



# **Migration von On- Premise auf Multi und Hybrid Cloud**

## **THM Seminar 2**

## Gliederung

- Literatur
- Einführung in die Aktualität von Cloud Computing
- Problemstellung und Motivation
- Grundbegriffe
- Allgemeine Cloud Migration
- Migration von Individualsoftware
  - Architektur
  - Cloud Native
  - Serverless
  - DevSecOps
- Fazit

# Über mich



## Beruf

- Seit 2024 Full Stack Engineer bei romeisIE
- Seit 2019 Selbstständig im Bereich Softwareentwicklung und Code Trainer
- 2022-2024 IT-Consultant bei Sopra Steria im Bereich Public sowie Web Technologies
- 2018 - 2020 Ideas Embedded im Bereich .Net Entwicklung sowie Entwicklung in C im Bereich Automotiv

## Persönlich

- Aktiv in Software Communities
  - Main Kinzig Nerds
  - Go User Group Frankfurt
  - Kubernetes User Group Frankfurt
- Aktiv bei der Feuerwehr



kubernetes

Email (THM): [cong.chanh.vinzenz.nguyen@mnd.thm.de](mailto:cong.chanh.vinzenz.nguyen@mnd.thm.de)

Email: [congchanh@outlook.de](mailto:congchanh@outlook.de)

LinkedIn: <https://www.linkedin.com/in/cong-chanh-vinzenz-nguyen-28b50219a>

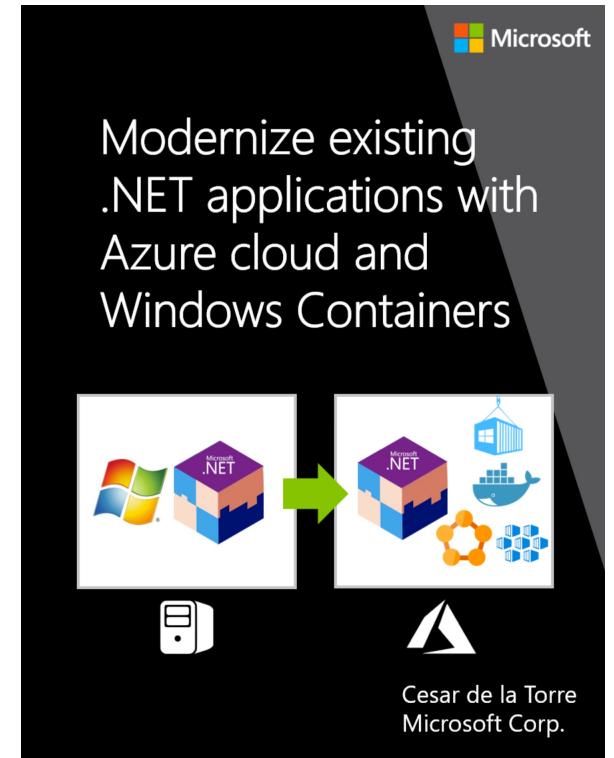


# Literatur

- Modernizing existing .NET application with Azure cloud and windows containers

## Links:

- Cloud Native Computing Foundation
- Microsoft Dokumentation



# Cloud Computing

- Was ist „Cloud“
- Wie wird das Konzept Cloud umgesetzt?
- Was sind Vor und Nachteile von Cloud?
- Wie kann Cloud sinnvoll und effizient genutzt werden?
- Wie wird Cloud sinnvoll in eine bestehende Unternehmensarchitektur integriert?
- Wie können bestehende Systeme in die Cloud migriert werden?



# Warum ist das Thema so wichtig?

- Deutschland ist das Land des Klein und Mittelstands
- Hoher Bedarf an Digitalisierung



**IT-ZENTRUM BUND**

NEWSROOM PRESSE LEICHTE SPRACHE DEUTSCHE GEBÄRDENSPRACHE

LEISTUNGSPORTFOLIO IT-LÖSUNGEN DIGITALE MISSION DAS ITZBUND KARRIERE

**Die Bundescloud – eine exklusive, private Cloud für die Bundesverwaltung**

Startseite → IT-Lösungen → E-Government → Die Bundescloud – eine exklusive, private Cloud für die Bundesverwaltung

**Der Staat wird digital**

Flexibel, effizient und sicher: Bundesbehörden und Ministerien arbeiten zunehmend über alle Ebenen und Bereiche hinweg zusammen. Cloudbasierte IT-Anwendungen über die Bundescloud sind die ideale Lösung dafür.

[1] <https://www.forbes.com/sites/bernardmarr/2023/10/09/the-10-biggest-cloud-computing-trends-in-2024-everyone-must-be-ready-for-now/>

[2] <https://t3n.de/news/bundescloud-cybersecurity-millionen-1555164/>

[3] <https://www.cloudcomputing-insider.de/branchenspezifische-clouds-effizienz-und-compliance-a-dd5814cb3148833af17c283ea9410eec/>

Spezialisiertes Cloud Computing

Branchenspezifische Clouds läuten  
neue Ära ein

22.01.2024 · Von Anna Kobylinska und Filipe Pereira Martins\* · 8 min Lesedauer ·

Während die digitale Transformation verschiedene Branchen durchdringt, nimmt die Popularität branchenspezifischer Clouds in vielen wettbewerbsintensiven Sektoren zu. Durch die Anpassung von Cloud-Lösungen an die spezifischen Bedürfnisse der jeweiligen Branche bringen diese Clouds neue Maßstäbe in Sachen Effizienz und Compliance.



Spezial- oder Branchenclouds sind genau für eine bestimmte Branche entwickelt und können die Geschäftsprozesse der Unternehmen in diesem Sektor effizienter unterstützen als allgemeine Clouds.  
(Bild: dusanpetkovic1 - stock.adobe.com)

**Forbes**

FORBES > INNOVATION > ENTERPRISE TECH

**The 10 Biggest Cloud Computing Trends In 2024 Everyone Must Be Ready For Now**

Opinions expressed by Forbes Contributors are their own.

Bernard Marr Contributor

Follow

Oct 9, 2023, 02:33am EDT



The 10 Biggest Cloud Computing Trends In 2024 Everyone Must Be Ready For Now  
ADOBESTOCK

# Historie

- Windows Desktop PC
- Notebooks

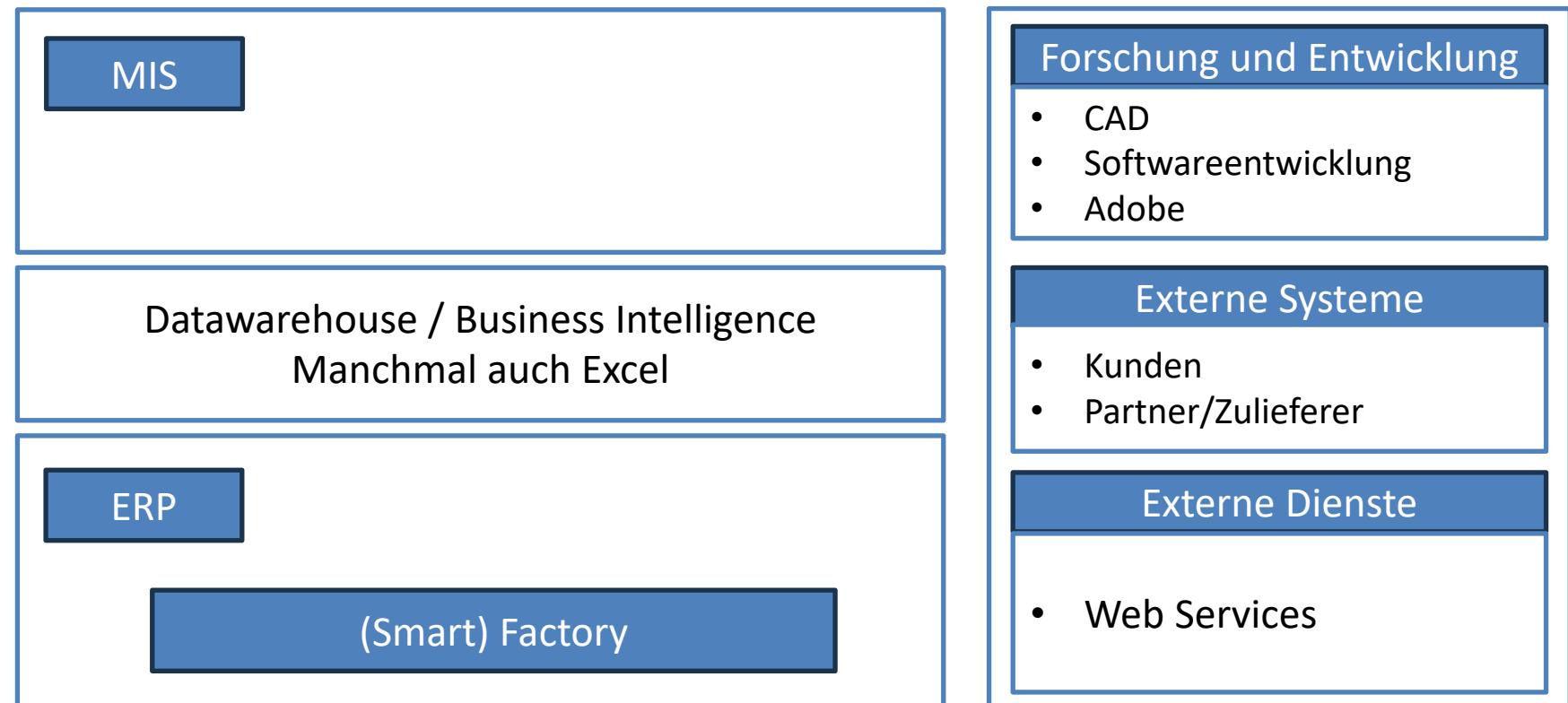


- CPU: Parallelisierung statt schneller
- Computer/Notebooks
  - Mac OS
  - ChromeBook
  - Windows 10/11
- Tablet
  - Windows
  - IPad OS
  - Android
  - ChromeBook
  - HarmonieOS
- Smartphones
  - IOS
  - Android
  - HarmonieOS
- Weiter Geräte
  - Smartwatch
    - Android
    - Apple Watch
  - Autos/LKW/Traktoren mit Betriebssystem
  - Smart Devices
    - Waschmaschinen
    - Türen/Fenster
    - Airtags
  - VR und AR
  - Smart Factory
  - Anzeigetafeln



## Ist-Zustand vieler Organisationen

- Die meisten Organisationen haben bereits vorhanden IT Systeme
- Standardsoftware
  - Office Suite
  - ERP, SCM, CRM...
- Individualsoftware
  - Gekauft
  - Eigenentwicklungen
- Hardware
  - Computer
  - Smartphones
  - Ggf. Smart Devices
  - Ggf. Smart Factory
  - Server



