

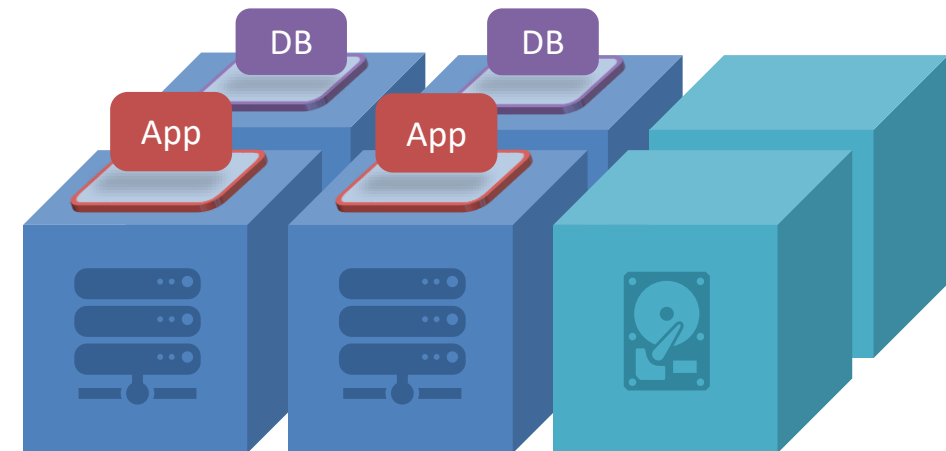
Cloud Native Plattformen

FWP Aktuelle Technologien zur Entwicklung
verteilter Java-Anwendungen

KLASSIFIZIERUNG VON CLOUD-DIENSTEN

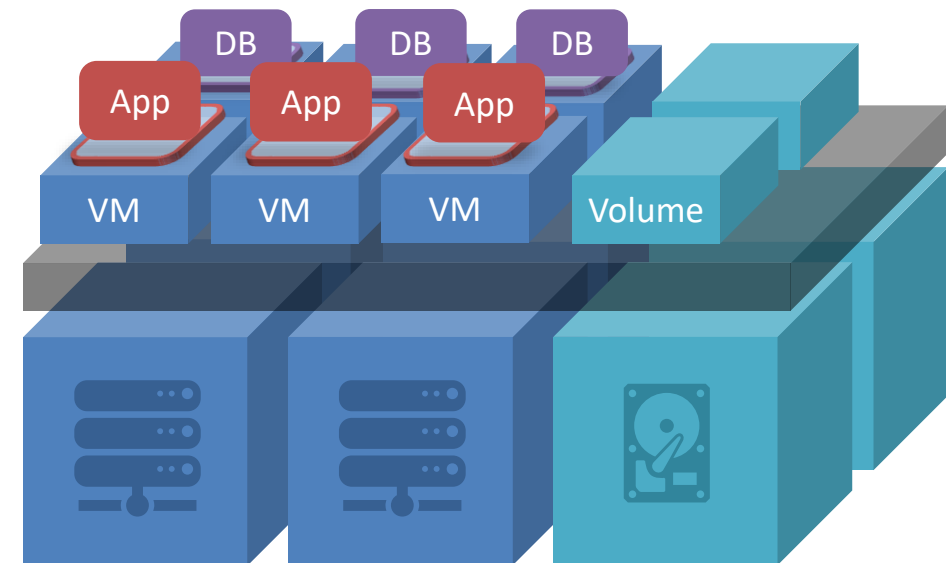
Wie alles begann mit der Infrastruktur

- Applikationen und Datenbanken u.ä. werden auf Bare Metal betrieben
- Rechner und Storage müssen umständlich über lange Zeit provisioniert werden
- Netzwerke und Netzwerkkomponenten müssen umständlich über lange Zeit provisioniert werden
- Beahlt wird alles was in Betrieb ist (auch wenn es nicht benötigt wird)
- Benötigte Laufzeitumgebung muss auf die Rechner gebracht werden



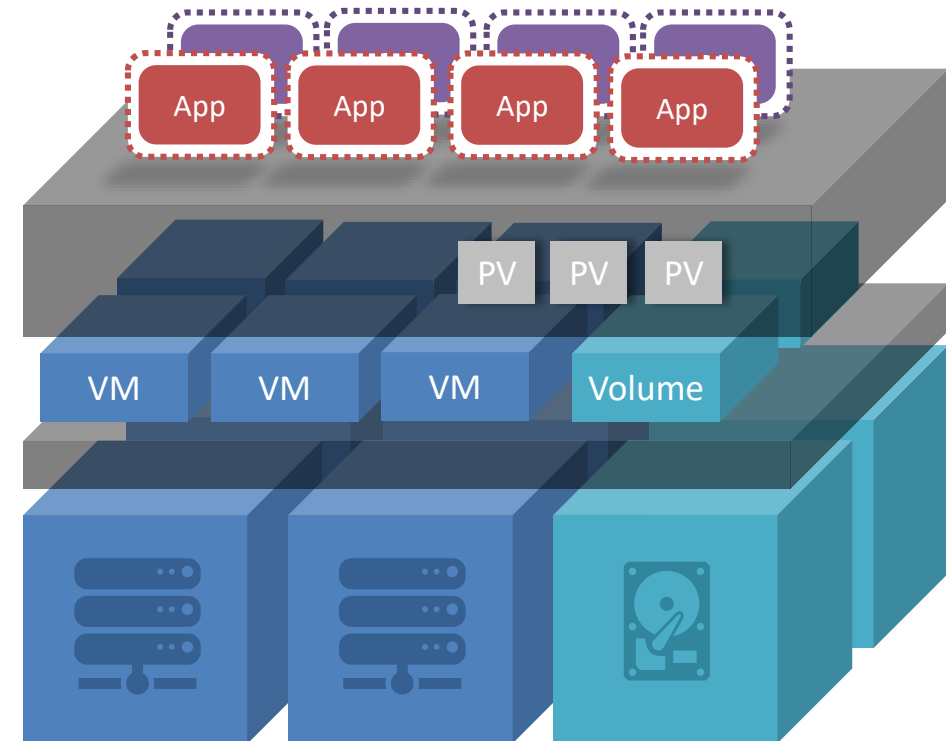
Infrastructure as a Service (IaaS)

- Applikationen und Datenbanken werden auf VMs betrieben
- Jede VM hat frei wählbares Image (Betriebssystem) und frei wählbare Ressourcen (RAM/CPU + Volumes)
- Kurzfristige automatisierbare Provisionierung von VMs, Netzwerken und Netzwerk-komponenten
- Ausstattung von VMs kann nachträglich angepasst werden
- Beahlt wird nur das was tatsächlich benutzt worden ist (CPU + Storage + Traffic / Zeit)
- Benötigte Laufzeitumgebung muss auf die VMs gebracht werden



