

KAPITEL 2: GRUNDLAGEN



LERNZIELE

- Cluster und Cloud Computing differenzieren
- Infrastructure, Platform und Software as a Service unterschieden und erklären
- Verschiedene Arten des Cloud Betriebs erklären
- Spannung zwischen Entwicklung und Betrieb darlegen
- Prinzipien des DevOps nennen und erklären
- Lebenszyklus von DevOps erklären

2.1 CLUSTER COMPUTING



Parallel Computing ist die simultane Ausführung einer aufgeteilten Aufgabe auf mehreren Prozessoren, um das Ergebnis schneller zu erhalten.

Source: https://en.wikipedia.org/wiki/Parallel_computing



Supercomputer Fugaku

- Cores: 7,630,848
- Memory: 5,087,232 GB
- Processor: A64FX 48C 2.2GHz
- Linpack Performance (Rmax): 442,010 TFlop/s
- Costs: US\$1 billion (total programme cost)

Source: <https://www.top500.org/system/179807/> and
<https://www.nytimes.com/2020/06/22/technology/japanese-supercomputer-fugaku-tops-american-chinese-machines.html>

- Supercomputer sind teuer
- Workstations sind "günstig"
- Supercomputer aus "Workstations" ist "günstiger"?
- TOP 500 enthält in den Top 10 auch Cluster-Architekturen



Cluster computing or High-Performance computing frameworks is a form of computing in which bunch of computers (often called nodes) that are connected through a LAN (local area network) so that, they behave like a single machine.

Source: <https://ecomputernotes.com/computernetworkingnotes/network-technologies/cluster-computing>

Die Beowulf-Definition lautet: Standardcomputer, die über ein privates Clusternetzwerk verbunden sind und auf denen eine Open-Source-Softwareinfrastruktur für skalierbare Hochleistungsrechner läuft.

Source: https://en.wikipedia.org/wiki/Beowulf_cluster

Hochverfügbarkeit Cluster

- steigern die Verfügbarkeit und verbessern die Ausfallsicherheit
- im Fehlerfall werden die Dienste vom defekten Host auf einen anderen Host übertragen
- Typische Anwendungsfälle sind Szenarien, in denen Ausfallzeiten pro Jahr maximal einige Minuten betragen sollen
- Sonderfall *stretched Cluster*: Räumliche Trennung der verschiedenen Cluster

High Performance Computing Cluster

- Verwendet für Berechnungsverfahren, die auf mehrere Hosts verteilt werden
- Arten der Aufgabenverteilung
- Zerteilung einer Aufgabe in Pakete und Scheduling im Rechen-Cluster
- Verteilung einzelner Aufgaben über verschiedene Rechen-Instanzen im Cluster
- Anwendungsfälle zum Beispiel im wissenschaftlichen Bereich (überwiegender Einsatzzweck), Rendern von 3D-Grafiken und -Animationen

Load-Balancing Cluster

- Verteilung von Last auf mehrere Rechen-Instanzen
- Aus Nutzer-Perspektive ist nur eine Einheit sichtbar, die aber aus mehreren vernetzten Systemen besteht
- Leistungssteigerung durch hinzunahme von weiteren Rechen-Instanzen im Cluster
- Anwendungsfälle sind Szenarien mit hohen Anforderungen an die Rechenleistung
- Layer 4 Load-Balancing: Nutzt Paket-Informationen (z.B. TCP als Layer 4 Protokoll), um Anfragen auf Rechenkapazitäten weiterzuleiten
- Layer 7 Load-Balancing: Nutzt Informationen der Anwendung (z.B. HTTP als Layer 7 Protokoll), um Anfragen auf Rechenkapazitäten weiterzuleiten
- DNS-basiertes Load-Balancing: Mehrere IP-Einträge sind früher möglich gewesen, aber heute nicht mehr empfehlenswert (Client-Support schwindet), Alternative sind Geo-basierte DNS

Abgrenzung

- Parallel und Cluster Computing haben einen Bezug zu den Themen dieser Veranstaltung
- Fokus wird auf Anwendungskonzepten liegen, die in verteilten Systemen arbeiten, aber nicht zwingend dem Parallel Computing oder dem Cluster Computing zuzuordnen sind
- Es geht weniger darum einen Algorithmus auf Prozessoren für schnellere Berechnung aufzuteilen oder in einem Cluster spezielle Software zur Anwendung zu bringen

ÜBUNG: CLUSTER COMPUTING

Welche Aussagen stimmen?

- Parallel Computing ist im Cluster nicht möglich
- Parallel Computing ist die simultane Ausführung einer aufgeteilten Aufgabe auf mehreren Prozessoren
- Supercomputer und Cluster-Computer ist das selbe
- Beowulf Cluster beschreiben Rechnernetzwerke aus Standardcomputer die über eine Open-Source-Softwareinfrastruktur parallele Berechnungen durchführen

ÜBUNG: ARTEN VON CLUSTER COMPUTING (1)

Welche Aussagen stimmen?

- High Performance Cluster verteilten Aufgaben als Pakete auf mehrere Rechen-Instanzen
- Hochverfügbarkeits Cluster werden zur Steigerung der Ausfallsicherheit verwendet
- Load-Balancing erlaubt das verteilen einer Aufgabe auf mehrere Rechen-Instanzen

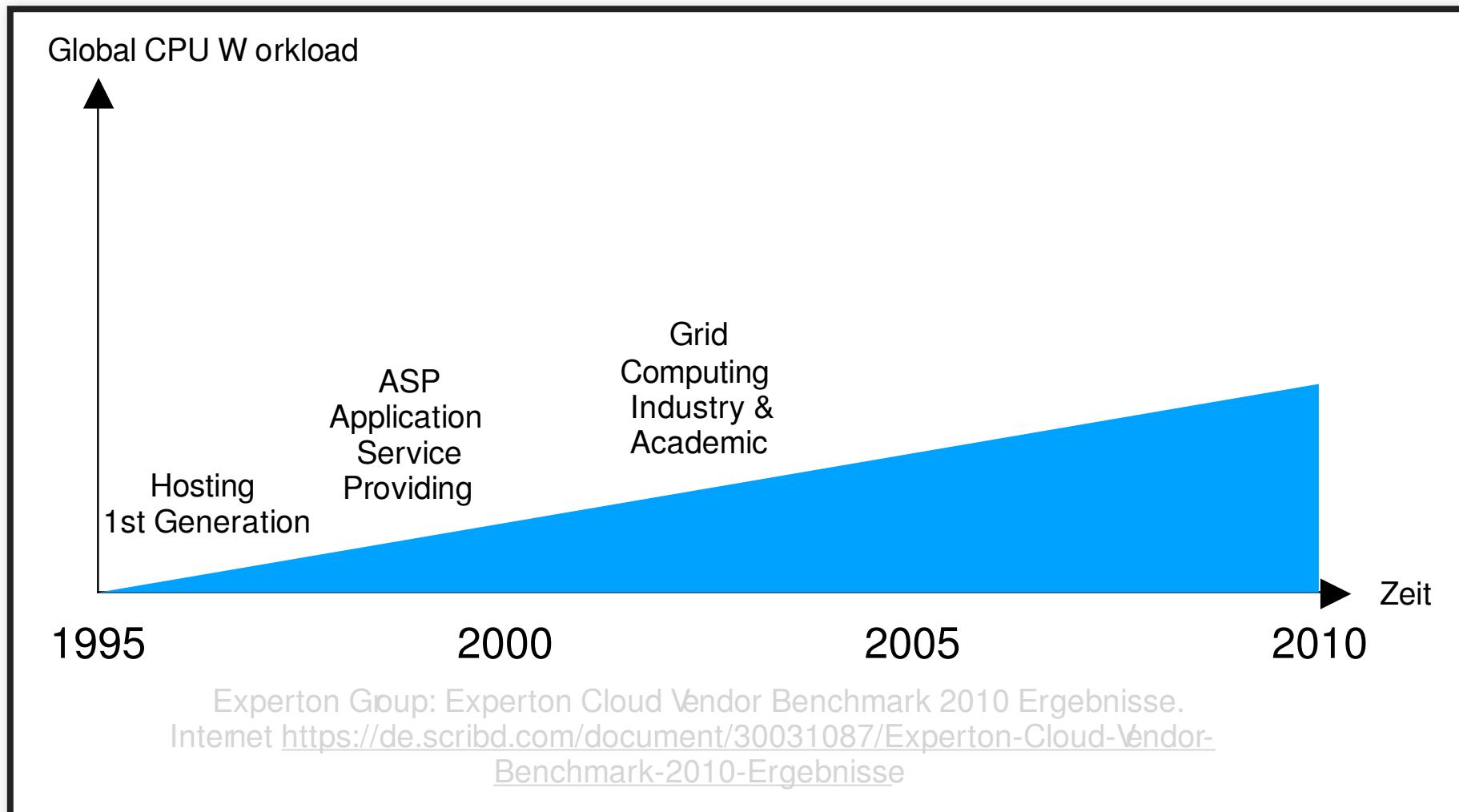
ÜBUNG: ARTEN VON CLUSTER COMPUTING (2)

Welche Aussagen stimmen?

- Hochverfügbarkeits Cluster sind räumlich nicht trennbar
- Load-Balancing erlaubt das verteilen von Anfragen basierend auf dem Inhalt der Anfrage
- High Performance Cluster verteilt eine Aufgaben verfügbare Rechen-Instanzen

