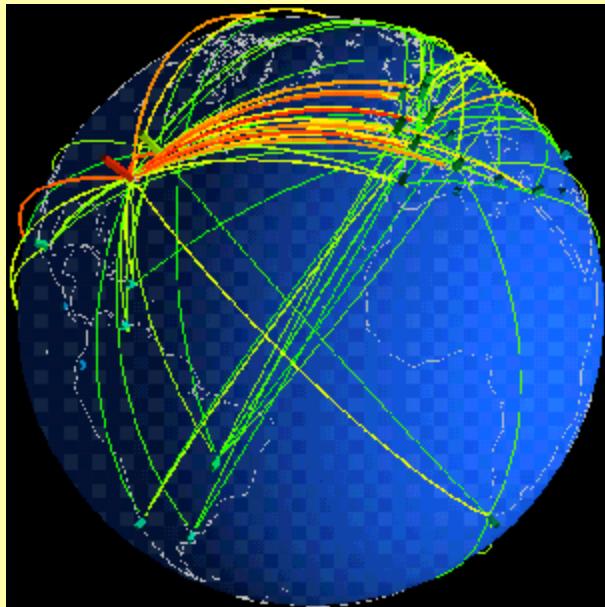


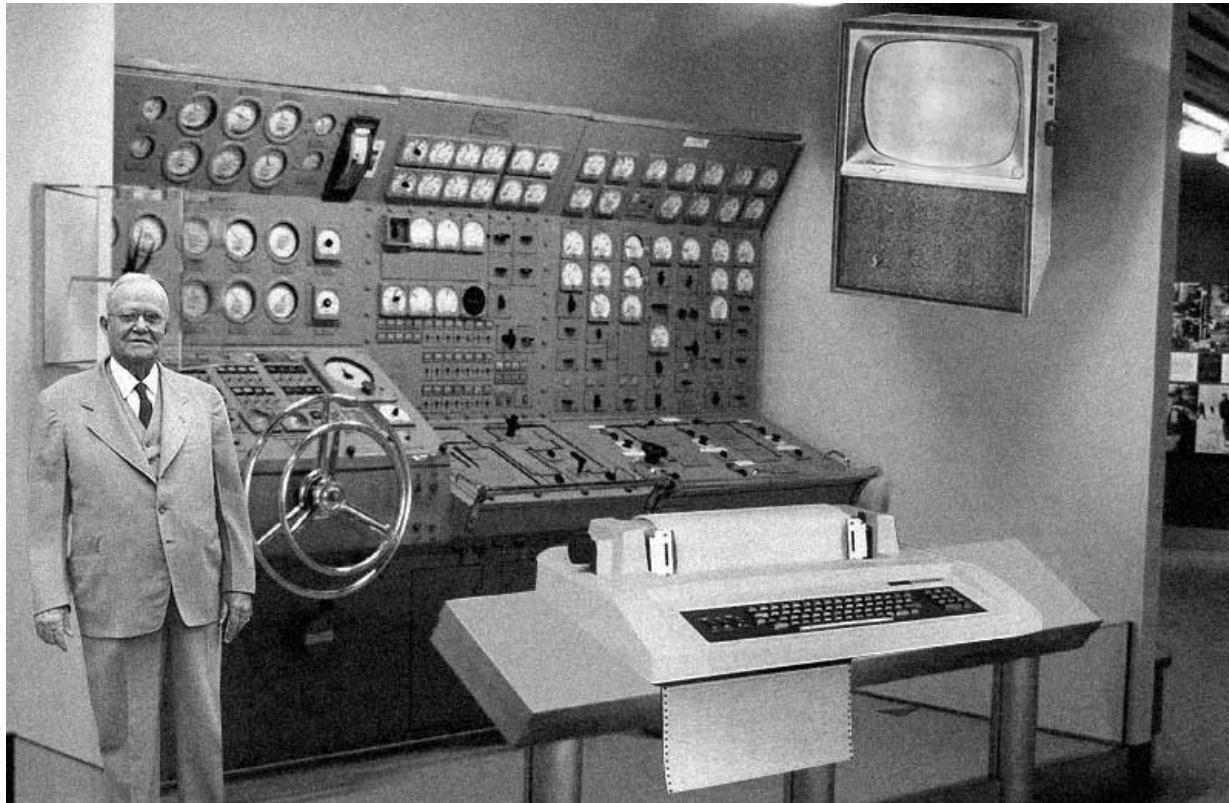
Web Engineering

Einleitung



Prof. Dr. Andreas Judt

Visionen früher und heute: Home Computer im Jahr 2004



Scientists from the RAND Corporation have created this model to illustrate how a "home computer" could look like in the year 2004. However the needed technology will not be economically feasible for the average home. Also the scientists readily admit that the computer will require not yet invented technology to actually work, but 50 years from now scientific progress is expected to solve these problems. With teletype interface and the Fortran language, the computer will be easy to use and only

Berühmte Irrtümer über Computer

- 64 kByte Arbeitsspeicher sollten jedem genügen.
 - Bill Gates, Gründer von Microsoft, 1981
- Es gibt keinen Grund, warum irgend jemand in der Zukunft bei sich zu Hause einen Computer haben sollte.
 - Ken Olsen, Gründer und Vorstand Digital Equipment, 1977
- Computer der Zukunft werden vielleicht nicht mehr als 1,5 to wiegen.
 - Zeitschrift Popular Mechanics, 1949
- Ich denke, dass es einen Weltmarkt für vielleicht fünf Computer gibt.
 - Thomas Watson, Vorstandsvorsitzender IBM, 1943
- Es gibt nichts Neues mehr. Alles was man erfinden kann, ist schon erfunden worden.
 - Charles H. Duell, US-Patentamt, 1899

Fazit

Computer und Internet entwickeln
sich mit unbekannter Geschwindigkeit
in eine unbekannte Richtung!

Was ist das Internet eigentlich?

- Überall verwendetes Netzwerk für multimediale Daten
 - Bilder, Video, Telefon, Daten
- Das Internet ist vergleichbar mit einer fast unendlichen Bibliothek, die nicht aufgeräumt wurde.
 - Es gibt nicht nur Texte, sondern alle möglichen Informationen, die man elektrisch transportieren kann.
 - Das Finden von Informationen stellt das größte Problem dar.
 - Informationen sind oft nicht mehr aktuell oder einfach falsch.
- Also: Informationen auf Webseiten immer kritisch ansehen
 - z.B. Wikipedia

Wie alles anfing...

- der erste Internetanschluss an der Uni Karlsruhe...
 - nutzte den Anschluss des BGH
 - 200 Baud = 0,2 kBit/s (kilo Bits pro Sekunde, kilo:= $2^{10}=1024$)
 - Fax: 9600 Baud = 9,6 kBit/s
 - Modem: 14,4 kBit/s
 - ISDN: 64 kBit/s
 - DSL 6000: 6 MBit/s = 6000 kBit/s, nur theoretisch!
- 1 Bit = 0 oder 1, 1 Byte = 8 Bit = 1 Zeichen
- Internet verband Universitäten weltweit
 - keine Sicherheitsbeschränkungen erforderlich
 - freundliches Umfeld
 - Austausch von E-Mails und kleinen Datenmengen

Einführung: Wie freundlich ist das Internet heute?

- Internet ist ein feindliches Umfeld!
 - Datenklau führt zu materiellem Verlust
 - z.B. Ausspähen der Kreditkarten-Nummer
 - illegaler Tausch von Musik („MP3“) und Kinofilmen
 - Viele unseriöse Angebote
 - ca. 40% der Auktionen bei ebay
 - Händler, die Vorkasse verlangen und dann keine Ware senden
 - gefälschte Markenware
 - kriminelle Aktivitäten
 - Terroristen tauschen Informationen
 - Schäden werden oft zum Spaß verursacht
 - Hacker, Viren, Trojaner

Internet-Zugang

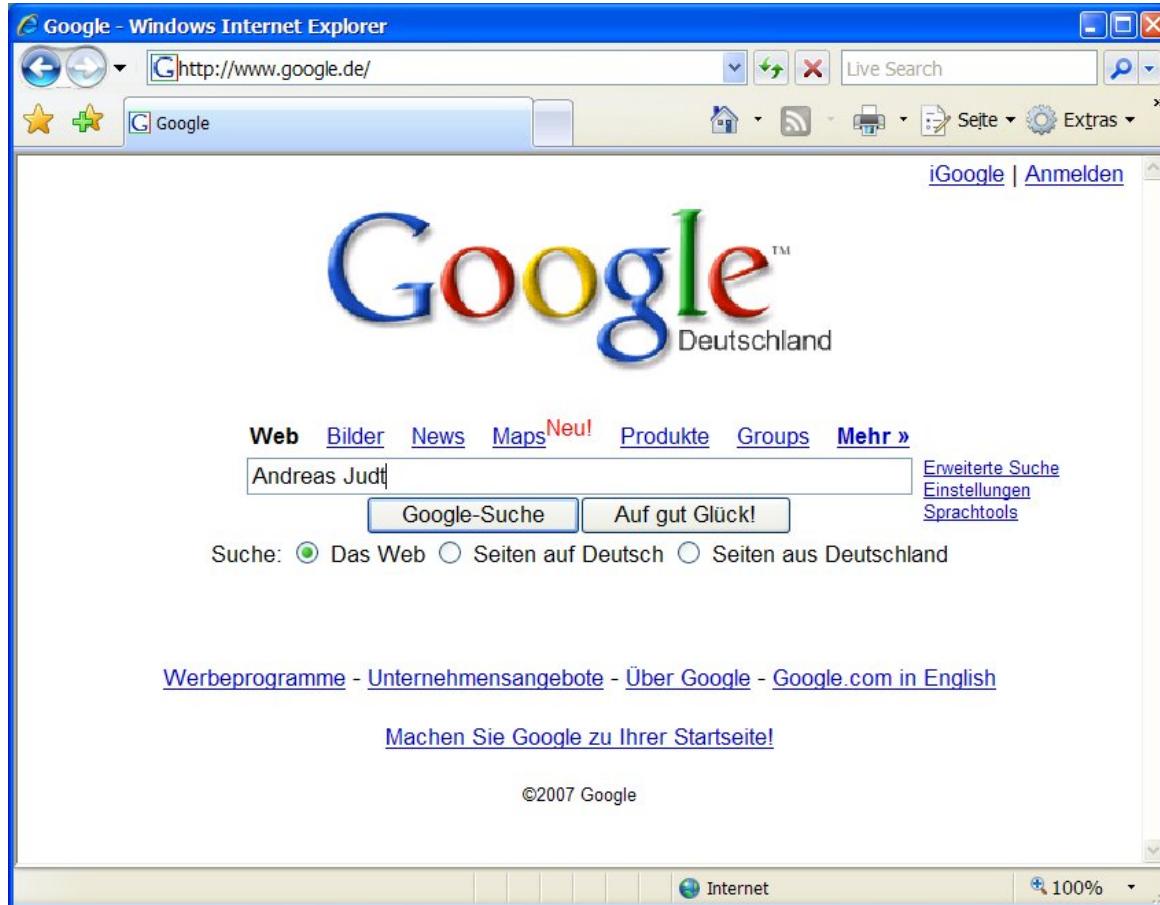
DSL, ISDN, analoges Modem

- analoges Modem (Modulator und Demodulator)
 - an TAE-Dose (Telekommunikations-Anschluss-Einheit) parallel zum Telefon angeschlossen
 - entweder Internet ODER telefonieren
- ISDN (integrated services digital network)
 - 2 Kanäle, Telefon UND Internet, „doppelte Kosten“
- DSL (digital subscriber line)
 - Internet und Telefon gleichzeitig
 - Splitter trennt Telefonanschluss und Internet-Zugang
 - Internet-Zugang über DSL-Modem (nur 1 Rechner) oder WLAN-Router (per Funk)

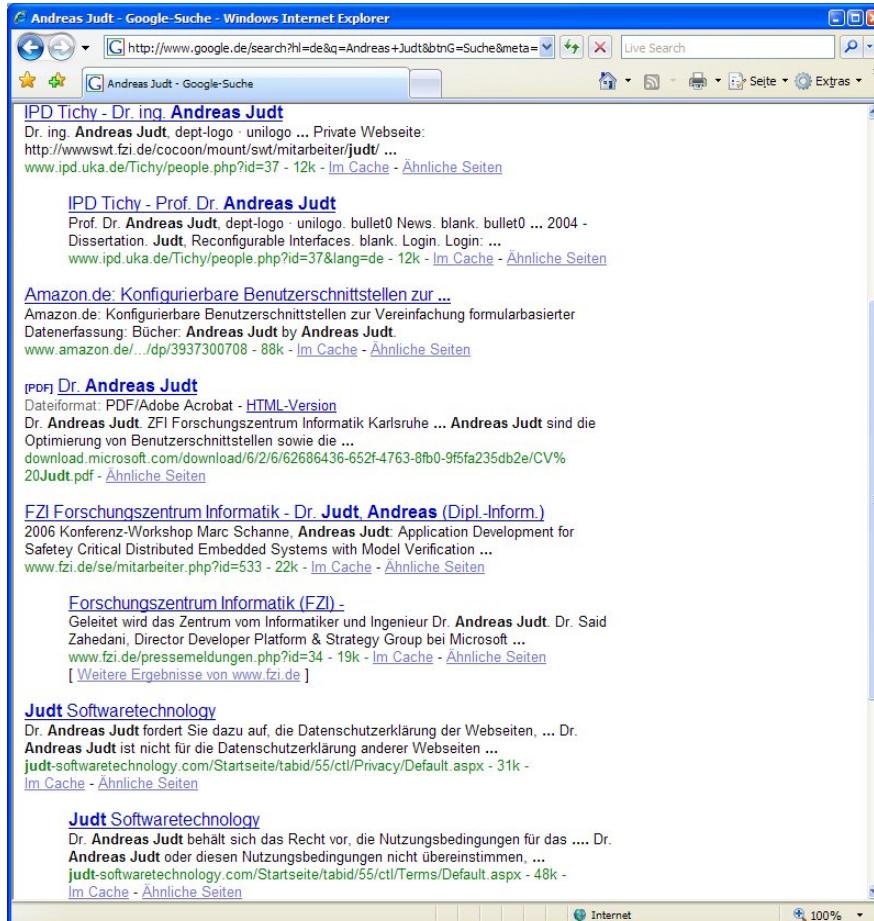
Meine persönlichen Daten im Internet

- Was weiß das Internet über mich?
 - Informationen aus Webseiten, z.B.
 - Ergebnislisten
 - Mitarbeiterseiten
 - Vereinsmitteilungen
- Informationen werden teilweise archiviert
 - z.B. Way-Back-Machine liefert den Inhalt einer Webseite zu einem vergangenen Zeitpunkt

Suchen mit Google: Andreas Judt



Google Ergebnisse: > 700.000 Treffer



Beispiel einer Reaktion von Google

- Bei der Bundestagswahl 2009 hat FDP-Mann Guido Westerwelle in einer Pressekonferenz einen Reporter der englischen BBC aufgefordert, seine Fragen in deutscher Sprache zu stellen.
- Google hat den Aspekt verarbeitet und zeigte bei einer entsprechenden Suchanfrage passende Werbung.

Suche nach Guido Westerwelle, Oktober 2009

guido westerwelle - Google-Suche - Windows Internet Explorer
http://www.google.de/search?hl=de&source=hp&fkt=1528&fsdt=3598&q=guido+westerwelle&meta=&aq=0&aqj=0

guido westerwelle - Google-Suche

Web Bilder Videos Maps News Shopping E-Mail Mehr ▾ Anmelden

Google guido westerwelle Suche Erweiterte Suche Einstellungen

Suche: Das Web Seiten auf Deutsch Seiten aus Deutschland

Web Ergebnisse 1 – 10 von ungefähr 811.000 für **guido westerwelle**. (0,07 Sekunden)

News-Ergebnisse für guido westerwelle

Vizekanzler in spe **Westerwelle** ist noch nie wirklich angekommen - vor 15 Stunden gefunden
Oktober 2009, 17:48 Uhr FDP-Chef **Guido Westerwelle** gilt als fleißig und zuverlässig. ...
Natürlich kommt **Guido Westerwelle** seinen Spötttern entgegen. ...
WELT ONLINE - 166 weitere Artikel »
Merkel und **Westerwelle**: Ich und ich - Handelsblatt - 2245 weitere Artikel »
Englisch mit **Guido Westerwelle** glaubt, dass er ein "Running Gag" wird -
WELT ONLINE - 166 weitere Artikel »

Guido Westerwelle - Vorsitzender der FDP Bundestagsfraktion und ...
Präsentation des Fraktions- und Parteivorsitzenden der Liberalen mit Einzelheiten zu seiner Person sowie Ausführungen zu seiner politischen Arbeit in ...
Kontakt - Person - Weitere Termine - Zur Übersicht
www.guido-westerwelle.de/ - Im Cache - Ähnlich

Guido Westerwelle, FDP - Lebenslauf, Fotos, Wahlkreis
Mit 19,1 Prozent der Erststimmen hat **Guido Westerwelle** in seinem Bonner Heimat-Wahlkreis ... **Guido Westerwelle** war von 1994 bis 2001 FDP-Generalsekretär. ...
www.guido-westerwelle.de/Person/.../index.html - Im Cache - Ähnlich

Weitere Ergebnisse anzeigen von www.guido-westerwelle.de

Guido Westerwelle – Wikipedia
Guido Westerwelle hatte zunächst an der Realschule seine Mittlere Reife abgeschlossen. Nach der Schulzeit in Oberdollendorf, Königswinter und Bonn mit ...
Ausbildung - Partei - Abgeordneter - Familie und Privates
de.wikipedia.org/wiki/Guido_Westerwelle - Im Cache - Ähnlich

Eraebnisste Bildersuche nach **guido westerwelle** - Bilder melden

Anzeigen

Shame for your English?
Der Aufschwung für Ihre Auftritte:
Mit Top-Sprachkursen von Berlitz!
www.berlitz.de/Englisch_verbessern

Schalten Sie hier Ihre Anzeige »

Einige Fakten über Google, publiziert 09/2011

- Google hat einen Leistungsbedarf von 260 Megawatt.
 - vergleichbar mit einer Stadt mit 200.000 Haushalten
 - davon entfallen 12,5 Megawatt auf die Google-Suche bei 1 Mrd. Suchanfragen täglich
- Eine Suche benötigt 0,3 Wattstunden.
 - vergleichbar mit einer 9W Energiesparlampe, die 2 Minuten brennt

E-Mail = elektronische Post

- E-Mail=elektronische Post (engl. electronic mail)
 - 1 Absender
 - beliebig viele Empfänger
 - Brieftext
 - Dateianhänge
- Bearbeitung von E-Mails
 - in einem speziellen Programm, z.B. Microsoft Outlook
 - Webseiten im Internet
- Stichworte zur Konfiguration
 - Mails abholen: POP, POP3
 - Mails senden: SMTP
 - neuere Technik: IMAP oder Exchange

Was ist Spam?

- Werbe-E-Mails, die meistens anonym versendet werden.
- Enthalten eine Aufforderung, eine bestimmte Webseite anzuklicken.
 - oft verbunden mit einem Virus oder einem unseriösen Angebot
- SPAM ursprünglich: Spiced Pork And Ham
 - übersetzt: Frühstücksfleisch

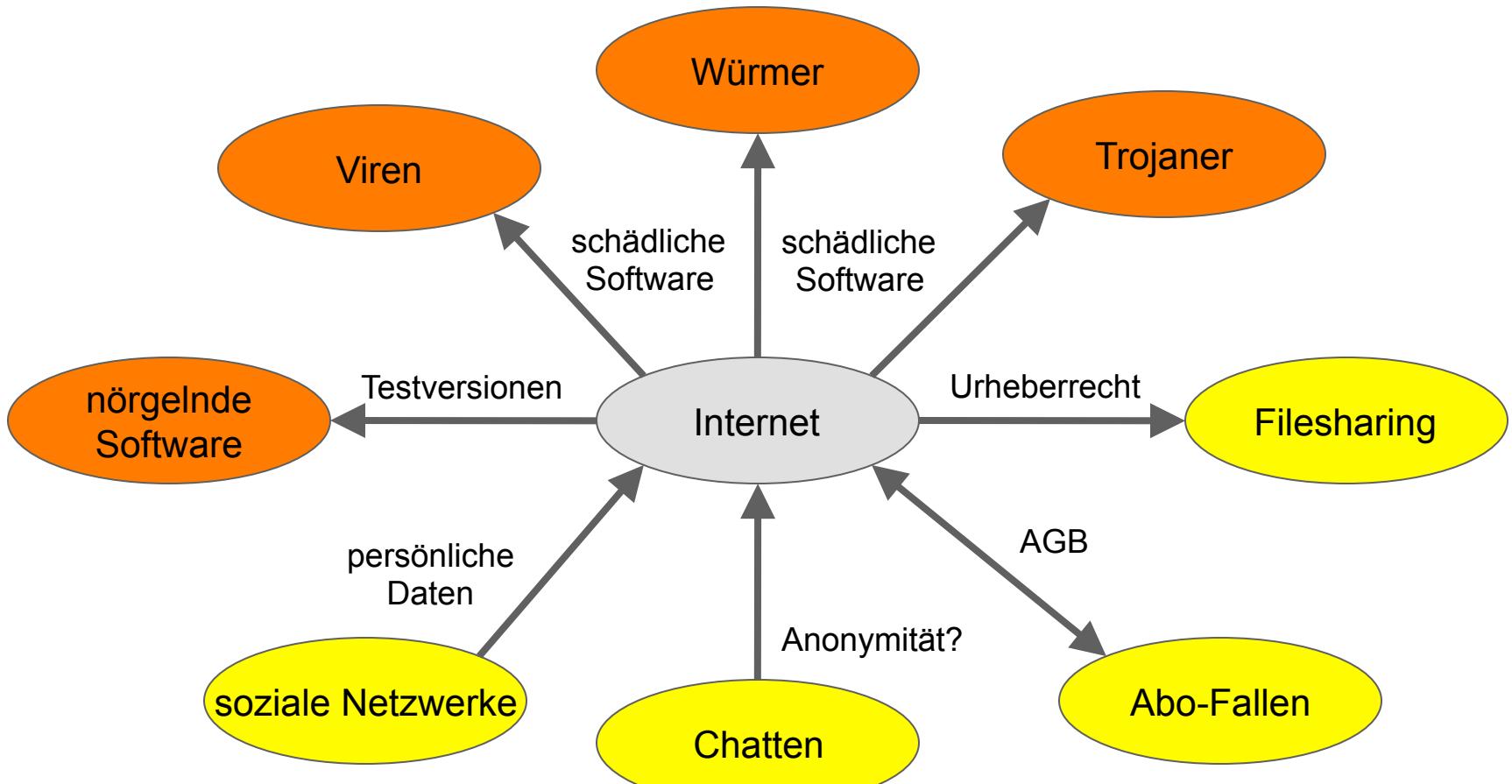
Spam ursprünglich



Schädliche Mails: Aufforderung zum Selbstverrat

- E-Mails können leicht gefälscht werden.
 - E-Mail Adresse des Absenders gefälscht, z.B. Mail von einem persönlich bekannten, die von einem Virus erzeugt wurde
 - Absender ist angeblich eine Bank oder ein Anwalt, z.B. Aufforderung eine Datei zu öffnen oder eine Webseite zu besuchen (Link in der Mail)
 - Im schlechtesten Fall genügt das Herunterladen der Mail auf den Computer!

Die wichtigsten Begriffe



Was ist ein Virus?

- Computerviren sind mit biologischen Viren vergleichbar.
 - schaden dem Wirt (Computer) und richten Schaden an (z.B. Datenverlust)
- Viren verbreiten sich nicht selbst sondern werden weitergegeben
 - z.B. durch verseuchte Dateien per E-Mail
 - der Virus wird ausgeführt, wenn die verseuchte Datei geöffnet oder gestartet wird
- Schutz gegen Viren: VirensScanner
 - sollte spätestens alle 3 Tage aktualisiert werden
 - meistens kostenfrei, z.B. <http://www.freeav.de>

Was ist ein Wurm?

- Würmer dringen aktiv in Computersysteme ein und verbreiten sich weiter
 - Würmer nutzen Fehler (Sicherheitslücken) im Betriebssystem aus
 - Würmer können Spuren vertuschen und Sicherheitslöcher selbst nach Eindringen beseitigen
- Typische Angriffspunkte
 - Systeme ohne oder mit schwachen Passwörtern
 - Systeme, die nicht aktualisiert werden
 - E-Mail Programme

Was ist ein Trojaner?

- Ein Trojaner wird mit einer scheinbar harmlosen Software eingespielt
 - Computerspiel
 - Video
 - Word-Dokument
 - Bild
 - andere
- Trojaner...
 - spähen den Rechner aus
 - protokollieren Passwörter
 - zeichnen das Surf-Verhalten im Internet auf
 - untersuchen Dateien auf der Festplatte

Was ist nörgelnde Software?

- Nörgelnde Software ist typischerweise als Testversion eines kostenpflichtigen Programms getarnt.
- typisches Verhalten
 - erzeugt viele falsche Fehlermeldungen
 - weist darauf hin, dass die Vollversion die angeblichen Fehler behebt
 - lässt sich nicht mehr entfernen

Was sind Kostenfallen?

- Webseiten locken mit vermeintlich kostenlosen Angeboten.
 - Zugriffe führen zunächst auf eine Registrierung, bei der Nutzungsbedingungen bzw. AGB akzeptiert werden müssen.
- Spätestens in den AGB findet sich dann die Vertragsbedingung für ein mehrjähriges Abo.
 - Die Fallensteller verfügen über bestens ausgerüstete Anwälte und Inkassobüros.
 - Oft wird den Geschädigten dann noch massiv mit angeblichen Kosten und Schufa-Einträgen gedroht.
 - Fallensteller agieren oft aus dem Ausland. Strafrechtlich sollte dann der sog. Admin-C verfolgt werden.

Wie erkennt man Kostenfallen?

- Eine interessante Dienstleistung wird vermeintlich kostenfrei bzw. gegen Registrierung angeboten.
 - Auf der Startseite findet sich ein kaum erkennbarer Kostenhinweis.
- Ziel des Betreibers: persönliche Informationen sammeln und nicht kündbaren Abo-Vertrag in Rechnung stellen.
 - Registrierung
 - Gewinnspiel / Gewinnbenachrichtigung

Beispiel 1: hausaufgaben.de

hausaufgaben-heute.com - Windows Internet Explorer
http://www.hausaufgaben-heute.com/affiliate/layout2/?wid=88pid=25001&sub=hausaufgaben_de

hausaufgaben-heute.com

Widerrufsrecht:
(Download als PDF/Word)
Diese Internet-Seiten sind ein Angebot von:
Andreas & Manuel Schmidlein OHG
Vor der Huber 3
D-64572 Büttelborn
UmsatzsteuerID: DE230209554
Handelsregister: Amtsgericht Darmstadt, HRA 83418
Telefon: 0180-5060530-928 (0,14€/Min a.d. dt. Festnetz,
ggf. abweichende Preise a.d. Mobilfunknetz)
E-Mail an: support@hausaufgaben-heute.com

Informationen zum Widerrufsrecht:
Der Kunde kann, sofern er Verbraucher im Sinne des § 13 BGB ist, die Vertragserklärung innerhalb von zwei Wochen ohne Angabe von Gründen in Textform (z.B. Brief, Fax, E-Mail) widerrufen. Die Frist beginnt frühestens mit Erhalt dieser Belehrung. Zur Wahrung der Widerrufsfrist genügt die rechtzeitige Absendung des Widerrufs. Der Widerruf ist zu richten an folgende Adresse, E-Mail oder Fax:
Andreas & Manuel Schmidlein OHG
Vor der Huber 3, D-64572 Büttelborn
Fax: 0180-5060530-329 (0,14€/min aus dem dt. Festnetz,
ggf. abweichende Preise a.d. Mobilfunknetz)
E-Mail an: support@hausaufgaben-heute.com

Das Widerrufsrecht des Kunden erlischt vorzeitig, wenn der Dienstleister mit der Ausführung der Dienstleistung mit ausdrücklicher Zustimmung des Kunden vor Ende der Widerrufsfrist begonnen hat oder der Kunde diese selbst veranlasst hat. Die Inanspruchnahme der Dienstleistung vor Ablauf des Testzeitraums (§ 3 Abs. 4 und 5 AGB) bleibt dabei unberücksichtigt.

Widerrufserfolgen:
Im Falle eines wirksamen Widerrufs sind die beiderseits empfangenen Leistungen zurückzugewähren und ggf. gezogene Nutzungen (z.B. Zinsen) herauszugeben. Kann der Kunde die vom Dienstleister empfangene Leistung ganz oder teilweise nicht oder nur in verschlechtertem Zustand zurückgewähren, muss er dem Dienstleister insoweit ggf. Wertersatz leisten. Verpflichtungen zur Erstattung von Zahlungen hat der Kunde innerhalb von 30 Tagen nach Absendung seiner Widerrufserklärung zu erfüllen.

Informationen zu einzelnen Bestimmungen des Vertrages:
Es gelten die jeweils gültigen Allgemeinen Geschäftsbedingungen. Merkmale der Dienstleistungen und Preisangaben sind unmittelbar im Zusammenhang mit der angebotenen Dienstleistung beschrieben.
Der Vertrag zwischen Ihnen und uns kommt in folgender Weise zustande: Sie geben in der Registrierungsmaske die erforderlichen Daten (Name, Anschrift, E-Mail-Adresse und Geburtsdatum) ein. Die Registrierungsdaten können Sie bis zur Betätigung des Buttons »Anmelden« ändern. Die Änderungen können mittels Maus und/oder Tastatur vorgenommen werden. Nach der Betätigung des Buttons »Anmelden« geben Sie gegenüber Andreas & Manuel Schmidlein GbR eine verbindliche Erklärung über die kostenpflichtige Nutzung des Memberbereiches ab. Gleichzeitig akzeptieren Sie unsere Allgemeinen Geschäftsbedingungen.

Impressum

ANMELDEN UND 3000 HAUSAUFGABEN DOWNLOADEN!

Anrede

Vorname / Name

Straße / Nr.

PLZ / Wohnort

Land

Geburtsdatum

E-Mail-Adresse *

Ich akzeptiere die [AGB](#), die [Datenschutzerklärung](#) und das [Widerrufsrecht](#).

ANMELDEN

* Die Zugangsdaten werden Ihnen per E-Mail zugeschickt.

Willkommen auf hausaufgaben-heute.com



Memberbereich / Login | Support | Kontakt | Datenschutzerklärung | **Widerrufsrecht** | Partnerprogramm | AGB

Beispiel 2:

genealogie.de - Wissenschaftliche Datenbank für Namens- und Ahnenforschung - Windows Internet Explorer

http://www.genealogie.de/

genealogie.de - Wissenschaftliche Datenbank für Namens- und Ahnenforschung

Wissenschaftliche Datenbank
für Namens- und Ahnenforschung

Was bedeutet Dein Name? Hast Du adlige Ursprünge?

dein Vorname ... dein Nachname ... abfragen

GENEALOGIE.DE
NAMENS- UND AHNENFORSCHUNG

NAMENFORSCHUNG
112.332 Namen in der Datenbank
Bedeutungen, Babynamen, lustige Namen

AHNENFORSCHUNG
Mit wem bist du verwandt?
Professioneller Verwandtschaftscheck

Was bedeutet dein Vorname?
Was bedeutet dein Nachname?

HAST DU PROMINENTE VERWANDTSCHAFT?

A. Einstein Sissi Napoleon Kaiser Wilhelm II

HERZLICH WILLKOMMEN BEI GENEALOGIE.DE

Datenbank für Namens- & Ahnenforschung

Durchstöber jetzt unsere wissenschaftliche Datenbank mit über 112.332 Namen - Bedeutungen, Babynamen, lustige Namen oder die derzeit am häufigsten genutzten Namen. Außerdem kannst du deinen eigenen Stammbaum (Verwandtschaft) dokumentieren.

Eigener Stammbaum

Willst du wissen mit wem du verwandt bist? Hast du adlige Ursprünge oder prominente Verwandtschaft? Erstelle jetzt deinen eigenen Stammbaum und erforschen die Wurzeln deines Ursprungs. Sicher wirst auch du vom Fieber der Genealogie (Ahnenforschung) angesteckt.

Starte jetzt Deine eigene Namens- oder Ahnenforschung und gewinne ein großes Familienfest oder 5.000,00 Euro in Bar!

teilnehmen

BABYNAMEN

Marie, Leonie, Laura, Lea, Noel, Lorena, Emmely Mia, Kennet, Leon, Alexander, Leandro, Finley

weitere Namen

ÜBER 100.000 VORNAMEN

Adrian, Alvin, Achim, Alicia, Agustin, Andrey, Adele, Aeriel, Alain, Antoine, Adelina, Adelheid

weitere Namen

Zugangsdaten vergessen · AGB/Verbraucherinformationen/Datenschutz · Impressum

http://www.genealogie.de/index.php?w=&ac=anmelden

Internet 100%

The screenshot shows the homepage of genealogie.de. At the top right, there's a search bar for names and a button to search. The main header is "GENEALOGIE.DE NAMENS- UND AHNENFORSCHUNG". On the left, a sidebar has links for Startseite, Namensforschung, Ahnenforschung, Genealogie, Bist du adlig?, Vornamen, Nachnamen, Babynamen, and Gewinnspiel. Below that is a "GEWINNSPIEL" section with a photo of a family and a button to participate. The central part of the page is divided into three main sections: "NAMENFORSCHUNG" (with a book icon and a baby), "AHNENFORSCHUNG" (with icons of famous people), and "HERZLICH WILLKOMMEN BEI GENEALOGIE.DE" (with a large photo of a family). There are also sections for "BABYNAMEN" (listing names like Marie, Leonie, Laura, Lea, Noel, Lorena, Emmely Mia, Kennet, Leon, Alexander, Leandro, Finley) and "ÜBER 100.000 VORNAMEN" (listing names like Adrian, Alvin, Achim, Alicia, Agustin, Andrey, Adele, Aeriel, Alain, Antoine, Adelina, Adelheid). At the bottom, there are links for forgotten login data, terms and conditions, privacy policy, and a footer with the URL and a 100% zoom indicator.

Beispiel 3: opendownload.de

The screenshot shows a Windows Internet Explorer window displaying the login page for opendownload.de. The URL in the address bar is <http://www.opendownload.de/anmelden.php?name=OpenOffice>. The page features a sidebar with categories like Sicherheit, Grafik, Multimedia, System-Tools, Hobby, Internet, Telekommunikation, Büro, Spiele, and Schule. A section titled 'Vorteile als Mitglied' explains benefits for registered users. The main content area highlights 'OpenOffice 3.0.1' with a 5-star rating and a screenshot of the software interface. A message prompts users to fill out personal data fields. The 'Persönliche Daten' section includes fields for Anrede, Vorname, and Nachname. The 'Adressdaten' section includes fields for Straße / Nr., Postleitzahl, Wohnort, and Land. The 'sonstige Angaben' section includes fields for Geburtsdatum and E-Mailadresse, along with a checkbox for accepting terms and conditions. A large blue 'Anmelden' button with a key icon is at the bottom. The status bar at the bottom right shows '100%'.

Impressum von opendownload.de

The screenshot shows a Windows Internet Explorer window displaying the Impressum page of the openDownload.de website. The URL in the address bar is <http://www.opendownload.de/impressum.php>. The page title is "OPENDOWNLOAD.DE". The main content area displays the Impressum information:

Impressum

Content Services Ltd.
Mundenheimer Straße 70
68219 Mannheim

Handelsregister: HRB 703297, Amtsgericht Mannheim

Umsatzsteuer-ID: DE258263766

Kontakt: www.support-online-center.com

Telefon: +49-1805-88204486 (0,14 EUR/min) *
Telefax: +49-1805-88204487 (0,14 EUR/min) *
(Montag bis Freitag / 09.00 - 18.00 Uhr)

* aus dem deutschen Festnetz; gegebenenfalls abweichende Preise aus dem Mobilfunknetz

Vorteile als Mitglied

Als registrierter Benutzer bei opendownload.de erhalten Sie Zugriff zu allen Programmen inkl. Bewertungen. Wir veröffentlichen regelmäßig neue Programme.

Inhalte

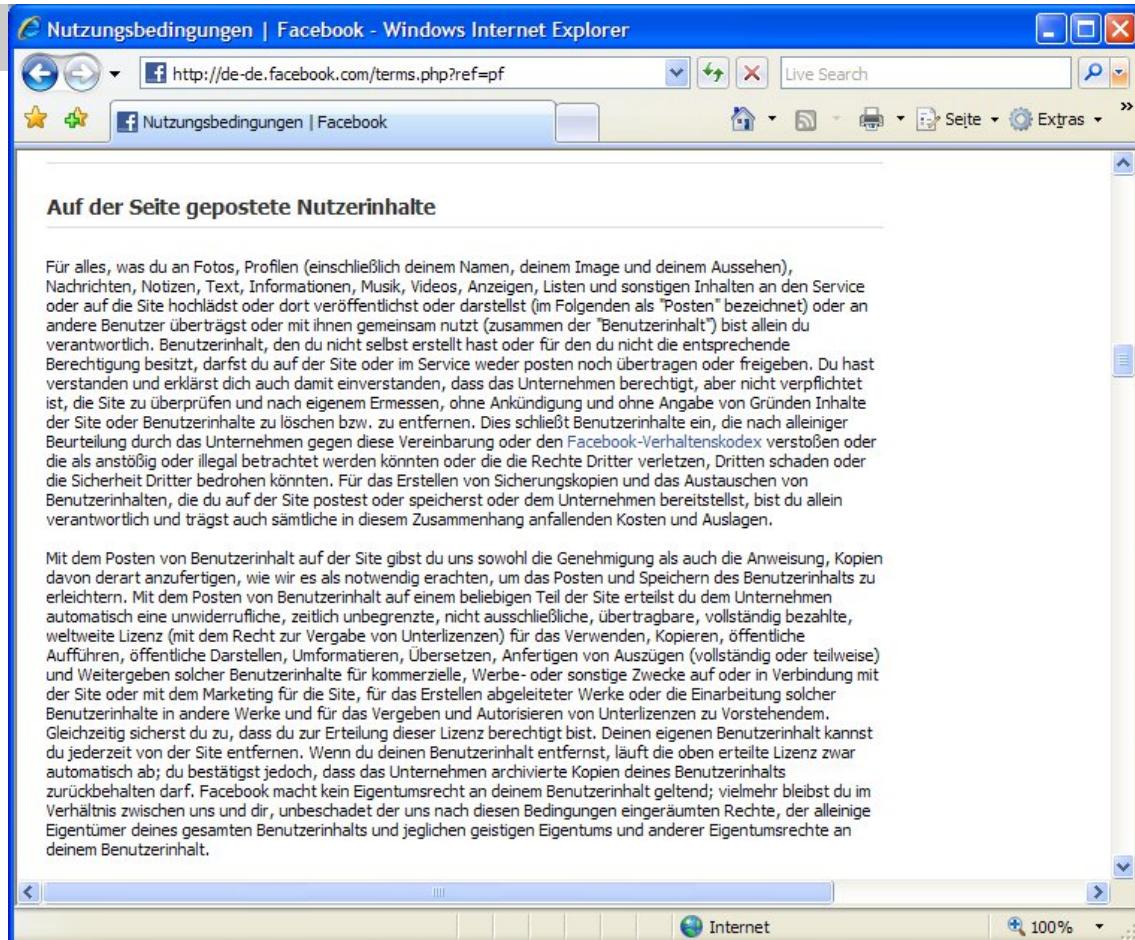
selbstständige Zweigniederlassung der

Content Services Ltd.
5 Jupiter House, Calleva Park
Aldermaston
Reading Berkshire RG7 8NN
Company No. 06326552 (Cardiff)
Director: Alexander Varin

Beispiel 4: facebook



Nutzungsbedingungen von facebook



Was ist Filesharing?

- In Filesharing-Netzwerken werden große Dateien ausgetauscht.
 - Programme, Spiele, Videos, Musik
- Die Leistung des Netzwerks entsteht dadurch, dass jeder Downloader die Daten sofort wieder bereitstellt.
- typische, auch als Peer-to-Peer bezeichnete Netzwerke
 - BitTorrent, eMule, Limewire

Nutzen von Peer-to-Peer Netzwerken

- Programme, Spiele Musik und Videos sind häufig urheberrechtlich geschützt.
- Die Bereitstellung urheberrechtlich geschützter Daten ist strafbar.
 - Bei Peer-to-Peer Netzwerken ist damit auch der Download eine Verletzung des Urheberrechts.
- Die Abmahnung von Nutzern dieser Netzwerke ist in Deutschland ein riesiger Markt für Anwälte!
 - Dafür werden in den Netzwerken gezielt Fallen gestellt.

Musik aus dem Internet

- früher: napster
 - wurde auf Drängen der Musikindustrie verboten
 - offene Tauschbörse, praktisch alle Titel waren online verfügbar
- heute: viele Anbieter
 - musicload.de
 - napster (Name wurde aus Marketing-Gründen übernommen)
 - AOL music
 - Saturn download
- Jeder heruntergeladene Titel muss einzeln bezahlt werden.
 - Gekaufte Titel als Windows Media Formate (z.B. WMA) können vom Rechner gegen den Willen des Benutzers nach einer bestimmten Zeit einfach gelöscht werden!

Soziale Netzwerke

- Soziale Netzwerke sind Plattformen für Nutzer mit gleichen Interessen.
- In den Netzwerken geben die Nutzer typischerweise persönliche Daten preis.
 - Adresse, Fotos, Hobbies, Urteile über andere, ...
- Beispiele
 - schülerVZ, studiVZ, wer-kennt-wen, spickmich, meinprof, xing

Was ist Cyber-Mobbing?

- In sozialen Netzwerken können gezielt falsche Informationen oder Diffamierungen platziert werden.
- Betroffene können sich praktisch nicht wehren.
- Beispiele für Mobbing
 - doppelte Profile mit falschen Aussagen
 - Beleidigungen
 - peinliche Fotos

Meine Identität im Internet

- Informationen über eine Person können leicht zu einem Profil zusammengestellt werden:
 - Informationen auf öffentlichen Webseiten
 - Profile in sozialen Netzwerken
 - Vorratsdatenspeicherung
 - IP-Adresse
 - Verlauf der Webseiten-Besuche
- Im Internet sind Daten unbegrenzt haltbar und selbst nach Jahrzehnten noch abrufbar!
 - Durch Verknüpfung der Datenquellen können dann Rückschlüsse gezogen werden.

