



# Service Engineering



## IVS – Arbeitsgruppe Softwaretechnik

Abschnitt: Part 1



# Agenda – Part 1



- Motivation
- Anwendungsarchitektur vs. IT-Unternehmensarchitekturen
- Überblick zum SOA/WOA-Ansatz
- Verändertes API-Verständnis
- Komposition von Serviceangeboten



# Motivation



## frühe Motivation



„...SOA ist ein technologieunabhängiges Architekturkonzept, das Softwarearchitekturen einfacher und flexibler machen soll und dabei die Wiederverwendung bestehender Komponenten unterstützt. SOA ist nicht an bestimmte Technologien wie etwa Web Services gebunden. ...“

Quelle: Reinheimer, S.; Lang, F.; Purucker, J.; Brüggemann, H.: 10 Antworten zu SOA, in HMD 253, dpunkt.verlag, Heidelberg, Februar 2007



# Begriff der IT-Architektur



## Architekturebenen:

- Geschäftsarchitektur – Arbeitsabläufe und Organisationsstrukturen
- Anwendungsarchitektur – verwendete Unternehmensanwendungen
- Technologische Architektur – Betrieb SW- und HW-Infrastruktur

CIO Organisation  
(Informationsmanagement)

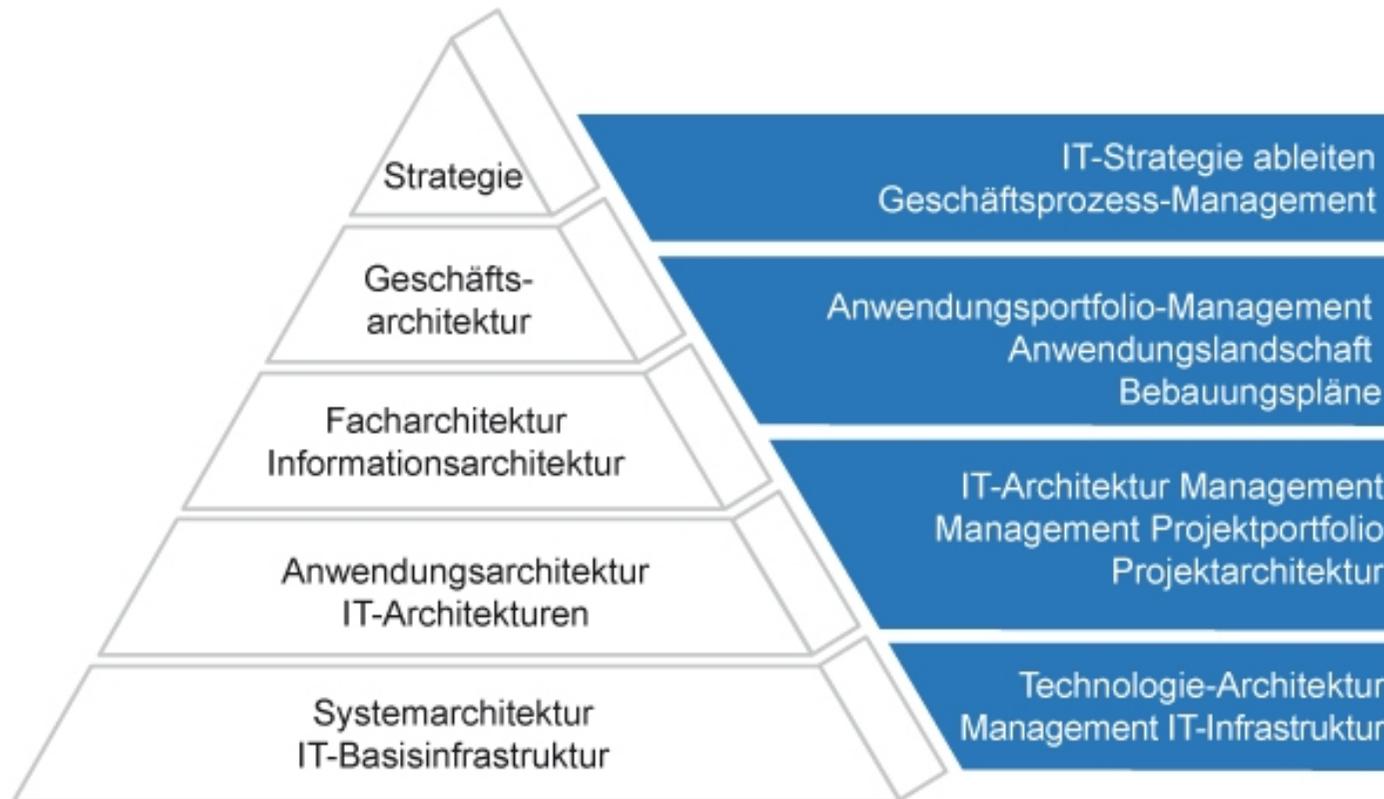
Lösungsanbieter  
(Entwicklung bzw. Integration)

IT Betreiber  
(IT-Management)

Quelle: Conrad, S.; Hasselbring, W.; Koschel, A.; Tritsch, R.: Enterprise Application Integration, Spektrum Akademischer Verlag, München 2006



