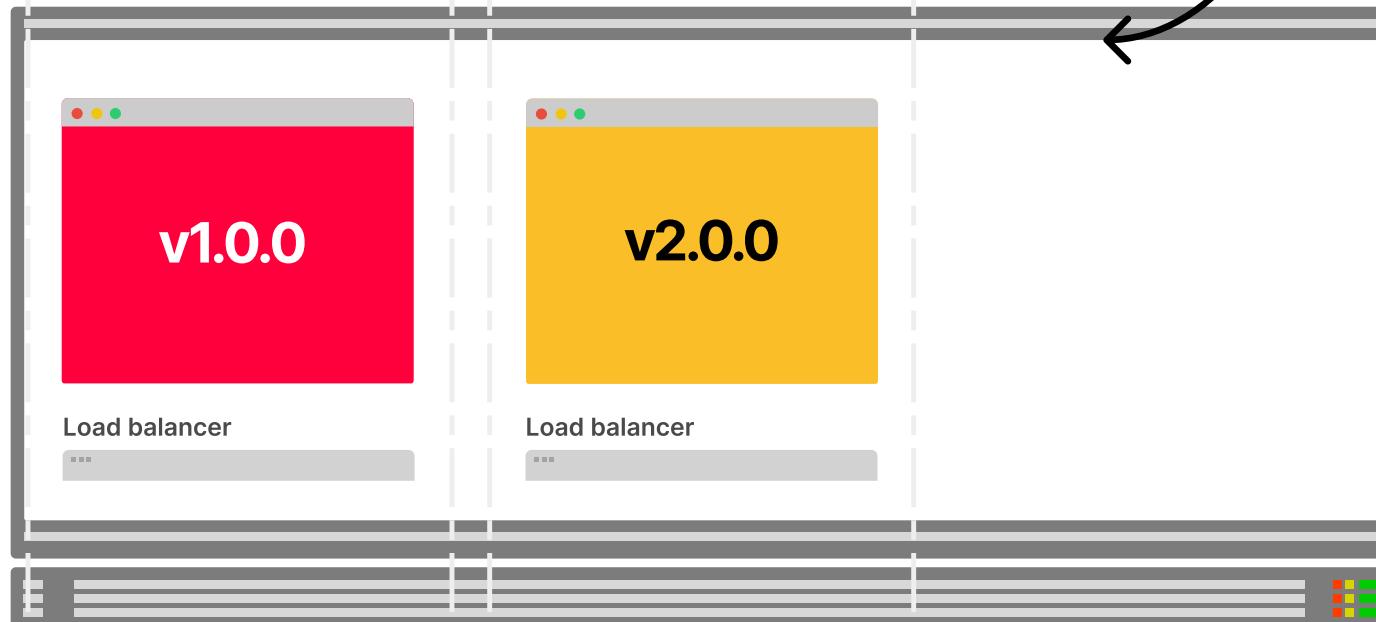


KUBERNETES MULTI-TENANCY

Explore soft vs hard multi-tenancy in Kubernetes (and the other options in between).

namespace namespace

single cluster



VS

multiple clusters



When planning your infrastructure, one of the fundamental questions is: *how many Kubernetes clusters should you have?*

One big cluster or multiple smaller clusters?

Should the team share resources, or to each their own?

In this thread, you will explore **three different multi-tenant options with stronger isolation and their trade-offs.**

In particular, we will look at three concrete implementations: the **Hierarchical Namespace controller**, **vCluster** and **Karmada**.