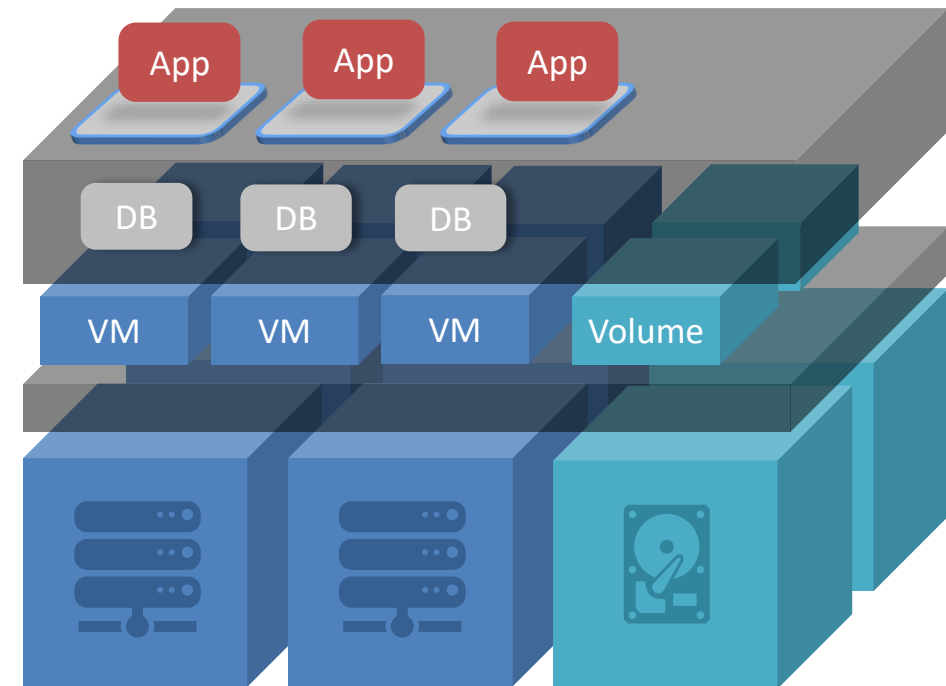
Container as a Service (CaaS)

- Applikationen und Datenbanken werden als Container im Cluster betrieben
- Benötigte Laufzeitumgebung zusammen mit Applikation in Container verpackt
- Praktisch alle Applikations- und Diensttypen durch Containerisierung möglich
- Unterstützung von persistenten Volumes
- Benötigte Ressourcen (RAM/CPU + Volumes) und Services werden beim Deployment festgelegt
- Routing zum Container wird von der Plattform zur Verfügung gestellt
- Plattform sorgt für Verfügbarkeit und Skalierbarkeit



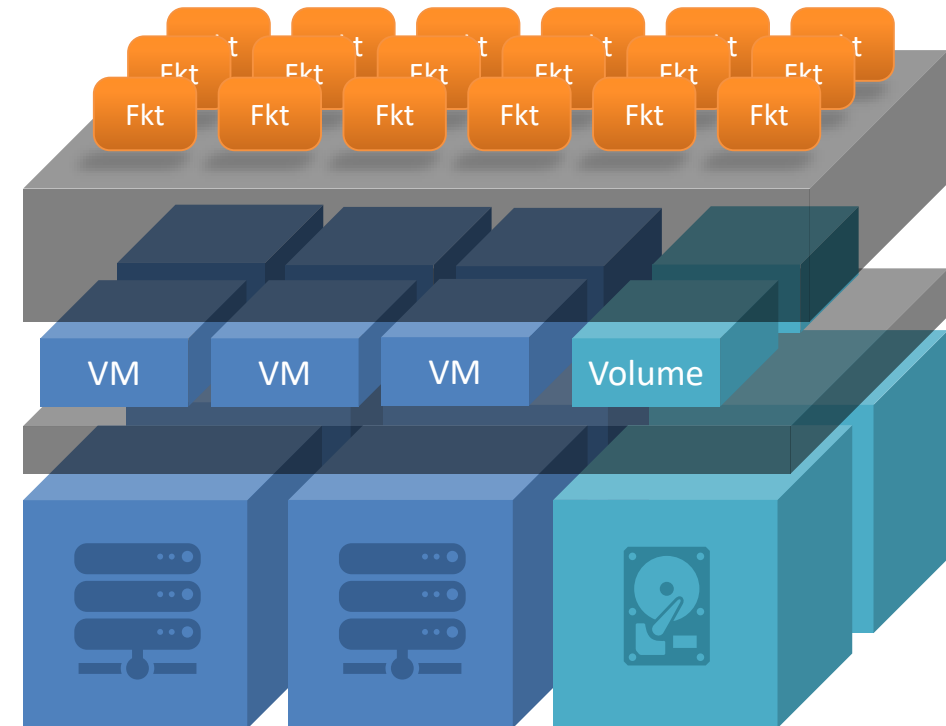
Platform as a Service (PaaS)

- Betrieb von Applikationen direkt auf der Plattform in bereitgestellter Laufzeitumgebung
- Datenbanken und ähnliche Services sind provisionierbare Bestandteile der Plattform
- Externe Services über Service Broker
- Benötigte Ressourcen und Services werden beim Deployment festgelegt
- Plattform stellt Routing zur Verfügung
- Plattform sorgt für Verfügbarkeit und Skalierbarkeit
- Mögliche Applikationstypen von den angebotenen Laufzeitumgebungen abhängig
- Lizenzkosten für Plattform müssen berücksichtigt werden



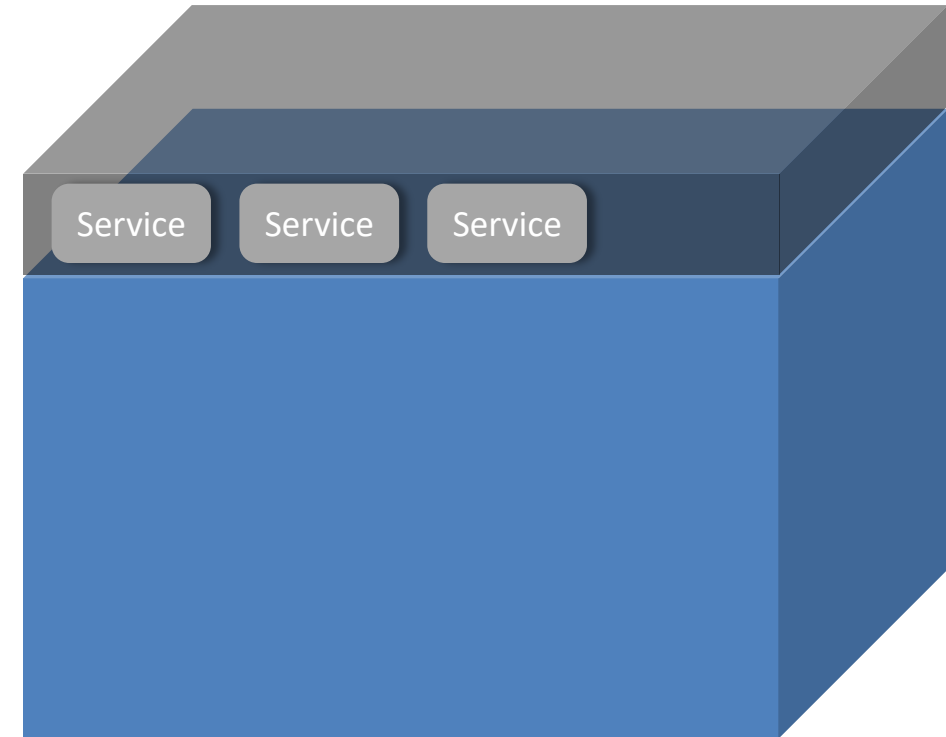
Function as a Service (FaaS)

- Jüngstes Mitglied der *aaS-Familie
- Funktionalität wird eigenständig bereitgestellt und betrieben
- Extrem kurze Bereitstellungszeiten
- Kein Servermanagement, keine Kapazitätsplanung
- Beahlt wird pro Aufruf und verbrauchte Rechenzeit
- PlattfRouting zur Funktion wird von der Plattform zur Verfügung gestellt
- Plattform sorgt für Verfügbarkeit und Skalierbarkeit

