

Forum

Geschäftsprozesse 07

Damit Sie mit BPM, SOA & Business Integration erfolgreich sind!

23. und 24. Mai 2007, Vista3 Event Location, Wien



SOA, EAI, Web Services

Was ist der richtige Ansatz für eine prozessorientierte IT-Architektur



ZT Prentner IT GmbH . Ziviltechnikergesellschaft für Informations- und Kommunikationstechnologie – staatlich befugt und beeidet.

A-1010 Wien . Walfischgasse 8/18 office@zt-prentner-it.at . www.zt-prentner-it.at



Inhaltsverzeichnis

- Einleitung
- IT-Architektur
- Service Orientierte Architekturen – Alter Hut neu verpackt ?
- Die Entwicklung der letzten 10 Jahre
- Zentrale Elemente heutiger SOA - Architekturen
- Entwicklungsumgebungen für SOA - Architekturen
- Fallbeispiele: Österr. Service Orientierte (verteilte) Software Lösungen
- Zusammenfassung



Einleitung



ZT Dr. Wolfgang Prentner

- IT-Ziviltechniker und gerichtlich zertifizierter Sachverständiger
- Geschäftsführer der ZTPRENTNERIT GmbH
- Vorsitzender der Bundesfachgruppe Informations-Technologie der Bundeskammer der Architekten und Ingenieurkonsulenten
- E-Government Beauftragter des Bundeskomitees der Freien Berufe Österreichs. Dazu zählen die Kammern der Ärzte, Apotheker, Notare, Patentanwälte, Rechtsanwälte, Wirtschaftstreuhänder und Ziviltechniker
- Mitglied der Plattform Digitales Österreich im Bundeskanzleramt

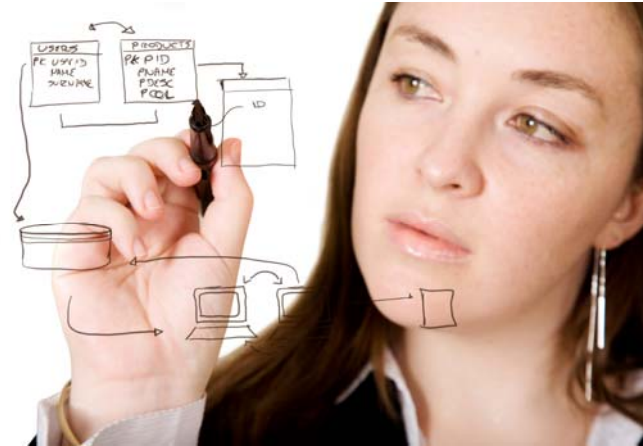




ZTPRENTNERIT – IT-Architekt und Generalplaner

gesetzlich geregelte Aufgaben

- Beraten
 - Planen
 - Prüfen
 - Überwachen
 - Koordinieren
 - Treuhandschaften
 - Errichtung von Urkunden – Zertifikate (staatlich befugt und beeidet)
-
- keine ausführenden Tätigkeiten
 - keine Entwicklung und Vertrieb von Software und Hardware
 - Unabhängiger Dritter
 - gesetzliche Grundlage - Ziviltechnikergesetz 1993



ZTPRENTNERIT – IT-Architektur



IT-Architektur
beraten, planen und prüfen



Marktakzeptanz

Die Akzeptanz des IT**Ziviltechnikers** beim Kunden im Management und in den IT-relevanten Abteilungen ist äußerst hoch.

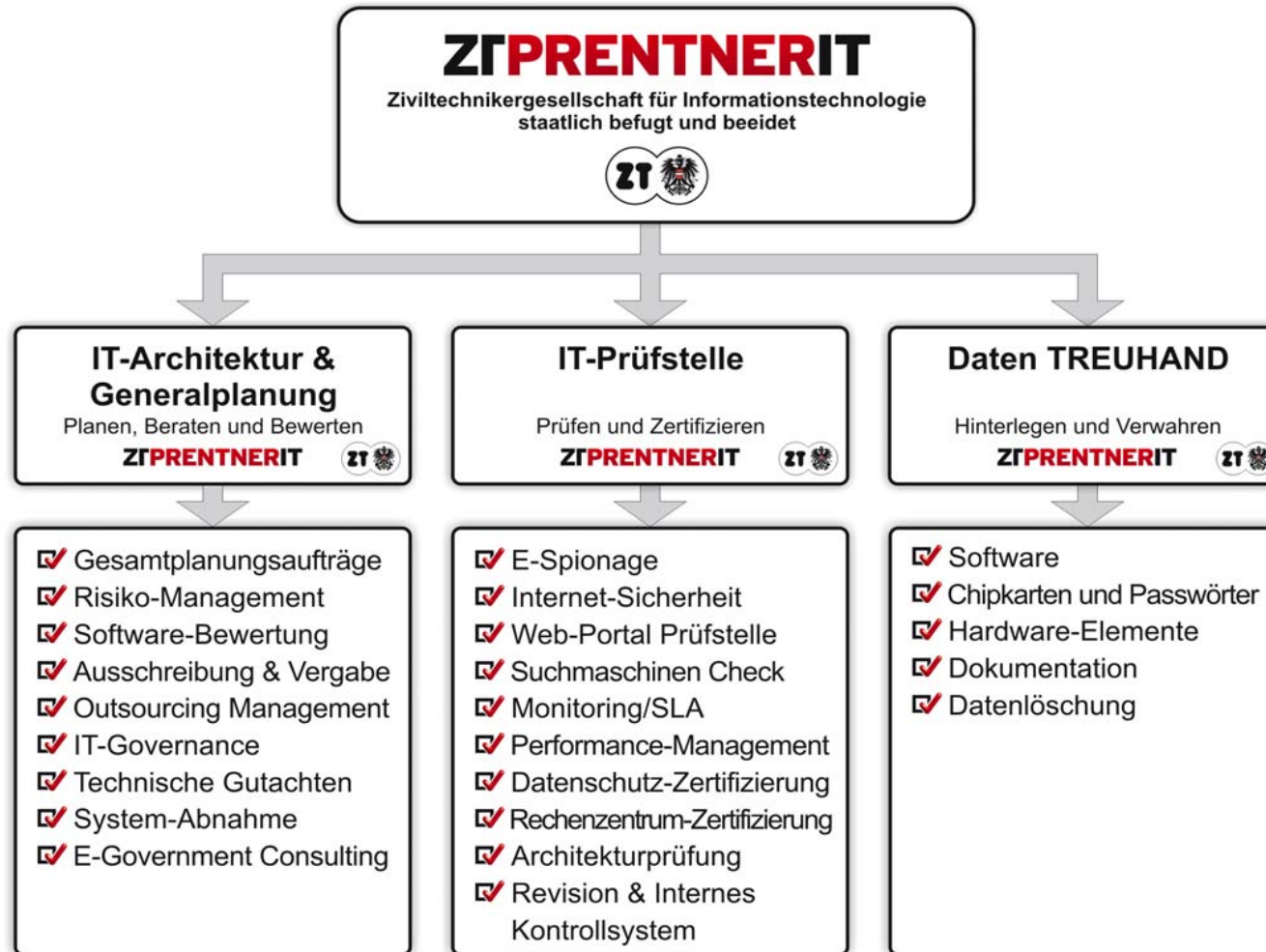
- Fachliche Kompetenz
- Unabhängigkeit
- Objektivität
- Vertrauen und Sicherheit
- Staatliche Reputation



.... sichern den Projekterfolg.

- Beispiele: Ap, N, As

Dienstleistungsüberblick



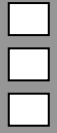


IT-Architektur



Was ist IT-Architektur

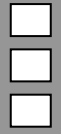
- Architektur ist **Struktur**
- Architektur schafft **Qualität**
- Architektur macht Systeme **verständlich**
- Architektur ist der Rahmen für **flexible Systeme**



IT-Architektur

Es ist eine offensichtliche Wahrheit, dass auch eine perfekte Architektur nutzlos bleibt, wenn sie nicht verstanden wird ...

Die Architektur zu dokumentieren, ist der kritische, krönende Schritt zu ihrer Erschaffung.
[Bachmann2000]



Wir bauen Software wie Kathedralen
zuerst bauen wir - dann beten wir.

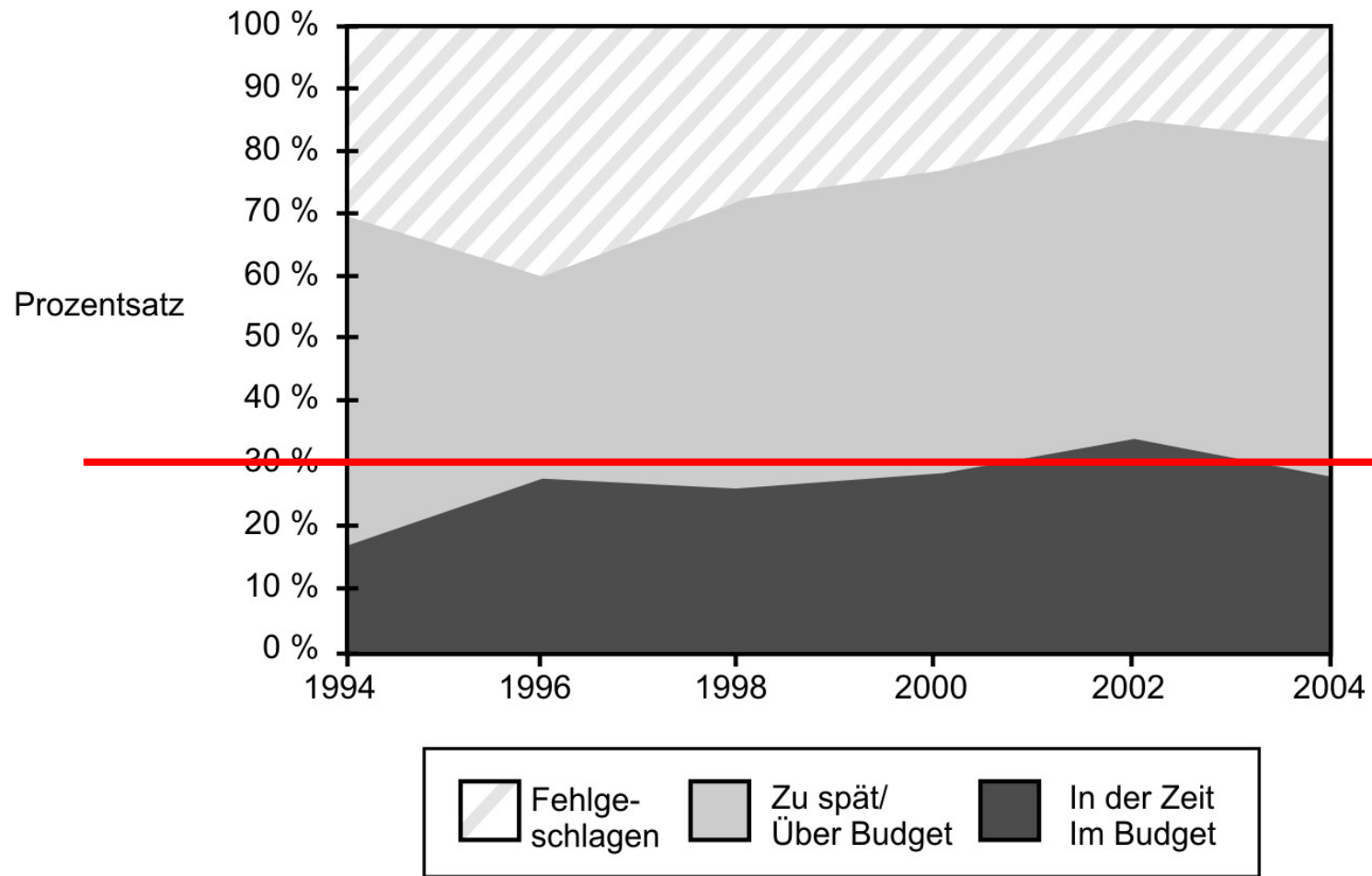
[Gerhard Chroust]



Softwarearchitektur:
Das ist die Königsdisziplin des Software-
Engineering.

