

EINFÜHRUNG WEBSERVICES

- Florian Wachs
- florian.wachs@th-rosenheim.de

WARUM WEBSERVICES ?

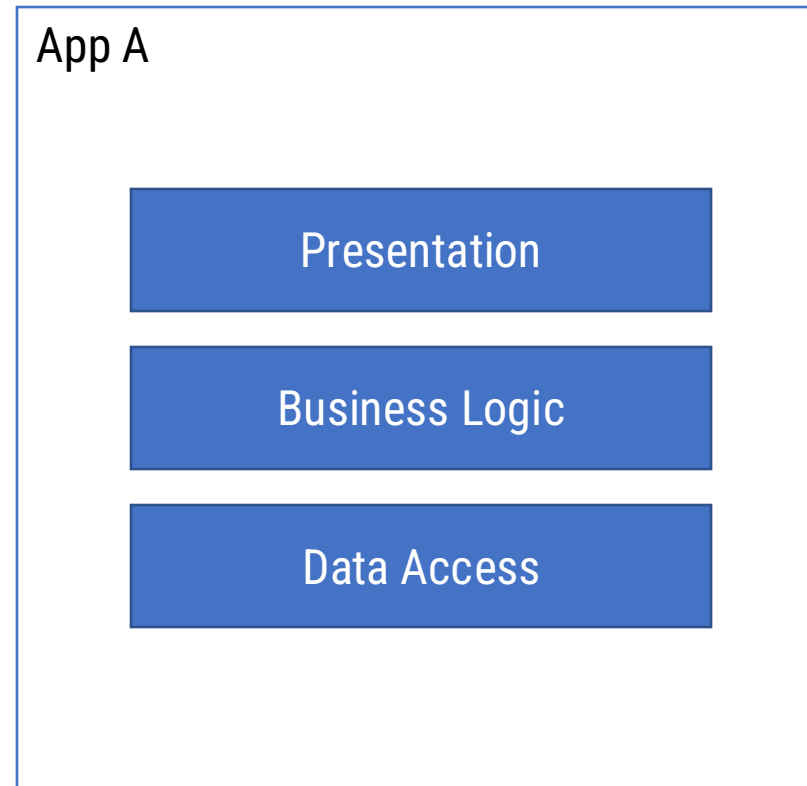
KLASSISCHE ANWENDUNGEN

Presentation

Business Logic

Data Access

KLASSISCHE ANWENDUNGEN



Enterprise-Resource-Planning (ERP)

Presentation

Business Logic

Data Access

Auftrags-
verwaltung

Rechnungs-
erstellung

Lagerplanung

CRM

Analytics /
Berichte

E-Mail-Versand

KLASSISCHE ANWENDUNGEN

Enterprise-Resource-Planning (ERP)

Presentation

Website / Desktopapplikation

Business Logic

Auftrags-
verwaltung

Rechnungs-
erstellung

Lagerplanung

CRM

Analytics /
Berichte

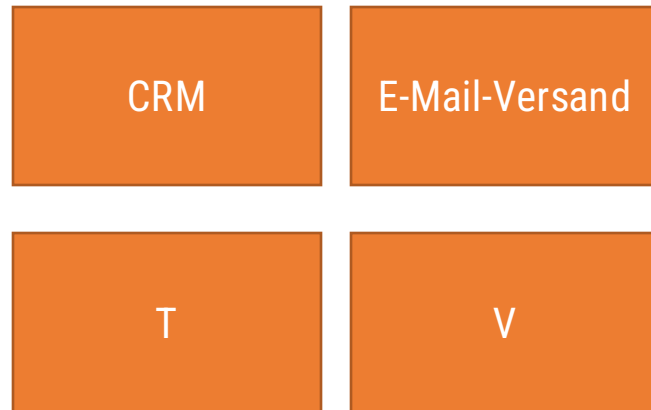
E-Mail-Versand

Data Access

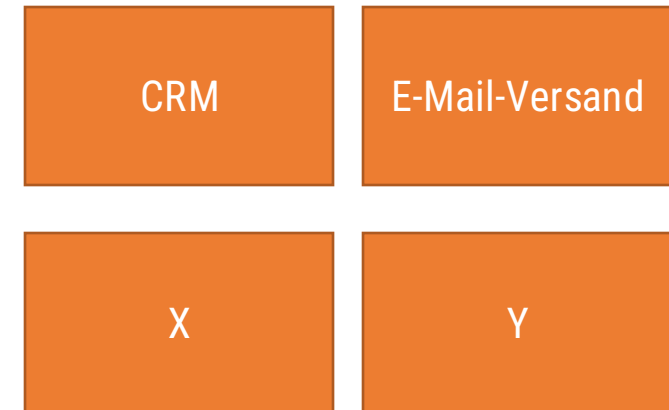
Relationale Datenbank(en)