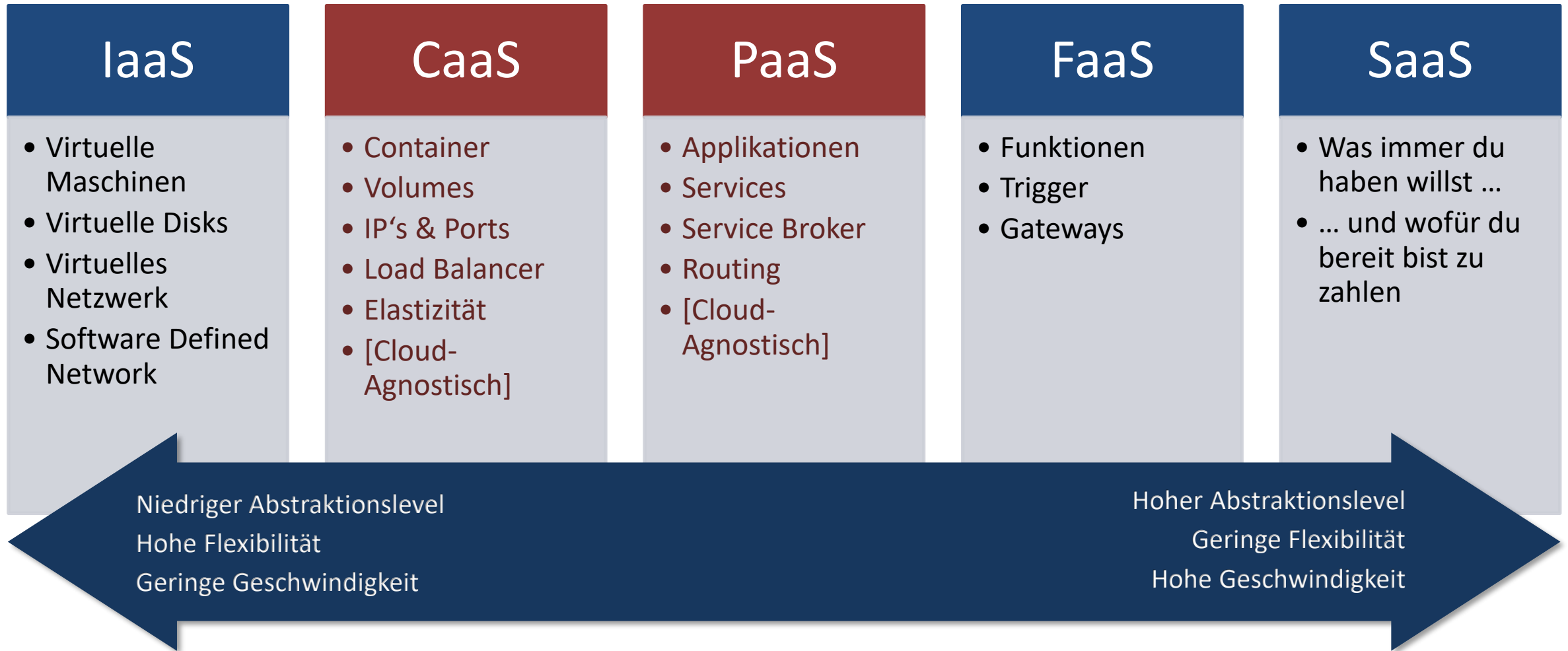


Software as a Service (SaaS)

- Software und IT-Infrastruktur werden von einem Anbieter als Gesamtpaket zur Verfügung gestellt
- Kunden nutzt die bereitgestellte Funktionalität als Dienst
- Beahlt wird pro genutztem Dienst
- Keine Entwicklungskosten auf Kundenseite
- Keine Betriebs- oder Administrationskosten auf Kundenseite



Leistungsmerkmale Cloud-Plattformen



Produkte und Anbieter für Cloud-Plattformen

IaaS

- OpenStack
- vmware vSphere
- Azure Stack

- AWS EC2/VPC
- Google GCE
- Azure VMs

CaaS

- Kubernetes
- DC/OS
- Docker EE
- Pivotal PCS

- AWS ECS
- AWS EKS
- Google GKE
- Azure ACS

PaaS

- CloudFoundry
- Pivotal PCF
- OpenShift Origin
- RedHat OpenShift

- AWS Elastic Beanstalk
- Google App Engine
- Azure Web Apps
- Heroku
- Jelastic

FaaS

- OpenWhisk
- Fission
- IronFunctions

- AWS Lambda
- Google GCF
- Azure Functions

SaaS

- SAP
- Oracle
- Salesforce

PUBLIC CLOUD PROVIDER

AWS und Azure sind die wichtigsten Provider



[Gartner: Magic Quadrant for Cloud Infrastructure as a Service, Worldwide, 15.06.2017](#)

Amazon Web Services (AWS)



<https://aws.amazon.com/de/>

- Seit 2006 Pionier unter den Public Cloud Providern
- Breitesten Dienstangebot mit deutlichem Technologievorsprung
- Mit 8 Regionen weltweit vertreten
- 3 Rechenzentren auf deutschem Boden (Frankfurt/Main)
- Alle Dienste sind nach dem **Cloud Computing Compliance Controls Catalog (C5)** des **BSI** zertifiziert
- Zugang über AWS Console oder AWS CLI oder AWS SDK

Microsoft Azure

Microsoft Azure

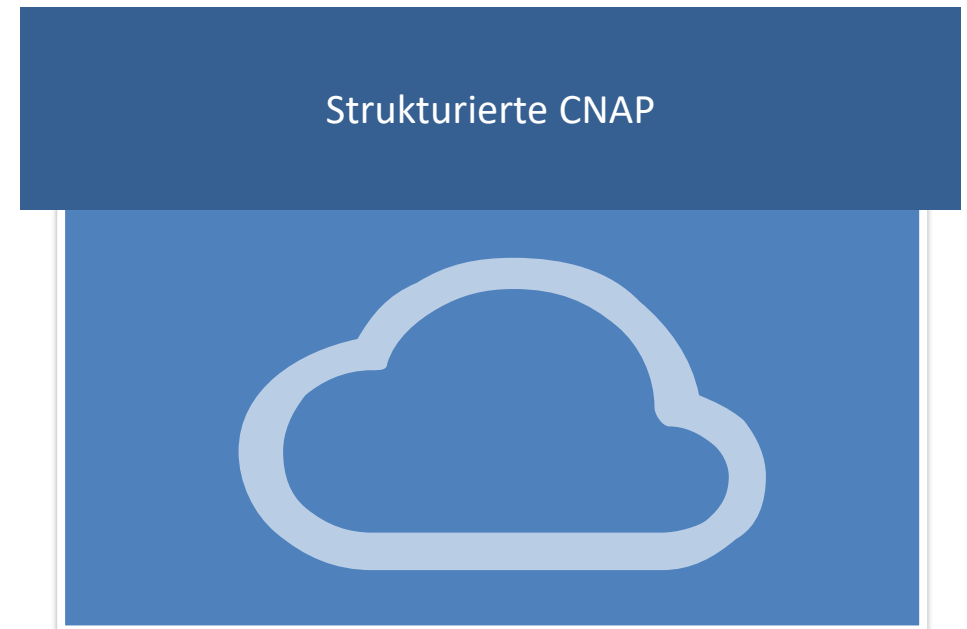
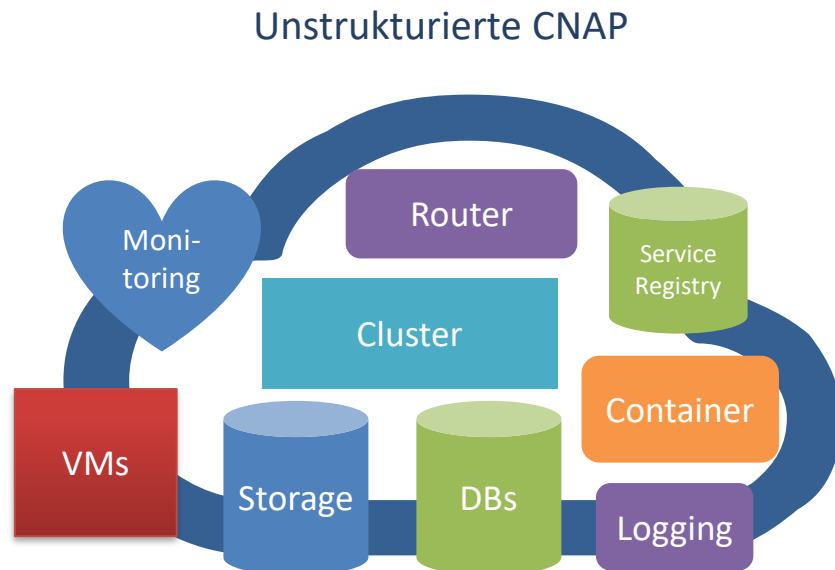
<https://azure.microsoft.com/de-de//>