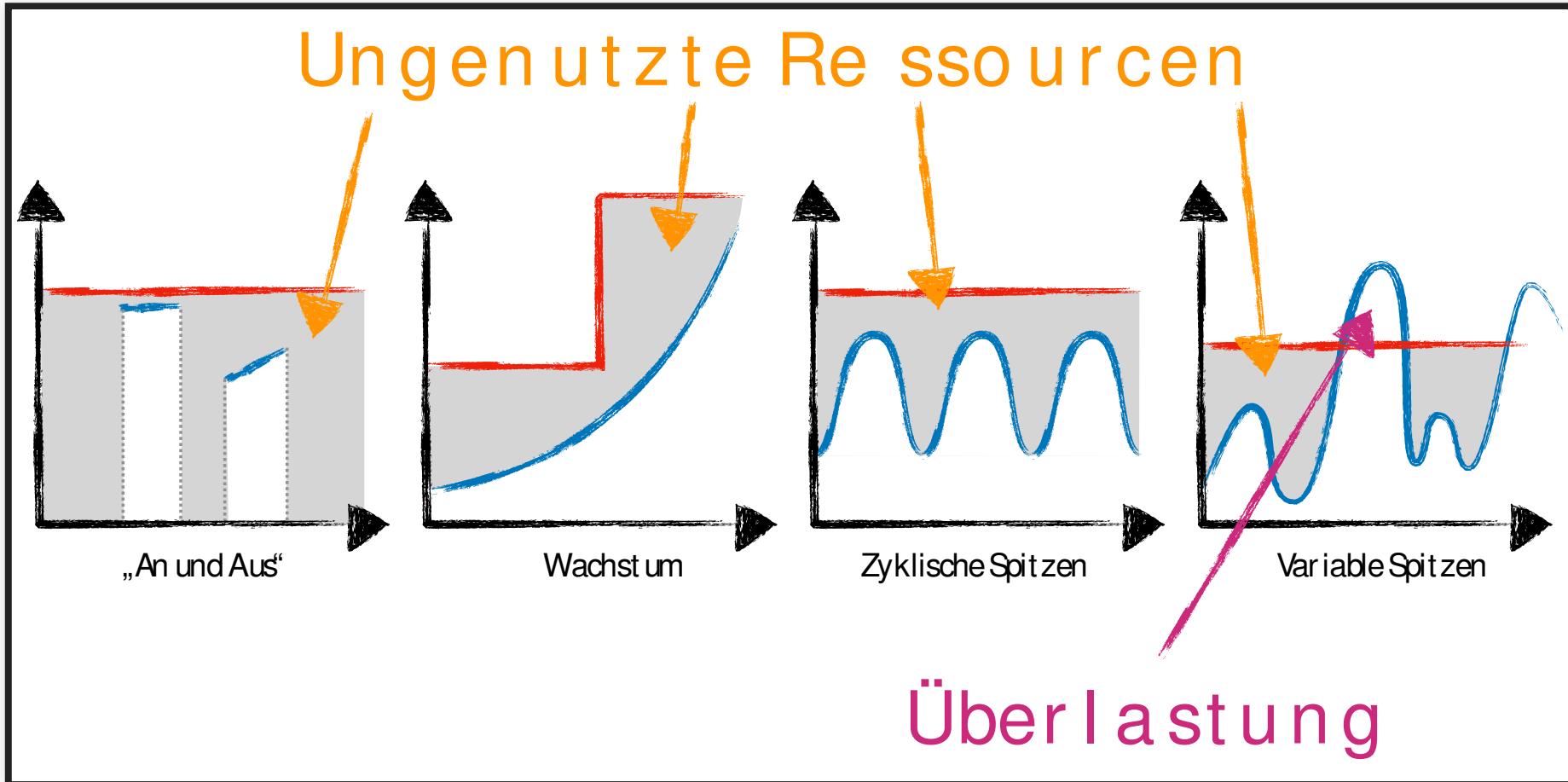
2.2 CLOUD COMPUTING

Was war vor der Cloud



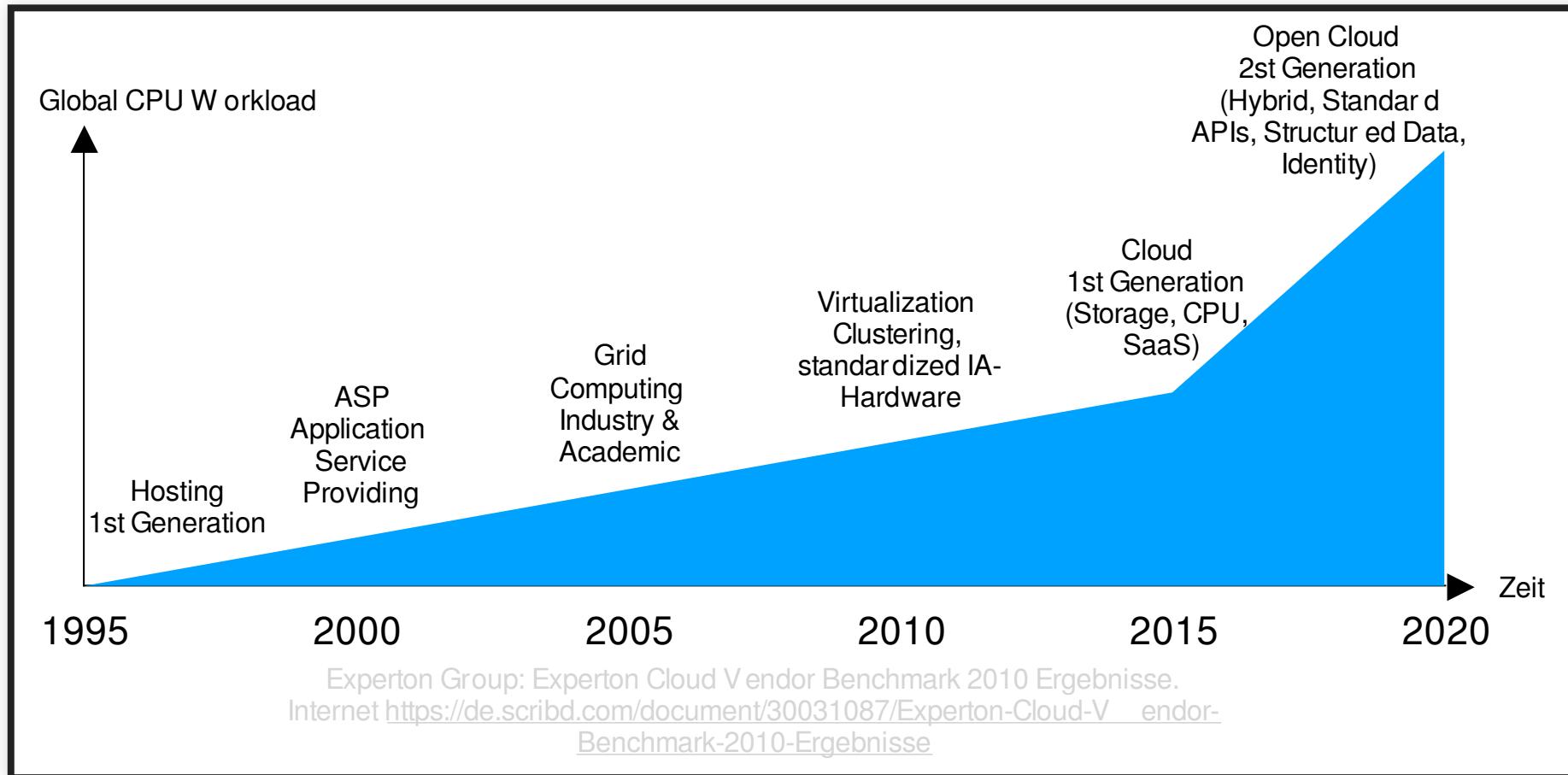
- Erste Generation der Hosting-Angebot in den 90iger Jahren
- In den frühen 2000'der Jahre erste Application Service Provider zur Bereitstellung von Anwendungen im Internet
- Mitte 2000'der Jahre waren Grid-Computing insbesondere im industriellen und akademischen Bereich im Fokus

Typische Problemstellungen



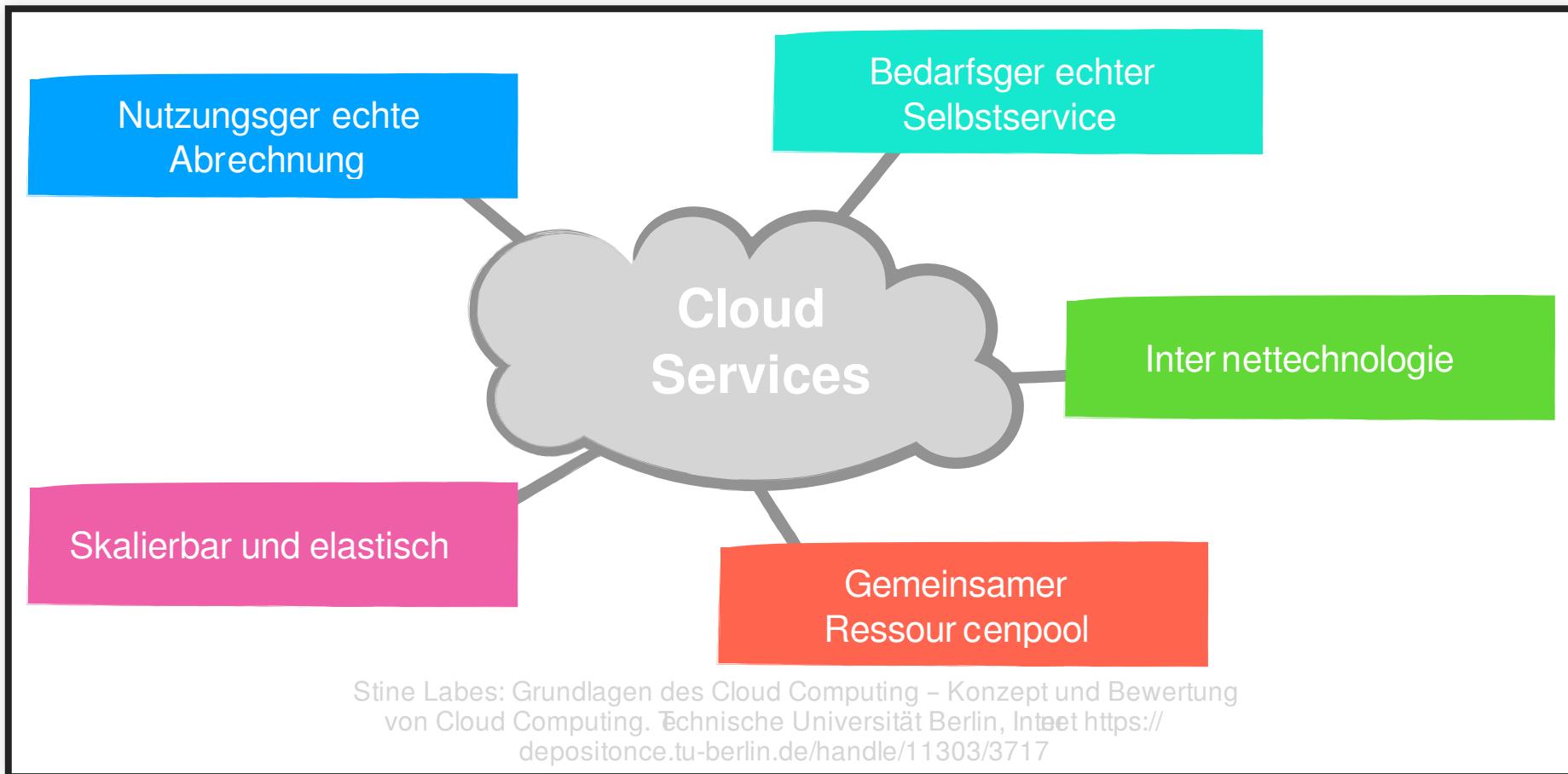
- **An und Aus:** Sporadisch genutzte Last, die nur bei Bedarf erforderlich ist
- **Wachstum:** Kontinuierlicher Wachstum, die Anwendung benötigt mehr und mehr Last
- **Zyklische Spitzen:** Über den Tages oder Wochenverlauf gibt es unterschiedliche Lastanforderungen
- **Variable Spitzen:** Nicht direkt vorhersehbare Lastbedarfe
- Rote Linie beschreibt klassische Ressourcenbereitstellung, zum Beispiel durch Server-Instanzen
- Grauer Bereich beschreibt dann ungenutzte Ressourcen

*Ergebnis: Skalierbare Dienstleistungen für verschiedene Anforderungen
zusammengefasst unter dem Begriff Cloud*



- Neuere Ansätze setzen auf Virtualisierung von Hardware
- Erste Generationen von Clouds erwachsen, bei denen Kunden notwendige Ressourcen für kurze oder längere Phasen nach Bedarf beziehen
- Zukünftig ist eine weitere Standardisierung der Cloud-Angebote, z.B. über APIs zu erwarten

Technologie-Merkmale von Cloud Computing





- **Bedarfsgerechter Selbst-Service**

Der Verbraucher kann den benötigten Umfang des Services selbst zusammenstellen, ohne direkte physische Interaktion mit dem Anbieter

- **Internettechnologie**

Die Services werden mit Hilfe von gegebenen Standards über das Internet zur Verfügung gestellt

- **Gemeinsamer Ressourcenpool**

Die Services verfügen über einen gemeinsam nutzbaren Ressourcenpool, so dass der Anbieter von Größenvorteilen profitieren kann

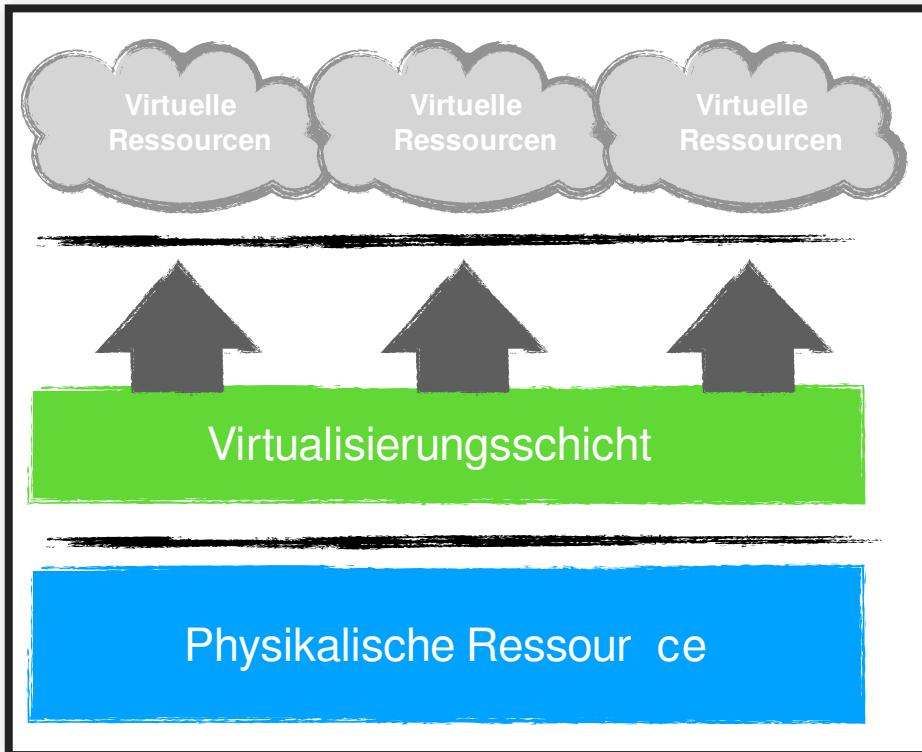
- **Skalierbar und elastisch**

Die Services können bei Bedarf skaliert werden, so dass Ressourcen jederzeit hinzugefügt oder entfernt werden können

