Platform Engineering Day EUROPE

Platform
Engineering Day
EUROPE

Building a Platform Engineering API Layer with kcp

Marvin Beckers, Kubermatic GmbH

Marvin Beckers

Team Lead at KUBERMATIC

github.com/embik

linkedin.com/in/marvinbeckers

hachyderm.io/@embik

Agenda

1. Kubernetes as an API Layer
2. Lightweight Clusters to the Rescue
3. What is kcp?
    a. Workspaces
4. The API Marketplace
    a. Create APIs with APIExports
    b. Enable APIs with APIBindings
    c. Implement APIs with Virtual Workspaces
5. Wrapping Up

Platform Engineering Day
EUROPE

Kubernetes as the API layer

for your platform

Kubernetes’ API design is awesome!

(that’s it. that’s the ~~tweet~~ slide)

Kubernetes as an API layer

But …

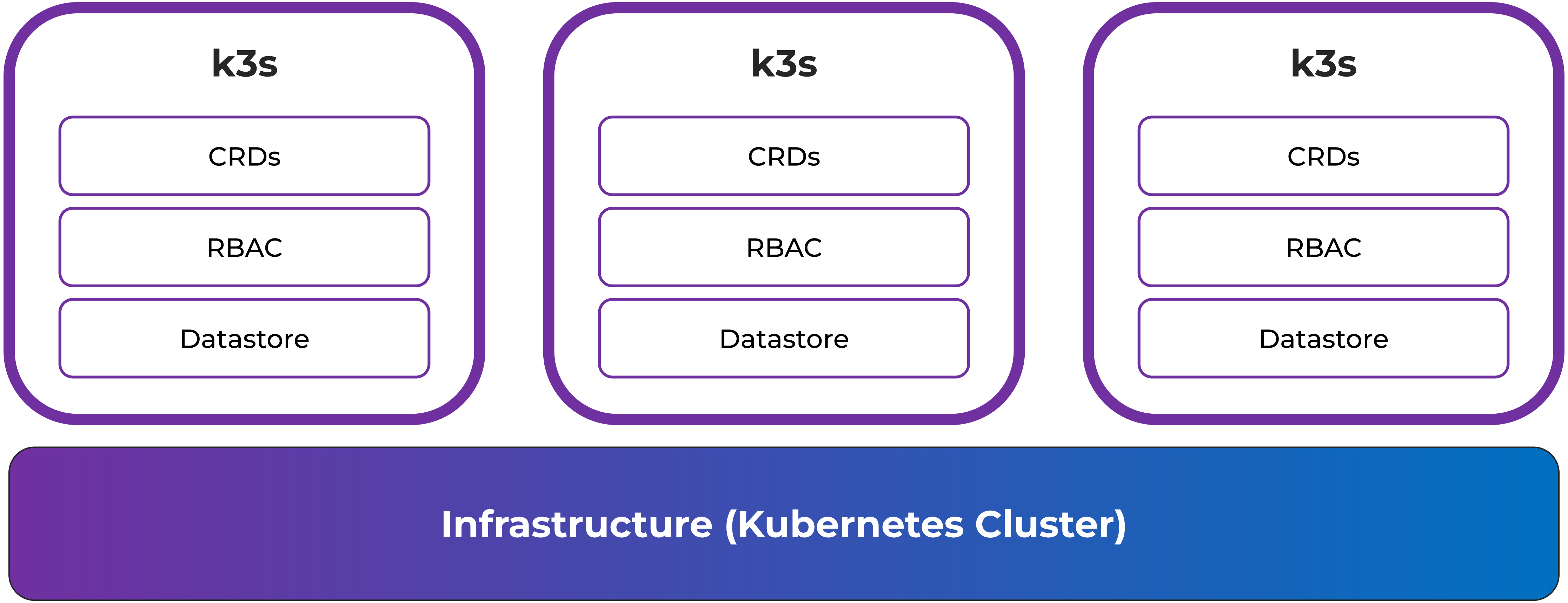
- CRDs are cluster-scoped, so everyone shares them.
- Scaling becomes a problem at some point.
- You don’t need the workload orchestration part for APIs.