KLASSISCHE ANWENDUNGEN

App A



App B



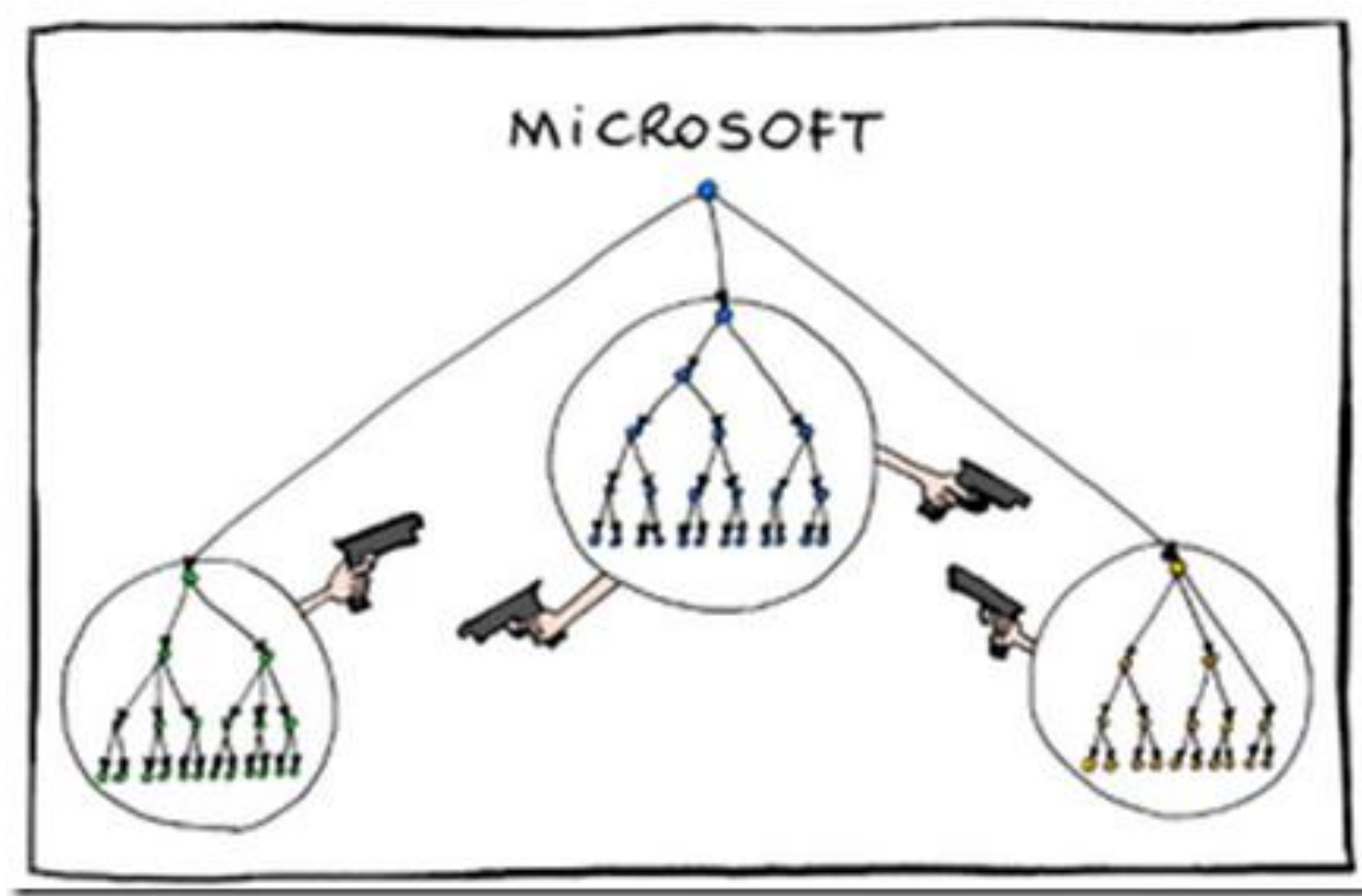
KLASSISCHE ANWENDUNGEN

App A



App B





PROBLEME

Daten und Funktionalität sind nur innerhalb der Applikationsdomäne verfügbar

Identische Funktionalität (z.B. CRM) muss mehrfach erstellt werden*

Kaum Möglichkeiten der Skalierung

Aktualisierungen / Wartungen beeinträchtigen meist die Verfügbarkeit

Daten und Funktionalität können meist nur sinnvoll über den Presentation-Layer verwendet werden

SERVICE ORIENTED ARCHITECTURE

- Buzzword „Wie Web 2.0 nur für Services...“
- SOA ist keine Technologie sondern eine Design Philosophie
