HashiCorp
an IBM Company

GPU sharing done right

Adrian Todorov
Staff Solutions Architect
HashiCorp

Scott McAllister
Principal Developer Advocate
Depot

depot

Scott McAllister

Principal Developer Advocate

Depot

HashiCorp
an IBM Company

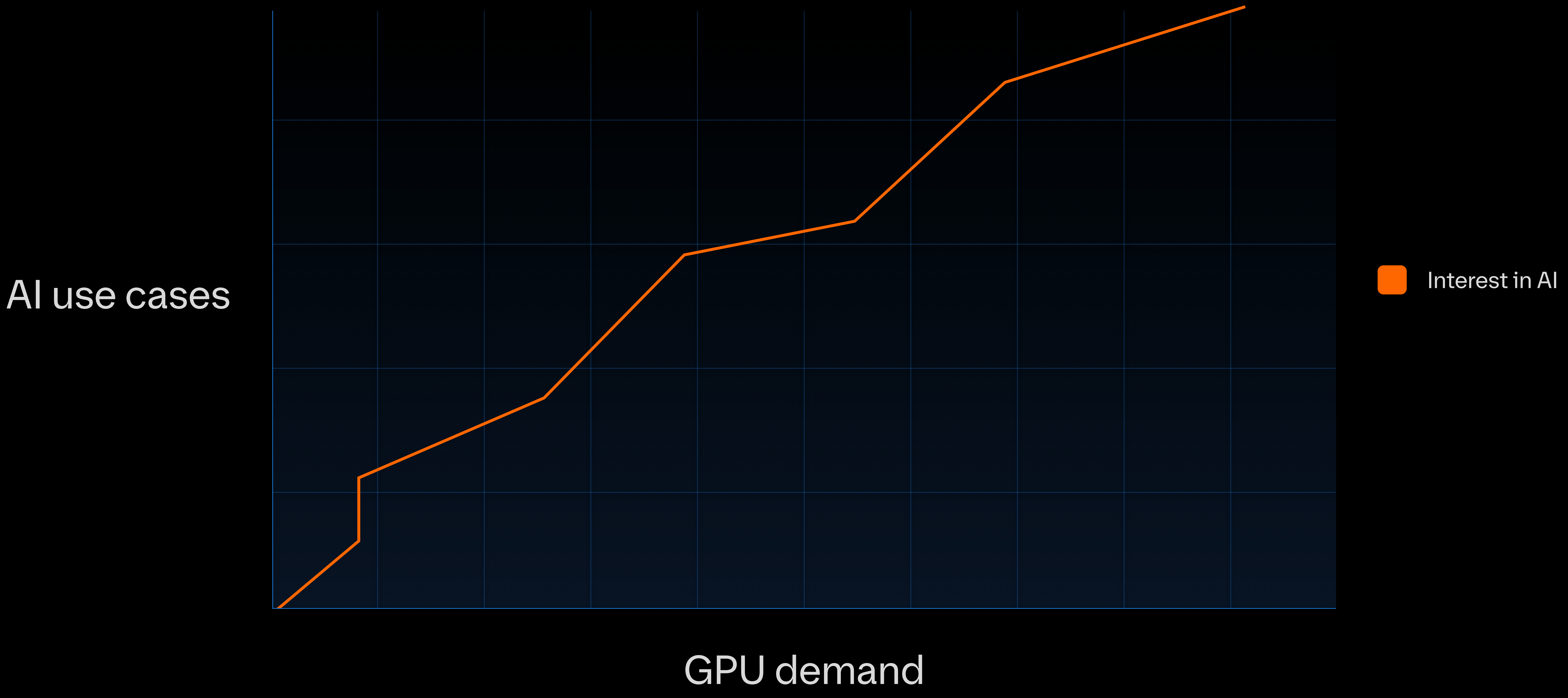
Adrian Todorov

Staff Solutions Architect @ HashiCorp,
ex-SRE, ex-developer
atodorov.me

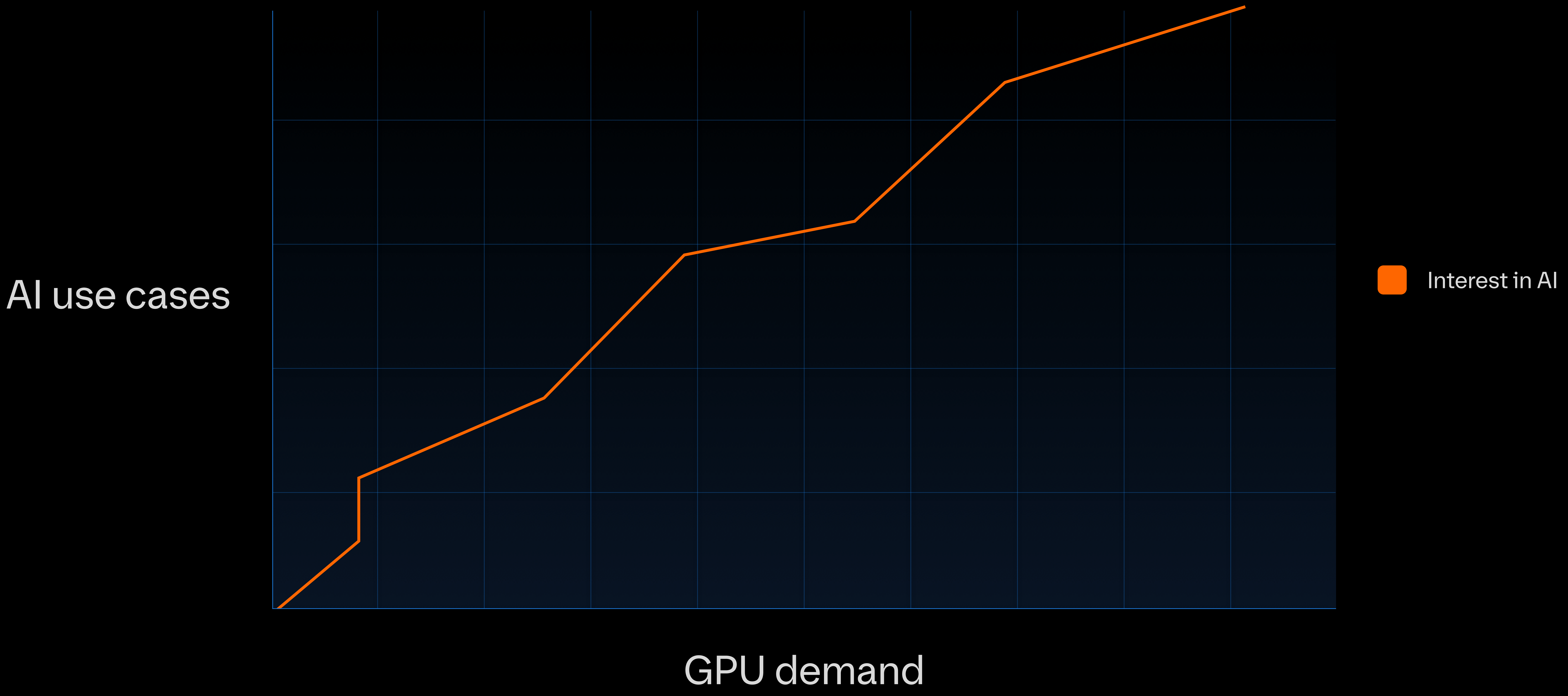
Traditional clusters are like gamers.

Traditional clusters
don't like to share GPU.

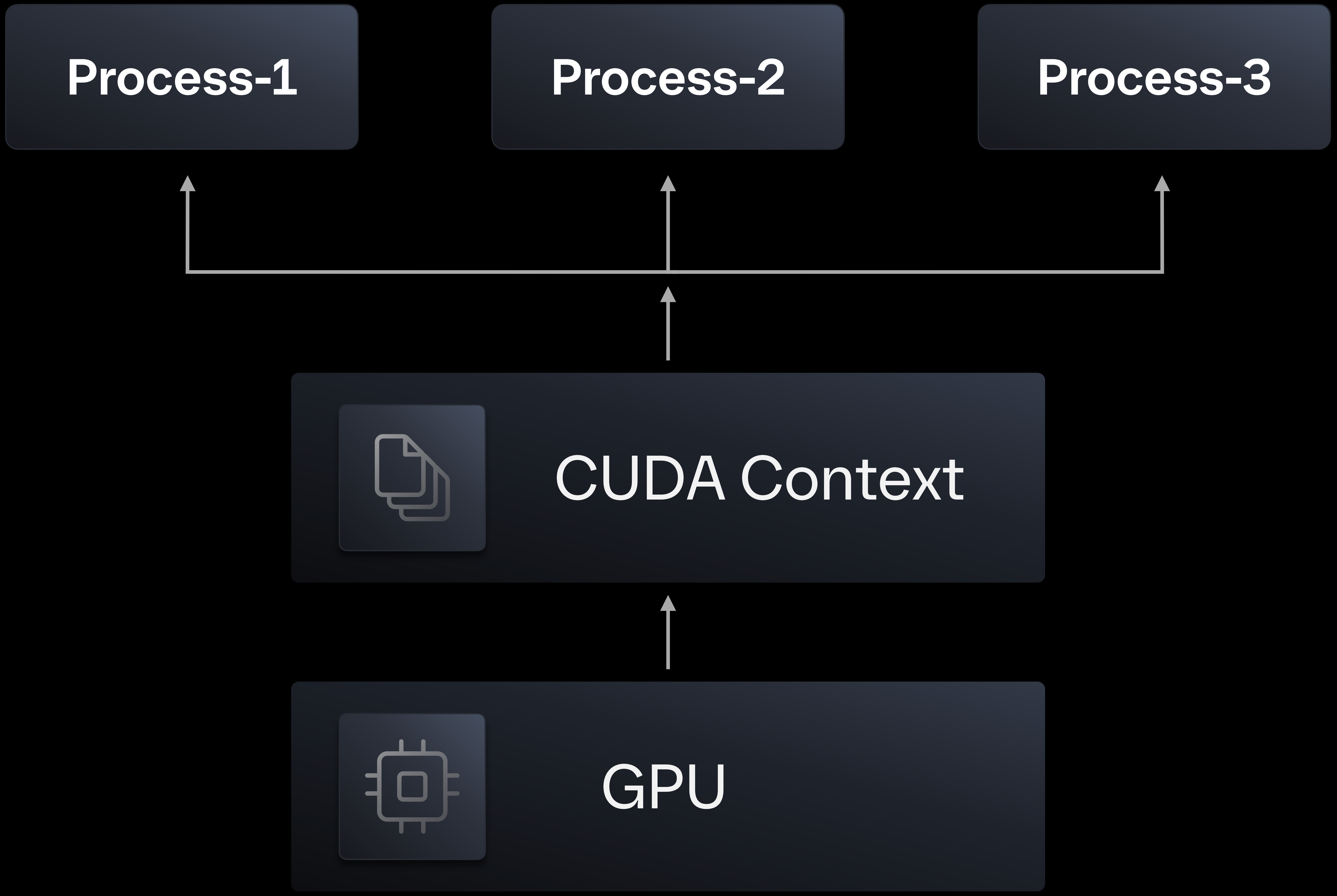
What’s happening



What’s happening



Multi-Process Service (MPS)



