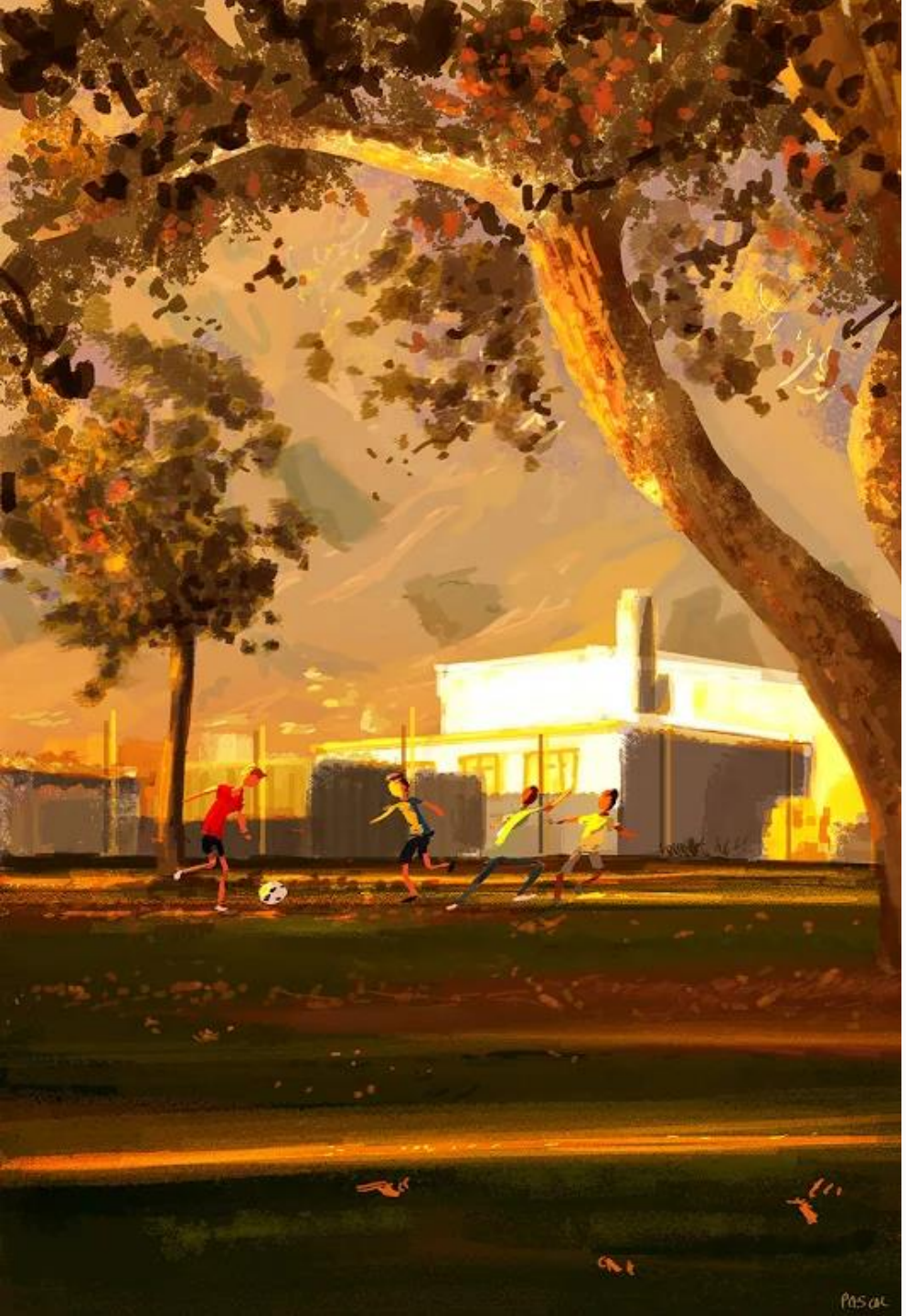


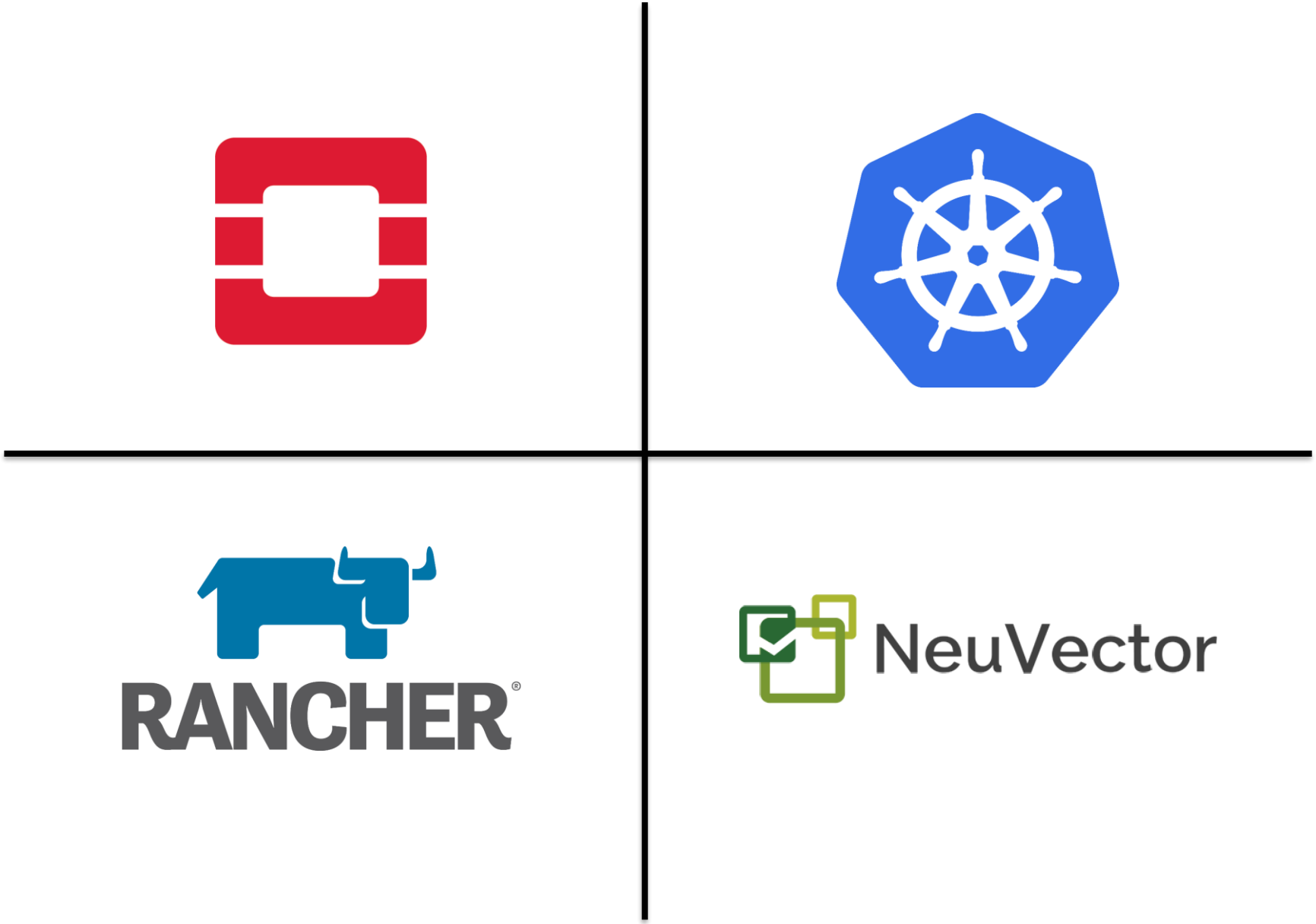
Kubernetes Rechte- und Rollenkonzepte im Unternehmen

