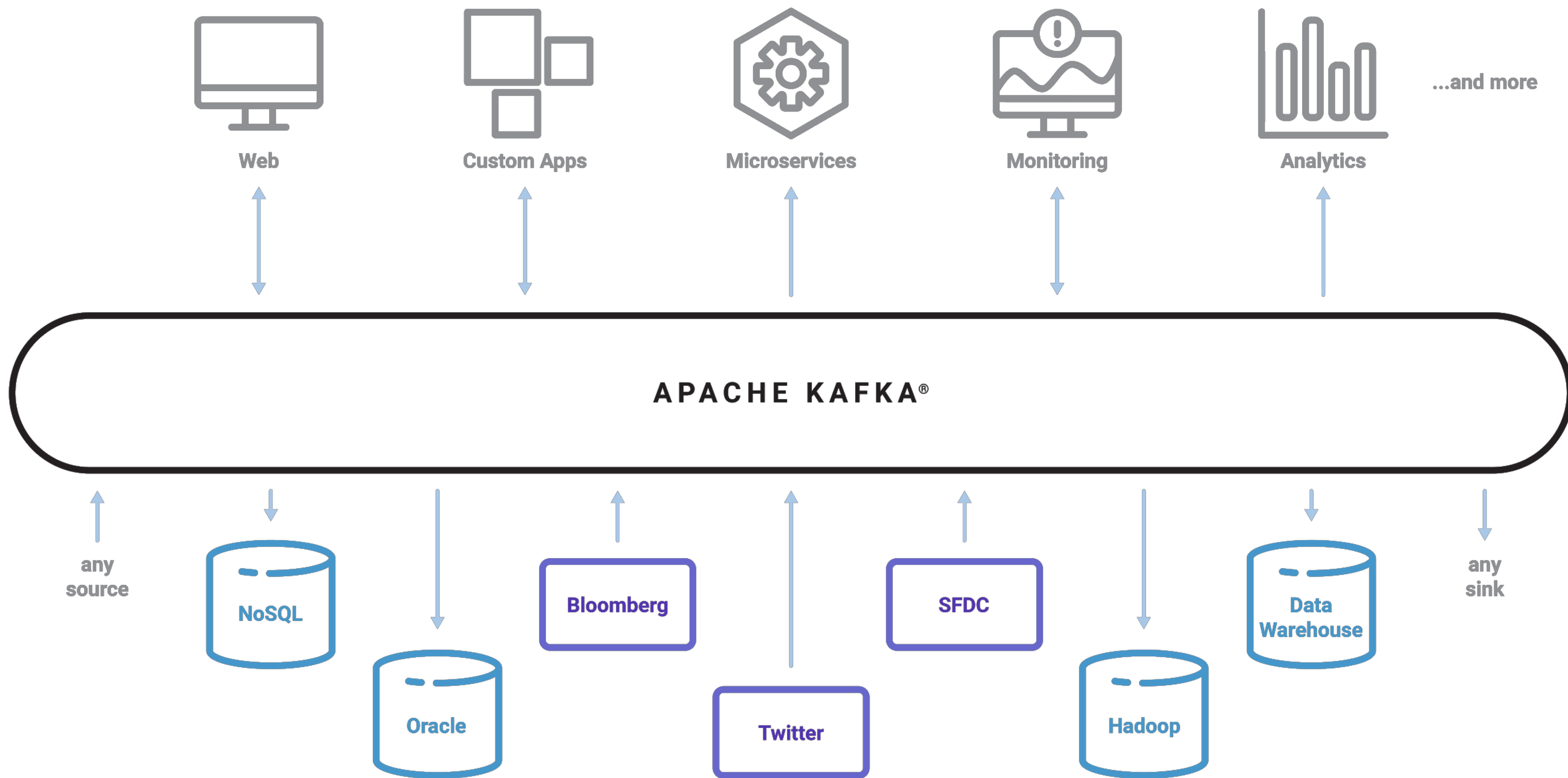
Why streams?

