



# Kubernetes für die Lehre an der ETH Zürich

# Agenda

- Kubernetes-Infrastruktur
- Anwendung 1: SafeExamBrowser Server
- Anwendung 2: JupyterHub

Komplette Link-Liste als Handout...



# Kubernetes-Infrastruktur

Ziele: unabhängig vom Anbieter, selbständig, agil, automatisiert

Unterbau: VMware/Rancher, bereitgestellt durch zentrale Informatikdienste

Aufbau Winter 20/21, Start Herbstsemester 2021

~ 20 JupyterHubs im HS2022, 15 Prüfungen mit SEB-Server

# Aufbau Kubernetes-Infrastruktur

Must have:

- Persistenter Speicher: NFS
- Ingress/Loadbalancing: metallb (LB noch nicht ansteuerbar)

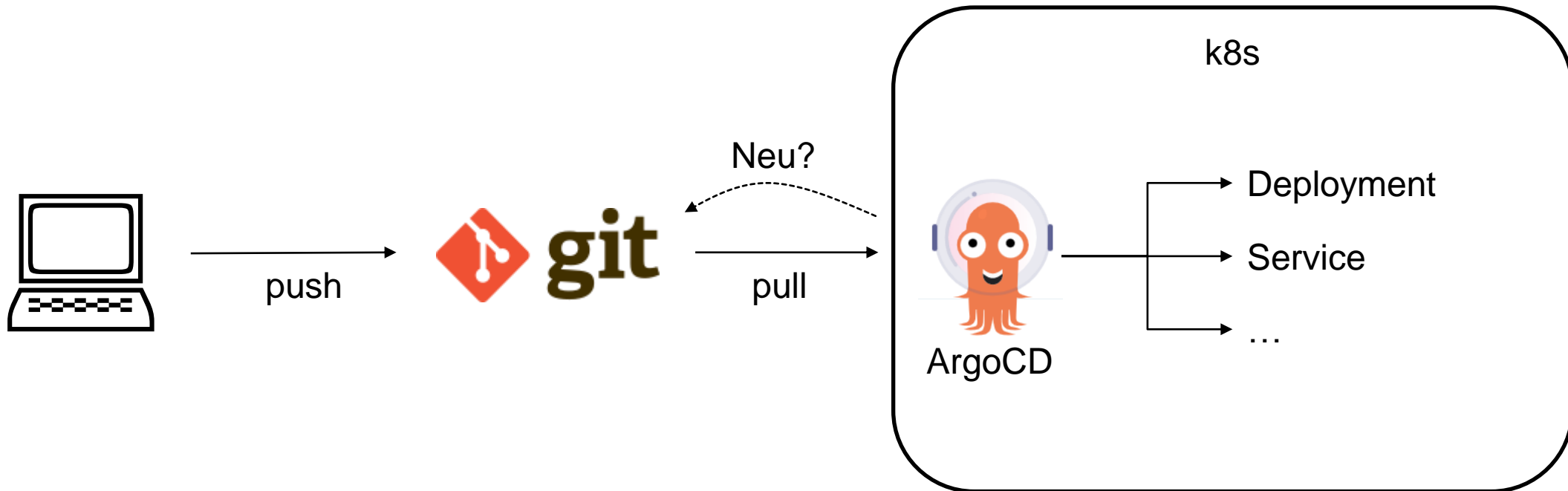
Automatisierung:

- Secrets in Git verschlüsseln: sealed-secrets
- NFS-Provisioner: Disks bereitstellen
- SSL-Zertifikate ausstellen: cert-manager (LetsEncrypt)
- DNS-Management: alias-manager (Kubernetes-Operator, Eigenentwicklung)
- ArgoCD als Deployer; ausser JupyterHub → Deployment-Scripts per Ansible