Datenverlust durch Sicherheitslücken- eine Auswahl

- 28.04.2011: UNESCO
 - Bewerberunterlagen stehen öffentlich zum Download
- 28.04.2011: TomTom
 - TomTom verkauft Streckendaten von Anwendern an niederländische Strafverfolgungsbehörden
- 19.04.2011: Sony
 - 70 Mio. Kundendatensätze gestohlen
- 04.05.2010: SchülerVZ
 - 1,6 Mio. Datensätze durch Konzeptfehler gestohlen
- 31.03.2010: Facebook
 - Facebook publiziert irrtümlich 400 Mio. private Mailadressen
- 16.10.2009: Finanzdienstleister AWD
 - NDR bekommt 27.000 Kundendatensätze zugespielt

Datenverlust durch Sicherheitslücken- eine Auswahl

- 28.02.2009: US Regierung
 - Daten über Obamas Hubschrauber über P2P im Iran heruntergeladen
- 27.01.2009: Telekom
 - Datensätze von Neukunden im Internet abrufbar
- 07.12.2008: diverse Callcenter
 - 21 Mio. Datensätze über Sicherheitslücken gestohlen
- 24.09.2008: Polizei Berlin
 - Handynummern von Ermittlern auf den Webseiten der GdP
- 03.09.2008: Beate Uhse
 - Google findet Kundendatenbank und indiziert Namen
- 16.08.2008: Kinderkanal KI.KA
 - Mehrere Hundert private Datensätze von Kindern öffentlich auf den Webseiten sichtbar

Neue Geschäftsmodelle im Internet: Erpressung mit Trojanern

- Eine neue Generation von Trojanern befällt Computer und erpresst Geld.
- Ähnlich wie bei Klingeltönen wird eine SMS an eine bestimmte Nummer gesendet.
 - Als Antwort wird ein Code zum Freischalten des Rechners zurückgesendet – gegen entsprechende Gebühr.
 - Erpressung kann zu beliebigen Zeitpunkten wiederholt werden.
 - Modell funktioniert nur, wenn die Identität des Anschlussinhabers gedeckt wird.

Beispiel: Kreditkartendaten sammeln mit Scareware

Copyright violation: copyrighted content detected

Windows has detected that you are using content that was downloaded in violation of the copyright of its respective owners. Please read the following bulletin and try solving the problem in one of the recommended ways.

[English](#)

What has happened?

During the system scan Antipiracy foundation scanner has detected copyright issues. Please take a look at the list and choose an action: pass the case to a court or settle it in pre-trial order by paying a fine.

Files detected

Evidence list

[Show details...](#)

Lawsuits preview

Page 1/2 Page 2/2

Used IPs log

176.4.2.0
176.4.2.0
176.4.2.0

Type of violation

p2p\warez movie download
p2p\warez games download
p2p\warez mp3 download

Antipiracy news

12/02/2010
[New antipiracy measures are being taken against illegal content](#)

26/12/2009
[Antipiracy client updated: Download version 2.0 and scan your PC for illegal and spyware content \(e. g. music and videos\)](#)

Choose an action

If you are sure that you can't have download that content to your PC or there was nothing you could do to avoid it, press "Pass the case to court" button and pass the case to court

If these files belong to you, but you would rather avoid all the expenses associated with settling the issue in court, you can settle your case in pre-trial order by pressing "Solve..." button

Pass the case to court

Settle case in pre-trial order

[Enter a previously purchased license code](#)

<http://icpp-online.com/> - your source for copyright initiative

All rights reserved by their respective owners, 2010

Beispiel: Kreditkartendaten sammeln mit Scareware

- Der Trojaner meldet angebliche Urheberrechtsverstöße und schlägt eine außergerichtliche Einigung gegen die Überweisung von 400 Dollar vor.
(Quelle: www.heise.de, 14.04.2010)
- Tatsächlich sammelt der Trojaner Daten von Kreditkarten.
- Der Forderung wird durch Austauschen des Bildschirmhintergrunds Nachdruck verliehen:

Beispiel: Kreditkartendaten sammeln mit Scareware



Web Engineering

- Ingenieurmäßige Entwicklung von dynamischen Web-Anwendungen
 - Web bezeichnet eigentlich eine beliebige Vernetzung, die heute größtenteils durch das Internet abgedeckt wird.
- Vorlesung zeigt
 - Entwicklung der Technologien
 - Stand der Technik heute
 - Entwicklung zukünftiger Standards

Kontakt

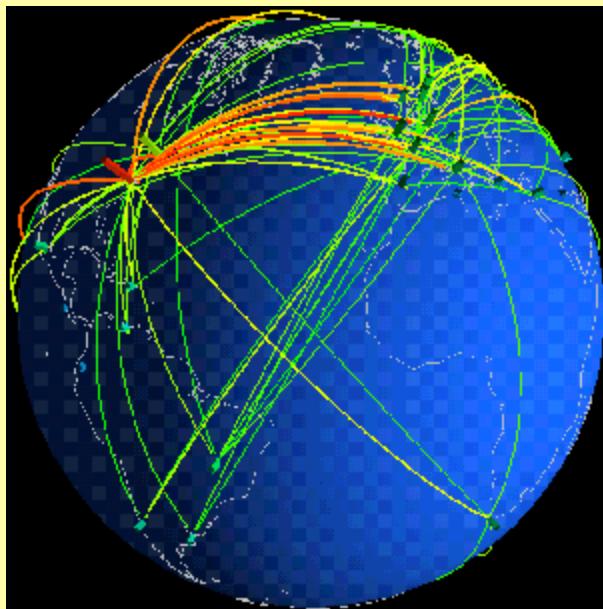
Duale Hochschule BW Ravensburg
Campus Friedrichshafen

Prof. Dr. Andreas Judt
Informationstechnik

judt@dhbw-ravensburg.de

Web Engineering

Überblick HTML5 mit CSS3 und AJAX



Prof. Dr. Andreas Judt

Klassisches HTML: strukturierte Dokumente

- Die Hyper Text Markup Language (HTML) ist eine Dokumentbeschreibungssprache, die
 - logische Strukturen zur Beschreibung eines HTML-Dokuments beinhaltet.
 - Befehle (Tags) definiert, die nicht angezeigt werden und Eigenschaften der angezeigten Elemente oder des Dokuments bestimmt.
- Transport: HTTP (Hypertext Transport Protocol)

HTML Tags

- In einer HTML-Seite vorkommende Tags können nach Art und Weise ihrer Verwendung eingruppiert werden:
 - Tags, die die Struktur einer HTML-Seite definieren
 - Tags zur Textformatierung
 - Tags zur Definition von Verweisen (links)
 - Tags für Bilder und andere Objekte
 - Tags zur Erstellung von Tabellen
 - Tags zur Erstellung von Formularen

Merkmale der ersten Versionen

- Einfaches Editieren in einem Texteditor
- Plattformunabhängigkeit
- Strukturierte Darstellung formatierter Textdokumente
- Logische Verknüpfung zwischen Dokumenten
- Geringer Bedarf an Rechenleistung

Erste Dynamisierung: JavaScript

- JavaScript ermöglichte Interaktion mit dem Anwender ohne das Neuladen über eine HTTP-Anfrage.
 - Wesentlicher Gewinn an Performanz für die Web-Anwendung.
 - Einfache Algorithmen rechnen lokal und entlasten den Web-Server.
- JavaScript setzte auf die Modewelle Java auf.
 - Anders als behauptet ist JavaScript nicht objektorientiert.
- JavaScript war aufgrund der Konkurrenz der Browserhersteller und mangelnder Spezifikation nur für triviale Aufgaben verwendbar.
 - Erst die Spezifikation von AJAX sicherte das Überleben von JavaScript durch eine sinnvolle Spezifikation.

Beispiel: Konkurrenzkampf in JavaScript bei HTML 3

- Applets müssen browserspezifisch integriert werden!
 - Lösung: J2SDK HTML Konverter
- Mit HTML5: mit OBJECT-Tag integriert.
- Beispiel J2SDK 6.0:

```
<script language="Javascript">
  var _app = navigator.appName;
  if (_app == 'Mozilla') {
    document.write('<embed code="Applet1.class"', 'width="200"',
      'height="200"', 'type="application/x-java-applet;version=1.6">');
  } else if (_app == 'Microsoft Internet Explorer') {
    document.write('<OBJECT ', 'classid="',
      "clsid:8AD9C840-044E-11D1-B3E9-00805F499D93"', '',
      'width="200"', 'height="200">', '<PARAM name="code"',
      'value="Applet1.class">', '</OBJECT>');
  } else {
    document.write('<p>Sorry, unsupported browser.</p>'); }
</script>
```

Übung: HTML 4

- Arbeiten Sie die Tutorials HTML/XHTML und JavaScript des Projekts JavaScript/DOM durch.

Sie finden es unter

<http://de.selfhtml.org>

- Bemerkung:
 - Sie können auch weitere Kapitel des Tutorials begleitend zu dieser Vorlesung bearbeiten.

Geschichte von HTML (Hypertext Markup Language) 1/4

- 1989 spezifiziert am europäischen Kernforschungszentrum CERN, Untermenge von SGML (Standard Generalized Markup Language).
- 1993 wurde der Mosaic Browser für Sun Microsystems publiziert.
- 1994 wird HTML 2 verabschiedet – nachdem Forderungen nach nicht-textuellen Inhalten wie Bildern) und Strukturierungsmöglichkeiten laut wurden. Netscape und das W3C wurden im gleichen Jahr gegründet.

Geschichte von HTML (Hypertext Markup Language) 2/4

- 1995 wurde HTML 3 publiziert. Wesentlicher Bestandteil war das Tabellenmodell CALS. Im gleichen Jahr wurde der Internet Explorer (als Version 2) publiziert. Forderungen an das W3C:
 - Frames, Scrollbars, Cascading Style Sheets (CSS) und Internationalisierung (i18n)
- 1997 wurden die Forderungen in der Version 3.2 umgesetzt.
- 1998 wurden HTML 4.01 und XHTML 1.0 mit drei verschiedenen Document Type Definitions (DTDs) publiziert. HTML Seiten basieren auf dem Document Object Model (DOM).

Geschichte von HTML (Hypertext Markup Language) 3/4

- 2002 bis 2006 wurden von W3C 8 Entwürfe für XHTML 2 publiziert.
- 2004 publizierte die Web Hypertext Application Technology Working Group einen Entwurf von HTML5, damals als Web Applications 1.0.
- 2007 publiziert W3C HTML5 als Entwurf.

Geschichte von HTML (Hypertext Markup Language) 4/4

- 2010 kündigt Apple-Gründer Steve Jobs an, dass Flash neben HTML5 keine Zukunft haben wird.
- Letzer Aufruf für Anforderungen war im Mai 2011. Eine stabile Version wird 2014 erwartet.
- 2011 publiziert Microsoft-Chef Steve Ballmer, dass Microsoft zukünftig auf HTML5 besonders bei mobilen Applikationen setzt.
- Skeptiker rechnen mit einer ersten vollständigen Implementierung bis 2022.

Wesentliche Neuerungen bei HTML5

1/3

- Verbesserte Tags, überholte Techniken eliminiert
 - Neue Techniken zur Verwendung von Daten und Datentypen
- Multimedia mit weniger Zusatzinstallationen über Plugins
 - Flash, Silverlight, Audio und Vektorgrafiken sind integriert.
- Bessere Interaktionsmöglichkeiten mit Anwendungen
 - Der Umfang von HTML5 macht Technologien von Drittanbietern überflüssig.
 - Leichtere Bedienbarkeit mit intelligenter Anpassung an verwendete Geräte.

Wesentliche Neuerungen bei HTML5

2/3

- Kommunikation mit anderen Dokumenten
 - Kommunikation zwischen Dokumenten unterschiedlicher Domains.
 - Bei früheren Versionen strikt verboten!
- Web Sockets
 - Dauerhafte Socketverbindungen zwischen Browser und einem Web Socket Server.
- Verbesserte grafische Gestaltungsmöglichkeiten
 - CSS3 und Canvas bieten neue Funktionen.

Wesentliche Neuerungen bei HTML5

3/3

- Verbesserte Formulare
 - Neue Interaktionselemente, einfache Prüfung von Datentypen und Wertebereichen.
- Verbesserte Barrierefreiheit
 - Neue Dokumentstruktur besser Einsatzmöglichkeiten von Bildschirmlesegeräten.
- Neue visuelle Effekte
 - Schatten, Bilder, Abrundungen und Verläufe bringen neue Möglichkeiten.

Aufwärtskompatibilität

- Browserhersteller kommen mit ersten (Teil-) Implementierungen
- HTML5 wird wie bisher über DTD gekennzeichnet, ohne Angabe der URL wird automatisch HTML5 gesetzt:
`<!DOCTYPE html>`
- Nicht implementierte Funktionalitäten können mit Ersatzimplementierungen in JavaScript kompensiert werden.

Veraltete HTML Tags

- Gestalterische Tags
 - basefont, big, center, font, s, strike, tt, u
- Frame Tags
 - frame, frameset, noframes
- Strukturtags
 - target bei Links
 - profile bei head
 - longdesc bei img und ifram

Veraltete HTML Attribute

- Attribute
 - align, link, vlink, alink, text bei body, bgcolor, height, width
 - scrolling für iframe
 - valign, hspace, vspace
 - cellpadding, cellspacing und border für table
- Layout-Strategien werden nur noch mit CSS implementiert.

Abwärtskompatibilität

- Bestehende HTML-Implementierungen müssen für HTML5 angepasst werden.
- Es kann durchaus sein, dass sich Browser bei der Aufwärtskompatibilität unterscheiden.

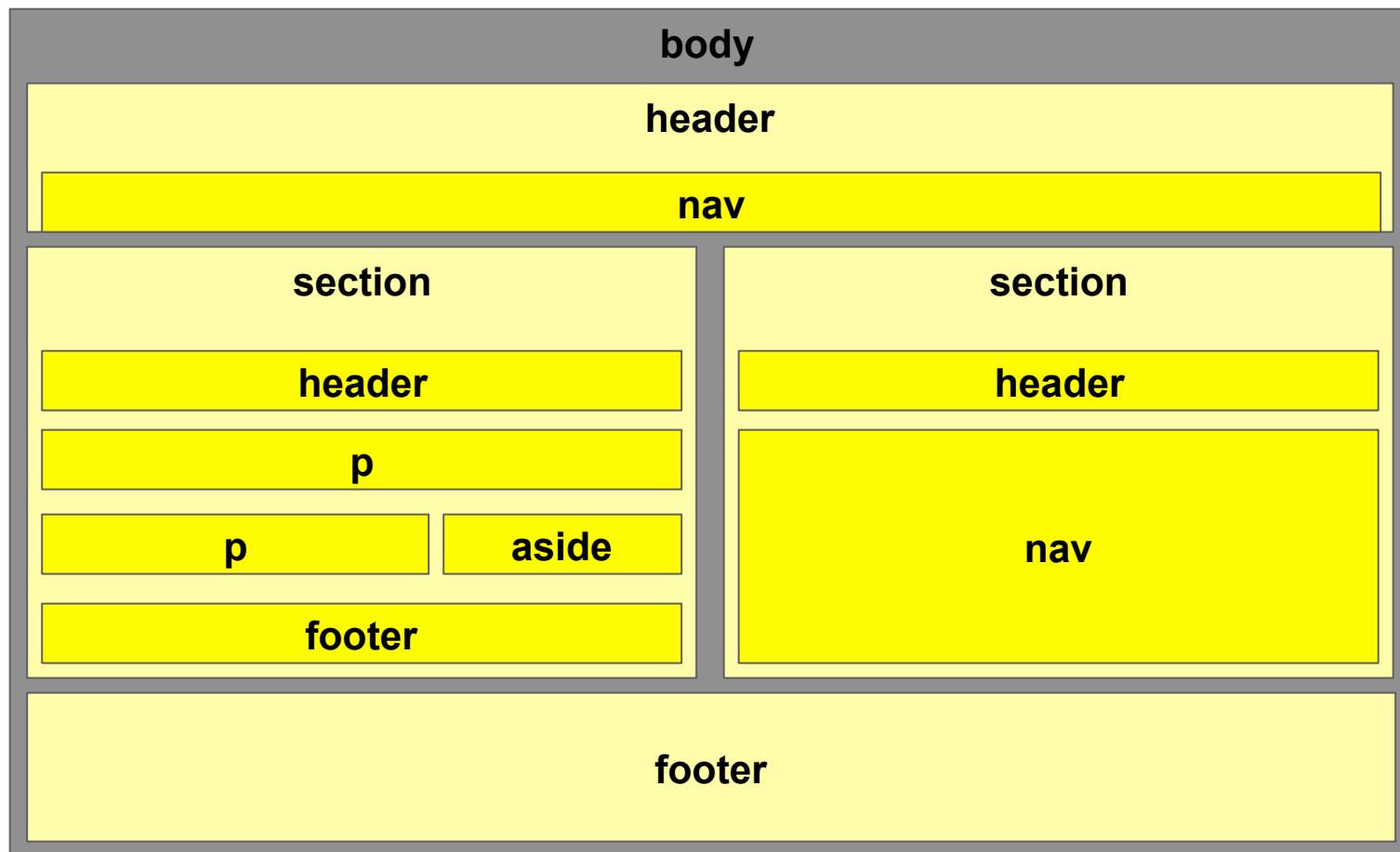
Typische Struktur eines HTML5 Dokuments 1/4

- <header>
 - Kopfbereich einer Seite oder eines Abschnitts <section>
- <footer>
 - Fußzeile einer Seite oder eines Abschnitts
- <nav>
 - Navigationsbereich einer Seite oder eines Abschnitts
- <section>
 - Logischer Abschnitt einer Seite oder einer Gruppe von Inhalten <article>, <aside>, etc.

Typische Struktur eines HTML5 Dokuments 2/4

- <article>
 - Artikel oder in sich abgeschlossenes Inhaltselement
- <aside>
 - Sekundäre oder ähnliche Inhalte, z.B. Erläuterungen
- <meter>
 - Wert innerhalb eines Wertebereichs
- <progress>
 - Fortschrittsanzeige, evtl. nicht unterstützt

Typische Struktur eines HTML5 Dokuments 3/4



Typische Struktur eines HTML5 Dokuments 4/4

- HTML5 wird als vereinfachter XML DOCTYPE definiert:

```
<!DOCTYPE html>
<html lang=„de-DE>
...
</html>
```

Beispiel: Kontodaten ändern mit editierbarem Text



... der Quellcode dazu ...

```
<body id="forms"><h1>Benutzerdaten</h1>
<section id="edit_profile_link"><p><a href="">Konto ändern</a></p></
section><div id="status"></div>
<ul>
    <li><b>Name</b><span id="name" contenteditable="true">Judt</
span></li>
    <li><b>Straße</b><span id="strasse"
contenteditable="true">Fallenbrunnen</span></li> <li><b>Ort</
b><span id="ort" contenteditable="true">FN</span></li>
    <li><b>PLZ</b><span id="plz" contenteditable="true">88045</
span></li>      <li><b>Email</b><span id="email"
contenteditable="true">judt@dhw-ravensburg.de</span></li> </ul>
...</body>
```

Medientypische Benutzerschnittstellen mit CSS3 Media Queries 1/2

- Bereits mit JavaScript waren auflösungsspezifische Darstellungen möglich.
- CSS3 bietet die Ermittlung von Informationen für das Style Sheet:
 - Auflösung
 - Ausrichtung (hoch/quer)
 - Breite/Höhe des Geräts
 - Breite und Höhe des Browserfensters

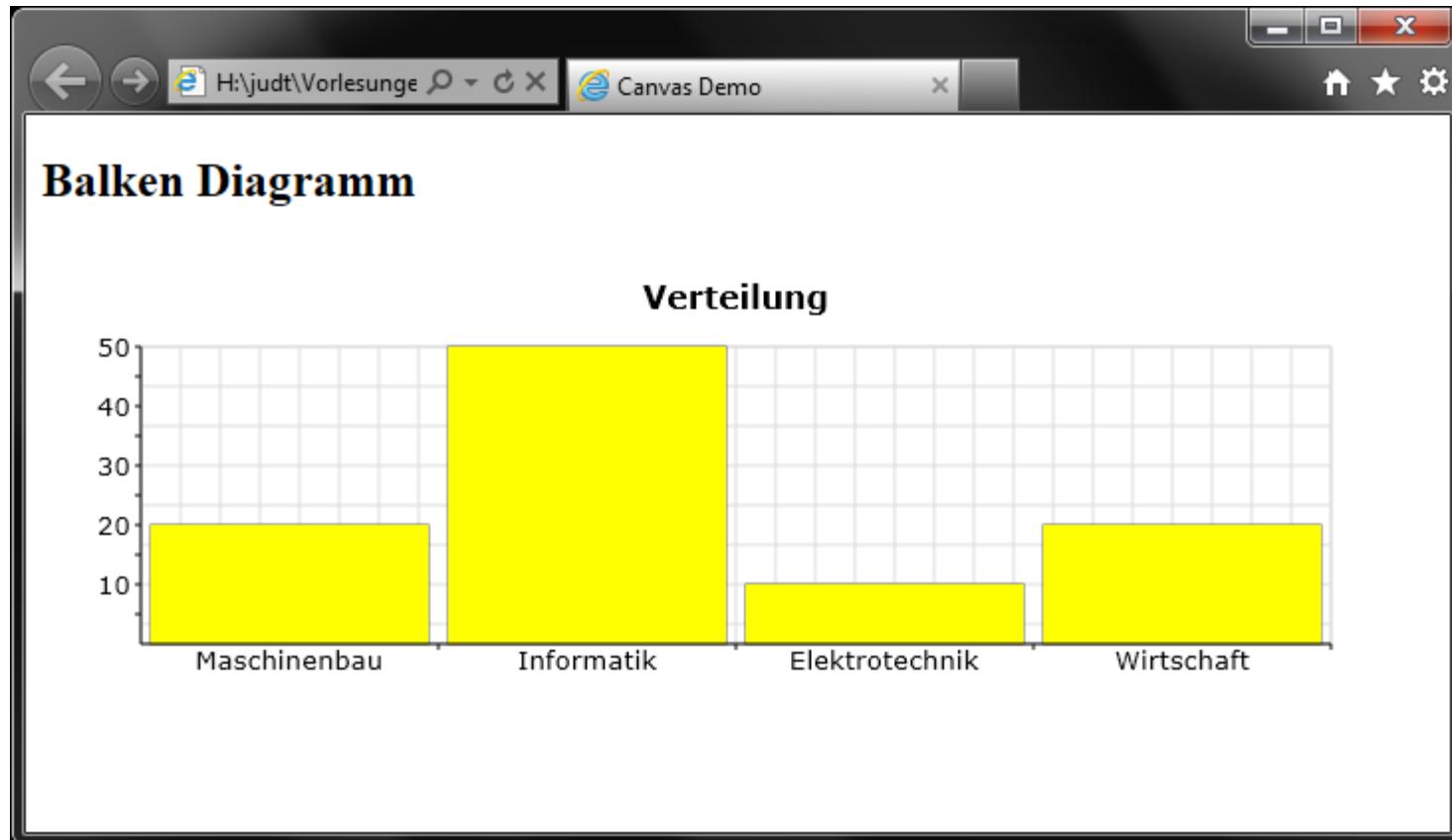
Medientypische Benutzerschnittstellen mit CSS3 Media Queries 2/2

- Der speziell für mobile Geräte definierte Medientyp „handheld“ wird oft von Smartphones ignoriert, um Webseiten selbst besser darstellen zu können.
 - Besser wird ein eigener Style an eine maximale Auflösung gekoppelt.

Zeichnen mit <canvas>

- <canvas> ermöglicht die Darstellung von vektorbasierten Grafiken mit JavaScript.
 - <canvas> kann auch Audiodateien abspielen
 - <canvas> ist vielmehr eine Programmierschnittstelle als ein Tag.

Beispiel: Balkendiagramm in <canvas>



Neue lokale Datenspeicher

- localStorage
 - Speichert Daten in Schlüssel/Wert-Paaren, mit einer Domain verknüpft und über die Browsersitzung haltbar.
 - Vergleichbar mit Cookies
- sessionStorage
 - Speichert Daten in Schlüssel/Wert-Paaren, mit einer Domain verknüpft und zur Laufzeit des Browsers existent.
- Web SQL Databases
 - Vollständig relationale Datenbanken mit Transaktionsmechanismus, über die Browsersitzung haltbar.
- Offline Webanwendungen
 - Definition von Dateien, die für Offline-Verwendung im Cache gespeichert werden.

Konkurrenz belebt das Geschäft – und verhindert Standardisierung

- Bisherige Implementierungen der Browserhersteller weichen im Funktionsumfang stark voneinander ab.
 - Oft sind Fallunterscheidungen erforderlich.
 - Viele Funktionalitäten kommen aus Funktionsbibliotheken, deren Zukunft ungewiss ist.
- Die Strategien der Browserhersteller sind ungewiss.
 - Möglicherweise wird es wieder Machtkämpfe der Hersteller geben.
- Fazit: für Unternehmen ist der kommerzielle Einsatz von HTML5 bisher unrealistisch.

Übung: eine browser-lokale Applikation mit HTML5

- Für eine Klausurauswertung sollen Namenslisten, Aufgaben und die von den Teilnehmern erreichten Punkte lokal in einer Web-Anwendung gespeichert werden.
 - Implementieren Sie hierfür in geeigneter Gruppenaufteilung Teillösungen, die Sie am Ende der Übung zu einer Gesamtlösung zusammensetzen.
- Hinweise
 - Verwenden Sie evtl. Google Chrome als geeignete Implementierung oder ein geeignete Plugin.
 - Prüfen Sie die Verwendbarkeit der jQuery Bibliothek für Ihre Lösung.

Selbstkontrolle

1. Für welche Anwendung wurde HTML ursprünglich entwickelt?
2. Wie hängen HTML und XML zusammen?
3. Warum sind HTML-Dokumente oft nicht fehlerfrei?
4. Was unterscheidet localStorage und sessionStorage?
5. Warum ist der Einsatz einer lokalen SQL-Datebank kritisch?
6. Warum ist HTML5 nicht unbedingt für Unternehmensanwendungen geeignet?

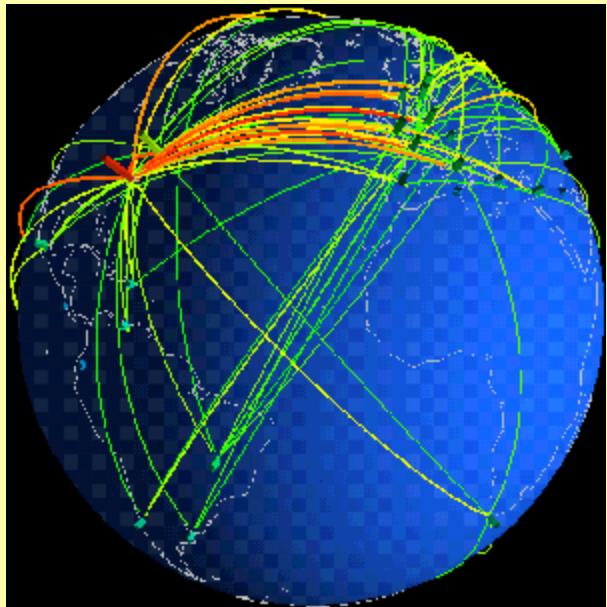
Kontakt

Duale Hochschule BW Ravensburg
Campus Friedrichshafen

Prof. Dr. Andreas Judt
Informationstechnik

judt@dhbw-ravensburg.de

Web Engineering Applets



Prof. Dr. Andreas Judt

Was sind Applets?

- Kleine Java-Applikationen, die innerhalb einer Webseite dargestellt werden
 - alternativ: Sun Applet Viewer, ist in J2SDK enthalten
- Applets wurden entwickelt, um Webseiten dynamischer zu gestalten.
 - Kommunikation mit dem umfassenden HTML-Dokument möglich, z.B. mit JavaScript-Funktionen

Applets heute

- kaum praktische Relevanz aufgrund angeborener technischer Probleme
 - Vielfalt der Browser erzeugt extrem hohe Aufwände
- Applets werden heute als multimediales Plugin ähnlich wie Flash behandelt.
- häufigste Anwendung
 - Java-Installationen über das Intranet/Internet
 - Geo-Applikationen
- Bemerkung:
 - Applets können auch unsichtbar sein!

Applets ursprünglich

- Applet-Tag legt einen Bereich in einer HTML-Seite fest, in dem ein Applet angezeigt wird.
 - WIDTH, HEIGHT: definieren die Größe des Applets auf der Seite

```
<applet code=„MyApplet“ archive="applet.jar,"  
        width="200" height="100">  
    Java required!  
</applet>  
<applet code=„My2ndApplet“ codebase="path"  
        width="200" height="100">  
    Java required!  
</applet>
```

Parameter

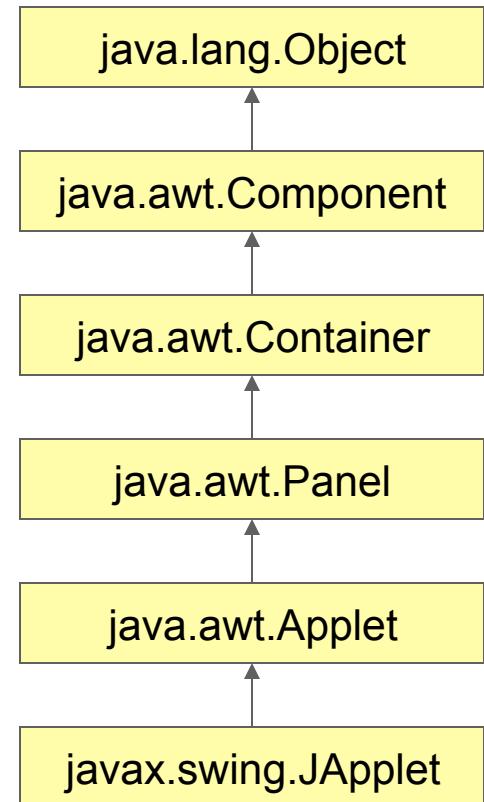
- Parameter werden an das Applets mit dem Tag `<param>` in `<applet></applet>` übergeben:
 - `<param name="name" value="Judt">`
 - `<param name="vorname" value="Andreas">`
- Parameter benutzen: Methode `getParameter()` in Applet-Klasse
- `java.applet.Applet` ist eine Kindklasse von `java.awt.Panel`

Ein Beispiel: map24.de

The screenshot shows the map24.de homepage with a search bar for "friedrichshafen, seestraße". The main content area displays a map of Friedrichshafen with several numbered points of interest (1, 2, 3, 4) and a red circle highlighting a location near the B30/B31/E54 intersection. The map includes labels for streets like Europa, Wendelgardstrasse, Marienstrasse, Charlottenstrasse, Bismarckstrasse, Ailinger Strasse, Eckenstrasse, and Weiderstrasse. A sidebar on the left lists "Themen in Friedrichshafen", "Map24 Rubriken", and "Nachrichten". Below the map, there's a "Suchergebnisse" section with a sponsored link for Europcar Autovermietung and a "Hotels in Friedrichshafen suchen und buchen Neu!" button. At the bottom, there's a "Immobilien in Friedrichshafen" section and footer links for Kontakt, Werbung, AGB, Presse, Newsletter, and Jobs.

Applets mit Swing

- Die Swing-Version des Applets: JApplet bietet grafische Benutzerschnittstellen mit Swing
- erforderlich: Browser mit J2SE JVM
- allgemein:
 - eigene Klassen müssen entweder über den CLASSPATH der JVM im Browser oder von einem Web Server geladen werden



Lebenszyklus

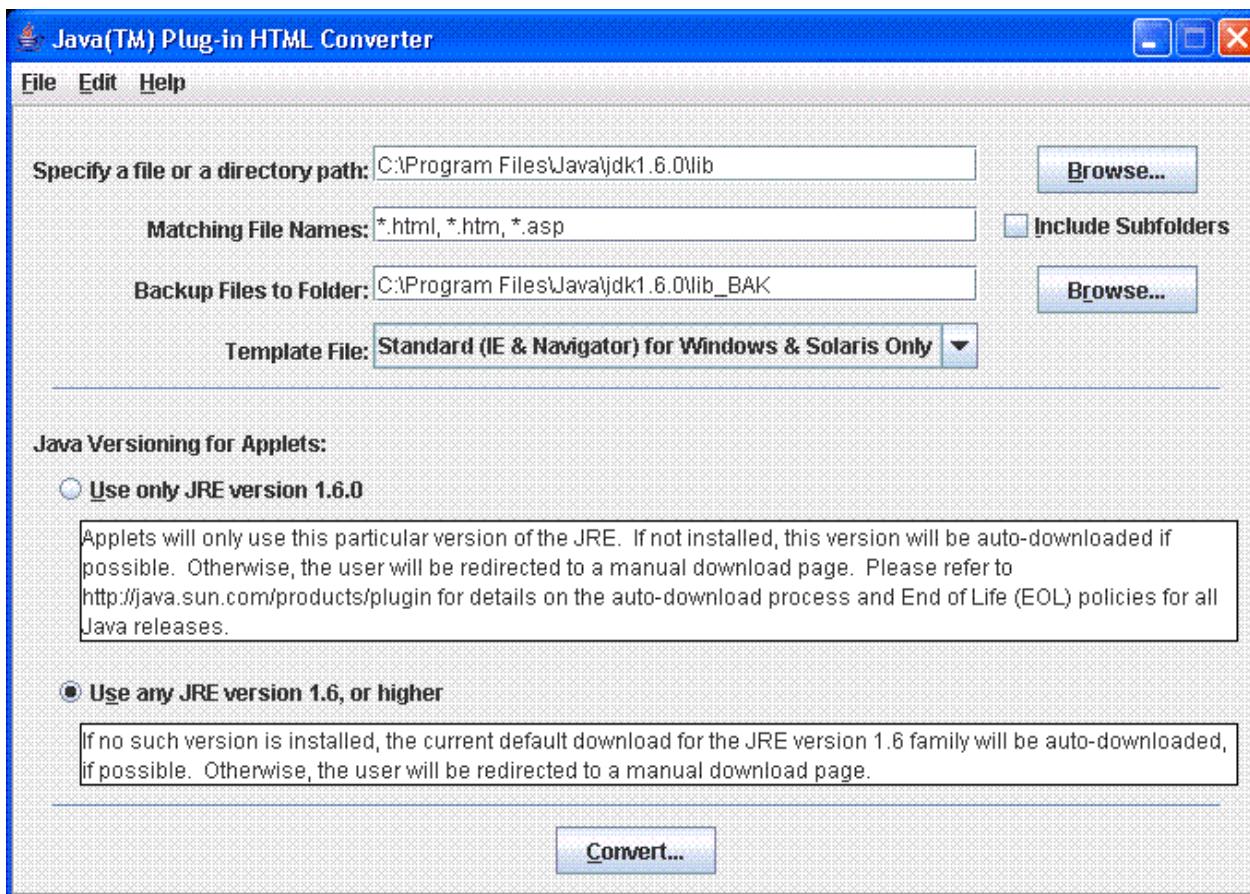
- **init()**
 - wird aufgerufen, wenn das Applet zum ersten Mal geladen wird
- **start()**
 - wird aufgerufen, wenn die Seite dargestellt wird
- **stop()**
 - wird aufgerufen, wenn die Seite verlassen oder entfernt wird
- **destroy()**
 - wird aufgerufen, wenn der Browser die Seite aus dem Cache entfernt

Applets in verschiedenen Browsern

- Applets müssen browserspezifisch integriert werden!
 - Lösung: J2SDK HTML Konverter
- Beispiel J2SDK 6.0:

```
<html>
<script language="Javascript">
  var _app = navigator.appName;
  if (_app == 'Mozilla') {
    document.write('<embed code="Applet1.class"', 'width="200"',
      'height="200"', 'type="application/x-java-applet;version=1.6">');
  } else if (_app == 'Microsoft Internet Explorer') {
    document.write('<OBJECT ', 'classid="',
      "clsid:8AD9C840-044E-11D1-B3E9-00805F499D93"',
      'width="200"', 'height="200">', '<PARAM name="code"',
      'value="Applet1.class">', '</OBJECT>');
  } else {
    document.write('<p>Sorry, unsupported browser.</p>'); }
</script>
</html>
```

J2SDK HTML Konverter



Aktueller Stand bei der Verwendung von Applets

- Moderne Browser, die HTML5 unterstützen, erlauben wieder die Verwendung des Tags <applet>.
- Die verwendete Java-Version spielt dabei keine Rolle.
- Das Oracle JDK installiert optional viele Beispiele zur Verwendung von Applets.

Sicherheit

- Jeder ByteCode wird vor Ausführung verifiziert.
- Der Klassenlader lädt den ByteCode, der ausgeführt werden soll.
- Es ist sichergestellt, dass bei Existenz von zwei Implementierungen diejenige geladen wird, die „sicherer“ ist.
- Beispiel: es kann niemand von außen eine modifizierte IO-Bibliothek laden, da die lokal installierte für sicherer gehalten wird.

Sicherheit

- Java versucht zu verhindern, dass Applets den Rechner des Benutzers unzulässig beeinflussen.
- Daher ist nicht alles erlaubt:
 - File-IO ist verboten und muss vom Benutzer explizit erlaubt werden.
 - Stichwort: security policy
 - Nur Netzverbindungen zum Host, von dem das Applet stammt, sind erlaubt.
 - Native Methoden sind nicht erlaubt.

Java Sandkasten (engl. sandbox)

- Durch die Einführung der Java 2 Plattform mit der JDK-Version 1.2 hat Sun das Sandkasten-Prinzip für Applets auf beliebige Applikationen erweitert und erlaubt auch bei Applets die gezielte Freigabe einzelner Rechte.
- Mit Policy-Dateien können jetzt einfach Rechte für bestimmte Klassen, signierte Java-Archive und CodeBase-Adressen vergeben werden.
- Die systemweite Policy-Datei im lib/security Verzeichnis des Runtime Environment kann durch eine lokale Datei erweitert werden (policytool).

Signierte Applets

- Applets Archive (JAR) können digital signiert werden. Damit akzeptieren Browser Applets mit Zugriffen außerhalb der Sandkiste, ohne eine Sicherheitswarnung zu erzeugen.
- Zum Erzeugen von signierten Applet-JARs bietet das J2SDK ein zusätzliches Tool: jarsigner
- Für die Verwaltung von Zertifikaten: keytool
 - browser-akzeptierte Signaturen müssen üblicherweise gekauft werden!
 - Mit der Anwendung keytool kann der Entwickler eines Applets leicht ein eigenes Schlüsselpaar für die Verschlüsselung und die digitale Signatur mit RSA erzeugen.
`keytool -genkey -alias <alias> -keystore <keystore>`

Java Web Start

- Weiterentwicklung der Applets (im Browser)
 - Verteilung ganzer Applikationen übers Netzwerk
 - Java Network Launcher Protocol (JNLP)
- Installation des Basissystems mit dem JRE
- Start über einen Link auf einer Web-Seite
- Web Start Descriptor (JNLP-Datei)
- Update übers Netzwerk, basiert auf einer Konfigurations-Datei
- Unterstützung von Offline-Ausführung (Caching)
- JNLP Datei
 - XML Deskriptor
 - vergleichbar mit einer ausführbaren Datei
 - Sicherheit wird über Policy-Einstellungen eingestellt
 - JNLP API interpretiert die Webstart Konfigurationsdatei

Übung: Interaktion zwischen Webseite und Applet

- Implementieren Sie eine Webseite mit eingebettetem Applet. Die Webseite soll eine Adresse aufnehmen.
- Ihre Aufgabe:
 - Recherchieren und implementieren Sie Möglichkeiten, die vom Anwender eingegebene Adresse in das Applet zu übernehmen und aus dem Applet wieder in das Formular zurückzuschreiben.
 - Der Abgleich soll entweder mit einem Knopf in der Webseite oder einem Knopf im Applet erfolgen.

Selbstkontrolle

1. Wie werden Applets in Webseiten eingebettet?
2. Welche Aufgabe hat der HTML Konverter?
3. Welche Möglichkeiten gibt es zur Interaktion zwischen Webseite und Applet?
4. Gibt es Applets ohne grafische Benutzerschnittstelle?

