

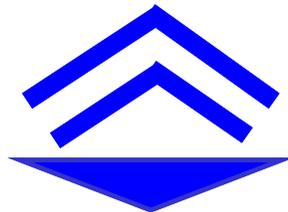
Scriptum zur Lehrveranstaltung

# Rechnernetze

(Architektur, Schichten, Protokolle, Internet und WWW,  
ausgewählte Netze und Dienste)

Auswahl

*- Internet und WWW -*



Studiengang Informatik, Kernfach Rechnernetze bzw. Verteilte Systeme (UL)  
Studiengang Praktische Informatik (BA)  
Umfang: 2 SWS  
15 Wochen

Prof. Dr.-Ing. habil. Klaus Irmischer  
Universität Leipzig  
Institut für Informatik  
Lehrstuhl Rechnernetze und Verteilte Systeme (em.)

Dresden, den 12. Dezember 2012

**Teil 1 (Architektur von Rechnernetzen): Kap. 1 ... 9**

**Teil IntW3 (Internet und WWW): Kap. 10 ... 11**

**Teil 2 (Ausgewählte Netze): Kap. 12 ... 18**

**Teil 3 (Übertragungssysteme): Kap. 19 ... 23**

**Gliederung**

1	Einführung.....	3
2	Netzwerkarchitekturen .....	3
3	Bitübertragungsschicht (Physical Layer) .....	3
4	Sicherungsschicht (Data Link Layer).....	3
5	Medienzugriffsverfahren (Media Access Control).....	3
6	Vermittlungsschicht .....	3
7	Transportschicht .....	3
8	Sicherheit in Rechnernetzen.....	3
9	Aspekte der Anwendungsschicht .....	3
10	Internet .....	4
10.1	Pilotprojekt ARPAnet .....	4
10.2	Entwicklung und Organisation des Internet.....	5
10.3	Architektur des Internet.....	7
10.3.1	Protokollstack des Internet.....	7
10.3.2	Struktur des Internet .....	8
10.3.3	Dienste im Internet.....	10
10.3.4	Internet-Schicht und Host-to-Host-Schicht.....	11
10.4	Verzeichnisdienste im Internet.....	13
10.4.1	Internet Domain Name System (DNS).....	13
10.4.2	X.500 Directory Service.....	13
10.4.3	Lightweight Directory Access Protocol (LDAP).....	14
10.5	Internet-Zugang.....	15
11	World Wide Web (WWW) .....	17
11.1	Merkmale, Dienste und Entwicklung des WWW .....	17
11.2	Hypertext-Konzept.....	21
11.3	Architektur des WWW.....	23
11.4	HTML und Gestaltung von WWW-Dokumenten.....	25
11.4.1	Strukturierung und Darstellung von Inhalten.....	25
11.4.2	HTML (HyperText Markup Language).....	26
11.4.3	Cascading Style Sheets (CSS).....	33
11.5	Interaktionen im WWW .....	34
11.5.1	Interaktive HTML-Seiten.....	34
11.5.2	Clientseitige Interaktionssteuerung (JavaScript, Applets) .....	35
11.5.3	Common Gateway Interface (CGI).....	37
11.5.4	Personal Hypertext Pre-Processor (PHP).....	38
11.5.5	Multimedia in HTML (Image Maps, Plug-ins).....	40
11.5.6	Verteilte Verarbeitung im WWW .....	40
11.6	Internet-Technologien (Web-Applikationen).....	41
11.6.1	Web - Basis moderner Internet-Technologien .....	41
11.6.2	Technologien Web-basierter Anwendungen (u.a. AJAX) .....	43
11.6.3	Cloud Computing .....	46
11.6.4	Future Internet Portal .....	48
11.7	Web 2.0 (mediales Internet).....	48
11.7.1	Neues Netzverständnis im Web .....	48

11.7.2	Social Networks (Websites, Online-Portale, Info-Dienste) .....	50
11.7.3	Gestaltung von Web 2.0-Anwendungen .....	60
11.8	Beispiel: Formulare und CGI .....	61
12	Abbildungsverzeichnis (Teil IntW3).....	64
13	Literatur.....	64

## **Teil 1: Architektur von Rechnernetzen**

### **1 Einführung**

### **2 Netzwerkkonstrukturen**

### **3 Bitübertragungsschicht (Physical Layer)**

### **4 Sicherungsschicht (Data Link Layer)**

### **5 Medienzugriffsverfahren (Media Access Control)**

### **6 Vermittlungsschicht**

### **7 Transportschicht**

### **8 Sicherheit in Rechnernetzen**

### **9 Aspekte der Anwendungsschicht**

## Teil IntW3: Internet und WWW

### 10 Internet

#### 10.1 Pilotprojekt ARPAnet

##### ARPA: Advanced Research Projects Agency

Initiierung/Koordinierung durch US-Verteidigungsministerium (DoD: Dept. of Defense).

##### Zielstellungen:

Netz (Wide Area Network) für militärischen Bereich, insbes. Luft- und Nuklearüberwachung. Pilotnetze an Univ. in Utah, Kalifornien, MIT (Boston). Informationsaustausch in Wissenschaft, Forschung, Bildung. Prototyp für weitere Rechnernetz-Entwicklungen.

##### Entwicklung:

1969: 1. experimentelles Netz (4 Knoten, Basis Honeywell 516), sog. ARPAnet

1972: DARPA (Defense ARPA)

Ausbreitung ARPA über gesamte USA (incl. Hawaii), Nordamerika.

Einbeziehung von Telefonnetzen, Satellitennetzen, Paketvermittlungsnetzen und Paketfunk.

1975: Herauslösen des militärischen Netzes --> **MILNET**: unterliegt strengen Sicherheitsvorkehrungen. Zivile Teile, Forschung, Lehre, Entwicklung verbleiben im ARPAnet.

1983: Protokollstack TCP/IP ersetzt das bisherige Transportprotokoll auf Basis des NCP (Network Control Protocol).

Es entsteht der Begriff des **Internet** (Zusammenfassung von Netzen, die auf dem Protokollstack TCP/IP aufbauen).

Ende 1989: NSF (National Science Foundation) übernimmt mit dem Backbone **NSFnet** die Aufgaben des ARPAnet. DoD löst das ARPAnet auf

Anm.:

DELTA := Rechnernetz der AdW/Berlin, TU Dresden, HU Berlin, Prag. Basis BESM-6, ES 1020, KRS 4200 (ca. 1980).

##### Architektur des ARPAnet

Kernstück: Paketvermittlungsnetz mit Vermittlungsknoten IMP bzw. TIP

\* IMP: Interface Message Processor (Paketvermittlungsknoten), später durch PSN (*Packet Switch Node*) ersetzt.

Transportprotokoll NCP (*Network Control Protocol*), später durch TCP/IP ersetzt.

\* TIP: Terminal Interface Processor (= Terminal-IMP). Erlaubt Direktanschluss von Terminals (analoge Architektur in DELTA).

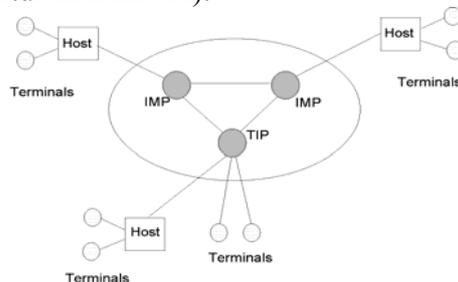


Abbildung 10.1: Topologische Struktur im ARPAnet

Kommunikationspfade: Telefonleitungen; später auch Koaxialkabel, Lichtwellenleiter, Satellitenübertragung. Datenübertragungsraten: 56 kbit/s ... 230,4 kByte/s (Leitungsverbindung). Erstmals gleichberechtigte Partner (peer-to-peer), im Gegensatz zu Master-Slave. Territorial unabhängige Nutzung (Resource Sharing). Dienste: FTP, Telnet, Email.

1969: 1. experimentelles Netz: 4 PV-Knoten, Honeywell 516; FTP, Telnet

### Protokoll-Architektur

Keine OSI-Architektur (Open System Interconnection). ARPA: erstes paketvermitteltes Netzwerk.

Bekannte Protokolle und Dienste:

Anwendungsschicht	Telnet, FTP, SMTP
Transportschicht	(Host-to-Host) NCP (Network Control Protocol)
Vermittlungsschicht	u.a. ARP (Address Resolution Protocol)
Lokales Netzwerk	Telefonnetz, Ethernet, ...

Basisdienste (seit 1969/71): Dienste und Protokolle der Anwendungsschicht:

- \* Telnet (Fernanschluss, entferntes Einloggen; 1969)
- \* FTP: File Transfer Protocol (Datei-Übertragung; 1969)
- \* SMTP: Simple Mail Transfer Protocol (1971)
  - Einfaches Post-Transferprotokoll
  - Aufbau von Message Handling Systems (E-Mail)
  - Postformat RFC 822 (POP): Adressierungsschema <nutzer>@<domain>
- \* Protokoll der Transportschicht: NCP (Network Control Protocol) := TCP + IP

## 10.2 Entwicklung und Organisation des Internet

### Einsatz TCP/IP-Protokolle

Übernahme der Basisdienste vom ARPAnet ins Internet (Telnet, FTP, SMTP; Basis: TCP/IP).

1978: Beschluss US-Regierung zum Einsatz TCP/IP.

1983: Ausschließlicher Einsatz von TCP/IP im ARPAnet ~> „Internet“.

*Einsatz von Internet*

1985: ca. 1961 Hosts angeschlossen, 1988: 80 000 Hosts

1994: ca. 2 Mio. Rechner in über 30 000 Netzen angeschlossen (DE ca. 100 000 Rechner).

1997: 16 146 000 Computer angeschlossen ~> 2002 ca. 1 Mrd. (insbes. Web-Server)

Zahl der Internet-Benutzer verdoppelt sich ca. alle 10 Monate (90er Jahre).

### Weiterentwicklung Internet in USA

ISO: Beginn Standardisierung (CCITT / ITU-T --> OSI), als ARPAnet bereits im Einsatz.

TCP/IP kein ISO-Standard, jedoch weltweite und dominierende Nutzung.

*Beschluß US-Regierung:* TCP/IP für alle staatlichen und öffentlichen Projekte vorgeschrieben; Förderung UNIX als offenes Betriebssystem (z.B. Berkeley-UNIX: enthält TCP/IP, FTP, Telnet, E-Mail). TCP/IP heute in allen UNIX-Systemen (incl. LINUX) und MS-Windows enthalten.

### NSFnet

NSF (National Science Foundation, USA, amerikanische Forschungsorganisation).



Abbildung 10.2: NSFnet (Auszug) und Transatlantic Links

1985: Gründung NSFnet: Backbone der US-Internets, trägt in 90er die Hauptlast im Internet. Zugang zum Internet, insbes. für amerikanische Universität. Integration weiterer Netze, u.a. CSnet, Usenet.

1989: NSF übernimmt Funktion des ARPAnet (ARPAnet wird vom DoD aufgelöst).

Weiterführung im ANSnet, NREN; kommerziell Sprint, MCI, WorldCom, ...

Backbone NSFnet / ANSnet:

- \* Verbindung aller Rechenzentren, Anschluss für Campusnetze & WAN
- \* Datenraten: 45 Mbit/s 1990, 90 Mbit/s 1996, 140 Mbit/s, 622 Mbit/s 1998, 2.5 Gbit/s 1999.
- \* Entwicklung zum Internet 2: vBNS (622 Mbit/s), Abilene/NGnet (2.5 Gbit/s 1999).

### **Verbund von Wissenschaftsnetzen in Europa**

1985/86: Gründung **RARE** (Réseaux Associés pour la Recherche Européenes)

- \* Europaweite Koordinierung der Kommunikationsaktivitäten
- \* COSINE-Projekt (Cooperation for an Open System Interconnection Networking in Europe, OSI-like)

Im Ergebnis von **COSINE**: Paneuropäische Backbones für Wissenschaftsnetze

- IXI (International X.25 Interconnect, X.25) --> Weiterführung durch EuropaNET
- EuropaNET (Multiprotokoll-Backbone, X.25/IP, Start Februar 1993) --> TEN-34
- TEN-34 (ATM/IP 1997) --> TEN-155 1998 --> GÉANT (SDH/WDM 2001, 10.5 Gbit/s)

Anschluss an europäische/amerikanische Internets und NorduNET.

### **Europäischer Internet-Backbone**

Beginn der 90er: Überschwappen Internet nach Europa, insbes. durch WWW.

Gründung **RIPE** (Réseaux IP Européenes), Koordination Internet-Verkehr in Europa.

**Ebone**: Europäischer Internet-Backbone.

Inbetriebnahme: 1992 (Datenraten: 34 Mbit/s), Gateway zu EuropaNET / TEN / GÉANT

### **Erweiterte Dienste im Internet**

*Informationsdienste*

- WWW-Browser: Mosaic, Netscape Communicator, Mozilla Firefox, Internet Explorer, ...  
Ab 2004: Web 2.0, mit neuen Diensten wie YouTube, Flickr, Twitter usw. -> Social Web
- E-Mail (POP, IMAP): elm, pine, Netscape Messenger, Mozilla Thunderbird, ...
- Suchmaschinen (z.B. Alta Vista, Dino, Google, Yahoo Search, Bing)
- Diskussionsdienst NetNews (aus Usenet)

*Telefonieren im Internet (Voice-over-IP)*: Verbesserte Sprachqualität, Konkurrenz zu Telefon

*MBone* (Multicast Backbone): mit Tools für Audio/Videokonferenz und Shared Applications (z.B. Whiteboard).

*MIME* (Multipurpose Internet Mail Extension) : Unterstützung für Nicht-ASCII-Texte, Multimedia (Audio/Video), Attachments

*Electronic Commerce*: Home-Banking, Tele-Shopping, e-cash

### **Weiterentwicklung des Internet (Provider: „keep simple“)**

*Neue Protokolle*

IPnG (IP next Generation, z.B. IPv6): erweiterte IP-Adresse und Standardoptionen:

Echtzeitkommunikation, höhere Übertragungsleistungen, vereinfachtes Protokoll.

Erweiterte Adressierungsbreite (128 bit), QoS, Sicherheit, mobile IP.

Einsatz u.a. im Core Network des UMTS, Phase 2 („all IP“).

mobile IP: drahtloser Zugang zum Internet (IPv6, auch bereits IPv4)

*Internet 2 (Gigabit-Netzwerk)*

WAN-Testnetz vBNS (622 Mbit/s, ATM; Vorreiter für Internet 2).

In USA gibt es mehrere 622 Mbit/s-Netze, u.a. vBNS, NASA-Netz. Oberhalb 622 Mbit/s sind es regionale Netze mit ATM bzw. WDM-Technik, u.a. Rainbow II (Los Alamos), MONET (New Jersey/Washington D.C.).

Einsatz Abilene / NGnet (1999): 622 Mbit/ bzw. 2.5 Gbit/s -> 10 Gbit/s (Basis: SDH/WDM).

### Organisation (Überblick)

USA: Internet Society (ISOC, 1992):

Koordination und Weiterentwicklung, keine zentrale Abstimmungen.

Technische Organisationen: IAB (Architecture Board; RFC's), IETF, IRTF

Europa: RIPE (Réseaux IP Européens): Koordination Internet-Verkehr in Europa.

Europäischer Internet-Backbone: Ebone (Inbetriebnahme 1992, 34 Mbit/s; Gateway zu EuropaNET / TEN-34/155 / GÉANT).

Deutschland: Nutzergruppen und Dienstanbieter (Service Provider):

Nutzergruppen:

Hochschulen/Forschungseinrichtungen, Wirtschaftsunternehmen, Privatpersonen.

Service Provider (Auswahl):

DFN e.V.: S-WiN (X.25), B-WiN (ATM), G-WiN (SDH/WDM), X-WiN (dark fiber)

X.25-Kom., X.400 Mail-Verbund, Internet, EARN/Bitnet

XLINK (--> NTG: Network und Telematic GmbH): Internet-Dienste, E-Mail, NetNews

EUnet Deutschland GmbH: E-Mail, UsenetNews, Internet

Individual Network (IN) e.V.: Privatpersonen; Partner von XLINK, EUnet, DFN

Verwaltung: DE-NIC (DE-Network Information Center): Karlsruhe.

Vergabe / Koordination Domain-Namen (Namens-Räume / Internet-Adresse).

Betreiben des Primary Name Server für Domain DE.

## 10.3 Architektur des Internet

### 10.3.1 Protokollstack des Internet

TCP/IP – Protokollfamilie (protocol suite, protocol stack): Sammlung von Protokollen mit i.d.R. jeweils einem Protokoll je Schicht.

OSI-Schicht				DoD-Schicht			
Anwendung	File Transfer	Elektronic Mail	Terminal Emulation	Usenet News	Domain Name Service	Trivial File Transfer	Prozess / Applikation
Darstellung	File Transfer Protocol	Simple Mail Transfer Protocol	Telnet Protocol	Network News Transfer Protocol	Domain Name System	Trivial File Transfer Protocol	
Sitzung	(FTP) RFC 959	(SMTP) RFC 821	(TELNET) RFC 854	(NNTP) RFC 977	(DNS) RFC 1034	(TFTP) RFC 1350	
Transport	Transmission Control Protocol (TCP) RFC 793				User Datagram Protocol (UDP) RFC 768		Host-to-Host
Netzwerk	Address Resolution Protocol (ARP) RFC 826		Internet Protocol (IP) RFC 791		Internet Control Message Protocol RFC 792		Internet
Sicherung	Ethernet, Token Ring, DQDB (802.x), FDDI, ATM, SDH/WDM, WLAN, GPRS SLIP, PPP (Wählverbindung)						lokales Netzwerk oder Netzzugriff
Bit-übertragung	Übertragungsmedium (wired, wireless) Doppelader, Koaxialkabel, Lichtwellenleiter, Funkwellen						

Abbildung 10.3: Protokollsuite des Internet

### Internet

Rechnernetz auf Basis eines TCP/IP-Protokollstacks. Hierarchisches Schichtenmodell (~> DoD- bzw. TCP/IP-Referenzmodell). 4 Schichten

- \* *Lokales Netzwerk* (unterste Schicht 1): Mediengriffverfahren (z.B. Ethernet), Prozeduren/protokolle (z.B. HDLC, PPP), keine Festlegungen.

- \* *Internet-Schicht* (Schicht 2): Vermittlung, Routing, IP-Protokoll (IPv4 bzw. IPv6).
- \* *Host-to-Host-Schicht* (Schicht 3): Transportschicht, Ende-zu-Ende-Verbindung.  
Bekannte Protokolle: TCP, UDP. Programmierschnittstelle: Sockets
- \* *Anwendungsschicht* (Schicht 4): Anwendungen und zugehörige anwendungsspezifische Protokolle (z.B. WWW mit HTTP-Protokoll).

### 10.3.2 Struktur des Internet

#### Topologie

Anfangs (um 1980) bestand Internet aus wenigen Campusnetzen, verbunden in einer flachen Topologie. Adressraum mit 32 Bit erschien ausreichend (IPv4). Adressbereich durch Router über Wegauswahltabellen problemlos verwaltbar. Explosionsartiges Wachsen des Internet (ab 1990)  $\leadsto$  Anwachsen der Tabellen, Austauschmenge von Routing-Informationen und Suchaufwand  $\leadsto$  schwierige Skalierbarkeit, insbes. für Multimillionen-Nutzersysteme.

1982: Einführung sog. **Autonomer Systeme (AS)**. Heutiges Internet (ab 1990) in AS aufgeteilt  $\leadsto$  sichert administrative Unabhängigkeit der Netzbetreiber. Autonome Systeme bieten Möglichkeit zur hierarchischen Aggregation von Routing-Informationen  $\leadsto$  verbesserte Skalierbarkeit.

1995: Neues Protokoll (IPv6). Erweiterte Adressbreite (128 Bit) und Berücksichtigung von Dienstgüte (flow label), Sicherheit (AAA) und mobile IP im Protokoll, gleitende Einführung.

#### Routing

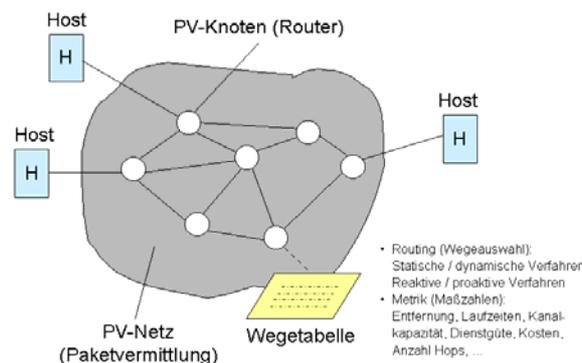


Abbildung 10.4: Paketvermittlungsnetz (Routing)

Routing: Wegauswahl zwischen Netzknoten eines vermaschten Paketvermittlungsnetzes (Algorithmen und Protokolle, Algorithmus nach Dijkstra, modifiziert).

Im Internet: dynamisches (*adaptives*) Routing, d.h. aktualisierte Weginformationen:

- *Distance Vector Protocol*: älteres Verfahren, Zustandsinformationen zu benachbarten Knoten (Basis: Heartbeat-Algorithmus, Neighborhood-Routing),
- *Link State Protocol*: neueres Protokoll, Zustandsinformationen des gesamten Netzes, schnellere Konvergenz (Basis: Probe-Echo-Algorithmus, Flooding).

Bei Einteilung in Autonome Systeme (AS) wird Wegeauswahl aufgeteilt in Teilbereiche des Inter-Domain-Routing zwischen AS und Intra-Domain-Routing innerhalb eines AS.

#### Autonome Systeme (AS)

Nach außen stellt ein Autonomes System eine homogene Einheit dar, kann als ein Knoten mit vollständiger Konnektivität zu seinen benachbarten AS angesehen werden  $\leadsto$  damit Komplexität reduziert.

Wegeauswahl (Routing) aufgeteilt in Inter-Domain-Routing zwischen AS und Intra-Domain-Routing innerhalb eines AS. Aufteilung erlaubt auch die Umsetzung individueller Routing-

Strategien innerhalb einer Routing-Domain. Verbindung zwischen den AS erfolgt über sog. Grenzrouter. Diese verknüpfen Intra-Domain- und Inter-Domain-Routing, so dass

- intern erreichbare Netze Informationen über andere autonome Systeme erhalten und
- umgekehrt Informationen dieser Netze auch anderen AS bekannt werden.

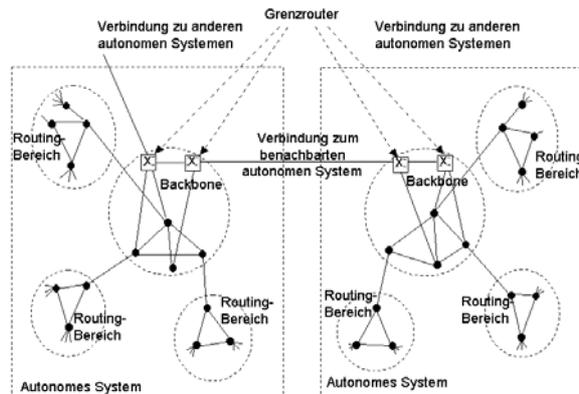


Abbildung 10.5: Autonome Systeme und Grenzrouter (Beispiel)

Autonome Systeme werden durch eine eindeutige Nummer (16 Bit) gekennzeichnet. Die Verwaltung und Zuordnung obliegt der Internet Assigned Number Authority (IANA) bzw. einer nachfolgenden autorisierten regionalen Registratur.

2002: 10 698 autonome Systeme registriert, davon 8 852 als sog. Stub-AS klassifiziert.

Stub-AS: sind über ein anderes AS mit dem Internet verbunden, bieten anderen AS keine Transportdienste an. Häufig beinhaltet ein Stub-AS mehrere Intranets, die über das sog. Backbone-Netz des AS untereinander sowie mit benachbarten AS verbunden sind. Beispiel eines Stub-AS: G-WiN / X-WiN des DFN.

Die verbleibenden 1 846 autonomen System wurden als Transit-AS klassifiziert. Transit-AS besitzen Verbindungen zu anderen AS, können Transit- und auch lokalen Verkehr führen.

### Domain-Routing

Routing in autonomen Systemen (Algorithmen, Protokolle) wird aufgeteilt in Teilbereiche:

- zwischen den AS: Inter-Domain-Routing (Protokolle: EGP, BGP),
- innerhalb eines AS: Intra-Domain-Routing (Protokolle: RIP, OSPF).

**Inter-Domain-Routing:** Protokolle dienen zum Austausch von Erreichbarkeits- und Weginformationen zwischen autonomen Systemen:

*EGP (Exterior Gateway Protocol):* älteres Protokoll (Vorläufersystem).

- *Baumtopologie* mit zentralem Backbone und Eltern-Kind-Beziehung zwischen den AS.
- Eingeschränkte Leistungsfähigkeit, u.a. keine Multihome-AS möglich.

*BGP (Border Gateway Protocol):* moderneres, flexibleres Protokoll, z.Zt. Version 4 (2005).

- Über BGP werden *Pfadvektoren* an benachbarte AS verteilt.
- Pfadvektor enthält eine Sequenz von Nr. autonomer Systeme, über die ein bestimmtes Netz erreicht werden kann (Pfadvektor nur Erreichbarkeitsinformation, keine Information zur Metrik der Teilpfade). Damit auch Erklärung zur Bereitschaft, Verkehr des benachbarten AS zu diesem Ziel durch die eigene Routing-Domain zu transportieren.
- Flutungsmodelle zur Verbreitung der Pfadinformationen für BGP daher nicht geeignet. Dazu werden durch das *TCP (Transmission Control Protocol)* gesicherte Verbindungen zwischen den Grenzroutern genutzt.
- BGP bietet nicht zwingend die kürzesten Pfade. Jedes AS nutzt individuell festgesetzte Regeln für die beste Route und entscheidet, ob diese den benachbarten AS mitgeteilt werden (kommerzielles Problem, ISP: Internet Service Provider).
- Beziehungen zwischen Paaren autonomer Systeme:

*Kunden-Anbieter-Beziehungen* (ISP offeriert Kunden eine Verbindung zum Internet),  
*Peer-Peer-Beziehungen* (stellen Verbindungen zwischen Kunden bereit).

AS zur Administrierung auch in Routing-Areas unterteilbar; diese durch gemeinsames Backbone-Netz verbunden. Routing-Bereiche eines Stub-AS (sog. "Intranets", häufig mit statischem Routing) werden i.allg. von einem Betreiber administriert.

**Intra-Domain-Routing:** Für das Verteilen von Wegeinformationen und Aktualisierung innerhalb von AS dient das Interior Gateway Protocol (IGP). IGP reagieren schnell auf Veränderung der Topologie eines AS, mit schneller Konvergenz ohne Schleifenbildung.

Bekannte IGP:

- *RIP (Routing Information Protocol):* älteres Protokoll. Ausgangspunkt in ARPA.  
 Basis: Distance Vector Protocol, Kenntnis der Metrik zu den benachbarten Knoten.
- *OSPF (Open Shortest Path First):* leistungsfähigeres Protokoll, von IETF empfohlen.  
 Basis: Link State Protocol, ständige Überwachung der Verfügbarkeit und Qualität der Verbindungen, schnellere Konvergenz. Verteilung der Informationen mittels Flutungsverfahren. Metriken zur Bewertung der Pfade, u.a. Übertragungsverzögerung, Fehlerraten, Unterstützung mehrerer Pfade zu einem Ziel, externe Routen udgl.

### 10.3.3 Dienste im Internet

#### Dienste und Protokolle

*Basisdienste:*

- Telnet     Entfernter Aufruf
- FTP        File Transfer Protocol
- SMTP       Simple Mail Transfer Protocol: RFC 822, <nutzer>@<computer>.<domain>
- NetNews    Nichtaktives Konferenzsystem (aus Usenet)
- DNS        Domain Name System (logische <-> physische Adresse)

*Informationsdienste auf Basis TCP/IP:*

Alex/NFS, Finger, Netfind, Wais, Whois, Gopher, Archie, ..., WWW

WWW: World Wide Web (Basis: Client/Server-Architektur; HTML, HTTP, CGI, ...)