Process-1

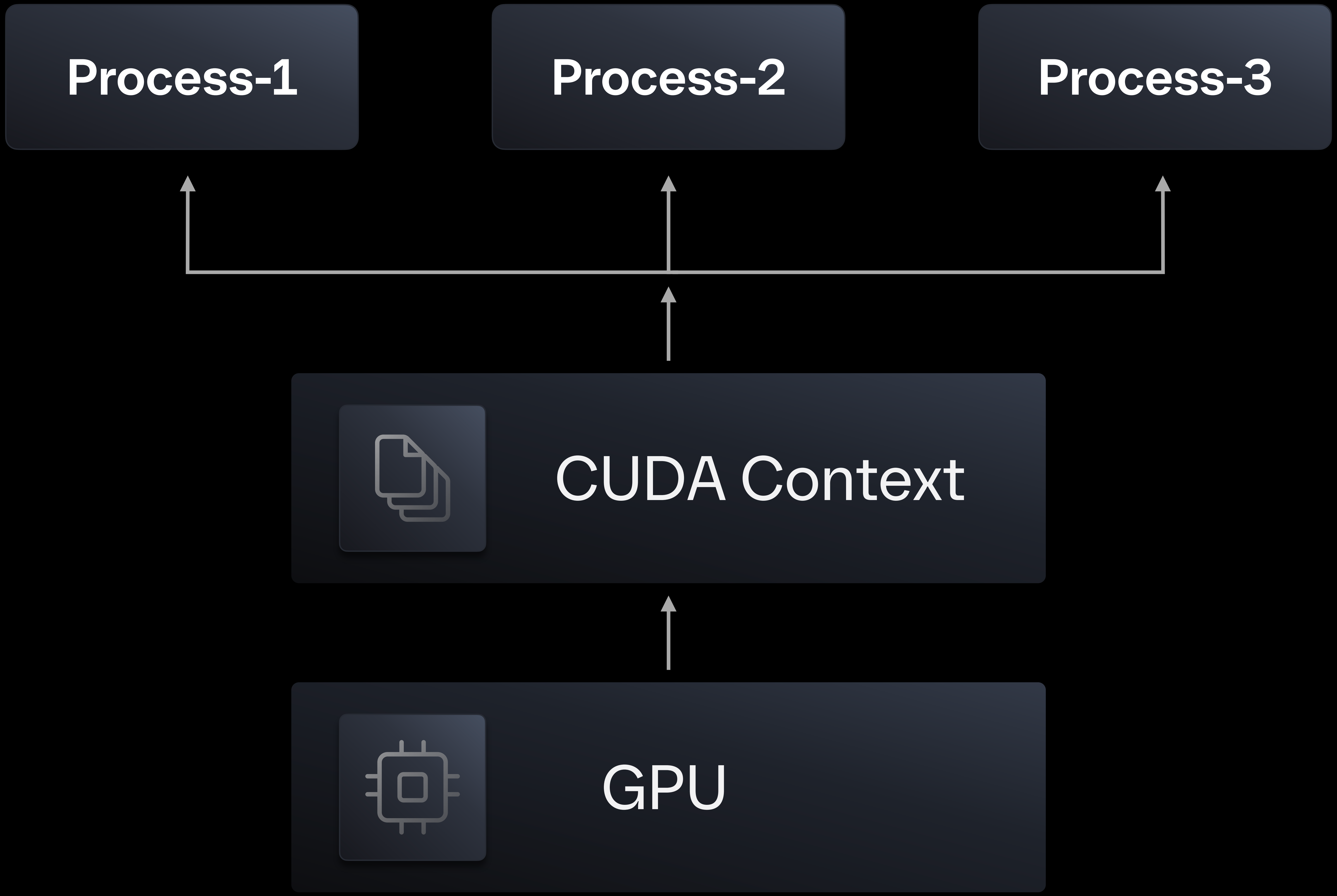
Process-2

Process-3

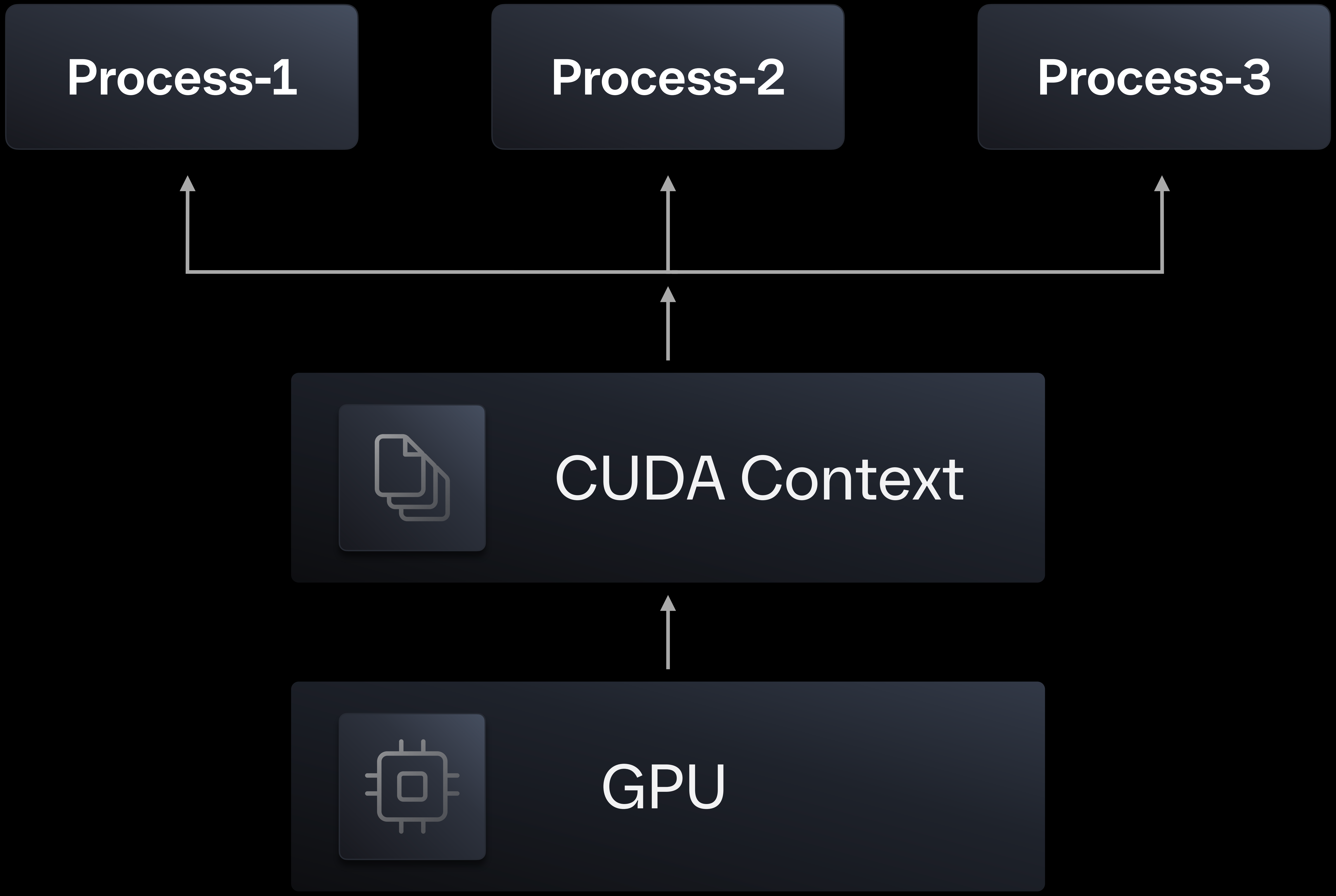
CUDA Context

GPU

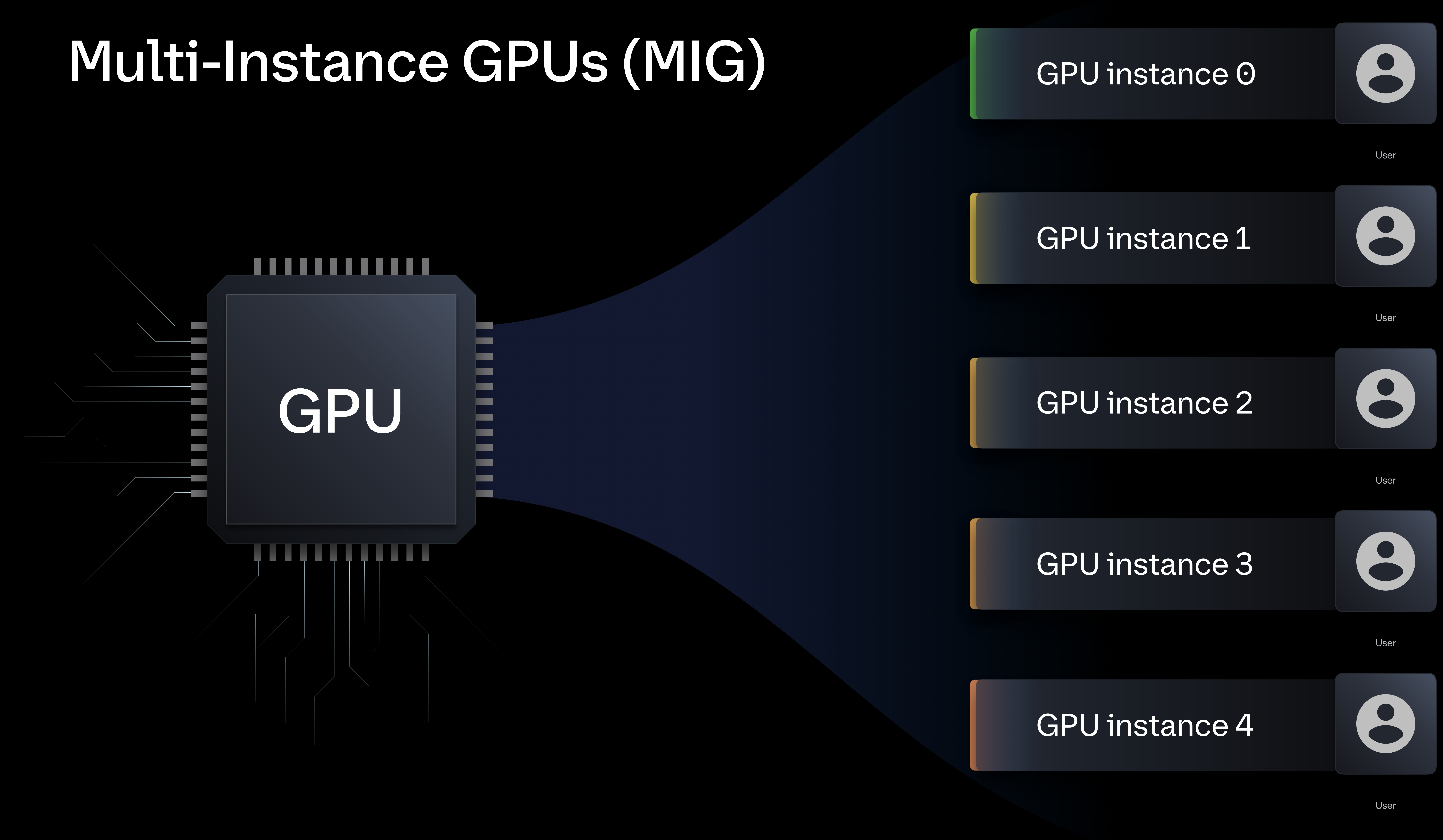
Multi-Process Service (MPS)



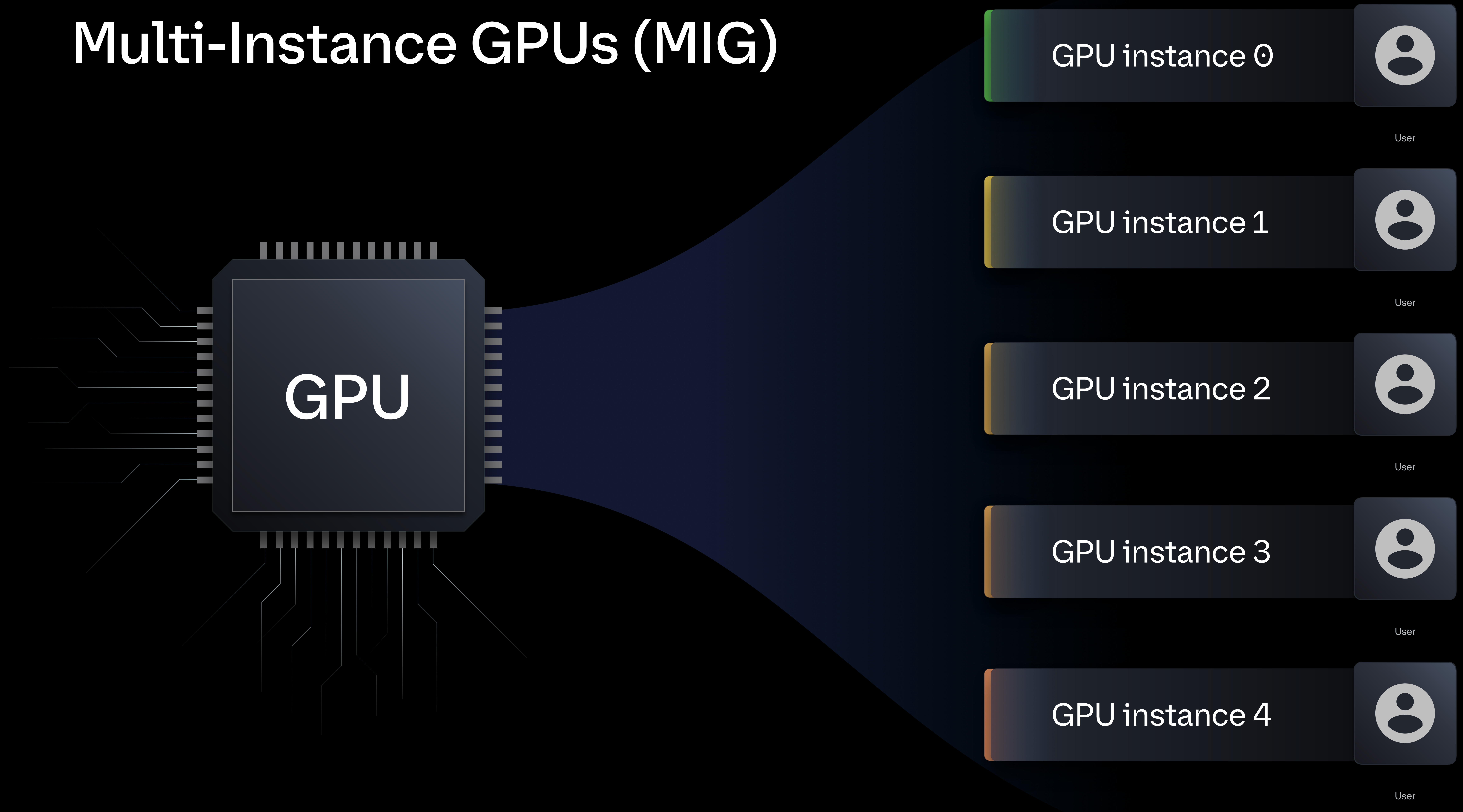
Multi-Process Service (MPS)



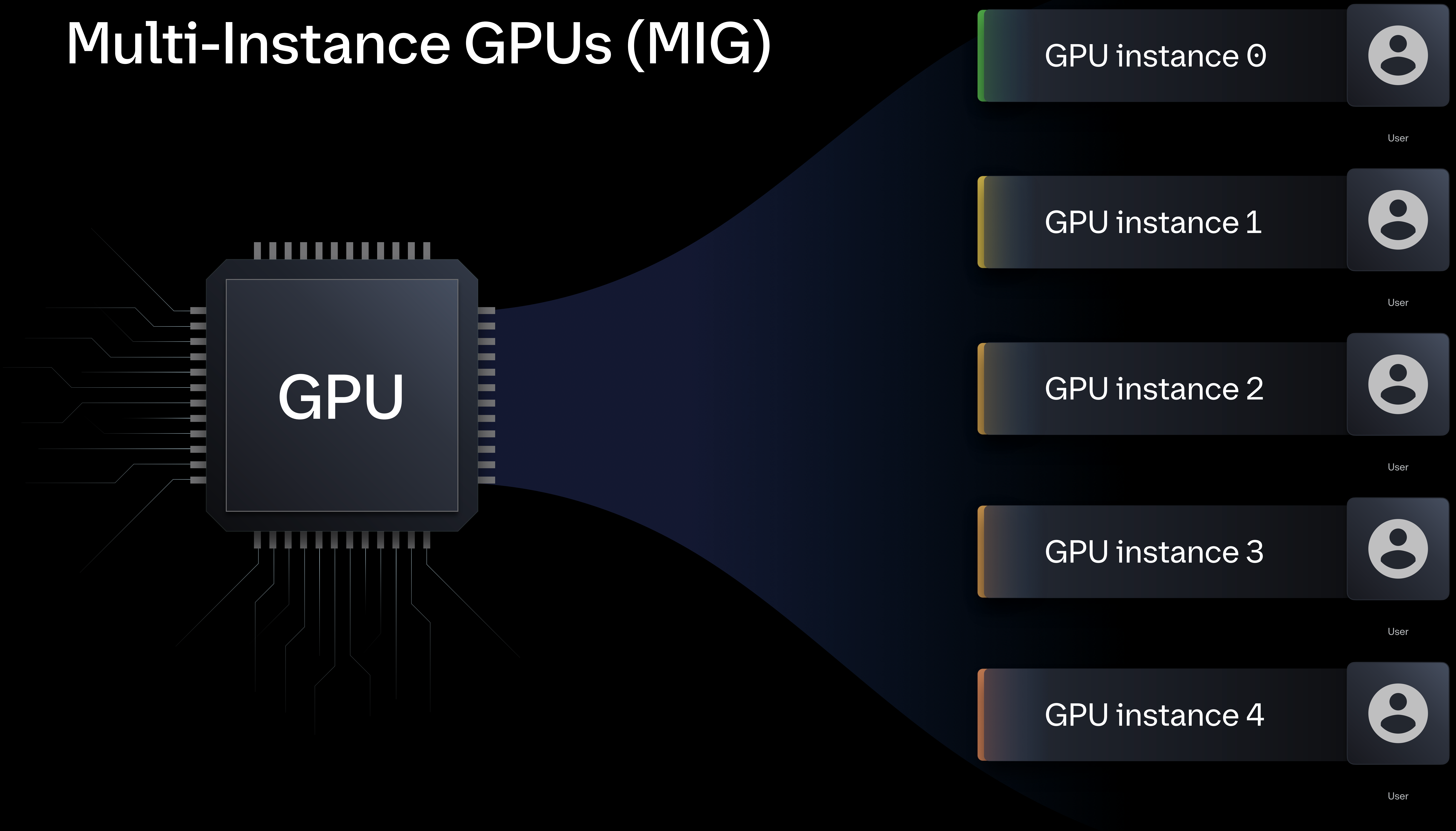
Multi-Instance GPUs (MIG)



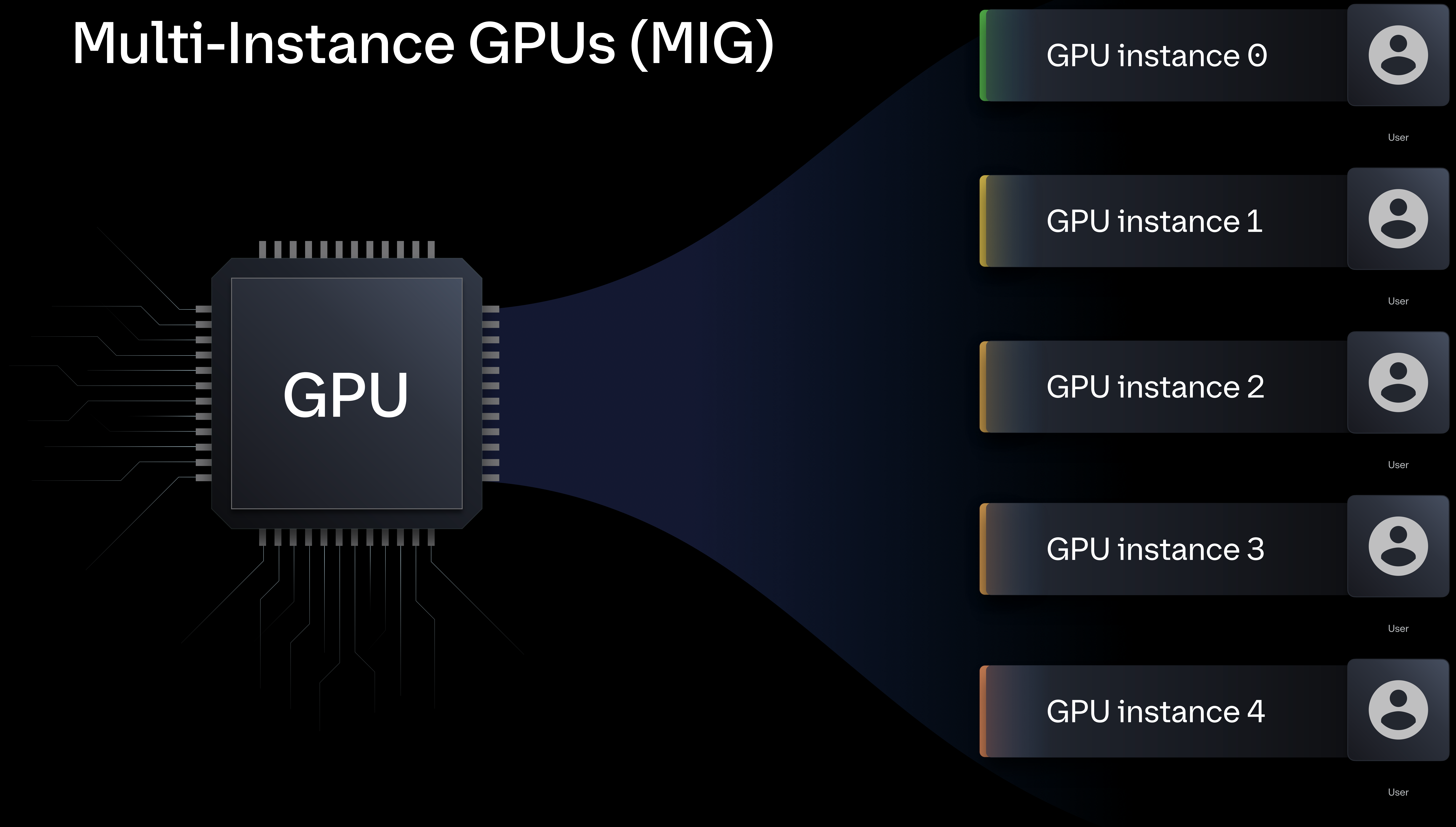
Multi-Instance GPUs (MIG)