Tao of the streaming platform

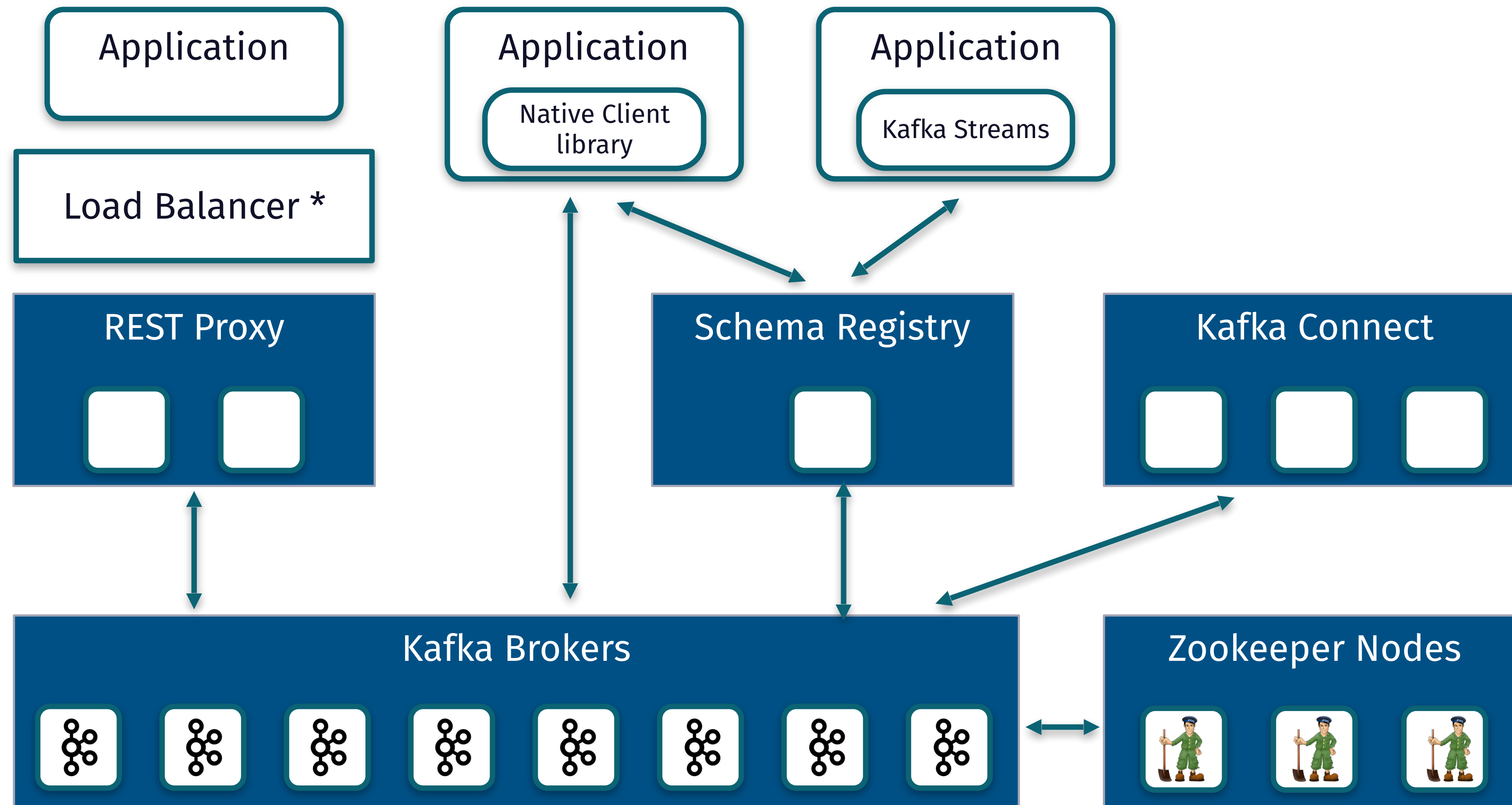




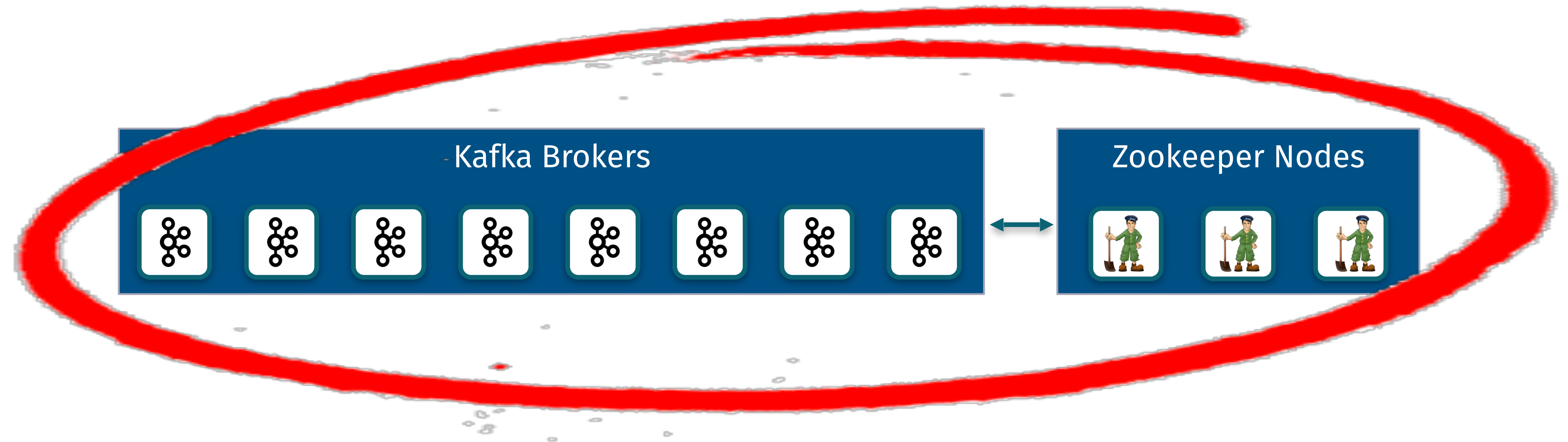




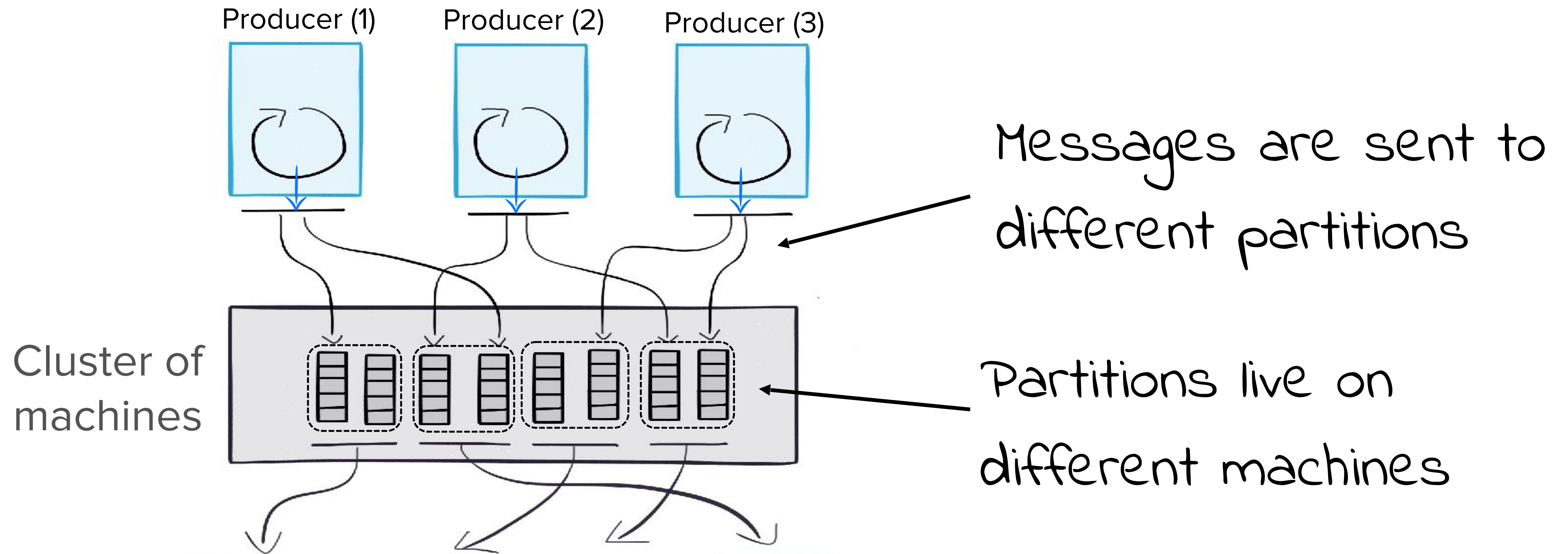
Streaming Platform Architecture



Bare minimum

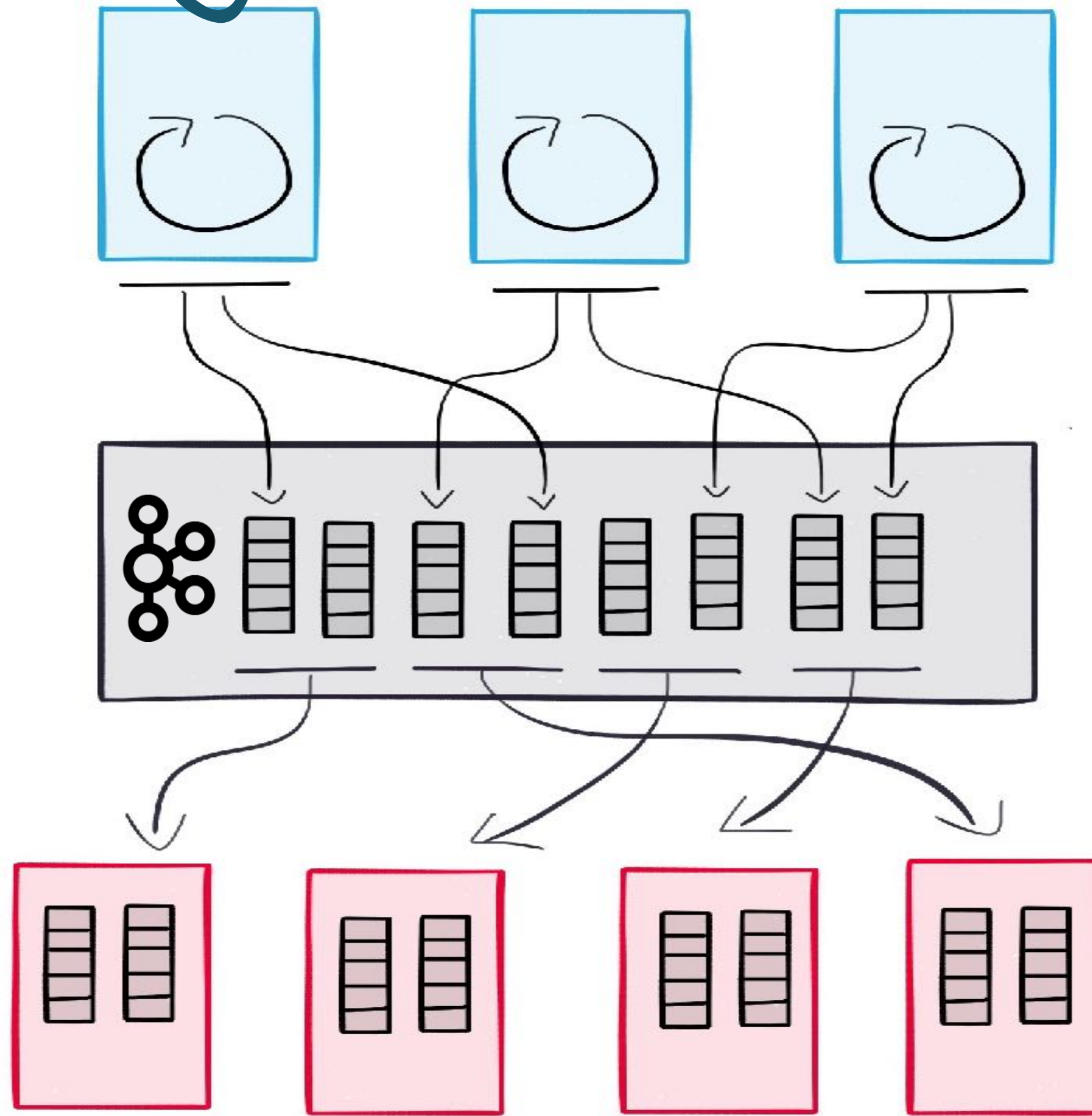


Shard data to get scalability



Linearly Scalable Architecture

Producers



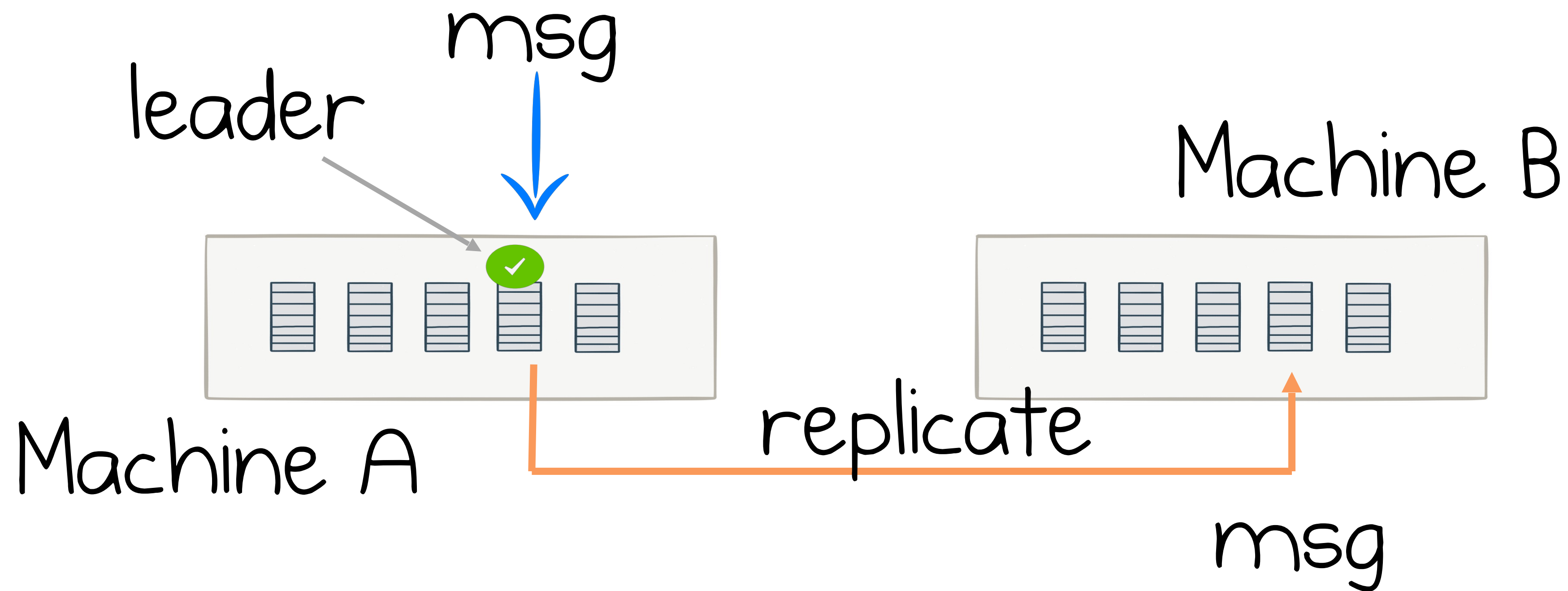
Consumers

Single topic:

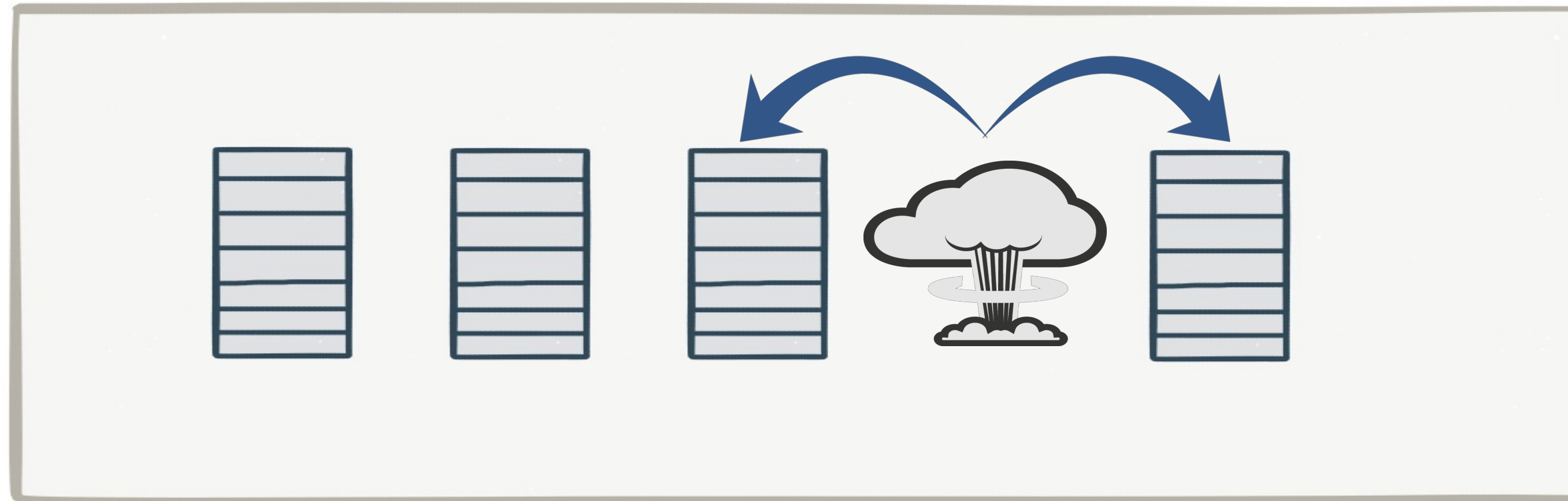
- Many producers machines
- Many consumer machines
- Many Broker machines

No Bottleneck!!

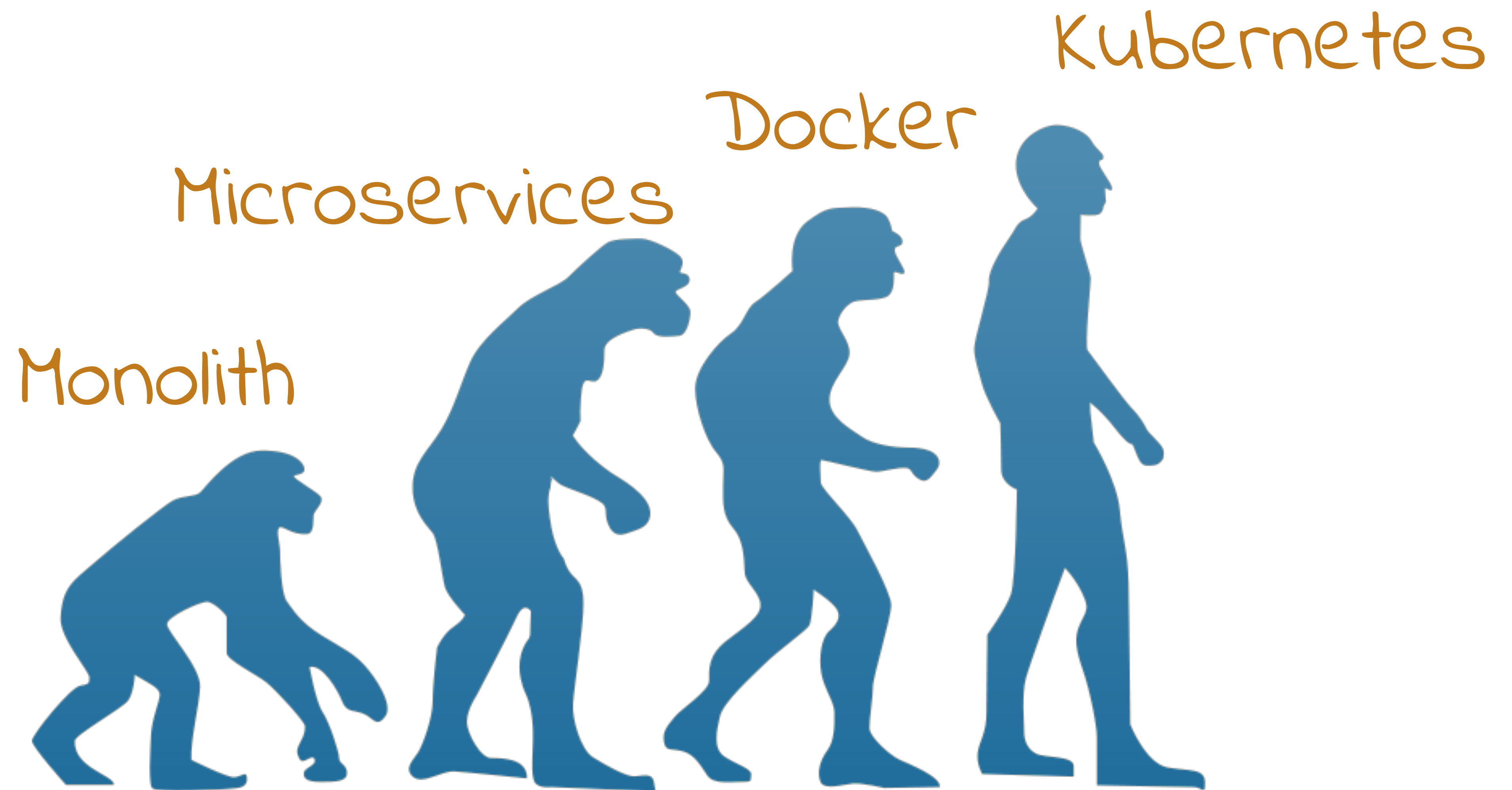
Replicate to get fault tolerance



Replication provides resiliency



A 'replica' takes over on machine failure





Syed Aqueel Haider

@sahrizv

Follow

2014 - We must adopt [#microservices](#) to solve all problems with monoliths
2016 - We must adopt [#docker](#) to solve all problems with microservices
2018 - We must adopt [#kubernetes](#) to solve all problems with docker