<https://dzone.com/articles/gitops-how-to-ops-your-git-the-right-way>

# Kubernetes-Infrastruktur

GitOps-Workflow: Infrastructure as Code + Pull statt Push



<https://dzone.com/articles/gitops-how-to-ops-your-git-the-right-way>

# Ressourcenbedarf

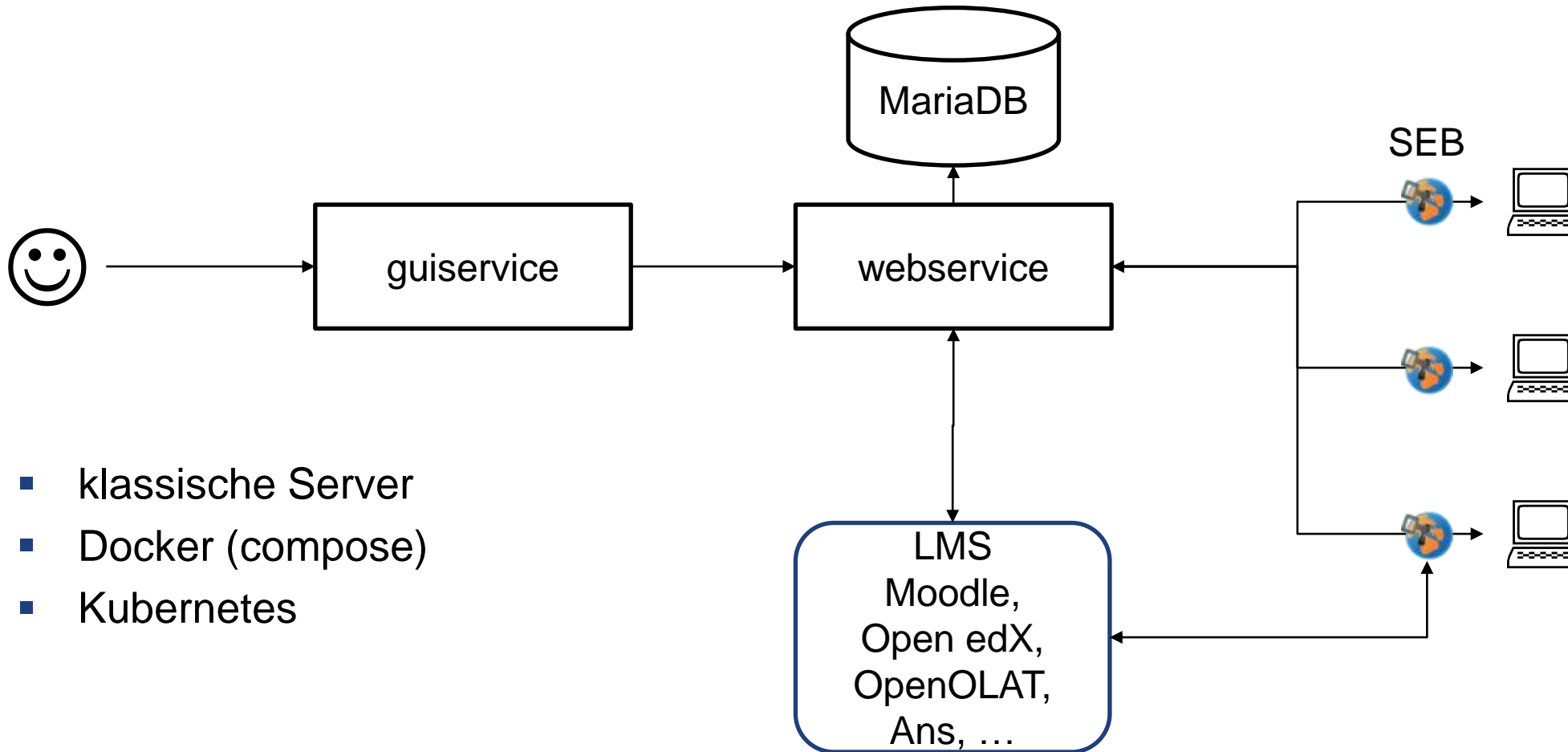
- Cluster-Auslegung ausgerichtet auf JupyterHub
    - 8 Worker-Nodes (+3 etcd-Nodes)
    - 48 GB/12 Cores: total 384 GB/96 Cores
    - keine GPUs (keine Taints/Affinities)
  - Ressourcen für ein Jupyter-Notebook: 1(-8) GB RAM, 1(-3) CPU
- ⇒ 380 Standard-Notebooks bei Vollausslastung; theoretisch, denn:
- Scheduler nicht optimal, und kein Rebalancing
  - Hub und System brauchen Ressourcen
  - nicht alle Notebooks laufen am Limit
- ⇒ ~500 bei statistischer Verteilung

# SafeExamBrowser (SEB) und SEB-Server



- SafeExamBrowser (SEB): abgesicherte Browser-Applikation (Kiosk-Modus für iOS, Windows, macOS), um Online-Prüfungen auf Learning Management Systemen (LMS) und Prüfungssystemen zuverlässig durchführen zu können
- SEB-Server: skalierbare Server-Komponente, um Online-Prüfungen mit SEB zentralisiert aufsetzen und sicher durchführen zu können
  - Konfiguration von SEB auf Clients
  - Status-Monitoring Clients
  - Live-Proctoring (Zoom, Jitsi Integration)
- Weitere Infos und Download: <https://safeexambrowser.org/>