MULTI-TENANCY OPTIONS

1

Soft multi-tenancy

Hierarchical Namespace controller, Capsule

2

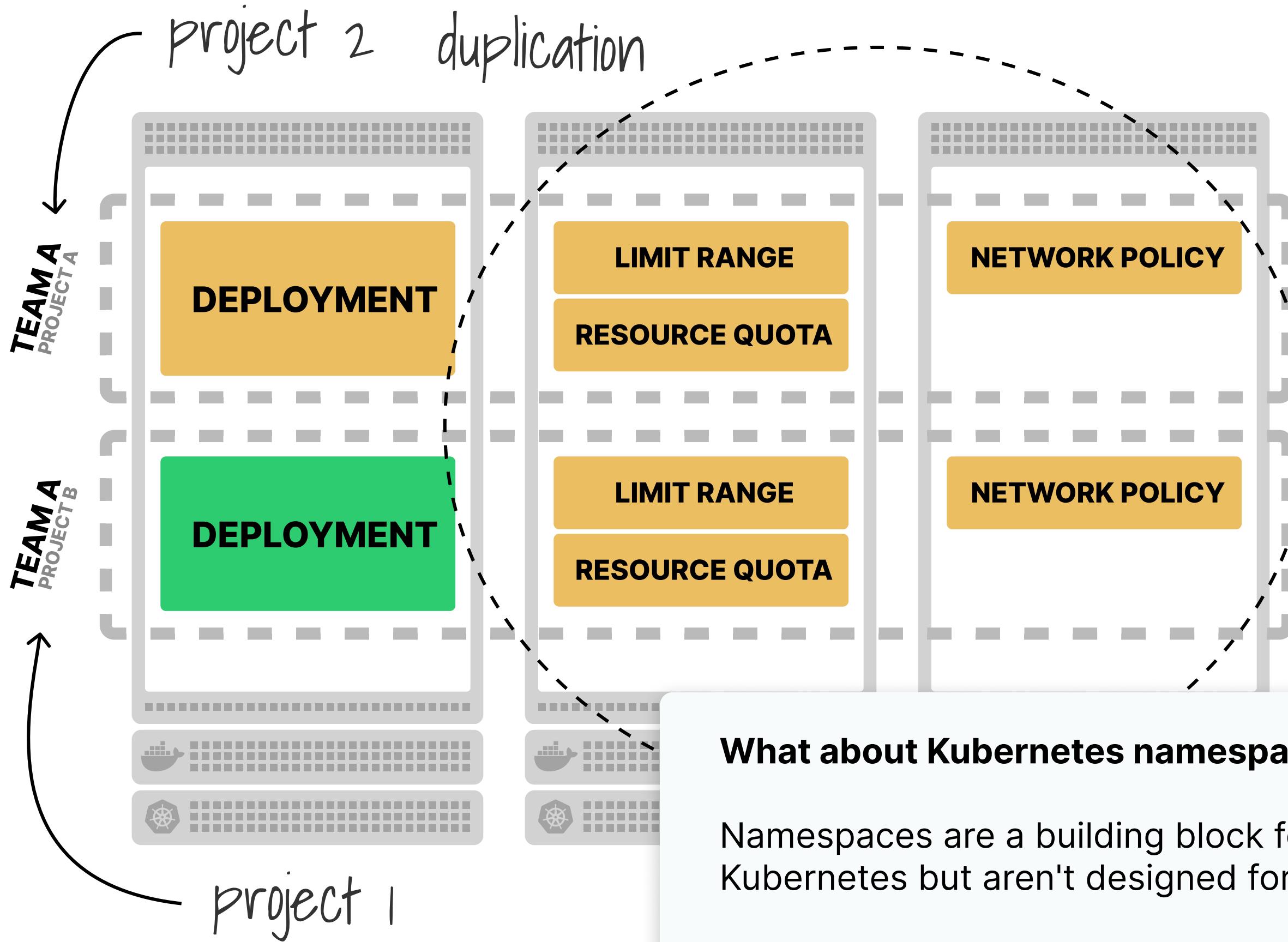
Control plane isolation

vCluster, Kamaji*, Hypershift*

3

Hard multi-tenancy

Karmada

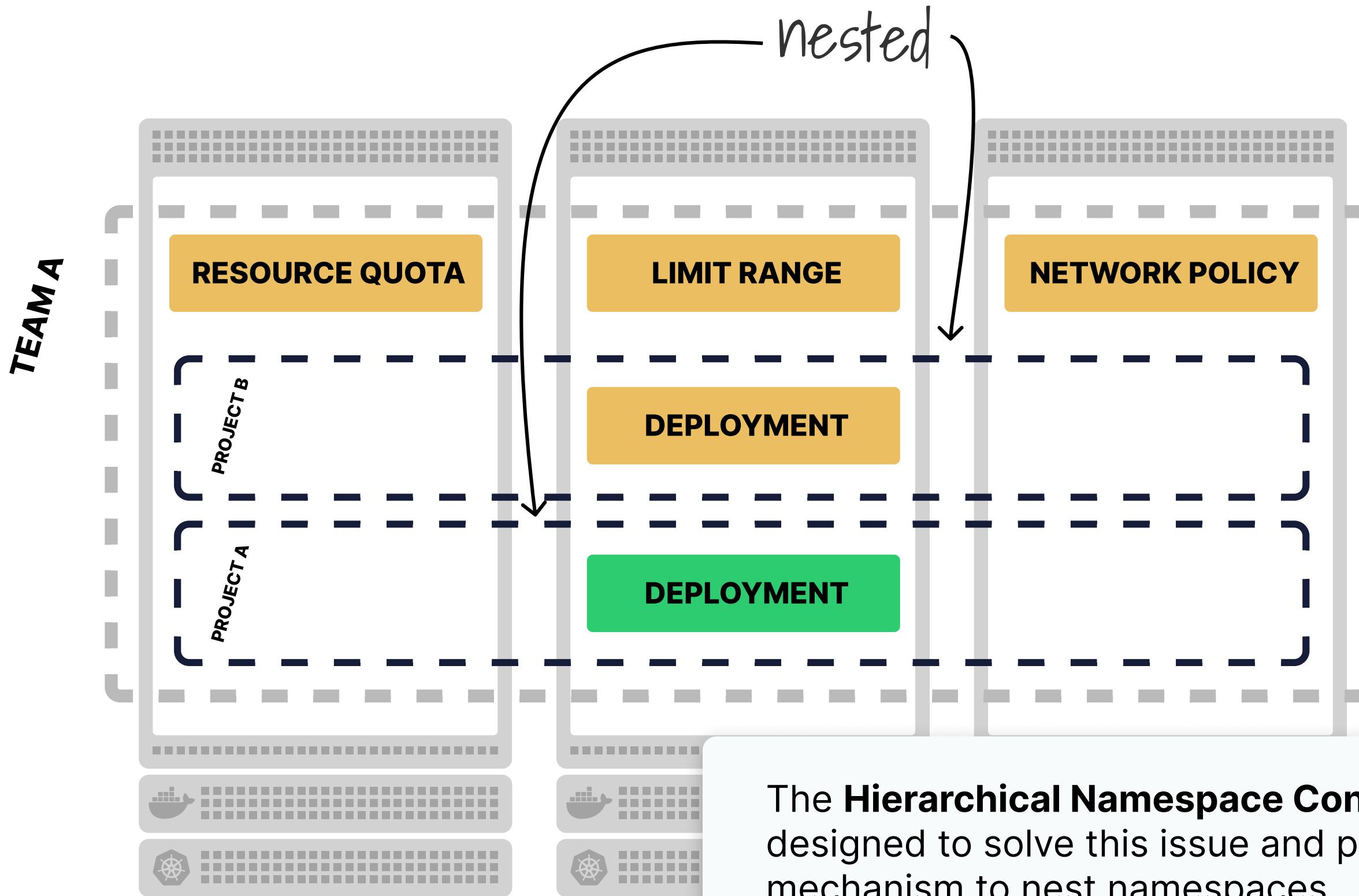


What about Kubernetes namespaces?

Namespaces are a building block for isolation in Kubernetes but aren't designed for multitenancy.