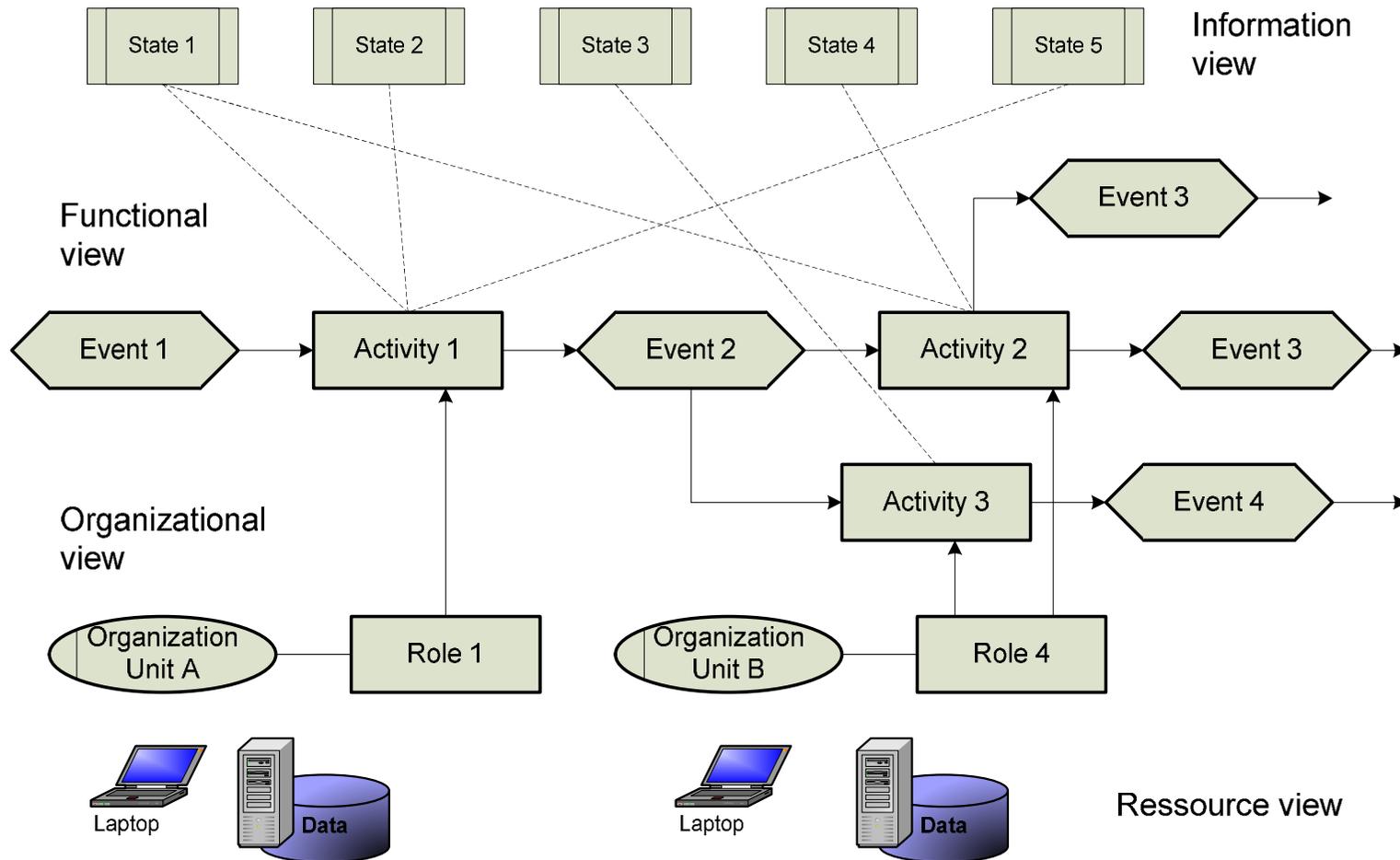
# „Ganzheitliche“ IT-Architektur



Quelle: [http://winfwiki.wi-fom.de/images/9/91/Architekturpyramide\\_dern.jpg](http://winfwiki.wi-fom.de/images/9/91/Architekturpyramide_dern.jpg), Abruf: Oktober 2016