Platform Engineering Day
EUROPE

Lightweight Clusters

to the rescue

Lightweight Clusters



k3s

CRDs

RBAC

Datastore

k3s

CRDs

RBAC

Datastore

k3s

CRDs

RBAC

Datastore

Infrastructure (Kubernetes Cluster)

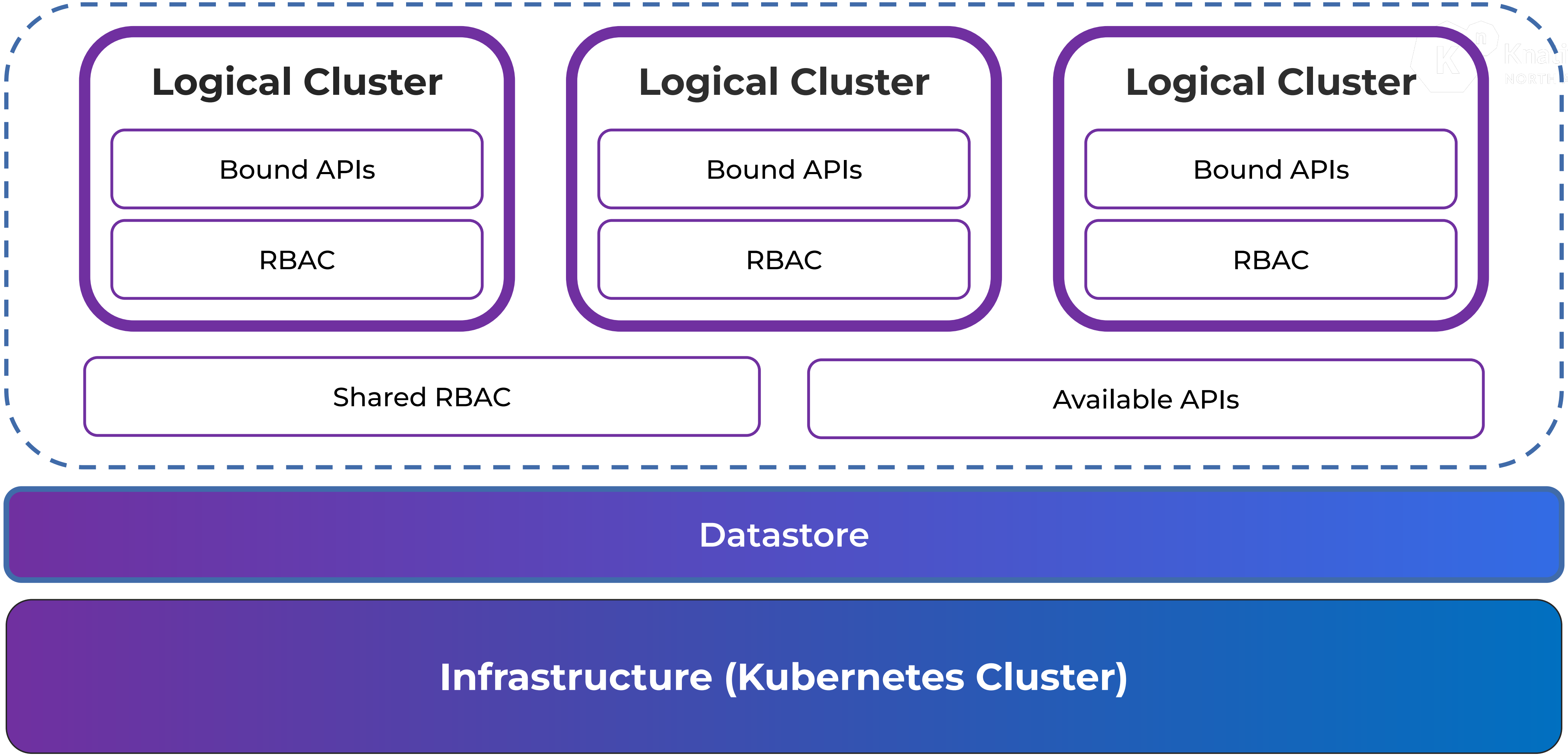
Lightweight Clusters

But what if …

Lightweight Clusters

we “virtualized” API servers?