If you have a team with two projects, should those be in the same or different namespace?

And are you forced to duplicate some resources if they are in different namespaces?

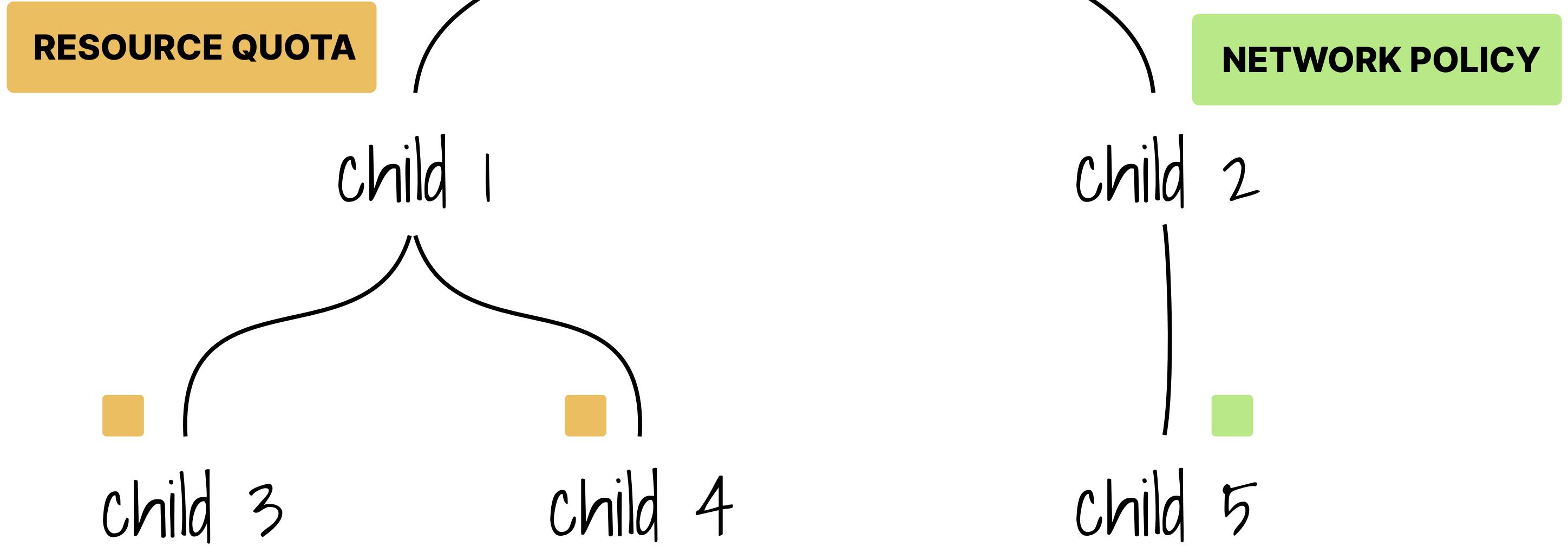


The **Hierarchical Namespace Controller (HNC)** is designed to solve this issue and provide a mechanism to nest namespaces.

