The Kubernetes Platform at DESY

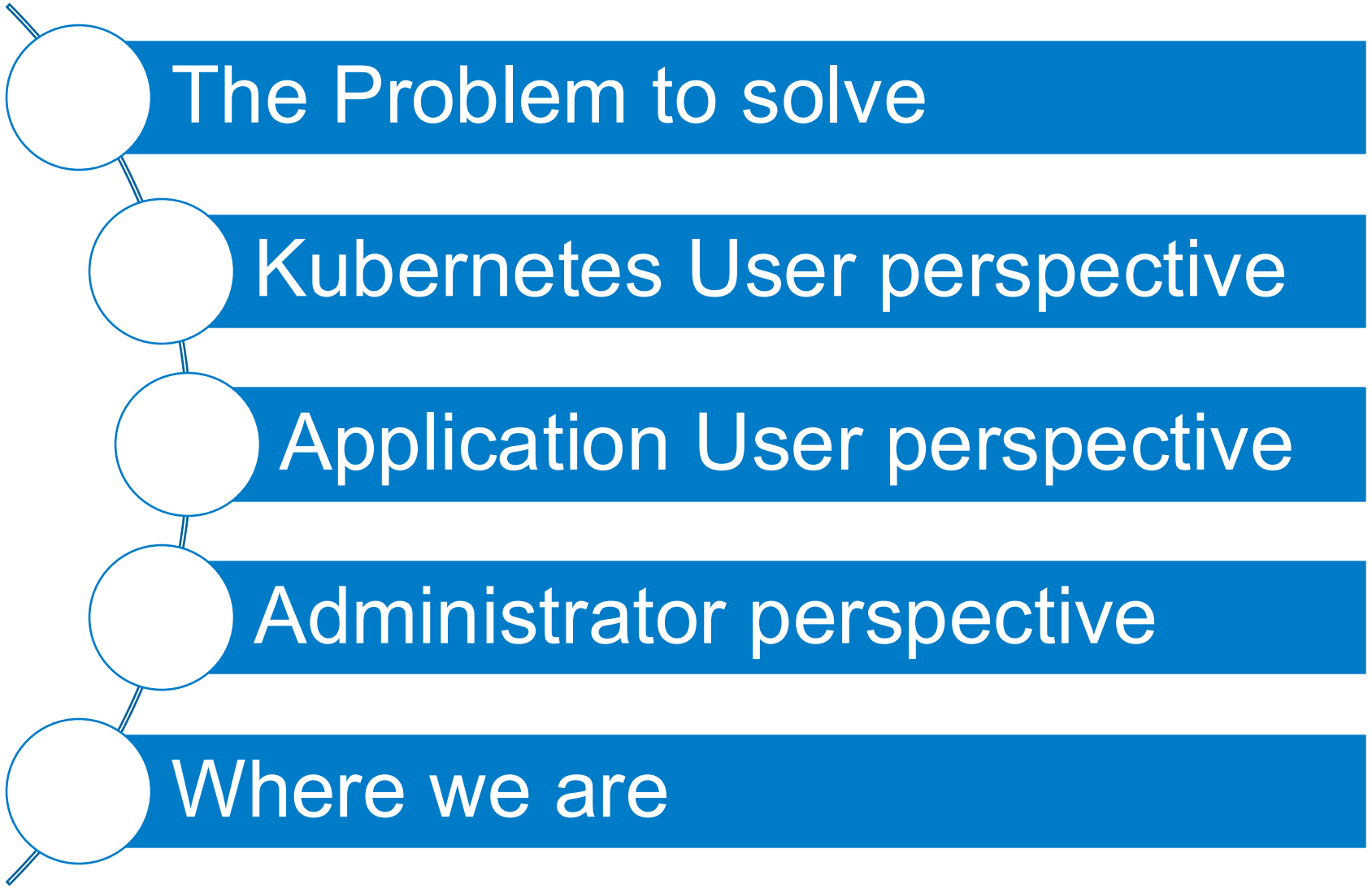
DESY-SESAME Scientific Computing Collaboration Meeting

