Kubernetes Services

- The theory of Services
- Create a Service the iterative way
- Create a Service the declarative way
- Real-world application
- Recap

Kubernetes Services

The Theory

```
$ kubectl get pods
NAME             READY     STATUS    RESTARTS   AGE
hello-rc-1qch1   1/1       Running   0          8d
hello-rc-39q0s   1/1       Running   0          8d
hello-rc-3fr7t   1/1       Running   0          8d
hello-rc-5qzpl   1/1       Running   0          8d
…
```

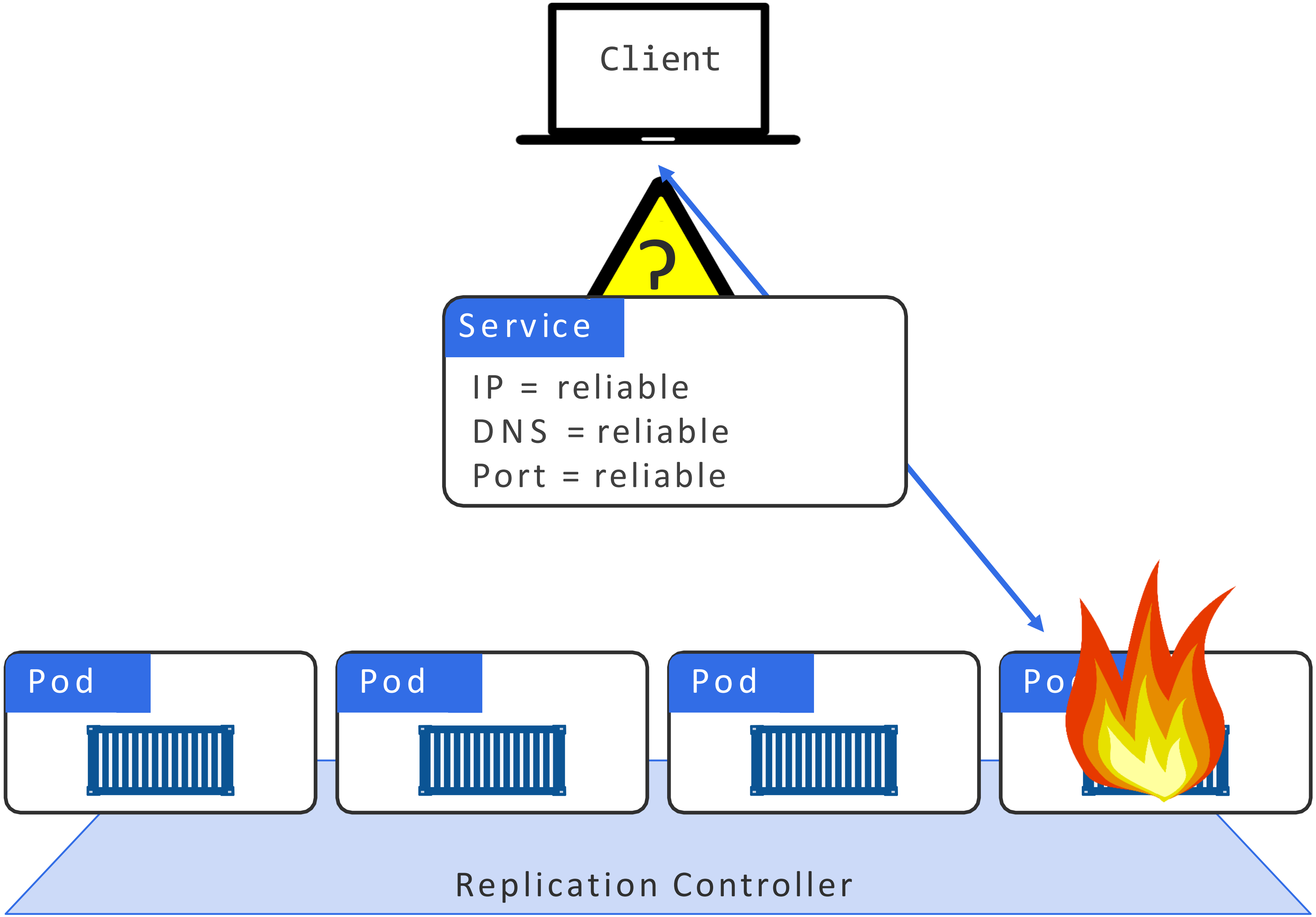
Q.