All child namespaces inherit resources from the parent and can be infinitely nested.

The controller is in charge of propagating those resources.

root namespace



It's a clever idea that lets you propagate policy from the top rather than policing every single namespace.

The controller has a (minimal) mechanism for overrides, but the idea is that you shouldn't use it unless it's necessary.

There's another open-source tool that provides a similar experience: Capsule.

✗ Namespaces

✗ Nodes

✗ Persistent Volumes

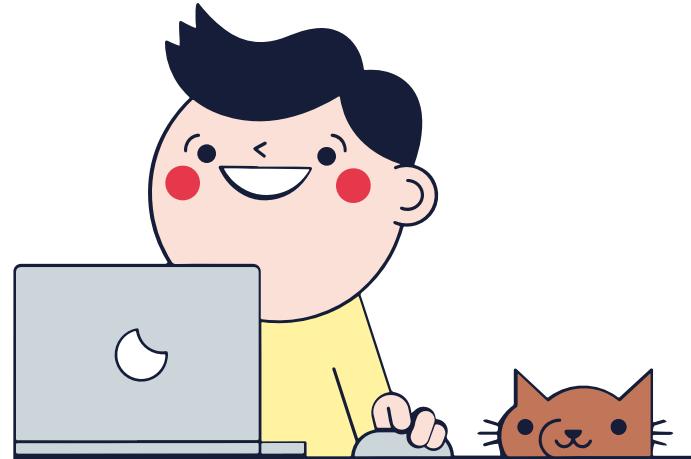
✗ CRDs

But they both point to namespace
global resources.

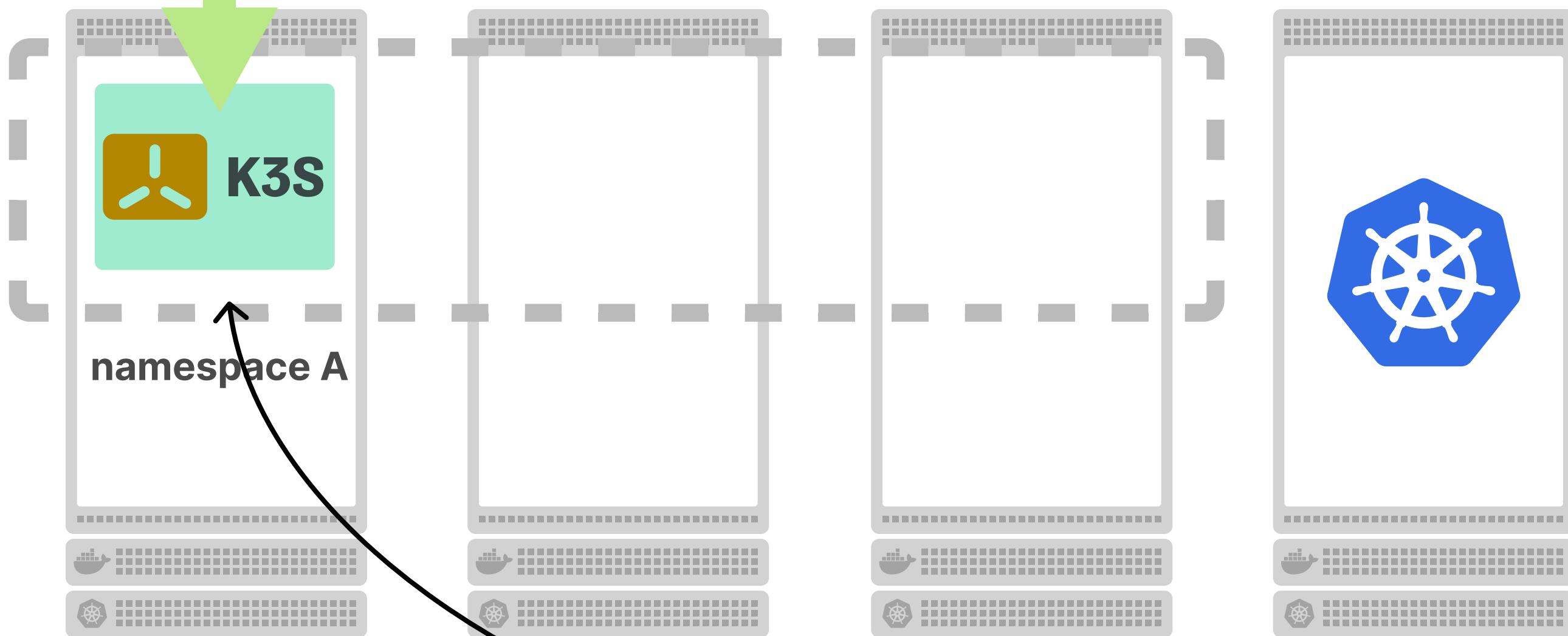
A **Custom Resource Definition (CRD)** is
global by definition in Kubernetes.