Logical Clusters



What is kcp?

What is kcp?

A horizontally scalable control plane for Kubernetes-style APIs.

What is kcp?

CLOUD NATIVE
COMPUTING FOUNDATION

Sandbox project (since end of 2023)

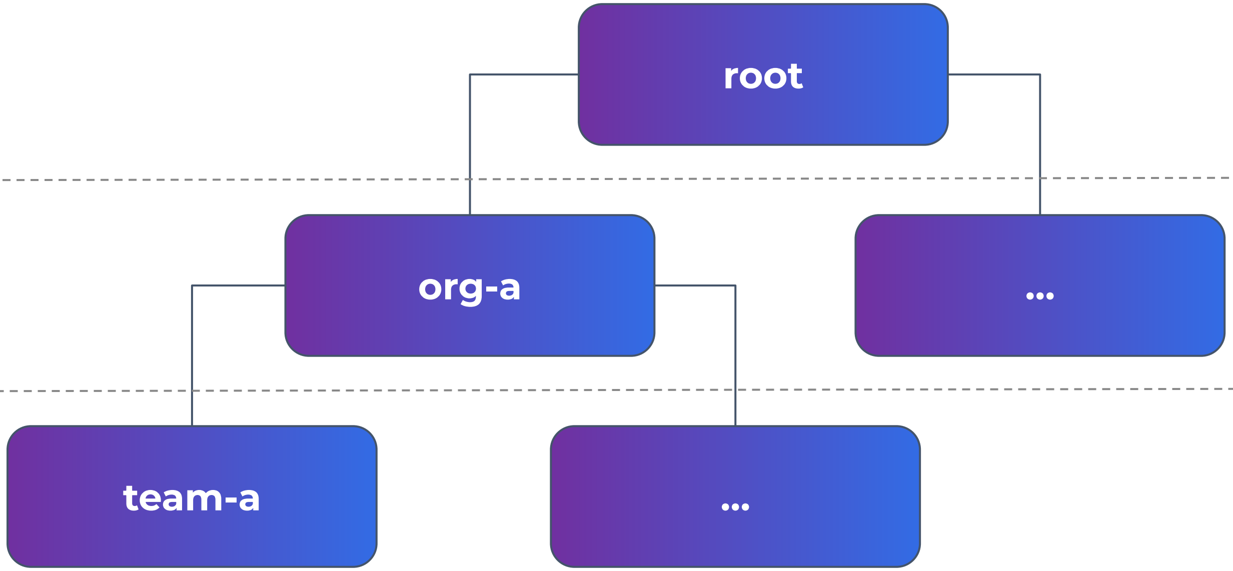
Workspaces

Workspaces implement the concept of logical clusters.