# Herausforderungen

Große Mengen an Daten

Abhängigkeiten von Anwendungen, Betriebssystem und Hardware

Vernetzung der einzelnen Geräte

Flexibilität und Mobilität

Verfügbarkeit von Diensten

## Wie erkennt man das Problem?

- Heterogene Plattformen durch unterschiedliche Betriebssysteme nicht immer kompatibel
- Potential von Informationen und Technologie wird nicht sinnvoll genutzt:
  - Information als Kostenfaktor
  - Neue Technologie und Potenziale werden nicht umgesetzt
  - Unzureichende Informationsversorgung des Managements
- Langfristiger Wettbewerbsnachteil
- “Digitalisierungsrückstau”



## Was sind die Ziele einer Organisation?

Kostensenkung

- Abläufe optimieren

Wachstum

- Neue Geschäftsmodelle
- Optimierte Kapazitäten und Ressourcennutzung

Transformation

- Prozesse und Organisation effizienter machen

Innovation

- Wissen und Informationen besser nutzen

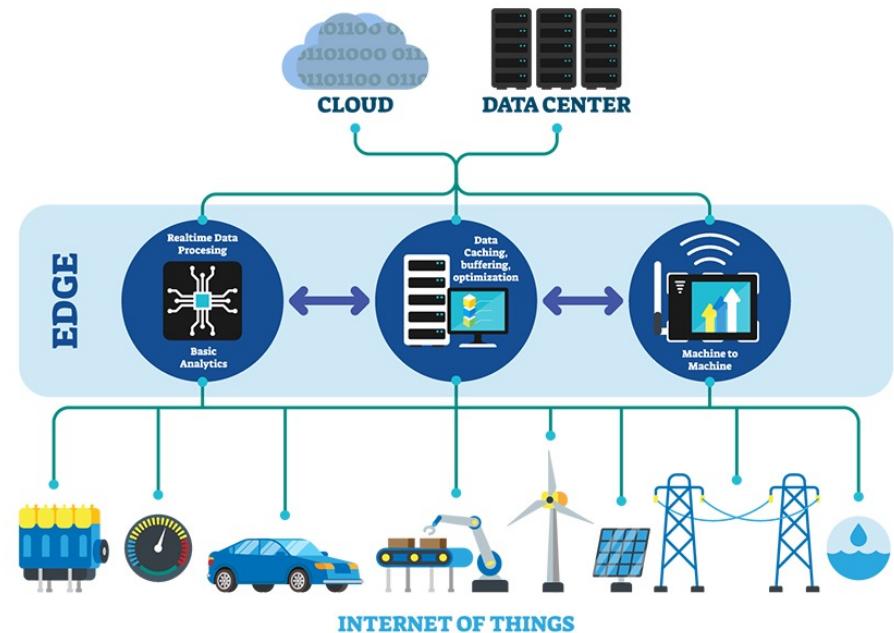
Qualitätssicherung

- Controlling
- Konstistente Ergebnisse und Abläufe

## Welche Rolle spielt Cloud Computing dabei?

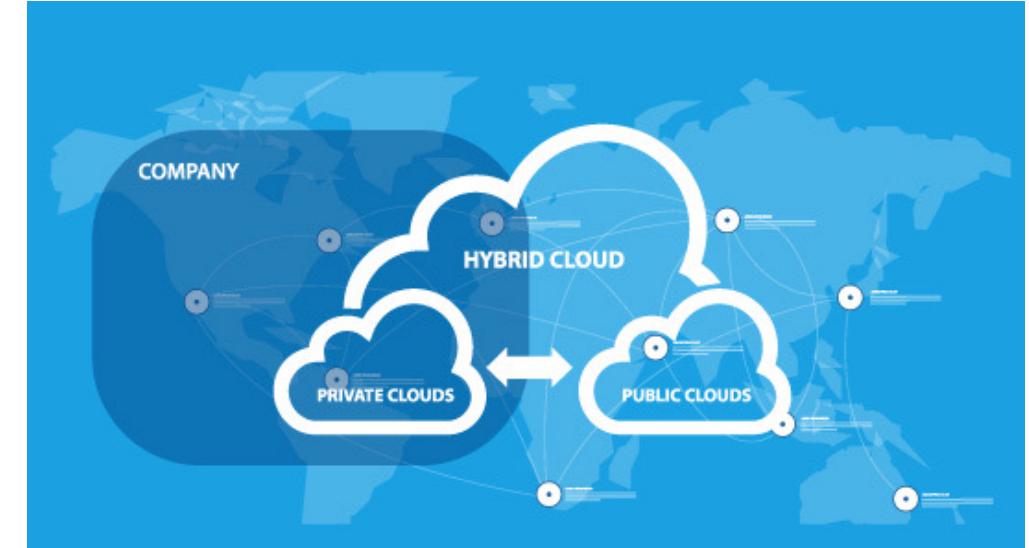
- Intelligente Vernetzung der Systeme
- Homogenisierung und Harmonisierung der Systeme
- IT als Enabler
- Nutzung neuer Technologien mit und durch die Cloud
  - Künstliche Intelligenz
  - Business Intelligence und Big Data
  - Internet of things
  - Edge Computing
  - Serverless
- Wettbewerbsvorteil
- Langfristige Kosten Senkung durch Outsourcing in die Cloud
- => Motivation für den Umstieg

### Edge Computing



## Grundbegriffe

- Definition von Cloud
- Service Modelle
- Deployment Modelle
- OnPremise
- Marktübericht



## Essential of cloud computing - Definition (NIST)



On-Demand-Self-Service



Broad network access



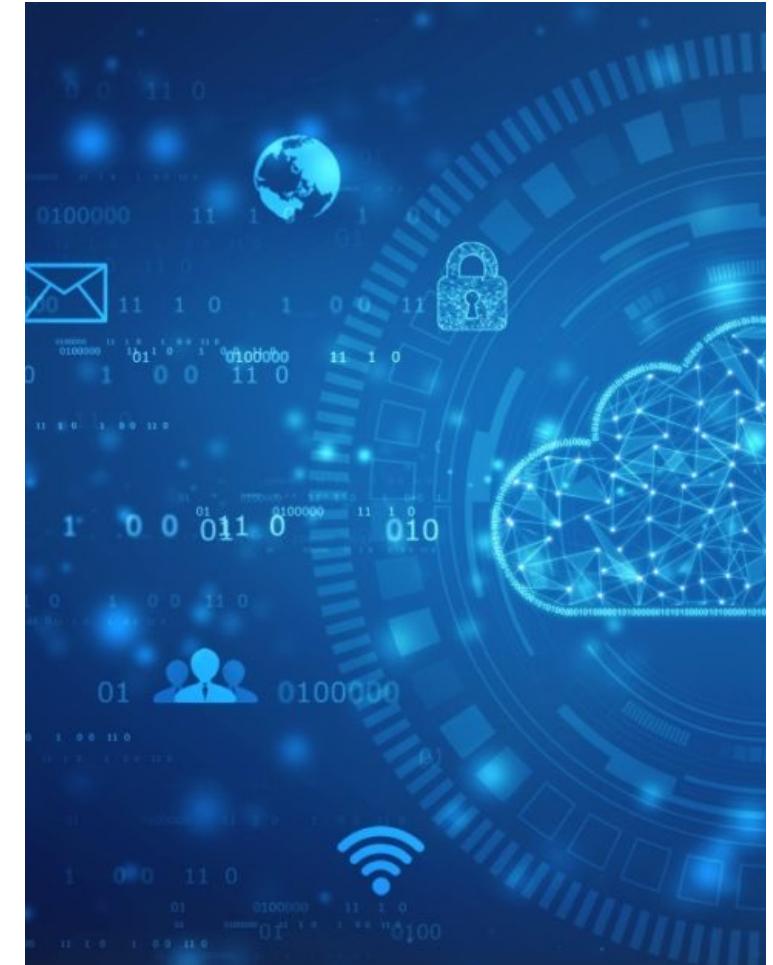
Resource pooling



Rapid elasticity



Measured service



## Service Modelle



**Software as a  
Service (SaaS)**



**Platform as a  
Service (PaaS)**



**Infrastructure as  
a Service (IaaS)**

# Deployment Modelle

Private  
Cloud

Community  
Cloud

Public  
Cloud

Hybrid  
Cloud

## OnPremise



Betreiben von eigener IT Infrastruktur

Eigene Server

geringe Elastizität

Betrieb von Anwendungen auf eigener Hardware

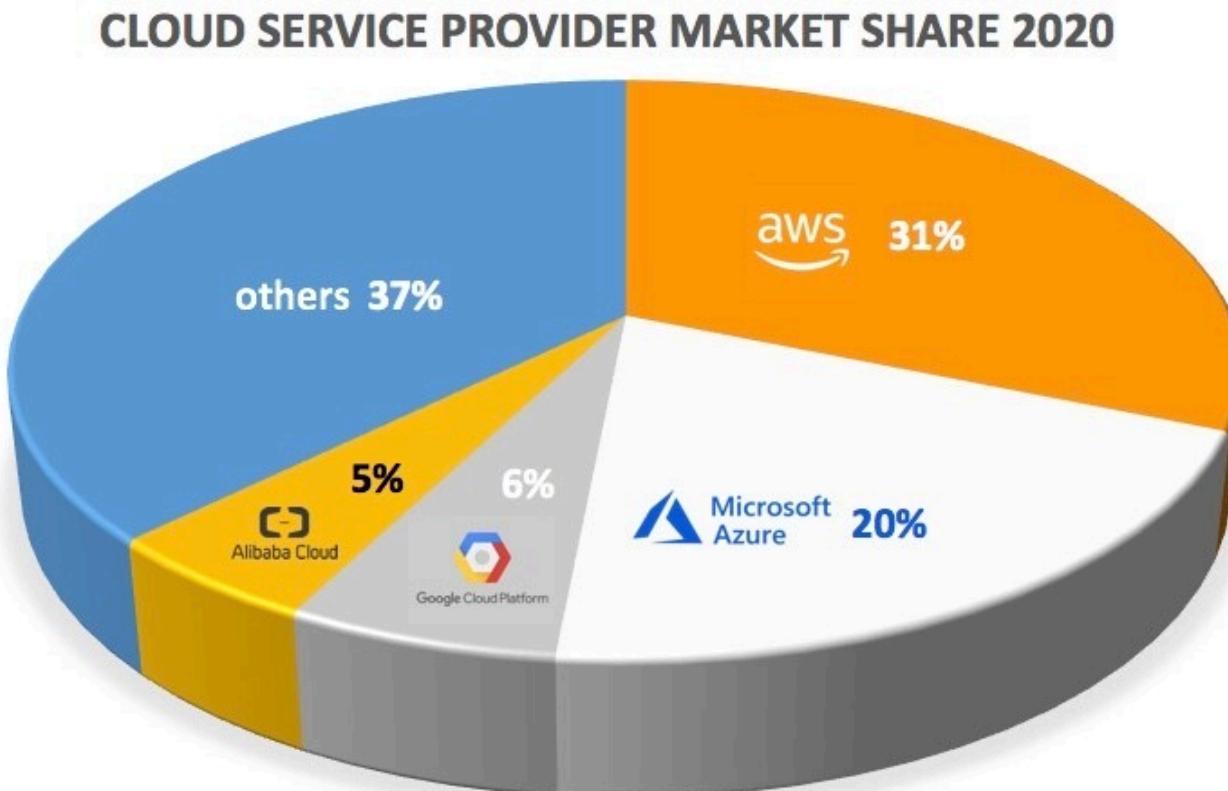
IT-Abteilung benötigt für Administration