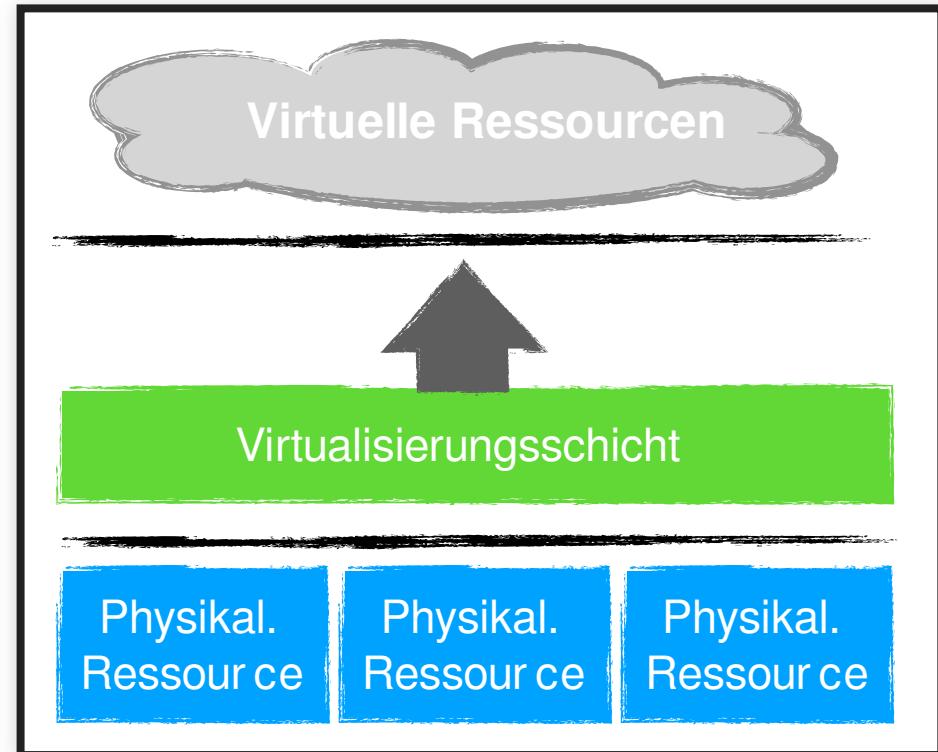
- **Nutzungsgerechte Abrechnung**

Die Nutzung der Services wird durch Nutzungskennzahlen protokolliert, so dass eine Kostentransparenz gegeben ist

Bedarfsgerechte Ressourcen und effiziente Hardwarenutzung



Virtuelle Ressourcen, die größer sind als die zugrundeliegende Hardware



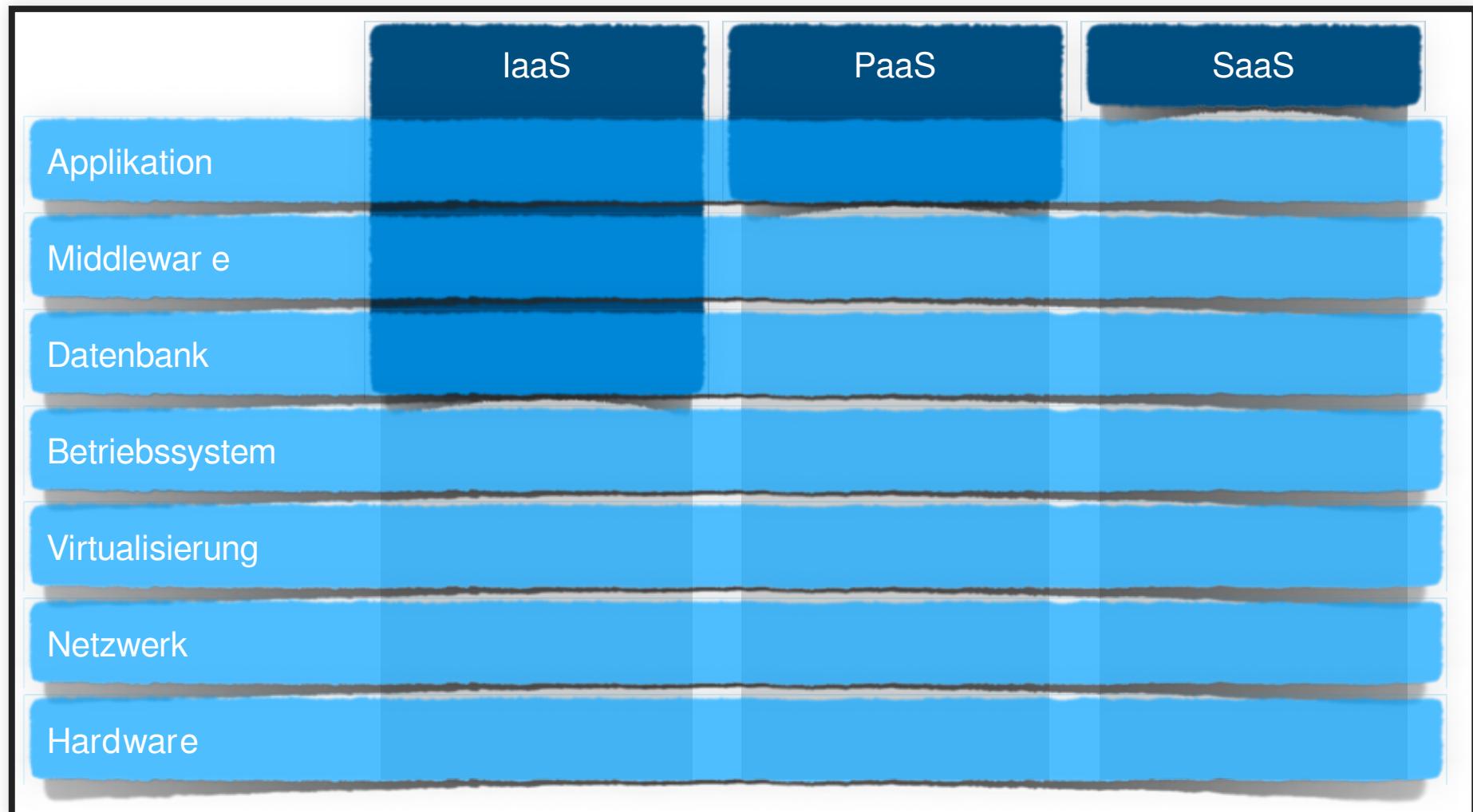
- Virtualisierung in der Cloud als übliche Strategie
- Dies ist möglich durch den Betrieb von mehreren Kundeninstanzen auf einer Hardware, um diese bedarfsgerecht auszulasten
- Ebenso ist es denkbar, dass mehrere Hardware-Instanzen virtuelle Ressourcen schaffen, die so als einzelne Hardware-Ressource nicht möglich gewesen wäre

ÜBUNG: CLOUD COMPUTING

Welche Aussagen stimmen?

- Internettechnologien spielen keine entscheidende Rolle im Kontext von Cloud Computing
- Zu den Technologie-Merkmalen gehört bedarfsgerechter Selbstservice und gemeinsamer Ressourcenpool
- Cloud Computing hilft ungenutzte Ressourcen zu reduzieren und Überlastungen zu kompensieren
- Cloud Computing nur meint das Anmieten eines vorkonfigurierten Clusters

2.3 AS A SERVICE



Infrastructure as a Service



- Bereitstellung virtualisierter IT-Infrastruktur
- Die Infrastruktur umfasst z. B. den Server, die Netzkapazitäten, die Kommunikationsgeräte, die Archivierungs- und Backup-Systeme, sowie weitere Komponenten aus RZ- und Netzwerkinfrastruktur
- Es gibt eine grundlegende Unterscheidung zwischen Cloud Computing und Cloud Storage
- Angebote heute oftmals standardisiert (vgl. AWS oder Azure Cloud)



Typische IaaS Dienste

- *Compute as a Service*: Ext. Bereitstellung von Rechenleistung, Skalierung auf Basis der Lastanforderung und nutzungsabhängige Bezahlung
- *Storage as a Service*: Sicherung und Archivierung von Daten zur orts- und medienunabhängigen Nutzung und Bezahlung von genutzttem Volumen.
- *Network Services*: Bereitstellung von Datenpaketen über Content Delivery Networks (CDNs), welche in verschiedenen Netzwerkzonen repliziert sind und eine schnelle Bereitstellung durch Nutzung räumlich naher Endpunkte.
- *Managed Services*: Hosting, Infrastruktur, Sicherheits und Verwaltungsdienstleistungen als eine der ältesten Formen des Cloud Computings.

Platform as a Service



- Dienstanbieter bietet Nutzern des Dienstes die Möglichkeit, eigene Software auf einer Plattform in der Cloud bereitzustellen
- Nutzer sind typischerweise Web-Entwickler
- Provider stellt gesamte Infrastruktur von der Plattform bis zur Hardware zur Verfügung
- Zusätzlich wird oft auch die Entwicklungsumgebung angeboten
- Anbieter legt Rahmenbedingungen fest, wie z.B. Programmiersprache und nutzbare Bibliotheken

Typische PaaS Dienste

- *Anwendungsentwicklung*: Bereitstellung von Entwicklungsplattformen und -werkzeugen zur Erstellung und Ausführung von Anwendungen und Vermeidung von Aufwänden zum Betrieb eigener Infrastruktur.
- *Anwendungsbetrieb*: Betrieb eigener Anwendungen in der Cloud, Bezahlung nach Ressourcennutzung und zentrale Verwaltung ohne Bezug zu realen oder virtuellen Maschinen

Software as a Service



- Dem Endnutzer wird die Software zum unmittelbaren Einsatz angeboten
- Betrieb der Software liegt vollständig beim Anbieter, inklusive Wartung, Aktualisierung, Fehlerbeseitigung, Weiterentwicklung, Lizenzierung und Backup
- Üblicherweise mandantenfähige Software
- Zugriff in der Regel über den Browser

Beispielanwendungen sind in verschiedenen Bereichen zu finden:

- Rechnungswesen, Buchhaltung und Controlling
- Kollaborationswerkzeuge und Projekt Management
- Content und Dokument Management Systeme
- Customer Relationship Management
- E-Commerce, Marketing und Sales

ÜBUNG: INFRASTRUCTURE AS A SERVICE

Welche Aussagen stimmen? IaaS ...

- ... bezieht sich auf die Bereitstellung von IT-Infrastruktur
- ... gibt Rahmenbedingungen (z.B. welche Programmiersprache verwendet werden muss) vor
- ... bietet Endnutzer-Software in der passenden Infrastruktur die direkt genutzt werden kann
- ... umfasst Angebote für Rechenleistung, Speicher und Netzwerke

ÜBUNG: PLATFORM AS A SERVICE

Welche Aussagen stimmen? PaaS ...