Christian Frank

Presentation - September 26th, 2019

OWL Tech & Innovation Day, Paderborn

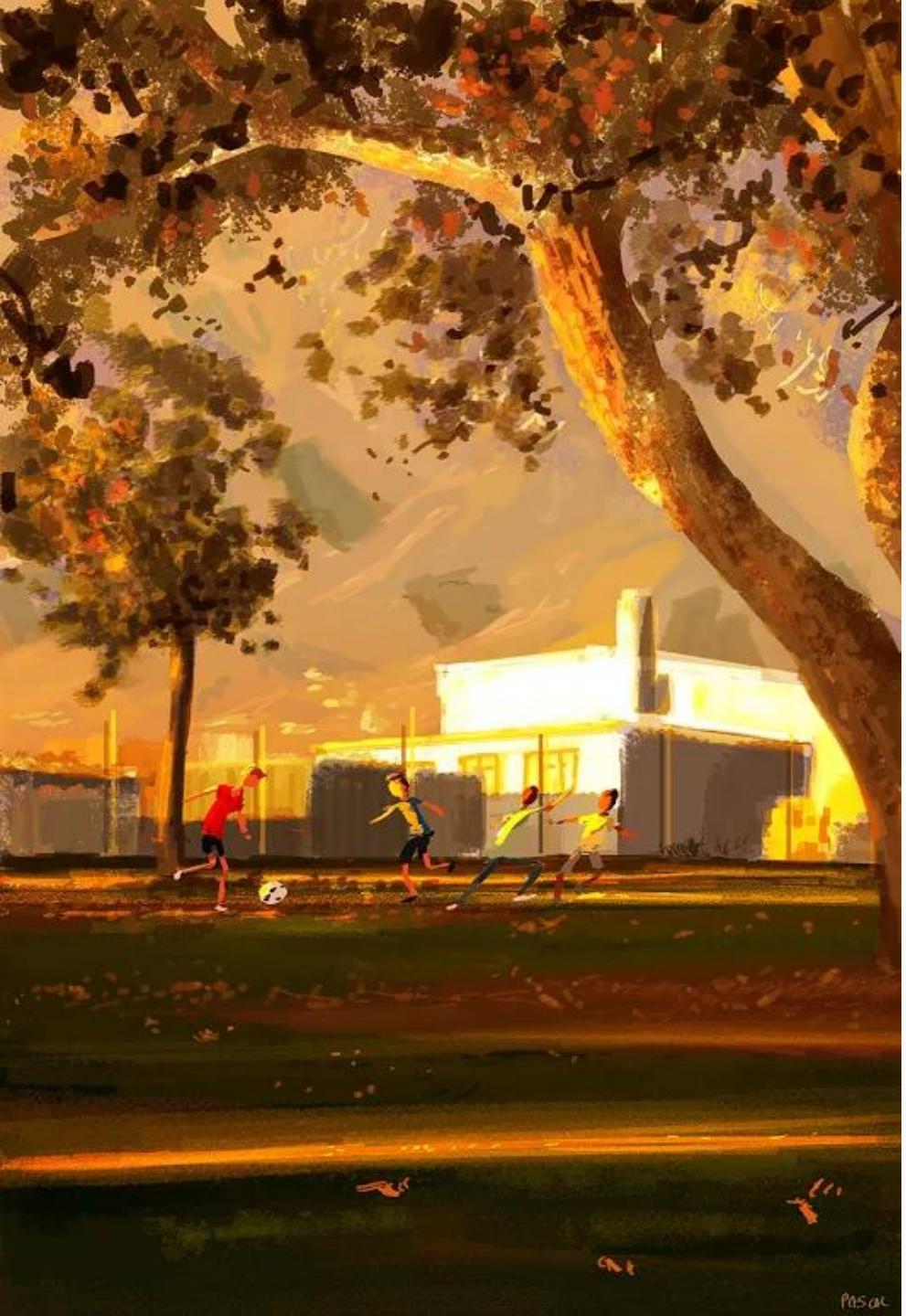
Experience



Key NIST Recommendations

- Tailor the organization’s operational culture and technical processes to support the new way of developing, running, and supporting applications made possible by containers.
- Segmenting containers by purpose, sensitivity, and threat posture provides additional defense in depth
- Orchestrators should use a least privilege access model in which users are only granted the ability to perform the specific actions on the specific hosts, containers, and images their job roles require

