

Dr. Stefan Fütterling – The Open Group Distinguished Architect – stefan.fuetterling@capgemini.com
Martin Bachmaier – IT Versatilist – mbachmaier@lenovo.com
Daniel Amor - The Open Group Master Architect - danny@dxc.com



Duale Hochschule Baden-Württemberg Stuttgart
IT Architekturen, 2025-10

IT Architekturen

Gesamtsicht der Vorlesung

▪ Einführung

- 1.1 Einführung in IT Architektur
- 1.2 Dynamische IT Infrastrukturen
- 1.3 Cloud Computing (mit Übungsaufgabe)

▪ Server Virtualisierung

- 2.1 Einführung in die Server Konsolidierung und Virtualisierung
- 2.2 Virtuelle Maschinen (VMs) am Beispiel VMware vSphere (ESXi)
- 2.3 OS Containers am Beispiel Linux LXC und Docker (mit Übungsaufgabe)
- 2.4 Deep Dive x86 Virtualisierung (Typ 1 Hypervisor)

▪ Zentralisierter Storage

- 3.1 Storage Area Networks (SAN) und Network Attached Storage (NAS)
- 3.2 RAID Levels
- 3.3 Disksysteme und Hyperconverged Infrastructure

▪ Clusterarchitekturen

- 4.1 Einführung in Clusterarchitekturen (LB-Cluster, HPC Cluster, HA Cluster)
- 4.2 Scale Out Data Center
- 4.3 Clustersoftware am Beispiel parallele Datenbanksysteme und Big Data Analysis Cluster (Hadoop)

▪ IT Betrieb

- 5.1 Überblick DevOps, Application Management und Systems Management
- 5.2 IT Service Management (ITIL)



Organisatorisches

■ Vorlesung

- 10x Dienstag und/oder Freitag
- Vorlesungstermine werden im Rapla bekannt gegeben
- Nicht benötigte Termine werden rechtzeitig aus dem Rapla entfernt

■ Prüfungsleistung

- Case Study
- Bearbeitung in 3er Gruppen
- Referat 20 Minuten
- Benotet

■ Spielregeln

- Geräuschpegel bitte niedrig halten
- Fragen sind jederzeit willkommen

■ Kursunterlagen

- Werden zu den Vorlesungsterminen auf Moodle bereitgestellt

Kursinhalt und Organisatorisches

- **Vorstellungsrunde**

Name

Vorkenntnisse, Praktikumserfahrungen

(allgemein und zu Themen der Vorlesung)

Interessen

Erwartungen an die Vorlesung

...

1.1 Einführung in IT Architektur

Wie entsteht eine IT Lösung?

- **Business Strategie**
- **Geschäftsprozesse**
- **Business Capabilities**
- **Leistungsumfang der IT Lösung**
- **Anwendungsentwicklung**
- **IT Infrastruktur**
- **IT Betrieb**



Business Consultants

Business Sicht /
Business Anforderungen
(Fokus auf Geschäftsprozesse)

Business Architekten

Anwendungs-Architekten

Funktionale Sicht /
Funktionale Anforderungen
(Fokus auf die Anwendung)

Infrastruktur-Architekten

Operationale Sicht /
Nicht-funktionale Anforderungen
(Fokus auf Infrastruktur und Betriebsthemen)

Enterprise Architekten

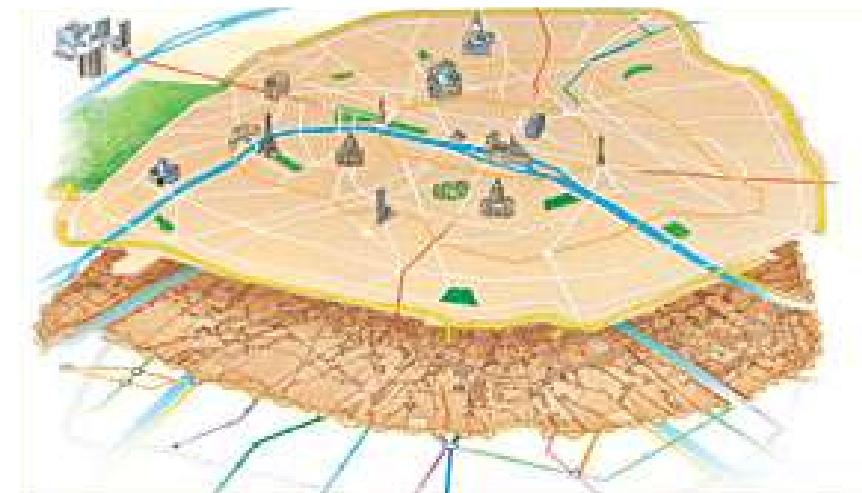
Beispiel

Business Strategie	Ein Versandhändler für Bekleidung möchte zukünftig auch Reisen verkaufen Die Geschäftsleitung beschließt, dazu mit einem Reiseveranstalter zu kooperieren
Geschäftsprozesse und geschäftliche Fähigkeiten	Auswahl des Reiseziels (Katalog mit Bildern und Beschreibungen) Auswahl Reisedatum und Prüfen der Verfügbarkeit Buchen Bezahlen, Reiserücktrittversicherung abschließen, Umbuchen, ...
Geschäftliche Anforderungen	Webshop soll rund um die Uhr erreichbar sein 2000 Kunden gleichzeitig auf den Reise-Webseiten, 300 Reisebuchungen pro Tag
IT Anforderungen	250 Hotels in 14 Ländern, ca. 1,5 TB Katalogdaten
Randbedingungen	Sehr gutes Antwortzeitverhalten beim Sichten der Katalogdaten Das IT System des Reiseveranstalters befindet sich in dessen eigenem Rechenzentrum in einer ca. 200 km entfernten Lokation
Leistungsumfang der IT Lösung	Erweiterung des bestehenden Webshops
Anwendungsentwicklung	Anzeigen der Katalogdaten => Großes Datenvolumen, geringe Änderungsrate => Daten sollten lokal vorliegen, also repliziert werden
	Prüfen der Verfügbarkeit => Sollte direkt gegen das IT System des Reiseveranstalters erfolgen => Kleines Datenvolumen, hohe Änderungsrate => API vorhanden?
IT Infrastruktur und Betrieb	Zusätzliche Webserver Storage bzw. Datenbank für die replizierten Katalogdaten Ausfallsichere Netzwerkanbindung zum Rechenzentrum des Reiseveranstalters Monitoring der Netzwerkanbindung

Arten von IT Architektur

- Die **Enterprise Architektur** macht **übergreifende Vorgaben** für alle IT Projekte eines Unternehmens, z.B.:

- Festlegung Warenkörbe für HW und SW
- Vorgaben für Plattformen
- Vorgaben für IT Sicherheit, z.B. Datenklassifikation
- Management von Anwendungslandschaften
- Festlegung von Architekturprinzipien
- Vorgaben für die Softwareentwicklung
- Überwachung (Governance)



- Die **Anwendungs-Architektur** und die **Infrastruktur-Architektur** entwerfen gemeinsam die funktionale und die operationale Seite eines IT Systems:

- funktionale Anforderungen an das IT System
- Anwendungskomponenten
- Benutzerschnittstellen
- Datenstrukturen
- Kopplung von Anwendungen
- nicht-funktionale Anforderungen eines IT Systems
- operationale Modelle zur Festlegung der Infrastruktur (bestehend aus Netzwerken, Servern und Storage)
- Tools und Verfahren für den IT Betrieb

Schwerpunkt
der Vorlesung

Was kann passieren, wenn die IT Architektur nicht ausreichend berücksichtigt wird?



Startseite > Wirtschaft > Technik von gestern : Banken droht der Infarkt

WIRTSCHAFT

Technik von gestern

Banken droht der Infarkt

Aus dem Artikel:

In ihren Rechenzentren setzen Banken auf ihre **veraltete Technik**. Doch wie lange geht das noch gut? "Große Finanzinstitute sind wie Museen der Technologie", warnt der frühere Barclays-Boss Jenkin.

Kaum ein Außenstehender bekommt sie jemals zu Gesicht. Doch in atombombensicheren Rechenzentren entscheidet sich womöglich die Zukunft der Bankenbranche. Denn die oft **jahrzehntealte Software** vieler Institute kommt angesichts der fortschreitenden Digitalisierung an ihre Grenzen.

Die Informatik ist Analysten zufolge mit einem Anteil von 15 bis 25 Prozent der **zweitgrößte Kostenblock** der meisten Geldhäuser. Bei Großbanken geht das in die Milliarden. Bis zu **80 Prozent der Informatik-Kosten geben die Institute aber für den Unterhalt der bestehenden Technologie** aus, schätzt Ben Robinson von der Bankensoftwarefirma Temenos. Damit bleibt oft zu wenig Geld für neue Angebote. Die IT-Budgets dürften in Zukunft deshalb eher noch aufgestockt werden.

Quelle: <https://www.n-tv.de/wirtschaft/Banken-droht-der-Infarkt-article20203964.html>

Was kann passieren, wenn die IT Architektur nicht ausreichend berücksichtigt wird?

Teurer technischer Wildwuchs

Bahn nutzt 3700 IT-Systeme

»Veraltet«, »uneinheitlich«, »Do-it-yourself«: Im Bahn-Konzern macht sich laut »Business Insider« Unmut über die IT-Struktur breit. Die Technik sei unnötig kompliziert und teuer.

01.03.2024, 16.36 Uhr

Aus dem Artikel:

- “Hochrangige Bahn-Insider sprechen laut dem Bericht von einem »**IT-Chaos**«, das sich durch alle Tochtergesellschaften ziehe.”
- “Im Vorstand der Bahn sei von veralteter, uneinheitlicher »**Do-it-yourself-IT**« die Rede.”
- “Zwar gebe es Pläne, die **Anzahl der IT-Systeme zu reduzieren** und auf eine **gemeinsame Plattform** zu übertragen; Manager hegten aber Zweifel an der Umsetzung und dem Tempo dieses Vorhabens, hieß es weiter.”

Quelle: <https://www.spiegel.de/wirtschaft/unternehmen/bahn-nutzt-3700-it-systeme-teurer-technischer-wildwuchs-a-92c3cdf5-b58d-4b7e-978b-2ea4ca0010ac>

Die vier Domänen des Architekturmodells nach TOGAF (The Open Group Architecture Framework)

Die **Business Architecture Domain** definiert die Geschäftsstrategie, Geschäftsfähigkeiten, Geschäftsprozesse, Organisation und Verantwortlichkeiten.

Geschäftsprozesse
(SCM, CRM, etc.)

Organisation

Portfolio

Standorte

Business
Anforderungen

Die **Data Architecture Domain** ermöglicht die Entkopplung der Business Architektur von der Application Arch. und bietet die Basis für Service Orientierte Architekturen.

Logisches Datenmodell

Physisches Datenmodell

Datenflüsse

Die **Application Architecture Domain** beschreibt die IT Applikations- und IT-Integrations-landschaft in Form von Bebauungsplänen, deren Bezug zu den Geschäftsprozessen und Informationsflüssen.

Logisches
Informationssystem

Physisches
Informationssystem

Anwendungskopplung
(API, Messaging, BPM, etc.)

Funktionale
Anforderungen

Die **Technology Architecture Domain** definiert die IT Infrastruktur und Betriebsverfahren zur Bereitstellung der benötigten IT-Funktionen.

