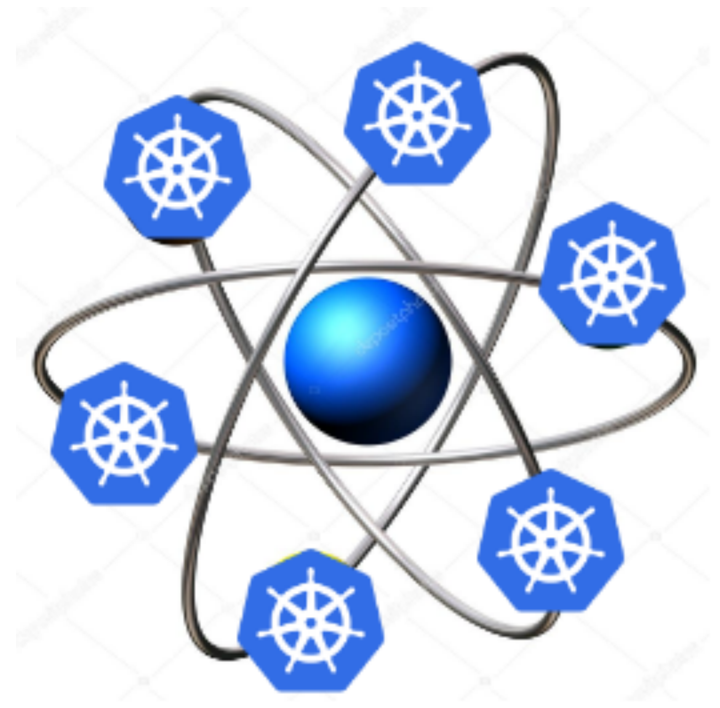
Managing RBAC

Cross Multiple Kubernetes Clusters



Alena Prokharchyk, Engineering manager

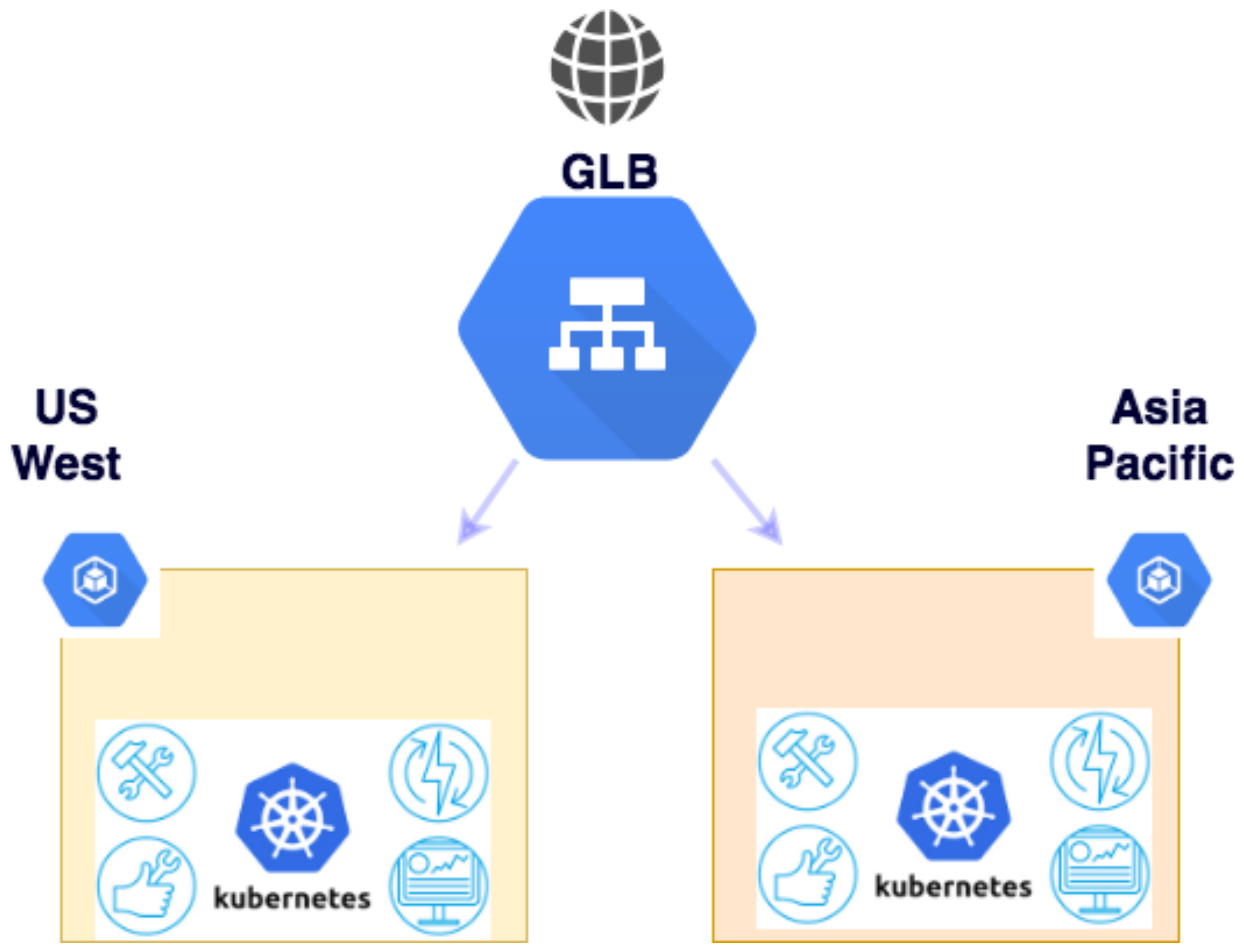
@RancherLabs

Kubernetes has become a commodity across public and private cloud ecosystem