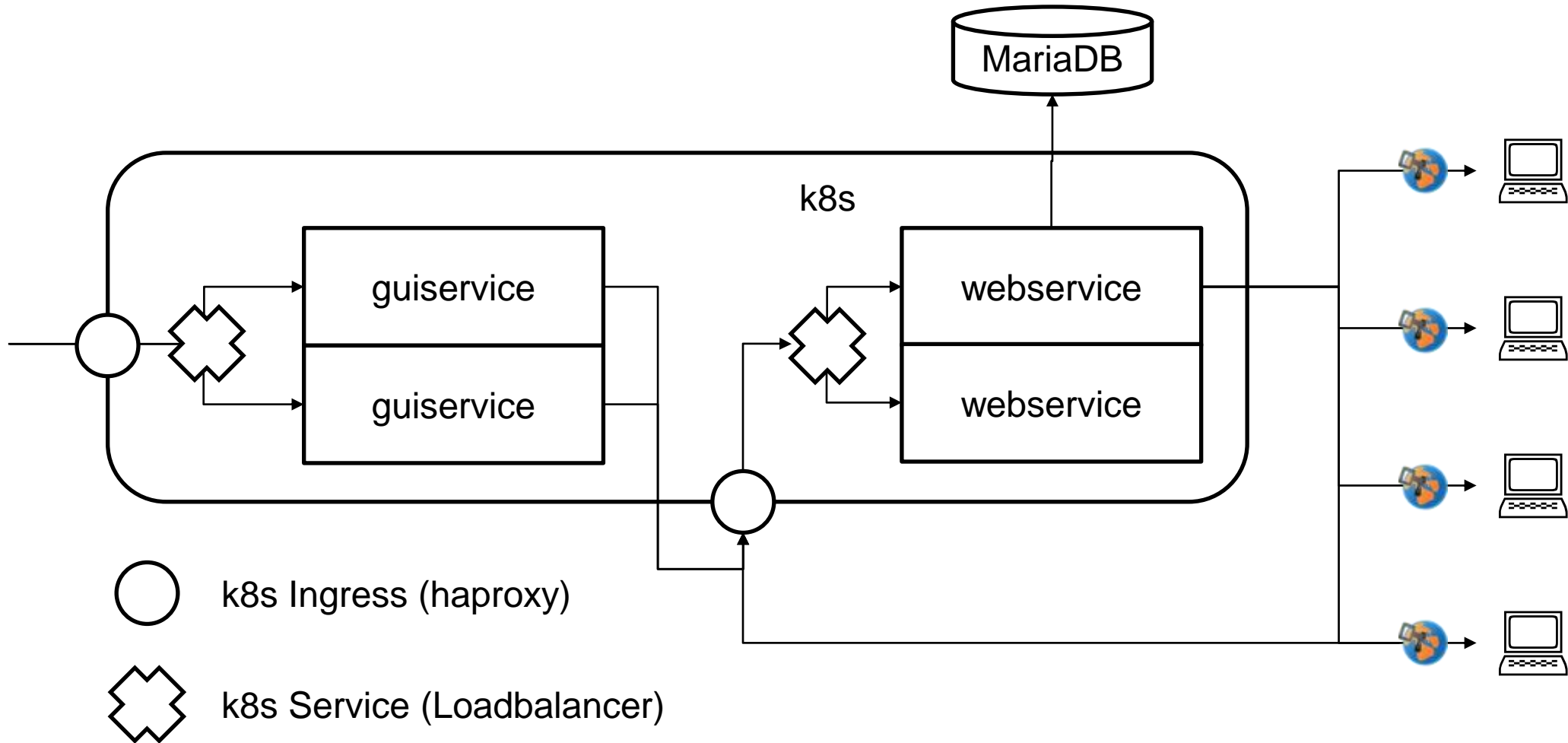
# SEB-Server



- klassische Server
- Docker (compose)
- Kubernetes



# SEB-Server

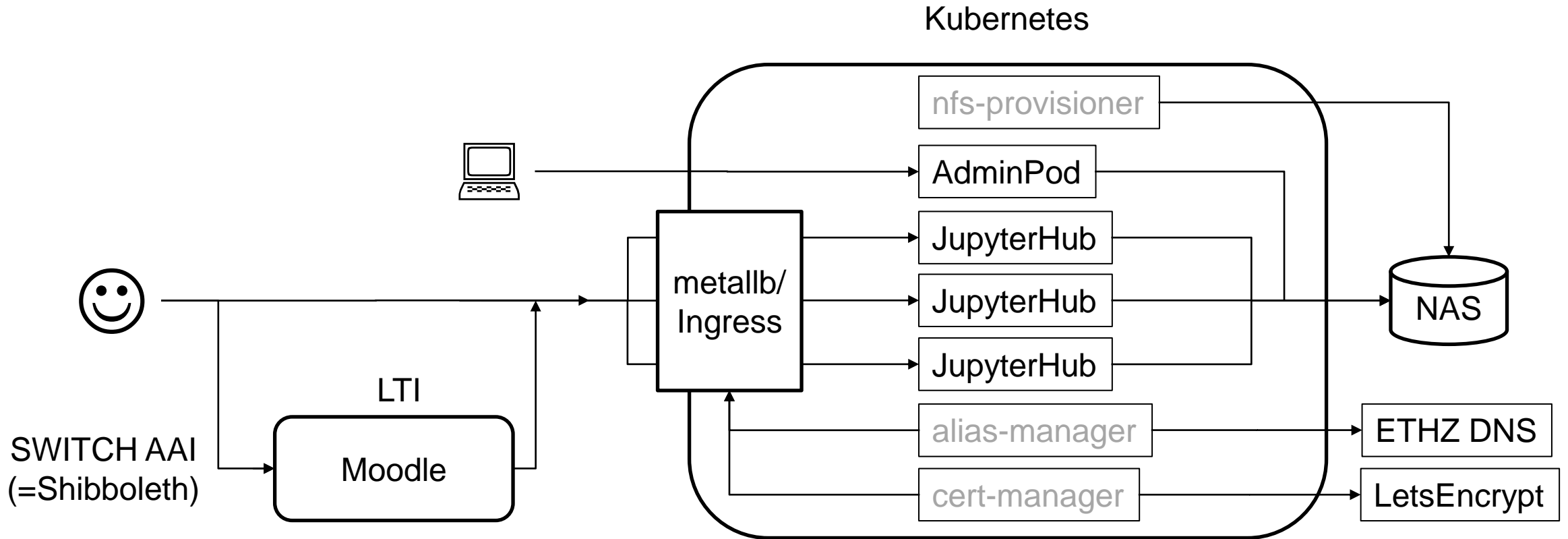


# JupyterHub

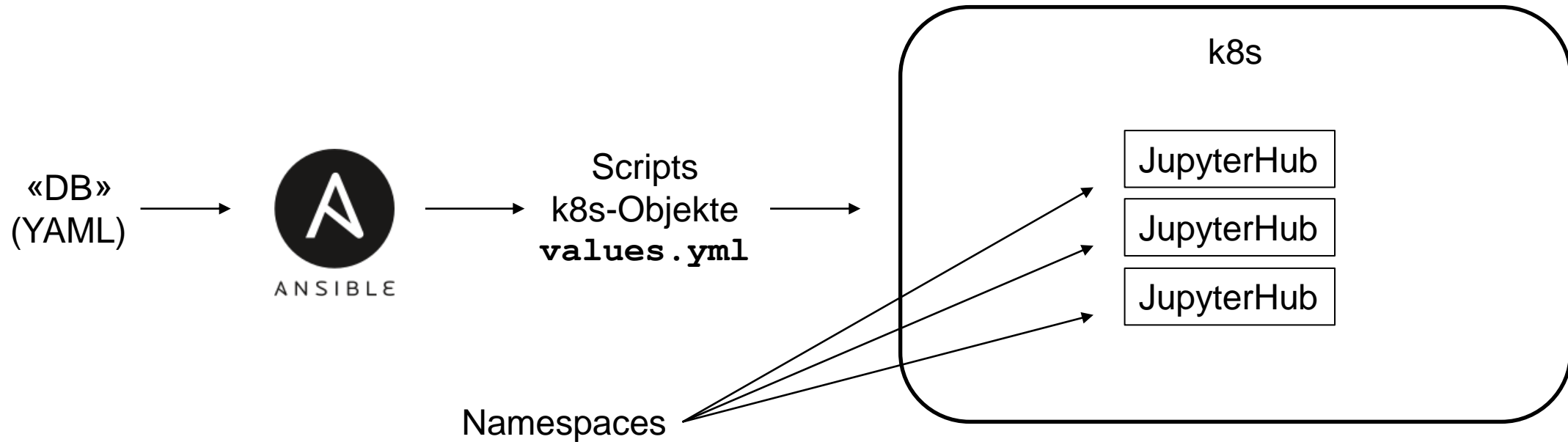


- JupyterHub: Server-basierte Variante von JupyterNotebook
- niederschwelliger Zugang zur Programmierung in Python, Julia und R
- eingesetzt im Rahmen der *Computational Competence Initiative*
  
