

DevOps Technologies

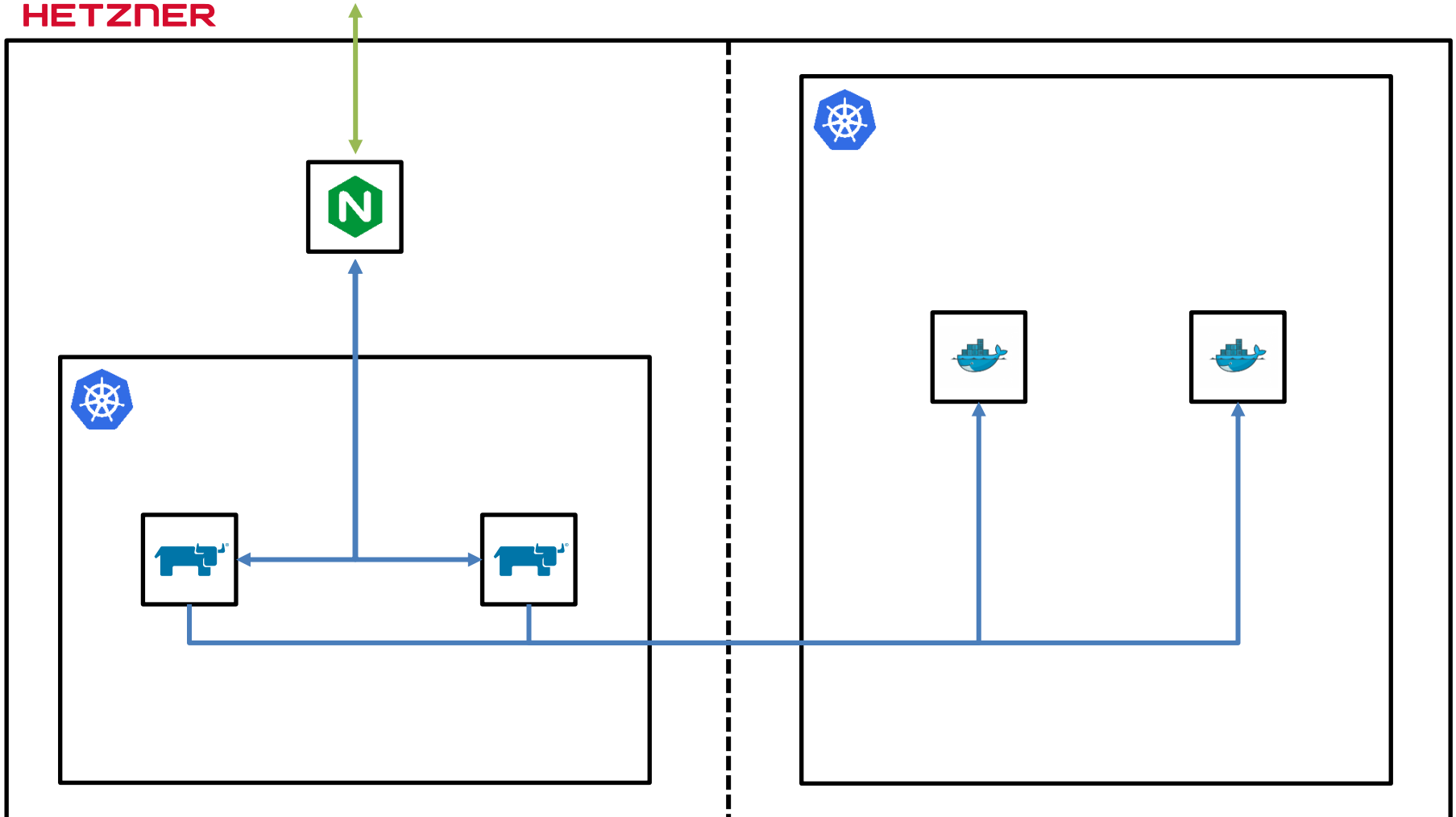
Lernziele Praxisblock

Software Engineering Research Group
Universität Kassel

Sebastian Copei
sco@cs.uni-kassel.de

Die Cloud – Infrastrukturlayer

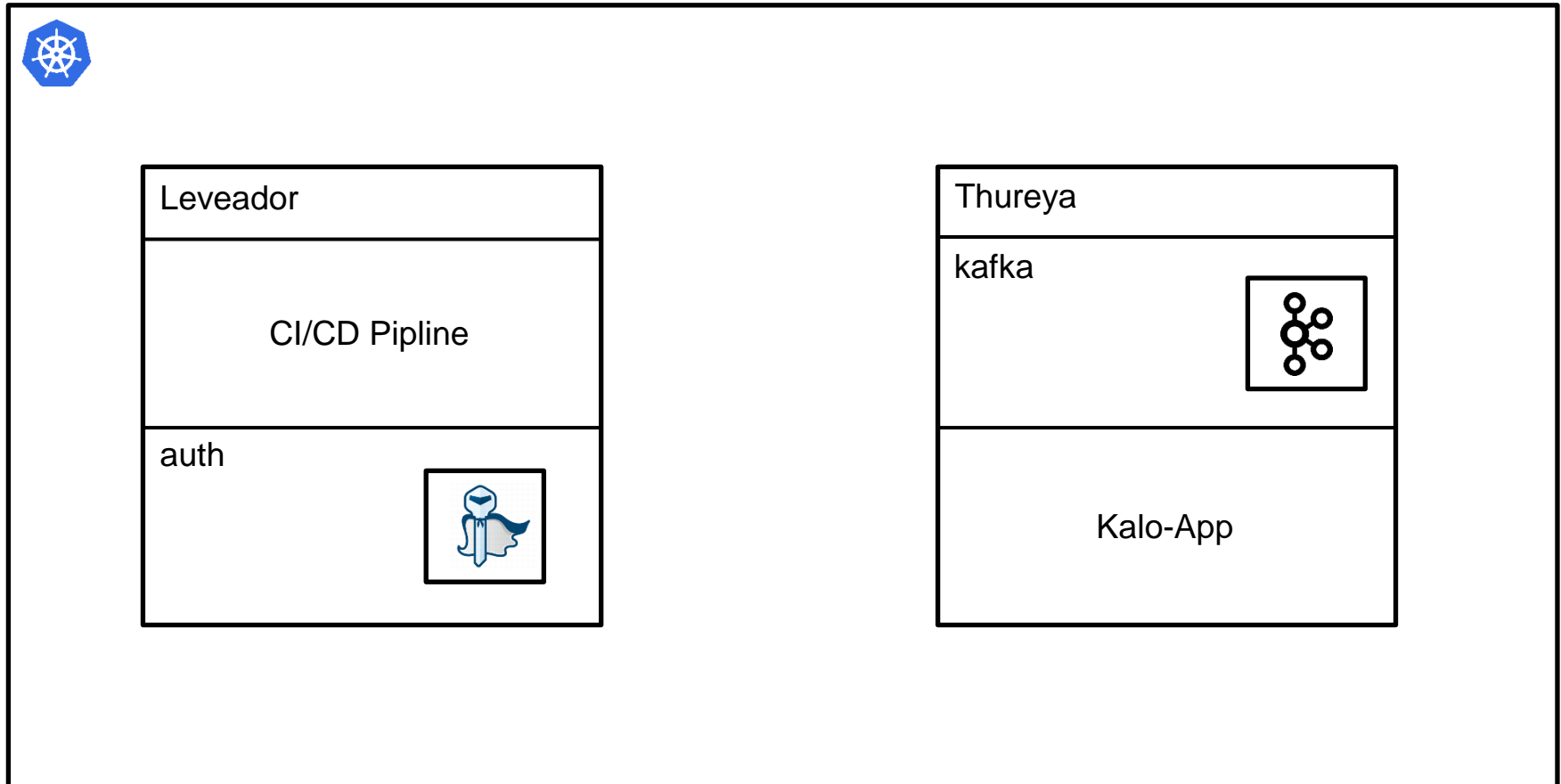
HETZNER



1. Meilensteinabgabe

- **Abgabefrist: 29.05**
- **Abgabeformat: Zoom, Termine über Doodle**
- **Inhalt:**
 - 1x Rancher Kubernetes Cluster mit
 - 1x Load Balancer
 - 2x Rancher Nodes
 - Rancher Server auf beiden Nodes installiert
 - 1x Downstream User Kubernetes Cluster mit
 - 2x User Nodes

User Cluster - Infrastrukturlayer



2. Meilensteinabgabe

- **Abgabefrist: 12.06**
- **Abgabeformat: Zoom, Termine über Doodle**
- **Inhalt:**
 - Ein Projekt mit Namespaces: dev, stag, prod, kafka, auth
 - Keycloak in auth namespace installieren
 - Kafka in kafka namespace installieren
 - Eine Datenbank, zb. Mongo, pro env (dev, stag, prod)
 - Alle Datenbank basierten Systeme müssen mit Passwörtern gesichert sein
 - GitHub für die Verwendung von Rancher Pipelines konfiguriert
 - Pipeline für Dev, Stag, Prod Env konfigurieren, mit allen Build Stages wie gezeigt.

Von Ops über Dev zu DevOps

- **Infrastruktur aufgesetzt (Ops)**



- **Services entwickeln (Dev)**



- **Pipelines konfigurieren (DevOps)**



3. Meilensteinabgabe

- **Abgabefrist: 10.07**
- **Abgabeformat: Zoom, Termine über Doodle**
- **Inhalt:**
 - Alles was vorher anstand
 - TIGK Stack für das Applikation Logging (Telegraf, Influx, Grafana, Kafka)
 - Dienst mit:
 - Keycloak Anbindung
 - Kafka Anbindung
 - Logging über Kafka
 - Query Anfragen über REST