[Prof. Ernst Denert in Siedersleben04]

Erfolgsfaktor – Software Projekte





Das „Rad“ der Software Industrie

In der Regel wird das »Rad« mehrmals neu erfunden. Damit erlaubt sich die Software-Industrie einen Luxus, der auf Dauer nicht bezahlt werden kann. (Prof. Balzert)

Fazit:

Trennen von Planung und Entwicklung. Sie senken Kosten, steigern die Qualität, die Transparenz und die Wirtschaftlichkeit Ihrer Projekte.

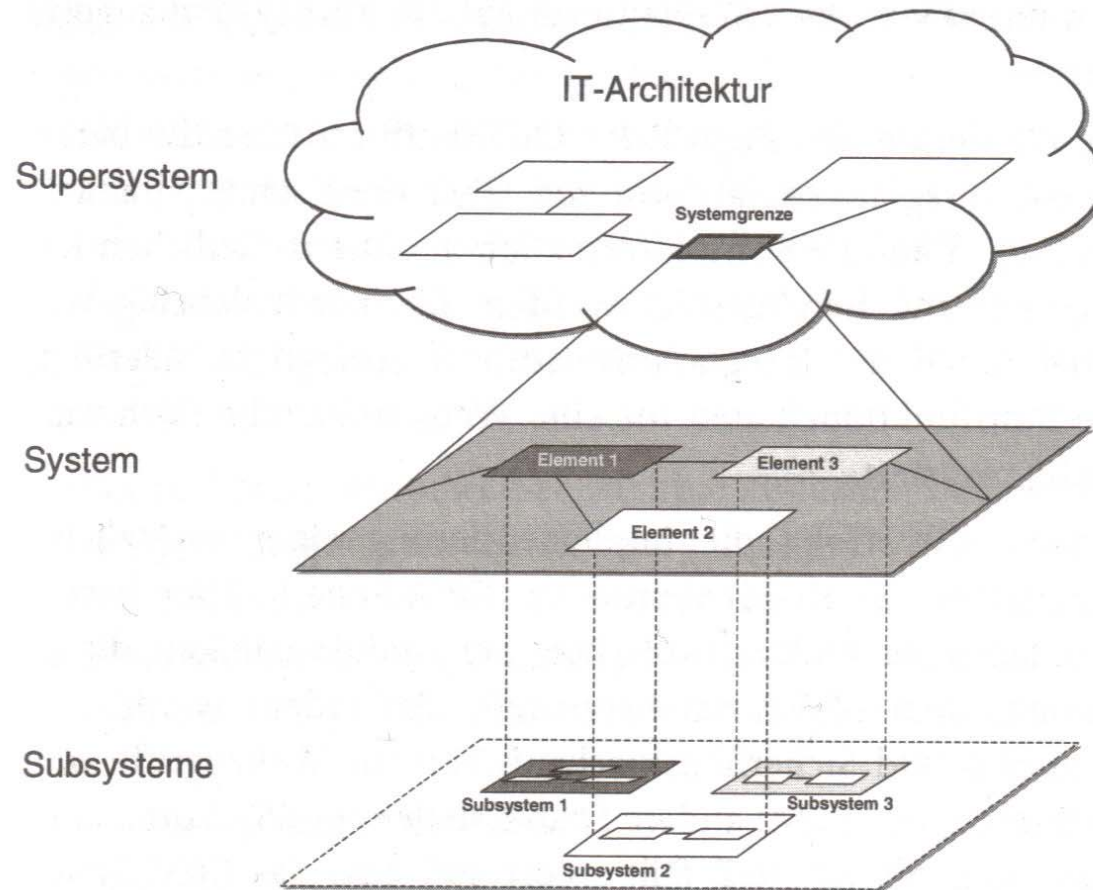




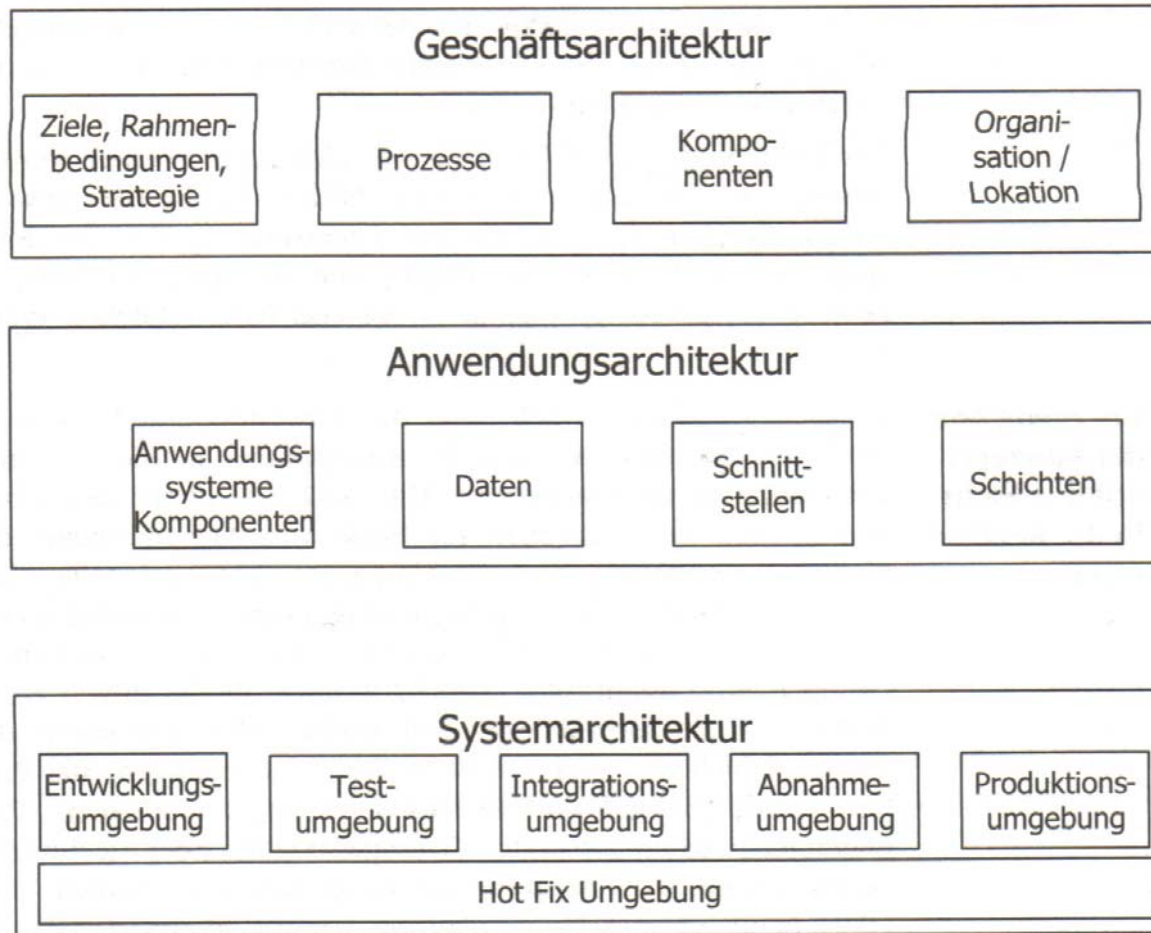
IT-Architektur und ZTPRENTNERIT

- Wir vereinfachen
- Wir konstruieren und entwerfen
- Wir prüfen und bewerten
- Wir kommunizieren und dokumentieren
- Wir senken Kosten
- Wir haften für Beratungs- und Planungsfehler