- SOA kann, muss aber nicht mit Web Services umgesetzt werden
- Es gibt kein „one-fits-all“
- SOA ist eine lose-gekoppelte Architektur die designt wird, um realen Geschäftsanforderungen des Unternehmens zu befriedigen

SOA| Service Oriented Architecture



Business value over technical strategy



Strategic goals over project-specific benefits



Intrinsic interoperability over custom integration



Shared services over specific-purpose implementations



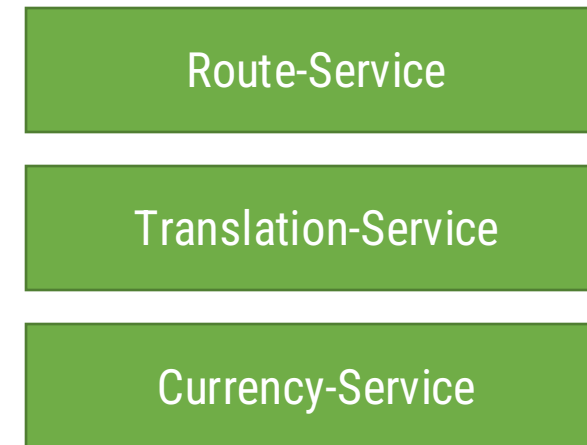
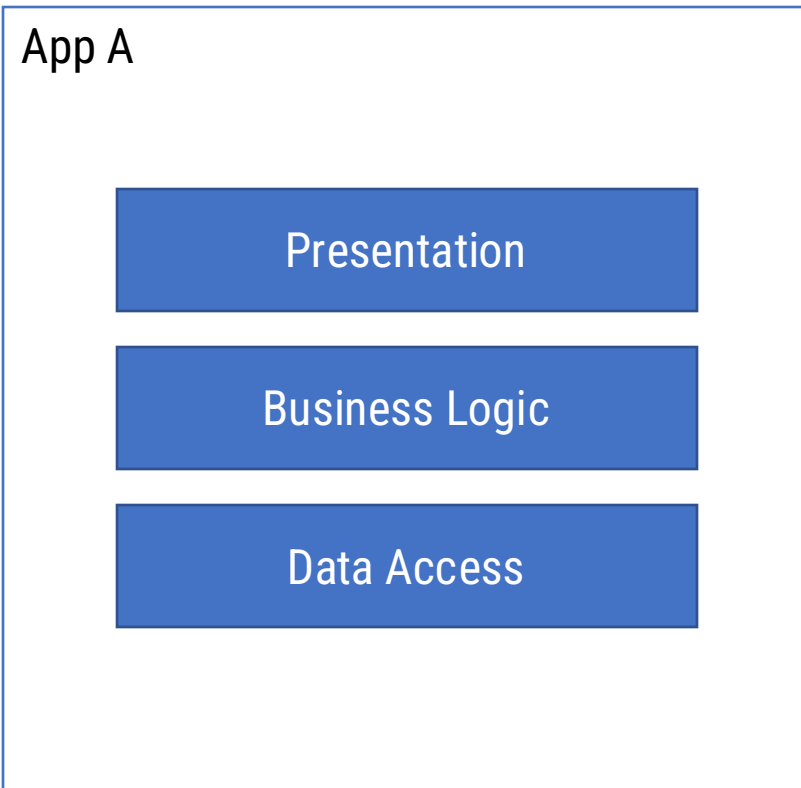
Flexibility over optimization