Hamburg, 14th August 2025

HELMHOLTZ

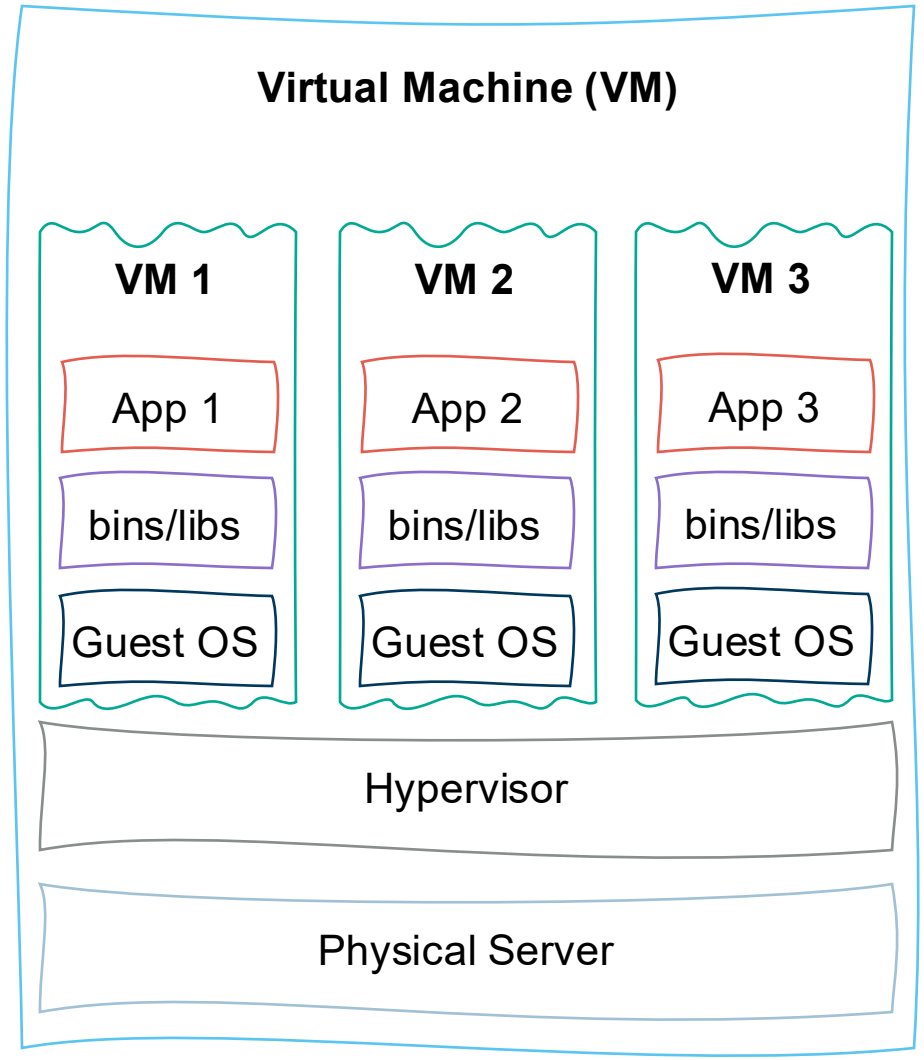
DESY.