- HTTP     HyperText Transfer Protocol: Protokoll einer Client/Server-Architektur
- HTML    HyperText Markup Language: Dokumenten-Format (.html); HT-Editoren
- URL     Uniform Resource Locator: Lokalisierung der HTML-Dokumente und Links

Client/Server: Clients (Browser): Anforderung/Darstellung der Dokumente; Links.

u.a. Mosaic, Netscape, Mozilla Thunderbird, Internet Explorer (MS), Hotjava (Sun).

Server: Speicherung/Bereitstellung der Dokumente (Hypertext-Format; Dateityp .html)

Peer-to-Peer (P2P): File-Sharing (KaZa), Tauschbörsen (Gnutella, Napster).

Prozess/ Applikation	Alex/ NFS	Finger	Gopher	HyTelnet	Netfind	WAIS	Whois	WWW	Archie	Prospero
	XDR RFC 1014	Finger User Information Protocol	Internet Gopher Protocol	Telnet Protocol	Finger DNS SMTP	Z39.50	Nickname Whois	HyperText Transfer Protocol		Prospero Protocol
	RPC RFC 1057	RFC 1288	RFC 1436	RFC 854			RFC 954			
Host-to-Host	Transmission Control Protocol (TCP)								User Datagram Protocol (UDP)	
Internet	Address Resolution Protocol (ARP)		Internet Protocol (IP)				Internet Control Message Protocol (ICMP)			
lokales Netzwerk oder Netzzugriff	Ethernet, Token Ring, DQDB (802.X), FDDI, ATM, SDH/WDM,...									
	Übertragungsmedium (wired, wireless) Doppelader, Koaxkabel, Lichtwellenleiter (Glasfaser), Funkwellen									

Abbildung 10.6: Informationsdienste im Internet (Protokolle)

### **Electronic Mail (E-Mail):**

Asynchrone Nachrichtenkommunikation über Netzwerk, Briefkastenprinzip (Speicherung in Mailbox, Abruf, Folder).

Rundsendungen, forward-Verkettungen; Textinformationen und Multimedia-Mails.

SMTP: Simple Mail Transfer Protocol

- \* Internet-Protokoll für E-Mail (aus ARPAnet, 1971) ~> E-Mail-Systeme, u.a. elm, pine
- \* Adressierung: <user>@<rechner><domain>, z.B. meier@isun.informatik.uni-leipzig.de

MIME: Multipurpose Internet Mail Extensions

- \* Protokoll kompatibel zu RFC 822, Unterstützung Nicht-ASCII-Texte (z.B. Umlaute), Multimedia-Dokumente (Text, Grafik, Audio, Videosequenzen), Attachments

X.400: MHS (Message Handling System) der CCITT

- \* Standard der ISO/OSI: CCITT X.400 bzw. ISO MOTIS (Message Oriented Text Interchange), beide weitgehend angeglichen; MHS-Produkt: OSITEL
- \* Adressierung: z.B.

C = de; A = d400; P = uni-leipzig; O = informatik; OU = lehrstuhl; S = meier

mit C: Country Code, A: Administration Domain (öffentlicher Bereich),

P: Private Domain (privater Versorgungsbereich), O: Organization,

OU: Organization Unit, I: Initials (Initiale), G: Given Name (Vorname),

S: Surname (Personenname)

### **MBone (Multicast Backbone)**

- Internet-Backbone für Audio/Video-Übertragung; Start 1992 (Konferenz San Diego)
- Overlay-Netzwerk, Protokollbasis: IPv4
- Multicast-Router (Multicast-Adressierung, Gruppenkommunikation), Tunneling der Pakete über Nicht-Multicastfähige Router
- Tools für Audio/Videokonferenzen, Shared Applications, u.a. vic, vat, nv, sd, wb
- MBone-DE (Basis IPv4): verwaltet vom DFN. Multicast-Backbone in DE.
- 6Bone (Basis IPv6): verwaltet von Uni Münster
- QBone: echtzeitfähige Dienste (USA)

## **10.3.4 Internet-Schicht und Host-to-Host-Schicht**

### **Internet Protocol (IP)**

Das Internet Protocol (IP) gehört zur Internet-Schicht (OSI-Schicht 3: Netzwerkschicht). Zusammen mit TCP (OSI-Schicht 4) bildet es das zentrale Protokollpaar des Internet. Insgesamt gehören zur Internet-Schicht (Version IPv4):

- Internet Protocol (IP): RFC 791
- Internet Control Message Protocol (ICMP): RFC 792
- Address Resolution Protocol (ARP): RFC 826

Hauptaufgaben des IP:

- \* Adressieren von Rechnern (RFC 826)
- \* Routing (Leitweglenkung)
- \* Fragmentieren von Paketen der darüberliegenden Schicht

IP bietet einen verbindungslosen (unzuverlässigen) Dienst an. Falls zuverlässiger Dienst erforderlich (z.B. bei Dateiübertragung), ist dies durch übergeordnetes Protokoll (z.B. TCP) zu erfüllen.

### **Internet-Nummernsystem (Internet-Adresse)**

Jeder Rechner im Internet benötigt eine eindeutige Adresse: Internet-Adresse (IP-Adresse)

- 32-bit langes Wort (4 Octetts) in IPv4 (128 bit in IPv6)
- 5 Netztypen (davon nur 3 gegenwärtig von Bedeutung für übliche Netze)
  - Class-A-Adressen: große Netze mit sehr vielen Hosts

Class-B-Adressen: mittelgroße Netze  
 Class-C-Adressen: kleine lokale Subnetze, max. 254 Hosts  
 Class-D/E-Netze: Multicast-Adresse

- 32 bit (IPv4) unterteilt in  $4 * 8$  bit, jeweils eine Dezimal-Zahl zugeordnet

### Domain Name System (DNS)

Um statt IP-Adressen auch logische Namen zuzuordnen: <host> . <domain>

z.B. Logische IP-Adresse: drossel.informatik.uni-leipzig.de

Physische IP-Adresse: 139.18.12.38

Anfangs erfolgte die Zuordnung über eine zentral gehaltene Datei (/etc/hosts auf UNIX-Systemen) des Network Information Centers (NIC). Diese Adresse wurde jeder Domain regelmäßig über ftp verschickt. Unproblematisch, als Adressraum noch flach war. Wachsen Internet => hierarchischer Adressraum => Verschicken Host-Adresse mit ftp zu aufwendig.

/etc/hosts ist mittlerweile eine Datenbank, aufgeteilt in Zonen. Verwaltung innerhalb einer Zone durch dedizierte Rechner, sog. Domain Name Server. DNS verwalten nur einen Teil des Domain Name Space. DNS setzt sich aus 3 Hauptkomponenten zusammen

- Domain Name Space (baumartiger, hierarchisch strukturierter Namensraum)
- Name Server (Verwaltung des Namensraum, Primary / Secondary Name Server, NIC)
- Resolver (stellt Anfragen für Client an Server).

### Transportprotokoll TCP (Transmission Control Protocol)

Protokoll der Host-to-Host-Schicht (OSI-Schicht 4). Sichert zuverlässige Datenübertragung (ISO / OSI: "verbindungsorientierter Dienst"). Stellt bidirektionale Verbindung zwischen 2 Endpartnern her. Zuverlässig = gesicherte Datenübertragung und Anwendung gängiger Sicherungsverfahren, u.a.

- \* Sequenz-Nr.-Vergabe,
- \* Prüfsummenbildung mit Empfangsquittungen,
- \* Quittungen mit Zeitüberwachung (Time-Out),
- \* Sliding-Window-Verfahren (Fenster- und Kreditsteuerung).

3 Phasen der Übertragung: Verbindungsaufbau, Datenaustausch, Verbindungsabbau.

### Transportprotokoll UDP (User Datagram Protocol)

Anwendung für nicht-gesicherte Verbindung zwischen 2 Kommunikationspartnern, insbes. für UNIX-Anwendungen (z.B. falls im LAN die Sicherung gewährleistet), Mbone.

UDP ist ein einfach gestalteter, verbindungsloser (Datagram)-Dienst. Enthält gegenüber IP lediglich noch zusätzlich Portnummern und eine Prüfsumme. UDP besitzt nur die DÜ-Phase; schneller gegenüber TCP.

#### Portnummern

Kommunikation auf Transportebene erfolgt über sog. Portnummern. Sie ermöglichen das Ansprechen unterschiedlichster Dienste.

Port-Nr. und IP-Adresse bilden den Kommunikationsendpunkt (T-SAP). Port-Nr.: 16 bit => von einem Rechner aus max. 65 535 unterschiedliche Ports realisierbar.

Für einige Standarddienste werden von der IANA (Internet Assigned Numbers Authority) sog. *Well-Known-Portnummern* vergeben. Es sind Vereinbarungen, auf welche Ports bestimmte Dienste hören, z.B.

FTP-Server	Port-Nr. 21
Telnet	Port-Nr. 23
Gopher-Server	Port-Nr. 70
WWW-Server	Port-Nr. 80

## 10.4 Verzeichnisdienste im Internet

### 10.4.1 Internet Domain Name System (DNS)

**DNS** wurde für die Nutzung im **Internet** entwickelt. Der ursprünglich flache Namensraum konnte nur noch sehr ineffizient gehandhabt werden, weswegen DNS **stark partitionierte Namensräume** mit einer tiefen Hierarchie verwalten kann.

DNS-Namen beschreiben von rechts nach links den Weg von der Wurzel zum jeweiligen Blatt des Namensbaumes. Die einzelnen Komponenten des Namens werden durch Punkte getrennt und veranschaulichen näherungsweise die hierarchische Partitionierung des Namensraumes.

Zur Auflösung eines Namens ist stets der komplette Name notwendig, relative Pfade werden nicht verarbeitet. Es werden zwei Anfragen durch Clients unterstützt. Die Namensauflösung (**Host Name Resolution**) liefert zu einem Namen (z.B. isun.informatik.uni-leipzig.de) die entsprechende IP-Adresse zurück. Die **Mail Host Location** Anfrage liefert zu einer Email-Adresse eine geordnete Liste mit Rechnernamen, die für diese Adresse eine Mailbox enthalten.

Einige Implementierungen verarbeiten weitere Anfragen, wie **Reverse Resolution** (liefert Namen zu IP-Adressen), **Host Information** (Informationen über einen Rechner, aber Sicherheitsproblem) oder **Well Known Services** (gibt Auskunft über dort verfügbare wohlbekannteste Dienste, wie telnet, ftp oder www und deren Protokolle).

Die Namensdaten sind in Zonen unterteilt. Eine Zone enthält folgende Daten:

1. Attribute aller Namen der Domäne (außer der Subdomäne),
2. Namen und Adressen der Namensserver, die für die Zone maßgebend sind,
3. Namen der Server mit maßgebenden Daten von Subzonen,
4. Managementparameter (u.a. Caching, Replikation).

Jede Zone muss mindestens einmal repliziert werden (Primär- und Sekundärserver), wobei die Sekundärserver die Daten periodisch vom Primärserver laden. Cache-Einträge auf diesen Servern besitzen ein Verfallsdatum (**time-to-live**), nach dem sie aus dem Cache gelöscht werden. DNS ist ein primitiver Namensdienst, der nur wenige Funktionen anbietet. Daher oft mit anderen lokalen Diensten, wie dem **Network Information Service (NIS)** kombiniert, der dann unter anderem Benutzernamen und Passwörter liefert.

### 10.4.2 X.500 Directory Service

**X.500** ist ein **hochwertiger, attributierter Verzeichnisdienst**, standardisiert durch CCITT (ITU-T), und beschreibt eine Anwendung der 7. Schicht des OSI-Referenzmodells. Er wird zusammen mit GNS im DCE eingesetzt, genauso in einigen Implementierungen von CORBA. Der Namensbaum von X.500 (Directory Information Tree, DIT) ist als Datenstruktur mit allen Attributwerten in einer Datenbank, der **Directory Information Base (DIB)** gespeichert.



Abbildung 10.7: Ausschnitt aus einem X.500 DIT

Jeder DIB-Eintrag besteht aus einem Namen, der zur Auflösung immer komplett (absolut) angegeben werden muss, und einer Menge von Attributen. Diese besitzen einen ausgezeichneten Typ und mehrere Werte und können zwingend oder optional sein. Typen sind durch Typnamen und eine Syntaxdefinition in ASN.1 spezifiziert. Alle Einträge werden zudem jeweils durch ein **Object-Class-Attribut** einer Objektklasse zugeordnet, wobei neue Klassen definierbar sind.

Der Name eines DIB-Eintrages wird als **Distinguished Attribute** zur Positionierung im Baum verwendet und dient als Navigationsanker für den gesamten Eintrag. Laut Standard soll es auf der Welt nur eine einzige DIB geben, deren Daten auf individuelle X.500-Server verteilt sind.

In der Nomenklatur von X.500 heißen die Server **Directory Service Agent (DSA)** und die Clients **Directory User Agent (DUA)**. Ist ein Verzeichnis auf mehrere DSAs verteilt, so kommunizieren diese untereinander, um eine Suchanfrage zu bearbeiten.

Der X.500-Standard schreibt weder eine Implementierung noch die Navigationsform vor. Ein Nachteil des Standards ist, dass die Definition der Objektklassen auf nationaler und internationaler Ebene erfolgen muss, um weltweites Suchen zu ermöglichen.

### 10.4.3 Lightweight Directory Access Protocol (LDAP)

#### Entwicklung LDAP

X.500 zu komplex und schwierig auf PC implementierbar ~> Beginn 90er Entwicklung zweier unabhängiger Protokolle für vereinfachten X.500-Zugriff (über IETF):

- *Directory Assistance Service (DAS, RFC 1202)* und
- *Directory Interface to X.500 Implemented Efficiently (DIXIE, RFC 1249)*.

Beide Protokolle setzen als X.500 direkt auf TCP/IP auf und beruhen auf Prinzip eines *Übersetzungsservers*, der Befehle des einfachen Protokolls in das komplexe, X.500-eigene Directory Access Protocol (DAP) übersetzt.

IETF: Verstärkung der Entwicklung, Fokussieren auf leichtere Protokolle -> Ergebnis: *LDAP (Lightweight Directory Access Protocol)*. Erste Veröffentlichung 1993 (RFC 1487). LDAPv2 erste nutzbare Version (RFC 1777), LDAPv3 erschien 1997.

#### Verzeichnisdienst LDAP

LDAP: Lightweight Directory Access Protocol: Standardisiertes Protokoll ~> regelt den Austausch zwischen Verzeichnisdiensten.

Vereinfachung des X.500 Protokolls (Ursprung im X.500). Einfache Struktur ~> rasante Verbreitung und hohe Akzeptanz. Es definiert:

- Datenaustausch und Datenformate,
- Aufbau von LDAP-Verzeichnissen und Zugriff auf die Dienste.

LDAP sichert Interoperabilität zwischen verschiedenen (LDAP kompatiblen) Verzeichnisdiensten: Datentransferformat LDIF, Programmierinterface LDAP API.

Modelle im LDAP:

- Informationsmodell (Speicherung der Daten),
- Namensmodell (Strukturierung der Daten),
- Funktionsmodell (Ablage und Auslesen von Informationen),
- Sicherheitsmodell (Zugriffsschutz der Daten).

#### Standalone-Dienst (slapd)

Implementation von X.500 Server Software immer noch kompliziert. Zugriffsstatistik: nahezu 99% aller Anfragen auf X.500 Server erfolgen über LDAP.

Somit Entwicklung eines eigenen Verzeichnisservers auf Basis LDAP. Entfernen des zwischen Client und X.500-Server geschalteten LDAPD (LDAP Daemon), der nur LDAP-

Anfragen in X.500-DAP-Anfragen und andersherum übersetzte. Dadurch Komplexität des Gesamtsystems drastisch reduziert. Außerdem wurde Ziel eines weltweiten X.500-Verzeichnisses immer unwahrscheinlicher.

1995 Entwicklung des *Standalone LDAP Daemons (SLAPD)* (Unterstützung durch NSF: National Science Foundation).

Erste *slapd*-Version im LDAP Package 3.2 der Universität Michigan (1995). Damit LDAP unabhängig von X.500. Mit LDAPv2 Aufbau von SLAPD Verzeichnisse-Server-Netzen möglich.

1996 Ernennung LDAPs zum *Internet Verzeichnisdienst-Protokoll erster Wahl*.

LDAP Version 3 (LDAPv3) ist wesentlich ausgereifter als Version 2. Heute anerkannter Internet Standard. Verbesserung in vielen Bereichen.

### **Funktionsweise LDAP**

LDAP ist ein Client/Server-Protokoll: Client sendet Anfrage an Server und erhält vom Server eine positive oder negative Antwort nach vorangegangener Bearbeitung. Bekannte Beispiele für Client/Server-Protokolle: HyperText Transfer Protocol (HTTP) oder Internet Message Access Protokoll (IMAP). Grundgedanke bei dieser Form der Protokolle: Rechenarbeit wird von Systemen übernommen, die dafür konzipiert sind.

LDAP ist ein nachrichtenorientiertes Protokoll, d.h. Client erzeugt eine Nachricht, die den Request enthält, und schickt diese an der Server. Server verarbeitet den Request und antwortet dem Client in einer Serie von Nachrichten, u.a. einen Result Code.

Dadurch, dass mehrere Anfragen parallel abgearbeitet werden können, und die Reihenfolge die Antworten unabhängig von der der Anfrage ist, kann LDAP wesentlich effizienter arbeiten als beispielsweise HTTP, dass eine schrittweise Abarbeitung durchführt.

Insges. 9 Basis-Protokoll-Operationen, die in 3 Kategorien zusammengefasst werden können:

- Anfrage-Operationen: *search, compare*
- Update-Operationen: *add, delete, modify DN (rename)*
- Authentications- und Control-Operationen: *bind, unbind, abandon*

Mit *bind* und *unbind* meldet sich Nutzer nach einer anonymen Anmeldung am LDAP-Server an und ab (er „bindet“ quasi seine Identität an die vorhandene Verbindung). Durch Kombination dieser 9 Operationen sind komplexe Anfragen und Aufgaben erfüllbar.

## **10.5 Internet-Zugang**

### **Internet-Anschluss**

Internet-Zugang („Internet-Anschluss“ / „Internet-Verbindung“): kennzeichnet die Nachrichtenverbindung eines Computers oder Netzwerkes mit dem Internet (Quelle: wikipedia).

Unterscheidung zwischen

- *Breitband-Datenfernübertragung*: i.d.R. per DSL- oder Kabelmodem (DSL: Digital Subscriber Line),
- *Schmalbandverbindungen*: per Modem (analog) oder ISDN (digital),
- *Drahtlose Verbindungen*: per MFN (Paketfunk) GPRS, UMTS, HSDPA, EDGE, LTE.

### **Protokolle**

Für den authentifizierten Internet-Zugang werden bspw. genutzt:

- PPP (Point-to-Point Protocol): über Wählleitung (früher SLIP),
- PPTP (Point-to-Point Tunneling Protocol): häufig in Österreich, Italien und Belgien, weniger in Deutschland.

### **Voraussetzungen für Internet-Zugang**

Internet-Verbindung i.d.R. nur über *Dienstleistungsvertrag* mit einem Internet-Zugangsanbieter (ISP: Internet Service Provider) möglich. Gilt auch für Internet-by-Call-Verbindung.

Verbindung wird über einen Einwahlknoten bzw. Breitband-PoP (POP: Post Office Protocol) und einem kundenseitig installierten Modem oder ISDN-Endgerät aufgebaut. Im Falle eines

leitungsgebundenen Zugangs erfolgt dies regelmäßig über die Teilnehmeranschlussleitung oder das TV-Kabelnetz.

Endgerät muss mit dem Computer oder lokalem Rechnernetz verbunden sein:

- per Kabel (drahtgebunden),
- Wireless LAN, Bluetooth (Funk) oder
- Stromnetz.

Weiterhin ist für das Modem ein Gerätetreiber erforderlich (steuert den Datendurchsatz), z.B. bei DSL mindestens 1024 kbit/s download und 128 kbit/s upload.

#### ***Mögliche Anschlussverbindungen***

- Wählleitungen (dial-up), ISDN, DSL (ADSL, SDSL, VDSL, Portabel DSL, SkyDSL),
- Kabelmodem, WiFi, WiBro, Wimax,
- Mobilfunk (CSD, HSCSD, GPRS, EDGE, UMTS, HSDPA, LTE),
- Glasfasernetz,
- Trägerfrequenzanlage (TFA, über das Stromnetz) und xMax (nur Nordamerika).

#### **Kosten**

- Grundgebühr und Verbindungsentgelte (je nach Vertrag).  
Verbindungsentgelte richten sich pro Abrechnungsperiode nach
  - dem Datenvolumen der übertragenen Daten (Volumentarif) oder
  - dem Zeitraum der Nutzung einer Verbindung (Zeittarif) oder
  - unlimitiert hinsichtlich Volumen und Zeit (pauschal per Flatrate).
- Anbieter freier Funknetze bieten kostenlosen Internetzugang (aus Idealismus oder künftiger Kundenbindung).

#### **Nutzung**

##### ***Nutzungsszenarien***

- IP-Telefonie und ISDN over IP,
- Internet-Surfen:
  - Ansteuern von Dateien von einem Client aus, die über Server erreichbar sind (u.a. Homepages/Websites/Netzpublikationen),
  - E-Mails (sofern via POP 3),
  - Webcam,
- Datenübertragung zwischen Server und Client, oder dezentral zwischen vielen Clients untereinander (z.B. durch FTP oder Peer-to-Peer),
- IPTV (Fernsehen).

##### ***Benutzte Protokolle***

- Internet Protocol (IP), Internet-Protokollfamilie,
- für moderne Mobiltelefone: WAP und i-mode.

##### ***Technische Aspekte***

Eine Internet-Verbindung besteht ab der Einwahl in das Netz des ISP, nicht zwingend durch das Aktivieren des Browser-Fensters. Nach 24 h Online-Zeit wird bei vielen Providern eine Zwangstrennung vorgenommen. Internet-Zugang möglich an

- Arbeitsplätzen (Festnetz bzw. WLAN)
- Notebook, PDA, Smartphone und Tablet PC mit Paketfunk (Internet-Zugang),
- Internet-Terminals (Münzeinwurf oder Wertkarte).

Viele Internet-Anwendungen erfordern zunehmend einen Breitband-Anschluss.

***Verbreitung*** (i.w. Industrienationen und Schwellenländer): weitgehend in

Australien, Dänemark, Deutschland, Finnland, Frankreich, Großbritannien, Hongkong, Irland, Island, Italien, Japan, Kanada, Luxemburg, Macau, Neuseeland, Niederlande, Norwegen, Österreich, Polen, Portugal, Schweden, Schweiz, Singapur, Südkorea, Thailand, Türkei, Vereinigte Staaten.

## 11 World Wide Web (WWW)

### 11.1 Merkmale, Dienste und Entwicklung des WWW

#### Charakteristika des WWW (World Wide Web, W3, Web)

Weltweites, in HTML realisiertes Informationssystem im Internet, das interaktiven Zugriff auf verknüpfte Dokumente (Geflecht, "Web") im Internet erlaubt. Links auf andere (verteilte) Dokumente, eindeutige Adressierung (URL).

Verteiltes Hypermedia-System (Multimedia-Daten: Text, Audio, Video und Animationen)

Hypertext / Hypermedia-Format: HTML, xHTML (Basis: SGML, XML) ... HTML5

Kommunikationsprotokolle: HTTP, TCP/UDP, IP

Weitere Informationssysteme: Finger, Gopher, Netfind, Wais, Whois u.a.

#### Services im WWW

Navigationsystem (Anzeige, Suchen, Darstellung, Speicherung: Hypertext)

Asynchrone Kommunikation: E-Mail (POP, IMAP)

Interaktionen im Web: JavaScript, Applets, JSP, ASP, CGI, PHP, AJAX

Virtuelle Räume: VRML

Diskussionssystem: NetNews ("schwarzes Brett")

Verteilte Systeme (kooperatives System): Web-Services (SOAP, WSDL, XML, UDDI), SOA (Service Oriented Architecture)

Web-Applikationen (Web 2.0: Youtube, Flickr, RSS, Blogs, Twitter, Wiki, ...).

#### Grundlagen

Client/Server-Architektur: Client (Browser), Server.

Speicherung: HTML-Format (HyperText Markup Language)

Protokolle: HTTP (HyperText Transfer Protocol) als dienstspezifisches Protokoll.

TCP/IP-Protokollstack als transportierende Protokolle.

Hypertext-Konzept (erweitert Hypermedia): Geflecht logisch zusammenhängender Dokumente (Web). Mit Verknüpfungen (Links), spezifiziert durch URL (Uniform Resource Locator).

Auszeichnungssprachen: HTML, xHTML (statisch) - Basis: SGML

(Datentypen) dHTML (interaktiv), ..., XML (kooperativ), HTML5

Präsentationssprachen. CSS, XSL

Gestaltung von Interaktionen: CGI (Common Gateway Interface), PHP, JavaScript, AJAX

Navigationsystem: erlaubt freie Bewegung von Dokument zu Dokument, unabhängig von der Lokalisierung im Netz.

#### Entwicklung des Internet-Informationssystems WWW

##### Ursprung:

1989, CERN (Schweiz, Genf): Europäisches Zentrum für Teilchenphysik. Tim Berners-Lee. Aufgabe: Entwicklung eines neuartigen Systems zum Informationsaustausch.

Verwendung **Hypertext-Technologie**: Erzeugung eines Geflechts (engl.: **Web**) von logisch zusammenhängenden Dokumenten. Dokumente auf vielfältigste Art und Weise nach Informationen durchsuchbar. Verschiedenste Verknüpfungen nach innen und außen (über sog. **Links**) (keine hierarchische Struktur bzw. lineare Liste).

Implementation des Hypertext-Konzept in einem **Navigationsystem**. Dies erlaubt innerhalb des Netzes freie Bewegung von Dokument zu Dokument, unabhängig von der Lokalisierung der einzelnen Dokumente.

*Tim Berners-Lee und Robert Caillau (beide CERN)*: Vorschlag eines auf einer Client/Server-Architektur aufbauenden, hypertextbasierten Systems.

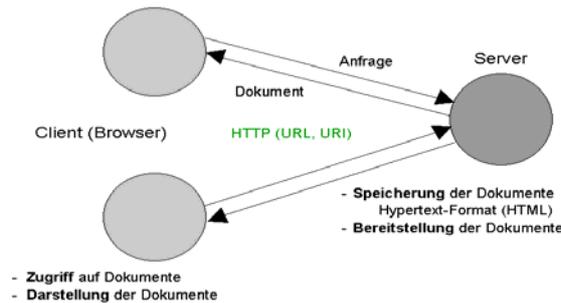


Abbildung 11.1: Client/Server-Architektur des WWW

### Kommunikationsprotokolle

- HTTP (HyperText Transfer Protocol)
- TCP / IP (Transmission Control Protocol / Internet Protocol)

### Adressierung

- URL (Uniform Resource Locator): Adressierung eines Endpunktes auf einem Rechner
- URI (Uniform Resource Identifier): Adressierung einer auf dem Rechner befindlichen Ressource, Übermenge der URL.

### Entwicklung des WWW

Weitere Browser-Entwicklungen: Mosaic 2.0, WinMosaic, MacMosaic.

- Marc Andreesen und James Clark gründen Firma Mosaic Communications Corp.
- 1993: ca. 100 öffentliche Web-Server; Vorstellung WWW auf der "Online-Publishing"

Weitere Browser (für WS/X-Window (Unix) bzw. PC/MS-Windows), u.a.

Netscape Communicator, Mozilla Firefox, Internet Explorer (MicroSoft), Opera, Chrome  
Bereits 1998 mehr als 16 Mio. Web-Server.

1994: Gründung des World Wide Web Consortiums (W<sub>3</sub>C, Vereinbarung CERN und MIT):  
Organisation zur Weiterentwicklung Web, Protokoll-Standardisierung und Interoperabilität  
zwischen Web-Sites. Direktor: *Tim Berners-Lee*.