- Hartnäckigster Verfolger von AWS fokussiert auf Innovation
- Leistungsfähige Plattform mit steigender New Feature Velocity
- Strategischer Cloud Provider für Kunden mit Microsoft Technologien und Produkten im Einsatz
- Kann mit Azure Stack auch on-premise aufgebaut werden
- Hervorragende weltweite Verteilung mit 42 Regionen
- Treuhänder Telekom betreibt deutsche Rechenzentren

CLOUD NATIVE APPLICATION PLATFORMS (CNAP)

Cloud Native Application Platforms (CNAP)

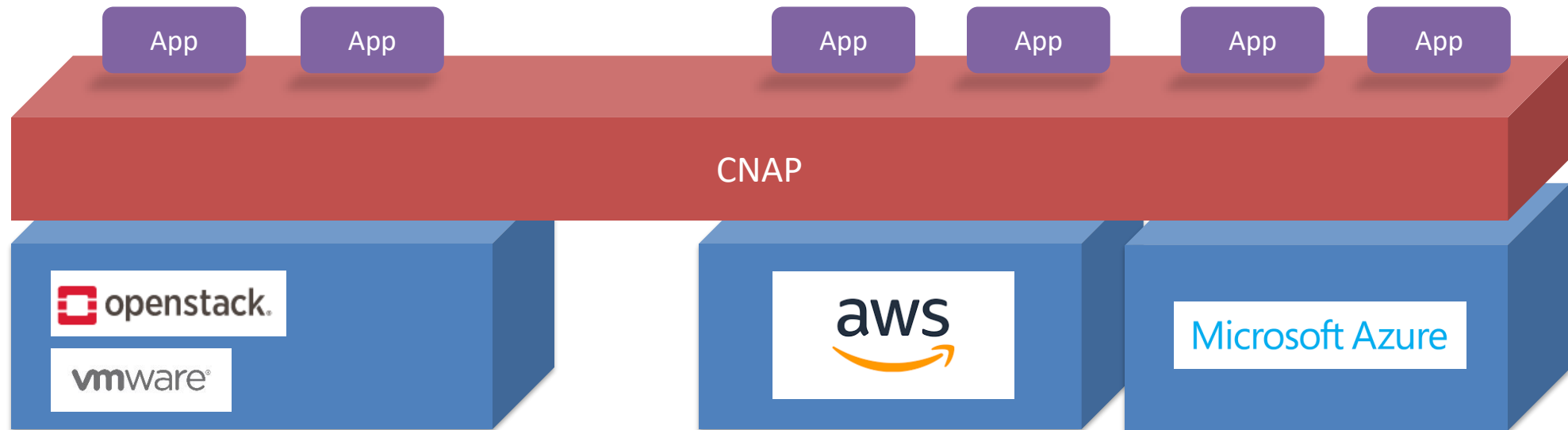
- Cloud Native Application Plattformen kommen in zwei Arten
 - Unstrukturiert (Bring and Build Your Own)
 - Strukturiert (Rundum-Sorglos Paket)



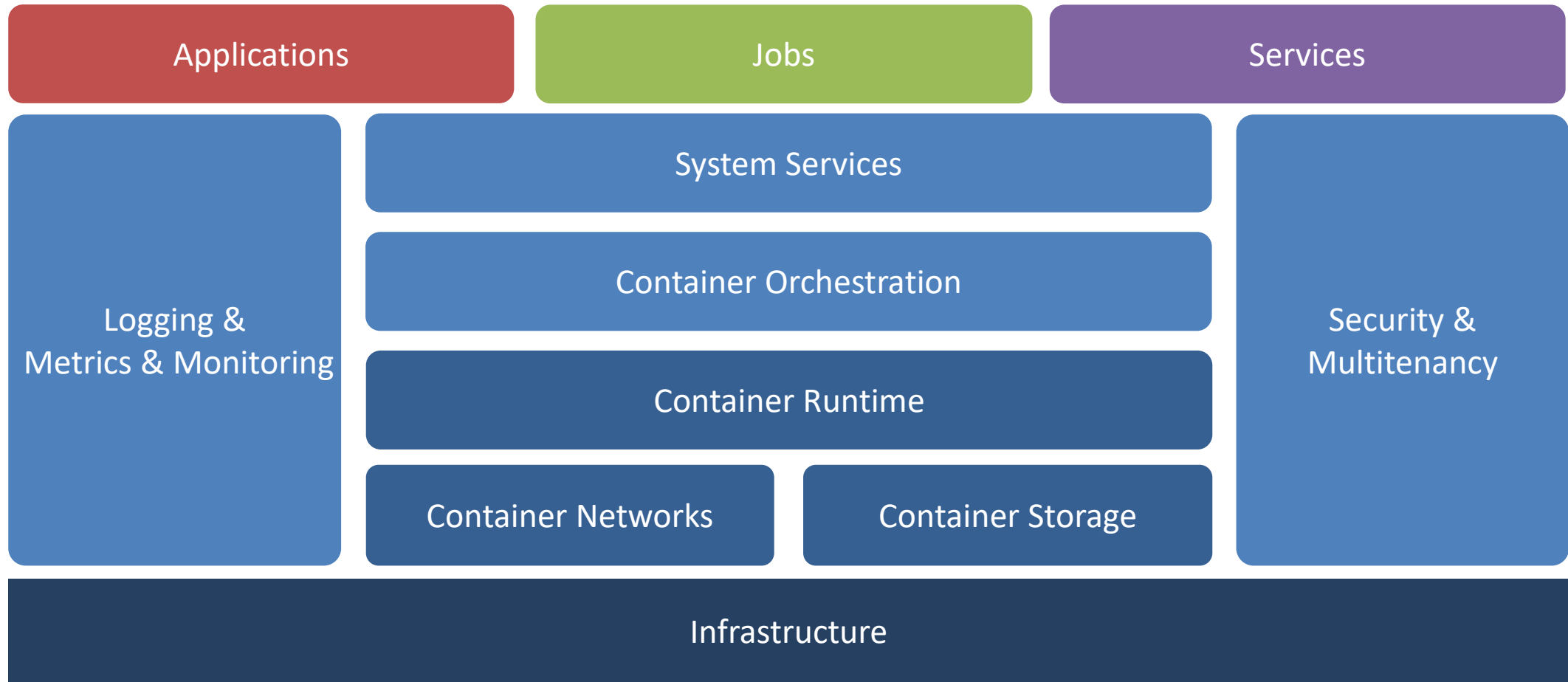
CNAPs schaffen Unabhängigkeit

On-Premise / Private Cloud

Off-Premise / Public Cloud



Allgemeiner Aufbau von CNAPs



Karl Isenberg, Mesosphere Container Orchestration Wars (2017) <https://youtu.be/NRZ6N4e-Mko>