# Begriff der IT-Architektur

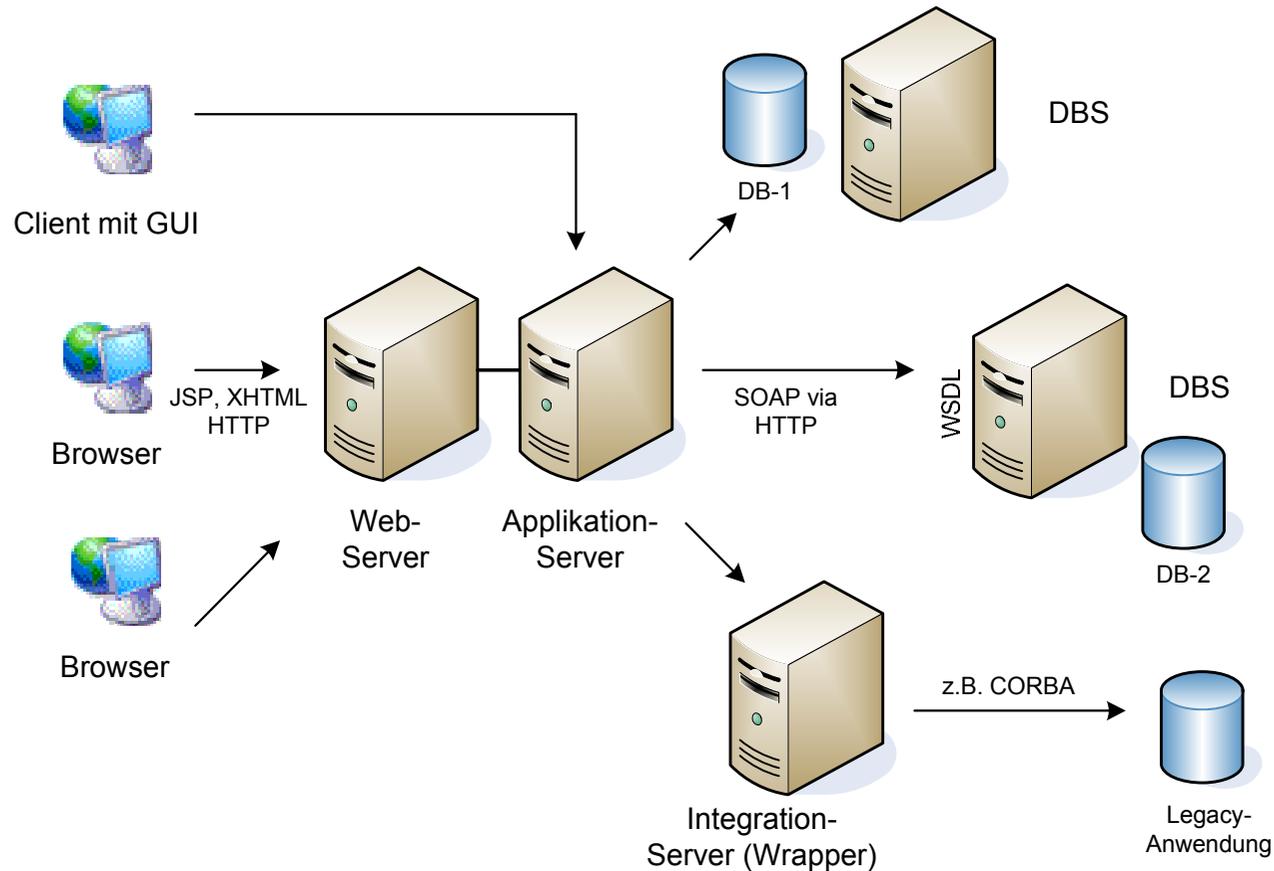




# Anwendungsarchitektur vs. IT-Unternehmensarchitekturen

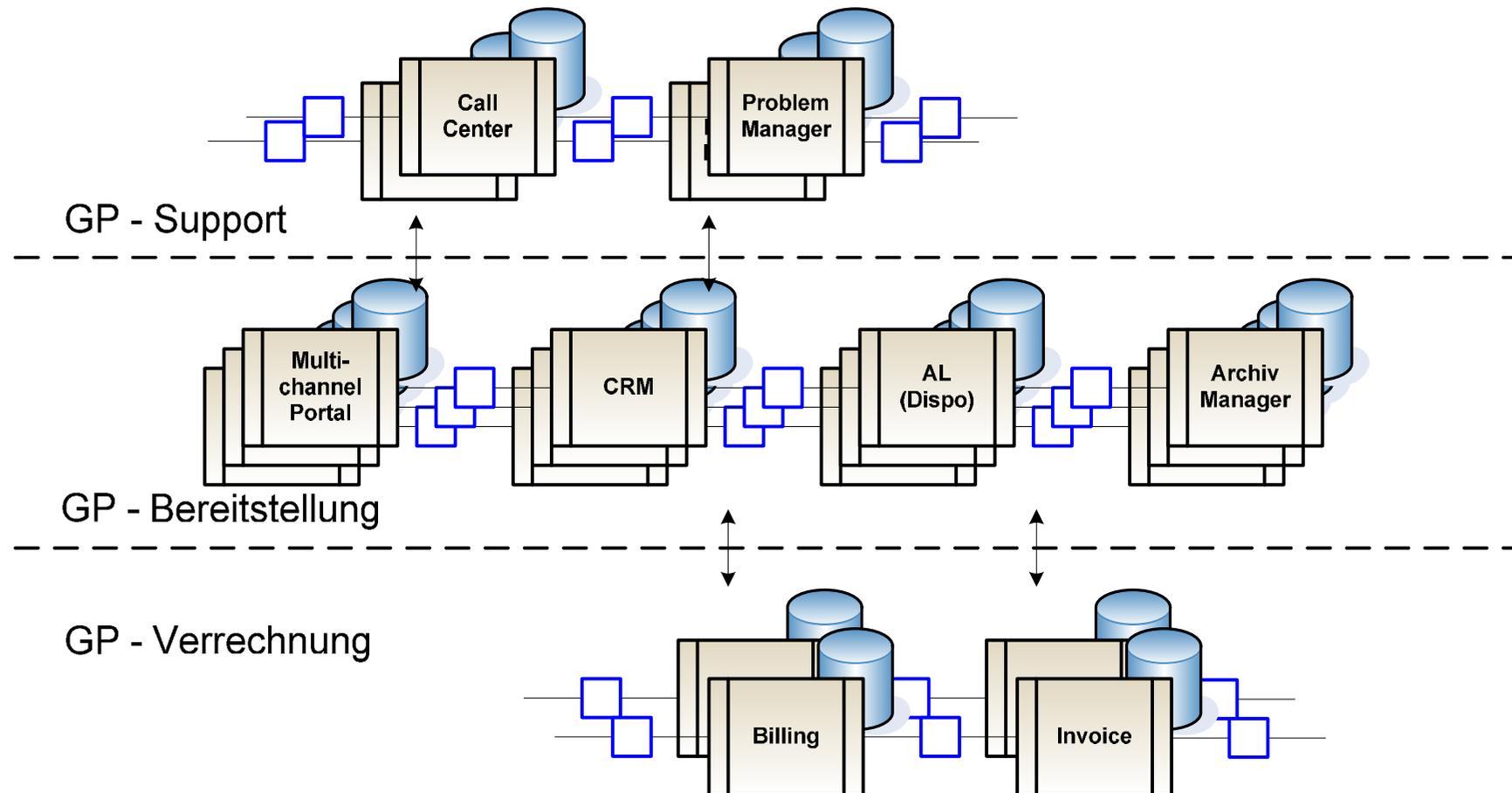


# Client/Server-Architektur moderner Applikationen





# Produkt- und Organisations-zentrierte IT-Architekturen





## Typische Problembereiche

- „Silo“-Applikationen für spezielle Produkte und Dienstleistungen
- Unübersichtliche Anzahl benötigter Schnittstellen (Spagetti)
- Enge Kopplung der verwendeten Applikationen
- Schwerfällige Umgestaltung IT-gestützter Prozesse
- Fallende IT-Budgets (bis zu 40% innerhalb der letzten 7 Jahre)
- Hochgradige Daten- und Funktionsredundanz
- Performance der GP schwer nachvollziehbar
- Heterogenität der verwendeten Technologien bzw. Techniken