- ... dient der Bereitstellung von webbasierten Mandantensystemen
- ... bietet fertige Infrastruktur von der Plattform bis zur Hardware für einen unkomplizierten Betrieb eigener Anwendungen
- ... umfasst freie Nutzungsmöglichkeiten, in der Regel keine speziellen Rahmenbedingungen
- ... ist vor typischerweise für Web-Entwickler interessant

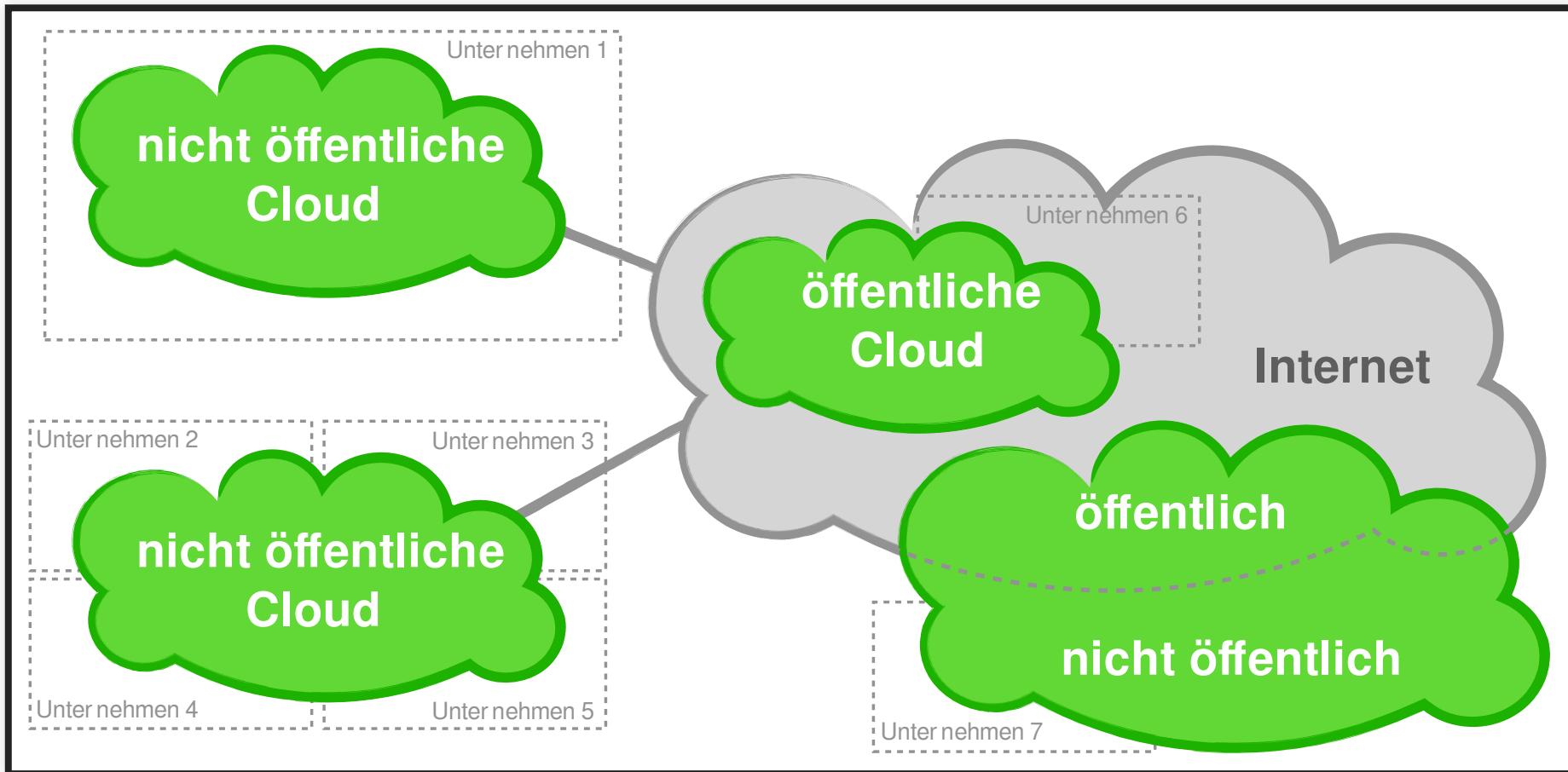
ÜBUNG: SOFTWARE AS A SERVICE

Welche Aussagen stimmen? SaaS ...

- ... bezieht sich auf Software zum Betrieb von Anwendungen in der Cloud
- ... bietet Software die vollständig beim Anbieter betrieben wird
- ... erfordert die Installation durch den Kunden in einer eigenen VM

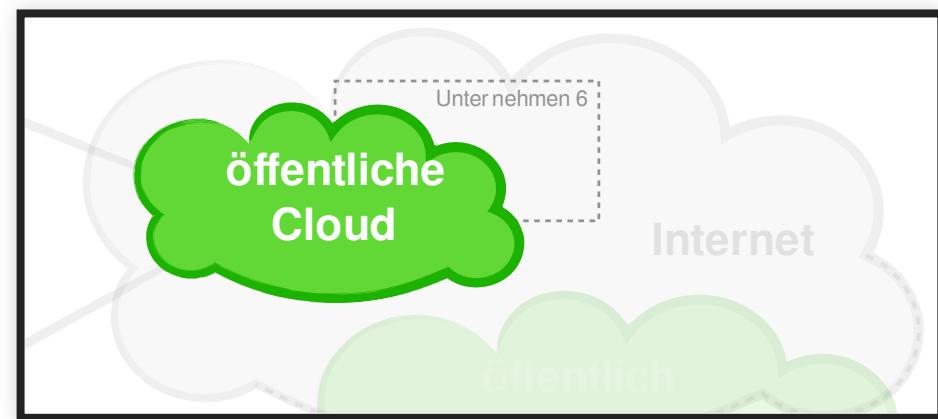
2.4 BETREIBERMODELLE

Vier Arten des Cloud-Betriebs



Public Cloud

- Infrastruktur ist öffentlich zugänglich und wird von einem IT-Dienstleister angeboten
- Hohe Anzahl an Nutzern um hohe Skaleneffekte zu erzielen



- Individualanforderungen nur bedingt realisierbar
- Es kann in zwei Typen unterschieden werden: Open Cloud und Exclusive Open Cloud

Private Cloud



- Organisationen / Unternehmen betreiben eigene Cloud Infrastruktur
- Möglichkeit zur individuellen Ausgestaltung und Zugriffssteuerung, sowie Einfluss auf Datensicherheit und die Compliance
- Weniger Skaleneffekte und Kostenersparnis
- Es kann in drei Typen unterschieden werden: Corporate Cloud, Managed Cloud und Outsourced Cloud

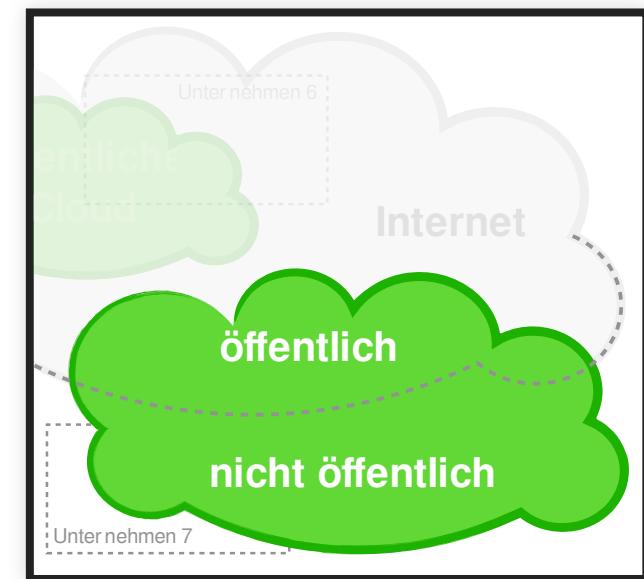
Community Cloud



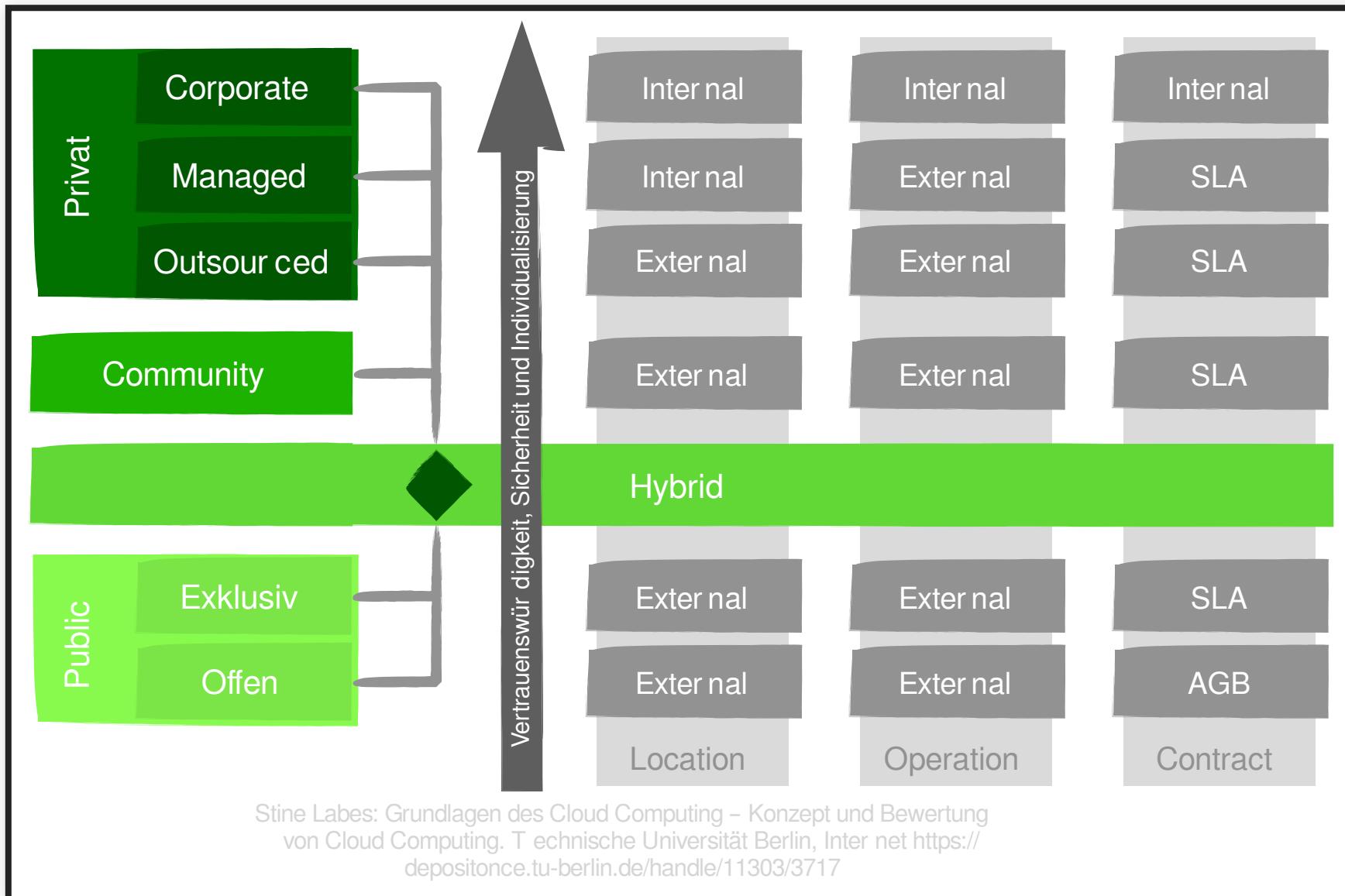
- Zusammenschluss von interessensverwandter Organisationen zur Nutzung einer Infrastruktur
- Verbesserte Skaleneffekte und Kostensparnis gegenüber Privater Cloud
- Geht zu lasten der individuellen Ausgestaltung und Zugriffssteuerung, sowie Einfluss auf Datensicherheit und die Compliance

Hybrid Cloud

- Kombination aus geschlossenen und offenen Cloud Infrastrukturen
- Zum Beispiel lassen sich Ressourcen Engpässe durch Public Cloud Nutzung ausgleichen und gleichzeitig Sicherheitsaspekte mit der Privat Cloud realisieren