So, if one of the tenants wants to install a
CRD, all other tenants will also be able to
see them.

How can you fix this?



kubectl apply



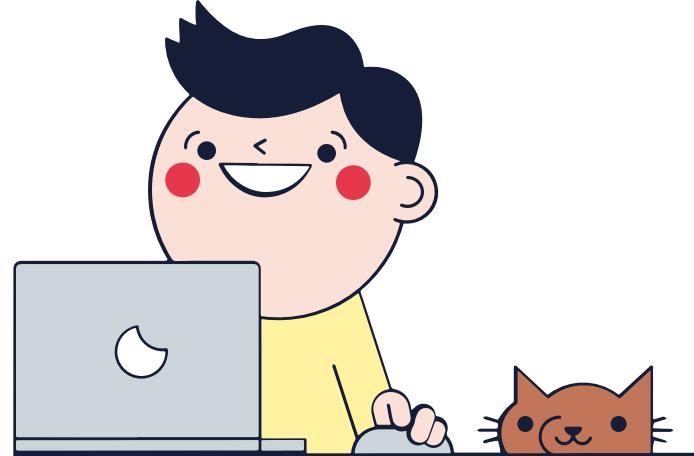
Global resources are stored in the control plane.

What if you could have a control plane per tenant?

That's the idea of tools such as vCluster that run a **K3s control plane as a pod**.

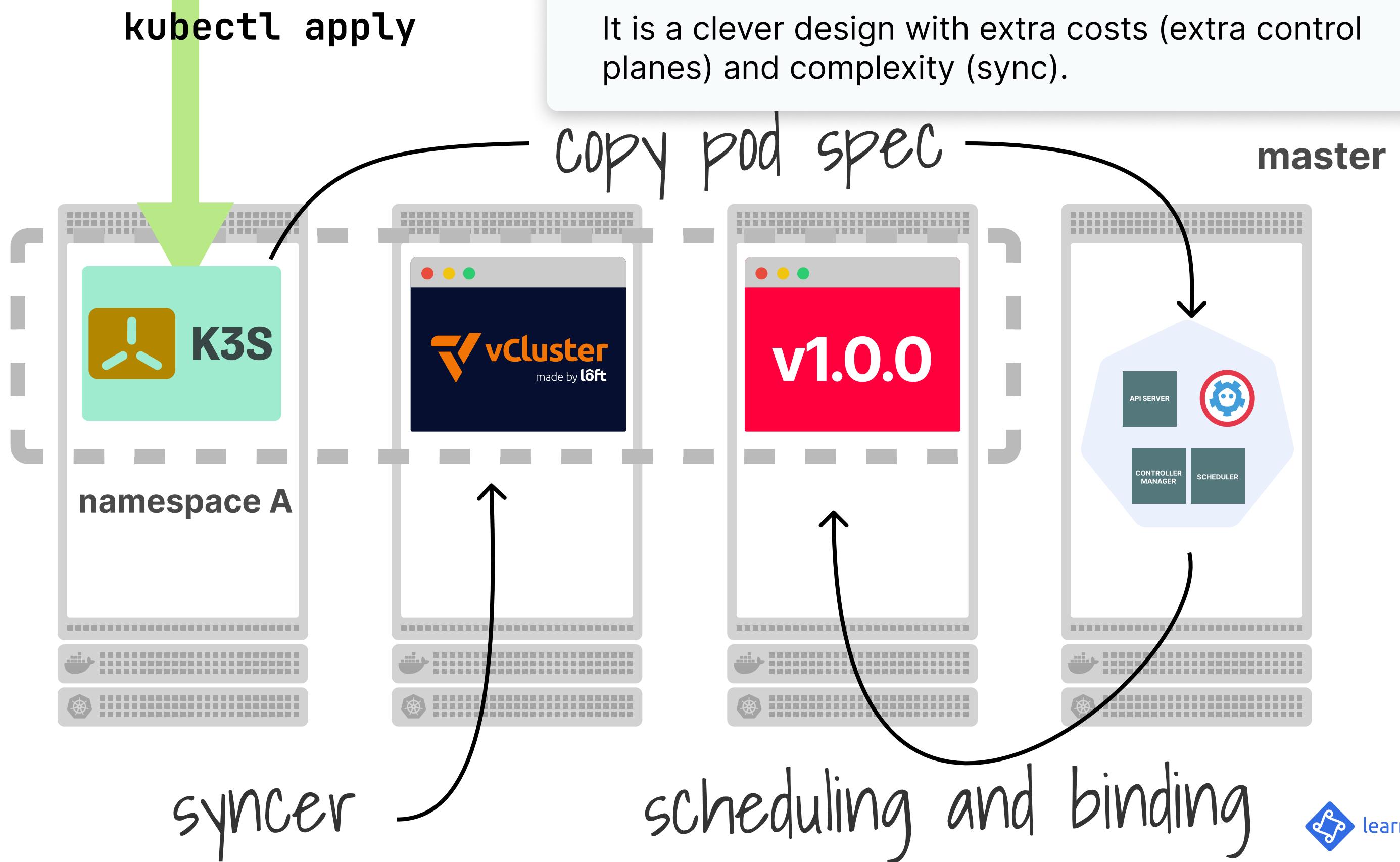
All kubectl commands are sent to that control plane, and no global resource is shared with others (unless you want to).