Systemebenen im Kontext der IT-Architektur

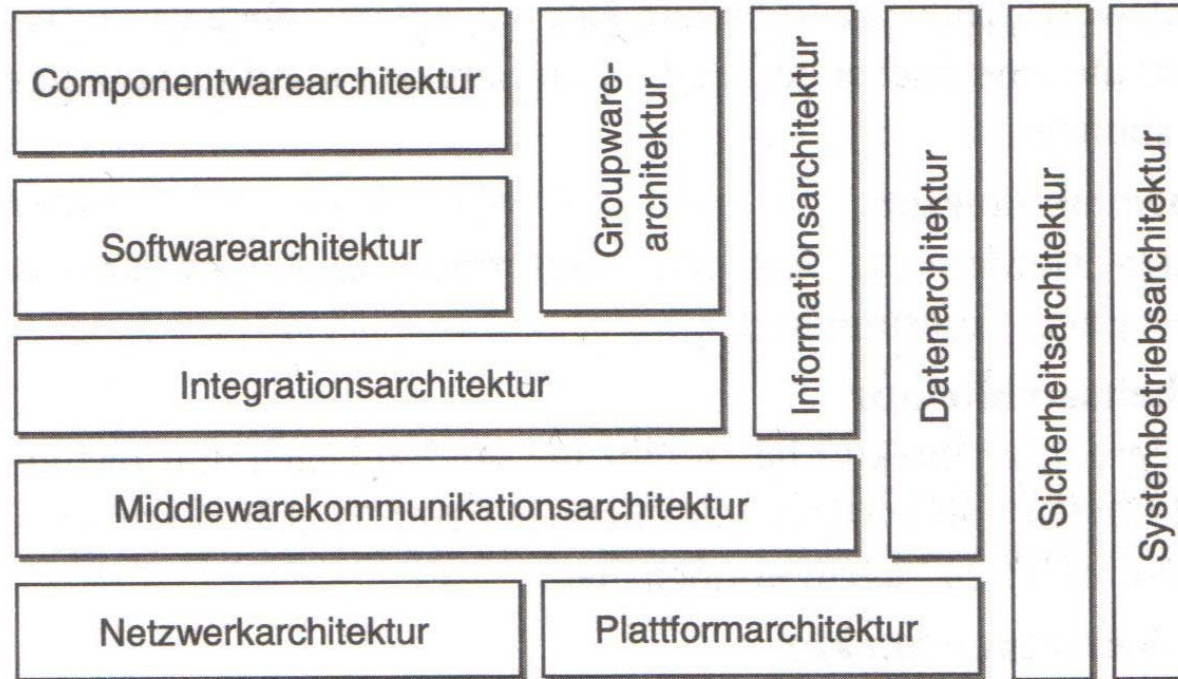


IT-Architekturschichten



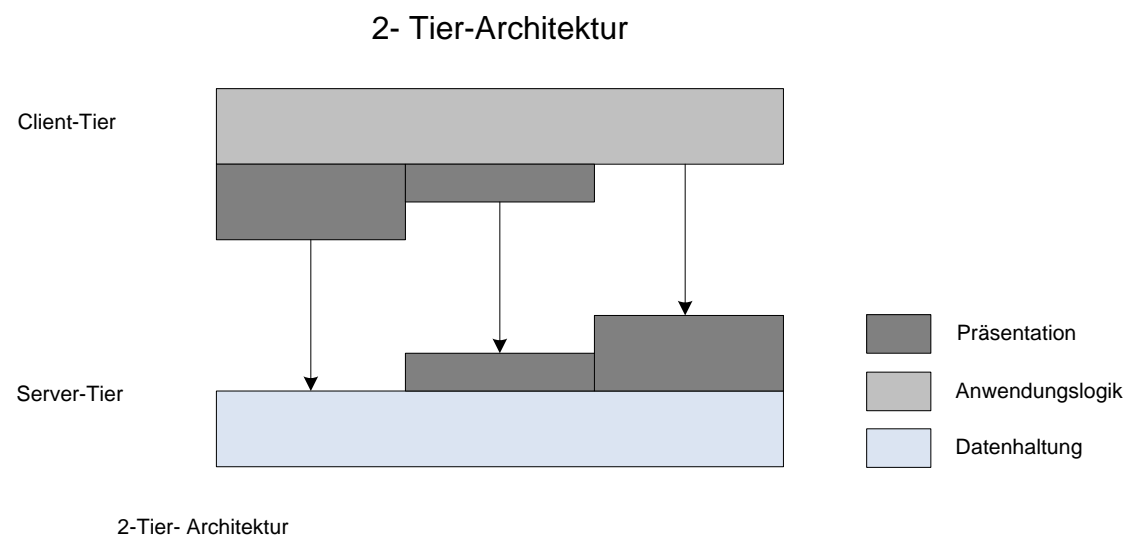


Architektur-Bausteine



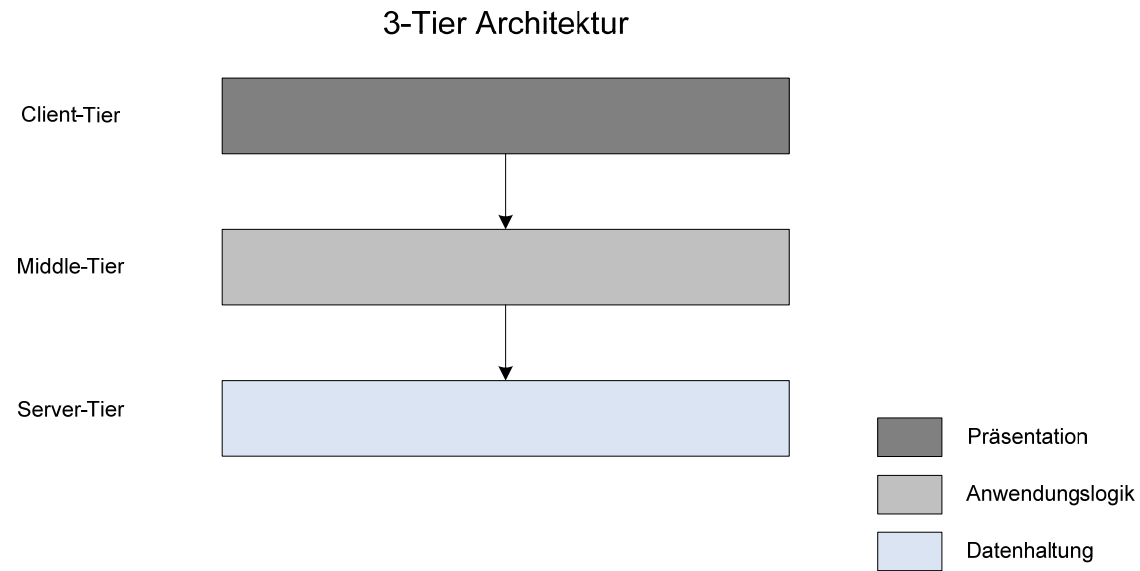


2-Tier Architektur

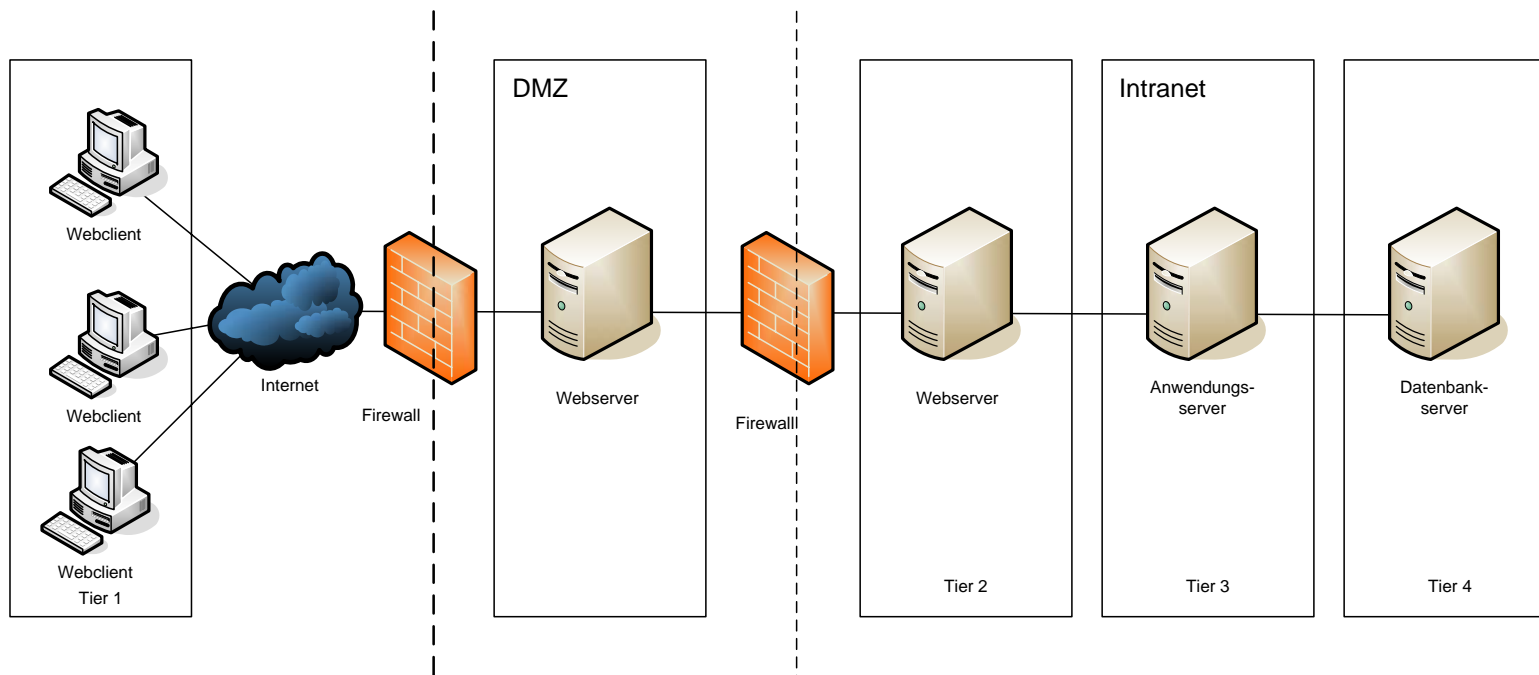




3-Tier Architektur



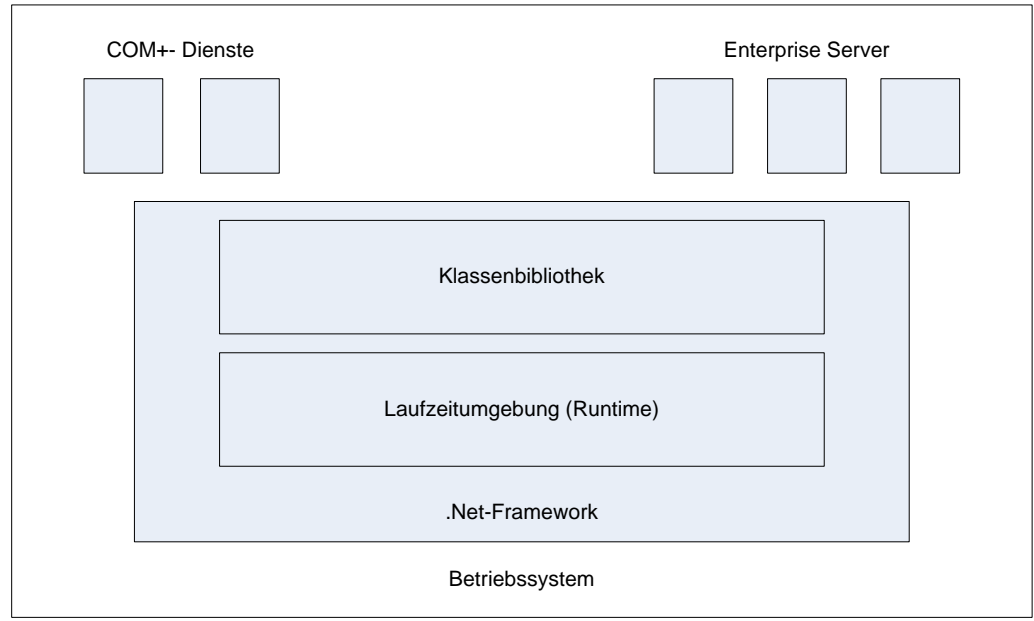
4-Tier Architektur



Beispiel einer typischen 4-Tier-Internetanwendung

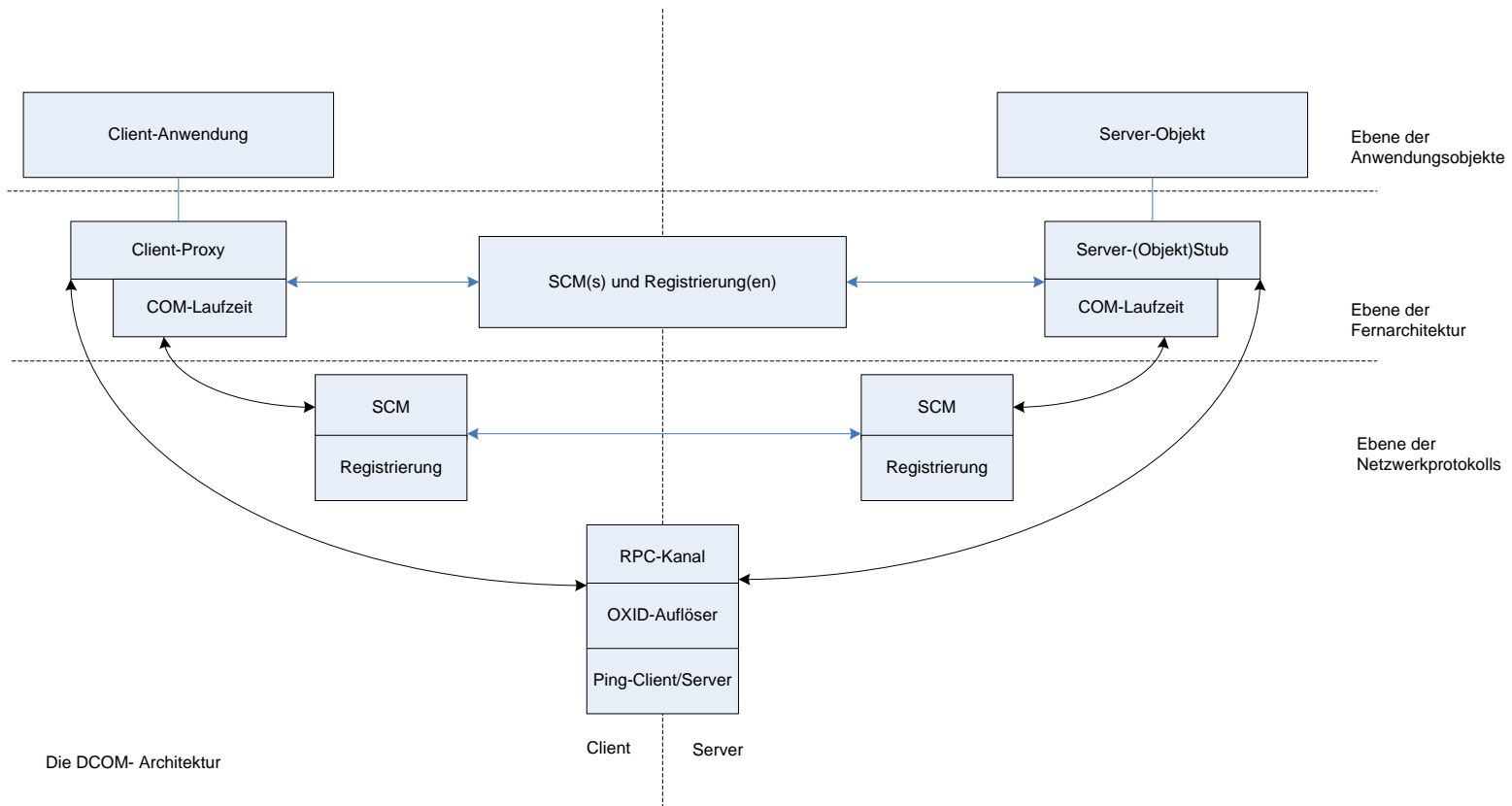


.Net Plattform



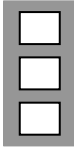