Begrenzte Rechenleistung

Vollständige Kontrolle über Daten und Netzwerk

# Anbieter für Cloud Dienste

- Große Anbieter:
  - Amazon AWS
  - Microsoft Azure
  - IBM Cloud
  - Google Cloud
  - Alibaba Cloud
  
- Weitere Anbieter:
  - OVH Cloud
  - Digital Ocean
  - Adobe Cloud



# Wie erfolgt die Transformation auf die Cloud?

# Technischer Herausforderungen



Kompatibilität bestehender Systeme in die Cloud

Datenschutz

Compliance

DevOps

Gewachsene Strukturen

Softwarearchitektur

## Bestehende Systeme

- Standardsoftware
  - Office Suite -> z.B. Google Workspace oder MS Office 365
  - ERP, SCM, CRM -> z.B. SAP4HANA
- Individualsoftware
  - Gekauft -> Einzelfall abwägung oder VM
  - Eigenentwicklungen -> **SOA/Serverless/Cloud Native**
- Hardware
  - Computer (PWA, Web Interfaces)
  - Smartphones (PWA, Web Interfaces)
  - Smart Devices (Edge Computing)
  - Smart Factory (Edge Computing)
  - Server (Data center z.B. Azure Stack)

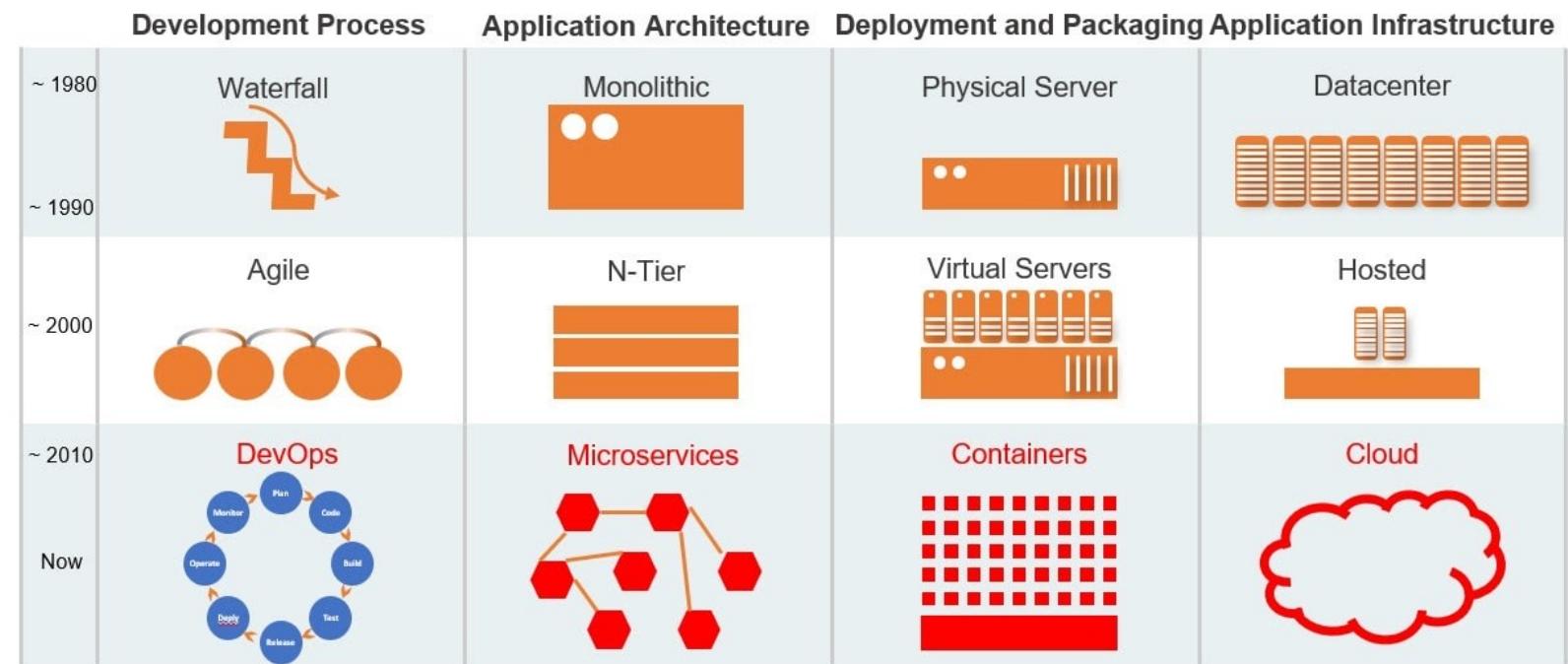
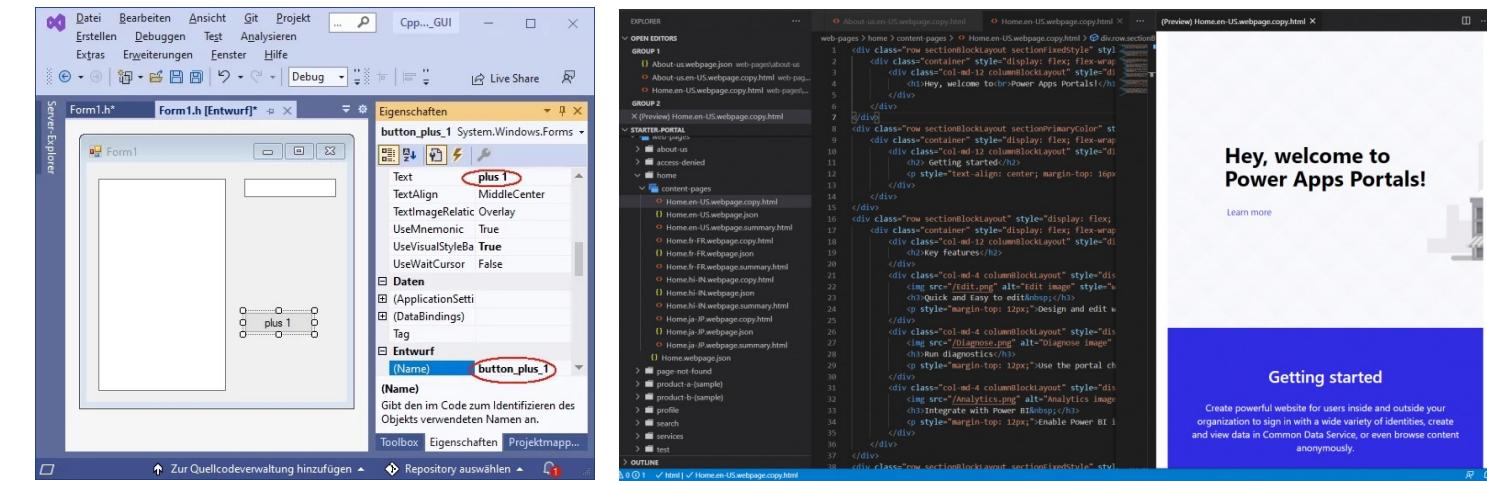
## Wann welche Cloud Variante?

- Anbieter – Spezifika bei Verwendung von Serverless – Nutzung von Multi Cloud
- Public/Private – Je nach Datenschutzrichtlinie – Nutzung von Hybrid Cloud
  
- In Betrachtziehen von politischen Aspekten und Abhängigkeiten
  - Alibaba Cloud
  - USA vs. Europäische Souveränität und Datenschutz
  - Eigenes Deployment vs. AWS Lambda oder Azure Functions



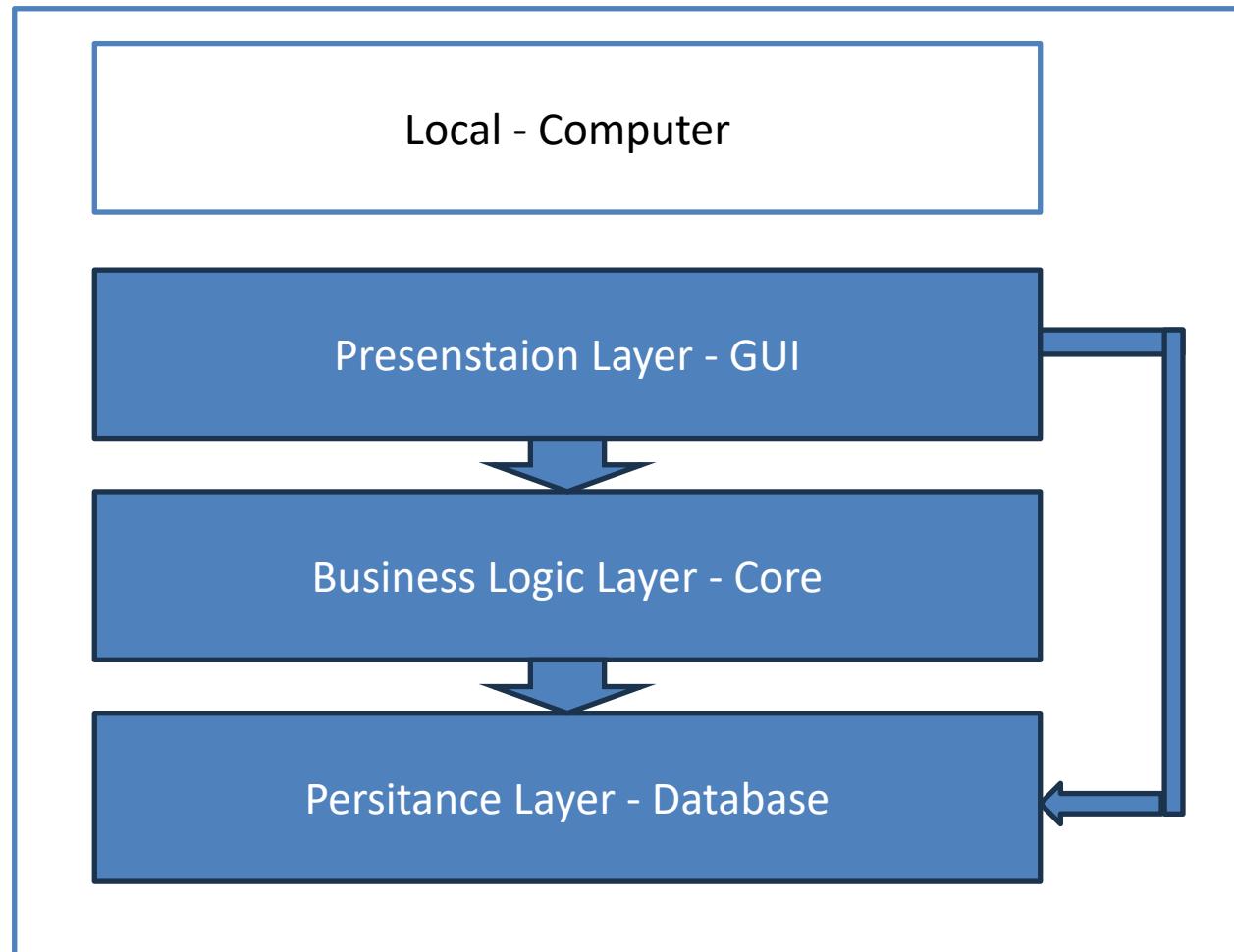
# Eigenentwicklung

- Monolith
- Server-Client
- Service Oriented Architecture (SOA)
- Microservices
- Serverless
- Rehosting (Lift and Shift)
- Replattforming
- Refactoring
- Rearchitecture