https://kcp:6443/clusters/root

https://kcp:6443/clusters/root:org-a

https://kcp:6443/clusters/root:org-a:team-a



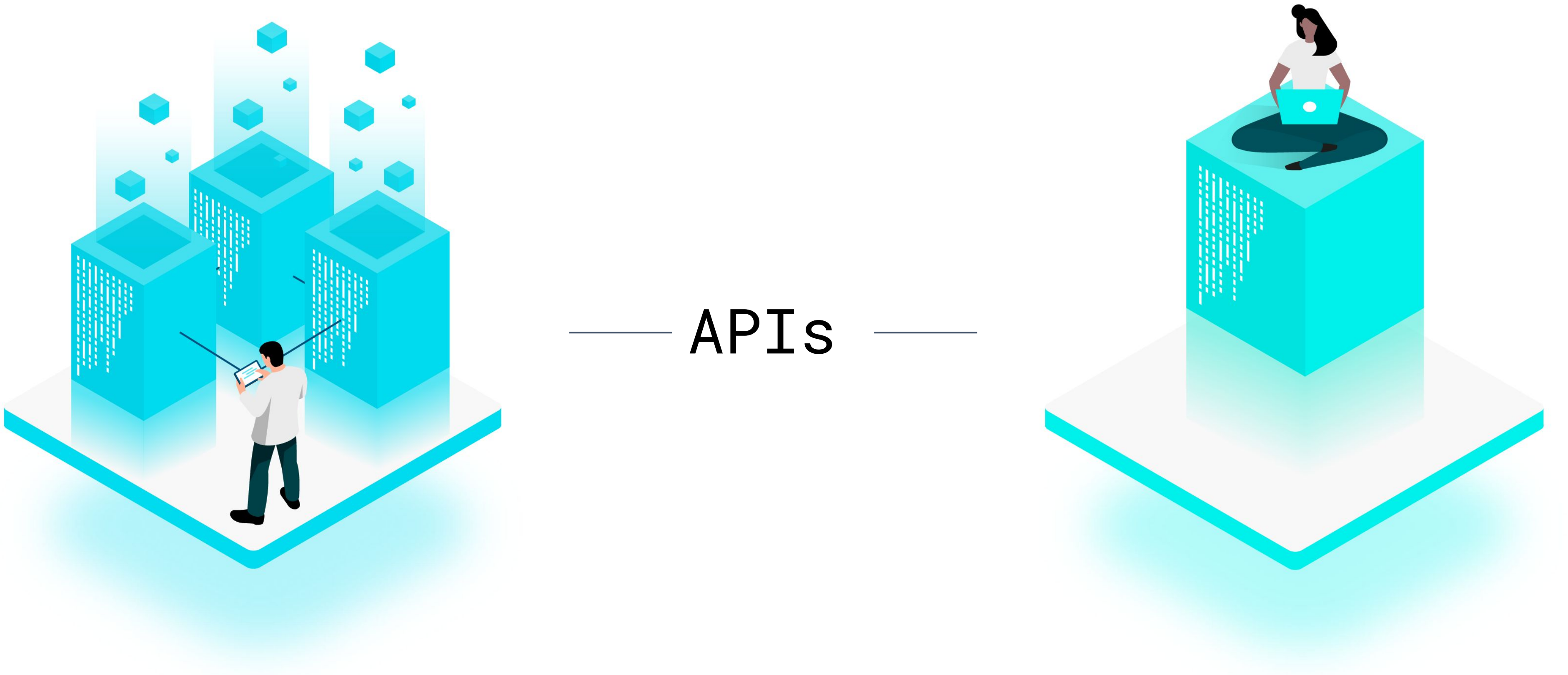
Workspaces

And you can navigate them!

```
$ kubectl ws .
Current workspace is "root".

$ kubectl get ws
NAME    TYPE           REGION   PHASE   URL          AGE
org-a   organization            Ready   https://…    69d
org-b   organization            Ready   https://…    65d

$ kubectl ws org-a
Current workspace is "root:org-a" (type root:organization).

$ kubectl get ws
NAME     TYPE   REGION   PHASE   URL         AGE
team-a   team            Ready   https://…   3m23s
team-b   team            Ready   https://…   3m18s


$ kubectl ws team-a
Current workspace is "root:org-a:team-a" (type root:team).
```

The API Marketplace

The API Marketplace



APIs

Infrastructure + Service Providers

Developers + Other Users

The API Marketplace

Infrastructure + Service teams are not in the business of making APIs discoverable and consumable.

Platform teams are.

Platform
Engineering Day
EUROPE

Create APIs

with APIExports

APIResourceSchema

```
apiVersion: apis.kcp.io/v1alpha1
kind: APIResourceSchema
  name: v1.certificates.demo.embik.me
spec:
  group: certificates.demo.embik.me
  names:
    kind: Certificate
    listKind: CertificateList
    plural: certificates
    singular: certificate
  scope: Namespaced
  versions:
    [...]
```

Hey, this is a CRD!