.Net-Plattform

DCOM und DCE RPCArchitektur



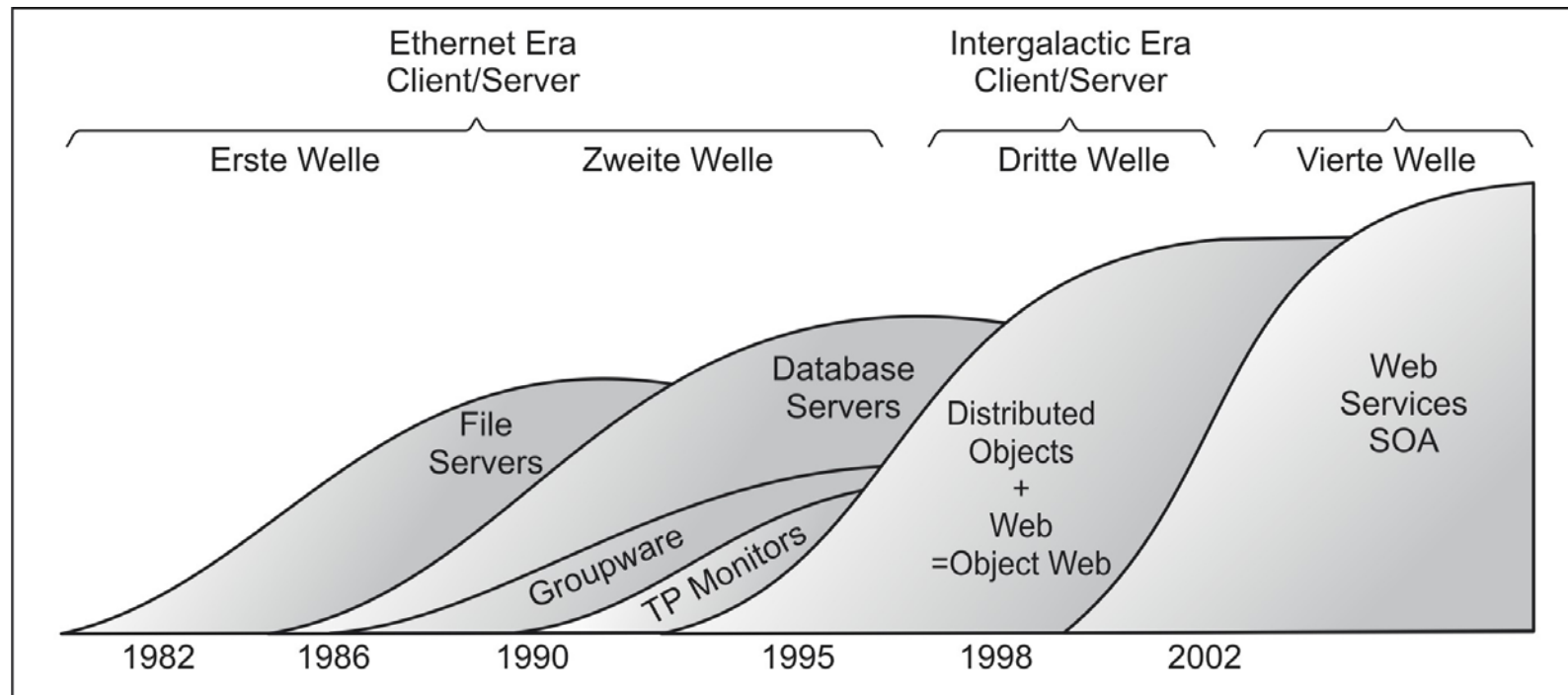
Die DCOM- Architektur

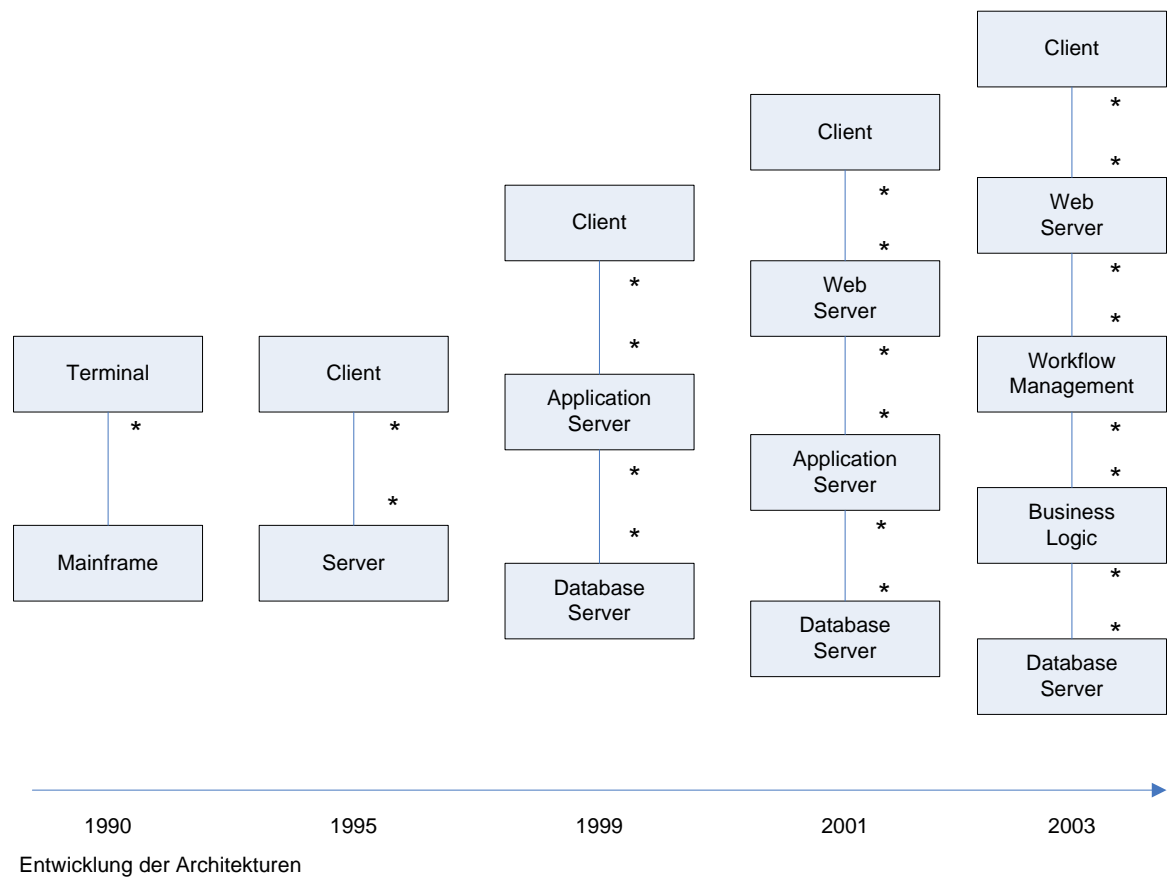
SCM – Service Control Manager



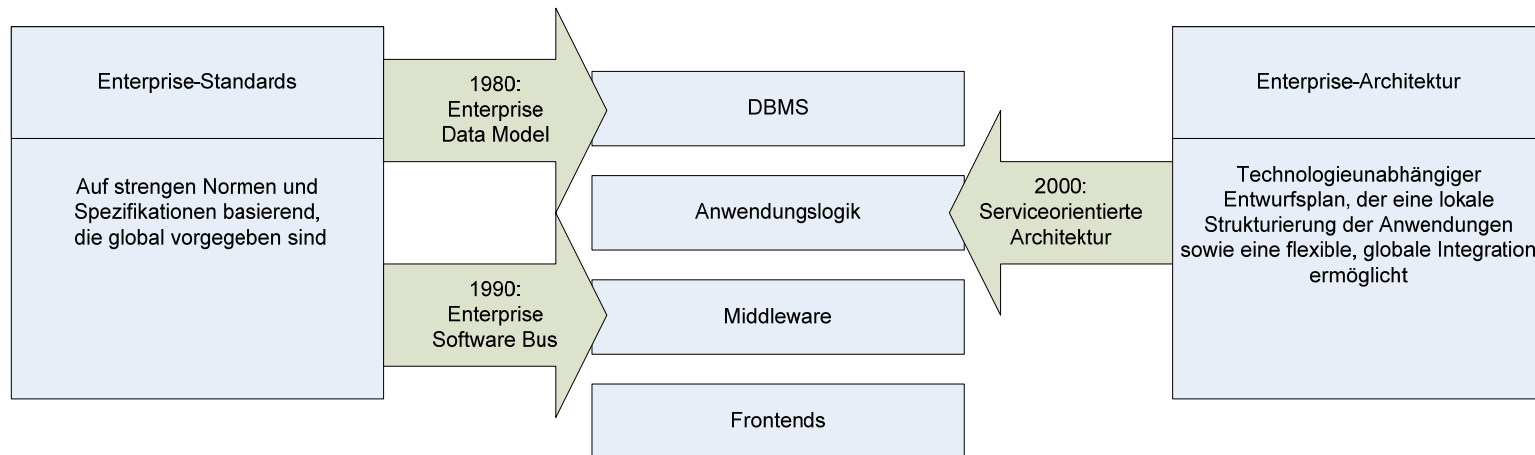
Zentrale Elemente heutiger SOA - Architekturen

Entwicklungszyklen



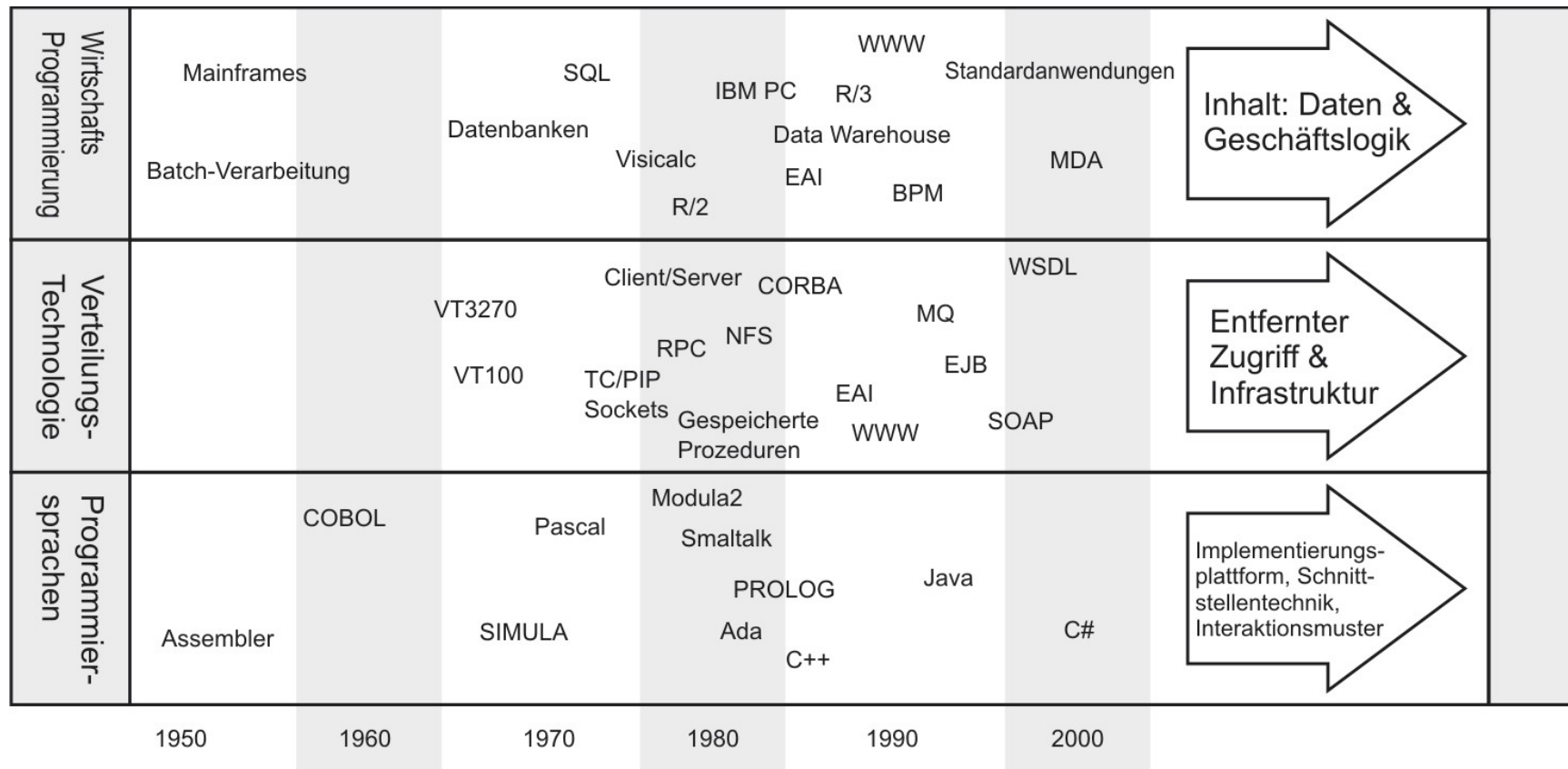


Ansätze 1980, 1990 und ab dem Jahr 2000



Enterprise-Datenmodelle und Software-Busse waren beliebte Ansätze für die Herausforderungen der Enterprise-Programmierung in den 80er und 90er Jahren.