## Monolithische Anwendung

- Oft ältere Software (z.B. Windows Forms)
- Keine saubere Schichtentrennung
- Keine Parallelisierung oder verteilten Systeme
- Niedriger Abstraktionsgrad
  - Einfacherer Entwicklung und Debugging
  - Performance
  - Datenschutz und Netzwerksicherheit
- Geeignete Strategie:
  - Sofort: Rehosting für sofortige Cloudfähigkeit (IaaS)
  - Mittelfristig: Replattforming, z.B. Cloud Datenbank (PaaS)

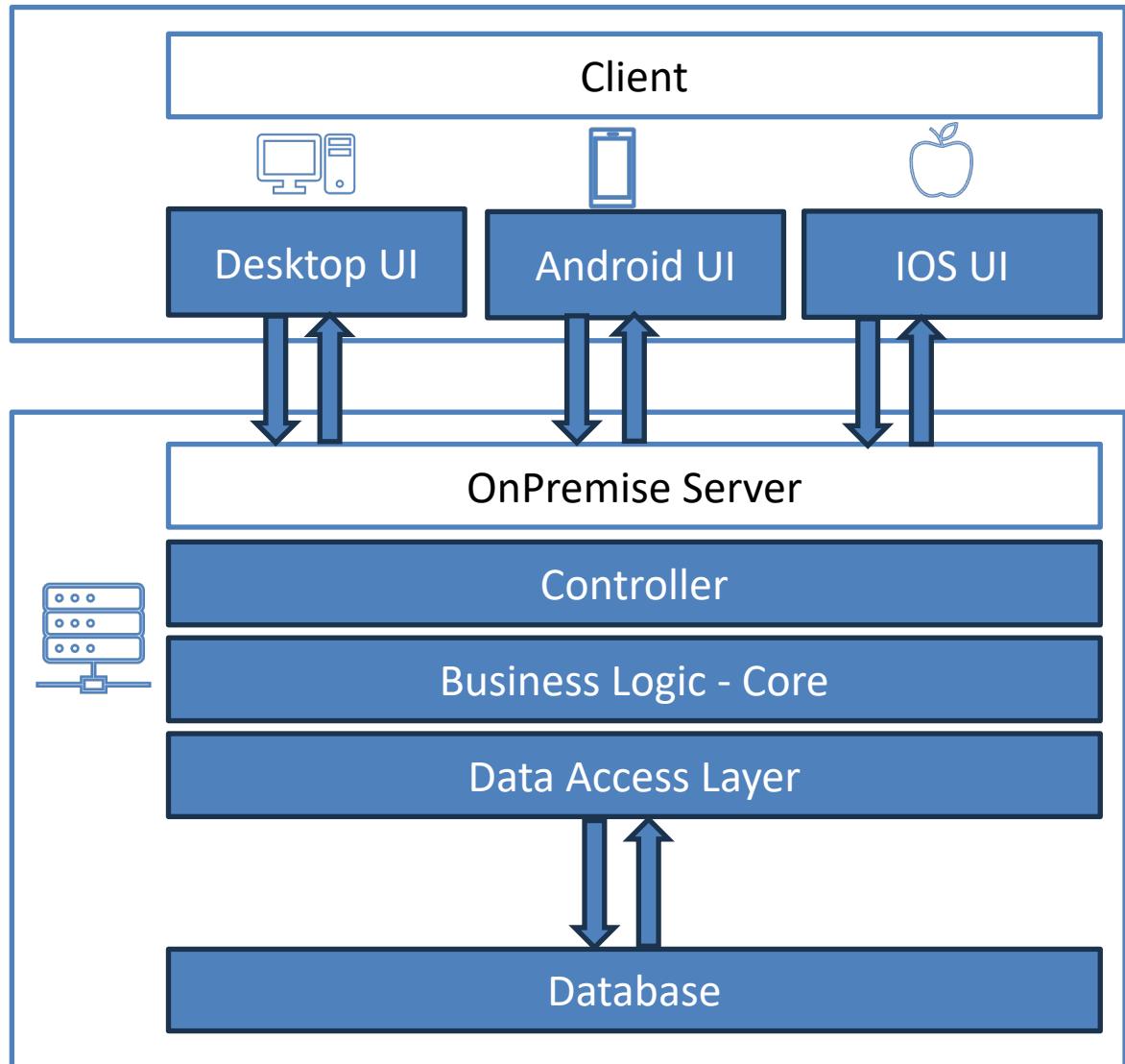


## Server-Client Architektur

- Client Anwendungen
- Zentraler Server
- Kommunikation mittels API
- Beispiel ASP.Net

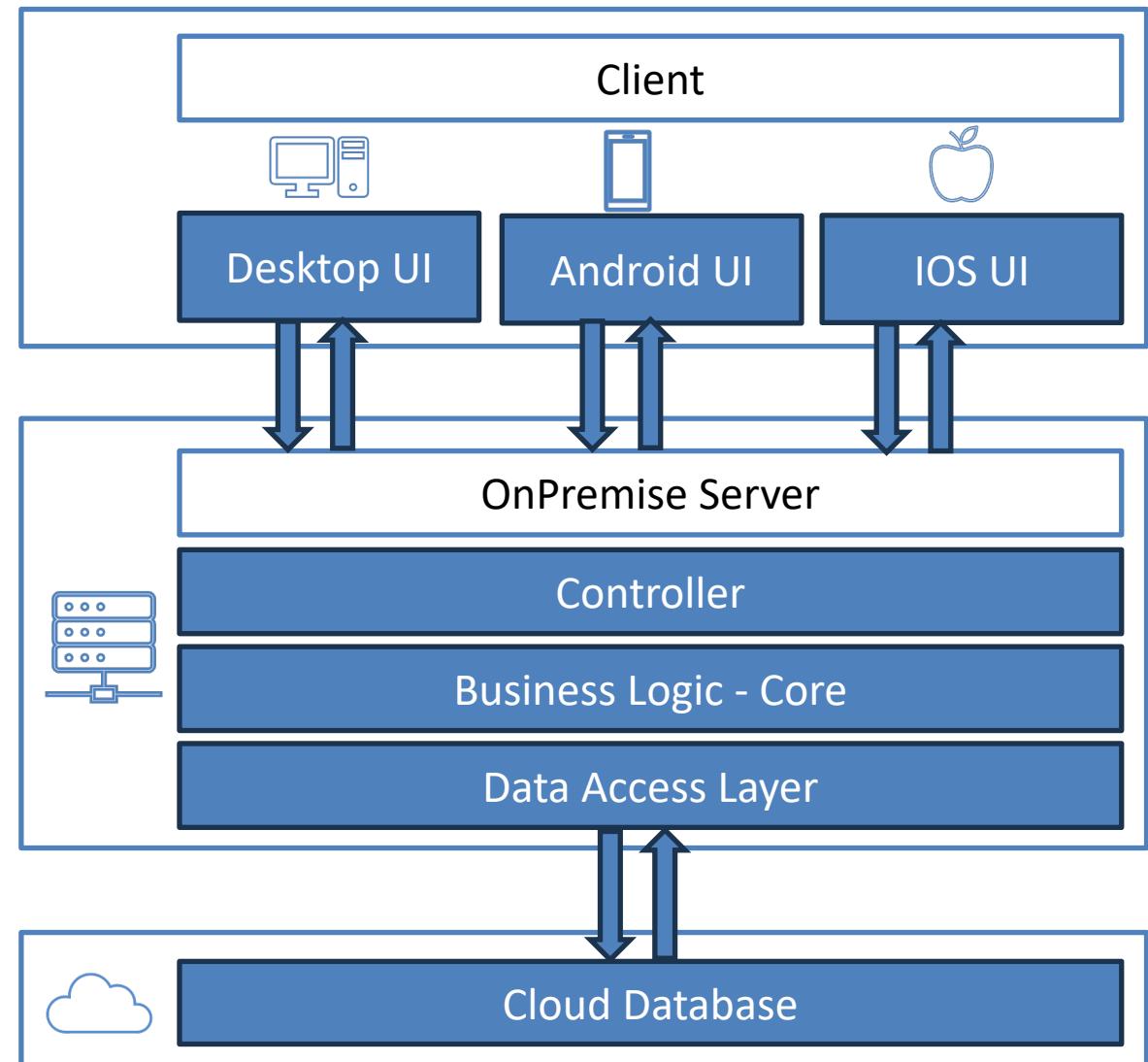
## Cloud Strategien:

- Rehosting vom Server
- Replattforming von Datenbank
- Refactoring vom einzelnen Funktionen und verwenden von Cloud Services