MIT führt US-Teil des Konsortiums, INRIA (französisches Forschungszentrum) führt europä-  
ischen Teil. Home-Page des Konsortiums: <http://www.w3.org>

### Paradigmen des WWW

Ursprung (1989): Textdokumente, Navigation. Klassisches Web (Web 1.0) ist statisch, nicht-  
interaktiv, synchron.

Paradigmenwechsel (ab 2000): dynamisch, multimedial, interaktiv, asynchron, semantisch,  
kooperativ (Web-Services), mobiler Zugang.

- Dynamisch und multimedial:

*Konventionelles Web 1.0: HTML-Seiten über längeren Zeitraum unverändert im Netz (sta-  
tisch), ggf. Überarbeitung und Austausch nach bestimmter Zeit. Abhilfen:*

Content-Management-Systeme, um Inhalte dynamisch auszutauschen.

Web 2.0 (social Web): neues Netzverständnis (weniger neue Architektur – vs. W<sub>3</sub>C)

Mitmachnetz, Kombination mit anderen Seiten, Read/Write-Techniken,

Abonnementdienste (Feeds, Pull-Dienst für Austausch): Formate RSS, Atom u.a.

Neue Social-NW-Dienste, wie Youtube, Flickr, Wiki, Weblog, Facebook, Twitter u.a.

Podcast: Veröffentlichung von Audio- und Videodateien.

- Interaktiv:

HTML-Seiten dynamisch, benutzerabhängig und interaktiv gestalten (bereits im Web 1.0)

Clientseitig: Scriptsprachen (z.B. JavaScript, VBScript), Java-Applets, ...

Serverseitig: vom Server werden neue HTML-Seiten gesendet: JSP, ASP,

CGI (Common Gateway Interface), PHP (Personal Hypertext Pre-Processor), ...

- Asynchron:  
*Konventionelles Web 1.0: Client blockiert bis Serverantwort. Abhilfe:*  
 AJAX (Asynchronous JavaScript and XML) ermöglicht asynchrone Kommunikation (nicht-blockierend) zwischen Browser und Server. Nutzung XMLHttpRequest-Objekt in modernen Browsern.
- Kooperativ:  
 Einsatz Web-Services zur Gestaltung verteilter Applikationen im Web.

### Vision des W<sub>3</sub>C

#### Ursprung:

Tim Berners-Lee: CERN, 1989, Paper „Information Management: A Proposal“: „In 10 years, there may be many commercial solutions to the problems above, while today we need something to allow us to continue“. 15 Jahr später: „Sir Timothy“ by Queen Elizabeth II.

#### WWW Consortium (W<sub>3</sub>C):

WWW machte Internet zum Massenmedium: private Kommunikation, e-Commerce, Web 2.0.

W<sub>3</sub>C (1994): Standardisierung für eine offene und einheitliche Infrastruktur:

- Weiterentwicklung, Patente (seit 2003), Website (www.w3.org),
- Standards (Empfehlungen, sog. W<sub>3</sub>C recommendations), u.a.  
 HTML (Hypertext Markup Language), XML (Extensible Markup Language, 1998),  
 Mobilität (Geräteunabhängigkeit, Multimodalität), Präsentationstechnik (grafische Schnittstellen, Multimedia) Datenbanken, Web Services, Semantisches Web

#### Entwicklung des WWW:

- Web-Clients (Browser): Basis HTML, fehlende Autorenwerkzeuge für korrektes HTML. Proprietäre Entwicklungen: HTML 2.0 -> XHTML -> HTML5<sub>2009</sub>.
- Standard der W<sub>3</sub>C: XML (1998): Basis aller weiteren Entwicklungen, auch für .NET und J2EE. XML bietet einheitliches Datenformat für Anwendungen und Systeme.
- Entwicklung „social“ Web 2.0 (kollaborative Anwendungen, wie YouTube, Flickr, Wiki, Blogs, Tagging ...). Einsatz AJAX für asynchrone Kommunikation zw. Browser - Server.
- Entwicklung zum semantischen Web (Web 3.0?: Semantik, Sprache ...).

#### Das Web der Zukunft (nach Vision des W<sub>3</sub>C und WHATWG):

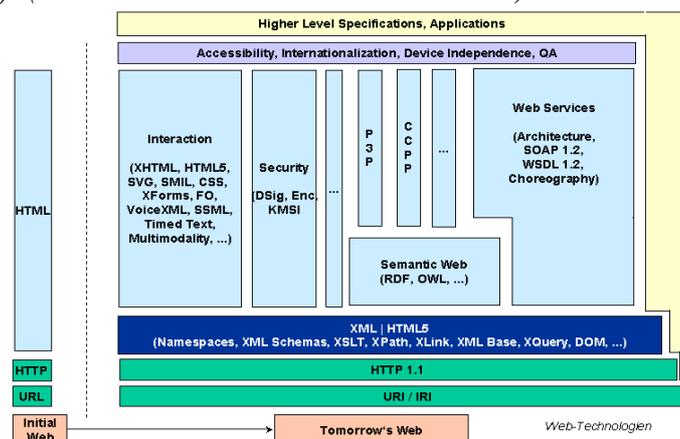


Abbildung 11.2: Tomorrow's Web-Technologien

#### Geräte- und Softwareunabhängigkeit von Informationen:

- Browser in PC, PDA, Fernseher, Handy, Drucker, Audioradios ... präsentieren WWW. Information an logischer Struktur nicht an Präsentationsform orientieren -> HTML, XML
- WWW auch interaktiv (nicht Einwegkommunikation) ~> geräteunabhängige Modellierung von Dialogen, Stilen, Multimedia (Grafik, Video, Sprache).

- XML-basierter Standard XForms erlaubt gleichen **Dialog** über bspw. Desktop Computer, WAP Handy, normales Telefon via Sprachsteuerung.
- Medienabhängige Definition von **Präsentationsstilen** durch die Stylesheet-Standards Cascading Stylesheets (CSS) und XML Style Language (XSL) ~> damit Präsentation geräteunabhängig und gemäß Benutzeranforderung.
- Hohe grafische und multimediale Anforderungen durch die **XML-Sprachen** Scalable Vector Graphics (SVG) und Synchronized Multimedia Integration Language (SMIL).  
Für Dialoge mit gesprochener Sprache: **Voice XML**.

Multimodalität als Schlüssel zur Mobilität:

- Mensch kommuniziert parallel über verschiedene **Sinne**: Gehör, Auge, Geruch, Berührung ergänzen die Nachrichtenübermittlung. Unterstützung durch Computer (Multimodalität): Spracheingabe, Erkennung von Handschriften, ... ~> Extensible MultiModal Annotation Markup Language (**EMMA**).
- Positionsbestimmung und **Navigation**: GPS-Tools, Routenplaner in Autos bzw. PDA. Bordcomputer arbeiten automatisch über Audioeingabe und -ausgabe, automatische Aktualisierung gemäß Verkehrslage und Fahrtfortschritt.
- **Synchronisation** mehrerer Geräte in einer gemeinsamen Sitzung, z.B. Handy und Laptop ~> Bestandteil des „*Multimodal Interaction Frameworks*“.

Web-Services:

- Kooperative Arbeit: Verteiltes System, Nachrichtenaustausch, nomadic Computing.  
Voraussetzungen: Nachrichtenaustausch zwischen den jeweiligen Diensten.
- Standard für Web-Messaging: SOAP 1.1 (Simple Object Access Protocol). Wie Email (asynchron), aber auch Austausch für Nicht-Email-programme.
- Beschreibung der Dienstschnittstellen (Format der Daten, Dienstfunktionen, ...):  
Web Services Description Language (WSDL).
- Veröffentlichung der Dienste: Universal Description, Discovery, and Integration (UDDI).  
Verwaltet durch OASIS.
- In Vorbereitung: Standard für Zusammenarbeit von Webdiensten: Web Services Choreography. Festlegung zu Sender und Empfänger und welche Nachrichten.

Semantisches Web:

Weiterer Schritt Berners-Lee u.a. 2001: Einbezug wissensintensiver Dienste. „Wissen“ maschinenlesbar speichern ~> erfordert Begriffe, logische Beziehungen und deren formale Beschreibung. Dazu 2 Standards durch W<sub>3</sub>C: RDF und OWL.

- Resource Description Framework (RDF): Damit können Relationen von in Dokumenten benutzten Begriffen definiert werden.
- Web Ontology Language (OWL): erlaubt Definition von Objektklassen, Klassenzugehörigkeiten und Eigenschaften. Verbindung von Web-Services über Ontologien befreit von Notwendigkeit, Informationen über vordefinierte Nachrichtenstrukturen auszutauschen.

Semantic Web sehr anspruchsvoll. Realisierung dahin, das immer mehr Informationen im Web mit Semantik definierenden Metadaten (via RDF, OWL) markiert werden. Diese vorab von *Suchmaschinen* genutzt, aber zunehmend auch von abarbeitenden Agenten.

## Web 2.0

Web 2.0-Konferenz 2004: Dale Dougherty, Craig Cline, 2005: Tim O'Reilly – vs. W3C.

Schlüsselprinzipien, u.a. datengetriebene Anwendungen, Mitwirken der Nutzer, Komponenten von verschiedene Entwicklern

Benutzer erstellen und verteilen Inhalte selbst, unterstützt von interaktiven Anwendungen. Vom Push-Prinzip (aktive Verteilung) zum Pull-Prinzip (Abonnement). Mediale Nutzung: Social Network, mit Reihe interaktiver und kollaborativer Elemente.

Techniken:

- Abonnementdienste (sog. Feeds) mit Formaten RSS bzw. Atom. Ermöglichen den Infor-

mationsaustausch zwischen Web-Seiten,  
 - Asynchrone Kommunikation (AJAX), kollaborative Verarbeitung (Web-Services).  
 Neue Anwendungen im Web 2.0:

- Weblog, Youtube, Flickr, Wiki, Facebook, Twitter u.a.
- Podcast (Audio/Video), social Bookmarks (persönliche Links), Media Sharing Platforms (Profile anlegen, Daten speichern, Inhalte anderer Nutzer konsumieren).

## HTML5

Textbasierte Auszeichnungssprache zur Strukturierung und semantischen Auszeichnung von Inhalten (Texte, Bilder, Hyperlinks) in Dokumenten.

- Nachfolge von HTML4 (z.Zt. 2011 noch in Entwicklung). Entwicklungsteams der W3C (ursprünglich XML-orientiert -> XHTML) und des Browserhersteller-Consortiums WHATWG („versionsloses“ Konzept -> HTML, Gründer Ian Hickson, Mitarbeiter bei Google) → HTML5. W3C-Recommendation für HTML5: 2014 erwartet.  
 Seit 2009 wird HTML5 als nächste Generation für Auszeichnungssprachen im Web festgelegt (keine neue Variante von XHTML !). Bereits in meisten Browsern implementiert.
- HTML5 ersetzt die Dokumentenbeschreibungsstandards HTML 4.01, XHTML 1.0 und DOM HTML Level 2.
- HTML5 bietet viele neue Funktionalitäten, u.a. Video, Audio, lokale Speicher, dynamische 2D- und 3D-Grafiken (somit keine zusätzlichen Plugins wie Adobe Flash erforderlich).

## 11.2 Hypertext-Konzept

### Hypertext und Hypermedia im WWW

*Ausdruck „Hypertext“:*

Ted Nelson (1960er Jahre): definiert es als „nichtlineares Schreiben“ (nicht nur zum Suchen sondern auch zum Schreiben). *Xanadu* - von Nelson verbreitete Vision eines weltweiten Hypertext-Systems. Vor dem WWW wurden eine Vielzahl von Hypertextsystemen implementiert. Tim Berners-Lee und Robert Caillau verwendeten die Hypertext-Idee in Zusammenhang mit einer Internet-basierten Client/Server-Architektur.

*Definition von HyperText:* „HyperText is a way to link and access informations of various kinds as a web of nodes in which the user can browse at will“.

*Hypertext-System:* Menge von Textdokumenten mit Verweisen auf andere Dokumente. Hypermedia: Erweiterung um Multimedia-Elemente (audio, video, Grafik). Hypertext enthält

- Text (sog. „Plain Text“)
- Auszeichnungen („Tags“): vordefiniert bzw. einstellbar (Style Sheets),
- Formatierungen (in Erweiterung auch Listen, Tabellen, Frames udgl.)
- Querverweise („Hyperlinks“) zu anderen Textstellen innerhalb des Textes bzw. zu anderen Textdokumenten; i.allg. farblich markiert und mit Mausclick aktivierbar.

### HTML: HyperText Markup Language

Im WWW verwendete Hypertext-Sprache. In diesem Format werden die Dokumente auf dem Server abgespeichert (sog. Hypertext-Format). HTML-Texte nutzen bei Verweisen auf andere Dokumente die Möglichkeiten des Internet; d.h. Dokument auch auf entferntem Rechner.

Realisierung Client/Server-Verbindung: HTTP (HyperText Transfer Protocol)

Lokalisierung (Dokumente, Links): URL (Uniform Resource Locator = weltweite Adresse)

*Darstellung am Browser:*

Dokumente im Web werden in der Hypertext Markup Language (HTML) geschrieben. HTML ist standardisiert, erlaubt formatierte Anzeige von Inhalten (sog. Auszeichnungssprache zur Strukturierung von Inhalten). Erweiterung durch Präsentationssprachen, sog. Style Sheets (z.B. CSS: Cascading Style Sheets), zur Darstellung von Inhalten; ermöglichen inhaltsunabhängige Formatierung, während Inhalte weiterhin mit HTML ausgezeichnet werden.

HTML allerdings kein spezielles Design eines Dokuments, lediglich Richtlinien für Darstellung, z.B. Schriftgrößen (über Wichtungsmaße) u.ä.

Die HTML-Erweiterungen wurden durch die Browser-Entwicklungen bewirkt. Die physische Darstellung einer Seiten ist vom jeweiligen Browser abhängig. Elemente, die ein Browser nicht kennt, werden ignoriert. Damit können ältere Browser HTML-Dokumente neuerer Version anzeigen (Gefahr unstandardisierter Browser-Elemente).

*Standard:* SGML (Standard Generalized Markup Language): ISO-Norm (ISO 8879, 1986)  
Definition strukturierter Datentypen

HTML-Versionen: 2.0, 3.0 (3.1/3.2), 4.0 (4.01), 5<sub>2009</sub> (W3C-recommendation 2014). Einstellung mittels DTD (Document Type Definition), legt Regelwerk für Syntax fest (HTML-DTD).

W3C setzte ursprünglich auf XML (eXtensible Markup Language): W3C-Standard für Web (insbes. Web 2.0) und Web Services. Im Web dafür XHTML als Auszeichnungssprache bereitgestellt (XHTML 1.0, XHTML 2.0). XHTML-Versionen nutzen XML als Metasprache und definieren ein neues XML-DTD.

Browserhersteller-Consortium WHATWG (Ian Hickson / Google): HTML5 -> seit 2009 zusammen mit W3C als Auszeichnungssprache im Web festgelegt (keine neue Variante für XHTML) -> ersetzt HTML 4.01, XHTML 1.0 und DOM HTML Level 2. HTML5 bereits in meisten Browsern implementiert (wenn auch noch unvollständig).

Erweiterungen bzw. branchenspezifisch: MathML (Mathematical ML), CML (Chemical ML), WML (wireless ML), HDML (Handheld Devices ML), VRML (Virtual Reality Modeling Language), SMIL (Synchronized Multimedia Integration Language, Web 2.0), XML, ...

Weiterhin häufiger Einsatz von

- CGI (Common Gateway Interface): Realisierung serverseitiger Interaktion. Verarbeitung von Eingabewerten (Formulartechnik).
- PHP (Personal Hypertext Pre-Processor): Scriptsprache, serverseitige Web-Technologie, Nutzung Formulartechnik (analog CGI) für Eingabewerte. Unterstützung Web-Services.
- PERL (Practical Extraction and Reporting Language): Erweiterte Scriptsprache. Mischung aus klassischer PS (C) und Scriptsprache (Unix-Shellscript). Häufig zusammen mit Formulartechnik als CGI-Programm am Server für Auswertung der Formulardaten eingesetzt.
- JavaScript: Scriptsprache, clientseitige Interaktion, Dialog Browser – Server (über das in gängigen Webbrowsern vorhandene XMLHttpRequest-Objekt).
- AJAX (Asynchronous JavaScript and XML): Konzept zur asynchronen Kommunikation zwischen Webclient und Webserver. Erforderlich: JavaScript und XMLHttpRequest-Objekt.

### **Bekannte Medientypen im Hypertext/Hypermedia**

Text: Zeichensätze ASCII, Unicode, EBCDIC (IBM).

Textformate: Postscript-Formate PDF, PS, EPS. Proprietäre Textformate, wie MS Word.

Grafik (Vektorgrafik) und Bild (Pixelbild, Graubild, Farbbild):

Bildformate: Bitmap (.bmp), Tagged Image File (.tif), Joint Photographic Expert Group (JPEG, .jpeg), Graphics Interchange Format (.gif), Portable Network Graphics (.png).

Sprache und Ton (Audio):

Audioformate: Waveform Audio Format (\*.wav), Real Audio Format (\*.ra), Music Instrumental Digital Interface Format (MIDI, Musikformat, \*.mid), Motion Picture Expert Group (MPEG): Standard MPEG I für Audiokompression, z.B. Layer 3 (MP3, \*.mp3).

Film und Animation (Video):

Videoformate: Audio Video Interleave-Format (\*.avi), Motion-Joint Picture Expert Group Format (M-JPEG), Motion Picture Expert Group Standard (MPEG-Format, Standards MPEG I, II, III, IV).

### 11.3 Architektur des WWW

#### Client/Server-Architektur des Web

Tim Berners-Lee und Robert Caillau (CERN/Genf): Hypertext-basiertes Informationssystem auf Basis einer Client/Server-Architektur. *Client (Browser)*: Dienstnutzer (Anfrage, Ergebnisdarstellung), *Server*: Diensterbringer (Speicherung, HyperText-Format)

Weltweiter Informationsdienst im Internet (Schichtenaufbau; Protokolle TCP/IP, HTTP).

Ablauf:

1. Client: Anfrage an Server, Lokalisierung Dokument über URL (lokal oder im Netz)
2. Server: Rückantwort (Bereitstellung des HTML-Dokuments)
3. Client: Interpretation des HTML-Dokuments, Auflösen der Links (URL), Darstellung des HTML-Dokuments auf dem Bildschirm

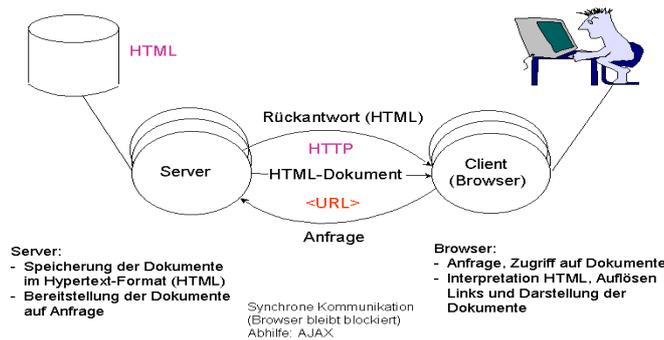


Abbildung 11.3: Komponenten und Ablauf im WWW

#### Architektur des Web-Browsers

Wegen der vielen Details der Dokumente (Zugriff, Darstellung) höhere Komplexität als der Server. Bestandteile eines Browsers: Controller, verschiedene Clients und Interpreter.

*Controller*: Hauptbestandteil eines Browsers.

- Verwaltung der Benutzereingaben.
- Aufruf der Komponenten, um durch Anwender angegebene Operationen auszuführen, z.B. bei HTML-Referenz:  
Aufruf HTTP-Client durch Controller --> lädt Dokument vom entsprechenden Server. Danach startet Controller einen HTML-Interpreter, der das Dokument in geeigneter Form auf dem Bildschirm anzeigt.

*Permanente Komponenten eines Web-Browsers. HTTP-Client und HTML-Interpreter:*

*Optionale Komponenten (Beispiele):*

- Optionale Clients: FTP-Client, Email-Client, VRML-Client, Java Virtual Machine, ...
- Optionale Interpreter : JavaScript, Jscript, ...
- Browser erkennt dies aus der in der URL angegebenen Protokollversion (Zugriffsart). Damit wird jeweilige Komponente gestartet, die das Dokument vom entfernten Server lädt.

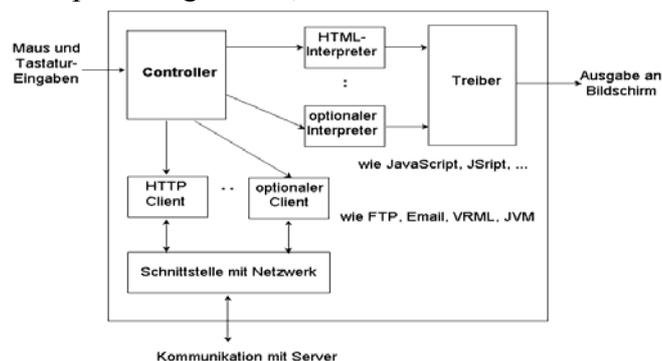


Abbildung 11.4: Architektur eines Web-Browsers

**WWW - Browser**

*Browser* (engl.: to browse - sich umsehen): Clients in Client/Server-Architektur.

Stellen Suchabfrage an die Server, Darstellung des HyperText-Dokuments und Auflösen der Verknüpfungen (Links). Moderne Browser mit Interpreter, u.a. JavaScript-Interpreter, JVM (Java Virtual Machine). WWW-Software für Browser auf den meisten Plattformen verfügbar  
 Fensterorientierte Browser: Besonders geeignet für Navigation im WWW; Auswahl für PC/Windows: Mosaic, Netscape Communicator, Mozilla Firefox, Internet Explorer. Damit

- Darstellung/Anzeige *multimedialer Dokumente* (Texte, Grafiken, Videos)
- Einstellen *lokaler Konfigurationen*, z.B. Home Document (Home Page: Seite, die bei Aufruf des Browser angezeigt wird), lokale Drucker, Positionen der Hilfedateien.

*Auswahl moderner Browser*

- Für Betriebssysteme von Microsoft (Windows 7, Vista, XP, ...), Apple, Unix/Linux und PDA/Smartphones gibt es eine Reihe von Browsern unterschiedlicher Leistungsfähigkeit.
- Auf Betreiben der EU muss Microsoft ab 17.03.2010 (in DE) den Nutzern neben dem vorinstallierten Internet Explorer auch den Zugang zu anderen Browsern erleichtern.
- Auswahl der verfügbaren Browser auf der Web-Seite des „BrowserChoice“-Konsortiums (Microsoft, [www.browserchoice.eu](http://www.browserchoice.eu)), z. Zt. 12 Programme (Stand 03/2010):
 

- Firefox (Mozilla)	50 % in DE, Spitzenreiter seit 12/2009
- Internet Explorer 8 (Microsoft, Windows)	41,3 % -> IE 9 <sub>2011</sub>
- Safari (Apple)	3,6 %
- Opera (insbes. für Linux)	2,8 %
- Chrome (Google)	2,3 %
- Flock (Mozilla-Klon), Sleipnir, FlashPeak, Green Browser, Maxthon, K-Meleon, Avant.	

**Uniform Resource Locator (URL)**

Dient zur Identifizierung von Dokumenten im Netz (weltweit eindeutig), repräsentiert die Adresse eines Dokuments (Web-Seite) auf einem Server. Quasi-Standard für Quelladressen im Internet, insbes. WWW, FTP, Gopher, AFS, News, Mailto, Wais, Telnet ...

Allgemeine Adressierungsform:

URL ::= zugriffsart:// rechnername [: port]/ dateiname bzw. verzeichnis  
 zugriffsart ::= http | ftp | gopher | news | ... (verwendetes Zugriffsprotokoll)  
 rechnername ::= dienst.IP-Adresse  
 dienst ::= www | ftp | gopher | ...  
 port ::= Port-Nr., über den das Dokument abgerufen wird (optional)  
 Syntax: alle druckbaren Zeichen (außer Leerzeichen)

*Beispiele:*

WWW: <http://www.informatik.uni-leipzig.de/index.html> (Home-Page des IfI)

<http://www.w3.org> (Homepage des WWW-Consortiums)

FTP: <ftp://ftp.ask.uni-karlsruhe.de/pub/info/dfg-info/cipformulare.ps.Z> (CIP-Antrag)

Gopher: <gopher://gopher.fh-heilbronn.de/00/will/konPer> (Gopher-Server der FH Heilbronn)

News: <news:comp.infosystems.www> (Newsgruppe im Usenet)

**HyperText Transfer Protocol (HTTP)**

HTTP: zustandsloses, objektorientiertes Protokoll für ein verteiltes Hypermedia-Informationssystem. Aktuell: HTTP / 1.1 (RFC 2086).

HTTP regelt Datenaustausch zwischen Web-Server und Browser (*HTTP-Transaktionen*). Datenaustausch besteht aus einer *Anforderung* (request), womit Browser die Daten anfordert, bzw. einer *Antwort* (response), in der Server die Daten an Browser überträgt.

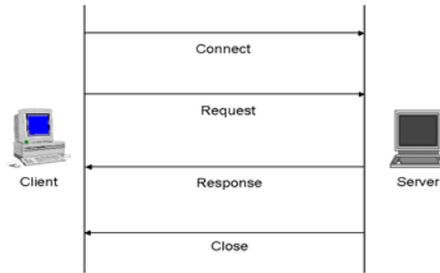


Abbildung 11.5: HTTP-Dienstprimitive

4 Operationen (Dienstprimitive):

*Connect:* Client initiiert eine Verbindung zum Server, die von ihm bestätigt wird. Über TCP erfolgt der Verbindungsaufbau entweder über WellKnown Port 80 oder über jedem anderen nicht reservierten Port (der explizit anzugeben ist).

*Request:* Client stellt eine Anfrage an Server (request message).

*Response:* Server sendet Antwort an Client (response message).

*Close:* Verbindungsabbau durch Server oder durch Clienten infolge Abbruch.

Alle HTTP-Mitteilungen bestehen aus

- einer Reihe von Header-Informationen,
- einer Leerzeile (zwischen Header und Inhalt),
- Inhalt der Mitteilung.

*Header:* Geben Auskunft über die Mitteilung, die Art der Anforderung oder Antwort und über den Inhalt selbst. Die Header unterliegen keiner Ordnungsfolge. Sie bestehen aus einzelnen Zeilen der Form `Headername: Headerwert`.

*Inhalt eines Requests:*

- Request-Zeile: `Methode Request-URL HTTP-Version`
- Allgemeiner Header (Datum, Mime-Version, ...) und/oder
- Request-Header (Autorisierung, From, ...) und/oder
- Inhalts-Header (Allow, Content-Type, Expires, ...)
- Leerzeile
- Inhalt der Nachricht

Jeder Request beginnt mit auszuführender Methode, dann URL und HTTP-Version. Request-Header legt zusätzliche Anforderungen an Server fest, z.B. vom Client akzeptierte Mime-Typen (Art der Dokumente: Text, Bilder, ...), erwarteter Zeichensatz (z.B. ASCII) und ob Server eine Autorisierung des Benutzers (über Passwort) erlauben darf.

## 11.4 HTML und Gestaltung von WWW-Dokumenten

### 11.4.1 Strukturierung und Darstellung von Inhalten

#### Trennung von Struktur und Inhalt

Dazu Meta-, Auszeichnungs- und Präsentationsprachen entwickelt und standardisiert.

Meta-Auszeichnungssprachen	SGML, XML
Auszeichnungssprachen	HTML, MathML, CML, VRML, ...
Meta-Präsentationssprachen	DSSSL, XSL
Präsentationssprachen	CSS1, CSS2, ...

#### Auszeichnungssprachen (AZS)

Dienen zur Strukturierung und Gliederung eines Dokuments.