Entwicklung – Inhalte, Technologie, Umgebungen





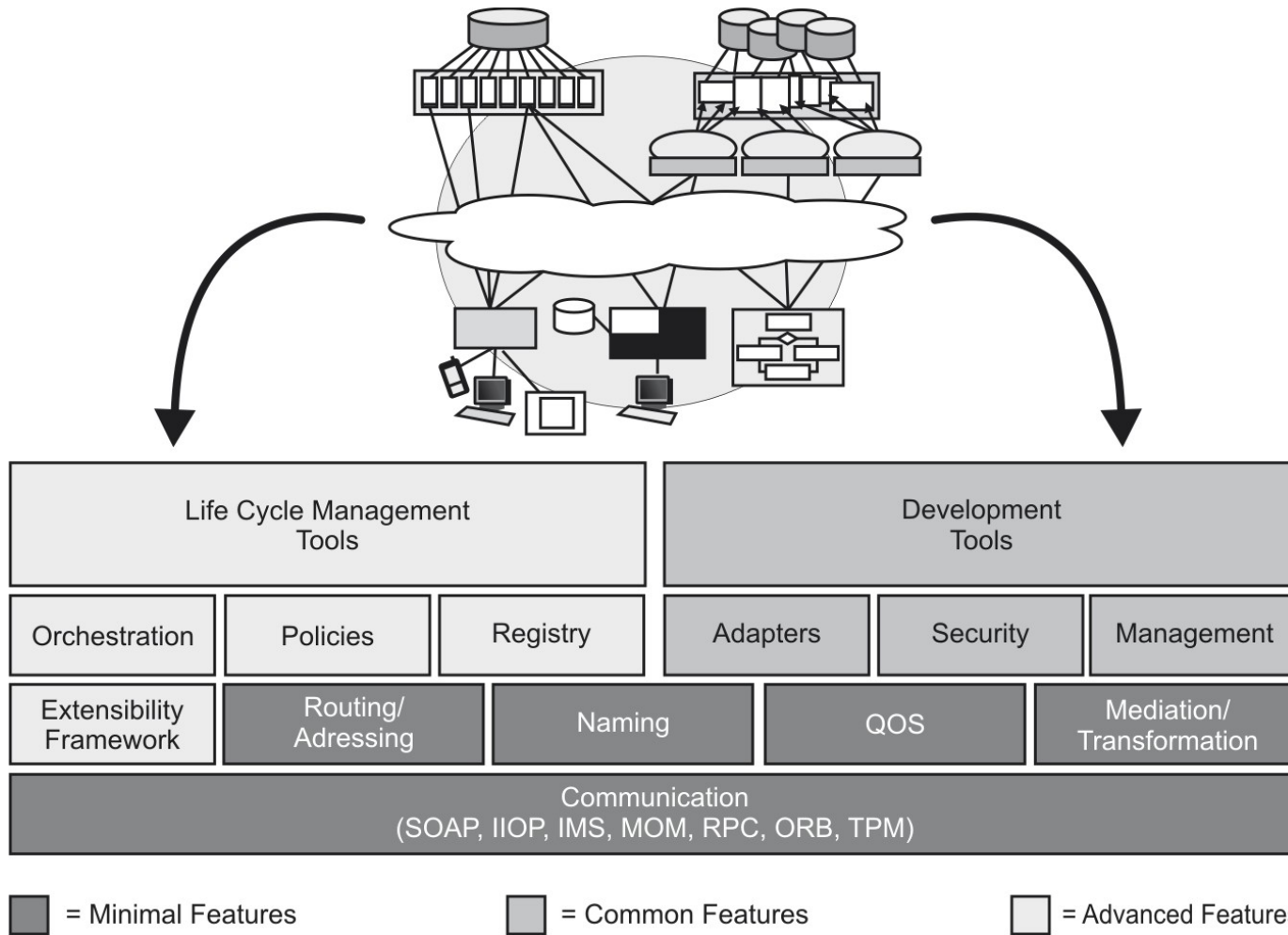
SOA - Serviceorientierte Architektur



SOA Einführung Geschäftsführungsebene

Die integrierte Betrachtung von Geschäftsprozessanpassungen und daran anschließende IT-Unterstützungsanpassung ist der Grund für die hohe Aufmerksamkeit, die diesem Thema auf Geschäftsführungsebene entgegengebracht wird.

SOA - Infrastruktur





Definition – SOA Service Orientierte Architektur

SOA ist (1) ein Managementkonzept und
(2) ein Systemarchitekturkonzept.

1. Das Managementkonzept strebt eine an den gewünschten Geschäftsprozessen ausgerichtete Infrastruktur an, die schnell auf veränderte Anforderungen im Geschäftsumfeld reagieren kann.
2. Das Systemarchitekturkonzept sieht die Bereitstellung fachlicher Dienste und Funktionalitäten in Form von Services vor.



Definition – Web Services

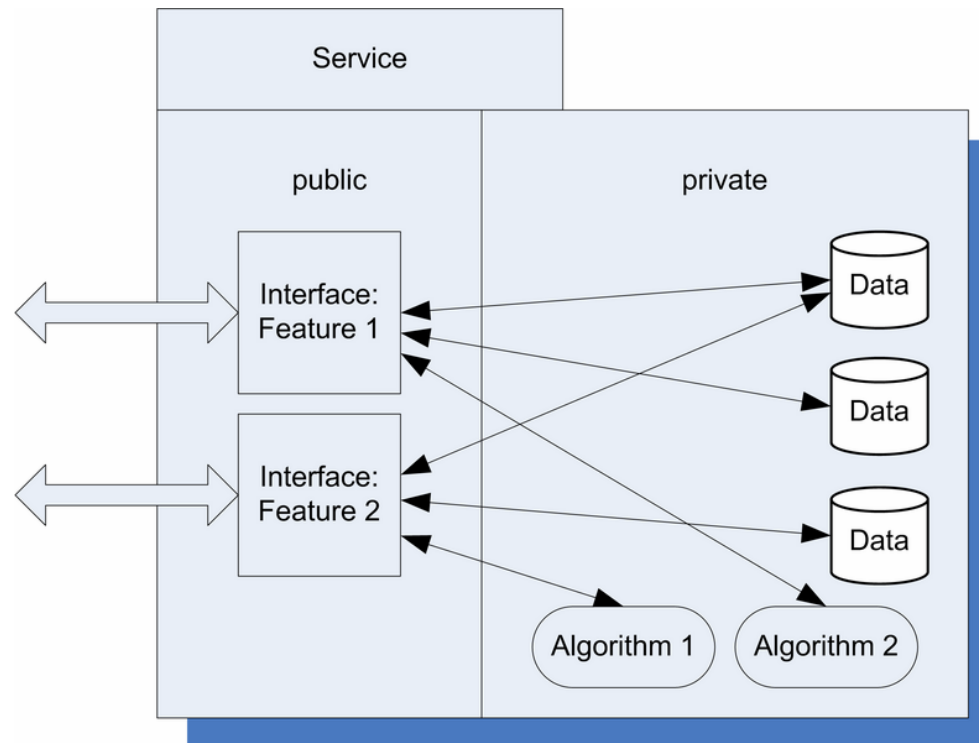
In SOA wird ein **Dienst** bzw. **Service** als eine Komponente bezeichnet, die eine wohl definierte Funktionalität über eine standardisierte Schnittstelle anderen Services oder Anwendungen zur Verfügung stellt.

Services können in unterschiedliche Typen eingeordnet werden:

- Basic services
- Intermediary services
- Process centric services
- Public enterprise services

Werden für die Identifizierung und Beschreibung eines Services sowie für die Datenübertragung von und zu diesem Web-Technologien und -Standards (z.B. XML, HTTP, WSDL, SOAP) benutzt, so spricht man auch von Web Services.

Definition – SOA Service/Dienst





Ziele - SOA

- Primärziel ist:
 - die historisch gewachsene, heterogene Systemlandschaft effizient an Änderungen im Geschäftsprozess anpassen zu können.
 - Software erstellt die:
 - einfach an neue Bedürfnisse angepasst werden kann (Flexibilität)
 - wieder verwendbar ist
 - verteilt installiert werden kann
 - an Geschäftsprozesse angepasst ist.

- Sekundärziele sind:
 - Kostenvorteile durch schnelle Optimierung
 - höhere Produktivität der Softwareentwickler durch Wiederverwendung von Services
 - schnelle Reaktion auf Herausforderungen möglich
 - mittelfristige Einsparungen
 - Reduzierung der Komplexität durch Aufbrechen monolithischer IT-Systeme
 - schrittweise Restrukturierung komplexer Anwendungssysteme.



Elementarer Grundgedanke

- Identifikation von Services
 - 'service consumer' und 'service provider',

- Zuständigkeiten
 - Trennung der Zuständigkeiten nach fachlichen Gesichtspunkten

- Datenkapselung
 - ein Service ist ausschließlich zuständig für „seine Daten“, jeder 'service consumer' muss sein Interesse an den Daten bzw. an dem Service beim 'service provider' anmelden.



Kurzübersicht – SOA I

- Anwendungssysteme
 - zur durchgängigen Unterstützung von Geschäftsprozessen lassen sich durch Aneinanderreihung von Serviceaufrufen realisieren.
- Programmlogik
 - Die Programmlogik ist nicht in einem einzigen Programm zu finden, sondern verteilt über mehrere unabhängige Dienste.



Kurzübersicht – SOA II

- Dienste
 - Die SOA sieht eine Menge voneinander unabhängiger, lose gekoppelter Dienste vor.
 - Ein Dienst wird von einem *service provider* angeboten. Ein *service consumer* stellt eine Anfrage (*service request*) an einen Dienst und bekommt daraufhin eine Antwort (*service response*) vom Anbieter.



Kurzübersicht – SOA III

- SOAP, WSDL und UDDI
 - Oft werden für SOAs Web Services auf Basis der wenigen bestehenden Standards wie SOAP, WSDL und UDDI eingesetzt.

- CORBA, DCOM, WCF oder EJB
 - Doch kann eine SOA prinzipiell auf jeder dienstbasierten Technologie wie zum Beispiel CORBA, DCOM, Windows Communication Foundation (WCF) oder Enterprise Java Beans (EJB) aufgebaut werden.

- Programmiersprachen und Systemplattformen
 - Da Services in unterschiedlichen Programmiersprachen und auf unterschiedlichen Systemplattformen realisiert werden können, wird eine SOA häufig zur Anwendungsintegration genutzt.



Kurzübersicht – SOA IV

- Kapselung von Persistenz
 - Kapselung von persistenten Daten durch Dienste, die exklusives Lese- und Schreibrecht auf „ihre“ Daten besitzen.
 - Modularität, Flexibilität und Betriebskosten: Modularität führt zu geringen Redundanzen und einer höheren Flexibilität der IT-Systeme, was häufig zu niedrigeren Betriebskosten führt.



Kurzübersicht – SOA V

- Vorteile
 - Modularität, Flexibilität und Betriebskosten:
- Nachteile
 - erheblicher initialer Entwicklungsaufwand
 - die Geschäftslogik bereits existierender Unternehmensanwendungen werden durch die Entwicklung geeigneter Adapter zu Services umgebaut werden.



Umfeld

Der Begriff serviceorientierte Architektur ist in das folgende Umfeld einzuordnen

- **Geschäftsprozessmanagement (GPM)**
 - Die Definition der Prozesse des Business, die durch die IT unterstützt werden.
- **Business Service Management (BSM)**
 - Die Verbindung zwischen Prozessmanagement und ITSM.
- **IT-Service-Management (ITSM)**
 - Methoden, die nötig sind, um die bestmögliche Unterstützung von Geschäftsprozessen (GP) durch die IT-Organisation zu erreichen.
 - Der hier bekannte de-facto Standard ist die IT Infrastructure Library (ITIL).