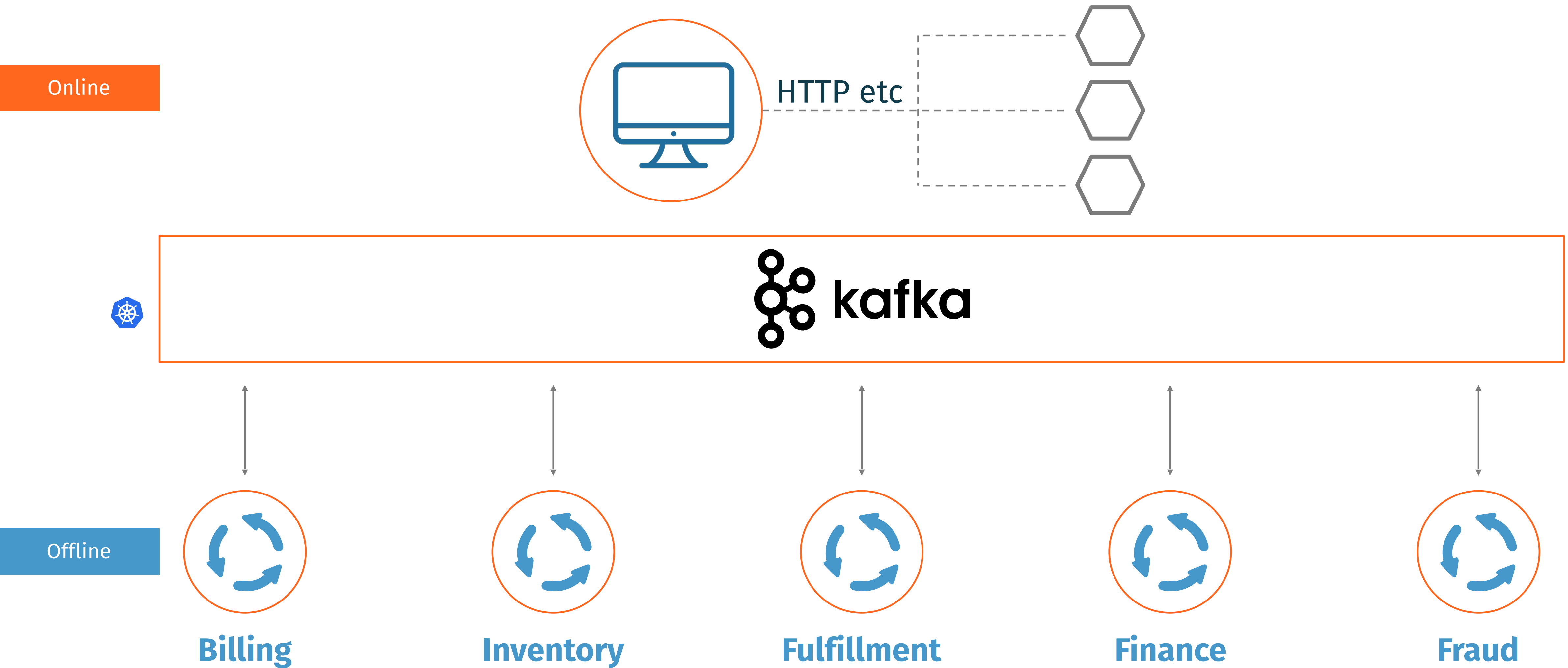


1:25 PM - 14 Jul 2018

1,622 Retweets 2,263 Likes



Kafka as Microservices Backbone

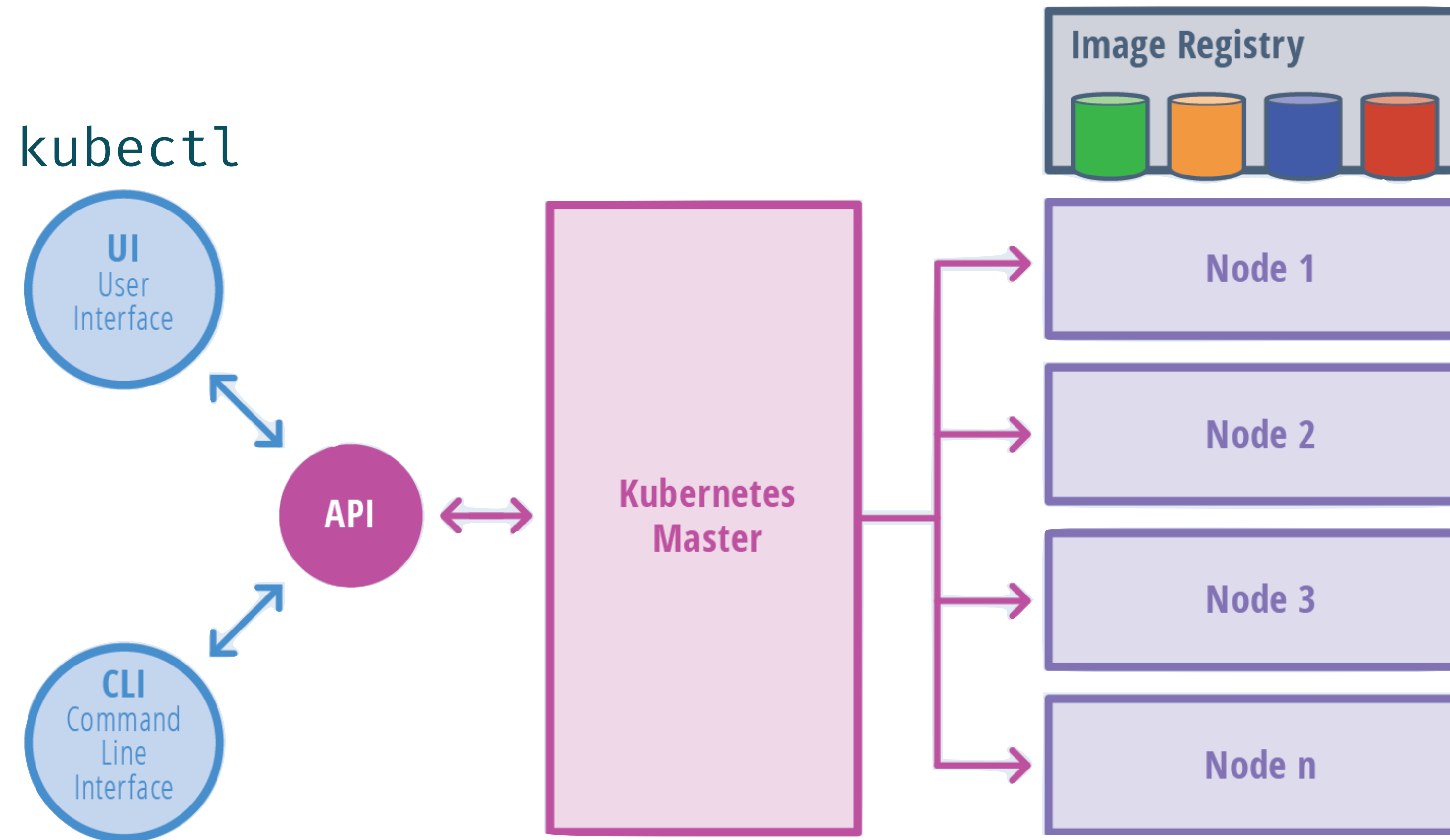




Stateful Workloads

Key Concepts

Refresher - Kubernetes Architecture



Source: Janakiram VSV

THE NEW STACK

<https://thenewstack.io/kubernetes-an-overview/>

@gamussa

#devoops

@confluentinc

Controller

Brain behind Kubernetes

resources

e.g. replication

controller, namespace

controller etc.



Pod

Basic Unit of Deployment
in Kubernetes

A collection of containers
sharing:

Namespace

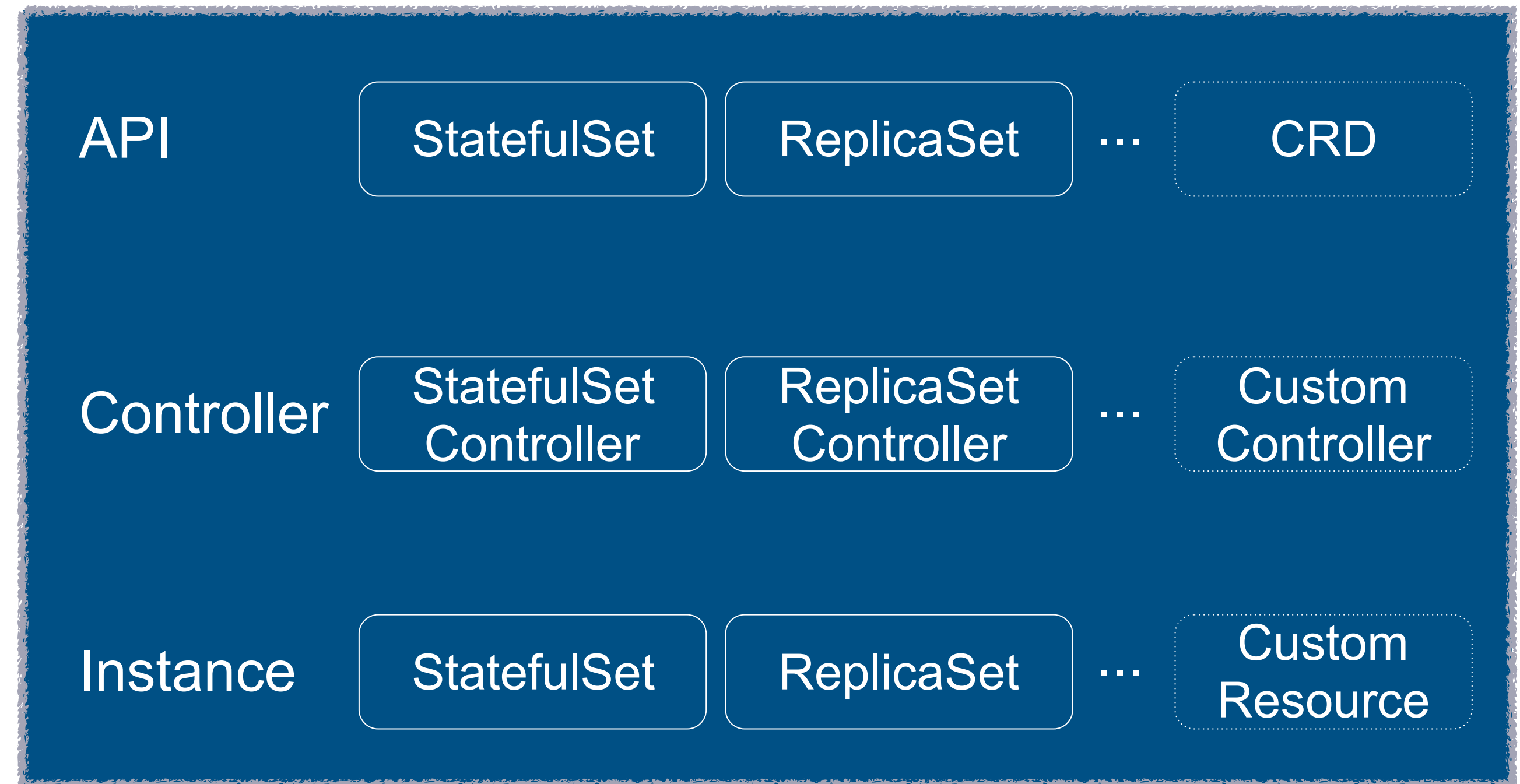
Network

Volumes



Custom Resource Definition(CRD)