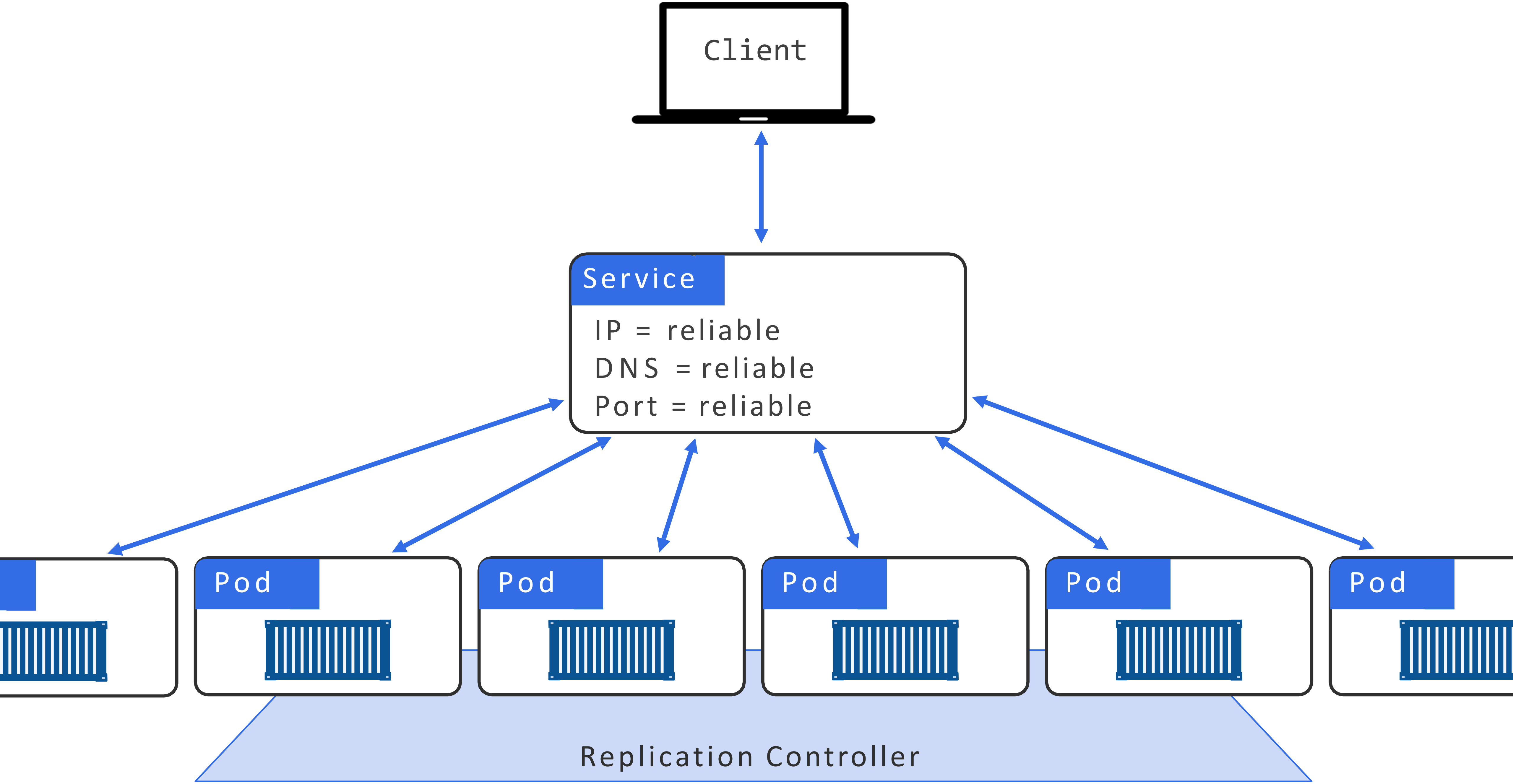
How do we access our app?

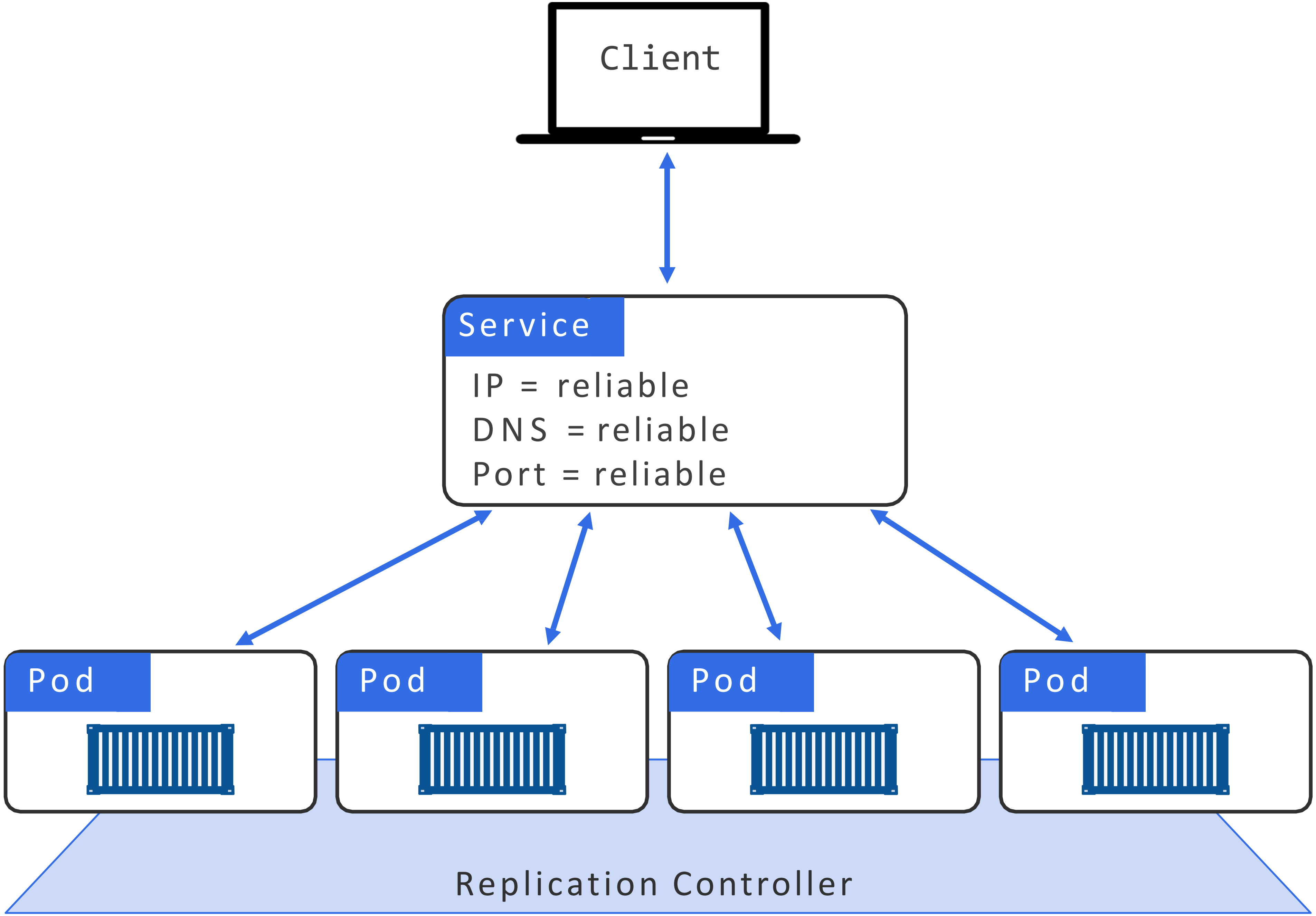
- From outside the cluster
- From inside the cluster