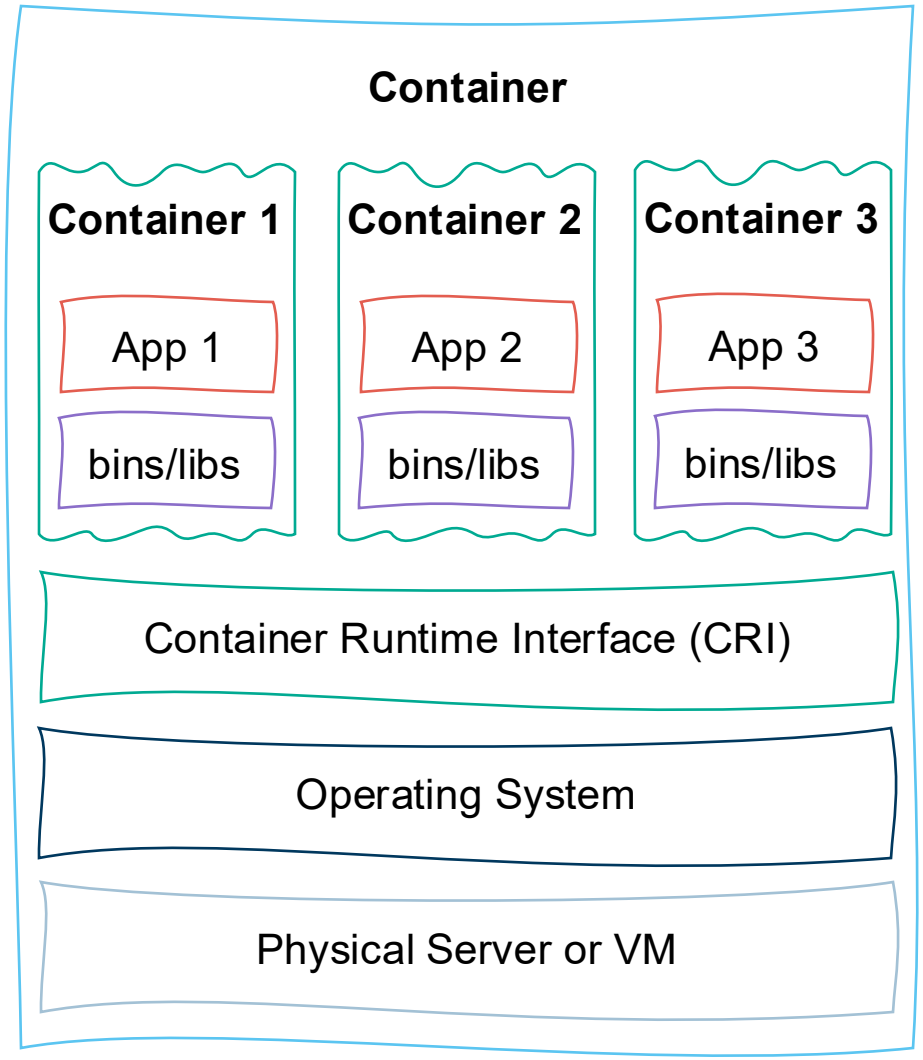
Agenda


The Problem to solve
Kubernetes User perspective
Application User perspective
Administrator perspective
Where we are