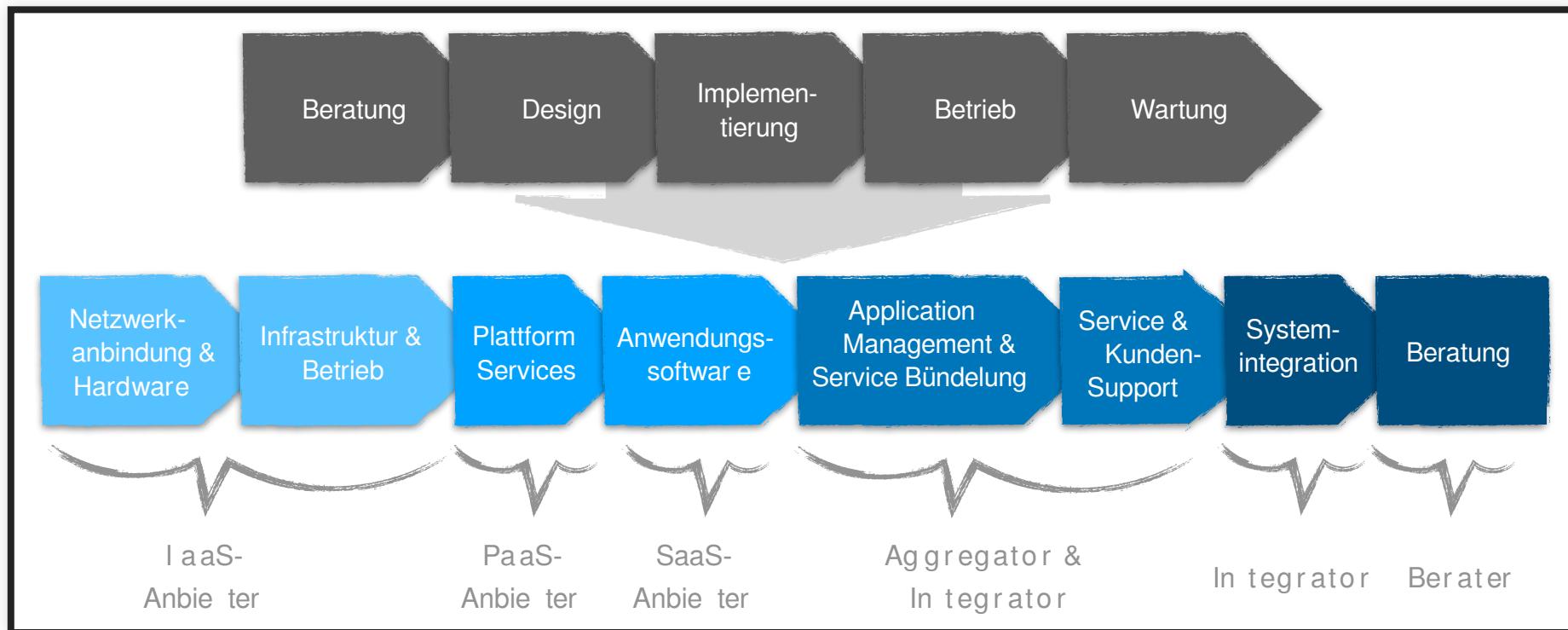


Taxonomie der Cloud Bereitstellungsmodelle



- Location: Beschreibt wo die Cloud betrieben wird (innerhalb oder außerhalb des Unternehmens)
- Operation: Beschreibt von wo aus der Betrieb der Cloud realisiert wird (innerhalb oder außerhalb des Unternehmens)
- Contract: Beschreibt welche Art von Verträgen der in der Cloud betriebenen Komponenten zugrunde liegt.
- SLA: Service-Level-Agreement
- AGB: Allgemeine Geschäftsbedingungen

Wertschöpfungskette





- Klassisches Geschäftsmodell (Beratung, Design, Implementierung, ...) verändert sich
- Es platzieren sich im Markt Anbieter für IaaS-Ressourcen aber auch Anbieter, um IaaS-Angebote zu *betreiben* (Operation vs. Provider)
- Darauf basierend etablieren sich Angebote im Bereich PaaS
- Software-Anbieter können auf den Angeboten aufsetzen und Ihren Mandanten über SaaS-Angebote entsprechende Möglichkeiten schaffen
- Aggregations- und Integrationsangebote helfen Software in Unternehmen zu betreiben
- Integration in interne Systeme von Unternehmen als auch Beratungsangebote was in welchem Kontext für Unternehmen hilfreich ist

ÜBUNG: ARTEN DES CLOUD-BETRIEB

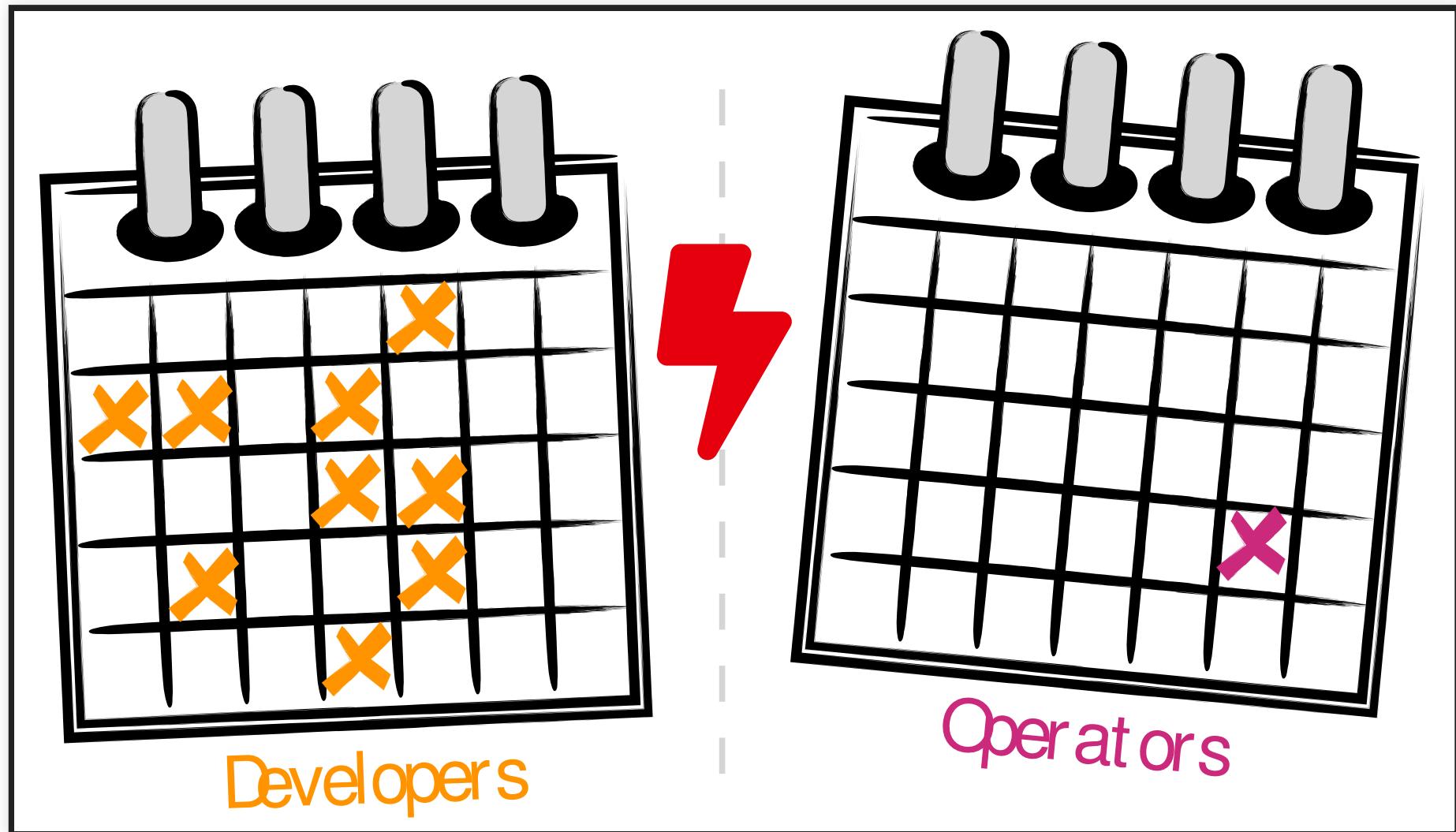
Welche Aussagen stimmen?

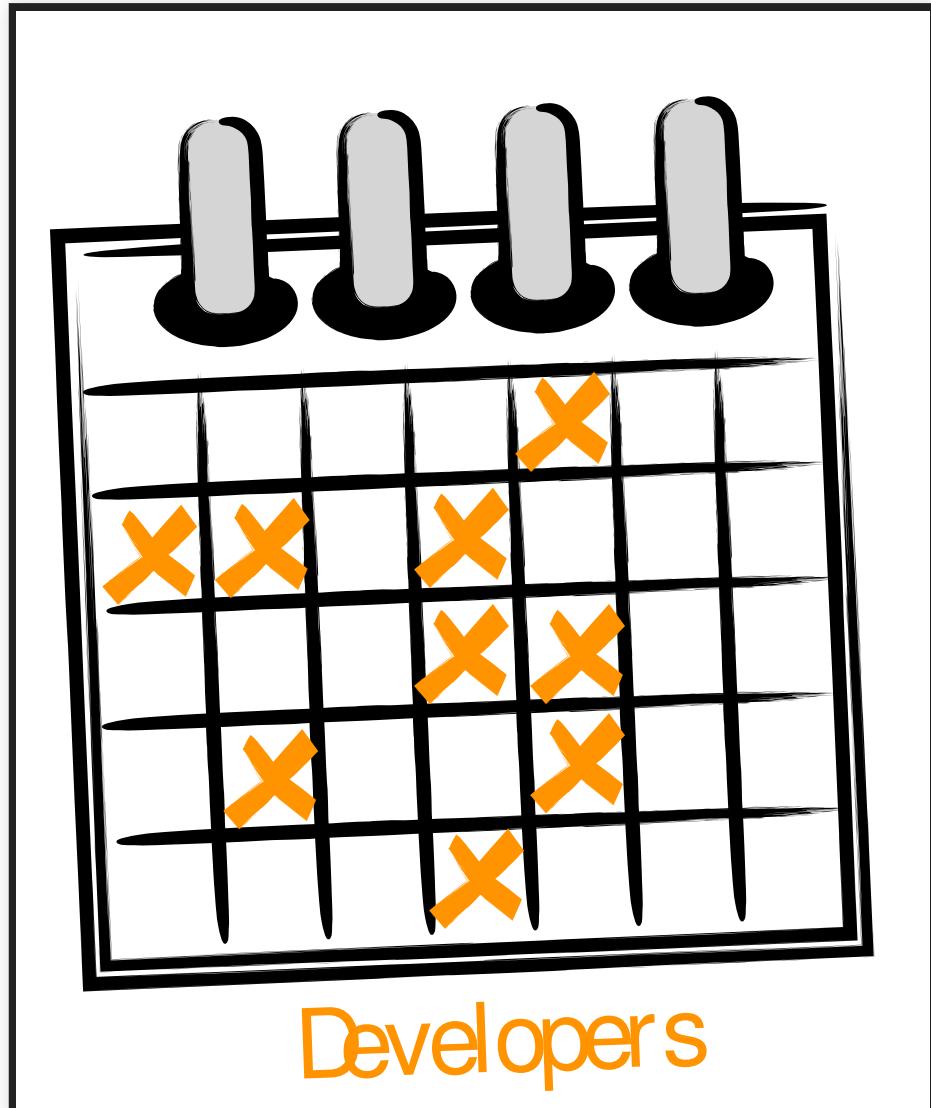
- Individuelle Anforderungen lassen sich nur bedingt in Public Clouds umsetzen
- Community Clouds sind ehrenamtlich betriebene Cloud-Infrastrukturen
- Public Clouds bieten öffentlich zugängliche Infrastruktur
- Hybride Clouds bezieht sich auf eine Kombination verschiedener Community Clouds
- Private Clouds haben keine Anbindung an das Internet