# Überblick zum SOA-Ansatz



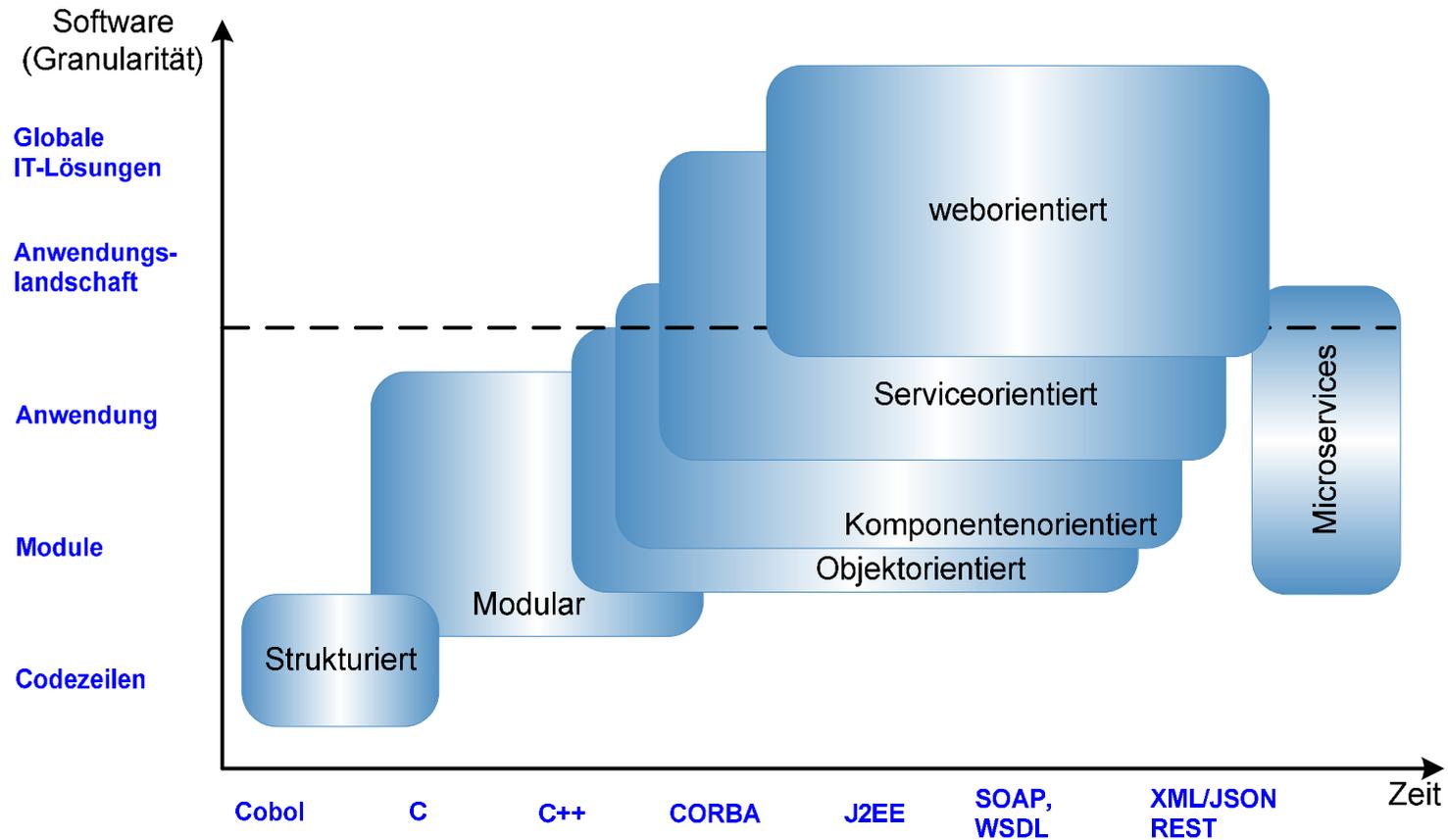
# Zielstellungen einer SOA/WOA



- Schnelle Bereitstellung neuer Lösungen (Integration vs. Entwicklung)
- Transparenz/Kennntnis verfügbarer fachlicher Serviceangebote
- Identifizierung potentieller Funktions- und Datenredundanzen
- Verhindern redundanter Entwicklungen
  - Wiederverwendung von bereits vorhandenen Services
  - Reduktion der Kosten für Entwicklung & Wirkbetrieb
- Etablierung von internen und externen Servicemarktplätzen
  - Evolutionäre Optimierung vorhandener Servicelandschaften
  - Reduktion der Kundenbindung - Flexibilisierung
- Einheitliche Spezifikation der vorhandenen Services