The Problem to solve

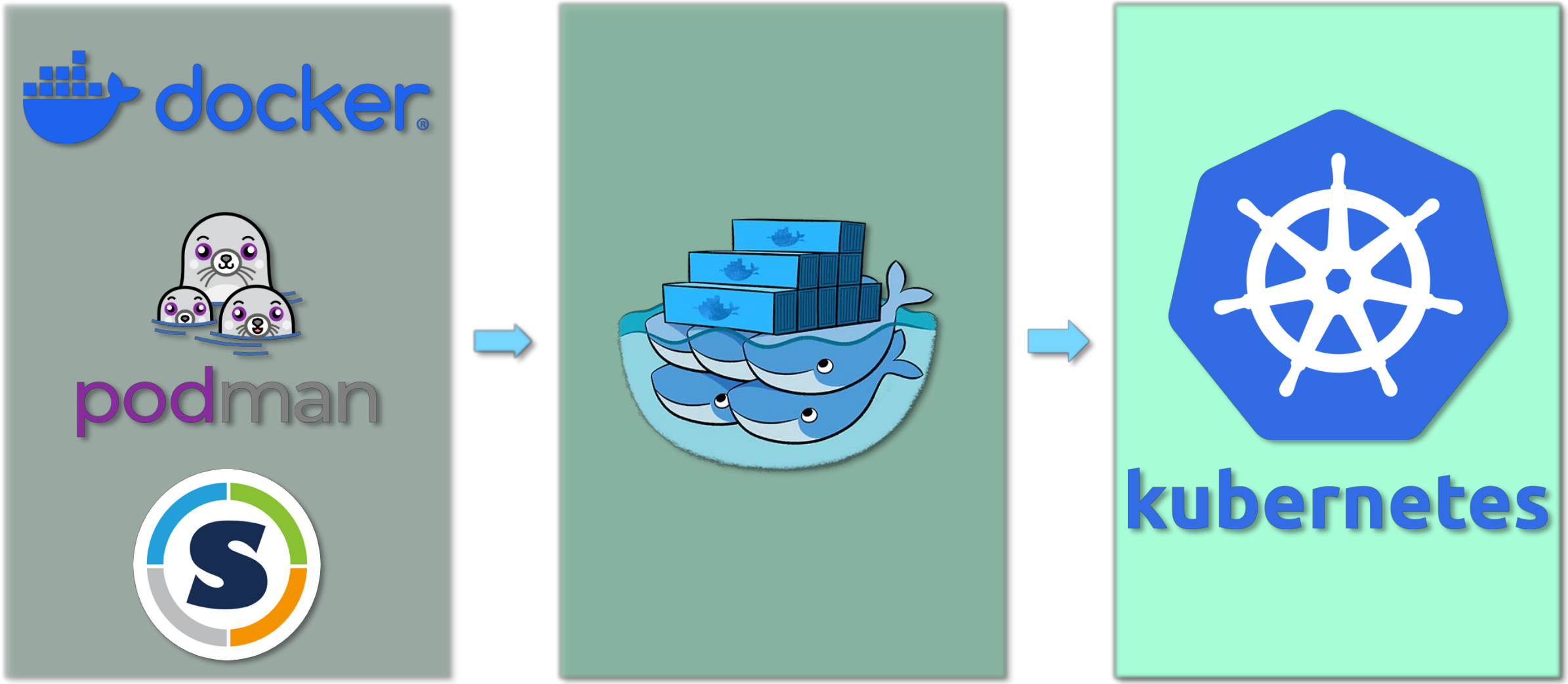
Containers vs. Virtual Machines





The Problem to solve

The brief History of Container Orchestration



The Problem to solve

Kubernetes

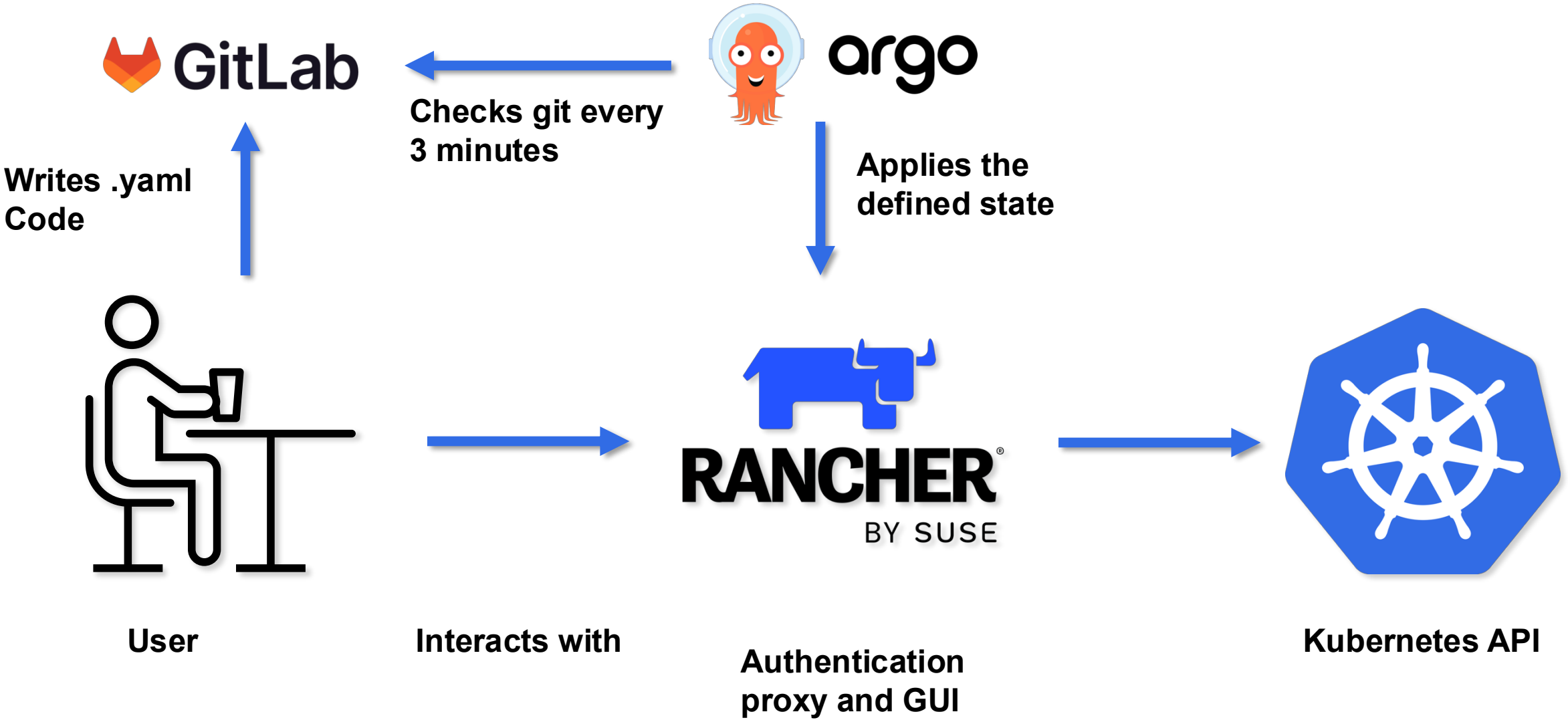
- Released on 9th September 2014 by the Cloud Native Computing Foundation (CNCF)
- 100% Opensource, Apache License 2.0
- Everything is controlled through the Kubernetes API
- Specifies core concepts and interfaces to adapt nearly every use-case
- Scalable from one to more than 65000 nodes
- Is the de facto industry standard for container orchestration