Having multiple Kubernetes clusters
is a new de facto

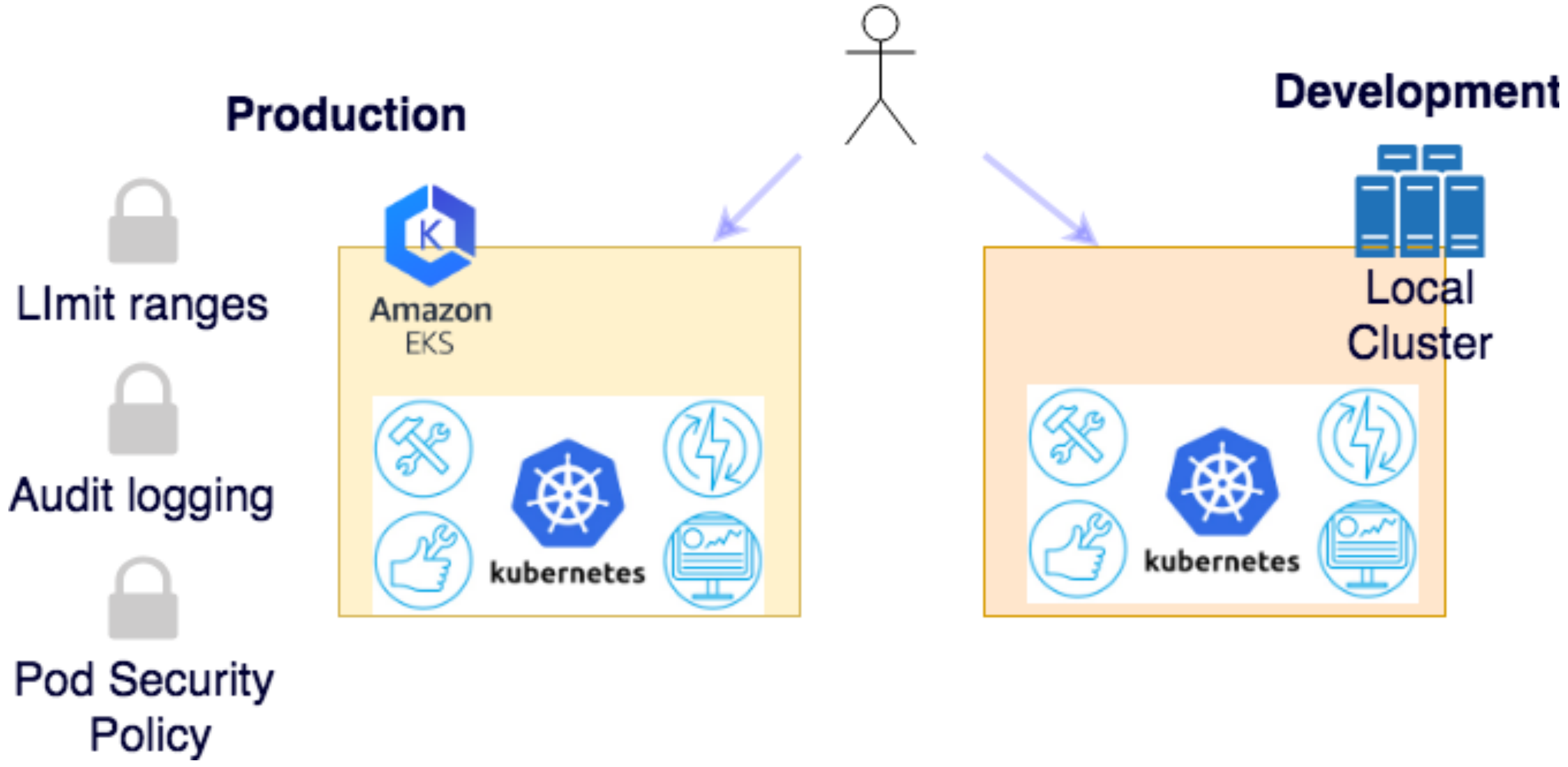
Usecase #1 - Geographical separation

- Cluster per region
- Front faced by GEO LB



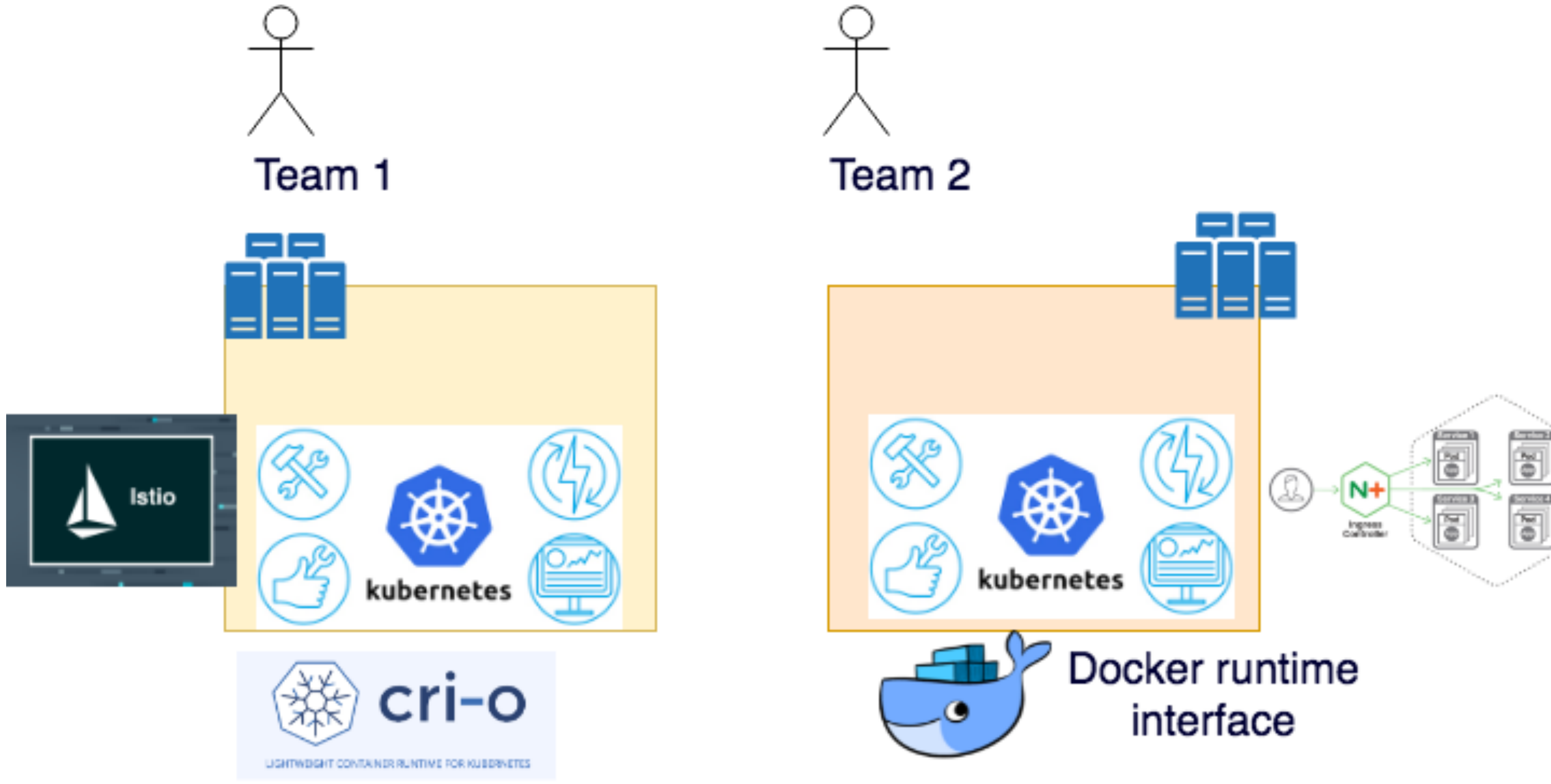
Usecase #2 - Logical Separation driven by security reasons