*Meta-Auszeichnungssprachen:* zur Definition konkreter AZS

- Erster Standard: **SGML** (Standard Generalized Markup Language).  
Allgemeine ASZ, aber weniger für Auszeichnungen, eher grammatikalisches Regelwerk zur Definition von ASZ, sehr mächtig ~> ISO Standard 8879 (1986).
- i.w. alle konkreten AZS von SGML abgeleitet, u.a. das weit verbreitete HTML 2.0.
- Teilmenge als eigene, weniger komplexe Sprache: **XML** (eXtended Markup Language).  
*Konkrete Auszeichnungssprachen:* zur Strukturierung und Gliederung eines Dokuments.  
Bekannteste Auszeichnungssprache im WWW: **HTML** (HyperText Markup Language).  
Fester Dokumententyp: Anzahl der Elemente festgelegt und nicht erweiterbar.  
Erster Standard (1994): HTML 2.0 („kleinster gemeinsamer Nenner“)  
~> 3.0, 4.01, XHTML, HTML 5 (2009 als AZS festgelegt – keine XHTML-Variante !).  
Aber Grenzen der Leistungsfähigkeit, u.a. keine komplexen mathematischen oder chemischen Formeln gestaltbar. Deshalb auf Basis SGML bzw. XML durch W<sub>3</sub>C spezifische Auszeichnungssprachen definiert, wie
- **MathML** (Mathematical ML), **CML** (Chemical ML), **SMIL** (Synchronized Multimedia Integration Language) zur Synchronisation von Audio- und Videoströmen mit Texten, Bildern udgl.
- **WML** (Wireless Markup Language) für WAP-Formate (Strukturierung in Decks / Cards) auf PDA,
- **VRML** (Virtual Reality Modeling Language): Beschreibung interaktiver 3D-Objekte und Welten.
- **XHTML**: HTML-Erweiterung mit XML-Auszeichnungen (HTML-DTD -> XML-DTD).

### Präsentationssprachen

Dienen zur Darstellung abstrakter Inhalte.

*Meta-Präsentationssprachen:*

- DSSSL (Document Style Semantics and Specification Language) als „Mutter“ like SGML,
- und aus DSSSL abgeleitete vereinfachte Form XSL (Extensible Stylesheet Language).

*Konkrete Präsentationssprachen im WWW:*

Konzept der Style Sheets: sie bestimmen unabhängig vom Inhalt die Darstellung.

- Damit zentrale Einstellungen und Änderungen möglich sowie
- Dokument unterschiedlich darstellbar, z.B. Ausgabe auf Drucker, Bildschirm oder PDA, Sprachausgaben mit unterschiedlichen Stimmen usw.

Bekannteste Form im Zusammenhang mit HTML: CSS (Cascading Style Sheets).

Standards: CSS1 (i.d.R. angewandt), CSS2.

## 11.4.2 HTML (HyperText Markup Language)

### Entwicklung und Versionen

Sprache zur Gestaltung und Darstellung von Web-Seiten auf dem Browser. *Standard:*

SGML: Standard Generalized Markup Language (sehr mächtig, HTML als Untermenge)

ISO-Norm (ISO 8879, 1986) - Definition von strukturierten Dokumententypen.

Versionen: HTML 1.0 / 2.0 / 3.2 / 4.01 / XHTML / HTML5 (W<sub>3</sub>C und WHATWG)

Erweiterungen: XML, MathML, CML, WML (mobil), VRML (virtuelle Realität)...

*HTML – Versionen (Auswahl)*

- **HTML 2.0**: Nov. 1994 vom W<sub>3</sub>C-Konsortium als offizieller Sprachstandard verabschiedet von SGML abgeleitet, durch eigene DTD (Document Type Definition) festgelegt. Kleinsten gemeinsamer Nenner; unterstützt keine Frames und Tabellen.
- **HTML 4.01**: fehlerbereinigte Version von HTML 4.0, Dezember 1999. Sichert neben Frames auch Einbindung von CSS (Cascading Style Sheets) und von Script-Sprachen in HTML. Unterstützt fast alle heute weit verbreiteten Ergänzungs-Technologien für HTML.  
Alle Versionen ab HTML 2.0 / 3.0 / 4.0 basieren auf einer sog. HTML-DTD, abgeleitet von SGML.
- **XHTML** (eXtensible HyperText Markup Language): durch W<sub>3</sub>C, auf Basis XML. Nachfolge für HTML 4.01 ~> XHTML 1.0 / 1.1 / 2.0 (aber komplex und nicht abwärtskompatibel !). Mit der Meta-Auszeichnungssprache XML definiert, basiert auf einer XML-DTD.
- **HTML5**: W<sub>3</sub>C und Browser-Consortium WHATWG (Gründer: Ian Hickson / Google; kein SGML). Seit 2009 als nächste Generation für Web-AZS festgelegt (keine neue Variante von

XHTML !). Ersetzt die Dok.-Beschreibungsstandards HTML 4.01, XHTML 1.0 und DOM HTML Level 2. W<sub>3</sub>C-Recommendation für 2014 erwartet; bereits in meisten Browsern implementiert. Vokabular: vorangegangene HTML-Spezifikationen und XHTML-Elemente. Eigener HTML-Parser (nicht SGML).

### Charakterisierung HTML

HTML kennzeichnet

- Struktureigenschaften eines Dokuments, z.B. Absätze, Listen, Überschriften.
- Verweise auf andere Dokumente (sog. Links) im WWW.

Bildschirm-Layout und Übertragung von HTML-Dokumenten

- exakte Schriftarten- und größeren Einrückungen abhängig von der Browser-Einstellung.
- HTML-Dokumente mit eingebetteten Formatierungsanweisungen können wesentlich schneller übertragen werden als komplett formatierte Dokumente mit fertigem Layout.
- Unbekannte Elemente werden durch Browser ignoriert.

### HTML-DTD (Document Type Definition)

HTML ist als eine SGML-Anwendung definiert ~> zu Beginn des Dokuments ist eine eigene DTD festzulegen. Legt Regelwerk für die Syntax fest:

- Zeichen, Zeichenfolge, Schlüsselwörter, Markierungsbefehle, ...
- korrekte (strict) Struktur des HTML-Dokuments.

Alle HTML-Dokumente besitzen gleiche Struktur:

- !DOCTYPE (Document Type Declaration, nicht DTD!), am Beginn (für HTML 2.0 implizit enthalten), stellt Verbindung zum SGML her (bei Erstellung verwendete HTML-Version, Zuordnung zur Document Type Definition DTD)
- eigentliches HTML-Dokument (<HTML> ... </HTML>)

Beispiel für sauber (strict) programmiertes HTML-Dokument für HTML 4.01

```
<!DOCTYPE HTML "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<HTML> ... </HTML>
```

### XHTML (eXtensible HTML)

Bisherige HTML-Versionen erforderten eine neue SGML-basierte HTML-DTD, um die zulässigen HTML-Elemente festzulegen ~> dazu neue Browser-Version installieren. Mit XHTML wird die XML-Welt in den HTML-Standard transportiert: Aufbau wie HTML, Auszeichnungen aber durch XML-Elemente.

- XHTML-Befehlssatz kann erweitert werden, ohne neuen Browser zu benötigen.
- Dies möglich, da XML-fähige Browser die DTD nicht mehr fest im Programm integrieren, sondern die für das XHTML-Dokument gültige DTD wird geladen.

XHTML-Quellcode muss wohlgeformt und gültig sein, da sonst vom XML-Parser nicht korrekt bearbeitbar.

XHTML-DTD (Document Type Declaration)

- !DOCTYPE stellt Verbindung zum XML her (bei Erstellung verwendete XHTML-Version, Zuordnung zur Document Type Definition DTD)
- eigentliches HTML-Dokument (<HTML> ... </HTML>)

Beispiel:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<HTML> xmlns="http://www.w3.org/1999/xhtml">
<HEAD>
  <meta content="text/html; charset=UTF-8" http-equiv="Content-Type" />
  <title>Beispiel für xhtml-dtd</title>
  <meta content="Klaus Irmscher" name="author" />
```

```

    <meta content="index" http-equiv="robots" />
</HEAD>
<BODY>
    ... Text und Auszeichnungen des HTML-Dokuments
</BODY>
</HTML>

```

### Syntx von HTML

HTML-Elemente (Auszeichnungen, Tags) in spitzen Klammern gesetzt (sog. Spitzklammernkonvention). Beispiel Zeilenumbruch: <BR>

Für jedes HTML-Element ist in der HTML-DTD eine sog. „Element Type Declaration“ definiert: legt Start-Tag, Form des Inhalts und Ende-Tag fest:

```
<elementname> ... </elementname>
```

Ausnahmeregelungen: nur Start-Tag, z.B. <BR>, ohne Ende-Tag, z.B. Listenelement <li>

Ergänzung der HTML-Elemente durch Attribute: zusätzliche Informationen im HTML-Element, z.B. Einbinden von Bildern (Elementname IMG): wird dem Element als Attribut zugeordnet und ist innerhalb des Start-Tags zu definieren,

z.B. <IMG SRC="passbild.gif">. Auch mehrere Attribute zuordenbar.

### Aufbau der HTML-Dokumente

Vollständiges Dokument besteht aus

- Document Type Declaration ( !DOCTYPE, am Beginn, Zuordnung zur DTD)
- Eigentliches HTML-Dokument (zwischen <HTML> und </HTML>), aufgebaut aus
  - Kopfteil (Header): allgemeine Informationen zum Dokument (<HEAD> und </HEAD>)
    - \* Meta-Informationen zum Dokument, wie Titel, Autor, Dateipfade, ...
    - \* Schlüsselwörter, Sprachen, Style Sheets, Scripte, ...
  - Rumpfteil (Body): Inhalt des Dokuments (zwischen Tags <BODY> und </BODY>)
    - \* Eigentlicher Inhalt des Dokuments (Texte udgl.)
    - \* HTML-Auszeichnungen (sog. Tags): diese kennzeichnen Struktur und Formatierung eines Dokuments, Listen, Frames, Formulare, Verweise auf Dokumente (Hyperlinks) und eingefügte Medien (Bilder).

Der Body-Inhalt wird von Programmen für die Ausgabe interpretiert und entsprechend dargestellt (z.B. Browser).

Struktur eines vollständigen Dokuments:

```

<!DOCTYPE HTML "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<HTML>
  <HEAD>
    <TITLE>Grundstruktur eines HTML-Dokuments</TITLE>
    ... ggf. weitere HEAD-Elemente
  </HEAD>
  <BODY>
    ... Eigentlicher Inhalt des Dokuments
  </BODY>
</HTML>

```

HTML-Elemente können verschachtelt werden, allerdings nicht überschneidend, z.B. Überschrift mit Fettdruck und Kursivschrift

```

<h1>Eine <b>Überschrift <i>vom</i> Rang</b> eins</h1>
~> Eine Überschrift vom Rang eins (Ausgabe am Standard-Browser)

```

Sonderzeichen: < > & “ Umlaute und sonstige Sonderzeichen: werden außerhalb der spitzen Klammern durch ein &-Zeichen eingeleitet, danach das Sonderzeichen (als Dezimal- oder Hexadezimalzahl bzw. Unicode) und mit einem Semikolon abgeschlossen.

Beispiele:	Zeichen	im HTML-Quelltext	
	„“	&#8364	Anführungszeichen
	Ä	&Auml;	Umlaut
	©	&Copy;	Copyright

Beispiele von Elementen im Body-Teil (Auswahl)

- <H1> bis <H6> Überschriften (in 6 verschiedenen Größen)
- Anchors (HyperText-Referenzen zu anderen Dokumenten),  
z.B. <A HREF = "link" > Content </A>
- Integration von Grafiken (Icons oder kleine Grafiken)  
z.B. <IMG SRC = "dateiname" ALIGN = TOP | MIDDLE | BOTTOM

### Elemente im HTML-Dokument (Auswahl)

Überschriften und Umlaute

<Hi> Ueberschrift der Groesse i</Hi>i: 1, 2, 3, ...  
<Hi>&Uuml;berschrift der Gr&ouml;&szlig;e i</Hi>

Textformatierungen

<B>fetter Text</B>  
<I>kursiver Text</I>  
<PRE>unveraendert uebernommener Text</PRE>

Einseitige Gestaltungselemente (nur öffnende Tags)

Absatzmarke <P>  
Zeilenumbruch <BR>  
horizontale Linie <HR>

Listengestaltung (numeriert, unnumeriert, deskriptiv)

numerierte Liste	unnumerierte Liste	beschreibende Liste
<OL>	<UL>	<DL>
<LI>WWW 1. WWW	<LI>WWW .WWW	<DT>WWW WWW
<LI>HTML 2. HTML	<LI>HTML .HTML	<DD>HTML HTML
</OL>	</UL>	</DL>

Setzen einer Marke

<A NAME="Anfang">Textstelle Anfang </A> Setzen einer Marke Anfang

Verweise auf Textstelle im Dokument oder auf andere Dokumente (Anchors)

<A HREF="#Anfang">Textstelle Anfang</A>  
Verweis auf Textstelle *Anfang* innerhalb des Dokuments  
<A HREF="muster.html">Verweis auf muster.html</A>  
Verweis auf ein anderes lokales HTML-Dokument im gleichen Verzeichnis  
<A HREF="http://www.tu-freiberg.de/index.html">IfI-Homepage</A>  
Verweis auf ein HTML-Dokument auf einem anderen WWW-Server  
<A HREF="/u/prak0/prak11/Pacman.gif">Pacman</A>  
Verweis auf eine Grafik (.gif-Datei)

Integration einer Grafik in den Text (Image)

<IMG SRC="/u/prak0/prak11/Malpalette.gif"> Integration einer Grafik (.gif-Datei)

### Beispiel (HTML 2.0)

Dokument im HTML-Format

Ausgabe auf Bildschirm

```
<HTML>
<HEAD>
<TITLE>Muster-Datei muster.html</TITLE>
```

</HEAD>	
<BODY>	
<A NAME="Anfang">	
<H1>&Uuml;berschrift der Gr&ouml;&szlig;e 1</H1>	Überschrift der Größe 1
<H2>Textformatierungen</H2>	Textformatierungen
<B>Fetter Text</B>	<b>Fetter Text</b>
<I>Kursiver Text</I>	<i>Kursiver Text</i>
<PRE>Nicht von HTML formatierter Text</PRE>	Nicht von HTML formatierter Text
Anschlie&szlig;end folgt eine horizontale Linie	Anschließend folgt eine horizontale Linie
<HR>	<hr/>
	<Leerzeile>
Beispiel f&uuml;r eine nummerierte Liste:	Beispiel für eine nummerierte Liste:
<OL>	
<LI> Erstens	1. Erstens
<LI> Zweitens	2. Zweitens
</OL>	
<H3>Verweise</H3>	Verweise (in Größe H3)
Auf Textstelle innerhalb des Dokuments:	Auf Textstelle innerhalb des Dokuments:
<A HREF="#Anfang">Zum Anfang</A>	Zum Anfang
Auf lokales HTML-Dokument im gleichen Verzeichnis:	Auf lokales HTML-Dokument im gleichen Verzeichnis:
<A HREF="html-guide.html">Englischer HTML-Primer</A>	Englischer HTML-Primer
Auf HTML-Dokument auf entferntem Server:	Auf HTML-Dokument auf entferntem Server:
<A HREF="http://www.informatik.tu-freiberg.de/.../">Liste</A>	Liste
<P>	
Beispiel f&uuml;r eine in Text integrierte Grafik: 	Beispl. für eine in Text integrierte Grafik
<IMG SRC="/u/icon/Pacman.gif">	<Bild zu Pacman>
</BODY>	
</HTML>	

**Weitere HTML-Elemente (Auswahl)**

**Tabellen**

Vorgegebene Anzahl von Zeilen und Spalten bzw. Zellen. Einfachste Struktur: Zeilen werden mit dem Element TR ("table row") definiert, die Zellen mit dem Element TD ("table data") voneinander abgegrenzt.

Zelle: 1. Zeile, 1. Spalte	Zelle: 1. Zeile, 2. Spalte	Zelle: 1. Zeile, 3. Spalte
Zelle: 2. Zeile, 1. Spalte	Zelle: 2. Zeile, 2. Spalte	Zelle: 2. Zeile, 3. Spalte

```
<table border>
  <tr>
    <td>Zelle: 1. Zeile, 1. Spalte</td>
    <td>Zelle: 1. Zeile, 2. Spalte</td>
    <td>Zelle: 1. Zeile, 3. Spalte</td>
  </tr>
  <tr>
    <td>Zelle: 2. Zeile, 1. Spalte</td>
    <td>Zelle: 2. Zeile, 2. Spalte</td>
    <td>Zelle: 2. Zeile, 3. Spalte</td>
  </tr>
</table>
```

**Frames (Frameset)**

Damit mehrere Fenster zur gleichen Zeit in einem geteilten Browserfenster darstellen. Festlegung DTD: Frameset-Dokument muss statt Body-Element ein FRAMESET-Element besitzen.

Beispiel: 2 gleichgroße, nebeneinander angeordnete Fenster:

```
<!DOCTYPE HTML PUBLIC "-//W3C/DTD HTML 4.01 Frameset//EN
"http://www.w3.org/TR/html4/frameset.dtd">
<HTML>
  <HEAD>
    <TITLE> Beispieldatei eines Framesets</TITLE>
  </HEAD>
  <FRAMESET COLS=","">
    <FRAME SRC="Datei_Fenster1.html">
    <FRAME SRC="Datei_Fenster2.html">
    <NOFRAMES>
      Hier steht der Inhalt, der dargestellt wird, wenn ein
      Browser keine Frames darstellen kann.
    </NOFRAMES>
  </FRAMESET>
</HTML>
```

## Hyperlinks

Vernetzung von Dokumenten (Web, „Geflecht“). Im HTML 2 Typen unterstützt:

- im Body: sog. *Hypertextlink*. Stellt unidirektionale Verbindung von einem Quellanker zu einem Zielanker her. Beide Anker durch ein A-Element (Anchor) ausgezeichnet. Quellanker definiert im Attribut HREF das Ziel der Verbindung, Zielanker im Attribut NAME (oder ID) einen beliebigen, einmaligen Anker. Zielanker im gleichen Dokument, in anderen Dokumenten und auf anderen Rechnern (URI).

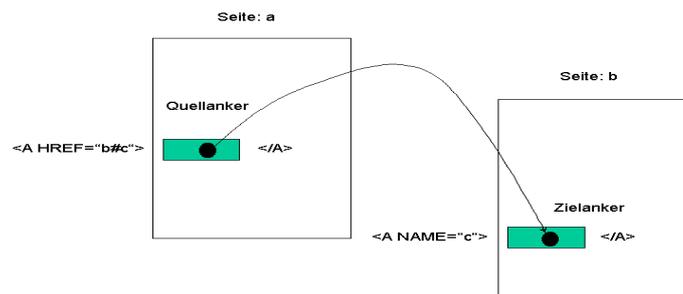


Abbildung 11.6: Hyperlinks im WWW

Bei Verwendung einer URI (Uniform Resource Identifier) kann man das Ziel auch definieren, ohne den Zielanker explizit zu benennen.

Beispiel: Nutzung Adressangabe und Email-Adresse aus der eigenen Homepage (Ziel)

```
<A HREF="http://www.informatik.uni-leipzig.de/~irmscher/index.html">
Prof. Dr. Klaus Irscher</A><BR>
Tel.: +49-341-97-32271<BR>
eMail: <A HREF="mailto:irmscher@informatik.uni-leipzig.de">
irmscher@informatik.uni-leipzig.de</A><BR>
...          Liefert Ausdruck:   Prof. Dr. Klaus Irscher
```

Tel.: +49-341-97-32271

eMail: irmscher@informatik.uni-leipzig.de

- im Header: Verknüpfung mit LINK-Element für ganze Dokumente. Beispiel Buch:

```
<HEAD>
  <TITLE>Kapitel 3</TITLE>
  <LINK REL="Index" HREF="../inhalt.html">
  <LINK REL="Next" HREF="Kapitel4.html">
  <LINK REL="Prev" HREF="Kapitel2.html">
</HEAD>
```

## Formulare

Formulare (eingeschlossen von Start- und Ende-Tag des Elements FORM) im HTML ermöglichen Interaktion zwischen Client (Browser) und Server.

Dazu werden verschiedene Formularfelder zur Eingabe bzw. Auswahl von Parametern (Element INPUT) und Steuerfelder (Element SUBMIT, sog. Klick-Button) bereitgestellt, wobei mehrere Formularfelder möglich sind:

Eingabefelder: Texteingaben (Textzeile)

Auswahlfelder: Auswahl (Auswahlliste, Radio Button oder Check Box)

Steuerfelder: Klick-Button (Submit: bei Drücken Aktion ausgelöst, z.B. Übergabe Daten an Server oder Starten eines Scripts; Reset: Rücksetzen der Eingabewerte)

Das entsprechende Programm wird mit seiner URL bestimmt, die im Attribut ACTION des Start-Tags des FORM-Elements anzugeben ist. Zusätzlich ist im Attribut METHOD die Datenübertragungsmethode festzulegen:

POST: Formulardaten mit HTTP (für Anwender unsichtbar) an die im ACTION-Attribut angegebene URI übertragen

GET: Formulardaten als Wertepaare (sichtbar) an die im ACTION-Attribut festgelegte URI mittels ? angehängt; mehrere Wertepaare (name=wert) durch & getrennt.

Datenübermittlung durch Drücken des Klick-Buttons SUBMIT ausgelöst.

Ab HTML 4.0 weitere Elemente, z.B. TEXTAREA-Element für Eingabefelder statt nur Zeilen, Attribut ACCESSKEY für Maussteuerung usw.

Die über die Formularfelder eingegebenen Werte können über eine CGI-Schnittstelle (CGI: Common Gateway Interface) an einen Server übertragen werden: für ein ausführbares Programm, ein Script (z.B. Perl) oder eine Email-Anwendung. CGI nicht unproblematisch ~> einfacher mit PHP, Python o.ä.

Serverseitig ist sicherzustellen, dass eine CGI- bzw. PHP-Implementation unterstützt ist. Die übertragenen Parameter sind auszuwerten und eine Bearbeitung auszulösen. Als Ergebnis wird eine HTML-Seite(n) an den Client zurückgeführt.

## Beispiel: Formulartechnik und CGI

Formular (FORM-Tag), CGI, Übergabe an Programm BESTELL, Methode POST

```

...
<FORM ACTION="http://www.servername.de/cgi-bin/BESTELL.EXE" METHOD=POST>
  <P>Informationsmaterial: <BR>
  <INPUT NAME="basis" TYPE=CHECKBOX> Basisinformationen
  <INPUT NAME="voll" TYPE=CHECKBOX> vollständige Informationen
  <P>Zahlung mit:
  <INPUT NAME="creditcard" TYPE=RADIO VALUE="Mastercard"> Mastercard
  <INPUT NAME="creditcard" TYPE=RADIO VALUE="Visa"> Visa
  <P>Kreditkartennummer:
  <INPUT NAME="creditcardnr" TYPE=TEXT SIZE=16 MAXLENGTH=16>
  <P>
  <INPUT TYPE="SUBMIT">
</FORM>
...

```

Bildschirmausgabe am Browser:

Informationsmaterial:  
 Basisinformationen     vollständige Informationen

Zahlung mit:    Mastercard    Visa

Kreditkartennummer:

Zugehörige PERL-Script-Datei (*bestell.pl*: Ergebnisanzeige der Formularwerte als HTML-Ausgabe für Browser): Wertübergabe durch form-Tag: name=wert&name=wert&... (Felder Addition, Multiplikation, Variable\_a, Variable\_b)

```
#!/usr/bin/perl
#read from standard input
read (STDIN, $Daten, $ENV{"CONTENT_LENGTH"});
#create and print HTML-document
print "Content-type: text/html\n\n";

print "<html><head><title>CGI-Feedback</title></head>\n";
print "<body><h1>CGI-Feedback vom Programm <i>rechner.pl</i></h1>\n\n";
print "Werte der Formulareingabe:\n";
@Formularfelder = split(/&/, $Daten); #Aufbrechen der Daten (&)
foreach $Feld (@Formularfelder) {
($name, $value) = split(/=/, $Feld); #Aufbrechen der Felder (=)
$value =~ tr/+// ;
$value =~ s/%([a-fA-F0-9][a-fA-F0-9])/pack("C", hex($1))/eg;
$value =~ s/<!--(.|\n)*-->//g;
$Formular{$name} = $name;
$Formular{$value} = $value;
print $Formular{$name} = $Formular{$value}, "<br>\n";
}
print "</body></html>\n";
```

### 11.4.3 Cascading Style Sheets (CSS)

#### Konzept der Style Sheets

HTML erlaubt Strukturierung und Gliederung von Inhalten, eine formatierte Darstellung ist aber nur begrenzt möglich. Um die Darstellung unabhängig vom Inhalt durchzuführen, wurde Konzept der Style Sheets entwickelt. Für WWW wurde die Präsentationssprache CSS (Cascading Style Sheets) entwickelt. Standardisierung durch W3C: CSS1 und CSS2

<http://www.w3.org/TR/REC-CSS1> bzw. [-CSS2](http://www.w3.org/TR/REC-CSS2)

Einem Dokument können mehrere Style Sheets gleichzeitig zugeordnet werden. Style Sheets unterstützen das Prinzip der Kaskadierung: sequentielle Auswertung der Vorgaben.

#### Definition und Syntax der CSS-Regeln

Ein Selector ist die Verbindung zwischen dem Style Sheet und dem betroffenen HTML-Element und bestimmt, für welches Element die Declaration gilt.

Syntax für CSS-Regeln: Selector: Selector [Universal Declarations]

Universal Declarations ~><http://www.un.org/Overview/rights.html>

Declaration unterteilt in 2 separate Teile: Eigenschaft und zugeordneter Wert

```
Element { Eigenschaft: Wert }
```

CSS-Standard definiert für jedes einzelne HTML-Element verschiedene Eigenschaften (CSS1 ca. 50, CSS2 ca. 120). Werte (in Abhängigkeit der Eigenschaften): Farbwerte, Längeneinheiten, Zeichenketten (z.B. URLs) oder in CSS definierte Schlüsselwörter.

Beispiel für eine Überschrift:

```
H1 { color: navy } -> Überschrift H1 in Farbe blau
```

In einer Regel können mehrere Selectors durch Kommata und mehrere Declarations durch Semikolon gruppiert werden. Beispiel für mehrere Überschriften (blau, Fettdruck):

```
H1, H2, H3 { color: navy ; font-weight : bold }
```

#### Darstellungsklassen

Die bisherige CSS-Syntax erlaubte nicht, die Darstellungsform im HTML-Dokument zu variieren, z.B. Absätze mit dem P-Element farbig unterschiedlich zu gestalten. Dafür zusätzliche Darstellungsklassen eingeführt, die spezifische Zuordnungen gestatten. Zugehörige Syntax:

Element.Klassenname { Eigenschaft : Wert }  
 bzw. ohne Zuordnung der Klasse zu einem bestimmten HTML-Element:  
 .Klassenname { Eigenschaft : Wert }

Beispiel für Zuordnungen:

```
<HTML> <HEAD>
  <TITLE>Klassen und CSS</TITLE>
  <STYLE TYPE="text/css">
    P {color:black}
    P.important {color:red}
    .marker {background:yellow}
  </STYLE>
  ...
</HEAD> <BODY>
  <H1 CLASS="marker">Eine mit gelb unterlegte Ueberschrift</H1>
  <P> Dies ist ein normaler Abschnitt in schwarz.
  <P CLASS="important"> Dieser Abschnitt ist wichtig, daher rot.
  <P CLASS="marker"> Dieser Abschnitt ist schwarz und gelb unterlegt.
  ...
</BODY> </HTML>
```

Die Klasse important kann hierbei nur für das P-Element genutzt werden; dagegen steht die Klasse marker allen Elementen zur Verfügung.

**CSS-Declarations (Auswahl)**

Häufigste CSS-Declaration: Farbeinstellungen durch die einfache Syntax bzw. Darstellungsklassen (siehe vorangegangene Beispiele), z.B.