Multi-Instance GPUs (MIG)

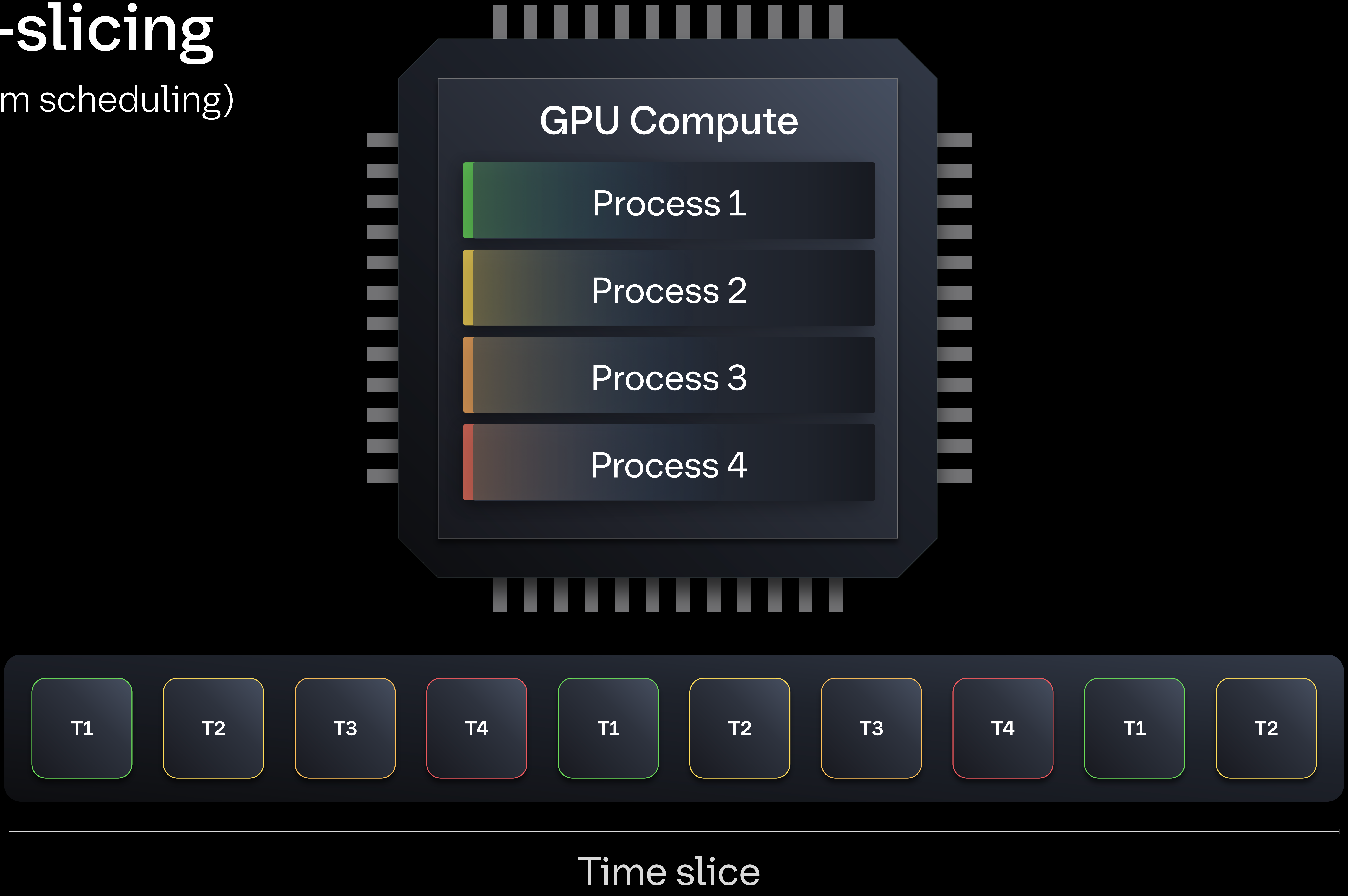


Multi-Instance GPUs (MIG)



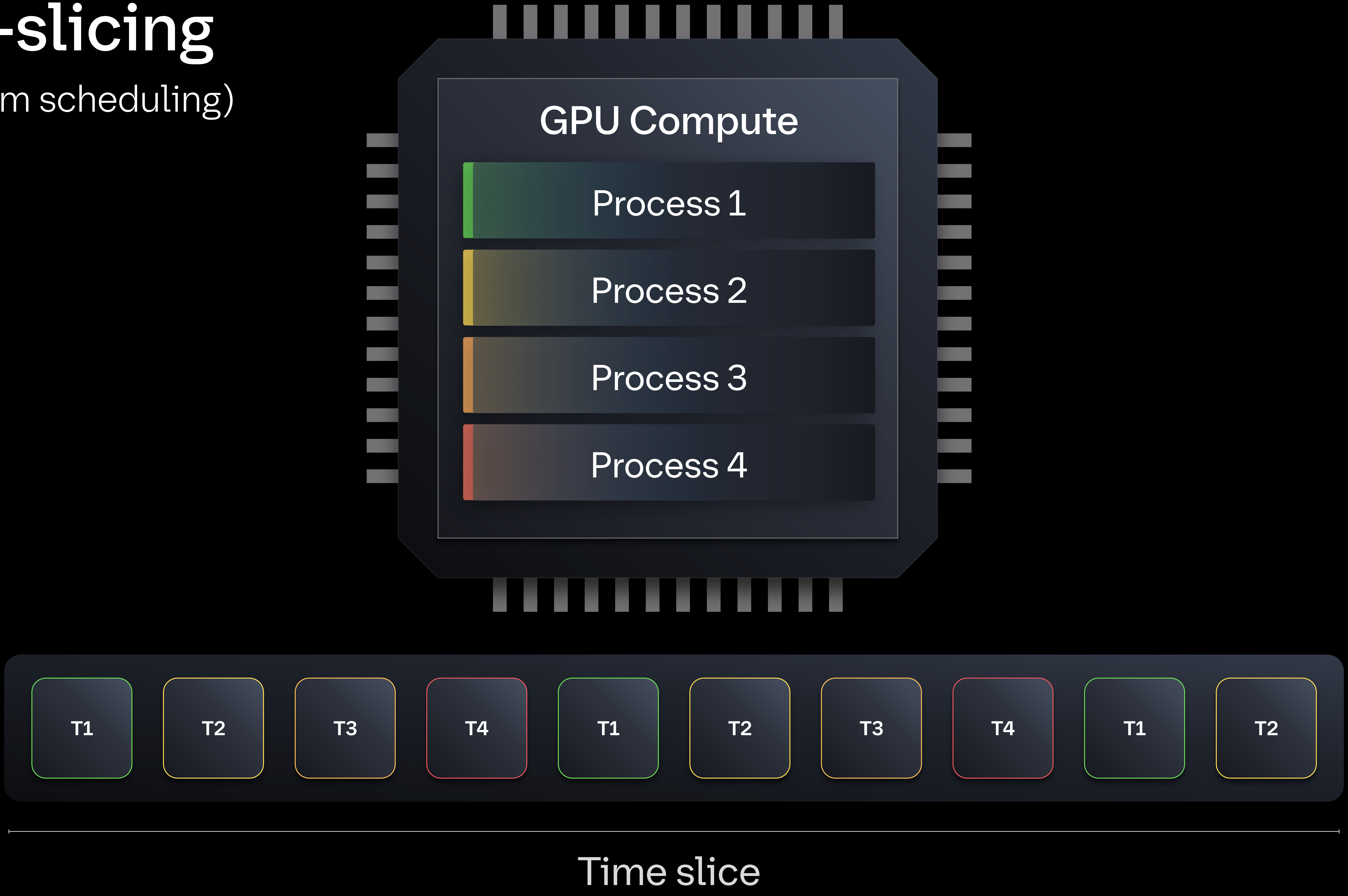
Time-slicing

(Or: custom scheduling)



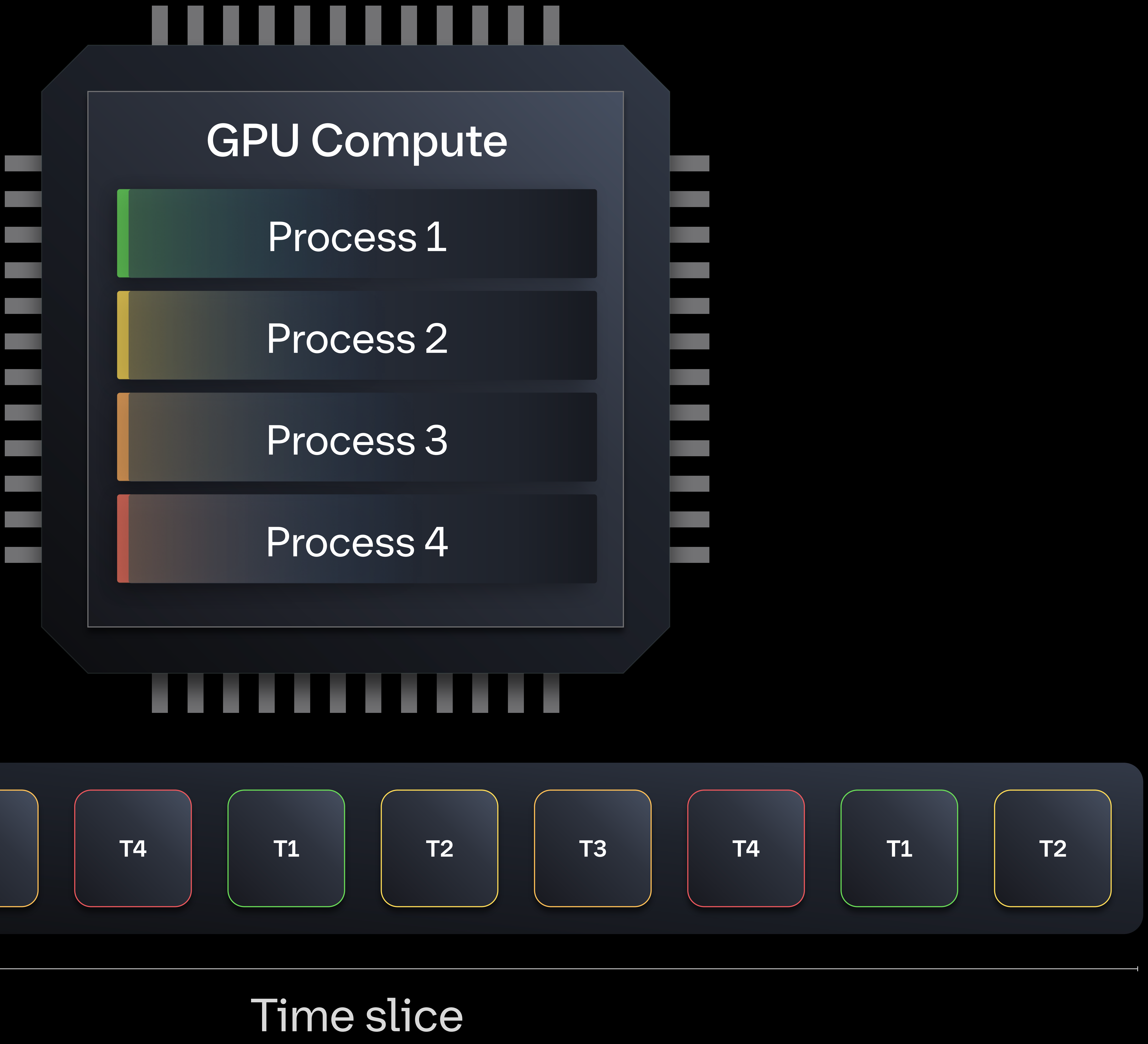
Time-slicing

(Or: custom scheduling)



Time-slicing

(Or: custom scheduling)