2.5 DEVOPS



John Arundel und Justin Domingus: Cloud Native DevOps mit Kubernetes





*Functionality – die Entwicklung im Fokus
(Developers)*

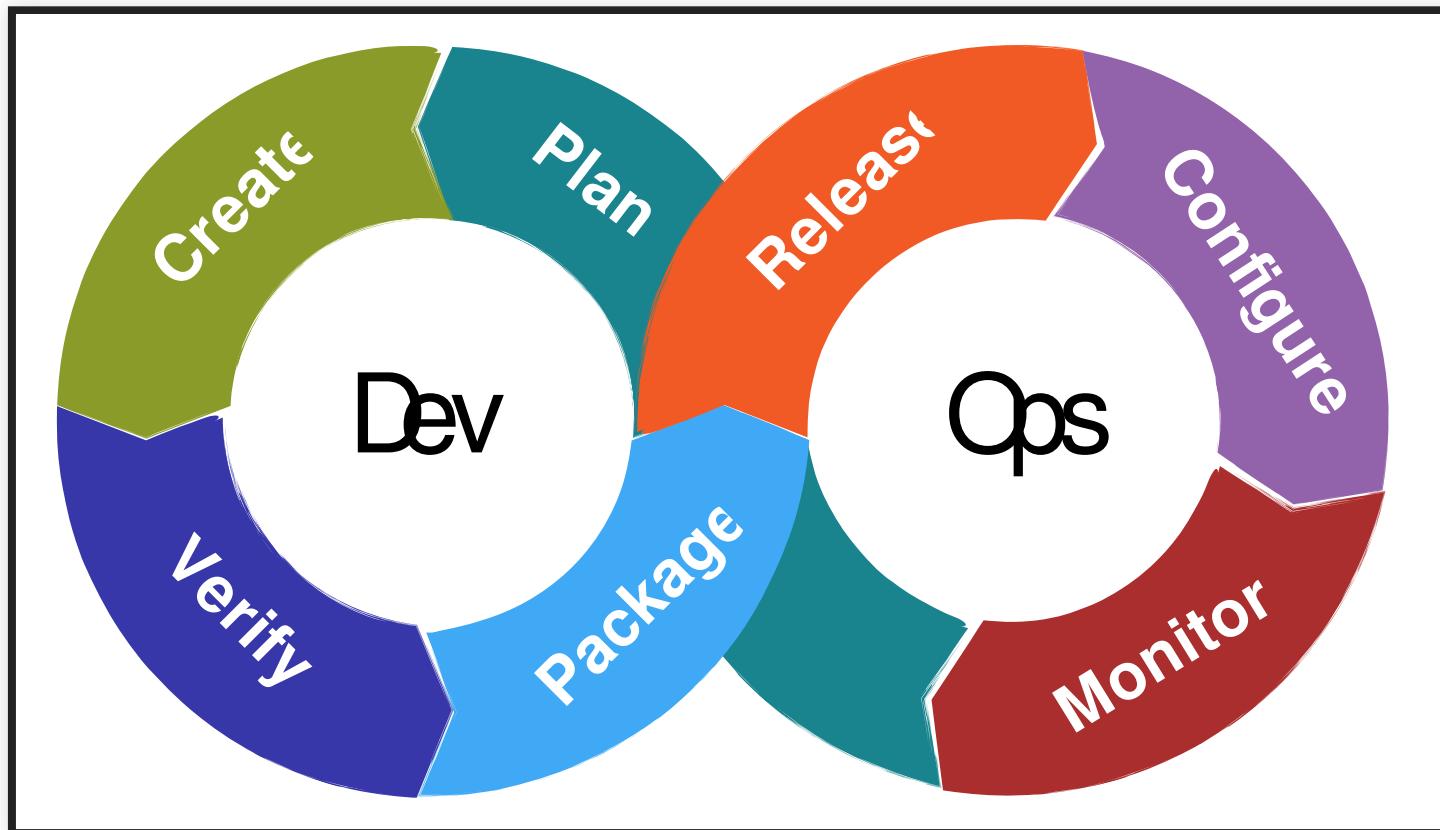
- Weniger komplexes Testen und Debuggen, ein einziger Lieferant
- Schnellere Markteinführung Ihrer Produkte
- Mögliche Zeit-/Kostensparnis für Entwickler

Portability – der Betrieb im Fokus (Operators)

- Vermeiden von Zugriff durch Lieferanten, i.A. Entwickler
- Unterstützung von rechenzentrumsbasierten Anwendungsfällen
- Mögliche Kosteneinsparungen bei Betrieb



Typischer Ablauf zur Entwicklung und Bereitstellung von Anwendungen





- Create: Allgemeine Entwicklungsarbeit, Umsetzung von Lösungen
- Verify: Testen der umgesetzten Lösungen
- Packaging: Erzeugen notwendiger Artefakte, z.B. Java Archive, Docker-Images, usw.
- Releasing: Bereitstellen einer neuen Version
- Configuring: Bereitstellen / Installation einer Version in der Infrastruktur
- Monitoring: Überwachen der laufenden Installation
- Plan: Planen der nächsten Umsetzungsziele



DevOps wendet agile und schlanke Prinzipien über die gesamte Software Supply Chain an.

Ziel ist die Maximierung der Geschwindigkeit zur Lieferung eines Produkts oder einer Dienstleistung; von der ersten Idee über die Produktionsfreigabe bis hin zum Kundenfeedback und den darauf basierenden Verbesserungen.



MOTIVATION

Verlagerung von Systems of Record zu Systems of Engagement



Systems of Record

Traditionelle Softwareanwendungen sind große und komplexe Systeme, die als Erfassungs- und Aufzeichnungssysteme fungieren, große Datenmengen und/oder Transaktionen enthalten und auf Zuverlässigkeit und Stabilität ausgelegt sind.

Diese Anwendungen müssen nicht oft geändert werden, Unternehmen können ihre Kunden und ihre eigenen geschäftlichen Anforderungen erfüllen, indem sie nur ein oder zwei große neue Versionen pro Jahr bereitstellen.



Systems of Engagement

Mit der mobilen Kommunikation und der Verbreitung von Webanwendungen werden die Aufzeichnungssysteme durch Systems of Engagements ergänzt, auf die die Kunden direkt zugreifen und mit dem Unternehmen interagieren können.

Solche Anwendungen müssen einfach zu bedienen und leistungsstark sein, und die Systeme müssen sich schnell ändern, um sich den Bedürfnissen der Kunden und den Marktkräften anzupassen.



ZIELSTELLUNG

Verbesserte Kundenzufriedenheit

Unternehmen müssen kontinuierlich Kundenfeedback einholen und darauf reagieren, was Mechanismen erfordert, um schnelles Feedback von allen Stakeholdern in der Softwareanwendung zu erhalten: Kunden, Branchen, Anwender, Lieferanten, Partner, etc.

Erhöhte Innovationsfähigkeit

Moderne Unternehmen nutzen Lean Thinking-Ansätze, um ihre Innovationsfähigkeit zu steigern. Ihre Ziele sind die Reduzierung von Verschwendungen und Nacharbeit sowie die Verlagerung von Ressourcen in höherwertige Aktivitäten.



Schnelleres Time to Value

Entwicklung einer Kultur, Praxis und Automatisierung, die eine schnelle, effiziente und zuverlässige Softwarebereitstellung bis hin zur Produktion ermöglicht.

DevOps bietet, wenn es als Geschäftsfähigkeit angenommen wird, die notwendigen Werkzeuge und die Kultur, um eine effiziente Release-Planung, Vorhersagbarkeit und Erfolg zu ermöglichen.



PRINZIPIEN



PRINZIPIEN

Entwicklung und Test gegen produktionsähnliche Systeme



PRINZIPIEN

Entwicklung und Test gegen produktionsähnliche Systeme

Deployment mit wiederholbaren, zuverlässigen Prozessen



PRINZIPIEN

Entwicklung und Test gegen produktionsähnliche Systeme

Deployment mit wiederholbaren, zuverlässigen Prozessen

Überwachung und Validierung der Betriebsqualität



PRINZIPIEN

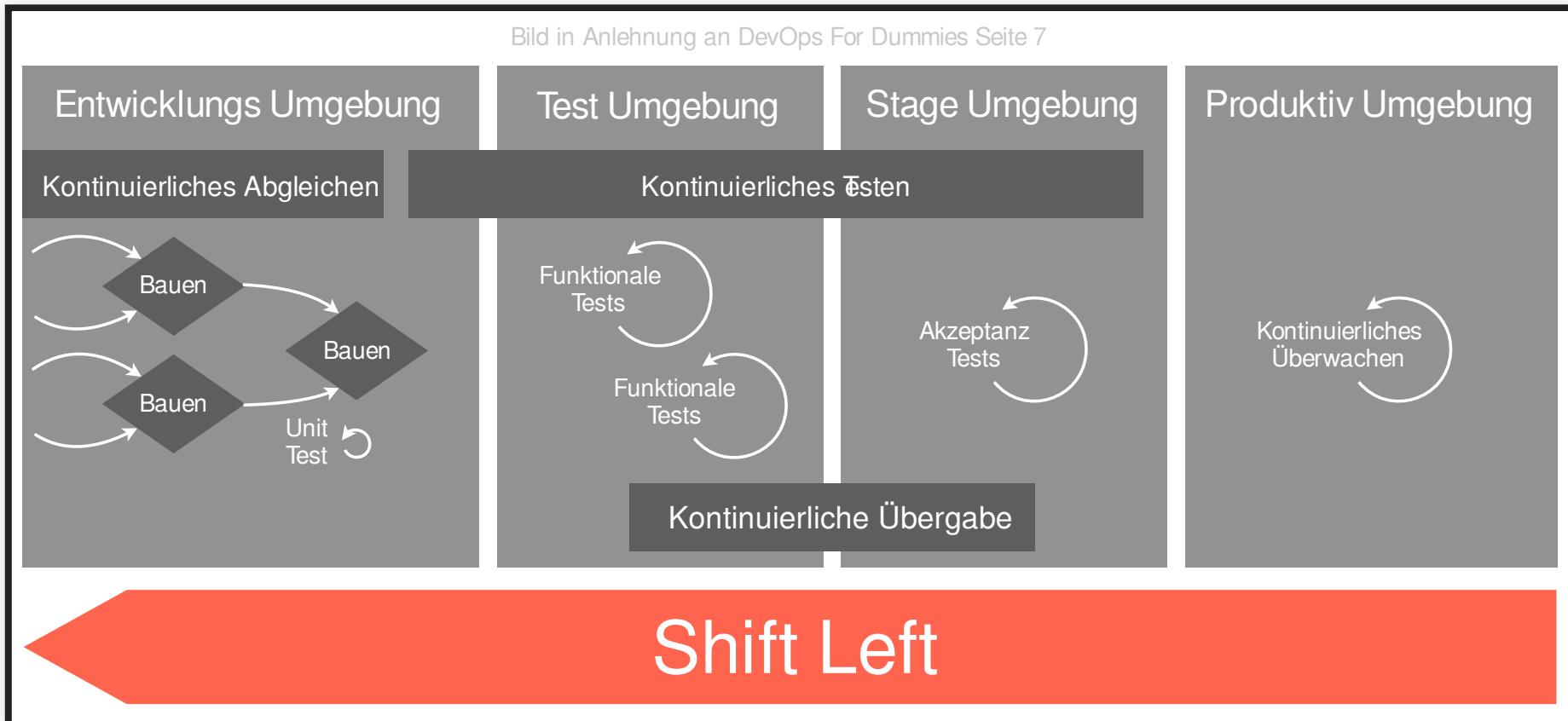
Entwicklung und Test gegen produktionsähnliche Systeme

Deployment mit wiederholbaren, zuverlässigen Prozessen

Überwachung und Validierung der Betriebsqualität

Rückkopplungsschleifen verstärken

Entwicklung und Test gegen produktionsähnliche Systeme





Deploy mit wiederholbaren, zuverlässigen Prozessen

- Entwickeln und Operieren die Möglichkeit für einen agilen (oder zumindest iterativen) Softwareentwicklungsprozesses bis hin zur Produktion ermöglichen
- Erfordert Automatisierung der iterativen, häufigen, wiederholbaren und zuverlässigen Prozesse
- Lieferpipelines, die eine kontinuierliche, automatisierte Bereitstellung und Prüfung ermöglicht