Evolutionary refinement over pursuit of initial perfection

SZENARIEN FÜR WEBSERVICES

ERWEITERUNG BESTEHENDER APPLIKATIONEN

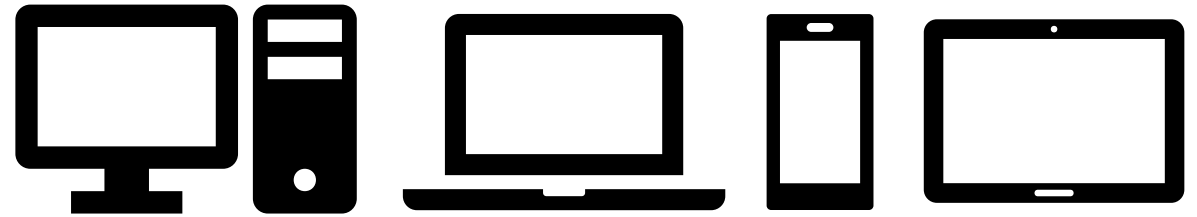


UNTERSTÜTZUNG VERSCHIEDENER CLIENT-TECHNOLOGIEN

Service

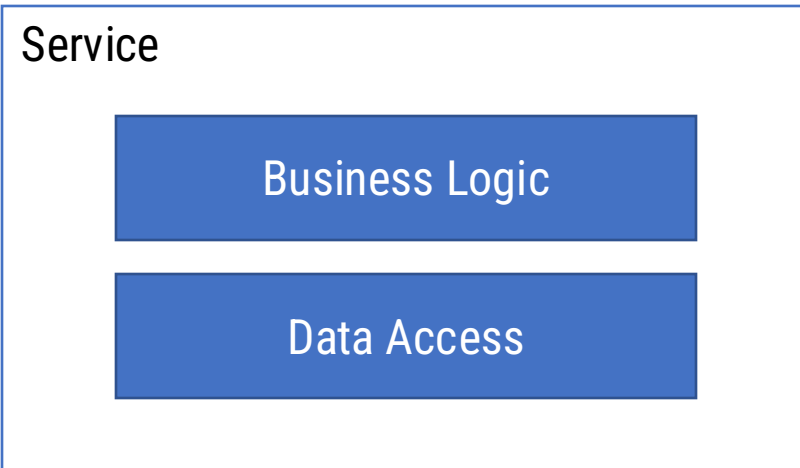
Business Logic

Data Access

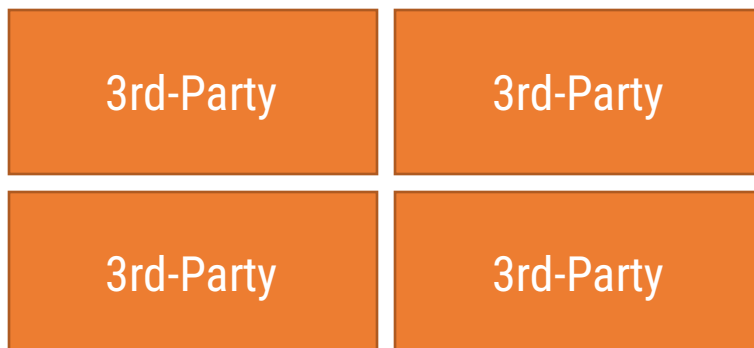


BUSINESS-LOGIK SELBST ALS SERVICE ANBIETEN

BUSINESS AS A SERVICE



BUSINESS AS A SERVICE



BUSINESS AS A SERVICE

**„IMMER MEHR UNTERNEHMEN SEHEN SICH
SELBST ALS SERVICE ANBIETER“**

BUSINESS AS A SERVICE

Die üblichen „Verdächtigen“...

- Microsoft
 - AI Services
 - Bing-Services (Map, Route, Wetter, Finanzen, ...)
 - Azure (Cloud, Infrastructure as a Service, ...)
 - Office 365 / Microsoft Graph
- Google
- Facebook
- Twitter
- Wetterdienste
- ...

