

# Praxissemester

Enrico Stemmer

February 11, 2026

# Vorstellung

- **Name:** Enrico Stemmer
- **Beruf:** Dualer Student bei ARCA-Consult (Firmeninterne IT)
- **Zeitraum:** 01.08.2025 - 18.12.2025

# Vorstellung

- **Name:** Enrico Stemmer
- **Beruf:** Dualer Student bei ARCA-Consult (Firmeninterne IT)
- Zeitraum: 01.08.2025 - 18.12.2025
- Präsentation auf <https://github.com/h3rmt-thi/praxis>

- Spezialisiert auf IT-Sicherheit und Informationssicherheit
- Beratung zu Compliance und Normen (ISO 27001, etc.)
- Implementierung von Information Security Management Systems (*ISMS*)
- Identity Access Management (*IAM*) Lösungen
- Unterstützung für Unternehmen bei IT-Sicherheitsstrategien

# Alternative Infrastruktur zu Microsoft Cloud

- Kostensteigerungen bei Microsoft 365
- Viele "unnötige" Features
- Spezifische Workflows lassen sich nur schwer implementieren oder debuggen

# Alternative Infrastruktur zu Microsoft Cloud

- Ziel: Alternative Infrastruktur aufbauen, um Kosten zu senken und mehr Kontrolle zu haben
- Fokus auf Moderne Open-Source/Source-Available-Tools und Selbsthosting

# Alternative Infrastruktur zu Microsoft Cloud

- Ziel: Alternative Infrastruktur aufbauen, um Kosten zu senken und mehr Kontrolle zu haben
- Fokus auf Moderne Open-Source/Source-Available-Tools und Selbsthosting
- Viel mehr Infrastruktur notwendig als Anfangs gedacht (VPN, Metrics, Backups, DNS, etc.)

# Alternative Infrastruktur zu Microsoft Cloud



- Ausgangssituation: Dedicated Server mit Proxmox bei OVH
- Test-Umgebung mit Windows-VMs für IAM-Deployments
- Kein Monitoring, keine Backups, kein ReverseProxy, etc.
- Pritunl als VPN (OpenVPN/WireGuard Web UI)



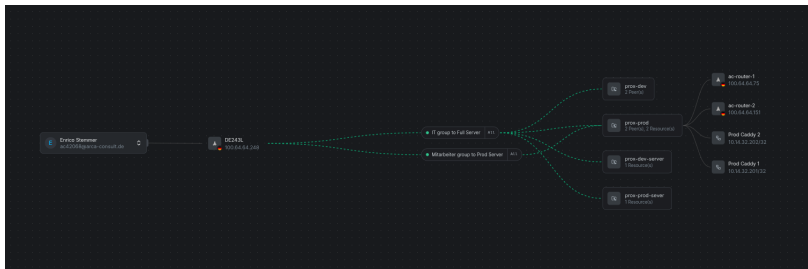
# Alternative Infrastruktur zu Microsoft Cloud

- Ausgangssituation: Dedicated Server mit Proxmox bei OVH
- Test-Umgebung mit Windows-VMs für IAM-Deployments
- Kein Monitoring, keine Backups, kein ReverseProxy, etc.
- Pritunl als VPN (OpenVPN/WireGuard Web UI)

- Ausgangssituation: Dedicated Server mit Proxmox bei OVH
  - Test-Umgebung mit Windows-VMs für IAM-Deployments
  - Kein Monitoring, keine Backups, kein ReverseProxy, etc.
  - Pritunl als VPN (OpenVPN/WireGuard Web UI)
- 
- Nächste Schritte: Weitere Tools evaluieren und Infrastruktur skalieren
  - Ersetzen von Tools durch bessere Alternativen

- **Keycloak** für Identity Access Management und SSO
- **NetBird** als moderne VPN-Alternative
  - SSO-Integration und benutzerfreundliche Authentifizierung
  - WireGuard-basiert für bessere Performance
  - Granulare Zugriffskontrolle und Netzwerk-Segmentierung