Kubernetes



<https://kubernetes.io//>

- Platzhirsch auf dem Sektor der Container-Cluster (CaaS)
- Basiert auf Google's Erfahrungen im Betrieb von produktiven Workloads aus den letzten 15 Jahren
- Fokussiert auf den Betrieb von Docker-Containern

Wesentliche Features

- Automatisches Placement von Container mit dem Ziel der optimalen Ressourcenauslastung
- Horizontale Skalierung
- Automatisierte Rollouts und Rollbacks mit Zero Downtime
- Storage Orchestrierung
- Selbstheilender Betrieb von Containern
- Service Discovery und Load Balancing
- Secret und Configuration Management
- Batch-Ausführungen
- Federation

Red Hat OpenShift



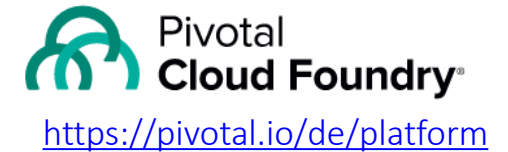
<https://www.openshift.com/>

- Basiert auf **Kubernetes** als Container Cluster Manager
- Bei allen gängigen Public Cloud Providern und On-Premise betreibbar
- Offiziell PaaS, tatsächlich aber CaaS mit netter UI
- Deployment von Docker-Containern und Applikationen

Wesentliche Features

- Orchestrierung von Containern mit Kubernetes
- Container Storage mit QoS Labels
- Unterstützung vieler Sprachen mit polyglotter Persistenz
- Automatisierung von Builds, Deployments, Skalierung, Health Management
- Erweiterte Mandantenfähigkeit
- Selbstbedienungs-Plattform
- Source To Image (S2I)

Pivotal Cloud Foundry



- Kommerzielle PaaS Plattform von Pivotal auf Basis von Cloud Foundry
- Bei allen gängigen Public Cloud Providern und On-Premise betreibbar

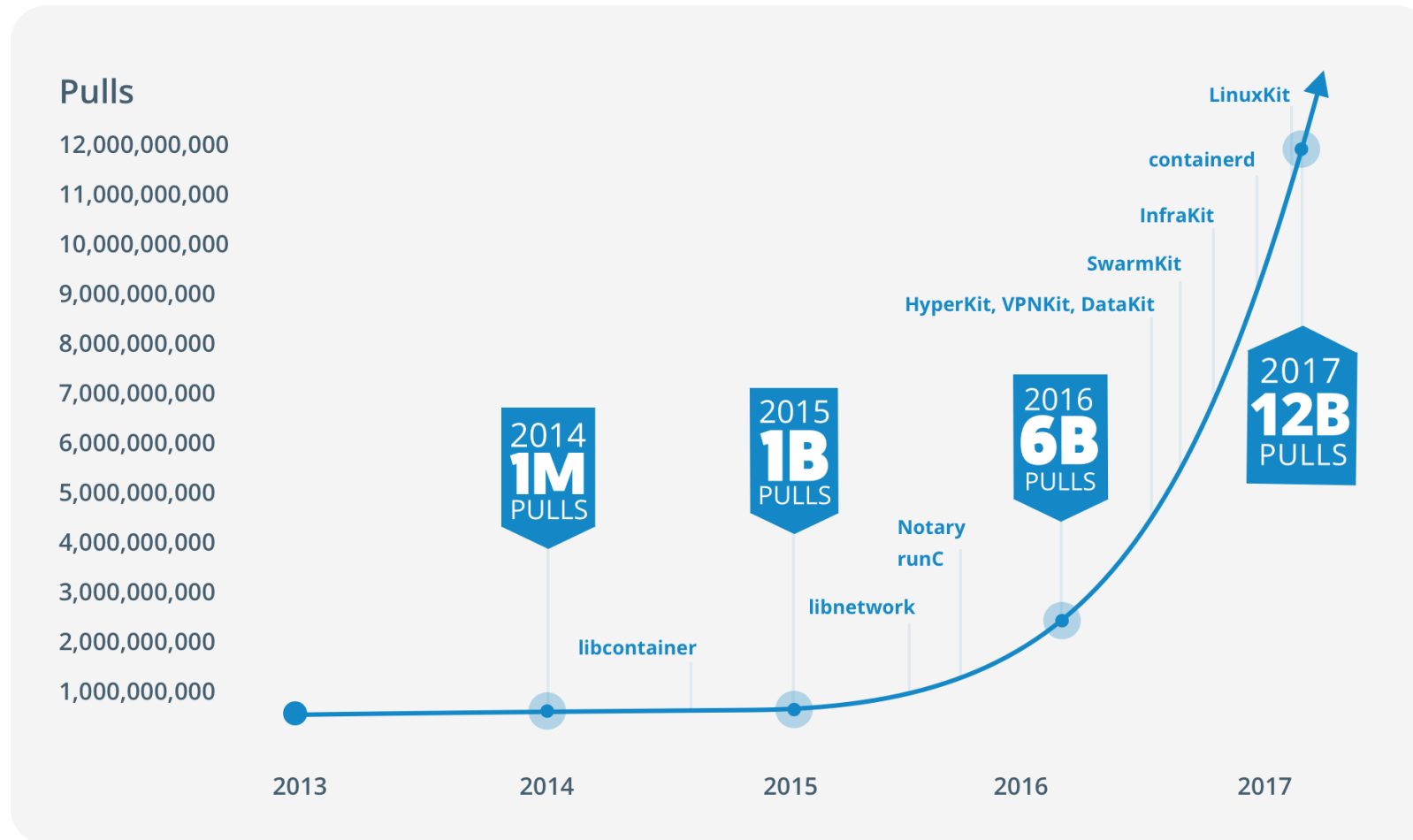


Wesentliche Features

- Gefühltes größtes Leistungsspektrum aller CNAPs
- Schwerpunkt auf Deployment von Apps (kann aber auch Docker Container)
- Beste Runtime für Spring und Spring Boot
- Schlüsselfertige Microservices mit Spring Cloud
- Reichhaltiger Marktplatz für Services
- Service Broker
- Umfangreiches Monitoring
- Automatische horizontale Skalierung (Autoscaler)
- Volume Services für Stateful Apps
- Unterstützung für viele Routing Protokolle