APIResourceSchema

```
$ kubectl kcp crd snapshot --filename=crd.yaml --prefix=vc517860e
```

– or –

```
kcpv1alpha1.CRDToAPIResourceSchema(crd, prefix)
```

APIExport

```
apiVersion: apis.kcp.io/v1alpha1
kind: APIExport
metadata:
  name: demo.embik.me
spec:
  latestResourceSchemas:
    - v1.certificates.demo.embik.me
    - v1.pizzas.demo.embik.me
```

Enable APIs

with APIBindings

APIs in Workspaces

```
https://kcp:6443/clusters/root:org-a/api
```

```
$ kubectl api-resources
NAME                   SHORTNAMES    APIVERSION                 NAMESPACED   KIND
configmaps             cm            v1                         true         ConfigMap
events                 ev            v1                         true         Event
namespaces             ns            v1                         false        Namespace
resourcequotas         quota         v1                         true         ResourceQuota
secrets                              v1                         true         Secret
serviceaccounts        sa            v1                         true         ServiceAccount
[...]
workspaces             ws            tenancy.kcp.io/v1alpha1    false        Workspace
workspacetypes                       tenancy.kcp.io/v1alpha1    false        WorkspaceType
[...]
```

Binding APIExports

```
$ kubectl get apibindings
NAME                    AGE   READY
tenancy.kcp.io-3wb5h    30d   True
topology.kcp.io-cua3o   30d   True
```

```yaml
apiVersion: apis.kcp.io/v1alpha1
kind: APIBinding
metadata:
  name: tenancy.kcp.io-3wb5h
spec:
  reference:
    export:
      name: tenancy.kcp.io
      path: root
```

This references an APIExport in a different workspace!

Binding APIExports

Binding to exported APIs requires RBAC permissions on the **`APIExport`**

(in the APIExport workspace)

```
apiVersion:
rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
   name: bind-apiexport
rules:
- apiGroups:
  - apis.kcp.io
  resources:
  - apiexports
  verbs:
  - use
  resourceNames:
  - demo.embik.me
```

Binding APIExports

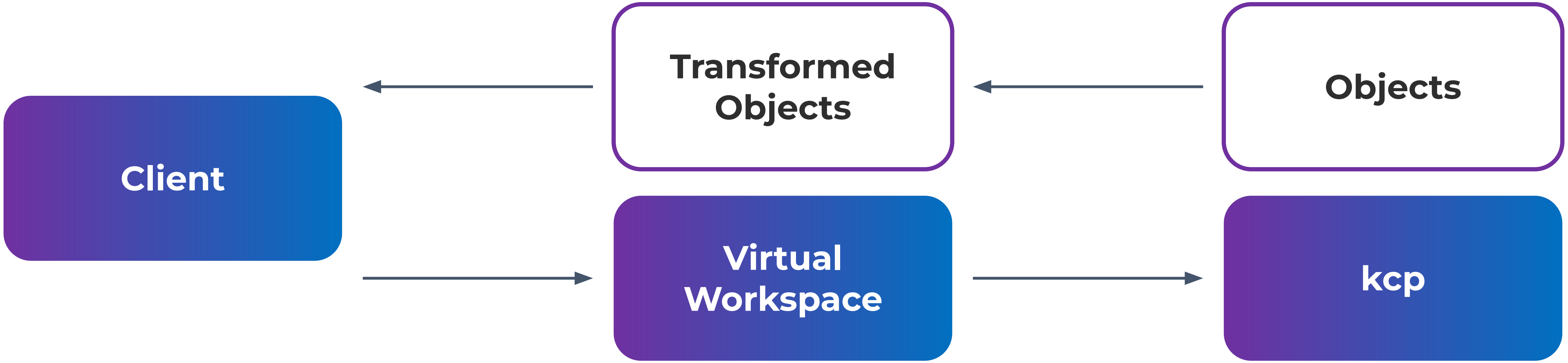
```
$ kubectl api-resources
NAME                 SHORTNAMES   APIVERSION               NAMESPACED   KIND
configmaps           cm           v1                       true         ConfigMap
events               ev           v1                       true         Event
namespaces           ns           v1                       false        Namespace
resourcequotas       quota        v1                       true         ResourceQuota
secrets                           v1                       true         Secret
serviceaccounts      sa           v1                       true         ServiceAccount
[...]
workspaces           ws           tenancy.kcp.io/v1alpha1  false        Workspace
workspacetypes                    tenancy.kcp.io/v1alpha1  false        WorkspaceType
[...]
certificates         certs        demo.embik.me/v1         true         Certificate
pizzas                            demo.embik.me/v1         false        Pizza
```