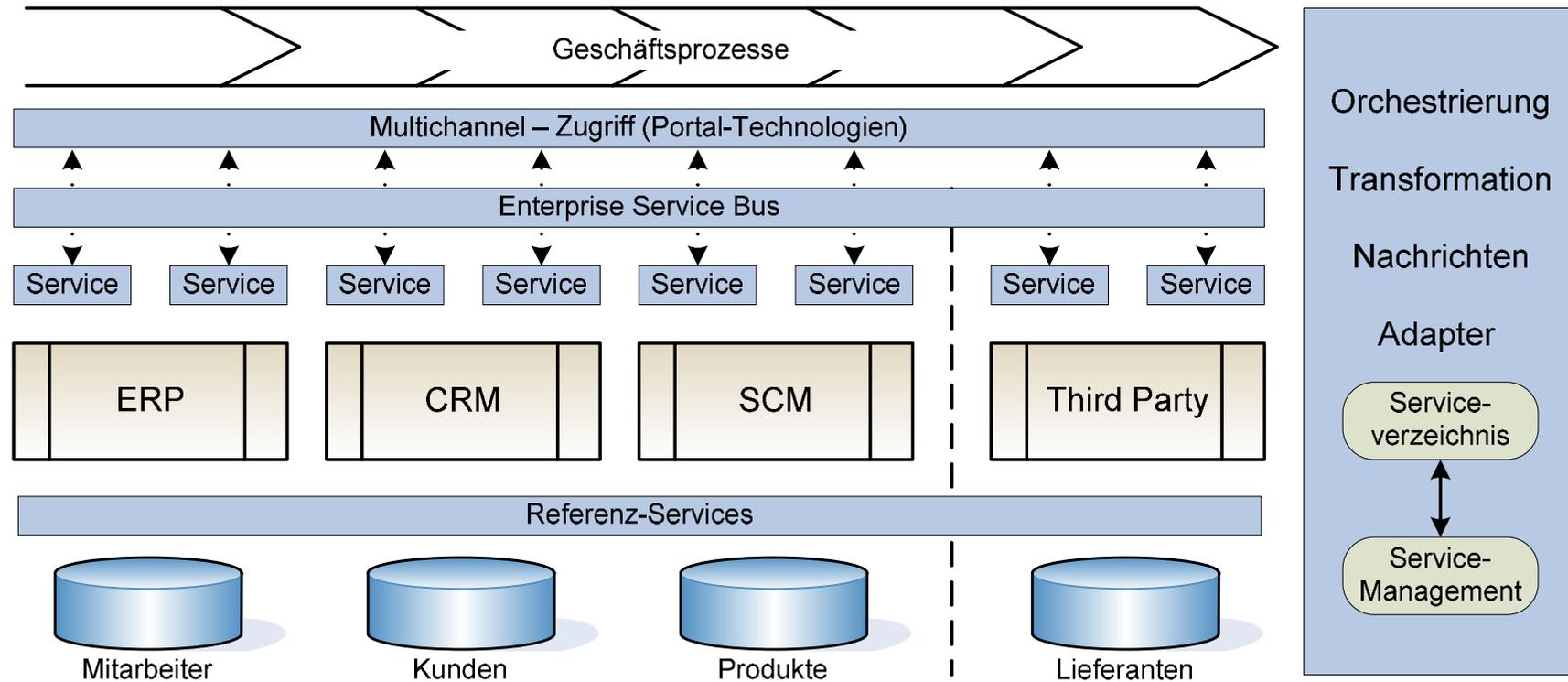
# Bezugsbereich einer SOA/WOA



Unter Verwendung von: Frisius, O.: Service Oriented Architecture – Revolution oder Buzzword, sd&m AG, GI / ACM Stuttgart, Dezember 2005