Server

Storage

Netzwerk

Virtualisierung

Cluster

Standards

Data Center

Middleware

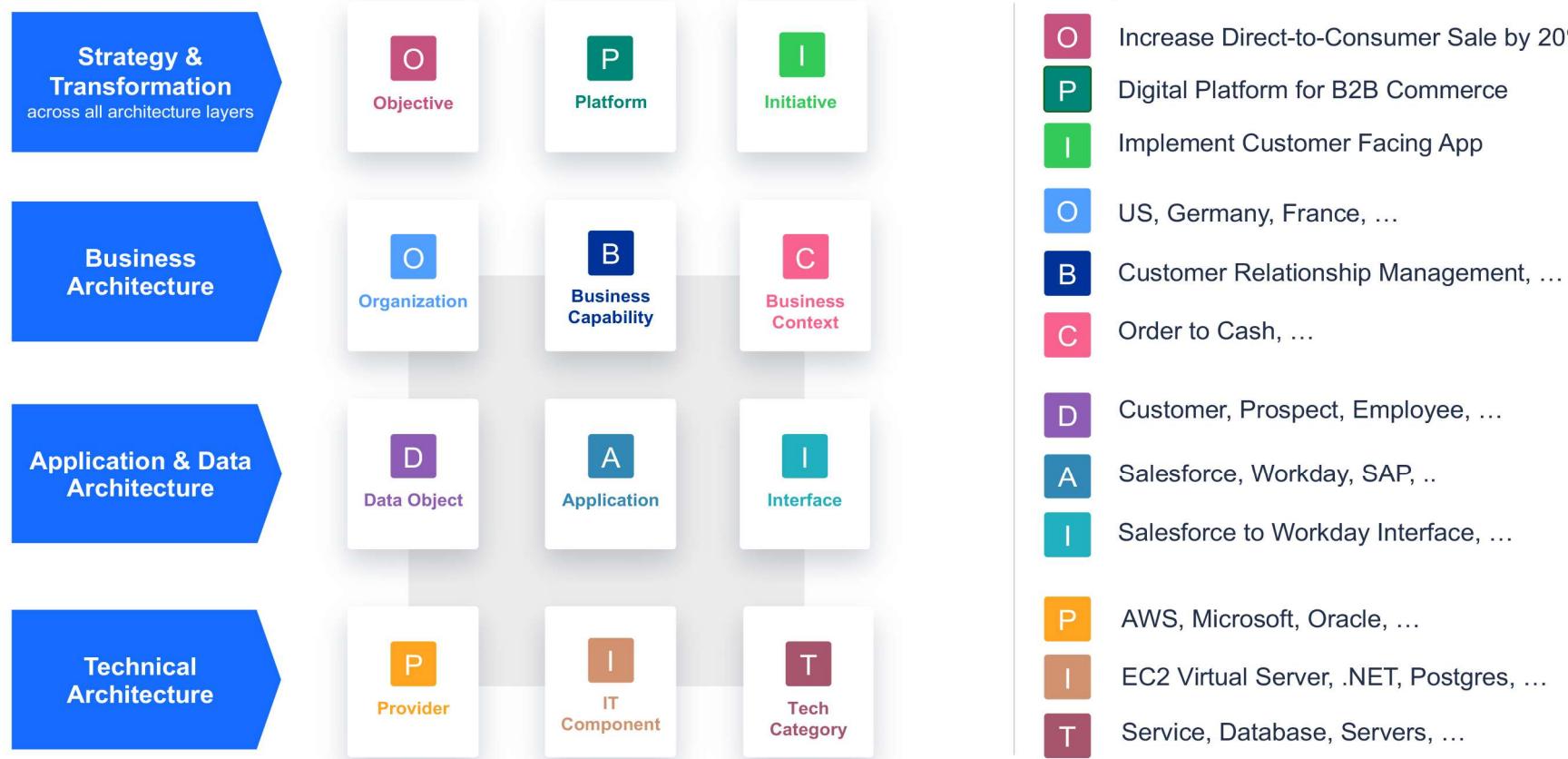
Tools

IT Betriebsprozesse

Nicht-funktionale
Anforderungen

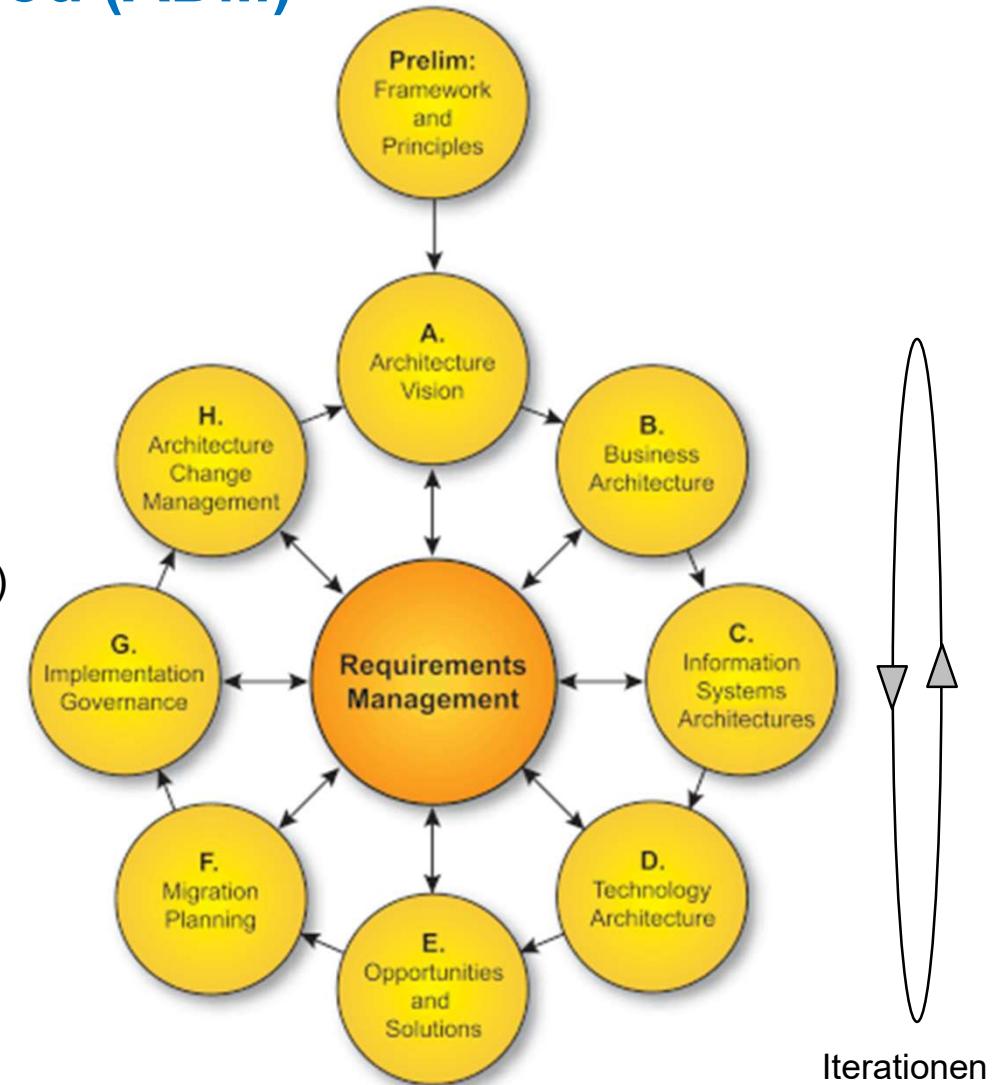
Schwerpunkt
der Vorlesung

Abbildung der Enterprise Architektur in EAM Tools: Beispiel LeanIX Metamodell



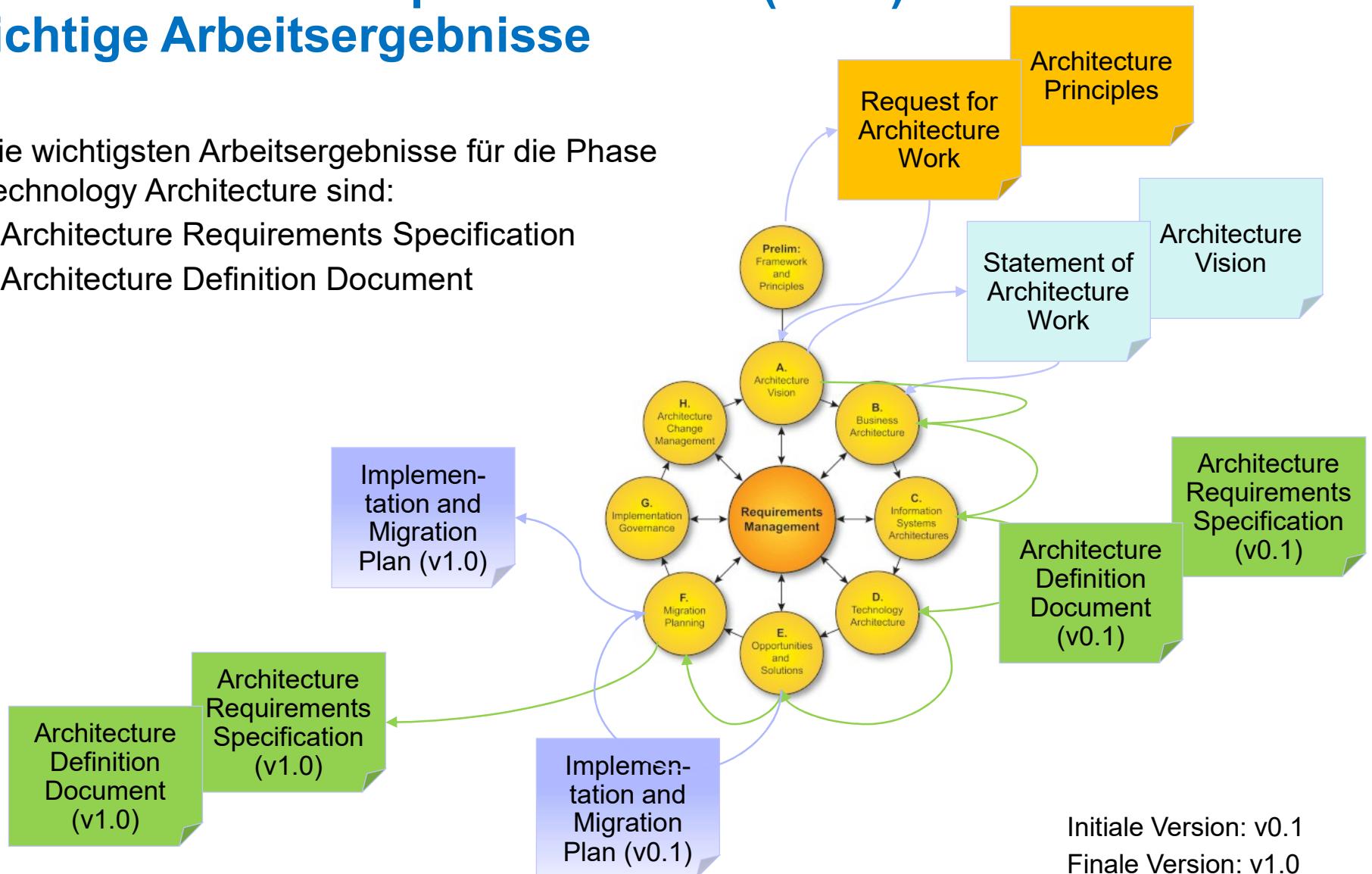
Architektur Methoden am Beispiel TOGAF: Architecture Development Method (ADM)