GPU Compute

Process 1

Process 2

Process 3

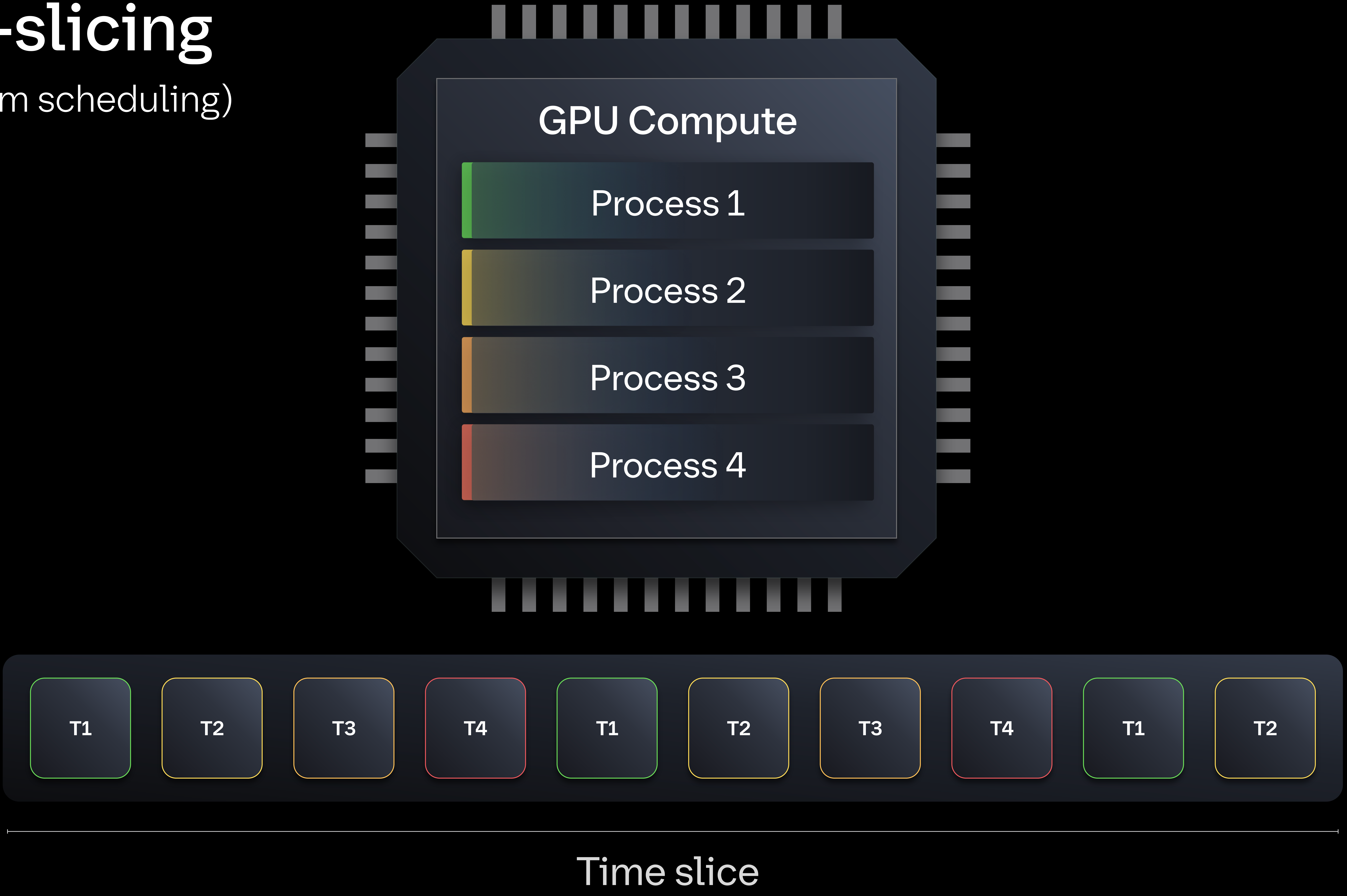
Process 4

T1	T2	T3	T4	T1	T2	T3	T4	T1	T2

Time slice

Time-slicing

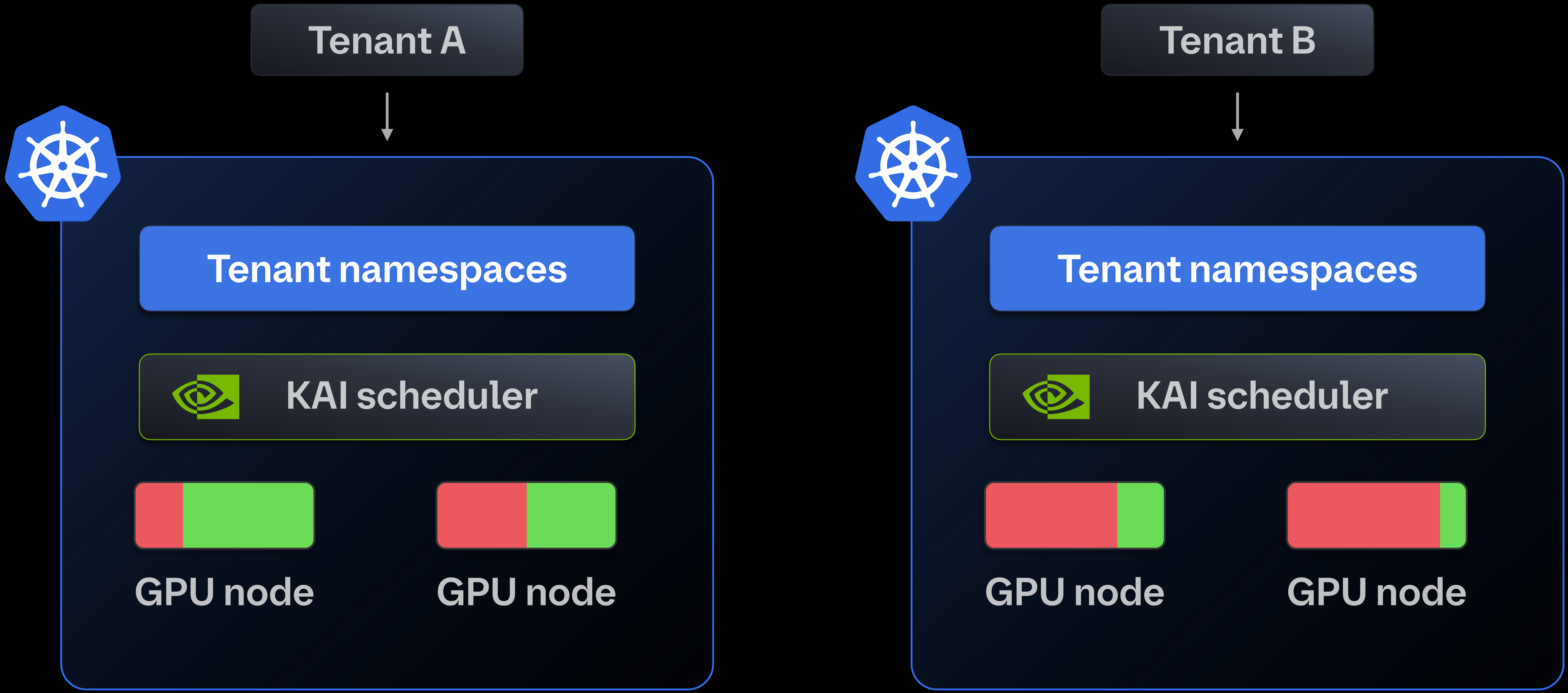
(Or: custom scheduling)