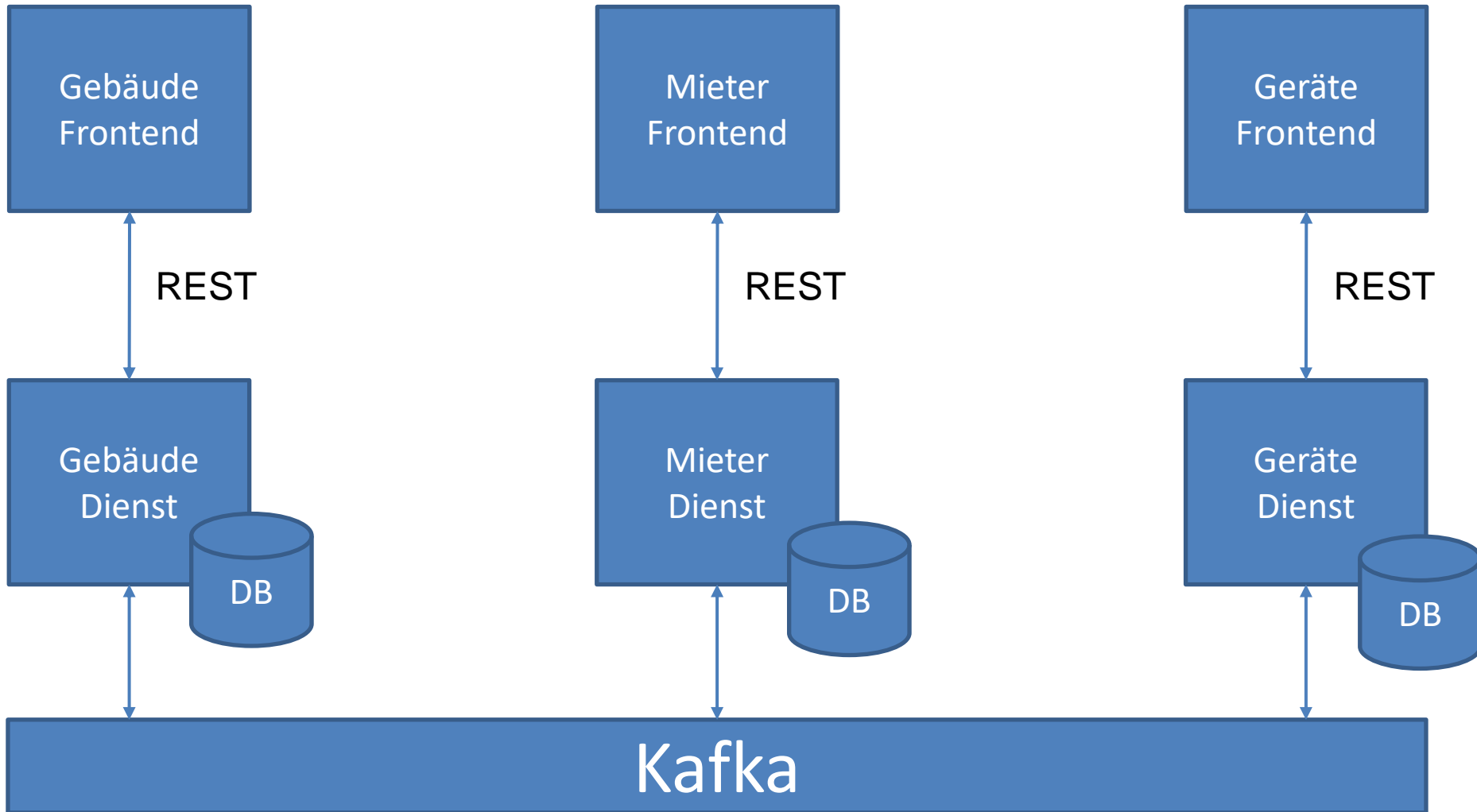
Projektabgabe

- **Abgabefrist: 28.08 (6 Wochen)**
- **Abgabeformat: Zoom, alle zusammen, Termin über Doodle**
- **Inhalt:**
 - Infrastruktur wie gezeigt mit:
 - Umgebungen Dev, Stag, Prod
 - Jeweils separate Namespaces für Keycloak und Kafka
 - Pro Umgebung 1x MongoDB, 1x Influx, 1x Grafana, 1x Telegraf
 - Falls Ressourcen zu knapp => Influx, Telegraf, Grafana in einen separaten Namespace und über Dashboard Tiles eine Unterscheidung zwischen den Umgebungen machen

Projektarbeit

- **Abgabefrist: 28.08 (6 Wochen)**
- **Abgabeformat: Zoom, alle zusammen, Termin über Doodle**
- **Inhalt:**
 - Jeweils ein Dienst für Gebäude, Mieter und Geräte mit den Inhalten erläuterten Inhalten
 - Sprache der Dienste egal. Support vom Betreuer gibt es für Typescript mit NestJS
 - Pro Dienst eine Oberfläche, Sprache und Framework egal. Support vom Betreuer gibt es für Typescript mit Angular.
 - Jeder Dienst integriert folgende Technologien:
 - Keycloak
 - Kafka für asynchrone Kommunikation und Logging
 - Command/Event Pattern wie gezeigt

Mögliche Architektur



Mögliche Architektur

