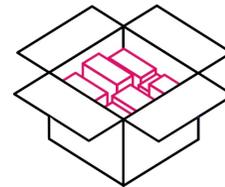


# Containerisierung und Virtualisierung

Hoai Viet Nguyen – TH Köln

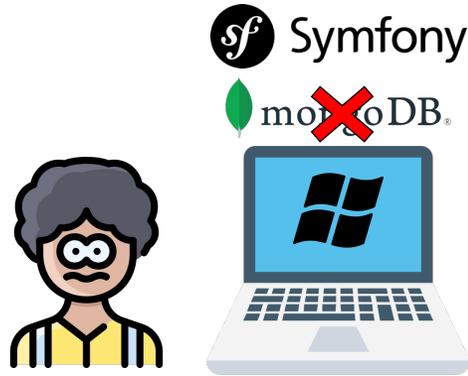
Technology  
Arts Sciences  
TH Köln



# Es war einmal im Jahr 2016 ...



Viet, Leiter der Softwareentwicklung



Einrichtung Arbeitsrechner für neuen Entwickler  
Installation MongoDB aufgrund inkompatibler Treiber nicht möglich  
Installation MacOS auf nicht Apple-Hardware ist verboten  
Verwendung von Linux war zu kompliziert



Kauf eines neuen MacBooks für Entwickler

# Herausforderungen beim Deployment

- **Installation von Abhängigkeiten wie z.B.**
  - Bibliotheken
  - Laufzeitumgebungen
  - Treibern
- **Ausführung von Code Dritter**
- **Verschiedene Betriebssysteme**
- **Inkompatibilität von Anwendungen und Betriebssystem**

# Virtualisierung

- **Ausführung Betriebssystem bzw. Software in virtuelle Maschine (VM)**
- **Vorteile**
  - Keine zusätzliche Hardware notwendig
  - Mehrere VM auf einer Hardware
  - Erstellung von Snapshots
  - Vermeidung von Betriebssysteminkompatibilitäten
- **Nachteile**
  - Höhere Hardwareleistung notwendig
  - Hoher Ressourcenverbrauch
  - Installation eines zusätzlichen und vollständigen Betriebssystems