- Die TOGAF Architecture Development Method (ADM) definiert eine **strukturierte Vorgehensweise zur Entwicklung von Architekturen**
- ADM gliedert den Gesamtablauf in **Phasen**
- Die 4 Architektur Domänen finden sich in
 - Phase B = Business Architecture
 - Phase C = Information Systems Architecture (beinhaltet Data und Application Architecture)
 - Phase D = Technology Architecture
- ADM definiert die **Arbeitsergebnisse** (Artefakte) für jede Phase
- Die Phasen werden **iterativ** durchlaufen, um die Architektur schrittweise zu verfeinern, z.B. mehrere Durchläufe in den Phasen B bis D



Architecture Development Method (ADM): Wichtige Arbeitsergebnisse

- Die wichtigsten Arbeitsergebnisse für die Phase Technology Architecture sind:
 - Architecture Requirements Specification
 - Architecture Definition Document



Requirements Management: Nicht-funktionale Anforderungen

Architecture
Requirements
Specification

- **Beispiele für Anforderungen, die das laufende System erfüllen muss:**

Leistungsfähigkeit (Performance), z.B. Transaktionsrate einer Datenbank
Kapazität (Capacity), z.B. Anzahl der verbundenen Benutzer, Datenvolumen
Verfügbarkeit (Availability), z.B. %-Angabe bezogen auf die Servicezeit
Sicherheit (Security, Safety), z.B. Benutzerverwaltung oder Datensicherheit
Systems Management, z.B. Monitoring, Backup

- **Beispiele für Anforderungen, die Änderungen am System betreffen:**

Wartbarkeit (Maintainability), z.B. Fehleranalyse, Einspielen von Softwareupdates
Portabilität (Portability), z.B. Betriebssystem-unabhängiges Design
Skalierbarkeit (Scalability), z.B. bei ansteigender Benutzeranzahl

- **Beispiele für Anforderungen, die die Lösung beeinflussen oder einschränken:**

Geschäftsbedingte Anforderungen, z.B. weltweiter Zugriff
Technische Vorgaben, z.B. Produktkataloge

Technology Architektur: Operationale Modelle

Architecture
Definition
Document

- **Diagramme und Tabellen zur Darstellung des Gesamtsystems**

Standorte und geografische Verteilung, Lokationen

Netzwerktopologie und Zonen

Server (Nodes) mit Betriebssystem

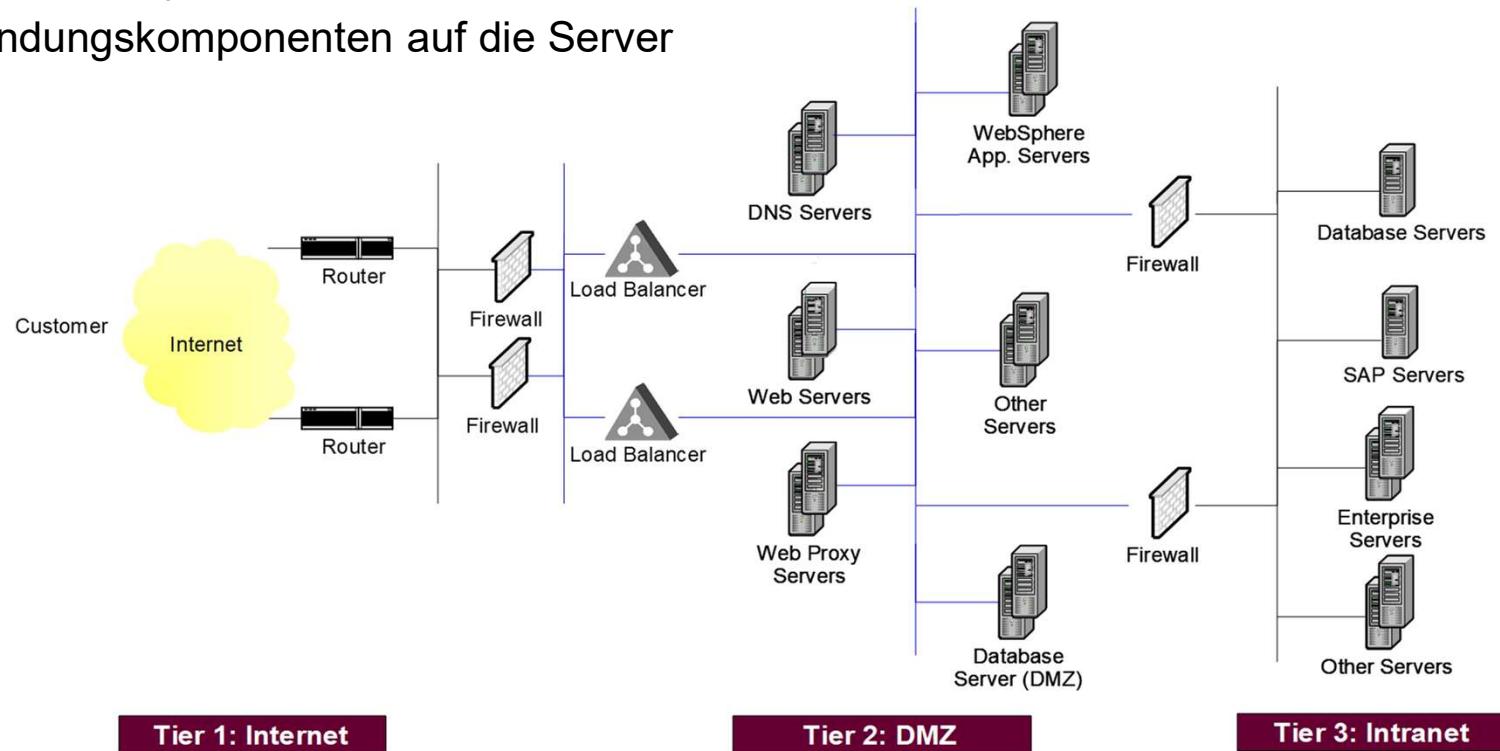
Verteilung der Anwendungskomponenten auf die Server

Middleware

Benutzerzugänge

- **Detaillierung:**

- Logischer Level
- Physischer Level
(Produktauswahl,
Sizingangaben)



Dr. Stefan Fütterling – The Open Group Distinguished Architect – stefan.fuetterling@capgemini.com
Martin Bachmaier – IT Versatilist – mbachmaier@lenovo.com
Daniel Amor - The Open Group Master Architect - danny@dxc.com



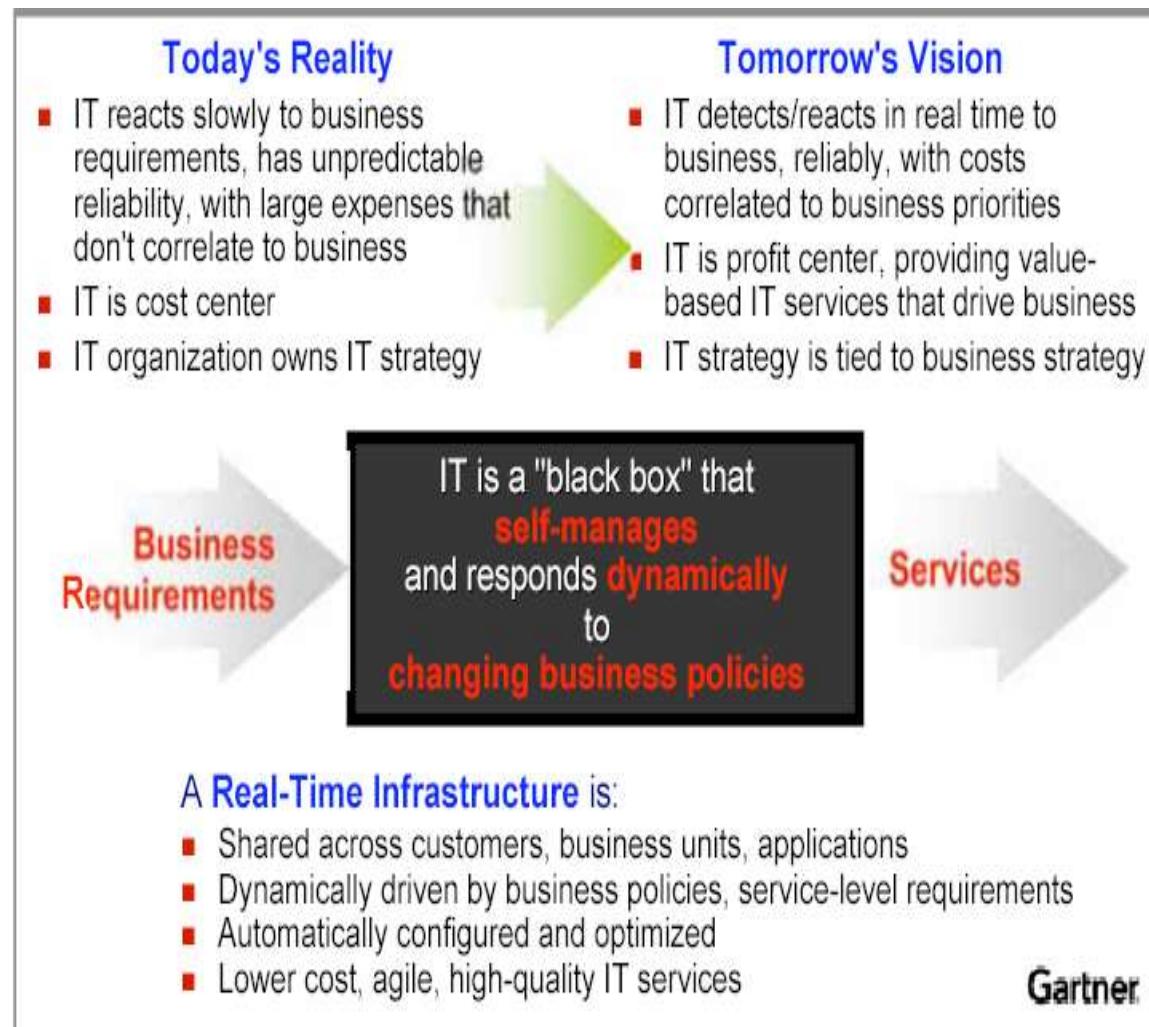
Duale Hochschule Baden-Württemberg Stuttgart
IT Architekturen, 2025-10