CONTAINERISIERUNG MIT DOCKER

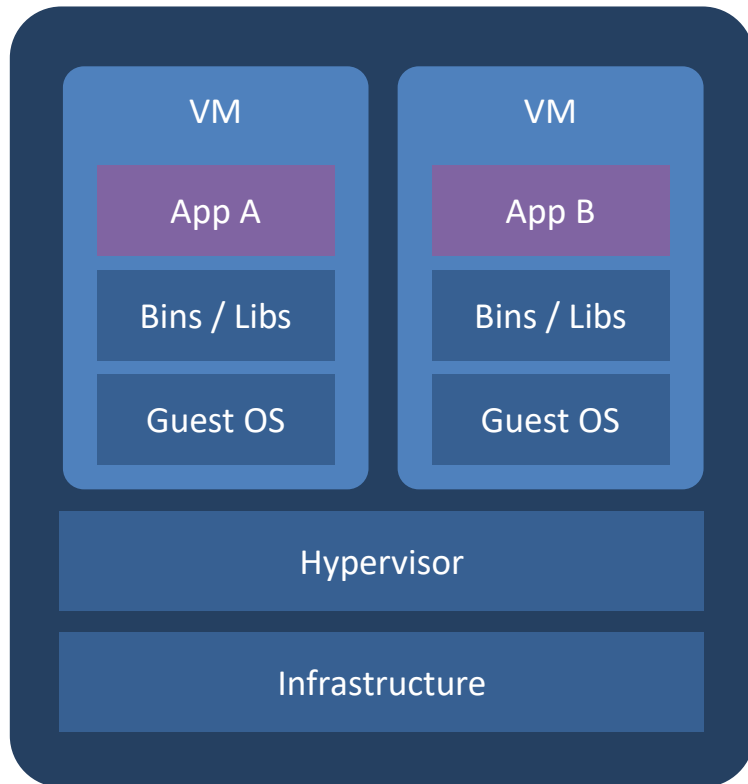
Docker hat die Welt verändert



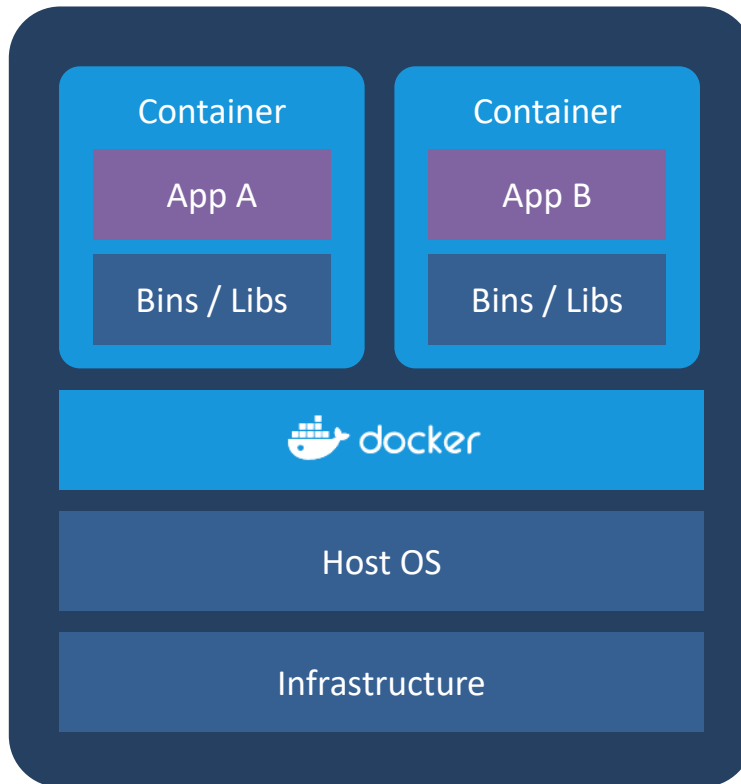
<https://www.docker.com/what-container>

Docker vs VMs

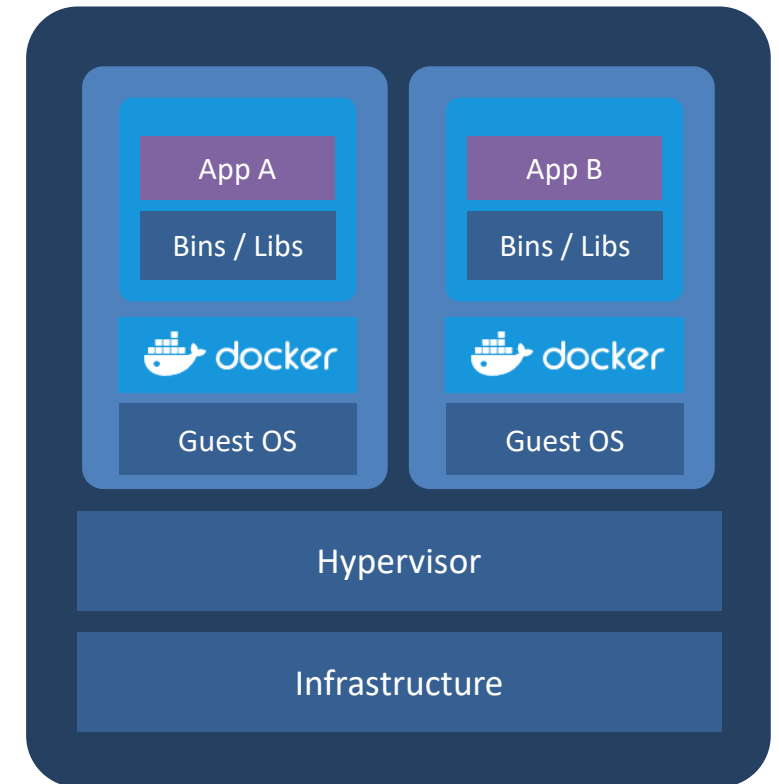
VM



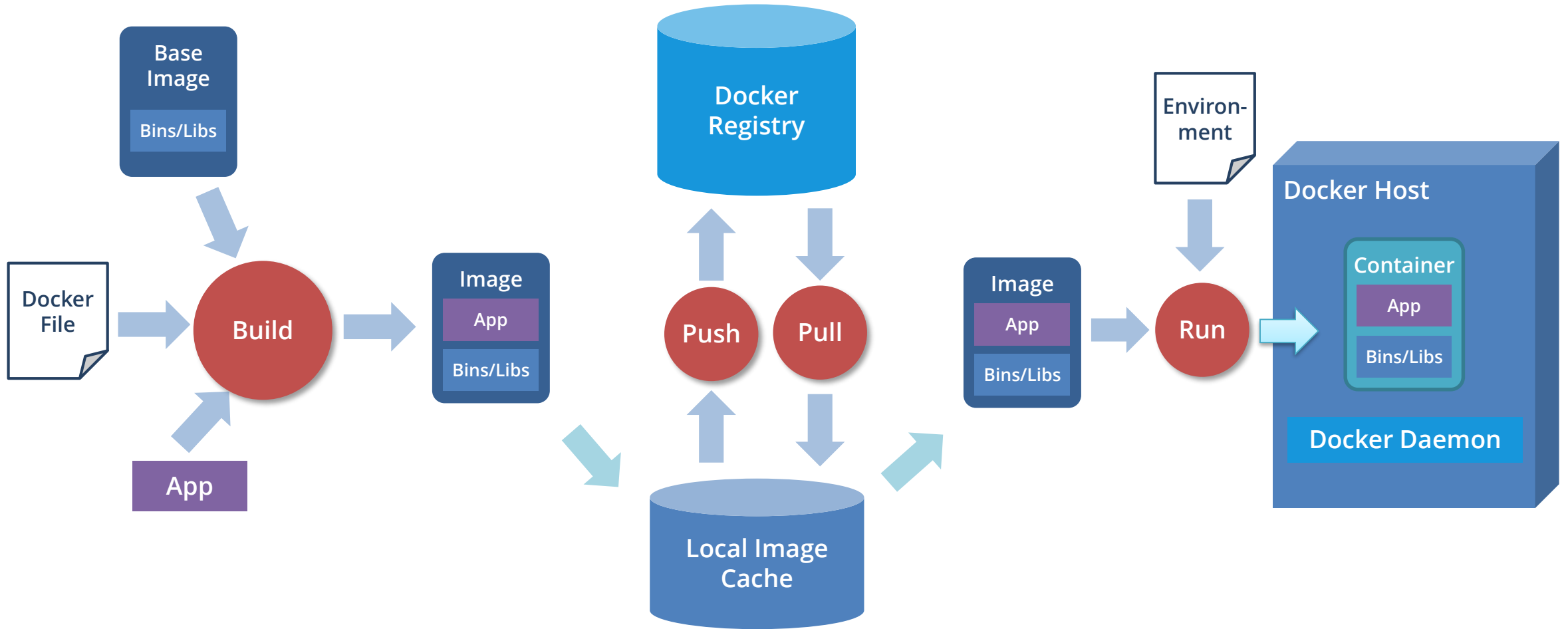
Docker



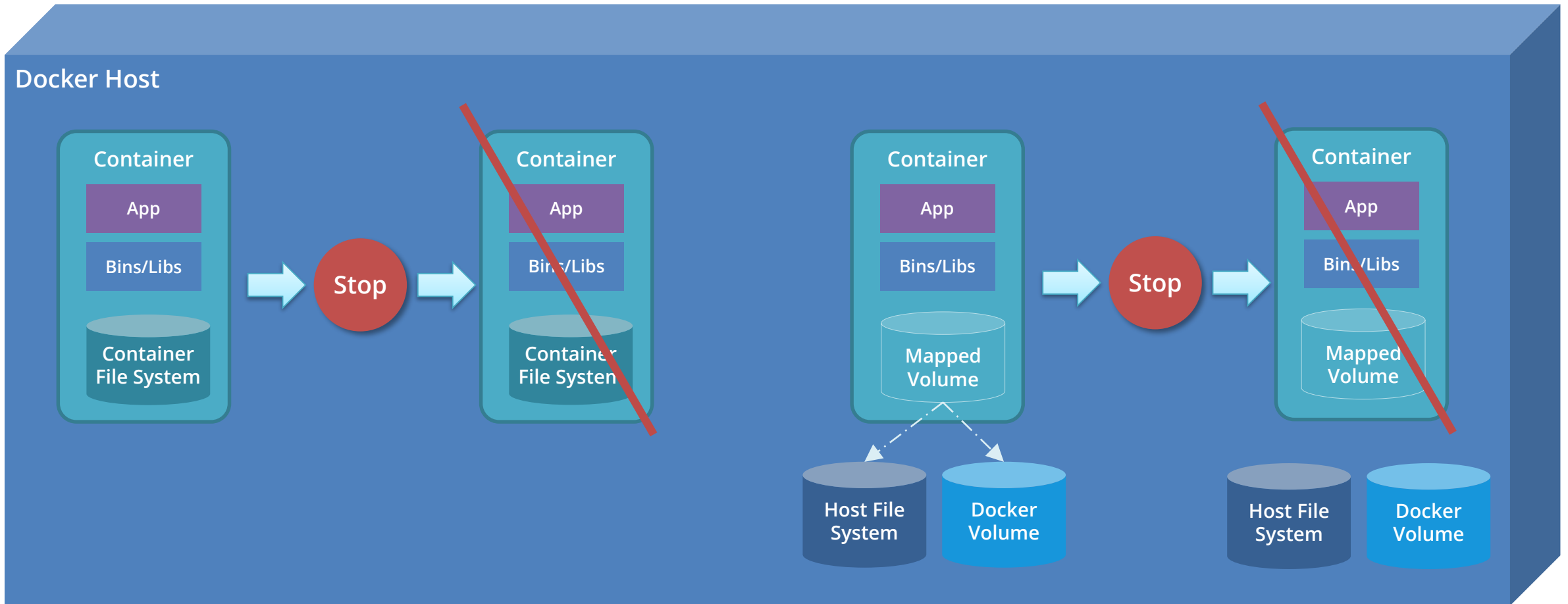
VM + Docker



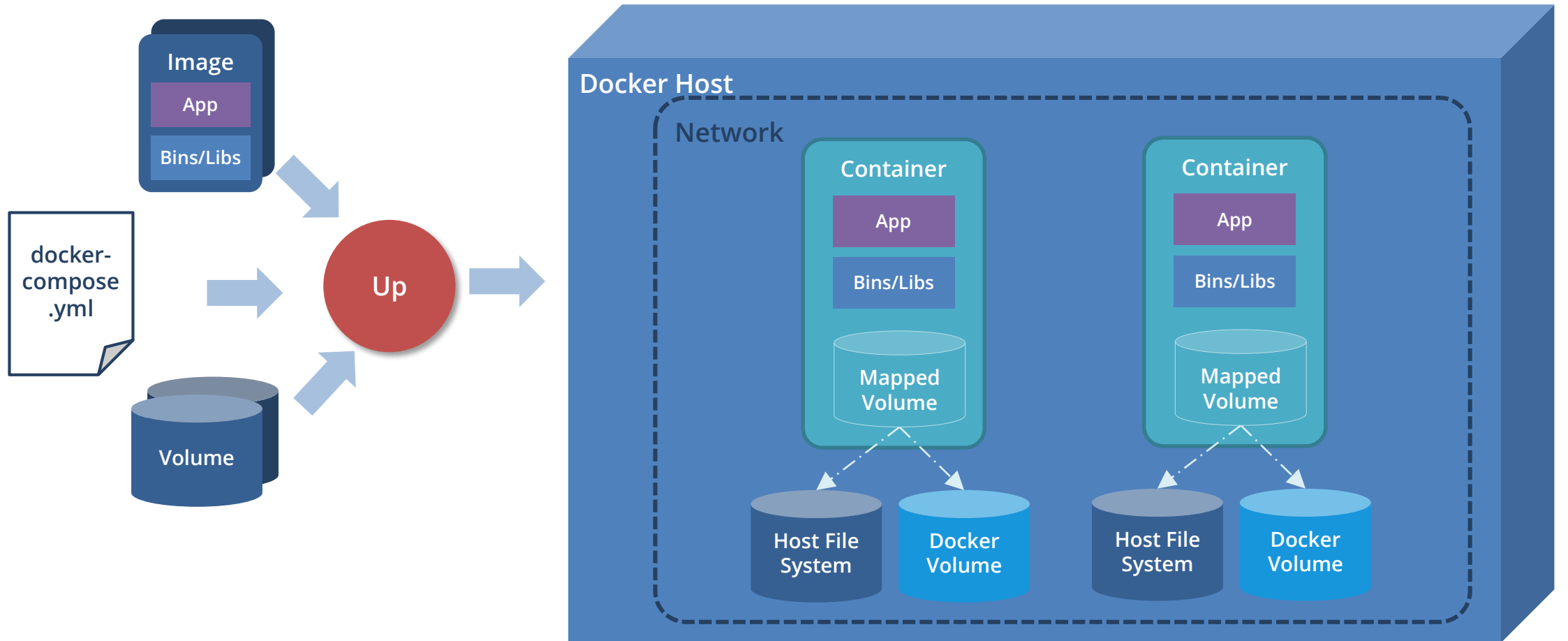
Docker Files, Images & Container



Docker Volumes