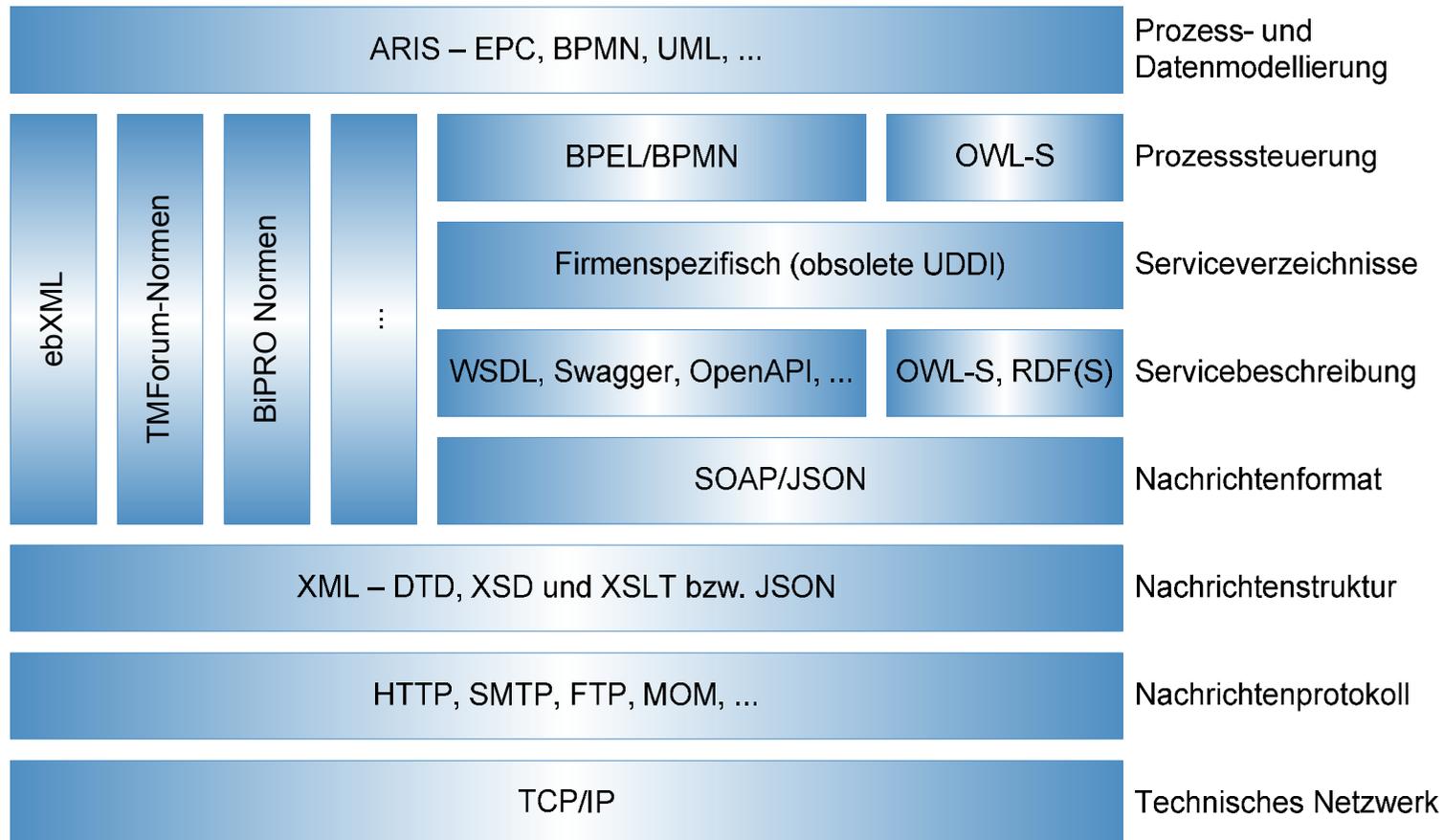


# Bestandteile einer SOA





# Web Services als Implementierungsalternative





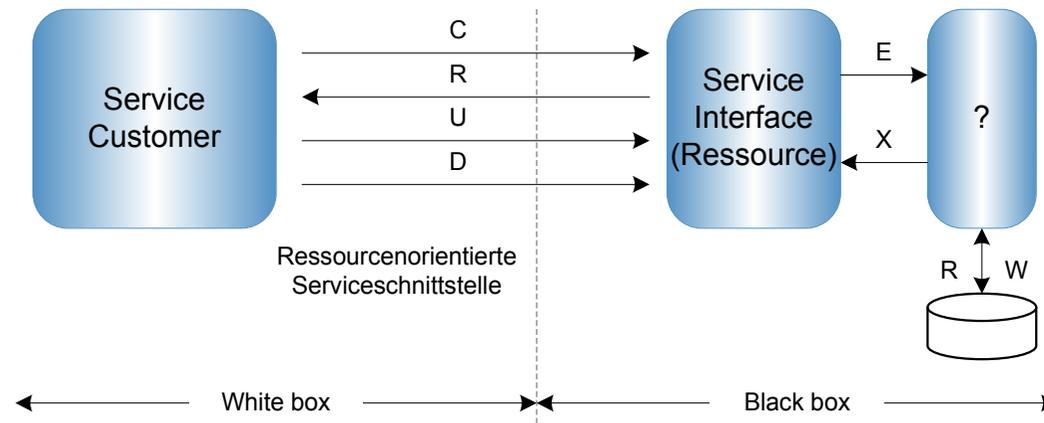
# Architekturstile



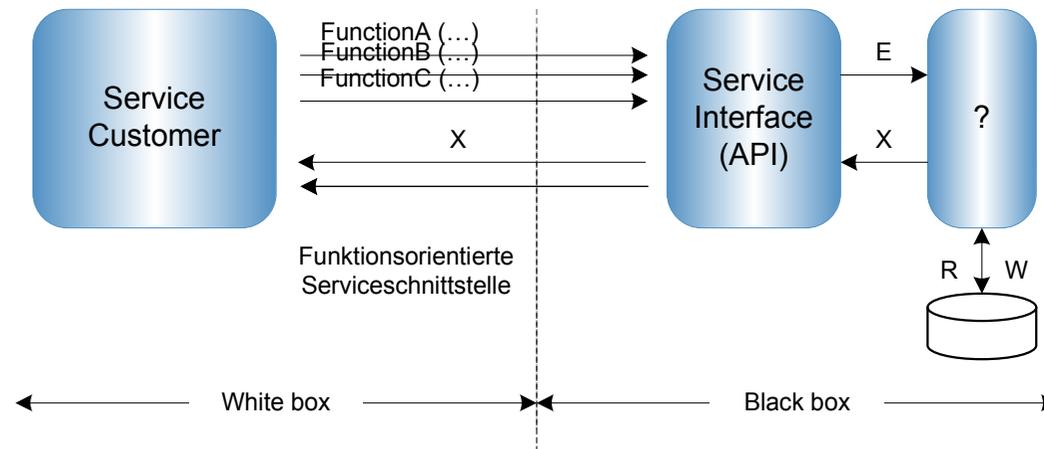
- Schnittstellenorientierter Architekturstil (z.B. Object Web).
  - DCE RPC
  - CORBA/RMI
- Nachrichtenorientierter Architekturstil (z.B. SOA).
  - MOM/JMS (z.B. IBM WebSphere)
  - SOAP-Services (Transportprotokollunabhängig)
- Ressourcenorientierter Architekturstil (z.B. WOA).
  - Originärer Architekturstil des Web (HTTP/URI)
  - Ähnliche Vorgehensweise bei SQL



# Architekturstile im Vergleich



REST-Style



RPC-Style



# Verändertes API-Verständnis



# APIs als geschäftlicher Faktor



„Application Programming Interfaces (API's) have gone from a something that only developers and architects once discussed to emerge as a capability that is central to many successful companies business strategies and a key focus of many of their senior leadership teams.”