Extend existing
Kubernetes API

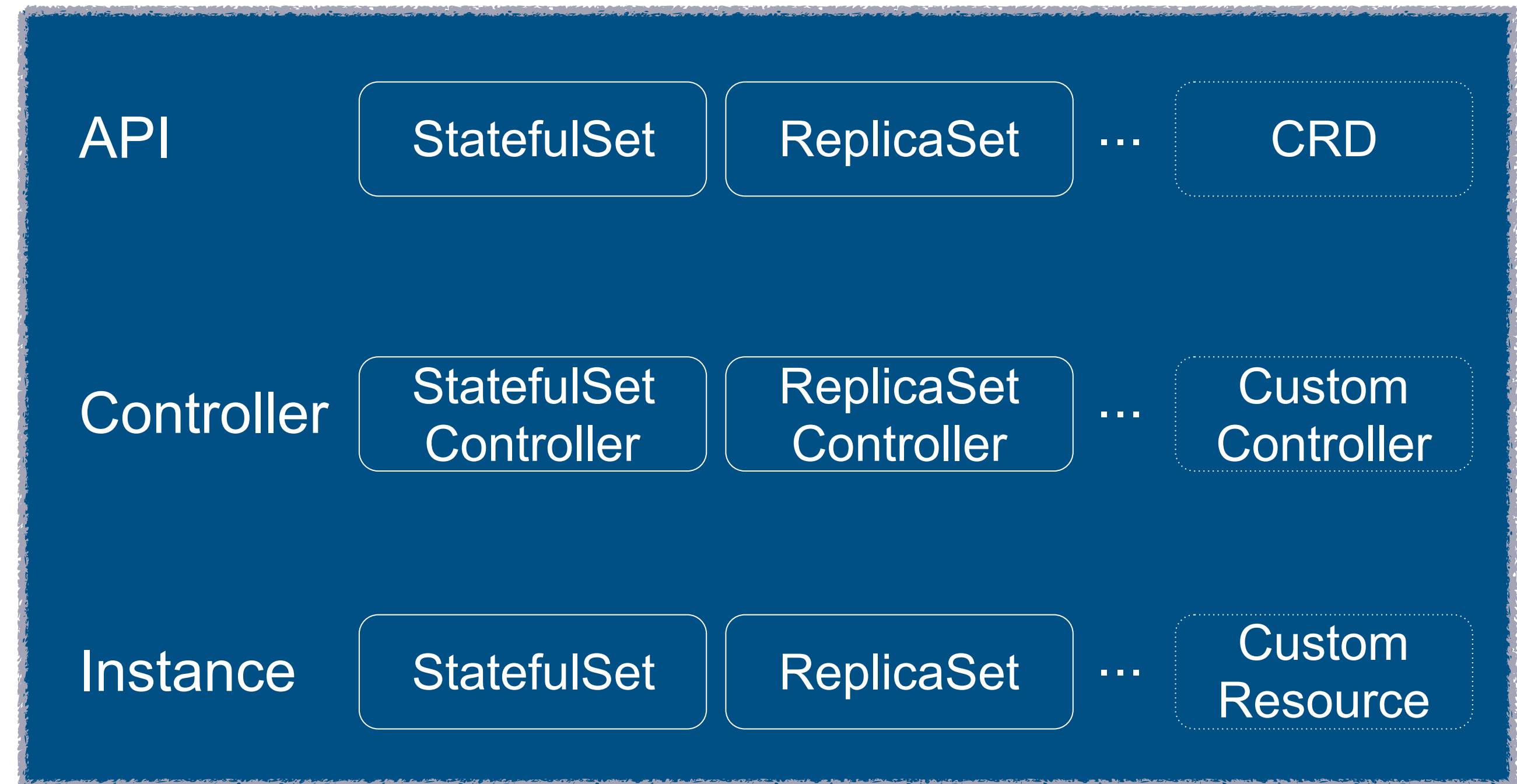


Custom Resource Definition(CRD)

usually works together

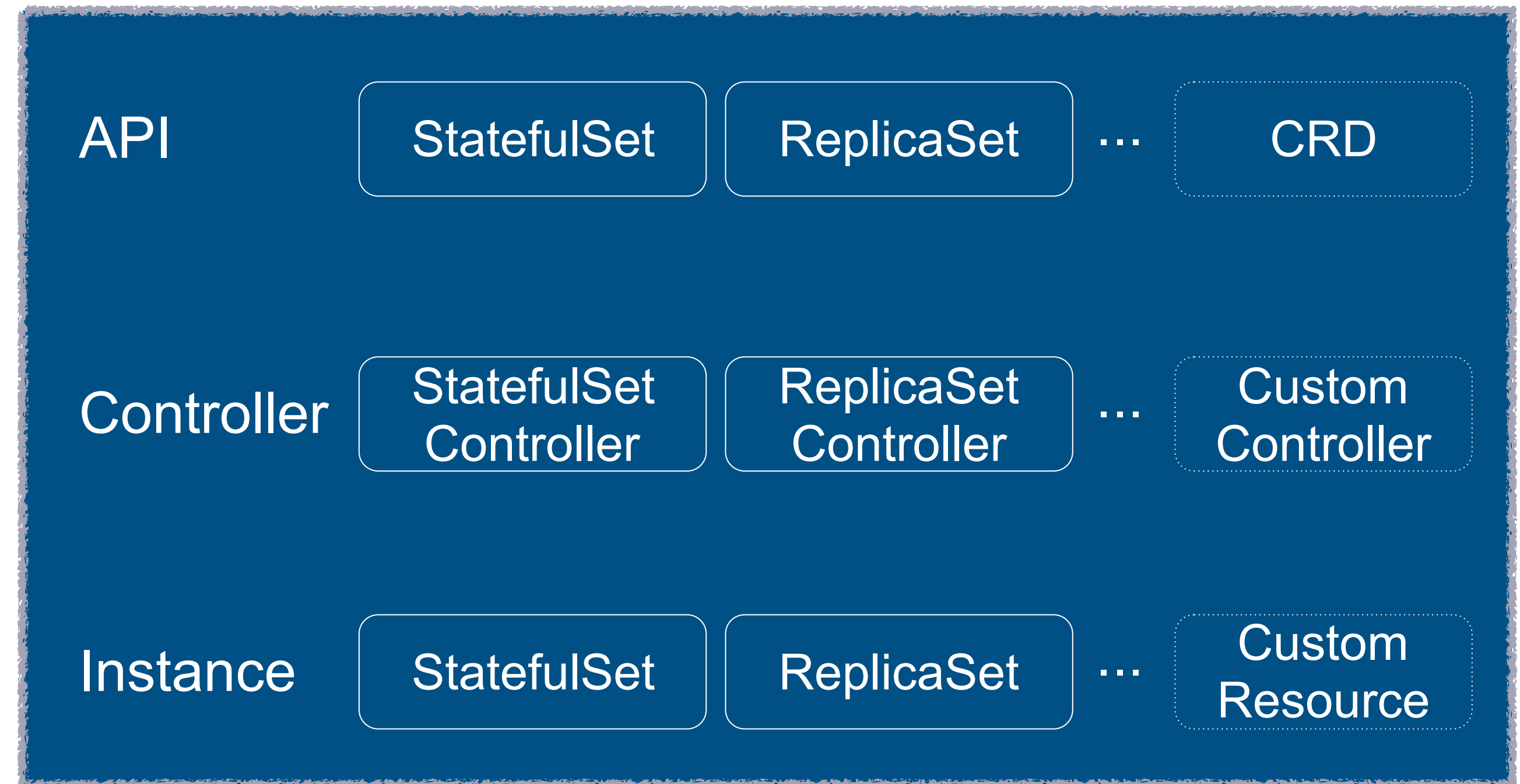
Custom Controller

(operator pattern)



Custom Resource Definition(CRD)

Users can create and access Custom Resources with kubectl, just as they do for built-in resources like pods.



Storage

Persistent Volume (PV) &
Persistent Volume Claim (PVC)

Both PV and PVC are
'resources'



Storage

Persistent Volume (PV) &
Persistent Volume Claim (PVC)

PV is a piece of storage
that is provisioned dynamic
or static of any individual
pod that uses the PV



Storage

Persistent Volume (PV) &

Persistent Volume Claim (PVC)

PVC is a request for storage
by a user



Storage

Persistent volume (PV) &
Persistent volume Claim (PVC)

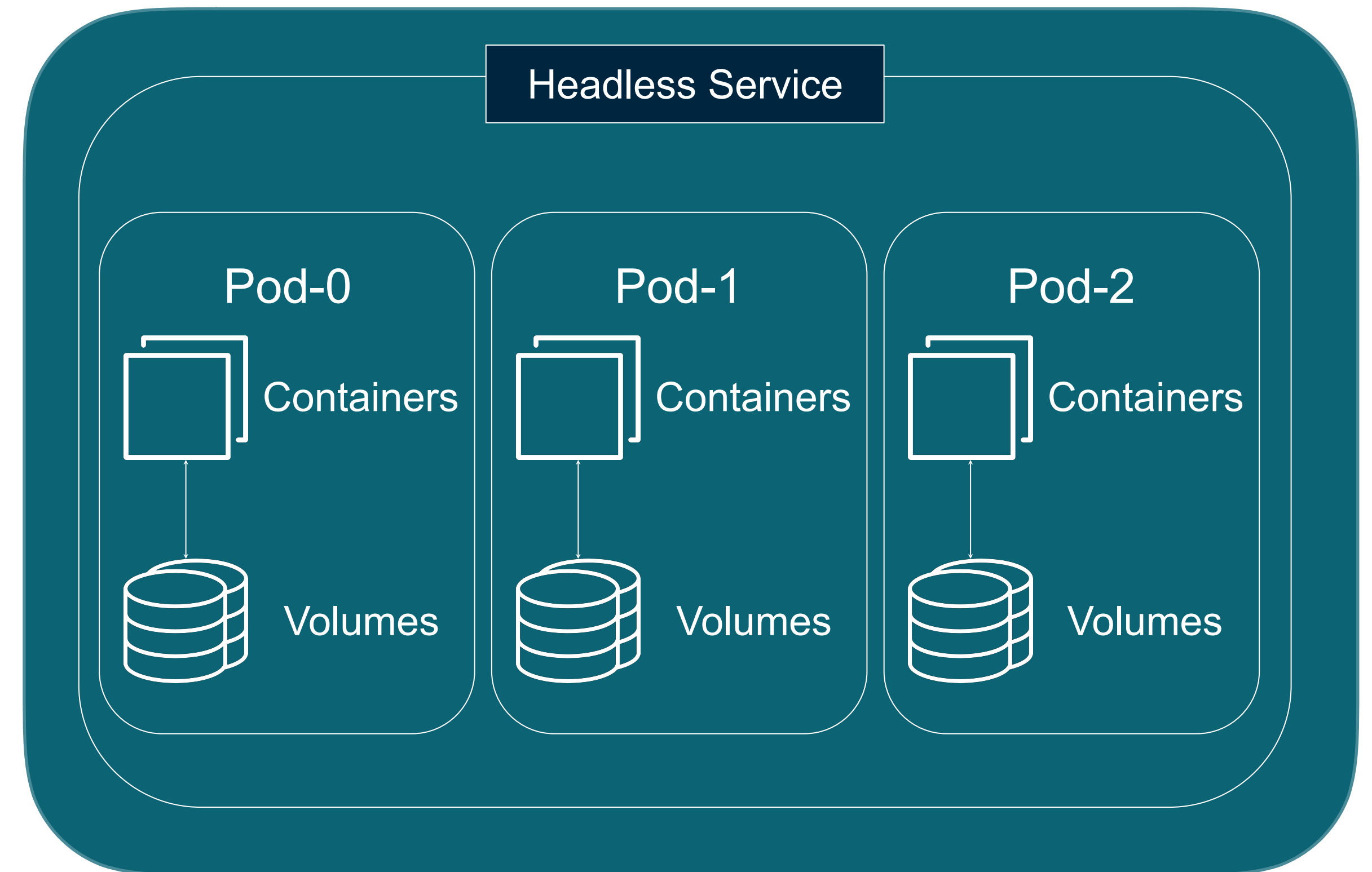
PVCs consume PV



StatefulSet

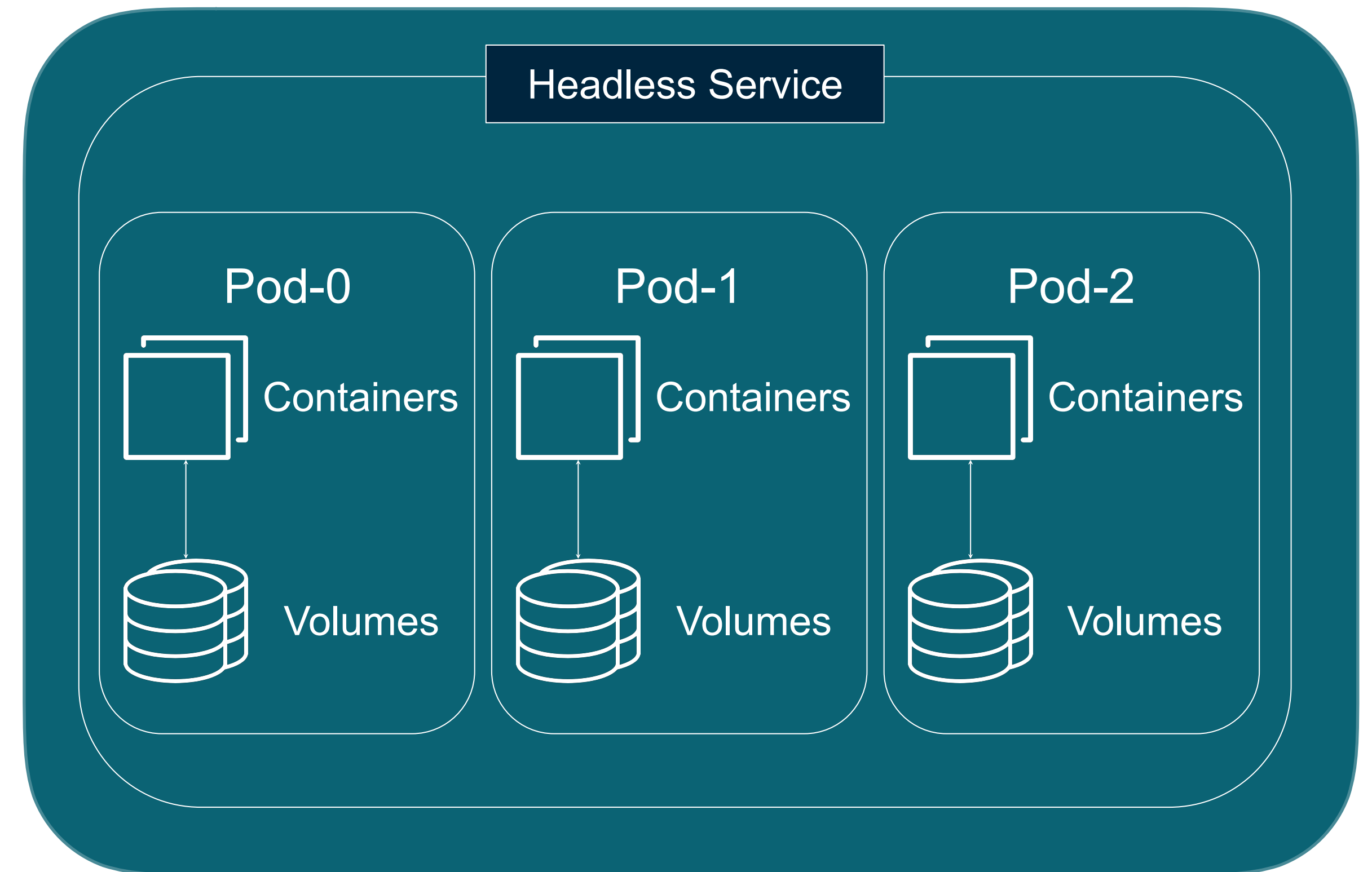
Rely on Headless Service to
provide network identity