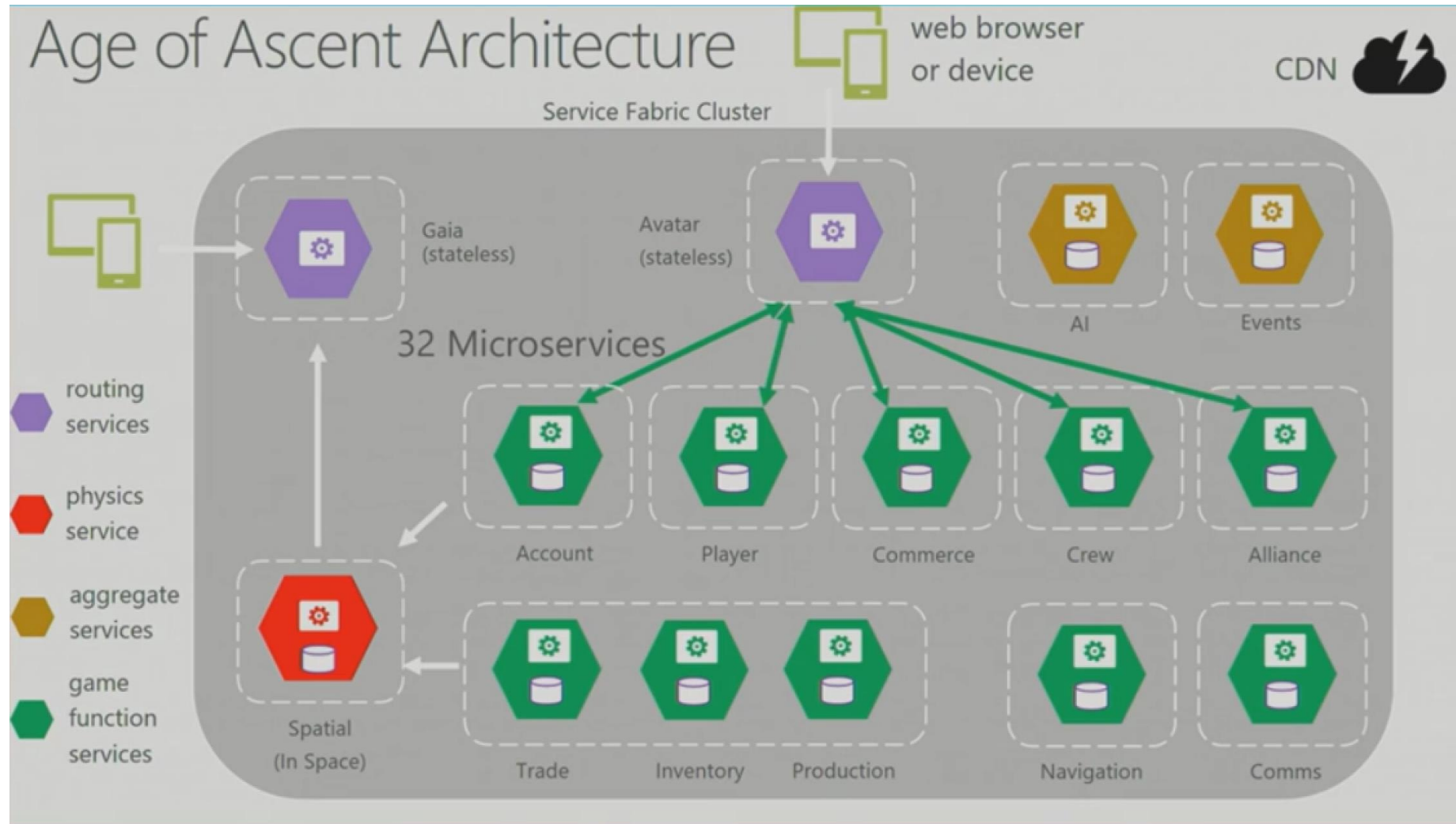
Die „neuen“

- AI
- BIG DATA – Unternehmen u. Start Ups
- Fintec
- Verarbeitende Industrien
- Internet der Dinge (IoT)