- Cluster per project
- Different level of protection



Usecase #3 - Logical separation driven by functionality reasons

- Cluster per team
- Different teams = different best practices



Kubernetes cloud types

Homogeneous

Heterogeneous

Challenges

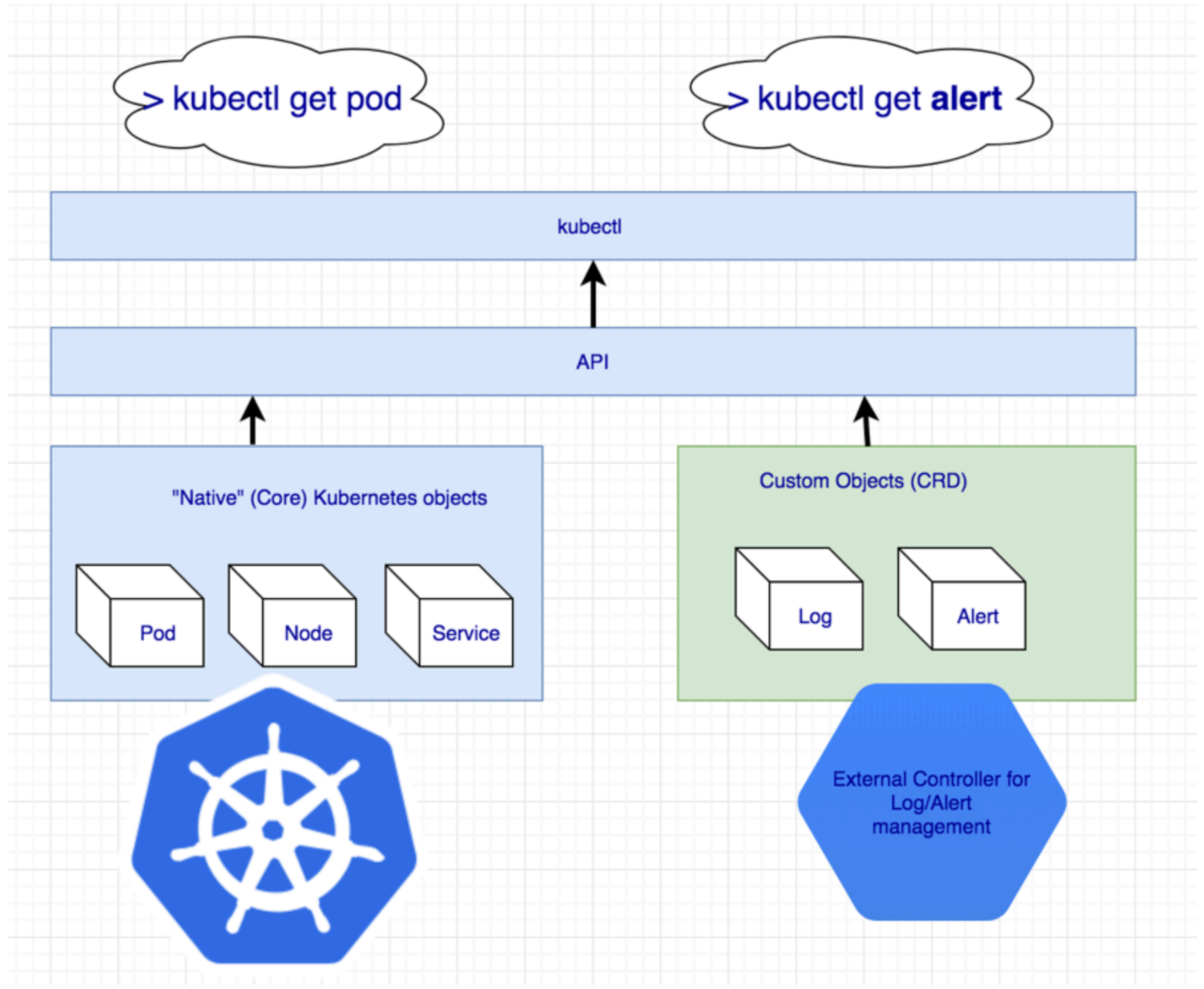
- Different hosted Kubernetes provider - different authentication type
- Authentication strategy on a hosted provider can’t be changed
- Configuring role based access rules for the same user cross clusters is a Herculean task

Our goal was to build an authenticaion and authorization management system, that is:

- Open source
- Written and developed as a Kubernetes native application
- Extends Kubernetes APIs using CRDs
- Logic is implemented as custom controllers

What is CRD?

A way to extend
Kubernetes API server



Custom Controller

- Watches for the resource changes
- Executes custom logic based on the resource spec or status
- Updates the resource status with the result
- There can be multiple controllers updating the same object

Kubernetes Native App

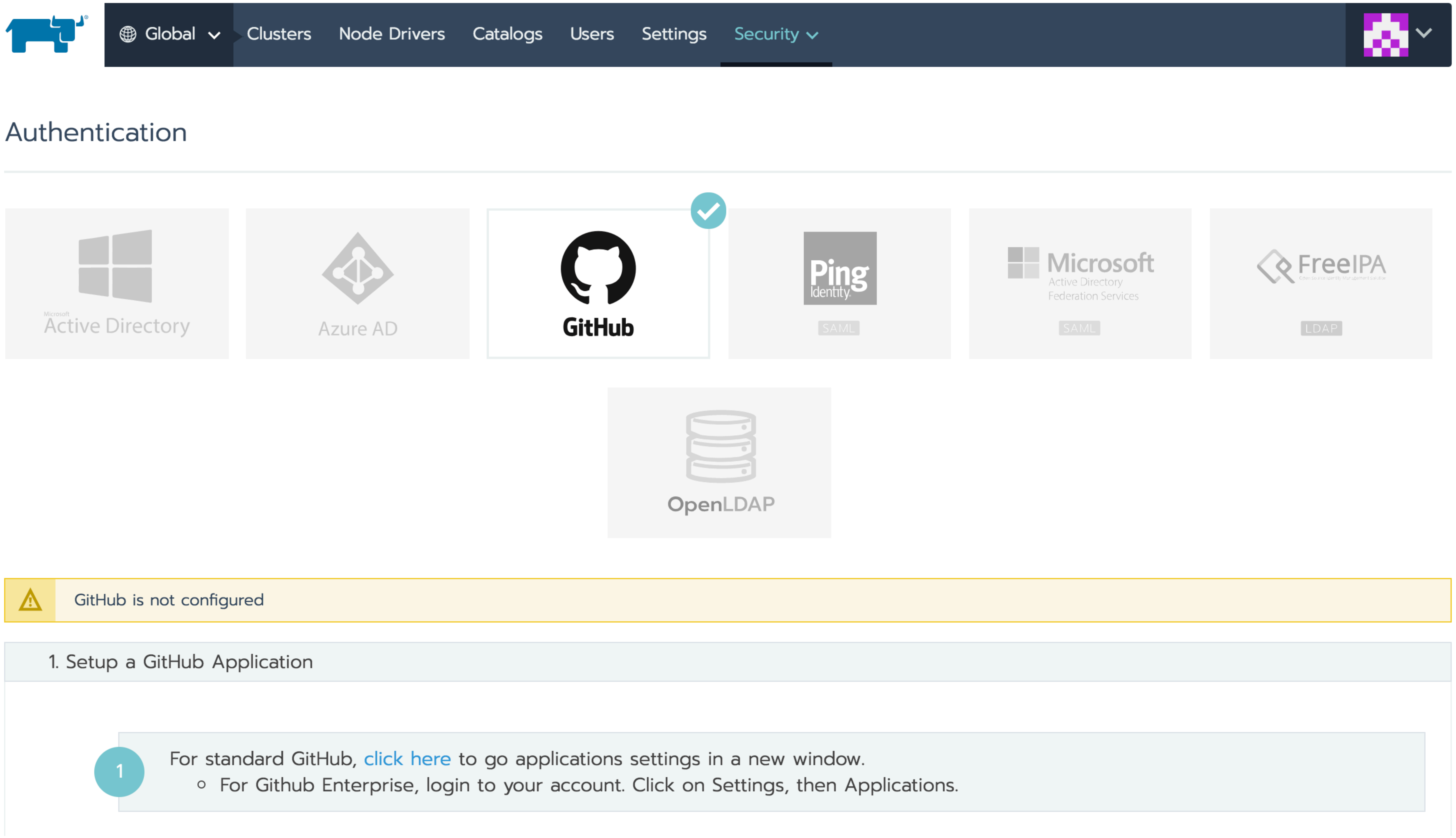
- Runs in Kubernetes pod
- Deployed using Kubernetes yaml manifest
- Utilizes Kubernetes constructs like ConfigMaps, Secrets
- Managed via Kubernetes APIs