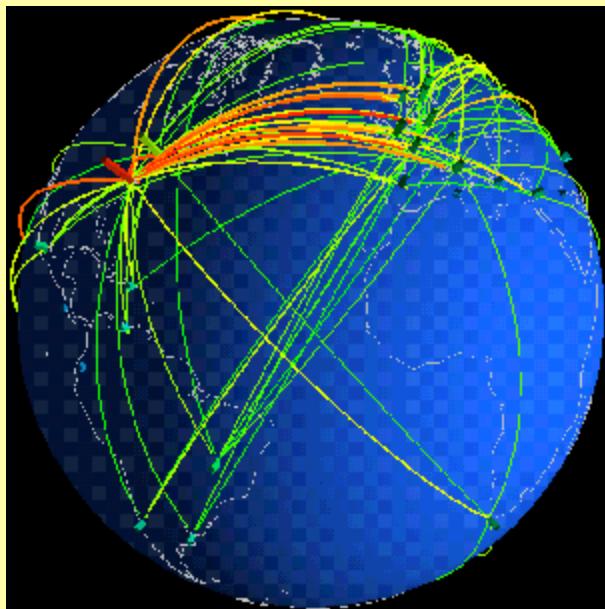
Kontakt

Berufsakademie Ravensburg
Außenstelle Friedrichshafen

Prof. Dr. Andreas Judt
Informationstechnik

judt@ba-ravensburg.de

Web Engineering CGI und Perl

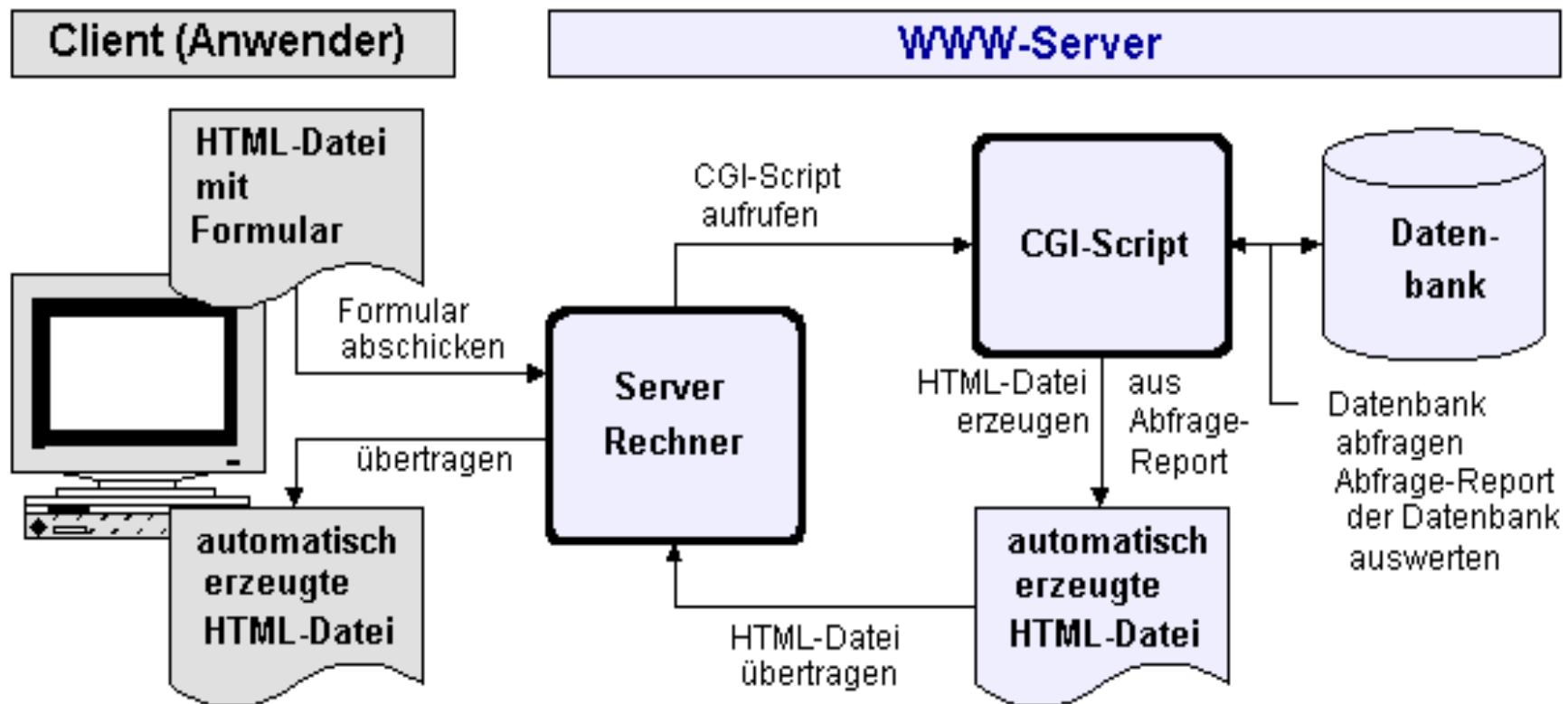


Prof. Dr. Andreas Judt

Common Gateway Interface (CGI)

- Das Common Gateway Interface (CGI) ermöglicht den Entwurf von interaktiven, benutzergesteuerten Web-Applikationen.
 - erste Technologie für nicht-statische Informationen im Internet
- Der WWW-Server ruft auf Serverseite ein CGI-Programm auf und liefert die Ausgabe (als HTML oder anderen MIME-Typ) zurück an den Browser.
 - MIME = multipurpose internet mail extension
- Parameterübergabe erfolgt mit HTML-Formularen oder in der URL.
 - URL = uniform ressource locator
- Ein CGI-Skript ist ein ‘normales’ Programm.
- Mit CGI können Inhalte dynamisch erzeugt werden.
 - z.B. aus einer Datenbank

Interaktion zwischen Browser und Web-Server



Datenübertragung über HTTP

- HTTP = hyper text transfer protocol
- GET(URL):
 - liefert Daten einer URL zurück
 - Parameter werden in der URL übertragen
 - kann abgehört werden
 - Standard
- POST(URL):
 - liefert Daten einer URL zurück
 - Parameter werden getrennt übertragen (und verschlüsselt, z.B. HTTPS)
 - kann schwerer abgehört werden: IP Sniffing

Aufruf von CGI-Programmen

- GET
 - Parameterübergabe in einem String am Ende der URL, die das CGI-Script liest (mit ? getrennt). Der String wird in einer Umgebungsvariable (QUERY_STRING) gespeichert. Beispiel:
 - CGI Programm
- POST
 - Formulardaten werden in der Standardeingabe an das CGI-Script übertragen. Beispiel:
 - <FORM ACTION="/cgi-bin/program.pl" METHOD="post">

CGI Umgebungsvariable

- GATEWAY_INTERFACE
- SERVER_NAME
 - Hostname oder IP-Adresse des Servers.
- SERVER_SOFTWARE, SERVER_PROTOCOL, SERVER_PORT
- REQUEST_METHOD
 - Methode mit der das Skript aufgerufen wurde (GET, POST).
- PATH_INFO
 - Genauere Pfadinformation. So ist möglich dem Skript einen Pfad zu übergeben, der hinter der URL des Skripts steht.
 - Beispiel:

`http://www.host.de/cgi-bin/script.cgi/pfad/info/hier`
setzt PATH_INFO auf /pfad/info/hier

CGI Umgebungsvariable

- DOCUMENT_ROOT
 - Verzeichnis der HTML-Dokumente auf dem Server.
- PATH_TRANSLATED
 - enthält Pfadangaben entsprechend der Verzeichnisstruktur auf dem Server-Rechner wenn die GET-Methode verwendet wird.
- SCRIPT_NAME
 - lokale URL des aufgerufenen Skripts.
 - Beispiel: /cgi-bin/script.cgi
- QUERY_STRING
 - Bei GET-Methode eine Zeichenkette mit Daten, die dem Skript übergeben werden, z.B. Formulardaten.
- REMOTE_HOST, REMOTE_ADDR

CGI Umgebungsvariable

- AUTH_TYPE
 - Authentifizierungsmethode, um Anwender zu identifizieren.
- REMOTE_USER
 - Benutzername, des Aufrufenden (nur bei eingeschalteter wenn Server-Authentifizierung).
- REMOTE_IDENT
 - Protokollinformationen, wenn auf dem Server das Protokoll ident für geschützte Zugriffe läuft.
- CONTENT_TYPE
 - MIME-Typ der Anfragedaten (z.B. “text/html“).
- CONTENT_LENGTH
 - Anzahl der Datenbytes (zum Lesen aus STDIN bei POST-Methode).

CGI Umgebungsvariable

- **HTTP_FROM**
 - e-Mail-Adresse des Benutzers (nicht alle Browser).
- **HTTP_ACCEPT**
 - Liste der MIME-Typen, die vom Klienten unterstützt werden.
- **HTTP_USER_AGENT**
 - Browsertyp (“Mozilla???“).
 - Vorsicht bei Falschaussagen!
 - z.B. Opera oder frühe Versionen von Microsoft Internet Explorer
- **HTTP_REFERER**
 - URL der letzten besuchten Seite (nicht alle Browser).

Perl

- Perl steht für Practical Extraction and Report Language und wurde 1987 für die Unix-Welt von Larry Wall entwickelt.
- Perl ist eine offene Sprache und durch Erweiterungen ab der Version 5.0 unterstützt Perl z.B. auch den Ansatz der objekt-orientierten Programmierung.
- Perl gilt als Hacker-Sprache, als Sprache, in der man unglaubliche Dinge in einer einzigen Programmanweisung erledigen kann. Perl-Skripte werden interpretiert.
- Besonders die Mächtigkeit des Perl-Interpreters bei der Verarbeitung von Strings in Verbindung mit regulären Ausdrücken hat sie als Lieblingswerkzeug von CGI-Programmierern im World Wide Web etabliert.

Ein Perl-Beispiel

```
if ($ENV{ 'REQUEST_METHOD '} eq 'GET ') {  
    $Daten = $ENV{ 'QUERY_STRING '};  
} else {  
    read(STDIN, $Daten, $ENV{ 'CONTENT_LENGTH '});  
}  
@Formularfelder = split(/&/, $Daten);  
foreach $Feld (@Formularfelder) {  
    ($name, $value) = split(/=/, $Feld);  
    $value =~ tr/+/ /;  
    $value =~ s/%([a-fA-F0-9][a-fA-F0-9])/pack(„C“, hex($1))/eg;  
    $value =~ s/<!—(.|\n)*->//g;  
    $Formular[$i] = $name;  
    $i = $i + 1;  
    $Formular[$i] = $value;  
    $i = $i + 1;  
}
```

Erläuterung des Beispiels

- Der Perl-Code prüft zuerst, ob das HTML-Formular die Methode GET oder POST anwendet, und liest abhängig davon die übergebenen Formulardaten.
- \$Daten enthält dann die Formulardaten, in HTTP-codierter Darstellung.
- Mit split(/&/, \$Daten) werden die einzelnen Attribut-Wert-Paare in das Array @Formularfelder gespeichert.
- Anschließend werden die einzelnen Felder bearbeitet, dekodiert und in der Liste \$Formular gespeichert, mit \$Formular[0], \$Formular[2], ... als Feldname und \$Formular[1], \$Formular[3], ... als Wert.
- Diese Liste kann im weiteren Verlauf des Perl-Skripts weiterverarbeitet werden.

HTML Ausgabe erzeugen

```
print "Content-type: text/html\n\n";
print "<html>\n<head>\n";
print "<title>Danke!</TITLE>\n";
print "</head><body bgcolor=\"#FFFFFF\" text=\"#000000\" .
"link=\"#009900\" vlink=\"#006600\" alink=\"#0000FF\">\n";
print "<H1>Danke</H1>\nWir freuen uns, da&szlig; Sie unser " .
"Formular ausgef&uuml;llt haben.\n";
print "Hier zur Kontrolle Ihre Angaben:<p>\n";
for($i=0; $i<=$max; $i=$i+2) {
    print "<b>$Formular[$i]:</b> $Formular[$i+1]<br>\n";
}
print "<p><hr noshade size=1><p>";
print "<i>Ihr Name, <a href=\"mailto:IhrName@Provider.xx\">" .
"IhrName@Provider.xx</a></i>";
print "</body>\n";
print "</html>\n";
```

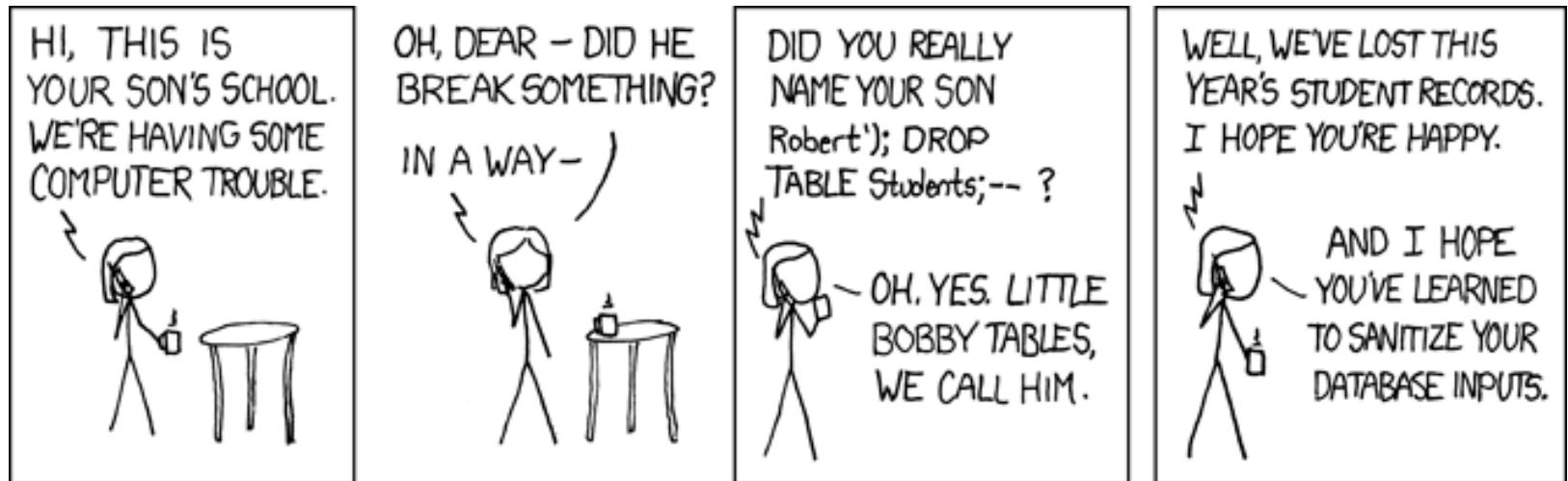
Erläuterung zur HTML Ausgabe

- Der Wechsel zwischen HTTP-Kopf und HTML-Daten werden in der Perlausgabe an die Standardausgabe durch eine leere Zeile (\n\n) markiert.
- Außer dem Content-Type können im Kopfteil (engl. header) weitere Felder definiert werden.
 - RFC für HTTP/1.1: <http://www.ietf.org/rfc/rfc2616.txt>

Perl und Lesbarkeit: ein Gedicht

'Love was'
&& 'love will be' if
(I, ever-faithful),
do wait, patiently;
"negative", "wordly", values disappear,
@last, 'love triumphs';
join (hands,checkbooks),
pop champagne-corks,
"live happily-ever-after".
"not so" ?
tell me: "I listen",
(do-not-hear);
push (rush, hurry) && die lonely if not-careful;
"I will wait."
&wait
Sharon Hopkins, June 26th, 1991
rush (a perl poem)

Comic: SQL Injection



Bewertung

- Bei CGI wird für jedes Programm (oder Skript) ein eigener Prozess bzw. Interpretierer gestartet.
- Austausch von Informationen oder serverseitige Speicherung von temporären Daten ist schwierig.
 - z.B. ein Warenkorb
- Fazit:
 - CGI ist kompliziert
 - CGI ist nicht skalierbar
 - CGI erlaubt oft unkontrollierten Zugriff auf das unterliegende Betriebssystem
- Ansätze zur Weiterentwicklung:
 - fastCGI, Web Sockets, Node.js
 - PHP, Python
 - Servlets, Java Server Pages, Web Services

Übung: Python-basierter Web-Server

- Entwickeln Sie einen Python-basierten Web Server, der den Zustand eines Algorithmus (z.B. ein Zähler) im Prozess des Web-Servers aus einer lokalen Variable zwischen den Aufrufen einer URL speichert und wiedergeben kann. Führen Sie Ihre Lösung anhand einer Beispielimplementierung vor.
- Verwenden Sie dabei keine Cookies, Sessions oder außerhalb von Python liegende Speichertechnologien.
- Eine Web-Server API finden Sie im Standardumfang von Python.

Selbstkontrolle

1. Warum werden dynamische Webseiten benötigt und warum ist es erforderlich, dass diese auf der Serverseite rechnen?
2. Warum sind CGI-Programme nicht für den Hochlastbereich geeignet?
3. Können CGI-Programme aus Applets aufgerufen werden?
4. Wie werden aus herkömmlichen Programmiersprachen CGI-fähige gemacht?
5. Welchen Zweck erfüllen die CGI-Variable?

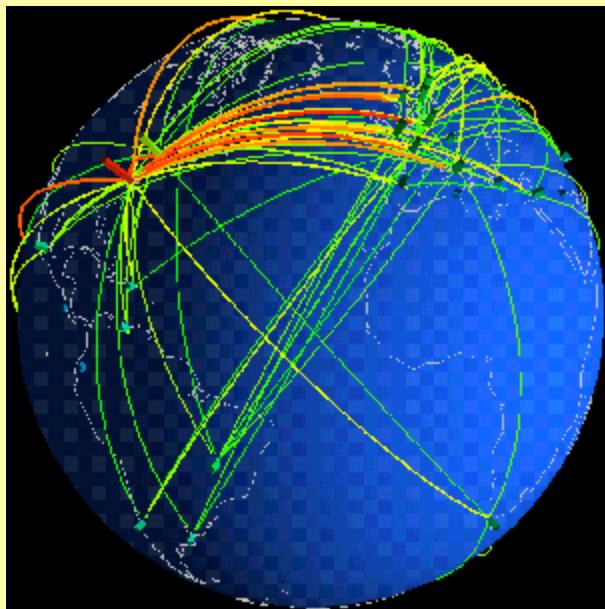
Kontakt

Duale Hochschule BW Ravensburg
Campus Friedrichshafen

Prof. Dr. Andreas Judt
Informationstechnik

judt@dhbw-ravensburg.de

Web Engineering Servlets



Prof. Dr. Andreas Judt

Was ist ein Servlet?

- Servlet:
 - Java Komponente
 - läuft im Web-Server ab
 - erweitert Funktionalität des Web-Servers
 - kann dynamisch Web-Seiten erstellen
 - ist transparent für den Benutzer

Servlet und Applet im Vergleich

Servlet

- Java Komponente
- plattformunabhängig
- läuft auf Server-Seite (im Web-Server)
- bleibt auf dem Web-Server
- stellt Dienste zur Verfügung
- hat keine direkte Benutzerschnittstelle
- hat URL

Applet

- Java Komponente
- plattformunabhängig
- läuft auf Client-Seite (im Browser)
- wird auf Client geladen
- benutzt Dienste, präsentiert Ergebnisse
- implementiert Benutzerschnittstelle

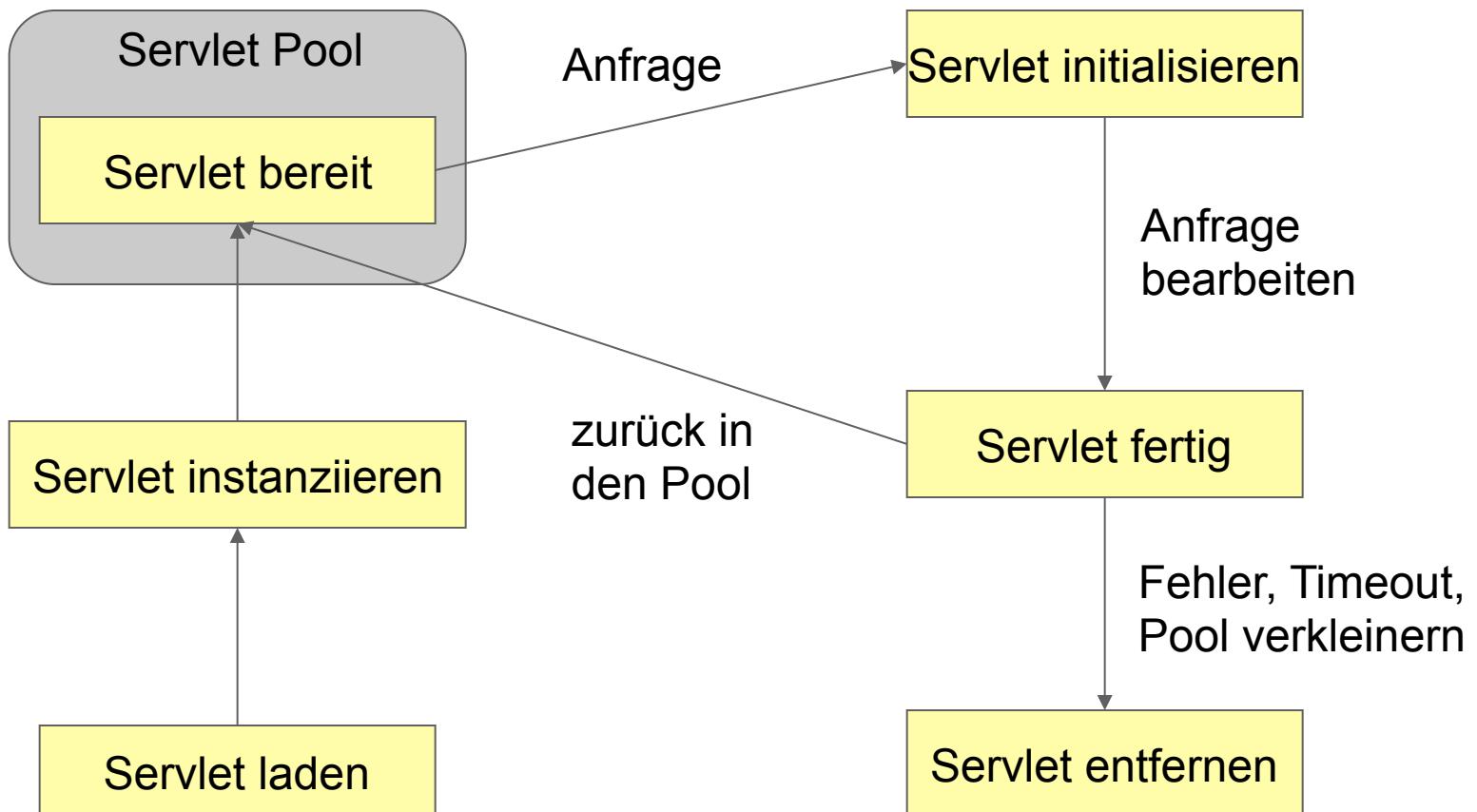
Warum Servlets einsetzen?

- Servlets
 - sind schneller als CGI-Skripts
 - besitzen ein anderes Prozessmodell
 - Servlet-API von wird heute von vielen Servern unterstützt
 - werden in Java entwickelt
 - können alle APIs von Java benutzen

Wie funktioniert ein Servlet?

- ein Servlet:
 - ist eine Java Klasse
 - wird nach bestimmten Richtlinien programmiert
 - wird vom Programmierer übersetzt und im Web-Server installiert (engl. deployment)
 - als sog. Webarchiv (WAR)
 - kann von jedem Browser angesprochen werden

Lebenszyklus



Servlet Methoden

- **init(...):**
 - wird vom Server aufgerufen, wenn das Servlet geladen wird
 - Servlet kann benötigte Ressourcen belegen
 - Servlet erhält ein ServletConfig-Objekt
- **service(ServletRequest, ServletResponse):**
 - Aufruf der Geschäftslogik
- **destroy():**
 - wird vom Server aufgerufen, wenn Servlet gelöscht wird
 - Servlet kann belegte Ressourcen freigeben

HTTP Servlet

- Web-Server:
 - ruft Methoden des Servlets auf, wenn passende URL verlangt wird
- die Klasse HttpServlet besitzt eine Standard-Implementierungen und spezialisiert die Klasse Servlet
- Bearbeitung der http Anfrage::
 - doGet() für GET-Operationen
 - doPost() für POST-Operationen
 - Parameter:
 - HttpServletRequest: die http Anfrage als Objekt
 - HttpServletResponse: Objekt, das das Ergebnisdokument enthält
 - Web-Server erzeugt Anforderungs- und Ergebnisobjekt selbständig

Beispiel

```
public class MyServlet extends HttpServlet {  
  
    public void init(ServletConfig cfg) throws ServletException {  
        // Allokation von Ressourcen  
    }  
    public void doGet(HttpServletRequest req, HttpServletResponse res)  
        throws ServletException, IOException {  
        res.setContentType("text/html");  
        PrintWriter out = new PrintWriter(res.getOutputStream());  
        out.println("<p>Das ist ein Servlet</p>");  
        out.println("<p>Die aktuelle Uhrzeit ist: " + new Date() + "</p>");  
        out.close();  
    }  
    public void doPost( ... ) { doGet(req, res); }  
}
```

Anfrage und Ergebnis

- **HttpServletRequest:**
 - enthält Informationen über Anforderung
 - Parameter, Rechner, Benutzer, Cookies, etc.
 - enthält eine Sitzung (engl. session) als Objekt
 - z.B. einen Warenkorb
- **HttpServletResponse:**
 - empfängt Ausgaben des Servlets
 - kann Cookies setzen
 - enthält eine Sitzung als Objekt

Anfrage-Objekt

- **HttpServletRequest:**
 - kann Parameter enthalten
 - aus URL (z.B. .../MySrv?name=judt&ort=friedrichshafen)
 - aus POST Datenblock, z.B. aus einem HTML-Formular
 - Parameter nicht vorhanden: Ergebnis == null

```
public void doGet(...) {  
    String name = req.getParameter("name");  
    String ort = req.getParameter("ort");  
    ...  
}
```

Ergebnis-Objekt

- **HttpServletResponse:**
 - enthält OutputStream
 - Datenstrom (engl. stream) kann beliebige Daten aufnehmen
 - MIME-Typ muss angegeben werden
 - andere (bequemere) Datenströme können benutzt werden
 - Stichwort: Kaskade von Datenströmen in Java

Übung

- Programmieren Sie ein Servlet, das http-GET unterstützt und alle in der URL übergebenen Parameter und deren Werte ausgibt. Informieren Sie sich über die Parameter-Methoden der Klasse ServletRequest.
- Tipp: Benutzen Sie für Zeilenumbrüche:
`out.println("<p>" + irgendwas + "</p>");`
- Achten Sie auf Null-Werte.

Sitzungen

- http-Protokoll ist zustandslos
- Web-Server bietet Sessions an:
 - Sessions werden vom Server verwaltet
 - Sessions können einen Zustand speichern
 - z.B. angemeldeter Benutzer
 - z.B. Warenkorb
 - z.B. Surf-Verhalten

```
HttpSession s =  
req.getSession(true);  
  
if (s.isNew()) { // neue Session  
Vector v = new Vector();  
s.setAttribute("Namen", v);  
  
}  
  
Vector v = (Vector)s.getAttribute(  
"Namen");  
}
```

Sitzungen

- Server verwendet verschiedene Methoden:
 - Cookies
 - geänderte URLs
 - transparent für Servlet und Browser

```
public void doGet(...) {  
    // Wenn URLs angegeben  
    // werden sollen, kann so die  
    // session-id einbezogen werden  
  
    String neueUrl =  
        res.encodeUrl(req.getRequest  
        URI());  
  
    out.println(... + neueUrl + ...);
```

Sitzungen

- Sessions beenden
 - nach Gebrauch Session beenden
 - Objekte in Session werden gelöscht

```
public void logout(...) {  
    // Session beenden  
    // Daten werden gelöscht  
  
    session.invalidate();  
}
```

HTML Formulare

- Servlets können Daten aus Formularen bearbeiten
 - Formulare können Parameter erfragen
 - Parameter werden an das Servlet übergeben
 - Vorsicht: Reihenfolge unklar bei Mehrfachauswahlen
 - bei nicht ausgefüllten Feldern wird ggf. kein Parameter übergeben
 - Parameterübergabe:
 - GET oder POST

Web Archive

- Seit der Einführung der Servlet 2.2 API werden Servlets, Java Server Pages und andere Web-Applikationen in einem speziellen JAR Archiv (engl. web application archive, WAR) gespeichert und die dynamischen Komponenten der Web-Anwendung werden mit /WEB-INF/web.xml genauer beschrieben.
- Die Datei web.xml enthält die Konfiguration der enthaltenen Servlets.
 - z.B. Abbildung Servlet \Leftrightarrow URL
- Neben diesen Konfigurationen enthält eine WAR-Datei beliebige Dateien.
- Klassen und Java-Pakete müssen sich unterhalb von /WEB-INF/classes/ befinden.
- WAR Dateien werden im Projekt automatisch erzeugt (z.B. Eclipse)

Web Archive

- Struktur eines Web Archivs
 - HTML und JSP auf oberster Ebene
 - Bilder im Verzeichnis /images
 - Deskriptoren in /WEB-INF
 - Klassen in /WEB-INF/classes
- Beispiel
 - /index.html
 - /login.jsp
 - /images/mylogo.gif
 - /WEB-INF/web.xml
 - /WEB-INF/classes/mypackage/MyHelpClass.class
 - /WEB-INF/classes/MyPackage/MyServlet.class
 - /WEB-INF/lib/mylibrary.jar
 - /META-INF/MANIFEST.MF

Request Header

- Mit Request Headers kann der Browser Zusatzinformationen über den Request an den Server schicken.
- Der Server kann seine Reaktionen an die Zusatzinformationen anpassen.
- Request Headers sind optional, außer der CONTENT_LENGTH (Länge der mittels POST-Methode übertragenen Daten).
 - getHeaderNames() liefert Aufzählung der verwendeten Header
 - getHeader() liefert den String-Wert des jeweiligen Headers.
- Manche Header habe spezielle Zugriffsmethoden:
 - getCookies() nähere Infos später!
 - getAuthType(), getRemoteUser() zerlegt den Authorization-Header in seine Bestandteile.

Übung

- Erstellen Sie ein Servlet, das die Request Header anzeigt. Verwenden Sie dazu die Methoden `getMethod()`, `getRequestURI()` & `getProtocol()`, sowie die Methoden `getHeaderNames()` und `getHeader()` um tabellarisch darzustellen, welche optionalen RequestHeaders vorkommen.
- Informationen über Request Headers können Sie unter <http://www.w3.org/protocols/> finden.

CGI Variable und Servlets

- CGI-Variablen beinhalten Informationen über den Server und über die vom Browser erhaltene Anfrage.
- Es gibt zwei Kategorien von CGI-Variablen:
 - Variablen, die unabhängig vom jeweiligen Zugriff einen festen Wert haben (server-bezogen)
 - Variablen, deren Wert abhängig ist vom jeweiligen Zugriff ist (client-bezogen)
- Auf CGI-Variable kann in Servlets zugegriffen werden.

CGI Variable und Servlets

AUTH_TYPE	request.getAuthType()
CONTENT_LENGTH	request.getContentLength()
CONTENT_TYPE	request.getContentType()
DOCUMENT_ROOT	getServletContext().getRealPath("/")
PATH_INFO	request.getPathInfo()
PATH_TRANSLATED	request.getPathTranslated()
QUERY_STRING	request.getQueryString()
REMOTE_ADDR	request.getRemoteAddr()
REMOTE_HOST	request.getRemoteHost()
REMOTE_USER	request.getRemoteUser()
REQUEST_METHOD	request.getMethod()
SCRIPT_NAME	request.getServletPath()
SERVER_NAME	request.getServerName()
SERVER_PORT	request.getServerPort()
SERVER_PROTOCOL	request.getProtocol()
SERVER_SOFTWARE	getServletContext().getServerInfo()

Response Header

- Die Antwort eines Web-Servers beinhaltet normalerweise
 - eine Statuszeile
 - ein oder mehrere Response Headers
 - eine leere Zeile
 - das geforderte Dokument
- Response Headers werden verwendet u.a.
 - um Cookies zu spezifizieren
 - letzte Änderungen zu signalisieren
 - den Browser zu veranlassen, die Seite nach einer bestimmten Zeit neu zu laden.

Response Header

- Die Spezifikation des Response Header geht am einfachsten mit der `setHeader()`-Methode aus der Klasse `HttpServletResponse`.
- Parameter der Methode sind:
 - Name des Headers
 - Wert des Headers
- Der Response Header muss gesetzt sein, bevor das Dokument vom Server an den Browser gesendet wird.
 - Dokument abgeschlossen, wenn der `OutputStream` geschlossen wurde.

Methoden der Klasse HttpServletResponse

- addHeader() fügt einen neuen Header ein.
- setContentType() setzt den Content-Type des Headers.
- setContentLength() gibt die Anzahl Bytes an, die gesendet werden sollen.
- setDateHeader() gibt die aktuelle Uhrzeit (GMT) an.
- addCookie() setzt ein Cookie.
- sendRedirect() setzt neue Location, von der der Browser das Dokument laden soll.
- setHeader(Refresh, 10; URL=http://...) bewirkt, dass in 10 Sekunden das Laden der Seite gestartet wird.

Sitzungen speichern und verfolgen

- engl. session tracking
- Es gibt drei verschiedene Lösungsansätze für Session Tracking:
 - Cookies
 - URL Rewriting
 - z.B. wenn Cookies-Speicherung abgeschaltet ist)
 - als HIDDEN deklarierte Felder in HTML-Formularen

Sitzungen mit Cookies

- Die Verwendung von Cookies ist die am weitesten verbreitete Methode zum Session Tracking.
- Cookies speichern Informationen über die aktuelle Sitzung, die im Verlauf der Sitzung gelesen werden können.
- Servlets bieten eine leicht verwendbare Schnittstelle, um Cookies zu bearbeiten.
- überflüssige und empfindliche Informationen nie in Cookies speichern (Kreditkartennummer)!

Sitzungen mit geänderten URL

- engl. URL rewriting
- Bei URL rewriting werden Zusatzinformationen an die URL angehängt. Anhand dieser Informationen kann der Server die aktuelle Sitzung und die damit verbundenen Daten identifizieren.

Sitzungen in Formularen

- HTML-Formulare können als HIDDEN deklarierte Felder enthalten, Beispiel:
`<INPUT TYPE="hidden" NAME="key" VALUE="xy">`
- Diese Felder können zum Speichern von sitzungs-spezifischen Daten verwendet werden.
 - Vorsicht: Daten erscheinen in Quelltext!
- Die Daten werden per POST oder GET Methode vom Klienten an den Server übermittelt.
- Verwendbar nur, wenn die Web-Seiten dynamisch erzeugt werden.

Sitzungen im Servlet bearbeiten

- HttpSession API in Servlets:
 - verwendet Cookies oder URL rewriting
 - benötigte Infos werden automatisch behandelt
- Vorgehensweise bei der Verwendung der API:
 - überprüfen, ob ein Objekt zur aktuellen Sitzung existiert
 - Speichern der benötigten Informationen im Sitzungsobjekt
 - Löschen der verlassenen oder alten Sitzungen

Sitzungen im Servlet bearbeiten

- Überprüfen, ob ein Objekt zur aktuellen Sitzung existiert.

```
HttpSession session = request.getSession(true);
```

- liefert ein neues Session-Objekt, falls noch keines vorhanden ist.

- Session-Objekt liegt im Server.
- Session-Objekt besitzt eine Datenstruktur zum Speichern von sitzungsbezogenen Informationen. Eigenschaften:
 - beliebig viele
 - von beliebige Typ (Serialisierbarkeit nicht erforderlich!)

Sitzungen im Servlet bearbeiten

- Ein Attribut in ein Session-Objekt einfügen:
`session.setAttribute(key, value);`
wobei key vom Typ String und value ein Objekt beliebigen Typs sein kann
- `setAttribute()` überschreibt existierende Einträge.
- Ein Objekt aus einem Session-Objekt lesen:
`Typ value = (Typ) session.getAttribute(key);`
mit Typ der Typ des gespeicherten Objekts.
- `getAttributeNames()` liefert String-Array mit allen Namen

Sitzungen im Servlet bearbeiten

- `getId()` liefert die ID einer Sitzung.
- `isNew()` liefert true, wenn die Sitzung neu ist, sonst false.
- `getCreationTime()` liefert die Zeit (in Millisekunden), die die Sitzung aktiv war.
- `getLastAccessedTime()` gibt an (in Millisekunden), wann zuletzt vom Browser zugegriffen wurde.
- `getMaxInactiveInterval()` gibt die Zeit an (in Sekunden), die eine Sitzung aktiv bleiben soll, wenn kein Zugriff vom Browser erfolgt.

Übung

- Schreiben Sie ein Servlet, das tabellarisch angibt, wann eine Sitzung gestartet wurde, sowie wann und wie oft sie besucht wurde.

Cookies

- Cookies sind textuelle Informationen, die der Web-Server im Browser hinterlegt. Der Browser schickt sie an den Server zurück, wenn er wieder die gleiche Seite besucht (Cookies werden vom Browser nicht verändert).
- Cookies werden verwendet, z.B.:
 - zur Identifizierung des Anwenders,
 - um wiederholte Eingabeaufforderungen zur User- und Passwortangabe zu vermeiden,
 - als “Gedächtnis“ für den Server.

Missverständnisse über Cookies

- Cookies ...
 - sind nicht ausführbar
 - werden nicht interpretiert
 - können keine Viren o.ä. einschleusen
 - können die Platte des Zielrechners nicht überfüllen
 - Browser akzeptieren i.A.
 - pro Seite max. 20 Cookies,
 - insgesamt max. 300 Cookies
 - können max. 4 kB groß sein.

Cookie API

- Cookies mit new Cookie(name, value) erzeugen.
- In beiden Parametern dürfen keine Leerzeichen, [,], (,), =, “, /, ?, @, : oder ; vorkommen.
- Für Cookie-Attribute gibt es setXyz() und getXyz()-Methoden.
- Cookie-Attribute sind:
 - comment, domain, maxAge, name, path, secure, value, version

Cookies lesen und schreiben

- Ein Cookie wird mittels addCookie() aus der Klasse HttpServletResponse eingefügt.
- Beispiel:
 - `Cookie newCookie = new Cookie("user", "judt");
response.addCookie(newCookie);`
- Cookies können vom Browser mit der Methode getCookies() aus der Klasse HttpServletRequest gelesen werden.
 - `getCookies()` liefert ein Array von Cookie-Objekten.
- Cookie-Attribute können über get-/set-Methoden gelesen und geschrieben werden.

Beispiel

- Methode sucht das übergebene Array nach einem Cookie-Namen ab und liefert bei einem Treffer den dazugehörigen Wert zurück, sonst einen Standardwert.

```
public static String getCookieValue(Cookie[] cookies,  
        String cookieName, String defaultValue) {  
    for (int i=0; i < cookies.length; i++) {  
        Cookie cookie = cookie[i];  
        if (cookieName.equals(cookie.getName())))  
            return cookie.getValue();  
    }  
    return defaultValue;  
}
```

Übung

1. Erstellen Sie eine eigenes Cookie-Objekt, dessen Lebensdauer einen Monat (30 Tage) beträgt.
2. Erstellen Sie eine Web-Seite, die Cookies verwendet, um dem Benutzer das wiederholte Ausfüllen eines Formulars zu erleichtern. Das Formular beinhaltet zwei Felder, eines für den Benutzernamen, das andere fürs Passwort. Im Formular soll der Benutzernamen per Cookie gesetzt werden. Verwenden Sie die obige Cookie-Klasse.

Selbstkontrolle

1. Warum hat ein Servlet keine Methode main(...)?
2. Wie würden Sie ein Servlet für FTP-Kommunikation konzipieren?
3. Skizzieren Sie die Implementierung eines Warenkorbes mit Session-Objekten.
4. Warum können Sitzungsdaten nicht vollständig bei Anwender gespeichert werden?

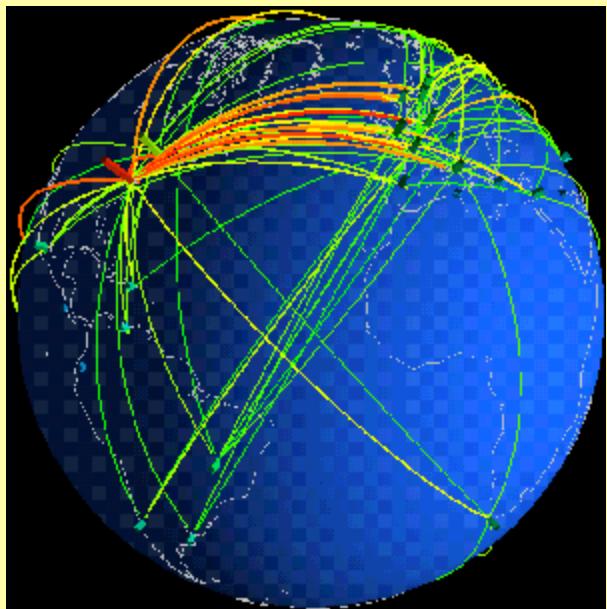
Kontakt

Duale Hochschule BW Ravensburg
Campus Friedrichshafen

Prof. Dr. Andreas Judt
Informationstechnik

judt@dhbw-ravensburg.de

Web Engineering Java Server Pages

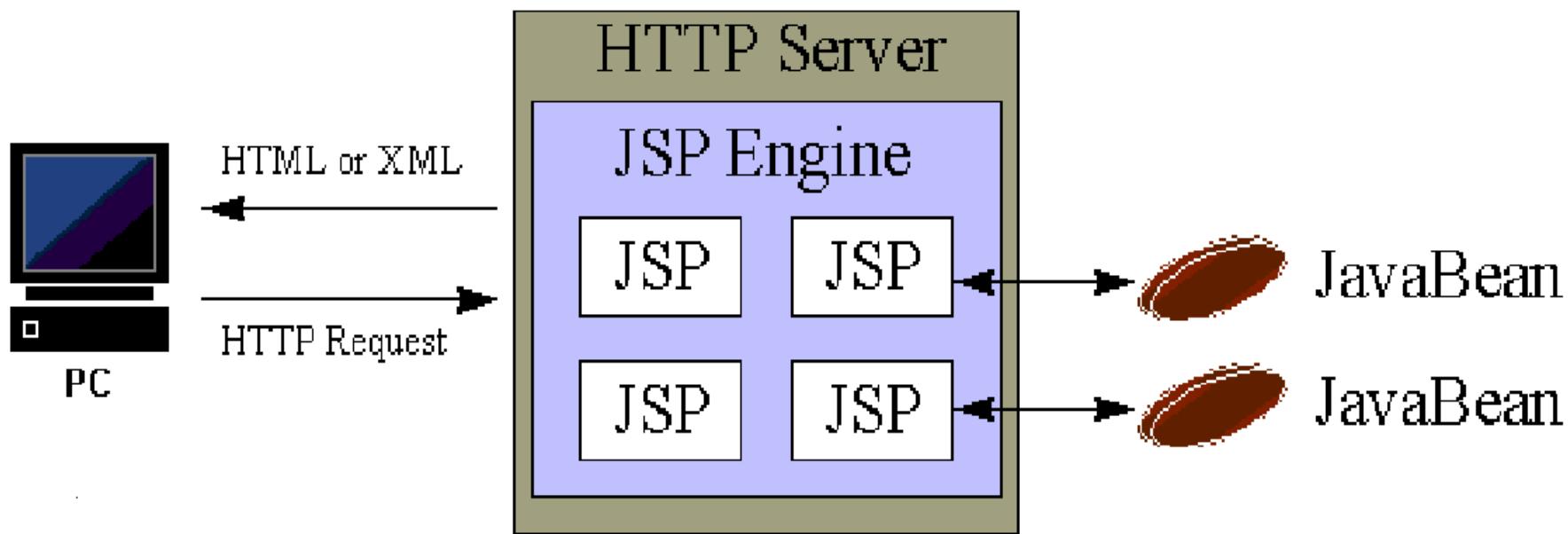


Prof. Dr. Andreas Judt

Was sind JSP?