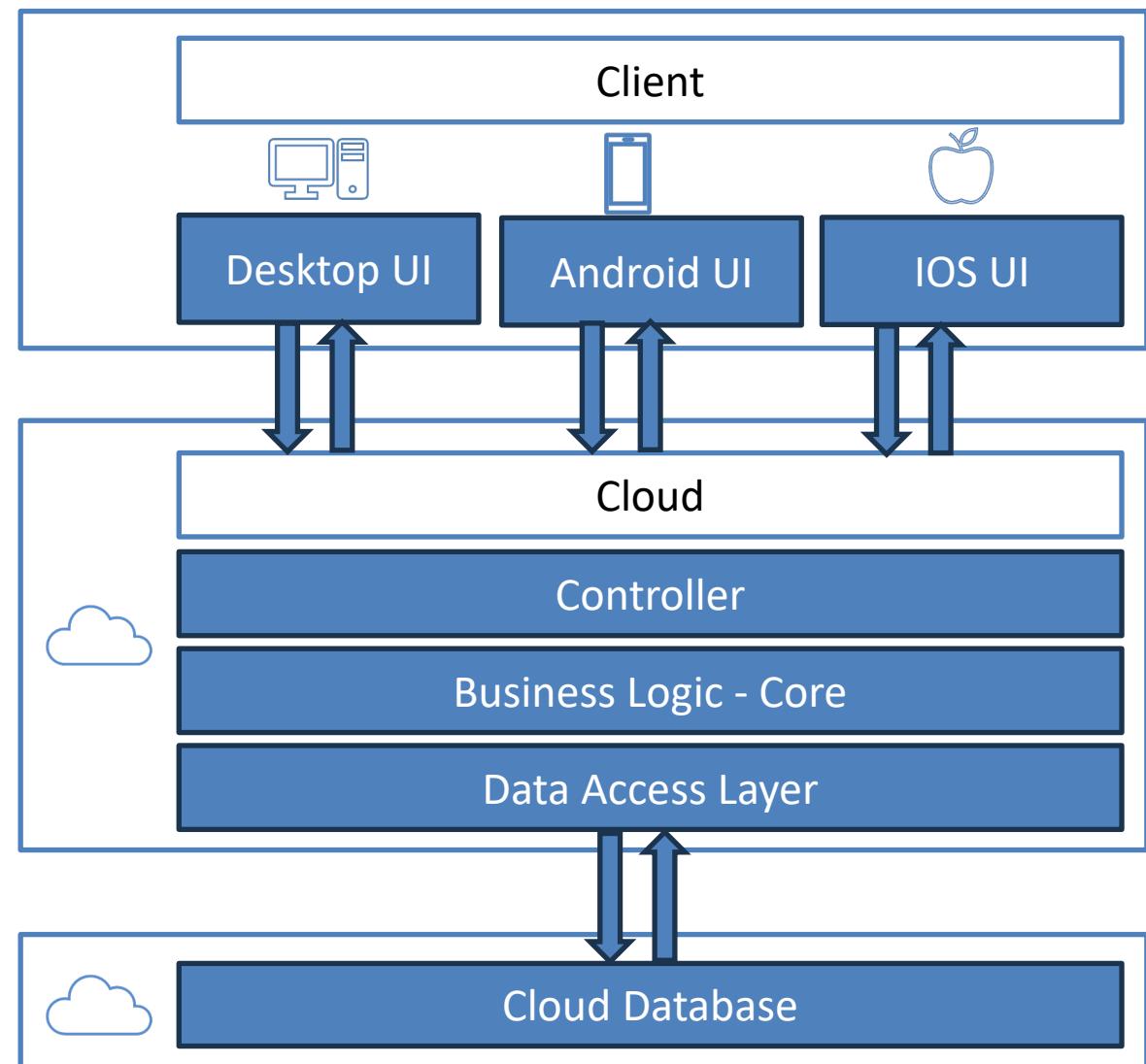
## Replattforming Database

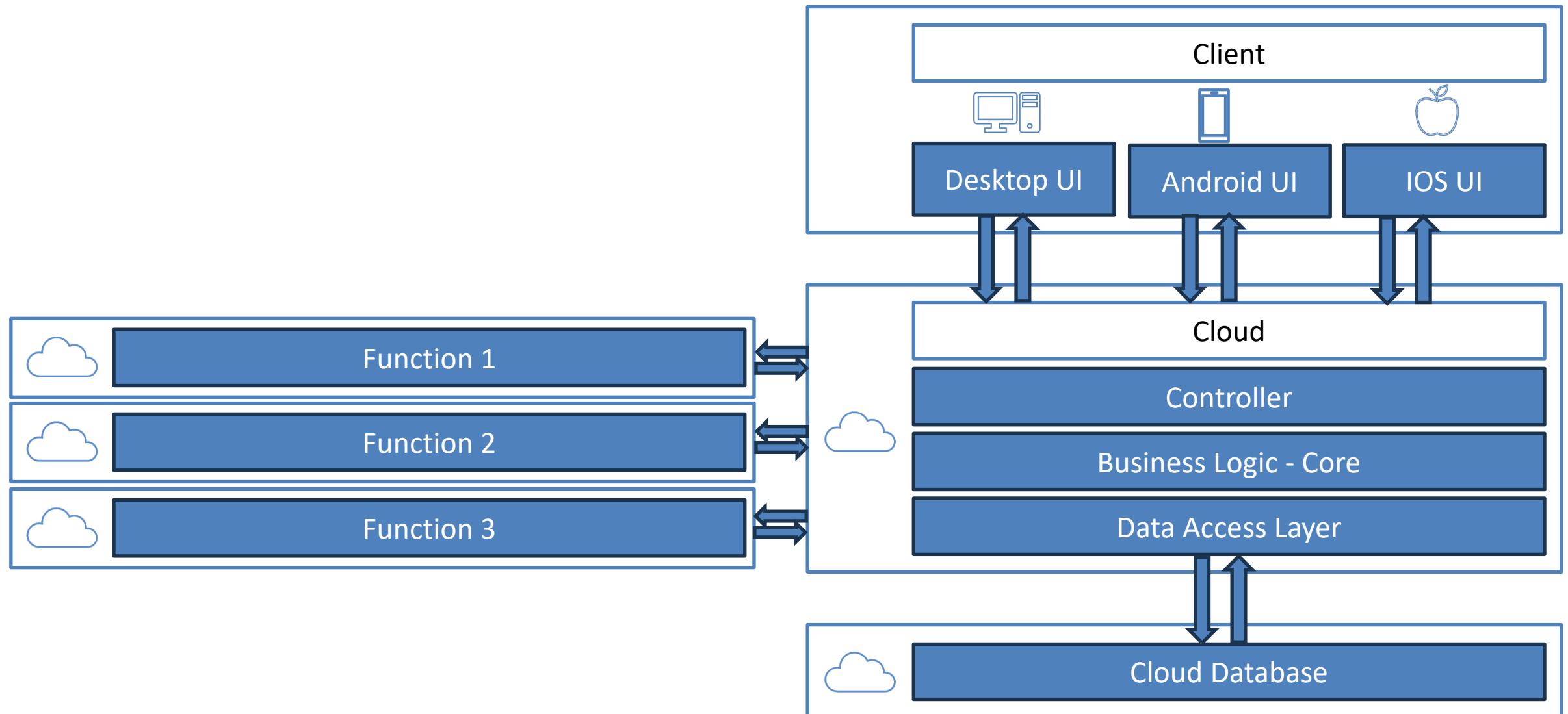
- Voraussetzung:
  - Saubere Schichtentrennung
  - Saubere Architektur und Code
  - Sauberes ORM und Migration
  - Konkretes Data Access kann einfach ersetzt werden
    - IRepository / IUnitOfWork
    - Keine direkten Zugriffe auf DB
  - Administration von DB über Server fällt weg
  - Monitoring über die Cloud



## Rehosting Server

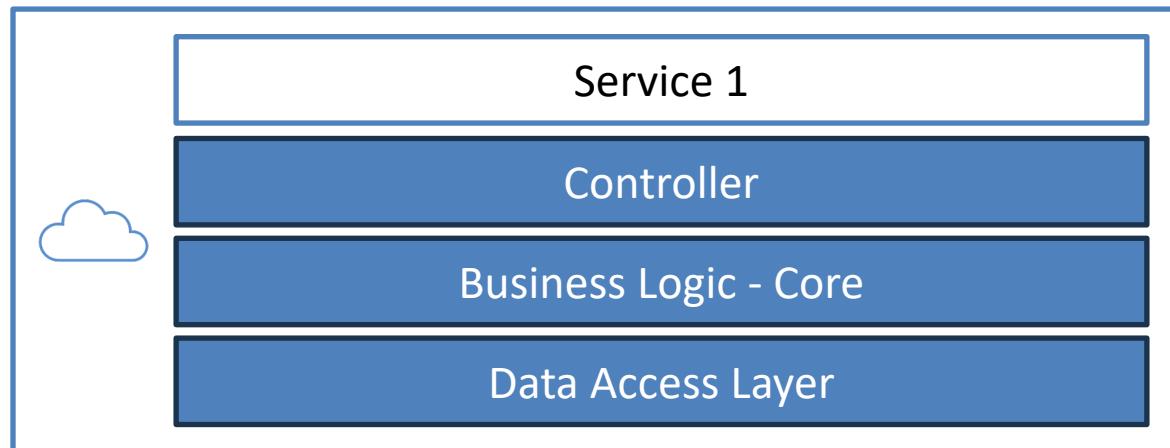
- Kein eigener Server mehr notwendig
- Eigene Administration und Netzwerktätigkeiten fallen weg
- Konfiguration auf der Cloud
- Monitoring auf der Cloud