Docker Compose



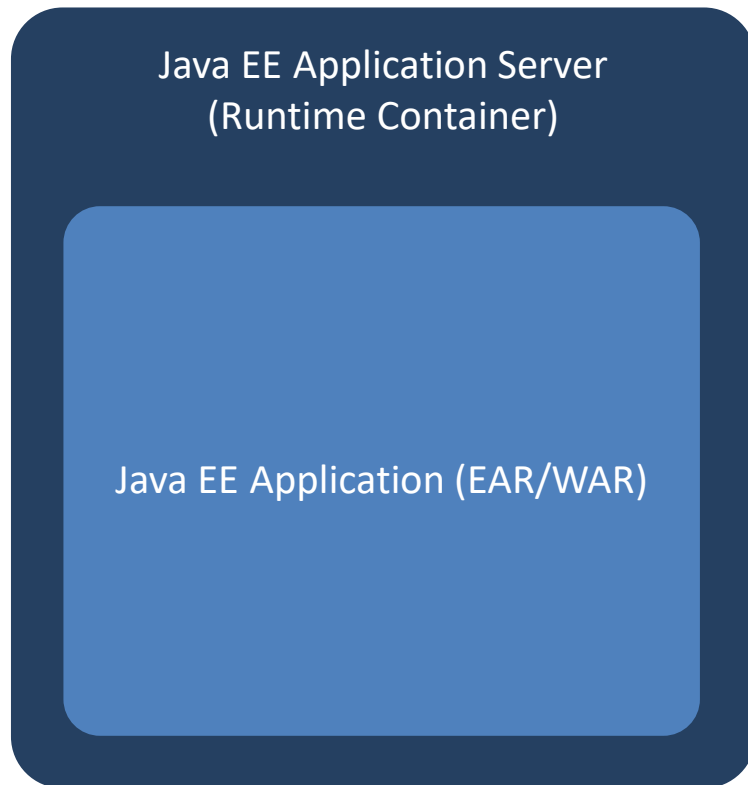
Docker Fact Sheet

- Docker Images können vererbt werden (Inheritance)
- Docker Images sind immutable: Konfiguration bei Ausführung
- Benötigte Ressourcen (CPU, RAM) können beschränkt werden
- Virtuelles Netzwerk mappt interne Ports auf externe Ports
- Maven-Plugins für Steuerung von Docker vorhanden
- Unterstützung von Docker unter Windows verbesserungswürdig => auf Linux setzen