RANCHER

https://github.com/rancher/rancher

Cross Clusters Authentication

One time configuration



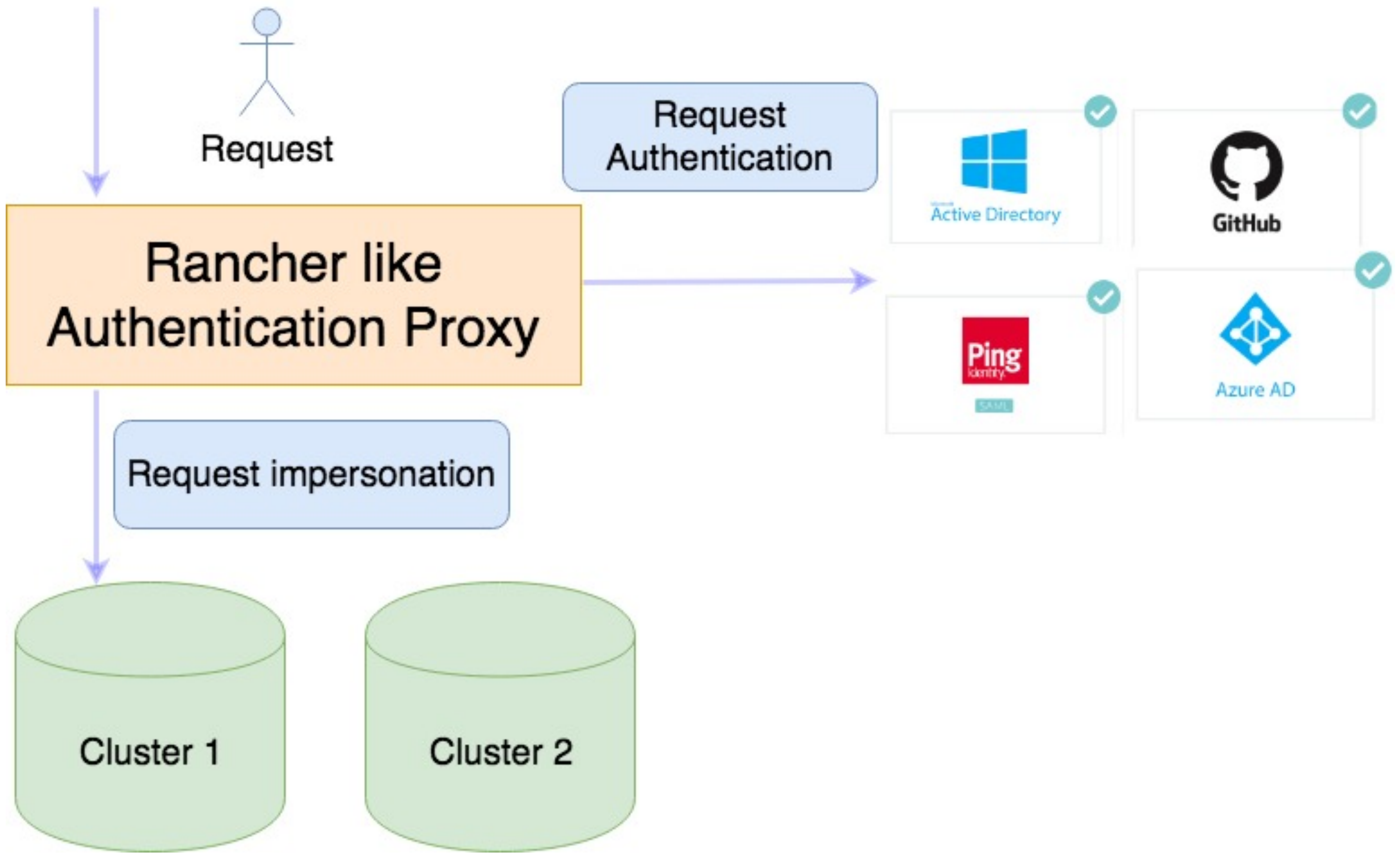
Authentication

GitHub is not configured

1. Setup a GitHub Application

1 For standard GitHub, click here to go applications settings in a new window.
- For Github Enterprise, login to your account. Click on Settings, then Applications.