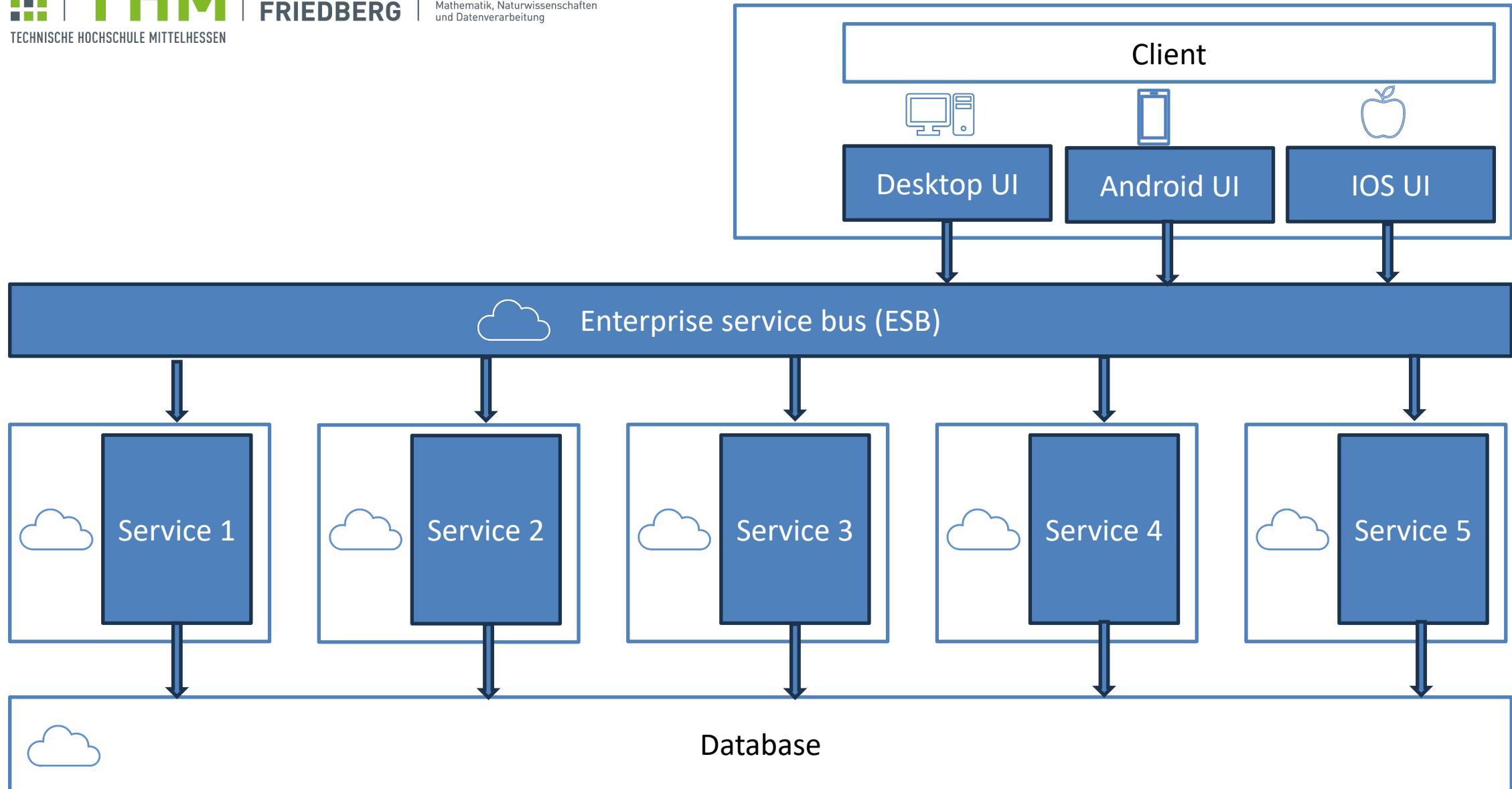




## Service oriented architecture (SOA)

- Einzelne Anwendungen als Service zu verstehen
- Ggf. Redesign nötig bei unsauberer Architektur
- Serviceorientiertes Design
- Verwendet shared resources wie z.B. Database oder gemeinsamer Code
  
- Kommunikation wird zunehmend wichtiger
- Enterprise Service Bus (ESB) z.B. BizTalk
- Message Oriented Middleware z.B. RabbitMQ
- Windows Communication Foundation (WCF)
  
- Erste Form von “Cloud”
- Microsoft nennt das Cloud optimized



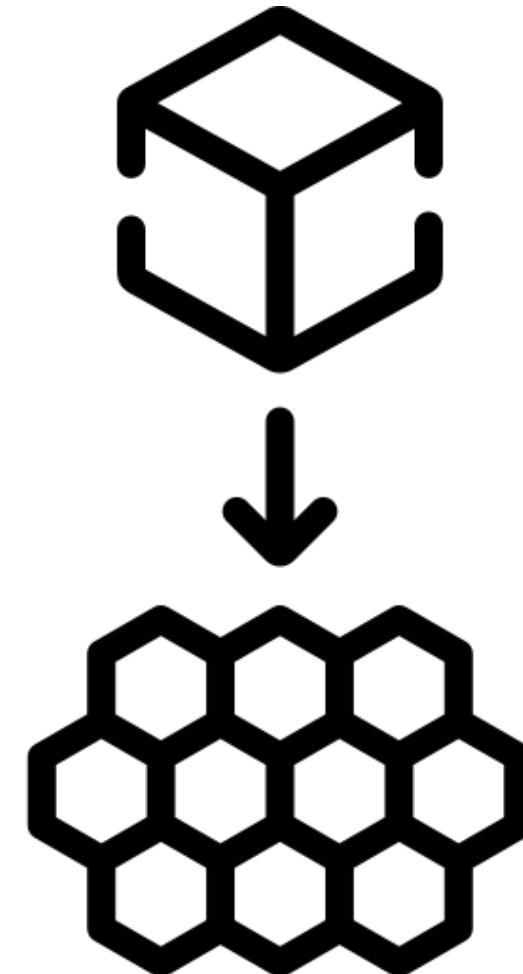


## Service oriented architecture (SOA)

- Vorteile:
  - Bedingt skalierbar
  - Keine eignene Infrastruktur nötig
  - Interoperabilität von Technologien
  - Bestehender Code kann wiederverwendet werden
  - Biete einen direkten Einstieg in die Cloud
- Nachteile
  - Harte Skalierung nicht möglich
  - Bedingte Flexibilität
  - Bedingte Fehlertoleranz
- Anmerkungen:
  - Anforderungen an die IT stark verändert

## Microservices

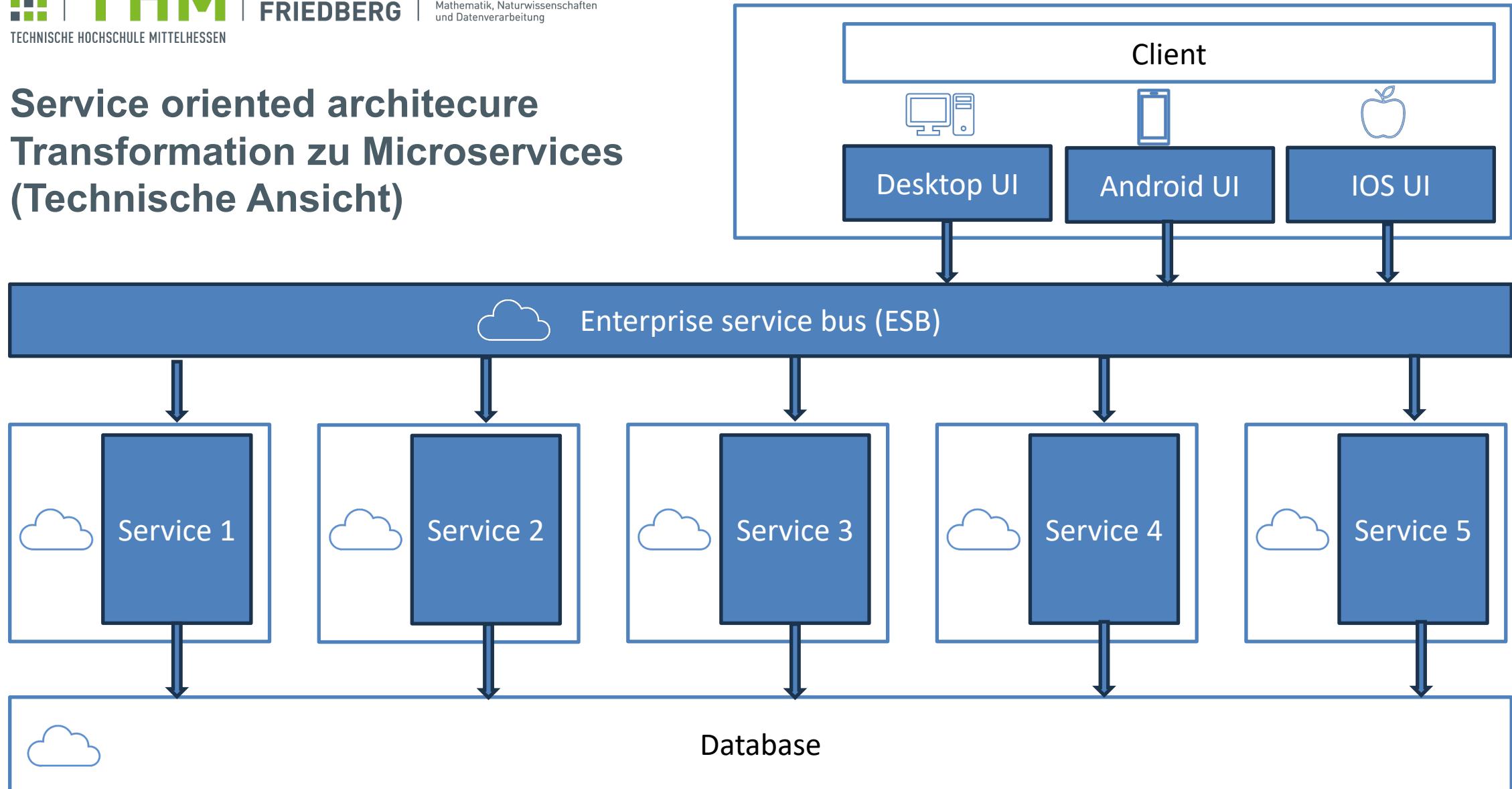
- Feingranularer als SOA
- Sofort cloudfähig und hart skalierbar
- Erfordert viel Konfigurationsarbeit
- Hohe Anforderungen an die Softwarearchitektur
  - Entkopplung
  - Unabhängigkeit
- Sehr viel Kommunikation und Netzwerkverkehr
- Mehr Flexibilität
- Fehlertolerant und selbstheilend
- Monitoring und Telemetry gewinnen zunehmend an Bedeutung
- Cloud Native
- **Service wird von genau einem Team  
gemaintained!**