```
color : black | red | navy | blue | yellow | green
background : yellow ...
```

Mit font-Befehl kann man auch alle anderen Eigenschaften der Textdarstellung definieren: Textgröße (absolut oder relativ), Schriftart (z.B. Times New Roman), Kusivschrift, Fettdruck, Sperrung, Unterstreichung, Umrahmung, Blinken usw. Bspw.

```
font-style : italic (kursiv)
font-weight : bold (fett)
```

**11.5 Interaktionen im WWW**

**11.5.1 Interaktive HTML-Seiten**

**Dynamische HTML-Seiten**

Ursprüngliches HTML ist statisch, nicht-interaktiv, synchron. Möglichkeiten, um HTML-Seiten interaktiv und damit dynamisch, benutzerabhängig zu gestalten:

*Clientseitig:* indem man Java-Code, dHTML-Seiten oder Skripte an den Browser schickt und auf dem Browser abarbeitet.

*Serverseitig:* dem Benutzer werden jedesmal vom Web-Server andere HTML-Seiten gesendet (CGI, ASP, JSP, PHP); Bearbeitung auf Server.

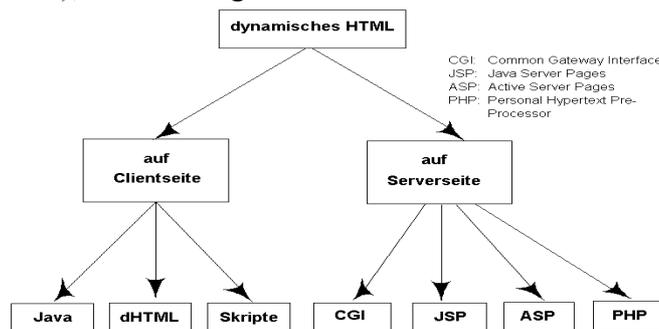


Abbildung 11.7: Interaktionen im WWW

Erweiterung im Web (insbes. für Web 2.0): Asynchrone Kommunikation (nicht-blockierend) zwischen Browser und Server durch Einsatz **AJAX** (Asynchronous JavaScript and XML), Nutzung **XMLHttpRequest-Objekt** im Web-Browser.

**Java:** Dem Browser wird ein komplettes Java-Programm geschickt, Ausführung im Browser (z.B. JVM: Java Virtual Machine). Beispiele: **Applets** (vorcompiliertes Java-Programm),

**dHTML:** HTML um dynamische Elemente erweitert. Durch Aktivitäten des Benutzers (Maus ziehen, anklicken) ändert sich das Aussehen der angezeigten Seite.

**Skripte** (VBScript, JScript, JavaScript): werden auf Client (Browser) ausgeführt (entsprechende Interpreter im Browser). Damit kann man mit Web-Server, der nur statische HTML-Seiten an Browser sendet, komplette Anwendungen gestalten.

**CGI:** Damit Informationen über CGI-Schnittstelle an Programme auf Server übergeben. Auf Server-Seite werden jedesmal neue HTML-Seiten generiert und zum Benutzer geschickt.

**JSP (Java Server Pages):** Es werden HTML-Seiten geschrieben, in die zusätzliche Programmieranweisungen eingefügt werden, z.B. Verzweigungen (IF), Schleifen usw. Hierbei auch sog. **Java Servlets** anwendbar (Programme/Prozeduren, die auf den Servern laufen).

**PHP** (Personal Hypertext Pre-Processor): Serverseitige Skriptsprache, die es erlaubt, dynamische HTML-Seiten zu entwickeln. PHP-Interpreter filtert PHP-Anweisungen serverseitig und führt die PHP-Zeilen aus, bevor die Übergabe an den Client erfolgt.

**Cookies:** Erlauben einem Server, Client-spezifische Informationen auf seinem Rechner zwischenspeichern.

## 11.5.2 Clientseitige Interaktionssteuerung (JavaScript, Applets)

### JavaScript

Proprietäre Skriptsprache, gegenüber Java/Java-Applets nur eingeschränkte Programmierfahrung notwendig. JavaScript-Anwendungen laufen auf Client-Seite. JavaScript ist eine plattformunabhängige Programmiersprache, objektbasiert (nicht objektorientiert). Anweisungen einzeln interpretiert (nicht kompiliert), um Maschinen-ausführbaren Code zu erzeugen. JavaScript-Anweisungen werden als Teil einer regulären HTML-Datei erzeugt. JavaScript-Code wird über den Script-Tag in HTML eingebunden (oder als externe Datei). Die Script-Tags werden vom Server vollständig zum Browser übertragen und dort zur Laufzeit von einem speziellen Browser-integrierten JavaScript-Interpreter interpretiert und ausgeführt.

*Beispiel* eines HTML-Dokuments mit JavaScript:

Ausgabe eines Strings Hallo, der in Anführungszeichen steht.

\* JavaScript steht zwischen den Anweisungen `<SCRIPT>` und `</SCRIPT>`

\* `<BR>` erzeugt eine neue Zeile

```
<HTML>
<HEAD>
  <TITLE>JavaScript-Beispiel</TITLE>
</HEAD>
<BODY>
  <SCRIPT LANGUAGE="JavaScript">
    <!-- Anzeige des Script ausschalten
    "document.write(<BR>");
    // Anzeige wieder einschalten ->
  </SCRIPT>
</BODY>
</HTML>
```

Mit JavaScript lässt sich ein statisches HTML-Dokument in eine dynamische Anwendung transformieren. Anwendung:

- Verarbeitung der Tastatur- und Mauseingaben (Ausführung auf Browser), bevor diese Formularwerte (z.B. über CGI) an Server weitergereicht werden.
- Kommunikation mit dem Server (Nutzung des XMLHttpRequest-Objekts im Browser).

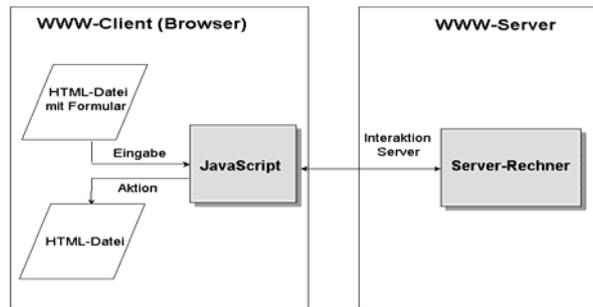


Abbildung 11.8: JavaScript im Web-Client/Server-Modell

Sicherheitslücken bei JavaScript, u.a. Programme können History-Datei, Cookies, URL-Caches lesen, Email-Adressen, Login-Namen und Passwörter (*Spoofing*) stehlen -> somit häufig von Anwendungen unterbunden.

### Java-Applets

Applets: vorkompilierte Java-Programme (-> Java-Klassenbibliothek, auf Server). Sie werden mit dem Applet-Tag ins HTML-Dokument eingebunden. Beim Zugriff wird das Applet vom Server des Anbieters zum lokalen Browser übertragen und dort eigenständig von einer Browser-integrierten Java Virtual Machine (JVM) ausgeführt. JVM ist integraler Bestandteil der heutigen Browser, wie Netscape (ab Version 2.0), MS Internet Explorer, Mozilla Firefox.

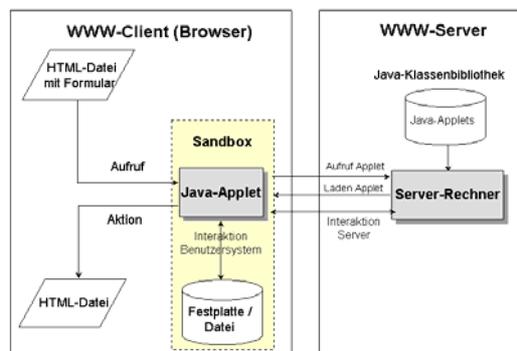


Abbildung 11.9: Java-Applet im Web-Client/Server-Modell

Die Ausführung erfolgt in einer „geschützten“ Umgebung (sog. Sandbox), über die mit anderen Anwendungen oder dem Dateisystem interagiert werden kann.

Applets werden vom HTML-Dokument getrennt gespeichert und bei Aufruf vom Server in die JVM des Browsers geladen. Sie unterliegen aus *Sicherheitsgründen* bestimmten Beschränkungen, u.a. keine lokalen Dateien öffnen, keine Anwendungen starten, Interaktion nur mit dem Server, von dem sie heruntergeladen wurden. Erweiterung: signierte Java-Applets (Zertifikat sichert die Integrität der Person).

Einfache Entwicklung eines *Java-Applets* und Einbettung in eine Web-Seite:

- \* Übersetzung des Quellcodes des Applets mit einem Compiler,
- \* Dadurch wird eine sog. CLASS - Datei erzeugt, Integration in Web-Seite.

*Beispiel:*

1. Erzeugung eines Applets mit Namen *Beispiel*. Übersetzung in Datei *Beispiel.class*.
2. Integration in eine Web-Seite in der Größe 240 und Breite 240

```
<HTML>
<HEAD>
```

```

<TITLE>Java-Beispiel</TITLE>
</HEAD>
<BODY>
  <APPLET CODE="Beispiel.class" WIDTH=240 HEIGHT=240>
</APPLET>
</BODY>
</HTML>

```

### 11.5.3 Common Gateway Interface (CGI)

#### CGI-Mechanismus für Server-seitige Interaktion

CGI (Common Gateway Interface): Möglichkeit zur Gestaltung von Interaktionen im Web (HTML). Erlaubt einem WWW-Browser, über einen WWW-Server Programme auszuführen. Diese Programme (sog. *CGI-Programme*) sind oftmals nur eine Schnittstelle zu anderen Programmen / Servern (u.a. Datenbanken, Perl-Scripts). Solche Programme (oder Scripts) können bspw. Formulareingaben aus HTML-Dateien verarbeiten, auf dem Server-Rechner Daten speichern und dort gespeicherte Daten auslesen.

Formulare (FORM-Tag im HTML) erzeugen Eingabefelder und Schaltflächen (Buttons). Mit den eingegebenen Daten bzw. Schaltaktionen wird auf Web-Server ein Programm gestartet, das die Daten analysiert und eine HTML-Ausgabe am Browser erzeugt.

Somit WWW-Seiten zu Oberflächen für "Anwendungen", z.B. E-Commerce: elektronische Warenbestellung (Teleshopping), Datenbankabfrage (Warehouse), Homebanking.

CGI-Programme stehen auf Server in einem speziellen Verzeichnis, i.allg. cgi-bin. Sie werden vom Client über die URL adressiert.

Parameterübergabe vom Client an das CGI-Programm bei Formulartechnik (Form-Tag):

1. Parameter an URL angehängt (nach ? werden die Parameter (Benutzereingaben) z.B. als Schlüsselwortpaare name=wert übergeben, Wertepaare durch & getrennt).
2. Über sog. Umgebungsvariable.

Ergebnisse des CGI-Programms werden als HTML-Dokument an den Client zurückgegeben.

CGI-Programme können nicht nur Formulareingaben aus HTML-Dateien verarbeiten. Auch Programmaufrufe aus dem HTML-Code (`href / img src`, z.B. Anklicken eines Bildes). Damit dynamische Informationen und Interaktivität erzeugbar. Unterstützung durch verschiedene Script-Sprachen (z.B. PERL, Modul CGI.pm). CGI nicht unproblematisch, u.a. schwerfällige Prozesse, Server mit CGI-Verwaltung, ...

#### CGI-Ablauf (Interaktion mit Formularen)

CGI realisiert eine serverseitige Interaktion, wobei Programme im Web bereitgestellt werden, die von HTML-Dateien aus aufgerufen werden können, und die selbst HTML-Code erzeugen und an einen WWW-Browser senden können.

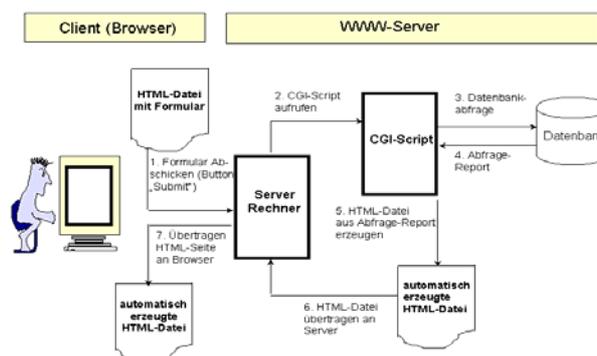


Abbildung 11.10: Ablaufgestaltung mit CGI-Script

## 11.5.4 Personal Hypertext Pre-Processor (PHP)

### Scriptsprache PHP

PHP (Personal Hypertext Pre-Processor) ist eine weit verbreitete Scriptsprache im Bereich der Internettechnologien. Es ist eine serverseitige Technologie, entwickelt für die Erstellung dynamischer HTML-Seiten.

Ähnlich C/C++, angelehnt an Java, C und Perl, aber einfacher zu programmieren. PHP erlaubt die prozedurale und objektorientierte Nutzung (~> leichter Einstieg für Programmierer von C und C++). PHP ist eine interpretierte Scriptsprache, die sowohl an der Kommandozeile als auch als Modul eines Webservers läuft ~> dadurch einfaches Debuggen möglich.

PHP setzt auf dem Web-Server eine PHP-Implementation voraus (z.B. Apache-Server). Das Aufrufschema ist analog wie bei CGI: über Formulartechnik oder direkten Aufruf. Allerdings ist PHP leistungsfähiger und weniger vom Systemadministrator abhängig. Somit Ablösung des CGI bei individueller Nutzung. Günstig für Realisierung von Web-Services.

Online-Quellen:

- Sebastian Bergmann: Professionelle Software-Entwicklung mit PHP 5 – Objektorientierung. Entwurfsmuster. Modellierung. Fortgeschrittene Datenbankprogrammierung. 1. Auflage, 2005. [http://www.professionelle-softwareentwicklung-mit-php5.de/erste\\_auflage/](http://www.professionelle-softwareentwicklung-mit-php5.de/erste_auflage/)
- Thomas Theis: PHP 4 – Webserver-Programmierung für Einsteiger. Galileo Computing, 1. Auflage, 2005. <http://www.galileocomputing.de/openbook/php4/>
- PHP Handbuch. <http://www.php.net/manual/de/>

### PHP Programmierung

PHP-Programme werden direkt in die HTML-Seite durch das SGML-Tag `<?php` und `?>` eingebettet. Dadurch keine separaten Script-Tags für PHP-Dateien erforderlich.

HTML-Dateien, die PHP-Code enthalten, werden nicht direkt an den anfordernden Client übertragen, sondern erst durch einen sog. PHP-Interpreter gefiltert. Dieser gibt das HTML unverändert an den Client weiter, führt aber zuvor die PHP-Programmzeilen aus (siehe Beispiel „Berechnung einer Summe“).

Mit PHP somit dynamische Dokumente erzeugbar, ohne den HTML-Code verändern zu müssen, z.B. Einfügen der aktuellen Uhrzeit, Datenbankeinträge usw.

Häufige Nutzungsform im HTML: Eingabe der Parameterwerte und Programmaufruf über Formulartechnik (FORM-Tag). Das aufgerufene PHP-Programm wird über die URL spezifiziert, Übertragungsmethode mit POST oder GET (analog zu CGI).

Beispiel „Berechnung einer Summe“:

auf Server: HTML-Datei mit PHP	HTML-Code, nachdem PHP-Interpreter den PHP-Code ausgeführt hat, und so vom Server an Client übermittelt wird
<pre>&lt;HTML&gt;&lt;HEAD&gt; &lt;TITLE&gt;PHP-Beispiel&lt;/TITLE&gt; &lt;/HEAD&gt;&lt;BODY&gt; hier steht normaler HTML-Code &lt;P&gt;Rechnungsergebnis: &lt;?php \$a=11; \$b=22; \$c=\$a+\$b; print \$c; ?&gt; ... &lt;/BODY&gt;&lt;/HTML&gt;</pre>	<pre>&lt;HTML&gt;&lt;HEAD &lt;TITLE&gt;PHP-Beispiel&lt;/TITLE&gt; &lt;/HEAD&gt;&lt;BODY&gt; hier steht normaler HTML-Code &lt;P&gt;Rechnungsergebnis: 33 ... &lt;/BODY&gt;&lt;/HTML&gt;</pre>

**Beispiel HTML-Formular und PHP-Script**

Auswertung eines HTML-Formulars (xHTML), Gestaltung mit Formularen (form-Tag).  
Web-Server mit PHP-Interpreter.

Auf Server sei ein öffentliches Verzeichnis (public\_html) eingerichtet für die HTML-Datei (index.html) und das PHP-Script (index.php). Dateinamen:

HTML: /home/irmscher/public\_html/pki/index.html

PHP: /home/irmscher/public\_html/pki/index.php

**HTML-Datei (index.html)**

```
<html>
<head><title>html und php</title></head>
<body>
  <form action="index.php" method="POST">
    Variable a: <input name="a" value="1" type="text"> <BR>
    Variable b: <input name="b" value="2" type="text"> <BR>
    <input type="submit">
  </form>
</body>
</html>
```

Variable a:	<input type="text" value="1"/>
Variable b:	<input type="text" value="2"/>
<input type="submit" value="Submit Query"/>	

**PHP-Datei (index.php)**