Third-Party Scheduling Tools

Third-Party Scheduling Tools

Per-team GPU cluster



Tenant A

Tenant namespaces

KAI scheduler

GPU node

GPU node

Tenant B

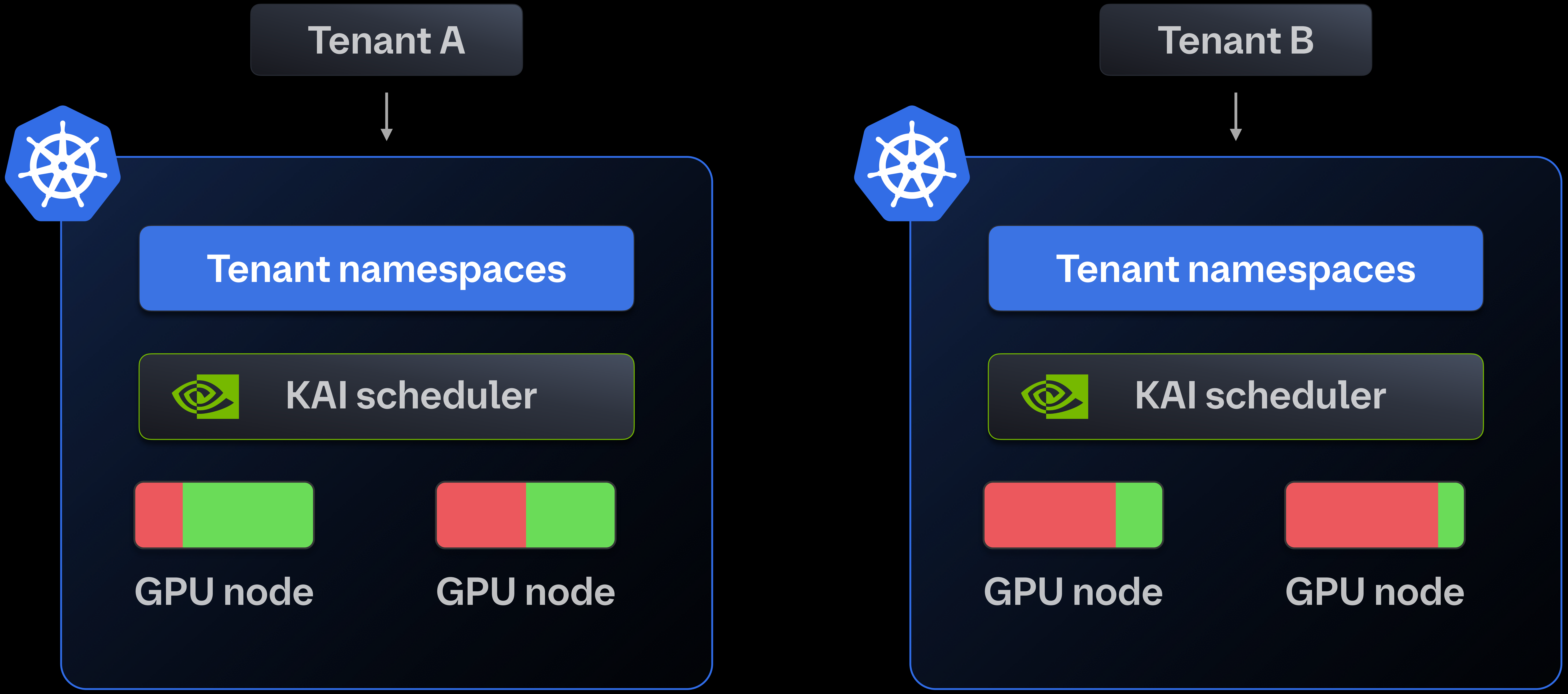
Tenant namespaces

KAI scheduler

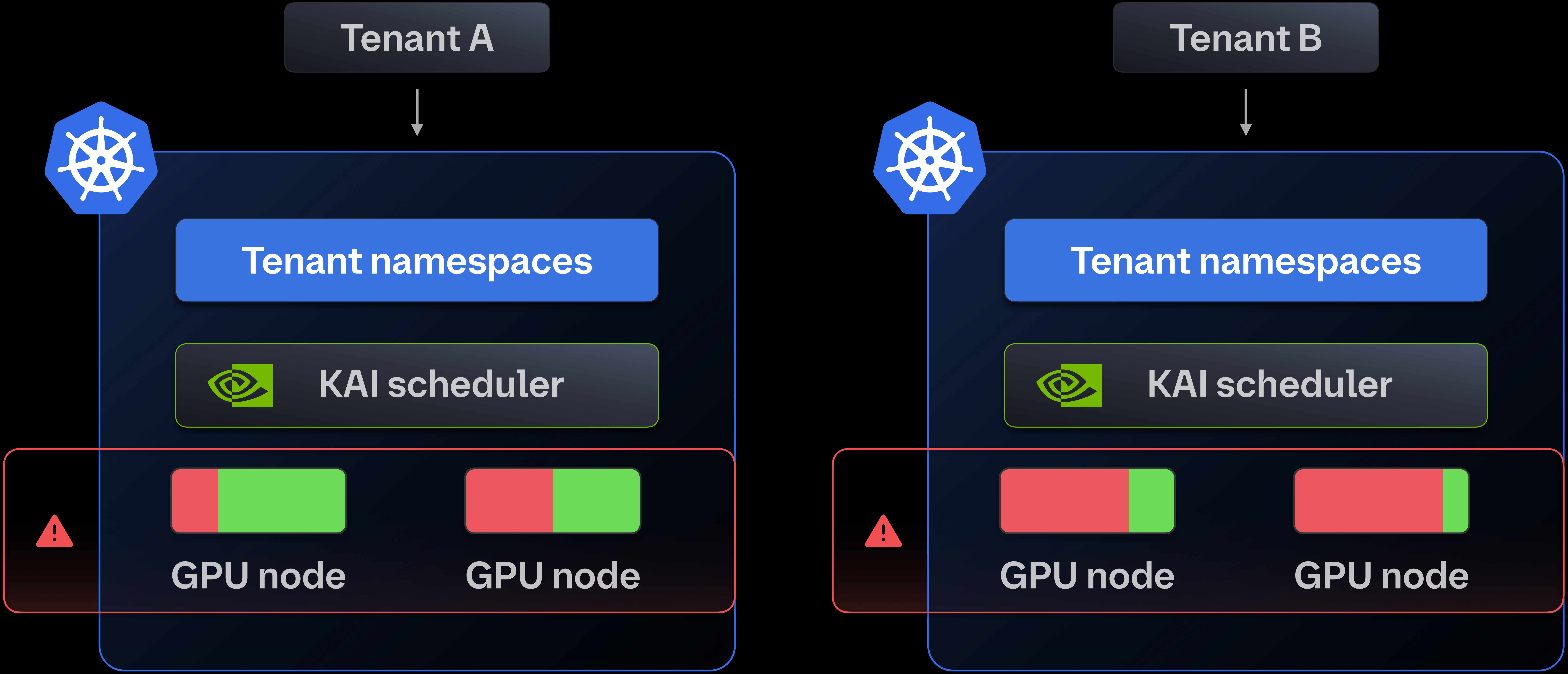
GPU node

GPU node

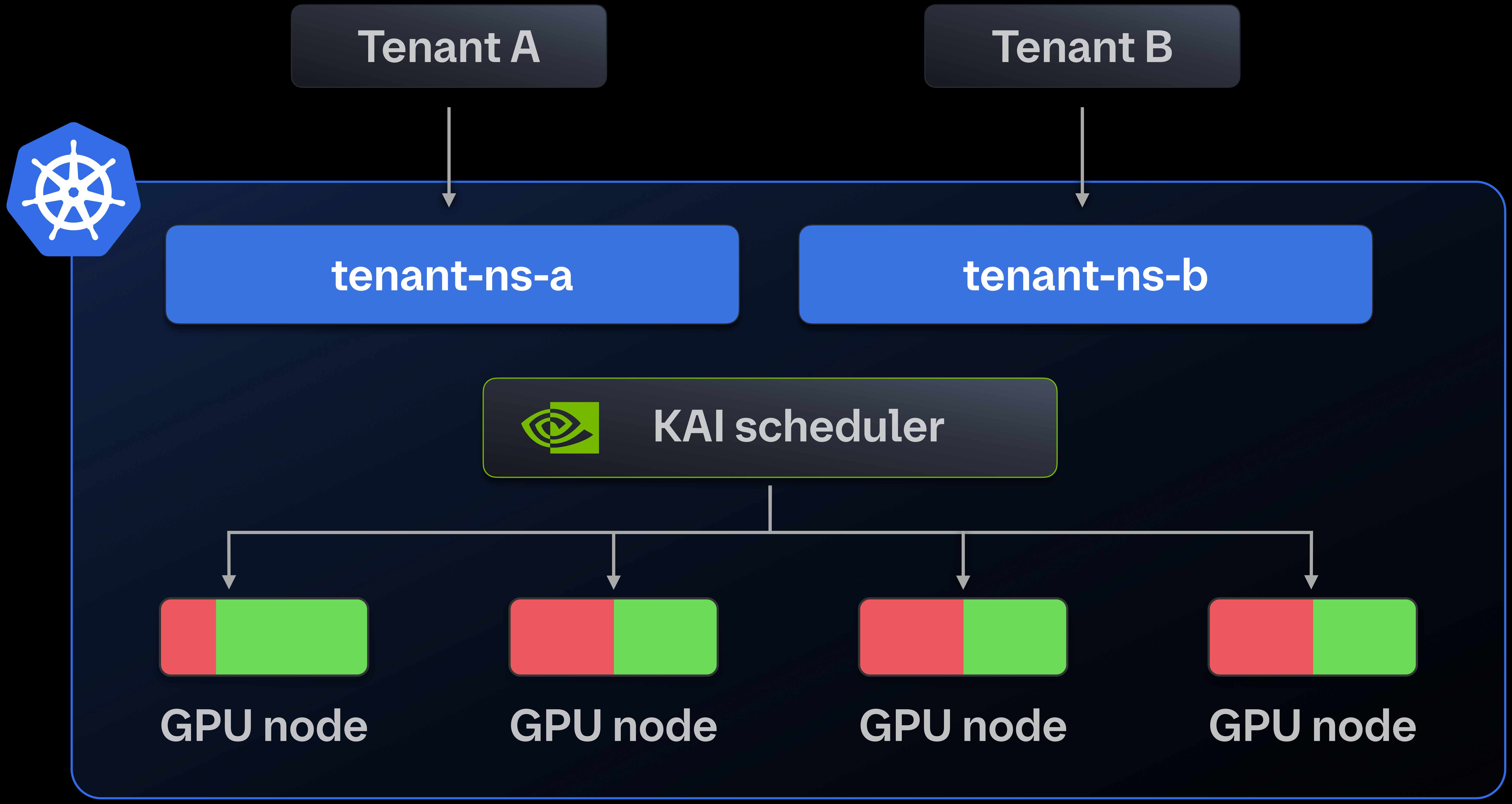
Per-team GPU cluster



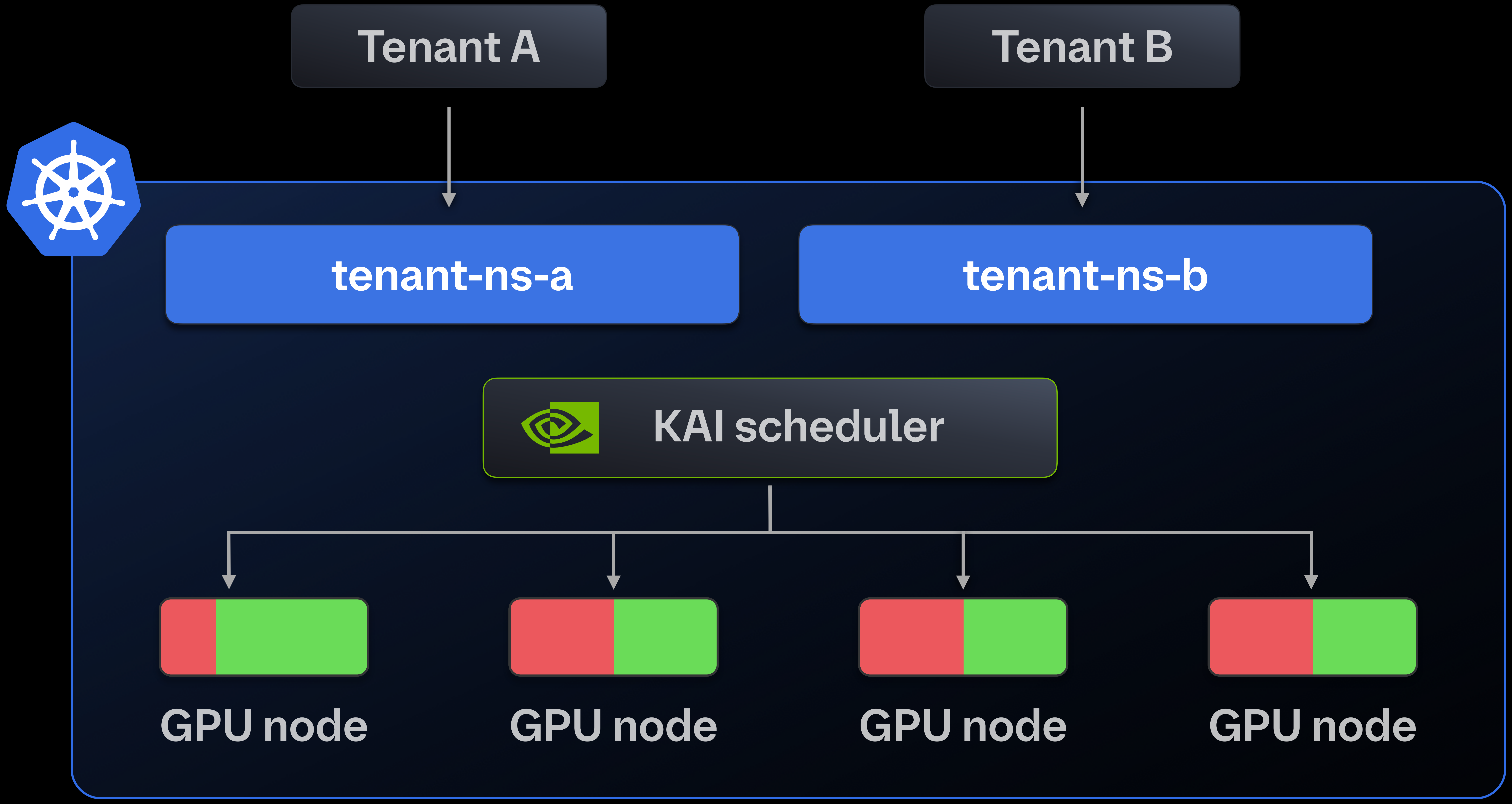
Per-team GPU cluster



Namespace-based multi-tenancy



Namespace-based multi-tenancy



Tenant A

Tenant B

tenant-ns-a

tenant-ns-b

KAI scheduler

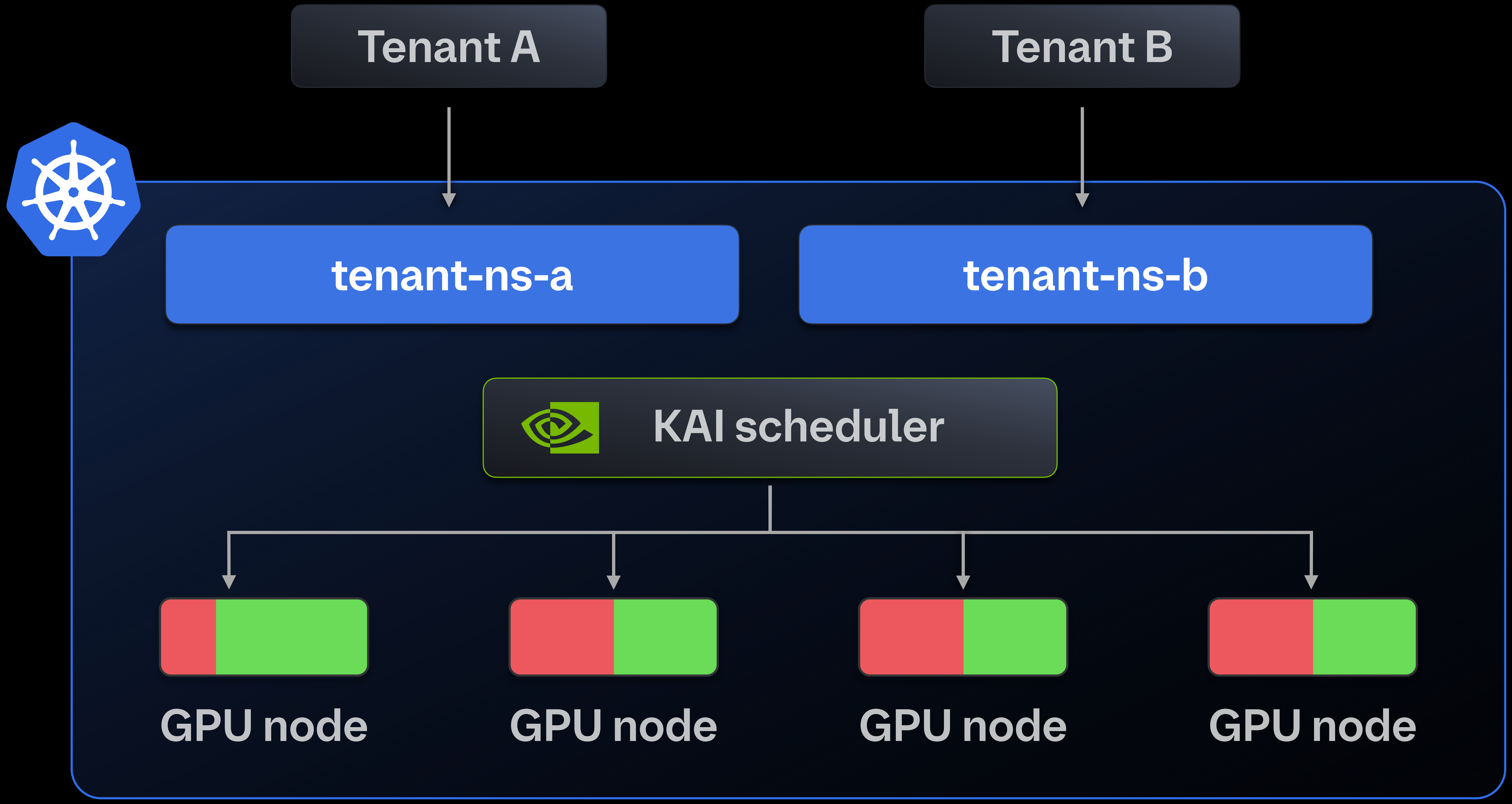
GPU node

GPU node

GPU node

GPU node

Namespace-based multi-tenancy



Tenant A

Tenant B

tenant-ns-a

tenant-ns-b

KAI scheduler

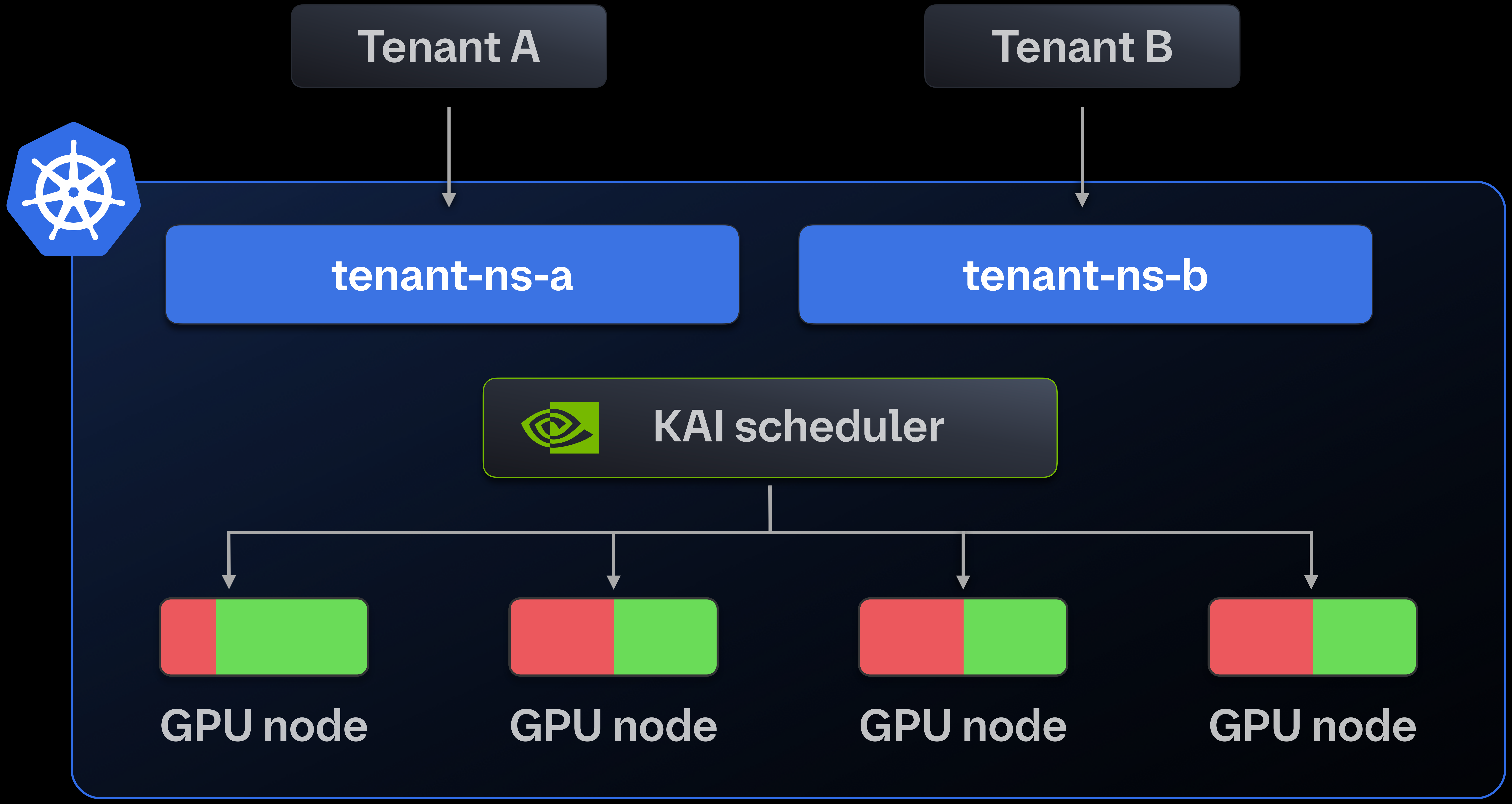
GPU node

GPU node

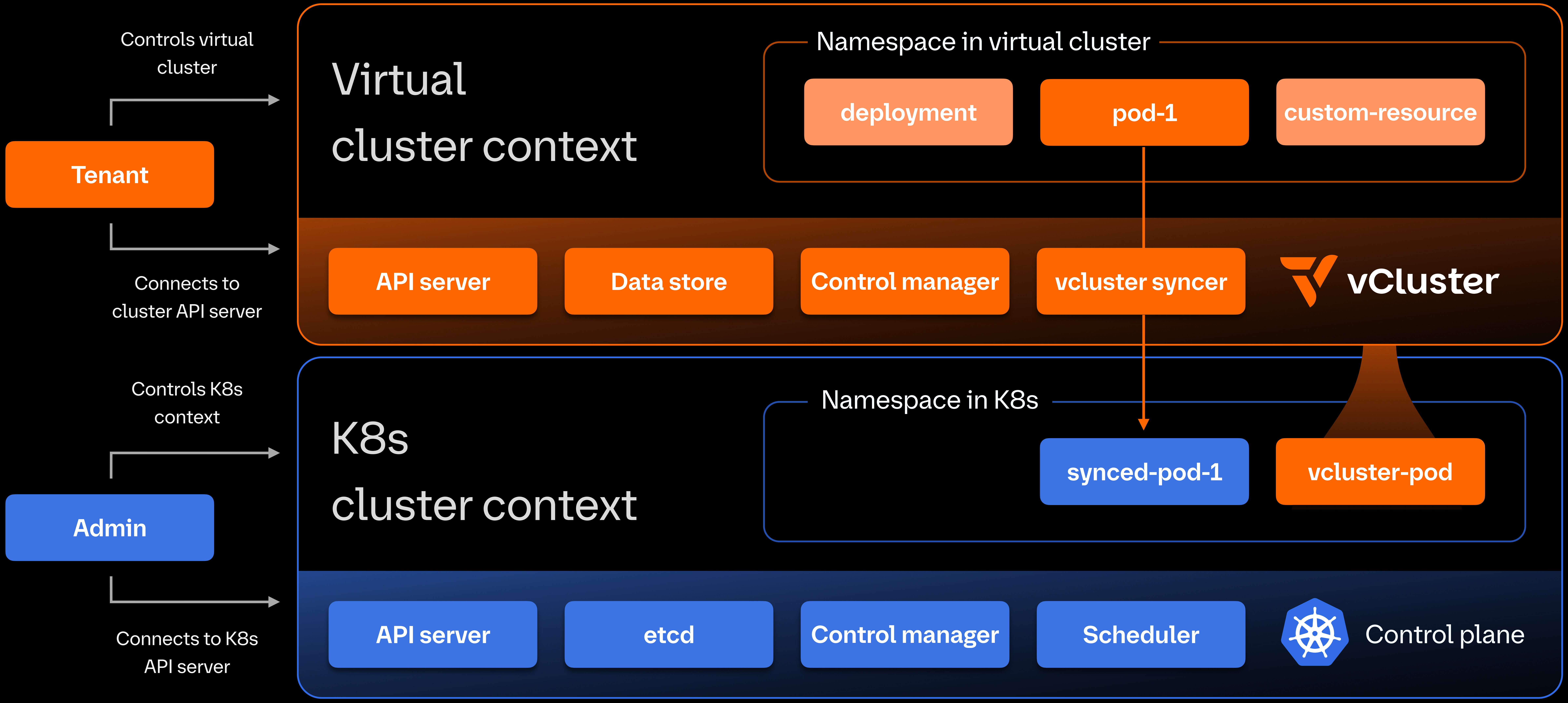
GPU node

GPU node

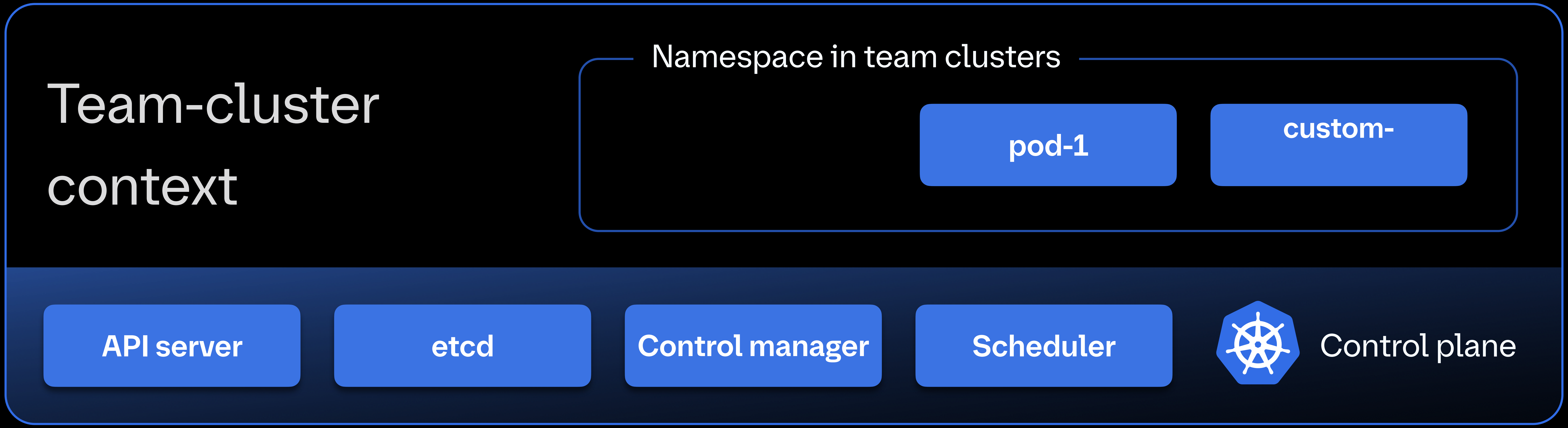
Namespace-based multi-tenancy



Team-shared cluster



Team-shared cluster



GPU sharing only works when you understand your trust boundaries.