Implement APIs

with virtual workspaces

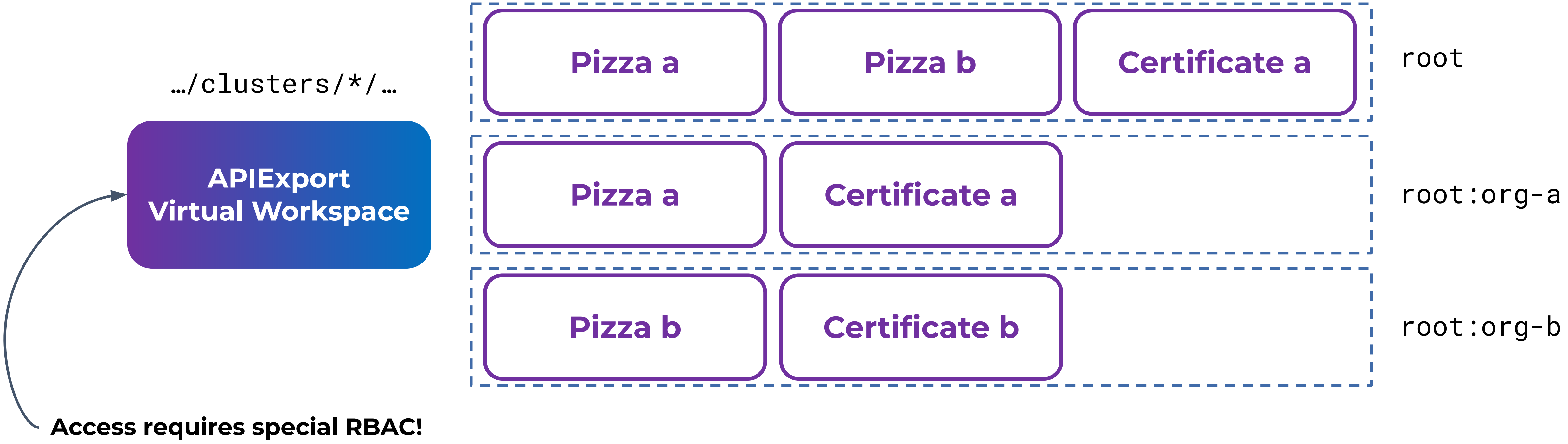
Virtual Workspaces

Proxies that provide computed views of the data in kcp.



APIExport Virtual Workspace

Provides a view on all resources created from an APIExport.



Access requires special RBAC!

Building kcp-aware controllers

1 Use kcp-aware client and cache

```
MapperProvider: kcp.NewClusterAwareMapperProvider,
NewClient:      kcp.NewClusterAwareClient,
NewCache:       kcp.NewClusterAwareCache,
NewAPIReader:   kcp.NewClusterAwareAPIReader,
```

2 Reconcile in VW via `Cluster`

```
sigs.k8s.io/controller-runtime/pkg/cluster.Cluster
```

3 Reconcile with logical cluster in context

```
ctx = kontext.WithCluster(ctx, logicalcluster.Name(request.ClusterName))
```

github.com/kcp-dev/controller-runtime

Platform Engineering Day EUROPE

Let’s wrap up

Let’s wrap up

1. kcp is building a global control plane for API-driven platforms.
2. Workspaces allow to reconstruct organizational hierarchy.
3. Providing Kubernetes-style APIs at scale is incredibly easy.

It’s a community project! We welcome everyone to build the project’s future together.

Thank You!

Catch me at the **Kubermatic** booth – **L11, Hall 7.2**.

Session Feedback

#kcp-dev

on Kubernetes Slack