```
<?php header('Content-Type: text/html; charset=utf-8'); ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta content="text/html; charset=UTF-8" http-equiv="Content-Type" />
  <title>php-script</title>
  <meta content="index" http-equiv="robots" />
</head>
<body>
Werte der Formulareingabe und Ergebnis <br>
<?php
  echo $_POST['a'] . " + " . $_POST['b'] . " = " . ($_POST['a'] +
  $_POST['b']) . "<br>\n";
?>
<hr>
</body>
</html>
```

Werte der Formulareingabe und Ergebnis <a> + <b> = <summe> <hr/>
--

**PHP-Server**

Verschiedene Server unterstützen CGI, PHP u.a. Web-Technologien, wie Web-Services.

- Lizenzfreies Programmpaket für Testzwecke: xampplite. Enthält u.a. Apache-Server mit PHP der Version 5.x.

Dies verfügt auch über eine leistungsfähige SOAP-Implementierung. Im Verzeichnis freeservices werden Beispieldienste angeboten (über SOAP-Aufruf). Die Dienstbeschreibung erfolgt über WSDL und kann über diese Implementierung abgerufen werden.

- Handling: Dateien müssen im Verzeichnis hotdocs stehen.

xampplite

xampplite-control → Start Apache-Server (Port 80, läuft im Hintergrund)

xampplite-control → Stop und Exit (Beenden)

Dateien *index.html* und *index.php* im Verzeichnis *xampplite/hotdocs/pki/...*

Aufruf im Browser (z.B. Firefox): <http://localhost/pki/index.html>

### 11.5.5 Multimedia in HTML (Image Maps, Plug-ins)

#### Image Maps im HTML

Image Map: Sensitive Grafik, die in verschiedenen Bereichen des Bildes mit unterschiedlichen Aktionen (i.allg. Links) hinterlegt ist. Beispiel: Stadtplan ~> Auswahl der einzelnen Gebiete per Mausklick (jeweils einzelne Grafikdatei mit Ausschnittvergrößerungen angezeigt).

Mit IMG-SRC-Tag können Image Maps in HTML eingebettet werden (ab HTML 2.0).

Server-seitige Image Maps:

Funktionieren strukturell wie CGI-Anwendungen: Server liefert Image Map an Browser. Koordinaten des Mauszeigers (Benutzer-Input) werden als Argument von Browser an Server geliefert. Server bearbeitet Vorgang und liefert Output an Browser.

Client-seitige Image Maps:

Server liefert zur sensitiven Grafik direkt auch die Aktionsanweisungen (incl. zugehöriger Darstellungselemente) zu den einzelnen Koordinatenbereiche der Grafik mit, die dann Client-seitig abgearbeitet werden.

#### Plug-ins

Client-seitige Zusatzprogramme zum Browser. Dienen zur Präsentation von in WWW-Seiten eingebetteten Multimedia-Elementen, deren Formate nicht mit Browser-eigenen Funktionen handhabbar sind (Audio, Video, spezielle grafische Formate). Entwicklung von Netscape.

Die meisten Plug-ins sind über Multimedia-Anbieter frei erhältlich. Bei der Installation tragen sich die Plug-ins selbst in eine Liste ein. Diese signalisiert dem Browser, welches Zusatzprogramm bei welchem Multimedia-Format aufgerufen werden soll.

### 11.5.6 Verteilte Verarbeitung im WWW

#### Verteilte Anwendungen im Web

W<sub>3</sub>C: Entwicklung von RDF (Resource Description Framework) zur Definition und Standardisierung von Metadaten.

Basis: XML (eXtensible Markup Language), vermittelt die Informationen (content information) und bewertet gleichzeitig die Informationen (content rating). Erste Anwendungen:

1. EDI (Electronic Data Interchange)

Standard: EDIFACT (EDI For Administration, Commerce and Transport) -> EBxml. Datei-protokoll zum Austausch von Geschäftsdokumenten. EDI-Dateien enthalten umfangreiche Informationen über Art und Aufbau der Geschäftsdokumente.

2. Verschiedene Hersteller bieten XML-Applikationen für ODBMS an.

#### Neue Technologien für verteilte Web-Anwendungen

Aktuelle Technologien verteilter Web-Anwendungen

- JavaScript, PHP, HTML5 (ab 2009).

- AJAX (Asynchronous JavaScript and XML): asynchrone Kommunikation zw. Client und Server, Nutzung XmlHttpRequest-Objekt für Datenaustausch; s. auch Mono, Gecko.

- Web-Services (ab 2000): SOAP, WSDL, XML, UDDI.

Unterstützung durch SWE-Technologien:

- komponentenbasiert: J2EE/EJB, .NET/DCOM

- dienstorientiert: .NET/WCF (Indigo), SOA (Service Oriented Architecture).

Weitere Anwendungen im neuen Web (incl. SW-Tools, wie WordPress, Podcast, Ruby...)

- Web 2.0 („social Web“, ab 2004), u.a. YouTube, Flickr, RSS, Facebook, Twitter, ...

- Webtop-Technologien (Amazon S3 und EC2, MS Azure),

Neue Internet-Technologien, u.a. Cloud Computing (IBM, Sun u.a.).

- Speicher- und Verarbeitungskapazität im Internet („Wolke“), Zugriff über Browser.

## 11.6 Internet-Technologien (Web-Applikationen)

### 11.6.1 Web - Basis moderner Internet-Technologien

#### Technologie-Entwicklungen (Auswahl)

Konventionelles Web (Web 1.0): Informationssystem, HTML, XML (-> XHTML), CSS

Interaktionen: clientseitig (JavaScript, ...), serverseitig (CGI, PHP, ...)

Entwicklungen, z.T. außerhalb W<sub>3</sub>C, u.a. Web 2.0, HTML5 (WHATWG, W<sub>3</sub>C).

Neue Technologien im Web:

Weiterentwicklungen gemäß W<sub>3</sub>C, u.a. URI, XML, EMMA, SMIL, Voice XML

Web-Services (SOAP, ...), asynchrone Kommunikation (AJAX)

Semantisches Web (Web 3.0 ?)

Web 2.0 (~> Social Network): Blogs, Feeds, Applikationen (YouTube, Facebook, Twitter, ..).

Weitere Technologien (z.T. in Konkurrenz zu Technologien des SWE, wie .NET, EJB, ...):

SOA (Service Oriented Architecture)

Webtop (Amazon EC2, S3), Cloud Computing

Future Internet Portal

#### Aktuelle Auszeichnungssprachen

- HTML (HyperText Markup Language): Textbasierte Auszeichnungssprache für Dokumente im Web zur Strukturierung und Auszeichnung von Inhalten (Texte, Bilder, Hyperlinks).

Problem: sauberer HTML-Code auf den verschiedenen Browsern.

Hinzu Präsentationssprachen zur Darstellung der Inhalte: CSS (Cascading Style Sheet)

- W<sub>3</sub>C: auf Basis *SGML*: HTML 2.0<sub>1994</sub> (kleinster gemeinsamer Nenner) ... **HTML 4.01**<sub>1999</sub>  
(unterstützt nahezu alle Ergänzungs-Technologien für HTML, u.a. Frames, CSS).

auf Basis *XML* (strengere und leichter zu parsende Teilmenge von SGML):

**XHTML 1.0** (eXtensible HyperText Markup Language<sub>2000</sub>), XHTML 1.1 / 2.0.

- Browserhersteller-Consortium WHATWG (Gründer Ian Hickson / Google) zusammen mit W<sub>3</sub>C: **HTML5** (eigener HTML5-Parser, kein SGML)

Seit 2009 wird HTML5 als nächste Generation für Auszeichnungssprachen im Web festgelegt (keine neue Variante von XHTML !).

HTML5 ersetzt die Dokumentenbeschreibungsstandards HTML 4.01, XHTML 1.0 und DOM HTML Level 2.

#### HTML5

Textbasierte Auszeichnungssprache zur Strukturierung und semantischen Auszeichnung von Inhalten (Texte, Bilder, Hyperlinks) in Dokumenten [Quelle: Wikipedia]:

- Soll Nachfolge von HTML 4.01 werden. Sprache z.T. noch in Entwicklung <sub>2011</sub>.

- Bereits vorliegend: ausgereifte Entwürfe der Entwicklerteams WHATWG und W<sub>3</sub>C.

- Ersetzt die Beschreibungsstandards HTML 4.01, XHTML 1.0 und DOM HTML Level 2.

- Viele neue Funktionalitäten, u.a. für Video, Audio, lokale Speicher, dynamische 2D- und 3D-Grafiken, die zuvor nicht unterstützt wurden (erforderten zusätzliche Plugins wie Adobe Flash).

Verschiedene Arbeitsmodelle:

- WHATWG: Consortium von Browserherstellern (Gründer: **Ian Hickson** / Google).

~> versionsloses Modell für HTML, ohne „5“ („Living Standard“): sog. HTML-Standard.

- W<sub>3</sub>C: Spezifikation unter dem Namen HTML5 ~> vordefiniertes Prozedere bis zur W<sub>3</sub>C-Empfehlung (Recommendation): Ziel für vollständige Spezifikation: 2014.

Entstehung:

- HTML bis zum HTML 4.01 entwickelt, aber HTML seit 12/1997 nicht mehr wesentlich erweitert.

- W3C setzte auf XML, sollte zum Nachfolger von HTML werden (HTML 4.01 -> XHTML 1.0 mit Funktionsumfang von HTML 4.01) -> XHTML 1.1 -> XHTML 2.0 (aber Abkehr von HTML 4.01, kompliziert, keine Abwärtskompatibilität).
- Mitte 2004: WHATWG entgegnet diesem mit “*Web Applications 1.0*“ (1. Vorschlag für HTML5). Seit 2006 in W3C neue Arbeitsgruppe für neues HTML (auf Basis einer WHATWG-Version). Vorerst HTML5 und XHTML 2.0 als verwandte Sprachen mit unterschiedlichen Zielen bearbeitet. Mai 2007: W3C verwendet Web-Applications 1.0-Entwurf der WHATWG für neues HTML. Seitdem entwickeln W3C und WHATWG gemeinsam an einer HTML5-Spezifikation. 2009: W3C gibt bekannt, dass XHTML 2.0 ab Ende 2011 nicht mehr weiterentwickelt wird. Nächste Generation der Web-Auszeichnungssprachen ist HTML5 (keine neue XHTML-Variante).

Zeitplan für HTML5:

Ausruf „Last Call“-Status: WHATWG (Okt. 2009), W3C (Mai 2011); endgültig als Recommendation: 2014. In meisten Browsern ist HTML5 bereits implementiert <sup>2011</sup> (wenn auch unvollständig).

Anm.:

WHATWG-HTML-Spezifikation integriert mehrere Teilspezifikationen, die seitens W3C in einzelne Dokumente aufgesplittet werden:

HTML5-Hauptspezifikation, Microdata-Metadaten, 2D-Zeichenkontext des Canvas-Elements (Zeichnen 2-dimensionaler Formen) und dokumenteübergreifende Nachrichten (HTML5 Web Messaging).

Unterschiede zu HTML 4.01

- Dokumententypangabe in HTML5-Dokumenten: Zeichenkette `<!DOCTYPE html>` (keine Versionsangabe, Groß- und Kleinschreibung spielt keine Rolle).
  - Einfache Einbindung von SVG und MathML.
  - Neue Elemente, u.a. strukturierende, Gruppenbildung, Textauszeichnung, Multimedia (z.B. Einbindung Audio-/Video-Dateien, Hinzufügen einer Zeichenoberfläche (Canvas-Element) zum Zeichnen 2-dim.-Bilder in Scripten), Formulare (erweitertes input), interaktive Elemente, ...
  - Unterschied zu DOM Level 2 (DOM: Document Object Model), u.a. Drag & Drop, History
- Kritik an HTML5, u.a.  
zu sehr Browserhersteller-lastig, an WHATWG, alleiniges Schreibrecht für Ian Hickson, ...

*Aufbau HTML5* (nach W3C, bestehend aus mehreren Spezifikationen und Dokumenten)

- HTML5: Hauptspezifikation (wichtigsten Grundlagen und Schnittstellen für HTML) mit
  - Vokabular: Elemente vorangegangener HTML-Spezifikationen, Teile aus Ruby-Gruppen von XHTML 1.1 sowie ein Strukturmodell (Regelwerk).
  - HTML-Parser: erstmalig HTML nicht als SGML-Sprache, sondern eigene „SGML“-ähnliche Sprache -> dafür aber neuer HTML5-Parser erforderlich (rückwärtskompatibel).
  - Schnittstellen für HTML, XHTML und das DOM: jedes Element und Attribut von HTML5 ist im Data Object Model (DOM) definiert ~> somit sind auch die Schnittstellen zum DOM-Interface zu definieren.
  - Standarddarstellung (insbes. durch CSS-Eigenschaften).
  - Browserkontext: in jedem Browserkontext wird ein Dokument geladen bzw. weitere Browserkontexte erzeugt (z.B. History-Objekt: speichert Abfolge der besuchten Webseiten)
- HTML Microdata: Spezifikation für das Einbinden maschinenlesbarer Information in das HTML, z.B. andere Formate wie RDF und JSON.
- HTML-Canvas-2D-Kontext: Schnittstellen zum Zeichnen 2-dim. Formen. Zeichenoberfläche: Canvas-Element. Objekte: Linien, Schatten, Konturen, Texte, enthaltene Bilder.

- HTML5-Nachrichtenübermittlung: es definiert 2 Methoden, um den Datenaustausch zwischen 2 unabhängigen Browserkontexten zu erlauben:
  - *cross-document messaging*: Kommunikation von ineinander eingebetteten Dokumenten (z.B. über iframes),
  - *channel messaging*: Kommunikation für voneinander unabhängigen Dokumenten (z.B. getrennte Browserfenster).
- HTML+RDFa: Mechanismus zum Einbetten von RDF in HTML.

## 11.6.2 Technologien Web-basierter Anwendungen (u.a. AJAX)

### JavaScript

Scriptsprache (Interpreter), implementiert auf Web-Browsern. Ermöglicht clientseitigen Dialog zwischen Browser und Server. Dynamisch während Laufzeit interpretierte Programmiersprache, speziell für den Einsatz in Browsern konzipiert.

*Entwicklung:*

- Ausgangspunkt ECMAScript (ECMA: European Computer Manufacturers Association).
- Brian Eich (Netscape): LiveScript (1995).
- Umbenennung in JavaScript, zusammen mit Sun Microsystems (Syntax zwar ähnlich wie Java, aber sonst keine Übereinstimmung).
- Microsoft-Variante JScript (1996).

Anfangs große Unterschiede zwischen den verschiedenen Browser-Versionen. Mit Standard ECMA-262 wurde zusammen mit Netscape 1997 die Grundelemente einer Scriptsprache für Webbrowser definiert. 1998 zur ISO-Norm erhoben und auf allen Browsern umgesetzt.

JavaScript ermöglicht die Interaktionen mit dem Benutzer und kann Objekte innerhalb der Browser-Umgebung steuern. Anfangs nur für Fenster, Formulare und Bilder. Später Erweiterungen, z.B. kann über das Document Object Model (DOM) eine vom Server geladene HTML-Seite direkt im Browser modifiziert werden ~> damit Teile von Web-Seiten zu animieren, zu entfernen, zu ergänzen oder völlig neue Teile dynamisch zu generieren.

Außerdem: über XMLHttpRequest-Objekt (in allen gängigen Webbrowsern vorhanden) kann aus Skriptsprache heraus mit Web-Server kommuniziert werden (wichtig für AJAX).

Möglichkeiten zur Einbindung von JavaScript in Web-Seite: zwischen JavaScript-Tags im HTML-Quelltext / als HTML-Links / als Attribut eines beliebigen HTML-Tags / in externen Datei.

JavaScript mit Sicherheitsproblem gegenüber Angreifern. Zwar kein Schreiben auf Festplatte durch JavaScript unterstützt, aber damit möglich bspw. das Auslesen von installierten PlugIns, Simulieren von Benutzeraktionen (Submit, Mail, ...), Laden aktiver Inhalte (z.B. ActiveX). Führt i.allg. zum Öffnen vieler Fenster bis zum PC-Absturz. [Dagegen ernsthafte Gefahr bei JScript und VBScript (Microsoft) infolge der Funktion zum Schreiben auf lokalen Datenträger].

JavaScript nicht nur in Browsern. Es existieren auch eigenständige Implementierungen, z.B. für Mono/.NET (Novell) und Java ~> erlauben ein Skripting innerhalb von eigenständigen Anwendungen.

### AJAX (Asynchronous JavaScript and XML)

AJAX - Konzept zur asynchronen Kommunikation zwischen Webclient und Webserver. Damit Webseite aktualisierbar, ohne diese erneut vollständig abzurufen. Asynchrones Konzept erlaubt eine nicht-blockierende Arbeitsweise im Client.

Voraussetzungen einer solchen Web-Anwendung: JavaScript und die Möglichkeit, aus der Skriptsprache heraus mit dem Webserver zu kommunizieren. Dies bei allen gängigen Webbrowsern über das XMLHttpRequest-Objekt möglich.

AJAX ermöglicht, bisherige Technologien zusammen zu nutzen, wie HTML oder XHTML bzw. HTML5, Cascading Style Sheets (CSS), JavaScript, Document Object Model (DOM), XML, XSLT und das XMLHttpRequest-Objekt.

Jesse James Garrett (Adaptive Path, Febr. 2005): Artikel über neues Konzept „AJAX“ für Web-Anwendungen. AJAX fundamentales Konzept für moderne Web-Anwendungen. Es beinhaltet mehrere Technologien:

- standardgerechte Präsentation mit XHTML und CSS (bzw. HTML5),
- dynamisches Anzeigen und Interaktivität mittels DOM (Document Object Models),
- Datenaustausch und Datenmanipulation mittels XML und XSLT,
- asynchrone Datenabfrage unter Verwendung von XMLHttpRequest,
- JavaScript.

Das *klassische Web-Modell* arbeitet synchron: Benutzeraktionen am Browser lösen einen HTTP-Request zum Web-Server aus. Server bearbeitet die Anfrage und sendet HTML-Seite als Ergebnis an Browser zurück. Browser bleibt so lange blockiert.

Mit AJAX-Konzept wird eine sog. AJAX-Engine zwischen Browser und Server geschaltet. AJAX-Engine in JavaScript geschrieben und i.allg. in einem Frame versteckt. Aufgaben:

- Darstellung des Interfaces,
- Kommunikation mit dem Server,
- asynchrone Arbeitsweise (Interaktion mit User unabh. von Kommunikation mit Server).

*Web-Anwendungs-Modell mit AJAX* arbeitet asynchron:

statt HTTP-Request wird JavaScript-Aufruf an AJAX-Engine abgesetzt.

- falls kein Serveraufruf unbedingt erforderlich (z.B. Datenvalidierung, Datenbearbeitung im Client, einfache Navigation), realisiert die Engine die Anfrage selbst.
- falls Serverantwort erforderlich (z.B. DB-Abfrage, Nachladen weiterer Interface-Code), erfolgt dies asynchron mittels XML, ohne die Interaktion zum User zu beeinflussen: AJAX-Engine schickt HTTP-Request an Server weiter, dieser antwortet i.allg. mit XML data an die Engine, die das Ergebnis als HTML + CSS data an Browser leitet.

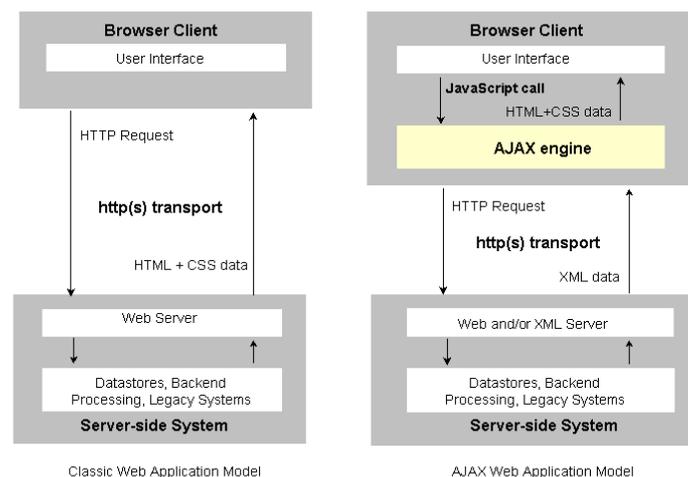


Abbildung 11.11: Modelle von Web-Anwendungen

*Klassische Web-Anwendung:*

Synchrone Kommunikation (blockierende Arbeitsweise). Client (Browser) stellt Anfrage an Web-Seite (über Link oder Absenden eines Formulars). Web-Server bearbeitet Anfrage und sendet Antwort als HTML-Seite zurück. Client solange blockiert.

*Web-Anwendung mit AJAX:*

Asynchrone Kommunikation (nicht-blockierende Arbeitsweise). Einsatz AJAX im Client: Erlaubt, asynchrone Anfragen an Server zu stellen. Client kann weiterarbeiten.

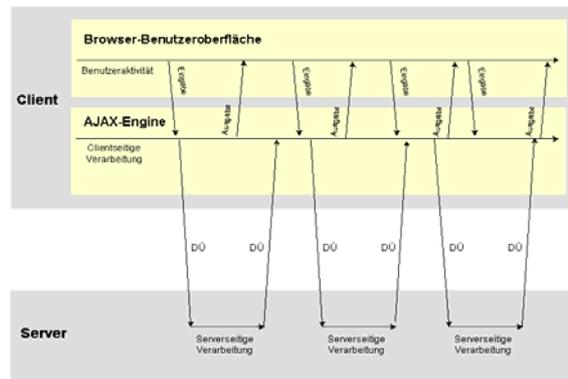


Abbildung 11.12: AJAX-Modell einer Web-Anwendung (asynchrone Kommunikation)

JavaScript sendet asynchrone Anfrage an Server. Für Erhalt der Antwort am Client wird auf diesem eine sog. Callback-Funktion eingerichtet. Um diese Antwort am Client darstellen zu können, ist eine Änderung in der Dokumentenstruktur der HTML-Seite erforderlich. Dies erfolgt durch JavaScript, CSS und DOM. Antwortformat des Servers ist bei AJAX nicht nur auf XML beschränkt, auch JSON (JavaScript Object Notation), Text oder HTML möglich.

*AJAX als Basistechnologie neuer Web-Anwendungen:*

- Server-Komponente: beliebiges Programm, unabh. ob synchrone o. asynchrone Anfrage
- Client-Komponente: Datenaustausch zwischen AJAX und Server erfolgt über das XMLHttpRequest Objekt von JavaScript (von allen gängigen Browsern unterstützt, aber unterschiedliche Implementationen).

Mit einer Instanziierung ist das XMLHttpRequest-Objekt (XHR) erstellt. Zur Realisierung einer Anfrage dient die Methode `open()` des XHR-Objekts. Zum Erstellen der Anfrage benötigt die Methode zwei Parameter: die http-Methode und die URL des Serverprogramms. Als Attribute für die http-Methode sind GET, POST oder HEAD zulässig. Zum Versenden der Anfrage an Server bietet das XMLHttpRequest-Objekt die Methode `send()` an. Ein Event-Handler informiert den Client, wann Server mit Bearbeitung der Anfrage fertig ist und eine Antwort bereit steht. Event-Handler ist meist eine Fkt., sog. Callback-Funktion. Das XHR-Objekt besitzt mit `onreadystatechange` eine solche Eigenschaft mit fünf Zuständen zum Status der Abarbeitung.

### **Resumé:**

AJAX durch Web 2.0 und die neuen Web-Applikationen für Web-Entwicklungen gepusht. Zahlreiche Frameworks für AJAX entwickelt, z.B. Microsoft, Mozilla, Sun Microsystems. ASP.NET Version 3.5 enthält standardmäßig AJAX, für Vorgängerversionen (wie ASP.NET Version 2.0) können die AJAX-Komponenten separat heruntergeladen werden. Da AJAX die JavaScript-Technologie enthält, gelten die gleichen Sicherheitsaspekte.

*AJAX-basierte Web-Anwendungen:*

- Orkut, Google Mail, Google Groups, Google Suggest, Google Maps.
- Flickr, A9.com (Suchdienst bei Amazon).
- SW von Adaptive Path.
- Social Network Implementationen, Tendenz zunehmend.

*Alternativen:*

Alternativ zu AJAX bietet die bei Mozilla.org entwickelte Gecko-Engine die Möglichkeit, per SOAP mit dem Webserver zu kommunizieren. SOAP: Protokoll zum Aufruf von in WSDL formulierten Web-Services.

[http://developer.mozilla.org/en/docs/SOAP\\_in\\_Gecko\\_based\\_Browsers](http://developer.mozilla.org/en/docs/SOAP_in_Gecko_based_Browsers)

### 11.6.3 Cloud Computing

#### Cloud IT

Cloud Computing (Synonym: Cloud IT, Rechnen in der Wolke).

Anwendungen und Daten nicht mehr auf lokalem Rechner gehalten, sondern auf einem Rechner im Internet gespeichert und ausgeführt (in der Wolke, „Cloud“):

- Verbindung zu den entfernten Systeme über ein Netzwerk (i.d.R. Internet),
- Zugriff auf die Anwendungen und Konfigurationen über einen Web-Browser.

Gesamte IT (Rechenzentrum ... Softwareentwicklung ... Business Intelligence) nicht mehr durch Anwender selbst bereitgestellt, sondern über einen oder mehrere Anbieter bezogen und bei Bedarf genutzt.

Somit neues Konzept im IT-Management: von Betrieb, Bereitstellung, Support, Management, Sourcing, Architektur bis zur Verwendung von IT. Online-Lexikon „IT Wissen“: „*Cloud Computing ist eine Strategie, bei der Hard- und Software von einem oder mehreren Providern bereitgestellt und auf Anforderung genutzt werden kann*“.

Nutzer können sich per Internet-Zugang die benötigten Ressourcen aus dem Netz holen: Betriebssystem, Speicherplatz, Prozessorleistung, spezielle Software. Damit Verzicht auf eigene Speicher- und Verarbeitungskapazität. Eigener PC nur noch spartanisch ausgestattetes Endgerät mit schneller Internet-Anbindung (Webtop). Typische Anwendung: Online-Speicherung von Fotos.

Cloud Computing: ein Schwerpunkt auf Computermesse CeBIT 2011 in Hannover („work and life with the cloud“). Branchenverband BITCOM prognostiziert Wachstum für IT-Markt (DE um 2% in 2011).

#### Neuer IT-Begriff

Begriff Cloud Computing (Prof. Ramnath K. Chellappa): Cloud Computing besteht aus

- Pool aus abstrahierter, hochskalierbarer und verwalteter IT-Infrastruktur,
- On-Demand-Infrastruktur (Rechner, Speicher, Netze) und On-Demand-Software (Betriebssystem, Anwendungen, Middleware, Entwicklungstools), dynamisch an die jeweiligen Geschäftsprozesse anpassbar.

Noch keine klare Definition, ggf. Aggregation verschiedener Ansätze wie Software as a Service (SaaS), Organic Computing, Island Computing und Konzepte (Visualisierung).

Vorteile: Ressourcen-Effizienz, Einsparungen, Verringerung Komplexität.

Nachteile: Datensicherheit (~> SSL/TLS-Verschlüsselung), Zugriff auf private Daten.

Nutzerverhalten noch unklar: Großkunde ... Privatnutzer. Gefahr einer „dot-com-Blase“ ?

#### Cloud IT - Architektur

3-Schichten-Architektur („bottom up“-Sichtweise):

- Infrastruktur („Cloud Foundation“): z.B. Elastic Compute Cloud (EC2) von Amazon, Go-Grid, Amazon S3, Linode. Der Service stellt eine Umgebung von virtuellen Servern zur Verfügung (Cloud Hosting). Vorteil: Skalierbarkeit, Anwender hat vollen Zugriff auf die virtuellen Server und kann Anwendungen selbst installieren.
- Plattform: Anwender erstellt die Anwendung und lädt sie auf dem Server. Dieser übernimmt Aufteilung auf die physischen Server (darauf keinen direkten Zugriff); z.B. Service App Engine von Google
- Anwendung: Anforderungen wie Design soll modular und serviceorientiert sein, Anwendung soll dynamisch, verteilt und mandantenfähig sein; z.B. Google Docs.

#### Cloud IT - Anwendungen

Cloud-Dienste:

- Als Teil der Cloud-IT-Infrastruktur („Cloud Foundation“), z.B. im Web  
Amazon EC2 (Elastic Compute Cloud): Mieten temporärer Verarbeitungskapazität auf

einem Server im Web,

Amazon S3 (Simple Storage Service): virtuelle Festplatte im Web.

Services stellen eine Umgebung von virtuellen Servern zur Verfügung (Cloud Hosting), Anwender mit vollem Zugriff.

- Online-Plattformen im Web 2.0:
  - Online Foto-Plattformen: Flickr (Yahoo!), Picasa (Google), Snapfish (HP)
  - Online-Video-Plattformen: YouTube (Google), Sevenload, Clipfish, ...

*Weiterhin bereits genutzt Cloud-Anwendungen:*

- Email-Dienste (Emails in abstrakter Serverumgebung gelagert, Zugriff vom PC oder Smartphone), u.a. Gmail, GMX.
- Speicherdienste, wie Dropbox, Mozy (bis zu 2 GByte kostenlos, unbegrenzt 5 € p.m.), Wuala, Dienst des Speicherherstellers LaCie (für 10 GB virtueller Speicher 19 € p.a.). Für Gäste auch Zugang mit eingeschränkten Benutzungsrechten einrichtbar, u.a. für Urlaubsbilder, Videos, MP3s.
- Online-Backup-Dienst, wie MyDrive (Schweiz): 2 GB kostenfrei, darüber gestaffelte K.

*Datenverluste/Sicherungen:*

- Regelmäßige Sicherungskopien, sichere Server und jederzeit zugänglich für autorisierte Benutzer: u.a. US-Firma Mozy: mehr als 80 Milliarden MByte, weltweit verteilt auf RZn.
- Verschlüsselung: Software Truecrypt (kostenfrei): empfehlenswert, insbes. für USA.

### **Microsoft Windows Azure**

Microsoft Windows Azure Platform: Cloud-Computing-Plattform von Microsoft. Vorrangig für SW-Entwickler; verfügbar seit 01.02.2010; wichtigster Dienst: Windows Azure.

Cloud Computing stellt Anwendungen und Datenbanken bereit, Nutzung ist netzbasiert.

Dateien des Anwenders ebenfalls auf Servern des Providers.

Microsoft konzentriert sich auf Internet-basierte Services:

- Käufertrend zu billigeren und leistungsschwächeren Rechnern (wie Netbooks),
- Eindämmung von Raubkopien (SW ist nicht mehr an Endnutzer zu vertreiben).

Ray Ozzie (Chief Software Architect, MS) kündigte 2008 in Los Angeles auf *Professional Developers Conference* Windows Azure als Strategiewechsel (Umgestaltung der SW) an. (zuvor Steve Palmer als Windows Cloud).

Dienste der Windows Azure Platform:

- Windows Azure (Hauptteil): netzbasierter Zugang zu Cloud Computing,
- neue Dienste, u.a. eine Datenbank und neue Version des .NET Frameworks,
- Dienste zur Synchronisation von Daten, die auf dem System der Microsoft Windows SharePoint Services basieren.

Zur Absicherung der Last werden 20 neue Rechenzentren (je 1 Mrd. Dollar) aus bis zu 2500 Servern (in Containern zusammengefasst) erstellt. Daten auf verschiedenen Servern redundant gespeichert (99,9 % Verfügbarkeit zugesichert).

Microsoft konkurriert damit gegen bereits etablierte Anbieter wie Google App Engine, Google Text & Tabellen, Iwork.com (Apple) oder Elastic Compute Cloud (Amazon).

### **Microsoft Windows 8**

Erste interne Details <sup>2011</sup> über das italienische Portal Windowsette (verstärkt Trend-Themen Apps und Cloud Computing).

Zentraler Punkt von Windows 8: *Cloud Computing* (Steve Palmer: „riskanteste Strategie“).

Alle Einstellungen der Benutzer künftig online gespeichert:

- damit bei technischen Problemen und Neuinstallationen alles einfach und sicher wiederherstellbar,
- persönliche Einstellungen der Nutzer bequem auf verschiedenen Geräte übertragbar und synchronisierbar.

Windows 8 auch für verschiedene Geräteklassen konzipiert, wie Tablets, Netbooks, Notebooks, All-in-Ones; somit Anmeldung und Registrierung eher Nutzer-abhängig (statt wie bisher Geräte-abhängig). Unterstützung Touch-Screen.

Bereitstellung: 2012 / 2013 (lt. CES 2011, Betaversion 2012).

„Project Thunderbay“ (Codename): Microsoft entwickelt einen eigenen AppStore für Windows 8, den sog. Windows Store (in Ergänzung zu den Handy-Apps für Windows Phone 7).

Eine One-Click-Recovery-Funktion ermöglicht, Windows neu zu installieren, aber persönliche Daten zu behalten (Nutzung Windows Store). Alle bereits installierten Apps eines Nutzers können so auf einmal neu installieren ~> Microsoft folgt dem Apple-Trend.

Windows 8 soll auch für energieärmere ARM-basierte Systeme geeignet sein (wie Windows 7). Damit für Smartphones und Tablets vorgesehen ~> Anpassung an Windows Phone 7.

#### 11.6.4 Future Internet Portal

Zusammenfassung verschiedener nationaler und internationaler Forschungsinitiativen für das Internet der Zukunft (sowohl Technik als auch Anwendung):

- insbes. bessere Unterstützung von Mobilität, Dienstgüte (QoS), Sicherheit.
  - Zielstellung global noch unklar: kompatibel zum existierenden Internet oder neu ?
- Experimentierplattformen, z.B. GENI-Projekt (USA), German-Lab (DE; Ende 2008).

Technisch orientierte Initiativen:

- FIND (Future Internet Network Design): Initiative des NSF (USA)
- Next Generation Service Delivery Platforms & Service Overlay Networks (DE, Wien)

Auch nicht-technische Aspekte (sozio-ökonomische, geschäftliche, ökologische Belange):

- FIA (Future Internet Assembly): europäische Initiativen für einen globalen Markt.

### 11.7 Web 2.0 (mediales Internet)

#### 11.7.1 Neues Netzverständnis im Web

##### Ursprung des Web 2.0

Internet als Paketvermittlungsnetz (IP) seit 1983 dominierend in USA, mit Entwicklung des Web (1992) auch in Europa (X.25 -> IP). Anwendungen: FTP, Email, WWW ...

Ursprüngliches Web statisch ~> 90er Jahre: multimedial und interaktiv. Ab 3. Jahrtausend Entwicklung einer neuen Gemeinschaftsform: das sog. „Social Network“ (kollaboratives Mitmachnetz, „neues Netzverständnis“) -> vor allem in den neuen TV-Medien.

Begriff „Web 2.0“ erstmalig Dez. 2003 von Eric Knorr (CIO Magazin) verbreitet: „Web als universelle, Standard-basierte Integrations-Plattform“. Auf WWW-Konferenz Okt. 2004 erstmalig Begriff des Web 2.0 geprägt (Craig Cline, Dale Dougherty: „Renaissance des Web“). Tim O'Reilly (2006): definiert Web 2.0 als *Veränderung in der Geschäftswelt und als neue Bewegung in der Computerindustrie, mit Internet als Plattform* – vs. Tim Berners-Lee.

Begriff des Web 2.0 bezieht sich auf

- Spezifische Technologien und auf eine veränderte Nutzung und Wahrnehmung des Internet.
- Benutzer erstellen / bearbeiten / verteilen Inhalte selbst, unterstützt von interaktiven Anwendungen. Inhalte nicht mehr nur von Medienunternehmen, sondern durch viele Nutzer erstellt, die sich untereinander mittels sozialer Software vernetzen ~> Social Networks.
- Im Marketing: vom Push-Prinzip („Stoßen“, aktive Verteilung, z.B. Blackberry-E-mails im Mobile Computing) zum Pull-Prinzip („Ziehen“, aktive Sammlung, z.B. Abonnementdienste).

##### Dynamisierung von HTML-Seiten

Web 1.0: statisch (Seiten lange unverändert im Netz). Wenige „Bearbeiter“, viele „Nutzer“.

Abhilfe: Entwicklung von

- Content-Management-Systeme und DB-Systeme: dynamische Verwaltung (Seiten von mehreren Nutzern bearbeitet und ausgetauscht);
- Web 2.0: führt zur veränderten Nutzung des Internet
  - \* Verschwinden der Trennung von lokaler und zentraler Datenhaltung / Anwendung, Module bei Bedarf geladen, Browser als Nutzerschnittstelle.
  - \* Nutzer: stellen eigene Beiträge auf Server, pflegen Weblogs, private Daten ins Netz.
  - \* Webinhalte verschiedener Dienste werden über offene Programmierschnittstellen zu neuen Diensten verbunden (Mashups).
  - \* Nutzer nehmen aktiv an der Verteilung der Informationen teil.

### **Schlüsselprinzipien** (nach O'Reilly und Battelle, lt. Wikipedia)

- Web als Plattform (statt des lokalen Rechners), Datengetriebene Anwendungen,
- Vernetzung, verstärkt durch „Architektur des Mitmachens“ (jeder kann mitwirken),
- Kombination von Komponenten verschiedener Entwickler,
- Verteilte gemeinsame Nutzung von Inhalten und technischen Diensten,
- Projekte immer im Beta-Stadium (Ende des klassischen SW-Lebenszyklus),
- Massenanwendungen.

### **Techniken** (z.T. bereits in 2. Hälfte der 1990er entwickelt)

- Datenformat XML; Auszeichnungssprachen XHTML -> seit 2009 HTML5.
- Abonnementsdienste (sog. Feeds): damit können Web-Seiten abonniert, Informationen zwischen Web-Seiten ausgetauscht werden. Inhalte automatisch heruntergeladen (Pull-Dienst) und durch geeignetes Programm angezeigt, z.B. Schlagzeilen. Formate: i.w. RSS o. Atom.
- Web-Services: über Web abrufbares SW-Angebot zur Erstellung verteilter Anwendungen im Web (SOAP, WSDL, UDDI, ...).
- AJAX (Asynchronous JavaScript and XML): asynchrone Kommunikation Client – Server (nicht-blockierend), Einsatz AJAX-Engine im Browser.
- Semantisches Web: Idee von Tim Berners-Lee. Technologieentwicklung durch Einsatz von Standards wie XML, RDF und OWL.  
Verwendung von FOAF und XFN zur Beschreibung sozialer Netzwerke, Entwicklung von Folksonomies (vereinfachte Form von Ontologien), Einsatz von Geotags und RDF-basierten Feeds, Wikis und Mikroformaten.  
RSS zählt zu den ersten Anwendungsgebieten des semantischen Webs.
- Weiterentwicklung zum Web 3.0 (?): Schwerpunkt Semantic Web (Semantik, Sprache, ... ).  
Einbeziehung der Beziehungen der Artikel, Inhalte, Autoren zueinander: inhaltlich, strukturell und kontextuell. Vorboten dieser neuen Welt:
  - Cloud Computing (Bereitstellung von Speicher- und Verarbeitungskapazität im Internet)
  - „web trend map“ des Beratungsunternehmens iA (information Architects).

### **Technische Struktur**

- Technische Basis: Web Tomorrow des W<sub>3</sub>C  
URI, HTTP 1.1, XML, XHTML, ..., HTML5 (W<sub>3</sub>C, WHATWG)  
Geräte- und Softwareunabhängigkeit (XForms, CSS / XSL, SVG / SMIL, Voice XML),  
Multimodalität (EMMA, Navigation), Web-Services (SOAP, WSDL, ...),  
AJAX (asynchrone Client/Server-Kommunikation), Einsatz AJAX-Engine,  
Semantisches Web (RDF, OWL).
- Generelle Problemstellungen:
  - Sicherung der Daten und Integrität, insbes. bei persönlichen Daten.
  - Kontrolle der Zugänge und der Inhalte.
  - Aktualität der Inhalte, Datenmüll, Akzeptanz.
 Letztlich geht es um reale Dinge („das Internet ist kein Paralleluniversum“).

## Neue Web 2.0-Anwendungen

Internet-Anwendungen, die direkt dem Begriff Web 2.0 zugeordnet werden können:

- Blogs (Weblogs): neues Homepage-Konzept („Blogosphäre, Tagebuch im Internet“): Autorenkreis verfasst Einträge, die chronologisch in umgekehrter Folge aufgelistet werden. Leser kann dazu Kommentare verfassen (Read/Write).
  - Wiki: Ansammlung von Webseiten, die von Benutzern frei erstellt und überarbeitet werden kann. Bekannteste Wikis: Wikipedia, WikiLeaks.
  - RSS-Feeds: Ticker auf der Webseite und Link auf Originalseite. Pull-Dienst (Abo).
  - YouTube, Flickr, Picasa, Facebook, Google+, VZnet (StudiVZ, SchülerVZ, MeinVZ), Xing: Informationen auf gemeinsamer Plattform bereitgestellt, gemeinsames Bearbeiten, auch mit Tagging (gemeinschaftliches Indexieren).
  - Twitter (Mini-Blogging), FriendFeed (Aggregator), Online-Messaging (für Smartphones, u.a. WhatsApp): Dienste zum Austausch von Kurznachrichten.
  - Podcast: Veröffentlichung von Audio- und Videodateien im Internet.
  - Soziale Netzwerke (Einsatz der Online-Plattformen zur „sozialen Vernetzung“): Nutzer können Profile erstellen, Kontakte verwalten und Nachrichten austauschen (i.allg. in Gruppen). Bekannte Beispiele (Bitkom 2010: ca. 30 Mio. Deutsche in einem SN):
    - \* Facebook, MySpace, VZnet (studiVZ, ...), Google's SN (Orkut, Buzz, Me, Google+), Facebook ca. 600<sub>2010</sub> / 800<sub>2011</sub> / 950<sub>2012</sub> Mio. TN weltweit; Google+: ca. 10<sub>2011</sub> / >100<sub>2012</sub> Mio. TN USA.
    - \* Jappy, wer-kennt-wen.de, StayFriends, Jamba, Kwick, Knuddels, Lokalisten,
    - \* beruflich: LinkedIn, Xing,
    - \* für Generation 50plus (Themen in Chats/Foren: Gesundheit, Reisen, Partnerschaften): www.feierabend.de, seniorentreff.de, platinnetz.de, freizeitfreunde.de, fitiesnet.de, 50plus-treff.de). Trend: steigend.
- Facebook und Google sind die größten Sammler privater Daten. Nutzung kostenlos, Finanzierung über Werbung (je ca. 3 Mrd. Web-Seiten-Zugriffe monatlich).
- Virtuelle Welten: dreidimensionale Plattform im Internet.
  - Social Bookmarks: Möglichkeit zur Speicherung und Kategorisg. von persönlichen Links.
  - Media Sharing Platforms: Möglichkeit zum Anlegen von Profilen, Speicherung von Mediendaten (Fotos, Videos) und Konsumieren/Bewerten von Inhalten anderer Nutzer.

Weitere neue Anwendungen im Internet / Web auf Basis der neuen Web-Technologien

- Google TV: Plattform für nahtloses Nutzen von Web-Diensten und TV-Sendungen (z.B. 2 Fenster: TV und YouTube-Videos/Twitter). Partner: Sony, Intel, Logitech. Betriebssystem Android, Web-Browser Chrome, TV-Gerät / Receiver, auch Smartphones integriert, spezifischer Chipsatz Intel, passende Peripherie-Geräte von Logitech. Google-TV soll in TV-Geräten, Blu-Ray-Playern und Settop-Boxen integriert werden. Erste Geräte: Sony/Bravia ( „Sony Internet TV“). Ähnliche Lösungen: Microsoft Windows Media Center, Apple TV (aber eher Flop).
- Internet-Telefonie: Technik VoIP (Voice-over-IP). SW-Programm wandelt Stimme in Datenpakete, ggf. auch mit Kamera. Anbieter: Skype, Jajah, Vyke, Google. Verstärkt internationale Gespräche. Skype: für PC, Notebooks, Smartphones; BS Windows, Linux o. MacOS (weltweit 370 Mio. registrierte Nutzer); Skype 5.0: Videotelefon-Konferenz (5 Tln.).