# Alternative Infrastruktur zu Microsoft Cloud

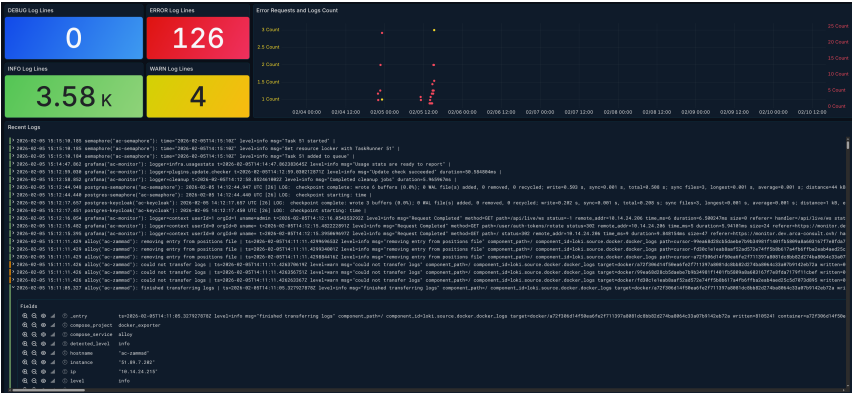


- **Keycloak** für Identity Access Management und SSO
- **NetBird** als moderne VPN-Alternative
  - SSO-Integration und benutzerfreundliche Authentifizierung
  - WireGuard-basiert für bessere Performance
  - Granulare Zugriffskontrolle und Netzwerk-Segmentierung
- **Caddy/Traefik** als Reverse Proxy und Load Balancer
- **Prometheus, Loki, Tempo + Grafana** für Monitoring und Logging

# Alternative Infrastruktur zu Microsoft Cloud



# Alternative Infrastruktur zu Microsoft Cloud



# Alternative Infrastruktur zu Microsoft Cloud





- **Keycloak** für Identity Access Management und SSO
- **NetBird** als moderne VPN-Alternative
  - SSO-Integration und benutzerfreundliche Authentifizierung
  - WireGuard-basiert für bessere Performance
  - Granulare Zugriffskontrolle und Netzwerk-Segmentierung
- **Caddy/Traefik** als Reverse Proxy und Load Balancer
- **Prometheus, Loki, Tempo + Grafana** für Monitoring und Logging
- **Ansible** für automatisierte, konsistente Deployments und Konfigurationsverwaltung
- **Semaphore UI** für vereinfachte Verwaltung und Ausführung von Ansible Playbooks

# Entwickeln eines Prototypen für die Verwaltung von Normen und Maßnahmen

- Früher: Next.js, Ohne REST-API, Direkter Datenbankzugriff im Backend
- Next.js verursachte Probleme:
  - langsamer Build
  - zu aggressives Caching
  - unpassende Defaults
  - schwer vorhersehbarer Controlflow

# Entwickeln eines Prototypen für die Verwaltung von Normen und Maßnahmen

- Neu: klare Trennung Frontend / Backend
- **Frontend:** React + TanStack Router + TanStack Query
- Vorteile:
  - Schnellerer Build (2+min → 10s)
  - bessere Caching-Strategien, mehr Kontrolle über Routing und Daten-Fetching
  - Wiederverwenden von vielen Komponenten des alten Prototyps
  - Automatisierte Generierung von API-Clients mit orval.js
- Nachteile:
  - Managen von API Specs und Code-Generation
  - Anpassung von Komponenten an neue Daten-Fetching-Strategien

# Entwickeln eines Prototypen für die Verwaltung von Normen und Maßnahmen

- **Backend:** JVM, Ktor als Framework, JOOQ als ORM
- Sprache: Kotlin statt Java
  - Kürzere, ausdrucksstarke Syntax, weniger Boilerplate
  - Null-Safety und Coroutines
  - Gute Interoperabilität mit Java-Ökosystem und Libraries
  - Flexible Programmierparadigmen

- **Datenzugriff:** verschiedene ORMs / Query-Builder ausprobiert
  - Kotlin Exposed
  - Komapper
  - JOOQ, Migrations mit Flyway
- **Observability:** OpenTelemetry-Integration (Logs, Traces, Metrics)  
End-to-end Tracing (Frontend → Backend → DB)
- **OpenAPI:** smiley4 ktor-openapi, Natives OpenAPI-Support in Ktor derzeit experimentell

Vielen Dank für eure Aufmerksamkeit!