it has no nodes!

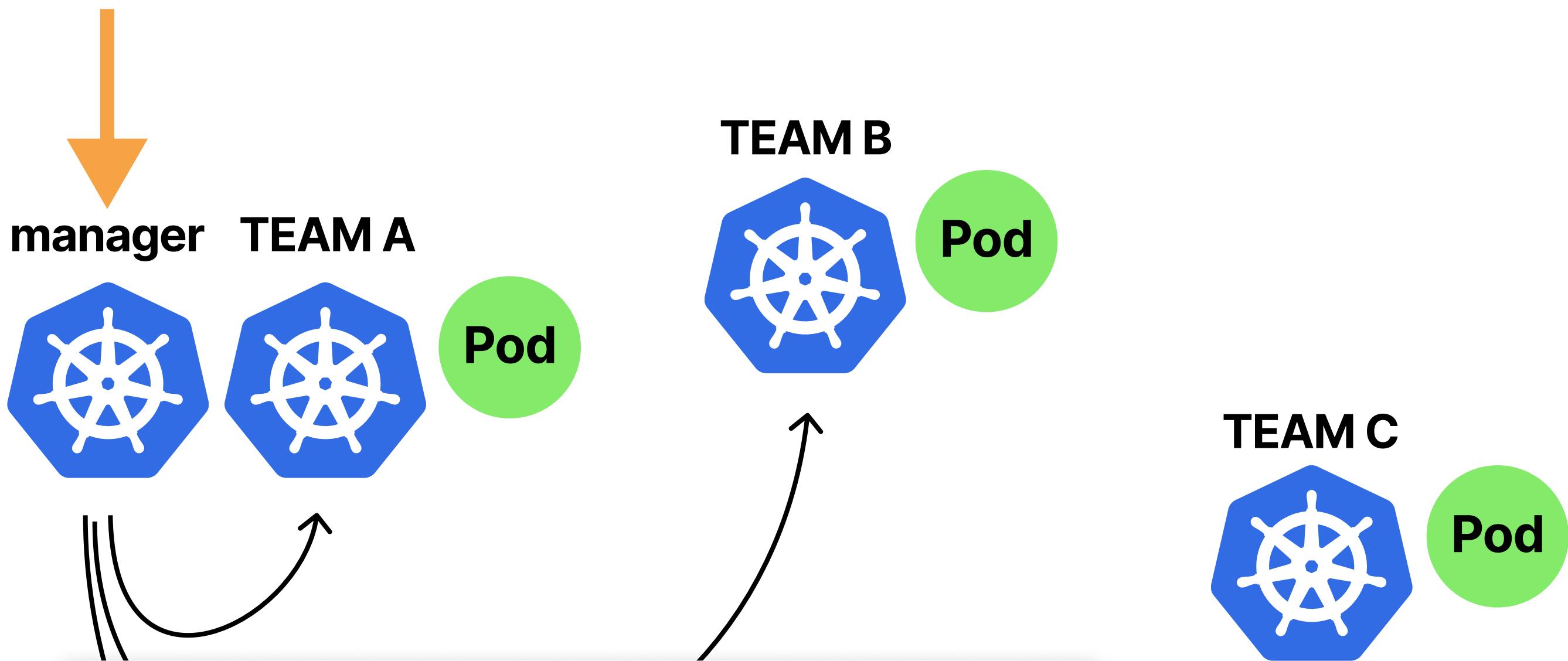


How does a nested control plane create the pods, though? It doesn't.

Pods are stored in the control plane, and **the external controller copies them from the tenant to the global cluster**, where they are scheduled on the nodes.