1.2 Dynamische IT Infrastrukturen

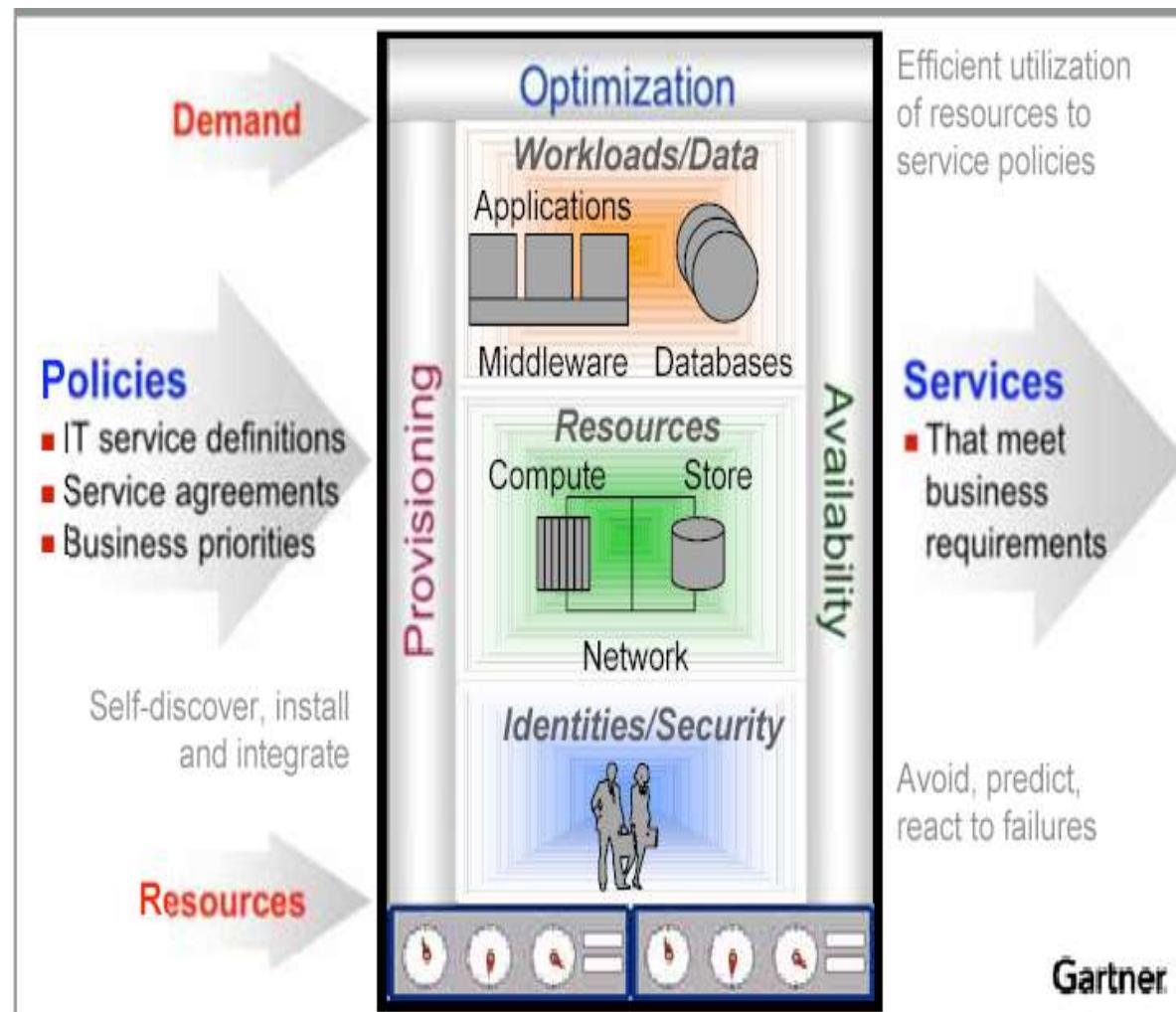
Gartner Studie zur Gegenwart und Zukunft der IT



Gartner Studie zu den “Real Time” IT Infrastrukturen

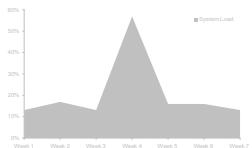


Gartner Studie zu den “Real Time” IT Infrastrukturen



Nachteile traditioneller IT Infrastrukturen

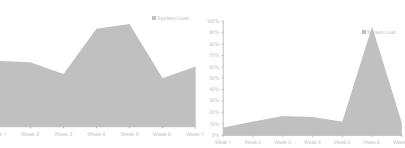
Dedicated Server 1



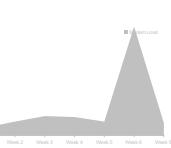
Dedicated Server 2



Dedicated Server 3

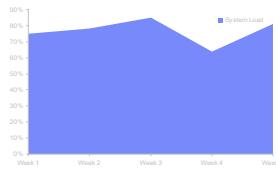


Dedicated Server 4

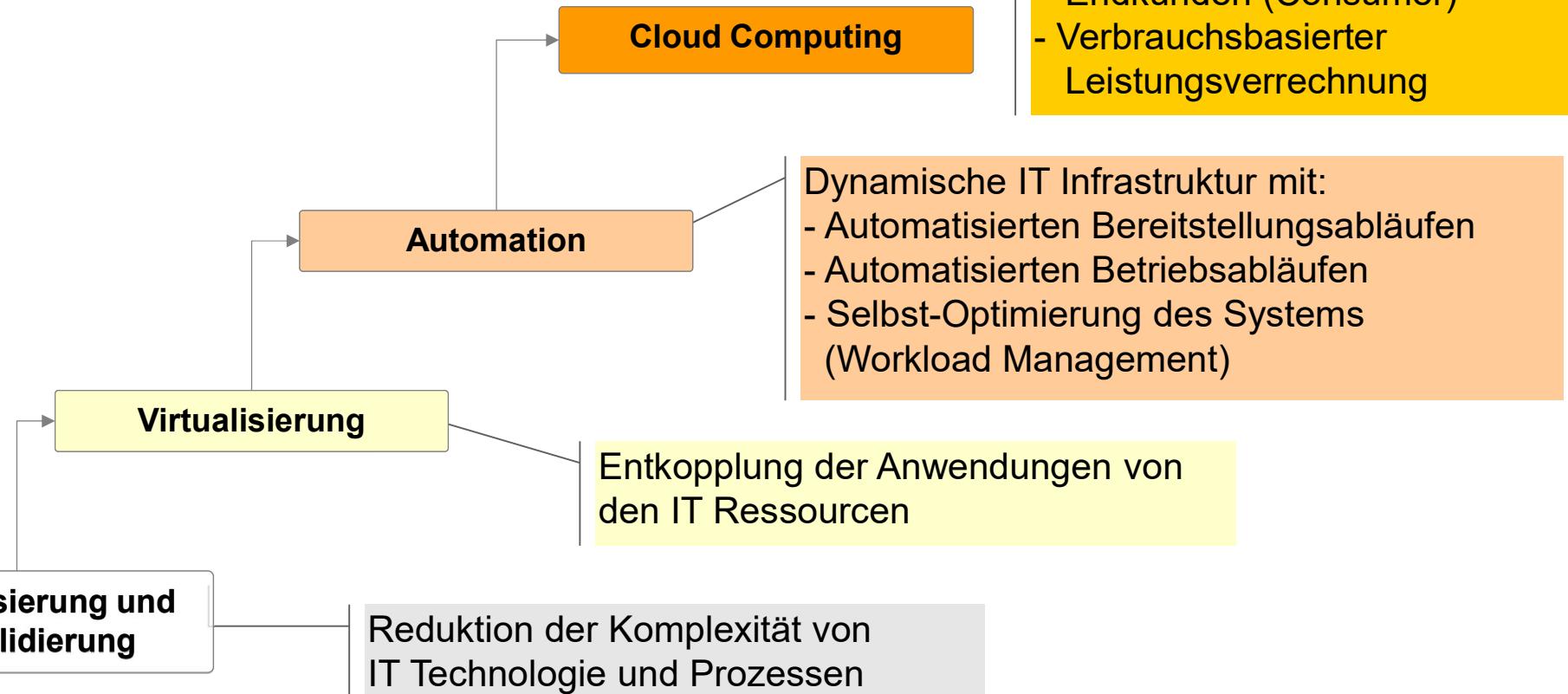


- Viele verschiedenartige Systeme, hohe Heterogenität und Komplexität
- Dedizierte Server pro Applikation (Isolation)
- Geringe Auslastung der Systeme (nur etwa 15-50 %, sogenannte Underutilization)
- Systeme sind meist für Lastspitzen ausgelegt (Overprovisioning)
- Lange Bereitstellungszeiten für neue Server und Anwendungen (Slow Provisioning)

Shared Server



Entwicklungsstufen einer dynamischen IT Infrastruktur



Harmonisierung und Konsolidierung

■ Motivation

Manage fewer things better

Reduktion der Komplexität der IT Infrastruktur (Technologie)

Vereinfachung von Betriebsprozessen

■ Nutzen

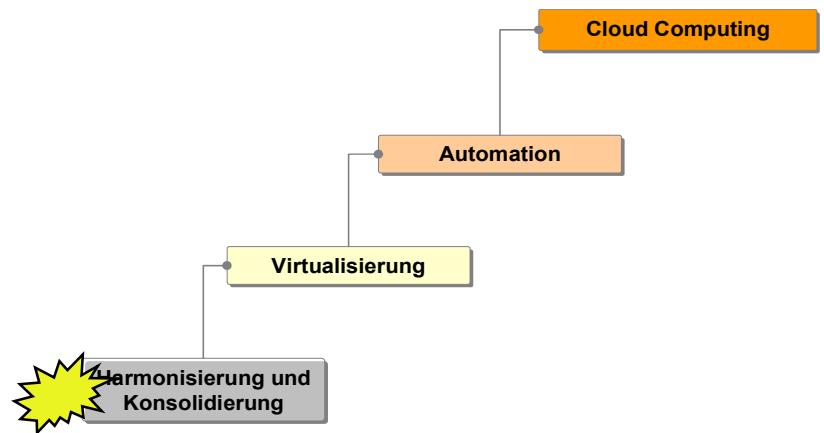
Höhere Flexibilität

Höhere Verfügbarkeit und Recovery Fähigkeit

Geringere Betriebs- und Wartungsaufwände

geringere Gesamtkosten

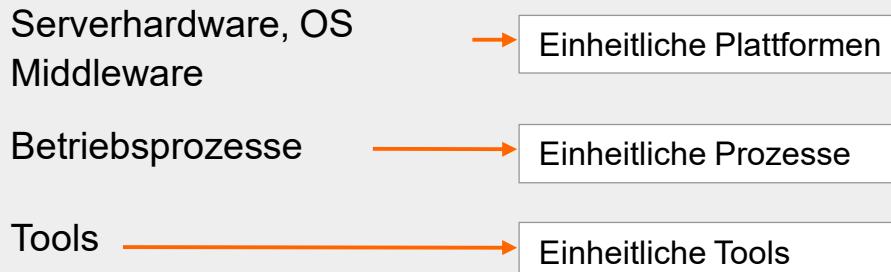
Voraussetzung für weitere Schritte



Handlungsfelder Harmonisierung und Konsolidierung

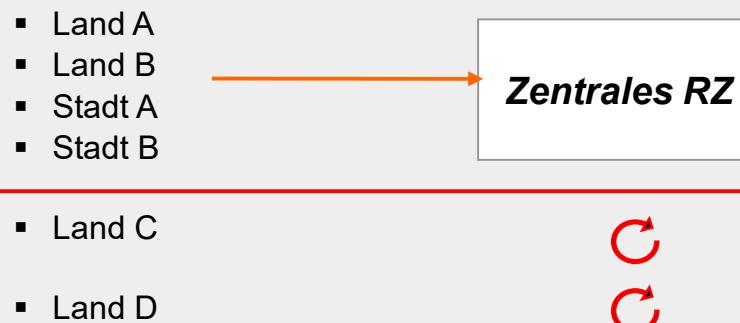
Harmonisierung und Standardisierung

Definition/Optimierung von einheitlichen Plattformen, Betriebsprozessen, Tools



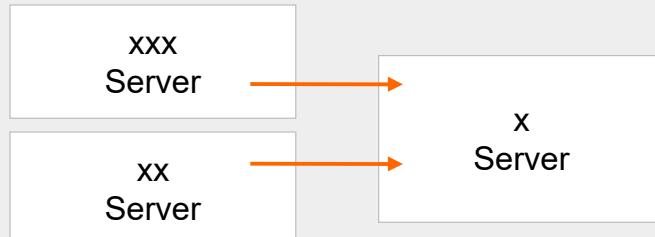
Rechenzentrums Konsolidierung

Umzug vorhandener Server in ein neues RZ



Physische Konsolidierung

Reduzierung der Anzahl von Servern durch den Ersatz vieler kleiner durch wenige große Server.