- augenscheinlich: HTML kombiniert mit Java
 - HTML-Tags
 - JSP-Tags
 - Deklarationen
 - Scriptlets (kleine Java-Codes)
 - XML-Tags
 - JavaBeans
- erleichtert die Erstellung dynamischer Web-Seiten
 - JSP-Seite beinhaltet
 - JSP und Servlets sind in ihrer Funktionalität äquivalent

JSP Schema



JSP Architektur

- Die Ausführung von JSP hat zwei Phasen
 - Übersetzung:
 - JSP-Generator erzeugt aus der JSP-Datei ein Java Servlet. Das Servlet wird mit dem Java Übersetzer (javac) übersetzt.
 - Anfrage (Request): (GET oder POST)
 - `_jspService` des Servlets wird aufgerufen



Beispiel: HalloWelt.jsp

- Seite sieht aus wie eine HTML-Seite.
- Java-Befehl innerhalb des <%...%> Tags
 - auffällige Ähnlichkeit zu PHP!

```
<html>
  <head>
    <title>Hallo Welt !</title>
  </head>
  <body>
    <h1>Hallo Welt!</h1>
    <p>Heute ist der <%= new java.util.Date() %></p>
  </body>
</html>
```

erzeugt: HelloWorld.java

```
...
public class jsphallo_jsp_0 extends HttpJspBase {
    public void _jspService(HttpServletRequest request,
                           HttpServletResponse response) ... {
    ...
    // 'implizite' Objekte
    application = pageContext.getServletContext();
    config = pageContext.getServletConfig();
    session = pageContext.getSession();
    out = pageContext.getOut(); ...
    response.setContentType("text/html"); ...
    out.write("<html>\r\n<head>\r\n<title>Hallo Welt !</title>
              \r\n</head>\r\n<body>\r\n<h1>Hallo Welt!</h1>
              \r\n<p>Heute ist der ");
    out.print( new java.util.Date() );
    out.write("</p>\r\n</body>\r\n</html>\r\n"); ...
}
```

Beispiel: http Anfrage

- <%= ... %> kennzeichnet einen Ausdruck

```
<html>
  <body bgcolor="white">
    <h1> Request Information </h1>
    Methode: <%= request.getMethod() %>
    Angeforderte URL: <%= request.getRequestURI() %>
    Adresse Ihres Rechners: <%= request.getRemoteAddr()%>
    Ihr Browser: <%= request.getHeader("User-Agent") %>
  </body>
</html>
```

Skriptelemente

- JSP „Skriptelemente“ ermöglichen das Einfügen von Java-Programmabschnitten in das Servlet, das aus der JSP generiert wird.
 - <%! Java-Deklaration %>
 - neue Methoden und Variablen
 - <% Java-Code %>
 - beliebige Anweisungsfolgen
 - <%= Java-Ausdruck %>
 - auszuwertender Ausdruck

Skriptelemente

<%= Ausdruck %>

- Der Ausdruck wird ausgewertet und das Ergebnis ausgegeben <%= new java.util.Date() %>
- Implizite Objekte:
 - werden im Servlet angelegt
 - request: HttpServletRequest
 - response: HttpServletResponse
 - session: HttpSession
 - out: PrintWriter
- Beispiel:

<%= request.getHostName() %>

Skriptelemente

<% Code %>

- Java-Code Abschnitte, implementieren die Geschäftslogik
 - können mehrzeilig sein
 - werden direkt ins Servlet übertragen

```
<ul>
<%
for (int i = 0; i < 20; i++) {
    if (i%2 == 0) {%
        <li><%= i %>
    } else {%
        <li><h1><%= i %></h1>
    }
}
</ul>
```

Skriptelemente

<%! Deklaration %>

- Java-Deklarationen
- Variablen, Methoden, ...
- können mehrzeilig sein
- werden direkt ins Servlet übertragen

```
<%! public int counter = 0;
```

```
    public int getCounter() {  
        return(counter);  
    }  
%>
```

```
...
```

```
<p>Sie sind der <%= getCounter() %>. Besucher.</p>  
<% counter++ %>
```

JSP Anweisungen

- JSP Anweisungen (directives) wirken auf Gesamtstruktur des Servlets
 - JSP Anweisung:
`<%@ anweisung attribut = "wert" %>`
 - oder:
`<%@ anweisung_1 attribut_1 = "wert_1"
attribut_2 = "wert_2"
...
attribut_n = "wert_n" %>`
- Haupttypen der Anweisungen:
 - page
 - include

JSP Anweisungen

<%@ page %>

- Die page-Anweisung ermöglicht:
 - Klassen zu importieren
 - das Verhalten des Servlets zu beeinflussen
- Mit der page-Anweisung können ein oder mehrere Attribute definiert werden:
- import: Import von Java Bibliotheken (Packages)

```
<%@ page import="java.util.*" %>
```

JSP Anweisungen

<%@ page %>

- contentType: Festlegung des verwendeten MIME-Typs

```
<%@ contentType="MIME-Type; charset=Char-Set"%>
<%@ page contentType="text/plain" %>
```
- isThreadSafe="true|false": Angabe, ob mehrere Requests gleichzeitig von einer Servlet-Instanz bearbeitet werden können, oder ob die Requests nacheinander abgearbeitet werden.

```
<%@ page isThreadSafe="true" %>
```

JSP Anweisungen

<%@ page %>

- session="true|false": gibt an, ob das Servlet (die JSP) ein Session-Objekt zugeordnet bekommt. Ein Session-Objekt speichert Zustandsdaten über mehrere Requests hinweg.
- extends="Klassenname": Legt die Oberklasse des Servlets fest.
- errorPage="URL": gibt an, zu welcher Seite im Fehlerfall gesprungen werden soll.
- isErrorPage="true|false": Gibt an, ob die aktuelle Seite als Fehlerseite von anderen Seiten verwendet werden kann.

JSP Anweisungen

<%@ include %>

- Durch die Anweisung include können andere Dateien in die aktuelle Seite eingefügt werden.

```
<%@ include file="relative_url" %>
```

- Beispiel zur Verwendung: Alle HTML-Seiten eines Unternehmens haben einen identischen Seitenkopf. Dieser kann problemlos durch die include Anweisung in alle Seiten eingefügt werden.

Scriptlets Beispiel

```
<ul>
    <li><b>Expression</b>
        Hostname: <%=request.getRemoteHost()%>.
    <li><b>Skriptlet</b><%out.println("GET-Daten " +
        request.getQueryString());%>
    <li><b>Deklaration und Ausdruck</b>
        <%! private int zugriffsZaehler = 0%>
        Zugriffe seit letzten Booten: <%= ++zugriffsZaehler%>
    <li><b>Anweisung und Ausdruck</b>
        <%@ page import= "java.util.*" %>
        Die aktuelle Zeit ist: <%= new Date() %>
</ul>
```

Aktionen

- JSP Aktionen verwenden XML Syntax, um das Verhalten des Servlets zu steuern.
- Aktionen sind:
 - `jsp:include`
 - `jsp:setProperty`
 - `jsp:getProperty`
 - `jsp:forward`
 - `jsp:plugin`
 - `jsp:useBean`

Aktionen: include

- Fügt eine Seite in die erzeugte ein.

```
<jsp:include page="relative URL" flush="true"/>
```

- Unterschied zur include-Anweisung:
 - bei der include-Anweisung wird die Seite dann eingefügt, wenn die JSP zum Servlet umgewandelt wird
 - bei der include-Aktion wird die Seite erst bei Anfrage eingefügt

Aktionen: forward, plugin

- **jsp:forward:**
 - Weiterleitung des Requests auf eine andere Seite.
 - Page: relative URL
 - <jsp:forward page="/utils/fehlerReport.jsp"/>
 - <jsp:forward page="<%=javaExpression%>" />
 - <jsp:forward page="<%=req.getParam(\"param\")%>" />

- **jsp:plugin:**
 - fügt ein Browser-spezifisches OBJECT oder EMBED Statement in die Seite ein

Fehlerbehandlung: errorPage

- JSP verwendet eine Fehlerseite:

```
<%@ page errorPage="fehler.jsp" %>
...
<% if (Fehler) {
    throw(new Exception(...));
}%>
```

- Fehlerseite:

```
<%@ page isErrorPage="true" %>
...
<p>Ein Fehler ist aufgetreten:</p>
<p><%= exception.getMessage() %></p>
```

JSP und JavaBeans

- JSP Komponentenmodell verwendet JavaBean Komponenten
- JavaBean API legt Regeln zur Erstellung von wiederverwendbaren Komponenten fest.
- Die Verwendung von JavaBeans erleichtert die Erstellung von dynamischen Web-Seiten.

JavaBeans

- Ein Bean ist eine Klasse, die entsprechend den im JavaBean API festgelegten Regeln programmiert ist.
 - Die Regeln betreffen:
 - Namensvergabe
 - Software-Design

JavaBean API und JSP

- JavaBean Konventionen ermöglichen einem JSP-Behälter eine Interaktion mit einer Bean ohne jegliches Wissen über die Klasse
- Für JSPs von Interesse:
 - Konstruktor der Bean
 - Zugriffsmethoden der Bean
- Instanzen einer Bean-Klasse sind Objekte.
- Zugriff auf die Objekte
 - direkt aus einem Programm heraus
 - aus einer JSP

Bean Konventionen

- Bean Konventionen ermöglichen den Zugriff auf Methoden des Bean-Objektes
- Konstruktor:
 - Es muß ein Konstruktor ohne Argumente vorhanden sein. Dieser wird in JSP-Tag <jsp:useBean> zur Instanziierung des Bean-Objektes verwendet.

Bean Konventionen: Konstruktor

- Die Verwendung des Wortes Bean im Name der Klasse steht nicht in den JavaBean-Regeln
- Konstruktor initialisiert das Attribut (Property) name
- Für den Zugriff auf das Property eigene Methoden schreiben:

```
public class NameBean {  
    private String name;  
    public NameBean(){  
        this.name = "defaultName";  
    }  
}
```

Bean Konventionen

- Properties werden durch passende Zugriffsmethoden definiert.
- Zugriffsmethoden verwendet um
 - den Wert eines Property zu liefern (get)
 - den Wert eines Property zu verändern (set)
- Zugriffsmethoden: set<prop>/get<prop>
 - public void setName(Type value)
 - public Type getName()
- Zugriffsmethoden müssen public sein
- betroffene Instanzvariable sollten private sein

Bean Konventionen: Beispiel

- Die Methoden liefern und setzen ein Property.
- Property nicht mit einem Attribut verwechseln: ein Property ist definiert über die Methodennamen

```
public class NameBean {  
    private String xyzName; // name  
    public NameBean(){  
        this.xyzName = "defaultName";  
    }  
    public void setName(String n){  
        xyzName = n;  
    }  
    public String getName(){  
        return xyzName;  
    }  
}
```

Verwendung von Beans

- Für Properties müssen nicht beide Methoden (setter/getter) implementiert werden.

```
<jsp:useBean id="bean_name" class="NameBean"/>
<html>
<body>
Der Name ist <jsp:getProperty name="bean_name"
    property="name" />
</body>
</html>
```

JavaBeans: indizierte Properties

- Properties können mehrere Werte beinhalten.
- Alle Werte müssen vom gleichen Typ sein.
- sinnvolle Zugriffsmethoden:
 - Alle Elemente:
 - public Type[] getName()
 - public void setName(Type[])
 - Zugriff über Index:
 - public Type getName(int index)
 - public void setName(int index, Type value)
 - Behältergröße: public int getNameSize()

JavaBeans: Boolesche Werte

- Konvention für getter-Methoden für boolesche Properties: Prefix is
- Bei boolschen Werten wird die Methode mit dem Prefix is automatisch gesucht. Wenn nicht vorhanden, wird die entsprechende get-Methode gesucht.
- Ergebnis ist ein boolscher Wert.
- Die Namensvergabe der setter-Methoden bleibt unverändert.

Typumwandlung: primitive Datentypen

- primitive Datentypen: boolean, byte, char, short, int, long, float, double
- Alle Properties, auf die durch <jsp:getProperty> zugegriffen wurde, werden in String umgewandelt.
- Bei primitiven Datentypen automatische Typumwandlung durch JSP:
 - getter-Methode muß nicht explizit einen String liefern
 - setter-Methode wandelt einen String automatisch in einen korrekten Typ um.

Typumwandlung: komplexe Datentypen (Objekte)

- JSP ruft automatisch die `toString()`-Methode des jeweiligen Objektes auf.
- Bei komplexen Datentypen mehrere Vorgehensweisen:
 - getter- und setter-Methoden so gestalten, daß diese mit Strings oder mit primitiven Datentypen operieren und notwendige Umwandlungs-Operationen intern durchführen.
 - Mehrere setter-Methoden aber nur eine getter-Methode erstellen (z.B. `setStd()`, `setMin()`, `getUhrzeit()`)

JavaBeans konfigurieren

- Konfiguration einer Bean mittels Übergabe von Informationen an den Konstruktor nicht möglich:
`<jsp:useBean>`
- Properties zur Wertübergabe verwenden
- Beans können auch in einem „Scriptlet“ (`<% ... %>`) erzeugt werden
 - Zugriff auf Properties wie gehabt:
 - `<jsp:setProperty>`
 - `<jsp:getProperty>`

JavaBeans konfigurieren: Beispiel

```
public class AktienBean{  
    private double kauf_kurs;  
    private double min_kurs;  
    private double max_kurs;  
  
    public AktienBean(){this(100.0);}  
    public AktienBean(double kauf_kurs){  
        this.kauf_kurs = 100.0;  
        init();}  
  
    public double getKauf_kurs() {return kauf_kurs;}  
    public void setKauf_kurs(double kk) {kauf_kurs=kk;}  
  
    private void init(){  
        min_kurs = this.kauf_kurs * 0.75;  
        max_kurs = this.kauf_kurs * 1.25;  
    }  
}
```

Aktionen: useBean

- Laden von JavaBeans in eine JSP-Seite:
`<jsp:useBean id="name" class="package.class"/>`
- Bedeutung: " Instanziere ein Objekt der Klasse class und binde es mit der Variable id"
- id: Name der Variable der Bean. Ein neues Objekt wird erzeugt, wenn kein weiteres Bean-Objekt dieses Namens sichtbar ist (scope).
- class: Bean-Klasse
- scope: Gültigkeitsbereich von id.
 - Parameter: page,request, session, application

Aktionen: setProperty

- Zuweisung eines Wertes an eine Bean:

- beim Erzeugen des Objekts

```
<jsp:useBean id="myBean" ...>  
...  
<jsp:setProperty name="myBean" property="time"  
    value="12:23"/>  
</jsp:useBean>
```

- nach dem Erzeugen des Objekts

```
<jsp:useBean id="myBean" .../>  
...  
<jsp:setProperty name="myBean" property="time"  
    value="12:23"/>
```

Aktionen: setProperty

- Parameter:

- name: Bezeichner der Bean
- property: Attribut der Bean
- value: Wert
- param: Parameter aus dem Request-Objekt, dessen Wert zugewiesen werden soll

Aktionen: setProperty

- Spezialformen:

- alle Parameter eines Requests übergeben

```
<jsp:setProperty name="myBean" property="*" />
```

- Parameternamen != Properties:

```
<jsp:setProperty name="myBean" property="instruction"  
param="action" />
```

Aktionen: getProperty

- liefert den Wert eines Properties als String
- Wert lesen:

```
<jsp:getProperty name="myBean" property="time"/>
```

- Parameter:
 - name: Bezeichner der Bean
 - property: Attribut der Bean

Übung

- Erstellen Sie eine JSP, die anzeigt, wie lange die aktuelle Sitzung (Session) schon dauert. Die Anzeige soll in Minuten und Sekunden erfolgen.
- Hinweise:
 - die Klasse `java.util.Date` enthält beim Erzeugen die aktuelle Systemzeit in Millisekunden (`getTime()`)
 - benutzen Sie eine Bean zum Speichern der Zeit
 - Welchen Gültigkeitsbereich verwenden Sie für Ihr Objekt ?

Tag-Bibliotheken

- Durch Definition einer Tag-Bibliothek können Java-Entwickler kompliziertes serverseitiges Verhalten auf einfache und benutzerfreundliche „html-like“ Elemente (Tags) reduzieren, die Entwickler von Web-Seiten leichter in ihre JSP-Seiten einbinden können.
- Dies bietet weitere Möglichkeiten zur Trennung zwischen Präsentations- und Programmierlogik.
- Der Einsatz von selbstdefinierten JSP-Tags besteht aus 3 getrennten Komponenten:
 - JSP-Datei
 - Tag-Handler-Klasse
 - Deskriptorklasse für Tag-Bibliothek

Beispiel: HelloWorldTag

- Jetzt kann der neu implementierte Tag in einer JSP-Datei verwendet werden.

```
<%@ taglib uri="jsptags.tld" prefix="jsptags" %>
<h1><jsptags:hello/></h1>
```

- Dieser einfache Tag besitzt weder Rumpf noch Attribute und deshalb genügt ein Überschreiben der doStartTag()-Methode in der Tag-Handler-Klasse. Rückgabewert muss in diesem Fall SKIP_BODY sein.

HelloWorldTag: Tag Handler Klasse

- Java-Klasse, die das Interface javax.servlet.jsp.Tag implementiert. Sie definiert das Verhalten des neuen Tags.

```
public class HelloWorldTag extends TagSupport {  
    public int doStartTag() {  
        try {  
            JspWriter out = pageContext.getOut();  
            out.print("Hello World!");  
        } catch (IOException ioe) {  
            System.out.println("Fehler in HelloWorldTag");  
        }  
        return SKIP_BODY;  
    }  
}
```

HelloWorldTag: Deskriptor-Datei

- Mit einer Tag-Lib-Deskriptordatei wird die Java-Klasse mit einem speziellen XML-Tag-Namen verbunden.

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE taglib ...>
<taglib>
  <tlibversion>1.0</tlibversion>
  <jspversion>1.1</jspversion>
  <shortname>jsptag</shortname>
  <urn></urn>
  <tag>
    <name>hello</name>
    <tagclass>hello.HelloWorldTag</tagclass>
    <info>DSimple Hello World</info>
    <bodycontent>EMPTY</bodycontent>
  </tag></taglib>
```

Bewertung

- JSP trennen Geschäfts- und Präsentationslogik
 - Geschäftslogik in Java Beans ausgelagert (Wiederverwendbarkeit!)
 - Geschäftslogik in JSP Tags und Scriptlets
 - Präsentation ist ohne Java-Kenntnisse realisierbar
 - Applikationslogik ohne HTML-Kenntnisse realisierbar
- reines Java; JSP haben alle Vorteile der Java Plattform.
- hohe Leistung und Skalierbarkeit
 - wie Servlets
 - sehr ähnlich zu PHP, aber mit anderem Prozessmodell

Selbstkontrolle

1. Welcher Zusammenhang besteht zwischen Servlets und JSP?
2. Auf welche Datei beziehen sich Fehlermeldungen bei der Übersetzung einer JSP? Welche Konsequenzen sehen Sie?
3. Die Syntax von PHP und JSP ist sehr ähnlich. Vergleichen Sie die Leistungsfähigkeit beider Technologien.
4. Zeigen Sie anhand eines Beispiels, dass JSP-Quellcode Missverständnisse beim Programmierer erzeugen kann.
5. Nennen Sie ein Kriterium: Wann sollten JSP statt Servlets verwendet werden?
6. Bewerten Sie, ob Programmierer durch die Beschränkung auf XML oder HTML als Anfrageergebnis einer JSP benachteiligt sind.

Kontakt

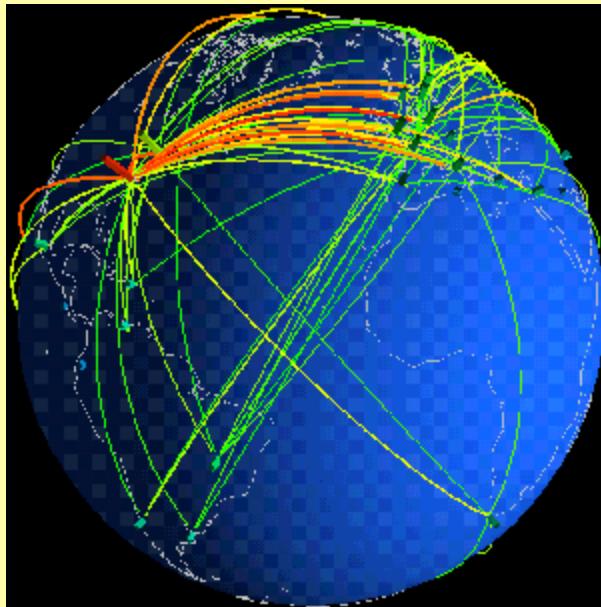
Duale Hochschule BW Ravensburg
Campus Friedrichshafen

Prof. Dr. Andreas Judt
Informationstechnik

judt@dhbw-ravensburg.de

Web Engineering

Java Database Connectivity (JDBC)



Prof. Dr. Andreas Judt

Was ist JDBC?

- Standard - Schnittstelle (API) zum Zugriff auf relationale Datenbanken in Java.
- JDBC™ wurde von Sun Microsystems als Warenzeichen registriert.
- JDBC ist eine Sammlung von Java-Klassen, Interfaces und Methoden.

Datenbankzugriffe früher

- einfache Programmierschnittstellen der Datenbank-Hersteller, einfache Netzwerkprotokolle
- für jede Datenbank eine eigene Technik
- gleichzeitiger Zugriff auf verschiedene Datenbanksysteme nur mit hohem Aufwand
- hoher Lernaufwand
- Fazit: Standardisierungsbestrebungen

Geschichte

- frühere Standards:
 - X/Open SQL CLI
 - ODBC
- Probleme:
 - (C/C++)-Schnittstellen, Pointerarithmetik
 - hohe Komplexität
 - hoher Entwicklungsaufwand, wenig Sicherheit, kaum Robustheit und Portabilität

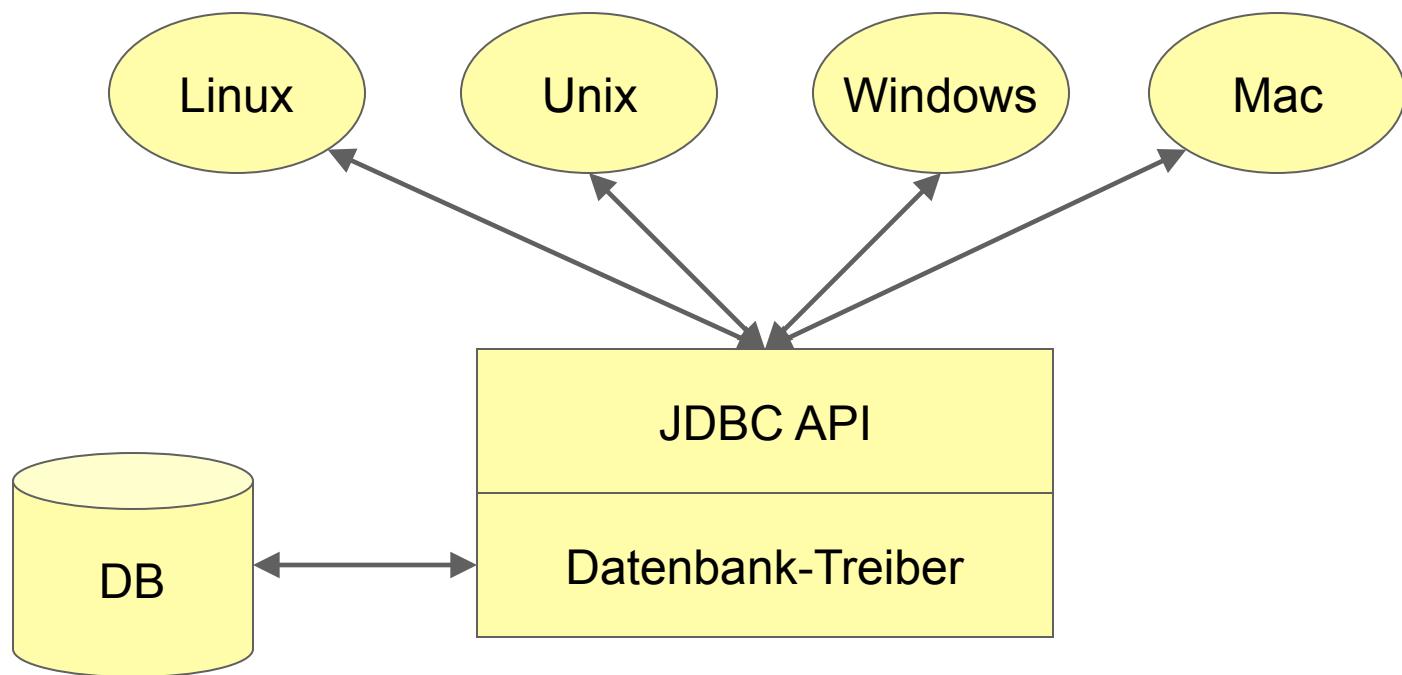
Anforderungen an modernen Datenbanktechnologien

- übersichtliche Schnittstelle
- einfache Programmierung
- Sicherheit
- Robustheit
- Plattformunabhängigkeit
- Datenbankunabhängigkeit
- Leistungsfähigkeit
- Unterstützung durch viele Datenbank-Hersteller

JDBC als moderne Datenbanktechnologie

- plattformunabhängig, robust und sicher durch Java
- einfache Programmierung durch einfache Schnittstelle
- datenbank-unabhängig, aber Spezialitäten einzelner Datenbanksysteme trotzdem nutzbar
- Inzwischen von praktisch allen Datenbank-Herstellern unterstützt.

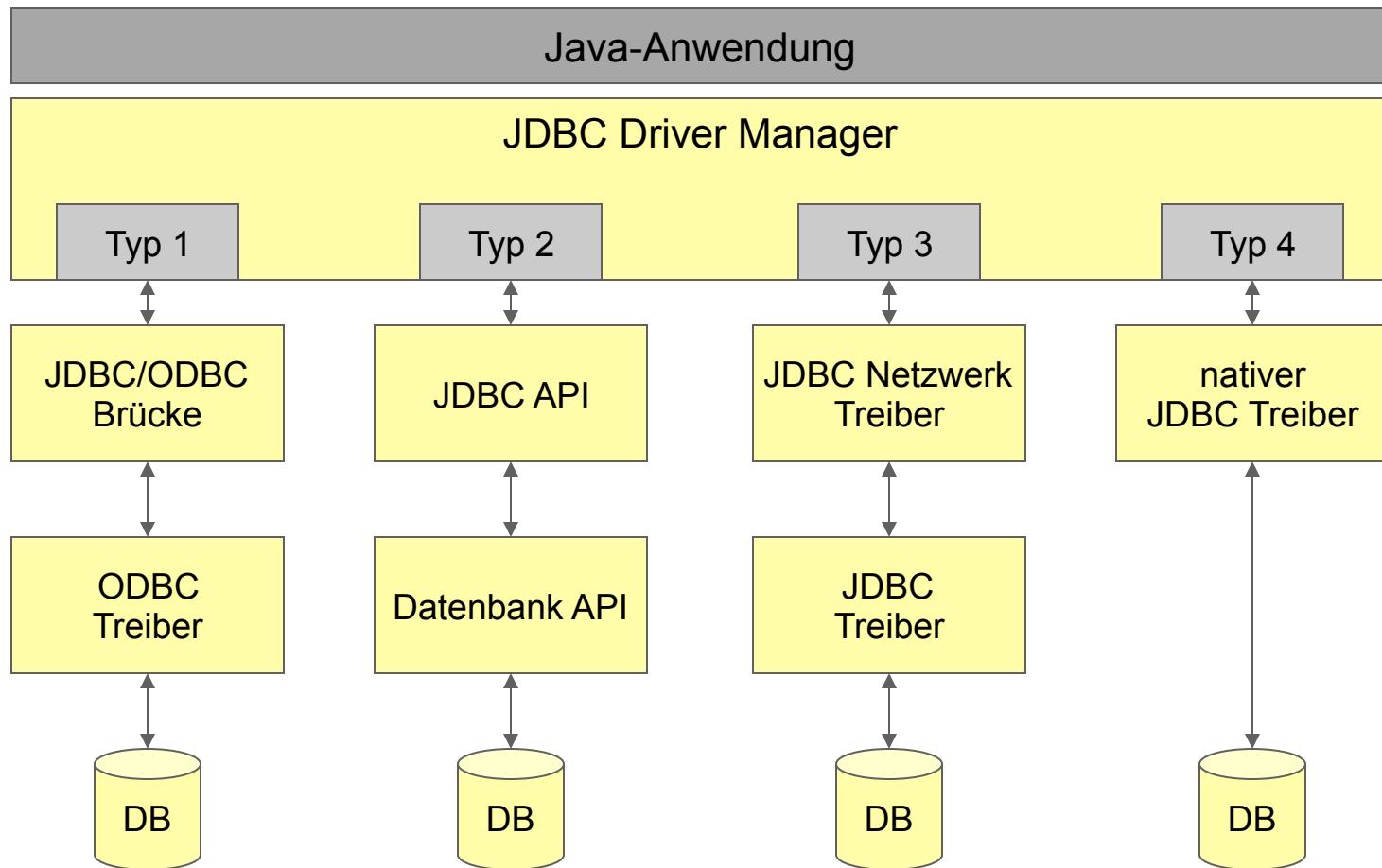
Datenbankzugriff mit JDBC



Funktionsweise eines JDBC Treibers

- JDBC-API stellt eine einheitliche Java-Schnittstelle zur Verfügung.
- Der JDBC-Treiber kümmert sich um die Umsetzung auf die spezifische Datenbank und organisiert die Kommunikation über das Netzwerk.

JDBC Treiberarchitekturen



JDBC Treiberarchitekturen

- Unterscheidungskriterien
 - vollständig in Java oder plattformabhängige Teile
 - Nur reine Java Treiber sind über das Netzwerk dynamisch ladbar.
 - verwendetes Netzprotokoll

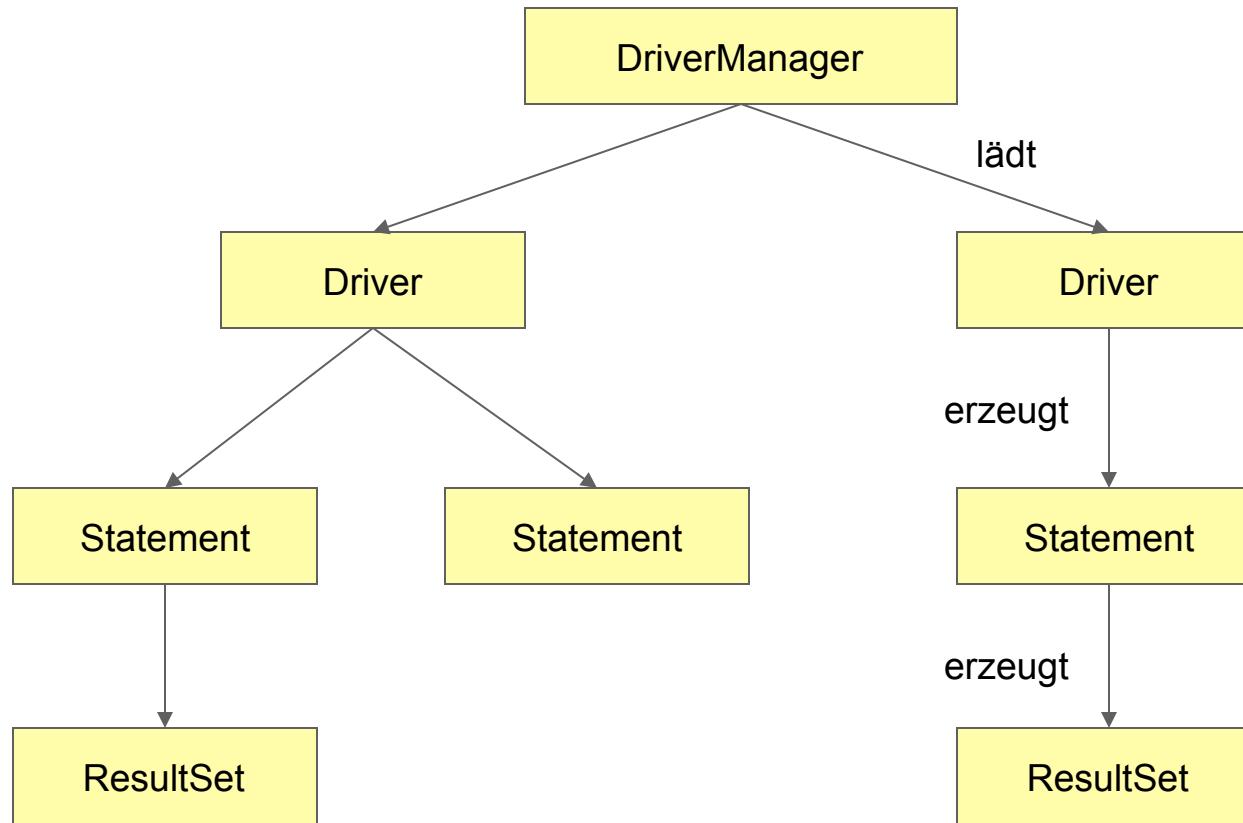
JDBC Treiberarchitekturen

- Typ 1 Treiber (JDBC/ODBC Brücke)
 - verwenden die Microsoft ODBC (open database connectivity) Treiber
- Typ 2 Treiber (Brückentreiber)
 - sind für die Einführungsphase einer Datenbank gedacht.
- Typ 3 Treiber (JDBC Netzwerk)
 - sind DB-unabhängig
- Typ 4 Treiber (nativ)
 - sind DB-spezifisch aber plattformunabhängig
 - heute Standard für Datenbank-Treiber

JDBC API

- JDBC API im Paket `java.sql`
- im wesentlichen 8 Interfaces, die wichtigsten:
 - Driver
 - Connection
 - Statement
 - ResultSet
- DriverManager übernimmt die Verwaltung der benutzten Treiber innerhalb der Anwendung.

Struktur beliebiger Datenbankverbindungen



Kommunikation mit einer Datenbank

- Ablauf
 1. Laden eines Treibers
 2. Herstellen der Verbindung (Öffnen der DB)
 3. Senden und Ausführen von SQL-Anweisungen
 4. Auslesen und Bearbeiten der Ergebnisse
- Schritte 1 und 2 nur einmal (am Anfang)
- Schritte 3 und 4 evtl. wiederholt...

Laden eines Datenbank-Treibers

- Mit der Zeile

```
Class.forName(„Treibername“);
```

wird ein Treiber (eine Java-Klasse !) geladen und beim DriverManager registriert.

- Es ist möglich, mehrere Treiber in einer Anwendung zu laden.
 - Unterscheidung der Treiber erfolgt in der URI der Datenbank.

Verbindung herstellen

- Verbindung mit der Datenbank aufbauen:

```
Connection conn =  
    DriverManager.getConnection(  
        uri, db-login, db-passwd);
```

- Ergebnis ist eine geöffnete Datenbank-Verbindung.
- url (universal ressource locator) spezifiziert den Pfad zur Datenbank.

- Beispiel:

```
getConnection(“jdbc:mysql://localhost:3306/MyDatabase”,  
    “user”, “password”);
```

Erzeugen von Anweisungen (engl. statements)

- Vor der Ausführung von SQL-Anweisungen muß ein Statement-Objekt erzeugt werden:

```
Statement stmt =  
    conn.createStatement();
```

- Für die Ausführung sind 3 Arten von Statements zu unterscheiden:
 - kein Ergebnis, z.B. SQL create table
 - Ergebnis ist ein Integer-Wert, z.B. SQL update
 - Ergebnis sind Tabellenzeilen, z.B. SQL select

Ausführen von Anweisungen

- Statements, die keine Ergebnisse zurückliefern, werden mit dem Kommando

```
stmt.execute(Statement);
```

ausgeführt.

- Bei Update-Kommandos kann die Anzahl der geänderten Datensätze erfragt werden:

```
int anzahl = stmt.executeUpdate(Statement);
```

Ausführen von Anweisungen

- Anfragen, die Datensätze zurückliefern, werden mit dem Kommando

ResultSet rs = stmt.executeQuery(Statement)
ausgeführt.

- Ergebnis ist ein Objekt vom Typ ResultSet.

Ausführen von Anweisungen

- Fehler bei der Ausführung von Statements (d.h. insbesondere auch Fehlermeldungen von der Datenbank) treten hier in Form von SQL-Exceptions auf.
- Objekte vom Typ SQLException enthalten
 - eine textuelle Fehlermeldung
 - einen datenbankspezifischen Fehlercode
 - einen genormten, datenbankunabhängigen Fehlercode (X/Open Standard)

Ergebniszeilen (engl. result sets)

- Ein ResultSet ist eine Tabelle mit Zeilen und Spalten und einem Zeilencursor.
- Cursor in die nächste Zeile bewegen:

```
boolean b = rs.next();
```

- Der Wert einer Spalte der aktuellen Zeile lesen:

```
result = rs.getxxx(Spaltenangabe);
```

- xxx = Java-Typ (also z.B. Int, Float , String,...)

Datenbankzugriff bisher

- Statements werden unverändert an die Datenbank weitergegeben.
- Auf diese Art sind auch Spezialitäten einzelner Datenbank-Hersteller nutzbar.
- vollständig erzeugte Datenbank-Anweisungen
 - Klient generiert Anfrage
 - Datenbank muss die Anfrage entgegennehmen, parsen, in ausführbare Operatorfolge übersetzen und ausführen
 - bei mehrfach verwendeten Anfragen entsteht unnötiger Verarbeitungsaufwand

Vorübersetzte Datenbank-Anweisungen (engl. prepared statements)

- PreparedStatements sind vorübersetzte Anfragen, die immer wieder verwendet werden können.
- Es werden für die Anfragen nur die Parameter ausgetauscht, die eigentlichen Datenbankanfrage bleibt unverändert.
- PreparedStatement erzeugen:

```
PreparedStatement ps =  
    conn.prepareStatement(String Statement);
```

Vorübersetzte Datenbank-Anweisungen (engl. prepared statements)

- Positionen im Statement, an denen Parameter eingesetzt werden sollen, werden durch „?“ markiert.
- Beispiel:

```
UPDATE person SET city = „?“ WHERE plz = „?“;  
SELECT * FROM person WHERE name LIKE „?“;
```

Parameter setzen

- Parameter eines Prepared Statements setzen:
`ps.setxxx(int Index, Wert);`
- Index ist der Index des einzusetzenden Wertes im Statement (Indizes beginnen bei SQL mit 1!)
- xxx ist der Java-Typ von Wert (also Int, Float,...)

Aufrufsschnittstellen (engl. callable statements)

- Callable Statements erweitern das Konzept der serverseitigen Ablage von Aufträgen an die Datenbank. Sie bilden die Verbindung zu kompletten, auf dem Server gespeicherten und ausführbaren Prozeduren (engl. stored procedures).
- Stored Procedures kapseln in der Regel eine Menge von SQL-Anweisungen, die eine bestimmte Aufgabe erledigen.
- Die Aufrufsyntax für Stored Procedures ist in verschiedenen Datenbanken unterschiedlich. JDBC verwendet aber ein einheitliches Format (sog. Escape-Format):

```
CallableStatement cs = conn.prepareCall(  
    „{call procname(?, ?)}“);
```

Aufrufsschnittstellen (engl. callable statements)

- (Ausgabeparam.: ? = call Procname(...);)
- Das Escape-Format wird vom Treiber in das hersteller-spezifische Format umgesetzt.
- Eingabeparameter werden wie bei Prepared Statements mit „?“ markiert und mittels setxxx(index)-Methoden gesetzt.

Bisher: Datenbankstruktur muss bekannt sein

- Starres Datenschema: Struktur der Tabellen zum Programmstart beim Klienten bekannt.
- Wünschenswert: Informationen über Tabellen, Datenbanken, ResultSets, also Metadaten (= Daten über Daten)
- Beispiel:
 - Feldnamen, -Längen, -Typen

Metadaten

- JDBC definiert Metadaten für zwei Klassen:

- für ein ResultSet:

```
java.sql.ResultSetMetaData  
rsmd = rs.getMetaData();
```

- für die ganze Datenbank:

```
java.sql.DatabaseMetaData  
dbmd = conn.getMetaData();
```

ResultSetMetaData

- Informationen über die Spalten eines ResultSets erhalten werden.
- Die wichtigsten:
 - int getColumnCount()
(Anzahl der Spalten des ResultSet)
 - String getColumnLabel(index);
 - String getColumnName(index);
 - int getColumnType(index);
 - String getColumnTypeName(index);

DatabaseMetaData

- Hier bekommt man alle möglichen Informationen über die gerade geöffnete Datenbank. (insgesamt über 150 Methoden definiert).
- Beispiele:
 - String getUserName();
 - boolean supportsOuterJoins();
 - ResultSet getTypeInfo();

Transaktionen

- Anwendungen führen Transaktionen im Auto-Commit-Modus aus: jedes SQL-Statement gilt als separate Transaktion und wird automatisch finalisiert (engl. commit), d.h. die Ergebnisse werden sofort in der Datenbank festgeschrieben und für andere Transaktionen sichtbar.
- Will man mehrere Anweisungen innerhalb einer Transaktion ausführen, so kann man Auto-Commit abschalten.
 - manuelles Finalisieren
 - manuelles Rückgängigmachen (engl. rollback)

Transaktionen

- Ohne Auto-Commit hält JDBC für jede Verbindung eine implizite Transaktion, die erst beendet wird, wenn die Methode commit aufgerufen wird.
- Will man noch nicht finalisierte Änderungen verwerfen, kann man den Zustand der Datenbank mit der Methode rollback zurücksetzen.

Ressourcenfreigabe

- Normalerweise werden Ressourcen freigegeben, sobald keine Referenz mehr darauf zeigt (und sie durch den Garbage-Collector eingesammelt werden).
- Dabei werden evtl. geöffnete Verbindungen, Cursors, Statements, und ResultSets automatisch geschlossen.
- Es ist auch explizite Freigabe möglich.

Bisher: JDBC 1.2

- Bisher wurde JDBC1.2 beschrieben
- Mit Java 2 wurde JDBC 2.0 spezifiziert, die entsprechende API ist seit JDK 1.2 verfügbar.
- JDBC 2.1 wird von den meisten DB-Herstellern heute noch nicht voll unterstützt.
- JDBC 2.x ist abwärtskompatibel.
- JDBC 3.0 korrigiert Fehler aus 2.x, ist aber wegen seiner Komplexität kaum verbreitet.

JDBC 2.0

- Scrollable und Updatable ResultSets
- Batch Updates
- Advanced Data Types
- Rowsets
- JNDI (Java Naming und Directory Services)
- Connection Pooling
- verteilte Transaktionen (engl. distributed transactions)

java.sql und javax.sql

- Um das Entwurfsziel der Übersichtlichkeit und Einfachheit von JDBC nicht zu gefährden, wurde in JDBC 2.0 die Basisfunktionalität von den Spezialfunktionen getrennt.
- Der bisherige Funktionsumfang (Core API) verbleibt mit leichten Verbesserungen in java.sql.
- Die neu hinzugekommenen Spezialfunktionen wurden in javax.sql ausgelagert.
- Optionale Funktionen sind im Optional Package API in javax.sql definiert.

JDBC 3.0

- Erweiterung und Fehlerbehebung von JDBC 2.0
- Optional Package API wird Pflicht
- Anpassung an moderne Datenbanktechnologien
- Optional Package API schwierig zu implementieren
- kaum vollständige Unterstützung der Datenbank-Hersteller

Übung: Datenbanken analysieren

- Teilen Sie den Kurs in Zweiergruppen auf.
- Wählen Sie eine beliebige Datenbank, die einen JDBC-Treiber besitzt.
 - z.B. mySQL, Oracle, MS SQL Server, Hypersonic, H2, Excel JDBC Driver
 - siehe: <http://java-source.net/open-source/database-engines>
- In jeder Zweiergruppe implementiert ein Mitglied eine beliebige Datenbank über die JDBC-Schnittstelle.
 - Die Datenbank soll mindestens zwei Tabellen beliebiger Struktur enthalten.
 - Die Datenbankstruktur soll vor dem zweiten Gruppenmitglied geheim gehalten werden.
- Das zweite Mitglied der Gruppe implementiert eine Analyse der unbekannten Datenbank und vergleicht seine Ergebnisse am Ende mit der Ausgangsimplementation.

Selbstkontrolle

1. In welcher Vererbungsbeziehung müssten Statement, PreparedStatement und CallableStatement stehen?
2. Worin besteht der Unterschied zwischen DatabaseMetadata und ResultsetMetadata?
3. Warum kann eine Anwendung trotz Typ-4 Treiber herstellerabhängig sein?
4. Mit welchem Wert beginnt die Indizierung bei Datenbanken? Worauf müssen Sie dadurch bei der Verwendung komplexer Datentypen achten?
5. Wie unterscheidet der Driver Manager verschiedene JDBC Treiber? Wie wird der geeignete Treiber für eine Verbindung gewählt?
6. Können Sie bei mehreren gleichen Datenbanken auf einem Rechner, die auf verschiedenen Ports antworten, den gleichen Treiber verwenden? Warum?

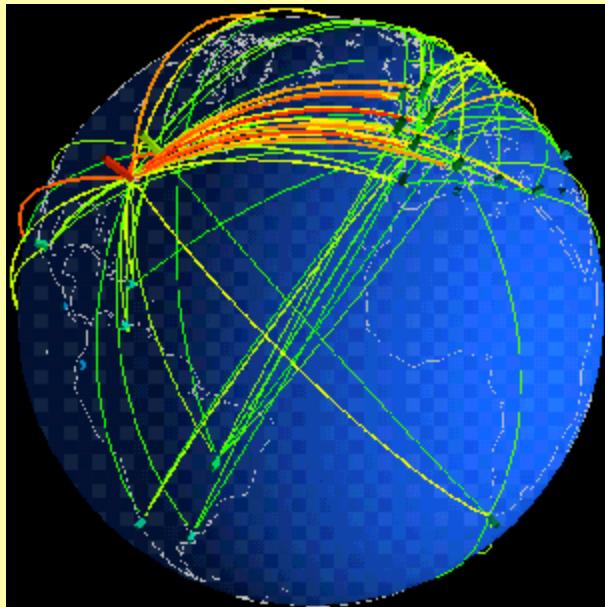
Kontakt

Duale Hochschule BW Ravensburg
Campus Friedrichshafen

Prof. Dr. Andreas Judt
Informationstechnik

judt@dhbw-ravensburg.de

Web Engineering RMI und CORBA

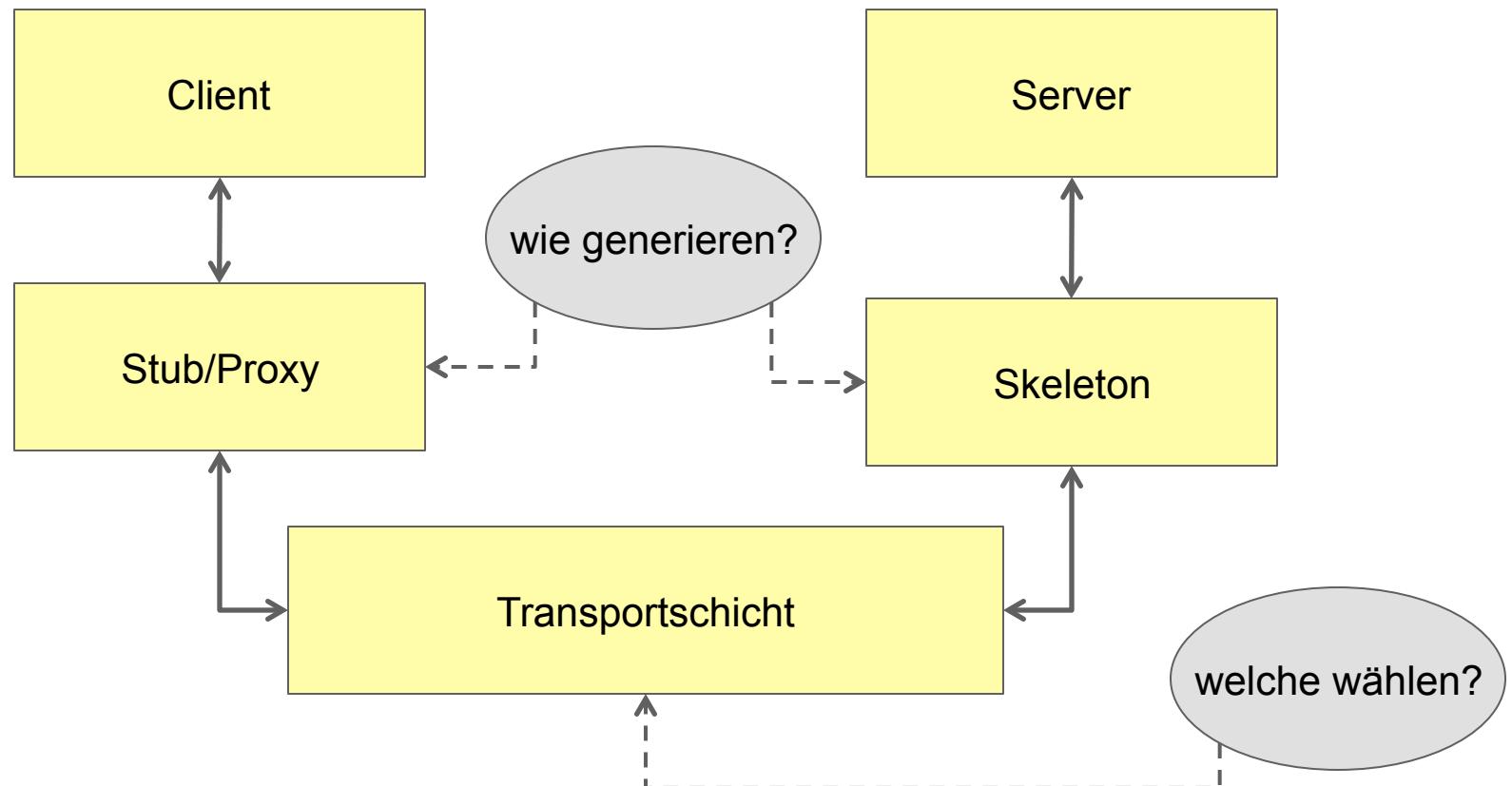


Prof. Dr. Andreas Judt

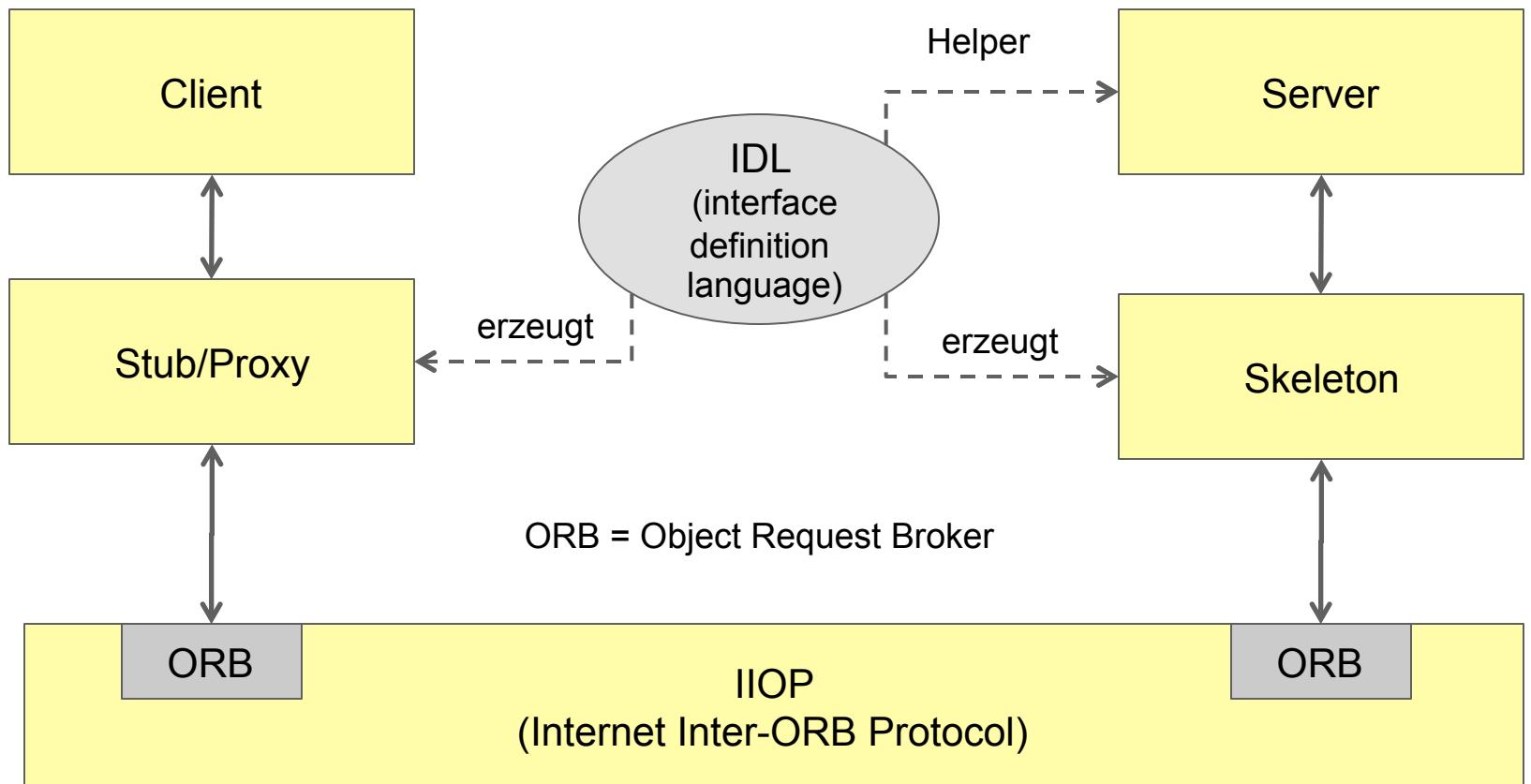
Überblick

- RMI=Remote Message Invocation
- CORBA=Common Object Request Broker Architecture
- CORBA und RMI: Zwei Möglichkeiten entfernter Methodenaufrufe in Java
- Verteiltes HelloWorld: RMI und CORBA ähneln sich in Basisfunktionalitäten
- CORBA oder RMI?

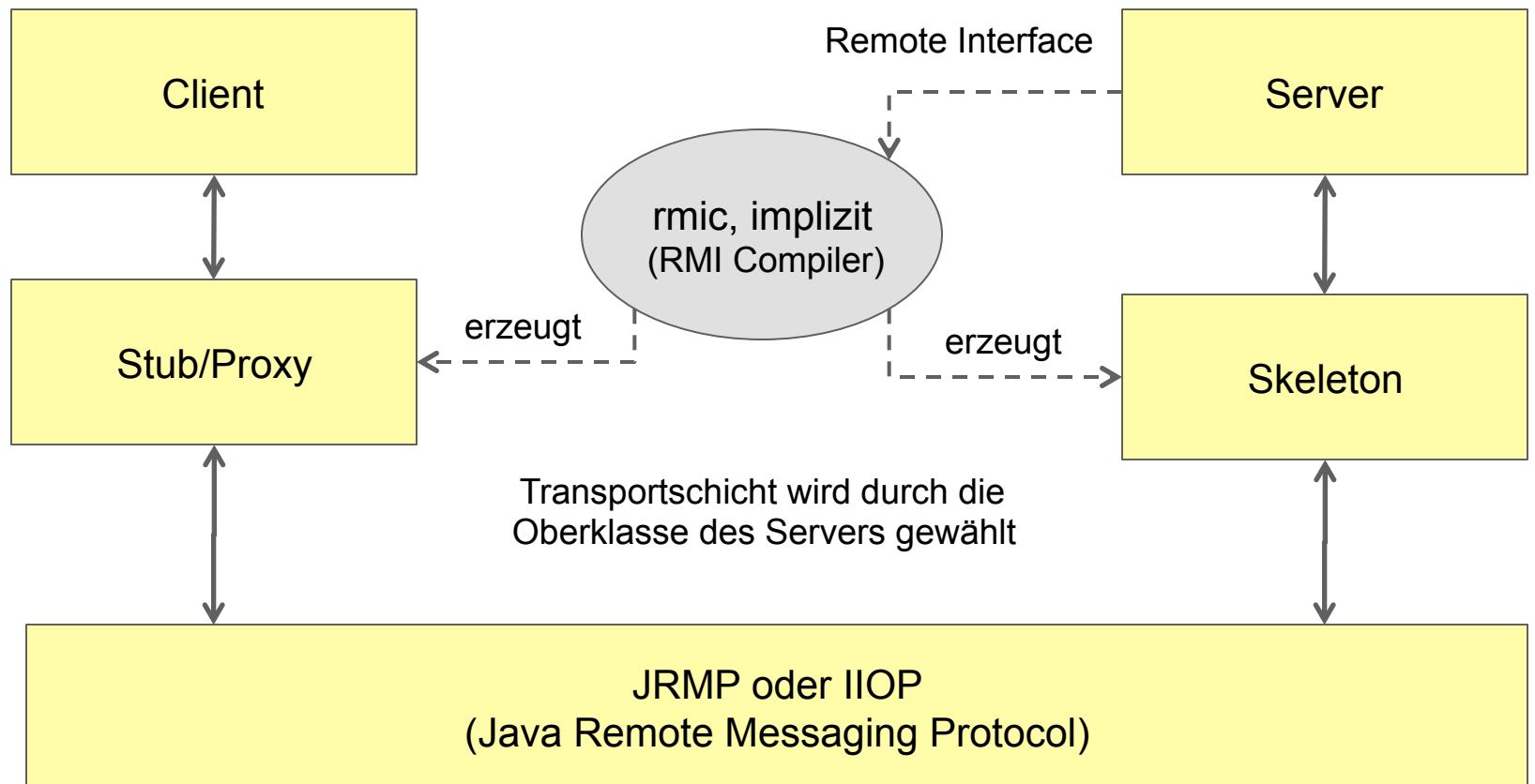
Prinzip des entfernten Methodenaufrufs



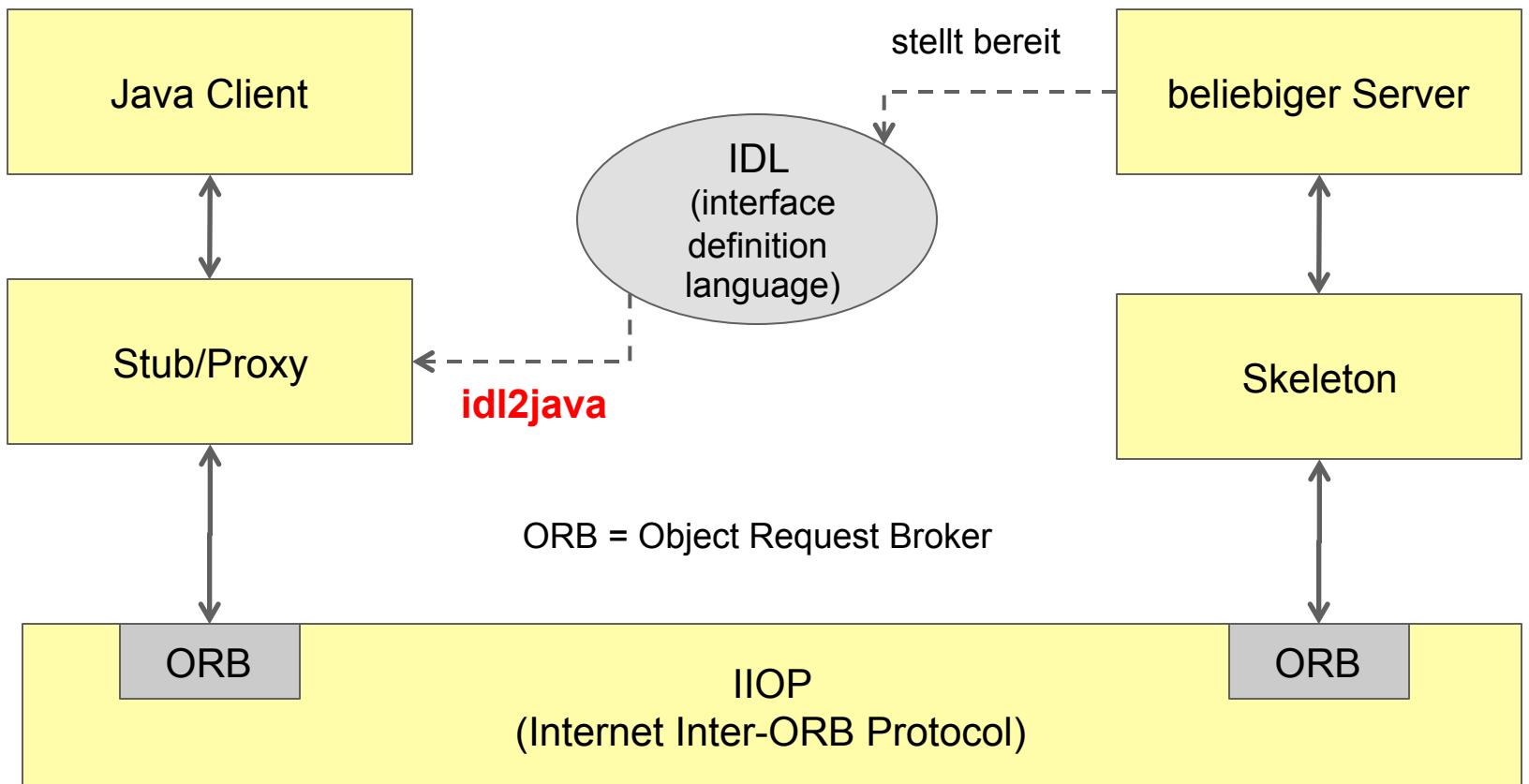
Entfernter Methodenaufruf mit CORBA



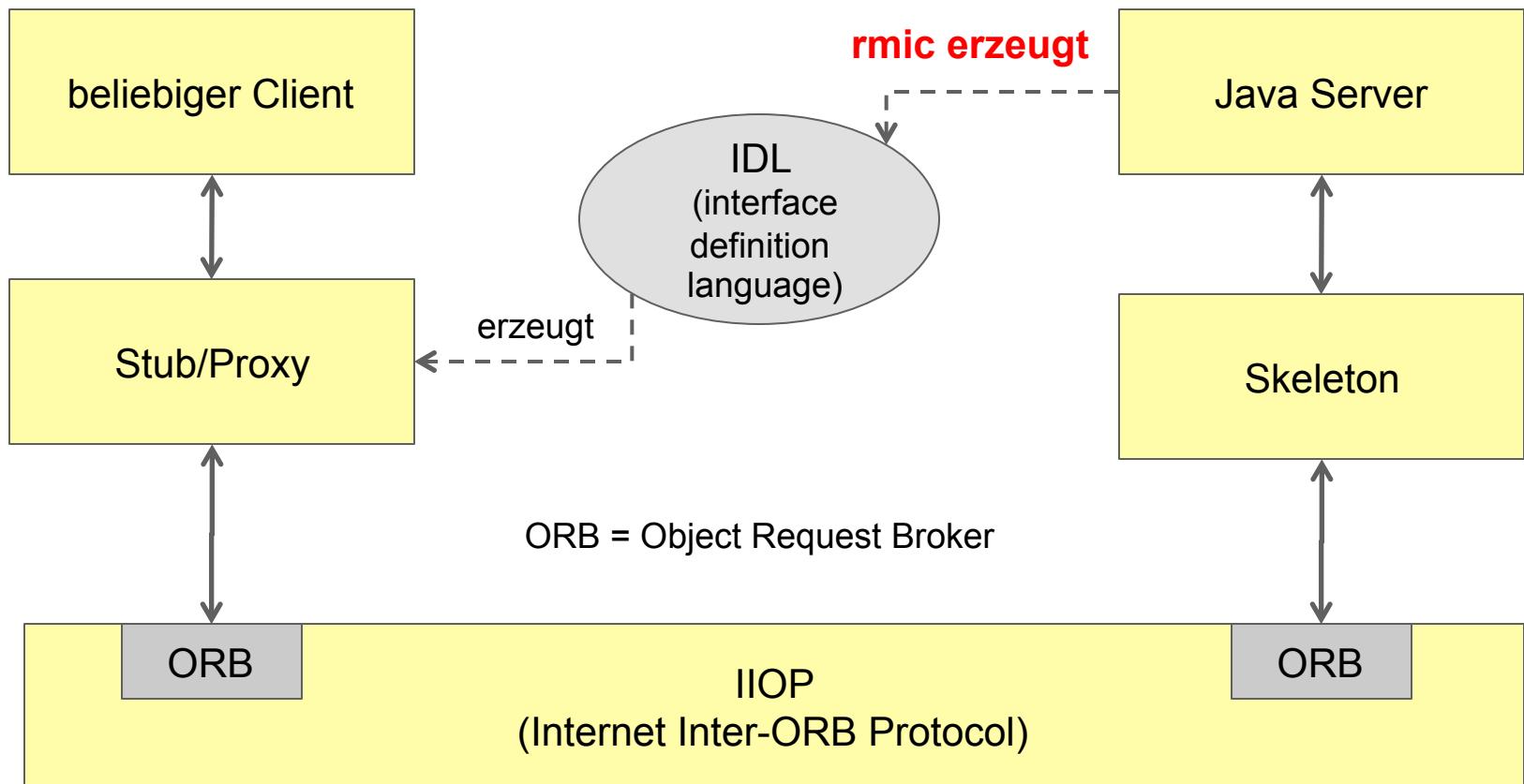
Entfernter Methodenaufruf mit RMI



Java Client kommuniziert mit beliebigen Server über CORBA



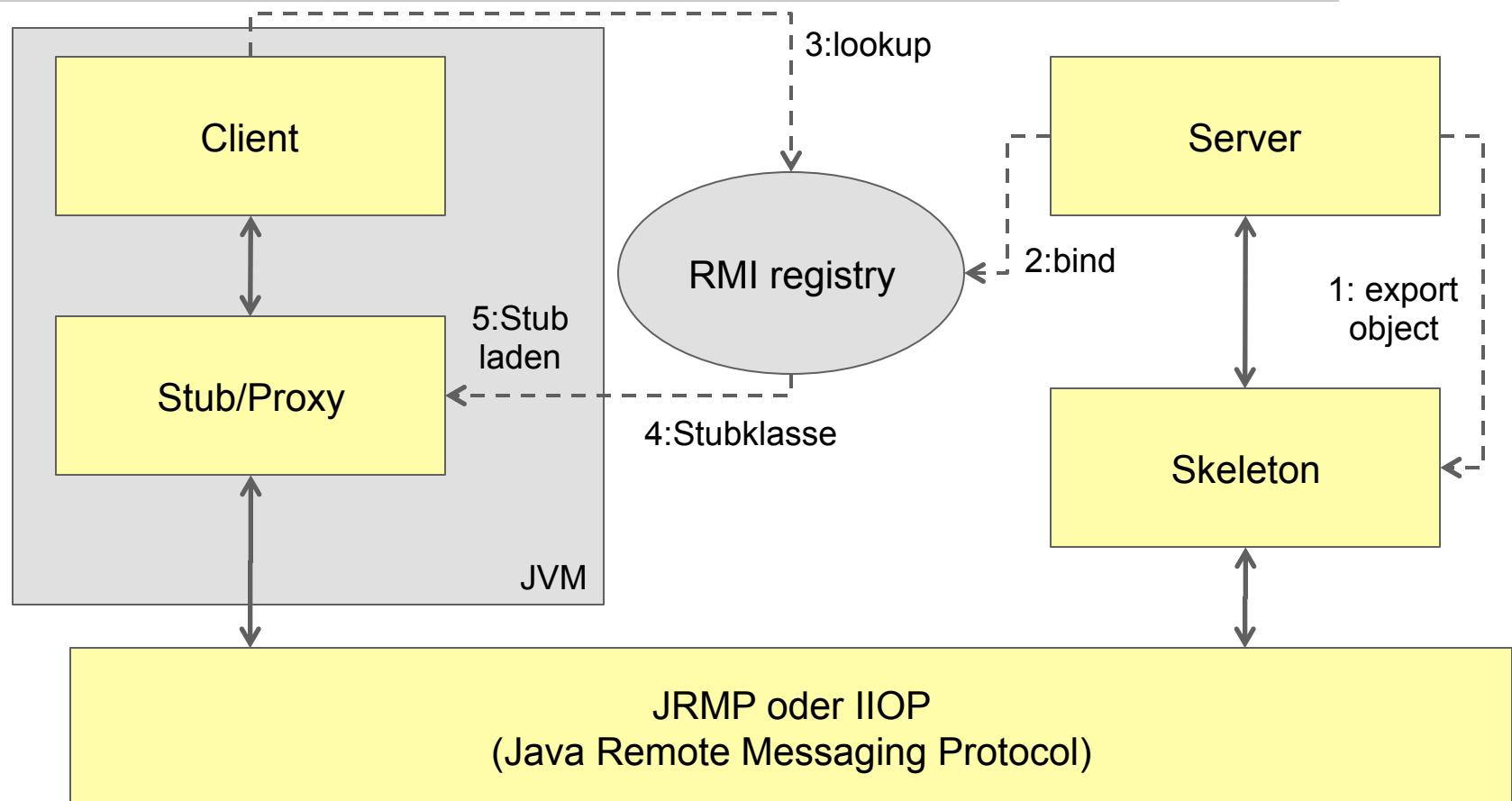
Beliebiger Client kommuniziert mit Java Server über CORBA



Remote Method Invocation (RMI)

- reiner Java-Mechanismus, kein (teurer) ORB nötig, keine Kompatibilitätsprobleme zwischen CORBA-Herstellern.
“write once, run anywhere, run everywhere”
- wirkliches objektorientiertes Programmieren, statt Sprachabbildung auf den kleinsten gemeinsamen Nenner.
- Applet-Idee verallgemeinert: Laufzeit-Code-Verteilung
- zentraler Streitpunkt zwischen Sun und Microsoft

Entfernter Methodenaufruf mit RMI



RMI: Server, entferntes Objekt

- Das entfernte Objekt bietet Methoden an, die von einer anderen JVM aus aufgerufen werden können.
 - genannt: entfernte Methoden
 - entfernte Methoden werden in einer entfernten Schnittstelle deklariert (nicht IDL, sondern Java Interface)
- Es wird vom Garbage Collector erst dann entsorgt, wenn keine andere JVM mehr eine Referenz darauf besitzt.
- EJB und EJB-Container sind als entfernte Objekte realisiert.

RMI: entfernte Schnittstelle

- Ein entferntes Objekt implementiert eine (oder mehrere) entfernte Schnittstellen (Unterklasse von `java.rmi.Remote`).
- Die entfernte Schnittstelle gibt an, welche Methoden von anderen JVMs aus aufgerufen werden können.
- Dem Klienten muss nur die entfernte Schnittstelle bekannt sein, nicht aber die Stub-Klasse oder die Klasse des entfernten Objekts.
- Enterprise Java Beans (EJB) nutzen entfernte Schnittstellen.

RMI: Stub und Skeleton

- Ein Stub vertritt ein entferntes Objekt auf der Seite des Aufrufers.
- Stub und Skeleton werden mit Hilfe des rmic Übersetzers (heute implizit) aus der class-Datei des entfernten Objekts generiert.
- Die Transportschicht wird über die Wahl der Oberklasse des Servers eingestellt.
- Der Aufrufer eines entfernten Objekts erhält eine Referenz stets in Form eines Stub-Objekts.

Verteiltes HelloWorld

RMI

- Server/Entferntes Objekt
 - Code für entfernte Schnittstelle
 - Code für entferntes Objekt
 - entferntes Objekt bekanntgeben
 - übersetzen, Stub & Skeleton generieren, Nameserver starten
- Client/Entfernter Aufrufer
 - entferntes Objekt nachschlagen
 - entfernter Methodenaufruf

CORBA

- IDL Definition
 - Stub & Skeleton, Helper und Rahmenimplementierungen generieren
- Server/Entferntes Objekt
 - Code für entferntes Objekt
 - entferntes Objekt bekanntgeben
- Client/Entfernter Aufrufer
 - entferntes Objekt nachschlagen
 - entfernter Methodenaufruf

Verteiltes HelloWorld: entfernte Schnittstelle

RMI:

```
package HelloApp;  
public interface Hello extends java.rmi.Remote {  
    String sayHello() throws java.rmi.RemoteException;  
}
```

CORBA IDL:

```
module HelloApp {  
    interface Hello {  
        string sayHello();  
    };  
};
```

Verteiltes HelloWorld: entfernte Schnittstelle

```
class HelloServant extends _HelloImplBase {  
    public String sayHello() {  
        return "Hello World!";  
    }  
}
```

- Durch `_HelloImplBase` wird Funktionalität geerbt, die die entfernte Aufrufbarkeit realisiert.
- `_HelloImplBase` wird aus IDL generiert.
- Objekterzeugung allein macht das Objekt nicht nach außen sichtbar. Expliziter connect-Aufruf nötig.
- Nur Methoden in der entfernten Schnittstelle können entfernt aufgerufen werden.

RMI: Übersetzen, Stub/Skeleton generieren, Nameserver starten

- javac *.java
 - übersetzen wie üblich
- rmic HelloServant
 - generiert Stubklasse
 - ab JDK 1.5 implizit
- rmiregistry
 - startet Nameserver
- java HelloServant
 - vorausgesetzt obige main Methode liegt in dieser Klasse.

RMI: Entferntes Objekt erreichen, entfernte Methode aufrufen

```
try {  
    Remote robj =  
        Naming.lookup("rmi://host:port/HWServer");  
    Hello hw = (Hello)robj;  
    System.out.println(hw.sayHello());  
} catch (Exception e) {  
    ... //do something  
}
```

- Entfernte Schnittstellen werden benutzt.
- Schutz vor java.rmi.RemoteException unterschiedlichster Art ist erforderlich.
- Es kann auch JNDI-Naming verwendet werden.

CORBA: Entferntes Objekt erreichen, entfernte Methode aufrufen