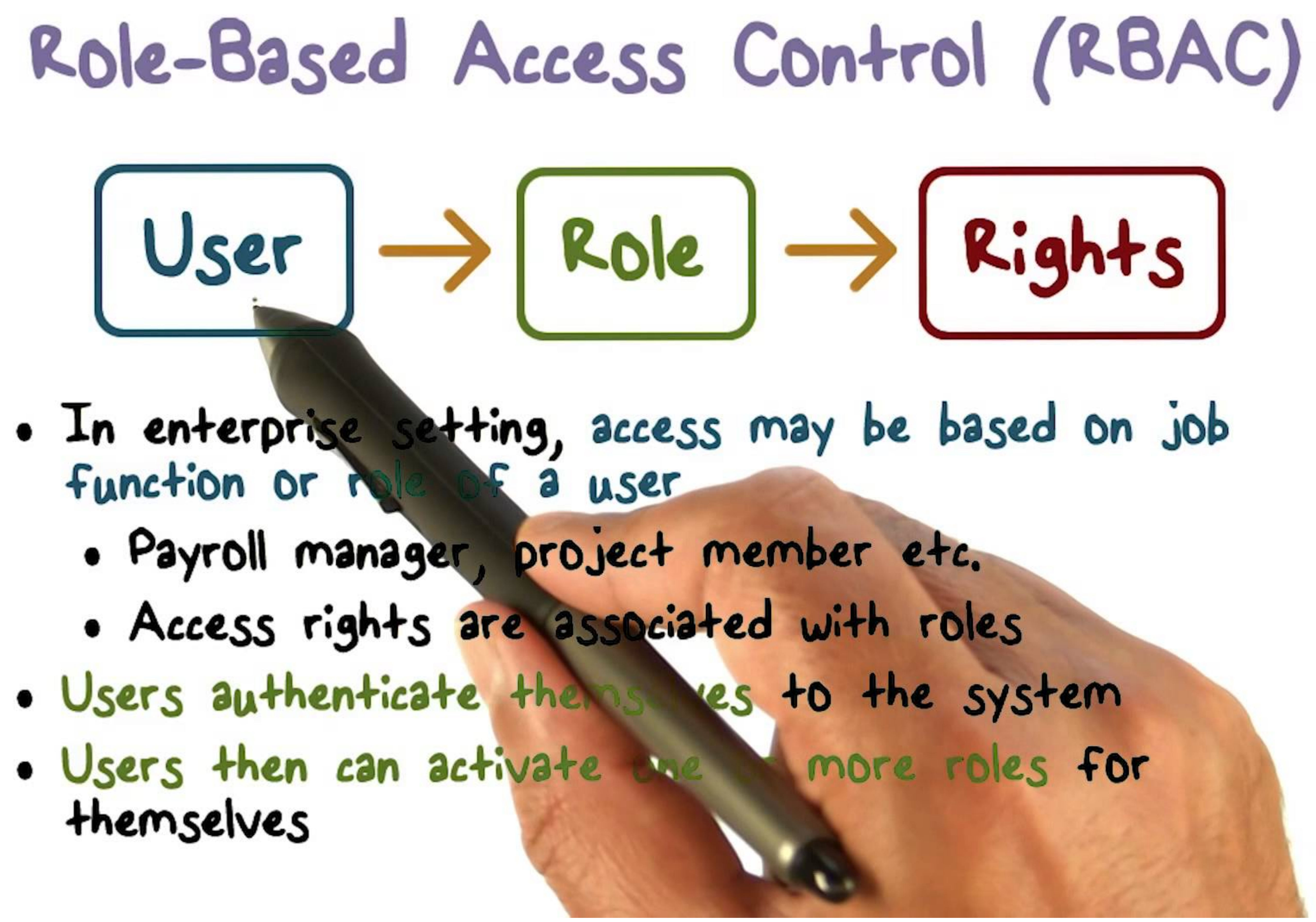
Centralized authentication



Implementation details

- User and Group are first class objects represented by CRDs
- Admin can grant permissions on per user/group to a particular cluster
- Kubernetes token based authentication is being leveraged when authenticate to a cluster

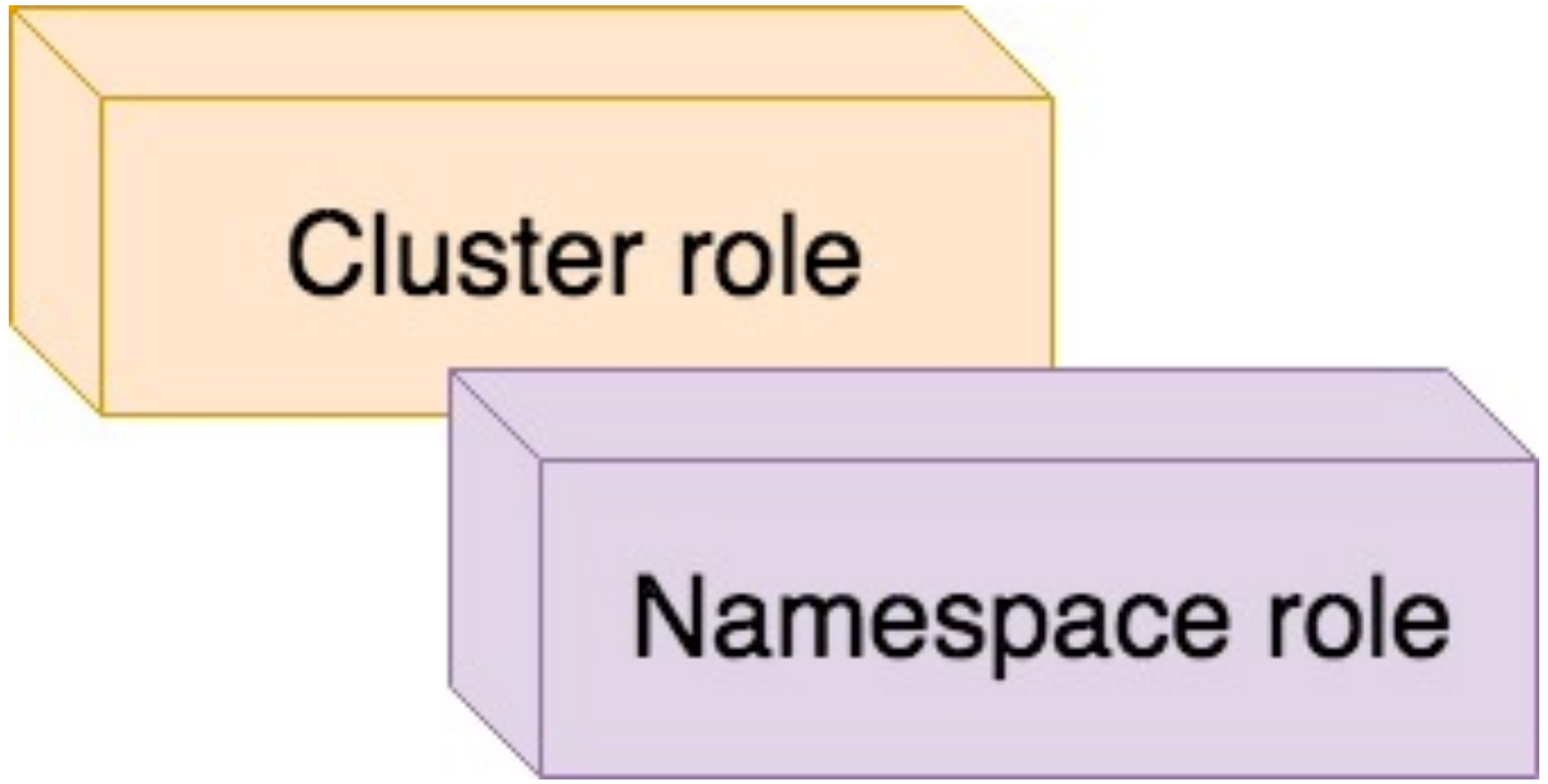
RBAC authorization cross clusters


User
Role
Rights

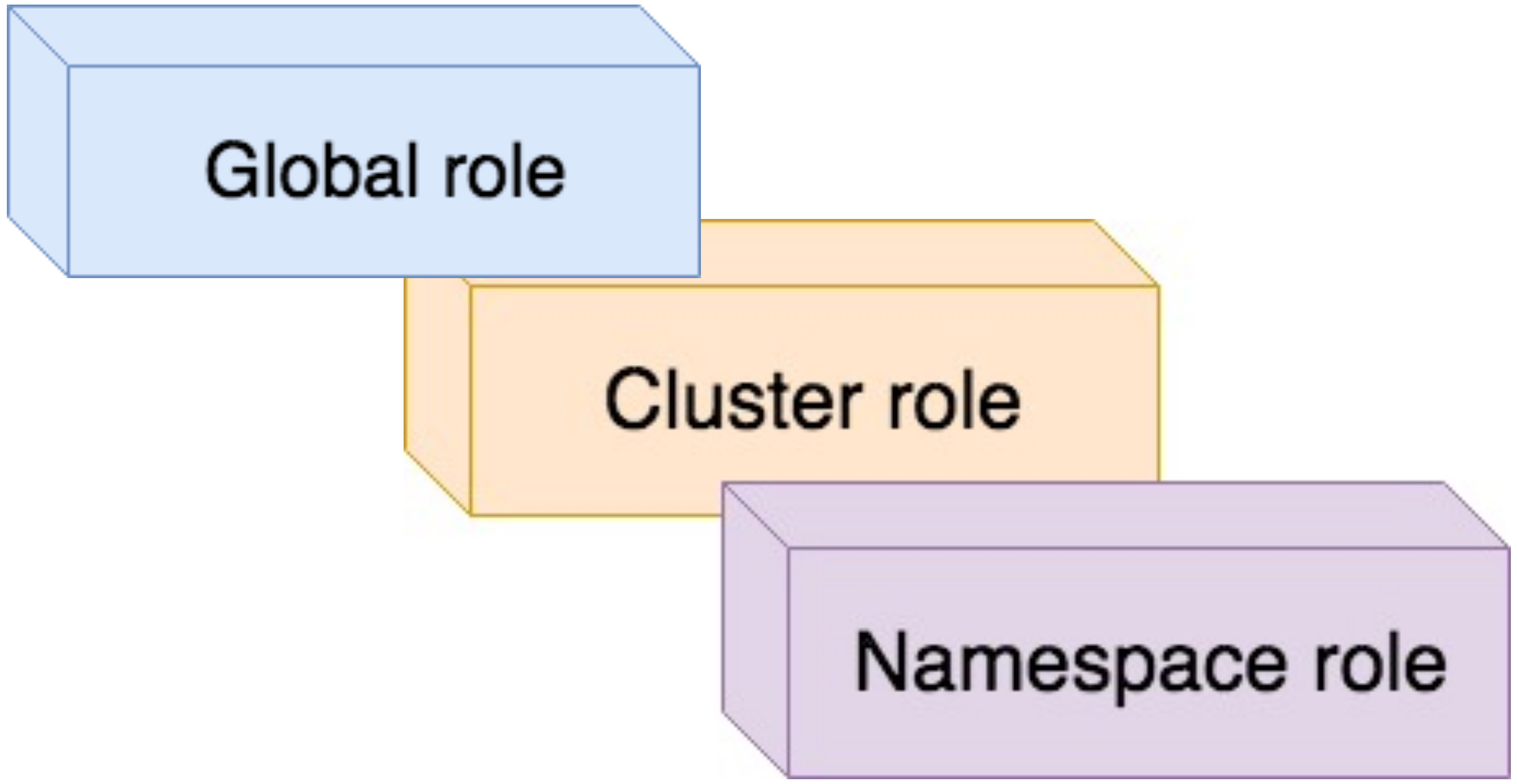
Role-Based Access Control (RBAC)