Namespaces provide logical segmentation, but not isolation.

Per-team clusters address isolation, but not efficiency.

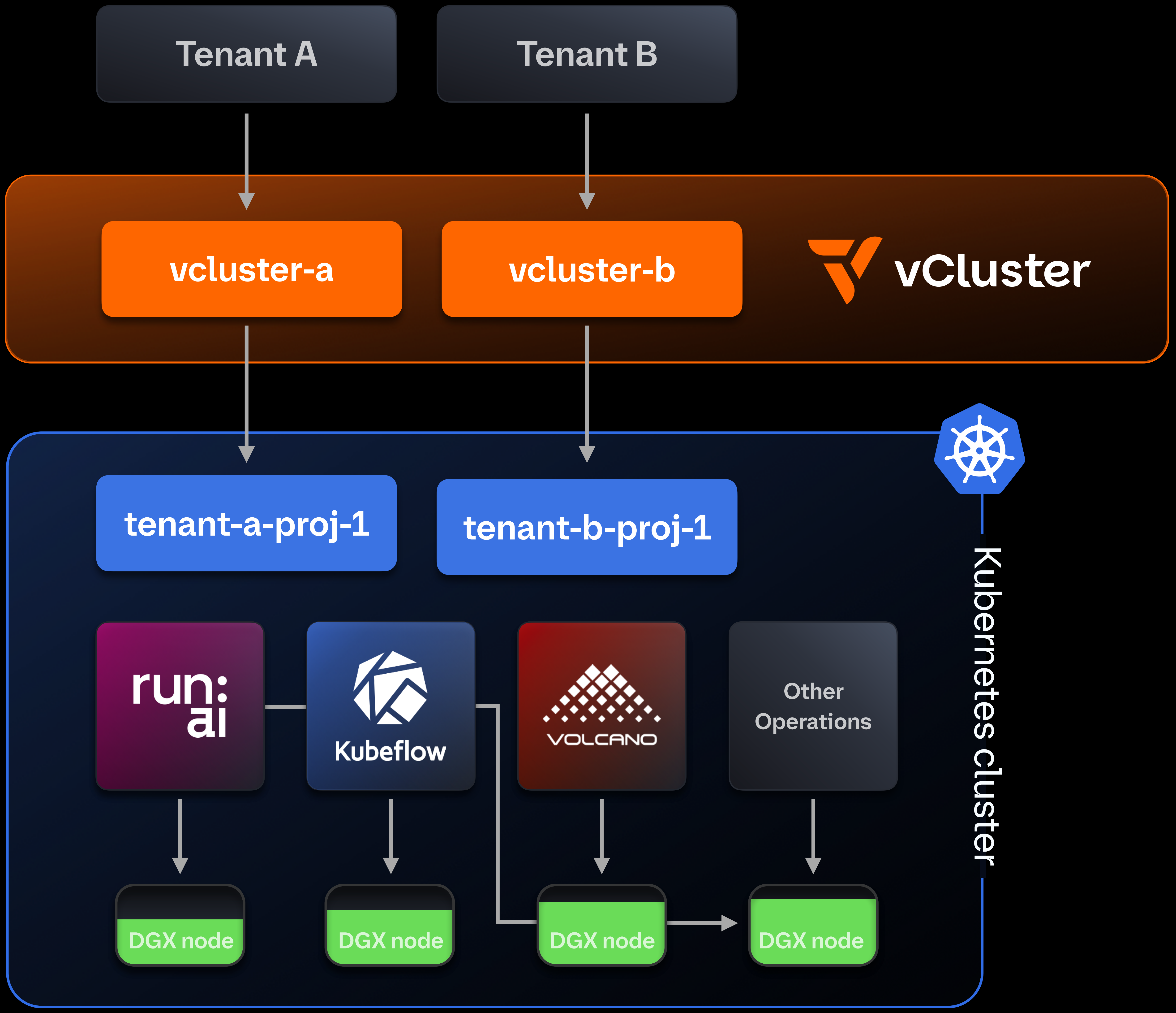
Multi-team clusters address efficiency, but not isolation.

02

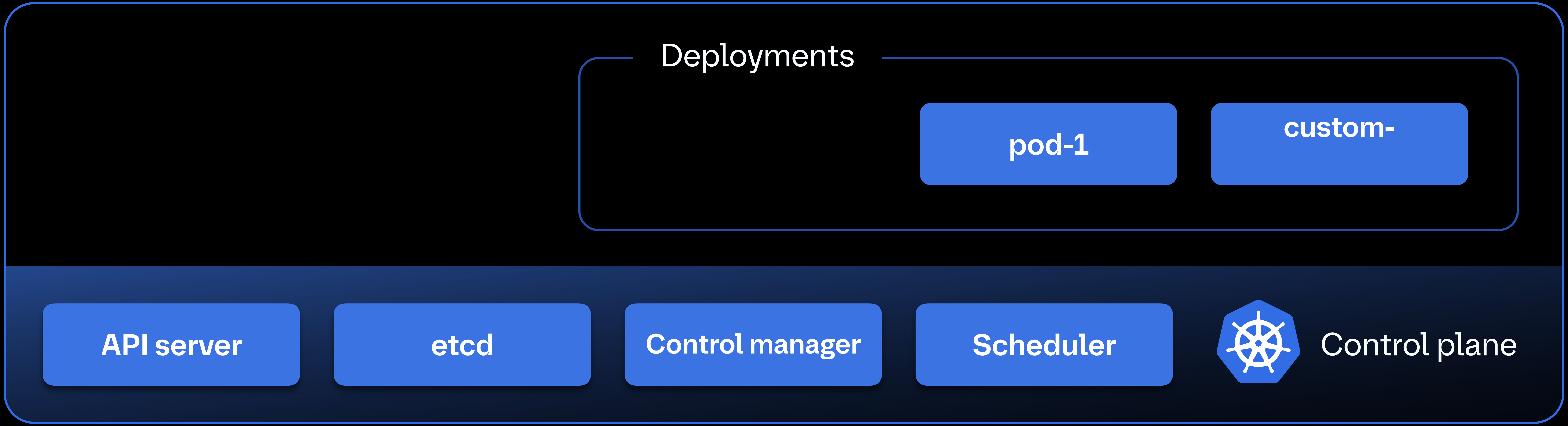
Where we need to be at

How do you deal with cluster-level resource?