Vorgehen I

5 Einstiegspunkte in die Arbeit mit SOA (von IBM):

1. **Mitarbeiter** – Steigerung der Produktivität durch Zusammenarbeit und Interaktion mit Anwendungs- und Informationsservices, die die Geschäftsprozesse unterstützen.
2. **Prozesse** – Werkzeuge und Services zur Optimierung des Managements von Geschäftsprozessen
3. **Informationen** – Bereitstellung des Zugriffs auf komplexe, heterogene Datenquellen innerhalb des Unternehmens
4. **Konnektivität** – Verknüpfung von Mitarbeitern, Prozessen und Informationen im Unternehmen mit einem nahtlosen Nachrichten- und Informationsfluss, und das unabhängig vom Standort und Zeitpunkt sowie unter Verwendung beliebiger Werkzeuge.
5. **Wiederverwendung** – kontinuierlichen Nutzen bereits getätigten Investitionen



Vorgehen II

Hat man sich erst einmal für einen Einstiegspunkt entschieden, hat sich das folgende Vorgehen als günstig erwiesen (IBM):

- Think BIG, Start small (denke groß, fange klein an)
- Fachabteilung einbinden
- Bestandsaufnahme
- Erste Services einbinden
- Registry installieren
- Governance regeln
- Sicherheit planen
- Messaging Infrastructure einrichten
- Service Management einrichten
- Services orchestrieren



Anwendungsbeispiel

Online-Brokerage

der Abschluss von Wertpapiergeschäften über das Internet –
kann in einer SOA auf folgenden Diensten aufsetzen

- Wertpapierdienst, der handelbare Wertpapiere (Aktien, Fonds, ...) nennt
- Marktdatendienst, der zu einem Wertpapier aktuelle Börsenkurse liefert
- Depotdienst, der Zugriff auf das Wertpapierdepot des Anwenders erlaubt
- Orderdienst, der Kauf-/Verkaufsaufträge des Anwenders an eine Börse übermittelt
- Archivdienst, der alle Daten der Transaktion „revisionssicher archiviert“

Diese Dienste werden aus einer Webanwendung heraus aufgerufen.



Exkurs



SOAP - Geschichtlicher Abriss

- Ursprüngliche Entwicklung durch Dave Winer und Microsoft. Erstes Produkt - XML-RPC.
- 1999: SOAP V1.0 erste Veröffentlichung (mit der Unterstützung durch Don Box)
- 2000: SOAP V1.1 IBM, Microsoft, DevelopMentor (Don Box) und UserLand Software (Dave Winer) die Spezifikation von SOAP 1.1 beim World Wide Web Consortium (W3C) einreichen.
- 2003: SOAP V1.2, wird als Empfehlung (engl. *recommendation*) anerkannt.



SOAP

Die **Abkürzung** SOAP stand ursprünglich für „Simple Object Access Protocol“ ein plattformunabhängiges Kommunikationsprotokoll bei Web Services

SOAP ist ein **leichtgewichtiges Protokoll** zum Austausch XML-basierter Nachrichten

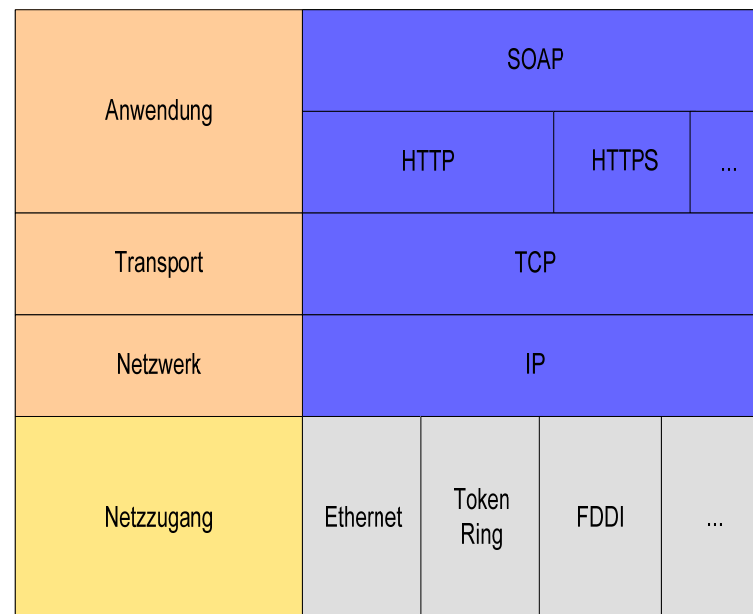
Zum Senden von Nachrichten können **beliebige Transportprotokolle** verwendet werden, beispielsweise FTP, SMTP oder HTTP(S).

Die Kommunikation mit SOAP ermöglicht die Kopplung von Systemen, der offene Entwurf von SOAP ermöglicht jedoch lediglich den Aufbau **schwach gekoppelter** Systeme.

Die Flexibilität des Konzeptes wird durch **Nachteile** in Übertragungsvolumen und Rechenaufwand erkauft

SOAP – ein Netzwerkprotokoll

- Datenaustausch zwischen Systemen über RPC und HTTP
- SOAP stützt sich auf die Dienste anderer Standards:
 - XML zur Repräsentation der Daten und
 - Internet-Protokolle zur Übertragung der Nachrichten.
- Gängigste Kombination ist SOAP über HTTP und TCP.





Definition – XML **Extensible Markup Language**

XML

- dt.: Erweiterbare Auszeichnungssprache
- engl.: Extensible Markup Language
- Auszeichnungssprache zur Darstellung hierarchisch strukturierter Daten in Form von Textdateien.
- Austausch von Daten zwischen unterschiedlichen IT-Systemen, speziell über das Internet.



Auszeichnungssprache

- Auszeichnungssprachen besitzen eine
 - Syntax
 - Grammatik
 - Semantik.

- „Tags“ als Markierungselement.

- Descriptive Markup Languages
 - SGML, HTML, MathML, WML, XAML oder SVG.

- Procedural Markup Languages
 - TeX, PDF und PostScript.

- Programmiersprachen
 - Abgrenzung ist fließend.
 - PostScript beispielsweise ist auch eine Programmiersprache (Drucker- und Bildschirmweisungen).

SGML - Standard Generalized Markup Language, dt. **Normierte Verallgemeinerte Auszeichnungssprache**

Definition – ML Auszeichnungssprache

(ML - Markup Language)

Vergleichstabelle

Beispiel für ...	HTML	LaTeX	Wikitext
... Überschrift	<code><h1>&Uuml;berschrift</h1></code>	<code>\section{Überschrift}</code>	<code>== Überschrift ==</code>
... Aufzählung	<code> Punkt 1 Punkt 2 Punkt 3 </code>	<code>\begin{itemize} \item Punkt 1 \item Punkt 2 \item Punkt 3 \end{itemize}</code>	<code>* Punkt 1 * Punkt 2 * Punkt 3</code>
... fetten Text	<code>fett</code>	<code>\textbf{fett}</code>	<code>''' fett '''</code>
... kursiven Text	<code><i>kursiv</i></code>	<code>\textit{kursiv}</code>	<code>'' kursiv ''</code>