Services:

- REST objects in the K8s API
- Abstractions


Client
?
Service
IP = reliable
DNS = reliable
Port = reliable
Pod
Pod
Pod
Po
Replication Controller


Client
Service
IP = reliable
DNS = reliable
Port = reliable
Pod
Pod
Pod
Pod
Pod
Replication Controller



Client

Service

IP = reliable

DNS = reliable

Port = reliable

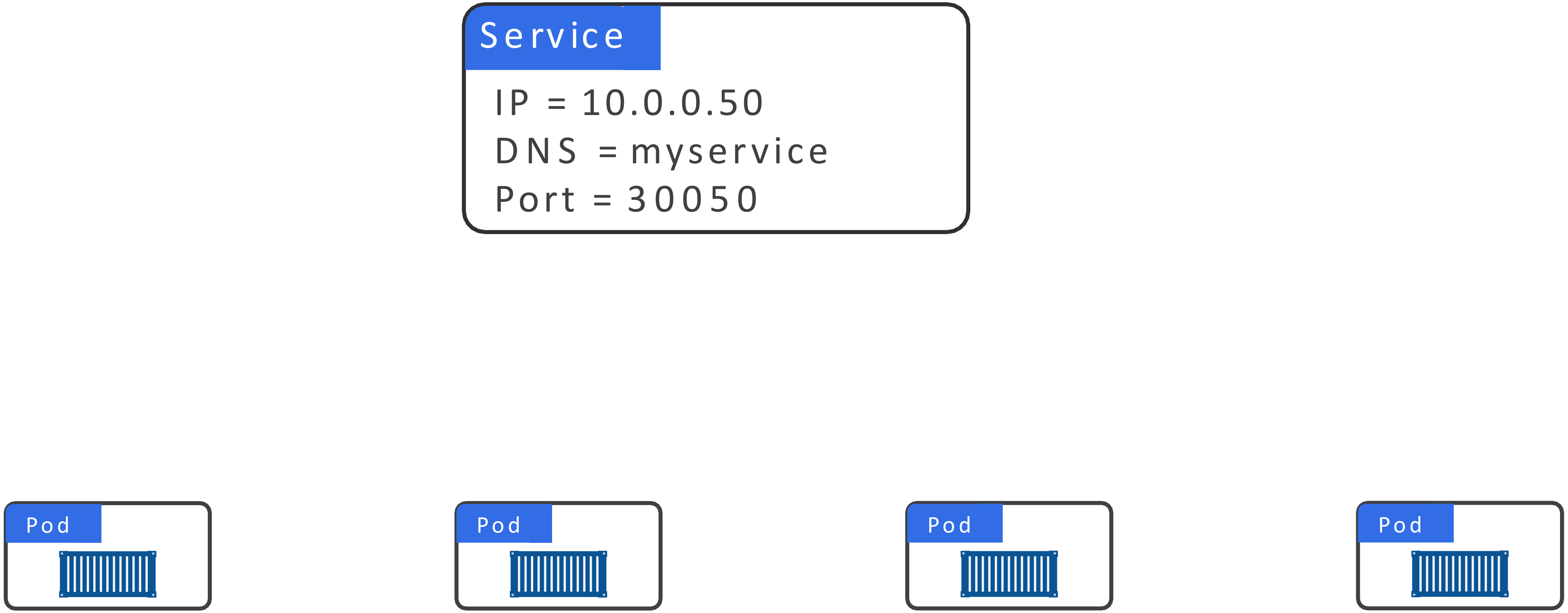
Pod

Pod

Pod

Pod

Replication Controller



Service

IP = 10.0.0.50

DNS = myservice

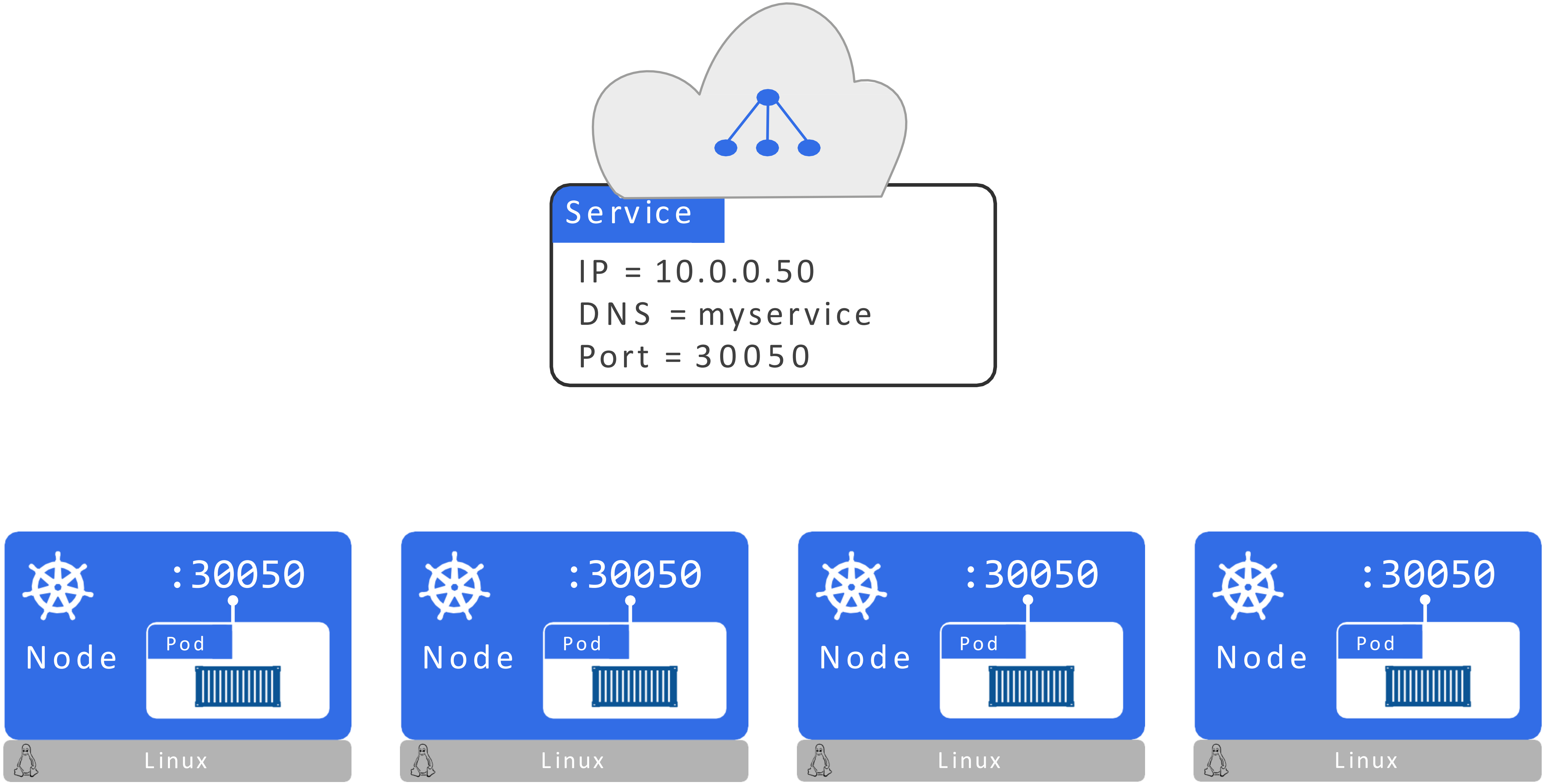
Port = 30050

Pod

Pod

Pod

Pod



Service

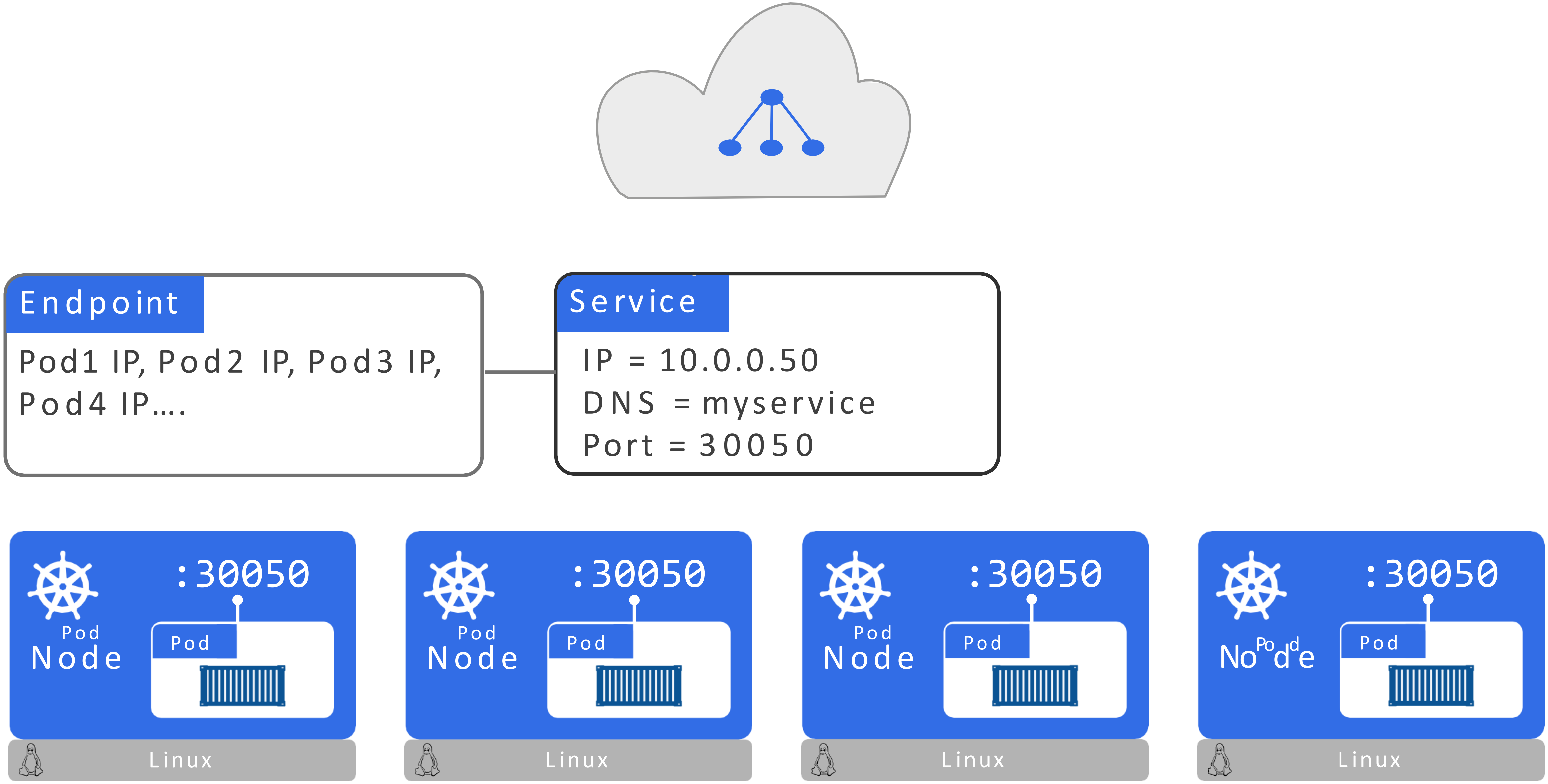
IP = 10.0.0.50
DNS = myservice
Port = 30050