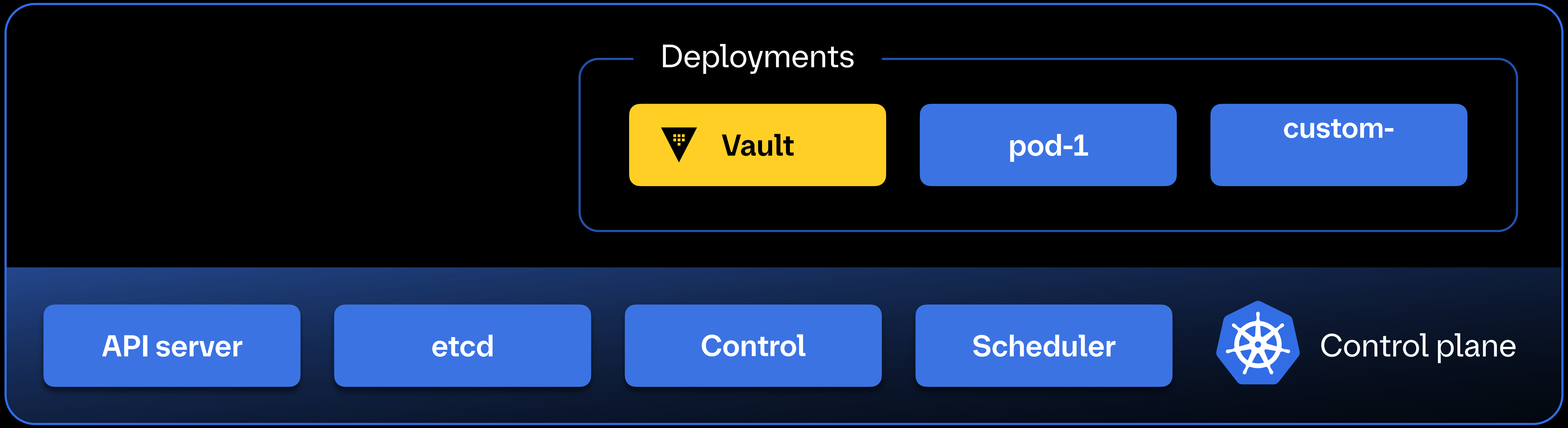
Multi-tenant cluster



Team-specific virtual cluster

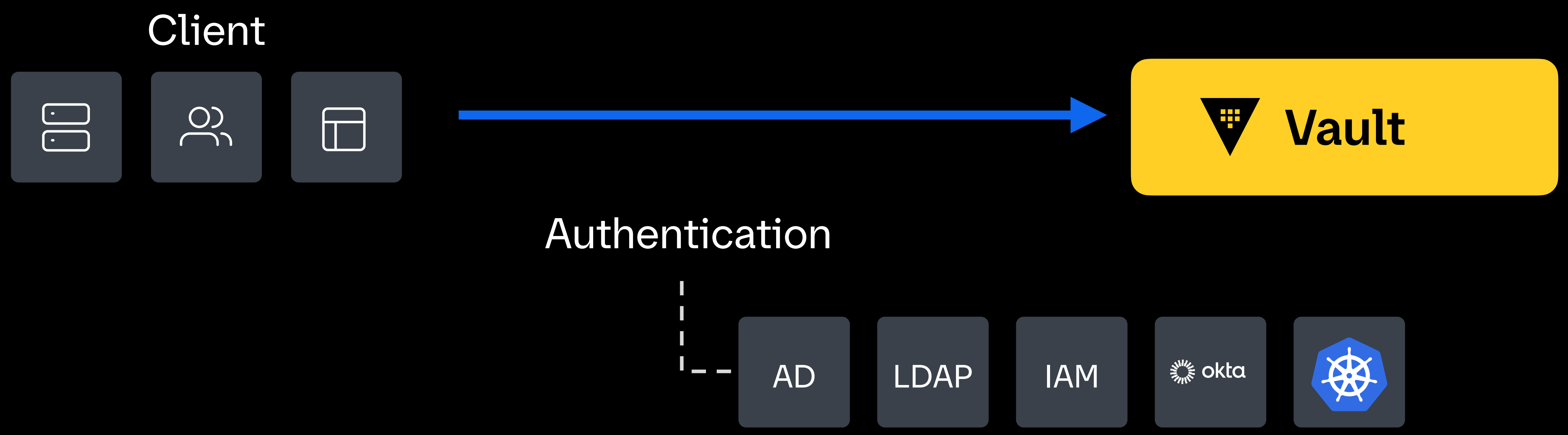


Team-specific virtual cluster

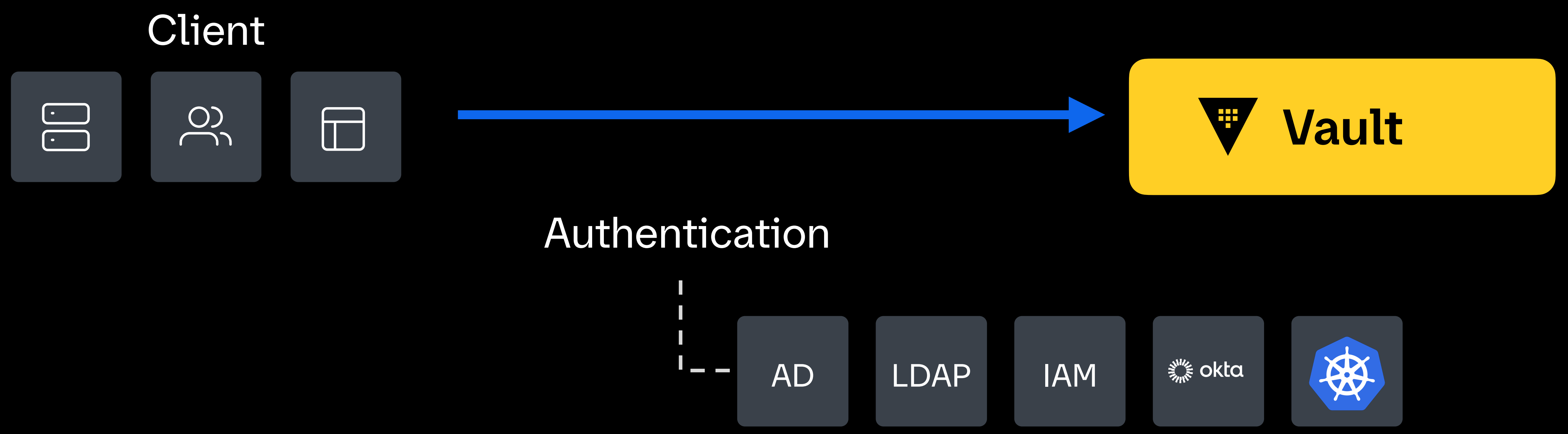


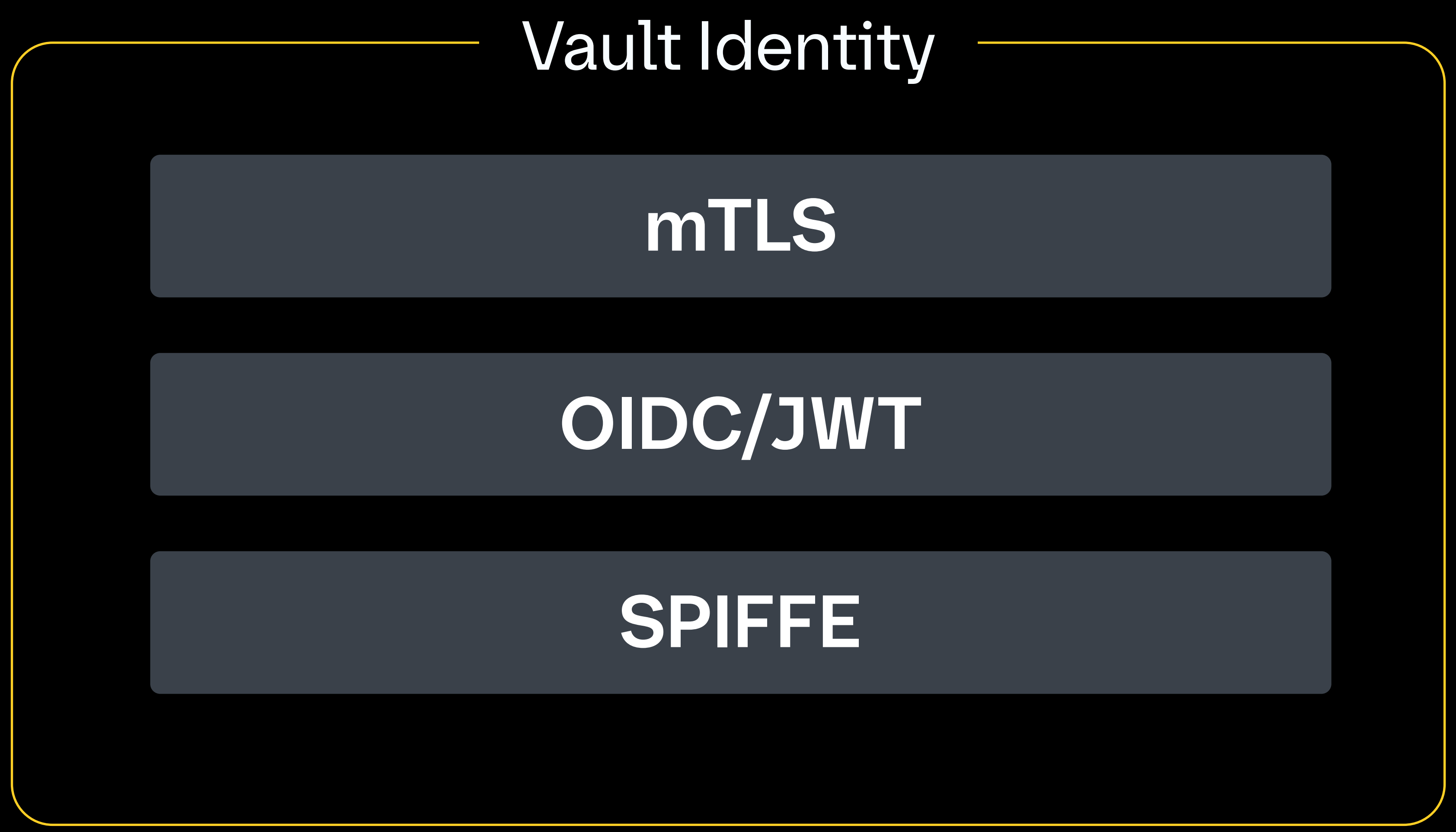
Security is important

Why Vault

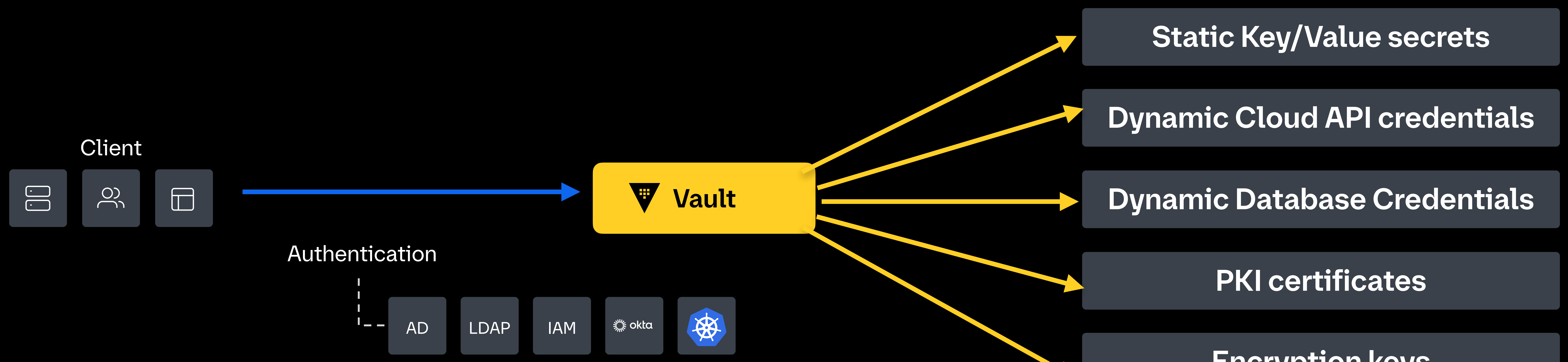


Why Vault


Client
Vault
Authentication
AD
LDAP
IAM
okta


Vault Identity
mTLS
OIDC/JWT
SPIFFE

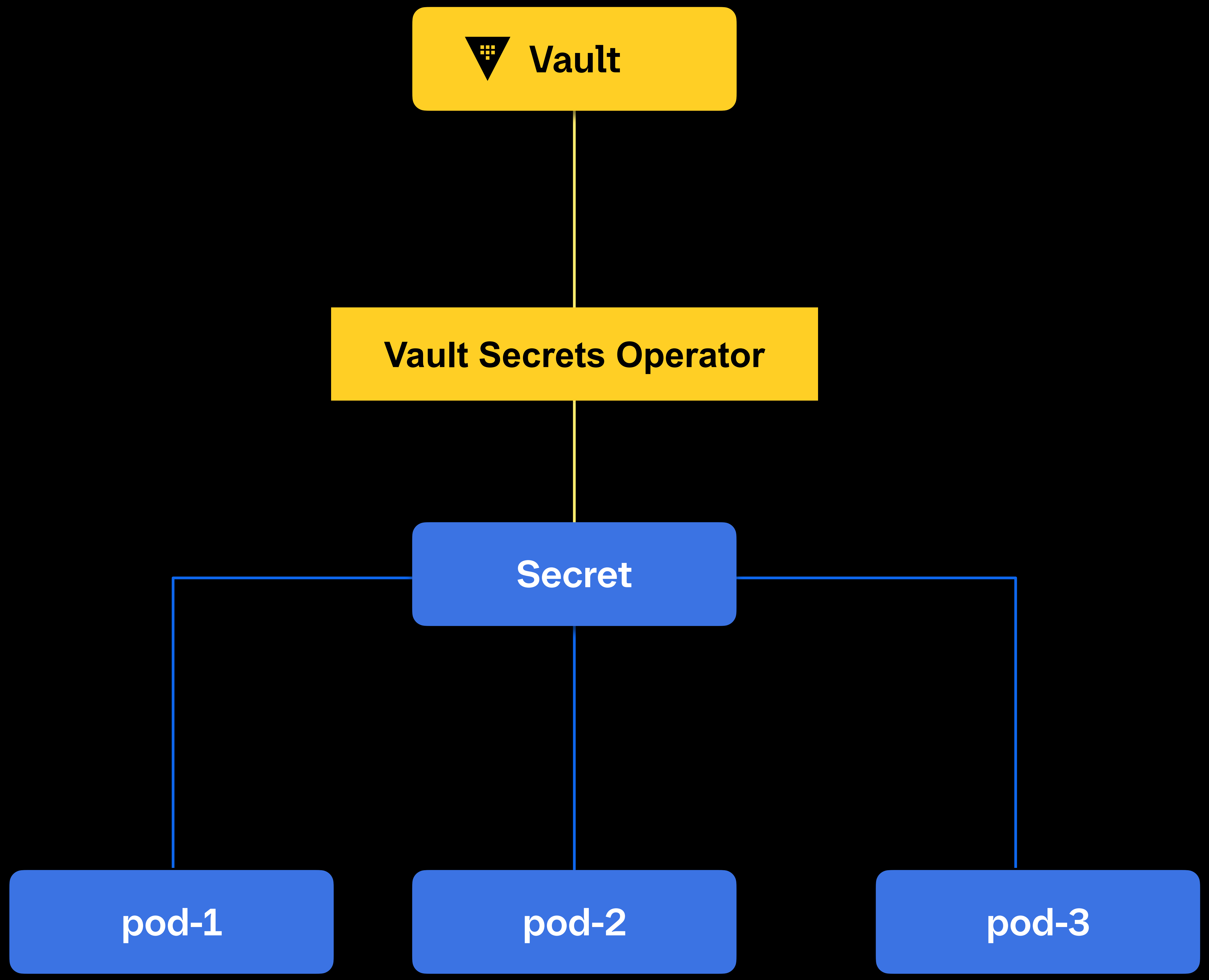
Why Vault


Client
Authentication
AD
LDAP
IAM
okta
Vault
Static Key/Value secrets
Dynamic Cloud API credentials
Dynamic Database Credentials
PKI certificates
Encryption keys