# Service oriented architecture

## Transformation zu Microservices

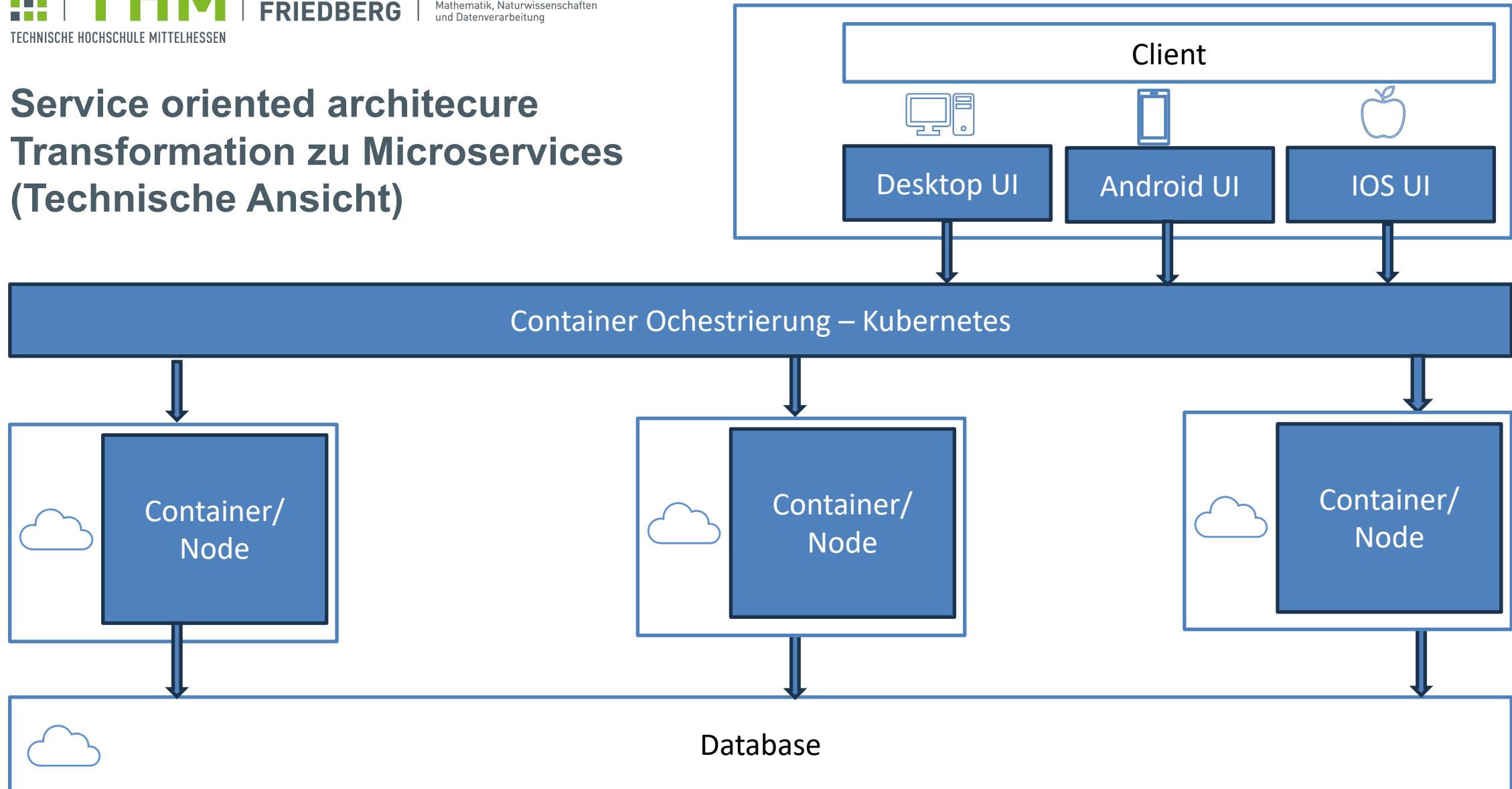
### (Technische Ansicht)



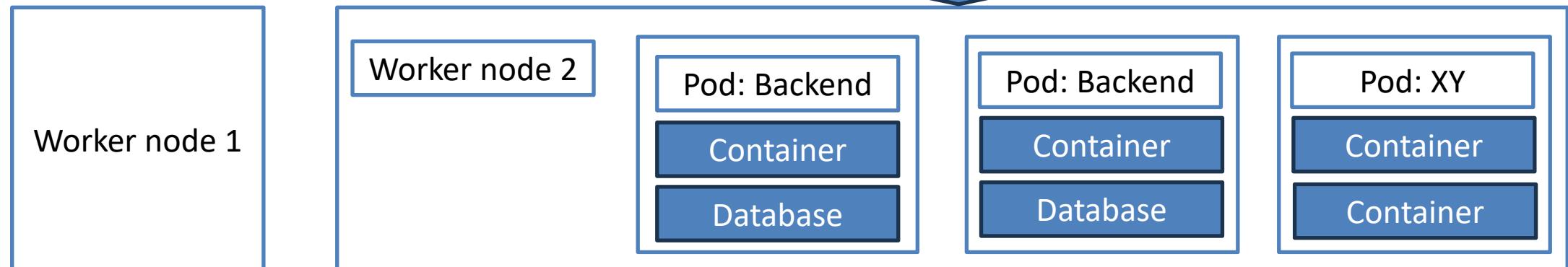
# Service oriented architecture

## Transformation zu Microservices

### (Technische Ansicht)



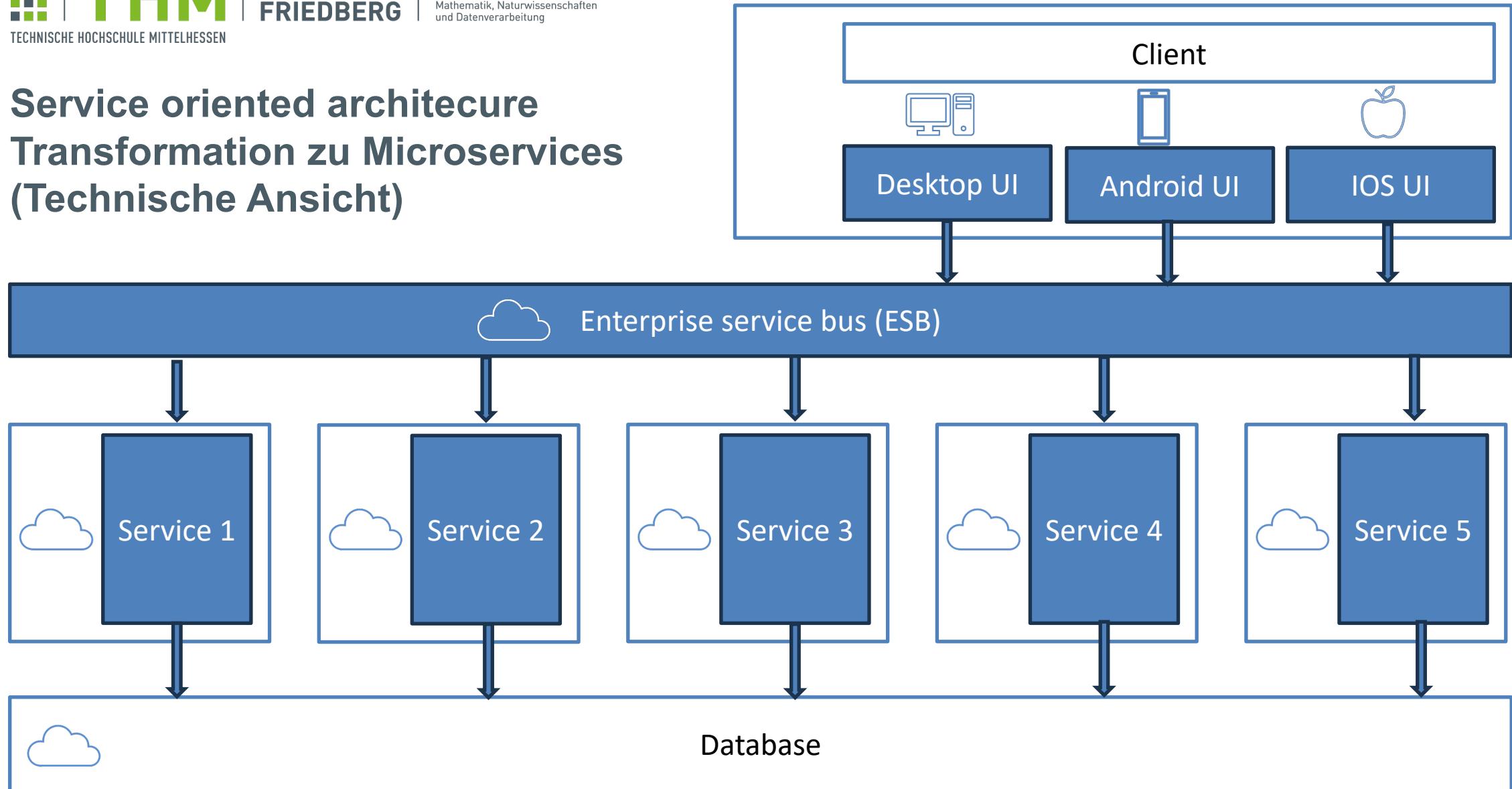
# Service oriented architecture Transformation zu Microservices (Technische Ansicht)



# Service oriented architecture

## Transformation zu Microservices

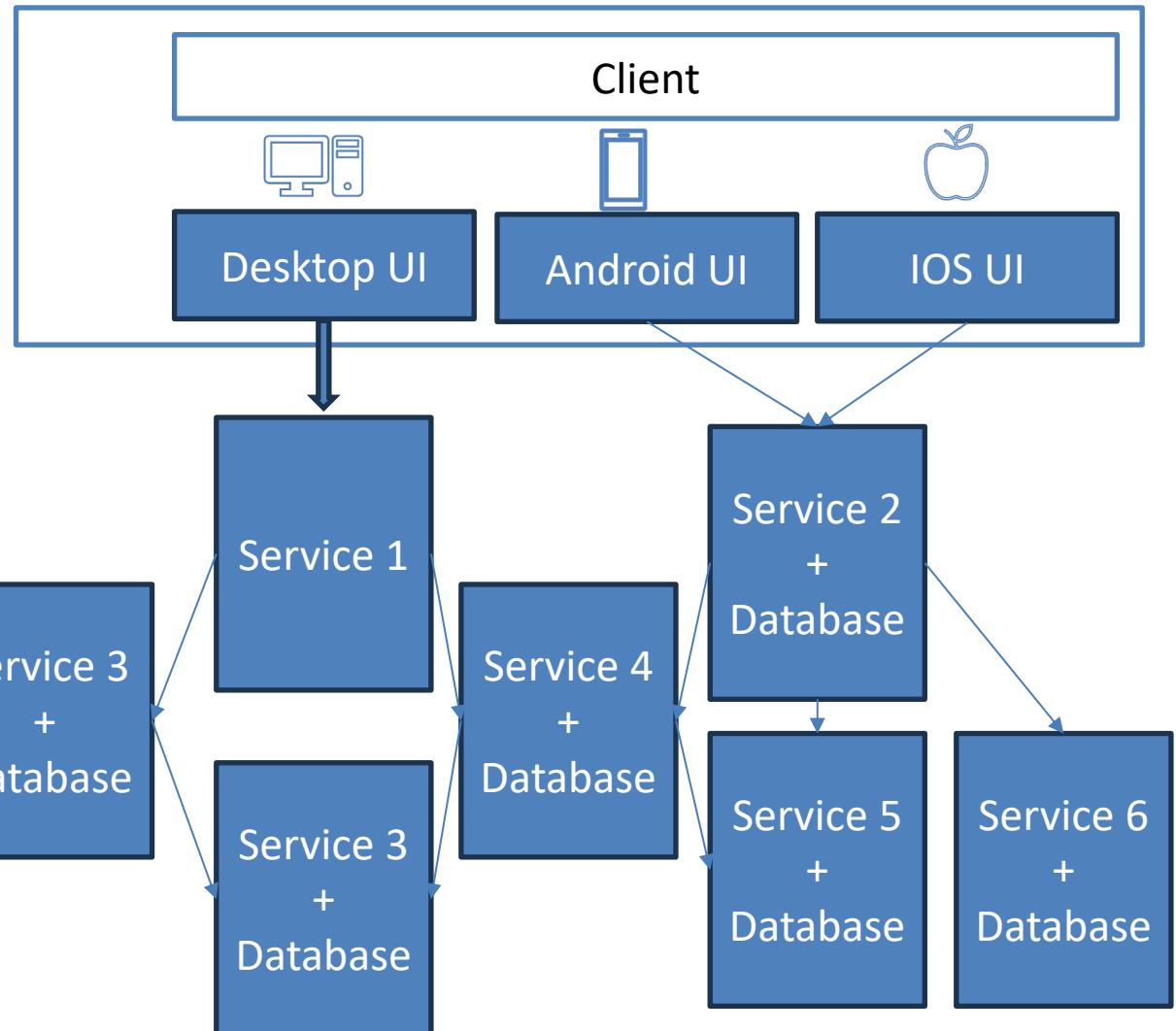
### (Technische Ansicht)