- Weitere Infos und Download: <https://jupyter.org/hub>

# JupyterHub



# JupyterHub-Setup



- Isolation
- individuelle Konfigurationen (Mem-/CPU-Limits, Notebook-Image)
- vereinfachtes Lifecycle-Management

# Security

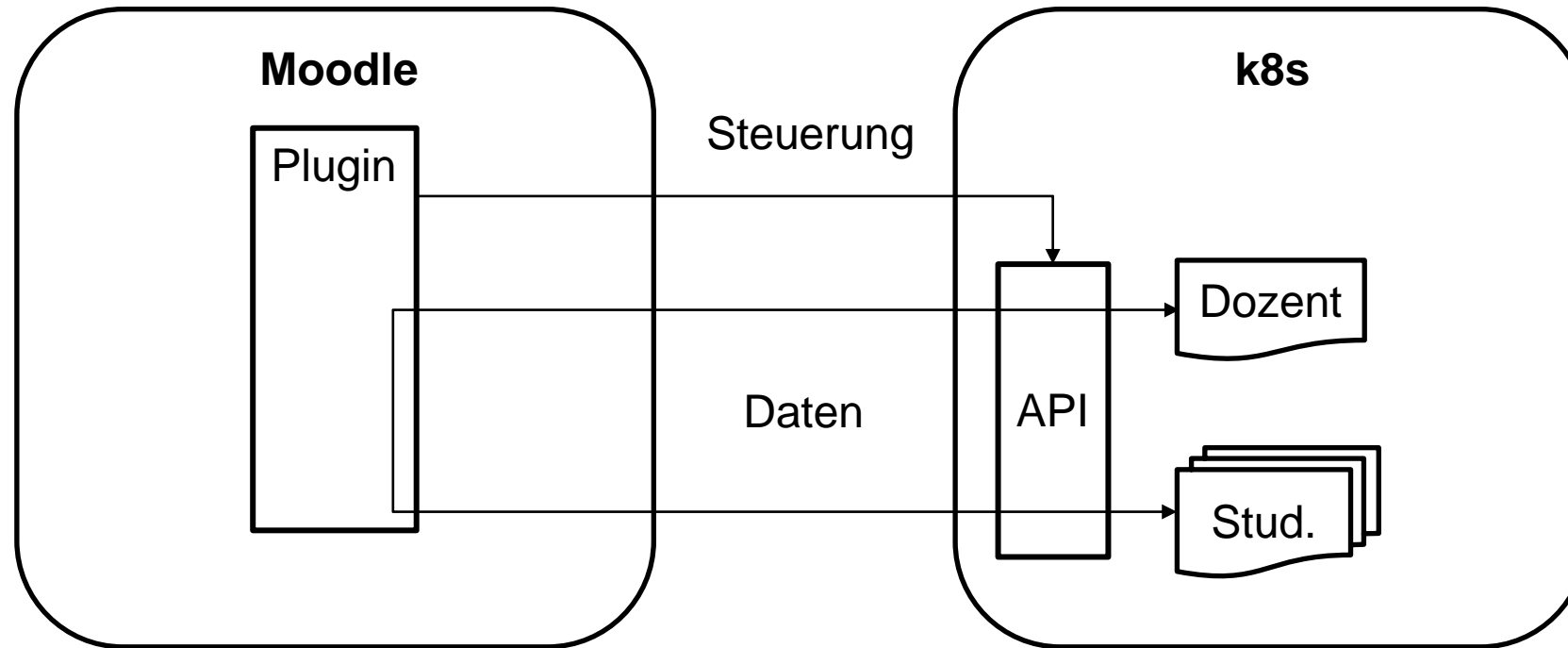
Studierende bekommen ein fast vollständiges Linux!

- Security-Scan der Notebook-Images (trivy)
- JupyterHub-Chart blockt bereits Zugriff auf Kubernetes-API
- Blacklist für *Kubernetes Network Policies* zum Schutz der lokalen FW-Zone

# Datentransfer: Problematik in Kubernetes

- Alle sind “jovian”, alle sind PID 1000  
→ Möglichkeiten für gemeinsame Shares begrenzt (readonly)
- Isolierung der Notebook-Container  
→ Transfer muss via “draussen” erfolgen

# Datentransfer: Moodle-Plugin und API-Server



# Datentransfer: laufende Arbeit

- Kooperation ETH Zürich und ETH Lausanne (EPFL)
- Plugin und API-Entwicklung gestartet durch EPFL: klassischer Server mit IAM-Anbindung
- API von uns angepasst an Kubernetes
- *JupyterHub* Assignment Moodle-Plugin: Auftragsarbeit von externer Firma (*enovation*)

Details dazu im Vortrag am Nachmittag.