Node :30050 Pod Linux

Node :30050 Pod Linux

Node :30050 Pod Linux

Node :30050 Pod Linux



Endpoint

Pod1 IP, Pod2 IP, Pod3 IP, Pod4 IP….

Service

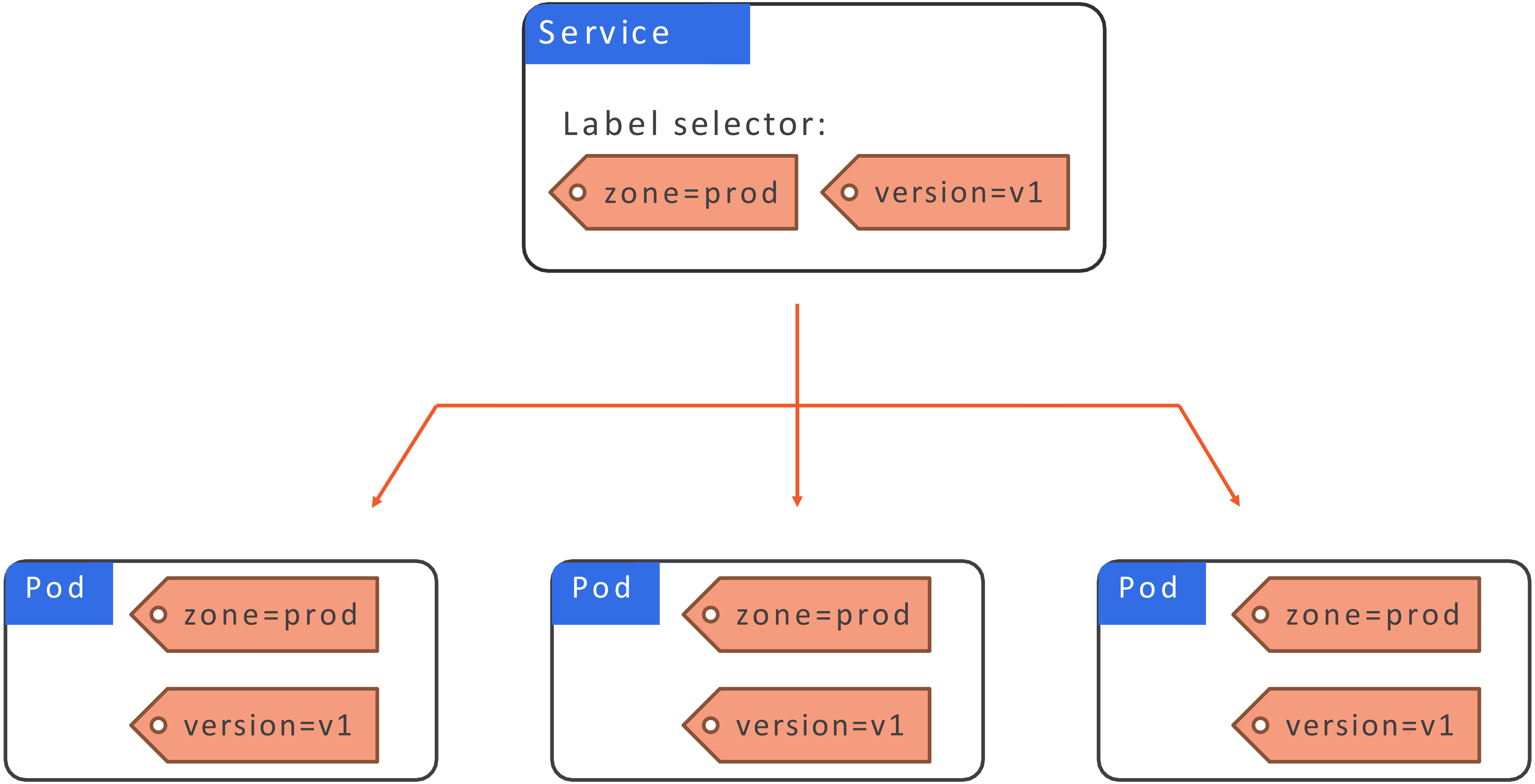
IP = 10.0.0.50
DNS = myservice
Port = 30050

:30050
Pod
Node
Pod
Linux

:30050
Pod
Node
Pod
Linux

:30050
Pod
Node
Pod
Linux

:30050
Pod
Node
Pod
Linux


Service
Label selector:
zone=prod
version=v1
Pod
zone=prod
version=v1
Pod
zone=prod
version=v1
Pod
zone=prod
version=v1