Ideal for highly available
stateful workloads



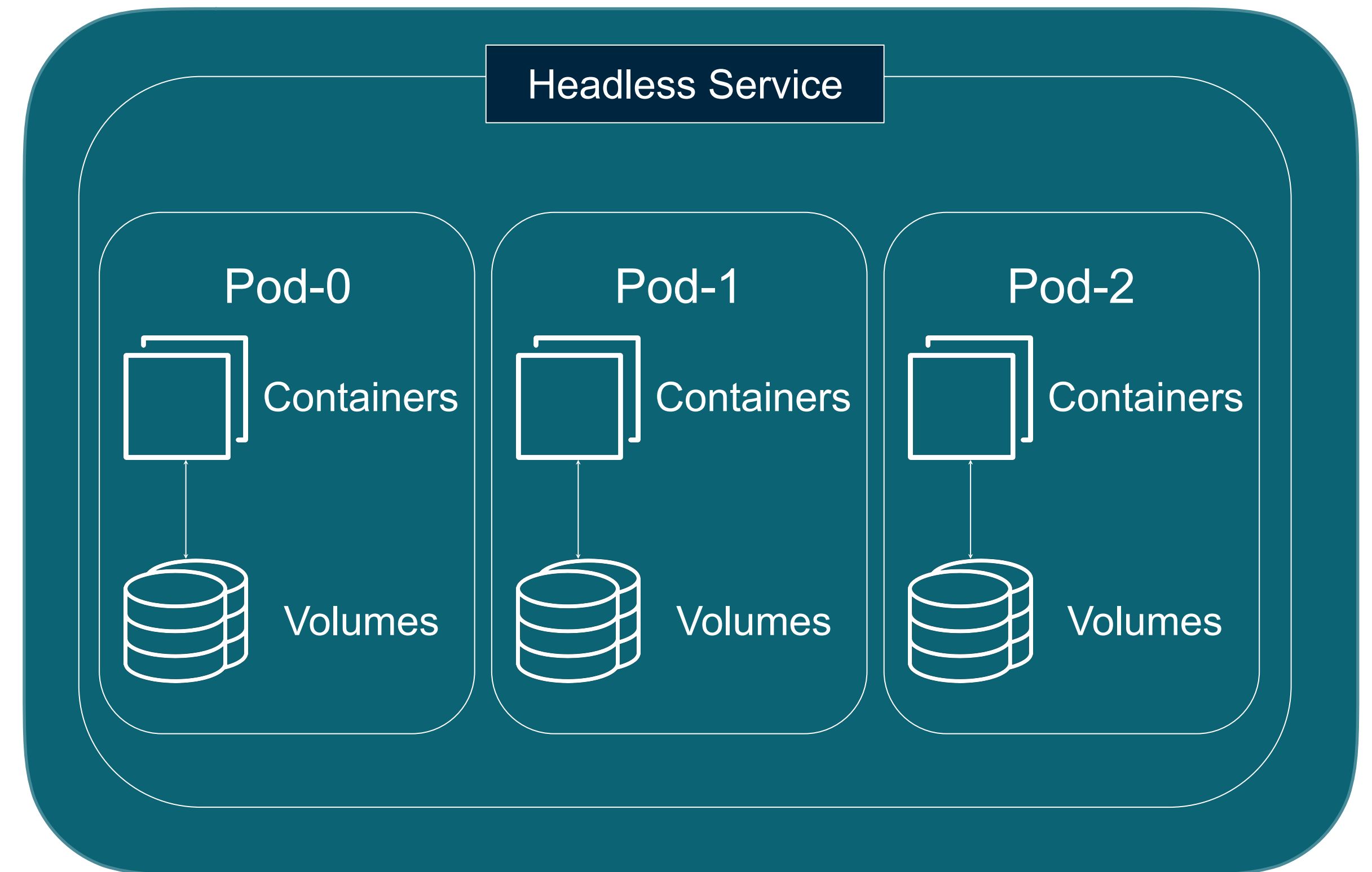
StatefulSet

Rely on Headless
Service to provide
network identity



StatefulSet

Ideal for highly
available stateful
workloads

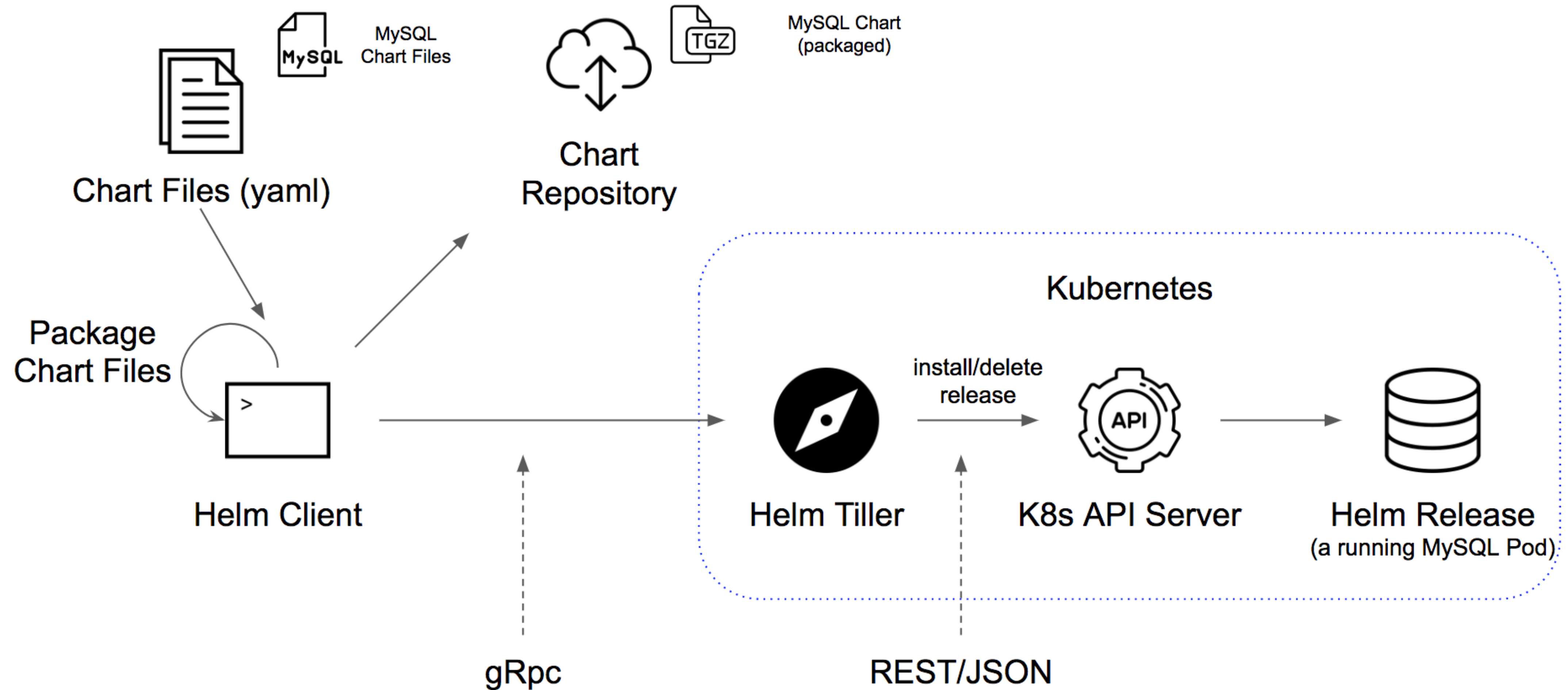




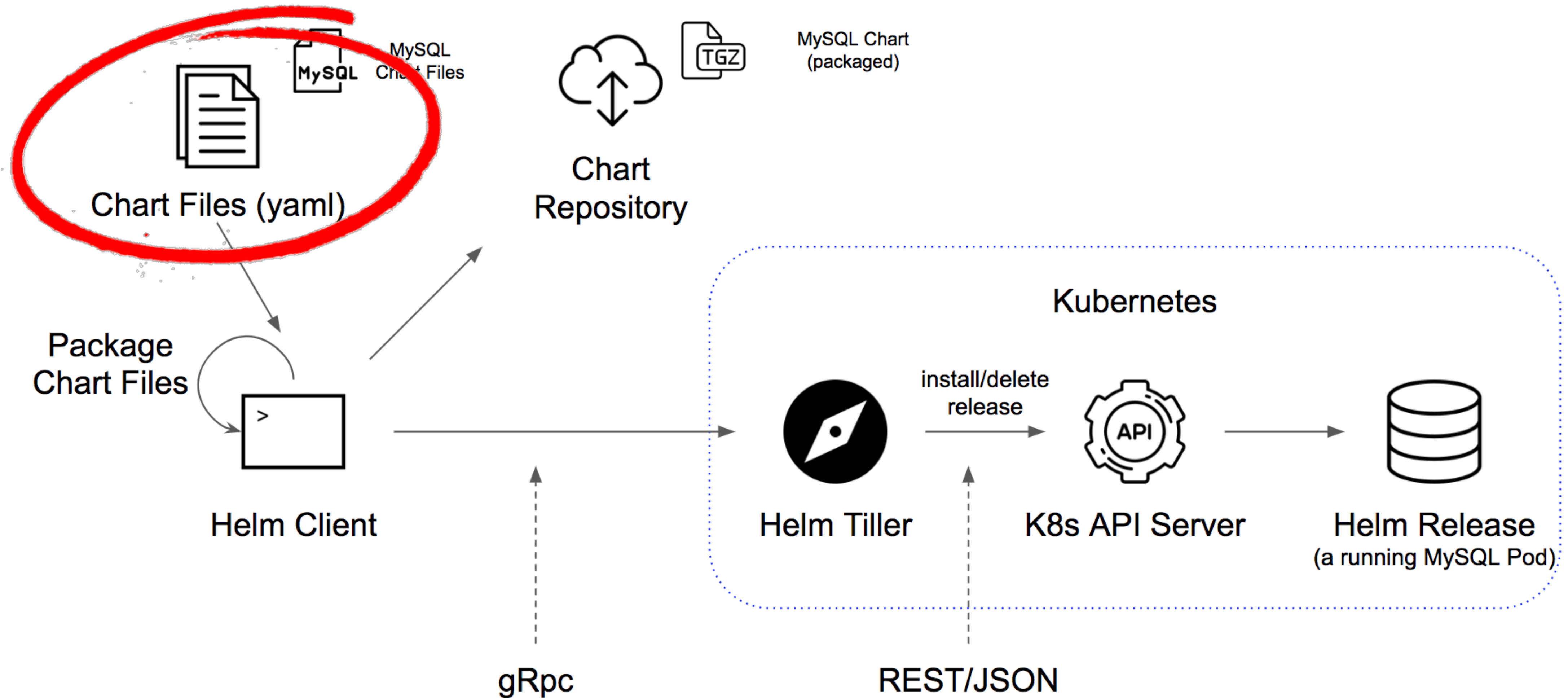
Workloads Deployment

Enter Helm `#ihateyaml`

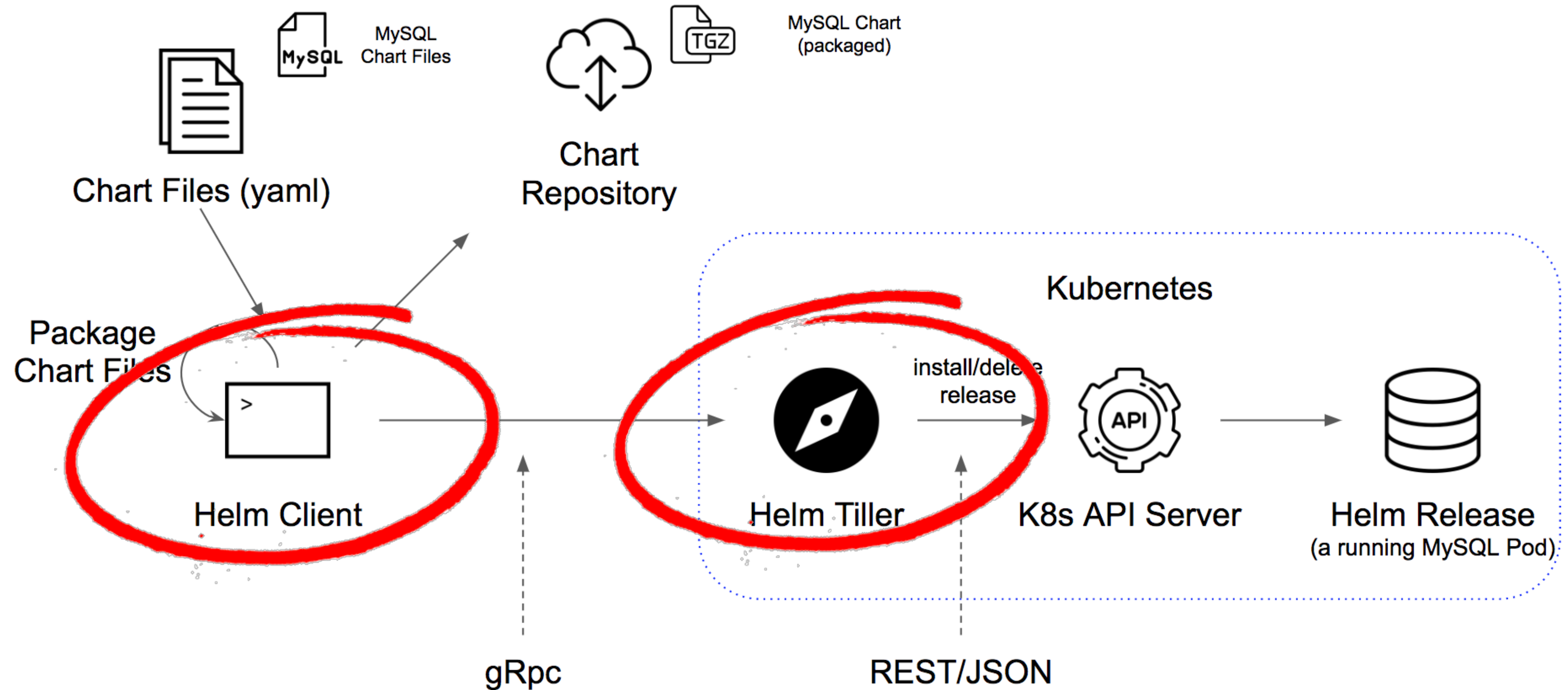
Helm Charts



Helm Charts



Helm Charts

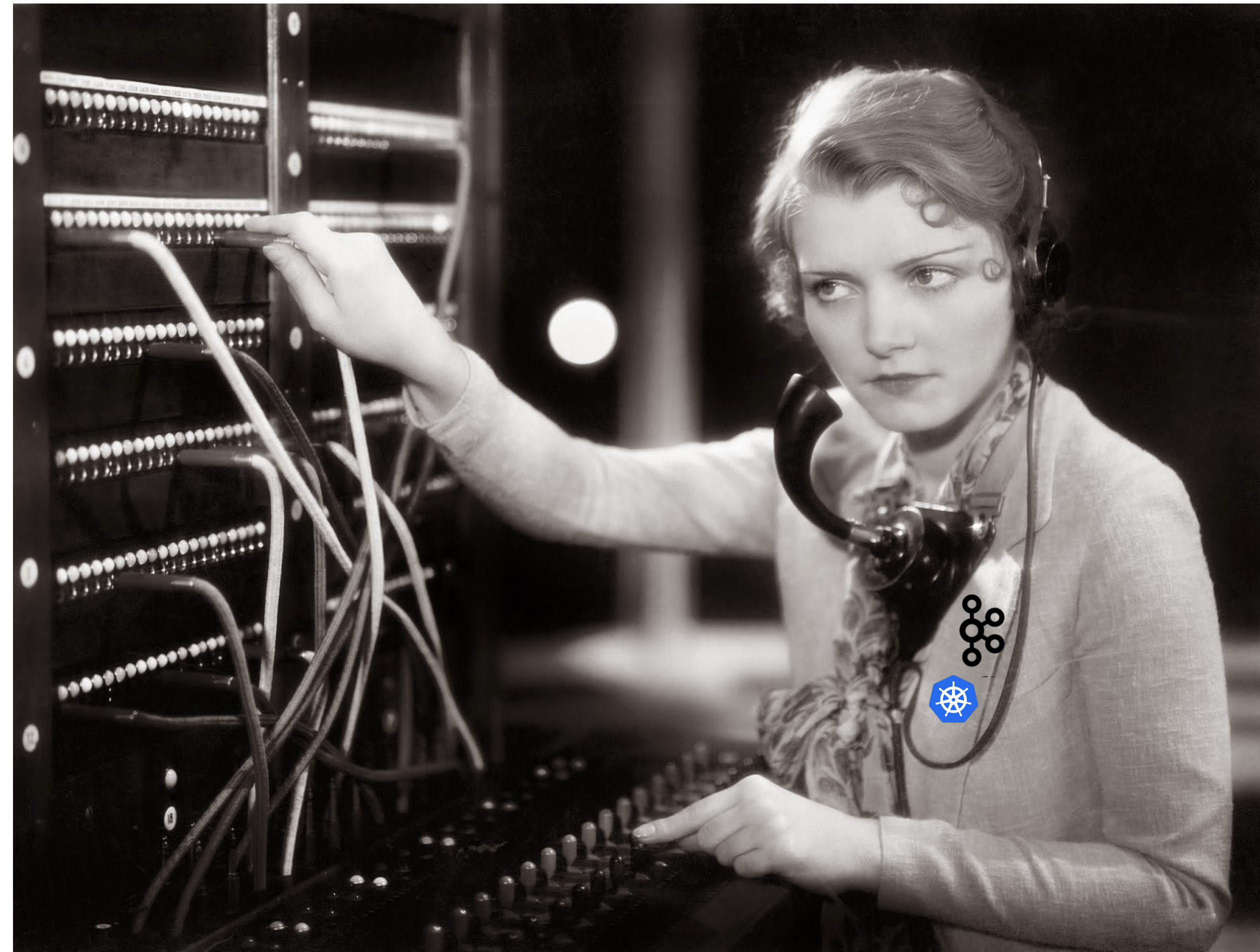