https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-190.pdf



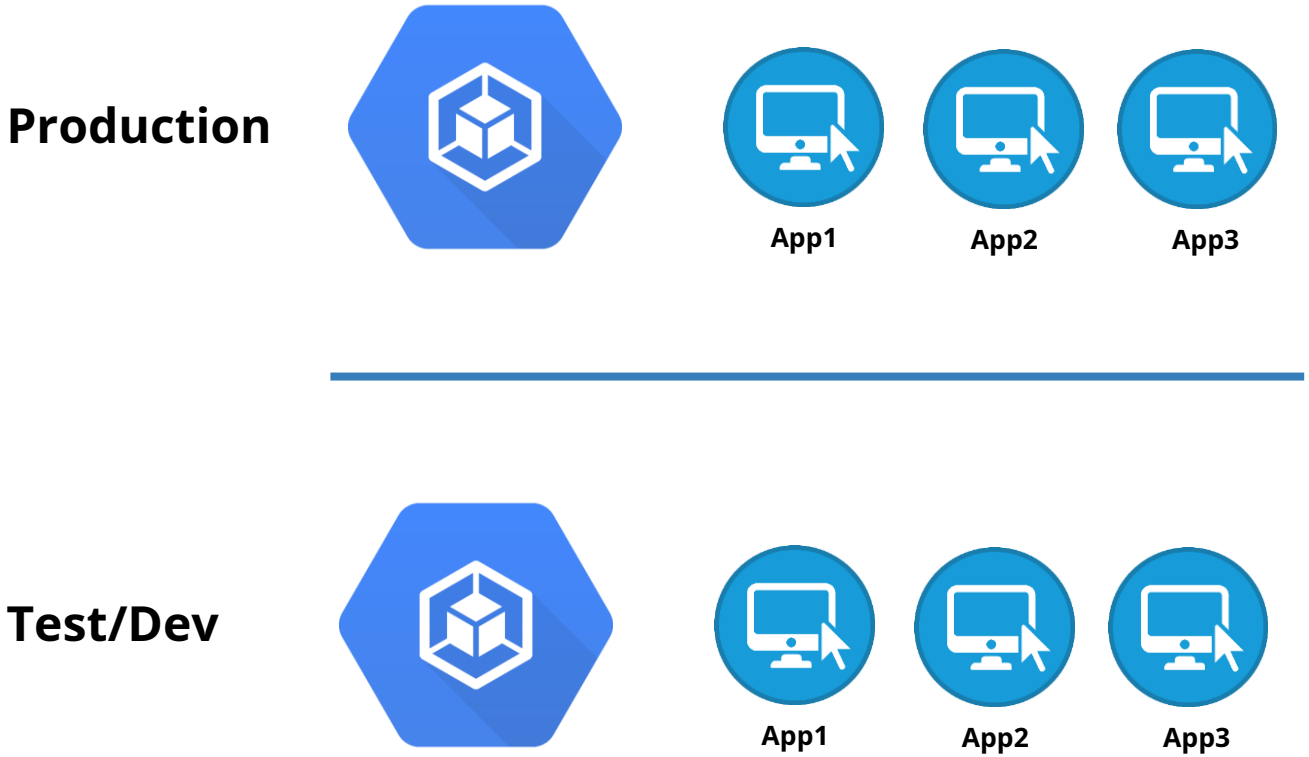
Segmentation

Segmentation Steps

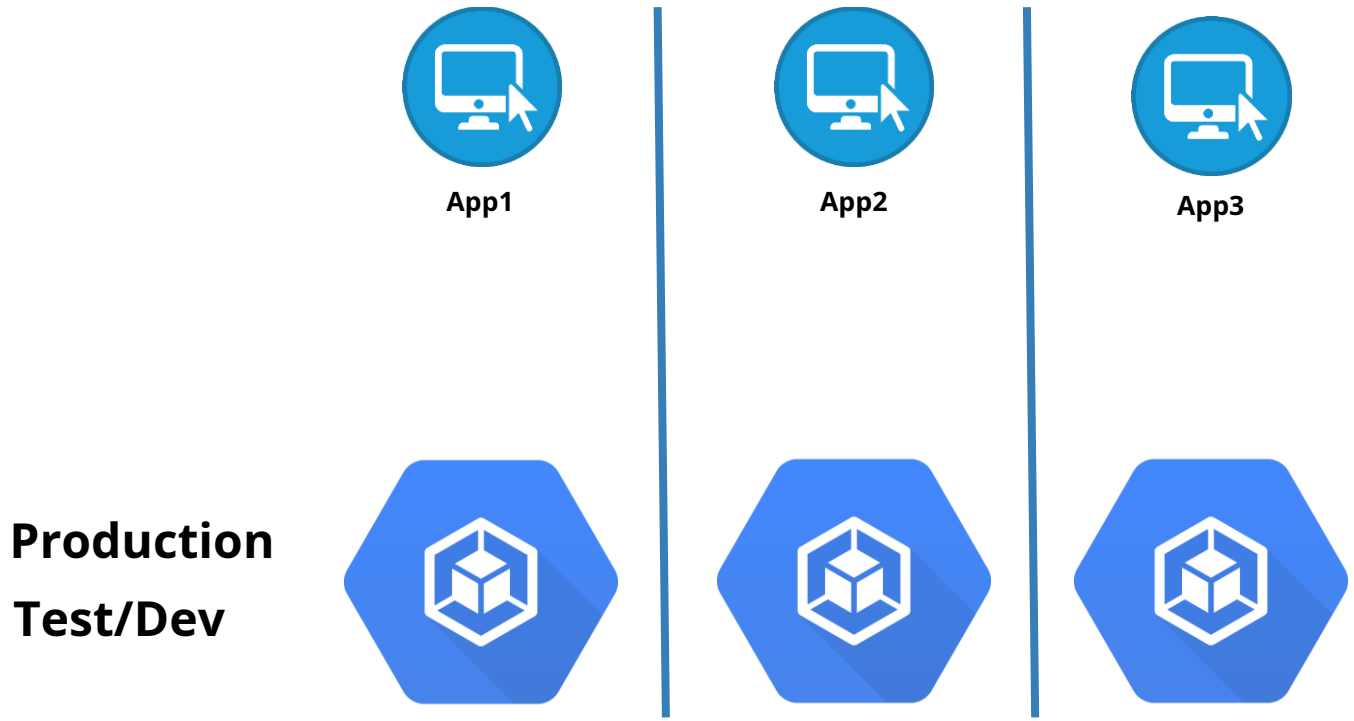
- Identify security classes, by function (Dev, Test, Prod) and by security / confidentiality level (DGSVO)
- Evaluate the applications and group them into the classes defined above
- Select one or more segmentation methods:

➢ By function
➢ By project
➢ Both

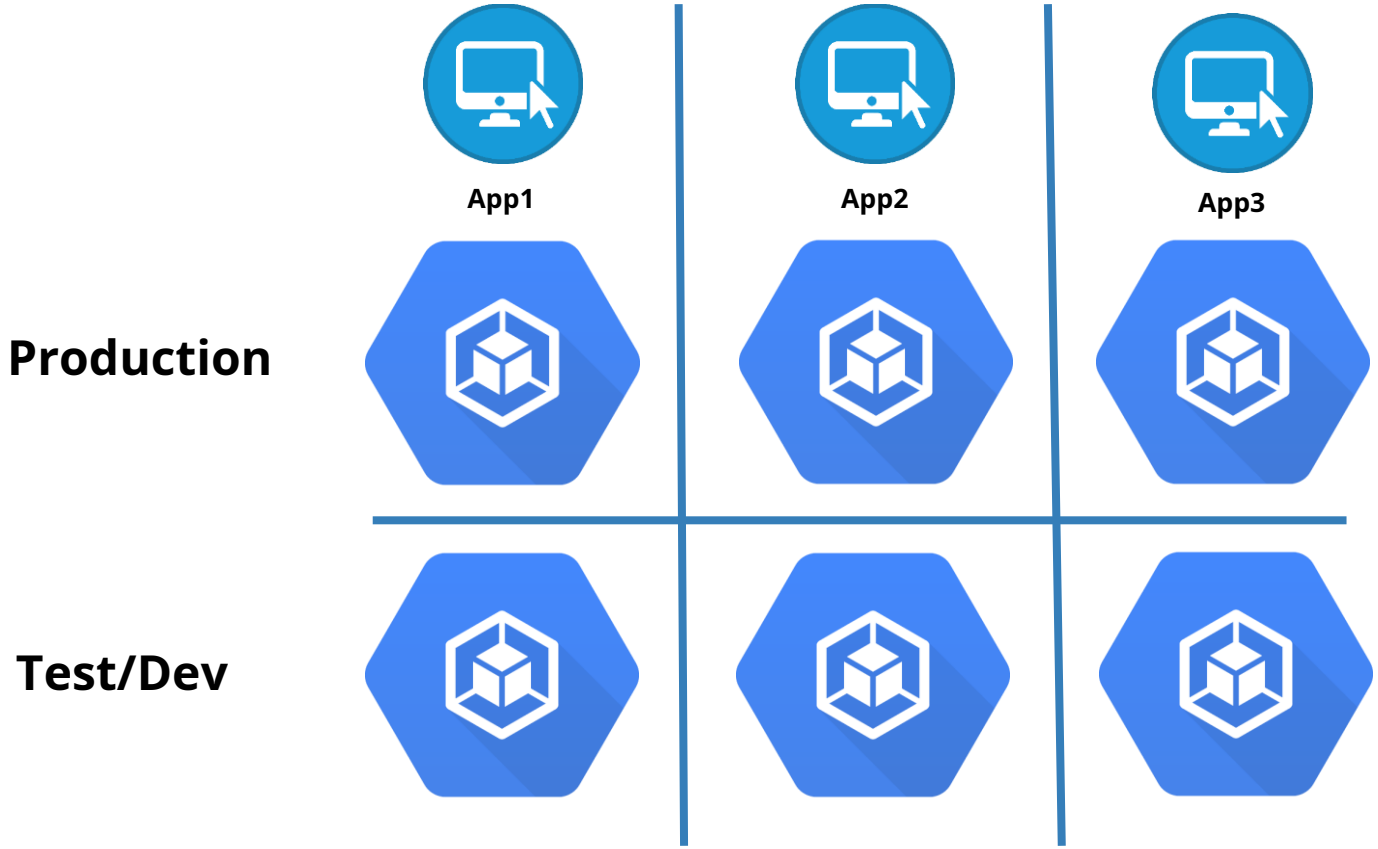
Segmentation by Function



Segmentation by Project



Segmentation by Function and Project