Quelle: [http://syd.apidays.io/APIdays\\_program.pdf](http://syd.apidays.io/APIdays_program.pdf)



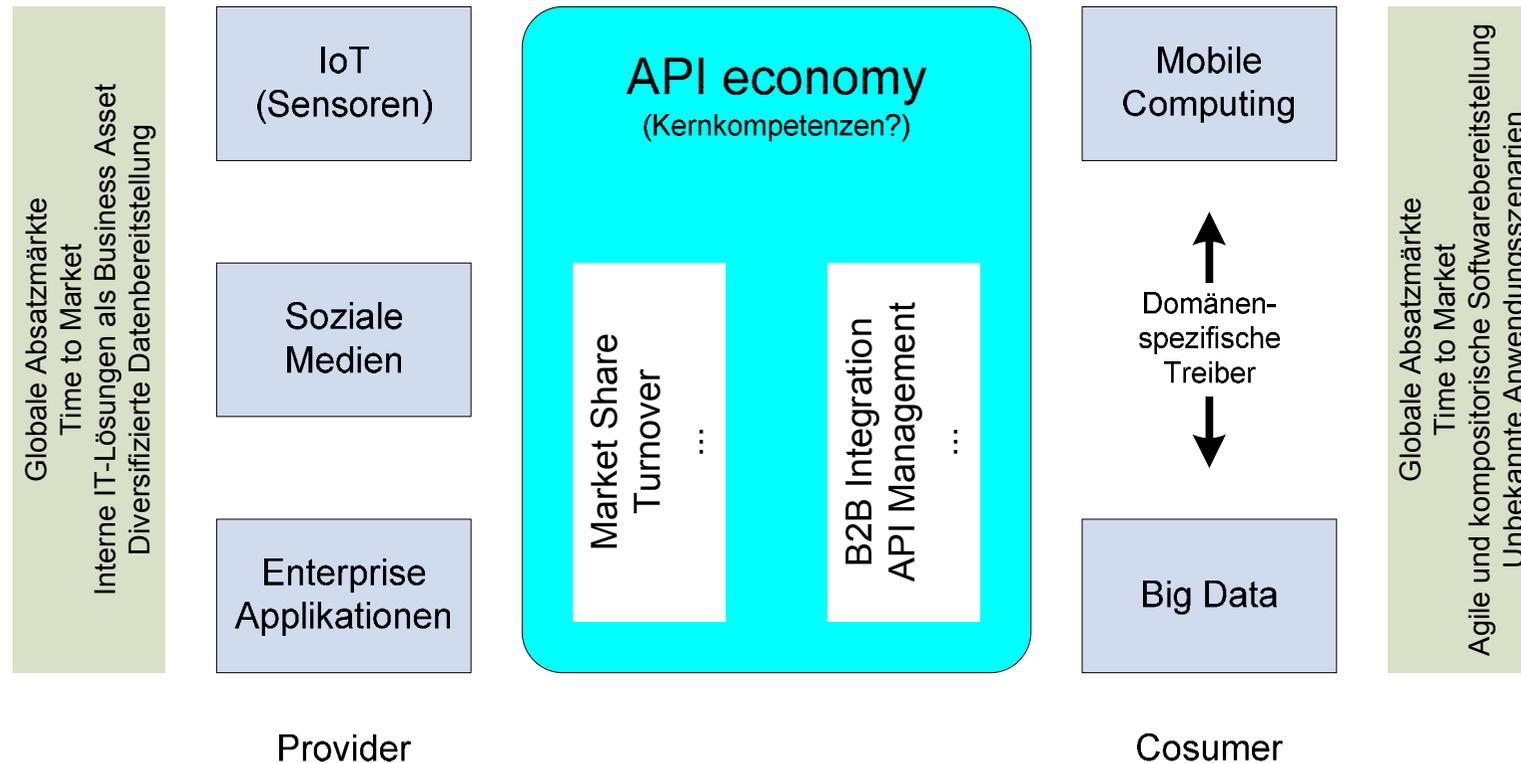
# Eigenschaften Web APIs



- Interoperabilität via TCP/IP und HTTP/HTTPS.
- Lose Kopplung und Modularisierung.
- Performance/Skalierbarkeit (Cluster- und Cloud-Lösungen).
- Separierung von Verantwortlichkeiten (Arbeitsteilung).
- Wiederverwendung (kompositorische Softwareentwicklung).
- Einfachheit in der Anwendung.



# Fokussierung auf Kernkompetenzen





# Begriff der API-economy



„Geschäftliches Handeln, das auf automatisierten, multilateralen, dynamischen und vergleichsweise anonymen Kompetenznetzwerken hoch spezialisierter Partner beruht.“

Quelle: Resch, O.: API-Economy – eine Situationsbestimmung, in Proc. BSOA/BCloud2015, Shaker-Verlag Aachen



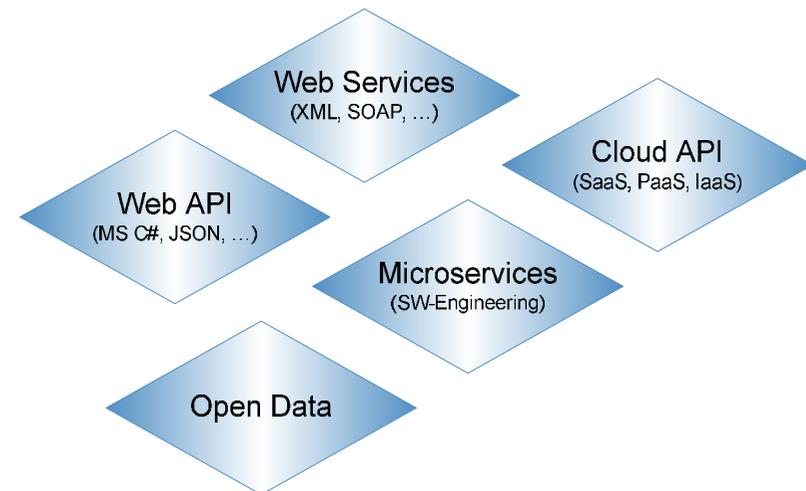
# Komposition von Serviceangeboten



# Komposition von Serviceangeboten



“In the past, we've been builders of custom software, or deployers of packages. In the new, agile application development, we'll find that reuse and assembly will be the keys. Application development organizations can't code themselves into the future!”