```
try {  
    ORB orb = ORB.init(args, null);  
    org.omg.CORBA.Object objRef =  
        orb.resolve_initial_references("NameService");  
    NamingContext ncr = NamingContextHelper.narrow(objRef);  
    NameComponent nc = new NameComponent("Hello", "");  
    NameComponent path[] = {nc};  
    Hello hw = HelloHelper.narrow(ncr.resolve(path));  
    System.out.println(hw.sayHello());  
} catch (Exception e) {  
    ... //do something  
}
```

RMI und CORBA im Vergleich

- RMI und CORBA in Basisfunktionalität ähnlich
- Parameterübergabe ist der zentrale Unterschied
- RMI:
 - dynamisches Laden von Klassen
 - Firewall-Durchtunnelung per HTTP, aber: CORBA RFC
 - wirklich objektorientiertes Programmieren
- CORBA:
 - Sprach-Interoperabilität, aber: RMI-IIOP
 - CORBAservices, aber: JTS, ...

Parameter entfernter Methoden

- Parameter-Typen:
 - CORBA: Parametertypen müssen in IDL erklärt sein.
Alle derartigen Objekte sind “entfernte Objekte”.
 - RMI & CORBA: Übergabe entfernter Objekte per Referenz
 - RMI: auch “normale” Java-Objekte möglich, die nicht entf. aufrufbar sind
 - Aufrufer erhält Kopien der übergebenen lokalen Objekte.
 - Argumente implementieren java.io.Serializable
- Parameter-Semantik:
 - CORBA kennt in-, out- und inout-Parameter.
 - RMI kennt nur in-Parameter, wegen Java Semantik.

RMI: Dynamisches Klassenladen

- Bisher sind wir davon ausgegangen, dass der Client:
 - automatisch Stubs instanziert,
 - die Methodenaufruf-Parameter deserialisiert und dabei passende Objekte instanziert.
- Dazu muss der Client den ByteCode laden können, der zu den Klassen gehört:
 - Bytecode kann aus CLASSPATH des Clients stammen.
 - Bytecode kann von einem Web-Server stammen!
- Sicherheitsprüfung, “signed”, SecurityManager, ...

RMI oder CORBA?

- Beide Mechanismen haben ihre Stärken.
- Verwende CORBA:
 - für integrative Anwendungen,
 - für Anwendungen mit viel existierendem Code.
- Verwende RMI:
 - wenn Client und Server vorwiegend Java-Programme sind
 - für Web-Anwendungen
- Es gibt Brücken zwischen beiden Welten.
- EJB verwenden innerhalb des Applikationsservers RMI. Der Applikationsserver kann IIOP, JRMP oder ein sonstiges Protokoll für die Kommunikation mit den Clients verwenden.

Übung: RMI Kommunikation

- Entwickeln Sie einen entfernten Methodenaufruf `getTime()` über RMI. Die Methode soll die Systemzeit des Servers als `java.util.Date` zurückliefern.
- Hinweise
 - Starten Sie Client und Server in getrennten JVM.
 - Server und RMI Registry müssen sich auf dem gleichen Rechner befinden.
 - Sie können die RMI Registry außerhalb der JVM starten.
 - Laden Sie die Stellvertreterklassen über den Klassenpfad des Client.

Übung: CORBA Kommunikation

- Implementieren Sie die RMI-Übung in CORBA.
- Hinweise
 - Beachten Sie, dass sich RMI und CORBA im wesentlichen in der Wahl der Oberklasse und bei der Erzeugung der Objekte unterscheiden.
 - Ein CORBA ORB ist in der J2SE enthalten.

Selbstkontrolle

1. Warum gibt es in Java RMI als Alternative zu CORBA?
2. Was ist ein ORB?
3. Erläutern Sie, warum Skeletons für RMI nicht vom RMI Compiler (rmic) erzeugt werden.
4. Erläutern Sie, wie sie mit einem RMI Server das CORBA Protokoll IIOP verwenden können.
5. Wann sollten Sie CORBA verwenden, wann RMI?
6. Warum sind weder RMI noch CORBA für die Verwendung durch Klienten außerhalb des Unternehmens geeignet?

Kontakt

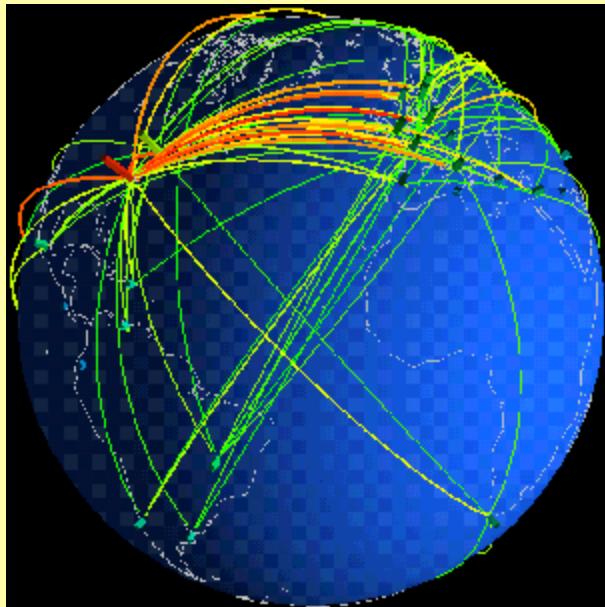
Duale Hochschule BW Ravensburg
Campus Friedrichshafen

Prof. Dr. Andreas Judt
Informationstechnik

judt@dhbw-ravensburg.de

Web Engineering

EJB 2.x und 3.x



Prof. Dr. Andreas Judt

Unternehmensanwendungen sind verteilt

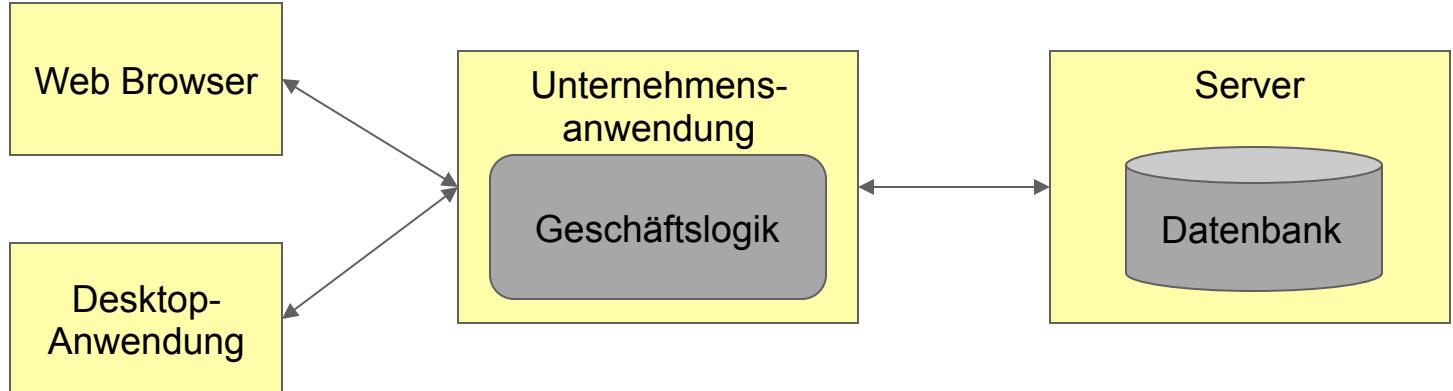
- Man spricht von einer verteilten Anwendung, wenn die Teile des Systems in verschiedenen Adressräumen arbeiten.
- Mögliche Vorteile:
 - System ist flexibel, skalierbar, erweiterbar
 - Lastverteilung
 - Plattformunabhängigkeit
 - Herstellerunabhängigkeit
- Eine Ausprägung verteilter Systeme ist die Dienstnehmer / Dienstgeber (Client / Server) Struktur. Dabei bietet ein Teil des Systems Dienste an, die von anderen Teilen des Systems benutzt werden können.

n-Säulen Modell

n=2



n=3

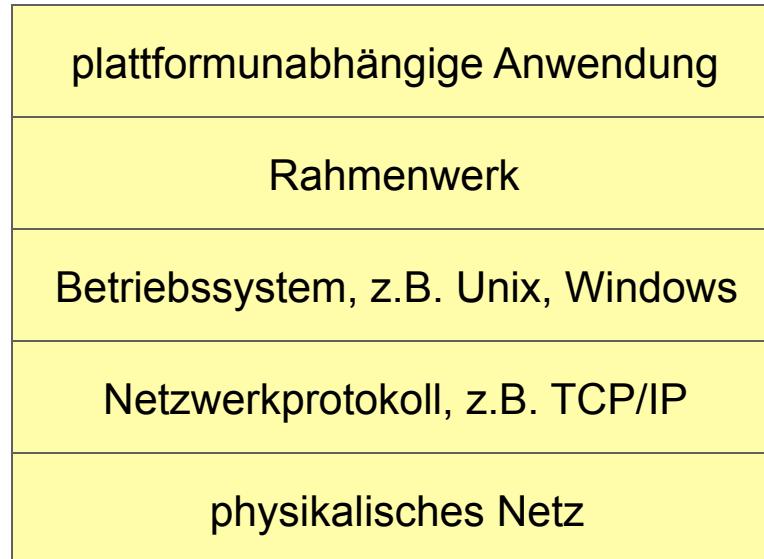


Bewertung des 3-Säulen Modells

- wird auch als 3-Schicht-Modell bezeichnet
- Vorteile
 - flexibel
 - skalierbar (können dynamisch erweitert werden)
 - Leistung und Verlässlichkeit können dynamisch verbessert werden
- Nachteile
 - erhöhte die Komplexität
 - die Leistung des Gesamtsystems muss betrachtet werden
 - viele Fehlerquellen
 - schlechte Werkzeugunterstützung

Struktur einer Unternehmensanwendung

- auch als Middleware bezeichnet



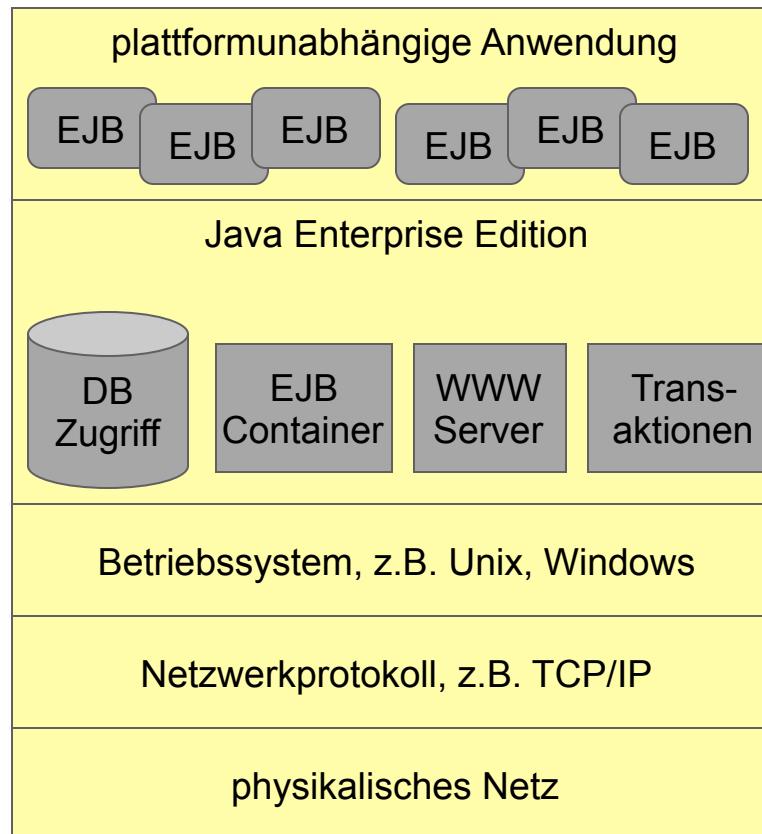
Rahmenwerke

- Rahmenwerk, engl. framework
- Unterschied zur Klassenbibliothek
 - Rahmenwerk bestimmt die Software-Architektur
- Eigenschaften
 - erweiterbares System kooperierender Klassen
 - implementieren einen wiederverwendbaren Entwurf für einen Anwendungsbereich
 - besteht aus konkreten und abstrakten Klassen, die Schnittstellen definieren

Entwurf mit einem Rahmenwerk

- Bildung von Unterklassen, um Rahmenwerk zu verwenden und anzupassen
- Selbst definierte Unterklassen empfangen und verarbeiten Nachrichten aus den Klassen des Rahmenwerks
 - Hollywood-Prinzip: „Don ‘t call us – we ‘ll call you.“
- Software-Architekt kann sich auf die Funktionalität des Zielsystems konzentrieren.

Java Enterprise Edition (Java EE)



JEE Spezifikationen im Wandel

- 1997: erste EJB Spezifikation von IBM
 - Java Professional Edition (JPE)
- 1998: EJB 1.0 (Konferenzbeitrag auf JavaOne)
- 1999: J2EE 1.2, EJB 1.1
 - XML Deployment Deskriptoren
 - RMI über IIOP (CORBA)
 - rollenbasierte Sicherheitsregeln
 - Entity Beans werden Pflicht
 - Java Database Connectivity (JDBC) 2.0
 - Java Naming and Directory Interface (JNDI) 1.2

JEE Spezifikationen im Wandel

- 2001: J2EE 1.3 (JSR 58)
 - EJB 2.0
 - Java Transaction API /Service
 - J2EE Connector Architecture
 - Java API for XML Processing (JAXP)
 - Java Authentication and Authorization Service (JAAS)

JEE Spezifikationen im Wandel

- 2003: J2EE 1.4 (JSR 151)
 - EJB 2.1
 - Java Server Pages (JSP) 2.0
 - JDBC 3.0
 - Web Services für J2EE inkl. SOAP und XML Registries
 - EJB Query Language (EJB-QL)
 - XML Schema statt DTD
 - EJB Timer Service
 - Message Driven Beans kommunizieren technologienutral

JEE Spezifikationen im Wandel

- 2006: JEE 5 (JSR 244)
 - EJB 3.0
 - Annotationen statt Deployment Deskriptoren
 - Java Server Faces (JSF)
 - SOAP with Attachments API for Java (SAAJ)
 - Java Persistence API (JPA)
 - JDBC 3.0
 - Streaming API for XML (StAX)

JEE Spezifikationen im Wandel

- 2009: JEE 6 (JSR 316)
 - EJB 3.1
 - Java API für RESTful Web Services (JAX-RS)
 - Enterprise Web Services
 - Java API for XML-Based Web Services (JAX-WS)
 - Java Architecture for XML Binding (JAXB)
 - Web Services Metadata for Java Platform
 - Java API for XML-based RPC (JAX-RPC)
 - Java APIs for XML Messaging
- Mit höheren Spezifikationen stiegen die Streitigkeiten um Lizenzvereinbarungen und Nutzungsbedingungen.

Problem der Spezifikationen: Komplexität macht Technologie unbrauchbar!

- Das Programmiermodell bis EJB 2.1 war extrem komplex und erforderte viel Erfahrung mit Softwareentwicklung, speziell bei verteilten Systemen.
 - Es gab kaum ausgereifte Entwicklungswerzeuge auf dem Markt.
 - Je mehr JEE Projekte implementiert wurden, desto größer wurde die Kritik an der Technologie.

Kritik an EJB 2.1

- Hohes Maß an Komplexität
 - Zusammenspiel von Komponenten in XML-Deskriptoren
 - Zu implementierende Methoden wurden in der Spezifikation nicht vollständig definiert.
- Viele Fehlerquellen, Ursachen oft nicht nachvollziehbar
 - z.B. Fehler in XML Deskriptoren oder fehlende Methoden
 - Lösung oft nur durch Erfahrungsaustausch in Foren

Kritik an EJB 2.1

- Hohe Entwicklungsaufwände
 - Enorme Implementierungsaufwände selbst bei kleinen Architekturen
 - Kritische Kostenfaktoren bei großen Projekten
- Unzureichende Persistenzmechanismen
 - Speicherung von Daten über mehrere Tabellen ist automatisiert (CMP) kaum oder nur mit großem Aufwand implementierbar.
 - Häufig mussten die Entwickler eigene Lösungen herstellen (BMP).

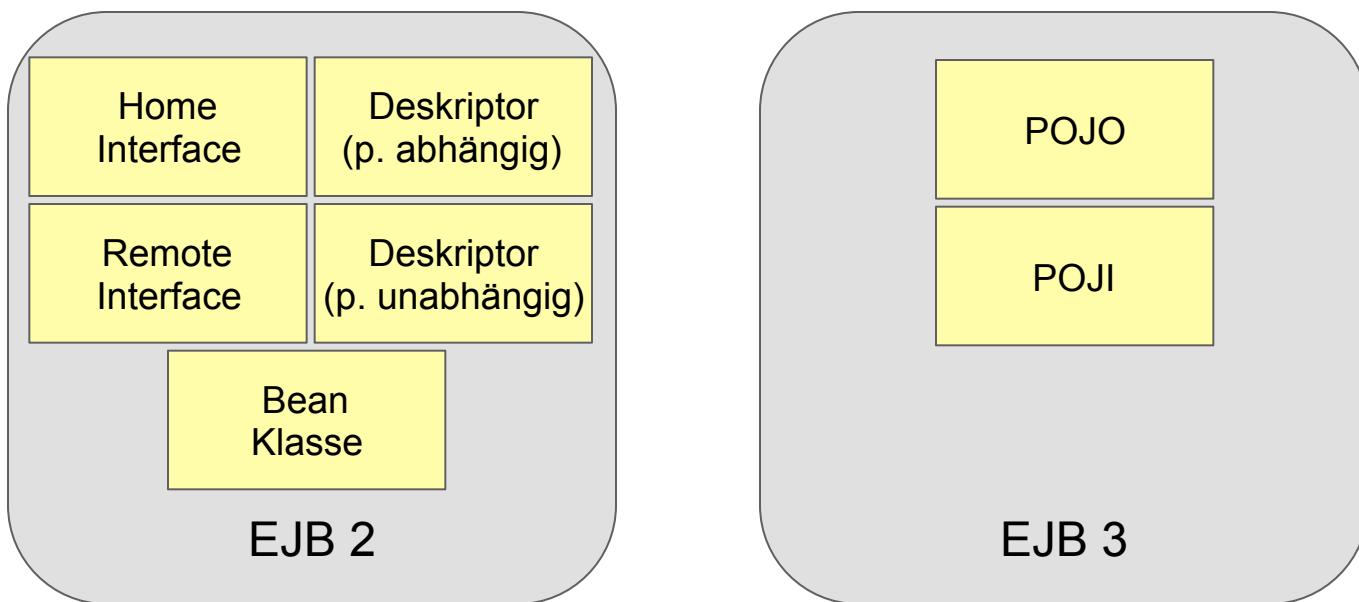
Vereinfachungen der Spezifikation 2.1

- Geringere Komplexität der Mikroarchitektur
 - POJO = plain old java objects
 - POJI = plain old java interfaces
 - Reduzierung der Bestandteile einer EJB
- Annotationen statt Deskriptoren
 - weniger XML-Dateien
 - Wo werden produktspezifische Informationen platziert?
- Abwärtskompatibilität
 - alte Implementierungen nicht modernisieren!

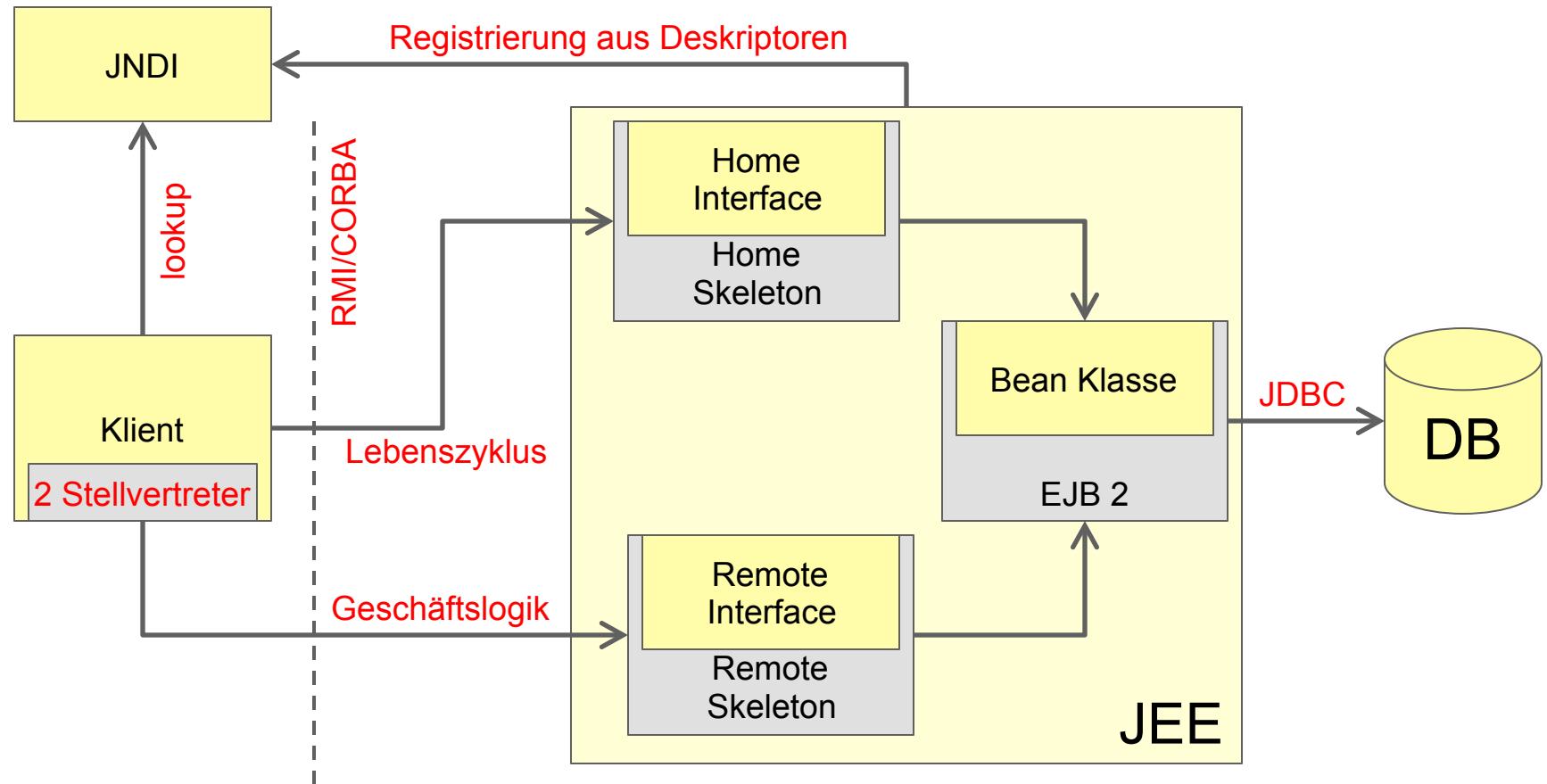
Vergleich der Mikroarchitekturen EJB 2 und EJB 3

- EJB 2...
 - hat mindestens 5 Bestandteile: Home Interface, Remote Interface, Bean Klasse, plattformunabhängiger Deployment Deskriptor (ejb-jar.xml), plattformabhängiger Deployment Deskriptor (z.B. jboss.xml)
 - fordert eine Vielzahl von Methoden, z.B. für den Lebenszyklus
- EJB 3...
 - reduziert sich auf POJO und POJI
 - macht viele Lebenszyklusmethoden überflüssig
 - Geschäftsmethoden lassen sich leichter testen

Mikroarchitekturen EJB 2 und 3



Verwendung einer EJB 2

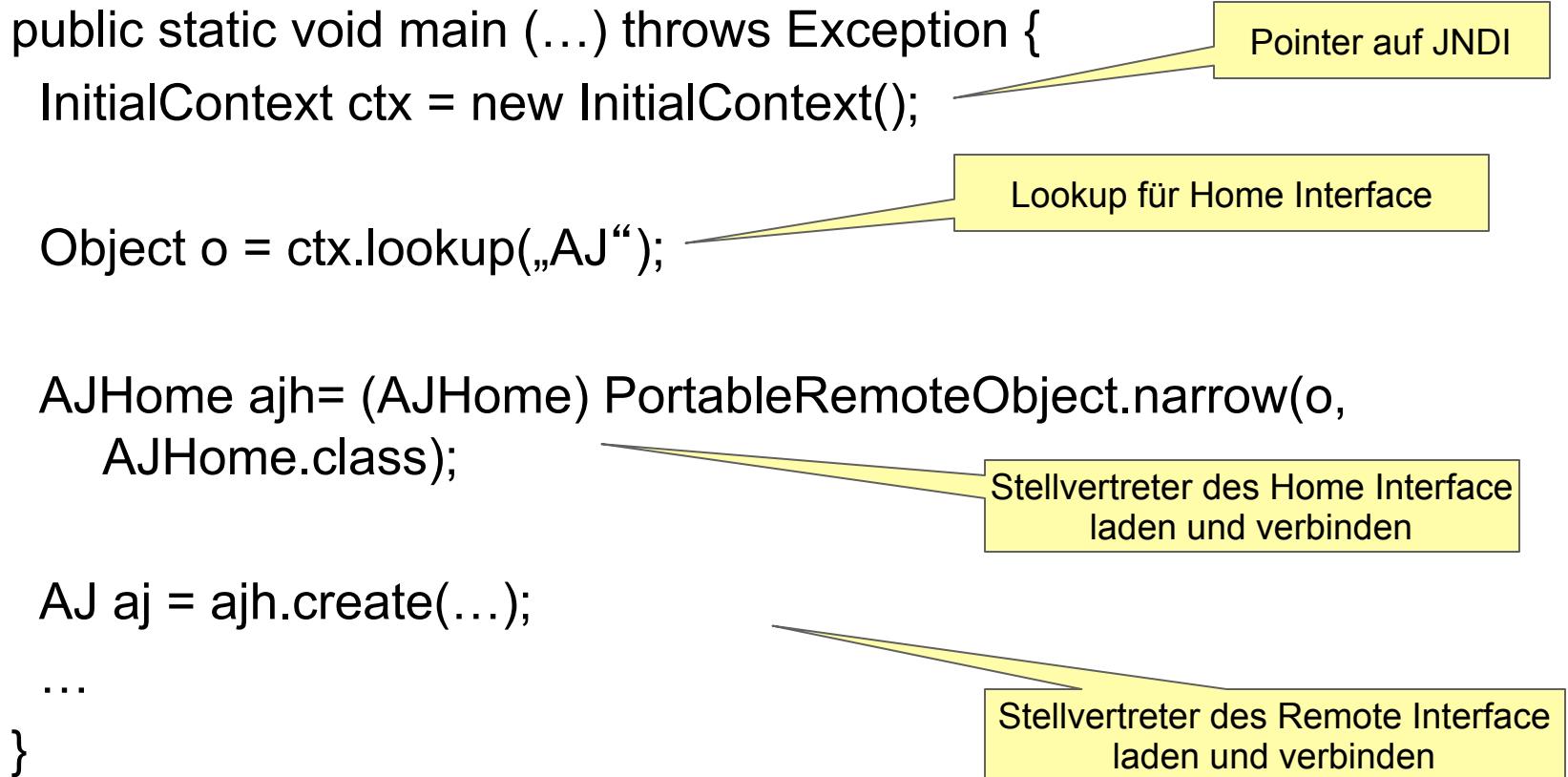


Verwendung einer EJB 2

1. JEE registriert eine EJB bei JNDI während der Installation (deployment). Die Deployment Deskriptoren enthalten die notwendigen Informationen.
2. Klient sucht EJB über logischen Namen (lookup) bei JNDI.
 - Name muss dem Klienten bekannt sein.
 - Ergebnis ist eine Stellvertreterklasse (Stub) des Home Interface, von der lookup ein Objekt erzeugt. Der Klient muss den Bytecode des Stub laden können!
3. Klient erstellt ein oder mehrere Stellvertreter von EJB (create / find). Sie stellen die Methoden der Geschäftslogik zur Verfügung.

Codebeispiel: Zugriff auf eine EJB 2

```
public static void main (...) throws Exception {  
    InitialContext ctx = new InitialContext();  
  
    Object o = ctx.lookup(„AJ“);  
  
    AJHome ajh= (AJHome) PortableRemoteObject.narrow(o,  
        AJHome.class);  
  
    AJ aj = ajh.create(...);  
  
    ...  
}
```



Annotations:

- Annotation 1: "Pointer auf JNDI" points to the line `InitialContext ctx = new InitialContext();`
- Annotation 2: "Lookup für Home Interface" points to the line `Object o = ctx.lookup(„AJ“);`
- Annotation 3: "Stellvertreter des Home Interface laden und verbinden" points to the line `AJHome ajh= (AJHome) PortableRemoteObject.narrow(o, AJHome.class);`
- Annotation 4: "Stellvertreter des Remote Interface laden und verbinden" points to the line `AJ aj = ajh.create(...);`

Codebeispiel: Zugriff auf eine EJB 3