- In enterprise setting, access may be based on job function or role of a user
  - Payroll manager, project member etc.
  - Access rights are associated with roles
- Users authenticate themselves to the system
- Users then can activate one or more roles for themselves

RBAC Roles level in Kubernetes



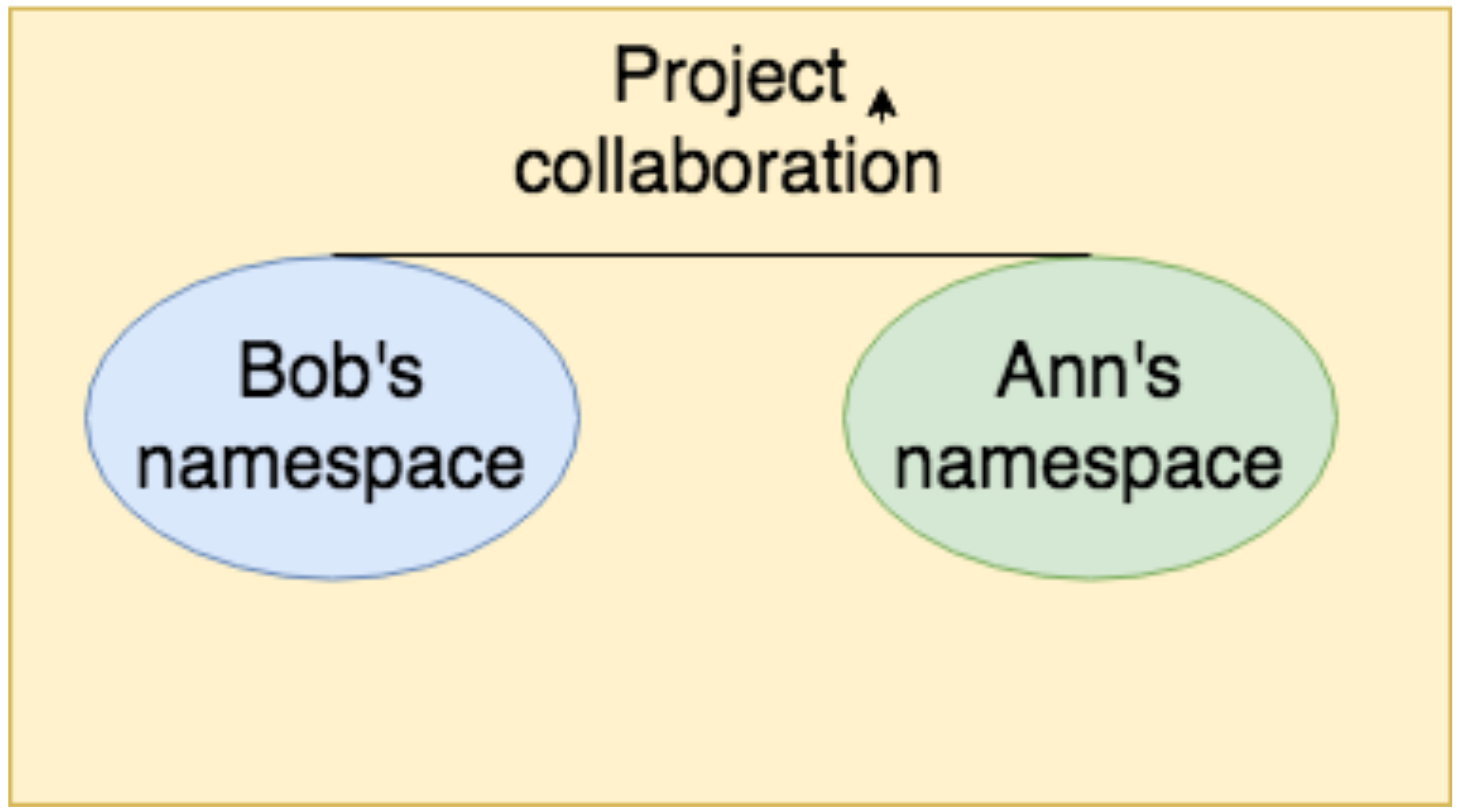
Multi cluster management roles

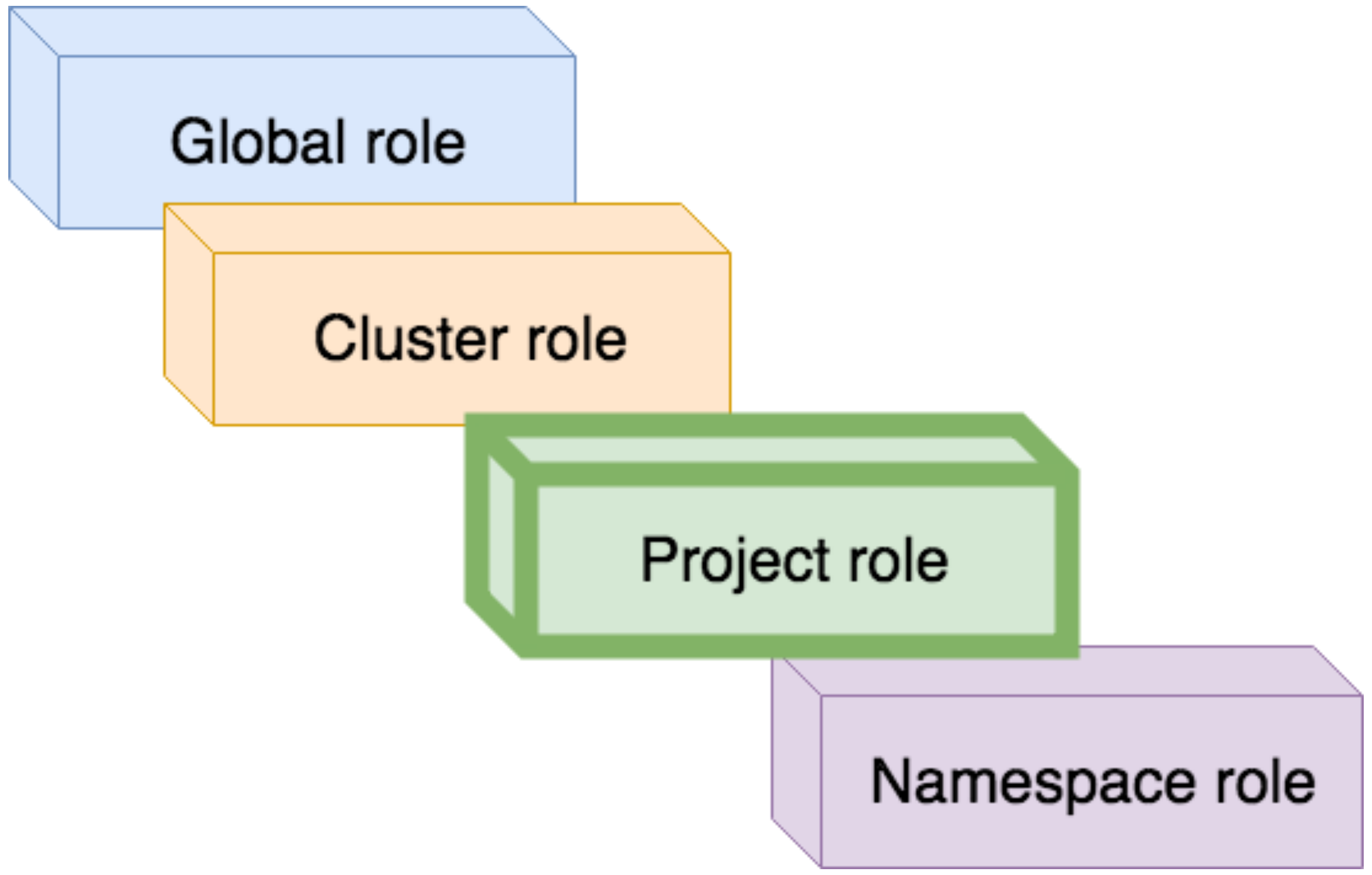