# Service oriented architecture

## Transformation zu Microservices

### (Fachliche Ansicht)



## Microservices aus technischer Sicht

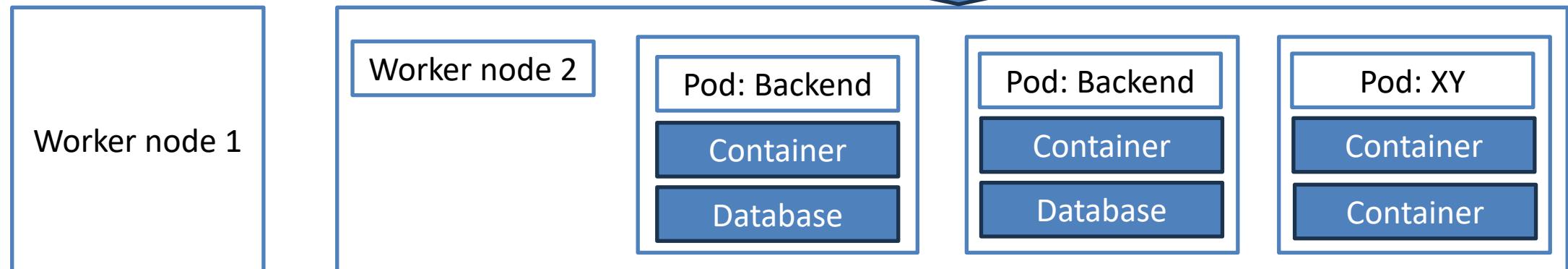
- Kommunikation wird essentiell (REST, gRPC)
  - Ggf. Schlechtere Performance
  - Netzwerksicherheit
- Erfordert mehr Planung als andere Architekturen
- Software Design muss neu gedacht werden
  - Stateless als Paradigma
  - CQRS
  - Event Sourcing
  - Domain Driven Design
- Neue und vielfältigere Anzahl an Technologien
- Programmieren ist nicht mehr die Haupttätigkeit
- Abstraktionsgrad nimmt stärker zu
- Fachlichkeit und Modellierung wird noch wichtiger
- Konfigurationen erfordern viel Arbeit





PaaS  
Serverless

# Service oriented architecture Transformation zu Microservices (Technische Ansicht)



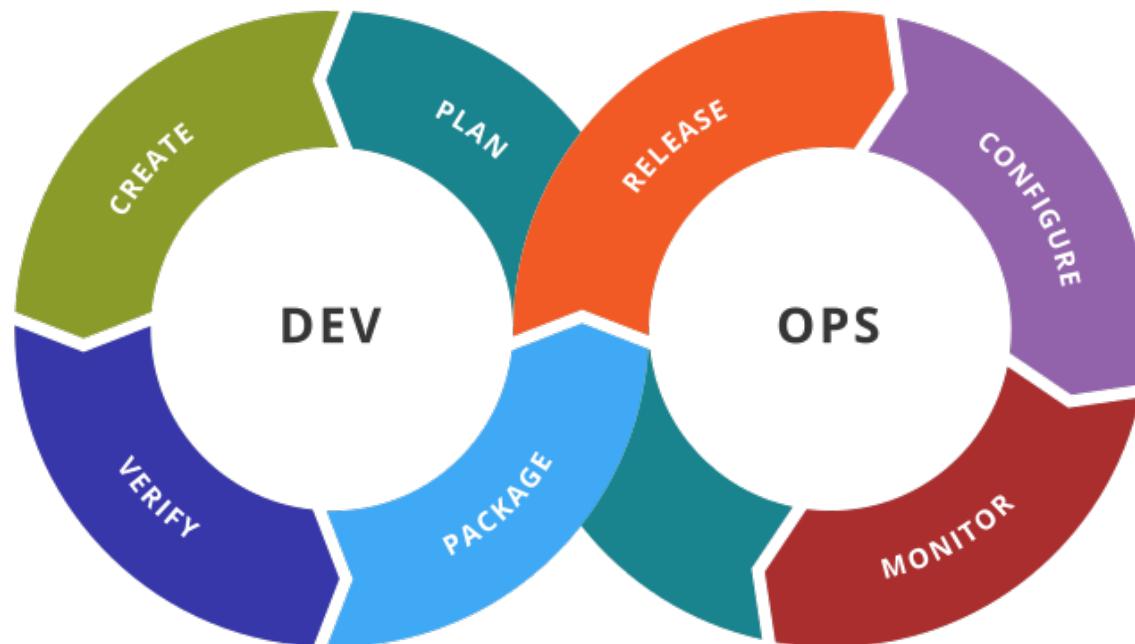
# Serverless

## Nachteile von Kubernetes/Docker

- Dockerfile
- Docker Compose
- Taggen
- Registry
- Ingress.yaml
- Deployment.yaml
- Container neu starten
  
- Ein Haufen Konfiguration und Schritte bis es läuft
- Geht das nicht einfacher?

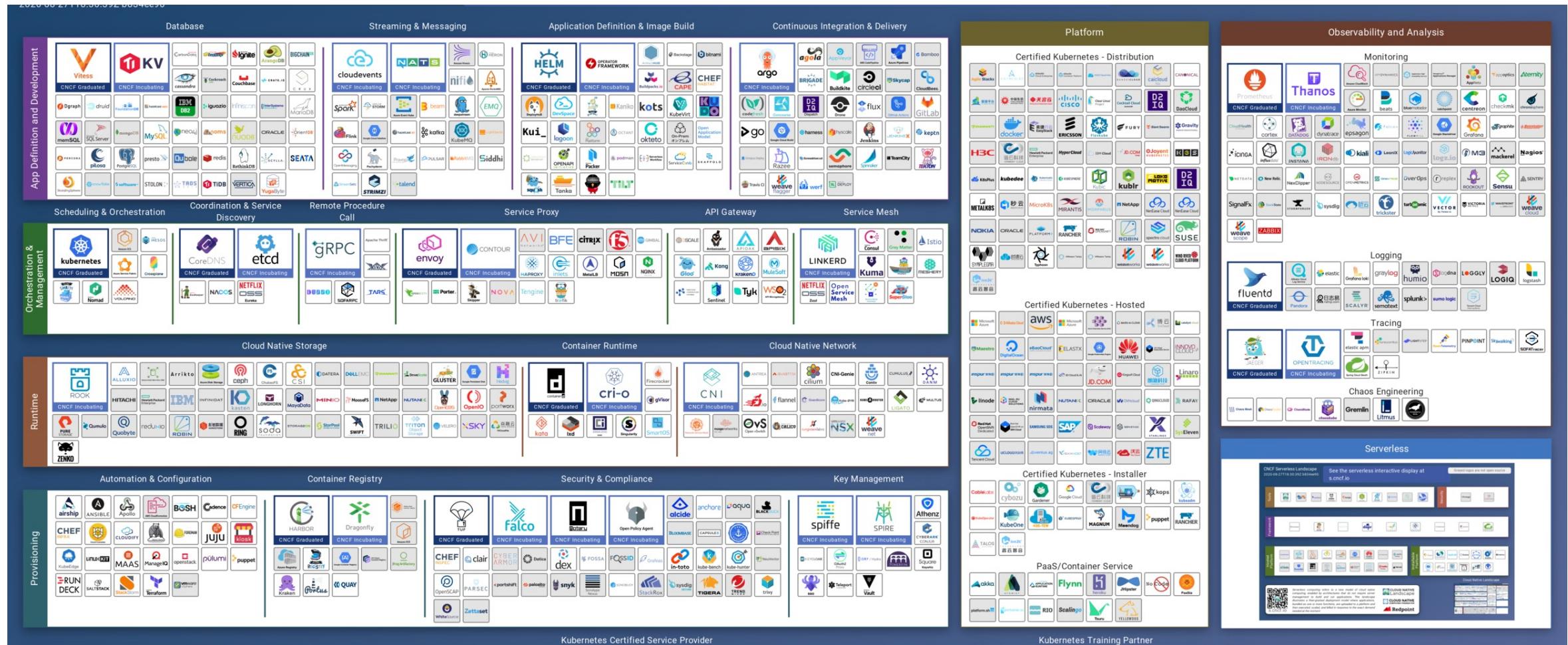
- Weniger Aufwand für die Softwareentwicklung
- Serverless ist wie Strom - Rainer Strophek
  
- Anwendungen direkt in die Cloud publishen
- Beispiel: Visual Studio + .Net + Azure bieten eine direkte Integration
  
- Gefahr von Vendor Lock
- Gibt es noch andere Möglichkeiten?
- Ausblick DevSecOps

## Ausblick: DevSecOps



- Everthing as Code
- Infrastructure as Code
- Automatisierung
  
- Monitoring
- Telemetry Daten
  
- CI/CD Pipelines und Automatisierung
- Automatisiertes Testen

# Cloud Native - Technologien



# Fazit

- Cloud ist kein Trend, sondern der Standard
- Die Migration von On Premise auf die Cloud ist nicht trivial
- Der Umzug von Standardsoftware mittels SaaS ist deutlich einfacher als bei Individualsoftware
- Cloud ist kein alleinstehendes Thema
  - IoT und smarte Vernetzung
  - Künstliche Intelligenz
  - Big Data
  - Edge und Fog Computing
- Es gibt verschiedene Wege und Strategien Individualsoftware auf die Cloud zu bringen
  
- Die Software Welt ist im Wandel und wird sich weiter ändern
- Software Menschen müssen das auch

# Weitere Angaben

## GitHub

- Seminararbeit als PDF
- Präsentation als PowerPoint
- Weitere Veröffentlichungen:
  - Clean Code
  - Funktionale Programmierung
- Link: <https://github.com/Koschnag/>

Email (THM): [cong.chanh.vinzenz.nguyen@mnd.thm.de](mailto:cong.chanh.vinzenz.nguyen@mnd.thm.de)

Email (Beruflich): [c.nguyen@r-ie.de](mailto:c.nguyen@r-ie.de)

LinkedIn: <https://www.linkedin.com/in/cong-chanh-vinzenz-nguyen-28b50219a>

# Fragen, Diskussion und Feedback