- Very, very wide opensource ecosystem around

kubernetes

Kubernetes User perspective

Our Concept

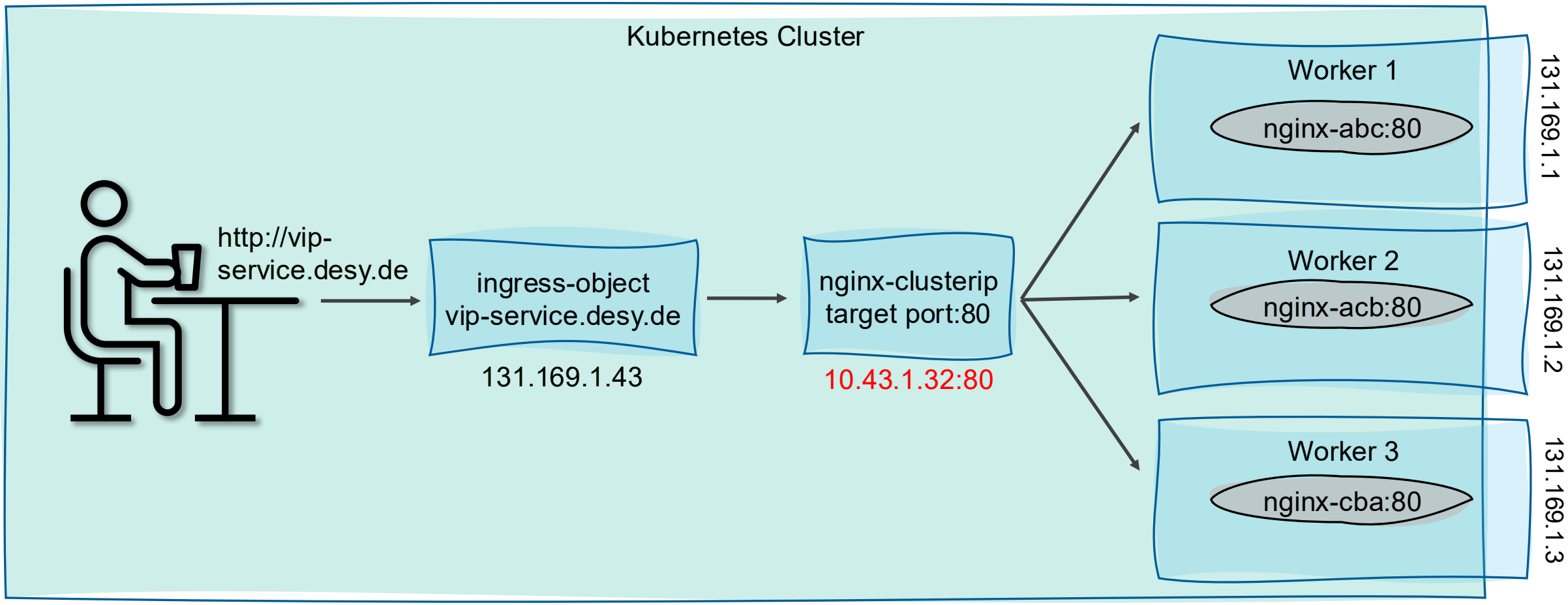


Kubernetes User perspective

Our Concept

- We deploy Kubernetes clusters on XEN-VMs in the internal or dmz network
- We manage Kubernetes with Rancher
- We separate applications namespace wise or cluster wise if necessary
- We offer ArgoCD to manage Kubernetes according to the GitOps approach
- We manage and provide for the users
  - Central metrics collection
  - Central container log collection
  - Ingress-Service
  - Software loadbalancer with DESY-IPs
  - Automatic renewal and certificate management
  - Central storage from NetApp and Ceph.