Global role is a new CRD used to

- Manage users
- Manage user roles
- Manage authentication configs

Need for teams collaboration calls for an extra role



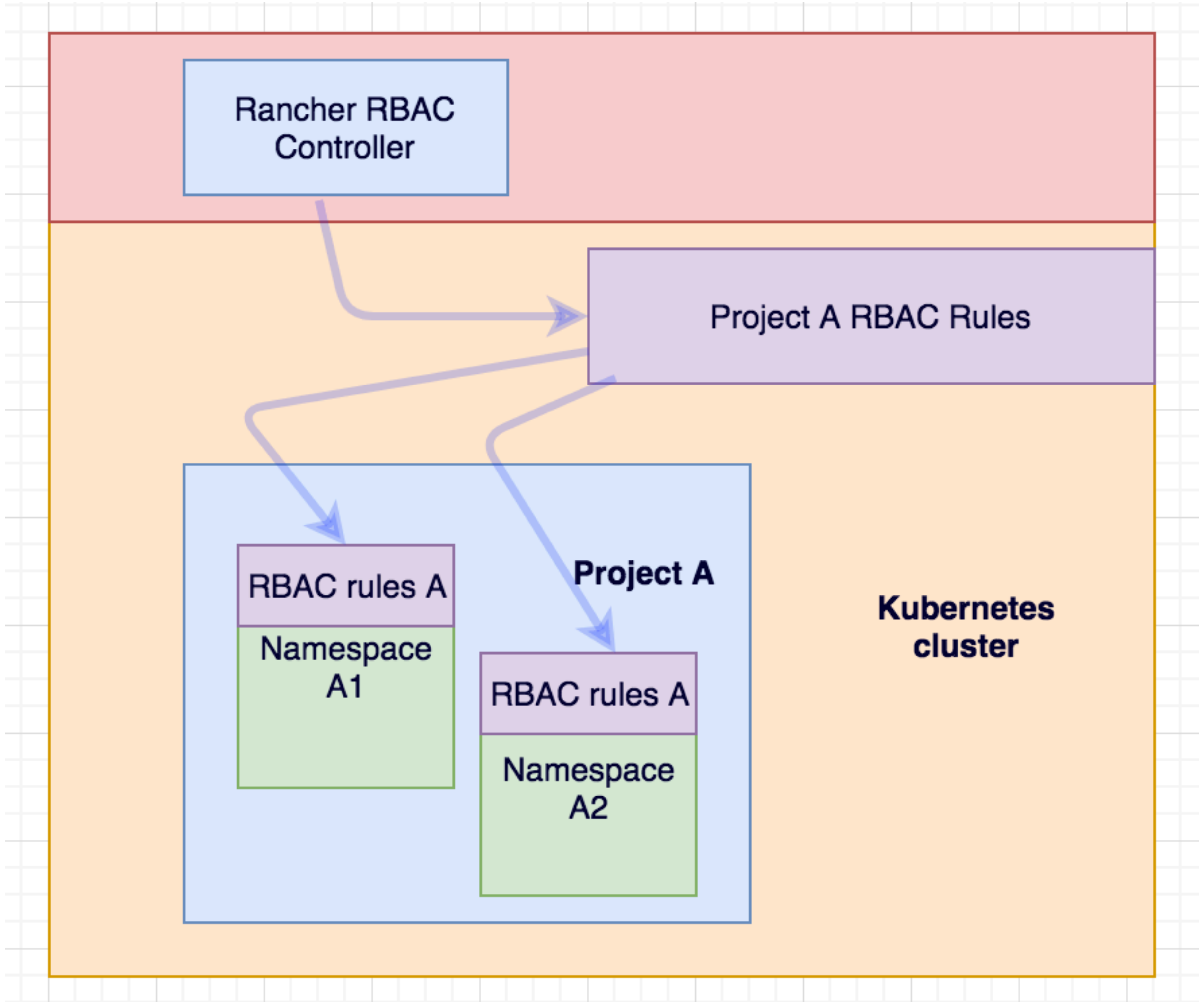


Project is

- A collection of namespaces
- A way to define RBAC rules **once** for a group of namespaces
- Ensures automatic RBAC inheritance once the user is added to the project