Zusammenlegung des Storage in Disksystemen
Reduzierung der Netzwerkkomponenten

Logische Konsolidierung

Betriebsprozesse/Organisation



Tools



Skills



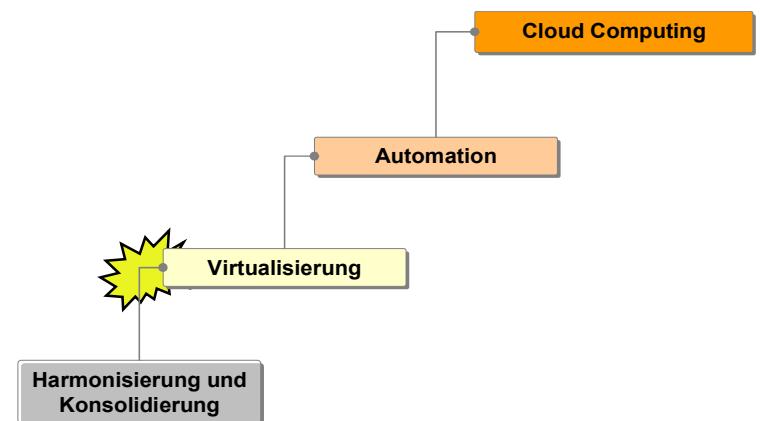
Virtualisierung

■ Motivation

Entkopplung der Anwendung von den IT Ressourcen (IT Infrastruktur)
z.B. Server, Storage, Netzwerke

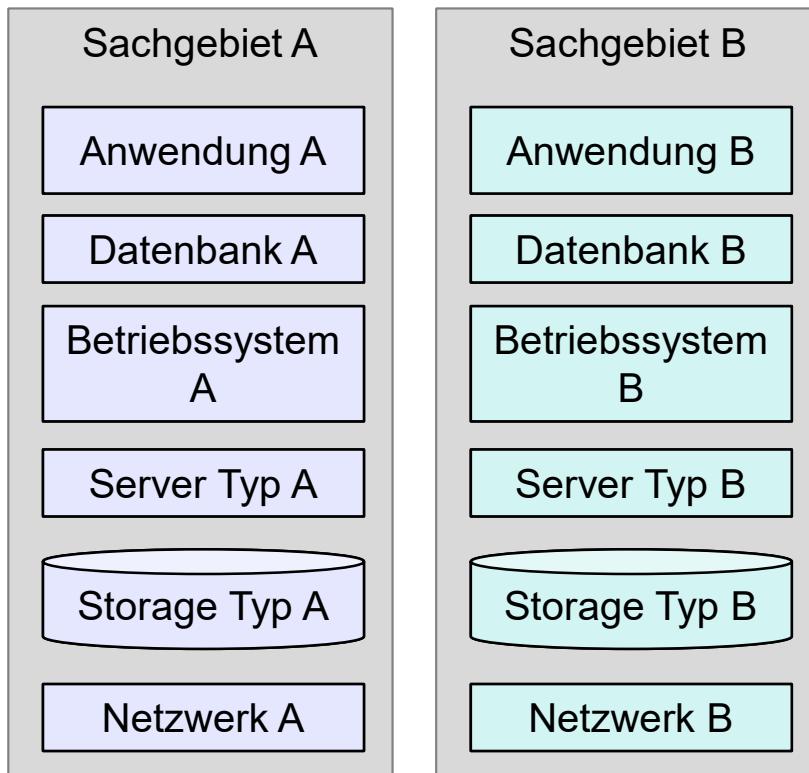
■ Nutzen

Höhere Flexibilität der IT Infrastruktur
Aufhebung von Hardwaregrenzen (z.B. beim Storage)
Bessere Ressourcenausnutzung
Schnellere Bereitstellung
Niedrigere Gesamtkosten
Voraussetzung für weitere Schritte

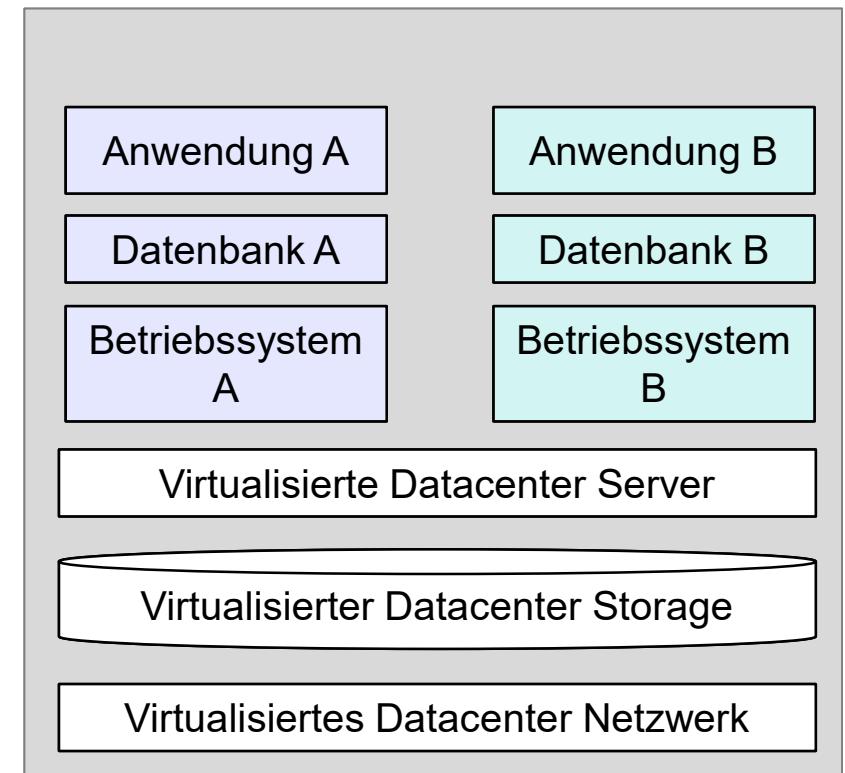


Virtualisierung: Entkopplung der Anwendungen von den physischen IT-Ressourcen

dedizierte IT-Ressourcen pro Anwendung
„Silos“



gemeinsame Verwendung der IT-Ressourcen für mehrere Anwendungen



„meine Server“ - „mein Storage“ - „mein Betriebsteam“

Beispiele für Virtualisierung

■ Server Virtualisierung

Mehrere Betriebssystem-Instanzen auf einem physischen Server

x86: VMware vSphere (ESXi), Citrix XEN Server, Microsoft HyperV, KVM

IBM System Z (Mainframe): LPAR und VM Technologie

Linux Containers

■ Storage Virtualisierung

Viele physische Disks in einem Storage-System zusammengefasst

Mehrere logische Speicherbereiche auf einem Storage-System

Network Attached Storage (NAS)

Storage Area Networks (SAN)

Beispiele für Virtualisierung

■ Netzwerk Virtualisierung

Mehrere logische Netzwerke auf einer physischen Netzwerkinfrastruktur

- Virtuelle LANs (VLANs)
- Virtuelle IP Adressen (VIP) auf Load Balancer
- Virtuelle private Netzwerke (VPN)

■ Anwendungsvirtualisierung

Mehrere Anwendungsinstanzen für eine Anwendung

Parallele Web Server, Application Server, Datenbanken etc.

Virtualisierte Anwendungen oder Arbeitsplätze

Zentral gehostete Anwendungen, z.B. Citrix XENapp, VMware View

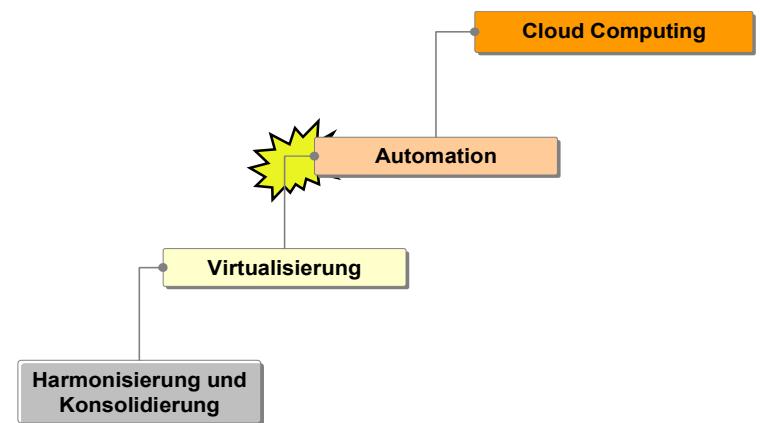
Automation

■ Motivation

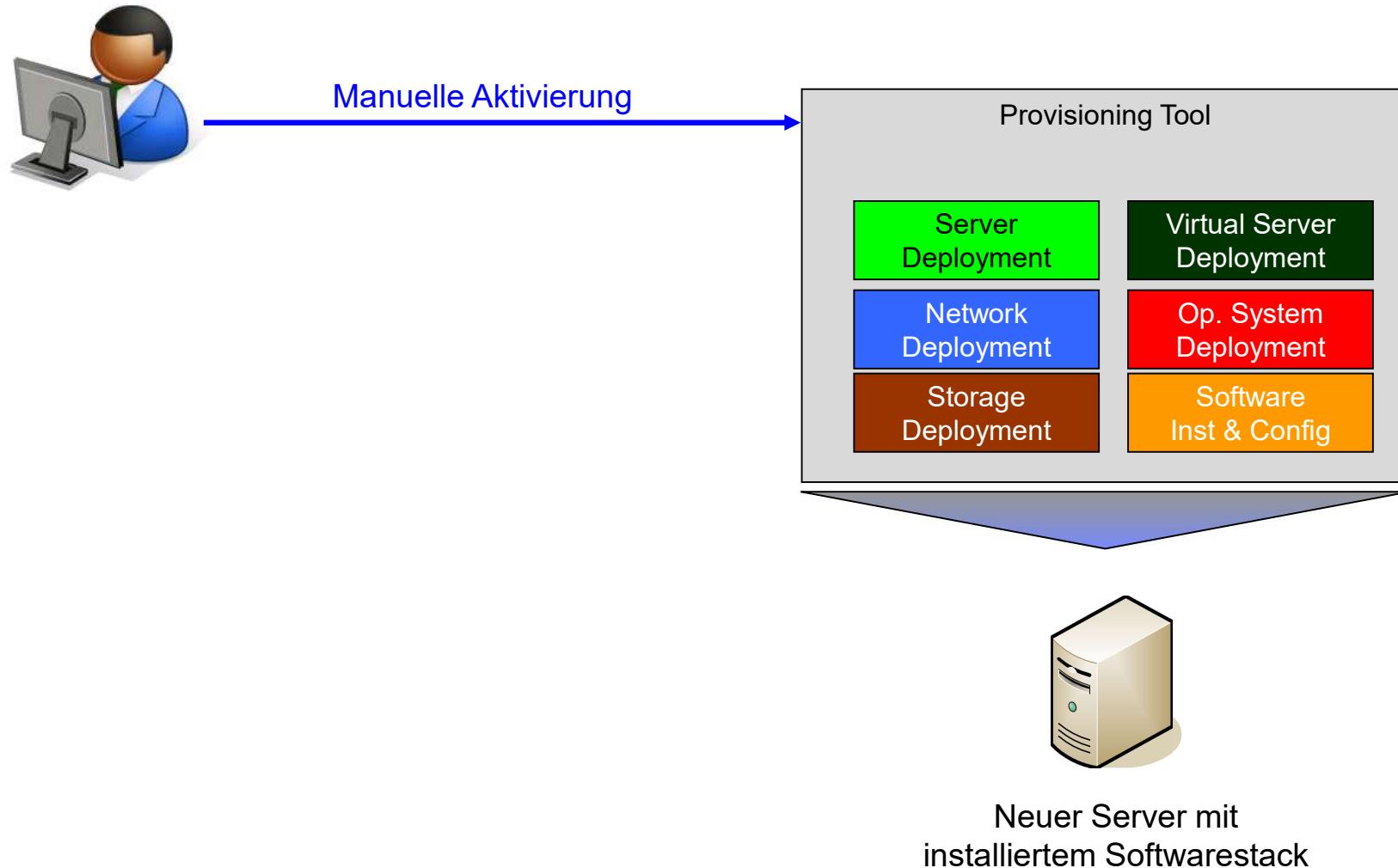
- Automatisierung von Bereitstellungs- und Betriebsabläufen
- Verringerung manueller Eingriffe
- Verkettung von Prozessschritten