# Container ...

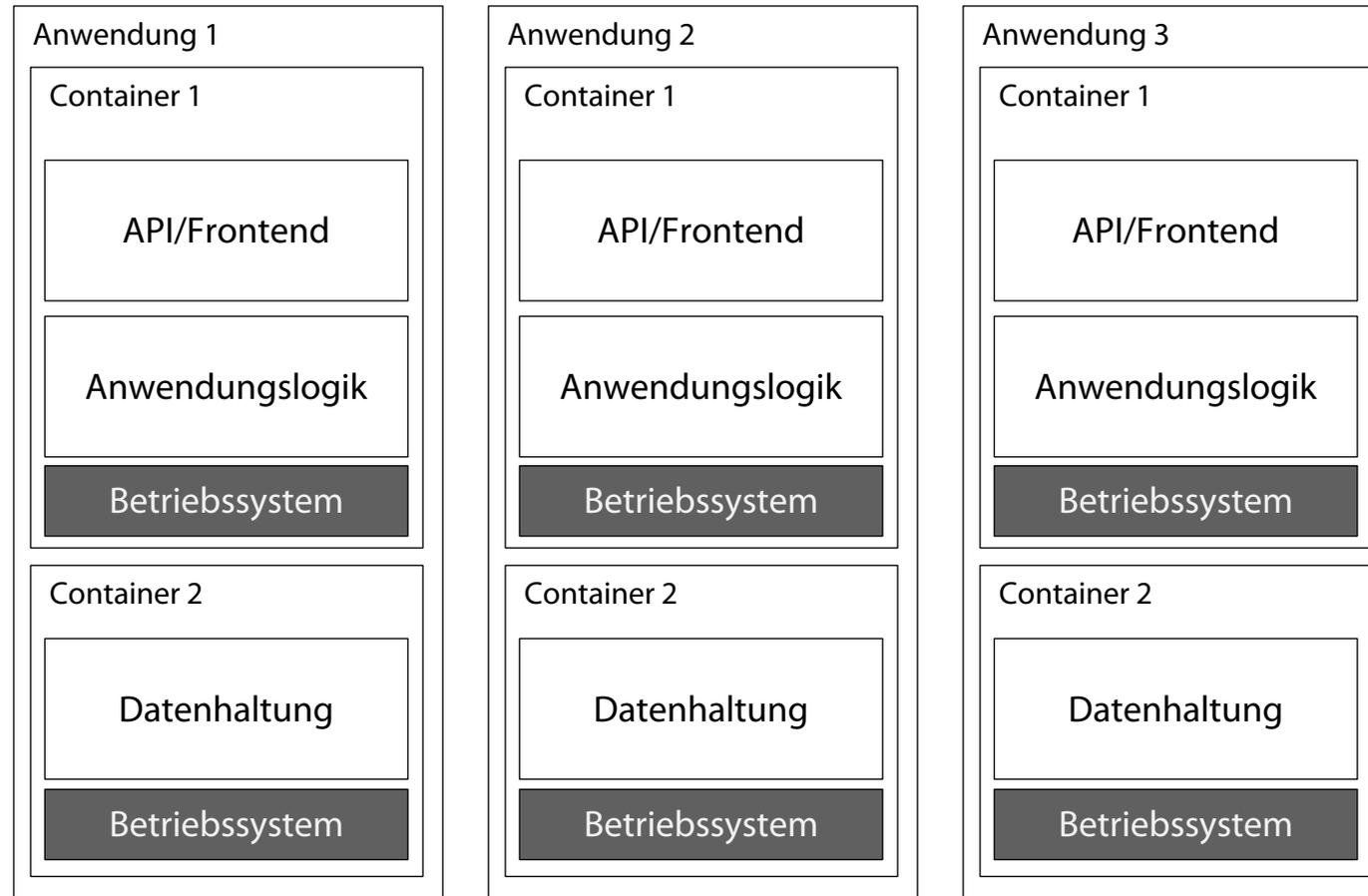
- bündeln Code mit benötigten Betriebssystembibliotheken und Abhängigkeiten
- Ergebnis ist eine ausführbare Datei, die als Container bezeichnet wird
- laufen konsistent auf jeder Infrastruktur
- portabler und ressourceneffizienter als konventionelle Virtualisierung
- haben sich als de-facto Standard moderner cloudnativer Anwendungen etabliert