Überwachung und Validierung der Betriebsqualität

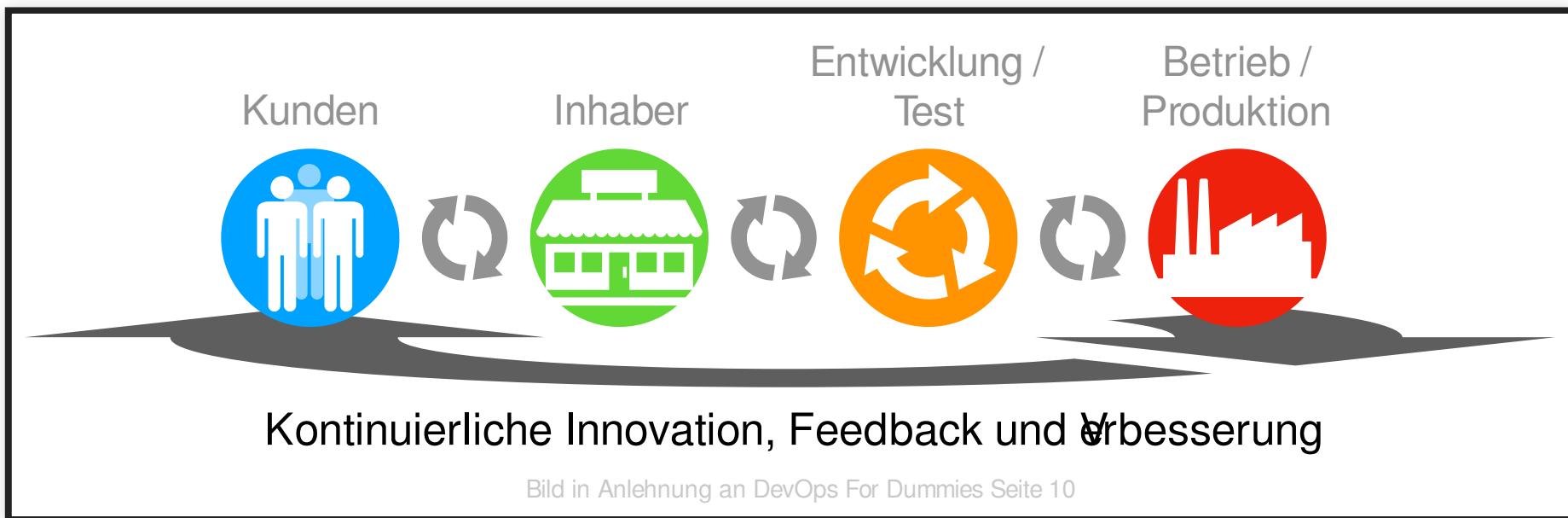
- Häufig existieren Anwendungen und Systeme in der Produktion zu Überwachung, die die Metriken der Produktionssysteme in Echtzeit erfassen. Aber sie überwachen in der Regel isoliert und getrennt!
- Das Prinzip zur Überwachung verschiebt selbige zu einem früheren Zeitpunkt im Lebenszyklus. Es wird verlangt, dass automatisierte Tests frühzeitig und häufig im Lebenszyklus durchgeführt werden, um funktionale und nicht-funktionale Eigenschaften der Anwendung zu überwachen.
- Wann immer eine Anwendung eingesetzt und getestet wird, sollten Qualitätsmetriken erfasst und analysiert werden.



Rückkopplungsschleifen verstärken

- Schnelles Feedback und schnelles lernen aus jeder Aktion
- Es sind Kommunikationskanäle zu schaffen, die es allen Stakeholdern ermöglichen, auf Feedback zuzugreifen und darauf zu reagieren.
 - Developer können handeln, indem sie ihre Projektpläne oder Prioritäten anpassen
 - Operators können handeln, indem sie die Ausführungsumgebungen verbessern
 - Companies können handeln, indem sie ihre Veröffentlichungspläne ändern

LEBENSZYKLUS





Kontinuierliche Innovation, Feedback und Verbesserung



Kontinuierliche Innovation, Feedback und Verbesserung

Planen und Messen



Kontinuierliche Innovation, Feedback und Verbesserung

Planen und Messen

Entwickeln und Testen



Kontinuierliche Innovation, Feedback und Verbesserung

Planen und Messen

Entwickeln und Testen

Release und Deploy



Kontinuierliche Innovation, Feedback und Verbesserung

Planen und Messen

Entwickeln und Testen

Release und Deploy

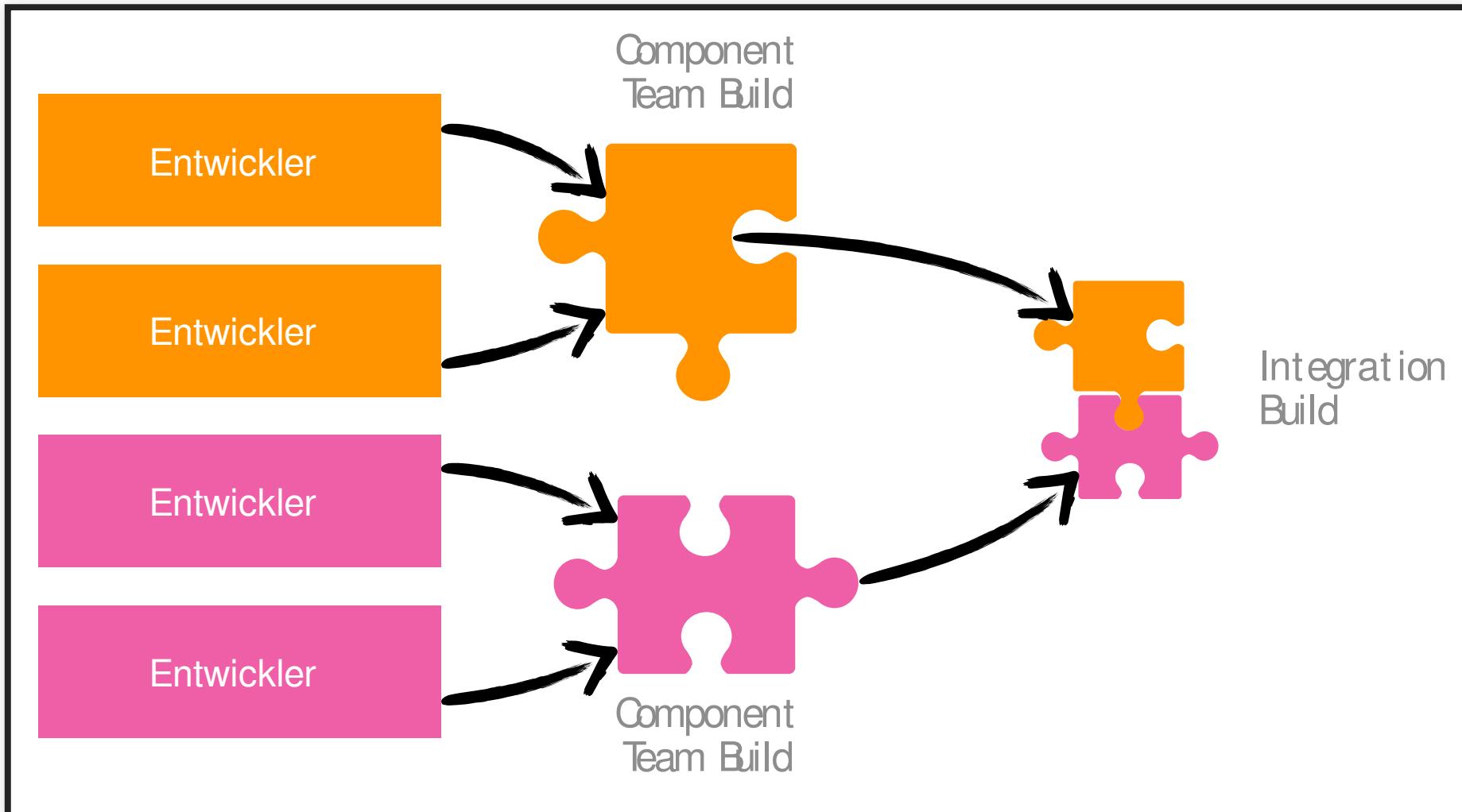
Überwachen und Optimieren



Planen und Messen

- Kontinuierliche Unternehmensplanung
- Unternehmen müssen agil sein und schnell auf Kundenfeedback reagieren können.
- Unternehmen messen den Fortschritt und finden heraus, was die Kunden wirklich wollen. Daraus ändert sich dann die Richtung, Geschäftspläne werden entsprechend aktualisiert, so dass sie in einer Umgebung mit begrenzten Ressourcen kontinuierliche Kompromissentscheidungen treffen können.

Entwickeln und Testen



Collaboration via Continuous Integration



Collaborative Development

- Häufig sind eine große Anzahl von funktionsübergreifende Teams involviert.
- Praktiker aus diesen Teams arbeiten auf mehreren Plattformen und können über mehrere Standorte verteilt sein.
- Die kollaborative Entwicklung ermöglicht es diesen Praktikern, zusammenzuarbeiten, indem sie einen gemeinsamen Satz von Praktiken und eine gemeinsame Plattform zur Verfügung stellen, auf der sie Software entwickeln und bereitstellen können.

Wie kollaborativ Entwickeln?

- Source Code Management
- Continunous Integration



Continuous Testing

Continuous Testing bedeutet frühzeitiges und kontinuierliches Testen über den gesamten Lebenszyklus, was zu reduzierten Kosten und kurzen Testzyklen führt und ein kontinuierliches Feedback zur Qualität ermöglicht. Dies geschieht durch den Einsatz von automatisiertem Testen und Service-Virtualisierung.

Wie kontinuierlich Testen?

- Automated Testing
- Service Virtualization



Release und Deploy

- Definition eines Release und Deployment Prozesses bilden die Basis von DevOps Fähigkeiten
- Continuous Release und Deployment bringen das Konzept der Continuous Integration in den nächsten Schritt
- Prozesse, die den Release und das Deployment ermöglichen, ermöglichen auch die Erstellung einer Lieferpipeline
- Lieferpipeline ermöglichen das Continuous Deployment von Software in der Qualitätssicherung und anschließend in der Produktion auf effiziente und automatisierte Weise
- Das Ziel von Continuous Release und Deployment ist es, neue Funktionen so schnell wie möglich für Kunden und Benutzer freizugeben.

Wie automatisieren?

- Beschreibung von Abhängigkeiten und Bauprozessen
- Unabhängige Systeme die bestimmte Prozesse ausführen
- Ausführbare Artefakte und Konfigurationsmanagement



Überwachen und Optimieren

Continuous Monitoring

Continuous Monitoring liefert Daten und Metriken für Operations, QS, Entwicklung, Fachabteilungen und andere Stakeholder über Anwendungen in verschiedenen Phasen des Lieferzyklus.



Continuous Customer Feedback and Optimization

Daten darüber, wie Kunden die Anwendung nutzen und Feedback, das diese Kunden bei der Nutzung der Anwendung geben.

Dieses Feedback ermöglicht es verschiedenen Stakeholdern, geeignete Maßnahmen zu ergreifen, um die Anwendungen zu verbessern und die Kundenzufriedenheit zu erhöhen.

Wie überwachen und optimieren?

- Logprocessing
- System Metriken
- Kundenfeedback Erfassung und Verarbeitung



WAS IST KEIN DEVOPS

- DevOps ist **keine** neue Abteilung
- DevOps ist **kein** Marketingbegriff (Buzz)
- DevOps ist **kein** neuer Job
- DevOps ist **keine** Tool-Suite
- DevOps erlaubt es Entwicklern **nicht**, am Produktivsystem zu arbeiten



WAS IST DEVOPS

DevOps ist eine Mischung aus Mustern zur Verbesserung der Zusammenarbeit zwischen Entwicklung und Betrieb.