```
public static void main (...) throws Exception {
```

```
    InitialContext ctx = new InitialContext();
```

Pointer auf JNDI

```
    AJ aj = ctx.lookup(„AJ“);
```

```
    ...
```

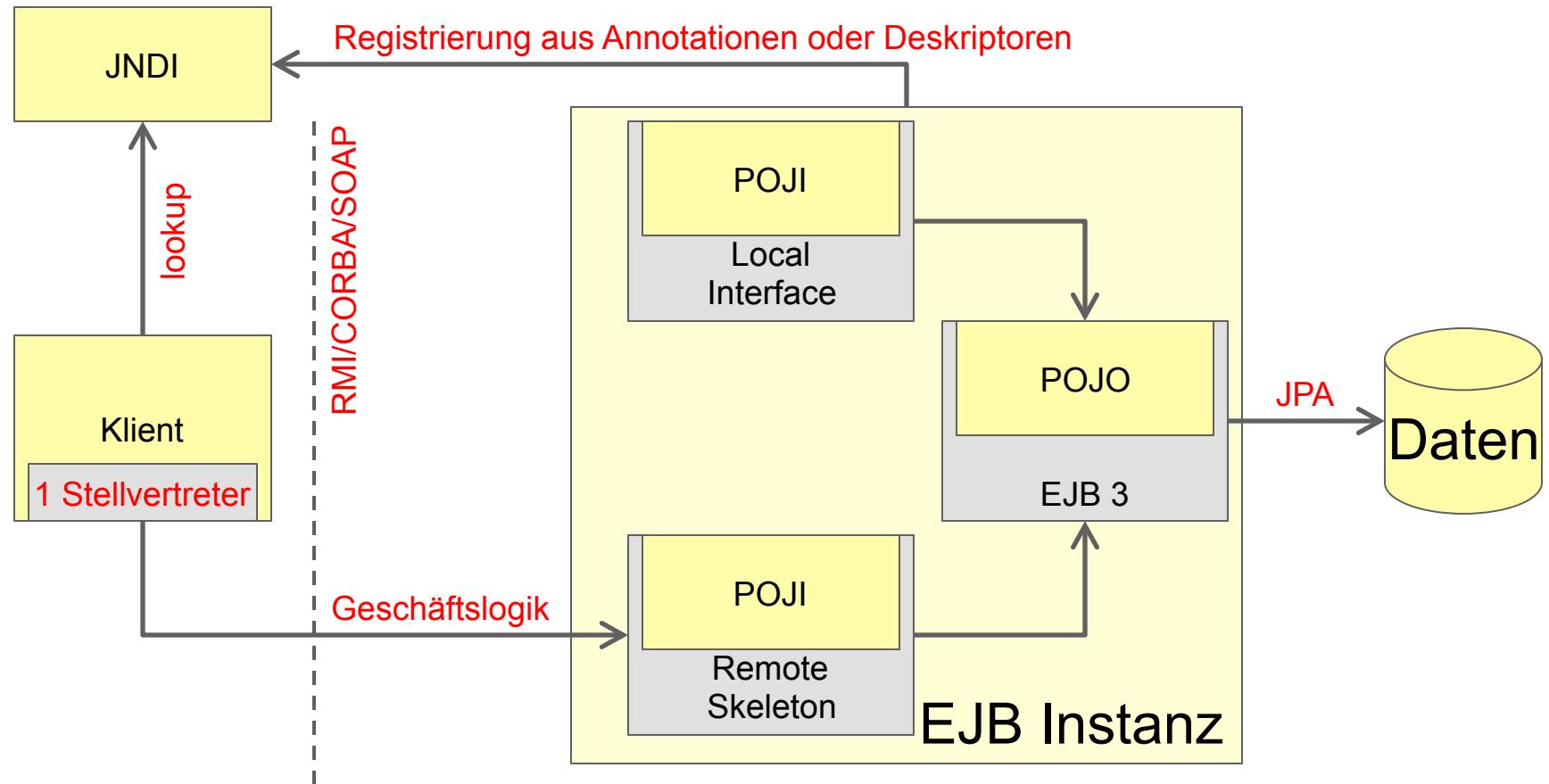
```
}
```

Lookup und Stellvertreter
der EJB
laden und verbinden

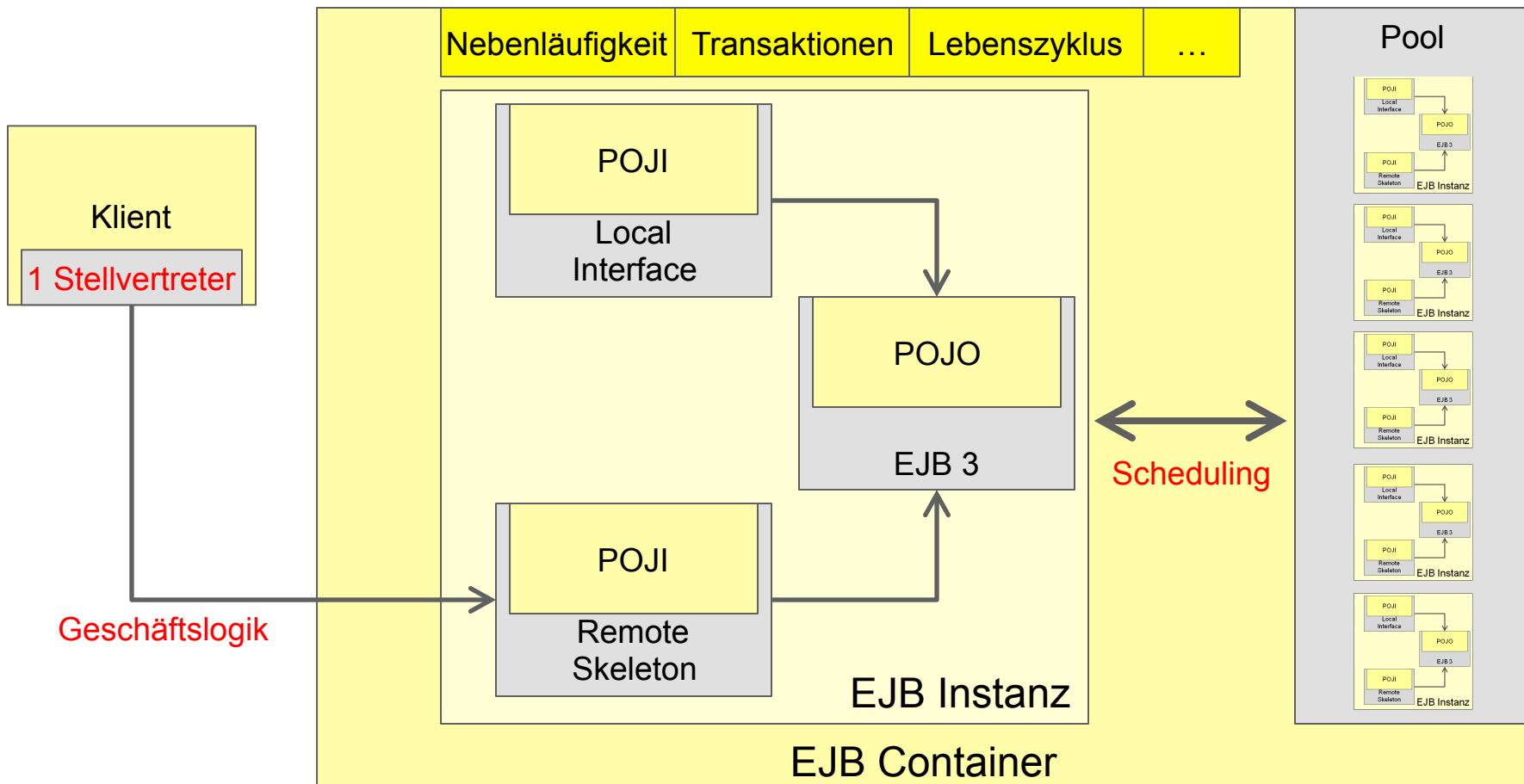
Verwendung einer EJB 3

1. JEE registriert eine EJB bei JNDI während der Installation (deployment). Die Annotationen im Quellcode oder alternativ die Deployment Deskriptoren (abwärtskompatibel!) enthalten die notwendigen Informationen.
 2. Klient sucht EJB über logischen Namen (lookup) bei JNDI.
 - Name muss dem Klienten bekannt sein.
 - Ergebnis ist eine Stellvertreterklasse (Stub) des Remote Interface, von der lookup ein Objekt erzeugt. Der Klient muss den Bytecode des Stub laden können!
- Finder-Methoden werden über die Java Persistence Query Language (JPQL) implementiert.

Verwendung einer EJB 3



EJB Container



EJB Container

- Der EJB Container bildet die Laufzeitumgebung für die EJB-Instanzen.
 - Er überwacht die Ausführung der jeweiligen EJB Instanzen.
 - Dazu benötigte Funktionen bietet er selbst oder nutzt Schnittstellen des JEE Servers.

EJB Container

- Die wichtigsten Funktionen und Schnittstellen...
 - JNDI Namensdienst
 - Persistenzmechanismen
 - Transaktionssteuerung
 - Nebenläufigkeit / konkurrierende Zugriffe
 - Zugriffsrechte auf EJB-Instanzen und deren Methoden
 - Steuerung des Lebenszyklus von EJB-Instanzen
 - Pools von EJB-Instanzen

JNDI Namensdienst

- Ein Klient nutzt JNDI, um einen Stellvertreter für eine EJB-Instanz zu erhalten.
 - Der Namensdienst wird über das Laden einer Klasse erzeugt und mit geeigneten Parametern (Properties) parametrisiert, u.a.
 - Klassenname
 - Rechnername
 - Port
 - Die Instanziierung kann durchaus herstellerabhängig sein und sollte modular im Code verwendet werden.

Persistenzmechanismen

- Bei EJB 2 wurden Daten mit Entity Beans gespeichert.
 - Container Managed Persistence (CMP)
 - Bean Managed Persistence (BMP)
- Ab EJB 3 ist der EJB Container nicht mehr für die Persistenz zuständig.
 - Die Implementierung erfolgt durch das Java Persistence API.
 - Hier können deutlich flexibler als mit JDBC viele Technologien verwendet werden.
 - z.B. Hibernate bei Jboss
 - z.B. TopLink bei Oracle

EJB 2: CMP vs. BMP bei Entity Bean

- Container Managed Persistence (CMP)
 - Container erzeugt eine Speicherung in der Datenbank per JDBC aus Quellcode und Deployment Deskriptor bzw. Annotationen.
 - SQL Datenbank wird dabei automatisch erzeugt.
- Bean Managed Persistence (BMP)
 - Entwickler der EJB muss sich selbst um die Speicherung der Daten kümmern.
 - Es kann eine bestehende SQL Datenbank (z.B. Altsystem) weiter verwendet werden.
- Transaktionsmechanismus kann ebenfalls Container Managed (CMT) oder Bean Managed (BMT) implementiert werden.

Transaktionssteuerung

- Transaktionsverwaltung wird EJB als Dienstleistung von der JEE-Implementierung bereitgestellt.
- Transaktionsattribute werden deklarativ über Annotationen oder Deployment Deskriptor eingestellt.
- Die Steuerung von Transaktionen erfolgt durch das Fangen (catch) bzw. Werfen (throw) von Fehlern (exceptions).

Nebenläufigkeit / konkurrierende Zugriffe

- Konkurrierende Zugriffe werden abhängig vom Typ der EJB geregelt.
- Stateful Session Bean
 - exklusiv einem Klienten zugeordnet
- Stateless Session Beans /Message Driven Bean
 - einem Klienten nur für einen Methodenaufruf zugeordnet
- Singleton Session Bean
 - abhängig von der Konfiguration oder POJO-Implementierung
- Persistent Entities (früher Entity Bean)
 - abhängig von Transaktionsattributen, vom Persistence Provider implementiert

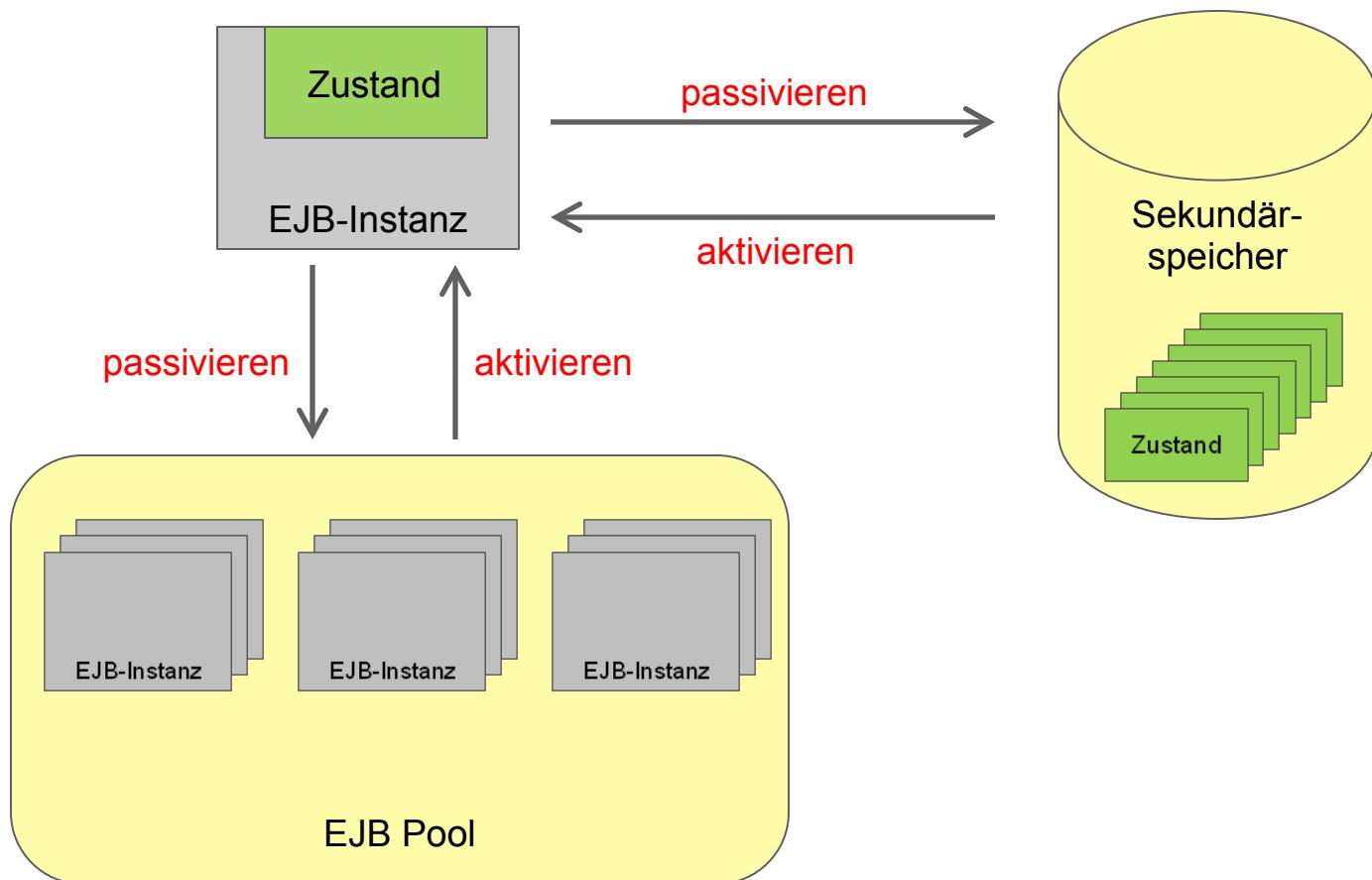
Zugriffsrechte auf EJB-Instanzen und deren Methoden

- Zugriffsrechte auf EJB werden durch einen Rollenmechanismus implementiert.
 - Methoden in EJB werden durch Annotation oder Deployment Deskriptor konfiguriert.
- EJB Container wertet Zugriffsrechte zur Laufzeit aus.
 - implementiert im Java Authentication and Authorization Service (JAAS)

Steuerung des Lebenszyklus von EJB-Instanzen

- Der EJB Container übernimmt die Steuerung des Lebenszyklen für jeden EJB-Typen.
- Erzeugen
 - Zahl freier Instanzen liegt unter einem konfigurierten Limit.
- Löschen
 - Zahl freier Instanzen liegt über einem konfigurierten Limit.
- Passivieren
 - EJB-Zustand wird von der Instanz getrennt und auf einen Sekundärspeicher ausgelagert.
- Aktivieren
 - EJB-Zustand wird von einem Sekundärspeicher auf eine Instanz geladen.

Instanzen und Zustände trennen: aktivieren und passivieren



Pools von EJB-Instanzen

- EJB Pools ermöglichen die Wiederverwendung von Objekten.
- Der größte Nachteil der OO wird in einen Vorteil gewandelt.
- EJB Pools ermöglichen
 - Skalierung über mehrere Server
 - geschickte Lastverteilung
 - sparsame Nutzung von Ressourcen
 - Sehr schnelle Reaktionszeiten

Typen von EJB

- Die EJB Spezifikation 3.1 unterscheidet im wesentlichen 3 Typen:
 - Session Beans
 - Message-Driven Bean
 - Persistent Entity

Session Beans

- Session Beans sind Bestandteile der Geschäftslogik und werden nicht persistiert.
 - Ausschließlich Persistent Entities speichern Daten.
- Typen:
 - Stateful, Stateless, Singelton
- Transaktionsverhalten
 - Session Beans können in Transaktionen laufen, sofern der Entwickler das vorgesehen hat.

Session Beans

- Datenbankzugriffe
 - Session Beans können auf Datenbanken zugreifen, sollten aber nicht selbst persistieren.
- Session Beans als Web Services
 - Session Beans können über eine Web Service Schnittstelle zur Verfügung gestellt werden.

Stateless Session Beans (SLSB)

- Stateless Session Beans besitzen keinen Zustand
 - also keine Attribute
 - Weitergabe aus einem Pool durch Bereitstellen eines Zeigers
 - können als Funktionsbibliothek angesehen werden
- Bewertung
 - Vorteile
 - Stateless Session Beans sind hochleistungsfähig.
 - Passivieren / aktivieren ist nicht erforderlich.
 - Nachteile
 - Das Fehlen des Zustands schränkt die Einsetzbarkeit stark ein.

Stateful Session Beans (SFSB)

- Eine Stateful Session Beans wird für einen Klienten erzeugt und bleibt diesem bis zur Vernichtung zugeordnet.
 - Zustand wird durch Attributwerte definiert
 - Instanzen werden in einem Pool optimiert, EJB werden dazu aktiviert und passiviert.
- Bewertung
 - Vorteile
 - Sehr einfache Organisation von Daten und zugehöriger Algorithmik
 - Nachteile
 - Pools bringen Skalierbarkeit, jedoch viel weniger Performanz als Stateless Session Beans

Singleton Session Bean (SSB)

- Singleton Session Beans gibt es seit EJB 3.1.
- Er gibt maximal eine Instanz einer SSB.
 - Sie kann von mehreren Klienten gleichzeitig verwendet werden.
- Der typische Einsatz ist das Speichern globaler Informationen.
- Bewertung
 - Vorteile
 - Singleton Session Beans bieten zentrale Speichermöglichkeiten, ähnlich wie globale Variable.
 - Nachteile
 - Die Vermeidung konkurrierender Zufälle muss explizit implementiert werden (vgl. Schlüsselwort synchronized).

Message Driven Beans (MDB)

- Message Driven Beans ermöglichen die asynchrone Kommunikation zwischen JEE-Implementierungen.
 - Die Kommunikation erfolgt über Nachrichten, die von einem Nachrichtenvermittler (Message Broker) vermittelt werden.
- MDB können über Java Message Service (JMS) oder als Web Service benutzt werden.
 - Eine MDB ist zustandslos wie eine SLSB, kann aber einen internen Zustand besitzen.
- Über Deployment Diskriptor oder Annotationen kann das Transaktionsverhalten definiert werden.

Message Driven Beans (MDB)

- Ein Zugriff per JDBC auf eine Datenbank ist möglich, jedoch persistieren MDB nicht selbst.
- Bewertung
 - Lang laufende Prozesse können zeitlich entkoppelt werden.
 - Fremdsysteme (sog. Legacy Systeme) mit nachrichtenbasierter Kommunikation können in die JEE-Anwendung integriert werden.

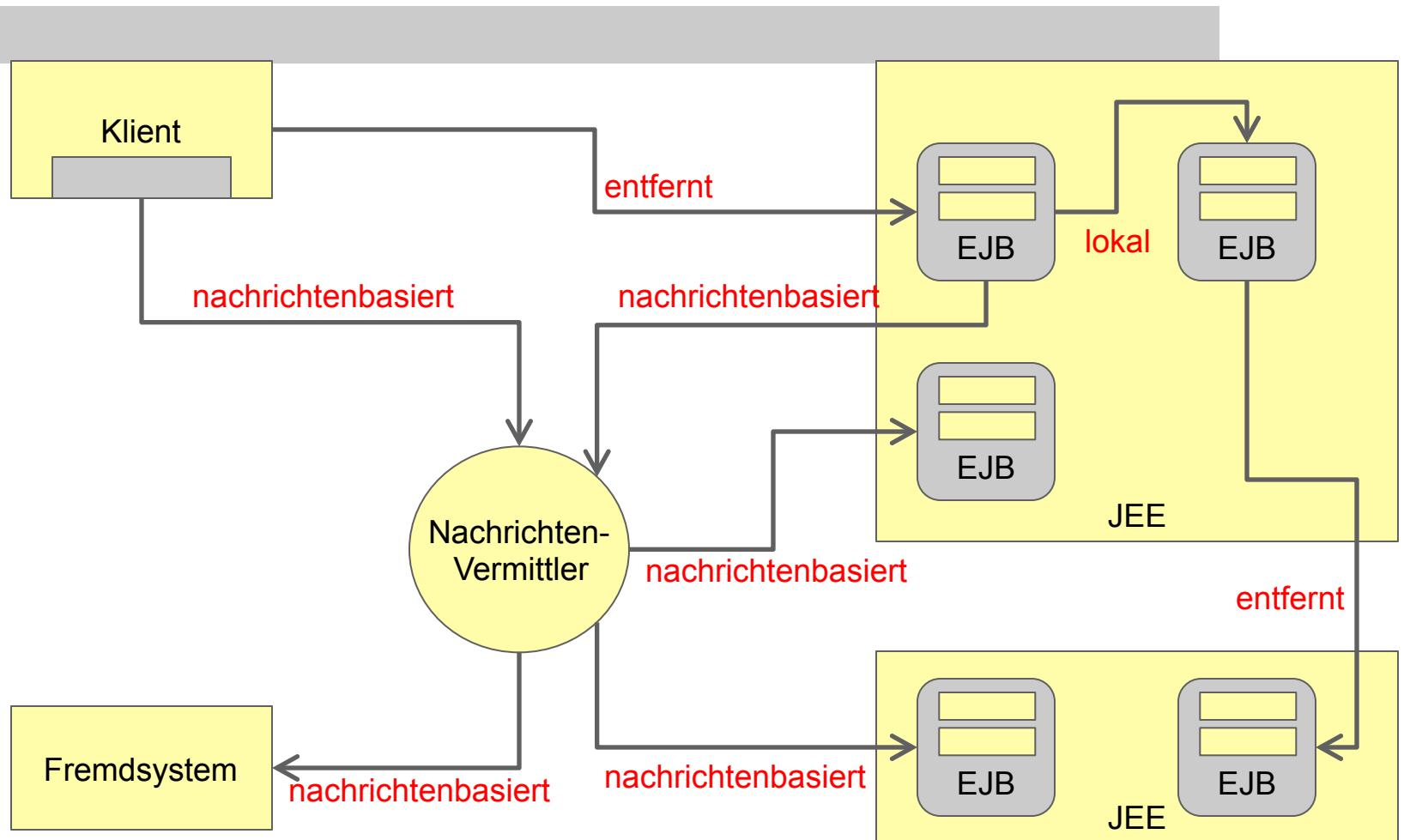
Persistent Entity

- Eine Persistent Entity implementiert eine objektorientierte Schnittstelle zu einem Datensatz, der typischerweise in einer Datenbank gespeichert wird.
 - Das Persistieren der Daten wird vom Entity Manager übernommen.
 - Alternativ kann SQL über JDBC im Quellcode der EJB implementiert werden (vgl. Bean Managed Entity Beans, Abwärtskompatibilität zu EJB 2).
- Persistent Entities können an Transaktionen teilnehmen.

Aufrufmodelle zur Verteilung einer JEE Applikation

- JEE kennt drei Aufrufmodelle für EJB 3.1
 - lokal (synchron, asynchron)
 - entfernt (synchron, asynchron)
 - nachrichtenbasiert (asynchron)
- Motivation für eine Verteilung
 - Lastverteilung
 - Steigerung der Gesamtrechenleistung
 - Zusammenbau neuer Applikationen aus bestehenden Teilsystemen
 - Integration von Altsystemen in neue Unternehmensanwendungen

Aufrufmodelle zur Verteilung einer JEE Applikation



Asynchrone Kommunikation bei Session Beans

- Asynchrone Kommunikation erlaubt Session Beans das Abrufen von Ergebnissen zu einen späteren Zeitpunkt.
 - Wird über eine Annotation (@Asynchronous) im Quellcode eingestellt.
 - Methodenaufruf ist dann ergebnislos.
 - Codeerzeugung liefert Methoden get und isDone, cancel, isCancelled zur Steuerung der Methodenbearbeitung durch den Aufrufer.
- Transaktionen dürfen nur container-managed verwendet werden.

Aufrufmodelle zur Verteilung einer JEE Applikation

EJB-Typen	Aufrufmodelle		
	entfernt	lokal	nachrichtenbasiert
Singleton Session Bean	synchron, asynchron	synchron, asynchron	
Stateless Session Bean	synchron, asynchron	synchron, asynchron	
Stateful Session Bean	synchron, asynchron	synchron, asynchron	
Message Driven Bean			asynchron
Persistent Entity		synchron	

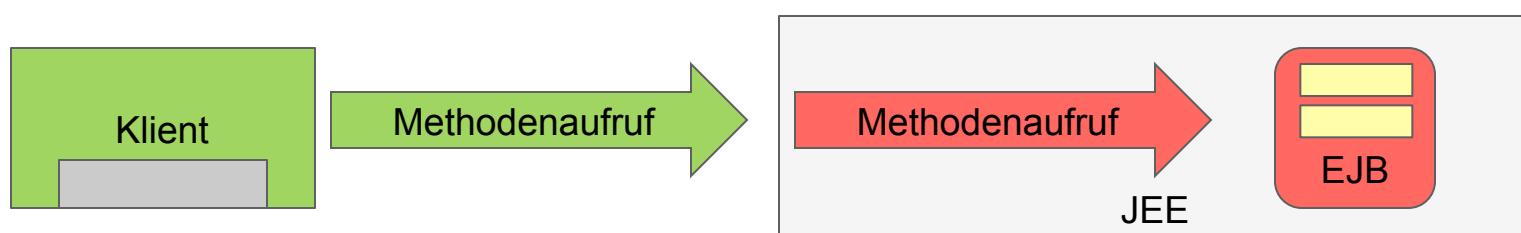
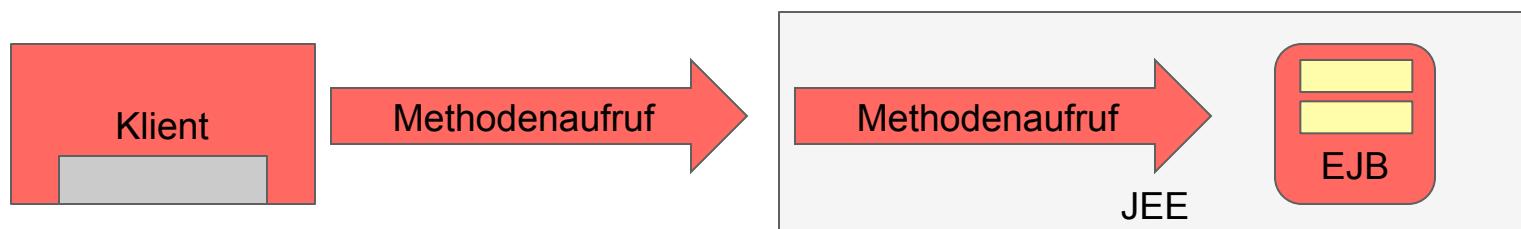
Transaktionsmechanismen: Container-Managed

- Die deklarative Einstellung des Transaktionsverhaltens von EJB (Container-Managed Transaction Demarcation) ist eine wesentliche Eigenschaft bei der Komposition von EJB.
 - Transaktionsattribute werden per Annotation oder Deployment Deskriptor eingestellt.
 - Die Steuerung im Programm erfolgt durch das Werfen und Verarbeiten von Exceptions (sog. Configuration by Exception).
 - Standardeinstellung: Required

Übersicht der Transaktionsattribute

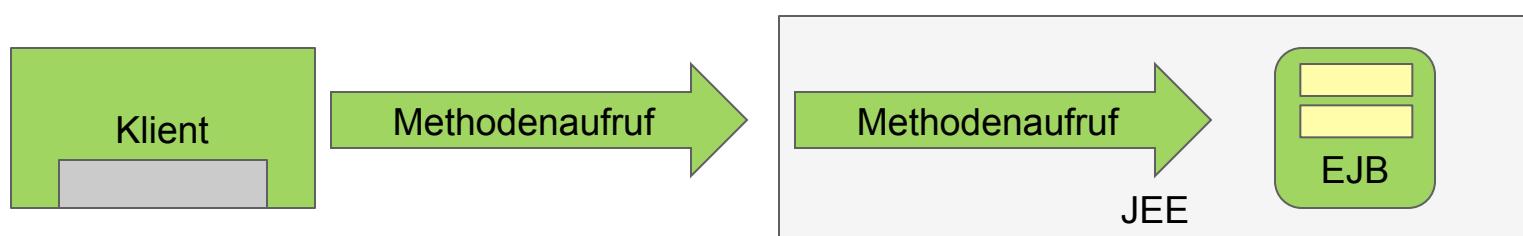
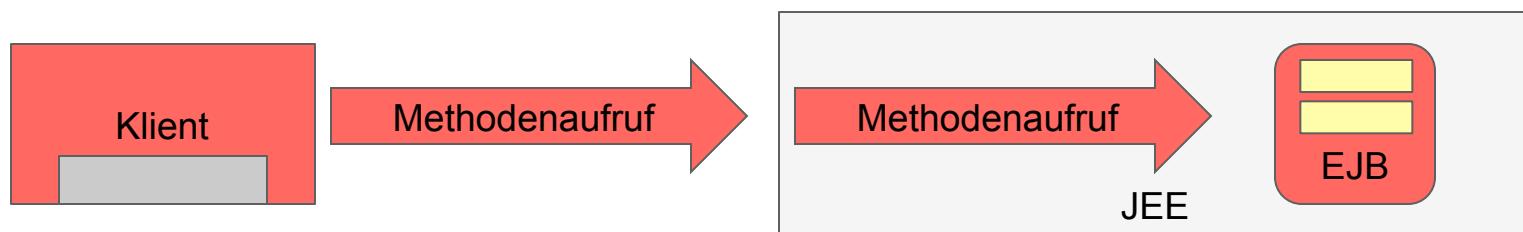
Transaktionsattribut / Annotation	Beschreibung
NotSupported / NOT_SUPPORTED	Die Methode wird nicht in die laufende Transaktion integriert.
Supports / SUPPORTS	Die Methode wird in die laufende Transaktion integriert.
Required / REQUIRED	Die Methode muss in einer Transaktion laufen.
RequiresNew / REQUIRES_NEW	Für die Methode wird eine neue Transaktion gestartet.
Mandatory / MANDATORY	Der Aufrufer der Methode muss in einer Transaktion laufen.
Never / NEVER	Der Aufrufer der Methode darf nicht in einer Transaktion laufen.

Transaktionsattribut NotSupported / NOT_SUPPORTED



Transaktion 1 Transaktion 2 Keine Transaktion

Transaktionsattribut Supports / SUPPORTS

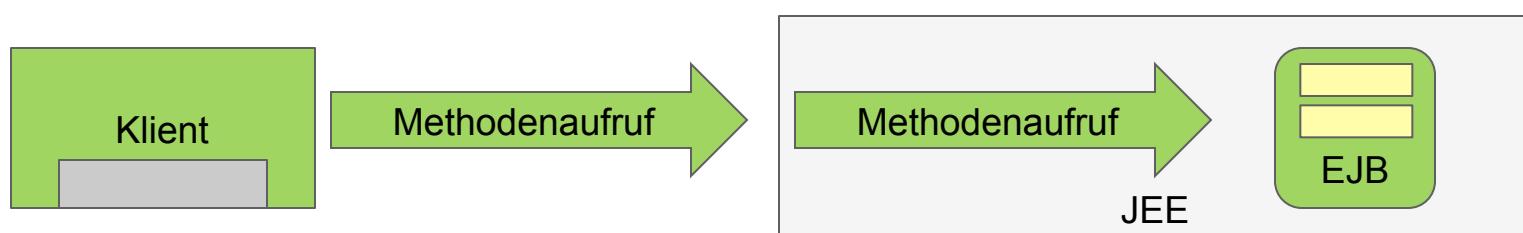
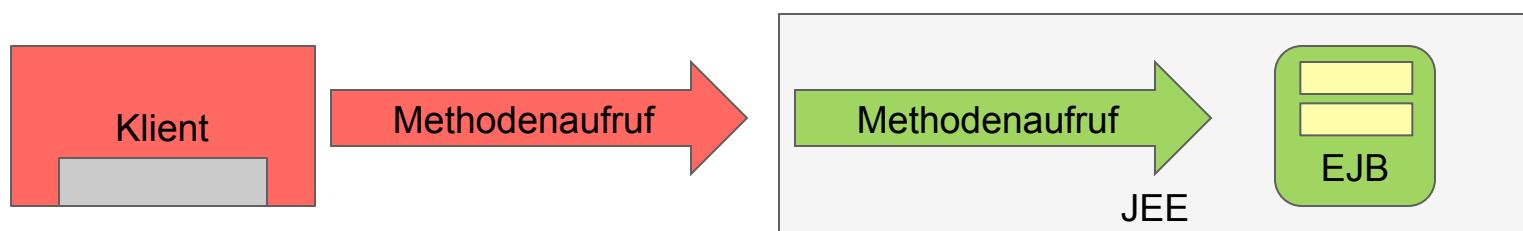


Transaktion 1

Transaktion 2

Keine Transaktion

Transaktionsattribut Required / REQUIRED

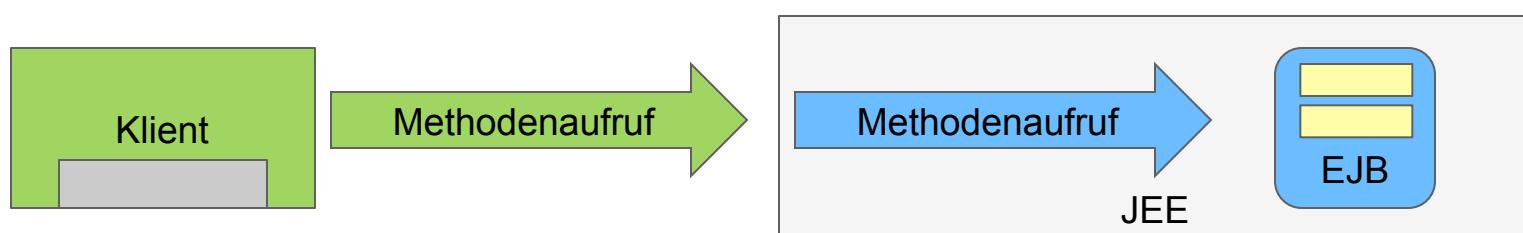
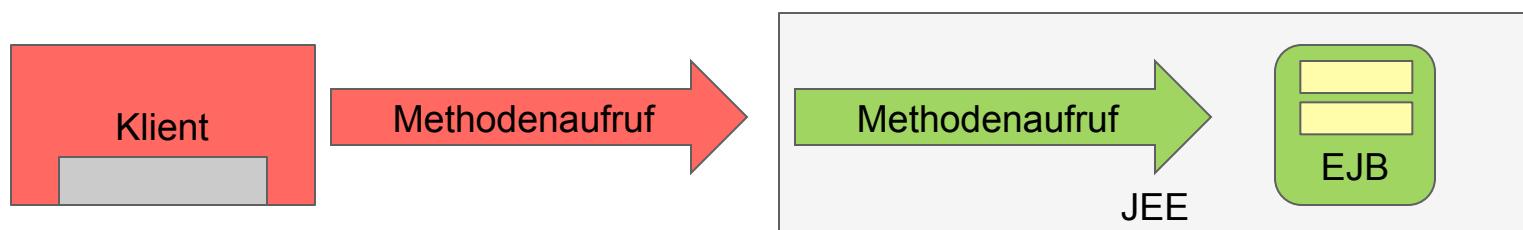


Transaktion 1

Transaktion 2

Keine Transaktion

Transaktionsattribut RequiresNew / REQUIRES_NEW

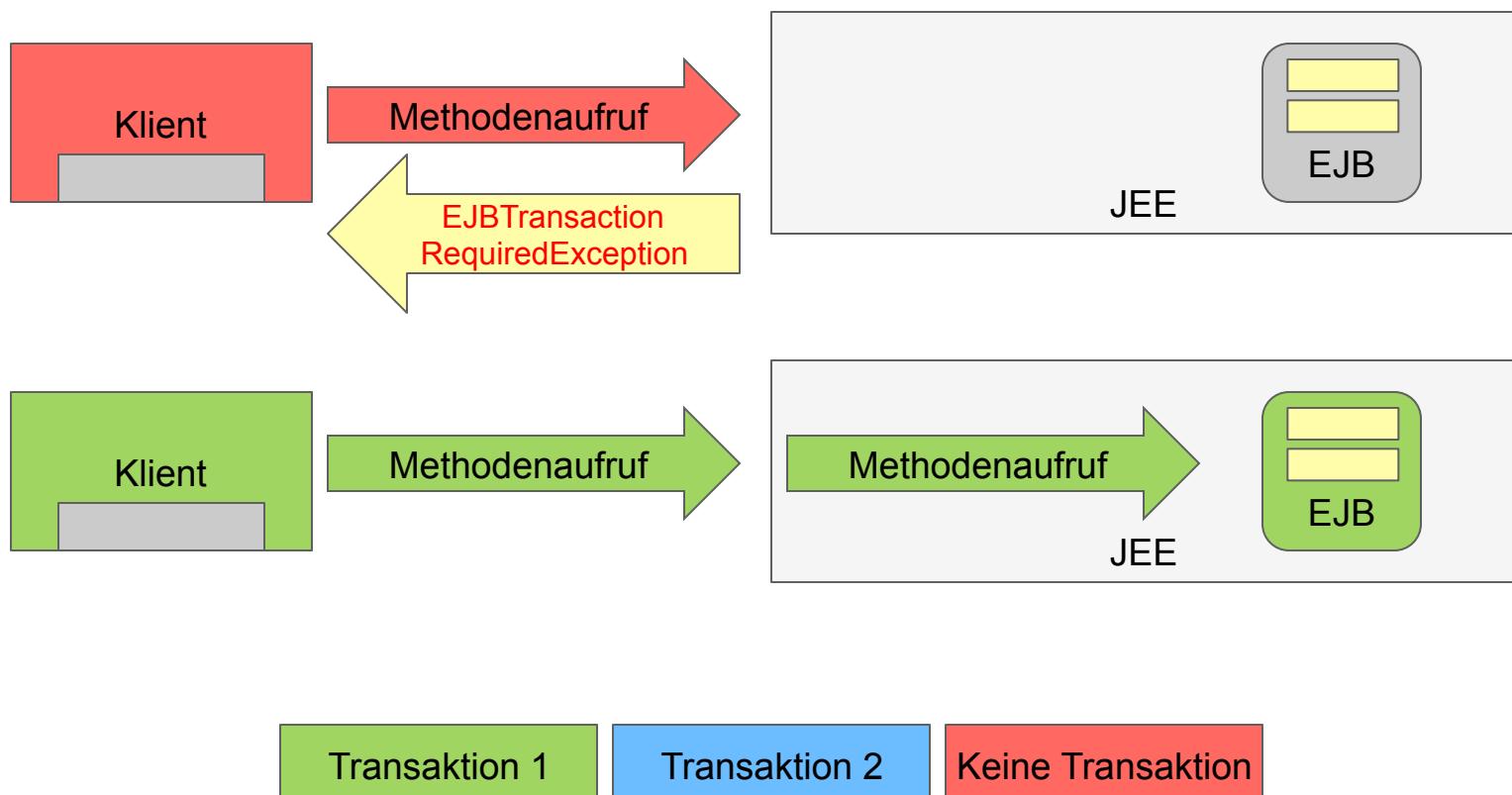


Transaktion 1

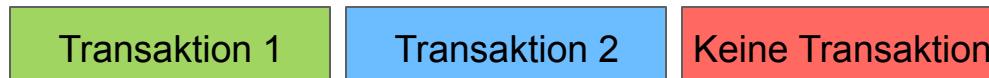
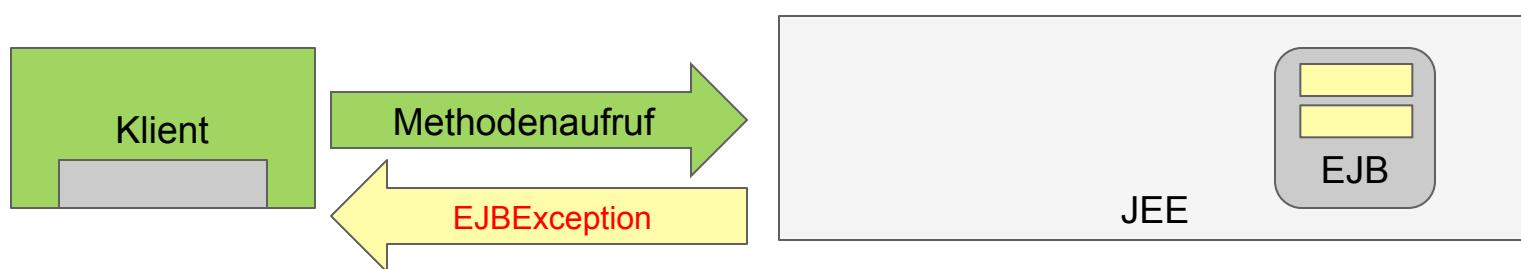
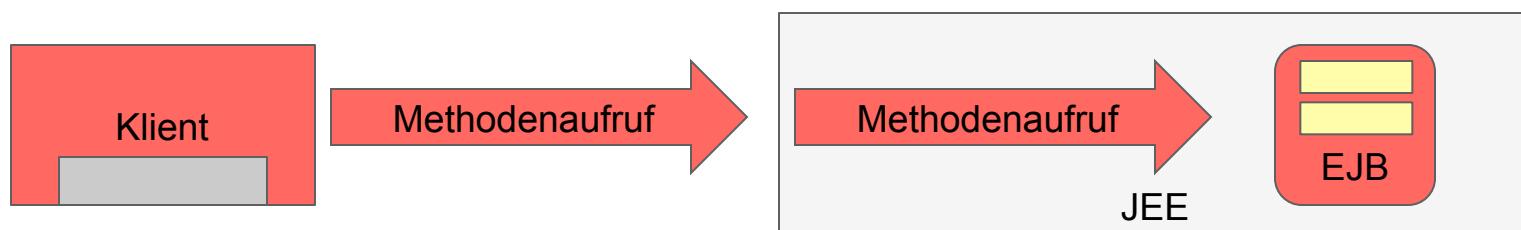
Transaktion 2

Keine Transaktion

Transaktionsattribut Mandatory / MANDATORY



Transaktionsattribut Never / NEVER



Transaktionsmechanismen: Bean-Managed

- Im Vergleich zu Container-Managed Transaction Demarcation muss der Entwickler den Transaktionsmechanismus selbst implementieren.
 - Für eigene Implementierungen steht die Java Transaction API (JTA) zur Verfügung.
 - Technisch sind benutzerdefinierte Transaktionen Implementierungen der Schnittstelle UserTransaction.
 - Eigene Transaktionen können bei der Integration von Fremdsystemen erforderlich sein.
- Transaktionsattribut: Bean / BEAN

Synchronisation konkurrierender Zugriffe

- Für die Verlässlichkeit von verteilten Systemen muss das ACID-Prinzip eingehalten werden.
 - Abgeschlossenheit (Atomicity)
 - Eine Operation wird immer vollständig oder gar nicht ausgeführt.
 - Konsistenzherhaltung (Consistency)
 - Die Beendigung einer Operation hinterlässt einen vorher konsistenten System wieder in einem konsistenten Zustand.
 - Abgrenzung (Isolation)
 - Der Isolationsgrad beschränkt den Einfluss nebenläufig rechnender Operationen.
 - Isolationsgrade: READ_COMMITTED, READ_UNCOMMITTED, REPEATABLE_READ, SERIALIZABLE

Synchronisation konkurrierender Zugriffe

- ACID Prinzip (Fortsetzung)
 - Dauerhaftigkeit (Durability)
 - Nach dem erfolgreichen Ablauf einer Operation muss das Ergebnis dauerhaft verfügbar sein.
 - Die Dauerhaftigkeit bezieht sich auf den Systemzustand oder auf die Persistierung von Daten.
- Mögliche Effekte bei Verletzung des ACID-Prinzips:
 - Verlorene Updates
 - Bei parallelen Schreibzugriffen werden Daten unkontrolliert überschrieben.

Synchronisation konkurrierender Zugriffe

- Effekte (Fortsetzung)
 - Phantomproblem
 - Verwendete Daten können sich währende der Transaktion durch einen konkurrierenden Zugriff ändern.
 - Schreib-Lese Konflikt
 - Eine Transaktion liest Daten, die von einer anderen Transaktion geschrieben, aber noch finalisiert (commit) sind.
 - Nichtwiederholbares Lesen
 - Eine doppelt ausgeführte Leseoperation liefert innerhalb einer Transaktion verschiedene Ergebnisse

Synchronisation konkurrierender Zugriffe

- Synchronisation Container-Managed
 - Der Container bietet einen Sperrmechanismus an, der durch konfigurierbare Zeitschränke abgebrochen werden kann.
 - LockType.READ: beliebige nebenläufige Zugriffe
 - LockType.WRITE: exklusive Sperre für einen Zugriff
 - Einstellung über Annotationen oder Deployment Deskriptor
- Synchronisation Bean-Managed
 - Der Entwickler verwendet Mechanismen der Sprache Java.
 - synchronized, volatile: verschieden implementierte Sperrmechanismen

Bewertung

- JEE 6 und EJB 3.1 sind wesentlich umfangreicher, als in diesem Kapitel dargestellt.
- Verfügbare Technologien wie Dependency Injection sind nette Programmierwerkzeuge, jedoch werden große Softwaresysteme schnell unüberschaubar und im schlechtesten Fall unkontrollierbar.
- Der Umfangs der zur Verfügung stehenden APIs ist nur schwer überschaubar.

Bewertung

- EJB 3 wurde gegenüber EJB 2 erheblich vereinfacht.
- Der hohe Grad der Codegenerierung erfordert ein weitreichendes Verständnis der Technologie.
- Einfache Applikationen sind leicht erstellbar, komplexe verteilte Systeme benötigen ein hohes Maß an Erfahrung, da ingenieurmäßige Vorgehensweisen kaum existieren.
- Es bleibt fraglich, ob JEE 6 Anwendungen tatsächlich zwischen verschiedenen Herstellern portierbar sind.

Bewertung

- Abwärtskompatibilität und häufige Versionswechsel einzelner APIs schaffen immer neue Probleme.
- Die Entscheidung für oder gegen die Verwendung von JEE wird oft ohne fachliche Begründung getroffen.
- Eine Konkurrenz von Microsoft wurde mit dem .NET Application Server für 2012 angekündigt.
 - Das bisher als Konkurrent propagierte .NET Framework ist lediglich eine Klassenbibliothek.

Übung: Einarbeitung in JEE

- Von vielen Herstellern findet man Software und Tutorials. Beispielhaft sei hier JBoss (<http://www.jboss.org>) genannt.
- Ihre Aufgabe:
 - Suchen Sie ein geeignetes Tutorial für JEE 6 / EJB 3.1.
 - Installieren Sie die dazu erforderliche Software.
 - Arbeiten Sie das Tutorial durch und implementieren Sie die zugehörigen Beispiele.

Selbstkontrolle

1. Was unterscheidet eine Klassenbibliothek von einem Rahmenwerk?
2. Welche Verbesserungen erwartet man von der Spezifikation EJB 3.1 gegenüber EJB 2.1?
3. Worin unterscheiden sich Stateful und Stateless Session Beans?
4. Warum war die Einführung von Singleton Session Beans sinnvoll?
5. Wie wird Hibernate mit Persistent Entities verknüpft?
6. Wie funktioniert nachrichtenbasierte Kommunikation?
7. Wie wird eine Transaktion in Java abgebrochen?
8. Wie werden konkurrierende Zugriffe auf EJB synchronisiert?

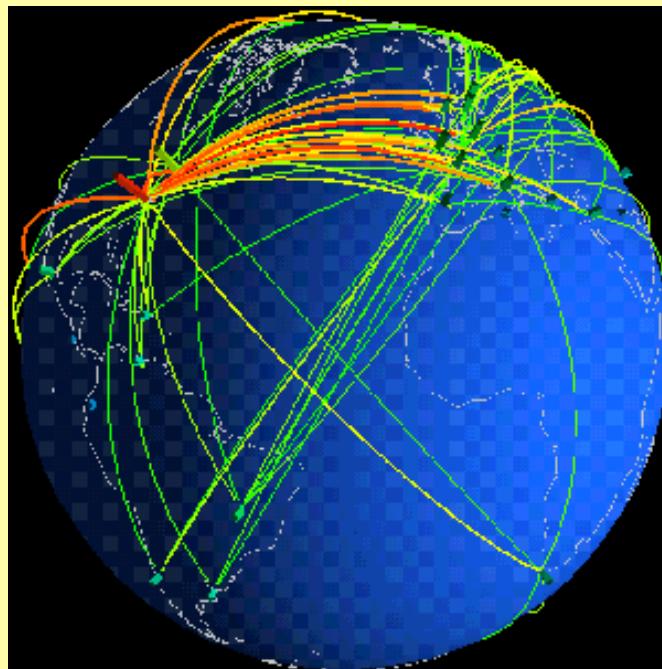
Kontakt

Duale Hochschule BW Ravensburg
Campus Friedrichshafen

Prof. Dr. Andreas Judt
Informationstechnik

judt@dhbw-ravensburg.de

Web Engineering Peer-to-Peer Networking



Prof. Dr. Andreas Judt

Überblick

- Entwicklung des Internets im Wandel der Zeit
- Client/Server Paradigma
- Peer-to-peer Netzwerke
 - Prinzip
 - Organisationsformen
- Suchen in P2P Netzwerken
- Effizienzbetrachtungen

Das Internet im Wandel (1)

- Früher ...
 - Datenkommunikation zwischen Forschungseinrichtungen
 - gemeinsame Ziele
 - Vertrauensbeziehungen zwischen Nutzern
 - technisch versierte Benutzer
 - konsistente und kohärente Architektur

Das Internet im Wandel (2)

- Heute ...
 - globale Infrastruktur der Informationsgesellschaft
 - neue Interessensgruppen und Kommerzialisierung (ISPs, Dienstanbieter)
 - Verlust der Vertrauensbeziehungen
 - Normalverbraucher, technisch unbedarf
 - zur Interessensdurchsetzung werden Techniken und Erweiterungen eingebbracht, die
 - im Hinblick auf die kurzfristige Bedarfsdeckung erfolgen
 - größtenteils ohne architekturelles Denken erfolgen
 - nicht im Einklang mit der Internet-Architektur stehen
 - die Kohärenz des Internets gefährden

Das Internet im Wandel (3)

- Probleme (1) ...

- Die heutige Internet-Architektur kann zahlreiche zukünftige Anforderungen unterschiedlicher Interessensgruppen nicht mehr erfüllen
- Nie erwartete Vielzahl angeschlossener Systeme
 - IP-Adressen werden Mangelware
 - neue Version von IP: IPv6

Das Internet im Wandel (4)

- Probleme (2) ...
 - Erweiterte Anforderungen an die Übertragungscharakteristika
 - Dienstgüte für Multimedia-Anwendungen
 - differenzierte Behandlung von Daten
 - Mobile Systeme
 - Erweiterungen von IP erforderlich: Mobile IP
 - Telekooperation: netzbasiertes Arbeiten/Lernen im Team
 - Erweiterungen für Gruppenkommunikation erforderlich
 - Einführung neuer Dienste im Internet
 - Steigende Zahl unterschiedlicher Dienste
 - flexible Diensteplattformen, Peer-to-Peer

Anforderungen an Techniken der Zukunft

- Unterstützung unterschiedlicher Dienste
 - Anwendungen haben unterschiedliche Bedürfnisse
 - z.B. Audio, Video
- Kommunikation in Gruppen
 - Internet als Plattform für Gruppenarbeit
- Flexible Bereitstellung von Diensten
 - Einfache Erweiterbarkeit des Dienstspektrums
 - Einfache Erreichbarkeit/Bereitstellung neuer Dienste

Anwendungen von Gruppenkommunikation

- Groupware, kollaborative Arbeit
- Videokonferenzen mit Whiteboard
- E-Learning
- Gezielte Informationsverteilung (Push-Techniken)
 - z.B. Wetter, Verkehr, etc.
- Auktionen
 - z.B. ebay
- Spiele
- IP-Telefonie

Client/Server Paradigma (1)

- Konzentration am Server
 - Unterlast in anderen Netzteilen
 - Asymmetrischer Verkehr
 - Erhöhung der Bandbreiten nutzt nur an wenigen Stellen im Netz
 - Problem der Skalierbarkeit (Speicherplatz, Rechenleistung)
- Information im Netz
 - World Wide Web, Request/Response-Protokoll: HTTP

Client/Server Paradigma (2)

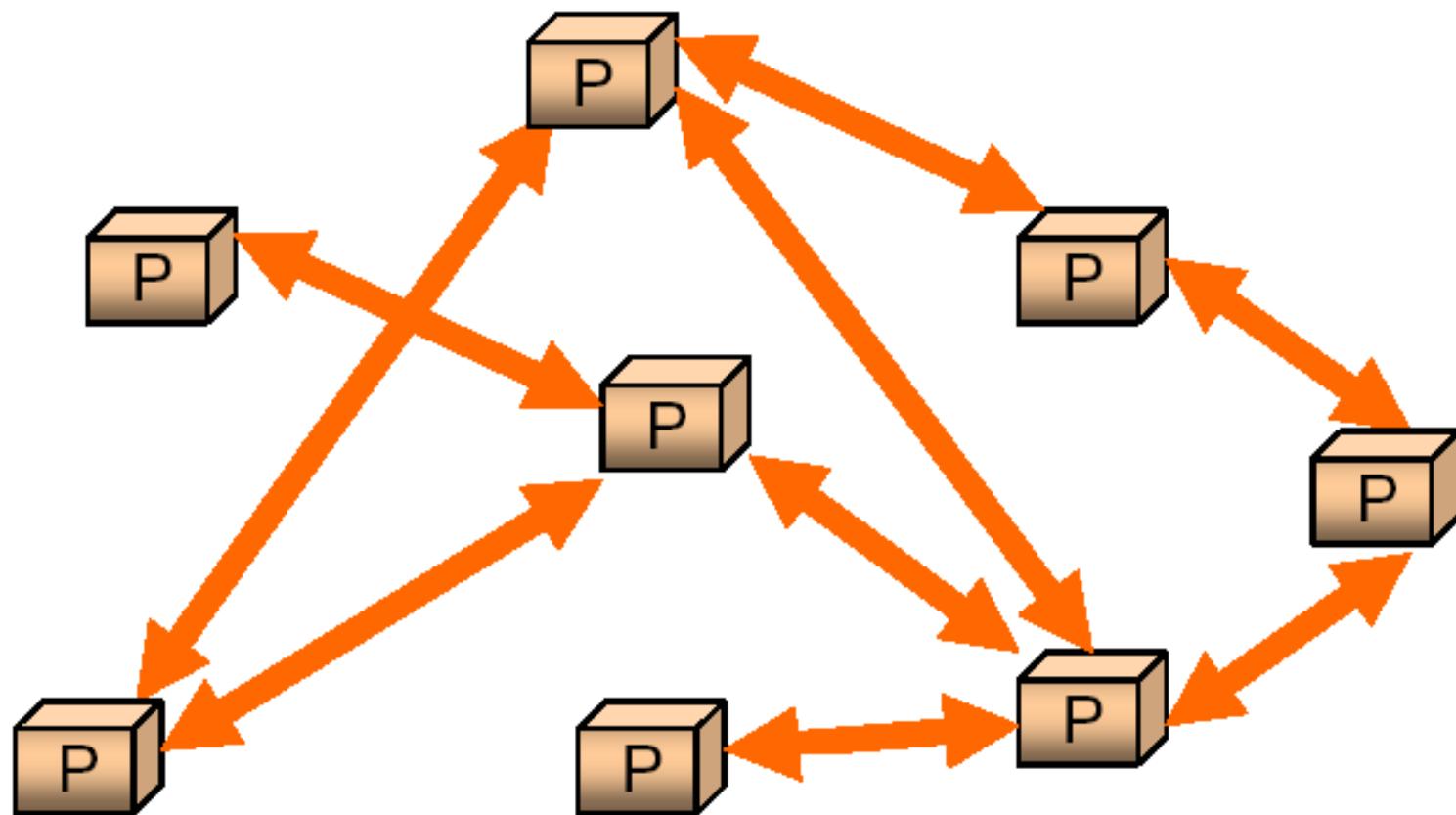
- Internet nicht komplett von Suchmaschinen erfasst:
 - Produktion von 2×10^{18} Byte pro Jahr
 - Veröffentlichung von 3×10^{12} Byte pro Jahr
 - Google indexiert nur 1.3×10^8 Webseiten
- Ungenutzte Ressourcen
 - CPU
 - Speicherplatz
 - Information
- Server stellt Single-Point-of-Failure dar!

Peer-to-Peer Netzwerke (1)

Definition von Peer-to-Peer-Systemen

- Peer-to-Peer-System ...
 - Verteiltes System bzw. Anwendung charakterisiert Interaktion von (gleichberechtigten) Endsystemen (Peers)
 - Gemeinsame Nutzung von Ressourcen in Endsystemen
 - Keine zentrale Kontrolle oder Nutzung zentraler Dienste
 - Gleichberechtigte und autonome Systeme
 - Selbstorganisation des Systems

Peer-to-peer Netzwerke (2)



Interaktion von Endsystemen (1)

- Diensterbringung durch direkte Kooperation der Peers
- Minimale Anforderungen an Infrastruktur (Netzwerk, Dienste)
- Annahme einer unzuverlässigen Infrastruktur (Ausfälle, Mobilität, etc.)
- Keine Nutzung zentraler Dienste (DNS, Datenhaltung, etc.)

Interaktion von Endsystemen (2)

- einfache Umsetzung
- keine ISP-Unterstützung notwendig
 - lediglich Software installieren
- keine aufwendigen, zentralen Server mit Skalierbarkeitsproblemen
- keine Standardisierung notwendig

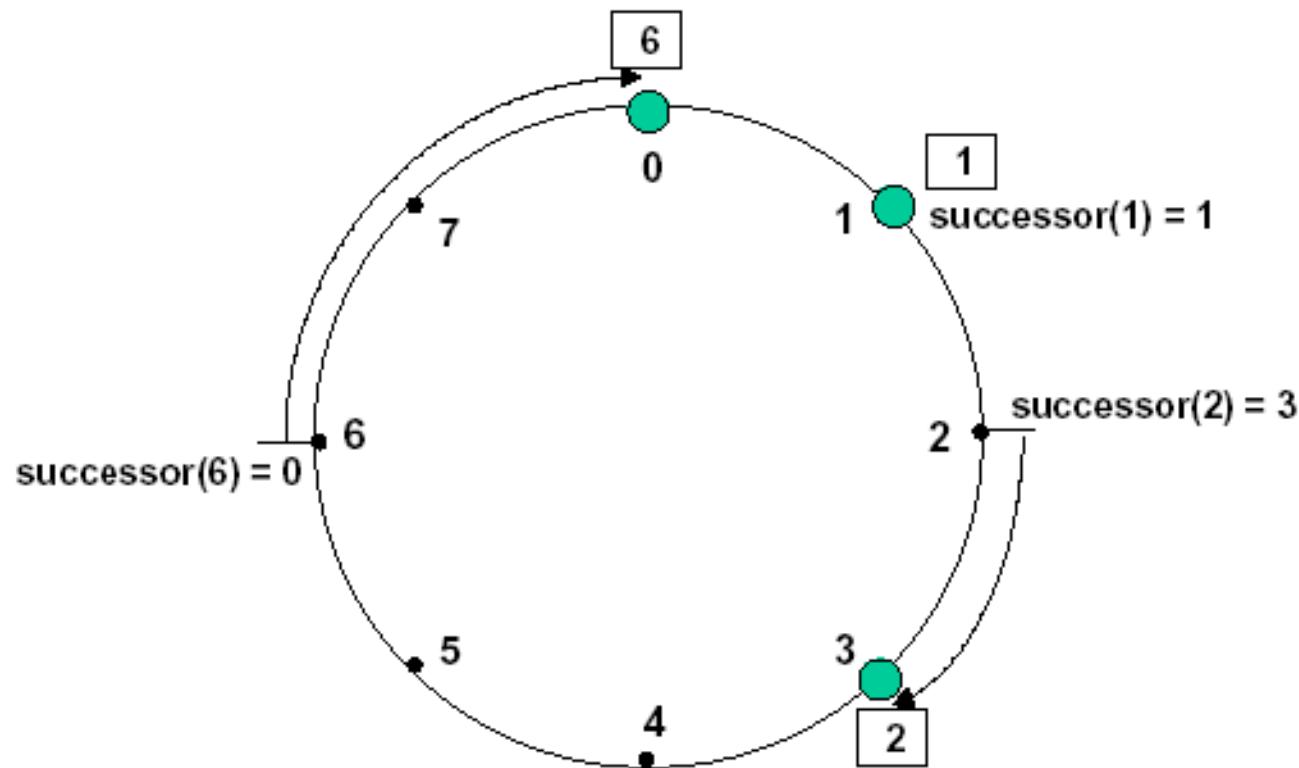
Organisationsformen

- Unterscheidung der Netzwerkstruktur
 - Strukturierte P2P
 - Unstrukturierte P2P
- Es gibt viele Organisationsformen für verschiedenste Anwendungen
 - Kollaboratives Spiegeln
 - Tauschbörsen
 - Topologisch organisierte Kommunikation
- Basis: Overlay Netzwerke
 - Physikalische Struktur unterscheidet sich von der logischen Struktur.

Strukturierte P2P

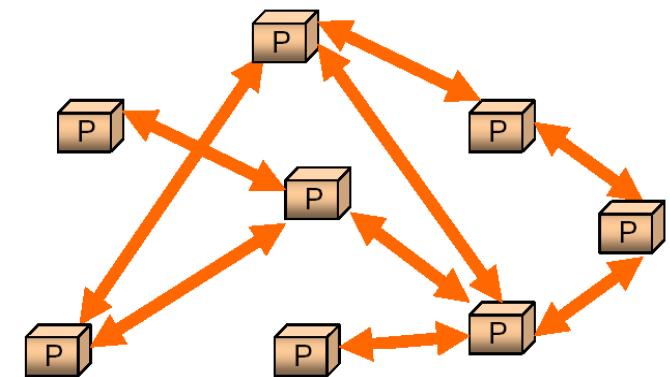
- Definierte Organisation von Rechnern
 - Ringbasiert oder topologisch
 - Globale Anwendung verwaltet und organisiert Daten
- Effiziente Suche: $O(\log N)$
- Speicherung von $O(\log N)$ Informationen pro Rechner
 - Reorganisation der Daten bei Verlassen bzw. Entern von Rechnern
- Probleme ...
 - Datenverlust durch plötzliche Ausfälle von Rechnern
 - Hoher Aufwand bei der Reorganisation von Daten
 - Ausfälle bzw. hinzugefügte Rechner werden nicht sofort erkannt

Ringbasiertes, strukturiertes P2P



Unstrukturierte P2P

- Hinzukommende Rechner erhalten wenige Verbindungen zu kontaktwilligen Rechnern des Netzes.
 - Keine Vorschrift zur Organisation der Struktur.
- Jeder Rechner kennt nur seine direkten Nachbarn.
 - Nachbarn organisieren die Suche bzw. den Download von Daten.
- Vollständige Suche: $O(N)$
 - Alternative: Verzeichnisdienst
 - Heterogenes Netzwerk
 - Wie werden Informationen aktualisiert?



Einige Zahlen

- Annahme: 100000 Rechner
- Annahme: Rechner bleibt statistisch 60 Minuten im Netzwerk
- Resultat: Pro Minute verlassen bzw. entern ca. 1600 Rechner das Netzwerk.

Vorteile von unstrukturierten P2P

- Meistens anonymes, zensurfreies Netzwerk
- Erhöhung der Downloadkapazität des Netzes mit jedem Download
 - Auch bereits bei laufenden Downloads
- Zeitliche Entkopplung des Downloads
 - Segmentierung von Daten
 - Verteilte Downloads

Probleme bei unstrukturierten P2P

- Häufige Knotenwechsel
 - Neue IP bei erneuter Einwahl ins Internet
- Finden von Daten
 - Suchen von Daten erfolgt mittels Durchfragen
 - Wen sollte man fragen?
 - Zentrale Verzeichnisse sind nie aktuell
 - Seltene Daten sind kaum auffindbar
- Heterogene Rechnerstrukturen
 - Rechenleistung
 - Festplattenkapazität
 - Netzwerkleistung

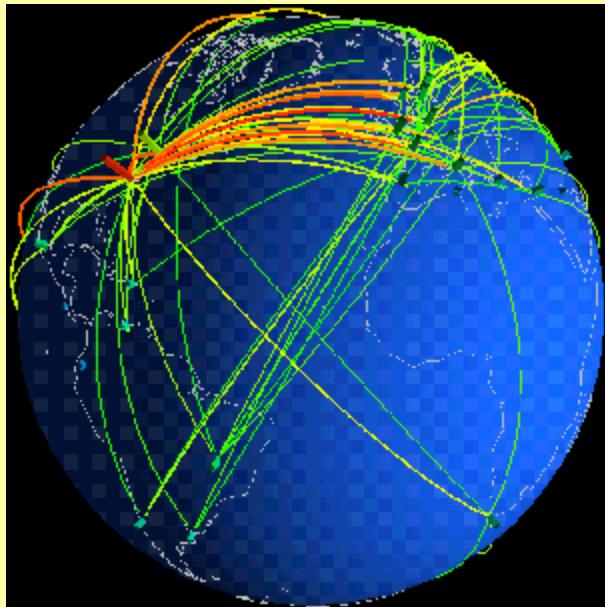
Kontakt

Duale Hochschule BW Ravensburg
Campus Friedrichshafen

Prof. Dr. Andreas Judt
Informationstechnik

judt@dhbw-ravensburg.de

Web Engineering Web Services



Prof. Dr. Andreas Judt

Was sind Web Services?

- Web Services:
 - Dienste die auf XML-Standards basieren.
 - Vereinfachen die Integration von Daten und Anwendungen.
 - orientieren sich am Anwender und nicht an einem bestimmten Gerät, einem Netzwerk oder einer Anwendung
 - Ermöglichen eine Kooperation von Benutzern, Gruppen und Organisationen
 - Steigern die Produktivität von Eigenentwicklungen
- XML Web Service ist ein Softwaredienst, der im Web über SOAP zur Verfügung gestellt, mit einer WSDL-Datei beschrieben und in der UDDI-Technologie registriert wird.

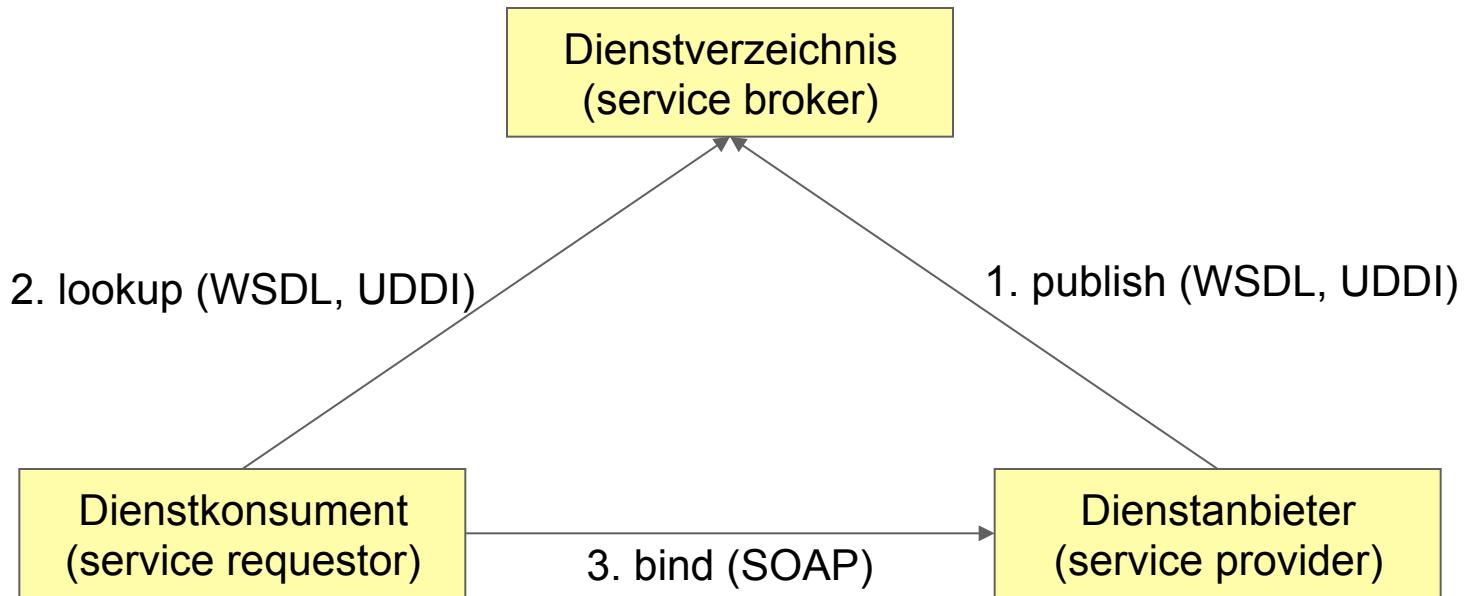
Anwendung von Web Services

- Integration von Anwendungen bzw. Anwendungskomponenten
- Ziele der Integration:
 - B2B-Anwendungen
 - EAI
 - Anwendungen, die keine lokale Installation benötigen, sondern über das Internet verfügbar gemacht werden können
- Vorteile des Komponentenmodells
 - Flexibilität → Kosten bei Softwareänderungen
 - Wiederverwendbarkeit → Kosten bei der Softwareentwicklung
 - Im Internet: Verteilung und Verfügbarkeit der Anwendungen und Daten
- Standardisierte plattformunabhängige Protokolle

Service-oriented Architecture SOA

- SOA konzentriert sich auf die Beschreibung und Verwaltung von Diensten die zur Laufzeit gesucht, dynamisch gebunden und genutzt werden.
- Service-Provider veröffentlicht die Beschreibung eines von ihm angebotenen Dienstes beim Service-Broker.
- Service-Broker (auch als „Registry“ genannt) ordnet den Dienst in Kategorien.
- Service-Requestor findet einen Dienst anhand von Suchmaschinen und der Kategorien und liefert Informationen über die Nutzung des Dienstes
- Service-Requestor kommuniziert mit Service-Provider, um den Dienst zu verwenden

Service-oriented Architecture SOA



Was ist SOAP?

- das Kommunikationsprotokoll für XML Web Services um Daten zwischen Systemen auszutauschen und entfernte Methodenaufrufe durchzuführen.
- Datenformat XML
- Kommunikation: HTTP und TCP (gewöhnlich)
- SOAP-Nachricht besteht aus
 - Header-Element
 - Metainformationen der Nachricht (wie Routing, Verschlüsselung usw.)
 - Body-Element
 - Nutzdaten, die vom Empfänger interpretiert werden

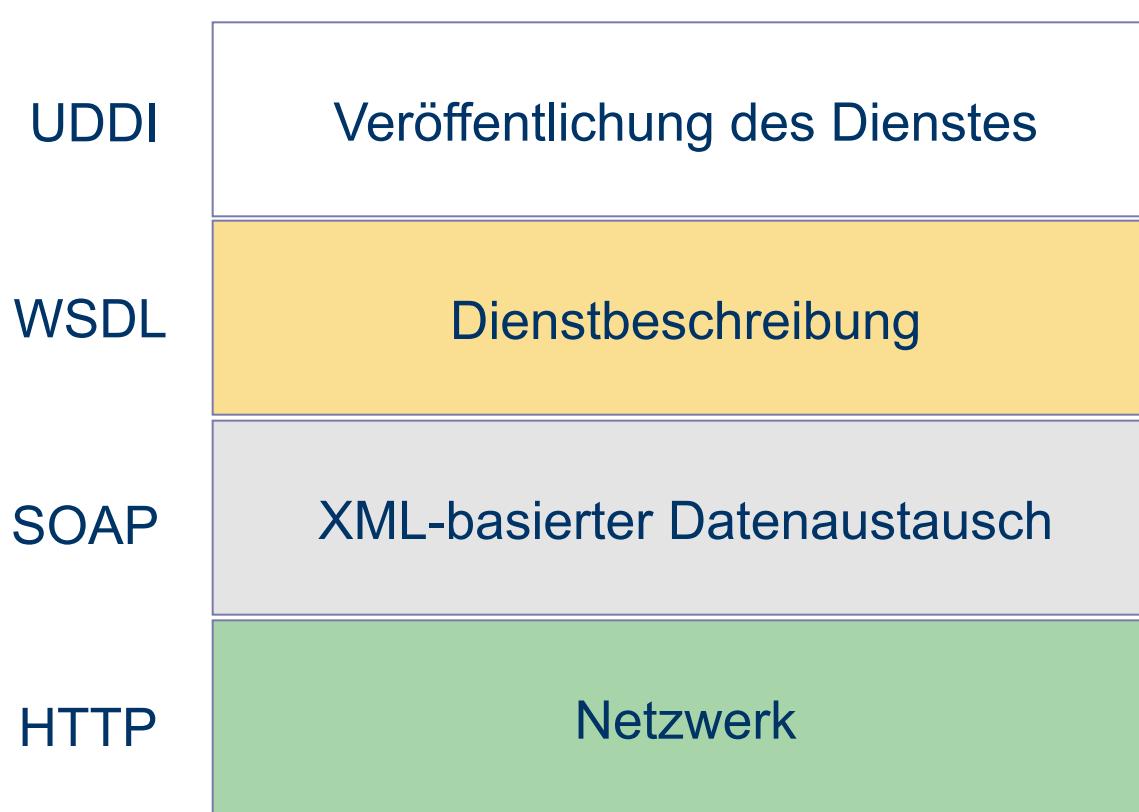
Was ist WSDL?

- WSDL - Web Services Description Language
- Eine Sprache, womit die Schnittstellen sowie Ein- und Ausgabeparametern beschrieben werden.
- Verwendung in Kombination mit SOAP und XML-Schema, um Web Services im Internet anzubieten

Was ist UDDI?

- UDDI (Universal Description, Discovery and Integration) bezeichnet einen Verzeichnisdienst. Er hat eine zentrale Rolle im Umfeld von dynamischen Web Services.
- UDDI besitzt eine SOAP-Schnittstelle
- Unterscheidung zwischen drei Arten von Informationen:
 - White Pages
 - Yellow Pages
 - Green Pages

Protokoll-Stack für Web-Services



Web Services als Grundlage für Microsoft .NET

- Entscheidende Herausforderung in der Geschäftswelt: optimale Verknüpfung von Computern, Endgeräten und Diensten über das Internet – zwei Aspekte:
 - Verknüpfung mit der Anwendungen mit Lieferanten und Kunden
 - Integrierbarkeit der verschiedenen Endgeräte in die eigene IT-Infrastruktur
- Web Services verbinden Daten, Geräte und Mitarbeiter auf einheitliche und personalisierte Weise.
- Web Services ermöglichen den Datenaustausch zwischen unabhängigen Quellen

Web Services: Datenaustausch zwischen unabhängigen Quellen

- Client-to-Client:
 - Intelligente Clients oder Geräte können Web Services bereitstellen und anwenden, die den Austausch von Daten überall und jederzeit ermöglichen.
- Client-to-Server:
 - Web Services können über das Internet Daten aus einer Serveranwendung, einem Desktop-Computer oder einem mobilen Endgerät, wie ein Tablet PC, PDA oder Mobiltelefon gemeinsam nutzen.
- Server-to-Server:
 - Web Services stellen in einer Umgebung von unabhängigen Servern eine gemeinsame Schnittstelle zwischen vorhandenen Anwendungen bereit.
- Service-to-Service:
 - Web Services können aufeinander folgend zusammenarbeiten, um komplexere Datenprozesse zu ermöglichen.

Web Services im Unternehmen

- Rasche Entwicklung und Verwaltung von Web Services als ein Schlüssel für geschäftlichen Erfolg
- Einsatzmöglichkeiten:
 - Integration von Unternehmensanwendungen
 - Verbesserung von Business-to-Business-Diensten
 - Verbesserung von Business-to-Customer-Diensten
 - Unterstützung von Mitarbeitern im Außendienst
 - Entwickeln von effizienteren verteilten Anwendungen sowie Peer-to-Peer-Anwendungen

Web Services im Unternehmen

- Anforderung:
 - Vielseitige und zuverlässige Plattform, die die erforderlichen Merkmale und Funktionen zum Entwickeln, Bereitstellen und Betreiben von Web Services und Unternehmensanwendungen bereitstellt.

Integration von Unternehmensanwendungen

- Integration von Unternehmensanwendungen mit geringen Kosten und beherrschbaren Risiken
- Erschließung von neuen und innovativen Einnahmequellen durch die Bereitstellung von Geschäftsprozessen über das Web
- Bereitstellung von Unternehmensdaten in neuen Anwendungen oder Analysetools

Business-to-Business-Dienste

- Berücksichtigung modularer Aspekte der Softwareanwendungen und -entwicklung
- Kommunikation über standardisierte Internetprotokolle SOAP (Simple Object Access Protocol) und UDDI (Universal Description Discovery and Integration)
- Integration von unternehmensinternen und -externen Applikationen und Daten → Zeit- und Kostensparnis

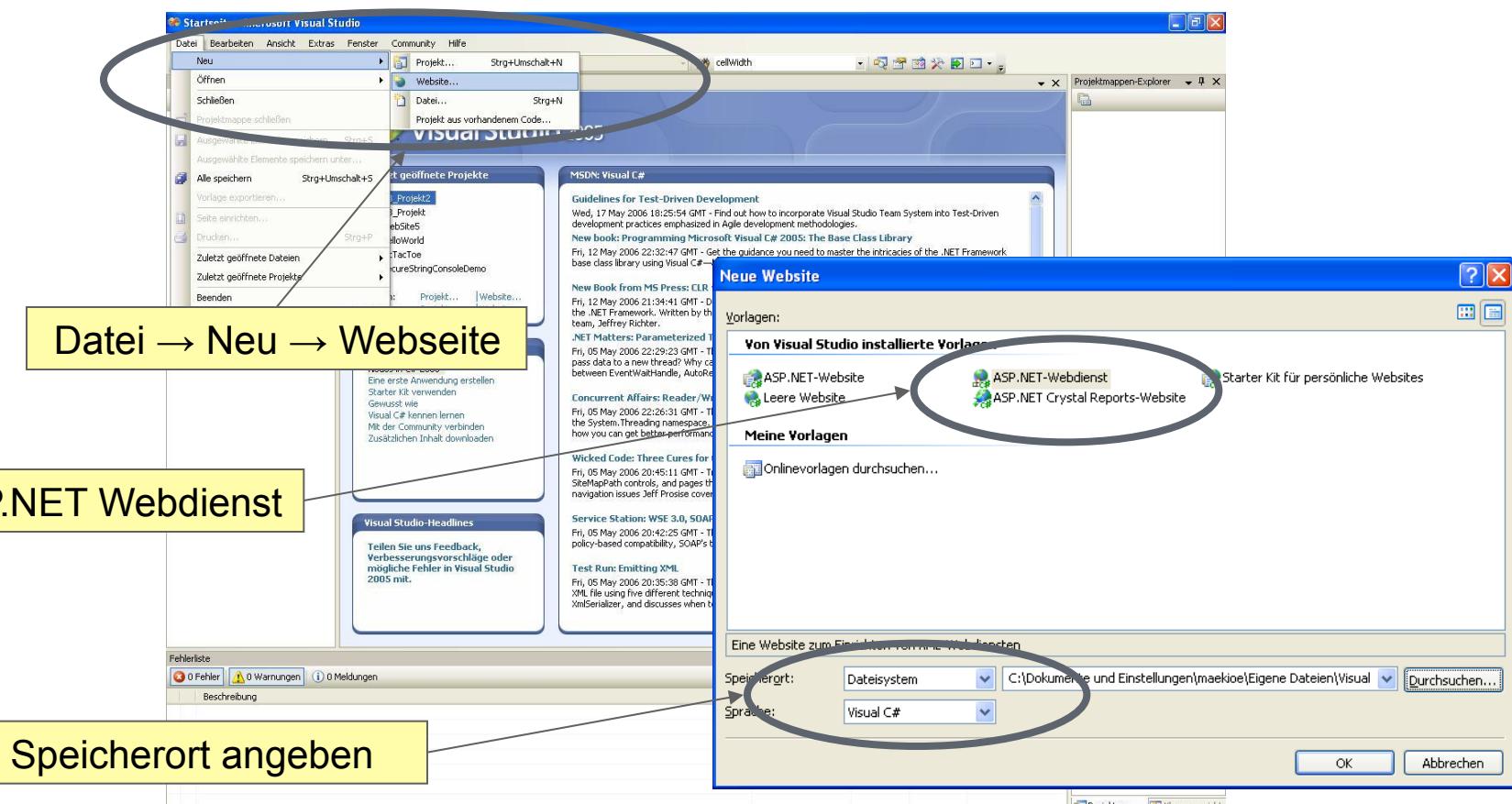
Business-to-Customer-Dienste

- Offenlegung von Betriebsprozessen über das Internet
- Erweiterung der Kommunikation mit Kunden
- Individuelle und personalisierte Anwendungen durch Kombination von Daten und Funktionen von mehreren Quellen
- Verkettung von mehreren Web Services

Unterstützung von Mitarbeitern im Außendienst

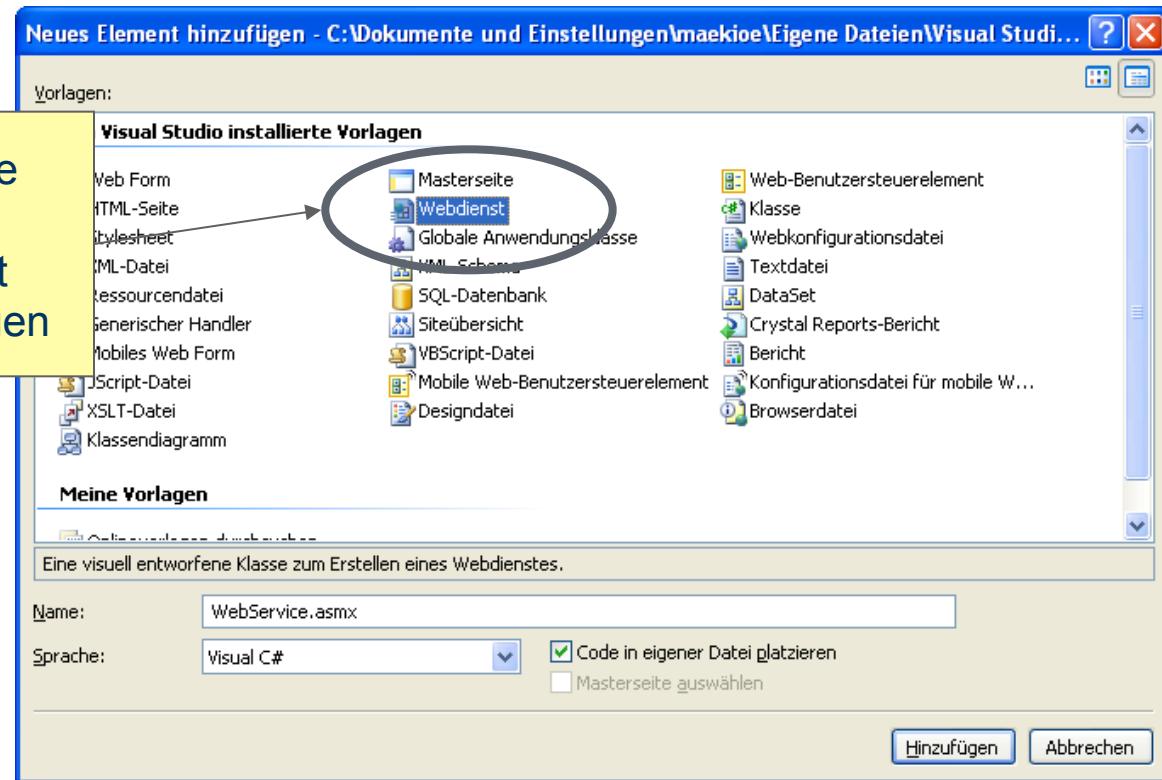
- Webanwendungen, die auf mobile Geräte (z.B.: Mobiltelefone, Pager und PDAs) abgestimmt sind
- Gerät optimierte intelligente Clientanwendungen
 - z.B. anziehbare Computer bei Servicetechnikern

Web Services erstellen mit Visual Studio



Web Services erstellen mit Visual Studio

Alternativ ASP.NET-Website
erstellen und über das
Website-Menü ein Element
vom Typ Webdienst hinzufügen



- Programmiersprachen:
 - C#
 - Visual Basic

Generierter Code eines Web Service

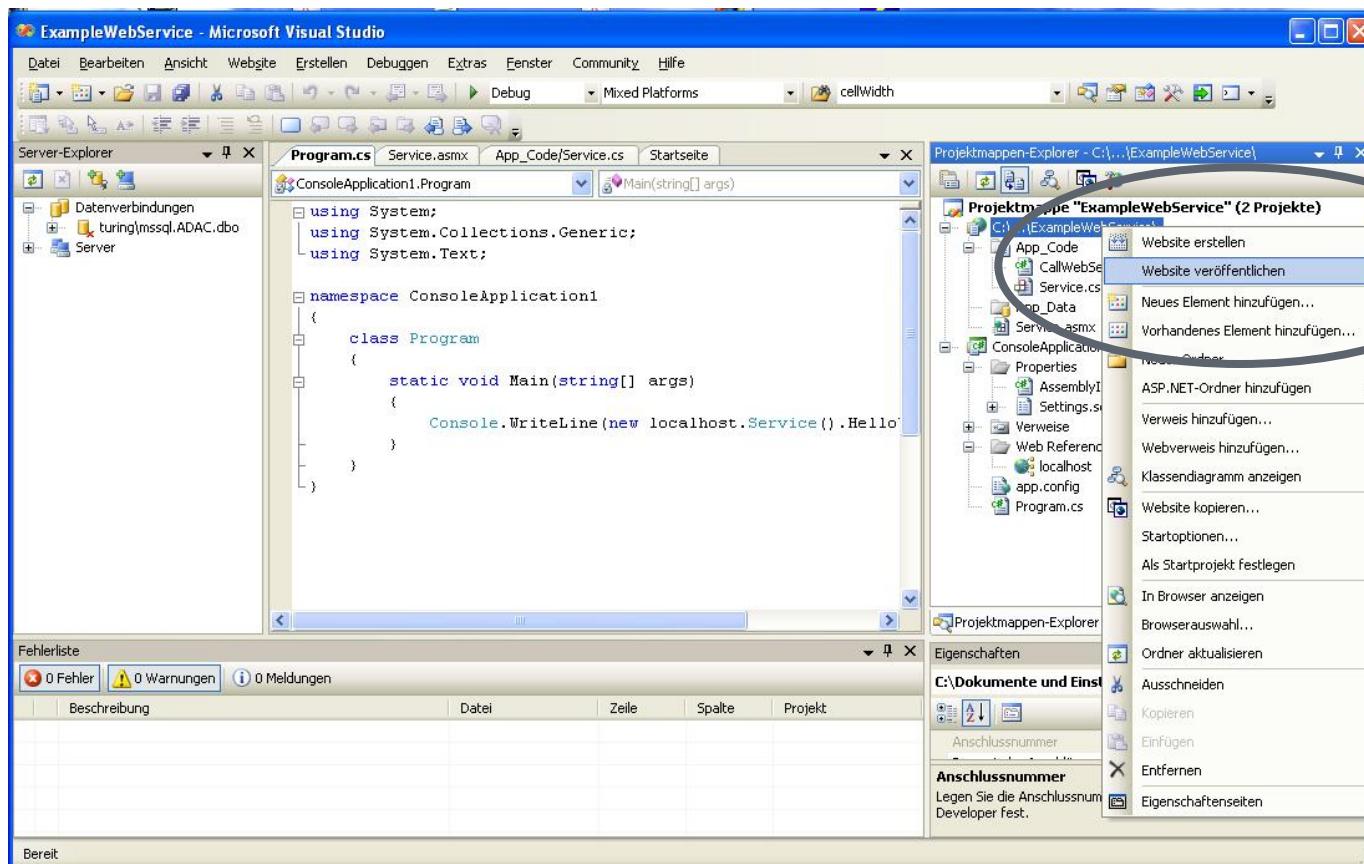
```
using System;
using System.Web;
using System.Web.Services;
using System.Web.Services.Protocols;