■ Nutzen

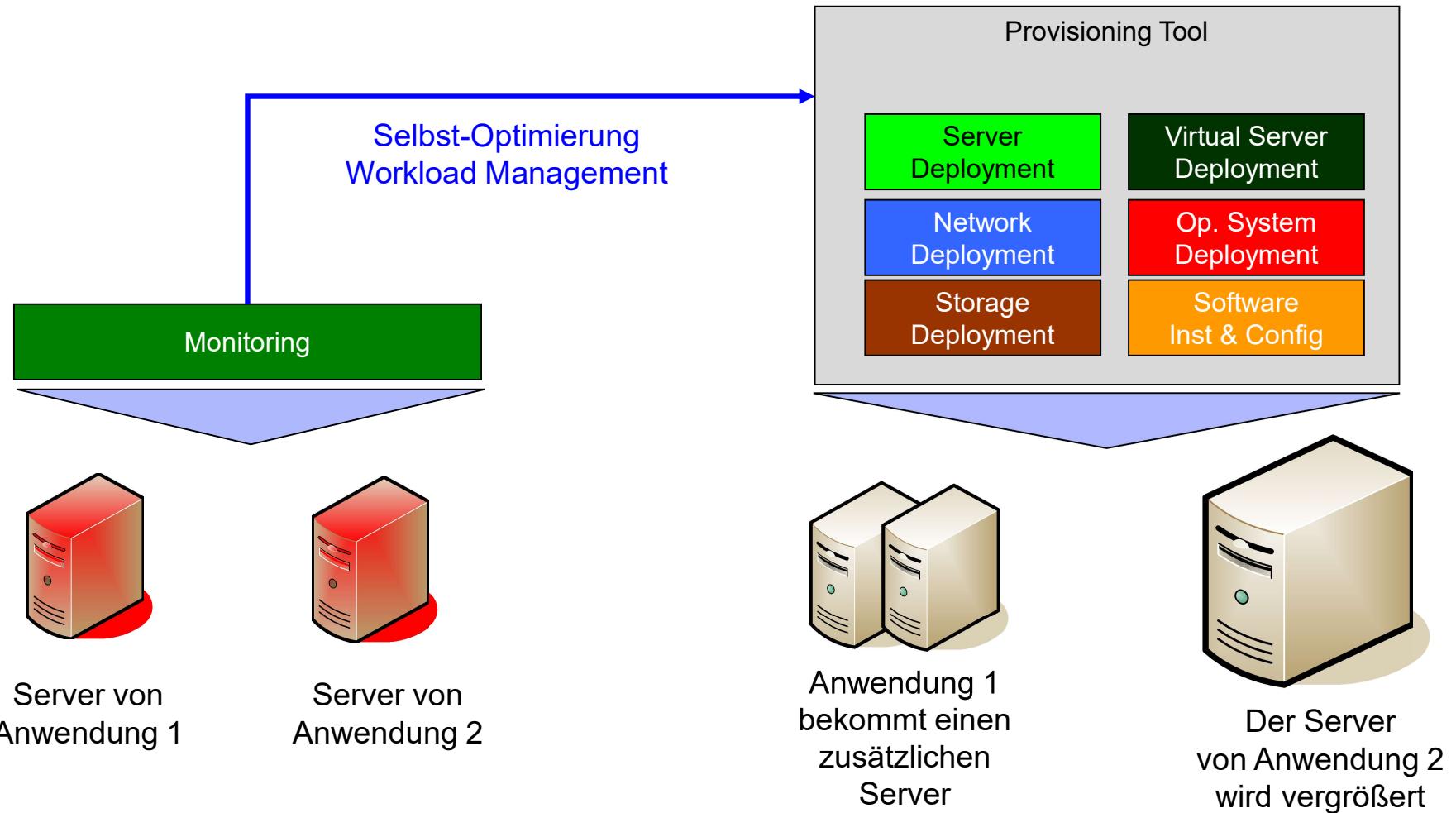
- Höherer Standardisierungsgrad
- Höhere Flexibilität der IT Infrastruktur
- Optimale Nutzung der Ressourcen
- Höherer Automatisierungsgrad
- Kürzere Bereitstellungszeiten
- Niedrigere Gesamtkosten
- Voraussetzung für weitere Schritte



Automatisierte Bereitstellung von IT Ressourcen (Provisioning) am Beispiel eines neuen Servers



Automatisierte Bereitstellung von IT Ressourcen (Provisioning) am Beispiel von zusätzlicher Rechenleistung



Architektur von Provisioning Tools

Orchestration und Workload Mgmt Komponente

Optimiert die Ressourcenzuweisung:

- Nimmt **Messwerte** aus der IT Umgebung, z.B. über ein Monitoring Tool
- Vergleicht mit definierten **Service Levels** (SLAs) und erkennt Engpass
- Ermittelt freie “passende” IT Ressourcen aus **Resource Pools**
- Sendet einen Deployment Request an die Deployment Komponente

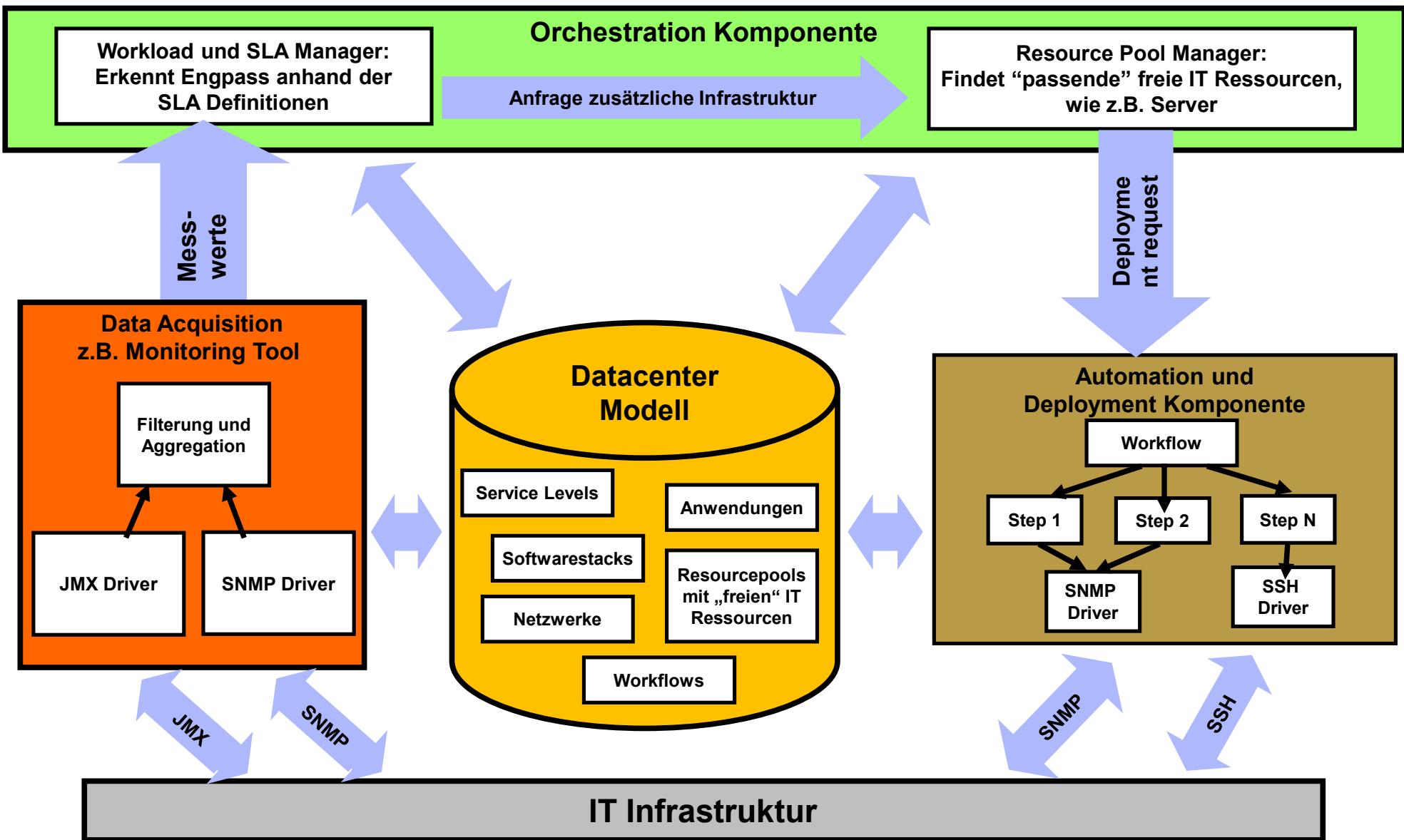
Accounting Komponente

Protokolliert, welche Anwendung wieviele Ressourcen verbraucht hat

Automation und Deployment Komponente

Führt die automatisierte Bereitstellung durch:

- Hat **Zugang** zu den IT Ressourcen (z.B. über Virtualisierungsschicht, SSH, SNMP, JDBC)
- Steuert den automatisierten Bereitstellungsablauf über anpassbare **Workflows**
- Protokolliert Änderungen im **Datacenter Modell**



Beispiel: Provisionierung eines zusätzlichen Web Servers

- 1. Orchestration Komponente erkennt Überlastung des vorhandenen Web Servers**
 - a. Sammlung und Auswertung von Auslastungsdaten
 - b. Suchen eines freien Servers im Ressourcenpool für Web Server
- 2. Deployment Komponente installiert und konfiguriert neuen Server**
 - a. Installation des Software Stacks (Betriebssystem und Anwendungen)
 - b. TCP/IP Konfiguration auf dem neuen Server
- 3. Deployment Komponente konfiguriert Netzwerkkomponenten**
 - a. Konfiguration des Switch Ports, an dem der neue Server hängt
 - b. Konfiguration des Load Balancers (virtuelle IP der Anwendung)