AGE OF ASCENT

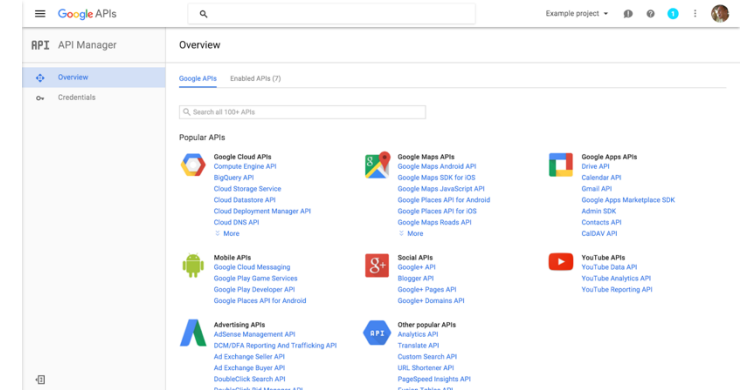
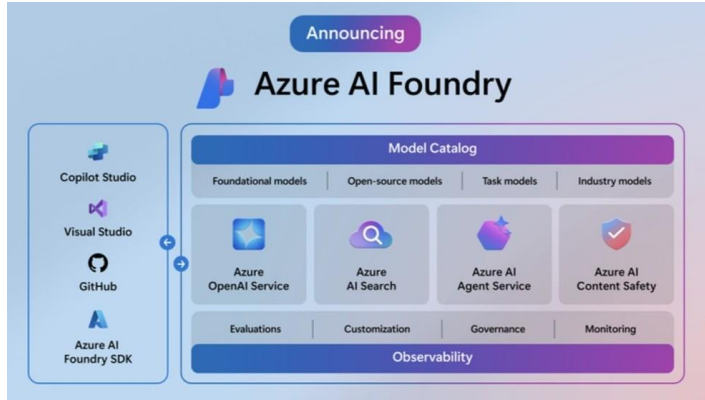


<https://www.ageofascent.com/>



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

WEBSERVICES



- https://developers.facebook.com/docs/graph-api?locale=de_DE
- <https://dev.twitter.com/rest/public>
- <https://developer.microsoft.com/de-de/graph/>
- <https://developer.github.com/v3/>
- <https://api.stackexchange.com/>

WEBSERVICES, EIN ÜBERBLICK

ÜBERBLICK



Standards



Technologie-Stacks



Sicherheit



Management



Dokumentation



Architekturmodelle



Hosting



Software as a Service

Funktionalität meist über Web Services als Mietmodell

Anbieter stellt eigene Software für User bereit, z.B. Office 365



Platform as a Service

Spezielle Laufzeitumgebungen die angemietet werden können und meist eine abstraktere Ebene als IaaS darstellen (Azure, Google App Engine)

Meist muss lediglich definiert werden, welche Komponenten benötigt werden, die PaaS-Lösung kümmert sich dann darum die notwendige Infrastruktur aufzubauen