Kubernetes Operator

Custom Resource
Definition (CRDs)

Extending K8s

Controller



Kubernetes Operator

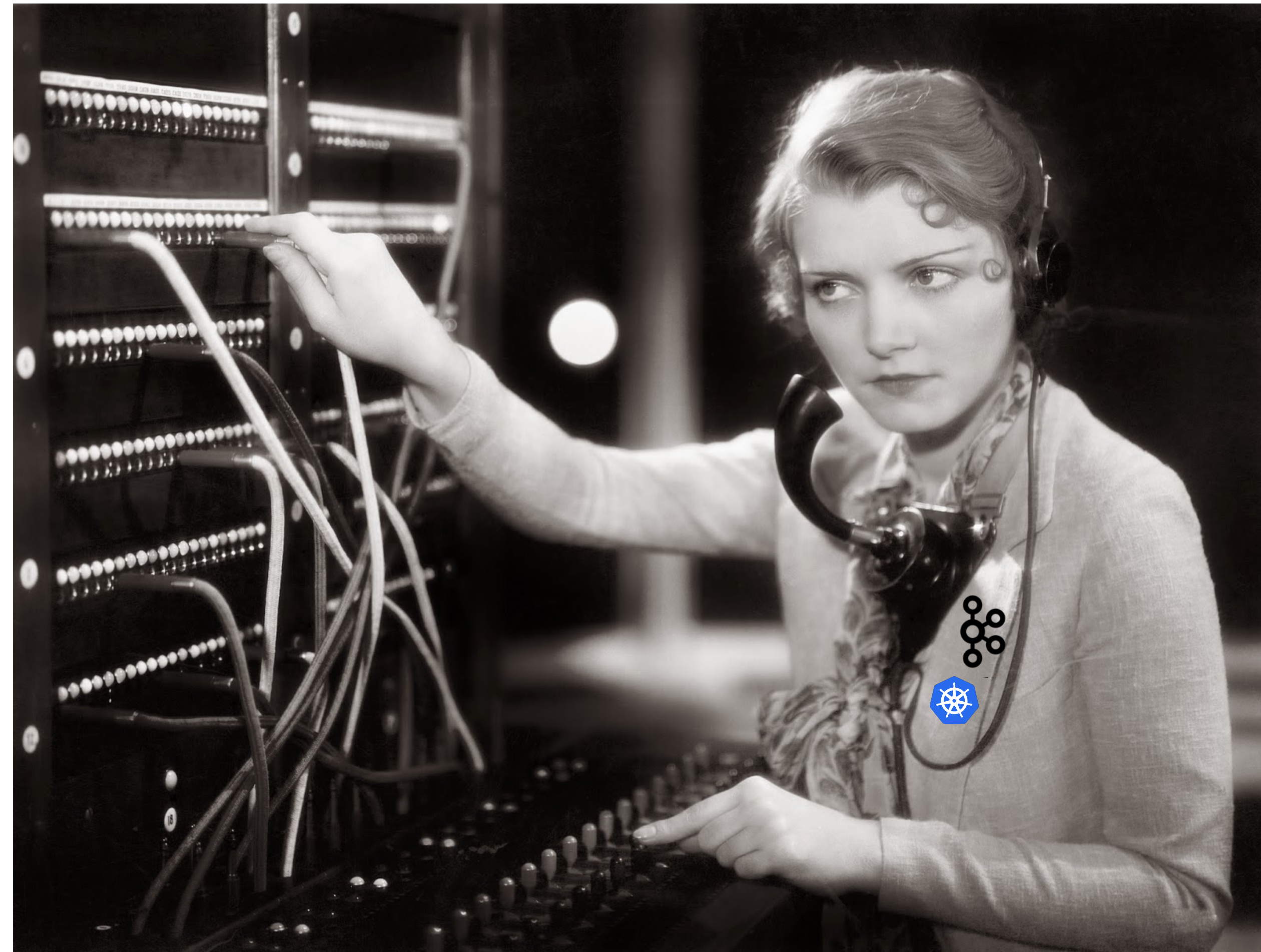
Embedded with operational knowledge
of both data software and
Kubernetes

Backup/restore

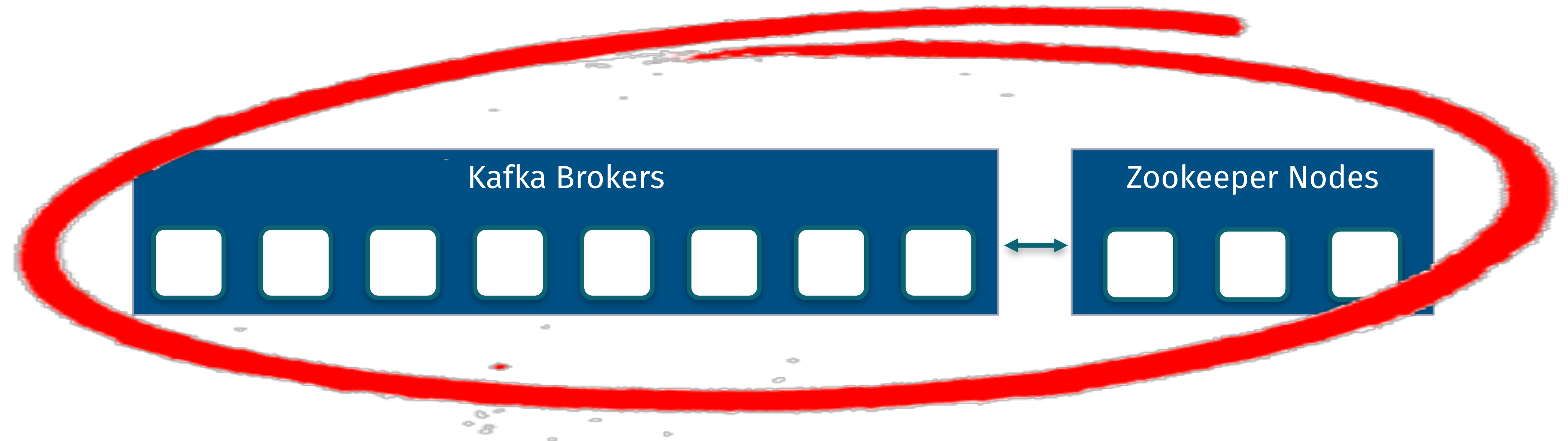
Scale up/down

Rebalance data

Regular health checks



Bare minimum



ZK & Kafka install



PVC for Storage

StatefulSet for 3-node zk

Optional Pod Anti-Affinity to spread the ZK ensemble across nodes.