[WebService(Namespace = "http://tempuri.org/")]
[WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
public class Service : System.Web.Services.WebService
{
    public Service () {
        //Auskommentierung der folgenden Zeile bei
        //Verwendung von Designkomponenten aufheben
        //InitializeComponent();
    }
    [WebMethod]
    public string HelloWorld() {
        return "Hello World";
    }
}
```

Web Service veröffentlichen, Schritt 1

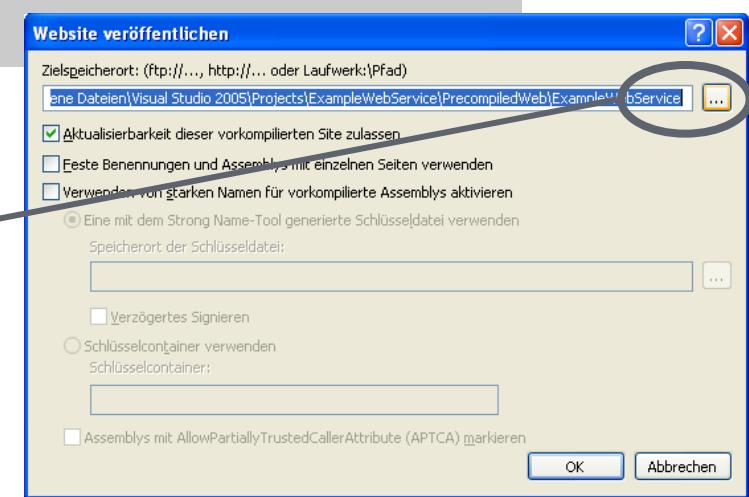
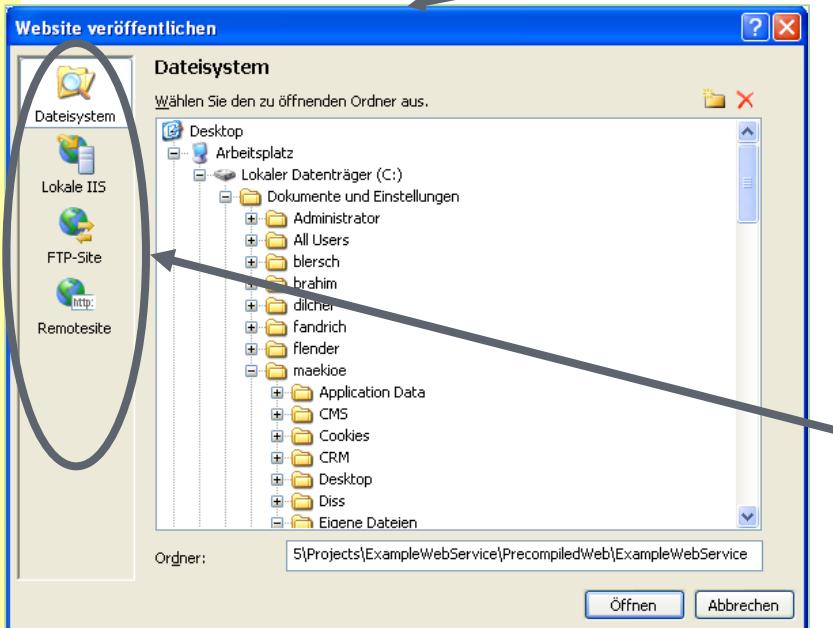
- In dem Web Service-Projekt mit dem rechten Maustaste klicken und „Webseite veröffentlichen auswählen“

Web Service veröffentlichen, Schritt 1



Web Service veröffentlichen, Schritt 2

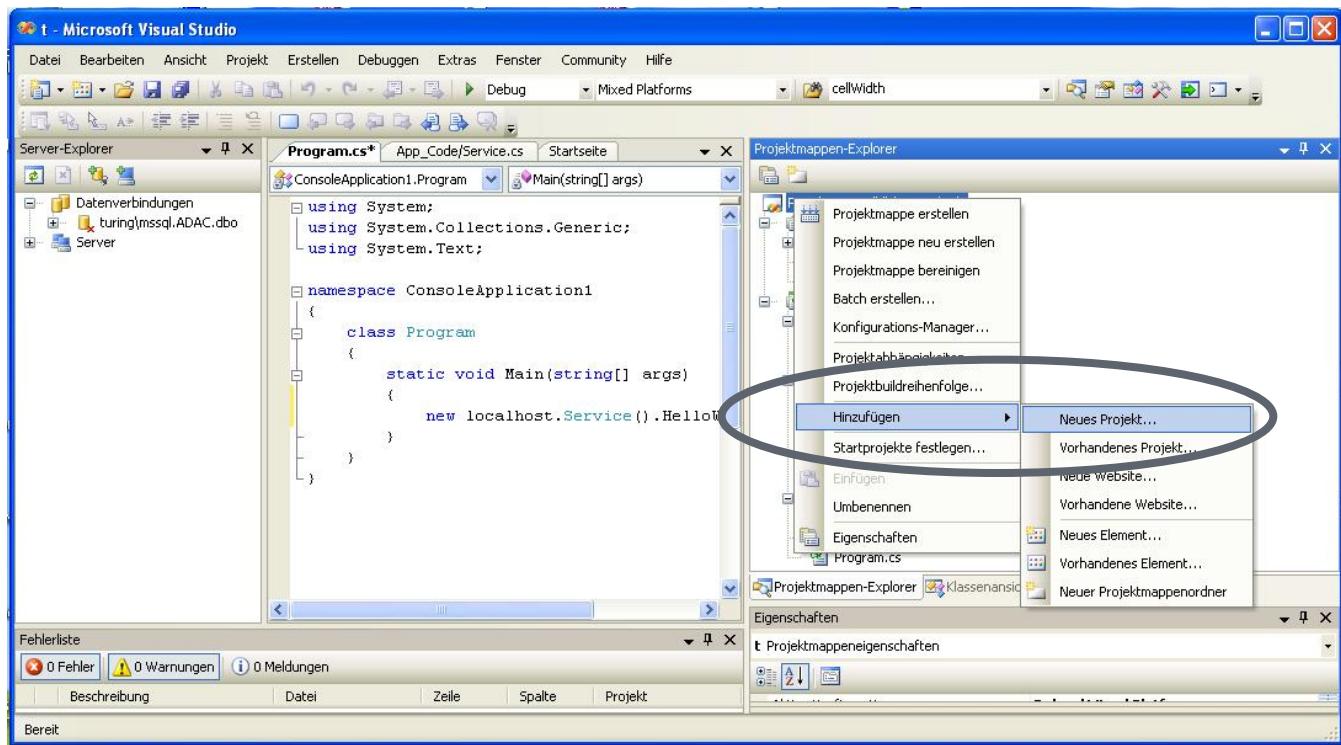
- Speicherort des Web Service angeben



- Wählen, ob ins Dateisystem, lokale IIS, FTP-Site oder Remotesite

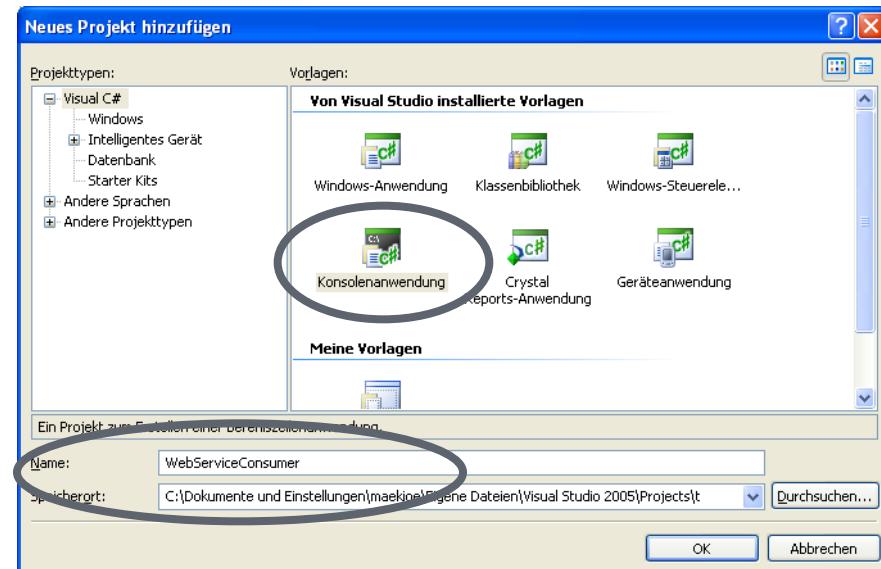
Web Service konsumieren – eine Konsoleanwendung, Schritt 1

- Neues Projekt in die Projektmappe einfügen



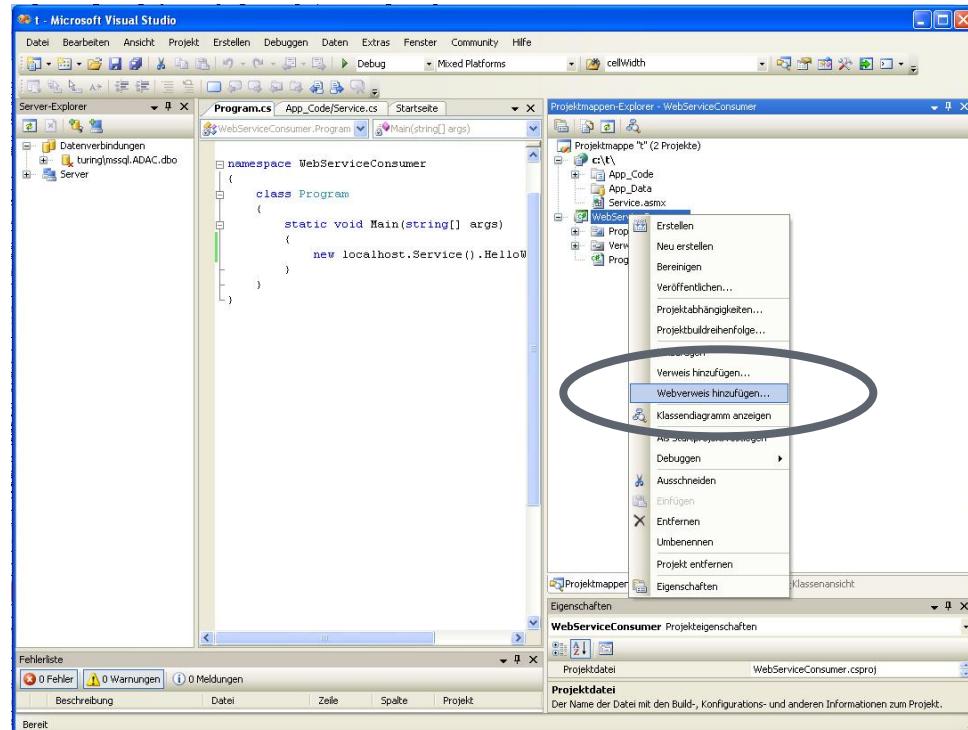
Web Service konsumieren – eine Konsoleanwendung, Schritt 2

- Typ der Anwendung selektieren: hier Konsolenanwendung
- Namen der Anwendung angeben



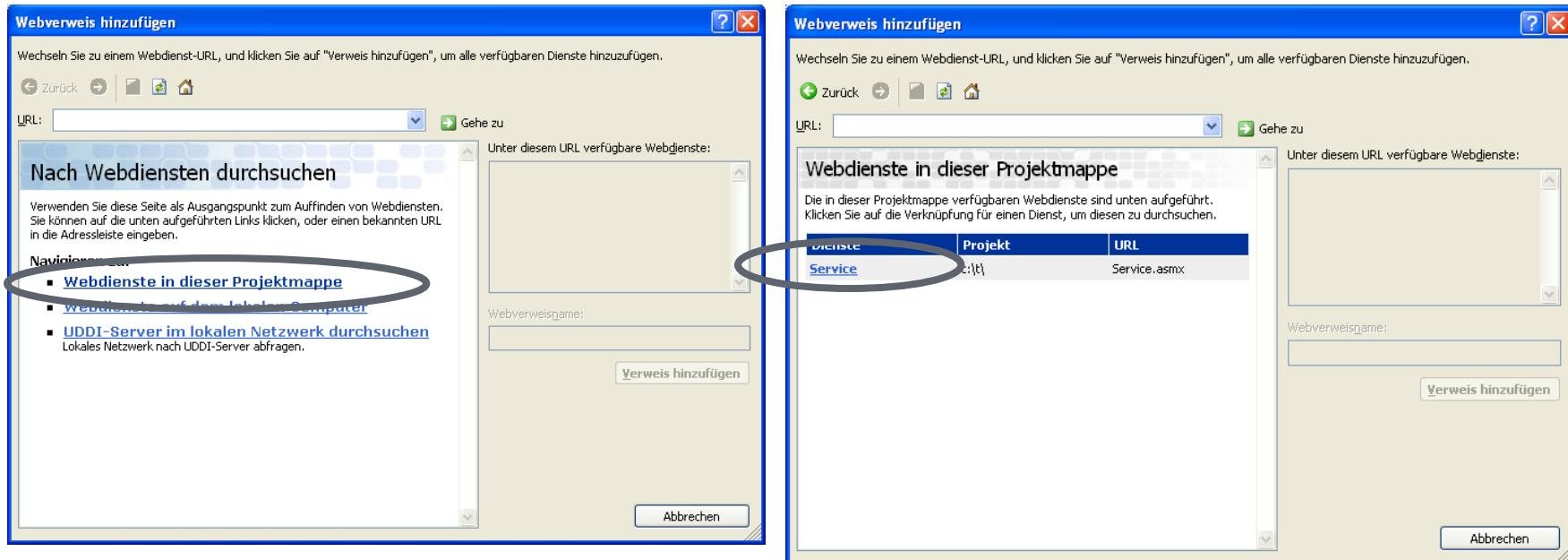
Web Service konsumieren – eine Konsoleanwendung, Schritt 3

- Webverweis in die Konsolenanwendung hinzufügen



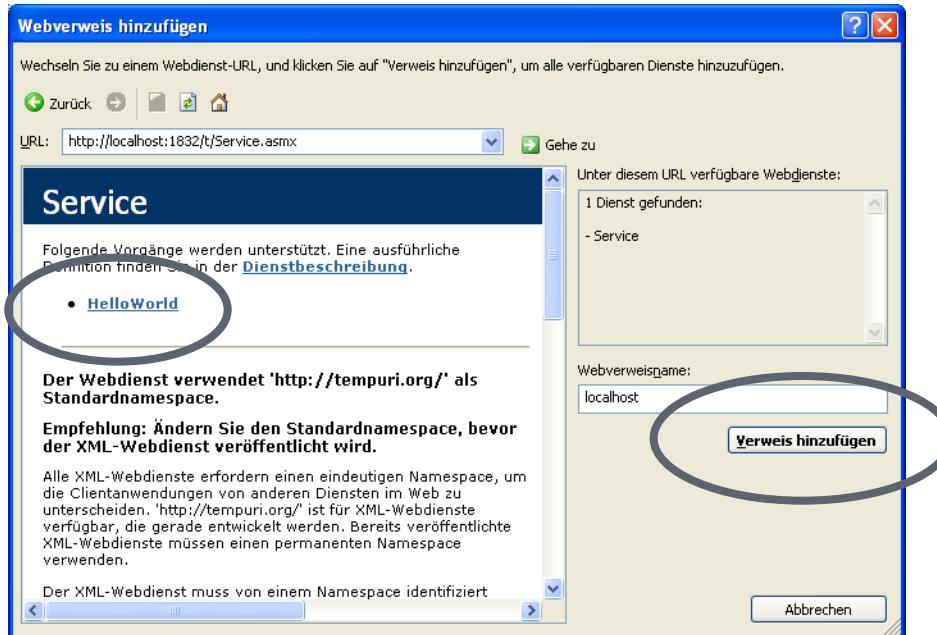
Web Service konsumieren – eine Konsoleanwendung, Schritt 4

- Webdienste in dieser Projektmappe selektieren
- einen Webdienst auswählen



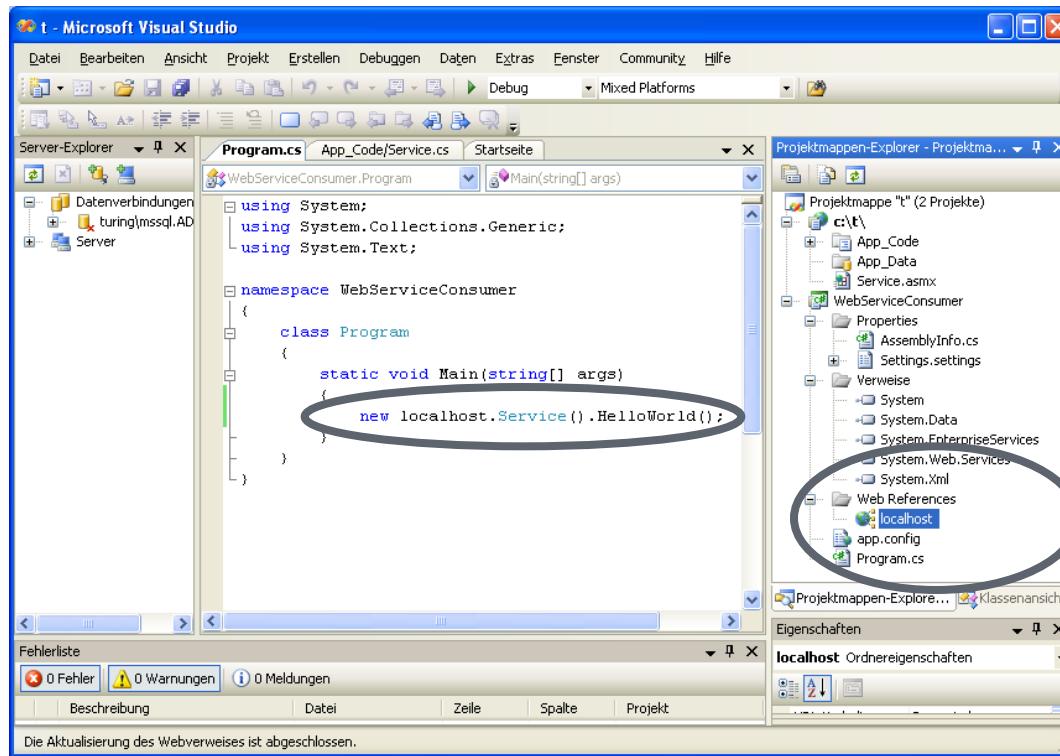
Web Service konsumieren – eine Konsoleanwendung, Schritt 5

- Die verfügbaren Web Services werden angezeigt.
- „Verweis hinzufügen“ klicken



Web Service konsumieren – eine Konsoleanwendung, Schritt 6

- Diese Dinge werden automatisch hinzugefügt...
- Der Aufruf des Web Services geht über den Webverweis...



Übungen

1. Erstellen sie eine Beschreibung der wichtigsten Funktionalitäten einer SOAP-Nachricht.
2. Beschreiben Sie wie eine SOAP-Nachricht mit Web Services zusammenhängt.
3. Beschreiben Sie die wichtigsten Elemente sowie deren Funktionen innerhalb eines WSDL-Dokuments.
4. Erarbeiten Sie eine Gegenüberstellung der drei Informationsarten eines UDDI. Beschreiben Sie die Rolle des UDDIs in Web Services.

Kontakt

Duale Hochschule BW Ravensburg
Campus Friedrichshafen

Prof. Dr. Andreas Judt
Informationstechnik

judt@dhbw-ravensburg.de