Service Discovery

- DNS based (best)
- Environment variables

ServiceType:

ClusterIP: Stable internal cluster IP

NodePort: Exposes the app outside of the cluster by adding a cluster-wide port on top of ClusterIP

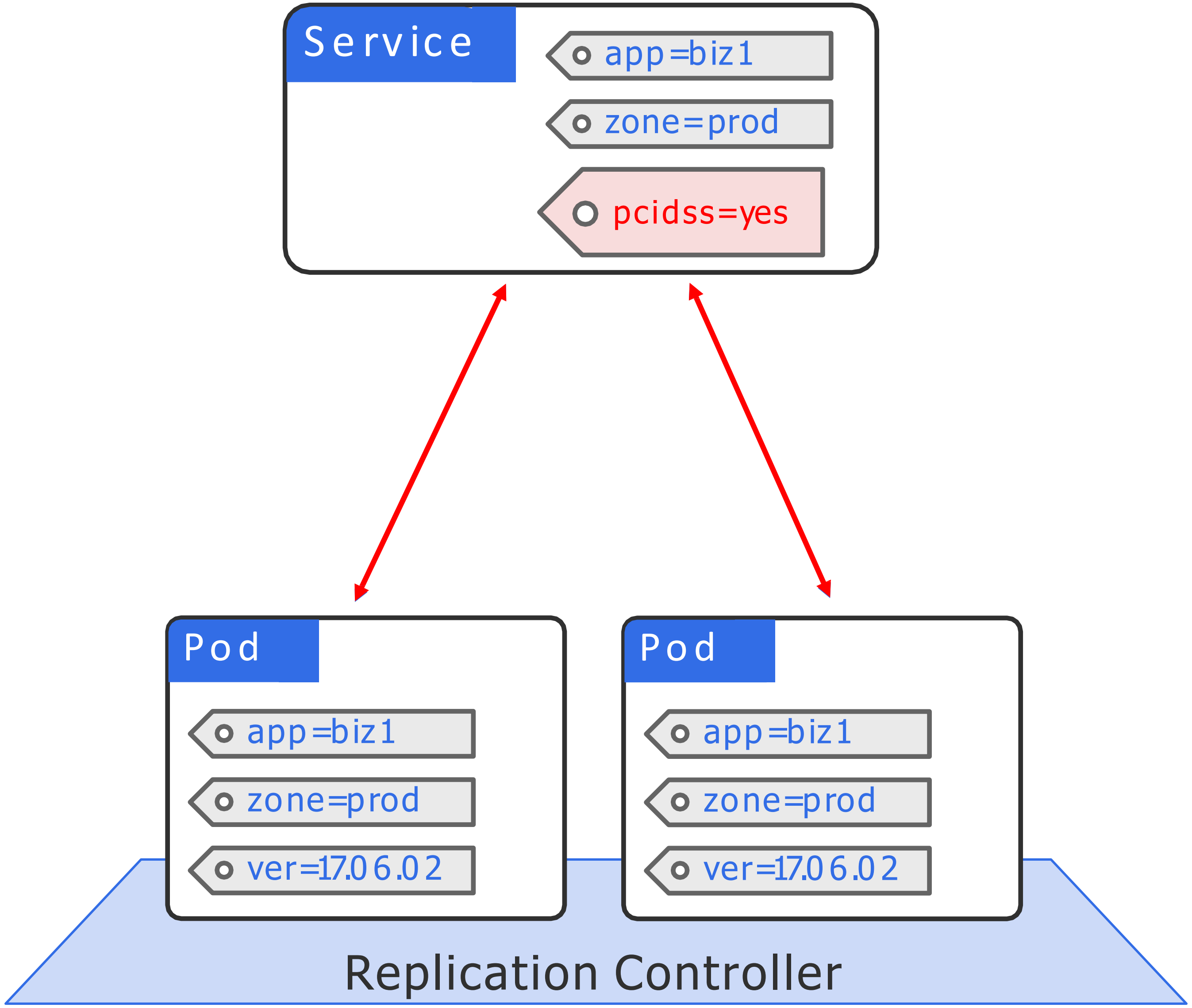
LoadBalancer: Integrates NodePort with cloud-based load balancers