Application User perspective



Administrator perspective

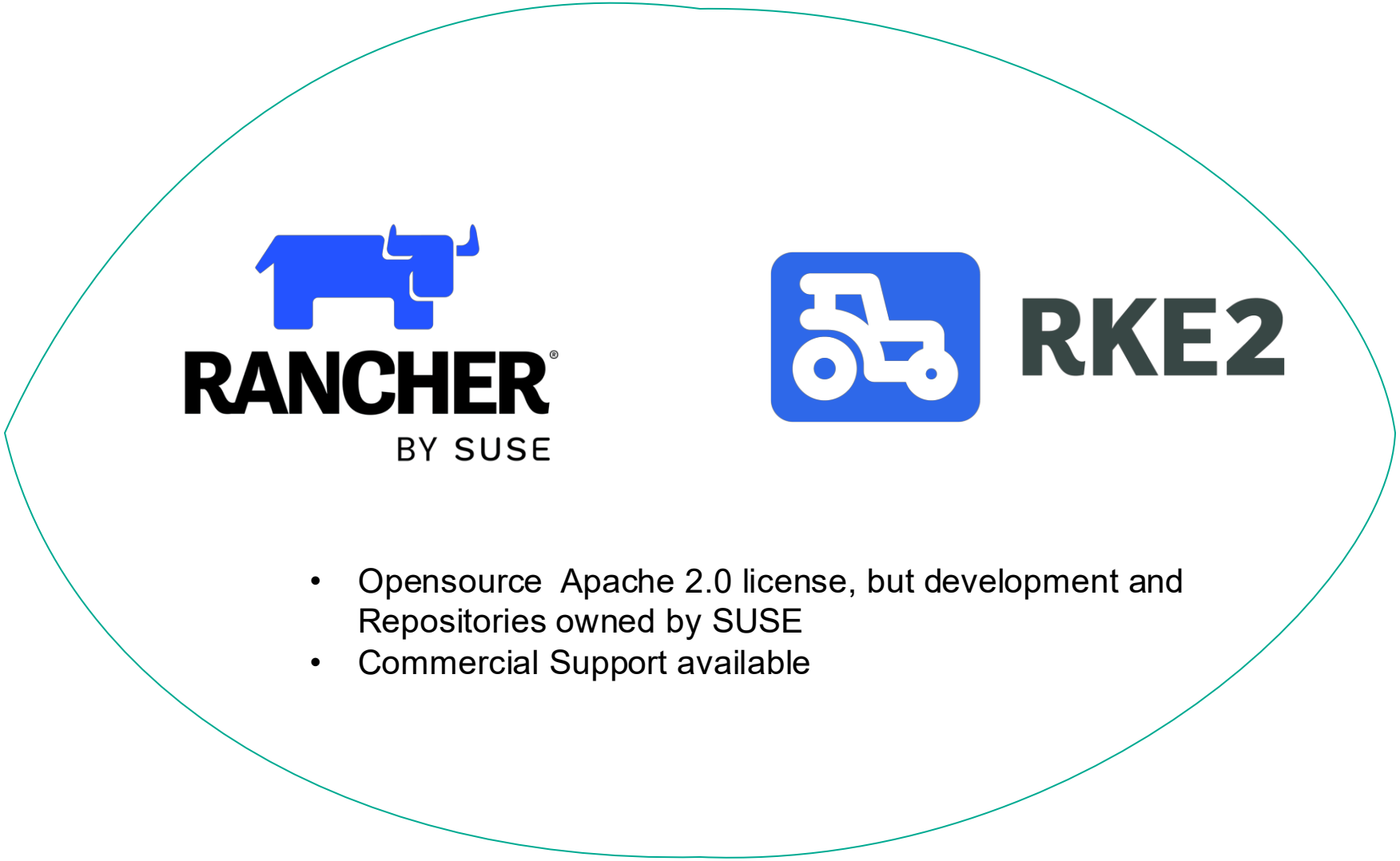
Our Kubernetes Tool stack

Every Software so far is backed by the Cloud Native Computing Foundation (CNCF)

The CNCF is part of the Linux Foundation focused on the Kubernetes and Cloud Native ecosystem

Administrator perspective

Our Kubernetes Tool stack



- Opensource Apache 2.0 license, but development and Repositories owned by SUSE
- Commercial Support available

Administrator perspective

Rancher

- We got a rancher instance in the **dmz** and in the **internal** network.
- Both instances are HA-Setups on Kubernetes
- It provides
  - A GUI
  - proxies and filters the requests against the Kubernetes API of managed clusters
  - extensive user management
  - cluster provisioning

- Intern: https://it-rancher.desy.de
- DMZ: https://it-rancher-dmz.desy.de

Administrator perspective

Argo CD

- We got **one** central instance in the **internal** DESY-Network
- Why we do not have an instance in the DMZ?
  - ArgoCD works with a pull-push mechanism
  - It pulls from git, renders templates internally and pushes the rendered output to the cluster
  - Because of this, a DMZ-Cluster can never ever do API-Calls against ArgoCD or another cluster.
- It provides:
  - A GUI
  - User management
  - GitOps mechanisms for Kubernetes
- https://argocd.desy.de

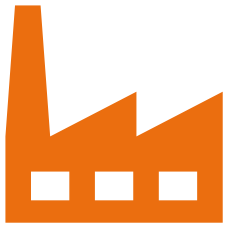
Administrator perspective

Putting all together

- We deploy Kubernetes clusters on XEN-VMs
- We manage Kubernetes with Rancher and RKE2
- We separate applications namespace wise or cluster wise if necessary
- We offer ArgoCD to manage Kubernetes according to the GitOps approach
- We manage and provide in every Cluster:
  - Central metrics collection (Prometheus + Thanos)
  - Central container log collection (fluentbit + ELK)
  - Ingress-service (Traefik + ingress nginx)
  - Software loadbalancer (metallb)
  - Certificate management (cert-manager)
  - Central storage from NetApp or Ceph

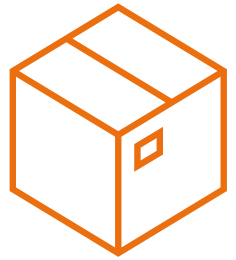
Where we are

In Numbers



14 production clusters on around 70 virtual machines

7 development clusters on around 30 virtual machines



Orchestration of **~500** user application **pods** (replicas included) in **35** Rancher projects

Where we are

What we already run on Kubernetes

- Messaging-service “Mattermost”
- Different GitLab Runners
- Container registry “Harbor”
- Databases “MogoDB, PostgreSQL”
- dCache CI/CD pipeline
- Multiple SciCat instances and development environments
- Helmholtz service portal
- Monitoring application “Thanos”
- Multiple small web applications
- Multiple development environments

Where we are

Future Plans

- Create a solution to use Scientific Posix Storage like dCache or IBM Spectrum Scale(GPFS)
- Test integration of slurm and HTCondor with the Kubernetes API
- Introduce the Kubernetes API for automatized Analysis pipelines
- Create a developer Kubernetes self-service portal

Thank you