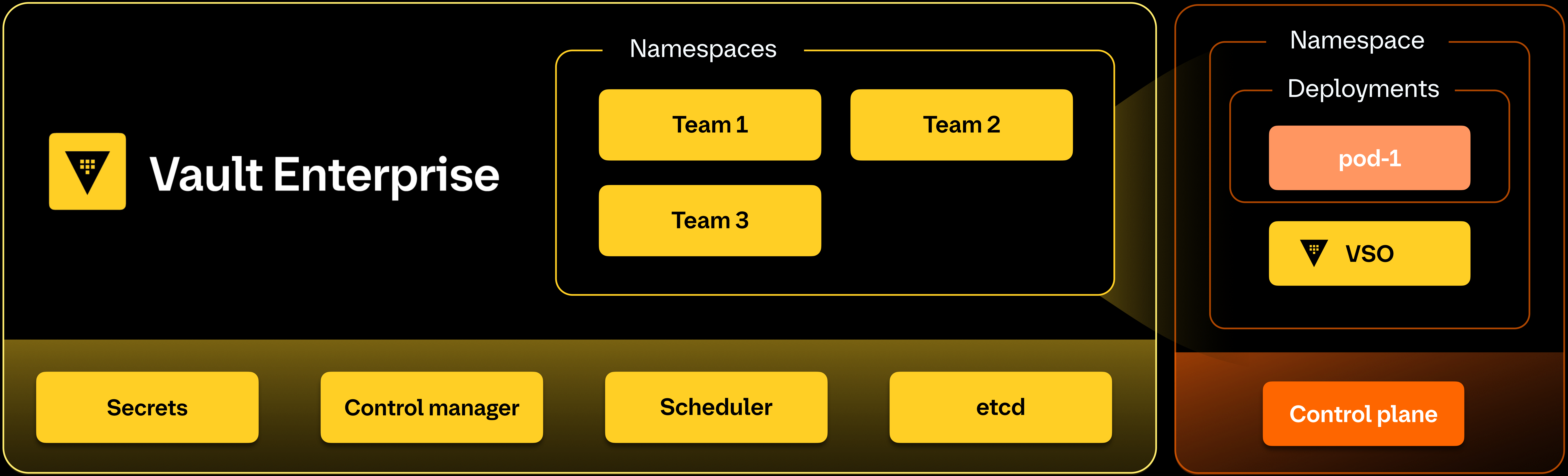
Eliminate forever credentials

Audit All The Things

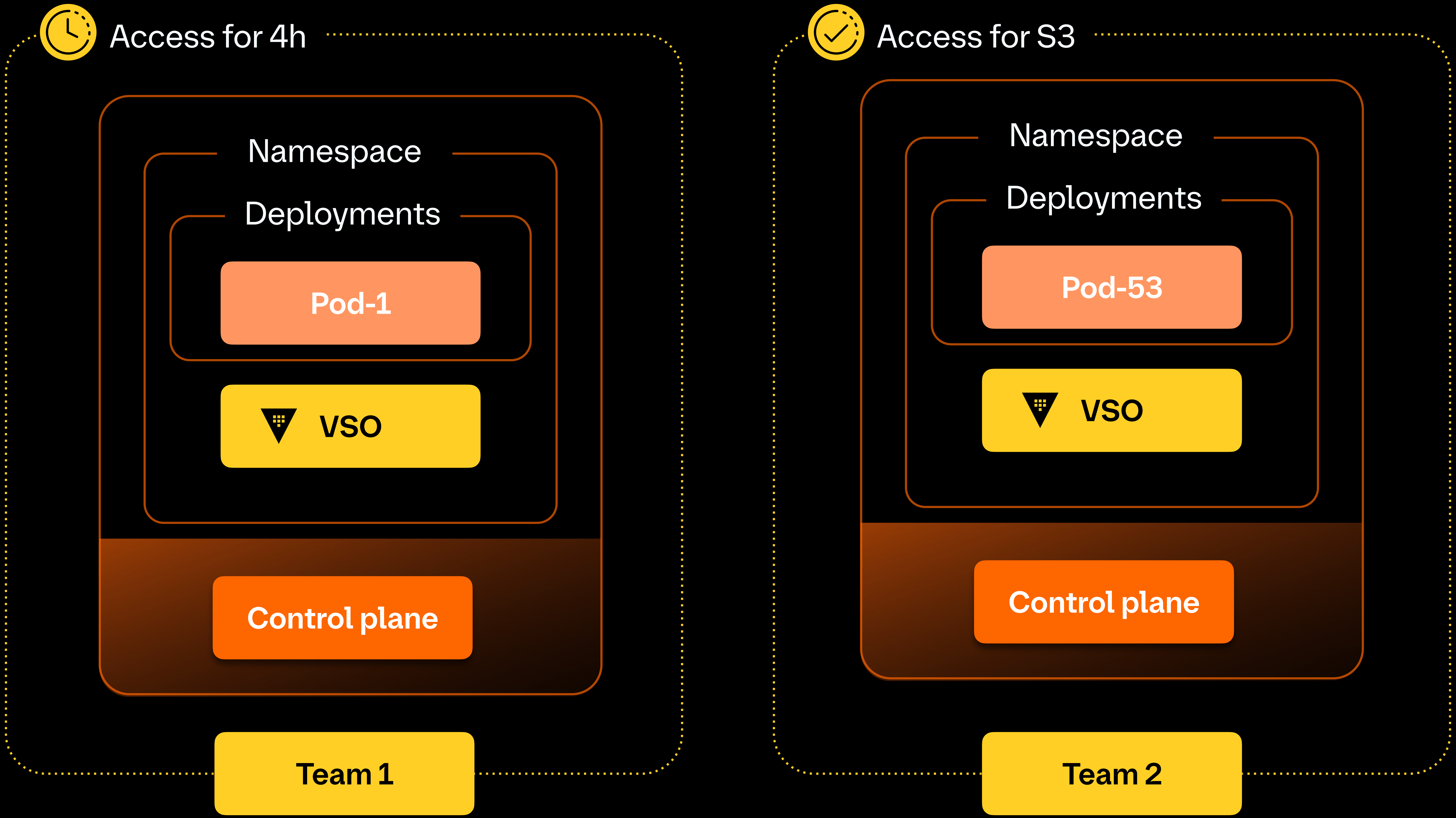
Vault and Kubernetes



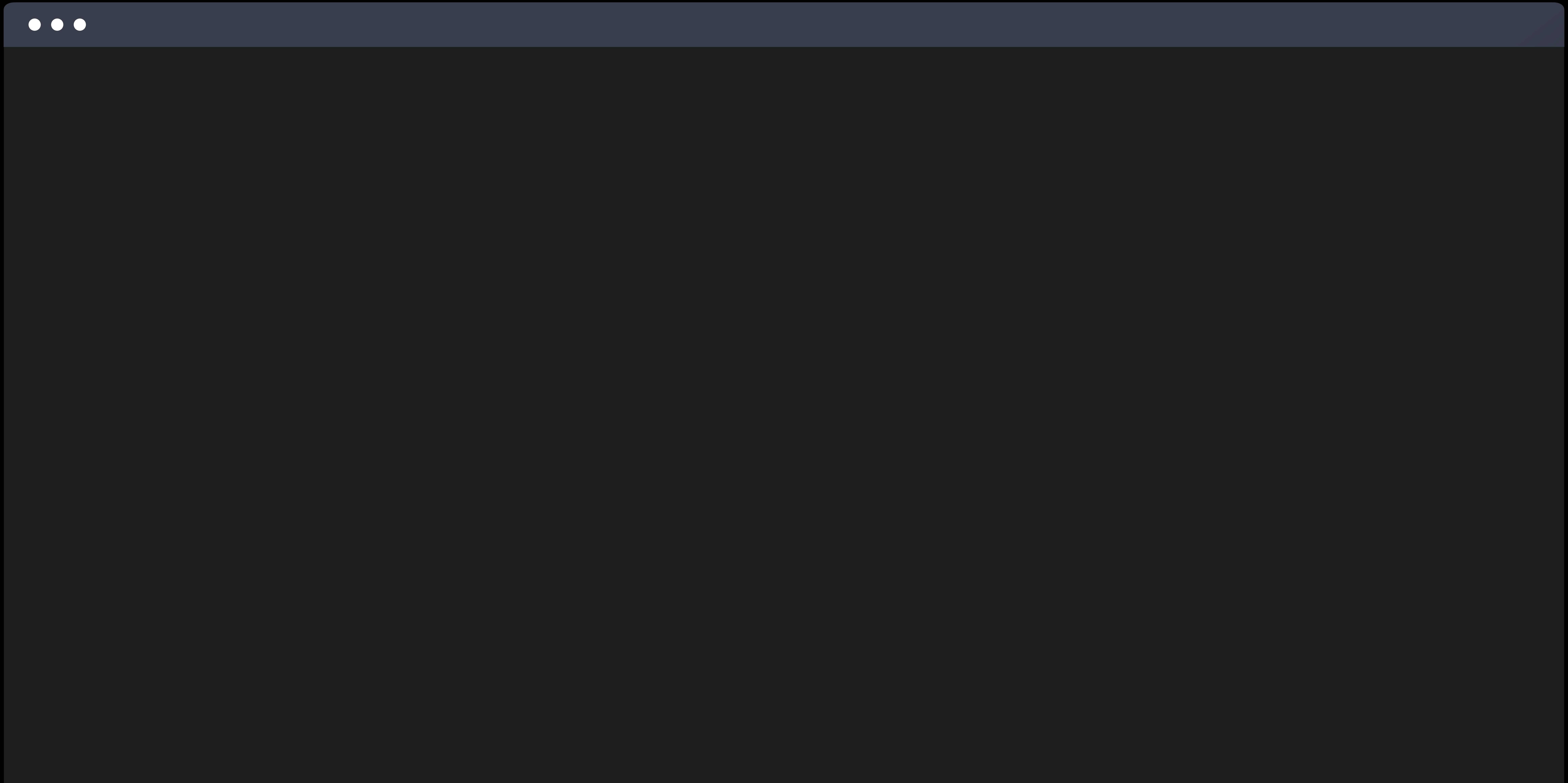
Team-specific security



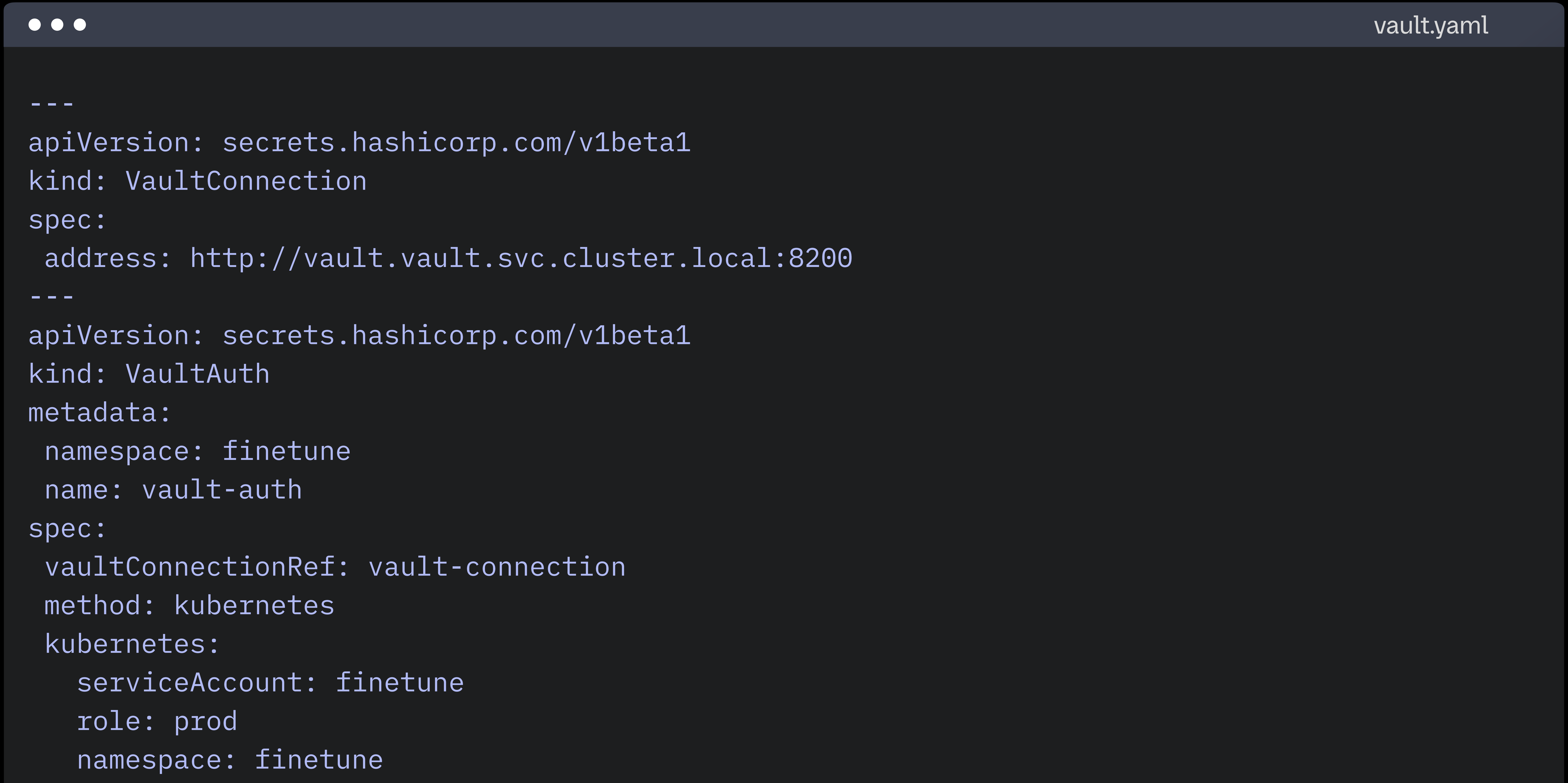
Team-shared cluster



Configuration



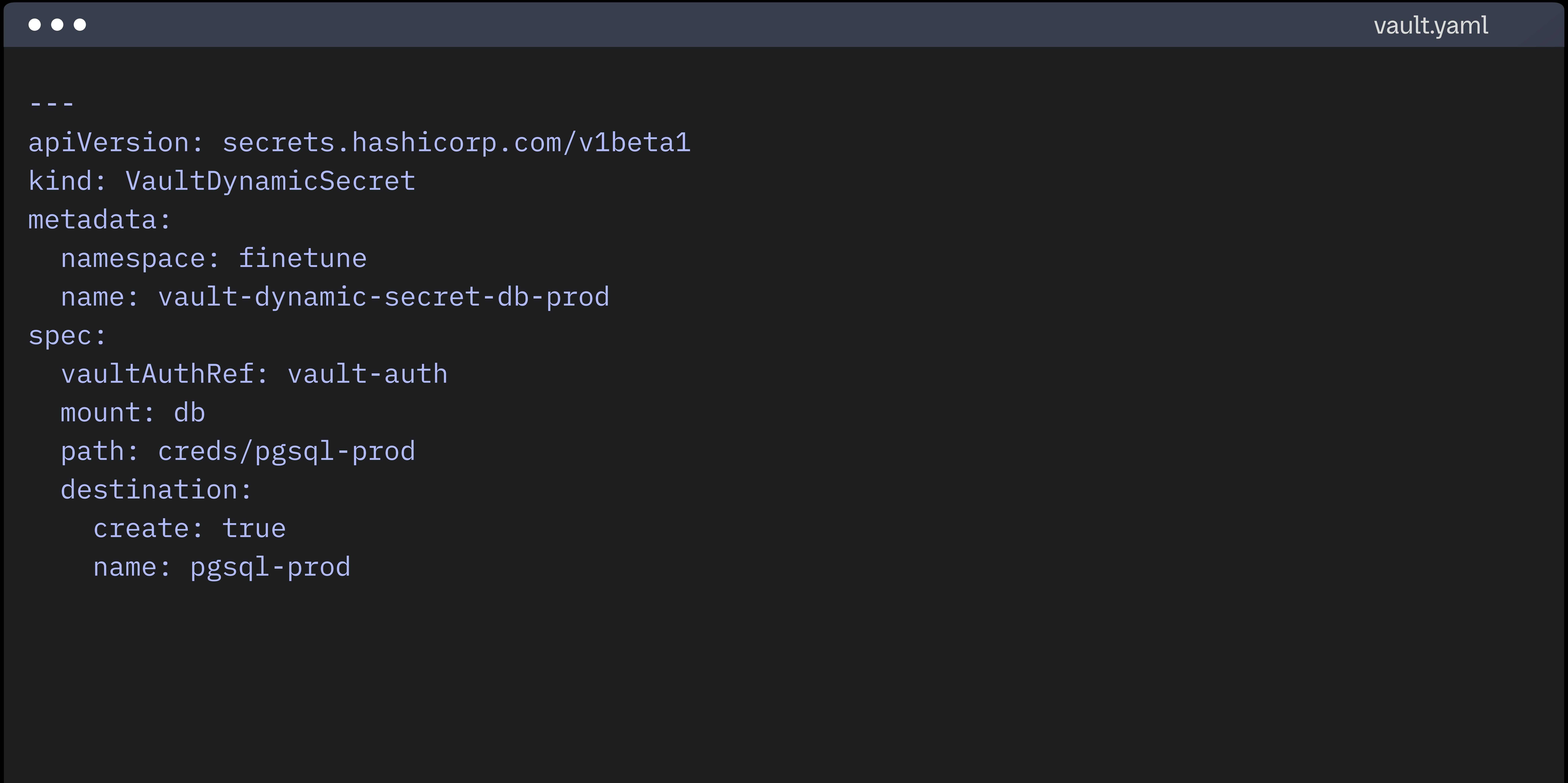
Kubernetes Configuration



vault.yaml

```yaml
---
apiVersion: secrets.hashicorp.com/v1beta1
kind: VaultConnection
spec:
 address: http://vault.vault.svc.cluster.local:8200
---
apiVersion: secrets.hashicorp.com/v1beta1
kind: VaultAuth
metadata:
 namespace: finetune
 name: vault-auth
spec:
 vaultConnectionRef: vault-connection
 method: kubernetes
 kubernetes:
   serviceAccount: finetune
   role: prod
   namespace: finetune
```

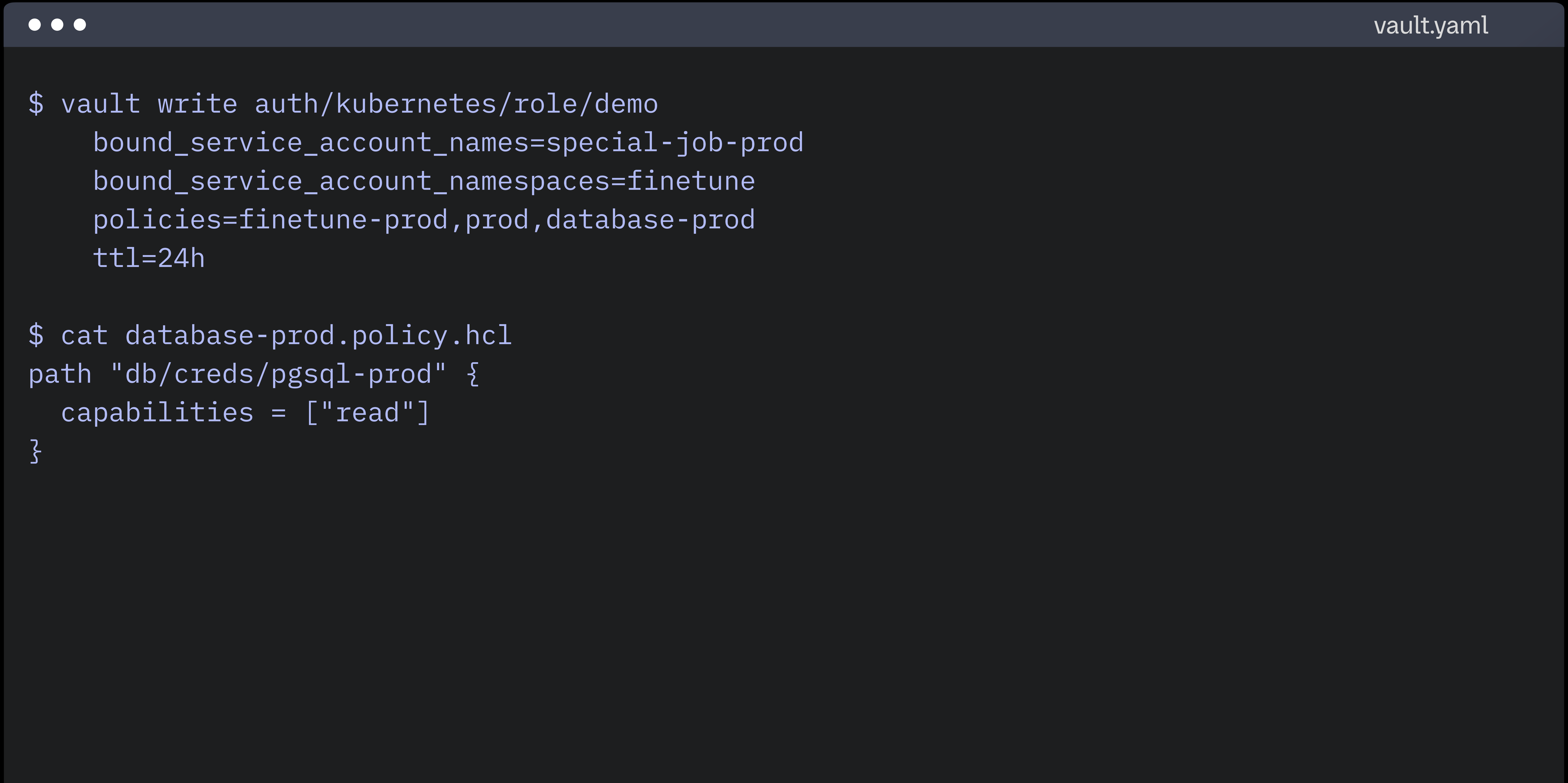
Kubernetes Configuration



vault.yaml

```yaml
---
apiVersion: secrets.hashicorp.com/v1beta1
kind: VaultDynamicSecret
metadata:
  namespace: finetune
  name: vault-dynamic-secret-db-prod
spec:
  vaultAuthRef: vault-auth
  mount: db
  path: creds/pgsql-prod
  destination:
    create: true
    name: pgsql-prod
```

Vault Configuration

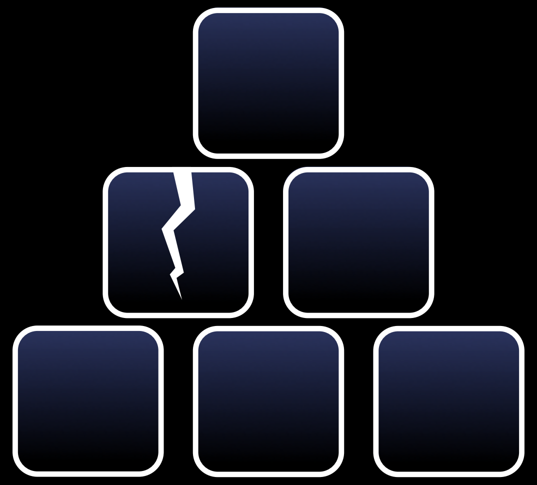


```
$ vault write auth/kubernetes/role/demo
    bound_service_account_names=special-job-prod
    bound_service_account_namespaces=finetune
    policies=finetune-prod,prod,database-prod
    ttl=24h

$ cat database-prod.policy.hcl
path "db/creds/pgsql-prod" {
  capabilities = ["read"]
}
```

03

What we need to remember



Architect for failure.

☑

Isolation is key.

Defense in-depth is defense that works.

Virtual clusters limit blast-radius.

Virtual clusters enable easier auditing.

Thank you

speakerdeck.com/stmcallister
atodorov.me