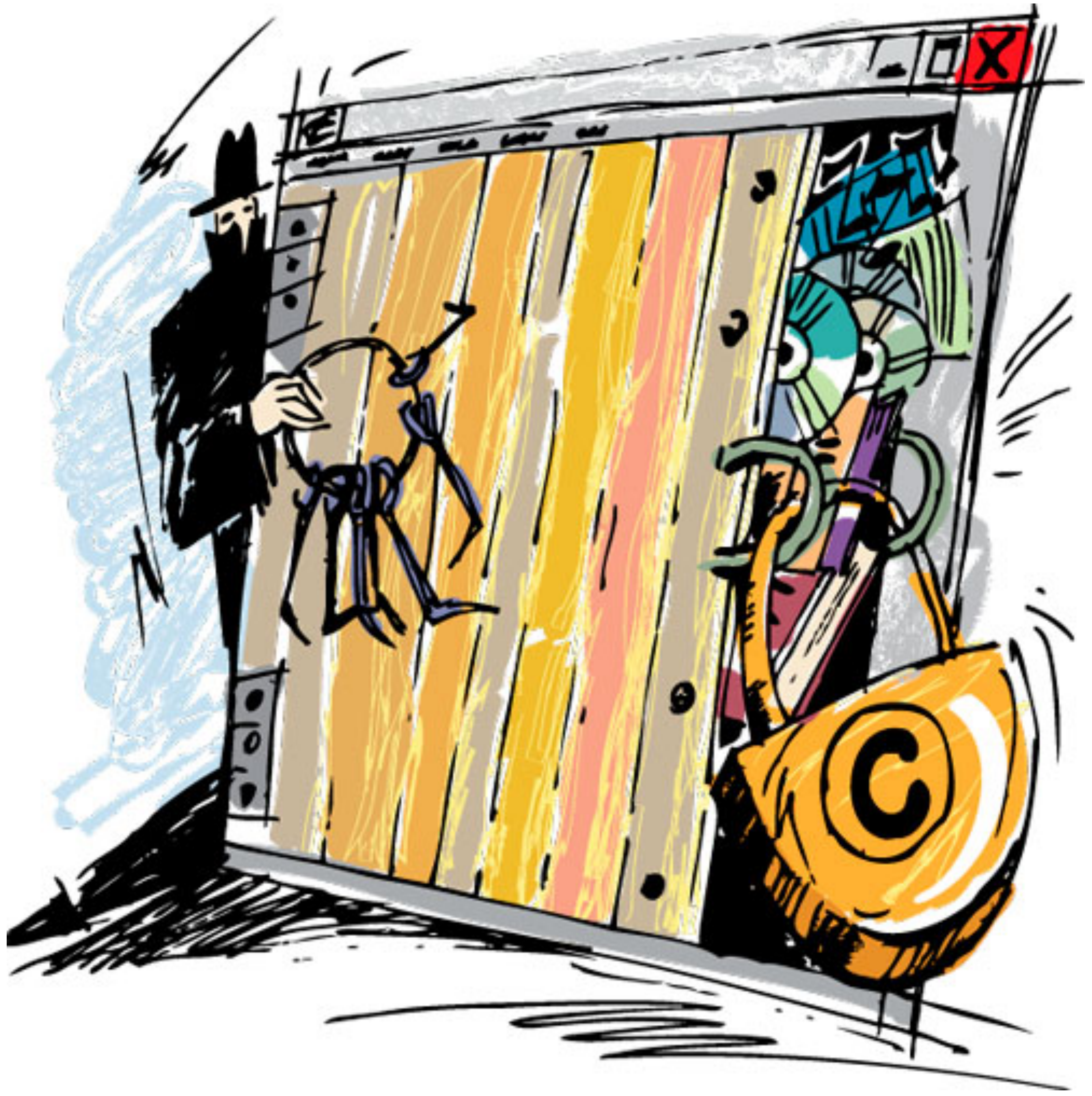
Project RBAC controller

- Controller subscribes to user add/remove events
- Copies RBAC rule to every underlying namespace
- Once user is removed from the project, the RBAC rules are revoked from namespaces

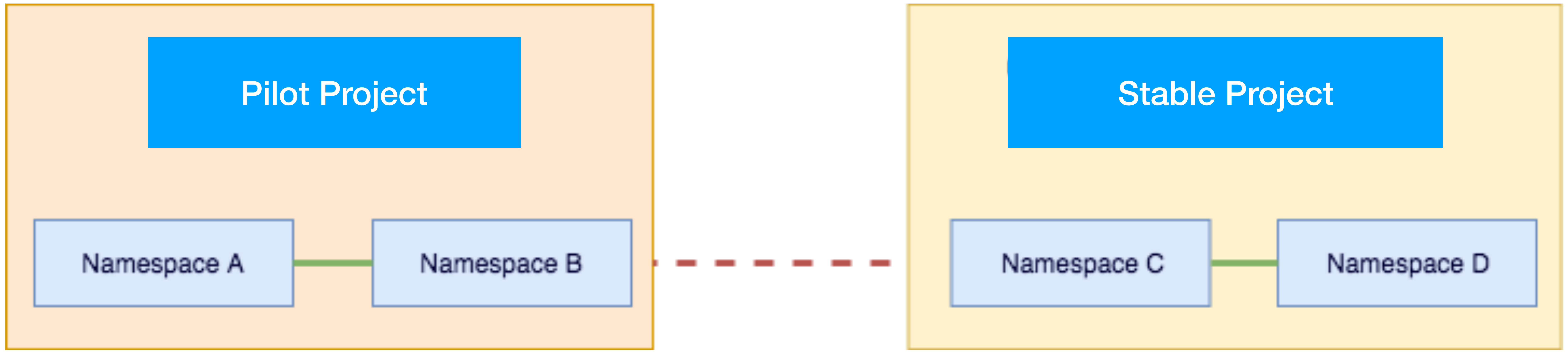


Infrastructure protection on a project level

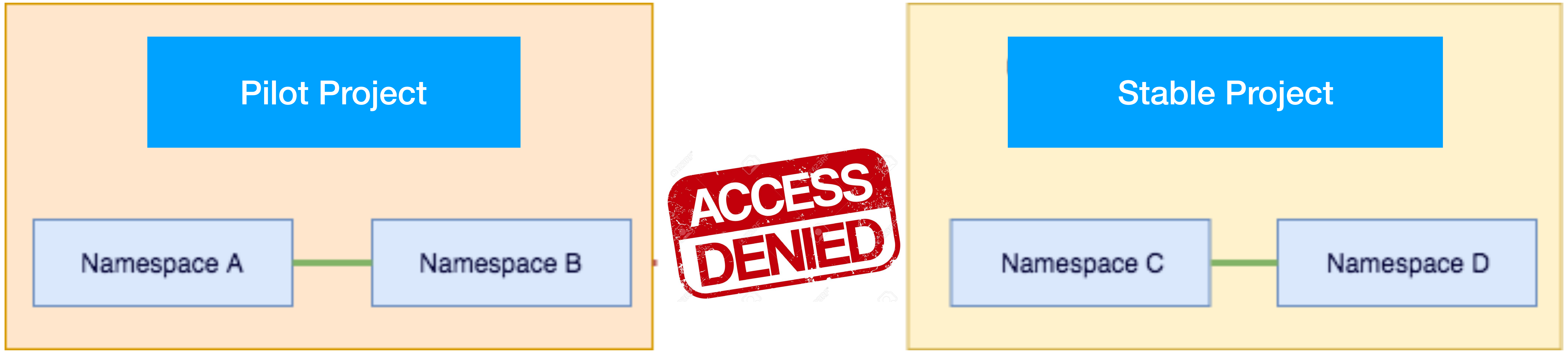
- Network access
- Pod Security policy
- Resource quota management



Network policy on a project is a great way to support multitenancy



Network policy on a project is a great way to support multitenancy



Demo time

Thank you!

Alena Prokharchyk,

Principal Software Engineer @RancherLabs

@lemonjet

alena1108