### 11.7.2 Social Networks (Websites, Online-Portale, Info-Dienste)

#### Mediales Internet

Mit den Technologien des Web 2.0 wurden neue Anwendungsfelder erschlossen, die vor allem durch die TV-Medien genutzt und jüngere Nutzerkreise inspirieren. Auswahl:

- Dynamische Webseiten
  - Read/Write-Web (Blog), Enzyklopädie (Wiki), Abonnementdienste (Pull-Verfahren, RSS).

- Online-Portale (Plattformen)  
Videos (YouTube), Fotos (Flickr, Picasa, Snapfish), personalisierte Webseiten (Facebook, Google+, VZnet, ...), soziale Vernetzung
- Nachrichtenorientierte Dienste  
Kurznachrichten, Mini-Blogging (Twitter), Aggregatordienste (FriendFeed), Instant Messaging (z.B. Online-Messaging für Smartphone-Apps, vs. SMS)

## ***Dynamische Webseiten***

### **Weblogs**

Weblog (abgekürzt Blog) - einfach einzurichtende, dynamische Webseite: Darauf veröffentlichen die Betreiber (Blogger) Inhalte in Journal-ähnlicher Form. Alle Einträge chronologisch geordnet (neuestes Datum an oberster Stelle).

Blogger nutzen dies zur Darstellung bestimmter (meist subjektiver) Ereignisse: persönliche, verschiedene Themenkreise bzw. Gruppendiskussionen (im Unternehmen). Inhalte können von anderen Nutzern kommentiert werden. Viele Blogger bieten dem Leser auch Beteiligungen am Schreiben an (Read-Write-Web).

Weblog-Software kann sowohl auf eigenem Webspace installiert (z.B. MovableTyp) oder von Application Service Providern als Dienst genutzt werden (z.B. Livejournal, Blogger). Gesamtheit aller Blogs: Blogospäre.

Unterschied zur statischen Web 1.0: starke Vernetzung und hohe Aktualisierungsfrequenz:

- Feeds informieren ständig über aktuelle Änderungen,
- mittels Trackback-Funktion können Daten zwischen 2 Blogs automatisch ausgetauscht werden, falls Artikel gegeneinander verweisen.

Weblog somit Werkzeug zur einfachen Publikation, dialogorientiert und zum Verweisen auf andere Seiten. Jeder Blögeintrag kann unter einer festen URL (sog. Permalink) aufgerufen werden. Weitergabe der Schlagworte über Blog-Suchmaschinen (z.B. Technorati), Kommentare, Pings und Trackbacks (Nutzung Blog-SW, wie Wordpress, Feeds und RSS).

### **Wiki**

Wiki (hawaiisch: wiki „schnell“, wikiwiki „sehr schnell“): Software oder Webseite, bei der jeder Benutzer Einträge verfassen oder ändern kann. Damit können auch verschiedene Autoren an einem gemeinsamen Artikel arbeiten.

Erstes Wiki vom SW-Autor Ward Cunningham (USA, 1995). Bekannteste Wikis:

Online-Enzyklopädie Wikipedia (Wikipedia Foundation, Jimmy Wales, 2001, San Francisco).

<http://de.wikipedia.org>

WikiLeaks (Island: anonyme geheime Dokumente, Julian Assange, 2006, z.B. Iraq War Diaries, u.a. Diary Dig, War Logs, Iraq War Logs)

<http://www.wikileaks.org> (speziell <http://cablegate.wikileaks.org> “Angela Teflon Merkel“), seit 03.12.2010 auf Servern in der Schweiz unter Hilfestellung der Piratenpartei (.ch).

Syntax Wiki einfacher als HTML. Syntax kann sich je nach Wiki ändern (über 100 verschiedene Wiki-Engines). Gefahr der Manipulation von Inhalten.

Als Pendant der Wikis (Web 2.0-Anwendung) gelten die CMS (Content-Management-Systems) der Web 1.0-Anwendungen: „*Ein CMS ist ein Anwendungsprogramm, das die gemeinschaftliche Erstellung und Bearbeitung des Inhalts von Text- und Multimediale-Dokumententen (Content) ermöglicht*“. Definition für Wiki und CMS zutreffend.

### **RSS-Feed**

RSS: Familie von Formaten für einfache und strukturierte Veröffentlichung von Änderungen auf Webseiten (z.B. News, Blogs, Audio-/Video-Logs). Entwicklung seit 1997 bzw. 1999 (Userland, Netscape) ~> Versionen “0.90“ ... “2.0“, standardisiertes plattform-unabhängiges Format (Basis XML).

RSS-Dienste in Form spezieller Service-Web-Seiten (sog. RSS-Channels) angeboten. RSS-Channel versorgt Empfänger (ähnlich Nachrichtenticker) mit kurzen Informationsblöcken: bestehend aus Schlagzeile mit kurzem Textanriss und Link zur Originalseite.

Bereitstellung von Daten im RSS-Format als RSS-Feed bezeichnet (to feed – füttern, einspeisen zuführen). Abonnementdienst: wenn Benutzer einen RSS-Channel abonniert hat, sucht Client in regelmäßigen Abständen beim Server nach Aktualisierungen im RSS-Feed (pull).

Bekannte Anwendungen: myNetscape.com und als Syndication-Format (u.a. BBC, CNN, Spiegel Online, tagesschau.de).

### ***Feeds („Bereitsteller“)***

Feeds (engl.: to feed – zuführen, versorgen): Dienst zur automatischen Bereitstellung von Webseiten-Inhalten (Abonnementdienst).

Um nach Änderungen nicht jedesmal aktiv in einzelnen Kategorien, Gruppen, Channels, Kommentaren oder persönlichen Streams von Webseiten suchen zu müssen, können von den Nutzern sog. Feeds („Bereitsteller“) abonniert werden.

Feeds als Pull-Dienst:

- Feeds stellen automatisch gewünschte Informationen bereit, falls Nutzer diese abonniert hat.
- Client stellt in regelmäßigen Abständen Anfragen zur Aktualisierung des Feeds an Server.
- Aktivität geht vom Client aus.
- Formate: RSS (verschiedene Versionen), Atom (Weiterentwicklung von RSS 2.0).

### ***RSS***

#### *Ticker auf Web-Seite*

- Service auf Webseiten, der Überschriften mit einem kurzen Textanriss (ähnlich Nachrichtenticker) und einen Link zur Originalseite enthält. Daten im RSS-Format bereitgestellt (sog. RSS-Feed): XML-basiertes Dateiformat. Falls RSS abonniert, werden automatisch die neuen Einträge geliefert (Pull-Verfahren).
- Nachdem RSS-Feed abonniert (durch Adresseintragung im Feed-Reader), kann Abonnent die Nachricht im Feed-Reader einlesen: direkt den angebotenen Links folgen und vollständige Meldung lesen.
- RSS als Pull-Dienst ausgeführt: Im Gegensatz zu Emails (Push-Dienst) geht bei RSS die Initiative vom Abonnenten aus, d.h. Anbieter kann Leser nicht auswählen, Abonnent muss sich aber selbst verwalten. RSS günstig zur Beobachtung großer Quellangaben (z.B. Blogs), falls diese aktualisiert werden.

#### *Funktionen*

- Lesen: Neue Feeds regelmäßig in bestimmten Abständen selbständig vom Feed-Reader auf die Endgeräte (PC, PDA, Mobiltelefon, ...) geladen. Zum Lesen eines RSS-Feeds dienen i.allg. Web-Browser, ggf. Email-Programme mit RSS-Funktionen und spezielle Programme (z.B. Bildschirmschoner).
- Syndication: RSS für maschinelle Weiterverarbeitung geeignet, z.B. in Web-Seite integrieren oder auf verschiedenen Endgeräten darstellen. Damit Webseiten automatisch aktualisierbar, ohne Aktion des Seitenbetreibers (Unterstützung durch einige CMS).

#### *Entwicklung:*

- RSS vor allem durch Weblog eingeführt (Autoren mit RSS oder Nutzung Weblog-Systeme wie WordPress oder MovableType, die RSS automatisch erzeugen), vorrangig Text.
- Mittels Podcast RSS auch für Audio und Video genutzt, z.B. für tragbare Abspielgeräte.

#### *Versionen von RSS (und Bezeichnung):*

Rich Site Summary (RSS-Versionen 0.9x): seit 1999, basierend auf RDF, My Netscape Network.

RDF Site Summary (RSS-Versionen 0.9 und 1.0): seit 2000, auch RDF, UserLand Software.

Really Simple Syndication (RSS-Version 2.0): seit 2002, Weiterentwicklung, UserLand Software.

Atom: Konkurrenz zu RSS; XML-Format (Weiterentwicklung von RSS 2.0). Nicht kompatibel zu RSS, jedoch umwandelbar.

### **Anwendung RSS-Feeds**

News in Web-Seiten oder Blogs können mit RSS-Feeds abonniert werden, um sie dann auf dem Rechner oder Smartphone zu lesen. Symbol in Adresszeile des Browsers zeigt an, dass eine Web-Seite einen RSS-Feed anbietet. Die Nachrichten können dann direkt im Browser, in einem speziellen Newsreader-Programm oder mithilfe eines online-basierten Dienstes gelesen werden. Anklicken des Logos liefert einen Link zum Feed. Typische Beiträge von Nachrichtenseiten (wie tagesschau.de, sz-online, Spiegel Online) oder Blogs.

Man kann sich aber auch über neue Ergebnisse in der „News“-Suche von Google informieren lassen ~> persönliche Online-Zeitung. Weitere Anwendungen auf Web 2.0-Plattformen wie Facebook oder MySpace. Nutzer von flickr können bspw. Fotos zu einem bestimmten Thema per Feed abrufen. Auch eBay informiert per RSS über neue Angebote.

Abonnieren von RSS-Feeds kann ohne zusätzliches Programm durch den Browser oder einige Email-Programme erfolgen. Bspw. haben Firefox und Thunderbird eine integrierte Funktion, für andere Programme gibt es Erweiterungen. Die integrierten Funktionen haben aber i.allg. einen verringerten Funktionsumfang (Firefox: nur Anzeige einer gekürzten Überschrift; vollständiger Text erst nach Anklicken der Schlagzeile und Weiterleitung über Link).

Mehr Komfort bieten spezielle RSS-Programme (i.d.R. für Profi-Anwender), u.a.

- Feedreader und Feedbuster,
- Mobiltelefone mit RSS-Lesefunktionen,
- Google: listet für registrierte Nutzer die zuvor ausgewählten Feeds übersichtlich auf. Man kann auch alle Feeds durchsuchen. Da der Google-Reader das Nutzerverhalten beobachtet, werden die Feeds geordnet angeboten.
- Bloglines: ordnet Newsfeeds online (weit verbreitet).

Weiteren Komfort bieten sog. personalisierte Startseiten. Damit können sich die Nutzer eine persönliche Seite einrichten und diese mit gesonderten Inhalten und Anwendungen („Widgets“) ihren Bedürfnissen anpassen, u.a. iGoogle, Netvibes und Pageflakes.

### **Online-Portale (Plattformen)**

#### **YouTube**

Plattform für Online-Videos (YouTube.com). Damit kann Nutzer ein eigenes Profil anlegen, Videos einstellen bzw. hochladen. Inhalte von YouTube auch auf eigener Webseite, per Email oder auf mobilen Endgeräten bereitstellbar. Plattform Febr. 2005 gegründet (Gründer Chad Husley), im Okt. 2006 von Google gekauft ~> Video-Plattform von Google. Pro Tag mehrere Millionen Zugriffe. (<http://www.youtube.com/t/about>).

#### **Flickr**

Plattform für Online-Fotos (flickr.com). Eigene Profile, Kommentare, Vernetzungen udgl.; zusätzlich Geotags (Lokalisierung Aufnahmeort).

Entwicklung durch Ludicorp, z.T. kostenpflichtig. Aufkauf durch Yahoo März 2005 (Übernahme auf dem flickr-Blog). Über 3 Mio. registrierte Nutzer.

[http://blog.flickr.com/flickrblog/2005/03/yahoo\\_actually\\_.html](http://blog.flickr.com/flickrblog/2005/03/yahoo_actually_.html)

Merkmale beider Plattformen (YouTube und Flickr): starke Vernetzung der Nutzer und deren Inhalte untereinander. Anlegen eigene Profile, Kommentieren der Fotos und Videos ~> soziale Vernetzung. Mittels Tags können einzelne Inhalte mit Schlagworten gekennzeichnet werden ~> Erleichterung der Suche.

Flickr bietet weitere Form der Tags, sog. Geotags: Damit möglich, den Nutzern auf einer Landkarte zu zeigen, wo das jeweilige Foto oder Video aufgenommen wurde.

#### **Weitere Online-Video-Plattformen**

Konkurrenz zu YouTube, insbes. für Musikvideos und als interaktives Musikfernsehen (oftmals Clips länger verfügbar als bei YouTube):

- Myvideo (Bukarest), Sevenload (multimediale Web 2.0-Plattform, Köln),
- Qtom (Berlin), Tape.tv (Berlin), Putpat (Köln),
- MTV Networks Germany (Berlin, [www.mtv.de](http://www.mtv.de)),
- Clipfish (große Plattform, auch über Google erreichbar),
- Veoh.com, Vimeo.com.

#### **Weitere Online-Foto-Plattformen**

- Picasa: Plattform für Online-Fotos bei Google
- Snapfish: Plattform für Online-Fotos bei HP

#### **Facebook**

- Facebook („Buch der Gesichter“): Web-Seite zur Bildung und Unterhaltung sozialer Netzwerke (Internet-Portal). <http://www.facebook.com>. Gründer: Mark Zuckerberg (Harvard Uni/Mass., 2004; Psychologie/Informatik, jüngster Milliardär).  
Ursprung: Facemash (Gesichtsbrei), 2003: Fotos von Kommilitonen  
Plattform „Harvard Connection“ von Cameron und Tyler Winklevoss  
4. Febr. 2004: Start seiner PF „thefacebook“ als SN  
Fa. Facebook Inc, Palo Alto (California, „Casa Facebook“), 1400 Mitarbeiter.
- Facebook ist eines der größten sozialen Netzwerke:
  - Vernetzen von Personen, Kontakte, Brieffreundschaften, Poesiealben, Aushänge udgl.
  - Herzstück: Profile seiner Nutzer (~ digitale Visitenkarte): Name und weitere Infos.
- Äquivalente soziale NW: studiVZ, meinVZ, MySpace, Google+, Xing, Lokalisten.
- Plattform: Ursprünglich für Studenten in USA, später auch für ausländische Hochschulen (seit 2006 frei für Deutsche, seit 2008 auch deutsche Seiten). Mai 2007 Plattform für Drittanbieter geöffnet (Bereitstellung einer Programmier-SS, API).  
Angebote: Spiele (u.a. Farmville), Kommunikation (Chat, Email), Spaßprogramme, Nintendo DSi, „Like“-Button („Gefällt-mir“ – ursprünglich intern, nun für sog. „Promis“ wie Lady Gaga, Barack Obama udgl.). Auch Suchergebnisse aus Web (über Suchmaschine Bing).
- Facebook ist ein aggressives und für die Privatsphäre gefährliches Programm zum Datensammeln. Facebook und Google sind größten Sammler privater Daten.
- Finanzierung über Werbung (ca. 3 Mrd. Web-Seiten-Zugriffe pro Monat), Umsatz >1 Mrd.\$ 2010, Nutzung ist kostenfrei.
- Buchverfilmung „The Social Network“, Okt. 2010 im Kino, mit Jesse Eisenberg.

Wie funktioniert Facebook:

- Browser ([www.facebook.com](http://www.facebook.com)), Startseite, registrieren (Name, Email, PW, ...),
- Profil anlegen: Angaben zur Person (alles freiwillig), u.a. Schule, Beruf, Arbeitsstelle,
- Foto hochladen,
- Freunde finden: Suche im Email-Postfach, Zuneigung anzeigen („anstupsen“), als Freund erklären (falls er zustimmt).

Über Aktivitäten der Facebook-Freunde wird Nutzer ständig informiert: wer als Freund hinzugefügt ist, neue Bilder und Kommentare usw. Neben diesen Grundfunktionen viele Programme integriert, z.B. Onlinespiele (z.B. Bauernhof-Simulation „Farmville“, „Mafia Wars“), Weltkarte (Markieren von Reisezielen). Zusammen mit Yahoo auch Twitteranzeigen auf der Facebook-Seite.

Connect: Mit Funktion *Facebook Connect* können Nutzer des sozialen NW ihre Anmelde-daten auch auf anderen Web-Seiten verwenden, ohne sich registrieren zu müssen.

Benutzer auf Facebook eingeteilt in Universitäts-, Schul-, Arbeitsplatz- und Regions-NW.

Funktionen (lt. Wikipedia):

- Pro Benutzer eine Profilseite zur Vorstellung und zum Hochladen von Fotos / Videos.
- Auf einer Pinnwand des Profils können Besucher öffentlich sichtbare Nachrichten hinterlassen oder Notizen/Blogs veröffentlichen.

- Zusätzlich können sich Benutzer persönliche Nachrichten schicken oder chatten.
  - Freunde können zu Gruppen oder Events eingeladen werden.
  - Auf Facebook-Marktplatz können Kleinanzeigen aufgegeben und eingesehen werden.
  - Eine Beobachtungsliste informiert über Neuigkeiten, z.B. Pinnwandeinträge.
- Seit 2009 ist FriendFeed integriert (Echtzeitdienst für Kurzbotschaften, Twitter-Prinzip).  
10. Aug. 2009 von Facebook gekauft. Erlaubt, Internet-Nutzer in Echtzeit zu verfolgen.

#### Statistik:

- Juli 2009: > 250 Millionen aktive Nutzer, Angebote in 20 Sprachen. Monatlich 1 Mrd. Bilder und 10 Mio. Videos hochgeladen. Dez. 2009: 350 Mio. Nutzer, davon 5,6 Mio. Deutsche (expon. Wachsen: ca. 600 Mio. Ende 2010, 750 Mio. Mitte 2011, 800 Mio. Ende 2011 Nutzer).
- Durchschnittlicher Nutzer: täglich ca. 1 h Nutzung, 130 Freunde, 8 Freundschaftsanfragen pro Monat; ca. 50% der Nutzer unter 26 Jahre, 15% über 40 Jahre alt; Tendenz steigend.

#### Probleme mit Facebook (Auswahl):

- Gefahr der unerlaubten Vermarktung der Profile, mangelnder Datenschutz,
- Nichtmitglieder erscheinen bei Facebook (~> Auflistung bei [www.tweakpc.de](http://www.tweakpc.de)):  
Facebook durchforstet Email-Adressbücher, Adressbuchsynchronisation (automatisches Abgleichen iPhone-Kontakte mit Facebook-Freundeslisten),
- Facebook lässt sich austricksen (falsch adressierte „Freunde“).

#### Facebook mit Ortungsdienst:

- Facebook kann Freund orten:  
Mit dem Geodienst Places kann man seinen aktuellen Aufenthaltsort melden. Facebook-Mitglieder per Handy bei Places melden (freiwillig). Für ihre Freunde im soz. NW werden sie dann als Punkte auf einer Karte sichtbar. Nutzung z.B. im Smartphone FB.
- Ortungsfunktion seit 05.10.2010 auch in DE verfügbar. Erforderlich: iPhone/Apple mit Facebook-App oder anderes mobiles Gerät mit Standard HTML 5 und Geotargeting (Möglichkeit, aus Adressen in Computernetzen geografische Angaben zu ermitteln).
- lt. Facebook: es sind „keine Bewegungsprofile“. Standard-Einstellung nur für Freunde, Nutzer müssen erst aktiv einwilligen (Heise Online: nicht sicher, Zusatzeinstellungen!).
- Äquivalente Dienste, Vorgänger: Foursquare, Gowalla, Google Latitude (aber mit fortlaufender Lokalisierung).
- Günstige Anwendung: Local Based Services (Konkurrenten: Google, Nokia).

#### Google+

- Online-Portal und Social Network wie Facebook (Konkurrenz !). Start: 28. Juni 2011.  
Finanzierung durch Werbung (Anzeigen-Erlöse).
- Funktionen analog wie bei Facebook:
  - für eine Anmeldung benötigt man aber *noch* (?) die Einladung eines Freundes,
  - bessere Verwaltung des Freundeskreises,
  - Nutzung des Google Suchdienstes und Email-Dienstes GMail (dem hat Facebook einen eigenen Dienst entgegengestellt: Adresse mit Endung @fb.com).
- Statistik (lt. Comscore): erstaunliches Wachstum
  - jeden Tag 1 Mio. neue Nutzer (in DE 920 000)
  - 25 Mio. Nutzer 08/2011, davon 6,4 Mio. in USA, in DE 920 000
  - für die Nutzerzahlen benötigte mySpace 22 Monate, Twitter 33 Monate, FB 7 Jahre.

#### VZnet

VZnet Netzwerke Ltd.: Web-basiertes Social Network. Deutsches Unternehmen, seit 25. Aug. 2009, aus studiVZ Ltd. Hervorgegangen, seit 2007 zur Verlagsgruppe Georg von Holtzbrinck. Sitz: Berlin (Leitung: Markus Berger-dé Leon); über 15,5 Mio. Mitglieder (Aug. 2009).

Zugehörige Plattformen:

- studiVZ.net: 11. Nov. 2005, Zielgruppe Studenten
- schülerVZ.net: 28. Febr. 2007, Zielgruppe Schüler
- meinVZ.net: 28. Febr. 2008, allgemeine Zielgruppe

**studiVZ** (Studenten-Verzeichnis, <http://www.studivz.net/>):

- ähnlich US-Web-2.0-Plattform Facebook. Sprache: Deutsch (Engl., Franz. usw. ab 2009 eingestellt). Über 6 Mio. registrierte Nutzer (2008).
- StartUp, Gründung: Nov. 2005 (Dennis Bemman, Ehssan Darianni) als Private Limited Company by Shares (Ltd.). Hauptsitz Birmingham, Zweigniederlassung Berlin.
- studiVZ 2007 in GmbH gewandelt. Übernahme 2007 durch Holtzbrinck Networks.

**schülerVZ** (Schüler-Verzeichnis, <http://www.schuelervz.net/>):

- Für Schüler ab 12 Jahre, ca. 5,5 Mio. Kinder, Sprache Deutsch. Aus jugendschutzrechtlichen Gründen keine Verbindung mit den beiden anderen Plattformen.

**meinVZ** (allgemeines Verzeichnis, <http://www.meinvz.net/>):

- Plattformen für Nichtstudenten oder Absolventen, Sprachen Deutsch und Englisch.
- ca. 4 Mio. Nutzer registriert.
- studiVZ- und meinVZ-Plattform über Schnittstelle verbunden, so dass auf Profile von beiden Seiten aus zugegriffen werden kann.

*Nutzung:* Kostenlos, neuerdings mit Werbung.

*Technik:* 750 Server (Linux-Systeme), zunehmend. Verwendung von PHP, MySQL-Datenbanken, Memcache-Server, Apache-Webserver. Hosting erfolgt in einem Rechenzentrum der Telefónica Deutschland.

*Funktionen:* Anlegen von Profilen, Erfassen von Daten, Suche, Anzeige von Verbindungen (Kontakte), Bildung von Gruppen mit Gruppen-Diskussionsforen, Erstellen von Fotoalben, Foto-Tagging, Gruscheln (Kontaktaufnahme), Melden Regelverstöße. Ignorieren, Plauderkasten (Chat-Dienst), Video-Serie „Pietshow“: Buschfunk (Twitter-ähnlicher Dienst, Nachrichten max. 140 Zeichen auf Startseite aller „Freunde“ angezeigt).

**StayFriends**

- Online-Plattform StayFriends: Web-Seite zum Finden und Verwalten von Schulfreunden. Führender Suchdienst für Schulfreunde in Deutschland (2009: über 9 Mio. Einträge, ca. 3 Mio. Nutzer/Monat). Basisdienst kostenlos, voller Funktionsumfang kostenpflichtige Gold-Mitgliedschaft.
- StayFriends GmbH, Sitz Erlangen. Gründung 2002 (Michel Lindenberg). Web-Seite: <http://www.stayfriends.de>. Seit 2004 als Tochtergesellschaft von Classmates Online (Renton, USA), dies wiederum Tochter von United Online (USA). Schwesterseiten in Schweden, Frankreich, Österreich, Schweiz.
- Funktionen:
  - Profil-Eintrag mit Namen, Schule, Abschlussjahr, Fotos.
  - Suchanfragen zu Schulfreunden.
  - Nachrichtenaustausch (für Mitglieder), Anzeigen von Fotos und Profil-Informationen.
  - Anzeige Poesiealben (Gästebuch) und eigener Seite, abh. von Goldmitgliedschaft.

**Xing**

- Webbasierte Plattform für natürliche Personen für vorrangig geschäftliche (aber auch private) Kontakte zu anderen Personen.
- Kernfunktion der Web-Seite: Sichtbarmachen des Kontaktnetzes, z.B. kann Nutzer abfragen, über „wie viele Ecken“ (über welche andere Mitglieder) er einen anderen kennt. Dez. 2009: 9 Mio. Nutzer in 200 Ländern, davon ca. 3 Mio. Deutsche. Basisdienst (Profil mit Gästebuch, Kontakte verwalten) kostenlos, Premium-Mitgliedschaft kostenpflichtig (Nachrichtenaustausch, Jobsuche udgl.).

- Xing AG, Sitz Hamburg. Gründung 2003 (Lars Hinrichs und Bill Liao). Bis Ende 2006: openBC (open Business Club). Web-Seite: <http://www.xing.com>.
- Funktionen:
  - Kernfunktion: Sichtbarmachung des Kontaktnetzes.
  - Community-Funktionen: wie Kontaktseite, Suche nach Interessengebieten, Foren, Unternehmens-Webseiten und 26.000 Fachgruppen.
- Xing hat eigene Jobbörse (Gefahr durch Headhunters in NWn !)

### **LinkedIn**

- Webbasiertes soziales Netzwerk zur Pflege bestehender Geschäftskontakte und zum Knüpfen neuer Verbindungen.
- Weltweit größtes Business-NW für internationale Kontakte. Dez. 2009: 53 Mio. Nutzer in 200 Ländern, ca. 500 000 Mitglieder in Deutschland. Zählt zu den 500 weltweit meistbesuchten Websites.
- LinkedIn Inc., Sitz Mountain View, CA. Gründung Mai 2003 (Reid Hoffman). Web-Seite: <http://www.linkedin.com>
- Funktionen: Verlinkung auf eine eigene Webseite, Angabe Lebenslauf, Knüpfen neuer Kontakte, Möglichkeiten andere Mitglieder zu empfehlen.

### ***Nachrichtenorientierte Dienste***

#### **Twitter („Zwitschern“)**

Twitter (*to tweet - zwitschern*): Mikro-Blogging-Dienst und kostenlose Plattform zum Publizieren von Kurznachrichten (seit 2006). Auch als soziales NW oder öffentlich einsehbares Tagebuch im Internet definiert.

Angemeldete Nutzer können eigene Textnachrichten (max. 140 Zeichen, sog. „Updates“) eingeben, die allen Nutzern angezeigt werden, die diesem Nutzer folgen. Herausgeber der Nachricht steht auf der Webseite des Dienstes mit einer Abbildung als alleiniger Autor.

Das Mikro-Blog bildet für Autor und Leser ein einfach zu bedienendes Echtzeit-Medium zur Darstellung spezifischer Themen. Kommentare oder Diskussionen der Leser zum Beitrag sind möglich. Nutzung des Medium zum Austausch von Informationen sowie auch zur Kommunikation.

Das *soziale Netzwerk* beruht darauf, dass man Nachrichten anderer Benutzer folgt (*engl. „following“*), d.h. Updates anderer Benutzer abonnieren kann.

- Nachrichten (Beiträge): „Tweets“ (*to tweet – zwitschern*) oder „Updates“ (Text, max. 140 Zeichen).
- Autoren: „Twitterer“ (seltener „Tweeps“); „twittern“: Tätigkeit des Schreibens auf Twitter. Absender kann entscheiden, ob Beitrag allen oder nur einer Gruppe zur Verfügung steht. „ReTweet“: referenziertes Wiederholen (forwarding) eines Beitrages einer anderen Person, z.B. für schnelle Weiterverbreitung als Eilmeldung.
- Leser („Follower“: *to follow - folgen*) sind Benutzer, die die Beiträge eines Autors abonniert haben. Die Beiträge der Personen, denen man folgt, werden in einem Log, einer abwärts chronologisch sortierten Liste von Einträgen, dargestellt. „Timeline“ (kurz „TL“): Liste der abonnierten Beiträge.

Twitter Inc.: Kapitalgesellschaft, gegründet 2006. Sitz: San Francisco, CA. Entwicklung:

- Jack Dorsey, Biz Stone, Evan Williams (2006): Podcast-Firma Odeo (San Francisco).
- Seit 2007 eigenständige Firma: Twitter Inc. (Sitz: San Francisco, CA).
- 2008: Akquisition Fa. Summize: ermöglicht über den Twitter-XMPP-Stream die Twitter-Konversation in Echtzeit

Twitter-Startseite (Web): <http://twitter.com>

- Anmeldung (Benutzername, Passwort),

Eintragen als Follower (Leser), dann können Nachrichten der Autoren empfangen werden.

Falls selbst Autor: Profil mit Bild und Selbstdarstellung (max. 140 Zeichen). Nachricht wird dann gleichzeitig an alle Follower gesendet.

- Eingabe von Updates (Eingabefenster, 140 Zeichen). Absender kann entscheiden, ob Updates allen verfügbar oder auf Gruppe beschränkt ist.
- Updates der Personen sehen, denen man folgt (sortiert nach Zeit, als TL).

#### *Benutzer-Kommunikation:*

- Eingabehilfen über Twitter-Homepage (RSS) oder Desktop-Software, Handy (falls mobiler Internet-Zugang) bzw. SMS (nur noch USA, Kanada, Indien).
- Mittels Erweiterungen können zusätzliche Informationen zum Absender oder zur Empfängergruppe angezeigt werden, z.B. jeweiliger Standort auf Google Maps.

#### *Technologie:*

Twitter setzt zur Erzeugung von Webseiten Programmiersprachen wie RoR oder Scala ein. Twitter ursprünglich in **Ruby on Rails** („Rails“ / „RoR“) geschrieben: Quelloffenes Web Application Framework zur SW-Entwicklung. D.H. Hansson, PS Ruby (Juli 2004), Nutzung Web-Services, XML-RPC, REST, Webserver WEBrick, Apache oder CGI ... Wegen Skalierungsproblemen in Infrastruktur des Backends seit 2009 in der PS **Scala** neu implementiert.

#### *Programmierschnittstelle (API):*

Twitter-Dienst auch über eine Programmierschnittstelle (**API**) verfügbar. Über Komplementoren werden Twitter-Nachrichten zur Verfügung gestellt, so dass Updates auf verschiedenen Kanälen von vielen Diensten abgerufen und wieder eingespeist werden können.

#### *Komplementoren:*

Twitter-API erlaubt Integration von Twitter in andere Web-Dienste / Anwendungen, u.a. in

- spezialisierten Clients, wie Seismic Desktop,
- Kundenbeziehungsmanagement Dienst Salesforce.com,
- Instant-Messaging-Client Diensten Adium, Digsby oder im Flock Browser.

T-Mobile (USA) hat Twitter (neben Facebook) in sein Mobiltelefon Sidekick integriert.

Spezialisierte Clients, wie TweetDeck, können Nachrichten übersichtlicher darstellen.

#### *Ausfallsicherheit:*

Stabilitätsprobleme -> Ausfälle insbes.2008. Anzeige Ausfall-Wal („fail whale“: *too many tweets!*). Zeitweise Entfernung von Funktionen, z.B. freies SMS, Instant Messaging ab 2008.

#### *Prominente Anwendungen von Twitter, u.a.*

- Cisco, Sun Microsystems, JetBlue Airways: Bereitstellung Produkt-/Dienstleistungs-Informationen.
- Präsidentschaftswahlen Barack Obama (2008): Wahlkämpfer Nader/Gonzales, in Echtzeit.
- Georgia Institut of Technology, Univ. of Texas at San Antonio: Infos an Studenten.
- US-Airways-Flug 1549 (15.01.09): Notlandung Hudson River, Erstmeldung über Twitter.

#### *Weitere Versionen:*

Japanische Version: mit englischer Benutzer-OF, mit Werbung.

Deutsche Version: bleeper.de.

#### *Twitter-Nutzung*

Schreiben der Beiträge normalerweise über Twitter-Startseite: über Eingabefenster, max. 140 Zeichen (Text). Um Zeichen zu sparen, werden i.allg. Links als Kurz-URLs angegeben.

Zahlreiche Drittanbieter ermöglichen

- Twittern über Desktop-Applikationen, Mobiltelefon oder Email.
- Beiträge mit mehr als 140 Zeichen: restlicher Text über weiterführenden Link angezeigt.
- Einbinden von Bildern und Bildunterschriften.

Statistik: - weltweit: ca. 45 Mio. Nutzer (2009).

- DE: 1,8 Mio Nutzer (Juni 2009), durchschnittliches Alter: 32 Jahre (Comscore).

### *Twitter und andere Medien bzw. Portale*

- Google News bringt „Updates“ auf der amerikanischen Google-Seite, analog Technorati.
- Yahoo integriert Twitter-Kurzmitteilungen in sein Such-Portal, ohne Web-Seite verlassen zu müssen (2010, vorerst nur USA). Analoge Verknüpfung auch Yahoo mit Facebook.
- Microsoft liefert für Suchmaschine Bing eine Echtzeit-Suche für sog. Tweets (2010).
- Apps für Twitter-Nutzung auf verschiedenen Smartphones<sub>2011</sub> (z.B. HTC, iPhone).

### **FriendFeed**

Echtzeit-Feed-Aggregator zum Zusammenlegen/Verdichten von Updates aus social Media und social Network Websites, social Bookmarking Websites, Blogs und Micro-Blogging sowie beliebigen weiteren Typen von RSS/Atom-Feeds. Auch möglich, diesen Informationsstream in benutzerdefinierte Feeds zu teilen bzw. für Diskussionen (Posts) zu nutzen.

- Firmensitz: Mountain View (California); Übernahme durch Facebook (10. Aug. 2009).  
Gründer: ehemalige Mitarbeiter von Google Inc. (insbes. aus Google Mail und Google Maps), u.a. Paul Buchheim, Jim Norris, Sanjeev Sing und Bret Taylor.
- Entwicklung eines einfachen Update-Protokolls zur Verringerung der Aggregator-Last.  
Zielstellung: verbesserter Zugang zu Web-Inhalten durch Nutzung existierender sozialer Netzwerke und Tools zum Entdecken von Informationen.

Nutzer: Einzelpersonen, Unternehmen oder Org. 2009 ca. 1 Mio. Nutzer (Twitter 45 Mio.).  
Kritik durch andere Medien: Nutzer lesen FriendFeed statt Original-Blogs.

### **Online Messaging-Dienst für Smartphone-Apps**

Mit Hilfe von Apps bieten Apple, Google u.a. Nachrichtendienste zum Austausch von Kurznachrichten an (als Instant Messaging, äußerlich wirksam wie SMS, adressiert über Tel.-Verz.), die sich die Kunden auf ihr Smartphone laden können.

Viele Online-Messaging-Dienste sind i.d.R. kostenlos, wie WhatsApp, iMessage, Tango. Solche Nachrichtendienste auch bei Social Networks wie Facebook oder Twitter.

Es sind Web-Dienste (Instant Messaging, Austausch von Texten, Push-Dienst, Client/Server): vom Browser werden die Nachrichten per sendmail-Prozess auf einen Server gespeichert und von dort mittels push-Prinzip an die Empfänger verteilt.

Damit große Konkurrenz für die klassischen Kurznachrichtendienste SMS der Telekom (Nutzung Signalisierkanal des Zellularfunknetzes, 160 Zeichen; speichervermittelt). SMS vs. WhatsApp, ... viele Nutzer abgesprungen, großer Umsatzverlust (über 9 %<sub>2011</sub>).

Hoffnung der Mobilfunkbetreiber wie Deutsche Telekom, Vodafone und O2 ruhen auf **RCS-e** (Rich Communication Suite enhanced). Start zur MWC 2012 (GSMA: Joyn).

Technisch komplizierte Lösung: intelligente Kombination aus WhatsApp und FaceTime.

Kunden können damit sowohl Textnachrichten als auch Fotos und Videos versenden. Während eines Telefonats kann eine Video-Verbindung zwischen den TN aufgebaut werden (unabhängig von Betriebssystem und Betreiber).

**RCS-e (Rich Communication Suite enhanced)**, kommerziell „Joyn“.

Industriestandard für drahtlose Übertragung von Texten, Sprache, Bildern, Videos (plattform- und provider-unabhängig). Entwicklung anfangs Nokia<sub>2008</sub>, dann Branchenverband GSMA, Start zur MWC 2012. SMS-Nachfolger, damit auch Videotelefonie und Textchats. Nutzung: Eingebunden in die Tarifstruktur und teilweise kostenlos.

Für Betriebssysteme von Microsoft (Windows Mobile/Phone), Apple (iOS) und Google (Android). Zunächst als App geplant, später in die Betriebssysteme integriert.

### **WhatsApp**

Proprietäre Instant-Messaging-Anwendung für Smartphones zum Austausch von Kurznachrichten (pull-Dienst). Verwendung Extensible Messaging and Presence Protocol.

WhatsApp Inc., Firmensitz: Santa Clara/CA, gegründet 2009 (Brian Acton, Jan Koum)

Betriebssysteme: Apple iOS, Google Android, BlackBerry OS, Windows Phone 7

Zusätzliche Funktionen:

- Austausch von Foto-, Kontakt-, Video- und Audiodateien
  - Mitteilung eigener Standort (mittels GPS),
  - Synchronisation mit Adressbuch des Telefons (separates Adressbuch nicht erforderlich).
- Statistik: täglich mehr als 10 Milliarden WhatsApp 12/2012 ,  
Probleme: Kosten (jährlich/einmalig/frei), Sicherheitslücken (Hijacking Benutzerkonten).

### 11.7.3 Gestaltung von Web 2.0-Anwendungen

#### Charakteristika von Web 2.0-Anwendungen

7 Prinzipien von Web 2.0-Anwendungen (Tim O'Reilly: What is web 2.0 (2005)):

- *Web als Plattform*: frei verfügbare API's, z.B. Google Maps.
- *Nutzung kollektiver Intelligenz*: Netzwerkeffekt (Wissen und Korrektiv). Bsp: Enzyklopädie Wikipedia, Auktionshaus ebay, Buchshops Amazon, Barnesandnoble.
- *Daten als Kernkompetenz*: Datenlieferanten, z.B. NavTeq (Kartenmaterial), und Verwalter des Datenaufkommens, z.B. Kontaktbörsen, Online-Bezahlsysteme (Paypal).
- *Software im ewigen Betastadium*: SW als Dienst, Wartung/Mitwirkung des Nutzers.
- *Lightweight Programming Models*: einfache statt komplexe Web Services, z.B. RSS oder REST; Einsatz vorhandener Komponenten (z.B. Google Maps / Flickr).
- *Geräteunabhängige Software*: Plattform für verschiedenste Geräte, unabh. Gerätetyp.
- *Rich User Experience (RUE)*: Anwendungen mit Nutzern erstellt, Benutzerführungen in Web-Browsern (Interaktionen). Bsp: GMail, Google Maps (AJAX-Technologie).

#### Vom Desktop zum Webtop

Übertragen der Funktionalität einer Desktop-Anwendung auf das Web (Web-Applikationen, sog. „Weblications“). Beispiele:

- Office-Suiten Google Docs & Spreadsheet und Zoho Virtual Office,
- Projektverwaltung Basecamp, Rechnungsverwaltung Blinksale,
- Web-PIM ...

Anwendungen arbeiten auf dem Server, nutzbar von jedem Client aus ~> einfache SW-Wartung, multiple Nutzung, einfacher Zugriff, gemeinsame Datennutzung (über Internet).

Bereits im Einsatz: Webfähige Email-Clients wie GMail, Yahoo Mail oder GMX.de.

Verlagerung der Verarbeitung global ins Netz, PC mehr zu reinen Terminals. Lokale Speicher, USB-Sticks, DVD's verlieren an Bedeutung.

Speicherplatz im Web angeboten (virtuelle Festplatten). Beispiel Amazon S3 (Amazon Simple Storage Service) ~> damit Daten von dezentraler Stelle aus überall zugreifbar.

(<http://aws.amazon.com/s3/>).

Probleme der Daten-Sicherheit und -integrität. Private Daten werden „öffentlich“, trotz abgesicherter Übertragung (SSL).

Auch Verlagerung der Verarbeitungskapazität ins Netz. Beispiele:

- Amazon EC2 (Amazon Elastic Compute Cloud). Dienst seit 2006 angeboten ~> Mieten temporärer Rechenkapazität. (<http://aws.amazon.com/ec2/>).
- IBM Cloud Computing (Cebit 2009) ~> leistungsstarke Rechner im Netz.

Somit Vision eines dezentralen, komplett integrierten Netzes: alle Teile (PC, Server, mobile Endgeräte, Anwendungen, Netzwerkkomponenten) dynamisch im Netz verteilt.

Ähnliche Konzeption bei Microsoft Betriebssystem Windows Azure (Strategiewechsel). Nutzung Plattform Internet: Zugriff auf Daten und Software über Internet auf riesige Datenspeicher (verbundene Server). Analog auch Microsoft Windows 8.

## 11.8 Beispiel: Formulare und CGI

### Aufgabenstellung

Systemtechnische Voraussetzungen:

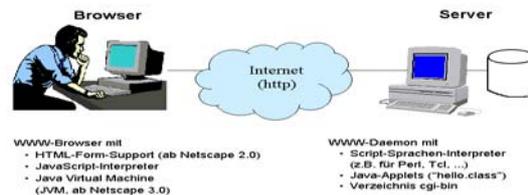


Abbildung 11.13: Beispiel (Voraussetzungen)

HTML-Dokument mit Formularen (Form-Tags) und daran geknüpften CGI-Mechanismus (Ausgabe Eingabefenster und 2 Buttons "SUBMIT", "RESET").

JavaScript-Code (Script-Tag) mit Ausgabe eines Fensters (Text "Hi Folks") unter Nutzung der alert-Funktion.

Java-Applet (vorcompiliertes Java-Programm mit Ausgabe "HI FOLKS").

CGI-Script (in Perl; Ausführung entsprechend gedrücktem Button).

### Editieren

HTML-Datei demo.html

Text-Editor (z.B. ed) oder HTML-Editor

Speichern <html> bzw. <htm>

Im Home-Verzeichnis des Nutzers, z.B. /home/nutzername/public\_html

Zugriff über rechnername/~nutzername (nutzt index.html)

PERL-Datei echo.pl.cgi

Text-Editor

Speichern /cgi-bin/echo.pl.cgi

Weitere Vorbereitungen: Konfigurieren Server:Angabe, in welchem Directory ausführbare Dateien oder Scripte stehen. Vorbereiten Applet: Applet steht in public\_html und ist über Dateiname hello.class erreichbar (Java-Klasse).

### HTML.Datei (Autor: Hendrik Schulze, Universität Leipzig, RNVS)

demo.html