- Best Practices für Docker beachten!

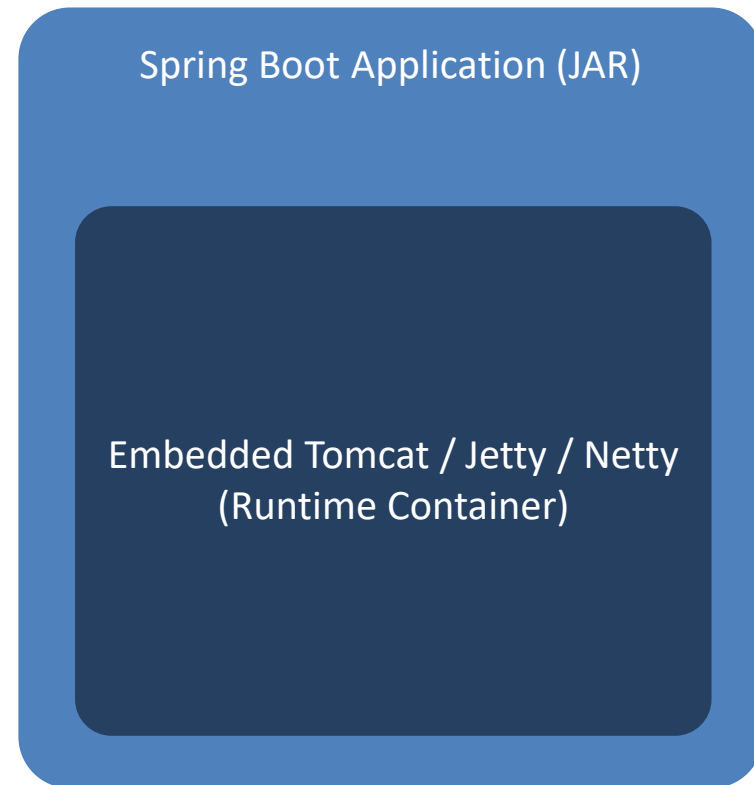
CLOUD NATIVE JAVA MIT SPRING BOOT

Spring Boot vs Java EE

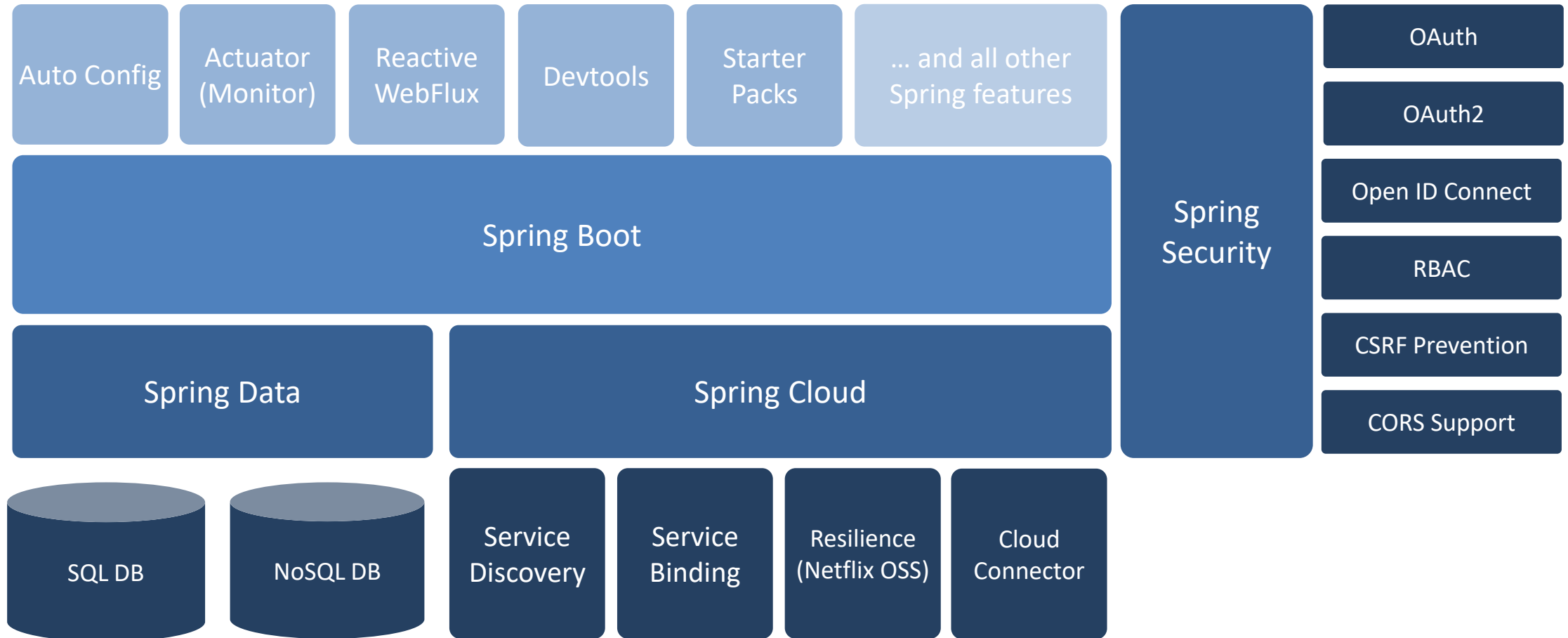
Java EE Classic



Spring Boot



Bausteine aus dem Spring Boot Universum



Fragen?



ANHANG

Kontakt



Michael Theis

Lehrbeauftragter Hochschule München

email michael.theis@hm.edu

mobile + 49 170 5403805

web <http://www.tschutschu.de/Lehrauftrag.html>