Quelle: <https://www.ibm.com/de-de/topics/containerization>

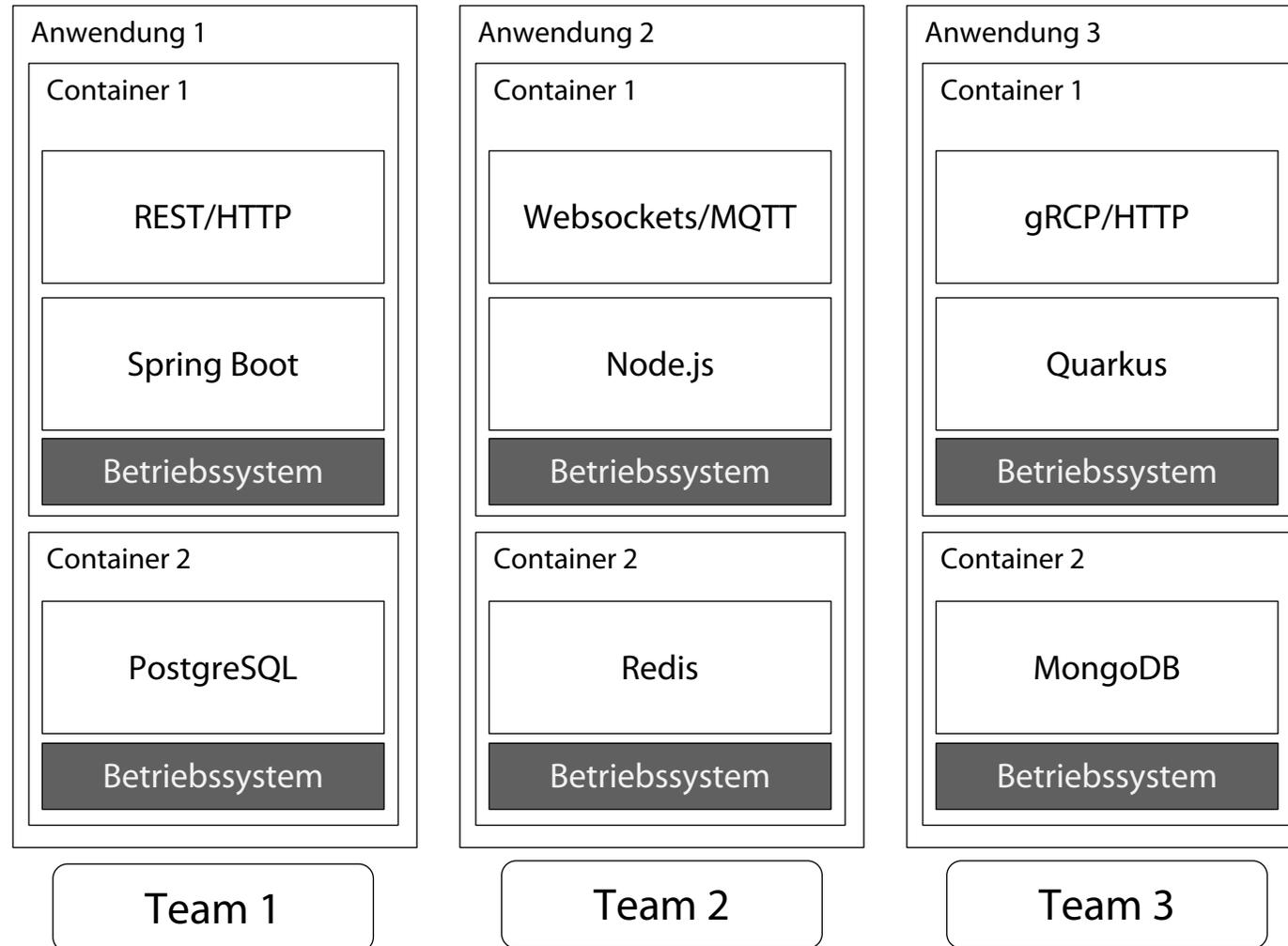
# Vorteile Containerisierung

- **Ressourceneffizient:** Container verbrauchen weniger Ressourcen als konventionelle Virtualisierungstechnologien
- **Portabilität:** Verschiebung auf andere Hosts und Cloud-Umgebung mit wenig Aufwand
- **Automatisiertes Deployment:** Betriebssystem, Laufzeitumgebungen, Bibliotheken werden automatisch installiert und konfiguriert
- **Skalierung:** Bei hoher Last können weitere Container automatisiert deployed werden

# Container in Microservices Architektur



# Container in Microservices Architektur mit Implementierungen



# Containerisierungstechnologien

- **Container Engines (Auszug)**
  - Docker
  - Podman
  - Vagrant
  - Buildah
- **Containerverwaltung (Auszug)**
  - Kubernetes
  - Coolify
  - Portainer
  - Nomad
  - Docker Swarm

# Zusammenfassung

- **Virtualisierung**
  - Plattformunabhängige Ausführung von Betriebssystem bzw. Software in VM
  - Keine zusätzliche Hardware notwendig
- **Containerisierung**
  - eine Form der Virtualisierung
  - weniger Ressourcenverbrauch als konventionelle Virtualisierung
  - fester Bestandteil moderner Softwaresysteme
  - ermöglichen plattformunabhängiges automatisiertes Deployment
  - vermeiden (im idealen Fall) Betriebssystem-Inkompatibilitäten

# Lernzielkontrolle

- Was sind die Herausforderungen beim Deployment?
- Was ist der Unterschied zwischen Containerisierung und Virtualisierung?
- Was sind die Vorteile von Containerisierung?
- Welche Containerisierungstechnologien gibt es?

# Danksagung

Die verwendeten Icons stehen unter der Flaticon Basic Lizenz (flaticon.com) und stammen von

- Freepik
- Atif
- juicy fish
- BZZRINCANTATION