Headless Service

ConfigMap for Prometheus JMX exporter

Uses ZK Headless Svc

PVC for Storage

StatefulSet for n-node Kafka

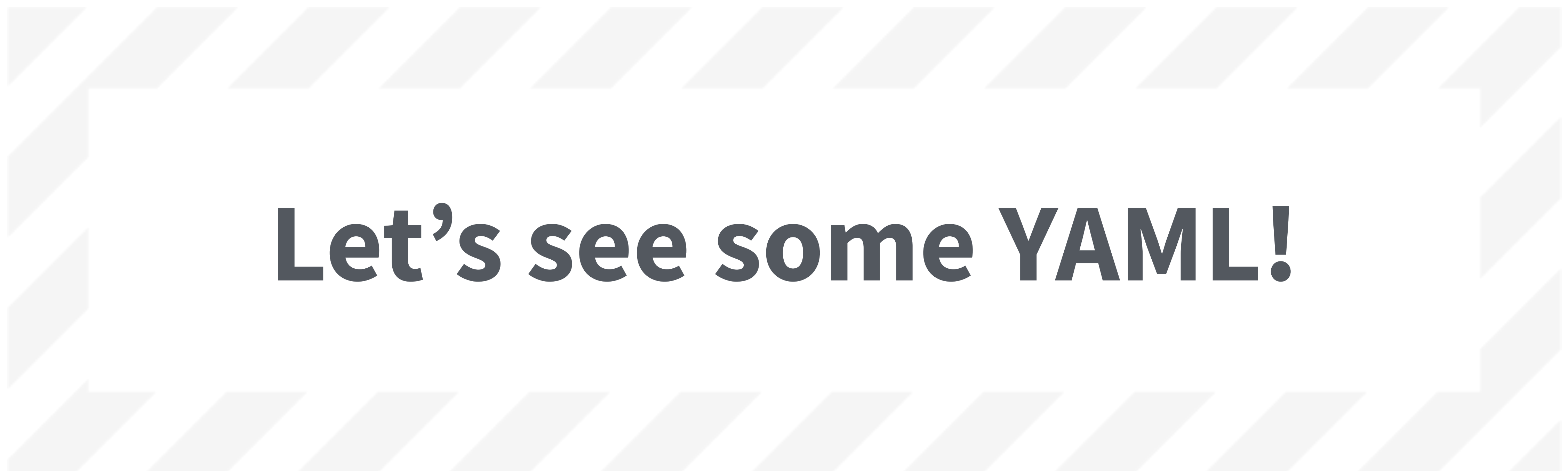
A group of NodePort Services for external traffic

ConfigMap for Prometheus JMX exporter

Kafka Kubernetes journey: Step 1

<https://cnfl.io/cp-helm>





Let's see some YAML!

Step 2: Operator

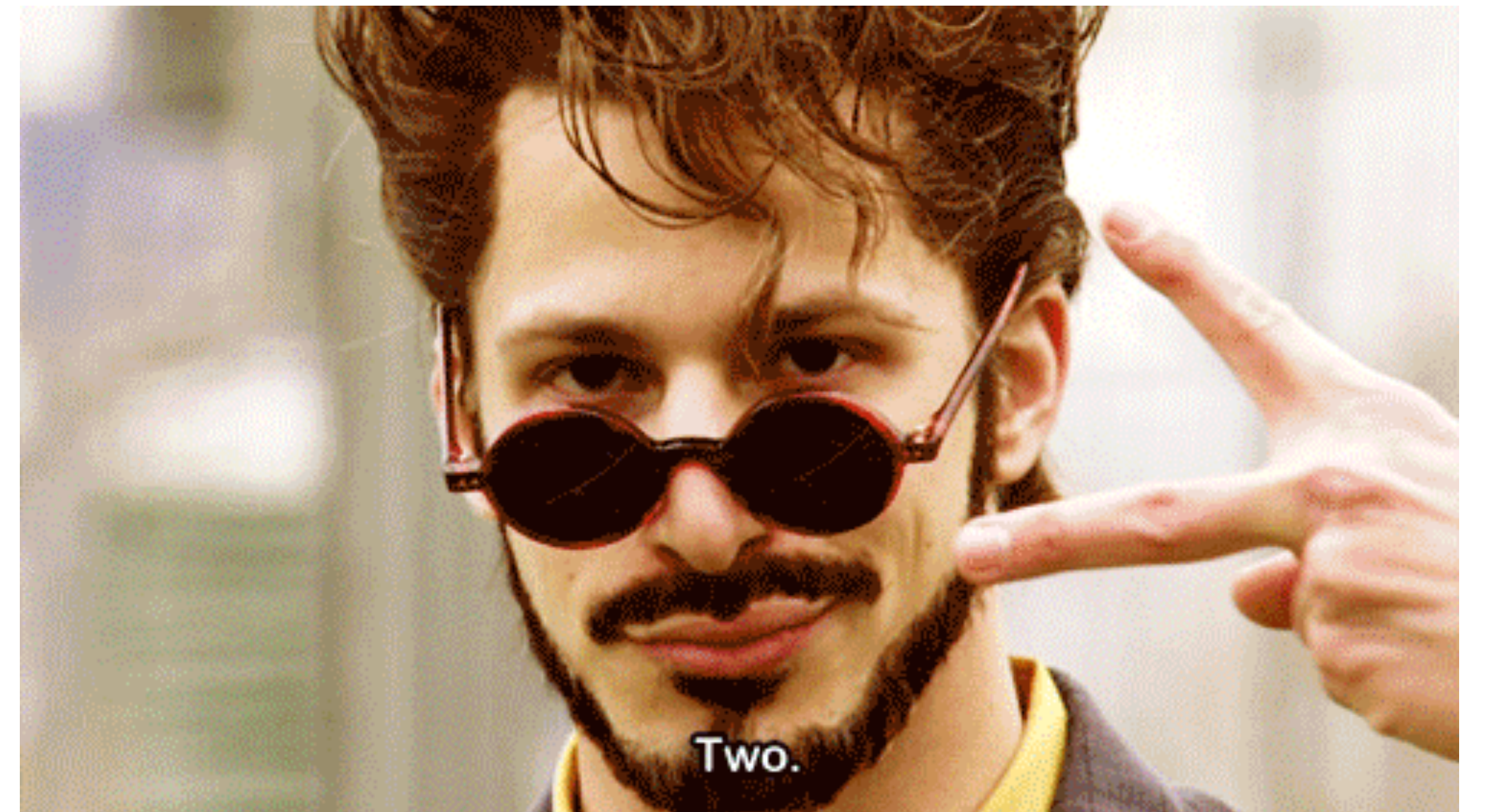
Deploy and Manage your production streaming platform with Confluent Operator.

Automated Provisioning

Platform Operations

Resiliency

Monitoring



Confluent Platform Reference Architecture

Each Confluent Platform component has specific characteristics:

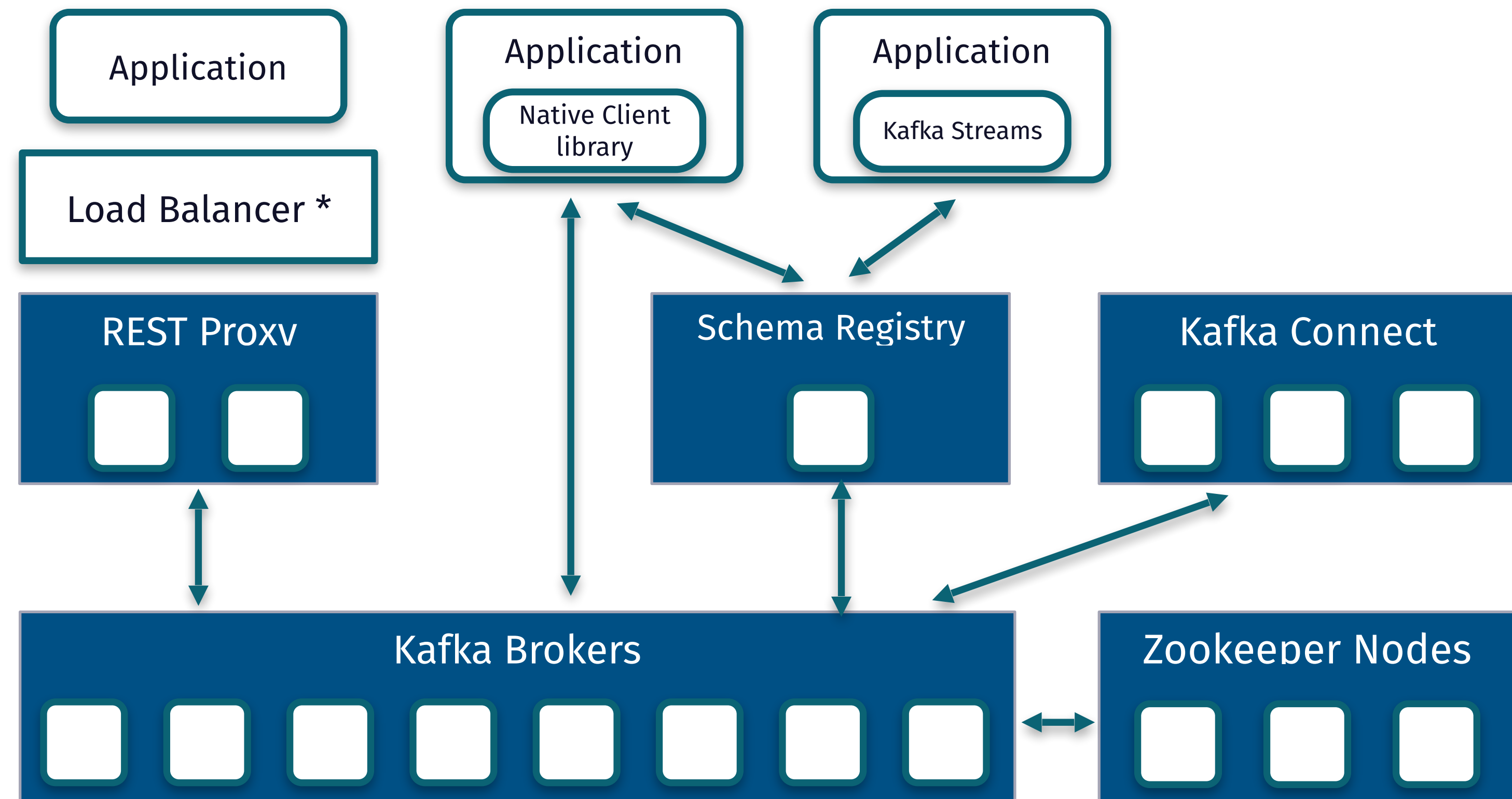
Security (SSL certificates)

DNS names and zones

Host selection

Fault tolerance

Scaling

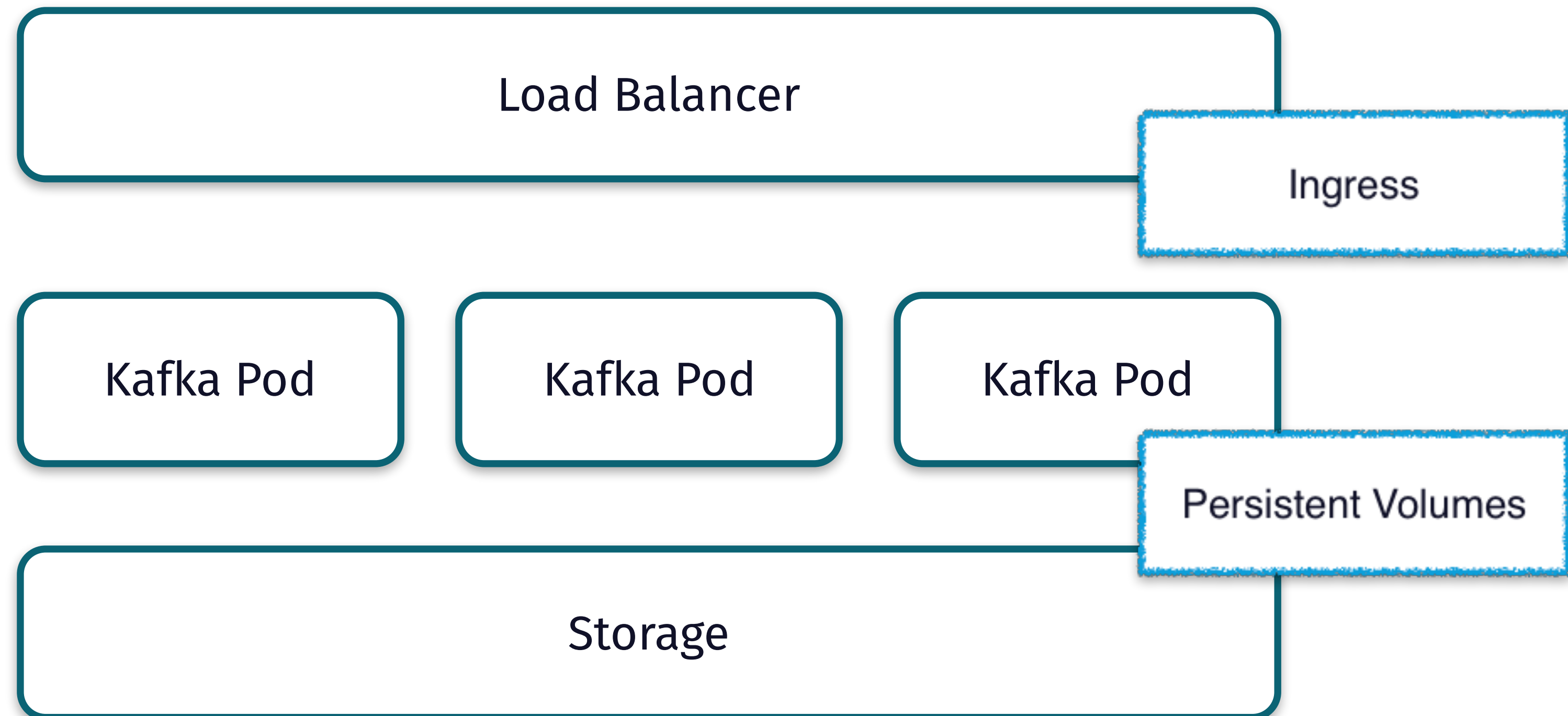


Confluent Operator: Automated Provisioning

```
> cat kafka.yml

## Kafka Cluster
##
kafka:
  name: kafka
  replicas: 3
  resources:
    cpu: 200m
    memory: 1Gi
  external:
    enabled: false
    domain: ""
  tls:
    enabled: true
    domain: devoops.ru
    fullchain: |-
    privkey: |-

> helm install -f kafka.yml --name kafka
```