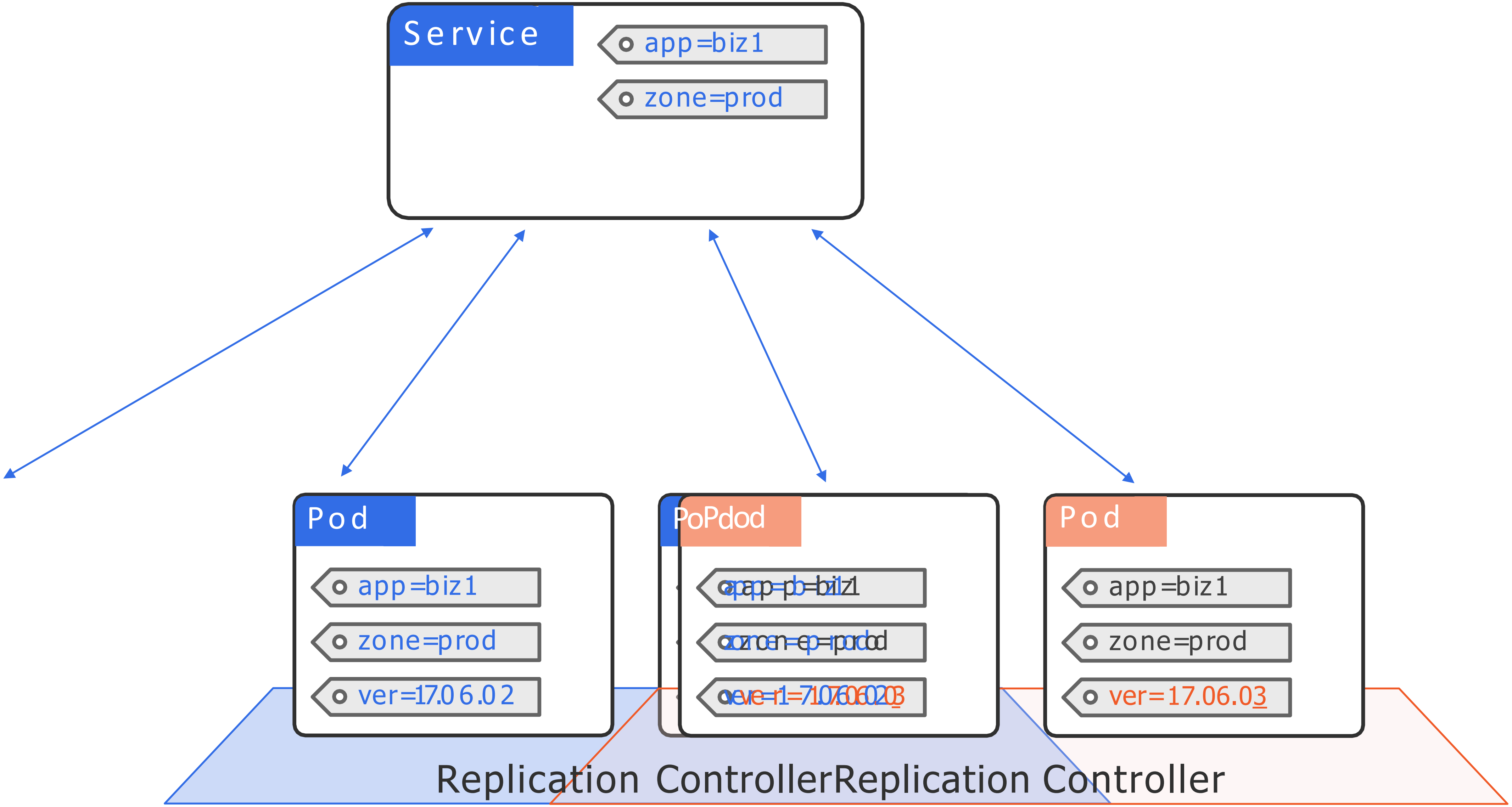
Kubernetes Services

In the real world

Kubernetes Services

UpdatInintgheBreuaslinweosrlsdApps


Service
app=biz1
zone=prod
pcidss=yes
Pod
app=biz1
zone=prod
ver=17.06.02
Pod
app=biz1
zone=prod
ver=17.06.02
Replication Controller