Quelle: Hotle, M.; Vecchio, D.: AD Scenario – Tying Agile Development, BPM and Architecture Together, Gartner Symposium ITxpo, Cannes/France 2005



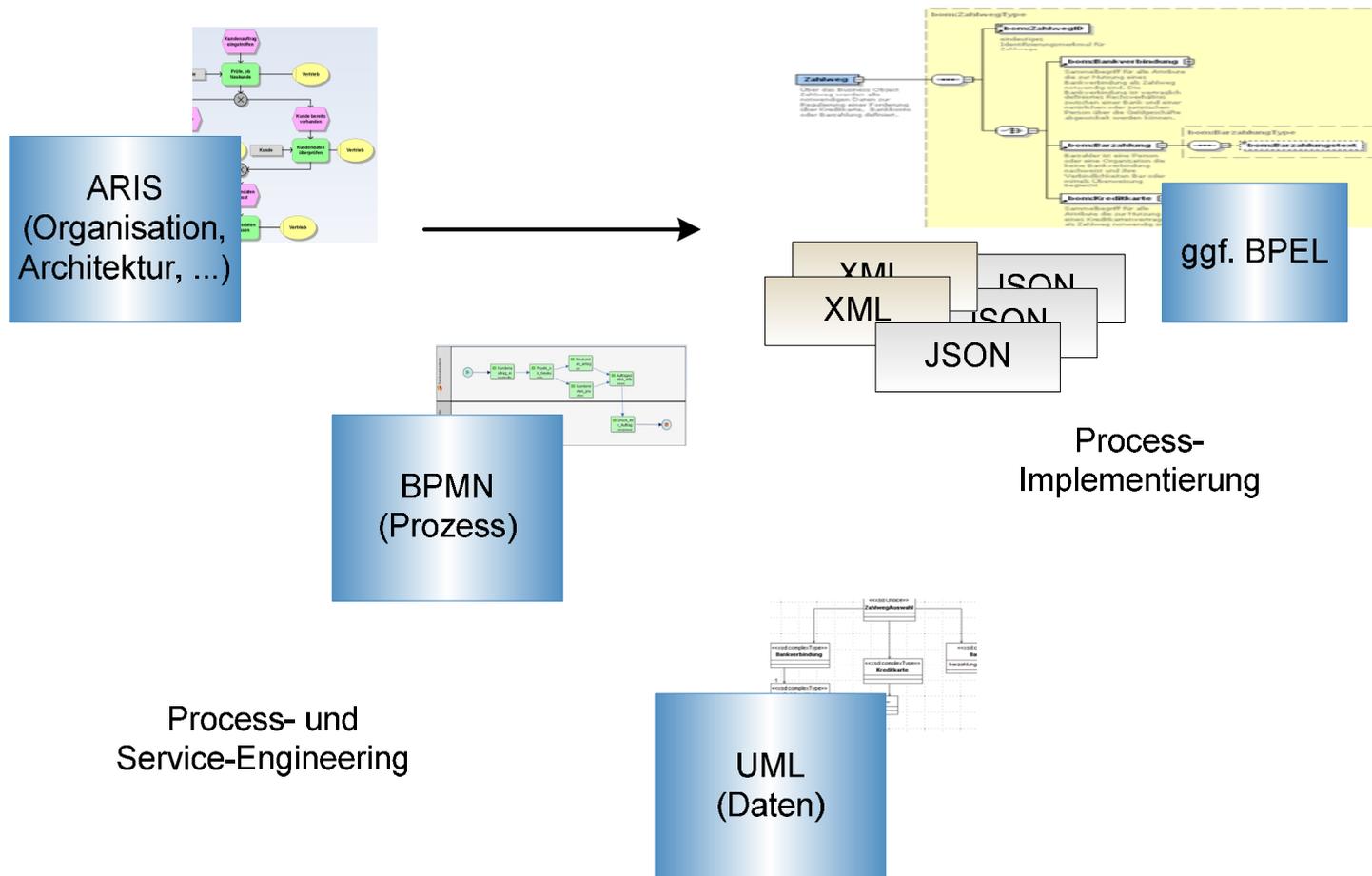
# Komposition von Serviceangeboten



- BPEL (Business Process Execution Language) wurde im Jahr 2003 von IBM, BEA und Microsoft eingeführt. BPEL ist eine XML-basierte Sprache zur Beschreibung von Geschäftsprozessen, deren einzelne Aktivitäten durch Webservices implementiert sind. Weiterhin kann mit BPEL ein Webservice selbst beschrieben werden.
- Web Service - Choreography Definition Language (Ablösung von WSCI) ist ein W3C-Standard zur Modellierung abstrakter Prozesse (Choreografie). WS-CDL wurde entworfen als komplementäre Modellierungssprache zu anderen Sprachen, die zur Orchestrierung von ausführbaren Prozessen verwendet werden.



# Komposition von Serviceangeboten





# Komposition von Serviceangeboten