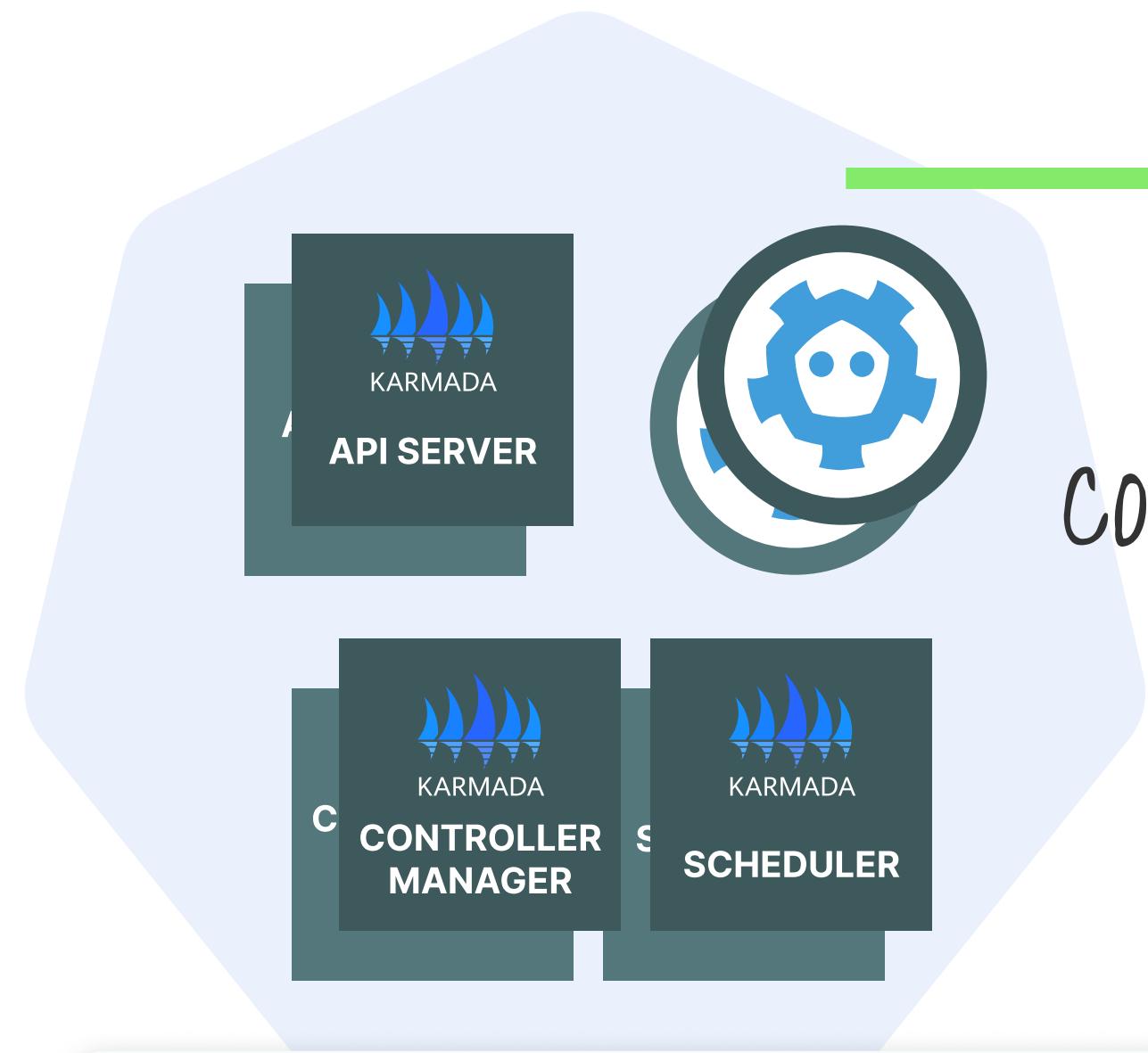
kubectl



What if you need to segregate workloads into different clusters for regulatory reasons?

Another option is to have a dedicated cluster per tenant.