Implementation

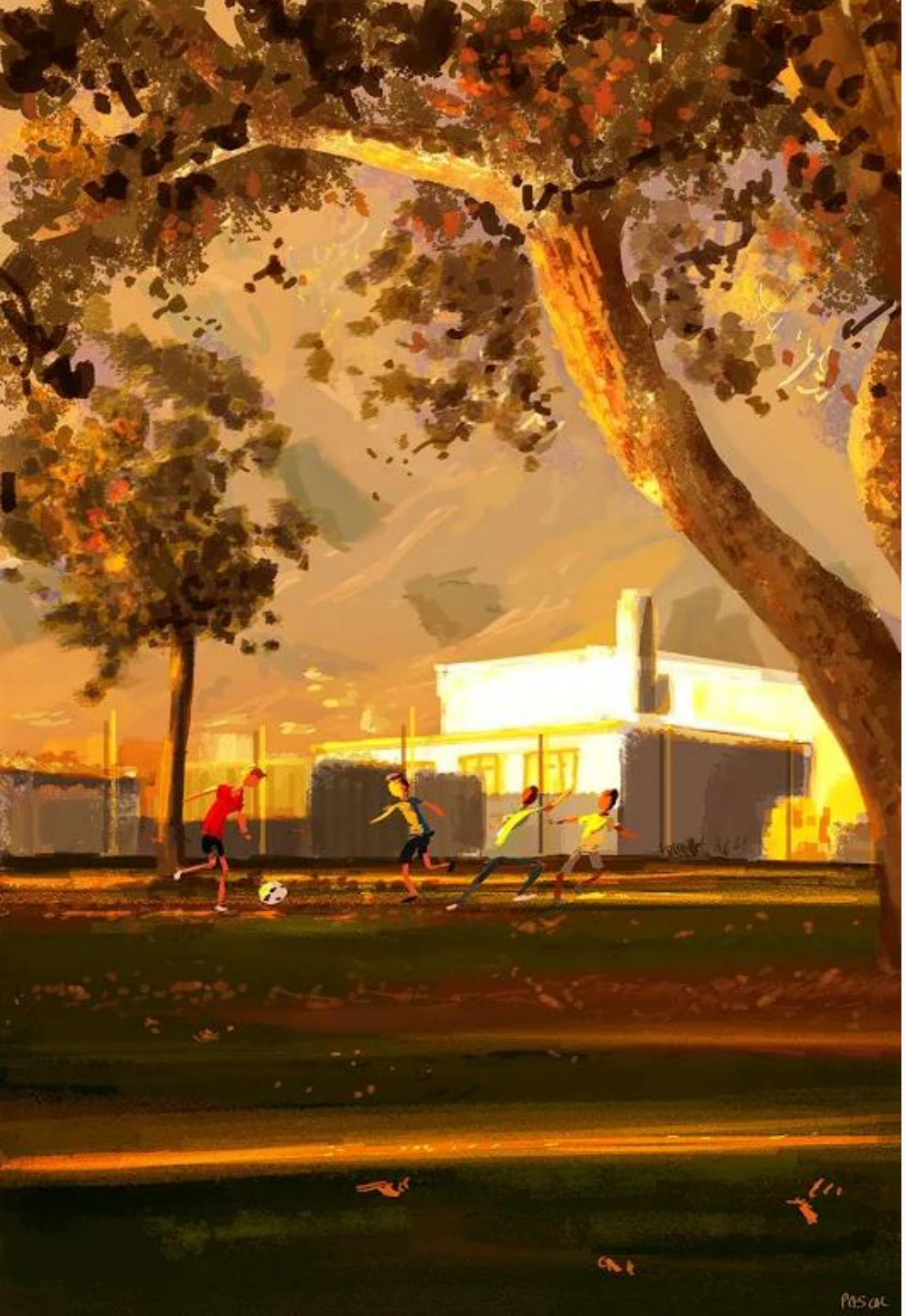
Kubernetes does not (yet) allow for hard tenancy.
There are a couple of ways to implement separation:

- Multiple clusters
- Using Network-plugins, such as Calico, that will allow network separation on a project level (with Rancher)

Considerations

- Administration effort
- “Blast radius”
- Compliance requirements
- Corporate Governance
- ISO 27001 (if applicable)

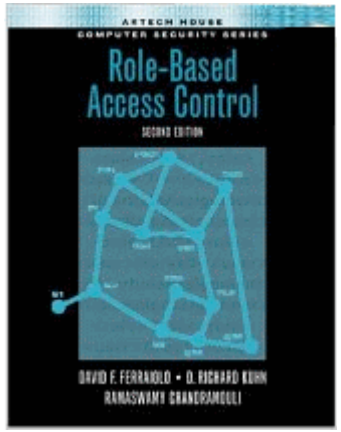
➢ In any scenario, I’d recommend an Enterprise Management Platform, such as Rancher



RBAC

RBAC Definition

- Role-based access control (RBAC) is a method of access control that is based on a user's role within an organization.



Role-Based Access Control, 2nd edition (2007)
by David Ferraiolo, Ramaswamy Chandramouli, and D. Richard Kuhn
https://csrc.nist.gov/Projects/Role-Based-Access-Control/RBAC-Library

RBAC

Applying RBAC to a system means:

- To implement a mechanism that allows users to access only the information or resources in the system that are necessary to perform their job, while preventing them from accessing information that is not relevant to their specific role.

See Catherine Cai: 3 Realistic Approaches to Kubernetes RBAC - https://thenewstack.io/three-realistic-approaches-to-kubernetes-rbac/

RBAC Role Engineering

The process of developing an RBAC structure for an organization has become known as "role engineering.“

Role engineering can be a complex undertaking; for example, in implementing RBAC for a large European bank with over 50,000 employees and 1400 branches serving more than 6 million customers, approximately 1300 roles were discovered

https://csrc.nist.gov/Projects/Role-Based-Access-Control/Role-Engineering-and-RBAC-Standards

RBAC Principle

Sounds familiar:

- RBAC is based on the information security **principle of least privilege.**

RANCHER

RBAC Implementation

- Identify cluster landscape. Multi-tenant, dedicated, etc.
- Identify Namespace “strategy”: What scopes are provided by namespaces? Are these scopes meaningful for RBAC?
- Identify roles of the users that access your clusters
- Identify projects that will access your clusters
- Identify additional policies that are relevant for RBAC configuration: Compliance rules, classified data etc.
- Create a good naming scheme

RANCHER

RBAC Elements in Kubernetes

- **Subjects**: Who wants to perform an operation in the cluster - User, Group, or Process
- **Roles**: Specify rules for accessing specific resources in the cluster or namespaces scope. A rule defines what operations (verbs) can be performed on a set of resources
- **RoleBindings**: Assigns permissions (roles) to a specific subject

RANCHER

Identity and Access Management

- Kubernetes does not provide any options to manage Users
- Best practices are to manage user identities via a central Identity Provider such as Active Directory or other directory services
- In addition, it could be important for enterprises to consider Single Sign-On (SSO), so that development and operations teams have a good user experience

RANCHER

Recap

- Proper RBAC management is critical to Kubernetes security
- Implementing RBAC requires knowledge of organizational roles, compliance and application security requirements
- It also requires knowledge of the built-in API resources
- Proper namespacing is critical for effective RBAC
- A good naming scheme will be required

RANCHER

Thank you

cfrank@chfrank.net