Beispiel einer XML-Datei

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<enzyklopädie>
  <titel>Wikipedia Städteverzeichnis</titel>
<eintrag>
  <stichwort>Genf</stichwort>
  <eintragstext>Genf ist der Sitz von ...</eintragstext>
</eintrag>
<eintrag>
  <stichwort>Köln</stichwort>
  <eintragstext>Köln ist eine Stadt, die ...</eintragstext>
</eintrag>
</enzyklopädie>
```

WSDL

Web Services Description Language

definiert eine plattform-, programmiersprachen- und protokollunabhängige XML-Spezifikation zur Beschreibung von Netzwerkdiensten (Web Services) zum Austausch von Nachrichten.



BPEL - Business Process Execution Language

- BPEL: XML-basierte Sprache zur Beschreibung von Geschäftsprozessen, deren einzelne Aktivitäten durch Webservices implementiert sind.
- BPEL: Die im Jahr 2002 von IBM, BEA und Microsoft eingeführte Sprache wird dabei zur Beschreibung von so genannten Webservice-Orchestrierungen verwendet.
- BPEL: Die Beschreibung selbst wird ebenfalls in Form eines Webservice bereitgestellt und kann als ein solcher verwendet werden.



Definition – UDDI Verzeichnisdienst

▪ UDDI

- **Universal Description, Discovery and Integration**
- Verzeichnisdienst für dynamischen Web Services
- SOAP-Schnittstelle mit Unternehmen, ihre Daten und ihre Services.
- drei Arten von Informationen

▪ White Pages

- Namensregister, sortiert nach Namen
- Auflistung der Anbieter mit allen Detailangaben
- Kontaktinformationen (Telefon, Telefax,...)

▪ Yellow Pages

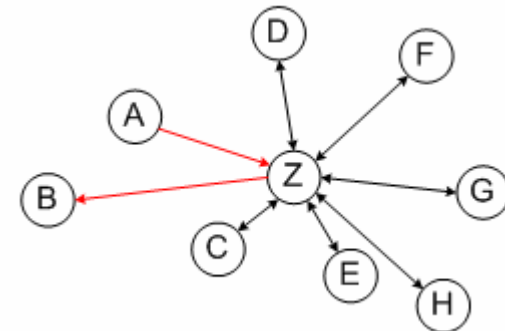
- Branchenverzeichnis
- Spezifische Suche gemäß verschiedener Taxonomien (Ort, Dienstart,...)
- Verweist auf White Pages
- Klassifiziert die Services anhand internationaler Standards wie UNSPSC

▪ Green Pages

- Informationen über das Geschäftsmodell des Unternehmens
- Technische Details zu den angebotenen Web Services
- Auskunft über Geschäftsprozesse

Definition EAI erste Ansätze

Im Vergleich zu früheren Ansätzen wie EAI, ist der SOA-Ansatz verbunden mit einem Umdenken.



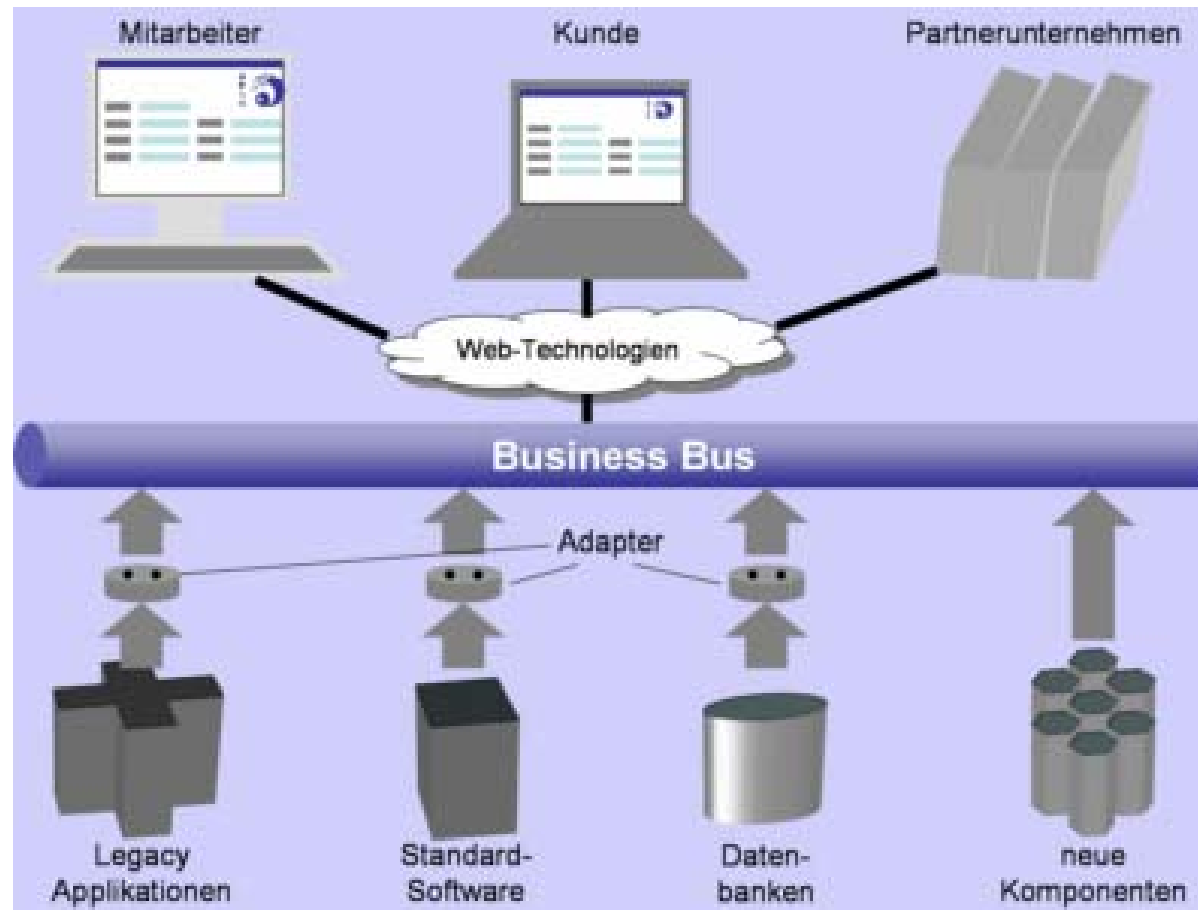
War EAI noch technisch in einem Hub-und-Spoke-System aufgebaut, so ist SOA ein fachlich getriebener, dezentraler, service-getriebener Ansatz.



EAI – Enterprise Application Integration

- Adaptern
 - keine Veränderung der bestehenden Services
 - Einsatz von Adaptern (keine Funktionsintegration oder der Datenintegration)
 - Alle funktionalen Schnittstellen werden mittels Adaptern (Schnittstellenumsetzer) abstrahiert.
- E-Business und Portale
 - EAI findet praktisch in allen Bereichen der Prozessintegration insbesondere im E-Business und in Portalen Anwendung.
- Großunternehmen
 - Darüber hinaus ersetzt EAI in vielen großen Unternehmen mit komplexen IT-Landschaften klassische Middleware-Produkte und wird zu einem wichtigen IT-Architektur-Element.

EAI – Systembeispiel





Entwicklungsumgebungen für SOA - Architekturen



SOAP - Implementierungen

- Apache SOAP, Apache Axis (J2EE)
- Microsoft .NET (.NET)
- AWS (ada)
- PEAR SOAP-Projekt (PHP)
- RemObjects SDK
- cSOAP und gSOAP für C/C++
- SOAP für Delphi
- SOAP für Perl
- SOAP für PHP5
- SOAP für Python
- SOAP4R für Ruby
- SOAP für Tcl
- SOAP für VisualWorks Smalltalk
- WebMethods GLUE
- Codehaus XFire (Java)
- kSOAP für Java inklusive Java ME

Web Applikations-Entwicklung: Platform Stacks

Platform Name	Java/J2EE	.NET/Visual C#	LAMP (Linux/Apache/MySQL/PHP)
IDE/Development Layer	Eclipse, NetBeans (evtl. IDEA, Jbuilder)	Visual Studio .NET	Any (Eclipse stark verbreitet)
Framework Layer	J2EE, Spring, etc...	.NET Platform	PHP, Ruby on Rails
Web Layer	JSP (Java Server Pages), JSF (Java Server Faces)	ASP.NET	PHP, Ruby on Rails
Application Server Layer	Tomcat/Apache, Tomcat, Websphere, ...	IIS Internet Information Server	Apache
Language Layer	Java VM (Platform independent)	C# CLR	Nicht anwendbar (Scriptsprachen)

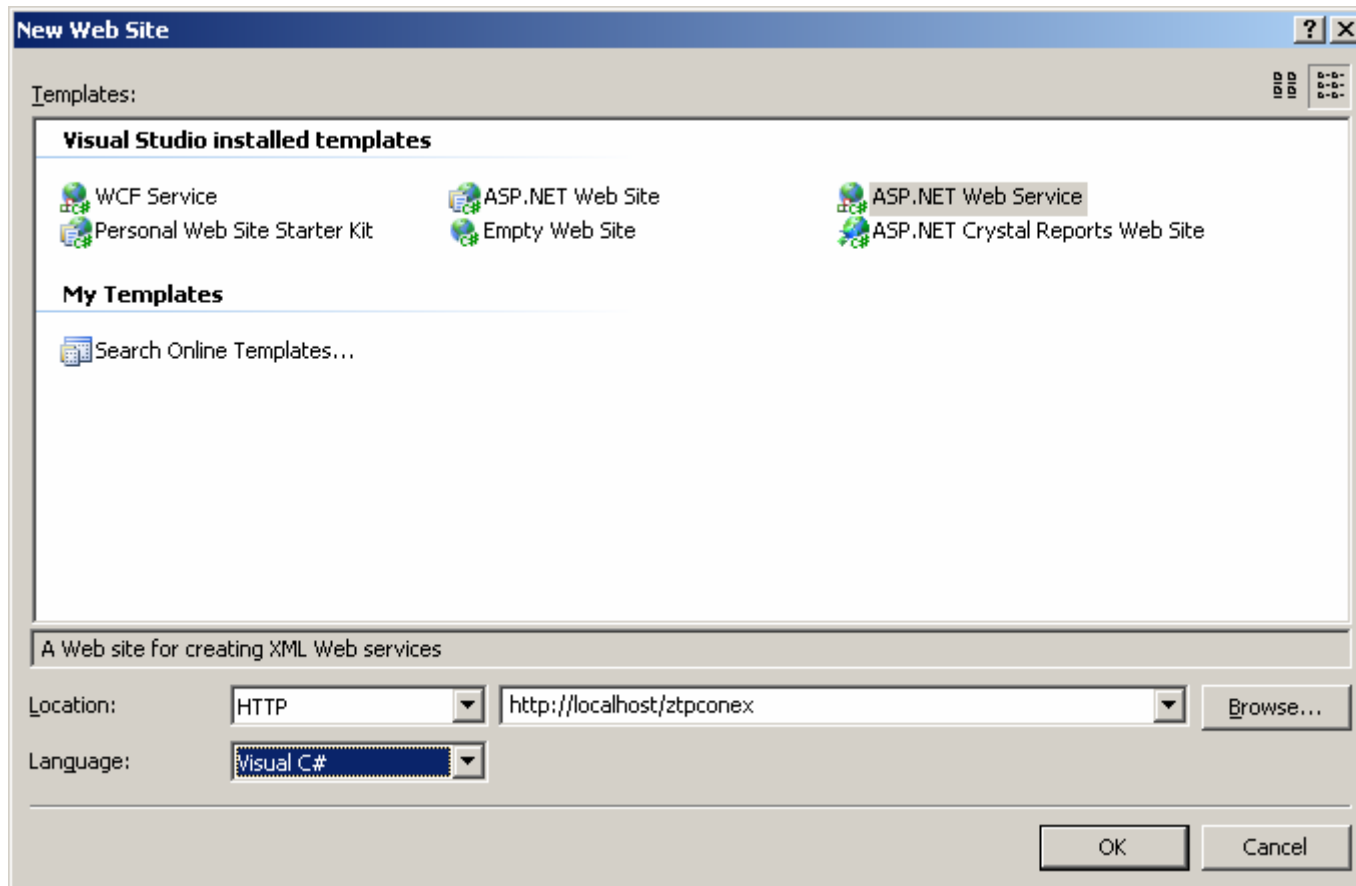
(c) DI Hager Michael



Web Service



Web-Service Programmierung I



Web-Service Programmierung II

The screenshot displays the Microsoft Visual Studio IDE with the following components:

- Code Editor:** Shows the implementation of a web service in `App_Code/Service.cs`. The code defines a `ZTPContact` struct and a `GetZTPContact` web method.
- Output Window:** Shows the build output, indicating a successful build.
- Solution Explorer:** Shows the project structure for `http://localhost/ztpconex/`, including `App_Code`, `App_Data`, and `ztp.asmx`.

```
public struct ZTPContact
{
    public String Name;
    public String EMailAddress;
    public String PhoneNumber;
}

public Service () {

    //Uncomment the following line if using designed components
    //InitializeComponent();
}

[WebMethod]
public ZTPContact GetZTPContact (String Name, String EMailAddress, String Phone) {
    ZTPContact ztpc = new ZTPContact();
    ztpc.Name = "Bakar Azur";
    ztpc.EMailAddress = "bakar@zt-prentner-it.at";
    ztpc.PhoneNumber = "+43 (1) 5324686-17";
    return ztpc;
}
```

Output:

```
Validating Web Site
Building directory '/ztpconex/App_Code/'.
Building directory '/ztpconex/'.

Validation Complete
***** Build: 1 succeeded or up-to-date, 0 failed, 0 skipped *****
```

Build succeeded

Web-Service Programmierung III

Service Web Service - Windows Internet Explorer

http://localhost/ztpconex/ztp.asmx

File Edit View Favorites Tools Help

Service

The following operations are supported. For a formal definition, please review the [Service Description](#).

- [GetZIPContact](#)

This web service is using <http://tempuri.org/> as its default namespace.

Recommendation: Change the default namespace before the XML Web service is made public.

Each XML Web service needs a unique namespace in order for client applications to distinguish it from other services on the Web. <http://tempuri.org/> is available for XML Web services that are under development, but published XML Web services should use a more permanent namespace.

Your XML Web service should be identified by a namespace that you control. For example, you can use your company's Internet domain name as part of the namespace. Although many XML Web service namespaces look like URLs, they need not point to actual resources on the Web. (XML Web service namespaces are URIs.)

For XML Web services creating using ASP.NET, the default namespace can be changed using the `WebService` attribute's `Namespace` property. The `WebService` attribute is an attribute applied to the class that contains the XML Web service methods. Below is a code example that sets the namespace to "`http://microsoft.com/webservices/`":

C#

```
[WebService(Namespace="http://microsoft.com/webservices/")]
public class MyWebService {
    // implementation
}
```

Visual Basic

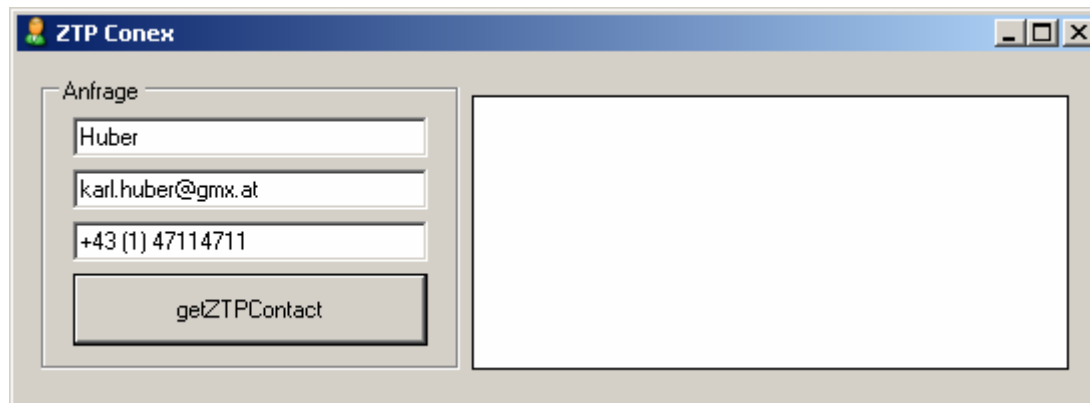
```
<WebService(Namespace:="http://microsoft.com/webservices/")> Public Class MyWebService
    ' implementation
End Class
```

C++