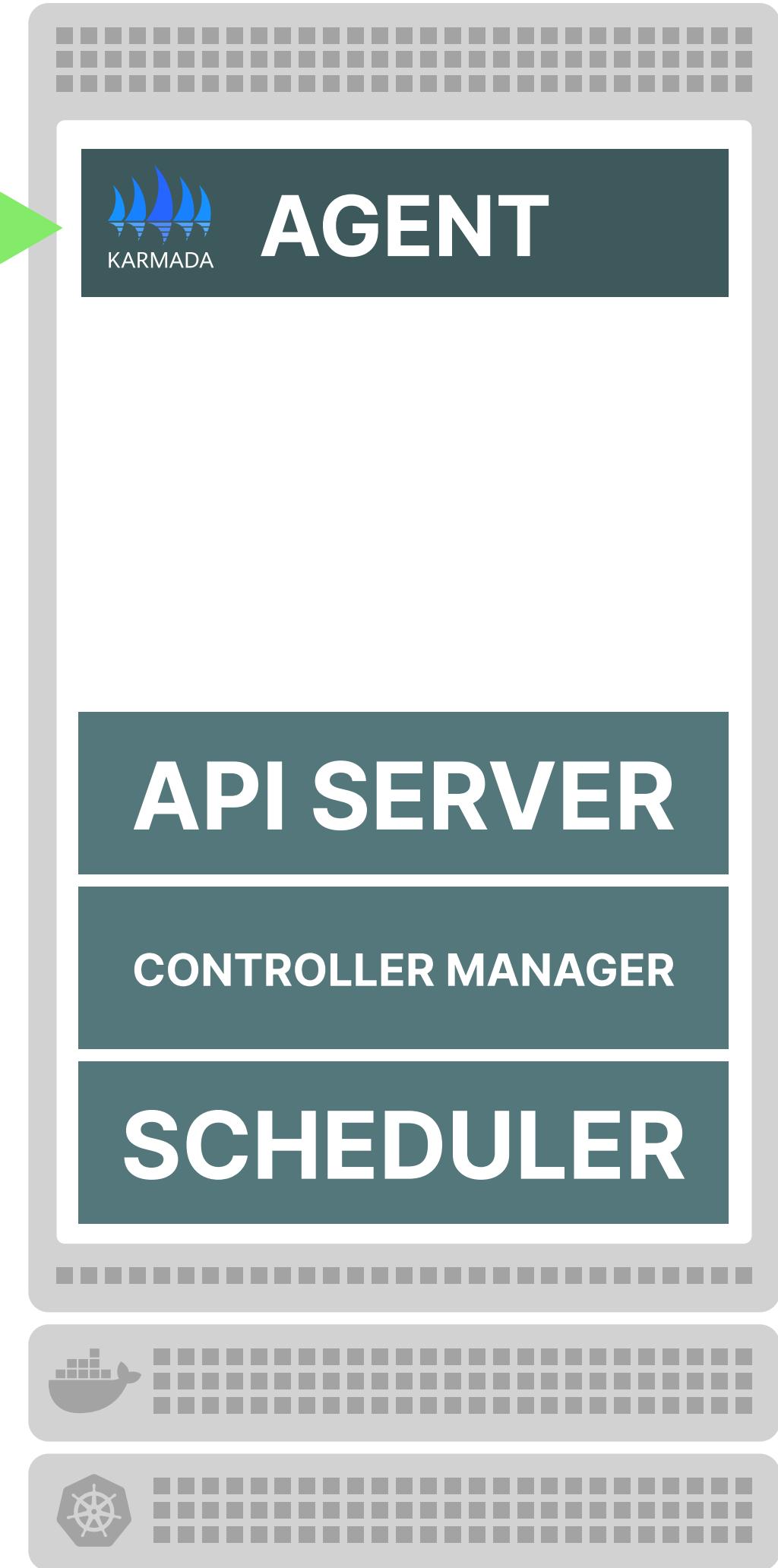
You could use **Karmada** to manage the tenant cluster and deploy common workloads all once across all clusters.



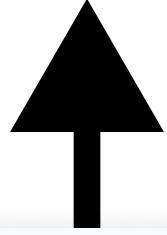
Karmada's architecture is similar to vcluster.

Instead of a controller, **Karmada employs an agent that receives instructions from the Karmada control plane and forwards them to the local cluster.**

Of all options, this is the most expensive to maintain and operate. It's also the solution that offers more isolation.



Cost



Isolations and ease of management aren't the only trade-offs you should consider.

Each option has costs, too.

Running extra management clusters (Karamada) or control planes (vCluster) costs money compared to a simple controller that propagates resources (HNC).



K8S

HNC

vCluster Karmada

Join my session on building Kubernetes platforms
this Thursday at 8am PT / 5pm CET!

ARCHITECTING KUBERNETES SINGLE CLUSTER OR TO EACH THEIR OWN?

29th of Feb

8am PT | 5pm CET



REGISTER HERE

bit.ly/multitenancy1

	COST-EFFICIENCY	EASE OF MANAGEMENT	RESILIENCE	APPLICATION SECURITY
LARGE SHARED CLUSTER	High	High	Low	Low
CLUSTER PER ENVIRONMENT	Medium	Medium	Medium	Medium
CLUSTER PER APPLICATION	Low	Low	High	High
SMALL SINGLE-USE CLUSTERS	Very Low	Very Low	Very High	Very High



Daniele Polencic