DevOps adressiert gemeinsame Ziele und Anreize sowie gemeinsame Prozesse und Werkzeuge. Aufgrund der natürlichen Konflikte zwischen verschiedenen Gruppen sind gemeinsame Ziele und Anreize nicht immer erreichbar. Sie sollten jedoch zumindest aufeinander abgestimmt sein.

DevOps respektiert die Tatsache, dass die Unternehmen und Projekte spezifische Kulturen haben und dass die Menschen wichtiger sind als Prozesse, die wiederum wichtiger sind als Werkzeuge. DevOps akzeptiert die Unvermeidlichkeit von Konflikten zwischen Entwicklung und Betrieb.

ÜBUNG: DEVOPS

Welche Aussagen stimmen?

- Dev steht für Developers und Ops für Operators
- DevOps reduziert die Spannungen zwischen Entwicklung und Betrieb
- DevOps ist nur im agilen Kontext möglich
- Dev steht für Development und Ops für Operations

ÜBUNG: DEVOPS IST ...

- eine Kultur
- ein Organisationsstil
- neuer Denkansatz
- ein Titel
- ein Tool-Stack
- ein Job-Titel

ÜBUNG: DEVOPS PRINZIPIEN

Markieren Sie Prinzipien des DevOps

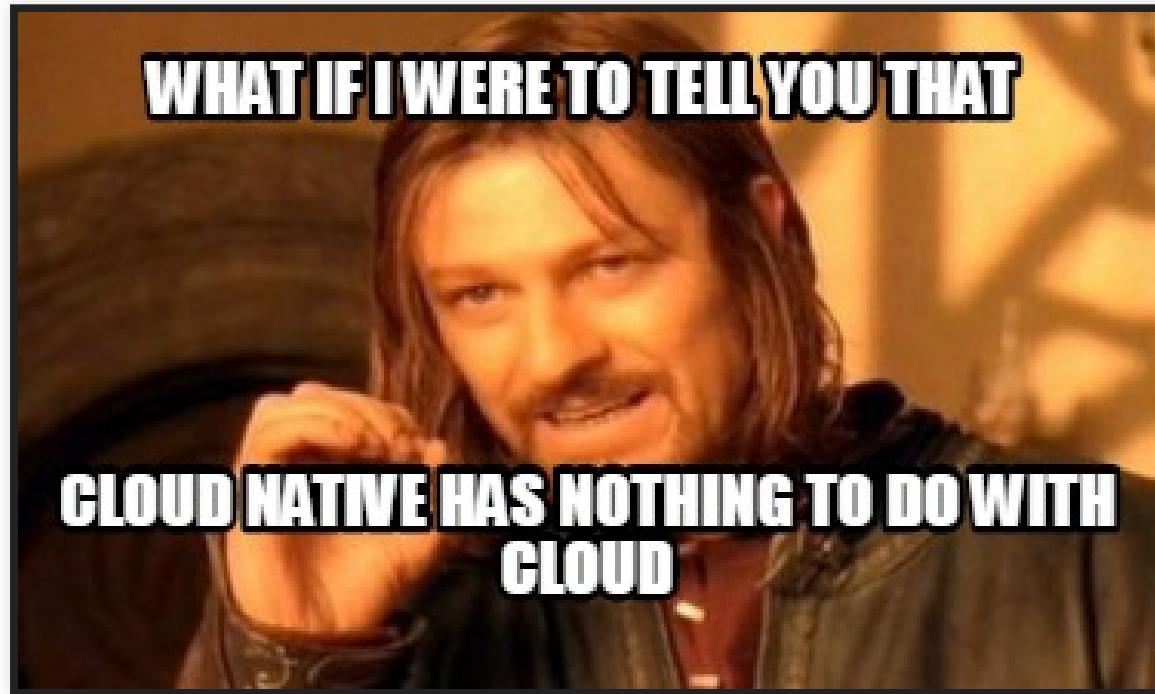
- Entwicklung und Test gegen speziellen Test-Systeme
- Schnelles Deployment durch verbesserte Prozessen
- Überwachung und Validierung der Betriebsqualität
- Rückkopplungsschleifen verstärken

ÜBUNG: DEVOPS LEBENSZYKLUS

Der Lebenszyklus besteht aus?

- Anforderungsauswahl > Umsetzung > Test > Inbetriebnahme
- Fehler-Identifikation > Entwickeln und Testen > Release und Deploy > Feedback-Analyse
- Planen und Messen > Entwickeln und Testen > Release und Deploy > Überwachen und Optimieren

2.6 CLOUD-NATIVE, CLOUD-ARGNOSTIC UND MULTI-CLOUD





Cloud-Native-Anwendungen laufen in der Cloud – darüber gibt es keine Diskussionen. Aber einfach nur eine bestehende Anwendung zu nehmen und sie auf einer Recheninstanz in der Cloud laufen zu lassen, macht sie nicht Cloud Native.

John Arundel und Justin Domingus: Cloud Native DevOps mit Kubernetes.



Cloud native technologies empower organizations to build and run scalable applications in modern, dynamic environments such as public, private, and hybrid clouds.

Containers, service meshes, microservices, immutable infrastructure, and declarative APIs exemplify this approach.

Quelle: <https://www.cncf.io/about/who-we-are/>



Was umfasst Cloud-Native?



Was umfasst Cloud-Native?

Automatisierbar



Was umfasst Cloud-Native?

Automatisierbar

Überall einsetzbar und flexibel



Was umfasst Cloud-Native?

Automatisierbar

Überall einsetzbar und flexibel

Widerstandsfähig und skalierbar



Was umfasst Cloud-Native?

Automatisierbar

Überall einsetzbar und flexibel

Widerstandsfähig und skalierbar

Dynamisch



Was umfasst Cloud-Native?

Automatisierbar

Überall einsetzbar und flexibel

Widerstandsfähig und skalierbar

Dynamisch

Observability



Was umfasst Cloud-Native?

Automatisierbar

Überall einsetzbar und flexibel

Widerstandsfähig und skalierbar

Dynamisch

Observability

Verteilt



Automatisierbar

Wenn Anwendungen nicht von Menschen, sondern von Maschinen deployt und verwaltet werden sollen, müssen sie verbreitete Standards, Formate und Schnittstellen erfüllen. Kubernetes bietet diese Standard-Schnittstellen auf eine Art und Weise an, dass Anwendungsentwickler sich gar nicht darum kümmern müssen.

John Arundel und Justin Domingus: Cloud Native DevOps mit Kubernetes.



Überall einsetzbar und flexibel

Weil sie von realen Ressourcen wie Festplatten oder spezifischem Wissen über den Rechenknoten entkoppelt sind, auf dem sie laufen werden, können Containerisierte Microservices leicht von einem Knoten auf einen anderen verschoben werden – oder sogar von einem Cluster in ein anderes.

John Arundel und Justin Domingus: Cloud Native DevOps mit Kubernetes.



Widerstandsfähig und skalierbar

Klassische Anwendungen tendieren dazu, einzelne Schwachstellen (Single Points of Failure) zu entwickeln: Die Anwendung läuft nicht mehr, wenn ihr Hauptprozess abstürzt, die zugrunde liegende Maschine einen Hardware-Defekt hat oder eine Netzwerk-Ressource überlastet ist. Cloud-Native-Anwendungen lassen sich durch Redundanz und Graceful Degradation hochverfügbar machen, weil sie schon inhärent verteilt sind.

John Arundel und Justin Domingus: Cloud Native DevOps mit Kubernetes.



Dynamisch

Ein Container-Orchestrator wie Kubernetes kann Container so schedulen, dass sie die verfügbaren Ressourcen maximal ausnutzen. Er kann viele Kopien von ihnen laufen lassen, um eine hohe Verfügbarkeit zu erreichen, und rollierende Updates durchführen, um Services still und leise zu aktualisieren, ohne irgendwelchen Traffic zu verlieren.

John Arundel und Justin Domingus: Cloud Native DevOps mit Kubernetes.



Observability

Cloud-Native-Anwendungen lassen sich aufgrund ihrer Natur schlechter inspizieren und debuggen. Daher ist eine zentrale Anforderung verteilter Systeme ihre Beobachtbarkeit (Observability): Monitoring, Logging, Tracing und Metriken helfen Entwicklern und Technikern dabei, zu verstehen, was ihre Systeme tun (und warum sie sich falsch verhalten).

John Arundel und Justin Domingus: Cloud Native DevOps mit Kubernetes.



Verteilt

Cloud Native ist ein Ansatz zum Bauen und Ausführen von Anwendungen, der sich die Vorteile der verteilten und dezentralisierten Natur der Cloud zunutze macht. Es geht nicht darum, wo Ihre Anwendung läuft, sondern wie sie funktioniert. Statt Ihren Code als einzelne Entität zu deployen (auch als Monolith bekannt), tendieren Cloud-Native-Anwendungen dazu, aus vielen miteinander kooperierenden, verteilten Microservices zusammengesetzt zu sein. Ein Microservice ist einfach ein in sich abgeschlossener Service, der eine Aufgabe erledigt. Bringen Sie genug Microservices zusammen, erhalten Sie eine Anwendung.

John Arundel und Justin Domingus: Cloud Native DevOps mit Kubernetes.



Cloud Native Computing Foundation

Technologien und Projekte im Kontext von Cloud Native

- [Landscape](#)
- [Radars](#)



Was ist Cloud-Agnostic?

Cloud agnostic refers to tools, platforms, or applications that are compatible with any cloud infrastructure and can be moved to and from different cloud environments without any operational issues.

Quelle: <https://www.cloudzero.com/blog/cloud-agnostic>



Ziel ist es eine Anwendung so zu entwickeln, dass sie in jeder Cloud ausgeführt werden kann. Dafür kommt der Einsatz von Werkzeugen, Platformen und Anwendungen zum Tragen, die sich in verschiedenen Clouds wiederfinden. Dies sind u.a. zum Beispiel Container-Technologien und entsprechende Orchestrierungsplattformen wie Kubernetes.

Hierbei sind typische Fragen entscheidend, wie Service-Discovery, Gateways und Load-Balancers, Skalierung und Persistierung von Daten. In der Entwicklung sind entsprechende Entscheidungen hinsichtlich der Architektur und den Abhängigkeiten zu treffen, die es erlauben ohne weitere Anpassungen ein Rollout bei verschiedenen Cloud-Anbietern zu ermöglichen.



Was ist Multi-Cloud?

Multi-Cloud bedeutet [...] Hyperscaler verschiedener Hersteller einzusetzen. In den meisten Fällen werden zwei Szenarien abgebildet: Entweder entscheidet sich das Unternehmen für jeden Workload, je nach Situation, einen anderen Cloud-Anbieter auszuwählen. Oder ein Workload wird über eine Abstraktion der Cloud-Services auf verschiedene Cloudprovider verteilt.

Quelle: <https://www.adesso.de/de/news/blog/cloud-native-multi-cloud-oder-doch-lieber-cloud-agnostic.jsp>



ÜBUNG: CLOUD-NATIVE

Welche Aussagen stimmen?

- Cloud-Native Anwendungen machen sich die verteilte und dezentrale Struktur der Cloud zunutze
- Cloud-Native Anwendungen lassen sich nur in einer speziellen Cloud betrieben, z.B. AWS
- Skalierbarkeit steht nicht im Zentrum von Cloud-Native Anwendungen
- Anwendungen die auf einer virtuellen Maschine, die in der Cloud läuft, betrieben werden, sind Cloud-Native
- Cloud-Native-Anwendungen laufen in der Cloud, nicht jede Anwendung ist Cloud-Native
- Cloud-Native Anwendungen setzen auf Cloud-basierte Dienste (Load-Balancing, Service Discovery, Skalierung, usw.)

2.7 REFERENZEN



- Sanjeev Sharama und Bernie Coyne: DevOps für Dummies - 2. limitierte auflage von IBM. John Wiley & Sons, Inc.
- Michael Hüttermann: DevOps for Developers. Apress. 2012.
- Arundel, J., & Domingus, J. (2019): [Cloud Native DevOps with Kubernetes: building, deploying, and scaling modern applications in the Cloud](#). O'Reilly Media.