```
[WebService(Namespace="http://microsoft.com/webservices/")]
public ref class MyWebService {
    // implementation
}
```

Local intranet 100%

Web-Service Programmierung IV



ZTP Conex

Anfrage

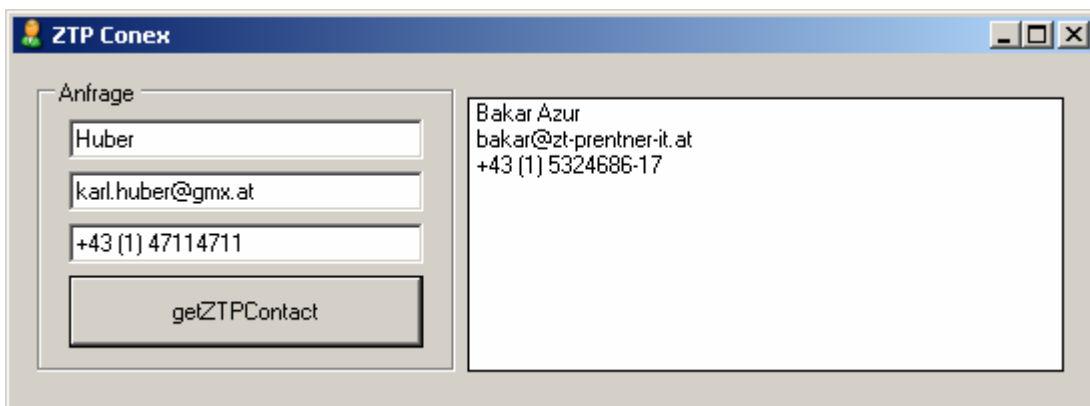
Huber

karl.huber@gmx.at

+43 (1) 47114711

getZTPContact

This screenshot shows the ZTP Conex web service interface. It features a title bar with the text 'ZTP Conex' and standard window controls. Below the title bar, there is a section labeled 'Anfrage' (Request). This section contains three text input fields: the first contains 'Huber', the second contains 'karl.huber@gmx.at', and the third contains '+43 (1) 47114711'. Below these fields is a button labeled 'getZTPContact'. To the right of the input fields is a large, empty rectangular area, likely intended for displaying the results of the service call.



ZTP Conex

Anfrage

Huber

karl.huber@gmx.at

+43 (1) 47114711

getZTPContact

Bakar Azur
bakar@zt-prentner-it.at
+43 (1) 5324686-17

This screenshot shows the ZTP Conex web service interface after a successful service call. The input fields and the 'getZTPContact' button are visible on the left. The large rectangular area on the right now displays the returned data in a text format: 'Bakar Azur', 'bakar@zt-prentner-it.at', and '+43 (1) 5324686-17'.



Fallbeispiele

Österreichweite - Service Orientierte (verteilte)
Software Lösungen

- Raiffeisen ELBA Internet
- Bundesrechenzentrum
- Österreichische Notariatskammer
- Apothekerkammer



SOA, EAI, Web Services – Was ist der richtige Ansatz für eine prozessorientierte IT-Architektur?



Zusammenfassung

SOA(P) - Alter Hut neu verpackt (1997-2007)?

Basistechnologien und –konzepte bestehen zum Teil noch, wurden aber durch neue innovative Protokolle, Paradigmen und Lösungen ergänzt.

Der Weg geht von der technologiegetriebenen IT hin zu flexiblen Business Process und Service Orientierter Lösungen.

Entwicklungskosten/-aufwände sind nur bedingt zurück gegangen.

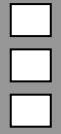




Besten Dank für Ihre Aufmerksamkeit.

Haben Sie noch Fragen?





Akronymitis

- EAI - Enterprise Application Integration
- BAM - Business Activity Monitoring
- BPM - Business Process Management
- BRM - Business Rules Management
- EAM - Enterprise Architecture Management
- EDA - Event Driven Architecture
- RPC – Remote Procedure Call
- SOA - Service Oriented Architecture
- SOAP – Simple Object Access Protocol



Auf SOAP basierende Erweiterungen

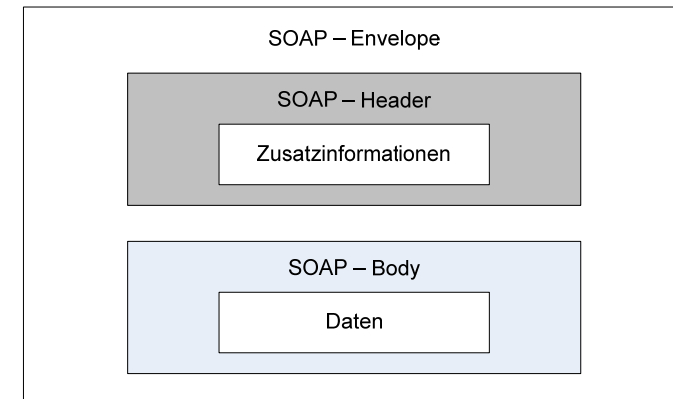
- **WS-Reliability (Web Services Reliability):**
 - Sicherheitsmechanismen, um z. B. Transaktionen verlässlich abwickeln zu können
- **WS-Security (Web Services Security):**
 - Sicherstellen von Integrität und Vertraulichkeit von Nachrichten
- **WSRP (Web Services for Remote Portlets):**
 - Integration von Präsentationslogik in Portale

Aufbau von SOAP-Nachrichten

- Minimale SOAP-Nachricht besteht aus einem
 - Envelope mit Namespace-Attributen
 - Header-Element (optional) mit Meta-Informationen, beispielsweise zum Routing, zur Verschlüsselung oder zu Transaktionsidentifizierung, untergebracht werden.
 - Body-Element: Nutzdaten.

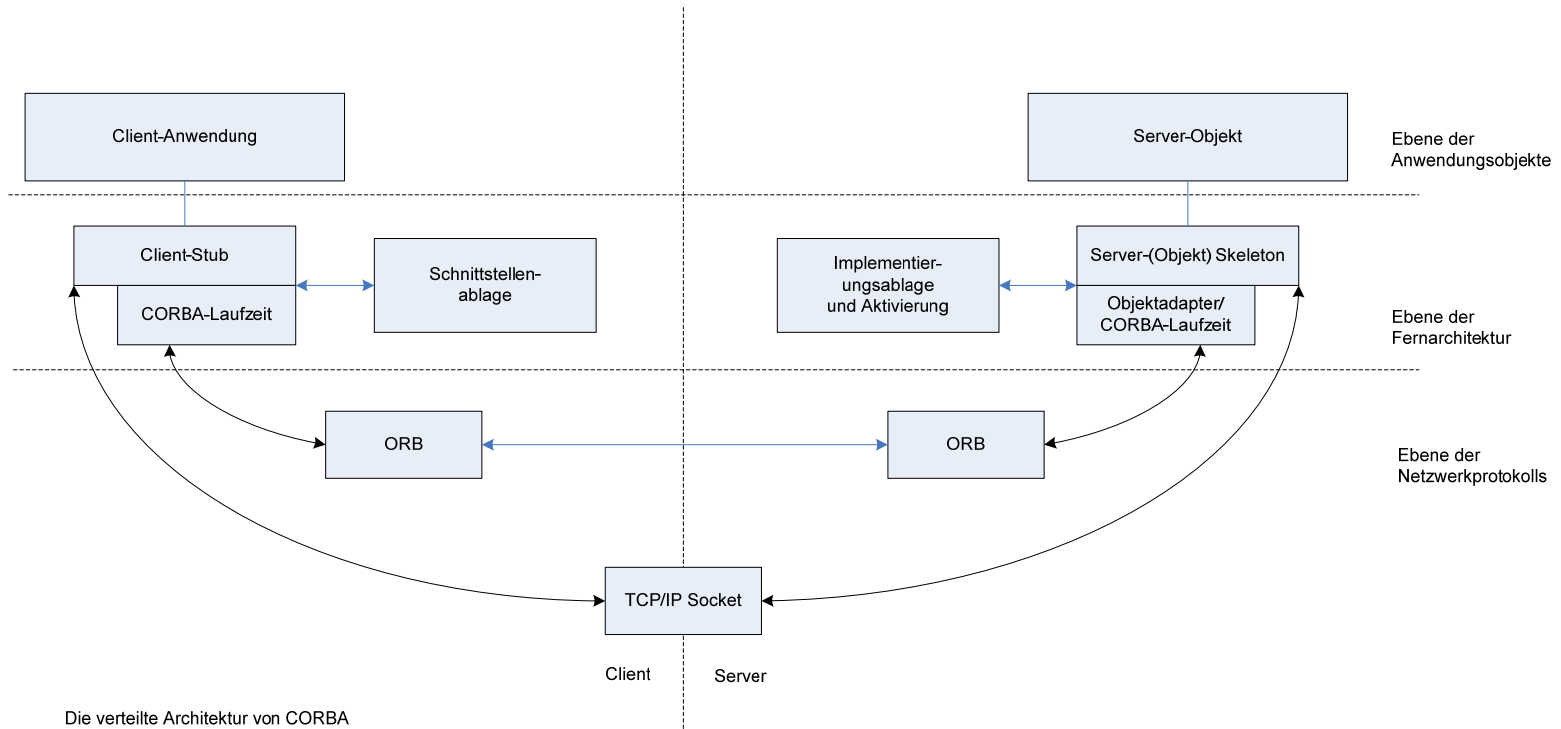
```
<?xml version="1.0"?>
<s:Envelope xmlns:s="http://www.w3.org/2001/12/soap-envelope">
  <s:Header> <m:RequestID xmlns:m="http://www.lecture-
db.de/soap">a3f5c109b</m:RequestID> </s:Header>
  <s:Body> </s:Body>
</s:Envelope>
```

- Body-Elements
 - Informationen zum Datenaustausch, als auch
 - Anweisungen für einen entfernten Prozeduraufruf stehen
- Datenbankabfrage
 - über eine Internet-Schnittstelle z.B. für Google, eBay oder auch Amazon

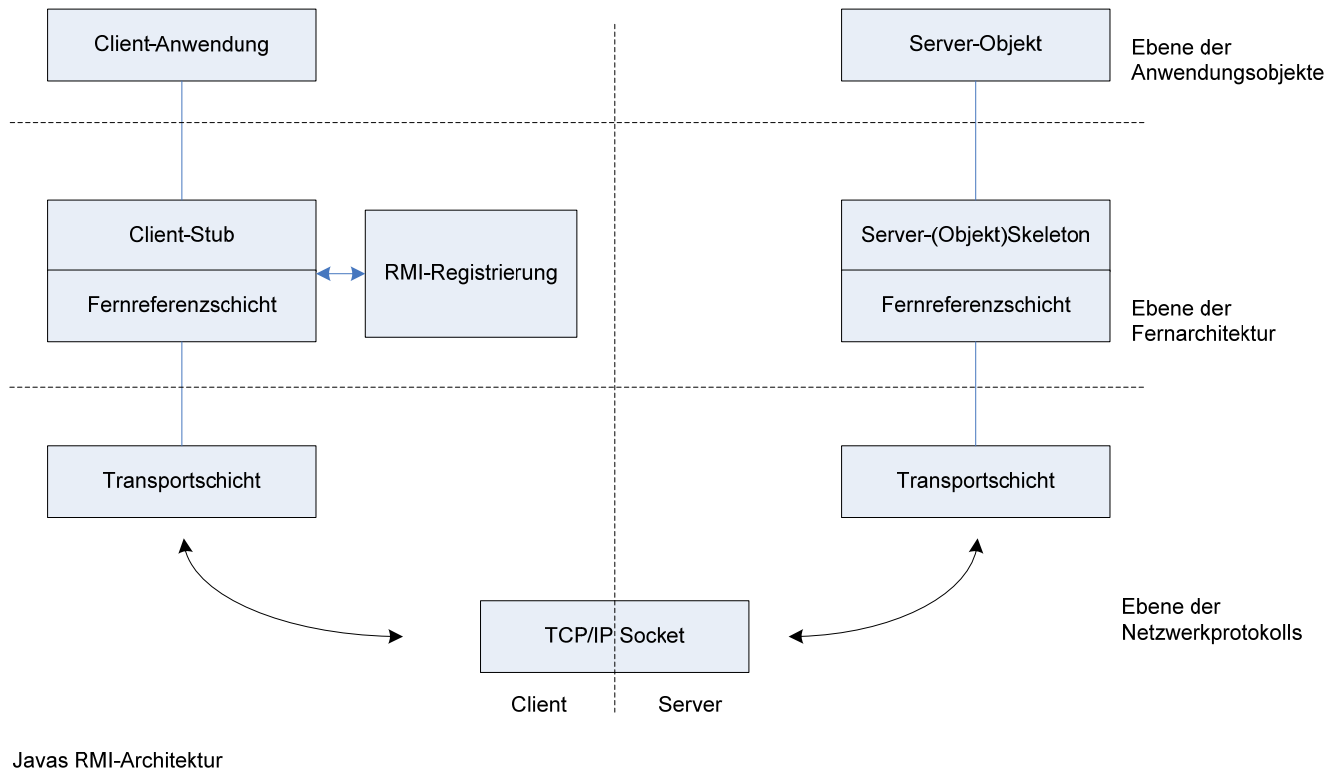


Aufbau einer Soap-Nachricht

CORBA Architektur



Java RMI-Architektur





Quellen

- i'X Computer-Zeitschrift, 2007 heise.de;
- Java and Server, R. Orfali & Dan Harkey, Wiley Verlag, 1997;
- SOAP developer's guide, K. Scribner, M. Stiver; Markt+Technik 2001;
- www.wikipedia.de (SOA, SOAP, EAI)
- Verteilte Systeme und Anwendungen, Ulrike Hammerschall, 2005;
- Enterprise SOA, D. Krafzig, K. Banke, D. Slama, mitp 2007
- EAI – S. Conrad, R. Tritsch, Spektrum Verlag, 2006;