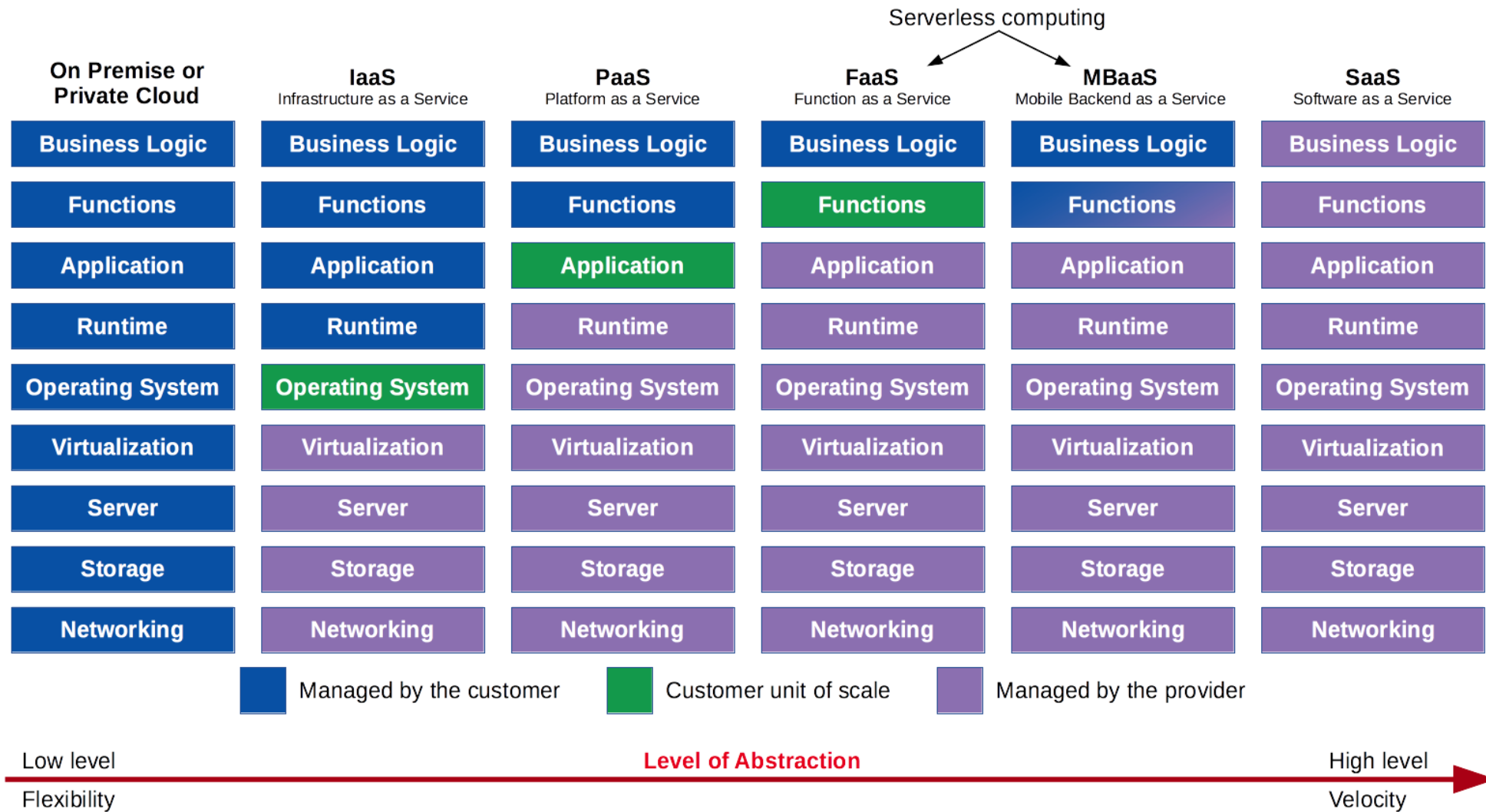


Infrastructure as a Service

Infrastruktur (Server, Speicher, Sicherheit) „on-demand“ über Amazon, Azure, Google

Für eine Anwendung muss die benötigte Infrastruktur (Rechenleistung, Speicher, Netzwerk) selbst zusammengestellt werden

IaaS, PaaS, SaaS



mediacomem.github.io

IaaS, PaaS, SaaS, FaaS

VORTEILE VON WEBSERVICES

VORTEILE



Werden von den „Großen“
vorangetrieben (Google,
Facebook, IBM, Microsoft)



Basieren auf offenen Protokollen,
keine Lizenzkosten



Basieren auf allgegenwärtigen
Internetprotokollen und
Infrastruktur

VORTEILE



Kostengünstig Realisierung von Projekten möglich („Build or buy“)



Interoperabilität zwischen Plattformen, unterstützt auch Heterogenität der Systeme



Fördert dienstorientierte Systemarchitektur die effiziente B2B- und EAI-Lösungen ermöglicht

GRENZEN

Web Services sind nur eine technische Basis für Service Orientierte Architekturen (SOA)

Im Umfeld von SOA werden weitere Standards benötigt und vorhandene Standards müssen weiterentwickelt werden

Bei Interoperabilität gibt es manchmal bedingt durch Lücken in Standards Probleme

Die Frage "Was bewirkt der Aufruf eines Web Service" kann noch nicht ausreichend allgemeingültig beschreiben werden. (Semantik eines Web Services)

Gerade größere Umsetzungen mit Web Services können hohe Herausforderungen an Performance und Wartbarkeit des Systems stellen