Beispiel: Workflow zur Server Provisionierung

Workflow:  Default Add Server to Cluster	Logical Operation:  Cluster.Add Server
Description: Default Add Server to Cluster	
▷  this (ClusterID, ServerID):	Transitions
<p>▷ Get Current Deployment Request Id () : RequestId</p> <p>▷ Get Cluster Attributes (ClusterID) : ?, ?, ?, ?, ?, LoadBalancerId, ?, ?, ?, PoolId, ?, ClusterVLANId, ClusterVIP, ClusterVirtualAddressId, ?</p> <p>▷ Get VLAN Attributes (ClusterVLANId) : ?, ?, ClusterSubnetworkId, ?</p> <p>▷ RM Allocate Server (ClusterID, PoolId, RequestId) : ServerID</p> <p>▷ Get Server Attributes (ServerID) : ?, ?, ?, ?, ?, ServerIP, NicId, ?, ?, NetworkInterfaceId</p> <p>▷ Get Software Stack for Device (ClusterID) : ClusterSoftwareStackId</p> <p>▷ SoftwareStack.Install (ServerID, ClusterSoftwareStackId) :</p> <p>▷ RM Allocate IP Address (NicId, ClusterSubnetworkId) : NewServerIP</p> <p>▷ Get Network Interface Attributes (NetworkInterfaceId) : ?, ?, ?, ?, NetworkInterfaceName, ?, ?</p> <p>▷ IPSystem.Add IP Address (NULL, ClusterSubnetworkId, ServerID, NewServerIP, 'true', NicId, NULL) : NetworkInterfaceName</p> <p>▷ IPSystem.Apply Routing Table (ServerID, NetworkInterfaceId, NULL, true, ClusterID) :</p> <p>▷ Get NIC Attributes (NicId) : ?, ?, ?, ?, SwitchPortModule, SwitchPort, SwitchId, ?, PoolVLANId</p> <p>▷ Switch.Move Port to VLAN (ClusterVLANId, SwitchPortModule, SwitchPort, PoolVLANId, SwitchId) :</p> <p>▷ SNMP Ping (NewServerIP, 'public', '30000') : ?</p> <p>▷ RM Add Server (ClusterID, RequestId, ServerID) :</p>	

Dr. Stefan Fütterling – The Open Group Distinguished Architect – stefan.fuetterling@capgemini.com
Martin Bachmaier – IT Versatilist – mbachmaier@lenovo.com
Daniel Amor - The Open Group Master Architect - danny@dxc.com



Duale Hochschule Baden-Württemberg Stuttgart
IT Architekturen, 2025-10

1.3 Cloud Computing

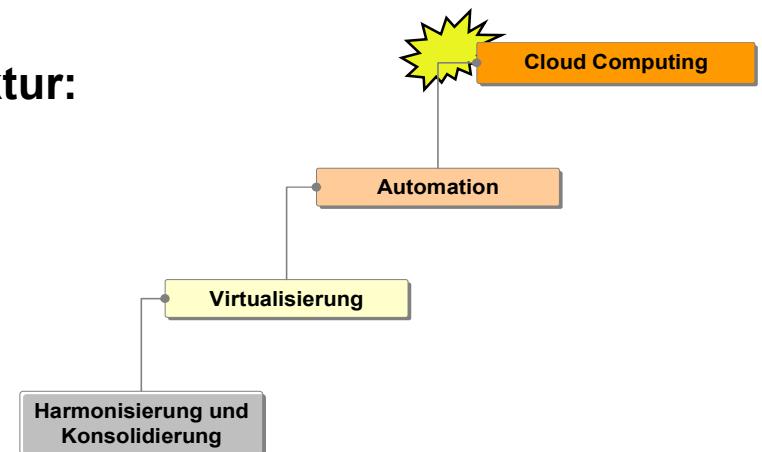
Cloud Computing

■ Geschäftsmodell einer Cloud:

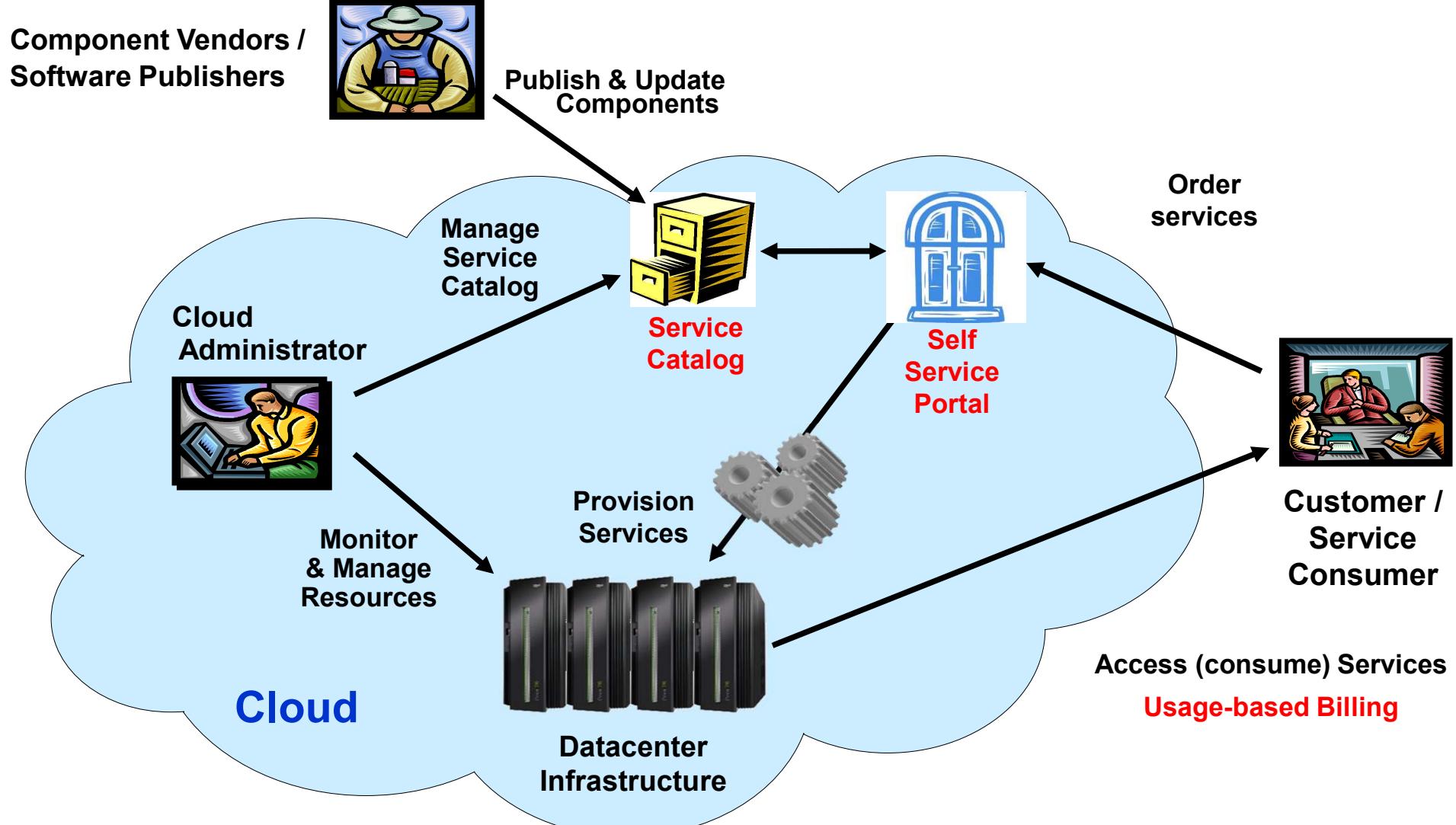
1. Bereitstellung von **standardisierten IT Services (Servicekatalog)**
2. Bestellung des Services durch den Kunden (Consumer) über ein **Self Service Portal**
3. **Verbrauchsbasierte Abrechnung** der genutzten Ressourcen, z.B.
 - Server nach Größe und Nutzungsdauer (Stunden)
 - Storage nach belegtem Speichervolumen (GB)
 - Netzwerk nach übertragenem Datenvolumen (GB)

■ Anforderungen an die zu Grunde liegende IT Infrastruktur:

- Breitbandiger Netzwerkzugang zur Cloud
- Pools von meist virtualisierten IT Ressourcen
- Automation der Bereitstellungsabläufe
- Capacity Management, Skalierbarkeit (Elastizität)
- Sicherheit



Geschäftsmodell einer Cloud



Arten von Clouds (Deployment Models)

▪ Nutzer:

- Die Cloud Nutzung ist offen für alle Kunden/Consumer (**Public Cloud**)
 - Public Cloud Anbieter bieten ihren Kunden oft einen logisch abgegrenzten Bereich innerhalb einer Public Cloud an ("Virtual Private Cloud VPC", „VNet“)
- Die Nutzung der Cloud erfolgt exklusiv durch einen Kunden (**Private Cloud**)
- Nutzung der Cloud durch eine definierte Gruppe von Consumern (**Community Cloud**)

▪ Standort:

- Die Cloud steht im Datacenter des Cloud Anbieters (**Off Premises Cloud**)
- Die Cloud steht im Datacenter des Kunden / des Unternehmens (**On Premises Cloud**)
 - Aufgebaut und betrieben von der eigenen IT-Organisation des Unternehmens
 - Aufgebaut und betrieben von einem Cloud Anbieter
- Eine Anwendung nutzt mehrere Clouds in verschiedenen Datacentern (**Hybrid Cloud**)
 - Oft eine Kombination aus On-Premises und Off-Premises Clouds
 - Idealerweise mit der Möglichkeit, Workloads zwischen den beiden Clouds zu verschieben

▪ Beispiele für Public Cloud Anbieter:

- Amazon Web Services (AWS)
- Microsoft Azure
- Google Cloud Platform (GCP)
- IBM Cloud
- Salesforce

Vgl.: National Institute for Standardisation and Technology (NIST)
<http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf>

Cloud Servicemodelle

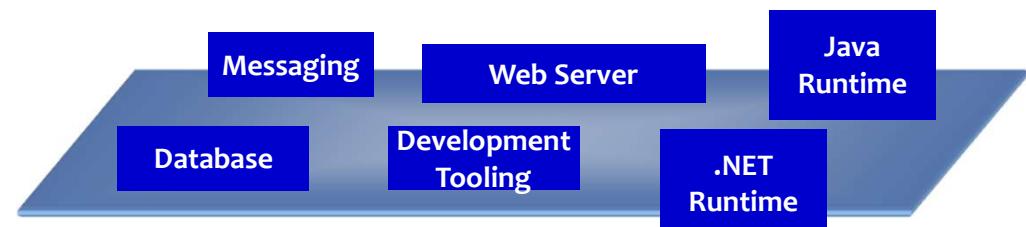
Software as a Service (SaaS):

- Cloud Anbieter stellt **Anwendung** bereit
- Cloud Consumer hat keinen Zugriff auf die darunter liegende Infrastruktur und die Laufzeitumgebungen
- Bsp.: Microsoft Azure Office 365



Platform as a Service (PaaS):

- Cloud Anbieter stellt **Anwendungsplattform incl. Laufzeitumgebung** bereit
- Cloud Consumer deployt seine eigene (meist selbst entwickelte) Anwendung
- Bsp.: Microsoft Azure App Services