```
<html>
<head>
<title>Tag Demos</title>
</head>
<body>
<!-- ***** CGI Section ***** -->
<!-- create formular and call CGI-Script -->
Insert a String:
<FORM METHOD="POST" ACTION="/cgi-bin/echo.pl.cgi">
<INPUTNAME="FORMULAR" TYPE="TEXT" SIZE="10" MAXLENGTH="20">
<P>
<Input TYPE="SUBMIT">
<Input TYPE="RESET">
</FORM>
<!-- ***** End of CGI-Section ***** -->
<!-- ***** JavaScript Section ***** -->
<script language="JavaScript">
<!--
/*JavaScript code....
```

```

eg:*/
alert ("Hi Folks");
//-->
</script>
<!-- ***** End of JavaScript Section ***** -->
<P>
<!-- ***** Java Section ***** -->
Output of Java Applet is:
<APPLET CODE="hello.class" WIDTH=150 HEIGHT=25>
</APPLET>
<!-- ***** End of Java Section ***** -->
</body>
</html>

```

**CGI-Script: echo.pl.cgi**

```

#!/usr/bin/perl
# read from standardinput
read(STDIN, $DATA, $ENV{'CONTENT_LENGTH'});
#create html-document
print "Content-type: text/html\n\n";
print "<html><head><title>CGI-ECHO</title></head>\n";
print "<body><h1>CGI-ECHO from <i>echo.pl.cgi</i></h1>\n";
#print received data
print "I got string: ", $DATA, "\n";
#complete html-document
print "</body></html>\n";

```

**1. Ausgabefenster**



Abbildung 11.14: Beispiel (Erstes Ausgabefenster)

**2. Ausgabefenster**



Abbildung 11.15: Beispiel (Zweites Ausgabefenster)

## Ablauf

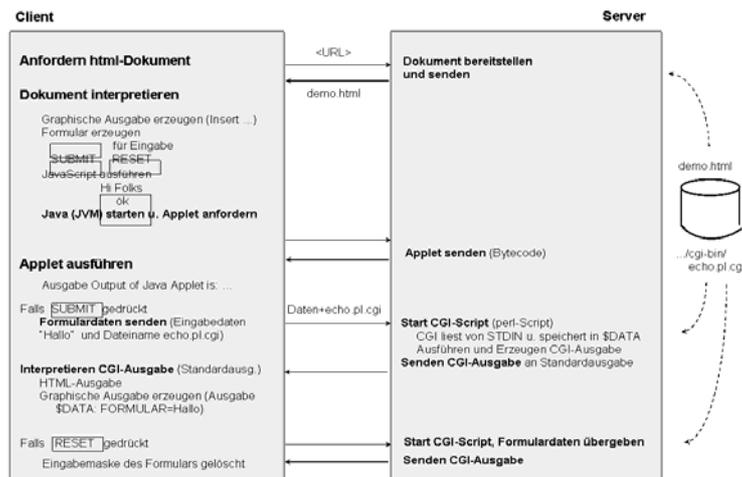


Abbildung 11.16: Beispiel (Ablauf der Interaktionen)

## Teil 2: Ausgewählte Netze

12. **Flächendeckende Netze (WAN)**
13. **Next Generation Internet**
14. **Lokale Rechnernetze (LAN)**
15. **Satellitennetze**
16. **Metropolitan Area Networks (MAN)**
17. **Entwicklung zur HighSpeed-Kommunikation**
18. **Mobilfunknetze**

## Teil 3: Übertragungssysteme

19. **Standardisiertes Breitbandnetz (B-ISDN/ATM)**
20. **Photonische Netze**
21. **Zugangnetzwerke (Access Networks)**
22. **ISDN – Integrated Services Digital Networks**
23. **Funkkanal**

## 12 Abbildungsverzeichnis (Teil IntW3)

Abbildung 10.1: Topologische Struktur im ARPAnet.....	4
Abbildung 10.2: NSFnet (Auszug) und Transatlantic Links.....	5
Abbildung 10.3: Protokollsuite des Internet .....	7
Abbildung 10.4: Paketvermittlungsnetz (Routing) .....	8
Abbildung 10.5: Autonome Systeme und Grenzrouter (Beispiel).....	9
Abbildung 10.6: Informationsdienste im Internet (Protokolle).....	10
Abbildung 10.7: Ausschnitt aus einem X.500 DIT .....	13
Abbildung 11.1: Client/Server-Architektur des WWW .....	18
Abbildung 11.2: Tomorrow's Web-Technologien.....	19
Abbildung 11.3: Komponenten und Ablauf im WWW .....	23
Abbildung 11.4: Architektur eines Web-Browsers .....	23
Abbildung 11.5: HTTP-Dienstprimitive .....	25
Abbildung 11.6: Hyperlinks im WWW .....	31
Abbildung 11.7: Interaktionen im WWW.....	34
Abbildung 11.8: JavaScript im Web-Client/Server-Modell.....	36
Abbildung 11.9: Java-Applet im Web-Client/Server-Modell.....	36
Abbildung 11.10: Ablaufgestaltung mit CGI-Script.....	37
Abbildung 11.11: Modelle von Web-Applikationen.....	44
Abbildung 11.12: AJAX-Modell einer Web-Anwendung (asynchrone Kommunikation).....	45
Abbildung 11.13: Beispiel (Voraussetzungen) .....	61
Abbildung 11.14: Beispiel (Erstes Ausgabefenster) .....	62
Abbildung 11.15: Beispiel (Zweites Ausgabefenster) .....	62
Abbildung 11.16: Beispiel (Ablauf der Interaktionen) .....	63

## 13 Literatur

- Comer, D.E.: The Internet Book. Prentice Hall, Englewood Cliffs, 1995
- Goldfarb, C.F.; Prescod, P.: Das XML Handbuch. Anwendungen, Produkte, Technologien. Verlag Addison-Wesley, München, 2000
- Herrmann, E.: CGI Programming with Perl 5. Sams Net, 1997
- Krol, Ed : Die Welt des Internet. Handbuch und Übersicht. O'Reilly / Int. Thomson, 1995
- Kröner, Peter: HTML5 – Webseiten innovativ und zukunftssicher. Open Source Press, 2011
- Kurose, J.F.; Ross, K.W.: Computernetze (Schwerpunkt Internet). Pearson, 2002  
Pearson Education/Addison Wesley, München, 2002
- Maurer, R.: HTML und CGI-Programmierung. Dynamische WWW-Seiten. dpunkt, 1997
- Morris, M.E.S.: HTML - WWW effektiv nutzen. dpunkt-Verlag, 1996
- Müller, G.; Eymann, T.; Kreutzer, M.: Telematik- und Kommunikationssysteme in der vernetzten Wirtschaft. Oldenbourg Verlag München/Wien, 2003
- Münz, S.: SelfHTML. URL: <http://selfhtml.teamone.de/> ~> für Selbststudium
- Peterson, L.L.; Davie, B.S.: Computernetze. dpunkt, 2000
- Pilgrim, Mark: Durchstarten mit HTML5. O'Reilly Verlag. 2010
- Rose, M.T.: TCP/IP-Netze. Carl-Hanser, München, 1994
- Tanenbaum, A.S.: Computer-Netzwerke. Pearson Studium, München, 2003
- Tolksdorf, R.: Die Sprachen des Web: HTML und XHTML. dpunkt, Heidelberg, 2000
- Wall, L.; Schwartz, R.L.: perl. O'Reilly & Associates, 1996
- Wilde, E.: World Wide Web. Technische Grundlagen. Springer, Heidelberg, 2000