Confluent Operator: Scale Horizontally

Automate scaling:

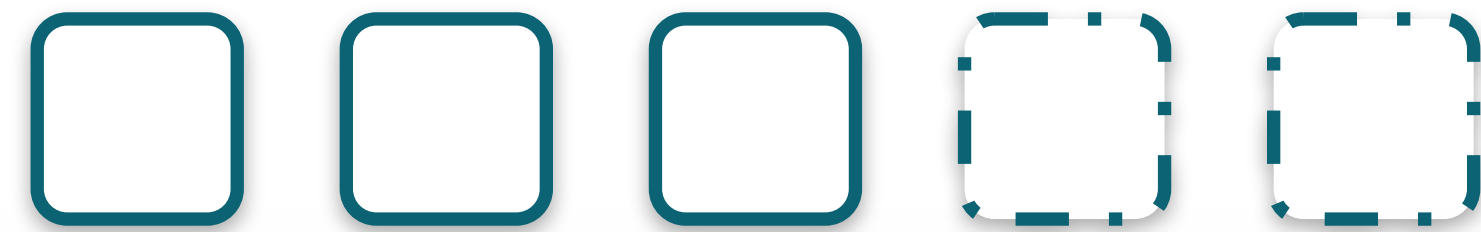
Spin up new broker pod(s)

Distribute partitions to the new broker(s)

Determine balancing plan

Execute balancing plan

Monitor resources



```
> cat kafka_new.yml
```

```
## Kafka Cluster
```

```
##
```

```
kafka:
```

```
  name: kafka
```

```
  replicas: 5
```

```
  version: 5.0.0
```

```
> helm upgrade -f kafka_new.yml --name kafka
```

Confluent Operator: Rolling Upgrade

Automated rolling upgrade with no downtime for Kafka.

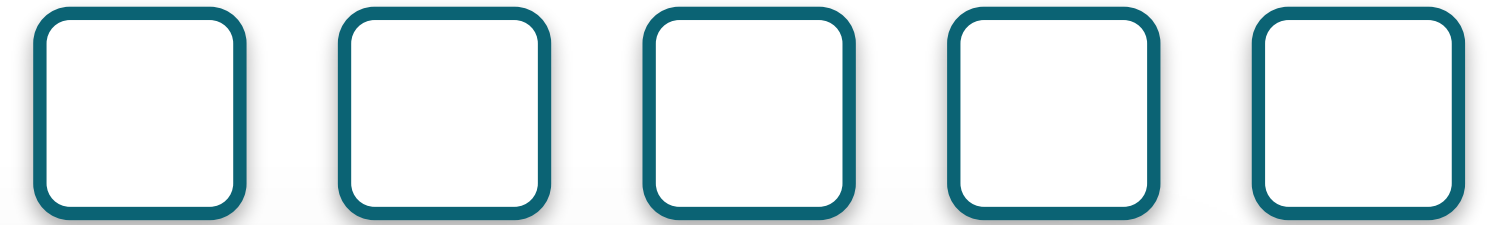
Stop broker

wait for leader election to complete

Start broker with new version

wait for zero under-replicated-partitions

Repeat



```
> cat kafka_new.yml
```

```
## Kafka Cluster  
##
```

```
kafka:  
  name: kafka  
  replicas: 5  
  version: 5.1.0
```

```
> helm upgrade -f kafka_new.yml --name kafka
```




Will it fly? Let's see

Confluent Operator



Automate provisioning



Scale your Kafkas and CP clusters elastically



Monitor SLAs through Confluent Control Center or Prometheus

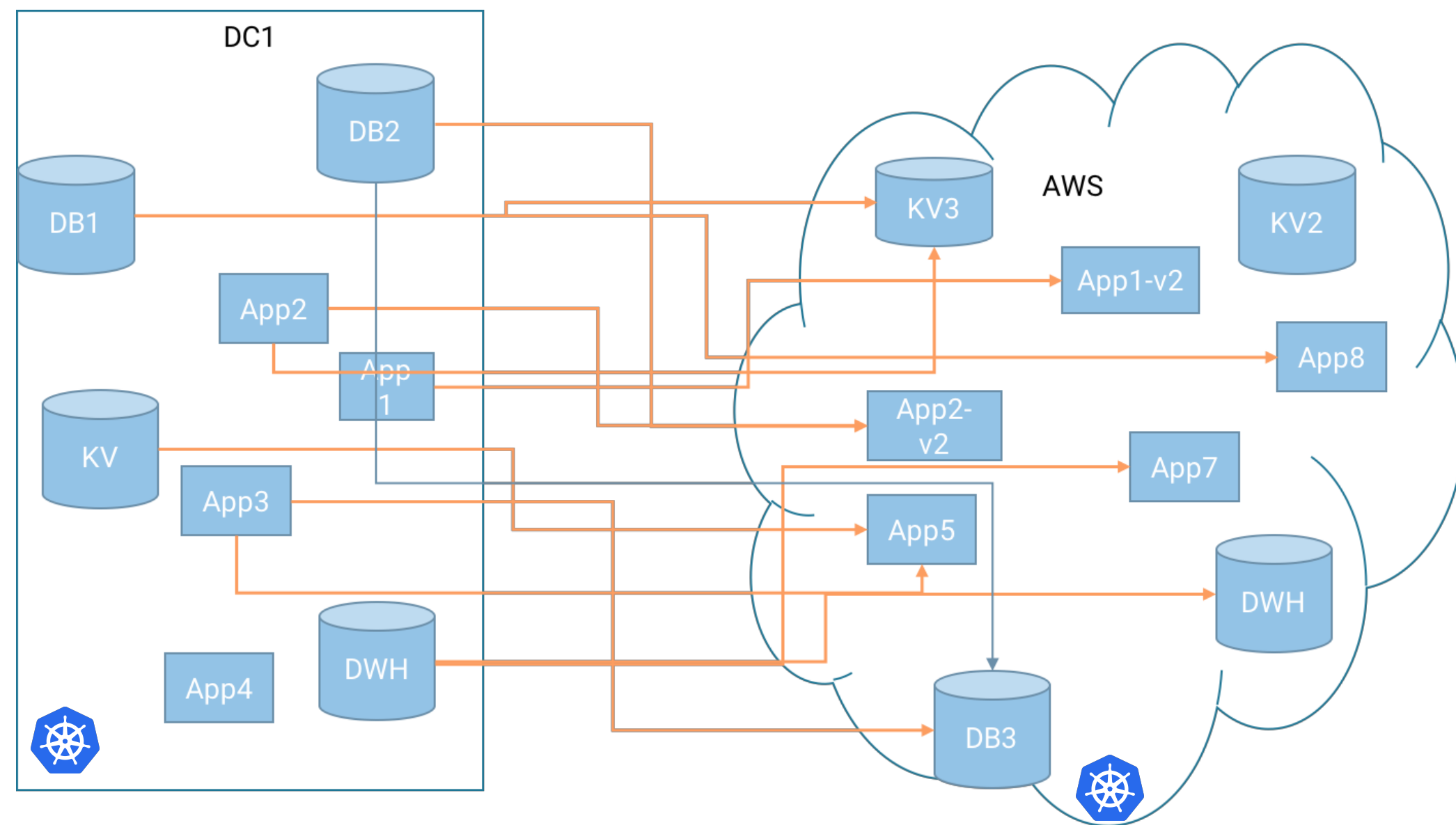


Operate at scale with enterprise support from Confluent

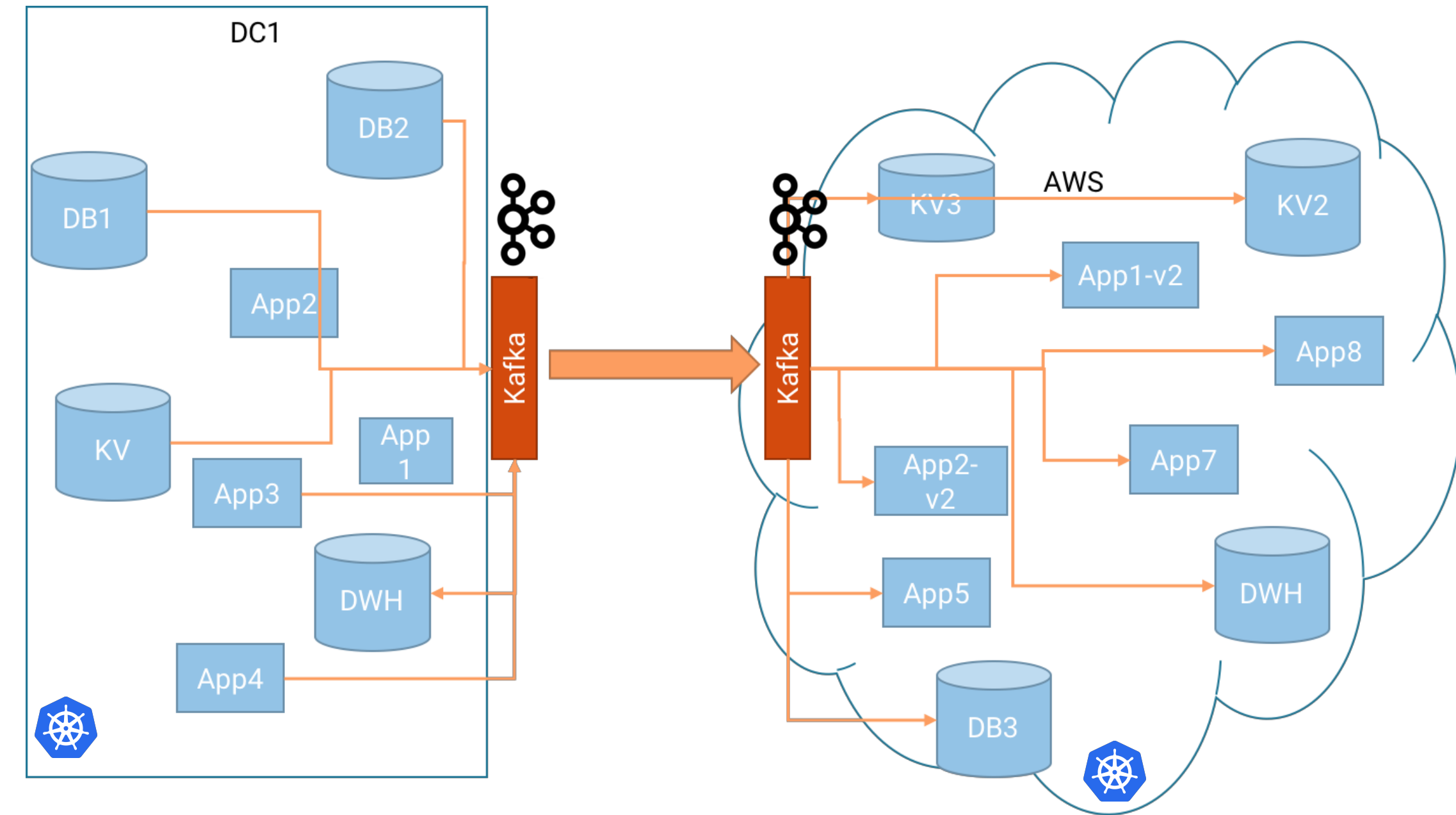


Future

vision of the future



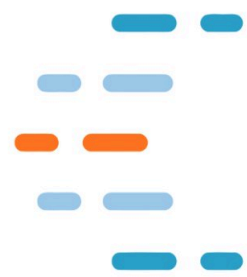
VS.



Resources and Next Steps



<https://cnfl.io/cp-helm>



<https://cnfl.io/k8s>



<https://slackpass.io/confluentcommunity>
#kubernetes

THANKS!



@gamussa
viktor@confluent.io

we are hiring!
<https://www.confluent.io/careers/>