Service

app=biz1

zone=prod

Pod

app=biz1

zone=prod

ver=17.06.02

Pod

app=biz1

zone=prod

ver=17.06.02

Pod

app=biz1

zone=prod

ver=17.06.03

Pod

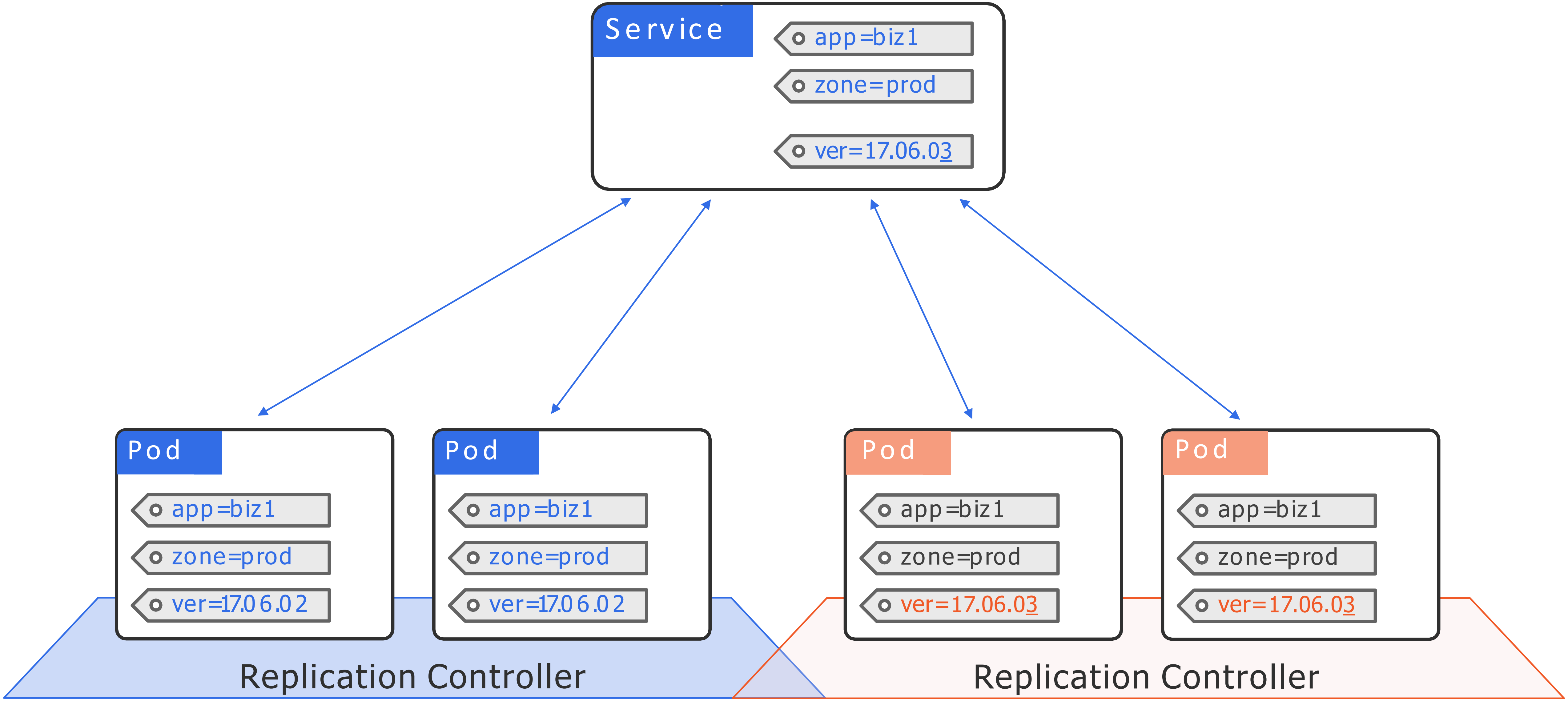
app=biz1

zone=prod

ver=17.06.03

Replication Controller

Replication Controller



Service

app=biz1

zone=prod

ver=17.06.03

Pod

app=biz1

zone=prod

ver=17.06.02

Pod

app=biz1

zone=prod

ver=17.06.02

Pod

app=biz1

zone=prod

ver=17.06.03

Pod

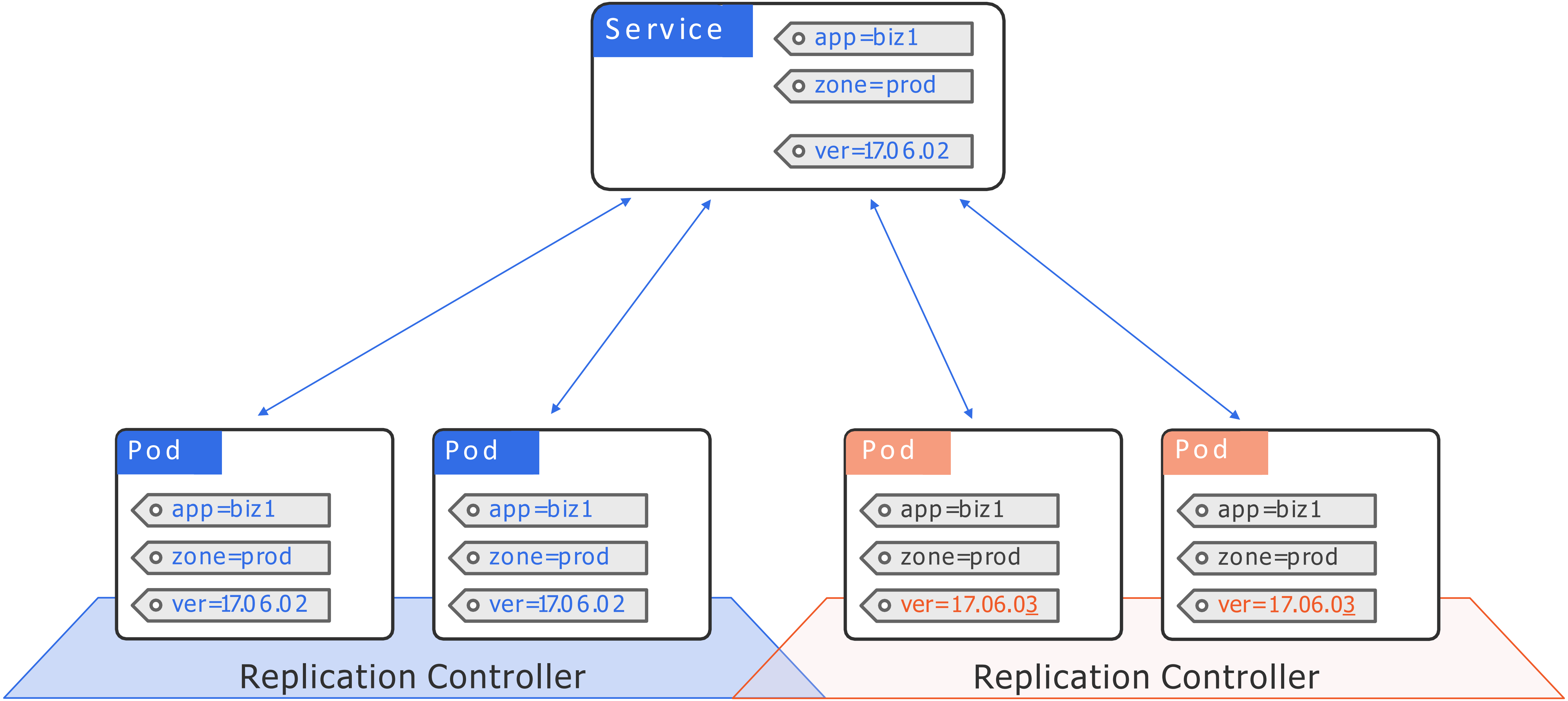
app=biz1

zone=prod

ver=17.06.03

Replication Controller

Replication Controller


Service
app=biz1
zone=prod
ver=17.06.02
Pod
app=biz1
zone=prod
ver=17.06.02
Pod
app=biz1
zone=prod
ver=17.06.02
Pod
app=biz1
zone=prod
ver=17.06.03
Pod
app=biz1
zone=prod
ver=17.06.03
Replication Controller
Replication Controller

Kubernetes Services Summary

Reliable network endpoint

IP address

DNS name

Port

Expose Pods to the outside world

NodePort
Provides a cluster-wide port

LoadBalancer
Integrates with cloud-based load balancers

Coming up…

Kubernetes Deployments