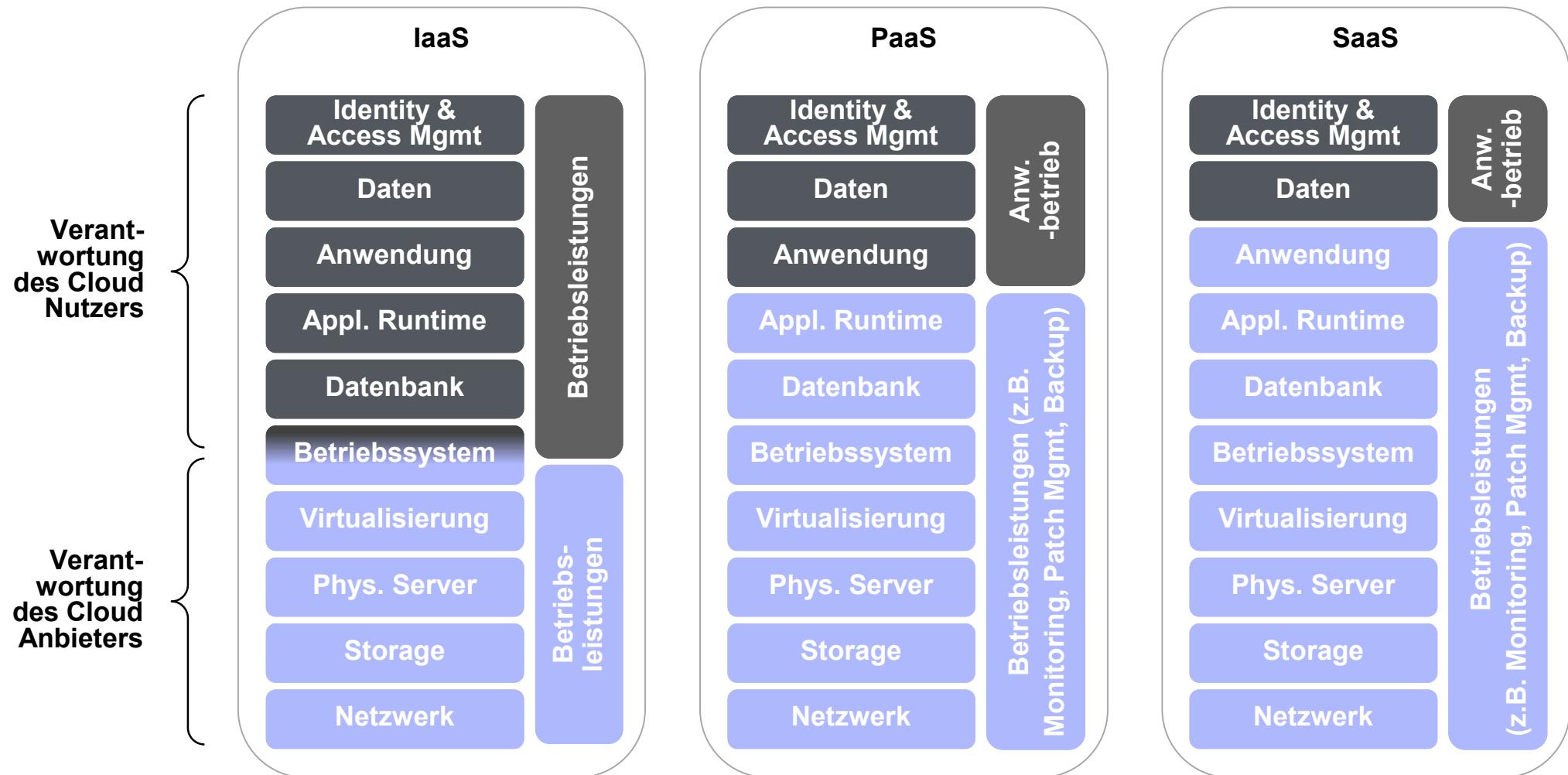


Infrastructure as a Service (IaaS):

- Cloud Anbieter stellt **meist virtuelle Maschinen incl. Betriebssystem** bereit
- Cloud Consumer installiert seine eigene Laufzeitumgebung und Anwendung
- Bsp.: Microsoft Azure Virtual Machines



Verantwortlichkeiten bei IaaS, PaaS und SaaS Cloud Services



Migration von Anwendungen in die Cloud

Unternehmens-
Datacenter

Retire: Die Anwendung wird nicht mehr benötigt und wird abgeschaltet.

X

Retain: Die Anwendung verbleibt erst einmal im Unternehmens-Datacenter und wird zu einem späteren Zeitpunkt wieder betrachtet.

X

Rehost: Die gesamte VM mit der Anwendung wird in das Cloud Datacenter verlagert.

Public Cloud

Replatform: Die Anwendung wird auf einer neuen Runtime Umgebung deployt, z.B.:
Wildfly statt Websphere
Amazon RDS Service statt Oracle Datenbank

Refactor: Die Anwendung wird umgebaut, z.B.:
Paketierung in Container
Aufteilung eines Monolithen in Microservices

Repurchase: Die Anwendung wird anders lizenziert, z.B.:
SaaS statt eigener Lizenz

Automation in der Cloud: Infrastructure as Code (IaC)

Beispielaufgabe:

Anlegen einer kompletten Umgebung in Azure mit:

- 2 VMs
- 1 DB
- Storage zuordnen
- Loadbalancer anlegen
- Firewallregeln setzen

Und das Ganze gleich 3x als DEV, INT und PROD Umgebung

Siehe z.B. Quickstart Templates auf:
<https://azure.microsoft.com/de-de/resources/templates/>

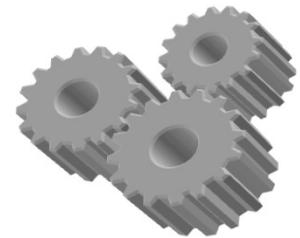


Manuelles Anlegen im Azure Portal:

- Lange Bereitstellungszeit
- Fehleranfällig
- Gute Skills nötig

ARM Template:
Beschreibung der
Umgebung im
JSON Format
**= Infrastructure
as Code**

Parameter für DEV,
INT und PROD:
• DNS Namen
• Adminuser
• ...

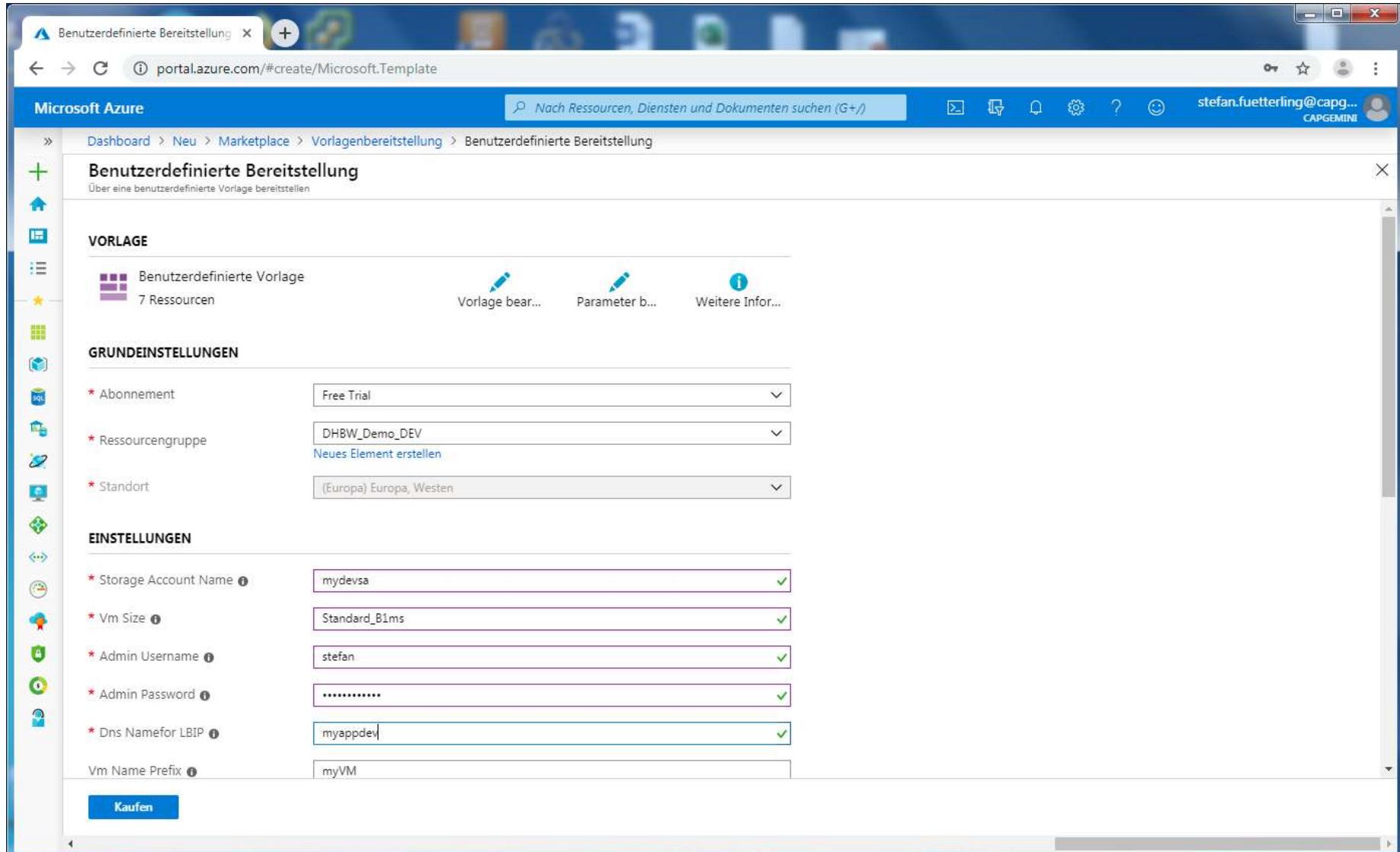


Ausführung im
Azure Resource
Manager (ARM)

Automatisiertes Anlegen über Azure Resource Manager (ARM):

- Kurze Bereitstellungszeit
- Wenig fehleranfällig
- Wiederholbar für jede neue Umgebung

Infrastructure as Code (IaC) Übung



The screenshot shows the Microsoft Azure portal interface for creating a custom deployment template. The left sidebar lists various service icons. The main content area has a blue header bar with the URL portal.azure.com/#create/Microsoft.Template. The page title is "Benutzerdefinierte Bereitstellung".
VORLAGE
- Benutzerdefinierte Vorlage (7 Ressourcen)
- Vorlage bear...
- Parameter b...
- Weitere Infor...
GRUNDEINSTELLUNGEN
- * Abonnement: Free Trial
- * Ressourcengruppe: DHBW_Demo_DEV
- Neues Element erstellen
- * Standort: (Europa) Europa, Westen
EINSTELLUNGEN
- * Storage Account Name: mydevsa
- * Vm Size: Standard_B1ms
- * Admin Username: stefan
- * Admin Password:
- * Dns Namefor LBIP: myappdev
- Vm Name Prefix: myVM
Kaufen

Infrastructure as Code (IaC) Übung

1. Microsoft Azure Account

Azure for Students Account einrichten

<https://azure.microsoft.com/de-de/free/free-account-students-faq/>

Mit dem „xxx@lehre.dhbw-stuttgart.de“ Account und Passwort anmelden

2. Legen Sie manuell einen neuen virtuellen Server (VM) an mit:

Storage

Netzwerk

Darauf soll Tomcat laufen

3. Legen Sie eine ganze Umgebung über eine Vorlage (Template) an:

Ressourcengruppe anlegen

+ Ressource erstellen => Vorlagenbereitstellung

Eigene Vorlage => Datei laden => ITA_15_azuredeploy.json

